REGRESSIONE LINEARE

Quando si osserva un fenomeno reale tre sono i passi da svolgere per cercare di costruire un modello che lo descriva a pieno senza errori o fraintendimenti:

- *semplificazione della realtà:* riproduzione degli aspetti essenziali ed eliminazione di quelli ritenuti superflui;
- *analogia con la realtà:* il modello deve essere una riproduzione della realtà;
- *rappresentazione necessaria della realtà:* anche se è semplificato il modello è necessario per capire la realtà grazie alle relazioni semplici che lo compongono.

Il modello da cui iniziamo è quello della *Regressione lineare;* esso ci fornisce una legge che ci permette di capire se i dati che stiamo osservando si adattano o meno a distribuirsi lungo una retta. Appare chiaro come quindi non sempre tale modello potrà risultare applicabile: ci saranno casi in cui le osservazioni che avremo a disposizione, seguendo un comportamento lineare, vi si adatteranno bene, altri invece meno.

La struttura che presenta la regressione lineare semplice è la seguente:

$$y = ax + b$$

dove:

- y = variabile dipendente (l'output: quello che vogliamo saper predire);
- x = variabili indipendenti (chiamate anche predittori o input);
- a = coefficiente angolare (l'inclinazione della retta);
- b = costante.

Tale algoritmo ha quindi lo scopo di valutare, entro i limiti dei dati osservati, come variabile dipendente e indipendente dipendano o si influenzino fra loro: *quale possa essere il valore della prima al variare della seconda.*
A tale equazione va inoltre aggiunta una certa percentuale di errore (è impensabile di non farne) che punta ad essere la minore possibile grazie alla **regola dei minimi quadrati**.

La storia della regressione lineare vede le sue origini tra la fine del '700 e gli inizi del '800 ad opera di Adrien-Marie Legendre e Carl Friedrich Gauss. Sebbene la paternità di tale scoperta venga normalmente attribuita al secondo in realtà essa venne concepita in modo disgiunto da entrambi basandosi, per l'appunto sulla sopra citata regola dei minimi quadrati.
Successivamente l'impiego in tale contesto, del termine *Regressione*, col quale ancora oggi è conosciuta, si deve grazie al lavoro svolto al biologo Francis Galton che nel 1886 esaminando le altezze dei figli (Y) in funzione di quelle dei genitori (X) vi notò la presenza di una relazione funzionale: più alti erano i genitori e più alti si presentavano i figli, e viceversa. Tuttavia per i genitori che si collocavano agli estremi (molto bassi o molto alti) non corrispondevano figli altrettanto estremi. Da tale osservazione se ne concluse che l'altezza dei figli si spostava verso un valore medio costituendo quindi una *regression towards mediocrity.* Ecco che tale relazione prese il nome di "modello di regressione".

Nello specifico, la regola dei minimi quadrati sul quale si fonda l'algoritmo, si basa sulla sommatoria delle differenze (chiamate anche scarti o errori), che vi sono fra i vari punti osservati (x_i, y_i) e i loro reali corrispettivi presenti nella retta di regressione (x_i, $ax_i + b$); tale somma di differenze viene poi elevata al quadrato in modo da enfatizzarne l'effetto su ciascun punto (quelli più vicini alla retta avranno un peso minore, mentre quelli lontani maggiore), per poi individuarne il minimo essendo interessati a trovare la funzione ottima.

La funzione da minimizzare è dunque: $$\phi(a,b) = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

Il grafico a seguire mostra bene quanto appena enunciato.

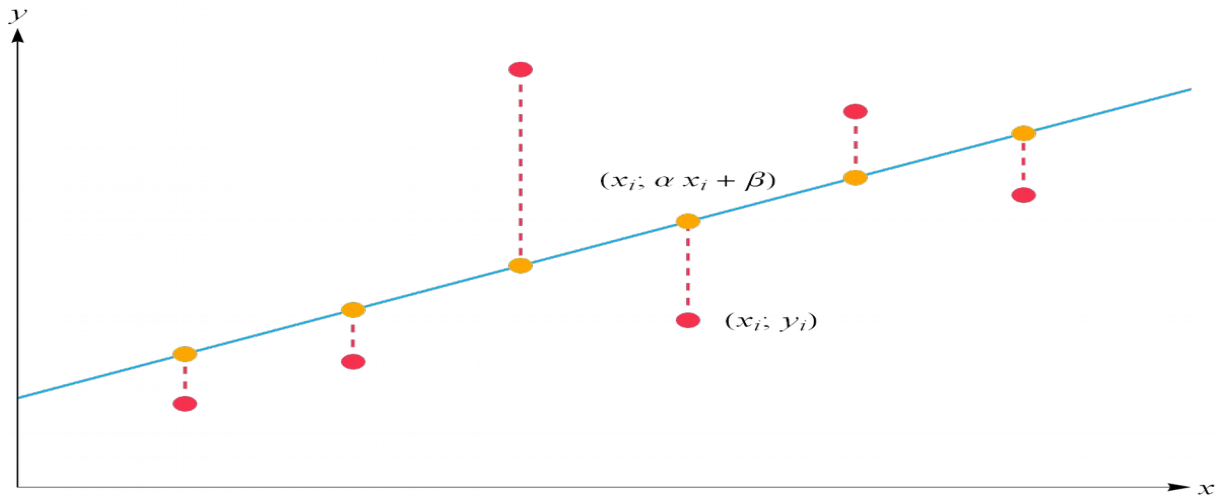


Figura 1: Esempio di retta di Regressione lineare

A questo punto fondamentale risulta porsi una domanda:

Come si fa a capire la bontà del nostro modello di regressione lineare e in che misura?

Molto utile a tal proposito è la misura del **coefficiente di determinazione (R^2)**; esso infatti ci permette proprio di capire quanto buono è il nostro modello, affermando se esso dia informazioni in più o in meno rispetto ad un modello di riferimento, individuato facendo la media dei valori di y. Questo infatti risulta essere il modello di riferimento più adeguato presentando degli errori molto elevati a causa della non conoscenza delle variabili indipendenti.

Nello specifico:

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

dove:

- TSS: *devianza totale*, indica quanto spiega il modello di riferimento (calcolato sulla media), rispetto ai valori osservati;
- ESS: *devianza spiegata*, ossia quanto bene spiega il modello che ho ottenuto dalla regressione rispetto al modello di riferimento;
- RSS: *devianza residua*, ciò che non viene spiegato dal mio modello (l'errore).

In particolare si vuole che la devianza totale e quella spiegata siano il più simili possibile, in modo che la quantità di non spiegato sia minima.

Ecco quindi che più tale coefficiente risulta prossimo ad 1 maggiore è la precisione e la bontà con cui spiega i dati, fino ad arrivare ad 1 dove li spiega perfettamente; al contrario un valore di R^2 prossimo allo 0 sta ad indicare una mancanza di adattamento del modello a quanto osservato con conseguente qualità della previsione pessima (non va a spiegare nulla di più di quanto predetto dal modello di riferimento).

Una particolare attenzione è comunque da porsi ai valori troppo elevati di R^2: essi potrebbero voler significare che diamo troppa fiducia ai dati, con conseguente rischio di overfitting. Ecco che un valore di circa l'85%, con conseguente 0,15 di non spiegato, è già da considerarsi un risultato soddisfacente.

