PROGETTO DI STATISTICA E ANALISI DEI DATI

Anno Accademico 2019-2020

Docente:
Prof.ssa Amelia Giuseppina Nobile

Studente:
Francesca Festa
0522500745

Indice

Introduzione

Il seguente progetto è formato da due parti:

1. la prima parte consiste di un'indagine statistica, in cui vengono analizzati i dati forniti dall'applicativo web dell'ISTAT, per ottenerne una visione più chiara e dettagliata, con l'ausilio di tabelle e grafici statistici;

2. la seconda parte è incentrata sull'inferenza statistica, il cui scopo è quello di trarre conclusioni su di una popolazione in base al campione estratto; viene quindi studiata una popolazione descritta da una variabile aleatoria osservabile X la cui funzione di distribuzione ha una forma nota ma contiene uno o più parametri non noti.

Capitolo 1

Prima parte

L'indagine campionaria "Aspetti della vita quotidiana" fa parte di un sistema integrato di indagini sociali e rileva le informazioni fondamentali circa la vita quotidiana degli individui e delle famiglie. Tutte le informazioni che vengono raccolte permettono di conoscere le abitudini dei cittadini. Per quanto riguarda abitudini alimentari diversi sono gli aspetti interessanti, come ad esempio, consumo di bevande alcoliche, quantitativi di carne/pesce/latticini/verdure assunti, indice corporeo, ecc. L'Istituto nazionale di statistica Italiano (ISTAT) ha condotto una serie di indagini per scoprire quali sono le abitudini alimentari più diffuse.

1.1 Dataset

Il dataset preso in considerazione è formato da una tabella con 20 regioni italiane come righe e 9 diversi tipi di abitudini alimentari come colonne. Nel dataset i dati fanno riferimento a 100 persone, con le stesse caratteristiche, di 3 anni e più, per abitudine alimentare nell'ultimo anno. Il periodo preso in considerazione è l'anno 2017.
Gli scopi di questa analisi sono quelli di:

- individuare regioni che hanno la maggioranza di un tipo di abitudine alimentare;
- individuare relazioni tra i diversi tipi di abitudini alimentari;
- individuare la distribuzione delle abitudini alimentari nelle diverse regioni;
- individuare eventuali anomalie e valori fuori dal range accettabile;
- dividere le regioni in gruppi in base al tipo di abitudine alimentare.

È possibile osservare il dataset in Figura 1.

Tipo abitudine alimentare	colazione adeguata	colazione con latte	pasto in casa	pasto in mensa	pasto al ristorante	pasto al bar	pasto sul posto di lavoro	pasto principale pranzo	pasto principale cena
Territorio									
Piemonte	81,2	40,6	68,1	9,1	2,7	3,6	9,7	61,7	29
Valle d'Aosta	79	38,7	63,4	11,5	5,2	3,4	8,3	69,1	21
Liguria	85,4	47,3	69,2	8	3,4	3,9	9,4	62,2	28,9
Lombardia	81,3	39,7	61,8	12,4	5,2	2,4	11,1	59,1	29,3
Trentino Alto Adige	84,5	38,9	67,8	11,5	6,8	1,3	6,5	75,1	13,2
Veneto	83,3	39,3	69,3	9,3	4,1	2,2	8,3	67,3	22,5
Friuli-Venezia Giulia	82,7	40,6	66,9	11,3	3,9	1,4	7,2	63,7	25,3
Emilia-Romagna	86,2	43,9	69,2	10,2	3	2,1	9,5	62,8	26,7
Toscana	86,8	48,4	71,8	8,8	2,3	2,8	8,2	61,9	26,1
Umbria	87,4	49,6	79,3	4,4	2,2	1,2	7,3	69,9	18,7
Marche	86	47,2	76,7	5,8	2,2	1,5	7,4	69,1	17,2
Lazio	83,2	44,9	65,8	9,4	2,1	4,2	11,1	56,5	30,8
Abruzzo	84,2	41,9	83,2	4,4	1,9	0,8	5,3	72,8	13
Molise	80,8	46,5	83,9	5,3	1,2	0,4	5,4	81,4	11,8
Campania	74,4	39,6	79,3	3,9	1,3	0,7	8,6	71,5	18,8
Puglia	79,9	48,8	87	3,2	0,7	0,7	3,3	82,8	9,1
Basilicata	78,3	40,8	84,3	4,4	0,7	0,6	5	81,5	9
Calabria	78,9	41,1	84,9	3,2	1	0,6	5,5	75,1	12,1
Sicilia	76,5	37,2	84,8	2,3	1,1	1,1	4,9	73,2	13,6
Sardegna	78,4	41,9	81,1	5,3	0,7	0,7	7,6	75,5	14,1

Per una maggiore chiarezza, i nomi delle colonne saranno rinominati, rispettivamente, nel seguente modo:

- colazione adeguata
- colazione con latte
- casa
- mensa
- ristorante
- bar
- lavoro
- pranzo
- cena

1.1.1 Introduzione all'analisi dei dati

Prima di iniziare l'analisi, viene definita la matrice sulla quale si opererà in seguito. Con il comando **cbind()**, è possibile creare matrici componendo vettori della stessa lunghezza e matrici delle stesse dimensioni. Questa funzione usa i vettori per creare le colonne. Con il comando **rownames** è possibile rinominare le righe della matrice; analogamente **colnames** permette di rinominare le colonne.

```
> matriceAbitudiniAlimentari <- cbind(
+ c(81.2,79,85.4,81.3,84.5,83.3,82.7,86.2,86.8,87.4,86,
+ 83.2,84.2,80.8,74.4,79.9,78.3,78.9,76.5,78.4),
+ c(40.6,38.7,47.3,39.7,38.9,39.3,40.6,43.9,48.4,49.6,47.2,
+ 44.9,41.9,46.5,39.6,48.8,40.8,41.1,37.2,41.9),c(68.1,63.4,69.2,61.8,67.8,69.3,66.9,69.2,71.8,
+ 79.3,76.7,65.8,83.2,83.9,79.3,87,84.3,84.9,84.8,81),c(9.1,11.5,8,12.4,11.5,9.3,11.3,10.2,8.8,
+ 4.4,5.8,9.4,4.4,5.3,3.9,3.2,4.4,3.2,2.3,5.3),
+ c(2.7,5.2,3.4,5.2,6.8,4.1,3.9,3,2.3,2.2,2.2,
+ 2.1,1.9,1.2,1.3,0.7,0.7,1,1.1,0.7),c(3.6,3.4,3.9,2.4,1.3,2.2,1.4,2.1,2.8,1.2,
+ 1.5,4.2,0.8,0.4,0.7,0.7,0.6,0.6,1.1,0.7),c(9.7,8.3,9.4,11.1,6.5,8.3,7.2,9.5,8.2,7.3,
+ 7.4,11.1,5.3,5.4,8.6,3.3,5,5.5,4.9,7.6),c(61.7,69.1,62.2,59.1,75.1,67.3,63.7,62.8,61.9,69.9,69.1,
+ 56.5,72.8,81.4,71.5,82.8,81.5,75.1,73.2,75.5),
+ c(29,21,28.9,29.3,13.2,22.5,25.3,26.7,26.1,18.7,
+ 17.2,30.8,13,11.8,18.8,9.1,9,12.1,13.6,14.1))
> rownames(matriceAbitudiniAlimentari) <- c("Piemonte", "Valle_d_Aosta", "Liguria", "Lombardia",
+  "Trentino_Alto_Adige", "Veneto", "Friuli_Venezia_Giulia", "Emilia_Romagna",
+  "Toscana","Umbria","Marche","Lazio",
+  "Abruzzo","Molise","Campania","Puglia",
+  "Basilicata","Calabria","Sicilia","Sardegna")
>
> colnames(matriceAbitudiniAlimentari) <- c("colazione_adeguata", "colazione_con_latte", "casa", "mensa",
+  "ristorante", "bar", "lavoro", "pranzo", "cena")
```

A partire dalla matrice *matriceAbitudiniAlimentari*, è possibile creare dei vettori contenenti gli elementi delle singole colonne. Per una questione di praticità si è preferito definire le colonne nel seguente modo:

```
> colazione_adeguata = matriceAbitudiniAlimentari[,1]
> colazione_adeguata
             Piemonte         Valle_d_Aosta               Liguria
                 81.2                  79.0                  85.4
            Lombardia    Trentino_Alto_Adige                Veneto
                 81.3                  84.5                  83.3
Friuli_Venezia_Giulia       Emilia_Romagna               Toscana
                 82.7                  86.2                  86.8
               Umbria                Marche                 Lazio
                 87.4                  86.0                  83.2
              Abruzzo                Molise              Campania
                 84.2                  80.8                  74.4
               Puglia           Basilicata              Calabria
                 79.9                  78.3                  78.9
              Sicilia             Sardegna
                 76.5                  78.4
> colazione_con_latte = matriceAbitudiniAlimentari[,2]
> colazione_con_latte
             Piemonte         Valle_d_Aosta               Liguria
                 40.6                  38.7                  47.3
            Lombardia    Trentino_Alto_Adige                Veneto
                 39.7                  38.9                  39.3
Friuli_Venezia_Giulia       Emilia_Romagna               Toscana
                 40.6                  43.9                  48.4
               Umbria                Marche                 Lazio
                 49.6                  47.2                  44.9
              Abruzzo                Molise              Campania
                 41.9                  46.5                  39.6
               Puglia           Basilicata              Calabria
                 48.8                  40.8                  41.1
              Sicilia             Sardegna
                 37.2                  41.9
```

```
> casa = matriceAbitudiniAlimentari[,3]
> casa
             Piemonte         Valle_d_Aosta               Liguria
                 68.1                  63.4                  69.2
            Lombardia    Trentino_Alto_Adige               Veneto
                 61.8                  67.8                  69.3
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                 66.9                  69.2                  71.8
               Umbria               Marche                Lazio
                 79.3                  76.7                  65.8
             Abruzzo               Molise             Campania
                 83.2                  83.9                  79.3
              Puglia           Basilicata             Calabria
                 87.0                  84.3                  84.9
             Sicilia            Sardegna
                 84.8                  81.0
  '

> mensa= matriceAbitudiniAlimentari[,4]
> mensa
             Piemonte         Valle_d_Aosta               Liguria
                  9.1                  11.5                   8.0
            Lombardia    Trentino_Alto_Adige               Veneto
                 12.4                  11.5                   9.3
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                 11.3                  10.2                   8.8
               Umbria               Marche                Lazio
                  4.4                   5.8                   9.4
             Abruzzo               Molise             Campania
                  4.4                   5.3                   3.9
              Puglia           Basilicata             Calabria
                  3.2                   4.4                   3.2
             Sicilia            Sardegna
                  2.3                   5.3
  '

> ristorante= matriceAbitudiniAlimentari[,5]
> ristorante
             Piemonte         Valle_d_Aosta               Liguria
                  2.7                   5.2                   3.4
            Lombardia    Trentino_Alto_Adige               Veneto
                  5.2                   6.8                   4.1
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                  3.9                   3.0                   2.3
               Umbria               Marche                Lazio
                  2.2                   2.2                   2.1
             Abruzzo               Molise             Campania
                  1.9                   1.2                   1.3
              Puglia           Basilicata             Calabria
                  0.7                   0.7                   1.0
             Sicilia            Sardegna
                  1.1                   0.7
```

```
> bar= matriceAbitudiniAlimentari[,6]
> bar
             Piemonte         Valle_d_Aosta               Liguria
                  3.6                   3.4                   3.9
            Lombardia    Trentino_Alto_Adige               Veneto
                  2.4                   1.3                   2.2
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                  1.4                   2.1                   2.8
               Umbria                Marche                 Lazio
                  1.2                   1.5                   4.2
             Abruzzo               Molise              Campania
                  0.8                   0.4                   0.7
              Puglia            Basilicata              Calabria
                  0.7                   0.6                   0.6
             Sicilia              Sardegna
                  1.1                   0.7


> lavoro= matriceAbitudiniAlimentari[,7]
> lavoro
             Piemonte         Valle_d_Aosta               Liguria
                  9.7                   8.3                   9.4
            Lombardia    Trentino_Alto_Adige               Veneto
                 11.1                   6.5                   8.3
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                  7.2                   9.5                   8.2
               Umbria                Marche                 Lazio
                  7.3                   7.4                  11.1
             Abruzzo               Molise              Campania
                  5.3                   5.4                   8.6
              Puglia            Basilicata              Calabria
                  3.3                   5.0                   5.5
             Sicilia              Sardegna
                  4.9                   7.6

> pranzo= matriceAbitudiniAlimentari[,8]
> pranzo
             Piemonte         Valle_d_Aosta               Liguria
                 61.7                  69.1                  62.2
            Lombardia    Trentino_Alto_Adige               Veneto
                 59.1                  75.1                  67.3
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                 63.7                  62.8                  61.9
               Umbria                Marche                 Lazio
                 69.9                  69.1                  56.5
             Abruzzo               Molise              Campania
                 72.8                  81.4                  71.5
              Puglia            Basilicata              Calabria
                 82.8                  81.5                  75.1
             Sicilia              Sardegna
                 73.2                  75.5
```

```
> cena= matriceAbitudiniAlimentari[,9]
> cena
             Piemonte         Valle_d_Aosta               Liguria
                 29.0                  21.0                  28.9
            Lombardia    Trentino_Alto_Adige               Veneto
                 29.3                  13.2                  22.5
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                 25.3                  26.7                  26.1
              Umbria               Marche                Lazio
                 18.7                  17.2                  30.8
             Abruzzo               Molise             Campania
                 13.0                  11.8                  18.8
              Puglia           Basilicata             Calabria
                  9.1                   9.0                  12.1
             Sicilia             Sardegna
                 13.6                  14.1
```

1.2 Tabelle e Grafici

1.2.1 Distribuzioni di frequenza semplici

Nel caso dei dati forniti dal dataset scelto, non ha senso calcolare le distribuzioni di frequenza semplici; trattandosi infatti di dati ottenuti da 100 persone, è difficile che questi possano ripetersi più volte, permettendo quindi di calcolare, ad esempio, la frequenza assoluta.
Si preferisce quindi raccogliere le informazioni concernenti le *variabili quantitative* (o numeriche) in classi e calcolare le frequenze con cui gli elementi del vettore cadono nelle varie classi.
Per svolgere questa operazione si usa la funzione **cut()**, che permette di suddividere i dati relativi ad un vettore in intervalli, indicando nel parametro *breaks* gli estremi degli intervalli che sono chiusi a destra e aperti a sinistra. Se si desidera ottenere invece intervalli chiusi a sinistra e aperti a destra, all'interno della funzione **cut()** occorre settare il parametro **right=FALSE**.
In riferimento ai dati presi in analisi, si considerano le seguenti classi (0,25], (25,50], (50,75], (75,90].
Per ottenerle in R, si usa il seguente comando:

```
> frequenzeClassi <- c(0,25,50,75,90)
```

Consideriamo una variabile X e indichiamo con z_1, z_2, ..., z_k le diverse modalità che essa assume. Consideriamo poi un campione (x_1, x_2, ..., x_n) costituito da n osservazioni di X. Se indichiamo con n_i il numero di volte in cui ciascuna modalità z_i è presente nel campione, ossia la *frequenza assoluta* con cui essa appare nel campione, l'insieme $\{(z_i, n_i), i = 1, 2, ..., k\}$ prende il nome di **distribuzione di frequenza**.

In R per ottenere la **distribuzione di frequenza** di una *variabile qualitativa*, è usata la funzione **table(),** che è applicata a un vettore contenente i dati per poterne poi visualizzare le frequenze assolute. Questa funzione ordina le modalità della variabile in ordine alfabetico; se si desidera ottenere un risultato in base a uno specifico ordine, si considera il vettore come *fattore* e si ordinano i livelli di quest'ultimo nel modo che si preferisce.
Applichiamo quindi la funzione **table()** ai diversi vettori del dataset preso in analisi:

1 Percentuali di persone che hanno consumato una colazione adeguata nell'ultimo anno:

```
> table(cut(colazione_adeguata, frequenzeClassi))

(0,25] (25,50] (50,75] (75,90]
     0       0       1      19
```

La maggior parte delle regioni italiane (19 in questo caso) ha una percentuale molto alta di persone che consumano una colazione adeguata.

2 Percentuali di persone che hanno consumato una colazione con latte nell'ultimo anno:

```
> table(cut(colazione_con_latte, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
      0      20       0       0
```

Tutte le regioni italiane hanno una percentuale non molto alta (tra il 25 e il 50%) di persone che fanno colazione con latte.

3 Percentuali di persone che hanno consumato un pasto in casa nell'ultimo anno:

```
> table(cut(casa, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
      0       0      10      10
```

10 delle regioni italiane hanno una percentuale compresa tra 50 e 75 %, le restanti 10 invece una percentuale compresa tra 75 e 90%; in entrambi i casi la percentuale di persone che ha consumato un pasto in casa è molto alta.

4 Percentuali di persone che hanno consumato un pasto in mensa nell'ultimo anno:

```
> table(cut(mensa, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
     20       0       0       0
```

Tutte le regioni italiane hanno una percentuale molto bassa di persone che hanno consumato un pasto in mensa nell'ultimo anno, compresa tra lo 0 e il 25%.

5 Percentuali di persone che hanno consumato un pasto al ristorante nell'ultimo anno:

```
> table(cut(ristorante, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
     20       0       0       0
```

Tutte le regioni italiane presentano una percentuale molto bassa di persone che hanno consumato un pasto al ristorante nell'ultimo anno, compresa tra lo 0 e il 25%.

6 Percentuali di persone che hanno consumato un pasto al bar nell'ultimo anno:

```
> table(cut(bar, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
     20       0       0       0
```

Tutte le regioni italiane mostrano una percentuale molto bassa di persone che hanno consumato un pasto al bar nell'ultimo anno, compresa tra lo 0 e il 25%.

7 Percentuali di persone che hanno consumato un pasto al lavoro nell'ultimo anno:

```
> table(cut(lavoro, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
     20       0       0       0
```

Tutte le regioni italiane segnalano una percentuale molto bassa di persone che hanno consumato un pasto al lavoro nell'ultimo anno, compresa tra lo 0 e il 25%.

8 Percentuali di persone che hanno pranzato nell'ultimo anno:

```
> table(cut(pranzo, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
      0       0      14       6
```

La maggior parte delle regioni (14) ha consumato uno dei pasti principali della giornata, il pranzo, con una percentuale compresa tra il 50 e il 75%; le persone delle 6 restanti regioni hanno invece consumato un pasto con una percentuale compresa tra il 75 e il 90%.

9 Percentuali di persone che hanno cenato nell'ultimo anno:

```
> table(cut(cena, frequenzeClassi))

 (0,25] (25,50] (50,75] (75,90]
     13       7       0       0
```

13 regioni hanno una percentuale molto bassa, compresa tra 0 e 25 per cento, di persone che hanno cenato nell'ultimo anno; le restanti 7 regioni invece hanno una percentuale compresa tra 25 e 50 di persone che hanno cenato.

Per calcolare la distribuzione delle *frequenze relative*, è sufficiente dividere le frequenze assolute per il numero di elementi all'interno del vettore preso in considerazione, ossia **table(X)/lenght(X)**. Si può applicare anche in questo caso alle classi calcolate; i risultati sarebbero comunque gli stessi, ma espressi in modo diverso.
Per le *frequenze assolute cumulate*, si usa il comando **cumsum(table(X))**.
Per le *frequenze relative cumulat*e invece, è sufficiente dividere le frequenze assolute cumulate per il numero di elementi del vettore con il comando **cumsum(table(X))/lenght(X)**.

1.2.2 Grafici per statistiche univariate

R mette a disposizione dell'utente una serie di funzioni grafiche per produrre diverse tipologie di grafici in base al tipo di dati forniti. Nel dataset preso in considerazione, saranno applicate le funzioni per ottenere grafici utili alla descrizione dei fenomeni qualitativi e quantitativi per statistiche univariate. I grafici principali di questo tipo sono 3:

1. Grafico a barre
2. Grafico a bastoncino
3. Grafico a torta

Nel caso preso in analisi, non ha senso considerare i grafici riguardanti le frequenze assolute del tipo di abitudine alimentare. Infatti, dato che si parla di un valore preciso espresso in decimali, nessuna regione o quasi, avrà un numero esattamente uguale ad un'altra; per questo motivo non verrà considerato il tipo di grafico a bastoncino, che fa uso della funzione **plot(table(X))**; saranno considerati quindi solo quello a barre e a torta.
Usando la funzione **barplot()** o la funzione **pie()**, è possibile creare rispettivamente un grafico a barre e uno a torta, per poter osservare ed analizzare il comportamento delle singole colonne. Applichiamo quindi la funzione **barplot()** per tutte le colonne della matrice *matriceAbitudiniAlimentari:*

```
> barplot(colazione_adeguata, sub="colazione adeguata", col=1:20, las=2)
```

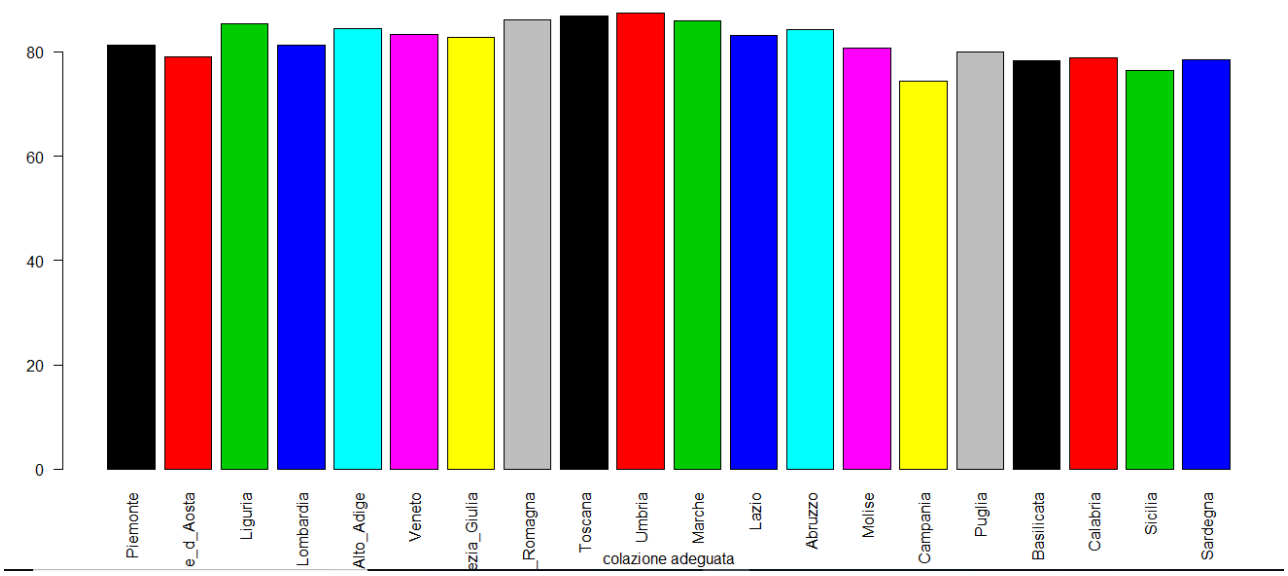


Figura 1.1: Grafico a barre colazione adeguata

Dal grafico in Figura 1.1 è possibile notare che le persone che nell'ultimo anno hanno consumato una colazione adeguata sono state quelle della regione Umbria, invece quelle che meno hanno fatto una colazione adeguata sono state quelle della regione Campania.

```
> barplot(colazione_con_latte,col=1:20,las=2)
```

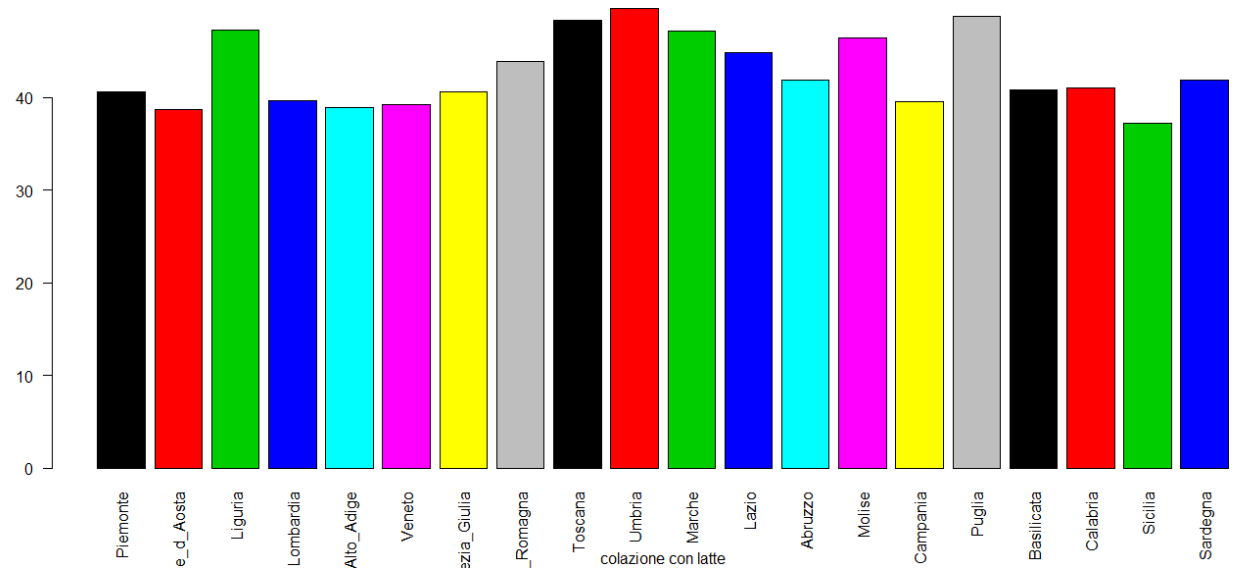


Figura 1.2: Grafico a barre colazione con latte

Analizzando il grafico in Figura 1.2, si può osservare che le persone che hanno maggiormente fatto colazione con latte sono quelle della regione Puglia e della regione Umbria; la regione invece in cui questo tipo di pasto è meno diffuso è la Sicilia.

```
> barplot(casa,sub="pasto in casa",col=1:20,las=2)
```

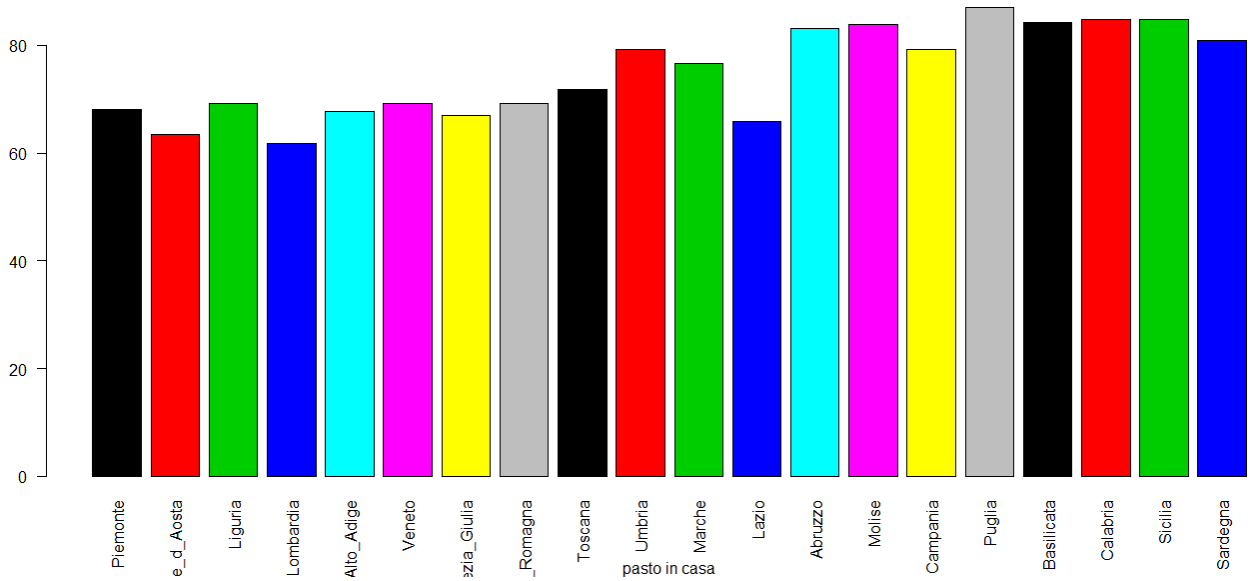


Figura 1.3: Grafico a barre pasto in casa

Osservando la Figura 1.3 si nota che le persone che più hanno consumato pasti in casa sono quelle della regione Puglia, quelle invece che meno li hanno consumati sono quelle della Lombardia.

```
> barplot(mensa,sub="pasto in mensa",col=1:20,las=2)
```

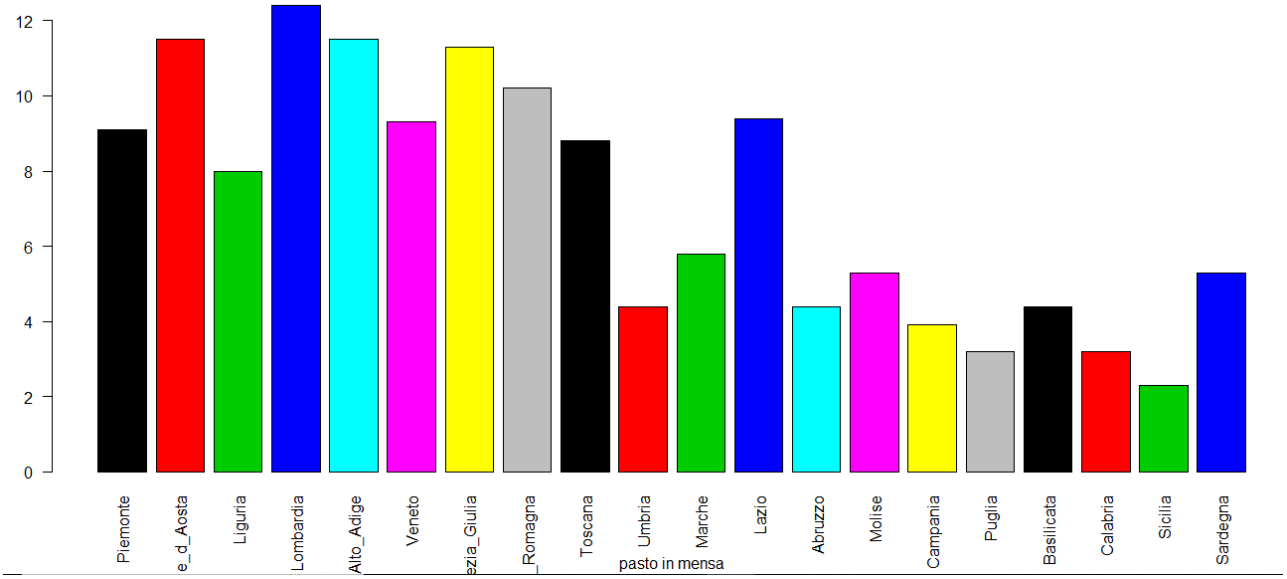


Figura 1.4: Grafico a barre pasto in mensa

Analizzando la Figura 1.4 si nota che le persone che più hanno consumato pasti in mensa sono quelle della regione Lombardia, quelle invece che meno li hanno consumati sono quelle della Sicilia.

```
> barplot(ristorante,sub="pasto al ristorante",col=1:20,las=2)
```

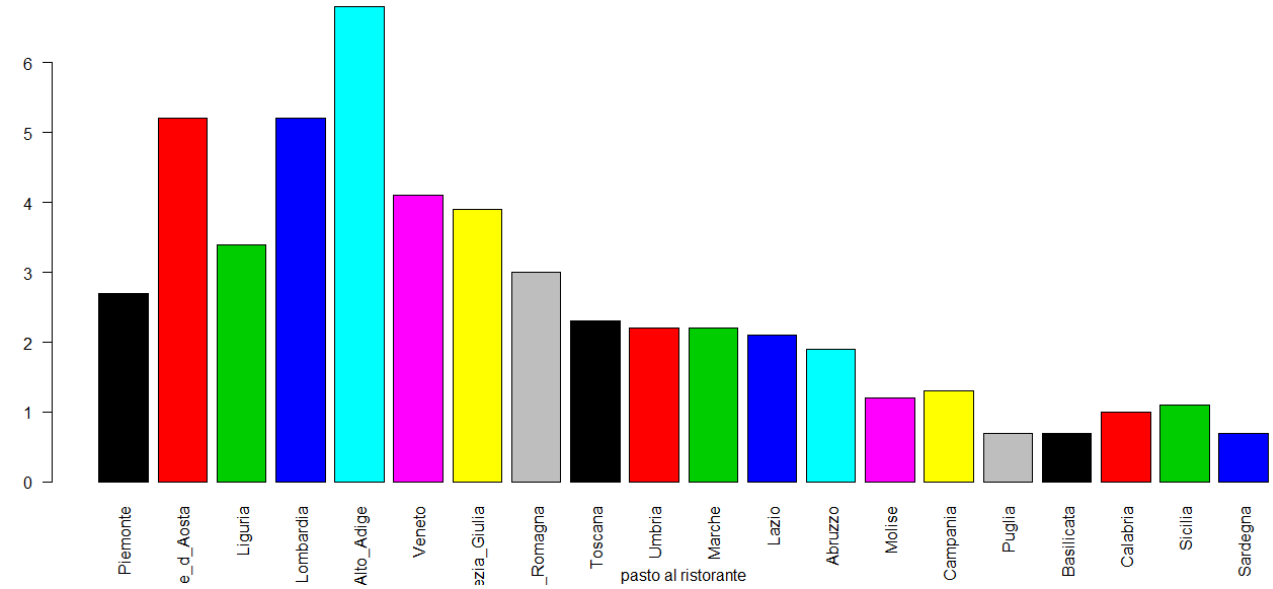


Figura 1.5: Grafico a barre pasto al ristorante

Guardando la Figura 1.5 si osserva che la regione in cui le persone si sono prevalentemente recate al ristorante per un pasto è il Trentino Alto-Adige, quelle in cui questo fenomeno è meno presente sono invece la Puglia, Sardegna e Basilicata.

```
> barplot(bar,sub="pasto al bar",col=1:20,las=2)
```

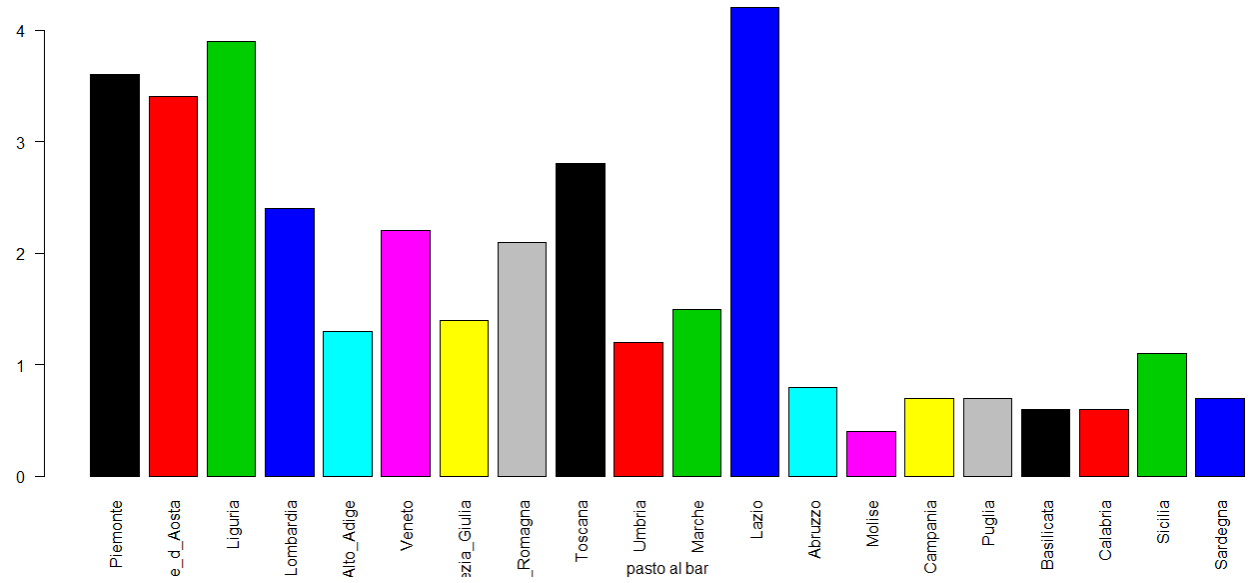


Figura 1.6: Grafico a barre pasto al bar

Dalla Figura 1.6 notiamo che le persone che maggiormente hanno consumato un pasto al bar sono quelle appartenenti alla regione Lazio; quelle che invece minormente ne hanno consumato sono quelle del Molise.

```
> barplot(lavoro,sub="pasto al lavoro",col=1:20,las=2)
```

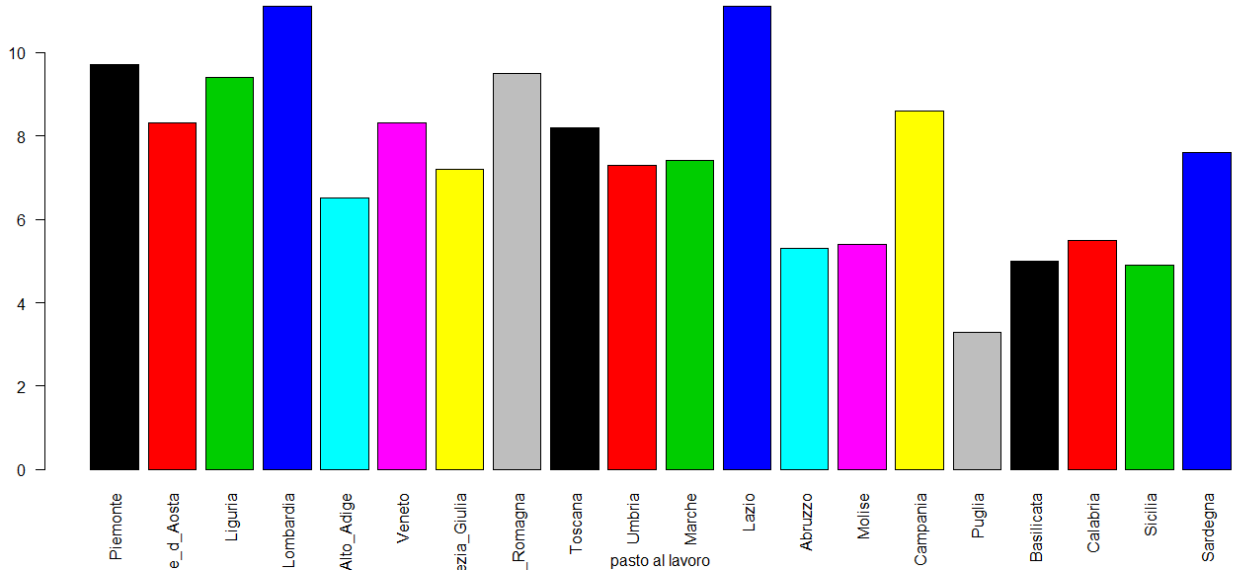


Figura 1.7: Grafico a barre pasto al lavoro

Nella Figura 1.7 ci si accorge che le persone che prevalentemente consumano pasti sul posto di lavoro sono quelle del Lazio e della Lombardia, quelle che invece meno ne consumano appartengono alla Puglia.

```
> barplot(pranzo,sub="pranzo",col=1:20,las=2)
```

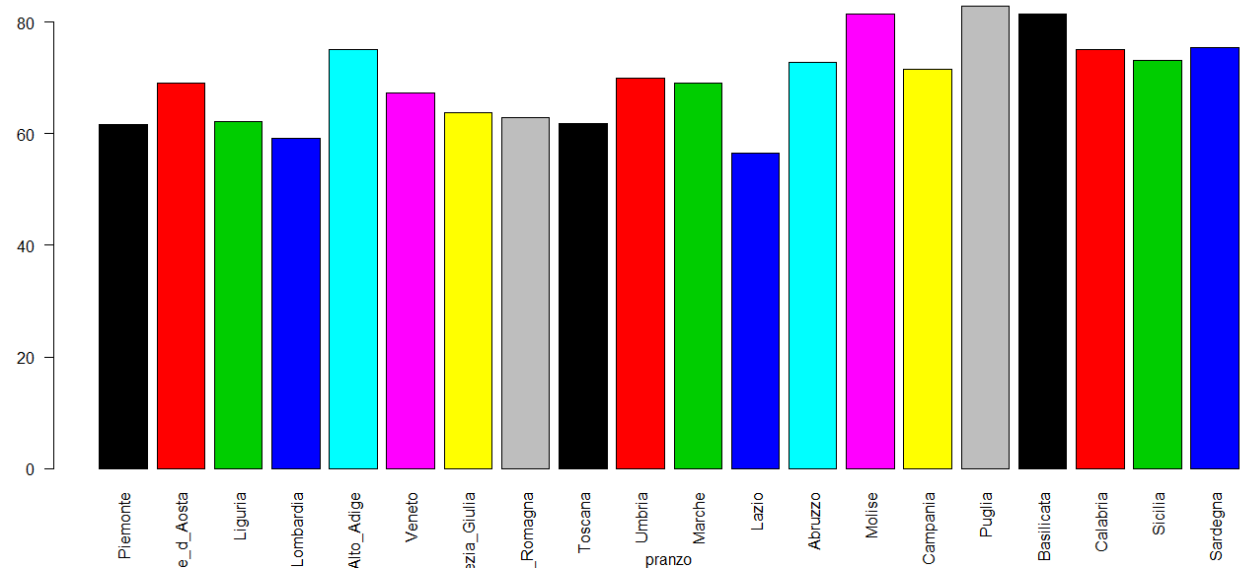


Figura 1.8: Grafico a barre pranzo

Osservando la Figura 1.8 è possibile notare che la quantità di persone che più ha pranzato è situata nella Puglia, quella che invece meno ha pranzato è nella regione Lazio.

```
> barplot(cena,sub="cena",col=1:20,las=2)
```

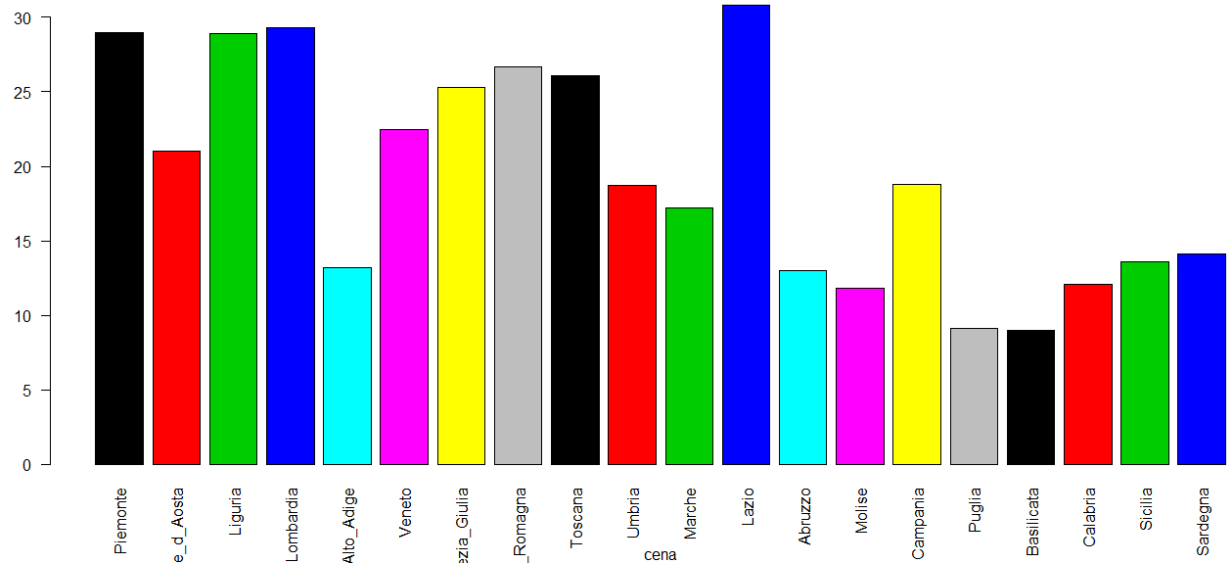


Figura 1.9: Grafico a barre cena

Dalla Figura 1.9 si evince che le persone che hanno maggiormente cenato sono quelle del Lazio, quelle che invece meno hanno cenato sono quelle della Basilicata.

Se invece si vuole analizzare i tipi di abitudini alimentari che posseggono le persone nelle varie regioni, è possibile lavorare sulle righe.
Consideriamo quindi la regione *Lazio* e analizziamo i dati con un grafico a torta, ottenuto con la funzione **pie().**

Otteniamo quindi prima il vettore *Lazio*

```
> Lazio <- matriceAbitudiniAlimentari[12,]
> Lazio
 colazione_adeguata colazione_con_latte                casa
               83.2                44.9                65.8
             mensa          ristorante                 bar
               9.4                 2.1                 4.2
           lavoro              pranzo               cena
               11.1                56.5                30.8
```

Con il comando

```
> coloriGraficoTorta = c("red", "yellow", "green", "violet", "orange", "blue", "pink", "cyan","black")
```

viene creato un vettore di colori, che successivamente sarà assegnato al parametro *col* della funzione **pie().**
Realizziamo quindi il grafico in Figura 1.10 col comando

```
> pie(Lazio, col=coloriGraficoTorta)
```

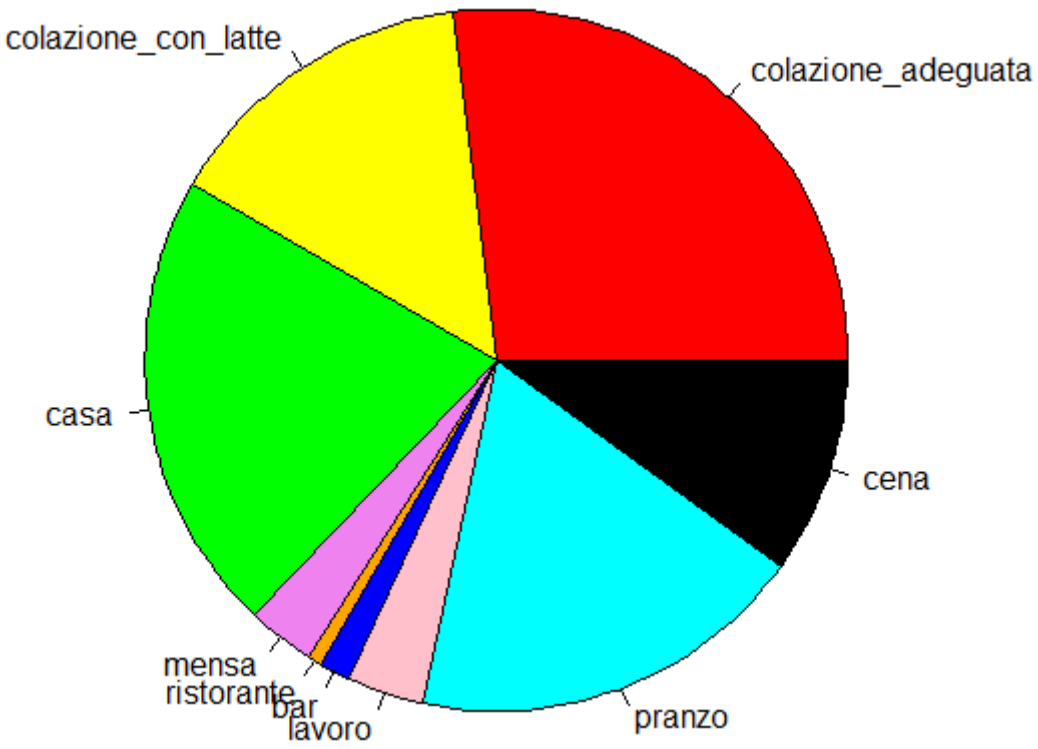


Figura 1.10: Grafico a torta per la regione Lazio

Dal seguente grafico si ha una visione chiara circa le abitudini alimentari nell'ultimo anno nella regione Lazio. La maggior parte ha consumato una colazione adeguata, invece il pasto meno consumato risulta essere quello al ristorante.
Per quanto concerne il grafico a torta, è possibile usare un particolare tipo di tratteggio per le varie sezioni della circonferenza, al posto dei vari colori.
Per fare ciò è sufficiente utilizzare i parametri *density* e *angle*, come mostrato nella seguente linea di codice:

```
> pie(Lazio, density=20, angle=18+10*(1:9), col=coloriGraficoTorta)
```

che permettono di ottenere il seguente grafico:

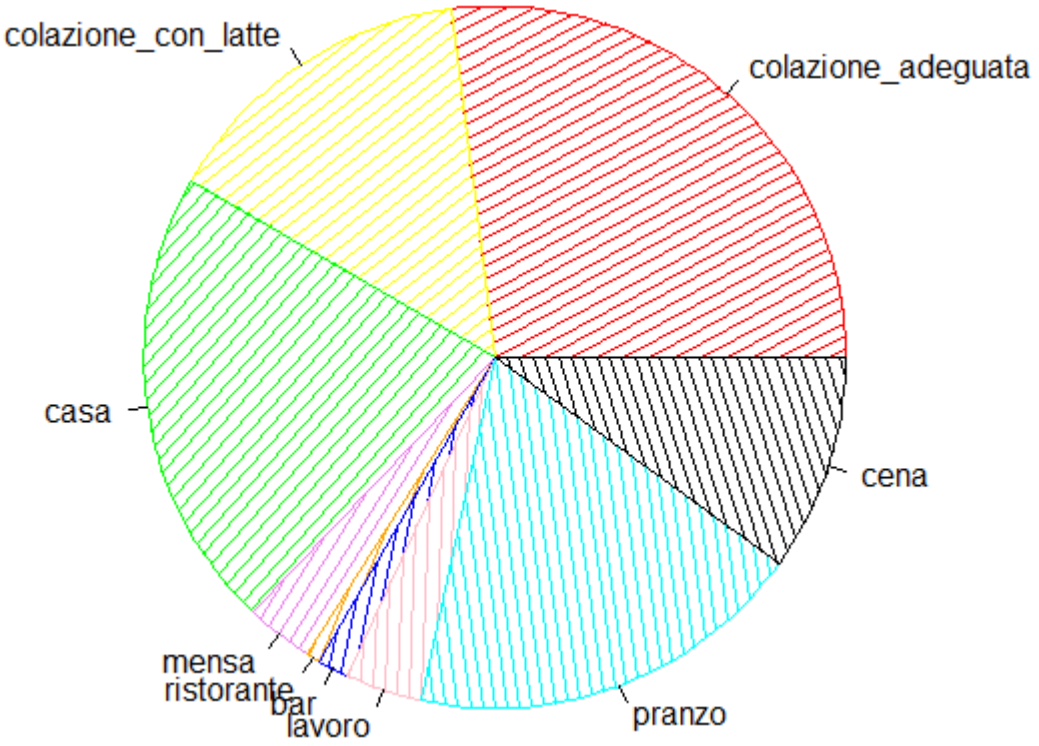


Figura 1.11: Grafico a torta con tratteggio per la regione Lazio

Esistono altri tipi di rappresentazioni che *non* forniscono informazioni significative circa la distribuzione di frequenza. Ad esempio, si considera una variabile quantitativa *colazione_adeguata* e valori numerici assunti dalle varie regioni.
Tramite il comando **plot()**, viene illustrato l'andamento dei valori assunti dal tipo di pasto preso in considerazione rispetto alle relative regioni.
Questo comando non fornisce quindi informazioni significative circa la distribuzione di frequenza.
Il seguente codice produce il grafico rappresentato in Figura 1.12, che illustra le percentuali di persone che hanno fatto una colazione adeguata nell'ultimo anno, considerato per ognuno delle 20 regioni individuate dalle loro posizioni nel vettore.

```
plot(colazione_adeguata, ylab="colazione adeguata", col="blue")
```

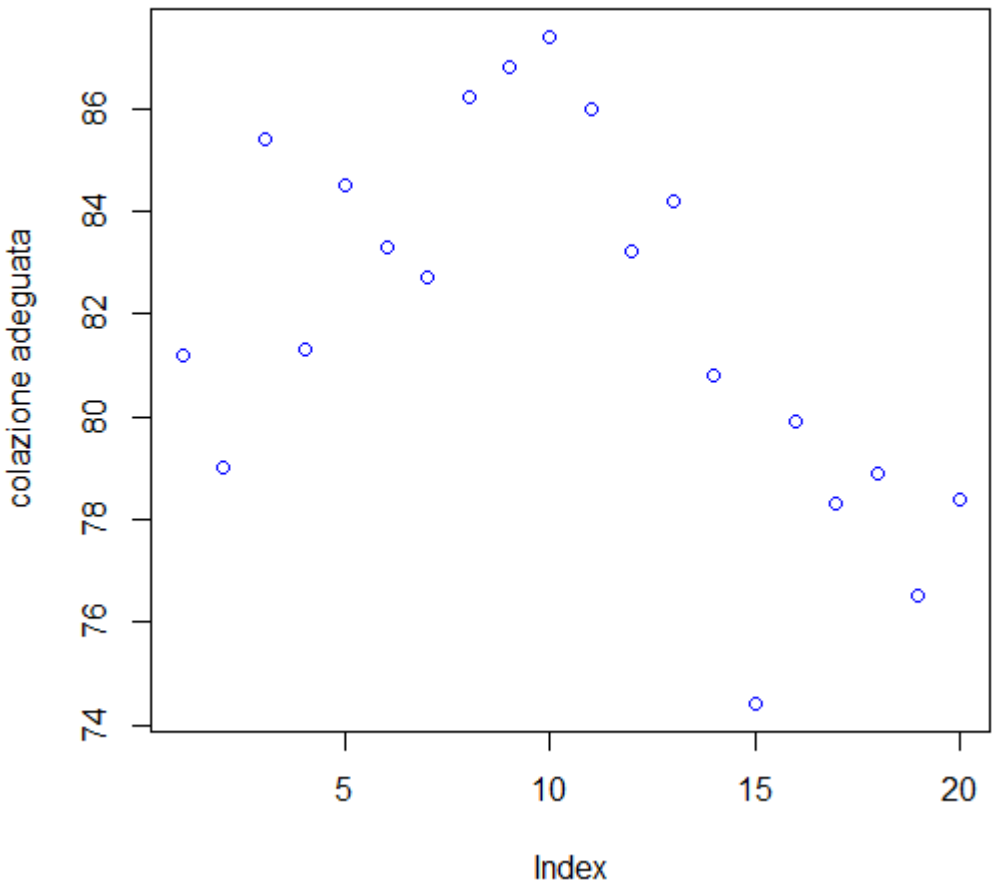


Figura 1.12: Rappresentazione dei valori assunti dal vettore numerico *colazione_adeguata* in corrispondenza dei propri indici.

Dal momento che la rappresentazione è abbastanza dispersiva, viene ritenuto opportuno collegare i punti del grafico in Figura 1.12 mediante delle linee, creando così una *serie storica* del tipo di pasto colazione adeguata. È quindi usata la funzione **plot()** con il parametro **type=l** (L) per creare linee interconnesse, come mostrato da Figura 1.13

```
> plot(colazione_adeguata, type="l", ylab="colazione adeguata", col="blue")
```

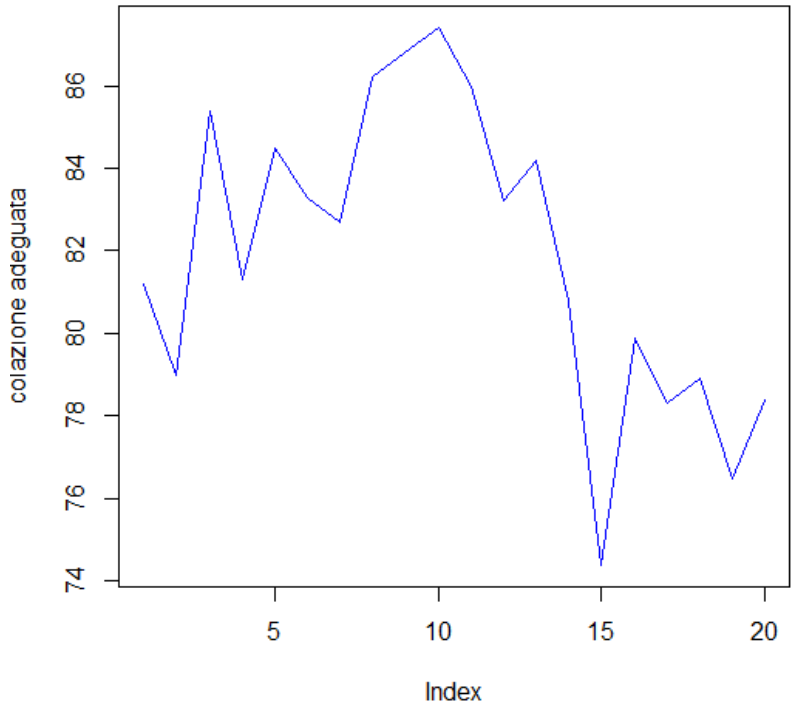


Figura 1.13: Rappresentazione dei valori assunti dal vettore numerico *colazione_adeguata* in corrispondenza dei propri indici con linee interconnesse.

Tenendo in considerazione il modo in cui R lavora con le finestre grafiche, permettendo di aggiungere dettagli, è possibile procedere anche in questo modo, aggiungendo punti alle linee interconnesse:

```
> plot(colazione_adeguata, ylab="colazione adeguata", col="black")
> lines(colazione_adeguata, col="red")
```

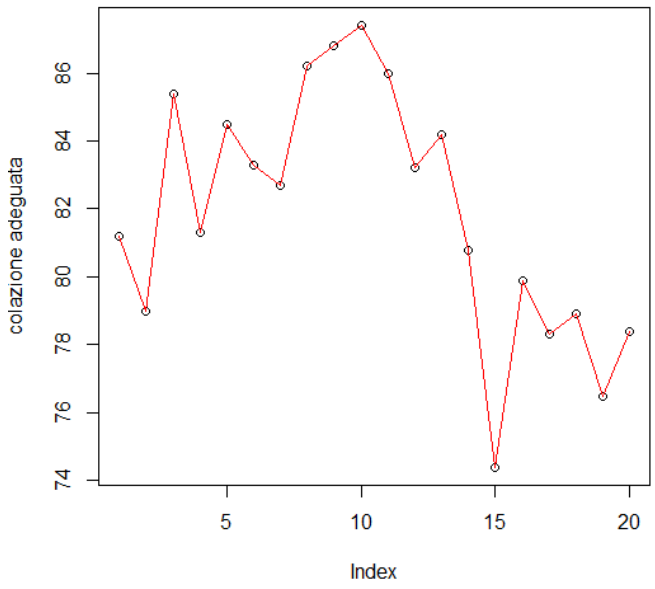


Figura 1.14: Rappresentazione dei valori assunti dal vettore numerico *colazione_adeguata* in corrispondenza dei propri indici con linee interconnesse e punti.

Come detto da precedenza, questo tipo di grafico non dà informazioni significative sulla distribuzione dei dati presi in considerazione.
Consideriamo quindi gli istogrammi.

1.2.3 Istogrammi

Affinché la rappresentazione della distribuzione di frequenza dei dati numerici sia corretta, è necessario considerare delle classi. Consideriamo un campione (x_1, x_2, ..., x_n) costituito da n osservazioni, suddivise in classi, in cui ogni osservazione cade in una e soltanto una classe (o intervallo). Gli istogrammi sono una particolare rappresentazione grafica di una distribuzione di frequenza in classi, ottenuta mediante rettangoli adiacenti che hanno per base segmenti i cui estremi coincidono con gli estremi delle classi. Una volta fissate le basi, le altezze devono essere tali che l'area di ogni rettangolo risultante sia uguale alla frequenza (assoluta o relativa) della classe stessa.
Se si usano le frequenze assolute delle classi, l'area di ogni rettangolo è uguale alla frequenza assoluta della classe; quindi il rettangolo i-esimo ha area uguale a $n_i = b_i \times h_i$, dove:

- n_i è il numero di valori che cadono nella classe i-esima;
- b_i è la base della classe i-esima;
- h_i è l'altezza della classe i-esima.

