UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA

Corso di Laurea Triennale in Ingegneria Informatica

Relazione Statistica Applicata

Autori:

Antonio LANGELLA

Mat. 0612704729

Davide RISI

Mat. 0612704701

Salvatore PAOLINO

Mat. 0612704705

ANNO ACCADEMICO 2021/2022

INDICE

CAPITOLO 1

ANALISI DEL DATASET

L'obiettivo di questo capitolo è quello di ricorrere agli strumenti della statistica descrittiva per esporre il dataset e, successivamente, analizzare la correlazione tra le variabili.

1.1 Presentazione dei dati

Il dataset assegnato contiene un campione di 100 osservazioni, ciascuna costituita dalla determinazione di 7 variabili quantitative continue: una variabile dipendente Y e 6 regressori X_i. Nello specifico esse sono:

- Y: Prestazioni software calcolatore;
- X_1: Indice standardizzato e centrato della velocità della CPU;
- X_2: Indice standardizzato e centrato della dimensione dell'HDD;
- X_3: Indice standardizzato e centrato legato al numero di processi software;
- X_4: Indice standardizzato e centrato legato all'aging del software;
- X_5: Indice standardizzato e centrato legato alle prestazioni della scheda audio;
- X_6: Indice standardizzato e centrato legato alle prestazioni della RAM.

Di seguito sono riportati i plot di tutte queste variabili, in cui sull'ascissa sono rappresentate le posizioni delle osservazioni campionarie nel dataset, mentre sull'ordinata sono rappresentati i valori delle variabili stesse.

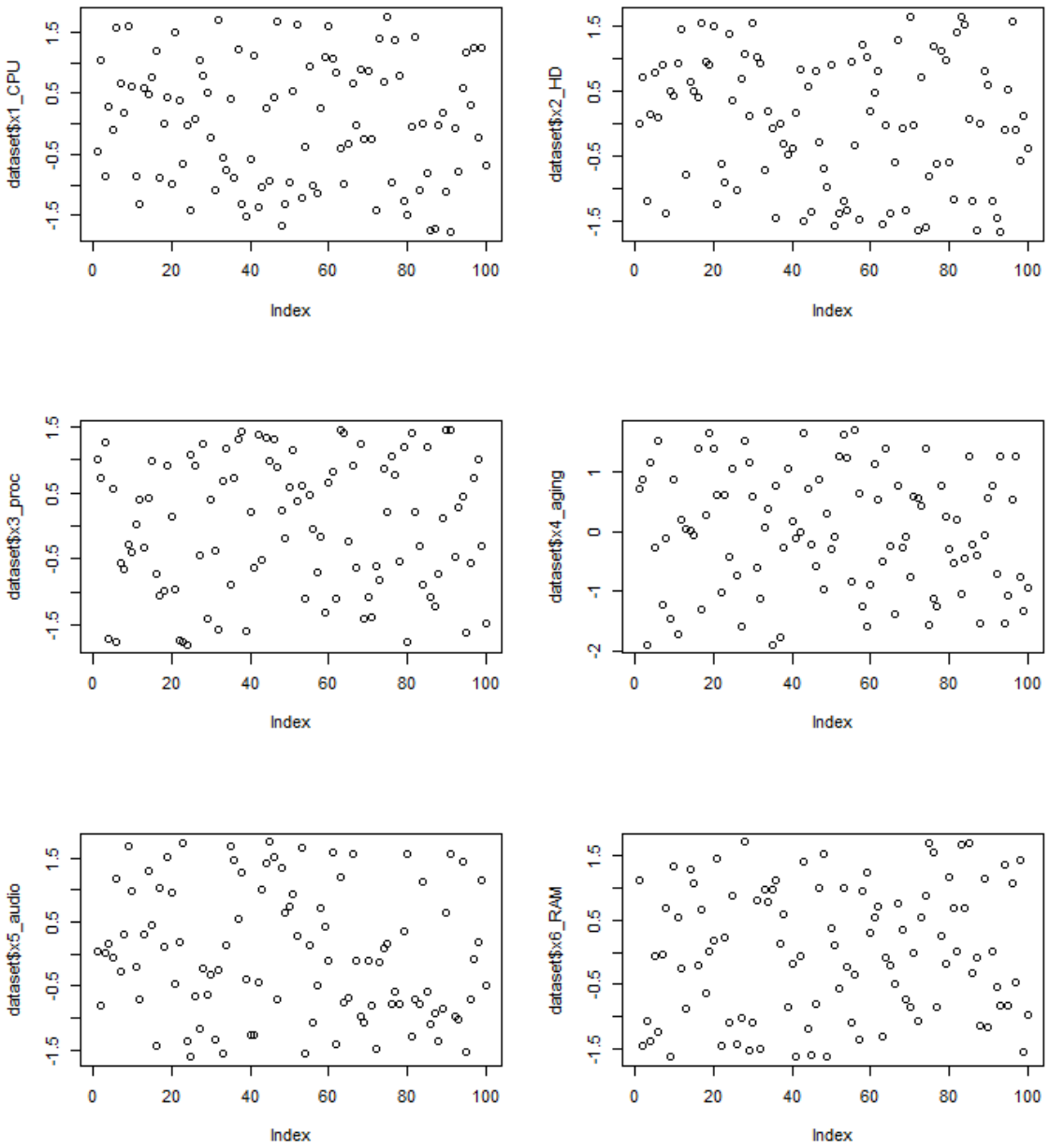

Figura 1.1: Plot variabili X_i

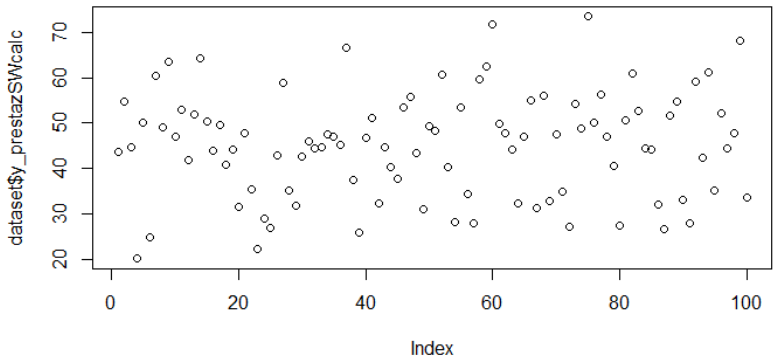

Figura 1.2: Plot variabile Y

Grazie al comando `summary(dataset)` è possibile osservare media, mediana, quartili, massimo e minimo di tutte e 7 le variabili, mentre col comando `var(dataset$variabile)` otteniamo la varianza campionaria delle variabili:

y_prestazSWcalc	x1_CPU	x2_HD	x3_proc	x4_aging	x5_audio	x6_RAM
Min. :20.13	Min. :-1.77049	Min. :-1.67735	Min. :-1.7981	Min. :-1.89497	Min. :-1.6120	Min. :-1.62719
1st Qu.:35.20	1st Qu.:-0.89990	1st Qu.:-0.83732	1st Qu.:-0.7435	1st Qu.:-0.78246	1st Qu.:-0.7844	1st Qu.:-0.85442
Median :45.56	Median :-0.01438	Median : 0.08109	Median : 0.1291	Median :-0.02367	Median :-0.1045	Median :-0.01476
Mean :45.08	Mean : 0.00000	Mean : 0.00000	Mean : 0.0000	Mean : 0.00000	Mean : 0.0000	Mean : 0.00000
3rd Qu.:52.22	3rd Qu.: 0.80668	3rd Qu.: 0.85869	3rd Qu.: 0.9105	3rd Qu.: 0.77336	3rd Qu.: 0.9426	3rd Qu.: 0.88105
Max. :73.43	Max. : 1.74903	Max. : 1.64504	Max. : 1.4628	Max. : 1.72437	Max. : 1.7537	Max. : 1.71705

Figura 1.3: Dataset summary

```
> var(dataset$y_prestazSWcalc)
[1] 134.4179
> var(dataset$x1_CPU)
[1] 1
> var(dataset$x2_HD)
[1] 1
> var(dataset$x3_proc)
[1] 1
> var(dataset$x4_aging)
[1] 1
> var(dataset$x5_audio)
[1] 1
> var(dataset$x6_RAM)
[1] 1
```

Notiamo che effettivamente i regressori X_i sono stati standardizzati e centrati con media campionaria nulla e varianza campionaria unitaria.
È possibile sintetizzare graficamente tali informazioni tramite dei boxplot, ottenibili mediante il comando `boxplot()`. In un boxplot:

- la linea nera rappresenta la mediana;
- la linea immediatamente inferiore e quella immediatamente superiore alla mediana rappresentano, rispettivamente, il 25° e 75° percentile, ovvero il primo e il terzo quartile;
- la lunghezza del box è data dalla differenza tra primo e terzo quartile e si chiama range interquartile (IQR);
- le linee che si estendono oltre il box sono dette baffi e hanno una lunghezza massima pari a $1.5 * IQR$ e potrebbero avere lunghezza inferiore se il massimo e/o il minimo non sono outliers;
- se ci sono ulteriori osservazioni non incluse nei baffi esse vengono rappresentate come punti isolati e sono detti outliers.

Di seguito vengono riportati i boxplot di tutte le variabili.

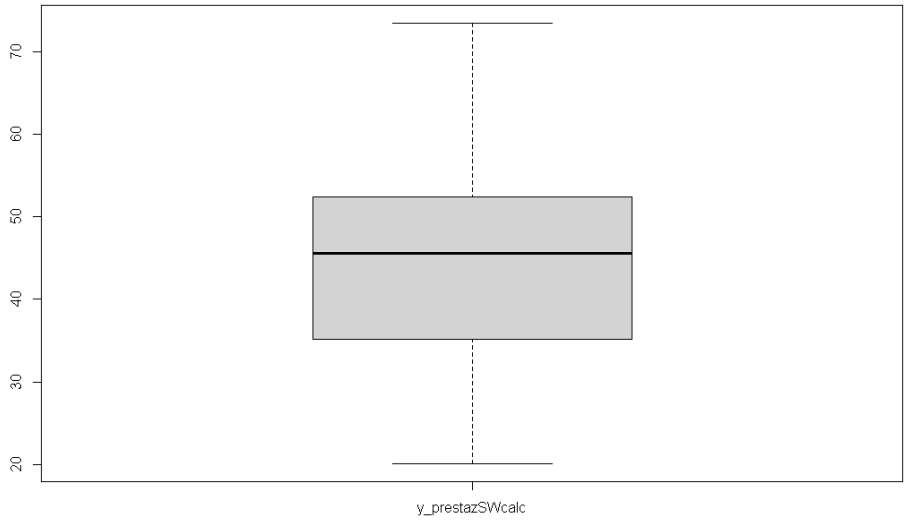

Figura 1.4: Boxplot variabile Y

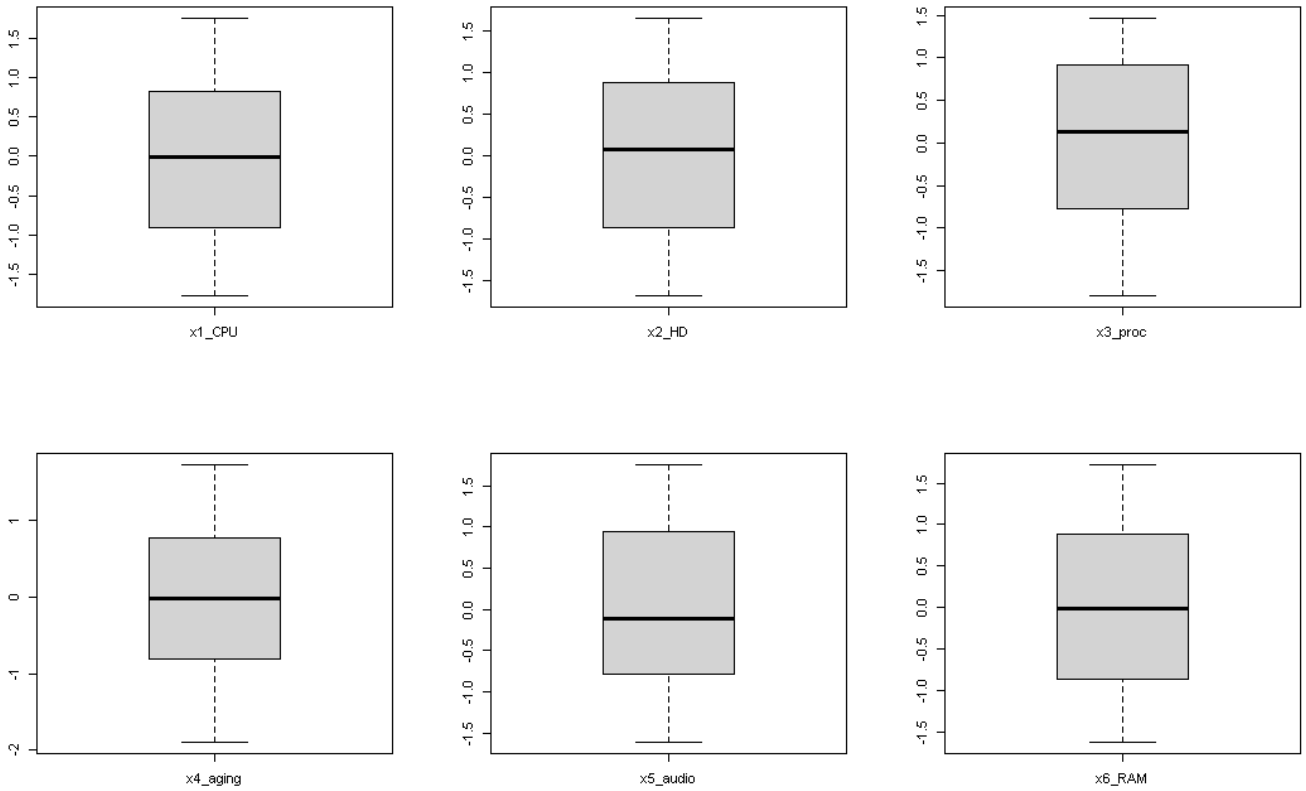

Figura 1.5: Boxplot variabili X_i

Tramite il comando `hist(dataset$nome_variabile,freq=F)` possiamo, inoltre, realizzare gli istogrammi della distribuzione di frequenza relativa delle 7 variabili del dataset:

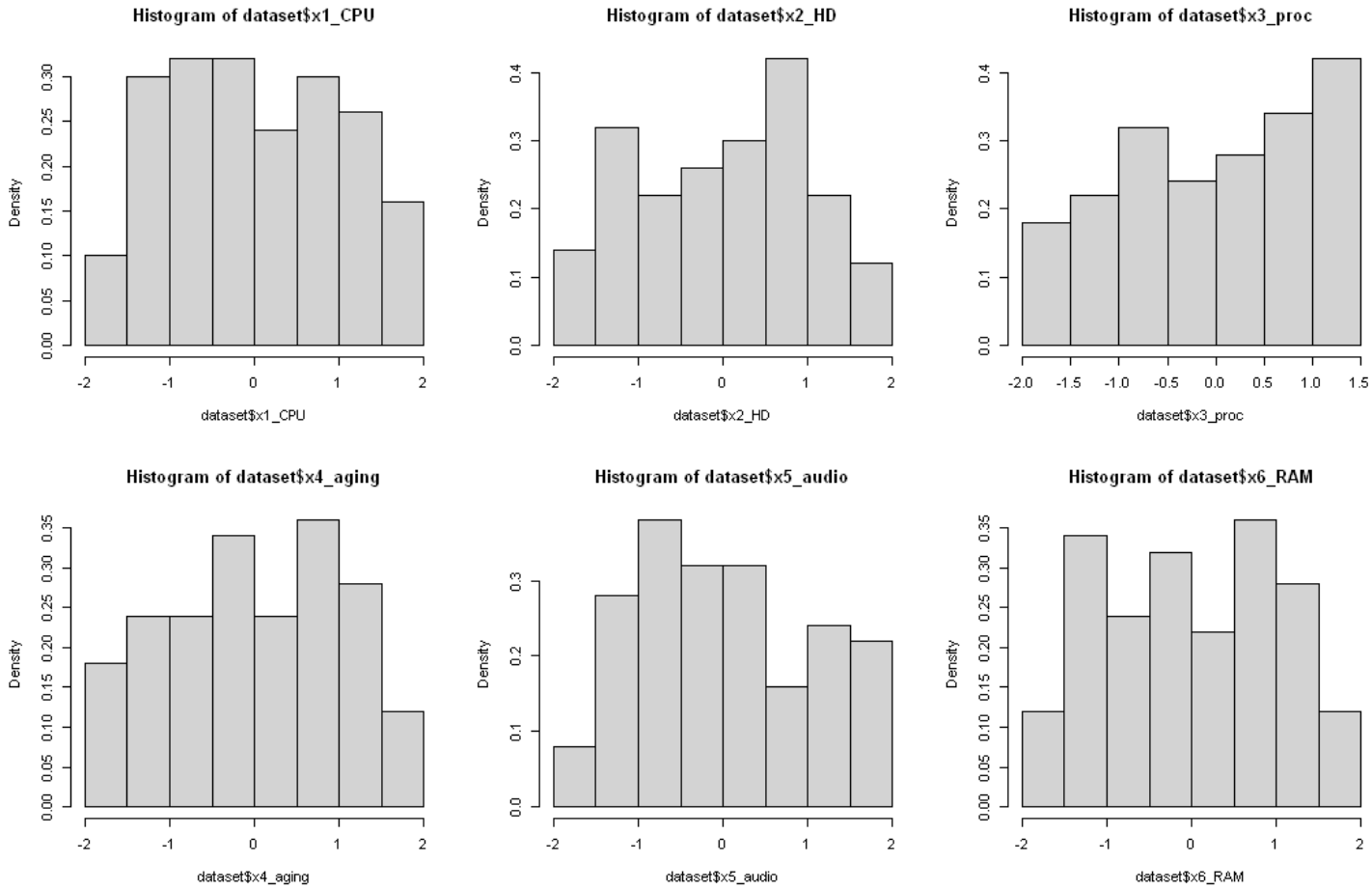

Figura 1.6: Istogrammi variabili X_i

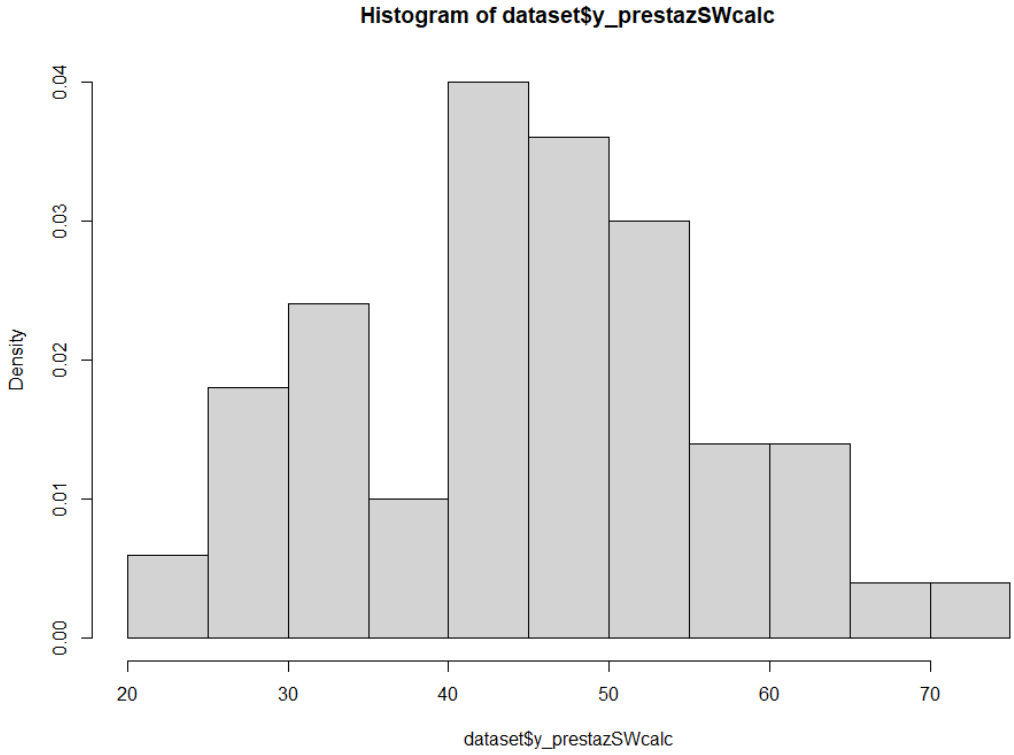

Figura 1.7: Istogramma variabile Y

Si osservi che R ha suddiviso il range delle variabili continue X_i in 8 classi e quello della variabile Y in 11 classi. Osservando questi istogrammi possiamo notare che essi non sono immediatamente riconducibili a distribuzioni note, complice anche la bassa numerosità campionaria.

Verifichiamo quindi se è possibile ricondurre le distribuzioni delle variabili a delle distribuzioni Normali tramite dei normal Q-Q plot, realizzabili con il comando `qqnorm(dataset$variabile)`. Tali grafici mettono a confronto i quantili delle osservazioni campionarie con quelli noti di una normale standard, per cui usualmente sarebbe necessario standardizzare i dati col comando `scale(dataset$variabile)` prima di effettuare il confronto. Ciononostante, dal momento che le variabili indipendenti X_i sono già standardizzate, sarà necessario effettuare tale operazione solo per Y. Una volta tracciato il grafico, se i punti del Q-Q plot si dispongono attorno alla bisettrice del primo e del terzo quadrante, è possibile concludere che i dati sono distribuiti secondo una Normale.

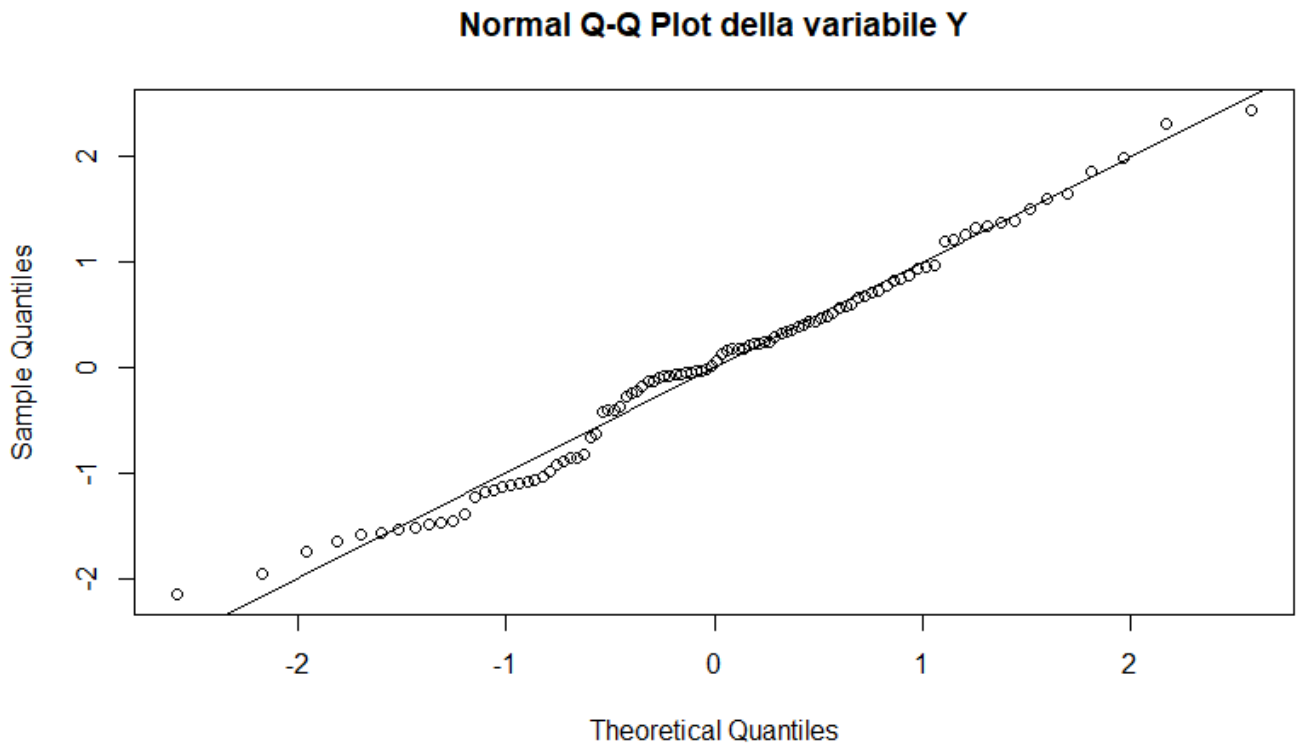

Figura 1.8: Q-Q plot della variabile Y

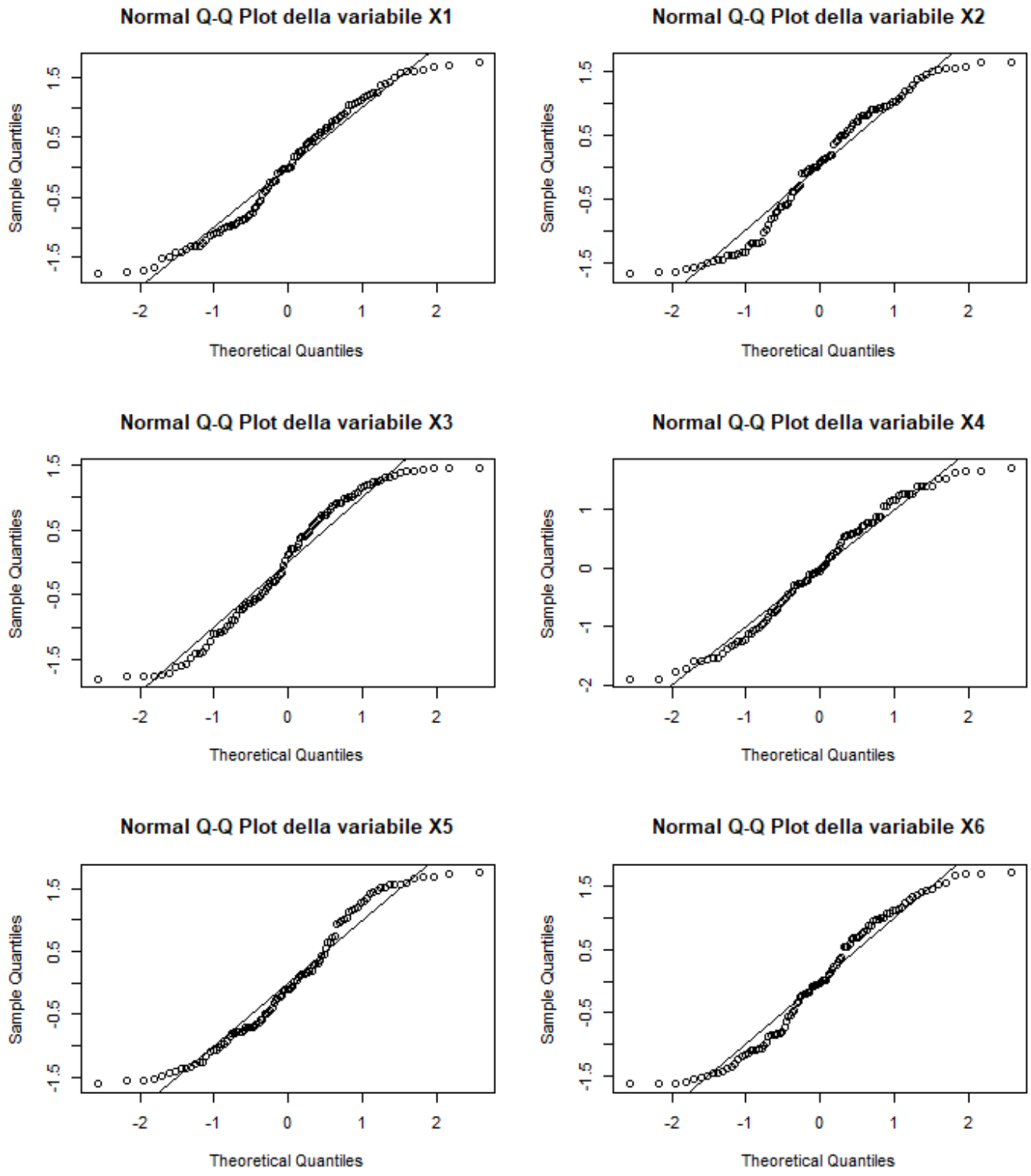

Figura 1.9: Q-Q plot delle variabili indipendenti X_i

A partire dai Q-Q plot si evince che tutte le variabili indipendenti si distribuiscono approssimativamente come una Normale standard con media nulla e varianza unitaria. Inoltre, possiamo osservare che anche la variabile dipendente Y si distribuisce approssimativamente come una Normale.

1.2 Analisi di correlazione e valutazione delle relazioni tra le variabili

Per controllare se fra due attributi vi è una relazione, è possibile tracciare uno scatter plot tramite il comando `scatterplot(y_prestazSWcalc ~ x1_CPU, data=dataset)` del package `car`. Tale grafico fornisce una serie di elementi che permettono di individuare con estrema facilità eventuali relazioni (non necessariamente lineari) fra le differenti variabili. In particolare la linea retta blu indica la migliore retta di regressione che si adatta ai dati, mentre la linea tratteggiata contraddistingue una curva che approssima l'andamento dei dati e tiene conto della loro variabilità.

Una misura quantitativa del grado di dipendenza lineare tra due variabili è inoltre fornita dal coefficiente di correlazione R:

$$R = \frac{\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \overline{X})^2 \sum_{i=1}^{N}(Y_i - \overline{Y})^2}}$$

con X, Y le variabili da analizzare e $\overline{X}$, $\overline{Y}$ le corrispettive medie campionarie. In R è possibile calcolare il coefficiente di correlazione tra due variabili del dataset mediante il comando `cor(dataset$x,dataset$y)`.

I coefficienti delle rette di regressione tra la variabile dipendente Y e le variabili indipendenti X_i saranno calcolati col metodo dei minimi quadrati attraverso il comando `coefficients(model)` sul modello di regressione lineare `lm(Y ~ Xi,data=dataset)`.
Si procede ora all'analisi di correlazione di tutte le variabili del dataset.

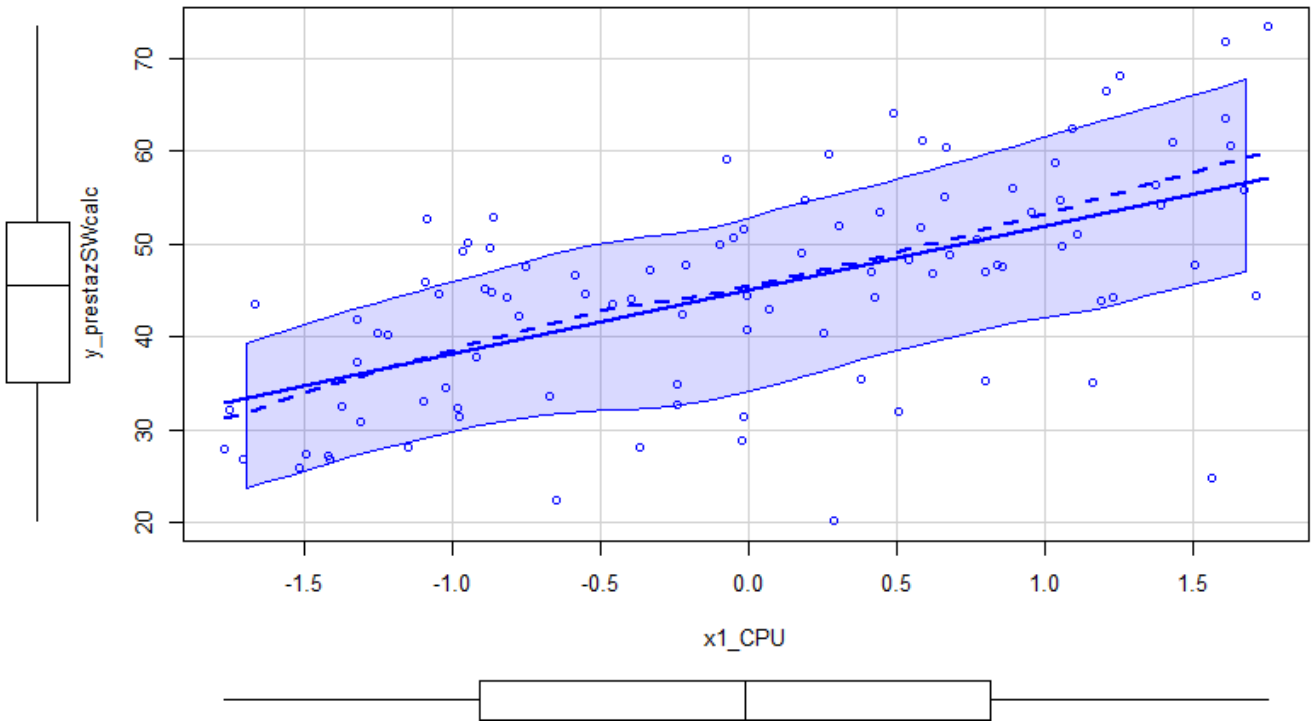

Figura 1.10: Scatter plot X_1/Y

Dal grafico si nota una correlazione abbastanza marcata fra la velocità della CPU X_1 e la variabile dipendente Y. Tale ipotesi trova conferma nel valore del coefficiente di correlazione, infatti quest'ultimo è pari a **0.5928266**.

