Relazione di statistica applicata

Analisi esplorativa di dati finanziari

Cardani Alberto
20024060@studenti.uniupo.it
Maccario Alessandro
20023776@studenti.uniupo.it

Anno accademico 2019 - 2020

Introduzione: descrizione dei dati.

A differenza del database originale, abbiamo deciso di ricercare nuovi dati, vista la difficoltà di applicare le metodologie viste durante il corso allo specifico data-set.

La nuova raccolta dei dati è avvenuta tramite la ricerca all'interno del sito di Yahoo Finance e dell'ISTAT. I seguenti link rimandano alle pagine esatte che sono state interrogate per la raccolta dei dati e la formazione del database su cui fare le elaborazioni viste durante il corso, divisi per dato raccolto:

- *Prezzo della Diasorin*[1]
- *Direzione Diasorin*: La direzione del prezzo della Diasorin, "Up" e "Down", è stato ottenuto tramite Excel come differenza fra il prezzo al tempo t, il prezzo al tempo t-1 rapportato al prezzo al tempo t-1: essa risulta interessante per fini predittivi (ad esempio mediante utilizzo del metodo della regressione logistica), ma è stata esclusa da questo data-set ai fini della relazione.

$$\frac{prezzo_t - prezzo_{t-1}}{prezzo_{t-1}}$$

- *Prezzo della QGEN* [2]
- *Prezzo del petrolio* [3]
- *Prezzo della Bayer* [4]
- *Cambio EUR/USD* [5]
- *LevMib (Ftse Mib)*: Il LevMib è un exchange-traded fund (ovvero, un tipo di fondo di investimento avente il fine di replicare l'indice di riferimento del FTSE MIB, il più significativo indice azionario della Borsa Italiana. Esso rappresenta il paniere che racchiude le azioni delle 40 società italiane quotate sul MTA o sul MIV con maggiore capitalizzazione, flottante e liquidità che rappresentano oltre l'80% della capitalizzazione complessiva a livello italiano). [6]
- *Futures sul caffè* [7]
- *Clima fiducia consumatori, NIC, occupati* [8]

L'obiettivo dell'analisi è stato quello dell'esplorazione dei dati e della ricerca di possibili correlazioni fra le variabili, nello specifico fra il prezzo della Diasorin e gli altri prezzi o indici. Questo lavoro preliminare potrebbe essere sviluppato ulteriormente con l'applicazione di tecniche più sofisticate in grado di fornire una percentuale di correttezza nella previsione dell'andamento crescente o decrescente del prezzo futuro delle azioni della Diasorin.

Prime considerazioni sul data-set

Per prima cosa occorre impostare la "directory", ovvero, dobbiamo dire ad R dove deve prendere il database. I comandi sono "setwd()" per indicare la cartella in cui si trova il file e, poichè il database è stato realizzato su Excel occorre richiamare la libreria "readxl" (con "library(readxl)") e impostare la variabile con cui indicheremo ad R di prendere i dati dal data-set per svolgere i comandi che gli chiediamo. Nel nostro caso questo, per importare il dataset si usa:

```
mydatabase <- read_excel("D:/Università/
                         Statistica applicata/
                         Relazione/File_Finale/
                         Modificato_
                         Database_CARDANI-
                         MACCARIO.xlsx")
```

Procediamo ora ai primi risultati.

```
dim(mydatabase)
str(mydatabase)
names(mydatabase)
summary(mydatabase)
head(mydatabase)
```

Il data-set si compone di 12 variabili (inclusa la variabile "data") e lo spazio degli individui è composto da 119 individui, che sono le osservazioni mensili delle variabili instudio nel corso di cinque anni (`dim(mydatabase)`). Attraverso il comando `str()` osserviamo la struttura del database: esso si compone di 10 variabili quantitative, 1 qualitativa (la direzione) e la variabile data; questo comando presenta per ogni variabile le prime osservazioni e anche la dimensione del data-set.
Attraverso il comando `names()` si possono vedere i nomi delle variabili del data-set.

```
> names(mydatabase)
 [1] "Data"                                  "Prezzo - DIA.MI"
 [3] "Direzione - DIA.MI"                    "Prezzo - QGEN"
 [5] "Prezzo - Petrolio"                     "Prezzo - Bayer"
 [7] "Cambio EUR/USD"                        "FTSE.MIB"
 [9] "Fut Caffè"                             "Clima.fiducia.consumatori(2010=100)"
[11] "NIC(2010=100)"                         "Occupati"
```

Osservazione: *L'indice "FTSE.MIB" in realtà è l'indice "levMib". Quest'ultimo è un indicatore che replica l'andamento del "FTSE.MIB", ma per semplicità è stato rinominato con quel nome.*

Se volessimo osservare gli indicatori di sintesi relativi alle variabili trattate singolarmente, dobbiamo utilizzare il comando `summary()`.

```
> summary(mydatabase[,c(-1,-3)])
 Prezzo - DIA.MI  Prezzo - QGEN   Prezzo - Petrolio Prezzo - Bayer
 Min.   : 17.36   Min.   :13.78   Min.   : 33.54    Min.   : 31.36
 1st Qu.: 27.78   1st Qu.:19.89   1st Qu.: 51.65    1st Qu.: 51.30
 Median : 37.43   Median :23.81   Median : 68.57    Median : 69.81
 Mean   : 48.60   Mean   :25.45   Mean   : 72.18    Mean   : 69.88
 3rd Qu.: 70.51   3rd Qu.:32.05   3rd Qu.: 93.33    3rd Qu.: 87.00
 Max.   :121.30   Max.   :42.80   Max.   :113.93    Max.   :114.24
```

Come si può osservare dall'esempio qua sopra (per una questione di spazio sono state riportate soltanto 4 delle 12 variabili) esso riporta: valore minimo e massimo registrato, quartili e media.
Proseguendo nell'esplorazione abbiamo ritenuto interessante fornire una rappresentazione grafica attraverso box-plot appaiati (`boxplot()`) di queste ultime variabili. Questo serve al solo scopo di visualizzare meglio la diversità dei dati.

```
boxplot(mydatabase$`Prezzo - DIA.MI`,
        mydatabase$`Prezzo - QGEN`,
        mydatabase$`Prezzo - Petrolio`,
        mydatabase$`Prezzo - Bayer`,
        names = c("DIA.MI", "QGEN",
        "Petrolio", "Bayer"),
        main = "Prezzi")
```

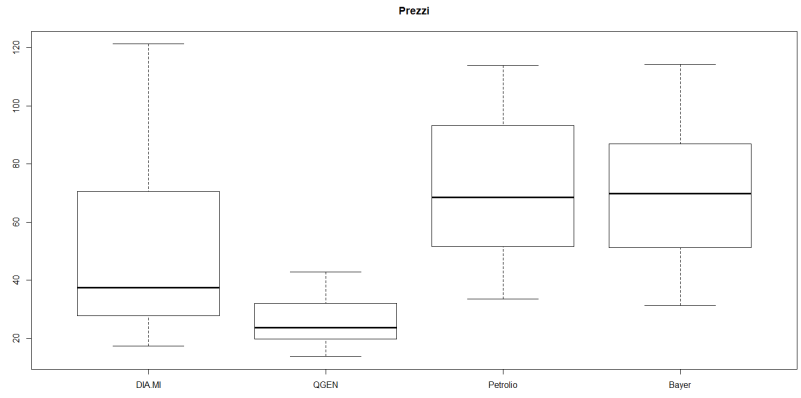


Figura 1

Alla luce del grafico (Figura 1) abbiamo riscontrato una certa somiglianza tra la variabile "petrolio" e la variabile "Bayer", quindi abbiamo ritenuto opportuno confrontarle mediante dei `t.test()`:

```
t.test(mydatabase$`Prezzo - DIA.MI`,
       mydatabase$`Prezzo - QGEN`)
t.test(mydatabase$`Prezzo - Petrolio`,
       mydatabase$`Prezzo - Bayer`)
```

```
> t.test(mydatabase$`Prezzo - DIA.MI`, mydatabase$`Prezzo - QGEN`)

        Welch Two Sample t-test

data:  mydatabase$`Prezzo - DIA.MI` and mydatabase$`Prezzo - QGEN`
t = 8.9379, df = 134.51, p-value = 2.687e-15
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 18.02752 28.27275
sample estimates:
mean of x mean of y
 48.60263  25.45250

> t.test(mydatabase$`Prezzo - Petrolio`, mydatabase$`Prezzo - Bayer`)

        Welch Two Sample t-test

data:  mydatabase$`Prezzo - Petrolio` and mydatabase$`Prezzo - Bayer`
t = 0.79398, df = 235.89, p-value = 0.428
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.412313  8.019629
sample estimates:
mean of x mean of y
 72.18269  69.87903
```

Sono strutturati in modo tale da confrontare le medie delle variabili considerate all'interno del `t.test()`; ovvero, se il p-value assume un valore inferiore a 0.05 (come nel primo caso) rifiuto l'ipotesi nulla (ovvero che la differenza tra le medie sia 0, ovvero che le medie siano identiche). Fornisce poi una stima per un intervallo di confidenza per la possibile differenza tra le medie e una stima delle medie della prima e della seconda variabile in esame. Come si può notare, nel primo caso il p-value è molto basso, dunque le medie sono differenti tra loro molto probabilmente; mentre nel secondo caso abbiamo un p-value molto alto, il che implica che la probabilità che le medie siano molto diverse tra loro risulta essere molto bassa. Infatti, la media della variabile "petrolio" è circa 72.18 e la media della variabile "Prezzo.Bayer" è circa 69.88, mentre le mediane delle due variabili sono pressochè molto simili.

