Università degli Studi di Salerno

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

Statistica ed Analisi Dati

Mobilità e condizioni stradali in Italia

Indagine statistica sul numero di famiglie con problemi stradali nella zona in cui abitano per ogni regione italiana

Docente

Prof.ssa

Nobile Amelia Giuseppina

Studente

De Rosa Gerardo

Matr. 0522500762

Anno Accademico 2019-2020

Sommario

Introduzione 4

1 Tabelle e Grafici 5

1.1 Distribuzioni di Frequenza 5

1.2 Rappresentazioni Grafiche 6

1.2.1 Diagramma di Pareto 6

1.2.2 Grafici a Barre 9

1.2.3 Grafici a Torta 11

1.3 Istogrammi 15

1.4 Boxplot 18

1.5 Scatterplot 21

2 Statistica Descrittiva Univariata 22

2.1 Funzione di Distribuzione Empirica Discreta 22

2.2 Funzione di Distribuzione Empirica Continua 23

2.3 Indici di Sintesi 25

2.3.1 Media, mediana e moda 25

2.3.3 Percentili Decili e Quartili 26

2.3.4 Variazione, deviazione standard e coefficiente di variazione 27

2.3.5 Forma di una Distribuzione di Frequenza 29

3 Statistica descrittiva bivariata con R 30

3.1 Covarianza e correlazione campionaria 32

3.2 Regressione lineare semplice 35

3.2.1 Residui 36

3.3 Regressione lineare multipla 40

3.3.1 Residui 41

3.4 Regressione non lineare 43
3.4.1 Regressione Polinomiale 43
4 Analisi dei Cluster 45
4.1 Distanza e similarità 46
4.1.1 Metrica Euclidea 47
4.1.2 Altre metriche 48
4.1.3 Misure di similarità 48
4.2 Misura di non omogeneità totale 49
4.3 Metodi di ottimizzazione 50
4.4 Metodi gerarchici 50
4.4.1 Metodo del legame singolo 52
4.4.2 Metodo del legame completo 54
4.4.3 Metodo del legame medio 56
4.4.5 Metodo della mediana 59
4.5 Analisi del Dendrogramma 60
4.5.1 Disegnare rettangoli che evidenziano i cluster 60
4.5.2 Inserire gli individui nei cluster 62
4.5.3 Misure di sintesi associate ai cluster 63
4.5.4 Misure di non omogeneità statistiche 64
4.6 Metodi non gerarchici 66

Introduzione

L'oggetto di questa tesi riguardante concetti di statistica e di analisi dei dati, risulta essere lo studio di un dataset contente informazioni sulla qualità della vita nel contesto urbano delle famiglie italiane; in particolare, i dati prendono in esame le ***difficoltà nel trovare parcheggio***, ***la quantità di traffico presente***, **il** ***collegamento coi mezzi pubblici***, **l'*adeguatezza dell'illuminazione* e del *manto stradale***, ***l'inquinamento dell'aria***, **il** ***rumore*** **ed infine il** ***rischio di criminalità.***

L'indagine si pone come scopo quello di conoscere ed analizzare i problemi affrontati dagli utenti, quanto sono soddisfatti della zona in cui vivono e delle condizioni delle strade che ogni giorno popolano; l'idea alla base della tesi nasce da una ricerca effettuata dall'ISTAT che fa parte di un sistema integrato di indagini sociali, le Indagini Multiscopo sulle famiglie, il cui scopo è quello di rilevare informazioni fondamentali relative alla vita quotidiana degli individui e delle famiglie.

Di seguito è illustrato il dataset oggetto di studio, prelevato dal sito ufficiale dell'ISTAT;

Dataset: Aspetti della vita quotidiana - Famiglie - 2018								
Misura	Valori in Migliaia							
Tipo dato	Famiglie per presenza di alcuni problemi nella zona in cui abitano							
Territorio	*Difficoltà di parcheggio*	*Difficoltà di collegamento con mezzi pubblici*	*Traffico*	*Scarsa illuminazione stradale*	*Cattive condizioni stradali*	*Inquinamento dell'aria*	*Rumore*	*Rischio di criminalità*
Piemonte	708	647	700	539	1037	228	168	109
Valle d'Aosta	14	17	12	11	25	2	4	1
Liguria	403	185	313	261	411	65	75	46
Lombardia	1589	1301	1767	1137	2089	718	486	292
Trentino Alto Adige	134	86	133	69	109	20	25	15
Provincia Autonoma Bolzano	81	31	81	34	57	13	18	9
Provincia Autonoma Trento	54	54	52	34	52	7	7	6
Veneto	562	635	805	584	832	232	182	119
Friuli-Venezia Giulia	145	126	186	111	195	42	42	14
Emilia-Romagna	548	517	686	556	982	209	151	119
Toscana	601	529	649	541	898	161	169	125
Umbria	87	112	118	126	282	40	25	31
Marche	162	184	188	188	334	23	32	23
Lazio	1209	924	1385	1175	1987	375	351	285
Abruzzo	149	141	163	144	290	28	35	32
Molise	39	41	31	45	67	5	7	2
Campania	955	1070	889	880	1274	321	257	191
Puglia	613	466	642	462	923	212	171	116
Basilicata	67	88	57	77	135	14	12	4
Calabria	209	336	228	283	427	30	39	25
Sicilia	812	760	860	882	1199	226	201	85
Sardegna	247	222	241	247	439	26	50	28

Figura 1 - Dataset utilizzato

1 Tabelle e Grafici

In questo paragrafo procederemo alla costruzione di grafici e tabelle di frequenza tramite il linguaggio R, utilizzando i dati contenuti all'interno del dataset di cui sopra.

Il primo passo per calcolare tutto ciò, è stato importare il dataset nell'ambiente di RStudio, ciò è stato possibile tramite il comando:

```
library(readxl)
DatasetStrada <- read_excel("C:/Users/FauxL/Desktop/DatasetStrada.xlsx",
                            skip = 3)
View(DatasetStrada[1:23,1:2])
```

Figura 2 - Import Dataset in R

1.1 Distribuzioni di Frequenza

Di seguito verranno elencati i vari script per il calcolo delle frequenze assolute, relative, cumulate ed i rispettivi risultati, che si riferiscono alla tipologia di problemi presenti presa in osservazione dal nostro dataset;

```
> colnames(frequenze_assolute) <- c("Difficoltà di parcheggio", "Difficoltà di collegamento con mezzi pubblici",
+ "Traffico", "Scarsa illuminazione stradale", "Cattive condizioni stradali", "Inquinamento dell'aria",
+ "Rumore", "Rischio di criminalità")
>
> for ( i       in      1:8) {
+ frequenze_assolute[1,i] <- sum(DatasetStrada [ , i+1 ]) }
> frequenze_assolute
     Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici Traffico Scarsa illuminazione stradale
[1,]                     9388                                         8472    10186                           8386
     Cattive condizioni stradali Inquinamento dell'aria Rumore Rischio di criminalità
[1,]                       14044                   2997   2507                   1677
```

Figura 3 - Frequenza Assoluta

Per calcolare la frequenza cumulativa, utilizziamo la funzione *cumsum()*:

```
> frequenze_cumulative <- array(dim=c(1,8))
> frequenze_cumulative <- cumsum(frequenze_assolute[1,1:8])
> frequenze_cumulative
                     Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici                                   Traffico
                                         9388                                        17860                                      28046
               Scarsa illuminazione stradale                   Cattive condizioni stradali                       Inquinamento dell'aria
                                        36432                                        50476                                      53473
                                       Rumore                        Rischio di criminalità
                                        55980                                        57657
```

Figura 4 - Frequenza Cumulativa

Per ottenere invece la frequenza relativa si utilizza la funzione *prop.table():*

```
> frequenze_relative <- prop.table(frequenze_assolute[1,1:8])
> frequenze_relative
                    Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici                                  Traffico
                                  0.16282498                                  0.14693793                                0.17666545
              Scarsa illuminazione stradale                   Cattive condizioni stradali                   Inquinamento dell'aria
                                  0.14544635                                  0.24357840                                0.05197981
                                     Rumore                         Rischio di criminalità
                                  0.04348128                                  0.02908580
```

Figura 5 - Frequenza Relativa

Infine per avere la frequenza relativa cumulata utilizziamo la funzione *cumsum()* sull'output della funzione *prop.table():*

```
> frequenze_relative_cumulate <- cumsum(prop.table(frequenze_assolute[1,1:8]))
> frequenze_relative_cumulate
                    Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici                                  Traffico
                                   0.1628250                                   0.3097629                                 0.4864284
              Scarsa illuminazione stradale                   Cattive condizioni stradali                   Inquinamento dell'aria
                                   0.6318747                                   0.8754531                                 0.9274329
                                     Rumore                         Rischio di criminalità
                                   0.9709142                                   1.0000000
```

Figura 6 - Frequenza Relativa Cumulata

1.2 Rappresentazioni Grafiche

In questo paragrafo si procederà con l'illustrare i dati tramite alcune rappresentazioni grafiche, tra cui: grafici a barre, grafici a torta, istogrammi e diagrammi di Pareto.

1.2.1 Diagramma di Pareto

Nel 1897 l'economista italiano Vilfredo Pareto ispirò la "legge 80/20", una legge empirica nota anche con il nome di **principio di Pareto**, sintetizzabile nell'affermazione: "la maggior parte degli effetti `e dovuto ad un numero ristretto di cause. Secondo la legge 80/20, l'80% dei risultati dipende dal 20% delle cause". Il diagramma di Pareto consiste in un **diagramma a barre verticali con le modalità ordinate in ordine decrescente rispetto alle loro frequenza relativa;** inoltre le frequenze relative sono visualizzate anche nella loro forma cumulata, mediante una sequenza di segmenti crescenti; esso permette di determinare le poche variabili che influenzano in modo significativo i risultati finali del fenomeno. L'analisi basata sul diagramma di Pareto è suddivisa nelle seguenti fasi:

Decidere come **classificare** i dati; rilevare i dati ed **ordinarli**; disegnare il diagramma; costruire la linea cumulativa; aggiungere le informazioni di base.

Dopo aver deciso di creare un grafico di Pareto riguardante la somma di tutte le occorrenze di problemi stradali, raggruppate per regione, procediamo quindi al calcolo delle frequenze relative e a quello delle relative cumulate, senza dimenticare di ordinare in modo crescente in base alle occorrenze, cosi da adempiere ai primi due punti di cui sopra; di seguito è illustrato il codice:

```
## DIAGRAMMA DI PARETO

Dati<- rowSums(DatasetStrada[,labelss])

FreqRel <- prop.table(Dati)
ParetoTable<-data.frame(Regioni,Dati,FreqRel,FreqRelCum)
ParetoTable<- ParetoTable[rev(order(ParetoTable[,2])),]
FreqRelCum <-  paste (format ( cumsum ( unlist(ParetoTable[,3]) ) * 100 , digits = 2) , "%" )
ParetoTable[,4] <- (FreqRelCum)
```

Figura 7 - Creazione Tabella per Grafo di Pareto

La prima parte del codice ivi riportato, crea una lista formata dalla somma di tutti i problemi per ogni regione; di seguito viene calcolata la frequenza relativa e poi cumulata, infine il data frame cosi creato viene ordinato in maniera decrescente in base alla quantità di occorrenze di problematiche stradali.

Regioni	Dati	FreqRel	FreqRelCum
Lombardia	9379	0.162668887	16 %
Lazio	7691	0.133392303	30 %
Campania	5837	0.101236623	40 %
Sicilia	5025	0.087153338	48 %
Piemonte	4136	0.071734568	56 %
Veneto	3951	0.068525938	62 %
Emilia-Romagna	3768	0.065351995	69 %
Toscana	3673	0.063704320	75 %
Puglia	3605	0.062524932	82 %
Liguria	1759	0.030508004	85 %
Calabria	1577	0.027351406	87 %
Sardegna	1500	0.026015922	90 %
Marche	1134	0.019668037	92 %
Abruzzo	982	0.017031757	94 %
Friuli-Venezia Giulia	861	0.014933139	95 %
Umbria	821	0.014239381	97 %
Trentino Alto Adige	591	0.010250273	98 %
Basilicata	454	0.007874152	98 %
Prov.Aut. Bolzano	324	0.005619439	99 %
Prov. Aut. Trento	266	0.004613490	99 %
Molise	237	0.004110516	100 %
Valle d'Aosta	86	0.001491580	100 %

Figura 8 - Tabella Di Pareto

Dalla tabella è evidente come la maggior quantità di problemi sia presente in Lombardia, seguita da Lazio e Campania, mentre le regioni meno “problematiche” risultano la Valle D’Aosta il Molise e le province autonome di Bolzano e Trento.

```
x <- barplot ( unlist(ParetoTable[,3]), ylim = c (0 , 1.05) , main = " Diagramma di Pareto " , col =1:22)
lines (x, cumsum (unlist(ParetoTable[,3])) , type = "b" , pch = 16)
text ( x - 0.2 , cumsum ( unlist(ParetoTable[,3]) ) + 0.03 , paste ( format ( cumsum ( unlist(ParetoTable[,3])
) * 100 , digits = 2) , "%" ))
text ( x , par("usr")[3] , labels = ParetoTable[,1] , srt = 40 , adj= c ( 1 , 1.1) , xpd = TRUE)
```

Figura 9 - Codice per Grafo di Pareto

Il codice sopra riportato è quello atto alla creazione del grafo di Pareto, in R non è presente una vera e propria funziona atta a questo scopo, ma è possibile ottenerlo combinando varie funzioni: abbiamo innanzitutto usato la funzione ***barplot()*** per rappresentare il diagramma a barre; in seguito, abbiamo usato il comando a basso livello ***lines()*** per rappresentare le linee che collegano i punti riguardanti i dati, dove i **punti** sono i risultati delle **somma cumulative delle frequenze relative** del vettore *FreqRel.* Infine, con il comando ***text()*** abbiamo inserito il testo in corrispondenza di ogni punto indicante le somme cumulate delle frequenze relative e le regioni italiane.

In seguito verranno prese in esame in dettaglio le varie problematiche, le cui frequenze verranno illustrate tramite grafici a barra.

