Laboratorio di Statistica con R

R è un vero e proprio linguaggio di programmazione. Il suo nome, è dovuto probabilmente al nome dei suoi sviluppatori:Robert Gentleman e Ross Ihaka

Le principali funzioni di R possono essere così riassunte:

- esecuzione di calcoli
- analisi statistica ed elaborazione dei dati
- rappresentazione grafica dei dati

R è un linguaggio object-oriented (come C++ e Java): è un ambiente statistico di programmazione ad oggetti (vettori, tabelle, dataset), in cui ogni oggetto viene trattato con un metodo specifico. R è un linguaggio *free* quindi si può scaricare gratuitamente da Internet all'indirizzo: http://cran.r-project.org/. E' un linguaggio in costante sviluppo.

Prime nozioni di base:
Avviando R si apre una finestra, chiamata "R Console", con cui interagisce l'utente. Dopo alcune informazioni che vengono riportate automaticamente all'avvio del programma, compare il simbolo > (prompt di R), dopo il quale si possono inserire i comandi. I comandi possono essere in riga uno dietro l'altro, separati da un punto e virgola, oppure andando ogni volta a capo con il tasto Invio.
Importante: R è un linguaggio *case-sensitive*, ovvero distingue le lettere maiuscole dalle minuscole.
R è dotato di una guida on-line che permette di ottenere informazioni sui comandi. Per accedere alla guida è necessario scrivere
>help() per avere informazioni generali

>help(nome comando) per avere informazioni su un comando specifico
Per conoscere invece delle dimostrazioni di alcune funzioni (come ad esempio costruire dei grafici) si utilizza
>demo(nome funzione)

Gli oggetti in R: scalari e vettori
Come abbiamo già accennato R lavora con gli oggetti: gli oggetti fondamentali sono i numeri (scalari) e i vettori.
Per assegnare una variabile si utilizza il comando <-

> x<-4 assegna a x il valore 4

Se si vuole sapere quale è il valore di x basta richiamarlo nel prompt

>x
[1] 4

compare così il valore assegnato. Il simbolo [1] significa che R interpreta x come un vettore di lunghezza 1.
Per creare un vettore con più elementi si utilizza il comando c(.,.,.,.,.) dove gli elementi sono separati da virgole.
Ad esempio:

> y<-c(3,7,9,45) assegna a y il vettore (3,7,9,45)

Se vogliamo creare un vettore costituito da una successione di numeri che differiscono tra di loro sempre per lo stesso passo, si usa la funzione
seq (minimo, massimo, incremento)

>seq(1,5,1) otteniamo
>[1] 1 2 3 4 5

Il comando lenght(x) dà come risultato la lunghezza del vettore x.

Matrici

Le matrici sono tabelle di dati organizzati in righe e colonne.
In R il comando per inserire una matrice è
matrix(dati, n° righe, n° colonne)
Esempio:

```
>A<-matrix(c(1,2,3,4),2,2)
>A
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

Come si può osservare R di default costruisce le matrici per colonne: se si vuole costruire la matrice per righe, bisogna aggiungere il comando byrow=TRUE dopo il numero di colonne.
Se vogliamo estrarre un elemento di una matrice, una riga o una colonna è sufficiente utilizzare le parentesi quadre [].

```
>A[2,2] estrae l'elemento (2,2) dalla matrice cioè 4
>A[1,]   estrae la prima riga di A
>A[,2]   estrae la seconda colonna di A
```

Ambiente di lavoro

Per visualizzare tutti gli oggetti creati nel nostro ambiente di lavoro (workspace) si deve digitare

```
>ls()
```

mentre se vogliamo rimuovere la variabile x dall'ambiente di lavoro o tutto ciò che è stato creato

