Corso di Laurea Magistrale in Informatica

Progetto di statistica e analisi dei dati

Professoressa:

Amelia G. Nobile

Studentessa:

Anna Tomeo

Anno 2018/2019

Prendiamo in esame un database contenente, per ogni regione, il numero di musei (e simili) aperti in diversi periodi.

I periodi presi in esame sono: tutto l'anno, stagionalmente, periodicamente, occasionalmente. Inoltre, il database contiene anche il numero di musei, per ogni regione, dai quali l'Istat non ha ricevuto risposta.

La situazione presa in esame è relativa a numeri.

L'Ambiente utilizzato: R.

Il database utilizzato è il seguente:

Periodo apertura	tutto l'anno	stagionalmente	periodicamente	occasionalmente	mancata risposta
Territorio					
Piemonte	198	101	63	45	20
Valle d'Aosta / Vallée d'Aoste	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
Provincia Autonoma Bolzano / Bozen	31	55	4	2	6
Provincia Autonoma Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

Osservazione: la regione Trentino Alto Adige è divisa in p.a.Bolzano e p.a. Trento

Di seguito è riportata tale tabella contenente dati quantitativi:

```
pApertura<-
cbind(tuttoAnno=c(198,29,131,259,31,40,188,101,303,330,107,184,274,61,32,166,104,31,120,200,15),
stag=c(101,22,29,54,55,38,47,29,55,73,23,65,12,32,3,19,13,5,19,19 ,30),
period=c(63,23,27,64,4,8,49,37,65,85,34,48,36,17,3,12,13,2,5,9,16),
occas=c(45,5,20,17,2,4,14,12,33,29,2,33,16,7,0,10,9,2,17,9,19),
mancataRisp=c(20,5,10,15,6,1,17,6,22,31,10,15,10,4,4,12,14,3,11,20,25))

rownames(pApertura)<-c ("Piemonte", "Valle d'Aosta", "Liguria", "Lombardia", "p.a. Bolzano", "p.a.
Trento", "Veneto", "Friuli-Venezia-Giulia", "Emilia-Romagna", "Toscana", "Umbria", "Marche",
"Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia", "Sardegna")
```

	tuttoAnno	stag	period	occas	mancataRisp
Piemonte	198	101	63	45	20
Valle d'Aosta	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
p.a. Bolzano	31	55	4	2	6
p.a. Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia-Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

Osservazione: per tutta la matrice dei dati abbiamo la stessa unità di misura, ossia numeri

Facciamo ora un'analisi che ci permette di fotografare la situazione in generale. Calcoliamo le frequenze marginali sulle colonne, tramite il comando margin.table:

```
> margin.table(pApertura,2)
```

tuttoAnno	stag	period	occas	mancataRisp
3047	743	620	305	261

Questo mostra il totale dei musei aperti per ognuno dei periodi considerati, senza tenere conto della regione. *Notiamo che i musei sono aperti più che altro per l'intero anno.*

Calcoliamo le frequenze marginali sulle righe:

```
> margin.table(pApertura,1)
```

```
       Piemonte    Valle d'Aosta          Liguria 
            427               84              217 
      Lombardia     p.a. Bolzano      p.a. Trento 
            409               98               91 
         Veneto Friuli-Venezia-Giulia Emilia-Romagna 
            315              185              478 
        Toscana           Umbria           Marche 
            548              176              345 
          Lazio         Abruzzo           Molise 
            348              121               42 
       Campania          Puglia       Basilicata 
            219              153               43 
       Calabria         Sicilia        Sardegna 
            172              257              248 
```

In tale caso, invece, abbiamo il totale dei musei aperti per ogni regione, senza tenere conto del periodo. *Notiamo che la Toscana è la regione con il maggior numero di musei aperti.*

Grafici

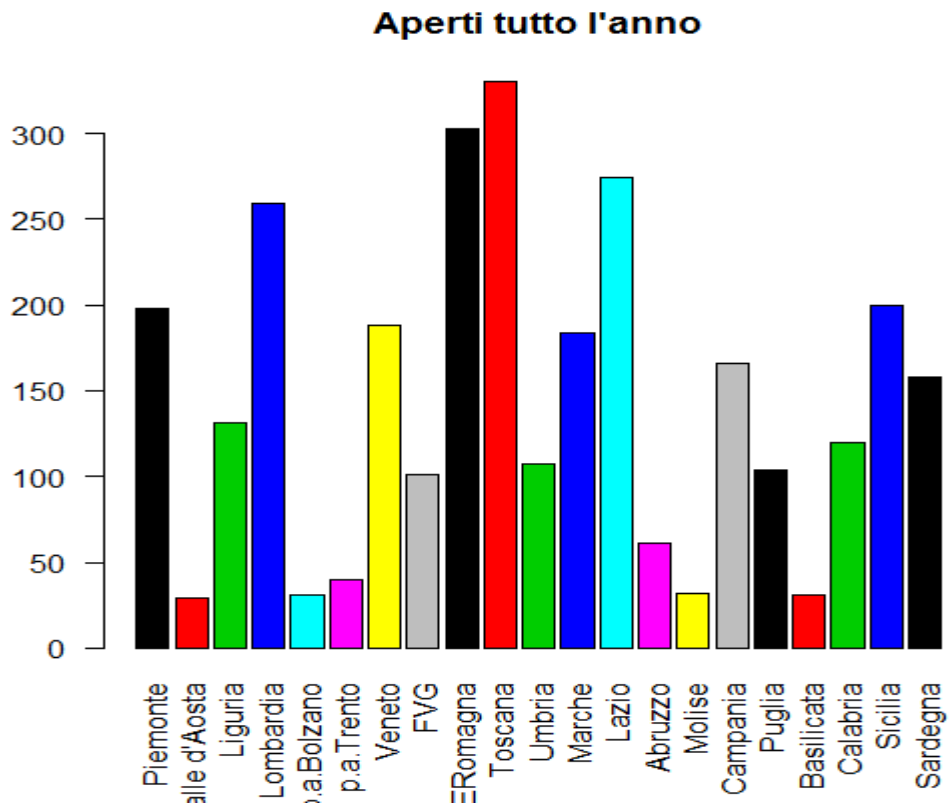
grafici a barre (per periodo)

Consideriamo una variabile qualitativa X e indichiamo con $z_1,\ldots, z_k$ le modalità distinte da essa assunte. Consideriamo poi un campione x = $(x_1,\ldots, x_n)$ costituito da n osservazioni di X. Disponiamo sull'asse orizzontale ed in modo equispaziato le modalità assunte da X e sull'asse verticale riportiamo le frequenze assolute o le frequenze relative. Tracciamo dei rettangoli centrati sulle modalità z_i tutti della stessa base e altezza pari alle frequenze (assolute o relative), ottenendo un grafico (o diagramma) a barre. Tale grafico viene prodotto in R tramite il comando barplot.

Per prima cosa consideriamo la nostra matrice dei dati pApertura e, per ogni colonna, tracciamo un grafico a barre in cui l'asse orizzontale contiene le regioni e l'asse verticale contiene le frequenze assolute:

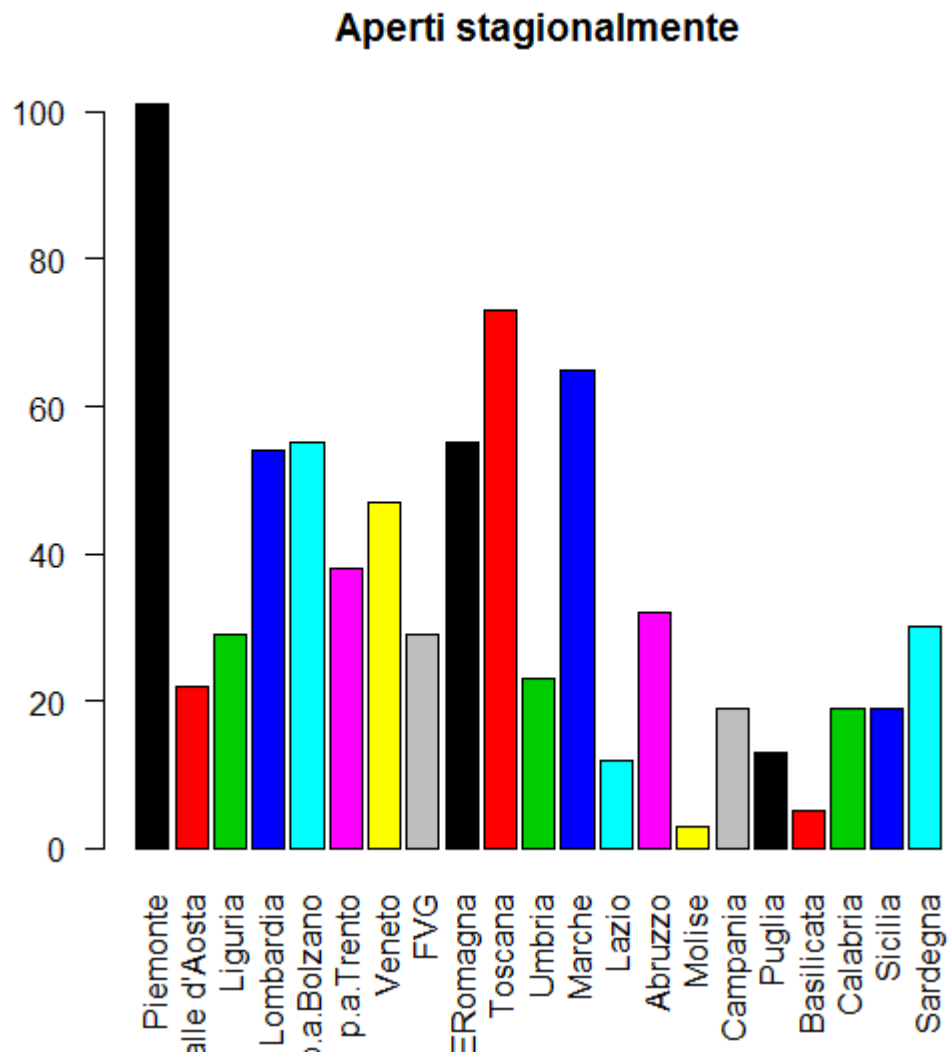
prima colonna - tutto l'anno

```
> tuttoAnno<-pApertura[ ,1]
> tuttoAnno
> barplot (tuttoAnno , main="Aperti tutto l'anno", col =1:20, las=2)
```



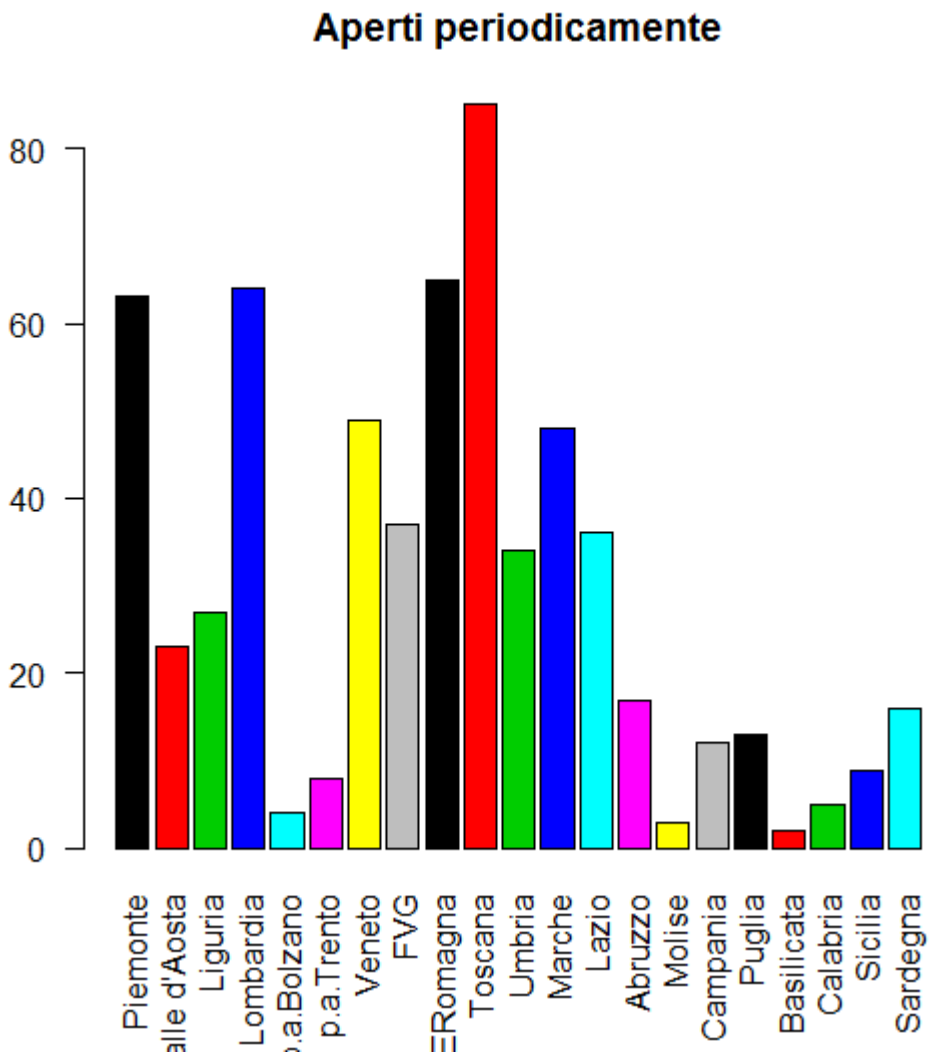
seconda colonna - stagionalmente

```
> stag<-pApertura[ ,2]
> stag
> barplot(stag , main="Aperti stagionalmente", col =1:22, las=2)
```



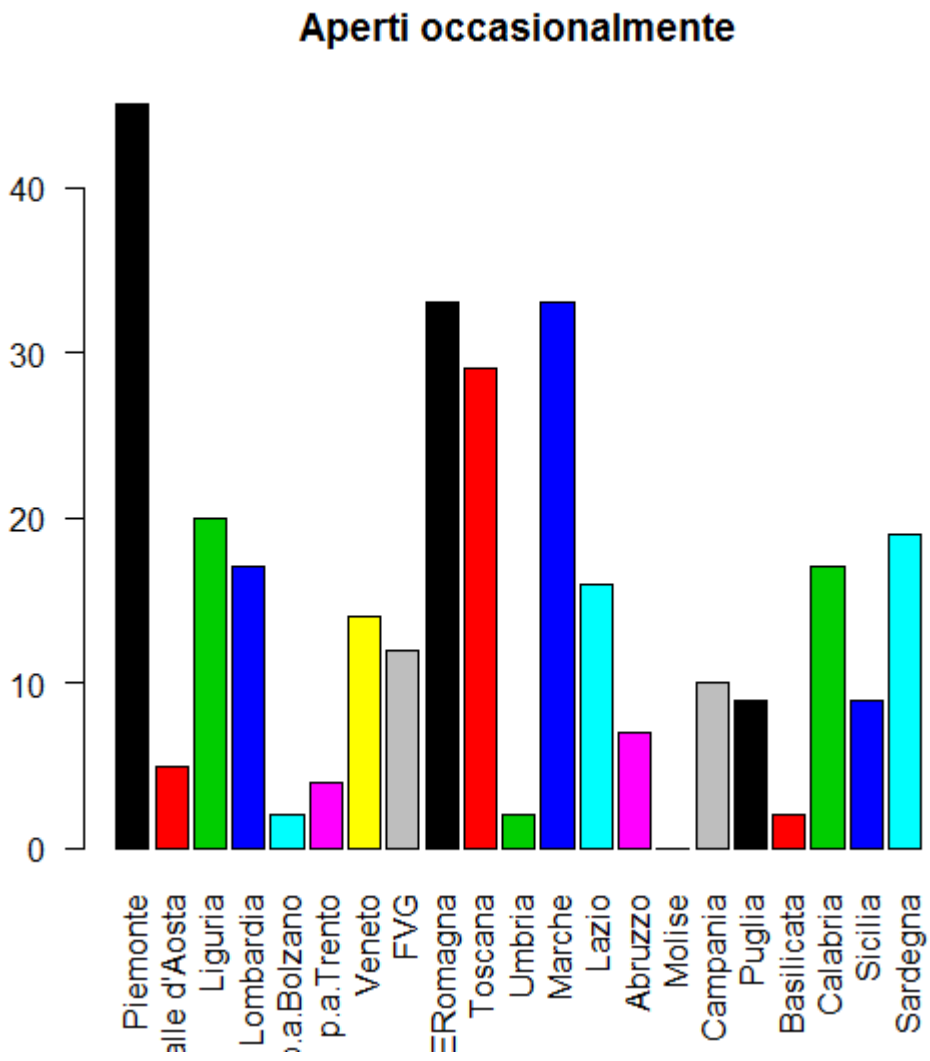
terza colonna - periodicamente

```
> period<-pApertura[ ,3]
> period
>  barplot(period, main="Aperti periodicamente", col =1:22, las=2)
```



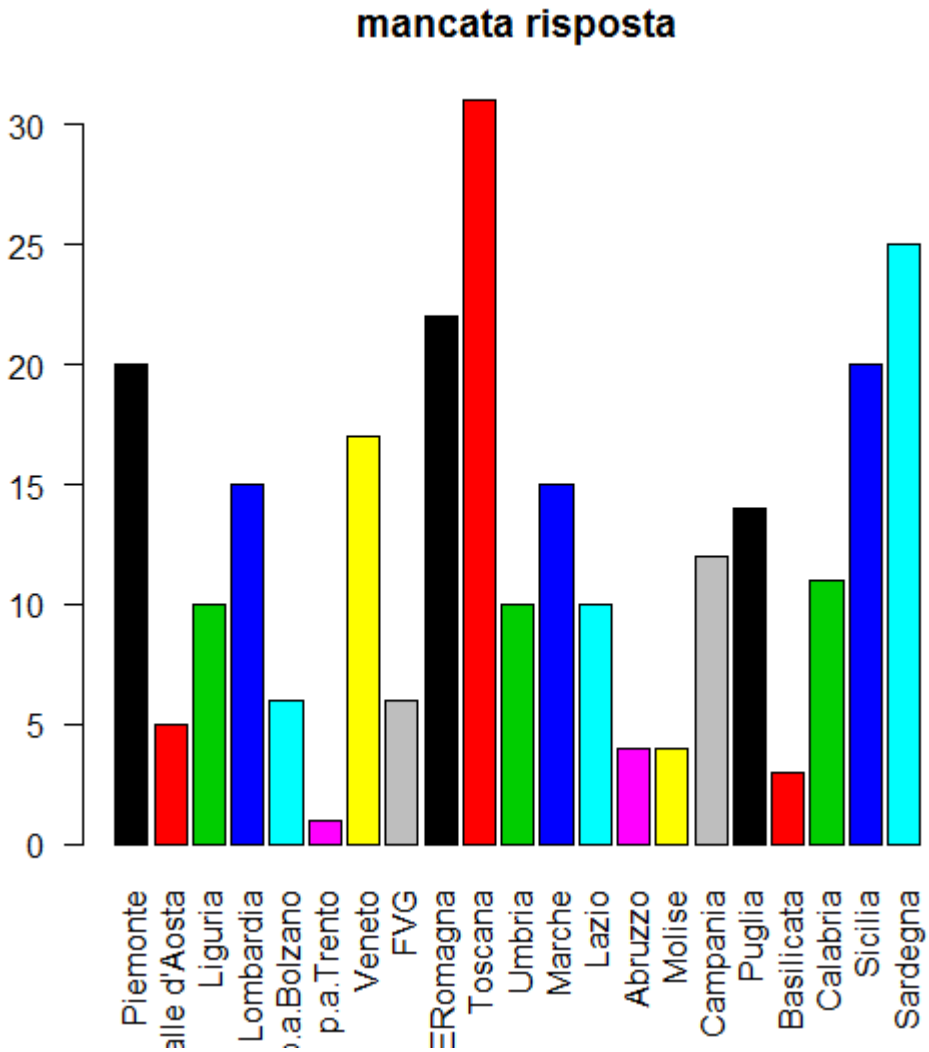
quarta colonna - occasionalmente

```
> occas<-pApertura[ ,4]
> occas
>  barplot(occas, main="Aperti occasionalmente", col =1:22, las=2)
```



quinta colonna – mancata risposta

```
> mancataRisp<-pApertura[ ,5]
> mancataRisp
>  barplot(mancataRisp, main="mancata risposta", col =1:22, las=2)
```



Analizziamo i risultati ottenuti:

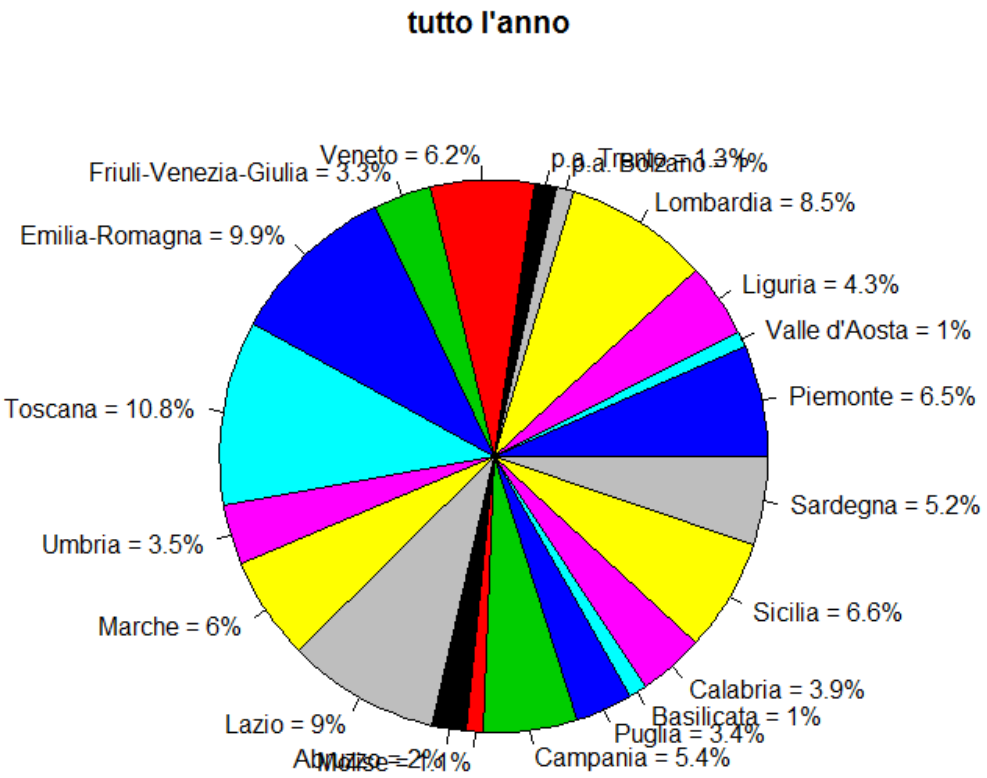
1. la regione che ha il maggior numero di musei aperti per *tutto l'anno* e *periodicamente* risulta essere la *Toscana*
2. la regione che ha il maggior numero di musei aperti *stagionalmente* e *occasionalmente* risulta essere il *Piemonte*
3. la regione che ha il maggior numero di musei i quali non hanno dato *nessuna risposta* è la *Toscana*

a. la regione che ha, invece, il minor numero di musei aperti tutto l'anno è la Valle d'Aosta
b. la regione che ha il minor numero di musei aperti stagionalmente è il Molise
c. la regione che ha il minor numero di musei aperti periodicamente è la Basilicata
d. la regione che ha il minor numero di musei aperti occasionalmente è il Molise
e. la provincia autonoma di Trento ha il minor numero di mancate risposte da parte dei musei

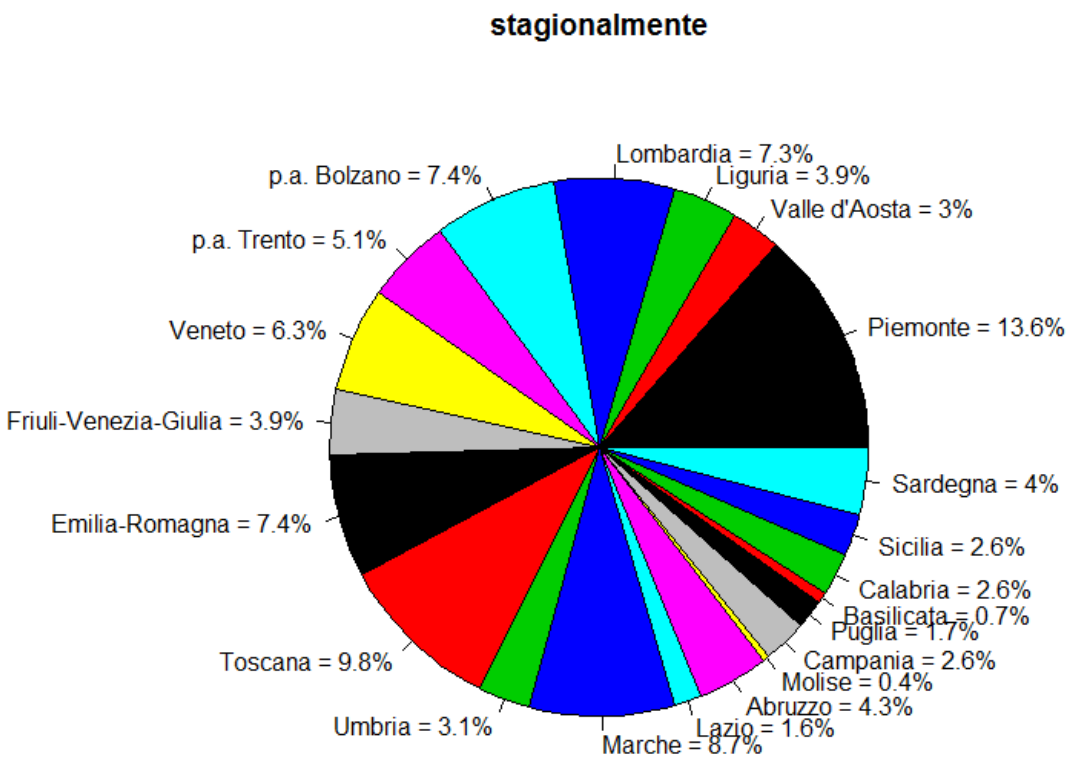
grafici a torta (per periodo)

un altro tipo di grafico è quello a torta. Un grafico a torta si costruisce tracciando un cerchio e suddividendolo in tanti settori circolari (fette o spicchi) quante sono le modalità distinte di dati. Tale grafico in R viene prodotto tramite il comando pie().

