UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

PROGETTO INTERNO PER
LAUREA MAGISTRALE IN
INFORMATICA

Progetto Statistica e Analisi dei Dati

Docente	Studenti	
Prof.ssa	**Marrazzo Vincenzo**	**Zizzari Antonio**
Amelia Giuseppina Nobile	**0522501325**	**0522501309**

Anno Accademico 2021/2022

Indice

1. *Introduzione*

Lo scopo dell'indagine statistica presentata in questo progetto è quello di analizzare i dati forniti dell' **UNIVERSITY REPORT 2021** sullo stipendio lordo annuale dei più importanti atenei italiani rispetto all'età.
L'obiettivo è ottenere informazioni dalle quali ricavare ulteriore conoscenza e poter fare considerazioni sul fenomeno analizzato.
Un'analisi approfondita sullo stipendio della popolazione italiana che ha conseguito la laurea presso diversi atenei può essere molto interessante per fornire dati affinché si possa ampliare queste analisi e dare un contributo, basato su dati oggettivi, all'importantissimo e sempre attuale dibattito sul ruolo dell'istruzione terziaria nel percorso professionale dei giovani.

1.1 Dataset

Il dataset considerato è una raccolta di dati scomposti per ateneo che riflettono gli andamenti della crescita economica di un paese, l'Italia. I primi quattro atenei in classifica sono o privati o situati al Nord del paese, o entrambi: l'Università Commerciale Luigi Bocconi (34.662 euro), il Politecnico di Milano (32.308 euro), la Libera università internazionale degli studi sociali Guido Carli (31.870 euro) e l'Università Cattolica del Sacro Cuore (31.735 euro). Gli ultimi nella lista sono gli atenei di Perugia (29.013 euro) e Cagliari (28.706 euro).

Il campione considerato è composto da persone ordinate per fasce d'età (costituite dalle colonne), e per ciascun ateneo è indicato lo stipendio medio lordo.
Di seguito, dunque, la tabella:

ATENEO	25-34 anni	35-44 anni	45-54 anni	da 25-34 a 45-54
Università Cattolica del Sacro Cuore	31,7	42,0	58,0	82,8
LUISS Libera università internazionale degli studi sociali Guido Carli	31,9	42,2	57,1	79,0
Università Commerciale Luigi Bocconi	34,7	44,8	59,3	71,2
Politecnico di Torino	31,1	41,2	52,8	69,7
Università degli Studi di Perugia	29,0	37,4	48,9	68,6
Università degli Studi di Verona	29,7	37,7	49,8	67,6
Politecnico di Milano	32,3	41,7	53,7	66,2
Università degli Studi di Brescia	30,5	39,4	50,7	66,1
Università degli Studi di Modena e Reggio Emilia	30,2	40,2	50,1	66,1
Università degli Studi di Bergamo	30,3	38,0	49,7	64,1
Università degli Studi di Milano	29,9	38,2	49,0	64,0
Università di Roma La Sapienza	30,3	38,0	49,3	62,9
Università degli Studi di Parma	30,7	39,1	50,0	62,6
Università degli Studi di Pisa	30,6	38,8	49,7	62,5
Università Politecnica delle Marche	30,3	38,0	48,8	61,1
Alma mater studiorum Università di Bologna	30,0	38,0	48,2	60,6
Università Ca Foscari di Venezia	29,7	38,7	47,3	59,3
Università degli Studi di Roma Tor Vergata	31,1	39,0	49,4	59,1
Università degli Studi di Padova	30,7	39,7	48,7	58,8
Università degli Studi di Siena	31,1	38,0	49,4	58,7
Università degli Studi di Trieste	30,3	37,8	48,1	58,4
Università degli Studi di Udine	30,7	38,4	48,3	57,2
Università degli Studi di Genova	30,5	38,1	47,7	56,4
Università degli Studi di Pavia	30,9	38,5	48,3	56,2
Università degli Studi di Catania	29,6	37,0	46,0	55,5
Università degli Studi di Trento	30,5	39,0	47,3	55,1
Università degli Studi Roma Tre	30,5	38,2	47,3	55,0
Università degli Studi di Torino	30,0	36,8	46,4	54,5
Università degli Studi dell'Aquila	29,9	37,5	46,2	54,4
Università degli Studi di Bari	29,1	35,7	44,9	54,2
Università degli Studi di Cagliari	28,7	35,4	44,2	54,0
Università degli Studi di Milano Bicocca	29,9	38,4	46,0	53,9
Politecnico di Bari	30,5	38,4	46,6	52,8
Università degli Studi di Ferrara	29,6	38,8	44,9	51,8
Università degli Studi di Firenze	29,6	37,4	44,7	50,8
Università degli Studi di Palermo	30,2	36,5	45,1	49,4
Università degli Studi di Napoli Federico II	30,6	37,3	44,8	46,4
Università degli Studi di Messina	29,1	35,8	42,5	46,2
Università degli Studi di Napoli Parthenope	29,5	36,1	43,1	46,0
Università degli Studi della Calabria	30,0	36,0	43,1	43,6

Per la realizzazione del progetto si è scelto di utilizzare un linguaggio di programmazione R.

1.2 Inizializzazione dati matrice

Una azione necessaria è quella di inizializzare la matrice che verrà utilizzata durante l'indagine statistica.

Per ciascuna colonna creeremo un array:

```
atenei<-c("Cattolica del Sacro Cuore","LUISS Guido Carli",
          "Luigi Bocconi","P.Torino","Perugia","Verona",
          "P.Milano","Brescia","Modena e Reggio Emilia",
          "Bergamo","U.Milano","La Sapienza","Parma","Pisa",
          "Marche","Bologna","Venezia","Roma Tor Vergata",
          "Padova","Siena","Trieste","Udine","Genova","Pavia",
          "Catania","Trento","Roma Tre","U.Torino","Aquila",
          "U.Bari","Cagliari","Bicocca","P.Bari","Ferrara",
          "Firenze","Palermo","Federico II","Messina","Parthenope",
          "Calabria")

anni.25to34<-c(31.7,31.9,34.7,31.1,29.0,29.7,32.3,30.5,30.2,30.3,
               29.9,30.3,30.7,30.6,30.3,30.0,29.7,31.1,30.7,31.1,
               30.3,30.7,30.5,30.9,29.6,30.5,30.5,30.0,29.9,29.1,
               28.7,29.9,30.5,29.6,29.6,30.2,30.6,29.1,29.5,30.0)

anni.35to44<-c(42.0,42.2,44.8,41.2,37.4,37.7,41.7,39.4,40.2,38.0,
               38.2,38.0,39.1,38.8,38.0,38.0,38.7,39.0,39.7,38.0,
               37.8,38.4,38.1,38.5,37.0,39.0,38.2,36.8,37.5,35.7,
               35.4,38.4,38.4,38.8,37.4,36.5,37.3,35.8,36.1,36.0)

anni.45to54<-c(58.0,57.1,59.3,52.8,48.9,49.8,53.7,50.7,50.1,49.7,
               49.0,49.3,50.0,49.7,48.8,48.2,47.3,49.4,48.7,49.4,
               48.1,48.3,47.7,48.3,46.0,47.3,47.3,46.4,46.2,44.9,
               44.2,46.0,46.6,44.9,44.7,45.1,44.8,42.5,43.1,43.1)

anni.from.25to34.and.45to54<-c(82.8,79.0,71.2,69.7,68.6,67.6,66.2,
                               66.1,66.1,64.1,64.0,62.9,62.6,62.5,
                               61.1,60.6,59.3,59.1,58.8,58.7,58.4,
                               57.2,56.4,56.2,55.5,55.1,55.0,54.5,
                               54.4,54.2,54.0,53.9,52.8,51.8,50.8,
                               49.4,46.4,46.2,46.0,43.6)
```

A questo punto è possibile procedere con la creazione della matrice contenente tutti gli array definiti precedentemente:

```
mtxStipendiLordi<-cbind(anni.25to34,anni.35to44,anni.45to54,
                        anni.from.25to34.and.45to54)
rownames(mtxStipendiLordi)<- atenei
titoli<-c("(25-34)","(35-44)","(45-54)","(Da 25-34 a 45-54)")
colnames(mtxStipendiLordi)<-titoli
mtxStipendiLordi
```

La matrice risultante dalle operazioni precedenti è la seguente:

	(25-34)	(35-44)	(45-54)	(Da 25-34 a 45-54)
Cattolica del Sacro Cuore	31.7	42.0	58.0	82.8
LUISS Guido Carli	31.9	42.2	57.1	79.0
Luigi Bocconi	34.7	44.8	59.3	71.2
P.Torino	31.1	41.2	52.8	69.7
Perugia	29.0	37.4	48.9	68.6
Verona	29.7	37.7	49.8	67.6
P.Milano	32.3	41.7	53.7	66.2
Brescia	30.5	39.4	50.7	66.1
Modena e Reggio Emilia	30.2	40.2	50.1	66.1
Bergamo	30.3	38.0	49.7	64.1
U.Milano	29.9	38.2	49.0	64.0
La Sapienza	30.3	38.0	49.3	62.9
Parma	30.7	39.1	50.0	62.6
Pisa	30.6	38.8	49.7	62.5
Marche	30.3	38.0	48.8	61.1
Bologna	30.0	38.0	48.2	60.6
Venezia	29.7	38.7	47.3	59.3
Roma Tor Vergata	31.1	39.0	49.4	59.1
Padova	30.7	39.7	48.7	58.8
Siena	31.1	38.0	49.4	58.7
Trieste	30.3	37.8	48.1	58.4
Udine	30.7	38.4	48.3	57.2
Genova	30.5	38.1	47.7	56.4
Pavia	30.9	38.5	48.3	56.2
Catania	29.6	37.0	46.0	55.5
Trento	30.5	39.0	47.3	55.1
Roma Tre	30.5	38.2	47.3	55.0
U.Torino	30.0	36.8	46.4	54.5
Aquila	29.9	37.5	46.2	54.4
U.Bari	29.1	35.7	44.9	54.2
Cagliari	28.7	35.4	44.2	54.0
Bicocca	29.9	38.4	46.0	53.9
P.Bari	30.5	38.4	46.6	52.8
Ferrara	29.6	38.8	44.9	51.8
Firenze	29.6	37.4	44.7	50.8
Palermo	30.2	36.5	45.1	49.4
Federico II	30.6	37.3	44.8	46.4
Messina	29.1	35.8	42.5	46.2
Parthenope	29.5	36.1	43.1	46.0
Calabria	30.0	36.0	43.1	43.6

2. *Distribuzione di frequenza*

Il primo approccio è la predisposizione e l'organizzazione dei dati tramite la distribuzione di frequenza. Essa si occuperà di organizzarli in maniera che siano significativi, in modo tale da poter facilitare estrapolazione di informazioni che possono sembrare non immediatamente evidenti.
Nel nostro caso la soluzione ideale è stata quella di raccogliere le informazioni in classi utilizzando la funzione `quantile()`.
La scelta sulle classi è stata fatta dunque nel seguente modo:

```
(28.0,35.2] (35.2,42.8] (42.8,50.7] (50.7,82.8]
```

Che abbiamo ottenuto dal seguente codice in R:

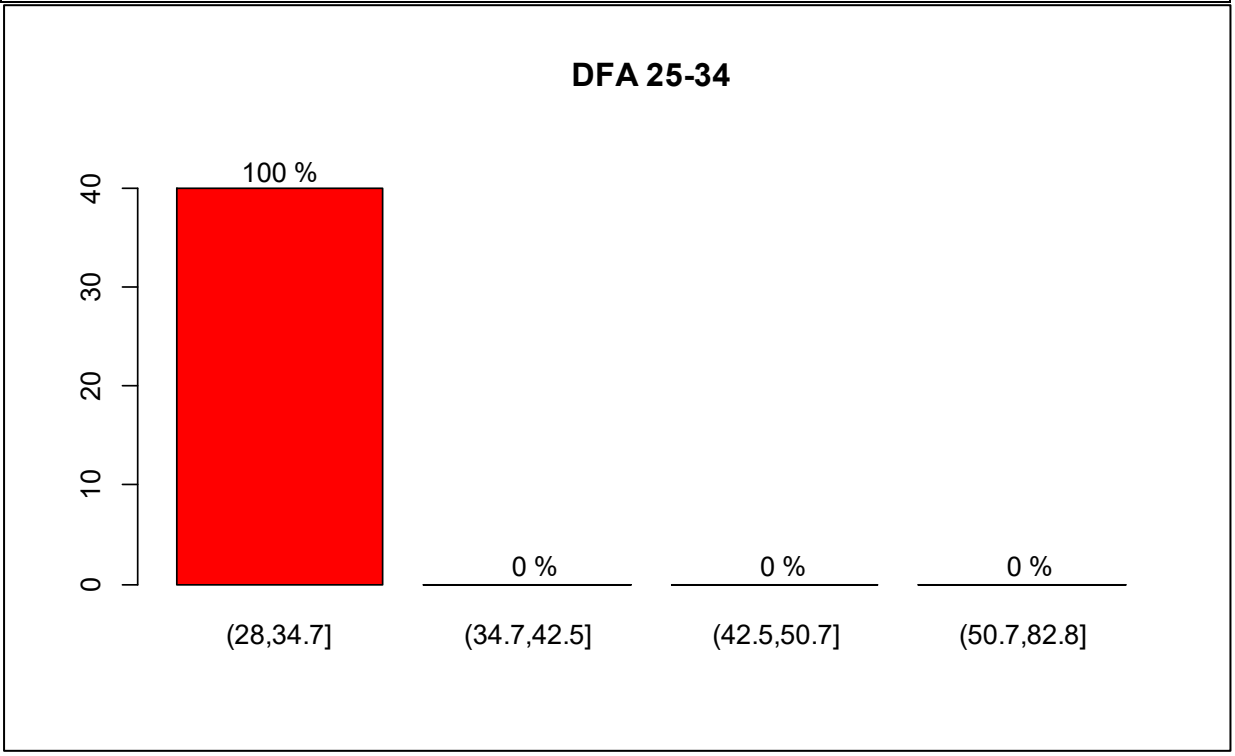
```
classe<-quantile(c(28,anni.25to34, anni.35to44, anni.45to54,
anni.from.25to34.and.45to54))
```

Per lavorare con intervalli l'ambiente R ci fornisce la funzione `cut()` che prende in input un vettore e gli estremi degli intervalli considerati.
Per calcolare la frequenza sull'output di cut utilizziamo invece la funzione `table()`.
Vediamo dunque la **distribuzione di frequenza assoluta** e la **distribuzione di frequenza relativa**.

Utilizziamo i grafici a barre per graficare i risultati.

2.1 Fascia d'età 25-34

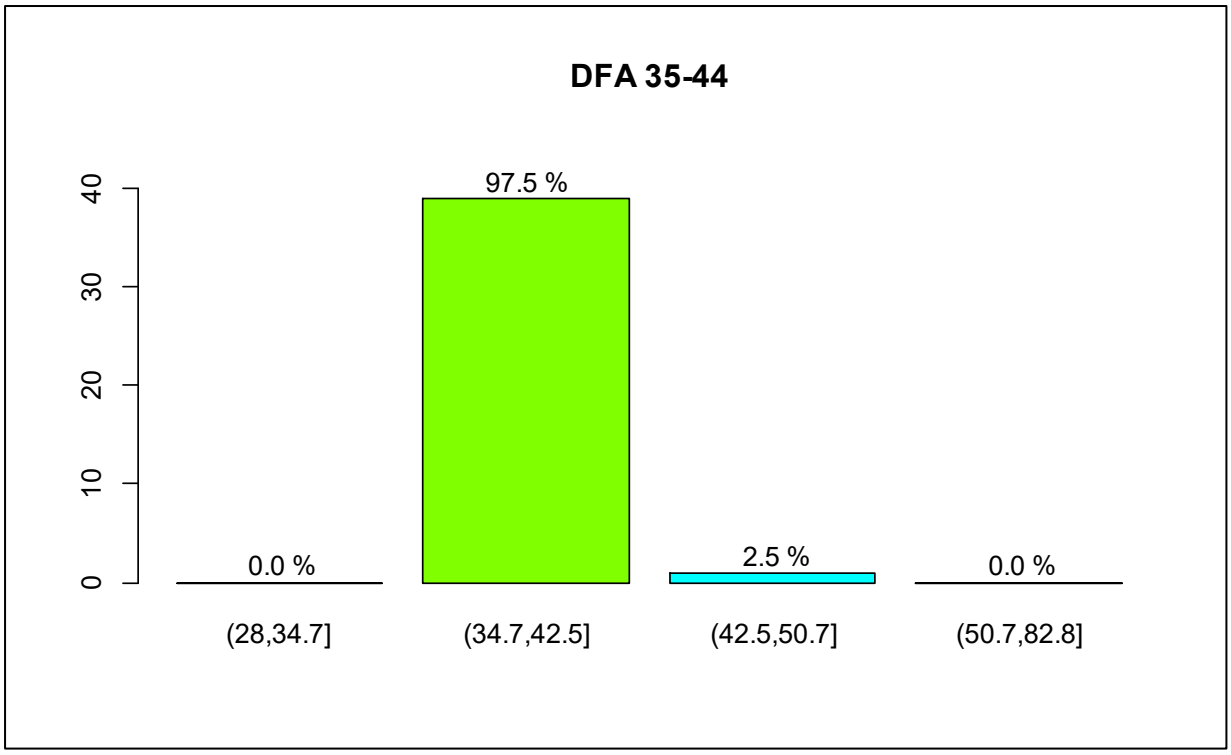
```
DFA1<-table(cut(anni.25to34,classe))
p1<-DFA1/length(DFA1)*10
x1<-barplot(DFA1,main="DFA 25-34",col=rainbow(4),ylim=c(0, 45))
text(x1, DFA1+2, labels = paste(format(p1,digits=2),"%"))
```



Da questo risultato capiamo che la fascia d'età tra i 25 e i 34 anni tutte le università offrono in media una carriera lavorativa con lo stesso stipendio.

2.2 Fascia d'età 35-44

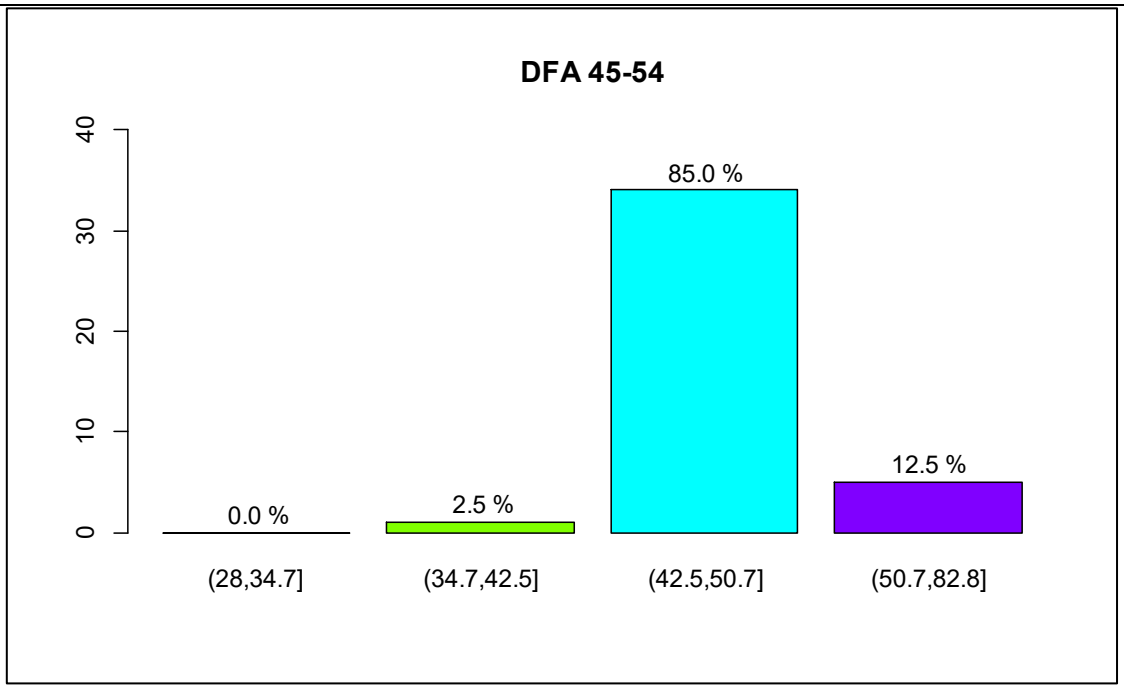
```
DFA2<-table(cut(anni.35to44,classe))
p1<-DFA1/length(DFA1)*10
x2<-barplot(DFA2,main="DFA 35-44",col=rainbow(4),ylim=c(0, 45))
text(x2, DFA2+2, labels = paste(format(p2,digits=2),"%"))
```



Da questo risultato capiamo che la fascia d'età tra i 35 e i 44 anni nel quadro generale italiano tutte le università offrono una crescita della carriera in modo omogeneo, fatta eccezione per Università Commerciale Luigi Bocconi e LUISS Libera università internazionale degli studi sociali Guido Carli che si attestano su un valore superiore alla media.

2.3 Fascia d'età 45-54

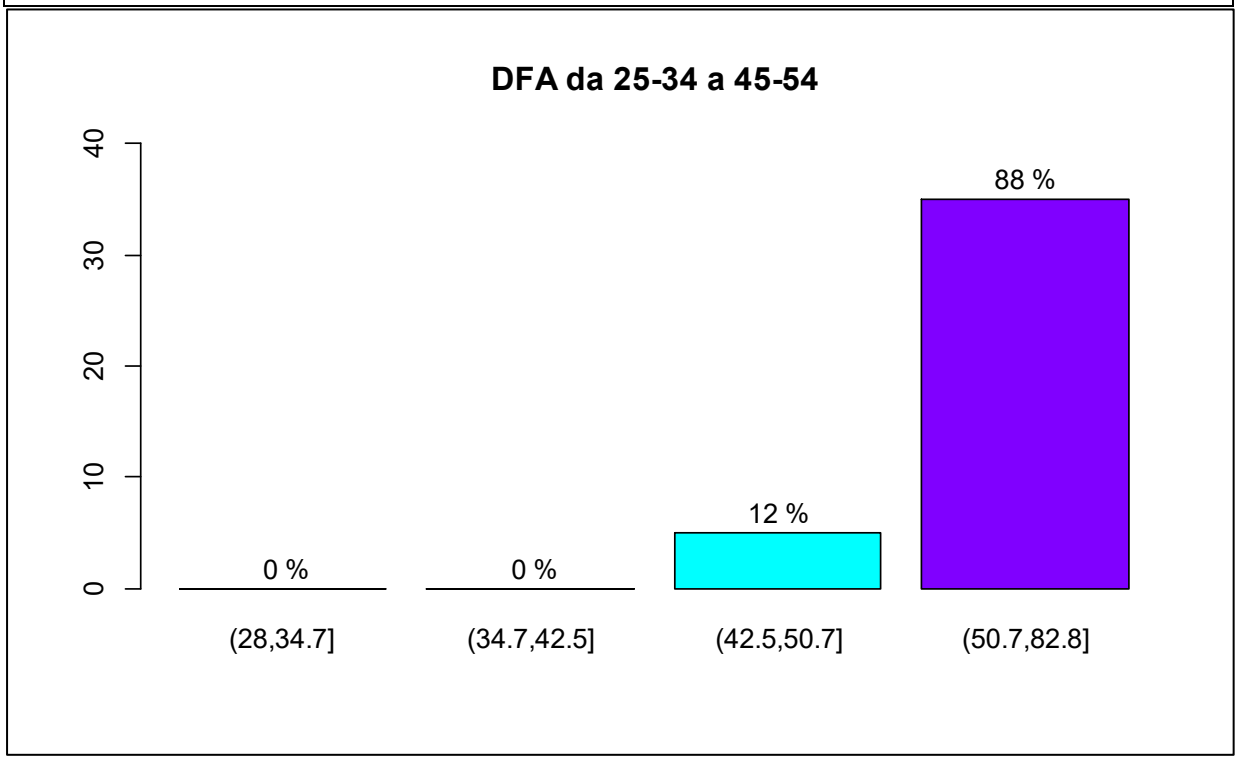
```
DFA3<-table(cut(anni.45to54,classe))
p3<-DFA3/length(DFA3)*10
x3<-barplot(DFA3,main="DFA 45-54",col=rainbow(4),ylim=c(0, 40))
text(x3, DFA3+2, labels = paste(format(p3,digits=2),"%"))
```



Da questo risultato capiamo che la fascia d'età tra i 45 e i 54 anni è leggermente divaricante, infatti si ha che il 2,5% degli stipendi medi è inferiore alla media ed esso appartiene alla università che si trovano nel Sud Italia. Invece al Nord troviamo una percentuale del 12,5% che supera il valore medio ed esso appartiene alla quasi totalità delle università al Nord Italia

2.4 Fascia d'età da 25-34 a 45-54

```
DFA4<-table(cut(anni.from.25to34.and.45to54,classe))
p4<-DFA4/length(DFA4)*10
x4<-barplot(DFA4,main="DFA da 25-34 a 45-54",col=rainbow(4),ylim=c(0, 40))
text(x4, DFA4+2, labels = paste(format(p4,digits=2),"%"))
```



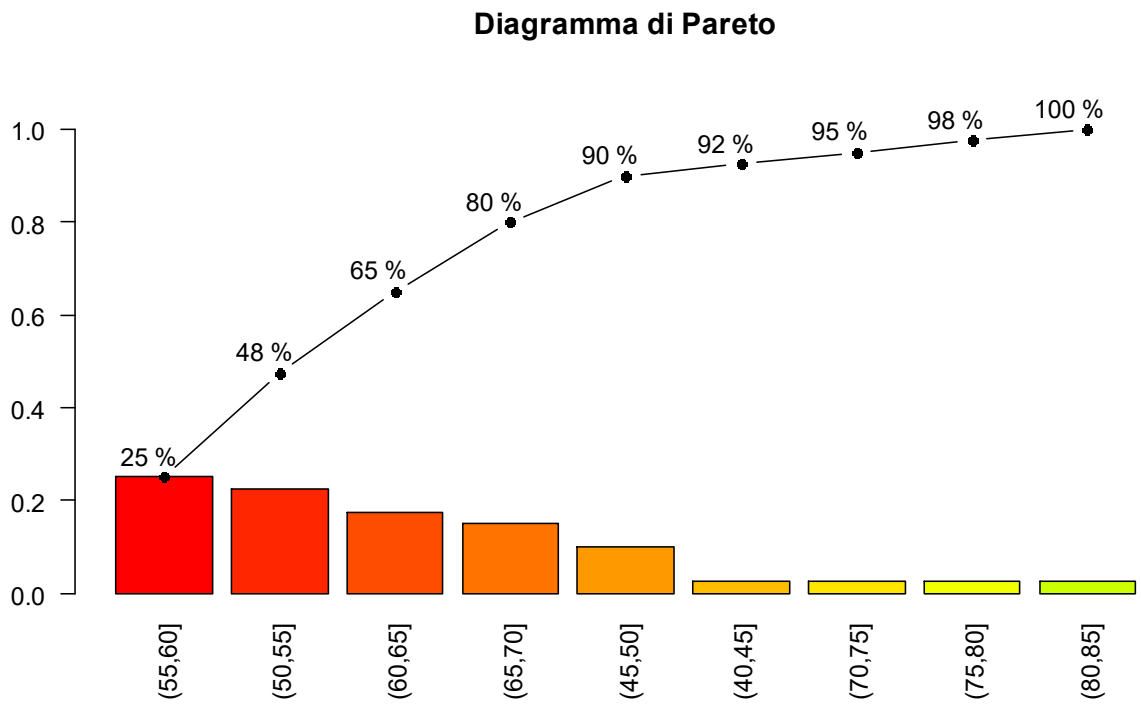
Da questo risultato capiamo che la fascia d'età da 25-54 a 45-54 ha un grafico che è abbastanza omogeneo, però il restante 12% appartiene, anche in questo caso, prevalentemente alle università del Sud.

3. Diagramma di Pareto

Il diagramma di Pareto ha la capacità di mostrare immediatamente quali modalità si presentano con maggiore frequenza e può aiutare a stabilire quali di esse principalmente influenzano un determinato fenomeno. Una caratteristica importante dell'analisi di Pareto è legata alla sua versatilità e facilità di applicazione in ogni campo in cui occorre analizzare aspetti di qualità, di efficienza, di sicurezza, di affidabilità, di costi, ecc.
Esso quindi è un utile strumento di analisi nei processi decisionali, in particolare si è analizzato lo stipendio per ateneo e si è valutata la frequenza relativa e la percentuale cumulata in relazione all'80/20 % dell'analisi.

In R si ha il seguente codice:

```
allAnni<-c(anni.from.25to34.and.45to54)
pareto<-hist(allAnni,freq=TRUE)

taba<-table(cut(allAnni,pareto$breaks))

ordinato<-(sort(taba,decreasing = TRUE))
propord<-prop.table(ordinato)
x<-barplot(propord,ylim=c(0,1.1),main="Diagramma di Pareto",col=rainbow(40),las=2)
lines(x,cumsum(propord),type = "b",pch=16)
text(x-0.2, cumsum(propord)+0.05, paste(format(cumsum(propord)*100, digits=2), "%"))
```



L'80 % degli stipendi più ricorrenti è costituito dai seguenti 32 atenei:

```
 [1] "P.Torino"              "Perugia"               "Verona"
 [4] "P.Milano"              "Brescia"               "Modena e Reggio Emilia"
 [7] "Bergamo"               "U.Milano"              "La Sapienza"
[10] "Parma"                 "Pisa"                  "Marche"
[13] "Bologna"               "Venezia"               "Roma Tor Vergata"
[16] "Padova"                "Siena"                 "Trieste"
[19] "Udine"                 "Genova"                "Pavia"
[22] "Catania"               "Trento"                "Roma Tre"
[25] "U.Torino"              "Aquila"                "U.Bari"
[28] "Cagliari"              "Bicocca"               "P.Bari"
[31] "Ferrara"               "Firenze"
```

E il restante 20 %, costituente la minoranza, risultano essere i seguenti 8 atenei:

```
[1] "Cattolica del Sacro Cuore" "LUISS Guido Carli"         "Luigi Bocconi"
[4] "Palermo"                   "Federico II"               "Messina"
[7] "Parthenope"                "Calabria"
```

4. *Barplot per vettori*

Attraverso la funzione barplot possiamo visualizzare l'andamento dei valori assunti dai vari stipendi. Consideriamo dunque la funzione barplot() che ci permette di visualizzare gli stipendi relativi di ogni ateneo.