```
>rm(x)            rimuove solo la variabile x
>rm(list=(ls))    rimuove tutti gli oggetti
```

Per uscire da una situazione di errore digitare (“.”)

Importazione di dati da file Excel
Per importare in R dei dati salvati su un foglio di calcolo Excel si deve eseguire la seguente procedura:

- salvare il file.xls in formato “testo delimitato da tabulazione
- assicurarsi che la directory di R sia la stessa di quella dove è stato salvato il file (altrimenti cambiarla dal menu di R)
- richiamare il file in R usando il comando >read.delim(“nomefile.txt”)

A questo punto R visualizza tutti i dati organizzati in righe e colonne come erano strutturati in Excel

Costruzione di un dataframe
Per poter analizzare dal punto di vista statistico i dati caricati al punto precedente è utile costruire un dataframe, ovvero un oggetto simile ad una matrice , ma usato per rappresentare dati sperimentali.
Ogni riga contiene un’unità statistica, mentre le colonne le variabili misurate sulle unità statistiche. Le colonne possono contenere sia variabili numeriche che di testo.
Supponiamo di avere importato in R il file anamnesi.txt dei dati anamnestici avente come colonne le variabili: Età, Luogo di Nascita, Peso, Altezza. Si vuole creare un dataframe con questi dati.
Assegniamo innanzitutto un nome al file:

>anamnesi<-read.delim(“anamnesi.txt”)

Per creare il dataframe usiamo il comando:

>data.frame(anamnesi)
>dati<-data.frame(anamnesi)

Con questo ultimo comando abbiamo assegnato al dataframe il nome dati, quindi volendo richiamarlo basterà digitare sul prompt

>dati

Ora si possono avere delle informazioni di tipo statistico.
Il comando

>summary(dati)

da informazioni su ogni colonna: se si tratta di dati di testo, come ad esempio Luogo di Nascita, fornisce il numero di persone nate in ogni luogo, mentre se si tratta di dati numerici vengono dati gli indici statistici (minimo, 1°quartile, mediana, media, 3° quartile, massimo).

Per selezionare una sola colonna si usa il simbolo $:

>dati$Peso

restituirà solo i valori dei pesi (analogo discorso per gli altri dati)

Regressione Lineare
In R il comando lm (linear model) calcola la retta di regressione tra due variabili.
Esempio: vogliamo vedere se c'è una relazione lineare tra il peso e l'altezza del dataframe.
Innanzitutto assegniamo le variabili:

>x<-dati$Altezza i dati delle altezze sono memorizzati in x

>y<-dati$Peso i dati dei pesi sono memorizzati in y

>lm(y~x) restituisce la retta di regressione

GRAFICI

Per una prima analisi descrittiva dei dati è molto utile rappresentarli graficamente. Ogni volta che viene usato un comando per disegnare un grafico si apre automaticamente una finestra nuova su cui compare il grafico. Questo può essere copiato e incollato in altri ambienti di lavoro.

Il comando plot:

>plot(x,y) è il diagramma di dispersione di y (vettore ordinate), rispetto ad x (vettore ascisse).
Con l'opzione type="l" il grafico viene visualizzato con una linea continua invece che per punti.
Plot viene anche utilizzato per creare un grafico a colonna per la rappresentazione di variabili qualitative.

>plot(dati$Luogo di Nascita)

rappresenterà un grafico dove ogni colonna ha un'altezza pari al numero di persone nate in quel luogo (frequenze assolute).
E' possibile avere in ordinate le frequenze relative in questo modo: il comando table () restituisce le frequenze assolute. Quindi

>plot(table(dati$Luogo di Nascita)/lenght(dati$Luogo di Nascita)

disegnerà un grafico a colonna con frequenze relative.

Il comando curve:

>curve(funzione, minimo, massimo, n° di punti)
Richiede un'espressione di una funzione che dipenda da una variabile x, gli estremi dell'intervallo su cui vogliamo disegnare la funzione ed eventualmente il numero di punti.
Esempio:

>curve(sin(x), -5,5) è la funzione sin(x) nell'intervallo [-5,5].

Istogramma:
L'istogramma è la rappresentazione grafica tipica per una variabile quantitativa (come ad esempio peso, altezza..).
Il comando per creare un istogramma di dati contenuti in un vettore x è:

>hist(x, vettore delle classi di frequenza)

Se si vuole un istogramma con frequenze relative

>hist(x, freq=FALSE)

perché di default R costruisce istogrammi con in ordinata le frequenze assolute.

Box-Plot
Il box-plot è un tipo di rappresentazione che da indicazioni sulla simmetria o asimmetria di una distribuzione.
Il box-plot è costituito da una scatola, i cui estremi sono il 1° ed il 3° quartile (Q1, Q3), divisa dalla mediana avente dei baffi in corrispondenza dei valori minimo e massimo. In presenza di outliers (valori anomali), che per R sono quei valori che stanno al di fuori dell'intervallo
[Q1-1.5(Q3-Q1),Q3+1.5(Q3-Q1)], i baffi vengono posti in corrispondenza delle osservazioni più vicine agli estremi di tale intervallo e interne ad esso.
Gli outliers vengono evidenziati da R con dei puntini.
Il comando per disegnare un box-plot è

>boxplot(x)

Opzioni per i grafici

R ha a disposizione diverse opzioni riguardanti i grafici.
Gli argomenti “lty”, “lwd” e “col” definiscono, rispettivamente, il tipo di tratteggio, lo spessore e il colore del tratto.

- lty e lwd hanno come argomento i numeri da 2 a 5 (lty=1 è la linea continua e lwd=1 è lo spessore prestabilito)
- col ha la sintassi: col=”nome colore” (in inglese)

Se si vuole disegnare due curve nello stesso grafico si utilizza il comando curve con l’opzione add=TRUE