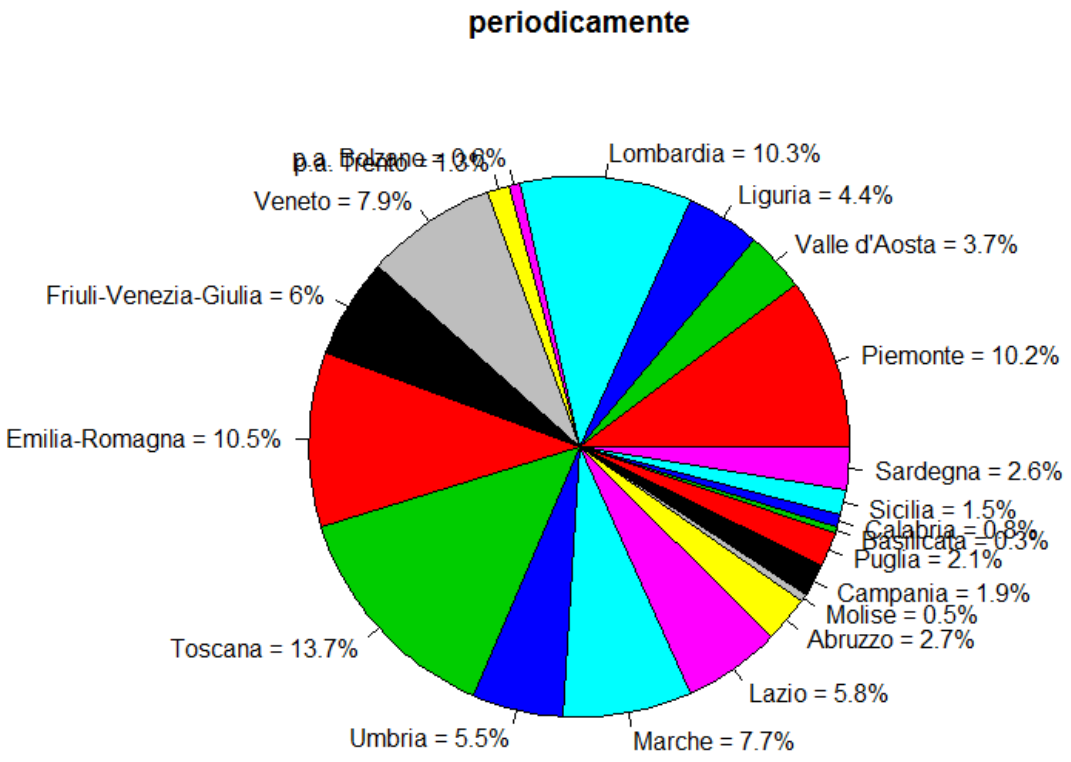
```
table1<-pApertura[ ,1]

labs<-c("Piemonte", "Valle d'Aosta", "Liguria", "Lombardia", "p.a. Bolzano", "p.a. Trento",
"Veneto", "Friuli-Venezia-Giulia", "Emilia-Romagna", "Toscana", "Umbria", "Marche",
"Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia",
"Sardegna")

pct<-round((((table1)/3047)*100),1)

eti<-paste(labs," = ",pct, "%", sep="")

pie(table1, main="tutto l'anno",labels=eti, col=10:49)
```



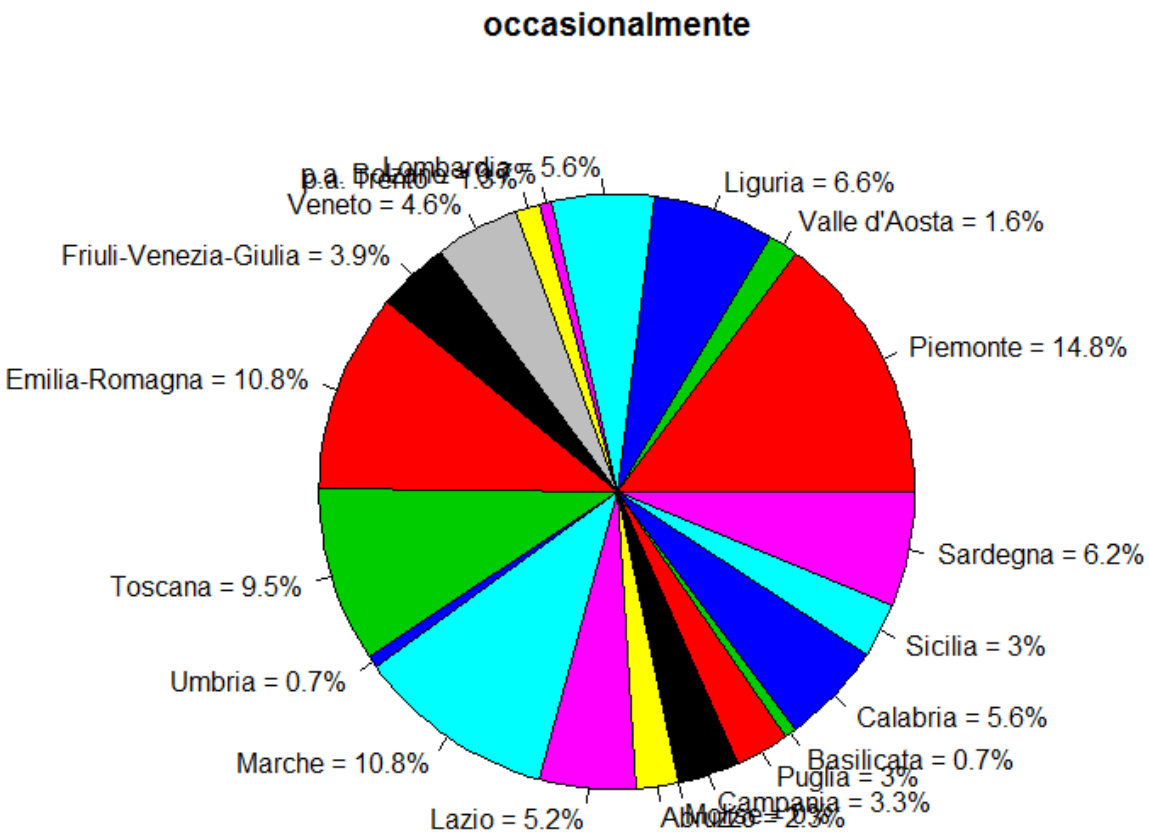
```
table1<-pApertura[ ,2]

labs<-c("Piemonte", "Valle d'Aosta", "Liguria", "Lombardia", "p.a. Bolzano", "p.a. Trento",
"Veneto", "Friuli-Venezia-Giulia", "Emilia-Romagna", "Toscana", "Umbria", "Marche",
"Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia",
"Sardegna")

pct<-round((((table1)/743)*100),1)

eti<-paste(labs," = ",pct, "%", sep="")

pie(table1, main="stagionalmente",labels=eti, col=10:49)
```



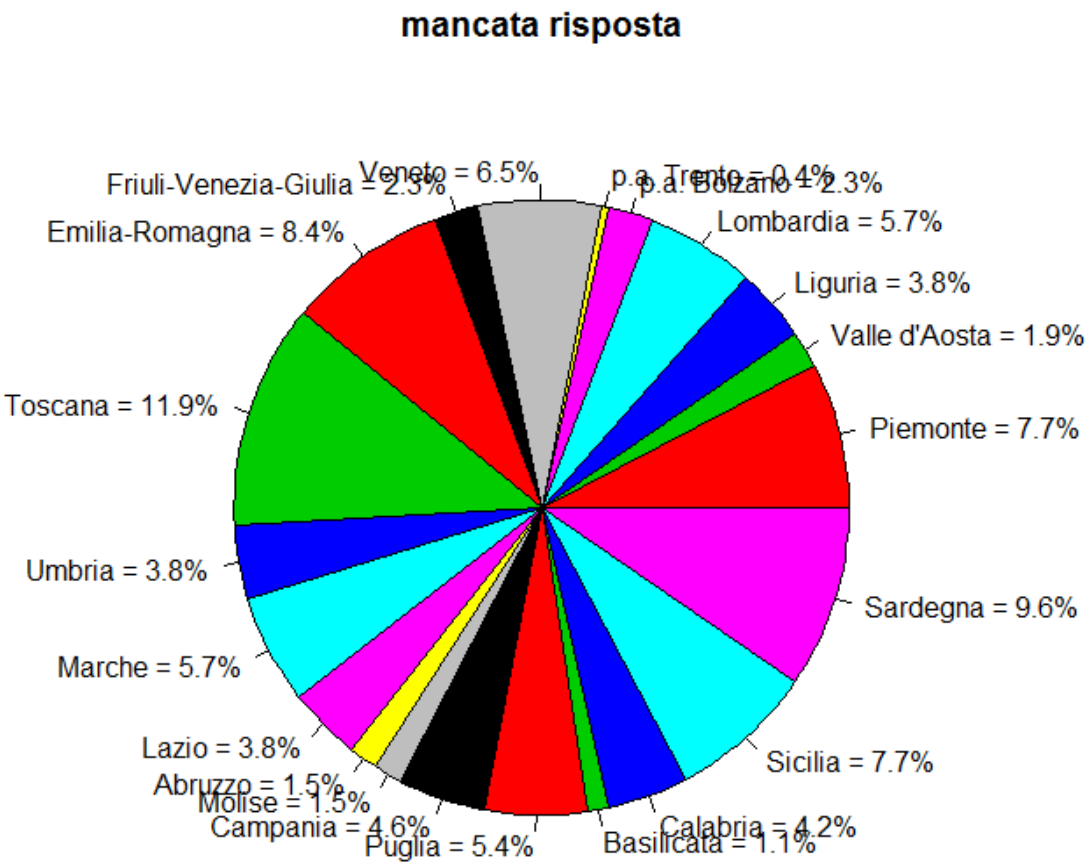
```
table1<-pApertura[ ,3]

labs<-c("Piemonte", "Valle d'Aosta", "Liguria", "Lombardia", "p.a. Bolzano", "p.a. Trento",
"Veneto", "Friuli-Venezia-Giulia", "Emilia-Romagna", "Toscana", "Umbria", "Marche",
"Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia",
"Sardegna")

pct<-round((((table1)/620)*100),1)

eti<-paste(labs," = ",pct, "%", sep="")

pie(table1, main="periodicamente",labels=eti, col=10:49)
```



```
table1<-pApertura[ ,4]

labs<-c("Piemonte", "Valle d'Aosta", "Liguria", "Lombardia", "p.a. Bolzano", "p.a. Trento",
"Veneto", "Friuli-Venezia-Giulia", "Emilia-Romagna", "Toscana", "Umbria", "Marche",
"Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia",
"Sardegna")

pct<-round((((table1)/305)*100),1)

eti<-paste(labs," = ",pct, "%", sep="")

pie(table1, main="occasionalmente",labels=eti, col=10:49)
```



```
table1<-pApertura[ ,5]

labs<-c("Piemonte", "Valle d'Aosta", "Liguria", "Lombardia", "p.a. Bolzano", "p.a. Trento",
"Veneto", "Friuli-Venezia-Giulia", "Emilia-Romagna", "Toscana", "Umbria", "Marche",
"Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia",
"Sardegna")

pct<-round((((table1)/261)*100),1)

eti<-paste(labs," = ",pct, "%", sep="")

pie(table1, main="mancata risposta",labels=eti, col=10:49)
```



Notiamo che per i grafici a torta abbiamo gli stessi risultati che abbiamo avuto per i grafici a barre.

Diagramma di Pareto

Il principio di Pareto afferma che *la maggior parte degli effetti è dovuta ad un numero ristretto di cause (considerando grandi numeri).* Tale osservazione ispirò la cosiddetta "legge 80/20" secondo la quale l'80% dei risultati dipende dal 20% delle cause.

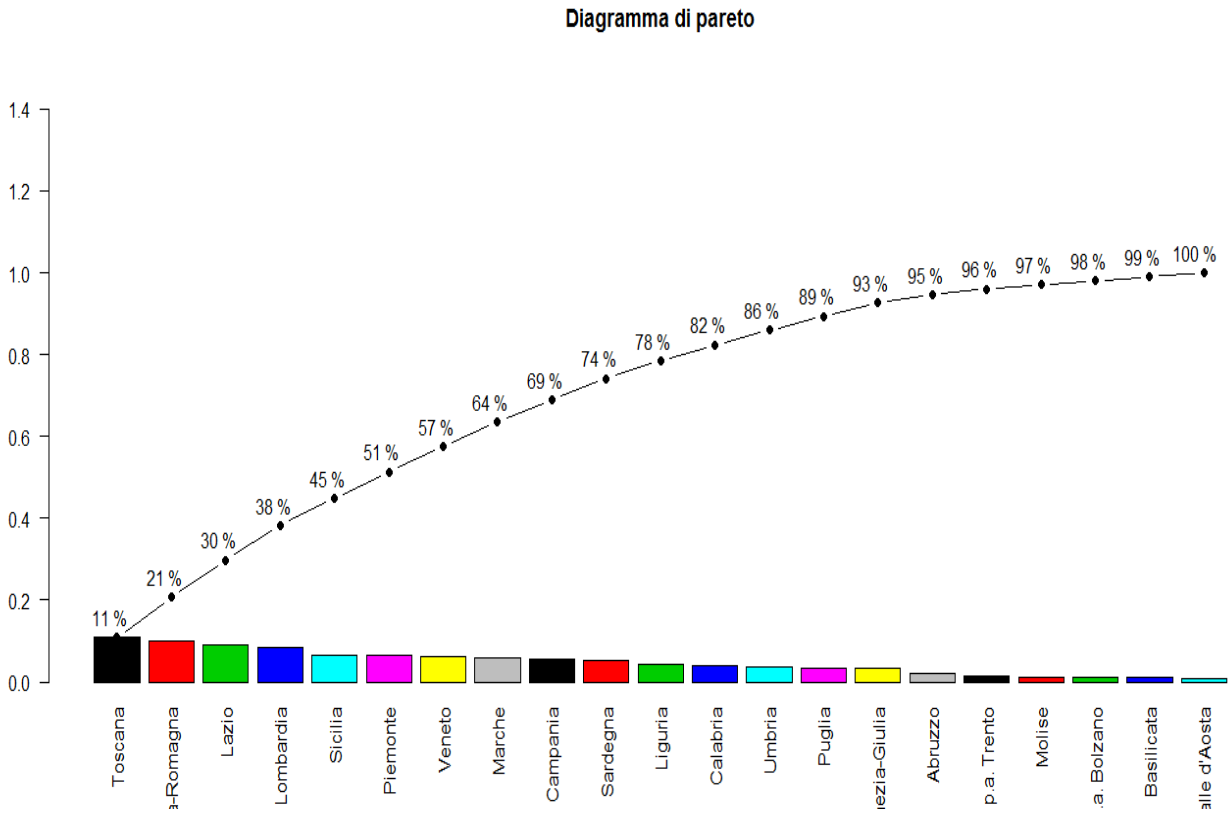
Il diagramma di Pareto consiste di un diagramma a barre verticali con le modalità ordinate in modo decrescente rispetto alla frequenza relativa; inoltre le frequenze sono visualizzate anche nella loro forma cumulata, mediante una sequenza di segmenti crescenti.

Costruiamo, quindi, il diagramma di Pareto sul periodo di apertura '*tutto l'anno*':

Le fasi per costruire il diagramma di Pareto sono le seguenti:

1. decidere come classificare i dati;
2. rilevare i dati e ordinarli;
3. disegnare il diagramma;
4. costruire la linea cumulativa;
5. aggiungere le informazioni di base.

```
a<-pApertura[ ,1]
b<-sum(a)
c<-a/b
ord<-sort(c, decreasing=TRUE)
x<-barplot(ord, ylim=c(0,1.5), main="Diagramma di pareto", col=1:21, las=2)
lines(x,cumsum(ord),type="b", pch=16)
text(x-0.2, cumsum(ord)+0.05, paste(format(cumsum(ord)*100, digits=2), "%"))
```



Da tale diagramma di Pareto notiamo che fino alla Calabria l'80% dei musei è aperto maggiormente, mentre il restante 20% ha un numero di musei aperti meno significativo. Quindi, *bisogna incentivare le regioni che fanno parte di quel 20%.*

Istogrammi

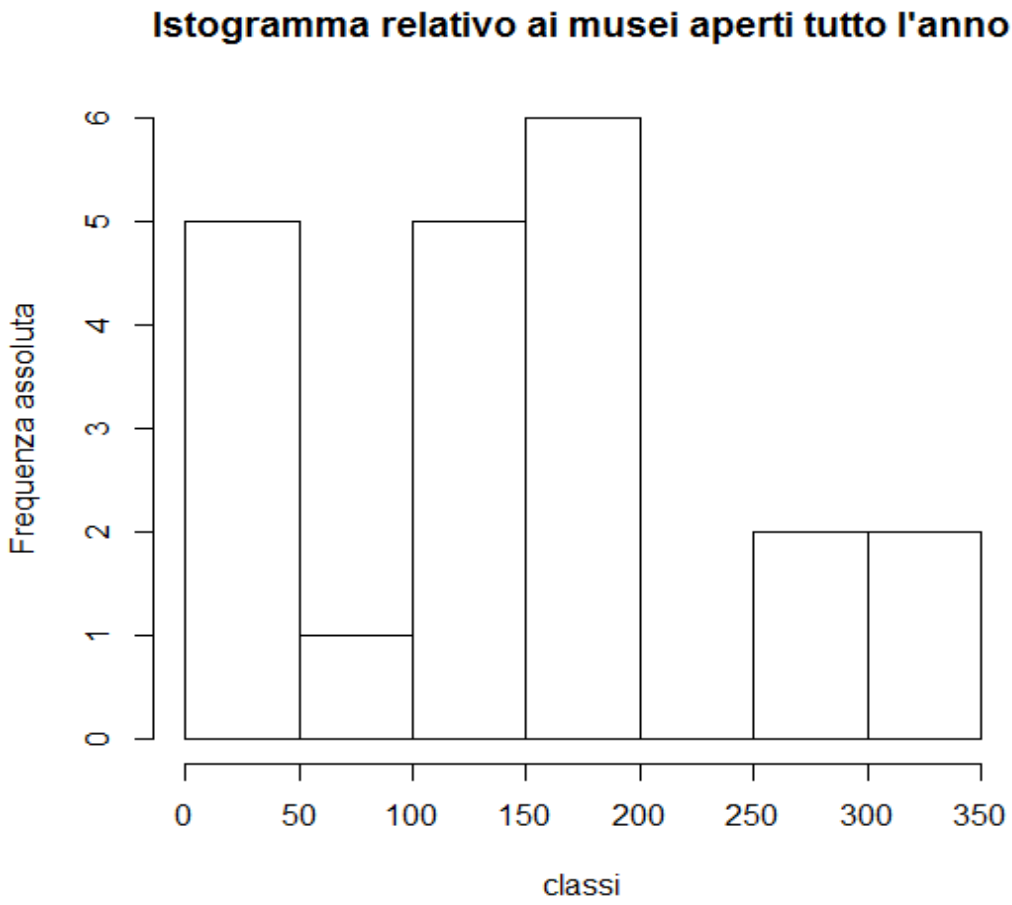
Gli istogrammi, che si utilizzano per variabili quantitative, sono una particolare rappresentazione grafica di una distribuzione di frequenza in classi. Consideriamo un campione $(x_1,..., x_n)$ costituito da n osservazioni, che suddividiamo in classi; ogni osservazione deve cadere in una ed una sola classe (o intervallo). Gli istogrammi sono una rappresentazione grafica ottenuta mediante rettangoli adiacenti aventi per basi segmenti i cui estremi corrispondono agli estremi delle classi. Fissate le basi, le altezze devono essere tali che l'area di ogni rettangolo risultante sia uguale alla frequenza (relativa o assoluta) della classe stessa.

Se si utilizzano le frequenze assolute delle classi, l'area di ogni rettangolo è uguale alla frequenza assoluta della classe e l'area totale dei rettangoli è uguale all'ampiezza del campione.

In R, per l'istogramma viene fatta una divisione in classi automatica. Sulle ordinate abbiamo le frequenze assolute.

Applichiamo, quindi, alla matrice pApertura la funzione hist:

```
> h<-hist(pApertura[ ,1], freq=TRUE, main="Istogramma relativo ai musei aperti tutto l'anno", ylab="Frequenza assoluta", xlab="classi")
```



Notiamo che 5 musei cadono nella prima classe, 1 nella seconda, 5 nella terza, 6 nella sesta, 0 nella settima e 2 nelle ultime due classi.

```
> str(h)
List of 6
 $ breaks  : num [1:8] 0 50 100 150 200 250 300 350
 $ counts  : int [1:7] 5 1 5 6 0 2 2
 $ density : num [1:7] 0.004762 0.000952 0.004762 0.005714 0 ...
 $ mids    : num [1:7] 25 75 125 175 225 275 325
 $ xname   : chr "pApertura[, 1]"
 $ equidist: logi TRUE
 - attr(*, "class")= chr "histogram"
```

Breaks fornisce i punti di suddivisione in classi;

counts le frequenze assolute delle classi;

density la densità delle classi;

mids fornisce i punti centrali delle classi.