In R la funzione che realizza istogrammi è **hist()**. È possibile lasciare scegliere a R il numero di classi da usare, oppure indicargli il numero di classi da utilizzare col parametro *breaks*.
Per realizzare un istogramma in base alle frequenze assolute, viene settato il parametro *freq=TRUE* nella funzione **hist()** (per le frequenze relative invece *freq=FALSE*).
In riferimento al vettore *colazione_adeguata*, il seguente codice produce l'istogramma in figura.

```
> hist(colazione_adeguata, main="Istogramma colazione adeguata", ylab="Frequenza assoluta classi", col=1:7)
```

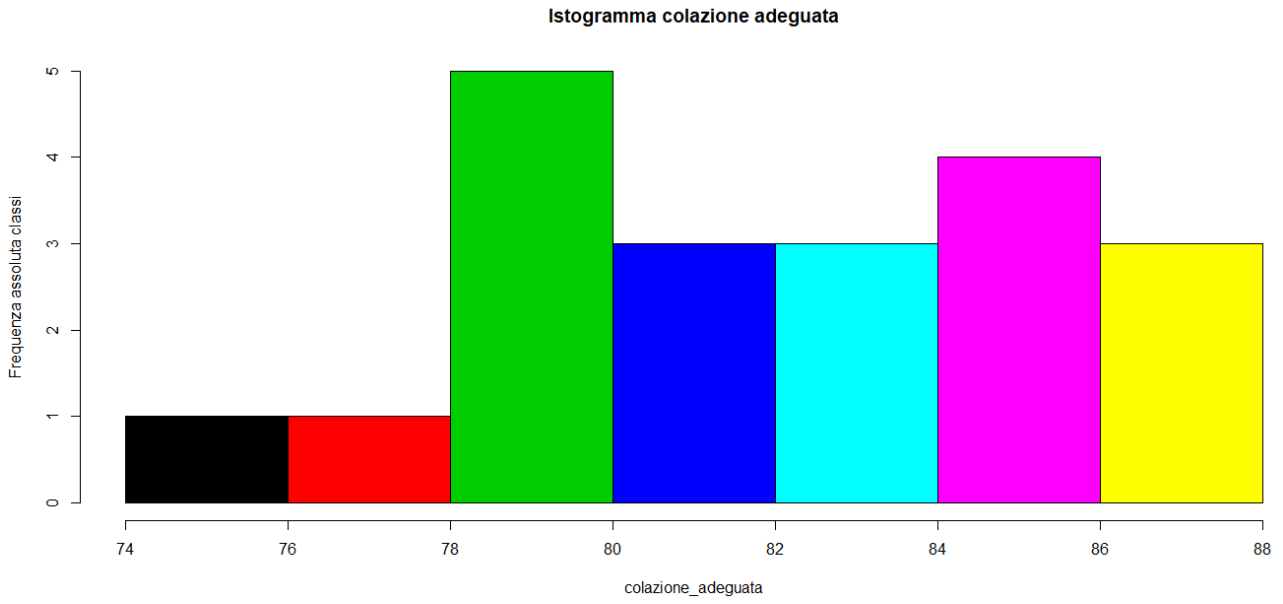


Figura 1.15: Istogramma relativo al vettore *colazione_adeguata* in base alle frequenze assolute delle classi.

La funzione **hist()** fornisce anche una serie di informazioni sulla natura dell'istogramma, che possono essere salvate in una variabile *h* di tipo *list*.
Con il comando **str(h)** è possibile visualizzare queste informazioni. Nel caso preso in esame si ottiene:

```
> h <- hist(colazione_adeguata, main="Istogramma colazione adeguata", ylab="Frequenza assoluta classi", col=1:7)
>
> str(h)
List of 6
 $ breaks  : int [1:8] 74 76 78 80 82 84 86 88
 $ counts  : int [1:7] 1 1 5 3 3 4 3
 $ density : num [1:7] 0.025 0.025 0.125 0.075 0.075 0.1 0.075
 $ mids    : num [1:7] 75 77 79 81 83 85 87
 $ xname   : chr "colazione_adeguata"
 $ equidist: logi TRUE
 - attr(*, "class")= chr "histogram"
```

La funzione **str(h)** 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*).

```
> h$breaks
[1] 74 76 78 80 82 84 86 88
> h$counts
[1] 1 1 5 3 3 4 3
> h$density
[1] 0.025 0.025 0.125 0.075 0.075 0.100 0.075
> h$mids
[1] 75 77 79 81 83 85 87
```

La suddivisione in classi scelta da R è la seguente: (74,76], (76,78], (78,80], (80,82], (82,84], (84,86], (86,88]; dei 20 valori, 1 cade nella prima classe, 1 nella seconda, 5 nella terza, 3 nella quarta, 3 nella quinta, 4 nella sesta, 3 nella settima.
Con l'informazione *density*, è possibile ottenere le frequenze relative associate alle classi dell'istogramma. Per calcolarle si moltiplica *density* per l'ampiezza di ogni intervallo dell'istogramma (in questo caso 2):

```
> f <- 2*h$density
> f
[1] 0.05 0.05 0.25 0.15 0.15 0.20 0.15
>
> sum(f)
[1] 1
```

Sommando le frequenze relative, otteniamo 1.
Si vuole ora fissare le classi dell'istogramma mediante il parametro *breaks*:

```
> classi <- c(70,75,80,85,90)
>  hist(colazione_adeguata, freq=TRUE , main=" Istogramma colazione adeguata", breaks=classi,
+  ylab=" Frequenza   assoluta   delle   classi", col=1:7)
```

Producendo quindi l'istogramma in Figura 1.16.
Gli intervalli unitari sono aperti a sinistra e chiusi a destra, ossia del tipo (k, k+1] dove k = 70, 75, 80, 85, 90. L'area totale è uguale all'ampiezza del campione. Sottolineiamo che troppe classi rendono un istogramma simile a un grafico a barre, troppe poche invece lo rendono piatto.

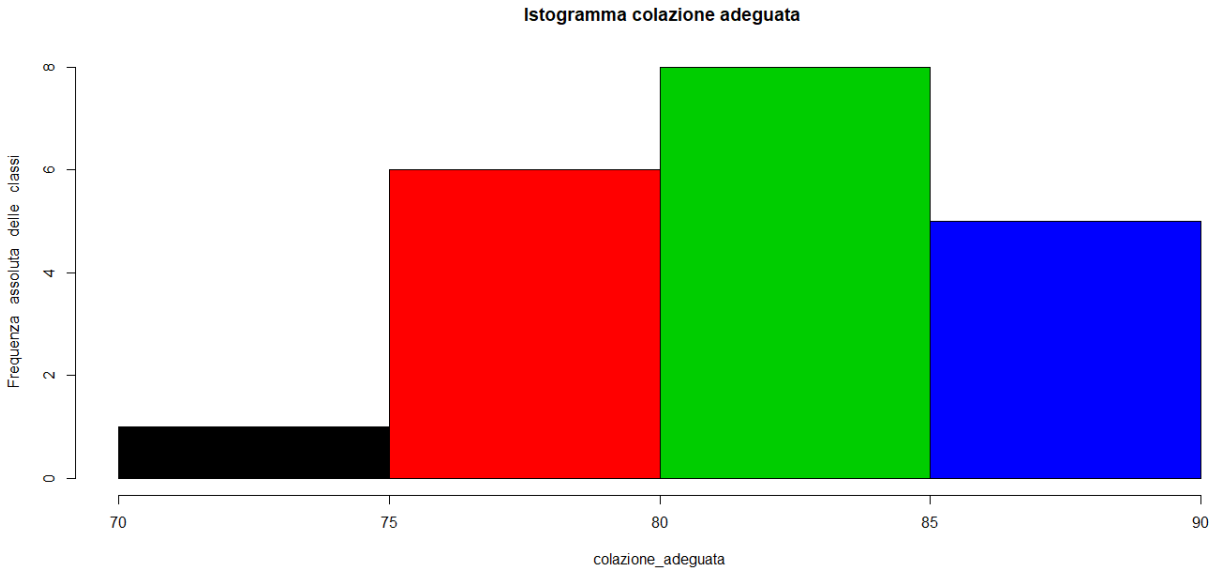


Figura 1.16: Istogramma relativo al vettore *colazione_adeguata* in base alle frequenze assolute delle classi scelte.

Se invece vengono utilizzate le frequenze relative delle classi, l'area di ogni rettangolo è uguale alla frequenza relativa della classe stessa e l'area totale è uguale a 1. Dunque, l'area del rettangolo *i*-esimo è uguale a $f_i = n_i/n = b_i \times h_i$, dove f_i è la frequenza relativa dei valori della classe *i*-esima, b_i è l'ampiezza e h_i è l'altezza della classe i-esima. In questo caso l'altezza di ogni rettangolo esprime la densità di frequenza associata alla classe considerata.
Per realizzare un istogramma in base alle frequenze relative, si setta il parametro *freq=FALSE* nella funzione **hist().** Consideriamo ancora il vettore *colazione_adeguata*.
Considerando le seguenti linee di codice:

```
> hist(colazione_adeguata, freq=FALSE, main="Istogramma colazione adeguata",
+ ylab="Densità di frequenza delle classi", col=1:7)
```

Si ottiene il grafico in Figura 1.17 in base alla densità di frequenza delle classi. In questo caso il numero delle classi è scelto automaticamente da R e l'area totale è unitaria.

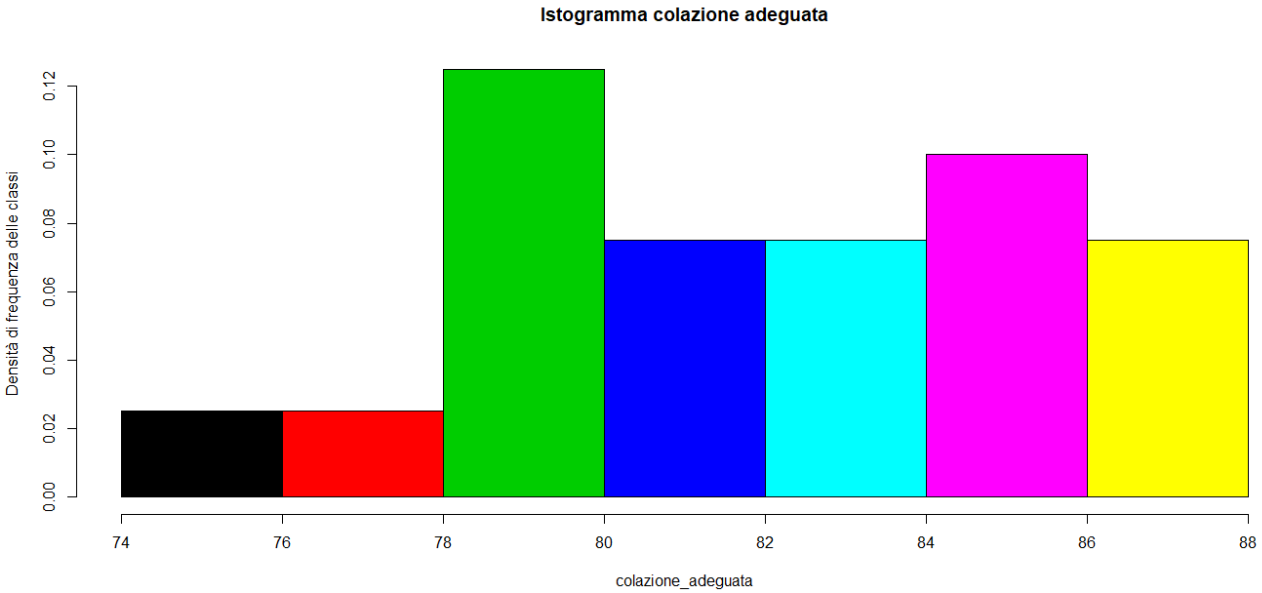


Figura 1.17: Istogramma relativo al vettore *colazione_adeguata* fissando il numero di classi in base alle densità di frequenza delle classi.

Osservando il grafico, si nota che applicando **$density** si ottengono le altezze relative ai rettangoli associati alle 7 classi.

1.2.4 Boxplot

Consideriamo un campione (x_1, x_2, ..., x_n) dei valori assunti da una variabile quantitativa X. Ordiniamo i valori del campione in ordine crescente. Si chiama *primo quartile*, indicato con Q_1, il valore per il quale il 25% dei dati sono alla sua sinistra e il restante 75% alla sua destra. Analogamente, prende il nome di *terzo quartile*, indicato con Q_3, il valore per il quale il 75% dei dati sono alla sua sinistra e il 25% alla sua destra. Il *secondo quartile*, Q_2, corrisponde al valore per il quale il 50% dei dati sono alla sua sinistra e il 50% dei dati alla sua destra, detto anche **mediana**. Q_0 e Q_4 corrispondono rispettivamente al minimo e al massimo dei valori del campione.
In R i quantili si calcolano con la funzione **quantile(nomeVettore)**, mentre la funzione **summary(nomeVettore)** permette di calcolare i valori precisi del minimo, del massimo, della media, della mediana, del primo e del terzo quantile.
Consideriamo ancora il vettore *colazione_adeguata*:

```
> quantile(colazione_adeguata)
     0%     25%     50%     75%    100%
74.400  78.975  82.000  84.725  87.400
```

Da cui si deduce che Q_0=74.40, Q_1=78.975, Q_2=82.00, Q_3=84.725, Q_4=87.40; inoltre si ha:

```
> summary(colazione_adeguata)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  74.40   78.97   82.00   81.92   84.72   87.40
```

Il **boxplot** è una rappresentazione grafica di una scatola, i cui estremi sono Q_1 e Q_3, tagliata da una linea orizzontale in corrispondenza di Q_2, cioè della **mediana**.
In basso e in alto sono presenti altre due linee orizzontali, dette *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* o *scarto 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 $(Q_1 - 1.5 * (Q_3 – Q_1), Q_3 + 1.5 * (Q_3 – Q_1))$ sono visualizzati nel grafico sotto forma di punti, detti *valori anomali* o *outlier*. Questi valori infatti costituiscono una "anomalia" rispetto alla maggior parte dei valori osservati e pertanto è necessario identificarli per poterne analizzare le caratteristiche e le eventuali cause che li hanno determinati.
Il boxplot è usato per illustrare alcune caratteristiche di una distribuzione di frequenza: *centralità, forma, dispersione e presenza di outlier*.
Per ottenere un boxplot si usa la funzione **boxplot(nomeVettore)**. La figura può essere disegnata sia in orizzontale che in verticale, a seconda del parametro **horizontal** che di default è **FALSE** nel caso verticale.
Prendiamo in considerazione il vettore *colazione_adeguata* e produciamo un boxplot con la seguente linea di codice:

```
> boxplot(colazione_adeguata, xlab="colazione adeguata",
+ main="Boxplot del vettore colazione_adeguata", col="green")
```

Dando vita al grafico in Figura 1.18

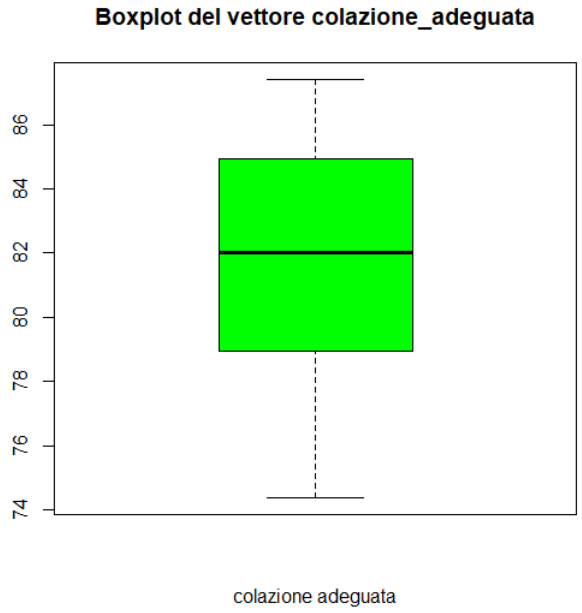


Figura 1.18: Boxplot relativo al vettore *colazione_adeguata*.

Dal grafico si può osservare che gli estremi della scatola sono Q_1=78.975 e Q_3=84.725; essa è tagliata da una linea orizzontale in corrispondenza di Q_2=82.00. Il baffo inferiore corrisponde alla più piccola tra le osservazioni che risulta maggiore o uguale di (Q_1 - 1.5 * (Q_3 – Q_1)) = (78.975 – 1.5 * (84.725 - 78.975)) = 70.35 ossia 74.40, il baffo superiore invece corrisponde al valore più grande delle osservazioni che risulta minore o uguale a (Q_3 + 1.5 * (Q_3 – Q_1)) = (84.725 + 1.5 * (84.725 - 78.975)) = 93.35 ossia 87.40. I baffi sono quindi posti in corrispondenza di 74.40 (minimo valore) e di 87.40 (massimo valore). Dato che tutti i valori sono compresi tra l'intervallo [70.35, 93.35], non ci sono outliers. Di seguito sono rappresentati i boxplot per tutte le colonne del dataset preso in considerazione.

- Vettore *colazione_con_latte*

```
> quantile(colazione_con_latte)
    0%    25%    50%    75%   100%
37.200 39.675 41.500 46.675 49.600
> summary(colazione_con_latte)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  37.20   39.67   41.50   42.84   46.67   49.60
>
> boxplot(colazione_con_latte, xlab="colazione con latte",
+ main="Boxplot del vettore colazione_con_latte", col="red")
```

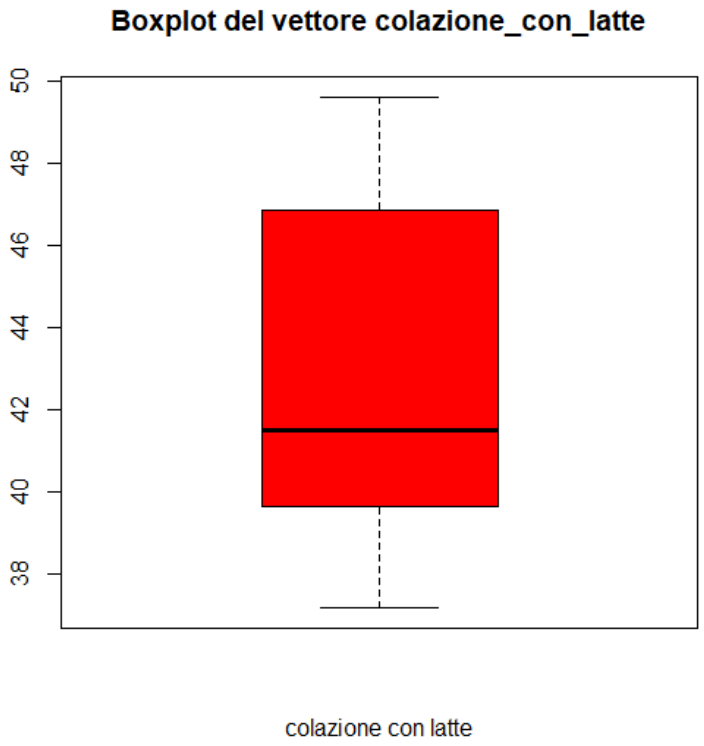


Figura 1.19: Boxplot relativo al vettore *colazione_con_latte*.

❖ Vettore *casa*

```
> quantile(casa)
    0%    25%    50%    75%   100%
61.800 68.025 74.250 83.375 87.000
> summary(casa)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  61.80   68.03   74.25   74.89   83.38   87.00
>
> boxplot(casa, xlab="casa",
+ main="Boxplot del vettore casa", col="yellow")
```

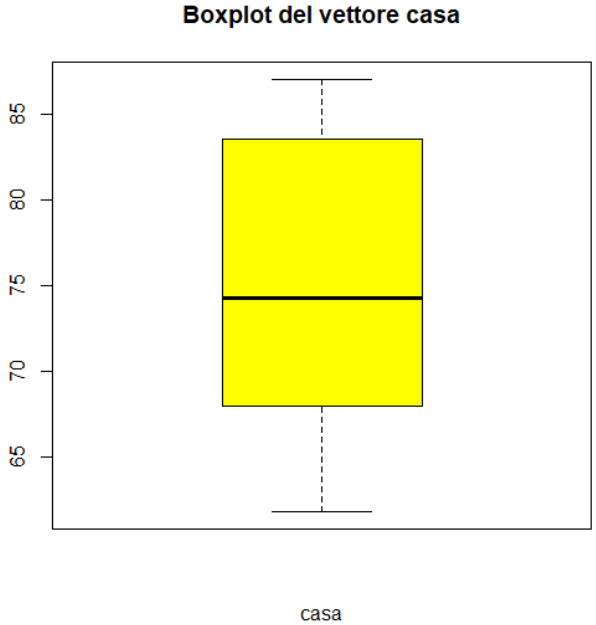


Figura 1.20: Boxplot relativo al vettore *casa*.

❖ Vettore *mensa*

```
> quantile(mensa)
  0%  25%  50%  75% 100%
 2.3  4.4  6.9  9.6 12.4
> summary(mensa)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.300   4.400   6.900   7.185   9.600  12.400
>
> boxplot(mensa, xlab="mensa",
+ main="Boxplot del vettore mensa", col="blue")
```

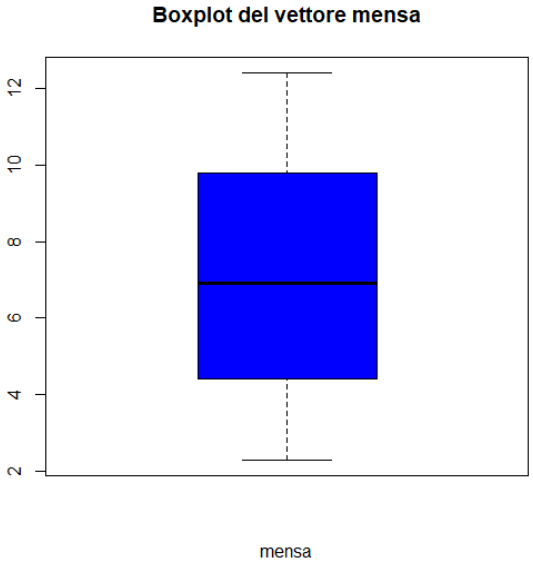


Figura 1.21: Boxplot relativo al vettore *mensa*.

❖ Vettore *ristorante*

```
> quantile(ristorante)
   0%   25%   50%   75%  100%
0.700 1.175 2.200 3.525 6.800
> summary(ristorante)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.700   1.175   2.200   2.585   3.525   6.800
>
> boxplot(ristorante, xlab="ristorante",
+ main="Boxplot del vettore ristorante", col="orange")
```

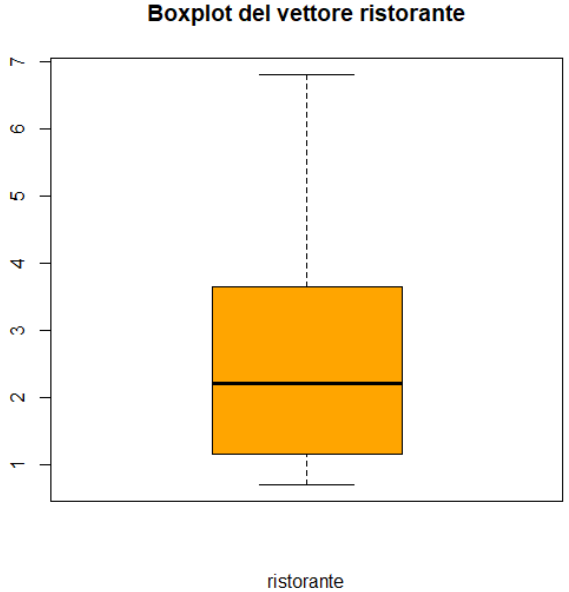


Figura 1.21: Boxplot relativo al vettore *ristorante*.

❖ Vettore *bar*

```
> quantile(bar)
  0%  25%  50%  75% 100%
0.40 0.70 1.35 2.50 4.20
> summary(bar)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.40    0.70    1.35    1.78    2.50    4.20
>
> boxplot(bar, xlab="bar",
+ main="Boxplot del vettore bar", col="pink")
```

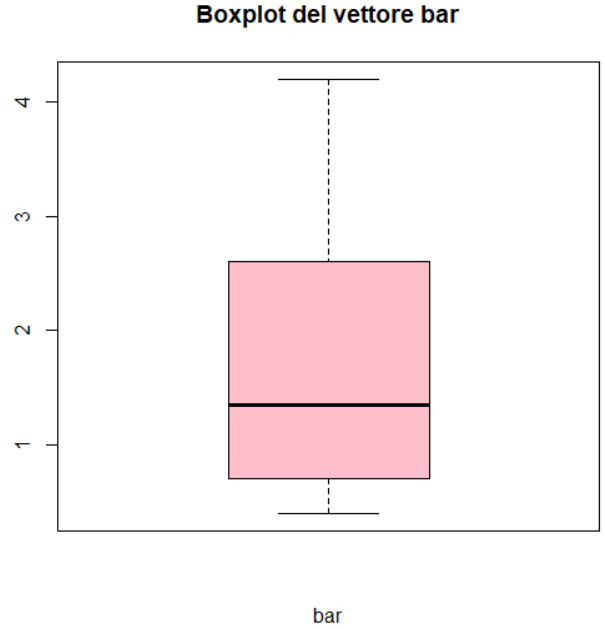


Figura 1.22: Boxplot relativo al vettore *bar*.

❖ Vettore *lavoro*

```
> quantile(lavoro)
    0%    25%    50%    75%   100%
 3.300  5.475  7.500  8.800 11.100
> summary(lavoro)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.300   5.475   7.500   7.480   8.800  11.100
>
> boxplot(lavoro, xlab="lavoro",
+ main="Boxplot del vettore lavoro", col="cyan")
```

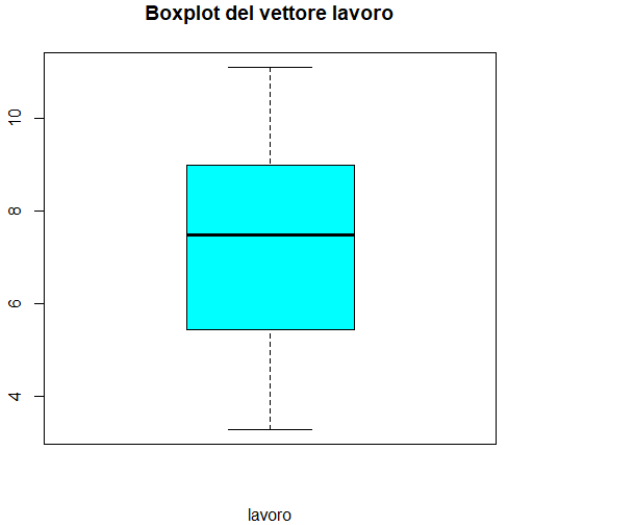


Figura 1.23: Boxplot relativo al vettore *lavoro*.

❖ Vettore *pranzo*

```
> quantile(pranzo)
   0%   25%   50%   75%  100%
56.50 62.65 69.50 75.10 82.80
> summary(pranzo)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  56.50   62.65   69.50   69.61   75.10   82.80
>
> boxplot(pranzo, xlab="pranzo",
+ main="Boxplot del vettore pranzo", col="gray")
```

Figura 1.24: Boxplot relativo al vettore *pranzo*.

❖ Vettore *cena*

```
> quantile(cena)
   0%   25%   50%   75%  100%
 9.00 13.15 18.75 26.25 30.80
> summary(cena)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   9.00   13.15   18.75   19.51   26.25   30.80
>
> boxplot(cena, xlab="cena",
+ main="Boxplot del vettore cena", col="brown")
```

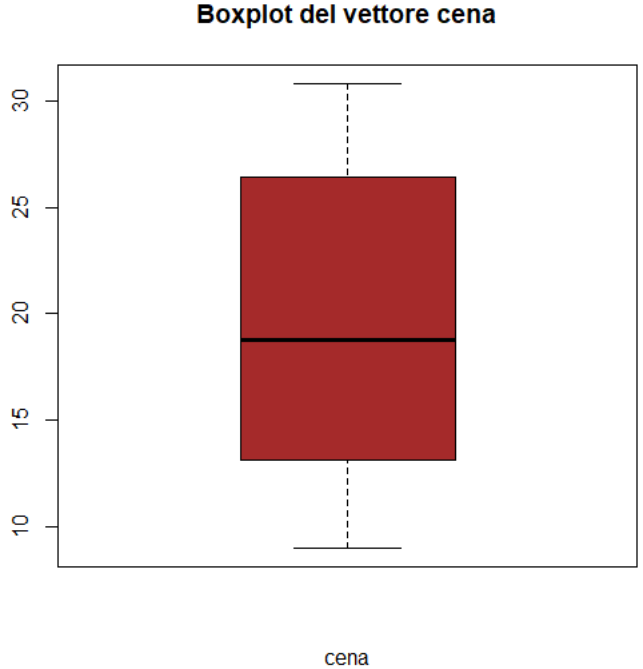


Figura 1.25: Boxplot relativo al vettore *cena*

L'analisi dei nove boxplot evidenzia non esiste la simmetria poiché la linea non si trova mai al centro della scatola.

1.2.5 Rappresentazioni grafiche per confrontare le variabili

Per confrontare le diverse variabili che descrivono insiemi di dati numerici di uno stesso fenomeno quantitativo, è possibile costruire un grafico che contiene i diversi boxplot delle distribuzioni associate alle diverse variabili.
Per esempio, si vuole analizzare il numero di persone di una stessa regione per due tipi di pasti diversi, i cui valori, sono contenuti all'interno dei vettori *lavoro* e *mensa*.

```
> plot(lavoro, pch="+", ylim=c(3,13), ylab="Confronto", col="red")
> points(mensa, pch="x", col="blue")
> legend (15 ,12 , c("Lavoro","Mensa"),pch=c("+","x"),
+ col =c("red","blue"),bg="gray",cex=0.6)
```

Il seguente codice produce il grafico in Figura 1.26, in cui vengono rappresentate assieme le persone che hanno consumato, nell'ultimo anno, pasti al lavoro e a mensa.
Tramite il parametro *pch* si è potuto rappresentare i due vettori con due simboli diversi ("+" per lavoro e "x" per mensa). La funzione **points()** permette di aggiungere punti al grafico precedente, creato con la funzione **plot()**; viene inoltre inserita una legenda con la funzione **legend()**, per descrivere al meglio il grafico. Nonostante questo, il grafico appare poco chiaro poiché, osservando la figura, è possibile notare alcuni valori sovrapposti. Per questo motivo un boxplot risulta in questo caso più efficiente e chiaro per confrontare i valori di due vettori diversi.

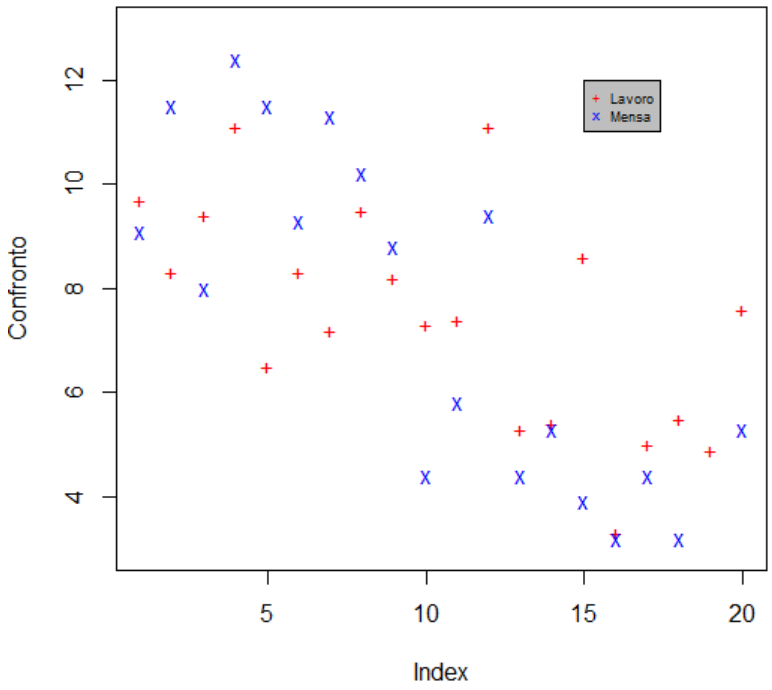


Figura 1.26: Rappresentazione dei valori nei due tipi di pasti.

Con la seguente linea di codice quindi, si produce il grafico in Figura 1.27.

```
> boxplot(lavoro, mensa, names=c("Lavoro", "Mensa"), col=c("pink", "green"))
```

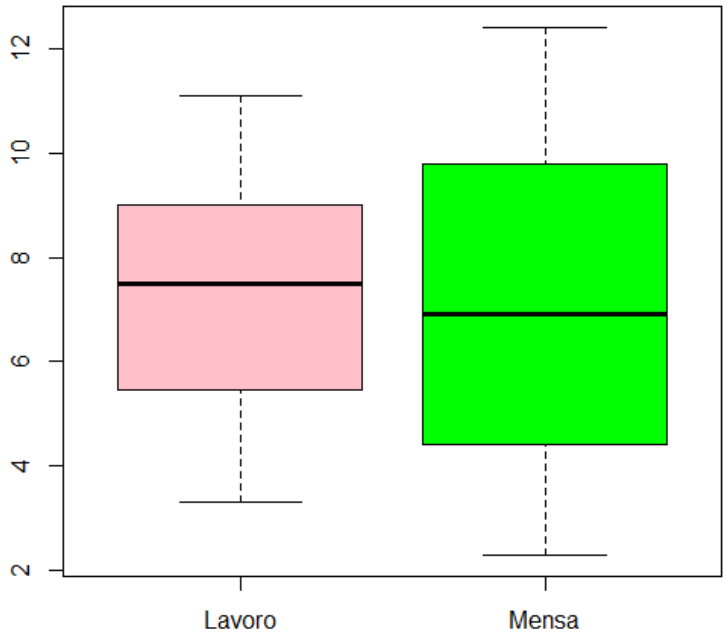


Figura 1.27: Confronto dei due boxplot relativi ai vettori *lavoro* e *mensa*.

Inoltre, applicando la funzione **summary()** ai due vettori, è possibile visualizzare le principali misure statistiche che si possono visualizzare nel grafico contenente i due boxplot.

```
> summary(lavoro)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.300   5.475   7.500   7.480   8.800  11.100
> summary(mensa)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.300   4.400   6.900   7.185   9.600  12.400
```

Da questi dati si deduce che la percentuale di persone che nell'ultimo anno ha consumato pasti al lavoro non va oltre il 11.1%, mentre quella che ha consumato pasti a mensa non va oltre l'12.4%.

1.2.6 Scatterplot

Sia X una variabile di tipo quantitativo e indichiamo con $z_1, z_2, ..., z_h$ i valori distinti da essa assunti e sia Y un'altra variabile di tipo quantitativo e indichiamo con $w_1, w_2, ..., w_k$ i valori distinti da essa assunti. Consideriamo un campione $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ costituito da n osservazioni di (X, Y). Per rappresentare graficamente le relazioni tra variabili quantitative è possibile usare i *diagrammi di dispersione* (**scatterplot**) in cui ogni coppia di osservazioni è rappresentata sotto forma di un punto in un piano euclideo. Dopo aver posto sull'asse delle ascisse la *variabile indipendente* e sull'asse delle ordinate quella *dipendente*, si disegnano dei punti in corrispondenza delle coppie, ottenendo come risultato finale, con la funzione **plot(x, y)** una nuvola di punti. Così facendo è possibile evidenziare se esiste e di quale tipo è la relazione tra le variabili.
Preso il dataset in esame, le seguenti linee di codice definiscono un dataframe contenente i 9 diversi tipi di pasti.

```
> tipiDiPasto <- data.frame(colazione_adeguata=c(81.2,79,85.4,81.3,84.5,83.3,82.7,86.2,86.8,87.4,86,
+ 83.2,84.2,80.8,74.4,79.9,78.3,78.9,76.5,78.4),
+ colazione_con_latte=c(40.6,38.7,47.3,39.7,38.9,39.3,40.6,43.9,48.4,49.6,47.2,
+ 44.9,41.9,46.5,39.6,48.8,40.8,41.1,37.2,41.9),
+ casa=c(68.1,63.4,69.2,61.8,67.8,69.3,66.9,69.2,71.8,
+ 79.3,76.7,65.8,83.2,83.9,79.3,87,84.3,84.9,84.8,81),
+ mensa=c(9.1,11.5,8,12.4,11.5,9.3,11.3,10.2,8.8,
+ 4.4,5.8,9.4,4.4,5.3,3.9,3.2,4.4,3.2,2.3,5.3),
+ ristorante=c(2.7,5.2,3.4,5.2,6.8,4.1,3.9,3,2.3,2.2,2.2,
+ 2.1,1.9,1.2,1.3,0.7,0.7,1,1.1,0.7),
+ bar=c(3.6,3.4,3.9,2.4,1.3,2.2,1.4,2.1,2.8,1.2,
+ 1.5,4.2,0.8,0.4,0.7,0.7,0.6,0.6,1.1,0.7),
+ lavoro=c(9.7,8.3,9.4,11.1,6.5,8.3,7.2,9.5,8.2,7.3,
+ 7.4,11.1,5.3,5.4,8.6,3.3,5,5.5,4.9,7.6),
+ pranzo=c(61.7,69.1,62.2,59.1,75.1,67.3,63.7,62.8,61.9,69.9,69.1,
+ 56.5,72.8,81.4,71.5,82.8,81.5,75.1,73.2,75.5),
+ cena=c(29,21,28.9,29.3,13.2,22.5,25.3,26.7,26.1,18.7,
+ 17.2,30.8,13,11.8,18.8,9.1,9,12.1,13.6,14.1))
```

La funzione **pairs()** permette la visualizzazione, su una sola finestra grafica, di più grafici per punti ottenuti mettendo il relazione tutte le coppie di variabili quantitative definite nel dataframe. Viene quindi generato uno **scatterplot** che mette in relazione le diverse variabili rappresentanti i 9 tipi di pasto. La seguente linea di codice produce quindi il grafico in Figura 1.28.

```
> pairs(tipiDiPasto, main="Scatterplot per le coppie di variabili")
```

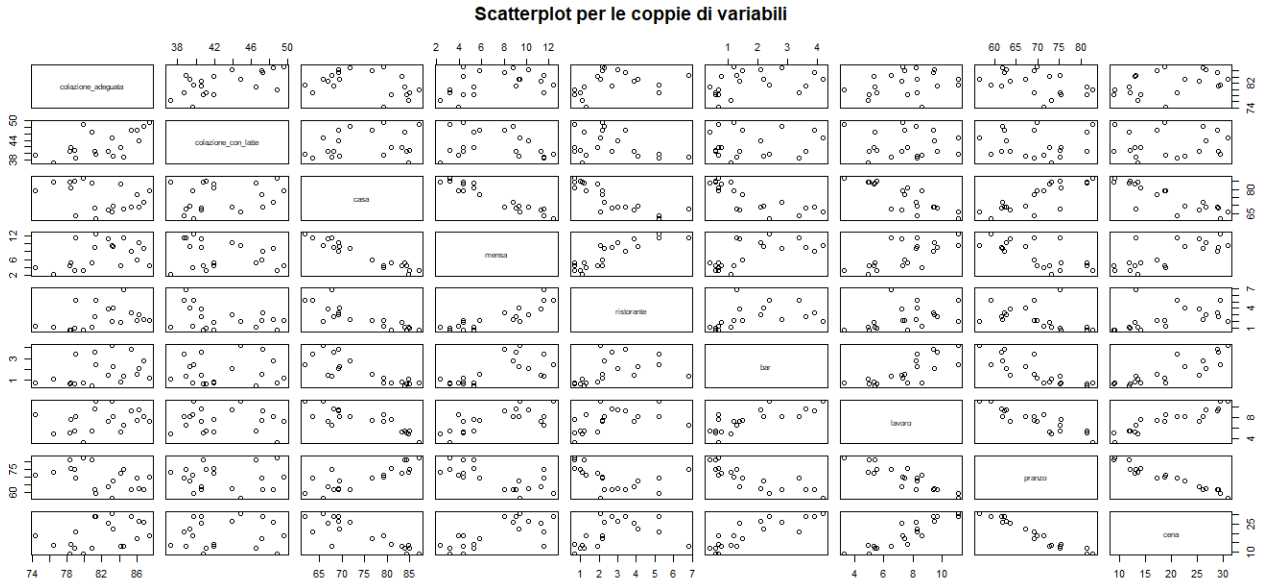


Figura 1.28: Scatterplot.

Successivamente verrà valutato se esiste una relazione tra le variabili *casa* e *mensa* e in caso positivo, verrà analizzato quale tipo di dipendenza esiste.
Verrà studiata la regressione per questa coppia di variabili quando tratteremo la statistica bivariata.

1.3 Statistica descrittiva univariata

La statistica descrittiva è costituita da un insieme di metodi di natura logica e matematica atti a raccogliere, elaborare, analizzare ed interpretare dati allo scopo di descrivere fenomeni collettivi e di estendere la descrizione di certi fenomeni osservati ad altri fenomeni dello stesso tipo non ancora osservati.

1.3.1 Funzione di distribuzione empirica

Per analizzare i *fenomeni quantitativi*, è utile definire la **funzione di distribuzione empirica**, che può essere *discreta* o *continua*.

1.3.1.1 Funzione di distribuzione empirica discreta

Nel caso *discreto* questa funzione è definita a partire dalle frequenze relative cumulate.
Consideriamo una variabile quantitativa X e indichiamo con z_1, z_2, ..., z_k i valori distinti da essa assunti e assumiamo che essi siano ordinati in ordine crescente, ossia $z_1 < z_2 < \ldots < z_k$.
Consideriamo poi un campione x_1, x_2, ..., x_k costituito da n osservazioni di X. Se indichiamo con n_i il numero di volte in cui ciascun valore z_i è presente nel campione, ossia la *frequenza assoluta* con cui esso appare nel campione, e con $f_i = n_i/n$ le *frequenze relative*; è possibile definire le **frequenze relative cumulate** nel seguente modo:

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

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 $z_1 < z_2 < \ldots < z_k$, allora la funzione di distribuzione empirica $F(x)$ è così definita:

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

dove # indica la cardinalità dell'insieme. La funzione di distribuzione empirica $F(x)$ è definita per ogni x reale ed è una *funzione a gradini* in cui ogni gradino indica quale proporzione di dati presenta un valore minore o uguale di quello indicato sull'asse delle ascisse.
La funzione empirica $F(x)$ gode delle seguenti proprietà:

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

Il linguaggio R dispone della classe **stepfun** che implementa una serie di metodi per trattare funzioni a gradino. In particolare, la funzione **ecdf()** *(empirical cumulative distribution function)* permette di disegnare il grafico della funzione di distribuzione empirica per variabili quantitative discrete.

Per le informazioni contenute nei dati presi in analisi, una funzione di distribuzione empirica discreta non risulta di particolare interesse ai fini poiché i dati risultano assumere tutti valori distinti. Rappresentiamo però la funzione di distribuzione empirica discreta del vettore *colazione_adeguata* a titolo di esempio. Usiamo quindi la funzione **ecdf()** applicata al vettore:

```
> plot(ecdf(colazione_adeguata), main="Funzione di distribuzione empirica discreta", verticals="TRUE", col="blue")
```

Ottenendo quindi la Figura 1.29

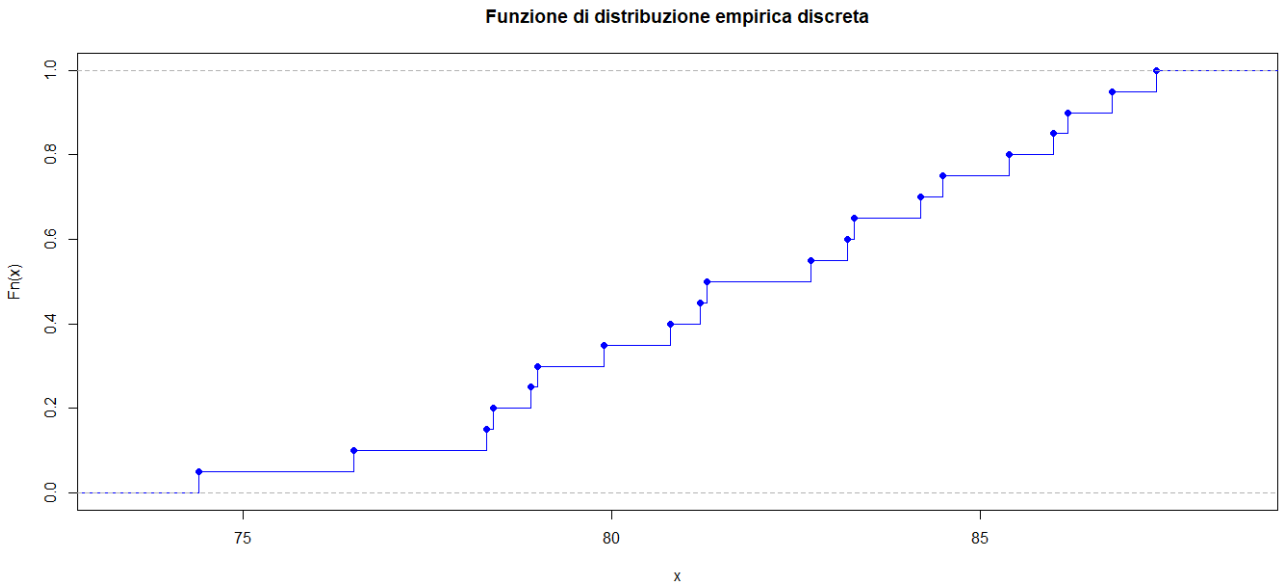


Figura 1.29: Funzione di distribuzione empirica discreta del vettore *colazione_adeguata*

Settando il parametro **verticals=TRUE,** i segmenti sono uniti grazie ad altri segmenti verticali aggiuntivi. Se invece si vuole conoscere il valore della funzione di distribuzione empirica discreta nel punto x=80, si usa il seguente comando:

```
> ecdf(colazione_adeguata)(80)
[1] 0.35
```

che, osservando il grafico, corrisponde a 0.35.

1.3.1.2 Funzione di distribuzione empirica continua

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

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

Si nota che $F(x) = 0$ per $x < z_1$, $F(x) = 1$ per $x \geq z_{k+1}$, mentre se $z_i < x < z_{i+1}$ la funzione di distribuzione empirica continua coincide con il segmento che passa per i punti (z_i, F_i) e (z_{i+1}, F_{i+1}), ossia

$$\frac{y - F_i}{x - z_i} = \frac{F_{i+1} - F_i}{z_{i+1} - z_i},$$

Ad esempio, considerando il vettore *colazione_adeguata*, introduciamo le seguenti classi: [69,74), [74, 79), [79,84), [84,89), [89,94) in cui le classi [69,74) e [89,94) sono delle classi fittizie che servono per tracciare la linea y = 0 nell'intervallo [69,74) e la linea y = 1 nell'intervallo [89,94) nel grafico della funzione di distribuzione empirica continua. Le seguenti linee di codice permettono di visualizzare le frequenze relative cumulate associate alle classi scelte, associando in primo luogo gli estremi delle classi a un vettore e successivamente a Fi un vettore che contiene le frequenze relative cumulative dei valori del vettore *colazione_adeguata*.
Per ottenere tali frequenze, si usa la funzione **cut()** con il parametro **right=FALSE** per costruire classi con intervalli chiusi a sinistra e aperti a destra. In R la funzione **cut()** permette di trasformare i dati numerici in dati qualitativi tramite la loro collocazione in opportune classi sulla base di quanto specificato nel parametro **breaks**.

```
> classiColazioneAdeguata <- c(69,74,79,84,89,94)
> Fi <- cumsum(table(cut(colazione_adeguata, breaks=classiColazioneAdeguata, right=FALSE)))/length(colazione_adeguata)
> Fi
[69,74) [74,79) [79,84) [84,89) [89,94)
   0.00    0.25    0.65    1.00    1.00
```

Con le seguenti linee di codice si ottiene invece il grafico in Figura 1.30:

```
> Fi <- c(0,Fi)
> Fi
        [69,74) [74,79) [79,84) [84,89) [89,94)
   0.00    0.00    0.25    0.65    1.00    1.00
> plot(classiColazioneAdeguata, Fi, type="b", axes=F,
+ main="Funzione di distribuzione empirica continua", col="green")
> axis(1, classiColazioneAdeguata)
> axis(2, format(Fi, digits=2))
> box()
```

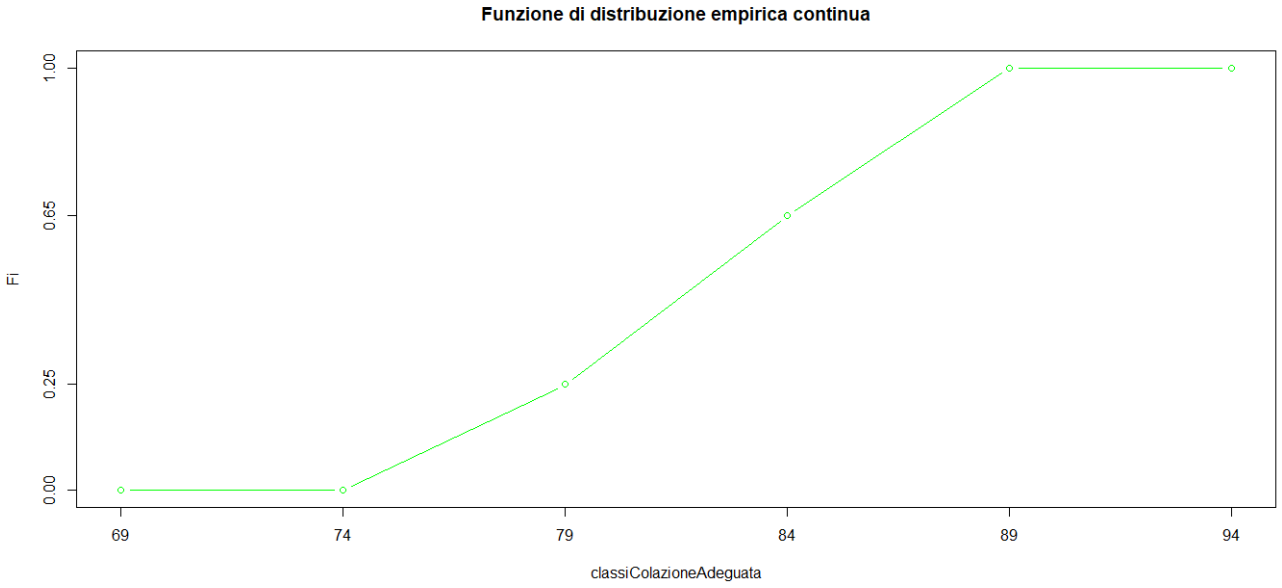


Figura 1.30: Grafico funzione di distribuzione empirica continua nel vettore *colazione_adeguata* usando le classi [69,74), [74, 79), [79,84), [84,89), [89,94).

Il comando **c(0, Fi)** permette di aggiungere uno zero all'inizio del vettore delle frequenze relative cumulative. Il parametro **type=b** permette la congiunzione di punti successivi del grafico mediante linee continue ai cui estremi sono presenti dei cerchietti, mentre il parametro **axes=F** permette di non tracciare gli assi, che verranno invece disegnati in un secondo momento.
Tramite le istruzioni **axis(1, classiColazioneAdeguata)** e **axis(2, format(Fi, digits=2))** si ottiene rispettivamente l'asse orizzontale e l'asse verticale con l'opportuna formattazione dei numeri.

Tramite **box()** invece si racchiude il grafico in un rettangolo.

1.3.2 Indici di sintesi

Informazioni aggiuntive sono fornite dagli **indici di sintesi**, che vengono distinti in *indici di posizione* e *indici di dispersione.*
Gli **indici di posizione** possono essere *centrali*, come la media, la mediana e la moda, o *non centrali* come quantili, percentili, decili e quartili. Gli **indici di dispersione** invece sono la varianza campionaria, deviazione standard campionaria e coefficiente di variazione.
Prendendo in considerazione il dataset scelto, con le seguenti linee di codice si ottiene il boxplot dei vari vettori, come mostra la Figura 1.31.

```
> par(mfrow=c(3,4))
> boxplot(colazione_adeguata, horizontal=TRUE, col="pink", main="Boxplot colazione_adeguata")
> boxplot(colazione_con_latte, horizontal=TRUE, col="blue", main="Boxplot colazione_con_latte")
> boxplot(casa, horizontal=TRUE, col="yellow", main="Boxplot casa")
> boxplot(mensa, horizontal=TRUE, col="green", main="Boxplot mensa")
> boxplot(ristorante, horizontal=TRUE, col="seagreen1", main="Boxplot ristorante")
> boxplot(bar, horizontal=TRUE, col="magenta", main="Boxplot colazione_adeguata")
> boxplot(lavoro, horizontal=TRUE, col="gray", main="Boxplot bar")
> boxplot(pranzo, horizontal=TRUE, col="red", main="Boxplot pranzo")
> boxplot(cena, horizontal=TRUE, col="brown", main="Boxplot cena")
```

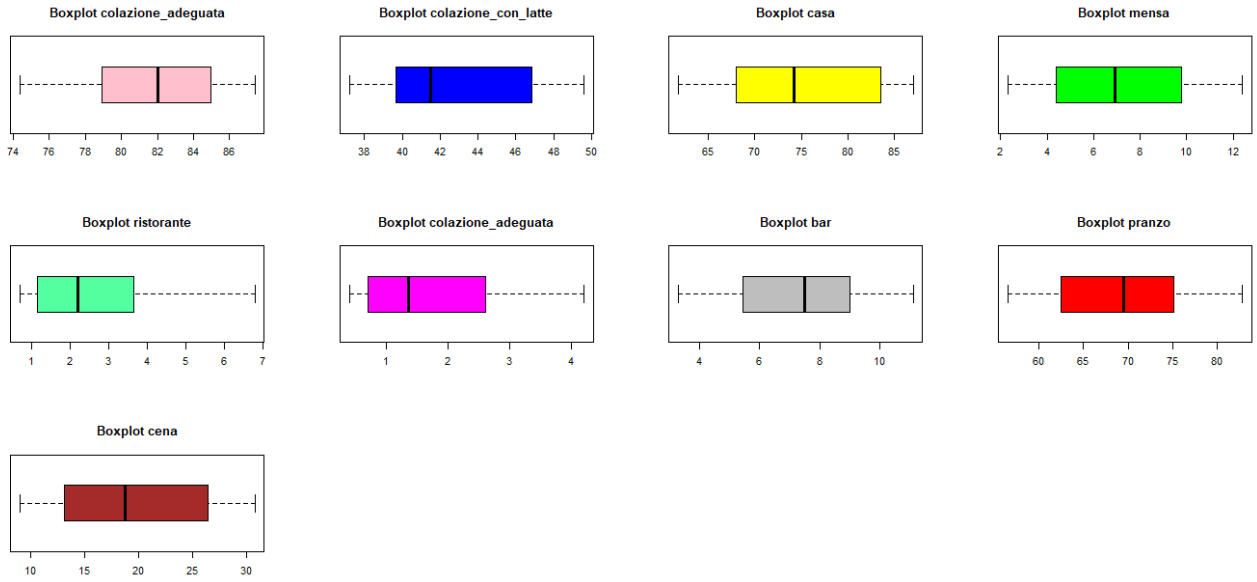


Figura 1.31: Boxplot dei vettori della matrice *matriceAbitudiniAlimentari*.

Il **boxplot**, come già detto, è un grafico relativo a caratteri quantitativi che descrive le caratteristiche salienti della distribuzione di frequenza di un dato campione numerico. Per rappresentare una distribuzione in modo sintetico, il boxplot è un'ottima possibilità: con poche informazioni, si riesce a comprendere la sua forma, simmetrica o asimmetrica che sia.
Nei grafici precedenti, è possibile osservare come i nove boxplot relativi ai 9 vettori dei dati presi in considerazione evidenzino efficacemente l'asimmetria della distribuzione dei caratteri descritti.
Gli *indici di sintesi* sono fondamentali per poter misurare quantitativamente alcune delle caratteristiche osservate qualitativamente nei grafici appena illustrati delle distribuzioni di frequenza e nei boxplot.

1.3.3 Media, mediana e moda campionarie

Media, mediana e moda campionarie sono detti *indici di posizione* o indici di *tendenza centrale* poiché descrivono attorno a quali valori è centrato l'insieme di dati.

Media

Supponiamo di avere un insieme x_1, x_2, …, x_n di n valori (dati statistici numerici), detto campione di ampiezza pari a n. La *media campionaria* è la media aritmetica di questi valori.
La media campionaria è denotata con $\bar{x}$ ed è la quantità:

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

La media campionaria gode della *proprietà di linearità*.
Inoltre se si denotano con z_1, z_2, …, z_k i valori distinti assunti dai dati, con n_1, n_2, …, n_k le frequenze assolute e con f_1, f_2, …, f_k le frequenze relative, allora possiamo scrivere:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\sum_{i=1}^{k} z_i\, n_i = \sum_{i=1}^{k} \frac{n_i}{n}\, z_i = \sum_{i=1}^{k} f_i\, z_i,$$

che mostra che la media campionaria è una media pesata dei valori distinti assunti dai dati. Ogni valore distinto usa come peso la sua frequenza relativa, ovvero la frazione dei dati uguale a tale valore. Per ogni valore x_i si definisce lo *scarto dalla media campionaria* la quantità:

$$s_i = x_i - \overline{x} \qquad (i = 1, 2, \ldots, n)$$

che indica il grado di scostamento del singolo valore x_i dalla media campionaria $\bar{x}$. Si nota immediatamente che la *somma algebrica degli scarti dalla media campionaria è sempre nulla*.
Risulta quindi:

$$\sum_{i=1}^{n} s_i = \sum_{i=1}^{n} (x_i - \overline{x}) = \sum_{i=1}^{n} x_i - n\,\overline{x} = \overline{x} - \overline{x} = 0$$

Mediana

Un altro indice di centralità di un insieme di dati è la *mediana campionaria*.
Assegnato un insieme di dati di ampiezza n, lo si ordina dal minore al maggiore. Se n è dispari, viene definita 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 $n/2 + 1$.
Questa definizione della mediana campionaria bipartisce le osservazioni in due gruppi di uguale numerosità, in modo che lo stesso numero di valori cada sia a sinistra che a destra della mediana stessa. Media e mediana campionarie sono entrambe statistiche utili per descrivere misure di centralità dei dati. La media campionaria usa tutti i dati ed è influenzata in maniera sensibile da valori eccezionalmente alti o bassi; la mediana campionaria invece dipende solo da uno o due valori centrali dei dati e non risente dei valori estremi. Inoltre, l'uso della mediana come indice per descrivere le caratteristiche dei dati ha lo svantaggio di dover prima riordinare i dati in ordine crescente, il che non è richiesto per il calcolo della media.

Moda

La moda campionaria di un insieme di dati, se esiste, è la modalità a cui è associata la frequenza, assoluta o relativa che sia, più elevata. Se esistono più modalità con frequenza massima, ognuna di esse è detta valore modale. La moda quindi rappresenta il valore prevalente nell'insieme dei dati, cioè quello che è presente più frequentemente.
La moda viene quindi usata quando si trattano dati di tipo *qualitativo*, per i quali non è possibile calcolarne media e mediana. Inoltre può non esistere e non essere unica; nel caso in cui sia unica, la distribuzione viene chiamata *unimodale*, quando ci sono due o più mode differenti si parla di distribuzione *bimodale* o *multimodale*.
La moda è un buon indice di sintesi quando si presenta con una frequenza nettamente superiore rispetto alle frequenze delle altre modalità; invece può non essere utile quando i dati sono numerosi e per la maggior parte diversi tra di loro o se le modalità presentano all'incirca tutte la stessa frequenza.
In R esistono le funzioni **mean()** e **median()** che permettono di calcolare la media e la mediana di un insieme di dati. Possiamo quindi applicare la media e la mediana al dataset preso in considerazione:

```
> mean(colazione_adeguata)
[1] 81.92
> median(colazione_adeguata)
[1] 82
> mean(colazione_con_latte)
[1] 42.845
> median(colazione_con_latte)
[1] 41.5
> mean(casa)
[1] 74.885
> median(casa)
[1] 74.25
> mean(mensa)
[1] 7.185
> median(mensa)
[1] 6.9
> mean(ristorante)
[1] 2.585
> median(ristorante)
[1] 2.2
> mean(bar)
[1] 1.78
> median(bar)
[1] 1.35
> mean(lavoro)
[1] 7.48
> median(lavoro)
[1] 7.5
> mean(pranzo)
[1] 69.61
> median(pranzo)
[1] 69.5
> mean(cena)
[1] 19.51
> median(cena)
[1] 18.75
```

È possibile notare che per ogni vettore di dati, alcuni dati sono vicini tra loro: la media e la mediana del vettore *lavoro* assumono valori 7.48 e 7.5, evidenziando una buona distribuzione dei dati.
Non esiste invece in R una funzione per calcolare la moda di una distribuzione di dati, dato che quest'ultima è ricavabile osservando il grafico delle frequenze assolute.

Definiamo quindi una funzione che calcoli i valori modali di un vettore numerico.

```
> calcoloModa <-function (v){
+ y <-table (v)
+ z <-which (y== max (y))
+ return (c(z))
+ }
```

Possiamo quindi applicare questa funzione al dataset che abbiamo preso in considerazione:

```
> calcoloModa(colazione_adeguata)
74.4 76.5 78.3 78.4 78.9   79 79.9 80.8 81.2 81.3 82.7 83.2 83.3 84.2 84.5 85.4
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
  86 86.2 86.8 87.4
  17   18   19   20
> calcoloModa(colazione_con_latte)
40.6 41.9
   7   10
> calcoloModa(casa)
69.2 79.3
   7   11
> calcoloModa(mensa)
4.4
  4
> calcoloModa(ristorante)
0.7
  1
> calcoloModa(bar)
0.7
  3
> calcoloModa(lavoro)
 8.3 11.1
  13   18
> calcoloModa(pranzo)
69.1 75.1
   9   14
> calcoloModa(cena)
   9  9.1 11.8 12.1   13 13.2 13.6 14.1 17.2 18.7 18.8   21 22.5 25.3 26.1 26.7
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
28.9   29 29.3 30.8
  17   18   19   20
```

Da come è possibile osservare, la moda è un buon indice di sintesi nel caso dei vettori *colazione_con_latte*, *casa*, *mensa*, *ristorante*, *bar*, *lavoro e pranzo* in quanto i valori modali sono presenti con una frequenza nettamente maggiore rispetto alle frequenze degli altri valori assunti dal campione.
Al contrario, la moda campionaria non risulta essere utile nel caso dei vettori *cena* e *colazione_adeguata*, in quanto i dati sono numerosi e per la maggior parte diversi tra loro; infatti tutte le modalità presentano all'incirca la stessa frequenza e quindi in questo caso la moda non è un indice di sintesi significativo.

Adesso si può descrivere la forma della distribuzione di frequenza confrontando media e mediana campionarie. Se le due misure risultano essere uguali, la distribuzione di frequenza tende ad essere simmetrica; se invece la media campionaria è sensibilmente maggiore alla mediana campionaria, la distribuzione di frequenza è più sbilanciata verso destra, altrimenti verso sinistra.

