ALCUNE FUNZIONI STATISTICHE IN "R"

BY BB STUDENT

FUNZIONI UTILI IN R

LEGGERE E/O APRIRE FILE

I file che possono essere manipolati in R devono essere salvati principalmente in formato .txt o .csv. altro punto importante è il "luogo" (directory) in cui sono salvati, per facilità è bene salvare il file nella stessa cartella in cui R salva i suoi documenti. Caratteristica interessante di R è che facilmente si possono utilizzare anche file depositati nel "web" collegandosi mediante l'indirizzo.

>Esempio = read.csv("file.csv", header = TRUE) # al posto di file.csv va inserita o il nome del file se salvato nel pc, o l'url se si tratta di file su web.

>Esempio = read.table("file.txt", header = FALSE) #simile a prima con la differenza che l'estensione è .txt.

Dopo aver letto e attribuito il file alla variabile "Esempio", sarà bene caricarli su R per poterli manipolare magari facendoci dei grafici o dei test statistici. Per questo si usa la funzione:

>attach(Esempio)

Un punto a favore di R, in particolare mediante l'IDLE "R-STUDIO", è la funzione "help":

>help(funzione)

Oppure alla rapida interfaccia "grillo parlante" che compare mentre digitiamo la funzione, se non comparisse con il tasto TAB la forziamo ad uscire dalla tana.

Quando abbiamo importato I dati possiamo fare alcuni facili operazione con la tabella, per esempio usando la funzione:

>head(X)

Possiamo avere in output la "testa" della tabella con indici e I primi 6 dati

Viceversa usando la funzione:

>tail(X)

Si ottengono in output gli ultimi sei valori della tabella o dataframe su cui si sta lavorando

Usando invece la funzione:

>str(X)

Possiamo avere in output le caratteristiche del dataframe, gli indici di colonna con accanto la classe a cui appartengono, eventuali fattori e livelli, e/o alcuni valori di ciascuno.

Usando la funzione:

>length(X)

Si ha in output la lunghezza dell'oggetto passato, se passiamo un dataframe ci darà in output il numero delle colonne, se passiamo un array il numero dei valori che contiene e così via.

È da ricordare che tutti i parametri delle funzioni possono essere cambiati,(per esempio se voglio avere il numero di righe invece che di colonne), utilizzando l'help della funzione o la rapida interfaccia.

FUNZIONI SULLA DISTRIBUZIONE NORMALE

DNORM(X)

Dando un valore alla funzione questa restituisce il valore dell'altezza della distribuzione in quel punto indicato al posto di X. Se si inserisce solo il valore, R considera la normale standard, eventualmente si può modificarla andando a selezionare i parametri, in quanto la funzione in forma estesa si presenta in questo modo:

```
> dnorm(x,mean=N,sd=N)
```

PNORM(X)

Questa funzione calcola la probabilità che un numero casuale distribuito normalmente sarà inferiore a quel numero inserito al posto di X. Similmente al caso precedente anche qui se si inserisce solo il valore X il programma elabora il calcolo sulla normale standard.

```
> pnorm(x, mean=N, sd=N)
```

QNORM(X)

Questa funzione è l'inverso di pnorm, praticamente data una **probabilità** restituisce il numero che rappresenta nella funzione quella probabilità, cioè l'area sottesa dalla curva. Anche in questo caso si può considerare di cambiare i parametri della curva.

```
>qnorm(x, mean = N, sd = N)
```

RNORM(X)

Questa funzione genera un numero X, di valori presi da una distribuzione normale standard.

```
>rnorm(x, mean=N, sd=N)
```

Da ricordare che in tutti e tre i casi è possibile modificare i parametri "mean" e "sd" in modo da ottenere la normale desiderata.

FUNZIONI SULLA DISTRIBUZIONE BINOMIALE

DBINOM (X,N,P)

Questa funzione rappresenta la probabilità che l'evento X si verifichi nelle ripetizioni N; mentre P rappresenza la probabilità che si verifiche l'evento X in modo indipentende

> dbinom(X, N, P)

DNBINOM (X,Y,P)

Questa funzione rappresenta la probabilità che l'evento X accada prima di Y, considerando P la probabilità che accada l'evento X.

> dnbinom(X,Y,P)

QBINOM(P^{TH},N,P)

Questa funzione restituisce in output il quantile corrispondente al percentile inserito. Utile quando voglio sapere a quale **quantile** corrisponda l'n-esimo **percentile**.

