MATLAB

Sommario

Il comando clc permette di pulire la console, mentre il comando clear permette di pulire l'ambiente di lavoro (cancellare le variabili)

Assegnamento e visualizzazione

```
x=5;
disp(x);
class(x); %ci permette di vedere la classe/tipo di x
```

Array e size

```
A=[1,2 3];
size(A);
B=zeros(3,4);  %crea matrice 3 per 4 di zeri
B(2,3)=4;  %nella cella 2,3 inseriamo il valore 4
disp(B);
```

Per concatenare vettori si usa la cat(dim,A,B) , concatena B ad A , dim specifica come.
Se dim = 1 lo fa “per colonne”, se è 2 lo fa “per righe”

Lettura tabella strutturata su un file di testo

```
T=readtable("speed-and-density.txt");
disp(size(T));
% Otteniamo un tipo table n*m (che non è una matrice)
%Possiamo convertire poi una sua colonna in un array in questo modo
speed=table2array(T(:,1)); %T(:,1) ottiene tutte le righe,indicate con :, e solo la
%colonna 1 , in pratica otteniamo la colonna 1 che trasformiamo in array
disp(speed);
```

Matrici

```
m=[1 2 3;4 5 6;7 8 9]; %creiamo matrice 3*3 , gli spazi valgono come delle virgole
disp(m); %mostriamo la matrice originale
disp(m'); %mostriamo la trasposta
disp(inv(m)); %mostriamo l’inversa di m (se esiste)

vett=1:10; %otteniamo il vettore 1,2,3 … 10
disp(vett);
disp(10:-4:1);  %da 10 andiamo verso 1 a passo -4 , si ottiene 10,6,2 in questo caso
disp(2:3:17); %da 2 andiamo verso 17 a passo 3, si ottiene 2,5,8,11,14,17 così

O=ones(3,3); %otteniamo una matrice 3*3 di 1
disp(O);

indice=find(vett<5); %otteniamo un vettore dove sono contenuti gli indici che
%rispettano la condizione booleana all’interno di find
disp(indice);

a=[1 2 3];
b=[1 -3 2];
disp(a.^2); %otteniamo un vettore della stessa lunghezza di a dove ogni elemento è
elevato a 2



mat1=[1 -3 4; 3 -4 -6;1 2 -2];
mat2=[1 1 0;3 -7 4;1 2 -2];
disp(mat1*mat2); %prodotto righe per colonne tra matrici
disp(mat1 .* mat2); %prodotto elemento per elemento tra matrici
disp(mat1/mat1); %divisione tra matrici, in questo caso si ottiene I
```

Celle

```
data={[1,2,3],"strings"};
class(data); %torna 'cell'
disp(data);
disp(data(1,2)); %torna la riga 1 della colonna 2 ovvero "strings" però sottoforma di
%cella che ha al suo interno un array
% In questo caso viene tornata { [ "strings" ] }
```

Struct

Si può creare una struttura dati semplice in questo modo

```
f=struct('name',{'dario'},'age',30);
%il campo 'name' ha valore {'dario'} (è una cella con dentro una stringa) , mentre
%'age' ha valore 30
disp(f);
a = extractfield(f,'name'); %questa funzione richiede Mapping toolbox, estrae valore di
'name' in f
```

Funzioni

```
f=@(x)x.^2; %abbiamo creato una funzione, chiamata f, che ritorna il valore al quadrato
%dell'input(che sia un array o un valore singolo non importa.
disp(f(2)); %vediamo il valore di output con 2

x=-1:0.01:1;
Il vettore x va da -1 a 1 con passo 0.01
y=f(x); %otteniamo un vettore di pari lunghezza a x dove ogni elemento è al quadrato

Possiamo costruire una funzione anche con un'altra sintassi, la precedente è usata in
genere se la funzione è semplice come un'elevazione a quadrato

function f=parabola(x)
    f=x.^2;
end

Otteniamo una funzione chiamata parabola dove il valore di output all'interno della
funzione è f. Il valore restituito è f (l'ultimo valore in pratica)
Nel caso si crei una funzione così nello stesso file dello script da eseguire: la
sintassi di matlab richiede che la funzione sia posta alla fine del file (tutto giù)

p=parabola(x);
```