È stata scritta la funzione *createBarplot* che prende in input come parametro lo stipendio relativo alla fascia d'età.
Il codice è il seguente:

```
createBarplot<-function(stipendi, main){

  x<-barplot(stipendi, ylab="Stipendi", main=main,
             col=1:40,
             las=2,
             names.arg=atenei,
             cex.axis=0.80,
             cex.names=0.80,
             font.lab=2)
  abline(h=mean(stipendi),col="red",lty=2,xpd=FALSE)

}
```

Grafico relativo alla fascia d'età 25-34:

```
createBarplot(anni.25to34, "Stipendi anni 25-34")
```

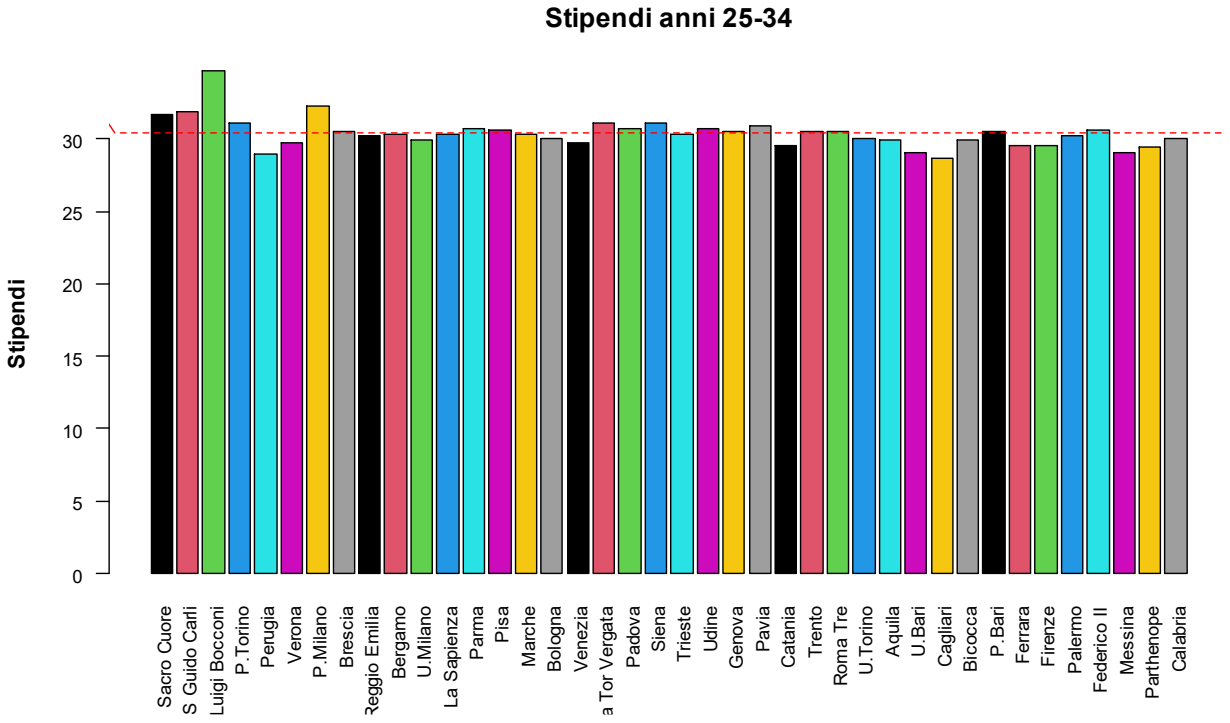


Grafico relativo alla fascia d'età 35-44:

```
createBarplot(anni.35to44, "Stipendi anni 35-44")
```

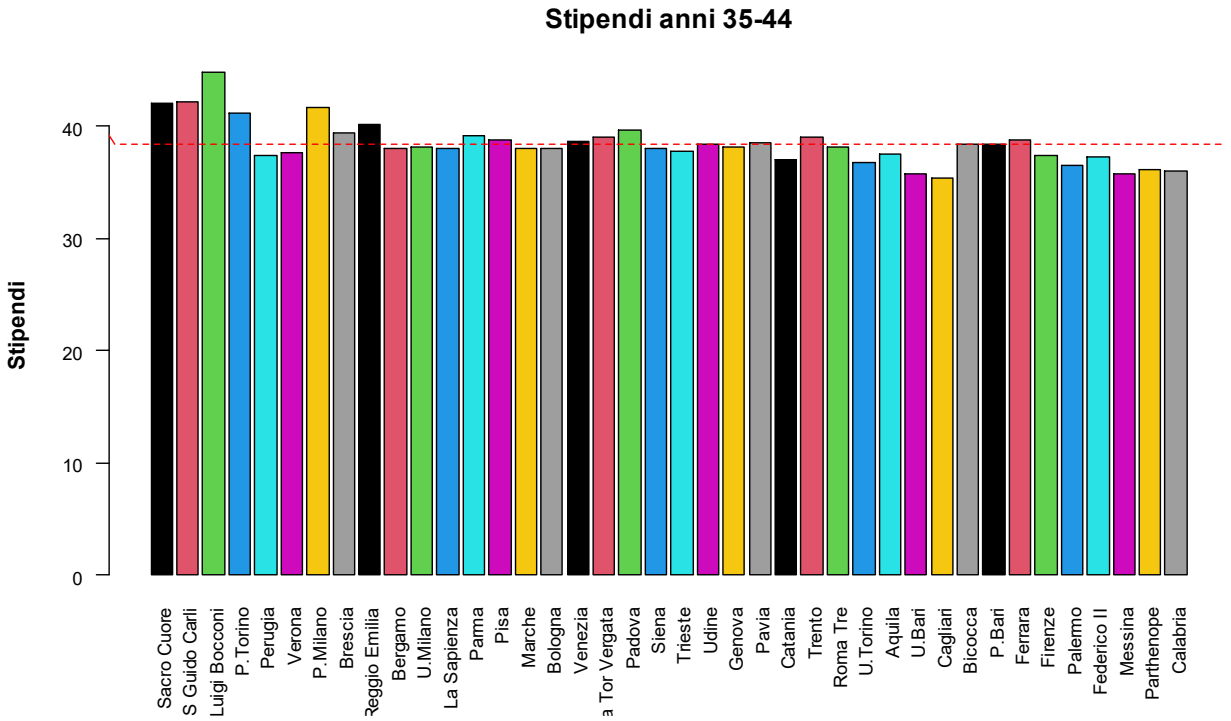


Grafico relativo alla fascia d'età 45-54:

```
createBarplot(anni.45to54, "Stipendi anni 45-54")
```

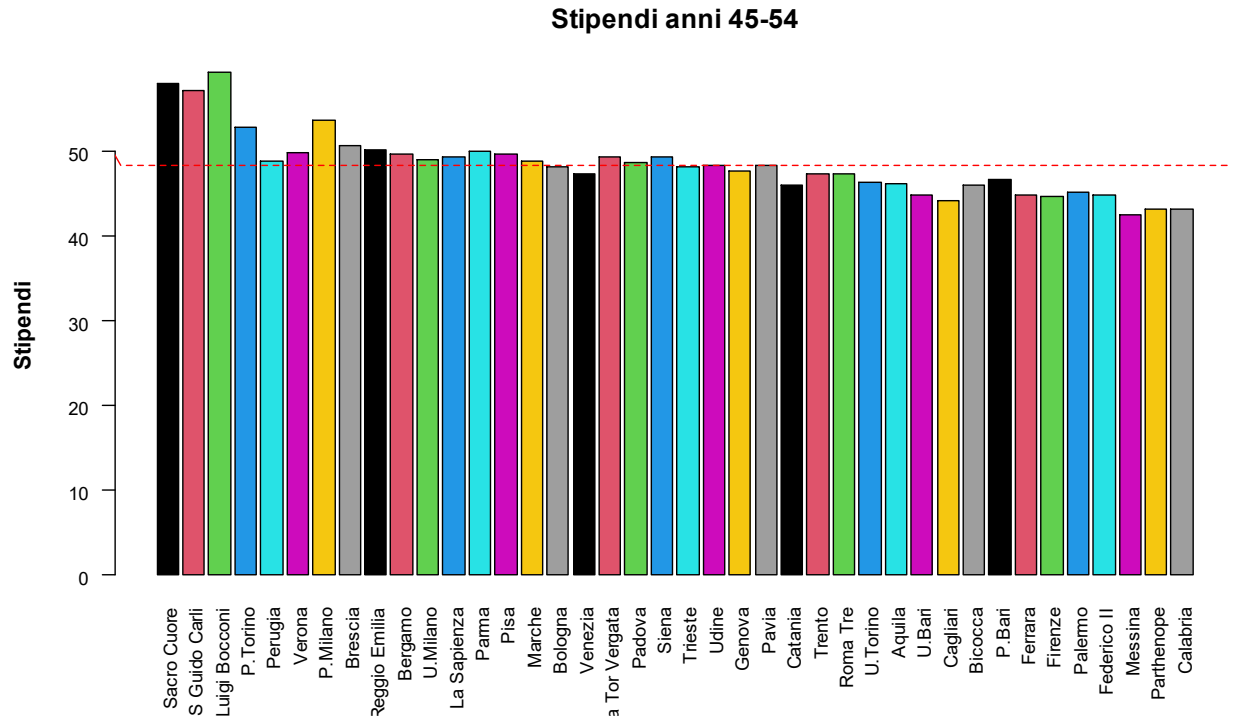
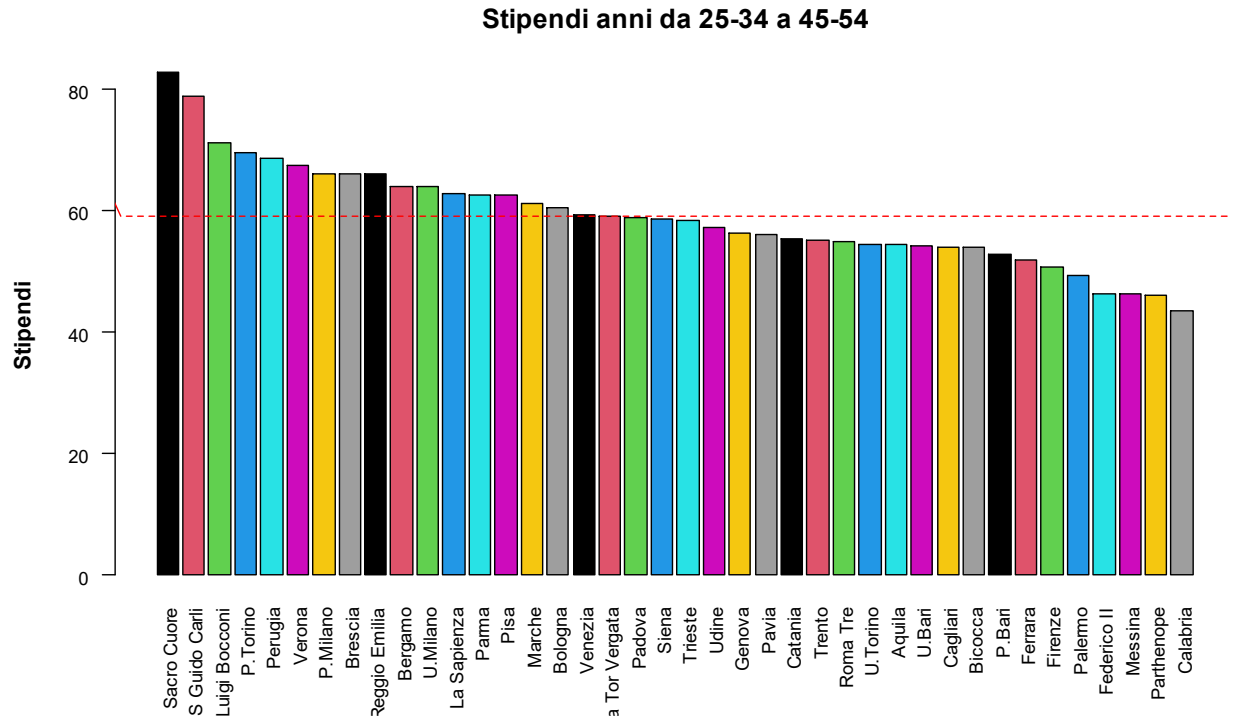


Grafico relativo alla fascia d'età da 25-34 a 45-54:

```
createBarplot(anni.from.25to34.and.45to54, "Stipendi anni da 25-34 a 45-54")
```



Anche se questi dati non hanno una rilevante importanza danno una visione completa dei dati elaborati.

5. *Istogrammi*

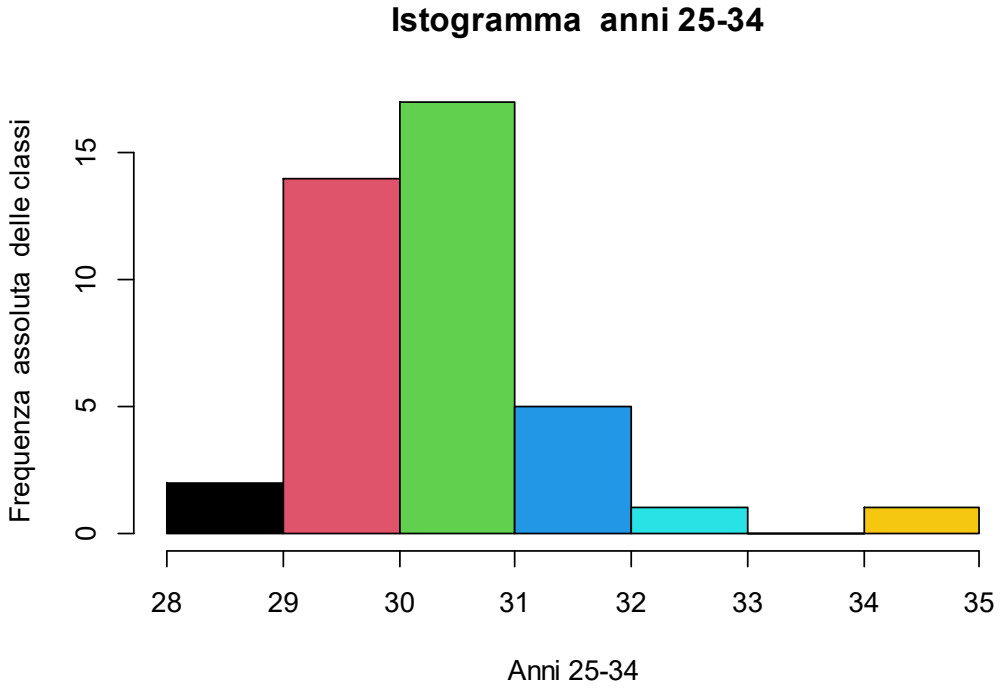
Un istogramma è una rappresentazione alternativa della distribuzione di dati numerici. Quello che si ottiene da un istogramma è una serie di rettangoli adiacenti aventi la base sull'asse delle ascisse e di altezza dipendente dalla frequenza delle classi. In R la funzione `hist()` permette di disegnare un istogramma. Il numero di classi può essere definito dall'utente, ma può anche essere lasciato a discrezione di R che deciderà il numero di classi che ritiene adeguato.

La funzione `hist()` genera oltre al grafico anche altre informazioni utili: i breaks definiti per la creazione dell'istogramma, le frequenze assolute delle classi, la densità delle classi e i punti centrali delle classi.

Vediamo adesso gli istogrammi relativi alle varie fasce d'età:

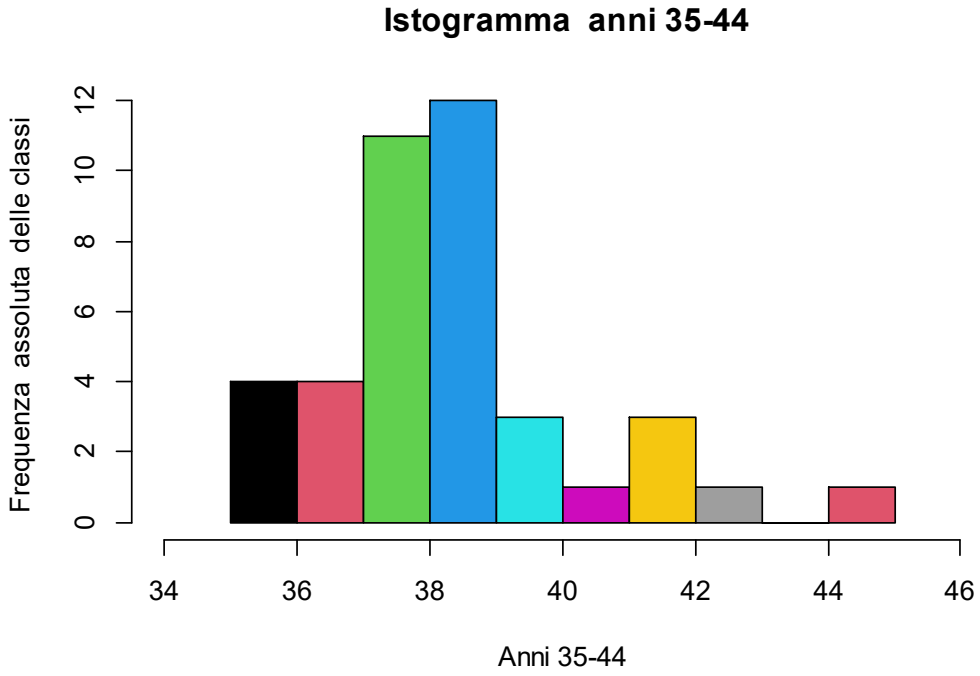
5.1 Fascia d'età 25-34

```
hist(anni.25to34 ,freq=TRUE ,main=" Istogramma  anni 25-34",
     xlab="Anni 25-34",
     ylab="Frequenza  assoluta  delle classi ",col = 1:40)
```



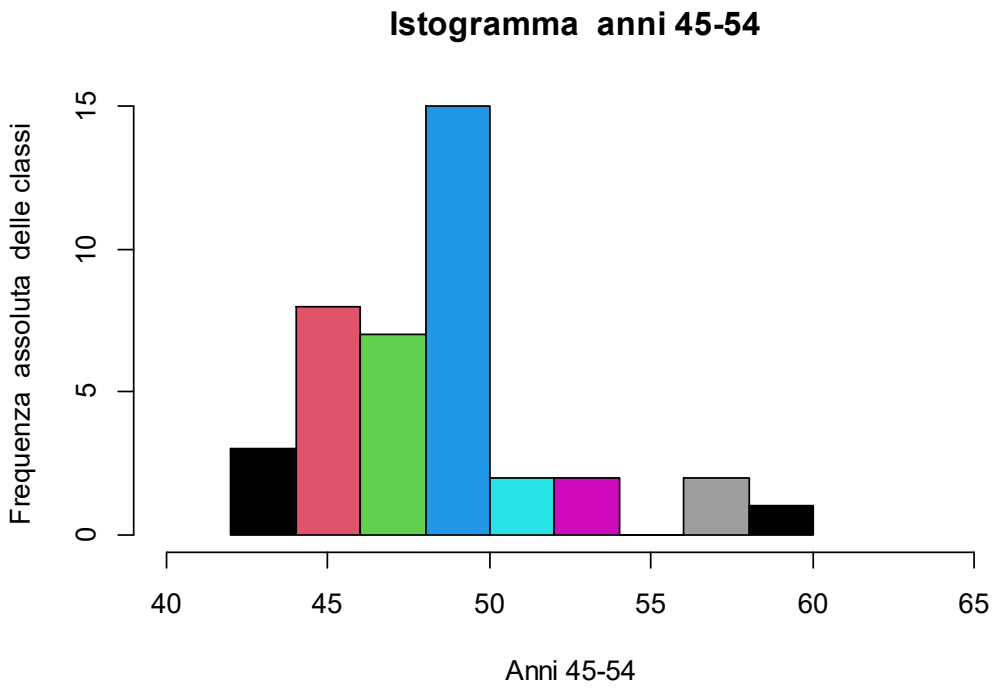
5.2 Fascia d'età 35-44

```
hist(anni.35to44 ,freq=TRUE ,main=" Istogramma  anni 35-44",xlim = c(34,46),
     xlab="Anni 35-44",
     ylab="Frequenza  assoluta  delle classi ",col = 1:40)
```



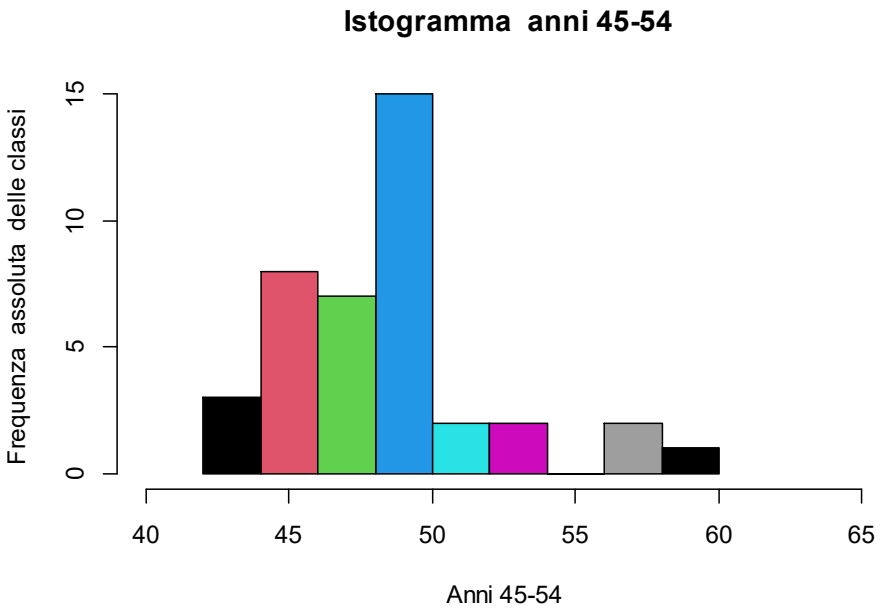
5.3 Fascia d'età 45-54

```
hist(anni.45to54 ,freq=TRUE ,main=" Istogramma  anni 45-54",xlim = c(40,65),
     xlab="Anni 45-54",
     ylab="Frequenza  assoluta  delle classi ",col = 1:40)
```



5.4 Fascia d'età da 25-34 a 45-54

```
hist(anni.from.25to34.and.45to54 ,freq=TRUE ,main=" Istogramma  anni da 25-34 a
45-54",xlim = c(40,90),
     xlab="Anni da 25-34 a 45-54",
     ylab="Frequenza  assoluta  delle classi ",col = 1:40)
```



6. *Boxplot*

Nella statistica descrittiva un boxplot è un metodo per raffigurare graficamente gruppi di dati numerici attraverso i quartili. Si procede ordinando i valori del campione in ordine crescente. Si chiama primo quartile, e si indica con Q1, il valore per il quale il 25% dei dati sono alla sua sinistra e il restante 75% alla sua destra. Analogamente si chiama terzo quartile, e si indica con Q3, il valore per il quale il 75% dei dati sono alla sua sinistra e il restante 25% alla sua destra. Il secondo quartile Q2, ossia il valore per il quale 50% dei dati sono alla sua sinistra e il restante 50% è alla sua destra è detto mediana. Q0 e Q4 forniscono il minimo e il massimo dei valori del campione. I quartili si calcolo tramite la funzione `quantile()` in R, mediante la funzione `summary()` è possibile restituire i valori precisi dei quartili.

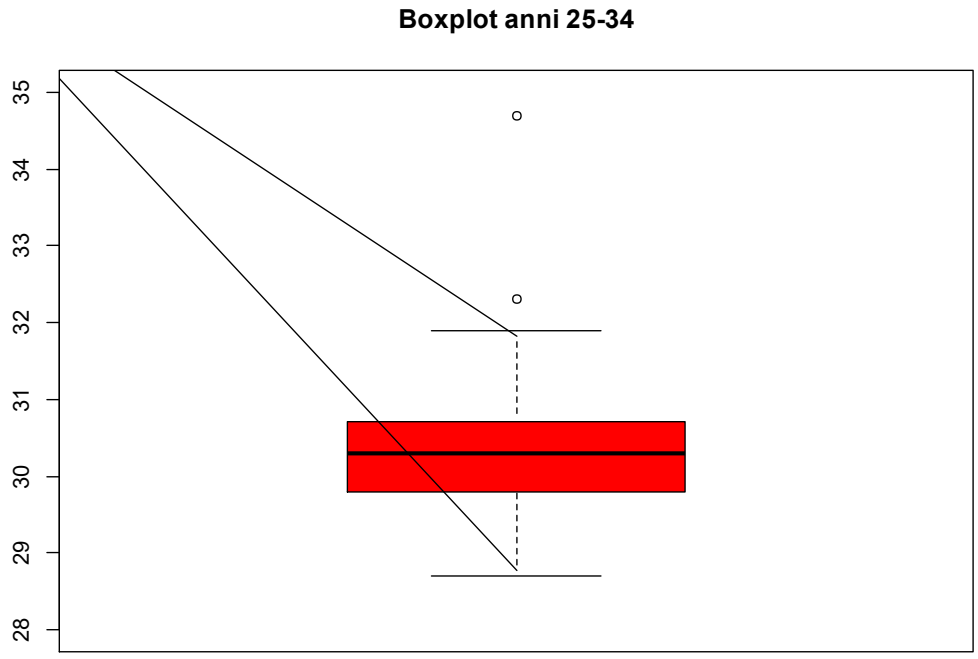
Il boxplot, detto anche scatola con baffi, è il disegno di una scatola i cui estremi sono Q1 e Q3, tagliata da una linea orizzontale in corrispondenza di Q2, ossia della mediana. In basso e in alto sono presenti altre due linee orizzontali, dette i baffi. Il baffo inferiore corrisponde al valore più 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 $Q3 + 1.5 \cdot (Q3-Q1)$. La distanza tra il primo e il terzo quartile è detta intervallo interquartile o scarto interquartile.
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 fuori dell'intervallo $(Q1 - 1.5 \cdot (Q3 - Q1), Q3 + 1.5 \cdot (Q3 - Q1))$ 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 e pertanto è necessario identificarli per poterne analizzare le caratteristiche e le eventuali cause che li hanno determinati.

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, detti "outlier". La centralità è espressa dalla mediana. La forma simmetrica o asimmetrica può essere dedotta esaminando le distanze del primo e del terzo quartile dalla linea 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.

Per ottenere le informazioni del boxplot si utilizza la funzione `str()`.
Vediamo adesso i boxplot delle relative fasce d'età:

6.1 Fascia d'età 25-34

```
b1<-boxplot(anni.25to34,main="Boxplot anni 25-34", col="red")
```



Il grafico presenta due anomalie corrispondenti agli atenei di Luigi Bocconi e Politecnico di Milano, che risultano essere maggiori del baffo superiore che è pari a 31,9 . Le informazioni ricavate sono:

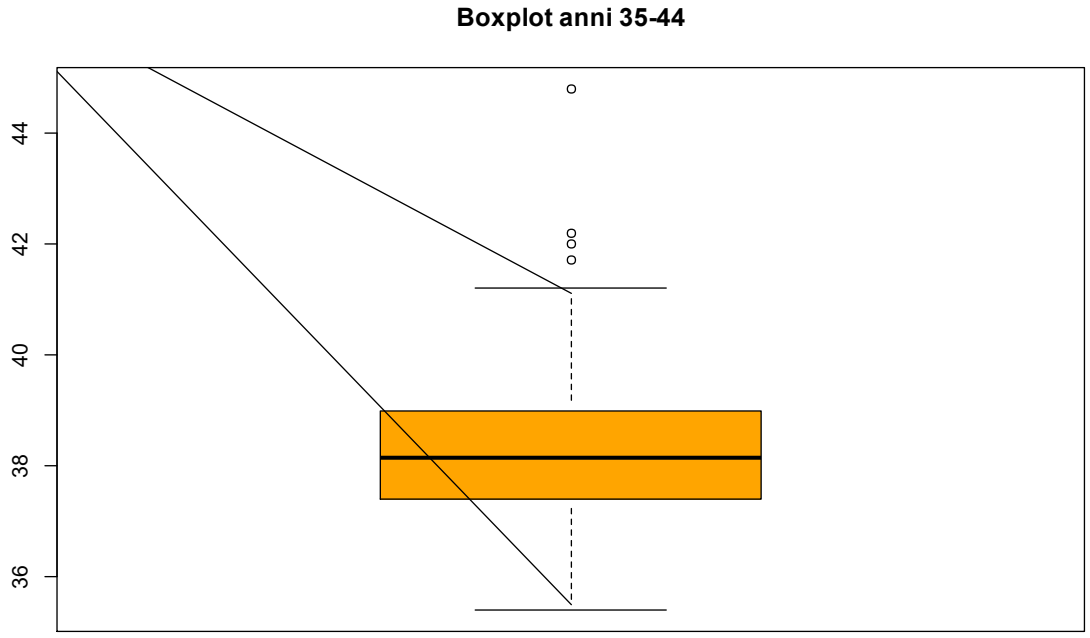
```
> str(b1)
List of 6
 $ stats: num [1:5, 1] 28.7 29.8 30.3 30.7 31.9
 $ n    : num 40
 $ conf : num [1:2, 1] 30.1 30.5
 $ out  : num [1:2] 34.7 32.3
 $ group: num [1:2] 1 1
 $ names: chr ""
```

Possiamo affermare che i valori anomali contenuti nel campo *out* sono: 34.7, 32.3 .

Poiché Q3 – Q2 = **0.4** è di poco minore a Q2 – Q1 = **0.5** allora possiamo affermare che i dati sono abbastanza simmetrici.