Dai dati si può ricavare la retta di regressione (la linea blu nello scatter plot), che ha equazione:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1$$

Tramite i comandi:

```
1        ds=dataset
2        lin_fit1=lm(ds$y_prestazSWcalc~ds$x1_CPU)
3        coefficients(lin_fit1)
```

si calcolano i coefficienti di regressione, che risultano essere pari a

$$\beta_0 = 45.079155 \qquad \beta_1 = 6.873158$$

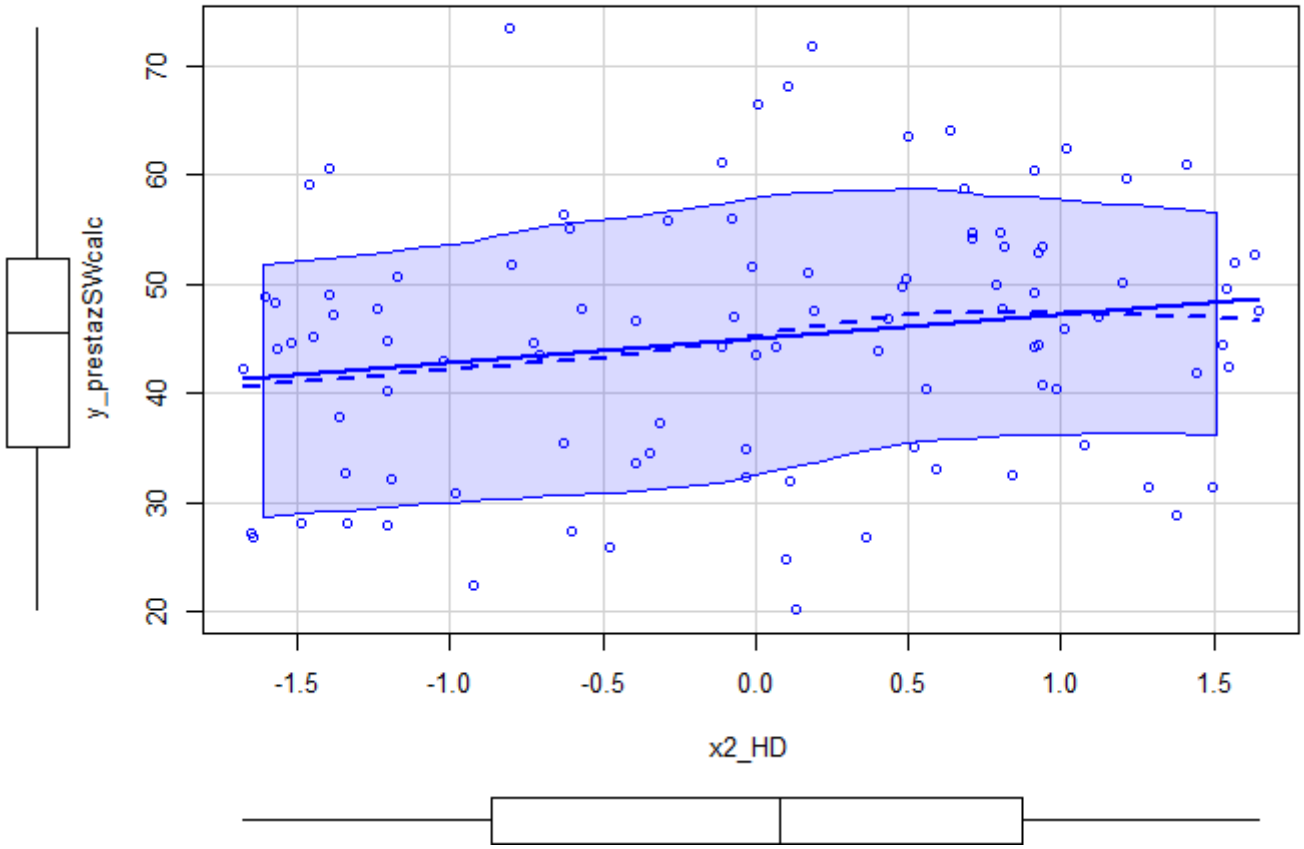

Figura 1.11: Scatter plot X_2/Y

La relazione fra la variabile dipendente Y e la dimensione dell'HDD X_2 sembra essere debole, come si può osservare dalla distribuzione dei punti nello scatter-plot, in effetti il coefficiente di correlazione fra le due variabili risulta essere molto basso essendo pari a **0.1929187**.

A partire dai dati si possono calcolare i coefficienti della retta di regressione $\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_2$ col metodo dei minimi quadrati:

$$\beta_0 = 45.079155 \qquad \beta_1 = 2.236676$$

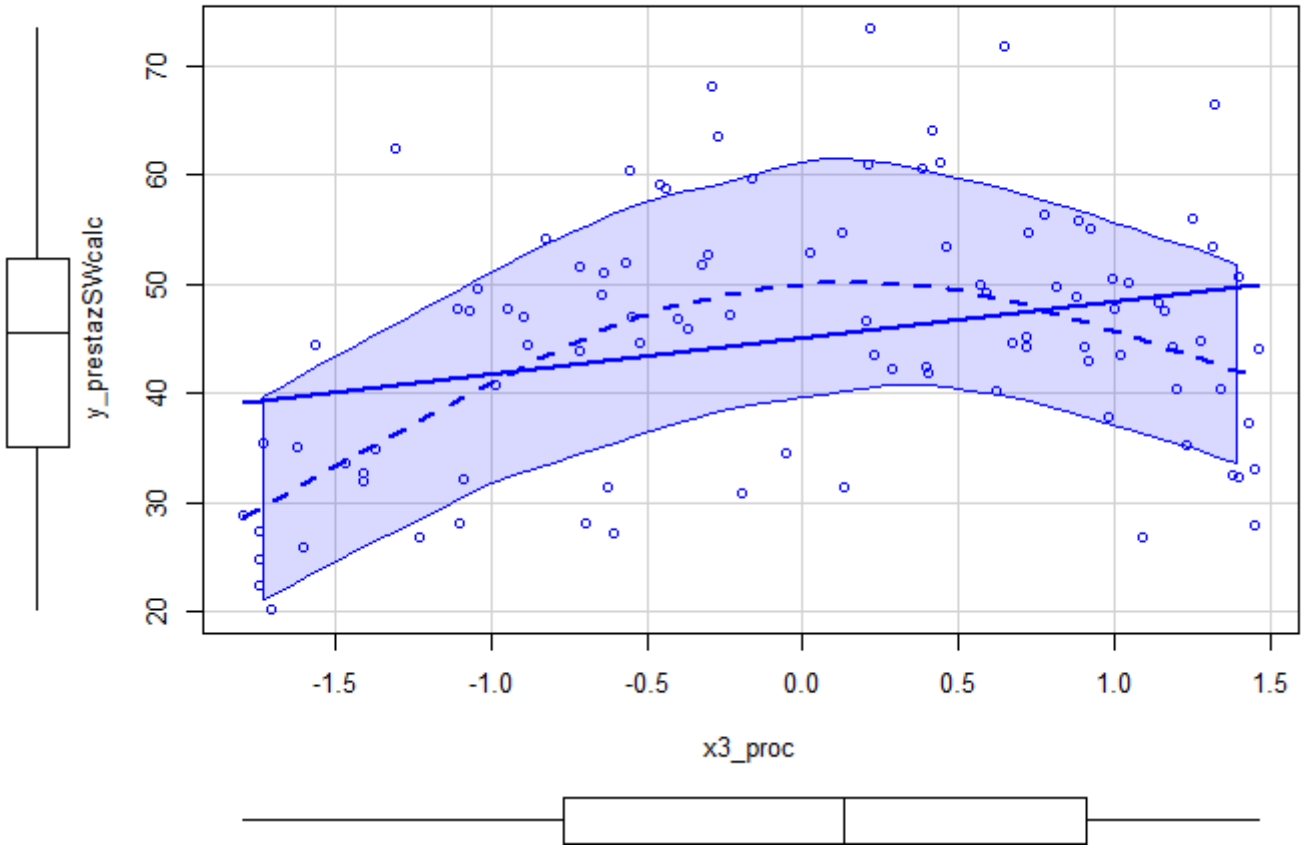

Figura 1.12: Scatter plot X_3/Y

Dallo scatterplot di Y e X_3 si nota come i punti tendono a distribuirsi come una sorta di parabola rovesciata. La retta di regressione non sembra spiegare bene l'andamento dei dati, per cui si sospetta che ci sia una bassa correlazione tra le variabili. Ciò trova conferma dal calcolo del coefficiente di correlazione che è uguale a **0.2857017**.
La retta di regressione $\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_3$ ha i coefficienti di regressione pari a:

$$\beta_0 = 45.079155 \qquad \beta_1 = 6.873158$$

Ma la disposizione dei punti nello scatter plot lascia pensare che una relazione più verosimile tra le due variabili possa essere del tipo:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_3^{\boldsymbol{2}}$$

con coefficienti:

$$\beta_0 = 52.210872 \qquad \beta_1 = -7.203755$$

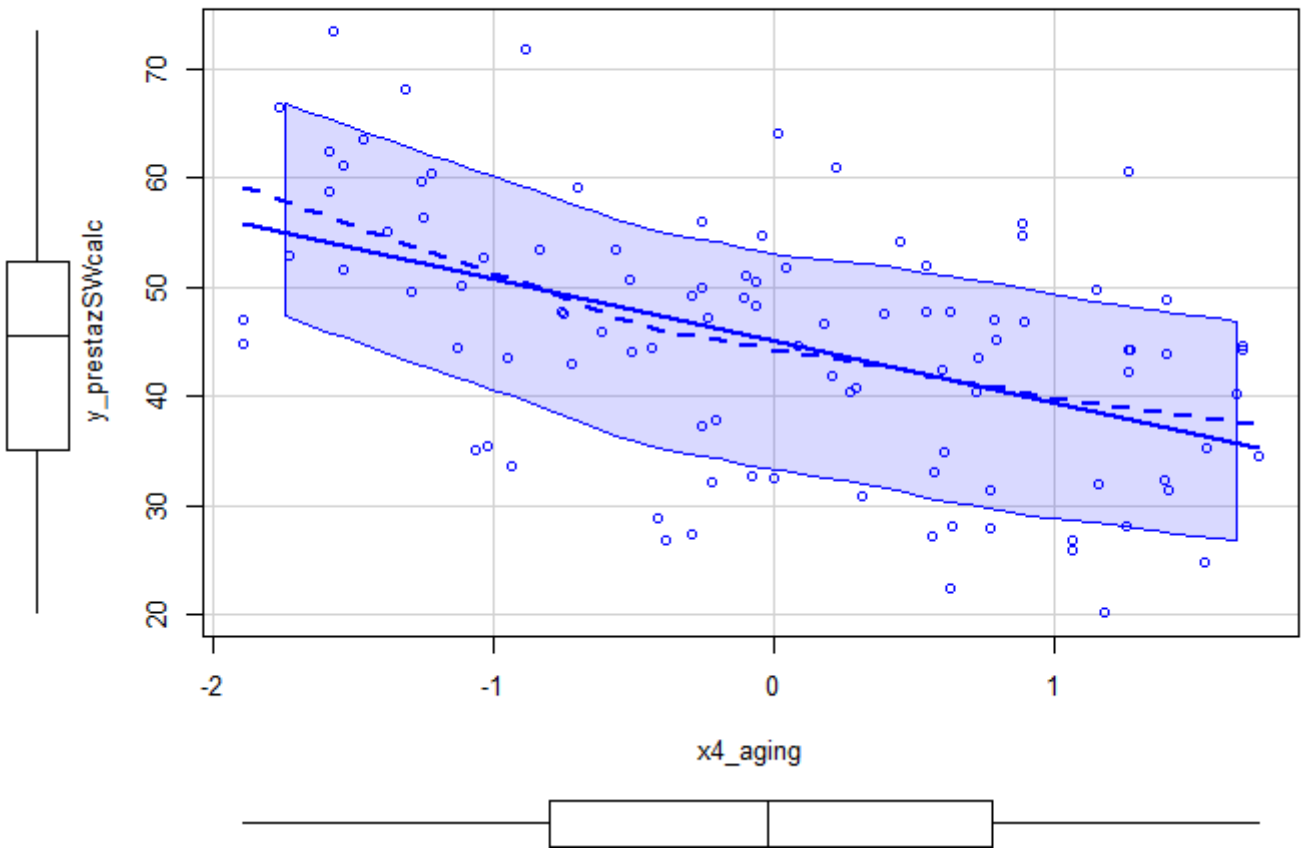

Figura 1.13: Scatter plot X_4/Y

L'aging del software X_4 manifesta una certo legame con le prestazioni del software di calcolo Y. Infatti, la variabile dipendente presenta una decrescita all'aumentare della variabile indipendente. Calcolando il coefficiente di correlazione, che risulta essere pari a **-0.493331**, si evidenza la presenza di una discreta correlazione negativa.

La retta di regressione $\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_4$ ha i seguenti coefficienti:

$$\beta_0 = 45.079155 \qquad \beta_1 = -5.719618$$

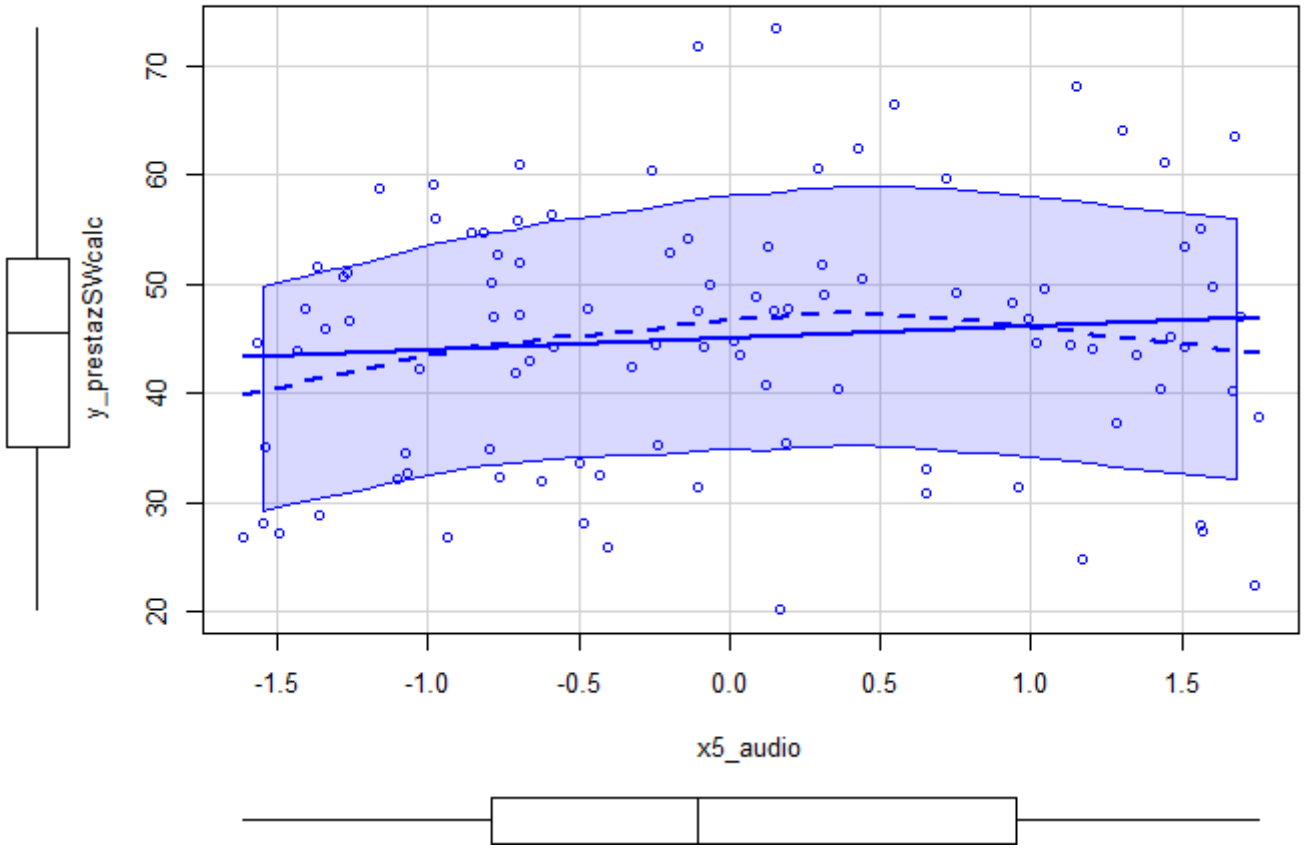

Figura 1.14: Scatter plot X_5/Y

Il modo con cui i punti sono distribuiti all'interno dello scatter plot tra Y e X_5 suggerisce una quasi totale assenza di correlazione, in quanto i punti non sembrano aderire ad uno specifico pattern. L'indice di correlazione conferma la precedente ipotesi fornendo un valore di **0.09336966**.