Un altro discorso da affrontare è poi *il come* i dati a cui si applicata la regressione si distribuiscono. Nel caso infatti in cui essi manifestino una forma non propriamente idonea per l'algoritmo (e.g: seguano una distribuzione a parabola invece che a retta), risulta essere molto utile eseguirvi delle trasformazioni,

differenti da caso a caso, e solo dopo applicarvi l’algoritmo di regressione. Nel caso di una forma originaria a parabola si potrebbe, ad esempio, eseguirvi la radice, in modo da annullare l’azione dell’elevazione a potenza; od ancora, per dati eteroschedastici (distribuzione a triangolo), dove la varianza cresce col progredire dell’asse delle ordinate (asse x), caratterizandosi per la prevalenza delle operazioni di % e * (un animale mangia in percentuale al suo peso, e più è grande maggiore sarà la quantità ingerita) può essere risolta trasformando tali operazioni in somme.

R: Analisi dei dati e confutazione dei risultati ottenuti

Grazie all'impiego di R si sono analizzati vari tipi di collezioni di dati; se ne sono osservate le distribuzioni, vi ci sono stati applicati modelli di regressione lineari predittivi e i risultati così ottenuti sono stati poi esaminati con senso critico. In tal modo si sono potuti individuare pattern comportamentali specifici che dessero una spiegazione rigorosa a quanto osservato ed ottenuto.

Dataset *iris*

Viene qui mostrata l'analisi del dataset *Iris*; vi si include il codice relativo al linguaggio R accompagnandolo con le immagini e i commenti dei risultati ottenuti:

```
library(datasets)
data(iris)

help(iris) ##mi da' informazioni sul dataset
summary(iris) ##mi permette di vedere la "tabella" di iris

dim(iris) ##mi dice che ho  entry con 5 colonne in totale
n <- nrow(iris) ##assegno ad n=150
n #stampo n

summary(iris$Petal.Width)  ##mi da' informazioni utili sulla variabile (mediana, quartili ecc.)

plot(iris$Petal.Length,iris$Sepal.Width)  ##mostra il grafico (x, y): noto che vi è una buona separazione dei dati
```

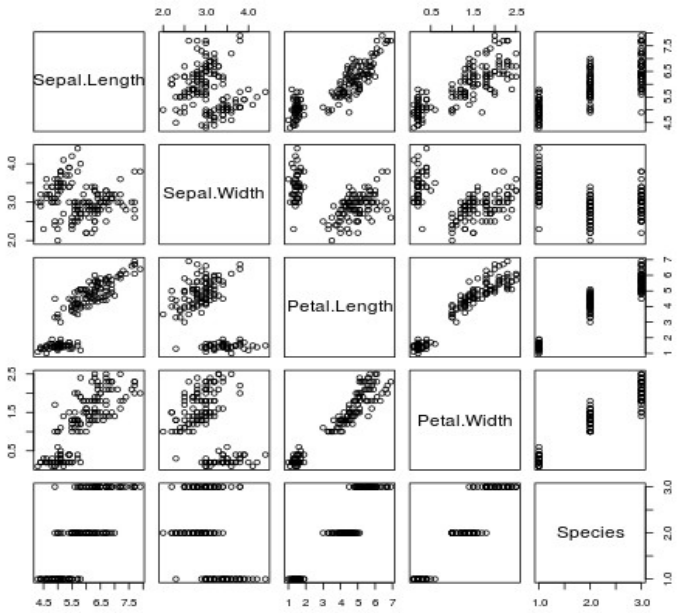


Figura 1: Vari grafici di Iris

```
plot(iris) #mostra i vari grafici di iris; posso capire quali sono gli assi che di volta in volta mostrano una distribuzione migliore dei punti, es. y=Petal.Lenght  e x=Petal.Width
```

lm(Sepal.Length ~ Petal.Width, data=iris) ##creo il modello di regressione (y, predittore), inoltre mi da' informazioni utili su intercetta e coefficiente angolare

modello <- lm(Sepal.Length ~ Petal.Width, data=iris)
##assegno il modello a una variabile cosi la posso usare con più semplicità

summary(modello)
##mi da' informazioni utili sulla regressione fatta

```
> summary(modello)

Call:
lm(formula = Sepal.Length ~ Petal.Width, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.38822 -0.29358 -0.04393  0.26429  1.34521

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.77763    0.07293   65.51   <2e-16 ***
Petal.Width  0.88858    0.05137   17.30   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.478 on 148 degrees of freedom
Multiple R-squared:  0.669,     Adjusted R-squared:  0.6668
F-statistic: 299.2 on 1 and 148 DF,  p-value: < 2.2e-16
```

Figura 2: Informazioni modello

In particolare:

- noto che per quanto riguarda i residui la loro distribuzione è buona in quanto la mediana è prossima allo zero (circa metà) e 1 e 3 quartile sono rispettivamente -0.29... e 0.26...., inoltre anche min e max sono simmetrici;

- abbiamo:

  h0: x e y non sono dipendenti
  h1: x e y sono dipendenti

  - ✗ **valori della retta**: intercetta e coefficiente angolare;
  - ✗ **standard error**: indica la distanza delle stime dal valore vero, valuta la precisione. Qui la distanza e' molto piccola, quindi abbiamo una buona stima;
  - ✗ **t-value**: valore generato dal test t; presenta un valore pari a 0 quando h0 vale, che cresce man mano l'ipotesi nulla non è più verificabile. Qui i valori sono elevati caratteristica che mi porta a rifiutare l'ipotesi nulla;
  - ✗ **livello di significatività del test**: posso vederlo come il p-value ossia il minor valore per cui rifiuto h0 (e' improbabile che la relazione fra x e y sia dovuta al caso, ecco quindi che sono sicuramente dipendenti fra lori). Più basso è il suo valore più il risultato è significativo. Qui e' molto basso. Inoltre ci sono i 3 *** che mi dicono che tale predittore è molto importante.

- *RSE*: misura la distanza media tra i valori stimati e quelli osservati; più piccolo è il suo valore migliore è l'adattamento del modello hai dati; qui risulta piccolo;
- *R2* è comunque un valore abbastanza buono spiegando lo 0.6 della devianza (probabilmente ci sarà qualche modello migliore);

- *F-statistica*: corrisponde alla statistica-test, da' un giudizio complessivo sulla bontà esplicativa del modello: probabilità che il modello non sia significativo. Se F=1 allora non vi e' alcuna relazione tra y e x, d'altra parte come in questo caso, con F>1 accetto h1.

