Formule laboratorio di Probabilità e Statistica

STATISTICA DESCRITTIVA:

x = c(1, 2, 3, 4) // Creazione di un vettore di valori “x”

x = c(“A”, “B”, “C”) // Creazione di un vettore di stringhe “x”

Esempio:

mat = c(25, 28, 30)

prog = c(23, 26, 28)

studenti = c(“Mario”, “Luca”, “Pier”)

x = data.frame(y, z) // Creazione di un'unica tabella “x”

Esempio:

voti = data.frame(studenti, mat, prog)

media = rowMeans(x[righe, colonne]) // Calcolo la media di una tabella o matrice “x”

Esempio:

media = rowMeans(voti[1:3, 2:4])

round(x, cifre_decimali) // Arrotondo “x” di n “cifre_decimali”

Esempio:

media = round(media, 2)

barplot(x, names.arg = y, col = heat.colors(n)) // Creazione di un istogramma, dove “x” è un
// vettore di numeri e “y” è un vettore di nomi (stringhe)

Esempio:

barplot(mat, names.arg = studenti, col = heat.colors(4))

pie(x, labels = y) // Creazione di un grafico a torta, dove “x” è un vettore di numeri e “y” è un

// vettore di nomi (stringhe)

Esempio:

pie(mat, labels = studenti)

ALCUNE UTILI FUNZIONI:

sum() // Calcola la somma degli elementi di un vettore di dati

length() // Restituisce la numerosità di un vettore

range() // Per trovare il minimo e il massimo di un vettore

mean() // Calcola la media

weighted.mean(x, pesi) // Calcola la media ponderata

median() // Calcola la mediana

sd() // Calcola lo scarto quadratico medio campionario di un vettore di dati

var() // Calcola la varianza campionaria di un vettore di dati o la covarianza tra due vettori

cor() // Calcola la correlazione tra due vettori

summary() // Riporta le principali statistiche descrittive di un vettore o di una matrice di dati

TABELLE A DOPPIA ENTRATA E MISURA CONNESSIONE TRA 2 FENOMENI:

// Creazione di una matrice di valori (letti per riga):

x = matrix(c(valore_1, valore_2, valore_3), nrow = numero_righe, byrow = TRUE)

// Le righe e le colonne possono avere dei nomi tramite il comando:

dimnames(x) = list(nome_righa, nome_colonna)

// Dove “nome_righa” e “nome_colonna” sono:

nome_righa = c(“righa_1”, “righa_2”, “righa_3”)

nome_colonna = c(“colonna_1”, “colonna_2”, “colonna_3”)

// Creazione del grafico a mosaico:

mosaicplot(x)

// Calcolo del test del chi-quadrato, dopodiché questo valore andrà confrontato con quello della tabella e se quello trovato è maggiore di quello della tabella allora si rifiuta l'ipotesi di indipendenza e ci sarà connessione tra i 2 fenomeni:

```
testchiq = chisq.test(x)
```

// Calcolo dell V di Cramer, ovvero quanto è forte questa connessione:

```
chiquadrato = testchiq$statistic
N = sum(x)       // Numero totale di elementi presenti
V = sqrt(chiquadrato / (N * (min(righe, colonne) – 1))       // Da 0 a 0,2 = Bassa connessione
                                                              // Da 0,2 a 0,4 = Discreta connessione
                                                              // Da 0,4 a 0,6 = Buona connessione
                                                              // Da 0,6 in su = Alta connessione
```

Esempio:

```
SO = matrix(c(100, 180, 320, 60, 120, 50, 50, 60, 60), nrow=3, byrow=TRUE)
SOpc = c("Windows", "Mac OS", "Linux")
SOsmart = c("Windows", "iOS", "Android")
dimnames(SO) = list(SOpc, SOsmart)
mosaicplot(SO)
testchiq = chisq.test(SO)
```

// Risultato di 78,1 ovvero molto più alto di quello della tabella, quindi c'è connessione tra i 2 fenomeni

```
Chiquadrato = testchiq$statistic
N = sum(SO)
V = sqrt(chiquadrato / (N*(3-1)))
```

// Risultato di 0,1975, quindi c'è una bassa connessione tra i 2 fenomeni

LA REGRESSIONE LINEARE:

// Creazione del grafico delle variabili:

```
plot(x, y)
```

// Con "x" e "y":

```
x = c(1, 2, 3, 4, 5)  y = c(6, 7, 8, 9, 10)
```

// Regressione lineare fra le 2 variabili (invertite):

retta = lm(y ~ x)

// Per disegnare la retta di regressione lineare:

abline(retta, col = “blue”)

// Per aggiungere dei segmenti che collegano la retta ai singoli punti:

segments(x, fitted(retta), x, y, lty = 2)

// Per dare un titolo al grafico:

title(main = "Esempio retta interpolare")

// Per disegnare il grafico dei residui:

plot(fitted(retta), residuals(retta))

// Per disegnare la retta orizzontale delle ordinate zero:

abline(0, 0)

// Calcolo del coefficiente di correlazione lineare:

R = cor(x, y) // Se R = -1 -> Perfetta relazione lineare inversa

// Se R = 0 -> Indipendenza lineare

// Se R = 1-> Perfetta relazione lineare diretta

// Calcolo del coefficiente di determinazione:

R2 = R ^ 2

// Se R2 = 0 il modello teorico Y’ non riesce a spiegare nulla della variabilità delle osservazioni Y

// Se R2 = 1 il modello teorico Y’ spiega in maniera perfetta la variabilità delle osservazioni Y

Esempio:

orestudio = c(30, 50, 40, 85, 60, 80, 70)

voto = c(10, 18, 16, 30, 20, 28, 26)

retta = lm(voto~orestudio)

plot(orestudio, voto)

abline(retta, col = "blue")

segments(orestudio, fitted(retta), orestudio, voto, lty = 2)

title(main = "Regressione lineare fra Ore di studio e Voto in statistica")

Y’ = 0.55241+0.37431*orestudio // Modello teorico

plot(fitted(retta), residuals(retta))

abline(0, 0)

R = cor(orestudio, voto)

R2 = R^2

// Il risultato del coefficiente di determinazione è 0,9784 quindi il modello teorico usato si adatta molto bene ai valori osservati

VARIABILI CASUALI BINOMIALE:

// Calcolo la densità di probabilità:

x = dbinom(k, n, p) // Con "k" il vettore degli "n", "n" il numero totale di elementi e
// "p" la probabilità che si verifichi l'evento

// Per disegnare il grafico della probabilità:

barplot(x, names.arg = k)

Esempio:

// Se la probabilità di passare l'esame di statistica è del 70% e si presentino 5 studenti, descrivere

// con una variabile casuale le probabilità che gli studenti vengano promossi:

k = c(0:5)

passati = dbinom(k, 5, 0.7)

barplot(passati, names.arg = k)

// Cumulata delle probabilità:

x = pbinom(k, n, p)

Esempio:

// Supponiamo che Tizio debba fare un test con 30 domande, ciascuna domanda con 3 risposte;

// descrivere con una variabile casuale le probabilità che, rispondendo a caso, azzecchi un numero

// di domande compreso tra 0 e 10:

test10p = pbinom(10, 30, 1/3)

// E a più di 10 domande?

test10k = 1 – test10p

// Inversa della densità di probabilità (restituisce il valore “k” di una certa probabilità):

x = qbinom(percentuale, n, p)

// Possiamo usarla per calcolare la mediana:

mediana = qbinom(0.5, n, p)

// Il primo quartile (che corrisponde al 25%):

primo_quartile = qbinom(0.25, n, p)

// Il terzo quartile (che corrisponde al 75%):

terzo_quartile = qbinom(0.75, n, p)

// Per generare valori random che seguono lo schema binomiale si usa:

x = rbinom(numero_tentativi, n, p)

Esempio:

// Ipotizziamo che Tizio provi 5 volte l’esame rispondendo a caso, quali voti prenderà?

voti = rbinom(5, 30, 1/3)

// La variabile di Poisson si calcola come:

x = dpois(k, λ)

// Per creare un grafico della variabile si usa:

barplot(x, names.arg = k)

Esempio:

// La probabilità che una macchina produca un pezzo difettoso è in media di 2 pezzi all’ora (λ = 2),

