Università degli Studi di Salerno

Dipartimento di Informatica
Corso di Laurea Magistrale in Informatica

Progetto di Statistica e Analisi dei Dati

Capacità delle famiglie italiane di arrivare a fine mese

Costante Marco 0522501330

Anno Accademico 2021-2022

Indice

Capitolo 1

Introduzione

Lo scopo di questo progetto è fornire un'analisi statistica dettagliata dei dati relativi alla **capacità delle famiglie italiane di arrivare a fine mese**, al fine di avere una visione chiara e dettagliata del fenomeno nella sua interezza.
Il problema della povertà è una realtà sempre più vicina. Nonostante ci si sia abituati a credere che quelli della povertà e della difficoltà ad arrivare a fine mese siano problemi che non riguardano i paesi occidentali, i dati mostrano altro.
Le famiglie che nel 2019 si trovano a vivere situazioni di forte difficoltà economiche non sono poche, e i dati ottenuti attraverso questo lavoro confermano che il problema esiste anche in Italia.
Obiettivo di questo lavoro è analizzare e approfondire il problema al fine di porre in essere riflessioni e soluzioni.

1.1 Descrizione del dataset

Il dataset, mostrato in figura 1.1, è stato ottenuto attraverso il sito ufficiale dell'**Istituto Nazionale di Statistica**.
Esso mostra i dati, relativi all'**anno 2019**, raccolti su una vasta gamma di cittadini residenti nelle varie regioni italiane, al fine di ottenere il grado di difficoltà economica di queste ultime.
Per ciascuna regione è indicata la **percentuale** di:

- Famiglie con grandi difficoltà economiche;
- Famiglie con difficoltà;
- Famiglie con qualche difficoltà, ma comunque con una certa facilità;
- Famiglie che arrivano facilmente o molto facilmente a fine mese,

Dataset:Famiglie per capacità di arrivare a fine mese

Tipo dato	famiglie che arrivano e che non arrivano a fine mese (composizione percentuale)			
Seleziona periodo	2019			
Giudizio sulla condizione economica percepita	con grande difficoltà	con difficoltà	con qualche difficoltà e con una certa facilità	con facilità e con molta facilità
Territorio				
Piemonte	5,0	12,7	69,3	13,0
Liguria	4,6	8,7	75,8	10,9
Lombardia	4,8	10,0	72,2	13,0
Trentino Alto-Adige	9,0	18,9	61,6	10,5
Veneto	2,9	7,2	78,7	11,2
Friuli-Venezia Giulia	3,4	12,1	71,6	12,9
Emilia-Romagna	3,6	9,1	79,6	7,7
Toscana	3,6	8,9	76,3	11,3
Umbria	2,5	9,0	80,7	7,8
Marche	3,0	14,6	73,2	9,2
Lazio	8,9	22,3	62,0	6,9
Abruzzo	11,7	12,3	66,8	9,2
Molise	12,4	20,8	59,6	7,1
Campania	24,9	28,5	42,4	4,2
Puglia	10,5	19,0	64,4	6,1
Basilicata	8,8	12,7	71,6	7,0
Calabria	8,2	13,0	75,9	2,8
Sicilia	11,6	18,1	63,6	6,7
Sardegna	12,0	26,2	54,9	6,8

Figura 1.1: DataSet di partenza.

Il progetto che sarà descritto nei capitoli successivi è stato realizzato mediante l'uso del linguaggio di programmazione e ambiente integrato **R**.

1.1.1 Definizione della matrice dei dati

Per poter utilizzare la vasta gamma di funzionalità offerte da R, è necessario inglobare i dati descritti precedentemente in una struttura apposita. In particolare è stata costruita una matrice attraverso le seguenti linee di codice.
Prima di tutto costruiamo i vettori che andranno a rappresentare le colonne della nostra matrice:

```
#INIZIALIZZAZIONE DEI VETTORI COLONNA

grande_difficolta<-c(5.0, 4.6, 4.8, 9.0, 2.9, 3.4, 3.6, 3.6, 2.5, 3.0,
8.9, 11.7, 12.4, 24.9, 10.5, 8.8, 8.2, 11.6, 12.0)

media_difficolta<-c(12.7, 8.7, 10.0, 18.9, 7.2, 12.1, 9.1, 8.9, 9.0, 14.6,
22.3, 12.3, 20.8, 28.5, 19.0, 12.7, 13.0, 18.1, 26.2)

poca_difficolta<-c(69.3, 75.8, 72.2, 61.6, 78.7, 71.6, 79.6, 76.3, 80.7, 73.2,
62.0, 66.8, 59.6, 42.4, 64.4, 71.6, 75.9, 63.6, 54.9)

molta_facilita<-c(13.0, 10.9, 13.0, 10.5, 11.2, 12.9, 7.7, 11.3, 7.8, 9.2,
6.9, 9.2, 7.1, 4.2, 6.1, 7.0, 2.8, 6.7, 6.8)
```

Per favorire la leggibilità del dataset inizializziamo anche due vettori per rappresentare rispettivamente i nomi delle righe (ragioni) e i nomi delle colonne.

```
#INIZIALIZZAZIONE DEL VETTORE PER I NOMI DELLE REGIONI

regioni<-c("Piemonte", "Liguria", "Lombardia", "Trentino Alto-Adige", "Veneto", "Friuli-Venezia Giulia",
"Emilia-Romagna", "Toscana", "Umbria", "Marche", "Lazio", "Abruzzo", "Molise", "Campania",
"Puglia", "Basilicata", "Calabria", "Sicilia", "Sardegna")

#INIZIALIZZAZIONE DEL VETTORE PER I NOMI DELLE COLONNE

colonne<-c("Grande difficolta'", "Media difficolta'", "Poche difficolta'", "Molta facilita'")
```

A questo punto è possibile finalmente costruire la matrice che verrà utilizzata nel corso di questa analisi statistica.

```
#COSTRUZIONE MATRICE

matrice_capacita_arrivare_fine_mese<-cbind(grande_difficolta, media_
    difficolta,
poca_difficolta, molta_facilita)

rownames(matrice_capacita_arrivare_fine_mese)<-regioni
colnames(matrice_capacita_arrivare_fine_mese)<-colonne
```

Ottenendo come matrice il risultato in figura 1.2.

Il linguaggio R mette a disposizione il comando **c()** per costruire un vettore, il comando **cbind()** per creare colonne di matrici ottenute componendo vettori di uguale lunghezza e i comandi **rownames()** e **colnames()** per modificare le intestazioni delle righe e delle colonne di una matrice.

	Grande difficoltà	Media difficoltà	Poche difficoltà	Molta facilità
Piemonte	5.0	12.7	69.3	13.0
Liguria	4.6	8.7	75.8	10.9
Lombardia	4.8	10.0	72.2	13.0
Trentino Alto-Adige	9.0	18.9	61.6	10.5
Veneto	2.9	7.2	78.7	11.2
Friuli-Venezia Giulia	3.4	12.1	71.6	12.9
Emilia-Romagna	3.6	9.1	79.6	7.7
Toscana	3.6	8.9	76.3	11.3
Umbria	2.5	9.0	80.7	7.8
Marche	3.0	14.6	73.2	9.2
Lazio	8.9	22.3	62.0	6.9
Abruzzo	11.7	12.3	66.8	9.2
Molise	12.4	20.8	59.6	7.1
Campania	24.9	28.5	42.4	4.2
Puglia	10.5	19.0	64.4	6.1
Basilicata	8.8	12.7	71.6	7.0
Calabria	8.2	13.0	75.9	2.8
Sicilia	11.6	18.1	63.6	6.7
Sardegna	12.0	26.2	54.9	6.8

Figura 1.2: Matrice di partenza.

Capitolo 2

Rappresentazioni grafiche dei dati

Nell'ambito della statistica e dell'analisi dei dati, l'uso di rappresentazioni grafiche è un ottimo **strumento di presentazione** che affianca quella tabellare e favorisce la comprensione del fenomeno statistico in esame.
Questo capitolo mira a presentare il fenomeno attraverso le più note rappresentazioni grafiche offerte dallo strumento R.

2.1 Grafici a barre

Consideriamo una variabile X e indichiamo con $z_1, ..., z_k$ le modalità distinte da essa assunte. Consideriamo poi un campione $x = (x_1, ..., x_n)$ costituito da n osservazioni di X. Disponiamo sull'asse orizzontale le modalità assunte da X e sull'asse verticale riportiamo le frequenze assolute o le frequenze relative. Tracciamo dei rettangoli centrati sulle modalità z_i tutti della stessa base e altezza pari alle frequenze, ottenendo un grafico a barre.

In R usiamo il comando **barplot()** per generare grafici di questo tipo.

In questa sezione verranno analizzati i dati presenti nella matrice definita nel primo capitolo attraverso l'uso di grafici a barre. In particolare per ogni colonna della matrice verrà definito il rispettivo grafico. Ogni grafico avrà sull'asse delle X le varie regioni del dataset, sull'asse delle Y le percentuali per ogni categoria.

2.1.1 Percentuale di famiglie con grandi difficoltà

```
#GRAFICO A BARRE PERCENTUALE DI PERSONE CON GRANDI DIFFICOLTA'

barplot(matrice_capacita_arrivare_fine_mese[,1], col=1:19, main="
    Percentuale di famiglie con grandi difficolt  ad arrivare a fine
     mese",
names.arg = etichette_regioni)
```

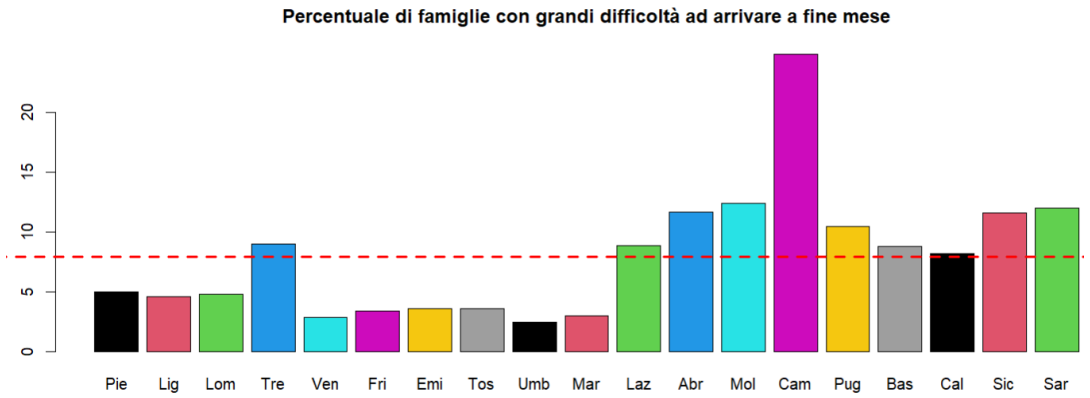


Figura 2.1: Grafico a barre per percentuale di famiglie con grande difficoltà.

Da questo grafico notiamo immediatamente che la più alta percentuale di famiglie che hanno gravi difficoltà ad arrivare a fine mese sono concentrate nella **Campania**. Si nota inoltre come in generale la più alta concentrazione di famiglie a rischio siano concentrate nel **sud-Italia**.

2.1.2 Percentuale di famiglie con media difficoltà

```
#GRAFICO A BARRE PERCENTUALE DI PERSONE CON MEDIA DIFFICOLTA'

barplot(matrice_capacita_arrivare_fine_mese[,2], col=1:19, main="
    Percentuale di famiglie con media difficolt  ad arrivare a fine
    mese",
names.arg = etichette_regioni)
```

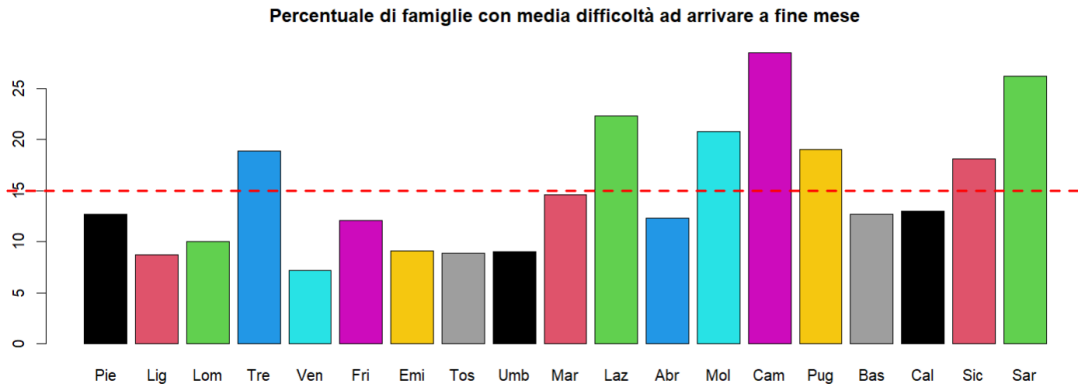


Figura 2.2: Grafico a barre per percentuale di famiglie con media difficoltà.

Anche in questo caso notiamo come vi sia la tendenza a percentuali più alte in relazione alle regioni del **sud-Italia**, in particolare della **Campania** e della **Sardegna**.

2.1.3 Percentuale di famiglie con poche difficoltà

```
#GRAFICO A BARRE PERCENTUALE DI PERSONE CON POCHE DIFFICOLTA'

barplot(matrice_capacita_arrivare_fine_mese[,3], col=1:19, main="
    Percentuale di famiglie con poche difficolt  ad arrivare a fine
    mese",
names.arg = etichette_regioni)
```

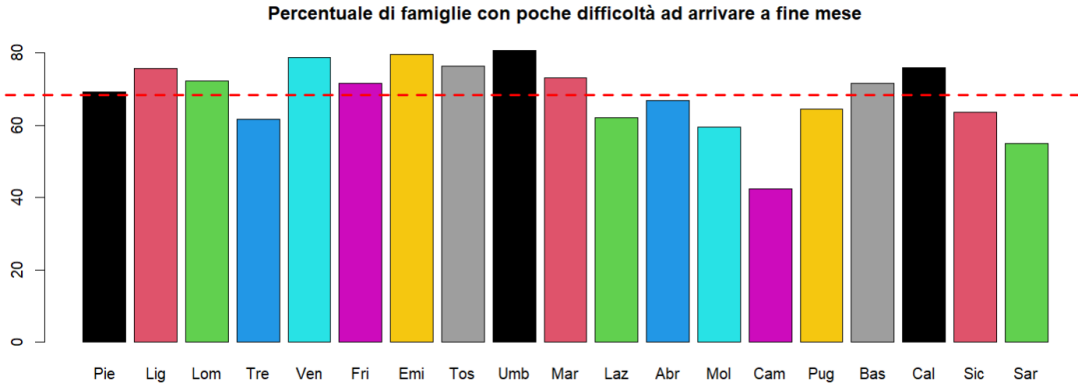


Figura 2.3: Grafico a barre per percentuale di famiglie con poche difficoltà.

Si nota in questo caso 2.3 come il grafico a barre tenda ad appiattarsi a dimostrazione del fatto che la percentuale di famiglie con poche difficoltà ad arrivare a fine mese sia distribuita nelle varie regioni. Anche in questo caso, come ci si potrebbe aspettare, notiamo barre più corte della **Campania** e della **Sardegna**.

2.1.4 Percentuale di famiglie senza difficoltà

```
#GRAFICO A BARRE PERCENTUALE DI PERSONE SENZA DIFFICOLT /MOLTA
    FACILITA'

barplot(matrice_capacita_arrivare_fine_mese[,4], col=1:19, main="
    Percentuale di famiglie senza difficolt  ad arrivare a fine mese
    ",
names.arg = etichette_regioni)
```

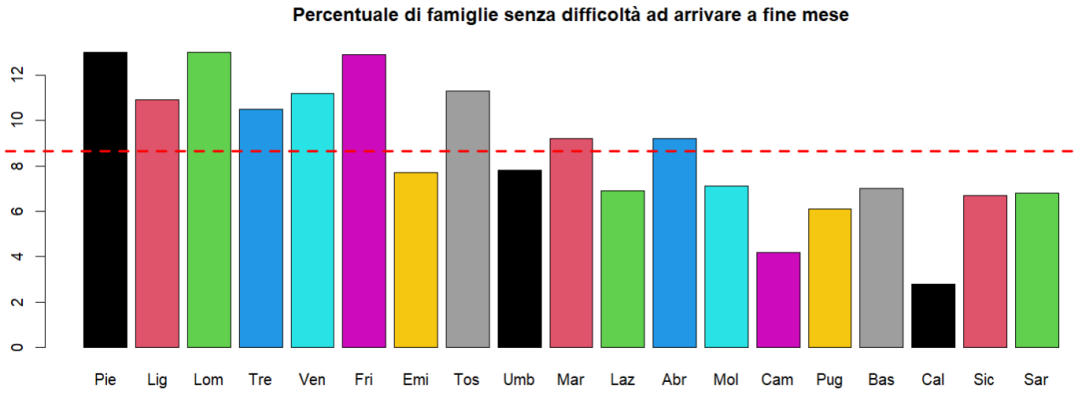


Figura 2.4: Grafico a barre per percentuale di famiglie senza difficoltà.

Notiamo immediatamente che in questo caso si ottenga un grafico quasi speculare al primo. Infatti nelle regioni del **nord-Italia** si hanno percentuali decisamente più alte di famiglie che non hanno difficoltà o addirittura arrivano a fine mese con molta facilità. In particolare **Piemonte**, **Lombardia** e **Friuli-Venezia-Giulia** risultano essere quelle con l'andamento migliore.

2.2 Grafici a torta

I diagrammi a torta permettono di attribuire ciascuna modalità della variabile qualitativa in esame ad un settore circolare di un cerchio, la cui ampiezza è proporzionale alle frequenze.

Analizziamo, attraverso un grafico a torta la situazione generale in **Italia**, in relazione al fenomeno in esame.
R offre il comando **pie()** per realizzare grafici a torta.
Ricordiamo che le percentuali in Italia, ottenute dal dataset originale, risultano essere:

- Famiglie con grande difficoltà: 7.9;
- Famiglie con media difficoltà: 14.6;
- Famiglie con poca difficoltà: 68.4;
- Famiglie senza difficoltà: 9.2;

```
1 #COSTRUIAMO IL VETTORE ITALIA
2
3 italia<-c(rep("Grande difficolta", 7.9), rep("Media difficolta",
      14.6),
4 rep("Poca difficolta", 68.4), rep("Senza difficolt /molta facilita",
       9.2))
5
6 #COSTRUZIONE GRAFICO A TORTA
7
8 pie(table(italia), col=1:4, main="Percentuale italiana della
      capacit  delle famgilie di arrivare a fine mese")
```

Dal grafico a torta 2.5 notiamo che la maggioranza delle famiglie italiane riesce ad arrivare a fine mese, seppure con **qualche difficoltà**.
Una piccola parte riesce ad arrivare a fine mese con molta facilità, e fin troppe famiglie hanno difficoltà **medio-grandi** ad arrivare a fine mese.

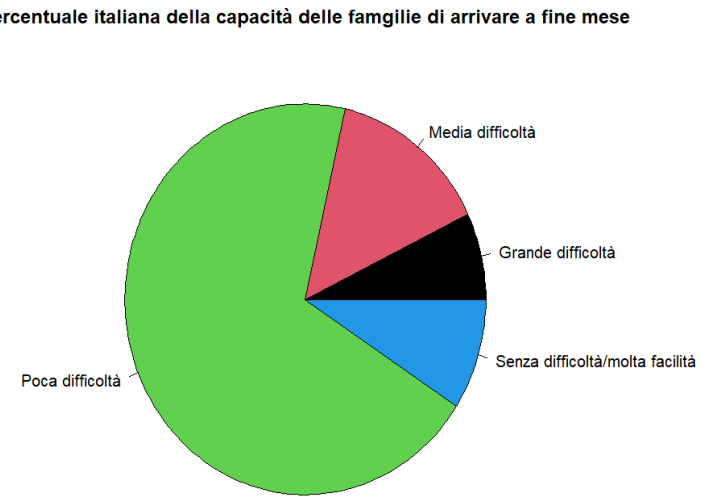


Figura 2.5: Grafico a torta per percentuale di famiglie con grande difficoltà.