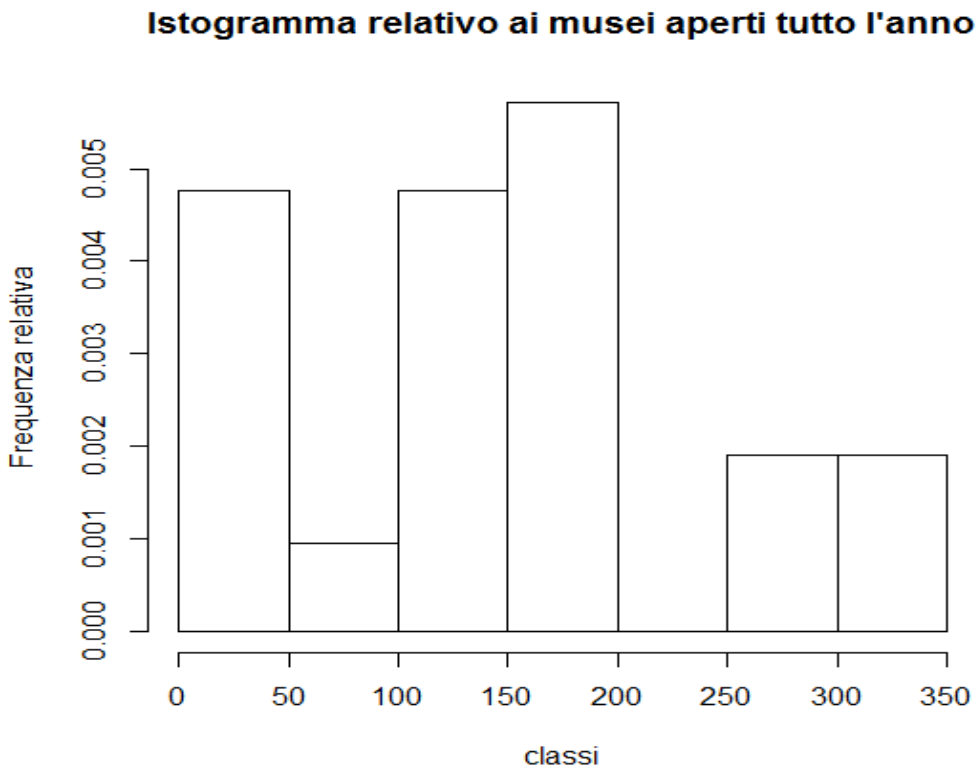
Per ottenere le frequenze relative moltiplichiamo l'ampiezza delle classi (50) per h$density

```
> f<-50*h$density
> f
[1] 0.23809524 0.04761905 0.23809524 0.28571429 0.00000000 0.09523810 0.09523810
> sum(f)
[1] 1
```

Infatti, se sommiamo le frequenze relative ci aspettiamo che sia proprio 1 il risultato.

Ora, considerando sempre la colonna relativa al numero di musei aperti tutto l'anno, calcoliamo l'istogramma con le frequenze relative:

```
> h<-hist(pApertura[ ,1], freq=FALSE, main="Istogramma relativo ai musei aperti tutto l'anno", ylab="Frequenza relativa", xlab="classi")
```



Boxplot

I boxplot, o scatola con baffi, è, appunto, il disegno di una scatola i cui estremi sono Q1 e Q3, tagliata da una linea orizzontale in corrispondenza di Q2, ossia della mediana. In basso e in alto sono presenti due linee orizzontali, dette baffi. Il baffo inferiore corrisponde al valore più piccolo tra le osservazioni che risulta maggiore o uguale di $Q1 - 1.5 \cdot (Q3 - Q1)$, mentre il baffo superiore corrisponde al valore più grande delle osservazioni che risulta minore o uguale a $Q3+1.5\cdot(Q3-Q1)$. Quindi, se tutti i dati rientrano nell'intervallo $(Q1 - 1.5 \cdot (Q3 - Q1), Q3 + 1.5 \cdot (Q3 - Q1))$ i baffi sono posti in corrispondenza del minimo valore e del massimo valore del campione. Gli eventuali valori al di questo intervallo sono visualizzati nel grafico sotto forma di punti, detti valori anomali o outlier. Questi valori infatti costituiscono una "anomalia" rispetto alla maggior parte dei valori osservati.

Il boxplot viene utilizzato per illustrare alcune caratteristiche di una distribuzione di frequenza: la centralità, la forma, la dispersione e la presenza di eventuali valori anomali.

La centralità è espressa dalla mediana.

I baffi, superiore e inferiore, forniscono informazioni sulla dispersione e sulla forma della distribuzione ed anche sulle code della distribuzione. Infatti, la dispersione è deducibile esaminando le distanze del baffo superiore da Q3 e del baffo inferiore da Q1.

Nella nostra analisi costruiamo 5 boxplot, uno per ogni periodo di apertura e ne studiamo le caratteristiche.

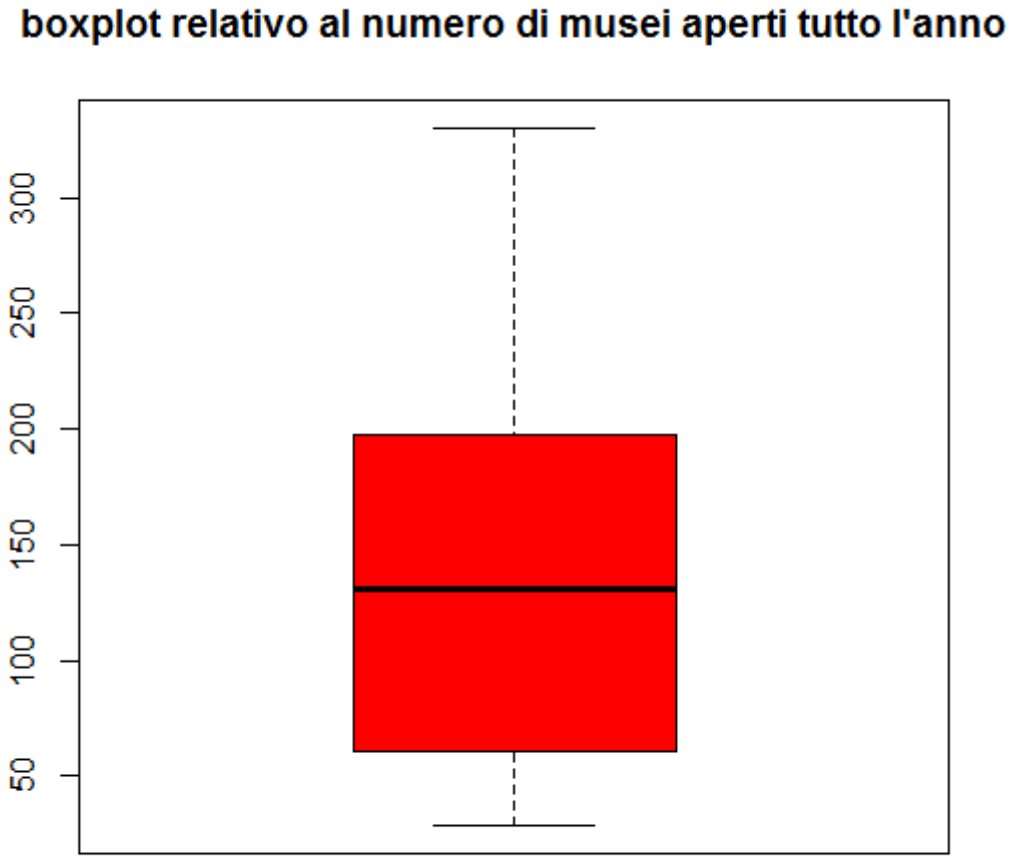
```
> a<-pApertura[ ,1]
> quantile(a)
 0%  25%  50%  75% 100%
 29   61   131   198   330
```

Tramite il comando summary vediamo in dettaglio quali sono i valori di Q1, Q2, Q3, min, max.

```
> summary(a)
 Min.   1st Qu.  Median    Mean    3rd Qu.    Max.
 29.0    61.0     131.0     145.1    198.0     330.0
```

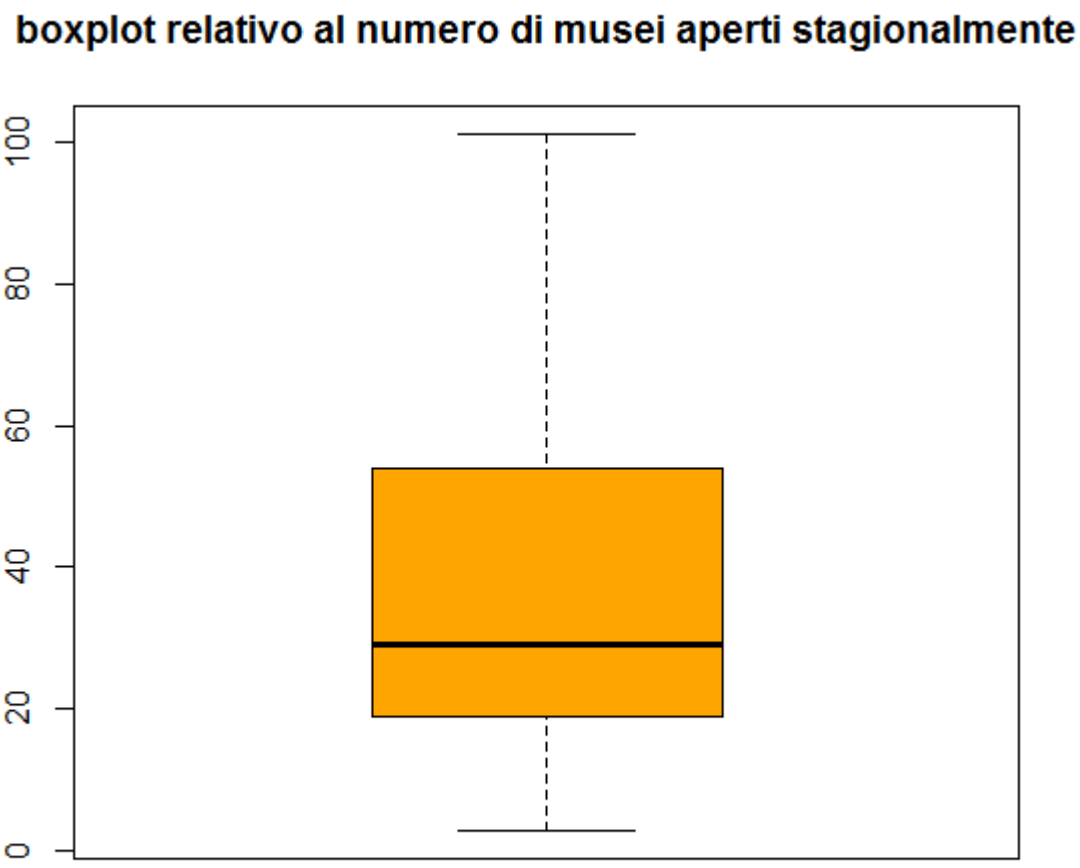
Disegniamo il boxplot:

```
> boxplot(a, main="boxplot relativo al numero di musei aperti tutto l'anno",col="red")
```



Possiamo notare che in questo boxplot sono ben evidenti sia il baffo superiore e sia quello inferiore. Abbiamo una buona centralità dei dati. Un altro dato che possiamo notare è che c'è abbastanza simmetria, in quanto le distanze del primo e del terzo quartile dalla mediana sono pressoché uguali (anche se media e mediana sono diverse). C'è abbastanza dispersione dei dati, poiché la distanza tra il baffo inferiore e il primo quartile, e quello superiore e il terzo quartile non è ben proporzionata, infatti notiamo una dispersione verso l'alto data l'elevata distanza tra il baffo superiore e il terzo quantile. Non ci sono valori anomali, in quanto ogni valore ricade nell'intervallo (-144.5, 403.5)

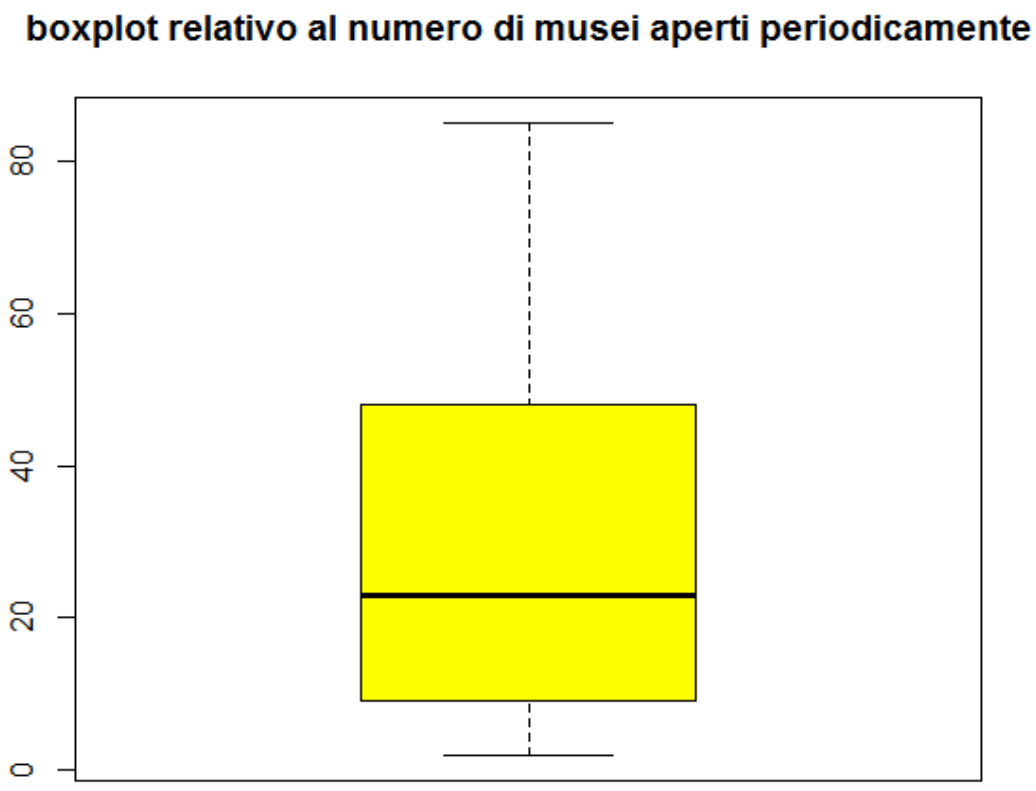
```
> a<-pApertura[ ,2]
> quantile(a)
  0%  25%  50%  75% 100%
   3   19   29   54  101
> summary(a)
   Min. 1st Qu.  Median    Mean   3rd Qu.    Max.
  3.00   19.00   29.00   35.38   54.00   101.00
> boxplot(a, main="boxplot relativo al numero di musei aperti
stagionalmente",col="orange")
```



Di questo boxplot possiamo dire subito che c'è poca simmetria, poiché la distanza tra mediana e terzo quartile è molto più grande di quella tra mediana e primo quartile. I baffi

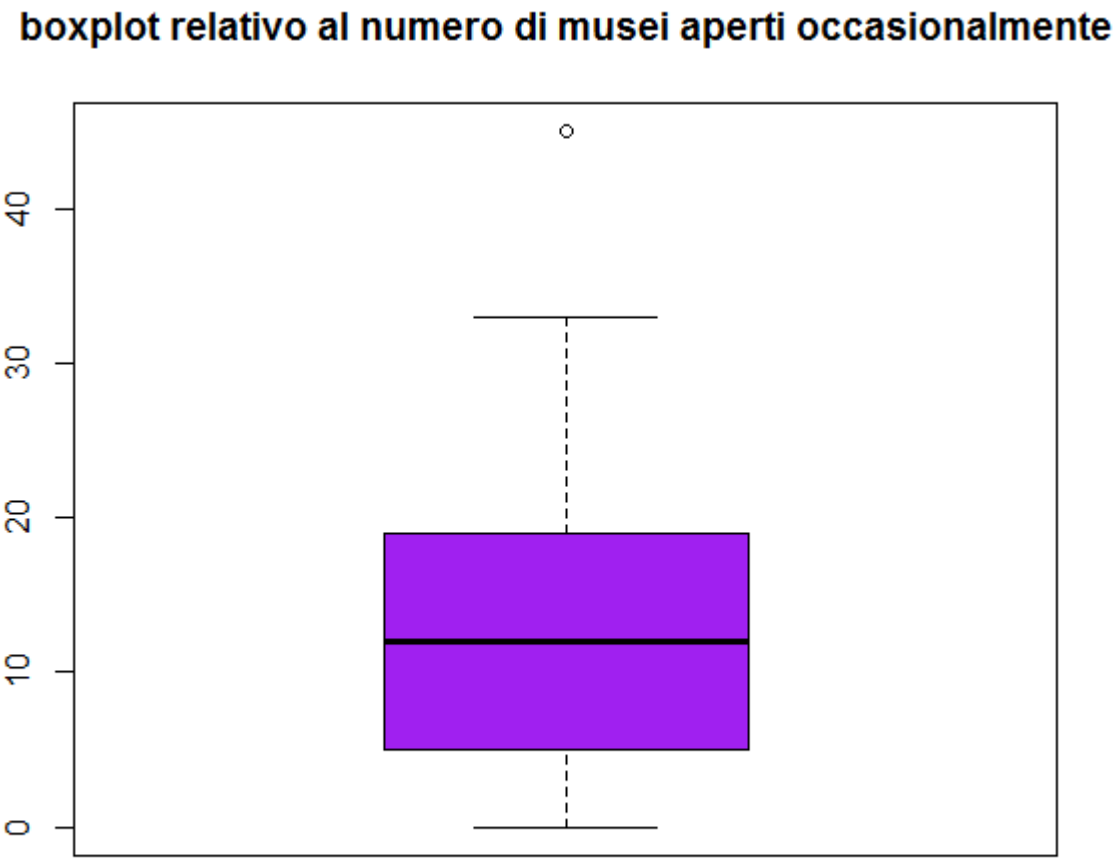
sono entrambi ben visibili. Rispetto al grafico precedente la dispersione dei dati è più o meno uguale. La centralità dei dati, invece, è bassa: la media è maggiore della mediana e quindi vediamo che i dati sono distribuiti verso destra. Non ci sono valori anomali poiché tutti i valori ricadono nell'intervallo (-33.5, 106.5).

```
> a<-pApertura[ ,3]
> quantile(a)
  0%  25%  50%  75% 100%
   2    9   23   48   85
> summary(a)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    9.00   23.00   29.52   48.00   85.00
> boxplot(a, main="boxplot relativo al numero di musei aperti periodicamente",col="yellow")
```



Questa situazione è molto simile alle precedenti: abbiamo una dispersione maggiore nella parte alta della scatola. L'intervallo in cui risiedono i valori è (-49.5, 106.5), quindi non ci sono valori anomali.

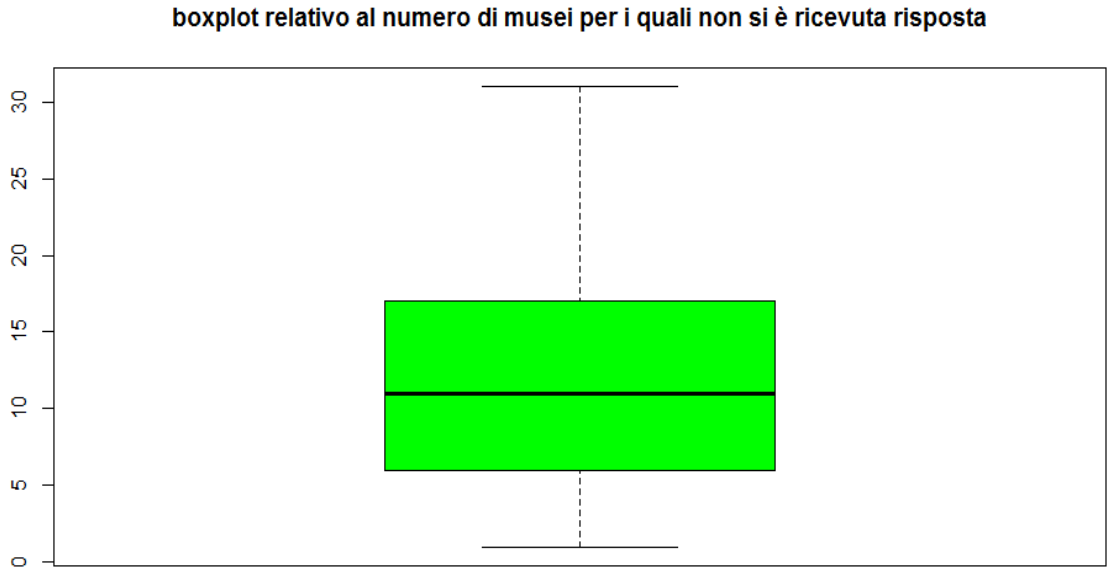
```
> a<-pApertura[ ,4]
> quantile(a)
  0%  25%  50%  75% 100%
   0    5   12   19   45
> summary(a)
   Min.  1st Qu.  Median    Mean  3rd Qu.    Max.
   0.00    5.00   12.00   14.52   19.00   45.00
> boxplot(a, main="boxplot relativo al numero di musei aperti
occasionalmente",col="purple")
```



In questo caso è presente un'anomalia, ciò significa che c'è una regione che ha un numero molto maggiore di musei aperti occasionalmente. Notiamo che per questo periodo c'è una

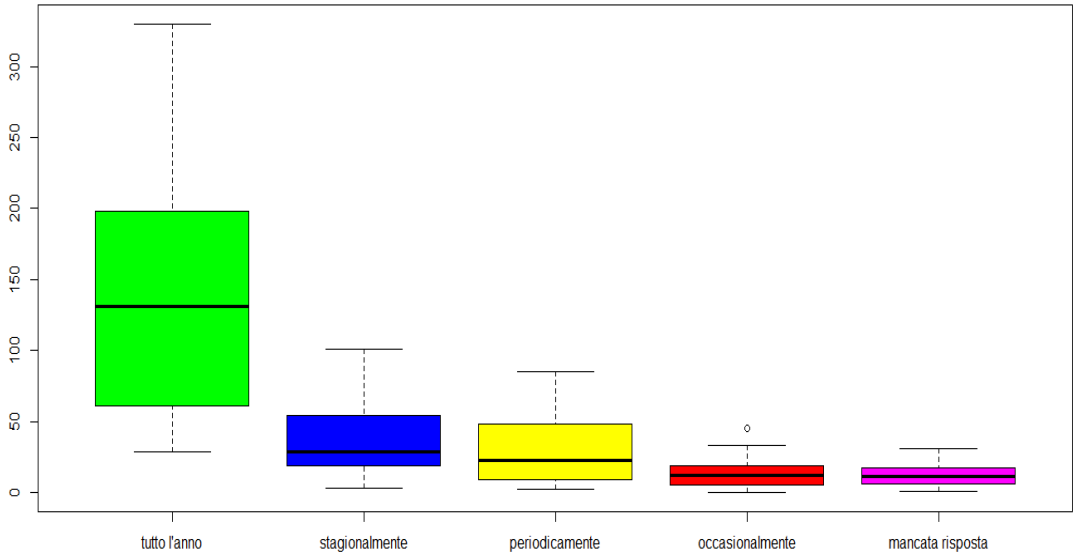
regione, il Piemonte che ha un numero molto più elevato di musei aperti occasionalmente rispetto alle altre regioni, e questo produce un'anomalia. L'intervallo è (-16, 40), il valore della regione Piemonte è 45 che non rientra nell'intervallo.

```
> a<-pApertura[ ,5]
> quantile(a)
  0%  25%  50%  75% 100%
   1    6   11   17   31
> summary(a)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    6.00   11.00   12.43   17.00   31.00
> boxplot(a, main="boxplot relativo al numero di musei per i quali non si è ricevuta risposta",col="green")
```



In quest'ultimo caso, abbiamo un'altissima centralità (infatti la mediana è molto vicina alla media), di conseguenza abbiamo una buona simmetria. Per quanto riguarda la dispersione siamo nella situazione simile ai casi precedenti. La mancanza delle anomalie è dovuta ancora una volta al fatto che i valori sono compresi nell'intervallo (-10.5, 33.5).

