Appendice B

Codice informatico elaborato per l'analisi dei dati

Sommario

In questa appendice sono riportate tutte le operazioni effettuate in ambiente R tramite RStudio versione 1.4 (https://www.rstudio.com), dove non specificato tutto il codice va inteso come relativo a R. Per alcune particolari operazioni sono invece stati utilizzati QGIS versione 3 (https://qgis.org/en/site) e GRASS GIS versione 7.6 (https://grass.osgeo.org), in questi casi il cambio di linguaggio è specificato nel testo di accompagnamento al codice.

Indice

1 Modello predittivo etnoarcheologico

Per il calcolo del modello predittivo si è operato attraverso RStudio, in connessione con GRASS GIS. Ci si è avvalsi dei seguenti componenti aggiuntivi per RStudio:

```
library(rgrass7)
library(sp)
library(raster)
use_sp()
library(rcompanion)
library(Matching)
library(car)
library(MASS)
library(spdep)
library(pgirmess)
library(ggplot2)
library(QuantPsyc)
library(pROC)
```

1.1 Preparazione dei dati spaziali

Le operazioni preliminari di manipolazione dei dati spaziali avvengono tramite **GRASS GIS 7.6**.

1.1.1 Impostazione del progetto e caricamento della cartografia

```
Database directory: (D:/Archeologia/Carona/GRASS)
Location: SBC
Projection: WGS84/UTM32N - EPSG: 32632
Mapset: Pred_Mod_SBC
```

importazione DTM 5X5 - Modello digitale del terreno prov. BG e SO - 2015 - Regione Lombardia)

```
r.in.gdal input=D:/Cartografia/EUROPA/ITALIA/LOMBARDIA/prov
BERGAMO/DTM/DTM5_BG.img output=DTM_BG
```

Importazione area di studio: shapefile prodotto tramite QGIS a partire da dati territoriali della Regione Lombardia (CTR, DTM, Limiti Amministrativi, Idrografia)

```
v.in.ogr input=D:/Archeologia/Carona/GIS/Area_Studio.shp
```

Impostazione risoluzione e regione computazionale

```
g.region
[vector=Area_Studio@Pred_Mod_SBC]
res=25
```

Creazione DTM relativo alla sola area di studio attraverso una maschera coincidente con l'area

```
r.mask
[vector=Area_Studio@Pred_Mod_SBC]
r.patch
[input=DTM_BG@Pred_Mod_SBC],[DTM_SO@Pred_Mod_SBC]
output=DTM_SBC
r.mask -r
[vector=Area_Studio@Pred_Mod_SBC]
```

Importazione e rasterizzazione dell'idrografia, a partire da dati vettoriali della Regione Lombardia, rimaneggiati in QGIS.

```
v.in.ogr
input=D:/Archeologia/Carona/GIS/Cartografia/Idrografia_Perenne_SBC.shp
v.in.ogr input=D:/Archeologia/Carona/GIS/Cartografia/Laghi_SBC.shp
v.to.rast
```

```
[input=Idrografia_Perenne_SBC@Pred_Mod_SBC]
output=Fiumi use=val
v.to.rast
[input=Laghi_SBC@Pred_Mod_SBC]
output=Laghi use=val
r.patch
[input=Laghi@Pred_Mod_SBC],[Fiumi@Pred_Mod_SBC]
output=Idrografia
```

Importazione e rasterizzazione della cartografia tematica, disponibile sul portale cartografico della Regione Lombardia.

```
v.in.ogr input=D:/Archeologia/Carona/GIS/Cartografia/Geologia_SBC.shp
v.in.ogr input=D:/Archeologia/Carona/GIS/Cartografia/Litologia_SBC.shp
v.in.ogr
input=D:/Archeologia/Carona/GIS/Cartografia/Permeabilita_SBC.shp
v.in.ogr input=D:/Archeologia/Carona/GIS/Cartografia/Valanghe_SBC.shp
v.to.rast
[input=Geologia_SBC@Pred_Mod_SBC]
output=Geologia use=attr attribute_column=SBC_COD
v.to.rast
[input=Litologia_SBC@Pred_Mod_SBC]
output=Litologia use=attr attribute_column=SBC_COD
v.to.rast
[input=Permeabilita_SBC@Pred_Mod_SBC]
output=Permeabilita use=attr attribute_column=SBC_COD
v.to.rast
[input=Valanghe_SBC@Pred_Mod_SBC]
output=Valanghe use=attr attribute_column=SBC_COD
```

1.1.2 Variabili indipendenti

Slope, Profile Curvature

```
r.slope.aspect
[elevation=DTM_SBC@Pred_Mod_SBC]
slope=Slope pcurvature=Profile
```

Northerness - Easterness (Esposizione)

```
r.northerness.easterness
[elevation=DTM_SBC@Pred_Mod_SBC]
```

Morphometric Features

```
r.param.scale
[input=DTM_SBC@Pred_Mod_SBC]
output=MorFeat method=feature
Parametri:1=Planar; 2=Pit; 3=Channel; 4=Pass(Saddle); 5=Ridge; 6=Peak
```

Topographic Wetness Index

```
r.topidx
[input=DTM_SBC@Pred_Mod_SBC]
output=TWI
```

Distanza dai fiumi e dai laghi *(calcolate separatamente come distanza di costo, derivata dalla Slope)*

```
r.cost
[input=Slope@Pred_Mod_SBC]
output=River_cost\
r.cost
[input=Slope@Pred_Mod_SBC]
```

```
output=Lakes_cost
```

1.1.3 Variabili dipendenti

Importazione dei punti delle strutture d'alpeggio ("siti"), posizionate in QGIS e creazione dei punti casuali da utilizzare come "non-siti", con maschera sulle posizioni dei siti e sui corpi d'acqua.

```
v.in.ogr input=D:/Archeologia/Carona/GIS/Siti/Alpeggi/Alpeggi_SBC.shp
v.to.rast
[input=Alpeggi_SBC@Pred_Mod_SBC]
output=Alpeggi use=val
r.patch
[input=Idrografia@Pred_Mod_SBC],[Alpeggi@Pred_Mod_SBC](
output=IdroAlpeggi r.mask -i --overwrite
[raster=IdroAlpeggi@Pred_Mod_SBC]
r.random --overwrite
[input=Slope@Pred_Mod_SBC]
npoints=90 vector=Random90
r.mask -r --overwrite
[raster=IdroAlpeggi@Pred_Mod_SBC]
```

1.2 Valutazione delle variabili

Caricamento delle variabili in RStudio e loro validazione statistica.

1.2.1 Upload delle variabili e creazione del dataset

```
initGRASS(gisBase = "C:/Program Files/GRASS GIS 7.6", gisDbase = "D:/Archeologia/Carona/GRASS",
    location = "SBC", mapset = "Pred_Mod_SBC", override = TRUE)
```

```
gisdbase    D:/Archeologia/Carona/GRASS
location    SBC
mapset      Pred_Mod_SBC
rows        221
columns     356
north       5100124
south       5094603
west        561597.4
east        570487.2
nsres       24.98108
ewres       24.97131
```

```
Carona <- stack(readRAST(c("DTM_SBC", "Slope", "TWI", "River_cost",
    "Lakes_cost", "North", "East", "Profile", "Geologia", "Litologia",
    "Permeabilita", "Valanghe", "MorFeat"), cat = c(F, F, F,
    F, F, F, F, F, T, T, T, T, T), plugin = F))

Sites <- readVECT("Alpeggi_SBC")
```

```
OGR data source with driver: GPKG
Source: "D:\Archeologia\Carona\GRASS\SBC\Pred_Mod_SBC\.tmp\unknown\453.0.gpkg", layer: "Alpeggi_SBC"
with 35 features
It has 13 fields
```

```
No_Sites <- readVECT("Random90")
```

```
OGR data source with driver: GPKG
Source: "D:\Archeologia\Carona\GRASS\SBC\Pred_Mod_SBC\.tmp\unknown\431.0.gpkg", layer: "Random90"
with 90 features
It has 2 fields
```

```
sitestab <- data.frame(extract(Carona, Sites), name = rep(1,
    length(Sites)))

nositestab <- data.frame(extract(Carona, No_Sites), name = rep(0,
    length(No_Sites)))

tab <- rbind(sitestab, nositestab)

tab$Geologia <- as.factor(tab$Geologia)
levels(tab$Geologia) <- c(1, 2, 3, 4)

tab$Litologia <- as.factor(tab$Litologia)
levels(tab$Litologia) <- c(1, 2, 3, 4)

tab$Permeabilita <- as.factor(tab$Permeabilita)
levels(tab$Permeabilita) <- c(1, 2, 3)

tab$Valanghe <- as.factor(tab$Valanghe)
levels(tab$Valanghe) <- c(0, 1, 2)

tab$MorFeat <- as.factor(tab$MorFeat)
levels(tab$MorFeat) <- c(1, 2, 3, 4, 5, 6)

table(is.na(tab))
```

```
FALSE
 1750
```

Separazione dei dati qualitativi nominali da quelli quantitativi continui

```
tab_cat <- data.frame(tab$Geologia, tab$Litologia, tab$Permeabilita,
    tab$Valanghe, tab$MorFeat, tab$name)
colnames(tab_cat) <- c("Geologia", "Litologia", "Permeabilita",
    "Valanghe", "MorFeat", "name")

table(is.na(tab_cat))
```

```
FALSE
  750
```

```
tab_num <- data.frame(tab$DTM_SBC, tab$Slope, tab$TWI, tab$River_cost,
    tab$Lakes_cost, tab$North, tab$East, tab$Profile, tab$name)
colnames(tab_num) <- c("DTM_SBC", "Slope", "TWI", "River_cost",
    "Lakes_cost", "North", "East", "Profile", "name")

table(is.na(tab_num))
```

```
FALSE
 1125
```

```
sites_num <- data.frame(sitestab$DTM_SBC, sitestab$Slope, sitestab$TWI,
    sitestab$River_cost, sitestab$Lakes_cost, sitestab$North,
    sitestab$East, sitestab$Profile)
colnames(sites_num) <- c("DTM_SBC", "Slope", "TWI", "River_cost",
    "Lakes_cost", "North", "East", "Profile")

nosites_num <- data.frame(nositestab$DTM_SBC, nositestab$Slope,
    nositestab$TWI, nositestab$River_cost, nositestab$Lakes_cost,
    nositestab$North, nositestab$East, nositestab$Profile)
```

```
colnames(nosites_num) <- c("DTM_SBC", "Slope", "TWI", "River_cost",
    "Lakes_cost", "North", "East", "Profile")

table(is.na(sites_num))
```

```
FALSE
  280
```

```
table(is.na(nosites_num))
```

```
FALSE
  720
```

```
sites_cat <- subset(tab_cat, name == 1)

nosites_cat <- subset(tab_cat, name == 0)
```

1.2.2 Grafici esplorativi

```
apply(sitestab, 2, range)
```

```
      DTM_SBC     Slope       TWI River_cost Lakes_cost
[1,] 1616.254  4.857067  4.387334   13.38833   17.67918
[2,] 2281.953 31.099014 10.295825 1110.29170 1211.71488
          North       East      Profile Geologia Litologia
[1,] -0.9999832 -0.9999688 -0.013962802        1         1
[2,]  0.9998110  0.9979231  0.007413552        4         4
     Permeabilita Valanghe MorFeat name
[1,]            1        1       1    1
[2,]            3        2       5    1
```

```
apply(nositestab, 2, range)
```

```
      DTM_SBC     Slope       TWI River_cost Lakes_cost
[1,] 1245.944  3.821155  2.846464   36.33603   37.57962
[2,] 2630.954 57.109684 10.707932 1613.74030 2095.20218
          North       East      Profile Geologia Litologia
[1,] -0.9999853 -0.9999814 -0.007485544        1         1
[2,]  0.9994192  0.9637925  0.012376428        4         4
     Permeabilita Valanghe MorFeat name
[1,]            1        1       1    0
[2,]            3        3       5    0
```

```
Area <- readVECT("Area_Studio")
```

```
OGR data source with driver: GPKG
Source: "D:\Archeologia\Carona\GRASS\SBC\Pred_Mod_SBC\.tmp\unknown\124.0.gpkg", layer: "Area_Studio"
with 1 features
It has 2 fields
```

```
# Cartografia
jpeg("Variabili_Carte.jpg", width = 6000, height = 4000, units = "px",
    res = 300)
par(oma = c(1, 1, 3, 1))
plot(Carona, cex.main = 1.2, main = c("Altitudine", "Pendenza",
    "Topographic Wetness Index", "Distanza dai fiumi (costo)",
    "Distanza dai laghi (costo)", "Esposizione NS", "Esposizione EW",
    "Curvatura del Profilo", "Geologia", "Litologia", "Permeabilita'",
    "Rischio Valanghe", "Morphometric Features"), asp = 1)
dev.off()
```

```r
# Istogrammi
jpeg("Variabili_Grafici.jpg", width = 5000, height = 6000, units = "px",
    res = 300)
par(oma = c(0, 0, 2, 0))
layout(matrix(c(1, 3, 5, 7, 9, 2, 4, 6, 8, 10, 11, 13, 15, 17,
    19, 12, 14, 16, 18, 20, 21, 23, 25, 27, 27, 22, 24, 26, 28,
    28), nrow = 6, byrow = TRUE))
# 1 Altititudine
hist(sitestab$DTM_SBC, xlab = "metri", main = "Altitudine - Siti",
    las = 1, col = "red", xlim = c(1200, 2700), ylim = c(0, 10),
    cex.axis = 0.8, breaks = c(seq(1200, 2700, 100)))
Axis(side = 1, at = seq(1200, 2700, by = 100), labels = FALSE)
abline(v = mean(sitestab$DTM_SBC), col = "white", lty = 1, lwd = 1)
abline(v = mean(sitestab$DTM_SBC), col = "black", lty = 2, lwd = 1)
# 2
hist(nositestab$DTM_SBC, xlab = "metri", main = "Altitudine - Non Siti",
    las = 1, col = "red4", xlim = c(1200, 2700), ylim = c(0,
        20), cex.axis = 0.8, breaks = c(seq(1200, 2700, 100)))
Axis(side = 1, at = seq(1200, 2700, by = 100), labels = FALSE)
abline(v = mean(nositestab$DTM_SBC), col = "white", lty = 1,
    lwd = 1)
abline(v = mean(nositestab$DTM_SBC), col = "black", lty = 2,
    lwd = 1)
# 3 Slope
hist(sitestab$Slope, xlab = "gradi", main = "Pendenza - Siti",
    las = 1, col = "greenyellow", xlim = c(0, 90), ylim = c(0,
        20), breaks = c(seq(0, 90, 10)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 90, by = 10), labels = FALSE)
abline(v = mean(sitestab$Slope), col = "white", lty = 1, lwd = 1)
abline(v = mean(sitestab$Slope), col = "black", lty = 2, lwd = 1)
# 4
hist(nositestab$Slope, xlab = "gradi", main = "Pendenza - Non Siti",
    las = 1, col = "green4", xlim = c(0, 90), ylim = c(0, 40),
    breaks = c(seq(0, 90, 10)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 90, by = 10), labels = FALSE)
abline(v = mean(nositestab$Slope), col = "white", lty = 1, lwd = 1)
abline(v = mean(nositestab$Slope), col = "black", lty = 2, lwd = 1)
# 5 TWI
hist(sitestab$TWI, xlab = "T.W.I.", main = "T.W.I. - Siti", las = 1,
    col = "lightblue4", xlim = c(0, 14), ylim = c(0, 10), breaks = c(seq(0,
        16, 1)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 14, by = 1), labels = F)
abline(v = mean(sitestab$TWI), col = "white", lty = 1, lwd = 1)
abline(v = mean(sitestab$TWI), col = "black", lty = 2, lwd = 1)
# 6
hist(nositestab$TWI, xlab = "T.W.I.", main = "T.W.I. - Non Siti",
    las = 1, col = "lightblue3", xlim = c(0, 14), ylim = c(0,
        25), breaks = c(seq(0, 14, 1)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 16, by = 1), labels = F)
abline(v = mean(nositestab$TWI), col = "white", lty = 1, lwd = 1)
abline(v = mean(nositestab$TWI), col = "black", lty = 2, lwd = 1)
# 7 Fiumi
hist(sitestab$River_cost, xlab = "Costo", main = "Distanza Fiumi - Siti",
    las = 1, col = "blue", xlim = c(0, 2000), ylim = c(0, 10),
    breaks = c(seq(0, 2000, 100)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 2000, by = 100), labels = F)
```

```
abline(v = mean(sitestab$River_cost), col = "white", lty = 1,
    lwd = 1)
abline(v = mean(sitestab$River_cost), col = "black", lty = 2,
    lwd = 1)
# 8
hist(nositestab$River_cost, xlab = "Costo", main = "Distanza Fiumi - Non Siti",
    las = 1, col = "blue4", xlim = c(0, 2000), ylim = c(0, 12),
    breaks = c(seq(0, 2000, 100)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 2000, by = 100), labels = F)
abline(v = mean(nositestab$River_cost), col = "white", lty = 1,
    lwd = 1)
abline(v = mean(nositestab$River_cost), col = "black", lty = 2,
    lwd = 1)
# 9 Laghi
hist(sitestab$Lakes_cost, xlab = "Costo", main = "Distanza Laghi - Siti",
    las = 1, col = "lightblue", xlim = c(0, 2500), ylim = c(0,
        10), breaks = c(seq(0, 2500, 100)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 2500, by = 100), labels = F)
abline(v = mean(sitestab$Lakes_cost), col = "white", lty = 1,
    lwd = 1)
abline(v = mean(sitestab$Lakes_cost), col = "black", lty = 2,
    lwd = 1)
# 10
hist(nositestab$Lakes_cost, xlab = "Costo", main = "Distanza Laghi - Non Siti",
    las = 1, col = "lightblue4", xlim = c(0, 2500), ylim = c(0,
        12), breaks = c(seq(0, 2500, 100)), cex.axis = 0.8)
Axis(side = 1, at = seq(0, 2500, by = 100), labels = F)
abline(v = mean(nositestab$Lakes_cost), col = "white", lty = 1,
    lwd = 1)
abline(v = mean(nositestab$Lakes_cost), col = "black", lty = 2,
    lwd = 1)
# 11 North
hist(sitestab$North, xlab = "aspect", main = "Esposizione N-S - Siti",
    las = 1, col = "yellow", ylim = c(0, 20))
abline(v = mean(sitestab$North), col = "white", lty = 1, lwd = 1)
abline(v = mean(sitestab$North), col = "black", lty = 2, lwd = 1)
# 12
hist(nositestab$North, xlab = "aspect", main = "Esposizione N-S - Non Siti",
    las = 1, col = "yellow4", breaks = 4, ylim = c(0, 50))
abline(v = mean(nositestab$North), col = "white", lty = 1, lwd = 1)
abline(v = mean(nositestab$North), col = "black", lty = 2, lwd = 1)
# 13 East
hist(sitestab$East, xlab = "aspect", main = "Esposizione E-W - Siti",
    las = 1, col = "tan", ylim = c(0, 15))
abline(v = mean(sitestab$East), col = "white", lty = 1, lwd = 1)
abline(v = mean(sitestab$East), col = "black", lty = 2, lwd = 1)
# 14
hist(nositestab$East, xlab = "aspect", main = "Esposizione E-W - Non Siti",
    las = 1, col = "tan3", breaks = 4, ylim = c(0, 30))
abline(v = mean(nositestab$East), col = "white", lty = 1, lwd = 1)
abline(v = mean(nositestab$East), col = "black", lty = 2, lwd = 1)
# 15 Profile Curvature
hist(sitestab$Profile, xlab = "curvature", main = "Curvatura del Profilo - Siti",
    las = 1, col = "lightpink", xlim = c(-0.02, 0.02), ylim = c(0,
        15), breaks = c(seq(-0.02, 0.02, 0.005)), cex.axis = 0.8)
Axis(side = 1, at = seq(-0.02, 0.02, by = 0.005), labels = FALSE)
```

```
abline(v = mean(sitestab$Profile), col = "white", lty = 1, lwd = 1)
abline(v = mean(sitestab$Profile), col = "black", lty = 2, lwd = 1)
# 16
hist(nositestab$Profile, xlab = "curvature", main = "Curvatura del Profilo - Non Siti",
    las = 1, col = "lightpink3", xlim = c(-0.02, 0.02), ylim = c(0,
        50), breaks = c(seq(-0.02, 0.02, 0.005)), cex.axis = 0.8)
Axis(side = 1, at = seq(-0.02, 0.02, by = 0.005), labels = FALSE)
abline(v = mean(nositestab$Profile), col = "white", lty = 1,
    lwd = 1)
abline(v = mean(nositestab$Profile), col = "black", lty = 2,
    lwd = 1)
# 17 Geologia
plot(sites_cat$Geologia, xlab = "categoria", ylab = "Count",
    main = "Geologia - Siti", las = 1, col = "darkolivegreen1",
    ylim = c(0, 20), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 25, by = 5), labels = FALSE)
# 18
plot(nosites_cat$Geologia, xlab = "categoria", ylab = "Count",
    main = "Geologia - Non Siti", las = 1, col = "darkolivegreen4",
    ylim = c(0, 50), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 50, by = 10), labels = FALSE)
# 19 Litologia
plot(sites_cat$Litologia, xlab = "categoria", ylab = "Count",
    main = "Litologia - Siti", las = 1, col = "gold", ylim = c(0,
        30), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 30, by = 5), labels = FALSE)
# 20
plot(nosites_cat$Litologia, xlab = "categoria", ylab = "Count",
    main = "Litologia - Non Siti", las = 1, col = "gold4", ylim = c(0,
        50), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 50, by = 10), labels = FALSE)
# 21 Permeabilita
plot(sites_cat$Permeabilita, xlab = "categoria", ylab = "Count",
    main = "Permeabilita' - Siti", las = 1, col = "lightslateblue",
    ylim = c(0, 30), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 30, by = 5), labels = FALSE)
# 22
plot(nosites_cat$Permeabilita, xlab = "categoria", ylab = "Count",
    main = "Permeabilita' - Non Siti", las = 1, col = "violetred4",
    ylim = c(0, 50), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 50, by = 10), labels = FALSE)
# 23 Valanghe
plot(sites_cat$Valanghe, xlab = "categoria", ylab = "Count",
    main = "Rischio Valanghe - Siti", las = 1, col = "violetred1",
    ylim = c(0, 30), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 30, by = 5), labels = FALSE)
# 24
plot(nosites_cat$Valanghe, xlab = "categoria", ylab = "Count",
    main = "Rischio Valanghe - Non Siti", las = 1, col = "violetred4",
    ylim = c(0, 50), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 50, by = 10), labels = FALSE)
# 25 Morphometric Feat
plot(sites_cat$MorFeat, xlab = "category", ylab = "Count", main = "Morph. Feat. - Siti",
    las = 1, col = "gray", ylim = c(0, 25), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 25, by = 5), labels = FALSE)
# 26
```

```
plot(nosites_cat$MorFeat, xlab = "category", ylab = "Count",
    main = "Morph. Feat. - Non Siti", las = 1, col = "gray51",
    ylim = c(0, 50), cex.axis = 0.8)
Axis(side = 2, at = seq(0, 50, by = 10), labels = FALSE)
# 27 Siti
plot(Area, main = "Posizionamento Siti")
plot(Sites, pch = 20, col = "blue", cex = 0.5, add = T)
# 28 Non siti
plot(Area, main = "Posizionamento Non Siti")
plot(No_Sites, pch = 20, col = "red", cex = 0.5, add = T)
dev.off()
```

1.2.3 Collinearità

Funzione per il calcolo della collinearità

```
collin <- function(x) {
    for (i in 1:ncol(x)) {
        b <- x[, 1:i]
        rescol <- cor(x, b, method = "pearson")
        rownames(rescol) <- colnames(x)
    }
    return(rescol)
}
```

Calcolo della collinearità

```
coltab <- collin(tab_num)
colround <- round(coltab, digits = 4)
```

Tabella 1: Collinearità

	DTM_SBC	Slope	TWI	River_cost	Lakes_cost	North	East	Profile	name
DTM_SBC	1.0000	0.3119	-0.3577	0.0839	-0.0793	-0.1458	0.2216	0.2364	-0.3443
Slope	0.3119	1.0000	-0.5622	0.5529	0.5292	-0.0987	0.0698	0.1224	-0.5744
TWI	-0.3577	-0.5622	1.0000	-0.1554	-0.1981	-0.0510	0.0126	-0.3270	0.3181
River_cost	0.0839	0.5529	-0.1554	1.0000	0.5973	-0.1876	0.0772	0.0112	-0.2207
Lakes_cost	-0.0793	0.5292	-0.1981	0.5973	1.0000	-0.3186	-0.0349	0.0517	-0.2156
North	-0.1458	-0.0987	-0.0510	-0.1876	-0.3186	1.0000	-0.1224	0.0575	-0.0042
East	0.2216	0.0698	0.0126	0.0772	-0.0349	-0.1224	1.0000	0.1375	-0.0014
Profile	0.2364	0.1224	-0.3270	0.0112	0.0517	0.0575	0.1375	1.0000	-0.2020
name	-0.3443	-0.5744	0.3181	-0.2207	-0.2156	-0.0042	-0.0014	-0.2020	1.0000

1.2.4 Kolmogorov-Smirnov test

Funzione per applicare il K-S test in bootstrapping alle variabili numeriche continue. Ipotesi nulla = identità con una distribuzione casuale.
Soglia p.value = 0.05

```
# x = siti y = non siti nsim = numero di ripetizioni
ksb.col <- function(x, y, nsim) {
    ksb_tab <- list()
    for (i in 1:ncol(x)) {
        ksb_tab[[i]] <- ks.boot(x[, i], y[, i], nboots = nsim)
    }
```

```
    names(ksb_tab) <- colnames(x)
    return(lapply(ksb_tab, summary))
}
```

Esecuzione del test in bootstrapping (10.000 ripetizioni)

```
ks_file <- capture.output(ksb.col(sites_num, nosites_num, 10000))
ks_file
```

```
 [1] ""
 [2] "Bootstrap p-value:     2e-04 "
 [3] "Naive p-value:         1.9686e-05 "
 [4] "Full Sample Statistic: 0.46349 "
 [5] ""
 [6] ""
 [7] "Bootstrap p-value:     < 2.22e-16 "
 [8] "Naive p-value:         2.3435e-11 "
 [9] "Full Sample Statistic: 0.66984 "
[10] ""
[11] ""
[12] "Bootstrap p-value:     2e-04 "
[13] "Naive p-value:         0.00036715 "
[14] "Full Sample Statistic: 0.40159 "
[15] ""
[16] ""
[17] "Bootstrap p-value:     0.0405 "
[18] "Naive p-value:         0.044904 "
[19] "Full Sample Statistic: 0.26667 "
[20] ""
[21] ""
[22] "Bootstrap p-value:     0.0631 "
[23] "Naive p-value:         0.072128 "
[24] "Full Sample Statistic: 0.24921 "
[25] ""
[26] ""
[27] "Bootstrap p-value:     0.6096 "
[28] "Naive p-value:         0.63906 "
[29] "Full Sample Statistic: 0.14127 "
[30] ""
[31] ""
[32] "Bootstrap p-value:     0.5708 "
[33] "Naive p-value:         0.59854 "
[34] "Full Sample Statistic: 0.14603 "
[35] ""
[36] ""
[37] "Bootstrap p-value:     0.1041 "
[38] "Naive p-value:         0.11661 "
[39] "Full Sample Statistic: 0.23016 "
[40] ""
[41] "$DTM_SBC"
[42] ""
[43] "$Slope"
[44] ""
[45] "$TWI"
[46] ""
[47] "$River_cost"
[48] ""
```

```
[49] "$Lakes_cost"
[50] ""
[51] "$North"
[52] ""
[53] "$East"
[54] ""
[55] "$Profile"
[56] ""
```

Esclusione delle variabili: *Lakes_cost* (distanza di costo dai laghi); *North* (esposizione a Nord/Sud); *East* (esposizione a Est/Ovest); *Profile* (Profile Curvature).

1.2.5 Test del chi-quadro

Preparazione dei dati relativi alle aree delle singole categorie, per ogni variabile. Il dato è raccolto interrogando la cartografia in GRASS GIS.
Ipotesi nulla = identità con una distribuzione casuale.
Soglia p.value = 0.05

```
Geo_kmq <- c(14.312701, 2.493993, 5.367263, 9.683405)
Lit_kmq <- c(12.932209, 0.485948, 2.225755, 16.21345)
Perm_kmq <- c(16.189745, 6.132054, 9.535562)
Vala_kmq <- c(18.212762, 8.811319, 4.833281)
Mft_kmq <- c(0.682448, 0.017467, 15.487959, 0.023705, 14.807382,
    0.003119)
```

Funzione per l'applicazione del test alle variabili nominali

```
csq.arp <- function(x, y) {
    s <- data.frame(table(factor(x, levels = 1:length(y))))
    n <- data.frame(y/sum(y))
    t <- data.frame(cbind(s, n))
    colnames(t) <- c("cat", "sites", "pexp")
    return(chisq.test(t$sites, p = t$pexp))
}
```

Test del chi-quadro (Siti)

```
# Geologia
geocstest <- csq.arp(sitestab[, "Geologia"], Geo_kmq)
geofile <- capture.output(geocstest)
pergeo <- (Geo_kmq/sum(Geo_kmq))
geotab <- cbind(geocstest$observed, geocstest$expected, pergeo)
colnames(geotab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
9	15.724608	0.4492745
2	2.740018	0.0782862
5	5.896728	0.1684779
19	10.638645	0.3039613

```
geofile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
```

```
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 9.7835, df = 3, p-value = 0.0205"
[6] ""
```

```
# Litologia
litcstest <- csq.arp(sitestab[, "Litologia"], Lit_kmq)
litofile <- capture.output(litcstest)
perlit <- (Lit_kmq/sum(Lit_kmq))
littab <- cbind(litcstest$observed, litcstest$expected, perlit)
colnames(littab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
4	14.2079346	0.4059410
0	0.5338854	0.0152539
1	2.4453194	0.0698663
30	17.8128607	0.5089389

```
litofile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 17.06, df = 3, p-value = 0.0006868"
[6] ""
```

```
# Permeabilit\340
permcstest <- csq.arp(sitestab[, "Permeabilita"], Perm_kmq)
permfile <- capture.output(permcstest)
perperm <- (Perm_kmq/sum(Perm_kmq))
permtab <- cbind(permcstest$observed, permcstest$expected, perperm)
colnames(permtab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
21	17.786818	0.5081948
7	6.736964	0.1924847
7	10.476218	0.2993205

```
permfile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 1.7442, df = 2, p-value = 0.4181"
[6] ""
```

```
# Rischio valanghe
valacstest <- csq.arp(sitestab[, "Valanghe"], Vala_kmq)
valafile <- capture.output(valacstest)
pervala <- (Vala_kmq/sum(Vala_kmq))
valatab <- cbind(valacstest$observed, valacstest$expected, pervala)
colnames(valatab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
30	20.00940	0.5716971
5	9.68053	0.2765866
0	5.31007	0.1517163

```
valafile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 12.561, df = 2, p-value = 0.001872"
[6] ""
```

```
# Elementi Morfometrici
mftcstest <- csq.arp(sitestab[, "MorFeat"], Mft_kmq)
morffile <- capture.output(mftcstest)
permft <- (Mft_kmq/sum(Mft_kmq))
mfttab <- cbind(mftcstest$observed, mftcstest$expected, permft)
colnames(mfttab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
2	0.7699574	0.0219988
0	0.0197068	0.0005631
9	17.4739594	0.4992560
0	0.0267447	0.0007641
24	16.7061129	0.4773175
0	0.0035189	0.0001005

```
morffile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 9.309, df = 5, p-value = 0.09736"
[6] ""
```

Esclusione delle variabili: *Permeabilita* (Permeabilità) e *MorFeat* (Elementi Morfometrici)

Applicazione del test del chi-quadro ai "non-siti"

```
# Geologia
ngeocstest <- csq.arp(nositestab[, "Geologia"], Geo_kmq)
ngeofile <- capture.output(ngeocstest)
npergeo <- (Geo_kmq/sum(Geo_kmq))
ngeotab <- cbind(ngeocstest$observed, ngeocstest$expected, npergeo)
colnames(ngeotab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
42	40.434707	0.4492745
5	7.045761	0.0782862
11	15.163015	0.1684779
32	27.356517	0.3039613

```
ngeofile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 2.5857, df = 3, p-value = 0.46"
[6] ""
```

```
# Litologia
nlitcstest <- csq.arp(nositestab[, "Litologia"], Lit_kmq)
nlitofile <- capture.output(nlitcstest)
```

```
nperlit <- (Lit_kmq/sum(Lit_kmq))
nlittab <- cbind(nlitcstest$observed, nlitcstest$expected, nperlit)
colnames(nlittab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
39	36.534689	0.4059410
1	1.372848	0.0152539
8	6.287964	0.0698663
42	45.804499	0.5089389

```
nlitofile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 1.0498, df = 3, p-value = 0.7892"
[6] ""
```

```
# Permeabilit\340
npermcstest <- csq.arp(nositestab[, "Permeabilita"], Perm_kmq)
npermofile <- capture.output(npermcstest)
nperperm <- (Perm_kmq/sum(Perm_kmq))
npermtab <- cbind(npermcstest$observed, npermcstest$expected,
    nperperm)
colnames(npermtab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
44	45.73753	0.5081948
18	17.32362	0.1924847
28	26.93885	0.2993205

```
npermofile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 0.13422, df = 2, p-value = 0.9351"
[6] ""
```

```
# Rischio valanghe
nvalacstest <- csq.arp(nositestab[, "Valanghe"], Vala_kmq)
nvalaofile <- capture.output(nvalacstest)
write("Rischio Valanghe", file = "Chi_SQ_test_NS.txt", append = T)
npervala <- (Vala_kmq/sum(Vala_kmq))
nvalatab <- cbind(nvalacstest$observed, nvalacstest$expected,
    npervala)
colnames(nvalatab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
48	51.45274	0.5716971
32	24.89279	0.2765866

observed	expected	%areaexp
10	13.65447	0.1517163

```
nvalaofile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 3.239, df = 2, p-value = 0.198"
[6] ""
```

```
# Elementi Morfometrici
nmftcstest <- csq.arp(nositestab[, "MorFeat"], Mft_kmq)
nmorffile <- capture.output(nmftcstest)
npermft <- (Mft_kmq/sum(Mft_kmq))
nmfttab <- cbind(nmftcstest$observed, nmftcstest$expected, npermft)
colnames(nmfttab) <- c("observed", "expected", "%areaexp")
```

observed	expected	%areaexp
2	1.9798905	0.0219988
0	0.0506746	0.0005631
47	44.9330383	0.4992560
0	0.0687720	0.0007641
41	42.9585760	0.4773175
0	0.0090487	0.0001005

```
nmorffile
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 0.31308, df = 5, p-value = 0.9974"
[6] ""
```

Per tutte le variabili associate ai non-siti non è possibile rigettare l'ipotesi nulla.

1.3 Calcolo del modello predittivo

1.3.1 Regressione Logistica Univariata

Funzioni per la regressione logistica univariata, in due versioni: con e senza la presentazione dei risultati in forma testuale.