6.2 Fascia d'età 35-44

```
b2<-boxplot(anni.35to44,main="Boxplot anni 35-44", col="orange")
```



Il grafico presenta tre anomalie corrispondenti agli atenei: Luigi Bocconi, LUISS Guido Carli, Cattolica del Sacro Cuore e Politecnico di Milano che risultano essere maggiori del baffo superiore che è pari a 41.2 . Le informazioni ricavate sono:

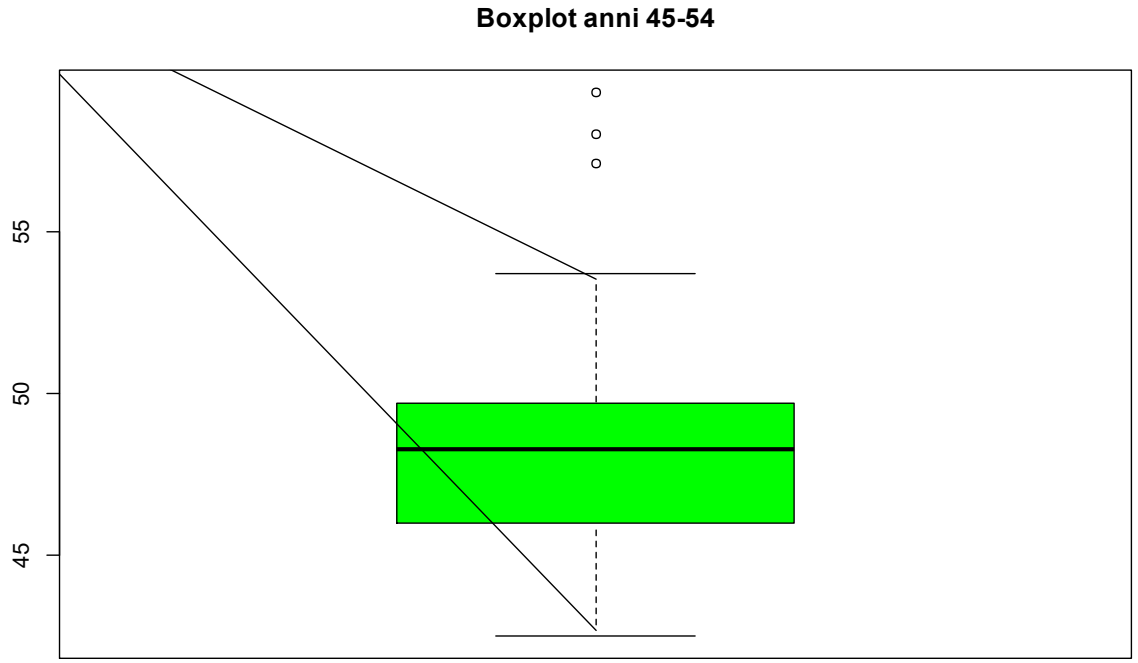
```
> str(b2)
List of 6
 $ stats: num [1:5, 1] 35.4 37.4 38.2 39 41.2
 $ n    : num 40
 $ conf : num [1:2, 1] 37.8 38.5
 $ out  : num [1:4] 42 42.2 44.8 41.7
 $ group: num [1:4] 1 1 1 1
 $ names: chr ""
```

Possiamo affermare che i valori anomali contenuti nel campo *out* sono: 42, 42.2, 44.8, 41.7 .

Poiché Q3 – Q2 = **0.8** è di poco minore a Q2 – Q1 = **0.8** allora possiamo affermare che i dati sono abbastanza simmetrici.

6.3 Fascia d'età 45-54

```
b3<-boxplot(anni.45to54,main="Boxplot anni 45-54", col="green")
```



Il grafico presenta tre anomalie corrispondenti agli atenei: Luigi Bocconi, LUISS Guido Carli e Cattolica del Sacro Cuore che risultano essere maggiori del baffo superiore che è pari a 53.7. Le informazioni ricavate sono:

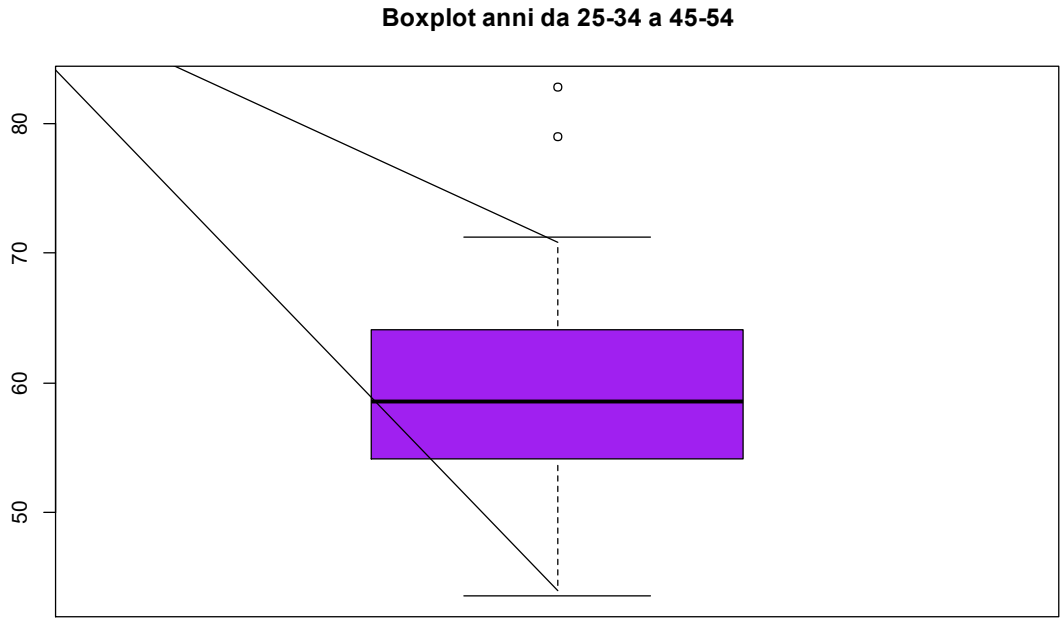
```
> str(b3)
List of 6
 $ stats: num [1:5, 1] 42.5 46 48.2 49.7 53.7
 $ n    : num 40
 $ conf : num [1:2, 1] 47.3 49.2
 $ out  : num [1:3] 58 57.1 59.3
 $ group: num [1:3] 1 1 1
 $ names: chr ""
```

Possiamo affermare che i valori anomali contenuti nel campo *out* sono: 42, 42.2, 44.8, 41.7 .

Poiché Q3 – Q2 = **1.5** è di poco minore a Q2 – Q1 = **2.2** allora possiamo affermare che i dati sono discretamente simmetrici.

6.4 Fascia d'età da 25-34 a 45-54

```
b4<-boxplot(anni.from.25to34.and.45to54,main="Boxplot anni da 25-34 a 45-54",
col="purple")
```



Il grafico presenta tre anomalie corrispondenti agli atenei: LUISS Guido Carli e Cattolica del Sacro Cuore che risultano essere maggiori del baffo superiore che è pari a 71.2 . Le informazioni ricavate sono:

```
> str(b4)
List of 6
 $ stats: num [1:5, 1] 43.6 54.1 58.5 64 71.2
 $ n    : num 40
 $ conf : num [1:2, 1] 56.1 61
 $ out  : num [1:2] 82.8 79
 $ group: num [1:2] 1 1
 $ names: chr ""
```

Possiamo affermare che i valori anomali contenuti nel campo *out* sono: 42, 42.2, 44.8, 41.7 .

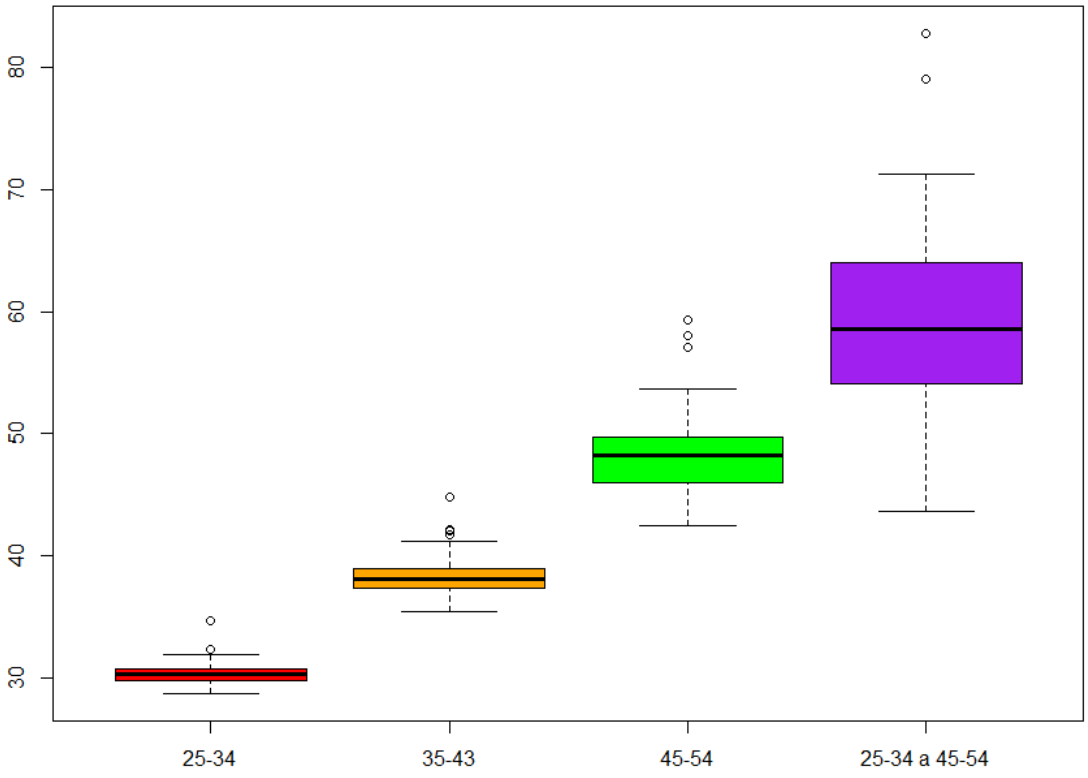
Poiché Q3 – Q2 = **5.5** è di poco minore a Q2 – Q1 = **4,4** allora possiamo affermare che i dati sono discretamente simmetrici.

6.5 Confronto tra boxplot

R permette di confrontare i vari boxplot inserendoli in un unico grafico in modo da avere un quadro completo.
Il codice R è il seguente:

```
boxplot(anni.25to34,anni.35to44,anni.45to54,anni.from.25to34.and.45to54,
        col = c("red","orange","green","purple"),
        names = c("25-34", "35-43", "45-54", "25-34 a 45-54"))
```

Che produce il seguente grafico:



Riguardo a questa analisi siamo venuti a conoscenza che il nostro dataset presenta una discreta asimmetria denotata dalle mediane dei singoli boxplot. Infine siamo venuti a conoscenza dei tutti valori anomali in ogni categoria d'età che risultano essere maggiormente associati agli atenei: Luigi Bocconi, LUISS Guido Carli e Cattolica del Sacro Cuore.

7. Statistica descrittiva

La statistica descrittiva è una statistica di sintesi che descrive quantitativamente e sintetizza le caratteristiche di una collezione di dati.

La statistica descrittiva si distingue dalla statistica inferenziale (o induttiva), in quanto quella descrittiva ha come proprio scopo quello di sintetizzare un campione con pochi numeri o grafici significativi, cioè si occupa di fotografare una data situazione e di sintetizzarne le caratteristiche salienti, mentre quella inferenziale si concentra piuttosto sull'utilizzo dei dati statistici, anche sintetizzati mediante la statistica descrittiva, per fare previsioni probabilistiche. La statistica descrittiva non si sviluppa sulle basi della teoria della probabilità e sono statistiche non parametriche.

Alcune misure che sono comunemente utilizzate per descrivere un insieme di dati sono le misure di centralità e le misure di dispersione o variabilità.

Del primo gruppo fanno parte la media campionaria, la mediana e la moda, mentre misure di variabilità includono la varianza, la deviazione standard, la curtosi campionaria (un indice che permette di misurare la densità dei dati intorno alla media è la curtosi campionaria) e la skewness campionaria (un indice che permette di misurare la simmetria di una distribuzione di frequenze è la skewness campionaria (coefficiente di simmetria)).

La statistica descrittiva fornisce semplici misure di sintesi sul campione e sulle osservazioni che sono state fatte. Queste sintesi possono essere quantitative o visuali e possono essere sia la base di un'iniziale descrizione dei dati come parte di un'analisi più approfondita, ma anche sufficienti da soli per particolari indagini.

Vediamo dunque la statistica descrittiva univariata, in primo luogo, e successivamente la statistica descrittiva bivariata.

7.1 Statistica descrittiva Univariata

La statistica descrittiva **univariata** descrive la distribuzione di una **singola variabile** e include gli indici di posizione **centrali** (media, mediana, moda) e **non centrali** (quantili: quartili, decili, percentali) e gli indici di dispersione (varianza, deviazione standard, coefficiente di variazione) che misurano quanto si disperdono i dati rispetto alla media.

Descriviamo la forma della distribuzione invece attraverso gli indici di **skewness e curtosi**.

Nel corso della trattazione dei vari indici di sintesi, questi saranno accompagnati con opportuni grafici.

7.1.1 Funzione di distribuzione empirica

Quando abbiamo a che fare con fenomeni quantitativi, come nel nostro caso, è utile definire la funzione di distribuzione empirica.

Vediamo quindi la funzione di distribuzione empirica discreta e successivamente quella continua.

7.1.1.1 Funzione di distribuzione empirica discreta:

La funzione viene definita a partire dalle frequenze relative cumulate.

Prendiamo in considerazione una delle nostre variabili quantitative come, ad esempio, gli stipendi della fascia d'età 25-34 per ogni ateneo. Diciamo che i valori distinti che può assumere questa variabile sono $z_1, z_2, \ldots, z_k$ e assumiamo che siano ordinati in ordine crescente. Consideriamo poi le frequenze relative e le frequenze relative cumulate:

$$F_i = f_1 + f_2 + \cdots + f_i = \frac{n_1 + n_2 + \cdots + n_i}{n} \qquad (i = 1, 2, \ldots, k),$$

dove la generica F_i rappresenta la proporzione dei dati del campione minori o uguali di z_i. Una funzione di distribuzione empirica discreta risulta:

$$F(x) = \frac{\#\{x_i \leq x, i = 1, 2, \ldots, n\}}{n} = \begin{cases} 0, & x < z_1 \\ F_1, & z_1 \leq x < z_2 \\ & \ldots \\ F_i, & z_i \leq x < z_{i+1} \\ & \ldots \\ 1, & x \geq z_k \end{cases}$$

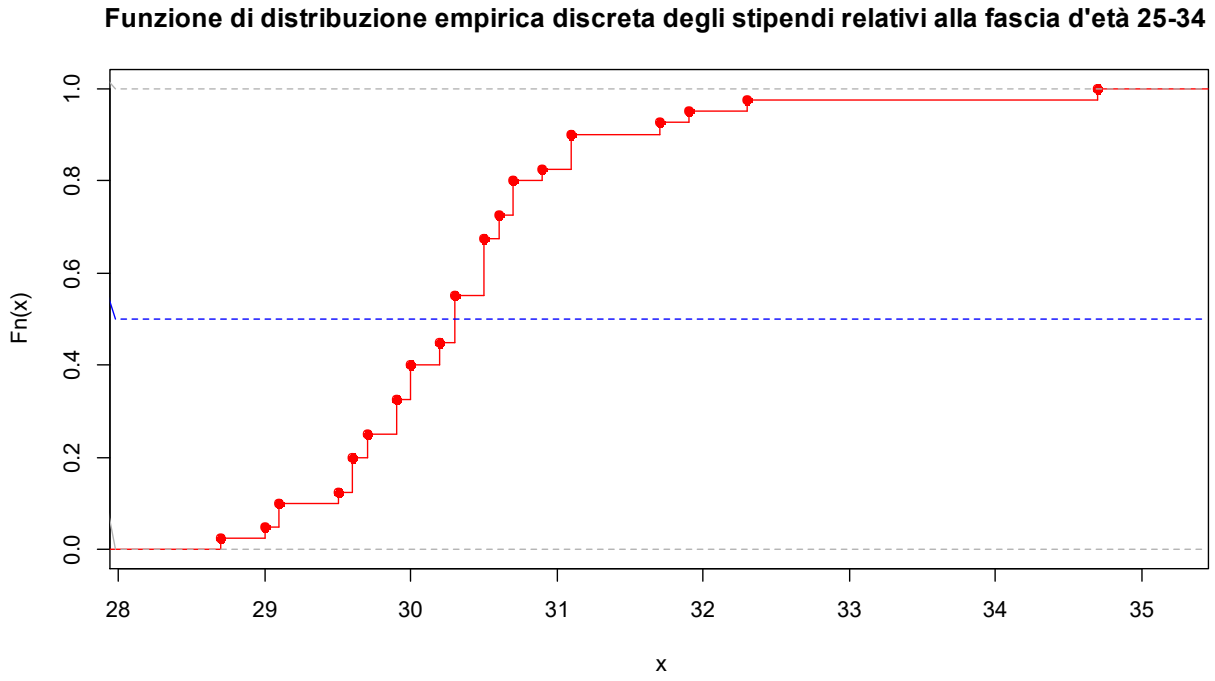
La funzione è definita per ogni x reale e ha diverse caratteristiche:

- È una funzione a scalini non decrescente
- In corrispondenza di ogni salto assume il valore a sinistra sull'asse delle ascisse
- La funzione vale:
    - 0 per ogni valore minore dell'osservazione minima
    - 1 per ogni valore maggiore dell'osservazione massima

R mette a disposizione la funzione `ecdf()` (**e**mpirical **c**umulative **d**istribution **f**unction) che ci permette di disegnare il grafico della funzione di distribuzione empirica per variabili quantitative discrete e determinare per ogni x reale il valore di tale funzione.

Dato che i valori contenuti nei nostri vettori sono quasi tutti diversi, la funzione non ci è molto di aiuto nella nostra indagine.
Vediamo un esempio di applicazione sul vettore relativo agli stipendi dell'età compresa 25-34:

```
plot(
  ecdf(anni.25to34),
  main="Funzione di distribuzione empirica discreta degli stipendi relativi alla
fascia d'età 25-34 ",
  verticals = TRUE , col="red")
abline(h=0.5,col="blue",lty=2)
```



7.1.1.2 Funzione di distribuzione empirica continua:

La funzione di distribuzione empirica continua viene invece utilizzata quando si lavora con dati raccolti in classi. Infatti, considerando la tipologia del nostro campione di dati è più opportuno suddividere le informazioni a disposizione in k distinte classi. Anche in questo caso la funzione viene definita a partire dalle frequenze relative cumulate.

La funzione di distribuzione empirica continua è dunque così definita:

$$F(x) = \begin{cases} 0, & x < z_0 \\ \cdots\cdots & \\ F_{i-1}, & x = z_{i-1} \\ \dfrac{F_i - F_{i-1}}{z_i - z_{i-1}}\, x + \dfrac{z_i\, F_{i-1} - z_{i-1}\, F_i}{z_i - z_{i-1}}, & z_{i-1} < x < z_i \\ F_i, & x = z_i \\ \cdots\cdots & \\ 1, & x \geq z_k, \end{cases}$$

Si nota che F(x) = 0 per $x < z_0$, F(x) = 1 per $x \geq z_k$, mentre se $z_{i-1} < x < z_i$ la funzione di distribuzione empirica continua coincide con il segmento che passa per i punti (z_{i-1}, F_{i-1}) e (z_i, F_i), ossia:

$$\frac{y - F_{i-1}}{x - z_{i-1}} = \frac{F_i - F_{i-1}}{z_i - z_{i-1}} \qquad (i = 1, 2, \ldots, k),$$

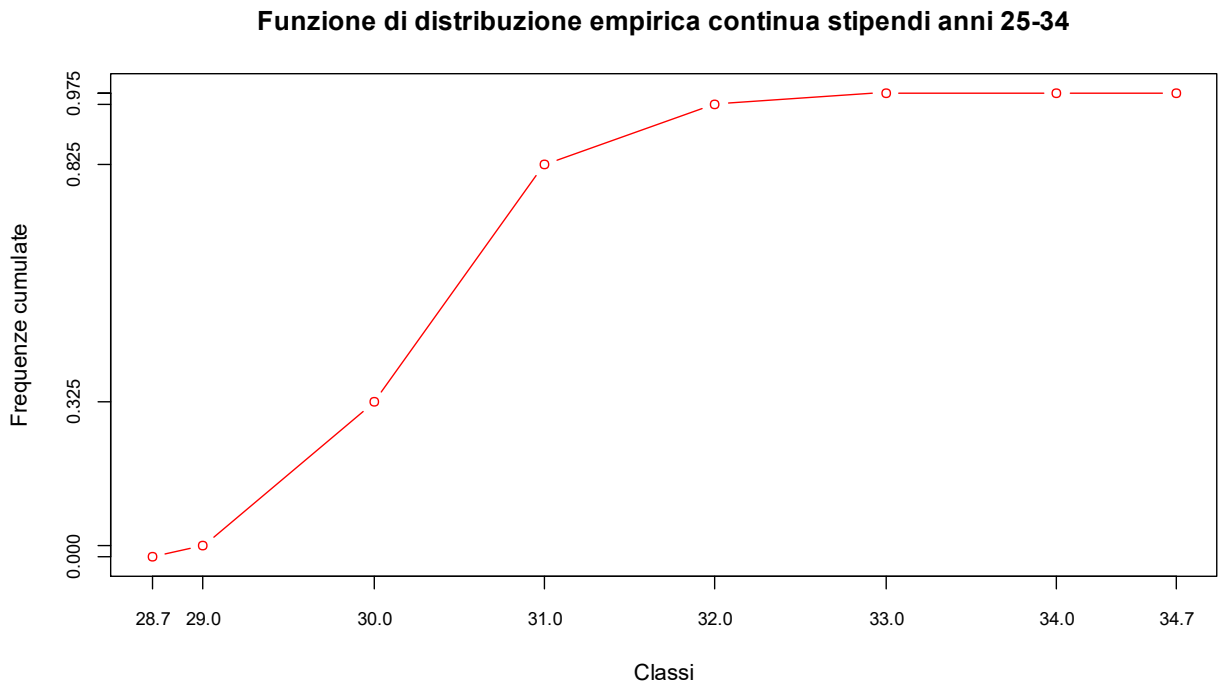
Introduciamo la funzione di distribuzione empirica continua raggruppando per classi sfruttando l'istogramma

Descriviamo ora come ottenere il grafico della funzione di distribuzione empirica continua per quanto riguarda le fasce d'età, creando una funzione apposita:

```
Distribuzione.Empirica.Continua<-function(vettore,main){

  D.E.Continua <-hist(vettore,freq=TRUE)
  classi.D.E.C<-c(min(vettore),D.E.Continua$breaks[3:length(D.E.Continua$breaks)-
1],max(vettore))
  FrequenzaCumulate <- cumsum(table (cut(vettore,
                                         breaks =classi.D.E.C,
                                         right =FALSE ))))/length (vettore)

  FrequenzaCumulate<-c(0,FrequenzaCumulate)
  plot(classi.D.E.C,
       FrequenzaCumulate, type = "b", axes = FALSE ,
       main = paste("Funzione di distribuzione empirica continua ",main,sep=""),
       col="red", xlab = "Classi", ylab = "Frequenze cumulate")
  axis(1, classi.D.E.C, cex.axis=0.80)
  axis(2, format (FrequenzaCumulate, digits = 2), cex.axis=0.80)
  box()

}
```

Fascia d'età 25-34:

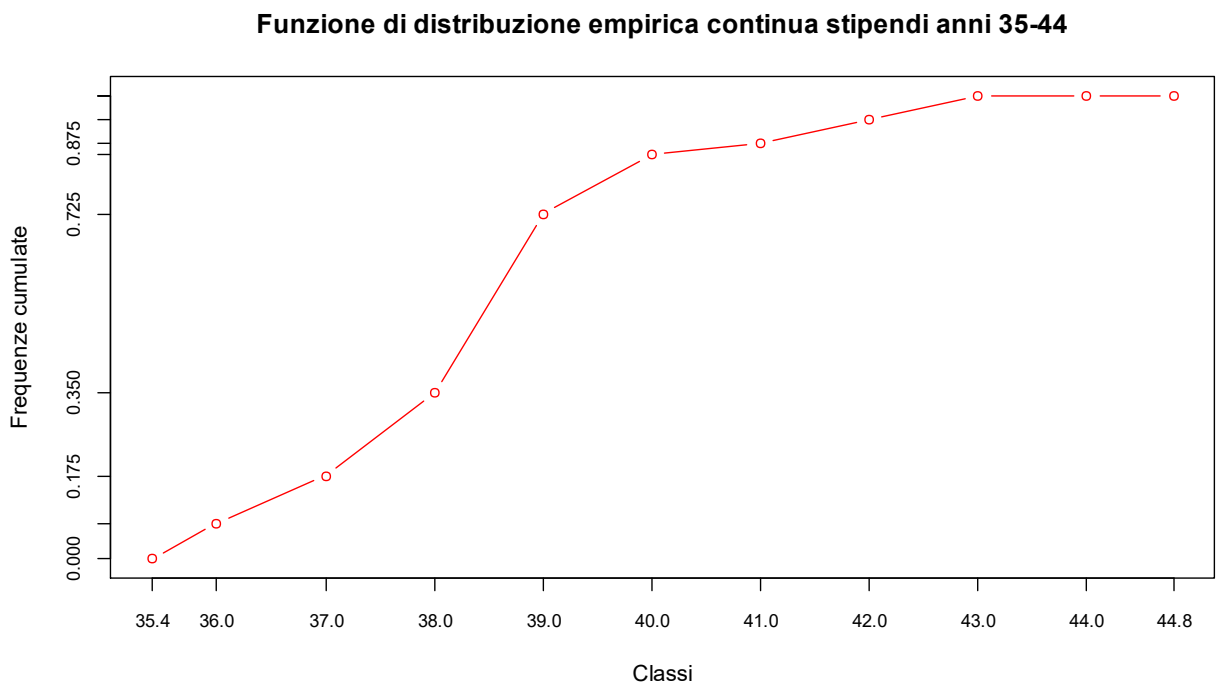
```
Distribuzione.Empirica.Continua(anni.25to34, "stipendi anni 25-34")
```



È possibile notare dal grafico come ci sia una forte concentrazione dei dati nelle prime tre classi, in particolare si ha una concentrazione dell'**82,5 %**

Fascia d'età 35-44:

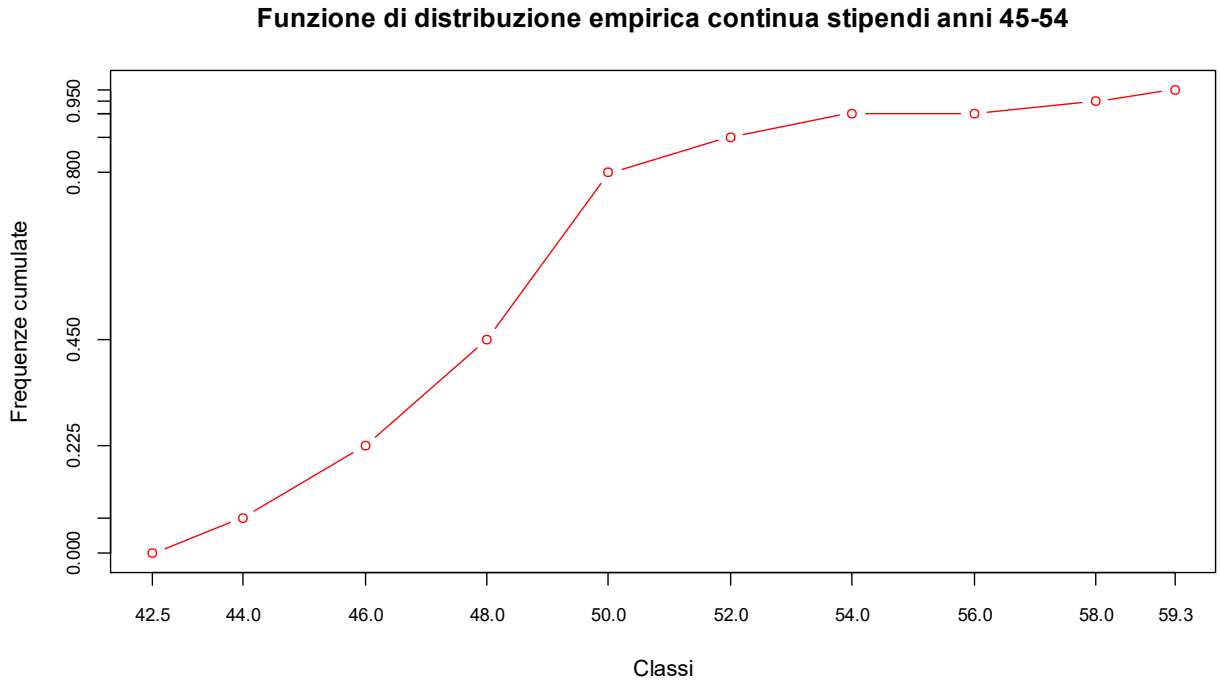
```
Distribuzione.Empirica.Continua(anni.35to44, "stipendi anni 35-44")
```



I dati sono concentrati principalmente nelle classi centrali.