A questo punto si calcolano i coefficienti della retta di regressione $\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_5$ che risultano essere pari a:

$$\beta_0 = 45.079155 \qquad \beta_1 = 1.082516$$

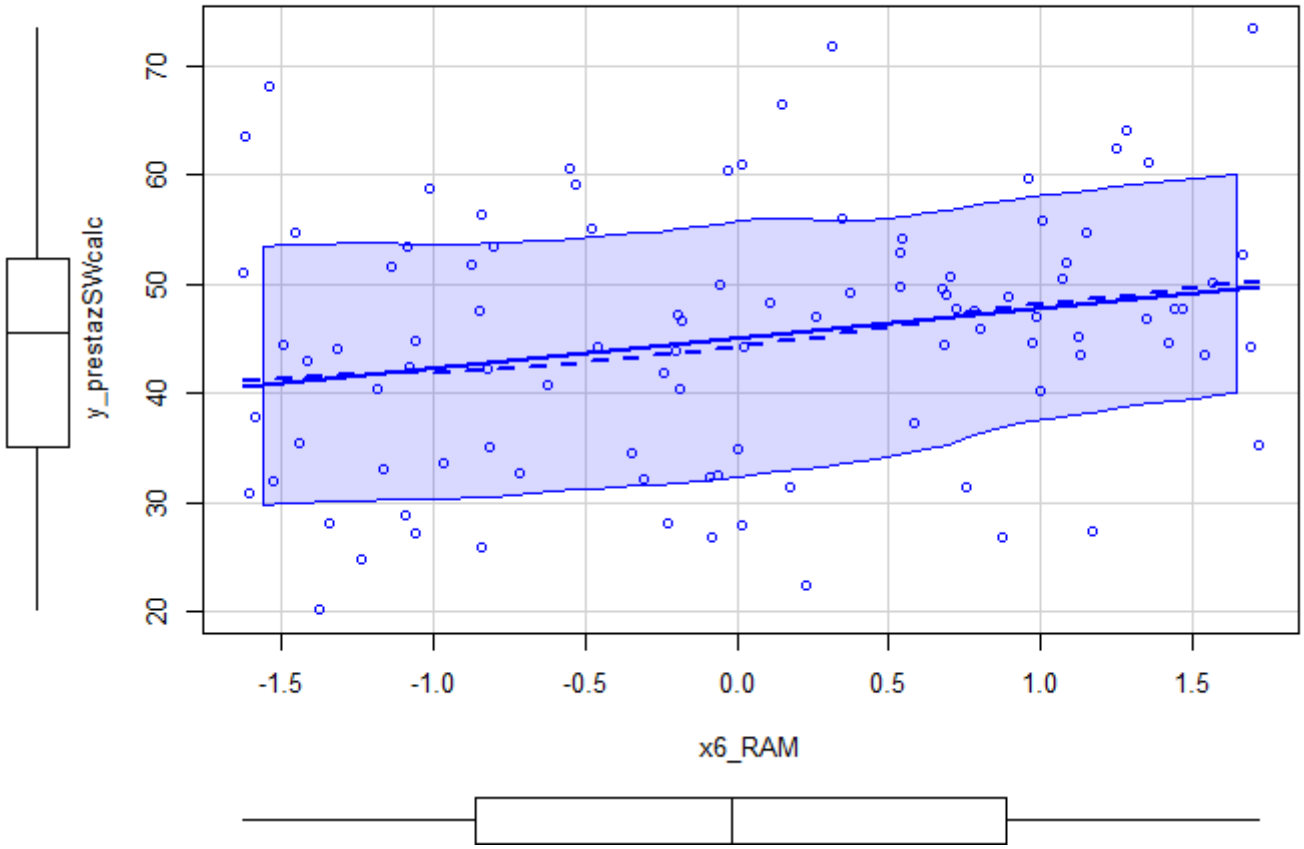

Figura 1.15: Scatter plot X_6/Y

Anche l'ultimo scatter plot non manifesta un'evidente relazione. Le prestazioni della RAM X_6 palesano un legame molto debole con la variabile dipendente Y, con un coefficiente di correlazione pari a **0.2382755**.

La retta di regressione $\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_6$ ha coefficienti:

$$\beta_0 = 45.079155 \qquad \beta_1 = 2.762536$$

Di seguito è riportato una tabella che riassume i rapporti di correlazione che sussistono fra tutte le possibili coppie di variabili del dataset, ottenuta tramite il comando `corrplot(cor(dataset))`.

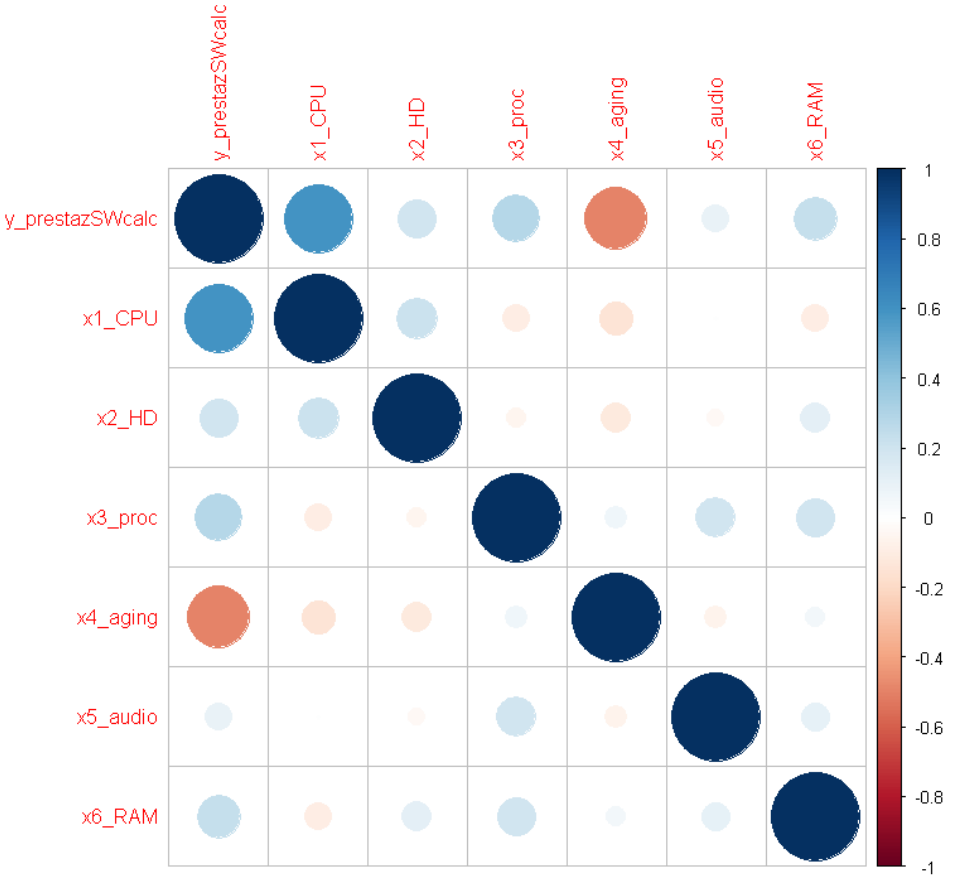

Figura 1.16: Plot correlazione tra tutte le variabili

Da questo grafico si evince l'assenza di multicollinearità, ovvero nessun regressore può essere espresso mediante una combinazione lineare degli altri.

A valle dell'analisi di correlazione, si riassumono i principali risultati ottenuti:

- i regressori X_1 e X_4 hanno una marcata correlazione lineare con la variabile dipendente Y e potrebbero essere significativi ai fini della regressione;
- il regressore X_3^2 potrebbe, anch'esso, essere rilevante ai fini della regressione;
- non c'è correlazione tra le variabili dipendenti X_i.

CAPITOLO 2

REGRESSIONE LINEARE MULTIPLA

Sulla base di quanto osservato durante l'analisi preliminare, si vuole ora trovare un modello di regressione lineare che descriva l'andamento della variabile dipendente Y in funzione dei regressori $X_1, ..., X_6$.

2.1 Richiami teorici

In generale un modello di regressione lineare multipla con p predittori si presenta nella forma:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_2 \boldsymbol{X}_2 + ... + \beta_p \boldsymbol{X}_p + \boldsymbol{\varepsilon} \tag{2.1}$$

con $\boldsymbol{\varepsilon}$ variabile aleatorea che descrive la variabilità della quantità osservata Y non spiegata dalla regressione.
Se per ciascuna variabile sono presenti n osservazioni si può scrivere:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \varepsilon_i \qquad \forall i = 1, ..., n \tag{2.2}$$

o, più sinteticamente, in forma matriciale:

$$\underline{y} = \underline{\underline{X}}\, \underline{\beta} + \underline{\boldsymbol{\varepsilon}} \tag{2.3}$$

con

$$\underline{y} = \begin{bmatrix} y_1 \\ ... \\ y_n \end{bmatrix} \qquad \underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ ... \\ \beta_p \end{bmatrix} \qquad \underline{\boldsymbol{\varepsilon}} = \begin{bmatrix} \boldsymbol{\varepsilon_1} \\ ... \\ \boldsymbol{\varepsilon_n} \end{bmatrix} \qquad \underline{\underline{X}} = \begin{bmatrix} 1 & x_{11} & x_{12} & ... & x_{1p} \\ 1 & x_{21} & x_{22} & ... & x_{2p} \\ ... & ... & ... & ... & ... \\ 1 & x_{n1} & x_{n2} & ... & x_{np} \end{bmatrix} \tag{2.4}$$

Si assumerà inoltre che $n > p + 1$ e che

$$\underline{\boldsymbol{\varepsilon}} \sim MVN(\underline{0}, \sigma^2 \, I_n) \tag{2.5}$$

ovvero che le $\boldsymbol{\varepsilon_i}$ siano gaussiane a media nulla, varianza costante (omoschedasticità) e indipendenti tra loro. La validità di tali ipotesi di lavoro dovrà poi essere verificata con opportuni metodi.
La devianza residua (o RSS) indica la variabilità non spiegata dal modello ed è definita come

$$RSS(\underline{\beta}) = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 = (\underline{y} - \underline{\underline{X}}\,\underline{\beta})^T(\underline{y} - \underline{\underline{X}}\,\underline{\beta}) \tag{2.6}$$

Per stimare i coefficienti β_i è possibile applicare il metodo dei minimi quadrati (OLS) che consiste nel trovare i coefficienti che minimizzano l'RSS. Per fare ciò è possibile porre

$$\nabla_{\underline{\beta}} RSS(\underline{\beta}) = \underline{0} \tag{2.7}$$

e verificare che il punto trovato sia un minimo accertandosi che l'Hessiana sia definita positiva. È possibile dimostrare che la soluzione della 2.7 è

$$\underline{\hat{\beta}} = (\underline{\underline{X}}^T \underline{\underline{X}})^{-1} \underline{\underline{X}}^T \underline{y} \tag{2.8}$$

che, date le ipotesi precedenti, rappresenta un vettore di stimatori lineari $\hat{\beta}_i$ per i coefficienti β_i, i quali sono non distorti e a minima varianza. Grazie all'ipotesi di normalità delle ε_i tali stime coincidono con quelle a massima verosimiglianza (ML). In particolare $\underline{\hat{\beta}}$ è distribuita come una multivariata gaussiana così caratterizzata:

$$\underline{\hat{\beta}} \sim MVN(\underline{\beta}, (\underline{\underline{X}}^T \underline{\underline{X}})^{-1}\sigma^2) \tag{2.9}$$

Uno stimatore non distorto per la varianza σ^2 è dato da

$$\hat{\sigma^2} = \frac{RSS}{n - (p + 1)} \tag{2.10}$$

La statistica test

$$(n - p - 1)\,\frac{\hat{\sigma^2}}{\sigma^2} \sim \chi^2_{n-p-1} \tag{2.11}$$

è distribuita come una chi-quadrato con $\nu = n - p - 1$ gradi di libertà ed è utile per realizzare stime intervallari e test d'ipotesi sul parametro σ^2. Analogamente la statistica test

$$t_i = \frac{\hat{\beta}_i - \beta_i}{SE(\hat{\beta}_i)} = \frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}\sqrt{v_i}} \sim T_{(n-p-1)} \tag{2.12}$$

è distribuita come una T di Student con $\nu = n - p - 1$ gradi di libertà ed è utile per realizzare stime intervallari e test d'ipotesi sui coefficienti β_i del modello. Si osservi che $SE(\hat{\beta}_i) = \hat{\sigma}\sqrt{v_i}$ rappresenta lo standard error e $v_i = [(\underline{\underline{X}}^T \underline{\underline{X}})^{-1}]_{ii}$ rappresenta l'elemento i-esimo sulla diagonale della matrice

$(\underline{\underline{X}}^T \underline{\underline{X}})^{-1}$.

Infine, un indice utile per quantificare la bontà di adattamento del modello è l'indice di determinazione multipla R^2

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} = 1 - \frac{RSS}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{2.13}$$

il quale è il rapporto tra la variabilità spiegata dal modello e quella totale. Per tale ragione $0 \leq R^2 \leq 1$ e il modello è tanto più adattato ai dati quanto più R^2 si avvicina a 1. Dal momento che all'aumentare del numero predittori X nel modello aumenta anche il valore di R^2, nei modelli di regressione lineare multipla è più conveniente utilizzare una versione corretta dell'indice R^2, che compensa questo fenomeno. Tale indice è detto coefficiente di determinazione multipla corretto ed è pari a

$$R^2_{adj} = 1 - \frac{n-1}{n-p-1}\frac{RSS}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{2.14}$$

Dopo aver trovato un modello di regressione, per ciascuna delle ipotesi fatte (specificazione del modello, vedi le 2.5) è necessario verificare la loro validità (test di specificazione), eventualmente apportando modifiche al modello qualora esse non siano verificate. La verifica delle ipotesi avviene analizzando i residui e_i, in particolare si verificheranno:

- **Linearità** tra la variabile dipendente e le variabili indipendenti: in particolare se il modello presenta il termine di intercetta β_0 si deve verificare se gli errori abbiano media statisticamente nulla. Si verificherà tale ipotesi graficamente tramite scatter plot dei residui e analiticamente mediante test t.

- **Omoschedasticità** dei residui, verificata graficamente tramite scatter plot e analiticamente tramite test di Breusch-Pagan.

- **Normalità** dei residui, verificabile graficamente tramite Q-Q plot dei residui (eventualmente standardizzati) e analiticamente tramite test di Shapiro-Wilk.

- **Indipendenza tra i residui** verificata tramite test di Durbin-Watson.

Inoltre l'analisi dei residui permette di effettuare la diagnostica del modello: i residui hanno diverse proprietà utili per confermare se il modello stimato è sostenibile in rapporto al campione osservato. In particolare verificheremo l'eventuale presenza di valori anomali (outliers).

Nel modello di regressione un valore anomalo si presenta quando un'osservazione della variabile dipendente Y è atipica rispetto alla relazione ipotizzata dal modello. Per individuare gli outliers occorre considerare i residui standardizzati, definiti dalla relazione

$$\tau_i = \frac{\hat{e_i}}{s\sqrt{1-h_{ii}}} \quad con \quad i = 1, 2, ..., n \tag{2.15}$$

con h_{ii} elemento (i,i)-esimo della matrice di predizione (hat matrix):

$$\underline{\underline{H}} = \underline{\underline{X}}(\underline{\underline{X}}^T\underline{\underline{X}})^{-1}\underline{\underline{X}}^T \tag{2.16}$$

Si considereranno outliers tutti i residui standardizzati per cui

$$|\tau_i| > 3 \tag{2.17}$$

Si noti che un numero eccessivo di outlier è indice di un modello poco accurato.

2.2 Modello lineare

I fattori che indicano la bontà di un modello quali il coefficiente di determinazione R^2, l'AIC o il BIC assumono significato nel confronto tra più modelli. Per questa ragione si inizia definendo un primo modello di riferimento lineare rispetto ai 6 regressori, del tipo

$$\boldsymbol{Y} = \beta_0 + \beta_1\boldsymbol{X}_1 + \beta_2\boldsymbol{X}_2 + \beta_3\boldsymbol{X}_3 + \beta_4\boldsymbol{X}_4 + \beta_5\boldsymbol{X}_5 + \beta_6\boldsymbol{X}_6 + \boldsymbol{\varepsilon} \tag{2.18}$$

si procede ora a stimare i parametri del modello.

