UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

DATA SCIENCE E MACHINE LEARNING

# PROGETTO DI STATISTICA E ANALISI DEI DATI

**Violenza di genere durante la pandemia: aggravamento o attenuazione dei numeri?**

DOCENTE

Prof. Amelia Giuseppina Nobile

STUDENTI

Maria Natale, matricola: 0522500967

Gaetano Casillo, matricola: 0522501057

ANNO ACCADEMICO 2020-2021

# SOMMARIO

# 1 INTRODUZIONE

Si sente spesso parlare di violenza di genere, ma che cosa vuol dire? Le Nazioni Unite hanno definito la violenza di genere come "*ogni atto legato alla differenza di sesso che provochi o possa provocare un danno fisico, sessuale, psicologico o una sofferenza della donna, compresa la minaccia di tali atti, la coercizione o l'arbitraria privazione della libertà sia nella vita pubblica che nella vita privata*". Per supportare le vittime delle violenze di genere è stato attivato il numero verde 1522 attivo 24 ore su 24 offrendo accoglienza in italiano, inglese, francese, spagnolo e arabo.

## 1.1 CASO DI STUDIO

Nel 2020 è stato vissuto il lockdown per 3 mesi, in questo periodo molto si è parlato del lato economico, della scuola, ma poco si è discusso del lato sociale di questo evento. Si è pensato pertanto di analizzare le chiamate e i messaggi effettuate al 1522, numero verde contro lo stalking e la violenza, confrontandole con lo stesso periodo (marzo/giugno) degli anni precedenti. In particolare, nell'analisi statistica effettuata si fa riferimento agli utenti e alle vittime per regione di provenienza e anno. Secondo quanto rilevato dall'Istat, che ha analizzato i dati messi a disposizione dal numero antiviolenza 1522, tra marzo e giugno 2020 le telefonate e le comunicazioni via chat con il centralino sono più che raddoppiate rispetto allo stesso periodo dell'anno precedente.

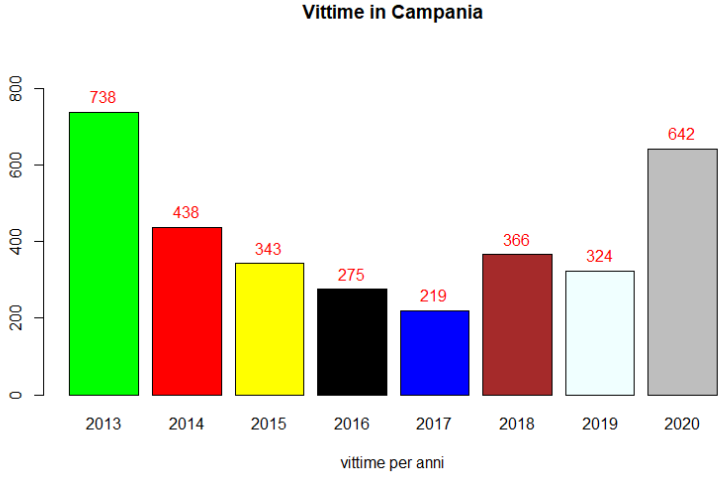
Per l'analisi del fenomeno in esame si considerano i dati relativi alle vittime del numero antiviolenza 1522 effettuate nei mesi di marzo-giugno suddivisi per regione ed anno (2013-2020). In particolare, nell'analisi statistica univariata, verranno esaminate nei dettagli le curve relativi ai dati della regione Campania e la media delle chiamate degli utenti e delle vittime effettuate sull'intero territorio nazionale.

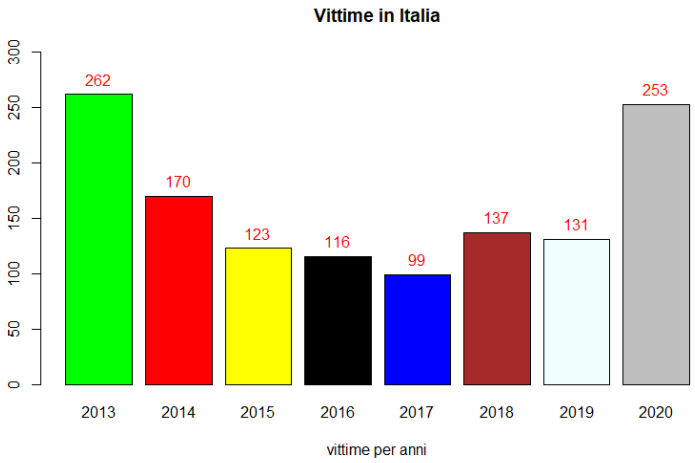
Nella seguente tabella vengono mostrati i dati relativi alle vittime del numero 1522 suddivisi per regione ed anno.

| | Regioni | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Regioni | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 |
| 2 | Piemonte | 406 | 310 | 222 | 202 | 225 | 232 | 248 | 431 |
| 3 | Valle d'Ao | 13 | 8 | 7 | 4 | 3 | 2 | 4 | 8 |
| 4 | Liguria | 152 | 100 | 82 | 69 | 58 | 80 | 71 | 134 |
| 5 | Lombardia | 847 | 550 | 435 | 427 | 413 | 475 | 450 | 990 |
| 6 | Trentino-A | 40 | 23 | 19 | 25 | 9 | 23 | 35 | 50 |
| 7 | Trento | 29 | 20 | 15 | 18 | 8 | 19 | 29 | 35 |
| 8 | Bolzano | 10 | 3 | 4 | 7 | 1 | 4 | 6 | 13 |
| 9 | Veneto | 395 | 283 | 179 | 151 | 134 | 219 | 229 | 396 |
| 10 | Friuli-Ven | 83 | 48 | 41 | 30 | 28 | 50 | 39 | 60 |
| 11 | Emilia-Ro | 377 | 248 | 162 | 161 | 118 | 191 | 171 | 377 |
| 12 | Toscana | 340 | 219 | 135 | 122 | 140 | 187 | 218 | 385 |
| 13 | Umbria | 98 | 40 | 41 | 29 | 28 | 43 | 38 | 64 |
| 14 | Marche | 146 | 76 | 84 | 75 | 53 | 59 | 60 | 127 |
| 15 | Lazio | 724 | 559 | 364 | 380 | 298 | 430 | 422 | 759 |
| 16 | Abruzzo | 136 | 69 | 70 | 57 | 52 | 73 | 63 | 116 |
| 17 | Molise | 19 | 13 | 15 | 15 | 8 | 6 | 10 | 28 |
| 18 | Campania | 738 | 438 | 343 | 275 | 219 | 366 | 324 | 642 |
| 19 | Puglia | 441 | 268 | 164 | 161 | 108 | 213 | 142 | 344 |
| 20 | Basilicata | 36 | 26 | 21 | 13 | 14 | 13 | 26 | 33 |
| 21 | Calabria | 106 | 62 | 43 | 39 | 28 | 42 | 47 | 80 |
| 22 | Sicilia | 438 | 254 | 171 | 200 | 155 | 217 | 194 | 346 |
| 23 | Sardegna | 199 | 115 | 91 | 84 | 73 | 60 | 57 | 148 |

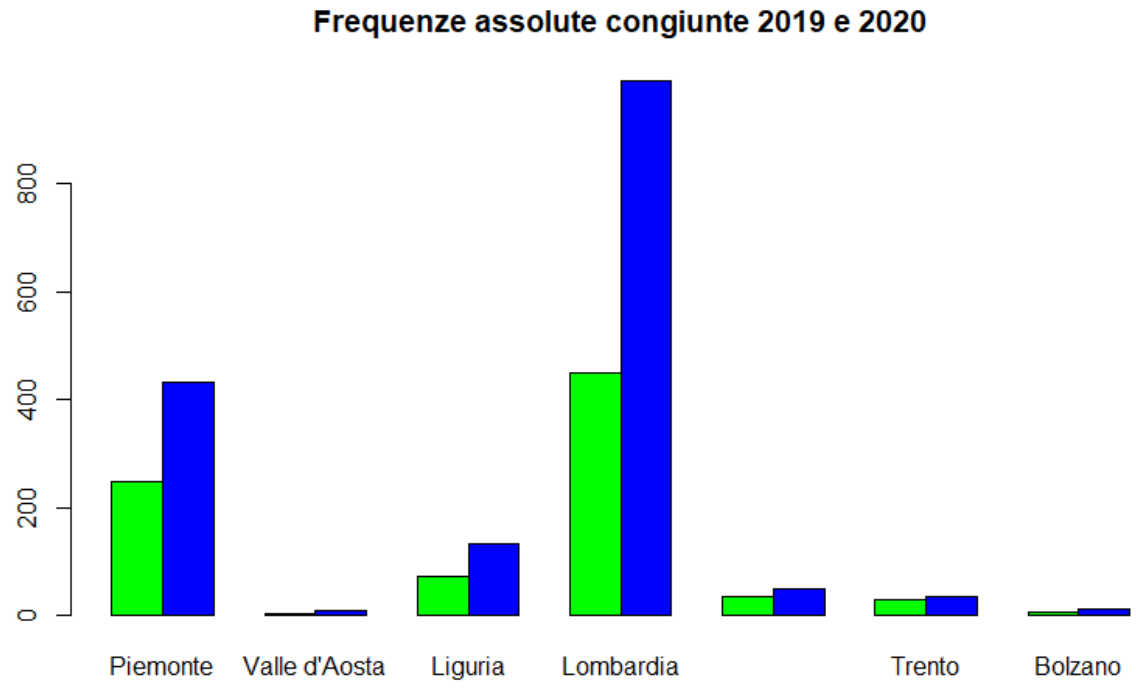
## 1.2 RAPPRESENTAZIONE GRAFICA

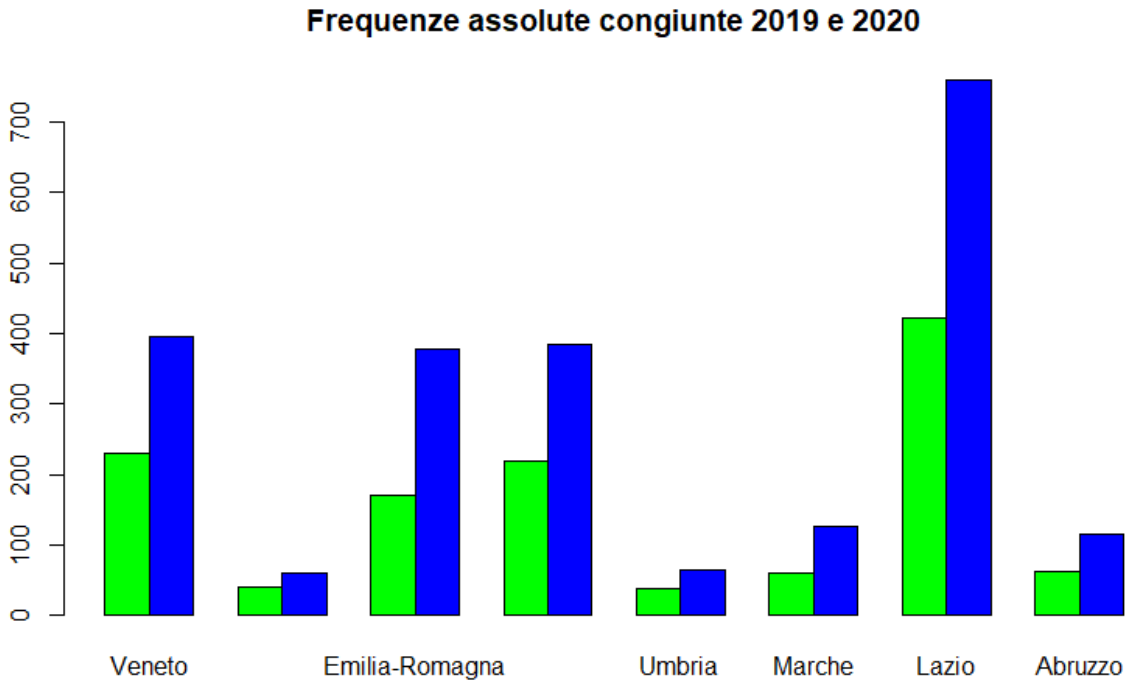
Di seguito vengono mostrati i due barplot relativi ai dati della Campania e della media sull'intero territorio nazionale per quanto riguarda la tabella Vittime. In entrambi i casi si può notare che la modalità a cui è associata la frequenza più alta è il 2013, ma nel 2020, dopo diversi anni di decrescita del numero di vittime, si è avuto un aumento pari al doppio delle vittime del 2019.

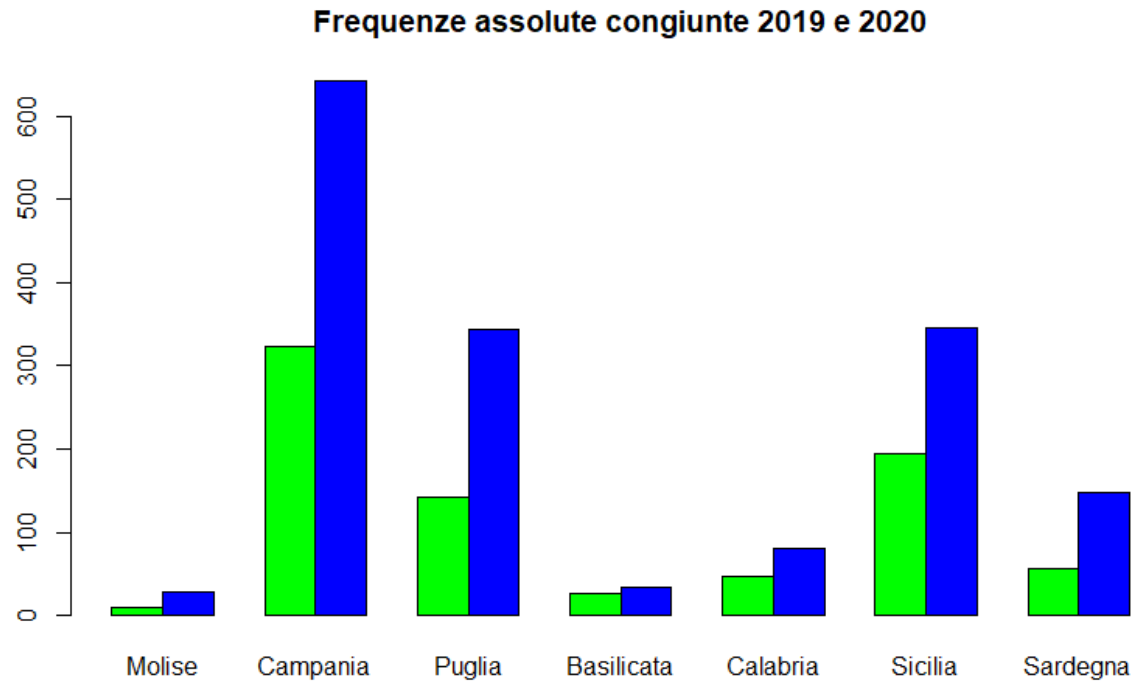





Nei seguenti grafici vengono mostrate le frequenze assolute delle chiamate effettuate nelle varie regioni, mostrando in verde quelle effettuate nel 2019 e in blu quelle effettuate nel 2020. In percentuale si è avuto un aumento medio del 90.28%. Per alcune regioni, tuttavia si è avuto un aumento ancora più significativo, ad esempio la Lombardia nel 2020 ha visto aumentare il numero di chiamate del 120% rispetto all'anno precedente. Altre regioni, ad esempio, la Toscana hanno visto un aumento meno significativo rispetto alle altre regioni, essa infatti ha registrato un aumento del 76.6%. La Campania, invece, con un aumento del 98.14% non è troppo lontana dall'aumento medio.




Frequenze assolute congiunte 2019 e 2020
0
100
200
300
400
500
600
700
Veneto
Emilia-Romagna
Umbria
Marche
Lazio
Abruzzo




Il **diagramma di Pareto** è utile per analizzare un insieme di dati in modo da determinare le poche variabili che influenzano in modo significativo i risultati finali. Il diagramma è composto da barre che indicano l'incidenza percentuale sul fenomeno in esame dei singoli elementi. Le barre più alte corrispondono agli elementi che incidono maggiormente sul fenomeno. Nel diagramma di Pareto è inoltre presente una linea che rappresenta le incidenze degli elementi sommate l'una all'altra.