Infine, il comando `head()` fornisce le prime 6 osservazioni relative ad ogni variabile, viceversa il comando `tail()` manda in console le ultime 6 realizzazioni relative agli ultimi individui.

Siti come yahoo-finance, BorsaItaliana, ecc. sono soliti presentare grafici circa l'andamento nel tempo dei prezzi di un titolo. Tuttavia, non vi è la possibilità di mettere a confronto grafici di più titoli per individuare eventuali relazioni tra i vari grafici; a tal proposito l'implementazione con R della comparazione degli andamenti di mercato delle quattro variabili di prezzo (Diasorin, QGEN, petrolio, Bayer) è opportuna. Per farlo e abbiamo optato per un comando diverso dai metodi visti a lezione, comando questo, che verrà riportato solo in parte a causa della sua lunghezza:

```
library(ggplot2)
ggplot(mydatabase, aes(x = Data)) + [...]
```

La scelta dell'uso di questo particolare comando è stata data da un fattore grafico, ovvero che ricorda più i grafici di borsa e per la possibilità di inserire una legenda che ne rende più semplice la lettura.

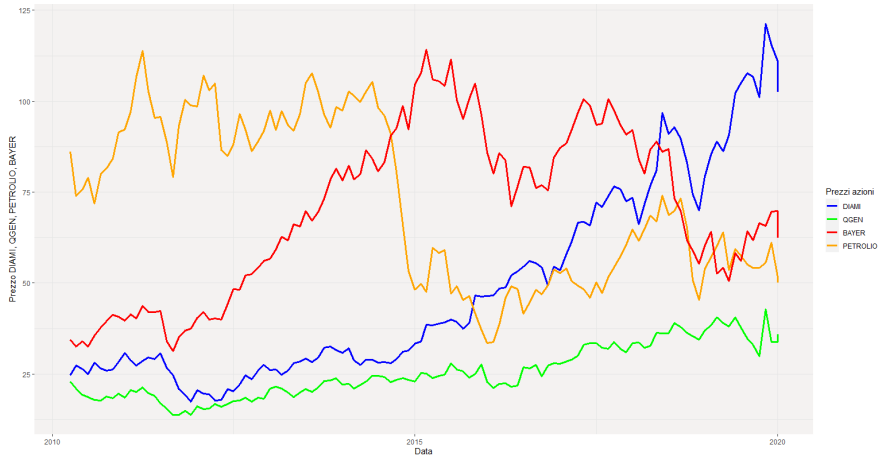


Figura 2

Come si può notare dal grafico (Figura 2) sembrerebbe che l'andamento della QGEN (in verde) segua, in qualche modo, l'andamento della Diasorin (in blu), infatti sul primo tratto la crescita è moderata in entrambi i prezzi, mentre quando il prezzo della Diasorin inizia a crescere, anche il prezzo della QGEN (competitor della Diasorin) comincia a crescere, seppur in modo più pacato. Un altro risultato interessante è dato dall'andamento del prezzo della Bayer (altra casa farmaceutica), sembrerebbe, fino al 2015 che il suo andamento predica l'andamento della Diasorin negli anni successivi: potrebbe essere interessante approfondire questo aspetto ad esempio con un "permutation test". Per quanto riguarda il tratto dal 2015 in poi sembrerebbe che il prezzo della Bayer sia decrescente, dunque, se dal "permutation test" dovessero emergere collegamenti tra Diasorin e Bayer di questo tipo, potremmo aspettarci prezzi inferiori per la Diasorin nei prossimi 5 anni (anche se con l'emergenza coronavirus[10], dato che si parla di case farmaceutiche, potremmo aspettarci un trend crescente per i prezzi di queste tre aziende).

Il prezzo del petrolio, invece, ha andamento opposto a quello delle altre 3 variabili.
Possiamo aspettarci che esista una qualche correlazione (forse non lineare azzardando) tra tutte e quattro le variabili. Inoltre, nello studio della regressione lineare è possibile che il petrolio abbia un coefficiente di correlazione lineare ρ negativo; mentre tra le altre 3 questo coefficiente potrebbe essere positivo.

I fini della relazione abbiamo escluso la variabile "data" e la variabile "direzione":

```
db.finale <- data.frame(mydatabase[,c(-1,-3)])
```

per farlo è stato creato un data.frame (`data.frame()`). Essenzialmente al data-set iniziale sono state tolte le le variabili 1 e 3. Utilizzando i soliti comandi `dim(db.finale)` e `names(db.finale)` infatti:

```
> dim(db.finale)
[1] 119  10
> names(db.finale)
 [1] "Prezzo...DIA.MI"                    "Prezzo...QGEN"
 [3] "Prezzo...Petrolio"                  "Prezzo...Bayer"
 [5] "Cambio.EUR.USD"                     "FTSE.MIB"
 [7] "Fut.Caffè"                          "Clima.fiducia.consumatori.2010.100."
 [9] "NIC.2010.100."                      "Occupati"
```

Concludendo, diamo una rappresentazione grafica globale delle variabili:

```
pairs.panels(db.finale, pch = 1,
             cex.cor = 3, smooth = FALSE,
             ellipses = FALSE, lm = TRUE, stars = TRUE
```

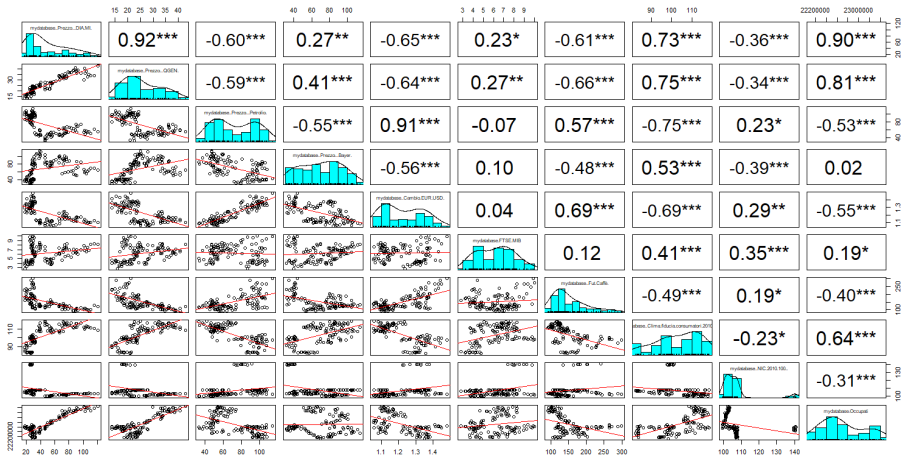


Figura 3

Cluster gerarchico

Il cluster gerarchico si presta molto bene per questa analisi. Infatti, da una prima occhiata del "pairs.panels" iniziale sembra si possa individuare una mistura formata da 2 gruppi.

```
hc.mydatabase <- hclust(dist(scale(db.finale,
                        center = T, scale = T)),
                        method = "complete")
```

Consideriamo, quindi, le distanze tra i singoli individui attraverso il comando "hclust()" racchiusi nella matrice delle distanze (dist()).
Le variabili sono state riscalate, in quanto le grandezze per ogni individuo sono molto eterogenee tra loro.
Il metodo più adatto all'analisi sembra essere il metodo "complete": consideriamo la distanza massima tra gli individui; Dal grafico sarà possibile riscontrare che questo metodo risulta essere il più adatto ai nostri fini rispetto agli altri studiati.

```
plot(hc.mydatabase, hang = -1)
k = 2
sottoinsiemi.hclust <- rect.hclust(hc.mydatabase,
                                   k = k)
hc.cluster <- cutree(hc.mydatabase, k = k)
```

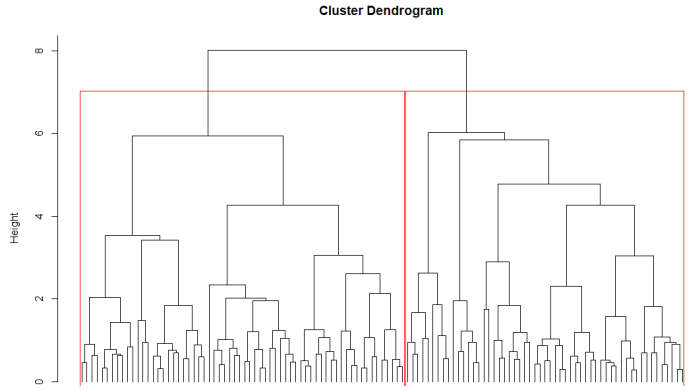


Figura 4

Rappresento con il comando `plot(hc.mydatabase, hang = -1)` il dendrogramma, raffigurante le distanze tra i singoli individui mediante altezze. Da questo si possono notare (come dal grafico: Figura 4) $k = 2$ gruppi distinti, dunque fissato k, mediante il comando `rect.hclust()` tagliamo il

grafico mediante rettangoli rossi; il singolo rettangolo conterrà tutti gli individui appartenenti ad un gruppo. Per scindere i due gruppi in R si utilizza il comando `cutree()`.

```
> hc.cluster
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [49] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [97] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Questo assegna "1" e "2" a tutti gli individui che appartenenti, rispettivamente, al primo e al secondo rettangolo nel dendrogramma: ovvero ai due gruppi.