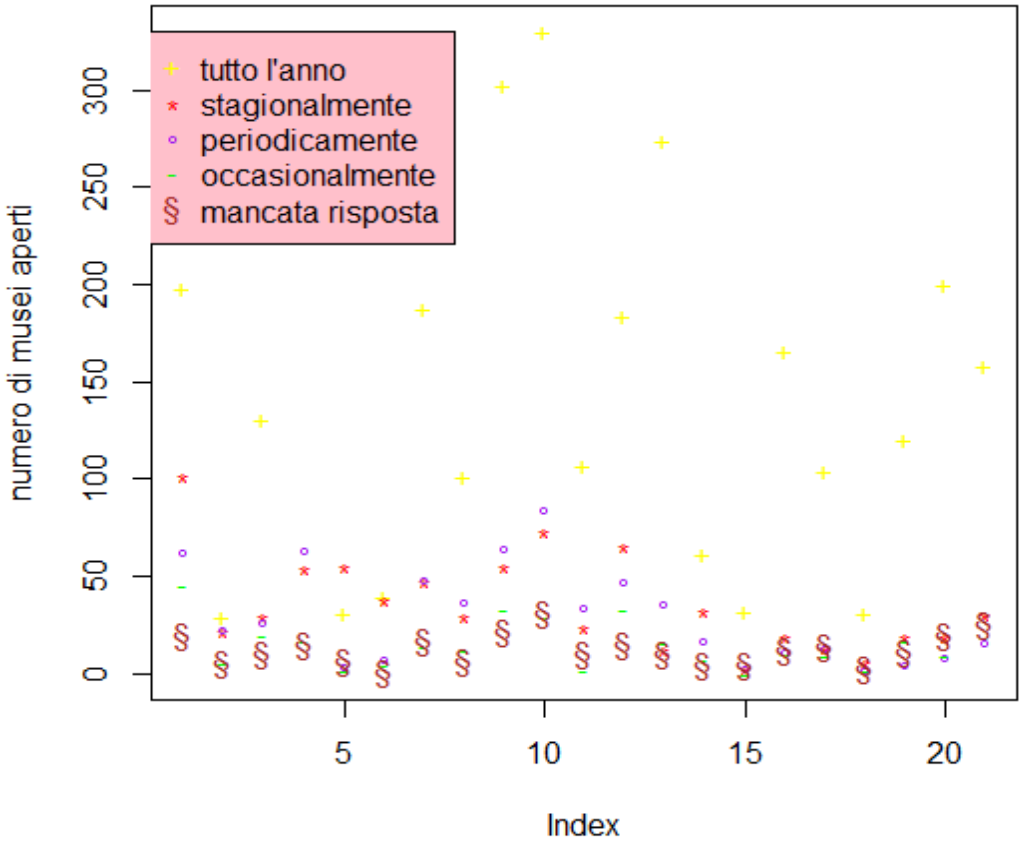
Ora confrontiamo, in un unico grafico, i 5 boxplot che abbiamo ottenuto in precedenza:



Notiamo a prima vista che il numero di musei aperti tutto l'anno supera di molto le aperture negli altri periodi. Dai precedenti risultati del comando summary notiamo la stessa cosa.

Ora facciamo un grafico congiunto con i dati a nostra disposizione:

```
plot(tuttoAnno, pch="+", ylim=c(0,330), ylab="numero di musei aperti", col="yellow")
points(pApertura[ ,2], pch="*", col="red")
points(pApertura[ ,3], pch="°", col="purple")
points(pApertura[ ,4], pch="-", col="green")
points(pApertura[ ,5], pch="§", col="brown")
legend(0,330, c("tutto l'anno", "stagionalmente", "periodicamente", "occasionalmente", "mancata risposta"), pch=c("+", "*", "°", "-", "§"), col=c("yellow", "red", "purple", "green", "brown"), bg="pink", cex=1)
```



Notiamo che il numero di musei aperti tutto l'anno (simboli gialli) è maggiore rispetto agli altri periodi (gli altri simboli al massimo arrivano in corrispondenza del 100).

Scatterplot

Siano X e Y due variabili di tipo quantitativo e indichiamo con $z_1,\ldots, z_h$ e $w_1,\ldots, w_k$ i valori distinti da esse assunti.

Consideriamo un campione $\{(x_1, y_1),\ldots,(x_n, y_n)\}$ costituito da n osservazioni di (X,Y).

Gli scatterplot mettono in evidenza le relazioni tra le variabili e il tipo di relazione. Il grafico che si ottiene mira a evidenziare se le coppie di punti presentano qualche forma di regolarità. Sull'asse delle ascisse c'è la variabile indipendente e su quella delle ordinate c'è la variabile dipendente. Tale grafico dà come risultato finale una nuvola di punti.

Per prima cosa costruiamo il nostro data frame:

```
df<-
data.frame(tuttoAnno=c(198,29,131,259,31,40,188,101,303,330,107,184,274,61,32,166,104,31,120,200,158),

stag=c(101,22,29,54,55,38,47,29,55,73,23,65,12,32,3,19,13,5,19,19 ,30),

period=c(63,23,27,64,4,8,49,37,65,85,34,48,36,17,3,12,13,2,5,9,16),

occas=c(45,5,20,17,2,4,14,12,33,29,2,33,16,7,0,10,9,2,17,9,19),

mancataRisp=c(20,5,10,15,6,1,17,6,22,31,10,15,10,4,4,12,14,3,11,20,25))

> df
```

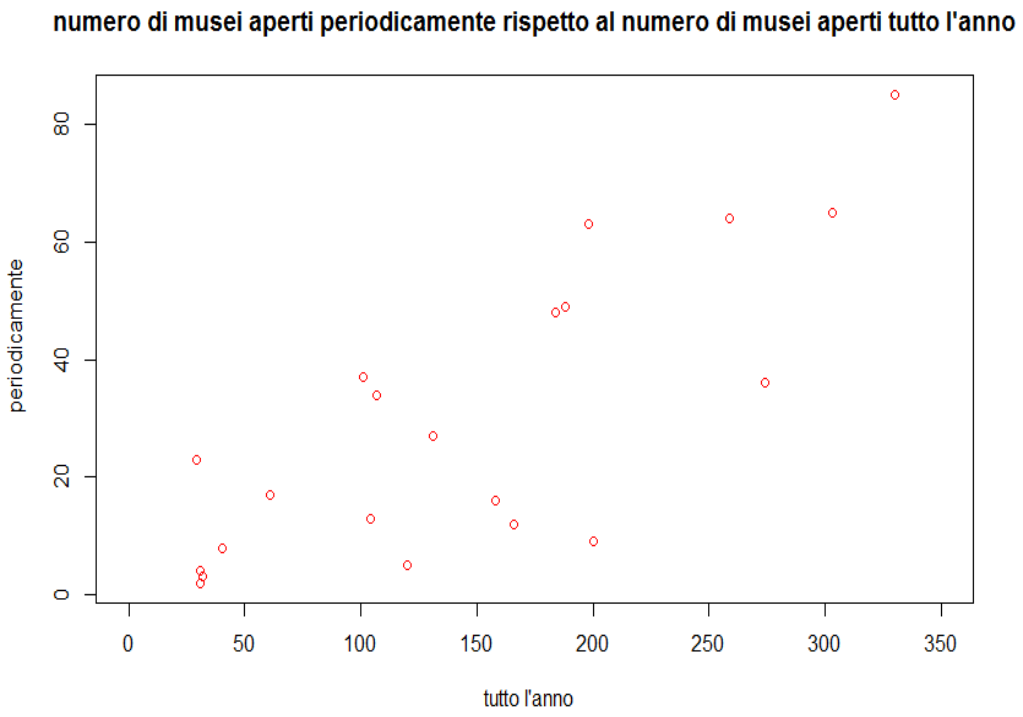
	tuttoAnno	stag	period	occas	mancataRisp
Piemonte	198	101	63	45	20
Valle d'Aosta	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
p.a. Bolzano	31	55	4	2	6
p.a. Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia-Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

Vediamo qual è la relazione tra queste due variabili:

tutto l'anno: variabile indipendente;

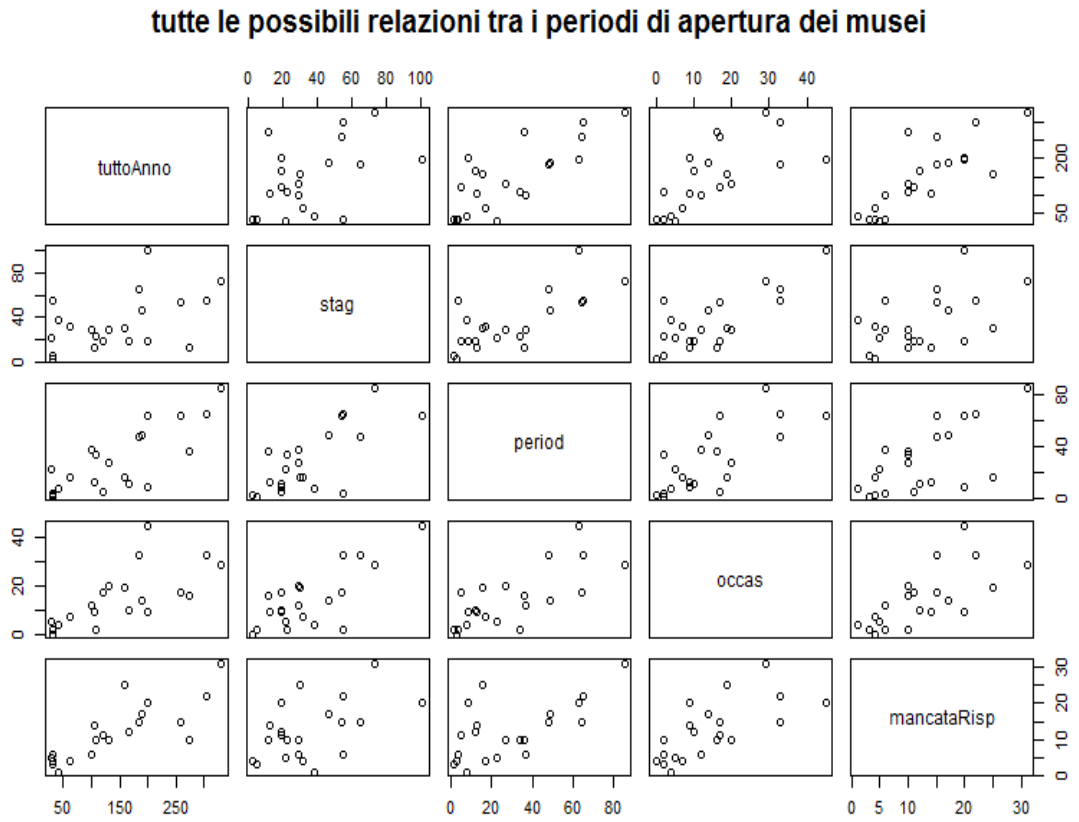
periodicamente: variabile dipendente.

```
> plot(df$tuttoAnno ,df$period , main =" numero di musei aperti periodicamente rispetto al numero di musei aperti tutto l'anno ",  xlab="tutto l'anno  ",ylab=" periodicamente", xlim=c(0 ,350) ,col ="red ")
```



Ora, mostriamo tutte le possibili relazioni tra i periodi di apertura dei musei, tramite il comando pairs():

```
> pairs(df, main="tutte le possibili relazioni tra i periodi di apertura dei musei")
```



Per vedere se c'è una qualche correlazione tra due variabili, utilizziamo il comando cor.

```
> cor(df$tuttoAnno, df$mancataRisp)
[1] 0.8140978
> cor(df$tuttoAnno, df$period)
[1] 0.7894135
```

La relazione *tutto l'anno - mancata* risposta risulta essere quella maggiormente correlata. Anche *tutto l'anno - periodicamente* risulta avere una buona correlazione. Quindi quando tratteremo la statistica bivariata studieremo la regressione lineare di queste relazioni.

Statistica univariata

Per i fenomeni quantitativi è spesso utile definire la funzione di distribuzione empirica. Ci sono due tipi di funzione di distribuzione empirica: discreta e continua.

funzione di distribuzione empirica continua

Per fenomeni quantitativi continui la funzione di distribuzione empirica è una funzione continua. In particolare, se i dati sono raccolti in k distinte classi C_1, = $[z_1, z_2,)$, C_2, = $[z_2, z_3,)$,..., C_k,= $[z_k, z_{k+1})$, con z_1, < z_2, < ... z_k, < z_{k+1}, la funzione di distribuzione empirica è così definita:

$$F(x) = \begin{cases} 0, & x < z_1 \\ \dots & \\ F_i, & x = z_i \\ \dfrac{F_{i+1} - F_i}{z_{i+1} - z_i}\, x + \dfrac{z_{i+1} F_i - z_i F_{i+1}}{z_{i+1} - z_i}, & z_i < x < z_{i+1} \\ F_{i+1}, & x = z_{i+1} \\ \dots & \\ 1, & x \geq z_{k+1} \end{cases}$$

L'obiettivo è quello di calcolare le frequenze relative cumulate e poi tracciare il segmento che passa per i punti (z_i , F_i) e (z_{i+1} , F_{i+1}).

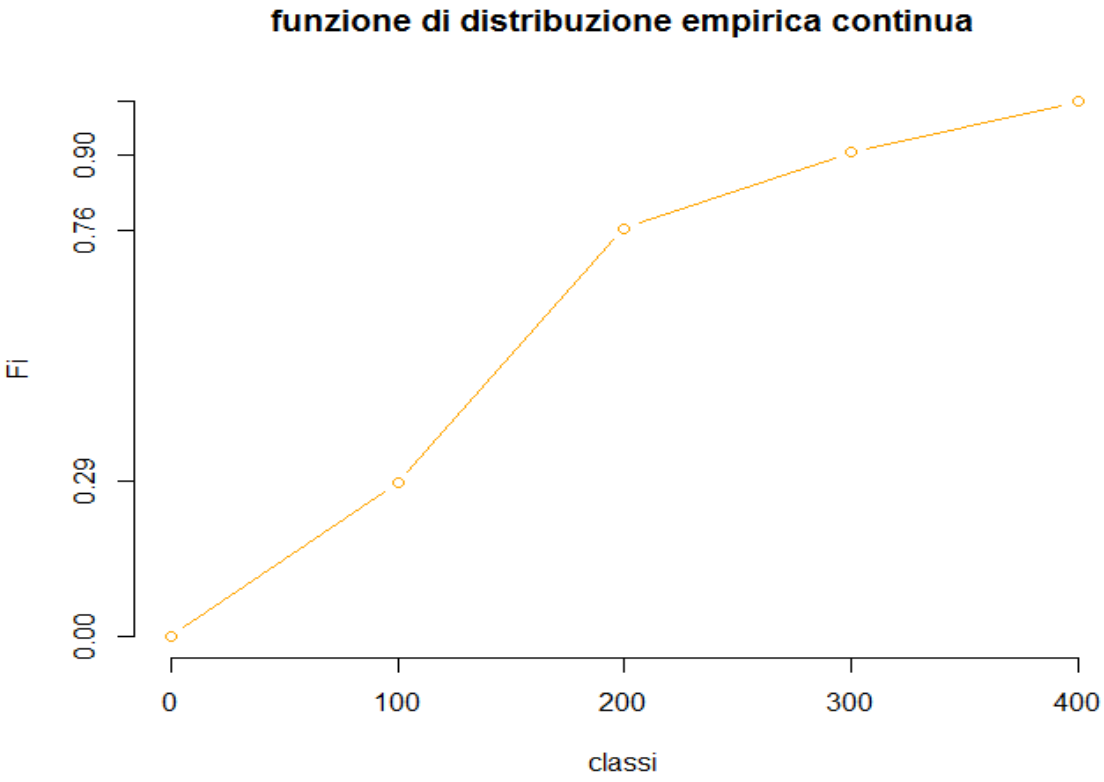
Nel nostro caso la suddivisione in classi è la seguente:

(0,100,200,300,400).

Calcoliamo le frequenze relative cumulate delle classi e disegniamo il grafico che congiunge i punti successivi mediante linee continue.

```
> classi<-c(0,100,200,300,400)
> Fi<-cumsum(table(cut(tuttoAnno, breaks=classi,right=FALSE)))/length(tuttoAnno)
> Fi<-c(0,Fi) #permette di aggiungere uno 0 all'inizio del vettore delle frequenze relative cumulate
> plot(classi, Fi, type="b", axes=FALSE, main="funzione di distribuzione empirica continua", col="orange")
> axis(1, classi)
```

```
> axis(2, format(Fi, digits=2))
> Fi
                  [0,100)     [100,200)     [200,300)    [300,400)
0.0000000    0.2857143    0.7619048    0.9047619    1.0000000
```



Notiamo che tra 100 e 200 abbiamo più elementi. Questa osservazione può essere vista anche negli istogrammi.

Indici di sintesi

indici di posizione

Tra gli indici di posizione abbiamo la media e la mediana campionarie.

La media campionaria $\bar{x}$ è definita come:

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

Assegnato un insieme di dati di ampiezza n, lo si ordini dal minore al maggiore. Se n è dispari, si definisce mediana campionaria il valore che è in posizione (n + 1)/2, mentre se n è pari la mediana campionaria è invece definita come la media aritmetica dei valori che occupano le posizioni n/2 e n/2 + 1.

Questi indici descrivono attorno a quali valori è centrato l'insieme dei dati.

Nel paragrafo relativo ai boxplot abbiamo calcolato la media e la mediana per ogni periodo di apertura usando il comando summary. Possiamo ottenere gli stessi risultati tramite i comandi mean e median. Con questi indici, quindi, possiamo vedere come sono distribuiti i dati: se la media è maggiore della mediana sono distribuiti verso destra, se la media è minore della mediana sono distribuiti verso sinistra. Nel caso in cui la media sia uguale alla mediana si ha la simmetria dei dati (nel nostro caso non abbiamo simmetria, però nel boxplot relativo a mancata risposta abbiamo quasi una simmetria, in quanto media e mediana sono molto vicine tra di loro). Nel nostro caso, si verifica sempre una distribuzione verso destra, in quanto la media è sempre maggiore della mediana. Questa osservazione è visibile graficamente nei boxplot.

Indici di dispersione

Questi indici misurano quanto si disperdono i dati rispetto alla media. Si chiamano varianza campionaria e deviazione standard campionaria.

Assegnato un insieme di dati numerici $x_1,\ldots, x_n$ la varianza campionaria è così definita:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2 \qquad (n = 2, 3, \ldots),$$

Inoltre, si definisce deviazione standard campionaria la radice quadrata della varianza campionaria, ossia:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \qquad (n = 2, 3, \ldots).$$

Entrambi questi indici sono tanto più grandi quanto più i dati si discostano dalla media.

Calcoliamo mediante i comandi *var* (periodo) e *sd* (periodo) tali indici:

periodo	tutto l'anno	stagionalmente	periodicamente	occasionalmente	Mancata risposta
Varianza	8740.99	606.7476	602.5619	145.6619	63.25714
Deviazione standard	93.49326	24.63225	24.54714	12.06905	7.953436

Ne risulta che per il periodo 'tutto l'anno' i dati si discostano maggiormente dalla media, mentre per 'mancata risposta' i dati di discostano di meno rispetto alla media.

E' utile calcolare anche il coefficiente di variazione standard per poter fare un confronto tra i vari dati, siccome essi non hanno lo stesso range di variazione, ossia differenti massimi e minimi dei dati.

Assegnato un insieme di dati numerici x_1,..., x_n si definisce coefficiente di variazione il rapporto tra la deviazione standard campionaria e il modulo della media campionaria, ossia:

$$CV = \frac{s}{|\overline{x}|}.$$

Questo indice è un numero adimensionale, poiché la media campionaria e la deviazione standard campionaria sono espresse in identiche unità di misura. Ha senso calcolare tale coefficiente se la media campionaria non è nulla. In R calcoliamo tale coefficiente tramite il comando cv(periodo)

periodo	Tutto l'anno	stagionalmente	periodicamente	occasionalmente	Mancata risposta
Cv	0.644	0.696	0.831	0.830	0.639

Differentemente da quanto potevamo immaginare, nonostante il valore della varianza più alto fosse quello relativo a tutto l'anno, con questo confronto dei dati, notiamo che la dispersione massima è quella di periodicamente e occasionalmente.

Un'altra misura che possiamo calcolare è la **skewness**, per vedere se la distribuzione di frequenza è asimmetrica positiva, negativa o simmetrica.

Assegnato un insieme di dati numerici x_1,..., x_n si definisce skewness campionaria il valore:

$$\gamma_1 = \frac{m_3}{m_2^{3/2}}$$

dove m_3 denota il momento centrato campionario di ordine 3. In generale, il momento centrato campionario di ordine j è così definito:

$$m_j = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^j \qquad (j = 1, 2, \ldots).$$

Vediamo come cambia la distribuzione di frequenza in base al valore che assume γ_1:

Se $\gamma_1 > 0$ asimmetrica positiva;

se $\gamma_1 < 0$ asimmetrica negativa;

se $\gamma_1 = 0$ simmetrica.

In R non c'è un comando che calcola la skewness direttamente, ma la funzione che abbiamo utilizzato lo fa:

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

Applichiamo skw a tutti i periodi:

periodo	tutto l'anno	stagionalmente	periodicamente	occasionalmente	Mancata risposta
skw	0.3996848	0.9623764	0.7074005	0.9437928	0.5860378
Distribuzione di frequenza	Asimmetrica positiva	Asimmetrica positiva	Asimmetrica positiva	Asimmetrica positiva	Asimmetrica positiva

L'asimmetria positiva significa che la distribuzione di frequenza ha una coda sbilanciata verso destra.

Infine, siamo poi interessati a valutare in che modo queste distribuzioni di frequenza si discostano da una normale, cioè se sono più piatte (platicurtiche) o più piccate (leptocurtiche).

Assegnato un insieme di dati numerici $x_1,\ldots, x_n$ si definisce **curtosi** campionaria il valore:

$$\gamma_2 = \beta_2 - 3,$$

dove

$$\beta_2 = \frac{m_4}{m_2^2},$$

Avendo denotato con m4 il momento centrato campionario di ordine 4.

Gli indici γ_2 e β_2 permettono di confrontare la distribuzione di frequenza dei dati con una densità di probabilità normale, caratterizzata da β_2 = 3 e indice di curtosi γ_2 = 0. Se risulta

β_2 < 3, γ_2 < 0: la distribuzione di frequenza si definisce platicurtica, ossia la distribuzione di frequenza è più piatta di una normale;

β_2 > 3, γ_2 > 0: la distribuzione di frequenza si definisce leptocurtica, ossia la distribuzione di frequenza è più piccata di una normale;

β_2 = 3, γ_2 = 0: la distribuzione di frequenza si definisce normocurtica, ossia piatta come una normale.