Questo è il codice usato per creare il grafo:

```
tableNaz<-table(c(rep("2013", mediavittimeitalia[1]),
rep("2014",mediavittimeitalia[2]), rep("2015",mediavittimeitalia[3]),
                  rep("2016",mediavittimeitalia[4]),
rep("2017",mediavittimeitalia[5]), rep("2018",mediavittimeitalia[6]),
                  rep("2019",mediavittimeitalia[7]),
rep("2020",mediavittimeitalia[8])))
ord<-sort(tableNaz, decreasing = TRUE)
propOrd <- prop.table (ord)
x <- barplot (propOrd , ylim = c(0, 1.05) , main = "Diagramma di Pareto
Italia", col =1:8 , las =2)
lines (x, cumsum ( propOrd ), type = "b", pch = 16)
text(x - 0.2, cumsum (propOrd) + 0.03 , paste (format ( cumsum ( propOrd
) * 100, digits = 2) , "%"))
```

Considerando gli ultimi 8 anni, il diagramma di Pareto mostra che il solo anno 2013 incide per il 20% sul totale delle chiamate registrate sulla media nazionale. Si tratta di una percentuale abbastanza alta in quanto un numero equo di chiamate per anno corrisponderebbe al 12.5%.

È da notare anche quasi a parità, segue per altro 20% il 2020, questo significa che nel 2020 si è avuto un numero di casi pari al 2013, anno di massima.

Per la Campania la situazione cambia di poco, il 2013 è sempre maggiore del 20%, 22% per esattezza, mentre è a seguire il 2020 con il 19% delle chiamate totali.

Confrontando i due diagrammi di Pareto ottenuti si nota, innanzitutto che nell'anno 2013 si è il maggior numero dei casi di violenza denunciati rispetto ad altri anni. Inoltre, effettuando un'analisi più approfondita è possibile notare che c'è una leggera differenza tra l'incidenza degli anni 2019 e 2015 per la Campania e la media nazionale. In Campania l'anno 2015 risulta incidere maggiormente rispetto all'anno 2019 mentre per la media nazionale si nota che l'anno 2019 incide maggiormente rispetto al 2015. Tuttavia, si tratta di una differenza minima.

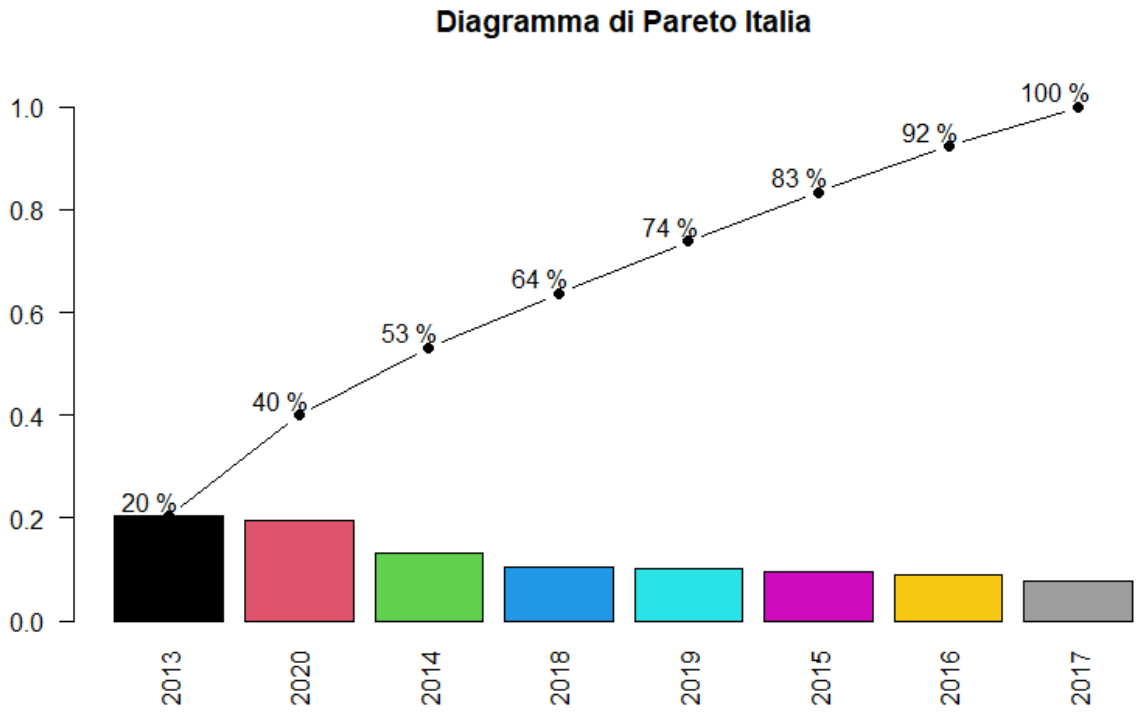


Diagramma di Pareto Italia
1.0
0.8
0.6
0.4
0.2
0.0
20 %
40 %
53 %
64 %
74 %
83 %
92 %
100 %
2013
2020
2014
2018
2019
2015
2016
2017

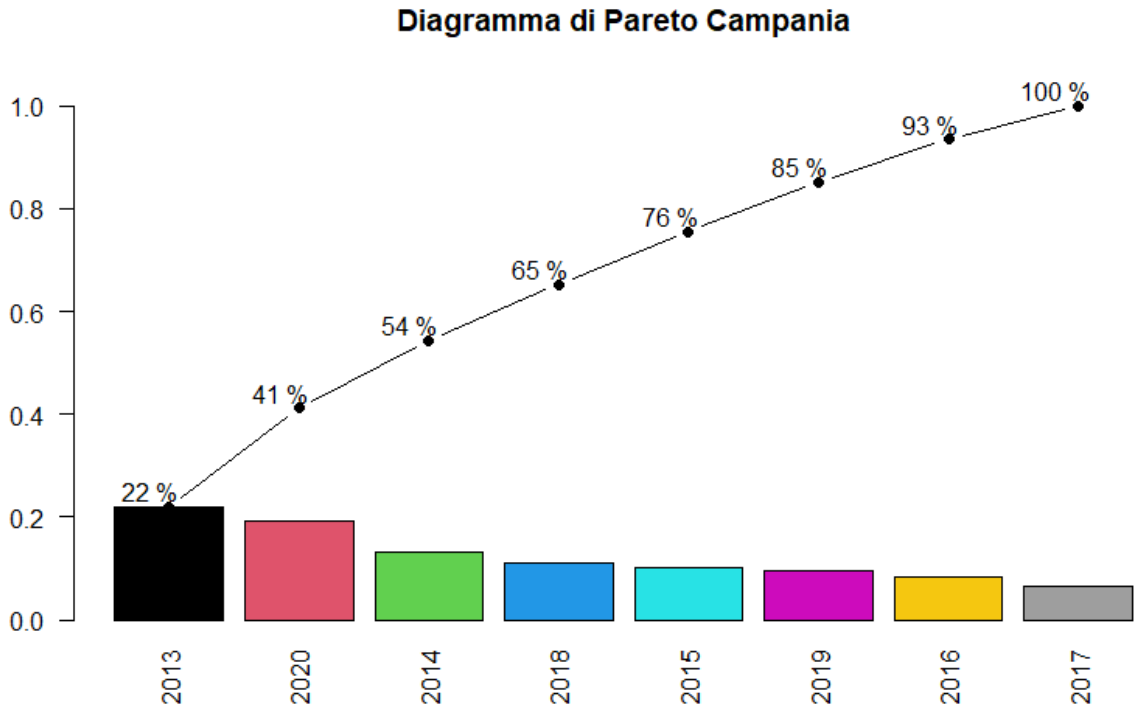


Diagramma di Pareto Campania
1.0
0.8
0.6
0.4
0.2
0.0
22 %
41 %
54 %
65 %
76 %
85 %
93 %
100 %
2013
2020
2014
2018
2015
2019
2016
2017

# 2 STATISTICA DESCRITTIVA UNIVARIATA

In questo capitolo verranno mostrati i risultati relativi all'analisi statistica univariata. In particolare, verrà mostrata la funzione di distribuzione empirica continua, i valori degli indici di sintesi, gli indici di dispersione. Infine, verrà analizzata la forma della distribuzione di frequenze attraverso il calcolo della skewness campionaria e della curtosi campionaria. Le varie analisi verranno effettuate prendendo in esame i dati della Campania e della media nazionale negli anni 2013-2020, analizzando le tabelle Vittime.

## 2.1 FUNZIONE DI DISTRIBUZIONE EMPIRICA CONTINUA

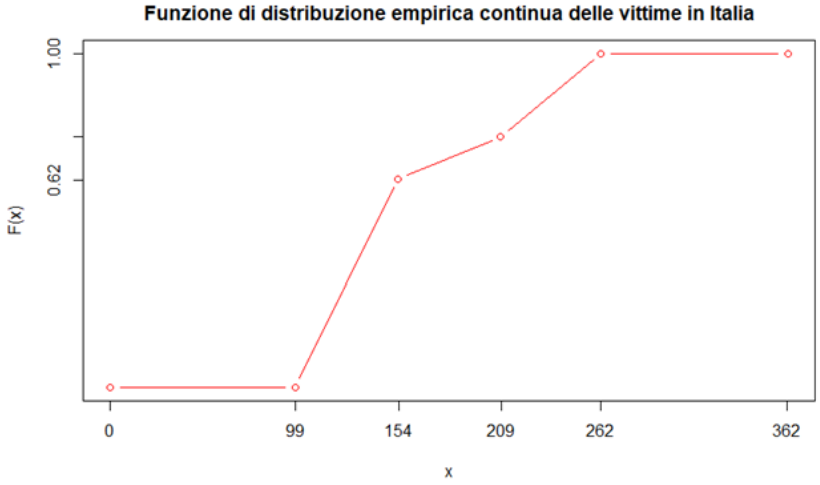
La funzione di distribuzione empirica continua viene utilizzata nel caso di dati continui che vengono strutturati in classi. Ad esempio, se si vuole considerare k classi distinte, le classi saranno così caratterizzate: $C_1 = [z_0, z_1)$, $C_2 = [z_1, z_2)$, … $C_k=[z_{k-1}, z_k]$ con $z_0 < z_1 < \ldots < z_{k-1} < z_k$, dove $z_0$ corrisponde al minimo delle osservazioni e $z_k$ corrisponde al massimo delle osservazioni. La funzione di distribuzione empirica continua viene calcolata a partire dalle frequenze relative cumulative associate alle varie classi.

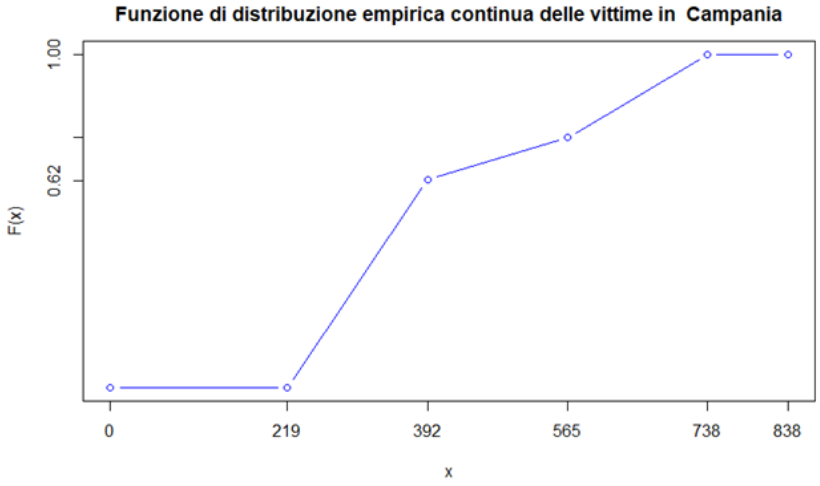
Per calcolare la funzione di distribuzione continua le osservazioni sono state divise in tre classi:

Per la Campania: $C_1 = [219,392)$, $C_2 =[392,565)$, $C_3 = [565,738]$

Per la media nazionale: $C_1 = [99,154)$, $C_2 = [154,209)$, $C_3 = [209,262]$

Di seguito verranno mostrati grafici che mostrano le frequenze di distribuzione continua in Campania e in Italia.



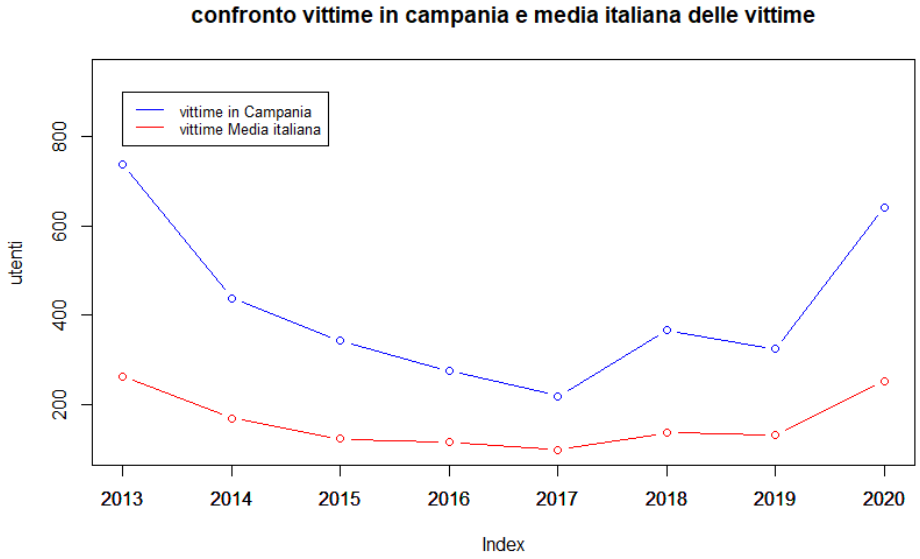




Dai grafici si può notare che avendo diviso i dati in 3 classi, le classi di entrambi i campioni di dati presentano le stesse frequenze relative. In particolare, la prima classe ha una frequenza relativa di 0.62, la seconda di 0.12 e la terza di 0.250.

## 2.2 INDICI DI SINTESI

Nel grafico seguente vengono mostrate le due curve relative ai dati che si stanno analizzando.

Di seguito è riportato il grafico che rappresenta le curve dei dati che si stanno analizzando, è stata preferenza del programmatore rappresentare le due curve in un solo grafo per mostrare meglio la differenza di numeri, ma andamento simile tra le vittime in media nazionale e le vittime in Campania




Il picco è presente in entrambi i casi nel 2013, per poi avere un andamento discendente fino al 2017 (anno del me too), per poi risalire dal 2018 e arrivare ad un incremento vertiginoso nel 2020.

È da ricordare che i numeri che si stanno analizzando fanno parte di denunce da donne, quindi, è lecito pensare che un grosso movimento quale il me too abbia dato coraggio alle donne che ricevevano abusi di farsi avanti e denunciare i propri aguzzini, nel 2020 l'obbligo della convivenza forzata ha solo incrementato quello che era già presente negli anni precedenti.

Alcuni indici di sintesi utili a descrivere i dati sono media, mediana, moda, varianza, deviazione standard e coefficiente di variazione. Le prime tre sono misure di centralità dei dati mentre le altre misurano la loro dispersione.

Supponiamo di avere un insieme, $x_1, x_2, \ldots, x_n$ di n valori numerici. Si definisce **media campionaria** $\bar{x}$ la quantità:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Le medie campionarie dei due campioni di dati negli anni risultano essere:

```
mean(utenti)
#418.125
mean(mediavittimeitalia)
#161.267
```

Pertanto, è possibile vedere quali sono gli anni in cui ci sono state più chiamate rispetto alla media e gli anni in cui ci sono state meno chiamate.

Sia per la Campania che per la media nazionale gli anni in cui ci sono state più chiamate rispetto alla loro media sono 2013, 2014 e 2020.