Conclusione: t-value e statistica-F presentano valori ampiamente superiori ai valori tabulati (difatti i relativi p-value sono di molto inferiori a 0.05). S*i rifiuta pertanto l'ipotesi nulla*:

La regressione è significativa (il valore del coefficiente angolare così calcolato è statisticamente differente da zero).

```
plot(modello)
##mi mostra vari grafici del mio modello di regressione
```

Residui vs Leverage: evidenzia se ci sono valori anomali influenti, ovvero la cui eliminazione porterebbe una sostanziale variazione al modello di regressione. Essendo che tutti i punti si trovano entro le linee di distanza di Cook non è questo il caso;

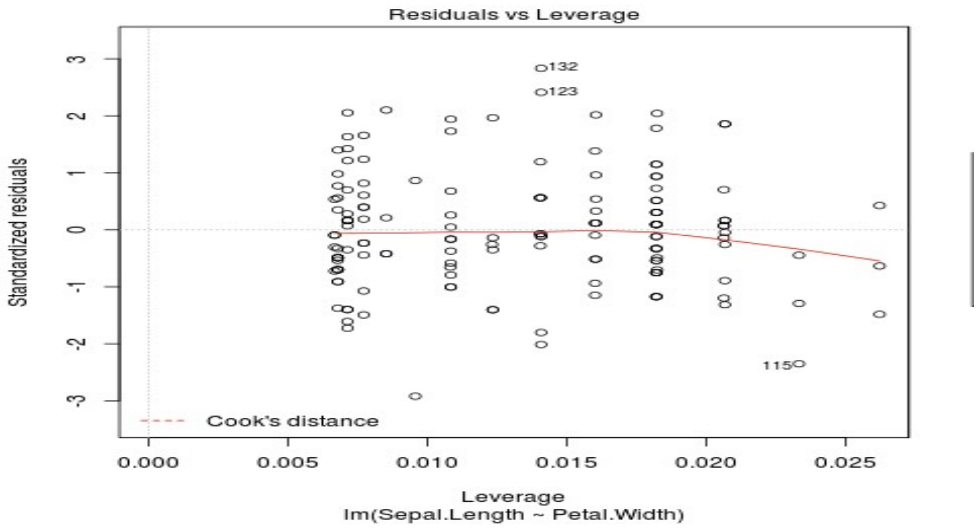


Figura 3: Residui vs Leverage

Scaled-Location: mostra se i residui sono distribuiti equamente lungo gli intervalli dei predittori; rappresenta inoltre il modo in cui è possibile verificare l'ipotesi di uguale varianza (omoscedasticità). Ecco che quando la distribuzione si presenta, come mostrato sotto, casuale e omogenea e con la presenza di una linea orizzontale, significa che il principio di omoscedasticità è rispettato;

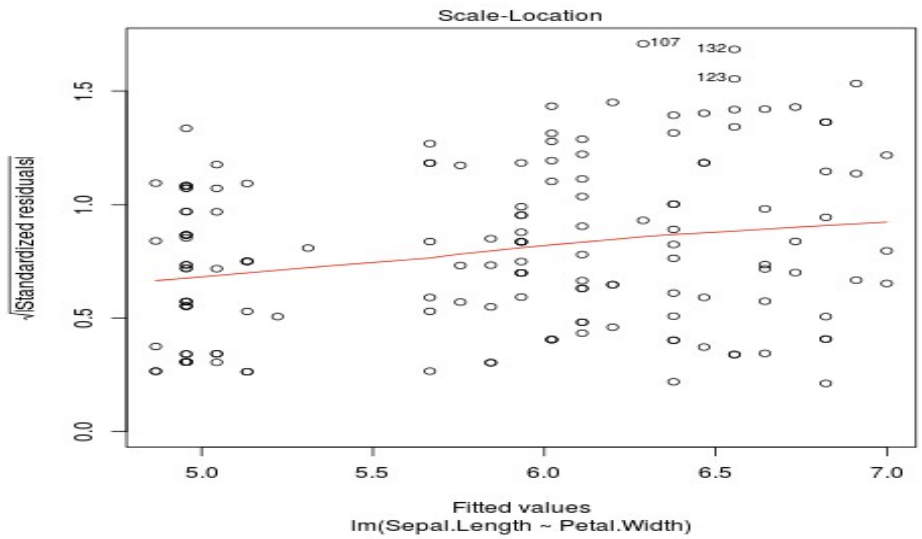


Figura 4: Scale-Location

Q-Q Normal: rappresentazione grafica dei quantili di una distribuzione; confronta la distribuzione cumulata della variabile osservata (residui) con la distribuzione cumulata della normale (distribuzione teorica). Il fatto che questi valori si mostrino tutti abbastanza vicini alla diagonale, tranne per alcuni che agli estremi, è una buona cosa, rappresentando una distribuzione dei dati molto vicino alla normale. Si possono inoltre notare alcuni valori particolari: 107, 123 e 132; questi sono punti il cui peso influisce parecchio sul calcolo della regressione lineare. Non è questo il caso, ma quando tali punti si presentano eccessivamente lontani vanno eliminati.

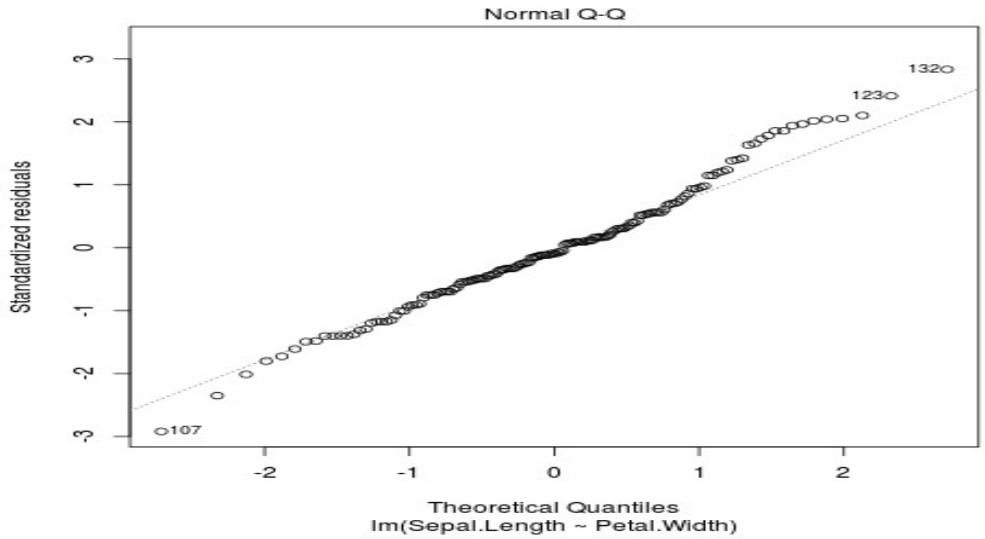


Figura 5: Q-Q Normal

Residual vs Fitted*:* Questo diagramma mostra la presenza o assenza di relazioni lineari fra i predittori ed il predetto. Quando come nel caso in esame, si evidenziano residui sparsi su una linea orizzontale, senza schemi distinti, questa è una buona indicazione che le relazioni sono tutte lineari