	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
Media	81.92	42.845	74.885	7.185	2.585	1.78	7.48	69.61	19.51
Mediana	82	41.5	74.25	6.9	2.2	1.35	7.5	69.5	18.75

1.3.3.1 Mediana per una distribuzione di frequenze

Un modo di procedere differente per definire la mediana consiste nel considerare le frequenze relative cumulate. Sia X una variabile quantitativa e siano z_1, z_2, ..., z_k le modalità distinte da essa assunte, con $z_1 < z_2 < \ldots < z_k$. Consideriamo un campione x_1, x_2, ..., x_n, siano:

$$F_i = f_1 + f_2 + \ldots + f_i (i = 1, 2, \ldots, k)$$

le frequenze relative cumulate.

La *mediana* per una *distribuzione di frequenze* è definita come la modalità i-esima (i=1, 2, ..., k) che soddisfa la doppia disuguaglianza:

$$F_{i-1} < 0.5, F_i \geq 0.5$$

Come si evince dalla definizione, la mediana di una distribuzione di frequenza è un valore di sintesi che indica un punto centrale intorno al quale si dispone la distribuzione di frequenza.

La mediana di una distribuzione di frequenza può essere individuata graficamente a partire dalla *funzione di distribuzione empirica discreta*.
Si traccia la funzione di distribuzione empirica e sull'asse delle ordinate si individua il punto 0.5 e da questo si traccia una linea orizzontale. Il minimo valore osservato sulle ascisse, la cui funzione di distribuzione empirica supera 0.5 è proprio la mediana per una distribuzione di frequenze.
Analizzando i valori presi in considerazione, attraverso le seguenti righe di codice si ottiene il grafico in Figura 1.32.

```
> par(mfrow=c(3,4))
> plot(ecdf(colazione_adeguata), main="Funzione di distribuzione empirica\n discreta di colazione_adeguata", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(colazione_con_latte), main="Funzione di distribuzione empirica\n discreta di colazione_con_latte", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(casa), main="Funzione di distribuzione empirica\n discreta di casa", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(mensa), main="Funzione di distribuzione empirica\n discreta di mensa", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(ristorante), main="Funzione di distribuzione empirica\n discreta di ristorante", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(bar), main="Funzione di distribuzione empirica\n discreta di bar", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(lavoro), main="Funzione di distribuzione empirica\n discreta di lavoro", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(pranzo), main="Funzione di distribuzione empirica\n discreta di pranzo", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
> plot(ecdf(cena), main="Funzione di distribuzione empirica\n discreta di cena", verticals=TRUE, col="red")
> abline(h=0.5, lty=2, col="blue")
```

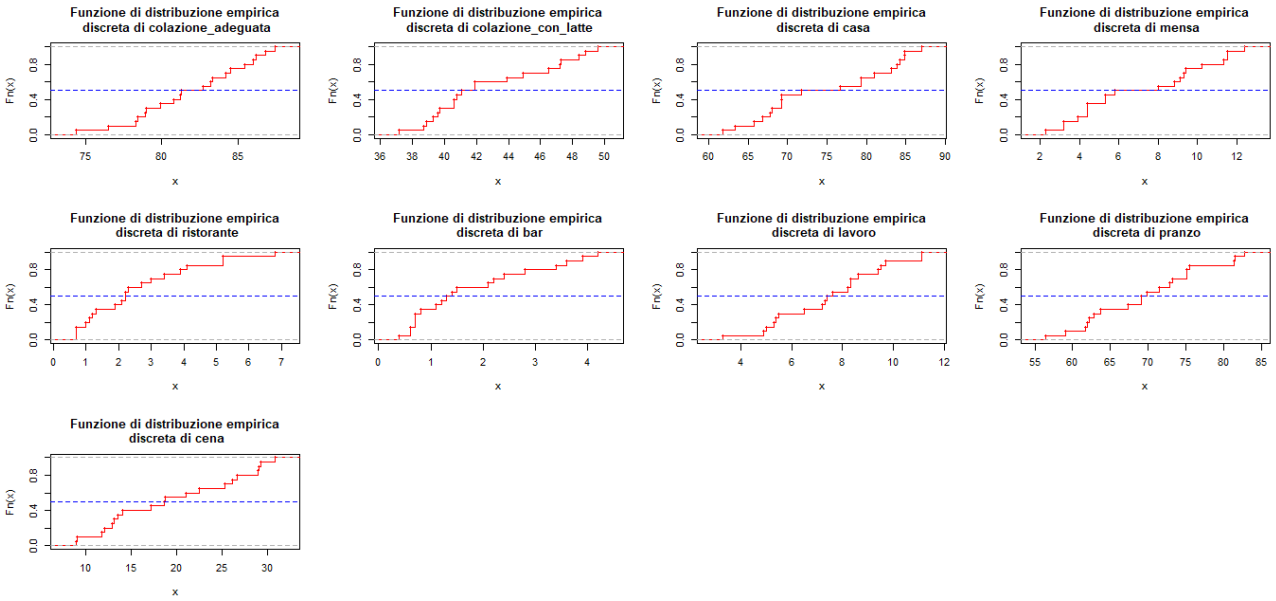


Figura 1.32: Mediana per la distribuzione di frequenze dei vettori della matrice *matriceAbitudiniAlimentari*.

Attraverso la funzione **quantile()**, si risale all'ascissa che corrisponde all'ordinata 0.5 tracciata con **abline().**

```
> quantile(colazione_adeguata, 0.5, type=1)
 50%
81.3
> quantile(colazione_con_latte, 0.5, type=1)
 50%
41.1
> quantile(casa, 0.5, type=1)
 50%
71.8
> quantile(mensa, 0.5, type=1)
50%
5.8
> quantile(ristorante, 0.5, type=1)
50%
2.2
> quantile(bar, 0.5, type=1)
50%
1.3
> quantile(lavoro, 0.5, type=1)
50%
7.4
> quantile(pranzo, 0.5, type=1)
 50%
69.1
> quantile(cena, 0.5, type=1)
 50%
18.7
```

1.3.4 Quantili, percentili, decili e quartili

Oltre la mediana, che è quel valore che divide a metà un insieme di dati ordinati, si possono definire altri indici di posizione, detti *quantili*, che dividono l'insieme dei dati ordinati in un fissato numero di parti uguali.
Sia X una variabile quantitativa e sia $x_1, x_2, ..., x_n$ 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 $\alpha-1$ numeri che consentono tale suddivisione sono i quantili di ordine α. Ad esempio, possiamo suddividere i dati in α = 4 parti mediante 3 quantili (detti *quartili*), oppure in α = 10 parti mediante 9 quantili (detti *decili*) oppure anche in α = 100 parti mediante 99 quantili (detti *percentili*). I *quantili* (percentili) sono indici di posizione non centrali utilizzati per insiemi numerosi di dati.
In R esistono 9 differenti algoritmi per calcolare i quantili ottenibili utilizzando la funzione **quantile(v, probs = , type = j)**, dove *v* è un vettore numerico, *probs* è il vettore di probabilità e j=1, 2, …, 9 è il tipo di algoritmo selezionato. R utilizza di default per il calcolo dei quantili **l'algoritmo di tipo 7**, basato su tecniche di interpolazione tra i punti.
I percentili che vengono maggiormente utilizzati sono il 25-*esimo*, il 50-*esimo* e il 75-*esimo*, detti rispettivamente primo quartile (Q1), il secondo quartile (Q2) e il terzo quartile (Q3). Se si omette *probs*, vengono di default calcolati i quartili con *type = j* e la funzione restituisce il *minimo*, il *massimo* e i tre *quartili* Q1, Q2 e Q3. Se si omette *type*, vengono di default calcolati i percentili specificati nel vettore delle probabilità utilizzando di default l'algoritmo di tipo 7.
Si consideri un campione di ampiezza *n*, ordinato in ordine crescente. Con il termine P_k viene indicato il percentile di interesse. Calcolando l'indice *h* si ottiene che:

L'algoritmo di tipo 2 calcola i percentili come:

(v[h] + v[h+1])/2, se $h = np$ è un intero;
v[h*] (cioè si arrotonda $h = np$ per eccesso al primo intero successivo) se $h = np$ non è un intero.

L'algoritmo di tipo 7 calcola i percentili come:

v[h*] + (h- h*) × [v(h* + 1) - v(h*)], dove h* rappresenta il più grande intero minore o uguale di h.

L'algoritmo di tipo 1 invece, assegnata una probabilità p, $0 < p < 1$, calcola il quantile di ordine p come la modalità i-esima che soddisfa la doppia disuguaglianza

$$F_{i-1} < p, \qquad F_i \geq p.$$

Si può facilmente mostrare che per p = 0.5 e k = 50, l'algoritmo di tipo 2 e l'algoritmo di tipo 7 conducono allo stesso risultato nel calcolo della mediana.

Applicando la funzione **quantile()** ai nove vettori, si ottiene il seguente risultato:

```
> quantile(colazione_adeguata)
    0%    25%    50%    75%   100%
74.400 78.975 82.000 84.725 87.400
> quantile(colazione_con_latte)
    0%    25%    50%    75%   100%
37.200 39.675 41.500 46.675 49.600
> quantile(casa)
    0%    25%    50%    75%   100%
61.800 68.025 74.250 83.375 87.000
> quantile(mensa)
  0%  25%  50%  75% 100%
 2.3  4.4  6.9  9.6 12.4
> quantile(ristorante)
   0%   25%   50%   75%  100%
0.700 1.175 2.200 3.525 6.800
> quantile(bar)
  0%  25%  50%  75% 100%
0.40 0.70 1.35 2.50 4.20
> quantile(lavoro)
    0%    25%    50%    75%   100%
 3.300  5.475  7.500  8.800 11.100
> quantile(pranzo)
   0%   25%   50%   75%  100%
56.50 62.65 69.50 75.10 82.80
> quantile(cena)
   0%   25%   50%   75%  100%
 9.00 13.15 18.75 26.25 30.80
```

È possibile osservare come i due diversi tipi di algoritmi, applicati sullo stesso vettore, possono produrre risultati diversi. Ad esempio, applicando rispettivamente l'algoritmo di tipo 7 e quello di tipo 2 al vettore *casa*

```
> quantile(casa)
    0%    25%    50%    75%   100%
61.800 68.025 74.250 83.375 87.000
> quantile(casa, type=2)
   0%   25%   50%   75%  100%
61.80 67.95 74.25 83.55 87.00
```

si nota come il calcolo dei quartili coincida per il secondo, ma risulta essere diverso per il primo e il terzo; infatti nell'algoritmo di tipo 7 il primo risulta essere uguale a 68.025 e il terzo risulta essere uguale a 83.375. Nell'algoritmo di tipo 2 invece il primo quartile è uguale a 67.95 mentre il terzo a 83.55.

1.3.5 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 di una distribuzione di frequenza sono la *varianza campionaria* e la *deviazione standard campionaria*, detta anche *scarto quadratico medio campionario*.
Assegnato un insieme di dati numerici x_1, x_2, ..., x_n, si definisce *varianza campionaria* e si denota con s^2 la quantità:

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

dove $\bar{x}$ denota la media campionaria dei dati.
Inoltre si definisce *deviazione standard campionaria*, la radice quadrata della varianza campionaria, cioè:

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

Varianza campionaria e deviazione standard campionaria sono detti *indici di dispersione* o *indici di variabilità* poiché misurano la dispersione dei dati intorno alla media. La varianza e la deviazione standard sono tanto più grandi quanto più i dati si discostano dalla media. I valori della varianza campionaria s^2 e della deviazione standard campionaria s dipendono dall'unità di misura dei dati. In particolare, la deviazione standard campionaria s misura la dispersione dei dati con la stessa unità di misura dei dati sperimentali e quindi con la stessa unità di misura della media campionaria. Inoltre, la varianza campionaria, a differenza delle media campionaria, non gode della proprietà di linearità; infatti se si considera un insieme di dati tale che:

$$y_i = ax_i + b$$

si ottiene che $s^2_y = a^2 + s^2_x$.
Pertanto, sommare una costante a ciascuno dei dati non fa cambiare la varianza campionaria, mentre moltiplicare ciascuno dei dati per un fattore costante fa sì che la varianza campionaria dell'insieme iniziale dei dati risulta moltiplicata per il quadrato di tale fattore.

In R è possibile calcolare la *varianza campionaria* di un vettore numerico *v* attraverso la funzione **var(v)**. La funzione **sd(v)** serve invece per calcolare la *deviazione standard campionaria*. Applicandole ai vettori presi in analisi, si ottengono i seguenti valori:

```
> var(colazione_adeguata)
[1] 13.27537
> var(colazione_con_latte)
[1] 15.0605
> var(casa)
[1] 69.21608
> var(mensa)
[1] 10.81503
> var(ristorante)
[1] 2.965553
> var(bar)
[1] 1.494316
> var(lavoro)
[1] 4.546947
> var(pranzo)
[1] 59.07674
> var(cena)
[1] 54.07463
```

```
> sd(colazione_adeguata)
[1] 3.643538
> sd(colazione_con_latte)
[1] 3.880786
> sd(casa)
[1] 8.31962
> sd(mensa)
[1] 3.288621
> sd(ristorante)
[1] 1.722078
> sd(bar)
[1] 1.222422
> sd(lavoro)
[1] 2.132357
> sd(pranzo)
[1] 7.686139
> sd(cena)
[1] 7.353546
```

Per confrontare le variazioni esistenti tra diversi campioni di dati è utile introdurre il *coefficiente di variazione*.
Assegnato un insieme di dati numerici $x_1, x_2, \ldots, x_n$, si definisce *coefficiente di variazione* il rapporto tra la *deviazione standard campionaria* e *il modulo della media campionaria*, ossia:

$$\mathrm{CV} = \frac{s}{|\bar{x}|}$$

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. Dalla formula deriva che il coefficiente di variazione è un indice di dispersione che ha senso soltanto per campioni aventi la *media campionaria non nulla*. In R non è definita una funzione che calcola il coefficiente di variazione. Tale funzione può essere comunque facilmente implementata in R nel seguente modo:

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

Possiamo quindi applicarla ai vettori della matrice:

```
> cv(colazione_adeguata)
[1] 0.04447678
> cv(colazione_con_latte)
[1] 0.09057734
> cv(casa)
[1] 0.1110986
> cv(mensa)
[1] 0.4577064
> cv(ristorante)
[1] 0.666181
> cv(bar)
[1] 0.686754
> cv(lavoro)
[1] 0.2850745
> cv(pranzo)
[1] 0.1104172
> cv(cena)
[1] 0.3769116
```

Nella seguente tabella sono riportate la media, la varianza, la deviazione standard e il coefficiente di variazione dei vettori.

	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
Media	81.92	42.845	74.885	7.185	2.585	1.78	7.48	69.61	19.51
Varianza	13.27537	15.0605	69.21608	10.81503	2.965553	1.494316	4.546947	59.07674	54.07463
Deviazione standard	3.643538	3.880786	8.31962	3.288621	1.722078	1.222422	2.132357	7.686139	7.353546
Coefficiente di variazione	0.04447678	0.09057734	0.1110986	0.4577064	0.666181	0.686754	0.2850745	0.1104172	0.3769116

Come è possibile notare dalla tabella, il coefficiente di variazione più alto si ottiene col vettore *bar*, i cui dati si discostano maggiormente dalla media campionaria, come è possibile intuire dal relativo boxplot in Figura 1.22.

1.3.6 Forma di una distribuzione di frequenza

La media, la mediana e la moda sono utili a comprendere la forma delle distribuzioni di frequenza, nel senso che differenze sostanziali tra questi indici indicano uno *sbilanciamento eccessivo della distribuzione di frequenza verso destra o verso sinistra*. Inoltre, anche se la media e la mediana coincidono, le misure di dispersione dei dati possono essere sostanzialmente differenti implicando una differente forma delle distribuzioni di frequenza.
Esistono degli *indici statistici* che permettono di misurare quando una distribuzione di frequenza presenta *simmetria* o *asimmetria* oppure se essa è *più* o *meno piccata*.

Un indice che permette di misurare la *simmetria* di una distribuzione di frequenza è la *skewness campionaria*.
Assegnato un insieme di dati numerici x_1, x_2, ..., $x_k n$, si definisce *skewness campionaria* il valore:

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

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

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

Si può quindi dedurre che:

- se γ_1 assume il valore 0, la distribuzione di frequenza è *simmetrica*;
- se $\gamma_1 > 0$, si ha *l'asimmetria positiva* (ossia la distribuzione di frequenza ha la coda di destra più allungata);
- se $\gamma_1 < 0$ si ha *l'asimmetria negativa* (ossia la distribuzione di frequenza ha la coda di sinistra più allungata).

Si nota che γ_1 è un *indice adimensionale*, ossia è indipendente dall'unità di misura dei dati. In R, il calcolo della *skewness campionaria* viene implementato nel seguente modo:

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

Facendo riferimento ai vettori dei dati in analisi, si ottengono i seguenti valori:

```
> skw(colazione_adeguata)
[1] -0.2620831
> skw(colazione_con_latte)
[1] 0.4250065
> skw(casa)
[1] 0.004962314
> skw(mensa)
[1] 0.1097116
> skw(ristorante)
[1] 0.8914946
> skw(bar)
[1] 0.7111131
> skw(lavoro)
[1] -0.05308885
> skw(pranzo)
[1] 0.1101431
> skw(cena)
[1] 0.1198655
```

da cui è possibile notare che la *skewness* non risulta essere mai nulla. Di conseguenza, la distribuzione di frequenza non è *mai simmetrica*. Invece, la *skewness* presenta un'*asimmetria negativa* per i vettori *colazione_adeguata* e *lavoro;* presenta invece *asimmetria positiva* per i rimanenti vettori.
Un indice che permette di misurare la densità dei dati intorno alla media è la *curtosi campionaria*.

Assegnato un insieme di dati numerici x_1, x_2, ..., $x_k n$, si definisce *curtosi campionaria* il valore:

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

dove

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

avendo denotato con m_4 il momento centrato campionario di ordine 4.

Gli indici γ_2 e β_2 permettono di confrontare *la distribuzione di frequenza dei dati* con *una densità di probabilità normale*, caratterizzata da $\beta_2 = 3$ e *indice di curtosi* $\gamma_2 = 0$. Infatti, se risulta:

- $\beta_2 < 3$, $\gamma_2 < 0$, la distribuzione di frequenza si definisce *platicurtica*, ossia la distribuzione di frequenza è più *piatta di una normale*;
- $\beta_2 > 3$, $\gamma_2 > 0$, la distribuzione di frequenza si definisce *leptocurtica*, ossia la distribuzione di frequenza è più *piccata di una normale*;
- $\beta_2 = 3$, $\gamma_2 = 0$, la distribuzione di frequenza si definisce *normocurtica*, ossia *piatta come una normale*.

Si nota che β_2 è un *indice adimensionale*, ossia è indipendente dall'unità di misura dei dati. In R, il calcolo della curtosi campionaria viene implementato nel seguente modo:

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

Facendo riferimento ai vettori dei dati in analisi, si ottengono i seguenti valori:

```
> curt(colazione_adeguata)
[1] -0.8412845
> curt(colazione_con_latte)
[1] -1.231185
> curt(casa)
[1] -1.519087
> curt(mensa)
[1] -1.452644
> curt(ristorante)
[1] -0.0160571
> curt(bar)
[1] -0.8527998
> curt(lavoro)
[1] -0.7336669
> curt(pranzo)
[1] -0.9604487
> curt(cena)
[1] -1.426161
```

da cui si evince una curtosi negativa per tutti i vettori, che si definisce quindi *platicurtica*, ossia più piatta di una normale.

1.4 Statistica descrittiva bivariata

La statistica descrittiva bivariata si occupa dei metodi grafici e statistici atti descrivere le relazioni che intercorrono tra due variabili.
Siano X e Y due variabili di tipo quantitativo. Si consideri un campione (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) costituito da n osservazioni di (X, Y).
Le relazioni tra variabili quantitative possono essere rappresentate graficamente mediante *diagrammi di dispersione (scatterplot)* in cui ogni coppia di osservazioni viene rappresentata sotto forma di un punto o un cerchietto in un piano euclideo. Dopo aver scelto la variabile da porre sulle *ascisse* (*variabile indipendente*) e la variabile da porre sulle *ordinate* (*variabile dipendente*), si disegnano dei punti in corrispondenza delle coppie (x_i, y_i). Il risultato finale è una *nuvola di punti* che può essere ottenuta con la funzione **plot(x, y)**. Il grafico che si ottiene mira ad evidenziare se le *coppie di punti* presentano *qualche forma di regolarità*. Inoltre, il grafico di dispersione mostra se *esiste una relazione tra le variabili* e di quale tipo è tale relazione (lineare, quadratica, ...).
Si consideri la percentuale di persone nelle 20 regioni italiane che nell'anno 2017 hanno consumato almeno una volta pasti a *casa* o in *mensa*.
Sono già stati calcolati gli indici statistici di posizione e di dispersione relativi alle singole variabili. Sono riportati nella seguente tabella:

	Casa	Mensa
Media	74.885	7.185
Mediana	74.25	6.9
Deviazione standard	8.31962	3.288621

Successivamente viene realizzato lo *scatterplot* considerando **casa** come variabile *indipendente* e **mensa** come variabile *dipendente*.

Sullo *scatterplot* sono tracciate anche delle linee orizzontali e verticali in corrispondenza delle mediane campionarie e delle medie campionarie dei due vettori. Le seguenti linee di codice

```
> plot(tipiDiPasto$casa, tipiDiPasto$mensa,
+ main="mensa in funzione di casa",
+ xlab="mensa", ylab="casa", col="red"
+ )
> abline(v=median(tipiDiPasto$casa), lty=1, col="green")
> abline(v=mean(tipiDiPasto$casa), lty=2, col="blue")
> abline(h=median(tipiDiPasto$mensa), lty=1, col="green")
> abline(h=mean(tipiDiPasto$mensa), lty=2, col="blue")
> legend(76,10, c("Mediana","Media"), pch=0, col=c("green","blue"), cex=0.8)
```

producono lo scatterplot in Figura 1.33.

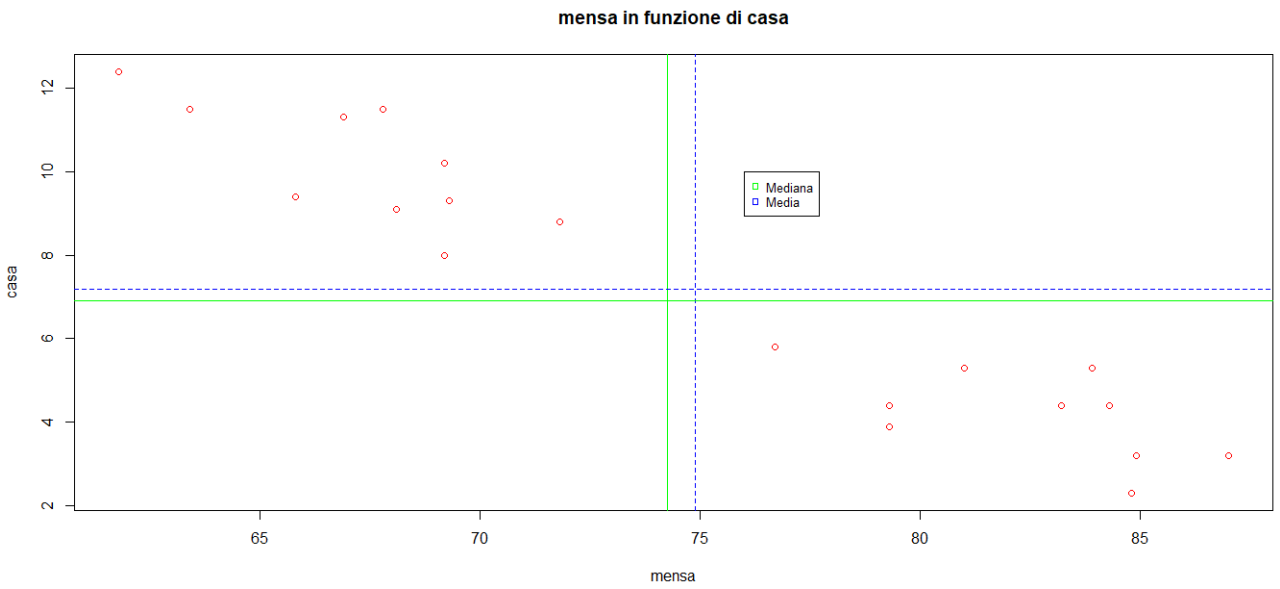


Figura 1.33: Scatterplot dei vettori *casa* e *mensa*

Si nota che i dati sembrano posizionati intorno ad una retta discendente e ciò induce a pensare che esiste una correlazione lineare negativa tra le variabili.

1.4.1 Covarianza e correlazione campionaria

Spesso 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 o scatterplot.
Per ottenere una misura quantitativa della correlazione tra le variabili si considera la *covarianza campionaria*:

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

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

Se l'intero campione presenta una *forte correlazione*, c'è da aspettarsi che la sommatoria assuma un valore molto positivo o molto negativo. Di norma, si usa normalizzare tale sommatoria dividendo per n - 1, in maniera tale da ottenere la varianza campionaria nel caso in cui $x_i = y_i$ per ogni i=1, 2, ..., n.

La covarianza campionaria può avere segno positivo, negativo o nullo.

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

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

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

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

così definito:
Il coefficiente di correlazione ha lo *stesso segno della covarianza*. In generale:

- se $r_{xy} > 0$, si dice che le variabili sono *correlate positivamente*;
- se $r_{xy} < 0$, le variabili sono *correlate negativamente*;
- se $r_{xy} = 0$, le variabili sono *non correlate*

Il coefficiente di correlazione r_{xy} gode delle seguenti proprietà:

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

La proprietà (1) afferma che il coefficiente di correlazione campionario è compreso nell'intervallo [-1, 1].
Le proprietà (2) e (3) mostrano che i valori limite -1 e +1 sono effettivamente raggiunti solo quando tra *X* e *Y* sussiste una *relazione lineare*, ossia quando *i punti dello scatterplot giacciono tutti su di una retta*.
La proprietà (4) afferma che il quadrato del coefficiente di correlazione non cambia se sommiamo costanti o moltiplichiamo per costanti tutti i valori di *X* e/o di *Y*. Ciò significa che il *coefficiente di correlazione non dipende dalle unità di misura scelta per rappresentare i dati.*

Va ricordato che il coefficiente di correlazione campionario r_{xy} misura *la forza del legame di natura lineare esistente tra due variabili quantitative*. Eventuali relazioni tra le variabili che assumono una forma curvilinea non possono essere individuate con tale coefficiente.
Riassumendo si può dire che il segno di r_{xy} indica la direzione della retta interpolante e indica la presenza di una tra le seguenti situazioni:

- $r_{xy} = 1$ (correlazione perfetta positiva) tutti i punti sono allineati su una linea retta ascendente;

- r_{xy} compreso tra 0 e 1 estremi esclusi (correlazione positiva) i punti (x_i, y_i) sono posizionati in una nuvola attorno ad una *linea retta interpolante ascendente* (in tal caso x_i e y_i tendono ad essere grandi e piccoli insieme);

- $r_{xy} = 0$ (nessuna correlazione) i punti sono completamente dispersi in una nuvola che non presenta alcuna evidente direzione di natura lineare;

- r_{xy} compreso tra -1 e 0 estremi esclusi (correlazione negativa) i punti (x_i, y_i) sono posizionati in una nuvola attorno ad una *linea retta interpolante discendente* (in tal caso x_i è grande y_i è piccolo e viceversa);

- $r_{xy} = -1$ (correlazione perfetta negativa) tutti i punti sono allineati su una

linea retta discendente;

In R, le *covarianze campionarie* e le *correlazioni campionarie* fra una coppia di variabili numeriche *X* e *Y* possono essere ottenute con le funzioni **cov(X, Y)** e **cor(X, Y)**.
Si considerano nuovamente i vettori *casa* e *mensa* e vengono calcolate la covarianza campionaria e il coefficiente di correlazione campionario.

```
> cov(tipiDiPasto$casa, tipiDiPasto$mensa)
[1] -26.14445
> cor(tipiDiPasto$casa, tipiDiPasto$mensa)
[1] -0.9555693
```

I dati dei due vettori *casa* e *mensa* sono negativamente correlati essendo la covarianza campionaria uguale a -26.14445. Inoltre, il coefficiente di correlazione è uguale a -0.9555693 ed è compreso tra -1 e 0, ciò vuol dire che vi è una correlazione negativa e che i punti (x_i, y_i) sono posizionati in una nuvola attorno ad una linea retta interpolante discendente.

Gli *scatterplot* sono dei potenti mezzi per visualizzare le eventuali relazioni che possono intercorrere tra variabili quantitative. Infatti, alcune volte osservando il grafico si nota che i punti si dispongono attorno a qualche linea orientata in qualche direzione. Tuttavia per avere un'idea più accurata del fenomeno, è necessario utilizzare altre tecniche statistiche in grado di misurare con maggiore precisione questo legame.

Il *modello lineare* viene di solito utilizzato per spiegare, descrivere, o anche prevedere un andamento futuro sulla base della relazione che si instaura tra una variabile *Y*, chiamata variabile dipendente, e una o più altre variabili che assumono il significato di variabili indipendenti X_1, X_2, ..., X_p. Nel caso in cui p = 1, l'analisi prende il nome di *regressione semplice*, mentre se p = 2, 3... si parla di *regressione multipla.*

1.4.2 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:

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

dove
- α è l'intercetta;
- β è il coefficiente angolare. 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 di regressione decrescente;
  - un coefficiente angolare nullo, $\beta = 0$, indica una retta orizzontale.

L'*intercetta* α 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*.
I *coefficienti di regressione* sono i valori α e β per i quali la somma *Q* dei quadrati degli errori

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

sia minima, dove *n* è il numero di osservazioni, $(x_1, x_2, ..., x_n)$ sono i valori osservati della variabile *X* e $(y_1, y_2, ..., y_n)$ sono i valori osservati della variabile *Y*. Derivando *Q* rispetto a α e β e

uguagliando a zero le derivate parziali ottenute si ha:

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

Si nota quindi che le medie campionarie, le deviazioni standard campionarie e il coefficiente di correlazione permettono di stimare i parametri α e β della retta di regressione. Facendo riferimenti ai vettori precedenti, le seguenti linee di codice calcolano il coefficiente angolare β e l'intercetta α:

```
> beta <-(sd(tipiDiPasto$mensa)/sd(tipiDiPasto$casa))*
+ cor(tipiDiPasto$casa,tipiDiPasto$mensa)
> alpha <-mean (tipiDiPasto$mensa)-beta*mean (tipiDiPasto$casa)
> c(alpha , beta)
[1] 35.4707245 -0.3777222
```

Dal momento che $\beta < 0$, la retta di regressione è *decrescente*. Attraverso α si può asserire che la retta di regressione interseca l'asse delle y nel punto 35.4707245, quindi:

$$Y = 35.4707245 - 0.3777222X$$

Per eseguire le analisi di regressione lineare, si utilizza la funzione **lm(y˜x)** (linear model), in cui l'argomento indica che *y dipende da x*, ossia che x è la *variabile indipendente* e y quella *dipendente*. Applichiamola quindi ai vettori in analisi:

```
> lm(tipiDiPasto$mensa~tipiDiPasto$casa)

Call:
lm(formula = tipiDiPasto$mensa ~ tipiDiPasto$casa)

Coefficients:
     (Intercept)  tipiDiPasto$casa
         35.4707           -0.3777
```

si nota che la funzione restituisce come attributi i coefficienti di regressione calcolati in precedenza attraverso la formula ricavata.
La funzione **lm()** fornisce anche altri attributi dell'oggetto linear model, da cui è possibile estrarre i singoli valori

```
> linearmodel <- lm(tipiDiPasto$mensa~tipiDiPasto$casa)
> attributes(linearmodel)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"

>
> linearmodel$coefficients
     (Intercept) tipiDiPasto$casa
      35.4707245       -0.3777222
```

La rappresentazione della retta $Y = 35.4707245 – 0.3777222X$, può essere aggiunta allo scatterplot facendo uso della funzione **abline(lm(y˜x))**;

```
> plot(tipiDiPasto$casa, tipiDiPasto$mensa, main="Retta di regressione",
+ xlab="casa", ylab="mensa", col="magenta")
> abline(lm(tipiDiPasto$mensa~tipiDiPasto$casa), col="blue")
```

Ottenendo la Figura 1.34

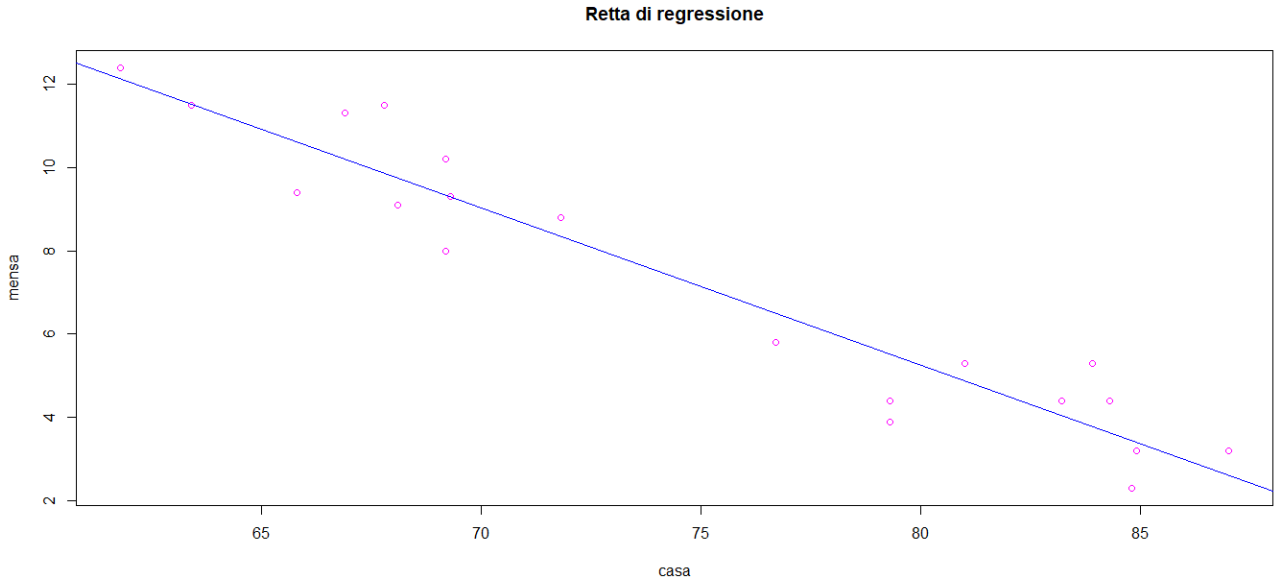


Figura 1.34: Scatterplot retta interpolante stimata (retta di regressione).

Residui

Una volta calcolati i valori dei coefficienti α e β e disegnata la retta di regressione che interpola la nuvola dei punti nel corrispondente scatterplot, è possibile osservare quanto questa retta si adatta ai punti che individuano le osservazioni. In generale, esisteranno degli scostamenti (*residui*) tra le ordinate dei punti y_i (*valori osservati*) e i corrispondenti *valori stimati*

$$\hat{y}_i = \alpha + \beta x_i \ (i = 1, 2, \ldots, n)$$

ottenuti mediante la reta di regressione. I punti $(x_1, \hat{y}_1)$, $(x_2, \hat{y}_2)$, ..., $(x_n, \hat{y}_n)$ sono posizionati sulla retta di regressione.
In generale, la *media campionaria dei valori stimati* è uguale alla media campionaria $\bar{y}$ delle osservazioni.
Se invece si considerano i *residui* come la differenza tra i valori osservati e i valori stimati

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

si ha che la *media campionaria dei residui* $\bar{E}$ *è nulla*, ossia in media gli scostamenti positivi e negativi si compensano. Di conseguenza, la *varianza campionaria dei residui* è:

$$s_E^2 = \frac{1}{n-1}\sum_{i=1}^{n}(E_i - \overline{E})^2 = \frac{1}{n-1}\sum_{i=1}^{n}E_i^2.$$

I residui mostrano di quanto si discostano i valori osservati dai valori stimati con la retta di regressione. In R, è possibile calcolare il vettore dei *valori stimati* attraverso la funzione **fitted(lm(y~x)),** con y che dipende da x. Le seguenti righe di codice determinano i valori stimati:

```
> stime <- fitted(lm(tipiDiPasto$mensa~tipiDiPasto$casa))
> stime
        1         2         3         4         5         6         7
 9.747845 11.523139  9.332351 12.127495  9.861162  9.294578 10.201112
        8         9        10        11        12        13        14
 9.332351  8.350273  5.517357  6.499434 10.616606  4.044240  3.779835
       15        16        17        18        19        20
 5.517357  2.608896  3.628746  3.402112  3.439885  4.875229
```

che possono essere estratti anche a partire dall'oggetto **linearmodel**. Ad esempio, volendo ottenere il primo valore, è sufficiente digitare il seguente comando:

```
> linearmodel$fitted.values[[1]]
[1] 9.747845
```

che corrisponde a

$$\hat{y}_1 = \alpha + \beta x_1 = 35.4707245 - 0.3777222 * 68.1 = 9.747845$$

dal momento che il primo valore nel vettore *casa* è 68.1.
Per calcolare invece il vettore dei *residui*, si può utilizzare la funzione **resid(lm(y˜x)).**
Le seguenti linee di codice individuano i residui:

```
> residui <- resid(lm(tipiDiPasto$mensa~tipiDiPasto$casa))
> residui
           1            2            3            4            5
-0.647844907 -0.023139094 -1.332350522  0.272505438  1.638838443
           6            7            8            9           10
 0.005421694  1.098888492  0.867649478  0.449727113 -1.117356630
          11           12           13           14           15
-0.699434266 -1.216605892  0.355759823  1.520165340 -1.617356630
          16           17           18           19           20
 0.591104060  0.771254207 -0.202112492 -1.139884709  0.424771054
```

che possono essere estratti anche a partire dall'oggetto **linearmodel**. Ad esempio, volendo ottenere il primo valore, è sufficiente digitare il seguente comando

```
> linearmodel$residuals[[1]]
[1] -0.6478449
```

E corrisponde a:

$$E_1 = y_1 - \hat{y}_1 = 9.1 - 9.747845 = -0.647845.$$

Poiché il primo valore nel vettore *mensa* è 9.1 e avendo calcolato prima $\hat{y}_1 = 9.747845$. La media dei residui è nulla, mentre per mediana, varianza campionaria e deviazione standard campionaria si ottiene:

```
> median(linearmodel$residuals)
[1] 0.1389636
> var(linearmodel$residuals)
[1] 0.939689
> sd(linearmodel$residuals)
[1] 0.9693756
```

Non si può invece calcolare il coefficiente di variazione, essendo la media campionaria dei residui nulla.

È possibile rappresentare graficamente i residui in tre modi diversi:

1. tracciando dei segmenti verticali che congiungono i valori stimati $\hat{y}_i$ (sulla retta di regressione) e i valori osservati y_i;
2. rappresentando i valori dei residui E_i rispetto alle osservazioni x_i (variabile indipendente);
3. rappresentando i residui standardizzati E_i/s_E rispetto ai valori stimati $\hat{y}_i$.

(1) Segmenti che congiungono i valori stimati e i valori osservati

Si vuole realizzare il grafico dei residui ottenuto aggiungendo, al grafico contenente lo *scatterplot* e la *retta di regressione*, dei segmenti verticali che visualizzano i residui. Questi segmenti sono ottenuti sovrapponendo al grafico ottenuto in Figura 1.34 dei segmenti che congiungono i seguenti due punti (x_i, y_i) e $(x_i, \hat{y}_i)$. Le seguenti linee di codice producono il grafico illustrato in Figura 1.35.

```
> plot(tipiDiPasto$casa, tipiDiPasto$mensa, main="Retta di regressione e residui",
+ xlab="casa", ylab="mensa", col="magenta")
> abline(lm(tipiDiPasto$mensa~tipiDiPasto$casa), col="blue")
>
> stime <- fitted(lm(tipiDiPasto$mensa~tipiDiPasto$casa))
> segments(tipiDiPasto$casa, stime, tipiDiPasto$casa ,
+ tipiDiPasto$mensa, col="red")
```

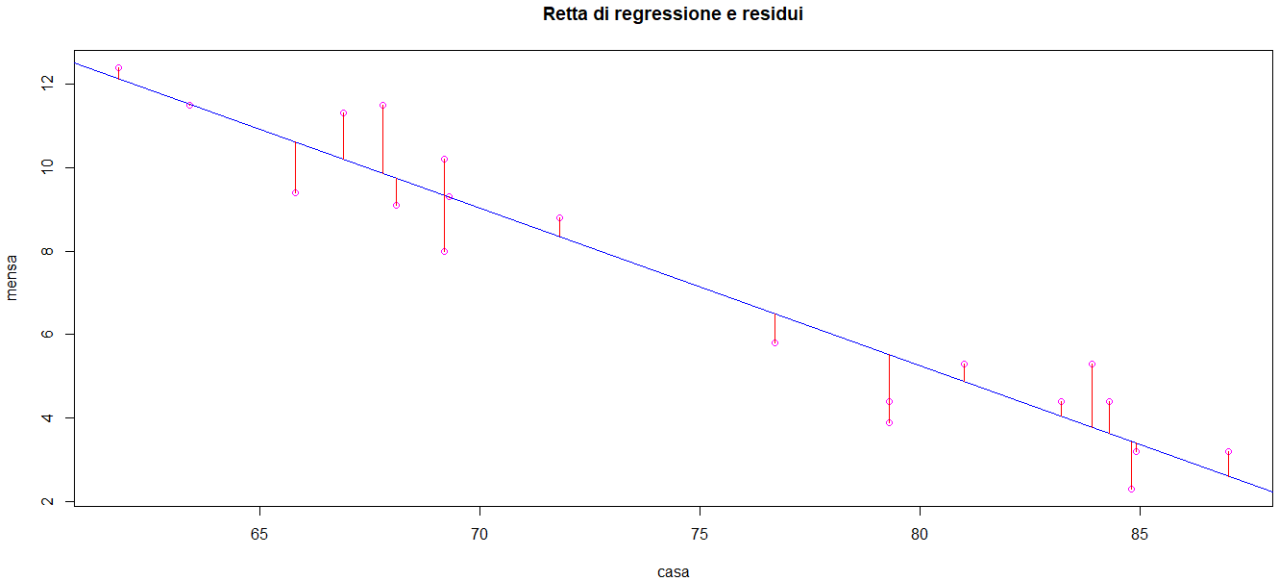


Figura 1.35: Grafico dei residui (tra valori osservati e valori stimati con la retta di regressione).

(2) Valori dei residui rispetto alle osservazioni della variabile indipendente

Un esame più accurato del modo con cui la retta di regressione interpola i dati e di come i residui si dispongano intorno alla retta interpolante influenzandone la posizione, può essere ottenuto attraverso il *diagramma dei residui* che è un grafico in cui *i valori dei residui sono posti sull'asse delle ordinate e quelli della variabile indipendente sull'asse delle ascisse.*
Si desidera realizzare il diagramma dei residui ponendo i valori dei residui sull'asse delle ordinate e quelli della variabile indipendente sull'asse delle ascisse. Le seguenti linee di codice producono il grafico in Figura 1.36.

```
> residui <- resid(lm(tipiDiPasto$mensa~tipiDiPasto$casa))
> plot(tipiDiPasto$casa, residui, main="Diagramma dei residui",
+ xlab="casa", ylab="residui", pch=9, col="magenta")
> abline(h=0, col="blue", lty=2)
```

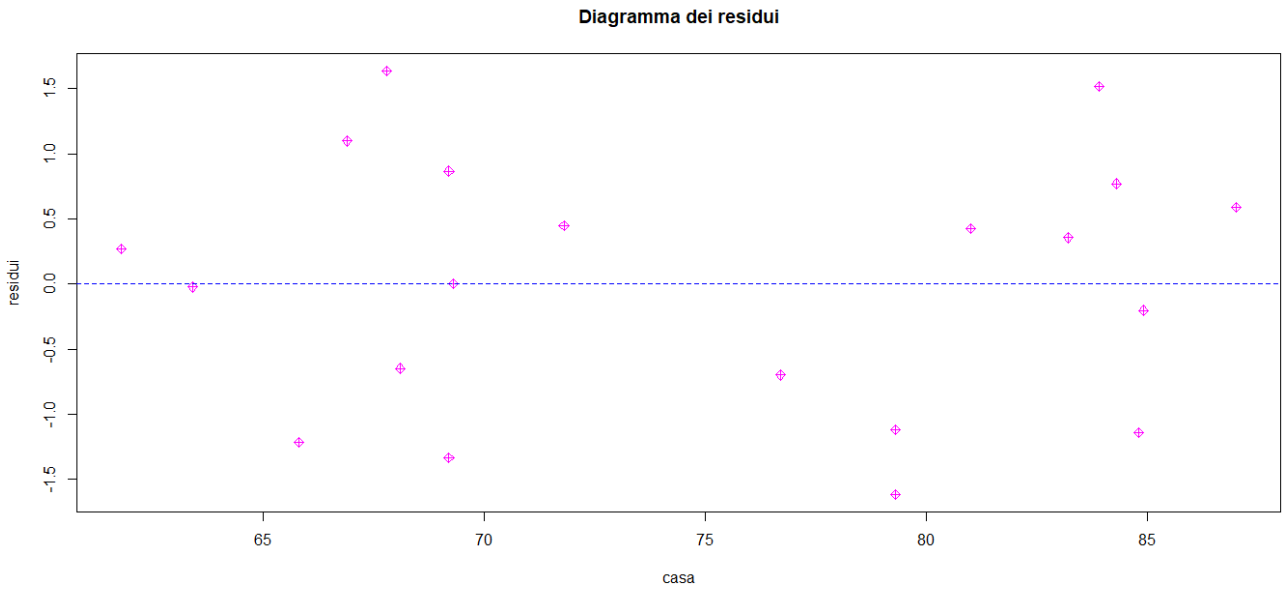


Figura 1.36: Residui in funzione dei valori in *casa*.

I punti indicano dove si collocano i residui rispetto ai valori del vettore *casa*.
La retta orizzontale è posizionata nello zero e corrisponde alla media campionaria dei residui, che è nulla.
Il diagramma dei residui aiuta a comprendere quale è l'adattamento della retta di regressione rispetto ai dati, consentendo di identificare quali sono le informazioni che hanno una forte influenza sulla collocazione e direzione della retta di regressione. Si nota che i punti sono disposti quasi casualmente e non si evidenzia nessun comportamento particolare nella distribuzione dei punti.
Occorre notare che la posizione della retta di regressione è fortemente influenzata dalla presenza di eventuali *valori anomali* che si discostano in modo significativo dagli altri. L'analisi dei residui aiuta ad individuare eventuali punti isolati (valori anomali) dovuti ad errori nella stima. Tali valori possono perturbare significativamente la stima dei parametri di regressione e influenzare l'interpretazione dei residui. Eliminando i valori anomali la varianza campionaria dei residui diminuisce.

(3) Valori dei residui standardizzati rispetto ai valori stimati

Spesso è interessante calcolare i *residui standardizzati* così definiti:

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

che risultano essere caratterizzati da media campionaria nulla e varianza unitaria. Nel caso preso in analisi, si ha:

```
> residui <- resid(lm(tipiDiPasto$mensa~tipiDiPasto$casa))
> residuistandard <- residui/sd(residui)
> residuistandard #visualizza vettori dati standardizzati
          1           2           3           4           5
-0.668311571 -0.023870103 -1.374442033  0.281114408  1.690612495
          6           7           8           9          10
 0.005592976  1.133604489  0.895060190  0.463934856 -1.152656071
         11          12          13          14          15
-0.721530737 -1.255040807  0.366998959  1.568190281 -1.668452030
         16          17          18          19          20
 0.609778171  0.795619608 -0.208497614 -1.175895855  0.438190387
```

È poi possibile realizzare un grafico in cui i residui standardizzati (ordinate) vengono disegnati in funzione dei valori stimati (ascisse) mediante la retta di regressione. Il seguente codice produce il grafico in Figura 1.37.

```
> stime <- fitted(lm(tipiDiPasto$mensa~tipiDiPasto$casa))
> residui <- resid(lm(tipiDiPasto$mensa~tipiDiPasto$casa))
> residuistandard <- residui/sd(residui)
> plot(stime, residuistandard, main="Residui standard rispetto ai valori stimati",
+ xlab="valori stimati", ylab="residui standard", pch=5, col="magenta")
> abline(h=0, col="blue", lty=2)
```

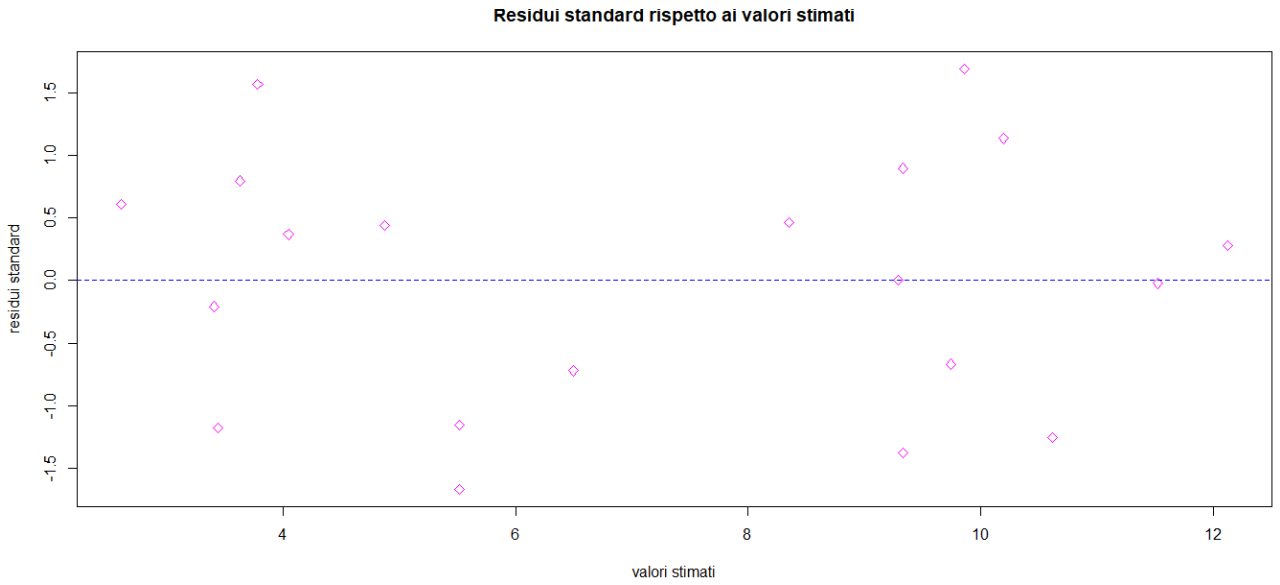


Figura 1.37: Residui in funzione dei valori stimati con la retta di regressione.

I punti indicano la posizione dove si collocano i residui standardizzati rispetto ai valori stimati con la retta di regressione. La retta orizzontale è posizionata nello zero, che corrisponde alla media campionaria dei residui standardizzati. Non si evidenzia nessuna tendenza particolare nella distribuzione dei punti.

1.4.2.1 Coefficiente di determinazione

Poiché si è interessati a vedere quanto la retta si adatta ai dati, l'accento può essere posto sul quadrato del coefficiente di correlazione e su quanto esso si avvicini ad uno. È chiaro che r^2_{xy} molto vicino ad 1 indicherà che tutti i punti tenderanno ad allinearsi lungo la retta di regressione, mentre r^2_{xy} prossimo a 0 esprime una completa incapacità della retta di rappresentare la distribuzione dei dati considerati.

Se si denota con $(y_1, y_2, ..., y_n)$ *il vettore dei dati della variabile dipendente, con* $\bar{y}$ *la sua media campionaria e con* $(\hat{y}_1, \hat{y}_2, ..., \hat{y}_n)$ *i valori stimati attraverso la retta di regressione, il coefficiente di determinazione (detto anche r-square) è così definito:*

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

Si nota che il coefficiente di determinazione è *il rapporto tra la varianza dei valori stimati tramite la retta di regressione e la varianza dei valori osservati*.

Nel caso di regressione lineare semplice, il *coefficiente di determinazione* coincide con il *quadrato del coefficiente di correlazione*, ossia

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

Per calcolare tale indice in R si può utilizzare il quadrato del coefficiente di correlazione oppure **summary(lm(y~x))$r.square**. Si ha quindi:

```
> (cor(tipiDiPasto$casa,tipiDiPasto$mensa))^2
[1] 0.9131126
> summary(lm(tipiDiPasto$mensa~tipiDiPasto$casa))$r.square
[1] 0.9131126
```

La retta si adatta quindi bene ai dati.

1.4.3 Regressione lineare multipla

In molte applicazioni dell'analisi di regressione sono coinvolte situazioni con più di una singola variabile indipendente. Il modello di *regressione lineare multipla* viene utilizzato per spiegare la relazione tra una variabile quantitativa Y, detta *variabile dipendente*, e le variabili quantitative *indipendenti* X_1, X_2, ..., X_p. Se si definisce un data frame *dfm*, contenente *n* osservazioni delle $p + 1$ *variabili* $Y, X_1, X_2, ..., X_p$, allora **cov(dfm)** e **cor(dfm)** forniscono due matrici di dimensioni $(p+1)\cdot(p+1)$ i cui elementi sono le *covarianze* e le *correlazioni* tra coppie di variabili. In particolare, tali matrici sono *simmetriche*; la **matrice delle covarianze** contiene sulla diagonale principale la *varianza* delle singole colonne del data frame, mentre la **matrice delle correlazioni** contiene il *numero* 1 sulla diagonale principale. La *matrice di correlazione* evidenzia tutte le correlazioni tra le coppie di variabili.
Ad esempio, considerando il dataframe *tipiDiPasto* precedentemente definito

```
> tipiDiPasto <- data.frame(colazione_adeguata=c(81.2,79,85.4,81.3,84.5,83.3,82.7,86.2,86.8,87.4,86,
+ 83.2,84.2,80.8,74.4,79.9,78.3,78.9,76.5,78.4),
+ colazione_con_latte=c(40.6,38.7,47.3,39.7,38.9,39.3,40.6,43.9,48.4,49.6,47.2,
+ 44.9,41.9,46.5,39.6,48.8,40.8,41.1,37.2,41.9),
+ casa=c(68.1,63.4,69.2,61.8,67.8,69.3,66.9,69.2,71.8,
+ 79.3,76.7,65.8,83.2,83.9,79.3,87,84.3,84.9,84.8,81),
+ mensa=c(9.1,11.5,8,12.4,11.5,9.3,11.3,10.2,8.8,
+ 4.4,5.8,9.4,4.4,5.3,3.9,3.2,4.4,3.2,2.3,5.3),
+ ristorante=c(2.7,5.2,3.4,5.2,6.8,4.1,3.9,3,2.3,2.2,2.2,
+ 2.1,1.9,1.2,1.3,0.7,0.7,1,1.1,0.7),
+ bar=c(3.6,3.4,3.9,2.4,1.3,2.2,1.4,2.1,2.8,1.2,
+ 1.5,4.2,0.8,0.4,0.7,0.7,0.6,0.6,1.1,0.7),
+ lavoro=c(9.7,8.3,9.4,11.1,6.5,8.3,7.2,9.5,8.2,7.3,
+ 7.4,11.1,5.3,5.4,8.6,3.3,5,5.5,4.9,7.6),
+ pranzo=c(61.7,69.1,62.2,59.1,75.1,67.3,63.7,62.8,61.9,69.9,69.1,
+ 56.5,72.8,81.4,71.5,82.8,81.5,75.1,73.2,75.5),
+ cena=c(29,21,28.9,29.3,13.2,22.5,25.3,26.7,26.1,18.7,
+ 17.2,30.8,13,11.8,18.8,9.1,9,12.1,13.6,14.1))
```

La matrice delle covarianze è

```
> cov(tipiDiPasto)
                 colazione_adeguata colazione_con_latte       casa      mensa  ristorante        bar      lavoro      pranzo       cena
colazione_adeguata          13.275368           8.2164211 -11.225474  4.701895   2.2803158  1.5820000   2.1162105 -12.4402105  10.682421
colazione_con_latte          8.216421          15.0605000   6.562816 -2.746132  -2.1971842  0.3198947  -0.4585263  -0.2494211   1.468474
casa                       -11.225474           6.5628158  69.216079 -26.144447 -11.8170789 -7.9876842 -14.5418947  51.6748947 -51.078263
mensa                        4.701895          -2.7461316 -26.144447  10.815026   4.8713421  2.5891579   4.7691579 -16.8077368  17.019105
ristorante                   2.280316          -2.1971842 -11.817079  4.871342   2.9655526  0.9502105   1.6517895  -5.8651053   5.622263
bar                          1.582000           0.3198947  -7.987684  2.589158   0.9502105  1.4943158   2.0037895  -7.6629474   7.644421
lavoro                       2.116211          -0.4585263 -14.541895  4.769158   1.6517895  2.0037895   4.5469474 -14.5292632  14.290737
pranzo                     -12.440211          -0.2494211  51.674895 -16.807737  -5.8651053 -7.6629474 -14.5292632  59.0767368 -54.734316
cena                        10.682421           1.4684737 -51.078263  17.019105   5.6222632  7.6444211  14.2907368 -54.7343158  54.074632
```

La matrice delle correlazioni è

```
> cor(tipiDiPasto)
                   colazione_adeguata colazione_con_latte       casa      mensa ristorante         bar      lavoro       pranzo        cena
colazione_adeguata          1.0000000         0.581085047 -0.3703206  0.3924062  0.3634284  0.35519103  0.27238025 -0.444218038  0.39870310
colazione_con_latte         0.5810850         1.000000000  0.2032671 -0.2151730 -0.3287713  0.06743203 -0.05540955 -0.008361904  0.05145762
casa                       -0.3703206         0.203267068  1.0000000 -0.9555693 -0.8248098 -0.78540959 -0.81970491  0.808105113 -0.83490265
mensa                       0.3924062        -0.215173041 -0.9555693  1.0000000  0.8601655  0.64405588  0.68009244 -0.664947259  0.70376255
ristorante                  0.3634284        -0.328771343 -0.8248098  0.8601655  1.0000000  0.45138365  0.44982326 -0.443113257  0.44397812
bar                         0.3551910         0.067432026 -0.7854096  0.6440559  0.4513837  1.00000000  0.76872488 -0.815579687  0.85040660
lavoro                      0.2723803        -0.055409548 -0.8197049  0.6800924  0.4498233  0.76872488  1.00000000 -0.886493135  0.91137660
pranzo                     -0.4442180        -0.008361904  0.8081051 -0.6649473 -0.4431133 -0.81557969 -0.88649314  1.000000000 -0.96839973
cena                        0.3987031         0.051457619 -0.8349027  0.7037625  0.4439781  0.85040660  0.91137660 -0.968399730  1.00000000
```

Si nota che esiste una forte correlazione negativa tra il vettore *mensa* e *casa*.
Il modello di regressione lineare multipla con *p* variabili indipendenti è esprimibile attraverso l'equazione:

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

Dove

- α e l'intercetta, ossia il valore di Y quando $X_1=X_2=...=X_p = 0$;
- $\beta_1, \beta_2, ..., \beta_p$, sono i *regressori*. In particolare, β_1 rappresenta l'inclinazione di Y rispetto alla variabile X_1 tenendo costanti le variabili $X_2, X_3, ..., X_p$ e β_p rappresenta l'inclinazione di Y rispetto alla variabile X_p tenendo costanti le variabili $X_1, X_2, ... X_{p-1}$.

Per determinare le stime di $\alpha, \beta_1, ..., \beta_p$ si ricorre al metodo dei *minimi quadrati*. Occorre quindi minimizzare la quantità:

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

dove *n* è il numero di osservazioni, $(x_{1,j}, x_{2,j}, \ldots, x_{n,j})$ sono i valori osservati della variabile X_j (j=1, 2, ..., p) e $(y_1, y_2, \ldots, y_n)$ i valori osservati della variabile Y.
Si calcola la derivata rispetto a $\alpha, \beta_1, \beta_2. ..., \beta_p$ e si pone uguale a 0. Da ciò deriva che:

$$\alpha = \overline{y} - \beta_1 \overline{x}_1 - \beta_2 \overline{x}_2 - \ldots - \beta_p \overline{x}_p.$$

Utilizziamo ora la *regressione lineare multivariata* in R con la funzione

$$\mathrm{lm}(y \tilde{} x_1 + x_2 + \ldots x_p)$$

L'argomento $y \tilde{} x_1 + x_2 + ... + x_p$ passato alla funzione **lm()** indica che y dipende da $x_1 + x_2 + ... + x_p$, ossia che $x_1 + x_2 + ... + x_p$ è la variabile *indipendente* e y la variabile *dipendente*.
Dapprima si specifica la variabile *dipendente* (y), quindi si specificano le variabili *indipendenti*, tra loro unite da un segno più, che indica l'inclusione della variabile nel modello (il segno più non corrisponde ad una addizione).
Nel caso in analisi si considera il vettore *mensa* come variabile *dipendente* e tutte le altre come variabili *indipendenti*.

```
> lm(tipiDiPasto$mensa~tipiDiPasto$colazione_con_latte+tipiDiPasto$casa+
+ tipiDiPasto$colazione_adeguata+tipiDiPasto$ristorante+tipiDiPasto$bar+tipiDiPasto$lavoro+
+ tipiDiPasto$pranzo+tipiDiPasto$cena
+ )

Call:
lm(formula = tipiDiPasto$mensa ~ tipiDiPasto$colazione_con_latte +
    tipiDiPasto$casa + tipiDiPasto$colazione_adeguata + tipiDiPasto$ristorante +
    tipiDiPasto$bar + tipiDiPasto$lavoro + tipiDiPasto$pranzo +
    tipiDiPasto$cena)

Coefficients:
                      (Intercept)  tipiDiPasto$colazione_con_latte
                         35.04560                         -0.09733
                tipiDiPasto$casa   tipiDiPasto$colazione_adeguata
                         -0.68197                          0.20315
          tipiDiPasto$ristorante                 tipiDiPasto$bar
                         -0.67680                         -0.62693
              tipiDiPasto$lavoro               tipiDiPasto$pranzo
                         -0.34208                          0.20652
                tipiDiPasto$cena
                          0.09150
```

Come nel caso della regressione lineare semplice, la funzione **lm()** restituisce una lista di attributi tra cui i coefficienti. Si osserva che i segni dei *regressori* β_1, β_2, β_4, β_5 e β_6 sono negativi, ciò implica che i valori di *colazione_con_latte, casa, ristorante, bar* e *lavoro* hanno un effetto negativo sulla percentuale di persone che hanno consumato un pasto in mensa. Per i restanti *regressori* invece, il segno è positivo e quindi i restanti vettori avranno un effetto positivo sulla percentuale di persone che ha consumato un pasto in mensa.