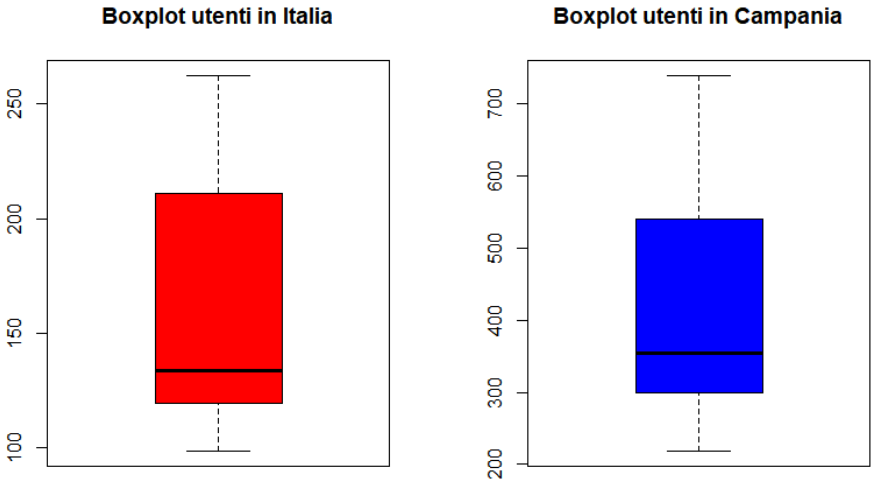
| *Media nazionale* | |
|---|---|
| *2013* | 262 |
| *2014* | 169 |
| *2015* | 123 |
| *2016* | 115 |
| *2017* | 98 |
| *2018* | 136 |
| *2019* | 131 |
| *2020* | 253 |

| *Campania* | |
|---|---|
| *2013* | 738 |
| *2014* | 438 |
| *2015* | 343 |
| *2016* | 275 |
| *2017* | 219 |
| *2018* | 366 |
| *2019* | 324 |
| *2020* | 642 |

Prima di illustrare i dati attraverso un boxplot è utile ricordare i concetti di quantili e di mediana. Dato un campione di dati ordinato in maniera crescente, si definisce la **mediana** (o **valore mediano**) come il valore/modalità assunto dalle unità statistiche che si trovano nel mezzo della distribuzione. Se n è dispari, la mediana sarà il valore in posizione (n+1)/2; se n è pari la mediana sarà la media aritmetica dei valori in posizione n/2 e n/2+1. La mediana, quindi è quel valore che divide a metà l'insieme dei dati ordinati. Oltre a questo indice si possono considerare altri indici di posizione detti quantili che consentono di suddividere l'insieme dei dati ordinati in un fissato numeri di parti uguali. In particolare, verranno considerati i quartili che consentono di dividere l'insieme dei dati ordinati in quattro parti uguali.

Il grafico seguente mostra, invece, i boxplot di entrambi i campioni di dati per illustrare alcune caratteristiche della distribuzione di frequenza come centralità, dispersione, forma e la presenza di eventuali valori anomali. Il boxplot, detto anche "scatola con i baffi", rappresenta una scatola i cui estremi sono $Q_1$ (primo quartile) e $Q_3$ (terzo quartile) tagliata da una linea orizzontale in corrispondenza di $Q_2$ (secondo quartile). Sono inoltre presenti due ulteriori linee che rappresentano i

baffi in alto e in basso. Il baffo inferiore corrisponde al valore più piccolo tra le osservazioni che risulta maggiore o uguale a $Q_1 - 1.5 * (Q_3 - Q_1)$, mentre il baffo superiore corrisponde al valore più grande delle osservazioni che risulta minore o uguale a $Q_3 + 1.5 * (Q_3 - Q_1)$. Se tutti i dati rientrano nell'intervallo $\left(Q_1 - 1.5 * (Q_3 - Q_1), Q_3 + 1.5 * (Q_3 - Q_1)\right)$, i baffi risultano essere posti in corrispondenza del minimo e del massimo dei dati del campione. I valori anomali al di fuori di tale intervallo vengono visualizzati sotto forma di punti nel grafico.




Entrambi i boxplot rivelano la presenza di asimmetria nei dati in quanto le distanze tra primo e terzo quartile dalla linea della mediana non sono molto diverse tra loro, ma nonostante ciò si può intuire che le curve hanno una coda più allungata a destra e ciò verrà confermato attraverso il calcolo della skewness campionaria.

Utilizzando la funzione summary in R è possibile calcolare minimo, massimo, media, mediana, primo e terzo quartile

```
> summary(vittimecampania)
"Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
219.0   311.8   354.5   418.1   489.0   738.0
> summary(mediavittimeitalia)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
98.77  121.23  133.80  161.27  190.48  262.41
```

Avendo ottenuto il valore dei quartili, è possibile calcolare il valore dei baffi del boxplot della Campania.

$(Q_1 - 1.5 * (Q_3 - Q_1)) = 311.8 - 1.5 * (489 - 311.8) = 46$

$(Q_3 + 1.5 * (Q_3 - Q_1)) = 489 + 1.5 * (489 - 311.8) = 754.8$

Tutti i dati rientrano nell'intervallo (46, 754.8) pertanto i baffi sono posti in corrispondenza del minimo e del massimo delle osservazioni.

Valore dei baffi nel boxplot della media nazionale:

$(Q_1 - 1.5 * (Q_3 - Q_1)) = 121.23 - 1.5 * (190.48 - 121.23) = 17.355$

$(Q_3 + 1.5 * (Q_3 - Q_1)) = 190.4 + 1.5 * (190.48 - 121.23) = 294.75$

Tutti i dati rientrano nell'intervallo (17.355, 294.75) pertanto i baffi sono posti in corrispondenza del minimo e del massimo delle osservazioni.

La **moda campionaria** di un insieme di dati è il valore a cui è associata la frequenza più elevata, non è obbligatorio che la moda esista in ogni insieme di dati e se esiste, è possibile che ne esista più di una; in questo caso, ogni valore è detto "valore modale". Se si hanno insieme di dati raggruppati in classi, la classe a cui è associata la frequenza più alta viene detta classe modale.

Per individuare la moda si considerano gli istogrammi delle frequenze dei dati considerando la loro suddivisione nelle seguenti cinque classi: $C_1 = [0, 500)$, $C_2 = [500, 1000)$, $C_3 = [1000, 1500)$ $C_4 = [1500, 2000)$, $C_5 = [2000, 2500]$.

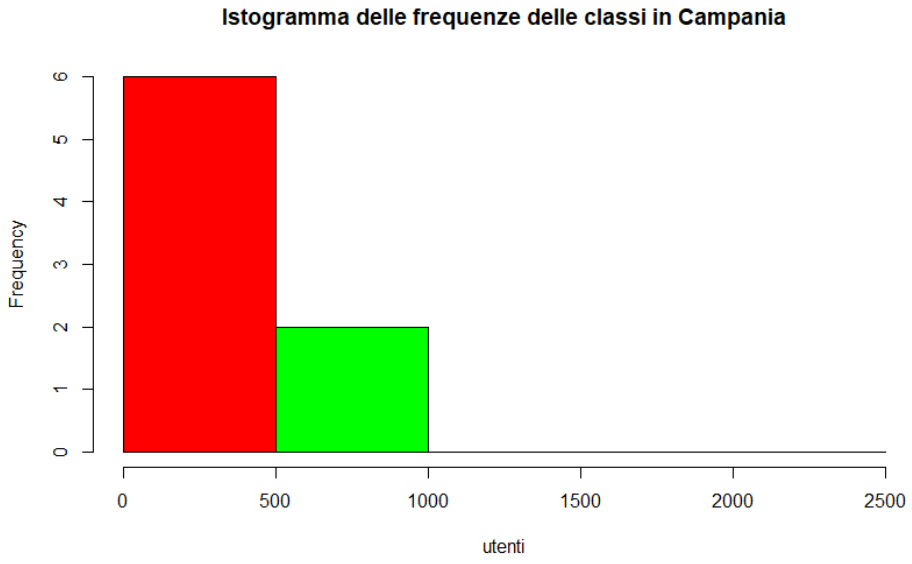
Sia per il primo grafico, che per il secondo, la classe modale è la prima: $C_1 = [0, 500)$

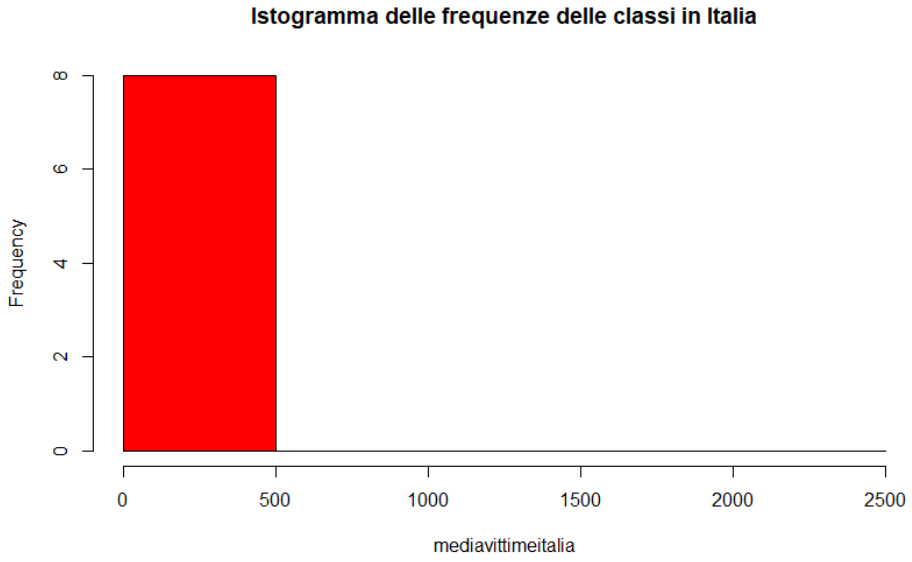
Il codice è:

```
#calcolo delle frequenze associate alle classi
classi<-c(0, 500, 1000, 1500, 2000, 2500)
fclassiCampania <-table (cut (utenti, breaks = classi,right = FALSE, dig.lab =
10))
for (i in 1:length(utenti)){
  if(utenti[i]==2500)
    fclassiCampania[3]<-fclassiCampania[3]+1
}
fclassiItalia <-table (cut (mediavittimeitalia, breaks = classi,right = FALSE,
dig.lab=10))
for (i in 1:length(mediavittimeitalia)){
  if(mediavittimeitalia[i]==2500)
    fclassiItalia[3]<-fclassiItalia[3]+1
}


#creazione degli istogrammi per le classi
hist(utenti, breaks=classi, col=rainbow(3), main="Istogramma delle frequenze
delle classi in Campania")
```

```
hist(mediavittimeitalia, breaks=classi, col=rainbow(3), main="Istogramma delle frequenze delle classi in Italia")
```




Avendo un insieme di dati numerici $(x_1 x_2 x_3 x_4 \dots \dots x_n)$, si definisce **varianza campionaria** e si indica con $s^2$, la quantità:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad (n = 2,3 \dots)$$

Si definisce **deviazione standard campionaria** la radice quadrata della varianza ossia:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \quad con \quad (n = 2,3 \dots)$$

Assegnato un campione di dati numerici $x_1$ , $x_2$ , …, $x_n$ , si definisce **coefficiente di variazione** il rapporto tra la deviazione standard campionaria e il modulo della media campionaria: $CV = \frac{s}{|\bar{x}|}$.

Il seguente codice permette di mostrare i valori della varianza, della deviazione standard e del coefficiente di variazione dei due campioni di dati.

```
> var(utenti)
[1] 32884.41
> sd(utenti)
[1] 181.3406
> coefficienteVariazioneCampania
[1] 0.4336995
>
```

```
> var(mediavittimeitalia)
[1] 3954.902
> sd(mediavittimeitalia)
[1] 62.88801
> coefficienteVariazioneItalia
[1] 0.3899619
```

Per entrambi i vettori, la varianza è abbastanza alta e possiamo dire che i valori si discostano dalla media.

La varianza e la deviazione standard di entrambi i campioni risultano essere dei valori grandi e da tali valori non si riesce ad avere una effettiva misura della dispersione, pertanto si considera il coefficiente di variazione. La varianza e la deviazione standard di entrambi i campioni risultano essere dei valori grandi e da tali valori non si riesce ad avere una effettiva misura della dispersione, pertanto si considera il coefficiente di variazione.

Il coefficiente di variazione del campione di dati della Campania è circa **0.4336**, mentre quello della media nazionale è circa **0.3899**. I due coefficienti sono tra loro vicini, tuttavia il coefficiente di variazione della Campania risulta essere maggiore rispetto a quello della media nazionale pertanto la media della Campania risulta essere meno attendibile e i singoli valori risultano essere più distanziati da essa.

## 2.3 FORMA DELLA DISTRIBUZIONE DI FREQUENZE

In questo paragrafo verranno descritti gli indici statistici che permettono di analizzare la forma della distribuzione di frequenze misurando se essa presenta asimmetrie (positive o negative) o se essa è

più o meno piccata rispetto ad una distribuzione di frequenze normale standard. Prima di definire tali indici è utile introdurre il concetto di momento campionario e di momento centrato.

Assegnato un insieme di dati numerici $x_1$ , $x_2$ , …, $x_n$ , si definisce **momento campionario** di ordine j la quantità: $M_j = \frac{1}{n}\sum_{i=1}^{n} x_i^j$

Assegnato un insieme di dati numerici $x_1$ , $x_2$ , …, $x_n$ , si definisce **momento campionario centrato** attorno alla media di ordine j la quantità: $m_j = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^j$

La skewness campionaria permette di misurare la simmetria di una distribuzione di frequenze. Assegnato un insieme di dati numerici $x_1$ , $x_2$ , …, $x_n$ , si definisce **skewness campionaria** il valore: $\gamma_1 = \frac{m_3}{m_2^{3/2}}$

Se la distribuzione è simmetrica il valore $\gamma_1$ è nullo, $\gamma_1 > 0$ se la distribuzione ha un'asimmetria positiva (ovvero una coda a destra più allungata), $\gamma_1 < 0$ se la distribuzione ha un'asimmetria negativa (ovvero una coda a sinistra più allungata).

Il codice per calcolare la skewness campionaria in R è:

```
skw <-function (x){
  n<-length (x)
  m2 <-(n -1) *var (x)/n
  m3 <- (sum ( (x- mean(x))^3) )/n
  m3/(m2 ^1.5)
}
```

**Skewness campionaria**. Per quanto riguarda la skewness campionaria, l'indice è positivo per entrambi i dati, ciò quindi significa che nella distribuzione di frequenze, la coda di destra è più allungata.

```
skw(utenti)
#0.8126429
skw(mediavittimeitalia)
#0.8410663
```

La **curtosi campionaria** è un indice che permette di misurare la densità dei dati intorno alla media.

Essa si calcola con la seguente equazione:

*Assegnati un insieme di dati numerici* $(x_1 x_2 x_3 \ldots . x_n)$*si definisce curtosi campionaria il valore*

$$\gamma_2 = \beta_2 - 3$$

Dove $\beta_2 = \frac{m_4}{m_2^2}$ è l'indice di Pearson e $m_2 \; ed \; m_4$ sono rispettivamente il momento centrato campionario di ordine 2 ed ordine 4.

Da notare anche che $\beta_2$ è indipendente dall'unità di misura dei dati.

Gli indici $\gamma_2 e \, \beta_2$ permettono di confrontare la dei dati con una densità di probabilità normale standard

- Se $\beta_2 < 3 \; e \; quindi \; \gamma_2 < 0$ abbiamo una distribuzione di frequenze platicurtica, è quindi più piatta di una normale
- Se $\beta_2 > 3 \; e \; quindi \; \gamma_2 > 0$ abbiamo una distribuzione di frequenze leptocurtica, è quindi più piccata di una normale
- Se $\beta_2 = 3 \; e \; quindi \; \gamma_2 = 0$ abbiamo una distribuzione di frequenze normocurtica, è quindi piatta come una normale

Il codice per calcolare la curtosi in R è:

```
curt <-function (x){
  n <-length (x)
  m2 <-(n -1) *var (x)/n
  m4 <- (sum ((x-mean(x))^4) )/n
  m4/(m2 ^2) -3
}
```

**Curtosi campionaria.** Il valore di entrambe le curtosi campionarie è negativo e quindi la distribuzione di frequenze è più piatta di una normale.

```
curt(utenti)
#-0.7215766
curt(mediavittimeitalia)
# -0.9292483
```

Confrontando i valori ottenuti da questi due indici si ha un'ulteriore conferma del fatto che l'andamento negli anni delle due curve considerate risulta essere molto simile anche se i dati relativi all'intera nazione sono più bassi in quanto sono ottenuti dalla media di tutte le regioni, che viene fortemente influenzata dai valori bassi presenti in molte regioni con meno abitanti rispetto alla Campania.

# 3 STATISTICA DESCRITTIVA BIVARIATA

In questo capitolo verranno mostrate le analisi di regressione lineare semplice e di regressione lineare multipla calcolando il modello lineare, i residui e il coefficiente di determinazione. Le analisi verranno effettuate sulla tabella Vittime e si cercherà di individuare eventuali correlazioni lineari tra i vari anni considerati.

La statistica descrittiva bivariata si occupa dei metodi grafici e statistici atti a descrivere le relazioni che intercorrono tra due variabili X e Y.