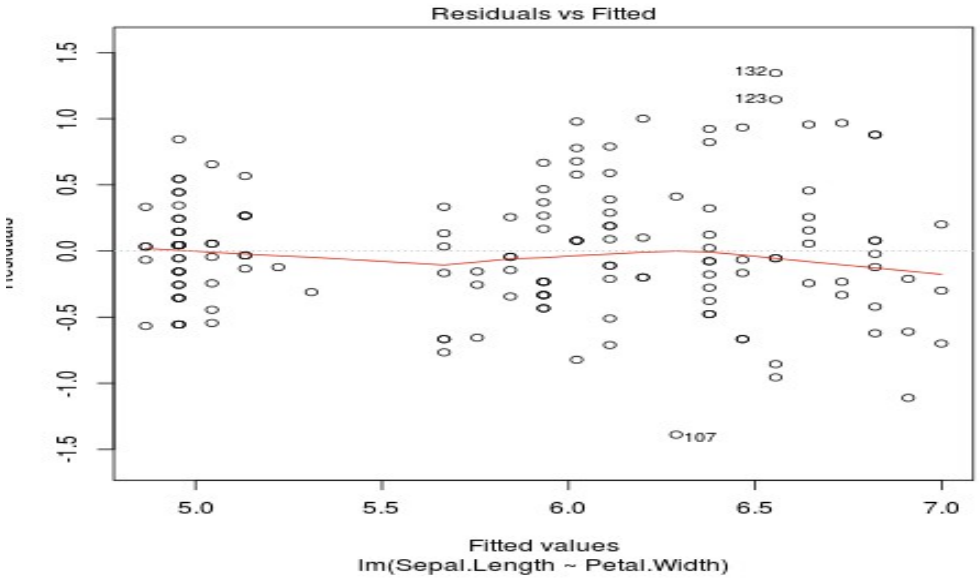


Figura 6: Residual-Fitted

Partendo adesso da quello iniziale facciamo adesso un po' di modelli, in modo da comprendere l'importanza dei vari predittori caso per caso.

```
modello1 <- lm(Sepal.Length ~ Petal.Length+Petal.Width, data=iris)
 summary(modello1)
```

```
> summary(modello1)

Call:
lm(formula = Sepal.Length ~ Petal.Length + Petal.Width, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.18534 -0.29838 -0.02763  0.28925  1.02320

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.19058    0.09705  43.181  < 2e-16 ***
Petal.Length  0.54178    0.06928   7.820 9.41e-13 ***
Petal.Width  -0.31955    0.16045  -1.992   0.0483 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4031 on 147 degrees of freedom
Multiple R-squared:  0.7663,    Adjusted R-squared:  0.7631
F-statistic:   241 on 2 and 147 DF,  p-value: < 2.2e-16
```

Figura 7: Informazioni modello1

Il fatto che ho aggiunto *Petal.Length* rispetto al modello precedente non ha pertanto cambiamenti importanti, ecco che esso non è un predittore molto significativo.

```
modello2<- lm(Sepal.Length ~ Sepal.Width+Petal.Width, data=iris)
 summary(modello2)
```

```
> summary(modello2)

Call:
lm(formula = Sepal.Length ~ Sepal.Width + Petal.Width, data = iris)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2076 -0.2288 -0.0450  0.2266  1.1810

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.45733    0.30919   11.18  < 2e-16 ***
Sepal.Width  0.39907    0.09111    4.38 2.24e-05 ***
Petal.Width  0.97213    0.05210   18.66  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4511 on 147 degrees of freedom
Multiple R-squared:  0.7072,    Adjusted R-squared:  0.7033
F-statistic: 177.6 on 2 and 147 DF,  p-value: < 2.2e-16
```

Figura 8: Informazioni modello 2

In realtà per selezionare il modello maggiormente informativo non si applica la tecnica del *best subsets selection*, dove quindi si fa una ricerca esaustiva fra tutti i modelli combinandone le varie esplicative assegnate (sarebbe un metodo troppo dispendioso); ma vi sono tre strategie differenti molto più idonee fra cui scegliere:

1. Forward selection: si parte inserendo nel modello la covariata che presenti la maggiore correlazione significativa (test t) e stabilendo un livello di significatività. A questo punto si inseriscono man mano i predittori successivi selezionandoli fra quelli che presentino un coefficiente di correlazione parziale che sia il più elevato e significativo. Il procedimento termina quando si individua un coefficiente che non rientra nel livello di significatività precedentemente stabilito; il modello definitivo è quello ottenuto al penultimo passo.

2. Backward selection: si parte dal modello che include tutte le variabili, e come sopra si fissa poi un livello di significatività. A ad ogni passo vanno tolte le variabili col coefficiente di regressione meno significativo in base al test t; inoltre le stime dei coefficienti delle variabili rimaste dovranno essere ricalcolate di volta in volta. Si ripeterà tale procedimento sino a quando le covariate non risultino tutte significative rispetto al livello prefissato.

3. Stepwise regression: mix dei due criteri precedenti. Prima di tutto aggiungo le variabili seguendo il metodo forward selection. A un certo punto aggiungendo una nuova variabile, i coefficienti di regressione delle variabili già incluse potrebbero risultare singolarmente non significativi a causa della forte correlazione con essa. Pertanto dopo l'inserimento di ciascuna variabile il modello viene riconsiderato attraverso il backword selection dove si controlla se vi è qualche variabile da eliminare.

Se proviamo ad usare il secondo metodo (partendo quindi da un modello contenente tutte le esplicative, ne risulta il seguente autput:

```
> bk<-lm(Sepal.Length ~ Petal.Length+Petal.Width+Sepal.Width, data=iris) ##backword selection
> summary(bk)

Call:
lm(formula = Sepal.Length ~ Petal.Length + Petal.Width + Sepal.Width,
    data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-0.82816 -0.21989  0.01875  0.19709  0.84570

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.85600    0.25078   7.401 9.85e-12 ***
Petal.Length  0.70913    0.05672  12.502  < 2e-16 ***
Petal.Width  -0.55648    0.12755  -4.363 2.41e-05 ***
Sepal.Width   0.65084    0.06665   9.765  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3145 on 146 degrees of freedom
Multiple R-squared:  0.8586,    Adjusted R-squared:  0.8557
F-statistic: 295.5 on 3 and 146 DF,  p-value: < 2.2e-16
```

Figura 9: Modello con tutte le covariate

Si può notare come non sia necessario fare altri step in quanto non vi sono variabili inutili a fini informativi. Facendo inoltre un controllo incrociato anche con le altre combinazioni di modelli questo risulta infatti quello con la maggiore quantità di varianza spiegata; ne dettaglio si ha che:

PREDITTORI	OSSERVAZIONI
Petal.Length+Petal.Width+Sepal.Length	• Tutti i predittori sono significativi • R2=0,859
Sepal.Width+Petal.Width	• Entrambi i predittori sono significativi • R2=0,707
Petal.Length+Petal.Width	• Petal.Length risulta non significativo • R2=0,767
Petal.Length+Sepal.Width	• Entrambi i predittori sono significativi • R2=0,840

Possiamo quindi concludere che:

Sepal.Length ~ Petal.Length+Petal.Width+Sepal.Width

Questo sebbene potesse essere un risultato anche abbastanza prevedibile non essendoci un numero eccessivo di variabili dipendenti, mostra l'utilità di saper individuare il contributo informativo che ciascuna variabile porta al modello, soprattutto se ci si trova a trattare con un numero di dimensioni superiori.