```
library(psych)
pairs.panels(db.finale, pch = 21,
             cex.cor = 3, bg = (1:k)[hc.cluster],
             ellipses = F, lm = T,
             smooth = F, stars = T)
```

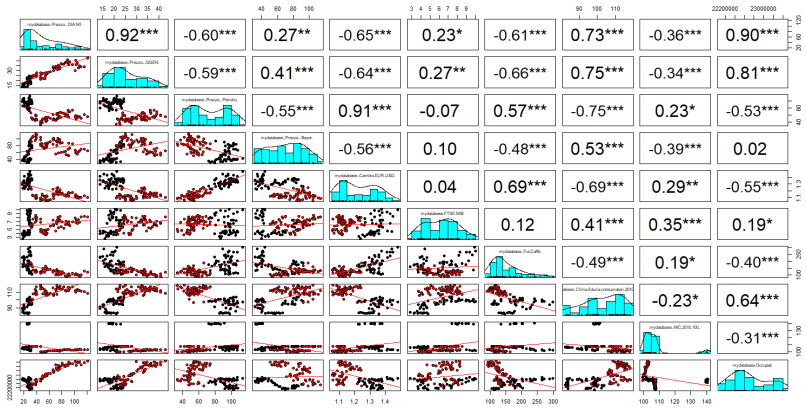


Figura 5

Se ora rappresentiamo il pairs.panels di tutte le variabili, mediante "bg = (1:k)[hc.cluster]" coloriamo gli individui con due colori distinti a seconda del gruppo di appartenenza (come si può vedere dal grafico. Figura 5); in questo modo prendiamo il vettore creato dalla funzione `cutree()` ed assegnamo ad ogni singola unità di questo vettore ogni individuo del data-set. In altre parole, ogni individuo appartenente al primo gruppo verrà colorato con il colore "1" (ovvero il "nero"), mentre gli individui del secondo gruppo verranno colorati con il colore "2" ("rosso").
Si può, infine, notare che dagli istogrammi sulla diagonale principale del pairs.panels che sono presenti delle misture di due gruppi, dunque le considerazioni iniziali sono rinforzate da questo risultato.

Cluster non gerarchico

Oltre al clustering gerarchico appena svolto, abbiamo voluto applicare anche la metodologia del clustering non gerarchico per poter confrontare i risultati e per visualizzare eventuali differenze.
L'algoritmo del k-means si basa sulla scelta a priori del numero di cluster da cercare di individuare. Dal pairs.panel precedente, avevamo deciso che fossero presenti due cluster all'interno della nostra popolazione.
Viene quindi scelto:

```
k <- 3
```

Con il comando `kmeans()` si applica la clusterizzazione non gerarchica e si scala il data-set considerato date le diverse unità di misura presenti nelle differenti variabili: ogni variabile standardizzata e quindi ad ogni osservazione verrà sottratta la media della variabile corrispondente e divisa per la sua deviazione standard diviso n, ovvero il numero delle osservazioni.

```
kmeans.mydatabase <- kmeans((scale(db.finale,
                                   center = T,
                                   scale = T)), k)
```

Applichiamo quindi il comando `pairs.panels()` con l'accortezza di colorare le osservazioni che ricadono all'interno di un gruppo o dell'altro tramite differenti colori.

```
pairs.panels(db.finale, pch = 21,
             bg = (1:k)[kmeans.mydatabase$cluster],
             ellipses = F, lm = T, smooth = F,
             cex.cor = 2, stars = T)
```

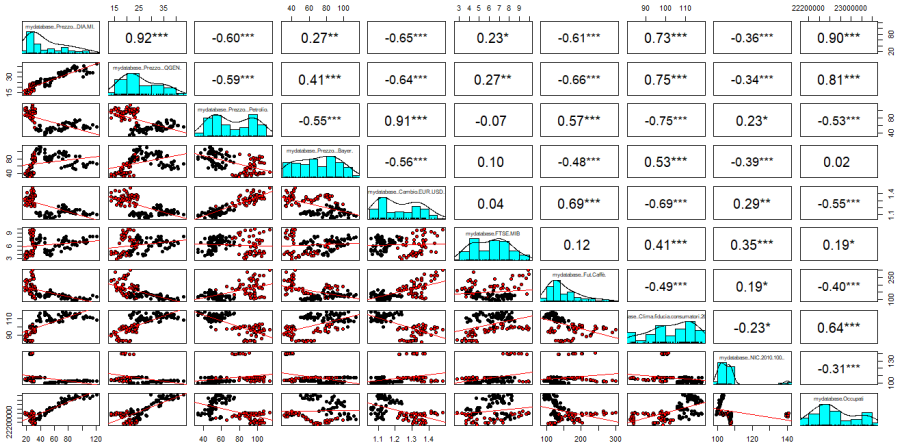


Figura 6

Anche se rispetto al pairs.panels precedente i colori sono stati invertiti, il risultato appare identico.
Per fornire una misura di quanto "corretta" sia stata la procedura di clustering con metodo del k-means, si procede al calcolo del valore dell'R^2, equivalente alla proporzione di varianza spiegata dal partizionamento indotto dal clustering: la metodologia in esame mira a costruire gruppi che siano al loro interni omogenei, quindi con varianza interna minimizzata. Allo stesso tempo, si cerca la massimizzazione della varianza tra gruppi. L'R^2 fornisce calcolato come:

```
r.quadro <- (1-kmeans.mydatabase$tot.withinss/
              kmeans.mydatabase$totss)
```

fornisce una misura di quanta varianza viene spiegata dalla totalità dei gruppi.
Piuttosto che prendere il valore di R^2se ne studia la sua variazione al variare dei k gruppi.
Ricordandoci che con un numero maggiore di k, la varianza interna diminuisce e aumenta l'R^2, applichiamo la riga di codice appena vista con k = 2. Il risultato sarà pari a:

```
> r.quadro <- (1-kmeans.mydatabase$tot.withinss/kmeans.mydatabase$totss)
> r.quadro
[1] 0.4997155
```

Aumentando k e portandolo a $k = 3$ e reiterando la procedura otteniamo R^2 pari a:

```
> r.quadro <- (1-kmeans.mydatabase$tot.withinss/kmeans.mydatabase$totss)
> r.quadro
[1] 0.6083264
```

Reiterando con $k = 4$, otteniamo un R^2 pari a:

```
> r.quadro <- (1-kmeans.mydatabase$tot.withinss/kmeans.mydatabase$totss)
> r.quadro
[1] 0.6300715
```

La variazione massima sembra essere presente fra $k = 2$ e $k = 3$.
La metodologia del k-means, analiticamente, sembra indicare la presenza di tre cluster all'interno del data.set a differenza del clustering gerarchico nel quale si erano scelti due cluster.

Graficamente:

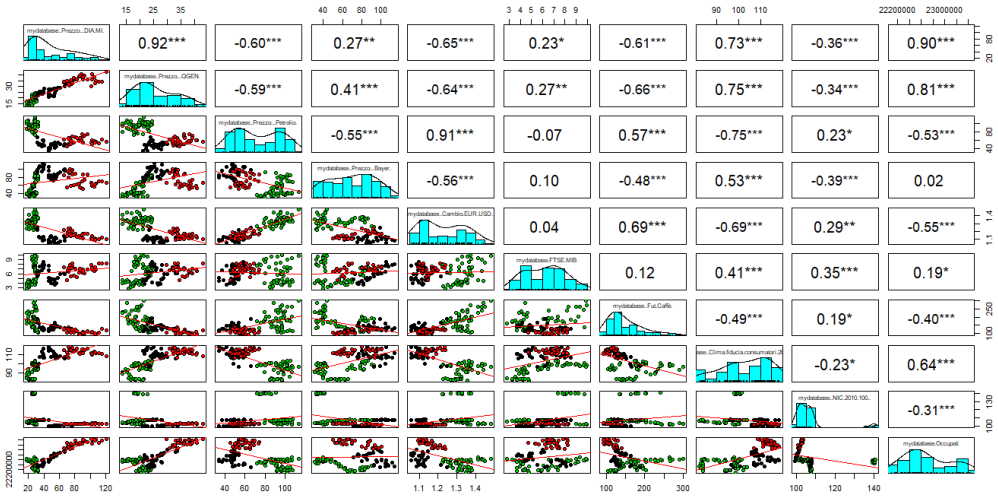


Figura 7

Questa discrepanza fra i due risultati può essere dovuta ad una delle caratteristiche della metodologia del k-means: essa è adatta per individuare cluster sferici. Come si può vedere dal pairs.panels non sono facilmente individuabili cluster che si dispongono in tale maniera.

Analisi delle componenti principali

Con l'analisi successiva ci siamo focalizzati sull'applicare l'ACP, ovvero l'analisi delle componenti principali. Poichè il nostro data-set presenta 10 variabili quantitative (esclusa la data e la direzione del prezzo della Diasorin), abbiamo voluto cercare di ridurre il numero di variabili per spiegare più facilmente la variabilità del nostro fenomeno poichè l'ACP basa proprio i suoi obiettivi sulla riduzione della dimensione del data-set di partenza.
Tramite la funzione `prcomp()` e gli argomenti `scale = TRUE` e `center = TRUE`, si sono scalate le variabili per evitare che la variabilità di alcune di esse potesse essere preponderante sulla variabilità delle restanti.