## 3.1 REGRESSIONE LINEARE SEMPLICE

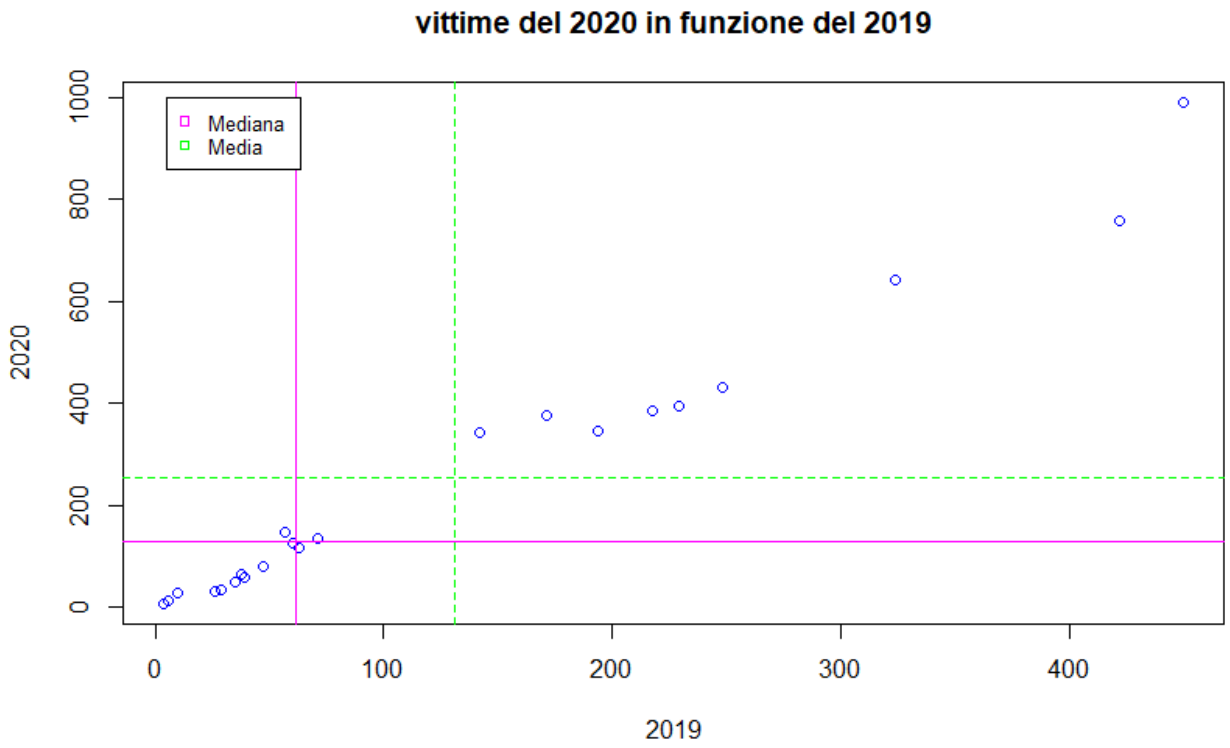
Il modello di regressione lineare semplice viene utilizzato per spiegare la relazione che esiste tra una variabile dipendente Y e una variabile indipendente X. In questa analisi verrà considerata come variabile indipendente l'anno 2019 e come variabile dipendente l'anno 2020.

Si calcolano gli indici di posizione e di dispersione relativi alle due coppie di variabili.

```
> median(dataframe$"2019")
[1] 61.5
> mean(dataframe$"2019")
[1] 131.0455
> sd(dataframe$"2019")
[1] 134.5527
> print("cambio anno")
[1] "cambio anno"
> median(dataframe$"2020")
[1] 130.5
> mean(dataframe$"2020")
[1] 253
> sd(dataframe$"2020")
[1] 269.0815
```

Un primo passo per indagare l'eventuale dipendenza tra due variabili X e Y consiste nel disegnare il diagramma di dispersione o scatterplot. Il grafico che si ottiene mira ad evidenziare se le coppie di punti presentano qualche forma di regolarità.

Nello scatterplot si pone sull'asse delle ascisse la variabile indipendente 2019 e sulle ordinate la variabile dipendente 2020. Vengono poi tracciate delle linee orizzontali e verticali in corrispondenza delle mediane e delle medie delle due variabili.




Dallo scatterplot si può notare come tutti i dati siano posizionati lungo una retta ascendente quindi si può dedurre che esiste una correlazione positiva tra le variabili considerate.

Per vedere se ciò è matematicamente provato, bisogna controllare la covarianza e la correlazione campionaria: la covarianza deve essere positiva e la correlazione deve essere prossima ad 1.

Per ottenere una misura quantitativa della correlazione tra le variabili si calcola la covarianza campionaria, che è così definita:

*Assegnato un campione bivariato* $(x_1,y_1),(x_2,y_2), \ldots,(x_n,y_n)$ *di una variabile quantitativa bi-dimensionale (X,Y), siano* $\bar{x}$ *e* $\bar{y}$ *rispettivamente le medie campionarie di* $x_1,x_2, \ldots,x_n$ *e di* $y_1,y_2, \ldots,y_n$. *La covarianza campionaria tra le due variabili X e Y è così definita:*

$$C_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})\ (y_i - \bar{y})$$

Se $C_{xy} > 0$ le variabili sono correlate positivamente, se $C_{xy} < 0$ le variabili sono correlate negativamente, se $C_{xy} = 0$ le variabili non sono correlate.

```
> cov(dataframe$"2019", dataframe$"2020")
[1] 35798.62
```

Per ottenere una misura quantitativa della correlazione tra le variabili si può anche considerare il coefficiente di correlazione campionario, che è così definito:

*Assegnato un campione bivariato $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$ di una variabile quantitativa bidimensionale (X,Y), siano $\bar{x}$ e $s_x$ la media campionaria e la deviazione standard di $x_1,x_2,\ldots,x_n$ ed inoltre siano $\bar{y}$ e $s_y$ la media campionaria e la deviazione standard di $y_1,y_2,\ldots,y_n$. Il* ***coefficiente di correlazione campionario*** *tra le due variabili X e Y è cosi definito:*

$$r_{xy} = \frac{C_{xy}}{s_x s_y}$$

Il coefficiente di correlazione campionario $r_{xy}$ misura la forza del legame di natura lineare esistente tra due variabili quantitative. In particolare, $-1 \leq r_{xy} \leq 1$ e il suo valore indica la direzione della retta interpolante.

- $r_{xy} = -1$: (correlazione perfetta negativa), tutti i punti sono allineati lungo una retta discendente;
- $-1 < r_{xy} < 0$ (correlazione negativa), i punti sono posizionati in una nuvola attorno ad una retta interpolante discendente;
- $r_{xy} = 0$: (nessuna correlazione), i punti sono completamente dispersi in una nuvola che non presenta alcuna evidente direzione di natura lineare;
- $0 < r_{xy} < 1$: (correlazione positiva), i punti sono posizionati in una nuvola attorno ad una retta interpolante ascendente;
- $r_{xy} = 1$ : (correlazione perfetta positiva), tutti i punti sono allineati lungo una retta ascendente.

```
> cor(dataframe$"2019",dataframe$"2020")
[1] 0.9887581
```

Le due variabili sono positivamente correlate poiché la correlazione è prossima ad 1.

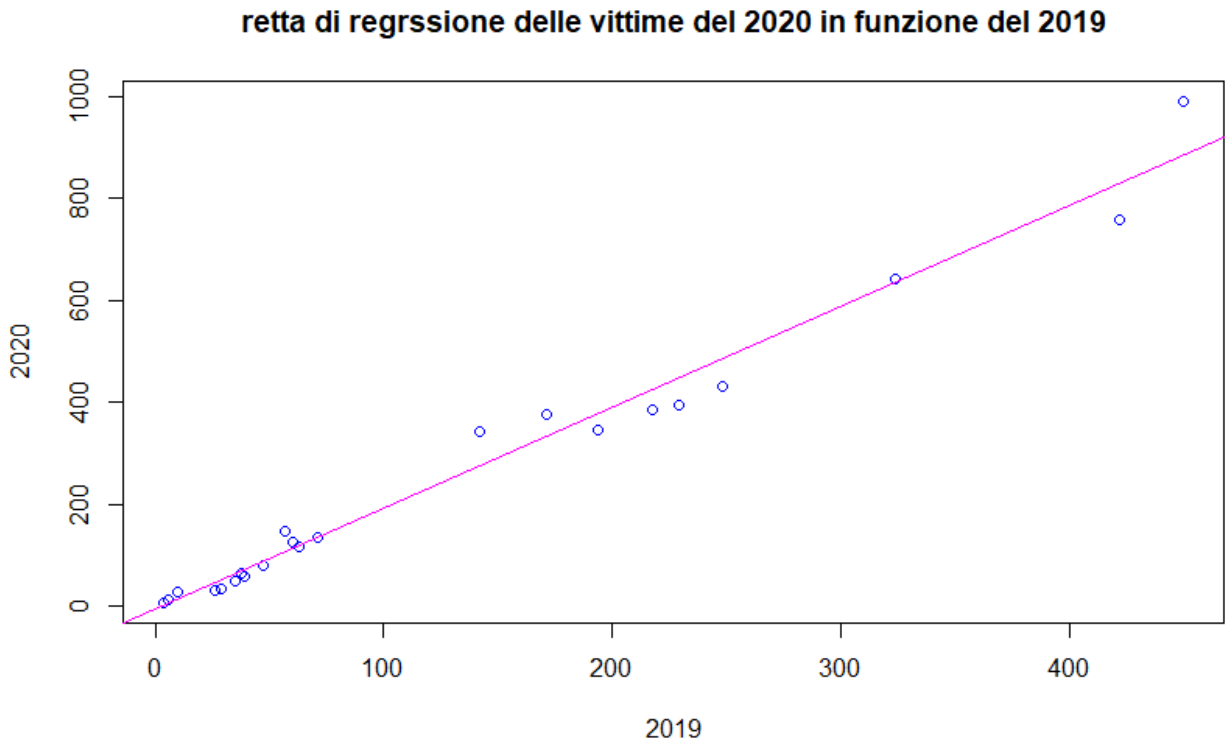
Il seguente grafico mostra lo scatterplot relativo ai dati del 2019 e del 2020 con la retta interpolante stimata.

Siccome il coefficiente di correlazione è uguale a 0.9924 che è prossimo ad 1 i dati presentano un'altissima correlazione positiva. Per calcolare la retta interpolante di questi punti si utilizza il modello di regressione lineare semplice.

Il modello di regressione lineare semplice è esprimibile tramite l'equazione di una retta $Y = \alpha + \beta X$ che riesce ad interpolare la nuvola di punti dello scatterplot meglio di tutte le altre possibili rette, dove $\alpha$ è l'intercetta e $\beta$ è il coefficiente angolare. Se $\beta > 0$ la retta di regressione è crescente, se $\beta < 0$ la retta di regressione è discendente, se $\beta = 0$ la retta è orizzontale. L'intercetta $\alpha$ corrisponde invece al punto di intersezione della retta interpolante con l'asse delle ordinate.

Il seguente codice permette di realizzare lo scatterplot relativo ai dati del 2019 e del 2020 con la retta interpolante stimata.

```
plot(df$"2019", df$"2020", main="Retta di regressione 2020 in funzione di
2019", col="blue",
     xlab="2019", ylab="2020")
abline(lm(df$"2020"~df$"2019"), col="magenta")
```




La funzione `lm(y~x)` permette di eseguire le analisi di regressione lineare della variabile dipendente y in funzione della variabile indipendente x.

Il seguente codice permette di ottenere il modello di regressione lineare per le due variabili. In particolare, l'intercetta vale -6.122, mentre il coefficiente angolare vale 1.977. Siccome il coefficiente angolare è positivo, la retta è ascendente. L'equazione della retta risulta quindi:

$$Y = -6.122 + 1.977x$$

```
Call:
lm(formula = dataframe$"2020" ~ dataframe$"2019")
Coefficients:
     (Intercept)  dataframe$"2019"
          -6.122             1.977
```

Dopo aver calcolato la retta interpolante, è possibile notare che esistono degli scostamenti tra i valori osservati del campione e i valori stimati attraverso la retta di regressione. Le differenze tra le ordinate dei punti dei valori osservati e le ordinate dei punti dei valori stimati prendono il nome di residui. Se si indica con $y_i$ il valore osservato e con $\hat{y}_i$ il valore stimato, i **residui** sono così definiti:

$$E_i = y_i - \hat{y}_i = y_i - (\alpha + \beta x_i) \quad (i = 1,2,\dots,n)$$

Il codice seguente permette di visualizzare i valori stimati.

```
> linearmodel$fitted.values
         1          2          3          4          5          6          7          8          9         10
484.258982   1.787853 134.269679 883.681802  63.085414  51.221370   5.742534 446.689509  70.994777 332.003749
        11         12         13         14         15         16         17         18         19         20
424.938761  69.017436 112.518932 828.316262 118.450954  13.651897 634.536875 274.660868  45.289348  86.813503
        21         22
377.482585 106.586910
>
```

Il seguente codice permette di visualizzare i residui, ossia di quanto i valori osservati si discostano dai valori stimati.

```
> round(linearmodel$residuals,3)
     1      2      3      4      5      6      7      8      9     10     11     12     13     14
-53.259   6.212  -0.270 106.318 -13.085 -16.221   7.257 -50.690 -10.995  44.996 -39.939  -5.017  14.481 -69.316
    15     16     17     18     19     20     21     22
 -2.451  14.348   7.463  69.339 -12.289  -6.814 -31.483  41.413
>
```

Valore dei residui standardizzati rispetto alla deviazione standard. Si può osservare che i valori sono molto piccoli.

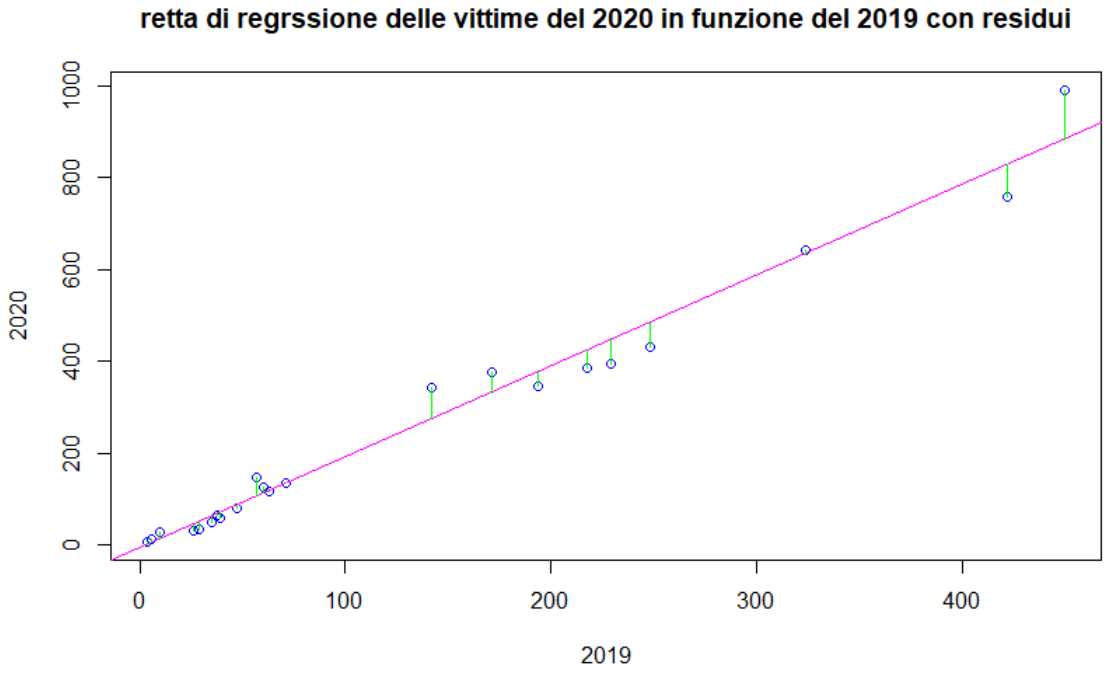
```
> residuistandard
          1           2           3           4           5           6           7           8
-1.323724094  0.154399661 -0.006702739  2.642483121 -0.325231120 -0.403173660  0.180380507 -1.259861185
          9          10          11          12          13          14          15          16
-0.273269427  1.118358257 -0.992656985 -0.124705755  0.359919366 -1.722819389 -0.060917173  0.356614581
         17          18          19          20          21          22
 0.185492074  1.723387789 -0.305445311 -0.169346041 -0.782483145  1.029300666
>
```

Le seguenti linee di codice mostrano i valori della mediana, della varianza e della deviazione standard dei residui.

```
> print(median(linearmodel$residuals))
[1] -3.734195
> print(var(linearmodel$residuals))
[1] 1618.791
> print(sd(linearmodel$residuals))
[1] 40.2342
```