Plot

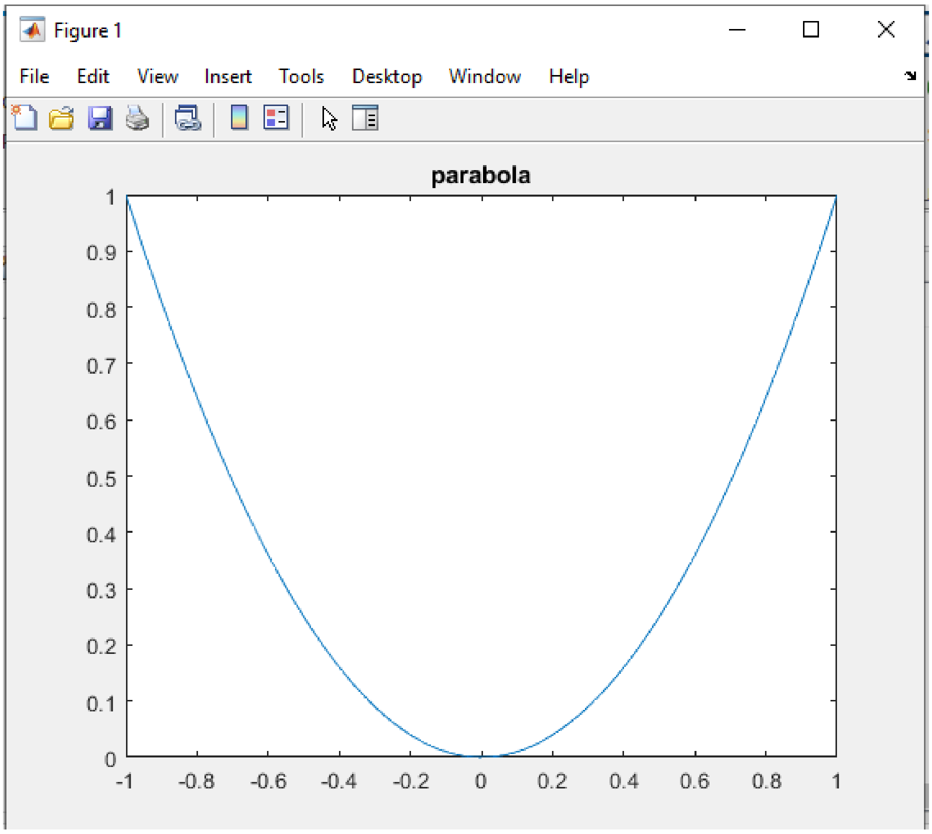
plot(x,y); %crea un grafico dove abbiamo y in funzione di x

Tornando alle funzioni e valori di prima possiamo fare una cosa del genere:

```
x=-1:0.01:1;
p=parabola(x);
plot(x,p);
title("parabola"); %da un nome al grafico

function f=parabola(x)
    f=x.^2;
end
```

Otteniamo una finestra simile a questa



Meshgrid

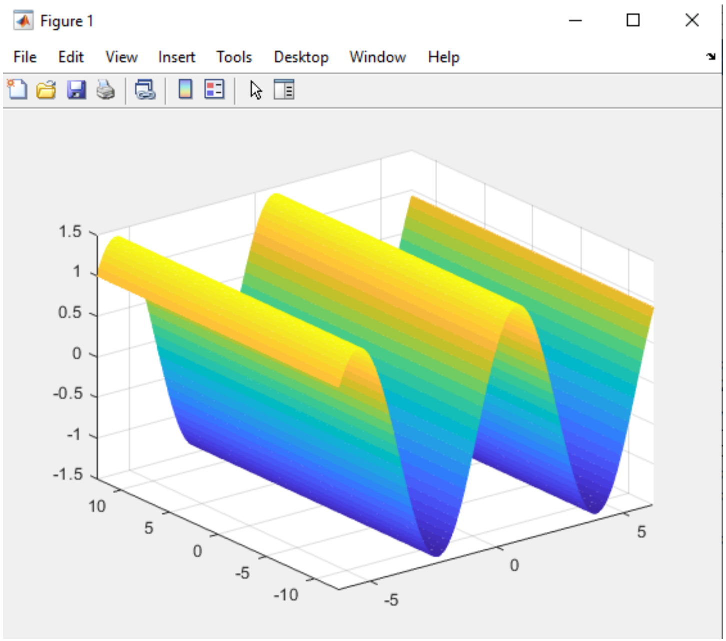
```
[X,Y]=meshgrid(-2*pi:0.1:2*pi , -4*pi:0.1:4*pi);
```

Ritorna una griglia 2D basata sulle coordinate del primo vettore e del secondo.
X è una matrice dove ogni riga è la copia del primo array di input mentre Y è una matrice dove ogni colonna è la copia del secondo array. La griglia rappresentata ha queste dimensioni:

1. lenght(-4*pi:0.1:4*pi) righe
2. lenght(-2*pi:0.1:2*pi) colonne

```
Z=sin(X)+cos(X); %creiamo una terza matrice in funzione delle prime 2
xlabel("X"); %diamo un nome all'asse X
ylabel("Y");
zlabel("Z");
surf(X,Y,Z); %simile alla funzione plot, creiamo un grafico 3D
```

```
hold on %mantiene i grafici
mesh(X,Y,Z); %crea una grafico mesh a 3 dimensioni
```



Esempio plot e lettura da file

```
T=readtable("speed-and-density.txt");
disp(T);
speed=table2array(T(:,1));
dens=table2array(T(:,2));
plot(speed,dens);
```

Plottiamo speed in funzione di dens

Ciclo for

```
a=1:5:100;
for i=1:length(a)
    disp(a(i));
end
```

Per i che va da 1 a length(a) facciamo le operazioni all'interno del ciclo (il passo con cui procede non è specificato, quindi sarà uguale ad 1)

IF statement

```
if u < -Z_uno_meno_alpha
    disp("Rifiuto H0");
elseif condizione
else
    disp("Non rifiutare H0");
end
```

Correlazione tra dati,coefficiente di Pearson

```
f=corrcoef(data1,data2); %otteniamo la matrice di correlazione tra data1 e data2
disp("la matrice di correlazione è");
disp(f);
```