```
univar <- function(x, y) {
    lrunivar <- list()
    for (i in x) {
        f <- formula(paste("name", "~", i))
        lrunivar[[i]] <- glm(f, data = y, family = binomial(logit))
    }
    return(lrunivar)
}

univar.s <- function(x, y) {
    lrunivar <- list()
    for (i in x) {
        f <- formula(paste("name", "~", i))
        lrunivar[[i]] <- summary(glm(f, data = y, family = binomial(logit)))
    }
    return(lrunivar)
}
```

Calcolo della regressione

```
indep <- c("DTM_SBC", "Slope", "TWI", "River_cost", "Geologia",
    "Litologia", "Valanghe")

unifile <- capture.output(univar.s(indep, tab))
glm_uni <- univar(indep, tab)
glm_uni
```

```
$DTM_SBC

Call:  glm(formula = f, family = binomial(logit), data = y)

Coefficients:
(Intercept)      DTM_SBC
    5.17331     -0.00308

Degrees of Freedom: 124 Total (i.e. Null);  123 Residual
Null Deviance:      148.2
Residual Deviance: 132.7    AIC: 136.7

$Slope

Call:  glm(formula = f, family = binomial(logit), data = y)

Coefficients:
(Intercept)        Slope
     3.5872      -0.2002

Degrees of Freedom: 124 Total (i.e. Null);  123 Residual
Null Deviance:      148.2
Residual Deviance: 93.35    AIC: 97.35

$TWI
```

```
Call:  glm(formula = f, family = binomial(logit), data = y)

Coefficients:
(Intercept)          TWI
    -4.0104       0.4677

Degrees of Freedom: 124 Total (i.e. Null);  123 Residual
Null Deviance:      148.2
Residual Deviance: 135.5    AIC: 139.5

$River_cost

Call:  glm(formula = f, family = binomial(logit), data = y)

Coefficients:
(Intercept)   River_cost
  -0.287679    -0.001374

Degrees of Freedom: 124 Total (i.e. Null);  123 Residual
Null Deviance:      148.2
Residual Deviance: 141.7    AIC: 145.7

$Geologia

Call:  glm(formula = f, family = binomial(logit), data = y)

Coefficients:
(Intercept)    Geologia2    Geologia3    Geologia4
    -1.5404       0.6242       0.7520       1.0191

Degrees of Freedom: 124 Total (i.e. Null);  121 Residual
Null Deviance:      148.2
Residual Deviance: 143.1    AIC: 151.1

$Litologia

Call:  glm(formula = f, family = binomial(logit), data = y)

Coefficients:
(Intercept)   Litologia2   Litologia3   Litologia4
    -2.2773     -13.2888       0.1978       1.9408

Degrees of Freedom: 124 Total (i.e. Null);  121 Residual
Null Deviance:      148.2
Residual Deviance: 130.7    AIC: 138.7

$Valanghe

Call:  glm(formula = f, family = binomial(logit), data = y)

Coefficients:
(Intercept)    Valanghe1    Valanghe2
     -0.470       -1.386      -17.096

Degrees of Freedom: 124 Total (i.e. Null);  122 Residual
Null Deviance:      148.2
```

```
Residual Deviance: 133.2    AIC: 139.2
```

```
nulldev <- sapply(glm_uni, function(x) {
    x$null.deviance
})
resdev <- sapply(glm_uni, function(x) {
    x$deviance
})
ratio <- capture.output(1 - (pchisq(nulldev - resdev, 1)))
ratio
```

Likelihood Ratio Test

```
[1] "     DTM_SBC        Slope          TWI   River_cost "
[2] "8.055267e-05 1.274536e-13 3.649724e-04 1.049518e-02 "
[3] "    Geologia    Litologia     Valanghe "
[4] "2.385105e-02 2.812563e-05 1.079426e-04 "
```

```r
jpeg("Univariate_plot.jpeg", width = 4000, height = 1000, units = "px",
    res = 300)
par(oma = c(0, 0, 2, 0))
layout(matrix(c(1, 2, 3, 4), nrow = 1, byrow = TRUE))
# 1
plot(tab$DTM_SBC, jitter(tab$name, 0.1), ylim = c(-0, 1), ylab = "",
    xlab = "")
par(new = T)
plot(tab$DTM_SBC, fitted(glm_uni$DTM_SBC), col = "red", pch = 16,
    type = "p", ylim = c(-0, 1), ylab = "Site Presence", xlab = "Altitude")
# 2
plot(tab$Slope, jitter(tab$name, 0.1), ylim = c(-0, 1), ylab = "",
    xlab = "")
par(new = T)
plot(tab$Slope, fitted(glm_uni$Slope), col = "red", pch = 16,
    type = "p", ylim = c(-0, 1), ylab = "Site Presence", xlab = "Slope")
# 3
plot(tab$River_cost, jitter(tab$name, 0.1), ylim = c(-0, 1),
    ylab = "", xlab = "")
par(new = T)
plot(tab$River_cost, fitted(glm_uni$River_cost), col = "red",
    pch = 16, type = "p", ylim = c(-0, 1), ylab = "Site Presence",
    xlab = "Distance from Rivers")
# 4
plot(tab$TWI, jitter(tab$name, 0.1), ylim = c(-0, 1), ylab = "",
    xlab = "")
par(new = T)
plot(tab$TWI, fitted(glm_uni$TWI), col = "red", pch = 16, type = "p",
    ylim = c(-0, 1), xlab = "Topographic Wetness Index", ylab = "Site Presence")

mtext("Logistic Univariate Regression", outer = TRUE, cex = 1.5)
dev.off()

# Plot Univariate n2

jpeg("Univariate_plot2.jpeg", width = 3000, height = 3000, units = "px",
    res = 300)
par(oma = c(0, 0.5, 2, 0))
layout(matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE))
# 1
plot(tab$DTM_SBC, jitter(tab$name, 0.1), ylim = c(-0, 1), ylab = "",
    xlab = "")
par(new = T)
plot(tab$DTM_SBC, fitted(glm_uni$DTM_SBC), col = "red", pch = 16,
    type = "p", ylim = c(-0, 1), ylab = "Site Presence", xlab = "Altitude")
# 2
plot(tab$Slope, jitter(tab$name, 0.1), ylim = c(-0, 1), ylab = "",
    xlab = "")
par(new = T)
plot(tab$Slope, fitted(glm_uni$Slope), col = "red", pch = 16,
    type = "p", ylim = c(-0, 1), ylab = "Site Presence", xlab = "Slope")
# 3
plot(tab$River_cost, jitter(tab$name, 0.1), ylim = c(-0, 1),
    ylab = "", xlab = "")
par(new = T)
```

```
plot(tab$River_cost, fitted(glm_uni$River_cost), col = "red",
    pch = 16, type = "p", ylim = c(-0, 1), ylab = "Site Presence",
    xlab = "Distance from Rivers")
# 4
plot(tab$TWI, jitter(tab$name, 0.1), ylim = c(-0, 1), ylab = "",
    xlab = "")
par(new = T)
plot(tab$TWI, fitted(glm_uni$TWI), col = "red", pch = 16, type = "p",
    ylim = c(-0, 1), xlab = "Topographic Wetness Index", ylab = "Site Presence")

mtext("Logistic Univariate Regression", outer = TRUE, cex = 1.5)
dev.off()
```

Resa grafica dei risultati

1.3.2 Regressione Logistica Multivariata

```
glm_mul <- glm(name ~ DTM_SBC + Slope + TWI + River_cost + Geologia +
    Litologia + Valanghe, data = tab, family = binomial(logit))

glmfile <- capture.output(summary(glm_mul))
glmfile
```

```
 [1] ""
 [2] "Call:"
 [3] "glm(formula = name ~ DTM_SBC + Slope + TWI + River_cost + Geologia + "
 [4] "    Litologia + Valanghe, family = binomial(logit), data = tab)"
 [5] ""
 [6] "Deviance Residuals: "
 [7] "     Min        1Q    Median        3Q       Max  "
 [8] "-2.20353  -0.39170  -0.06428   0.31097   2.07228  "
 [9] ""
[10] "Coefficients:"
[11] "              Estimate Std. Error z value Pr(>|z|)    "
[12] "(Intercept)  1.103e+01  4.537e+00   2.432   0.0150 *  "
[13] "DTM_SBC     -2.784e-03  1.384e-03  -2.011   0.0443 *  "
[14] "Slope       -2.837e-01  6.978e-02  -4.065 4.81e-05 ***"
[15] "TWI         -2.003e-01  2.495e-01  -0.803   0.4220    "
[16] "River_cost   1.525e-03  1.240e-03   1.230   0.2189    "
[17] "Geologia2   -6.031e-01  1.610e+00  -0.375   0.7079    "
[18] "Geologia3   -6.776e-02  1.109e+00  -0.061   0.9513    "
[19] "Geologia4   -5.859e-01  7.670e-01  -0.764   0.4449    "
[20] "Litologia2  -1.565e+01  6.523e+03  -0.002   0.9981    "
[21] "Litologia3  -7.227e-01  1.556e+00  -0.465   0.6423    "
[22] "Litologia4   1.347e+00  9.077e-01   1.484   0.1379    "
[23] "Valanghe1    7.647e-02  9.326e-01   0.082   0.9346    "
[24] "Valanghe2   -1.768e+01  1.641e+03  -0.011   0.9914    "
[25] "---"
[26] "Signif. codes:  "
[27] "0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1"
[28] ""
[29] "(Dispersion parameter for binomial family taken to be 1)"
[30] ""
[31] "    Null deviance: 148.238  on 124  degrees of freedom"
[32] "Residual deviance:  68.847  on 112  degrees of freedom"
[33] "AIC: 94.847"
[34] ""
```

```
[35] "Number of Fisher Scoring iterations: 17"
[36] ""
```

Variance inflation factor

```
vif_MUL <- vif(glm_mul)
vifile <- capture.output(vif_MUL)
vifile
```

```
[1] "               GVIF Df GVIF^(1/(2*Df))"
[2] "DTM_SBC    1.351004  1        1.162327"
[3] "Slope      2.646221  1        1.626721"
[4] "TWI        1.564455  1        1.250782"
[5] "River_cost 2.374989  1        1.541100"
[6] "Geologia   1.625853  3        1.084377"
[7] "Litologia  1.371380  3        1.054046"
[8] "Valanghe   1.839186  2        1.164545"
```

Standardizzazione dei coefficienti

```
bxm <- summary(glm_mul)$coef[-1, 1]

ngkmul <- nagelkerke(glm_mul)
Rmul <- sqrt(ngkmul$Pseudo.R.squared.for.model.vs.null[3, 1])
ngkmul$Likelihood.ratio.test
```

```
 Df.diff LogLik.diff  Chisq    p.value
     -12     -39.696 79.392 5.3904e-12
```

```
sxm_dtm <- sd(as.numeric(unlist(glm_mul$model[2])))
sxm_slope <- sd(as.numeric(unlist(glm_mul$model[3])))
sxm_TWI <- sd(as.numeric(unlist(glm_mul$model[4])))
sxm_river <- sd(as.numeric(unlist(glm_mul$model[5])))
sxm_geo <- sd(as.numeric(unlist(glm_mul$model[6])))
sxm_lito <- sd(as.numeric(unlist(glm_mul$model[7])))
sxm_vala <- sd(as.numeric(unlist(glm_mul$model[8])))

sxm <- cbind(sxm_dtm, sxm_slope, sxm_TWI, sxm_river, sxm_geo,
    sxm_geo, sxm_geo, sxm_lito, sxm_lito, sxm_lito, sxm_vala,
    sxm_vala)

sym <- sd(as.numeric(unlist(glm_mul$model[1])))

(bxm * sxm * Rmul)/sym
```

```
       sxm_dtm sxm_slope    sxm_TWI sxm_river   sxm_geo
[1,] -1.411698 -6.064525 -0.5761692  1.132491 -1.515542
       sxm_geo   sxm_geo  sxm_lito  sxm_lito sxm_lito
[1,] -0.170283 -1.472368 -39.98937 -1.847094 3.442302
       sxm_vala  sxm_vala
[1,] 0.08951173 -20.69004
```

1.3.3 Akaike Information Criterion

```
AIC_glm <- stepAIC(glm_mul)
```

```
Start:  AIC=94.85
name ~ DTM_SBC + Slope + TWI + River_cost + Geologia + Litologia +
    Valanghe
```

```
             Df Deviance     AIC
- Geologia    3   69.601  89.601
- Litologia   3   73.215  93.215
- TWI         1   69.513  93.513
- River_cost  1   70.380  94.380
<none>            68.847  94.847
- Valanghe    2   74.576  96.576
- DTM_SBC     1   72.990  96.990
- Slope       1  100.824 124.824

Step:  AIC=89.6
name ~ DTM_SBC + Slope + TWI + River_cost + Litologia + Valanghe

             Df Deviance     AIC
- Litologia   3   73.579  87.579
- TWI         1   70.524  88.524
- River_cost  1   70.959  88.959
<none>            69.601  89.601
- DTM_SBC     1   73.728  91.728
- Valanghe    2   75.921  91.921
- Slope       1  102.927 120.927

Step:  AIC=87.58
name ~ DTM_SBC + Slope + TWI + River_cost + Valanghe

             Df Deviance     AIC
- TWI         1   75.287  87.287
<none>            73.579  87.579
- River_cost  1   75.783  87.783
- Valanghe    2   81.229  91.229
- DTM_SBC     1   81.347  93.347
- Slope       1  113.311 125.311

Step:  AIC=87.29
name ~ DTM_SBC + Slope + River_cost + Valanghe

             Df Deviance     AIC
<none>            75.287  87.287
- River_cost  1   77.445  87.445
- Valanghe    2   81.872  89.872
- DTM_SBC     1   81.458  91.458
- Slope       1  118.190 128.190
```

```
aicfile <- capture.output(summary(AIC_glm))
aicfile
```

```
 [1] ""
 [2] "Call:"
 [3] "glm(formula = name ~ DTM_SBC + Slope + River_cost + Valanghe, "
 [4] "    family = binomial(logit), data = tab)"
 [5] ""
 [6] "Deviance Residuals: "
 [7] "    Min       1Q   Median       3Q      Max  "
 [8] "-2.4801  -0.4178  -0.1133   0.3385   1.7829  "
 [9] ""
[10] "Coefficients:"
[11] "              Estimate Std. Error z value Pr(>|z|)    "
```

```
[12] "(Intercept)  9.965e+00  2.789e+00   3.573 0.000353 ***"
[13] "DTM_SBC     -2.978e-03  1.235e-03  -2.411 0.015889 *  "
[14] "Slope       -2.504e-01  5.380e-02  -4.654 3.26e-06 ***"
[15] "River_cost   1.592e-03  1.091e-03   1.459 0.144521    "
[16] "Valanghe1    3.840e-02  8.186e-01   0.047 0.962591    "
[17] "Valanghe2   -1.773e+01  1.642e+03  -0.011 0.991386    "
[18] "---"
[19] "Signif. codes:  "
[20] "0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1"
[21] ""
[22] "(Dispersion parameter for binomial family taken to be 1)"
[23] ""
[24] "    Null deviance: 148.238  on 124  degrees of freedom"
[25] "Residual deviance:  75.287  on 119  degrees of freedom"
[26] "AIC: 87.287"
[27] ""
[28] "Number of Fisher Scoring iterations: 17"
[29] ""
```

1.3.4 Bayesian Information Criterion

```
BIC_glm <- stepAIC(glm_mul, k = log(length(tab$name)))
```

```
Start:  AIC=131.61
name ~ DTM_SBC + Slope + TWI + River_cost + Geologia + Litologia +
    Valanghe

             Df Deviance    AIC
- Geologia    3   69.601 117.89
- Litologia   3   73.215 121.50
- TWI         1   69.513 127.45
- Valanghe    2   74.576 127.69
- River_cost  1   70.380 128.32
- DTM_SBC     1   72.990 130.93
<none>            68.847 131.62
- Slope       1  100.824 158.76

Step:  AIC=117.88
name ~ DTM_SBC + Slope + TWI + River_cost + Litologia + Valanghe

             Df Deviance    AIC
- Litologia   3   73.579 107.38
- TWI         1   70.524 113.98
- River_cost  1   70.959 114.41
- Valanghe    2   75.921 114.55
- DTM_SBC     1   73.728 117.18
<none>            69.601 117.89
- Slope       1  102.927 146.38

Step:  AIC=107.38
name ~ DTM_SBC + Slope + TWI + River_cost + Valanghe

             Df Deviance    AIC
- TWI         1   75.287 104.26
- River_cost  1   75.783 104.75
- Valanghe    2   81.229 105.37
<none>            73.579 107.38
```

```
- DTM_SBC     1   81.347 110.32
- Slope       1  113.311 142.28

Step:  AIC=104.26
name ~ DTM_SBC + Slope + River_cost + Valanghe

             Df Deviance    AIC
- Valanghe    2   81.872 101.19
- River_cost  1   77.445 101.59
<none>            75.287 104.26
- DTM_SBC     1   81.458 105.60
- Slope       1  118.190 142.33

Step:  AIC=101.19
name ~ DTM_SBC + Slope + River_cost

             Df Deviance     AIC
- River_cost  1   84.531  99.016
<none>            81.872 101.185
- DTM_SBC     1   88.311 102.796
- Slope       1  125.454 139.939

Step:  AIC=99.02
name ~ DTM_SBC + Slope

          Df Deviance     AIC
<none>         84.531  99.016
- DTM_SBC  1   93.349 103.005
- Slope    1  132.693 142.350
```

```
bicfile <- capture.output(summary(BIC_glm))
summary(BIC_glm)
```

```
Call:
glm(formula = name ~ DTM_SBC + Slope, family = binomial(logit),
    data = tab)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2358  -0.4733  -0.1727   0.4213   2.0143

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 10.047158   2.574629   3.902 9.53e-05 ***
DTM_SBC     -0.003264   0.001155  -2.826  0.00472 **
Slope       -0.202717   0.041422  -4.894 9.88e-07 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 148.238  on 124  degrees of freedom
Residual deviance:  84.531  on 122  degrees of freedom
AIC: 90.531

Number of Fisher Scoring iterations: 6
```

```
round(BIC_glm$coefficients, digits = 8)
```

```
(Intercept)     DTM_SBC       Slope
10.04715789 -0.00326407 -0.20271673
```

Le variabili selezionate sono l'**altitudine** e l'**inclinazione dei versanti**.
Dato che questo è il modello più parsimonioso verrà utilizzato per tutte le operazioni successive.

Standardizzazione dei coefficienti

```
bx <- summary(BIC_glm)$coef[-1, 1]

ngkbic <- nagelkerke(BIC_glm)
Rbic <- sqrt(ngkbic$Pseudo.R.squared.for.model.vs.null[3, 1])

sx_dtm <- sd(as.numeric(unlist(BIC_glm$model[2])))
sx_slope <- sd(as.numeric(unlist(BIC_glm$model[3])))
sx <- cbind(sx_dtm, sx_slope)
sy <- sd(as.numeric(unlist(BIC_glm$model[1])))

(bx * sx * Rbic)/sy
```

```
        sx_dtm  sx_slope
[1,] -1.525598 -3.994118
```

1.3.5 Analisi dei residui

Applicata soltanto ai risultati della Regressione Logistica Multivariata BIC

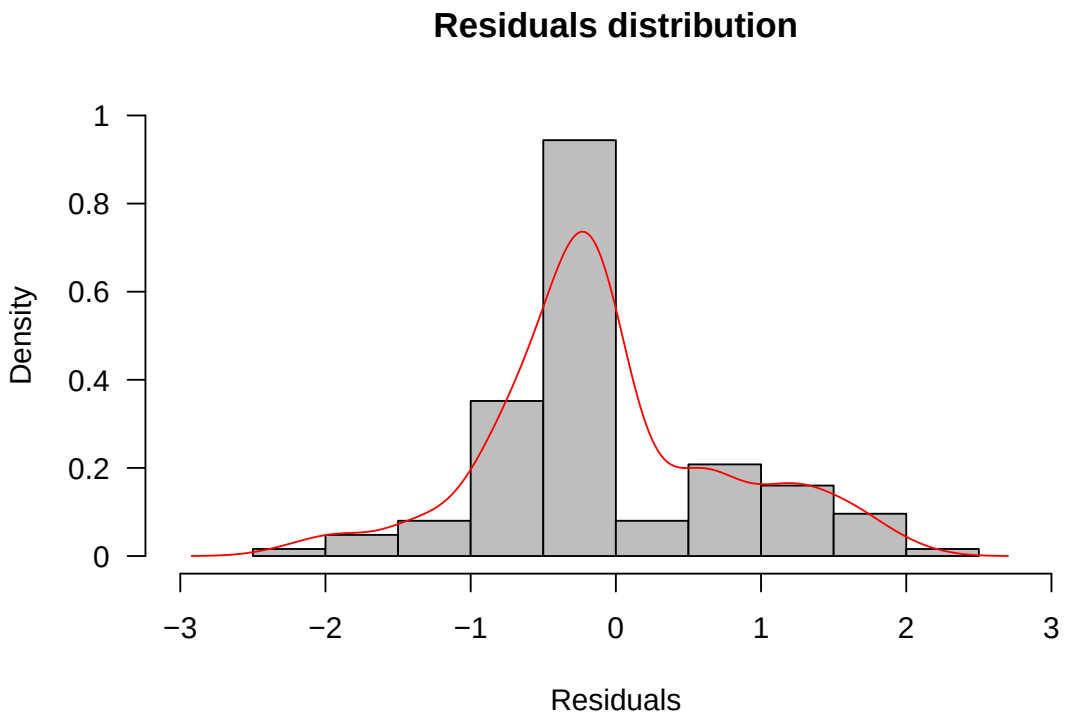
```
BIC_glm_resid <- residuals(BIC_glm)
bicresfile <- capture.output(summary(BIC_glm_resid))
min(BIC_glm_resid)
```

```
[1] -2.235811
```

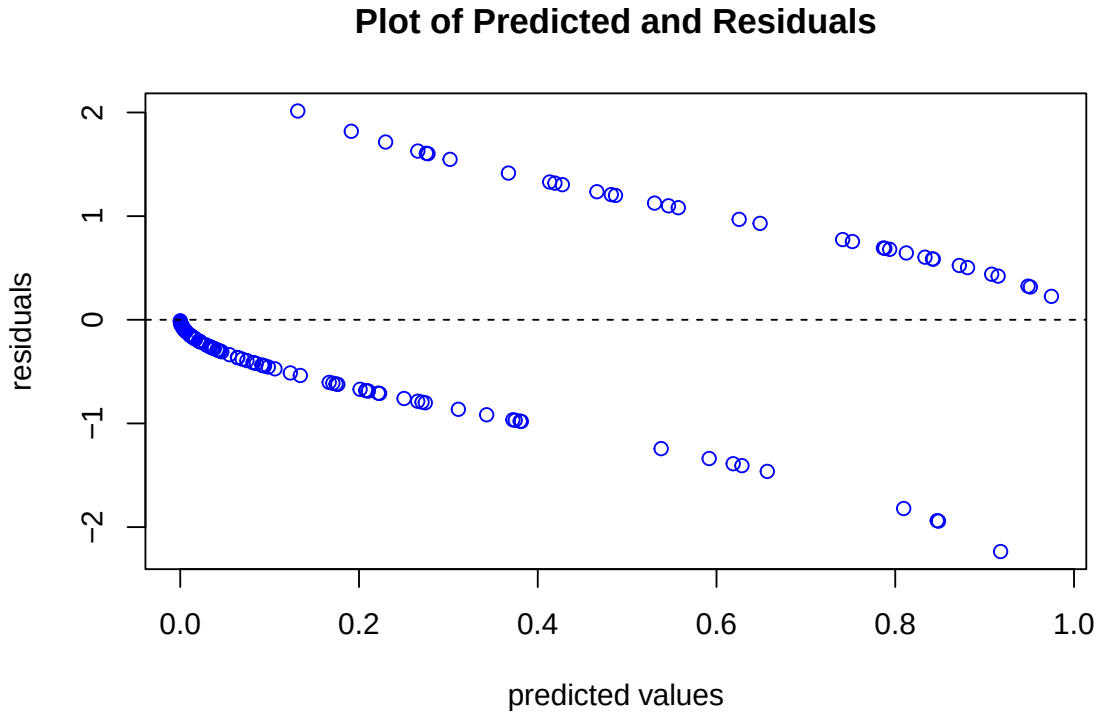
```
max(BIC_glm_resid)
```

```
[1] 2.014352
```

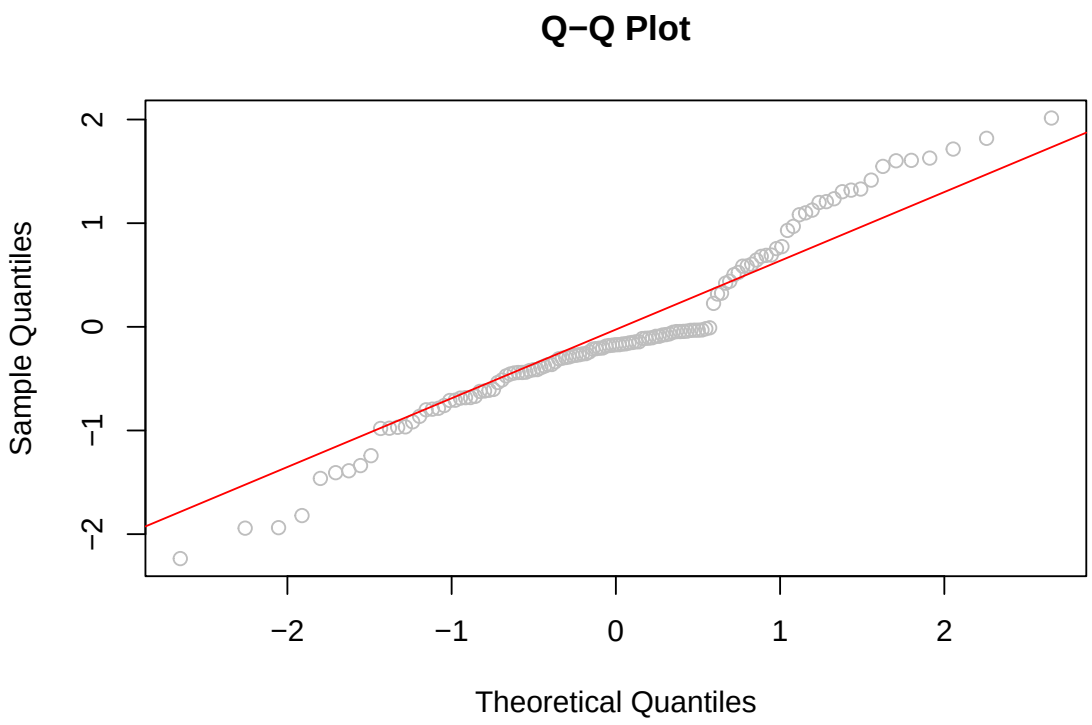
```
hist(BIC_glm_resid, las = 1, col = "gray", ylim = c(0, 1), xlim = c(-3,
    3), main = "Residuals distribution", xlab = "Residuals",
    axes = F, mar = c(0, 0, 1, 0), freq = F)
lines(density(BIC_glm_resid), col = "red")
Axis(side = 1, at = seq(-3, 3, by = 1), labels = seq(-3, 3, 1))
Axis(side = 2, at = seq(0, 1, by = 0.2), labels = seq(0, 1, 0.2),
    las = 1)
```




```
plot(fitted(BIC_glm), BIC_glm_resid, xlab = "predicted values",
    ylab = "residuals", main = "Plot of Predicted and Residuals",
    col = "blue")
abline(h = 0, lty = 2)
```



```
qqnorm(BIC_glm_resid, col = "gray", main = "Q-Q Plot")
qqline(BIC_glm_resid, col = "red")
```




```
site_coor <- as.data.frame(coordinates(Sites))
nosite_coor <- as.data.frame(coordinates(No_Sites))
coor <- as.matrix(rbind(site_coor, nosite_coor))

library(spdep)

# 1st Nearest Neighbour
nb1 <- knn2nb(knearneigh(coor, k = 1))
moran1file <- capture.output(moran.test(BIC_glm_resid, listw = nb2listw(nb1,
    style = "W")))
moran1file
```

Autocorrelazione spaziale

```
 [1] ""
 [2] "\tMoran I test under randomisation"
 [3] ""
 [4] "data:  BIC_glm_resid  "
 [5] "weights: nb2listw(nb1, style = \"W\")    "
 [6] ""
 [7] "Moran I statistic standard deviate = 0.73188, p-value"
 [8] "= 0.2321"
 [9] "alternative hypothesis: greater"
[10] "sample estimates:"
[11] "Moran I statistic       Expectation          Variance "
[12] "      0.075534240      -0.008064516       0.013047243 "
[13] ""
```

```r
# 2nd Nearest Neighbour
nb2 <- knn2nb(knearneigh(coor, k = 2))
moran2file <- capture.output(moran.test(BIC_glm_resid, listw = nb2listw(nb2,
    style = "W")))
moran2file
```

```
 [1] ""
 [2] "\tMoran I test under randomisation"
 [3] ""
 [4] "data:  BIC_glm_resid  "
 [5] "weights: nb2listw(nb2, style = \"W\")    "
 [6] ""
 [7] "Moran I statistic standard deviate = 0.69884, p-value"
 [8] "= 0.2423"
 [9] "alternative hypothesis: greater"
[10] "sample estimates:"
[11] "Moran I statistic       Expectation          Variance "
[12] "      0.048790758      -0.008064516       0.006618857 "
[13] ""
```

```r
# 3rd Nearest Neighbour
nb3 <- knn2nb(knearneigh(coor, k = 3))
moran3file <- capture.output(moran.test(BIC_glm_resid, listw = nb2listw(nb3,
    style = "W")))
moran3file
```

```
 [1] ""
 [2] "\tMoran I test under randomisation"
 [3] ""
 [4] "data:  BIC_glm_resid  "
 [5] "weights: nb2listw(nb3, style = \"W\")    "
 [6] ""
 [7] "Moran I statistic standard deviate = 0.239, p-value ="
 [8] "0.4056"
 [9] "alternative hypothesis: greater"
[10] "sample estimates:"
[11] "Moran I statistic       Expectation          Variance "
[12] "      0.008028049      -0.008064516       0.004533709 "
[13] ""
```

```r
# 4th Nearest Neighbour
nb4 <- knn2nb(knearneigh(coor, k = 4))
moran4file <- capture.output(moran.test(BIC_glm_resid, listw = nb2listw(nb4,
    style = "W")))
moran4file
```

```
 [1] ""
 [2] "\tMoran I test under randomisation"
 [3] ""
 [4] "data:  BIC_glm_resid  "
 [5] "weights: nb2listw(nb4, style = \"W\")    "
 [6] ""
 [7] "Moran I statistic standard deviate = -0.05725,"
 [8] "p-value = 0.5228"
 [9] "alternative hypothesis: greater"
[10] "sample estimates:"
[11] "Moran I statistic       Expectation          Variance "
[12] "     -0.011413531      -0.008064516       0.003422024 "
```