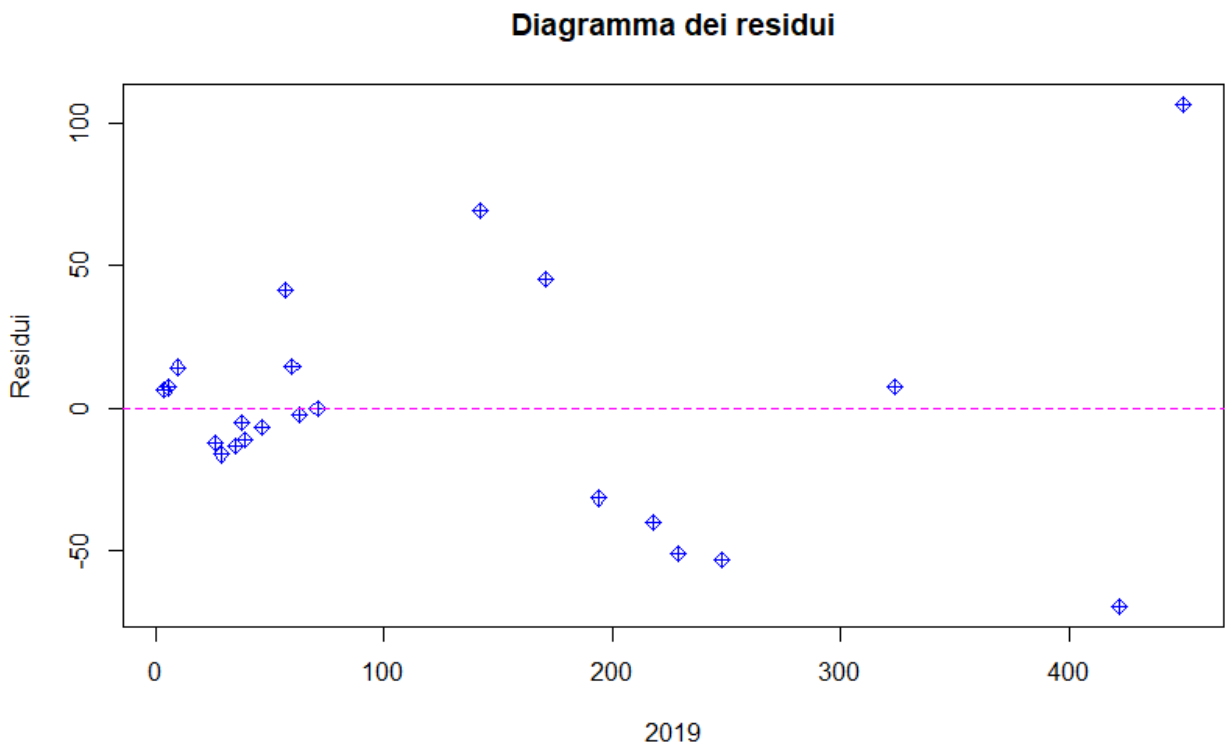
Di seguito viene mostrato il grafico che rappresenta lo scatterplot dei punti, la retta di regressione e i segmenti verticali che rappresentano i residui.

```
plot(df$"2019", df$"2020", main="Retta di regressione 2020 in funzione di 2019
con residui", col="blue",
     xlab="2019", ylab="2020")
abline(linearmodel, col="magenta")
segments (df$"2019", linearmodel$fitted.values, df$"2019", df$"2020"
,col="green")
```




Un esame più accurato del modo con cui la retta di regressione interpola i dati e di come i residui si dispongano intorno alla retta interpolante influenzandone la posizione, può essere ottenuto attraverso il diagramma dei residui che è un grafico in cui i valori dei residui sono posti sull'asse delle ordinate e quelli della variabile indipendente sull'asse delle ascisse.

```
plot(df$"2019", residui, main="Diagramma dei residui", xlab="2019",
ylab="Residui", col="blue", pch =9)
abline (h=0, col ="magenta",lty=2)
```




La linea tratteggiata è posizionata su 0 che indica la media campionaria dei residui. Si nota che i punti sono disposti casualmente attorno alla retta orizzontale e non si evidenzia nessun comportamento particolare nella distribuzione dei punti. La posizione della retta di regressione è fortemente influenzata dalla presenza di eventuali valori anomali che si discostano in modo significativo dagli altri. L'analisi dei residui aiuta ad individuare eventuali punti isolati (valori anomali) dovuti ad errori nella stima. Tali valori possono perturbare significativamente la stima dei parametri di regressione e influenzare l'interpretazione dei residui. Eliminando i valori anomali la varianza campionaria dei residui diminuisce.

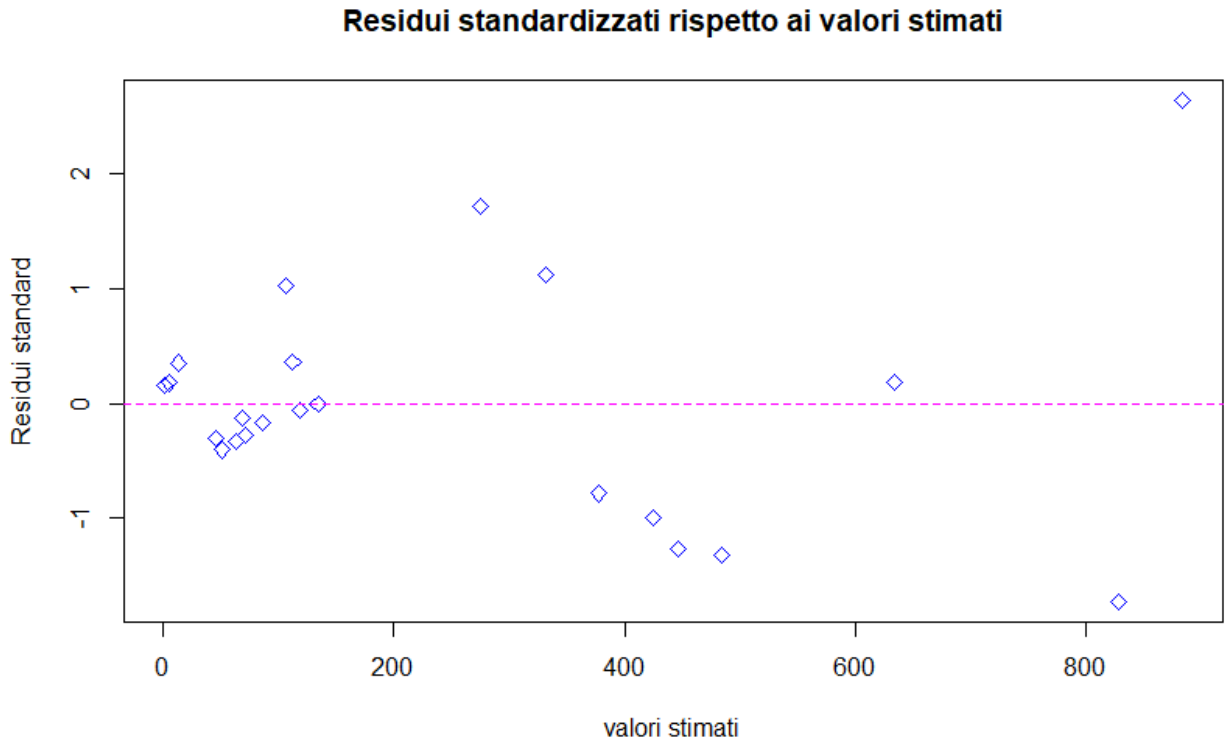
Spesso è utile calcolare i residui standardizzati, così definiti:

$$E_i^{(s)} = \frac{E_i - \bar{E}}{s_E} = \frac{E_i}{s_E}$$

I residui standardizzati sono caratterizzati da media nulla e varianza unitaria.

Successivamente è stato realizzato un grafico che mostra sulle ordinate i residui standardizzati e sulle ascisse i valori stimati.

La maggior parte dei residui standardizzati si concentra nell'intervallo [-1,1] e sono quei residui in corrispondenza di quei valori osservati che hanno lo stesso andamento dei valori attesi. Ci sono comunque valori che si discostano maggiormente dai propri valori attesi come Lombardia e Lazio. Per la Lombardia la differenza di ordinate tra valore osservato e valore stimato è circa 2.18 che è positivo, questo vuol dire che si è avuto un aumento di chiamate nel 2020 maggiore rispetto a quanto stimato dalla retta di regressione. Per il Lazio, invece, questa differenza vale circa -1.46 e si è quindi avuto un numero di chiamate nel 2020 più basso rispetto a quello stimato dalla retta di regressione (anche se è comunque un numero più alto rispetto all'anno precedente).




Per valutare quanto la retta di regressione si adatta ai dati si calcola il coefficiente di determinazione che si calcola effettuando il rapporto tra la varianza dei valori stimati tramite la retta di regressione e la varianza dei valori osservati. In questo caso il coefficiente di correlazione vale 0.9776425. Siccome è prossimo ad 1, significa che la retta descrive bene i dati considerati, infatti anche dai grafici visti precedentemente si nota che gli scostamenti dalla retta sono molto piccoli.

```
> print(summary(linearmodel)$r.square)
[1] 0.9776425
```

## 3.2 REGRESSIONE LINEARE MULTIPLA

Il modello di regressione lineare multipla viene utilizzato per spiegare la relazione tra una variabile quantitativa Y detta variabile dipendente e le variabili quantitative indipendenti $X_1, X_2, \ldots, X_p$.

Il modello di regressione lineare multipla con p variabili indipendenti è esprimibile attraverso l'equazione:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Dove:

- $\alpha$ è l'intercetta, ossia il valore di Y quando $X_1=X_2= \ldots = X_p=0$;
- $\beta_1, \beta_2, \ldots, \beta_p$ sono i regressori. In particolare, $\beta_1$ rappresenta l'inclinazione di Y rispetto alla variabile $X_1$ tenendo costanti le variabili $X_2, X_3, \ldots, X_p, \ldots, \beta_p$ rappresenta l'inclinazione di Y rispetto alla variabile $X_p$ tenendo costanti le variabili $X_1, X_2, \ldots, X_{p-1}$.

Si utilizza il modello di regressione lineare multipla per spiegare la relazione le variabili indipendenti: 2013, 2014, 2015, 2016, 2017, 2018, 2019 e la variabile dipendente: 2020

Valore degli indici di posizione e di dispersione (mediana, media e deviazione standard) relativi alle variabili:

```
> print(apply(dataframe, 2, median))
 2013  2014  2015  2016  2017  2018  2019  2020
149.0  88.0  83.0  72.0  55.5  66.5  61.5 130.5
> print(round(apply(dataframe, 2, mean),2))
  2013   2014   2015   2016   2017   2018   2019   2020
262.41 169.64 123.09 115.64  98.77 136.55 131.05 253.00
> print(round(apply(dataframe, 2, sd),2))
  2013   2014   2015   2016   2017   2018   2019   2020
255.50 174.45 123.72 120.29 108.55 142.43 134.55 269.08
>
```

Media e deviazione standard sono maggiori per la variabile 2013. La mediana è maggiore per l'anno 2020.

Matrice delle covarianze e sotto la matrice delle correlazioni che contiene tutte le correlazioni lineari tra le coppie di variabili:

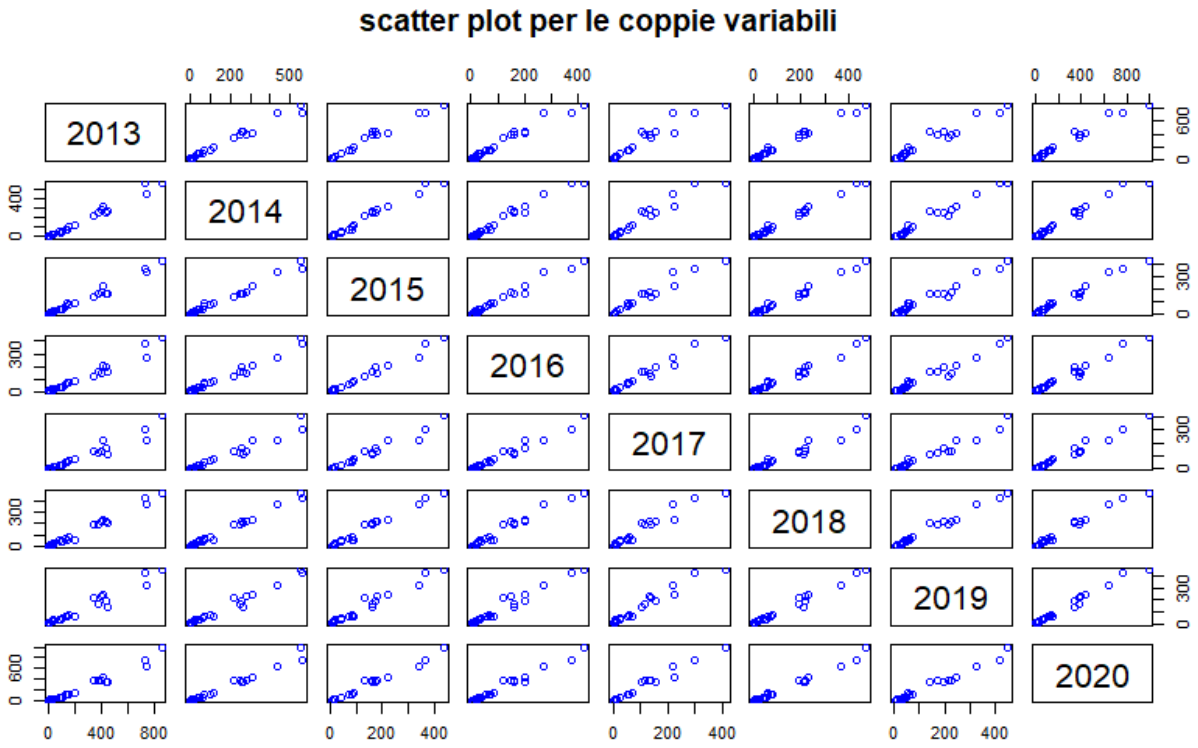
```
> round(cov(dataframe),2)
          2013     2014     2015     2016     2017     2018     2019     2020
2013 65279.49 44066.58 31249.87 30113.06 26492.38 36110.29 33403.50 67514.24
2014 44066.58 30433.00 21327.99 20706.86 18277.01 24726.59 23167.92 46237.95
2015 31249.87 21327.99 15307.61 14726.80 13126.55 17460.42 16321.19 32936.62
2016 30113.06 20706.86 14726.80 14469.86 12818.82 16925.59 15834.30 31934.29
2017 26492.38 18277.01 13126.55 12818.82 11784.18 14999.08 14266.96 28742.48
2018 36110.29 24726.59 17460.42 16925.59 14999.08 20287.12 18960.69 37990.71
2019 33403.50 23167.92 16321.19 15834.30 14266.96 18960.69 18104.43 35798.62
2020 67514.24 46237.95 32936.62 31934.29 28742.48 37990.71 35798.62 72404.86
> round(cor(dataframe),2)
     2013 2014 2015 2016 2017 2018 2019 2020
2013 1.00 0.99 0.99 0.98 0.96 0.99 0.97 0.98
2014 0.99 1.00 0.99 0.99 0.97 1.00 0.99 0.99
2015 0.99 0.99 1.00 0.99 0.98 0.99 0.98 0.99
2016 0.98 0.99 0.99 1.00 0.98 0.99 0.98 0.99
2017 0.96 0.97 0.98 0.98 1.00 0.97 0.98 0.98
2018 0.99 1.00 0.99 0.99 0.97 1.00 0.99 0.99
2019 0.97 0.99 0.98 0.98 0.98 0.99 1.00 0.99
2020 0.98 0.99 0.99 0.99 0.98 0.99 0.99 1.00
>
```

Si nota che esiste una forte correlazione lineare tra tutte le variabili considerate.

Dalla matrice della covarianza si può notare come tutte le coppie di variabili siano tra di loro positivamente correlate. Tali numeri sono però elevati e non suggeriscono quanto sia forte il legame tra le variabili pertanto viene considerato il coefficiente di correlazione.

Viene quindi mostrata la matrice delle correlazioni che contiene tutte le correlazioni lineari tra le coppie di variabili, ossia misura la forza del legame di natura lineare esistente tra tutte le coppie di variabili quantitative. La matrice delle correlazioni contiene 1 sulla diagonale principale.

Il seguente grafico visualizza in un'unica finestra tutti gli scatterplot ottenuti mettendo in relazione le varie coppie di variabili. Da tale grafico si può dedurre che le variabili sono altamente correlate e si intuisce che avranno un coefficiente di correlazione quasi pari ad 1.




Il modello di regressione lineare multipla con p variabili indipendenti è esprimibile attraverso l'equazione:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Dove:

- $\alpha$ è l'intercetta, ossia il valore di Y quando $X_1=X_2= \ldots = X_p=0$;
- $\beta_1, \beta_2, \ldots, \beta_p$ sono i regressori. In particolare, $\beta_1$ rappresenta l'inclinazione di Y rispetto alla variabile $X_1$ tenendo costanti le variabili $X_2$, $X_3$, …,$X_p$, …, $\beta_p$ rappresenta l'inclinazione di Y rispetto alla variabile $X_p$ tenendo costanti le variabili $X_1$, $X_2$, …,$X_{p-1}$.