Residui

Una volta calcolati i valori dei coefficienti α e β_1, β_2 e β_p, è possibile osservare gli scostamenti (*residui*) tra le ordinate dei punti y_i (*valori osservati*) e i corrispondenti valori stimati:

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

ottenuti mediante la regressione lineare multipla.
Anche per la regressione lineare multipla, si ha che la *media campionaria dei valori stimati coincide con la media campionaria dei valori osservati*.
I residui, che mostrano di *quanto si discostano* i valori osservati dai valori stimati con la retta di regressione, possono essere indicati nel modo seguente:

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

La *media campionaria dei residui* $\overline{E}$ *nulla*, ossia in media gli scostamenti positivi e negativi si compensano. La varianza campionaria dei residui è:

$$s_E^2 = \frac{1}{n-1}\sum_{i=1}^{n}(E_i - \overline{E})^2 = \frac{1}{n-1}\sum_{i=1}^{n} E_i^2$$

In R per calcolare il vettore dei valori stimati tramite regressione lineare multipla ($\hat{y}_1$, $\hat{y}_2$, ..., $\hat{y}_n$) si utilizza la funzione **fitted(lm(y~ x_1+x_2+...+x_p))** con y che dipende da x_1, x_2, ..., x_p.
Le seguenti linee di codice determinano i valori stimati:

```
> stimemult <- fitted(lm(tipiDiPasto$mensa~tipiDiPasto$casa+tipiDiPasto$colazione_con_latte+
+ tipiDiPasto$colazione_adeguata+tipiDiPasto$ristorante+tipiDiPasto$bar+tipiDiPasto$lavoro+
+ tipiDiPasto$pranzo+tipiDiPasto$cena
+ ))
> stimemult #visualizza il vettore delle stime
        1         2         3         4         5         6         7
 9.140947 11.792715  8.126773 11.617168 11.264708  9.846558 11.760801
        8         9        10        11        12        13        14
 9.907818  8.057432  5.301418  6.498968  9.339060  3.956422  4.697202
       15        16        17        18        19        20
 4.451258  3.087151  4.585621  2.856978  2.386182  5.024819
```

Per calcolare invece il vettore dei *residui*, si può utilizzare la funzione **resid(lm(y~ x_1+x_2+...+x_p))** Le seguenti righe di codice determinano i residui

```
> residuimult <- resid(lm(tipiDiPasto$mensa~tipiDiPasto$casa+tipiDiPasto$colazione_con_latte+
+ tipiDiPasto$colazione_adeguata+tipiDiPasto$ristorante+tipiDiPasto$bar+tipiDiPasto$lavoro+
+ tipiDiPasto$pranzo+tipiDiPasto$cena
+ ))
> residuimult #visualizza i residui
          1           2           3           4           5           6
-0.04094707 -0.29271506 -0.12677298  0.78283153  0.23529206 -0.54655810
          7           8           9          10          11          12
-0.46080074  0.29218153  0.74256800 -0.90141782 -0.69896799  0.06094007
         13          14          15          16          17          18
 0.44357833  0.60279790 -0.55125836  0.11284859 -0.18562085  0.34302161
         19          20
-0.08618191  0.27518124
```

La media dei residui è nulla. Invece, la mediana, la varianza campionaria e la deviazione standard campionaria dei residui sono:

```
> median(residuimult)
[1] 0.009996499
> var(residuimult)
[1] 0.2282264
> sd(residuimult)
[1] 0.4777304
```

Anche nel caso multivariato è interessante calcolare i *residui standardizzati* così definiti:

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

che risultano essere caratterizzati da media campionaria nulla e varianza unitaria. Ad esempio, per il data frame *tipiDiPasto* risulta

```
> multiplelinearmodel <- (lm(tipiDiPasto$mensa~tipiDiPasto$casa+tipiDiPasto$colazione_con_latte+
+ tipiDiPasto$colazione_adeguata+tipiDiPasto$ristorante+tipiDiPasto$bar+tipiDiPasto$lavoro+
+ tipiDiPasto$pranzo+tipiDiPasto$cena
+ ))
> residuimult <- resid(lm(tipiDiPasto$mensa~tipiDiPasto$casa+tipiDiPasto$colazione_con_latte+
+ tipiDiPasto$colazione_adeguata+tipiDiPasto$ristorante+tipiDiPasto$bar+tipiDiPasto$lavoro+
+ tipiDiPasto$pranzo+tipiDiPasto$cena
+ ))
> residuimultstandard <- residuimult/sd(residuimult)
> residuimultstandard
          1           2           3           4           5           6
-0.08571168 -0.61272015 -0.26536510  1.63864703  0.49252058 -1.14407223
          7           8           9          10          11          12
-0.96456227  0.61160336  1.55436618 -1.88687549 -1.46310129  0.12756164
         13          14          15          16          17          18
 0.92851180  1.26179510 -1.15391094  0.23621815 -0.38854727  0.71802339
         19          20
-0.18039863  0.57601783
```

È possibile quindi realizzare e un grafico in cui i *residui standardizzati* (ordinate) vengono

disegnati in funzione dei *valori stimati* (ascisse). Il seguente codice produce il grafico in Figura 1.38.

```
> residuimultstandard <- residuimult/sd(residuimult)
> plot(stimemult, residuimultstandard, main="Residui standard rispetto ai valori stimati",
+ xlab="valori stimati", ylab="residui standard", pch=5, col="magenta")
> abline(h=0, col="blue", lty=2)
```

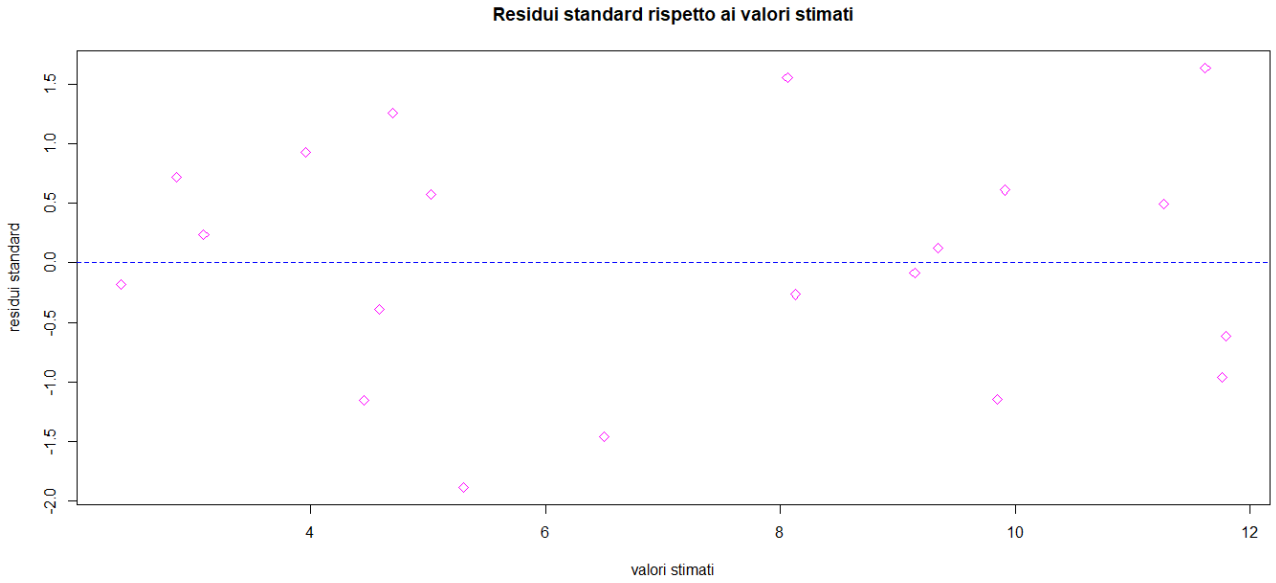


Figura 1.38: Residui in funzione dei valori stimati.

I punti indicano la posizione dove si collocano i residui standardizzati rispetto ai valori stimati con la retta di regressione. La retta orizzontale è posizionata nello zero, che corrisponde alla media campionaria dei residui standardizzati. Non si evidenzia nessuna tendenza particolare nella distribuzione dei punti.

1.4.3.1 Coefficiente di determinazione

Il coefficiente di determinazione di un modello di regressione lineare multipla è il rapporto tra la varianza dei valori stimati tramite la funzione di regressione e la varianza i valori osservati della variabile dipendente. Se si denota con $(y_1, y_2, ..., y_n)$ il vettore dei dati della variabile *dipendente*, con $\bar{y}$ la sua *media campionaria* e con $(\hat{y}_1, \hat{y}_2, ..., \hat{y}_n)$ i *valori stimati attraverso la funzione di regressione*, il *coefficiente di determinazione* è:

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

L'indice D^2 è adimensionale e risulta $0 \leq D^2 \leq 1$. Quando $D^2 = 0$ il modello di regressione multipla utilizzato non spiega per nulla i dati. Invece, quando $D^2 = 1$ il modello di regressione multipla utilizzato spiega perfettamente i dati. In R per calcolare l'indice D^2 basta utilizzare **summary(lm(y~ x_1+x_2+…+x_p))$r.square.**

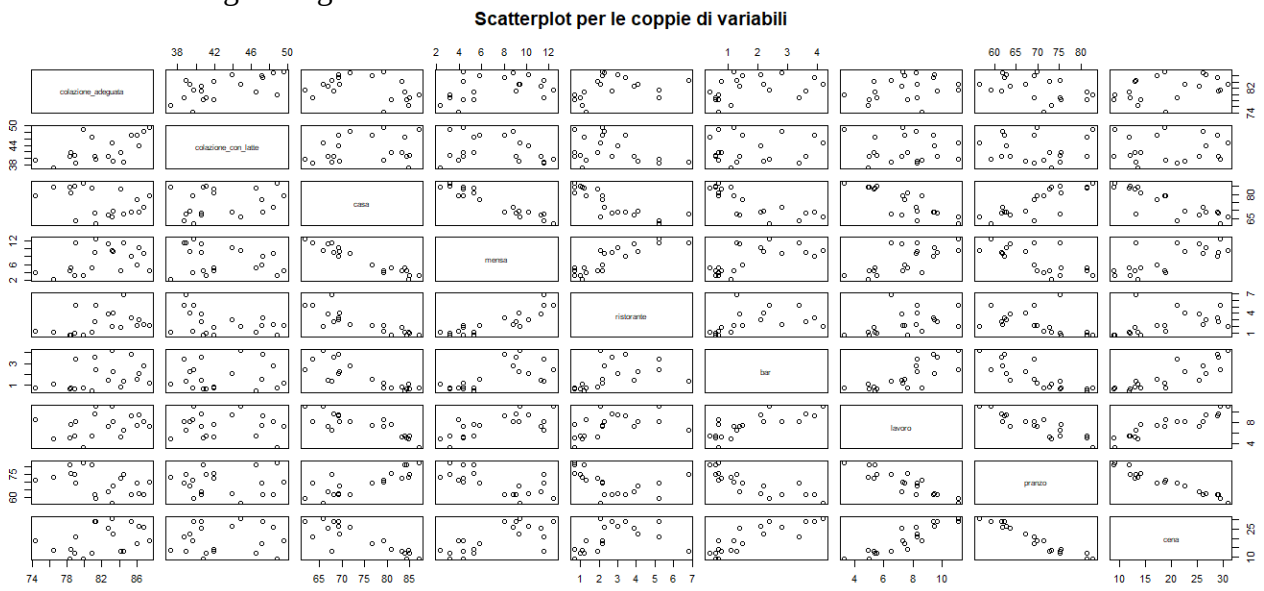
```
> summary(lm(tipiDiPasto$mensa~tipiDiPasto$casa+tipiDiPasto$colazione_con_latte+
+ tipiDiPasto$colazione_adeguata+tipiDiPasto$ristorante+tipiDiPasto$bar+tipiDiPasto$lavoro+
+ tipiDiPasto$pranzo+tipiDiPasto$cena))$r.square
[1] 0.9788973
```

Il coefficiente di determinazione è quindi 0.9788973, ossia il modello di regressione multipla utilizzato può spiegare i dati in modo significativo in quanto il valore è nettamente più vicino all'unità che allo zero.

1.4.4 Regressione non lineare

Spesso l'ipotesi di linearità di un modello non è accettabile, cioè capita che la retta non è la soluzione migliore per approssimare i dati. In questi casi, è utile quindi ricorrere alla *regressione non lineare*.

Considerando il seguente grafico



È possibile notare che quasi tutte le coppie di variabili considerate presentano relazioni lineari e quindi esprimibili attraverso una retta. Si desidera però osservare nel dettaglio la coppia *lavoro* e *colazione_con_latte*. Tramite le successive linee di codice si ottiene la Figura 1.39.

```
> plot(lavoro, colazione_con_latte, main="Scatterplot", xlab="lavoro", ylab="colazione con latte", col="red")
```

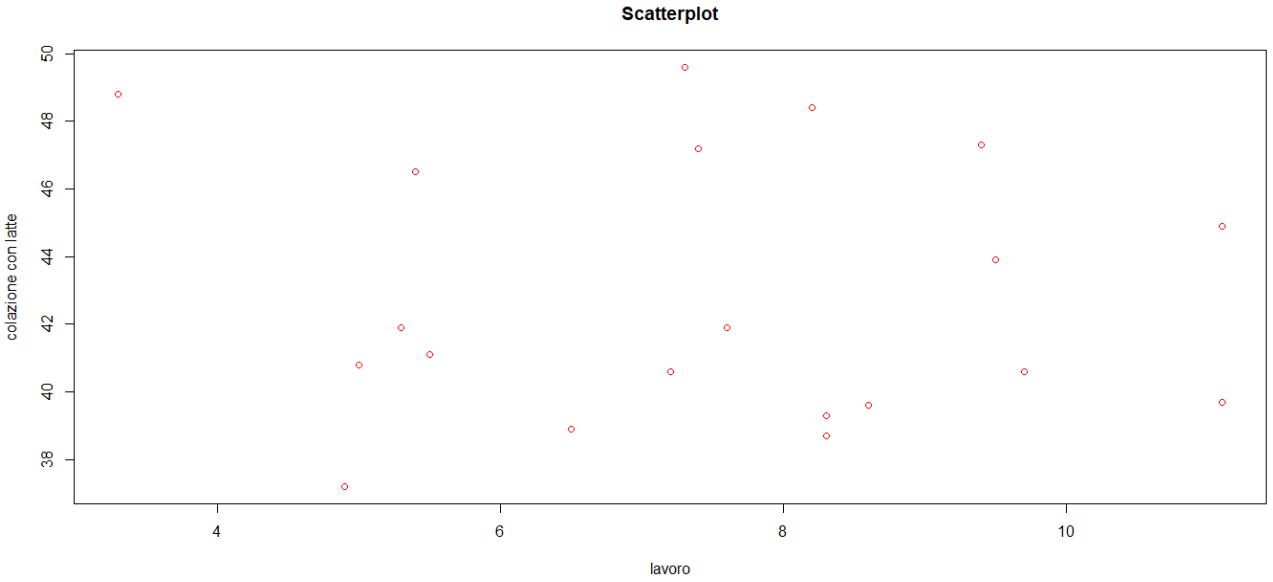


Figura 1.39: Scatterplot.

Osservando il grafico si nota che un modello lineare non approssima efficacemente i dati. Come ulteriore prova, si calcola il coefficiente di determinazione tramite la regressione lineare:

```
> summary(lm(colazione_con_latte~lavoro))$r.square
[1] 0.003070218
```

il valore risultante è 0.003070218, che risulta essere vicinissimo allo zero. A questo punto, viene dunque considerato il seguente modello non lineare

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

Per la stima dei parametri α, β e γ si può ricorrere alla regressione multipla

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

Con *regressori* $X_1 = X$ e $X_2 = X^2$.
Con R è facile stimare i parametri *α, β e γ* tramite la funzione **lm(y~ x + I(x∧2))**, dove *I()* è un *identificatore di variabile* e viene inserito quando si debbono effettuare operazioni matematiche (divisione, elevamento a potenza) nelle variabili della regressione.
Verrà ora verificato se l'approssimazione polinomiale è adeguata per stimare i dati considerati.

```
> pol2 <- lm(tipiDiPasto$colazione_con_latte~tipiDiPasto$lavoro + I((tipiDiPasto$lavoro)^2))
> pol2 #visualizza i coefficienti stimati

Call:
lm(formula = tipiDiPasto$colazione_con_latte ~ tipiDiPasto$lavoro +
    I((tipiDiPasto$lavoro)^2))

Coefficients:
              (Intercept)         tipiDiPasto$lavoro  I((tipiDiPasto$lavoro)^2)
                 48.20032                   -1.44565                    0.09056
```

Le seguenti linee di codice hanno mostrato che $\alpha = 48.20032$, $\beta = -1.44565$ *e* $\gamma = 0.09056$. Il modello polinomiale è quindi

$$Y = 48.20032 - 1.44565\ X_1 - 0.09056\ X_2$$

Per poter recuperare i parametri procediamo nel seguente modo:

```
> alpha <- pol2$coefficients[[1]]
> beta <- pol2$coefficients[[2]]
> gamma <- pol2$coefficients[[3]]
> c(alpha, beta, gamma)
[1] 48.20031943 -1.44564932  0.09056143
```

Viene poi calcolato il coefficiente di determinazione D^2

```
> summary(lm(tipiDiPasto$colazione_con_latte~tipiDiPasto$lavoro + I((tipiDiPasto$lavoro)^2)))$r.square
[1] 0.01658444
```

D^2 è uguale a 0.01658444, che non si avvicina all'unità ma è maggiore a quello ottenuto con il modello lineare semplice. Di conseguenza, il *modello di regressione polinomiale* è più adatto per i dati presi in considerazione.
Si può quindi disegnare la curva stimata sullo *scatterplot*. Le seguenti righe di codice producono il grafico in Figura 1.40.

```
> plot(tipiDiPasto$lavoro, tipiDiPasto$colazione_con_latte, main="Scatterplot e curva stimata", col="green")
> curve(alpha + beta * x + gamma * x^2, add=TRUE)
```

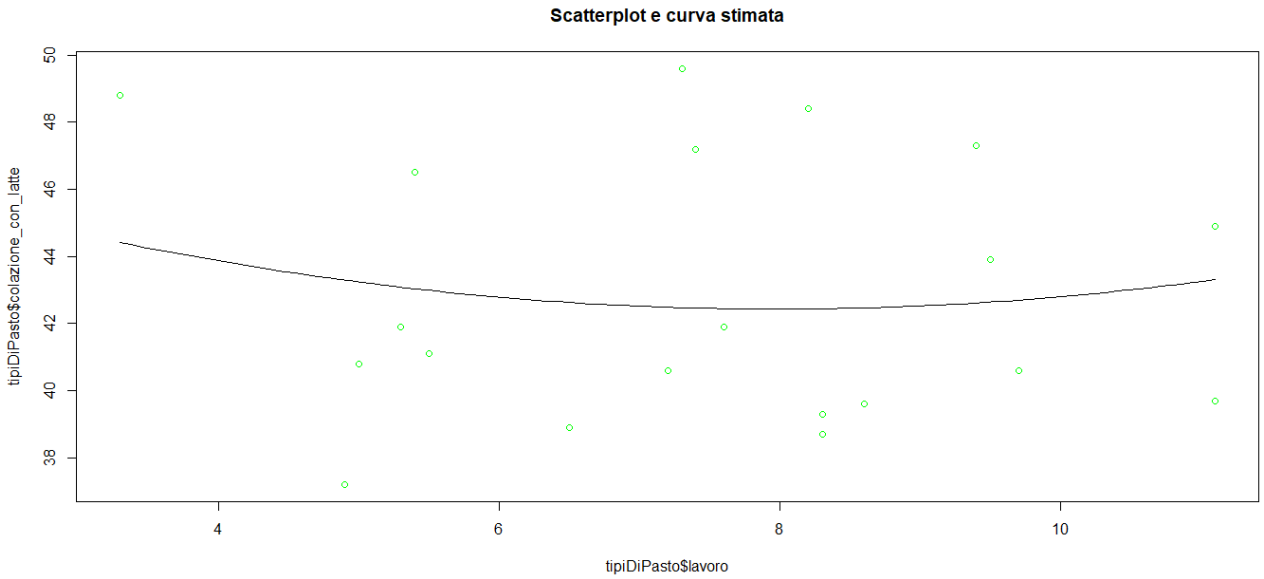


Figura 1.40: Rappresentazione dello scatterplot e della curva stimata.

1.5 Analisi dei cluster

L'analisi dei cluster è una metodologia che permette di raggruppare in sottoinsiemi, detti *cluster*, entità (unità) appartenenti ad un insieme più ampio. Esistono vari metodi per effettuare l'analisi dei cluster, tutti con il medesimo scopo: ottenere raggruppamenti in base alla somiglianza in modo che gli elementi di uno stesso gruppo siano tra di loro il più possibile simili e gli elementi appartenenti a gruppi distinti siano tra di loro il più possibile diversi. Sia $I = \{I_1, I_2, ..., I_n\}$ un insieme di n individui (entità o unità) appartenenti ad una popolazione ideale. Esiste inoltre un *insieme di caratteristiche* $C = \{C_1, C_2, ..., C_p\}$ che sono osservabili e possedute da ogni individuo in 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*. Nel nostro caso, si desidera che *ogni individuo (regione) con le sue caratteristiche appartenga ad uno solo degli insiemi.*
Uno dei problemi che si presenta nell'analisi dei cluster riguarda la standardizzazione o meno delle variabili, poiché attribuire un peso diverso a ciascuna caratteristica potrebbe condurre a risultati differenti circa la classificazione a seconda delle tecniche di clustering utilizzate. Per risolvere il problema dei clustering, è preferibile definire i termini *somiglianza* o *differenza* in modo quantitativo.
La *somiglianza* può essere definita mediante un *coefficiente di similarità* $s_{ij} = s(X_i, X_j)$ oppure mediante una *misura di distanza* $d_{ij} = d(X_i, X_j)$ tra due individui I_i e I_j (i = j). I *coefficienti di similarità* assumono valori compresi tra 0 e 1, mentre le *misure di distanza* possono assumere qualsiasi valore maggiore o uguale a zero. Quindi, si potrebbe assegnare due individui I_i *e* I_j ($i \neq j$) allo stesso cluster se il coefficiente di similarità tra i punti X_i e X_j è prossimo a 1 oppure
se la distanza tra i punti X_i e X_j è sufficientemente piccola e a differenti cluster se il coefficiente di similarità tra i punti è prossimo a 0 oppure se la distanza tra i punti è sufficientemente grande.

1.5.1 Distanza e similarità

Le *misure metriche di somiglianza* sono soprattutto basate sulle *funzioni distanza* tra i vettori delle caratteristiche. Viene definita dunque tale funzione:

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 (spazio Euclideo), cioè X_i è a distanza zero da se stesso e che ogni due punti a distanza nulla devono essere identici;
- d(X_i, X_j) ≥ 0 per ogni X_i e X_j in E_p, cioè la funzione distanza è non negativa;
- d(X_i, X_j) = d(X_j, X_i) per ogni X_i e X_j in E_p, che impone la simmetria richiedendo che la distanza X_i e X_j deve essere la stessa della distanza in X_j e X_i;
- 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, ovvero la *disuguaglianza triangolare*, richiede che la distanza X_i e X_j debba essere sempre minore o uguale della somma delle distanze di ognuno dei due vettori considerati da qualunque altro terzo vettore X_k.

In generale, le distanze tra tutte le possibili coppie di unità sono inserite in una matrice simmetrica *D* di cardinalità *n x n*, ossia

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

dove d_{ij} = d(X_i, X_j) *(i, j* = 1, 2, .., *n)*. I termini sulla diagonale principale sono tutti uguali a zero, mentre i termini simmetrici sono uguali a due a due.
In R la funzione **dist(X, method = "euclidean", diag = FALSE, upper = FALSE)** restituisce la matrice delle distanze *D* calcolata utilizzando le misure di distanza tra le righe della matrice *X* dei dati, dove:

- **X** rappresenta una matrice numerica o un data frame;
- **method** seleziona la misura di distanza da utilizzare (di default è euclidean);
- **diag** è posta uguale a TRUE se si desidera che la matrice delle distanze contenga anche i valori nulli sulla diagonale (di default è FALSE);
- **upper** è posta uguale a TRUE se si desidera che la matrice delle distanze contenga anche i valori al di sopra della diagonale principale (di default è FALSE).

Le opzioni disponibili per il calcolo della matrice delle distanze sono:

1. metrica euclidea (euclidean);
2. metrica del valore assoluto o metrica di Manhattan (manhattan);
3. metrica del massimo o metrica di Chebycev (maximum);
4. metrica di Minkowski (minkowski);
5. distanza di Canberra (canberra);
6. distanza di Jaccard (binary)

La più familiare misura di distanza utilizzata è la *metrica Euclidea*, che risulta però essere fortemente *influenzata dall'unità di misura* in base alla quale è valutata ciascuna delle *p* caratteristiche. Il metodo più utilizzato per ovviare a questo inconveniente è quello di scalare e standardizzare inizialmente le misure in maniera tale da realizzare la possibilità di un confronto tra le misure. Nel nostro caso ci si riferisce sempre di una percentuale quindi si può anche evitare di effettuare. Tuttavia, verrà mostrato ugualmente il funzionamento della funzione **scale()** sul data frame.

Come prima cosa, viene definito il dataframe:

```
> tipiDiPasto <- data.frame(colazione_adeguata=c(81.2,79,85.4,81.3,84.5,83.3,82.7,86.2,86.8,87.4,86,
+ 83.2,84.2,80.8,74.4,79.9,78.3,78.9,76.5,78.4),
+ colazione_con_latte=c(40.6,38.7,47.3,39.7,38.9,39.3,40.6,43.9,48.4,49.6,47.2,
+ 44.9,41.9,46.5,39.6,48.8,40.8,41.1,37.2,41.9),
+ casa=c(68.1,63.4,69.2,61.8,67.8,69.3,66.9,69.2,71.8,
+ 79.3,76.7,65.8,83.2,83.9,79.3,87,84.3,84.9,84.8,81),
+ mensa=c(9.1,11.5,8,12.4,11.5,9.3,11.3,10.2,8.8,
+ 4.4,5.8,9.4,4.4,5.3,3.9,3.2,4.4,3.2,2.3,5.3),
+ ristorante=c(2.7,5.2,3.4,5.2,6.8,4.1,3.9,3,2.3,2.2,2.2,
+ 2.1,1.9,1.2,1.3,0.7,0.7,1,1.1,0.7),
+ bar=c(3.6,3.4,3.9,2.4,1.3,2.2,1.4,2.1,2.8,1.2,
+ 1.5,4.2,0.8,0.4,0.7,0.7,0.6,0.6,1.1,0.7),
+ lavoro=c(9.7,8.3,9.4,11.1,6.5,8.3,7.2,9.5,8.2,7.3,
+ 7.4,11.1,5.3,5.4,8.6,3.3,5,5.5,4.9,7.6),
+ pranzo=c(61.7,69.1,62.2,59.1,75.1,67.3,63.7,62.8,61.9,69.9,69.1,
+ 56.5,72.8,81.4,71.5,82.8,81.5,75.1,73.2,75.5),
+ cena=c(29,21,28.9,29.3,13.2,22.5,25.3,26.7,26.1,18.7,
+ 17.2,30.8,13,11.8,18.8,9.1,9,12.1,13.6,14.1))
```

```
> rownames(tipiDiPasto) <- c("Piemonte", "Valle_d_Aosta", "Liguria", "Lombardia",
+  "Trentino_Alto_Adige", "Veneto", "Friuli_Venezia_Giulia", "Emilia_Romagna",
+  "Toscana","Umbria","Marche","Lazio",
+  "Abruzzo","Molise","Campania","Puglia",
+  "Basilicata","Calabria","Sicilia","Sardegna")
```

a cui successivamente viene applicata la funzione **dist()**

```
> distEuclidean <- dist(tipiDiPasto, method="euclidean", diag=TRUE, upper=TRUE)
```

Ottenendo

```
                     Piemonte Valle_d_Aosta   Liguria Lombardia
Piemonte             0.000000     12.779280  8.116650  8.240146
Valle_d_Aosta       12.779280      0.000000 16.599096 13.689047
Liguria              8.116650     16.599096  0.000000 12.916656
Lombardia            8.240146     13.689047 12.916656  0.000000
Trentino_Alto_Adige 21.935588     12.517987 22.917461 24.245206
Veneto               9.331131      8.163333 11.902521 13.877680
Friuli_Venezia_Giulia 6.224950     9.172786  9.742176  9.222256
Emilia_Romagna       6.869498     13.941306  5.064583 11.366178
Toscana             10.838819     17.872045  4.728636 15.936122
Umbria              21.261938     22.657449 17.341569 27.556306
Marche              18.852056     18.933832 16.036209 24.775391
Lazio                7.799359     18.415754  8.055433  8.803976
Abruzzo             25.842600     24.167540 25.167439 32.181983
Molise              31.842739     28.144449 30.648328 37.964984
Campania            20.282505     19.124069 22.425432 26.729759
Puglia              36.817659     33.324315 35.263579 43.106380
Basilicata          33.468044     28.789929 33.758999 39.433615
Calabria            28.588284     26.180909 28.822387 35.072069
Sicilia             27.568823     25.618938 29.078171 34.096041
Sardegna            24.892770     21.803669 25.325679 31.107716
```

	Trentino_Alto_Adige	Veneto	Friuli_Venezia_Giulia
Piemonte	21.935588	9.331131	6.224950
Valle_d_Aosta	12.517987	8.163333	9.172786
Liguria	22.917461	11.902521	9.742176
Lombardia	24.245206	13.877680	9.222256
Trentino_Alto_Adige	0.000000	12.936769	17.095613
Veneto	12.936769	0.000000	5.873670
Friuli_Venezia_Giulia	17.095613	5.873670	0.000000
Emilia_Romagna	19.727139	8.422589	6.244998
Toscana	22.019764	12.136309	10.830974
Umbria	19.612751	16.544485	19.733221
Marche	16.026540	13.151046	16.686821
Lazio	27.449590	15.652476	11.486514
Abruzzo	18.102210	19.002368	23.649736
Molise	21.074392	24.912045	29.504068
Campania	19.283413	15.810440	19.766386
Puglia	26.088503	30.181617	34.610548
Basilicata	21.523476	26.175561	31.113020
Calabria	20.828826	22.185130	26.841013
Sicilia	21.894977	21.810319	26.301521
Sardegna	17.281204	18.323755	23.033888

	Emilia_Romagna	Toscana	Umbria	Marche	Lazio
Piemonte	6.869498	10.838819	21.261938	18.852056	7.799359
Valle_d_Aosta	13.941306	17.872045	22.657449	18.933832	18.415754
Liguria	5.064583	4.728636	17.341569	16.036209	8.055433
Lombardia	11.366178	15.936122	27.556306	24.775391	8.803976
Trentino_Alto_Adige	19.727139	22.019764	19.612751	16.026540	27.449590
Veneto	8.422589	12.136309	16.544485	13.151046	15.652476
Friuli_Venezia_Giulia	6.244998	10.830974	19.733221	16.686821	11.486514
Emilia_Romagna	0.000000	5.759340	17.037605	14.895973	9.299462
Toscana	5.759340	0.000000	14.126217	12.979985	11.100901
Umbria	17.037605	14.126217	0.000000	4.407947	24.421302
Marche	14.895973	12.979985	4.407947	0.000000	22.574543
Lazio	9.299462	11.100901	24.421302	22.574543	0.000000
Abruzzo	23.363861	22.390176	11.396929	10.576389	31.072979
Molise	29.443675	28.245708	16.132266	16.318088	37.168401
Campania	21.052078	21.505348	16.570154	14.625321	26.573859
Puglia	34.423684	32.664354	19.866806	20.630560	41.852240
Basilicata	32.006874	31.620089	20.537770	19.743100	39.502405
Calabria	27.263896	26.509055	15.904088	15.148927	34.312680
Sicilia	27.391422	27.238576	18.737929	17.478558	33.754555
Sardegna	23.643181	23.366429	14.110280	12.573385	30.761014

	Abruzzo	Molise	Campania	Puglia	Basilicata
Piemonte	25.842600	31.842739	20.282505	36.817659	33.468044
Valle_d_Aosta	24.167540	28.144449	19.124069	33.324315	28.789929
Liguria	25.167439	30.648328	22.425432	35.263579	33.758999
Lombardia	32.181983	37.964984	26.729759	43.106380	39.433615
Trentino_Alto_Adige	18.102210	21.074392	19.283413	26.088503	21.523476
Veneto	19.002368	24.912045	15.810440	30.181617	26.175561
Friuli_Venezia_Giulia	23.649736	29.504068	19.766386	34.610548	31.113020
Emilia_Romagna	23.363861	29.443675	21.052078	34.423684	32.006874
Toscana	22.390176	28.245708	21.505348	32.664354	31.620089
Umbria	11.396929	16.132266	16.570154	19.866806	20.537770
Marche	10.576389	16.318088	14.625321	20.630560	19.743100
Lazio	31.072979	37.168401	26.573859	41.852240	39.502405
Abruzzo	0.000000	10.491902	12.782019	14.235168	11.423222
Molise	10.491902	0.000000	16.402439	5.841233	6.928925
Campania	12.782019	16.402439	0.000000	20.613103	15.851498
Puglia	14.235168	5.841233	20.613103	0.000000	8.938121
Basilicata	11.423222	6.928925	15.851498	8.938121	0.000000
Calabria	6.328507	8.834591	11.041286	11.744360	7.291090
Sicilia	9.474175	13.638915	9.316652	16.341971	10.551303
Sardegna	7.384443	8.978307	8.098765	13.676622	9.038805

	Calabria	Sicilia	Sardegna
Piemonte	28.588284	27.568823	24.892770
Valle_d_Aosta	26.180909	25.618938	21.803669
Liguria	28.822387	29.078171	25.325679
Lombardia	35.072069	34.096041	31.107716
Trentino_Alto_Adige	20.828826	21.894977	17.281204
Veneto	22.185130	21.810319	18.323755
Friuli_Venezia_Giulia	26.841013	26.301521	23.033888
Emilia_Romagna	27.263896	27.391422	23.643181
Toscana	26.509055	27.238576	23.366429
Umbria	15.904088	18.737929	14.110280
Marche	15.148927	17.478558	12.573385
Lazio	34.312680	33.754555	30.761014
Abruzzo	6.328507	9.474175	7.384443
Molise	8.834591	13.638915	8.978307
Campania	11.041286	9.316652	8.098765
Puglia	11.744360	16.341971	13.676622
Basilicata	7.291090	10.551303	9.038805
Calabria	0.000000	5.316954	5.401852
Sicilia	5.316954	0.000000	7.892401
Sardegna	5.401852	7.892401	0.000000

In R, per scalare e standardizzare le variabili, è sufficiente applicare la funzione **scale(X, center=TRUE, scale=TRUE)**, in cui:

- **X** rappresenta una matrice numerica o un data frame;
- **center** è posta uguale a TRUE se dagli elementi di ogni colonna della matrice X si sottrae il valore medio della corrispondente colonna (di default è TRUE);
- **scale** è posta uguale a TRUE se si dividono gli elementi centrati di ogni colonna della matrice X per la deviazione standard della corrispondente colonna (di default è TRUE).

Riferendoci ai nostri dati:

```
> scale(tipiDiPasto)
                    colazione_adeguata colazione_con_latte       casa
Piemonte                    -0.1976101          -0.5784911 -0.8155420
Valle_d_Aosta               -0.8014189          -1.0680826 -1.3804717
Liguria                      0.9551156           1.1479633 -0.6833245
Lombardia                   -0.1701643          -0.8104028 -1.5727882
Trentino_Alto_Adige          0.7081030          -1.0165466 -0.8516014
Veneto                       0.3787528          -0.9134747 -0.6713047
Friuli_Venezia_Giulia        0.2140776          -0.5784911 -0.9597794
Emilia_Romagna               1.1746824           0.2718521 -0.6833245
Toscana                      1.3393576           1.4314111 -0.3708102
Umbria                       1.5040327           1.7406268  0.5306733
Marche                       1.1197907           1.1221953  0.2181590
Lazio                        0.3513069           0.5295319 -1.0919970
Abruzzo                      0.6257654          -0.2435074  0.9994447
Molise                      -0.3073935           0.9418195  1.0835831
Campania                    -2.0639280          -0.8361708  0.5306733
Puglia                      -0.5544062           1.5344830  1.4561963
Basilicata                  -0.9935398          -0.5269551  1.1316622
Calabria                    -0.8288647          -0.4496512  1.2037809
Sicilia                     -1.4875651          -1.4546022  1.1917611
Sardegna                    -0.9660940          -0.2435074  0.7350095

                          mensa  ristorante        bar      lavoro
Piemonte              0.5823110  0.06677979  1.4888474  1.04110135
Valle_d_Aosta         1.3121002  1.51851426  1.3252378  0.38455095
Liguria               0.2478243  0.47326544  1.7342618  0.90041198
Lombardia             1.5857712  1.51851426  0.5071898  1.69765176
Trentino_Alto_Adige   1.3121002  2.44762433 -0.3926631 -0.45958528
Veneto                0.6431268  0.87975109  0.3435802  0.38455095
Friuli_Venezia_Giulia 1.2512845  0.76361233 -0.3108583 -0.13131008
Emilia_Romagna        0.9167977  0.24098792  0.2617754  0.94730844
Toscana               0.4910873 -0.16549773  0.8344090  0.33765449
Umbria               -0.8468596 -0.22356711 -0.4744679 -0.08441362
Marche               -0.4211492 -0.22356711 -0.2290535 -0.03751717
Lazio                 0.6735346 -0.28163649  1.9796763  1.69765176
Abruzzo              -0.8468596 -0.39777525 -0.8016871 -1.02234277
Molise               -0.5731886 -0.80426090 -1.1289063 -0.97544631
Campania             -0.9988990 -0.74619152 -0.8834919  0.52524032
Puglia               -1.2117542 -1.09460779 -0.8834919 -1.96027192
Basilicata           -0.8468596 -1.09460779 -0.9652967 -1.16303214
Calabria             -1.2117542 -0.92039966 -0.9652967 -0.92854986
Sicilia              -1.4854252 -0.86233028 -0.5562727 -1.20992860
Sardegna             -0.5731886 -1.09460779 -0.8834919  0.05627575
```

```
                          pranzo        cena
Piemonte               -1.02912525  1.29053393
Valle_d_Aosta          -0.06635321  0.20262335
Liguria                -0.96407309  1.27693505
Lombardia              -1.36739651  1.33133058
Trentino_Alto_Adige     0.71427277 -0.85808947
Veneto                 -0.30054100  0.40660658
Friuli_Venezia_Giulia -0.76891659  0.78737529
Emilia_Romagna         -0.88601049  0.97775964
Toscana                -1.00310438  0.89616634
Umbria                  0.03773026 -0.11015095
Marche                 -0.06635321 -0.31413418
Lazio                  -1.70566777  1.53531381
Abruzzo                 0.41503281 -0.88528724
Molise                  1.53393005 -1.04847383
Campania                0.24589718 -0.09655206
Puglia                  1.71607611 -1.41564365
Basilicata              1.54694048 -1.42924253
Calabria                0.71427277 -1.00767718
Sicilia                 0.46707454 -0.80369394
Sardegna                0.76631450 -0.73569953

attr(,"scaled:center")
 colazione_adeguata colazione_con_latte              casa
             81.920              42.845             74.885
              mensa           ristorante               bar
              7.185               2.585             1.780
             lavoro              pranzo              cena
              7.480              69.610             19.510
attr(,"scaled:scale")
 colazione_adeguata colazione_con_latte              casa
           3.643538            3.880786          8.319620
              mensa           ristorante               bar
           3.288621            1.722078          1.222422
              lavoro              pranzo              cena
           2.132357            7.686139          7.353546
```

dove i valori indicati negli attributi corrispondono alle medie campionarie e alle deviazioni standard campionarie delle colonne del data frame originario dei dati. Infatti, è possibile ottenere gli stessi valori utilizzando le funzioni **apply(X, 2, mean)** e **apply(X, 2, sd)**, calcolando la media campionaria, la varianza campionaria e la deviazione standard campionaria delle colonne del data frame originario.

```
> apply(tipiDiPasto, 2, mean)
 colazione_adeguata colazione_con_latte              casa
             81.920              42.845             74.885
              mensa           ristorante               bar
              7.185               2.585             1.780
             lavoro              pranzo              cena
              7.480              69.610             19.510
> apply(tipiDiPasto, 2, sd)
 colazione_adeguata colazione_con_latte              casa
           3.643538            3.880786          8.319620
              mensa           ristorante               bar
           3.288621            1.722078          1.222422
              lavoro              pranzo              cena
           2.132357            7.686139          7.353546
```

Notiamo sono gli stessi di quella scalata. Viene ora calcolata la funzione di distanza per il data frame *tipiDiPasto* dopo aver effettuato lo scalamento:

```
> tipiDiPastoScaled <- scale(tipiDiPasto)
> dist(tipiDiPastoScaled, method="euclidean", diag=TRUE, upper=TRUE)
                     Piemonte Valle_d_Aosta   Liguria Lombardia
Piemonte             0.0000000     2.4762596 2.1652837 2.2919916
Valle_d_Aosta        2.4762596     0.0000000 3.6195270 2.4365740
Liguria              2.1652837     3.6195270 0.0000000 3.3289895
Lombardia            2.2919916     2.4365740 3.3289895 0.0000000
Trentino_Alto_Adige  4.5456225     2.9702693 4.8532246 4.1063870
Veneto               2.0456519     1.9583859 2.8811829 2.4830425
Friuli_Venezia_Giulia 2.4612514    2.4083072 3.1972120 2.4346114
Emilia_Romagna       2.0973303     3.2703327 1.8929538 2.6135458
Toscana              2.7823313     4.1184201 1.4372837 3.8809931
Umbria               4.5204547     5.3028214 3.5477098 5.5569269
Marche               3.7952789     4.4658690 3.1377182 4.8602594
Lazio                1.7101948     3.7720836 1.7256024 2.9488114
Abruzzo              4.7609682     4.9631135 4.8516574 5.8620041
Molise               5.5662620     5.5949530 5.5162166 6.6411779
Campania             4.2387239     4.5667367 5.2830371 5.2698270
Puglia               6.6193664     6.6522041 6.4080828 7.8640642
Basilicata           5.7213146     5.5711153 6.2231302 6.8763576
Calabria             5.1874368     5.3314821 5.6601954 6.3966608
Sicilia              5.2415318     5.3019630 6.0927201 6.5117224
Sardegna             4.4405060     4.7037284 5.0379893 5.5878230

                     Trentino_Alto_Adige    Veneto Friuli_Venezia_Giulia
Piemonte                       4.5456225 2.0456519              2.4612514
Valle_d_Aosta                  2.9702693 1.9583859              2.4083072
Liguria                        4.8532246 2.8811829              3.1972120
Lombardia                      4.1063870 2.4830425              2.4346114
Trentino_Alto_Adige            0.0000000 2.6346304              2.8825532
Veneto                         2.6346304 0.0000000              1.2902156
Friuli_Venezia_Giulia          2.8825532 1.2902156              0.0000000
Emilia_Romagna                 3.9077335 1.8745509              1.9101242
Toscana                        4.7215149 2.9331869              2.9482597
Umbria                         4.8069878 3.8030550              4.0156783
Marche                         4.1397684 2.9857314              3.2548663
Lazio                          5.6831540 3.3554664              3.5663613
Abruzzo                        4.1668247 3.5532282              3.8866023
Molise                         4.9251084 4.5616024              4.7540245
Campania                       5.2133157 3.8515733              4.1701479
Puglia                         6.0126792 5.7549509              5.9256391
Basilicata                     5.1107178 4.7215599              5.0457550
Calabria                       5.0191510 4.2822870              4.6014913
Sicilia                        5.3501382 4.4965229              4.8691248
Sardegna                       4.7480833 3.6971907              4.0247735
```

	Emilia_Romagna	Toscana	Umbria	Marche	Lazio
Piemonte	2.0973303	2.7823313	4.5204547	3.7952789	1.7101948
Valle_d_Aosta	3.2703327	4.1184201	5.3028214	4.4658690	3.7720836
Liguria	1.8929538	1.4372837	3.5477098	3.1377182	1.7256024
Lombardia	2.6135458	3.8809931	5.5569269	4.8602594	2.9488114
Trentino_Alto_Adige	3.9077335	4.7215149	4.8069878	4.1397684	5.6831540
Veneto	1.8745509	2.9331869	3.8030550	2.9857314	3.3554664
Friuli_Venezia_Giulia	1.9101242	2.9482597	4.0156783	3.2548663	3.5663613
Emilia_Romagna	0.0000000	1.5924119	3.2724930	2.6638890	2.3959499
Toscana	1.5924119	0.0000000	2.5915613	2.2226769	2.5337423
Umbria	3.2724930	2.5915613	0.0000000	0.9611783	4.7633106
Marche	2.6638890	2.2226769	0.9611783	0.0000000	4.2257708
Lazio	2.3959499	2.5337423	4.7633106	4.2257708	0.0000000
Abruzzo	4.1341575	4.0923479	2.5861581	2.1879052	5.7326403
Molise	4.9671610	4.7333350	2.9964080	2.8228139	6.4143716
Campania	4.6647250	5.0622213	4.4998865	3.9453374	5.4210959
Puglia	6.1433685	5.6274230	3.7729855	3.7954017	7.3846724
Basilicata	5.5856883	5.6204658	4.2334081	3.8229980	6.8604950
Calabria	5.0500401	5.0601072	3.6780793	3.2827449	6.2830923
Sicilia	5.5587255	5.7038874	4.7276613	4.2346934	6.5275516
Sardegna	4.3169158	4.4903687	3.4674075	2.9279961	5.4814573

	Abruzzo	Molise	Campania	Puglia	Basilicata
Piemonte	4.7609682	5.5662620	4.2387239	6.6193664	5.7213146
Valle_d_Aosta	4.9631135	5.5949530	4.5667367	6.6522041	5.5711153
Liguria	4.8516574	5.5162166	5.2830371	6.4080828	6.2231302
Lombardia	5.8620041	6.6411779	5.2698270	7.8640642	6.8763576
Trentino_Alto_Adige	4.1668247	4.9251084	5.2133157	6.0126792	5.1107178
Veneto	3.5532282	4.5616024	3.8515733	5.7549509	4.7215599
Friuli_Venezia_Giulia	3.8866023	4.7540245	4.1701479	5.9256391	5.0457550
Emilia_Romagna	4.1341575	4.9671610	4.6647250	6.1433685	5.5856883
Toscana	4.0923479	4.7333350	5.0622213	5.6274230	5.6204658
Umbria	2.5861581	2.9964080	4.4998865	3.7729855	4.2334081
Marche	2.1879052	2.8228139	3.9453374	3.7954017	3.8229980
Lazio	5.7326403	6.4143716	5.4210959	7.3846724	6.8604950
Abruzzo	0.0000000	1.9775803	3.3169825	2.8708527	2.1975471
Molise	1.9775803	0.0000000	3.4079974	1.4970884	1.7311022
Campania	3.3169825	3.4079974	0.0000000	4.3590308	2.8412632
Puglia	2.8708527	1.4970884	4.3590308	0.0000000	2.3134197
Basilicata	2.1975471	1.7311022	2.8412632	2.3134197	0.0000000
Calabria	1.6572781	1.8292880	2.3180231	2.5193981	1.0619916
Sicilia	2.5881001	3.0925526	2.2513857	3.5508823	1.8157209
Sardegna	2.1166717	1.9637798	1.6733612	3.1127879	1.7020892

```
                          Calabria   Sicilia  Sardegna
Piemonte                 5.1874368 5.2415318 4.4405060
Valle_d_Aosta            5.3314821 5.3019630 4.7037284
Liguria                  5.6601954 6.0927201 5.0379893
Lombardia                6.3966608 6.5117224 5.5878230
Trentino_Alto_Adige      5.0191510 5.3501382 4.7480833
Veneto                   4.2822870 4.4965229 3.6971907
Friuli_Venezia_Giulia    4.6014913 4.8691248 4.0247735
Emilia_Romagna           5.0500401 5.5587255 4.3169158
Toscana                  5.0601072 5.7038874 4.4903687
Umbria                   3.6780793 4.7276613 3.4674075
Marche                   3.2827449 4.2346934 2.9279961
Lazio                    6.2830923 6.5275516 5.4814573
Abruzzo                  1.6572781 2.5881001 2.1166717
Molise                   1.8292880 3.0925526 1.9637798
Campania                 2.3180231 2.2513857 1.6733612
Puglia                   2.5193981 3.5508823 3.1127879
Basilicata               1.0619916 1.8157209 1.7020892
Calabria                 0.0000000 1.3679971 1.3313309
Sicilia                  1.3679971 0.0000000 2.1535893
Sardegna                 1.3313309 2.1535893 0.0000000
```

Come già detto, la metrica Euclidea viene molto utilizzata, ma ne esistono molte altre, ovvero quella di Manhattan, di Chebycev, di Minkowski, di Canberra. Saranno applicate come prova sulle prime 5 regioni. Esiste anche la distanza di Jaccard, che però può essere applicata esclusivamente per vettori binari.

```
> df5 <- data.frame(colazione_adeguata=c(81.2,79,85.4,81.3,84.5),
+ colazione_con_latte=c(40.6,38.7,47.3,39.7,38.9),
+ casa=c(68.1,63.4,69.2,61.8,67.8),
+ mensa=c(9.1,11.5,8,12.4,11.5),
+ ristorante=c(2.7,5.2,3.4,5.2,6.8),
+ bar=c(3.6,3.4,3.9,2.4,1.3),
+ lavoro=c(9.7,8.3,9.4,11.1,6.5),
+ pranzo=c(61.7,69.1,62.2,59.1,75.1),
+ cena=c(29,21,28.9,29.3,13.2))
```

```
> rownames(df5) <- c("Piemonte", "Valle_d_Aosta", "Liguria", "Lombardia", "Trentino_Alto_Adige")
> df5
                  colazione_adeguata colazione_con_latte casa mensa ristorante bar lavoro pranzo cena
Piemonte                        81.2                40.6 68.1   9.1        2.7 3.6    9.7   61.7 29.0
Valle_d_Aosta                   79.0                38.7 63.4  11.5        5.2 3.4    8.3   69.1 21.0
Liguria                         85.4                47.3 69.2   8.0        3.4 3.9    9.4   62.2 28.9
Lombardia                       81.3                39.7 61.8  12.4        5.2 2.4   11.1   59.1 29.3
Trentino_Alto_Adige             84.5                38.9 67.8  11.5        6.8 1.3    6.5   75.1 13.2
> df5scaled <- scale(df5)
```

1. Metrica del valore assoluto (metrica di Manhattan):

```
> dist(df5scaled, method="manhattan", diag=TRUE, upper=TRUE)
                   Piemonte Valle_d_Aosta   Liguria Lombardia Trentino_Alto_Adige
Piemonte           0.000000      8.900445  5.379334  7.919108           13.924173
Valle_d_Aosta      8.900445      0.000000 12.889219  7.394914            9.519115
Liguria            5.379334     12.889219  0.000000 12.361894           15.397999
Lombardia          7.919108      7.394914 12.361894  0.000000           13.173777
Trentino_Alto_Adige 13.924173      9.519115 15.397999 13.173777            0.000000
```

2. Metrica del massimo (metrica di Chebycev):

```
> dist(df5scaled, method="maximum", diag=TRUE, upper=TRUE)
                  Piemonte Valle_d_Aosta  Liguria Lombardia Trentino_Alto_Adige
Piemonte          0.000000      1.536318 1.872122  1.937957            2.519561
Valle_d_Aosta     1.536318      0.000000 2.438737  1.630223            2.095789
Liguria           1.872122      2.438737 0.000000  2.367166            2.438334
Lombardia         1.937957      1.630223 2.367166  0.000000            2.678224
Trentino_Alto_Adige 2.519561    2.095789 2.438334  2.678224            0.000000
```

3. Metrica di Minkowski:

```
> dist(df5scaled, method="minkowski", diag=TRUE, upper=TRUE)
                  Piemonte Valle_d_Aosta  Liguria Lombardia Trentino_Alto_Adige
Piemonte          0.000000      3.216374 2.613741  3.379799            5.203845
Valle_d_Aosta     3.216374      0.000000 4.759117  2.925005            3.770141
Liguria           2.613741      4.759117 0.000000  4.705872            5.592488
Lombardia         3.379799      2.925005 4.705872  0.000000            5.047800
Trentino_Alto_Adige 5.203845    3.770141 5.592488  5.047800            0.000000
```

4. Metrica di Canberra:

```
> dist(df5scaled, method="canberra", diag=TRUE, upper=TRUE)
                  Piemonte Valle_d_Aosta  Liguria Lombardia Trentino_Alto_Adige
Piemonte          0.000000      7.360454 3.247588  5.342899            7.738280
Valle_d_Aosta     7.360454      0.000000 8.342466  5.353204            5.197784
Liguria           3.247588      8.342466 0.000000  7.045085            7.455425
Lombardia         5.342899      5.353204 7.045085  0.000000            6.651264
Trentino_Alto_Adige 7.738280    5.197784 7.455425  6.651264            0.000000
```

1.5.1.1 Misure di similarità

Nella maggior parte delle tecniche di clustering occorre inizialmente calcolare la matrice *D* delle distanze oppure una matrice *S* delle similarità. Una *misura di similarità* fornisce un valore numerico compreso tra 0 e 1 e permette di definire in modo quantitativo la somiglianza o differenza tra due individui I_i e I_j, intendendo ovviamente con 0 l'assoluta assenza e con 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*:

- $s(X_i, X_j) = 1$, che implica che la misura di similarità è unitaria se i due punti sono identici;
- $0 \leq s(X_i, X_j) \leq 1$ che afferma che la misura di similarità è compresa tra 0 e 1;
- $s(X_i, X_j) = s(X_j, X_i)$ per ogni X_i e X_j, che impone la simmetria richiedendo che la misura di similarità X_i e X_j deve essere la stessa della misura di similarità in X_j e X_i.

La quantità s_{ij} è chiamata semplicemente *coefficiente di similarità* e risulta essere l'elemento nella riga i-esima e colonna j-esima della matrice di similarità *S*, così definita:

$$S = \begin{pmatrix} 1 & s_{12} & \dots & s_{1n} \\ s_{21} & 1 & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & 1 \end{pmatrix}$$

1.5.2 Misura di non omogeneità totale

Riconsideriamo un insieme $I = \{I_1, I_2, ..., I_n\}$ di n individui e assumiamo che esista un insieme di caratteristiche $C = \{C_1, C_2, ..., C_p\}$ che sono osservabili e sono possedute da ogni individuo in I. Sia

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix}$$

la *matrice delle misure* (supponiamo di effettuare lo scalamento se le colonne hanno unità di misura differenti) e sia

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

la *matrice delle distanze* di cardinalità $n \times n$, dove $d_{ij} = d(X_i, X_j)$ ($i, j = 1, 2, ..., n$).
Alla matrice X si può associare una matrice W_X di cardinalità p x p, detta *matrice delle varianze e covarianze* così definita:

$$W_X = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p1} & w_{p2} & \dots & w_{pp} \end{pmatrix}$$

In R, è possibile ottenere la matrice W_X delle *varianze e covarianze campionarie tra le varie caratteristiche* grazie alla funzione **cov(X)**. A partire dalla matrice delle covarianze è possibile definire la *matrice statistica di non omogeneità*.
La matrice statistica di non omogeneità per l'insieme I di individui, di cardinalità p x p, è così definita:

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

dove in generale, l'elemento $h_{rr} = (n - 1)\, s^2_r$.
Ciò che risulta interessante è la *traccia della matrice* H_I, ovvero la *somma degli elementi sulla diagonale della matrice* H_I, cioè:

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

La *traccia della matrice* H_I può essere espressa anche nel modo seguente:

$$\mathrm{tr}H_I = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=i}^{n} d_2^2(X_i, X_j),$$

ossia la traccia della matrice di non omogeneità statistica corrisponde al rapporto tra la somma dei

quadrati degli elementi al di sotto della diagonale principale della matrice delle distanze euclidee e il numero *n* di individui. La distanza euclidea gioca dunque un ruolo rilevante nel calcolo della misura di non omogeneità statistica. Inoltre, tale misura dipende *sia dall'omogeneità interna sia dalla numerosità del gruppo.*

1.5.3 Misura di non omogeneità tra cluster

Accanto alle misure di non omogeneità relative all'insieme totale di individui della popolazione, occorre anche definire delle misure di non omogeneità all'interno dei cluster e delle misure di non omogeneità (o disparità) tra cluster distinti. Al termine del procedimento di classificazione, gli individui appartenenti allo stesso cluster dovrebbero essere il più possibile omogenei tra di loro e il più possibile differenti da quelli appartenenti agli altri cluster individuati.
Si desidera partizionare un insieme $I = \{I_1, ..., I_n\}$ di *n* individui in *m* particolari cluster. Denotando con $G_1, G_2, ..., G_m$ una partizione degli *n* individui $I = \{I_1, ..., I_n\}$ in *m* cluster. Denotiamo con:

$$T = \mathrm{H_I}$$

la matrice di non omogeneità statistica relativa all'insieme totale $I = \{I_1, ..., I_n\}$ di *n* individui, con

$$S = H_{G1} + H_{G2} + ... + H_{Gm}$$

la somma delle matrici di non omogeneità statistica relative ai singoli *m* cluster (*within*) e con

$$B = \sum_{i<j} H_{G_i \cap G_j} + \sum_{i<j<k} H_{G_i \cap G_j \cap G_k} + \ldots + H_{G_1 \cap \ldots \cap G_m}$$

la matrice di non omogeneità statistica tra i vari cluster considerati (*between*), possiamo scrivere la seguente equazione matriciale:

$$T = S + B$$

Le matrici *T, S* e *B* hanno cardinalità *p* x *p*. La matrice *T* è fissata. Invece, le matrici *S* e *B* dipendono strettamente dalla partizione in cluster dell'insieme *I* di individui considerata. Per ogni partizione dell'insieme *I* degli *n* individui in *m* fissati cluster, otteniamo un'equazione matriciale come la precedente, da cui segue

$$tr\ T = tr\ S + tr\ B$$

o equivalentemente

$$1 = \frac{\mathrm{tr}\ S}{\mathrm{tr}\ T} + \frac{\mathrm{tr}\ B}{\mathrm{tr}\ T}.$$

Poiché tr *T* è univocamente determinata per ogni matrice X di cardinalità *n* x *p*, fissato il numero *m* di suddivisioni, i cluster dovrebbero essere individuati in modo da *minimizzare la misura di non omogeneità statistica all'interno dei cluster (within) e massimizzare la misura di non omogeneità statistica tra i gruppi (between).*

1.5.4 Metodi di ottimizzazione

Una volta scelta la misura di distanza (o di similarità) è necessario scegliere un algoritmo di raggruppamento delle unità osservate.
I metodi di raggruppamento si distinguono in tre tipi:

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

Supponiamo di considerare un insieme I_1, I_2, ..., I_n di n individui e sia m il numero di cluster.
Un *metodo di enumerazione completa* consiste nel valutare la traccia della somma delle matrici di non omogeneità relative ai singoli cluster per ogni possibile partizione dell'insieme totale degli n individui in m cluster.
La traccia di una matrice di non omogeneità di un insieme di individui fornisce una misura della dispersione dei dati intorno al valore medio dell'insieme dal quale è stata ricavata.
Il *problema* principale che si presenta utilizzando le tecniche di enumerazione completa (tecniche di ottimizzazione) è che esse sono computazionalmente onerose poiché prevedono il calcolo della funzione obiettivo per ogni possibile partizione dell'insieme totale di n individui in m cluster. Per questo, spesso si adottano i metodi di raggruppamento *gerarchici* e *non gerarchici* che operano su una sottoclasse delle partizioni degli n individui in cluster.

1.5.5 Metodi gerarchici

I metodi di clustering gerarchico possono essere di due tipi: *agglomerativi* e *divisivi*.
I *metodi gerarchici di tipo agglomerativo* partono da una situazione in cui si hanno m cluster distinti ognuno contenente un solo individuo per giungere, attraverso successive unioni dei cluster meno distanti tra loro, ad una situazione in cui si ha un solo cluster che contiene tutti gli n individui. Invece, i *metodi gerarchici di tipo divisivo* partono da una situazione in cui si ha un solo cluster che contiene tutti gli n individui per giungere, attraverso successive divisioni dei cluster più distanti tra loro, ad una situazione in cui si hanno m cluster distinti ognuno contenente un solo individuo. Lo *svantaggio* dei metodi gerarchici è che non consentono di riallocare gli individui che sono stati già classificati ad un livello precedente dell'analisi (cosa che invece consentono i metodi non gerarchici).
L'obiettivo dei metodi gerarchici è quello di ottenere una sequenza di partizioni che possono essere rappresentate graficamente mediante una struttura ad albero, il *dendrogramma*, nella quale sull'insieme delle ordinate sono riportati i livelli di distanza mentre sull'asse delle ascisse sono riportati i singoli individui. Considereremo i metodi gerarchici di tipo *agglomerativo*, dove inizialmente viene valutata la *matrice delle distanze D*. La funzione distanza comunemente utilizzata è quella *euclidea*.