```
acp.data <- prcomp(db.finale, scale = T, center = T)
```

Col successivo comando `summary(acp.data)` si chiede ad R di ritornare il riepilogo della variabile appena creata, ottenendo:

```
> summary(acp.data)
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8    PC9    PC10
Standard deviation     2.3413 1.2347 1.0784 0.9061 0.74060 0.41225 0.35045 0.26105 0.2408 0.20580
Proportion of Variance 0.5482 0.1525 0.1163 0.0821 0.05485 0.01699 0.01228 0.00681 0.0058 0.00424
Cumulative Proportion  0.5482 0.7006 0.8169 0.8990 0.95387 0.97087 0.98315 0.98997 0.9958 1.00000
```

Come si può vedere, ci vengono forniti tre risultati importanti: la deviazione standard, equivalente alla radice quadrata della varianza, che rappresenta la quantità di varianza spiegata dalla *i-esima* componente principale (sotto radice quadrata) in termini assoluti. In termini relativi, viene anche mostrata la proporzione di varianza spiegata dalla *i-esima* componente e la proporzione di varianza cumulata.
Sarà da quest'ultima riga di dati che andremo a decidere quante componenti principali mantenere per spiegare la maggiore variabilità possibile, ma mantenendo parsimonioso il modello scelto, ovvero scegliendo una quantità k di componenti principali minore delle p-variabili originali.

Per prendere una decisione sul numero di componenti si hanno due approcci principali: il primo approccio si basa sulla costruzione del F.E.V., ovvero il grafico della Fraction of Explained Variance, la frazione di varianza spiegata. Viene definito come:

$$FEV(i) = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{n} \lambda_j}$$

Dove al numeratore si ha la varianza spiegata dalle prime *i-esime* componenti principali e al denominatore la varianza totale, corrispondente alla traccia della matrice di varianza e covarianza.

Graficamente:

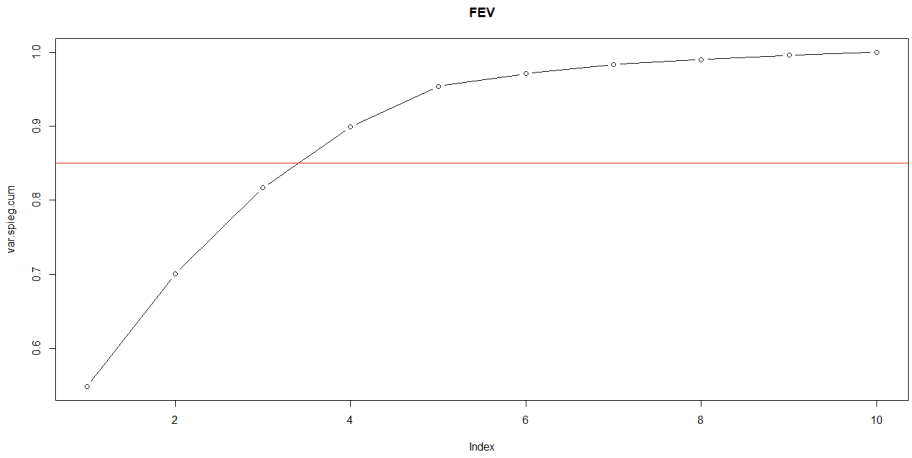


Figura 8

La linea in rosso rappresenta la soglia da noi decisa per definire la quantità di variabilità spiegata dalle prime componenti principali. Fissata quindi una soglia c appartenente all'intervallo (0,1), si sceglie il numero di componenti principali tali che:

$$k = \min\{k : FEV(k) > c\}$$

Nel nostro caso il numero da tenere è pari a 3 componenti principali.

Il secondo approccio riguarda la costruzione del grafico denominato screeplot, richiamato tramite la funzione `screeplot()`. Tale grafico di tipo qualitativo basa la scelta delle k componenti principali su una metodologia più euristica nella quale si ricerca il "gomito della curva": vengono rappresentati i λ_i che sono ottenuti in ordine decrescente dal calcolo svolto da R tramite la matrice di varianza e covarianza.

Dovrebbe quindi essere scelto il numero di k componenti principali nel punto in cui la curva presenta una riduzione della variazione di decrescita.

```
screeplot(acp.data, type = "l", main = "Screeplot")
```

Graficamente:

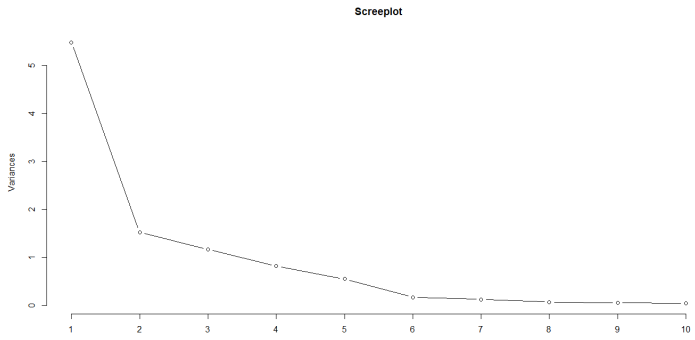


Figura 9

In questo caso, la curva tende a non decrescere più significativamente dalla sesta componente in avanti.
Rispetto al F.E.V. quindi, lo screeplot sembrerebbe suggerire la scelta di un numero doppio di componenti principali da mantenere.
Tenendo presente la volontà di spiegare la maggiore variabilità possibile del fenomeno, ma allo stesso tempo creare un modello parsimonioso che utilizzi un numero ristretto di nuove variabili, scegliamo le prime tre componenti che spiegano circa l'82% della variabilità.
Disegniamo quindi i barplot associati alle componenti principali. Per farlo, salviamo i parametri grafici correnti con il comando `par()`, dividiamo lo spazio grafico in tre righe e due colonne tramite il comando:

```
par(mfrow = c(3,2))
```

E applichiamo quindi il ciclo for per ottenere tutti i grafici dei barplot e focalizziamoci solamente sul primo il quale contiene le prime sei CP:

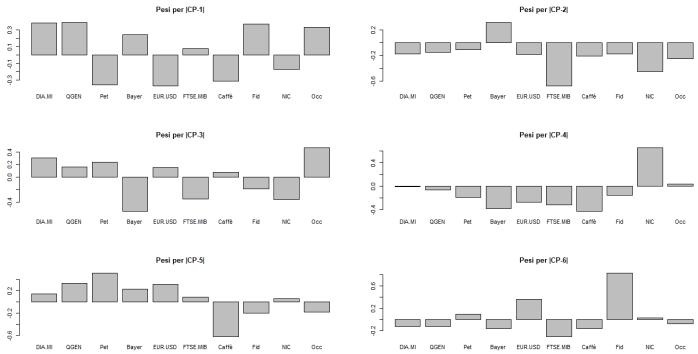


Figura 10

La prima componente principale mostra una relazione diretta fra i prezzi delle tre aziende, la fiducia dei consumatori e il numero degli occupati. Tale gruppo di variabili sono in relazione inversa con il prezzo del petrolio, il cambio EUR/USD e principalmente il valore dei future sul caffè.
La seconda componente principale rappresenta una relazione inversa fra il prezzo della Bayer e tutte le altre variabili, ma principalmente con l'indicatore LEV.MIB replicante il FTSE.MIB.
La terza e ultima componente principale che prendiamo in considerazione presenta un comportamento opposto del prezzo della Bayer rispetto al numero di occupati.
Osserviamo infine le nuove variabili estraendo dalla variabile acp.data la nuova matrice dei dati:

```
nuove.var <- acp.data$x
```

Guardiamone prima l'`head()`:

```
head(nuove.var)
```

```
> head(nuove.var)
          PC1       PC2       PC3      PC4         PC5         PC6         PC7         PC8         PC9         PC10
[1,] -1.931551 -3.182232 -1.225751 1.910199  0.68908237  0.18575511 -0.36501887 -0.07443649 -0.17882369  0.44112020
[2,] -1.662447 -2.234705 -1.150303 2.655793  0.02747901 -0.00431584 -0.25249577  0.11605022 -0.19914623  0.11563869
[3,] -2.044903 -2.207964 -1.195684 2.404819 -0.39232050 -0.12189453 -0.03445535  0.25393611 -0.11655032 -0.09148521
[4,] -2.451831 -2.839180 -1.285189 1.928766 -0.24856645  0.02970861 -0.12511939 -0.07825852 -0.07090966 -0.04422616
[5,] -2.300294 -2.239736 -1.237109 2.282853 -0.54623134 -0.03678280  0.12376526 -0.10832254  0.03124709 -0.23619519
[6,] -2.576355 -2.803938 -1.110537 1.718315 -0.22686429  0.29445140  0.22028337 -0.38169610  0.28788251  0.06233950
```

E poi il pairs.panels riferito alle sole prime tre componenti principali:

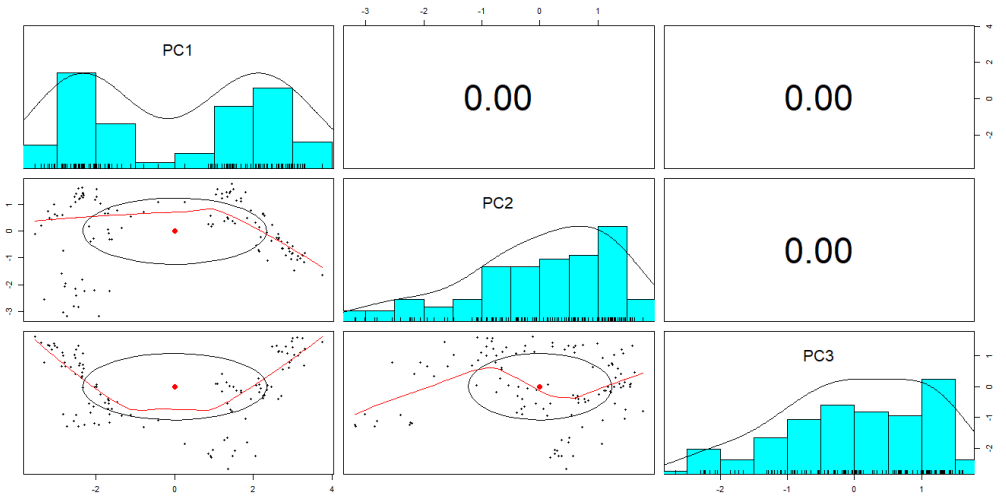


Figura 11

Regressione Lineare

Con il metodo della regressione lineare vogliamo verificare se sia possibile approssimare il modello con un modello lineare. Sia $Y = Prezzo - Diasorin$ la variabile da spiegare e siano tutte le altre (ad esclusione di data e direzione) i regressori $X_1, \ldots, X_p$; allora creiamo un nuovo data.frame che escluda la variabile da spiegare e le altre:

```
d1 <- data.frame(mydatabase[c(-1,-2,-3)])
```

Stimiamo, dunque, i coefficienti β_i del modello lineare con il comando `lm()`

```
linear.model1 <- lm(mydatabase$`Prezzo - DIA.MI`   .,
                    data = d1)
```

Richiamando poi `linear.model1` otteniamo la lista di tutti i coefficienti relativi alle variabili e l'intercetta.

```
> linear.model1

Call:
lm(formula = mydatabase$`Prezzo - DIA.MI` ~ ., data = d1)

Coefficients:
                       (Intercept)                       Prezzo...QGEN
                         -7.235e+02                           1.193e+00
                  Prezzo...Petrolio                      Prezzo...Bayer
                         -4.882e-04                          -5.803e-02
                     Cambio.EUR.USD                           FTSE.MIB
                         -5.645e+00                           1.830e+00
                        Fut.Caffè  Clima.fiducia.consumatori.2010.100.
                         -1.164e-01                           3.234e-02
                    NIC.2010.100.                            Occupati
                         -3.235e-01                           3.480e-05
```

Mentre se volessimo avere maggiori informazioni dovremmo utilizzare il comando `summary()`:

```
summary(linear.model1)
```

Esso fornisce informazioni sui residui del modello (valori di minimo e massimo, quartili e mediana) e la stima dei coefficienti (e dell'intercetta) del modello di regressione lineare. Inoltre presenta la stima degli errori standardizzati per ogni coefficiente e i valori assunti dalla statistica test. Un risultato interessante e molto utile, è dato dai p.value (per ogni coefficiente): se questo assume valori troppo elevati allora accetto H_0 (ovvero che il coefficiente su cui si effettua il test d'ipotesi potrebbe essere nullo); viceversa se questa probabilità condizionata è inferiore a 0.05 (ovvero che $\mathbb{P}(|t_{n-p-1}| > t_{calcolato}|H_0) > p\_v$) probabilmente il coefficiente non sarà nullo e rifiuto H_0.

```
> summary(linear.model1)

Call:
lm(formula = mydatabase$`Prezzo - DIA.MI` ~ ., data = d1)

Residuals:
     Min       1Q   Median       3Q      Max
-15.3234  -3.5194  -0.5646   3.4103  25.1696

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                       -7.235e+02  1.175e+02  -6.157 1.27e-08 ***
Prezzo...QGEN                      1.193e+00  2.702e-01   4.417 2.37e-05 ***
Prezzo...Petrolio                 -4.882e-04  8.350e-02  -0.006  0.99535
Prezzo...Bayer                    -5.803e-02  6.309e-02  -0.920  0.35977
Cambio.EUR.USD                    -5.645e+00  1.671e+01  -0.338  0.73610
FTSE.MIB                           1.830e+00  6.330e-01   2.891  0.00463 **
Fut.Caffè                         -1.164e-01  2.654e-02  -4.385 2.69e-05 ***
Clima.fiducia.consumatori.2010.100.  3.234e-02  1.399e-01   0.231  0.81766
NIC.2010.100.                     -3.235e-01  1.038e-01  -3.116  0.00235 **
Occupati                           3.480e-05  4.930e-06   7.059 1.64e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.142 on 109 degrees of freedom
Multiple R-squared:  0.9368,    Adjusted R-squared:  0.9316
F-statistic: 179.6 on 9 and 109 DF,  p-value: < 2.2e-16
```

Ancora, fornisce rapporto di variabilità spiegato dal modello R^2 e il coefficiente di determinazione corretto $\overline{R}^2$, dove questi sono rispettivamente:

$$R^2 = \frac{Var(\widehat{Y})}{Var(Y)} \qquad \overline{R}^2 = \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

Infine, la F-statistic testa se contemporaneamente tutti i coefficienti calcolati siano nulli; in questo caso, essendo il p.value prossimo a 0 rifiuto H_0 e quindi i coefficienti potrebbero non essere tutti uguali a 0.

Analisi dei residui e punti di leva

Un aspetto importante ai fini della regressione lineare riguarda l'analisi dei residui e dei punti di leva, a tal proposito implementiamo su R i seguenti comandi:

```
opar.lm <- par()
par(mfrow = c(2,2))
plot(linear.model1)
par <- opar.lm
```

Questa lista di comandi serve per rappresentare su un unico foglio bianco i quattro grafici che si otterrebbero con `plot(linear.model1)` premendo 4 volte invio.

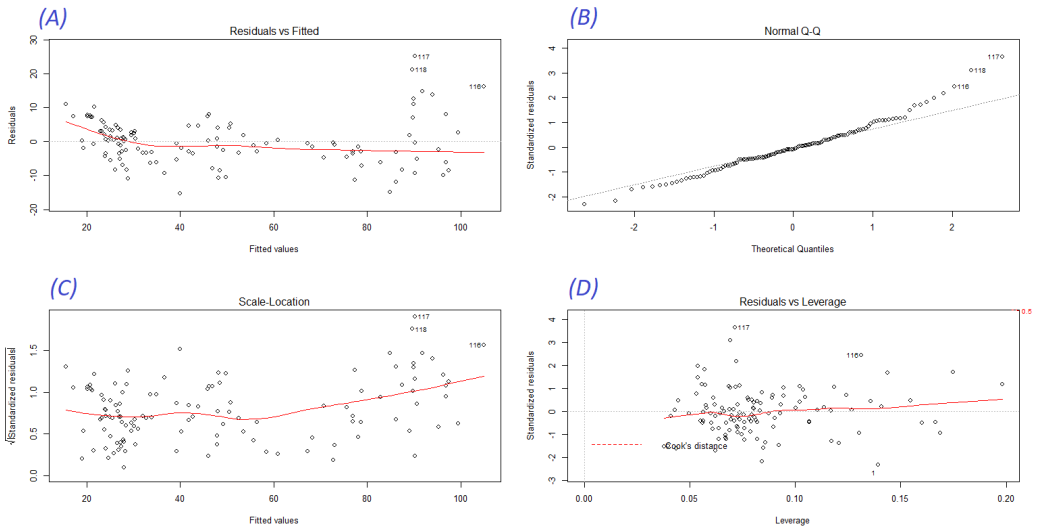


Figura 12

Dai risultati ottenuti sembrerebbe (Figura 12: (A)) che il modello sottostante non sia di tipo lineare, perchè gli errori dovrebbero essere incorrelati con i regressori e le medie condizionate $\mathbb{E}[\mathcal{E}|X]$ ed $\mathbb{E}[\mathcal{E}|Y]$ dovrebbero essere tutte nulle, quindi la funzione di regressione $r(x) = 0$. Deduciamo inoltre che gli individui 110, 117, 118 sono outliers per il modello.

Dal grafico "QQplot" (Figura 12, (B)), che mette in relazione i quantili di una normale $\mathcal{N}(0, 1)$ con i residui standardizzati del modello, sembrerebbe che sia presente una qualche normalità, seppur non fortissima specialmente nelle code.

Il grafico (C) mette in relazione i residui standardizzati in valore assoluto sotto radice contro i valori fittati. I residui standardizzati avranno media nulla e varianza pari a 1: se la linea in rosso fosse orizzontale equivarrebbe a sostenere che la varianza dei residui contro i valori fittati è costante come da ipotesi del modello di regressione lineare; nel nostro caso sembra esserci una tendenza alla crescita della varianza al variare dei valori previsti.

Il grafico (D) mostra eventuali punti outliers che dovranno essere analizzati nel dettaglio mediante il prossimo grafico.
Per quanto riguarda l'analisi dei punti di leva, attraverso l'`influenceplot()` possiamo vedere se il modello lineare sia influenzato e deviato da un qualche individuo del campione. Per farlo:

```
library(car)
influencePlot(linear.model1)
```

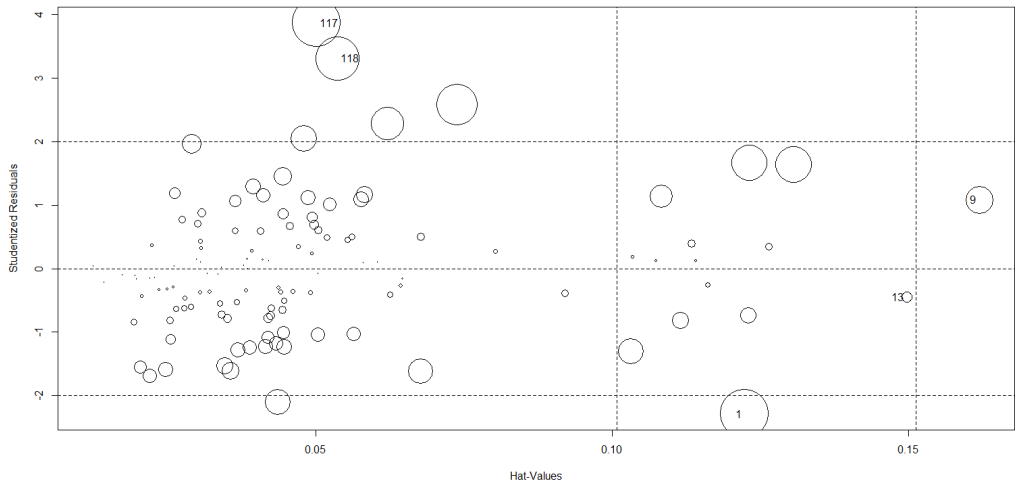


Figura 13

Esso mette in relazione i residui standardizzati con i valori h_{jj} della matrice di proiezione H. Come per gli altri grafici presenta gli outliers, ovvero tutti gli individui che stanno al di fuori dell'intervallo $[-\alpha, \alpha] = [-2, 2]$, ma prima di $2\mathbb{E}[h_{jj}] = 2\frac{p+1}{n}$ (in questo caso 117, 118). Inoltre, presenta i cosiddetti punti di leva: *buoni* ovvero quegli individui che stanno oltre $2\mathbb{E}[h_{jj}]$ ma dentro $[-2, 2]$ (l'individuo 9 e 13); *cattivi*, che sono situati oltre $2\mathbb{E}[h_{jj}]$ e dall'intervallo $[-2, 2]$ (in questo caso l'individuo 1).

Akaike information criterion (AIC)

Nella selezione delle variabili da inserire nel modello di regressione lineare un metodo dell'AIC con modalità backward.

```
akaike <- step(linear.model1, direction = "backward")
```

Con il comando `step()` si selezionano le migliori variabili per il modello lineare.

```
> akaike <- step(linear.model1, direction = "backward")
Step:  AIC=470.42
mydatabase$`Prezzo - DIA.MI` ~ Prezzo...QGEN + FTSE.MIB + Fut.Caffè +
    NIC.2010.100. + Occupati

                Df Sum of Sq     RSS    AIC
<none>                       5604.8 470.42
- FTSE.MIB       1     500.2  6105.0 478.59
- NIC.2010.100.  1     526.2  6131.0 479.10
- Prezzo...QGEN  1    1031.3  6636.1 488.52
- Fut.Caffè      1    1240.6  6845.4 492.21
- Occupati       1    7559.6 13164.3 570.03
```

Creiamo un dataframe con le variabili individuate dall'AIC:

```
db.akaike <- data.frame(...)
```

Eseguiamo nuovamente quanto fatto col modello lineare, con le nuove variabili.

```
linear.model.akaike <- lm(mydatabase$`Prezzo - DIA.MI`
                          ~., data = db.akaike[,-1])
summary(linear.model.akaike)
```

```
> linear.model.akaike <- lm(mydatabase$`Prezzo - DIA.MI` ~ ., data = db.akaike[,-1])
> summary(linear.model.akaike)

Call:
lm(formula = mydatabase$`Prezzo - DIA.MI` ~ ., data = db.akaike[,
    -1])

Residuals:
    Min      1Q  Median      3Q     Max
-14.827  -4.298  -0.600   3.405  25.061

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -8.105e+02  6.538e+01 -12.397  < 2e-16 ***
mydatabase..Prezzo...QGEN.  1.101e+00  2.415e-01   4.560 1.31e-05 ***
mydatabase..NIC.2010.100.. -2.725e-01  8.367e-02  -3.257  0.00149 **
mydatabase.FTSE.MIB         1.675e+00  5.276e-01   3.176  0.00193 **
mydatabase..Fut.Caffè.     -1.156e-01  2.312e-02  -5.001 2.11e-06 ***
mydatabase.Occupati         3.819e-05  3.094e-06  12.345  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.043 on 113 degrees of freedom
Multiple R-squared:  0.9363,    Adjusted R-squared:  0.9335
F-statistic: 332.3 on 5 and 113 DF,  p-value: < 2.2e-16
```

Per quanto riguarda analisi dei residui e punti di leva valgono le stesse cose dette primo.

Presentiamo il pairs.panels con le nuove variabili.

```
pairs.panels(db.akaike, pch = 1,
             cex.cor = 3, smooth = FALSE,
             ellipses = FALSE, lm = TRUE, stars = TRUE
```

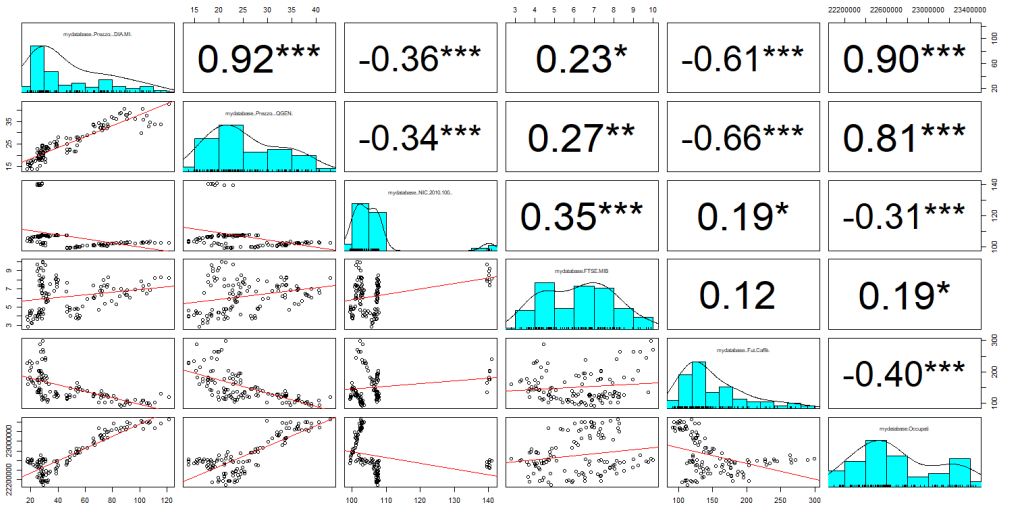


Figura 14

Approfondimento: test sui residui

Dopo aver stimato il modello di regressione, dobbiamo verificare che siano valide le ipotesi di base sugli errori legati agli assunti di base del modello della regressione lineare attraverso alcuni test statistici, per essere certi che tale modello sia il più adatto a spiegare il fenomeno sotto osservazione.
Ricordando che le ipotesi da verificare per l'analisi dei residui sono: Gli errori si distribuiscono come una normale con vettore delle medie nullo e matrice di varianza e covarianza uguale alla varianza costante moltiplicata per la matrice identità; La covarianza dell'errore *i-esimo* con l'errore *j-esimo* dovrà essere nulla (quando i è diverso da j).
Carichiamo la libreria `lmtest` e `car` per l'effettuazione dei nostri test. Verifichiamo quindi che la media degli errori non sia significativamente diversa da zero grazie al test "t di Student":

```
residui.linear.model <- residuals(linear.model.akaike)
t.test(residui.linear.model)
```

E con un p-value pari ad 1, siamo confidenti nel sostenere l'ipotesi nulla di media dei residui uguale a 0.