Utilizzando il modello di regressione lineare multipla si ottiene:

```
> modellomultiplo

Call:
lm(formula = dataframe$"2020" ~ dataframe$"2013" + dataframe$"2014" +
    dataframe$"2015" + dataframe$"2016" + dataframe$"2017" +
    dataframe$"2018" + dataframe$"2019")

Coefficients:
     (Intercept)  dataframe$"2013"  dataframe$"2014"  dataframe$"2015"  dataframe$"2016"  dataframe$"2017"
        -6.91403           0.21504          -0.39135           0.19724           0.01286           0.74660
dataframe$"2018"  dataframe$"2019"
         0.74042           0.52855

>
```

Da cui si ricava che l'intercetta è -6.91403 e i regressori sono: 0.21504, -0.39135, 0.19724, 0.01286, 0.74660, 0.74042, 0.52855.

$$Y = -6.91403 + 0.21504X_1 - 0.39135X_2 + 0.19724X_3 + 0.01286X_4 + 0.74660X_5 + 0.74042X_6 + 0.52855X_7$$

I segni dei regressori $\beta_1, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8$ sono positivi: questo indica che all'aumentare del numero di utenti nel 2013, 2015, 2016, 2017,2018,2019 aumenta il numero di utenti nel 2020. Mentre il regressore $\beta_2$ è negativo quindi all'aumentare del numero di utenti nel 2014 diminuisce il numero di utenti nel 2020.

Il regressore del 2016 è prossimo allo zero, ciò vuol dire che l'aumento di vittime nel 2016 non incide molto sulle vittime del 2020.

Valori stimati rispetto al modello di regressione multipla:

```
> stime
          1           2           3           4           5           6           7           8
476.29905154   0.01758681 143.76055356 949.16059313  39.00366306  30.05376675   1.82082585 387.75459455
          9          10          11          12          13          14          15          16
 79.16087743 331.02340780 366.89503935  79.79297592 127.23857971 770.60310167 136.03981015  10.93635980
         17          18          19          20          21          22
657.30910619 330.84838703  28.78152444  77.44312426 403.10101667 138.95605436
```

Valori dei residui standardizzati rispetto alla deviazione standard.

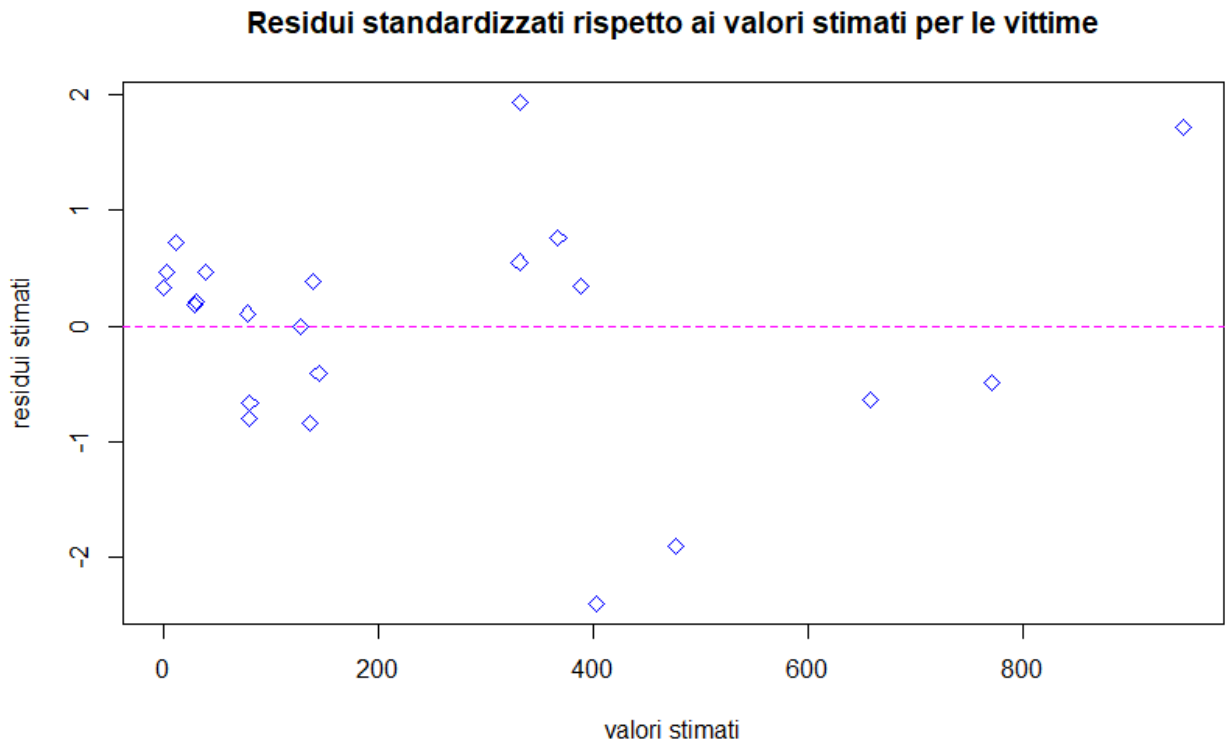
```
> residuistandardM
          1           2           3           4           5           6           7           8           9
-1.90479652  0.33565544 -0.41042512  1.71727128  0.46238903  0.20798599  0.47007722  0.34671409 -0.80570280
         10          11          12          13          14          15          16          17          18
 1.93328668  0.76130217 -0.66408467 -0.01003213 -0.48790310 -0.84266137  0.71751530 -0.64373825  0.55301702
         19          20          21          22
 0.17738423  0.10751501 -2.40106171  0.38029221
>
```

Residui dei valori osservati rispetto ai valori stimati.

```
      1       2       3       4       5       6       7       8       9      10      11      12      13      14
-45.299   7.982  -9.761  40.839  10.996   4.946  11.179   8.245 -19.161  45.977  18.105 -15.793  -0.239 -11.603
     15      16      17      18      19      20      21      22
-20.040  17.064 -15.309  13.152   4.218   2.557 -57.101   9.044
>
```

Di seguito viene mostrato il grafico che rappresenta i residui standardizzati in funzione dei valori stimati.

```
plot(stime, residuistandardM, main="Residui standardizzati rispetto ai valori
stimati", xlab="valori stimati"
     , ylab="Residui standard", pch=5, col="blue")
abline (h=0, col ="magenta",lty =2)
```




La linea tratteggiata è posizionata su 0 che indica la media campionaria dei residui. Si nota che i punti sono disposti casualmente attorno alla retta orizzontale e non si evidenzia nessun comportamento particolare nella distribuzione dei punti. La maggior parte dei punti sono concentrati nell'intervallo [-1,1] pertanto gli scostamenti dei valori osservati rispetto ai valori stimati risultano essere molto bassi. Solo per qualche regione tali scostamenti sono più elevati come Lombardia e Sicilia.

Anche in questo caso il coefficiente di determinazione è prossimo ad 1, infatti vale 0.9921889. Il modello di regressione lineare multipla descrive bene i dati considerati

```
> summary(modellomultiplo)$r.square
[1] 0.9921889
```

# 4 ANALISI DEI CLUSTER

L'**analisi dei cluster** è una tecnica matematica usata in informatica e altre discipline, essa si basa sul considerare diversi tipi di dati (numerici, persone, misure) ed unirli in gruppi che contengono tutti elementi che hanno somiglianze tra di loro. La creazione dei cluster può essere effettuata con diversi metodi, ma tutte le tecniche hanno in comune lo scopo di rendere quanto più possibili omogenei gli elementi all'interno di un gruppo e rendere quanto più eterogenei gruppi diversi così che il grado di associazione sia alto tra membri dello stesso gruppo e basso tra membri di gruppi diversi.

Le tecniche di raggruppamento tendono ad unire quei dati che sono tra di loro simili e svolgono questo lavoro basandosi sul concetto che ogni elemento di un certo insieme di dati ha delle caratteristiche osservabili che possono essere il colore degli occhi per le persone, o possono essere le denunce al numero verde 1522 fatte di anno in anno per una regione.

Per effettuare il partizionamento in cluster occorre definire delle misure di distanza o similarità tra i vari individui in base alle caratteristiche che si vogliono considerare. Una funzione a valori reali $(X_i X_j)$ è detta funzione distanza se e solo se soddisfa le seguenti condizioni:

- $d(X_i X_j) = 0$ se e solo se $X_i = X_j$ in $E_p$;
- $d(X_i X_j) \geq 0$ per ogni $X_i \; e \; X_j$ in $E_p$;
- $d(X_i X_j) = d(X_j X_i)$ per ogni $X_i \; e \; X_j$ in $E_p$;
- $d(X_i X_j) \leq d(X_i, \; X_k) + d(X_k, X_j)$ per ogni $X_i, X_k \; e \; X_j$ in $E_p$. (disuguaglianza triangolare)

In generale, verrà definita una matrice D contenente le distanze tra tutte le possibili coppie di individui.

Una funzione a valori reali $s_{ij} = s(X_i X_j)$ è detta misura di similarità se e soltanto se soddisfa le seguenti condizioni:

- $s(X_i X_i) = 1$;
- $0 \leq s(X_i X_j) \leq 1$;
- $s(X_i X_j) = s(X_j X_i)$ per ogni $X_i \; e \; X_j$.

È sempre possibile trasformare una misura di distanza in una misura di similarità, ma non viceversa in quanto le misure di similarità non godono della proprietà di disuguaglianza triangolare di cui invece godono le misure di distanza.

Per effettuare il partizionamento in cluster è stata utilizzata una misura di distanza, in particolare è stata utilizzata la **metrica Euclidea** così definita:

$$d_2(X_i X_j) = [\sum_{k=1}^{p} (x_{ik} - x_{jk})^2]^{1/2}$$

Dove $x_{ik}$ è il valore della k-esima caratteristica dell'individuo i.

Per effettuare il partizionamento in cluster un primo approccio a cui si potrebbe pensare è quello di considerare tutte le possibili suddivisioni. Tali metodi vengono detti metodi di enumerazione completa. Il numero totale di partizionare n individui in m cluster è dato dal numero di Stirling del secondo tipo così definito:

$$S(n,m) = \frac{1}{m!} \sum_{k=0}^{m} \binom{m}{k} (-1)^k (m-k)^n$$

Il codice per calcolare il numero di Stirling del secondo tipo in R è:

```
stirling2 <-function (n,m){
  s<-0
  if ((m >=1)&(m <=n)){
    for (k in seq (0,m)){
      s<-s+( choose (m,k)*(-1)^k*(m-k)^n/ factorial (m))}
    return (c(s))
  }
}
```

Se si volesse utilizzare tale metodo per gli individui considerati (22) per partizionarli in 2 cluster il numero di possibili partizionamenti sarebbe 2097151, mentre in 3 partizioni, il numero sarebbe 5228079450.

```
> stirling2(nrow(z),2)
[1] 2097151

> stirling2(nrow(z),3)
[1] 5228079450
```

Tali metodi risultano essere quindi molto onerosi, per questo vengono utilizzati i metodi non gerarchici e i metodi gerarchici.

Ottenuta la distanza dei vari elementi è necessario raggrupparli in cluster. Esistono due tipologie per creare i cluster.

- **Metodi gerarchici**: mirano a costruire gerarchie di cluster; si dividono in due tipologie di approcci diversi: L'approccio agglomerativo è un approccio "bottom-up", si parte dall'inserire ogni elemento in un singolo cluster e si procede ad accorparli a due a due; l'approccio divisivo è un approccio "top-down" che da un singolo cluster che comprende tutti gli elementi viene diviso in tanti sotto cluster. Tutti i metodi gerarchici producono una struttura ad albero chiamata "dendogramma".
  I metodi gerarchici hanno due vantaggi:
  - Forniscono una visione completa dell'insieme in termini di distanze;
  - Non comportano la scelta a priori del numero di cluster oppure la scelta a priori del numero di parametri da utilizzare per la determinazione automatica del loro numero.

  Uno svantaggio è che essi non consentono di riallocare gli individui che sono stati già classificati ad un livello precedente dell'analisi.
- **Metodi non gerarchici**: permettono di riposizionare elementi di un cluster qualora venga notato che un elemento piazzato in cluster conviene spostarlo in un altro, di questo metodo fa parte l'algoritmo k-means.

Per la suddivisione in cluster si è scelto inizialmente di considerare la suddivisione in 2 cluster. In seguito, si è deciso di effettuare un'ulteriore suddivisione in 3 cluster e di confrontare i risultati ottenuti. Tuttavia, al posto di considerare il data frame con i dati originali, si è scelto di scalarli sottraendo la media e dividendo per la deviazione standard, ottenendo dei dati standardizzati e più piccoli che risultano anche più semplici da gestire.

Il seguente codice permette di calcolare la matrice delle distanze euclidee a partire dal data frame Z scalato.

```
d<-dist(z, method="euclidean", diag=TRUE, upper=TRUE)
```

## 4.1 METODI GERARCHICI

Tra i metodi gerarchici ci sono:

- **Metodo del legame singolo**: Presa la matrice delle distanze, si parte dalla distanza 0; si trovano gli elementi che hanno la distanza minore, si rimuovono quegli elementi e si cercano i primi elementi che hanno la distanza minore. Se ci sono più coppie di elementi che hanno la distanza minore nella matrice se ne sceglie uno arbitrariamente.
- **Metodo del legame completo**: La distanza tra due gruppi g1 e g2, con n1 e n2 individui, è definita come la massima tra tutte le distanze di n1 e n2, questo metodo privilegia la differenza tra i gruppi piuttosto che l'omogeneità del gruppo stesso.
- **Metodo del legame medio**: nel metodo del legame medio si considera, come distanza tra due gruppi, la media di tutte le distanze calcolate a due a due tra tutti gli elementi dei due gruppi
- **Metodo del centroide**: la distanza tra i gruppi g1 e g2 è calcolata sulle medie campionarie dei due gruppi. La particolarità di questo metodo è che tende ad avere un effetto gravitazionale: I gruppi più grandi tendono ad assorbire i gruppi più piccoli.
- **Metodo della mediana**: il metodo è simile a quello del centroide, ma non è dipendente dalla numerosità del gruppo. Quando due gruppi si uniscono, il nuovo centroide è calcolato come la semisomma dei due gruppi precedenti.

Tra i metodi non gerarchici, il metodo usato nel progetto è stato "**k-means**", l'algoritmo funziona in diversi step:

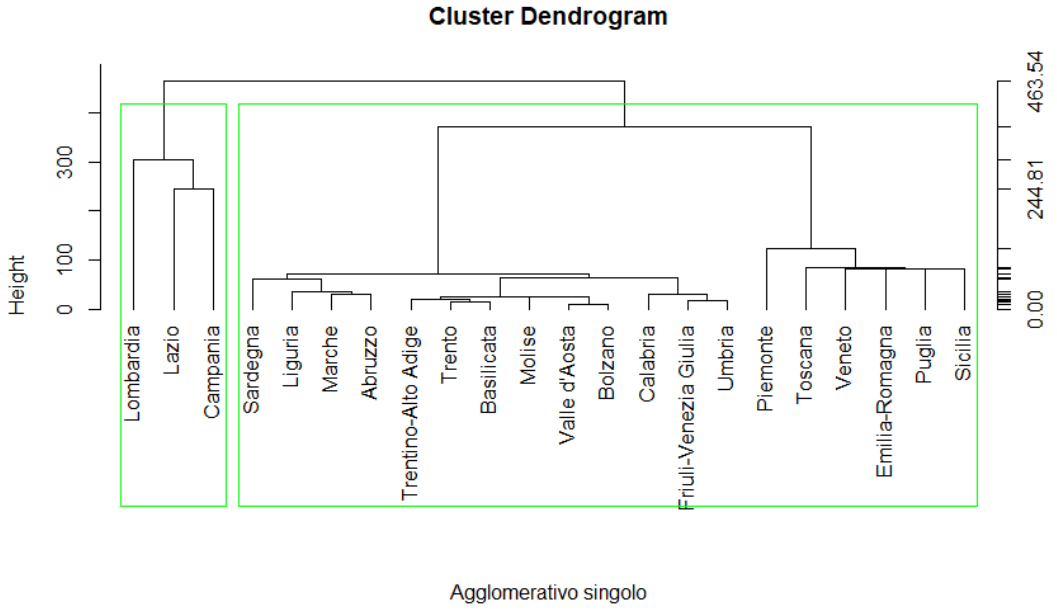
1. Si fissano a priori il numero dei cluster scegliendo però elementi che hanno determinate caratteristiche
2. Si considerano tutti gli elementi e si attribuisce ad ognuno un cluster basandosi sulla distanza minore dal punto di riferimento scelto per ogni cluster
3. Si ricalcolano i centroidi dei k gruppi costituendo il nuovo punto di riferimento per i cluster così ottenuti
4. Si rivalutano le distanze per ogni unità rispetto ai centroidi dei vari cluster. Se un elemento x ha una distanza minore ad un altro centroide rispetto a quello del proprio cluster, si riposiziona l'elemento.
5. Si ricalcolano i centroidi.
6. Si ripete dallo step 4, se si arriva ad un punto in cui non ci sono stati spostamenti tra elementi dei cluster, l'algoritmo si conclude.