>qbinom(p^{th},n,p)

RBINOM(NB, N, P)

Questa funzione genera un numero nb, di valori presi da una distribuzione binomiale dopo aver passato i valori di n e di p.

>rbinom (nb, n, p)

TEST STATISTICI

TEST DI SHAPIRO-WILK

Test statistico che si basa sulla verifica della normalità dei dati.

```
> shapiro.test(areainfl[il1b == “etero”])#vanno inseriti il vettore e gli elementi di un secondo vettore contro cui si voglia lanciare il controllo del test
```

T.TEST (x,mu) AD UN UNICO CAMPIONE

Il prerequisito per attuare il t-test è che i dati siano normali.
Il t-test è un test statistico utilizzato principalmente per analizzare se il valore medio di una distribuzione si avvicina ad un valore di riferimento

Si vuole verificare se la media campionaria è significativamente simile (si ponga il livello di significatività pari al 5%) alla media della popolazione, supponendo che la varianza della popolazione non sia nota.

65, 78, 88, 55, 48, 95, 66, 57, 79, 81

```
> a = c(65, 78, 88, 55, 48, 95, 66, 57, 79, 81)
>
> t.test (a, mu=75)
```

La funzione t.test a un solo campione (one sample in inglese) ci fornisce in output il valore di t calcolato; ci dà inoltre i gradi di libertà, l’intervallo di confidenza e la media.

T.TEST(a,b) PER DUE CAMPIONI

Il test T per campioni indipendenti consente di confrontare le medie relative a due gruppi di casi. Quello che si vuole osservare è quindi se le medie dei due gruppi sono significativamente diverse.

Si chiede di confrontare le medie delle altezze di due gruppi, estratti da due popolazioni a varianza non nota. I dati vengono riportati qui di seguito (i valori sono completamente inventati).

A: 175, 168, 168, 190, 156, 181, 182, 175, 174, 179

B: 185, 169, 173, 173, 188, 186, 175, 174, 179, 180

Richiamiamo quindi la funzione t.test per varianze omogenee (var.equal=TRUE)e campioni indipendenti (paired=FALSE, che si può anche omettere perché di default la funzione lavora su campioni indipendenti) in questo modo:

```
> t.test(a,b, var.equal=TRUE, paired=FALSE)
```

IMPLEMENTAZIONE UTILE DELLA FUNZIONE T.TEST

```
> t.test(a,b, paired=TRUE, alt="less")
```

con questa sintassi abbiamo chiesto a R di verificare se la media dei valori contenuti nel vettore *a* è minore (less, in inglese) della media dei valori contenuti nel vettore b.
Se avessimo scritto: t.test(a,b, paired=TRUE, alt=”greater”), avremmo chiesto a R di verificare se la media dei valori contenuti nel vettore a è maggiore della media dei valori contenuti nel vettore b.

T.TEST(A~B)
Questo è sempre un t-test anche se cambia l'input in quanto compare il simbolo TILDE. In questo caso la variabile "B" dovrà essere un fattore a 2 livelli e non di più in quanto il t-test accetta solo du e gruppi di confronto. La variabile "A" sarà invece una variabile che è rappresentata da un vettore numerico associato a "B".
Ad esempio:

	gender	il1b	smoke	areainfl
1	F	etero	low	39.970
2	M	wt	low	24.011
3	F	etero	low	35.774
4	M	etero	high	58.651
5	M	etero	low	27.712
6	M	etero	high	48.362
7	M	mut	low	44.970
8	F	wt	high	56.094
9	M	wt	high	68.816
10	F	etero	low	22.083
11	F	wt	low	62.455
12	F	mut	high	55.125
13	F	wt	high	67.383

Se diamo in input la funzione:
>t.test(areainfl~gender)

Praticamente andiamo confrontare il gruppo di valori della colonna "areainfl" che sono dei maschi, contro quelli delle femmine.
In questo modo effettuiamo un comune t-test contro due gruppi per valutare quanto siano diverse le media.