Molti metodi gerarchici hanno una struttura comune, che può essere riassunta tramite il seguente algoritmo:

- **Step 1**: A partire dalla matrice X (o quella scalata) calcolare la matrice delle distanze *D*;
- **Step 2**: individuare la coppia di cluster meno distanti e raggruppare in un unico cluster i due cluster meno distanti;
- **Step 3**: costruire una nuova matrice delle distanze ridotta di una riga e di una colonna e calcolare le nuove distanze tra i cluster;
- **Step 4**: ripetere la procedura a partire dal passo 2 fino ad ottenere una matrice 2x2;
- **Step 5**: rappresentare graficamente attraverso un dendrogramma che riporta sull'asse verticale il livello di distanza a cui avviene l'agglomerazione e sull'asse orizzontale riporta gli individui.

I vari metodi si differenziano principalmente nei passi 1 e 2. Infatti, può essere utilizzata, oltre alla matrice delle distanze *D* anche quella delle similarità *S*. Mentre per quanto riguarda il secondo step, esistono vari metodi.
L'analisi gerarchica di tipo agglomerativo viene effettuata in R attraverso la funzione **hclust(d, method="complete")**, dove *d* rappresenta un oggetto creato tramite la funzione **dist()**, mentre *method* seleziona il metodo gerarchico agglomerativo (di default è complete). Ulteriori metodi

sono i seguenti:

- Metodo del legame singolo (single);
- Metodo del legame completo (complete);
- Metodo del legame medio (average);
- Metodo del centroide (centroid);
- Metodo della mediana (median).

1.5.5.1 Metodo del legame singolo

In questo metodo la distanza tra i gruppi $G1$ (contenente n_1 individui) e $G2$ (contenente n_2 individui) è definita come la minima tra tutte le n_1n_2 distanze che si possono calcolare tra ogni individuo di $G1$ e ogni individuo di $G2$. Un *vantaggio* del metodo del legame singolo è di consentire di individuare gruppi di qualsiasi forma (anche non ellissoidali, ossia anche con forme allungate e scarsamente omogenei al loro interno) e di mettere in luce eventuali valori anomali meglio di altre tecniche gerarchiche. Uno *svantaggio* di tale metodo è che ad un certo livello di distanza d, si può verificare che si vengano a trovare nello stesso cluster individui piuttosto dissimili.
Dopo aver calcolato la matrice delle distanze utilizzando la metrica euclidea, viene applicato il *metodo del legame singolo*:

```
> d <- dist(tipiDiPasto, method="euclidean", diag=TRUE, upper=TRUE)
> hls <- hclust(d, method="single")
> str(hls)
List of 7
 $ merge      : int [1:19, 1:2] -10 -3 -8 -18 -20 -14 -6 -1 3 -13 ...
 $ height     : num [1:19] 4.41 4.73 5.06 5.32 5.4 ...
 $ order      : int [1:20] 10 11 15 13 20 18 19 17 14 16 ...
 $ labels     : chr [1:20] "Piemonte" "Valle_d_Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "single"
 $ call       : language hclust(d = d, method = "single")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

il comando **str(hls)** permette di visualizzare informazioni sull'oggetto cluster.
I risultati di $**merge**, che contiene la sequenza del processo di agglomerazione, vengono disposti su due colonne: i numeri con il segno negativo indicano i singoli individui, mentre quelli positivi

```
> hls$merge
      [,1] [,2]
 [1,]  -10  -11
 [2,]   -3   -9
 [3,]   -8    2
 [4,]  -18  -19
 [5,]  -20    4
 [6,]  -14  -16
 [7,]   -6   -7
 [8,]   -1    7
 [9,]    3    8
[10,]  -13    5
[11,]  -17    6
[12,]   10   11
[13,]  -12    9
[14,]  -15   12
[15,]   -2   13
[16,]   -4   15
[17,]    1   14
[18,]   -5   16
[19,]   17   18
```

indicano i cluster che si formano.

$**height** indica il livello di distanza a cui è avvenuta la fusione di due cluster.
$**order** è una permutazione delle regioni per costruire il dendrogramma.
$**labels** sono le etichette.
Per mostrare il raggruppamento in modo più chiaro, è stato rappresentato, attraverso le seguenti righe di codice, il dendrogramma in Figura 1.41:

```
> plot(hls, hang=-1, xlab="Metodo gerarchico agglomerativo", sub="del legame singolo")
> axis(side=4, at=round(c(0, hls$height),2))
```

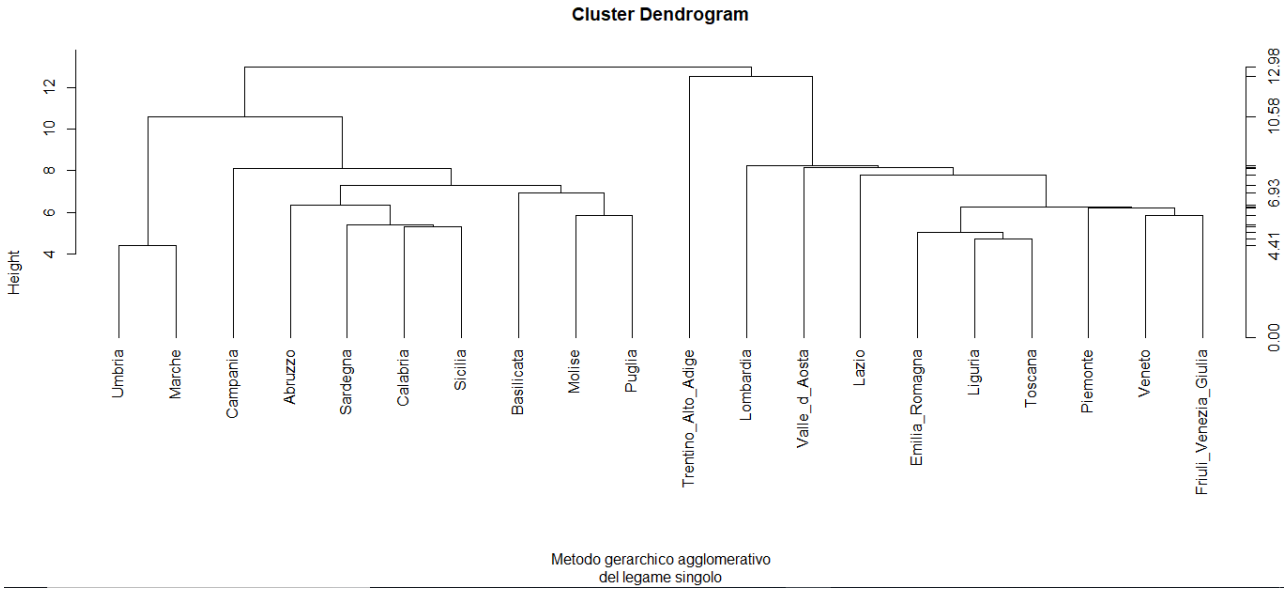


Figura 1.41: Dendrogramma ottenuto con il metodo del legame singolo.

1.5.5.2 Metodo del legame completo

In questo metodo la distanza tra i gruppi $G1$ (contenente n_1 individui) e $G2$ (contenente n_2 individui) è definita come la massima tra tutte le $n_1 n_2$ distanze che si possono calcolare tra ogni individuo di $G1$ e ogni individuo di $G2$. La massima distanza esistente tra gli individui dei due cluster rappresenta il *diametro della sfera che contiene tutti i punti appartenenti ai due gruppi*. Questo metodo identifica soprattutto gruppi di forma ellissoidale, ossia una serie di punti che si addensano intorno ad un nucleo centrale. Questo algoritmo *privilegia* l'omogeneità tra gli elementi del gruppo a *scapito* della differenziazione tra i gruppi.
Viene ora applicato alla matrice delle distanze d calcolata precedentemente.

```
> hlc <- hclust(d, method="complete")
> str(hlc)
List of 7
 $ merge      : int [1:19, 1:2] -10 -3 -18 -8 -14 -6 -13 -1 -4 -17 ...
 $ height     : num [1:19] 4.41 4.73 5.32 5.76 5.84 ...
 $ order      : int [1:20] 17 14 16 10 11 15 18 19 13 20 ...
 $ labels     : chr [1:20] "Piemonte" "Valle_d_Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "complete"
 $ call       : language hclust(d = d, method = "complete")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
> hlc$merge
      [,1] [,2]
 [1,]  -10  -11
 [2,]   -3   -9
 [3,]  -18  -19
 [4,]   -8    2
 [5,]  -14  -16
 [6,]   -6   -7
 [7,]  -13  -20
 [8,]   -1  -12
 [9,]   -4    8
[10,]  -17    5
[11,]   -2    6
[12,]    3    7
[13,]  -15   12
[14,]    4    9
[15,]   -5   11
[16,]    1   13
[17,]   10   16
[18,]   14   15
[19,]   17   18
```

Attraverso le seguenti righe di codice, viene poi costruito il dendrogramma in Figura 1.42.

```
> plot(hlc, hang=-1, xlab="Metodo gerarchico agglomerativo", sub="del legame completo")
> axis(side=4, at=round(c(0, hlc$height),2))
```

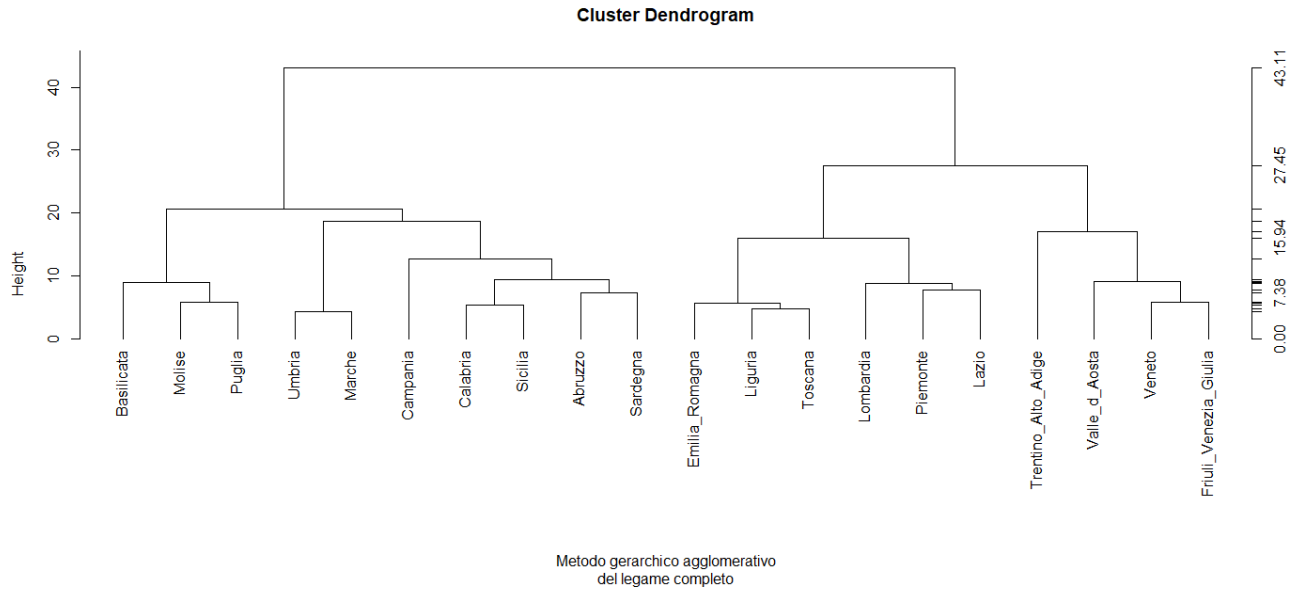


Figura 1.42: Dendrogramma ottenuto con il metodo del legame completo.

1.5.5.3 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. Uno *svantaggio* è che se le misure dei due cluster da unire sono troppo distanti, la distanza media tra gli elementi del cluster sarà molto vicina a quella del cluster più numeroso.

```
> hlm <- hclust(d, method="average")
> str(hlm)
List of 7
 $ merge      : int [1:19, 1:2] -10 -3 -18 -8 -14 -6 -20 -13 -1 -17 ...
 $ height     : num [1:19] 4.41 4.73 5.32 5.41 5.84 ...
 $ order      : int [1:20] 10 11 17 14 16 15 13 20 18 19 ...
 $ labels     : chr [1:20] "Piemonte" "Valle_d_Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "average"
 $ call       : language hclust(d = d, method = "average")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
> hlm$merge
      [,1] [,2]
 [1,]  -10  -11
 [2,]   -3   -9
 [3,]  -18  -19
 [4,]   -8    2
 [5,]  -14  -16
 [6,]   -6   -7
 [7,]  -20    3
 [8,]  -13    7
 [9,]   -1    6
[10,]  -17    5
[11,]   -4  -12
[12,]    4    9
[13,]  -15    8
[14,]   11   12
[15,]   -2   -5
[16,]   10   13
[17,]    1   16
[18,]   14   15
[19,]   17   18
```

Attraverso le seguenti righe di codice, viene poi costruito il dendrogramma in Figura 1.43.

```
> plot(hlm, hang=-1, xlab="Metodo gerarchico agglomerativo", sub="del legame medio")
> axis(side=4, at=round(c(0, hlm$height),2))
```

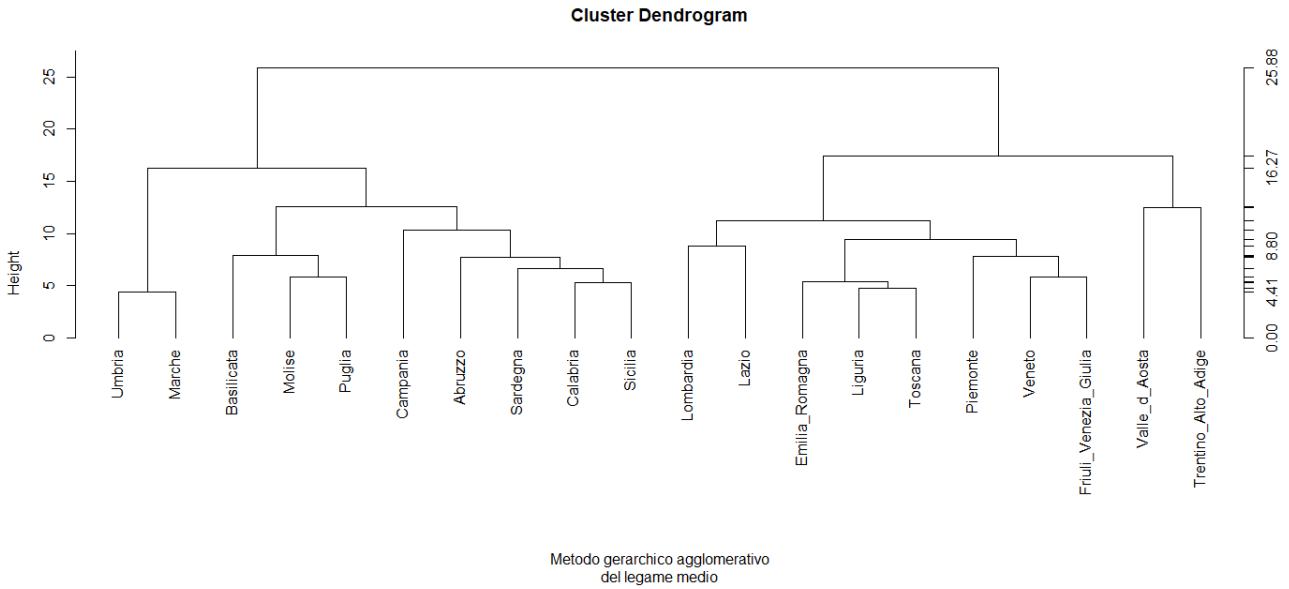


Figura 1.43: Dendrogramma ottenuto con il metodo del legame medio.

1.5.5.4 Metodo del centroide

In questo metodo la distanza tra il gruppo $G1$ e il gruppo $G2$ è definita come la distanza tra i centroidi, ossia tra le medie campionarie calcolate sugli individui appartenenti ai due gruppi. Questo metodo può dare origine a fenomeni gravitazionali, per cui i gruppi grandi tendono ad attrarre al loro interno i piccoli gruppi. Inoltre, le distanze in cui si verificano le successive agglomerazioni possono essere non crescenti. Uno *svantaggio* del metodo del centroide è che se le misure dei due cluster da unire sono molto differenti il centroide del nuovo cluster sarà molto vicino a quello del cluster più numeroso.
Nei metodi agglomerativi del legame singolo, del legame completo e del legame medio si può utilizzare una qualsiasi misura di distanza. Invece, nel metodo del centroide e nel metodo della mediana si considera la distanza euclidea e si lavora con una matrice D^2 che contiene i *quadrati delle singole distanze euclidee*.
Viene calcolata dunque la matrice contenente i quadrati delle distanze euclidee a cui viene poi applicato il metodo gerarchico del centroide:

```
> d2 <- d^2
> hc <- hclust(d2, method="centroid")
> str(hc)
List of 7
 $ merge      : int [1:19, 1:2] -10 -3 -8 -18 -14 -6 -20 -13 -1 -17 ...
 $ height     : num [1:19] 19.4 22.4 23.8 28.3 34.1 ...
 $ order      : int [1:20] 10 11 17 14 16 15 13 20 18 19 ...
 $ labels     : chr [1:20] "Piemonte" "Valle_d_Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "centroid"
 $ call       : language hclust(d = d2, method = "centroid")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
>
> hc$merge
      [,1] [,2]
 [1,]  -10  -11
 [2,]   -3   -9
 [3,]   -8    2
 [4,]  -18  -19
 [5,]  -14  -16
 [6,]   -6   -7
 [7,]  -20    4
 [8,]  -13    7
 [9,]   -1    6
[10,]  -17    5
[11,]    3    9
[12,]   -4  -12
[13,]   11   12
[14,]  -15    8
[15,]   10   14
[16,]   -2   -5
[17,]    1   15
[18,]   13   16
[19,]   17   18
```

Attraverso le seguenti righe di codice, viene poi costruito il dendrogramma in Figura 1.44.

```
> plot(hc, hang=-1, xlab="Metodo gerarchico agglomerativo", sub="del centroide")
> axis(side=4, at=round(c(0, hc$height),2))
```

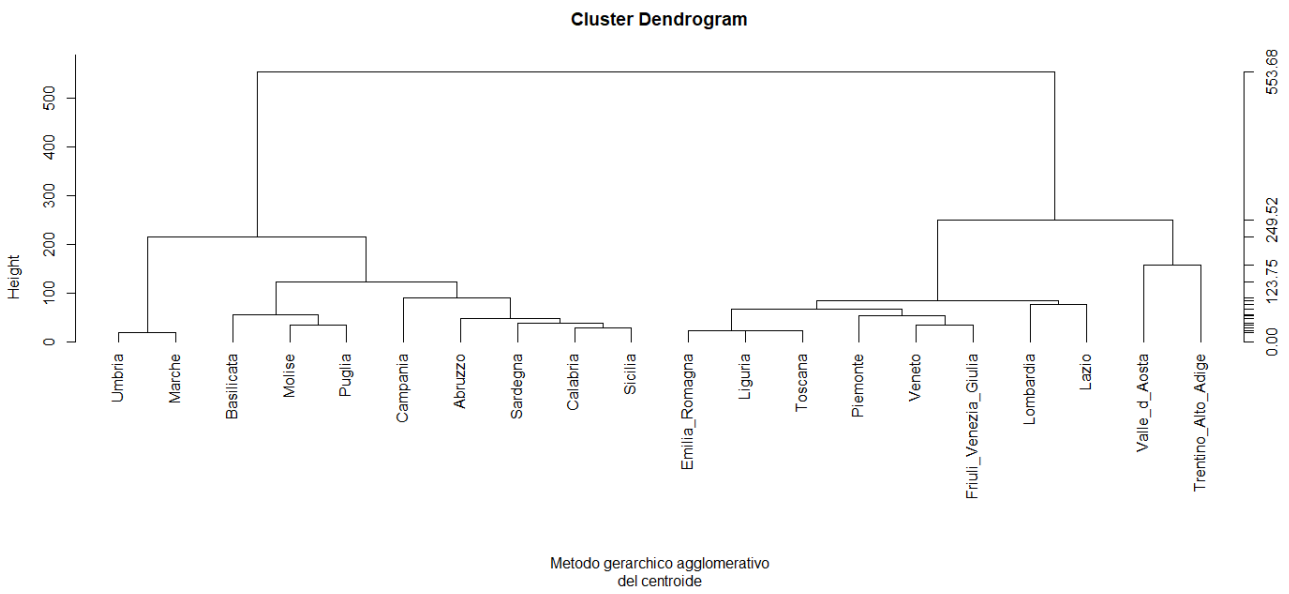


Figura 1.44: Dendrogramma ottenuto con il metodo del centroide.

1.5.5.5 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 metodo è necessario utilizzare la matrice contenente i quadrati delle distanze euclidee a cui viene poi applicato il metodo gerarchico della mediana:

```
> hmed <- hclust(d2, method="median")
> str(hmed)
List of 7
 $ merge      : int [1:19, 1:2] -10 -3 -8 -18 -14 -6 -20 -13 -1 3 ...
 $ height     : num [1:19] 19.4 22.4 23.8 28.3 34.1 ...
 $ order      : int [1:20] 10 11 15 13 20 18 19 17 14 16 ...
 $ labels     : chr [1:20] "Piemonte" "Valle_d_Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "median"
 $ call       : language hclust(d = d2, method = "median")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
> hmed$merge
      [,1] [,2]
 [1,]  -10  -11
 [2,]   -3   -9
 [3,]   -8    2
 [4,]  -18  -19
 [5,]  -14  -16
 [6,]   -6   -7
 [7,]  -20    4
 [8,]  -13    7
 [9,]   -1    6
[10,]    3    9
[11,]  -17    5
[12,]   -4  -12
[13,]   10   12
[14,]    8   11
[15,]   -2   -5
[16,]  -15   14
[17,]    1   16
[18,]   13   15
[19,]   17   18
```

Attraverso le seguenti righe di codice, viene poi costruito il dendrogramma in Figura 1.45.

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

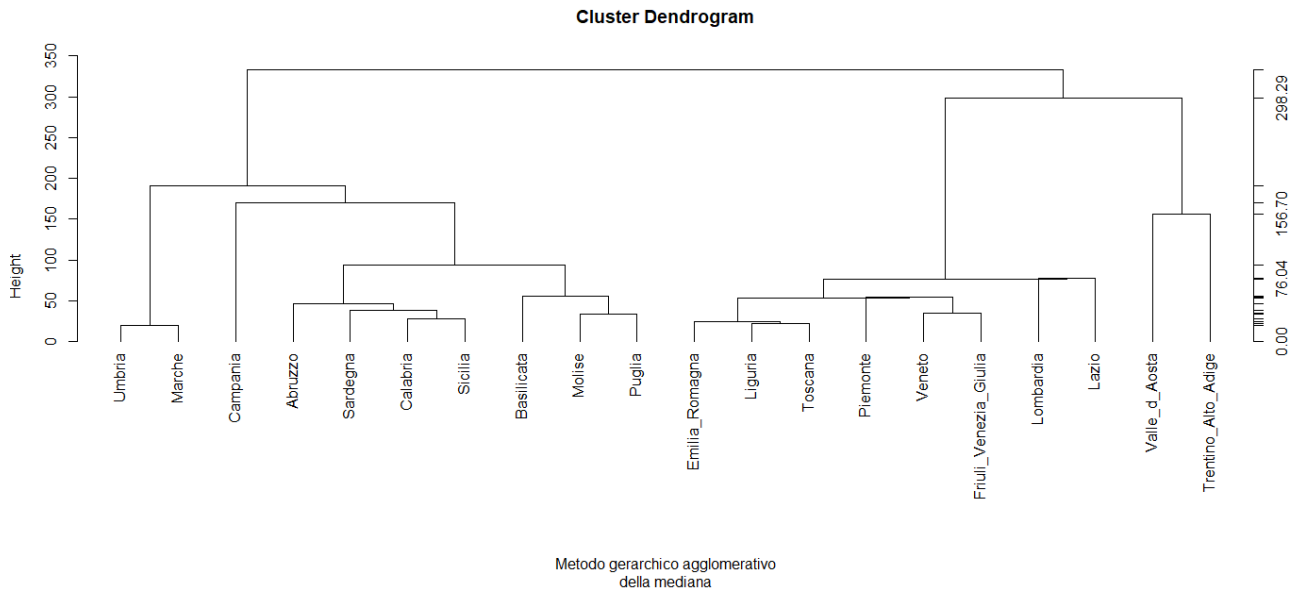


Figura 1.45: Dendrogramma ottenuto con il metodo della mediana.

1.5.5.5 Screeplot

Al fine di scegliere una buona partizione del dendrogramma, si può considerare una procedura empirica consistente nel costruire un grafico, detto *screeplot*: sull'asse delle ordinate si pongono i numeri di gruppi ottenibili con il metodo gerarchico e sull'asse delle ascisse le distanze a cui avvengono le successive aggregazioni tra i gruppi. Se nel passaggio da *k* gruppi a *k* - 1 gruppi si registra un forte incremento della distanza di aggregazione è consigliabile tagliare il dendrogramma in *k* gruppi. La procedura empirica basata sullo screeplot non sempre fornisce la suddivisione in cluster più adeguata; è sempre preferibile utilizzare le misure di non omogeneità statistiche. Tale procedura empirica, prende in considerazione la seguente quantità:

$$\delta_k = d_{k-1} - d_k \quad (k = 2, ..., n),$$

dove d_k rappresenta il livello di distanza a cui è stata effettuata l'agglomerazione in *k* gruppi e *n* è il numero iniziale di individui. Quando l'incremento δ_k risulta sufficientemente elevato, significa che i gruppi sono sufficientemente dissimili tra loro, per cui è possibile tagliare il dendrogramma all'altezza (al livello di distanza) corrispondente alla partizione in *k* gruppi.
Lo screeplot fornisce una visione di insieme delle altezze a cui sono avvenute le agglomerazioni e si potrebbe scegliere il valore di *j* per il quale $\delta_j = \max\{\delta_1, \delta_2, ..., \delta_n\}$.
È preferibile costruire lo screeplot a partire dal *metodo del legame singolo*, del *legame completo* o del *legame medio* in cui è utilizzata la funzione distanza. Invece, nel metodo del centroide e della mediana (che utilizzano i quadrati delle distanze) le successive agglomerazioni potrebbero verificarsi ad un livello di distanza minore o uguale rispetto alle precedenti agglomerazioni. Ciò comporta che lo *screeplot* ottenuto a partire dal metodo del centroide o della mediana potrebbe non essere regolare e non fornire indicazioni adeguate.
Verrà ora costruito lo screeplot relativo al metodo del legame completo.

Applicando il comando $**height** ai valori precedentemente ottenuti tramite il metodo del legame completo, si ottiene:

```
> hlc$height
 [1]  4.407947  4.728636  5.316954  5.759340  5.841233  5.873670  7.384443
 [8]  7.799359  8.803976  8.938121  9.172786  9.474175 12.782019 15.936122
[15] 17.095613 18.737929 20.630560 27.449590 43.106380
```

Attraverso la seguente riga di codice, è possibile ottenere lo screeplot rappresentato in Figura 1.47. La funzione **c(0, hlc$height)** permette di concatenare 0 con il vettore *hlc$height* delle altezze a cui sono avvenute le successive agglomerazioni.

```
> plot(rev(c(0, hlc$height)), seq(20,1), type="b", main="Screeplot",
+ xlab="Distanza di aggregazione", ylab="Numero di cluster", col="red")
```

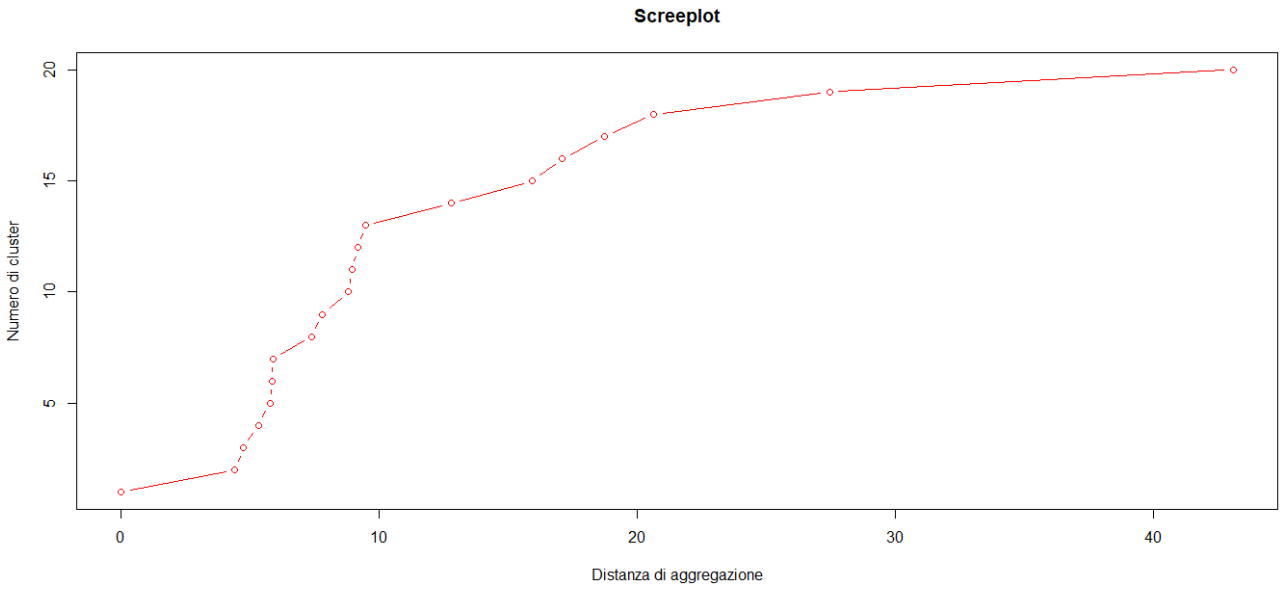


Figura 1.47: Screeplot per il metodo del legame completo

Lo screeplot in Figura 1.47 suggerisce di considerare una suddivisione in due gruppi. Infatti, applicando la formula si ha che il valore di *k* per il quale δ_k è massima è k = 2 dato da h_1 - h_2 = 43.106380 - 27.449590 = 15.65679. È preferibile quindi considerare una suddivisione in due cluster.

Tuttavia, come già precedentemente annunciato, lo screeplot è soltanto un grafico basato sulle altezze a cui sono avvenute le aggregazioni e non sempre fornisce il numero adeguato di cluster in cui suddividere gli individui.

1.5.6 Analisi del dendogramma

Adesso analizzeremo il dendrogramma ottenuto con un particolare metodo gerarchico e di calcolare, fissato il numero di cluster, le misure di non omogeneità della partizione individuata.

1.5.6.1 Disegnare rettangoli che evidenziano i cluster

Consideriamo un particolare dendrogramma ottenuto a partire dalla funzione **hclust**. La funzione **rect.hclust()** permette di disegnare dei rettangoli intorno ai cluster, individuati in base all'altezza *h* alla quale si opera il taglio del dendrogramma oppure in base al numero *k* di cluster che si vogliono ottenere attraverso la funzione:

```
> rect.hclust(z, h = NULL, k = NULL, border = "color")
```

dove

- **z** è l'oggetto creato (output) dalla funzione hclust;
- **h** è l'altezza alla quale si inserisce il taglio;
- **k** è il numero di cluster che si vogliono ottenere;
- **border** è il colore dei contorni dei rettangoli.

Si vuole evidenziare due partizioni all'interno del dendrogramma ottenuto dal metodo del legame completo. Lo si può realizzare tramite la seguente riga di codice, in cui *k* è il numero di cluster che si vogliono ottenere. L'output è mostrato in Figura 1.49.

```
> tree <- hclust(d, method='complete')
>
> plot(tree , hang=-1, xlab = "Metodo legame completo")
> axis( side =4, at= round (c(0, tree$height),2))
> rect.hclust(tree, k=2, border='red')
```

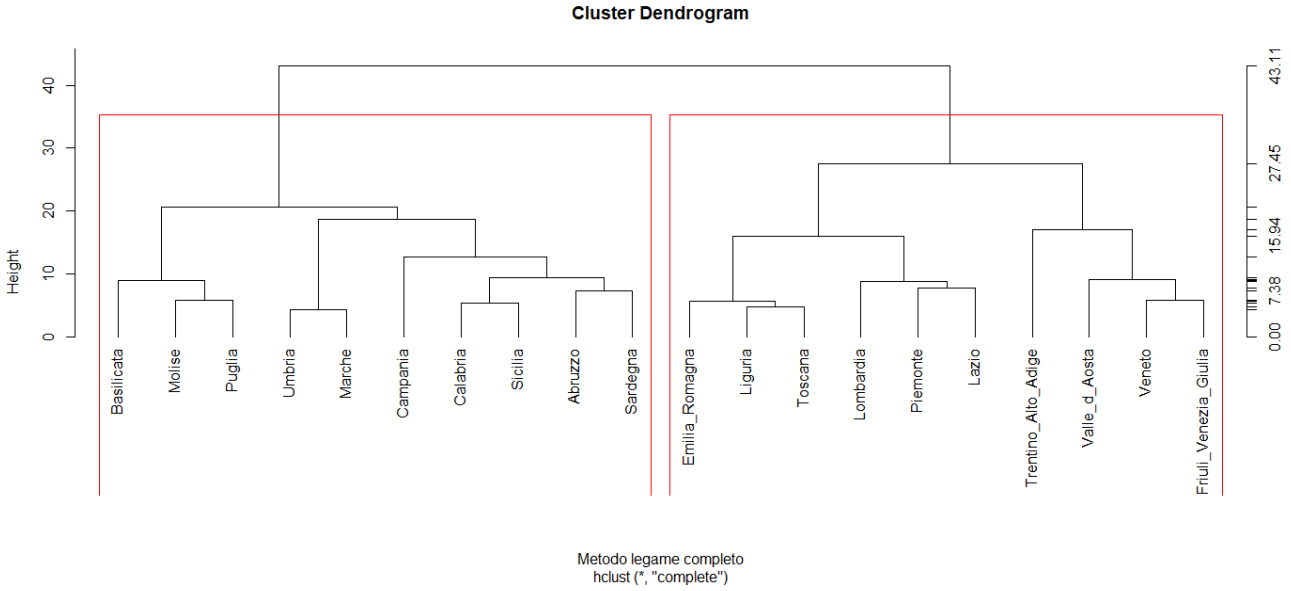


Figura 1.49: Rettangoli che evidenziano le due partizioni

Possiamo anche evidenziare la presenza di 3 partizioni con le seguenti linee di codice:

```
> plot(tree , hang=-1, xlab = "Metodo legame completo")
> axis( side =4, at= round (c(0, tree$height),2))
> rect.hclust(tree, k=3, border='green')
```

Il grafico prodotto è mostrato in Figura 1.50

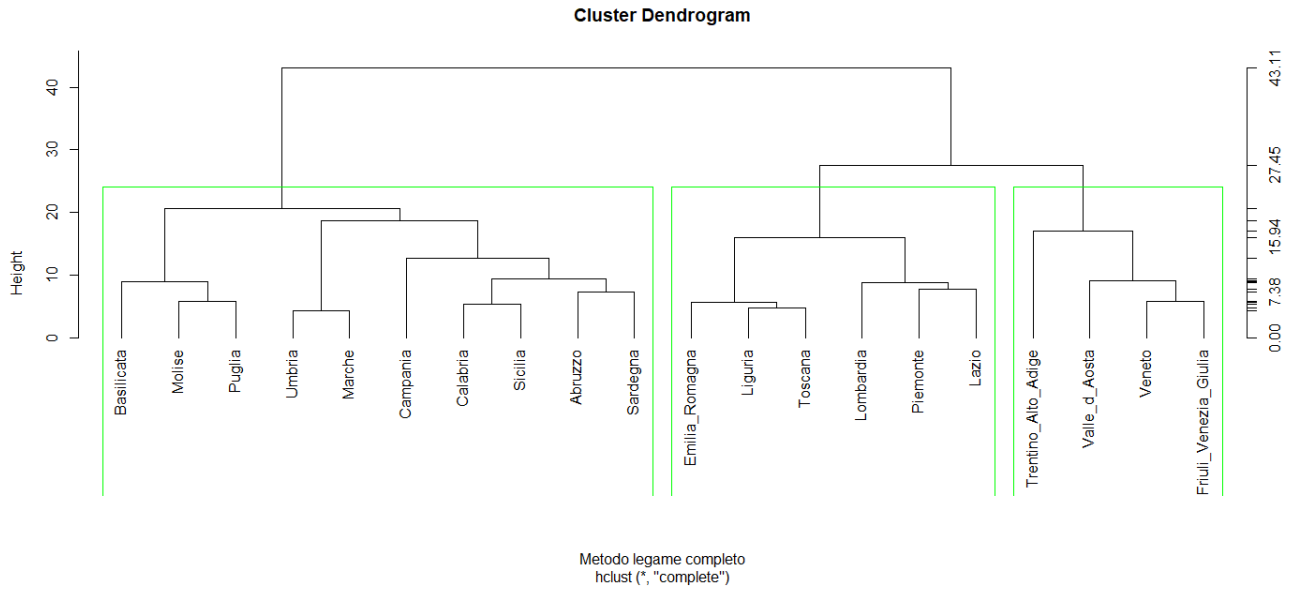


Figura 1.50: Rettangoli che evidenziano le tre partizioni

Dalle figure 1.49 e 1.50 è possibile capire che nel primo caso, nella suddivisione con k=2, si ha una suddivisione equa delle regioni, mentre nel secondo caso, di k=3 le regioni che prima erano in un solo cluster adesso sono divise in due.

1.5.6.2 Disegnare rettangoli che evidenziano i cluster

Considerato un particolare dendrogramma, per ottenere una suddivisione degli individui in cluster in corrispondenza di un determinato livello di distanza oppure in corrispondenza di un prefissato numero di cluster, R utilizza anche la funzione **cutree()** nel seguente modo:

$$> \text{cutree}(\text{tree}, k = \text{NULL}, h = \text{NULL})$$

dove:

- **tree** rappresenta un oggetto (che individua un dendrogramma) creato tramite la funzione **hclust();**
- **k** è il numero prefissato di cluster;
- **h** è l'altezza alla quale il dendrogramma viene tagliato.

Inoltre, per vedere come vengono classificati gli individui all'aumentare del numero di cluster si può considerare la funzione **cutree(tree, k=1:n)**, dove *n* indica il numero di individui.
L'output della funzione **cutree()** è un vettore contenente numeri interi positivi associati ai cluster in cui sono stati inseriti i vari individui.
La funzione **cutree()** viene dunque applicata ai vari metodi, mostrando le diverse suddivisioni per ognuno dei relativi dendrogrammi.

Metodo del legame singolo
Con k=2, il metodo del legame singolo ottiene una suddivisione nei vari cluster bilanciata, invece con k=3 la suddivisione risulta bilanciata, ma con un cluster aggiuntivo. Infatti, applicando la funzione **cutree()**, una volta con k=2 e un'altra con k=3, si ottiene la seguente suddivisione:

```
> cutree(tree, k=2, h=NULL)
             Piemonte          Valle_d_Aosta               Liguria
                    1                      1                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      1                     1
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    1                      1                     1
               Umbria                 Marche                 Lazio
                    2                      2                     1
             Abruzzo                 Molise              Campania
                    2                      2                     2
              Puglia             Basilicata              Calabria
                    2                      2                     2
             Sicilia               Sardegna
                    2                      2
> cutree(tree, k=3, h=NULL)
             Piemonte          Valle_d_Aosta               Liguria
                    1                      1                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      2                     1
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    1                      1                     1
               Umbria                 Marche                 Lazio
                    3                      3                     1
             Abruzzo                 Molise              Campania
                    3                      3                     3
              Puglia             Basilicata              Calabria
                    3                      3                     3
             Sicilia               Sardegna
                    3                      3
```

Infatti, nel caso di k=2, 10 regioni occupano un cluster e 10 un altro; la situazione quindi è bilanciata. Nel secondo caso con k=3, un cluster contiene 9 regioni, un altro 10 e un altro contiene solo il Trentino Alto-Adige.
Sebbene visivamente possa sembrare che la suddivisione migliore sia quella in due cluster, per isolare le regioni con valori anomali, è necessario avere un cluster con **k=3**.

```
> cutree(tree, k=3, h=NULL)
             Piemonte          Valle_d_Aosta               Liguria
                    1                      1                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      2                     1
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    1                      1                     1
               Umbria                 Marche                 Lazio
                    3                      3                     1
             Abruzzo                  Molise              Campania
                    3                      3                     3
              Puglia             Basilicata              Calabria
                    3                      3                     3
             Sicilia               Sardegna
                    3                      3
```

La linea di codice permette di individuare quindi la seguente partizione in tre cluster:

G_1 = {Lombardia, Valle d'Aosta, Emilia-Romagna, Liguria, Toscana, Piemonte, Veneto, Friuli-Venezia Giulia, Lazio}

G_2 = {Trentino Alto-Adige}

G_3= {Umbria, Marche, Campania, Abruzzo, Sardegna, Calabria, Sicilia, Basilicata, Molise, Puglia}

Metodo del legame completo

Applicando la funzione **cutree()** con k=3 si ottiene la seguente suddivisione:

```
> cutree(tree, k=3, h=NULL)
             Piemonte          Valle_d_Aosta               Liguria
                    1                      2                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      2                     2
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    2                      1                     1
               Umbria                 Marche                 Lazio
                    3                      3                     1
             Abruzzo                  Molise              Campania
                    3                      3                     3
              Puglia             Basilicata              Calabria
                    3                      3                     3
             Sicilia               Sardegna
                    3                      3
```

Le partizioni in due cluster sono quindi rappresentate in questo modo:

G_1 = {Lombardia, Emilia-Romagna, Liguria, Toscana, Piemonte, Lazio}

G_2 = {Trentino Alto-Adige, Valle d'Aosta, Veneto, Friuli Venezia-Giulia}

G_3= {Umbria, Marche, Campania, Abruzzo, Sardegna, Calabria, Sicilia, Basilicata, Molise, Puglia}

Metodo del legame medio

Applicando la funzione **cutree()** per il metodo del legame medio, con k=3 si ottiene:

```
> cutree(tree, k=3, h=NULL)
             Piemonte          Valle_d_Aosta               Liguria
                    1                      2                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      2                     1
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    1                      1                     1
                Umbria                Marche                 Lazio
                    3                      3                     1
               Abruzzo                Molise              Campania
                    3                      3                     3
                Puglia            Basilicata              Calabria
                    3                      3                     3
               Sicilia              Sardegna
                    3                      3
```

G_1 = {Lombardia, Emilia-Romagna, Liguria, Toscana, Piemonte, Lazio, Veneto, Friuli Venezia-Giulia}

G_2 = {Trentino Alto-Adige, Valle d'Aosta}

G_3= {Umbria, Marche, Campania, Abruzzo, Sardegna, Calabria, Sicilia, Basilicata, Molise, Puglia}

Metodo del centroide

La funzione **cutree()** applicando il metodo del centroide, permette di ottenere invece:

```
> cutree(tree, k=3, h=NULL)
             Piemonte          Valle_d_Aosta               Liguria
                    1                      2                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      2                     1
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    1                      1                     1
                Umbria                Marche                 Lazio
                    3                      3                     1
               Abruzzo                Molise              Campania
                    3                      3                     3
                Puglia            Basilicata              Calabria
                    3                      3                     3
               Sicilia              Sardegna
                    3                      3
```

G_1 = {Lombardia, Emilia-Romagna, Liguria, Toscana, Piemonte, Lazio, Veneto, Friuli Venezia-Giulia}

G_2 = {Trentino Alto-Adige, Valle d'Aosta}

G_3= {Umbria, Marche, Campania, Abruzzo, Sardegna, Calabria, Sicilia, Basilicata, Molise, Puglia}

Metodo della mediana

```
> cutree(tree, k=3, h=NULL)
             Piemonte          Valle_d_Aosta               Liguria
                    1                      2                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      2                     1
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    1                      1                     1
               Umbria                 Marche                 Lazio
                    3                      3                     1
             Abruzzo                 Molise              Campania
                    3                      3                     3
              Puglia             Basilicata              Calabria
                    3                      3                     3
             Sicilia               Sardegna
                    3                      3
```

G_1 = {Lombardia, Emilia-Romagna, Liguria, Toscana, Piemonte, Lazio, Veneto, Friuli Venezia-Giulia}

G_2 = {Trentino Alto-Adige, Valle d'Aosta}

G_3= {Umbria, Marche, Campania, Abruzzo, Sardegna, Calabria, Sicilia, Basilicata, Molise, Puglia}

È possibile osservare quindi che le partizioni ottenute col metodo del legame medio, del centroide e della mediana sono uguali.

1.5.6.3 Misure di sintesi associate ai cluster

In R è inoltre possibile ricavare misure di sintesi (ad esempio, la media campionaria, la varianza campionaria, la deviazione standard, …) sui singoli cluster, ottenuti tagliando il dendrogramma tramite la funzione **cutree()**, utilizzando la funzione **aggregate()** nel seguente modo:

```
> aggregate(X, by, FUN)
```

dove

- **X** rappresenta una matrice numerica o un data frame;
- **by** è una lista di indici sulla base dei quali le colonne di X vanno aggregate;
- **FUN** è la funzione da applicare alle colonne di X, separatamente per i vari gruppi individuati in base a **by**.

L'output della funzione **aggregate()** è una struttura contenente i valori ottenuti applicando la funzione **FUN** (ad esempio, *la media campionaria, la varianza campionaria, la deviazione standard*, ...) ad ognuna delle caratteristiche associate ai diversi cluster che sono stati aggregati. Tramite le seguenti linee di codice, applicati ai diversi partizionamenti, vengono calcolate le **misure di sintesi**:

Per rappresentare graficamente i cluster ottenuti con la funzione **cutree()**, si usa la seguente linea di codice:

```
taglio <- cutree (tree, k=3, h=NULL)
tagliolist <- list(taglio)
aggregate (tipiDiPasto, tagliolist, mean)
aggregate (tipiDiPasto, tagliolist, var)
aggregate (tipiDiPasto, tagliolist, sd)
```

Metodo del legame singolo

1. Media

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	83.2334	42.60	67.2778	10.00	3.54445	2.889	9.20	62.70	26.62
2	84.50	38.90	67.80	11.50	6.80	1.30	6.50	75.10	13.20
3	80.48	43.46	82.44	4.22	1.30	0.83	6.03	75.28	13.74

2. Varianza

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	6.5375	13.1525	9.9967	2.1050	1.32278	0.8837	1.7775	14.5525	10.7612
2	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	17.44622	17.8938	10.3694	1.1996	0.35556	0.1112	2.6179	24.964	12.5915

3. Deviazione standard

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	2.5569	3.6267	3.1618	1.4509	1.15021	0.9400	1.3333	3.81477	3.28054
2	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	4.1769	4.2302	3.2202	1.0953	0.59629	0.3335	1.61799	4.9963	3.54845

Metodo del legame completo

1. Media

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	84.0166	44.1333	67.65	9.65	3.116667	3.1666	9.83333	60.7	28.4667
2	82.375	39.375	66.85	10.9	5	2.075	7.575	68.8	20.5
3	80.48	43.46	82.44	4.22	1.3	0.8300	6.03	75.28	13.74

2. Varianza

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	6.08166	12.20266	11.9750	2.335	1.216667	0.7306	1.23866	5.86	3.0666
2	5.6225	0.729166	6.27	1.1466	1.7666667	0.9425	0.7825	22.68	26.86
3	17.44622	17.89377	10.3693	1.1995	0.3555556	0.1112	2.61788	24.964	12.591

3. Deviazione Standard

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	2.466104	3.493231	3.4604	1.52807	1.1232394	0.85479	1.112954	2.42074	1.75119
2	2.371181	0.853912	2.5039	1.07082	1.3291601	0.97082	0.88459	4.76235	5.182663
3	4.176868	4.23010	3.2201	1.09524	0.5962848	0.3335	1.61798	4.99639	3.548458

Metodo del legame medio, del centroide e della mediana

1. Media

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	83.7625	43.0875	67.7625	9.8125	3.3375	2.825	9.3125	61.90	27.325
2	81.7500	38.80	65.60	11.50	6	2.350	7.4	72.10	17.100
3	80.4800	43.46	82.44	4.22	1.3	0.830	6.03	75.28	13.740

2. Varianza

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	4.59125	12.58696	9.00083	2.04410	1.07125	0.96785	1.90125	10.0485	7.2192
2	15.125	0.20000	9.68000	0.00000	1.28000	2.20500	1.62000	18	30.42
3	17.44622	17.89378	10.36933	1.19955	0.35555	0.11122	2.61788	24.964	12.5915

3. Deviazione Standard

Cluster	Colazione adeguata	Colazione con latte	Casa	Mensa	Ristorante	Bar	Lavoro	Pranzo	Cena
1	2.142720	3.5478112	3.001398	1.429723	1.0350121	0.983797	1.378858	3.16994	2.686
2	3.889087	0.1414214	3.111270	0.000000	1.1313708	1.484924	1.272792	4.24264	5.515
3	4.176868	4.2301038	3.220145	1.095242	0.5962848	0.333500	1.617989	4.99639	3.548

Come precedentemente detto, il metodo del legame medio, del centroide e della mediana producono lo stesso partizionamento e quindi anche gli stessi indici di sintesi uguali.

1.5.6.4 Misure di non omogeneità statistiche

Dopo aver effettuato il taglio, siamo interessati a calcolare le misure di non omogeneità statistica relative all'insieme totale di individui (tr T), ai singoli cluster ottenuti effettuando il taglio e alla somma delle loro misure di non omogeneità (tr S) e alla misura di omogeneità tra i cluster (tr B):

$$\text{tr}\, T = \text{tr}\, S + \text{tr}\, B,$$

o equivalentemente:

$$1 = \frac{\text{tr}\, S}{\text{tr}\, T} + \frac{\text{tr}\, B}{\text{tr}\, T}.$$

Poiché per ogni fissata matrice X dei dati si ha che la tr T è fissata, i cluster dovrebbero essere individuati in modo da *minimizzare la misura di non omogeneità statistica all'interno dei cluster* ***(within)*** *e massimizzare la misura di non omogeneità statistica tra i gruppi* ***(between)****.*

Quindi, se due differenti metodi gerarchici conducono a due diverse partizioni con lo stesso numero di cluster, occorre scegliere quella partizione con misura di non omogeneità statistica all'interno dei cluster (tr S) più piccola, che corrisponde a maggiore omogeneità interna.

Analizziamo quindi le misure di non omogeneità statistiche per i diversi metodi gerarchici.

Metodo del legame singolo

Calcoliamo ora le misure di non omogeneità statistica relative all'insieme totale $I = G_1 \cup G_2 \cup G_3$ e ai tre cluster G_1, G_2 e G_3 individuati con il metodo del legame singolo. Per l'insieme totale I si ha:

```
> n <- nrow(tipiDiPasto)
> trHI <- (n-1)*sum(apply(tipiDiPasto,2,var))
> trHI #visualizza la misura di non omogeneità totale
[1] 4379.978
```

La misura di non omogeneità statistica totale è quindi $\mathbf{tr}H_I = \mathbf{4379.978}$. Calcoliamo ora le misure di non omogeneità statistiche dei tre gruppi G_1, G_2 e G_3 seguendo due metodi diversi:

- il primo permette di definire le matrici dei dati relative ai gruppi e a partire da esse determinare le misure di non omogeneità statistiche;
- il secondo invece calcola le misure di non omogeneità statistiche dei gruppi usando le funzioni **cutree()** e **aggregate()** senza calcolare le matrici.

Useremo il secondo metodo. *Occorre notare che se un cluster contiene un solo elemento, la sua misura di non omogeneità è nulla*. Con le seguenti linee di codice, si ottengono i seguenti dati:

```
> d <- dist(tipiDiPasto, method="euclidean", diag=TRUE, upper=TRUE)
> tree <- hclust(d, method='single')
> taglio <- cutree(tree, k=3, h=NULL)
> num <- table(taglio)
> tagliolist <- list(taglio)
> agvar <- aggregate(tipiDiPasto, tagliolist, var)[,-1]
>
> #Primo cluster
> trH1 <- (num[[1]]-1)*sum(agvar[1,])
> trH1
[1] 488.7222
>
> #Secondo cluster
> trH2 <- 0
> trH2
[1] 0
>
> #Terzo cluster
> trH3 <- (num[[3]]-1)*sum(agvar[3,])
> trH3
[1] 787.942
>
> #Misura di non omogeneità statistica interna ai 3 gruppi
> trS <- trH1+trH2+trH3
> trS
[1] 1276.664
>
> #Misura di non omogeneità statistica tra cluster
> trB <- trHI-trH1-trH2-trH3
> trB
[1] 3103.314
>
> trS/trHI
[1] 0.2914773
>
> trB/trHI
[1] 0.7085227
```

Da cui si ricava che la misura di non omogeneità del primo gruppo è *488.7222*, del terzo gruppo è *787.942* e del secondo gruppo è 0, dal momento che è formato da un solo elemento.
La misura di non omogeneità statistica interna ai 3 gruppi è definita come:

$$trS = trH1 + trH2 + trH3$$

e si ottiene quindi:

$$trS = 488.722 + 0 + 787.942 = 1276.664$$

La misura di non omogeneità tra cluster between è definita come segue:

$$trB = trHI - trH1 - trH2 - trH3$$

ottenendo quindi:

$$trB = 4379.978 - 488.7220 - 0 – 787.942 = 3103.314$$

Tenendo conto che trT = trHI:

$$\frac{trS}{trT} = \frac{1276.664}{4379.978} = 0.2914773$$

$$\frac{trB}{trT} = \frac{3103.314}{4379.978} = 0.7085227$$

Dai dati emerge che la misura di non omogeneità all'interno dei gruppi è più piccola rispetto alla misura di non omogeneità tra cluster, quindi il numero di cluster scelti è ottimale.

Metodo del legame completo

Calcoliamo ora le misure di non omogeneità statistica relative all'insieme totale $I = G_1 \cup G_2 \cup G_3$ e ai tre cluster G_1, G_2 e G_3 individuati con il metodo del legame completo. La misura di non omogeneità statistica totale è $\mathbf{tr}\boldsymbol{H_I} = \mathbf{4379.978}$.
Tramite le linee di codice:

```
> d <- dist(tipiDiPasto, method="euclidean", diag=TRUE, upper=TRUE)
> tree <- hclust(d, method='complete')
> taglio <- cutree(tree, k=3, h=NULL)
> num <- table(taglio)
> tagliolist <- list(taglio)
> agvar <- aggregate(tipiDiPasto, tagliolist, var)[,-1]
>
> #Primo cluster
> trH1 <- (num[[1]]-1)*sum(agvar[1,])
> trH1
[1] 223.76
>
> #Secondo cluster
> trH2 <- (num[[2]]-1)*sum(agvar[2,])
> trH2
[1] 200.4
>
> #Terzo cluster
> trH3 <- (num[[3]]-1)*sum(agvar[3,])
> trH3
[1] 787.942
>
> #Misura di non omogeneità statistica interna ai 3 gruppi
> trS <- trH1+trH2+trH3
> trS
[1] 1212.102
>
> #Misura di non omogeneità statistica tra cluster
> trB <- trHI-trH1-trH2-trH3
> trB
[1] 3167.876
> trS/trHI
[1] 0.276737
> trB/trHI
[1] 0.723263
```

Da cui si ricava che la misura di non omogeneità del primo gruppo è *223.76*, del secondo gruppo è *200.4* e del terzo gruppo è *787.942*.
La misura di non omogeneità statistica interna ai 3 gruppi è definita come:

$$trS = trH1 + trH2 + trH3$$

e si ottiene quindi:

$$trS = 223.76 + 200.4 + 787.942 = 1212.102$$

La misura di non omogeneità tra cluster between è definita come segue:

$$trB = trHI - trH1 - trH2 - trH3$$

ottenendo quindi:

$$trB = 4379.978 - 223.76 - 200.4 - 787.942 = 3167.876$$

Tenendo conto che trT = trHI:

$$\frac{trS}{trT} = \frac{1212.102}{4379.978} = 0.276737$$

$$\frac{trB}{trT} = \frac{3167.876}{4379.978} = 0.723263$$

Dai dati emerge che la misura di non omogeneità all'interno dei gruppi è più piccola rispetto alla misura di non omogeneità tra cluster.

Metodo del legame medio, del centroide e della mediana

Calcoliamo ora le misure di non omogeneità statistica relative all'insieme totale $I = G_1 \cup G_2 \cup G_3$ e ai tre cluster G_1, G_2 e G_3 individuati con il metodo del legame medio, del centroide e della mediana. I tre metodi hanno prodotto lo stesso partizionamento, possiamo quindi osservare che il loro comportamento è uguale.
La misura di non omogeneità statistica totale è **trH_I = 4379.978**.
Tramite le linee di codice:

```
> d <- dist(tipiDiPasto, method="euclidean", diag=TRUE, upper=TRUE)
> tree <- hclust(d, method='average')
> taglio <- cutree(tree, k=3, h=NULL)
> num <- table(taglio)
> tagliolist <- list(taglio)
> agvar <- aggregate(tipiDiPasto, tagliolist, var)[,-1]
>
> #Primo cluster
> trH1 <- (num[[1]]-1)*sum(agvar[1,])
> trH1
[1] 346.0725
>
> #Secondo cluster
> trH2 <- (num[[2]]-1)*sum(agvar[2,])
> trH2
[1] 78.35
>
> #Terzo cluster
> trH3 <- (num[[3]]-1)*sum(agvar[3,])
> trH3
[1] 787.942
>
> #Misura di non omogeneità statistica interna ai 3 gruppi
> trS <- trH1+trH2+trH3
> trS
[1] 1212.364
>
> #Misura di non omogeneità statistica tra cluster
> trB <- trHI-trH1-trH2-trH3
> trB
[1] 3167.614
> trS/trHI
[1] 0.2767969
> trB/trHI
[1] 0.7232031
```

Da cui si ricava che la misura di non omogeneità del primo gruppo è *346.0725*, del secondo gruppo è *78.35* e del terzo gruppo è *787.942*.
La misura di non omogeneità statistica interna ai 3 gruppi è definita come:

$$trS = trH1 + trH2 + trH3$$

e si ottiene quindi:

$$trS = 346.0725 + 78.35 + 787.942 = 1212.364$$

La misura di non omogeneità tra cluster between è definita come segue:

$$trB = trHI - trH1 - trH2 - trH3$$

ottenendo quindi:

$$trB = 4379.978 - 346.0725 - 78.35 - 787.942 = 3167.614$$

Tenendo conto che trT = trHI:

$$\frac{trS}{trT} = \frac{1212.364}{4379.978} = 0.2767969$$

$$\frac{trB}{trT} = \frac{3167.614}{4379.978} = 0.7232031$$

Dai dati emerge che la misura di non omogeneità all'interno dei gruppi è più piccola rispetto alla misura di non omogeneità tra cluster.

Facendo un riassunto dei risultati, $\frac{trB}{trT}$ ha un valore sempre alto per tutti i metodi usati. Dal momento che il valore è sempre di almeno il 70% per tutti i metodi, il partizionamento in tre cluster è una soluzione accettabile.
Mettendo a confronto la misura di non omogeneità statistica tra i cluster ottenuta attraverso i diversi metodi, essa risulta essere più alta nel metodo del legame completo.
Quindi, a parità di cluster, tra i vari metodi gerarchici i risultati migliori sono dati dal metodo del legame completo, quindi sarà considerata la suddivisione in cluster ottenuta tramite questo metodo.

1.5.7 Metodi non gerarchici

L'obiettivo dei metodi non gerarchici è quello di ottenere un'unica partizione degli *n* individui di partenza in cluster. A differenza dei metodi gerarchici, in tali tecniche è consentito riallocare gli individui già classificati ad un livello precedente dell'analisi.
Gli algoritmi di tipo non gerarchico procedono, data una prima partizione, a riallocare gli individui nel gruppo con centroide più vicino, fino a che per nessun individuo si verifica che sia minima la distanza rispetto al centroide di un gruppo diverso da quello a cui esso appartiene.
Il metodo più utilizzato prende il nome di **k-means**. Tale metodo richiede che il numero di cluster sia specificato a priori e fornisce in output un'unica partizione. Esso consiste dei passi descritti nel seguente algoritmo:

- **Step 1**: Fissare a priori il numero *k* di cluster specificando *m* punti di riferimento iniziali (scegliendo in maniera opportuna alcuni individui, o unità, o prendendo la configurazione determinata con una tecnica gerarchica) 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 baricentro (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 raggiungere una configurazione stabile, ossia gli individui all'interno di ogni cluster non cambiano al ripetersi del procedimento.

Nel metodo **k-means**, per garantire la convergenza della procedura iterativa, come misura di distanza tra i vettori delle caratteristiche e i centroidi viene utilizzata la **distanza euclidea** e, come per il metodo del centroide, si considera *la matrice contenente i quadrati delle distanze euclidee*.

I *vantaggi* del metodo k-means sono la velocità di esecuzione dei calcoli e l'estrema libertà che viene lasciata agli individui di raggrupparsi e allontanarsi.
Uno *svantaggio* è invece che la classificazione finale può essere influenzata dalla scelta iniziale dei *k* vettori delle caratteristiche come punti di riferimento.
Infatti, l'algoritmo potrebbe convergere ad un ottimo locale e non globale, il che significa che se si inizia con un diverso insieme di punti di riferimento si può giungere ad una differente partizione finale.

L'analisi con il metodo *k-mean*s si effettua in R mediante la funzione

kmeans (X, centers, iter.max = N, nstart = M)

dove:

- **X** è la matrice dei dati;

- **centers** è il numero dei cluster che si vogliono identificare o un vettore di lunghezza pari al numero di cluster contenente un insieme di centroidi iniziali dei cluster. Nel primo caso, ossia se è numero intero, l'algoritmo sceglie casualmente i punti di riferimento e tale insieme è utilizzato per individuare la partizione iniziale. Nel secondo caso, i centroidi iniziali possono essere derivati effettuando preliminarmente un'analisi di tipo gerarchico con il metodo del centroide.

- **iter.max** è il massimo numero di iterazioni permesse. Di default iter.max = 10.

- **nstart** fornisce il numero di volte in cui ripetere la procedura di scelta casuale dei punti di riferimento, nel caso in cui centers è il numero. Di default nstart = 1. Se nstart > 1, l'algoritmo riporta sempre come risultato la partizione con una misura di non omogeneità statistica totale all'interno dei cluster minima.

Si nota che nell'algoritmo *k-means* non occorre calcolare la *matrice iniziale delle distanze* (o dei quadrati delle distanze) così come invece si richiede nei metodi gerarchici.
Con l'analisi di tipo gerarchico si è visto che il numero ottimale di cluster è 3.
Si può usare questa informazione per la scelta del numero da considerare all'inizio dell'analisi con il metodo k-means.

Si considerano 3 diverse scelte iniziali:

1. Scelta casuale dei punti di riferimento
2. Ripartizione della procedura di scelta casuale dei punti di riferimento
3. Scelta dei centroidi come punti di riferimento

(1) Scelta casuale dei punti di riferimento

Le seguenti linee di codice mostrano l'applicazione del metodo non gerarchico k-means considerando una suddivisione in tre cluster ed effettuando una sola scelta casuale dei punti di riferimento con un numero massimo di iterazioni pari a 10.