A seguire se ne include la rappresentazione grafica. Essendo un modello di *regressione lineare multivariato* (k predittori > 1), abbiamo che la relazione tra un dato regressore e la variabile che si vuole prevedere (Sepal.Width), può venire influenzata dai restanti regressori; ecco che le varie relazioni vengono presentate singolarmente al netto dell'influenza degli altri repressori del modello.

```
bk<-lm(Sepal.Length ~ Petal.Length+Petal.Width+Sepal.Width, data=iris) ##modello
library(car) ##oackage necessario
ceresPlots(bk) ##grafico multivariato
```

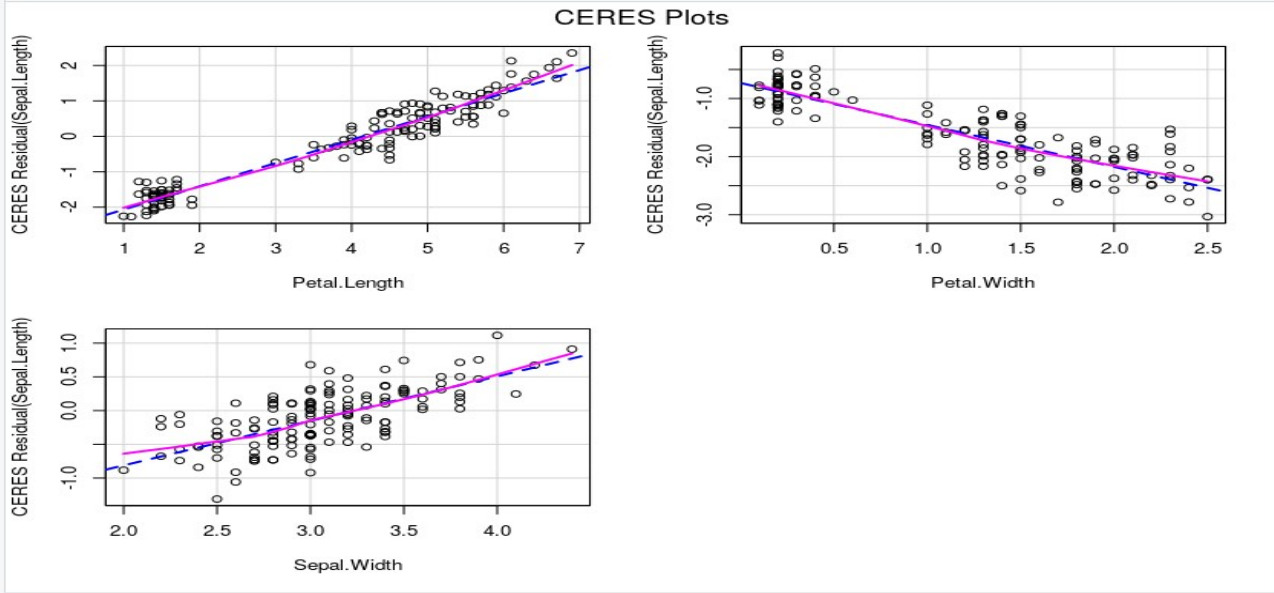


Figura 10: Grafici Regressione lineare multipla dei singoli regressori

Come si può notare la qualità della previsione (linea trattegiata) è molto buona rispetto alla retta di regressione dei dati reali (linea continua), cosa che ci aspettavamo vista la devianza spiegata coperta per più di un 80%.

Un altro elemento di cui è importante comprendere il funzionamento è il metodo *ANOVA;* esso consiste *nell'analisi della varianza*, e permette di individuare le differenze presenti tra più gruppi di dati confrontando la loro variabilità interna con la variabilità tra i gruppi. L'applicazione di tale metodologia richiede preventivamente il soddisfacimento di due proprietà:

- normalità (distribuzione gaussiana dei dati);
- omoschedasticità

Il nostro set di dati presenta entrambe queste caratteristiche quindi applichiamola; R presenta la funzionalità anova(), essa permette di confrontare fra loro vari modelli permettendoci di capire se le variabili presenti in più o in meno di un modello rispetto all'altro, apportano effettivamente un contributo significativo nello spiegare la variabile risposta (il tutto viene verificato tramite il test F e a quale è la probabilità che h0 sia vera).

Creiamo quindi i nostri due modelli su cui poi vorremmo applicare anova() per comprendere il peso informativo di ciascun predittore:

```
modello7 <- lm(Sepal.Length ~ Petal.Length, data=iris)
modello1 <- lm(Sepal.Length ~ Petal.Length + Petal.Width , data=iris)
anova(modello7, modello1)
```

L'output che ci viene fornito è il seguente:

```
> anova(modello7, modello1)
Analysis of Variance Table

Model 1: Sepal.Length ~ Petal.Length
Model 2: Sepal.Length ~ Petal.Length + Petal.Width
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1    148 24.525
2    147 23.881  1   0.64434 3.9663 0.04827 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figura 11: Risultati anova - confronto fra modelli

n sostanza significa che *Petal.Width* rispetto al modello che non la include, non è una variabile che porta moltissima informazione.

Attenzione: comunque un po' di informazione la variabile *Petal.Width la* porta (altrimenti non sarebbe indicato neppure lo *).
A riprova di ciò possiamo notare che se ci facciamo aiutare anche dal valore dell'*Akaike Information Criterion* esso non ci dice che il modello1 è inutile anzi, essendo quello con l'AIC minore vine selezionato

come migliore rispetto al modello7; semplicemente pone l'accento sulla necessità di attribuire ai numeri la giusta interpretazione in base al contesto in cui sono calati. Ecco che andando anche a guardare bene il peso che ciascuna variabile possiede nel modello, si comprende che *Petal.Length* e *Petal.Width* non sono probabilmente la combinazione migliore da attuare.

```
x <- c(AIC(modello7), AIC(modello1))
 delta <- x - min(x)  //il modello migliore risulta quello con AIC  più piccolo
 delta
```

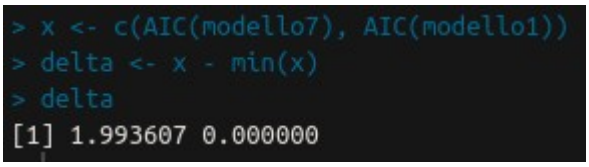


Figura 12: Valore AIC

Confutiamo quanto esposto andando ad osservare i dati per ciascun singolo modello.

```
> summary(modello7)

Call:
lm(formula = Sepal.Length ~ Petal.Length, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.24675 -0.29657 -0.01515  0.27676  1.00269

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.30660    0.07839   54.94   <2e-16 ***
Petal.Length  0.40892    0.01889   21.65   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4071 on 148 degrees of freedom
Multiple R-squared:   0.76,     Adjusted R-squared:  0.7583
F-statistic: 468.6 on 1 and 148 DF,  p-value: < 2.2e-16
```

```
> summary(modello1)

Call:
lm(formula = Sepal.Length ~ Petal.Length + Petal.Width, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.18534 -0.29838 -0.02763  0.28925  1.02320

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.19058    0.09705  43.181  < 2e-16 ***
Petal.Length  0.54178    0.06928   7.820 9.41e-13 ***
Petal.Width  -0.31955    0.16045  -1.992   0.0483 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4031 on 147 degrees of freedom
Multiple R-squared:  0.7663,    Adjusted R-squared:  0.7631
F-statistic:   241 on 2 and 147 DF,  p-value: < 2.2e-16
```

Figura 13: Confronto fra i due modelli

Che non fa altro che confutare quanto già affermato da anova.

Al netto di ciò e riprendendo il modello di regressione ottimo individuato in precedenza, si evince che:

La variabile *Petal.With* acquisisce la sua importanza massima quando si trova in combinazione con con *Sepal.Width*, perdendone in assenza.

```
> anova(modello1, modello7, bk)
Analysis of Variance Table

Model 1: Sepal.Length ~ Petal.Length + Petal.Width
Model 2: Sepal.Length ~ Petal.Length
Model 3: Sepal.Length ~ Petal.Length + Petal.Width + Sepal.Width
  Res.Df    RSS Df Sum of Sq       F  Pr(>F)
1    147 23.881
2    148 24.525 -1   -0.6443  6.5124 0.01174 *
3    146 14.445  2   10.0796 50.9375 < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 14: I tre modelli a confronto

Poniamoci quindi una domanda:

Cosa accade se inseriamo nel modello anche valori discreti?