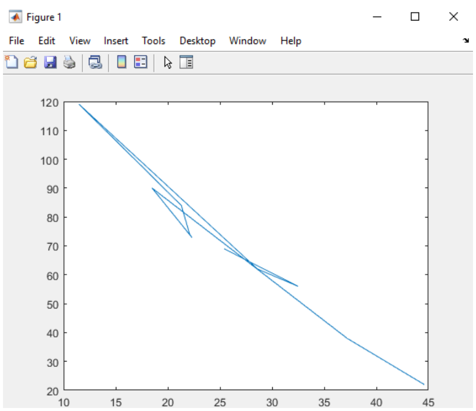
La matrice contiene i coefficienti di Pearson.
Il coefficiente di Pearson detto a volte anche solo coefficiente di correlazione si può trovare come sqrt(R2) dove R2 è il coeff. Di determinazione.
La matrice è una 2*2 dove in 0,0 c'è la correlazione tra data1 e data1, in 1,1 c'è la correlazione tra data2 e data2 mentre in 0,1 e 1,0 c'è la correlazione tra data1 e data2. Nella pratica ci interessa la correlazione tra data1 e data2 in quanto in generale è banale che la correlazione tra A e A è 1.

Esempio:

```
speedanddensity = readtable("speed-and-density.txt")
sp = table2array(speedanddensity(:,1));
dn = table2array(speedanddensity(:,2));
codd_data = corrcoef(sp,dn);
disp("coeff di correlazion è ");
disp(codd_data);
```

Dalla teoria dei coefficienti di correlazione si sa che se esso è tra 0.8 e 1 o tra -0.8 e -1 allora diciamo che i dati hanno una correlazione lineare.
Plottando i dati dovremmo notare un andamento simile a 1 retta.

```
xlabel("speed")
ylabel("density")
plot(sp,dn)
```



Esempio: creiamo una funzione che prende in input una matrice e 2 indici e calcola la correlazione tra 2 colonne, le 2 colonne referenziate tramite i 2 indici

```
function f=coeffCorrTab(Matrice,indice1,indice2)
    data1 = Matrice(:,indice1);
    data2 = Matrice(:,indice2);
    f = corrcoef(data1,data2);
    disp(f)
    plot(data1,data2,'o')
    xlabel('data1')
    ylabel('data2')
end


function f=corr_data(data1,data2)
    f=corrcoef(data1,data2);
    disp("la matrice di correlazione è")
    disp(f)
    plot(data1,data2,'x')
    xlabel('data1')
    ylabel('data2')
end
```

Funzione praticamente identica per la logica a ciò che è stato fatto prima

Dati statistici

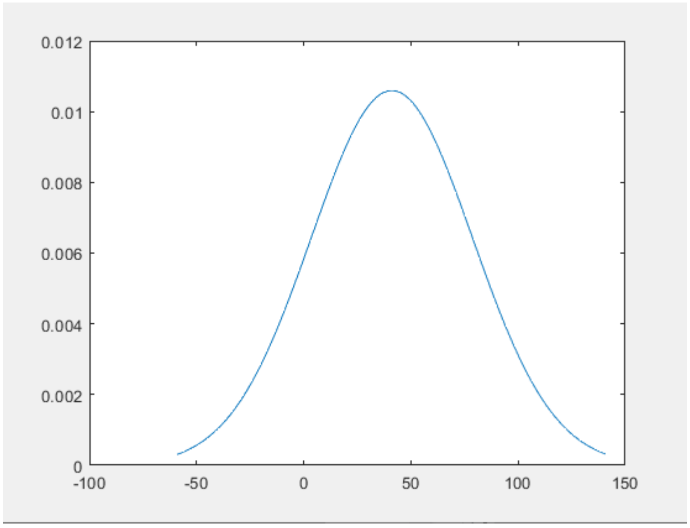
```
T = readtable('fish.txt');
p = T.("Price_1970"); %prendiamo la colonna chiamata con la stringa dentro alle parentesi, si potrebbe usare anche un indice
disp(p);
M = mean(p); %estraiamo la media
V = var(p); %estraiamo la varianza

S=std(p); %estraiamo la deviazione standard (è la radice quadrata della varianza)
disp(M);
disp(V);
```

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Creiamo una funzione inline che crea una gaussiana,

```
Normal = @(x)(1/S/sqrt(2*pi))*exp(-(x-M).^2/V/2);
disp(Normal(p));
Vett=(M-100):1.0:(M+100);
plot(Vett,Normal(Vett));
```




```
Covarianza:

cov(A,B);
Dove A e B sono 2 vettori di stessa lung.
```

```
Nota: quando plottiamo una gaussiana(o meglio, quando creiamo il vettore di output) facciamo attenzione che il vettore di input sia ordinato(in senso crescente)
```

Ordinamento

```
x = sort(x); %si potrebbe usare un altro parametro “direction” da mettere uguale a
%‘ascend’ o ‘descend’
```

Integrali

```
f=@(x)x.^2;
x=0:0.1:2;
y=f(x);
integral(f,0,1);
g=@(x)exp(-2*x.^2);
disp(integral(g,1,inf));
```

Calcolare integrali con 2 variabili ma rispetto a 1 sola variabile(utile per il metodo dei momenti)

```
syms t x f
assume(t,'real'); Con le varie assume appunto diamo delle caratteristiche ai syms
assume(t>=0);
f(x,t)=x*t*x^(t-1);
a = int(f,x,0,1); Integriamo f rispetto a x con limiti 0 e 1
disp(a); Otteniamo un’espressione in funzione di t
Per risolvere l’espressione a è abbastanza semplice
syms y
disp(solve(a==y,t)); troviamo il valore di t(rispetto a y)
```