Il seguente codice permette di calcolare la matrice delle distanze euclidee a partire dal dataframe Z.

```
d=dist(z,method = "euclidean",diag=TRUE,upper=TRUE)
```

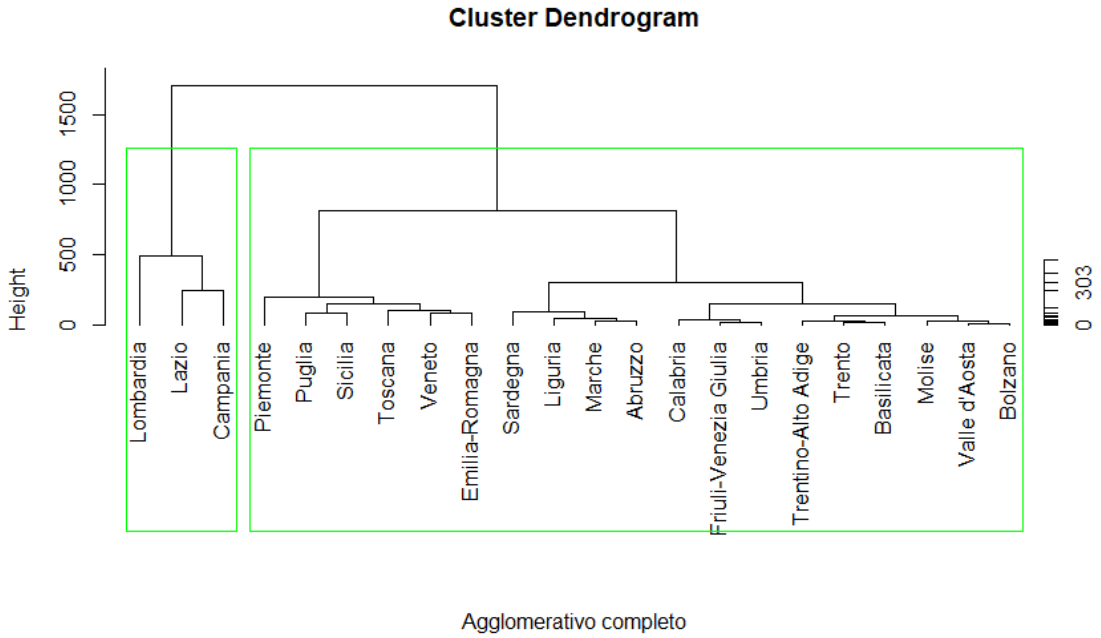
**Metodo del legame singolo**

```
hls=hclust(d,method = "single")
plot(hls,hang=-1,xlab = "Agglomerativo singolo",sub=" ")
rect.hclust(hls,k=2,border = "green")
axis(side=4,at=round(c(0,hls$height),2))
```




**Metodo del legame completo**

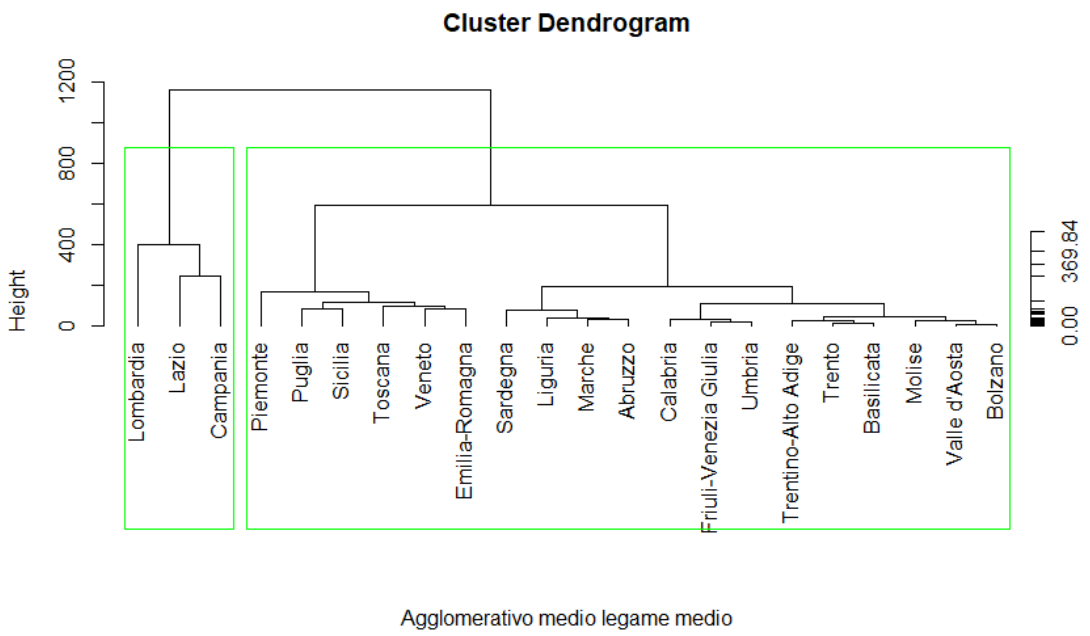
```
hlc=hclust(d,method = "complete")
plot(hlc,hang=-1,xlab="Agglomerativo completo",sub = " ")
rect.hclust(hlc,k = 2,border = "green")
axis(side=4,at=round(c(0,hls$height),0))
```




Agglomerativo completo

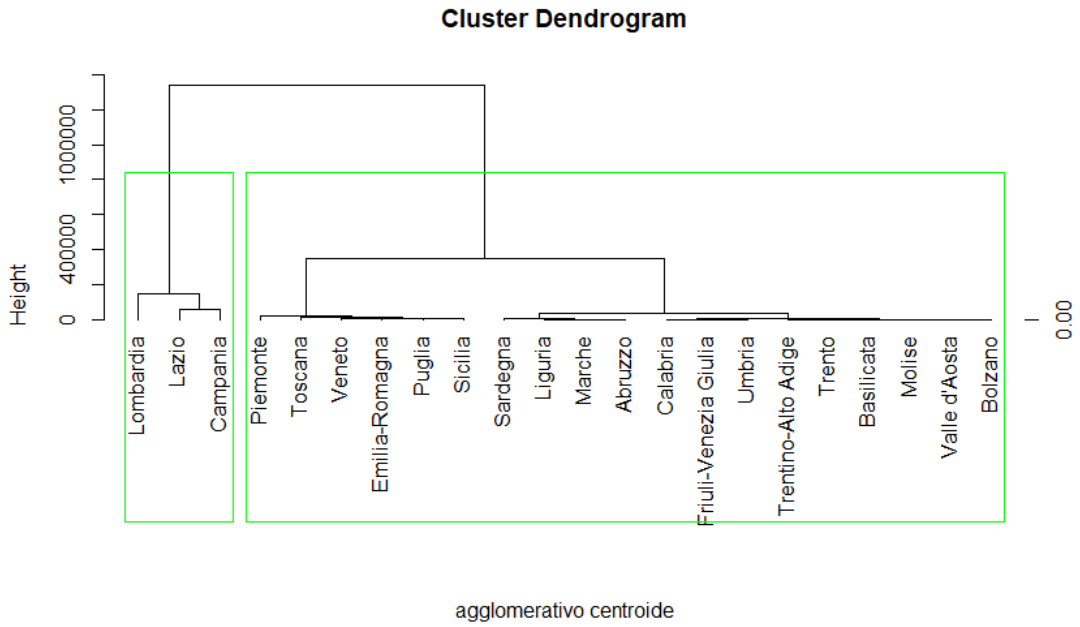
**Metodo del legame medio**

```
hlm=hclust(d, method="average")
plot(hlm, hang=-1, xlab="Agglomerativo medio legame medio", sub="")
rect.hclust(hlm, k=2, border="green")
axis(side=4, at=round(c(0, hls$height),2))
```



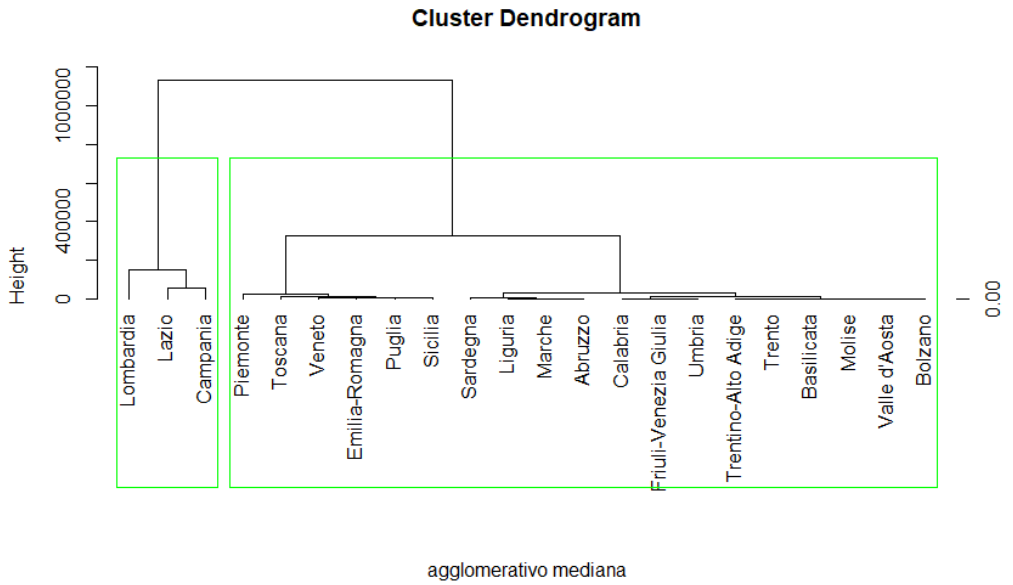
**Metodo del centroide**

```
d2=d^2
hc=hclust(d2,method = "centroid")
plot(hc,hang=-1,xlab = "agglomerativo centroide",sub=" ")
rect.hclust(hc,k=2,border = "green")
axis(side=4,at=round(c(0,hls$height),2))
```



**Metodo della mediana**

```
hmed=hclust(d2,method = "median")
plot(hmed,hang=-1,xlab = "agglomerativo mediana",sub=" ")
rect.hclust(hmed,k=2,border = "green")
axis(side=4,at=round(c(0,hls$height),2))
```




Tutti i metodi gerarchici: legame singolo, legame medio, legame completo, metodo del centroide e metodo della mediana hanno fornito il seguente partizionamento in due cluster.

Primo cluster: 19 individui

Secondo cluster: 3 individui

| | |
|---|---|
| **Cluster 2** | Piemonte, Valle d'Aosta, Liguria, Trentino-Alto Adige, Trento, Bolzano, Veneto, Friuli-Venezia Giulia, Emilia-Romagna, Toscana, Umbria, Marche, Abruzzo, Molise, Puglia, Basilicata, Calabria, Sicilia, Sardegna |
| **Cluster 1** | Lombardia, Lazio, Campania |

Per valutare quanto questa suddivisione è "buona" si calcolano le misure di non omogeneità relative all'insieme totale degli individui (trT), ai singoli cluster ottenuti e alla somma delle loro misure di non omogeneità (trS) e alla misura di non omogeneità tra i cluster (trB).

$$trT = trS + trB$$

Poiché per ogni fissata matrice X dei dati si ha che la trT è fissata, i cluster dovrebbero essere individuati in modo da minimizzare la misura di non omogeneità statistica all'interno dei cluster (within) e massimizzare la misura di non omogeneità statistica tra i gruppi (between). Se, fissato il numero di cluster, due metodi conducono a due partizioni differenti occorre scegliere la partizione con la misura di non omogeneità statistica all'interno dei cluster più piccola. Si calcola quindi il rapporto tra la misura di non omogeneità tra i gruppi e la misura di non omogeneità totale. Verrà quindi scelta la suddivisione che massimizza tale rapporto.

Si mostra in R il codice per il calcolo delle misure di non omogeneità per i cluster ottenuti con il metodo del legame singolo. Siccome il partizionamento ottenuto è uguale anche per gli altri metodi i risultati saranno uguali.

```
n<-nrow(Z)
trT<-(n-1)*sum(apply(Z,2,var)) #misura di non omogenità totale
taglio<-cutree(hls, k=2)
num <-table (taglio) #numero di elementi dei gruppi
tagliolist<-list(taglio) #lista di indici per i gruppi
agvar <- aggregate (Z, tagliolist, var)[, -1]
trH1<-(num[[1]]-1)*sum(agvar [1, ]) #misura di non omogenità del primo gruppo
trH2<-(num[[2]]-1)*sum(agvar [2, ]) #misura di non omogenità del secondo gruppo
trB<-trT-trH1-trH2 #misura di non omogenità tra i cluster
rapportoLegameSingolo<-trB/trH
```

Applicando la funzione `cuttree` si ottiene un vettore contenente numeri interi positivi per indicare i cluster a cui sono stati associati gli individui. Successivamente si ricava il numero di elementi associati a ciascun cluster con l'istruzione `num<-table(taglio).` Il primo cluster contiene 19 individui, il secondo ne contiene 3.

Si trasforma poi l'array ottenuto tramite `cuttree` in una lista di indici per i vari gruppi. La funzione `agvar<-aggregate(z, tagliolist, var)` permette di aggregare le colonne del dataframe z in base alla lista di indici passata che corrisponde quindi ai cluster. A tali gruppi viene applicata la funzione di varianza campionaria, avendo il seguente output.

```
1       1 0.3982756 0.3907965 0.3133831 0.3194185 0.3462675 0.3660319 0.3763633 0.3297782
2       2 0.0694603 0.1493225 0.1518417 0.4185481 0.8076079 0.1478935 0.2417825 0.4331063
> |
```

Prima di tutto viene calcolata la misura di non omogeneità totale all'interno del dataframe z utilizzando la seguente istruzione: `trT<-(n-1)*sum(apply(z,2,var))`. La funzione `apply`

permette di applicare la funzione varianza alle colonne del dataframe Z. Per calcolare la misura di non omogeneità i valori delle varianze delle singole colonne vengono sommati e si moltiplica il tutto per il numero di individui nel dataframe (a cui si sottrae 1). Pertanto, la misura di non omogeneità totale risulta:

$$trT = (22 - 1) * 8 = 168$$

Per calcolare la misura di non omogeneità all'interno del primo cluster si utilizza l'istruzione (num[[1]]-1)*sum(agvar [1, ]) che consente di sommare le colonne della prima riga della matrice agvar (ottenendo 2.840315) e successivamente si moltiplica tale valore per il numero di individui nel cluster -1. Quindi:

$trH1 = (19 - 1) * 2.840315 = 51.12566$

Per quanto riguarda il secondo cluster invece si ottiene:

$trH2 = (3 - 1) * 2.419563 = 4.839126$

Pertanto, la misura di non omogeneità tra i cluster risulta essere:

$trB = trT - trH1 - trH2 = 168 - 51.12566 - 4.839126 = 112.0352$

Il rapporto risulta $\frac{trB}{trT} = \mathbf{0.6668763}$

La suddivisione ottenuta con i metodi gerarchici risulta essere abbastanza buona in quanto in termini percentuali è del 66.6%.

## 4.2 METODI NON GERARCHICI

Tra i metodi non gerarchici, il metodo usato nel progetto è stato "**k-means**", l'algoritmo funziona in diversi step:

1. Si fissa a priori il numero dei cluster e si scelgono m punti di riferimento iniziali che inducono una prima partizione provvisoria;
2. Si considerano tutti gli elementi e si attribuisce ognuno al cluster individuato dal punto di riferimento da cui ha la distanza minore;
3. Si ricalcolano i centroidi dei k gruppi costituendo i nuovi punti di riferimento per i cluster;

4. Si rivalutano le distanze per ogni unità rispetto ai centroidi dei vari cluster. Se un elemento x ha una distanza minore in corrispondenza di un altro centroide rispetto a quello del proprio cluster, si riposiziona l'elemento;
5. Si ricalcolano i centroidi;
6. Si ripete dallo step 4, se si arriva ad un punto in cui non ci sono stati spostamenti tra elementi dei cluster, l'algoritmo si conclude.

Il metodo non gerarchico K-means ha fornito il seguente partizionamento in due cluster.