```
> t.test(residui.linear.model)

        One Sample t-test

data:  residui.linear.model
t = 4.6881e-16, df = 118, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -1.251092  1.251092
sample estimates:
   mean of x
2.961858e-16
```

Verifichiamo la normalità degli errori col test di Shapiro-Wilk:

```
shapiro.test(residui.linear.model)
```

```
> shapiro.test(residui.linear.model)

        Shapiro-Wilk normality test

data:  residui.linear.model
W = 0.96881, p-value = 0.007233
```

Poichè il p-value fornisce una misura di quanto sia buona H_0 nel supportare i valori osservati, e dato che l'ipotesi nulla è uguale ad affermare che gli errori si distribuiscono come una normale, concludiamo che poichè si ha un p-value molto vicino allo 0 si rifiuta H_0: gli errori non presentano distribuzione normale.
Come specifica il testo di Vito Ricci [9], anche se uno solo dei test ritorna un esito negativo, il metodo di stima dei minimi quadrati non può essere considerato valido e bisogna operare diversamente per la stima dei parametri.
Affrontiamo ancora i test gli ultimi test per essere certi della non gaussianità dell'errore.
Applichiamo quindi il test sull'eteroschedasticità (come ipotesi alternativa):

```
bptest(linear.model.akaike)
```

```
> bptest(linear.model.akaike)

        studentized Breusch-Pagan test

data:  linear.model.akaike
BP = 24.417, df = 5, p-value = 0.0001805
```

Il p-value minore di qualsiasi alfa scelto, rende non accettabile l'ipotesi nulla di omoschedasticità.
Guardiamo inoltre al test sulla varianza per assicurarci della sua costanza:

```
ncvTest(linear.model.akaike)
```

```
> ncvTest(linear.model.akaike)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 27.60901, Df = 1, p = 1.4849e-07
```

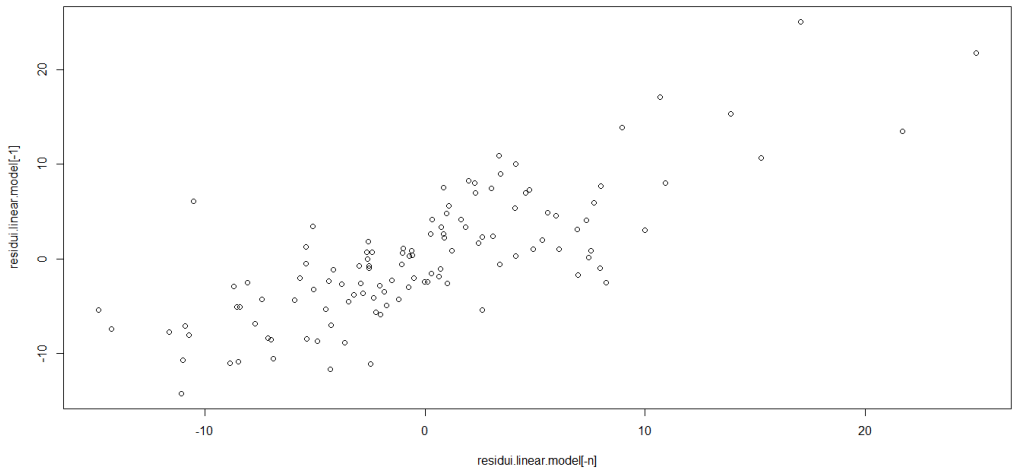
Se l'ipotesi nulla nel caso dell'`ncvTest()` è uguale ad affermare che la varianza sia costante, il p-value con un valore prossimo allo zero, implica il non accettare l'ipotesi nulla e quindi il non rifiutare l'ipotesi alternativa: la varianza non è costante.
Infine, controlliamo che la covarianza dell'errore *i-esimo* con l'errore *j-esimo* sia nulla, coiè che non sia presente autocorrelazione fra gli errori, ovvero che non vi sia correlazione seriale. Per farlo, applichiamo il test di Durbin-Watson:

```
dwtest(linear.model.akaike)
```

```
> dwtest(linear.model.akaike)

        Durbin-Watson test

data:  linear.model.akaike
DW = 0.41475, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

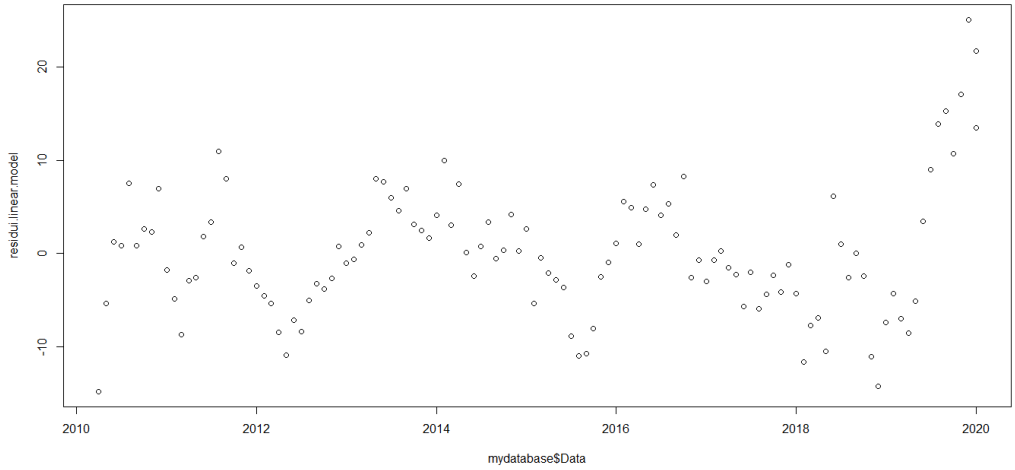
Il cui risultato è l'unico, oltre al t-test sulla media a segnalarci il rispetto delle ipotesi sul modello: il p-value uguale in pratica all'rrore macchina ci garantisce la possibilità di non rifiutare l'ipotesi nulla.
Oltre al test di Durbin-Watson, graficamente si può cogliere l'assenza di correlazione seriale tramite il grafico che confronta i residui (in ordinata) verso i residui precedenti (in ascissa) che non dovrebbe rilevare alcun pattern evidente (nessun TREND).

```
n <- length(residui.linear.model)
plot(residui.linear.model[-n],
     residui.linear.model[-1])
```



È chiaramente presente un pattern o correlazione positiva tra i valori.
Inoltre, poichè i nostri dati e quindi i residui sono ordinati temporalmente (mensilmente), è necessario anche considerare che non ci siano dipendenze dal tempo. Lo si valuta, considerando un diagramma a dispersione dei residui contro il tempo: se la distribuzione non cambia nel tempo, possiamo affermare l'assenza di correlazione.

```
plot(residui.linear.model   mydatabase$Data)
```



Dal grafico è visibile un andamento stagionale che dovrebbe essere eliminato, ma il cui procedimento non esaminiamo in questa sede.
Visti i precedenti risultati sugli altri test che non permettevano l'accettazione del modello di regressione lineare, concludiamo che non è possibile usare tale modello per spiegare il fenomeno in esame.

Modello polinomiale

Poichè dai vari test sul modello lineare, i risultati ci hanno mostrato la non gaussianità dei dati, vorremmo provare ad applicare un modello di regressione polinomiale per inserire dei termini nel modello di regressione lineare (lineare nei coefficienti) che siano di grado superiore ad uno (nei regressori). Così facendo, stiamo cercando quale sia il modello di regressione polinomiale che meglio si adatta ai dati in analisi.
Per farlo, applichiamo la funzione `poly()` tenendo conto delle variabili ottenute tramite il metodo dell'IC(p):

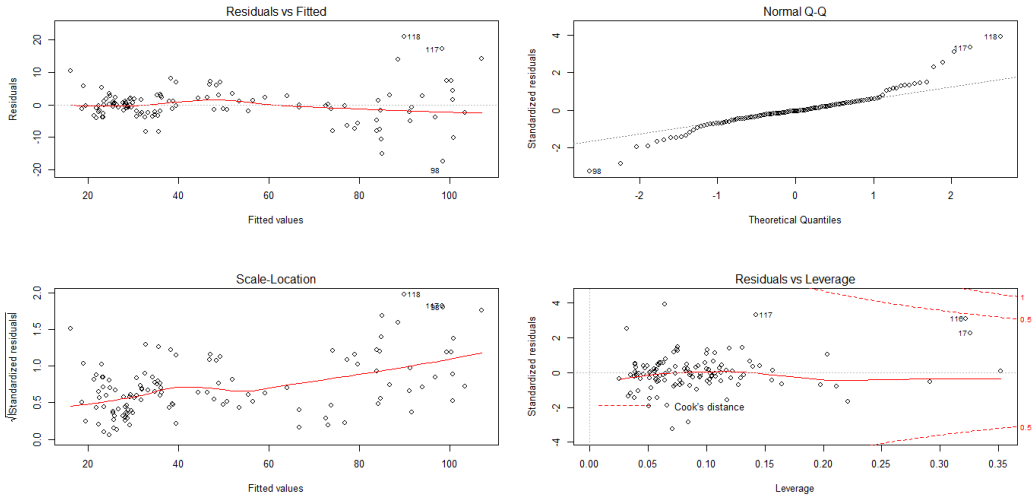
```
fit.model.poly_1 <- lm([...], degree = 2, raw = T))
```