Fascia d'età 45-54:

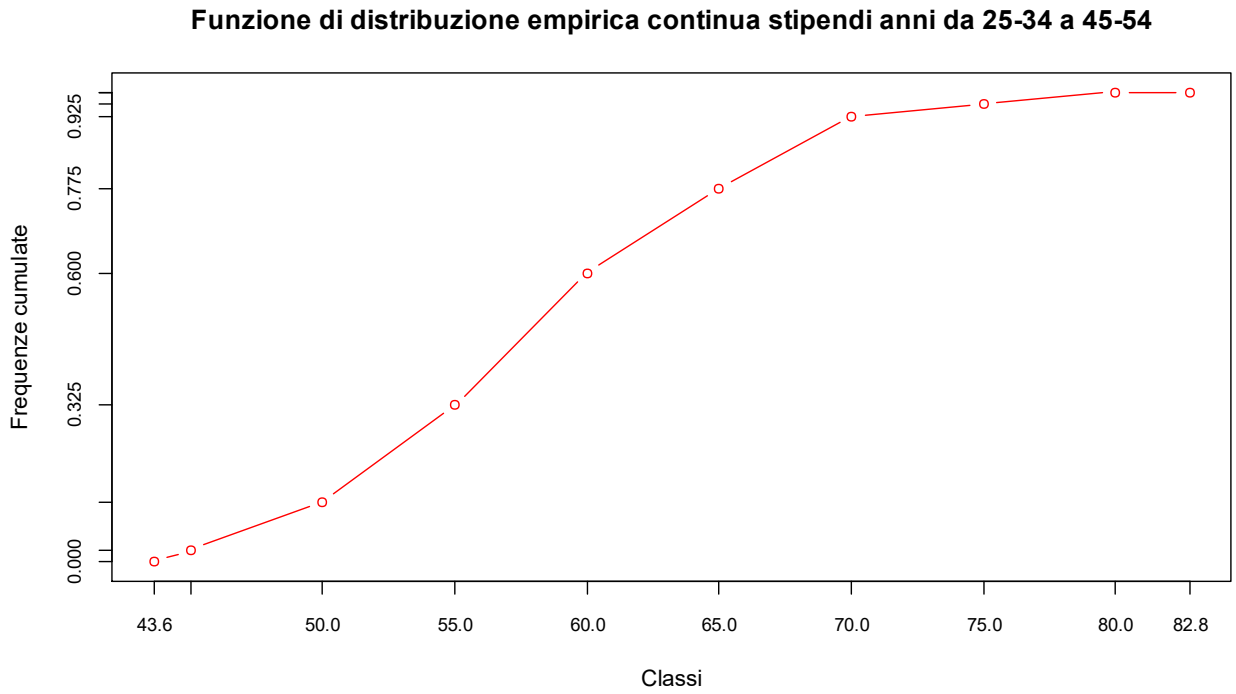
```
Distribuzione.Empirica.Continua(anni.45to54, "stipendi anni 45-54")
```



I dati sono concentrati in maniera quasi uniforme.

Fascia d'età 25-34 a 45-54:

```
Distribuzione.Empirica.Continua(anni.from.25to34.and.45to54, "stipendi anni da 25-34
a 45-54")
```



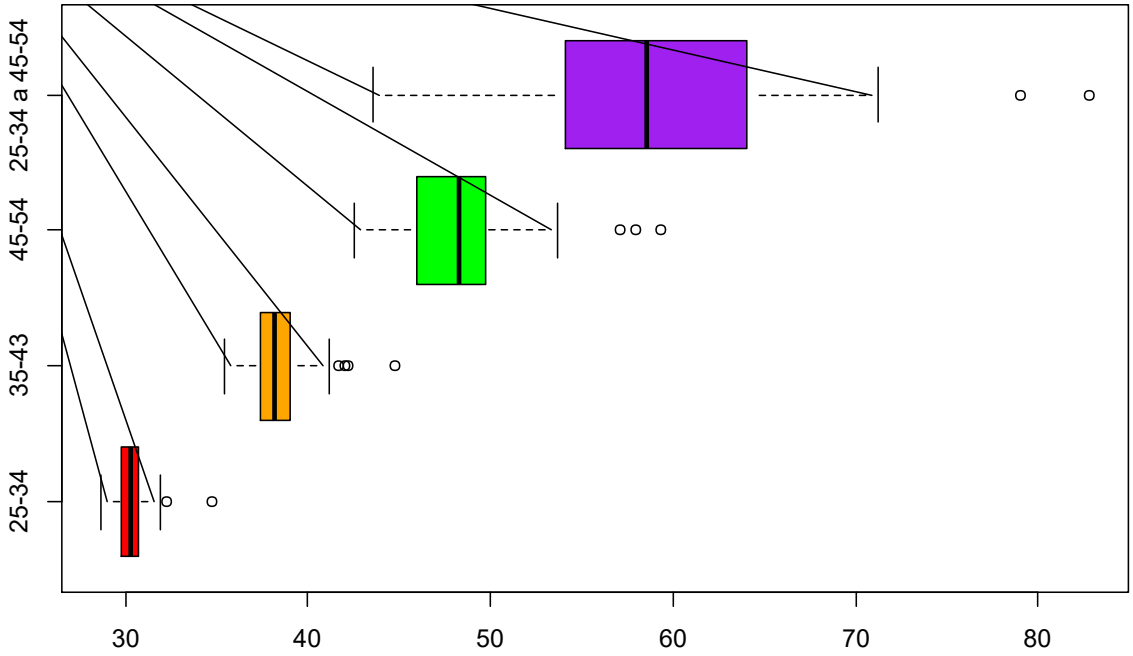
I dati sono concentrati in maniera omogenea.

7.1.2 Indici di sintesi

Gli indici di sintesi sono necessari per sintetizzare e classificare specifiche osservazioni sui dati che abbiamo. Per avere una visione d'insieme dei nostri dati e ricavare informazioni quanto più complete possibili è preferibile che gli indici di sintesi vadano osservati assieme.

Vediamo dunque attraverso i boxplot dei nostri vettori quali sono le caratteristiche delle loro distribuzioni di frequenza: possiamo comprenderne la forma e se i dati risultano molto variabili o meno.

Nel seguente grafico riportiamo tutti i boxplot osservati:



In tutti i grafici notiamo una certa simmetria presente nella totalità dei boxplot.

Gli indici che introdurremo nel seguito servono a misurare quantitativamente alcune delle caratteristiche che si possono intuire nei grafici delle distribuzioni di frequenza e nei boxplot.

7.1.2.1 Media campionaria

Supponiamo di avere un insieme $x_1, x_2, \ldots, x_n$ di n valori numerici (dati statistici quantitativi), detto campione di ampiezza o numerosità pari a n. La media campionaria è la media aritmetica di questi valori.

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

La media campionaria gode delle seguenti proprietà:

- **Proprietà di linearità:**
$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n}(ax_i + b) = a\bar{x} + b$$
- È una **media pesata dei valori distinti assunti** dai dati dove ogni valore distinto usa come peso la **frequenza**:
$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{k} z_i n_i = \sum_{i=1}^{k} \frac{n_i}{n} z_i = \sum_{i=1}^{k} f_i z_i$$
- La somma algebrica degli scarti della media campionaria è sempre **nulla**, dove s_i che indica il grado di scostamento dal singolo valore x_i:
$$\sum_{i=1}^{n} s_i = \sum_{i=1}^{n}(x_i - \bar{x}) = \sum_{i=1}^{n} x_i - n\bar{x} = n(\bar{x} - \bar{x}) = 0$$
- La media è **influenzata da tutti i dati** e in particolar modo da valori particolarmente grandi o piccoli, cioè da valori **anomali**

Il linguaggio R mette a disposizione la funzione `mean()` per calcolare la media campionaria dei nostri dati

7.1.2.2 Mediana campionaria

Un altro indice di posizione è la mediana.Data una distribuzione di un carattere quantitativo oppure qualitativo ordinabile si definisce mediana (o valore mediano) il valore che separa la metà più grande e la metà più piccola di un campione di dati, cioè il valore che si trova nel mezzo della distribuzione.
I passi per calcolare la mediana sono i seguenti:

- Si ordinano gli n elementi in **ordine crescente**
- Se il numero di dati è **dispari** la mediana corrisponde al valore $\frac{n+1}{2}$
- Se il numero di dati è pari allora la mediana è calcolato come la media aritmetica dei valori alla posizione $\frac{n}{2}$ e alla posizione $\frac{n}{2} + 1$

È importante notare come la mediana campionaria dipenda solo da uno o due valori e quindi non risente degli estremi.
In R per calcolare la mediana abbiamo a disposizione la funzione `median()`.

7.1.2.3 Tabella riassuntiva

Una volta calcolata la media e la mediana di tutte le fasce d'età mediante R, ci risultano i seguenti dati:

	25-34	35-44	45-54	Da 25-34 a 45-54
Media	30.3875	38.43	48.385	59.07
Mediana	30.3	38.15	48.25	58.55

Dalla tabella notiamo che il calcolo della media e della mediana ha prodotto risultati simili per ciascuna fascia d'età.

Confrontare media e mediana è utile poiché se queste misure sono simili, come nel nostro caso, la distribuzione di frequenza tende ad essere simmetrica.

7.1.2.4 Moda campionaria

La moda identifica la modalità che si presenta con la maggiore frequenza (assoluta o relativa) in un campione. Se ci sono più modalità con frequenza massima, ciascuna viene definita come valore modale. Sulla moda c'è da dire che questa, a differenza di media e mediana, può essere calcolata anche quando si ha a che fare con dati di tipo qualitativo. Nel **nostro caso è inutile calcolarla** in quanto non può non essere utile quando i dati sono numerosi e per la maggior parte diversi tra loro. Inoltre, se tutte le modalità presentano all'incirca la stessa frequenza, la moda non è un indice di sintesi significativo. Infine, sottolineiamo che media, mediana e moda sono detti indici di posizione centrale poiché descrivono attorno a quali valori è centrato l'insieme di dati.

7.1.2.5 Quantili

Sono definiti come indici di sintesi di posizione non centrali. I quantili dividono l'insieme dei dati in un fissato numero di parti uguali. In R esistono 9 differenti algoritmi, ma R di default utilizza l'algoritmo 7.
Vediamo dunque il calcolo dei quantili:

```
> quantile(anni.25to34)
   0%   25%   50%   75%  100% 
28.70 29.85 30.30 30.70 34.70 
> quantile(anni.35to44)
   0%   25%   50%   75%  100% 
35.40 37.40 38.15 39.00 44.80 
> quantile(anni.45to54)
   0%   25%   50%   75%  100% 
42.50 46.00 48.25 49.70 59.30 
> quantile(anni.from.25to34.and.45to54)
    0%    25%    50%    75%   100% 
43.600 54.150 58.550 64.025 82.800 
```

7.1.2.6 Varianza e deviazione standard campionaria

Gli indici di posizione sono importanti, ma le informazioni che ci danno non tengono conto della variabilità dei dati: può capitare di avere a che fare con distribuzioni di frequenza molto diverse tra di loro ma che presentano la stessa media campionaria ad esempio.

È dunque importante arricchire l'indagine su un dato campione di dati con degli indici per capaci di misurare la variabilità dei dati.

Introduciamo dunque la varianza campionaria e la deviazione standard campionaria.

La varianza campionaria è indicata con s^2 ed è definita:

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

La deviazione standard è invece indicata con s ed è definita come la radice quadrata della varianza campionaria:

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

Varianza e deviazione standard sono detti indici di dispersione in quanto misurano la dispersione dei dati intorno alla media.
La deviazione standard ha la stessa unità di misura dei valori osservati, mentre la varianza ha come unità di misura il quadrato dell'unità di misura dei valori di riferimento.
La deviazione standard viene detta anche scarto quadratico medio per come è definita.
Le proprietà della varianza sono le seguenti:

- Non gode della proprietà di linearità
- Sommare una costante a ciascuno dei dati non fa cambiare la varianza
- Moltiplicare i dati del campione per un fattore costante fa sì che la varianza campionaria risulti moltiplicata per il quadrato di tale fattore

Il valore della varianza è 0 quando tutti i valori sono uguali tra loro e risulta essere più grande tanto più le osservazioni si discostano dalla media.
In R per calcolare la varianza campionaria usiamo il comando `var()`, mentre per calcolare la deviazione standard usiamo la funzione `sd()`.
Valutiamo quindi quanto si disperdono i nostri dati rispetto alla media.
Calcoliamo prima la varianza per ogni fascia d'età:

```
> var(anni.25to34)
[1] 1.06266
> var(anni.35to44)
[1] 3.658564
> var(anni.45to54)
[1] 14.16387
> var(anni.from.25to34.and.45to54)
[1] 71.8601
>
> sd(anni.25to34)
[1] 1.030854
> sd(anni.35to44)
[1] 1.912737
> sd(anni.45to54)
[1] 3.763492
> sd(anni.from.25to34.and.45to54)
[1] 8.477034
```

Questi indici di dispersione sono relativi ai singoli dati presi in considerazione, se vogliamo però confrontare le variazioni tra diversi insiemi di dati si utilizza il **coefficiente di variazione**.

7.1.2.7 Coefficiente di variazione

Il coefficiente di variazione è definito come il rapporto tra la **deviazione standard campionaria e il modulo della media campionaria**:

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

Il coefficiente di variazione è un numero puro, è un indice adimensionale, cioè non dipende dall'unità di misura utilizzata (la media campionaria e la deviazione standard campionaria sono espressi in identiche unità di misura).

È facile capire che il coefficiente di variazione ha senso che sia calcolato solo con campioni con media campionaria non nulla. Non dobbiamo guardare alla varianza in senso assoluto, ma dobbiamo confrontarla con la media per fare una buona valutazione.
Essa è importante nel caso in cui si debba guardare al coefficiente di variazione perché ci interessa confrontare insiemi che hanno differenti range di variazione, cioè insiemi in cui la differenza tra il massimo e il minimo è molto diversa.

Il coefficiente risulta essere sempre maggiore di 0, ma se compreso tra 0 e 1 vuol dire che la media è più grande della deviazione standard. Invece più è grande più c'è dispersione nei dati e quindi il valore medio non è molto significativo.

Dato che in R non c'è una funzione per calcolare il coefficiente di variazione, introduciamo la seguente funzione:

```
cv<-function (x){
  sd(x)/abs(mean(x))
}
```

Utilizziamo quindi la funzione per ricavare il coefficiente di variazione:

```
> cv(anni.25to34)
[1] 0.03392362
> cv(anni.35to44)
[1] 0.04977198
> cv(anni.45to54)
[1] 0.0777822
> cv(anni.from.25to34.and.45to54)
[1] 0.1435083
```

Da come si può notare non c'è una grande variazione dei dati. Si nota che la variazione più grande tra i dati risulta appartenere alla categoria della fascia d'età da 25-34 anni a 45-54 anni.

7.1.2.8 Forma di distribuzione di frequenza

Gli indici trattati fino ad ora ci permettono già di intuire quale sia la forma della distribuzione di frequenza. Grazie ai dati sulla media e la mediana possiamo dire se c'è uno sbilanciamento verso destra o sinistra, vedendo se i valori differiscono e come differiscono, mentre con la moda possiamo dire se c'è un picco nella funzione, o più picchi.

Introduciamo formalmente ora gli indici che permettono di misurare la simmetria della funzione di distribuzione e la piccatezza: **skewness** e **curtosi campionaria**.

Skewness

Dato un insieme di dati numerici, si definisce skewness:

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

dove m_3 è il momento centrato campionario di ordine 3.

In base al valore della skewness abbiamo che:

- Se γ_1**=0**, la distribuzione di frequenza è simmetrica
- Se γ_1**>0**, la distribuzione di frequenza ha la coda di destra più allungata per l'asimmetria positiva
- Se γ_1**<0**, la distribuzione di frequenza ha la coda di sinistra più allungata per l'asimmetria negativa

Definiamo in R la funzione per il calcolo dell'indice:

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

skw(anni.25to34)
skw(anni.35to44)
skw(anni.45to54)
skw(anni.from.25to34.and.45to54)
```

E ne calcoliamo i valori:

```
> skw(anni.25to34)
[1] 1.876476
> skw(anni.35to44)
[1] 1.149409
> skw(anni.45to54)
[1] 1.108719
> skw(anni.from.25to34.and.45to54)
[1] 0.6053428
```

Notiamo che l'indice non è mai pari a 0, quindi non c'è mai simmetria.
Se guardiamo più attentamente i risultati ottenuti notiamo che i dati presentano una distribuzione di frequenza che ha la **coda di destra più allungata per l'asimmetria positiva.**

Curtosi

L'indice di curtosi invece permette di misurare la **densità dei dati intorno alla media** ed è definita:

$$\gamma_2 = \beta_2 - 3$$

dove con $\beta_2 = m_4/m_2^2$ si indica l'**indice di Pearson**. In base al valore ottenuto abbiamo che:

- Se **$\beta_2 < 3$ ($\gamma_2 < 0$)**: la distribuzione di frequenza si definisce **platicurtica**, ossia la distribuzione di frequenza è più piatta di una normale.
- Se **$\beta_2 > 3$ ($\gamma_2 > 0$)**: la distribuzione di frequenza si definisce **leptocurtica**, ossia la distribuzione di frequenza è più piccata di una normale.
- Se **$\beta_2 = 3$ ($\gamma_2 = 0$)**: la distribuzione di frequenza si definisce **normocurtica**, ossia segue la curva di una normale.

Definiamo una funzione per calcolare l'indice:

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

curt(anni.25to34)
curt(anni.35to44)
curt(anni.45to54)
curt(anni.from.25to34.and.45to54)
```

E ne calcoliamo i valori:

```
> curt(anni.25to34)
[1] 6.038266
> curt(anni.35to44)
[1] 1.873167
> curt(anni.45to54)
[1] 1.416298
> curt(anni.from.25to34.and.45to54)
[1] 0.5300441
```

Notiamo che la curtosi ha una distribuzione di frequenza leptocurtica, ossia la distribuzione di frequenza è **più piccata** di una normale.

7.2 Grafico di dispersione (Scatterplot)

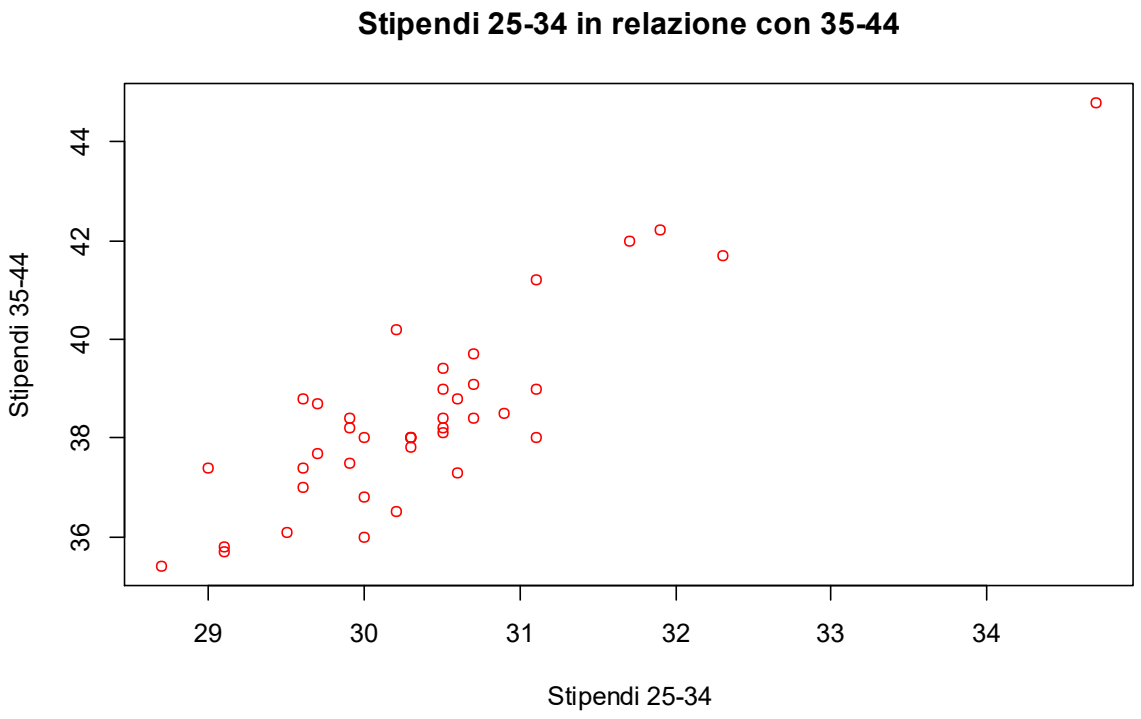
Un grafico di dispersione, anche chiamato Scatterplot, è un tipo di diagramma che utilizza coordinate cartesiane per mostrare tipicamente due variabili di un dato set. I dati vengono definiti come una collezione di punti, ognuno dei quali prende valore rispetto all'asse orizzontale o quello verticale in base all'insieme di appartenenza. La variabile posta sull'asse delle ascisse viene definita variabile indipendente, mentre quella posta sull'asse delle ordinate viene chiamata variabile dipendente.

Il risultato grafico di un diagramma di dispersione è quello di avere una nuvola di punti: questo tipo di rappresentazione serve a capire se esiste una relazione tra le variabili e di che tipo si tratta.

In R uno scatterplot di due vettori viene definito tramite il comando `plot(vettore1, vettore2)`.

Vediamo dunque un esempio di grafico di dispersione prendendo due categorie del nostro dataset, ad esempio confrontiamo l'insieme contenente gli stipendi dell'età comprese tra 25-34 anni in relazione con gli stipendi compresi tra 35-44:

```
plot(anni.25to34,anni.35to44,
     main = "Stipendi 25-34 in relazione con 35-44",
     xlab = "Stipendi 25-34", ylab = "Stipendi 35-44", col ="red")
```

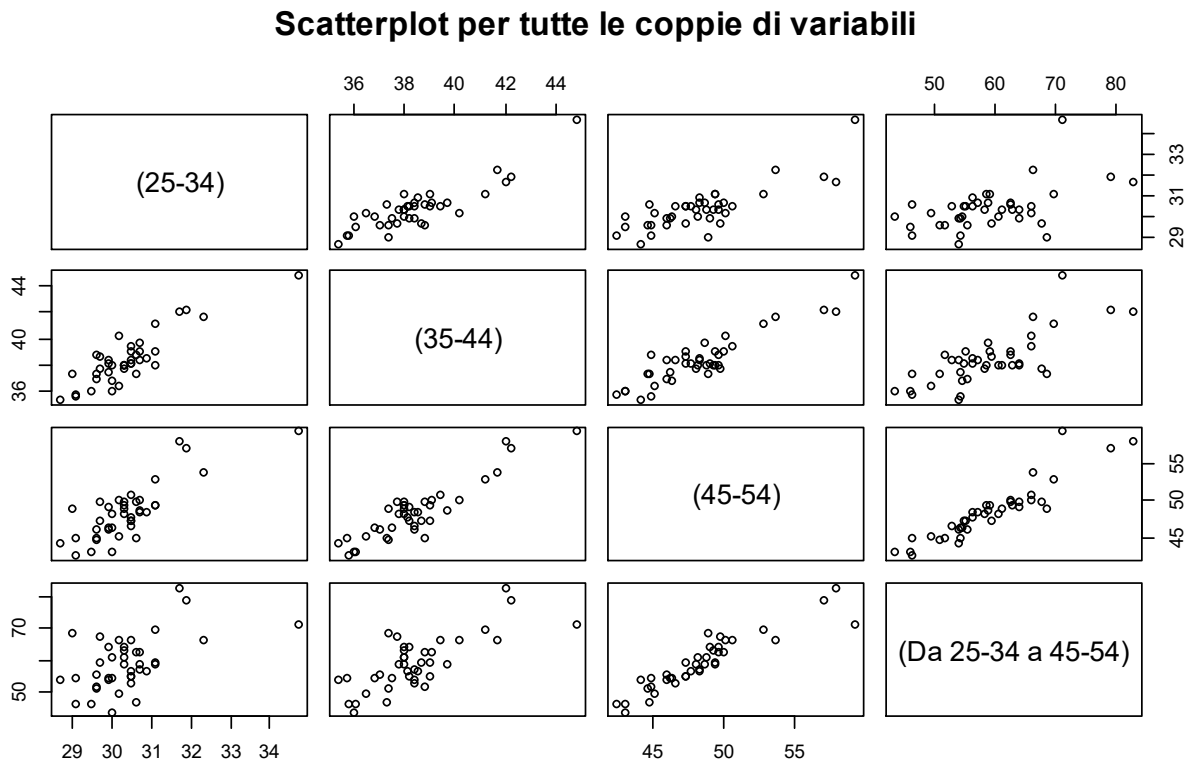


Un'altra possibilità che ci dà R è quella di visualizzare un grafico contenente uno scatterplot per ogni coppia di variabili del nostro dataset. Per farlo ci basta semplicemente utilizzare la funzione

`pairs()` e passare la matrice: in un unico grafico abbiamo tutti gli scatterplot mettendo in relazione tutte le possibili coppie.

Vediamo dunque questo tipo di grafico:

```
pairs(mtxStipendiLordi,
      main = "Scatterplot per tutte le coppie di variabili")
```



I vari grafici ottenuti mostrano le nuvole di punti che si ottengono prendendo in considerazione tutte le differenti coppie di variabili.

7.3 Statistica descrittiva Bivariata

Parliamo di statistica bivariata quando si va a confrontare e a studiare contemporaneamente due variabili di una determinata popolazione atte a descrivere le relazioni che intercorrono tra le due variabili.

Quello che ci interessa fare è dunque verificare e classificare le relazioni esistenti tra variabili quantitative relative alla stessa unità statistica.

Prima di tutto è opportuno definire uno scatterplot in cui ogni coppia di osservazioni è rappresentata nel piano euclideo, andando così a formare una nuvola di punti. Sull'ascisse viene posta quella che è definita variabile indipendente, mentre sull' ordinata viene posta quella che viene definita variabile dipendente, e si disegnano dei punti in corrispondenza delle coppie (x_i,y_i).

Lo scopo di questo grafico è quello di far risaltare eventuali pattern che legano le coppie di variabili, sia a riguardo della forma che alle relazioni tra le categorie analizzate.

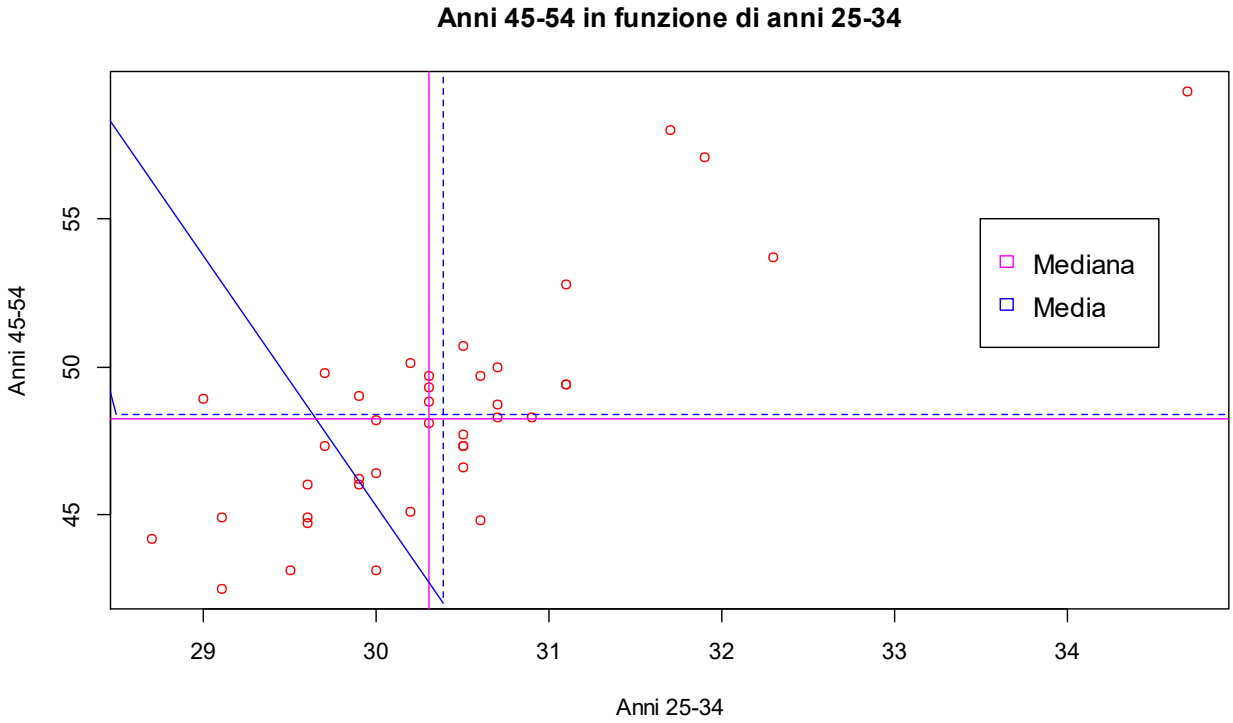
L'analisi verrà fatta prendendo in considerazione il dataset iniziale senza l'ultima colonna inerente alla fascia d'età da 25-34 a 45-54. Dunque, le due categorie che analizziamo del nostro dataset sono relative alle due classi d'età: **anni 25-34** e **anni 45-54**.

Di questi due vettori abbiamo già conoscenza degli indici di sintesi, che riportiamo di seguito:

	25-34	45-54
Media	30.3875	48.385
Mediana	30.3	48.25
Deviazione Standard	1.030854	3.763492

Lo scatterplot relativo ai dati considerati indicando **mediana e media campionaria** è il seguente:

```
var1<-anni.25to34
var2<-anni.45to54

plot(var1,var2,
     main="Anni 45-54 in funzione di anni 25-34",
     xlab = "Anni 25-34",ylab = "Anni 45-54", col="red")
abline(v=median(var1),lty=1, col="magenta")
abline(v=mean(var1),lty=2, col="blue")
abline(h=median(var2),lty=1, col="magenta")
abline(h=mean(var2),lty=2, col="blue")

legend(33.5,55,c("Mediana","Media"),pch=0, col = c("magenta","blue"),cex=1.2)
```



Si nota che i dati sembrano posizionati intorno ad una retta ascendente e ciò induce a pensare che esista una correlazione lineare positiva tra le variabili.

7.3.1 Covarianza campionaria

Continuiamo l'indagine sulla dipendenza delle nostre variabili e lo facciamo introducendo una nuova misura quantitativa che descrive la correlazione, cioè la **covarianza campionaria** definita come di seguito:

$$C_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \qquad (n = 2,3,\dots)$$

Da questa definizione notiamo che il prodotto interno alla sommatoria sarà **positivo** per le osservazioni (x_i,y_i) **in cui le componenti della coppia sono o entrambe maggiori della media campionaria della variabile di cui fanno parte o entrambe minori**; il prodotto invece sarà **negativo** negli altri casi, cioè quando **una parte della coppia risulta essere maggiore e l'altra minore**.

Un'altra cosa da notare è che nella definizione la sommatoria viene divisa per n−1 e questo viene fatto per normalizzarla in quanto nel caso in cui le variabili x e y siano uguali **si ottiene la varianza campionaria**.