1.2.2 Grafici a Barre

In R si usa l'istruzione ***barplot(x)*** per produrre un grafico a barre con i valori del vettore x sulla base delle frequenze. Il grafico a barre si ottiene disponendo sull'asse verticale le **modalità assunte dalla variabile *x*** e sull'asse orizzontale le **regioni italiane**. Il grafico è poi intersecato da una linea tratteggiata rappresentante la ***media*** dei valori assunti. Per rappresentare la linea è stata utilizzata la funzione ***abline()***, mentre per personalizzare le etichette delle regioni, la funzione ***text()***.Tutti i grafici prodotti nel ciclo scandito dal ***repeat{}*** e terminato dal break, sono stati salvati in un pdf tramite l'apposita funzione ***pdf();*** il codice completo utilizzato per la produzione dei grafici è quello rappresentato nella pagina seguente:

```
StreetPlot <- function() {

  j <- 1
  pdf ("GraficiABarreRegioni.pdf" )
  repeat {
  if ( j == 9 ) {
  dev.off ()
  break
  }
  valori <- as.numeric(unlist(DatasetStrada[,j+1]))
    x <- barplot ( valori , ylim = c (0,2500) ,cex.axis=10, main = labelss[j] ,
                   col = rainbow ( 22 ) , axes = FALSE, axisnames = FALSE)
    abline (h = mean(valori) , col = "blue" , lty = 2 )
    text ( x , par("usr")[3] , labels = regioni , srt = 35 , adj= c ( 1 , 1.1) , xpd = TRUE)
    axis (2)
    j <- j + 1
    }

}
```

Figura 10 – Codice Grafici a barra (Frequenza dei diversi Problemi x Regione)

Di seguito sono mostrati i grafi risultanti:

Rumore
0
500
1000
1500
2000
2500
Piemonte
Valle d'Aosta
Liguria
Lombardia
Trentino Alto Adige
Prov. Aut. Bolzano
Prov. Aut. Trento
Veneto
Friuli-Venezia Giulia
Emilia-Romagna
Toscana
Umbria
Marche
Lazio
Abruzzo
Molise
Campania
Puglia
Basilicata
Calabria
Sicilia
Sardegna

Rischio di criminalità
0
500
1000
1500
2000
2500
Piemonte
Valle d'Aosta
Liguria
Lombardia
Trentino Alto Adige
Prov. Aut. Bolzano
Prov. Aut. Trento
Veneto
Friuli-Venezia Giulia
Emilia-Romagna
Toscana
Umbria
Marche
Lazio
Abruzzo
Molise
Campania
Puglia
Basilicata
Calabria
Sicilia
Sardegna

Traffico
0
500
1000
1500
2000
2500
Piemonte
Valle d'Aosta
Liguria
Lombardia
Trentino Alto Adige
Prov. Aut. Bolzano
Prov. Aut. Trento
Veneto
Friuli-Venezia Giulia
Emilia-Romagna
Toscana
Umbria
Marche
Lazio
Abruzzo
Molise
Campania
Puglia
Basilicata
Calabria
Sicilia
Sardegna

Difficoltà di parcheggio
0
500
1000
1500
2000
2500
Piemonte
Valle d'Aosta
Liguria
Lombardia
Trentino Alto Adige
Prov. Aut. Bolzano
Prov. Aut. Trento
Veneto
Friuli-Venezia Giulia
Emilia-Romagna
Toscana
Umbria
Marche
Lazio
Abruzzo
Molise
Campania
Puglia
Basilicata
Calabria
Sicilia
Sardegna

Difficoltà di collegamento con mezzi pubblici
0
500
1000
1500
2000
2500
Piemonte
Valle d'Aosta
Liguria
Lombardia
Trentino Alto Adige
Prov. Aut. Bolzano
Prov. Aut. Trento
Veneto
Friuli-Venezia Giulia
Emilia-Romagna
Toscana
Umbria
Marche
Lazio
Abruzzo
Molise
Campania
Puglia
Basilicata
Calabria
Sicilia
Sardegna

Scarsa illuminazione stradale
0
500
1000
1500
2000
2500
Piemonte
Valle d'Aosta
Liguria
Lombardia
Trentino Alto Adige
Prov. Aut. Bolzano
Prov. Aut. Trento
Veneto
Friuli-Venezia Giulia
Emilia-Romagna
Toscana
Umbria
Marche
Lazio
Abruzzo
Molise
Campania
Puglia
Basilicata
Calabria
Sicilia
Sardegna

Da i seguenti grafici possiamo evincere come la maggior quantità di problemi delle strade italiane, sia legate proprio alle condizioni stradali; altro dato che si evince è come la maggior concentrazione di problemi sia presente principalmente in 3 regioni, *Lombardia, Campania e Lazio;* ciò è probabilmente dato dalla maggior grandezza e popolosità delle regioni.

1.2.3 Grafici a Torta

Sono stati utilizzati i grafici a torta per rappresentare la composizione per Regione della presenza di disagi stradali, per comodità di visualizzazione, i valori sono espressi in percentuale; La funzione che ho utilizzato per rappresentare la torta è ***pie3D(),*** presente nella libreria ***plotrix,*** il codice per il resto è simile a quello utilizzato in precedenza per la produzione dei grafici a barra ed è di seguito rappresentato:

```
## GRAFICI A TORTA

PieStreetPlot <- function(){
  j <- 1
  pdf("GraficiTorteProblemi.pdf")
  repeat {
    if(j==22){
      dev.off()
      break
    }
    valori <- as.numeric(unlist(DatasetStrada[j,2:9]))
    percentage = paste0(labelss, " ", round(valori / sum(valori) * 100, 1), "%")
    x <- pie3D(valori, height = 0.05, explode = 0.05,radius = 0.45, labels = NA,
               col=brewer.pal(8,"PRGn"),theta = 1)
    title(regioni[j], line=-0.1)
    legend(legend=percentage,bty="n","top", fill =brewer.pal(8,"PRGn"))
    j <- j + 1
  }
}
PieStreetPlot()
```

Figura 11 - Codice Grafici a Torta

Di seguito sono illustrati i grafi prodotti:

Liguria

- Difficoltà di parcheggio 22.9%
- Difficoltà di collegamento con mezzi pubblici 10.5%
- Traffico 17.8%
- Scarsa illuminazione stradale 14.8%
- Cattive condizioni stradali 23.4%
- Inquinamento dell'aria 3.7%
- Rumore 4.3%
- Rischio di criminalità 2.6%

Lombardia

- Difficoltà di parcheggio 16.9%
- Difficoltà di collegamento con mezzi pubblici 13.9%
- Traffico 18.8%
- Scarsa illuminazione stradale 12.1%
- Cattive condizioni stradali 22.3%
- Inquinamento dell'aria 7.7%
- Rumore 5.2%
- Rischio di criminalità 3.1%

Trentino Alto Adige

- Difficoltà di parcheggio 22.7%
- Difficoltà di collegamento con mezzi pubblici 14.6%
- Traffico 22.5%
- Scarsa illuminazione stradale 11.7%
- Cattive condizioni stradali 18.4%
- Inquinamento dell'aria 3.4%
- Rumore 4.2%
- Rischio di criminalità 2.5%

Prov.Aut. Bolzano

- Difficoltà di parcheggio 25%
- Difficoltà di collegamento con mezzi pubblici 9.6%
- Traffico 25%
- Scarsa illuminazione stradale 10.5%
- Cattive condizioni stradali 17.6%
- Inquinamento dell'aria 4%
- Rumore 5.6%
- Rischio di criminalità 2.8%

Prov. Aut. Trento

- Difficoltà di parcheggio 20.3%
- Difficoltà di collegamento con mezzi pubblici 20.3%
- Traffico 19.5%
- Scarsa illuminazione stradale 12.8%
- Cattive condizioni stradali 19.5%
- Inquinamento dell'aria 2.6%
- Rumore 2.6%
- Rischio di criminalità 2.3%

Veneto

- Difficoltà di parcheggio 14.2%
- Difficoltà di collegamento con mezzi pubblici 16.1%
- Traffico 20.4%
- Scarsa illuminazione stradale 14.8%
- Cattive condizioni stradali 21.1%
- Inquinamento dell'aria 5.9%
- Rumore 4.6%
- Rischio di criminalità 3%

Friuli−Venezia Giulia

- Difficoltà di parcheggio 16.8%
- Difficoltà di collegamento con mezzi pubblici 14.6%
- Traffico 21.6%
- Scarsa illuminazione stradale 12.9%
- Cattive condizioni stradali 22.6%
- Inquinamento dell'aria 4.9%
- Rumore 4.9%
- Rischio di criminalità 1.6%

Emilia−Romagna

- Difficoltà di parcheggio 14.5%
- Difficoltà di collegamento con mezzi pubblici 13.7%
- Traffico 18.2%
- Scarsa illuminazione stradale 14.8%
- Cattive condizioni stradali 26.1%
- Inquinamento dell'aria 5.5%
- Rumore 4%
- Rischio di criminalità 3.2%

Toscana

- Difficoltà di parcheggio 16.4%
- Difficoltà di collegamento con mezzi pubblici 14.4%
- Traffico 17.7%
- Scarsa illuminazione stradale 14.7%
- Cattive condizioni stradali 24.4%
- Inquinamento dell'aria 4.4%
- Rumore 4.6%
- Rischio di criminalità 3.4%

Umbria

- Difficoltà di parcheggio 10.6%
- Difficoltà di collegamento con mezzi pubblici 13.6%
- Traffico 14.4%
- Scarsa illuminazione stradale 15.3%
- Cattive condizioni stradali 34.3%
- Inquinamento dell'aria 4.9%
- Rumore 3%
- Rischio di criminalità 3.8%

Marche

- Difficoltà di parcheggio 14.3%
- Difficoltà di collegamento con mezzi pubblici 16.2%
- Traffico 16.6%
- Scarsa illuminazione stradale 16.6%
- Cattive condizioni stradali 29.5%
- Inquinamento dell'aria 2%
- Rumore 2.8%
- Rischio di criminalità 2%

Lazio

- Difficoltà di parcheggio 15.7%
- Difficoltà di collegamento con mezzi pubblici 12%
- Traffico 18%
- Scarsa illuminazione stradale 15.3%
- Cattive condizioni stradali 25.8%
- Inquinamento dell'aria 4.9%
- Rumore 4.6%
- Rischio di criminalità 3.7%

Abruzzo

- Difficoltà di parcheggio 15.2%
- Difficoltà di collegamento con mezzi pubblici 14.4%
- Traffico 16.6%
- Scarsa illuminazione stradale 14.7%
- Cattive condizioni stradali 29.5%
- Inquinamento dell'aria 2.9%
- Rumore 3.6%
- Rischio di criminalità 3.3%

Molise

- Difficoltà di parcheggio 16.5%
- Difficoltà di collegamento con mezzi pubblici 17.3%
- Traffico 13.1%
- Scarsa illuminazione stradale 19%
- Cattive condizioni stradali 28.3%
- Inquinamento dell'aria 2.1%
- Rumore 3%
- Rischio di criminalità 0.8%

Campania

- Difficoltà di parcheggio 16.4%
- Difficoltà di collegamento con mezzi pubblici 18.3%
- Traffico 15.2%
- Scarsa illuminazione stradale 15.1%
- Cattive condizioni stradali 21.8%
- Inquinamento dell'aria 5.5%
- Rumore 4.4%
- Rischio di criminalità 3.3%

Puglia

- Difficoltà di parcheggio 17%
- Difficoltà di collegamento con mezzi pubblici 12.9%
- Traffico 17.8%
- Scarsa illuminazione stradale 12.8%
- Cattive condizioni stradali 25.6%
- Inquinamento dell'aria 5.9%
- Rumore 4.7%
- Rischio di criminalità 3.2%

Basilicata

- Difficoltà di parcheggio 14.8%
- Difficoltà di collegamento con mezzi pubblici 19.4%
- Traffico 12.6%
- Scarsa illuminazione stradale 17%
- Cattive condizioni stradali 29.7%
- Inquinamento dell'aria 3.1%
- Rumore 2.6%
- Rischio di criminalità 0.9%

Calabria

- Difficoltà di parcheggio 13.3%
- Difficoltà di collegamento con mezzi pubblici 21.3%
- Traffico 14.5%
- Scarsa illuminazione stradale 17.9%
- Cattive condizioni stradali 27.1%
- Inquinamento dell'aria 1.9%
- Rumore 2.5%
- Rischio di criminalità 1.6%

Dai grafici si evince come nella maggior parte dei casi, la fetta più consistente sia rappresentata dalle **Cattive condizioni Stradali**, con qualche eccezione, presenti ad esempio nelle province autonome di Bolzano e Trento e nel Trentino Alto Adige, dove la preponderanza viene sostituita dalla **Difficoltà di Parcheggio.** La parte più piccola della torta è invece quasi sempre rappresentata dal **Rischio di Criminalità** e del **Rumore.**

1.3 Istogrammi

Gli istogrammi, che si utilizzano per variabili quantitative, sono una particolare rappresentazione grafica di una distribuzione di **frequenza in classi**. Nel caso dei problemi stradali, verrà fatta una suddivisione in classi sull'asse delle ascisse, che indicherà sostanzialmente **gli intervalli del numero di famiglie** disagiate per zone d'Italia e sull'asse delle ordinate la loro frequenza assoluta.

```
## ISTOGRAMMA
InstaPlot <- function() {
  j <- 2
  pdf ("IstogrammiFreqAss.pdf" )
  repeat {
    if ( j == 9 ) {
      dev.off ()
      break
    }
    istogramma <- hist (unlist(DatasetStrada[,j]), freq = TRUE, main = labelss[j-1],
                        ylab = "Frequenza assoluta delle classi", xlab = "Classi")
    j <- j + 1
  }
}
```

Figura 12 - Codice Istogrammi

La funzione che viene impiegata per la realizzazione degli istogrammi è quella ***hist();*** di seguito sarà illustrato il codice ed i relativi grafici prodotti:

Si può notare come più di 10 regioni siano nell'intervallo che va da 0 a 200, 2 nell'intervallo tra 200 e 400, 3 nell'intervallo da 400 a 600, altre 3 nell'intervallo successivo, e infine 1 regione per ciascuno dei 3 intervalli di frequenza finali. Le classi sono state scelte automaticamente da R; in aggiunta utilizzando la funzione ***str()*** possiamo ottenere ulteriori dettagli riguardo il grafo creato:

```
> str(istogramma)
List of 6
 $ breaks  : num [1:8] 0 200 400 600 800 1000 1200 1400
 $ counts  : int [1:7] 11 2 3 3 1 1 1
 $ density : num [1:7] 0.0025 0.000455 0.000682 0.000682 0.000227 ...
 $ mids    : num [1:7] 100 300 500 700 900 1100 1300
 $ xname   : chr "unlist(DatasetStrada[, 3])"
 $ equidist: logi TRUE
 - attr(*, "class")= chr "histogram"
```

Figura 13 - Dettagli Istogramma

La funzione ci mostra: l'attributo ***$breaks*** che si riferisce ai diversi intervalli, ***$counts*** che indica il numero di zone che appartengono ad un determinato intervallo, ***$density*** che racchiude la densità delle classi e ***$mids*** che rappresenta i punti centrali delle classi.