Nota: *Orange Canvas* infatti non lo lascia fare

Aggiungendoli al nostro modello, come mostrato a seguire, notiamo che sorprendentemente portano comunque un loro contributo al modello, andando anche ad influire sulla variabile *Petal.Width* che perde di significatività (passa da *** a *), e facendo R^2 anche se non mi moltissimo.

```
Call:
lm(formula = Sepal.Length ~ Petal.Length + Petal.Width + Sepal.Width +
    Species, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-0.79424 -0.21874  0.00899  0.20255  0.73103

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        2.17127    0.27979   7.760 1.43e-12 ***
Petal.Length       0.82924    0.06853  12.101  < 2e-16 ***
Petal.Width       -0.31516    0.15120  -2.084  0.03889 *
Sepal.Width        0.49589    0.08607   5.761 4.87e-08 ***
Speciesversicolor -0.72356    0.24017  -3.013  0.00306 **
Speciesvirginica  -1.02350    0.33373  -3.067  0.00258 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3068 on 144 degrees of freedom
Multiple R-squared:  0.8673,    Adjusted R-squared:  0.8627
F-statistic: 188.3 on 5 and 144 DF,  p-value: < 2.2e-16
```

Figura 14: Modello inclusivo delle specie

È chiaro quindi come anche tali variabili abbiano il loro peso all’interno del nostro modello, e come la loro inclusione o meno possa andare ad influire sui legami anche con le altre covariate.

Mostriamo ora come si distibuiscono i valori osservati differenziandoli per specie di appartenenza. Si include sia il codice impiegato che il risultato restituito da R.

Nell'asse y viene indicata sempre la nostra variabile dipendente, mentre la x cambia in base alla variabile selezionata:

```
library(RGraphics)
 library(grid)
 library(gridExtra)
 library(ggplot2)
 ##grafici carini in base alle specie
 g1<-ggplot(iris,aes(x=Sepal.Width,y=Sepal.Length, shape=Species, color=Species))+
       geom_point(size=2.5)
 g2<-ggplot(iris,aes(x=Petal.Width,y=Sepal.Length, shape=Species, color=Species))+
       geom_point(size=2.5)
 g3<-ggplot(iris,aes(x=Petal.Length,y=Sepal.Length, shape=Species, color=Species))+
       geom_point(size=2.5)

 grid.arrange(g1,g2,g3, nrow=2, ncol=2,  top = "Species Distributions")
```

I risultati non ci stupiscono in quanto corrispondono a quelli che avevamo già ottenuto con *Orange Canvas* ad una prima analisi.

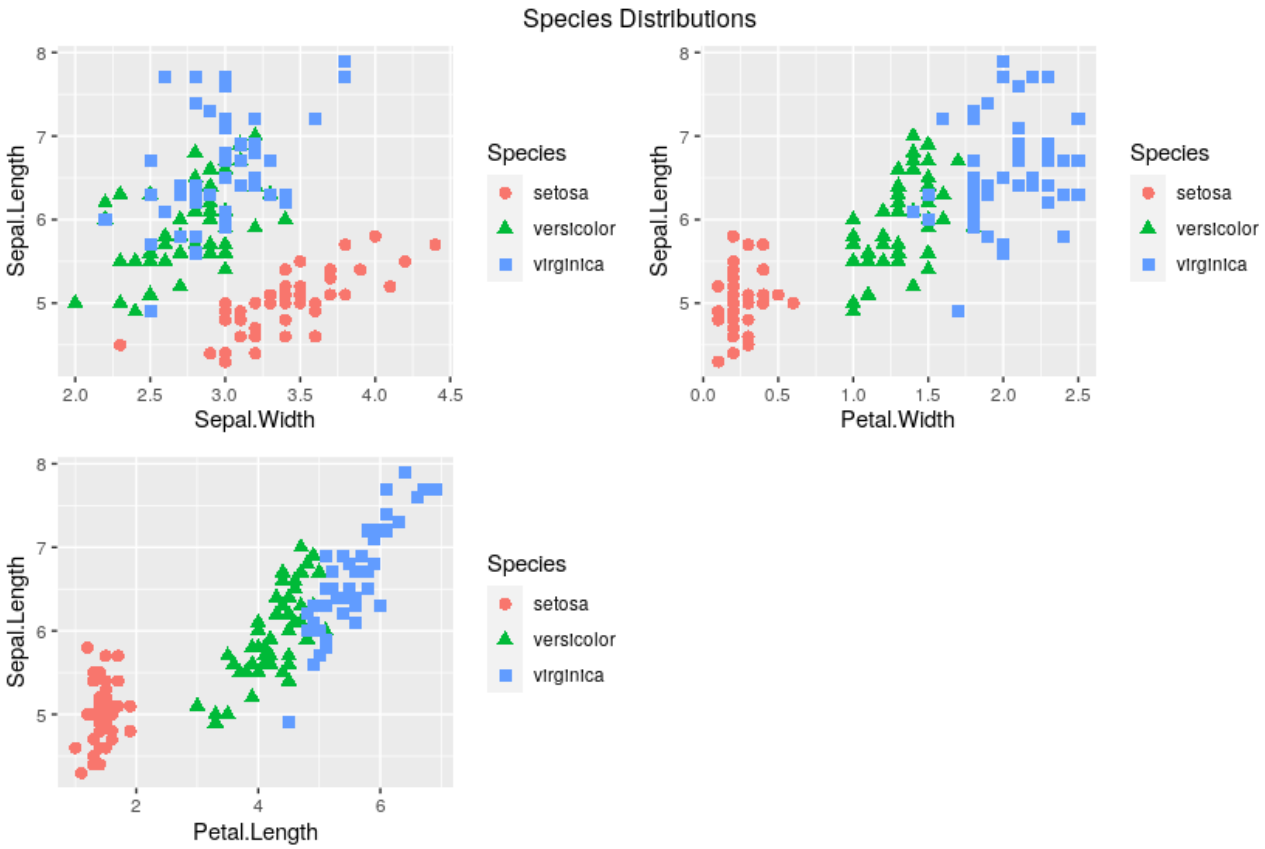


Figura 15: Distribuzione delle specie

Dataset Cars

Prendiamo ora in esame un altro dataset: *Cars.*

*N*on essendo mai stato esaminato prima (neppure con *Orange Canvas*) se ne fa anche una breve introduzione iniziale in modo da avere maggiormente chiara la situazione che si va ad esaminare. I dati cars s riguardano

la distanza percorsa da un auto, che viaggia ad una certa velocità prima di fermarsi. La distanza è espressa in piedi, la velocità in miglia orarie. Sono 50 osservazioni che risalgono agli anni venti.

```
> summary(cars)
     speed           dist       
 Min.   : 4.0   Min.   :  2.00  
 1st Qu.:12.0   1st Qu.: 26.00  
 Median :15.0   Median : 36.00  
 Mean   :15.4   Mean   : 42.98  
 3rd Qu.:19.0   3rd Qu.: 56.00  
 Max.   :25.0   Max.   :120.00  
> dim(cars)
[1] 50  2
```