```
[13] ""
```

```
library(pgirmess)

resid_correl2 <- data.frame(correlog(coor, BIC_glm_resid, method = "Moran"))
class(resid_correl2)
```

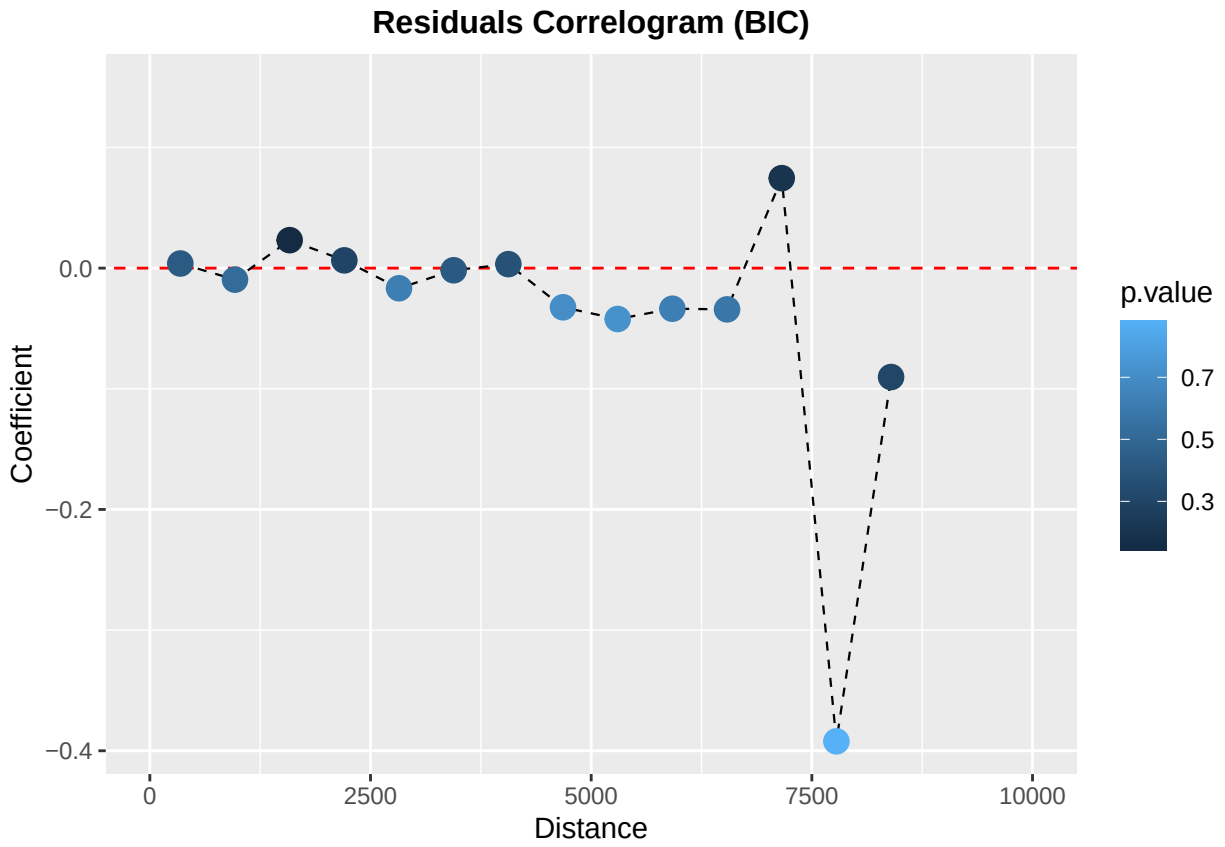
Correlogramma dei residui

```
[1] "data.frame"
```

```
resid_correl2
```

```
   dist.class         coef   p.value    n
1    345.0368  0.003827096 0.4269700  610
2    964.4672 -0.009541040 0.5167949 1670
3   1583.8975  0.023163189 0.1427952 2158
4   2203.3278  0.006447263 0.2983835 2370
5   2822.7581 -0.016693674 0.6228827 2298
6   3442.1884 -0.001761906 0.4196335 1904
7   4061.6188  0.003311204 0.3759657 1536
8   4681.0491 -0.032455852 0.7013438 1178
9   5300.4794 -0.042164871 0.7235024  860
10  5919.9097 -0.033648105 0.6272314  508
11  6539.3401 -0.034169568 0.5807538  250
12  7158.7704  0.074565156 0.1992776  118
13  7778.2007 -0.392196147 0.8831910   28
14  8397.6310 -0.090265810 0.3063590   12
```

```
library(ggplot2)

ggplot(resid_correl2, aes(x = dist.class, y = coef)) + geom_abline(intercept = 0,
    slope = 0, linetype = "dashed", colour = "red") + geom_line(linetype = "dashed",
    size = 0.4) + geom_point(data = resid_correl2, aes(colour = p.value),
    size = 4) + labs(title = "Residuals Correlogram (BIC)") +
    xlab("Distance") + ylab("Coefficient") + theme(plot.title = element_text(hjust = 0.5,
    size = 12, face = "bold")) + expand_limits(x = c(0, 10000),
    y = c(-0.15, 0.15))
```




```
# Valori dei residui
range(BIC_glm_resid)
```

```
[1] -2.235811  2.014352
```

```
summary(BIC_glm_resid)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.23581 -0.47329 -0.17269 -0.07008  0.42131  2.01435
```

```
length(BIC_glm_resid)
```

```
[1] 125
```

Report grafico dell'analisi dei residui

```
jpeg("Residuals_BIC.jpg", width = 4000, height = 1000, units = "px",
    res = 300)
par(oma = c(0, 0, 2, 0))
layout(matrix(c(1, 2, 3, 4), nrow = 1, byrow = TRUE))
# istogramma
hist(BIC_glm_resid, las = 1, col = "gray", ylim = c(0, 1), xlim = c(-3,
    3), main = "Residuals distribution", xlab = "Residuals",
    axes = F, mar = c(0, 0, 1, 0), freq = F)
x1 <- seq(-2, 2, length = 125)
y1 <- dnorm(x1)
lines(x1, y1, col = "blue", lwd = 2)
lines(density(BIC_glm_resid), col = "red", lwd = 2)
Axis(side = 1, at = seq(-4, 4, by = 1), labels = seq(-4, 4, 1))
Axis(side = 2, at = seq(0, 1, by = 0.2), labels = seq(0, 1, 0.2),
    las = 1)
```

```
# pred values
plot(fitted(BIC_glm), BIC_glm_resid, xlab = "predicted values",
    ylab = "residuals", main = "Plot of Predicted and Residuals",
    col = "blue")
abline(h = 0, lty = 2)
# Q-Q plot
qqnorm(BIC_glm_resid, col = "gray", main = "Q-Q Plot")
qqline(BIC_glm_resid, col = "red")
# residual correlogram
plot(resid_correl2$dist.class, resid_correl2$coef, ylim = c(-0.4,
    0.2), xlim = c(0, 10000), main = "Residuals Correlogram",
    xlab = "Distance", ylab = "Coefficient", axes = F, type = "l",
    col = "black")
par(new = T)
plot(resid_correl2$dist.class, resid_correl2$coef, ylim = c(-0.4,
    0.2), xlim = c(0, 10000), main = "", xlab = "", ylab = "",
    axes = F, type = "p", col = "blue")
Axis(side = 1, at = seq(0, 10000, by = 1000), labels = seq(0,
    10000, 1000))
Axis(side = 2, at = seq(-0.4, 0.2, by = 0.1), labels = seq(-0.4,
    0.2, 0.1), las = 1)
abline(h = 0, lty = 2, col = "red")
# titolo
mtext("Multivariate Regression Residuals (BIC)", outer = TRUE,
    cex = 1.5)
dev.off()

# 2 righe

jpeg("Residuals_BIC2r.jpg", width = 4000, height = 4000, units = "px",
    res = 300)
par(oma = c(0, 0.5, 2, 0))
layout(matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE))
# istogramma
hist(BIC_glm_resid, las = 1, col = "gray", ylim = c(0, 1), xlim = c(-3,
    3), main = "Residuals distribution", xlab = "Residuals",
    axes = F, mar = c(0, 0, 1, 0), freq = F)
x1 <- seq(-2, 2, length = 125)
y1 <- dnorm(x1)
lines(x1, y1, col = "blue", lwd = 2)
lines(density(BIC_glm_resid), col = "red", lwd = 2)
Axis(side = 1, at = seq(-4, 4, by = 1), labels = seq(-4, 4, 1))
Axis(side = 2, at = seq(0, 1, by = 0.2), labels = seq(0, 1, 0.2),
    las = 1)

# pred values
plot(fitted(BIC_glm), BIC_glm_resid, xlab = "predicted values",
    ylab = "residuals", main = "Plot of Predicted and Residuals",
    col = "blue")
abline(h = 0, lty = 2)
# Q-Q plot
qqnorm(BIC_glm_resid, col = "gray", main = "Q-Q Plot")
qqline(BIC_glm_resid, col = "red")
# residual correlogram
plot(resid_correl2$dist.class, resid_correl2$coef, ylim = c(-0.4,
```

```
    0.2), xlim = c(0, 10000), main = "Residuals Correlogram",
    xlab = "Distance", ylab = "Coefficient", axes = F, type = "l",
    col = "black")
par(new = T)
plot(resid_correl2$dist.class, resid_correl2$coef, ylim = c(-0.4,
    0.2), xlim = c(0, 10000), main = "", xlab = "", ylab = "",
    axes = F, type = "p", col = "blue")
Axis(side = 1, at = seq(0, 10000, by = 1000), labels = seq(0,
    10000, 1000))
Axis(side = 2, at = seq(-0.4, 0.2, by = 0.1), labels = seq(-0.4,
    0.2, 0.1), las = 1)
abline(h = 0, lty = 2, col = "red")
# titolo
mtext("Multivariate Regression Residuals (BIC)", outer = TRUE,
    cex = 1.5)
dev.off()
```

1.4 Creazione della superficie predittiva

In GRASS GIS si procede all'applicazione spaziale dei risultati della regressione logistica multivariata BIC.

```
r.mask -i
[raster=Laghi@Pred_Mod_SBC]
r.mapcalc --overwrite
expression=Regression = 10.047158 + ([DTM_SBC@Pred_Mod_SBC]\*-0.003264)
+([Slope@Pred_Mod_SBC]\*-0.202717)
r.mapcalc --overwrite
r.mapcalc expression=Probability_Surface = (exp([Regression@Pred_Mod_SBC]))
/(1+(exp([Regression@Pred_Mod_SBC] ) ))
```

Creazione delle mappe di probabilità categorizzate (step di .1)

```
r.mapcalc expression=Prob01 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.1\
r.mapcalc expression=Prob02 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.2\
r.mapcalc expression=Prob03 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.3\
r.mapcalc expression=Prob04 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.4\
r.mapcalc expression=Prob05 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.5\
r.mapcalc expression=Prob06 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.6\
r.mapcalc expression=Prob07 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.7\
r.mapcalc expression=Prob08 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0.8\
r.mapcalc expression=Prob09 =
[Probability_Surface@Pred_Mod_SBC]
```

```
\>= 0.9\
r.mapcalc expression=Prob10 =
[Probability_Surface@Pred_Mod_SBC]
\>= 1\
r.mapcalc --overwrite expression=Prob00 =
[Probability_Surface@Pred_Mod_SBC]
\>= 0\
```

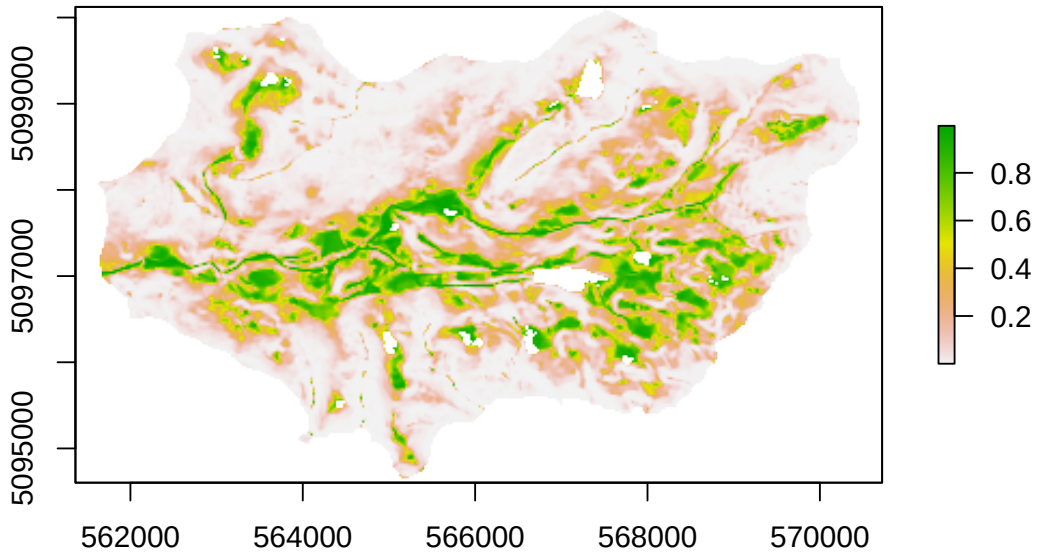
1.5 Validazione interna del modello

Caricamento e validazione della superficie predittiva e lettura del valore degli eventi.

```
predsurf <- raster(readRAST("Probability_Surface", plugin = F))
class(predsurf)
```

```
[1] "RasterLayer"
attr(,"package")
[1] "raster"
```

```
plot(predsurf)
```




```
sitespred <- data.frame(extract(predsurf, Sites))
str(sitespred)
```

```
'data.frame':   35 obs. of  1 variable:
 $ extract.predsurf..Sites.: num  0.951 0.949 0.915 0.302 0.419 ...
```

```
summary(is.na(sitespred))
```

```
 extract.predsurf..Sites.
 Mode :logical
 FALSE:35
```

```
summary(sitespred)
```

```
 extract.predsurf..Sites.
 Min.   :0.1315
 1st Qu.:0.4163
 Median :0.6254
 Mean   :0.6082
 3rd Qu.:0.8375
 Max.   :0.9748
```

```
class(sitespred)
```

```
[1] "data.frame"
```

```
nositespred <- data.frame(extract(predsurf, No_Sites))
str(nositespred)
```

```
'data.frame':   90 obs. of  1 variable:
 $ extract.predsurf..No_Sites.: num  0.005476 0.092593 0.000165 0.092413 0.000451 ...
```

```
summary(is.na(nositespred))
```

```
 extract.predsurf..No_Sites.
 Mode :logical
 FALSE:90
```

```
summary(nositespred)
```

```
 extract.predsurf..No_Sites.
 Min.   :0.0000433
 1st Qu.:0.0115144
 Median :0.0508388
 Mean   :0.1523893
 3rd Qu.:0.2080381
 Max.   :0.9178790
```

```
class(nositespred)
```

```
[1] "data.frame"
```

1.5.1 Test di Kolmogorov-Smirnov

```
library(Matching)

ksb.col <- function(x, y, nsim) {
    ksb_tab <- list()
    for (i in 1:ncol(x)) {
        ksb_tab[[i]] <- ks.boot(x[, i], y[, i], nboots = nsim)
    }
    names(ksb_tab) <- colnames(x)
    return(lapply(ksb_tab, summary))
}

kspredfile <- capture.output(ksb.col(sitespred, nositespred,
    10000))
kspredfile
```

```
[1] ""
[2] "Bootstrap p-value:     < 2.22e-16 "
[3] "Naive p-value:         9.3925e-14 "
```

```
[4] "Full Sample Statistic: 0.73175 "
[5] ""
[6] "$extract.predsurf..Sites."
[7] ""
```

1.5.2 Calcolo del *Gain*

```
a <- length(sitespred[sitespred > 0.5])
b <- length(sitespred[sitespred < 0.5])
c <- length(nositespred[nositespred > 0.5])
d <- length(nositespred[nositespred < 0.5])
n <- a + b + c + d
n1 <- a + b
n2 <- c + d
a
```

```
[1] 21
```

```
b
```

```
[1] 14
```

```
c
```

```
[1] 9
```

```
d
```

```
[1] 81
```

```
n1
```

```
[1] 35
```

```
n2
```

```
[1] 90
```

```
n
```

```
[1] 125
```

```
t <- (n * ((a * d) - (b * c))^2)/((n1 * n2) * (a + c) * (b +
    d))
t
```

```
[1] 34.53947
```

```
pval <- pchisq(t, df = 1, lower.tail = F)
pval
```

```
[1] 4.176942e-09
```

1.5.3 *Kvamme's Gain*

Calcolo del KG per i siti.

```
# Mappe predittive categorizzate
Cat_Pred <- (stack(readRAST(c("Prob01", "Prob02", "Prob03", "Prob04",
    "Prob05", "Prob06", "Prob07", "Prob08", "Prob09"), cat = c(T,
    T, T, T, T, T, T, T, T), plugin = F)) - 1)
```

```
Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended
```

```
Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended
```

```
# calcolo dei siti nelle aree di predizione numero siti per
# categoria

sitespredcat <- data.frame(extract(Cat_Pred, Sites))
str(sitespredcat)
```

```
'data.frame':   35 obs. of  9 variables:
 $ Prob01: num  1 1 1 1 1 1 1 1 1 1 ...
 $ Prob02: num  1 1 1 1 1 1 1 0 1 1 ...
 $ Prob03: num  1 1 1 1 1 0 1 0 1 1 ...
 $ Prob04: num  1 1 1 0 1 0 1 0 1 1 ...
 $ Prob05: num  1 1 1 0 0 0 1 0 1 1 ...
 $ Prob06: num  1 1 1 0 0 0 1 0 0 1 ...
 $ Prob07: num  1 1 1 0 0 0 0 0 0 1 ...
 $ Prob08: num  1 1 1 0 0 0 0 0 0 1 ...
 $ Prob09: num  1 1 1 0 0 0 0 0 0 1 ...
```

```
totsites <- colSums(sitespredcat)

# % siti per ogni step di predizione
percsit <- round((totsites/length(Sites)), digits = 2)

## calcolo delle aree di predizione
areapredcat <- rbind((freq(Cat_Pred, useNA = "no", merge = T)),
    (colSums(freq(Cat_Pred, useNA = "no", merge = T))))
percarea <- round((areapredcat[2, ]/areapredcat[3, ]), digits = 2)
```

```
percarea[, 1] <- NULL

## Calcolo del Kvamme's gain

Sites_KG <- (1 - (percarea/percsit))

KG_sites <- rbind(percarea, percsit, Sites_KG)
colnames(KG_sites) <- c(">0.1", ">0.2", ">0.3", ">0.4", ">0.5",
    ">0.6", ">0.7", ">0.8", ">0.9")
row.names(KG_sites) <- c("% Area", "% Siti", "Kvamme's Gain")
```

Tabella 12: Kvamme's Gain Siti

	>0.1	>0.2	>0.3	>0.4	>0.5	>0.6	>0.7	>0.8	>0.9
% Area	0.42	0.30	0.24	0.19	0.15	0.11	0.08	0.05	0.02
% Siti	1.00	0.94	0.83	0.77	0.60	0.51	0.46	0.31	0.14
Kvamme's Gain	0.58	0.68	0.71	0.75	0.75	0.78	0.83	0.84	0.86

Calcolo del KG per i non-siti

```
# numero non-siti predetti come siti
nositespredcat <- data.frame(extract(Cat_Pred, No_Sites))
str(nositespredcat)
```

```
'data.frame':   90 obs. of  9 variables:
 $ Prob01: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob02: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob03: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob04: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob05: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob06: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob07: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob08: num  0 0 0 0 0 0 0 0 0 ...
 $ Prob09: num  0 0 0 0 0 0 0 0 0 ...
```

```
wrong_ns <- colSums(nositespredcat)

# % non-siti predetti come siti
percnosit <- round((wrong_ns/length(No_Sites)), digits = 2)

# area inversa
areapredcat
```

```
  value Prob01 Prob02 Prob03 Prob04 Prob05 Prob06 Prob07
1     0  28636  34229  37493  39923  41928  43528  44936
2     1  20458  14865  11601   9171   7166   5566   4158
3     1  49094  49094  49094  49094  49094  49094  49094
  Prob08 Prob09
1  46444  47890
2   2650   1204
3  49094  49094
```

```
percnoarea <- round((areapredcat[1, ]/areapredcat[3, ]), digits = 2)
percnoarea[, 1] <- NULL

percarea + percnoarea
```

```
  Prob01 Prob02 Prob03 Prob04 Prob05 Prob06 Prob07 Prob08
2      1      1      1      1      1      1      1      1
  Prob09
2      1
```

```
# non siti predetti come non siti

right_ns <- length(No_Sites) - colSums(nositespredcat)
(wrong_ns + right_ns)/length(No_Sites)
```

```
Prob01 Prob02 Prob03 Prob04 Prob05 Prob06 Prob07 Prob08
     1      1      1      1      1      1      1      1
Prob09
     1
```

```
# % non siti predetti come non siti
percnono <- round((right_ns/length(No_Sites)), digits = 2)

percnosit + percnono
```

```
Prob01 Prob02 Prob03 Prob04 Prob05 Prob06 Prob07 Prob08
     1      1      1      1      1      1      1      1
Prob09
     1
```

```
# siti predetti erroneamente
wrong_s <- length(Sites) - totsites
wrong_s + totsites
```

```
Prob01 Prob02 Prob03 Prob04 Prob05 Prob06 Prob07 Prob08
    35     35     35     35     35     35     35     35
Prob09
    35
```

```
# % siti predetti erroneamente
percsno <- round((wrong_s/length(Sites)), digits = 2)

percsit + percsno
```

```
Prob01 Prob02 Prob03 Prob04 Prob05 Prob06 Prob07 Prob08
     1      1      1      1      1      1      1      1
Prob09
     1
```

```
# Kvamme's gain No-Siti dentro l'area predetta

NSS_KG <- (1 - (percarea/percnosit))

KG_ns_sites <- rbind(percarea, percnosit, NSS_KG)
colnames(KG_ns_sites) <- c(">0.1", ">0.2", ">0.3", ">0.4", ">0.5",
    ">0.6", ">0.7", ">0.8", ">0.9")
row.names(KG_ns_sites) <- c("% Area", "% No Sites", "Kvamme's Gain")
```

Tabella 13: Kvamme's Gain Non-Siti errati

	>0.1	>0.2	>0.3	>0.4	>0.5	>0.6	>0.7	>0.8	>0.9
% Area	0.42	0.30	0.24	0.19	0.15	0.11	0.08	0.05	0.02
% No Sites	0.36	0.28	0.17	0.10	0.10	0.08	0.04	0.04	0.01
Kvamme's Gain	-0.17	-0.07	-0.41	-0.90	-0.50	-0.38	-1.00	-0.25	-1.00

```
# Kvamme's gain No-Siti fuori dall'area predetta

NSNS_KG <- (1 - (percnoarea/percnono))

KG_ns_ns <- rbind(percnoarea, percnono, NSNS_KG)
colnames(KG_ns_ns) <- c(">0.1", ">0.2", ">0.3", ">0.4", ">0.5",
    ">0.6", ">0.7", ">0.8", ">0.9")
row.names(KG_ns_ns) <- c("% Area Unpred", "% No Sites", "Kvamme's Gain")
```

Tabella 14: Kvamme's Gain Non-Siti corretti

	>0.1	>0.2	>0.3	>0.4	>0.5	>0.6	>0.7	>0.8	>0.9
% Area Unpred	0.58	0.70	0.76	0.81	0.85	0.89	0.92	0.95	0.98
% No Sites	0.64	0.72	0.83	0.90	0.90	0.92	0.96	0.96	0.99
Kvamme's Gain	0.09	0.03	0.08	0.10	0.06	0.03	0.04	0.01	0.01

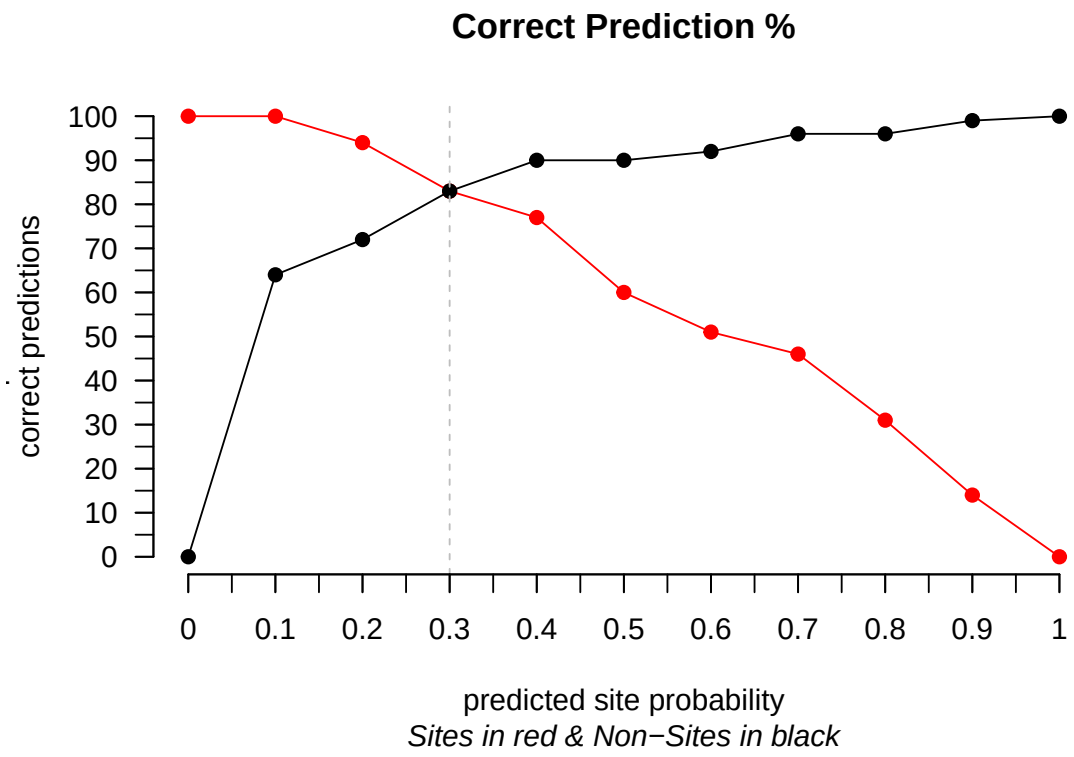
1.5.4 Soglia di corretta predizione

```
# sites
percsit["Prob0"] <- 1
percsit["Prob10"] <- 0
percfull <- as.data.frame(percsit)
percprob <- rbind(percfull[10, ], percfull[1, ], percfull[2,
    ], percfull[3, ], percfull[4, ], percfull[5, ], percfull[6,
    ], percfull[7, ], percfull[8, ], percfull[9, ], percfull[11,
    ])

# no sites
percnono["Prob0"] <- 1
percnono["Prob10"] <- 0
percfullns <- as.data.frame(percnono)
percprobns <- rbind(percfullns[11, ], percfullns[1, ], percfullns[2,
    ], percfullns[3, ], percfullns[4, ], percfullns[5, ], percfullns[6,
    ], percfullns[7, ], percfullns[8, ], percfullns[9, ], percfullns[10,
    ])

# table
cutoff <- c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)

tabperc <- cbind(cutoff, (percprob * 100), (percprobns * 100))
colnames(tabperc) <- c("cutoff", "%sites", "%random")
```

Tabella 15: Cumulative prediction

cutoff	%sites	%random
0.0	100	0
0.1	100	64
0.2	94	72
0.3	83	83
0.4	77	90
0.5	60	90
0.6	51	92
0.7	46	96
0.8	31	96

cutoff	%sites	%random
0.9	14	99
1.0	0	100




```
jpeg("Correct_Prediction.jpg", width = 1500, height = 2000, units = "px",
    res = 300)
par(oma = c(1, 0, 1, 0))
plot(tabperc[, 2] ~ tabperc[, 1], xlab = "predicted site probability",
    ylab = "percent
correct predictions", main = "Correct Prediction %",
    sub = "Sites in red & Non-Sites in black", col = "red", pch = 19,
    axes = F, font.sub = 3)
lines(tabperc[, 2] ~ tabperc[, 1], col = "red")
par(new = TRUE)
plot(tabperc[, 3] ~ tabperc[, 1], xlab = "", ylab = "", pch = 19,
    axes = F)
lines(tabperc[, 3] ~ tabperc[, 1])
abline(v = 0.3, lty = 2, col = "grey")
Axis(side = 1, at = seq(0, 1, by = 0.1), labels = seq(0, 1, by = 0.1),
    cex = 0.2)
Axis(side = 1, at = seq(0, 1, by = 0.05), labels = F)
Axis(side = 2, at = seq(0, 100, by = 10), labels = seq(0, 100,
    10), las = 1)
Axis(side = 2, at = seq(0, 100, by = 5), labels = F)
dev.off()
```

1.5.5 Area sotto la curva ROC

```
library(pROC)

sbcroc <- roc(BIC_glm$y, BIC_glm$fitted.values)
```

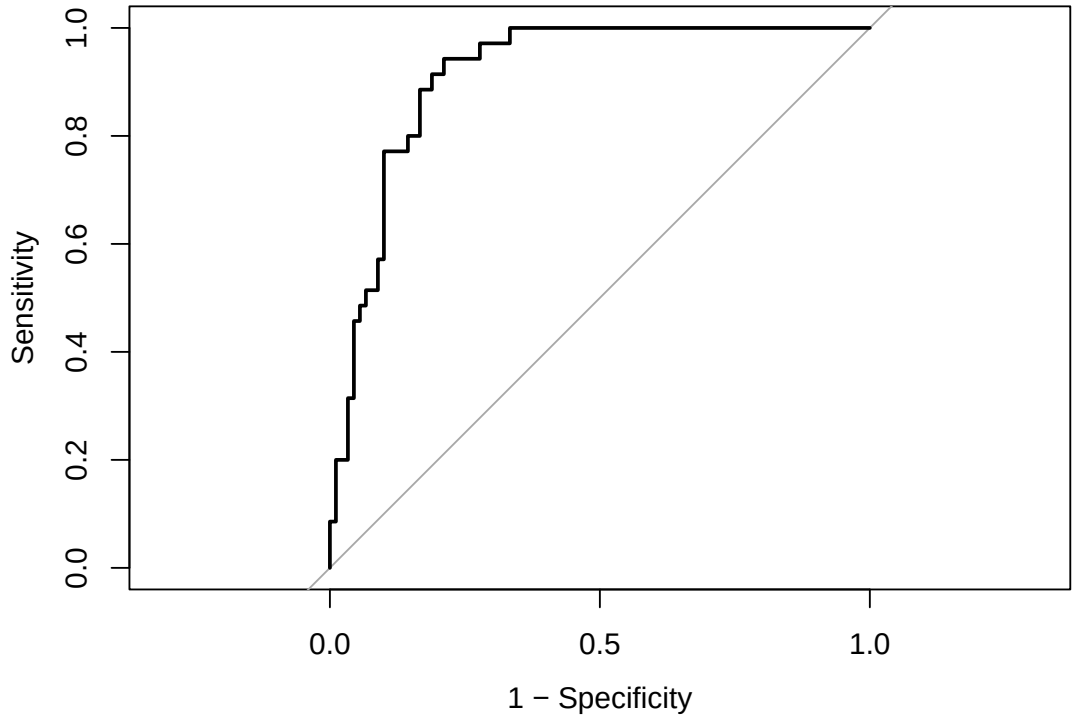
```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

```
sbcroc
```

```
Call:
roc.default(response = BIC_glm$y, predictor = BIC_glm$fitted.values)

Data: BIC_glm$fitted.values in 90 controls (BIC_glm$y 0) < 35 cases (BIC_glm$y 1).
Area under the curve: 0.9127
```

```
plot.roc(sbcroc, reuse.auc = T, legacy.axes = T)
```




```
capture.output(sbcroc)
```

```
[1] ""
[2] "Call:"
[3] "roc.default(response = BIC_glm$y, predictor = BIC_glm$fitted.values)"
[4] ""
[5] "Data: BIC_glm$fitted.values in 90 controls (BIC_glm$y 0) < 35 cases (BIC_glm$y 1)."
[6] "Area under the curve: 0.9127"
```

Discriminatory Ability (HOSMER et. al. 2013)
0.5 = No better than chance
0.5-0.7 = Poor
0.7-0.8 = Acceptable
0.8-0.9 = Excellent
0.9-1.0 = Outstanding

1.5.6 Resa grafica dei risultati della validazione del modello

```
jpeg("Model_Assessement.jpg", width = 4000, height = 1000, units = "px",
    res = 300)
par(oma = c(0, 0, 3, 0))
layout(matrix(c(1, 2, 3, 4), nrow = 1, byrow = T))
# sites
hist(sitespred$extract.predsurf..Sites., las = 1, col = "olivedrab3",
    ylim = c(0, 6), xlim = c(0, 1), main = "Sites", xlab = "probability",
    axes = F, mar = c(0, 0, 1, 0), freq = T)
abline(v = 0.3, lty = 2, col = "red")
Axis(side = 1, at = seq(0, 1, by = 0.1), labels = seq(0, 1, 0.1))
Axis(side = 2, at = seq(0, 6, by = 1), labels = seq(0, 6, 1),
    las = 2)
# no sites
hist(nositespred$extract.predsurf..No_Sites., las = 1, col = "lightpink3",
    ylim = c(0, 60), xlim = c(0, 1), main = "No Sites", xlab = "probability",
    axes = F, mar = c(0, 0, 1, 0), freq = T)
abline(v = 0.3, lty = 2, col = "red")
Axis(side = 1, at = seq(0, 1, by = 0.1), labels = F)
Axis(side = 1, at = seq(0, 1, by = 0.5), labels = seq(0, 1, 0.5))
Axis(side = 1, at = 0.3, labels = 0.3)
Axis(side = 2, at = seq(0, 60, by = 10), labels = seq(0, 60,
    10), las = 2)
# correct prediction
plot(tabperc[, 2] ~ tabperc[, 1], xlab = "predicted site probability",
    ylab = "percent
correct predictions", main = "Correct Prediction % \nSites in red & Non-Sites in black",
    col = "red", pch = 19, axes = F, font.sub = 3)
lines(tabperc[, 2] ~ tabperc[, 1], col = "red")
par(new = TRUE)
plot(tabperc[, 3] ~ tabperc[, 1], xlab = "", ylab = "", pch = 19,
    axes = F)
lines(tabperc[, 3] ~ tabperc[, 1])
Axis(side = 1, at = seq(0, 1, by = 0.1), labels = seq(0, 1, by = 0.1))
Axis(side = 2, at = seq(0, 100, by = 10), labels = seq(0, 100,
    10), las = 1)
Axis(side = 2, at = seq(0, 100, by = 5), labels = F)
abline(v = 0.3, lty = 2, col = "black")
# AUC
plot.roc(sbcroc, reuse.auc = T, legacy.axes = T, main = "Area under the ROC curve")
# titolo
mtext("Model Assessement", outer = TRUE, cex = 1.5)
dev.off()
```

1.6 Validazione esterna del modello

Applicazione del modello predittivo ad un'area esterna a quella utilizzata per il suo calcolo, ma contigua e con caratteristiche simili dal punto di vista fisico e antropico.