La covarianza può assumere i seguenti valori:

- Se $C_{xy}>0$ le variabili sono **correlate positivamente,**
- Se $C_{xy}<0$ le variabili sono **correlate negativamente,**

- Se C_{xy}=0 le variabili considerate **non risultano essere correlate tra loro**, ciò vuol dire che i punti all'interno dello scatterplot sono tutti **sparsi.**

Quando la covarianza campionaria assume valori positivi quello che ci si aspetta è che i cambiamenti della prima variabile siano corrispondenti anche nella seconda (se una cresce anche l'altra lo fa, se una decresce anche l'altra lo fa); non c'è concordanza invece in una covarianza negativa. Una covarianza nulla invece indica che i dati non sono in relazione diretta tra loro.

Tra le proprietà della covarianza campionaria quindi abbiamo:

- $\mathrm{Cov}(x,y) = \mathrm{Cov}(y,x)$
- $\mathrm{Cov}(x,x) = \mathrm{var}(x)$

Calcoliamo la covarianza tra le nostre variabili con il comando `cov()` fornitoci da R:

```
> cov(anni.25to34,anni.45to54)
[1] 3.146731
```

La covarianza risulta positiva, questo indica che le variabili valutate **sono correlate positivamente**.

7.3.2 Coefficiente di correlazione campionario

Il motivo per il quale introduciamo il **coefficiente di correlazione campionaria** è dovuto al fatto che ci fa perdere l'unità di misura, esso non è altro che un indice quantitativo della correlazione tra le variabili è il coefficiente di correlazione campionario che misura quanto è forte il legame di natura lineare tra le variabili considerate.
Il coefficiente ci indica se e come i punti sono posizionati attorno ad una retta interpolante, o se c'è una retta che allinea tutti i punti, e dunque non è possibile con questo coefficiente individuare relazioni curvilinee.

È definito nel seguente modo:

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

Il coefficiente ha le seguenti caratteristiche:

- Prende valore tra **-1 e 1**
- È un **numero puro** senza dimensione
- Può essere calcolato solo se **entrambe le variabili sono quantitative**
- È fortemente influenzato da **valori anomali**

Il coefficiente di correlazione ha lo stesso segno della covarianza, e come precedentemente il segno ci dice se le variabili sono correlate positivamente, negativamente o non correlate.

Sul coefficiente di correlazione c'è da dire che:

1. se esistono due numeri reali **a e b**, con **a>0**, tali che $y_i=ax_i+b$ per ogni i=1,2,..,n, allora $r_{xy}=1$
2. se esistono due numeri reali **a e b**, con **a<0**, tali che $y_i=ax_i+b$ per ogni i=1,2,..,n, allora $r_{xy}=-1$
3. se esistono quattro numeri reali **a, b, c e d**, tali che $z_i=ax_i+b$ e $w_i=cy_i+d$ per i=1,2,..,n, allora $r_{ZW}=r_{xy}$ se ac>0 e $r_{zw}=-r_{xy}$ se invece ac<0.

Il punto 1 e il 2 dimostrano che i valori limite -1 e +1 sono effettivamente raggiunti solo quando tra **X e Y sussiste una relazione lineare**, ossia quando i punti dello scatterplot **giacciono tutti su di una retta**.

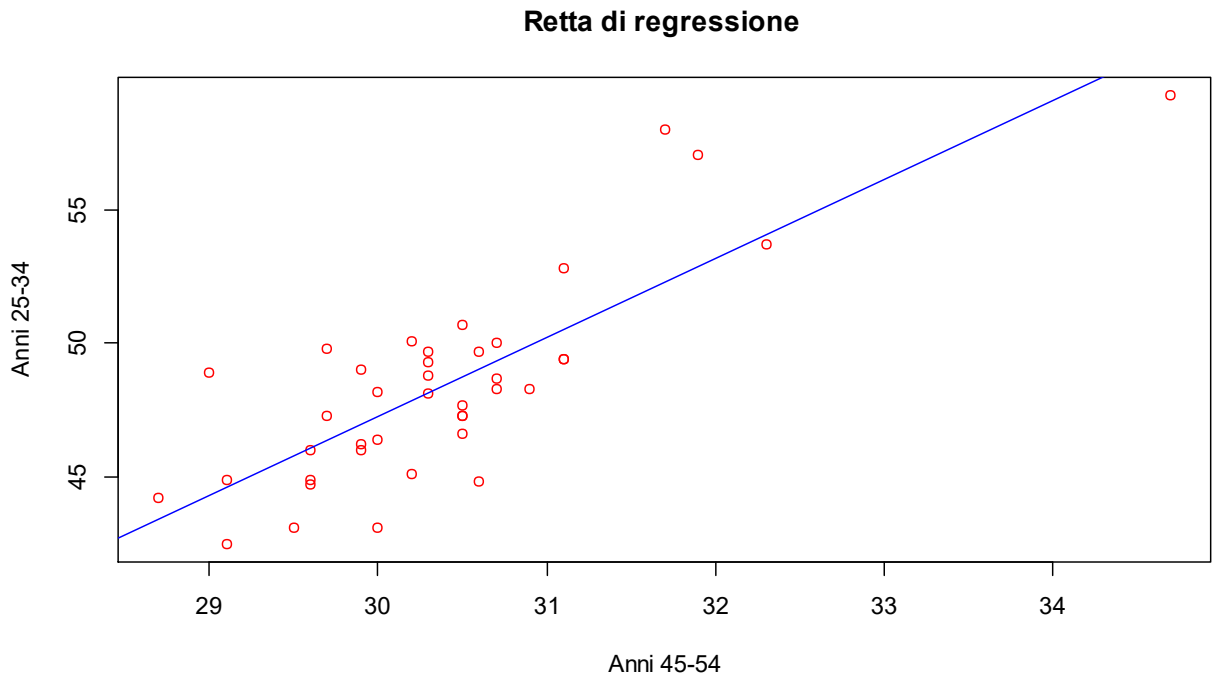
Il punto 3 invece ci dice che il quadrato del coefficiente non cambia se sommiamo o moltiplichiamo costanti a tutti i valori di x e/o y.

In R possiamo calcolare il coefficiente di correlazione con la funzione `Cor()`, vediamolo:

```
> cor(anni.25to34,anni.45to54)
[1] 0.8110943
```

Come è possibile notare il coefficiente è positivo, questo indica una decisa correlazione lineare con una retta ascendente tra i vettori presi in considerazione. Infatti, trattiamo più nel dettaglio il coefficiente di correlazione, attraverso il seguente codice in R:

```
cor(anni.25to34,anni.45to54)

plot(anni.25to34 ,anni.45to54 ,main="Retta di regressione ",
     xlab="Anni 45-54",ylab="Anni 25-34", col ="red")
abline(lm(anni.45to54~anni.25to34), col="blue")
```



7.3.3 Regressione lineare

In statistica si parla di regressione lineare indicando un approccio che modella le relazioni tra una variabile detta dipendente e una o più variabili dette indipendenti.

Il caso in cui la variabile indipendente sia una sola è detto **regressione lineare semplice**, mentre quando ci sono più variabili indipendenti abbiamo a che fare con la **regressione lineare multipla**.

Nella regressione lineare le relazioni sono modellate usando funzioni di predizione lineare i cui parametri del modello sono stimati dai dati. Questi modelli sono detti **modelli lineari**.

7.3.3.1 Regressione lineare semplice

Il modello di regressione lineare semplice è esprimibile attraverso l'equazione di una retta che riesce meglio di qualunque altra ad interpolare la nuvola di punti.

Data dunque l'equazione

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

Dove

1. **α** è l'intercetta, cioè indica l'ordinata con la quale la retta di regressione si interseca con l'asse delle ordinate
2. **β** è il coefficiente angolare, cioè indica la "pendenza" della retta.
   i. Se positivo allora la retta di regressione è crescente
   ii. Se negativo allora la retta di regressione è discendente
   iii. Se nullo allora la retta di regressione è orizzontale

Questa retta viene ottenuta con il metodo dei minimi quadrati.
Per calcolare i coefficienti di regressione è necessario considerare la **somma Q dei quadrati degli errori**

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

e minimizzarla. Quindi la variabile dipendente y viene "spiegata" attraverso una **relazione lineare della variabile indipendente x** (cioè: α + βx).
Il problema della regressione si traduce nella determinazione di α e β in modo da esprimere al meglio la relazione funzionale tra y e x.
Derivando Q rispetto α e β e ponendo le derivate parziali ottenute a 0, il metodo dei minimi quadrati conduce a:

$$\beta = \frac{S_y}{S_x} r_{xy}, \qquad a = \overline{y} - \beta \overline{x}$$

Notiamo come il coefficiente di correlazione influenza fortemente β e se quest'ultimo è 0, allora la retta è **orizzontale**.

Possiamo quindi calcolare α e β con le seguenti linee di codice:

```
x1<-anni.25to34
x2<-anni.45to54
beta <-(sd(x2 )/sd(x1 ))*cor(x1 ,x2)
alpha <-mean(x2)-beta*mean(x1)
c(alpha ,beta)
```

```
> c(alpha ,beta)
[1] -41.597928   2.961182
```

Con **α** riusciamo a stimare dove la retta di regressione intercetta l'asse delle y. Invece, **β** risulta essere positiva e ciò indica che la retta di regressione è crescente.

È possibile ottenere i medesimi risultati attraverso la funzione R `lm(y~x)` in cui gli argomenti indicano che **y dipende da x**.
Di seguito il codice:

```
> lm(x2~x1)

Call:
lm(formula = x2 ~ x1)

Coefficients:
(Intercept)           x1
    -41.598        2.961
```

In R la funzione `lm(y~x)` contiene altre informazioni ottenibili attraverso la funzione `attributes()`, ed essi risultano essere i seguenti:

```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"
```

In conclusione, possiamo dire che la retta di regressione ha equazione:

$$y = -41.598 - 2.961x$$

Residui

Dopo aver trovato la retta di regressione, possiamo osservare qual è il discostamento tra i valori osservati (le coppie (x_i,y_i)) e i valori stimati (le coppie $(x_i,\hat{y}_i)$).

I valori stimati sono espressi secondo l'equazione: $\hat{y}_i=\alpha+\beta x_i$ ottenuti mediante la retta di regressione. Risulta inoltre che la **media campionaria dei valori stimati è prossima alla media campionaria dei valori osservati**.

I residui sono dunque definiti come $E_i=y_i-\hat{y}_i=y_i-(\alpha+\beta x_i)$ e mostrano di quanto si discostano valori osservati e valori stimati.

Si nota sui residui:

- La media campionaria dei residui risulta sempre **nulla**
- La varianza dei residui è $S_E^2=\frac{1}{n-1}\sum_{i=1}^{n}E_i^2$ **in quanto la media campionaria è 0.**

Per calcolare il vettore dei valori stimati utilizziamo la funzione `fitted()`, passando come argomento `lm(y~x)`:

```
> fitted(lm(x2~x1))
       1        2        3        4        5        6        7        8
52.27155 52.86379 61.15510 50.49484 44.27636 46.34919 54.04826 48.71813
       9       10       11       12       13       14       15       16
47.82978 48.12590 46.94142 48.12590 49.31037 49.01425 48.12590 47.23754
      17       18       19       20       21       22       23       24
46.34919 50.49484 49.31037 50.49484 48.12590 49.31037 48.71813 49.90261
      25       26       27       28       29       30       31       32
46.05307 48.71813 48.71813 47.23754 46.94142 44.57248 43.38800 46.94142
      33       34       35       36       37       38       39       40
48.71813 46.05307 46.05307 47.82978 49.01425 44.57248 45.75695 47.23754
```

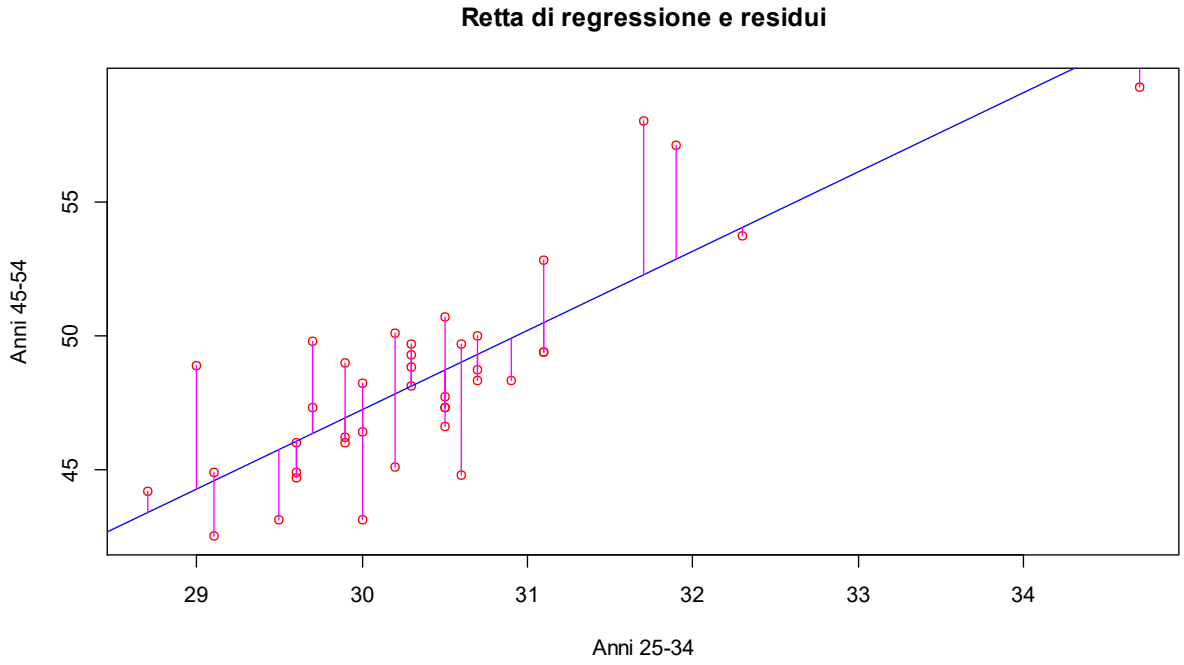
Per calcolare i residui invece di usa la funzione `resid()` passando anche in questo caso lo stesso argomento:

```
> resid(lm(x2~x1))
          1           2           3           4           5           6
 5.72844820  4.23621173 -1.85509878  2.30515759  4.62364048  3.45081285
          7           8           9          10          11          12
-0.34826120  1.98186699  2.27022169  1.57410345  2.05857638  1.17410345
         13          14          15          16          17          18
 0.68963052  0.68574876  0.67410345  0.96245815  0.95081285 -1.09484241
         19          20          21          22          23          24
-0.61036948 -1.09484241 -0.02589655 -1.01036948 -1.01813301 -1.60260594
         25          26          27          28          29          30
-0.05306892 -1.41813301 -1.41813301 -0.83754185 -0.74142362  0.32752224
         31          32          33          34          35          36
 0.81199517 -0.94142362 -2.11813301 -1.15306892 -1.35306892 -2.72977831
         37          38          39          40
-4.21425124 -2.07247776 -2.65695069 -4.13754185
```

Rappresentiamo dunque graficamente i valori dei residui, vediamo tre possibili rappresentazioni:

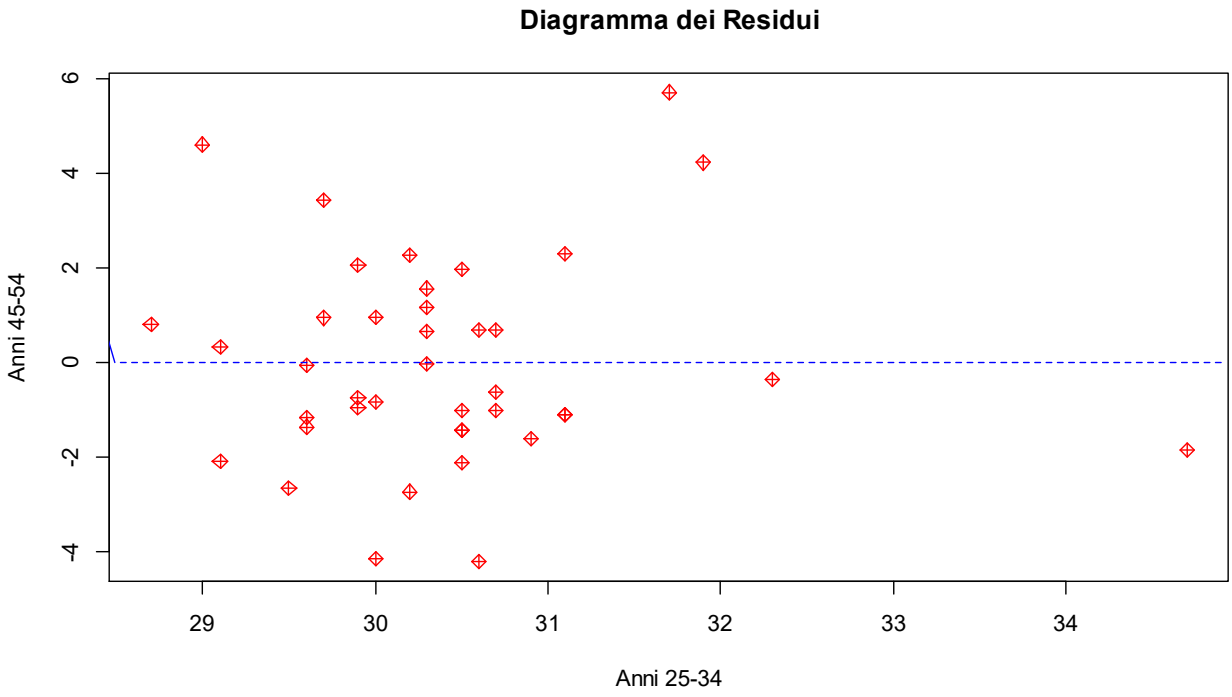
1. aggiungiamo al grafico dello scatterpolot e la retta di regressione dei **segmenti verticali che visualizzano i residui**: congiungiamo i punti (x_i,y_i) e i punti $(x_i, \hat{y}_i)$

```
plot(x1,x2,main="Retta di regressione e residui",
     xlab="Anni 25-34",ylab="Anni 45-54",
     col="red")
abline(lm(x2~x1),col="blue")
stime<-fitted(lm(x2~x1))
segments(x1,stime,x1,x2,col="magenta")
```



2. Usiamo un diagramma dei residui: un grafico in cui i valori dei residui sono posti sull'asse delle ordinate e quelli della variabile indipendente sull'asse delle ascisse

```
residui<-resid(lm(x2~x1))
plot(x1,residui,main="Diagramma dei Residui",
     xlab="Anni 25-34",ylab="Anni 45-54",
     col="red", pch =9)
abline(h=0,col="blue",lty=2)
```



Il diagramma dei residui aiuta a comprendere quale è **l'adattamento della retta di regressione rispetto ai dati**, consentendo di identificare quali sono le informazioni che hanno una forte **influenza sulla collocazione e direzione della retta di regressione**.

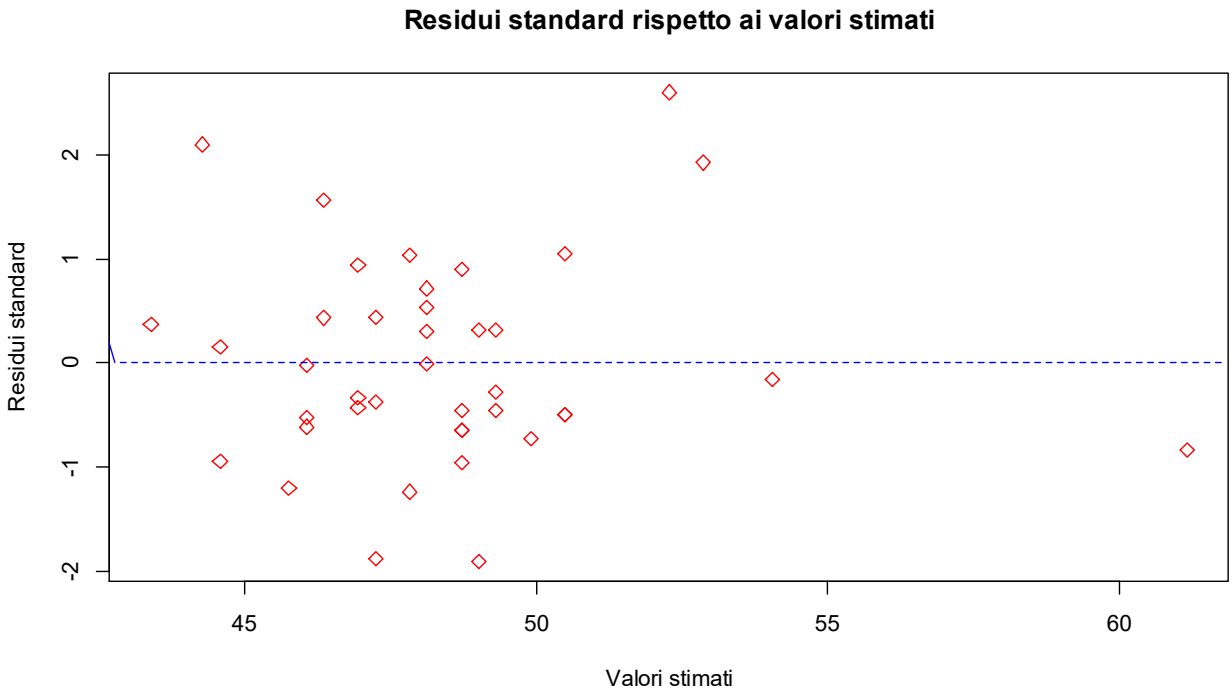
Occorre notare che la posizione della retta di regressione è fortemente influenzata dalla presenza di eventuali valori anomali che si discostano in modo significativo dagli altri. L'analisi dei residui aiuta ad individuare eventuali **punti isolati** (valori anomali) dovuti ad errori nella stima. Tali valori possono perturbare significativamente la stima dei parametri di regressione e influenzare l'interpretazione dei residui. Eliminando i valori anomali la varianza campionaria dei residui diminuisce.

3. Rappresentare i residui standardizzati rispetto ai valori stimati. I residui standardizzati sono definiti:

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

caratterizzati da **media nulla e deviazione standard pari a 1**.

```
residuistandard<-residui/sd(residui)
plot(stime,residuistandard,
     main="Residui standard rispetto ai valori stimati",
     xlab="Valori stimati",ylab="Residui standard",
     pch=5, col="red")
abline(h=0,col="blue",lty=2)
```



I punti indicano la **posizione dove si collocano i residui standardizzati rispetto ai valori stimati con la retta di regressione**. La retta orizzontale è posizionata nello zero, che corrisponde alla media campionaria dei residui standardizzati. Anche in questo caso i **punti sono disposti quasi casualmente attorno alla linea orizzontale e non si evidenzia nessuna tendenza particolare nella distribuzione dei punti**.

Coefficiente di determinazione

Un altro indice da considerare è il **coefficiente di determinazione**, definito come:

$$D^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

È definita cioè come il **rapporto tra varianza dei valori stimati** (con la retta di regressione) e **varianza dei valori osservati**.

Il coefficiente varia tra 0 e 1:

- Vicino ad **1 significa che i punti tenderanno ad allinearsi lungo la retta di regressione**
- Vicino a **0 non c'è una retta capace di interpolare i punti**

Nel caso della regressione lineare semplice il coefficiente di determinazione è il **quadrato del coefficiente di correlazione**, cioè

$$D^2 = r_{xy}^2$$

In R è possibile ottenete tale valore attraverso il seguente codice:

```
> (cor(anni.25to34,anni.45to54))^2
[1] 0.657874
```

Qui si può concludere che i punti tenderanno ad allinearsi un po' di più lungo la retta di regressione.

7.3.3.2 Regressione lineare multipla

Non è insolito avere dei casi in cui è opportuno e interessante avere più di una variabile indipendente, e in questi casi parliamo di **regressione lineare multipla**.

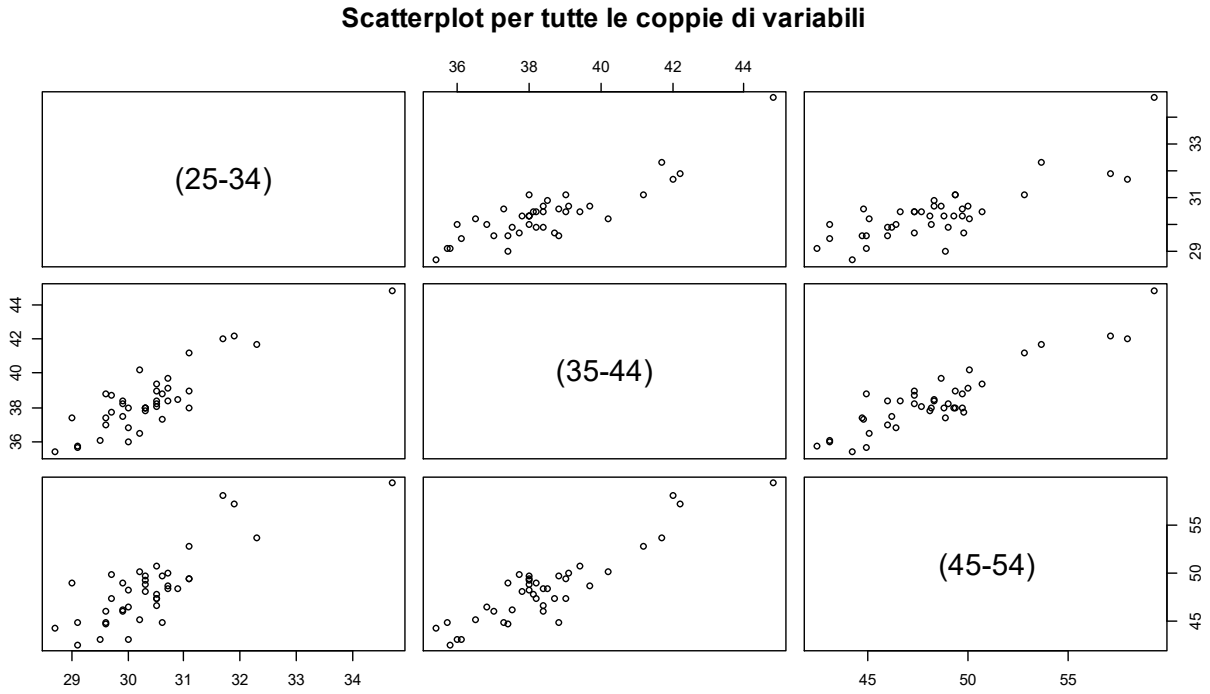
Vediamo dunque qual è la relazione tra la variabile dipendente che abbiamo scelto "Anni 45-54", che si trova sull'asse delle y, e le altre variabili del nostro dataset.

Le funzioni `cov()` e `cor()` applicate alla matrice che rappresenta il nostro campione forniscono due matrici di dimensione 4x4 i cui elementi **sono rispettivamente le covarianze e i coefficienti di correlazione** di ogni coppia di variabili.
Sulla diagonale nel primo caso avremo la varianza, mentre nel secondo caso avremo 1.
Come specificato all'inizio del capitolo consideriamo la matrice *mtxStipendiLordi2* che risulta contenere tutte le colonne delle fasce d'età tranne l'ultima

```
> cov(mtxStipendiLordi2)
         (25-34)  (35-44)   (45-54)
(25-34) 1.062660 1.711667  3.146731
(35-44) 1.711667 3.658564  6.544051
(45-54) 3.146731 6.544051 14.163872
> cor(mtxStipendiLordi2)
          (25-34)   (35-44)   (45-54)
(25-34) 1.0000000 0.8680938 0.8110943
(35-44) 0.8680938 1.0000000 0.9090764
(45-54) 0.8110943 0.9090764 1.0000000
```

Notiamo che la correlazione più forte la otteniamo in corrispondenza in "Anni 45-54" e "Anni 35-44"; mentre risultano poco correlate "Anni 25-34" e "Anni 45-54". Notiamo inoltre che la maggior parte delle variabili sono correlate tra loro positivamente.

Riportiamo il grafico contenente tutti gli scatterplot per le coppie di variabili per dare una visione di insieme con i dati appena visti:



Il modello di regressione lineare multipla con p variabili è esprimibile con la seguente equazione:

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

dove:

- α è l'intercetta
- $\beta_1, \beta_2, \ldots, \beta_p$ sono i **regressori**. β_i rappresenta **l'inclinazione di Y rispetto alla variabile X_i tenendo costanti le altre variabili**

Anche qui per stimare i parametri di regressione ricorriamo al **metodo dei minimi quadrati**, bisogna minimizzare la quantità:

$$Q = \sum_{i=1}^{n} \left[y_i - (\alpha + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p}) \right]^2$$

dove n è il numero delle osservazioni, $x_{1,j}, x_{2,j}, \ldots, x_{n,j}$ sono i valori osservati dalla variabile X_j e gli y_i sono i valori osservati dalla variabile dipendente.

Derivando rispetto ai parametri α, $\beta_1, \beta_2, \ldots, \beta_p$ si scopre che:

$$\alpha = \overline{y} - \beta_1 \overline{x_1} - \beta_2 \overline{x_2} - \cdots - \beta_p \overline{x_p}$$

In R per eseguire l'analisi di regressione lineare multipla usiamo sempre la funzione lm(y~x_1 + x_2 … +x_p), cioè y è la variabile dipendente da x_1, x_2, …, x_p.

Vediamo dunque il codice:

```
> linearModelMultiplo<-lm(anni.45to54~anni.25to34+
+    anni.35to44)
> linearModelMultiplo

Call:
lm(formula = anni.45to54 ~ anni.25to34 + anni.35to44)

Coefficients:
(Intercept)  anni.25to34  anni.35to44
   -24.3862       0.3249       1.6367
```

Quindi l'intercetta α= -24.3862, mentre i regressori sono $\beta_1 = 0.3249$, $\beta_2 = 1.6367$ e il modello di regressione multipla risulta pertanto:

$$y = -24.3862 + 0.3249x_1 + 1.6367x_2$$

Notiamo che tutti i regressori sono positivi, questo significa sono legate positivamente per l'età compresa tra i 25-34 e 35-44.

Anche in questo caso `lm()` restituisce una serie di attributi, tra cui anche i coefficienti di regressione.