2.2.1 Stima ai minimi quadrati dei parametri

Dopo aver importato il dataset, si definiscono la matrice di design `X` e il vettore colonna `y` secondo le 2.4 e quindi si effettuano le stime ai minimi quadrati dei coefficienti codificando la 2.8. Si utilizzano le funzioni `t()` per trasporre una matrice e `inv()` (contenuta nel package `matlib`) per invertire una matrice.

```
>    X=cbind(1,dataset$x1_CPU,dataset$x2_HD,dataset$x3_proc,
     dataset$x4_aging,dataset$x5_audio,dataset$x6_RAM)
>    y=as.matrix(dataset$y_prestazSWcalc)
>    library(matlib)
>    beta_hat=inv(t(X) %*% X) %*% t(X) %*% y
>    print(beta_hat)

             [,1]
[1,] 45.0791548
[2,]  6.7669666
[3,]  0.0528328
[4,]  3.8069612
[5,] -5.1648902
[6,] -0.3229424
[7,]  2.9464540
```

Listing 2.1: Stima ai minimi quadrati in R con design matrix

Per semplificare la stima dei coefficienti è anche possibile utilizzare il comando `lm(formula, data)`, nel quale "`formula`" rappresenta la descrizione simbolica del modello da stimare e "`data`" indica il nome del dataframe nel quale sono presenti le variabili del modello. Successivamente si usa il comando `coef(object)` per estrarne i coefficienti.

```
>   lin_fit=lm(y_prestazSWcalc ~ x1_CPU + x2_HD + x3_proc + x4_aging + x5_audio + x6_RAM,
    data=dataset)
>    coef(lin_fit)

(Intercept)      x1_CPU       x2_HD     x3_proc    x4_aging    x5_audio       x6_RAM
45.07915476  6.76696494  0.05282932  3.80696222 -5.16488753 -0.32294590  2.94645161
```

Listing 2.2: Stima ai minimi quadrati in R con lm() e output

2.2.2 Coefficiente di determinazione

Dopo aver trovato il modello e calcolato i parametri β_i, possiamo valutare l'indice di determinazione multipla R^2, facendo riferimento alla formula 2.13.

```
>   RSS=(t(y-X%*%beta_hat)) %*% (y-X%*%beta_hat) #RSS di beta_hat
>   y_m=mean(y)
>   TSS=(sum((y-y_m)^2))
>   R_quadro= 1-(RSS/TSS) #indice di determinazione multipla R quadro
>   R_quadro

          [,1]
[1,] 0.7184309
```

Inoltre dato che si tratta di regressione lineare multipla conviene usare l'indice di determinazione multipla corretto R^2_{adj}, facendo riferimento alla formula 2.14.

```
>   n=nrow(dataset)   #num campioni
>   p=6               #num regressori
>   R_quadro_adj=1-((n-1)/(n-p-1))*(RSS/TSS)
>   R_quadro_adj
          [,1]
[1,] 0.7002651
```

Tuttavia, esiste un modo più semplice per trovare gli indici di determinazione multipla R^2 e R^2_{adj}:

```
>   model_summary=summary(lin_fit)
>   print(model_summary$r.squared)

[1] 0.7184309
>   print(model_summary$adj.r.squared)

[1] 0.7002651
```

2.2.3 Analisi dei residui

Dopo aver stimato il modello di regressione, bisogna verificare la validità delle ipotesi di base esposte in precedenza tramite opportuni test statistici. In primo luogo, si verifica la linearità tra regressori e variabile di risposta, ovvero che la media degli errori non sia significativamente diversa da zero. A tale scopo eseguiamo il test t di Student tramite il comando `t.test(residui)`, il quale ha come ipotesi:

$$H_0 : \mu = 0 \qquad H_1 : \mu \neq 0$$

```
> residui=residuals(lin_fit)
> t.test(residui) #test t per verificare linearit  tra regressori e risposta

  One Sample t-test

data:  residui
t = 2.2558e-16, df = 99, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -1.220704  1.220704
sample estimates:
   mean of x
1.387779e-16
```

Fissato il rischio di I specie $\alpha = 0.05$ si accetta l'ipotesi nulla poiché p-value $= 1 > \alpha = 0.05$. Inoltre, tramite il comando `plot(lin_fit)` si ottiene il seguente scatter plot che rappresenta i residui in funzione delle stime $\hat{y}_i$:

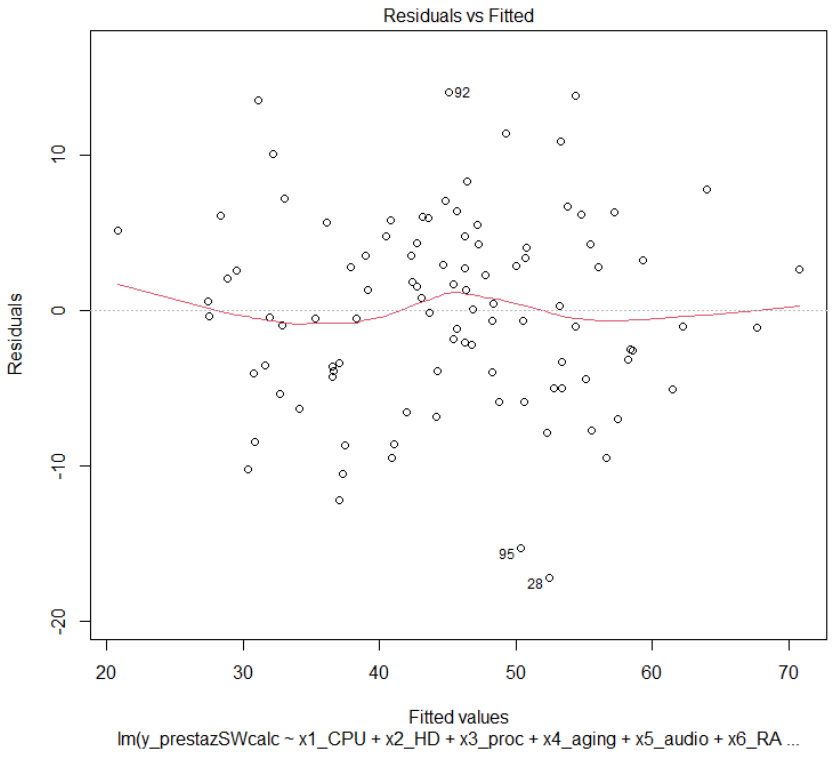

Figura 2.1: Scatterplot residui-stime

È possibile osservare anche graficamente come la media dei residui sia statisticamente nulla.

Successivamente si verifica la normalità della distribuzione degli errori con il test di Shapiro-Wilk, il quale ha come ipotesi:

$$H_0 : i\ residui\ sono\ distribuiti\ secondo\ una\ normale \qquad H_1 : H_0\ è\ falsa$$

```
1 >   shapiro=shapiro.test(residui)
2 >   print(shapiro)
3
4             Shapiro-Wilk normality test
5     data: residui
6     W = 0.9839, p-value = 0.2724
```

Fissato il rischio di I specie $\alpha = 0.05$, si può affermare che i campioni sono distribuiti come una variabile aleatorea Normale: dal momento che p-value$= 0.2724 > \alpha = 0.05$ si accetta l'ipotesi nulla H_0. Graficamente si può usare il Q-Q plot con il comando `qqnorm(scale(residui))` applicato ai residui standardizzati per verificare tale ipotesi.

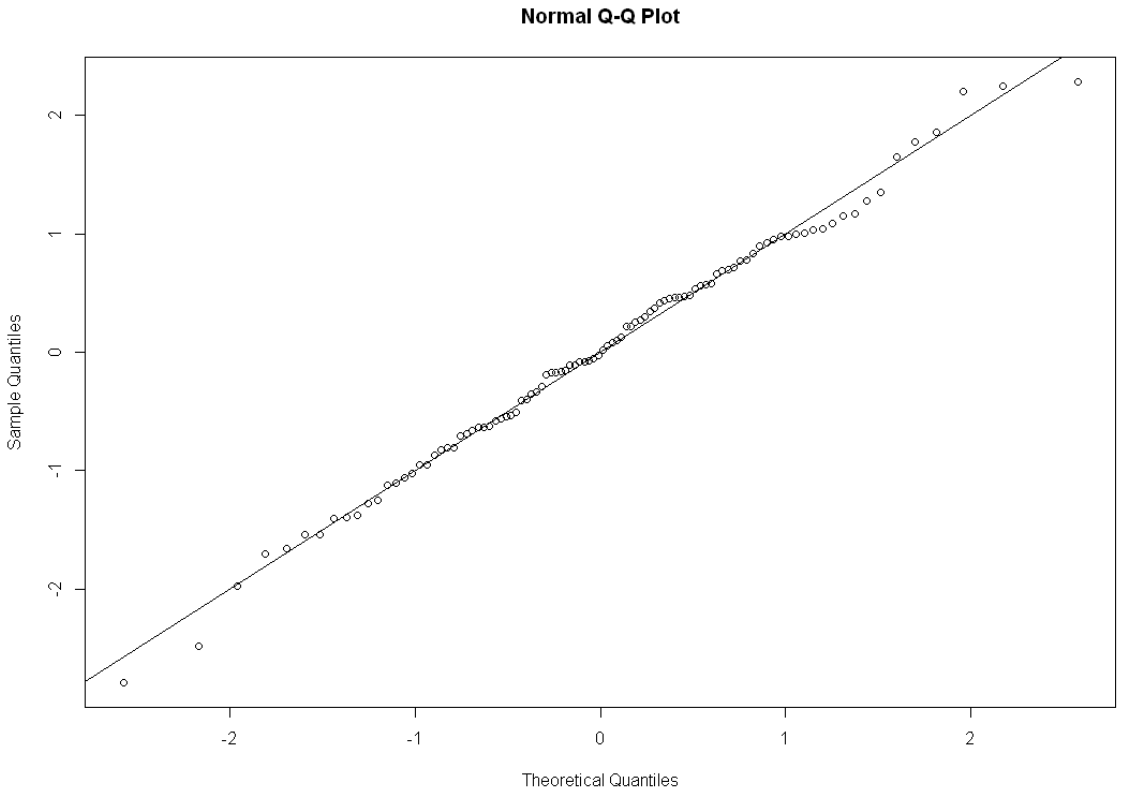

Figura 2.2: Normal Q-Q Plot

Bisogna poi verificare l'omoschedasticità dei residui utilizzando il test di Breusch-Pagan, tramite il comando `bptest()` della libreria `lmtest`:

```
> library(lmtest)
> modello=formula(lin_fit) ## memorizziamo la formula del modello in un oggetto
                           ## per facilita' di manipolazione
> testbp=bptest(modello,data=dataset) ## test di Breusch-Pagan
> testbp

        studentized Breusch-Pagan test
data: modello
BP = 4.9525, df = 6, p-value = 0.5499
```

Date le ipotesi

$$H_0 : si\ ha\ omoschedasticità \qquad H_1 : H_0\ è\ falsa$$

Fissato il rischio di I specie $\alpha = 0.05$, si osserva che è verificata l'ipotesi di omoschedasticità, in quanto p-value$= 0.5499 > \alpha = 0.05$. Graficamente tale ipotesi si può verificare osservando che la variabilità dei residui nel relativo scatter plot non varia significativamente in funzione dell'ascissa:

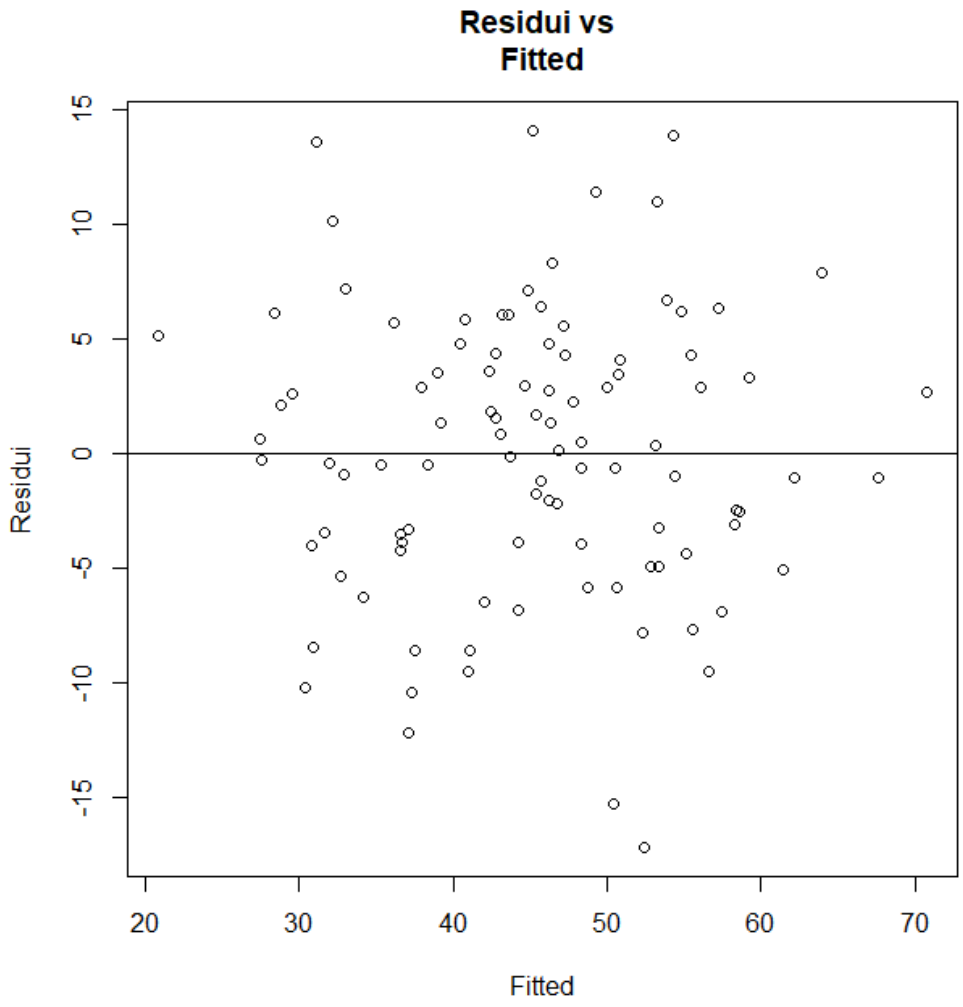

Figura 2.3: Scatter plot residui-stime

Si verifica infine l'assenza di autocorrelazione (che nel nostro caso equivale all'indipendenza) dei residui tramite il test di Durbin-Watson, le cui ipotesi sono

$$H_0 : L'autocorrelazione\ dei\ residui\ è\ 0 \qquad H_1 : L'autocorrelazione\ dei\ residui\ è\ maggiore\ di\ 0$$

tramite il comando `dwtest()`:

```
>   dw=dwtest(modello,data=dataset) ## test di Durbin-Watson
>   print(dw)

        Durbin-Watson test
    data: modello
    DW = 2.063, p-value = 0.6204
    alternative hypothesis: true autocorrelation is greater than 0
```

Fissato il rischio di I specie $\alpha = 0.05$, si osserva che è verificata l'assenza di autocorrelazione, perchè p-value$= 0.6204 > \alpha = 0.05$. Tutti i test di specificazione del modello hanno dato esito positivo, si può affermare che le ipotesi 2.5 alla base del modello di regressione sono valide.