Guardiamo al `summary()` di tale risultato:

```
> summary(fit.model.poly_1)

Call:
lm(formula = db.akaike$`Prezzo - DIA.MI` ~ poly(db.akaike$`Prezzo - QGEN`,
    degree = 2, raw = T) + poly(db.akaike$NIC, degree = 2, raw = T) +
    poly(db.akaike$FTSE.MIB, degree = 2, raw = T) + poly(db.akaike$Fut.Caffè,
    degree = 2, raw = T) + poly(db.akaike$Occupati, degree = 2,
    raw = T))

Residuals:
     Min       1Q   Median       3Q      Max
-17.4203  -2.3511  -0.2488   2.0978  21.0642

Coefficients:
                                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                                      2.394e+04  3.039e+03   7.878 2.81e-12 ***
poly(db.akaike$`Prezzo - QGEN`, degree = 2, raw = T)1  1.574e+00  8.022e-01   1.962   0.0524 .
poly(db.akaike$`Prezzo - QGEN`, degree = 2, raw = T)2 -2.009e-02  1.479e-02  -1.359   0.1771
poly(db.akaike$NIC, degree = 2, raw = T)1             -2.537e-03  9.619e-02  -0.026   0.9790
poly(db.akaike$NIC, degree = 2, raw = T)2             -2.403e-04  2.603e-04  -0.923   0.3581
poly(db.akaike$FTSE.MIB, degree = 2, raw = T)1        -1.018e+01  2.194e+00  -4.641 9.79e-06 ***
poly(db.akaike$FTSE.MIB, degree = 2, raw = T)2         4.070e-02  8.973e-03   4.535 1.50e-05 ***
poly(db.akaike$Fut.Caffè, degree = 2, raw = T)1        6.342e+00  2.744e+00   2.311   0.0227 *
poly(db.akaike$Fut.Caffè, degree = 2, raw = T)2       -3.624e-01  2.109e-01  -1.718   0.0887 .
poly(db.akaike$Occupati, degree = 2, raw = T)1        -2.089e-03  2.628e-04  -7.948 1.97e-12 ***
poly(db.akaike$Occupati, degree = 2, raw = T)2         4.677e-11  5.784e-12   8.086 9.74e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.549 on 108 degrees of freedom
Multiple R-squared:  0.9622,    Adjusted R-squared:  0.9587
F-statistic:   275 on 10 and 108 DF,  p-value: < 2.2e-16
```

Si può notare che sono considerati significativi alcuni regressori elevati al grado due. Sembra quindi che nei nostri dati sia presente un andamento di tipo quadratico.
Guardiamo al plot di questo modello dividendo tramite il comando `par(mfrow = c(2,2))` lo spazio dei grafici:

```
opar <- par()
par(mfrow = c(2,2))
plot(fit.model.poly_1)
par <- opar
```



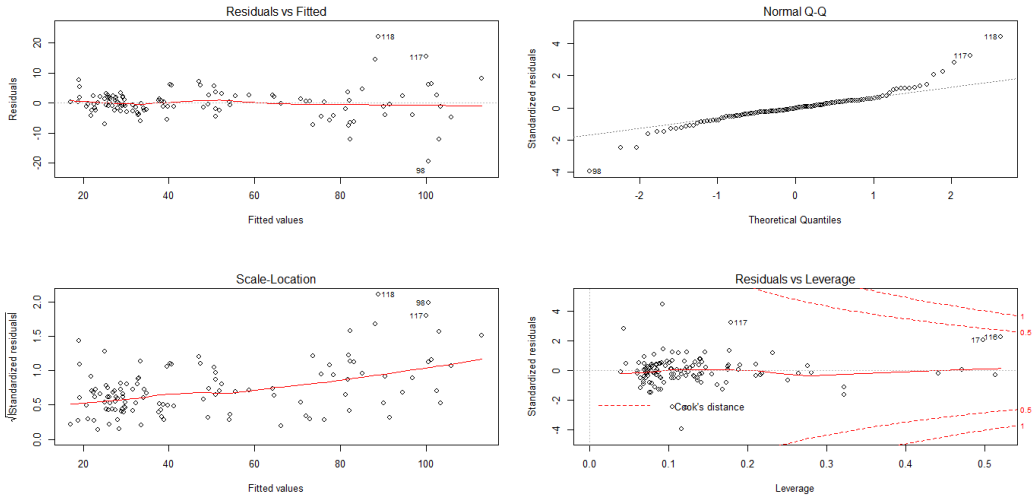
Rispetto al modello lineare di base precedentemente calcolato e rappresentato, sembra essere presente un leggero miglioramento visibile soprattutto dal primo grafico dei residui contro i valori previsti.
Volendo aumentare il valore del grado da usare nei regressori, proviamo a fittare nuovamente il modello con un grado pari a 3:

```
fit.model.poly_2 <- lm([...], degree = 3, raw = T)
```

E guardiamone nuovamente il `summary()`:

```
> summary(fit.model.poly_2)

Call:
lm(formula = db.akaike$`Prezzo - DIA.MI` ~ poly(db.akaike$`Prezzo - QGEN`,
    degree = 3, raw = T) + poly(db.akaike$NIC, degree = 3, raw = T) +
    poly(db.akaike$FTSE.MIB, degree = 3, raw = T) + poly(db.akaike$Fut.Caffè,
    degree = 3, raw = T) + poly(db.akaike$Occupati, degree = 3,
    raw = T))

Residuals:
    Min      1Q  Median      3Q     Max
-19.504  -2.189  -0.180   2.053  22.220

Coefficients:
                                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                                     -5.358e+05  1.989e+05  -2.693  0.00826 **
poly(db.akaike$`Prezzo - QGEN`, degree = 3, raw = T)1  1.543e+00  4.253e+00   0.363  0.71743
poly(db.akaike$`Prezzo - QGEN`, degree = 3, raw = T)2 -6.258e-03  1.621e-01  -0.039  0.96928
poly(db.akaike$`Prezzo - QGEN`, degree = 3, raw = T)3 -2.052e-04  1.917e-03  -0.107  0.91501
poly(db.akaike$NIC, degree = 3, raw = T)1              4.783e-01  4.575e-01   1.045  0.29833
poly(db.akaike$NIC, degree = 3, raw = T)2             -3.235e-03  2.528e-03  -1.280  0.20356
poly(db.akaike$NIC, degree = 3, raw = T)3              5.678e-06  4.415e-06   1.286  0.20126
poly(db.akaike$FTSE.MIB, degree = 3, raw = T)1         1.992e+02  1.315e+02   1.515  0.13288
poly(db.akaike$FTSE.MIB, degree = 3, raw = T)2        -1.813e+00  1.151e+00  -1.576  0.11811
poly(db.akaike$FTSE.MIB, degree = 3, raw = T)3         5.399e-03  3.316e-03   1.628  0.10659
poly(db.akaike$Fut.Caffè, degree = 3, raw = T)1       -2.057e+01  1.268e+01  -1.623  0.10775
poly(db.akaike$Fut.Caffè, degree = 3, raw = T)2        4.067e+00  2.010e+00   2.023  0.04563 *
poly(db.akaike$Fut.Caffè, degree = 3, raw = T)3       -2.316e-01  1.027e-01  -2.255  0.02626 *
poly(db.akaike$Occupati, degree = 3, raw = T)1         7.075e-02  2.613e-02   2.708  0.00793 **
poly(db.akaike$Occupati, degree = 3, raw = T)2        -3.157e-09  1.149e-09  -2.747  0.00710 **
poly(db.akaike$Occupati, degree = 3, raw = T)3         4.694e-17  1.684e-17   2.788  0.00632 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.257 on 103 degrees of freedom
Multiple R-squared:  0.9677,    Adjusted R-squared:  0.963
F-statistic: 205.5 on 15 and 103 DF,  p-value: < 2.2e-16
```

A diminuire è il numero dei possibili regressori da tenere in considerazione, come suggerisce il valore del p-value. Anche se sembra più difficile rifiutare l'ipotesi nulla di nullità dei coefficienti dei regressori.
Proviamo però a focalizzarci sui grafici di questo modello per confrontarne i risultati con la polinomiale precedente di grado 2:



Possiamo quindi concludere che inserire nel modello di regressione lineare dei regressori di terzo grado, migliora ancora il nostro risultato: infatti, come è visibile più chiaramente sia dal grafico dei residui contro i valori previsti e il grafico del QQ-Plot, non considerando gli outliers visibili, entrambi sembrano essere migliorati in termini di linearità e quindi di indipendenza e in termini di distribuzione maggiormente tendente alla gaussianità.

Fonti

[1] Prezzo Diasorin:
https://it.finance.yahoo.com/quote/DIA.MI?p=DIA.MI

[2] Prezzo QGEN:
https://it.finance.yahoo.com/quote/QGEN?p=QGEN&.tsrc=fin-srch

[3] Prezzo del petrolio:
https://it.finance.yahoo.com/quote/CL=F?p=CL=F&.tsrc=fin-srch

[4] Prezzo della Bayer:
https://it.finance.yahoo.com/quote/BAYN.DE?p=BAYN.DE&.tsrc=fin-srch

[5] Cambio EUR/USD:
https://it.finance.yahoo.com/quote/EURUSD%3DX?p=EURUSD%3DX

[6] Indice Ftse Mib:
https://it.finance.yahoo.com/quote/LEVMIB.MI?p=LEVMIB.MI&.tsrc=fin-srch

[7] Prezzi futures sul caffè:
https://it.finance.yahoo.com/quote/KC%3DF?p=KC%3DF

[8] Clima di fiducia dei consumatori, NIC, occupati:
http://dati.istat.it/#

[9] *Principali tecniche di regressione con R*, settembre 2006, sito web: "The R Project for Statistical Computing":
http://cran.r-project.org/doc/contrib/Ricci-regression-it.pdf

[10] Collaborazione Diasorin-QGEN:
https://it.finance.yahoo.com/notizie/diasorin-con-qiagen-nello-sviluppo-071149998.html

Altri riferimenti

Software utilizzati:

- R: *https://www.r-project.org/*
- LaTeX: *https://miktex.org/*