1.6.1 Calcolo della superficie predittiva

Le operazioni preliminari di manipolazione dei dati spaziali avvengono tramite **GRASS GIS** versione 7.6.

```
Database directory: (D:/Archeologia/Carona/GRASS)
Location: SBC
Projection: WGS84/UTM32N - EPSG: 32632
Mapset: Pred_Mod_Fondra
```

Importazione della cartografia di base

```
r.in.gdal input=D:/Cartografia/EUROPA/ITALIA/LOMBARDIA/prov
BERGAMO/DTM/DTM5_BG.img output=DTM_BG r.in.gdal
input=D:/Cartografia/EUROPA/ITALIA/LOMBARDIA/prov
SONDRIO/DTM/DTM5_SO.img output=DTM_SO v.in.ogr
input=D:/Archeologia/Carona/GIS/Area_Controllo.shp
g.region
[vector=Area_Controllo@Pred_Mod_Fondra]
res=25
r.mask
[vector=Area_Controllo@Pred_Mod_Fondra]
r.patch
[input=DTM_BG@Pred_Mod_Fondra],[DTM_SO@Pred_Mod_Fondra]
output=DTM_Controllo
r.mask -r
[vector=Area_Controllo@Pred_Mod_Fondra]
r.slope.aspect
[elevation=DTM_Controllo@Pred_Mod_Fondra]
slope=Slope_Controllo\
v.in.ogr
input=D:/Archeologia/Carona/GIS/Cartografia/Laghi_Controllo.shp
v.to.rast
[input=Laghi_Controllo@Pred_Mod_Fondra]
output=Laghi use=val
```

Applicazione dei coefficienti della regressione

```
r.mask -i
[raster=Laghi@Pred_Mod_Fondra]\
r.mapcalc expression=Regression_Control = 10.047158 + (
[DTM_Controllo@Pred_Mod_Fondra]
\* -0.003264 ) + ([Slope_Controllo@Pred_Mod_Fondra]\* -0.202717 )
r.mapcalc expression=
Prob_Surf_Control=(exp([Regression_Control@Pred_Mod_Fondra]))/
(1+(exp([Regression_Control@Pred_Mod_Fondra])))
r.mask -i -r
[raster=Laghi@Pred_Mod_Fondra]
r.out.gdal
[input=Prob_Surf_Control@Pred_Mod_Fondra]
output=D:/Archeologia/Carona/GIS/Pred_Mod/Modello_Controllo_Fondra.tif
format=GTiff type=Float64
```

Categorizzazione della superficie

```
r.mapcalc expression=Control01 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.1
r.mapcalc expression=Control02 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.2
r.mapcalc expression=Control03 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.3
r.mapcalc expression=Control04 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.4
r.mapcalc expression=Control05 =
[Prob_Surf_Control@Pred_Mod_Fondra]
```

```
\>= 0.5
r.mapcalc expression=Control06 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.6
r.mapcalc expression=Control07 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.7
r.mapcalc expression=Control08 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.8
r.mapcalc expression=Control09 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0.9
r.mapcalc expression=Control00 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 0
r.mapcalc expression=Control10 =
[Prob_Surf_Control@Pred_Mod_Fondra]
\>= 1
```

Lettura valori della superficie per i siti di controllo

```
v.in.ogr input=D:/Archeologia/Carona/GIS/Siti/Siti_Controllo.shp
v.what.rast
[map=Siti_Controllo@Pred_Mod_Fondra]
[raster=Prob_Surf_Control@Pred_Mod_Fondra]
column=pred_mod
```

1.6.2 Importazione dei dati in RStudio

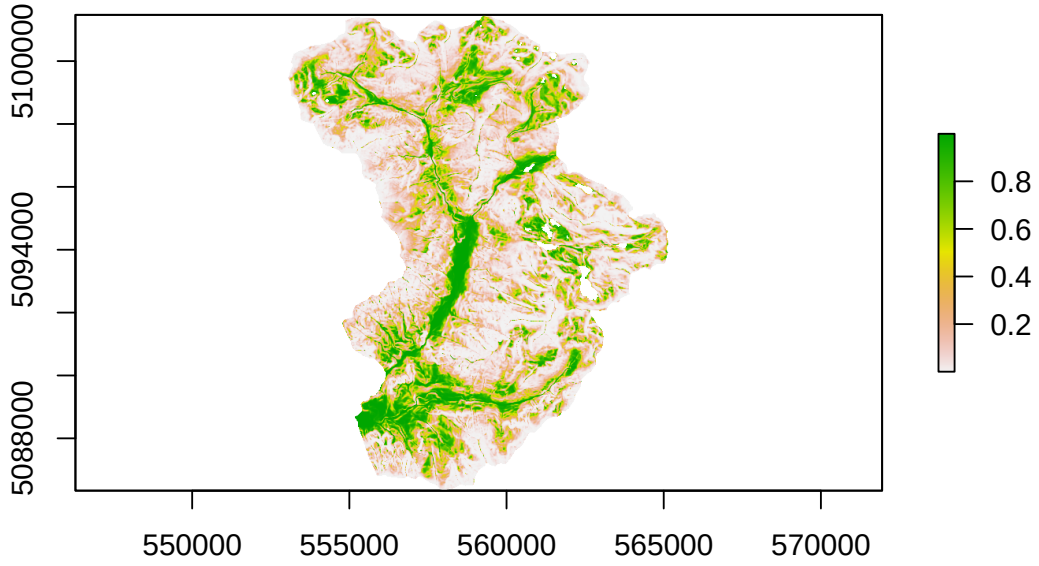
```
initGRASS(gisBase = "C:/Program Files/GRASS GIS 7.6", gisDbase = "D:/Archeologia/Carona/GRASS",
    location = "SBC", mapset = "Pred_Mod_Fondra", override = TRUE)
```

```
gisdbase    D:/Archeologia/Carona/GRASS
location    SBC
mapset      Pred_Mod_Fondra
rows        605
columns     486
north       5101472
south       5086336
west        553036.4
east        565197.8
nsres       25.01713
ewres       25.02352
```

```
predsurf_cont <- raster(readRAST("Prob_Surf_Control", plugin = F))
class(predsurf_cont)
```

```
[1] "RasterLayer"
attr(,"package")
[1] "raster"
```

```
plot(predsurf_cont)
```




```
Alpeggi_cont <- (readVECT("Siti_Controllo"))
```

```
OGR data source with driver: GPKG
Source: "D:\Archeologia\Carona\GRASS\SBC\Pred_Mod_Fondra\.tmp\unknown\467.0.gpkg", layer: "Siti_Contr
with 74 features
It has 10 fields
```

```
summary(is.na(Alpeggi_cont$pred_mod))
```

```
   Mode   FALSE
logical      74
```

```
class(Alpeggi_cont)
```

```
[1] "SpatialPointsDataFrame"
attr(,"package")
[1] "sp"
```

```
Cat_cont <- (stack(readRAST(c("Control01", "Control02", "Control03",
    "Control04", "Control05", "Control06", "Control07", "Control08",
    "Control09"), cat = c(T, T, T, T, T, T, T, T, T), plugin = F)) -
    1)
```

```
Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
```

```
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended

Warning in .read_rast_non_plugin(vname = vname, NODATA =
NODATA, driverFileExt = driverFileExt, : non-unique category
labels; category number appended
```

Resa grafica.

```
jpeg("Model_Validation_Fondra.jpg", width = 2000, height = 1500,
    units = "px", res = 300)
par(oma = c(0, 0, 2, 0))
hist(Alpeggi_cont$pred_mod, col = "purple", main = "Prediction Alpeggi SBB",
    xlim = c(0, 1), xlab = "", ylim = c(0, 20), ylab = "", las = 1,
    breaks = seq(0, 1, 0.1), axes = F)
Axis(side = 1, at = seq(0, 1, by = 0.1), labels = seq(0, 1, 0.1))
Axis(side = 2, at = seq(0, 20, by = 5), labels = seq(0, 20, 5),
    las = 1)
abline(v = 0.3, lty = 2, col = "red")
dev.off()
```

1.6.3 *Kvamme's Gain*

```
# lettura valori in ogni categoria predittiva

alppredcat <- data.frame(extract(Cat_cont, Alpeggi_cont))
str(alppredcat)
```

```
'data.frame':   74 obs. of  9 variables:
 $ Control01: num  1 1 1 1 1 1 1 1 1 1 ...
 $ Control02: num  1 1 1 1 1 1 1 1 1 1 ...
 $ Control03: num  1 1 1 1 1 1 1 1 1 1 ...
 $ Control04: num  1 1 1 0 0 1 1 1 1 1 ...
 $ Control05: num  1 1 1 0 0 1 0 1 1 1 ...
 $ Control06: num  1 1 1 0 0 0 0 1 1 1 ...
 $ Control07: num  1 1 1 0 0 0 0 1 1 1 ...
 $ Control08: num  1 1 1 0 0 0 0 1 1 1 ...
 $ Control09: num  0 0 0 0 0 0 0 0 0 0 ...
```

```
totalp <- colSums(alppredcat)
# percentuale alpeggi in area predittiva
percalp <- round((totalp/length(Alpeggi_cont)), digits = 2)
# calcolo aree
areapred <- rbind((freq(Cat_cont, useNA = "no", merge = T)),
    (colSums(freq(Cat_cont, useNA = "no", merge = T))))
percareacont <- round((areapred[2, ]/areapred[3, ]), digits = 2)
percareacont[, 1] <- NULL
# calcolo del Kvamme's Gain
Alpi_KG <- (1 - (percareacont/percalp))
KG_alp <- rbind(percareacont, percalp, Alpi_KG)
colnames(KG_alp) <- c(">0.1", ">0.2", ">0.3", ">0.4", ">0.5",
    ">0.6", ">0.7", ">0.8", ">0.9")
row.names(KG_alp) <- c("% Area", "% Siti", "Kvamme's Gain")
```

Tabella 16: Kvamme's Gain siti val Fondra

	>0.1	>0.2	>0.3	>0.4	>0.5	>0.6	>0.7	>0.8	>0.9
% Area	0.52	0.40	0.32	0.27	0.22	0.18	0.15	0.11	0.07
% Siti	1.00	0.92	0.82	0.73	0.61	0.54	0.47	0.41	0.15
Kvamme's Gain	0.48	0.57	0.61	0.63	0.64	0.67	0.68	0.73	0.53

1.6.4 Kolmogorov-Smirnov test

Effettuato in bootstrapping con 10.000 ripetizioni.

```
predalp <- data.frame(Alpeggi_cont$pred_mod)
table(is.na(predalp))
```

```
FALSE
   74
```

```
class(predalp)
```

```
[1] "data.frame"
```

```
library(Matching)

# x=valore predittivo punti controllo y=superficie
# predittiva z=maschera nsim=ripetizione

library(Matching)

ks.survey <- function(x, y, z, nsim) {
    randpts <- data.frame(sampleRandom((mask(y, z, inverse = T)),
        (10 * (nrow(x))), replace = F, na.rm = T))
    return(summary(ks.boot(x, randpts, nboots = nsim)))
}

ks_alpi <- capture.output(ks.survey(predalp, predsurf_cont, Alpeggi_cont,
    10000))
ks_alpi
```

```
[1] ""
[2] "Bootstrap p-value:     < 2.22e-16 "
[3] "Naive p-value:         8.8818e-16 "
[4] "Full Sample Statistic: 0.51216 "
[5] ""
```

2 Dati cronologici

Per l'analisi dei dati cronologici ci si è avvalsi di diversi componenti aggiuntivi per RStudio, utilizzati sia per la resa grafica che per l'analisi dei dati:

```
library(rcarbon)
library(ggplot2)
library(Bchron)
library(oxcAAR)
```

2.1 Datazioni radiocarboniche.

```
setwd("D:/Archeologia/Carona/Datazioni")
tab_14c <- read.csv("D:/Archeologia/Carona/Datazioni/14c_uncalibrated.csv")
class(tab_14c)
```

```
[1] "data.frame"
```

```
codone <- paste(tab_14c$Cod_Lab, tab_14c$Cod_MABG, tab_14c$Cod_SBC,
    sep = " / ")
caldat <- calibrate(x = tab_14c$Data, errors = tab_14c$Error,
    ids = codone, dateDetails = tab_14c$Descrizione, calCurves = "intcal20",
    normalised = T)
```

```
[1] "Calibrating radiocarbon ages..."
  |                                                                      |
[1] "Done."
```

```
cod_sbc <- paste(tab_14c$Cod_Lab, tab_14c$Cod_SBC, sep = " / ")
caldat_SBC <- calibrate(x = tab_14c$Data, errors = tab_14c$Error,
    ids = cod_sbc, dateDetails = tab_14c$Descrizione, calCurves = "intcal20",
    normalised = T)
```

```
[1] "Calibrating radiocarbon ages..."
  |                                                                      |
[1] "Done."
```

```
BPtoBCAD(medCal(caldat_SBC))
```

```
 [1]  654 1767  729 1836  702 1185  952  737 1352 1349 1830
[12] 1344 1019  716 1347 1186 1432 1803 1118
```

Resa grafica di tutte le date calibrate.

```
jpeg("Date_Carona_Calibrated.jpg", width = 5000, height = 6000,
    units = "px", res = 300)
par(oma = c(0, 0, 2, 0))
layout(matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
    15, 16, 17, 18, 19, 20), nrow = 5, byrow = TRUE))
# 1
plot(caldat, 1, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[1])
# 2
plot(caldat, 2, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[2])
# 3
plot(caldat, 3, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[3])
# 4
plot(caldat, 4, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[4])
```

```
# 5
plot(caldat, 5, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[5])
# 6
plot(caldat, 6, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[6])
# 7
plot(caldat, 7, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[7])
# 8
plot(caldat, 8, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[8])
# 9
plot(caldat, 9, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[9])
# 10
plot(caldat, 10, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[10])
# 11
plot(caldat, 11, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[11])
# 12
plot(caldat, 12, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[12])
# 13
plot(caldat, 13, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[13])
# 14
plot(caldat, 14, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[14])
# 15
plot(caldat, 15, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[15])
# 16
plot(caldat, 16, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[16])
# 17
plot(caldat, 17, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[17])
# 18
plot(caldat, 18, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[1])
# 19
plot(caldat, 19, HPD = TRUE, credMass = 0.95, calendar = "BCAD",
    main = caldat$metadata$DateID[18])
dev.off()
```

Calibrazione non normalizzata, per la creazione di una curva SPD alternativa (Crema & Bevan 2021; Weninger et al. 2015).

```
caldat.un <- calibrate(x = tab_14c$Data, errors = tab_14c$Error,
    ids = tab_14c$Cod_Lab, calCurves = "intcal20", normalised = F)
```

```
[1] "Calibrating radiocarbon ages..."
  |                                                                      |
[1] "Done."
```

Resa grafica delle curve calibrate in un unico plot.

```
jpeg("Multi14C_SBC.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
par(oma = c(1, 1, 3, 1), mar = c(3, 1, 1, 1), mgp = c(2, 0.5,
    0))
multiplot(caldat_SBC, decreasing = TRUE, HPD = TRUE, rescale = TRUE,
    calendar = "BCAD", xlim = c(1500, 0), cex.axis = 0.8)
Axis(side = 1, at = seq(400, 2000, by = 100), cex.axis = 0.8)
mtext(side = 3, line = 1, cex = 1, "Datazioni Radiocarboniche")
mtext(side = 3, line = 2, cex = 2, "Sorgenti del Brembo di Carona")
abline(v = 476, lty = 2, col = "black", lwd = 0.7)
abline(v = 1000, lty = 2, col = "black", lwd = 0.7)
abline(v = 1453, lty = 2, col = "black", lwd = 0.7)
abline(v = 1815, lty = 2, col = "black", lwd = 0.7)
abline(v = 1950, lty = 1, col = "black", lwd = 2)
# etichette periodi
shadowtext(738, -0.5, "Alto Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1226, -0.5, "Basso Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1634, -0.5, "Et\340 Moderna", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
dev.off()
```

2.1.1 Summed Probability Density (SPD)

Generazione delle SPD.

```
caldat.spd <- spd(caldat, timeRange = c(2600, 0))
```

```
[1] "Extracting and aggregating..."
[1] "Done."
```

```
caldat.spdAD <- spd(caldat, timeRange = c(1500, 0))
```

```
[1] "Extracting and aggregating..."
[1] "Done."
```

Generazione delle SPD non normalizzata.

```
caldat.un.spd <- spd(caldat.un, timeRange = c(2600, 0))
```

```
[1] "Extracting and aggregating..."
[1] "Done."
```

```
caldat.un.spdAD <- spd(caldat.un, timeRange = c(1500, 0))
```

```
[1] "Extracting and aggregating..."
[1] "Done."
```

Resa grafica.

```
# Normalizzata
jpeg("SPD_SBC.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
par(oma = c(2, 1, 5, 1), mar = c(2.5, 3, 1, 0.5), mgp = c(1.6,
    0.5, 0), cex = 0.8)
plot(caldat.spdAD, calendar = "BCAD", ylim = c(0, 0.05))
plot(caldat.spdAD, calendar = "BCAD", type = "simple", runm = 50,
    col = "orange3", lwd = 2, lty = 1, add = TRUE)
Axis(side = 1, at = seq(500, 1900, by = 100))
mtext("Datazioni Radiocaboniche S.B.C.", side = 3, line = 2.5,
```

```
    outer = TRUE, cex = 2)
mtext("Summed Probability Distributions (SPD)", side = 3, line = 0.9,
    outer = TRUE)
mtext("Normalised calibrations", side = 3, line = 0.1, outer = TRUE,
    cex = 0.8)
mtext("Curve Rolling Average: 50 years", side = 1, line = 0.5,
    outer = TRUE, cex = 0.6, adj = 0)
dev.off()

# Non Normalizzata
jpeg("SPD_un_SBC.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
par(oma = c(2, 1, 5, 1), mar = c(2.5, 3, 1, 0.5), mgp = c(1.6,
    0.5, 0), cex = 0.8)
plot(caldat.un.spdAD, calendar = "BCAD", ylim = c(0, 0.05))
plot(caldat.un.spdAD, calendar = "BCAD", type = "simple", runm = 50,
    col = "orange3", lwd = 2, lty = 1, add = TRUE)
Axis(side = 1, at = seq(500, 1900, by = 100))
mtext("Datazioni Radiocaboniche S.B.C.", side = 3, line = 2.5,
    outer = TRUE, cex = 2)
mtext("Summed Probability Distributions (SPD)", side = 3, line = 0.9,
    outer = TRUE)
mtext("Unnormalised calibrations", side = 3, line = 0.1, outer = TRUE,
    cex = 0.8)
mtext("Curve Rolling Average: 50 years", side = 1, line = 0.5,
    outer = TRUE, cex = 0.6, adj = 0)
dev.off()
```

2.2 Altre evidenze cronologiche

Raccolta dei dati relativi ad evidenze con valore cronologico diverse dalle date radiometriche.

2.2.1 Date incise su roccia

```
date_incise <- read.csv("D:/Archeologia/Carona/Datazioni/date_incise.csv",
    header = T)
class(date_incise)
```

```
[1] "data.frame"
```

```
class(date_incise$Anno)
```

```
[1] "character"
```

```
date_incise$Anno <- as.numeric(as.character(date_incise$Anno))
```

```
Warning: NAs introduced by coercion
```

```
sum(is.na(date_incise$Anno))
```

```
[1] 7
```

```
nrow(date_incise)
```

```
[1] 221
```

Calcolo del range cronologico in cui ricadono le date incise

```
# data pi\371 antica
min(date_incise$Anno, na.rm = T)
```

```
[1] 1476
```

```
# data pi\371 recente
max(date_incise$Anno, na.rm = T)
```

```
[1] 2004
```

Resa grafica dei dati raccolti.

```
jpeg("Date_Incise.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(date_incise, aes(x = Secolo)) + geom_bar(stat = "count",
    position = "dodge", fill = "maroon3", color = "magenta4") +
    labs(title = "Sorgenti del Brembo di Carona (BG): Date incise",
        x = "century", y = "count", subtitle = paste("Totale:",
            nrow(date_incise)), caption = "Fonti:
       Civico Museo Archeologico di Bergamo
       & Survey Croce E. 2019-2021") +
    expand_limits(y = c(0, 100)) + theme(axis.text = element_text(size = 10,
    colour = "black")) + theme_bw() + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        2), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

2.2.2 Reperti archeologici

```
# fibula V sec. a.C.
fibu <- seq(from = -499, to = -400, by = 1)
length(fibu)
```

```
[1] 100
```

```
# Aes Rude (x2) V-III sec. a.C.
aes1 <- seq(from = -499, to = -200, by = 1)
length(aes1)
```

```
[1] 300
```

```
aes2 <- seq(from = -499, to = -200, by = 1)
length(aes2)
```

```
[1] 300
```

```
# Brocca da Scarpa 45-25 a.C.
broc <- seq(from = -45, to = -25, by = 1)
length(broc)
```

```
[1] 21
```

```
# Bottone XVIII sec. d.C.
bott <- seq(from = 1700, to = 1799, by = 1)
length(bott)
```

```
[1] 100
```

```
# Gancio da fibbia XVIII sec. d.C.
fibb <- seq(from = 1700, to = 1799, by = 1)
length(fibb)
```

```
[1] 100
```

```
# chiodi vari
chio <- seq(from = 1400, to = 1799, by = 1)
length(chio)
```

```
[1] 400
```

```
# Cucchiaio inizi XVIII sec. d.C.
cucc <- seq(from = 1700, to = 1725, by = 1)
length(cucc)
```

```
[1] 26
```

```
reperti <- c(fibu, aes1, aes2, broc, bott, fibb, chio, cucc)
```

2.2.3 Incisioni rupestri

```
# figure CMS1 V sec. a.C.
lupi <- seq(from = -499, to = -400, by = 1)
length(lupi)
```

```
[1] 100
```

```
# iscrizioni in Leponzio III-I sec. a.C.
lepo <- seq(from = -299, to = 0, by = 1)
length(lepo)
```

```
[1] 300
```

```r
# alfabetario latino II-I sec. a.C.
lati <- seq(from = -199, to = 0, by = 1)
length(lati)
```

```
[1] 200
```

```r
# armato CMS1 XIII sec. d.C.
cava <- seq(from = 1201, to = 1300, by = 1)
length(cava)
```

```
[1] 100
```

```r
# figura umana AGA1 XVI sec. d.C.
lan1 <- seq(from = 1501, to = 1600, by = 1)
length(lan1)
```

```
[1] 100
```

```r
# figura umana CMS10 XVI sec. d.C.
lan2 <- seq(from = 1501, to = 1600, by = 1)
length(lan2)
```

```
[1] 100
```

```r
# incisioni torbiere e aga
ltbaga <- (seq(from = 1501, to = 2000, by = 1))

incisioni <- c(lupi, lepo, lati, cava, lan1, lan2, ltbaga)
```

2.3 Resa grafica delle evidenze raccolte

2.3.1 Resa grafica di tutte le evidenze

Unione delle evidenze, con datazioni radiocarboniche rappresentate dalla curva SPD non normalizzata.

```
jpeg("Cronologia_Carona_SPD.jpg", width = 6000, height = 3500,
    units = "px", res = 300)
par(oma = c(3, 0.5, 0, 0))
# Date Radiocarboniche
plot(caldat.un.spd, calendar = "BCAD", las = 1, xaxt = "n", main = "",
    xlim = c(-600, 2020), ylim = c(0, 0.06))
plot(caldat.un.spd, calendar = "BCAD", runm = 50, add = TRUE,
    type = "simple", col = "blue", lwd = 3, lty = 1, xaxt = "n",
    xlim = c(-600, 2020), ylim = c(0, 0.06))
abline(v = 1950, lty = 2)
Axis(side = 1, at = seq(-600, 2050, by = 50), labels = F)
Axis(side = 1, at = seq(-600, -100, by = 100), labels = seq(-600,
    -100, 100))
Axis(side = 1, at = seq(100, 2000, by = 100), labels = seq(100,
    2000, 100))
Axis(side = 1, at = 1, labels = 1)
# Date incise
lines(density(date_incise$Anno, na.rm = T), lwd = 6, col = "orange")
polygon(density(date_incise$Anno, na.rm = T), col = "gold1",
    border = NA)
# epoche storiche
abline(v = -27, lty = 3, col = "gray61", lwd = 2)
abline(v = 476, lty = 3, col = "gray61", lwd = 2)
abline(v = 1000, lty = 3, col = "gray61", lwd = 2)
abline(v = 1453, lty = 3, col = "gray61", lwd = 2)
abline(v = 1815, lty = 3, col = "gray61", lwd = 2)
# palinologia PDS
abline(v = 350, lty = 2, col = "khaki3", lwd = 3)
abline(v = 650, lty = 2, col = "khaki3", lwd = 3)
abline(v = 900, lty = 2, col = "khaki3", lwd = 3)
abline(v = 1200, lty = 2, col = "khaki3", lwd = 3)
abline(v = 1520, lty = 2, col = "khaki3", lwd = 3)
# documenti storici
abline(v = 1144, lty = 2, col = "purple", lwd = 2.5)
abline(v = 1148, lty = 2, col = "purple", lwd = 2.5)
abline(v = 1180, lty = 2, col = "purple", lwd = 2.5)
abline(v = 1278, lty = 2, col = "purple", lwd = 2.5)
abline(v = 1402, lty = 2, col = "purple", lwd = 2.5)
# palinologia armentarga
segments(x0 = 850, y0 = 0.052, x1 = 1570, y1 = 0.052, col = "khaki3",
    lty = 1, lwd = 8)
segments(x0 = -600, y0 = 0.052, x1 = 850, y1 = 0.052, col = "khaki3",
    lty = 2, lwd = 7)
# etichette
shadowtext(1200, 0.053, "Intenso impatto umano (Armentarga)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(150, 0.053, "Pascolo naturale? (Armentarga)", col = "black",
    bg = "khaki", cex = 1, r = 0.25)
# etichette
shadowtext(500, 0.058, "Prime attivit\340 antropiche \n (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(775, 0.0575, "Pascolo \n Attivit\340 di fuoco \n (Piani di Sasso)",
```

```
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1050, 0.058, "Diminuzione attivit\340 \n antropica \n (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1360, 0.0575, "Ripresa sfruttamento \n boschi e pascolo \n (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1760, 0.0582, "Predominanza dei pascoli (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1170, 0.048, "Donazioni \nVallalta", col = "black",
    bg = "plum1", cex = 1, r = 0.2)
shadowtext(1402, 0.048, "Locazione \nDossena", col = "black",
    bg = "plum1", cex = 1, r = 0.2)
shadowtext(1278, 0.043, "Permesso \ntrasporto \nderrate", col = "black",
    bg = "plum1", cex = 1, r = 0.2)
# fibula
segments(x0 = -500, y0 = 0.025, x1 = -400, y1 = 0.025, col = "red",
    lty = 1, lwd = 8)
text(-450, 0.026, "Fibula")
# aes rude
segments(x0 = -500, y0 = 0.02, x1 = -200, y1 = 0.02, col = "red",
    lty = 1, lwd = 8)
text(-350, 0.021, "Aes Rude")
# brocca da scarpa
segments(x0 = -45, y0 = 0.015, x1 = -25, y1 = 0.015, col = "red",
    lty = 1, lwd = 8)
text(-35, 0.016, "Brocca da scarpa")
# fibbia/bottone
segments(x0 = 1701, y0 = 0.03, x1 = 1800, y1 = 0.03, col = "red",
    lty = 1, lwd = 8)
text(1750, 0.031, "Fibbia e Bottone")
# cucchiaio
segments(x0 = 1700, y0 = 0.035, x1 = 1725, y1 = 0.035, col = "red",
    lty = 1, lwd = 8)
text(1713, 0.036, "Cucchiaio")
# chiodi
segments(x0 = 1400, y0 = 0.025, x1 = 1800, y1 = 0.025, col = "red",
    lty = 1, lwd = 8)
text(1580, 0.026, "Chiodi da Scarpa")
# alfabetario
segments(x0 = -200, y0 = 0.035, x1 = 1, y1 = 0.035, col = "darkgreen",
    lty = 1, lwd = 8)
shadowtext(-100, 0.0362, "Alfabetario Latino", col = "black",
    bg = "white", r = 0.2)
# leponzio
segments(x0 = -300, y0 = 0.04, x1 = 1, y1 = 0.04, col = "darkgreen",
    lty = 1, lwd = 8)
text(-150, 0.041, "Iscrizioni in Leponzio")
# fitomorfi e folgori
segments(x0 = 1, y0 = 0.03, x1 = 200, y1 = 0.03, col = "darkgreen",
    lty = 1, lwd = 8)
text(101, 0.031, "Fitomorfi e Folgori")
# figure
segments(x0 = -500, y0 = 0.04, x1 = -400, y1 = 0.04, col = "darkgreen",
    lty = 1, lwd = 8)
text(-450, 0.041, "Figure CMS1")
# armato
segments(x0 = 1201, y0 = 0.035, x1 = 1300, y1 = 0.035, col = "darkgreen",
```

```
    lty = 1, lwd = 8)
shadowtext(1258, 0.036, "Armato CMS1", col = "black", bg = "white",
    r = 0.2)
# mercenari
segments(x0 = 1501, y0 = 0.035, x1 = 1560, y1 = 0.035, col = "darkgreen",
    lty = 1, lwd = 8)
text(1530, 0.036, "Mercenari")
# LTB1
segments(x0 = 1501, y0 = 0.04, x1 = 2000, y1 = 0.04, col = "darkgreen",
    lty = 1, lwd = 8)
text(1750, 0.041, "Roccia LTB1")
# legenda
legend("topleft", legend = c("Incisioni", "Oggetti", "SPD 14C*",
    "Date Incise", "Doc. Storica", "Palinologia"), col = c("darkgreen",
    "red", "blue", "gold1", "purple", "khaki3"), lty = 1, cex = 1.3,
    lwd = 8)
mtext("Sorgenti del Brembo di Carona - Dati Cronologici", side = 3,
    line = 1, cex = 2, ylbias = 1)
mtext("Years BC/AD", side = 1, line = 3, cex = 1.5)
abline(h = 0, col = "black", lwd = 2)
# sottotitoli
mtext("*SPD: Summed Probability Distributions of radiocarbon dates with unnormalised calibrations - C
    side = 1, line = -0.5, outer = TRUE, cex = 1.2)
mtext("14C data: Civico Museo Archeologico Bergamo & UNITN - Calibration curve: IntCal20 (REIMER et a
    side = 1, line = 0.5, outer = TRUE, cex = 1.2)
mtext("Other data from: (BASSI 2010; CASINI et al. 2010,2012,2019; CASINI & FOSSATI 2013,2014,2016; C
    side = 1, line = 1.5, outer = TRUE, cex = 1.2)
dev.off()
```

Unione delle evidenze, con datazioni radiocarboniche rappresentate dalle singole curve calibrate.