```
> km <- kmeans(tipiDiPasto, center=3, iter.max=10, nstart=1)
> km
K-means clustering with 3 clusters of sizes 5, 8, 7

Cluster means:
  colazione_adeguata colazione_con_latte     casa    mensa ristorante      bar
1           84.04000            42.74000 71.30000 8.500000   4.100000 1.920000
2           78.92500            42.22500 83.55000 4.000000   1.075000 0.700000
3           83.82857            43.62857 67.54286 9.885714   3.228571 2.914286
    lavoro   pranzo     cena
1 7.560000 70.10000 18.52000
2 5.700000 76.72500 12.68750
3 9.457143 61.12857 28.01429

Clustering vector:
             Piemonte         Valle_d_Aosta               Liguria
                    3                     1                     3
            Lombardia   Trentino_Alto_Adige                Veneto
                    3                     1                     1
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                    3                     3                     3
               Umbria               Marche                 Lazio
                    1                     1                     3
             Abruzzo               Molise              Campania
                    2                     2                     2
              Puglia           Basilicata              Calabria
                    2                     2                     2
             Sicilia             Sardegna
                    2                     2

Within cluster sum of squares by cluster:
[1] 474.0320 434.3887 264.2171
 (between_SS / total_SS =  73.2 %)

Available components:

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

Individuando quindi tre cluster:

G_1 = *{Trentino Alto-Adige, Valle d'Aosta, Umbria, Marche, Veneto}*

G_2 = *{Campania, Abruzzo, Basilicata, Molise, Sardegna, Calabria, Sicilia, Puglia}*

G_3 = *{Lombardia, Emilia-Romagna, Liguria, Toscana, Piemonte, Lazio, Friuli Venezia-Giulia}*

I valori 474.0320, 434.3887, 264.2171 rappresentano le misure di non omogeneità statistiche associate ai cluster G_1, G_2 e G_3.
La funzione **str(km**) permette di ottenere una lista di informazioni relative all'oggetto *km* creato col metodo *kmeans*(), che si ottengono considerando le seguenti linee di codice:

```
> km$cluster
             Piemonte         Valle_d_Aosta               Liguria
                    3                     1                     3
            Lombardia   Trentino_Alto_Adige                Veneto
                    3                     1                     1
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                    3                     3                     3
               Umbria                Marche                 Lazio
                    1                     1                     3
             Abruzzo                Molise              Campania
                    2                     2                     2
              Puglia            Basilicata              Calabria
                    2                     2                     2
             Sicilia              Sardegna
                    2                     2
> km$centers
  colazione_adeguata colazione_con_latte     casa    mensa ristorante      bar
1           84.04000            42.74000 71.30000 8.500000   4.100000 1.920000
2           78.92500            42.22500 83.55000 4.000000   1.075000 0.700000
3           83.82857            43.62857 67.54286 9.885714   3.228571 2.914286
    lavoro   pranzo     cena
1 7.560000 70.10000 18.52000
2 5.700000 76.72500 12.68750
3 9.457143 61.12857 28.01429
> km$totss
[1] 4379.978
> km$tot.withinss
[1] 1172.638
> km$betweenss
[1] 3207.34
> km$size
[1] 5 8 7
```

Da notare che la misura di non omogeneità totale è 4379.978, la somma delle misure di non omogeneità interne ai cluster è 1172.638 (within) e la misura di non omogeneità tra cluster è 3207.34 (between). Mentre il rapporto

$$\frac{trB}{trT} = \frac{3207.34}{4379.978} = 0.732273$$

(2) Ripetizione della procedura di scelta casuale dei punti di riferimento

Ora è applicato il metodo non gerarchico k-means, così che l'algoritmo di aggregazione venga ripetuto otto volte in corrispondenza di otto ripetizioni della procedura di scelta casuale dei punti di riferimento con un numero di iterazioni massimo pari a 10.

```
> kmeans(tipiDiPasto, center=3, iter.max=10, nstart=8)
K-means clustering with 3 clusters of sizes 9, 8, 3

Cluster means:
  colazione_adeguata colazione_con_latte     casa     mensa ristorante
1           83.23333            42.60000 67.27778 10.000000   3.544444
2           78.92500            42.22500 83.55000  4.000000   1.075000
3           85.96667            45.23333 74.60000  7.233333   3.733333
       bar   lavoro   pranzo     cena
1 2.888889 9.200000 62.70000 26.62222
2 0.700000 5.700000 76.72500 12.68750
3 1.333333 7.066667 71.36667 16.36667

Clustering vector:
             Piemonte         Valle_d_Aosta               Liguria
                    1                     1                     1
            Lombardia   Trentino_Alto_Adige               Veneto
                    1                     3                     1
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                    1                     1                     1
               Umbria               Marche                 Lazio
                    3                     3                     1
             Abruzzo               Molise              Campania
                    2                     2                     2
              Puglia           Basilicata              Calabria
                    2                     2                     2
             Sicilia             Sardegna
                    2                     2

Within cluster sum of squares by cluster:
[1] 488.7222 434.3887 220.3133
 (between_SS / total_SS =  73.9 %)

Available components:

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

Individuando quindi tre cluster:

G_1 = {Valle d'Aosta, Veneto, Piemonte, Liguria, Lombardia, Friuli Venezia-Giulia, Lazio, Emilia-Romagna, Toscana}

G_2 = {Campania, Abruzzo, Basilicata, Molise, Sardegna, Calabria, Sicilia, Puglia}

G_3 = {Trentino Alto-Adige, Umbria, Marche}

I valori 488.7222, 434.3887 e 220.3133 rappresentano le misure di non omogeneità statistica associate ai cluster. Analizzando l'oggetto appena ottenuto si ha:

```
> km$cluster
             Piemonte         Valle_d_Aosta               Liguria
                    3                     3                     3
            Lombardia   Trentino_Alto_Adige                Veneto
                    3                     1                     3
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                    3                     3                     3
               Umbria                Marche                 Lazio
                    1                     1                     3
             Abruzzo                Molise              Campania
                    2                     2                     2
              Puglia            Basilicata              Calabria
                    2                     2                     2
             Sicilia              Sardegna
                    2                     2
> km$centers
  colazione_adeguata colazione_con_latte     casa     mensa ristorante      bar
1           85.96667            45.23333 74.60000  7.233333   3.733333 1.333333
2           78.92500            42.22500 83.55000  4.000000   1.075000 0.700000
3           83.23333            42.60000 67.27778 10.000000   3.544444 2.888889
    lavoro   pranzo     cena
1 7.066667 71.36667 16.36667
2 5.700000 76.72500 12.68750
3 9.200000 62.70000 26.62222
> km$totss
[1] 4379.978
> km$tot.withinss
[1] 1143.424
> km$betweenss
[1] 3236.554
> km$size
[1] 3 8 9
```

La misura di non omogeneità totale è 4379.978, la somma delle misure di non omogeneità interne ai cluster è di 1143.424 (within) e la misura di non omogeneità tra i cluster è 3236.554 (between). Mentre il rapporto risulta essere

$$\frac{trB}{trT} = \frac{3236.554}{4379.978} = 0.738943$$

I valori ottenuti con una ripetizione della procedura di scelta casuale dei punti di riferimento sono diversi, anche se di poco, rispetto a quelli ottenuti con una sola scelta casuale dei punti di riferimento.
In generale, l'output della funzione riporta la partizione per la quale la somma delle misure di non omogeneità statistica all'interno dei gruppi (within) è la più piccola tra le partizioni finali ottenute a partire dalle procedure ricavate dai diversi punti di riferimento scelti casualmente.

In generale non esiste una regola per determinare il numero ottimale di ripetizioni della procedura di scelta casuale dei punti di riferimento per ottenere un risultato stabile.
Empiricamente, si potrebbe provare con diversi valori crescenti di *nstart* fino a che il risultato non cambia.

(3) Scelta dei centroidi come punti di riferimento

In alternativa alla scelta casuale dei punti di riferimento, si possono usare i centroidi dei tre cluster ottenuti con la tecnica del *gerarchica del centroide* usando la funzione **aggregate**():

```
> d <- dist(tipiDiPasto, method="euclidean", diag=TRUE, upper=TRUE)
> d2 <- d^2
> hl <- hclust(d2, method='centroid')
> taglio <- cutree(hl, k=3, h=NULL)
> tagliolist <- list(taglio)
> centroidiIniziali <- aggregate(tipiDiPasto, tagliolist, mean)[,-1]
> centroidiIniziali
  colazione_adeguata colazione_con_latte    casa   mensa ristorante   bar lavoro
1            83.7625             43.0875 67.7625  9.8125     3.3375 2.825 9.3125
2            81.7500             38.8000 65.6000 11.5000     6.0000 2.350 7.4000
3            80.4800             43.4600 82.4400  4.2200     1.3000 0.830 6.0300
  pranzo   cena
1  61.90 27.325
2  72.10 17.100
3  75.28 13.740
```

Per motivi computazionali si elimina la prima colonna della matrice dei centroidi ottenuta tramite la funzione **aggregate()**, che si riferisce alle etichette dei cluster.
Usando questi centroidi si può ora applicare il *k-means*:

```
> km <- kmeans(tipiDiPasto, centers=centroidiIniziali, iter.max=10)
> km
K-means clustering with 3 clusters of sizes 7, 3, 10

Cluster means:
  colazione_adeguata colazione_con_latte     casa     mensa ristorante      bar
1           83.82857            43.62857 67.54286  9.885714   3.228571 2.914286
2           82.26667            38.96667 66.83333 10.766667   5.366667 2.300000
3           80.48000            43.46000 82.44000  4.220000   1.300000 0.830000
    lavoro   pranzo     cena
1 9.457143 61.12857 28.01429
2 7.700000 70.50000 18.90000
3 6.030000 75.28000 13.74000

Clustering vector:
             Piemonte         Valle_d_Aosta               Liguria
                    1                     2                     1
            Lombardia   Trentino_Alto_Adige                Veneto
                    1                     2                     2
Friuli_Venezia_Giulia        Emilia_Romagna               Toscana
                    1                     1                     1
               Umbria                Marche                 Lazio
                    3                     3                     1
             Abruzzo                Molise              Campania
                    3                     3                     3
              Puglia            Basilicata              Calabria
                    3                     3                     3
             Sicilia              Sardegna
                    3                     3

Within cluster sum of squares by cluster:
[1] 264.2171 130.2333 787.9420
 (between_SS / total_SS =  73.0 %)

Available components:

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

La suddivisione ottenuta risulta essere:

G_1 = {Piemonte, Liguria, Lombardia, Friuli Venezia-Giulia, Lazio, Emilia-Romagna, Toscana}

G_2 = {Trentino Alto-Adige, Veneto, Valle d'Aosta}

G_3 = {Campania, Abruzzo, Basilicata, Molise, Sardegna, Calabria, Sicilia, Puglia, Umbria, Marche}

Analizzando nel dettaglio l'oggetto ottenuto:

```
> km$cluster
             Piemonte          Valle_d_Aosta               Liguria
                    1                      2                     1
            Lombardia    Trentino_Alto_Adige                Veneto
                    1                      2                     2
Friuli_Venezia_Giulia         Emilia_Romagna               Toscana
                    1                      1                     1
               Umbria                 Marche                 Lazio
                    3                      3                     1
              Abruzzo                 Molise               Campania
                    3                      3                     3
               Puglia             Basilicata               Calabria
                    3                      3                     3
              Sicilia              Sardegna
                    3                      3
> km$centers
  colazione_adeguata colazione_con_latte     casa     mensa ristorante      bar
1           83.82857            43.62857 67.54286  9.885714   3.228571 2.914286
2           82.26667            38.96667 66.83333 10.766667   5.366667 2.300000
3           80.48000            43.46000 82.44000  4.220000   1.300000 0.830000
    lavoro   pranzo     cena
1 9.457143 61.12857 28.01429
2 7.700000 70.50000 18.90000
3 6.030000 75.28000 13.74000
> km$totss
[1] 4379.978
> km$tot.withinss
[1] 1182.392
> km$betweenss
[1] 3197.586
> km$size
[1]  7  3 10
```

La misura di non omogeneità totale è 4379.978, la somma delle misure di non omogeneità interne ai cluster è di 1182.392 (within) e la misura di non omogeneità tra i cluster è 3197.586 (between). Mentre il rapporto risulta essere

$$\frac{trB}{trT} = \frac{3197.586}{4379.978} = 0.730046$$

In conclusione, si può affermare che l'applicazione delle diverse procedure porta a risultati diversi, anche se di poco e i risultati più alti sono ottenuti attraverso la ripetizione della procedura di scelta casuale dei punti di riferimento.
Il partizionamento *migliore* risulta essere proprio quello con *ripetizione della procedura di scelta casuale dei punti di riferimento*, poiché la misura di non omogeneità interna ai cluster (within) è la più piccola rispetto agli altri due metodi e di conseguenza la misura di non omogeneità tra cluster (between) è ancora più grande.

Capitolo 2

Seconda parte

Di particolare importanza in statistica è *l'inferenza statistica*. Essa ha lo scopo di estendere le misure ricavate dall'esame di un campione alla popolazione da cui il campione è stato estratto. Uno dei problemi centrali dell'inferenza statistica è il seguente: si desidera studiare una popolazione descritta da una variabile aleatoria osservabile X la cui funzione di distribuzione ha una forma nota ma contiene un parametro $\vartheta \in \Theta$ non noto (o più parametri non noti).
Il termine *osservabile* significa che si possono osservare i valori assunti dalla variabile aleatoria X (ad esempio, eseguendo un esperimento casuale) e quindi il parametro non noto è presente soltanto nella legge di probabilità (funzione di distribuzione, funzione di probabilità, densità di probabilità). Ovviamente se ϑ è noto la legge di probabilità è completamente specificata.
Per ottenere informazioni sul parametro non noto ϑ della popolazione, si può fare uso dell'inferenza statistica considerando un campione estratto dalla popolazione ed effettuando su tale campione delle opportune misure. Affinché le conclusioni dell'inferenza statistica siano valide il campione deve essere scelto in modo tale da essere rappresentativo della popolazione.
L'inferenza statistica si basa su due metodi fondamentali di indagine:

- La **stima dei parametri** che ha lo scopo di determinare i valori non noti dei parametri di una popolazione (come il valore medio, la varianza, ...) per mezzo dei corrispondenti parametri derivati dal campione estratto dalla popolazione (come la media campionaria, la varianza campionaria, ...). Si possono usare *stime puntuali* o *stime per intervallo*. Si parla di *stima puntuale* quando si stima un parametro non noto di una popolazione usando un singolo valore reale. Alla stima puntuale di un parametro non noto di una popolazione (costituita da un unico valore) spesso si preferisce sostituire un intervallo di valori, detto *intervallo di confidenza*, ossia si cerca di determinare in base al campione osservato (x_1, x_2, ..., x_n) due limiti (uno inferiore e uno superiore) entro i quali sia compreso il parametro non noto con un certo grado di confidenza, detto anche *grado di fiducia*.

- La **verifica delle ipotesi** è un procedimento che consiste nel fare una congettura o un'ipotesi sul parametro non noto ϑ o sulla distribuzione di probabilità e nel decidere, sulla base del campione estratto se essa è accettabile.

Per affrontare i problemi di stima (puntuale o per intervallo) dei parametri e della verifica delle ipotesi statistiche possono essere usate le variabili aleatorie discrete o continue. Sarà poi trattata la stima puntuale e per intervallo e affronteremo alcuni problemi di verifica di ipotesi statistiche utilizzando una variabile aleatoria **continua** con una funzione di distribuzione **normale**.

Variabili aleatorie continue con R

Il sistema R mette a disposizione per ciascuna delle principali distribuzioni di probabilità continue:

• la funzione densità di probabilità;
• la funzione di distribuzione;
• le funzioni quantili;
• la funzione che simula tale variabile aleatoria mediante la generazione di numeri pseudocasuali;

Tutte queste funzioni utilizzano nomi che iniziano con una particolare lettera dell'alfabeto, in modo da indicare il tipo di funzione a cui fa riferimento, seguita dal nome della distribuzione teorica scelta. La particolare lettera dell'alfabeto può essere:

- *d* calcola la densità di probabilità di una variabile aleatoria in uno specifico punto o in un insieme di punti;
- *p* calcola la funzione di distribuzione di una variabile aleatoria in uno specifico punto o in un insieme di punti;
- *q* calcola la funzioni quantili;
- *r* calcola la funzione che simula una variabile aleatoria mediante la generazione di numeri pseudocasuali.

2.1 Distribuzione normale

La funzione di distribuzione normale, detta anche di Gauss o gaussiana, riveste estrema importanza nel calcolo delle probabilità e nella statistica anche in quanto essa costituisce una distribuzione limite alla quale tendono varie altre funzioni di distribuzioni sotto opportune ipotesi.

Una variabile aleatoria X di densità di probabilità

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \qquad x \in \mathbb{R} \qquad (\mu \in \mathbb{R}, \sigma > 0)$$

si dice avere distribuzione normale di parametri μ e σ.

Dalla relazione appena vista si evince che per ogni x ∈ R risulta $f_X(\mu - x) = f_X(\mu + x)$; pertanto la densità normale è simmetrica rispetto all'asse $x = \mu$. Il grafico di $f_X(x)$ esibisce una caratteristica forma a campana, simmetrica rispetto a $x = \mu$. La notazione $X \sim N(\mu, \sigma)$ sarà utilizzata per indicare che X ha distribuzione normale di parametri μ e σ, cioè che X è una variabile **normale**.

In R la densità normale si calcola con la funzione

dnorm(x, mean= mu, sd = sigma)

dove
- **x** è il valore assunto (o i valori assunti) dalla variabile aleatoria normale;
- **mean** e **sd** sono il valore medio e la deviazione standard della densità normale.

Il seguente codice permette di visualizzare la densità di $X \sim N(\mu, 1)$ con μ = −3,−2,−1, 0, 1, 2, 3.

```
> curve(dnorm(x, mean=-3, sd=1), from=-6, to=6, xlab="x", ylab="y", main="mu = -3,-2,-1, 0, 1, 2, 3; sigma=1")
> curve(dnorm(x, mean=-2, sd=1), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
> curve(dnorm(x, mean=-1, sd=1), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
> curve(dnorm(x, mean=0, sd=1), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
> curve(dnorm(x, mean=1, sd=1), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
> curve(dnorm(x, mean=2, sd=1), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
> curve(dnorm(x, mean=3, sd=1), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
```

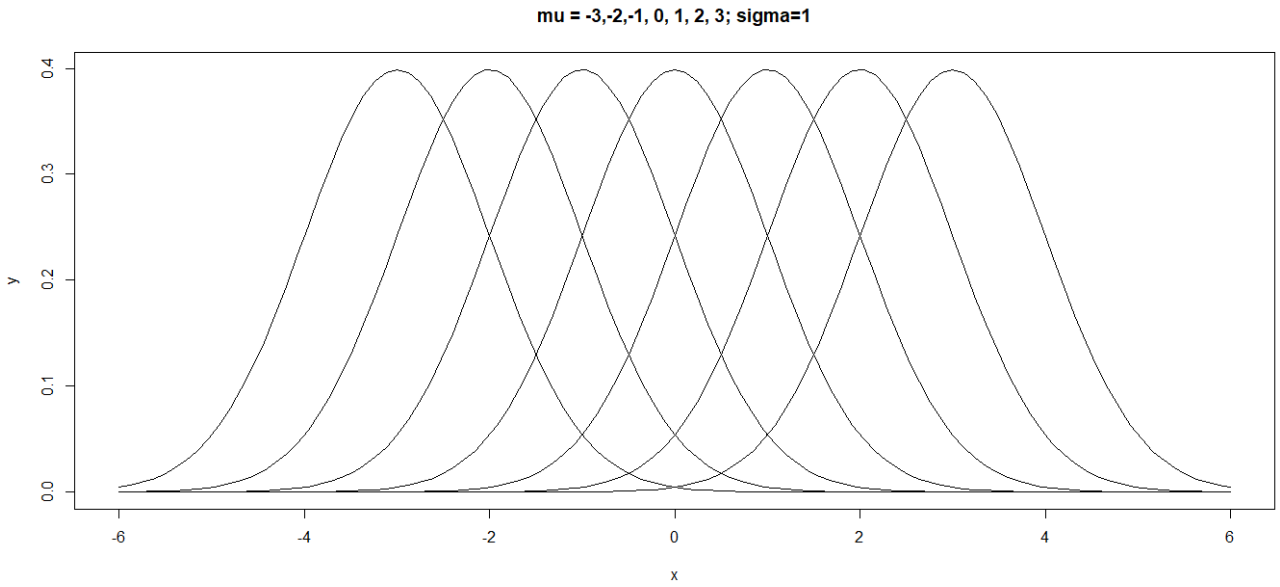


Figura 2.1: Densità normale al variare di μ = −3,−2,−1, 0, 1, 2, 3 (da sinistra verso destra)

Da come si evince nel grafico in figura 2.1 le variazioni del parametro μ comportano traslazioni della curva lungo l'asse delle ascisse; infatti, al crescere del parametro μ la curva si sposta lungo l'asse delle ascisse senza cambiare forma.

Il parametro σ, pari alla semi ampiezza tra i due punti di flesso, caratterizza la larghezza della funzione. Siccome l'ordinata massima è inversamente proporzionale a σ, al crescere di σ questa decresce, mentre l'area sottesa dalla densità deve rimanere unitaria.

Tramite le seguenti linee di codice si visualizza la densità $X \sim N(0, \sigma)$ con σ = 0.5, 1, 1.5.

```
> curve(dnorm(x, mean=0, sd=0.5), from=-4, to=4, xlab="x", ylab="y", main="mu = 0; sigma = 0.5,1,1.5")
> curve(dnorm(x, mean=0, sd=1), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
> curve(dnorm(x, mean=0, sd=1.5), from=-6, to=6, xlab="x", ylab="y", add=TRUE)
```

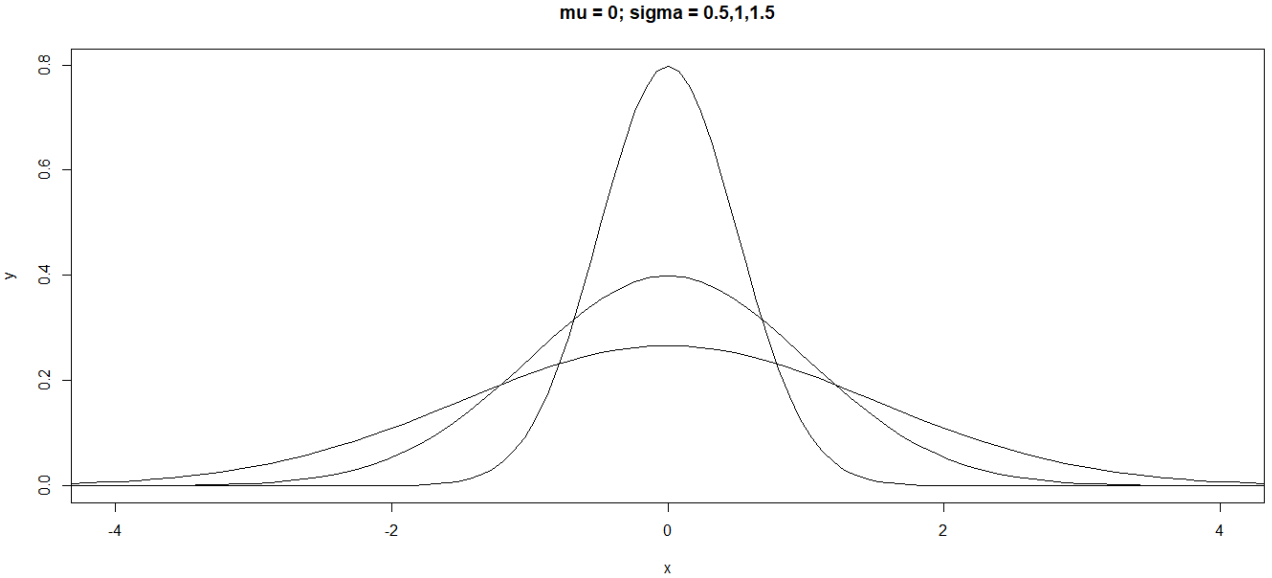


Figura 2.2: Densità normale al variare di σ = 0.5, 0, 1.5 (dall'alto verso il baso in prossimità dell'origine)

dal grafico in Figura 2.2 è possibile che al crescere di σ la curva diventa sempre più piatta, mentre al decrescere di σ essa si allunga verso l'alto restringendosi contemporaneamente ai lati.

La funzione di distribuzione di una variabile aleatoria $X \sim N(\mu, \sigma)$

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(y)\,dy = \Phi\left(\frac{x-\mu}{\sigma}\right) \qquad x \in \mathbb{R}$$

è la funzione di distribuzione di una variabile aleatoria $Z \sim N(0, 1)$, detta *normale standard*. Pertanto, se $X \sim N(\mu, \sigma)$ si ha:

$$P(a < X < b) = F_X(b) - F_X(a) = \Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right).$$

In R la funzione di distribuzione di una variabile $X \sim N(\mu, \sigma)$ si calcola tramite la funzione:

pnorm(x, mean= mu, sd = sigma, lower.tail = TRUE)

dove:

- **x** è il valore assunto (o i valori assunti) dalla variabile aleatoria normale;
- **mean** e **sd** sono il valore medio e la deviazione standard della densità normale;
- **lower.tail** se tale parametro è **TRUE** (caso di default) calcola $P(X \leq x)$, mentre se tale parametro è **FALSE** calcola $P(X > x)$.

Il seguente codice permette di visualizzare la funzione di distribuzione di $X \sim N(0, \sigma)$ con $\sigma = 0.5$, 1, 1.5:

```
> curve(pnorm(x, mean=0, sd=0.5), from=-4, to=4, xlab="x", ylab=expression(P(X<=x)),
+ main="mu = 0; sigma = 0.5,1,1.5", lty=2)
> text(-0.4,0.8,"sigma=0.5")
>
> curve(pnorm(x, mean=0, sd=1), add=TRUE)
> arrows(-1,0.1,0.5,0.2,code=1, length=0.10)
> text(0.8,0.2,"sigma=1")
>
> curve(pnorm(x, mean=0, sd=1.5), add=TRUE, lty=3)
> text(-2.2,0.2,"sigma=1.5")
```

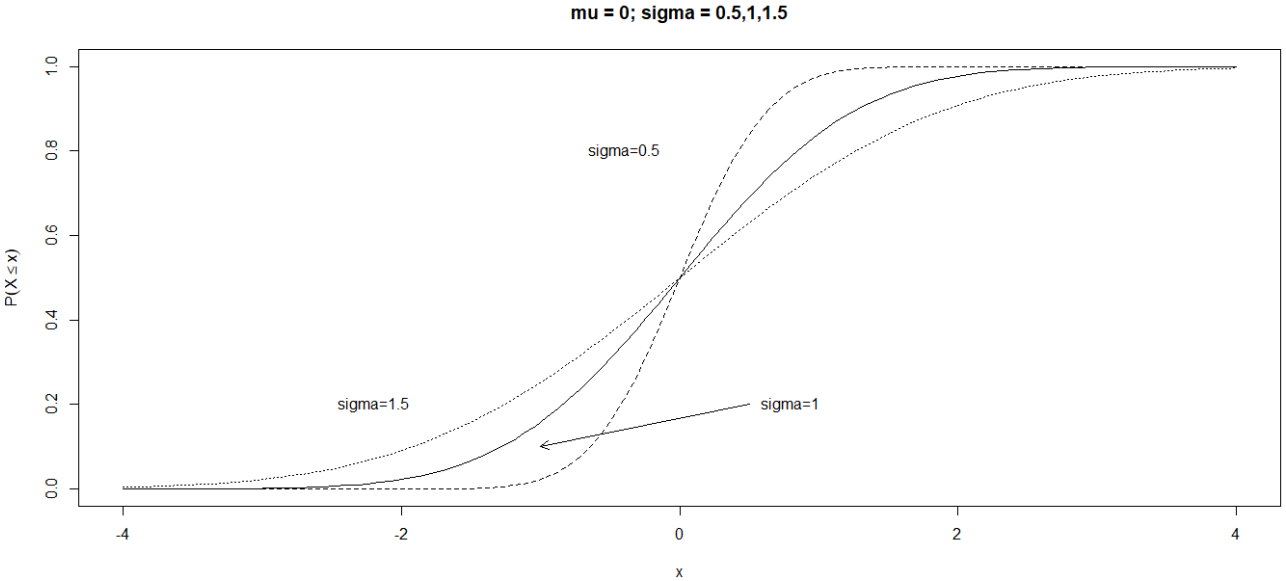


Figura 2.3: Funzione di distribuzione normale al variare di $\sigma = 0.5$, 0, 1.5

La funzione **arrows()** ha come argomenti le due coordinate della linea della freccia, il parametro **code** può assumere i valori 1, 2, 3 a seconda se la freccia deve essere unidirezionale verso sinistra,

unidirezionale verso destra oppure bidirezionale; il parametro **length** fornisce invece la grandezza della freccia.

Regola del 3σ

Per una qualsiasi variabile aleatoria normale $X \sim N(\mu, \sigma)$ risulta

$$P(\mu-3\sigma < X < \mu+3\sigma) = P\left(-3 < \frac{X-\mu}{\sigma} < 3\right) = P(-3 < Z < 3) = 0.9973002.$$

Quindi la probabilità che una variabile aleatoria $X \sim N(\mu, \sigma)$ assuma valori in un intervallo avente come centro μ e semiampiezza 3σ è prossima all'unità.
Questa proprietà delle variabili aleatorie normali è nota come *regola del 3σ*.

In R, per una variabile aleatoria normale $Z \sim N(0, 1)$ si ha:

```
> pnorm(3, mean=0, sd=1) - pnorm(-3, mean=0, sd=1)
[1] 0.9973002
```

La regola del 3σ permette di individuare l'intervallo $(\mu-3\sigma, \mu+3\sigma)$ in cui rappresentare la funzione densità di una variabile normale di valore medio μ e varianza σ^2 in maniera tale che l'area sottesa dalla curva sia circa unitaria e l'area delle code destra e sinistra sia trascurabile.

In R si calcolano anche i quantili (percentili) della distribuzione normale tramite la funzione:

qnorm(z, mean = mu, sd = sigma, lower.tail = TRUE)

dove

- **z** è il valore assunto (o i valori assunti) dalle probabilità relative al percentile $z \cdot 100$–esimo;
- **mean** e **sd** sono il valore medio e la deviazione standard della densità normale.
- **lower.tail** se **TRUE** (default) calcola $P(X \leq x)$, mentre se tale parametro è **FALSE** calcola $P(X > x)$.

Il risultato della funzione è il percentile $z \cdot 100$–esimo, ossia il più piccolo numero x assunto dalla variabile aleatoria normale X tale che sussista che $P(X \leq x) \geq z$. Ad esempio, se si considera una variabile normale standard $Z \sim N(0, 1)$, le seguenti linee di codice forniscono i quartili Q_0, Q_1, Q_2, Q_3, Q_4 .

```
> z <- c(0, 0.25, 0.5, 0.75, 1)
> qnorm(z, mean=0, sd=1)
[1]       -Inf -0.6744898  0.0000000  0.6744898        Inf
```

che mostra che il primo quartile (25–esimo percentile) è $Q_1 = -0.6744898$, il secondo quartile o mediana (50–esimo percentile) è $Q_2 = 0$ e il terzo quartile (75–esimo percentile) è $Q_3 = 0.6744898$ (per la simmetria intorno all'origine della densità normale standard). Il minimo è $Q_0 = -\infty$ e il massimo è $Q_4 = \infty$.

In R si può simulare la variabile aleatoria normale generando una sequenza di numeri pseudocasuali mediante la funzione

rnorm(N, mean= mu, sd= sigma)

Dove:

- **N** è la lunghezza della sequenza da generare;
- **mean** e **sd** sono il valore medio e la deviazione standard della densità normale;

Il codice seguente

```
> par(mfrow=c(2,2))
> curve(dnorm(x, mean=2, sd=1), from=-2, to=6, xlab="x", ylab="f(x)",
+ ylim=c(0,0.5), main="Densità normale mu=2; sigma=1")
>
> sim1 <- rnorm(500, mean=2, sd=1)
> hist(sim1, freq=F, xlim=c(-2,6), ylim=c(0,0.5), breaks=100, xlab="x",
+ ylab="Istogramma", main="Densità simulata, N=500")
>
> sim2 <- rnorm(5000, mean=2, sd=1)
> hist(sim2, freq=F, xlim=c(-2,6), ylim=c(0,0.5), breaks=100, xlab="x",
+ ylab="Istogramma", main="Densità simulata, N=5000")
>
> sim3 <- rnorm(50000, mean=2, sd=1)
> hist(sim3, freq=F, xlim=c(-2,6), ylim=c(0,0.5), breaks=100, xlab="x",
+ ylab="Istogramma", main="Densità simulata, N=50000")
```

Produce il grafico in Figura 2.4, che permette di confrontare la densità normale teorica con $\sigma = 1$ e $\mu = 2$ con la densità simulata scegliendo $N = 500, 5000, 50000$. All'aumentare del numero di simulazioni l'istogramma delle frequenze relative si avvicina sempre di più alla densità esponenziale teorica.

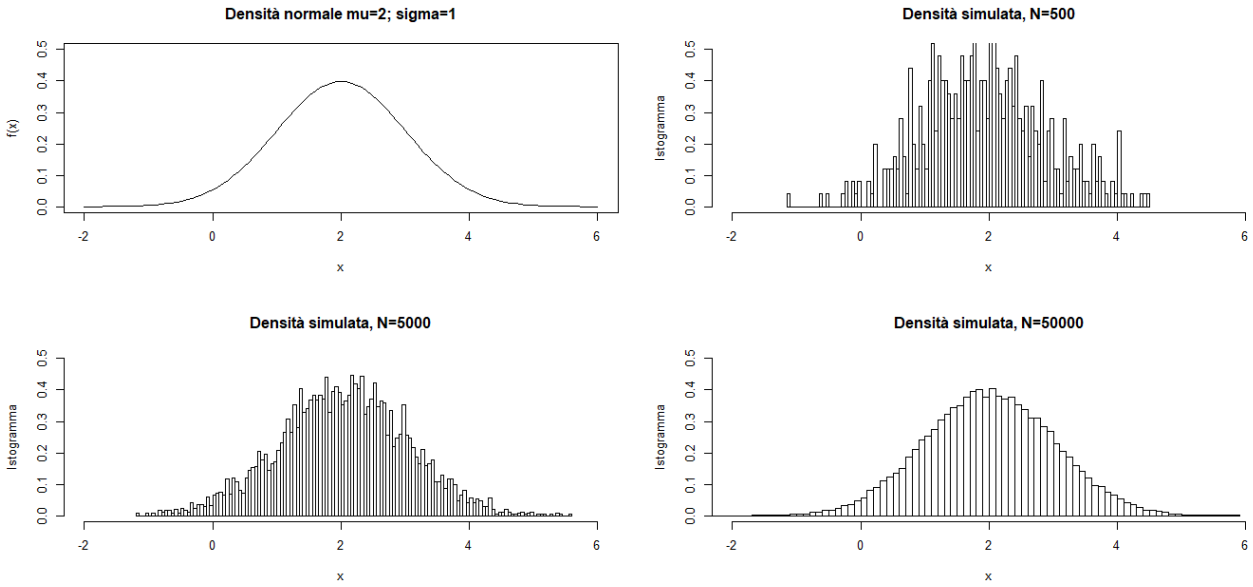


Figura 2.4: Confronto densità normale con $\sigma = 1$ e $\mu = 2$ con la densità simulata

2.1.1 Approssimazione della distribuzione binomiale con la distribuzione normale

Il calcolo delle **probabilità binomiali** diventa rapidamente oneroso al crescere di *n*. È quindi utile ricercare delle formule approssimate in grado di rendere agevole tale calcolo e, al contempo, accettabile l'errore derivante dall'approssimazione.

Saranno presi in considerazione:

- Teorema di De Moivre-Laplace
- Teorema centrale della convergenza

Teorema di De Moivre-Laplace *Sia $X_1, X_2, \ldots$ una successione di variabili aleatorie indipendenti distribuite alla Bernoulli con parametro p ($0 < p < 1$), e sia $Y_n = X_1 + X_2 + \ldots + X_n$. Allora per ogni $x \in R$ risulta:*

$$\lim_{n\to+\infty} P\left(\frac{Y_n - np}{\sqrt{np(1-p)}} \le x\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2}\, dy,$$

Ossia

$$\frac{Y_n - np}{\sqrt{np(1-p)}} \xrightarrow{d} Z,$$

converge in distribuzione alla variabile aleatoria Z normale standard.

Ricordiamo che se $X_1, X_2, \ldots$ sono variabili aleatorie indipendenti di Bernoulli di parametro *p*, allora

$$Y_n = X_1 + X_2 + \ldots + X_n$$

è una variabile aleatoria binomiale di valore medio np e varianza np(1−p).

Il teorema mostra che sottraendo a Y_n la sua media *np* e dividendo la differenza per la deviazione standard $\sqrt{n\,p\,(1\,-\,p)}$, si ottiene una variabile aleatoria standardizzata la cui funzione di distribuzione è per *n* grande approssimativamente normale standard.

La bontà dell'approssimazione dipende da *n* e da *p* e migliora al tendere di *p* a 1/2. In generale si suole assumere che l'approssimazione sia soddisfacente per $n > 10$ e per $5/n < p < 1 - 5/n$.
Esaminiamo ora l'approssimazione della binomiale alla normale

$$Y_n \simeq np + \sqrt{np(1-p)}\, Z,$$

Al variare di *n* con *p* fissato.
Il secondo membro è una variabile aleatoria con densità normale di valore medio *np* e varianza *np(1−p)*. Il seguente codice confronta la densità normale di valore medio *np* e varianza *np(1−p)* e la funzione di probabilità binomiale per $n = 25, 50, 75, 100$ e $p = 0.2$ il cui grafico è riportato in Figura 2.5. Si evince che l'approssimazione migliora al crescere di *n*.

```
> par(mfrow=c(2,2))
> p <- 0.2
> q <- 1-p
> x <- 0:25
> n <- 25
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=25 p=0.2")
> lines(x, dbinom(x, n, 0.2), type="h")
>
> x <- 0:50
> n <- 50
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=50 p=0.2")
> lines(x, dbinom(x, n, 0.2), type="h")
>
> x <- 0:75
> n <- 75
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=75 p=0.2")
> lines(x, dbinom(x, n, 0.2), type="h")
>
> x <- 0:100
> n <- 100
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=100 p=0.2")
> lines(x, dbinom(x, n, 0.2), type="h")
```

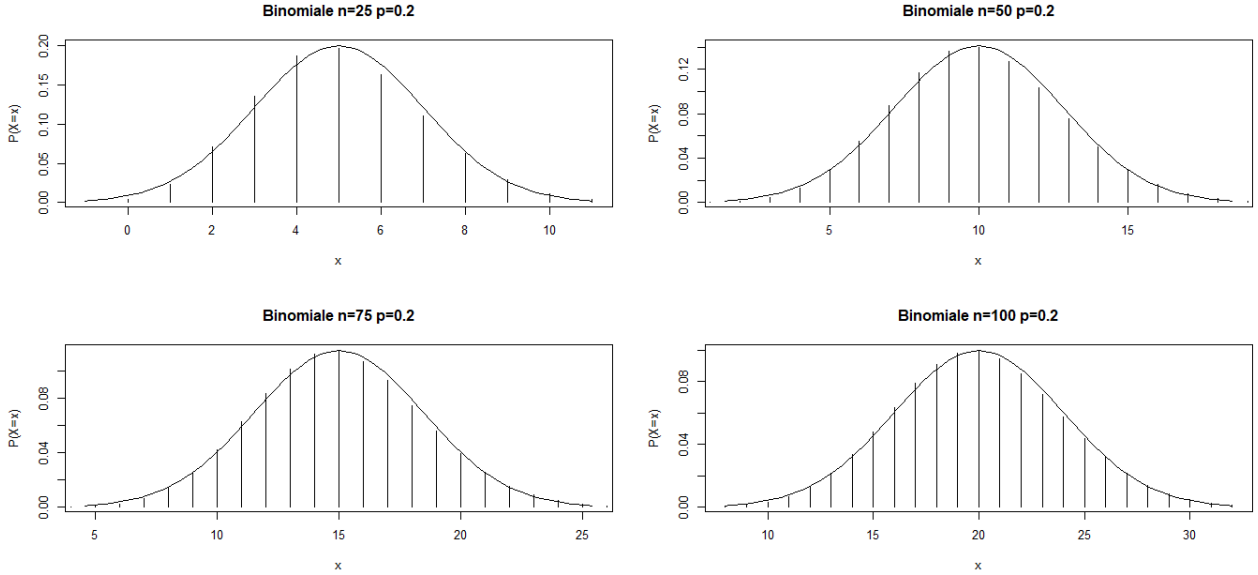


Figura 2.5: Confronto della probabilità binomiale della variabile $Y_n \sim B(n, 0.2)$ con la densità normale di valor medio $\mu = np$ e deviazione standard $\sigma = \sqrt{np\ (1\ -\ p)}$ per varie scelte di n.

Viene ora esaminata l'approssimazione $Y_n \sim np + \sqrt{np\ (1\ -\ p)}\ Z$ della binomiale alla normale al variare di p con n fissato.
Il seguente codice confronta la densità normale di valore medio np e varianza $np(1 - p)$ e la funzione di probabilità binomiale per $n = 20$ e $p = 0.125, 0.25, 0.375, 0.5$, il cui grafico è riportato in Figura 2.6. Si nota che l'approssimazione non è buona per piccoli valori di p e migliora al tendere di p a 1/2, diventando poi eccellente quando $p = 1/2$.

```
> par(mfrow=c(2,2))
> p <- 0.125
> q <- 1-p
> x <- 0:20
> n <- 20
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=20 p=0.125")
> lines(x, dbinom(x, n, 0.125), type="h")
>
> p <- 0.25
> q <- 1-p
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=20 p=0.25")
> lines(x, dbinom(x, n, 0.25), type="h")
>
> p <- 0.375
> q <- 1-p
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=20 p=0.375")
> lines(x, dbinom(x, n, 0.375), type="h")
>
> p <- 0.5
> q <- 1-p
> curve(dnorm(x, n*p, sqrt(n*p*q)), from=n*p-3*sqrt(n*p*q), to=n*p+3*sqrt(n*p*q),
+ xlab="x", ylab="P(X=x)", main="Binomiale n=20 p=0.5")
> lines(x, dbinom(x, n, 0.5), type="h")
```

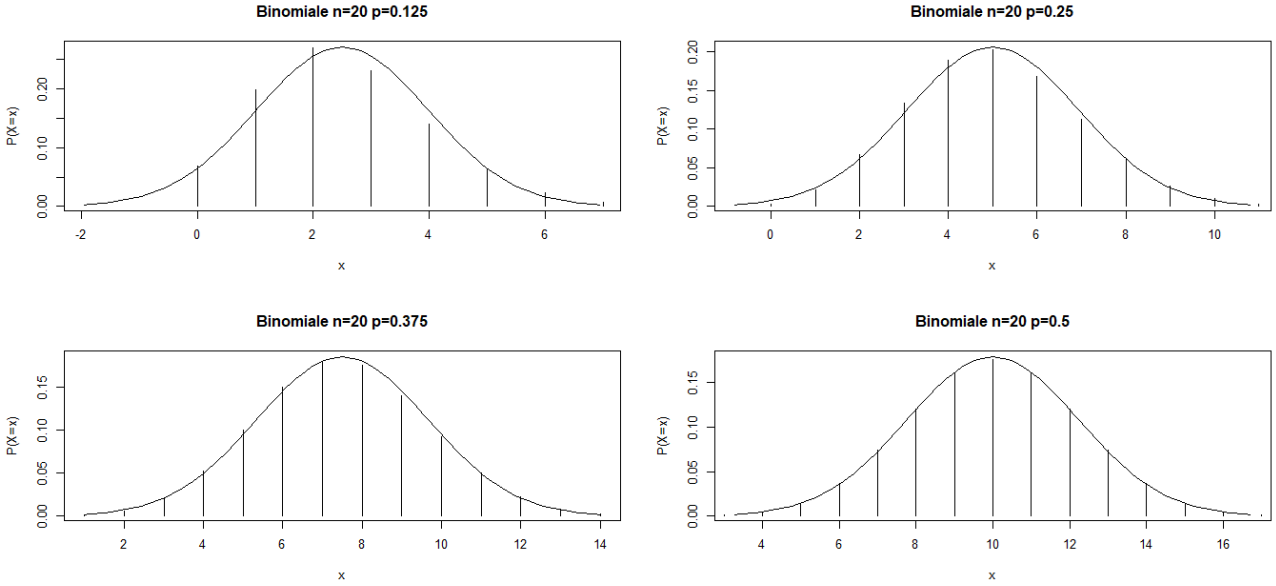


Figura 2.6: Confronto della probabilità binomiale della variabile $Y_n \sim B(20, p)$ con la densità normale di valor medio $\mu = np$ e deviazione standard $\sigma = np(1 - p)$ per varie scelte di p.

Teorema centrale di convergenza

Introduciamo ora uno dei più importanti risultati della teoria della probabilità, noto quale *teorema centrale di convergenza o teorema centrale del limite*, che fornisce una semplice ed utile approssimazione della distribuzione della somma di variabili aleatorie indipendenti, evidenziando al contempo la grande importanza della distribuzione normale nella statistica inferenziale.

Sia $X_1, X_2, \ldots$ una successione di variabili aleatorie, definite nello stesso spazio di probabilità, indipendenti e identicamente distribuite con valore medio μ finito e varianza σ^2 finita e positiva. Posto per ogni intero n positivo $Y_n = X_1 + X_2, \ldots + X_n$, per ogni $x \in R$ risulta:

$$\lim_{n\to+\infty} P\left(\frac{Y_n - n\mu}{\sigma\sqrt{n}} \le x\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2}\, dy = \Phi(x),$$

ossia la successione delle variabili aleatorie standardizzate

$$\frac{Y_n - E(Y_n)}{\sqrt{Var(Y_n)}} = \frac{Y_n - n\mu}{\sqrt{n}\sigma} \xrightarrow{d} Z,$$

converge in distribuzione alla variabile aleatoria normale standard.

Il Teorema mostra inoltre che sottraendo a $X_1 + X_2, \ldots + X_n$ la sua media $n\mu$ e dividendo la differenza per la deviazione standard di Y_n, ossia per $\sigma\sqrt{n}$, si ottiene una variabile aleatoria standardizzata la cui funzione di distribuzione è per n sufficientemente grande approssimativamente normale standard. Quindi, per n grande la distribuzione della media campionaria

$$Y_n = X_1 + X_2 + \ldots + X_n$$

È approssimativamente normale con valore medio $n\mu$ e varianza $n\sigma^2$.
Ovviamente, la bontà delle approssimazioni dipende da n e dal tipo di distribuzione delle variabili $X_1 + X_2, \ldots + X_n$. L'approssimazione migliora al crescere di n e nelle applicazioni spesso si verifica che essa è già soddisfacente per $n \geq 30$.

2.1.2 Approssimazione della distribuzione di Poisson con la distribuzione normale

Se, ad esempio, supponiamo che $X_1 + X_2, \ldots$ è una successione di variabili aleatorie indipendenti di Poisson di parametro λ allora $Y_n = X_1 + X_2, \ldots + X_n$ è ancora una variabile aleatoria di Poisson di parametro $n\lambda$. Quindi, il teorema centrale di convergenza afferma che per n grande la distribuzione di $Y_n = X_1 + X_2, \ldots + X_n$ è approssimativamente normale con valore medio $n\lambda$ e varianza $n\lambda$, ossia

$$Y_n \sim n\lambda + \sqrt{n\lambda}\,Z.$$

dove $n\lambda + \sqrt{n\lambda}\,Z$ è una variabile aleatoria con densità normale di valore medio $n\lambda$ e varianza $n\lambda$.
Esaminiamo ora l'approssimazione della distribuzione di Poisson di parametro $n\lambda$ alla normale di valore medio e varianza $n\lambda$ al variare del parametro $n\lambda$. Il seguente codice permette di visualizzare la Figura 2.7 in cui si confronta la probabilità di Poisson di parametro $n\lambda$ con la densità normale di valore medio e varianza $n\lambda = 5, 10, 25, 50$. Si nota che al crescere di $n\lambda$ aumenta l'accuratezza dell'approssimazione.

```
> par(mfrow=c(2,2))
> x <- 0:100
>
> curve(dnorm(x, 5, sqrt(5)), from=5-3*sqrt(5), to=5+3*sqrt(5),
+ xlab="x", ylab="P(X=x)", main="Poisson n lambda = 5")
> lines(x, dpois(x, 5), type="h")
>
> curve(dnorm(x, 10, sqrt(10)), from=10-3*sqrt(10), to=10+3*sqrt(10),
+ xlab="x", ylab="P(X=x)", main="Poisson n lambda = 10")
> lines(x, dpois(x, 10), type="h")
>
> curve(dnorm(x, 25, sqrt(25)), from=25-3*sqrt(25), to=25+3*sqrt(25),
+ xlab="x", ylab="P(X=x)", main="Poisson n lambda = 25")
> lines(x, dpois(x, 25), type="h")
>
> curve(dnorm(x, 50, sqrt(50)), from=50-3*sqrt(50), to=50+3*sqrt(50),
+ xlab="x", ylab="P(X=x)", main="Poisson n lambda = 50")
> lines(x, dpois(x, 50), type="h")
```

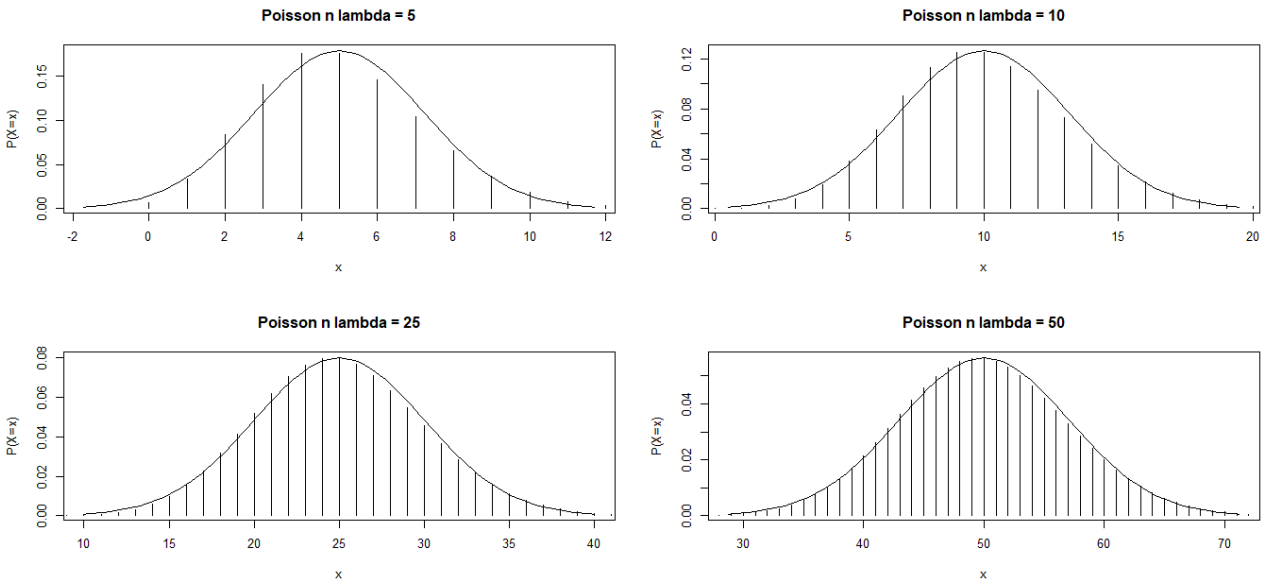


Figura 2.7: Confronto della probabilità di Poisson della variabile $X \sim P(n\lambda)$ con la densità normale di valor medio $\mu = n\lambda$ e deviazione standard $\sigma = \sqrt{n\lambda}$ per varie scelte di $n\lambda$.

2.2 Analisi di un campione normale

Viene ora generato un campione facente parte di una popolazione normale, per potervi effettuare analisi e stime relative ad una variabile aleatoria normale.
Per realizzarlo in R, sono sufficienti le seguenti istruzioni:

```
> campione <- rnorm(200, mean=4, sd=4)
```

Il campione mostrato sopra risulta avere N=200, valore medio μ = 4, varianza σ^2 = 16 e deviazione standard σ = 4.
Essendo stato generato in modo pseudocasuale, i valori di media campionaria, varianza campionaria e deviazione standard campionaria del campione sono i seguenti:

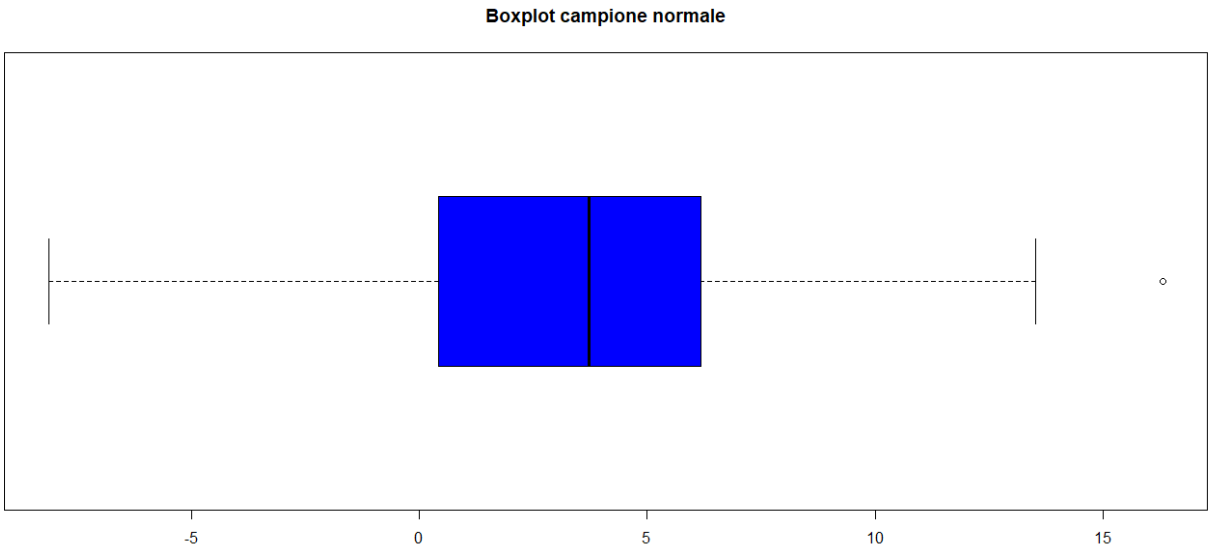
```
> mean(campione)
[1] 3.43899
> var(campione)
[1] 18.16173
> sd(campione)
[1] 4.261658
```

Possiamo anche calcolare la mediana e i quantili con le seguenti linee di codice:

```
> median(campione)
[1] 3.722701
> quantile(campione)
        0%        25%        50%        75%       100%
-8.1303741  0.4301937  3.7227010  6.1359029 16.3184037
```

Possiamo inoltre visualizzare il boxplot

```
> boxplot(campione, horizontal=T, main="Boxplot campione normale", col="blue")
```



Da cui è possibile vedere che il primo quantile (25-esimo percentile) è Q_1=0.4301937, il secondo quartile o mediana (50-esimo percentile) è Q_2=3.7227010 e il terzo quantile (75-esimo percentile) è Q_3=6.1359029 (per la simmetria intorno all'origine della densità normale).
Il minimo è Q_0 = -8.1303741 e il massimo è Q_4 = 16.3184037.

Osservando il boxplot, il campione risulta centrato con la mediana posta quasi in corrispondenza del valore 4. Si nota la presenza di un outlier. La distribuzione risulta essere abbastanza simmetrica da entrambi i lati. Questa ipotesi è verificata tramite il grafico in Figura 2.8 derivato dalle seguenti linee di codice:

```
> hist(campione, freq=F, xlim= c(-10,15), ylim=c(0,0.5), breaks=100,
+ xlab="x", ylab="Istogramma", main="Densità simulata N=200")
```

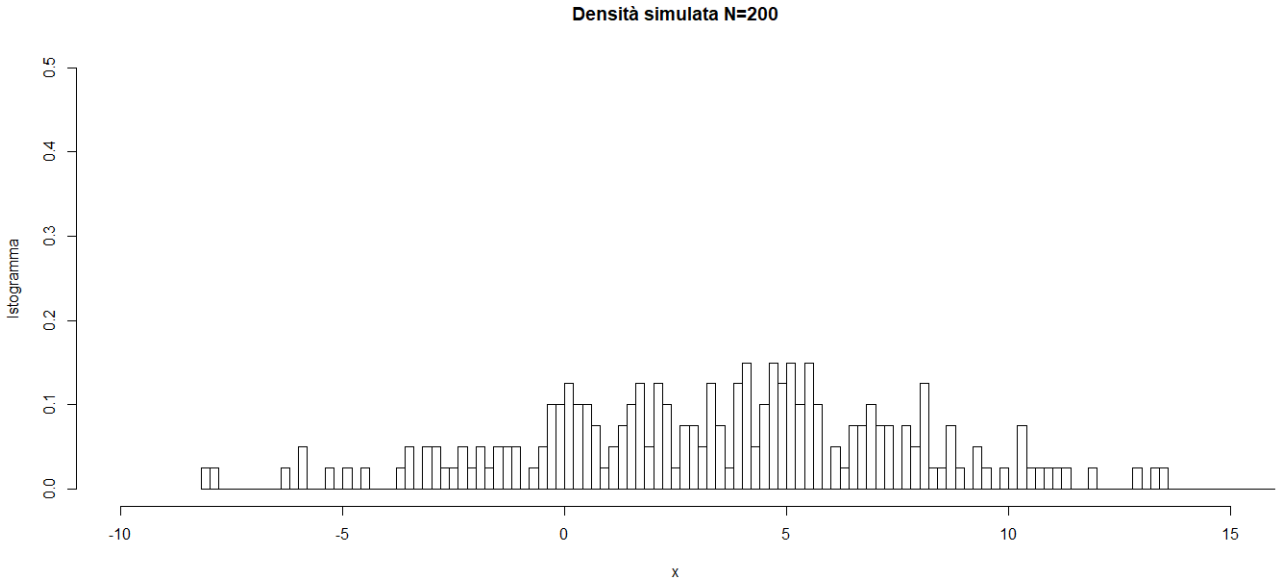


Figura 2.8: Grafico densità simulata N=200

2.3 Stima puntuale

2.3.1 Campioni casuali e stimatori

Uno dei problemi centrali dell'inferenza statistica è di *studiare una popolazione descritta da una variabile aleatoria osservabile X la cui funzione di distribuzione ha una forma nota ma contiene un parametro $\vartheta \in \theta$ non noto (o più parametri non noti)*.
Per ottenere informazioni sul parametro non noto ϑ della popolazione, si può fare uso dell'inferenza statistica considerando un campione estratto dalla popolazione ed effettuando su tale campione delle opportune misure. Affinché le conclusioni dell'inferenza statistica siano valide il campione deve essere scelto in modo tale da *essere rappresentativo della popolazione*. Molti metodi dell'inferenza statistica sono basati sull'ipotesi di campioni casuali.

Nei metodi di indagine dell'inferenza statistica si considera un campione casuale $X_1, X_2, ..., X_n$ di ampiezza *n* estratto dalla popolazione e si vuole ottenere informazioni circa il parametro non noto ϑ, facendo uso di alcune *variabili aleatorie*, che sono funzioni misurabili del campione casuale, dette **statistiche e stimatori**.