Come per la skewness, neanche la curtosi ha un comando in R che permette di calcolarla direttamente, però tramite la seguente funzione possiamo ottenerla:

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

applichiamo curtosi() a ogni periodo e otteniamo:

periodo	Tutto l'anno	Stagionalmente	Periodicamente	occasionalmente	Mancata risposta
curtosi	-0.8539201	0.4973891	-0.6191208	0.1804393	-0.3621489
Distribuzione di frequenza rispetto a una normale	Più piatta	Più piccata	Più piatta	Più piccata	Più piatta

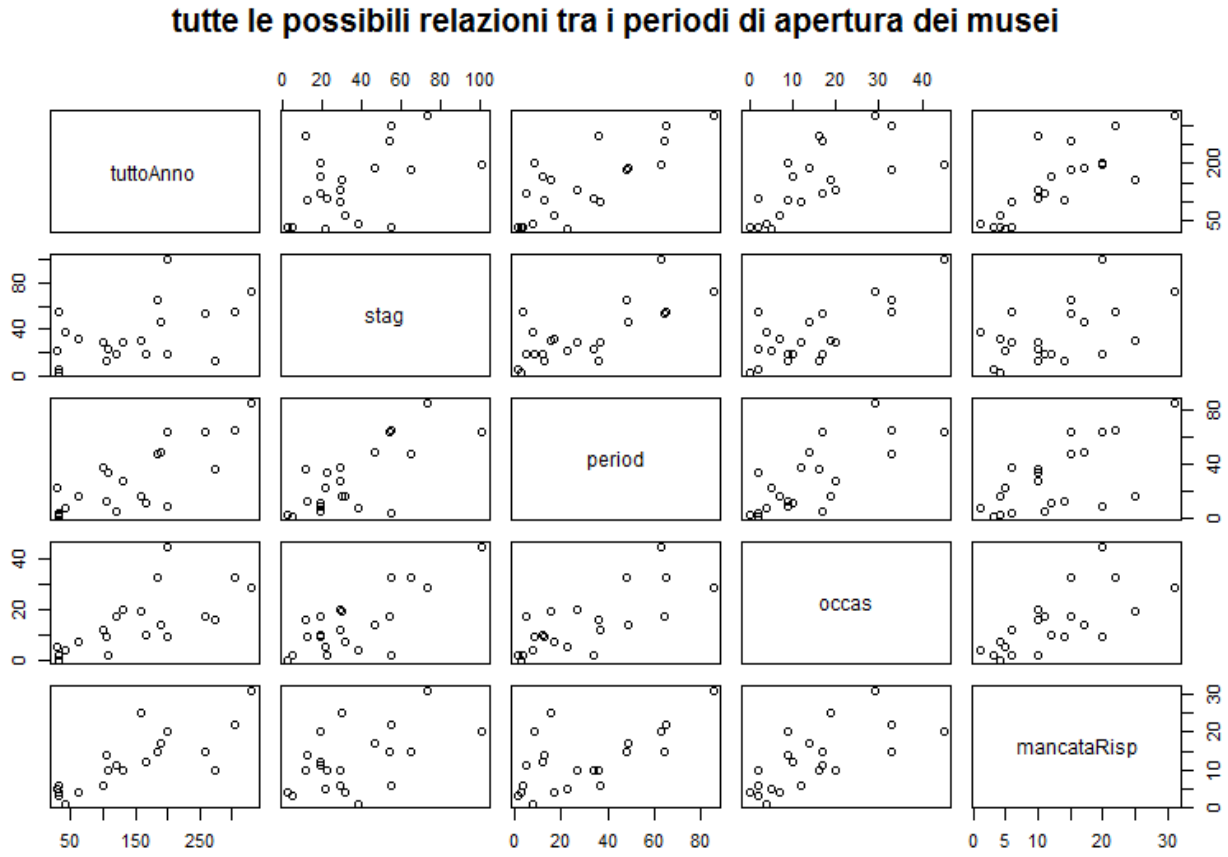
Osservazione: Allo stesso modo, e ottenendo gli stessi risultati, potevamo non sottrarre 3 a m4 nella funzione e confrontare i risultati della funzione curtosi con il numero 3(perché la normale ha una curtosi pari a 3).

Statistica bivariata

La statistica descrittiva bivariata è il ramo della statistica che si occupa dei metodi grafici e statistici atti descrivere le relazioni che intercorrono tra due variabili.

Precedentemente abbiamo visto la relazione che intercorre tra i periodi di apertura (le nostre variabili) tramite gli scatterplot. Tale grafico mira a evidenziare se le coppie di punti presentano qualche forma di regolarità, quindi se esiste una relazione lineare o non.

Di seguito riportiamo lo scatterplot, già in precedenza calcolato, che mostra tutte le possibili relazioni tra i periodi di apertura:



Grazie a questo tipo di grafico, possiamo vedere se c'è una relazione, e di che tipo è, tra due variabili. Nel nostro caso le variabili che prendiamo in esame sono i periodi di apertura. Oltre a visionare lo scatterplot e farci un'idea di che tipo è la relazione che intercorre tra ognuna di queste coppie di variabili, possiamo usare misure quantitative che indicano la correlazione che esiste tra le variabili. Queste misure sono la covarianza e la correlazione campionaria.

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

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

Da questa formula si evince che se il valore $x_i > \bar{x}$ la differenza è positiva, mentre se è minore la differenza è negativa. La stessa cosa vale anche per il secondo prodotto. Se il prodotto tra le due differenze è positivo allora (x_i, y_i) sono correlate positivamente, ossia $x_i > \bar{x}$ e $y_i > \bar{y}$ oppure $x_i < \bar{x}$ e $y_i < \bar{y}$. Se (x_i, y_i) sono correlate negativamente allora $x_i > \bar{x}$ e $y_i < \bar{y}$ oppure $x_i < \bar{x}$ e $y_i > \bar{y}$.

La covarianza campionaria, quindi, può avere segno positivo, negativo o nullo. Quando ha segno positivo si dice che le variabili sono correlate positivamente, se il segno è negativo le variabili sono correlate negativamente e se è uguale a 0 le variabili non sono correlate.

La covarianza campionaria viene calcolata in R tramite il comando cov.

Per ottenere una misura quantitativa della correlazione tra variabili si può anche considerare il coefficiente di correlazione campionaria:

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

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

Tale coefficiente ha lo stesso segno della covarianza ed è compreso tra -1 e 1.

Se

$r_{xy} > 0$: le variabili sono correlate positivamente;

$r_{xy} < 0$: le variabili sono correlate negativamente;

$r_{xy} = 0$: le variabili non sono correlate.

Se $r_{xy} =$ -1 le variabili hanno una forte correlazione negativa, ossia i punti si trovano su una retta decrescente. Al contrario, se $r_{xy} = 1$ le variabili hanno una forte correlazione positiva, ossia i punti sono disposti su una retta crescente.

In R il coefficiente di correlazione si calcola tramite il comando cor.

Abbiamo calcolato il coefficiente di correlazione e la covarianza di ognuna delle nostre variabili quantitative e abbiamo valori più alti per queste due coppie di variabili:

> cor(df$tuttoAnno, df$period)

[1] 0.7894135 ^ 2 = 0.6231737

> cov(df$tuttoAnno, df$period)

[1] 1811.698

> cor(df$tuttoAnno, df$mancataRisp)

[1] 0.8140978 ^ 2= 0.6627552

> cov(df$tuttoAnno, df$mancataRisp)

[1] 605.3571

Per qualsiasi coppia di periodi il valore della covarianza è positivo quindi i campioni sono correlati positivamente: i valori del primo e del secondo vettore tendono ad essere grandi e piccoli insieme.

Considerando invece la correlazione abbiamo il valore più alto per tutto l'anno-mancata risposta e tutto l'anno-periodicamente. Chiaramente, più questo valore è alto e più la correlazione è forte tra le variabili. Di seguito calcoliamo la retta di regressione per queste due coppie di periodi.

Regressione lineare semplice

Il modello di regressione lineare semplice è esprimibile attraverso l'equazione di una retta che riesce ad interpolare la nuvola di punti dello scatterplot meglio di tutte e altre possibili rette. Consideriamo l'equazione della retta:

$$Y = \alpha + \beta X$$

dove

α è l'intercetta e β è il coefficiente angolare.

Il coefficiente angolare β esprime quantitativamente la pendenza della retta: un coefficiente angolare positivo indica una retta di regressione crescente, un coefficiente angolare negativo indica una retta decrescente e un coefficiente angolare nullo indica una retta orizzontale.

L'intercetta α corrisponde all'ordinata del punto di intersezione della retta di regressione con l'asse delle ordinate.

Determineremo α e β mediante il metodo dei minimi quadrati, che consiste nel trovare quelli che minimizzano la somma:

$$Q = \sum_{i=1}^{n} \left[y_i - (\alpha + \beta x_i) \right]^2$$

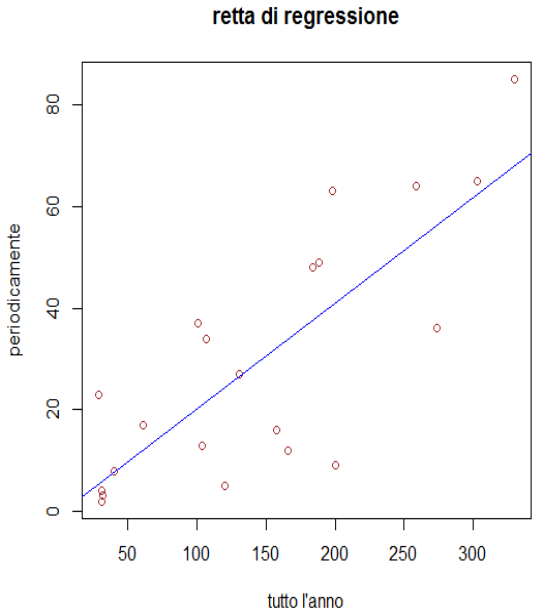
Otteniamo quindi:

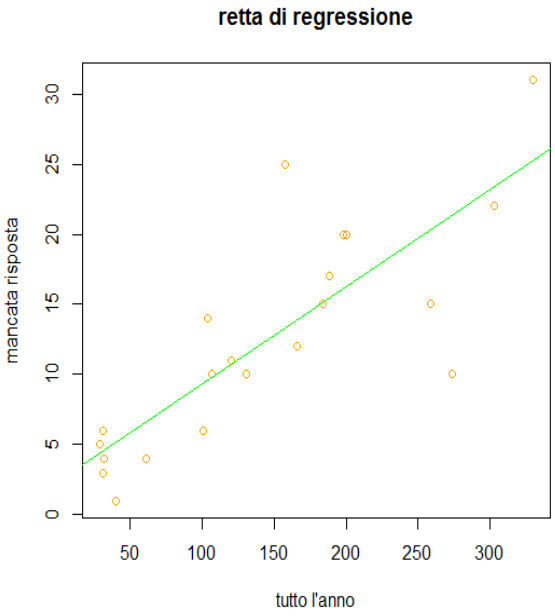
$$\beta = \frac{s_y}{s_x} r_{xy}, \qquad \alpha = \overline{y} - \beta \overline{x}.$$

Procediamo:

Consideriamo entrambe le coppie di variabili (in cui tuttoAnno è la variabile indipendente(X) per entrambe le coppie e period e mancataRisposta sono dipendenti(Y)). Cominciamo col considerare lo scatterplot con retta interpolante stimata, in parallelo per entrambe le coppie (per avere una visione più chiara, scriveremo il codice relativo a tutto l'anno – periodicamente, in quanto il codice relativo alla coppia tutto l'anno - mancata risposta si ha sostituendo a df$period df$mancataRisp, per ogni blocco di codice):

```
> plot(df$tuttoAnno, df$period, main="retta di regressione", xlab="tutto l'anno", ylab="periodicamente", col="brown")
> abline(lm(df$period~df$tuttoAnno), col="blue") # traccia la retta interpolante stimata
```





Lm (linear model) è un comando di R, che crea automaticamente la retta di regressione, che, come abbiamo specificato sopra, costruiamo con il metodo dei minimi quadrati:

> beta<-(sd(df$period)/sd(df$tuttoAnno))*cor (df$tuttoAnno,df$period)

> alpha<-mean (df$period)-beta*mean (df$tuttoAnno)

> c(alpha , beta)

[1] -0.5492924 0.2072646

per tutto l'anno-mancata risposta:

[1] 2.38000325 0.06925498

dove questi numeri sono i coefficienti delle rette di regressione. Notiamo che β è positivo e che quindi la retta è crescente

Ora, siamo interessati a sapere di quanto la retta si discosta dai dati osservati, per questo individuiamo i residui cioè gli scostamenti tra dati osservati e dati stimati.

Calcoliamo il vettore dei valori stimati:

$$\widehat{y_i} = \alpha + \beta\, x_i \qquad (i = 1, 2, \ldots, n)$$

stime <-fitted (lm(df$period~df$tuttoAnno)

stime

```
       1        2        3        4        5        6        7        8
40.489092  5.461380 26.602366 53.132231  5.875909  7.741290 38.416446 20.384429
       9       10       11       12       13       14       15       16
62.251872 67.848015 21.628016 37.587388 56.241199 12.093846  6.083174 33.856626
      17       18       19       20       21
21.006223  5.875909 24.322456 40.903621 32.198509
```

e poi quello dei residui:

$$E_i = y_i - \widehat{y}_i = y_i - (\alpha + \beta\, x_i) \qquad (i = 1, 2, \ldots, n)$$

```
residui<-resid (lm(df$period ~df$tuttoAnno))
residui
```

```
         1          2          3          4          5          6
 22.5109078  17.5386199   0.3976339  10.8677692  -1.8759092   0.2587096
         7          8          9         10         11         12
 10.5835535  16.6155710   2.7481281  17.1519848  12.3719836  10.4126118
        13         14         15         16         17         18
-20.2411994   4.9061537  -3.0831738 -21.8566260  -8.0062227  -3.8759092
        19         20         21
-19.3224558 -31.9036213 -16.1985094
```

Allo stesso modo per tutto l'anno-mancata risposta abbiamo:

stime:

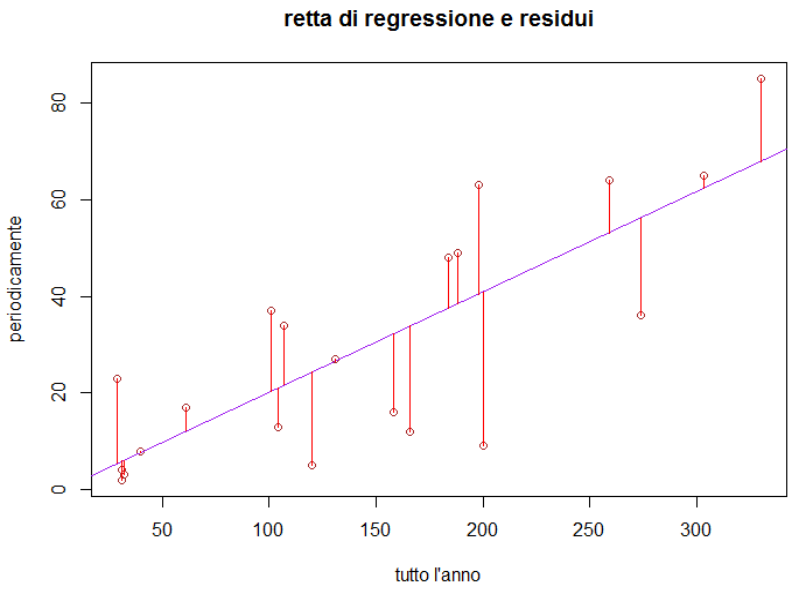
```
       1        2        3        4        5        6        7        8
16.092490  4.388398 11.452406 20.317044  4.526908  5.150203 15.399940  9.374756
       9       10       11       12       13       14       15       16
23.364263 25.234147  9.790286 15.122920 21.355868  6.604557  4.596163 13.876330
      17       18       19       20       21
 9.582521  4.526908 10.690601 16.231000 13.322290
```

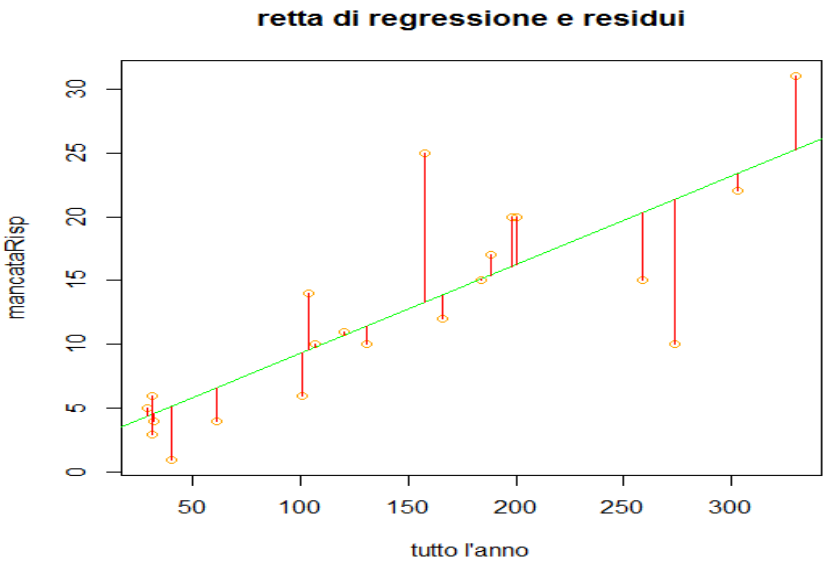
residui:

```
         1          2          3          4          5          6
  3.9075102  0.6116023 -1.4524060 -5.3170437  1.4730923 -4.1502025
         7          8          9         10         11         12
  1.6000600 -3.3747565 -1.3642630  5.7658525  0.2097136 -0.1229200
        13         14         15         16         17         18
-11.3558685 -2.6045572 -0.5961627 -1.8763303  4.4174786 -1.5269077
        19         20         21
  0.3093988  3.7690002 11.6777095
```

Costruiamo il grafico della retta di regressione e dei residui:

```
plot(df$tuttoAnno, df$period, main="retta di regressione e residui", xlab="tutto l'anno", ylab="periodicamente", col="brown")

abline(lm(df$period~df$tuttoAnno), col="purple")

stime<-fitted(lm(df$period~df$tuttoAnno))

segments (df$tuttoAnno ,stime ,df$tuttoAnno ,df$period,col="red")
```



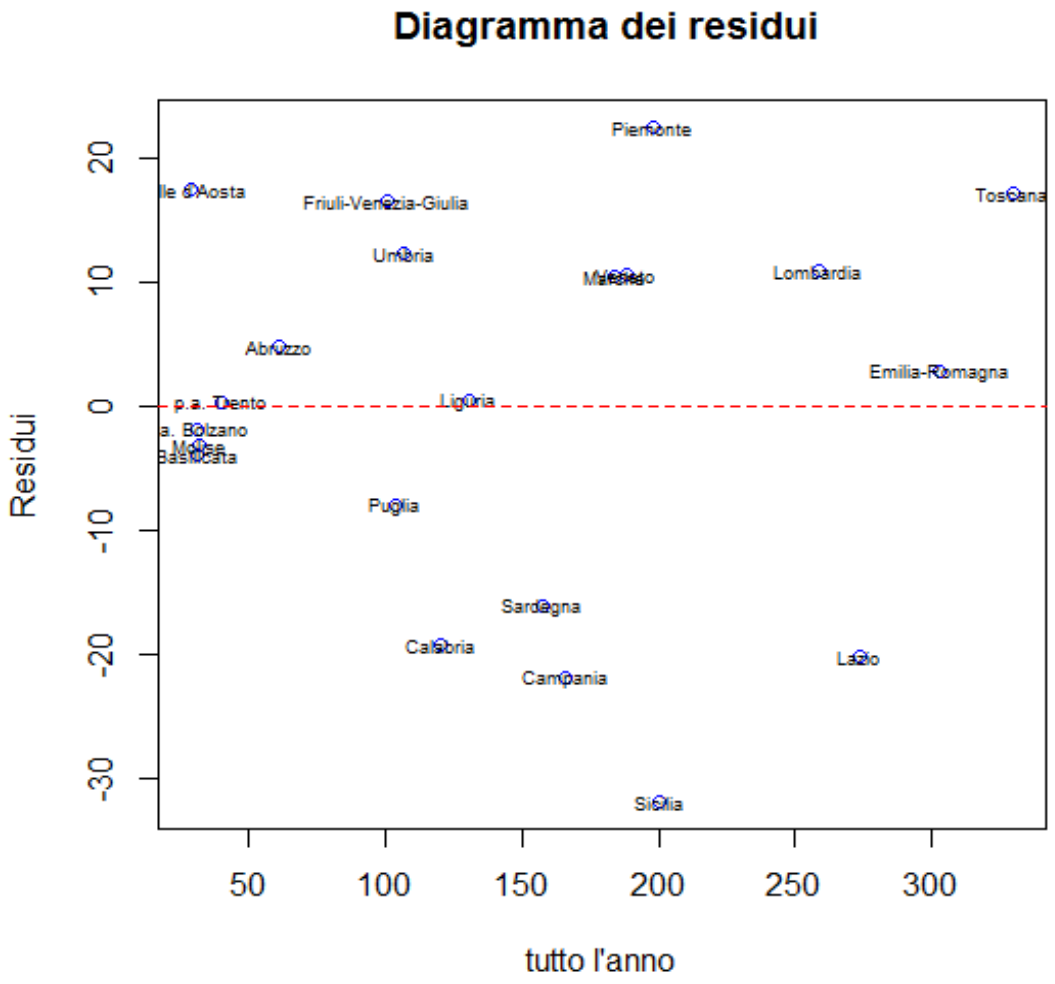


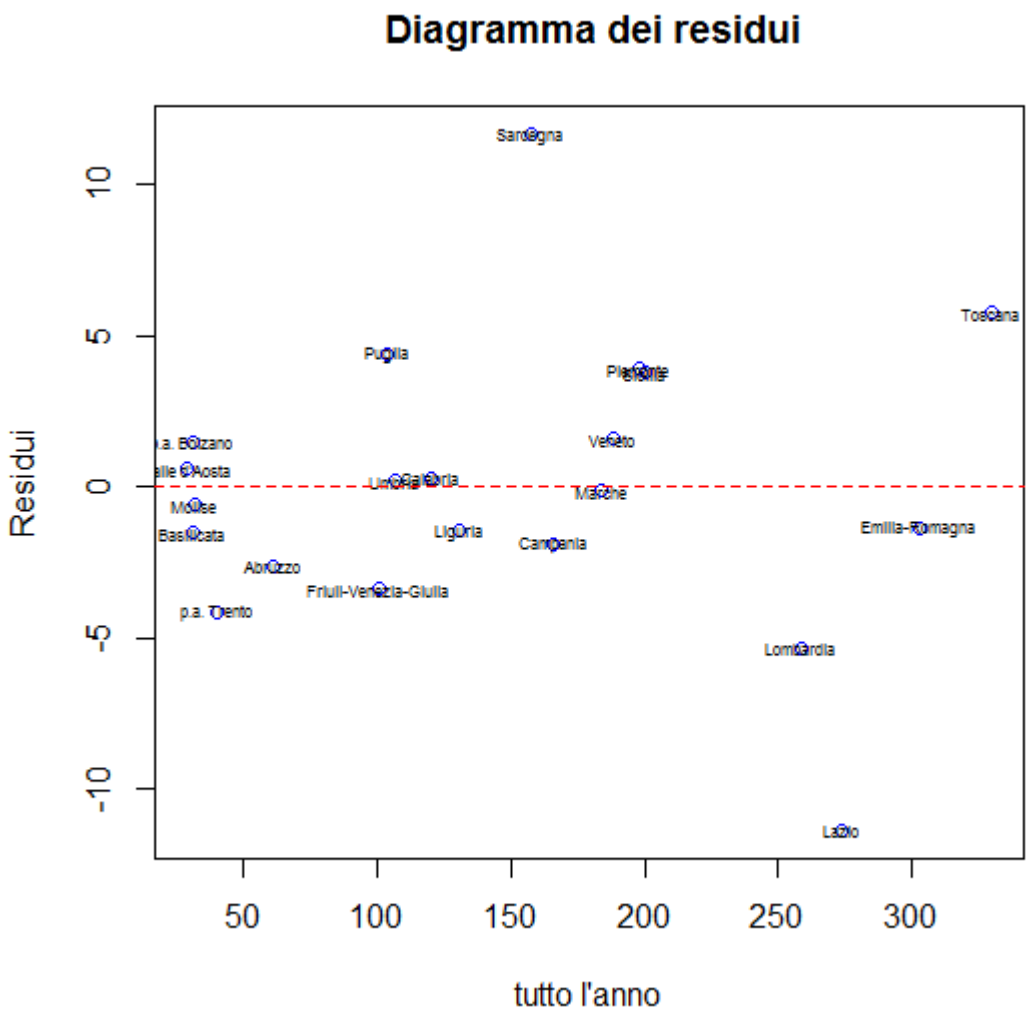
Entrambi le rette di regressione sono ascendenti, quindi i campioni esaminati sono correlati positivamente.

Per vedere come i residui si dispongono intorno alla retta interpolante e come ne influenzano la posizione, possiamo costruire 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.

Costruiamo il diagramma dei residui:

```
> residui<-resid(lm(df$period~df$tuttoAnno))
> plot(df$tuttoAnno,residui , main =" Diagramma dei residui", xlab="tutto l'anno",ylab =" Residui ",pch =9, col ="blue")
> text(x=df$tuttoAnno, y=residui+5, labels=rownames(df), cex=0.6)
> abline(h=0, col ="red",lty =2)
```





La retta orizzontale corrisponde alla media campionaria.