Funzione norm

Esistono già delle funzioni che calcolano la normale ovviamente

```
p=sort(p);
Normale=normpdf(p,M,S); %ritorna la pdf normale, vuole in input l’array/vettore, la
media e la deviazione standard
plot(p,Normale);
Normcdf = normcdf(p,M,S); %ritorna la cumulativa
RND=normrnd(M,S); %genera un numero casuale dalla normale con media M e deviazione S
disp(RND);
Per la casualità meglio applicare prima questa funzione: rng('default');
Inv = norminv(p,M,S) ritorna l’inversa della cdf(data la prob. p torna quel valore che
se messo nella normcdf darebbe p come risultato
```

Esercizio sulla normale

```
%distr normale media 0, sigma_quadro = 6
%determinare x_tilde tale che la prob di ottenere un valore assoluto di x
%minore di x_tilde sia uguale a 0.6

a=norminv(0.2,0,sqrt(6));
disp(-a);
```

Funzione T di Student

```
f1=@(x)tpdf(x,3); %3 gradi di libertà
f2=@(x)tpdf(x,8);
```

$$h(t) = \frac{\Gamma\left[(\nu+1)/2\right]}{\Gamma(\nu/2)\sqrt{\pi\nu}}\left(1+\frac{t^2}{\nu}\right)^{-(\nu+1)/2}, -\infty < t < +\infty.$$

Con ni(la v strana) che indica i gradi di libertà

E’ la distribuzione che segue la seguente variabile:

$$T = \frac{Z}{\sqrt{V/\nu}} \qquad T = \frac{(\bar{X}-\mu)/(\sigma/\sqrt{n})}{\sqrt{S^2/\sigma^2}}$$

Dove Z segue una normale standardizzat e V segue una chi quadro con ni(la v) gradi di libertà

Funzione chi quadro

```
fC=@(x)chi2pdf(x,3); %3 gradi di libertà
```

$f(x;\nu) = \frac{1}{2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)} x^{\frac{\nu}{2}-1} e^{-\frac{x}{2}}, x > 0$ con $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx$, per $\alpha > 0$

$\mu = \nu$ e $\sigma^2 = 2\nu$

Possibile esercizio sulla gaussiana

Data una gaussiana matlab può identificare il punto x dove corrisponde una percentuale, se vogliamo la percentuale tra due punti banalmente si calcola l'integrale della x più piccola e l'intergrale della x più grande. La differenza trai 2 integrali è la percentuale che vogliamo noi.

```
f=@(x)normpdf(x,5,1);
x= -15:0.1:25;
y=f(x);
plot(x,y);
disp(norminv(0.5,5,1));
%torna quella x per cui l'integrale
%tra -inf e x è 0.5
% cioè la probabilità degli elementi minori o uguali a questa x è 0.5

%tra 2 e 6 quanto è la probabilità?
f=@(x)normpdf(x,5,1);
disp(integral(f,2,6));

disp(norminv(0.8,5,1));
%otteniamo 5.8416
disp(integral(f,-inf,5.8416));
%otteniamo ovviamente 0.8
```

Distribuzione binomiale

$$b(x;n,p) = \binom{n}{x} p^x q^{n-x},\ \mu = np,\ \sigma^2 = npq$$

binopdf(x,n,p);
Torna la probabilità di avere x successi in n prove. Ogni prova ha probabilità p.

Distribuzione geometrica

$$g(x;p) = pq^{x-1},\ \mu = \frac{1}{p},\ \sigma^2 = \frac{1-p}{p^2}$$

Torna la probabilità che il primo successo sia al k-esimo tentativo
geo(k,p);
Da la possibilità che in una distr. con prob. p il primo successo si verifichi al k-esimo tentativo

Distribuzione di poisson

Data una densità detta lambda che definisce il numero di successi per unità di tempo, la poisson ci torna la probabilità che accadano x successi.

poisspdf(x,lambda);

$$p(x;\lambda t) = \frac{e^{-\lambda t}(\lambda t)^x}{x!},\ \mu = \lambda t,\ \sigma^2 = \lambda t$$

Distribuzione esponenziale

Torna la probabilità che il tempo necessario per il primo successo sia x, considerando una densità mu
exppdf(x,mu);

$f(x;\beta) = \frac{1}{\beta}e^{-\frac{x}{\beta}}$ per $x > 0$ e 0 altrove

$\mu = \beta$ e $\sigma^2 = \beta^2$

Distribuzione normale

normpdf(x,mu,sigma);

$$n(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

Distribuzione di Weibul

wblpdf(x,alpha,beta);

Da usare negli esercizi sui “guasti”

$f(x;\alpha,\beta) = \alpha\beta x^{\beta-1}e^{\alpha x^{\beta}}$ per $x > 0$ e con $\alpha > 0, \beta > 0$

$$\mu = \alpha^{-1/\beta}\Gamma\left(1+\frac{1}{\beta}\right), \quad \sigma^2 = \alpha^{-2/\beta}\left\{\Gamma\left(1+\frac{2}{\beta}\right) - \left[\Gamma\left(1+\frac{1}{\beta}\right)\right]^2\right\}$$

La funzione di ripartizione è $F(x) = 1 - e^{-\alpha x^{\beta}}$ per $x \geq 0$.