Di seguito verranno illustrati gli Istogrammi delle restanti classi:

Difficoltà di parcheggio
Frequenza assoluta delle classi
0
2
4
6
8
10
0
500
1000
1500
Classi

Traffico
Frequenza assoluta delle classi
0
2
4
6
8
10
12
0
500
1000
1500
2000
Classi

Scarsa illuminazione stradale
Frequenza assoluta delle classi
0
2
4
6
8
10
0
200
400
600
800
1000
1200
Classi

Cattive condizioni stradali
Frequenza assoluta delle classi
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
Classi

Inquinamento dell'aria
Frequenza assoluta delle classi
0
2
4
6
8
10
12
0
200
400
600
800
Classi

Rumore
Frequenza assoluta delle classi
0
2
4
6
8
10
12
0
100
200
300
400
500
Classi

1.4 Boxplot

Prendiamo in considerazione una variabile quantitativa X e le modalità che essa assume. Una volta ordinati i valori in **ordine crescente** possiamo definire il *primo quartile,* $\boldsymbol{Q_1}$, il valore per il quale il 25% dei dati sono alla sua sinistra e il restante 75% alla sua destra. Il *terzo quartile*, $\mathbf{Q_3}$, è il valore per il quale il 75% dei dati sono alla sua sinistra e il restante 25% alla sua destra. Il *secondo quartile* $\mathbf{Q_2}$ indica il valore per il quale 50% dei dati sono alla sua sinistra e il restante 50% è alla sua destra ed è detto ***mediana***. $\mathbf{Q_0}$ e $\mathbf{Q_4}$ forniscono il minimo e il massimo dei valori del campione. In R i quartili si calcolano tramite la funzione ***quantile()*** e la funzione ***summary()*** permette di determinare i valori precisi del minimo, del massimo, della media, della mediana, del primo e del terzo quantile.

```
> quantile(unlist(DatasetStrada[,2]))
     0%     25%     50%     75%    100%
  14.00   98.75  228.00  610.00 1589.00

summary(unlist(DatasetStrada[,2]))
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
14.00   98.75  228.00  426.73  610.00 1589.00
```

Il **boxplot**, è il disegno di una scatola i cui **estremi** sono Q_1 e Q_3, tagliata da una **linea orizzontale** in corrispondenza di Q_2, ossia della mediana. In basso e in alto sono presenti altre **due linee orizzontali**, dette i *baffi*. Il ***baffo inferiore*** corrisponde al **valore più piccolo tra le osservazioni** che risulta maggiore o uguale di $Q_1 - 1.5 \cdot (Q_3 - Q_1)$, mentre il ***baffo superiore*** corrisponde al valore **più grande** delle osservazioni che risulta minore o uguale a $Q_3+1.5\cdot(Q_3-Q_1)$. La **distanza** tra il primo e il terzo quartile è detta *intervallo interquartile* o ***scarto*** *interquartile*. I valori dei dati che sono al di sopra del baffo superiore o al di sotto del baffo inferiori costituiscono **un'anomalia nei dati**, e sono rappresentati da **punti**. Il boxplot viene utilizzato per illustrare alcune caratteristiche di una distribuzione di frequenza: la centralità, la forma, la dispersione e la presenza di eventuali valori anomali.

Nel boxplot di cui sopra che rappresenta la Difficoltà di parcheggio è presente appunto un'anomalia nei dati, rappresentata dalle famiglie della regione Lombardia; il risultato del summary ci informa invece che $Q_0 = 14$, $Q_1 = 98.75$, $Q_2 = 228.00$, $Q_3 = 610.00$, $Q_4 = 1589$.

Per generare i restanti boxplot è stato utilizzato il codice seguente, che ha permesso la visualizzazione in un unico grafo dei vari boxplot per ogni problema:

```
boxplot ( unlist(DatasetStrada[,2]), unlist(DatasetStrada[,3]),
          unlist(DatasetStrada[,4]), unlist(DatasetStrada[,5]),
          unlist(DatasetStrada[,6]), unlist(DatasetStrada[,7]),
          unlist(DatasetStrada[,8]), unlist(DatasetStrada[,9]),
          main =" Boxplot Problemi Stradali" , col = brewer.pal(8,"PRGn"), names=labelss)
```

Figura 14 - Creazione Boxplot

Nella pagina seguente è presente il grafo risultante.

Boxplot Problemi Stradali
0
500
1000
1500
2000
Difficoltà di parcheggio
Difficoltà di collegamento con mezzi pubblici
Traffico
Scarsa illuminazione stradale
Cattive condizioni stradali
Inquinamento dell'aria
Rumore
Rischio di criminalità

1.5 Scatterplot

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** 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**. Il risultato finale è una *nuvola di punti*. Il grafico che si ottiene mira ad evidenziare se le coppie di punti presentano **qualche forma di regolarità**, quindi ad evidenziare se esiste una relazione tra le variabili e di che tipo è tale relazione. Con il codice riportato di seguito, riferito al nostro dataset contenente le otto tipologie di problematiche stradali, ripartite nelle regioni italiane e provincie autonome; andiamo a realizzare uno scatterplot tramite la funzione ***pairs()*** che è in grado di visualizzare in un'unica finestra una pluralità di grafici per punti ottenuti mettendo in relazione tutte le coppie di variabili definite all'interno del nostro dataset.

```
pairs(DatasetStrada[,2:9], main = "Scatterplot")
```

Figura 15 - Creazione Scatterplot

2 Statistica Descrittiva Univariata

2.1 Funzione di Distribuzione Empirica Discreta

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 $z_1 < z_2 < ... < z_k$.

La **frequenza relativa cumulata** F_i rappresenta la proporzione dei dati del campione minori o uguali di $\mathbf{z_i}$. La funzione di distribuzione **empirica** F(x) è così definita:

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

Figura 16 - Funzione Distribuzione Empirica Discreta

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. 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**.

Come si nota, dalla F(x) si possono ricavare, in corrispondenza dei valori di x, le **rispettive probabilità**. Infatti, l'ampiezza del salto in *x* è pari alla probabilità P(X = *x*).

Di seguito è raffigurato il codice utilizzato per la creazione del grafo ed il calcolo della frequenza relativa cumulata:

```
> round (cumsum( table(DatasetStrada[,2]) /length(DatasetStrada[,2])) , 3 )
   14    39    54    67    81    87   134   145   149   162   209   247   403   548   562   601   613   708   812   955  1209
0.045 0.091 0.136 0.182 0.227 0.273 0.318 0.364 0.409 0.455 0.500 0.545 0.591 0.636 0.682 0.727 0.773 0.818 0.864 0.909 0.955
 1589
1.000
>
> plot(ecdf(DatasetStrada[,2]), main = "Funzione di distribuzione empirica discreta per difficoltà di parcheggio",
+      verticals="False", col="red")
```

Figura 17 - Codice Creazione FDED e Calcolo Frequenza Relativa Cumulata

2.2 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:

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

Figura 18 - Funzione Distribuzione Empirica Continua

La funzione di distribuzione empirica continua *coincide con il* ***segmento che passa per i punti*** (z_i,x_i) e (z_{i+1},x_{i+1}), ossia:

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

Vediamo come costruire il grafico della funzione di distribuzione empirica continua per il numero di famiglie, per le regioni, riferiti alla difficoltà di trovare parcheggio.

```
classi <- c(-100,14,100,400,700,1200,1590,2000)

FreqRelativeCumulate <- cumsum(table(cut(DatasetStrada[,2], breaks=classi,
                                         right = FALSE)))/length(DatasetStrada[,2])
```

Figura 19 - Creazioni Classi e calcolo FRC

Per prima cosa con il codice di cui sopra vengono scelte le classi, in questo caso [-200,0) [200,600) [600,1200) [1200,1600) [1600,2000) le classi [-200,0) e [1600,2000) sono fittizie, aggiunte per tracciare la linea y = 0 e y = 1. L frequenze Relative Cumulate calcolate sono le seguenti:

```
> FreqRelativeCumulate <- cumsum(table(cut(DatasetStrada[,2], breaks=classi,
+                                         right = FALSE)))/length(DatasetStrada[,2])
>
> FreqRelativeCumulate <- c(0, FreqRelativeCumulate)
> FreqRelativeCumulate
                    [-100,14)          [14,100)          [100,400)          [400,700)       [700,1.2e+03)
          0.0000000          0.0000000          0.2727273          0.5454545          0.7727273          0.9090909
[1.2e+03,1.59e+03)   [1.59e+03,2e+03)
          1.0000000          1.0000000
```

Nel codice di cui sopra, la funzione ***cut()*** con l'opzione ***right=FALSE*** consente di costruire classi con intervalli chiusi a sinistra invece che a destra, questa trasforma dati **numerici** in dati **qualitativi** mediante la loro *collocazione in classi*, sulla base di quanto specificato in **breaks**; di seguito vediamo come costruire il grafico in R delle frequenze relative cumulate:

```
plot (classi, FreqRelativeCumulate, type = "b", axes = FALSE, main = "Funzione di Distribuzione Empirica Continua", col = "red")
axis (1 , classi )
axis (2 , format (FreqRelativeCumulate , digits = 2) )
box ()
```

L'utilizzo della funzione di concatenazione ***c(0, FreqRelativeCumulate)*** permette di aggiungere uno zero all'inizio del vettore delle frequenze relative cumulative. La funzione ***plot()*** permette di rappresentare ordinatamente le coppie di punti contenuti nei vettori **classi** e **FreqRelativeCumulate**. Nella funzione ***plot()*** si è utilizzata l'opzione ***type = "b"*** che consente di congiungere i punti successivi del grafico mediante linee continue ai cui estremi sono presenti dei cerchietti, mentre l'opzione ***axes = FALSE*** consente di non tracciare gli assi. Infine mediante l'istruzione ***axis(1, classi)*** si disegna l'asse orizzontale in basso, mediante l'istruzione ***axis(2, format(FreqRelativeCumulate, digits = 2))*** si ottiene l'asse verticale di sinistra con una formattazione opportuna dei numeri.

2.3 Indici di Sintesi

Alcuni indici di sintesi sono *media, mediana, moda, varianza, deviazione standard e coefficiente di variazione*. La media, la mediana e la moda sono misure di **centralità**, mentre la varianza e la deviazione standard misurano la **dispersione dei dati**.

2.3.1 Media, mediana e moda

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

Il sistema R mette a disposizione le funzioni **mean()** e **median()** per calcolare rispettivamente la media e la mediana di un insieme di dati.

Per la difficoltà di parcheggio abbiamo che la media campionaria è:

```
> mean(DatasetStrada[,2])
[1] 426.7273
```

Ciò indica quante famiglie in media hanno difficoltà di parcheggiare per regione, si avvicina di molto al numero di famiglie che soffrono il disagio in Basilicata (454); di seguito sono elencate tutte le medie:

Difficoltà di parcheggio	426.72727
Difficoltà di collegamento con mezzi pubblici	385.09091
Traffico	463.00000
Scarsa illuminazione stradale	381.18182
Cattive condizioni stradali	638.36364
Inquinamento dell'aria	136.22727
Rumore	113.95455
Rischio di criminalità	76.22727

Figura 10 - Media Disagi per Regione

Si può notare come la media più alta sia presene in corrispondenza del problema delle “Cattive condizioni stradali”.

Per quanto riguarda la mediana invece:

```
> median(DatasetStrada[,2])
[1] 228
```

Data dalla somma dei valori di Sardegna (247) e Calabria (209); di seguito è rappresentata la mediana del resto dei disagi:

Difficoltà di parcheggio	228.0
Difficoltà di collegamento con mezzi pubblici	203.5
Traffico	234.5
Scarsa illuminazione stradale	254.0
Cattive condizioni stradali	419.0
Inquinamento dell'aria	41.0
Rumore	46.0
Rischio di criminalità	31.5

Figura 21 - Mediana Disagi per Regione

Infine la **moda** campionaria di un insieme di dati, se esiste, è la modalità a cui è associata la **frequenza** (assoluta o relativa) **più elevata**. Se esistono più modalità con frequenza massima, ciascuna di esse è detto valore moda.

Come abbiamo notato dai grafici precedenti, per tutte le regioni la moda, ovvero la modalità che si presenta con il valore maggiore, è la problematica rappresentata dalle “**Cattive condizioni stradali**”. Invece come regione più disagiata, abbiamo la “**Lombardia**”.

2.3.3 Percentili Decili e Quartili

Oltre alla mediana che divide a metà un insieme di dati ordinati, si possono definire altri **indici di posizione**, detti quantili, che suddividono l’insieme dei **dati ordinati** in un fissato numero di parti uguali; 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 delle probabilità e **j**=1, 2 … ,9 denota il tipo di algoritmo selezionato. Scegliendo **type=2** non si tiene conto dell’interpolazione dei dati. Se si omette **probs** vengono calcolati di default i quartili. Se si omette **type** per default viene implementato l’algoritmo di tipo 7.

```
quantile(DatasetStrada[,1], c(0,0.2,0.4,0.6,0.8,1), type=2)
0%  20%  40%  60%  80% 100%
14   81  149  548  708 1589
```

Figura 22 - Quantili per "Problemi di Parcheggio"

La funzione ***summary()*** restituisce oltre al minimo, massimo e ai tre quartili Q1, Q2 e Q3 anche la media campionaria.

```
> summary(DatasetStrada[,1])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  14.00   98.75  228.00  426.73  610.00 1589.00
```

2.3.4 Variazione, deviazione standard e coefficiente di variazione

Gli indici di posizione non tengono conto delle **variabilità** esistente tra i dati; infatti esistono distribuzioni di frequenza, che pur avendo la stessa media campionaria, sono molto diverse tra di loro. Indici significativi per **misurare la variabilità** di una distribuzione di frequenza sono la ***varianza campionaria*** e la ***deviazione standard***. Assegnato un insieme di dati numerici $x_1, x_2, \ldots, x_n$, 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 $\overline{x}$ denota la media campionaria dei dati. Inoltre, si definisce **deviazione standard** campionaria la ***radice quadrata*** *della varianza campionaria*, ossia:

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

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 media campionaria e la varianza campionaria sono i due indici di posizione e di dispersione dei dati maggiormente utilizzati. I valori della varianza campionaria e della deviazione standard campionaria **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.

In R la varianza campionaria di un vettore numerico si calcola utilizzando la **funzione *var()*** e la deviazione standard campionaria si calcola utilizzando la funzione ***sd().***

Per confrontare le **variazioni** esistenti **tra diversi campioni** di dati è utile introdurre ***coefficiente di variazione***, questo è il **rapporto** tra la **deviazione** standard campionaria e il **modulo** della **media** campionaria, ossia:

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

Il coefficiente di variazione è un numero **puro** ed è utilizzato per confrontare le variazioni di insieme di dati espressi in differenti unità di misura oppure insieme di dati aventi *differenti range di variazione*, cioè differenza tra il massimo ed il minimo dei dati. Nel nostro caso si può notare che vale il secondo caso in quanto **tra il minimo ed il massimo** c'è una grande differenza.

In R **non è definita** una funzione che calcola il coefficiente di variazione. Tale funzione può essere comunque, facilmente implementate in R nel seguente modo:

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

Calcoliamo la varianza, la deviazione standard e il coefficiente di variazione del numero di famiglie con difficoltà di parcheggio per ogni regione italiana:

```
> var(DatasetStrada[,1])
[1] 180191.2
>
> sd(DatasetStrada[,1])
[1] 424.4893
>
> cv(DatasetStrada[,1])
[1] 0.9947555
```

Possiamo notare che la varianza è molto alta in quanto i dati si **discostano molto dalla media** ponderata.

Nella tabella a lato sono presenti i valori di alcuni indici principali per le otto modalità di problemi presenti;

Come si può notare, le medie sono tutte diverse, anche se alcuni sono molto simili (Scarsa Illuminazione e Difficolta di collegamento); la modalità con i dati che **si discostano maggiormente dalla propria media** è "Cattive condizioni Stradali", perché ha varianza maggiore, così come la deviazione standard. Invece per il coefficiente di variazione non è così, la problematica con valore maggiore è "Inquinamento dell'aria".