Il diagramma dei residui ha aiutato a comprendere qual è l'adattamento della retta di regressione rispetto alle regioni, consentendo di identificare quali sono le informazioni che hanno una forte influenza sulla collocazione e direzione della retta di regressione.

Regressione lineare multipla

Utilizziamo questo tipo di regressione quando c'è una variabile dipendente e due o più variabili indipendenti. In generale, abbiamo p + 1 variabili Y, $X_{1,,}..., X_p$.

cov(df) e cor(df) forniscono due matrici di dimensioni (p + 1) · (p + 1) i cui elementi solo le covarianze e le correlazioni tra coppie di variabili. In particolare, tali matrici sono simmetriche.

Consideriamo come variabile dipendente il periodo *periodicamente* e tutte le altre come variabili indipendenti. Ricostruiamo il dataframe:

	tuttoAnno	stag	period	occas	mancataRisp
Piemonte	198	101	63	45	20
Valle d'Aosta	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
p.a. Bolzano	31	55	4	2	6
p.a. Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia-Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

e la relativa covarianza e correlazione:

La matrice delle covarianze cov(df) contiene sulla diagonale principale la varianza delle singole colonne del data frame. La matrice delle correlazioni cor(df) contiene il numero 1 sulla diagonale principale. La matrice di correlazione evidenzia tutte le correlazioni tra le coppie di variabili.

```
> cov(df)
            tuttoAnno      stag    period     occas mancataRisp
tuttoAnno   8740.9905 1079.9119 1811.6976 802.79762   605.35714
stag        1079.9119  606.7476  444.5905 227.49048   101.92857
period      1811.6976  444.5905  602.5619 220.56190   125.91429
occas        802.7976  227.4905  220.5619 145.66190    66.86429
mancataRisp  605.3571  101.9286  125.9143  66.86429    63.25714
> cor(df)
            tuttoAnno      stag    period     occas mancataRisp
tuttoAnno   1.0000000 0.4689256 0.7894135 0.7114637   0.8140978
stag        0.4689256 1.0000000 0.7352843 0.7652198   0.5202800
period      0.7894135 0.7352843 1.0000000 0.7444862   0.6449401
occas       0.7114637 0.7652198 0.7444862 1.0000000   0.6965727
mancataRisp 0.8140978 0.5202800 0.6449401 0.6965727   1.0000000
```

Supponendo, quindi, che periodicamente sia la variabile dipendente e tutte le altre siano indipendenti:

```
> lm(df$period ~ df$tuttoAnno+df$stag+df$occas+df$mancataRisp)

Call:
lm(formula = df$period ~ df$tuttoAnno + df$stag + df$occas +
    df$mancataRisp)

Coefficients:
   (Intercept)    df$tuttoAnno         df$stag        df$occas  df$mancataRisp
      -8.17976         0.19216         0.52508        -0.08959        -0.59981
```

α = -8.17976 e i regressori β_1 = 0.19216, β_2 = 0.52508, β_3 = -0.08959 e β_4 = -0.59981.

Pertanto, il modello di regressione multipla stimato è:

y = -8.17976 + 0.19216x_1 + 0.52508x_2 - 0.08959x_3 - 0.59981x_4

Notiamo che i periodi *stagionalmente* e *tutto l'anno* hanno un'influenza positiva su *periodicamente*. Ora calcoliamo i valori stimati:

$$\widehat{y_i} = \alpha + \beta_1\, x_{i,1} + \beta_2\, x_{i,2} + \ldots + \beta_p\, x_{i,p} \qquad (i = 1, 2, \ldots, n)$$

```
> stimemult<-fitted(lm(df$period ~ df$tuttoAnno+df$stag+df$occas+df$mancataRisp))
> stimemult
        1         2         3         4         5         6         7         8
66.873662  5.497712 24.430821 59.424138 22.878661 18.501595 41.174307 21.781930
        9        10        11        12        13        14        15        16
62.772238 72.372113 18.281049 49.354581 43.341772 17.318297 -2.854602 25.601892
       17        18        19        20        21
 9.427414 -1.575967 16.735197 27.426479 21.236710
```

e i residui:

$$E_i = y_i - \widehat{y}_i = y_i - (\alpha + \beta_1\, x_{i,1} + \beta_2\, x_{i,2} + \ldots \beta_p\, x_{i,p}) \qquad (i = 1, 2, \ldots, n)$$

```
> residmult<-resid(lm(df$period ~ df$tuttoAnno+df$stag+df$occas+df$mancataRisp))
```

```
> residmult
```

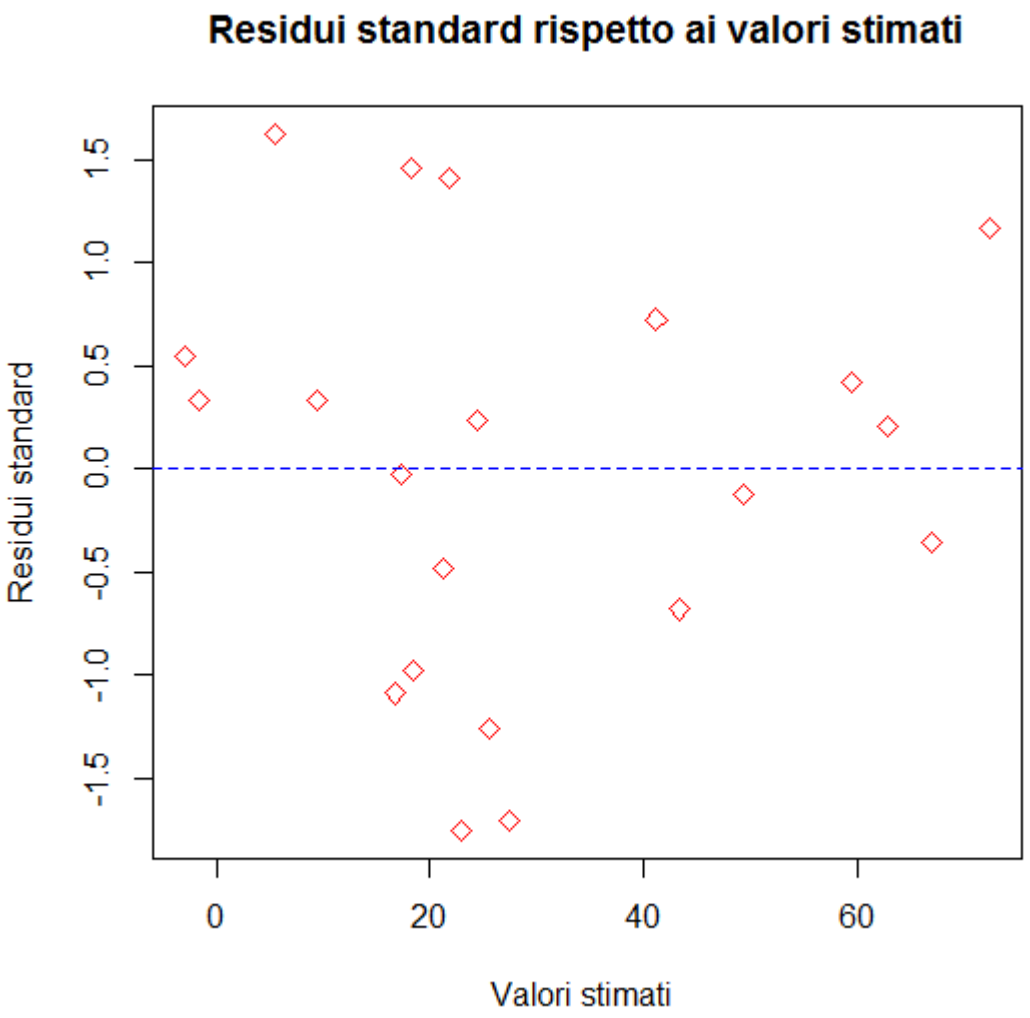
```
          1           2           3           4           5           6
 -3.8736617  17.5022885   2.5691786   4.5758620 -18.8786606 -10.5015945
          7           8           9          10          11          12
  7.8256926  15.2180699   2.2277617  12.6278868  15.7189512  -1.3545814
         13          14          15          16          17          18
 -7.3417721  -0.3182975   5.8546018 -13.6018919   3.5725859   3.5759670
         19          20          21
-11.7351973 -18.4264794  -5.2367095
```

Costruiamo il grafico:

```
> residuimultstandard <- residmult /sd( residmult )
```

```
> plot(stimemult , residuimultstandard , main=" Residui standard rispetto ai valori stimati ",
xlab="Valori stimati ",ylab =" Residui standard ",pch =5, col ="red ")
```

```
> abline (h=0, col ="blue ",lty =2)
```



In questo caso i punti sono disposti casualmente attorno alla linea orizzontale e non si evidenzia nessuna tendenza particolare nella distribuzione dei punti.

Calcoliamo il coefficiente di determinazione che è il rapporto tra la varianza dei valori stimati tramite la retta di regressione e la varianza dei valori osservati:

$$D^2 = \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}.$$

L'indice D^2 è adimensionale e risulta $0 \leq D^2 \leq 1$. Quando $D^2 = 0$ il modello di regressione multipla utilizzato non spiega per nulla i dati. Invece, quando $D^2 = 1$ il modello di regressione multipla utilizzato spiega perfettamente i dati.

In R tale coefficiente si calcola nel seguente modo:

```
> summary(lm(df$period ~ df$tuttoAnno+df$stag+df$occas+df$mancataRisp))$r.square

[1] 0.8070526
```

Possiamo quindi dire che il modello di regressione multipla utilizzato può spiegare significativamente i dati.

Nel caso di regressione lineare, D^2 coincide con il quadrato del coefficiente di correlazione che nel nostro caso (quello in cui period è variabile dipendente) è uguale a 0.6231737. Confrontando questo risultato con il coefficiente di determinazione, abbiamo che è meglio usare la regressione multipla, in quanto il coefficiente risulta essere più alto.

Regressione non lineare

Spesso l'ipotesi di linearità di un modello non è accettabile. Abbiamo provato ad approssimare le nostre coppie di variabili con i modelli di regressione non lineare studiati in questo corso, ma i modelli studiati non producono un buon risultato su questi dati.

Analisi dei cluster

L'analisi dei cluster ci permette di raggruppare in sottoinsiemi, detti cluster, dati appartenenti ad un insieme più ampio. I raggruppamenti vengono fatti in modo da inserire nello stesso cluster dati tra di loro sono più simili oppure che sono poco distanti, mentre inserire in cluster diversi dati che non sono simili oppure che sono molto distanti tra di loro. La nostra analisi prenderà in esame il data frame precedentemente studiato, ossia le regioni e i periodi di apertura dei musei di ogni regione. Siccome in generale si parla di individui e caratteristiche, e quindi raggruppare gli individui in base alle loro caratteristiche, nel nostro caso gli individui sono le regioni e le caratteristiche sono i periodi. Per prima cosa, ci costruiamo di nuovo il data frame su cui andremo a lavorare:

	tuttoAnno	stag	period	occas	mancataRisp
Piemonte	198	101	63	45	20
Valle d'Aosta	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
p.a. Bolzano	31	55	4	2	6
p.a. Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia-Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

Per risolvere il problema di clustering è chiaramente desiderabile definire i termini **somiglianza** o **differenza** in modo quantitativo, ossia occorre precisare cosa significa la somiglianza di due individui I_i e I_j assegnati allo stesso cluster e la differenza di due individui assegnati a differenti cluster. La somiglianza può essere definita mediante un coefficiente di similarità s_{ij} = s(X_i, X_j) oppure mediante una misura di distanza d_{ij} = d(X_i, X_j) tra due individui I_i e I_j (i ≠ j).

d(X_i, X_j) è detta funzione di distanza se e solo se soddisfa le seguenti condizioni:

$$(i)\ d(X_i, X_j) = 0 \ se\ e\ solo\ se\ X_i = X_j,\ con\ X_i\ e\ X_j\ in\ E_p;$$

$$(ii)\ d(X_i, X_j) \geq 0\ per\ ogni\ X_i\ e\ X_j\ in\ E_p;$$

$$(iii)\ d(X_i, X_j) = d(X_j, X_i)\ per\ ogni\ X_i\ e\ X_j\ in\ E_p;$$

$$(iv)\ d(X_i, X_j) \leq d(X_i, X_k) + d(X_k, X_j)\ per\ ogni\ X_i,\ X_j\ e\ X_k\ in\ E_p.$$

Mentre i coefficienti di similarità assumono valori compresi tra 0 e 1, le misure di distanza possono assumere qualsiasi valore reale maggiore o uguale a zero. In generale la distanza tra tutte le possibili coppie di unità sono inserite in una matrice simmetrica D di cardinalità n * n, ossia:

$$D = \begin{pmatrix} 0 & d_{12} & \dots & d_{1n} \\ d_{21} & 0 & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & 0 \end{pmatrix}$$

Noi prendiamo in esame la matrice delle distanze. Tale matrice può essere costruita mediante vari metodi, ma quello che useremo noi è il metodo Euclideo, ossia

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

Costruiamo la matrice delle distanze tramite il comando dist:

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

	Piemonte	Valle d'Aosta	Liguria	Lombardia
Piemonte	0.000000	195.517263	108.138800	82.097503
Valle d'Aosta	195.517263	0.000000	103.532604	236.323930
Liguria	108.138800	103.532604	0.000000	135.690825
Lombardia	82.097503	236.323930	135.690825	0.000000
p.a. Bolzano	188.496684	38.262253	107.447662	236.412775
p.a. Trento	184.390889	24.879711	95.184032	227.415919
Veneto	64.668385	163.728434	64.358372	72.993150
Friuli-Venezia-Giulia	128.662349	74.020267	32.863353	162.554606
Emilia-Romagna	115.295273	281.072944	178.932948	47.360321
Toscana	138.090550	313.525119	213.126723	79.012657
Umbria	130.977097	78.993671	31.384710	158.792317
Marche	43.428102	165.478095	68.847658	79.107522
Lazio	123.963704	245.845480	144.343341	52.905576
Abruzzo	165.366260	34.132096	72.208033	205.226704
Molise	207.463250	28.213472	107.186753	241.373155
Campania	107.879562	137.742513	40.669399	112.409964
Puglia	142.870571	76.830983	36.304270	168.439900
Basilicata	207.277592	27.331301	107.582526	242.070238
Calabria	130.525860	93.776330	26.739484	155.058054
Sicilia	104.594455	172.299158	73.525506	88.430764
Sardegna	97.729218	131.719399	32.817678	114.825955

	p.a. Bolzano	p.a. Trento	Veneto	Friuli-Venezia-Giulia
Piemonte	188.496684	184.390889	64.668385	128.662349
Valle d'Aosta	38.262253	24.879711	163.728434	74.020267
Liguria	107.447662	95.184032	64.358372	32.863353
Lombardia	236.412775	227.415919	72.993150	162.554606
p.a. Bolzano	0.000000	20.371549	164.325896	82.249620
p.a. Trento	20.371549	0.000000	154.990322	68.789534
Veneto	164.325896	154.990322	0.000000	90.343788
Friuli-Venezia-Giulia	82.249620	68.789534	90.343788	0.000000
Emilia-Romagna	280.930596	272.009191	118.029657	207.270355
Toscana	312.473999	304.596454	150.189880	239.989583
Umbria	87.840765	73.993243	86.919503	14.035669
Marche	162.748272	155.248188	26.570661	93.957437
Lazio	249.266925	237.573147	94.037227	173.928146
Abruzzo	40.336088	24.000000	132.649915	45.144213
Molise	52.096065	36.592349	169.567096	82.103593
Campania	140.303243	128.101522	51.749396	70.639932
Puglia	85.363927	70.285134	97.683161	30.232433
Basilicata	50.129831	34.842503	170.182255	82.522724
Calabria	97.303648	83.898749	85.959293	39.179076
Sicilia	173.571311	162.320670	50.616203	104.355163
Sardegna	132.468864	121.872885	48.651824	64.039051

	Emilia-Romagna	Toscana	Umbria	Marche
Piemonte	115.295273	138.090550	130.977097	43.428102
Valle d'Aosta	281.072944	313.525119	78.993671	165.478095
Liguria	178.932948	213.126723	31.384710	68.847658
Lombardia	47.360321	79.012657	158.792317	79.107522
p.a. Bolzano	280.930596	312.473999	87.840765	162.748272
p.a. Trento	272.009191	304.596454	73.993243	155.248188
Veneto	118.029657	150.189880	86.919503	26.570661
Friuli-Venezia-Giulia	207.270355	239.989583	14.035669	93.957437
Emilia-Romagna	0.000000	39.370039	203.730214	120.826322
Toscana	39.370039	0.000000	236.643191	151.726728
Umbria	203.730214	236.643191	0.000000	94.207218
Marche	120.826322	151.726728	94.207218	0.000000
Lazio	62.960305	99.337808	167.958328	106.616134
Abruzzo	249.793915	282.628732	50.467812	134.074606
Molise	285.310357	319.371257	83.821238	173.732553
Campania	153.306882	189.375289	63.631753	65.375837
Puglia	211.445028	246.067064	24.799194	104.431796
Basilicata	285.963284	320.042185	84.693565	173.867766
Calabria	196.880675	232.292919	35.383612	91.285267
Sicilia	124.983999	161.595173	97.154516	67.037303
Sardegna	155.743379	190.604302	59.059292	56.753854

	Lazio	Abruzzo	Molise	Campania	Puglia
Piemonte	123.963704	165.366260	207.463250	107.879562	142.870571
Valle d'Aosta	245.845480	34.132096	28.213472	137.742513	76.830983
Liguria	144.343341	72.208033	107.186753	40.669399	36.304270
Lombardia	52.905576	205.226704	241.373155	112.409964	168.439900
p.a. Bolzano	249.266925	40.336088	52.096065	140.303243	85.363927
p.a. Trento	237.573147	24.000000	36.592349	128.101522	70.285134
Veneto	94.037227	132.649915	169.567096	51.749396	97.683161
Friuli-Venezia-Giulia	173.928146	45.144213	82.103593	70.639932	30.232433
Emilia-Romagna	62.960305	249.793915	285.310357	153.306882	211.445028
Toscana	99.337808	282.628732	319.371257	189.375289	246.067064
Umbria	167.958328	50.467812	83.821238	63.631753	24.799194
Marche	106.616134	134.074606	173.732553	65.375837	104.431796
Lazio	0.000000	215.051157	245.002041	111.036030	171.741084
Abruzzo	215.051157	0.000000	43.897608	106.263823	48.270074
Molise	245.002041	43.897608	0.000000	135.856542	74.598928
Campania	111.036030	106.263823	135.856542	0.000000	62.337790
Puglia	171.741084	48.270074	74.598928	62.337790	0.000000
Basilicata	245.965445	43.358967	3.316625	136.623570	75.392307
Calabria	157.251391	62.793312	91.334550	47.063787	20.712315
Sicilia	80.018748	140.762211	169.861708	35.071356	96.457244
Sardegna	120.058319	99.995000	132.574507	21.236761	58.608873

	Basilicata	Calabria	Sicilia	Sardegna
Piemonte	207.277592	130.525860	104.594455	97.729218
Valle d'Aosta	27.331301	93.776330	172.299158	131.719399
Liguria	107.582526	26.739484	73.525506	32.817678
Lombardia	242.070238	155.058054	88.430764	114.825955
p.a. Bolzano	50.129831	97.303648	173.571311	132.468864
p.a. Trento	34.842503	83.898749	162.320670	121.872885
Veneto	170.182255	85.959293	50.616203	48.651824
Friuli-Venezia-Giulia	82.522724	39.179076	104.355163	64.039051
Emilia-Romagna	285.963284	196.880675	124.983999	155.743379
Toscana	320.042185	232.292919	161.595173	190.604302
Umbria	84.693565	35.383612	97.154516	59.059292
Marche	173.867766	91.285267	67.037303	56.753854
Lazio	245.965445	157.251391	80.018748	120.058319
Abruzzo	43.358967	62.793312	140.762211	99.995000
Molise	3.316625	91.334550	169.861708	132.574507
Campania	136.623570	47.063787	35.071356	21.236761
Puglia	75.392307	20.712315	96.457244	58.608873
Basilicata	0.000000	91.733309	170.716139	133.127758
Calabria	91.733309	0.000000	81.000000	43.428102
Sicilia	170.716139	81.000000	0.000000	45.376205
Sardegna	133.127758	43.428102	45.376205	0.000000

Quando ci troviamo di fronte a dati che non hanno la stessa unità di misura, c'è la necessità di utilizzare il comando scale per standardizzare le variabili. Nel nostro caso, abbiamo tutti numeri e quindi non c'è bisogno di utilizzare tale comando.

Calcoliamo le medie campionarie e le deviazioni standard campionarie:

```
> apply(df, 2, mean)
  tuttoAnno        stag      period       occas mancataRisp
  145.09524    35.38095    29.52381    14.52381    12.42857
> apply(df, 2, sd)
  tuttoAnno        stag      period       occas mancataRisp
  93.493264   24.632248   24.547136   12.069047    7.953436
```

Abbiamo calcolato la matrice delle distanze usando la distanza euclidea, possiamo calcolare tale matrice anche con altri tipi di metriche. Vediamo come funzionano.

Metrica di Manhattan è così definita:

$$d_1(X_i, X_j) = \sum_{k=1}^{p} |x_{ik} - x_{jk}|$$

Se si considerano due caratteristiche la metrica Manhattan corrisponde alla somma delle misure dei due cateti di un triangolo rettangolo. Si chiama "Manhattan" perché essa corrisponde alla lunghezza che si deve percorrere qualora sia consentito di muoversi solo nelle direzioni parallele agli assi.

Metrica di Chebycev è così definita:

$$d_{\infty}(X_i, X_j) = \max_{k=1,2\ldots,p} |x_{ik} - x_{jk}|.$$

Entrambe queste metriche sono computazionalmente semplici da calcolare con l'unica differenza che la metrica di Chebycev coinvolge anche una procedura di ordinamento.

Metrica di Minkowski è definita come:

$$d_r(X_i, X_j) = \left[\sum_{k=1}^{p} |x_{ik} - x_{jk}|^r\right]^{1/r},$$

dove r ≥ 1. Se r=2 si ha la metrica Euclidea, se r=1 si ottiene la metrica di Manhattan e se r=∞ si ottiene la metrica di Chebycev.

Metrica di Canberra è definita come:

$$d_c(X_i, X_j) = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{|x_{ik} + x_{jk}|},$$

in cui sono omessi i valori aventi zero al numeratore o al denominatore. Se si utilizza tale metrica non è necessario scalare la matrice dei dati, poiché i contributi alla somma sono adimensionali. Inoltre, la metrica di Canberra è poco sensibile alla presenza di eventuali valori anomali (outlier).

Metrica di Jaccard è definita come:

$$d(X_i, X_j) = 1 - \frac{\sum_{k=1}^{p} \min(x_{ik}, x_{jk})}{\sum_{k=1}^{p} \max(x_{ik}, x_{jk})}$$

In R è disponibile la distanza di Jaccard solo per vettori binari.