2.3 Andamento del fenomeno tramite plot diagram

In questa sezione andiamo a visualizzare per ogni vettore (colonna) della nostra matrice di dati l'andamento dei valori tramite la funzione **plot()**.

Considerando che le regioni posizionate sull'asse delle x sono orientativamente disposte in ordine di localizzazione (le regioni più a sinistra sono concentrate nel Nord-Italia quelle a destra nel Sud-Italia e isole), risulta interessante valutare l'andamento del fenomeno lungo le varie macro-aree della penisola.

2.3.1 Plot diagram per famiglie con grandi difficoltà

Usiamo la funzione plot per costruire il diagramma, tra i parametri settiamo il titolo per l'asse X, quello per l'asse Y e il tipo di rappresentazione.
Attraverso la funzione **axis()** visualizziamo graficamente i due assi con i valori ad esso associati.

```
#PLOT DIAGRAM FAMIGLIE CON GRANDI DIFFICOLTA
plot(grande_difficolta,
   ylab="Percentuale famiglie con grandi difficolta",
   xlab="Regioni", col=1:19,
   type="b", axes=FALSE)

axis(side=2)
axis(side=1, at=1:19, labels=etichette_regioni, cex.axis=0.50)
```

Si ottiene il grafico 2.6:

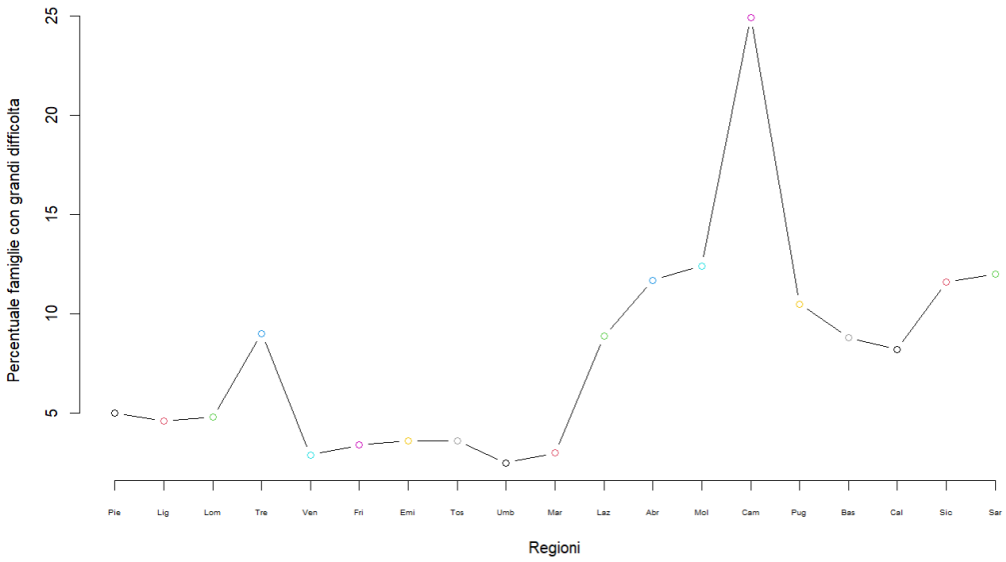


Figura 2.6: Plot diagram famiglie con grandi difficoltà

Risulta immediatamente evidente come nella parte sinistra, che fa riferimento alle zone del Nord l'andamento, eccezion fatta per il trentino, risulta avere un andamento senza grosse variazioni e per valori bassi, ad un certo punto (localizzato in prossimità delle Marche) si nota un repentino incremento dei valori a dimostrazione che nel sud-italia le famiglie con grandi difficoltà siano maggiori.

2.3.2 Plot diagram per famiglie con media difficoltà

```
#PLOT DIAGRAM FAMIGLIE CON MEDIA DIFFICOLTA

plot(media_difficolta,
  ylab="Percentuale famiglie con media difficolta",
  xlab="Regioni", col=20:40, type="b", axes=FALSE)

axis(side=2)
axis(side=1, at=1:19, labels=etichette_regioni, cex.axis=0.50)
```

Si ottiene il grafico 2.7:

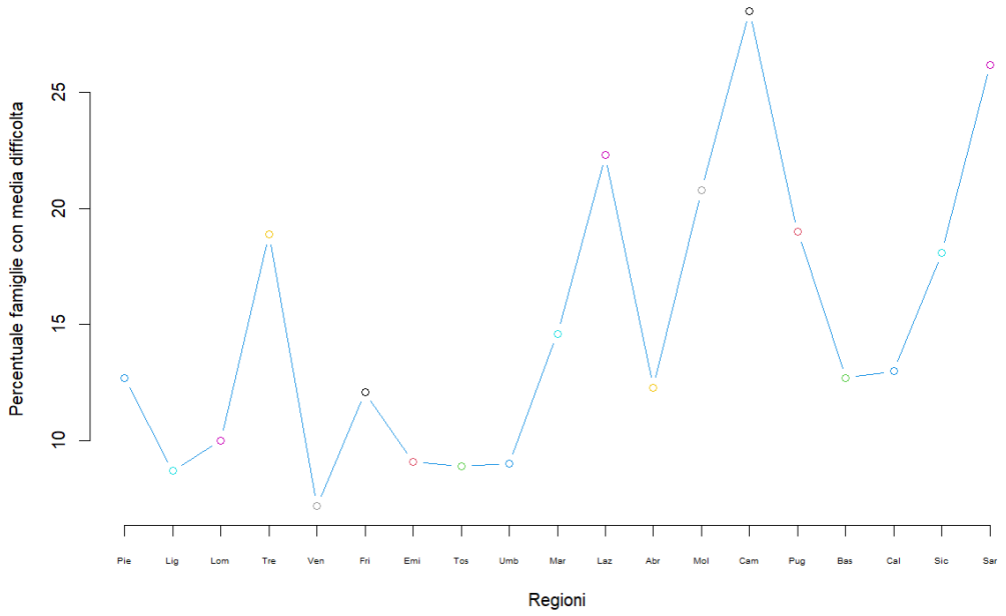


Figura 2.7: Plot diagram famiglie con media difficoltà

Si nota come in questo caso, l'andamento non risulta avere una distribuzione particolarmente caratterizzante per macro-aree, mostrando come, ad eccezione di alcune regioni, le famiglie che arrivano a fine mese con un livello di difficoltà medio siano approssimativamente distribuite su tutta la penisola.

2.3.3 Plot diagram per famiglie con poca difficoltà

```
#PLOT DIAGRAM FAMIGLIE CON POCA DIFFICOLTA
plot(poca_difficolta,
   ylab="Percentuale famiglie con poca difficolta",
   xlab="Regioni", col=1:19, type="b", axes=FALSE)

axis(side=2)
axis(side=1, at=1:19, labels=etichette_regioni, cex.axis=0.50)
```

Si ottiene il grafico 2.8:

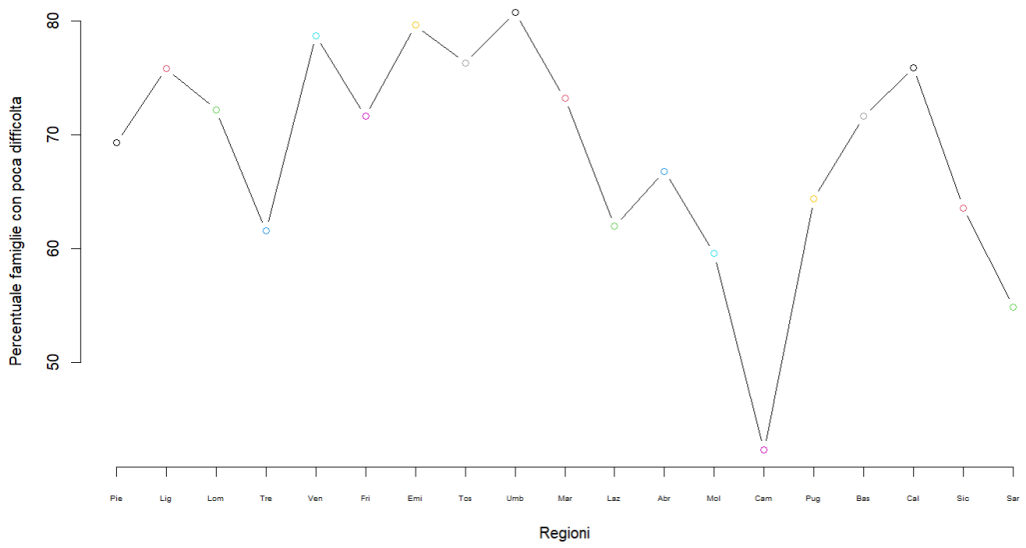


Figura 2.8: Plot diagram famiglie con poca difficoltà

Quello che risulta immediatamente evidente osservando questo grafico è il drastico andamento in discesa che si verifica nella porzione di regioni facente parte del Sud-Italia mostrando come in tale macro area la gran parte delle famiglie ha un certo grado di difficoltà ad arrivare a fine mese.

2.3.4 Plot diagram per famiglie senza difficoltà

```
#PLOT DIAGRAM FAMIGLIE SENZA DIFFICOLTA

plot(molta_facilita ,
   ylab="Percentuale famiglie senza difficolta",
   xlab="Regioni", col=1:19, type="b", axes=FALSE)

axis(side=2)
axis(side=1, at=1:19, labels=etichette_regioni , cex.axis=0.50)
```

Si ottiene il grafico 2.9:

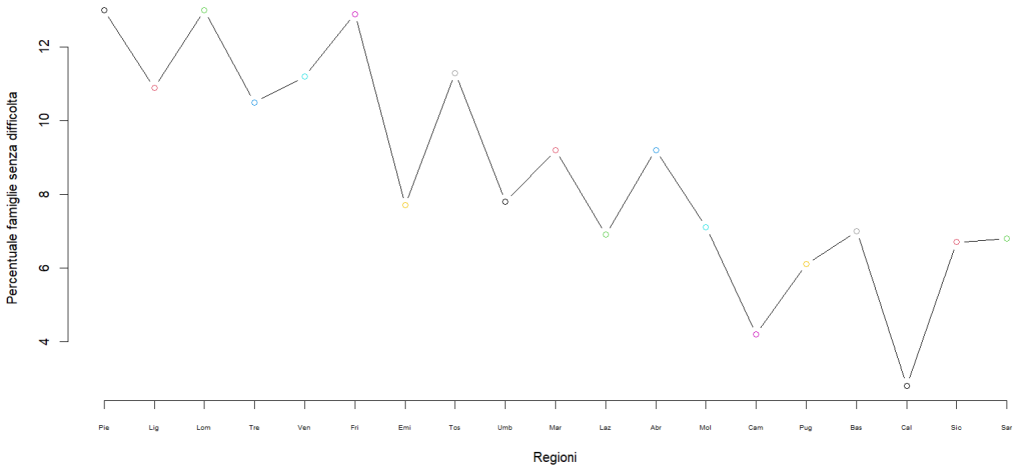


Figura 2.9: Plot diagram famiglie senza difficoltà

Risulta immediato notare come in questo ultimo caso, in maniera quasi complementare al primo, l'andamento sia purtroppo estremamente discendente, confermando quanto notato nel primo caso come nella macro area Sud arrivare a fine mese senza difficoltà risulta quasi essere un caso eccezionale.

2.4 Boxplot

Consideriamo un campione $(x_1, ..., x_n)$ dei valori assunti da una variabile quantitativa X. Procediamo ad ordinare i valori del campione in ordine crescente. Si chiama primo quartile, e si indica con Q_1, 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 Q_3, il valore per il quale il 75% dei dati sono alla sua sinistra e il restante 25% alla sua destra. Il secondo quartile Q_2, ossia il valore per il quale 50% dei dati sono alla sua sinistra e il restante 50% è alla sua destra è detto mediana. Q_0 e Q_4 forniscono il minimo e il massimo dei valori del campione. In R i quartili si calcolano tramite la funzione **quantile(nomeVettore)**.

Il boxplot, detto anche scatola con baffi, è il disegno di una scatola i cui estremi sono Q_1 e Q_3, tagliata da una linea orizzontale in corrispondenza di Q_2, 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 $Q_1 - 1.5 * (Q_3 - Q_1)$, mentre il baffo superiore corrisponde al valore più grande delle osservazioni che risulta minore o uguale a $Q_3 + 1.5 * (Q_3 - Q_1)$. La distanza tra il primo e il terzo quartile è detta intervallo interquartile. Quindi, se tutti i dati rientrano nell'intervallo $(Q_1 - 1.5 * (Q_3 - Q_1), Q_3 + 1.5 * (Q_3 - Q_1))$ i baffi sono posti in corrispondenza del minimo valore e del massimo valore del campione. Gli eventuali valori al di fuori dell'intervallo sono visualizzati nel grafico sotto forma di punti, detti valori anomali.

Il boxplot viene utilizzato per illustrare alcune caratteristiche di una distribuzione di frequenza: la centralità, la forma, la dispersione e la presenza di eventuali valori anomali.
La centralità è espressa dalla mediana. 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 Q_3 e del baffo inferiore da Q_1.

Di seguito saranno illustrati i quartili e i boxplot ad essi associati per ognuno dei nostri vettori.

2.4.1 Boxplot famiglie con grandi difficoltà

Otteniamo i vari quartili relativi alle famiglie con grandi difficoltà ad arrivare a fine mese e il boxplot risultante attraverso le seguenti linee di codice:

```
#QUANTILI E BOXPLOT FAMIGLIE CON GRANDI DIFFICOLTA
quantile(grande_difficolta)
summary(grande_difficolta)

boxplot(grande_difficolta,
     main="Boxplot famiglie con grandi difficolta",
     col="blue", axes=TRUE)
box()
```

Il risultato che otteniamo attraverso i primi due comandi è il seguente:

```
> quantile(grande_difficolta)
0%   25%   50%   75%  100%
2.50  3.60  8.20 11.05 24.90

> summary(grande_difficolta)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2.500   3.600   8.200   7.968  11.050  24.900
```

Da ciò deduciamo, in maniera chiara attraverso il comando **summary()** che $Q_0 = 2.50$, $Q_1 = 3.60$, $Q_2 = 8.20$, $Q_3 = 11.05$, $Q_4 = 24.90$.

Il boxplot invece è rappresentato in figura 2.10, mostra che gli estremi del box sono proprio Q_1 e Q_3, è tagliato da Q_2, il baffo inferiore è proprio il valore minimo Q_0 e quello superiore il valore massimo Q_4.

Volendo esaminare la simmetria del grafico risulta che:

- $Q_3 - Q_2 = 2.85$
- $Q_2 - Q_1 = 4.60$

Risulta esserci un dato anomalo in posizione $Q_4 = 24,9$ essendo il baffo superiore uguale a $11.05 + 1.5 * (11.05 - 3.60) = 22.225$

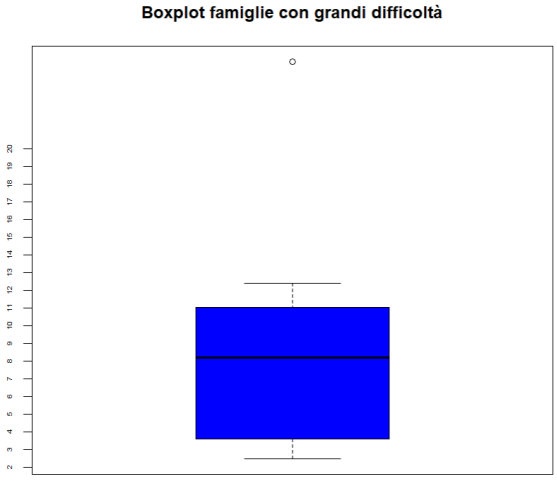


Figura 2.10: Boxplot famiglie con grandi difficoltà

2.4.2 Boxplot famiglie con media difficoltà

Otteniamo i vari quartili relativi alle famiglie con media difficoltà ad arrivare a fine mese e il boxplot risultante attraverso le seguenti linee di codice:

```
#QUANTILI E BOXPLOT FAMIGLIE CON MEDIA DIFFICOLTA
quantile(media_difficolta)
summary(media_difficolta)

boxplot(media_difficolta,
main="Boxplot famiglie con media difficolta", col="red", axes=FALSE)
axis(side=2, 2:20, cex.axis=0.5)
box()
```

Il risultato che otteniamo attraverso i primi due comandi è il seguente:

```
> quantile(media_difficolta)
0%   25%   50%   75%  100%
7.20  9.55 12.70 18.95 28.50

> summary(media_difficolta)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
7.20    9.55   12.70   14.95   18.95   28.50
```

Da ciò deduciamo, in maniera chiara attraverso il comando **summary()** che $Q_0 = 7.20$, $Q_1 = 9.55$, $Q_2 = 12.70$, $Q_3 = 18.95$, $Q_4 = 28.50$.

Il boxplot invece è rappresentato in figura 2.11, mostra che gli estremi del box sono proprio Q_1 e Q_3, è tagliato da Q_2, il baffo inferiore è proprio il valore minimo Q_0 e quello superiore il valore massimo Q_4.

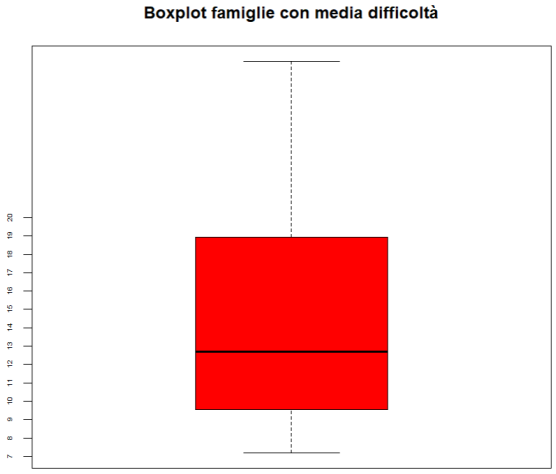


Figura 2.11: Boxplot famiglie con media difficoltà

Volendo esaminare la simmetria del grafico risulta che:

- $Q_3 - Q_2 = 6.25$
- $Q_2 - Q_1 = 3.15$

In questo notiamo un'asimmetria marcata dei dati.

2.4.3 Boxplot famiglie con poca difficoltà

Otteniamo i vari quartili relativi alle famiglie con poca difficoltà ad arrivare a fine mese e il boxplot risultante attraverso le seguenti linee di codice:

```
#QUANTILI E BOXPLOT FAMIGLIE CON POCA DIFFICOLTA
quantile(poca_difficolta)
summary(poca_difficolta)

boxplot(poca_difficolta,
main="Boxplot famiglie con poca difficolta", col="green", axes=FALSE)
axis(side=2, 2:20, cex.axis=0.5)
box()
```

Il risultato che otteniamo attraverso i primi due comandi è il seguente:

```
> quantile(poca_difficolta)
0%   25%   50%   75%  100%
42.40 62.80 71.60 75.85 80.70

> summary(poca_difficolta)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
42.40   62.80   71.60   68.43   75.85   80.70
```

Da ciò deduciamo, in maniera chiara attraverso il comando **summary()** che $Q_0 = 42.40$, $Q_1 = 62.80$, $Q_2 = 71.60$, $Q_3 = 75.85$, $Q_4 = 80.70$.

Il boxplot invece è rappresentato in figura 2.12, mostra che gli estremi del box sono proprio Q_1 e Q_3, è tagliato da Q_2, il baffo inferiore è proprio il valore minimo Q_0 e quello superiore il valore massimo Q_4.

Inoltre risulta che:

- $Q_3 - Q_2 = 4.25$
- $Q_2 - Q_1 = 8.0$

Risulta che ci sia un dato anomalo, questo perché essendo il baffo inferiore in corrispondenza di $62.80 - 1.5 * (75.82 - 62.80) = 43.27$ risulta $Q_0 = 42.40$.