	Difficoltà.di.parcheggio	Difficoltà.di.collegamento.con.mezzi.pubblici	Traffico	Scarsa.illuminazione.stradale	Cattive.condizioni.stradali	Inquinamento.dell.aria	Rumore	Rischio.di.criminalità
Media	426.727727	385.090091	463.000000	381.1818182	638.3636364	136.227273	113.954545	76.227273
Varianza	180191.1601732	137716.9437229	221241.809524	132600.4415584	366485.0995671	30174.660173	16076.426407	7535.422078
Deviazione Standard	424.4892934	371.1023359	470.363487	364.1434354	605.3801282	173.708550	126.792848	86.806809
Coefficiente di Variazione	0.9947555	0.9636746	1.015904	0.9553012	0.9483312	1.275138	1.112662	1.138789

2.3.5 Forma di una Distribuzione di Frequenza

Esistono degli indici statistici che permettono di misurare quando una *distribuzione di frequenza* presenta **simmetrie** o **asimmetrie** 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 x1, x2, . . . , xn, si definisce skewness campionaria il valore:

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

dove m3 denota il **momento centrato campionario di ordine 3**. In generale, il momento centrato campionario di **ordine j** è così definito:

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

Se la distribuzione di frequenza è **simmetrica** γ1 assume il **valore 0**, altrimenti **γ 1 > 0** per l'asimmetria **positiva** (ossia la distribuzione di frequenza ha la *coda di **destra** più allungata*) e **γ 1 < 0** per 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.

Un indice che invece permette di misurare la *densità dei dati intorno alla media* è la ***curtosi*** campionaria. Assegnato un insieme di dati numerici x1, x2, . . ., xn, si definisce curtosi campionaria il valore:

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

dove **m4** denota il momento centrato campionario di ordine 4.

Questo indice ci permette di *confrontare la distribuzione di frequenza dei dati* con una densità di probabilità normale che è caratterizzata da **curtosi uguale a 3**. Se la distribuzione di frequenza è più **piatta** di una normale(*platicurtica*) tale indice è **inferiore a 3**, mentre se è **più piccata** di una normale (*leptocurtica*) tale indice è **maggiore di 3**. Se invece tale indice è **uguale a 3** la distribuzione di frequenza si definisce ***normocurtica***, ossia piatta come una normale.

Di seguito verrà mostrato il codice per Il calcolo della curtosi campionaria e della skewness in R:

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

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

Di seguito verranno mostrati i risultati dell'esecuzione delle due funzioni per tutte le problematiche presenti:

Difficoltà di parcheggio	1.1983778
Difficoltà di collegamento con mezzi pubblici	0.9928185
Traffico	1.2551066
Scarsa illuminazione stradale	0.9195382
Cattive condizioni stradali	1.0472524
Inquinamento dell'aria	1.8651981
Rumore	1.4345609
Rischio di criminalità	1.3480913

Figura 23 - Skewness per problematiche

Difficoltà di parcheggio	0.76624119
Difficoltà di collegamento con mezzi pubblici	-0.02576547
Traffico	0.97303864
Scarsa illuminazione stradale	-0.31709500
Cattive condizioni stradali	0.25284351
Inquinamento dell'aria	3.65998954
Rumore	1.58076179
Rischio di criminalità	0.90911090

Figura 24 - Curtosi per problematiche

Per quanto riguarda la Skewness tutti i valori sono maggiori di 0, per cui è presente **asimmetria positiva,** la maggior asimmetria si verifica nell'Inquinamento dell'aria; Per quanto riguarda invece la curtosi possiamo notare tutte le problematiche siano caratterizzate da una composizione **platicurtica,** eccetta una, caratterizzata quindi da una forma **leptocurtica.**

3 Statistica descrittiva bivariata con R

Siano X e Y due variabili di tipo **quantitativo**. Considerato un campione (x1, y1), (x2, y2), . . . (xn, yn) 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 indipendente e la variabile dipendente, si disegnano dei punti in corrispondenza delle coppie (xi, yi). Il risultato finale è una nuvola di punti che può essere ottenuto 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 tende ad evidenziare se esiste una relazione tra le variabili e di quale tipo è tale relazione (lineare, quadratica, . . .).

Ad esempio mostriamo le due problematiche: “Traffico” e “Condizioni Stradali”:

	Cattive condizioni stradali	Traffico
Piemonte	1037	700
Valle d'Aosta	25	12
Liguria	411	313
Lombardia	2089	1767
Trentino Alto Adige	109	133
Prov.Aut. Bolzano	57	81
Prov. Aut. Trento	52	52
Veneto	832	805
Friuli-Venezia Giulia	195	186
Emilia-Romagna	982	686
Toscana	898	649
Umbria	282	118
Marche	334	188
Lazio	1987	1385
Abruzzo	290	163
Molise	67	31
Campania	1274	889
Puglia	923	642
Basilicata	135	57
Calabria	427	228
Sicilia	1199	860
Sardegna	439	241

Riportiamo in una tabella la mediana campionaria, la media campionaria e la deviazione standard dei due vettori:

	Cattive condizioni stradali	Traffico
Mediana	419.00	234.50
Media	638.36	463.00
Deviazione standard	605.38	470.36

Si nota come gli indici statistici pur essendo molto simili sono maggiori nel caso delle Cattive Condizioni Stradali;

Realizziamo lo scatterplot considerando **Traffico** come variabile **dipendente** e **Cattive condizioni stradali** come variabile **indipendente**; nello scatterplot sono visualizzate *20* coppie. Tracciamo nello scatterplot delle *linee orizzontali e verticali* in corrispondenza delle **mediane** campionarie e delle **medie** campionarie dei due vettori. Le seguenti linee di codice producono il diagramma di dispersione (scatterplot):

```
plot (  DatasetStrada$`Cattive condizioni stradali`,DatasetStrada$Traffico , main =" Traffico in funzione
di Cattive condizioni stradali" , xlab = "Cattive condizioni Stradali " , ylab ="Traffico" , col =" red ")
abline ( v= median(DatasetStrada$`Cattive condizioni stradali` ) , lty =1 , col =" magenta ")
abline ( v= mean(DatasetStrada$`Cattive condizioni stradali` ) , lty =2 , col =" blue ")
abline ( h= median(DatasetStrada$Traffico ) , lty =1 , col =" magenta ")
abline ( h= mean(DatasetStrada$Traffico ) , lty =2 , col =" blue ")
legend(0,1200,c("Mediana", "Media"),pch=0, col=c("magenta","blue"), cex=0.8)
```

3.1 Covarianza e correlazione campionaria

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

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

La covarianza campionaria può avere segno positivo, negativo o nullo. Quando **Cxy > 0** si dice che

le variabili sono *correlate* ***positivamente***, se **Cxy < 0** le variabili sono *correlate* ***negativamente*** e, infine, se **Cxy = 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*.

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

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

Si nota che il coefficiente di correlazione campionario **non fa distinzione** tra variabile dipendente e variabile indipendente, ossia $\mathbf{r_{xy} = r_{yx}}$, può essere **calcolato soltanto** se entrambe le variabili sono **quantitative**, *non* cambia al variare **dell'unità di misura** con cui sono espresse le variabili, è fortemente **influenzato** dalla presenza di eventuali valori **anomali**, così come accade per la media campionaria e la varianza campionaria.

Il coefficiente di correlazione campionario ha lo **stesso segno della covarianza**. Quando $r_{xy} > 0$ si dice che le variabili sono correlate positivamente, se $r_{xy} < 0$ le variabili sono correlate negativamente e, infine, se $r_{xy} = 0$ le variabili sono non correlate.

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 pertanto 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 relazioni:

- $r_{xy} = 1$ (*correlazione perfetta positiva*) tutti i punti sono allineati su una **retta ascendente**;
- r_{xy} compreso tra 0 e 1 estremi esclusi (*correlazione positiva*) i punti sono posizionati in una **nuvola** attorno ad una *linea retta interpolante ascendente*;
- $r_{xy} = 0$ (*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 sono posizionati in una

- nuvola attorno ad una *linea retta interpolante discendente*;
- r_{xy} = -1 (*correlazione perfetta negativa*) tutti i punti sono allineati su una linea **retta discendente.**

Nel linguaggio R le covarianze campionarie e le correlazioni campionarie fra una coppia X e Y di variabili numeriche possono essere immediatamente ottenute con le rispettive funzioni **cov(X,Y)** e **cor(X,Y)**.

```
> cov(DatasetStrada[,3], DatasetStrada[,5])
[1] 280599.4
> cor(DatasetStrada[,3], DatasetStrada[,5])
[1] 0.9854283
```

I dati dei due vettori Traffico e Cattive condizioni stradali, sono **positivamente correlati** essendo la covarianza campionaria uguale a 280599,4, ossia i valori assunti dal primo e dal secondo vettore tendono ad essere **grandi e piccoli insieme**. Inoltre, il coefficiente di correlazione è uguale a 0.9854283 ed è prossimo **all'unità**. Ciò indica, com'è evidenziato nel diagramma di dispersione (scatterplot), che esiste una **forte correlazione lineare tra i due vettori.**

```
plot(DatasetStrada$`Cattive condizioni stradali` , DatasetStrada$Traffico , main =" Retta di regressione " ,
      xlab = "Cattive condizioni Stradali" , ylab = "Traffico" , col =" red " )
abline ( lm(DatasetStrada$Traffico~ DatasetStrada$`Cattive condizioni stradali`) , col =" blue ")
```

Figura 25 - Codice creazione retta di regressione

3.2 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 le altre possibili rette. Consideriamo l'equazione della retta: *Y= α+ β X* dove α è l'intercetta e β è il coefficiente angolare.

Il coefficiente angolare β esprime **quantitativamente la pendenza** (*inclinazione*) della retta: un coefficiente angolare positivo (β > 0) indica una retta di **regressione crescente**, un coefficiente angolare negativo (β < 0) indica una retta **decrescente**; un coefficiente angolare nullo (β = 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

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

sia **minima**, dove n è il numero di osservazioni, (x1, x2, . . ., xn) sono i valori osservati della variabile X e (y1, y2, . . ., yn) sono i valori osservati della variabile Y. Il metodo dei minimi quadrati conduce a:

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

Le **medie** campionarie, le **deviazioni** standard campionarie e il **coefficiente di correlazione** permettono di stimare i parametri *α* e *β* della retta di regressione. Ricordiamo che in R la funzione $\mathbf{r_{xy}}$ viene calcolata tramite la funzione ***cor(x, y)*** e la funzione s^2x tramite la funzione ***var(x).***

Nel nostro esempio, riferendoci a *"Traffico"* e *"Cattive condizioni stradali"* le seguenti linee di codice calcolano il coefficiente angolare *β* e l'intercetta *α.*

```
CoefficienteAngolare <- ( sd ( DatasetStrada$Traffico ) / sd ( DatasetStrada$`Cattive condizioni stradali` ) ) *
  cor( DatasetStrada$Traffico , DatasetStrada$`Cattive condizioni stradali` )

Intercetta <- mean( DatasetStrada$Traffico ) - CoefficienteAngolare *mean(DatasetStrada$`Cattive condizioni stradali`)
```

La funzione ***lm(y~x)*** è utilizzata per eseguire le analisi di regressione lineari. Il nome lm della funzione rappresenta l'acronimo di linear model. L'argomento *y~x* passato alla funzione *lm()* indica che y dipende da x, ossia che x è la variabile indipendente e y la variabile dipendente. La funzione

abline(lm(y~x)) permette di aggiungere la retta di regressione al grafico dello scatterplot, con y che dipende da x.

Ad esempio, per i nostri due vettori possiamo vedere che

```
> lm(DatasetStrada$Traffico~ DatasetStrada$`Cattive condizioni stradali`)

Call:
lm(formula = DatasetStrada$Traffico ~ DatasetStrada$`Cattive condizioni stradali`)

Coefficients:
                                (Intercept)  DatasetStrada$`Cattive condizioni stradali`
                                  -25.7633                                        0.7657
```

3.2.1 Residui

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**

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

sono ottenuti mediante la retta 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. La media campionaria dei valori stimati ($\hat{y}1$, $\hat{y}2$, . . ., $\hat{y}n$) è uguale alla media campionaria delle osservazioni (y_1, y_2, . . ., y_n). I residui

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

mostrano di quanto si discostano i **valori osservati dai valori stimati** con la retta di regressione. La **media** campionaria dei **residui** è **nulla**, ossia in media gli scostamenti positivi e negativi si compensano. La varianza campionaria dei residui è

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

essendo la media dei residui = 0. Vediamo come si calcolano in R il vettore dei valori stimati e quello dei residui.

```
> stime <- array(fitted( lm(DatasetStrada$Traffico~ DatasetStrada$`Cattive condizioni stradali`)))
> `row.names<-`(stime,regioni)
            Piemonte         Valle d'Aosta              Liguria            Lombardia   Trentino Alto Adige     Prov.Aut. Bolzano
           768.216054            -6.622056           288.918962          1573.680175            57.692570            17.878754
    Prov. Aut. Trento               Veneto Friuli-Venezia Giulia        Emilia-Romagna              Toscana               Umbria
            14.050502           611.257741           123.538496           726.105287           661.790661           190.150073
               Marche              Lazio             Abruzzo              Molise             Campania               Puglia
           229.963889         1495.583844           196.275275            25.535257           949.675176           680.931919
           Basilicata            Calabria             Sicilia             Sardegna
            77.599478           301.169367           892.251403           310.357171
```

La funzione ***fitted(lm(y ~ x))*** calcola il vettore dei 22 valori stimati. Invece la funzione ***resid(lm(y ~ x))*** determina i 22 residui:

```
> residui <- array(resid( lm(DatasetStrada$Traffico~ DatasetStrada$`Cattive condizioni stradali`)))
> `row.names<-`(residui,regioni)
            Piemonte        Valle d'Aosta              Liguria             Lombardia   Trentino Alto Adige     Prov.Aut. Bolzano
           -68.216054            18.622056            24.081038            193.319825             75.307430            63.121246
    Prov. Aut. Trento               Veneto Friuli-Venezia Giulia        Emilia-Romagna               Toscana               Umbria
            37.949498           193.742259            62.461504            -40.105287            -12.790661           -72.150073
               Marche                Lazio              Abruzzo               Molise              Campania               Puglia
           -41.963889          -110.583844           -33.275275              5.464743            -60.675176           -38.931919
           Basilicata             Calabria              Sicilia             Sardegna
           -20.599478           -73.169367           -32.251403            -69.357171
```

Invece, la mediana, la varianza campionaria e la deviazione standard campionaria dei residui sono:

```
> linearModel <-  lm(DatasetStrada$Traffico~ DatasetStrada$`Cattive condizioni stradali`)
> median(linearModel$residuals)
[1] -26.42544
> var(linearModel$residuals)
[1] 6400.771
> sd(linearModel$residuals)
[1] 80.00482
```

Non si può invece calcolare il **coefficiente di variazione**, essendo la media campionaria dei residui nulla; di seguito vedremo come rappresentare graficamente i residui.

Segmenti che congiungono i valori stimati e i valori osservati

Realizziamo il grafico dei residui aggiungendo, al grafico contenente lo scatterplot e la **retta di regressione**, dei segmenti verticali che visualizzano i residui.

```
plot(DatasetStrada$`Cattive condizioni stradali` , DatasetStrada$Traffico , main =" Retta di regressione con residui " ,
     xlab = "Cattive condizioni Stradali" , ylab = "Traffico" , col =" red " )
abline ( lm(DatasetStrada$Traffico~ DatasetStrada$`Cattive condizioni stradali`) , col =" blue ")
segments(DatasetStrada$`Cattive condizioni stradali`, stime,DatasetStrada$`Cattive condizioni stradali`,
         DatasetStrada$Traffico, col="magenta" )
```

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. Facciamo ciò attraverso le seguenti linee di codice:

```
plot(DatasetStrada$Traffico, residui, main = "Diagramma dei residui", xlab="Cattive condizioni stradali",
     ylab="Traffico", pch=9, col="red")
abline(h=0, col = "blue", lty=2)
```

I punti indicano la posizione dove si collocano i **residui** rispetto al numero di famiglie che vivono in zone con strade in cattive condizioni nelle 22 regioni. La retta orizzontale è posizionata nello **zero** e corrisponde alla **media** campionaria dei **residui**, che è nulla. Si nota che i punti sono disposti quasi **casualmente** attorno alla linea orizzontale e non si evidenzia nessun comportamento nella distribuzione dei punti; l'unica caratteristica che si evince è che 3 regioni si distaccano di molto dalla media.