Residui

I residui come detto mostrano di quanto si discostano i valori osservati da quelli stimati con la retta di regressione.

Sono definiti come

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

In modo analogo con la regressione lineare semplice, si nota che:

- La media campionaria risulta sempre **nulla**
- La varianza dei residui è $S_E^2 = \frac{1}{n-1}\sum_{i=1}^{n} E_i^2$ dato che la media campionaria è 0

Per calcolare il vettore dei valori stimati utilizziamo la funzione `fitted()`,passando come argomento lm(y~x_1 + x_2 … +x_p):

```
linearModelMultiplo<-lm(anni.45to54~anni.25to34+
   anni.35to44)
```

```
> fitted(linearModelMultiplo)
       1        2        3        4        5        6        7        8
54.65440 55.04672 60.21187 53.15010 46.24839 46.96684 54.35835 50.00913
       9       10       11       12       13       14       15       16
51.22099 47.65280 47.85016 47.65280 49.58311 49.05962 47.65280 47.55532
      17       18       19       20       21       22       23       24
48.60352 49.54942 50.56512 47.91274 47.32546 48.43744 47.88145 48.66609
      25       26       27       28       29       30       31       32
45.78867 49.35446 48.04512 45.59131 46.70449 43.49853 42.87756 48.17750
      33       34       35       36       37       38       39       40
48.37245 48.73469 46.44334 45.16529 46.60460 43.66220 44.28317 44.28197
```

Per calcolare i residui invece di usa la funzione `resid()` passando anche in questo caso lo stesso argomento:

```
> residui
          1           2           3           4           5           6
 3.34559918  2.05327865 -0.91187352 -0.35010390  2.65161184  2.83316059
          7           8           9          10          11          12
-0.65835352  0.69086956 -1.12099384  2.04720195  1.14983709  1.64720195
         13          14          15          16          17          18
 0.41688730  0.64038288  1.14720195  0.64467979 -1.30351596 -0.14941549
         19          20          21          22          23          24
-1.86511863  1.48726105  0.77453726 -0.13743912 -0.18145093 -0.36609200
         25          26          27          28          29          30
 0.21132679 -2.05445982 -0.74511858  0.80869165 -0.50448932  1.40146936
         31          32          33          34          35          36
 1.32244277 -2.17749822 -1.77245389 -3.83469100 -1.74334383 -0.06529061
         37          38          39          40
-1.80460230 -1.16219829 -1.18317171 -1.18196711
```

Dei residui possiamo calcolare **mediana, varianza e deviazione standard**, mentre non è possibile calcolare il coefficiente di variazione in quanto la media dei residui è 0.

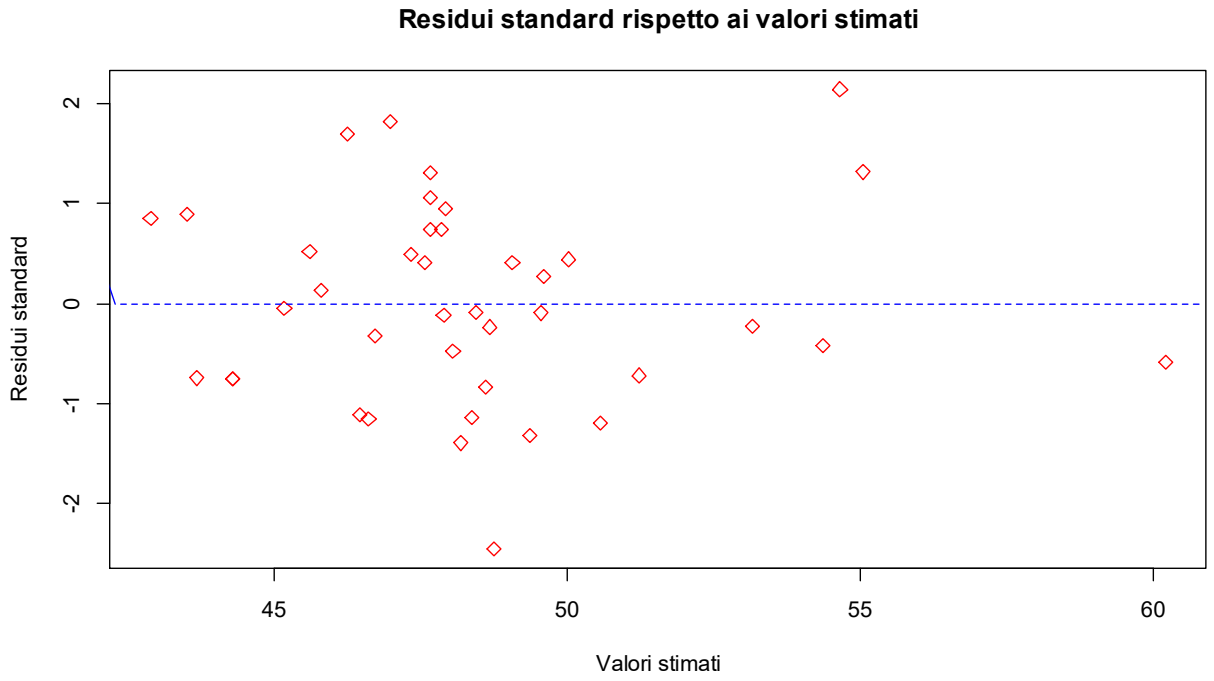
Vediamo dunque nel nostro caso:

```
> median(residui)
[1] -0.1434273
> var(residui)
[1] 2.430921
> sd(residui)
[1] 1.559141
```

Nel caso multivariato è interessante calcolare i **residui standardizzati**:

```
> residui/sd(residui)
          1           2           3           4           5           6
 2.14579605  1.31692919 -0.58485625 -0.22454919  1.70068735  1.81712885
          7           8           9          10          11          12
-0.42225393  0.44310902 -0.71898157  1.31303173  0.73748102  1.05648025
         13          14          15          16          17          18
 0.26738263  0.41072794  0.73579090  0.41348388 -0.83604737 -0.09583192
         19          20          21          22          23          24
-1.19624736  0.95389756  0.49677170 -0.08815052 -0.11637876 -0.23480361
         25          26          27          28          29          30
 0.13554050 -1.31768677 -0.47790319  0.51867760 -0.32356871  0.89887260
         31          32          33          34          35          36
 0.84818663 -1.39660098 -1.13681417 -2.45948913 -1.11814360 -0.04187601
         37          38          39          40
-1.15743348 -0.74540923 -0.75886113 -0.75808853
```

```
stimemult<-fitted(linearModelMultiplo)
residuimultstandard<-residui/sd(residui)
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)
```



I punti ci dicono dove si trovano i residui standardizzati rispetto ai valori stimati con la retta di regressione. In questo caso non si evidenzia nessuna tendenza particolare nel grafico.

Coefficiente di determinazione

Abbiamo già visto in precedenza che il coefficiente di determinazione è definito come il rapporto tra la varianza dei valori stimati con la retta di regressione e la varianza dei valori osservati della variabile dipendente.

L'indice D^2 risulta compreso tra 0 e 1, più è vicino a 1 meglio il modello usato riesce a spiegare i dati.
Per calcolarlo analogamente a prima si può fare:

```
> summary(linearModelMultiplo)$r.square
[1] 0.8283717
```

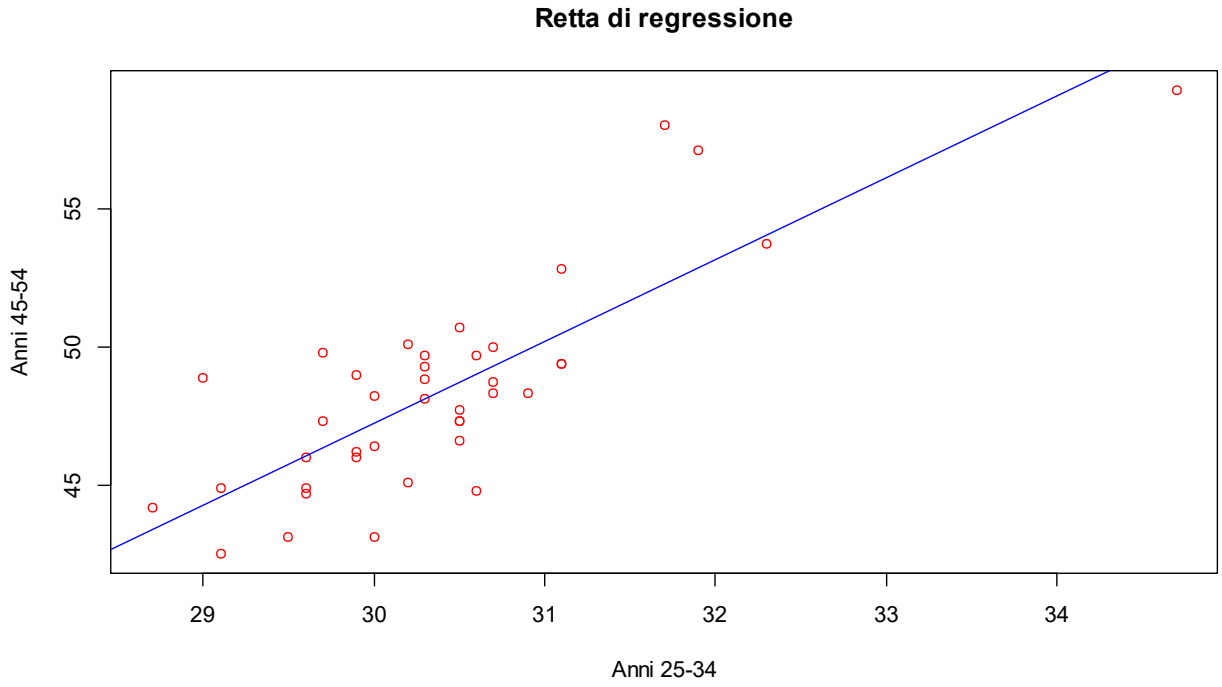
Nel nostro caso quindi, essendo il valore molto vicino ad 1, il **modello di regressione spiega in maniera molto significativa i dati**.

7.3.3.3 Regressione non lineare

L'utilizzo di un modello lineare non è sempre la scelta migliore, in alcuni casi non è opportuno utilizzare una retta per approssimare i nostri dati e quindi si ricorre alla **regressione non lineare**.

Di seguito il grafico:

```
x1<-anni.25to34
x2<-anni.45to54

plot(x1 ,x2 ,main="Retta di regressione",
     xlab="Anni 25-34",ylab="Anni 45-54", col ="red")
abline(lm(x2~x1), col="blue")
```



Come vediamo dal grafico, infatti, la retta di regressione non approssima i dati adeguatamente, mentre è intuibile che una curva potrebbe essere molto più adatta allo scopo.

Calcoliamo il coefficiente di determinazione tramite modello lineare:

```
> summary(lm(x2~x1))$r.square
[1] 0.657874
```

Come ci aspettavamo il risultato ci conferma quanto detto.

Attraverso alcune trasformazioni è possibile però linearizzare modelli che sembrano non lineari, questo ci permette di usare comunque un modello lineare.

Consideriamo dunque il modello non lineare

$$Y = \alpha + \beta X + \gamma X^2$$

Su questo modello possiamo però ricorrere alla regressione multipla per stimare i parametri α, β e γ:

$$Y = \alpha + \beta X_1 + \gamma X_2$$

con regressori X_1=X e X_2=X^2

Possiamo facilmente stimare i parametri in modo analogo a come fatto in precedenza servendoci questa volta anche di un **identificatore di variabile** utilizzato quando si devono effettuare **operazioni matematiche nelle variabili della regressione**:

```
> regressionePolinomiale <-lm(x2~x1+I((x1)^2))
> regressionePolinomiale

Call:
lm(formula = x2 ~ x1 + I((x1)^2))

Coefficients:
(Intercept)           x1    I((x1)^2)
  -95.48038      6.40631      -0.05496
```

Otteniamo dunque i seguenti valori $\alpha = -95.48038,\ \beta = 6.40631,\ \gamma = -0.05496$ quindi

$$Y = -95.48038 + 6.40631X - 0.05496X^2$$

Calcoliamo dunque anche il coefficiente di determinazione per verificare la correttezza del modello statistico utilizzato:

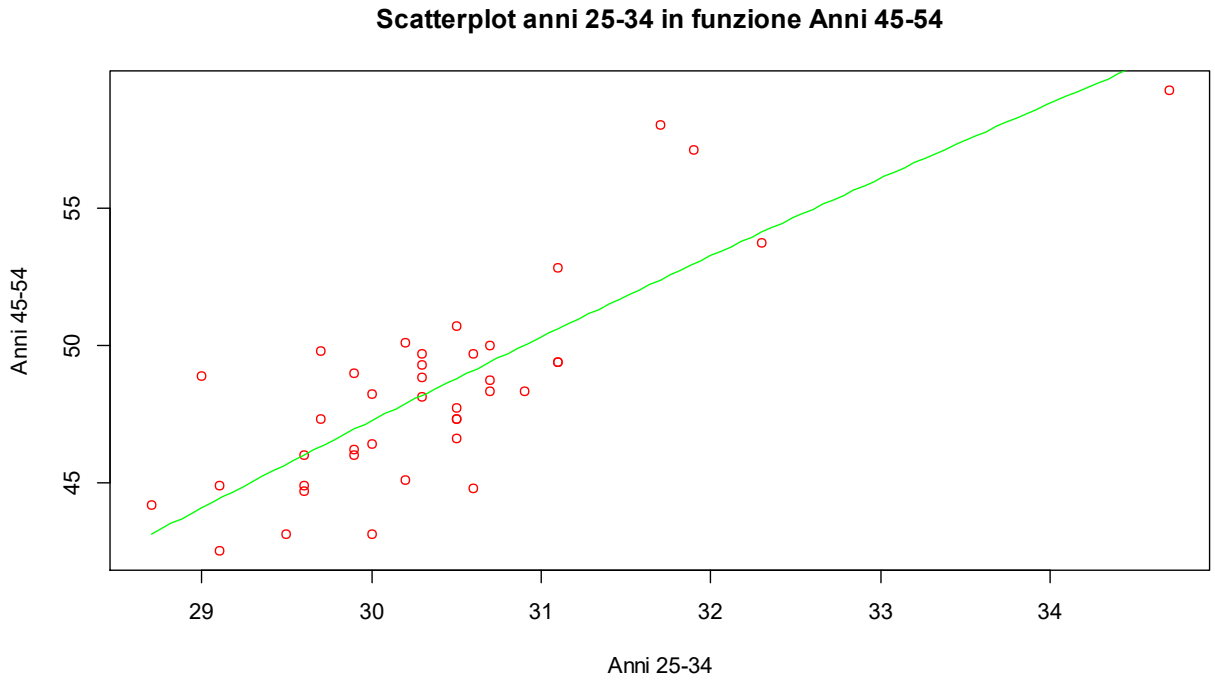
```
> summary (regressionePolinomiale)$r.square
[1] 0.6589346
```

Il risultato seppur di pochissimo, è migliore rispetto a quello lineare, quindi abbiamo "migliorato" la nostra stima.

	Valore
Lineare	0.657874
Non lineare	0.6589346

Disegniamo dunque la curva stimata sullo scatterplot:

```
plot(x1,x2,main="Scatterplot anni 25-34 in funzione Anni 45-54"
     ,xlab="Anni 25-34",ylab="Anni 45-54" , col = "red")
alpha <- regressionePolinomiale$coefficients[[1]]
beta <- regressionePolinomiale$coefficients[[2]]
gamma <- regressionePolinomiale$coefficients[[3]]

curve (alpha+beta*x+gamma*x^2, add=TRUE, col = "green")
```



8. Cluster

L'analisi dei cluster o clustering è il compito di **dividere in gruppi** un insieme di oggetti in modo tale che gli oggetti nello stesso gruppo (un gruppo viene detto cluster appunto) sono più simili tra loro rispetto agli oggetti degli altri gruppi (gli altri cluster).

L'obiettivo del clustering è quello di individuare i legami esistenti tra i dati in analisi, permettono infatti di:

- Esplorare i dati
- Generare ipotesi sulla natura dei dati e verificare ipotesi
- Ridurre la complessità dei dati
- Semplificazione dei problemi senza significative perdite di informazioni
- Classificazione in tipi

Esistono molti algoritmi che differiscono principalmente nella definizione della costituzione dei cluster e come efficientemente trovarli. In base all'obiettivo e ai parametri che si sceglie di utilizzare (funzione di distanza da usare, numero di cluster, …) ci sono algoritmi più o meno adatti; quindi, la scelta va fatta in base al proprio contesto (dati) e fine (utilizzo che se ne vuole fare dei risultati).

L'analisi dei cluster è un processo iterativo di scoperta e ottimizzazione che implica tentativi ed errori: non è insolito dover modificare i dati da processare e i parametri del modello finché i risultati non raggiungano le proprietà che si desiderano.

Più formalmente abbiamo un insieme $I=\{I_1,I_2, ..., I_n\}$ di n individui, mentre $C=\{C_1,C_2, ..., C_p\}$ è l'insieme delle caratteristiche osservabili e possedute da ogni individuo, il problema del clustering consiste nel determinare m sottoinsiemi di individui di I, tale che ogni I_i appartenga ad un solo insieme e che gli individui assegnati allo stesso cluster siano simili, mentre gli individui assegnati a diversi cluster siano dissimili.

Considerando dunque il nostro dataset, abbiamo 40 individui (atenei) e 4 caratteristiche per ciascun individuo (fasce d'età). Vogliamo trovare un certo numero di sottoinsiemi dei 40 atenei che siano concordi con la definizione di clustering data.

8.1 Problematiche clustering

Prima di procedere con la vera e propria analisi dei cluster bisogna considerare alcune problematiche che ci possono essere.

Bisogna considerare la **standardizzazione delle variabili**: se le caratteristiche avessero un peso diverso ci potrebbero essere risultati differenti in base alla tecnica di clustering considerata. La standardizzazione viene raccomandata e deve essere effettuata usando la media campionaria e la deviazione standard campionaria entrambe derivate dall'insieme completo di individui della popolazione. Nel nostro caso però è preferibile utilizzare i pesi proporzionali alle percentuali essendo già prive delle unità di misura, usiamo quindi delle misure non standardizzate.

Un altro problema riguarda la correlazione delle variabili. È preferibile, infatti, che non ci siano variabili correlate tra loro poiché tendono a falsare i risultati che si ottengono.

È opportuno infine ridurre le variabili tramite l'analisi delle componenti principali, tecnica che permette di diminuire il numero di variabili a solo quelle strettamente necessarie, cioè quelle principali. Ciò è necessario in quanto il numero di variabili fa crescere il tempo di calcolo di molto rendendo l'analisi più complessa.

Ovviamente non si applica al nostro caso visto che abbiamo un dataset molto piccolo.

8.2 Distanza e similarità

Per risolvere il problema è necessario definire cosa si intende per **somiglianza o differenza tra due individui**.

Possiamo usare come metrica per definire se due individui sono simili o meno i **coefficienti di similarità**, oppure le **misure di distanza**. I primi hanno la caratteristica di assumere i valori tra 0 e 1, mentre le distanze possono assumere qualunque **valore maggiore o uguale a 0**.

Introduciamo dunque il concetto di funzione distanza sul quale si basano molte delle misure di somiglianza.

Si dice che $d(X_i,X_j)$ è una funzione distanza se soddisfa le seguenti condizioni:

- $d(X_i,X_j) = 0$ se e solo se $X_I = X_J$, cioè **distanza nulla** implica uguaglianza
- $d(X_i,X_j) \geq 0 \ \ \forall \ X_i,X_j$, cioè distanza è **non negativa**
- $d(X_i,X_j) = d(X_j,X_i) \ \ \forall \ X_i,X_j$, cioè indica la **simmetria**
- $d(X_i,X_j) \leq d(X_i,X_k)+d(X_k,X_j) \ \ \forall \ \ X_i, X_j , X_k$, è la proprietà di **disuguaglianza triangolare**

Quello che bisogna fare è costruire una **matrice delle distanze**: tenendo conto che abbiamo n individui, per ogni individuo dobbiamo sapere la distanza con gli n-1 altri individui e tenendo conto che c'è la proprietà di simmetria, in totale dobbiamo conoscere $\frac{n(n-1)}{2}$ distanze.

In R per calcolare la matrice delle distanze si opera con il seguente comando:

```
dist(mtxStipendiLordi2, method = "euclidean",
     diag = FALSE, upper = FALSE)
```

Con le righe di codice scritte non abbiamo fatto altro che calcolare la matrice delle distanze, usando la distanza euclidea, tra gli individui della nostra matrice (sarebbero le righe, le regioni).

Esistono varie metriche per calcolare la distanza, come ad esempio la distanza euclidea che è stata usata nell'esempio di sopra ed è quella di default utilizzata dalla funzione dist.

Le varie opzioni sono:

- Metrica euclidea
- Metrica del valore assoluto o Manhattan
- Metrica del massimo o di Chebychev

- Metrica di Minkowski
- Distanza di Camberra
- Distanza di Jaccard

Vediamo dunque nel dettaglio la distanza euclidea e successivamente le altre metriche.

8.2.1 Metriche distanza

La **metrica euclidea** è la misura di distanza più famosa, definita così:

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

dove x_{ik} è il valore della k-esima caratteristica dell'individuo I_i.

La distanza euclidea è **influenzata dall'unità di misura** utilizzata e ne è legata in maniera non invariante, cioè non esiste una trasformazione che permetta di passare dai valori di una distanza euclidea a un'altra con valori che differiscono per unità di misura.
Già abbiamo visto come si calcola in R la distanza euclidea.

La metrica del valore assoluto è definita:

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

```
as.matrix(dist(mtxStipendiLordi2, method = "manhattan",
     diag = FALSE, upper = FALSE))
```

La metrica del massimo è definita:

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

```
as.matrix(dist(mtxStipendiLordi2, method = "maximum",
     diag = FALSE, upper = FALSE))
```

La metrica di **Minkowski** è definita:

```
as.matrix(dist(mtxStipendiLordi2, method = "minkowski",
     p=3, diag = FALSE, upper = FALSE))
```

dove *p* indica la potenza della distanza di Minkowski.

Se r ≥ 1. Si possono considerare dei casi speciali:

- **Se r=2** allora quella che si ottiene la metrica euclidea
- **Se r=1** si ottiene la metrica del valore assoluto
- **Se r =** ∞ si ottiene la metrica di Chebychev

Inoltre, per ogni coppia di valori X_i,X_j e per ogni intero r e k tali che r≥k vale la disuguaglianza $d_r(X_i,X_j) \leq d_k(X_i,X_j)$ ed implica che

$$d_\infty(X_i,X_j) \leq d_2(X_i,X_j) \leq d_1(X_i,X_j)$$

La metrica di **Canberra** è definita:

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

```
as.matrix(dist(mtxStipendiLordi2, method = "canberra",
     diag = FALSE, upper = FALSE))
```

A differenza delle precedenti metriche non è necessario scalare la matrice perché i contributi della somma sono adimensionali.
La metrica è definita per variabili non negative, e ha il problema che se uno dei due valori x_{ik},x_{jk} è uguale a zero allora il contributo nella sommatoria è pari a 1 (il massimo).
È inoltre poco sensibile (poco preciso) all'asimmetria delle distribuzioni e ai valori anomali.

Infine, vi è la **metrica di Jaccard** applicabile solo ai vettori binari in R, definito come:

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

8.2.2 Misure di similarità

Oltre a poter calcolare la matrice delle distanze, è possibile anche calcolare la **matrice delle similarità**.
Una misura di similarità differisce dalle misure di distanza fornendo un **valore compreso tra 0 e 1**, dove 0 indica l'assenza totale di similarità, mentre 1 la massima presenza di somiglianza.

Una funzione $s_{ij}=s(X_i,X_j)$ è una misura di similarità se:

- $s(X_i,X_i)=1$ (similarità è unitaria se i due punti sono incidenti)
- $0 \leq s(X_i,X_j) \leq 1$ (range di similarità)
- $s(X_i,X_j) = s(X_j,X_i), \quad \forall X_i,X_j$ (simmetria)

È importante dire che è sempre possibile trasformare una misura di distanza in una di similarità, ma non sempre è possibile il contrario (potrebbe non essere rispettata la disuguaglianza triangolare).

La formula che permette di trasformare la misura di distanza a quella di similarità è:

$$s_{ij} = \frac{1}{1 + d_{ij}} \qquad (i, j = 1,2, \dots, n)$$

8.3 Misure di non omogeneità totale

Abbiamo introdotto dunque fino ad ora la **matrice delle misure** che ci indica per **riga il vettore delle misure delle caratteristiche dell'individuo i-esimo**, mentre per **colonna ci indica le misure che assume la caratteristica j-esima per ogni individuo**, e successivamente abbiamo visto la matrice delle distanze che ci da informazioni riguardo la distanza tra due individui calcolata con uno dei metodi visti in precedenza.

Introduciamo ora una matrice W_X di cardinalità $p_x p$ definita come la matrice delle **varianze e covarianze** dove il generico elemento w_{rl} è uguale a

$$\frac{1}{n-1}\sum_{i=1}^{n}(x_{ir} - \bar{x}_r)(x_{il} - \bar{x}_l) \qquad (r, l = 1, 2, \dots, p)$$

Notiamo che se r = *l* allora W_{rl} è la **varianza campionaria** della caratteristica r-esima, altrimenti è la **covarianza** tra la caratteristica r-esima e l-esima effettuate entrambe su tutti gli individui.
La matrice è ottenibile applicando la funzione `cov()` sulla matrice delle misure delle caratteristiche.

Partendo dalla matrice W_X definiamo la **matrice statistica di non omogeneità**:

$$H_I = (n-1)W_I$$

dove l'elemento generico h_{rl} è uguale a:

$$\sum_{i=1}^{n}(x_{ir} - \bar{x}_r)(x_{il} - \bar{x}_l) = (n-1)w_{rl} \qquad (r, l = 1, 2, \dots, p)$$

Si nota che quando r = *l* l'elemento $h_{rl} = (n-1)Var(C_r) = (n-1)s_r^2$

Possiamo definire allora la **misura di non omogeneità statistica** di un dato insieme di individui I la traccia della matrice H_I:

$$trH_I = \sum_{r=1}^{p} h_{rr} = (n-1)\sum_{r=1}^{p} s_r^2$$

che può essere scritta anche come:

$$trH_I = \sum_{i=1}^{n} d_2^2(X_i, \bar{X})$$

dove d_2 indica la distanza euclidea e $\bar{X}$ è il vettore che contiene le **medie campionarie** di tutte le p caratteristiche sugli n individui.

Si dimostra inoltre che $trH_I = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} d_2^2\,(X_i, X_j)$, cioè che la traccia della matrice di non omogeneità corrisponde al **rapporto tra la somma dei quadrati degli elementi al di sotto della diagonale principale della matrice delle distanze euclidee e il numero n di individui**.

Sono tre i fattori che incidono sul calcolo della misura di non omogeneità statistica e sono la distanza euclidea che ha un ruolo rilevante da come abbiamo visto e dipende anche dalla numerosità del campione (n) e dalla varianza delle caratteristiche (omogeneità interna).

Oltre a considerare le misure di non omogeneità relative all'insieme totale di individui, dobbiamo introdurre anche le misure di non omogeneità interne ai cluster e misure di non omogeneità tra cluster.

8.4 Misure di non omogeneità tra cluster

Come abbiamo detto quello che vogliamo ottenere è che gli **individui appartenenti allo stesso cluster siano quanto più possibile omogenei tra loro e il più possibile differenti da quelli appartenenti agli altri cluster individuati**.

Quello che facciamo allora è considerare una misura di non omogeneità interna ai cluster (**within**) e una misura di non omogeneità tra cluster (**between**).

Consideriamo la seguente espressione:

$$T = S + B$$

- T è la matrice di non omogeneità statistica totale ed è fissata;
- S è la somma delle matrici di non omogeneità statistica relative ai singoli m cluster;
- B è la matrice di non omogeneità statistica tra i cluster.

Come è facile immaginare S e B dipendono da come avviene la suddivisione in cluster.

Per ogni partizione dell'insieme I degli n individui in m fissati cluster, otteniamo un'equazione come quella vista sopra da cui segue:

$$trT = trS + trB$$

O in modo equivalente

$$1 = \frac{trS}{trT} + \frac{trB}{trT}$$

La **traccia di T è univocamente determinata** per ogni matrice che descrive p caratteristiche di n individui, allora fissato un numero m di suddivisioni, i cluster devono essere individuati in modo da **minimizzare la misura di non omogeneità statistica interna ai cluster**, e **massimizzare la misura di non omogeneità statistica tra i gruppi**.

Utilizziamo queste misure per capire quale suddivisione in cluster risulta essere la migliore per il nostro insieme di 40 atenei.

8.5 Metodi di ottimizzazione

I metodi per decidere come effettuare il clustering si suddividono in tre tipologie:

- Metodi di **enumerazione completa**
- Metodi **gerarchici**
- Metodi **non gerarchici**.

Il primo metodo non è applicabile poiché si basa su tecniche di ottimizzazione che sono computazionalmente onerose dato che prevedono il calcolo della funzione obiettivo (minimizzare la traccia della matrice B, o massimizzare la traccia della matrice S) per **ogni possibile partizione dell'insieme totale di *n* individui in *m* cluster**.

Per tale motivo si vanno ad utilizzare i metodi di raggruppamento **gerarchici** e **non gerarchici** che operano su una sottoclasse delle partizioni degli individui in cluster.

8.6 Metodi non gerarchici

Quello che si vuole ottenere con questi metodi è una **partizione unica** degli *n* individui. Questa analisi permette di ottenere il numero di cluster da utilizzare per i metodi gerarchici agglomerativi. Per comprendere quale sia il numero di cluster ottimale bisogna prendere in considerazione il valore ottenuto da `between_SS / total_SS` che deve essere prossimo al 70%, ma non inferiore ad esso.

Generalmente un metodo non gerarchico, data una partizione iniziale, procedono riallocando gli individui nel gruppo con il **centroide più vicino**, fino ad arrivare al passo in cui per ogni individuo la **distanza rispetto al centroide del proprio gruppo è minima**. Il metodo più utilizzato è **k-means**. Per questo metodo bisogna specificare il numero di cluster che si vuole ottenere a priori.