<u>Nota</u>

Tasso di guasto per la distribuzione di Weibull

Affidabilità di un qualche componente al tempo t:

$R(t) = P(T > t) = \int_t^{\infty} f(t)dt = 1 - F(t)$

La probabilità condizionata che un componente si guasti nell’intervallo compreso tra $T = t$ e $T = t + \Delta t$, dato che ha funzionato fino al tempo t, è

$$\frac{F(t+\Delta t) - F(t)}{R(t)}.$$

Il **tasso di guasto** è

$$Z(t) = \lim_{\Delta t \to 0} \frac{F(t+\Delta t) - F(t)}{\Delta t}\frac{1}{R(t)} = \frac{F'(t)}{R(t)} = \frac{f(t)}{1-F(t)},$$

quindi $Z(t) = \alpha\beta t^{\beta-1}$.

Applicazioni delle distribuzioni su dei campioni

Quando abbiamo a che fare con dei campioni, quindi dei set di dati, non possiamo applicare esattamente le stesse regole che si hanno nella teoria.

Media campionaria

Banale: sommo tutti gli n valori e faccio diviso n

Varianza campionaria

Se sappiamo la varianza v dalla consegna la varianza campionaria è v/n , n cardinalità popolazione

Ora facciamo caso di non avere la v in input.

$$S_n^2 = \frac{\sum_{i=1}^{n}(x_i - \mathrm{m})^2}{n}$$

Se il valore atteso è noto E[X]=m:
Molte volte S ha 0 al pedice

$$S_{n-1}^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1},$$

Se il valore atteso è ignoto:
xbar indica la media campionaria.

Se il campione(grande n elementi) segue una normale dove ci sono media m e varianza v segue che la varianza campionaria a valore atteso noto segue una chi quadro con n gradi di libertà con v/n davanti a chi-quadro. Invece la varianza campionaria con valore atteso ignoto segue una chi-quadro con n-1 gradi con v/(n-1) davanti alla chi quadro.

$$S_0^2 \sim \frac{V}{n}\chi^2(\nu = n)$$

Quella nelle parentesi è una ni

$$S_n^2 \sim \frac{V}{n-1}\chi^2(\nu = n-1)$$

Qui S con n al pedice è quella che sarebbe S con n-1 al pedice

Proprietà:

$$\frac{\bar{X}_n - m}{\sqrt{\frac{S_n^2}{n}}} \sim T(\nu = n-1)$$

Xbar è la media campionaria di un campione che segue una normale, m la media,S_n indica la varianza con valore atteso incognito

Distribuzione media campionaria

Se un campione di n elementi segue una normale con media m e varianza v succede che la media campionaria xbar segue una normale con media m e varianza v/sqrt(n)

In caso di 2 popolazioni tenere a mente questa formula:

$$Z = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{(\sigma_1^2/n_1) - (\sigma_2^2/n_2)}}$$

Questa variabile segue una normale standard.
Con alcuni passaggi si dimostra che Xbar – Ybar segue una normale con mu=mu1 – mu2 e varianza uguale a sigma2_1/n1 sommata a sigma2_2/n2

Mediana

Numero che sta nel mezzo. M=median(x)
Ordinati i dati in ordine crescente con n dispari la media è quel valore di indice (n+1)/2
Se n pari potrei calcolare la media trai valori in n/2 e (n+1)/2

Moda
Il valore che si ripete più volte in un insieme   M=mode(x)

Quantile
Definito a(tra 0 e 1) il quanti q_a è quel valore per cui alla sua sinistra compare il 100*a% dei valori e alla sua destra compare il 100(1-a)% dei valori. Si suppone i dati siano ordinati ovviamente
Si dice percentile se a è una percentuale, ma il concetto è lo stesso
I quartili sono dei particolari quantili che sono q_0.25 q_0.50 e q_0.75

Stime di parametri
Un parametro è un valore numerico che descrive una caratteristica di una popolazione, è una grandezza associata alla distribuzione.
Una stima di un parametro è un valore appunto stimato usando un campione
Estimatore puntuale: valore che stima una misura. La media campionaria è un estimatore del valore atteso.

Stime intervallarie
Stima di un parametro dati 2 numeri.
"Misura è compresa tra 5,28 + o – 0,03 m"

Intervallo di confidenza per un parametro
Intervalli dove la probabilità che il parametro assuma un valore è 1-alpha
NB: se dobbiamo fare dei calcoli sugli intervalli e abbiamo a che fare con un campione di n elementi la media rimane la stessa ma non possiamo usare la varianza così come la troviamo.
Dobbiamo usare la varianza campionaria, se abbiamo la popolazione la calcoliamo, se abbiamo una varianza teorica v dataci in input dal problema la varianza da usare sarà v/n

Nel prossimo esempio si vede valutare un intervallo di confidenza data una certa percentuale alpha avendo in input la media e la varianza della popolazione e la sua cardinalità