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*. La posizione della retta di regressione è fortemente influenzata dalla presenza di eventuali valori **anomali**. 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.

Valori dei residui standardizzati rispetto ai valori stimati

Risulta di un certo interesse il calcolo dei **residui standardizzati** definiti come:

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

Caratterizzati da *media* campionaria **nulla** e *varianza* **unitaria**.

```
> `row.names<-`(residuiStandard,regioni)
            Piemonte        Valle d'Aosta              Liguria             Lombardia   Trentino Alto Adige
         -0.85264930            0.23276167           0.30099484            2.41635222            0.94128616
    Prov.Aut. Bolzano     Prov. Aut. Trento              Veneto Friuli-Venezia Giulia        Emilia-Romagna
          0.78896804            0.47434014           2.42163233            0.78072176           -0.50128588
             Toscana               Umbria               Marche                 Lazio               Abruzzo
         -0.15987363           -0.90182158          -0.52451701           -1.38221478           -0.41591588
              Molise              Campania              Puglia             Basilicata              Calabria
          0.06830517           -0.75839401          -0.48661967           -0.25747796           -0.91456199
             Sicilia              Sardegna
         -0.40311825           -0.86691240
```

Realizzeremo con il seguente codice, un grafico in cui i residui **standardizzati** (ordinate) vengono disegnati in funzione dei valori **stimati** (ascisse) mediante la retta di regressione

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

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

3.3 Regressione lineare multipla

Il modello di regressione lineare multipla viene utilizzato per spiegare la **relazione** tra una variabile **quantitativa Y**, detta variabile dipendente, e le **variabili quantitative indipendenti** X1, X2, . . ., Xp. Definendo un data frame con le **p+1 variabili**, ovvero **Y, X1, X2, … , Xp** avremo che le funzioni di R ***cov(dfm)*** e ***cor(dfm)*** restituiranno **matrici** di dimensioni (p+1)*(p+1) i cui elementi sono le covarianze e gli indici di correlazioni tra le variabili. In particolare, queste **matrici sono simmetriche**; la matrice delle **covarianze** contiene sulla **diagonale** principale la **varianza** delle singole **colonne** del data frame, mentre la matrice degli **indici di correlazione** contiene il numero **1 sulla diagonale** principale; quest'ultima indica le **correlazioni** che ci sono **tra le variabili**. Nel nostro caso cosideriamo il dataframe ***DatasetStrada;*** mostrato nell'introduzione creiamo le matrici della covarianza e del coefficiente di correlazione con il seguente codice:

```
View(round(cov(DatasetStrada), digits = 1 ))
View(round(cor(DatasetStrada), digits = 3 ))
```

	Difficoltà di parcheggio	Difficoltà di collegamento con mezzi pubblici	Traffico	Scarsa illuminazione stradale	Cattive condizioni stradali	Inquinamento dell'aria	Rumore	Rischio di criminalità
Difficoltà di parcheggio	180191.2	153005.6	197529.3	150161.1	252947.2	71434.6	53406.3	35579.7
Difficoltà di collegamento con mezzi pubblici	153005.6	137716.9	167982.0	131182.1	216026.6	60875.3	45326.1	30173.6
Traffico	197529.3	167982.0	221241.8	166229.2	280599.4	79545.3	59316.4	39640.4
Scarsa illuminazione stradale	150161.1	131182.1	166229.2	132600.4	217213.9	57588.1	44273.4	29936.1
Cattive condizioni stradali	252947.2	216026.6	280599.4	217213.9	366485.1	98976.3	75050.4	51080.0
Inquinamento dell'aria	71434.6	60875.3	79545.3	57588.1	98976.3	30174.7	21682.9	14233.1
Rumore	53406.3	45326.1	59316.4	44273.4	75050.4	21682.9	16076.4	10743.2
Rischio di criminalità	35579.7	30173.6	39640.4	29936.1	51080.0	14233.1	10743.2	7535.4

Figura 26 - Matrice di Covarianza

	Difficoltà di parcheggio	Difficoltà di collegamento con mezzi pubblici	Traffico	Scarsa illuminazione stradale	Cattive condizioni stradali	Inquinamento dell'aria	Rumore	Rischio di criminalità
Difficoltà di parcheggio	1.000	0.971	0.989	0.971	0.984	0.969	0.992	0.966
Difficoltà di collegamento con mezzi pubblici	0.971	1.000	0.962	0.971	0.962	0.944	0.963	0.937
Traffico	0.989	0.962	1.000	0.971	0.985	0.974	0.995	0.971
Scarsa illuminazione stradale	0.971	0.971	0.971	1.000	0.985	0.910	0.959	0.947
Cattive condizioni stradali	0.984	0.962	0.985	0.985	1.000	0.941	0.978	0.972
Inquinamento dell'aria	0.969	0.944	0.974	0.910	0.941	1.000	0.984	0.944
Rumore	0.992	0.963	0.995	0.959	0.978	0.984	1.000	0.976
Rischio di criminalità	0.966	0.937	0.971	0.947	0.972	0.944	0.976	1.000

Figura 27 - Matrice di Correlazione Campionaria

In generale si nota che **tutte le modalità** presentano **un'alta correlazione.**

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

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

In R la funzione **lm(y~x_1 + x_2 + . . . x_p)** è utilizzata per eseguire le **analisi di regressione lineari multivariate.**

Riferendoci al nostro data-frame, supponiamo la variabile **dipendente** sia **Traffico** e che le **altre 7** siano **indipendenti**.

```
> lm(DatasetStrada[,3]~DatasetStrada[,1]+DatasetStrada[,2]+DatasetStrada[,4]+DatasetStrada[,5]+DatasetStrada[,6]
+    +DatasetStrada[,7]+DatasetStrada[,8])

Call:
lm(formula = DatasetStrada[, 3] ~ DatasetStrada[, 1] + DatasetStrada[,
    2] + DatasetStrada[, 4] + DatasetStrada[, 5] + DatasetStrada[,
    6] + DatasetStrada[, 7] + DatasetStrada[, 8])

Coefficients:
       (Intercept)  DatasetStrada[, 1]  DatasetStrada[, 2]  DatasetStrada[, 4]  DatasetStrada[, 5]  DatasetStrada[, 6]
           27.7439             -0.2931             -0.1609              0.3302              0.2147              0.3343
DatasetStrada[, 7]  DatasetStrada[, 8]
            3.2750             -0.7794
```

Ciò mostra il seguente modello di regressione multipla:

$y = 27.7439 - 0.2931\, x_1 - 0.1609\, x_2 + 0.3302\, x_3 + 0.2147\, x_4 + 0.3343\, x_5 + 3.2750\, x_6 - 0.7794 x_7$

Vediamo che i regressori **β_3, β_4, β_5, β_6,** sono **positivi** e questo indica che questi disagi hanno un **effetto positivo** sul traffico (all'aumentare del numero di famiglie che soffrono di qualcuno degli altri 7 disagi aumenteranno anche quelle del traffico); invece i restanti **3** hanno un **effetto negativo**; ovvero all'aumentare di questi 3, il valore di "Traffico" **diminuisce**.

3.3.1 Residui

Una volta calcolati i valori dei coefficienti α e β1, β2, . . . , βp, 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 mostrano di quanto si discostano i valori osservati dai valori stimati con la retta di regressione. Utilizziamo, per calcolare i valori stimati e i residui, le funzioni ***fitted*** e ***resid***, di seguito mostrate per il DatasetStrada:

```
> stimemulti <- array(fitted(multilm))
> `row.names<-`(stimemulti,regioni)
            Piemonte        Valle d'Aosta              Liguria            Lombardia   Trentino Alto Adige
           658.27288             42.89515            285.81734           1780.84595             97.69552
   Prov.Aut. Bolzano     Prov. Aut. Trento               Veneto Friuli-Venezia Giulia        Emilia-Romagna
            78.76645             46.21060            713.22201            184.17999            650.06772
             Toscana               Umbria              Marche                Lazio              Abruzzo
           647.84044            157.47346            179.02291           1392.19426            170.25749
              Molise             Campania              Puglia            Basilicata             Calabria
            62.00113            939.96328            664.35832             89.22541            215.83321
             Sicilia             Sardegna
           883.77235            246.08414
> residiumulti <- array(resid(multilm))
> `row.names<-`(residiumulti,regioni)
            Piemonte        Valle d'Aosta              Liguria            Lombardia   Trentino Alto Adige
           41.727115            -30.895154            27.182655           -13.845951            35.304479
   Prov.Aut. Bolzano     Prov. Aut. Trento               Veneto Friuli-Venezia Giulia        Emilia-Romagna
            2.233550             5.789403            91.777994             1.820012            35.932283
             Toscana               Umbria              Marche                Lazio              Abruzzo
            1.159564            -39.473459             8.977093            -7.194263            -7.257488
              Molise             Campania              Puglia            Basilicata             Calabria
          -31.001134            -50.963275           -22.358318           -32.225408            12.166792
             Sicilia             Sardegna
          -23.772355             -5.084137
```

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

```
> median(multilm$residuals)
[1] -1.962286
> var(multilm$residuals)
[1] 1057.573
> sd(multilm$residuals)
[1] 32.52035
```

Calcolando i residui standardizzati otteniamo i seguenti risultati:

```
> residuimultistd <- array(residuimulti/sd(residuimulti))
> `row.names<-`(residuimultistd,regioni)
            Piemonte        Valle d'Aosta              Liguria            Lombardia   Trentino Alto Adige
          1.28310771           -0.95002518           0.83586595           -0.42576263            1.08561182
   Prov.Aut. Bolzano     Prov. Aut. Trento               Veneto Friuli-Venezia Giulia        Emilia-Romagna
          0.06868161            0.17802400           2.82217093            0.05596532            1.10491677
             Toscana               Umbria              Marche                Lazio              Abruzzo
          0.03565655           -1.21380785           0.27604538           -0.22122339           -0.22316756
              Molise             Campania              Puglia            Basilicata             Calabria
         -0.95328407           -1.56711939          -0.68751771           -0.99093046            0.37412854
             Sicilia             Sardegna
         -0.73099929           -0.15633707
```

È possibile realizzare un grafico in cui i residui standardizzati (ordinate) vengono disegnati in funzione dei valori stimati (ascisse) attraverso il seguente codice:

```
plot(stimemulti, residuimultistd, main="Residui standard rispetto ai valori stimati",
  xlab="valori stimati", ylab="Residui standard", pch=5, col ="red")
abline(h=0, col="blue", lty=2)
```

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

3.4 Regressione non lineare

Oltre all'analisi della regressione lineare, è stata effettuata anche un'analisi di regressione non lineare. In riferimento al nostro Dataset sono stati considerati sempre "Traffico" come variabile dipendente e "Cattive condizioni stradali" come variabile indipendente.

3.4.1 Regressione Polinomiale

Vediamo se l'approssimazione polinomiale è adatta per approssimare i nostri dati. Per stimare α, β e γ si può ricorrere alla regressione multipla

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

In cui i regressori sono $X_1 = X$ e $X_2 = X^2$.

Si usa la funzione **lm(y~x + I(x ∧ 2))** in R per stimare i parametri **α, β** e **γ**. La funzione ***I()*** è un ***identificatore di variabile*** e viene inserito quando si devono effettuare operazioni matematiche come divisione, elevamento a potenza, nelle **variabili della regressione**.

```
> pol <- lm(DatasetStrada[,3]~ DatasetStrada[,5]+ I((DatasetStrada[,5])^2))
> pol

Call:
lm(formula = DatasetStrada[, 3] ~ DatasetStrada[, 5] + I((DatasetStrada[,
    5])^2))

Coefficients:
              (Intercept)         DatasetStrada[, 5]  I((DatasetStrada[, 5])^2)
                4.538e+00                  6.388e-01                  6.688e-05
```

Si può ricavare il modello di regressione:

$$Y = 4.5 + 6.39\,X + 6.69\,X^2$$

Le successive linee di codice:

```
> alpha <- pol$coefficients[[1]]
> beta <- pol$coefficients[[2]]
> gamma <- pol$coefficients[[3]]
>
> c(alpha,beta,gamma)
[1] 4.538203e+00 6.388414e-01 6.687749e-05
```

Permettono di ricavare i nostri parametri.

Disegniamo ora la **curva** stimata sullo scatterplot, abbiamo in questo modo:

Come si può evincere, la curva ottenuta non è adatta ad approssimare i nostri dati.

4 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. Quest'analisi ha lo scopo di distribuire le osservazioni in gruppi in modo tale che il **grado** di naturale **associazione** sia **alto** tra i membri dello **stesso gruppo** e **basso** tra i membri di **gruppi diversi**.

Sia $I = \{I_1, I_2, \ldots, I_n\}$ un insieme di n individui (entità o unità) appartenenti ad una popolazione ideale. Assumiamo che esista un insieme di **caratteristiche** $C = \{C_1, C_2, \ldots, C_p\}$ che sono osservabili e sono possedute da ogni individuo in I. Il termine osservabile denota caratteristiche che danno origine a dati sia di tipo qualitativo che di tipo quantitativo (detti anche misure). Denotiamo con il simbolo $\mathbf{x_{ij}}$ il valore della misura della caratteristica j–esima relativa all'individuo I_i e con $\mathbf{X_i} = (x_{i1}, x_{i2}, \ldots, x_{ip})$ (i = 1, 2, . . . , n) il **vettore** di cardinalità 1×p **di tali misure**.

Il problema dell'analisi dei cluster consiste nel determinare **m sottoinsiemi**, detti cluster, di individui in I, con **m < 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**.

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. In molti metodi di clustering si raccomanda la standardizzazione di ogni variabile (caratteristica) usando la **media** campionaria e la **deviazione** standard campionaria entrambe derivate dall'insieme completo di individui della popolazione.

Per risolvere il problema di clustering è chiaramente desiderabile **definire** i termini **somiglianza** o **differenza** in modo **quantitativo**, cioè capire cosa si intende per somiglianza tra due individui appartenenti allo stesso cluster, e differenza tra due che appartengono a cluster differenti. La somiglianza tra due individui I_i e I_j ($i \neq j$) la si può vedere mediante un **coefficiente di similarità $s_{ij} = s(X_i, X_j)$** oppure mediante una misura di **distanza $d_{ij} = d(X_i, Xj)$.** Mentre i coefficienti di **similarità** assumono valori **compresi tra 0 e 1**, le misure di **distanza** possono assumere **qualsiasi valore reale maggiore o uguale a zero**. Quindi con questi due parametri possiamo capire come inserire due

individui nello stesso cluster, cioè se il coefficiente di similarità è **prossimo a 1 allora sono molto simili**, oppure se la **distanza** tra i due individui è **abbastanza piccola** allora possono essere **messi nello stesso cluster**. Altrimenti se il coefficiente di similarità è prossimo a 0, oppure la distanza è molto grande, allora i due individui sono dissimili e quindi non possono essere messi nello stesso cluster.