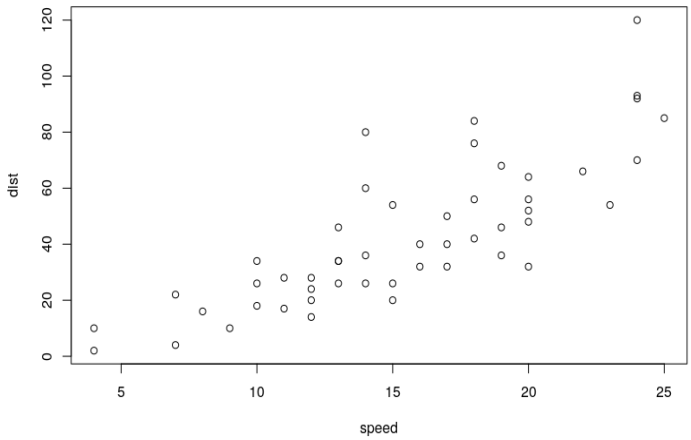


Figura 15: Informazioni generali Cars

Si nota subito che abbiamo un chiaro caso di eteroschedasticità (forma a trianngolo dei dati sul grafico alla destra), risulta quindi questo un buon caso studio per provare ad applicare le trasformazioni degli algoritmo Box- Cox.

Ecco quindi il nostro modello:

$$dist = speed + \xi$$

Ci interessa predire quindi la *distanza* avendo come unico regressore la variabile *velocità*; si tratta di un modello di regressione lineare semplice.

```
modello <- lm(dist ~ speed, data=cars)
summary(modello)
plot(modello)
```

Nella pagina a seguire osserviamo i dati che ci fornisce R sul modello prescelto, e alcuni plot sulla distribuzione delle osservazioni.

Quello che si evince è che:

- R^2 risulta un valore abbastanza buono allocandosi su un $\simeq$65%;
- *speed* si mostra come un predittore molto significativo (essendo anche l'unico).

Questo ci permette di rifiutare l'ipotesi che non vi sia nessuna dipendenza fra il predittore ed il predetto (ossia l'ipotesi nulla).

```
> modello <- lm(dist ~ speed, data=cars)
> summary(modello)

Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-29.069  -9.525  -2.272   9.215  43.201 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -17.5791     6.7584  -2.601   0.0123 *  
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438 
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Figura 16: Dati del modello di Regressione lineare semplice

Tuttavia visto anche il grafico incontrato prima è alquanto improbabile che tale relazione sia lineare, rendendo in tal modo non abbastanza qualitativo il modello di regressione applicato così com'è.

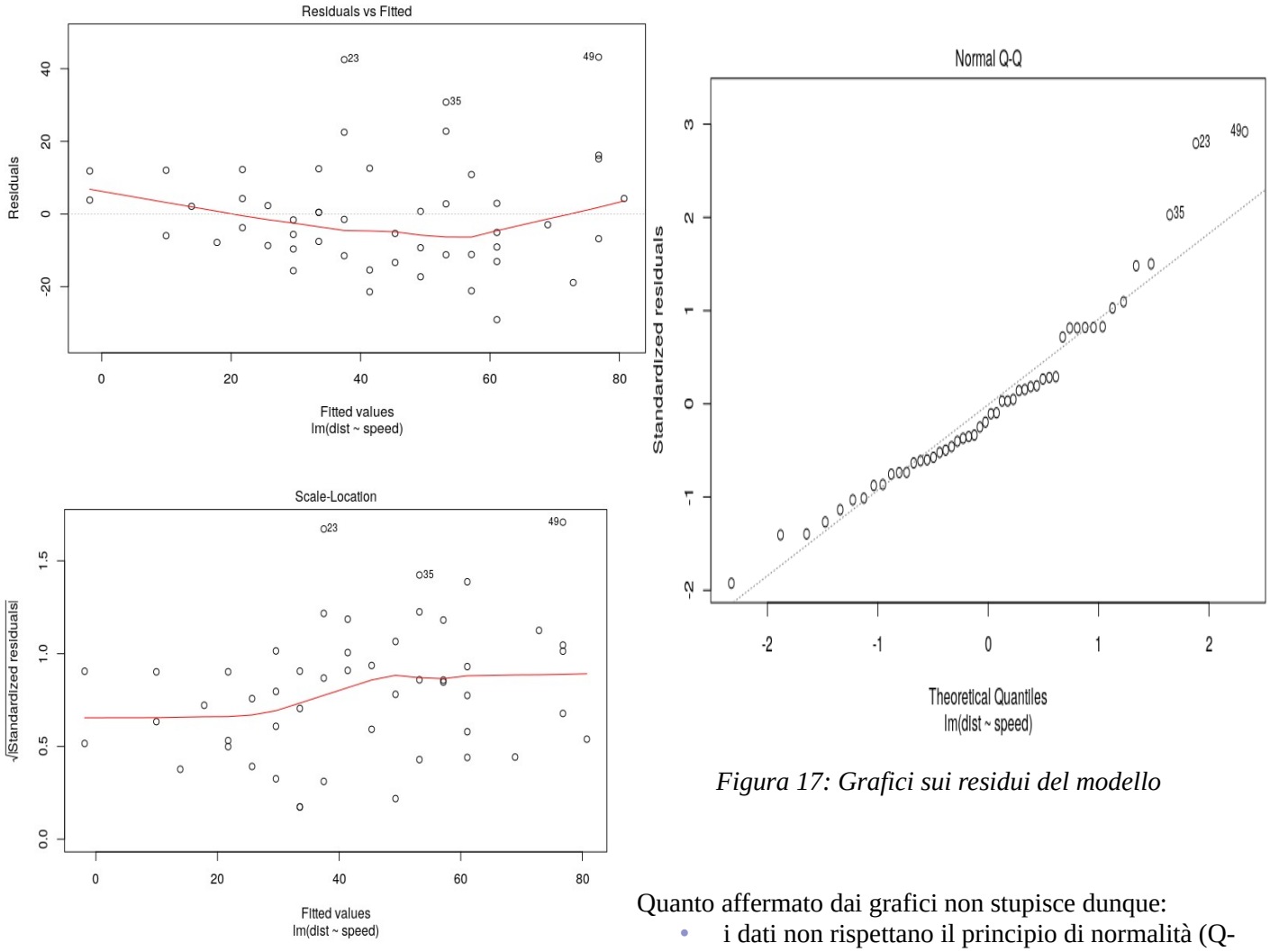


Figura 17: Grafici sui residui del modello

Quanto affermato dai grafici non stupisce dunque:

- i dati non rispettano il principio di normalità (Q-Q plot);
- la distribuzione non risulta uniforme, segno evidente che i legami non sono pienamente lineari (Residual Fitted);
- vine meno il principio di omoschedasticità (Scale Location);

- vi sono alcuni valori anomali molto distanti che si ripetono: come il 23, 35 e 49. Tuttavia facendo un confronto sempre con l'ausilio di R, con le distanze di Cook essi sembrano rientrarvi, probabilmente quindi sono valori anomali ma la loro considerazione o meno non dovrebbe portare a modifiche sostanziali del modello in questione.

Dopo tali preamboli applichiamo quindi il metodo *Box-Cox*, trasformiamo dunque la variabile dipendente e rendendola più idonea ad un modello lineare.

Esso si basa sull'individuazione di un valore λ grazie a qui il modello diventa nella forma:

$$\frac{dist^{\lambda} - 1}{\lambda} = speed + \xi$$

I dati vengono quindi trasformati impiegando le proprietà dell'elevazione a potenza, potendo così lavorare su dati più normali e con una distribuzione più stabile.

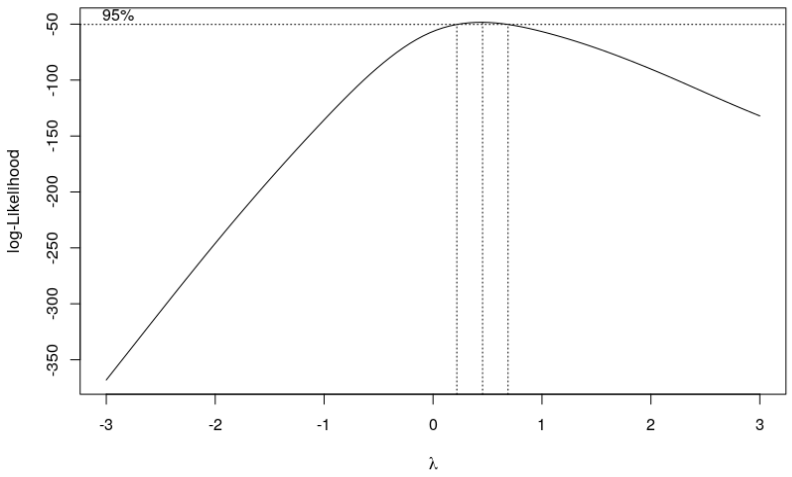
La formula generale è

$$y_{tras} = \frac{y^{\lambda} - 1}{\lambda} \text{ per } \lambda \neq 0$$

$$y_{tras} = \log(y) \text{ per } \lambda = 0$$

In generale λ può assumere qualunque valore da -3 a +3. È dunque la semplice elevazione a potenza del predetto a permettere la trasformazione del modello; da notare che per valori <-2 e >2 ha poco senso applicare Box-Cox.