Misura di non omogeneità totale

Dato il nostro data frame

	tuttoAnno	stag	period	occas	mancataRisp
Piemonte	198	101	63	45	20
Valle d'Aosta	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
p.a. Bolzano	31	55	4	2	6
p.a. Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia-Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

```
> apply(df, 2, mean)
  tuttoAnno        stag      period       occas mancataRisp
  145.09524    35.38095    29.52381    14.52381    12.42857
> apply(df, 2, sd)
  tuttoAnno        stag      period       occas mancataRisp
  93.493264   24.632248   24.547136   12.069047    7.953436
```

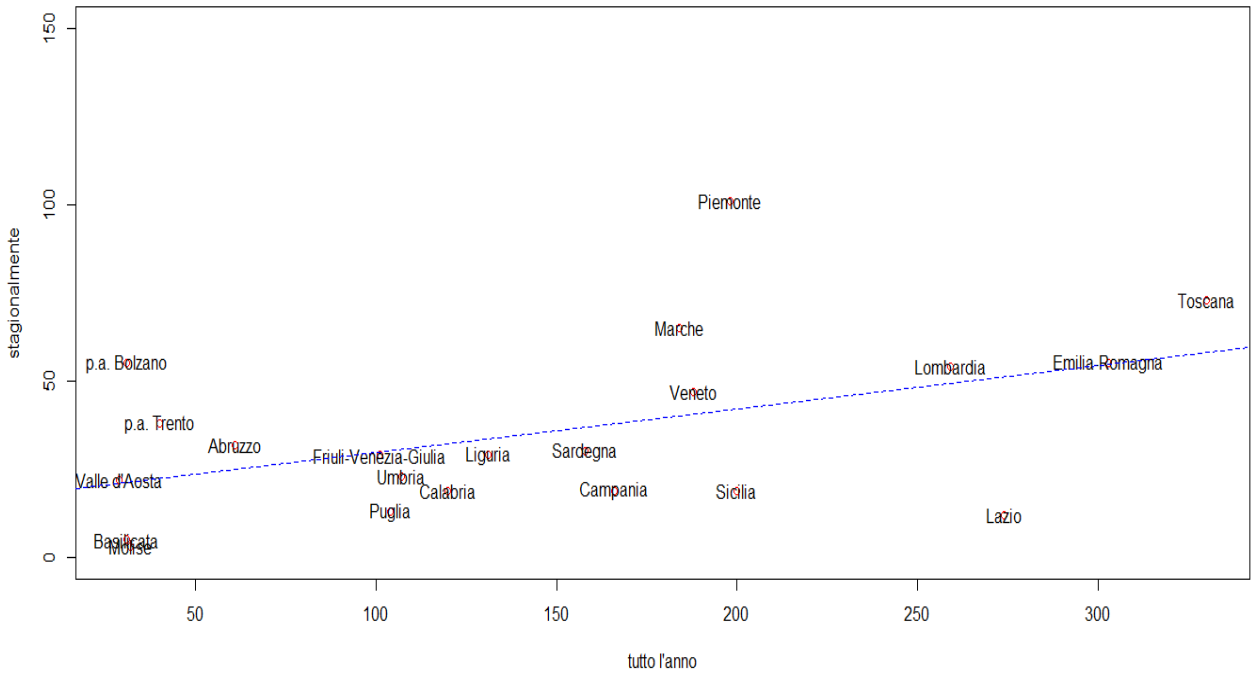
Alla nostra matrice df possiamo associare la matrice delle varianze e covarianze:

```
> WI<-cov(df)
> WI
             tuttoAnno      stag    period     occas mancataRisp
tuttoAnno   8740.9905 1079.9119 1811.6976 802.79762   605.35714
stag        1079.9119  606.7476  444.5905 227.49048   101.92857
period      1811.6976  444.5905  602.5619 220.56190   125.91429
occas        802.7976  227.4905  220.5619 145.66190    66.86429
mancataRisp  605.3571  101.9286  125.9143  66.86429    63.25714
```

Sulla diagonale principale abbiamo le varianze campionarie, tutti gli altri valori rappresentano la covarianza tra due periodi (per esempio: cov (tuttoAnno, stag) = 1079.9119)

Notiamo che tutti i periodi sono tra di loro correlati positivamente. Tracciamone il grafico scegliendo tutto l'anno e stagionalmente:

```
plot(df$tuttoAnno , df$stag,col ="red ",xlab="tutto l'anno",
ylab="stagionalmente",ylim=c(0 ,150) )

text(df$tuttoAnno, df$stag+0.1, c("Piemonte", "Valle d'Aosta", "Liguria", "Lombardia", "p.a. Bolzano", "p.a. Trento", "Veneto", "Friuli-Venezia-Giulia", "Emilia-Romagna", "Toscana", "Umbria", "Marche", "Lazio", "Abruzzo", "Molise", "Campania", "Puglia", "Basilicata", "Calabria", "Sicilia", "Sardegna") )

abline (lm(df$stag~df$tuttoAnno),lty =2, col ="blue ")
```



Calcoliamo per prima cosa la matrice statistica di non omogeneità, che è così definita:

$$H_I = (n-1)W_I = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1p} \\ h_{21} & h_{22} & \dots & h_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ h_{p1} & h_{p2} & \dots & h_{pp} \end{pmatrix}$$

dove,

$$h_{r\ell} = \sum_{i=1}^{n} (x_{ir} - \overline{x}_r)\,(x_{i\ell} - \overline{x}_\ell) = (n-1)\,w_{r\ell} \qquad (r, \ell = 1, 2 \dots, p).$$

Quando r = l, quindi sulla diagonale, si ha n-1 volte la varianza campionaria della caratteristica r-esima.

```
> n<-nrow (df)
> w<-cov(df)
> h<-(n-1) *w
> h
            tuttoAnno      stag    period     occas mancataRisp
tuttoAnno   174819.81 21598.238 36233.952 16055.952   12107.143
stag         21598.24 12134.952  8891.810  4549.810    2038.571
period       36233.95  8891.810 12051.238  4411.238    2518.286
occas        16055.95  4549.810  4411.238  2913.238    1337.286
mancataRisp  12107.14  2038.571  2518.286  1337.286    1265.143
```

Calcoliamo la misura di non omogeneità statistica, che è la *traccia* della matrice h, così definita:

$$\mathrm{tr}H_I = \sum_{r=1}^{p} h_{rr} = (n-1)\sum_{r=1}^{p} s_r^2.$$

```
n<-nrow(df)

trHI<-(n-1)*sum(apply(df,2,var))

trHI

[1] 203184.4
```

la quale può essere calcolata con altri metodi, tra cui:

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

> traccia<-sum(d^2)/n

> traccia

[1] 203184.4
```

Una volta calcolata la misura di non omogeneità totale, calcoliamo la misura di non omogeneità tra i cluster. Di seguito considereremo brevemente una suddivisione in due cluster per poi spiegare il motivo per cui non va bene tale suddivisione.

Consideriamo due cluster, il primo formato dalle prime 10 regioni e il secondo formato dalle restanti 11.

	tuttoAnno	stag	period	occas	mancataRisp
Piemonte	198	101	63	45	20
Valle d'Aosta	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
p.a. Bolzano	31	55	4	2	6
p.a. Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia-Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31

Ne calcoliamo la media, la varianza e la matrice di covarianza:

Costruiamo la matrice di non omogeneità statistica

```
> n1<-nrow(cluster1)
```

```
> h1<-(n1-1)*wi1
```

```
> h1
```

	tuttoAnno	stag	period	occas	mancataRisp
tuttoAnno	113892	13183.0	25967.0	10572.0	8811.0
stag	13183	5014.1	3410.5	2248.7	1354.1
period	25967	3410.5	6560.5	2665.5	2104.5
occas	10572	2248.7	2665.5	1752.9	923.7
mancataRisp	8811	1354.1	2104.5	923.7	788.1

e la traccia:

```
> tracciacluster1<-sum(diag(h1))
```

```
> tracciacluster1
```

```
[1] 128007.6
```

Ripetiamo la stessa procedura per il secondo cluster:

di nuovo ricreiamo il data frame del secondo cluster:

	tuttoAnno	stag	period	occas	mancataRisp
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

```
> tracciacluster2
```

[1] 62625.09

Calcoliamo la traccia interna ai cluster, sommando le due tracce precedentemente ottenute:

```
> tracciawithin<-tracciacluster1+tracciacluster2
```

```
> tracciawithin
```

[1] 190632.7

Calcoliamo la traccia tra i cluster, sottraendo alla traccia di non omogeneità totale precedentemente calcolata la traccia interna:

```
> tracciabetween<-trHI-tracciawithin
```

```
> tracciabetween
```

[1] 12551.69

Notiamo che la traccia tra i cluster (tracciabetween=12551.69) è minore di quella interna(tracciawithin=190632.7), ciò significa che la suddivisione dei cluster non è stata fatta nel modo giusto. Per avere una buona suddivisione dei cluster deve risultare che la traccia tra i cluster è maggiore di quella interna ai cluster. Di seguito vedremo come possiamo ottimizzare la scelta dei cluster. Partiamo utilizzando i metodi non gerarchici per poi portare avanti la scelta da essi ottenuta anche con i metodi gerarchici.

Metodi non gerarchici

L'obiettivo dei metodi non gerarchici è quello di ottenere una sola partizione degli n individui di partenza in cluster. A differenza dei metodi gerarchici, in tali tecniche è consentito riallocare gli individui già classificati ad un livello precedente dell'analisi.

Il metodo più utilizzato nei metodi non gerarchici è il k-means. Ecco consiste dei seguenti passi:

1. Fissare a priori in numero di cluster k specificando k punti di riferimento iniziali. Quindi produrre una partizione provvisoria;
2. Considerare tutti gli individui e attribuire ciascuno di essi al cluster individuato dal punto di riferimento da cui ha distanza minore;
3. In ogni cluster, calcolare i centroidi, che costituiranno i punti di riferimento;
4. Valutare la distanza di ogni unità da ogni centroide ottenuto al passo precedente. Se la distanza minima non è ottenuta in corrispondenza del centroide del gruppo di appartenenza, allora si procede a spostare l'individuo presso il cluster che il ha centroide più vicino;
5. Ricalcolare i centroidi dei k gruppi così ottenuti;
6. Ripetere il procedimento a partire dal punto (4) fino a che i centroidi non subiscono più variazioni rispetto all'iterazione precedente.

Nel metodo del k-means viene utilizzata la distanza euclidea e si considerano i quadrati delle distanze euclidee.

Di seguito mostriamo il codice che abbiamo utilizzato per scegliere il numero di cluster migliore:

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

d2<-d^2

hc<-hclust (d2 , method = "centroid")

taglio<-cutree (hc , k =3, h =NULL )

tagliolist<-list( taglio )

centroidiIniziali<-aggregate (df, tagliolist , mean)[,-1]

km3<-kmeans (df, centers = centroidiIniziali , iter.max = 10)

km3
```

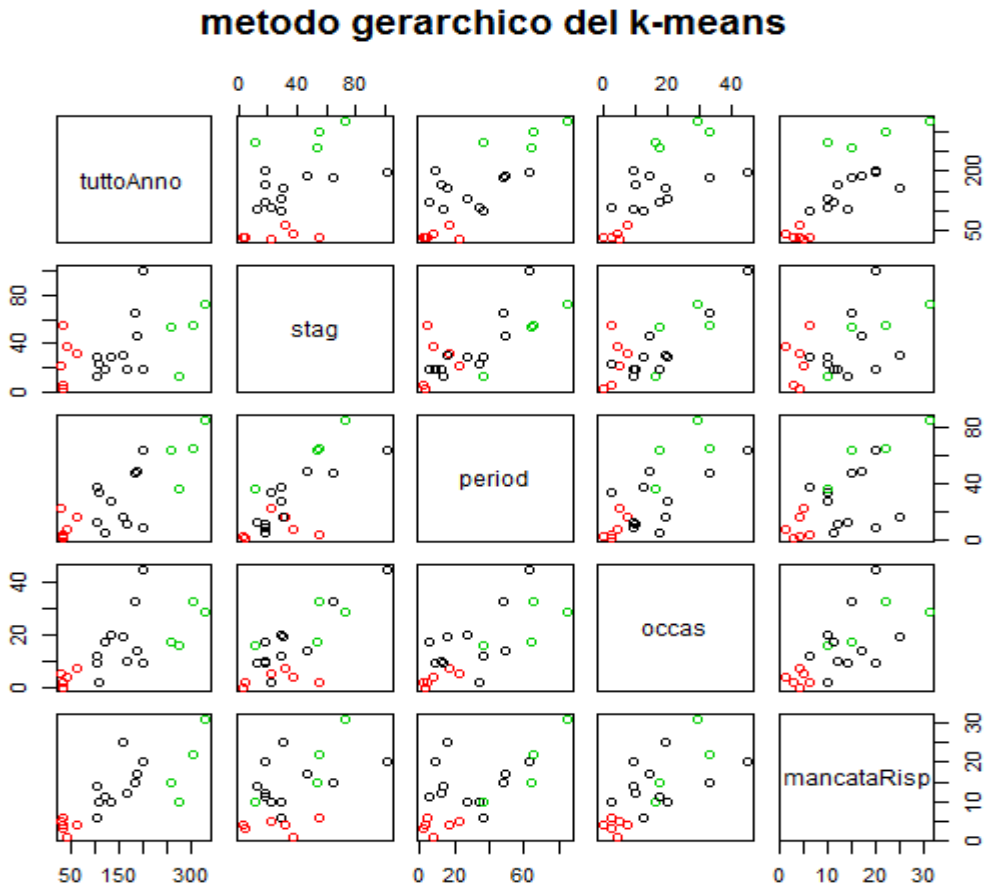
```
K-means clustering with 3 clusters of sizes 11, 6, 4

Cluster means:
  tuttoAnno     stag   period     occas mancataRisp
1 150.63636 35.81818 28.45455 17.272727   14.545455
2  37.33333 25.83333  9.50000  3.333333    3.833333
3 291.50000 48.50000 62.50000 23.750000   19.500000

Clustering vector:
             Piemonte         Valle d'Aosta               Liguria
                    1                     2                     1
            Lombardia          p.a. Bolzano          p.a. Trento
                    3                     2                     2
               Veneto Friuli-Venezia-Giulia       Emilia-Romagna
                    1                     1                     3
              Toscana               Umbria               Marche
                    3                     1                     1
                Lazio              Abruzzo               Molise
                    3                     2                     2
             Campania               Puglia           Basilicata
                    1                     1                     2
             Calabria              Sicilia            Sardegna
                    1                     1                     1

Within cluster sum of squares by cluster:
[1] 27725.818  3167.833  6666.750
 (between_SS / total_SS =  81.5 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

Osservazione: se centers fosse uguale a 3 la scelta dei punti di riferimento sarebbe casuale. Scegliendo come punti di riferimento i centroidi ottenuti con il metodo gerarchico del centroide non avremmo più una scelta casuale.

La nostra scelta è quella di suddividere in 3 cluster. Le partizioni in quattro, cinque, sei, sette, ecc presentano rapporti soddisfacenti, ma si perderebbe l'utilità dell'utilizzo dei cluster, in quanto i sottoinsiemi diventerebbero troppo piccoli. Pertanto, suddividendo in 3 cluster riusciamo ad ottenere un compromesso tra misure di non omogeneità ottimali e partizioni non troppo piccole.

Di seguito viene mostrata la partizione ottenuta dal metodo k-means:

Mostriamo graficamente le partizioni:



Procediamo con l'utilizzare i metodi gerarchici considerando una suddivisione in 3 cluster, ossia la suddivisione ottimale che abbiamo ricavato dai metodi non gerarchici.

Metodi di ottimizzazione gerarchici

I metodi gerarchici sono metodi agglomerativi, ossia partono da una situazione in cui si hanno tanti cluster quanti sono gli individui, dove ogni cluster contiene un solo individuo, per poi giungere, attraverso successive unioni dei cluster meno distanti tra di loro, ad una situazione in cui tutti gli individui sono in un unico cluster. Quindi si parte da una matrice delle distanze e si individua la coppia di individui meno distanti tra di loro e la si raggruppa in un unico cluster. Una volta creato questo nuovo cluster, si calcola la distanza tra questo cluster e tra gli altri gruppi già esistenti.

Ci sono molti metodi per determinare la distanza che intercorre tra il cluster creato e quelli già esistenti:

1. Metodo del legame singolo;
2. Metodo del legame completo;
3. Metodo del legame medio;
4. Metodo del centroide;
5. Metodo della mediana;
6. Metodo di Lance e Williams.

Tra questi vogliamo trovare quello che più si avvicina alla soluzione non gerarchica, ossia quello migliore.

Partiamo col considerare il metodo del legame singolo:

consideriamo il dataframe precedentemente creato

	tuttoAnno	stag	period	occas	mancataRisp
Piemonte	198	101	63	45	20
Valle d'Aosta	29	22	23	5	5
Liguria	131	29	27	20	10
Lombardia	259	54	64	17	15
p.a. Bolzano	31	55	4	2	6
p.a. Trento	40	38	8	4	1
Veneto	188	47	49	14	17
Friuli-Venezia-Giulia	101	29	37	12	6
Emilia-Romagna	303	55	65	33	22
Toscana	330	73	85	29	31
Umbria	107	23	34	2	10
Marche	184	65	48	33	15
Lazio	274	12	36	16	10
Abruzzo	61	32	17	7	4
Molise	32	3	3	0	4
Campania	166	19	12	10	12
Puglia	104	13	13	9	14
Basilicata	31	5	2	2	3
Calabria	120	19	5	17	11
Sicilia	200	19	9	9	20
Sardegna	158	30	16	19	25

non c'è bisogno di scalare, in quanto non abbiamo a che fare con unità di misura diverse.

Calcoliamo la matrice delle distanze (per rendere più chiara la lettura consideriamo la matrice delle distanze creata nel paragrafo precedente).

Per ogni metodo gerarchico calcoliamo le misure di non omogeneità e le riporteremo in una tabella alla fine di questo paragrafo.

Nel metodo del legame singolo la distanza tra due gruppi è definita come la minima tra tutte le distanze dei due gruppi che si possono calcolare tra ogni individuo del primo gruppo e ogni individuo del secondo gruppo. Applichiamo il metodo gerarchico del legame singolo alla matrice delle distanze:

```
> hls<-hclust(d, method="single")
```

```
> str(hls)
```

```
List of 7
 $ merge      : int [1:20, 1:2] -15 -8 -5 -17 -16 -14 2 -2 -7 -3 ...
 $ height     : num [1:20] 3.32 14.04 20.37 20.71 21.24 ...
 $ order      : int [1:21] 1 7 12 15 18 2 14 5 6 20 ...
 $ labels     : chr [1:21] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" ..
 $ method     : chr "single"
 $ call       : language hclust(d = distanze, method = "single")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

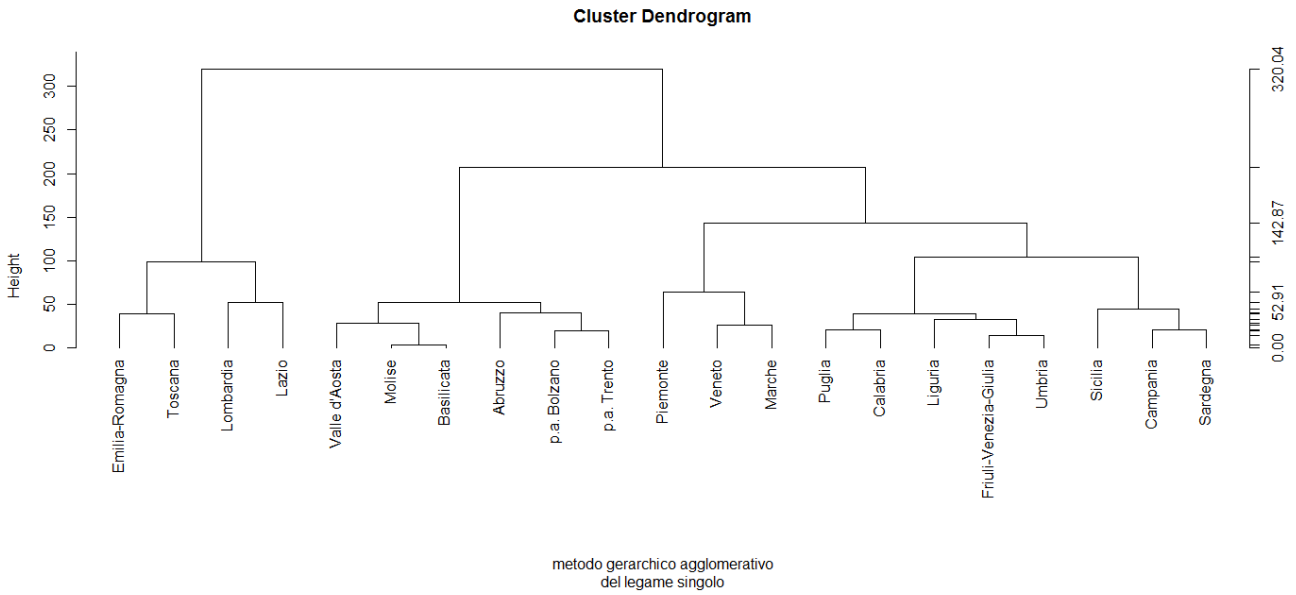
$merge indica i raggruppamenti. I numeri con segno negativo indicano i singoli individui mentre i numeri positivi indicano i cluster che si formano;

$height indica le distanze a cui sono avvenute le agglomerazioni;

$order è una permutazione delle regioni per costruire il dendrogramma;

$labels sono le etichette.