Vediamo dunque i passi di seguito:

- **Specificare a priori il numero k di cluster e specificare m punti di riferimento iniziali** per produrre una prima partizione provvisoria;
- Per ogni individuo determinare il cluster di appartenenza, cioè quello individuato dal **punto di riferimento da cui ha distanza minore;**
- Calcolare il **centroide di ognuno dei k gruppi ottenuti**: questi centroidi sono i punti di riferimento per i nuovi cluster;
- **Rivaluta la distanza di ogni individuo da ogni centroide** e nel caso in cui la distanza minima sia col centroide di un altro gruppo, l'individuo viene spostato in quel gruppo.
- Si **ricalcolano i centroidi;**
- Si ripete il passo 4 e 5 **finché non si arriva al punto che nessun individuo viene spostato.**

Come misura di distanza viene utilizzata la **distanza euclidea** e si considerano i **quadrati della matrice delle distanze**.

Non si tratta di un metodo di ottimizzazione, infatti **si ottengono ottimi locali**: in base alla partizione iniziale possono ottenere risultati migliori.

8.6.1 Test k-means a 2 cluster

Applichiamo dunque k-means alla nostra matrice come input un **numero di cluster pari a 2**. Si noti che per essere sicuri di trovare una buona suddivisione tra tutte quelle possibili, nstart è posto a 10 dunque ci saranno dieci tentativi e il numero massimo di iterazioni è posto a 20.

```
km<-kmeans(mtxStipendiLordi2, centers = 2,
           iter.max = 20, nstart = 10)
km
```

```
K-means clustering with 2 clusters of sizes 5, 35

Cluster means:
   (25-34)  (35-44)  (45-54)
1 32.34000 42.38000 56.18000
2 30.10857 37.86571 47.27143

Clustering vector:
Cattolica del Sacro Cuore         LUISS Guido Carli             Luigi Bocconi
                        1                         1                         1
                P.Torino                   Perugia                    Verona
                        1                         2                         2
                P.Milano                   Brescia  Modena e Reggio Emilia
                        1                         2                         2
                 Bergamo                  U.Milano             La Sapienza
                        2                         2                         2
                   Parma                      Pisa                   Marche
                        2                         2                         2
                 Bologna                   Venezia        Roma Tor Vergata
                        2                         2                         2
                  Padova                     Siena                  Trieste
                        2                         2                         2
                   Udine                    Genova                    Pavia
                        2                         2                         2
                 Catania                    Trento                Roma Tre
                        2                         2                         2
                U.Torino                    Aquila                  U.Bari
                        2                         2                         2
                Cagliari                   Bicocca                  P.Bari
                        2                         2                         2
                 Ferrara                   Firenze                 Palermo
                        2                         2                         2
            Federico II                    Messina              Parthenope
                        2                         2                         2
                Calabria
                        2

Within cluster sum of squares by cluster:
[1]  47.0680 231.2977
 (between_SS / total_SS =  62.2 %)

Available components:

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

L'output di k-means ci dice che dal clustering ottenuto tramite l'algoritmo risulta che la misura di non omogeneità statistica tra cluster è pari a:

```
> km$betweenss/km$totss
[1] 0.6220521
```

Il risultato ottenuto da k-means a due cluster non è soddisfacente poiché `between_SS / total_SS = 62.2 %` ci fornisce un valore al di sotto del 70 %

Vediamo se con un numero di cluster pari a 3 il risultato migliora.

8.6.2 Test k-means a 3 cluster

Applichiamo dunque k-means alla nostra matrice come input un **numero di cluster pari a 3**.

```
km<-kmeans(mtxStipendiLordi2, centers = 3,
           iter.max = 20, nstart = 10)
km
```

```
K-means clustering with 3 clusters of sizes 13, 5, 22

Cluster means:
   (25-34)  (35-44)  (45-54)
1 29.67692 36.82308 44.76154
2 32.34000 42.38000 56.18000
3 30.36364 38.48182 48.75455

Clustering vector:
Cattolica del Sacro Cuore         LUISS Guido Carli             Luigi Bocconi
                        2                         2                         2
                P.Torino                    Perugia                    Verona
                        2                         3                         3
                P.Milano                    Brescia    Modena e Reggio Emilia
                        2                         3                         3
                 Bergamo                   U.Milano               La Sapienza
                        3                         3                         3
                   Parma                      Pisa                    Marche
                        3                         3                         3
                 Bologna                    Venezia          Roma Tor Vergata
                        3                         3                         3
                  Padova                     Siena                   Trieste
                        3                         3                         3
                   Udine                    Genova                     Pavia
                        3                         3                         3
                 Catania                    Trento                  Roma Tre
                        1                         3                         3
                U.Torino                    Aquila                    U.Bari
                        1                         1                         1
                Cagliari                   Bicocca                    P.Bari
                        1                         1                         3
                 Ferrara                   Firenze                   Palermo
                        1                         1                         1
            Federico II                    Messina                Parthenope
                        1                         1                         1
               Calabria
                        1
```

```
Within cluster sum of squares by cluster:
[1] 35.11692 47.06800 39.55818
 (between_SS / total_SS =  83.5 %)

Available components:

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

L'output di k-means ci dice che dal clustering ottenuto tramite l'algoritmo risulta che la misura di non omogeneità statistica tra cluster è pari a:

```
> km$betweenss/km$totss
[1] 0.8347047
```

Il risultato ottenuto da k-means a tre cluster è soddisfacente poiché `between_SS / total_SS =  83.5 %` ci fornisce un valore al di sopra e prossimo al 70 %

Dunque, **il numero dei cluster** che utilizzeremo **per i metodi gerarchici agglomerativi sarà pari a 3**. Questo affinché possano essere confrontati a parità di cluster e scoprire qual è il migliore.

8.7 Metodi gerarchici

I metodi gerarchici operano eseguendo una sequenza ordinata di operazioni della stessa natura. Possiamo distinguere metodi gerarchici **agglomerativi** e metodi **gerarchici divisivi**.

- I primi operano partendo da *n* gruppi formati da un singolo individuo e procedono aggregando degli insiemi ad ogni passo fino ad ottenere un unico gruppo.
- Gli altri invece partono da un singolo gruppo formato da tutte le unità accorpate e procedono dividendo ad ogni passo i gruppi finché non si ottengono gruppi di un singolo elemento.

I metodi gerarchici utilizzano le distanze per determinare le aggregazioni o le divisioni, e forniscono dunque una visione dell'insieme in termini di distanza (**dendrogramma**) e non obbligano il dover scegliere i parametri a priori.
Uno svantaggio invece è quello che questi metodi non permettono di riallocare gli individui assegnati a un gruppo in un livello precedente.
L'obiettivo dei metodi gerarchici è quello di ottenere una sequenza di partizioni che vengono rappresentate graficamente tramite una struttura ad albero detto dendrogramma in cui sulle ordinate sono riportati i livelli di distanza, mentre sulle ascisse ci sono i singoli individui. Ad ogni livello corrisponde un partizionamento.

Attraverso un dendrogramma abbiamo un quadro completo della struttura dell'insieme in termini delle distanze tra gli individui.
Utilizzando il dendrogramma è facile capire a che livello fermarsi per ottenere un clustering buono.

8.7.1 Metodi gerarchici agglomerativi

Molti dei metodi di questa tipologia hanno una **struttura comune** divisa in vari passi che riportiamo di seguito:

1. Predisporre la matrice dei dati, scalata o meno in base al caso, e **calcolare la matrice D delle distanze** degli *n* individui.
2. Individuare la **coppia di cluster con distanza minore e unirli in un unico cluster**. Calcolare la distanza tra questo nuovo cluster e tutti gli altri cluster già esistenti.
3. Costruire la nuova matrice delle distanze D su questo nuovo gruppo di cluster (la matrice avrà una riga e una colonna in meno)
4. Operare analogamente al passo 2 sulla matrice ottenuta **fino ad esaurire tutti i possibili raggruppamenti** (n-1 passi)
5. Rappresentare il processo di agglomerazione tramite un **dendrogramma**

Tra i vari metodi gli unici cambiamenti che ci sono si verificano nel passo 1 e nel passo 2.

Nel passo 1 la **scelta della misura di distanza** influenza richiedendo più o meno forti proprietà.
Il passo 2 caratterizza i metodi in base a come vengono **individuati i cluster meno distanti** e per il modo in cui si **determinano le distanze** con i cluster ottenuti man mano.

L'analisi gerarchica di tipo agglomerativo viene effettuata in R attraverso la funzione a cui passiamo la matrice delle distanze e il metodo gerarchico specifico da utilizzare:

```
> hclust(dist(mtxStipendiLordi2),method="complete")

Call:
hclust(d = dist(mtxStipendiLordi2), method = "complete")

Cluster method   : complete
Distance         : euclidean
Number of objects: 40
```

Le opzioni disponibili per method sono:

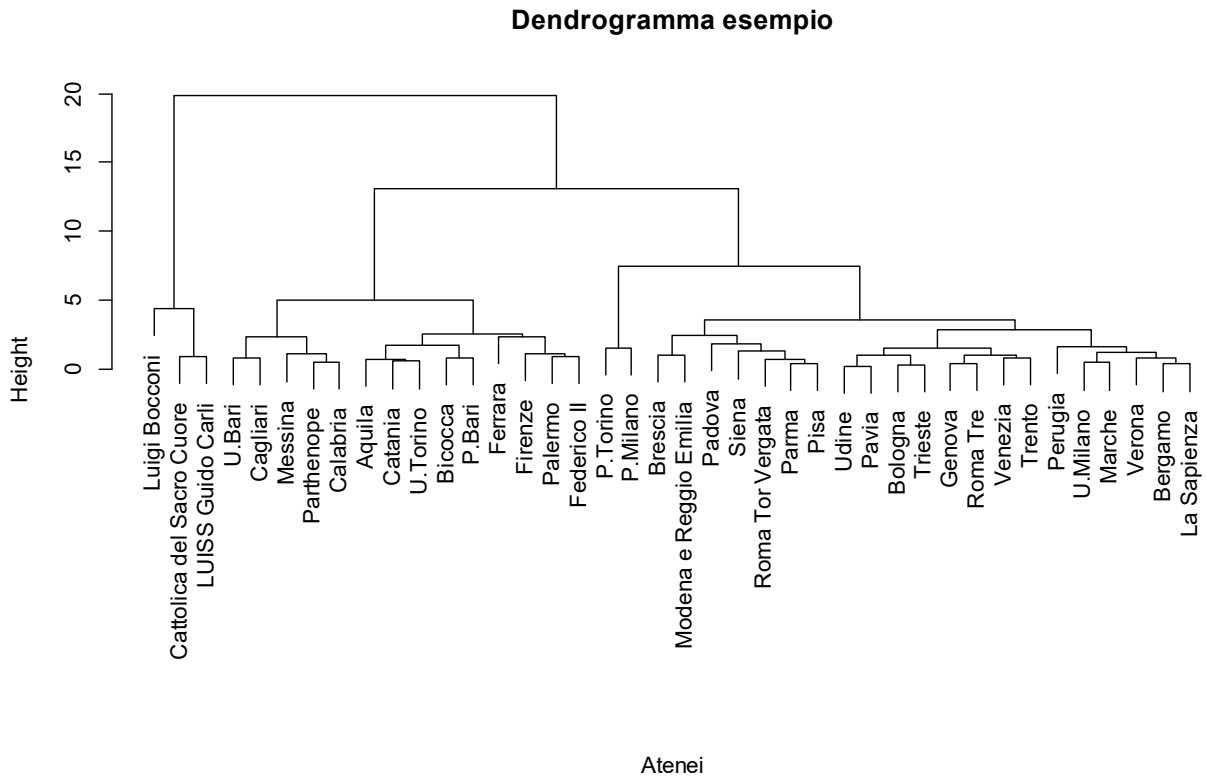
- **single** (legame singolo)
- **complete** (legame completo)
- **average** (legame medio)
- **centroid** (centroide)
- **median** (mediana)

Possiamo suddividere questi metodi in due gruppi, i primi tre **utilizzano la matrice delle distanze**, mentre gli ultimi due **utilizzano la matrice della distanza con gli elementi al quadrato** (non la matrice).

Di default method è posto a "*complete*" come nell'esempio.

Per disegnare il dendrogramma ottenuto invece è necessario passare l'output della funzione `hclust()` (che restituisce una lista di informazioni riguardo l'esecuzione) alla funzione `plot()`, ad esempio:

```
hclust(dist(mtxStipendiLordi2),method="complete")

plot(hclust(dist(mtxStipendiLordi2),method="complete"),
     main ="Dendrogramma esempio",
     xlab="Atenei",sub="")
```



Prima di iniziare a vedere il clustering vero e proprio, introduciamo delle metriche che ci saranno utili per analizzare i nostri risultati.
Prima di tutto vediamo cos'è uno **screeplot** e successivamente vediamo come **analizzare un dendrogramma** in maniera ottimale.

8.7.1.1 Screeplot

Lo screeplot è un **grafico** che rende più semplice la scelta su come partizionare un dendrogramma che si sta analizzando. **Sull'asse delle ordinate sono posti il numero di cluster ottenibili e sull'asse delle ascisse le distanze a cui avvengono le aggregazioni**.

Se nel passaggio **da k a k-1 gruppi la distanza viene incrementata di molto**, allora è consigliabile tagliare il dendrogramma in k gruppi.

Si tratta di un **metodo empirico** ed è consigliabile utilizzare le misure di non omogeneità statistiche (potrebbe suggerire una suddivisione errata).

Costruire uno screeplot è **sconsigliato con il metodo del centroide e della mediana** (vedremo in seguito che usano i **quadrati della distanza**) in quanto le agglomerazioni potrebbero verificarsi a livelli di distanza minore o uguale alle precedenti.

Sono invece **di aiuto col metodo del legame singolo, completo e medio** in cui si utilizza una funzione di distanza.

8.7.1.2 Analisi del dendrogramma

R mette a disposizione varie funzioni per arricchire il dendrogramma e darci una visione più chiara o per aiutarci ad analizzarlo, vediamole:

- La funzione `rect.hclust()` ci dà la possibilità di disegnare dei rettangoli intorno ai cluster individuati in base all'altezza a cui vogliamo operare il taglio, oppure specificando il numero di cluster che vogliamo ottenere.
- La funzione `cutree()` ci consente di ottenere una suddivisione degli individui in cluster in corrispondenza di un livello o di un numero di cluster indicato. Ci restituisce o un vettore o una matrice in cui vengono specificati gli individui in quali cluster sono inseriti.
- La funzione `aggregate()` ci consente di ricavare misure di sintesi (media, varianza, deviazione standard, …) sui singoli cluster che si sono ottenuti

Ci sono poi le misure più importanti e su cui valuteremo effettivamente la bontà del nostro clustering: **misure di non omogeneità statistiche**.

Come già detto infatti la traccia di T è univocamente determinata per ogni matrice che descrive p caratteristiche di n individui, allora fissato un numero m di suddivisioni, i cluster devono essere individuati in modo da **minimizzare** la misura di non omogeneità statistica interna ai cluster, e **massimizzare** la misura di non omogeneità statistica tra i gruppi.

Considereremo vari metodi gerarchici e vedremo se con lo stesso numero di cluster ci conducono a due diverse partizioni: se capita bisogna scegliere quella che presenta la misura di non omogeneità statistica all'interno dei cluster più piccola (trS), cioè puntare ad avere una maggiore omogeneità interna ai cluster.

Dato che nel paragrafo successivo si dovranno calcolare le tracce della matrice di non omogeneità statistica tra i cluster, calcoliamo la **misura di non omogeneità statistica**:

```
> numeroRighe <- nrow(mtxStipendiLordi2)
> trHI <- (numeroRighe-1) *sum(
+   apply(mtxStipendiLordi2, 2, var))
> trHI
[1] 736.5187
```

8.7.1.3 Metodo legame singolo

Il metodo del legame singolo, detto anche nearest neighbour method, individua la distanza tra due cluster come la distanza minima calcolata tra tutte le coppie di individui in cui il primo individuo appartiene al primo cluster, mentre il secondo all'altro cluster preso in considerazione.

- Al livello 0 l'algoritmo considera n cluster, uno per ogni individuo.
- Al passo 1 si cerca la coppia di individui con la distanza minore e si uniscono in un unico cluster. Si modifica poi la matrice delle distanze scegliendo la distanza come la minima tra quella del primo individuo e quella del secondo individuo del nuovo cluster.
- Ad ogni passo dopo che due cluster generici G_u e G_v sono stati uniti scegliendo la coppia di cluster meno distante, la distanza tra il nuovo cluster denotato G_{uv} e un altro cluster G_z è definita scegliendo dalla precedente matrice delle distanze:

$$d_{(uv),z} = \min(d_{uz}, d_{vz})$$

Un vantaggio del metodo del legame singolo è di **consentire di individuare gruppi di qualsiasi forma** e di **evidenziare la presenza di eventuali valori anomali** meglio di altre tecniche, ma ha anche il difetto di basarsi su un singolo legame e non è raro che si possano trovare nello stesso cluster individui piuttosto dissimili: si potrebbero originare delle **catene**.

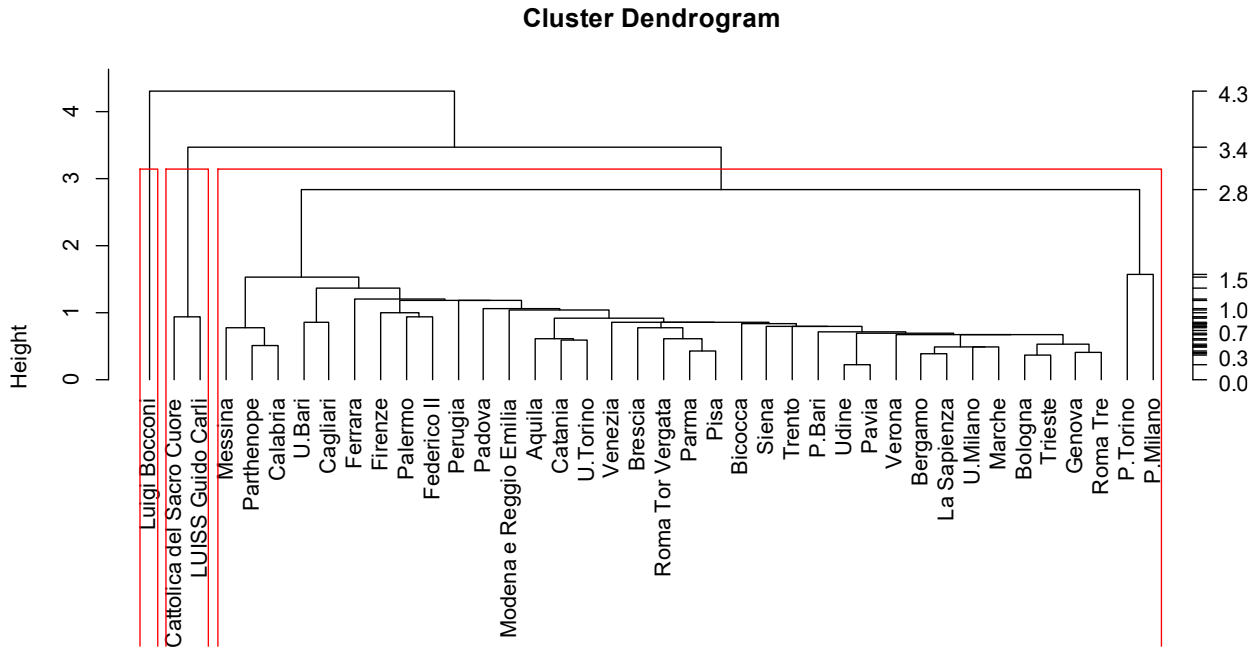
Può capitare che due gruppi ben delineati e distinti vengono inseriti nello stesso gruppo erroneamente; **dunque, non è sempre affidabile il legame singolo**. Effettuiamo il calcolo dei cluster usando il metodo del legame singolo, e vediamo il campo merge dell'output che indica come sono state effettuate le aggregazioni nel corso della computazione:

```
> legameSingolo <- hclust(dist(mtxStipendiLordi2),method="single")
> legameSingolo$merge
      [,1] [,2]
 [1,]  -22  -24
 [2,]  -16  -21
 [3,]  -10  -12
 [4,]  -23  -27
 [5,]  -13  -14
 [6,]  -11  -15
 [7,]    3    6
 [8,]  -39  -40
 [9,]    2    4
[10,]  -25  -28
[11,]  -29   10
[12,]  -18    5
[13,]    7    9
[14,]   -6   13
[15,]    1   14
[16,]  -33   15
[17,]  -38    8
[18,]   -8   12
[19,]  -26   16
[20,]  -20   19
[21,]  -32   20
[22,]   18   21
[23,]  -17   22
```

```
[24,]  -30  -31
[25,]   11   23
[26,]   -1   -2
[27,]  -36  -37
[28,]  -35   27
[29,]   -9   25
[30,]  -19   29
[31,]   -5   30
[32,]   28   31
[33,]  -34   32
[34,]   24   33
[35,]   17   34
[36,]   -4   -7
[37,]   35   36
[38,]   26   37
[39,]   -3   38
```

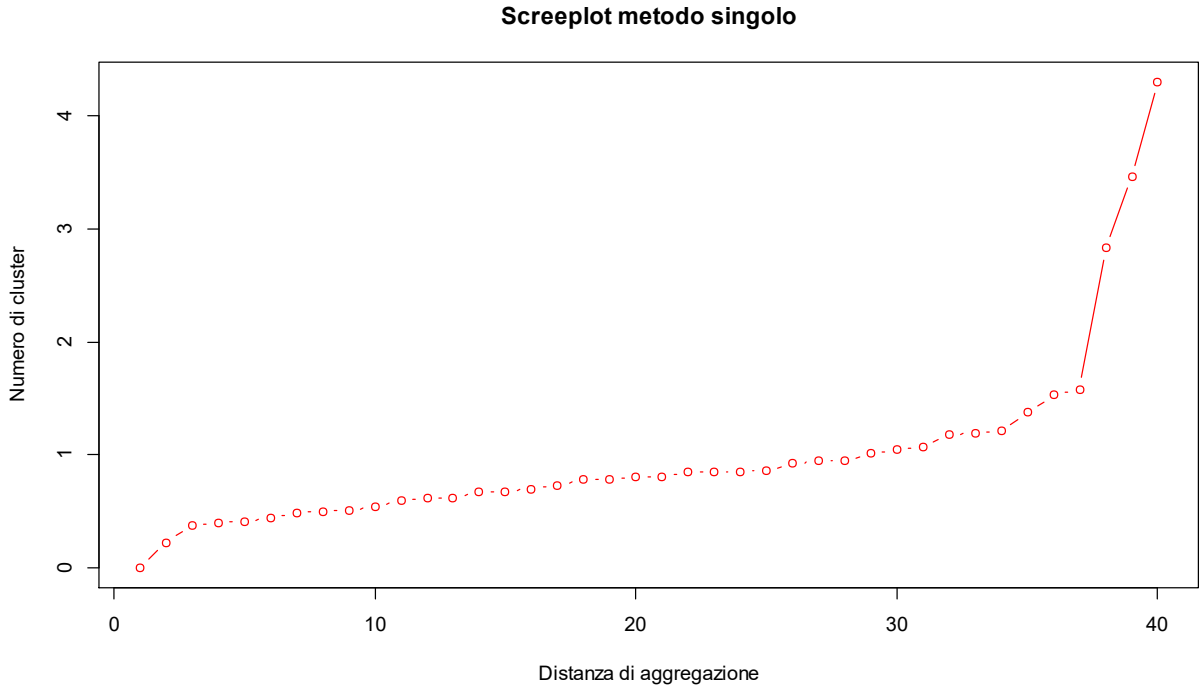
Di seguito il dendrogramma e lo screeplot ottenuti utilizzando il metodo:

```
plot(legameSingolo,hang=-1)
axis(side=4,at=round(c(0,legameSingolo$height),2),las=1)
rect.hclust(legameSingolo,k=3,border="red")
```



Metodo gerarchico agglomerativo
del legame singolo

```
plot(c(0,legameSingolo$height),type="b",col="red")
```



Valutiamo dunque partizionando in 3 gruppi la **misura di non omogeneità statistica tra i cluster**.

```
qualità.taglio<-function(gruppi){
  taglio<-cutree(legameSingolo,k=gruppi,h=NULL)
  num<-table(taglio )
  tagliolist<-list(taglio)
  agvar<-aggregate(mtxStipendiLordi2,tagliolist,var)[,-1]

  trH1<-(num[[1]]-1)*sum(agvar[1,])
  if(is.na(trH1))
    trH1 <- 0

  trH2<-(num[[2]]-1)*sum(agvar[2,])
  if(is.na(trH2))
    trH2<-0

  trH3<-(num [[3]]-1)*sum(agvar [3,])
  if(is.na(trH3))
    trH3<-0

  sum<-trH1+trH2+trH3
  trB<-trHI-sum
  trB/trHI
}
```

```
> qualità.taglio(3)
[1] 0.5523372
```

Vediamo che con **3** cluster il risultato è decisamente migliore a quello precedente e ci avviciniamo ad una buona suddivisione.

Infine, sappiamo che il **legame singolo si fa influenzare da valori anomali** e sappiamo che ce ne sono ben tre nel nostro dataset.

8.7.1.4 Metodo legame completo

Il metodo del **legame completo**, detto anche furthest neighbour method, individua la distanza tra due cluster come la **distanza massima calcolata tra tutte le coppie** di individui in cui il primo individuo appartiene al primo cluster, mentre il secondo all'altro cluster preso in considerazione.

- Al livello 0 l'algoritmo considera n cluster, uno per ogni individuo.
- Al passo 1 si cerca la coppia di individui con la distanza minore e si uniscono in un unico cluster. Si modifica poi la matrice delle distanze scegliendo la distanza con gli altri cluster individuata come la maggiore tra quella del primo individuo e quella del secondo individuo del nuovo cluster.
- Ad ogni passo dopo che due cluster generici G_u e G_v sono stati uniti scegliendo la coppia di cluster meno distante, la distanza tra il nuovo cluster denotato G_{uv} e un altro cluster G_z è definita scegliendo dalla precedente matrice delle distanze:

$$d_{(uv),z} = \max(d_{uz}, d_{vz})$$

Questo metodo è adatto per gruppi che si **addensono intorno a un elemento centrale**. Viene privilegiata **l'omogeneità dei gruppi** e si **evita l'effetto catena**. Si nota inoltre che il dendrogramma costruito con questo metodo ha **rami più lunghi** poiché le distanze sono maggiori. In linea generale il metodo del legame completo è ottimo da utilizzare.