```
jpeg("Cronologia_Carona.jpg", width = 6000, height = 3500, units = "px",
    res = 300)
par(oma = c(3, 0, 0, 1))
# Date Radiocarboniche
multiplot(caldat_SBC, decreasing = TRUE, HPD = TRUE, rescale = TRUE,
    calendar = "BCAD", label = T, xlim = c(2500, 0), cex.axis = 1e-05)
Axis(side = 4, at = seq(0, 15, by = 1), labels = F)
Axis(side = 4, at = seq(0, 15, by = 5), labels = seq(0, 15, 5),
    las = 1)
Axis(side = 1, at = seq(-600, 2050, by = 50), labels = F)
Axis(side = 1, at = 0, labels = 1)
Axis(side = 1, at = seq(-600, -100, by = 100), labels = seq(-600,
    -100, by = 100))
Axis(side = 1, at = seq(100, 2000, by = 100), labels = seq(100,
    2000, by = 100))
segments(x0 = 1950, y0 = 25, x1 = 1950, y1 = 0, lty = 2)
# epoche storiche
abline(v = -27, lty = 3, col = "gray61", lwd = 3)
abline(v = 476, lty = 3, col = "gray61", lwd = 3)
abline(v = 1000, lty = 3, col = "gray61", lwd = 3)
abline(v = 1453, lty = 3, col = "gray61", lwd = 3)
abline(v = 1815, lty = 3, col = "gray61", lwd = 3)
# palinologia PDS
segments(x0 = 350, y0 = 25, x1 = 350, y1 = 0, lty = 2, col = "khaki3",
    lwd = 3)
segments(x0 = 650, y0 = 25, x1 = 650, y1 = 0, lty = 2, col = "khaki3",
```

```
    lwd = 3)
segments(x0 = 900, y0 = 25, x1 = 900, y1 = 0, lty = 2, col = "khaki3",
    lwd = 3)
segments(x0 = 1200, y0 = 25, x1 = 1200, y1 = 0, lty = 2, col = "khaki3",
    lwd = 3)
segments(x0 = 1520, y0 = 25, x1 = 1520, y1 = 0, lty = 2, col = "khaki3",
    lwd = 3)
# documenti storici
segments(x0 = 1144, y0 = 0, x1 = 1144, y1 = 20, lty = 2, col = "purple",
    lwd = 2.5)
segments(x0 = 1148, y0 = 0, x1 = 1148, y1 = 20, lty = 2, col = "purple",
    lwd = 2.5)
segments(x0 = 1180, y0 = 0, x1 = 1180, y1 = 20, lty = 2, col = "purple",
    lwd = 2.5)
segments(x0 = 1278, y0 = 0, x1 = 1278, y1 = 20, lty = 2, col = "purple",
    lwd = 2.5)
segments(x0 = 1402, y0 = 0, x1 = 1402, y1 = 20, lty = 2, col = "purple",
    lwd = 2.5)
# palinologia armentarga
segments(x0 = 850, y0 = 20, x1 = 1570, y1 = 20, col = "khaki3",
    lty = 1, lwd = 8)
segments(x0 = -600, y0 = 20, x1 = 850, y1 = 20, col = "khaki3",
    lty = 2, lwd = 7)
# etichette
shadowtext(1200, 20.5, "Intenso impatto umano (Armentarga)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(155, 20.5, "Pascolo naturale? (Armentarga)", col = "black",
    bg = "khaki", cex = 1, r = 0.25)
shadowtext(500, 23.2, "Inizio impatto antropico \n (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(775, 23, "Pascolo \n Attivit\340 di fuoco \n (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1050, 23, "Diminuzione attivit\340 \n antropica \n (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1360, 23, "Ripresa sfruttamento \n boschi e pascolo \n (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1750, 23.4, "Predominanza pascoli (Piani di Sasso)",
    col = "black", bg = "khaki", cex = 1, r = 0.25)
shadowtext(1170, 3.5, "Donazioni \nVallalta", col = "black",
    bg = "plum1", cex = 1, r = 0.2)
shadowtext(1402, 3.5, "Locazione \nDossena", col = "black", bg = "plum1",
    cex = 1, r = 0.2)
shadowtext(1278, 2, "Permesso \ntrasporto \nderrate", col = "black",
    bg = "plum1", cex = 1, r = 0.2)
# Date incise
hist(date_incise$Anno, na.rm = T, breaks = 121, lwd = 6, col = "orange",
    border = "orange", add = TRUE)
# fibula
segments(x0 = -500, y0 = 16, x1 = -400, y1 = 16, col = "red",
    lty = 1, lwd = 8)
text(-450, 16.5, "Fibula", cex = 1.3)
# aes rude
segments(x0 = -500, y0 = 14, x1 = -200, y1 = 14, col = "red",
    lty = 1, lwd = 8)
text(-350, 14.5, "Aes Rude", cex = 1.3)
# brocca da scarpa
```

```
segments(x0 = -45, y0 = 12, x1 = -25, y1 = 12, col = "red", lty = 1,
    lwd = 8)
text(-35, 12.5, "Brocca da scarpa", cex = 1.3)
# fibbia/bottone
segments(x0 = 1701, y0 = 12, x1 = 1800, y1 = 12, col = "red",
    lty = 1, lwd = 8)
text(1750, 12.5, "Fibbia e Bottone", cex = 1.3)
# cucchiaio
segments(x0 = 1700, y0 = 14, x1 = 1725, y1 = 14, col = "red",
    lty = 1, lwd = 8)
text(1713, 14.5, "Cucchiaio", cex = 1.3)
# chiodi
segments(x0 = 1400, y0 = 10, x1 = 1800, y1 = 10, col = "red",
    lty = 1, lwd = 8)
text(1580, 10.5, "Chiodi da Scarpa", cex = 1.3)
# alfabetario
segments(x0 = -200, y0 = 6, x1 = 1, y1 = 6, col = "darkgreen",
    lty = 1, lwd = 8)
shadowtext(-100, 6.5, "Alfabetario Latino", col = "black", cex = 1.3,
    bg = "white", r = 0.2)
# leponzio
segments(x0 = -300, y0 = 8, x1 = 1, y1 = 8, col = "darkgreen",
    lty = 1, lwd = 8)
text(-150, 8.5, "Iscrizioni in Leponzio", cex = 1.3)
# fitomorfi e folgori
segments(x0 = 1, y0 = 4, x1 = 200, y1 = 4, col = "darkgreen",
    lty = 1, lwd = 8)
text(101, 4.5, "Fitomorfi e Folgori", cex = 1.3)
# figure
segments(x0 = -500, y0 = 10, x1 = -400, y1 = 10, col = "darkgreen",
    lty = 1, lwd = 8)
text(-450, 10.5, "Figure CMS1", cex = 1.3)
# armato
segments(x0 = 1201, y0 = 6, x1 = 1300, y1 = 6, col = "darkgreen",
    lty = 1, lwd = 8)
shadowtext(1258, 6.5, "Armato CMS1", col = "black", cex = 1.3,
    bg = "white", r = 0.2)
# mercenari
segments(x0 = 1501, y0 = 6, x1 = 1560, y1 = 6, col = "darkgreen",
    lty = 1, lwd = 8)
text(1530, 6.5, "Mercenari", cex = 1.3)
# LTB1
segments(x0 = 1501, y0 = 8, x1 = 2000, y1 = 8, col = "darkgreen",
    lty = 1, lwd = 8)
text(1750, 8.5, "Roccia LTB1", cex = 1.3)
# etichette periodi
shadowtext(-293, -0.5, "Et\340 del Ferro", col = "black", bg = "white",
    cex = 1, r = 0.2)
shadowtext(259, -0.5, "Et\340 Romana Imperiale", col = "black",
    bg = "white", cex = 1, r = 0.2)
shadowtext(738, -0.5, "Alto Medioevo", col = "black", bg = "white",
    cex = 1, r = 0.2)
shadowtext(1226, -0.5, "Basso Medioevo", col = "black", bg = "white",
    cex = 1, r = 0.2)
shadowtext(1634, -0.5, "Et\340 Moderna", col = "black", bg = "white",
    cex = 1, r = 0.2)
```

```
shadowtext(1932, -0.5, "Et\340 Contemporanea", col = "black",
    bg = "white", cex = 0.9, r = 0.19)
# legenda

legend("topleft", legend = c("Oggetti", "Incisioni", "Date Incise",
    "Doc. Storici", "Palinologia"), col = c("red", "darkgreen",
    "orange", "purple", "khaki3"), lty = 1, cex = 1.3, lwd = 8)
mtext("Sorgenti del Brembo di Carona - Dati Cronologici", side = 3,
    line = 1, cex = 2, ylbias = 1)
abline(h = 0, col = "black", lwd = 2)
# sottotitoli
mtext("Date 14C: Civico Museo Archeologico Bergamo & UNITN - Curva di calibrazione: IntCal20 (REIMER
    side = 1, line = 0.3, outer = TRUE, cex = 1.2)
mtext("Other data from: (BASSI 2010; CASINI et al. 2010,2012,2019; CASINI & FOSSATI 2013,2014,2016; C
    side = 1, line = 1.5, outer = TRUE, cex = 1.2)
dev.off()
```

2.3.2 Resa grafica delle evidenze divise per aree di interesse

```
# Armentarga

armentarga <- calibrate(x = c(1376, 205, 1268, 110), errors = c(26,
    29, 30, 35), calCurves = "intcal20", ids = c("PRS003", "ARM010",
    "ARM016", "ARM014"))

jpeg("Multi14C_Armentarga.jpg", width = 2500, height = 2000,
    units = "px", res = 300)
par(oma = c(0, 0, 1, 0), font.main = 1)
multiplot(armentarga, decreasing = TRUE, HPD = TRUE, rescale = TRUE,
    calendar = "BCAD", xlim = c(1500, 0), cex.axis = 0.8)
Axis(side = 1, at = seq(400, 1900, by = 100), cex.axis = 0.8)
mtext(side = 3, line = 2, cex = 1, "Datazioni Radiocarboniche")
mtext(side = 3, line = 3, cex = 2, "Armentarga")
abline(v = 476, lty = 2, col = "black", lwd = 0.7)
abline(v = 1000, lty = 2, col = "black", lwd = 0.7)
abline(v = 1453, lty = 2, col = "black", lwd = 0.7)
abline(v = 1815, lty = 2, col = "black", lwd = 0.7)
abline(v = 1950, lty = 1, col = "black", lwd = 2)
# etichette periodi
shadowtext(738, -0.1, "Alto Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1226, -0.1, "Basso Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1634, -0.1, "Et\340 Moderna", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
dev.off()

# Piani di Sasso

pianids <- calibrate(x = c(863, 1274, 1329, 1017, 1309, 585,
    601, 76, 660, 918), errors = c(45, 45, 45, 26, 29, 45, 45,
    23, 26, 40), calCurves = "intcal20", ids = c("S01_us33",
    "S01_us26", "S03_us34", "S03_us127", "S03_us114", "S04_us44",
    "S04_us87a", "S04_us74", "S04_us87b", "S03_us125"))

jpeg("Multi14C_Piani_Sasso.jpg", width = 2500, height = 2000,
```

```
    units = "px", res = 300)
par(oma = c(0, 0, 1, 0), font.main = 1)
multiplot(pianids, decreasing = TRUE, HPD = TRUE, rescale = TRUE,
    calendar = "BCAD", xlim = c(1500, 0), cex.axis = 0.8)
Axis(side = 1, at = seq(400, 1900, by = 100), cex.axis = 0.8)
mtext(side = 3, line = 2, cex = 1, "Datazioni Radiocarboniche")
mtext(side = 3, line = 3, cex = 2, "Piani di Sasso")
abline(v = 476, lty = 2, col = "black", lwd = 0.7)
abline(v = 1453, lty = 2, col = "black", lwd = 0.7)
abline(v = 1000, lty = 2, col = "black", lwd = 0.7)
abline(v = 1950, lty = 1, col = "black", lwd = 2)
abline(v = 1815, lty = 2, col = "black", lwd = 0.7)
# etichette periodi
shadowtext(738, -0.2, "Alto Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1226, -0.2, "Basso Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1634, -0.2, "Et\340 Moderna", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
dev.off()

# Carbonaie

carbonaie <- calibrate(x = c(1094, 612, 876), errors = c(35,
    29, 21), calCurves = "intcal20", ids = c("PDS002", "BDC004",
    "ARM001"))

jpeg("Multi14C_Carbonaie.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
par(oma = c(0, 0, 1, 0), font.main = 1)
multiplot(carbonaie, decreasing = TRUE, HPD = TRUE, rescale = TRUE,
    calendar = "BCAD", xlim = c(1500, 0), cex.axis = 0.8)
Axis(side = 1, at = seq(400, 1900, by = 100), cex.axis = 0.8)
mtext(side = 3, line = 2, cex = 1, "Datazioni Radiocarboniche")
mtext(side = 3, line = 3, cex = 2, "Grandi Carbonaie")
abline(v = 476, lty = 2, col = "black", lwd = 0.7)
abline(v = 1453, lty = 2, col = "black", lwd = 0.7)
abline(v = 1000, lty = 2, col = "black", lwd = 0.7)
abline(v = 1950, lty = 1, col = "black", lwd = 2)
abline(v = 1815, lty = 2, col = "black", lwd = 0.7)
# etichette periodi
shadowtext(738, -0.05, "Alto Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1226, -0.05, "Basso Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1634, -0.05, "Et\340 Moderna", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
dev.off()

# Valle del Sasso

vallesasso <- calibrate(x = c(480, 154), errors = c(45, 40),
    calCurves = "intcal20", ids = c("VMS001", "VMS043"))


jpeg("Multi14C_ValleSasso.jpg", width = 2500, height = 2000,
```

```
    units = "px", res = 300)
par(oma = c(0, 0, 1, 0), font.main = 1)
multiplot(vallesasso, decreasing = TRUE, HPD = TRUE, rescale = TRUE,
    calendar = "BCAD", xlim = c(1500, 0), cex.axis = 0.8)
Axis(side = 1, at = seq(400, 1900, by = 100), cex.axis = 0.8)
mtext(side = 3, line = 2, cex = 1, "Datazioni Radiocarboniche")
mtext(side = 3, line = 3, cex = 2, "Valle del Monte Sasso")
abline(v = 476, lty = 2, col = "black", lwd = 0.7)
abline(v = 1453, lty = 2, col = "black", lwd = 0.7)
abline(v = 1000, lty = 2, col = "black", lwd = 0.7)
abline(v = 1950, lty = 1, col = "black", lwd = 2)
abline(v = 1815, lty = 2, col = "black", lwd = 0.7)
# etichette periodi
shadowtext(738, -0.05, "Alto Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1226, -0.05, "Basso Medioevo", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
shadowtext(1634, -0.05, "Et\340 Moderna", col = "black", bg = "white",
    cex = 0.7, r = 0.2)
dev.off()
```

3 Evidenze raccolte durante il field survey

Per l'analisi dei dati archeologici ci si è avvalsi di diversi componenti aggiuntivi per RStudio, utilizzati sia per la resa grafica che per l'analisi dei dati:

```
library(sf)
library(dplyr)
library(tidyr)
library(scales)
library(ggplot2)
library(patchwork)
library(shadowtext)
```

3.1 Preparazione dei dati raccolti

Importazione dei dati raccolti, contenuti nel database *Spatialite* connesso a QGIS.

```
st_layers("D:/Archeologia/Carona/Database/SBC_DB.db")
```

```
Driver: SQLite
Available layers:
              layer_name        geometry_type features
1          Survey_Carona 3D Multi Line String      103
2    Sentieri_IGM1933-39    Multi Line String       90
3       Sentieri_CTR2016    Multi Line String      115
4             Scavi_MABG                Point       27
5            Alpeggi_SBC                Point       34
6  Condotte_Forzate_ENEL          Line String        3
7      Alpeggi_ValFondra                Point      109
8         Viabilit\340_1812          Line String       11
9      Gippabile Moderna          Line String        4
10   Mulattiere_Militari          Line String        3
11  Sentieri_Abbandonati          Line String       12
12          Sentieri_CAI 3D Multi Line String       18
13           Foto_VVN001                Point       33
14 Foto_SaggiMABG_nositi                Point       30
15           Scavi_UNITN                Point       10
16    Foto_Paesaggio_SBC                Point     8172
17         Datazioni_14C                Point       19
18   Toponimi_Catasto_LV                Point      239
19           Aree_Codici              Polygon       15
20         Foto_Siti_SBC                Point     7579
21              Siti_SBC                Point      829
   fields
1       6
2       4
3       4
4       7
5      12
6       3
7       8
8       2
9       2
10      2
11      3
12      6
13     18
14     17
15      7
```

```
16     17
17     16
18     11
19      5
20     16
21     43
```

Estrazione dei dati relativi alle evidenze mappate, comprendenti informazioni sulla loro posizione, descrizione e caratteristiche legate al territorio. Successiva verifica e riclassificazione di alcune tipologie di dati: il campo contenente i valori predittivi non deve contenere valori nulli, alcuni dati devono essere classificati come fattoriali per agevolarne l'analisi e la rappresentazione grafica.

```
Siti_SBC <- st_read("D:/Archeologia/Carona/Database/SBC_DB.db",
    "Siti_SBC")
```

```
Reading layer `Siti_SBC' from data source
  `D:\Archeologia\Carona\Database\SBC_DB.db'
  using driver `SQLite'
Simple feature collection with 829 features and 43 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 561685.4 ymin: 5094817 xmax: 569944.5 ymax: 5099935
Projected CRS: WGS 84 / UTM zone 32N
```

```
class(Siti_SBC)
```

```
[1] "sf"         "data.frame"
```

```
colnames(Siti_SBC)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
class(Siti_SBC$pred_mod)
```

```
[1] "numeric"
```

```
summary(is.na(Siti_SBC$pred_mod))
```

```
   Mode   FALSE
logical     829
```

```
class(Siti_SBC$Valanghe)
```

```
[1] "integer"
```

```
Siti_SBC$Valanghe <- factor(Siti_SBC$Valanghe, levels = c("1",
    "2", "3"))
class(Siti_SBC$Valanghe)
```

```
[1] "factor"
```

```
table(Siti_SBC$Valanghe)
```

```
  1   2   3
175  68 586
```

```
Siti_SBC$dusaf18 <- factor(Siti_SBC$dusaf18, levels = c("Boschi",
    "Cespuglieti", "Vegetazione rada", "Praterie arborate", "Praterie",
    "Non vegetato"))
levels(Siti_SBC$dusaf18)
```

```
[1] "Boschi"            "Cespuglieti"
[3] "Vegetazione rada"  "Praterie arborate"
[5] "Praterie"          "Non vegetato"
```

```
table(Siti_SBC$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
              204                26               151
Praterie arborate          Praterie      Non vegetato
               23               304               121
```

Codifica cromatica delle evidenze, dei gruppi e delle tipologie vegetali da utilizzare per la resa grafica.

```
type_colors <- c("lemonchiffon2", "firebrick3", "darkorange1",
    "dodgerblue3", "tan3", "darkolivegreen4", "lightpink2", "orchid4",
    "slategray", "turquoise3", "springgreen4", "violetred3",
    "gold2", "khaki3", "khaki4", "saddlebrown", "red", "wheat4",
    "palegreen1", "greenyellow")
names(type_colors) <- levels(factor(Siti_SBC$tipo))
type_colors
```

```
            Altro             Baita         Carbonaia
  "lemonchiffon2"      "firebrick3"     "darkorange1"
    Idroelettrico        Indefinito          Militare
    "dodgerblue3"            "tan3" "darkolivegreen4"
        Minerario           Miniera              Muro
     "lightpink2"         "orchid4"       "slategray"
           Ponte           Recinto           Reglana
     "turquoise3"    "springgreen4"      "violetred3"
         Ricovero            Riparo   Riparo Naturale
          "gold2"          "khaki3"          "khaki4"
          Roccolo          Sentieri           Smarino
    "saddlebrown"             "red"          "wheat4"
          Spiazzo           Stalla
     "palegreen1"     "greenyellow"
```

```
prod_colors <- c("darkorange1", "maroon3", "limegreen")
names(prod_colors) <- c("carbonaie", "minerarie", "pastorali")
prod_colors
```

```
    carbonaie     minerarie     pastorali
"darkorange1"     "maroon3"   "limegreen"
```

```
veg_colors <- c("springgreen4", "seagreen3", "peachpuff4", "olivedrab1",
    "olivedrab4", "palegreen")
names(veg_colors) <- c("Boschi", "Cespuglieti", "Non vegetato",
    "Praterie", "Praterie arborate", "Vegetazione rada")
veg_colors
```

```
         Boschi       Cespuglieti      Non vegetato
  "springgreen4"       "seagreen3"      "peachpuff4"
       Praterie Praterie arborate  Vegetazione rada
    "olivedrab1"      "olivedrab4"       "palegreen"
```

Creazione di un vettore di servizio, contenente i codici delle aree d'interesse.

```
aree_cod <- c("ARM", "BDC", "CAB", "DDS", "DSI", "FPP", "MAD",
    "MRS", "PAG", "PDS", "PRS", "VDF", "VLN", "VMS", "VSB")
aree_cod
```

3.2 Analisi generale delle evidenze raccolte

Conteggio delle evidenze per tipologia e calcolo percentuale della consistenza di queste ultime.

```
setwd("D:/Archeologia/Carona/Analisi")

contositi <- count(Siti_SBC, tipo)
contositi <- st_set_geometry(contositi, NULL)
contositi$perc <- contositi$n/nrow(Siti_SBC) * 100
contositi$area <- "SBC"
```

Rappresentazione grafica delle evidenze classificate per tipologia, tramite un ortogramma e un diagramma circolare.

```
# ortogramma
jpeg("Evidenze_Survey_2021_color.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(contositi, aes(x = n, y = reorder(tipo, n), fill = tipo)) +
    geom_col(position = position_dodge2(preserve = "single",
        padding = 0), width = 1, color = "black", show.legend = FALSE) +
    labs(title = "Evidenze antropiche rilevate", subtitle = paste("Totale:",
        sum(contositi$n)), x = "", y = "") + geom_text(aes(label = paste0(n,
    " (", round(perc, 1), "%)"), hjust = -0.1)) + scale_x_continuous(limits = c(0,
    175)) + scale_fill_manual(values = type_colors) + theme(axis.text = element_text(size = 12,
    colour = "black")) + theme(plot.title = element_text(hjust = 0.25,
    size = 20)) + theme(plot.margin = unit(c(10, 30, 10, 10),
    "pt"))
dev.off()

# diagramma circolare
jpeg("Evidenze_Survey_2021_pie.jpg", width = 2300, height = 2000,
    units = "px", res = 300)
ggplot(contositi, aes(x = area, y = perc, fill = tipo)) + geom_col(width = 1,
    color = "black", show.legend = TRUE) + coord_polar(theta = "y",
    direction = -1) + labs(title = "Evidenze antropiche rilevate",
    subtitle = paste("Totale:", sum(contositi$n)), x = "", y = "") +
    scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_shadowtext(aes(x = 1.55,
    y = perc, label = paste0(round(perc, 1), "%")), position = position_stack(vjust = 0.5),
    size = 2.5, bg.colour = "gray94", color = "black") + geom_shadowtext(aes(x = 1.1,
    label = ifelse(perc >= 2, paste0(tipo), "")), position = position_stack(vjust = 0.5),
    size = 3, bg.colour = "gray94", color = "black")
dev.off()
```

3.3 Baite

Analisi dei dati relativi alle baite, scorporati dal *record* generale dei siti. Riclassificazione di alcuni tipi di dato, per agevolare le successive operazioni di analisi e rappresentazione grafica.

```
setwd("D:/Archeologia/Carona/Analisi/Baite")

baite <- subset(Siti_SBC, tipo == "Baita")
class(baite)
```

```
[1] "sf"         "data.frame"
```

```
nrow(baite)
```

```
[1] 156
```

```
colnames(baite)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
baite$Degrado <- factor(baite$Degrado, levels = c("Sepolta",
    "Rasata", "Rudere", "Crollata", "Abbandonata", "In uso"))
table(baite$Degrado)
```

```
    Sepolta      Rasata      Rudere    Crollata Abbandonata
          9          36          28          24          14
     In uso
         45
```

```
baite$degrado_cat <- factor(baite$degrado_cat, levels = c("E",
    "D", "C", "B", "A"))
table(baite$degrado_cat)
```

```
 E  D  C  B  A
 9 36 28 24 59
```

```
baite$Costruzione <- factor(baite$Costruzione, levels = c("pre-XIXsec.",
    "pre-1810s", "1830s-1880s", "1910s-1930s", "1940s-1950s",
    "1960s", "post-1980s"))
table(baite$Costruzione)
```

```
pre-XIXsec.   pre-1810s 1830s-1880s 1910s-1930s 1940s-1950s
         38          57          23          15          13
      1960s  post-1980s
          5           5
```

```
baite$Abbandono <- factor(baite$Abbandono, levels = c("pre-XIXsec.",
    "1830s-1880s", "1910s-1930s", "1940s-1960s", "1960s-1970s",
```

```
    "1970s-1980s", "fine XXsec.", "inizi XXIsec.", "in uso"))
table(baite$Abbandono)
```

```
  pre-XIXsec.   1830s-1880s   1910s-1930s   1940s-1960s
           38             6             2            18
  1960s-1970s   1970s-1980s   fine XXsec. inizi XXIsec.
           13            10            11            13
       in uso
           45
```

```
baite$Durata_cat50 <- factor(baite$Durata_cat50, levels = c("ND",
    "200+", "150-200", "100-150", "50-100", "<50"))
table(baite$Durata_cat50)
```

```
     ND    200+ 150-200 100-150  50-100     <50
     38      21      28      22      32      15
```

```
baite$Durata_cat100 <- factor(baite$Durata_cat100, levels = c("ND",
    "200+", "100-200", "<100"))

table(baite$Durata_cat100)
```

```
     ND    200+ 100-200    <100
     38      21      52      45
```

```
baite$muratura <- factor(baite$muratura, levels = c("a secco",
    "con legante", "ND"))
table(baite$muratura)
```

```
    a secco con legante          ND
         80          71           5
```

```
baite$tecnica <- factor(baite$tecnica, levels = c("libera", "controterra",
    "controparete", "sottoroccia"))
table(baite$tecnica)
```

```
      libera  controterra controparete  sottoroccia
          57           73           14           12
```

```
baite$subclass <- factor(baite$subclass, levels = c("alpeggio",
    "mineraria", "altro", "villeggiatura", "indefinita", "Piani di Sasso",
    "Pagliari"))
table(baite$subclass)
```

```
      alpeggio      mineraria          altro  villeggiatura
           102             10              6              4
    indefinita Piani di Sasso       Pagliari
             8              6             20
```

Calcolo della media delle caratteristiche numeriche rilevate: misure interne delle strutture e spessore delle murature.

```
# MISURE INTERNE

# totale
round(mean(baite$Larghezza, na.rm = T), 1)
```

```
[1] 3.7
```

```
round(mean(baite$Profondita, na.rm = T), 1)
```

```
[1] 5.2
```

```
# integre
round(mean(subset(baite, baite$degrado_cat == "A")$Larghezza,
    na.rm = T), 1)
```

```
[1] 4.2
```

```
round(mean(subset(baite, baite$degrado_cat == "A")$Profondita,
    na.rm = T), 1)
```

```
[1] 4.8
```

```
# crollate
round(mean(subset(baite, baite$degrado_cat == "B")$Larghezza,
    na.rm = T), 1)
```

```
[1] 4.3
```

```
round(mean(subset(baite, baite$degrado_cat == "B")$Profondita,
    na.rm = T), 1)
```

```
[1] 6.7
```

```
# ruderi
round(mean(subset(baite, baite$degrado_cat == "C")$Larghezza,
    na.rm = T), 1)
```

```
[1] 3.5
```

```
round(mean(subset(baite, baite$degrado_cat == "C")$Profondita,
    na.rm = T), 1)
```

```
[1] 5.1
```

```
# rasate
round(mean(subset(baite, baite$degrado_cat == "D")$Larghezza,
    na.rm = T), 1)
```

```
[1] 3.3
```

```
round(mean(subset(baite, baite$degrado_cat == "D")$Profondita,
    na.rm = T), 1)
```

```
[1] 4.5
```

```
# sepolte
round(mean(subset(baite, baite$degrado_cat == "E")$Larghezza,
    na.rm = T), 1)
```

```
[1] 3.8
```

```
round(mean(subset(baite, baite$degrado_cat == "E")$Profondita,
    na.rm = T), 1)
```

```
[1] 4.9
```

```
# SPESSORE MURARIO

round(mean(baite$Spessore, na.rm = T), 2)
```

```
[1] 0.64
```

3.3.1 Caratteristiche strutturali

Resa grafica dei dati relativi al degrado strutturale e al tipo di murature.