Esempio sul calcolo di intervallo di confidenza
```
n=36; %popolazione
mu=2.6; %media
v=0.3; %varianza teorica in input
S=v/n; %varianza campionaria

%1
alpha=0.95;
x11=norminv((1-alpha)/2,mu,sqrt(S));
x12=norminv(alpha+(1-alpha)/2,mu,sqrt(S));

%2
alpha2=0.99;
x21=norminv((1-alpha2)/2,mu,sqrt(S));
x22=norminv(alpha2+(1-alpha2)/2,mu,sqrt(S));
```

Altro esempio

In questo caso vogliamo non tanto un intervallo ma un limite superiore, con alpha al 95%(in pratica vogliamo una solo estremo di un possibile intervallo)

```
n=25; %popolazione
sigma2=4;
mu=6.2;

%la dev sarà sqrt(sigma2/n)
t_0_95 = norminv(0.95,mu,sqrt(sigma2/n));
disp(t_0_95);
```

Esempio con la popolazione

In questo caso la varianza campionaria va calcolata non usando la varianza teorica

```
valori=[9.8,  10.2, 10.4,  9.8,  10.0,  10.2,  9.6];

mu=mean(valori);
sigma2=0.0;
for i=1:length(valori)
    sigma2=sigma2+(valori(i)-mu)^2;
end
sigma2=sigma2/(length(valori)-1);  E' la varianza con valore atteso ignoto(non viene
dato nella consegna)
sigma=sqrt(sigma2/length(valori));
x1=norminv((1-0.95)/2,mu,sigma);
x2=mu+abs(mu-x1);
%L'intervallo sta tra x1 e x2
disp(x1);
disp(x2);
```

Stimare i parametri, metodo della verosimiglianza

E' un metodo per stimare un parametro basato sulle derivate.
Data una funzione che ha dei parametri voglio trovare il valore di un parametro che massimizzi la funzione, in presenza di una popolazione di n valori da dare alla funzione.
Ad esempio la esponenziale ha 2 parametri, beta e x(l'input), possiamo avendo una popolazione di n valori x trovare il valore di beta che massimizzi la distribuzione esponenziale.
Quello che facciamo è prima calcolare L come produttoria con i da 1 a n della funzione in x(i) e theta(un generico altro parametro). Trovata la produttoria la poniamo uguale a 0 e risolviamo l'equazione in theta.

Calcolare derivate

Si può calcolare la derivata prima di una funzione con dei comandi abbastanza semplici.
La funzione è diff(fun,x); dove fun è la funzione da derivare e x la variabile rispetto a cui derivare.
Esempio:

```
syms beta
syms x
f=@(x,beta)1/beta * exp(-x/beta);
syms L
for i=1:10
    L=L*f(tempi(i),beta);
end
der=diff(L,beta);
```

der conterrà la derivata di L rispetto a beta.
L, x e beta vanno necessariamente definiti come simboli(syms).

Risolvere un’equazione

La funzione solve permette presa in input un’equazione e la variabile da trovare di darci la risposta.
Riprendendo l’esempio di prima poniamo la derivata uguale a 0 e troviamo beta.
sol=solve(der==0,beta);
L’oggetto sol contiene il valore di beta.

Esempio sulla stima della verosimiglianza

```
tempi=[14,17,27,18,12,8,22,13,19,12];
l=length(tempi);
xbar=mean(tempi);
%L(theta) = produttoria da 1 a 10 di f(xi,beta)
% produttoria da 1 a 10 di 1/beta e^(-xi/beta)
syms beta
syms x
f=@(x,beta)1/beta * exp(-x/beta);
syms L
for i=1:10
    L=L*f(tempi(i),beta);
end

%calcolare derivata di L rispetto a beta
%diff(fun,variabile)
der=diff(L,beta);
%Poniamo la derivata uguale a 0
%Troviamo beta
soluzione_beta=solve(der==0,beta);
disp(soluzione_beta);
```

Con questo procedimento troviamo beta che massimizza la f.
Si è assunto i valori seguissero un’esponenziale.

Test e ipotesi

Posso o confermare o rifiutare un’ipotesi Ho.
Si usa una variabile aleatoria u detta statistica test che segue una certa distribuzione.
Usiamo anche un livello di significatività alpha.

Test su una sola popolazione

Test sulla media con varianza nota v:
La media campionaria Xbar segue una normale con media mu e varianza v
Si normalizza, la variabile u=(Xbar – mu_0)/sqrt(v/n) segue una normale standard.
Xbar = media campionaria(ottenuta col campione)
mu_0 = media teorica
I test si suddividono per tipo.
Tipo 1 : Ho indica mu=mu_0 , H1 indica mu=/=mu_0
Tipo 2 : Ho indica mu=mu_0 o mu<=mu_0 , H1 indica mu > mu_0
Tipo 3: Ho indica mu=mu_0 o mu>=mu_0 , H1 indica mu < mu_0
Se siamo nel tipo 1 rifiuto Ho se |u| > Z_1-a/2
tipo 2 rifiuto Ho se u > Z_1-a
tipo 3 rifiuto Ho se u < -Z_1-a
Z_1-x si ottiene con norminv(1-x,0,1);

Esempio:

```
H0: mu = 8 , H1: mu =/= 8
mu_0 = 8;
Xbar=7.8;
dev=0.5;
v=dev^2;
n=50;
alpha=0.01;

u=(Xbar - mu_0)/sqrt(v/n);
%Siamo nel tipo 1
%calcoliamo Z_1menoalphamezzi
Z_1menoalphamezzi = norminv(1-alpha/2,0,1);

if abs(u) > Z_1menoalphamezzi
    disp("Rifiuto H0");
else
    disp("Non rifiuto H0");
end
```

Test sulla media con varianza incognita:

Si calcola la varianza campionaria incognita Sn2, se abbiamo v (o dev) dai dati dati usiamo quella

La variabile u=(Xbar – mu_0)/sqrt(Sn2/n) segue una ti di Student con n-1 gradi di libertà

Xbar = media campionaria(ottenuta col campione)

mu_0 = media teorica

I test si suddividono per tipo.