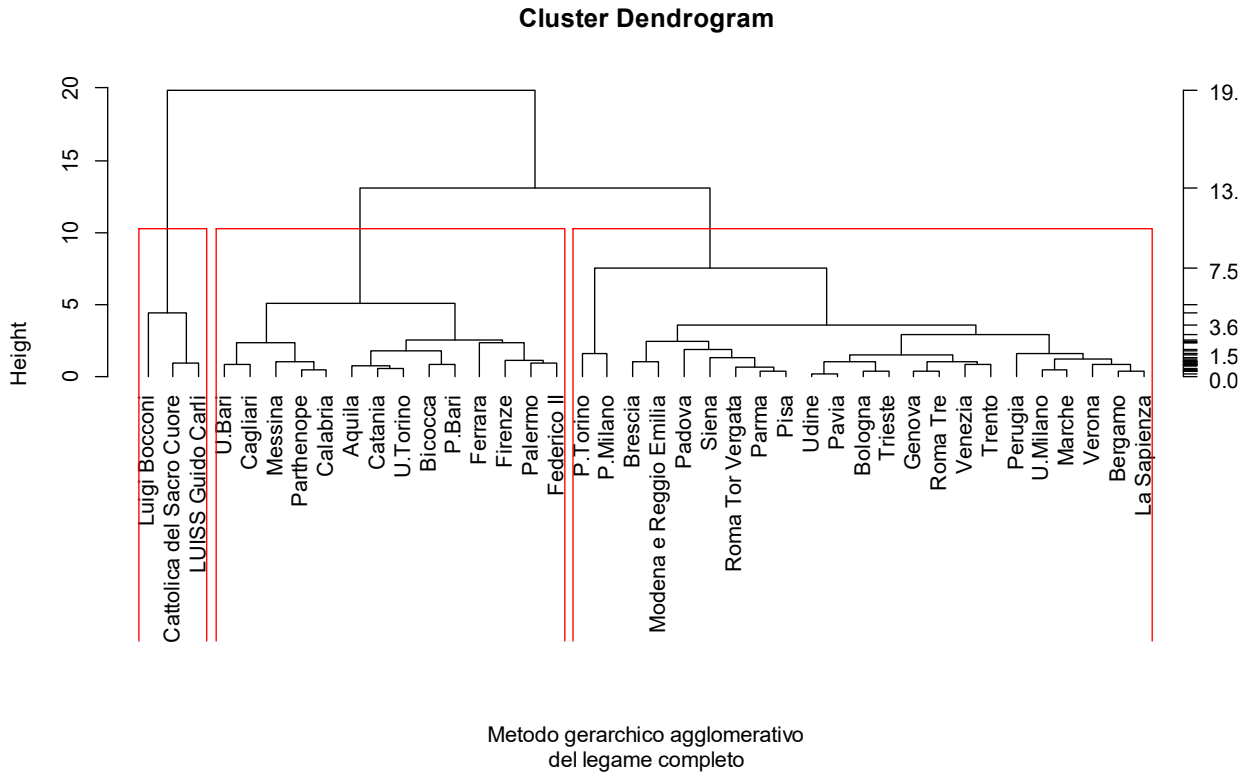
Effettuiamo il calcolo dei cluster usando il metodo del legame completo, e vediamo il campo merge dell'output che indica come sono state effettuate le aggregazioni nel corso della computazione:

```
legameCompleto <- hclust(dist(mtxStipendiLordi2), method="complete")
> legameCompleto$merge
      [,1] [,2]
 [1,]  -22  -24
 [2,]  -16  -21
 [3,]  -10  -12
 [4,]  -23  -27
 [5,]  -13  -14
 [6,]  -11  -15
 [7,]  -39  -40
 [8,]  -25  -28
 [9,]  -18    5
[10,]  -29    8
[11,]   -6    3
[12,]  -32  -33
[13,]  -17  -26
[14,]  -30  -31
[15,]   -1   -2
[16,]  -36  -37
[17,]    1    2
[18,]   -8   -9
[19,]    4   13
[20,]  -38    7
[21,]  -35   16
[22,]    6   11
```

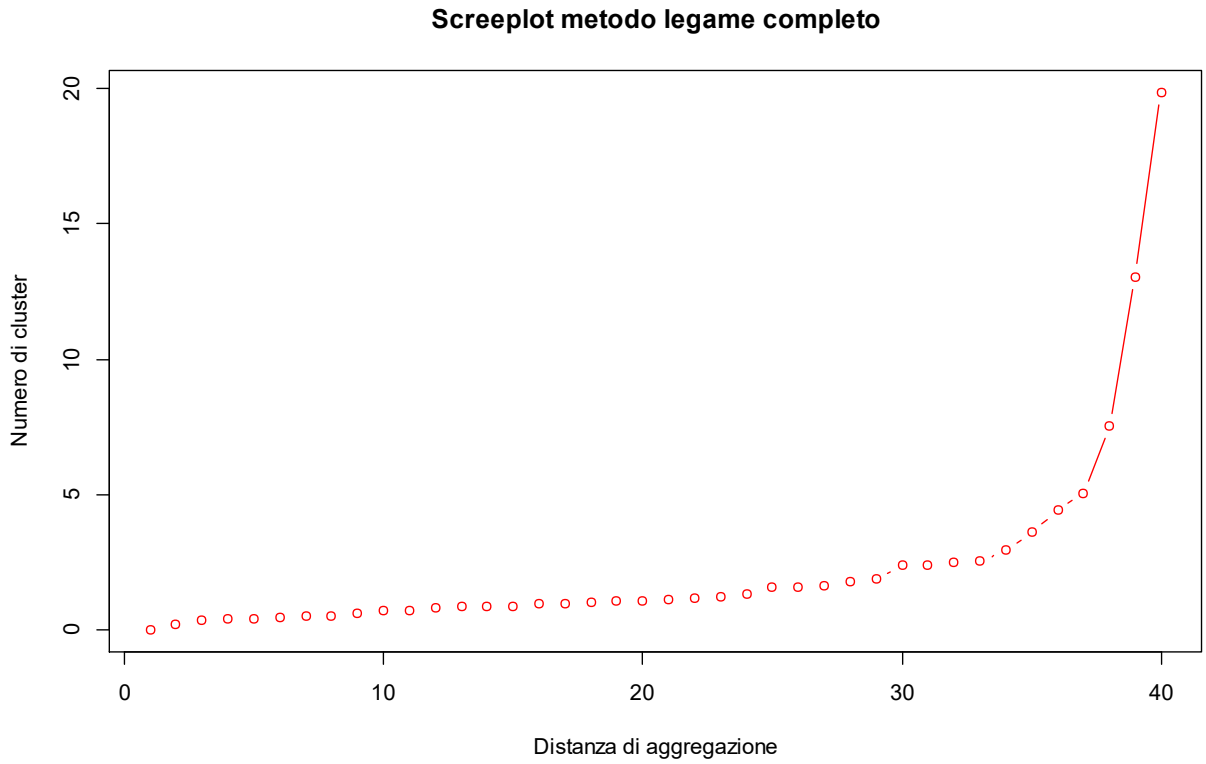
```
      [23,]  -20    9
      [24,]   17   19
      [25,]   -4   -7
      [26,]   -5   22
      [27,]   10   12
      [28,]  -19   23
      [29,]  -34   21
      [30,]   14   20
      [31,]   18   28
      [32,]   27   29
      [33,]   24   26
      [34,]   31   33
      [35,]   -3   15
      [36,]   30   32
      [37,]   25   34
      [38,]   36   37
      [39,]   35   38
```

Di seguito il dendrogramma e lo screeplot ottenuti utilizzando il metodo:

```
plot(legameCompleto,hang=-1, xlab="Metodo gerarchico agglomerativo",
     sub="del legame completo")
axis(side=4,at=round(c(0,legameCompleto$height),2),las=1)
rect.hclust(legameSingolo,k=3,border="red")
```




```
plot(c(0,legameCompleto$height),type="b",col="red")
```



Valutiamo dunque partizionando in 3 gruppi la **misura di non omogeneità statistica tra i cluster**.

```
qualità.taglio<-function(gruppi){
  taglio<-cutree(legameCompleto,k=gruppi,h=NULL)
  num<-table(taglio )
  tagliolist<-list(taglio)
  agvar<-aggregate(mtxStipendiLordi2,tagliolist,var)[,-1]

  trH1<-(num[[1]]-1)*sum(agvar[1,])
  if(is.na(trH1))
    trH1 <- 0

  trH2<-(num[[2]]-1)*sum(agvar[2,])
  if(is.na(trH2))
    trH2<-0

  trH3<-(num [[3]]-1)*sum(agvar [3,])
  if(is.na(trH3))
    trH3<-0

  sum<-trH1+trH2+trH3
  trB<-trHI-sum
  trB/trHI
}
```

```
> qualità.taglio(3)
[1] 0.8036132
```

Vediamo che con 4 cluster raggiungiamo già un **risultato ottimo**.

8.7.1.5 Metodo legame medio

Il metodo del **legame medio**, detto anche *average linkage method*, individua la distanza tra due cluster come la **media aritmetica delle distanze tra tutte le coppie di individui che compongono due gruppi**.

- Al livello 0 l'algoritmo considera n cluster, uno per ogni individuo.
- Al passo 1 si cerca la coppia di individui con la distanza minore e si uniscono in un unico
- cluster. Si modifica poi la matrice delle distanze scegliendo la distanza con gli altri cluster:
- viene individuata come la media calcolata tra i nuovi elementi del cluster (unione degli
- elementi dei due cluster aggregati) con gli altri insiemi già presenti.
- Ad ogni passo dopo che due cluster generici G_u e G_v sono stati uniti scegliendo la coppia di cluster meno distante, la distanza tra il nuovo cluster denotato G_{uv} e un altro cluster G_z è definita effettuando il calcolo di seguito in base alla matrice delle distanze precedente:

$$d_{(uv),z} = \frac{N_u}{N_u + N_v} d_{uz} + \frac{N_v}{N_u + N_v} d_{vz}$$

Dove N_u e N_v sono individui nel cluster G_u e G_v. $d_{(uv),z}$ rappresenta la misura di **distanza media tra gli elementi dei cluster** G_{uv} e G_z. La procedura si ripete fino ad ottenere un unico cluster formato da tutti gli individui.

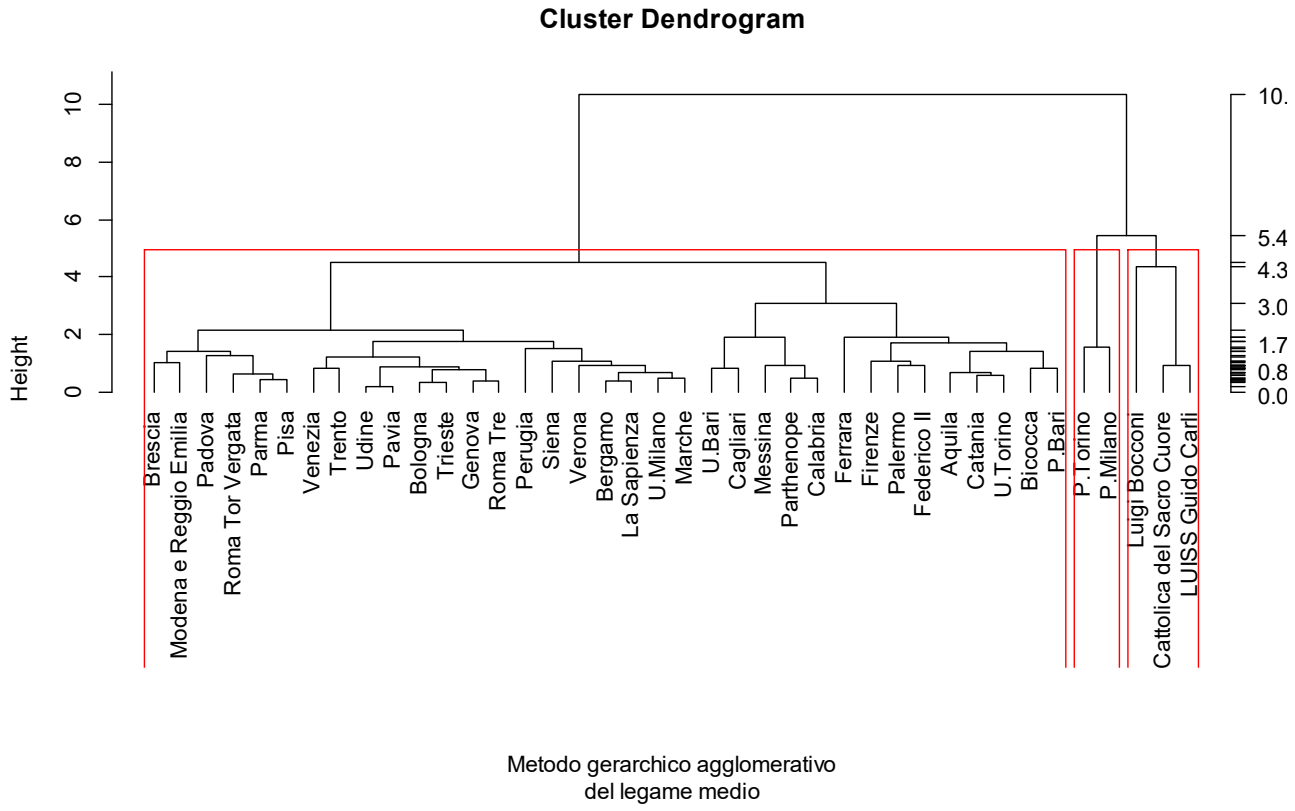
Questo metodo ha il seguente svantaggio: se il numero di elementi dei due cluster che si uniscono è molto diverso, la **distanza sarà più vicina a quella del cluster più numeroso**.

Effettuiamo il calcolo dei cluster usando il metodo del legame medio, e vediamo il campo merge dell'output che indica come sono state effettuate le aggregazioni nel corso della computazione:

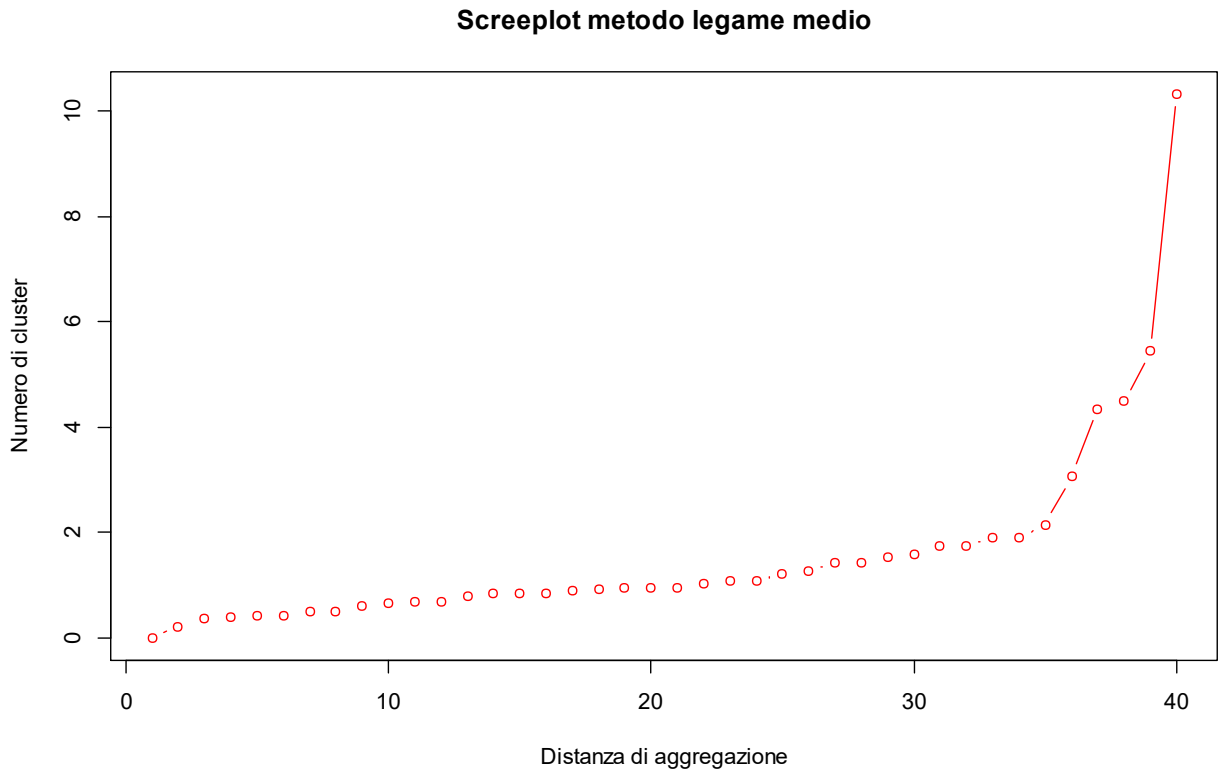
```
> legameMedio <- hclust(dist(mtxStipendiLordi2), method="average")
> legameMedio$merge
      [,1] [,2]
 [1,]  -22  -24
 [2,]  -16  -21
 [3,]  -10  -12
 [4,]  -23  -27
 [5,]  -13  -14
 [6,]  -11  -15
 [7,]  -39  -40
 [8,]  -25  -28
 [9,]  -18    5
[10,]  -29    8
[11,]    3    6
[12,]    2    4
[13,]  -32  -33
[14,]  -17  -26
[15,]  -30  -31
[16,]    1   12
[17,]   -6   11
[18,]  -38    7
[19,]   -1   -2
[20,]  -36  -37
```

```
      [21,]   -8   -9
      [22,]  -35   20
      [23,]  -20   17
      [24,]   14   16
      [25,]  -19    9
      [26,]   21   25
      [27,]   10   13
      [28,]   -5   23
      [29,]   -4   -7
      [30,]   22   27
      [31,]   24   28
      [32,]   15   18
      [33,]  -34   30
      [34,]   26   31
      [35,]   32   33
      [36,]   -3   19
      [37,]   34   35
      [38,]   29   36
      [39,]   37   38
```

Di seguito il dendrogramma e lo screeplot ottenuti utilizzando il metodo:



Metodo gerarchico agglomerativo
del legame medio



Valutiamo dunque partizionando in 3 gruppi la **misura di non omogeneità statistica tra i cluster**.

```
qualità.taglio<-function(gruppi){
  taglio<-cutree(legameMedio,k=gruppi,h=NULL)
  num<-table(taglio )
  tagliolist<-list(taglio)
  agvar<-aggregate(mtxStipendiLordi2,tagliolist,var)[,-1]

  trH1<-(num[[1]]-1)*sum(agvar[1,])
  if(is.na(trH1))
    trH1 <- 0

  trH2<-(num[[2]]-1)*sum(agvar[2,])
  if(is.na(trH2))
    trH2<-0

  trH3<-(num [[3]]-1)*sum(agvar [3,])
  if(is.na(trH3))
    trH3<-0

  sum<-trH1+trH2+trH3
  trB<-trHI-sum
  trB/trHI
}
```

```
> qualità.taglio(3)
[1] 0.6666737
```

8.7.1.6 Metodo del centroide

Il metodo del **centroide** individua la distanza tra due gruppi come la distanza tra i **centroidi**, la **distanza tra le medie campionarie calcolate sugli individui appartenenti ai due gruppi**.

Per questo metodo viene usata la matrice che contiene i **quadrati delle singole distanze euclidee**.

Questo metodo può portare gruppi di grandi dimensioni a portare dentro di sé piccoli gruppi (**fenomeni gravitazionali**).

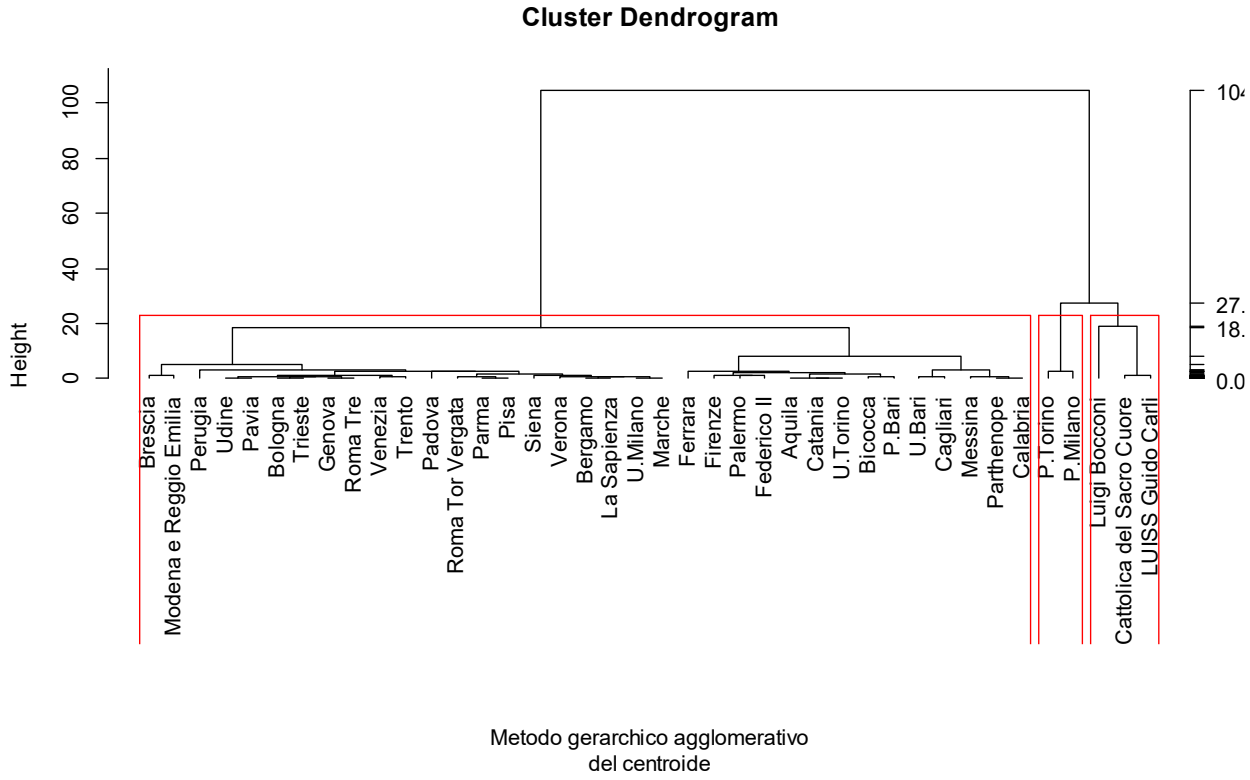
Analogamente a prima se uno dei due gruppi uniti ha una numerosità maggiore all'altro, allora il centroide risultante sarà molto vicino a quello del cluster più numeroso.

Calcoliamo dunque la matrice dei quadrati delle singole distanze euclidee che ci servirà per applicare il metodo ed effettuiamo il calcolo dei cluster usando il metodo del centroide, e vediamo il campo *merge* dell'output che indica come sono state effettuate le aggregazioni nel corso della computazione:

```
> legameCentroid <- hclust(dist(mtxStipendiLordi2)^2, method="centroid")
> legameCentroid$merge
      [,1] [,2]
 [1,]  -22  -24
 [2,]  -16  -21
 [3,]  -10  -12
 [4,]  -23  -27
 [5,]  -13  -14
 [6,]  -11  -15
 [7,]  -39  -40
 [8,]  -25  -28
 [9,]  -29    8
[10,]  -18    5
[11,]    3    6
[12,]    2    4
[13,]    1   12
[14,]  -32  -33
[15,]  -17  -26
[16,]   -6   11
[17,]  -30  -31
[18,]  -38    7
[19,]   -1   -2
[20,]  -36  -37
[21,]  -35   20
[22,]  -20   16
[23,]   13   15
[24,]   -8   -9
[25,]   10   22
[26,]    9   14
[27,]   21   26
[28,]   -4   -7
[29,]  -19   25
[30,]   23   29
[31,]  -34   27
[32,]   -5   30
[33,]   17   18
[34,]   24   32
[35,]   31   33
[36,]   34   35
[37,]   -3   19
[38,]   28   37
```

```
     [39,]   36   38
```

In questo caso calcoliamo solo il dendrogramma visto che lo screeplot non è molto utile per questo metodo (e il successivo che vedremo).
Di seguito il dendrogramma dunque:



Metodo gerarchico agglomerativo
del centroide

Valutiamo dunque partizionando in 3 gruppi la **misura di non omogeneità statistica tra i cluster**.

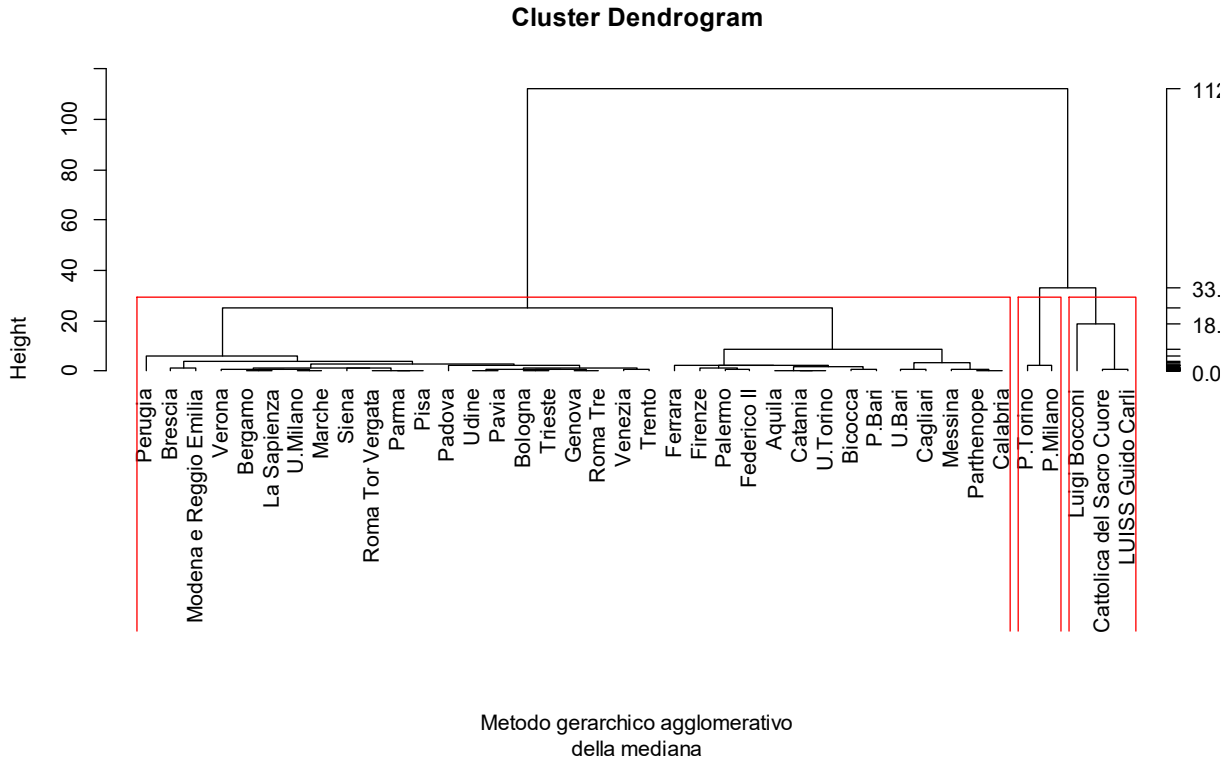
```
> qualità.taglio(3)
[1] 0.6666737
```

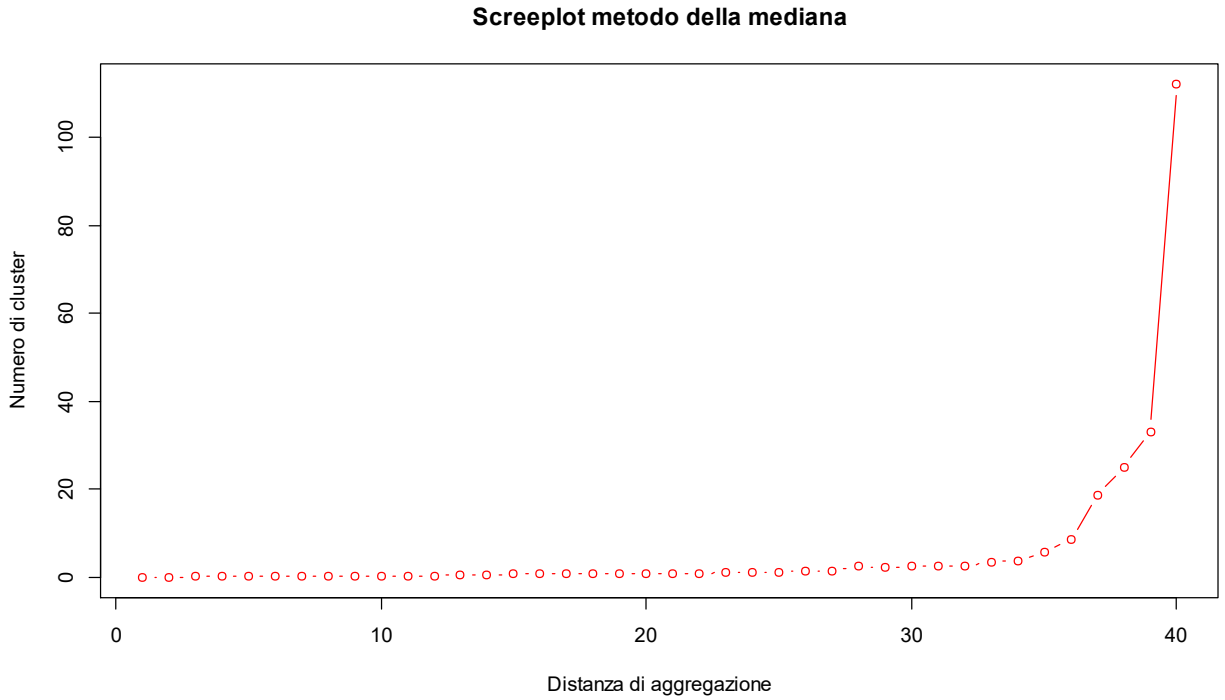
8.7.1.7 Metodo della mediana

Il metodo della mediana è simile a quello del centroide, con la differenza che la procedura è indipendente dalla numerosità dei cluster. Infatti, quando due gruppi si aggregano, il nuovo centroide è calcolato come la semisomma dei due centroidi precedenti.

```
> legameMediana <- hclust(dist(mtxStipendiLordi2)^2, method="median")
> legameMediana$merge
      [,1] [,2]
 [1,]  -22  -24
 [2,]  -16  -21
 [3,]  -10  -12
 [4,]  -23  -27
 [5,]  -13  -14
 [6,]  -11  -15
 [7,]  -39  -40
 [8,]  -25  -28
 [9,]  -29    8
[10,]  -18    5
[11,]    3    6
[12,]    2    4
[13,]    1   12
[14,]  -32  -33
[15,]  -17  -26
[16,]   -6   11
[17,]  -30  -31
[18,]  -38    7
[19,]   -1   -2
[20,]  -36  -37
[21,]  -35   20
[22,]  -20   10
[23,]   -8   -9
[24,]   13   15
[25,]   16   22
[26,]    9   14
[27,]   21   26
[28,]  -34   27
[29,]   -4   -7
[30,]  -19   24
[31,]   25   30
[32,]   17   18
[33,]   23   31
[34,]   -5   33
[35,]   28   32
[36,]   -3   19
[37,]   34   35
[38,]   29   36
[39,]   37   38
```

Di seguito il dendrogramma e lo screeplot ottenuti utilizzando il metodo:





Valutiamo dunque partizionando in 3, 4 o 5 gruppi come cambia la **misura di non omogeneità statistica tra i cluster**.

```
> qualità.taglio(3)
[1] 0.657965
```

8.7.1.8 In conclusione

In conclusione, da questa analisi abbiamo individuato che il **clustering migliore lo otteniamo utilizzando il metodo del legame completo con 3 cluster**.

Metodi	Valore
Singolo	0.5523372
Completo	0.8036132
Medio	0.6666737
Centroide	0.6666737
Mediana	0.657965

```
> cutree(legameCompleto, k=3, h=NULL)
Cattolica del Sacro Cuore         LUISS Guido Carli             Luigi Bocconi
                        1                         1                         1
                P.Torino                   Perugia                    Verona
                        2                         2                         2
                P.Milano                   Brescia  Modena e Reggio Emilia
                        2                         2                         2
                 Bergamo                  U.Milano             La Sapienza
                        2                         2                         2
                   Parma                     Pisa                   Marche
                        2                         2                         2
                 Bologna                   Venezia        Roma Tor Vergata
                        2                         2                         2
                  Padova                     Siena                  Trieste
                        2                         2                         2
                   Udine                    Genova                    Pavia
                        2                         2                         2
                 Catania                    Trento                 Roma Tre
                        3                         2                         2
                U.Torino                    Aquila                   U.Bari
                        3                         3                         3
                Cagliari                   Bicocca                   P.Bari
                        3                         3                         3
                 Ferrara                   Firenze                  Palermo
                        3                         3                         3
            Federico II                    Messina                Parthenope
                        3                         3                         3
                Calabria
                        3
```

9. *CONCLUSIONE*

In conclusione, dall'indagine statistica che è stata presentata, si è potuto ottenere e ricavare informazioni riguardanti un'analisi più approfondita sullo stipendio della popolazione italiana che ha conseguito la laurea nei più importanti atenei italiani rispetto all'età.
Infatti è stato molto interessante ampliare queste analisi poiché ha portato a conoscenza come gli atenei del Nord Italia, come ad esempio Luigi Bocconi, LUISS Guido Carli e Cattolica del Sacro Cuore, ripaghino con uno stipendio nettamente migliore rispetto agli atenei del Sud Italia.