```
# Plot degrado baite

jpeg("Degrado_totale_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(baite, aes(x = factor(degrado_cat))) + geom_bar(stat = "count",
    position = "dodge", fill = "goldenrod1", color = "firebrick4") +
    labs(title = "Classi di degrado delle Baite", subtitle = paste0("Totale evidenze:",
        nrow(baite)), x = "", y = "") + scale_x_discrete(labels = c("Sepolta",
    "Rasata", "Rudere", "Crollata", "Integra")) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 65)) + scale_y_continuous(breaks = seq(0,
    70, 10), labels = seq(0, 70, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Plot murature

jpeg("Leganti_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = muratura)) + geom_bar(stat = "count", position = "dodge",
    fill = "firebrick", color = "firebrick4") + labs(title = "Tipologie murarie delle Baite",
    subtitle = paste0("Totale evidenze:", nrow(baite)), x = "",
    y = "") + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 90)) + scale_y_continuous(breaks = seq(0,
    90, 10), labels = seq(0, 90, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

Analisi del rapporto tra il degrado strutturale e le tipologie murarie, e resa grafica.

```
degleg <- count(baite, Degrado, muratura)
degleg <- st_set_geometry(degleg, NULL)
degleg <- degleg %>%
    group_by(Degrado) %>%
    mutate(perc = n/sum(n) * 100)
```

```
jpeg("Degrado_Legante_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(degleg, aes(x = Degrado, y = n, fill = muratura)) + geom_bar(stat = "identity",
    position = "fill", color = "tan4") + labs(title = "Degrado strutturale delle baite (SBC)",
    subtitle = paste0("in rapporto alla tipologia di muratura"),
    x = "", y = "") + labs(fill = "Muratura:") + theme_bw() +
    theme(axis.text = element_text(size = 9, colour = "black")) +
    scale_fill_manual(values = c("bisque3", "darkgoldenrod3",
        "plum2")) + geom_text(stat = "identity", aes(label = paste0(round(perc,
    1), " %")), position = position_fill(0.5), size = 3, vjust = 0.5) +
    geom_text(stat = "identity", aes(label = n), position = position_fill(0.5),
        size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
```

```
dev.off()
```

3.3.2 Cronotipologia

Analisi del rapporto tra il degrado strutturale e la rappresentazione delle baite in cartografia storica.

```
degrado_baite <- data.frame(rbind(no_map = as.numeric(table(factor(subset(baite,
    baite$crono == 0)$degrado_cat, levels = c("A", "B", "C",
    "D", "E")))), `inizi XIXsec.` = as.numeric(table(factor(subset(baite,
    grepl(1812, baite$crono) | grepl(1819, baite$crono) | grepl(1833,
        baite$crono) | grepl(1855, baite$crono))$degrado_cat,
    levels = c("A", "B", "C", "D", "E")))), `1830s-1910s` = as.numeric(table(factor(subset(baite,
    grepl(1913, baite$crono))$degrado_cat, levels = c("A", "B",
    "C", "D", "E")))), `1920s-1940s` = as.numeric(table(factor(subset(baite,
    grepl(1933, baite$crono) | grepl(1939, baite$crono))$degrado_cat,
    levels = c("A", "B", "C", "D", "E")))), `1950s` = as.numeric(table(factor(subset(baite,
    grepl(1961, baite$crono))$degrado_cat, levels = c("A", "B",
    "C", "D", "E")))), `1960s` = as.numeric(table(factor(subset(baite,
    grepl(1971, baite$crono))$degrado_cat, levels = c("A", "B",
    "C", "D", "E")))), `1970s` = as.numeric(table(factor(subset(baite,
    grepl(1983, baite$crono))$degrado_cat, levels = c("A", "B",
    "C", "D", "E")))), `inzi XX sec.` = as.numeric(table(factor(subset(baite,
    grepl(2016, baite$crono))$degrado_cat, levels = c("A", "B",
    "C", "D", "E")))), `2010s` = as.numeric(table(factor(subset(baite,
    grepl(2020, baite$crono))$degrado_cat, levels = c("A", "B",
    "C", "D", "E"))))))

livelli <- c("no_map", "inizi XIXcent.", "1830s-1910s", "1920s-1940s",
    "1950s", "1960s", "1970s", "inzi XX sec.", "2010s")
degrado_baite$tot = rowSums(degrado_baite)
degrado_baite <- cbind(degrado_baite, livelli)
colnames(degrado_baite) <- c("A", "B", "C", "D", "E", "tot",
    "level")
degrado_baite$level <- factor(degrado_baite$level, levels = c("no_map",
    "inizi XIXcent.", "1830s-1910s", "1920s-1940s", "1950s",
    "1960s", "1970s", "inzi XX sec.", "2010s"))
```

Resa grafica dei dati relativi a degrado strutturale e livelli cartografici.

```
# Plot livelli cartografici

jpeg("Livelli_Cartografia_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(degrado_baite, aes(x = level, group = 1)) + geom_col(aes(y = tot),
    fill = "royalblue3", color = "midnightblue") + labs(title = "Baite mappate nella cartografia stor
    subtitle = paste0("Totale evidenze da field survey:", nrow(baite)),
    x = "", y = "") + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 90)) + scale_y_continuous(breaks = seq(0,
    90, 10)) + geom_text(stat = "identity", aes(y = tot, label = tot),
    size = 4, vjust = -2) + geom_text(stat = "identity", aes(y = tot,
    label = paste0("(", round((100 * tot)/nrow(baite), 1), " %)")),
    size = 4, vjust = -0.5) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Plot cartografia/degrado baite

jpeg("Degrado_Cartografia_Baite.jpg", width = 3000, height = 2000,
```

```
    units = "px", res = 300)
ggplot(degrado_baite, aes(x = level, group = 1)) + geom_line(aes(y = A,
    color = "Integra"), size = 1.2) + geom_line(aes(y = B, color = "Crollata"),
    size = 1.2) + geom_line(aes(y = C, color = "Rudere"), size = 1.2) +
    geom_line(aes(y = D, color = "Rasata"), size = 1.2) + geom_line(aes(y = E,
    color = "Sepolta"), size = 1.2) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + scale_colour_manual("Degrado \nattuale:",
    breaks = c("Integra", "Crollata", "Rudere", "Rasata", "Sepolta"),
    values = c("forestgreen", "gold1", "darkorange1", "firebrick1",
        "darkorchid4")) + labs(title = "Baite mappate in cartografia storica",
    subtitle = "e loro livello di degrado strutturale attuale",
    x = "", y = "") + scale_y_continuous(breaks = seq(0, 60,
    10))
dev.off()

# Plot pred_mod/degrado baite

jpeg("Degrado_predmod_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(baite, aes(x = degrado_cat, y = pred_mod)) + geom_boxplot() +
    labs(title = "Valori predittivi delle baite", subtitle = "Per le classi di degrado strutturale",
        x = "", y = "") + scale_y_continuous(breaks = c(0, 0.1,
    0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1), labels = c(0,
    0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)) + scale_x_discrete(labels = c("Sepolta",
    "Rasata", "Rudere", "Crollata", "Integra")) + geom_hline(yintercept = 0.3,
    color = "firebrick3")
dev.off()
```

La stima dei periodi di costruzione e abbandono delle strutture è stata ottenuta in ambiente QGIS e aggiunta al database delle evidenze attraverso il seguente codice:

```
ABBANDONO:
if("crono"LIKE'%2020%'AND "Degrado" = 'Abbandonata' , 'inizi XXIsec.',
if("crono"LIKE'%2020%'OR "Degrado" = 'In uso', 'in uso',
if("crono"LIKE'%2016%', 'inizi XXIsec.',
if("crono"LIKE'%1983%', 'fine XXsec.',
if("crono"LIKE'%1971%', '1970s-1980s',
if("crono"LIKE'%1961%', '1960s-1970s',
if("crono"LIKE'%1939%'OR"crono"LIKE '%1933%', '1940s-1960s',
if("crono"LIKE'%1913%', '1910s-1930s',
if("crono"LIKE'%1833%'OR"crono"LIKE'%1819%'OR"crono"LIKE'%1812%','1830s-1880s',
if("crono" = '0', 'pre-XIXsec.', 'NULL'))))))))))

COSTRUZIONE:
if("crono" = '0', 'pre-XIXsec.',
if("crono" LIKE'%1812%' OR "crono" LIKE '%1819%' OR "crono" LIKE
'%1833%','pre-1810s',
if("crono" LIKE'%1913%','1830s-1880s',
if("crono" LIKE'%1933%' OR "crono" LIKE '%1939%', '1910s-1930s',
if("crono" LIKE'%1961%','1940s-1950s',
if("crono" LIKE'%1971%', '1960s',
if("crono" LIKE'%1983%', '1970s',
if("crono" LIKE'%2016%' OR "crono" LIKE '%2020%', 'post-1980s', NULL))))))))
```

Resa grafica delle stime di costruzione e abbandono.

```
# Plot periodo di costruzione

jpeg("Costruzione_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = Costruzione)) + geom_bar(stat = "count",
    position = "dodge", color = "black", fill = "olivedrab4") +
    labs(title = "Stima del periodo di costruzione delle baite (SBC)",
        subtitle = paste0("Totale evidenze:", nrow(baite)), x = "",
        y = "") + theme_bw() + theme(axis.text = element_text(size = 9,
    colour = "black")) + expand_limits(y = c(0, 60)) + scale_y_continuous(breaks = seq(0,
    60, 10), labels = seq(0, 60, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Plot periodo di abbandono

jpeg("Abbandono_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = Abbandono)) + geom_bar(stat = "count",
    position = "dodge", color = "black", fill = c("orange4",
        "orange4", "orange4", "orange4", "orange4", "orange4",
        "orange4", "orange4", "khaki")) + labs(title = "Stima del periodo di abbandono delle baite (S
    subtitle = paste0("Totale evidenze:", nrow(baite)), x = "",
    y = "") + theme_bw() + theme(axis.text = element_text(size = 9,
    colour = "black")) + expand_limits(y = c(0, 50)) + scale_y_continuous(breaks = seq(0,
    50, 10), labels = seq(0, 50, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

Analisi del rapporto tra periodo di costruzione e degrado strutturale attuale.
Con relativa resa grafica.

```
antequem <- count(baite, Costruzione, Degrado)
antequem <- st_set_geometry(antequem, NULL)
antequem <- antequem %>%
    group_by(Costruzione) %>%
    mutate(perc = n/sum(n) * 100)
```

```
# Plot costruzione/degrado

jpeg("Costruzione_Degrado_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(antequem, aes(x = Costruzione, y = n, fill = Degrado)) +
    geom_bar(stat = "identity", position = "fill", color = "tan4") +
    labs(title = "Stima del periodo di costruzione delle baite (SBC)",
        subtitle = paste0("in rapporto allo stato attuale di degrado strutturale"),
        x = "", y = "") + labs(fill = "Degrado \nattuale:") +
    theme_bw() + theme(axis.text = element_text(size = 9, colour = "black")) +
    scale_fill_manual(values = c("darkorchid2", "firebrick1",
        "darkorange1", "gold1", "yellowgreen", "forestgreen")) +
    geom_text(stat = "identity", aes(label = paste0(round(perc,
```

```
        1), " %")), position = position_fill(0.5), size = 3,
        vjust = 0.5) + geom_text(stat = "identity", aes(label = n),
    position = position_fill(0.5), size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

Analisi del rapporto tra periodo di costruzione e tipologia muraria.
Con relativa resa grafica.

```
cleganti <- count(baite, Costruzione, muratura)
cleganti <- st_set_geometry(cleganti, NULL)
cleganti <- cleganti %>%
    group_by(Costruzione) %>%
    mutate(perc = n/sum(n) * 100)
```

```
# Plot costruzione/murature

jpeg("Costruzione_Murature_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(cleganti, aes(x = Costruzione, y = n, fill = muratura)) +
    geom_bar(stat = "identity", position = "fill", color = "tan4") +
    labs(title = "Stima del periodo di costruzione delle baite (SBC)",
        subtitle = paste0("con notazione della tipologia di muratura"),
        x = "", y = "") + labs(fill = "Muratura:") + theme_bw() +
    theme(axis.text = element_text(size = 9, colour = "black")) +
    scale_fill_manual(values = c("bisque3", "darkgoldenrod3",
        "plum2")) + geom_text(stat = "identity", aes(label = paste0(round(perc,
    1), " %")), position = position_fill(0.5), size = 3, vjust = 0.5) +
    geom_text(stat = "identity", aes(label = n), position = position_fill(0.5),
        size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

Analisi del rapporto tra periodo di abbandono e degrado strutturale attuale.
Con relativa resa grafica.

```
postquem <- count(baite, Abbandono, Degrado)
postquem <- st_set_geometry(postquem, NULL)
postquem <- postquem %>%
    group_by(Abbandono) %>%
    mutate(perc = n/sum(n) * 100)
```

```
# Plot abbandono/degrado

jpeg("Abbandono_Degrado_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(postquem, aes(x = Abbandono, y = n, fill = Degrado)) +
    geom_bar(stat = "identity", position = "fill", color = "tan4") +
    labs(title = "Stima del periodo di abbandono delle baite (SBC)",
        subtitle = paste0("in rapporto allo stato attuale di degrado strutturale"),
        x = "", y = "") + labs(fill = "Degrado \nattuale:") +
    theme_bw() + theme(axis.text.x = element_text(size = 8.5,
    colour = "black")) + scale_fill_manual(values = c("darkorchid2",
    "firebrick1", "darkorange1", "gold1", "yellowgreen", "forestgreen")) +
    geom_text(stat = "identity", aes(label = paste0(round(perc,
        1), " %")), position = position_fill(0.5), size = 3,
        vjust = 0.5) + geom_text(stat = "identity", aes(label = n),
    position = position_fill(0.5), size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
```

```
    30, 10, 10), "pt"))
dev.off()
```

Analisi del rapporto tra periodo di abbandono e tipologia muraria.
Con relativa resa grafica.

```
dleganti <- count(baite, Abbandono, muratura)
dleganti <- st_set_geometry(dleganti, NULL)
dleganti <- dleganti %>%
    group_by(Abbandono) %>%
    mutate(perc = n/sum(n) * 100)
```

```
# Plot abbandono/murature

jpeg("AbbandonoMuri_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(dleganti, aes(x = Abbandono, y = n, fill = muratura)) +
    geom_bar(stat = "identity", position = "fill", color = "tan4") +
    labs(title = "Stima del periodo di abbandono delle baite (SBC)",
        subtitle = paste0("con notazione della tipologia di muratura"),
        x = "", y = "") + labs(fill = "Muratura:") + theme_bw() +
    theme(axis.text = element_text(size = 9, colour = "black")) +
    scale_fill_manual(values = c("bisque3", "darkgoldenrod3",
        "plum2")) + geom_text(stat = "identity", aes(label = paste0(round(perc,
    1), " %")), position = position_fill(0.5), size = 3, vjust = 0.5) +
    geom_text(stat = "identity", aes(label = n), position = position_fill(0.5),
        size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

I dati sulla durata dell'uso delle baite, categorizzati in intervalli di 50 anni e di 100 anni sono stati ottenuti in ambiente QGIS e aggiunti direttamente al database delle evidenze, applicando il seguente codice **in linguaggio python**:

```
DATA MEDIA ABBANDONO:
if("Abbandono"='1830s-1880s',1861,if("Abbandono"='1910s-1930s',1923,
if("Abbandono"='1940s-1960s',1949,if("Abbandono"='1960s-1970s',1965,
if("Abbandono"='1970s-1980s',1975,if("Abbandono"='fine XXsec.',1999,
if("Abbandono"='inizi XXIsec.',2018,if("Abbandono"='in uso',2021,NULL))))))))

DATA MEDIA COSTRUZIONE:
if("Costruzione"='pre-1810s',1800,if("Costruzione"='1830s-1880s',1861,
if("Costruzione"='1910s-1930s',1923,if("Costruzione"='1940s-1950s',1949,
if("Costruzione"='1960s',1965, if("Costruzione"='post-1980s',2018,NULL))))))

DURATA CATEGORIZZATA (50):
if("durata"  >= 200,'200+',
if("durata"  < 200 AND "durata">=150, '150-200',
if("durata"  < 150 AND "durata"   >=  100, '100-150',
if("durata"< 100 AND
"durata"   >=  50, '50-100',
if("durata"  < 50, '<50','ND')))))

DURATA CATEGORIZZATA (100):
if("durata"  >= 200,'200+',
if("durata"  < 200 AND "durata">=100, '100-200',
if("durata"  < 100, '<100','ND')))
```

Resa grafica dei dati sulla durata.

```
# intervalli di 50 anni
jpeg("Durata50_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = Durata_cat50)) + geom_bar(stat = "count",
    position = "dodge", color = "firebrick4", fill = c("snow",
        "tan1", "tan1", "tan1", "tan1", "tan1")) + labs(title = "Stima della durata media delle Baite
    subtitle = paste0("Totale evidenze:", nrow(baite)), x = "anni",
    y = "") + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 50)) + scale_y_continuous(breaks = seq(0,
    50, 10), labels = seq(0, 50, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# intervalli di 100 anni
jpeg("Durata100_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = Durata_cat100)) + geom_bar(stat = "count",
    position = "dodge", color = "slateblue4", fill = c("snow",
        "tan3", "tan3", "tan3")) + labs(title = "Stima della durata media delle Baite (intervallo: 10
    subtitle = paste0("Totale evidenze:", nrow(baite)), x = "anni",
    y = "") + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 60)) + scale_y_continuous(breaks = seq(0,
    60, 10), labels = seq(0, 60, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

Rapporto tra degrado strutturale e durata stimata delle baite. Con relativa resa grafica.

```
deg50 <- count(baite, Durata_cat50, Degrado)
deg50 <- st_set_geometry(deg50, NULL)
deg50 <- deg50 %>%
    group_by(Durata_cat50) %>%
    mutate(perc = n/sum(n) * 100)

deg100 <- count(baite, Durata_cat100, Degrado)
deg100 <- st_set_geometry(deg100, NULL)
deg100 <- deg100 %>%
    group_by(Durata_cat100) %>%
    mutate(perc = n/sum(n) * 100)
```

Resa grafica delle stime di durata categorizzate.

```
# 50

jpeg("Durata50_Degrado_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(deg50, aes(x = Durata_cat50, y = n, fill = Degrado)) +
    geom_bar(stat = "identity", position = "fill") + labs(title = "Stima della durata media delle bai
    subtitle = paste0("in rapporto allo stato attuale di degrado strutturale"),
    x = "anni", y = "") + labs(fill = "Degrado \nattuale:") +
    theme_bw() + theme(axis.text.x = element_text(size = 10,
    colour = "black")) + scale_fill_manual(values = c("darkorchid2",
```

```
    "firebrick1", "darkorange1", "gold1", "yellowgreen", "forestgreen")) +
    geom_text(stat = "identity", aes(label = paste0(round(perc,
        1), " %")), position = position_fill(0.5), size = 3,
        vjust = 0.5) + geom_text(stat = "identity", aes(label = n),
    position = position_fill(0.5), size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# 100

jpeg("Durata100_Degrado_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(deg100, aes(x = Durata_cat100, y = n, fill = Degrado)) +
    geom_bar(stat = "identity", position = "fill") + labs(title = "Stima della durata media delle bai
    subtitle = paste0("in rapporto allo stato attuale di degrado strutturale"),
    x = "anni", y = "") + labs(fill = "Degrado \nattuale:") +
    theme_bw() + theme(axis.text.x = element_text(size = 10,
    colour = "black")) + scale_fill_manual(values = c("darkorchid2",
    "firebrick1", "darkorange1", "gold1", "yellowgreen", "forestgreen")) +
    geom_text(stat = "identity", aes(label = paste0(round(perc,
        1), " %")), position = position_fill(0.5), size = 3,
        vjust = 0.5) + geom_text(stat = "identity", aes(label = n),
    position = position_fill(0.5), size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

3.3.3 Rischio di valanghe

Test del chi-quadro, utilizzando la dimensione spaziale delle variabili (area) calcolata in ambiente GIS.

```
Vala_kmq <- c(8.811319, 4.833281, 18.212762)
s <- data.frame(table(factor(baite$Valanghe, levels = 1:3)))
n <- data.frame(Vala_kmq/sum(Vala_kmq))
t <- data.frame(cbind(s, n))
colnames(t) <- c("cat", "sites", "pexp")
```

Tabella 17: Rischio Valanghe - Baite

cat	sites	pexp
1	19	0.2765866
2	7	0.1517163
3	130	0.5716971

```
capture.output(chisq.test(t$sites, p = t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  t$sites"
[5] "X-squared = 43.931, df = 2, p-value = 2.887e-10"
[6] ""
```

Rapporto tra rischio valanghe e degrado strutturale attuale delle baite.

```
valadeg <- count(baite, Valanghe, Degrado)
valadeg <- st_set_geometry(valadeg, NULL)
valadeg <- valadeg %>%
```

```
    group_by(Valanghe) %>%
    mutate(perc = n/sum(n) * 100)
```

Resa grafica dei dati relativi al rischio di valanga.

```
jpeg("Valanghe_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)

p1 <- ggplot(baite, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = "dodge", fill = c("tomato4", "gold2", "olivedrab4"),
    color = "gray50") + labs(subtitle = paste0("Totale evidenze:",
    nrow(baite)), x = "", y = "") + scale_x_discrete(labels = c("Valanghe",
    "Zone Pericolose", "Nessun Rischio")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 140)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))

p2 <- ggplot(valadeg, aes(x = Valanghe, y = n, fill = Degrado)) +
    geom_bar(stat = "identity", position = "fill", color = "tan4") +
    labs(subtitle = paste0("Rapporto con il degrado strutturale"),
        x = "", y = "") + labs(fill = "Degrado \nattuale:") +
    theme_bw() + theme(axis.text.x = element_text(size = 8.5,
    colour = "black")) + scale_x_discrete(labels = c("Valanghe",
    "Zone Pericolose", "Nessun Rischio")) + scale_fill_manual(values = c("darkorchid2",
    "firebrick1", "darkorange1", "gold1", "yellowgreen", "forestgreen")) +
    geom_text(stat = "identity", aes(label = paste0(round(perc,
        1), " %")), position = position_fill(0.5), size = 3,
        vjust = 1.2) + geom_text(stat = "identity", aes(label = n),
    position = position_fill(0.5), size = 3, vjust = -0.3) +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))

p1 + p2 + plot_layout(ncol = 2) + plot_annotation(title = "Baite e rischio valanghe",
    caption = "Fonte dato valanghe: CLPV Reg. Lombardia (www.geoportale.regione.lombardia.it)") &
    theme(plot.title = element_text(size = 20, hjust = 0.5))
dev.off()
```

3.3.4 Tecnica costruttiva

Analisi del rapporto tra tecnica costruttiva e periodi di costruzione e abbandono delle baite. Con relative rese grafiche.

```
jpeg("Tecnica_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = tecnica)) + geom_bar(stat = "count", position = "dodge",
    fill = "saddlebrown", color = "firebrick4") + labs(title = "Tipologia costruttiva delle baite (SB
    subtitle = paste0("Totale evidenze:", nrow(baite)), x = "",
    y = "") + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 75)) + scale_y_continuous(breaks = seq(0,
    70, 10), labels = seq(0, 70, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

```
# costruzione
costec <- count(baite, Costruzione, tecnica)
```

```
costec <- st_set_geometry(costec, NULL)
costec$perc <- costec$n/sum(costec$n) * 100
costec <- costec %>%
    group_by(Costruzione) %>%
    mutate(perc.per = n/sum(n) * 100)
costec <- costec %>%
    group_by(tecnica) %>%
    mutate(perc.tec = n/sum(n) * 100)
sum(costec$n)
```

```
[1] 156
```

```
# abbandono
abbtec <- count(baite, Abbandono, tecnica)
abbtec <- st_set_geometry(abbtec, NULL)
abbtec$perc <- abbtec$n/sum(abbtec$n) * 100
abbtec <- abbtec %>%
    group_by(Abbandono) %>%
    mutate(perc.per = n/sum(n) * 100)
abbtec <- abbtec %>%
    group_by(tecnica) %>%
    mutate(perc.tec = n/sum(n) * 100)
sum(abbtec$n)
```

```
[1] 156
```

```
# costruzione
jpeg("Tecnica_Costruzione_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(costec, aes(x = Costruzione, y = n, fill = tecnica)) +
    geom_bar(stat = "identity", position = "fill", color = "black") +
    labs(title = "Stima del periodo di costruzione delle baite (SBC)",
        subtitle = paste0("con notazione della tecnica costruttiva"),
        x = "", y = "") + labs(fill = "Tecnica:") + scale_fill_manual(values = c("limegreen",
    "orange3", "thistle4", "maroon")) + theme_bw() + theme(axis.text = element_text(size = 9,
    colour = "black")) + geom_text(stat = "identity", aes(label = paste0(round(perc.per,
    1), " %")), position = position_fill(0.5), size = 3, vjust = 0.5) +
    geom_text(stat = "identity", aes(label = n), position = position_fill(0.5),
        size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
# abbandono
jpeg("Tecnica_Abbandono_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(abbtec, aes(x = Abbandono, y = n, fill = tecnica)) + geom_bar(stat = "identity",
    position = "fill", color = "black") + labs(title = "Stima del periodo di abbandono delle baite (S
    subtitle = paste0("con notazione della tecnica costruttiva"),
    x = "", y = "") + labs(fill = "Tecnica:") + scale_fill_manual(values = c("limegreen",
    "orange3", "thistle4", "maroon")) + theme_bw() + theme(axis.text = element_text(size = 9,
    colour = "black")) + geom_text(stat = "identity", aes(label = paste0(round(perc.per,
    1), " %")), position = position_fill(0.5), size = 3, vjust = 0.5) +
    geom_text(stat = "identity", aes(label = n), position = position_fill(0.5),
        size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

3.3.5 Copertura vegetale

Resa grafica dei dati della copertura vegetale del suolo relativi al posizionamento delle baite.

```
jpeg("Vegetazione_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F, color = "black") + scale_fill_manual(values = veg_colors) +
    labs(title = "Copertura del suolo nei siti delle baite",
        subtitle = paste0("Totale evidenze:", nrow(baite), " - ",
            round(nrow(baite)/nrow(Siti_SBC) * 100, 2), "% delle evidenze totali"),
        caption = "Dato copertura del suolo: DUSAF2018 Regione Lombardia - rielaborato E.Croce",
        x = "", y = "") + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 95)) + scale_y_continuous(breaks = seq(0,
    90, 10), labels = seq(0, 90, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.caption = element_text(size = 12,
    colour = "black", hjust = 0.5)) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

3.3.6 Funzione delle baite

Resa grafica dei dati relativi all'interpretazione funzionale delle baite.

```
jpeg("Funzione_Baite.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(baite, aes(x = subclass)) + geom_bar(stat = "count", position = "dodge",
    fill = "springgreen4", color = "goldenrod1") + labs(title = "Interpretazione funzionale delle bai
    subtitle = paste0("Totale evidenze:", nrow(baite)), caption = "Nota: nella classe Pagliari non so
    x = "", y = "") + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 110)) + scale_y_continuous(breaks = seq(0,
    110, 20), labels = seq(0, 110, 20)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

Classificazione dei valori sulla superficie predittiva delle baite in relazione alla loro funzione e alla soglia predittiva individuata. Con resa grafica.

```
predbaite1 <- data.frame(baite$cruz, baite$subclass, baite$pred_mod)
colnames(predbaite1) = c("cruz", "subclass", "pred_mod")
predbaite1 <- predbaite1 %>%
    mutate(pred_class = cut(predbaite1$pred_mod, breaks = c(-1,
        0.3, 1.1), labels = c("sotto", "sopra")))
predbaite <- count(predbaite1, subclass, pred_class)
predbaite$pred_class <- factor(predbaite$pred_class, levels = c("sopra",
    "sotto"))
predbaite <- predbaite %>%
    group_by(subclass) %>%
    mutate(perc = n/sum(n) * 100)
```

```
jpeg("Funzione_predmod_Baite.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(predbaite, aes(x = subclass, y = n, fill = pred_class)) +
    geom_bar(stat = "identity", position = "fill", color = "black") +
```

```
    labs(title = "Interpretazione funzionale delle baite (SBC)",
        subtitle = paste0("con notazione del valore predittivo"),
        x = "", y = "") + labs(fill = "Soglia di 0.3:") + scale_fill_manual(values = c("forestgreen",
    "firebrick3")) + theme_bw() + theme(axis.text = element_text(size = 9,
    colour = "black")) + geom_text(stat = "identity", aes(label = paste0(round(perc,
    1), " %")), position = position_fill(0.5), size = 3, vjust = 0.5) +
    geom_text(stat = "identity", aes(label = n), position = position_fill(0.5),
        size = 3, vjust = -1) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

3.4 Strutture pastorali

Le strutture correlabili con attività di allevamento sono i **recinti**, i **ricoveri**, le **stalle** e gli **spiazzi**.

```
setwd("D:/Archeologia/Carona/Analisi/Alpeggi")

# Recinti

recinti <- subset(Siti_SBC, tipo == "Recinto")
class(recinti)
```

```
[1] "sf"         "data.frame"
```

```
nrow(recinti)
```

```
[1] 53
```

```
colnames(recinti)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(recinti$pred_mod))
```

```
   Mode   FALSE
logical      53
```

```
table(recinti$Valanghe)
```

```
 1  2  3
20  4 29
```

```
table(recinti$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                0                 0                 9
Praterie arborate          Praterie      Non vegetato
                0                37                 7
```

```
# Ricoveri

ricoveri <- subset(Siti_SBC, tipo == "Ricovero")
class(ricoveri)
```

```
[1] "sf"         "data.frame"
```

```
nrow(ricoveri)
```

```
[1] 65
```

```
colnames(ricoveri)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(ricoveri$pred_mod))
```

```
   Mode   FALSE
logical      65
```

```
table(ricoveri$Valanghe)
```

```
 1  2  3
15  4 46
```

```
table(ricoveri$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                3                 0                18
Praterie arborate          Praterie      Non vegetato
                0                34                10
```

```
# Stalle

stalle <- subset(Siti_SBC, tipo == "Stalla")
class(stalle)
```

```
[1] "sf"         "data.frame"
```

```
nrow(stalle)
```

```
[1] 9
```

```
colnames(stalle)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
```

```
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(stalle$pred_mod))
```

```
   Mode   FALSE
logical       9
```

```
table(stalle$Valanghe)
```

```
1 2 3
0 0 9
```

```
table(stalle$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                1                 0                 0
Praterie arborate          Praterie      Non vegetato
                0                 8                 0
```

```
# Spiazzi

spiazzi <- subset(Siti_SBC, tipo == "Spiazzo")
class(spiazzi)
```

```
[1] "sf"         "data.frame"
```

```
nrow(spiazzi)
```

```
[1] 12
```

```
colnames(spiazzi)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(spiazzi$pred_mod))
```

```
   Mode   FALSE
logical      12
```

```
table(spiazzi$Valanghe)
```

```
1 2 3
5 1 6
```

```
table(spiazzi$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                2                 1                 5
```

```
Praterie arborate          Praterie      Non vegetato
                1                  2                 1
```

Analisi delle misure risultanti dal rilievo delle strutture.

```
# Recinti (superficie delimitata)

class(recinti$Superficie)
```

```
[1] "numeric"
```

```
# minima
round(min(recinti$Superficie), 1)
```

```
[1] 10.8
```

```
# massima
round(max(recinti$Superficie), 1)
```

```
[1] 3119.3
```

```
# media
round(mean(recinti$Superficie), 1)
```

```
[1] 584.9
```

```
# mediana
round(median(recinti$Superficie), 1)
```

```
[1] 173.3
```

```
sort(recinti$Superficie)
```

```
 [1]   10.842   12.579   18.673   20.410   21.270   21.483
 [7]   23.438   27.358   38.644   41.821   44.175   49.950
[13]   50.398   73.448   85.133   89.440   97.204   98.994
[19]  106.613  107.921  108.356  115.912  119.856  128.210
[25]  135.954  139.323  173.262  208.792  239.590  249.815
[31]  286.257  286.690  313.309  315.548  378.864  396.105
[37]  397.153  424.535  497.101  557.275  665.855  999.614
[43] 1282.668 1461.759 1632.485 1634.486 1656.045 1733.392
[49] 2251.283 2361.360 2815.730 2871.580 3119.299
```

```
# Recinti del sito VSB028 della val Sambuzza - Area totale
round(sum(subset(recinti, recinti$codice == "VSB028")$Superficie),
    2)
```

```
[1] 2392.06
```

```
# Ricoveri (misure interne)

# minima
round(min(ricoveri$Larghezza, na.rm = T), 1)
```

```
[1] 0.9
```

```
round(min(ricoveri$Profondita, na.rm = T), 1)
```

```
[1] 1.3
```

```
# massima
round(max(ricoveri$Larghezza, na.rm = T), 1)
```

```
[1] 4.2
```

```
round(max(ricoveri$Profondita, na.rm = T), 1)
```

```
[1] 5.3
```

```
# media
round(mean(ricoveri$Larghezza, na.rm = T), 1)
```

```
[1] 2
```

```
round(mean(ricoveri$Profondita, na.rm = T), 1)
```

```
[1] 2.5
```

```
# mediana
round(median(ricoveri$Larghezza, na.rm = T), 1)
```

```
[1] 1.7
```

```
round(median(ricoveri$Profondita, na.rm = T), 1)
```

```
[1] 2.3
```

3.4.1 Rischio di valanga

Analisi statistica della significatività della variabile per ogni classe di evidenze (chi-quadro con aree).

```
# area delle categorie in ordine 1,2,3
Vala_kmq <- c(8.811319, 4.833281, 18.212762)

# Recinti
Rec_s <- data.frame(table(factor(recinti$Valanghe, levels = 1:3)))
Rec_n <- data.frame(Vala_kmq/sum(Vala_kmq))
Rec_t <- data.frame(cbind(Rec_s, Rec_n))
colnames(Rec_t) <- c("cat", "sites", "pexp")
```

cat	sites	pexp
1	20	0.2765866
2	4	0.1517163
3	29	0.5716971

```
# Ricoveri
ric_s <- data.frame(table(factor(ricoveri$Valanghe, levels = 1:3)))
ric_n <- data.frame(Vala_kmq/sum(Vala_kmq))
ric_t <- data.frame(cbind(ric_s, ric_n))
colnames(ric_t) <- c("cat", "sites", "pexp")
```

cat	sites	pexp
1	15	0.2765866
2	4	0.1517163
3	46	0.5716971

```
# Spiazzi
spi_s <- data.frame(table(factor(spiazzi$Valanghe, levels = 1:3)))
spi_n <- data.frame(Vala_kmq/sum(Vala_kmq))
spi_t <- data.frame(cbind(spi_s, spi_n))
colnames(spi_t) <- c("cat", "sites", "pexp")
```

cat	sites	pexp
1	5	0.2765866
2	1	0.1517163
3	6	0.5716971

```
# Stalle
sta_s <- data.frame(table(factor(stalle$Valanghe, levels = 1:3)))
sta_n <- data.frame(Vala_kmq/sum(Vala_kmq))
sta_t <- data.frame(cbind(sta_s, sta_n))
colnames(sta_t) <- c("cat", "sites", "pexp")
```

cat	sites	pexp
1	0	0.2765866
2	0	0.1517163
3	9	0.5716971

```
# recinti
capture.output(chisq.test(Rec_t$sites, p = Rec_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  Rec_t$sites"
[5] "X-squared = 4.0325, df = 2, p-value = 0.1332"
[6] ""
```

```
# ricoveri
capture.output(chisq.test(ric_t$sites, p = ric_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  ric_t$sites"
[5] "X-squared = 6.0801, df = 2, p-value = 0.04783"
[6] ""
```

```
# spiazzi
capture.output(chisq.test(spi_t$sites, p = spi_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  spi_t$sites"
[5] "X-squared = 1.3291, df = 2, p-value = 0.5145"
[6] ""
```

```
# stalle
capture.output(chisq.test(sta_t$sites, p = sta_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  sta_t$sites"
[5] "X-squared = 6.7426, df = 2, p-value = 0.03434"
[6] ""
```

Resa grafica dei dati relativi al rischio di valanghe.