Uno **stimatore** $\widehat{\Theta} = t(X_1, X_2, ..., X_n)$ *è una funzione misurabile e osservabile del campione casuale* $X_1, X_2, ..., X_n$ *i cui valori possono essere usati per stimare un parametro non noto* ϑ *della popolazione. I valori* $\widehat{\vartheta}$ *assunti da tale stimatore sono detti stime del parametro non noto* ϑ.
Statistiche tipiche sono la media e la varianza campionarie.

Per le **statistiche** consideriamo $X_1, X_2, ..., X_n$ *un campione casuale. La statistica*

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

è detta media campionaria, mentre la statistica

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

è detta varianza campionaria.

Proposizione: *Sia* $X_1, X_2, ..., X_n$ *un campione casuale estratto da una popolazione descritta da una variabile aleatoria osservabile X caratterizzata da valore medio* $E(X) = \mu$ *finito e varianza* $Var(X) = \sigma^2$ *finita. Risulta:*

$$E(\overline{X}) = \mu, \qquad \mathrm{Var}\,(\overline{X}) = \frac{\sigma^2}{n}$$

Ciò è facilmente dimostrabile, in quanto si ha che per la proprietà di linearità del valore medio e l'identica distribuzione delle variabili aleatorie che costituiscono il campione, dalla precedente definizione di $\overline{X}$ si evince che:

$$E(\bar{X}) = E\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = \mu$$

Inoltre, poiché le variabili aleatorie che costituiscono il campione sono indipendenti e identicamente distribuite, dalla definizione di $\overline{X}$ si ottiene:

$$Var(\bar{X}) = Var\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i) = \frac{\sigma^2}{n}.$$

La Proposizione mostra che al crescere dell'ampiezza del campione la media campionaria fornisce una stima sempre più accurata del valore medio della popolazione.
Inoltre, dal teorema centrale di convergenza della probabilità scaturisce che per n sufficientemente grande (ossia per campioni di grande ampiezza) la funzione di distribuzione della media campionaria $\overline{X}$ è approssimativamente normale con valore medio μ e varianza σ^2/n.

2.3.2 Metodi per la ricerca di stimatori

Sia $X_1, X_2, ..., X_n$ un campione casuale estratto da una popolazione con funzione di probabilità (nel caso discreto) oppure densità di probabilità (nel caso assolutamente continuo) $f(x; \vartheta_1, \vartheta_2, ..., \vartheta_k)$ dove $\vartheta_1, \vartheta_2, ..., \vartheta_k$ denotano i parametri non noti della popolazione. Lo scopo del decisore, dopo aver osservato i valori assunti dal campione casuale, è quello di stimare i parametri non noti della popolazione. I principali metodi di stima puntuale dei parametri sono il *metodo dei momenti* e il *metodo della massima verosimiglianza*.

2.3.2.1 Metodo dei momenti

Il metodo dei momenti è uno dei più antichi metodi di stima dei parametri. Per illustrarlo occorre in primo luogo definire i **momenti campionari**.
Si definisce momento campionario r–esimo relativo ai valori osservati $x_1, x_2, ..., x_n$ del campione casuale il valore

$$M_r(x_1, x_2, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{n} x_i^r \qquad (r = 1, 2, \ldots)$$

Si nota quindi che il momento campionario r–esimo è la media aritmetica delle potenze r–esime delle n osservazioni effettuate sulla popolazione. In particolare, se $r = 1$ il momento campionario $M_1(x_1, x_2, ..., x_n)$ coincide con il valore osservato della media campionaria $\overline{X}$, ossia $M_1(x_1 + x_2 + ... + x_n) / n$.
Se esistono k parametri da stimare, il metodo dei momenti consiste nell'uguagliare i primi k momenti della popolazione in esame con i corrispondenti momenti del campione casuale. Quindi, se i primi k momenti esistono e sono finiti, tale metodo consiste nel risolvere il sistema di k equazioni

$$E(X^r) = M_r(x_1, x_2, \ldots, x_n) \qquad (r = 1, 2, \ldots, k)$$

Le incognite del sistema sono i parametri $\vartheta_1, \vartheta_2, ..., \vartheta_k$. Affinché il metodo dei momenti sia utilizzabile occorre che il sistema ammetta un'unica soluzione.
Le stime dei parametri ottenute con tale metodo dipendono dal campione osservato e quindi al variare dei possibili campioni osservati si ottengono gli stimatori $\widehat{\Theta}_1, \widehat{\Theta}_2, ..., \widehat{\Theta}_k$ dei parametri non noti della popolazione, detti *stimatori del metodo dei momenti*.
Applichiamo ora il metodo a una popolazione normale.

Popolazione normale
Si è interessati a determinare col metodo dei momenti gli stimatori dei parametri μ e σ^2 di una popolazione normale di densità di probabilità

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}. \qquad (x \in \mathbb{R},\ \mu \in \mathbb{R},\ \sigma > 0).$$

Occorre quindi stimare i parametri μ e σ^2. Poiché $E(X) = \mu$ e $E(X^2) = \mu + \sigma^2$ si ottiene un sistema di due equazioni (tante quante sono i parametri da stimare):

$$\begin{cases} E(X) = \bar{x} \\ E(X^2) = M_2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \end{cases} \Rightarrow \begin{cases} \hat{\mu} = \bar{x} \\ \sigma^2 + \mu^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \end{cases} \Rightarrow \begin{cases} \hat{\mu} = \bar{x} \\ \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \mu^2 \end{cases}$$

$$\Rightarrow \begin{cases} \hat{\mu} = \bar{x} \\ \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2 \end{cases} \Rightarrow \begin{cases} \hat{\mu} = \bar{x} \\ \sigma^2 = \frac{1}{n}\left[\sum_{i=1}^{n} x_i^2 - \bar{x}\sum_{i=1}^{n} x_i\right] \end{cases}$$

$$\Rightarrow \begin{cases} \hat{\mu} = \bar{x} \\ \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 (x_i - \bar{x})^2 \end{cases}$$

Il metodo dei momenti fornisce quindi come stimatore del valore medio μ la media campionaria $\overline{X}$ e come stimatore della varianza σ^2 la variabile aleatoria $(n-1)S^2/n$. Se consideriamo il campione ottenuto precedentemente, si ha che:

```
> stimamu <- mean(campione)
> stimamu
[1] 3.43899
> stimasigma2 <- (length(campione)-1)*var(campione)/length(campione)
> stimasigma2
[1] 18.07092
```

La stima del parametro μ col metodo dei momenti $\hat{\mu}$ = 3.43899 e la stima del parametro σ^2 col metodo dei momenti è $\hat{\sigma}^2$ = 18.07092.

2.3.2.2 Metodo della massima verosimiglianza

Il metodo della massima verosimiglianza è il più importante metodo per la stima dei parametri non noti di una popolazione e solitamente è preferito al metodo dei momenti. Per illustrare il metodo della massima verosimiglianza occorre introdurre in primo luogo la funzione di verosimiglianza.

Sia $X_1, X_2, ..., X_n$ un campione casuale di ampiezza n estratto dalla popolazione. La funzione di verosimiglianza $L(\vartheta_1, \vartheta_2, ..., \vartheta_k) = L(\vartheta_1, \vartheta_2, ..., \vartheta_k; x_1, x_2, ..., x_n)$ del campione osservato $(x_1, x_2, ...,x_n)$ è la funzione di probabilità congiunta (nel caso di popolazione discreta) oppure la funzione densità di probabilità congiunta (nel caso di popolazione assolutamente continua) del campione casuale $X_1, X_2, ..., X_n$ ossia

$$\begin{aligned} L(\vartheta_1, \vartheta_2, \ldots, \vartheta_k) &= L(\vartheta_1, \vartheta_2, \ldots, \vartheta_k; x_1, x_2, \ldots, x_n) \\ &= f(x_1; \vartheta_1, \vartheta_2, \ldots, \vartheta_k)\, f(x_2; \vartheta_1, \vartheta_2, \ldots, \vartheta_k) \cdots f(x_n; \vartheta_1, \vartheta_2, \ldots, \vartheta_k). \end{aligned}$$

Il metodo della massima verosimiglianza consiste nel massimizzare la funzione di verosimiglianza rispetto ai parametri $\vartheta_1, \vartheta_2, ..., \vartheta_k$. Tale metodo cerca quindi di determinare da quale funzione di probabilità congiunta (nel caso di popolazione discreta) oppure di densità di probabilità congiunta (nel caso di popolazione assolutamente continua) è *più verosimile* (è *più plausibile*) che provenga il campione osservato $(x_1, x_2, ..., x_n)$.

I valori di $\vartheta_1, \vartheta_2, ..., \vartheta_k$ che massimizzano la funzione di verosimiglianza sono indicati con $\hat{\vartheta}_1, \hat{\vartheta}_2, \ldots, \hat{\vartheta}_k$; essi costituiscono le stime di massima verosimiglianza dei parametri non noti $\vartheta_1, \vartheta_2, ..., \vartheta_k$ della popolazione. Tali stime dipendono dal campione osservato $(x_1, x_2, ..., x_n)$ e quindi al variare dei possibili campioni osservati si ottengono gli stimatori di massima verosimiglianza $\widehat{\Theta}_1, \widehat{\Theta}_2, \ldots, \widehat{\Theta}_k$ dei parametri non noti $\vartheta_1, \vartheta_2, ..., \vartheta_k$ della popolazione, detti *stimatori di massima verosimiglianza*.

Popolazione normale
Si è interessati a determinare lo stimatore di massima verosimiglianza dei parametri μ e σ^2 di una popolazione normale.
Le stime di massima verosimiglianza dei parametri μ e σ^2 sono rispettivamente

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \widehat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu})^2.$$

Lo stimatore di massima verosimiglianza di μ è la media campionaria $\overline{X}$. Invece lo stimatore di σ^2 è $(n-1)\,S^2/n$. Entrambi gli stimatori coincidono con quelli ottenuti con il metodo dei momenti. In relazione al campione precedentemente ottenuto, la stima del parametro μ col metodo della massima verosimiglianza è $\hat{\mu} = 3.43899$ e la stima del parametro σ^2 col metodo della massima verosimiglianza è $\hat{\sigma}^2 = 18.07092$.

2.3.3 Proprietà degli stimatori

In generale esistono molti stimatori che possono essere utilizzati per stimare il parametro non noto di una popolazione. Occorre quindi definire delle proprietà di cui può o meno godere uno stimatore. Alcune di queste proprietà sono:

- corretto (o equivalentemente non distorto),
- più efficiente di un altro,
- corretto e con varianza uniformemente minima,
- asintoticamente corretto,
- consistente.

Stimatore corretto
Uno stimatore $\hat{\Theta} = t(X_1, X_2, ..., X_n)$ del parametro non noto ϑ della popolazione è detto **corretto** *(non distorto) se e solo se per ogni $\vartheta \in \theta$ si ha:*

$$E(\hat{\Theta}) = \vartheta,$$

ossia se il valore medio dello stimatore $\hat{\Theta}$ è uguale al corrispondente parametro non noto della popolazione.

Stimatore asintoticamente corretto
Uno stimatore $\hat{\Theta}_n = t(X_1, X_2, ..., X_n)$ del parametro non noto ϑ della popolazione è detto **asintoticamente corretto** *se e solo se per ogni $\vartheta \in \theta$ si ha*

$$\lim_{n \to +\infty} E(\hat{\Theta}_n) = \vartheta,$$

ossia se il valore medio dello stimatore $\hat{\Theta}_n$ tende al crescere dell'ampiezza del campione casuale al corrispondente parametro non noto della popolazione.

Stimatore consistente
Uno stimatore $\hat{\Theta}_n = t(X_1, X_2, ..., X_n)$ del parametro non noto ϑ della popolazione è detto consistente se e solo se per ogni $\varepsilon > 0$ si ha

$$\lim_{n \to +\infty} P\left(|\hat{\Theta}_n - \vartheta| < \varepsilon\right) = 1 \qquad \forall \vartheta \in \Theta,$$

ossia se e solo se $\hat{\Theta}_n$ converge in probabilità a ϑ.

X	Metodo dei momenti	Metodo della massima verosimiglianza	Proprietà degli stimatori
Normale $E(X) = \mu$ $Var(X) = \sigma^2$	(*) $\overline{X}$ (**) $(n-1)S^2/n$	$\overline{X}$ $(n-1)S^2/n$	(*) Stimatore corretto con varianza minima e consistente per μ (**) Stimatore asintoticamente corretto e consistente per σ^2

Figura 2.9: Proprietà degli stimatori per la normale

Precisiamo che possono esistere diversi stimatori corretti di un parametro non noto di una popolazione, si deve quindi definire un criterio per confrontare più stimatori dello stesso parametro. Uno dei modi per fare ciò è tramite l'**errore quadratico medio** che fornisce una misura di quanto si discosta lo stimatore $\hat{\theta}$ dal parametro non noto ϑ della popolazione.

Errore quadratico medio
Sia $\hat{\theta} = t(X_1, X_2, ..., X_n)$ uno stimatore del parametro non noto ϑ della popolazione. Si chiama errore quadratico medio la quantità

$$MSE(\hat{\Theta}) = E\left[(\hat{\Theta} - \vartheta)^2\right].$$

Che può essere scritto anche in termini di varianza

$$MSE(\hat{\Theta}) = Var(\hat{\Theta}) + E[\hat{\Theta} - \vartheta]^2$$

Il principale problema del decisore consiste nello scegliere lo stimatore migliore del parametro ϑ, ossia lo stimatore che ha il più piccolo errore quadratico medio per ogni valore ammissibile di $\vartheta \in \theta$.
La ricerca dello stimatore con errore quadratico uniformemente minimo deve essere quindi effettuata in opportune classi come, ad esempio, nella *classe degli stimatori corretti*.

2.4 Intervalli di confidenza

Alla stima puntuale di un parametro non noto di una popolazione (costituita da un singolo valore reale) spesso si preferisce sostituire un intervallo di valori, detto *intervallo di confidenza* (o intervallo di fiducia), ossia si cerca di determinare in base ai dati del campione, due limiti (uno inferiore ed uno superiore) entro i quali sia compreso il parametro non noto con un certo *coefficiente di confidenza* (detto anche *grado di fiducia*).
Sia $X_1, X_2, ..., X_n$ un campione casuale di ampiezza n estratto da una popolazione con funzione di probabilità (nel caso discreto) oppure densità di probabilità (nel caso assolutamente continuo) $f(x;\vartheta)$, dove ϑ denota il parametro non noto della popolazione. Denotiamo con $\underline{C}_n = g_1(X_1, X_2, ..., X_n)$ e con $\overline{C}_n = g_2(X_1, X_2, ..., X_n)$ due statistiche (funzioni osservabili del campione casuale) che soddisfino la condizione $\underline{C}_n < \overline{C}_n$, cioè che godono della proprietà che per ogni possibile fissato campione osservato $x = (x_1, x_2, ..., x_n)$ risulti $g_1(x) < g_2(x)$.

Intervallo di confidenza
Fissato un coefficiente di confidenza $1 - \alpha$ ($0 < \alpha < 1$), se è possibile scegliere le statistiche $\underline{C}_n$ e $\overline{C}_n$ in modo tale che

$$P(\underline{C}_n < \vartheta < \overline{C}_n) = 1 - \alpha,$$

allora si dice che $(\underline{C}_n, \overline{C}_n)$ è un intervallo di confidenza (intervallo di fiducia) di grado $1 - \alpha$ per ϑ. Inoltre, le statistiche $\underline{C}_n$ e $\overline{C}_n$ sono dette limite inferiore e superiore dell'intervallo di confidenza.

Se $g_1(x)$ e $g_2(x)$ sono i valori assunti dalle statistiche $\underline{C}_n$ e $\overline{C}_n$ per il campione osservato $x = (x_1, x_2, ..., x_n)$, allora l'intervallo $(g_1(x), g_2(x))$ è detto *stima dell'intervallo di confidenza* di grado $1 - \alpha$ per ϑ ed i punti finali $g_1(x)$ e $g_2(x)$ di tale intervallo sono detti rispettivamente *stima del limite inferiore e stima del limite superiore dell'intervallo di confidenza*.

In generale esistono numerosi intervalli di confidenza dello stesso grado $1 - \alpha$ per un parametro non noto ϑ della popolazione. La scelta dell'intervallo di confidenza deve essere effettuata in base

ad alcune proprietà statistiche. Ad esempio, fissato un coefficiente di confidenza $1 - \alpha$, alcune proprietà desiderabili sono che la *lunghezza dell'intervallo di confidenza*

$$L(X_1, X_2, \ldots, X_n; 1-\alpha) = \overline{C}_n - \underline{C}_n$$

sia la più piccola possibile oppure che la lunghezza media di tale intervallo sia la più piccola possibile.

2.4.0.1 Metodo pivotale

Un metodo per la costruzione degli intervalli di confidenza è il **metodo pivotale**. Tale metodo consiste essenzialmente nel determinare una variabile aleatoria di pivot $\gamma\ (X_1, X_2, ..., X_n; \vartheta)$ che dipende dal campione casuale $X_1, X_2, ..., X_n$ e dal parametro non noto ϑ e la cui funzione di distribuzione non contiene il parametro da stimare. Tale variabile aleatoria non è una statistica poiché dipende dal parametro non noto ϑ e quindi non è osservabile.
Per ogni fissato coefficiente α $(0 < \alpha < 1)$ siano α_1 e α_2 $(\alpha_1 < \alpha_2)$ due valori dipendenti soltanto dal coefficiente fissato α tali che per ogni $\vartheta \in \theta$ si abbia:

$$P\big(\alpha_1 < \gamma(X_1, X_2, \ldots, X_n; \vartheta) < \alpha_2\big) = 1 - \alpha.$$

Se per ogni possibile campione osservato $(x_1, x_2, ..., x_n)$ e per ogni $\vartheta \in \theta$ si riesce a dimostrare che

$$\alpha_1 < \gamma(x; \vartheta) < \alpha_2 \iff g_1(x) < \vartheta < g_2(x)$$

con $g_1(x)$ e $g_2(x)$ dipendenti soltanto dal campione osservato, allora la formula precedente è equivalente a richiedere che

$$P(g_1(X_1, X_2, ..., X_n) < \vartheta) < g_2(X_1, X_2, ..., X_n)) = 1 - \alpha$$

Denotando con $\underline{C}_n = g_1\ (X_1, X_2, ..., X_n)$ e $\overline{C}_n = g_2\ (X_1, X_2, ..., X_n)$, segue che $(\underline{C}_n, \overline{C}_n)$ è un intervallo di confidenza di grado $1 - \alpha$ per il parametro non noto ϑ della popolazione.

2.4.1 Popolazione normale

Sia $X_1, X_2, ..., X_n$ un campione casuale di ampiezza n estratto da una popolazione normale con valore medio μ e varianza σ^2. Si possono analizzare i seguenti problemi:

- determinare un intervallo di confidenza di grado $1 - \alpha$ per il valore medio μ nel caso in cui la varianza σ^2 della popolazione normale è nota;
- determinare un intervallo di confidenza di grado $1 - \alpha$ per il valore medio μ nel caso in cui la varianza della popolazione normale è non nota;
- determinare un intervallo di confidenza di grado $1 - \alpha$ per la varianza σ^2 nel caso in cui il valore medio μ della popolazione normale è noto;
- determinare un intervallo di confidenza di grado $1 - \alpha$ per la varianza σ^2 nel caso in cui il valore medio della popolazione normale è non noto.

Intervallo di confidenza per μ e con σ^2 nota

Per determinare un intervallo di confidenza di grado $1 - \alpha$ per il valore medio μ nel caso in cui la varianza σ^2 della popolazione normale è nota, utilizziamo il metodo pivotale e consideriamo la variabile aleatoria

$$Z_n = \frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}},$$

che è una variabile aleatoria standardizzata (di valor medio nullo e varianza unitaria). Tale variabile aleatoria dipende dal campione casuale e dal parametro non noto μ (la varianza σ^2 è nota) e, quindi, può essere interpretata come una variabile aleatoria di pivot. Inoltre, essendo

$$Z_n = \frac{X_1 + X_2 + \ldots + X_n - n\mu}{\sigma\sqrt{n}}$$

tale variabile aleatoria è distribuita normalmente con valore medio nullo e varianza unitaria, ossia è una normale standard. Scegliendo nel metodo pivotale $\alpha_1 = -z_{\alpha/2}$ e $\alpha_2 = z_{\alpha/2}$, dove $z_{\alpha/2}$ è tale che

$$P(-z_{\alpha/2} < Z_n < z_{\alpha/2}) = 1 - \alpha$$

Ciò è evidenziato nella Figura 2.10:

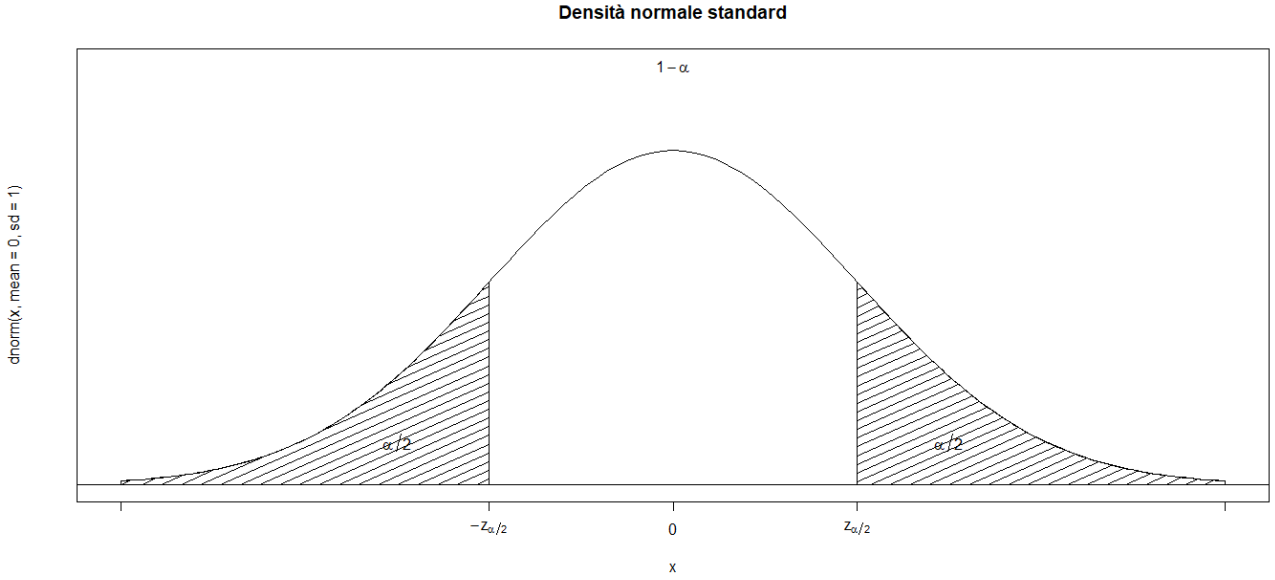


Figura 2.10: Densità normale standard e grado di fiducia 1 - α

Ottenuta con il seguente codice:

```
> curve(dnorm(x, mean=0, sd=1), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità normale standard")
> text(0,0.5, expression(1-alpha))
> axis(1, c(-3, -1, 0, 1, 3), c("", expression(-z[alpha/2]),
+ 0, expression(z[alpha/2]), ""))
>
> vals <- seq(-3,-1, length=100)
> x <- c(-3, vals, -1, -3)
> y <- c(0, dnorm(vals),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(1,3,length=100)
> x <- c(1, vals, 3, 1)
> y <- c(0, dnorm(vals),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(-1.5,0.05, expression(alpha/2))
> text(1.5,0.05, expression(alpha/2))
> box()
```

Proposizione *Sia (x_1, x_2, ..., x_n) un campione osservato di ampiezza n estratto da una popolazione normale con varianza nota σ^2. Una stima dell'intervallo di confidenza di grado* $1 - \alpha$ *per il valore medio* μ *è*

$$\overline{x}_n - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \mu < \overline{x}_n + z_{\alpha/2}\frac{\sigma}{\sqrt{n}},$$

Dove

$$\overline{x}_n = \frac{x_1 + x_2 + \ldots + x_n}{n}.$$

Denota la media campionaria delle n osservazioni.

Considerando il campione precedente generato, formato da n=200 osservazioni, si è trovato che $\overline{x}_{200}$= 3.43899. Supponendo che sia nota la varianza σ^2 =18 viene determinata una stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per il valore medio μ.
In questo caso $\alpha = 0.05$ e $\alpha/2 = 0.025$. Il valore $z_{\alpha/2} = z_{0.025}$ può essere determinato mediante R.

```
> alpha <- 1-0.95
> qnorm(1-alpha/2, mean=0, sd=1)
[1] 1.959964
> n <- length(campione)
> mean(campione)-qnorm(1-alpha/2, mean=0, sd=1)*4/sqrt(n)
[1] 2.884628
> mean(campione)+qnorm(1-alpha/2, mean=0, sd=1)*4/sqrt(n)
[1] 3.993351
```

Si nota quindi che $z_{0.025}$ = 1.959964. La stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per valore medio μ è quindi (2.884628, 3.993351). Si evince che la media campionaria $\overline{x}_{200}$= 3.43899 è compresa nell'intervallo.

Intervallo di confidenza per μ e con varianza non nota

Consideriamo Q_n distribuita con legge chi-quadrato con $n - 1$ gradi di libertà. Per determinare un intervallo di confidenza di grado $1 - \alpha$ per il valore medio μ nel caso in cui la varianza σ^2 della popolazione normale non è nota, viene utilizzato il metodo pivotale e considerata la variabile aleatoria di pivot.

$$T_n = \frac{\overline{X}_n - \mu}{S_n/\sqrt{n}}.$$

Tale variabile aleatoria dipende dal campione casuale e dal parametro non noto μ e, quindi, può essere interpretata come una variabile aleatoria di pivot. Inoltre, poiché

$$T_n = \frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}}\sqrt{\frac{\sigma^2}{S_n^2}} = \frac{Z_n}{\sqrt{Q_n/(n-1)}},$$

segue che T_n è distribuita con legge di Student con $n - 1$ gradi di libertà. Scegliendo nel metodo pivotale $\alpha_1 = -t_{\alpha/2,n-1}$ e $\alpha_2 = t_{\alpha/2,n-1}$, dove $t_{\alpha/2,n-1}$ è tale che

$$P\left(-t_{\alpha/2,n-1} < T_n < t_{\alpha/2,n-1}\right) = 1 - \alpha.$$

Come si può osservare dalla figura, ottenuta tramite il seguente codice:

```
> curve(dt(x, df=5), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità di Student con n-1 gradi di libertà")
> text(0,0.5, expression(1-alpha))
> axis(1, c(-3, -1, 0, 1, 3), c("", expression(-t[list(alpha/2, n-1)]),0,
+ expression(t[list(alpha/2, n-1)]), ""))
>
> vals <- seq(-3,-1, length=100)
> x <- c(-3, vals, -1, -3)
> y <- c(0, dt(vals, df=5),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(1,3,length=100)
> x <- c(1, vals, 3, 1)
> y <- c(0, dt(vals, df=5),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(-1.5,0.05, expression(alpha/2))
> text(1.5,0.05, expression(alpha/2))
> box()
```

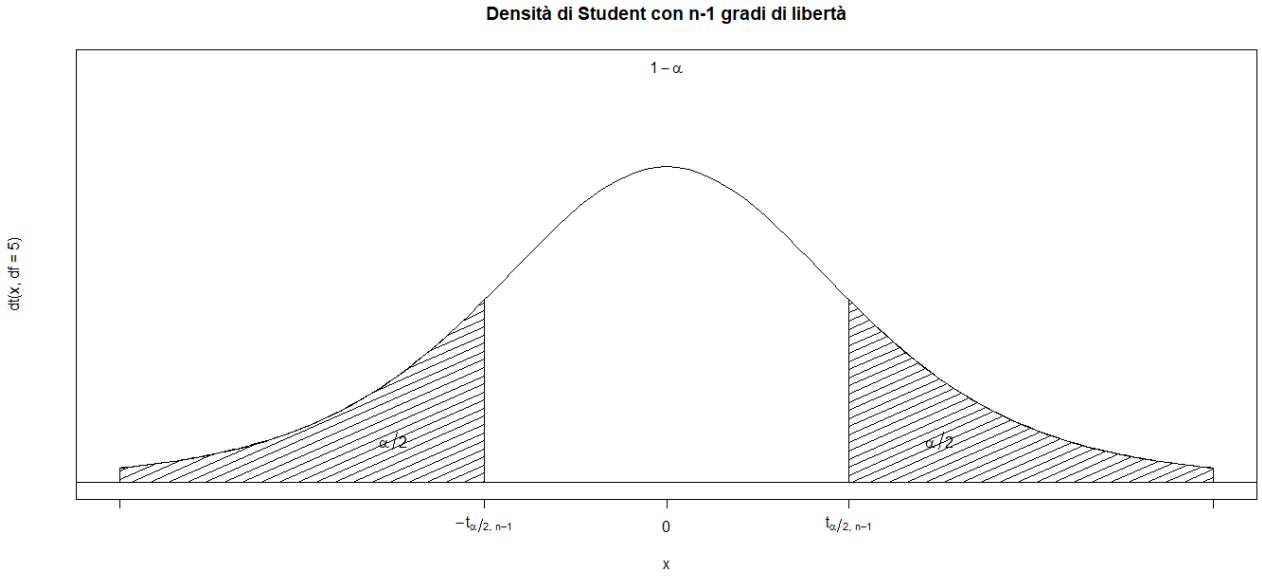


Figura 2.11: Densità di Student con n - 1 gradi di libertà e grado di fiducia 1 - α

Proposizione *Sia* $(x_1, x_2, ..., x_n)$ *un campione osservato di ampiezza n estratto da una popolazione normale con varianza non nota. Una stima dell'intervallo di confidenza di grado* $1 - \alpha$ *per il valore medio* μ *è*

$$\overline{x}_n - t_{\alpha/2,n-1}\frac{s_n}{\sqrt{n}} < \mu < \overline{x}_n + t_{\alpha/2,n-1}\frac{s_n}{\sqrt{n}}$$

Dove

$$\overline{x}_n = \frac{x_1 + x_2 + \ldots + x_n}{n}, \qquad s_n = \left\{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x}_n)^2\right\}^{1/2}$$

denotano rispettivamente la media campionaria e la deviazione standard campionaria delle n osservazioni.

Considerando il campione precedentemente generato formato n = 200 osservazioni, si è trovato che $\overline{x}_{200}$ = 3.43899 e che s_{200} = 4.261658. Viene determinata una stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per il valore medio μ.
In questo caso $\alpha = 0.05$ e $\alpha/2 = 0.025$. Il valore $t_{\alpha/2,n-1} = t_{0.025,199}$ può essere determinato mediante R.

```
> alpha <- 1-0.95
> n <- length(campione)
> qt(1-alpha/2, df=n-1)
[1] 1.971957
> mean(campione)-qt(1-alpha/2, df=n-1)*sd(campione)/sqrt(n)
[1] 2.844751
> mean(campione)+qt(1-alpha/2, df=n-1)*sd(campione)/sqrt(n)
[1] 4.033229
```

Si nota quindi che $t_{0.025,\,199} = 1.971957$. La stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per valore medio μ è quindi (2.844751, 4.033229). Si evince che la media campionaria $\overline{x}_{200} = 3.43899$ è compresa nell'intervallo.

Determiniamo ora una stima dell'intervallo di confidenza $1 - \alpha = 0.99$ per il valore medio μ. In questo caso $\alpha = 0.01$ e $\alpha/2 = 0.005$. Il valore $t_{\alpha/2,n-1} = t_{0.005,199}$ può essere determinato mediante R.

```
> alpha <- 1-0.99
> n <- length(campione)
> qt(1-alpha/2, df=n-1)
[1] 2.60076
> mean(campione)-qt(1-alpha/2, df=n-1)*sd(campione)/sqrt(n)
[1] 2.655265
> mean(campione)+qt(1-alpha/2, df=n-1)*sd(campione)/sqrt(n)
[1] 4.222715
```

Si nota quindi che $t_{0.005,\,199} = 2.60076$. La stima dell'intervallo di confidenza di grado $1 - \alpha = 0.99$ per valore medio μ è quindi (2.655265, 4.222715). Si evince che la media campionaria $\overline{x}_{200} = 3.43899$ è compresa nell'intervallo. Si osserva quindi che aumentando il grado di fiducia aumenta anche la lunghezza dell'intervallo di confidenza.

Intervallo di confidenza per σ^2 e con μ noto

Per determinare un intervallo di confidenza di grado $1 - \alpha$ per la varianza nel caso in cui il valore medio μ della popolazione normale è noto, utilizziamo nuovamente il metodo pivotale e consideriamo la variabile aleatoria di pivot

$$V_n = \sum_{i=1}^{n} \left(\frac{X_i - \mu}{\sigma} \right)^2 = \frac{1}{\sigma^2} \sum_{i=1}^{n} (X_i - \mu)^2$$

Tale variabile dipende dal campione casuale e dal parametro non noto σ^2 (essendo il valore medio μ noto) ed è distribuita con legge chi–quadrato con n gradi di libertà, essendo costituita dalla somma dei quadrati di n variabili aleatorie normali standard. Scegliendo nel metodo pivotale $\alpha_1 = \chi^2_{1-\alpha/2,n}$ e $\alpha_2 = \chi^2_{\alpha/2,n}$ osserviamo che

$$P(\chi^2_{1-\alpha/2,n} < V_n < \chi^2_{\alpha/2,n}) = 1 - \alpha.$$

Ciò si evidenzia nella seguente figura ottenuta col codice:

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n gradi di libertà")
> text(4,0.02, expression(1-alpha))
> axis(1, c(0,2,4,6,12), c("", expression({chi^2}[list(1-alpha/2, n)]),
+ expression(n-2), expression({chi^2}[list(alpha/2, n)]), ""))
>
> vals <- seq(0,2, length=100)
> x <- c(0, vals, 2, 0)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(6,12,length=100)
> x <- c(6, vals, 12, 6)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.2,0.02, expression(alpha/2))
> text(8.5,0.02, expression(alpha/2))
> box()
```

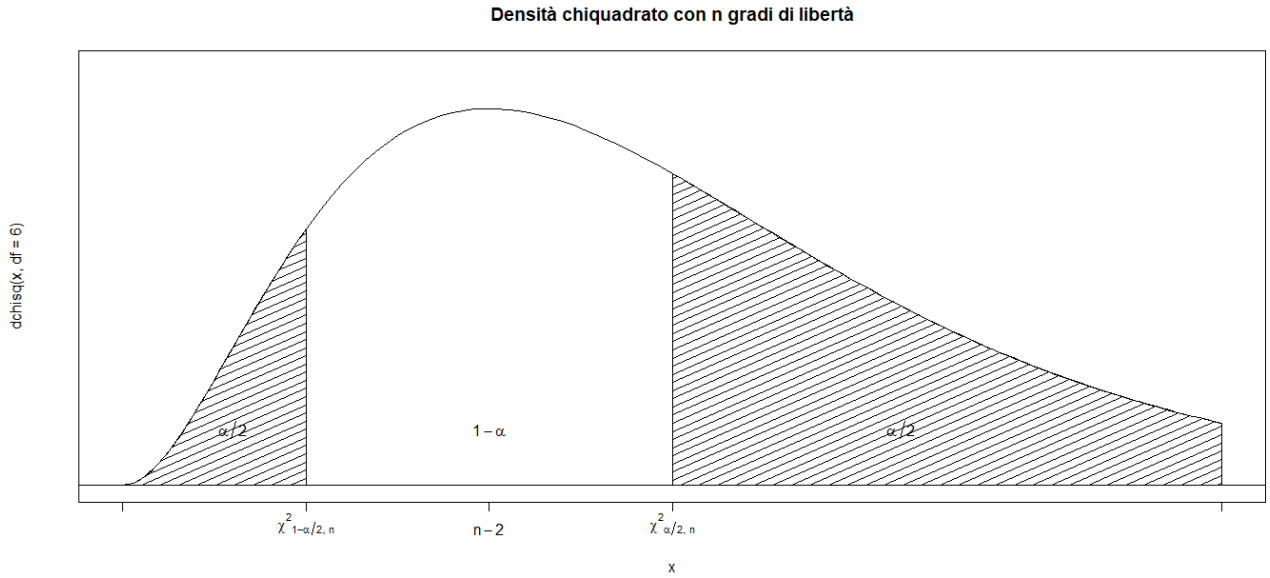


Figura 2.12: Densità chi-quadrato con *n* gradi di libertà e grado di fiducia 1 - *α*

Proposizione *Sia* $(x_1, x_2, ..., x_n)$ *un campione osservato di ampiezza n estratto da una popolazione normale con valore medio noto* μ. *Una stima dell'intervallo di confidenza di grado* $1 - \alpha$ *per la varianza* σ^2 *è*

$$\frac{(n-1)s_n^2 + n(\overline{x}_n - \mu)^2}{X^2_{\alpha/2,n}} < \sigma^2 < \frac{(n-1)s_n^2 + n(\overline{x}_n - \mu)^2}{X^2_{1-\alpha/2,n}}$$

dove

$$\overline{x}_n = \frac{x_1 + x_2 + ... + x_n}{n}, \qquad s_n^2 = \frac{1}{n-1}\sum_{i+1}^{n}(x_i - \overline{x}_n)^2$$

denotano rispettivamente la media campionaria e la varianza campionaria delle n osservazioni.

Considerando il campione precedentemente generato formato n = 200 osservazioni, si è trovato che $\overline{x}_{200} = 3.43899$ e che $s^2_{200} = 18.16173$. Supponendo che sia noto il valore medio $\mu = 3$ e varianza non nota σ^2, viene determinata una stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per la varianza σ^2.
In questo caso $\alpha = 0.05$, $\alpha/2 = 0.025$ e $1 - \alpha/2 = 0.975$. I valori $\chi^2_{1-\alpha/2,n} = \chi^2_{0.975,200}$ e $\chi^2_{\alpha/2,n} = \chi^2_{0.025,200}$ possono essere determinati mediante R.

```
> n <- length(campione)
> mu <- 3
> alpha <- 1-0.95
> qchisq(alpha/2, df=n)
[1] 162.728
> qchisq(1-alpha/2, df=n)
[1] 241.0579
> ((n-1)*var(campione)+n*(mean(campione)-mu)**2)/qchisq(1-alpha/2, df=n)
[1] 15.1529
> ((n-1)*var(campione)+n*(mean(campione)-mu)**2)/qchisq(alpha/2, df=n)
[1] 22.44682
```

Si nota quindi che $\chi^2_{1-\alpha/2,n} = \chi^2_{0.975,200} = 162.728$ e $\chi^2_{\alpha/2,n} = \chi^2_{0.025,200} = 241.0579$. La stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per la varianza è quindi (15.1529, 22.44682). Si evince che la varianza campionaria $\sigma^2 = 18.16173$ è compresa nell'intervallo.

Intervallo di confidenza per σ^2 con valore medio non noto

Per determinare un intervallo di confidenza di grado $1 - \alpha$ per la varianza nel caso in cui il valore medio della popolazione normale non è noto, consideriamo la variabile aleatoria di pivot

$$Q_n = \frac{(n-1)S_n^2}{\sigma^2} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2$$

Tale variabile aleatoria dipende dal campione casuale e dal parametro non noto σ^2 ed è distribuita con legge chi–quadrato con $n - 1$ gradi di libertà. Scegliendo nel metodo pivotale $\alpha_1 = \chi^2_{1-\alpha/2,n-1}$ e $\alpha_2 = \chi^2_{\alpha/2,n-1}$ in maniera tale che

$$P(\chi^2_{1-\alpha/2,n-1} < Q_n < \chi^2_{\alpha/2,n-1}) = 1 - \alpha.$$

Da come si evince nella figura:

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n-1")
> text(4,0.02, expression(1-alpha))
> axis(1, c(0,2,4,6,12), c("", expression({chi^2}[list(1-alpha/2, n-1)])
+ expression(n-2), expression({chi^2}[list(alpha/2, n-1)]), ""))
>
> vals <- seq(0,2, length=100)
> x <- c(0, vals, 2, 0)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(6,12,length=100)
> x <- c(6, vals, 12, 6)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.2,0.02, expression(alpha/2))
> text(8.5,0.02, expression(alpha/2))
> box()
```

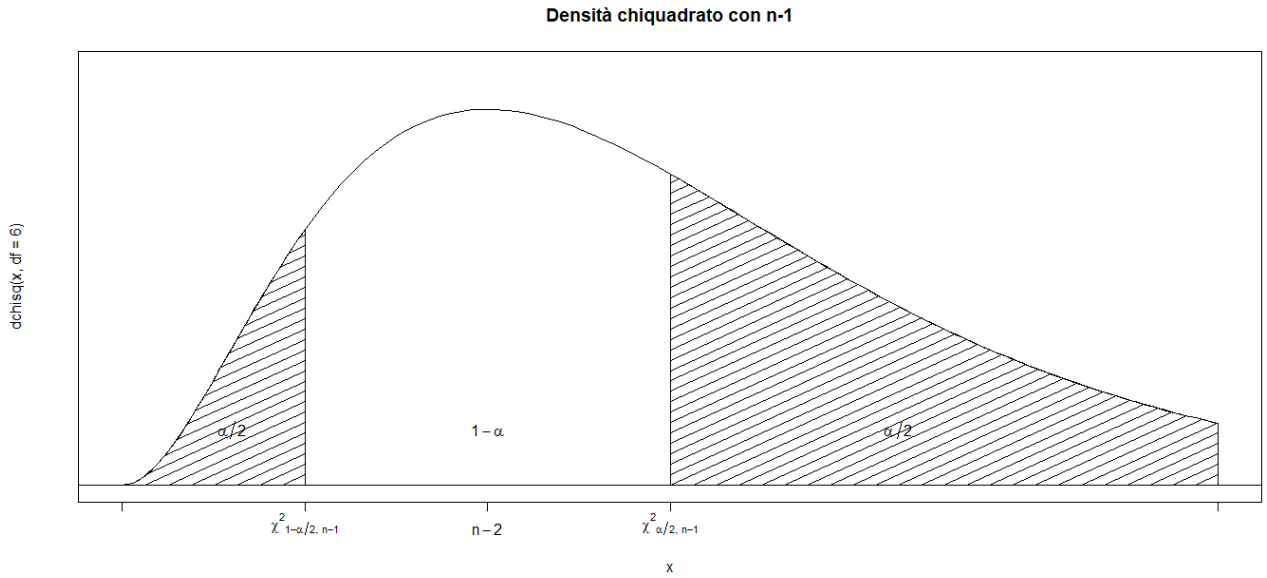


Figura 2.13: Densità chi-quadrato con *n* -1 gradi di libertà e grado di fiducia 1 - *α*

Proposizione *Sia* $(x_1, x_2, ..., x_n)$ *un campione osservato di ampiezza n estratto da una popolazione normale con valore medio non noto. Una stima dell'intervallo di confidenza di grado* $1 - \alpha$ *per la varianza* σ^2 *è*

$$\frac{(n-1)\,s_n^2}{\chi^2_{\alpha/2,n-1}} < \sigma^2 < \frac{(n-1)\,s_n^2}{\chi^2_{1-\alpha/2,n-1}}$$

Dove

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

denota la varianza campionaria delle n osservazioni.

Considerando il campione precedentemente generato formato n = 200 osservazioni, si è trovato che $\overline{x}_{200} = 3.43899$ e che $s^2_{200} = 18.16173$. Si è determinata una stima dell'intervallo di confidenza di grado 1 - $\alpha = 0.95$ per la varianza σ^2.
In questo caso $\alpha = 0.05$, $\alpha/2 = 0.025$ e 1 - $\alpha/2 = 0.975$. I valori $\chi^2_{1-\alpha/2,n-1} = \chi^2_{0.975,199}$ e $\chi^2_{\alpha/2,n-1} = \chi^2_{0.025,199}$ possono essere determinati mediante R.

```
> n <- length(campione)
>
> alpha <- 1-0.95
> qchisq(alpha/2, df=n-1)
[1] 161.8262
> qchisq(1-alpha/2, df=n-1)
[1] 239.9597
> (n-1)*var(campione)/qchisq(1-alpha/2, df=n-1)
[1] 15.06163
> (n-1)*var(campione)/qchisq(alpha/2, df=n-1)
[1] 22.33374
```

Si nota quindi che $\chi^2_{1-\alpha/2,n-1} = \chi^2_{0.975,199} = 161.8562$ e $\chi^2_{\alpha/2,n-1} = \chi^2_{0.025,199} = 239.9597$. La stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per la varianza è quindi (15.06163, 22.33374). Si evince che la varianza campionaria $\sigma^2 = 18.16173$ è compresa nell'intervallo.

Per una popolazione normale le stime per intervallo del valore medio μ e della varianza σ^2 della popolazione possono essere effettuate qualsiasi sia la dimensione del campione casuale osservato. Ciò dipende dalla circostanza favorevole di conoscere la distribuzione esatta della variabile

pivotale considerata: normale e di Student per la stima del valore medio e chi-quadrato per la stima della varianza.
Occorre anche sottolineare che per una popolazione normale i metodi di stima maggiormente utilizzati sono il (2) e il (4), ossia la determinazione di un intervallo di confidenza di grado $1 - \alpha$ per il valore medio μ nel caso in cui la varianza della popolazione normale è non nota e la determinazione un intervallo di confidenza di grado $1 - \alpha$ per la varianza σ^2 nel caso in cui il valore medio della popolazione normale è non noto.

2.5 Intervalli di fiducia approssimati

2.5.1 Differenza tra i valori medi

Vengono ora costruiti degli intervalli di confidenza per la differenza tra i valori medi di due popolazioni normali.

Popolazioni normali

Siano $X_1, X_2, ..., Xn_1$ e $Y_1, Y_2, ..., Yn_2$ due campioni casuali indipendenti di ampiezza n_1 e n_2 estratti rispettivamente da due popolazioni normali $N(\mu_1, \sigma_1^2)$ e $N(\mu_2, \sigma_2^2)$. Vogliamo analizzare i seguenti problemi:

1. determinare un intervallo di confidenza di grado $1 - \alpha$ per $\mu_1 - \mu_2$ quando entrambe le varianze σ_1^2 e σ_2^2 sono note;

2. determinare un intervallo di confidenza di grado $1 - \alpha$ per $\mu_1 - \mu_2$ quando le varianze σ_1^2 e σ_2^2 sono non note per campioni numerosi estratti dalle due popolazioni.

Per costruire gli intervalli di confidenza per la differenza tra i valori medi di due popolazioni normali, generiamo un campione di taglia 250 di una popolazione normale, tale che:

```
> campione2 <- rnorm(250, mean=5, sd=5)
```

Questo campione ha valore medio $\mu = 5$, varianza $\sigma^2 = 25$ e deviazione standard $\sigma = 5$.
Occorre notare che il campione è stato generato in modo pseudocasuale, quindi i valori di media campionaria, deviazione standard campionaria e varianza campionaria del campione risultano essere i seguenti:

```
> mean(campione2)
[1] 5.475405
> var(campione2)
[1] 25.99533
> sd(campione2)
[1] 5.098561
```

Possiamo inoltre calcolare altre misure, come mediana e quantile:

```
> median(campione2)
[1] 5.715716
> quantile(campione2)
        0%        25%        50%        75%       100%
-10.359664   2.353575   5.715716   8.738260  23.565952
```

Tramite le seguenti linee di codice, generiamo il grafico:

```
> hist(campione2, freq=F, xlim= c(-15,20), ylim=c(0,0.3), breaks=100,
+ xlab="x", ylab="Istogramma", main="Densità simulata N=250")
```

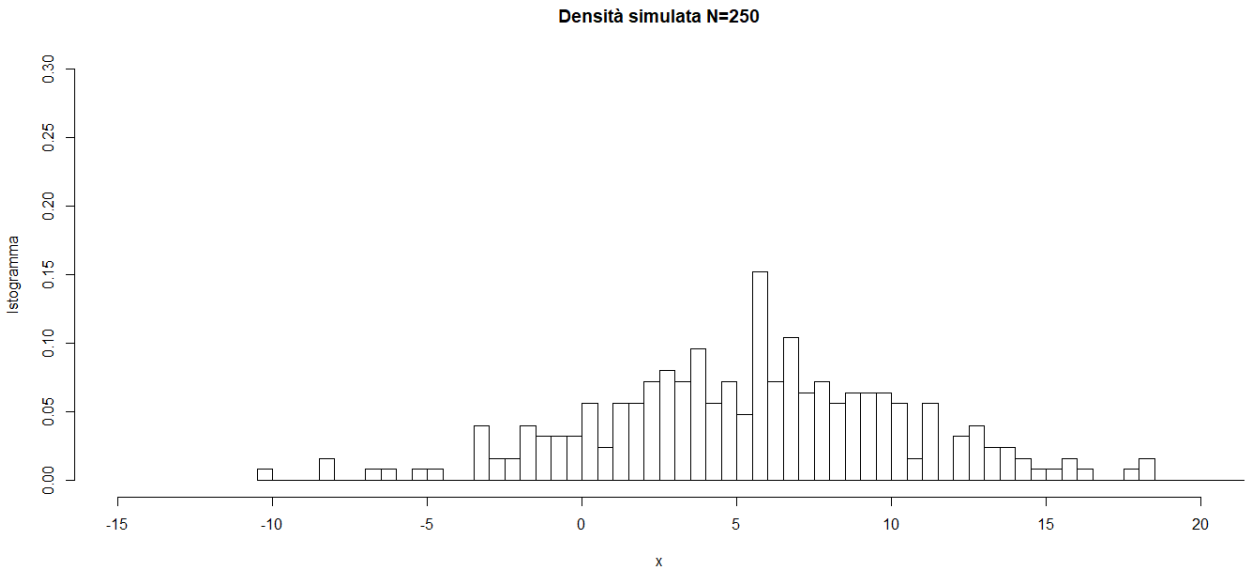


Figura 2.14: Grafico densità simulata N=250

1. Intervallo di confidenza per $\mu_1 - \mu_2$ quando entrambe le varianze σ_1^2 e σ_2^2 sono note

Proposizione *Siano ($x_1, x_2, ..., x_{n1}$) e ($y_1, y_2, ..., y_{n2}$) due campioni osservati indipendenti di ampiezza n_1 e n_2 estratti rispettivamente da due popolazioni normali $N(\mu_1, \sigma_1^2)$ e $N(\mu_2, \sigma_2^2)$ le cui varianze sono note. Una stima dell'intervallo di confidenza di grado $1 - \alpha$ per la differenza tra le due medie $\mu_1 - \mu_2$ è*

$$\overline{x}_{n_1} - \overline{y}_{n_2} - z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} < \mu_1 - \mu_2 < \overline{x}_{n_1} - \overline{y}_{n_2} + z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}},$$

Dove

$$\overline{x}_n = \frac{x_1 + x_2 + \ldots + x_{n_1}}{n_1} \qquad \overline{y}_n = \frac{y_1 + y_2 + \ldots + y_{n_2}}{n_2}.$$

Denotano rispettivamente le medie campionarie delle due osservazioni.

Prendendo in considerazione i due campioni generati $N(\mu_1, \sigma_1^2)$ e $N(\mu_2, \sigma_2^2)$ con le rispettive deviazioni standard $\sigma_1 = 4$ e $\sigma_2 = 5$, si determina quindi una stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per la differenza tra le due medie $\mu_1 - \mu_2$.

Considerando i campioni precedente generati si nota che $\overline{x}_{200} = 3.43899$ e $\overline{x}_{250} = 5.475405$ e che le varianze risultano essere $\sigma_1^2 = 18$ e $\sigma_2^2 = 25$.
In questo caso $\alpha = 0.05$ e $\alpha/2 = 0.025$. Il valore $z_{\alpha/2} = z_{0.025}$ può essere determinato mediante R.

```
> alpha <- 1-0.95
> qnorm(1-alpha/2, mean=0, sd=1)
[1] 1.959964
> n1 <- length(campione)
> n2 <- length(campione2)
> m1 <- mean(campione)
> m2 <- mean(campione2)
> sigma1 <- 4
> sigma2 <- 5
>
> m1-m2-qnorm(1-alpha/2, mean=0, sd=1)*sqrt(sigma1^2/n1+sigma2^2/n2)
[1] -2.867957
> m1-m2+qnorm(1-alpha/2, mean=0, sd=1)*sqrt(sigma1^2/n1+sigma2^2/n2)
[1] -1.204873
```

Si evince che $z_{\alpha/2} = z_{0.025} = 1.959964$. La stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per la differenza tra le due medie $\mu_1 - \mu_2$ è (-2.867957, -1.204873). La differenza dei valori medi è uguale a -2.036415 ed è compresa nell'intervallo.

2. Intervallo di confidenza per $\mu_1 - \mu_2$ quando entrambe le varianze σ_1^2 e σ_2^2 sono non note

Proposizione *Siano $(x_1, x_2, ..., x_{n1})$ e $(y_1, y_2, ..., y_{n2})$ due campioni osservati indipendenti di ampiezza n_1 e n_2 estratti rispettivamente da due popolazioni normali $N(\mu_1, \sigma_1^2)$ e $N(\mu_2, \sigma_2^2)$ le cui varianze sono non note. Una stima dell'intervallo di confidenza di grado $1 - \alpha$ per la differenza tra le due medie $\mu_1 - \mu_2$ è*

$$\overline{x}_{n_1} - \overline{y}_{n_2} - z_{\alpha/2}\sqrt{\frac{s_{n_1}^2}{n_1} + \frac{\tilde{s}_{n_2}^2}{n_2}} < \mu_1 - \mu_2 < \overline{x}_{n_1} - \overline{y}_{n_2} + z_{\alpha/2}\sqrt{\frac{s_{n_1}^2}{n_1} + \frac{\tilde{s}_{n_2}^2}{n_2}},$$

dove $\bar{x}_{n1}$ e $\bar{y}_{n2}$ denotano rispettivamente le medie campionarie delle due osservazioni e dove s_{n1}^2 e $\tilde{s}_{n2}^2$ denotano rispettivamente le varianze campionarie delle due osservazioni.

Considerando i due campioni generati $N(\mu_1, \sigma_1^2)$ e $N(\mu_2, \sigma_2^2)$ con varianze non note, si determina quindi una stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per la differenza tra le due medie $\mu_1 - \mu_2$.
Si nota che $\overline{x}_{200} = 3.43899$ e $\overline{x}_{250} = 5.475405$ e che le varianze risultano essere $s_{n1}^2 = 18.16173$ e $\tilde{s}_{n2}^2 = 25.99533$; inoltre $\alpha = 0.05$ e $\alpha/2 = 0.025$. Usando R si ottiene:

```
> alpha <- 1-0.95
> qnorm(1-alpha/2, mean=0, sd=1)
[1] 1.959964
> n1 <- length(campione)
> n2 <- length(campione2)
> m1 <- mean(campione)
> m2 <- mean(campione2)
> s1 <- sd(campione)
> s2 <- sd(campione2)
> sigma1 <- 4
> sigma2 <- 5
>
> m1-m2-qnorm(1-alpha/2, mean=0, sd=1)*sqrt(s1^2/n1+s2^2/n2)
[1] -2.901445
> m1-m2+qnorm(1-alpha/2, mean=0, sd=1)*sqrt(s1^2/n1+s2^2/n2)
[1] -1.171385
```

Si evince che $z_{\alpha/2} = z_{0.025} = 1.959964$. La stima dell'intervallo di confidenza di grado $1 - \alpha = 0.95$ per la differenza tra le due medie $\mu_1 - \mu_2$ è (-2.901445, -1.171385). La differenza dei valori medi è uguale a -2.036415 ed è compresa nell'intervallo.

2.6 Verifica delle ipotesi

In generale gli elementi che costituiscono il punto di partenza del procedimento di verifica delle ipotesi sono una popolazione descritta da una variabile aleatoria X caratterizzata da una funzione di probabilità o densità di probabilità $f(x; \vartheta)$, un'ipotesi su di un parametro non noto della popolazione ed un campione casuale $X_1, X_2, ..., X_n$ estratto dalla popolazione. Occorre in primo luogo precisare il significato di ipotesi statistica.

Definizione *Un'ipotesi statistica è un'affermazione o una congettura sul parametro non noto ϑ. Se l'ipotesi statistica specifica completamente $f(x; \vartheta)$ è detta ipotesi semplice, altrimenti è chiamata ipotesi composta.*

Per denotare un'ipotesi statistica useremo il carattere **H** seguito dai due punti e successivamente dall'affermazione che specifica l'ipotesi.

L'ipotesi soggetta a verifica viene in genere denotata con $\mathbf{H}_0$ e viene chiamata **ipotesi nulla**. Si chiama *test di ipotesi* il procedimento o regola con cui si decide, sulla base dei dati del campione, se accettare o rifiutare $\mathbf{H}_0$. La costruzione del test richiede la formulazione, in contrapposizione all'ipotesi nulla, di una proposizione alternativa. Questa proposizione prende il nome di **ipotesi alternativa** ed è di solito indicata con $\mathbf{H}_1$. L'ipotesi nulla, cioè l'ipotesi soggetta a verifica, si ha quando $\vartheta \in \theta_0$ e l'ipotesi alternativa si ha quando $\vartheta \in \theta_1$ e si scrive

$$\mathbf{H}_0 : \vartheta \in \theta_0, \ \mathbf{H}_1 : \vartheta \in \theta_1,$$

avendo denotato con θ_0 e θ_1 due sottoinsiemi disgiunti dello spazio θ dei parametri.
Il problema della verifica delle ipotesi consiste nel determinare un test ψ che permetta di suddividere, mediante opportuni criteri, l'insieme dei possibili campioni, ossia l'insieme delle n-ple $(x_1, x_2, ..., x_n)$ assumibili dal vettore aleatorio $X_1, X_2, ..., X_n$, in due sottoinsiemi: una **regione di accettazione** $\boldsymbol{A}$ dell'ipotesi nulla ed una **regione di rifiuto** $\boldsymbol{R}$ dell'ipotesi nulla.
Il test ψ può essere così formulato:

- accettare come valida l'ipotesi nulla se il campione osservato $(x_1, x_2, ..., x_n) \in A$;
- rifiutare l'ipotesi nulla se $(x_1, x_2, ..., x_n) \in R$

Nel caso si verifichi che l'ipotesi nulla sia falsa, l'ipotesi alternativa sarà vera e viceversa.
Nel seguire questo tipo di ragionamento si può incorrere in due tipi di errori:

- *rifiutare l'ipotesi nulla* $\boldsymbol{H_0}$ *nel caso in cui tale ipotesi sia vera*; si dice allora che si commette un errore di tipo *I* e si denota la probabilità di commettere tale errore con

$$\alpha(\vartheta) = P(\text{rifiutare } \boldsymbol{H_0}|\vartheta), \ \vartheta \in \theta_0;$$

- *accettare l'ipotesi nulla* $\boldsymbol{H_0}$ *nel caso in cui tale ipotesi sia falsa*; si dice allora che si commette un errore di tipo *II* e si denota la probabilità di commettere tale errore con

$$\beta(\vartheta) = P(\text{accettare } \boldsymbol{H_0}|\vartheta), \ \vartheta \in \theta_1;$$

In generale per campioni casuali di fissata ampiezza, se si diminuisce la probabilità di commettere un errore di tipo *I* aumenta la probabilità di commettere un errore di tipo *II* e viceversa. Nella costruzione del test conviene quindi fissare la probabilità di commettere un errore di tipo *I* e cercare un test ψ che minimizzi la probabilità di commettere un errore di tipo *II*, ciò deriva dal fatto che di solito le ipotesi vengono formulate in maniera tale che l'errore di tipo *I* sia più grave e quindi il decisore desidera imporre che la probabilità di commettere tale errore sia piccola.

I test statistici sono di due tipi: *test unilaterali* (detti anche unidirezionali) e *test bilaterali* (detti anche bidirezionali). Un **test bilaterale** è il seguente

$$\mathbf{H}_0 : \vartheta = \vartheta_0$$
$$\mathbf{H}_1 : \vartheta \neq \vartheta_0,$$

mentre **test unilaterali** sono i seguenti

$$\mathbf{H}_0: \vartheta \leq \vartheta_0 \qquad \mathbf{H}_0: \vartheta \geq \vartheta_0$$
$$\mathbf{H}_1: \vartheta > \vartheta_0 \qquad \mathbf{H}_1: \vartheta < \vartheta_0$$

2.6.1 Popolazione normale

Utilizzando test bilaterali e unilaterali, desideriamo affrontare i seguenti problemi:

1. Verifica di ipotesi sul valore medio μ nel caso in cui la varianza σ^2 della popolazione normale è nota;
2. Verifica di ipotesi sul valore medio μ nel caso in cui la varianza della popolazione normale è non nota;
3. Verifica di ipotesi sulla varianza σ^2 nel caso in cui il valore medio μ della popolazione normale è noto;
4. Verifica di ipotesi sulla varianza σ^2 nel caso in cui il valore medio della popolazione normale è non noto.

1. Test su μ con varianza σ^2 nota

Sia $X_1, X_2, ..., X_n$ un campione casuale estratto da una popolazione normale con varianza nota σ^2.