4.1 Distanza e similarità

Le misure metriche di somiglianza sono soprattutto basate sulle funzioni distanza tra i vettori delle caratteristiche. Le **distanze** tra tutte le possibili coppie di unità sono inserite in una matrice **simmetrica** D di cardinalità n × n dove dij = d(Xi,Xj) (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)

ritorna 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:

- metrica **euclidea** (euclidean);
- metrica del valore assoluto o metrica di Manhattan (manhattan);
- metrica del massimo o metrica di Chebycev (maximum);
- metrica di Minkowski (minkowski);
- distanza di Canberra (canberra);
- distanza di Jaccard (binary);

4.1.1 Metrica Euclidea

La più familiare misura di distanza è la metrica Euclidea:

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

dove $\mathbf{x_{ik}}$ è il valore della **k–esima caratteristica** dell'individuo $\mathbf{I_i}$.

La distanza Euclidea usata su tutti i dati è fortemente **influenzata dall'unità di misura** in base alla quale è valutata ciascuna delle **p** caratteristiche. Un metodo per ovviare a questo problema consiste nello **scalare** e standardizzare inizialmente le misure in maniera tale da realizzare la possibilità di un confronto tra le misure, cioè considerare delle nuove variabili così calcolate:

$$x_{ij}^* = \frac{x_{ij} - \overline{x}_j}{s_j} \qquad (i = 1, 2, \ldots, n; j = 1, 2, \ldots, p)$$

dove $\overline{x}_j$ e **sj** sono rispettivamente la **media** campionaria e la **deviazione** standard campionaria della j–esima caratteristica.

Mediante lo scalamento e la standardizzazione si ottengono dei **nuovi dati** le cui **medie** campionarie sono **nulle** e le **varianze** campionarie **unitarie**.

In R per **scalare e standardizzare** le variabili si utilizza la funzione:

scale(X, center = TRUE, scale = TRUE)

Dove:

- *X* rappresenta una matrice numerica o un data frame;
- ***center*** è posto 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).

Per il nostro dataset in R si è usato il comando:

datasetScalato<-scale(dati)

per scalare i dati e la funzione

dist(datasetScalato, method = "euclidean", diag = FALSE, upper = FALSE)

per calcolare la distanza.

4.1.2 Altre metriche

Oltre alla metrica euclidea esistono altre metriche utilizzate nelle tecniche di clustering, una di queste è la metrica del **valore assoluto** (Manhattan), detta metrica L_1 così definita:

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

Esiste un'altra metrica, detta del **massimo** (metrica di Chebycev), così definita:

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

I comandi R per le due metriche sono:

dist(datasetScalato, method = "manhattan", diag = TRUE, upper = TRUE)

dist(datasetScalato, method = "maximum", diag = TRUE, upper = TRUE)

Entrambe le metriche del valore assoluto e del massimo sono computazionalmente **semplici da calcolare** con l'unica differenza è che la metrica di **Chebycev** coinvolge anche una procedura di **ordinamento**.

Un'altra possibile metrica è la metrica di Canberra:

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

La metrica di **Canberra** è definita **per variabili non negative** ed ha la caratteristica di essere **sensibile** alle **differenze relative** piuttosto che a quelle assolute.

dist(datasetScalato, method = "canberra", diag = TRUE, upper = TRUE)

4.1.3 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:

- $s(X_i,X_i) = 1$;

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

La quantità s_{ij} è chiamata coefficiente di similarità ed è l'elemento nella riga i-esima e colonna j-esima della matrice di similarità S, che presenta la **diagonale** con tutti elementi pari ad **1**.

4.2 Misura di non omogeneità totale

Alla matrice X delle misure si può associare una matrice WX di cardinalità p × p, detta *matrice delle varianze e covarianze* così definita:

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

Dove l'elemento $\mathbf{w_{rl}}$ è: $w_{r\ell} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ir} - \overline{x}_r)(x_{i\ell} - \overline{x}_\ell) \qquad (r, \ell = 1, 2, \dots, p)$

Si nota che se **r=l** l'elemento $\mathbf{w_{rl}}$ è la **varianza campionaria relativa alla caratteristica r-esima** effettuata su tutti gli n individui, mentre **se r ≠ l** l'elemento $\mathbf{w_{rl}}$ **è la covarianza** campionaria effettuata su tutti gli n individui. In R utilizzando la funzione **apply(X,2,mean), apply(X, 2, var) e apply(X, 2, sd)** è possibile calcolare la media campionaria, la varianza campionaria e la deviazione standard campionaria delle colonne di una matrice X. Inoltre applicando la funzione **cov(X)** è possibile ottenere la **matrice $\mathbf{W_X}$** delle **covarianze** campionarie tra le *varie caratteristiche*.

La **matrice statistica di non omogeneità** per l'insieme **I** di individui, di cardinalità p × p, è così definita:

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

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

Quando r = ℓ, si nota che h_{rl} corrisponde a n−1 volte la **varianza** campionaria della caratteristica r-esima effettuata su tutti gli n individui.

La **traccia** della matrice H_I, denotata con $_{tr}H_I$, detta misura di non omogeneità statistica dell'insieme I di individui è $\mathrm{tr}H_I = \sum_{k=1}^{p} h_{kk} = (n-1) \sum_{k=1}^{p} s_k^2 = \sum_{i=1}^{n} d_2^2(X_i, \overline{X})$

$\overline{X}$

Dove d2 indica la distanza euclidea e dove è un vettore di cardinalità p il cui generico elemento xj rappresenta la **media** campionaria relativa alla **caratteristica** j-esima effettuata sugli n individui.

4.3 Metodi di ottimizzazione

Una volta scelta la misura di distanza (o di similarità) si pone il problema di procedere alla scelta di un idoneo algoritmo di raggruppamento delle unità osservate. I metodi di raggruppamento si distinguono in tre tipi:

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

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.**

I metodi gerarchici di clustering eseguono una sequenza ordinata di operazioni della stessa natura. Tali metodi sono distinti in ***agglomerativi*** che procedono per aggregazioni successive delle unità **partendo da n gruppi** formati da un **singolo individuo** e ***divisivi*** che partono da un **solo gruppo** formato da **tutte le unità** e procedono a **divisioni successive** fino a giungere a **gruppi** formati da una **sola unità**.

I metodi non gerarchici di clustering consentono, a differenza delle tecniche di tipo gerarchico, di **riallocare gli individui** già classificati ad un livello precedente dell'analisi. A differenza dei metodi gerarchici, **l'obiettivo** finale dei metodi **non gerarchici** è quello di ottenere **un'unica partizione degli n individui** di partenza in cluster.

4.4 Metodi gerarchici

I metodi di clustering gerarchico possono essere di due tipi: agglomerativi e divisivi. I metodi gerarchici di tipo **agglomerativo** procedono con una sequenza di **successive unioni** degli n individui iniziali in gruppi mentre i metodi gerarchici di tipo **divisivo** procedono con una sequenza di

successive divisioni dell'insieme degli n individui in partizioni sempre più fini. **L'obiettivo** finale dei metodi gerarchici non è quello di ottenere una singola partizione degli n individui di partenza, ma di **ottenere una sequenza di partizioni** che possono essere **rappresentate graficamente** mediante una struttura ad albero, detta **Dendrogramma**, nella quale, sull'insieme delle **ordinate,** sono riportati i **livelli di distanza,** mentre sull'asse delle **ascisse** sono riportati i **singoli individui**. Molti metodi di analisi gerarchica sono caratterizzati da una struttura comune che si riflette in un algoritmo generale che può essere esplicitato come segue:

Algoritmo

• **Step 1:** A partire dalla matrice X (o quella scalata) **calcolare** la matrice delle distanze;

• **Step 2:** individuare la coppia di cluster **meno distanti** e raggruppare in un **unico cluster**;

• **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.

L'analisi gerarchica di tipo agglomerativo viene effettuata in R attraverso la funzione:

> hclust(d,method = "complete")

Dove

- *d* rappresenta un oggetto (che individua **una struttura di dissimilarità**) creato tramite la funzione dist();
- *method* seleziona il **metodo gerarchico** agglomerativo (di default è **complete**).

La funzione hclust() produce come **output** una **lista**, i cui elementi sono:

- la **sequenza del processo** di agglomerazione ($merge)
- un **vettore**, la cui lunghezza corrisponde al numero di iterazioni, che indica **il livello di distanza** alla quale è avvenuta **l'unione tra due cluster** ($height)
- un'opportuna **permutazione delle unità** (individui) finalizzata alla **costruzione** del Dendrogramma ($order)
- un vettore delle **etichette** che contrassegnano le varie unità ($labels)

Per ottenere il Dendrogramma si impiega la funzione:

> plot(z, labels = NULL, hang = 0.1,main = "Dendrogramma", sub = NULL, xlab = NULL)

Dove

- *z* è l'oggetto creato (**output**) dalla funzione hclust();
- *labels* è un vettore di **etichette** per i rami del dendrogramma (di default impiega i nomi delle righe del data frame);
- *hang* determina **l'altezza** alla quale le **etichette** vengono **visualizzate** al di sotto del Dendrogramma (un valore negativo pone le etichette in corrispondenza dell'ordinata nulla);
- *main, sub, xlab* sono comandi per la finestra grafica.

Vediamo ora i diversi metodi agglomerativi

4.4.1 Metodo del legame singolo

In questo metodo la **distanza tra i gruppi G1**, contenente n1 individui, e **G2**, contenente n2 individui, è definita come la **minima** tra tutte le ***n1n2*** distanze che si possono calcolare tra ogni individuo di G1 e ogni individuo di G2. Vediamo ciò nella seguente figura:

Nel **primo passo** si vedono gli individui **più vicini** attraverso la *matrice delle distanze*, e si raggruppano nello **stesso** cluster, e si va avanti per tutti gli individui. 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 invece esso può dare origine alla formazione di una **catena** tra gli **individui**. Utilizziamo questo metodo con i nostri dati e costruiamo il Dendrogramma relativo. Innanzitutto calcoliamo la matrice delle **distanze**, sulla matrice **dei dati scalata**, utilizzando la metrica **euclidea** e applichiamo il metodo gerarchico del legame singolo.

```
> matriceDistanza <- dist(DatasetScaled, method = "euclidean", diag=TRUE, upper = TRUE)
> hls <- hclust(matriceDistanza, method = "single")
> str(hls)
List of 7
 $ merge      : int [1:21, 1:2] -7 -6 -2 -19 -13 -12 -5 -9 7 -20 ...
 $ height     : num [1:21] 0.0908 0.1472 0.1543 0.2142 0.2218 ...
 $ order      : int [1:22] 4 14 3 5 19 2 6 7 16 9 ...
 $ labels     : chr [1:22] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "single"
 $ call       : language hclust(d = matriceDistanza, method = "single")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

Analizziamo l'output del metodo del legame singolo visualizzando **$merge:**

```
> hls$merge
      [,1] [,2]
 [1,]   -7  -16
 [2,]   -6    1
 [3,]   -2    2
 [4,]  -19    3
 [5,]  -13  -15
 [6,]  -12    5
 [7,]   -5    4
 [8,]   -9    6
 [9,]    7    8
[10,]  -20  -22
[11,]    9   10
[12,]  -10  -11
[13,]  -18   12
[14,]   -3   11
[15,]   -8   13
[16,]   -1   15
[17,]  -21   16
[18,]  -17   17
[19,]   14   18
[20,]  -14   19
[21,]   -4   20
```

I numeri con il segno negativo indicano i **singoli individui**, mentre i numeri **positivi** indicano i **cluster** che si formano. Inoltre, **$height** indica la distanza in cui è avvenuta l'agglomerazione tra i cluster.

```
> hls$height
 [1] 0.09083747 0.14720857 0.15428784 0.21423430 0.22183188 0.22435638 0.27539426 0.29915295
0.30433431 0.35008978 0.37865899
[12] 0.38086618 0.39190928 0.55165773 0.55750814 0.56508164 1.17116169 1.67801202 2.06566441
2.34245621 2.75160653
```

Quindi al livello 1 si uniscono gli individui I_7 e I_{16}, la distanza è 0. 09083747; al livello 2 si uniscono il primo cluster e I_6 a distanza 0.14720857 e così via finché all'ultimo livello non si uniscono il ventesimo cluster e l'individuo 4 a 2.75160653.

L'istruzione *axis(side = 4, at = round(c(0, hls$height), 2))* permette di costruire l'asse delle altezze alla destra del grafico arrotondando i numeri alla seconda cifra decimale.

4.4.2 Metodo del legame completo

In questo metodo la **distanza tra i gruppi G1 e G2** è definita come la **massima** tra tutte le ***n1n2*** distanze che si possono calcolare tra ogni individuo di G1 e ogni individuo di G2.

Nella procedura si considera inizialmente, cioè al **livello 0**, un insieme di **n cluster** $\{I_1\}$, $\{I_2\}$, . . ., $\{I_n\}$. Al passo successivo si cerca nella **matrice D** delle distanze il **coefficiente di distanza minima** e si raggruppano nello **stesso cluster** G_{ij} i due individui $\mathbf{I_i}$ **e** $\mathbf{I_j}$ associati secondo tale coefficiente.

Al livello 1 quindi si **modifica la matrice** delle distanze valutando le distanze di G_{ij} da ogni altro individuo I_k non appartenente a G_{ij} mediante la seguente relazione $d(_{ij}),_k = \max(d_{ik}, d_{jk})$. Ad ogni passo successivo, dopo che i cluster Gu e Gv sono stati uniti scegliendo dalla precedente matrice delle distanze i **due cluster più vicini**, la distanza tra il nuovo cluster, denotato con G_{uv}, e un altro cluster Gz è definito come $d(_{uv}),_z = \max(d_{uz}, d_{vz})$. La quantità $\mathbf{d(_{uv}),_z}$ rappresenta la misura di distanza tra gli **elementi più lontani** dei cluster G_{uv} e Gz. Viene ripetuta la procedura finché non si ottiene un **unico cluster** formato da tutti gli individui.

```
> hls$merge
      [,1] [,2]
 [1,]   -7  -16
 [2,]   -2    1
 [3,]  -13  -15
 [4,]   -6  -19
 [5,]   -5    4
 [6,]   -9  -12
 [7,]  -20  -22
 [8,]  -10  -11
 [9,]    3    6
[10,]  -18    8
[11,]    2    5
[12,]   -1   -8
[13,]   10   12
[14,]   -3    7
[15,]    9   11
[16,]  -21   13
[17,]   14   15
[18,]  -14  -17
[19,]   -4   18
[20,]   16   17
[21,]   19   20
```

```
> hls <- hclust(matriceDistanza, method = "complete")
> str(hls)
List of 7
 $ merge      : int [1:21, 1:2] -7 -2 -13 -6 -5 -9 -20 -10 3 -18 ...
 $ height     : num [1:21] 0.0908 0.1917 0.2218 0.2518 0.2848 ...
 $ order      : int [1:22] 4 14 17 21 18 10 11 1 8 3 ...
 $ labels     : chr [1:22] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "complete"
 $ call       : language hclust(d = matriceDistanza, method = "complete")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

```
> hls$height
 [1]  0.09083747  0.19166805  0.22183188  0.25182988  0.28484855  0.34612664  0.35008978  0.38086618  0.39040372
[10]  0.41929513  0.53801911  0.57018339  0.70621096  0.76887709  1.04330673  1.67044609  1.74526739  2.34245621
[19]  4.27223702  5.02874194 10.17466917
```

Al livello 1 si uniscono l'individuo 7 e il 16 e la loro distanza è 0.09083747; al passo due vediamo che si uniscono il cluster 1 e l'individuo 2 a distanza 0.19166805 e così via finché non vediamo all'ultimo livello, il 21, che si uniscono il cluster 19 ed il cluster 20 a distanza 10.17466917. Il Dendrogramma è rappresentato a pagina seguente:

Il Dendrogramma costruito con questo metodo ha i **rami molto più lunghi** rispetto al Dendrogramma ottenuto con il metodo del legame singolo poiché i **gruppi** si formano **ad altezze maggiori**.