Tipo 1 : Ho indica mu=mu_0 , H1 indica mu=/=mu_0

Tipo 2 : Ho indica mu=mu_0 o mu<=mu_0 , H1 indica mu > mu_0

Tipo 3: Ho indica mu=mu_0 o mu>=mu_0 , H1 indica mu < mu_0

Se siamo nel tipo 1 rifiuto Ho se |u| > t_1-a/2

tipo 2 rifiuto Ho se u > t_1-a

tipo 3 rifiuto Ho se u < -t_1-a

t_1-x si ottiene con tinv(1-x,n-1);

Test sulla varianza in una normale:

Tipo 1 : Ho indica sigma2=sigma2_0 , H1 indica sigma2=/=sigma2_0

Tipo 2 : Ho indica sigma2=sigma2_0 o sigma2<=sigma2_0 , H1 indica sigma2 > sigma2_0

Tipo 3: Ho indica sigma2=sigma2_0 o sigma2>=sigma2_0 , H1 indica sigma2 < sigma2_0

Caso con media mu nota:

Calcoliamo la varianza campionaria con media nota S2_0

La variabile u=(n/sigma2)*S2_0 segue una chi-quadro con n gradi di libertà

Caso con media mu incognita:

Calcoliamo la varianza campionaria con media incognita S2_n

La variabile u=((n-1)/sigma2)*S2_n segue una chi-quadro con n-1 gradi di libertà

Se siamo nel tipo 1 rifiuto Ho se u < X2_alphamezzi o u > X2_1menoalphamezzi

tipo 2 rifiuto Ho se u > X2_1-a

tipo 3 rifiuto Ho se u < X2_a

X2_1-b si ottiene con chi2inv(1-x,n-1);

Confronto tra 2 popolazioni

X e Y seguono 2 normali, con mu_x e sigma2_x e mu_y e sigma2_y.
Il campione di X è di n elementi, quello di Y è di m elementi.

Verifichiamo ipotesi sulla media con varianza nota

$$\bar{X}_n - \bar{Y}_m \sim Nor\left(\mu = \mu_x - \mu_y,\ \sigma^2 = \frac{\sigma_x^2}{n} + \frac{\sigma_y^2}{m}\right)$$

$$\frac{(\bar{X}_n - \bar{Y}_m) - (\mu_x - \mu_y)}{\sqrt{\frac{\sigma_x^2}{n} + \frac{\sigma_y^2}{m}}} \sim Nor(0,1)$$

Questa variabile che chiamiamo u
Segue una normale standard

NB: in alcuni caso la diff. mu_x – mu_y potremmo considerarla pari a 0 per l'ipotesi H0

Tipo	Rifiuto H_0 se
I	$\lvert u \rvert > z_{1-\alpha/2}$
II	$u > z_{1-\alpha}$
III	$u < -z_{1-\alpha}$

Tipo 1 : Ho indica mu_x=mu_y , H1 indica mu_x=/=mu_y
Tipo 2 : Ho indica mu_x=mu_y o mu_x<=mu_y , H1 indica mu_x > mu_y
Tipo 3: Ho indica mu_x=mu_y o mu_x>=mu_y , H1 indica mu_x < mu_y

Verifichiamo ipotesi sulla media con varianze incognite uguali

Calcoliamo v^ che è la varianza combinata, si scrive anche S2_p

$$\sigma_x^2 = \sigma_y^2 = v$$

$$S_x^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X}_n)^2$$

$$S_y^2 = \frac{1}{m-1} \sum_{i=1}^{m} (Y_i - \bar{Y}_m)^2$$

$$\hat{v} = S_p^2 = \frac{(n-1)S_x^2 + (m-1)S_y^2}{n+m-2}$$

Tipo 1 : Ho indica mu_x=mu_y , H1 indica mu_x=/=mu_y
Tipo 2 : Ho indica mu_x=mu_y o mu_x<=mu_y , H1 indica mu_x > mu_y
Tipo 3: Ho indica mu_x=mu_y o mu_x>=mu_y , H1 indica mu_x < mu_y

La statistica test è data da

$$U = \frac{\bar{X}_n - \bar{Y}_m}{\sqrt{S_p^2\left(\frac{1}{n} + \frac{1}{m}\right)}} \overset{H_0}{\sim} T(\nu = n+m-2)$$

Tipo	Rifiuto H_0 se
I	$\lvert u \rvert > t_{1-\alpha/2;\ n+m-2}$
II	$u > t_{1-\alpha;\ n+m-2}$
III	$u < -t_{1-\alpha;\ n+m-2}$

La variabile segue una t di student con n+m-2 gradi.