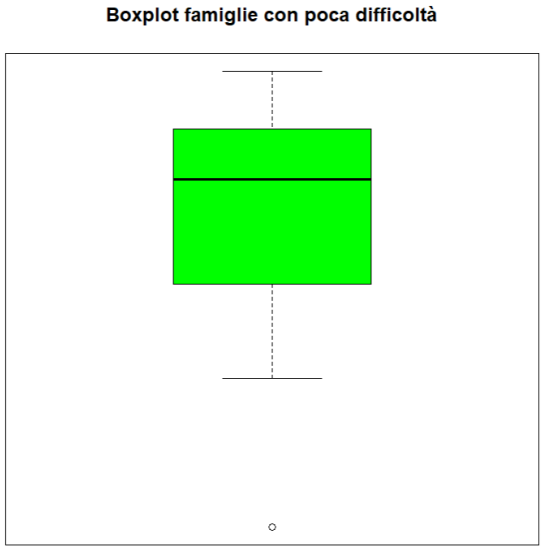


Figura 2.12: Boxplot famiglie con poca difficoltà

2.4.4 Boxplot famiglie senza difficoltà

Otteniamo i vari quartili relativi alle famiglie senza difficoltà ad arrivare a fine mese e il boxplot risultante attraverso le seguenti linee di codice:

```
#QUANTILI E BOXPLOT FAMIGLIE SENZA DIFFICOLTA
quantile(molta_facilita)
summary(molta_facilita)

boxplot(molta_facilita,
main="Boxplot famiglie senza difficolt ", col="yellow", axes=FALSE)
axis(side=2, 2:20, cex.axis=0.5)
box()
```

Il risultato che otteniamo attraverso i primi due comandi è il seguente:

```
> quantile(molta_facilita)
0%   25%   50%   75%  100%
2.80  6.85  7.80 11.05 13.00

> summary(molta_facilita)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2.800   6.850   7.800   8.647  11.050  13.000
```

Da ciò deduciamo, in maniera chiara attraverso il comando **summary()** che $Q_0 = 2.80$, $Q_1 = 6.85$, $Q_2 = 7.80$, $Q_3 = 11.05$, $Q_4 = 13.00$.

Il boxplot invece è rappresentato in figura 2.13, mostra che gli estremi del box sono proprio Q_1 e Q_3, è tagliato da Q_2, il baffo inferiore è proprio il valore minimo Q_0 e quello superiore il valore massimo Q_4.

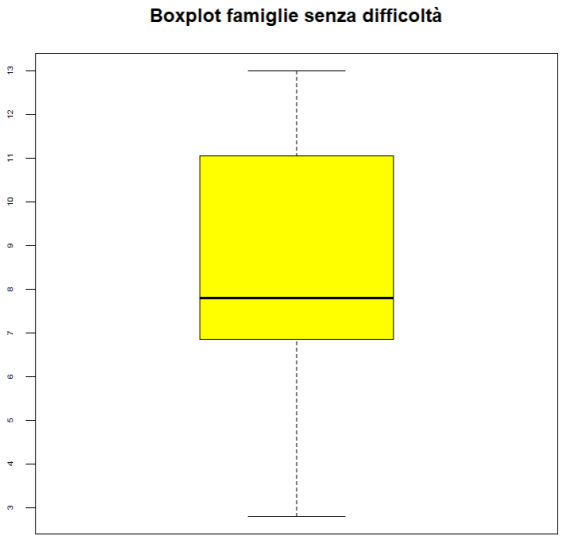


Figura 2.13: Boxplot famiglie senza difficoltà

Dal grafico risulta che:

- $Q_3 - Q_2 = 3.25$
- $Q_2 - Q_1 = 0.95$

Si nota inoltre una asimmetria dei dati.

Capitolo 3

Statistica descrittiva

3.1 Statistica descrittiva univariata

La **statistica descrittiva univariata** descrive la distribuzione di una singola variabile e include gli indici di posizione centrali e gli indici di dispersione che misurano quanto si disperdono i dati rispetto alla media.

3.1.1 Funzione di distribuzione empirica discreta

Nel caso discreto questa funzione è definita a partire dalle frequenze relative cumulative, così definite:

$$F_i = f_1 + ... + f_i = (n_1 + ... + n_i)/n$$

dove la generica F_i rappresenta la proporzione dei dati del campione minori o uguali di z_i.
Se supponiamo che i k valori distinti assunti dalla variabile quantitativa X siano ordinati in ordine crescente, ossia $z_1 < ... < z_k$, allora la funzione di distribuzione empirica $F(x)$ è definita come in figura 3.1

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

Figura 3.1: Funzione di distribuzione empirica discreta

La funzione:

- è non decrescente;
- assume il valore a sinistra in corrispondenza ad ogni punto di salto;
- vale 0 per ogni valore minore dell'osservazione minima e vale 1 per ogni valore maggiore o uguale dell'osservazione massima.

R offre la funzione **ecdf()**.

Ai fine della nostra analisi una funzione di questo tipo non è molto utile dato che i vettori che si stanno utilizzando sono composti quasi interamente da valori differenti, pertanto verrà mostrato solo un esempio ai fini di completezza.

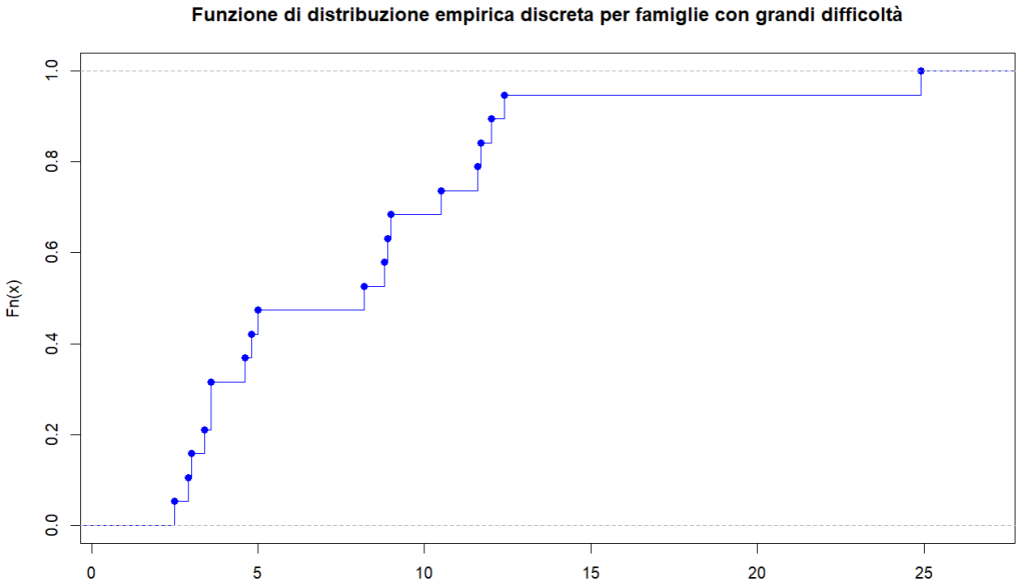


Figura 3.2: Funzione di distribuzione empirica discreta per famiglie con grandi difficoltà

3.1.2 Funzione di distribuzione empirica continua

Per fenomeni quantitativi continui occorre considerare la **funzione di distribuzione empirica continua**, ossia una funzione di distribuzione empirica strutturata in classi.
Essa risulta definita come in figura 3.3.

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

Figura 3.3: Funzione di distribuzione empirica continua

Dove $F_0 = 0$ e F_i denota la frequenza relativa cumulativa della classe C_i.

La funzione:

- vale 0 per ogni x minore di z_1;
- vale 1 per ogni x maggiore/uguale di z_{k+1};
- coincide con il segmento che passa per i punti (z_{i1}, F_{i1}) e (z, F_i);

Applichiamo ora questa funzione ai nostri dati e alle classi:

Distribuzione continua famiglie con grandi difficoltà

```
#DISTRIBUZIONE CONINTUA FAMIGLIE CON GRANDE DIFFICOLTA

classi_distribuzione_continua_grande<- c(0,2,6,10,14,18,22,26,30)

frequenza_cumulativa_grande_difficolta<-cumsum(table(cut(grande_
    difficolta,
  breaks=classi_distribuzione_continua_grande,
  right =FALSE)))/length(grande_difficolta)

frequenza_cumulativa_grande_difficolta<-c(0,frequenza_cumulativa_
    grande_difficolta)

plot(classi_distribuzione_continua_grande,
  frequenza_cumulativa_grande_difficolta,
  type = "b", axe =FALSE,
  main = "Funzione di distribuzione empirica continua famgilie con
      grandi difficolt ",
  col="blue", xlab = "Classi",
  ylab = "Frequenze cumulate")

axis(1, classi_distribuzione_continua_grande, cex.axis=0.80)
axis(2, format(frequenza_cumulativa_grande_difficolta, digits = 2),
  cex.axis=0.80)
box()
```

Che produce 3.4.

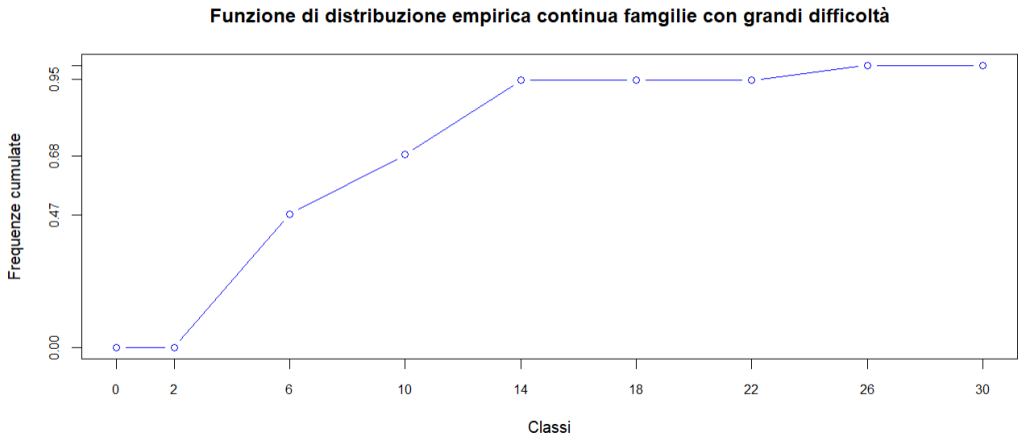


Figura 3.4: Funzione di distribuzione empirica continua famiglie con grande difficoltà

Si nota come facilmente che prima di arrivare in prossimità della classe che contiene il valore minimo, la funzione assuma valori pari a 0 e raggiunta la classe contenente il massimo valga 1.

Distribuzione continua famiglie con media difficoltà

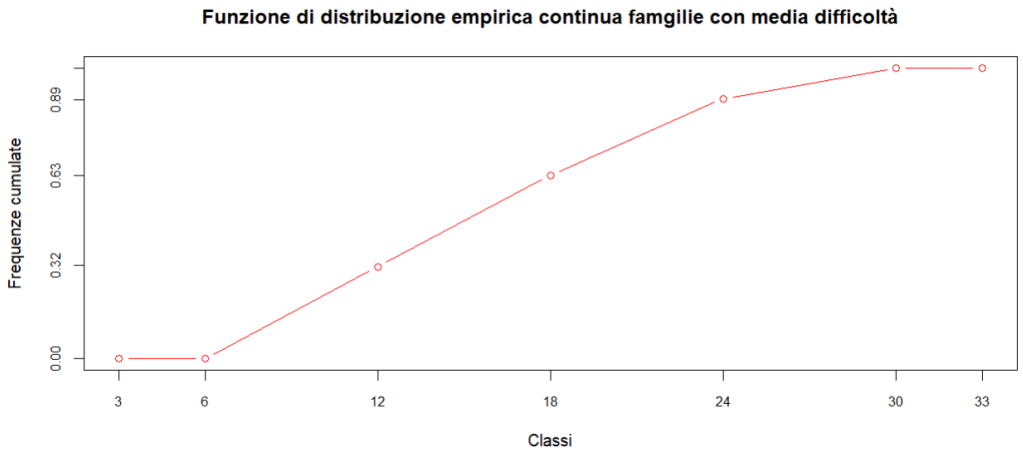


Figura 3.5: Funzione di distribuzione empirica continua famiglie con media difficoltà

Notiamo come la percentuale di famiglie che arrivano a fine mese con media difficoltà sia concentrata tra il 6 e il 30 per cento.

Distribuzione continua famiglie con poca difficoltà

Applichiamo la stessa logica a seconda del vettore considerato.

La percentuale di famiglie che arrivano a fine mese con poche difficoltà è concen-

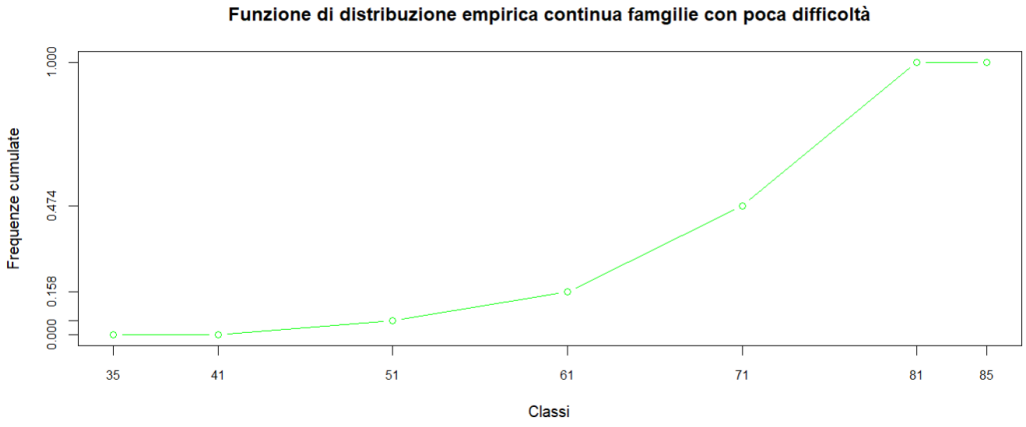


Figura 3.6: Funzione di distribuzione empirica continua famiglie con poca difficoltà

trata tra il 41 e 81 per cento.

Distribuzione continua famiglie senza difficoltà

Applichiamo la stessa logica a seconda del vettore considerato.

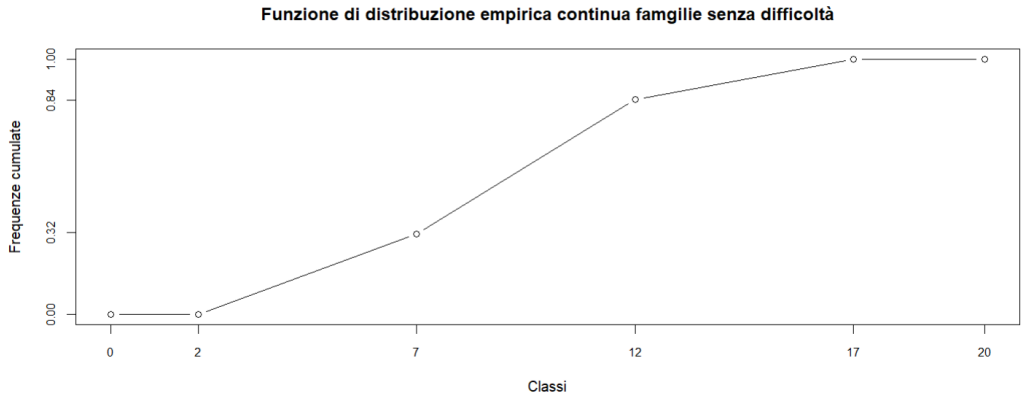


Figura 3.7: Funzione di distribuzione empirica continua famiglie senza difficoltà

Si hanno tra il 2 e 17 per cento di famiglie che riescono ad arrivare a fine mese senza difficoltà.

3.1.3 Indici di sintesi: media, mediana e moda campionaria

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 box plot analizzati nei capitoli precedenti.

Media campionaria

Supponiamo di avere un insieme $x_1, ..., x_n$ 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:

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

Vediamo quindi il valore della media campionaria sui nostri dati attraverso la funzione **mean()**.

```
#MEDIA CAMPIONARIA

media_grande_difficolta<-mean(grande_difficolta)
media_grande_difficolta
[1] 7.968421

media_media_difficolta<-mean(media_difficolta)
media_media_difficolta
[1] 14.95263

media_poca_difficolta<-mean(poca_difficolta)
media_poca_difficolta
[1] 68.43158

media_senza_difficolta<-mean(molta_facilita)
  media_senza_difficolta
[1] 8.647368
```

Nonostante la media ci dia un indicazione generale sui dati in questione, risulta conveniente derivare altri indici per valutare al meglio la distribuzione dei dati e la loro variabilità.

Mediana campionaria

Assegnato un insieme di dati di ampiezza n, lo si ordini in ordine crescente. Se n è dispari, si definisce mediana campionaria il valore che è in posizione $(n+1)/2$, mentre se n è pari la mediana campionaria è invece definita come la media aritmetica dei valori che occupano le posizioni $n/2$ e $n/2+1$.
Usiamo la funzione **median()** applicata ai nostri dati:

```
#MEDIANA CAMPIONARIA

mediana_grande_difficolta<-median(grande_difficolta)
mediana_grande_difficolta
[1] 8.2

mediana_media_difficolta<-median(media_difficolta)
mediana_media_difficolta
[1] 12.7

mediana_poca_difficolta<-median(poca_difficolta)
mediana_poca_difficolta
[1] 71.6

mediana_senza_difficolta<-median(molta_facilita)
mediana_senza_difficolta
[1] 7.8
```

Confrontando la media e la mediana campionaria si nota che la mediana campionaria è, ad eccezione del secondo caso, maggiore della media, ciò comporta che la distribuzione di frequenza sia più sbilanciata verso sinistra.

Moda campionaria

La moda campionaria di un insieme di dati, se esiste, è la modalità a cui è associata la frequenza (assoluta o relativa) più elevata. Se esistono più modalità con frequenza massima, ciascuna di esse è detto valore modale.
La moda è facilmente individuabile analizzando gli istogrammi delle frequenze.

Istogrammi

Gli istogrammi, che si utilizzano per variabili quantitative, sono una particolare rappresentazione grafica di una distribuzione di frequenza in classi. Consideriamo un campione $(x_1, ..., x_n)$ costituito da n osservazioni, che suddividiamo in classi; ogni osservazione deve cadere in una ed una sola classe.
Gli istogrammi sono quindi una particolare rappresentazione grafica ottenuta mediante rettangoli adiacenti aventi per basi segmenti i cui estremi corrispondono agli estremi delle classi. L'asse delle ascisse di un istogramma è sempre dotato di un'unità di misura.
Se si utilizzano le frequenze assolute delle classi, l'area di ogni rettangolo è uguale alla frequenza assoluta della classe e l'area totale dei rettangoli è uguale all'ampiezza del campione.
Se si utilizzano le frequenze relative delle classi l'area di ogni rettangolo è uguale alla frequenza relativa della classe stessa e l'area totale è uguale all'unità.

R offre la funzione **hist()** per generare istogrammi, ed ottenere una serie di informazioni che riguardano questi ultimi, come le frequenze, le densità e i punti centrali.