WILCOX.TEST(a,b)
Il **test di Wilcoxon** si applica nel caso in cui si chiede di confrontare le medie dei valori di due gruppi che **non seguono una distribuzione normale**. È l'equivalente del test t per campioni **indipendenti**.
Vediamo come risolvere il problema con R:

```
> a = c(6, 8, 2, 4, 4, 5)
> b = c(7, 10, 4, 3, 5, 6)
> wilcox.test(a,b, correct=FALSE)
```

L'output del comando darà un p-value ed un consuntivo del test W, in questo modo è possibile prendere una decisione in base all'esperimento.
W = 22, p-value = 0.5174

Caso con dati dipendenti

Ecco di seguito i valori di inquinamento atmosferico:

Con traffico: 214, 159, 169, 202, 103, 119, 200, 109, 132, 142, 194, 104, 219, 119, 234

Senza traffico: 159, 135, 141, 101, 102, 168, 62, 167, 174, 159, 66, 118, 181, 171, 112

Siamo di fronte a dati **appaiati**, cioè sono dati **dipendenti** perché esiste un vincolo tra le rilevazioni, consistente nel fatto che stiamo considerando la stessa città (con le sue peculiarità atmosferiche, ventilazione, etc.) seppure in due differenti giorni. Non potendo supporre una distribuzione gaussiana per i valori rilevati, dobbiamo procedere con un test non parametrico

```
> a = c(214, 159, 169, 202, 103, 119, 200, 109, 132, 142, 194, 104, 219, 119, 234)
> b = c(159, 135, 141, 101, 102, 168, 62, 167, 174, 159, 66, 118, 181, 171, 112)
>
> wilcox.test(a,b, paired=TRUE)
```

TEST DEL CHI-QUADRO—CHISQ.TEST(X,Y)
Questa funzione accetta in input un parametro X che può essere rappresentato da una matrice o un vettore; se è matrice allora Y viene ignorata, mentre se è un vettore allora Y deve essere un vettore della stessa lunghezza di X.
Questa funzione è utile al fine di verificare la corrispondenza fra i due parametri passati o fra elementi della tabella. La funzione da lanciare è:

```
>chisq.test(x, y)
```

Questo test è asintotico e può quindi non dare una stima così affidabile quanto richiesta.

TEST DI FISHER—FISHER.TEST(X,Y)
Questo test verifica la stessa ipotesi del test del Chi-quadro, cioè la corrispondenza fra i due parametri passati o fra elementi della tabella.
La funzione in questo caso è:

```
>fisher.test(x,y)
```

Questo test è da **preferire** rispetto al precedente in quanto è un test **esatto,** dà quindi in output una stima più affidabile.

Verrà usato l'esempio della dispensa "Anova e confronti multipli con R" per i successivi test;

Tabella 1: il dataset `tooth` ($N = 69$).

gender	il1b	smoke	areainfl
F	etero	low	39.970
M	wt	low	24.011
F	etero	low	35.774
M	etero	high	58.651
M	etero	low	27.712
M	etero	high	48.362
M	mut	low	44.970

TEST DI BARLETT

Test statistico che si basa sulla verifica dell'omoschedasticità dei dati.

> barlett.test(areainlf ~ il1b) #vanno inseriti i due vettori di valori che si intende confrontare

TEST DI SHAPIRO-WILK

Test statistico che si basa sulla verifica della normalità dei dati.

> shapiro.test(areainfl[il1b == "etero"])#vanno inseriti il vettore e gli elementi di un secondo vettore contro cui si voglia lanciare il controllo del test

TEST DI ANOVA

Il **metodo ANOVA è una analisi della varianza** permette di effettuare il confronto tra più gruppi, con metodo parametrico (supponendo cioè che i vari gruppi seguano una distribuzione gaussiana).
Per procedere con la verifica ANOVA, occorre dapprima verificare l'omoschedasticità (ossia effettuare test per l'omogeneità delle varianze) e la normalità dei dati.
Verrà usato l'esempio della dispensa ("Anova e confronti multipli con R");
I testi impiegati in questo caso sono:

- Per l'omoschedasticità si tratta del test di Balrlett.
- Per la normalità si può usare il test di Shapiro-wilk, se non sono normali, Wilcoxon (vedi sopra).

A questo punto se i dati rispondono ai prerequisiti ricercati si può andare a svolgere il test ANOVA:

>Esempio = aov(areainfl ~ il1b)
>summary(Esempio)

FUNZIONI SUI CONFRONTI MULTIPLI

Quando bisogna lavorare con più gruppi è bene tenere in considerazione che l'errore dato dai confronti multipli non è uguale all'errore dato da un singolo confronto, per questa ragione è necessario attuare delle correzioni che non facciano scartare l'ipotesi solo perché ci sono più gruppi.
Ancora una volta bisogna valutare se i dati siano etero o omoschedastici in quanto bisogna agire con comandi in R leggermente diversi.