Ora, costruiamo il dendrogramma che indica la sequenza delle partizioni avvenute:



Di seguito riportiamo il calcolo in R delle misure di non omogeneità nel caso singolo. Per gli altri metodi il codice è lo stesso, tranne per il fatto di usare una matrice delle distanze al quadrato per il metodo del centroide e mediana.

```
#calcolo misure di non omogeneità
> hls<-hclust(d, method="single")
> taglio<-cutree(hls, k=3, h=NULL)
> num<-table(taglio)
> tagliolist<-list(taglio)
> agvar<-aggregate(df, tagliolist, var)[ ,-1]
> trh1<-(num[[1]]-1)*sum(agvar[1,])
> trh2<-(num[[2]]-1)*sum(agvar[2,])
> trh3<-(num[[3]]-1)*sum(agvar[3,])
> within<-trh1+trh2+trh3
> bet<-trHI-within
> ris<-bet/trHI
> ris*100
```

Rifacciamo la stessa cosa usando il metodo gerarchico del legame completo, che prende la massima tra tutte le distanze che si possono calcolare in due gruppi.

```
> hlc<-hclust(d, method="complete")
```

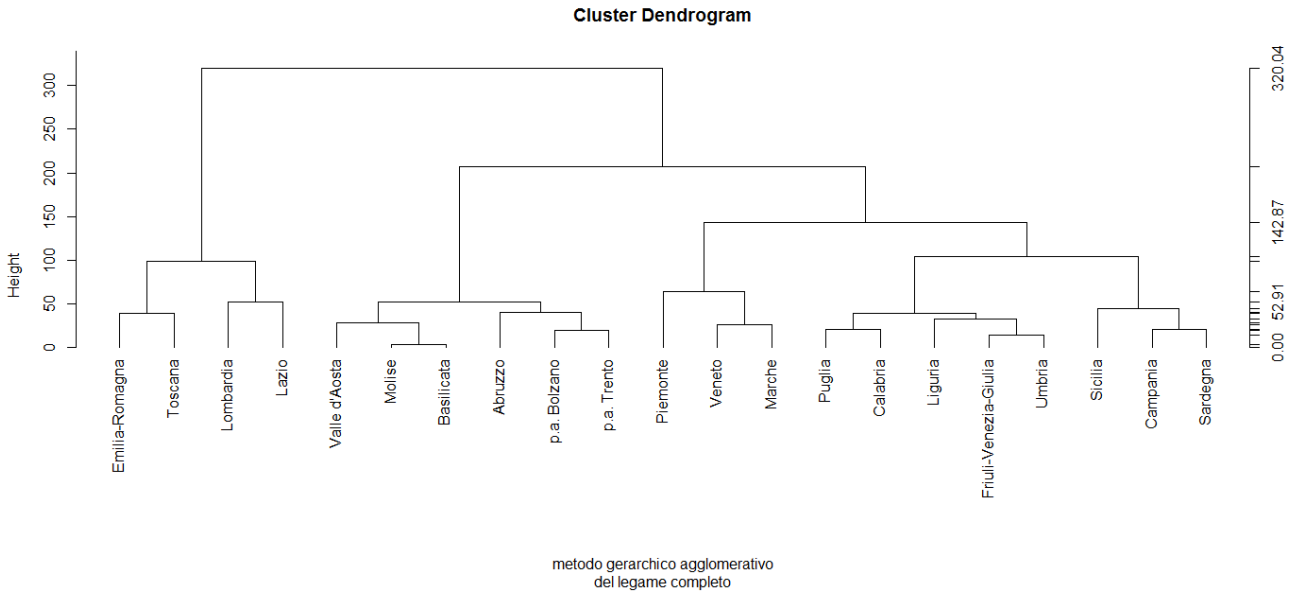
```
> str(hlc)
```

```
List of 7
 $ merge      : int [1:20, 1:2] -15 -8 -5 -17 -16 -7 -2 -3 4 -9 ...
 $ height     : num [1:20] 3.32 14.04 20.37 20.71 21.24 ...
 $ order      : int [1:21] 9 10 4 13 2 15 18 14 5 6 ...
 $ labels     : chr [1:21] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" .
 $ method     : chr "complete"
 $ call       : language hclust(d = distanze, method = "complete")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

Costruiamo il dendrogramma:

```
> plot(hlc, hang=-1, xlab="metodo gerarchico agglomerativo", sub="del legame completo")
```

```
> axis(side=4, at=round(c(0,hlc$height),2))
```



Notiamo il dendrogramma del legame singolo è simile al dendrogramma del legame completo

Procediamo con l'utilizzo del metodo gerarchico del legame medio.

Il metodo gerarchico del legame medio calcola la media aritmetica delle distanze tra tutte le coppie di unità che compongono i due cluster.

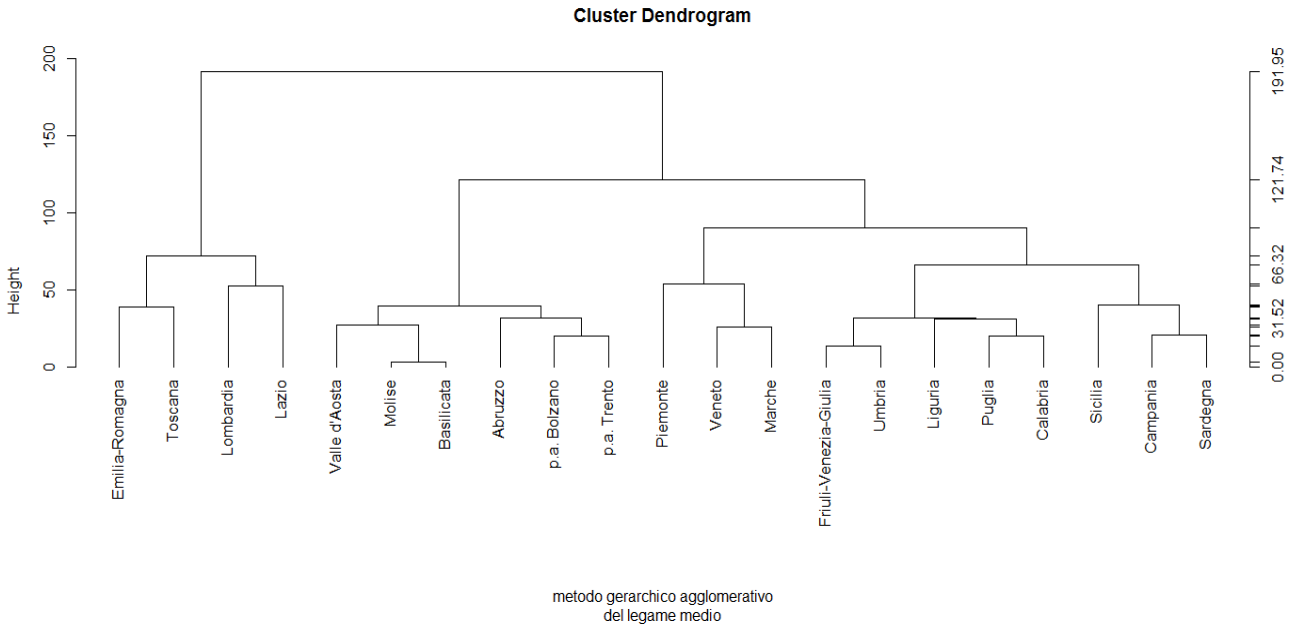
```
> hlm<-hclust(d, method="average")
```

```
> str(hlm)
```

```
List of 7
 $ merge      : int [1:20, 1:2] -15 -8 -5 -17 -16 -7 -2 -3 -14 2 ...
 $ height     : num [1:20] 3.32 14.04 20.37 20.71 21.24 ...
 $ order      : int [1:21] 9 10 4 13 2 15 18 14 5 6 ...
 $ labels     : chr [1:21] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "average"
 $ call       : language hclust(d = distanze, method = "average")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

```
> plot(hlm, hang=-1, xlab="metodo gerarchico agglomerativo", sub="del legame medio")
```

```
> axis(side=4, at=round(c(0, hlm$height), 2))
```



Tale dendrogramma è molto simile ai due precedenti. Tuttavia, i salti di questo risultano avere valori intermedi tra quelli del legame singolo e quelli del legame del completo. Passiamo ora al metodo del centroide.

Nei precedenti metodi agglomerativi si può utilizzare una qualsiasi misura di distanza. Nel metodo del centroide e nel metodo della mediana si considera la distanza euclidea e si lavora con una matrice delle distanze che contiene i quadrati delle singole distanze eulidee. Nel metodo del centroide la distanza tra due gruppi è definita come la distanza tra i centroidi, ossia le medie campionarie calcolate sugli individui appartenenti ai due gruppi.

Per prima cosa calcoliamo la matrice delle distanze al quadrato:

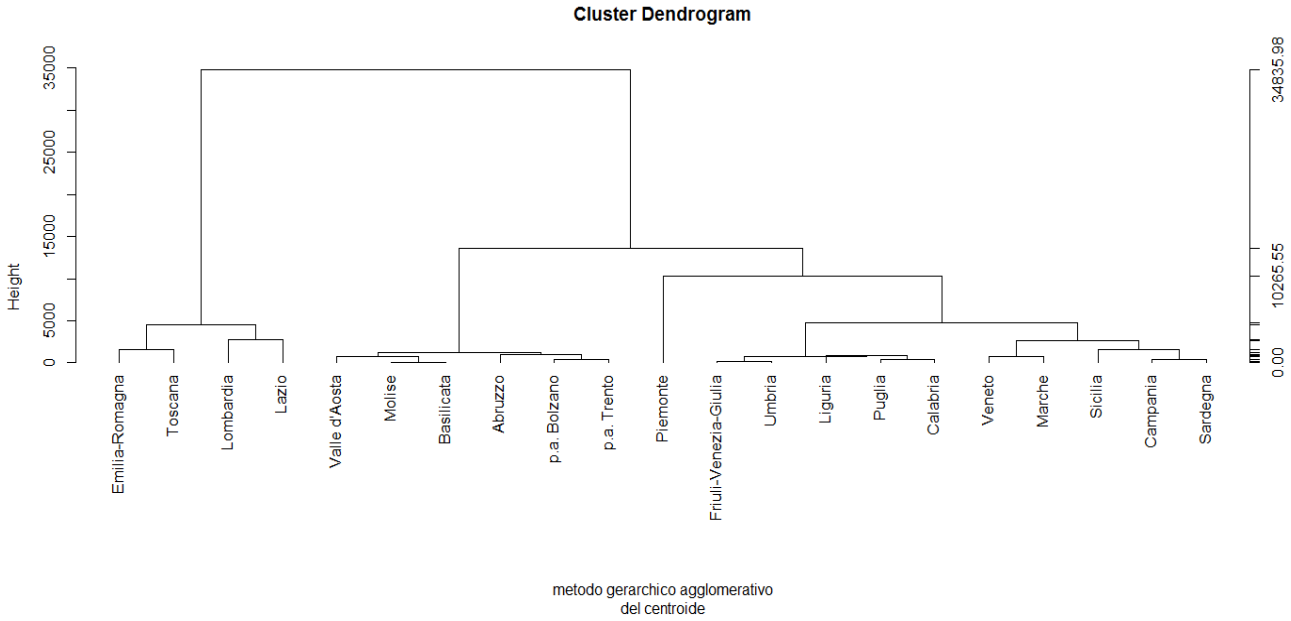
```
> hc<-hclust(d2, method="centroid")
```

```
> str(hc)
```

```
List of 7
 $ merge      : int [1:20, 1:2] -15 -8 -5 -17 -16 -7 -2 -3 2 -14 ...
 $ height     : num [1:20] 11 197 415 429 451 ...
 $ order      : int [1:21] 9 10 4 13 2 15 18 14 5 6 ...
 $ labels     : chr [1:21] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "centroid"
 $ call       : language hclust(d = distanze2, method = "centroid")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

```
> plot(hc, hang=-1, xlab="metodo gerarchico agglomerativo", sub="del centroide")
```

```
> axis(side=4, at=round(c(0, hc$height),2))
```



Metodo della mediana

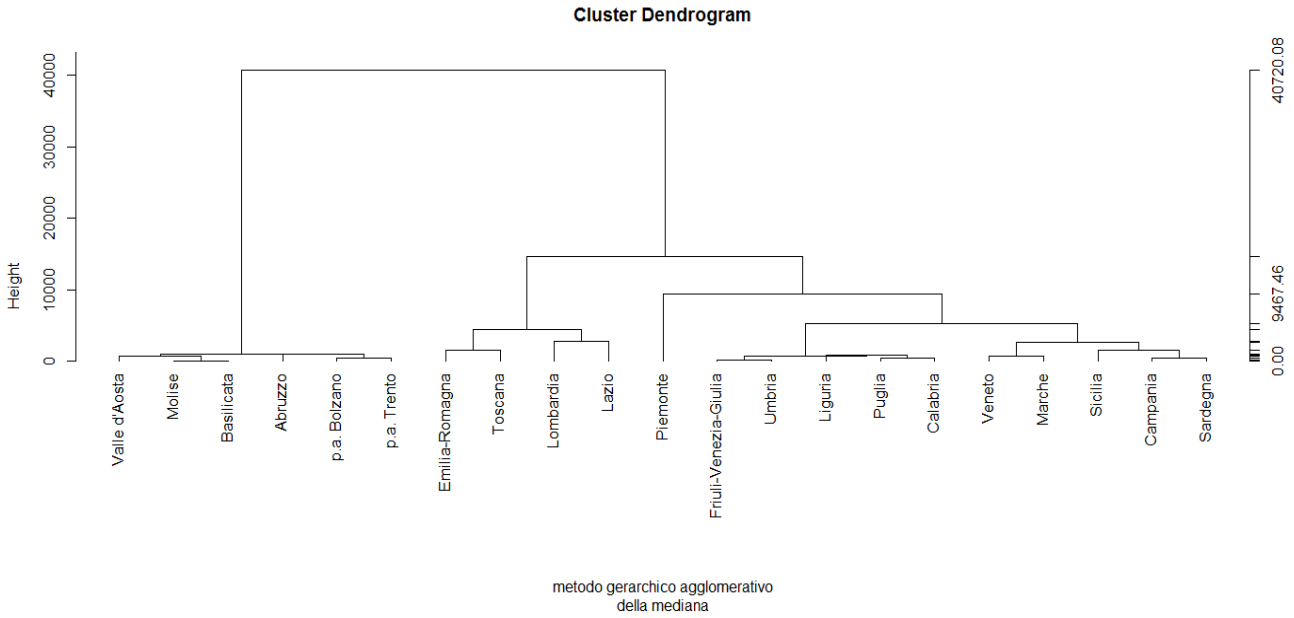
Questo metodo è simile a quello del centroide, ma è indipendente dalla numerosità dei cluster. Anche questo metodo, come quello del legame singolo può dare origine a delle catene.

```
> hmed<-hclust(d2, method="median")
```

```
> str(hmed)
```

```
List of 7
 $ merge      : int [1:20, 1:2] -15 -8 -5 -17 -16 -7 -2 -3 2 -14 ...
 $ height     : num [1:20] 11 197 415 429 451 ...
 $ order      : int [1:21] 2 15 18 14 5 6 9 10 4 13 ...
 $ labels     : chr [1:21] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "median"
 $ call       : language hclust(d = distanze2, method = "median")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

```
> plot(hmed, hang=-1, xlab="metodo gerarchico agglomerativo", sub="della mediana")
> axis(side=4, at=round(c(0, hmed$height),2))
```



Questo dendrogramma è molto simile al precedente. Cambia la permutazione delle regioni.

Metodo di Lance e Williams

Questo metodo è ricorsivo, infatti il calcolo della matrice dei quadrati delle distanze dipende unicamente dalla medesima matrice al livello precedente, esso include tutti i metodi precedentemente visti. Nella sua formula infatti compaiono dei coefficienti che variano a seconda del metodo che si vuole utilizzare.

Tabella riassuntiva B/T:

tipo	Numero di cluster	Distanza	aggregazione	B/T
k-means	3	-	-	81.5%
MG	3	Euclidea	Single	57.1%
MG	3	Euclidea	Complete	81.5%
MG	3	Euclidea	Average	81.5%
MG	3	Euclidea	Centroid	81.5%
MG	3	Euclidea	Median	81.5%

Possiamo, quindi, applicare tutti i metodi gerarchici, ad eccezione del metodo del legame singolo.

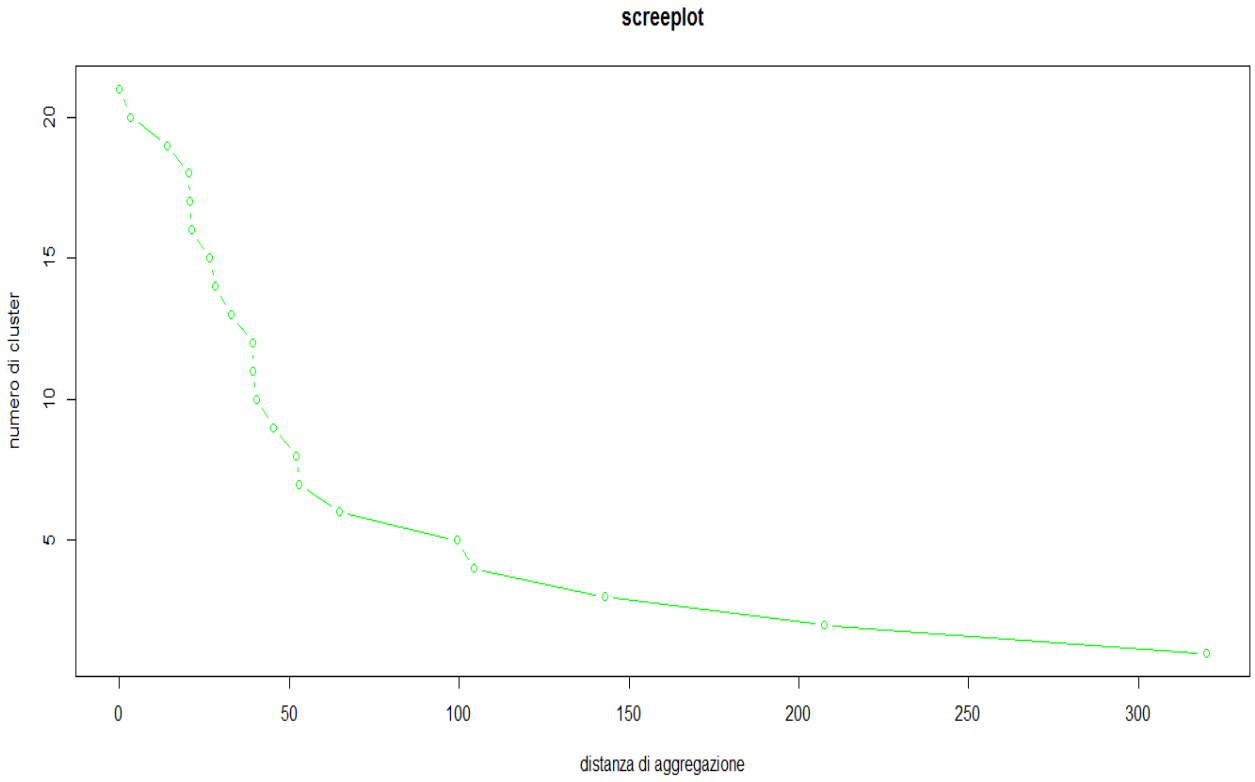
Screeplot

Un altro modo che possiamo utilizzare per ottenere una buona partizione del dendrogramma è quello di usare lo screeplot che è un grafico in cui sull'asse delle ordinate si pongono il numero di cluster ottenibili dal metodo gerarchico, mentre sull'asse delle ascisse le distanze di aggregazione. Se nel passaggio da k gruppi a k-1 gruppi si registra un forte incremento della distanza di aggregazione è consigliabile tagliare il dendrogramma in k gruppi. Questa procedura, però, non fornisce sempre la suddivisione in cluster più adeguata. **Conviene sempre calcolare le misure di non omogeneità statistiche.**

È consigliabile costruire tale grafico a partire dal metodo del legame singolo, legame completo o medio. Gli altri metodi potrebbero non essere regolari e non fornire informazioni adeguate.

Consideriamo il dendrogramma precedentemente calcolato usando il metodo gerarchico del legame completo e costruiamo lo screeplot relativo:

```
> plot(rev(c(0, hlc$height)), seq(1,21), type="b", main="screeplot", xlab="distanza di aggregazione", ylab="numero di cluster", col="green")
```

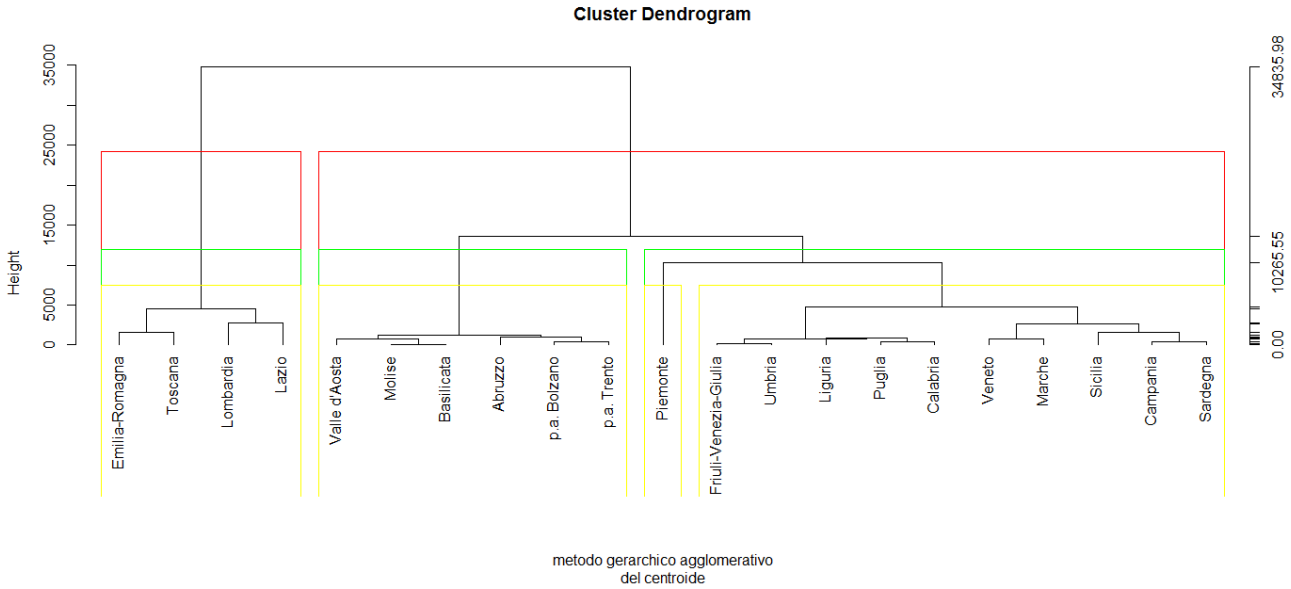



```
> c(0, hlc$height)
 [1]   0.000000   3.316625  14.035669  20.371549  20.712315  21.236761  26.570661  28.213472
 [9]  32.863353  39.179076  39.370039  40.336088  45.376205  52.096065  52.905576  64.668385
[17]  99.337808 104.355163 142.870571 207.463250 320.042185
```

Analisi del dendrogramma

Consideriamo il dendrogramma ottenuto usando il metodo del centroide. Disegniamo dei rettangoli che evidenziano i cluster.

```
> hc<-hclust(d2, method="centroid")
> plot(hc, hang=-1, xlab="metodo gerarchico agglomerativo", sub="del centroide")
> axis(side=4, at=round(c(0, hc$height), 2))
> rect.hclust(hc, k=2, border="red")
> rect.hclust(hc, k=3, border="green")
> rect.hclust(hc, k=4, border="yellow")
```



Ci aspettavamo proprio che l'Emilia-Romagna, Lazio, Toscana e Lombardia appartenessero allo stesso gruppo in quanto sono le regioni che presentano numeri più elevati.

```
> cutree(hc, k=3, h=NULL)
            Piemonte         Valle d'Aosta               Liguria             Lombardia
                   1                     2                     1                     3
        p.a. Bolzano          p.a. Trento               Veneto Friuli-Venezia-Giulia
                   2                     2                     1                     1
      Emilia-Romagna               Toscana               Umbria               Marche
                   3                     3                     1                     1
               Lazio              Abruzzo               Molise             Campania
                   3                     2                     2                     1
             Puglia           Basilicata             Calabria              Sicilia
                   1                     2                     1                     1
           Sardegna
                   1
```

Facciamo una fotografia alla situazione in cui si hanno da 1 a 10 cluster:

```
> cutree(hc, k=1:10)
                      1 2 3 4 5 6 7 8 9 10
Piemonte              1 1 1 1 1 1 1 1 1  1
Valle d'Aosta         1 1 2 2 2 2 2 2 2  2
Liguria               1 1 1 3 3 3 3 3 3  3
Lombardia             1 2 3 4 4 4 4 4 4  4
p.a. Bolzano          1 1 2 2 2 2 2 2 2  2
p.a. Trento           1 1 2 2 2 2 2 2 2  2
Veneto                1 1 1 3 5 5 5 5 5  5
Friuli-Venezia-Giulia 1 1 1 3 3 3 3 3 3  3
Emilia-Romagna        1 2 3 4 4 6 6 6 6  6
Toscana               1 2 3 4 4 6 6 6 7  7
Umbria                1 1 1 3 3 3 3 3 3  3
Marche                1 1 1 3 5 5 5 5 5  5
Lazio                 1 2 3 4 4 4 7 7 8  8
Abruzzo               1 1 2 2 2 2 2 2 2  2
Molise                1 1 2 2 2 2 2 2 2  2
Campania              1 1 1 3 5 5 5 8 9  9
Puglia                1 1 1 3 3 3 3 3 3  3
Basilicata            1 1 2 2 2 2 2 2 2  2
Calabria              1 1 1 3 3 3 3 3 3  3
Sicilia               1 1 1 3 5 5 5 8 9 10
Sardegna              1 1 1 3 5 5 5 8 9  9
```

Man mano notiamo come affiorano i dettagli.

Ora siamo interessati a conoscere media, varianza e deviazione standard campionaria di questi nuovi gruppi ottenuti. Facciamo questo con il comando aggregate.

```
> taglio<-cutree(hc, k=3, h=NULL)
> tagliolist<-list(taglio)
> aggregate(df, tagliolist, mean)
  Group.1 tuttoAnno     stag   period     occas mancataRisp
1       1 150.63636 35.81818 28.45455 17.272727   14.545455
2       2  37.33333 25.83333  9.50000  3.333333    3.833333
3       3 291.50000 48.50000 62.50000 23.750000   19.500000
> aggregate(df, tagliolist, var)
  Group.1 tuttoAnno     stag   period     occas mancataRisp
1       1 1534.6545 688.5636 369.6727 148.818182   30.872727
2       2  149.0667 401.3667  73.9000   6.266667    2.966667
3       3  992.3333 668.3333 405.6667  72.916667   83.000000
> aggregate(df, tagliolist, sd)
  Group.1 tuttoAnno     stag    period     occas mancataRisp
1       1  39.17467 26.24050 19.226875 12.199106    5.556323
2       2  12.20929 20.03414  8.596511  2.503331    1.722401
3       3  31.50132 25.85214 20.141168  8.539126    9.110434
```

Fine prima parte.