Istogrammi per famiglie con grandi difficoltà

Vediamo la costruzione di un istogramma relativo alla frequenza assoluta, ottenibile attraverso il parametro **freq=TRUE**, delle famiglie con grandi difficoltà. Interessante notare come R assegni automaticamente delle classi se queste ultime non vengono specificate.

```
#ISTOGRAMMA FAMIGLIE CON GRANDI DIFFICOLTA'
grandi_difficolta_isto<-hist(grande_difficolta, freq=TRUE,
main="Istogramma famiglie con grandi difficolta",
ylab="Frequenza assoluta delle classi", col=1:7)

str(grandi_difficolta_isto)
```

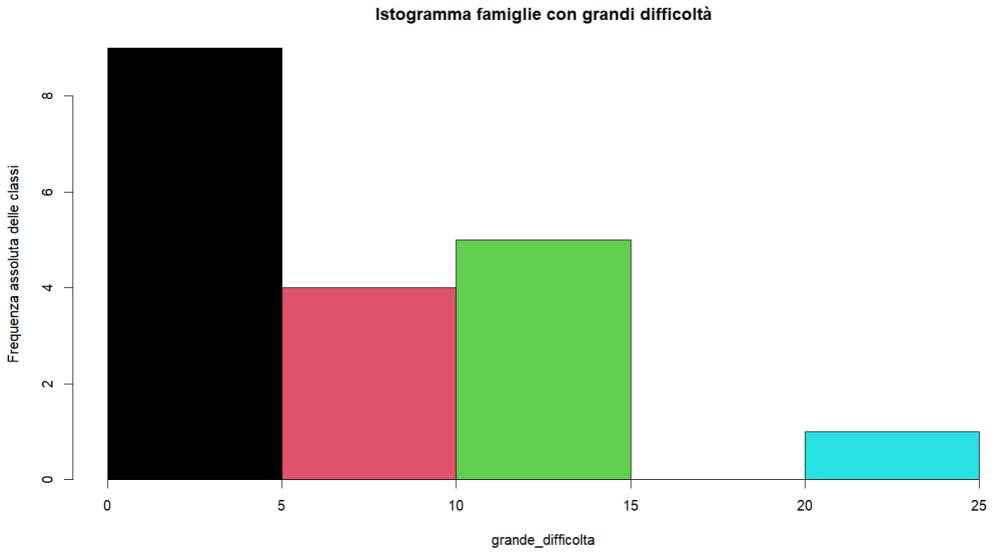


Figura 3.8: Istogramma frequenze assolute famiglie con grandi difficoltà

Come si vede la funzione str() fornisce i punti di suddivisione in classi(breaks), le frequenze assolute delle classi(counts), la densità delle classi(density) e i punti centrali delle classi(mids):

```
List of 6
$ breaks  : num [1:6] 0 5 10 15 20 25
$ counts  : int [1:5] 9 4 5 0 1
$ density : num [1:5] 0.0947 0.0421 0.0526 0 0.0105
$ mids    : num [1:5] 2.5 7.5 12.5 17.5 22.5
$ equidist: logi TRUE
- attr(*, "class")= chr "histogram"
```

Istogrammi per famiglie con medie difficoltà

Grafico in figura 3.9.

```
media_difficolta_isto<-hist(media_difficolta, freq=TRUE,
main="Istogramma famiglie con media difficolt ",
ylab="Frequenza assoluta delle classi", col=1:7)
```

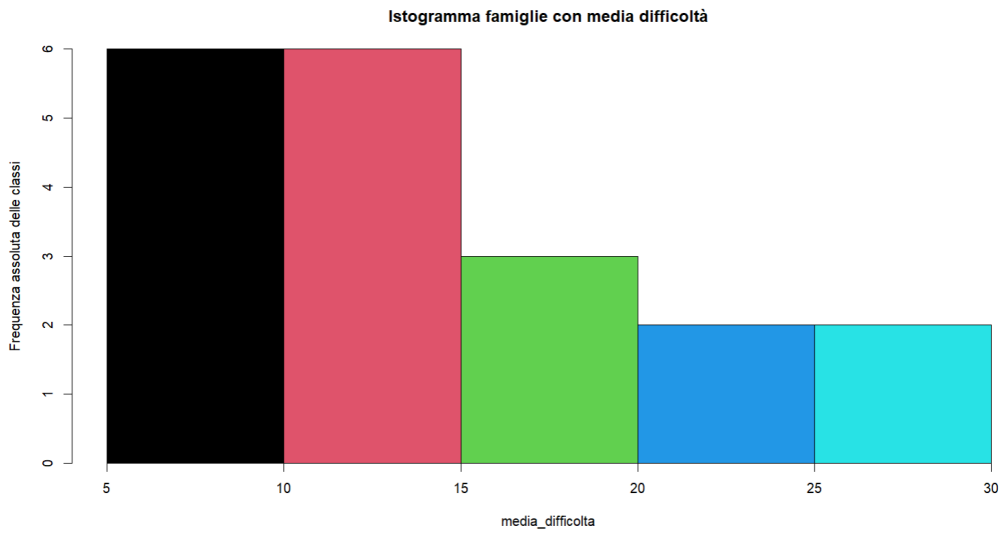


Figura 3.9: Istogramma frequenze assolute famiglie con media difficoltà

Output funzione **str()**:

```
List of 6
$ breaks  : int [1:6] 5 10 15 20 25 30
$ counts  : int [1:5] 6 6 3 2 2
$ density : num [1:5] 0.0632 0.0632 0.0316 0.0211 0.0211
$ mids    : num [1:5] 7.5 12.5 17.5 22.5 27.5
$ xname   : chr "media_difficolta"
$ equidist: logi TRUE
- attr(*, "class")= chr "histogram"
```

Istogrammi per famiglie con poche difficoltà

Grafico in figura 3.10.

```
#ISTOGRAMMA FAMIGLIE CON POCHE DIFFICOLTA ASSOLUTE
poche_difficolta_isto<-hist(poca_difficolta, freq=TRUE,
main="Istogramma famiglie con poca difficolta",
ylab="Frequenza assoluta delle classi", col=1:7)

str(poche_difficolta_isto)
```

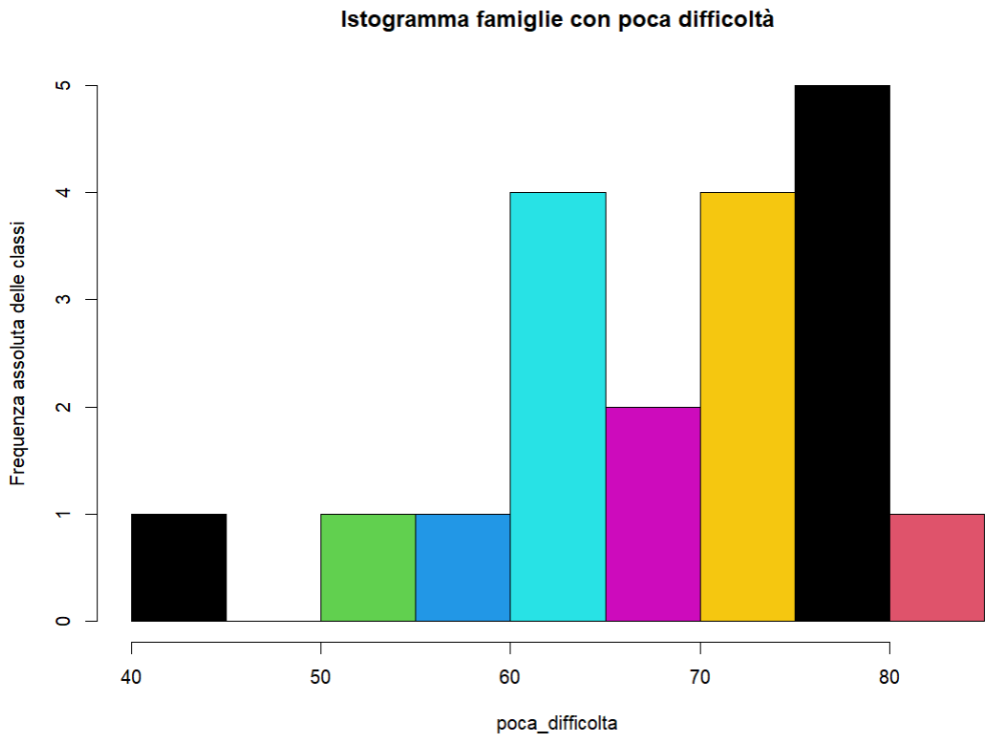


Figura 3.10: Istogramma frequenze assolute famiglie con poche difficoltà

Output funzione **str()**:

```
List of 6
$ breaks  : int [1:10] 40 45 50 55 60 65 70 75 80 85
$ counts  : int [1:9] 1 0 1 1 4 2 4 5 1
$ density : num [1:9] 0.0105 0 0.0105 0.0105 0.0421 ...
$ mids    : num [1:9] 42.5 47.5 52.5 57.5 62.5 67.5 72.5 77.5 82.5
$ xname   : chr "poca_difficolta"
$ equidist: logi TRUE
```

Istogrammi per famiglie senza difficoltà

Grafico in figura 3.11.

```
#ISTOGRAMMA FAMIGLIE SENZA DIFFICOLTA ASSOLUTE
senza_difficolta_isto<-hist(molta_facilita , freq=TRUE,
main="Istogramma famiglie senza difficolta",
ylab="Frequenza assoluta delle classi", col=1:7)

str(senza_difficolta_isto)
```

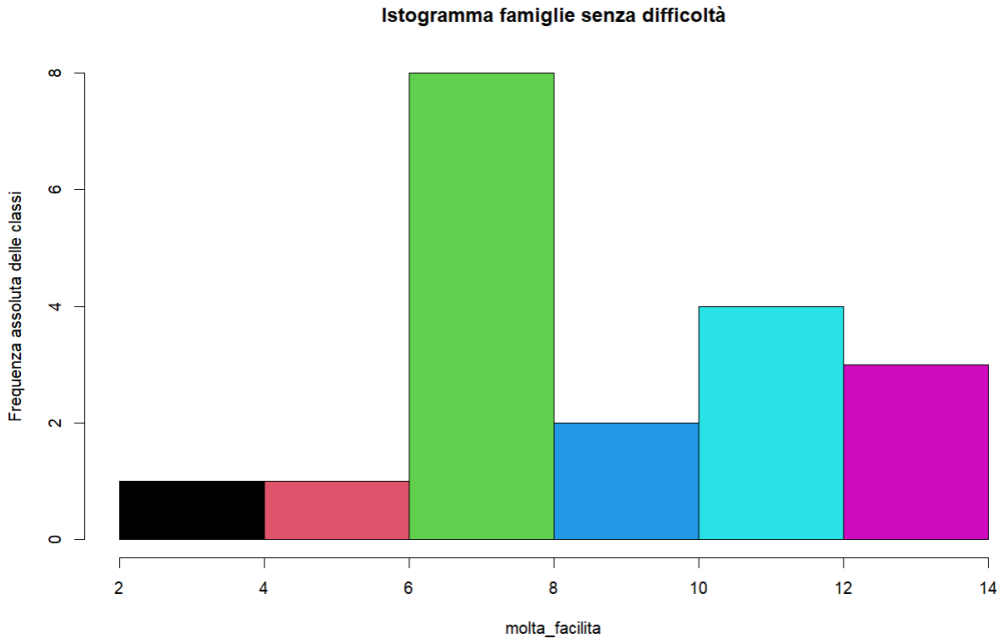


Figura 3.11: Istogramma frequenze assolute famiglie senza difficoltà

Output funzione **str()**:

```
List of 6
$ breaks   : int [1:7] 2 4 6 8 10 12 14
$ counts   : int [1:6] 1 1 8 2 4 3
$ density  : num [1:6] 0.0263 0.0263 0.2105 0.0526 0.1053 ...
$ mids     : num [1:6] 3 5 7 9 11 13
$ xname    : chr "molta_facilita"
$ equidist: logi TRUE
- attr(*, "class")= chr "histogram"
```

3.1.4 Esempio di kernel density plot applicato alle famiglie senza difficoltà

Come sappiamo la scelta degli intervalli delle classi in un istogramma è una scelta cruciale per l'aspetto finale del grafico.
Un modo per gestire gli aspetti che riguardano la forma di un istogramma consiste nell'utilizzare una stima della densità basata su **kernel**.
Con tale metodo, invece di raccogliere le osservazioni in barre, si traccia una curva continua determinata da un fattore K, detto **kernel**, e da un parametro h, detto **ampiezza della banda** (bandwidth).
Sia $(x_1, ..., x_n)$ un campione costituito da n osservazioni di una variabile la cui densità di frequenza f non è nota in ogni punto x. Siamo interessati a stimare la forma di questa funzione f in base al campione di osservazioni.
Un grafico può essere ottenuto utilizzando:

$$\hat{f}_h(x) = \frac{1}{(nh)} \sum_{i=1}^{n} K(\frac{(x-x_i)}{h})$$

dove n è l'ampiezza del campione, $K(x)$ è il kernel, ossia una funzione densità di probabilità non negativa con media zero e $h > 0$ è un parametro di smoothing, detto bandwidth.
La scelta del Kernel influenza la forma del grafico finale.
In questa sezione si mostra, ad esempio, l'applicazione di questo metodo alle famiglie che arrivano a fine mese senza difficoltà.
Nell'estratto di codice che segue è definito un kernel di tipo **gaussiano**:

```
#KERNEL DENSITY PLOT ISTOGRAMMA FAMIGLIE SENZA DIFFICOLTA
d1<-density(molta_facilita , kernel="gaussian")

plot(d1, lwd=2, main="Gaussian kernel", col="blue")
```

Che produce il grafico in figura 3.12.

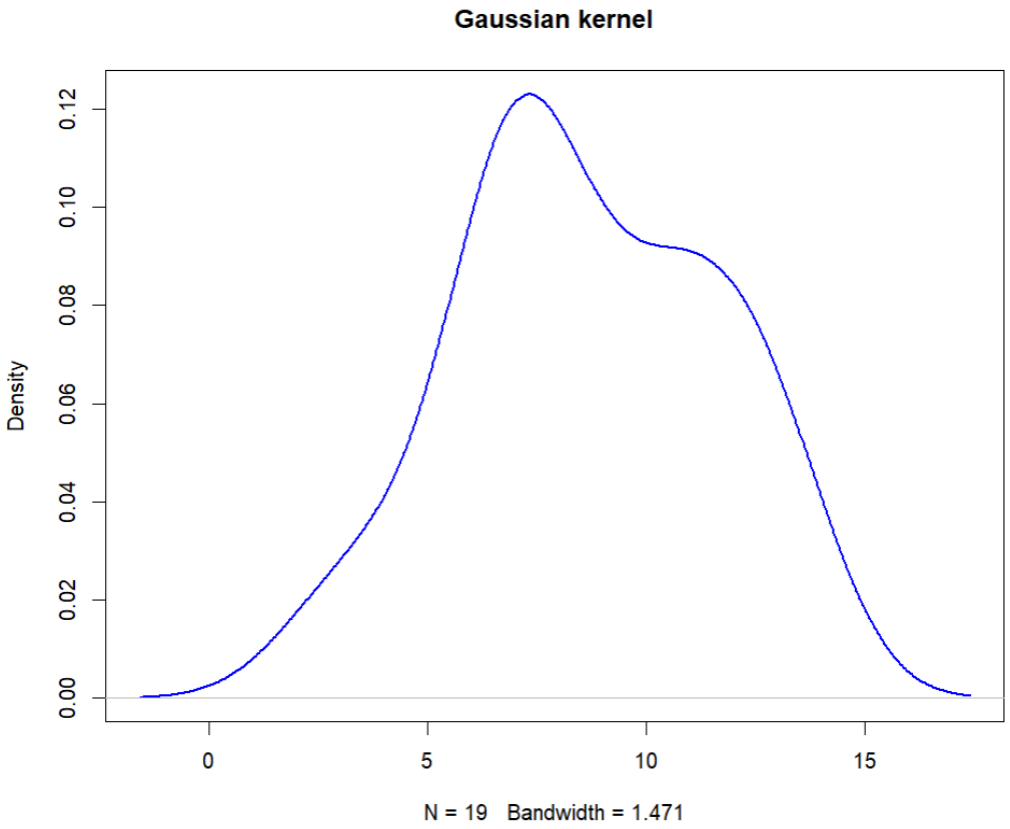


Figura 3.12: Kernel density plot famiglie senza difficoltà

3.1.5 Quantili, percentili, decili e quartili

Sia X una variabile quantitativa e sia $x_1, ..., x_n$ un campione di n osservazioni disposte in ordine crescente. Supponiamo di suddividere i dati ordinati in α gruppi, ognuno dei quali contenga (circa) lo stesso numero di osservazioni; gli $\alpha - 1$ gruppi che consentono tale suddivisione sono i quantili di ordine α. Ad esempio, possiamo suddividere i dati in $\alpha = 4$ parti mediante 3 quantili, detti quartili, oppure in 10 parti, mediante 9 quantili, detti decili, o addirittura in 100 parti mediante 99 quantili, detti percentili.

Vediamo il calcolo dei quartili per il vettore delle famiglie con grandi difficoltà e come cambia anche in base al tipo di algoritmo scelto in R:

```
#QUANTILI

quantile(grande_difficolta , type=7)#defualt
0%   25%   50%   75%  100%
 2.50  3.60  8.20 11.05 24.90
```

```
6
7   quantile(grande_difficolta, type=2)
8   0%  25%  50%  75% 100%
9   2.5  3.6  8.2 11.6 24.9
10
11  quantile(grande_difficolta, type=1)
12  0%  25%  50%  75% 100%
13  2.5  3.6  8.2 11.6 24.9
```

I boxplot associati ai quartili sono stati approfonditi nel capitolo terzo di questo lavoro.

3.1.6 Varianza, deviazione standard e coefficiente di variazione

Gli indici di posizione non tengono conto della variabilità dei dati, infatti, esistono distribuzioni di frequenza che sono molto diverse tra loro, pur avendo la stessa media campionaria. Indici significativi per misurare la variabilità dei dati sono la varianza campionaria e la deviazione standard campionaria. Tali indici sono detti indici di dispersione o indici di variabilità poiché misurano la dispersione dei dati intorno alla media.

Varianza e Deviazione standard

Assegnato un insieme di dati numerici $x_1, ..., x_n$, si definisce **varianza campionaria**, e si denota con s^2, la quantità:

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

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

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

Varianza campionaria e deviazione standard campionaria sono detti indici di dispersione o indici di variabilità poiché misurano la dispersione dei dati intorno alla media.
Calcoliamo in R questi due indici attraverso le funzioni **var()** e **sd()** e notiamo come, ad eccezione delle famiglie che arrivano a fine mese con facilità, i valori della varianza si discostino notevolmente dalla media.

```
#VARIANZA

var(grande_difficolta)
[1] 29.34895

var(media_difficolta)
[1] 38.85485

var(poca_difficolta)
[1] 93.3045

var(molta_facilita)
[1] 8.67152

#DEVIAZIONE STANDARD

sd(grande_difficolta)
[1] 5.417467

sd(media_difficolta)
[1] 6.233366

sd(poca_difficolta)
[1] 9.659426

sd(molta_facilita)
[1] 2.944745
```

Coefficiente di variazione

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

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

Si nota che il coefficiente di variazione è un numero puro, ossia è un indice adimensionale che non dipende dall'unità di misura utilizzata, poiché la media campionaria e la deviazione standard campionaria sono espressi in identiche unità di misura.
Nel nostro caso è importante 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.
In R non c'è una funzione ad hoc, tuttavia può essere facilmente ottenuto attraverso questa funzione:

```
#COEFFICIENTE DI VARIAZIONE

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

cv(grande_difficolta)
[1] 0.679867

cv(media_difficolta)
[1] 0.4168742

cv(poca_difficolta)
[1] 0.1411545

cv(molta_facilita)
[1] 0.3405365
```

Il coefficiente di variazione più alto si ha per le famiglie con grandi difficoltà ad arrivare a fine mese, in cui i dati si discostano maggiormente dalla media campionaria.

COLONNA	MEDIA	MEDIANA	VARIANZA	DS
Grandi difficoltà	7.968421	8.2	9.34895	5.417467
Media difficoltà	14.95263	12.7	38.85485	6.233366
Poche difficoltà	68.43158	71.6	93.3045	9.659426
Molta facilità	8.647368	7.8	8.67152	2.944745

3.1.7 Forma distribuzione di frequenze

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 un indice che permette di misurare la simmetria della funzione di distribuzione: **skewness**.