Omoschedastici ed eteroschedastici

È bene ricordare che dati **omoschedastici** sono dati che, graficamente, sono dispersi in modo abbastanza omogeneo, al di sopra o al di sotto, in rapporto ad una linea retta. Statisticamente sono quei dati che hanno una media prossima alla media teorica costruita su un modello; inoltre la loro varianza è costante.
Dati che non rispondono a queste caratteristiche sono detti **eteroschedastici.**
Per controllare questo parametro ricordo il **test di Barlett** descritto un po' più in su.

MULTCOMP e SANDWICH

"Mulcomp" è una libreria in R che contiene numerose funzioni fra cui "Sandwich".
Quest'ultima funzione è utile nel trattamento di dati eteroschedastici, viceversa se i dati sono omoschedastici non è necessario.
Per importare le librerie basta lanciare il comando:

```
>library(multcomp)
>library(sandwich)
```

A questo punto possiamo lanciare il comando vero e proprio per il confronto multiplo. Questa parte è **copiata** dalla guida da "Anova e confronti multipli con R":

Caso **ETEROSCHEDASTICO**

```
>posthoc = glht(modello, linfct = mcp(il1b = "Tukey"), vcov = sandwich)  #si attribuisce alla variabile "posthoc" il risultato del confronto.
```

Con '*glht*' si effettua un test di ipotesi lineare generale, rispetto ad una funzione lineare di interesse '*linfct*' che si ottiene dalla matrice dei contrasti parametrici '*mcp*' calcolata sul fattore 'il1b' con il metodo di Tukey.

Caso **OMOSCHEDASTICO**

```
>posthoc2 = glht(modello, linfct = mcp(il1b = "Tukey"))#Come si può osservare non è presente la funzione 'sandwich'.
```

Un'ultima parentesi va aperta per la "Correzione di Bonferroni" la quale deve essere applicata quando i dati sono **OMOSCHEDASTICI,** altrimenti va applicato il "test di Tukey", come in "posthoc".

```
>summary(posthoc2, test = adjusted(type = "bonferroni")) #questo comando è analogo al precedente con la sola differenza che l'output dei dati sarà immediata senza fare prima l'attribuzione alla variabile "posthoc2" e poi chiamarla con la funzione summary.
```

Il metodo da prediligere sarà quindi in **test di Tukey,** che permette una migliore stima in entrambi i casi.

MODELLO LINEARE

RETTA DI REGRESSIONE—LM()

Il comando di R per adattare un modello di regressione lineare è lm, che ha il seguente utilizzo:

>modello = lm(formula)

in cui 'formula' specifica l'espressione del modello che si vuole stimare. In genere formula è esprimibile nella forma:
(*variabile risposta ~ variabile esplicativa)*
Il risultato della funzione lm è una lista composta da numerosi elementi che riassumono la stima del modello.

REGRESSIONE MULTIPLA--LM()

Si tratta di una funzione analoga alla precedente con la differenza che in questo caso invece di basarsi solo su una variabile indipendente se ne utilizzano diverse.

>modello= lm(formula)

in cui 'formula' specifica l'espressione del modello che si vuole stimare. In genere formula è esprimibile nella forma:
(*variabile risposta ~ variabili esplicative separate da segni + o *)*
Anche in questo caso l'output della funzione sarà una lista composta da numerosi elementi che riassumono la stima del modello.
Utilizzare i separatori "+" o "*" risulta essere chiaramente diverso:

- "+" indica che la variabile risposta deve essere confrontata con ogni variabile digitata;
- "*" indica che la variabile risposta verrà confrontata con il rapporto che c'è fra le due variabili cioè quella che precede e antecede il segno "*".

DEVIANZA—ANOVA(X,Y)

Misura della dispersione dei quadrati dei residui rispetto alla retta di regressione. Questo test serve per controllare quanto i residui, cioè i dati che rappresentiamo nel piano cartesiano si discostino dalla retta che li approssima. Ovviamente più è alto l'output maggiore sarà la distanza dei dati e quindi è bene porsi qualche domanda.

>anova(modelloA, modelloB)

I modelli che diamo in pasto alla funzione "anova" sono praticamente i risultati ottenuti dalla precedente funzione "lm". La funzione, fra i vari output, ci darà un p-value che indicherà sostanzialmente se i due modelli sono approssimativamente simili o se si tratta di cose radicalmente diverse.