Test bilaterale: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con varianza σ^2 nota. Si considerino le ipotesi:

$$\mathbf{H}_0 : \mu = \mu_0, \qquad \mathbf{H}_1 : \mu \neq \mu_0$$

Essendo la varianza nota, l'ipotesi $\mathbf{H}_0$ è semplice, mentre l'ipotesi $\mathbf{H}_1$ è composta. Quando $\mathbf{H}_0$ è vera, in analogia a quanto visto per gli intervalli di confidenza, gioca un ruolo fondamentale la variabile aleatoria

$$Z_n = \frac{\overline{X}_n - \mu_0}{\sigma/\sqrt{n}},$$

che è distribuita normalmente con valore medio nullo e varianza unitaria. Il test bilaterale ψ di misura α per le ipotesi considerate è il seguente:

- si accetti $\mathbf{H_0}$ se $\quad -z_{\alpha/2} < \dfrac{\overline{x}_n - \mu_0}{\sigma/\sqrt{n}} < z_{\alpha/2}$

- si rifiuti $\mathbf{H_0}$ se $\quad \dfrac{\overline{x}_n - \mu_0}{\sigma/\sqrt{n}} < -z_{\alpha/2} \quad$ oppure $\quad \dfrac{\overline{x}_n - \mu_0}{\sigma/\sqrt{n}} > z_{\alpha/2}$

Nella Figura sottostante è possibile visualizzare la densità normale standard e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale.

```
> curve(dnorm(x, mean=0, sd=1), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità normale standard")
> text(0,0.5, expression(1-alpha))
> text(0,0.2, "Regione di \naccettazione")
> axis(1, c(-3, -1, 0, 1, 3), c("", expression(-z[alpha/2]),
+ 0, expression(z[alpha/2]), ""))
>
> vals <- seq(-3,-1, length=100)
> x <- c(-3, vals, -1, -3)
> y <- c(0, dnorm(vals),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(1,3,length=100)
> x <- c(1, vals, 3, 1)
> y <- c(0, dnorm(vals),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(-1.5,0.05, expression(alpha/2))
> text(-2.2,0.1, "Regione di \nrifiuto")
> text(1.5,0.05, expression(alpha/2))
> text(2.2,0.1, "Regione di \nrifiuto")
> box()
```

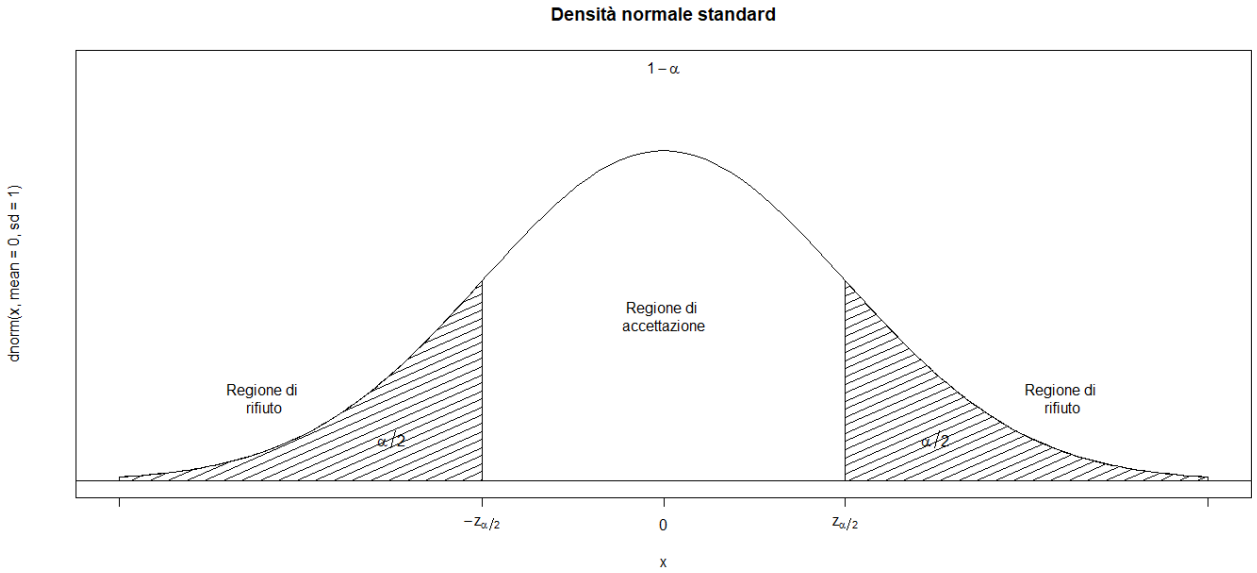


Figura 2.15: Densità normale standard e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale

```
> alpha <- 0.05
> mu0 <- 3
> sigma <- 4
> qnorm(1-alpha/2, mean=0, sd=1)
[1] 1.959964
> n <- 200
> meancamp <- 3.43899
> (meancamp-mu0)/(sigma/sqrt(n))
[1] 1.552064
```

Il valore $z_{\alpha/2}$ = 1.959964 e z = 1.552064 cade nella regione di accettazione, bisogna quindi accettare l'ipotesi nulla.

Test unilaterale sinistro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con varianza σ^2 nota. Si considerino le ipotesi:

$$\mathbf{H}_0 : \mu \leq \mu_0, \qquad \mathbf{H}_1 : \mu > \mu_0$$

Le ipotesi $\mathbf{H}_0$ e $\mathbf{H}_1$ sono entrambe composte. Il test unilaterale sinistro ψ di misura α per le ipotesi considerate è il seguente:

- si accetti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{\sigma/\sqrt{n}} < z_\alpha$

- si rifiuti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{\sigma/\sqrt{n}} > z_\alpha$

Nella Figura sottostante è possibile visualizzare la densità normale standard e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro.

```
> curve(dnorm(x, mean=0, sd=1), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità normale standard")
> text(0,0.5, expression(1-alpha))
> text(0,0.2, "Regione di \naccettazione")
> axis(1, c(-3, -1, 0, 1, 3), c("", " "," ", expression(z[alpha]), ""))
>
> vals <- seq(1,3,length=100)
> x <- c(1, vals, 3, 1)
> y <- c(0, dnorm(vals),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.5,0.05, expression(alpha))
> text(2.2,0.1, "Regione di \nrifiuto")
> box()
```

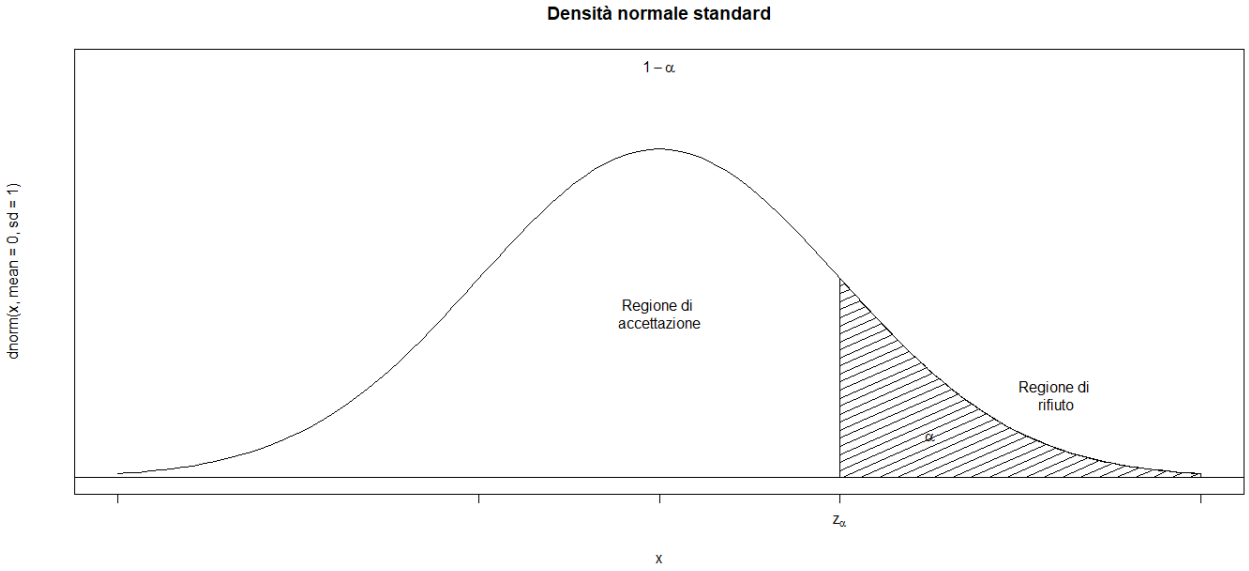


Figura 2.16: Densità normale standard e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro

```
> alpha <- 0.05
> mu0 <- 3
> sigma <- 4
> qnorm(1-alpha, mean=0, sd=1)
[1] 1.644854
> n <- 200
> meancamp <-  3.43899
> (meancamp-mu0)/(sigma/sqrt(n))
[1] 1.552064
```

L'estremo sinistro $z_\alpha = 1.644854$. Per accettare il test unilaterale sinistro, il valore calcolato deve essere minore di z_α. In questo caso la proprietà è soddisfatta poiché il valore calcolato risulta essere minore di z_α, che ricade quindi nella zona di accettazione. Il test unilaterale sinistro è accettato.

Test unilaterale destro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con varianza σ^2 nota. Si considerino le ipotesi:

$$\mathbf{H}_0 : \mu \geq \mu_0, \qquad \mathbf{H}_1 : \mu < \mu_0$$

Le ipotesi $\mathbf{H}_0$ e $\mathbf{H}_1$ sono entrambe composte. Il test unilaterale destro ψ di misura α per le ipotesi considerate è il seguente

- si accetti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{\sigma/\sqrt{n}} > -z_\alpha$

- si rifiuti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{\sigma/\sqrt{n}} < -z_\alpha$

Nella Figura sottostante è possibile visualizzare la densità normale standard e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro.

```
> curve(dnorm(x, mean=0, sd=1), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità normale standard")
> text(0,0.5, expression(1-alpha))
> text(0,0.2, "Regione di \naccettazione")
> axis(1, c(-3, -1, 0, 1, 3), c("", expression(-z[alpha]), " ", " ",""))
>
> vals <- seq(-3,-1, length=100)
> x <- c(-3, vals, -1, -3)
> y <- c(0, dnorm(vals),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(-1.5,0.05, expression(alpha))
> text(-2.2,0.1, "Regione di \nrifiuto")
> box()
```

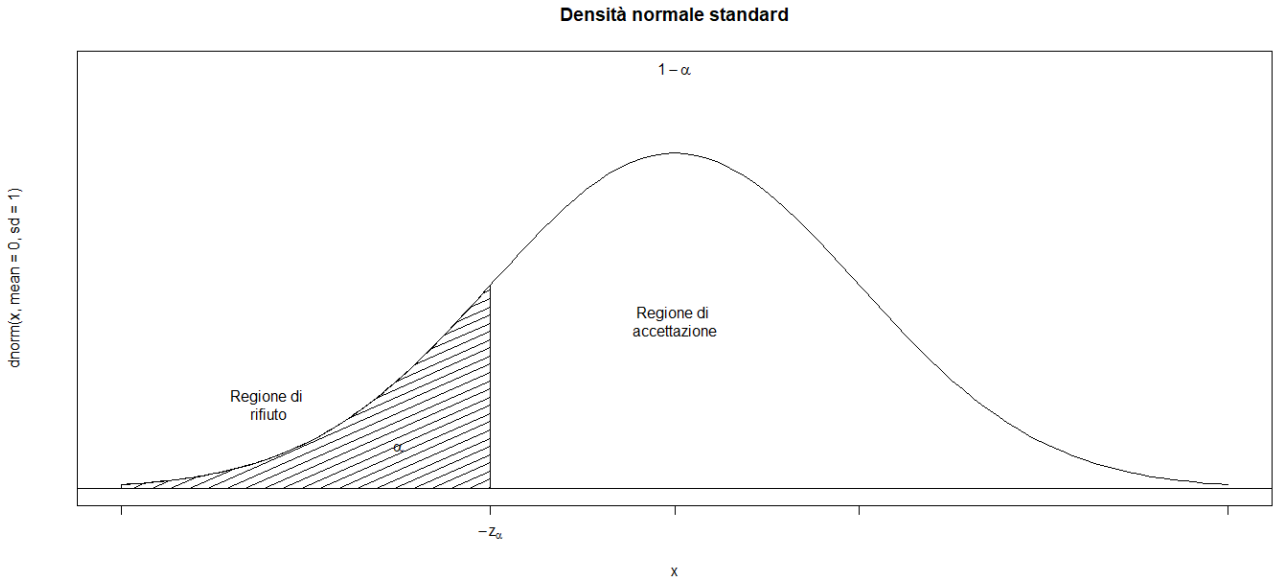


Figura 2.17: Densità normale standard e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro

```
> alpha <- 0.05
> mu0 <- 3
> sigma <- 4
> qnorm(alpha, mean=0, sd=1)
[1] -1.644854
> n <- 200
> meancamp <- 3.43899
> (meancamp-mu0)/(sigma/sqrt(n))
[1] 1.552064
```

L'estremo destro $-z_\alpha$ = -1.644854. Per accettare il test unilaterale destro, il valore calcolato z deve essere maggiore di $-z_\alpha$. Anche in questo caso la proprietà è soddisfatta e viene quindi accettata l'ipotesi nulla.

2. Test su μ con varianza non nota

Sia $X_1, X_2, ..., X_n$ un campione casuale estratto da una popolazione normale con varianza non nota σ^2.

Test bilaterale: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con varianza σ^2 non nota. Si considerino le ipotesi:

$$\mathbf{H}_0 : \mu = \mu_0, \qquad \mathbf{H}_1 : \mu \neq \mu_0$$

Essendo la varianza non nota, le ipotesi sono entrambe composte. Quando $\mathbf{H}_0$ è vera, in analogia a quanto visto per gli intervalli di confidenza, gioca un ruolo fondamentale la variabile aleatoria

$$T_n = \frac{\overline{X}_n - \mu_0}{S_n/\sqrt{n}}$$

che è distribuita con legge di Student con $n - 1$ gradi di libertà. Il test bilaterale ψ di misura α per le ipotesi considerate è il seguente:

- si accetti $\mathbf{H_0}$ se $\quad -t_{\alpha/2,n-1} < \dfrac{\overline{x}_n - \mu_0}{s_n/\sqrt{n}} < t_{\alpha/2,n-1}$

- si rifiuti $\mathbf{H_0}$ se $\quad \dfrac{\overline{x}_n - \mu_0}{s_n/\sqrt{n}} < -t_{\alpha/2,n-1}$ oppure $\dfrac{\overline{x}_n - \mu_0}{s_n/\sqrt{n}} > t_{\alpha/2,n-1}$

Nella Figura sottostante è possibile visualizzare la densità di Student con $n - 1$ gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale.

```
> curve(dt(x, df=5), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità di Student con n-1 gradi di libertà")
> text(0,0.5, expression(1-alpha))
> text(0,0.2, "Regione di \naccettazione")
> axis(1, c(-3, -1, 0, 1, 3), c("", expression(-t[list(alpha/2, n-1)]),0,
+ expression(t[list(alpha/2, n-1)]), ""))
>
> vals <- seq(-3,-1, length=100)
> x <- c(-3, vals, -1, -3)
> y <- c(0, dt(vals, df=5),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(1,3,length=100)
> x <- c(1, vals, 3, 1)
> y <- c(0, dt(vals, df=5),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(-1.5,0.05, expression(alpha/2))
> text(-2.2,0.1, "Regione di \nrifiuto")
> text(1.5,0.05, expression(alpha/2))
> text(2.2,0.1, "Regione di \nrifiuto")
> box()
```

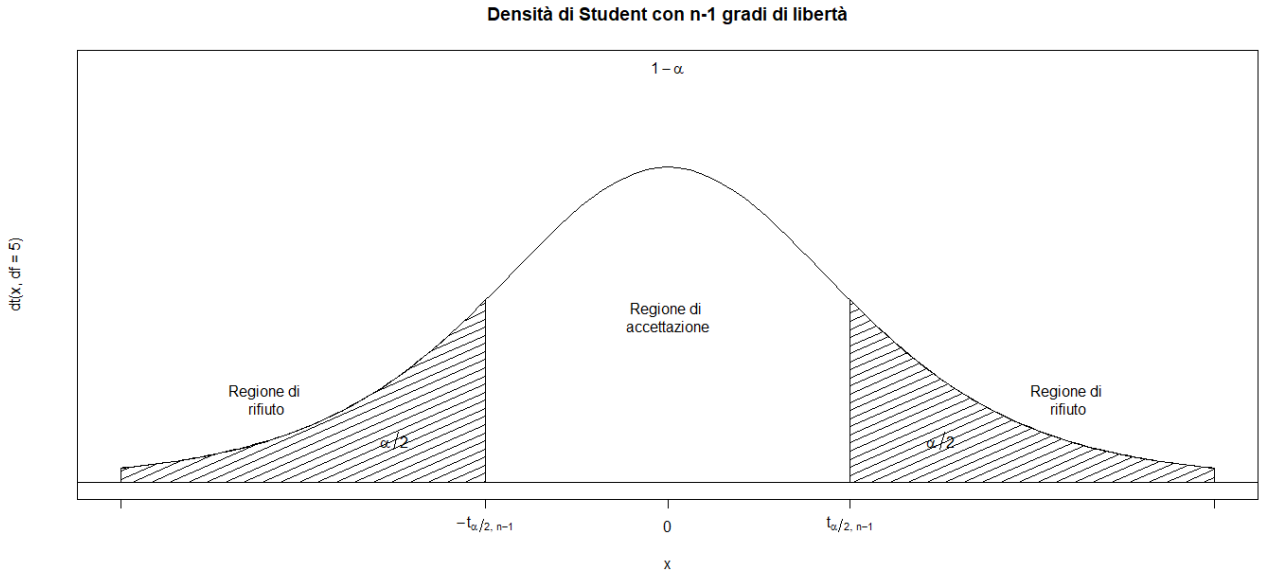


Figura 2.18: Densità di Student con $n - 1$ gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale

```
> alpha <- 0.01
> mu0 <- 3
> n <- 200
> qt(1-alpha/2, df=n-1)
[1] 2.60076
> meancamp <- 3.43899
> devcamp<- 4.261658
> (meancamp-mu0)/(devcamp/sqrt(n))
[1] 1.45677
```

Si nota che $t_{\alpha/2,n-1} = 2.60076$ e $t = 1.45677$ cade nella regione di accettazione. Occorre quindi

accettare l'ipotesi nulla.

Test unilaterale sinistro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con varianza σ^2 non nota. Si considerino le ipotesi:

$$\mathbf{H}_0 : \mu \leq \mu_0, \qquad \mathbf{H}_1 : \mu > \mu_0$$

Le ipotesi $\mathbf{H}_0$ e $\mathbf{H}_1$ sono entrambe composte. Il test unilaterale sinistro ψ di misura α per le ipotesi considerate è il seguente:

- si accetti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{s_n/\sqrt{n}} < t_{\alpha,n-1}$

- si rifiuti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{s_n/\sqrt{n}} > t_{\alpha,n-1}$

Nella Figura sottostante è possibile visualizzare la densità di Student con n – 1 gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro.

```
> curve(dt(x, df=5), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità di Student con n-1 gradi di libertà")
> text(0,0.5, expression(1-alpha))
> text(0,0.2, "Regione di \naccettazione")
> axis(1, c(-3, -1, 0, 1, 3), c("", " ", " ", expression(t[list(alpha, n-1)]), ""))
>
> vals <- seq(1,3,length=100)
> x <- c(1, vals, 3, 1)
> y <- c(0, dt(vals, df=5),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.5,0.05, expression(alpha))
> text(2.2,0.1, "Regione di \nrifiuto")
> box()
```

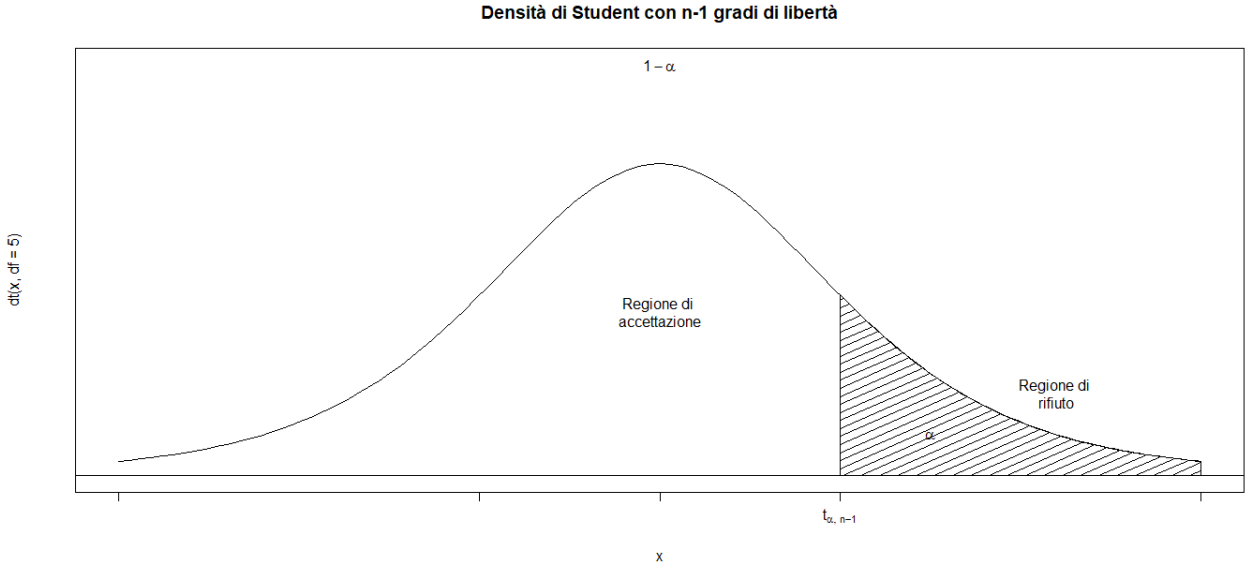


Figura 2.19: Densità di Student con n – 1 gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro

```
> alpha <- 0.01
> mu0 <- 3
> n <- 200
> qt(1-alpha, df=n-1)
[1] 2.345232
> meancamp <- 3.43899
> devcamp<- 4.261658
> (meancamp-mu0)/(devcamp/sqrt(n))
[1] 1.45677
```

Si nota che $t_{\alpha,n-1} = 2.345232$ e $t = 1.45677$ cade nella regione di accettazione. Occorre quindi accettare l'ipotesi nulla.

Test unilaterale destro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con varianza σ^2 non nota. Si considerino le ipotesi:

$$\mathbf{H}_0 : \mu \geq \mu_0, \qquad \mathbf{H}_1 : \mu < \mu_0$$

Il test unilaterale destro ψ di misura α per le ipotesi considerate è il seguente

- si accetti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{s_n/\sqrt{n}} > -t_{\alpha,n-1}$

- si rifiuti $\mathbf{H_0}$ se $\dfrac{\overline{x}_n - \mu_0}{s_n/\sqrt{n}} < -t_{\alpha,n-1}$

Nella Figura sottostante è possibile visualizzare la densità di Student con $n - 1$ gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro.

```
> curve(dt(x, df=5), from=-3, to=3, axes=FALSE, ylim=c(0,0.5),
+ main="Densità di Student con n-1 gradi di libertà")
> text(0,0.5, expression(1-alpha))
> text(0,0.2, "Regione di \naccettazione")
> axis(1, c(-3, -1, 0, 1, 3), c("", expression(-t[list(alpha, n-1)]),"",
+ "", ""))
>
> vals <- seq(-3,-1, length=100)
> x <- c(-3, vals, -1, -3)
> y <- c(0, dt(vals, df=5),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(-1.5,0.05, expression(alpha))
> text(-2.2,0.1, "Regione di \nrifiuto")
> box()
```

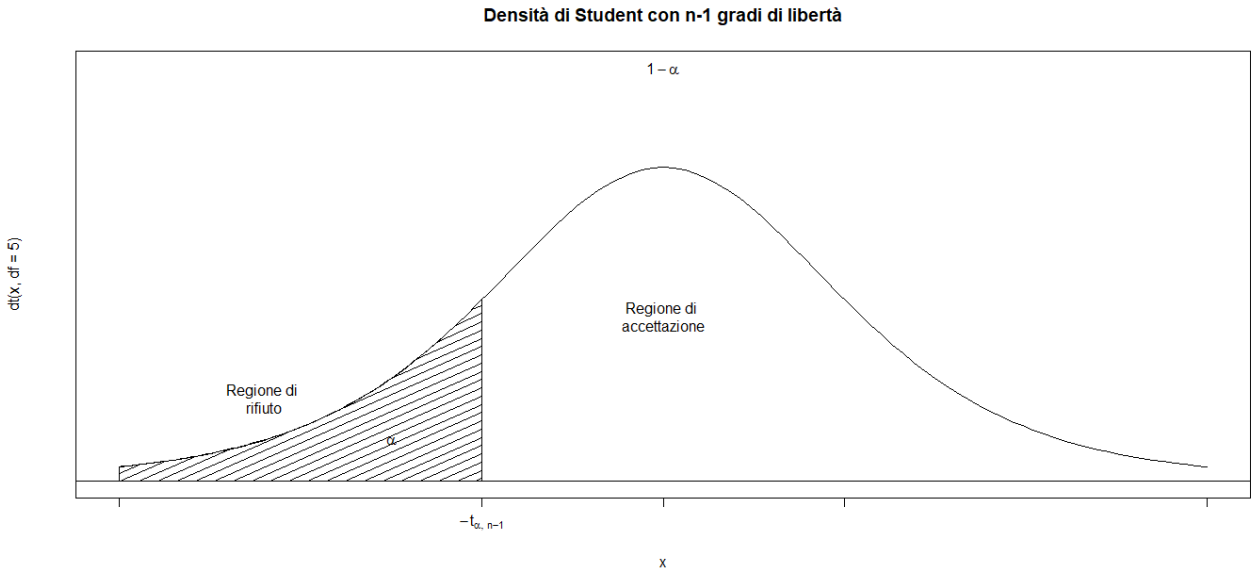


Figura 2.20: Densità di Student con $n - 1$ gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro

```
> alpha <- 0.01
> mu0 <- 3
> n <- 200
> qt(alpha, df=n-1)
[1] -2.345232
> meancamp <- 3.43899
> devcamp<- 4.261658
> (meancamp-mu0)/(devcamp/sqrt(n))
[1] 1.45677
```

Si nota che $t_{\alpha,n-1} = -2.345232$ e $t = 1.45677$ cade nella regione di accettazione. Occorre quindi accettare l'ipotesi nulla.

3. Test su σ^2 con valore medio μ noto

Sia $X_1, X_2, ..., X_n$ un campione casuale estratto da una popolazione normale con valore medio noto μ.

Test bilaterale: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con valore medio μ noto. Si considerino le ipotesi:

$$\mathbf{H}_0 : \sigma^2 = \sigma_0^2, \qquad \mathbf{H}_1 : \sigma^2 \neq \sigma_0^2$$

Essendo il valore medio noto, l'ipotesi $\mathbf{H}_0$ è semplice mentre $\mathbf{H}_1$ è composta. Quando $\mathbf{H}_0$ è vera, in analogia a quanto visto per gli intervalli di confidenza, gioca un ruolo fondamentale la variabile aleatoria

$$V_n = \sum_{i=1}^{n} \left(\frac{X_i - \mu}{\sigma_0} \right)^2 = \frac{(n-1)\, S_n^2}{\sigma_0^2} + \left(\frac{\overline{X}_n - \mu}{\sigma_0 / \sqrt{n}} \right)^2$$

che è distribuita con legge chi-quadrato con n gradi di libertà. Il test bilaterale ψ di misura α per le ipotesi considerate è il seguente:

- si accetti $\mathbf{H_0}$ se $\chi^2_{1-\alpha/2,n} < \sum_{i=1}^{n} \left(\frac{x_i - \mu}{\sigma_0} \right)^2 < \chi^2_{\alpha/2,n}$

- si rifiuti $\mathbf{H_0}$ se $\sum_{i=1}^{n} \left(\frac{x_i - \mu}{\sigma_0} \right)^2 < \chi^2_{1-\alpha/2,n}$ oppure $\sum_{i=1}^{n} \left(\frac{x_i - \mu}{\sigma_0} \right)^2 > \chi^2_{\alpha/2,n}$

Nella Figura sottostante è possibile visualizzare la densità chi-quadrato con *n* gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale.

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n gradi di libertà")
> text(4,0.02, expression(1-alpha))
> text(4,0.10, "Regione di \naccettazione")
> axis(1, c(0,2,4,6,12), c("", expression({chi^2}[list(1-alpha/2, n)]),
+ expression(n-2), expression({chi^2}[list(alpha/2, n)]), ""))
>
> vals <- seq(0,2, length=100)
> x <- c(0, vals, 2, 0)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(6,12,length=100)
> x <- c(6, vals, 12, 6)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.2,0.02, expression(alpha/2))
> text(0.5,0.07, "Regione di \nrifiuto")
> text(8.5,0.02, expression(alpha/2))
> text(8.8,0.08, "Regione di \nrifiuto")
> box()
```

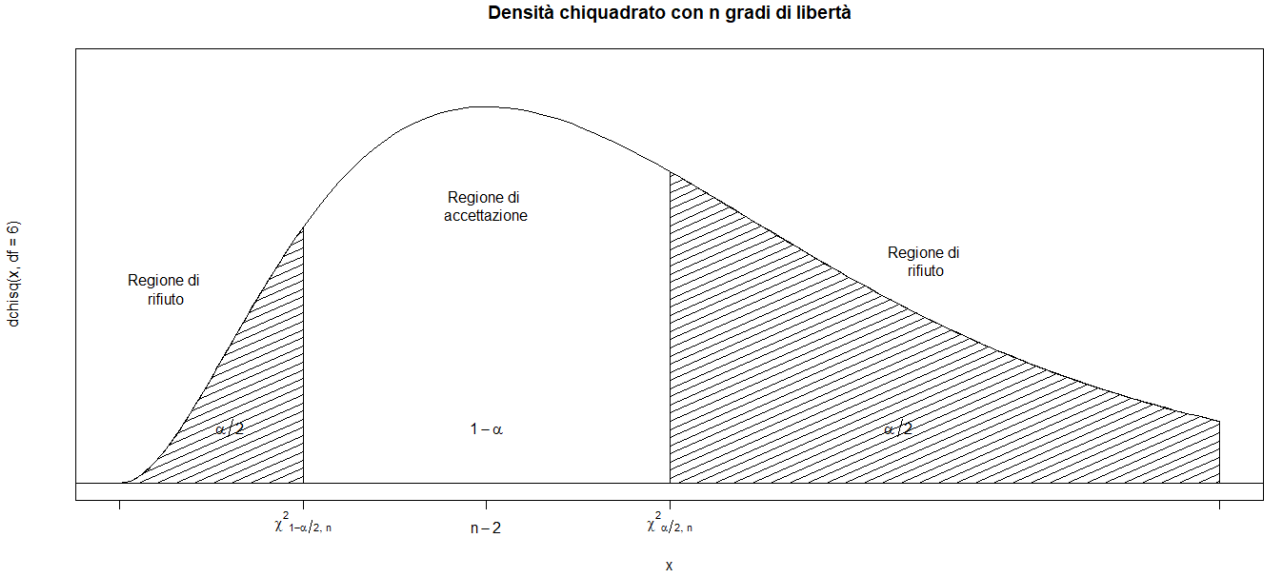


Figura 2.21: Densità chi-quadrato con *n* gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale

```
> alpha <- 0.05
> mu <- 3
> sigma02 <- 18
> n <- 200
> medcamp <- 3.43899
> varcamp <- 18.16173
> qchisq(alpha/2, df=n)
[1] 162.728
> qchisq(1-alpha/2, df=n)
[1] 241.0579
> (n-1)*varcamp/sigma02+n*(medcamp-mu)**2/sigma02
[1] 202.9293
```

Si nota quindi che $X_{1-\alpha/2,200}$ = 162.728, $X_{\alpha/2,200}$ = 241.0579, X^2 = 202.9293. Il valore osservato X^2 è compreso nella regione di accettazione, si accetta l'ipotesi nulla.

Test unilaterale sinistro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con valore medio μ noto. Si considerino le ipotesi:

$$\mathbf{H}_0 : \sigma^2 \leq \sigma_0^2 \qquad \mathbf{H}_1 : \sigma^2 > \sigma_0^2$$

Le ipotesi $\mathbf{H}_0$ e $\mathbf{H}_1$ sono entrambe composte. Il test unilaterale sinistro ψ di misura α per le ipotesi considerate è il seguente:

- si accetti $\mathbf{H_0}$ se $\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma_0}\right)^2 < \chi^2_{\alpha,n}$

- si rifiuti $\mathbf{H_0}$ se $\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma_0}\right)^2 > \chi^2_{\alpha,n}$

Nella Figura sottostante è possibile visualizzare la densità chi-quadrato con n gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n gradi di libertà")
> text(4,0.02, expression(1-alpha))
> text(4,0.10, "Regione di \naccettazione")
> axis(1, c(0,2,4,6,12), c("", "",
+ expression(n-2), expression({chi^2}[list(alpha, n)]), ""))
>
> vals <- seq(6,12,length=100)
> x <- c(6, vals, 12, 6)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(8.5,0.02, expression(alpha))
> text(8.8,0.08, "Regione di \nrifiuto")
> box()
```

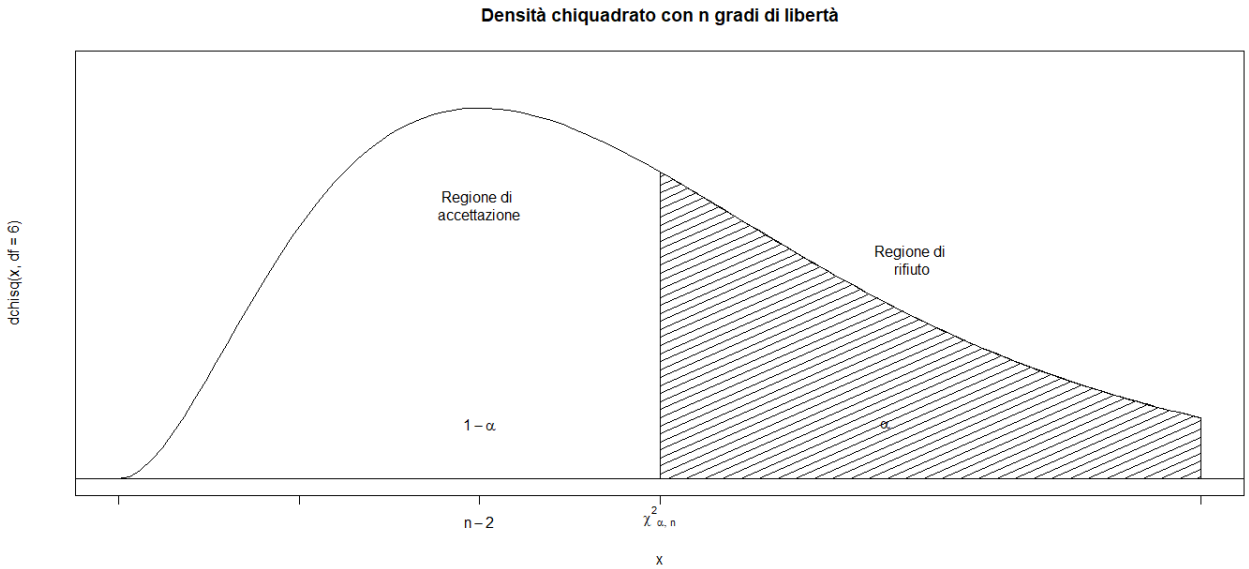


Figura 2.22: Densità chi-quadrato con n gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro

```
> alpha <- 0.05
> mu <- 3
> sigma02 <- 18
> n <- 200
> medcamp <- 3.43899
> varcamp <- 18.16173
> qchisq(1-alpha/2, df=n)
[1] 241.0579
> (n-1)*varcamp/sigma02+n*(medcamp-mu)**2/sigma02
[1] 202.9293
```

Si nota quindi che $X_{\alpha/2,200}$ = 241.0579, X^2 = 202.9293. Il valore osservato X^2 è compreso nella regione di accettazione, si accetta quindi l'ipotesi nulla.

Test unilaterale destro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con valore medio μ noto. Si considerino le ipotesi

$$\mathbf{H}_0 : \sigma^2 \geq \sigma_0^2 \qquad \mathbf{H}_1 : \sigma^2 < \sigma_0^2$$

Il test unilaterale destro ψ di misura α per le ipotesi considerate è il seguente

- si accetti $\mathbf{H_0}$ se $\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma_0}\right)^2 > \chi^2_{1-\alpha,n}$

- si rifiuti $\mathbf{H_0}$ se $\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma_0}\right)^2 < \chi^2_{1-\alpha,n}$

Nella Figura sottostante è possibile visualizzare la densità chi-quadrato con n gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro.

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n gradi di libertà")
> text(4,0.02, expression(1-alpha))
> text(4,0.10, "Regione di \naccettazione")
> axis(1, c(0,2,4,6,12), c("", expression({chi^2}[list(1-alpha, n)]),
+ expression(n-2), "", ""))
>
> vals <- seq(0,2, length=100)
> x <- c(0, vals, 2, 0)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.2,0.02, expression(alpha/2))
> text(0.5,0.07, "Regione di \nrifiuto")
> box()
```

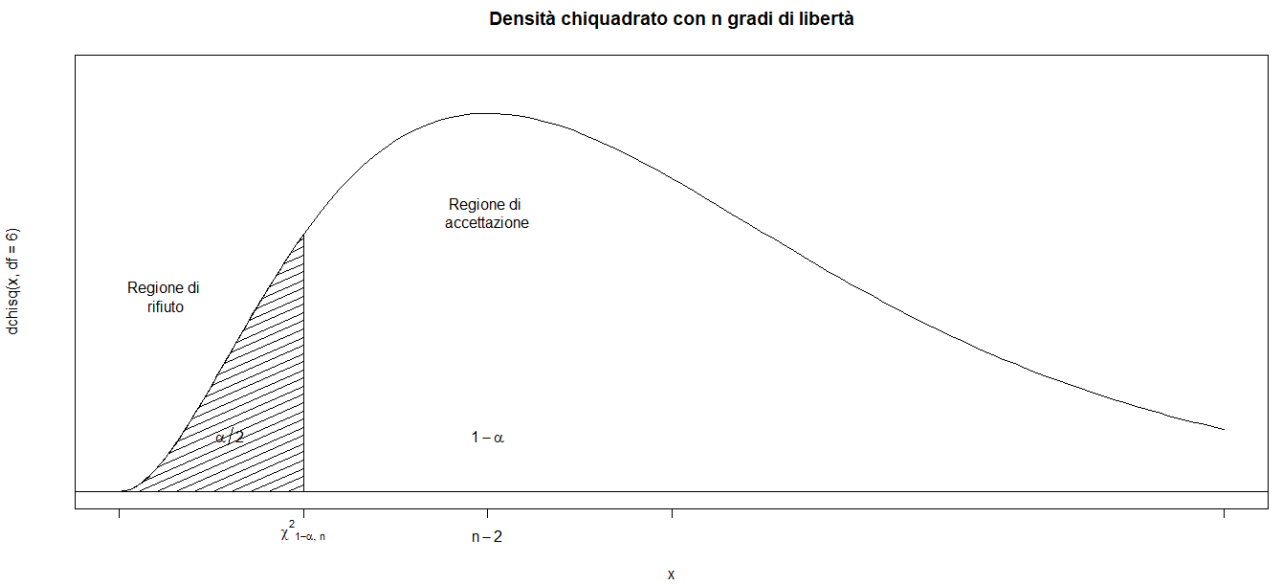


Figura 2.23: Densità chi-quadrato con *n* gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro

```
> alpha <- 0.05
> mu <- 3
> sigmaO2 <- 18
> n <- 200
> medcamp <- 3.43899
> varcamp <- 18.16173
> qchisq(alpha/2, df=n)
[1] 162.728
> (n-1)*varcamp/sigmaO2+n*(medcamp-mu)**2/sigmaO2
[1] 202.9293
```

Si nota quindi che $X_{1-\alpha/2,200} = 162.728$, $X^2 = 202.9293$. Il valore osservato X^2 è compreso nella regione di accettazione, si accetta quindi l'ipotesi nulla.

4. Test su σ^2 con valore medio μ non noto

Sia $X_1, X_2, ..., X_n$ un campione casuale estratto da una popolazione normale con valore medio non noto μ.

Test bilaterale: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza *n* estratto da una popolazione normale con entrambi i parametri non noti. Si considerino le ipotesi:

$$\mathbf{H}_0 : \sigma^2 = \sigma_0^2, \qquad \mathbf{H}_1 : \sigma^2 \neq \sigma_0^2$$

Entrambe le ipotesi sono composte. Quando $\mathbf{H}_0$ è vera, in analogia a quanto visto per gli intervalli di confidenza, gioca un ruolo fondamentale la variabile aleatoria

$$Q_n = \frac{(n-1)\,S_n^2}{\sigma^2} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2.$$

che è distribuita con legge chi-quadrato con $n - 1$ gradi di libertà. Il test bilaterale ψ di misura α per le ipotesi considerate è il seguente:

- si rifiuti $\mathbf{H_0}$ se $\dfrac{(n-1)\,s_n^2}{\sigma_0^2} < \chi^2_{1-\alpha/2,n-1}$ oppure $\dfrac{(n-1)\,s_n^2}{\sigma_0^2} > \chi^2_{\alpha/2,n-1}$

- si accetti $\mathbf{H_0}$ se $\chi^2_{1-\alpha/2,n-1} < \dfrac{(n-1)\,s_n^2}{\sigma_0^2} < \chi^2_{\alpha/2,n-1}$

Nella Figura sottostante è possibile visualizzare la densità chi-quadrato con *n* - 1 gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale.

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n-1 gradi di libertà")
> text(4,0.02, expression(1-alpha))
> text(4,0.10, "Regione di \naccettazione")
> axis(1, c(0,2,4,6,12), c("", expression({chi^2}[list(1-alpha/2, n-1)]),
+ expression(n-3), expression({chi^2}[list(alpha/2, n-1)]), ""))
>
> vals <- seq(0,2, length=100)
> x <- c(0, vals, 2, 0)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> vals <- seq(6,12,length=100)
> x <- c(6, vals, 12, 6)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.2,0.02, expression(alpha/2))
> text(0.5,0.07, "Regione di \nrifiuto")
> text(8.5,0.02, expression(alpha/2))
> text(8.8,0.08, "Regione di \nrifiuto")
> box()
```

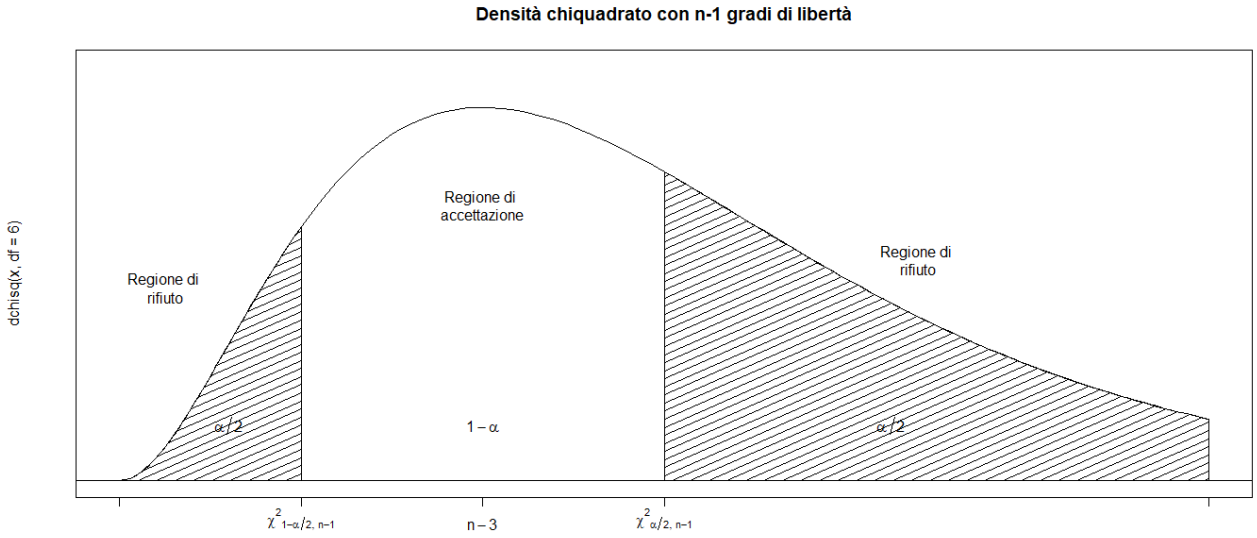


Figura 2.24: Densità chi-quadrato con *n* – 1 gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test bilaterale

```
> alpha <- 0.05
> sigma02 <- 18
> n <- 200
> varcamp <- 18.16173
> qchisq(alpha/2, df=n-1)
[1] 161.8262
> qchisq(1-alpha/2, df=n-1)
[1] 239.9597
> (n-1)*varcamp/sigma02
[1] 200.788
```

Si nota quindi che $X_{\alpha/2,n-1}$ = 161.8262, $X_{1-\alpha/2,n-1}$ = 239.9597, X^2 = 200.788. Il valore osservato X^2 è compreso nella regione di accettazione, si accetta quindi l'ipotesi nulla.

Test unilaterale sinistro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con entrambi i parametri non noti. Si considerino le ipotesi:

$$\mathbf{H}_0 : \sigma^2 \leq \sigma_0^2 \qquad \mathbf{H}_1 : \sigma^2 > \sigma_0^2$$

Il test unilaterale sinistro ψ di misura α per le ipotesi considerate è il seguente:

- si rifiuti $\mathbf{H_0}$ se $\dfrac{(n-1)s_n^2}{\sigma_0^2} > \chi^2_{\alpha,n-1}$

- si accetti $\mathbf{H_0}$ se $\dfrac{(n-1)s_n^2}{\sigma_0^2} < \chi^2_{\alpha,n-1}$

Nella Figura sottostante è possibile visualizzare la densità chi-quadrato con n - 1 gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro.

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n-1 gradi di libertà")
> text(4,0.02, expression(1-alpha))
> text(4,0.10, "Regione di \naccettazione")
> axis(1, c(0,2,4,6,12), c("", "",
+ expression(n-3), expression({chi^2}[list(alpha, n-1)]), ""))
>
> vals <- seq(6,12,length=100)
> x <- c(6, vals, 12, 6)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(8.5,0.02, expression(alpha))
> text(8.8,0.08, "Regione di \nrifiuto")
> box()
```

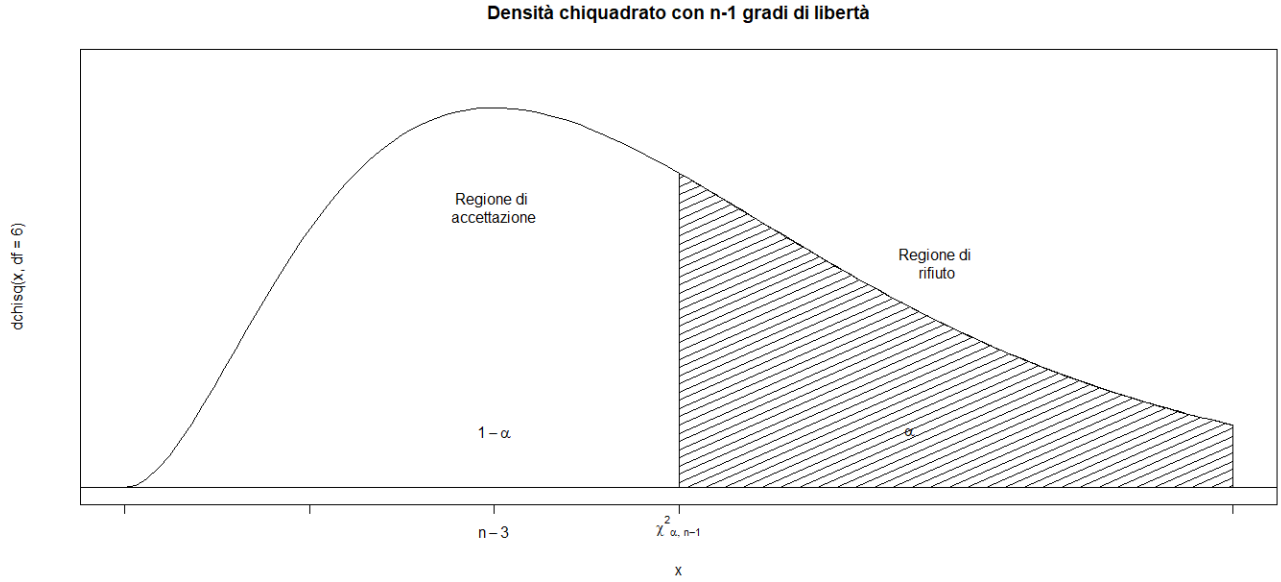


Figura 2.25: Densità chi-quadrato con n – 1 gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale sinistro

```
> alpha <- 0.05
> sigma02 <- 18
> n <- 200
> varcamp <- 18.16173
> qchisq(1-alpha/2, df=n-1)
[1] 239.9597
> (n-1)*varcamp/sigma02
[1] 200.788
```

Si nota quindi che $X_{\alpha/2,n-1}$ = 239.9597, X^2 = 200.788. Il valore osservato X^2 è compreso nella regione di accettazione, si accetta quindi l'ipotesi nulla.

Test unilaterale destro: Sia $(x_1, x_2, ..., x_n)$ un campione osservato di ampiezza n estratto da una popolazione normale con entrambi i parametri non noti. Si considerino le ipotesi:

$$\mathbf{H}_0 : \sigma^2 \geq \sigma_0^2 \qquad \mathbf{H}_1 : \sigma^2 < \sigma_0^2$$

Il test unilaterale destro ψ di misura α per le ipotesi considerate è il seguente

- si rifiuti $\mathbf{H_0}$ se $\dfrac{(n-1)s_n^2}{\sigma_0^2} < \chi^2_{1-\alpha,n-1}$

- si accetti $\mathbf{H_0}$ se $\dfrac{(n-1)s_n^2}{\sigma_0^2} > \chi^2_{1-\alpha,n-1}$

Nella Figura sottostante è possibile visualizzare la densità chi-quadrato con n - 1 gradi di libertà e sono sono riportate le zone di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro.

```
> curve(dchisq(x, df=6), from=0, to=12, axes=FALSE, ylim=c(0,0.15),
+ main="Densità chiquadrato con n-1 gradi di libertà")
> text(4,0.02, expression(1-alpha))
> text(4,0.10, "Regione di \naccettazione")
> axis(1, c(0,2,4,6,12), c("", expression({chi^2}[list(1-alpha/2, n-1)]),
+ expression(n-3), "", ""))
>
> vals <- seq(0,2, length=100)
> x <- c(0, vals, 2, 0)
> y <- c(0, dchisq(vals, df=6),0,0)
> polygon(x,y, density=20, angle=45)
>
> abline(h=0)
> text(1.2,0.02, expression(alpha))
> text(0.5,0.07, "Regione di \nrifiuto")
> box()
```

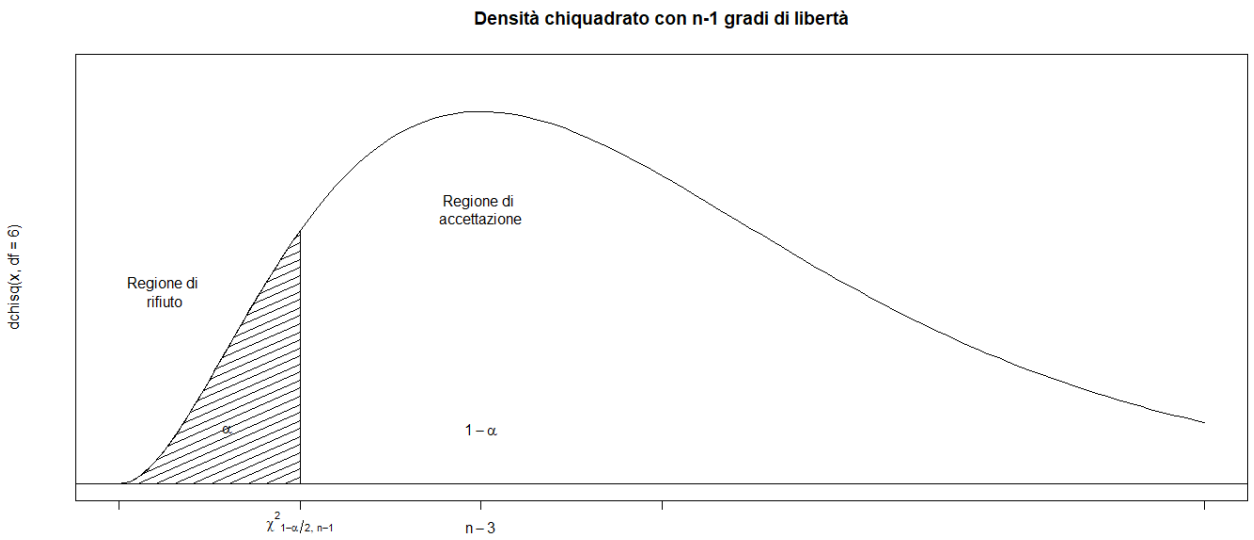


Figura 2.26: Densità chi-quadrato con n – 1 gradi di libertà e regioni di accettazione e di rifiuto dell'ipotesi nulla per il test unilaterale destro

```
> alpha <- 0.05
> sigma02 <- 18
> n <- 200
> varcamp <- 18.16173
> qchisq(alpha/2, df=n-1)
[1] 161.8262
> (n-1)*varcamp/sigma02
[1] 200.788
```

Si nota quindi che $X_{1-\alpha/2,n-1}$ = 161.8262, X^2 = 200.788. Il valore osservato X^2 è compreso nella regione di accettazione, si accetta quindi l'ipotesi nulla.

2.7 Criterio del chi-quadrato

Ci concentriamo ora sul *criterio di verifica delle ipotesi del chi-quadrato*. In molti problemi reali, si desidera verificare se il campione osservato può essere stato estratto da una popolazione descritta da una variabile aleatoria X con funzione di distribuzione $F_X(x)$. A questo scopo, utilizzeremo il **criterio di verifica delle ipotesi del chi–quadrato**, detto anche test del chi–quadrato o test del buon adattamento.

2.7.1 Criterio del chi-quadrato bilaterale

Con il criterio del chi–quadrato si desidera verificare l'ipotesi che un certa popolazione, descritta da una variabile aleatoria X, sia caratterizzata da una funzione di distribuzione $F_X(x)$, con k parametri non noti da stimare.
Denotando con $\mathbf{H}_0$ l'ipotesi soggetta a verifica (*ipotesi nulla*) e con $\mathbf{H}_1$ *l'ipotesi alternativa*, il test chi–quadrato di misura α mira a verificare l'ipotesi nulla

$\mathbf{H_0}$: *X ha una funzione di distribuzione $F_X(x)$ (avendo stimato k parametri non noti in base al campione)*

in alternativa all'ipotesi

$\mathbf{H_1}$: *X non ha una funzione di distribuzione $F_X(x)$,*

dove α è la probabilità massima di rifiutare l'ipotesi nulla quando essa è vera.

Occorre determinare un test ψ di misura α che permetta di determinare una regione di accettazione e di rifiuto dell'ipotesi nulla.
Suddividiamo l'insieme dei valori che la variabile aleatoria X può assumere in r sottoinsiemi I_1, I_2, ..., I_r (classi o categorie) in modo che risulti essere uguale a p_i la probabilità che, secondo la distribuzione ipotizzata, la variabile aleatoria assuma un valore appartenente a I_i, ossia

$$p_i = P(X \in I_i) \qquad (i = 1, 2, \ldots, r).$$

Si estrae poi un campione x_1, x_2, ..., x_n di ampiezza n e si osservano le frequenze assolute n_1, n_2, ..., n_r con cui gli n elementi si distribuiscono nei rispettivi insiemi I_1, I_2, ..., I_r. Quindi n_i rappresenta il *numero degli elementi del campione* che cadono nell'intervallo I_i (i = 1, 2, ..., r). È chiaro che

$$p_i \geq 0 \quad (i = 1, 2, \ldots, r), \qquad \sum_{i=1}^{r} p_i = 1;$$

$$n_i \geq 0 \quad (i = 1, 2, \ldots, r), \qquad \sum_{i=1}^{r} n_i = n.$$

Il numero medio di elementi che cadono nell'intervallo I_i è $n\,p_i$.
Si calcola poi la quantità

$$\chi^2 = \sum_{i=1}^{r} \left(\frac{n_i - n\,p_i}{\sqrt{n\,p_i}} \right)^2.$$

Il criterio chi-quadrato si basa sulla statistica

$$Q = \sum_{i=1}^{r} \left(\frac{N_i - n\,p_i}{\sqrt{n\,p_i}} \right)^2$$

dove N_i è la variabile aleatoria che descrive il numero degli elementi del campione casuale X_1, X_2, ..., X_n (costituito da n variabili aleatorie osservabili, indipendenti e identicamente distribuite con la stessa legge di probabilità $F_X(x)$ della popolazione) che cadono nell'intervallo I_i (i = 1, 2, ..., r).

Se la variabile aleatoria X ha una funzione di distribuzione $F_X(x)$ con k parametri non noti, si può dimostrare che per n sufficientemente grande la funzione di distribuzione della statistica Q è approssimabile con la funzione di distribuzione chi–quadrato con $r - k - 1$ gradi di libertà.
Si sottrae 1 da r a causa della prima delle condizioni secondo la quale se conosciamo $r - 1$ delle probabilità p_i la rimanente probabilità può essere univocamente determinata e si sottrae k poiché si suppone che siano k i parametri indipendenti non noti sostituiti da stime.

Per garantire che ogni classe contenga in media almeno 5 elementi, si ritiene valida l'approssimazione se risulta

$$\min(np_1, np_2, \ldots, np_r) \geq 5.$$

Si giunge alla definizione del *test chi–quadrato bilaterale*.

Proposizione *Per un campione sufficientemente numeroso di ampiezza n, il test chi–quadrato bilaterale di misura α è il seguente:*

- *si rifiuti l'ipotesi* H_0 *se* $\chi^2 < \chi^2_{1-\alpha/2,r-k-1}$ *oppure* $\chi^2 > \chi^2_{\alpha/2,r-k-1}$
- *si accetti l'ipotesi* H_0 *se* $\chi^2_{1-\alpha/2,r-k-1} < \chi^2 < \chi^2_{\alpha/2,r-k-1}$,

Considerando il campione generato in precedenza formato da n=200

```
> n<- length(campione)
> n
[1] 200
> m<-mean(campione)
> m
[1] 3.43899
> d<-sd(campione)
> d
[1] 4.261658
```

Notiamo quindi che la media campionaria $\bar{x}$ = 3.43899 e la deviazione standard s = 4.261658. Applicando il test del chi-quadrato di misura α = 0.05, vogliamo verificare se la popolazione da cui proviene il campione possa essere descritta da una variabile aleatoria X di densità normale

$$f_X(x) = \frac{1}{\sigma\sqrt{2\,\pi}}\,\exp\left\{-\frac{(x-\mu)^2}{2\,\sigma^2}\right\}, \qquad x \in \mathbb{R} \qquad (\mu \in \mathbb{R}, \sigma > 0)$$

Supponiamo di suddividere l'insieme dei valori che tale variabile aleatoria normale X può assumere in r = 5 sottoinsiemi $I_1, I_2, ..., I_5$ in modo che risulti essere uguale a p_i = 0.2 la probabilità che X assuma un valore appartenente a I_i (i = 1, 2, ..., 5). La condizione che la classe contenga almeno 5 elementi è verificata essendo $n\,p_i = 200 \cdot 0.2 = 40 \geq 5$. Ricordando che uno stimatore di μ è la media campionaria e uno stimatore di σ^2 è la varianza campionaria, utilizzando i quantili della distribuzione normale possiamo determinare i sottoinsiemi $I_1, I_2, ..., I_5$

```
> a <- numeric(4)
> for(i in 1:4)
+ a[i] <- qnorm(0.2*i, mean=m, sd=d)
> a
[1] -0.147712  2.359311  4.518669  7.025692
```

Gli intervalli $I_1, I_2, ..., I_5$ sono:

- $I_1 = (-\infty, -0.147712)$,
- $I_2 = [-0.147712, 2.359311)$
- $I_3 = [2.359311, 4.18669)$
- $I_4 = [4.18669, 7.025692)$
- $I_5 = [7.025692, +\infty)$.

Occorre ora determinare il numero di elementi del campione che cadono negli intervalli $I_1, I_2, ..., I_5$:

```
> r <- 5
> nint <- numeric(r)
> nint[1] <- length(which(campione < a[1]))
> nint[2] <- length(which((campione >= a[1]) & (campione < a[2])))
> nint[3] <- length(which((campione >= a[2]) & (campione < a[3])))
> nint[4] <- length(which((campione >= a[3]) & (campione < a[4])))
> nint[5] <- length(which((campione >= a[4])))
> nint
[1] 36 46 35 44 39
> sum(nint)
[1] 200
```

Ne segue che $n_1 = 36$, $n_2 = 46$, $n_3 = 35$, $n_4 = 44$, $n_5 = 39$. Calcoliamo ora X^2

```
> chi2<-sum(((nint-n*0.2)/sqrt(n*0.2))^2)
> chi2
[1] 2.35
```

Osserviamo quindi che $X^2 = 2.35$.
La distribuzione normale ha due parametri non noti (μ, σ^2) e quindi $k = 2$.
Pertanto, la funzione di distribuzione della statistica Q è approssimabile con la funzione di distribuzione chi–quadrato con $r - k - 1 = 2$ gradi di libertà.

Occorre quindi calcolare $X^2_{\alpha/2,2}$ e $X^2_{1-\alpha/2,2}$ con $\alpha = 0.05$.

```
> k<-2
> alpha <-0.05
> qchisq(alpha/2, df=r-k-1)
[1] 0.05063562
> qchisq(1-alpha/2, df=r-k-1)
[1] 7.377759
```

Da cui si evince che $X^2_{1-\alpha/2,r-k-1} = 7.377759$ e $X^2_{\alpha/2,2} = 0.05063562$. Siccome $0.05063562 < X^2 < 7.377759$, l'ipotesi $\mathbf{H}_0$ di popolazione normale è accettata.