Skewness

Assegnato un insieme di dati numerici $x_1, ..., x_n$, con m_3 che denota il momento centrato campionario di ordine 3, si definisce skewness campionaria il valore:

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

Si possono avere tre casi:

- Se $\gamma_1 = 0$, la distribuzione di frequenza è simmetrica;
- Se $\gamma_1 > 0$, la distribuzione di frequenza ha la coda di destra più allungata per l'asimmetria positiva;
- Se $\gamma_1 < 0$, la distribuzione di frequenza ha la coda di sinistra più allungata per l'asimmetria negativa;

In R la definiamo e otteniamo cosi:

```
#SKEWNESS

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

```
9
10  skw(grande_difficolta)
11  [1] 1.561747
12
13  skw(media_difficolta)
14  [1] 0.7370512
15
16  skw(poca_difficolta)
17  [1] -0.9956742
18
19  skw(molta_facilita)
20  [1] -0.08455453
```

Notiamo come i dati siano fortemente asimmetrici in tutti i casi, in particolare in due casi si ha una marcata asimmetria positiva e negli altri una negativa.

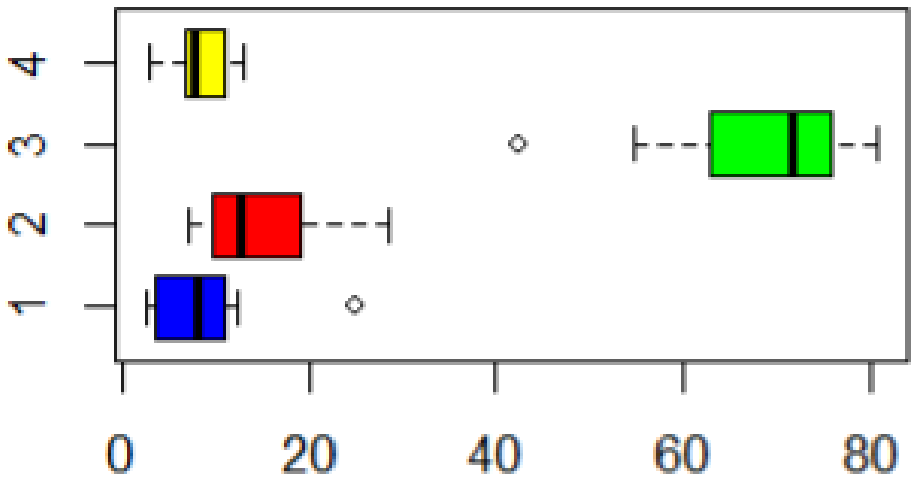


Figura 3.13: Confronto dei boxplot

3.2 Statistica descrittiva bivariata

In questo capitolo si tratta la statistica bivariata, ossia il ramo della statistica che si occupa dei metodi grafici e statistici atti a descrivere le relazioni che intercorrono tra due variabili quantitative.
Dopo aver scelto la variabile da porre sulle ascisse (variabile indipendente) X e la variabile da porre sulle ordinate (variabile dipendente) Y, si disegnano dei punti in corrispondenza delle coppie (x_i, y_i). Ciò che si ottiene disegnando tali punti mediante diagrammi di dispersione è una nuvola di punti che evidenziano, se esiste, una qualche forma di regolarità, o meglio una qualche forma di relazione tra le variabili.
In particolare, si analizzeranno le relazioni tra i dati presenti, attraverso indici di covarianza campionaria, coefficiente di correlazione campionario e vari grafici.
Successivamente si andrà ad osservare la regressione lineare semplice, multipla e non lineare.

R offre anche la possibilità di effettuare confronti attraverso una visualizzazione immediata di vari grafici affiancati mediante grafico a dispersione.

```
1 #RAPPRESENTAZIONE DI CONFRONTO TRAMITE DISPERSIONE
2
3 pairs(matrice_capacita_arrivare_fine_mese,
4 main = "Scatterplot per tutte le coppie di variabili")
```

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

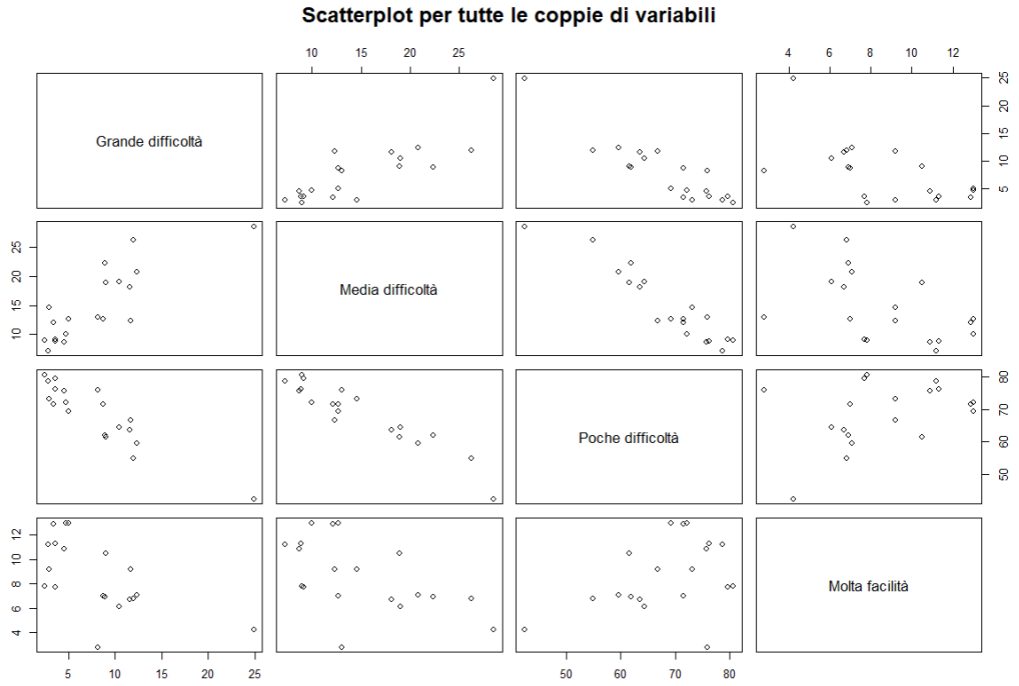


Figura 3.14: Confronto tramite grafico a dispersione

3.2.1 Covarianza e correlazione campionaria

Spesso nelle indagini statistiche si osservano più variabili quantitative per uno stesso gruppo di individui ed in tal caso è necessario vedere se esiste una correlazione tra le variabili. Un primo passo per indagare l'eventuale dipendenza tra due variabili X e Y consiste nel disegnare il diagramma di dispersione. Per ottenere una misura quantitativa della correlazione tra le variabili si considera la covarianza campionaria:

Assegnato un campione bivariato $(x_1, y_1), ..., (x_n, y_n)$ di una variabile quantitativa bidimensionale (X, Y).
La covarianza campionaria tra le due variabili X e Y è così definita:

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

Da questa definizione notiamo che il prodotto interno alla sommatoria sarà positivo per le osservazioni 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 essere:

- Se $C_{xy} > 0$ le variabili sono correlate positivamente;
- Se $C_{xy} < 0$ le variabili sono correlate negativamente;
- Se $C_{xy} = 0$ le variabili non sono correlate (ottenendo un grafico sparso);

Quando la covarianza campionaria assume valori positivi quindi quello che ci si aspetta è che i cambiamenti della prima variabile siano corrispondenti anche nella seconda. Non c'è concordanza invece in una covarianza negativa.
Calcoliamo la covarianza fra le varie categorie:

```
#COVARIANZA TRA COPPIE DI DATI

round(cov(matrice_capacita_arrivare_fine_mese), digits = 3)

                        Grande difficolta  Media difficolta
Grande difficolta              29.349            27.949
Media difficolta               27.949            38.855
Poche difficolta              -47.598           -56.815
Molta facilita                 -9.763           -10.078

              Poche difficolta  Molta facilita
Grande difficolta      -47.598         -9.763
Media difficolta       -56.815        -10.078
  Poche difficolta      93.305         11.217
Molta facilita          11.217          8.672
```

Si evince immediatamente come ci siamo coppie notevolmente correlate **negativamente** sopratutto come le famiglie con grandi difficoltà e senza difficoltà non crescano (o decrescano) insieme. Si hanno invece valori positivi per le coppie (grandeDifficoltà, mediaDifficoltà) e (pochedifficoltà, moltaFacilità).

Coefficiente di correlazione campionario

Assegnato un campione bivariato $(x_1, y_1), ..., (x_n, y_n)$ di una variabile quantitativa bidimensionale (X, Y), il coefficiente di correlazione campionario tra le due variabili X e Y è così definito:

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

Esso 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.
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.
Altre relazioni utili:

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

Otteniamo i valori in R attraverso il metodo **cor()**.

```
#COEFFICIENTE DI CORRELAZIONE

cor(matrice_capacita_arrivare_fine_mese)
```

```
                  Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
Grande difficoltà         1.0000000        0.8276505       -0.9095774     -0.6119746
Media difficoltà          0.8276505        1.0000000       -0.9436039     -0.5490503
Poche difficoltà         -0.9095774       -0.9436039        1.0000000      0.3943377
Molta facilità           -0.6119746       -0.5490503        0.3943377      1.0000000
```

Notiamo immediatamente che sulla diagonale ci sono tutti 1, ovviamente perché si confronta una variabile con se stessa.
Il coefficiente di correlazione campionario, inoltre, indica che se $0 < r_{xy} < 1$ allora i punti x_i, y_i sono posizionati in una nuvola attorno ad una linea retta interpolante ascendente.

3.2.2 Regressione lineare semplice

Il modello di **regressione lineare semplice** è esprimibile attraverso l'equazione di una retta che riesce ad interpolare la nuvola di punti dello scatterplot meglio di tutte e altre possibili rette.
Consideriamo l'equazione della retta: $Y = \alpha + \beta X$
Il coefficiente angolare β esprime quantitativamente la pendenza (inclinazione) della retta: un coefficiente angolare positivo ($\beta > 0$) indica una retta di regressione crescente, un coefficiente angolare negativo ($\beta < 0$) indica una retta decrescente; un coefficiente angolare nullo ($\beta = 0$) indica una retta orizzontale. L'intercetta invece corrisponde all'ordinata del punto di intersezione della retta interpolante (di regressione) con l'asse delle ordinate. L'identificazione di questa retta viene ottenuta applicando il metodo dei minimi quadrati, che conduce a:

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

Lo studio in R verrà effettuate considerando due categorie specifiche:
Variabile indipendente=**famiglie con grandi difficoltà** e Variabile dipendente=**famiglie con media difficoltà**.
Calcoliamo quindi α e β in R:

```
#CALCOLO DI ALPHA E BETA

beta<-(sd(media_difficolta)/sd(grande_difficolta))*cor(media_
    difficolta,grande_difficolta)
alpha<-mean(media_difficolta)-beta*mean(grande_difficolta)
c(alpha ,beta)

[1] 7.3643113 0.9522991
```

β risulta positiva e infatti la nostra retta è ascendente. Con α invece riusciamo a stimare dove la retta di regressione intercetta l'asse delle y.
Visualizziamo in R attraverso **lm(y ~ x)**, che indica y dipende da x(nel nostro caso famiglie con media difficoltà dipendono da famiglie con grandi difficoltà).

```
#INVOCAZIONE METODO lm()

lm(media_difficolta~grande_difficolta)

Call:
lm(formula = media_difficolta ~ grande_difficolta)

Coefficients:
(Intercept)  grande_difficolta
7.3643               0.9523
```

Che conferma i nostri calcoli, inoltre possiamo ottenere altre informazioni:

```
attributes(linear_model)
$names
[1] "coefficients"  "residuals"     "effects"       "rank"
[5] "fitted.values" "assign"        "qr"            "df.residual"
[9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"
```

Ad esempio restituiamo i coefficienti:

```
linear_model$coefficients

(Intercept) grande_difficolta
7.3643113          0.9522991
```

Completiamo la sezione mostrando una visualizzazione grafica dei dati 3.15.

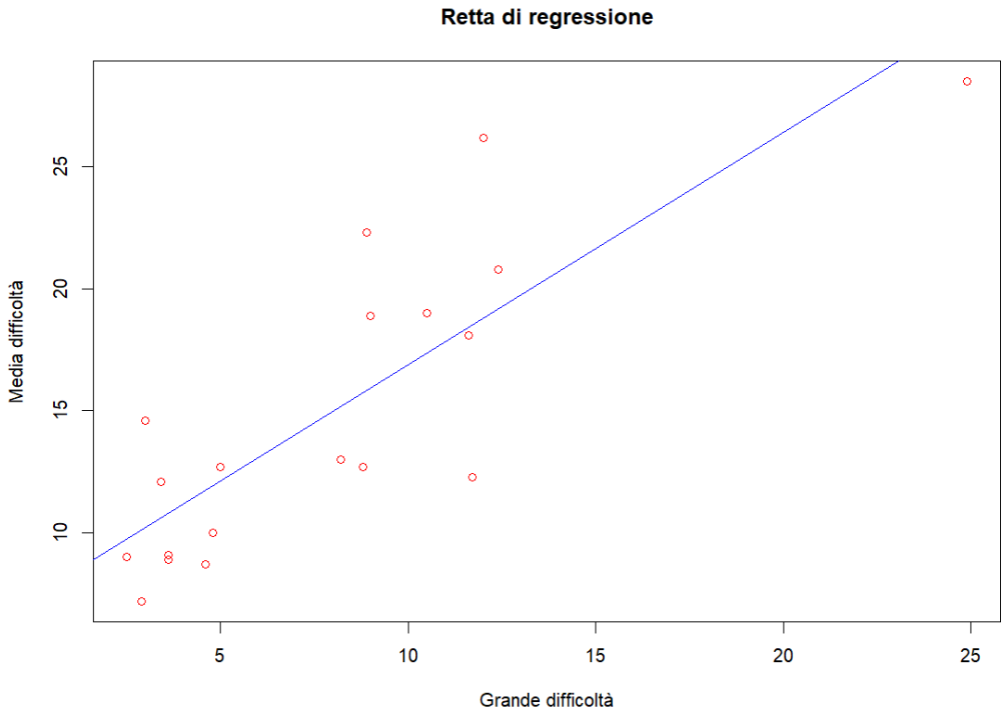


Figura 3.15: Regressione lineare semplice

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 uguale alla media campionaria dei valori osservati.

I residui sono quindi definiti come: $E_i = y_i \hat{y}_i = y_i(\alpha + \beta x_i)$.

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

```
1 #VALORI STIMATI
2
3 stimati<-fitted(lm(media_difficolta~grande_difficolta))
```

Da cui otteniamo.

```
       1        2        3        4        5        6        7        8        9       10       11       12       13
12.125807 11.744887 11.935347 15.935003 10.125979 10.602128 10.792588 10.792588  9.745059 10.221209 15.839773 18.506211 19.172820
      14       15       16       17       18       19
31.076559 17.363452 15.744543 15.173164 18.410981 18.791901
```

Per ottenere invece i valori residui usiamo la funzione **resid**:

```
#RESIDUI

residui<-resid(lm(media_difficolta~grande_difficolta))
residui
```

Da cui otteniamo:

```
       1        2        3        4        5        6        7        8        9       10       11       12
0.5741932 -3.0448872 -1.9353470  2.9649967 -2.9259787  1.4978717 -1.6925881 -1.8925881 -0.7450590  4.3787914  6.4602266 -6.2062109
      13       14       15       16       17       18       19
1.6271797 -2.5765592  1.6365480 -3.0445435 -2.1731640 -0.3109810  7.4080994
```

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:

```
#INDICI SUI RESIDUI

median(linear_model$residuals)
[1] -0.745059

var(linear_model$residuals)
[1] 12.23907

sd(linear_model$residuals)
[1] 3.498438
```

Vediamo ora delle possibili rappresentazioni grafiche.
La prima attraverso segmenti, produce il grafico 3.16:

```
#RAPPRESENTAZIONI GRAFICA DEI RESIDUI

plot(grande_difficolta, media_difficolta, main="Retta di regressione"
    ,
  xlab="Grande difficolta",
    ylab = "Media difficolta" , col="red" )

abline(lm(media_difficolta~grande_difficolta), col =" blue ")
stimati<-fitted(lm(media_difficolta~grande_difficolta))
```

```
9
10  segments(grande_difficolta, stimati, grande_difficolta,
11      media_difficolta, col="magenta")
```

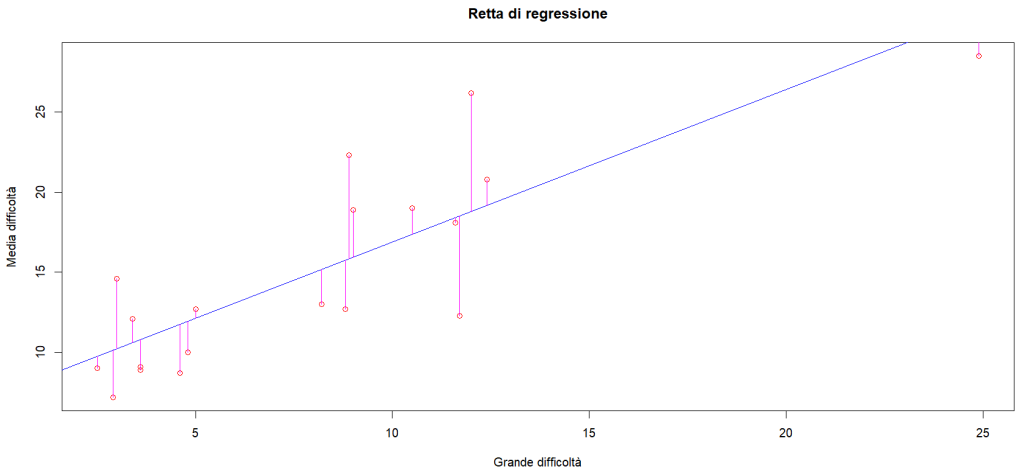


Figura 3.16: Rappresentazione grafica residui

La seconda, 3.17, attraverso un grafico dei residui, un modo per esaminare con più accuratezza il modo con cui la retta di regressione interpola i dati e di come i residui si dispongono intorno alla retta interpolante influenzandone la posizione, può essere ottenuti attraverso il diagramma dei residui, che è un grafico in cui i valori dei residui sono posti sull'asse delle ordinate e quelle della variabile indipendente sull'asse delle ascisse:

```
1  #GRAFICO DEI RESIDUI
2
3  residui<-resid(lm(media_difficolta~grande_difficolta))
4
5  plot(grande_difficolta, residui ,
6    main = "Diagramma dei residui",
7    xlab = "grande difficolt ",
8    ylab = " Residui ", pch =9 , col = "red")
9
10 abline (h =0 , col ="blue" , lty =2)
```

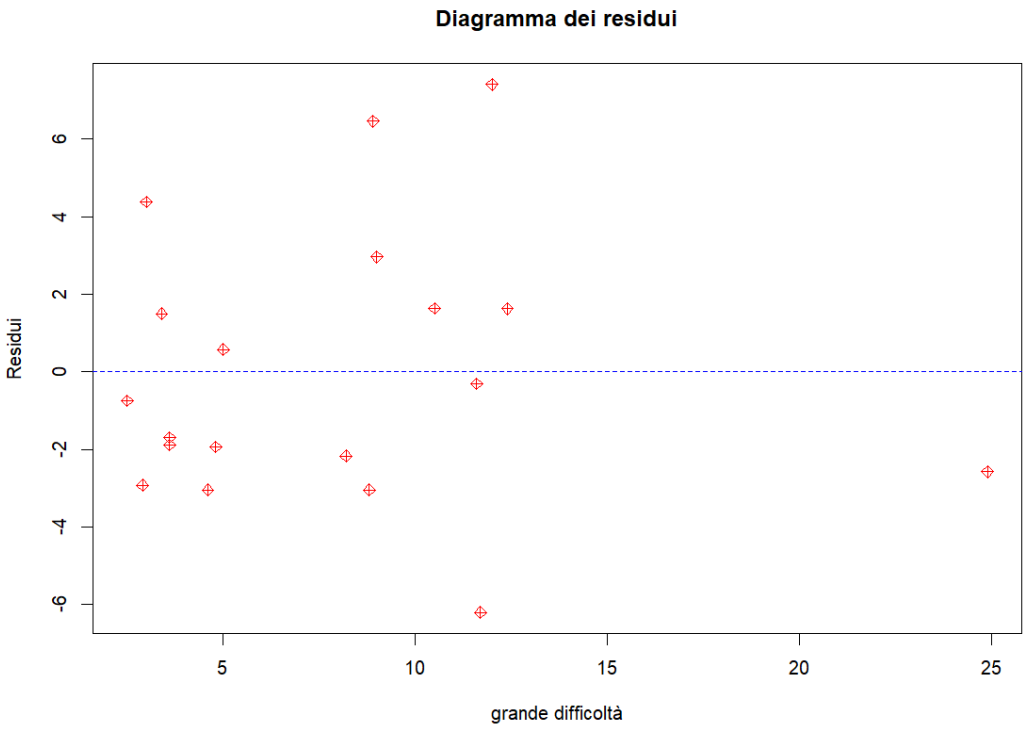


Figura 3.17: Diagramma dei residui

Il terzo, rappresentare i residui standardizzati rispetto ai valori stimati. I residui standardizzati sono definiti:

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

In R:

```
#RESIDUI STANDARDIZZATI

residui<-resid(lm(media_difficolta~grande_difficolta))
stimati<-fitted(lm(media_difficolta~grande_difficolta))

residuistandard <- residui/sd(residui)

plot(stimati, 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)
```

Che producono il grafico 3.18.

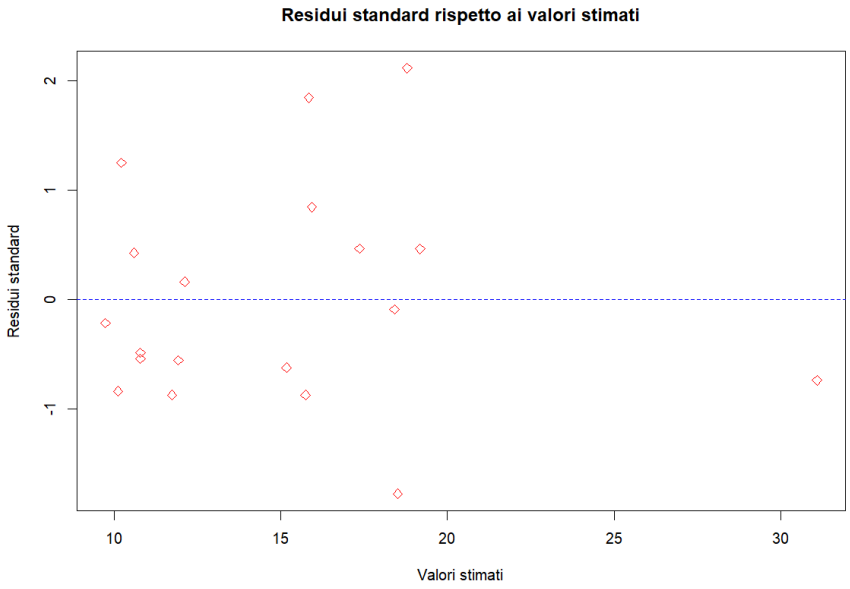


Figura 3.18: Diagramma dei residui standardizzati

Coefficiente di determinazione

Un altro indice molto utile, definita come il rapporto tra varianza dei valori stimati (con la retta di regressione) e varianza dei valori osservati:

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

Nel caso di regressione lineare semplice, il coefficiente di determinazione coincide con il quadrato del coefficiente di correlazione.
Calcoliamolo in R:

```
#COEFFICIENTE DI DETERMINAZIONE

(cor(grande_difficolta , media_difficolta))^2
[1] 0.6850054

summary(lm(media_difficolta~grande_difficolta))$r.square
[1] 0.6850054
```

3.2.3 Regressione non lineare

Spesso, osservando uno scatterplot, si nota che l'ipotesi di linearità di un modello non è accettabile poiché i dati sperimentali non evidenziano una correlazione di tipo lineare. In questo caso occorre ricorrere a modelli di regressione non lineare.
Attraverso alcune trasformazioni è possibile però linearizzare modelli che sembrano non lineari, questo ci permette di usare comunque un modello lineare.
Analizziamo la coppia di variabili precedente, analizziamolo e valutiamo il coefficiente di determinazione:

```
summary(lm(media_difficolta~grande_difficolta))$r.square
[1] 0.6850054
```

Consideriamo ora il modello non lineare $Y = \alpha + \beta X + \lambda X^2$, possiamo stimare i tre parametri attraverso regressione multipla.
In R:

```
regressione_nonlineare<-lm(media_difficolta~grande_difficolta+I(
    grande_difficolta^2))
regressione_nonlineare

Call:
lm(formula = media_difficolta ~ grande_difficolta + I(grande_
    difficolta^2))

Coefficients:
(Intercept)        grande_difficolta   I(grande_difficolta^2)
5.31611                    1.44098                  -0.02022
```

Su questi parametri e su questo modello non semplice rivalutiamo il coefficiente di correlazione:

```
summary(regressione_nonlineare)$r.square

[1] 0.7062573
```

Che ci restituisce, seppur di poco, un **risultato migliore** di quello ottenuto con la regressione semplice.

Visualizziamo ora la curva ottenuta sullo scatterplot:

```
#SCATTERPLORT REGRESSIONE NON LINEARE

plot(poca_difficolta,grande_difficolta,
main="Scatterplot",
xlab="poca difficolta",
ylab="grande difficolta" , col = "red")

alpha <- regressione_nonlineare$coefficients[[1]]
beta <- regressione_nonlineare$coefficients[[2]]
gamma <- regressione_nonlineare$coefficients[[3]]
curve(alpha+beta*x+gamma*x^2, add=TRUE, col = "green")
```

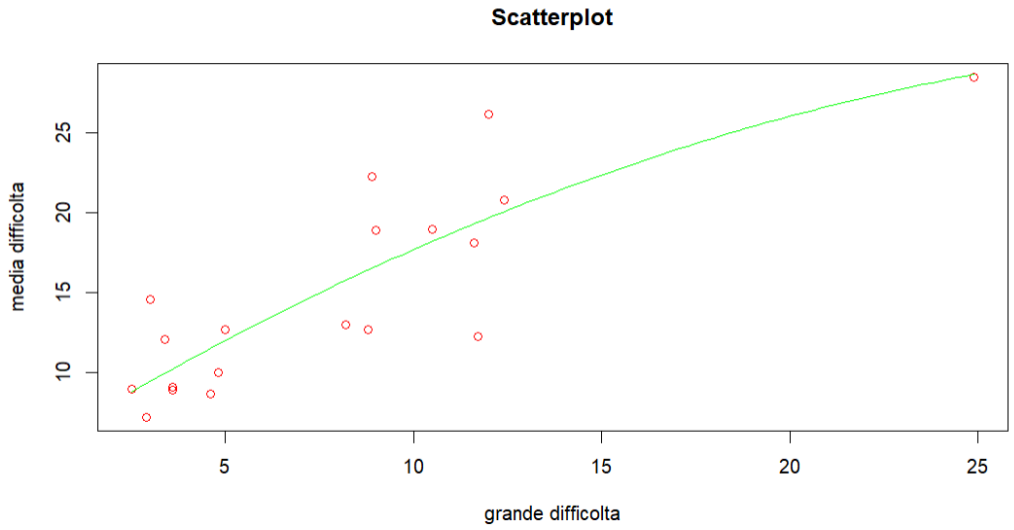


Figura 3.19: Regressione non lineare

Per completare l'analisi visualizziamo anche il discostamento dei valori:

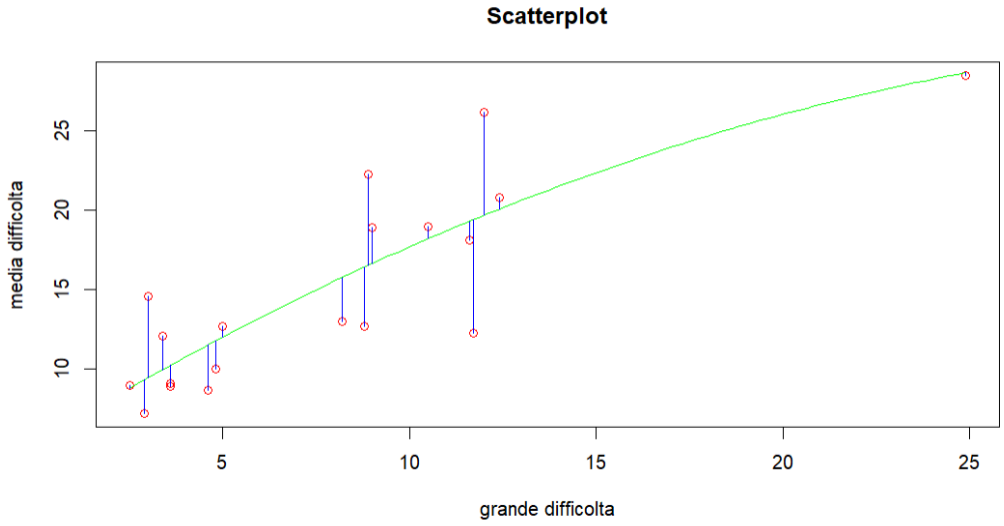


Figura 3.20: Regressione non lineare, residui

3.2.4 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.
Riportiamo quindi il coefficiente di correlazione e la covarianza per tutte le coppie della nostra matrice:

	Grande difficoltà	Media difficoltà	Poche difficoltà	Molta facilità
Grande difficoltà	29.349	27.949	-47.598	-9.763
Media difficoltà	27.949	38.855	-56.815	-10.078
Poche difficoltà	-47.598	-56.815	93.305	11.217
Molta facilità	-9.763	-10.078	11.217	8.672

Figura 3.21: Covarianza matrice

	Grande difficoltà	Media difficoltà	Poche difficoltà	Molta facilità
Grande difficoltà	1.0000000	0.8276505	-0.9095774	-0.6119746
Media difficoltà	0.8276505	1.0000000	-0.9436039	-0.5490503
Poche difficoltà	-0.9095774	-0.9436039	1.0000000	0.3943377
Molta facilità	-0.6119746	-0.5490503	0.3943377	1.0000000

Figura 3.22: Correlazione matrice

Notiamo dunque che nella maggior parte dei casi abbiamo una correlazione negativa, eccezion fatta per quella tra famiglie con medie difficoltà e grandi difficoltà.

Riportiamo anche lo scatterplot per coppie di variabili introdotto già nel quarto capitolo:

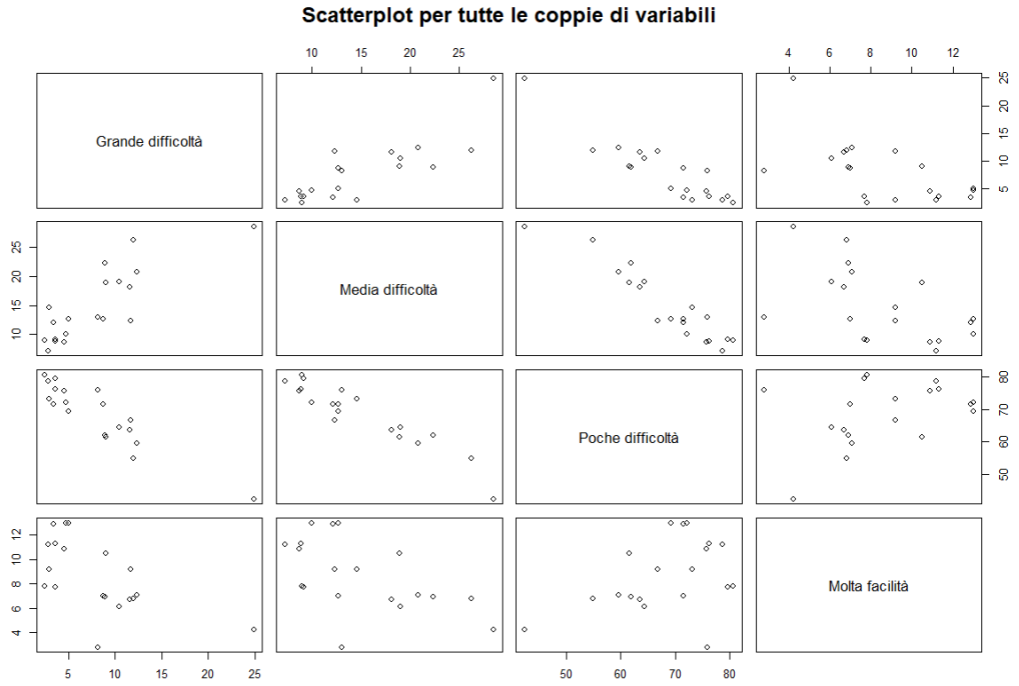


Figura 3.23: Scatterplot per coppie di variabili

Il modello di regressione lineare multipla con p variabili è esprimibile con la seguente equazione:
$Y = \alpha + \beta_1 1 X_1 + ... + \beta_p X_p.$, dove:

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

Anche in questo caso per ottenere i valori di α e β utilizziamo il metodo dei minimi quadrati che ci portano ad avere:

$\alpha = \bar{y}$ - $\beta_1 \bar{x_1}$ - $\beta_2 \bar{x_2}$ - ... - $\beta_p \bar{x_p}$.

Utilizziamo in R la funzione **lm(y ~ x1+...+xp)** per effettuare l'analisi di regressioni multiple.

```
#REGRESSIONE MULTIPLA

regressione_lineare_multipla<-lm(media_difficolta~
                    grande_difficolta+
                    poca_difficolta+
                    molta_facilita)
```

```
regressione_lineare_multipla

Call:
lm(formula = media_difficolta ~
      grande_difficolta+
      poca_difficolta+
      molta_facilita)

Coefficients:
(Intercept)  grande_difficolta     poca_difficolta     molta_facilita
99.7461           −0.9960             −0.9976             −0.9931
```

Avremo quindi $\alpha = 99.7461$ e i regressori $\beta_1 = -0.9960 \beta_2 = -0.9976 \beta_3 = -0.9931$. Notiamo che tutti i regressori sono negativi, questo significa che le altre variabili sono tutte legate negativamente alla percentuale delle famiglie con media difficoltà.

Residui

I residui mostrano di quanto si discostano i valori osservati da quelli stimati con la retta di regressione. Si definiscono come:
$E_i = y_i - \hat{y}_i = y_i - \alpha + \beta x_{i,1} + ... + \beta x_{i,p}$
Calcoliamo il vettore dei valori stimati attraverso la funzione **fitted()**:

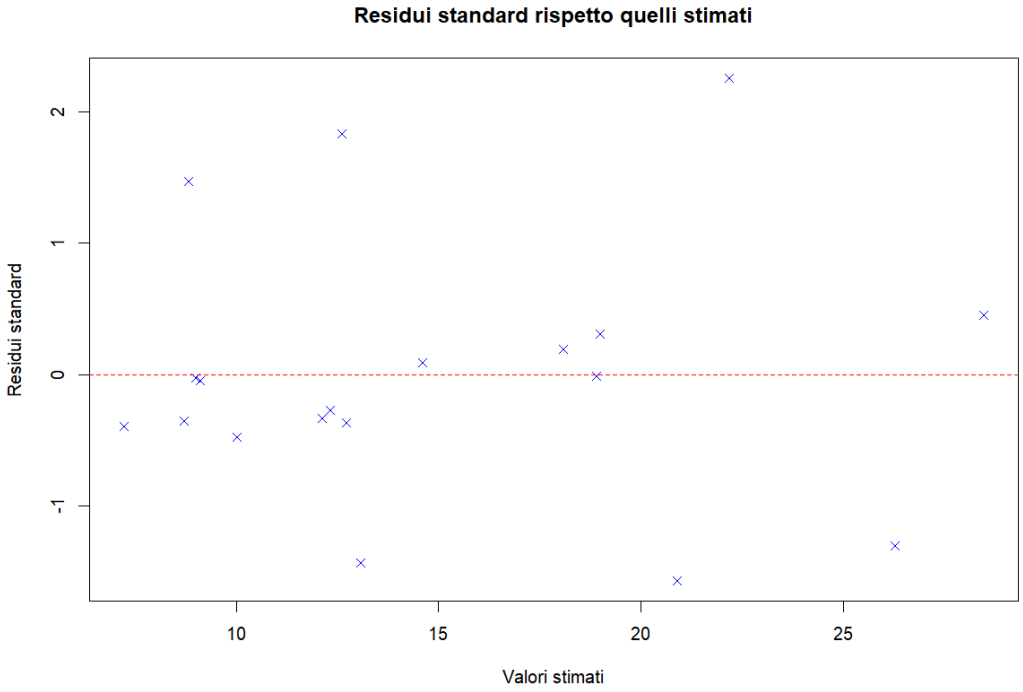
```
       1        2        3        4        5        6        7        8        9       10
12.720122  8.719519 10.026211 18.900641  7.221669 12.118495  9.102535  8.819456  9.001434 14.595246
      11       12       13       14       15       16       17       18       19
22.176430 12.314858 20.886110 28.475325 18.983032 12.599524 13.078450 18.089660 26.271280
```

Calcoliamo il vettore dei valori residui attraverso la funzione **resid()**:

```
           1            2            3            4            5            6            7
-0.0201220961 -0.0195192204 -0.0262112975 -0.0006412183 -0.0216689875 -0.0184953327 -0.0025346862
           8            9           10           11           12           13           14
 0.0805435631 -0.0014336723  0.0047537395  0.1235703146 -0.0148584903 -0.0861103773  0.0246746823
          15           16           17           18           19
 0.0169680732  0.1004757715 -0.0784500988  0.0103395824 -0.0712802493
```

Calcoliamo e visualizziamo infine i residui standardizzati:

```
          1           2           3           4           5           6           7           8           9
-0.36723122 -0.35622865 -0.47836004 -0.01170233 -0.39546222 -0.33754255 -0.04625840  1.46993190 -0.02616473
         10          11          12          13          14          15          16          17          18
 0.08675645  2.25517646 -0.27116964 -1.57152708  0.45031659  0.30966984  1.83369764 -1.43172587  0.18869890
         19
-1.30087506
```



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.

In R:

```
#COEFFICIENTE DI DETERMINAZIONE

summary(regressione_lineare_multipla)$r.square
[1] 0.9999227
```

Capitolo 4

Analisi dei cluster

L'analisi dei cluster è una metodologia che permette di raggruppare in sottoinsiemi, detti **cluster**, entità appartenenti ad un insieme più ampio.
Sia $I = I_1, ..., I_n$ un insieme di n individui appartenenti ad una popolazione ideale. Si assuma che esista un insieme di caratteristiche $C = C_1, ..., C_p$ che sono osservabili e sono possedute da ogni individuo in I. Il termine osservabile denota caratteristiche che danno origine a dati sia di tipo qualitativo che di tipo quantitativo, detti anche misure.
In generale, ciò che viene fatto, è porre in una matrice X, di cardinalità np, gli individui I e le misure C, dove denota il valore della misura della caratteristica j-esima relativa all'individuo I_i.
Il problema dell'analisi dei cluster consiste nel determinare m sottoinsiemi, detti cluster, di individui in I, con m intero minore di n, tali che I_i appartenga soltanto ad un unico sottoinsieme. Gli individui che sono assegnati allo stesso cluster sono detti simili, mentre gli individui che sono assegnati a differenti cluster sono detti dissimili.
Lo scopo è di distribuire le osservazioni in gruppi in modo tale che il grado di naturale associazione sia alto tra i membri dello stesso gruppo e basso tra i membri di gruppi diversi. In questo modo si otterrà quindi un'alta omogeneità all'interno dei gruppi e un'alta eterogeneità tra gruppi distinti.

Al fine di operare su dati uguali, non considerando l'unità di misura dunque, si raccomanda la standardizzazione di ogni variabile, utilizzando la media campionaria e la deviazione standard campionaria, tuttavia essendo il nostro dataset costituito da valori percentuali senza unità di misura questo non verrà effettuato.

4.1 Distanza e similarità

Per analizzare 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.
Una funzione a valori reali $d(X_i, X_j)$ è detta funzione distanza se e soltanto se essa soddisfa le seguenti condizioni:

- $d(X_i, X_j) = 0$ se e solo se $X_i = X_j$, con X_i e X_j in E_p;
- $d(X_i, X_j) >= 0$ per ogni X_i e X_j in E_p;
- $d(X_i, X_j) = d(X_j, X_i)$ per ogni X_i e X_j in E_p;
- $d(X_i, X_j) <= d(X_i, X_k) + d(X_k, X_j)$ per ogni X_i, X_j e X_k in E_p;

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 $n(n - 1)/2$ distanze.
In R per calcolare la matrice delle distanze si opera con il seguente comando:

```
#MATRICE DELLE DISTANZE

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

Il cui output (parziale) è il seguente:

```
                      Abruzzo    Molise  Campania    Puglia Basilicata  Calabria   Sicilia  Sardegna
Piemonte
Liguria
Lombardia
Trentino Alto-Adige
Veneto
Friuli-Venezia Giulia
Emilia-Romagna
Toscana
Umbria
Marche
Lazio
Abruzzo              0.000000
Molise              11.357376  0.000000
Campania            32.512152 22.798903  0.000000
Puglia               7.854935  5.557877 28.021777  0.000000
Basilicata           6.037384 14.919115 37.004459  9.758586   0.000000
Calabria            11.683749 19.043634 40.538254 13.580501   6.048140  0.000000
Sicilia              7.080960  4.908156 27.216539  1.737815  10.054352 14.285307  0.000000
Sardegna            18.457248  7.176350 18.295081 12.034534  21.712209 25.410234 11.894116  0.000000
```

Esistono varie metriche per calcolare la distanza, come ad esempio la distanza euclidea che è quella di default utilizzata dalla funzione **dist**. Altre metriche esistenti sono:

- Metrica del valore assoluto o Manhattan
- Metrica del massimo o di Chebychev
- Metrica di Minkowski
- Distanza di Camberra
- Distanza di Jaccard

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

- $s(X_i, X_i) = 1$;
- $0 <= s(X_i, X_j) <= 1$;
- $s(X_i, X_j) = s(X_j, X_i)$ per ogni X_i e X_j;

Interessante è che è sempre possibile trasformare una misura di distanza in una di similarità, ma non sempre è possibile il contrario.
Un possibile approccio è: $s_{ij} = 1/(1 + d_{ij})$.