LOGARITMO DELLA VEROSIMIGLIANZA—AIC(X)

Il logaritmo della verosimiglianza è un calcolo statistico mediante il quale è possibile stimare quale dei metodi usati è il più affidabile. Esiste tuttavia una funzione che ne permette il calcolo senza passare per laboriosi passaggi intermedi:

>AIC(modello)

l'AIC può essere interpretato solo in ottica comparativa: il modello migliore è sempre quello con l'AIC più basso. Quindi, quello a cui si deve badare non è il valore dell'AIC in sé, ma la differenza in AIC tra i modelli.

FUNZIONI GRAFICHE (usato il database “iris”)

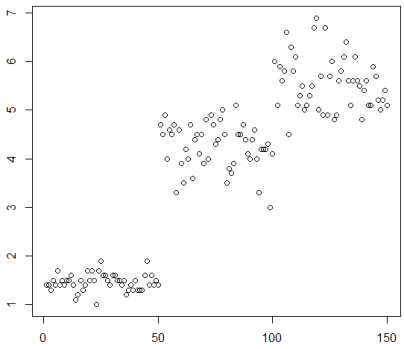
PLOT(X)

Questa funzione restituisce un output visivo in cui si rappresenta un grafico diverso in base all’input X che si dà. Per esempio se viene passato un vettore allora verrà illustrato uno scatterplot dei valori del vettore rispetto a degli indici, se è una matrice sarà una colonna rispetto all’altra. Utile soprattutto per capire in prima battuta come, graficamente, si dispongono i dati, utile prima di fare analisi più specifiche.

>plot(…)

> attach(iris) #utile per poter manipolare il dataframe, tabella, tabulazione di dati ecc.

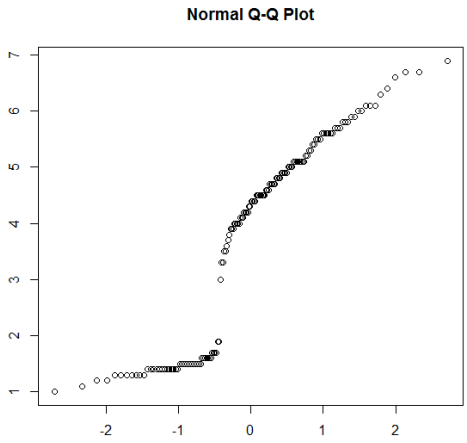
> plot(Petal.Length)



QQNORM(X)

produce un grafico quantile-quantile dei dati del vettore x rispetto ad una corrispondente distribuzione normale

> qqnorm(Petal.Length)



QQLINE(X)

come qqnorm, ma viene aggiunta una linea che passa attr averso il primo e terzo quartile

> qqline(Petal.Length)

QQPLOT(X,Y)

produce il grafico dei quantili dei dati del vettore x risp
etto ai quantili dei dati del vettore y.

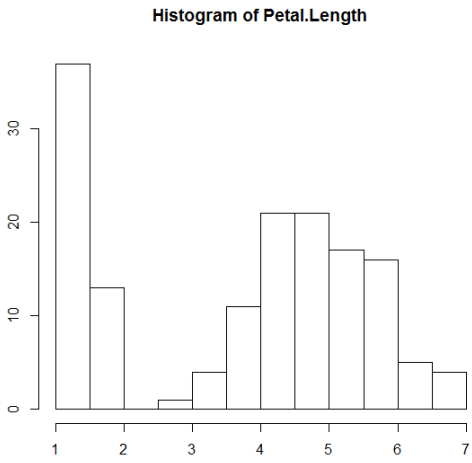
>qqplot(Petal.Length, Petal.Width)

HIST

genera istogrammi utilizzando un vettore numerico.

>hist(Petal.Length) #genera un istogramma utilizzando il vettore

X è per esempio un vettore di valori e grazie alla funzione hist si può, visimente, fare una prima analisi dei dati andando a vedere per esempio se i dati sono approssimabili ad una distribuzione normale.



BOXPLOT(X)

Forniscono una descrizione grafica sintetica di un insieme di dati utilizzando semplici statistiche. Anche in questo caso possiamo passare al programma diversi input, possiamo fare boxplot di vettori, matrici, dataframe, l'importante è capire cosa si sta ricercando e se il boxplot è effetttivamente lo strumento giusto.

L'utilizzo di questa funzione genera un output in cui vengono illustrati uno o più boxplot, utili a comprendere visivamente come siano distribuiti i dati, e alcuni dati statistici interessanti come la media o eventuali outlier.Usando un database di dati già caricato in R, "iris" possiamo ottenere diversi boxplot rappresentati i principali parametri in esso contenuti:

>boxplot ("iris")