Primo cluster: 18 individui

Secondo cluster: 4 individui

| | |
|---|---|
| **Cluster 1** | Valle d'Aosta, Liguria, Trentino-Alto Adige, Trento, Bolzano, Veneto, Friuli-Venezia Giulia, Emilia-Romagna, Toscana, Umbria, Marche, Abruzzo, Molise, Puglia, Basilicata, Calabria, Sicilia, Sardegna |
| **Cluster 2** | Piemonte, Lombardia, Lazio, Campania |

Il rapporto $\frac{trB}{trT} = \mathbf{0.7095953}$.

La suddivisione in cluster ottenuta con il metodo non gerarchico K-means risulta essere migliore in quanto supera il 70% mentre quella ottenuta con i metodi gerarchici era circa 66.6%.

### 4.3 SUDDIVISIONE CON 3 CLUSTER

*Ma che cosa succederebbe se si volesse suddividere l'insieme degli individui in 3 cluster anziché 2?*

Suddividendo in 3 cluster si è ottenuto sia con i metodi gerarchici che con il metodo k-means il seguente partizionamento:

Primo cluster: 3 individui

Secondo cluster: 13 individui

Terzo cluster: 6 individui

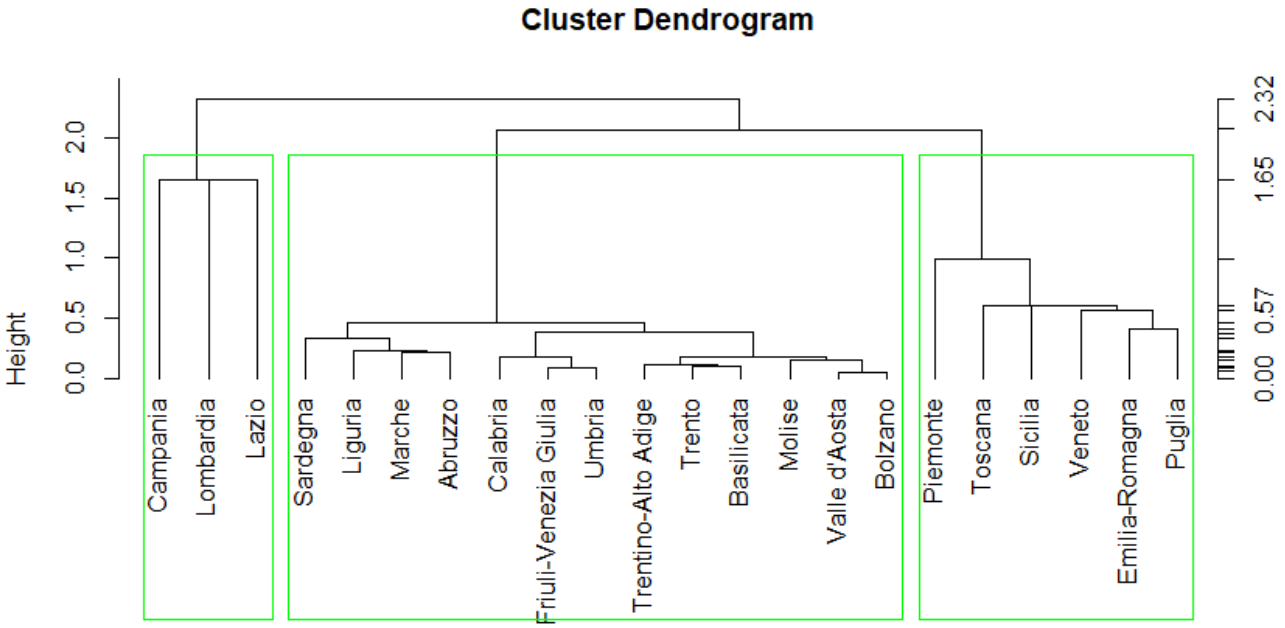
| | |
|---|---|
| **Cluster 1** | Campania, Lombardia, Lazio |
| **Cluster 2** | Sardegna, Liguria, Marche, Abruzzo, Calabria, Friuli-Venezia Giulia, Umbria, Molise, Valle d'Aosta, Bolzano, Trentino-Alto Adige, Trento, Basilicata |
| **Cluster 3** | Piemonte, Sicilia, Veneto, Puglia, Emilia-Romagna, Toscana |

**Tabella riassuntiva del rapporto tra misura di non omogeneità tra i cluster e misura di non omogeneità totale** $(\frac{trB}{trT})$

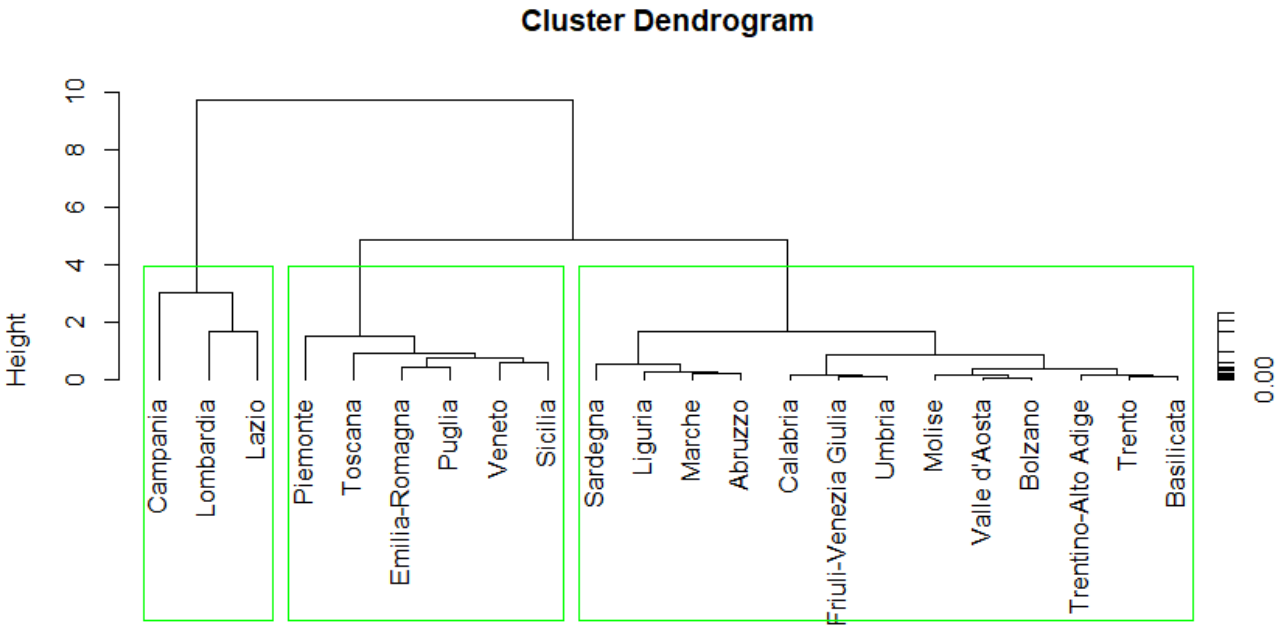
| Metodo | Rapporto con 2 cluster | Rapporto con 3 cluster |
|---|---|---|
| **Metodo del legame singolo** | 0.6668763 | 0.9326197 |
| **Metodo del legame completo** | 0.6668763 | 0.9326197 |
| **Metodo del legame medio** | 0.6668763 | 0.9326197 |
| **Medio del centroide** | 0.6668763 | 0.9326197 |
| **Metodo della mediana** | 0.6668763 | 0.9326197 |
| **Metodo k-means** | 0.7095953 | 0.9326197 |

Se si volesse suddividere l'insieme degli individui in 2 cluster, la suddivisione ottenuta con il metodo non gerarchico k-means risulta essere migliore. Se invece, si volesse suddividere l'insieme in 3 cluster la suddivisione risulta essere uguale pertanto si ottiene lo stesso rapporto. La suddivisione in 3 cluster risulta essere migliore in quanto 0.9375509 > 0.7095953.

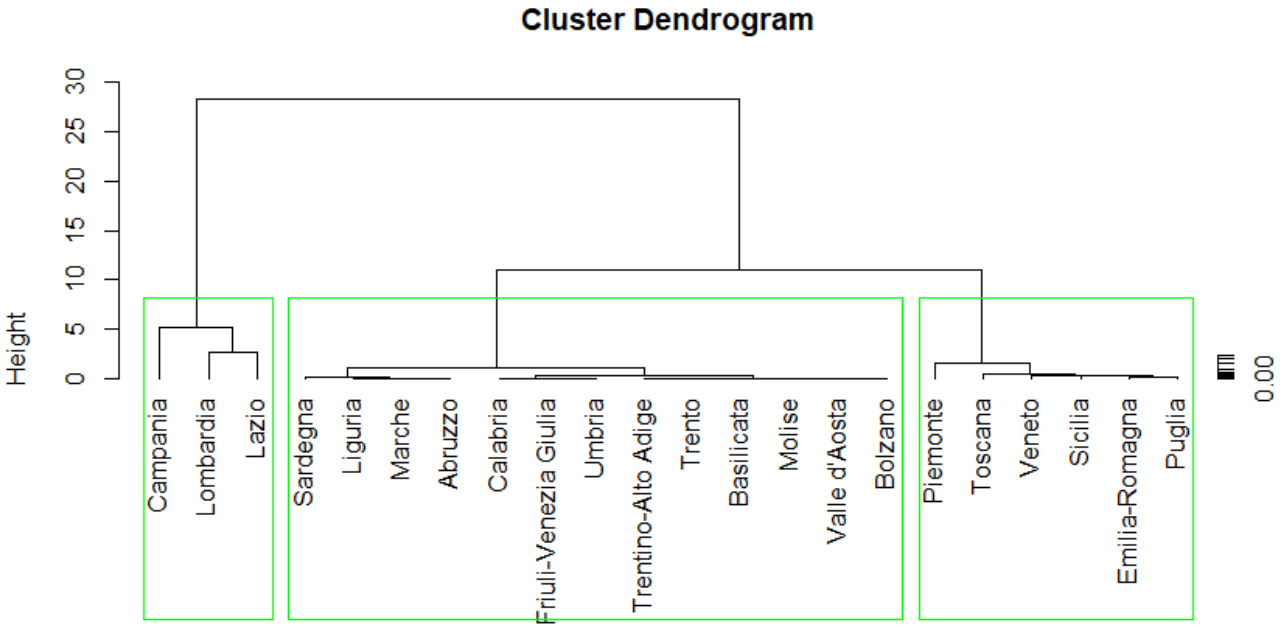
Di seguito, vengono mostrati i dendogrammi con la suddivisione in tre cluster:



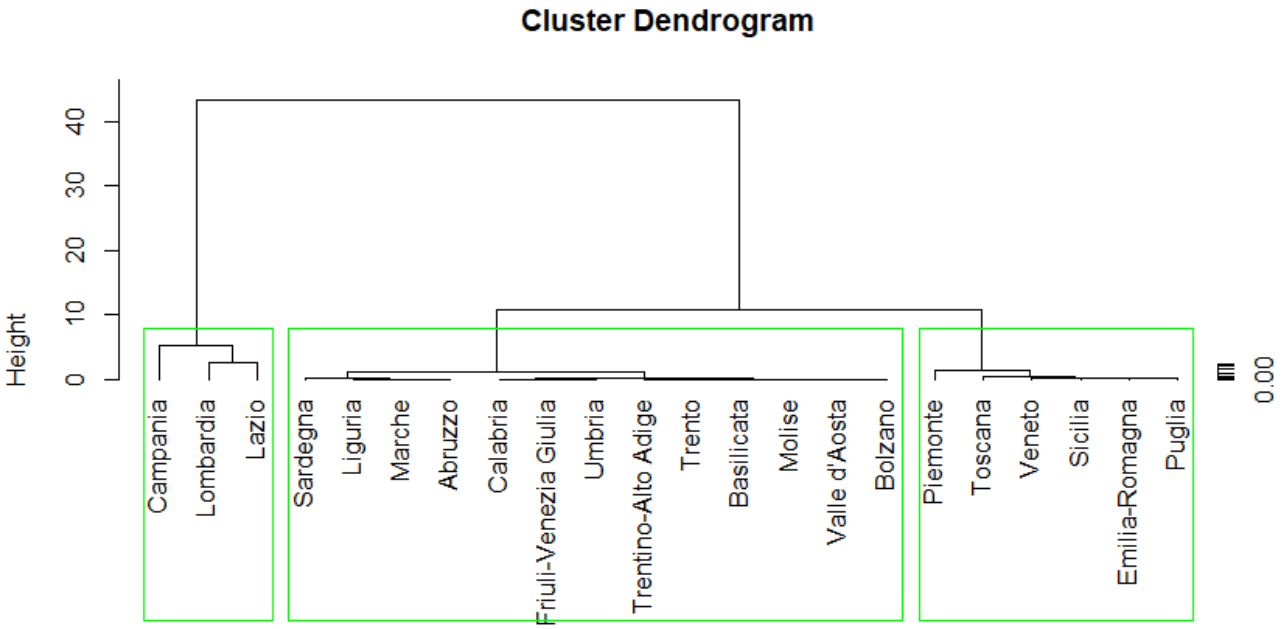

Metodo gerarchico agglomerativo
del legame singolo



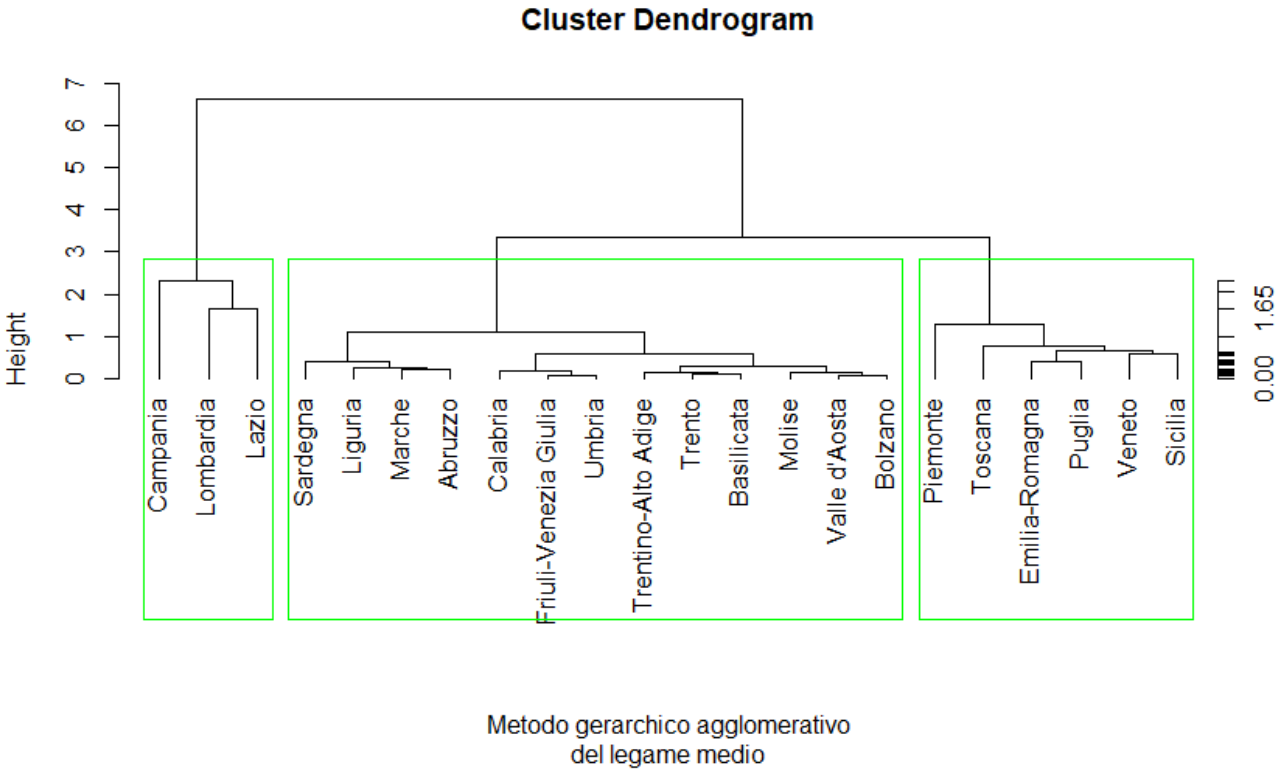

Metodo gerarchico agglomerativo
del legame completo




Metodo gerarchico agglomerativo
della mediana




Metodo gerarchico agglomerativo
del centroide


Cluster Dendrogram
Height
0 1 2 3 4 5 6 7
Campania
Lombardia
Lazio
Sardegna
Liguria
Marche
Abruzzo
Calabria
Friuli-Venezia Giulia
Umbria
Trentino-Alto Adige
Trento
Basilicata
Molise
Valle d'Aosta
Bolzano
Piemonte
Toscana
Emilia-Romagna
Puglia
Veneto
Sicilia
0.00 1.65
Metodo gerarchico agglomerativo
del legame medio