4.1.2 Misure di non omogeneità tra cluster

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: $T = S + B$, in cui 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.
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: $trT = trS + trB$.
trT (tr è la misura di non omogeneità statistica) 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.
Una volta scelta la misura di distanza (o di similarità) si pone il problema di procedere alla scelta di un idoneo algoritmo di raggruppamento delle unità osservate. I metodi di raggruppamento e di ottimizzazione si distinguono in tre tipi:

- metodi di enumerazione completa;
- metodi gerarchici;
- metodi non gerarchici;

Le misure di non omogeneità statistiche sono utilizzate per valutare, fissato il numero di cluster, la bontà della suddivisione in cluster ottenuta con i vari metodi (di enumerazione completa, gerarchici, 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 tr della matrice B, o massimizzare tr della matrice S) per ogni possibile partizione dell'insieme totale di n individui in m cluster.

4.2 Metodi non gerarchici

A differenza dei metodi gerarchici, con i metodi non gerarchici possiamo ricollocare gli individui già classificati in un livello precedente dell'analisi. Quello che si vuole ottenere con questi metodi è una partizione unica degli n individui.
In molti metodi non gerarchici numero di cluster va precisato all'inizio dell'analisi, mentre in alcuni viene determinato nel corso dell'analisi stessa. 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.

- Step 1: fissare a priori il numero k di cluster specificando m punti di riferimento iniziali che inducono una prima partizione provvisoria;
- Step 2: considerare tutti gli individui e attribuire ciascuno di essi al cluster individuato dal punto di riferimento da cui ha distanza minore;
- Step 3: calcolare il centroide di ognuno dei k gruppi così ottenuti. Tali centroidi costituiscono i punti di riferimento per i nuovi cluster;
- Step 4: valutare la distanza di ogni unità da ogni centroide ottenuto al passo precedente. Se la distanza minima non è ottenuta in corrispondenza del centroide del gruppo di appartenenza, allora si procede a spostare l'individuo presso il cluster che ha il centroide più vicino;
- Step 5: ricalcolare i centroidi dei k gruppi così ottenuti;
- Step 6: ripetere il procedimento a partire dal punto (4) fino a che i centroidi non subiscono ulteriori modifiche rispetto all'iterazione precedente. Si procede così iterativamente a spostamenti successivi fino a che gli individui all'interno di ogni cluster non cambiano al ripetersi del procedimento;

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. Applichiamo dunque **k-means** alla nostra matrice come input il un numero di cluster pari a 5 e valutiamone i risultati. 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 masso di iterazioni è posto a 20.

Vediamo in R:

```
> kmeans<-kmeans(matrice_capacita_arrivare_fine_mese, centers=5, iter.max = 20, nstart=10)
> kmeans
K-means clustering with 5 clusters of sizes 1, 4, 6, 5, 3

Cluster means:
  Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1         24.900000         28.50000         42.40000       4.200000
2          4.050000         12.35000         71.57500      12.025000
3         10.733333         20.88333         61.01667       7.350000
4          3.440000          8.58000         78.22000       9.780000
5          9.566667         12.66667         71.43333       6.333333

Clustering vector:
             Piemonte               Liguria             Lombardia   Trentino Alto-Adige               Veneto Friuli-Venezia Giulia
                    2                     4                     2                     3                    4                    2
       Emilia-Romagna               Toscana                Umbria                Marche                Lazio               Abruzzo
                    4                     4                     4                     2                    3                    5
              Molise              Campania                Puglia            Basilicata             Calabria              Sicilia
                    3                     1                     3                     5                    5                    3
            Sardegna
                    3

Within cluster sum of squares by cluster:
[1]   0.00000  32.61500 128.38500  36.69600  69.84667
 (between_SS / total_SS =  91.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"
[9] "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:

```
kmeans$betweenss/kmeans$totss

[1] 0.9126601
```

Notiamo come per un numero di cluster $k = 5$ otteniamo una misura di non omogeneità soddisfacente, procediamo all'analisi dei cluster con metodi non gerarchici, per valutare **a parità di numero di cluster quale metodo risulta essere il migliore**.

4.3 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.

Molti metodi di analisi gerarchica sono caratterizzati da una struttura comune che si riflette in un algoritmo generale che può essere così esplicitato:

- **Step 1:** a partire dalla matrice X dei dati o dalla matrice scalata, considerare la matrice delle distanze D tra gli individui considerati come singoli cluster;

- **Step 2:** individuare la coppia di cluster meno distanti e raggruppare in un unico cluster i due cluster meno distanti; calcolare la distanza di questo nuovo cluster, da tutti gli altri gruppi già esistenti;

- **Step 3:** costruire una nuova matrice di distanza, la quale risulterà ridotta di una riga e di una colonna rispetto a quella precedente;

- **Step 4:** operare sulla matrice così ottenuta a partire dal passo 2 fino ad esaurire tutte le possibilità di raggruppamento;

- **Step 5:** rappresentare graficamente il processo di agglomerazione attraverso un dendrogramma;

L'analisi gerarchica di tipo agglomerativo viene effettuata in R attraverso la funzione **hclust(d, method = "complete")**. Dove d rappresenta un oggetto (che individua una struttura di similarità o distanza) creato tramite la funzione dist() e method seleziona il metodo gerarchico agglomerativo (di default complete).

Vediamo dunque nel dettaglio i vari metodi gerarchici che abbiamo presentato nel paragrafo precedente.

4.3.1 Metodo del legame singolo

In questo metodo la distanza tra i gruppi $G1$ (contenente $n1$ individui) e $G2$ (contenente $n2$ individui) è definita come la minima tra tutte le $n1n2$ distanze che si possono calcolare tra ogni individuo di $G1$ e ogni individuo di $G2$.

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:
$d_{(uv),z} = min(d_{uz}, d_{vz})$
Questo metodo ha il vantaggio di essere applicabile a 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.

Procediamo all'analisi tramite metodo del legame singolo in R, merge permette di visualizzare l'intero processo di clusterizzazione, mentre height indica la distanza a cui è avvenuta l'agglomerazione tra cluster:

```
1  #METODO DEL LEGAME SINGOLO
2
3  legame_singolo<-hclust(matrice_distanze,method="single")
4  legame_singolo$merge
5
6          [,1] [,2]
7  [1,]    -2    -8
```

```
[2,]   -7   -9
[3,]  -15  -18
[4,]   -3   -6
[5,]   -1    4
[6,]   -5    1
[7,]    2    6
[8,]    5    7
[9,]  -11    3
[10,]  -13    9
[11,]  -10    8
[12,]   -4   10
[13,]  -12  -16
[14,]  -17   13
[15,]   11   14
[16,]   12   15
[17,]  -19   16
[18,]  -14   17
```

Possiamo vedere ad esempio come all'inizio siano stati raggruppati gli individui -2 e -8, poi -7 e -9 ecc.
Vediamo la rappresentazione grafica tramite dendogramma:

```
plot(legame_singolo, hang=-1,
  xlab="Metodo gerarchico agglomerativo",
  sub="del legame singolo")

axis(side=4, at=round(c(0,legame_singolo$height),2))
abline(h=1.2, lty=2, col="red")

rect.hclust(legame_singolo, k=5, border="blue")
```

Per disegnare dei rettangoli intorno ai cluster in R usiamo **rect.hclust()**, per visualizzare il taglio del dendrogramma in corrispondenza di un salto nelle distanze si utilizza la funzione **abline()**.
Si ottiene cosi 4.1:

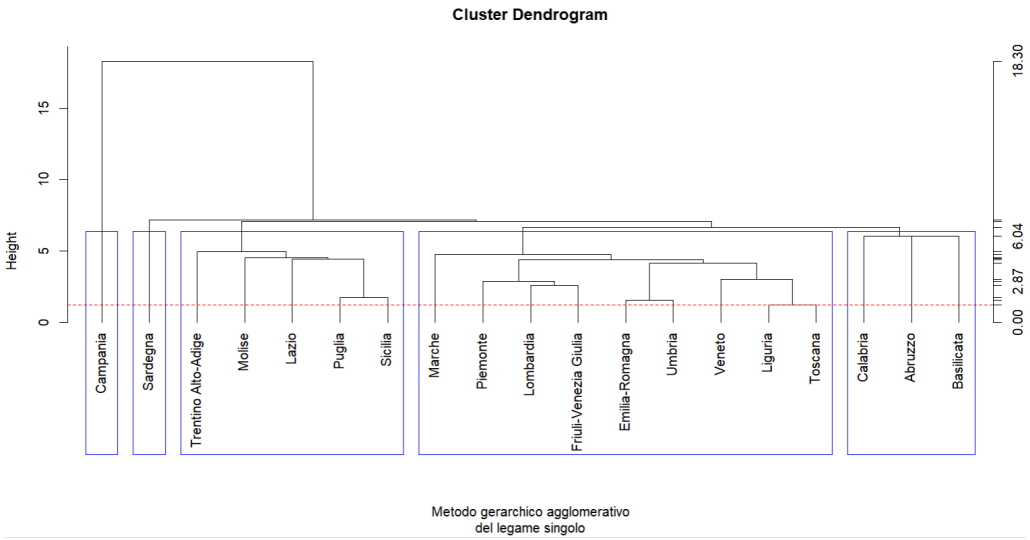


Figura 4.1: Dendogramma metodo del legame singolo

Se vogliamo visualizzare a quale cluster appartiene ogni individuo usiamo **cutree()**:

```
cutree(legame_singolo, k=5, h=NULL)

Piemonte          Liguria         Lombardia
    Trentino
1                   1               1               2

Veneto            Friuli    Emilia-Romagna
    Toscana
1                   1               1               1

Umbria            Marche             Lazio
    Abruzzo
1                   1               2               3

Molise          Campania           Puglia
    Basilicata
2                   4               2               3

Calabria          Sicilia         Sardegna
3                   2               5
```

Misure di sintesi per cluster

È possibile ricavare misure di sintesi, come la media campionaria, la varianza, la deviazione standard, ecc. sui singoli cluster, ottenuti tagliando il dendrogramma attraverso la funzione cutree(), utilizzando la funzione **aggregate()**.

```
#MEDIA LEGAME SINGOLO

taglio_singolo<-cutree(legame_singolo, k=5)
taglio_singolo_list<-list(taglio_singolo)
aggregate(matrice_capacita_arrivare_fine_mese, taglio_singolo_list,
    mean)

#VARIANZA LEGAME SINGOLO

aggregate(matrice_capacita_arrivare_fine_mese, taglio_singolo_list,
    var)

#DEVIAZIONE STANDARD LEGAME SINGOLO

aggregate(matrice_capacita_arrivare_fine_mese, taglio_singolo_list,
    sd)
```

```
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_singolo_list, mean)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1          3.711111         10.25556         75.26667      10.777778
2       2         10.480000         19.82000         62.24000       7.460000
3       3          9.566667         12.66667         71.43333       6.333333
4       4         24.900000         28.50000         42.40000       4.200000
5       5         12.000000         26.20000         54.90000       6.800000
>
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_singolo_list, var)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1         0.7986111        5.6027778         15.52000       4.459444
2       2         2.4070000        2.8970000          3.48800       3.028000
3       3         3.5033333        0.1233333         20.72333      10.573333
4       4                NA               NA               NA             NA
5       5                NA               NA               NA             NA
>
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_singolo_list, sd)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1         0.8936504        2.3670188         3.939543       2.111740
2       2         1.5514509        1.7020576         1.867619       1.740115
3       3         1.8717194        0.3511885         4.552289       3.251666
4       4                NA               NA               NA             NA
5       5                NA               NA               NA             NA
```

Figura 4.2: Indici metodo del legame singolo

Si può notare che alcuni risultati ottenuti sono NA, ciò indica che il cluster contiene solo un individuo e automaticamente la varianza e la deviazione standard non possono essere calcolate.

Misure di non omogeneità

Dopo aver effettuato il taglio, si è interessati a calcolare le misure di non omogeneità statistica relative all'insieme totale di individui (trT), ai singoli cluster ottenuti effettuando il taglio e alla somma delle loro misure di non omogeneità ($trS–within$) e alla misura di non omogeneità tra i cluster ($trB–between$).

Poiché per ogni fissata matrice X dei dati si ha che la trT è fissata, i cluster dovrebbero essere individuati in modo ma minimizzare la misura di non omogeneità statistica all'interno dei cluster e massimizzare la misura di non omogeneità statistica tra i gruppi.

Misure di non omogeneità statistica totale

Si definisce misura di non omogeneità statistica dell'insieme I di individui la tr matrice H_I:
Dove H_I indica la matrice statistica di non omogeneità per l'insieme I di individui,

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

di cardinalità pxp.
In R calcoliamola:

```
#MISURA NON OMEGENEITA STATISTICA

numero_righe<-nrow(matrice_capacita_arrivare_fine_mese)
trH<-(numero_righe-1)*
  sum(apply(matrice_capacita_arrivare_fine_mese,2, var))

trH
[1] 3063.237
```

In R utilizzando la funzione apply(X,2,var) è possibile calcolare la varianza campionaria delle colonne di una matrice.

La misura di non omogeneità totale trH è fissata per il dataset ottenuto, quindi, verrà utilizzata in seguito.

Siano $I = I_1, ..., I_{n1}$ e $J = J_1, ..., J_{n2}$ due cluster distinti di individui di una popolazione. La misura di non omogeneità statistica tra i cluster (between) può essere semplicemente calcolata come:

$trH_{I \cap J} = trH_{I \cup J}$-$trH_I$-$trH_j$

mentre la misura di non omogeneità nel cluster (within) può essere calcolata come

$trH_I + trH_J$

Applichiamo questi concetti ai nostri cluster in R:

```
#MISURA NON OMOGENEITA TOTALE

taglio_singolo<-cutree(legame_singolo, k=5, h=NULL)
num<-table(taglio_singolo)
taglio_singolo_list<-list(taglio_singolo)
agvar<-aggregate(matrice_capacita_arrivare_fine_mese, taglio_singolo_
    list, var)[, -1]

trH1_singolo<-(num [[1]]-1)*sum(agvar[1, ])
if(is.na(trH1_singolo))
   trH1_singolo<-0

trH2_singolo<-(num [[2]]-1)*sum(agvar[2, ])
  if(is.na(trH2_singolo))
  trH2_singolo<-0

trH3_singolo<-(num [[3]]-1)*sum(agvar[3, ])
if(is.na(trH3_singolo))
  trH3_singolo<-0

trH4_singolo<-(num [[4]]-1)*sum(agvar[4, ])
if(is.na(trH4_singolo))
    trH4_singolo<-0

trH5_singolo<-(num [[5]]-1)*sum(agvar[5, ])
if(is.na(trH5_singolo))
   trH5_singolo<-0

sum <- trH1_singolo+trH2_singolo+
      trH3_singolo+trH4_singolo+
```

```
30        trH5_singolo
31  trB <- trH - sum
32  trB/trH
33
34  [1]  0.8928671
```

La misura di non omogeneità statistica totale è $trH = 3063.237$, la misura di non omogeneità statistica all'interno dei gruppi (within) è pari a $sum = 328.1733$.

4.3.2 Metodo del 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 addensano 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.

Vediamo l'analisi in R, come nella sezione precedente:

```
1  #METODO DEL LEGAME COMPLETO
2
3  legame_completo<-hclust(matrice_distanze,method="complete")
4  legame_completo$merge
5
6        [,1] [,2]
7  [1,]   -2   -8
8  [2,]   -7   -9
```

```
     [3 ,]  -15  -18
     [4 ,]   -3   -6
     [5 ,]   -5    1
     [6 ,]   -1    4
     [7 ,]  -11  -13
     [8 ,]   -4    3
     [9 ,]    7    8
     [10 ,]  -12  -16
     [11 ,]    2    5
     [12 ,]  -10    6
     [13 ,]   10   12
     [14 ,]  -17   11
     [15 ,]  -19    9
     [16 ,]   13   14
     [17 ,]  -14   15
     [18 ,]   16   17
```

Dendogramma metodo del legame completo.

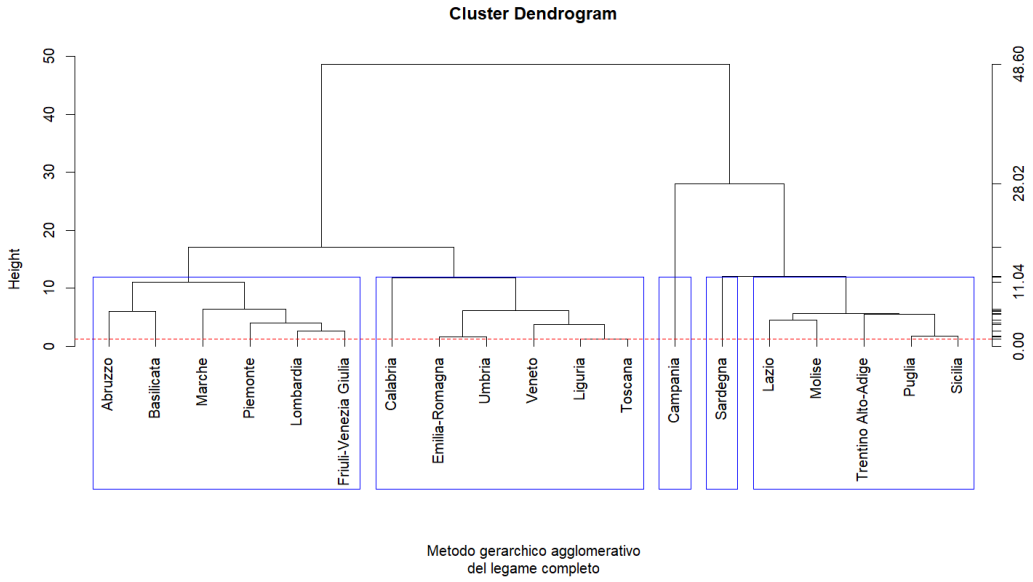


Figura 4.3: Dendogramma metodo del legame completo

Se vogliamo visualizzare a quale cluster appartiene ogni individuo:

```
cutree(legame_completo, k=5, h=NULL)

Pie          Lig          Lombardia          Trentino
1            2            1                  3