4.4.3 Metodo del legame medio

In questo metodo la **distanza tra i gruppi** G1 e G2 è definita come la **media aritmetica** delle distanze tra tutte le coppie di unità che compongono i due gruppi.

Nella procedura al livello zero si effettuano le stesse operazioni dei metodi precedenti andando a creare il cluster G_{ij} con due individui molto vicini. Quando si arriva al livello 1 si modifica la matrice delle distanze D valutando le distanze degli individui del cluster G_{ij} con gli individui non appartenenti al cluster mediante la relazione:

$$d_{(i,j),k} = \frac{1}{2}\left(d_{i,k} + d_{j,k}\right) \qquad (k = 1, 2, \ldots, n; k \neq i, j)$$

Cioè si **fa la media** delle **distanze**. Si va avanti fino ad arrivare ad avere un **unico cluster con n individui**.

```
> hlsa$merge
      [,1] [,2]
 [1,]   -7  -16
 [2,]   -6    1
 [3,]   -2    2
 [4,]  -13  -15
 [5,]  -19    3
 [6,]   -5   -9
 [7,]  -12    4
 [8,]  -20  -22
 [9,]  -10  -11
[10,]  -18    9
[11,]    6    7
[12,]   -1   -8
[13,]   10   12
[14,]    5   11
[15,]   -3    8
[16,]   14   15
[17,]  -21   13
[18,]  -17   17
[19,]   -4  -14
[20,]   16   18
[21,]   19   20
> hlsa$height
 [1] 0.09083747 0.17253749 0.20920603 0.22183188 0.26419280 0.30433431 0.30485231 0.35008978 0.38086618 0.40560221
[11] 0.41508567 0.57018339 0.62352171 0.64772583 0.66026741 1.10156781 1.43317775 2.26210124 2.75160653 3.71460282
[21] 7.20340269
```

```
> hlsa <- hclust(matriceDistanza, method = "average")
> str(hlsa)
List of 7
 $ merge      : int [1:21, 1:2] -7 -6 -2 -13 -19 -5 -12 -20 -10 -18 ...
 $ height     : num [1:21] 0.0908 0.1725 0.2092 0.2218 0.2642 ...
 $ order      : int [1:22] 4 14 19 2 6 7 16 5 9 12 ...
 $ labels     : chr [1:22] "Piemonte" "Valle d'Aosta" "Liguria" "Lombardia" ...
 $ method     : chr "average"
 $ call       : language hclust(d = matriceDistanza, method = "average")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

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 $\mathbf{D}^{(2)}$ che contiene i quadrati delle singole distanze euclidee.

4.4.4 Metodo del centroide

In questo metodo la distanza tra i gruppi 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.

Al livello 0 si considera la matrice $D^{(2)}$ che contiene il **quadrato delle distanze euclidee**, e si scelgono i due individui con la distanza minore che verranno messi nel cluster Gij. Al livello 1 quindi si modifica la matrice dei quadrati delle distanze valutando i quadrati delle distanze di Gij da ogni altro individuo non appartenente a Gij mediante la relazione:

$$d^2_{(ij),k} = \sum_{r=1}^{p} (\overline{x}_{(i,j),r} - \overline{x}_{k,r})^2 = \frac{1}{2}(d^2_{ik} + d^2_{jk}) - \frac{1}{4} d^2_{ij}, \qquad (k \neq i, j)$$

Dove:

$$\overline{x}_{(i,j),r} = \frac{1}{2}(x_{i,r} + x_{j,r}) \qquad \overline{x}_{k,r} = x_{k,r} \qquad (r = 1, 2, \ldots, p)$$

Successivamente, dopo aver creato la nuova matrice si effettueranno le misure delle distanze tra i vari cluster, fino ad arrivare ad avere un unico cluster con gli n individui.

```
> matriceDistanza2 <- matriceDistanza^2
> hc <- hclust(matriceDistanza2, method = "centroid")
> hc$merge
      [,1] [,2]
 [1,]   -7  -16
 [2,]   -2    1
 [3,]   -6    2
 [4,]  -13  -15
 [5,]  -19    3
 [6,]  -12    4
 [7,]   -5   -9
 [8,]  -20  -22
 [9,]    6    7
[10,]  -10  -11
[11,]  -18   10
[12,]   -1   11
[13,]   -8   12
[14,]    5    9
[15,]   -3    8
[16,]   14   15
[17,]  -21   13
[18,]  -17   17
[19,]   -4  -14
[20,]   16   18
[21,]   19   20
> hc$height
 [1]  0.008251446  0.028207827  0.039074711  0.049209382  0.060698628  0.087112179  0.092619370  0.122562854
 [9]  0.129887382  0.145059047  0.128435884  0.324004840  0.285426924  0.380147481  0.417108400  1.085687731
[17]  1.959521540  4.839053062  7.571338483 13.510093998 50.683822503
```

La suddivisione in cluster è uguale alla precedente. Le distanze invece sono cambiate.

4.4.5 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**. Infatti, il metodo della mediana, così come il metodo del legame singolo, può dare origine alla formazione di una **catena**

```
> hcm <- hclust(matriceDistanza2, method = "median")
> hcm$merge
      [,1] [,2]
 [1,]   -7  -16
 [2,]   -2    1
 [3,]   -6    2
 [4,]  -13  -15
 [5,]  -19    3
 [6,]  -12    4
 [7,]   -5    5
 [8,]   -9    6
 [9,]  -20  -22
[10,]  -10  -11
[11,]  -18   10
[12,]    7    8
[13,]   -1   -8
[14,]   11   13
[15,]   -3    9
[16,]   12   15
[17,]  -21   14
[18,]  -17   17
[19,]   -4  -14
[20,]   18   19
[21,]   16   20
> hcm$height
 [1]  0.008251446  0.028207827  0.046788779  0.049209382  0.062066959  0.087112179  0.090965867  0.092449497
 [9]  0.122562854  0.145059047  0.128435884  0.199574658  0.325109096  0.274175836  0.417108400  0.954921752
[17]  1.900369738  3.651080089  7.571338483 15.374374499 37.600680567
```

tra gli individui. Applichiamo il metodo della mediana al nostro caso di analisi, otteniamo questo raggruppamento:

4.5 Analisi del Dendrogramma

Analizziamo il Dendrogramma e calcoliamo, **fissato** il numero di cluster, le **misure di non omogeneità** della partizione individuata.

4.5.1 Disegnare rettangoli che evidenziano i cluster

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.

Vediamo cosa accade con il dendrogramma del metodo del legame singolo:

```
plot(hls, hang =-1, xlab="Metodo gerarchico agglomerativo", sub="del legame singolo")
rect.hclust(hls, k=2, border ="red")
```

Possiamo mostrare una diversa partizione per confrontare, usando sulla stessa finestra il comando:

```
rect.hclust(hls, k=3, border ="green")
```

4.5.2 Inserire gli individui nei cluster

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:

cutree(tree, k = NULL, h = 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**.

L'output della funzione ***cutree()*** è un vettore contenente numeri interi **positivi** associati ai cluster in cui sono stati inseriti **i vari individui.**

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

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 di tale funzione **cutree()** è una **matrice** in cui la **colonna k-esima** contiene numeri interi **positivi** associati ai **cluster in cui sono stati inseriti** i vari individui. Quest'ultima permette di *classificare gli individui all'aumentare del numero di cluster*, ad esempio, la partizione in **4 cluster** è **G1** = {Piemonte, Veneto, Emilia Romagna, Toscana, Puglia, Sicilia}, **G2** = {Valle d'Aosta, Liguria, Trentino, Prov.Aut. Bolzano, Prov.Aut. Trento, Friuli-Venezia Giulia, Umbria, Marche, Abruzzo, Molise, Basilicata, Calabria, Sardegna}, **G3** = {Lombardia } e **G4** = {Campania, Lazio}. Marche, Abruzzo, Molise, Basilicata, Calabria, Sardegna}, **G3** = {Lombardia } e **G4** = {Campania, Lazio}.

```
> cutree(hls, k=1:20)
                    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Piemonte            1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
Valle d'Aosta       1 1 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2  2  2  2
Liguria             1 1 2 2 2 3 3 3 3  3  3  3  3  3  3  3  3  3  3  3
Lombardia           1 2 3 3 3 4 4 4 4  4  4  4  4  4  4  4  4  4  4  4
Trentino Alto Adige 1 1 2 2 2 2 2 2 2  2  2  5  5  5  5  5  5  5  5  5
Prov.Aut. Bolzano   1 1 2 2 2 2 2 2 2  2  2  5  5  5  5  5  5  6  6  6
Prov. Aut. Trento   1 1 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2  2  2  2
Veneto              1 1 1 1 1 1 1 1 1  1  5  6  6  6  6  6  6  7  7  7
Friuli-Venezia Giulia 1 1 2 2 2 2 2 5 5  5  6  7  7  7  7  7  7  8  8  8
Emilia-Romagna      1 1 1 1 1 1 1 1 1  6  7  8  8  8  8  8  8  9  9  9
Toscana             1 1 1 1 1 1 1 1 1  6  7  8  8  8  9  9  9 10 10 10
Umbria              1 1 2 2 2 2 2 5 5  5  6  7  7  7  7  7 10 11 11 11
Marche              1 1 2 2 2 2 2 5 5  5  6  7  7  9 10 10 11 12 12 12
Lazio               1 2 3 4 4 5 5 6 6  7  8  9  9 10 11 11 12 13 13 13
Abruzzo             1 1 2 2 2 2 2 5 5  5  6  7  7  9 10 10 11 12 12 14
Molise              1 1 2 2 2 2 2 2 2  2  2  2  2  2  2  2  2  2  2  2
Campania            1 2 3 4 5 6 6 7 7  8  9 10 10 11 12 12 13 14 14 15
Puglia              1 1 1 1 1 1 1 1 1  6  7  8 11 12 13 13 14 15 15 16
Basilicata          1 1 2 2 2 2 2 2 2  2  2  5  5  5  5  5  5  6 16 17
Calabria            1 1 2 2 2 3 3 3 8  9 10 11 12 13 14 14 15 16 17 18
Sicilia             1 1 1 1 1 1 7 8 9 10 11 12 13 14 15 15 16 17 18 19
Sardegna            1 1 2 2 2 3 3 3 8  9 10 11 12 13 14 16 17 18 19 20
```

Usando **table()** viene mostrato che nel **primo** cluster ci sono 6 individui, nel **secondo** 13, nel **terzo** 1 e nel **quarto** 2 che è proprio la partizione vista in precedenza.

```
> table(cutree(hls, k=4, h=NULL))

 1  2  3  4
 6 13  1  2
```

4.5.3 Misure di sintesi associate ai cluster

Possiamo ricavare misure di sintesi sui singoli cluster ottenuti tagliando il dendrogramma tramite la funzione ***cutree(),*** utilizzando la funzione ***aggregate():***

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, come la media campionaria, ecc., ad ognuna delle caratteristiche associate ai diversi cluster che sono stati aggregati.

```
> taglio <- list(cutree(hls, k=4))
> aggregate(DatasetStrada,taglio, mean)
  Group.1 Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici  Traffico Scarsa illuminazione stradale
1       1                640.6667                                       592.3333  723.6667                     594.0000
2       2                137.7692                                       124.8462  138.6923                     125.3846
3       3               1589.0000                                      1301.0000 1767.0000                    1137.0000
4       4               1082.0000                                       997.0000 1137.0000                    1027.5000
  Cattive condizioni stradali Inquinamento dell'aria    Rumore Rischio di criminalità
1                    978.5000              211.33333 173.66667              112.16667
2                    217.1538               24.23077  28.53846               18.15385
3                   2089.0000              718.00000 486.00000              292.00000
4                   1630.5000              348.00000 304.00000              238.00000
> aggregate(DatasetStrada,taglio, var)
  Group.1 Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici   Traffico Scarsa illuminazione stradale
1       1                 10200.67                                      11713.467   7885.067                    21549.200
2       2                 10919.36                                       8076.974   8309.397                     8688.923
3       3                       NA                                             NA         NA                           NA
4       4                 32258.00                                      10658.000 123008.000                    43512.500
  Cattive condizioni stradali Inquinamento dell'aria    Rumore Rischio di criminalità
1                   16615.50               691.8667  278.2667               204.1667
2                   23776.97               307.3590  408.6026               192.8077
3                         NA                     NA        NA                     NA
4                  254184.50              1458.0000 4418.0000              4418.0000
>
> aggregate(DatasetStrada,taglio, sd)
  Group.1 Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici Traffico Scarsa illuminazione stradale
1       1                 100.9983                                      108.22877  88.7979                    146.79646
2       2                 104.4957                                       89.87199  91.1559                     93.21439
3       3                       NA                                             NA       NA                           NA
4       4                 179.6051                                      103.23759 350.7250                    208.59650
  Cattive condizioni stradali Inquinamento dell'aria   Rumore Rischio di criminalità
1                   128.9011               26.30336 16.68133               14.28869
2                   154.1978               17.53166 20.21392               13.88552
3                         NA                     NA       NA                     NA
4                   504.1671               38.18377 66.46804               66.46804
```

Abbiamo calcolato la mediana, varianza e deviazione standard. Il **cluster** G1 ha coordinate mostrate nella **prima riga** della prima funzione calcolata e cosi gli altri.

Ricordiamo che per calcolare **var** e **sd** c'è bisogno di avere almeno 2 individui, quindi risultano essere **NA** dove vi è un **solo elemento nel cluster**.

4.5.4 Misure di non omogeneità statistiche

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 **non omogeneità tra i cluster** (tr B): ***tr T = tr S + tr B.***

Il nostro obiettivo è quello di individuare i cluster in modo tale da **minimizzare** la misura di non omogeneità statistica **all'interno dei cluster** (within) e **massimizzare** la misura di non omogeneità statistica **tra i gruppi** (between).

Se 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**.

Omogeneità statistica completa:

```
> n <- nrow (DatasetScaled )
>  trHI <- (n-1)*sum( apply (DatasetScaled, 2 , var ))
> trHI
[1] 168
```

Calcoliamo ora le misure di non omogeneità statistica relative ai gruppi:

```
tree <- hclust(matriceDistanza2, method = "single")
taglio <- cutree(tree, k=4, h=null)
num <- table(taglio)
tagliolist <- list(taglio)
agvar <- aggregate(DatasetScaled, tagliolist, var)[,-1]
trh1 <- (num[[1]] -1) * sum(agvar[1,])
trh2 <- (num[[2]] -1) * sum(agvar[2,])
trh3 <- (num[[3]] -1) * sum(agvar[3,])
trh4 <- (num[[4]] -1) * sum(agvar[4,])
```

matriceDistanza2 è la matrice contenente i quadrati delle distanze euclidee e **num** è un vettore contenente il numero di individui nei 4 cluster. La misura di non omogeneità del **primo gruppo** è calcolata moltiplicando **n1−1** per la somma degli elementi della **prima riga** della **matrice ottenuta** con ***aggregate()*** ed è uguale a 6.41. Invece, la misura di non omogeneità del **secondo** gruppo è calcolata moltiplicando **n2 − 1** per la somma degli elementi della seconda riga della matrice ottenuta con ***aggregate()*** il risultato è 4.18. Dato che nel **terzo e quarto** gruppo abbiamo solo un individuo, il valore della misura di non omogeneità è **nullo**.