A questo punto si effettua la diagnostica del modello verificando la presenza di eventuali outliers secondo la 2.17. In particolare con la funzione `rstandard(residui)` è possibile calcolare i residui standardizzati e quindi realizzarne lo scatter plot in funzione delle stime $\hat{y}_i$:

```
> standard_res=rstandard(lin_fit) #calcola i residui standardizzati
> y_hat = predict(lin_fit,dataset) #stime di y
> plot(y_hat,standard_res, ylab="Residui standardizzati", xlab="Stime di yi",
  main="Possibili Outliers",ylim = c(-5,5))
> abline(3,0,col="blue")
> abline(-3,0,col="blue")
```

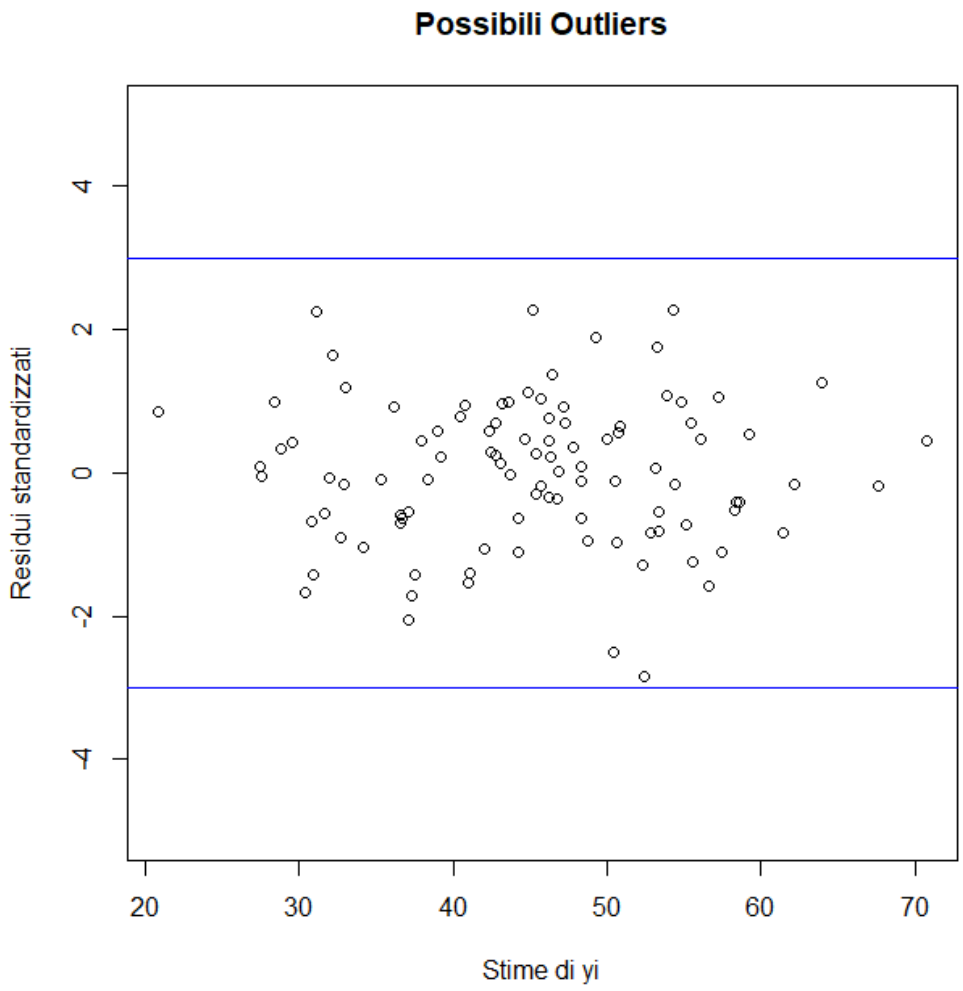

Figura 2.4: Possibili outliers

È possibile constatare l'assenza valori anomali.

2.2.4 Intervalli di confidenza per i coefficienti di regressione

Si vuole ora trovare le stime intervallari per i coefficienti β_i con un livello di confidenza $1-\alpha = 0.95$. A tale scopo è possibile utilizzare la statistica 2.12, distribuita nel caso in esame come un T di Student con $\nu = 100 - 1 = 99$ gradi di libertà. È possibile scrivere

$$Pr\left[t_1 < \frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}\sqrt{v_i}} \leq t_2\right] = 1 - \alpha$$

Volendo trovare un intervallo simmetrico rispetto allo 0 e tenendo conto della simmetria della distribuzione si ottiene

$$Pr\left[-t_{1-\frac{\alpha}{2};\nu} < \frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}\sqrt{v_i}} \leq t_{1-\frac{\alpha}{2};\nu}\right] = 1 - \alpha$$

$$\Leftrightarrow \qquad Pr\left[\hat{\beta}_i - \hat{\sigma}\sqrt{v_i}\, t_{1-\frac{\alpha}{2};\nu} \leq \beta_i < \hat{\beta}_i + \hat{\sigma}\sqrt{v_i}\, t_{1-\frac{\alpha}{2};\nu}\right] = 1 - \alpha$$

con $t_{1-\frac{\alpha}{2};\nu}$ il quantile $1-\frac{\alpha}{2}$ di una T di Student con ν gradi di libertà. In questo caso, scegliendo un livello di confidenza $1-\alpha=0.95$ è possibile calcolare gli intervalli di confidenza per i coefficienti $\beta_0,...,\beta_6$:

```
> n=nrow(dataset) #numero di campioni
> p=6
> alpha=0.05 #livello di rischio
> t_score=qt(p=alpha/2,df=(n-1),lower.tail=F) #quantile
> bounds=matrix(ncol=2,nrow=p+1) #matrice che conterr  i limiti
> colnames(bounds)=c("lower","upper")
> rownames(bounds)=c("beta_0","beta_1","beta_2","beta_3","beta_4","beta_5","beta_6")
> v=c()
> sigma_2_hat=RSS/(n-p-1) #stima della varianza
> sigma_hat=sqrt(sigma_2_hat) #stima della deviazione standard
> inv_Xt_X=inv(t(X)%*%X) #(X^T*X)^-1
> for(i in 1:(p+1)){
    v[i]=inv_Xt_X[i,i]
    bounds[i,1]=beta_hat[i]-sigma_hat*sqrt(v[i])*t_score
    bounds[i,2]=beta_hat[i]+sigma_hat*sqrt(v[i])*t_score
  }
> print(bounds)

           lower      upper
beta_0 43.819689 46.3386206
beta_1  5.450300  8.0836332
beta_2 -1.264810  1.3704761
beta_3  2.485600  5.1283228
beta_4 -6.456497 -3.8732830
beta_5 -1.623826  0.9779413
beta_6  1.631814  4.2610938
```

Gli intervalli di confidenza trovati, con un livello di confidenza $1-\alpha=0.95$, sono:

$$43.819689 \leq \beta_0 < 46.3386206$$
$$5.450300 \leq \beta_1 < 8.0836332$$
$$-1.264810 \leq \beta_2 < 1.3704761$$
$$2.485600 \leq \beta_3 < 5.1283228$$
$$-6.456497 \leq \beta_4 < -3.8732830$$
$$-1.623826 \leq \beta_5 < 0.9779413$$
$$1.631814 \leq \beta_6 < 4.2610938$$

Un modo più semplice per calcolare gli intervalli di confidenza per un dato modello di regressione è il comando `confint(model,level)` che prende come argomenti il modello lineare e il livello di confidenza:

```
> confint(lin_fit,level=0.95)
                 2.5 %     97.5 %
(Intercept) 43.818683 46.3396266
x1_CPU       5.449247  8.0846831
x2_HD       -1.265866  1.3715249
```

```
x3_proc       2.484545   5.1293794
x4_aging     -6.457527  -3.8722485
x5_audio     -1.624869   0.9789771
x6_RAM        1.630762   4.2621417
> confint(lin_fit,level=0.99)
                 0.5 %     99.5 %
(Intercept) 43.409953  46.748356
x1_CPU       5.021954   8.511976
x2_HD       -1.693476   1.799135
x3_proc      2.055729   5.558196
x4_aging    -6.876687  -3.453088
x5_audio    -2.047040   1.401148
x6_RAM       1.204126   4.688777
```

2.2.5 Test di ipotesi

Si verifica ora che i coefficienti di regressione siano significativamente diversi da zero, ovvero che esista una relazione significativa tra la variabile dipendente e il regressore. Questo è ottenuto tramite un test d'ipotesi

$$H_0 : \beta_i = 0 \qquad H_1 : \beta_i \neq 0$$

che è un particolare tipo di t-test, il quale più in generale è un test d'ipotesi con lo scopo di verificare se il valore medio di una distribuzione a varianza sconosciuta si discosta significativamente da un certo valore di riferimento.

Se H_0 è vera, per la 2.12 la statistica test

$$t_i = \frac{\hat{\beta}_i}{\hat{\sigma}\sqrt{v_i}} \sim T_{(n-p-1)}$$

è distribuita come una T di Student con $v = n - p - 1$ gradi di libertà. Fissato un rischio di I specie α, sia $I_{AC,i}$ la regione di accettazione dell'ipotesi $H_0 : \beta_i = 0$, si può scrivere:

$$Pr\left[\hat{\beta}_i \in I_{AC,i} \middle| H_0 \; vera\right] = 1 - \alpha$$

$$\Leftrightarrow Pr\left[-t_{1-\frac{\alpha}{2};\nu} < \frac{\hat{\beta}_i}{\hat{\sigma}\sqrt{v_i}} \leq t_{1-\frac{\alpha}{2};\nu}\right] = 1 - \alpha$$

$$\Leftrightarrow Pr\left[-\hat{\sigma}\sqrt{v_i}\, t_{1-\frac{\alpha}{2};\nu} < \hat{\beta}_i \leq \hat{\sigma}\sqrt{v_i}\, t_{1-\frac{\alpha}{2};\nu}\right] = 1 - \alpha$$

Per cui

$$I_{AC,i} = \left[-\hat{\sigma}\sqrt{v_i}\, t_{1-\frac{\alpha}{2};\nu}\,,\; \hat{\sigma}\sqrt{v_i}\, t_{1-\frac{\alpha}{2};\nu}\right]$$

Fissato un rischio di I specie $\alpha = 0.05$ e codificando il test di ipotesi in R si ottiene tale risultato:

```
> n=nrow(dataset) #numero di campioni
> p=6
> alpha=0.05 #rischio di prima specie
> t_score=qt(p=alpha/2,df=(n-1),lower.tail=F) #quantile
```

```
> bounds=matrix(ncol=2,nrow=p+1) #matrice che conterr  i limiti di I_AC
> v=c()
> sigma_2_hat=RSS/(n-p-1) #stima della varianza
> sigma_hat=sqrt(sigma_2_hat) #stima della deviazione standard
> inv_Xt_X=inv(t(X)%*%X) #(X^T*X)^-1
> for(i in 1:(p+1)){
    v[i]=inv_Xt_X[i,i]
    bounds[i,1]=-sigma_hat*sqrt(v[i])*t_score
    bounds[i,2]=sigma_hat*sqrt(v[i])*t_score
  }
> for(i in 1:(p+1)){
    if(beta_hat[i]<bounds[i,1] || beta_hat[i]>bounds[i,2]){
      print(paste("beta_", i-1, ": H_0 rifiutata",sep=""))
    } else {
      print(paste("beta_", i-1, ": H_0 accettata",sep=""))
    }
  }

[1] "beta_0: H_0 rifiutata"
[1] "beta_1: H_0 rifiutata"
[1] "beta_2: H_0 accettata"
[1] "beta_3: H_0 rifiutata"
[1] "beta_4: H_0 rifiutata"
[1] "beta_5: H_0 accettata"
[1] "beta_6: H_0 rifiutata"
```

ovvero i regressori X_2 e X_5 appaiono statisticamente non significativi.

Lo stesso risultato può essere ottenuto in maniera più semplice col comando `summary(modello)`, il quale mostra molte informazioni utili sul modello, tra cui:

- Minimo, massimo e quartili dei residui
- Stima dei coefficienti di regressione β e del loro standard error
- Valori delle t-statistic e dei p-value relativi al t-test sui coefficienti β
- Valori del coefficiente di determinazione semplice R^2 e corretto R^2_{adj}

```
> summary(lin_fit)

Call:
lm(formula = y_prestazSWcalc ~ x1_CPU + x2_HD + x3_proc + x4_aging +
    x5_audio + x6_RAM, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-17.1867  -3.9581  -0.0196   4.2739  14.0750

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.07915    0.63474  71.020  < 2e-16 ***
x1_CPU       6.76696    0.66357  10.198  < 2e-16 ***
x2_HD        0.05283    0.66406   0.080    0.937
```

```
x3_proc      3.80696    0.66594   5.717 1.30e-07 ***
x4_aging    -5.16489    0.65094  -7.934 4.64e-12 ***
x5_audio    -0.32295    0.65562  -0.493    0.623
x6_RAM       2.94645    0.66255   4.447 2.41e-05 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 6.347 on 93 degrees of freedom
Multiple R-squared:  0.7184,  Adjusted R-squared:  0.7003
F-statistic: 39.55 on 6 and 93 DF,  p-value: < 2.2e-16
```

In questo caso si giunge alla stessa conclusione osservando che per X_2 e X_5 risulta $p\text{-}value > \alpha = 0.05$, per cui si accetta l'ipotesi nulla (ovvero i regressori non sono significativi dal punto di vista statistico). Può quindi valere la pena considerare il modello semplificato

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_3 \boldsymbol{X}_3 + \beta_4 \boldsymbol{X}_4 + \beta_6 \boldsymbol{X}_6 + \boldsymbol{\varepsilon} \tag{2.19}$$

privo dei regressori X_2 e X_5.

```
> lin_fit_2=lm(y_prestazSWcalc ~ x1_CPU + x3_proc + x4_aging + x6_RAM, data=dataset)
> summary(lin_fit_2)

Call:
lm(formula = y_prestazSWcalc ~ x1_CPU + x3_proc + x4_aging +
    x6_RAM, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-16.9821  -3.9057  -0.1118   4.3037  14.2906

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  45.0792     0.6289   71.68  < 2e-16 ***
x1_CPU        6.7727     0.6430   10.53  < 2e-16 ***
x3_proc       3.7426     0.6475    5.78 9.44e-08 ***
x4_aging     -5.1446     0.6399   -8.04 2.49e-12 ***
x6_RAM        2.9302     0.6468    4.53 1.71e-05 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 6.289 on 95 degrees of freedom
Multiple R-squared:  0.7177,  Adjusted R-squared:  0.7058
F-statistic: 60.37 on 4 and 95 DF,  p-value: < 2.2e-16
```

Analizzando tale modello in R è possibile osservare che adesso tutti i regressori sono statisticamente significativi e il coefficiente di determinazione corretto risulta addirittura superiore, nonostante il modello sia più semplice. Tale modello è pertanto preferibile rispetto al precedente. Dai test eseguiti su R, inoltre, risulta che il modello soddisfa tutte le ipotesi.

Come verificato durante l'analisi di correlazione, i regressori di primo grado più significativi (con p-value più basso) sono X_1 e X_4.

CAPITOLO 3

REGRESSIONE STEPWISE

3.1 Richiami teorici

Uno dei principali problemi che si riscontra nell'analisi di regressione è quello relativo alla scelta dei regressori da aggiungere al modello.
Nel modello di regressione andrebbero aggiunte tutte quelle variabili che sono statisticamente significative. Tuttavia, può accadere che una variabile sia significativa esclusivamente per un fattore aleatorio vincolato al dataset specifico, oppure che, per lo stesso motivo, una variabile in realtà significativa venga esclusa dal modello di regressione. Inoltre, può accadere che, se le variabili sono correlate tra loro, il contributo di una di esse può cambiare quando si introduce nel modello un'altra variabile.

Osservata la sorgente del tedioso problema, conviene cercare un numero maggiore di modelli e confrontarli fra di loro e solo alla fine intraprendere una decisione su quale sia il modello ottimo.
Esistono più strategie che permettono di costruire nuovi modelli. Le principali sono:

- Forward: si parte dal modello di regressione semplice costituito da β_0 e il regressore la cui correlazione con Y è massima; si procede aggiungendo un regressore alla volta, scegliendo sempre il più significativo (ovvero quello con la statistica test maggiore in valore assoluto). La procedura continua finché tutte le variabili presenti nel modello siano significative.

- Backward: si parte dal modello completo, contenente tutti i possibili regressori, per poi rimuovere i regressori meno significativi uno alla volta. A ogni rimozione, si procede a stimare il nuovo modello con le variabili restanti. Si prosegue fino a quando tutte le variabili siano significative. Per determinare la significatività delle variabili si possono usare le statistiche test

(ad esempio il t-test). Il rischio di questa procedura è quello di rimuovere prematuramente variabili che sarebbero risultate significative nei modelli con meno regressori.

- Hybrid: è una combinazione dei due approcci precedentemente descritti, si parte dal modello completo e si alternano fasi di backward e fasi di forward: la procedura forward è usata per verificare quali variabili inserire nel modello, e la backward per verificare quali eliminare.

Esiste un ulteriore approccio chiamato *Best Subset Selection* che prevede l'analisi, tramite un criterio di scelta, di tutti i possibili modelli di regressione che si ottengono dalla combinazione di n regressori. Tale strategia produce il miglior modello possibile, ma ha la grossa limitazione di non poter essere utilizzata se il numero di regressori è elevato poiché la complessità computazionale cresce in maniera esponenziale.