```
jpeg("Valanghe_Alpeggi.jpg", width = 3000, height = 2000, units = "px",
    res = 300)

p1 <- ggplot(recinti, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = "dodge", fill = c("tomato4", "gold2", "olivedrab4"),
    color = "gray50") + labs(title = "Recinti", subtitle = paste0("Totale evidenze: ",
    nrow(recinti)), x = "", y = "") + scale_x_discrete(labels = c("Valanghe",
    "Zone Pericolose", "Nessun Rischio")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 35)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))

p2 <- ggplot(ricoveri, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = "dodge", fill = c("tomato4", "gold2", "olivedrab4"),
    color = "gray50") + labs(title = "Ricoveri", subtitle = paste0("Totale evidenze: ",
    nrow(ricoveri)), x = "", y = "") + scale_x_discrete(labels = c("Valanghe",
    "Zone Pericolose", "Nessun Rischio")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 60)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
p3 <- ggplot(spiazzi, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = "dodge", fill = c("tomato4", "gold2", "olivedrab4"),
    color = "gray50") + labs(title = "Spiazzi", subtitle = paste0("Totale evidenze: ",
    nrow(spiazzi)), x = "", y = "") + scale_x_discrete(labels = c("Valanghe",
    "Zone Pericolose", "Nessun Rischio")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 8)) + scale_y_continuous(breaks = seq(0,
    8, 2), labels = seq(0, 8, 2)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
p4 <- ggplot(stalle, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = position_dodge2(preserve = "single", padding = 0),
    fill = c("olivedrab4"), color = "gray50") + labs(title = "Stalle",
    subtitle = paste0("Totale evidenze: ", nrow(stalle)), x = "",
    y = "") + scale_x_discrete(labels = c("Nessun Rischio")) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 12)) + scale_y_continuous(breaks = seq(0,
    10, 2), labels = seq(0, 10, 2)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))

p1 + p2 + p3 + p4 + plot_layout(ncol = 2) + plot_annotation(title = "Strutture pastorali e rischio va
    theme(plot.title = element_text(size = 20, hjust = 0.5))
dev.off()
```

3.4.2 Copertura vegetale

Resa grafica dei dati sulla copertura vegetale nella posizione delle strutture pastorali.

```
jpeg("Vegetazione_Alpeggi.jpg", width = 3000, height = 2000,
    units = "px", res = 300)

p1 <- ggplot(recinti, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "Recinti",
    subtitle = paste0("Totale evidenze: ", nrow(recinti)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    geom_text(stat = "count", aes(label = ..count..), vjust = -2) +
    geom_text(stat = "count", aes(label = paste0("(", round((100 *
        (..count..))/sum(..count..), 1), "%)"), vjust = -0.6),
        size = 4) + expand_limits(y = c(0, 50)) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))

p2 <- ggplot(ricoveri, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "Ricoveri",
    subtitle = paste0("Totale evidenze: ", nrow(ricoveri)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    geom_text(stat = "count", aes(label = ..count..), vjust = -2) +
    geom_text(stat = "count", aes(label = paste0("(", round((100 *
        (..count..))/sum(..count..), 1), "%)"), vjust = -0.6),
        size = 4) + expand_limits(y = c(0, 50)) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))

p3 <- ggplot(spiazzi, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "Spiazzi",
    subtitle = paste0("Totale evidenze: ", nrow(spiazzi)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    geom_text(stat = "count", aes(label = ..count..), vjust = -2) +
    geom_text(stat = "count", aes(label = paste0("(", round((100 *
        (..count..))/sum(..count..), 1), "%)"), vjust = -0.6),
        size = 4) + expand_limits(y = c(0, 8)) + theme(axis.text.x = element_text(size = 6,
    colour = "black")) + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))

p4 <- ggplot(stalle, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "Stalle",
    subtitle = paste0("Totale evidenze: ", nrow(stalle)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    geom_text(stat = "count", aes(label = ..count..), vjust = -2) +
    geom_text(stat = "count", aes(label = paste0("(", round((100 *
        (..count..))/sum(..count..), 1), "%)"), vjust = -0.6),
        size = 4) + expand_limits(y = c(0, 10)) + scale_y_continuous(breaks = seq(0,
    10, 2), labels = seq(0, 10, 2)) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))

p1 + p2 + p3 + p4 + plot_layout(ncol = 2) + plot_annotation(title = "Strutture pastorali e copertura
    caption = "Dato copertura del suolo:
                  DUSAF2018 Regione Lombardia - rielaborato E.Croce") &
    theme(plot.title = element_text(size = 20, hjust = 0.5))
dev.off()
```

3.5 Carbonaie

Scorporo dei dati riguardanti le carbonaie dal record generale delle evidenze, con riclassificazione del campo relativo alla tipologia interna del tipo e controllo delle variabili da analizzare.

```
setwd("D:/Archeologia/Carona/Analisi/Carbonaie")

carbonaie <- subset(Siti_SBC, tipo == "Carbonaia")
class(carbonaie)
```

```
[1] "sf"         "data.frame"
```

```
nrow(carbonaie)
```

```
[1] 46
```

```
colnames(carbonaie)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(carbonaie$pred_mod))
```

```
   Mode   FALSE
logical      46
```

```
table(carbonaie$Valanghe)
```

```
 1  2  3
 4  1 41
```

```
table(carbonaie$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
               31                 1                 0
Praterie arborate          Praterie      Non vegetato
                4                10                 0
```

```
carbonaie$subclass <- factor(carbonaie$subclass, levels = c("Versante",
    "Piano", "Conca", "ND"))
table(carbonaie$subclass)
```

```
Versante    Piano    Conca       ND
      37        4        4        1
```

Calcolo delle misure medie delle carbonaie, per tipologia.

```
# carbonaie su versante
versante <- subset(carbonaie, subclass == "Versante")
round(mean(versante$Larghezza, na.rm = T), 1)
```

```
[1] 7.6
```

```
round(mean(versante$Profondita, na.rm = T), 1)
```

```
[1] 6.4
```

```
# carbonaie in conca
conca <- subset(carbonaie, subclass == "Conca")
round(mean(conca$Larghezza, na.rm = T), 1)
```

```
[1] 8
```

```
round(mean(conca$Profondita, na.rm = T), 1)
```

```
[1] 6.6
```

Resa grafica dei dati quantitativi e tipologici relativi alle carbonaie.

```
jpeg("Carbonaie_tipi.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(carbonaie, aes(x = subclass)) + geom_bar(stat = "count",
    position = "dodge", fill = "darkorange1", color = "black") +
    labs(title = "Tipologia delle carbonaie", subtitle = paste0("Totale evidenze: ",
        nrow(carbonaie), " - ", round(nrow(carbonaie)/nrow(Siti_SBC) *
            100, 1), "% delle evidenze totali"), x = "", y = "") +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 40)) + scale_y_continuous(breaks = seq(0,
    40, 10), labels = seq(0, 40, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + labs(caption = paste0("Tipo su versante: misura media =
    round(mean(versante$Profondita, na.rm = T), 1), " x ", round(mean(versante$Larghezza,
        na.rm = T), 1), " metri.\n", "Tipo in conca: misura media = ",
    round(mean(conca$Profondita, na.rm = T), 1), " x ", round(mean(conca$Larghezza,
        na.rm = T), 1), " metri.")) + theme(plot.caption = element_text(size = 12,
    colour = "black", hjust = 0.5)) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

3.5.1 Copertura vegetale

Correlazione tra tipologia delle carbonaie e copertura vegetale del suolo.

```
carbovegtip <- count(carbonaie, subclass, dusaf18)
carbovegtip <- st_set_geometry(carbovegtip, NULL)
carbovegtip <- carbovegtip %>%
    group_by(subclass) %>%
    mutate(perc.tip = n/sum(n) * 100)
carbovegtip$perc.tot <- carbovegtip$n/sum(carbovegtip$n) * 100
sum(carbovegtip$perc.tot)
```

```
[1] 100
```

```
sum(carbovegtip$n)
```

```
[1] 46
```

Resa grafica dei dati relativi alla copertura vegetale.

```
# dato generico
jpeg("Carbonaie_vegetazione.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(carbonaie, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
```

```
    position = "dodge", show.legend = F, color = "saddlebrown") +
    scale_fill_manual(values = veg_colors, breaks = c("Boschi",
        "Cespuglieti", "Praterie arborate", "Praterie")) + labs(title = "Copertura del suolo nei siti
    subtitle = paste0("Totale evidenze: ", nrow(carbonaie), " - ",
        round(nrow(carbonaie)/nrow(Siti_SBC) * 100, 1), "% delle evidenze totali"),
    caption = "Dato copertura del suolo:
       DUSAF2018 Regione Lombardia - rielaborato E.Croce",
    x = "", y = "") + theme_bw() +  x = "", y = "") +
    x = "", y = "") + theme_bw() +  theme_bw() +
theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 35)) + scale_y_continuous(breaks = seq(0,
    30, 10), labels = seq(0, 30, 10)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        2), "%)"), vjust = -0.6), size = 4) + theme(plot.caption = element_text(size = 12,
    colour = "black", hjust = 0.5)) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# dato diviso per tipo
jpeg("Carbonaie_vegetazione_tipi.jpg", width = 2500, height = 2000,
    units = "px", res = 300)
ggplot(carbovegtip, aes(x = subclass, y = perc.tip, fill = dusaf18)) +
    geom_col(width = 1000, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "Copertura vegetale nei siti delle carbon
    subtitle = paste0("Totale evidenze: ", paste0(sum(carbovegtip$n),
        " - ", round((sum(carbovegtip$n))/nrow(Siti_SBC) * 100,
            1), "% delle evidenze totali")), caption = "Dato copertura del suolo: DUSAF2018 - rielabo
    x = "", y = "") + scale_fill_manual(values = veg_colors,
    breaks = c("Boschi", "Cespuglieti", "Praterie", "Praterie arborate")) +
    theme_void() + geom_shadowtext(aes(x = 1.2, y = perc.tip,
    label = paste0(round(perc.tip, 1), "%")), position = position_stack(vjust = 0.5),
    size = 3, bg.colour = "gray94", color = "black") + facet_wrap(~subclass,
    ncol = 2) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt")) + theme(plot.title = element_text(margin = margin(b = 10))) +
    theme(plot.subtitle = element_text(margin = margin(b = 12)))
dev.off()
```

3.5.2 Rischio di valanga

Validazione statistica della variabile valanghe in relazione alle carbonaie.

```
Vala_kmq <- c(8.811319, 4.833281, 18.212762)

Car_s <- data.frame(table(factor(carbonaie$Valanghe, levels = 1:3)))
Car_n <- data.frame(Vala_kmq/sum(Vala_kmq))
Car_t <- data.frame(cbind(Car_s, Car_n))
colnames(Car_t) <- c("cat", "sites", "pexp")
```

Tabella 22: Carbonaie e aree valanghive

cat	sites	pexp
1	4	0.2765866
2	1	0.1517163
3	41	0.5716971

```
capture.output(chisq.test(Car_t$sites, p = Car_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  Car_t$sites"
[5] "X-squared = 19.322, df = 2, p-value = 6.372e-05"
[6] ""
```

Resa grafica del rischio valanghe associato alle carbonaie.

```
jpeg("Valanghe_Carbonaie.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(carbonaie, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = "dodge", fill = c("tomato4", "gold2", "olivedrab4"),
    color = "gray50") + labs(title = "Carbonaie e rischio valanghe",
    subtitle = paste0("Totale evidenze: ", nrow(carbonaie)),
    x = "", y = "") + scale_x_discrete(labels = c("Valanghe",
    "Zone Pericolose", "Nessun Rischio")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 50)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

3.6 Ripari

Scorporo dei dati riguardanti i **ripari** dal record principale delle evidenze e validazione dei campi da analizzare.

```
setwd("D:/Archeologia/Carona/Analisi/Ripari")

# Ripari

ripari <- subset(Siti_SBC, tipo == "Riparo")
class(ripari)
```

```
[1] "sf"         "data.frame"
```

```
nrow(ripari)
```

```
[1] 134
```

```
colnames(ripari)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(ripari$pred_mod))
```

```
   Mode   FALSE
logical     134
```

```
table(ripari$Valanghe)
```

```
 1  2  3
32 24 78
```

```
table(ripari$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
               20                 3                33
Praterie arborate          Praterie      Non vegetato
                1                26                51
```

Calcolo delle misure medie dei ripari.

```
round(mean(ripari$Altezza, na.rm = T), 1)
```

```
[1] 1.5
```

```
round(mean(ripari$Larghezza, na.rm = T), 1)
```

```
[1] 2.8
```

```
round(mean(ripari$Profondita, na.rm = T), 1)
```

```
[1] 2.1
```

Scorporo dei dati riguardanti i **ripari naturali** dal record principale delle evidenze e validazione dei campi da analizzare.

```
ripari_N <- subset(Siti_SBC, tipo == "Riparo Naturale")
class(ripari_N)
```

```
[1] "sf"         "data.frame"
```

```
nrow(ripari_N)
```

```
[1] 38
```

```
colnames(ripari_N)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(ripari_N$pred_mod))
```

```
   Mode   FALSE
logical      38
```

```
table(ripari_N$Valanghe)
```

```
 1  2  3
12  5 21
```

```
table(ripari_N$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                2                 5                 8
Praterie arborate          Praterie      Non vegetato
                0                 2                21
```

3.6.1 Rischio di valanga

Validazione statistica della variabile valanghe in relazione ai ripari e ai ripari naturali.

```
Vala_kmq <- c(8.811319, 4.833281, 18.212762)
# Ripari
Rip_s <- data.frame(table(factor(ripari$Valanghe, levels = 1:3)))
Rip_n <- data.frame(Vala_kmq/sum(Vala_kmq))
Rip_t <- data.frame(cbind(Rip_s, Rip_n))
colnames(Rip_t) <- c("cat", "sites", "pexp")
```

Tabella 23: Ripari

cat	sites	pexp
1	32	0.2765866
2	24	0.1517163
3	78	0.5716971

```
capture.output(chisq.test(Rip_t$sites, p = Rip_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  Rip_t$sites"
[5] "X-squared = 1.3794, df = 2, p-value = 0.5017"
[6] ""
```

```
# Ripari Naturali
RipN_s <- data.frame(table(factor(ripari_N$Valanghe, levels = 1:3)))
RipN_n <- data.frame(Vala_kmq/sum(Vala_kmq))
RipN_t <- data.frame(cbind(RipN_s, RipN_n))
colnames(RipN_t) <- c("cat", "sites", "pexp")
```

Tabella 24: Ripari Naturali

cat	sites	pexp
1	12	0.2765866
2	5	0.1517163
3	21	0.5716971

```
capture.output(chisq.test(RipN_t$sites, p = RipN_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  RipN_t$sites"
[5] "X-squared = 0.33688, df = 2, p-value = 0.845"
[6] ""
```

Resa grafica dei dati relativi al rischio di valanghe

```
jpeg("Valanghe_Ripari.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
p1 <- ggplot(ripari, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = "dodge", fill = c("tomato4", "gold2", "olivedrab4"),
    color = "gray50") + labs(title = "Ripari", subtitle = paste0("Totale evidenze:",
    nrow(ripari)), x = "", y = "") + scale_x_discrete(labels = c("Valanghe",
    "Zone Pericolose", "Nessun Rischio")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 90)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
p2 <- ggplot(ripari_N, aes(x = Valanghe)) + geom_bar(stat = "count",
    position = "dodge", fill = c("tomato4", "gold2", "olivedrab4"),
    color = "gray50") + labs(title = "Ripari Naturali", subtitle = paste0("Totale evidenze:",
    nrow(ripari_N)), x = "", y = "") + scale_x_discrete(labels = c("Valanghe",
```

```
    "Zone Pericolose", "Nessun Rischio")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + expand_limits(y = c(0, 25)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
p1 + p2 + plot_layout(ncol = 2) + plot_annotation(title = "Rischio Valanghe") &
    theme(plot.title = element_text(size = 20, hjust = 0.5))
dev.off()
```

3.6.2 Copertura vegetale

Resa grafica delle coperture vegetali nelle posizioni di ripari e ripari naturali.

```
jpeg("Vegetazione_Ripari.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
p1 <- ggplot(ripari, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "Ripari",
    subtitle = paste0("Totale evidenze:", nrow(ripari)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 55)) + geom_text(stat = "count", aes(label = ..count..),
    vjust = -2) + geom_text(stat = "count", aes(label = paste0("(",
    round((100 * (..count..))/sum(..count..), 1), "%)"), vjust = -0.6),
    size = 4) + theme(axis.text.x = element_text(angle = 45,
    hjust = 1)) + theme(plot.margin = unit(c(10, 30, 10, 10),
    "pt"))
p2 <- ggplot(ripari_N, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "Ripari Naturali",
    subtitle = paste0("Totale evidenze:", nrow(ripari_N)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 24)) + geom_text(stat = "count", aes(label = ..count..),
    vjust = -2) + geom_text(stat = "count", aes(label = paste0("(",
    round((100 * (..count..))/sum(..count..), 1), "%)"), vjust = -0.6),
    size = 4) + theme(axis.text.x = element_text(angle = 45,
    hjust = 1)) + theme(plot.margin = unit(c(10, 30, 10, 10),
    "pt"))
p1 + p2 + plot_layout(ncol = 2) + plot_annotation(title = "Copertura Vegetale",
    caption = "Dato copertura del suolo: DUSAF2018 - rielaborato E.Croce") &
    theme(plot.title = element_text(size = 20, hjust = 0.5))
dev.off()
```

3.7 Strutture minerarie

Le strutture classificabili come minerarie sono quelle attribuite alle classi: **miniera**, **minerario**, **reglana**, **smarino**. Scorporo dei dati dal *record* principale delle evidenze e validazione dei campi da analizzare.

```
setwd("D:/Archeologia/Carona/Analisi/Miniere")

# Miniere

miniere <- subset(Siti_SBC, tipo == "Miniera")
class(miniere)
```

```
[1] "sf"         "data.frame"
```

```
nrow(miniere)
```

```
[1] 41
```

```
colnames(miniere)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(miniere$pred_mod))
```

```
   Mode   FALSE
logical      41
```

```
table(miniere$Valanghe)
```

```
 1  2  3
13  6 22
```

```
table(miniere$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
               21                 0                14
Praterie arborate          Praterie      Non vegetato
                1                 4                 1
```

```
# Reglane

reglane <- subset(Siti_SBC, tipo == "Reglana")
class(reglane)
```

```
[1] "sf"         "data.frame"
```

```
nrow(reglane)
```

```
[1] 11
```

```
colnames(reglane)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(reglane$pred_mod))
```

```
   Mode   FALSE
logical      11
```

```
table(reglane$Valanghe)
```

```
1 2 3
4 2 5
```

```
table(reglane$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                4                 0                 2
Praterie arborate          Praterie      Non vegetato
                1                 4                 0
```

```
# Minerario

minerari <- subset(Siti_SBC, tipo == "Minerario")
class(minerari)
```

```
[1] "sf"         "data.frame"
```

```
nrow(minerari)
```

```
[1] 11
```

```
colnames(minerari)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
```

```
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(minerari$pred_mod))
```

```
   Mode   FALSE
logical      11
```

```
table(minerari$Valanghe)
```

```
1 2 3
5 2 4
```

```
table(minerari$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                0                 0                 2
Praterie arborate          Praterie      Non vegetato
                0                 6                 3
```

```
# Smarini

smarini <- subset(Siti_SBC, tipo == "Smarino")
class(smarini)
```

```
[1] "sf"         "data.frame"
```

```
nrow(smarini)
```

```
[1] 10
```

```
colnames(smarini)
```

```
 [1] "cruz"           "area"           "codice"
 [4] "struttura"      "tipo"           "subclass"
 [7] "name"           "comm_name"      "note"
[10] "rilievo"        "time"           "pred_mod"
[13] "dimensioni"     "Altezza"        "Larghezza"
[16] "Profondita"     "Spessore"       "Superficie"
[19] "descrizione"    "Degrado"        "degrado_cat"
[22] "crono"          "Costruzione"    "Abbandono"
[25] "m.cost"         "m.abba"         "durata"
[28] "Durata_cat50"   "Durata_cat100"  "muratura"
[31] "tecnica"        "Valanghe"       "TWI"
[34] "Water_cost"     "dusaf18"        "Quota_DTM"
[37] "Quota_GPS"      "picture"        "path"
[40] "cruz_controllo" "X"              "Y"
[43] "Vala_text"      "geom"
```

```
summary(is.na(smarini$pred_mod))
```

```
   Mode   FALSE
logical      10
```

```
table(smarini$Valanghe)
```

```
1 2 3
3 0 7
```

```
table(smarini$dusaf18)
```

```
           Boschi       Cespuglieti  Vegetazione rada
                6                 0                 2
Praterie arborate          Praterie      Non vegetato
```

```
                0                 2                 0
```

3.7.1 Rischio di valanga

Validazione statistica della variabile valanghe in relazione alle evideze minerarie.

```
# area delle categorie in ordine 1,2,3
Vala_kmq <- c(8.811319, 4.833281, 18.212762)

# Miniere
Min_s <- data.frame(table(factor(miniere$Valanghe, levels = 1:3)))
Min_n <- data.frame(Vala_kmq/sum(Vala_kmq))
Min_t <- data.frame(cbind(Min_s, Min_n))
colnames(Min_t) <- c("cat", "sites", "pexp")
```

Tabella 25: Miniere/Valanghe

cat	sites	pexp
1	13	0.2765866
2	6	0.1517163
3	22	0.5716971

```
capture.output(chisq.test(Min_t$sites, p = Min_t$pexp))
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  Min_t$sites"
[5] "X-squared = 0.3392, df = 2, p-value = 0.844"
[6] ""
```

```
# Reglane
reg_s <- data.frame(table(factor(reglane$Valanghe, levels = 1:3)))
reg_n <- data.frame(Vala_kmq/sum(Vala_kmq))
reg_t <- data.frame(cbind(reg_s, reg_n))
colnames(reg_t) <- c("cat", "sites", "pexp")
```

Tabella 26: Reglane/Valanghe

cat	sites	pexp
1	4	0.2765866
2	2	0.1517163
3	5	0.5716971

```
capture.output(chisq.test(reg_t$sites, p = reg_t$pexp))
```

```
Warning in chisq.test(reg_t$sites, p = reg_t$pexp): Chi-
squared approximation may be incorrect
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  reg_t$sites"
[5] "X-squared = 0.63114, df = 2, p-value = 0.7294"
[6] ""
```

```
# Minerari
mii_s <- data.frame(table(factor(minerari$Valanghe, levels = 1:3)))
mii_n <- data.frame(Vala_kmq/sum(Vala_kmq))
mii_t <- data.frame(cbind(mii_s, mii_n))
colnames(mii_t) <- c("cat", "sites", "pexp")
```

Tabella 27: Mineriari/Valanghe

cat	sites	pexp
1	5	0.2765866
2	2	0.1517163
3	4	0.5716971

```
capture.output(chisq.test(mii_t$sites, p = mii_t$pexp))
```

```
Warning in chisq.test(mii_t$sites, p = mii_t$pexp): Chi-
squared approximation may be incorrect
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  mii_t$sites"
[5] "X-squared = 2.1581, df = 2, p-value = 0.3399"
[6] ""
```

```
# Smarini
sma_s <- data.frame(table(factor(smarini$Valanghe, levels = 1:3)))
sma_n <- data.frame(Vala_kmq/sum(Vala_kmq))
sma_t <- data.frame(cbind(sma_s, sma_n))
colnames(sma_t) <- c("cat", "sites", "pexp")
```

Tabella 28: Smarini/Valanghe

cat	sites	pexp
1	3	0.2765866
2	0	0.1517163
3	7	0.5716971

```
capture.output(chisq.test(sma_t$sites, p = sma_t$pexp))
```

```
Warning in chisq.test(sma_t$sites, p = sma_t$pexp): Chi-
squared approximation may be incorrect
```

```
[1] ""
[2] "\tChi-squared test for given probabilities"
[3] ""
[4] "data:  sma_t$sites"
[5] "X-squared = 1.8249, df = 2, p-value = 0.4015"
[6] ""
```

3.7.2 Copertura vegetale

Resa grafica delle coperture vegetali nelle posizioni delle evidenze minerarie.

```
jpeg("Vegetazione_Miniere.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
```

```
p1 <- ggplot(miniere, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "miniere",
    subtitle = paste0("Totale evidenze: ", nrow(miniere)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 30)) + geom_text(stat = "count", aes(label = ..count..),
    vjust = -2) + geom_text(stat = "count", aes(label = paste0("(",
    round((100 * (..count..))/sum(..count..), 1), "%)"), vjust = -0.6),
    size = 4) + theme(axis.text.x = element_text(size = 7.5,
    colour = "black")) + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))

p2 <- ggplot(reglane, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "reglane",
    subtitle = paste0("Totale evidenze: ", nrow(reglane)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 8)) + scale_y_continuous(breaks = seq(0,
    8, 2), labels = seq(0, 8, 2)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(axis.text.x = element_text(size = 8,
    colour = "black")) + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
p3 <- ggplot(minerari, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "altre evidenze minerarie",
    subtitle = paste0("Totale evidenze: ", nrow(minerari)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 8)) + scale_y_continuous(breaks = seq(0,
    8, 2), labels = seq(0, 8, 2)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
p4 <- ggplot(smarini, aes(x = dusaf18, fill = dusaf18)) + geom_bar(stat = "count",
    position = "dodge", show.legend = F) + labs(title = "smarini",
    subtitle = paste0("Totale evidenze: ", nrow(smarini)), x = "",
    y = "") + scale_fill_manual(values = veg_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    expand_limits(y = c(0, 10)) + scale_y_continuous(breaks = seq(0,
    10, 2), labels = seq(0, 10, 2)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -2) + geom_text(stat = "count",
    aes(label = paste0("(", round((100 * (..count..))/sum(..count..),
        1), "%)"), vjust = -0.6), size = 4) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
p1 + p2 + p3 + p4 + plot_layout(ncol = 2) + plot_annotation(title = "Copertura Vegetale",
    caption = "Dato copertura del suolo: DUSAF2018 Regione Lombardia - rielaborato E.Croce") &
    theme(plot.title = element_text(size = 20, hjust = 0.5))
dev.off()
```

3.8 Analisi della distribuzione delle evidenze nelle singole aree

Estrazione dei dati delle singole aree in cui è stato suddiviso il territorio.

```
st_layers("D:/Archeologia/Carona/Database/SBC_DB.db")
```

```
Driver: SQLite
Available layers:
              layer_name        geometry_type features
1          Survey_Carona 3D Multi Line String      103
2    Sentieri_IGM1933-39    Multi Line String       90
3       Sentieri_CTR2016    Multi Line String      115
4             Scavi_MABG                Point       27
5            Alpeggi_SBC                Point       34
6  Condotte_Forzate_ENEL          Line String        3
7      Alpeggi_ValFondra                Point      109
8         Viabilit\340_1812          Line String       11
9      Gippabile Moderna          Line String        4
10   Mulattiere_Militari          Line String        3
11  Sentieri_Abbandonati          Line String       12
12          Sentieri_CAI 3D Multi Line String       18
13           Foto_VVN001                Point       33
14 Foto_SaggiMABG_nositi                Point       30
15           Scavi_UNITN                Point       10
16    Foto_Paesaggio_SBC                Point     8172
17         Datazioni_14C                Point       19
18   Toponimi_Catasto_LV                Point      239
19           Aree_Codici              Polygon       15
20         Foto_Siti_SBC                Point     7579
21              Siti_SBC                Point      829
   fields
1       6
2       4
3       4
4       7
5      12
6       3
7       8
8       2
9       2
10      2
11      3
12      6
13     18
14     17
15      7
16     17
17     16
18     11
19      5
20     16
21     43
```

```
Aree_SBC <- st_read("D:/Archeologia/Carona/Database/SBC_DB.db",
    "Aree_Codici")
```

```
Reading layer `Aree_Codici' from data source
  `D:\Archeologia\Carona\Database\SBC_DB.db'
```

```
  using driver `SQLite'
Simple feature collection with 15 features and 5 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 561597.4 ymin: 5094603 xmax: 570487.2 ymax: 5100124
Projected CRS: WGS 84 / UTM zone 32N
```

```
class(Aree_SBC)
```

```
[1] "sf"         "data.frame"
```

```
colnames(Aree_SBC)
```

```
[1] "codice" "alpe"   "nome"   "area"   "mq"     "geom"
```

```
nrow(Aree_SBC)
```

```
[1] 15
```

```
Aree_SBC$codice <- factor(Aree_SBC$codice)
Aree_SBC <- arrange(Aree_SBC, Aree_SBC$codice)
```

Creazione di una funzione per il conteggio dei siti in relazione alle aree.

```
count.area <- function(x) {
    a <- x$area
    kmq <- data.frame(matrix(ncol = 1, nrow = 0))
    colnames(kmq) <- c("kmq")
    for (i in a) {
        b <- data.frame(kmq = subset(Aree_SBC$area, Aree_SBC$codice ==
            i))
        kmq <- rbind(kmq, b)
    }
    return(cbind(x, kmq))
}
```

Creazione del *record* di dati relativi alle evidenze e alle aree.

```
siti_aree_SBC <- count(Siti_SBC, area, factor(tipo))
siti_aree_SBC <- st_set_geometry(siti_aree_SBC, NULL)
colnames(siti_aree_SBC) <- c("area", "tipo", "n")
siti_aree_SBC <- count.area(siti_aree_SBC)
siti_aree_SBC$dens <- round(siti_aree_SBC$n/siti_aree_SBC$kmq,
    2)
siti_aree_SBC <- siti_aree_SBC %>%
    group_by(tipo) %>%
    mutate(perc.tip = n/sum(n) * 100)
siti_aree_SBC <- siti_aree_SBC %>%
    group_by(area) %>%
    mutate(perc.area = n/sum(n) * 100)
siti_aree_SBC$perc.tot <- siti_aree_SBC$n/nrow(Siti_SBC) * 100
sum(siti_aree_SBC$perc.tot)
```

```
[1] 100
```

```
sum(siti_aree_SBC$n)
```

```
[1] 829
```

Resa grafica dei dati generali, relativi a tutte le aree di interesse.

```
setwd("D:/Archeologia/Carona/Analisi/Aree")
```

```
jpeg("TotaleSiti_Area_tot.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(siti_aree_SBC, aes(x = area, y = perc.area, fill = tipo)) +
    geom_col(width = 1000, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "Evidenze antropiche SBC",
    subtitle = paste0("Totale evidenze: ", sum(siti_aree_SBC$n)),
    x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_void() + labs(fill = "Strutture:") + facet_wrap(~area,
    ncol = 5) + theme(plot.margin = unit(c(10, 30, 10, 10), "pt")) +
    theme(plot.title = element_text(margin = margin(b = 10))) +
    theme(plot.subtitle = element_text(margin = margin(b = 12)))
dev.off()

jpeg("TotaleSiti_Area_dens.jpg", width = 6000, height = 4000,
    units = "px", res = 300)
ggplot(siti_aree_SBC, aes(x = area, y = dens, fill = tipo)) +
    geom_col(position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = TRUE) +
    labs(title = "Evidenze antropiche SBC - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(siti_aree_SBC$n),
            " - Area: ", sum(Aree_SBC$area), " kmq"), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    facet_wrap(~area, ncol = 3, scales = "free") + labs(fill = "Strutture:") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()
```

3.8.1 Singole aree

Rese grafiche dei dati relativi alle evidenze antropiche, effettuate separatamente per ogni singola area di pertinenza.

```
setwd("D:/Archeologia/Carona/Analisi/Aree/Singole Aree")

# ARM

jpeg("ARM_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "ARM"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "ARM - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "ARM")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "ARM")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# BDC

jpeg("BDC_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "BDC"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "BDC - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "BDC")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "BDC")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# CAB

jpeg("CAB_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "CAB"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "CAB - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "CAB")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "CAB")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
```

```
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# DDS

jpeg("DDS_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "DDS"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "DDS - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "DDS")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "DDS")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# DSI

jpeg("DSI_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "DSI"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "DSI - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "DSI")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "DSI")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# FPP

jpeg("FPP_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "FPP"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "FPP - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "FPP")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "FPP")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
```

```
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# MAD

jpeg("MAD_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "MAD"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "MAD - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "MAD")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "MAD")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# MRS

jpeg("MRS_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "MRS"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "MRS - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "MRS")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "MRS")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# PAG

jpeg("PAG_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "PAG"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "PAG - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "PAG")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "PAG")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
```

```
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# PDS

jpeg("PDS_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "PDS"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "PDS - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "PDS")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "PDS")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# PRS

jpeg("PRS_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "PRS"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "PRS - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "PRS")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "PRS")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# VDF

jpeg("VDF_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "VDF"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "VDF - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "VDF")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "VDF")$perc.tot), 2), "% del totale"), x = "",
```

```
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# VLN

jpeg("VLN_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "VLN"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "VLN - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "VLN")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "VLN")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# VMS

jpeg("VMS_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "VMS"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "VMS - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "VMS")$n), " - ", round(sum(subset(siti_aree_SBC,
        area == "VMS")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()

# VSB

jpeg("VSB_Area_tot.jpg", width = 2500, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, area == "VSB"), aes(x = area, y = perc.area,
    fill = tipo)) + geom_col(width = 1, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "VSB - numero di evidenze antropiche",
    subtitle = paste0("Totale evidenze: ", sum(subset(siti_aree_SBC,
        area == "VSB")$n), " - ", round(sum(subset(siti_aree_SBC,
```

```
        area == "VSB")$perc.tot), 2), "% del totale"), x = "",
    y = "") + scale_fill_manual(values = type_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_text(aes(x = 1.62, y = perc.area,
    label = ifelse(perc.area >= 2, paste0(round(perc.area, 2),
        "%"), "")), position = position_stack(vjust = 0.5)) +
    geom_shadowtext(aes(x = 1.1, label = ifelse(perc.area >=
        6, paste0(tipo), "")), position = position_stack(vjust = 0.5),
        size = 3.5, bg.colour = "gray94", color = "black")
dev.off()
```

Rese grafiche della distribuzione di ogni tipologia di evidenza nelle aree di interesse. I dati vengono rappresentati sia come numeri assoluti che come densità di evidenze al kilometro quadrato.