```
> trh1
[1] 6.410193
> trh2
[1] 4.180806
> trh3
[1] NA
> trh4
[1] NA
```

Riassumendo la misura di non omogeneità statistica **totale** è **168**, la misura di non omogeneità statistica all'interno dei 4 gruppi (within) è trHG1 + trHG2 + trHG3 + trHG4= 6.41 + 4.18 + 0 + 0 = **10.59** e la misura di non omogeneità tra i cluster (between) è trH(G1 ∩ G2 ∩ G3 ∩ G4) = trHI −trHG1 −trHG2 = 168- 10.59 = **157.41 .** In conclusione, la misura di non omogeneità **all'interno dei gruppi è quindi piccola** rispetto la misura di non omogeneità tra i cluster. Il nostro obiettivo di minimizzare within e massimizzare between è **soddisfatto**.

Abbiamo fatto la stessa cosa con gli altri metodi, ovvero quello completo, medio, centroide e mediana e i risultati sono quasi sempre gli stessi:

```
> tree <- hclust(matriceDistanza2, method = "median")
> taglio <- cutree(tree, k=4, h=null)
> num <- table(taglio)
> tagliolist <- list(taglio)
> agvar <- aggregate(DatasetScaled, tagliolist, var)[,-1]
> trh1 <- (num[[1]] -1) * sum(agvar[1,])
> trh2 <- (num[[2]] -1) * sum(agvar[2,])
> trh3 <- (num[[3]] -1) * sum(agvar[3,])
> trh4 <- (num[[4]] -1) * sum(agvar[4,])
>
> trh1
[1] 6.410193
> trh2
[1] 4.180806
> trh3
[1] NA
> trh4
[1] NA
```

```
> tree <- hclust(matriceDistanza2, method = "average")
> taglio <- cutree(tree, k=4, h=null)
> num <- table(taglio)
> tagliolist <- list(taglio)
> agvar <- aggregate(DatasetScaled, tagliolist, var)[,-1]
> trh1 <- (num[[1]] -1) * sum(agvar[1,])
> trh2 <- (num[[2]] -1) * sum(agvar[2,])
> trh3 <- (num[[3]] -1) * sum(agvar[3,])
> trh4 <- (num[[4]] -1) * sum(agvar[4,])
>
> trh1
[1] 6.410193
> trh2
[1] 4.180806
> trh3
[1] NA
> trh4
[1] NA
```

```
> tree <- hclust(matriceDistanza2, method = "centroid")
> taglio <- cutree(tree, k=4, h=null)
> num <- table(taglio)
> tagliolist <- list(taglio)
> agvar <- aggregate(DatasetScaled, tagliolist, var)[,-1]
> trh1 <- (num[[1]] -1) * sum(agvar[1,])
> trh2 <- (num[[2]] -1) * sum(agvar[2,])
> trh3 <- (num[[3]] -1) * sum(agvar[3,])
> trh4 <- (num[[4]] -1) * sum(agvar[4,])
>
> trh1
[1] 6.410193
> trh2
[1] 4.180806
> trh3
[1] NA
> trh4
[1] NA
```

```
> tree <- hclust(matriceDistanza2, method = "complete")
> taglio <- cutree(tree, k=4, h=null)
> num <- table(taglio)
> tagliolist <- list(taglio)
> agvar <- aggregate(DatasetScaled, tagliolist, var)[,-1]
> trh1 <- (num[[1]] -1) * sum(agvar[1,])
> trh2 <- (num[[2]] -1) * sum(agvar[2,])
> trh3 <- (num[[3]] -1) * sum(agvar[3,])
> trh4 <- (num[[4]] -1) * sum(agvar[4,])
>
> trh1
[1] 2.262433
> trh2
[1] 4.180806
> trh3
[1] NA
> trh4
[1] 2.743551
```

I risultati per gli altri metodi, sono uguali a quelli del legame singolo, tranne nel caso del legame completo, dove il totale non cambia ma **trHG1** = 2.26 e **trHG2** = 2.74; la misura di non omogeneità statistica all'interno dei 4 gruppi (within) è trHG1 + trHG2 + trHG3 + trHG4= 2.26 + 4.18 + 0 + 2.74 = **10.59** e la misura di non omogeneità tra i cluster (between) è trH(G1 ∩ G2 ∩ G3 ∩ G4) = trHI − trHG1 −trHG2 = 168- 10.59 = **157.41 .**

4.6 Metodi non gerarchici

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:

1. Fissare a priori il numero k di cluster, specificando **k punti di riferimento** iniziali che portano a una prima partizione provvisoria;
2. Considerare tutti gli **individui** e attribuire ciascuno di essi al **cluster** individuato dal punto di riferimento da cui ha **distanza minore**;
3. Calcolare il baricentro (il **centroide**) di ognuno dei **k gruppi** così ottenuti. Tali centroidi costituiscono i **punti di riferimento** per i nuovi cluster;
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;
5. **Ricalcolare** i centroidi dei k gruppi così ottenuti;
6. Ripetere il procedimento a partire dal punto (4) fino a che i *centroidi non subiscono ulteriori modifiche* rispetto all'iterazione precedente. Si procede così iterativamente a spostamenti successivi fino a raggiungere una configurazione stabile.

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.

L'analisi con il metodo k-means 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; *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.

L'output di k-means è formato da un **vettore** che indica in *quale cluster è **collocato** ogni individuo*, una **matrice** che contiene i **centroidi** dei cluster, un **vettore** contenente le **misure di non omogeneità** statistica calcolate **all'interno** di ognuno dei **cluster**; tali valori dipendono

dall'omogeneità interna e dalla numerosità del gruppo ($withins), e infine la *dimensione dei gruppi.*

L'applicazione dell'algoritmo è stata effettuata considerando un numero massimo di 10 iterazioni e fissando il numero di cluster a 3, facendo una scelta casuale dei punti di riferimento:

```
> kmeans(DatasetStrada, center = 3, iter.max = 10, nstart=1)
K-means clustering with 3 clusters of sizes 7, 13, 2

Cluster means:
  Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici  Traffico Scarsa illuminazione stradale
1                 685.5714                                       660.5714  747.2857                      634.8571
2                 137.7692                                       124.8462  138.6923                      125.3846
3                1399.0000                                      1112.5000 1576.0000                     1156.0000
  Cattive condizioni stradali Inquinamento dell'aria    Rumore Rischio di criminalità
1                   1020.7143              227.00000 185.57143               123.42857
2                    217.1538               24.23077  28.53846                18.15385
3                   2038.0000              546.50000 418.50000               288.50000

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

Within cluster sum of squares by cluster:
[1] 815926.6 728164.8 290112.0
 (between_SS / total_SS =  92.0 %)
```

Il metodo individua la seguente partizione in 3 cluster: **G1**={Campania, Emilia, Piemonte, Puglia, Sicilia, Toscana, Veneto}, **G2**={Abruzzo, Basilicata, Calabria, Friuli, Liguria, Marche, Molise, Sardegna, Trentino, Bolzano, Trento, Umbria, Valle D'Aosta} e **G3**={Lombardia, Lazio}. I centroidi hanno sette coordinate. I valori **815926.6, 728164.8** e **290112** appena ottenuti sono le **misure di non omogeneità statistica** associate ai **3 cluster**. Il valore di omogeneità è **92.0%**, il che è ottimo. Il valore è più alto all'interno di un cluster perché la regione con i valori più alti e discostanti dalle altre, ovvero le regioni **Lombardia e Lazio**, sono messe da sole **in un cluster**.

Poiché la **partizione** iniziale è scelta **casualmente**, non è detto che la procedura del k-means conduca sempre allo **stesso risultato**, soprattutto nel caso in cui il numero **n** di **individui** è **piccolo**. Per questa ragione è consigliabile scegliere casualmente diversi insiemi di punti di riferimento, ponendo nstart > 1.

Vediamo cosa cambia nel metodo k-means se scegliamo i **centroidi** come **punti di riferimento**. Abbiamo già calcolato la matrice delle distanze euclidee e la stessa, ma con le distanze al quadrato; abbiamo anche già calcolato il metodo del centroide con:

```
> tree <- hclust(matriceDistanza2, method = "centroid")
> taglio <- list(cutree(tree, k=4, h=null))
> centroidiIni <- aggregate(DatasetStrada,taglio,mean) [,-1]
> kmeans(DatasetStrada, center = centroidiIni, iter.max = 10, nstart=1)
K-means clustering with 4 clusters of sizes 7, 13, 1, 1

Cluster means:
  Difficoltà di parcheggio Difficoltà di collegamento con mezzi pubblici  Traffico Scarsa illuminazione stradale
1                 685.5714                                       660.5714  747.2857                     634.8571
2                 137.7692                                       124.8462  138.6923                     125.3846
3                1589.0000                                      1301.0000 1767.0000                    1137.0000
4                1209.0000                                       924.0000 1385.0000                    1175.0000
  Cattive condizioni stradali Inquinamento dell'aria    Rumore Rischio di criminalità
1                   1020.7143              227.00000 185.57143              123.42857
2                    217.1538               24.23077  28.53846               18.15385
3                   2089.0000              718.00000 486.00000              292.00000
4                   1987.0000              375.00000 351.00000              285.00000

Clustering vector:
             Piemonte         Valle d'Aosta               Liguria             Lombardia   Trentino Alto Adige     Prov.Aut. Bolzano
                    1                     2                     2                     3                     2                     2
    Prov. Aut. Trento               Veneto Friuli-Venezia Giulia        Emilia-Romagna               Toscana                Umbria
                    2                     1                     2                     1                     1                     2
               Marche                 Lazio               Abruzzo                Molise              Campania                Puglia
                    2                     4                     2                     2                     1                     1
           Basilicata              Calabria               Sicilia              Sardegna
                    2                     2                     1                     2

Within cluster sum of squares by cluster:
[1] 815926.6 728164.8      0.0      0.0
 (between_SS / total_SS =  93.3 %)
```

Ricordiamo che per motivi computazionali bisogna **eliminare** la **prima colonna** della matrice dei centroidi ottenuta con la funzione aggregate() che si riferisce alle **etichette** dei cluster.

Vediamo come fornendo i centroidi, kmeans trovi 4 cluster, a differenza di prima nel **terzo** cluster è presente solo la Lombardia, poiché il Lazio fa parte del **quarto**; gli altri cluster e le relative misure di non omogeneità statistica sono le stesse, meno il terzo in cui adesso è 0, ciò porta il valore di omogeneità ad aumentare al 93.3%.

Proviamo ad applicare il metodo **k-means** alla matrice **scalata** Z, considerando una suddivisione in **2 cluster**.

```
> kmeans(DatasetScaled, center=2, iter.max = 10, nstart=1)
K-means clustering with 2 clusters of sizes 13, 9

Cluster means:
  Difficoltà.di.parcheggio Difficoltà.di.collegamento.con.mezzi.pubblici   Traffico Scarsa.illuminazione.stradale
1               -0.6807193                                   -0.7012749 -0.6894831                    -0.7024628
2                0.9832612                                    1.0129526  0.9959201                     1.0146684
  Cattive.condizioni.stradali Inquinamento.dell.aria     Rumore Rischio.di.criminalità
1                  -0.6957774             -0.6447380 -0.6736664              -0.6689962
2                   1.0050117              0.9312882  0.9730737               0.9663279

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

Within cluster sum of squares by cluster:
[1]  4.180806 45.441197
 (between_SS / total_SS =  70.5 %)
```

Vediamo che la **frequenza ottenuta è bassa**. Applichiamo di nuovo il metodo con una suddivisione **in 5 cluster:**

```
> kmeans(DatasetScaled, center=5, iter.max = 10, nstart=1)
K-means clustering with 5 clusters of sizes 2, 6, 3, 4, 7

Cluster means:
  Difficoltà.di.parcheggio Difficoltà.di.collegamento.con.mezzi.pubblici   Traffico Scarsa.illuminazione.stradale
1                2.2904529                                     1.9601307  2.3662551                     2.1277829
2               -0.8525396                                    -0.8953260 -0.8546582                    -0.9232126
3               -0.3307361                                    -0.3703136 -0.4301638                    -0.3227167
4               -0.6854761                                    -0.6584192 -0.6362101                    -0.6561475
5                0.6097778                                     0.7423303  0.6043958                     0.6966357
  Cattive.condizioni.stradali Inquinamento.dell.aria     Rumore Rischio.di.criminalità
1                   2.3119959              2.3618453  2.4019135              2.4453465
2                  -0.9319714             -0.7257018 -0.8027888             -0.8070865
3                  -0.3513445             -0.5520393 -0.4675964             -0.4979710
4                  -0.5998110             -0.5928164 -0.6345354             -0.5901297
5                   0.6315877              0.5225576  0.5648338              0.5437511

Clustering vector:
             Piemonte         Valle d'Aosta               Liguria             Lombardia   Trentino Alto Adige     Prov.Aut. Bolzano
                    5                     2                     3                     1                     2                     2
    Prov. Aut. Trento               Veneto Friuli-Venezia Giulia        Emilia-Romagna               Toscana                 Umbria
                    2                     5                     4                     5                     5                     4
               Marche                 Lazio               Abruzzo                Molise              Campania                Puglia
                    4                     1                     4                     2                     5                     5
           Basilicata              Calabria               Sicilia              Sardegna
                    2                     3                     5                     3

Within cluster sum of squares by cluster:
[1] 3.7856692 0.2081085 0.3393537 0.1524374 6.4101930
 (between_SS / total_SS =  93.5 %)
```

Vediamo che con una suddivisione in **5** cluster e la **matrice scalata**, il **rapporto è molto alto**; questo indica che il **metodo** ha fornito una suddivisione **ottima**.

In conclusione, i metodi **gerarchici** presentano un evidente **vantaggio** dal punto di vista **computazionale** rispetto ai metodi di ottimizzazione; tuttavia risultano **sensibili** a vettori delle **caratteristiche anomali** e **non** consentono di **modificare** la configurazione raggiunta, ossia una volta che un individuo è stato attribuito ad un cluster permane al suo interno per sempre. I metodi **non gerarchici** non presentano questo problema poiché consentono di **riallocare** gli individui precedentemente classificati, ma richiedono una **scelta opportuna della configurazione iniziale**. È buona norma applicare una pluralità di metodi per verificare la stabilità dei gruppi. Si applica prima **un'analisi gerarchica** per identificare il **numero di gruppi** e gli eventuali vettori delle caratteristiche anomali; si applica successivamente una **tecnica non gerarchica** per consentire di **modificare**, se necessario, la **configurazione** precedentemente ottenuta.