Tra i criteri di scelta possibili vi sono il massimo R^2_{adj}, il minimo AIC o BIC e il minimo errore di predizione. In particolare l'Akaike's Information Criterion (AIC) e il Bayesian Information Criterion (BIC) sono criteri di scelta che tengono conto sia della bontà di adattamento (favoriscono i modelli che meglio spiegano i dati) che del numero di regressori, penalizzando i modelli più complessi. Nel confrontare due modelli con AIC (rispettivamente BIC), si sceglierà quello con AIC (rispettivamente BIC) più basso. Le formule per AIC e BIC sono le seguenti:

$$AIC = -2\ln(L(\underline{\hat{\beta}})) + 2d$$

$$BIC = -2\ln(L(\underline{\hat{\beta}})) + d\ln(n)$$

con $L(\underline{\hat{\beta}})$ valore massimo della verosimiglianza del modello, d numero di predittori inclusi nel modello e n il numero di osservazioni. La differenza principale è che il BIC penalizza maggiormente l'aggiunta di regressori. Si osservi come penalizzare modelli più complessi serve a evitare il problema dell'overfitting, ovvero che il modello si adatti eccessivamente allo specifico dataset perdendo di generalità.

3.2 Modello lineare con regressori quadratici

Si vuole ora applicare un approccio stepwise partendo dal modello di regressione 2.19 per investigare la possibilità di aggiungervi regressori nella forma X_i^2. L'analisi preliminare dei dati [par. 1.2] ha infatti evidenziato che la dipendenza tra X_3 e Y potrebbe essere meglio spiegata da una relazione non lineare. In particolare, si è intuito che il regressore X_3^2 potrebbe essere particolarmente significativo ai fini della regressione.

3.2.1 Ricerca del modello

Disponendo già di un modello da cui partire, risulta naturale scegliere un approccio forward. Si utilizza il comando `update()` per aggiungere separatamente i regressori $X_1^2, ..., X_6^2$ al modello e si procede a trovare quello col valore della statistica test maggiore (in valore assoluto) col seguente codice:

```
> poly_fit=vector(mode = "list", length = 6)
> base_model=lin_fit_2
> poly_fit[[1]]=update(base_model,.~.+I(x1_CPU^2))
> poly_fit[[2]]=update(base_model,.~.+I(x2_HD^2))
> poly_fit[[3]]=update(base_model,.~.+I(x3_proc^2))
> poly_fit[[4]]=update(base_model,.~.+I(x4_aging^2))
> poly_fit[[5]]=update(base_model,.~.+I(x5_audio^2))
> poly_fit[[6]]=update(base_model,.~.+I(x6_RAM^2))
> t.values=c()
> for(i in 1:6){
    t.values[i]=summary(poly_fit[[i]])[["coefficients"]][6,"t value"]
  }
> which.max(abs(t.values))
[1] 3
```

Per cui il prossimo regressore da aggiungere è X_3^2 e quanto ipotizzato in precedenza ha trovato conferma. Si controlla ora se il regressore aggiunto è statisticamente significativo con un livello di confidenza $1 - \alpha = 0.95$:

```
> summary(poly_fit[[3]])

Call:
lm(formula = y_prestazSWcalc ~ x1_CPU + x3_proc + x4_aging +
    x6_RAM + I(x3_proc^2), data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5604  -3.2001   0.1039   2.8136   9.8790

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   50.1033     0.7100  70.570  < 2e-16 ***
x1_CPU         6.1899     0.4718  13.119  < 2e-16 ***
x3_proc        2.8389     0.4809   5.903 5.62e-08 ***
x4_aging      -4.9525     0.4659 -10.631  < 2e-16 ***
x6_RAM         2.0734     0.4794   4.325 3.80e-05 ***
I(x3_proc^2)  -5.0749     0.5485  -9.252 7.16e-15 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 4.574 on 94 degrees of freedom
Multiple R-squared:  0.8522,  Adjusted R-squared:  0.8444
F-statistic: 108.4 on 5 and 94 DF,  p-value: < 2.2e-16
```

Il regressore è statisticamente significativo in quanto *p-value* $< \alpha$. Si osservi che il coefficiente di determinazione corretto è aumentato significativamente con un valore $R^2_{adj} = 0.8444$. Si investiga

ora quale sia il prossimo regressore da aggiungere secondo l'approccio forward:

```
> base_model=poly_fit[[3]]
> poly_fit=vector(mode = "list", length = 5)
> poly_fit[[1]]=update(base_model,.~.+I(x1_CPU^2))
> poly_fit[[2]]=update(base_model,.~.+I(x2_HD^2))
> poly_fit[[3]]=update(base_model,.~.+I(x4_aging^2))
> poly_fit[[4]]=update(base_model,.~.+I(x5_audio^2))
> poly_fit[[5]]=update(base_model,.~.+I(x6_RAM^2))
> t.values=c()
> for(i in 1:5){
    t.values[i]=summary(poly_fit[[i]])[["coefficients"]][7,"t value"]
  }
> which.max(abs(t.values))
[1] 4
```

Per cui il prossimo regressore da aggiungere sarebbe X_5^2. Ciononostante, analizzando il modello ottenuto, si evince che tale regressore non è statisticamente significativo.

```
> summary(poly_fit[[4]])

Call:
lm(formula = y_prestazSWcalc ~ x1_CPU + x3_proc + x4_aging +
    x6_RAM + I(x3_proc^2) + I(x5_audio^2), data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-12.9634  -2.6957   0.1554   2.8040   9.5560

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    50.4941     0.8509  59.345  < 2e-16 ***
x1_CPU          6.1269     0.4786  12.803  < 2e-16 ***
x3_proc         2.8309     0.4818   5.876 6.50e-08 ***
x4_aging       -4.9525     0.4666 -10.614  < 2e-16 ***
x6_RAM          2.0729     0.4802   4.317 3.95e-05 ***
I(x3_proc^2)   -5.0462     0.5505  -9.167 1.18e-14 ***
I(x5_audio^2)  -0.4235     0.5062  -0.837    0.405
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 4.581 on 93 degrees of freedom
Multiple R-squared:  0.8533,  Adjusted R-squared:  0.8439
F-statistic: 90.18 on 6 and 93 DF,  p-value: < 2.2e-16
```

Risulta infatti *p-value* $> \alpha$, per cui tale regressore non andrà inserito nel modello e la regressione stepwise può essere interrotta. Si osservi come anche il coefficiente di dererminazione corretto sia addirittura peggiorato, col valore $R^2_{adj} = 0.8439$. La regressione stepwise ha portato quindi al nuovo modello:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_3 \boldsymbol{X}_3 + \beta_4 \boldsymbol{X}_4 + \beta_6 \boldsymbol{X}_6 + \beta_7 \boldsymbol{X}_3^2 + \boldsymbol{\varepsilon} \tag{3.1}$$

Gli intervalli di confidenza per il nuovo modello, con un livello di confidenza $1 - \alpha = 0.95$, sono:

$$48.693658 \leq \beta_0 < 51.513041$$

$$5.253084 \leq \beta_1 < 7.126766$$
$$1.884077 \leq \beta_3 < 3.793809$$
$$-5.877521 \leq \beta_4 < -4.027543$$
$$1.121511 \leq \beta_6 < 3.025359$$
$$-6.164038 \leq \beta_7 < -3.985850$$

3.2.2 Analisi del modello

Dai test eseguiti su R consegue che il modello soddisfa tutte le ipotesi. Inoltre, si riportano in seguito i grafici diagnostici:

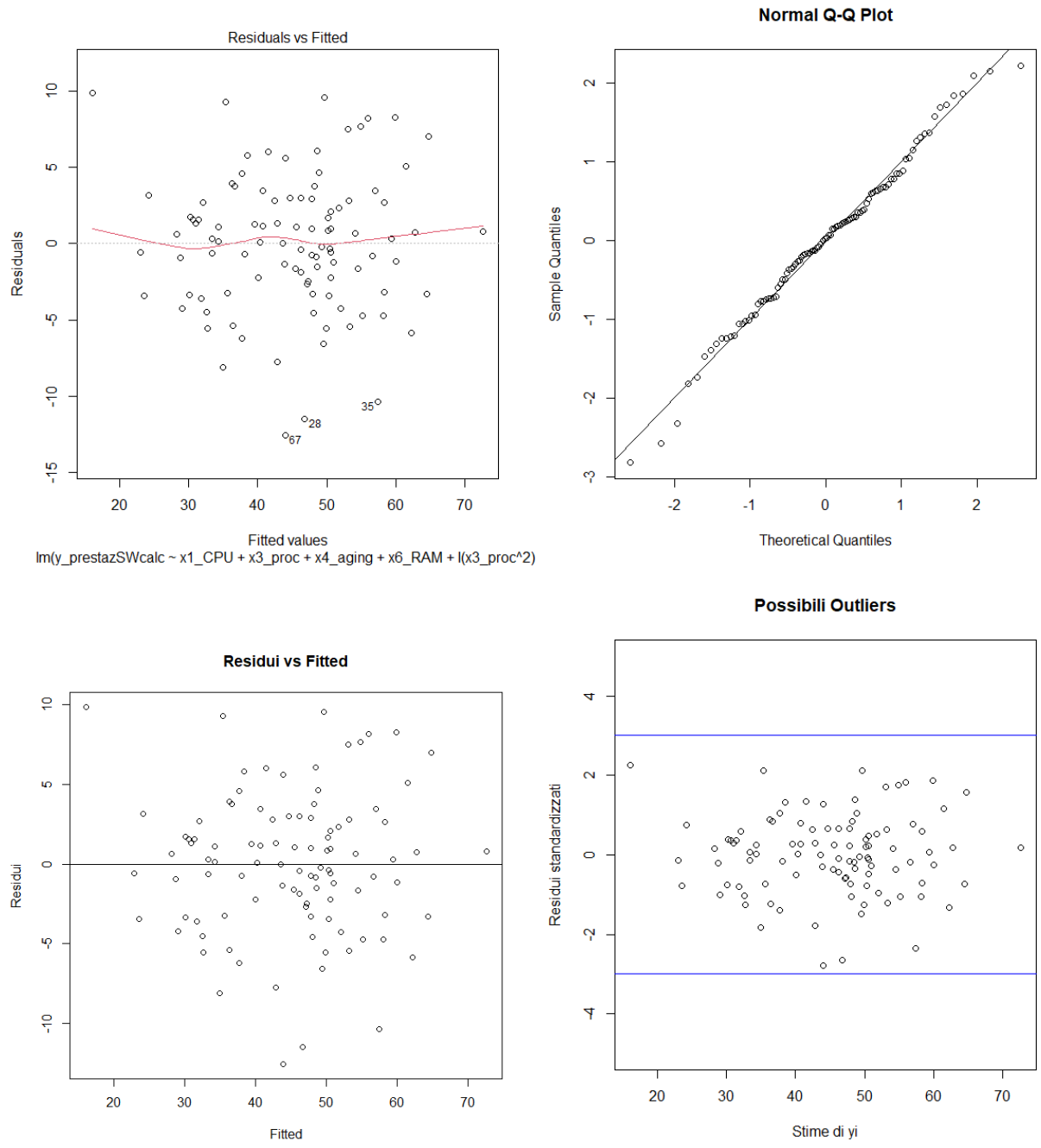

Figura 3.1: Scatterplot residui-stime, Normal Q-Q Plot, Scatterplot residui-stime, Possibili outliers

3.3 Modello lineare con termini di interazione

Il package `leaps` mette a disposizione la funzione `regsubsets()` che permette di costruire automaticamente modelli di regressione con l'approccio Best Subset Selection: a ogni step i vengono paragonati tutti i possibili modelli con i regressori per trovare il migliore. Per selezionare un modello tra quelli prodotti è poi possibile usare uno dei criteri di scelta (AIC,BIC,ecc).

3.3.1 Ricerca del modello

Si sfrutta tale funzione per valutare di aggiungere al modello termini incrociati del tipo $X_i X_j$ *con* $i \neq j$, i quali modellano le possibili interazioni tra coppie di regressori. In particolare si considerano tutti i regressori lineari, quadratici, e incrociati:

```
> bss_1=regsubsets(y_prestazSWcalc ~ x1_CPU + x2_HD + x3_proc + x4_aging + x5_audio
                  + x6_RAM + I(x1_CPU^2) + I(x2_HD^2) + I(x3_proc^2) + I(x4_aging^2)
                  + I(x5_audio^2) + I(x6_RAM^2) + (.)^2
                  , data = ds, nvmax = 10)
> sum_bss_1=summary(bss_1)
> print(sum_bss_1)
[OMISS]
Selection Algorithm: exhaustive
          x1_CPU x2_HD x3_proc x4_aging x5_audio x6_RAM I(x1_CPU^2) I(x2_HD^2)
1  ( 1 )  "*"    " "   " "     " "      " "      " "    " "          " "
2  ( 1 )  "*"    " "   " "     " "      " "      " "    " "          " "
3  ( 1 )  "*"    " "   " "     "*"      " "      " "    " "          " "
4  ( 1 )  "*"    " "   "*"     "*"      " "      " "    " "          " "
5  ( 1 )  "*"    " "   "*"     "*"      " "      "*"    " "          " "
6  ( 1 )  "*"    " "   "*"     "*"      " "      "*"    " "          " "
7  ( 1 )  "*"    " "   "*"     "*"      " "      "*"    " "          " "
8  ( 1 )  "*"    " "   "*"     "*"      " "      "*"    " "          " "
9  ( 1 )  "*"    " "   "*"     "*"      " "      "*"    " "          " "
10  ( 1 ) "*"    " "   "*"     "*"      " "      "*"    " "          " "
          I(x3_proc^2) I(x4_aging^2) I(x5_audio^2) I(x6_RAM^2) x1_CPU:x2_HD
1  ( 1 )  " "          " "           " "           " "          " "
2  ( 1 )  "*"          " "           " "           " "          " "
3  ( 1 )  "*"          " "           " "           " "          " "
4  ( 1 )  "*"          " "           " "           " "          " "
5  ( 1 )  "*"          " "           " "           " "          " "
6  ( 1 )  "*"          " "           " "           " "          " "
7  ( 1 )  "*"          " "           " "           " "          " "
8  ( 1 )  "*"          " "           " "           " "          " "
9  ( 1 )  "*"          " "           "*"           " "          " "
10  ( 1 ) "*"          " "           "*"           " "          " "
          x1_CPU:x3_proc x1_CPU:x4_aging x1_CPU:x5_audio x1_CPU:x6_RAM x2_HD:x3_proc
1  ( 1 )  " "            " "             " "             " "            " "
2  ( 1 )  " "            " "             " "             " "            " "
3  ( 1 )  " "            " "             " "             " "            " "
4  ( 1 )  " "            " "             " "             " "            " "
5  ( 1 )  " "            " "             " "             " "            " "
6  ( 1 )  " "            " "             " "             " "            " "
7  ( 1 )  "*"            " "             " "             " "            " "
8  ( 1 )  "*"            " "             " "             " "            " "
```

```
9  ( 1 )  "*"             " "              " "              " "              " "
10  ( 1 ) "*"             "*"              " "              " "              " "
          x2_HD:x4_aging x2_HD:x5_audio x2_HD:x6_RAM x3_proc:x4_aging x3_proc:x5_audio
1  ( 1 )  " "             " "              " "              " "              " "
2  ( 1 )  " "             " "              " "              " "              " "
3  ( 1 )  " "             " "              " "              " "              " "
4  ( 1 )  " "             " "              " "              " "              " "
5  ( 1 )  " "             " "              " "              " "              " "
6  ( 1 )  "*"             " "              " "              " "              " "
7  ( 1 )  "*"             " "              " "              " "              " "
8  ( 1 )  "*"             " "              " "              " "              " "
9  ( 1 )  "*"             " "              " "              " "              " "
10  ( 1 ) "*"             " "              " "              " "              " "
          x3_proc:x6_RAM x4_aging:x5_audio x4_aging:x6_RAM x5_audio:x6_RAM
1  ( 1 )  " "             " "                " "              " "
2  ( 1 )  " "             " "                " "              " "
3  ( 1 )  " "             " "                " "              " "
4  ( 1 )  " "             " "                " "              " "
5  ( 1 )  " "             " "                " "              " "
6  ( 1 )  " "             " "                " "              " "
7  ( 1 )  " "             " "                " "              " "
8  ( 1 )  "*"             " "                " "              " "
9  ( 1 )  "*"             " "                " "              " "
10  ( 1 ) "*"             " "                " "              " "
```

Prendendo come criterio di scelta quello del minimo BIC è possibile banalmente verificare che il modello che minimizza tale indice è quello a 6 regressori:

```
> which.min(sum_bss_1$bic)
[1] 6
```

ovvero il modello

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_2 \boldsymbol{X}_3 + \beta_3 \boldsymbol{X}_4 + \beta_4 \boldsymbol{X}_6 + \beta_5 \boldsymbol{X}_3^2 + \beta_6 \boldsymbol{X}_2 \boldsymbol{X}_4 + \boldsymbol{\varepsilon} \tag{3.2}$$

3.3.2 Analisi del modello

Si passa ora ad analizzarare il modello:

```
> bss_fit_1=lm(y_prestazSWcalc ~ x1_CPU + x3_proc + x4_aging + x6_RAM + I(x3_proc^2)
  + x2_HD:x4_aging,data=ds)
> sum_bss_fit_1=summary(bss_fit_1)
> print(sum_bss_fit_1)

Call:
lm(formula = y_prestazSWcalc ~ x1_CPU + x3_proc + x4_aging +
    x6_RAM + I(x3_proc^2) + x2_HD:x4_aging, data = ds)

Residuals:
     Min       1Q   Median       3Q      Max
-10.4888  -2.6449   0.0814   2.9168  11.4992

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)     49.7285     0.6946  71.589  < 2e-16 ***
x1_CPU           6.3246     0.4561  13.867  < 2e-16 ***
x3_proc          3.1726     0.4763   6.662 1.88e-09 ***
x4_aging        -4.9493     0.4480 -11.047  < 2e-16 ***
x6_RAM           1.9374     0.4634   4.181 6.57e-05 ***
I(x3_proc^2)    -4.8694     0.5321  -9.151 1.28e-14 ***
x4_aging:x2_HD  -1.4814     0.5043  -2.938  0.00417 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 4.399 on 93 degrees of freedom
Multiple R-squared:  0.8648,  Adjusted R-squared:  0.8561
F-statistic: 99.13 on 6 and 93 DF,  p-value: < 2.2e-16
```

È possibile osservare un aumento del coefficiente di determinazione corretto ($R^2_{adj} = 0.8561$).