Friul        Emilia-Romagna          Toscana          Umbria
1            2                       2                2

Lazio        Abruzzo          Molise          Campania
3            1                3               4

Basilicata        Calabria          Sicilia          Sardegna
1                 2                 3                5

Veneto       Marche       Puglia
2            1            3
```

Misure di sintesi associate:

```
> #INDICI LEGAME COMPLETO
>
> taglio_completo<-cutree(legame_completo, k=5)
> taglio_completo_list<-list(taglio_completo)
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_completo_list, mean)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1          6.116667        12.400000         70.78333       10.716667
2       2          4.233333         9.316667         77.83333        8.616667
3       3         10.480000        19.820000         62.24000        7.460000
4       4         24.900000        28.500000         42.40000        4.200000
5       5         12.000000        26.200000         54.90000        6.800000
>
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_completo_list, var)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1         11.689667         2.176000         5.449667        6.721667
2       2          4.290667         3.749667         4.462667       10.885667
3       3          2.407000         2.897000         3.488000        3.028000
4       4                NA               NA               NA              NA
5       5                NA               NA               NA              NA
>
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_completo_list, sd)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1          3.419015         1.475127         2.334452        2.592618
2       2          2.071392         1.936406         2.112502        3.299343
3       3          1.551451         1.702058         1.867619        1.740115
4       4                NA               NA               NA              NA
5       5                NA               NA               NA              NA
```

Figura 4.4: Indici metodo del legame completo

Misure di non omogeneità:

```
#MISURA NON OMOGENEITA

> numero_righe<-nrow(matrice_capacita_arrivare_fine_mese)
> trH<-(numero_righe-1)*sum(apply(matrice_capacita_arrivare_fine_mese
    ,2, var))
> trH
[1] 3063.237

> taglio_completo<-cutree(legame_completo, k=5)
> taglio_completo_list<-list(taglio_completo)
> num<-table(taglio_completo)
>
>
> agvar<-aggregate(matrice_capacita_arrivare_fine_mese, taglio_
    completo_list, var)[, -1]
>
> trH1_completo<-(num [[1]]-1)*sum(agvar[1, ])
> if(is.na(trH1_completo))
+   trH1_completo<-0
>
> trH2_completo<-(num [[2]]-1)*sum(agvar[2, ])
> if(is.na(trH2_completo))
+   trH2_completo<-0
>
> trH3_completo<-(num [[3]]-1)*sum(agvar[3, ])
> if(is.na(trH3_completo))
+   trH3_completo<-0
>
> trH4_completo<-(num [[4]]-1)*sum(agvar[4, ])
> if(is.na(trH4_completo))
+   trH4_completo<-0
>
> trH5_completo<-(num [[5]]-1)*sum(agvar[5, ])
> if(is.na(trH5_completo))
+   trH5_completo<-0
>
> sum <- trH1_completo+trH2_completo+trH3_completo+trH4_completo+trH5
    _completo
> trB <- trH - sum
> trB/trH

[1] 0.9038898
```

La misura di non omogeneità statistica totale è $trH = 3063.237$, la misura di non omogeneità statistica all'interno dei gruppi (within) è pari a $sum = 294.4083$.

4.3.3 Metodo del legame medio

In questo metodo la distanza tra i gruppi G_1 e G_2 è definita come la media aritmetica delle distanze tra tutte le coppie di unità che compongono i due gruppi.

Vediamo l'analisi in R, come nella sezione precedente:

```
#METODO DEL LEGAME MEDIO

legame_medio<-hclust(matrice_distanze,method="average")
legame_medio$merge


      [,1] [,2]
 [1,]   -2   -8
 [2,]   -7   -9
 [3,]  -15  -18
 [4,]   -3   -6
 [5,]   -5    1
 [6,]   -1    4
 [7,]  -11  -13
 [8,]    3    7
 [9,]    2    5
[10,]   -4    8
[11,]  -10    6
[12,]  -12  -16
[13,]    9   11
[14,]  -17   12
[15,]  -19   10
[16,]   13   14
[17,]   15   16
[18,]  -14   17
```

Dendogramma metodo del legame medio.

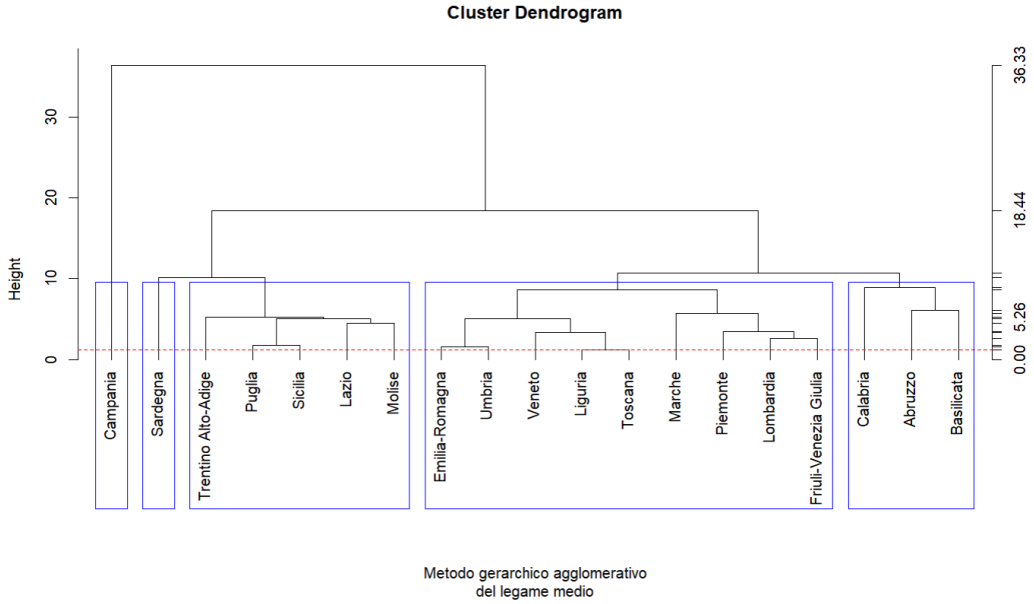


Figura 4.5: Dendogramma metodo del legame medio

Se vogliamo visualizzare a quale cluster appartiene ogni individuo:

```
1
2  cutree(legame_medio, k=5, h=NULL)
3
4   Piemonte          Liguria        Lombardia     Trentino
5          1                1                1            2
6
7     Friuli Emilia-Romagna          Toscana           Umbria
8          1                1                1            1
9
10     Lazio          Abruzzo           Molise         Campania
11         2                3                2            4
12
13 Basilicata         Calabria          Sicilia         Sardegna
14         3                3                2            5
15
16    Veneto          Marche           Puglia
17         1                1                2
```

Misure di sintesi associate:

```
> #INDICI LEGAME COMPLETO
>
> taglio_medio<-cutree(legame_medio, k=5)
> taglio_medio_list<-list(taglio_medio)
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_medio_list, mean)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1          3.711111         10.25556         75.26667      10.777778
2       2         10.480000         19.82000         62.24000       7.460000
3       3          9.566667         12.66667         71.43333       6.333333
4       4         24.900000         28.50000         42.40000       4.200000
5       5         12.000000         26.20000         54.90000       6.800000
>
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_medio_list, var)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1          0.7986111         5.6027778         15.52000       4.459444
2       2          2.4070000         2.8970000          3.48800       3.028000
3       3          3.5033333         0.1233333         20.72333      10.573333
4       4                 NA                NA               NA             NA
5       5                 NA                NA               NA             NA
>
> aggregate(matrice_capacita_arrivare_fine_mese, taglio_medio_list, sd)
  Group.1 Grande difficoltà Media difficoltà Poche difficoltà Molta facilità
1       1          0.8936504         2.3670188         3.939543       2.111740
2       2          1.5514509         1.7020576         1.867619       1.740115
3       3          1.8717194         0.3511885         4.552289       3.251666
4       4                 NA                NA               NA             NA
5       5                 NA                NA               NA             NA
```

Figura 4.6: Indici metodo del legame medio

Misure di non omogeneità:

```
#MISURA NON OMOGENEITA

> numero_righe<-nrow(matrice_capacita_arrivare_fine_mese)
> trH<-(numero_righe-1)*sum(apply(matrice_capacita_arrivare_fine_mese
    ,2, var))
> trH
[1] 3063.237

> taglio_medio<-cutree(legame_medio, k=5)
> taglio_medio_list<-list(taglio_medio)
> num<-table(taglio_medio)

> agvar<-aggregate(matrice_capacita_arrivare_fine_mese, taglio_medio_
    list, var)[, -1]

> trH1_medio<-(num [[1]]-1)*sum(agvar[1, ])
> if(is.na(trH1_medio))
+   trH1_medio<-0

> trH2_medio<-(num [[2]]-1)*sum(agvar[2, ])
> if(is.na(trH2_medio))
+   trH2_medio<-0

> trH3_medio<-(num [[3]]-1)*sum(agvar[3, ])
```

```
> if(is.na(trH3_medio))
+    trH3_medio<-0

> trH4_medio<-(num [[4]]-1)*sum(agvar[4, ])
> if(is.na(trH4_medio))
+    trH4_medio<-0

> trH5_medio<-(num [[5]]-1)*sum(agvar[5, ])
> if(is.na(trH5_medio))
+    trH5_medio<-0

> sum <- trH1_medio+trH2_medio+trH3_medio+trH4_medio+trH5_medio
> trB <- trH - sum
> trB/trH

[1] 0.8928671
```

La misura di non omogeneità statistica totale è $trH = 3063.237$, la misura di non omogeneità statistica all'interno dei gruppi (within) è pari a $sum = 328.1733$.

4.3.4 Metodo del centroide

In questo metodo si 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 se piccoli gruppi. 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.

Vediamo l'analisi in R, come nella sezione precedente:

```
#METODO DEL CENTROIDE

matrice_distanze_quadrata<-matrice_distanze^2
metodo_centroide<-hclust(matrice_distanze_quadrata, method="centroid")
metodo_centroide$merge

     [,1] [,2]
[1,]   -2   -8
[2,]   -7   -9
```

```
10      [3 ,]   -15   -18
11      [4 ,]    -3    -6
12      [5 ,]    -1     4
13      [6 ,]    -5     1
14      [7 ,]   -11   -13
15      [8 ,]     3     7
16      [9 ,]    -4     8
17      [10 ,]    2     6
18      [11 ,]  -10     5
19      [12 ,]  -12   -16
20      [13 ,]   11    12
21      [14 ,]   10    13
22      [15 ,]  -17    14
23      [16 ,]  -19     9
24      [17 ,]   15    16
25      [18 ,]  -14    17
```

Dendogramma metodo del centroide.

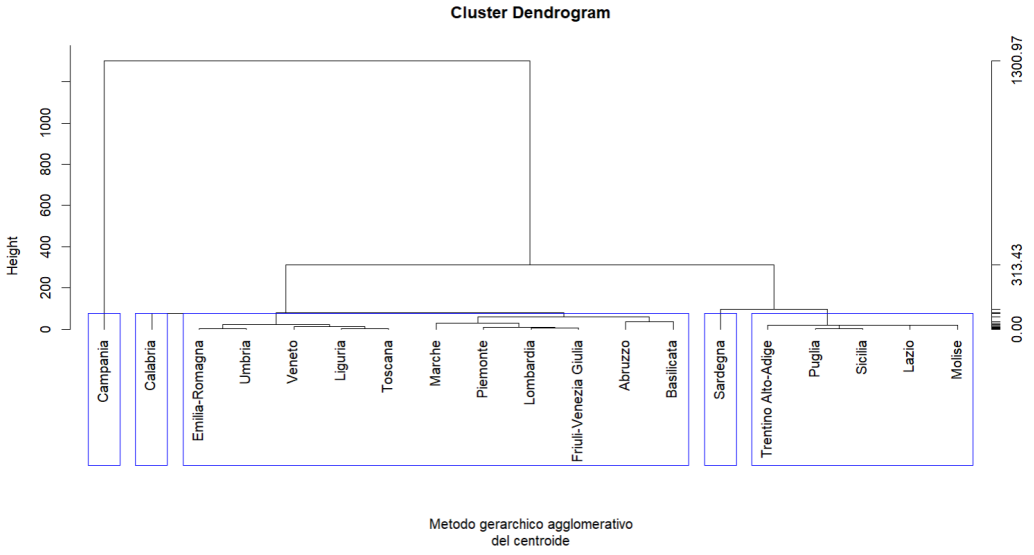


Figura 4.7: Dendogramma metodo del centroide

Se vogliamo visualizzare a quale cluster appartiene ogni individuo:

```
cutree(metodo_centroide, k=5, h=NULL)

Piemonte          Liguria         Lombardia         Trentino
1                 1               1                 2

Friuli     Emilia-Romagna         Toscana           Umbria
1                 1               1                 1

Lazio             Abruzzo         Molise            Campania
2                 1               2                 3

Basilicata        Calabria        Sicilia           Sardegna
1                 4               2                 5

Veneto        Marche        Puglia
1         1           2
```

Misure di non omogeneità:

```
#MISURA NON OMOGENEITA

> numero_righe<-nrow(matrice_capacita_arrivare_fine_mese)
> trH<-(numero_righe-1)*sum(apply(matrice_capacita_arrivare_fine_mese
    ,2, var))
> trH
[1] 3063.237
>
> taglio_centroide<-cutree(metodo_centroide, k=5)
> taglio_centroide_list<-list(taglio_centroide)
> num<-table(taglio_centroide)
>
> agvar<-aggregate(matrice_capacita_arrivare_fine_mese, taglio_
    centroide_list, var)[, -1]
>
> trH1_centroide<-(num [[1]]-1)*sum(agvar[1, ])
> if(is.na(trH1_centroide))
+   trH1_centroide<-0
>
> trH2_centroide<-(num [[2]]-1)*sum(agvar[2, ])
> if(is.na(trH2_centroide))
+   trH2_centroide<-0
>
> trH3_centroide<-(num [[3]]-1)*sum(agvar[3, ])
```

```
> if(is.na(trH3_centroide))
+   trH3_centroide<-0
>
> trH4_centroide<-(num [[4]]-1)*sum(agvar[4, ])
> if(is.na(trH4_centroide))
+   trH4_centroide<-0
>
> trH5_centroide<-(num [[5]]-1)*sum(agvar[5, ])
> if(is.na(trH5_centroide))
+   trH5_centroide<-0
>
> sum <- trH1_centroide+trH2_centroide+trH3_centroide+trH4_centroide+
    trH5_centroide
> trB <- trH - sum
> trB/trH

[1] 0.8606964
```

La misura di non omogeneità statistica totale è $trH = 3063.237$, la misura di non omogeneità statistica all'interno dei gruppi (within) è pari $sum = 426.72$.

4.3.5 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.
Anche in questo caso bisogna considerare la distanza al quadrato.

Vediamo l'analisi in R, come nella sezione precedente:

```
#METODO DELLA MEDIANA


matrice_distanze_quadrata<-matrice_distanze^2
metodo_mediana<-hclust(matrice_distanze_quadrata, method="median")
metodo_mediana$merge

     [,1] [,2]
[1,]   -2   -8
[2,]   -7   -9
[3,]  -15  -18
[4,]   -3   -6
```

```
13      [5 ,]   -1   4
14      [6 ,]   -5   1
15      [7 ,]  -11  -13
16      [8 ,]    3   7
17      [9 ,]   -4   8
18      [10 ,]   2   6
19      [11 ,] -10   5
20      [12 ,] -12  -16
21      [13 ,]  11   12
22      [14 ,] -17   13
23      [15 ,]  10   14
24      [16 ,] -19   9
25      [17 ,]  15   16
26      [18 ,] -14   17
```

Dendogramma metodo del centroide.

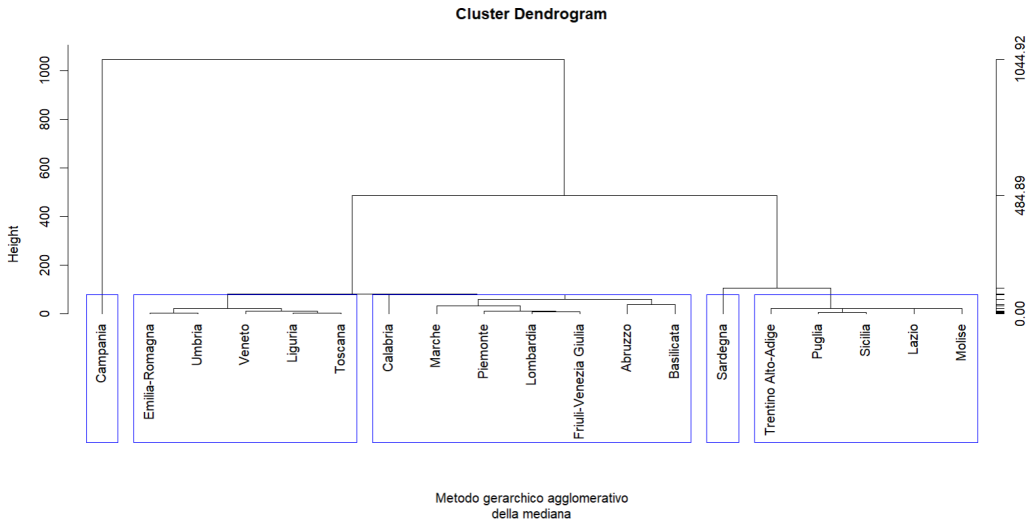


Figura 4.8: Dendogramma metodo della mediana

Se vogliamo visualizzare a quale cluster appartiene ogni individuo:

```
cutree(metodo_mediana, k=5, h=NULL)

Piemonte          Liguria          Lombardia     Trentino
1                 2                1             3


Friuli            Emilia-Romagna   Toscana       Umbria
1                 2                2             2

Lazio             Abruzzo          Molise        Campania
3                 1                3             4

Basilicata        Calabria         Sicilia       Sardegna
1                 1                3             5

Veneto            Marche      Puglia
2         1       3
```

Misure di non omogeneità:

```
#MISURA NON OMOGENEITA

> numero_righe<-nrow(matrice_capacita_arrivare_fine_mese)
> trH<-(numero_righe-1)*sum(apply(matrice_capacita_arrivare_fine_mese
    ,2, var))
> trH
[1] 3063.237

> taglio_mediana<-cutree(metodo_mediana, k=5)
> taglio_mediana_list<-list(taglio_mediana)
> num<-table(taglio_mediana)
>
> agvar<-aggregate(matrice_capacita_arrivare_fine_mese, taglio_
    mediana_list, var)[, -1]
>
> trH1_mediana<-(num [[1]]-1)*sum(agvar[1, ])
> if(is.na(trH1_mediana))
+   trH1_mediana<-0
>
> trH2_mediana<-(num [[2]]-1)*sum(agvar[2, ])
> if(is.na(trH2_mediana))
+   trH2_mediana<-0
>
```

```
> trH3_mediana<-(num [[3]]-1)*sum(agvar[3, ])
> if(is.na(trH3_mediana))
+   trH3_mediana<-0
>
> trH4_mediana<-(num [[4]]-1)*sum(agvar[4, ])
> if(is.na(trH4_mediana))
+   trH4_mediana<-0
>
> trH5_mediana<-(num [[5]]-1)*sum(agvar[5, ])
> if(is.na(trH5_mediana))
+   trH5_mediana<-0
>
> sum <- trH1_mediana+trH2_mediana+trH3_mediana+trH4_mediana+trH5_
    mediana
> trB <- trH - sum
> trB/trH

[1] 0.9039087
```

La misura di non omogeneità statistica totale è $trH = 3063.237$, la misura di non omogeneità statistica all'interno dei gruppi (within) è pari a$sum = 294.3503$.

TABELLA RIASSUNTIVA ANALISI DEI CLUSTER

METODO	MISURA
K-means	0.9126601
Legame singolo	0.8928671
Legame completo	0.9038898
Legame medio	0.8928671
Centroide	0.8606964
Mediana	0.9039087

I metodi migliori, a parità di numero di cluster, risultano essere quello del k-means, legame completo e mediana.