```
bc <- boxcox(modello, lambda = seq(-3, 3)) ##trasformata, mi da' lambda
lambda <- bc$x[which(bc$y==max(bc$y))] ##0.4545
summary(lambda)
```




```
> lambda <- bc$x[which(bc$y==max(bc$y))] ##0.4545
> summary(lambda)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4545  0.4545  0.4545  0.4545  0.4545  0.4545
```

Figura 18: Funzione λ

```
trasf <- (((cars$dist)^lambda)-1)/lambda ##trasformazione dalla variabile dipendente
model.inv <- lm(trasf ~ speed, data=cars) ##nuovo modello
```

```
> trasf <- (((cars$dist)^lambda)-1)/lambda
>
> model.inv <- lm(trasf ~ speed, data=cars)
> summary(model.inv)

Call:
lm(formula = trasf ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4737 -1.1661 -0.3283  0.9386  5.3205

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.89905    0.82243   1.093     0.28
speed        0.55029    0.05056  10.883 1.48e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.872 on 48 degrees of freedom
Multiple R-squared:  0.7116,    Adjusted R-squared:  0.7056
F-statistic: 118.4 on 1 and 48 DF,  p-value: 1.475e-14
```

Figura 19: Valutazione del modello trasformato

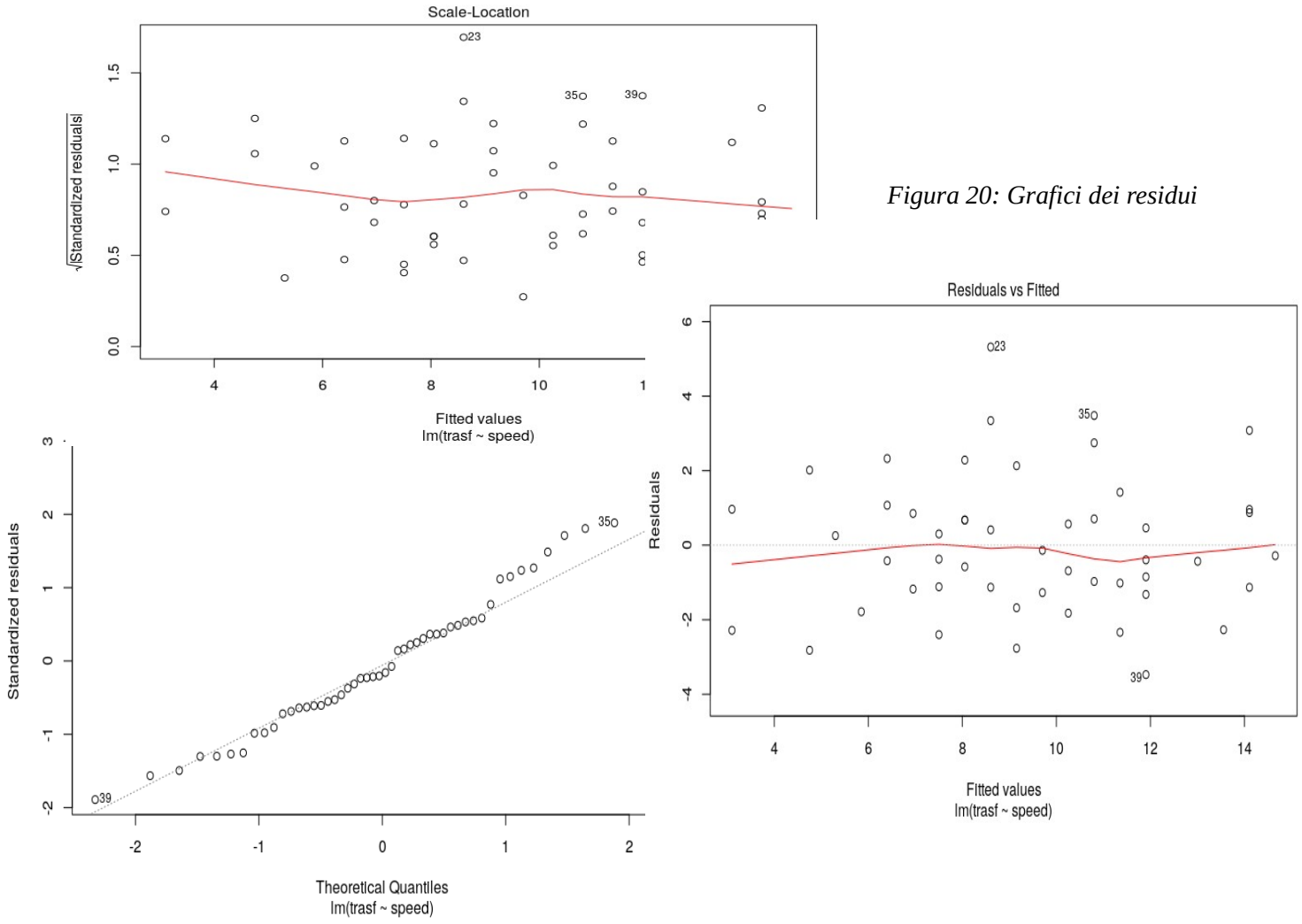


Figura 20: Grafici dei residui

Possiamo a questo punto concludere che applicando tale semplice trasformazione alla nostra y abbiamo ottenuto dei miglioramenti non indifferenti in quanto:

- la devianza spiegata dal modello è cresciuta di un 6%;
- le osservazioni presentano una maggiore linearità, omoschedasticità e normalità.