// descrivere con una variabile casuale la probabilità di avere un numero di pezzi difettosi

// all’ora da 0 a 5:

k = c(0:5)

difettosi = dpois(k, 2)

barplot(difettosi, names.arg = k)

VARIABILI CASUALI CONTINUE:

// Per creare un asse delle x:

x = seq(lunghezza_da, lunghezza_a, by = 0.01)

// Creo la distribuzione normale con la funzione “dnorm”:

y = dnorm(x, μ, σ) // Con “x” l’asse delle x, “μ” la media e “σ” lo scarto quadratico medio

// Per creare il grafico della probabilità:

plot(x, y, type = "l", xlab = "nome in x", ylab = "nome in y", col = "red")

// Se volessi aggiungere un altro dato:

z = dnorm(x, μ, σ)

lines(x, z, col = "blue")

// Per aggiungere un titolo generale:

title(main = "Titolo")

// Per aggiungere una legenda:

legend("topright", c("y", "z"), cex = 1, bty = "n", col = c("red", "blue"), lty = 1:2)

Esempio:

// Proviamo a disegnare la distribuzione di probabilità dell’altezza media delle donne italiane. Sappiamo che l’altezza media delle donne intorno ai 20 anni del bel Paese è di 168 cm e che la variabilità ha uno scarto quadratico medio di 12 cm:

x = seq(120, 240, by = 0.01)

donne = dnorm(x, 168, 12)

plot(x, donne, type = "l", xlab="altezza in cm", ylab = "densità di probabilità", col = "red")

// Se volessimo aggiungere il dato sull’altezza relativo agli uomini, possiamo farlo sapendo che l’altezza media è di 178 cm con una variabilità di 15:

uomini = dnorm(x, 178, 15)

lines(x, uomini, col = "blue")

title(main = "Distribuzione dell'altezza per Donne e Uomini italiani")

legend("topright", c("donne", "uomini"), cex = 1, bty = "n", col = c("red", "blue"), lty = 1:2)

// Se voglio conoscere i valori in un certo punto utilizzo la funzione “pnorm” (dopo aver scritto la funzione “dnorm”):

pnorm(valore_desiderato, μ, σ, lower.tail = ……)

// lower.tail = TRUE indica che voglio sapere i valori a sinistra di un determinato punto

// lower.tail = FALSE indica che voglio sapere i valori a destra di un determinato punto

// Se voglio calcolare la probabilità fra due punti:

pnorm(valore_più_alto, μ, σ, lower.tail = TRUE) - pnorm(valore_più_basso, μ, σ, lower.tail = TRUE)

// Per calcolare un valore di una certa percentuale uso la funzione “qnorm” (dopo aver scritto la funzione “dnorm”):

qnorm(percentuale_desiderata, μ, σ)

// La funzione “rnorm” genera numeri casuali distribuiti secondo una forma normale :

rnorm(numero_valori, μ, σ)

VERIFICA DI IPOTESI:

// Ipotesi tra un vettore di valori:

t.test(x, mu = valore, alternative = "two.sided", conf.level = valore)

// “x” è un vettore di dati, “alternative” specifica l'ipotesi alternativa, “mu” è l'ipotesi nulla sul valore della media (test su un campione), o sulla differenza tra medie (test su due campioni), di default mu = 0, “conf.level” è il livello di confidenza dell'intervallo, ovvero il complemento ad 1 della probabilità dell'errore di prima specie fissato, di default posto uguale a 0.95

// Se il livello di significatività alpha teorico (100 % - livello di confidenza) < p-value calcolato (dal t.test) si accetta l’ipotesi nulla H0 di uguaglianza

// Per confrontare le medie di due campioni indipendenti con varianze uguali:

t.test(x, y, var.equal = TRUE, conf.level = valore)

// Per confrontare le medie di due campioni indipendenti con varianze diverse:

t.test(x, y, var.equal = FALSE, conf.level = valore)

// Verifica di ipotesi per dati appaiati, a volte i dati vengono presentati come coppie di una situazione prima e dopo un dato trattamento:

// H0: diff(prima – dopo) <= 0 H1: diff(prima – dopo) > 0

t.test(prima, dopo, alternative = "greater", paired = TRUE, conf.level = valore)