```
>curve(dnorm(x), -5,5)
>curve(dt(x, df=1), -6,6, lty=2, add=TRUE)
```

rappresenterà la distribuzione normale e la distribuzione di t di Student a 1 grado di libertà nello stesso grafico.

Mentre se si vuole aprire in una finestra più sottofinestre in modo da avere dei grafici appaiati per confrontarli il comando è:

```
>par(mfrow=c( , ))
```

dove c (,) mi indica quante sottofinestre si vuole avere.

E’ possibile attribuire un titolo al grafico con il comando main=”titolo” e assegnare delle etichette agli assi cartesiani con xlab=”..” e ylab=”..”
Esempio:

```
>plot(dati$Luogo di Nascita, xlab=”luogo di nascita”,
        ylab=”frequenze assolute”, main=”grafico a colonne”)
```

Il comando “legend” consente di inserire una legenda all’interno di un grafico.
La sua sintassi è: coordinate del punto in cui si vuole inserire la legenda, un vettore di caratteri alfanumerici, un vettore di colori (col) o di diversi tratteggi (lty).
Ad esempio inseriamo una leggenda nel grafico delle due distribuzioni normale e t di Student di prima, indicando con Z la prima e t la seconda:

```
>legend(5,0.3, c(“Z, “t”), lty=c(1,3))
```

Il comando “text” aggiunge testi e notazioni a grafici preesistenti; vuole in input il vettore di coordinate del punto in cui inserire il testo, quindi il testo. Il comando “expression”, all’interno di “text”, consente di scrivere formule e simboli matematici.

Probabilità con R

R mette a disposizione numerose funzioni di distribuzione, sia discrete che continue.

- Le probabilità puntuali si calcolano in R premettendo il suffisso “d” al nome della variabile
- Il valore della funzione di ripartizione in un punto si calcola premettendo il suffisso “p”
- I quantili q della variabile si ottengono premettendo al nome della variabile il suffisso “q”

Distribuzione binomiale

Se indichiamo con p la probabilità di successo di un esperimento aleatorio X, per calcolare la probabilità di ottenere k successi su n prove useremo il comando:

>dbinom(k,n,p)

Per calcolare $P(X \leq k)$ (probabilità di avere al massimo k successi) significa che vogliamo il valore della funzione di ripartizione nel punto k, quindi:

>pbinom(k,n,p)

Infine per calcolare $P(X>k)=1-P(X \leq k)$ (probabilità di avere almeno k+1 successi) si utilizza l’opzione “lower.tail=FALSE” che determina la probabilità complementare:

>pbinom(k,n,p,lower.tail=FALSE)

Il comando per calcolare i quantili della distribuzione vuole prima il valore di $_{\alpha}$, dove $\alpha = P(X \leq q)$ e poi i parametri della distribuzione in esame.

>qbinom(α , n, p)

Distribuzione di Poisson:
La probabilità che un evento casuale si verifichi x volte quando in media si verifica λ volte (media della distribuzione) è data dalla distribuzione di Poisson. Il comando in R è il seguente:

>dpois(x, λ)

Analogamente si utilizzano gli altri comandi per la funzione di ripartizione e i quantili.

Tabelle di Contingenza:
Una tabella di contingenza è una tabella a doppia entrata utilizzata per analizzare le relazioni tra due o più variabili. In questo tipo di tabelle sono riportate le frequenze congiunte delle due variabili.
In R è possibile costruire delle tabelle di contingenza in modo poi da calcolare le probabilità condizionate e verificare se c'è dipendenza o meno tra le variabili prese in considerazione.
Esempio: relazione tra sottoporsi ad un vaccino e malattia

Vaccino/Malattia	Non si ammala	Si ammala
Vaccino Si	700	400
Vaccino No	1300	7600

Riportiamo questi dati così organizzati in R:

```
>vaccino<-c("Vaccino si", "Vaccino no")
>esito<-c("Non si ammala", "Si ammala")
>tab<-matrix(c(700,1300,400,7600),2,2,
                    dimnames=list(vaccino,esito))
```

Richiamando >tab otterremo la tabella.

L'opzione dimnames serve per aggiungere l'intestazione alla tabella in modo che i dati siano più leggibili.

Visto che la tabella è costruita tramite il comando matrix è possibile intervenire separatamente sulle righe e colonne.
In particolare possiamo ottenere le frequenze marginali con il comando:

```
>margin.table(tab,1)    frequenze marginali di riga
>margin.table(tab,2)    frequenze marginali di colonna
```

Per calcolare le probabilità condizionate si utilizza il comando

```
>prop.table(tab,1)      probabilità condiz.rispetto ai totali di
                        riga
>prop.table(tab,2)      probabilità condiz.rispetto ai totali di
                        colonna
```

Per sapere se c'è le due variabili sono indipendenti oppure no si può fare un test di indipendenza del Chi-quadro:

```
>chisq.test(tab)
```