Gli intervalli di confidenza per i parametri, con un livello di confidenza $1 - \alpha = 0.95$, sono:

$$48.349057 \leq \beta_0 < 51.1078815$$
$$5.418921 \leq \beta_1 < 7.2303304$$
$$2.226853 \leq \beta_2 < 4.1183356$$
$$-5.839048 \leq \beta_3 < -4.0596109$$
$$1.017153 \leq \beta_4 < 2.8576155$$
$$-5.926159 \leq \beta_5 < -3.8126901$$
$$-2.482785 \leq \beta_6 < -0.4799826$$

Dai test eseguiti su R consegue che il modello soddisfa tutte le ipotesi. Inoltre, si riportano in seguito i grafici diagnostici:

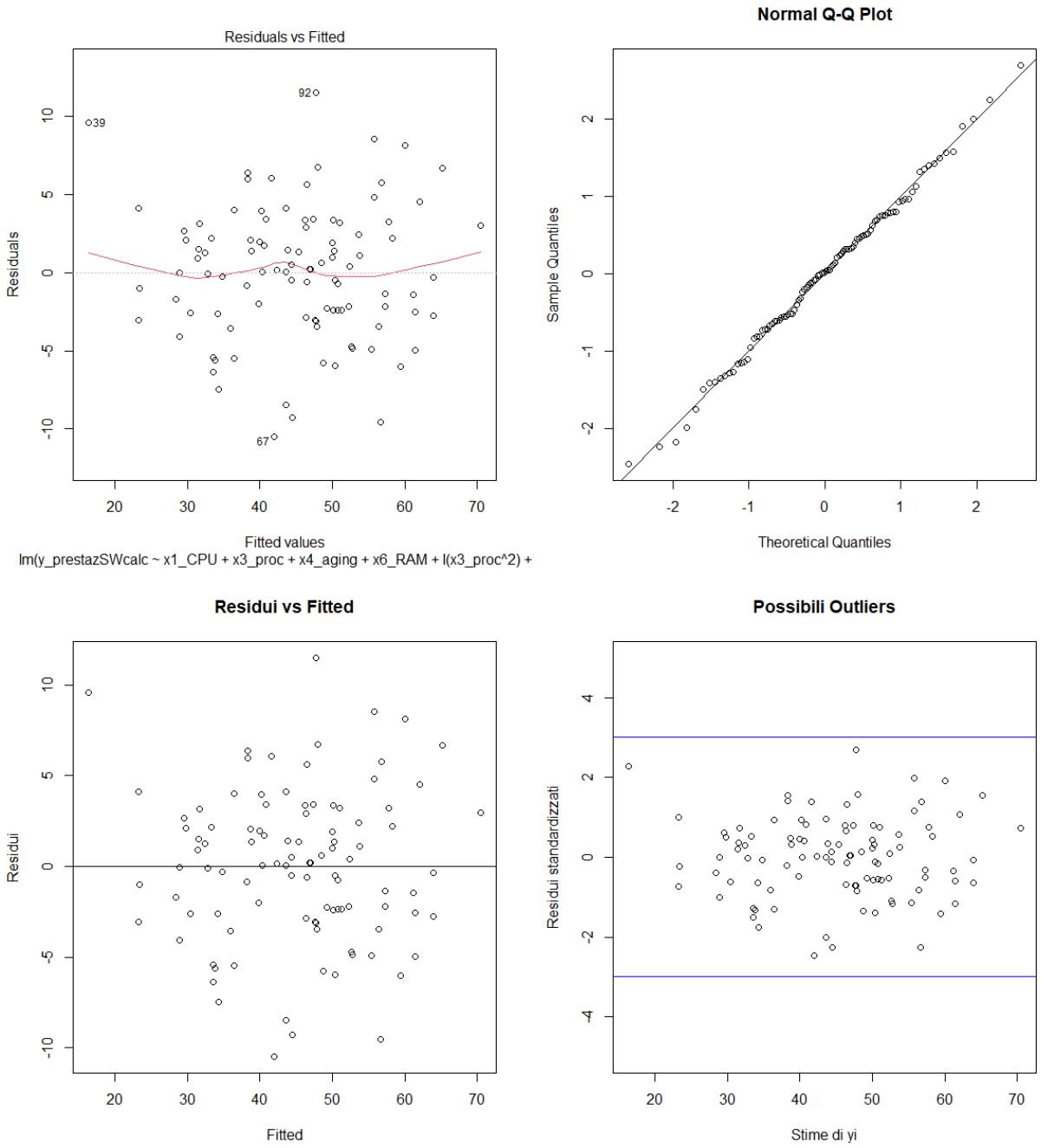

Figura 3.2: Scatterplot residui-stime, Normal Q-Q Plot, Scatterplot residui-stime, Possibili outliers

3.4 Modello lineare con termini esponenziali

Si valuta ora un modello più complesso, ottenuto con `regsubsets()` tenendo in considerazione altri regressori non lineari. In particolare si sono presi in considerazione i regressori nella forma e^{X_i}, e^{-X_i} e X_i^3, oltre a quelli già considerati nei modelli precedenti. Si noti che si è deciso di trascurare i termini incrociati di ordine superiore al secondo per evitare di introdurre regressori troppo complessi nel modello che possano favorire l'overfitting, nonostante i regressori $X_1X_4X_5$ e $X_4X_5X_6$ siano risultati statisticamente significativi: si è interpretato tale fenomeno come un fatto

aleatoreo legato al dataset poiché si è pensato che fosse inverosimile che l'interazione di più di due variabili fosse significativa nel caso in esame. Inoltre si decide di omettere il listato per motivi di compattezza. Applicando il criterio del minimo BIC (come è stato fatto nel paragrafo 3.3.1) si ottiene il seguente modello a 6 regressori:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_2 \boldsymbol{X}_4 + \beta_3 \boldsymbol{X}_6 + \beta_4 \boldsymbol{X}_3^2 + \beta_5 e^{\boldsymbol{X_3}} + \beta_6 \boldsymbol{X}_2 \boldsymbol{X}_4 + \boldsymbol{\varepsilon} \tag{3.3}$$

È possibile notare come l'introduzione del regressore e^{X_3} abbia reso statisticamente non significativo il regressore X_3, che per l'appunto non è più presente nel modello. Inoltre i regressori nella forma e^{-X_i} e X_i^3 non sono risultati abbastanza significativi da rientrare nel modello. Analizzando il nuovo modello con il solito approccio, si osserva come il coefficiente di determinazione corretto abbia subito un incremento modesto, a fronte di un aumento della complessità del modello.

```
> bss_fit_2=lm(y_prestazSWcalc ~ x1_CPU + x4_aging + x6_RAM + I(x3_proc^2)
  + I(exp(x3_proc)) + x2_HD:x4_aging,data=ds)
> sum_bss_fit_2=summary(bss_fit_2)
> print(sum_bss_fit_2)

Call:
lm(formula = y_prestazSWcalc ~ x1_CPU + x4_aging + x6_RAM + I(x3_proc^2) +
    I(exp(x3_proc)) + x2_HD:x4_aging, data = ds)

Residuals:
     Min       1Q   Median       3Q      Max
-10.8386  -2.6975   0.1827   2.7957  11.2888

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       47.2424     0.8477  55.727  < 2e-16 ***
x1_CPU             6.3792     0.4566  13.970  < 2e-16 ***
x4_aging          -4.9157     0.4473 -10.989  < 2e-16 ***
x6_RAM             1.9581     0.4624   4.235 5.37e-05 ***
I(x3_proc^2)      -6.1476     0.5225 -11.765  < 2e-16 ***
I(exp(x3_proc))    2.4471     0.3661   6.684 1.70e-09 ***
x4_aging:x2_HD    -1.4794     0.5036  -2.938  0.00417 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 4.394 on 93 degrees of freedom
Multiple R-squared:  0.8651,  Adjusted R-squared:  0.8564
F-statistic: 99.38 on 6 and 93 DF,  p-value: < 2.2e-16
```

Gli intervalli di confidenza trovati, con un livello di confidenza $1 - \alpha = 0.95$, sono:

$$45.558924 \leq \beta_0 < 48.9258109$$

$$5.472472 \leq \beta_1 < 7.2859928$$

$$-5.803955 \leq \beta_2 < -4.0274221$$

$$1.039867 \leq \beta_3 < 2.8762942$$

$$-7.185221 \leq \beta_4 < -5.1099221$$

$$1.720099 \leq \beta_5 < 3.1741526$$

$$-2.479460 \leq \beta_6 < -0.4793784$$

3.4.1 Analisi del modello

Dai test eseguiti su R consegue che il modello soddisfa tutte le ipotesi fatte. Inoltre, si riportano in seguito i grafici diagnostici:

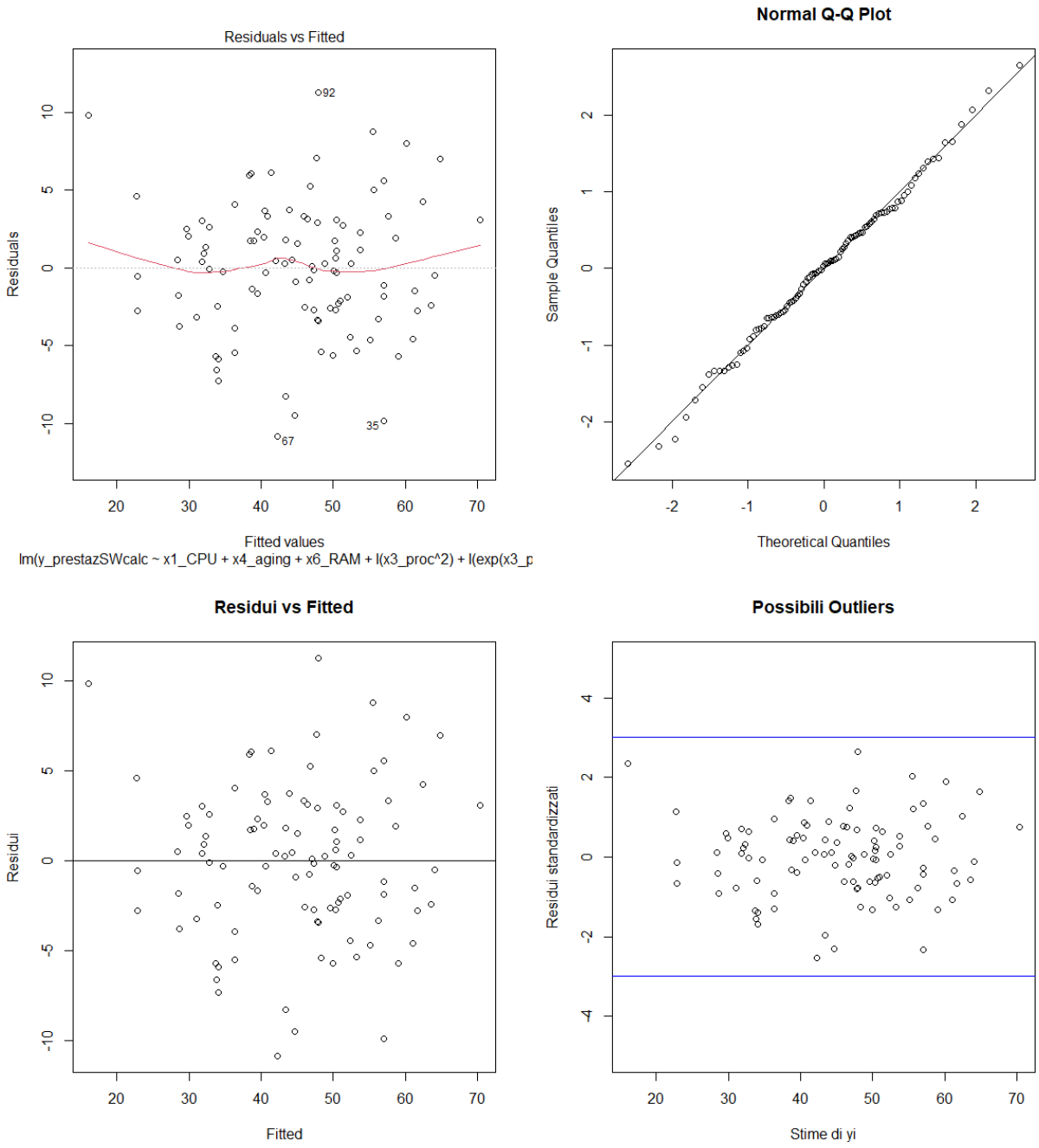

Figura 3.3: Scatterplot residui-stime, Normal Q-Q Plot, Scatterplot residui-stime, Possibili outliers

3.5 Conclusioni

Nonostante il fatto che, secondo il criterio del minimo BIC (o del massimo R^2_{adj}, avendo i modelli lo stesso numero di predittori), il modello 3.3 sia migliore del precendente, si potrebbe comunque preferire il modello 3.2: l'aumento del coefficiente di determinazione nel modello 3.2 è infatti molto piccolo, per cui si preferisce un modello più semplice e meno prono a overfitting. Il modello scelto a valle dell'analisi di regressione è quindi:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_2 \boldsymbol{X}_3 + \beta_3 \boldsymbol{X}_4 + \beta_4 \boldsymbol{X}_6 + \beta_5 \boldsymbol{X}_3^2 + \beta_6 \boldsymbol{X}_2 \boldsymbol{X}_4 + \boldsymbol{\varepsilon}$$

Si noti che, in linea di principio, per costruire il modello sarebbe stato possibile utilizzare tecniche di cross-validation al fine di dare priorità alla diminuzione dell'overfitting. La tecnica di cross-validation "hold out" prevede ad esempio di dividere il dataset in due parti (training set e test set): si costruisce il modello di regressione sul training set e si sceglie il modello che minimizza l'errore di predizione sul test set. Si è scelto tuttavia di non utilizzare tale tecnica data la bassa cardinalità del dataset: si è ritenuto che per ottenere un buon modello ogni osservazione fosse importante e preziosa. Per semplicità, inoltre, non sono state utilizzate tecniche di cross-validation avanzate adatte a piccoli dataset come la "k-fold" cross-validation.

Alcuni regressori complessi, come i termini incrociati di ordine superiore al secondo, sono stati deliberatamente ignorati ai fini della Best Subsets Selection per evitare l'overfitting del modello al dataset. Considerando anche tali regressori, il modello scelto secondo il criterio BIC sarebbe stato:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_2 \boldsymbol{X}_4 + \beta_3 \boldsymbol{X}_3^2 + \beta_4 e^{-\boldsymbol{X}_6} + \beta_5 e^{\boldsymbol{X}_3} + \beta_6 \boldsymbol{X}_2 \boldsymbol{X}_4 + \beta_7 \boldsymbol{X}_4 \boldsymbol{X}_5 \boldsymbol{X}_6 + \beta_8 \boldsymbol{X}_1 \boldsymbol{X}_4 \boldsymbol{X}_6 + \boldsymbol{\varepsilon}$$

con $R^2_{adj} = 0.8758$, ma si è ritenuto che tale modello fosse eccessivamente complesso.