```
setwd("D:/Archeologia/Carona/Analisi/Aree/Singoli Tipi")

# Altro

jpeg("Altro_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Altro"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Altro' - numero di evidenze", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Altro")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Altro")$n) + 1)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Altro")$n) + 1, 2), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Altro")$n) + 1, 2)) +
    geom_text(stat = "identity", aes(y = n, label = n), vjust = -1.5) +
    geom_text(stat = "identity", aes(y = n, label = paste0(round(perc.tip,
        2), "%")), vjust = -0.2) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

jpeg("Altro_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Altro"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Altro' - densit\340 per kmq", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Altro")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Altro")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Altro")$dens), 2), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Altro")$dens), 2)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Baita

jpeg("Baita_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
```

```
ggplot(subset(siti_aree_SBC, tipo == "Baita"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Baita' - numero di evidenze", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Baita")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Baita")$n) + 2)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Baita")$n) + 2, 4), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Baita")$n) + 2, 4)) +
    geom_text(stat = "identity", aes(y = n, label = n), vjust = -1.5) +
    geom_text(stat = "identity", aes(y = n, label = paste0(round(perc.tip,
        2), "%")), vjust = -0.2) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

jpeg("Baita_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Baita"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Baita' - densit\340 per kmq", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Baita")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Baita")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Baita")$dens), 4), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Baita")$dens), 4)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Carbonaia

jpeg("Carbonaia_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Carbonaia"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Carbonaia' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Carbonaia")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Carbonaia")$n) + 1)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Carbonaia")$n) + 1, 2), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Carbonaia")$n) + 1, 2)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()
```

```
jpeg("Carbonaia_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Carbonaia"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Carbonaia' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Carbonaia")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Carbonaia")$dens))) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Carbonaia")$dens), 2), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Carbonaia")$dens), 2)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Idroelettrico

jpeg("Idroelettrico_Area_tot.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Idroelettrico"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "darkgoldenrod1",
        show.legend = FALSE) + labs(title = "Categoria: 'Idroelettrico' - numero di evidenze",
    subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
        tipo == "Idroelettrico")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Idroelettrico")$n) + 1)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Idroelettrico")$n) + 1, 2), labels = seq(0,
        max(subset(siti_aree_SBC, tipo == "Idroelettrico")$n) +
            1, 2)) + geom_text(stat = "identity", aes(y = n,
    label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Idroelettrico_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Idroelettrico"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "darkgoldenrod1",
        show.legend = FALSE) + labs(title = "Categoria: 'Idroelettrico' - densit\340 per kmq",
    subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
        tipo == "Idroelettrico")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Idroelettrico")$dens))) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Idroelettrico")$dens), 2), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Idroelettrico")$dens), 2)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
```

```
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Indefinito

jpeg("Indefinito_Area_tot.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Indefinito"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Indefinito' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Indefinito")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Indefinito")$n) + 2)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Indefinito")$n) + 2, 4), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Indefinito")$n) + 2, 4)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Indefinito_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Indefinito"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Indefinito' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Indefinito")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Indefinito")$dens))) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Indefinito")$dens), 4), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Indefinito")$dens), 4)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Militare

jpeg("Militare_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Militare"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Militare' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Militare")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Militare")$n) + 0.5)) +
```

```
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Militare")$n), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Militare")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Militare_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Militare"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Militare' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Militare")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Militare")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Militare")$dens), 1),
    labels = seq(0, max(subset(siti_aree_SBC, tipo == "Militare")$dens),
        1)) + geom_text(stat = "identity", aes(y = dens, label = dens),
    vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Minerario

jpeg("Minerario_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Minerario"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Minerario' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Minerario")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Minerario")$n) + 0.5)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Minerario")$n), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Minerario")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Minerario_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Minerario"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Minerario' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
```

```
            tipo == "Minerario")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Minerario")$dens))) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Minerario")$dens), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Minerario")$dens), 1)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Miniera

jpeg("Miniera_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Miniera"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Miniera' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Miniera")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Miniera")$n) + 1)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Miniera")$n) + 1, 2),
    labels = seq(0, max(subset(siti_aree_SBC, tipo == "Miniera")$n) +
        1, 2)) + geom_text(stat = "identity", aes(y = n, label = n),
    vjust = -1.5) + geom_text(stat = "identity", aes(y = n, label = paste0(round(perc.tip,
    2), "%")), vjust = -0.2) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

jpeg("Miniera_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Miniera"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Miniera' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Miniera")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Miniera")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Miniera")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Miniera")$dens), 1)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Muro

jpeg("Muro_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Muro"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
```

```
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Muro' - numero di evidenze", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Muro")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Muro")$n) + 2)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Muro")$n) + 2, 4), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Muro")$n) + 2, 4)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Muro_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Muro"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Muro' - densit\340 per kmq", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Muro")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Muro")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Muro")$dens), 2), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Muro")$dens), 2)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Ponte

jpeg("Ponte_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Ponte"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Ponte' - numero di evidenze", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Ponte")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Ponte")$n) + 0.2)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Ponte")$n), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Ponte")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Ponte_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Ponte"), aes(x = area)) +
```

```
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Ponte' - densit\340 per kmq", subtitle = paste0("Totale evidenze:",
        sum(subset(siti_aree_SBC, tipo == "Ponte")$n)), x = "",
        y = "") + scale_fill_manual(values = type_colors) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Ponte")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Ponte")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Ponte")$dens), 1)) + geom_text(stat = "identity",
    aes(y = dens, label = dens), vjust = -0.5) + labs(fill = "") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

# Recinto

jpeg("Recinto_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Recinto"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Recinto' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Recinto")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Recinto")$n) + 2)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Recinto")$n) + 2, 4),
    labels = seq(0, max(subset(siti_aree_SBC, tipo == "Recinto")$n) +
        2, 4)) + geom_text(stat = "identity", aes(y = n, label = n),
    vjust = -1.5) + geom_text(stat = "identity", aes(y = n, label = paste0(round(perc.tip,
    2), "%")), vjust = -0.2) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

jpeg("Recinto_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Recinto"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Recinto' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Recinto")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Recinto")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Recinto")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Recinto")$dens), 1)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Reglana

jpeg("Reglana_Area_tot.jpg", width = 3000, height = 2000, units = "px",
```

```
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Reglana"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Reglana' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Reglana")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Reglana")$n) + 0.5)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Reglana")$n), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Reglana")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Reglana_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Reglana"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Reglana' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Reglana")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Reglana")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Reglana")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Reglana")$dens), 1)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Ricovero

jpeg("Ricovero_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Ricovero"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Ricovero' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Ricovero")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Ricovero")$n) + 2)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Ricovero")$n) + 2, 4), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Ricovero")$n) + 2, 4)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
```

```
    10), "pt"))
dev.off()

jpeg("Ricovero_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Ricovero"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Ricovero' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Ricovero")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Ricovero")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Ricovero")$dens), 4),
    labels = seq(0, max(subset(siti_aree_SBC, tipo == "Ricovero")$dens),
        4)) + geom_text(stat = "identity", aes(y = dens, label = dens),
    vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Riparo

jpeg("Riparo_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Riparo"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Riparo' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Riparo")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Riparo")$n) + 2)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Riparo")$n) + 2, 4), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Riparo")$n) + 2, 4)) +
    geom_text(stat = "identity", aes(y = n, label = n), vjust = -1.5) +
    geom_text(stat = "identity", aes(y = n, label = paste0(round(perc.tip,
        2), "%")), vjust = -0.2) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

jpeg("Riparo_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Riparo"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Riparo' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Riparo")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Riparo")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Riparo")$dens), 8), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Riparo")$dens), 8)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
```

```
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Riparo Naturale

jpeg("Riparo Naturale_Area_tot.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Riparo Naturale"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Riparo Naturale' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Riparo Naturale")$n)), x = "", y = "") +
    scale_fill_manual(values = type_colors) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + scale_x_discrete(limits = aree_cod) +
    expand_limits(y = c(0, max(subset(siti_aree_SBC, tipo ==
        "Riparo Naturale")$n) + 1)) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Riparo Naturale")$n) +
        1, 1), labels = seq(0, max(subset(siti_aree_SBC, tipo ==
    "Riparo Naturale")$n) + 1, 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Riparo Naturale_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Riparo Naturale"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Riparo Naturale' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Riparo Naturale")$n)), x = "", y = "") +
    scale_fill_manual(values = type_colors) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + scale_x_discrete(limits = aree_cod) +
    expand_limits(y = c(0, max(subset(siti_aree_SBC, tipo ==
        "Riparo Naturale")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Riparo Naturale")$dens),
    1), labels = seq(0, max(subset(siti_aree_SBC, tipo == "Riparo Naturale")$dens),
    1)) + geom_text(stat = "identity", aes(y = dens, label = dens),
    vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Roccolo

jpeg("Roccolo_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Roccolo"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Roccolo' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Roccolo")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
```

```
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Roccolo")$n) + 0.2)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Roccolo")$n), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Roccolo")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Roccolo_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Roccolo"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Roccolo' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Roccolo")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Roccolo")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Roccolo")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Roccolo")$dens), 1)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Sentieri

jpeg("Sentieri_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Sentieri"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Sentieri' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Sentieri")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Sentieri")$n) + 1)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Sentieri")$n) + 1, 2), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Sentieri")$n) + 1, 2)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Sentieri_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Sentieri"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
```

```
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Sentieri' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Sentieri")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Sentieri")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Sentieri")$dens), 1),
    labels = seq(0, max(subset(siti_aree_SBC, tipo == "Sentieri")$dens),
        1)) + geom_text(stat = "identity", aes(y = dens, label = dens),
    vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Smarino

jpeg("Smarino_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Smarino"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Smarino' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Smarino")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Smarino")$n) + 0.5)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Smarino")$n), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Smarino")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Smarino_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Smarino"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Smarino' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Smarino")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Smarino")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Smarino")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Smarino")$dens), 1)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Spiazzo
```

```
jpeg("Spiazzo_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Spiazzo"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Spiazzo' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Spiazzo")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Spiazzo")$n) + 0.5)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Spiazzo")$n), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Spiazzo")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Spiazzo_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Spiazzo"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Spiazzo' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Spiazzo")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Spiazzo")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Spiazzo")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Spiazzo")$dens), 1)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()

# Stalla

jpeg("Stalla_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Stalla"), aes(x = area)) +
    geom_col(aes(y = n, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Stalla' - numero di evidenze",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Stalla")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Stalla")$n) + 0.2)) +
    scale_y_continuous(breaks = seq(0, max(subset(siti_aree_SBC,
        tipo == "Stalla")$n), 1), labels = seq(0, max(subset(siti_aree_SBC,
        tipo == "Stalla")$n), 1)) + geom_text(stat = "identity",
    aes(y = n, label = n), vjust = -1.5) + geom_text(stat = "identity",
    aes(y = n, label = paste0(round(perc.tip, 2), "%")), vjust = -0.2) +
```

```
    labs(fill = "") + theme(plot.margin = unit(c(10, 30, 10,
    10), "pt"))
dev.off()

jpeg("Stalla_Area_dens.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(subset(siti_aree_SBC, tipo == "Stalla"), aes(x = area)) +
    geom_col(aes(y = dens, fill = tipo), position = position_dodge2(preserve = "single",
        padding = 0), width = 0.75, color = "black", show.legend = FALSE) +
    labs(title = "Categoria: 'Stalla' - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(subset(siti_aree_SBC,
            tipo == "Stalla")$n)), x = "", y = "") + scale_fill_manual(values = type_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + expand_limits(y = c(0,
    max(subset(siti_aree_SBC, tipo == "Stalla")$dens))) + scale_y_continuous(breaks = seq(0,
    max(subset(siti_aree_SBC, tipo == "Stalla")$dens), 1), labels = seq(0,
    max(subset(siti_aree_SBC, tipo == "Stalla")$dens), 1)) +
    geom_text(stat = "identity", aes(y = dens, label = dens),
        vjust = -0.5) + labs(fill = "") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
dev.off()
```

3.8.2 Evidenze produttive

Analisi della distribuzione nelle singole aree delle evidenze raggruppate secondo la loro destinazione produttiva: pastorale, mineraria, carbonaia.

```
# Strutture pastorali
Str.past <- subset(Siti_SBC, Siti_SBC$tipo == "Recinto" | Siti_SBC$tipo ==
    "Ricovero" | Siti_SBC$tipo == "Spiazzo" | Siti_SBC$tipo ==
    "Stalla")
Str.past <- count(Str.past, area)
Str.past <- st_set_geometry(Str.past, NULL)
Str.past$tipo <- "pastorali"
Str.past <- count.area(Str.past)
Str.past$dens <- round(Str.past$n/Str.past$kmq, 2)
Str.past <- Str.past %>%
    group_by(tipo) %>%
    mutate(perc.tip = n/sum(n) * 100)
sum(Str.past$perc.tip)
```

```
[1] 100
```

```
sum(Str.past$n)
```

```
[1] 139
```

```
# Strutture minerarie
Str.mine <- subset(Siti_SBC, Siti_SBC$tipo == "Minerario" | Siti_SBC$tipo ==
    "Miniera" | Siti_SBC$tipo == "Reglana" | Siti_SBC$tipo ==
    "Smarino")
Str.mine <- count(Str.mine, area)
Str.mine <- st_set_geometry(Str.mine, NULL)
Str.mine$tipo <- "minerarie"
Str.mine <- count.area(Str.mine)
Str.mine$dens <- round(Str.mine$n/Str.mine$kmq, 2)
Str.mine <- Str.mine %>%
    group_by(tipo) %>%
```

```
    mutate(perc.tip = n/sum(n) * 100)
sum(Str.mine$perc.tip)
```

```
[1] 100
```

```
sum(Str.mine$n)
```

```
[1] 73
```

```
# Carbonaie
Str.carb <- subset(Siti_SBC, Siti_SBC$tipo == "Carbonaia")
Str.carb <- count(Str.carb, area)
Str.carb <- st_set_geometry(Str.carb, NULL)
Str.carb$tipo <- "carbonaie"
Str.carb <- count.area(Str.carb)
Str.carb$dens <- round(Str.carb$n/Str.carb$kmq, 2)
Str.carb <- Str.carb %>%
    group_by(tipo) %>%
    mutate(perc.tip = n/sum(n) * 100)
sum(Str.carb$perc.tip)
```

```
[1] 100
```

```
sum(Str.carb$n)
```

```
[1] 46
```

```
# Unione dei dati
prod_SBC <- rbind(Str.past, Str.mine, Str.carb)
prod_SBC <- prod_SBC %>%
    group_by(area) %>%
    mutate(perc.area = n/sum(n) * 100)
```

Resa grafica dei dati, sia come numeri assoluti che come densità per kmq.

```
jpeg("Produzione_Area_tot.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(prod_SBC, aes(x = area, y = perc.area, fill = tipo)) +
    geom_col(width = 1000, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "Strutture produttive",
    subtitle = paste0("Totale evidenze: ", paste0(sum(prod_SBC$n),
        " - ", round((sum(prod_SBC$n))/nrow(Siti_SBC) * 100,
            2), "% delle evidenze totali"), x = "", y = "")) +
    scale_fill_manual(values = prod_colors) + theme_void() +
    labs(fill = "Strutture:") + geom_shadowtext(aes(x = 1.2,
    y = perc.area, label = paste0(round(perc.area, 2), "%")),
    position = position_stack(vjust = 0.5), size = 3, bg.colour = "gray94",
    color = "black") + facet_wrap(~area, ncol = 5) + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt")) + theme(legend.position = c(0.92, 0.15)) +
    theme(plot.title = element_text(margin = margin(b = 10))) +
    theme(plot.subtitle = element_text(margin = margin(b = 12)))
dev.off()

jpeg("Produzione_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(prod_SBC, aes(x = area, y = dens, fill = tipo)) + geom_col(position = position_dodge2(preserve
    padding = 0), width = 0.8, color = "black", show.legend = TRUE) +
    labs(title = "Strutture produttive - densita per kmq", subtitle = paste0("Totale evidenze: ",
        sum(prod_SBC$n), " - Area: ", sum(Aree_SBC$area), " kmq"),
        x = "", y = "") + scale_fill_manual(values = prod_colors) +
```

```
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + labs(fill = "Strutture:") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()

jpeg("Produzione_Area_dens2.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(prod_SBC, aes(x = area, y = dens, fill = tipo)) + geom_col(position = position_dodge2(preserve
    padding = 0), width = 0.75, color = "black", show.legend = TRUE) +
    labs(title = "Strutture produttive SBC - densit\340 per kmq",
        subtitle = paste0("Totale evidenze:", sum(prod_SBC$n)),
        x = "", y = "") + scale_fill_manual(values = prod_colors) +
    theme_bw() + theme(axis.text = element_text(size = 10, colour = "black")) +
    facet_wrap(~area, ncol = 3, scales = "free") + labs(fill = "Strutture:") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt")) + theme(legend.position = c(0.85,
    0.06))
dev.off()
```

3.8.3 Evidenze di protezione

Per evidenze protettive si intendono le strutture che permettono la protezione degli umani dagli eventi atmosferici: baite, ricoveri, ripari.

```
prote_SBC <- subset(Siti_SBC, Siti_SBC$tipo == "Baita" | Siti_SBC$tipo ==
    "Ricovero" | Siti_SBC$tipo == "Riparo")
prote_SBC <- count(prote_SBC, area, tipo)
prote_SBC <- st_set_geometry(prote_SBC, NULL)
prote_SBC <- count.area(prote_SBC)
prote_SBC$dens <- round(prote_SBC$n/prote_SBC$kmq, 2)
prote_SBC$perc <- prote_SBC$n/sum(prote_SBC$n) * 100
prote_SBC <- prote_SBC %>%
    group_by(tipo) %>%
    mutate(perc.tip = n/sum(n) * 100)
prote_SBC <- prote_SBC %>%
    group_by(area) %>%
    mutate(perc.area = n/sum(n) * 100)
sum(prote_SBC$perc)
```

```
[1] 100
```

```
sum(prote_SBC$n)
```

```
[1] 355
```

Resa grafica.

```
jpeg("Protezioni_Area_tot.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(prote_SBC, aes(x = area, y = perc.area, fill = tipo)) +
    geom_col(width = 1000, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "Strutture di protezione",
    subtitle = paste0("Totale evidenze: ", paste0(sum(prote_SBC$n),
        " - ", round((sum(prote_SBC$n))/nrow(Siti_SBC) * 100,
            2), "% delle evidenze totali"), x = "", y = "")) +
    scale_fill_manual(values = type_colors, breaks = c("Baita",
        "Ricovero", "Riparo")) + theme_void() + labs(fill = "Strutture:") +
    geom_shadowtext(aes(x = 1.2, y = perc.area, label = paste0(round(perc.area,
        2), "%")), position = position_stack(vjust = 0.5), size = 3,
        bg.colour = "gray94", color = "black") + facet_wrap(~area,
```

```
    ncol = 5) + theme(legend.position = "bottom") + theme(plot.margin = unit(c(10,
    10, 10, 10), "pt")) + theme(plot.title = element_text(margin = margin(b = 10))) +
    theme(plot.subtitle = element_text(margin = margin(b = 12)))
dev.off()

jpeg("Protezioni_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(prote_SBC, aes(x = area, y = dens, fill = tipo)) + geom_col(position = position_dodge2(preserv
    padding = 0), width = 0.75, color = "black", show.legend = TRUE) +
    labs(title = "Strutture di protezione - densit\340 al kmq",
        subtitle = paste0("Totale evidenze: ", paste0(sum(prote_SBC$n),
            " - ", round((sum(prote_SBC$n))/nrow(Siti_SBC) *
                100, 2), "% delle evidenze totali"), x = "",
            y = "")) + scale_fill_manual(values = type_colors,
    breaks = c("Baita", "Ricovero", "Riparo")) + theme_bw() +
    theme(axis.text = element_text(size = 10, colour = "black")) +
    scale_x_discrete(limits = aree_cod) + labs(fill = "Strutture:") +
    theme(plot.margin = unit(c(10, 30, 10, 10), "pt"))
dev.off()
```

3.8.4 Evidenze pastorali

Analisi delle strutture pastorali, scorporate nelle quattro tipologie di evidenze: recinto, ricovero, spiazzo, stalla.

```
pastor_SBC <- subset(Siti_SBC, Siti_SBC$tipo == "Recinto" | Siti_SBC$tipo ==
    "Ricovero" | Siti_SBC$tipo == "Spiazzo" | Siti_SBC$tipo ==
    "Stalla")
pastor_SBC <- count(pastor_SBC, area, tipo)
pastor_SBC <- st_set_geometry(pastor_SBC, NULL)
pastor_SBC <- count.area(pastor_SBC)
pastor_SBC$dens <- round(pastor_SBC$n/pastor_SBC$kmq, 2)
pastor_SBC$perc <- pastor_SBC$n/sum(pastor_SBC$n) * 100
pastor_SBC <- pastor_SBC %>%
    group_by(tipo) %>%
    mutate(perc.tip = n/sum(n) * 100)
pastor_SBC <- pastor_SBC %>%
    group_by(area) %>%
    mutate(perc.area = n/sum(n) * 100)
sum(pastor_SBC$perc)
```

```
[1] 100
```

```
sum(pastor_SBC$n)
```

```
[1] 139
```

Resa grafica dei dati, sia come numeri assoluti che come densità di evidenze al kilometro quadrato.

```
jpeg("Pastorali_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(pastor_SBC, aes(x = area, y = perc.area, fill = tipo)) +
    geom_col(width = 1000, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "Strutture pastorali",
    subtitle = paste0("Totale evidenze: ", paste0(sum(pastor_SBC$n),
        " - ", round((sum(pastor_SBC$n))/nrow(Siti_SBC) * 100,
            2), "% delle evidenze totali"), x = "", y = "")) +
    scale_fill_manual(values = type_colors, breaks = c("Recinto",
        "Ricovero", "Spiazzo", "Stalla")) + theme_void() + labs(fill = "Strutture:") +
    geom_shadowtext(aes(x = 1.2, y = perc.area, label = paste0(round(perc.area,
        2), "%")), position = position_stack(vjust = 0.5), size = 3,
        bg.colour = "gray94", color = "black") + facet_wrap(~area,
    ncol = 5) + theme(legend.position = "bottom") + theme(plot.margin = unit(c(10,
    10, 10, 10), "pt")) + theme(legend.position = c(0.92, 0.15)) +
    theme(plot.title = element_text(margin = margin(b = 10))) +
    theme(plot.subtitle = element_text(margin = margin(b = 12)))
dev.off()

jpeg("Pastorali_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(pastor_SBC, aes(x = area, y = dens, fill = tipo)) + geom_col(position = position_dodge2(preser
    padding = 0), width = 0.75, color = "black", show.legend = TRUE) +
    labs(title = "Strutture pastorali - densit\340 al kmq", subtitle = paste0("Totale evidenze: ",
        paste0(sum(pastor_SBC$n), " - ", round((sum(pastor_SBC$n))/nrow(Siti_SBC) *
            100, 2), "% delle evidenze totali"), x = "", y = "")) +
    scale_fill_manual(values = type_colors, breaks = c("Recinto",
        "Ricovero", "Spiazzo", "Stalla")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + scale_x_discrete(limits = aree_cod) +
    labs(fill = "Strutture:") + theme(plot.margin = unit(c(10,
    30, 10, 10), "pt"))
```

```
dev.off()
```

3.8.5 Evidenze minerarie

Analisi delle strutture pastorali, scorporate nelle quattro tipologie di evidenze: miniera, minerario, reglana, smarino.

```
mineral_SBC <- subset(Siti_SBC, Siti_SBC$tipo == "Minerario" |
    Siti_SBC$tipo == "Miniera" | Siti_SBC$tipo == "Reglana" |
    Siti_SBC$tipo == "Smarino")
mineral_SBC <- count(mineral_SBC, area, tipo)
mineral_SBC <- st_set_geometry(mineral_SBC, NULL)
mineral_SBC <- count.area(mineral_SBC)
mineral_SBC$dens <- round(mineral_SBC$n/mineral_SBC$kmq, 2)
mineral_SBC$perc <- mineral_SBC$n/sum(mineral_SBC$n) * 100
mineral_SBC <- mineral_SBC %>%
    group_by(tipo) %>%
    mutate(perc.tip = n/sum(n) * 100)
mineral_SBC <- mineral_SBC %>%
    group_by(area) %>%
    mutate(perc.area = n/sum(n) * 100)
sum(mineral_SBC$perc)
```

```
[1] 100
```

```
sum(mineral_SBC$n)
```

```
[1] 73
```

Resa grafica dei dati, sia come numeri assoluti che come densità di evidenze al kilometro quadrato.

```
jpeg("Minerarie_Area_tot.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(mineral_SBC, aes(x = area, y = perc.area, fill = tipo)) +
    geom_col(width = 1000, color = "black", show.legend = TRUE) +
    coord_polar(theta = "y", direction = -1) + labs(title = "Strutture minerarie",
    subtitle = paste0("Totale evidenze: ", paste0(sum(mineral_SBC$n),
        " - ", round((sum(mineral_SBC$n))/nrow(Siti_SBC) * 100,
            2), "% delle evidenze totali"), x = "", y = "")) +
    scale_fill_manual(values = type_colors, breaks = c("Minerario",
        "Miniera", "Reglana", "Smarino")) + theme_void() + labs(fill = "Strutture:") +
    geom_shadowtext(aes(x = 1.2, y = perc.area, label = paste0(round(perc.area,
        2), "%")), position = position_stack(vjust = 0.5), size = 3,
        bg.colour = "gray94", color = "black") + facet_wrap(~area,
    ncol = 4) + theme(legend.position = "bottom") + theme(plot.margin = unit(c(10,
    10, 10, 10), "pt")) + theme(plot.title = element_text(margin = margin(b = 10))) +
    theme(plot.subtitle = element_text(margin = margin(b = 12)))
dev.off()

jpeg("Minerarie_Area_dens.jpg", width = 3000, height = 2000,
    units = "px", res = 300)
ggplot(mineral_SBC, aes(x = area, y = dens, fill = tipo)) + geom_col(position = position_dodge2(prese
    padding = 0), width = 0.75, color = "black", show.legend = TRUE) +
    labs(title = "Strutture minerarie - densit\340 al kmq", subtitle = paste0("Totale evidenze: ",
        paste0(sum(mineral_SBC$n), " - ", round((sum(mineral_SBC$n))/nrow(Siti_SBC) *
            100, 2), "% delle evidenze totali"), x = "", y = "")) +
    scale_fill_manual(values = type_colors, breaks = c("Minerario",
        "Miniera", "Reglana", "Smarino")) + theme_bw() + theme(axis.text = element_text(size = 10,
    colour = "black")) + scale_x_discrete(limits = aree_cod) +
    labs(fill = "Strutture:") + theme(plot.margin = unit(c(10,
```

```
    30, 10, 10), "pt"))
dev.off()
```

3.9 Dati relativi agli allevamenti

Rappresentazione grafica dei dati storici relativi agli allevamenti bovini ed ovini in provincia di Bergamo.

```
setwd("D:/Archeologia/Carona/Analisi/Alpeggi")

armentiBG <- read.csv("D:/Archeologia/Carona/Analisi/Alpeggi/Animali_Allevati.csv")
class(armentiBG)
```

```
[1] "data.frame"
```

```
capi_tipoBG <- pivot_wider(armentiBG, names_from = "Tipo", values_from = "Capi")
```

Resa grafica dei dati.

```
jpeg("Armenti_BG.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(capi_tipoBG, aes(x = Anno)) + geom_line(aes(y = Bovini,
    color = "Bovini"), size = 1.2) + geom_line(aes(y = Ovicaprini,
    color = "Ovicaprini"), size = 1.2) + labs(title = "Allevamenti bovini ed ovini",
    subtitle = "Provincia di Bergamo", x = "anno", y = "capi") +
    scale_y_continuous(limits = c(0, 150000), labels = label_number(suffix = " K",
        scale = 0.001)) + scale_x_continuous(limits = c(1620,
    2030), breaks = seq(1600, 2020, 50), labels = seq(1600, 2020,
    50)) + scale_colour_manual("", values = c(Bovini = "saddlebrown",
    Ovicaprini = "goldenrod1")) + theme(legend.position = "bottom") +
    theme(plot.margin = unit(c(10, 30, 10, 30), "pt"))
dev.off()

jpeg("Armenti_BG_bar.jpg", width = 3000, height = 2000, units = "px",
    res = 300)
ggplot(armentiBG, aes(x = factor(Anno), y = Capi, fill = factor(Tipo,
    levels = c("Bovini", "Ovicaprini")))) + geom_bar(stat = "identity",
    position = "dodge", color = "gray40") + labs(title = "Allevamenti bovini ed ovini",
    subtitle = "Provincia di Bergamo", x = "", y = "") + labs(fill = "Tipo") +
    theme(plot.caption = element_text(hjust = 0.5)) + scale_fill_manual(values = c("burlywood4",
    "palegoldenrod")) + scale_y_continuous(labels = label_number(suffix = " K",
    scale = 0.001)) + theme(plot.margin = unit(c(10, 30, 10,
    30), "pt"))
dev.off()
```

Carico degli alpeggi bergamaschi nel corso del XX secolo d.C.
Fonte: CORTI M., 2004, Süssura de l aalp. Il sistema dell'alpeggio nelle Alpi lombarde, *Annali di San Michele*, 17, pp. 31-155.

```
anno <- c("1900", "1970", "2000", "1900", "1970", "2000", "1900",
    "1970", "2000")
tipo <- c("Bovini da latte", "Bovini da latte", "Bovini da latte",
    "Bovini asciutti", "Bovini asciutti", "Bovini asciutti",
    "Ovini", "Ovini", "Ovini")
capi <- c(12093, 4249, 3407, 11075, 8014, 5135, 3513, 18246,
    26572)
carico_BG <- data.frame(cbind(anno, tipo, capi))
class(carico_BG$capi)
```

```
[1] "character"
```

```
carico_BG$capi <- as.numeric(as.character(carico_BG$capi))
class(carico_BG$capi)
```

```
[1] "numeric"
```

```
caricowideBG <- pivot_wider(carico_BG, names_from = "tipo", values_from = "capi")
colnames(caricowideBG) <- c("Anno", "Latte", "Carne", "Ovini")
caricowideBG$Anno <- as.numeric(caricowideBG$Anno)
```

Resa grafica dei dati relativi agli alpeggi bergamaschi.

```
jpeg("Carico_Alpi_BG.jpg", width = 2000, height = 2000, units = "px",
    res = 300)
ggplot(caricowideBG, aes(x = Anno, group = 1)) + geom_line(aes(y = Latte,
    color = "Bovini da latte"), size = 1.2) + geom_line(aes(y = Carne,
    color = "Bovini asciutti"), size = 1.2) + geom_line(aes(y = Ovini,
    color = "Ovini"), size = 1.2) + labs(title = "Totale capi caricati in alpeggio",
    subtitle = "Provincia di Bergamo", caption = "Origine dati: CORTI M., 2004,
       S\374ssura de l aalp. Il sistema dell'alpeggio nelle Alpi lombarde,
       Atti di SPeA7 2002, Annali di San Michele, Trento, pp. 31-155.",
    x = "anno", y = "capi") + theme(plot.caption = element_text(hjust = 0.5)) +
    scale_colour_manual("", values = c(`Bovini da latte` = "saddlebrown",
        `Bovini asciutti` = "salmon3", Ovini = "goldenrod1")) +
    scale_y_continuous(limits = c(0, 30000), labels = label_number(suffix = " K",
        scale = 0.001)) + scale_x_continuous(labels = c(1900,
    1970, 2000), breaks = c(1900, 1970, 2000)) + theme(legend.position = "bottom") +
    theme(plot.margin = unit(c(10, 10, 10, 10), "pt"))
dev.off()

jpeg("Carico_Alpi_BG_bar.jpg", width = 2000, height = 2000, units = "px",
    res = 300)
ggplot(carico_BG, aes(x = anno, y = capi, fill = factor(tipo,
    levels = c("Bovini da latte", "Bovini asciutti", "Ovini")))) +
    geom_bar(stat = "identity", position = "dodge", color = "gray40") +
    labs(title = "Totale capi caricati in alpeggio", subtitle = "prov. Bergamo",
        x = "", y = "") + labs(fill = "Bestiame", caption = "Origine dati: CORTI M., 2004,
  Sussura de l aalp.
  Il sistema dell'alpeggio nelle Alpi lombarde,
       Atti di SPeA7 2002, Annali di San Michele, Trento, pp. 31-155.") +
    geom_text(stat = "identity", position = position_dodge(0.95),
        aes(y = capi, label = capi), vjust = -0.5) + theme(plot.caption = element_text(hjust = 0.5)) +
    scale_fill_manual(values = c("saddlebrown", "salmon3", "goldenrod1"))
dev.off()
```