verifichiamo ipotesi sulla media con varianze incognite diverse

$$U = \frac{\bar{X}_n - \bar{Y}_m}{\sqrt{\frac{S_x^2}{n} + \frac{S_y^2}{m}}} \overset{H_0}{\sim} T(\nu = \nu^*)$$

dove $\nu^* = P.I\left(\frac{\left(\frac{S_x}{n} + \frac{S_y}{m}\right)^2}{\frac{1}{n-1}\left(\frac{S_x^2}{n}\right)^2 + \frac{1}{m-1}\left(\frac{S_y^2}{m}\right)^2}\right)$

Dove P.I. indica parte intera

La tabella sul rifiuto o meno dell'ipotesi rimane la stessa del caso in cui le varianze sono uguali

al numeratore Sx e Sy sono elevate al quadrato

2 popolazioni stessa lunghezza varianza incognita

X e Y seguono delle normali con mu_x e mu_y.
Definiamo W = X – Y , segue mu_w = mu_x – mu_y. Wbar_n è la media campionaria, S2_w è la varianza campionaria con valore atteso noto.

$W \sim Nor(\mu_w, \sigma_w^2)$ dove $\mu_w = \mu_x - \mu_y$, σ_w^2 incognita

Tipo I: $\begin{cases} H_0: \mu_w = 0 \\ H_1: \mu_w \neq 0 \end{cases}$

Tipo II: $\begin{cases} H_0: \mu_w = 0 \quad o \quad \mu_w \leq 0 \\ H_1: \mu_w > 0 \end{cases}$

Tipo III: $\begin{cases} H_0: \mu_w = 0 \quad o \quad \mu_w \geq 0 \\ H_1: \mu_w < 0 \end{cases}$

dove $S_w^2 = \frac{1}{n-1} \sum_{i=1}^{n} (W_i - \bar{W}_n)$

$$U = \frac{\bar{W}_n}{\sqrt{\frac{S_w^2}{n}}} \overset{H_0}{\sim} T(\nu = n-1)$$

Tipo	Rifiuto H_0
I	$\lvert u \rvert > t_{1-\alpha/2}$
II	$u > t_{1-\alpha}$
III	$u < -t_{1-\alpha}$

$; n-1$

Nella sommatoria c'è un ^2 nella parentesi

Esercizio sul confronto tra 2 popolazioni

Prima delle vacanze: 5 corse con tempo medio 53,82
Dopo le vacanze: 6 corse con tempo medio 54,41
Varianza = 0.1 (per entrambe)
alpha = 0.05 (5%)
I tempi sono distribuiti secondo delle Gaussiane

H0: mu_x = mu_y ovvero le vacanze non hanno influito (nel calo di prestazioni)
H1: mu_x < mu_y ovvero le vacanze hanno influito (nel calo di prestazioni)

Siamo nel tipo III. Supponiamo H0 e usiamo la formula della Gaussiana

```
u=(53.82-54.41)/sqrt(0.1/5 + 0.1/6);
```

Dato che siamo nel tipo III calcoliamo -Z_uno_meno_alpha

```
Z_uno_meno_alpha=norminv(1-0.05 , 0 , 1);
```

Valutiamo ora se non rifiutare o rifiutare H0 (l'abbiamo quindi prima supposta, ora con questo test effettivamente si capisce se è un'ipotesi corretta)

```
if u < -Z_uno_meno_alpha
    disp("Rifiuto H0");
else
    disp("Non rifiuto H0");
end
```

Codice completo:

```
%esercizio corsa
u=(53.82-54.41)/sqrt(0.1/5 + 0.1/6);
%disp(u);
Z_uno_meno_alpha=norminv(1-0.05 , 0 , 1);
%disp(Z_uno_meno_alpha);
%disp(-Z_uno_meno_alpha);
%usiamo -Z_uno_meno_alpha perchè siamo nel tipo III
if u < -Z_uno_meno_alpha
    disp("Rifiuto H0");
else
    disp("Non rifiuto H0");
end
```

Altro esercizio

```
%In uno studio condotto presso il dipartimento di foreste e fauna della
% Virginia Tech è stata esaminata leinfluenza di un farmaco sui livelli di
% androgeni nel sangue. A questo scopo sono stati catturati 15 cervi
% selvatici a cui sono stati prelevati campioni di sangue dopo aver
% ricevuto un iniezione intramuscolare del farmaco. Dopo 30 minuti dal
% primo prelievo è stato prelevato un secondo campione di sangue per ogni
% cervo che veniva immediatamente liberato. I livelli di androgeni al
% momento della cattura e dopo 30 minuti, misurati in nanogrammi per
% millilitro (ng/ml), sono riportati in tabella. Assumendo che le
% popolazioni dei livelli di androgeni al momento della somministrazione
% e 30 minuti dopo siano distribuite normalmente, si verifichi se le
% concentrazioni di androgeni sono alterate dopo 30 minuti a un livello di
% significatività di 0.05

%Le due popolazioni hanno stessa cardinalità e la varianza è incognita
%Definiamo un nuovo vettore dove nella posizione i sta Xi – Yi. Le popolazioni sono
prima30min e dopo30min prese da file
alpha=0.05;
W=[];
for i=1:length(prima30min)
    W = cat(1,W,prima30min(i)-dopo30min(i));
end
Wbar = mean(W);
n=length(W);
S2_w = 0.0;
for i=1:length(W)
    S2_w = (W(i)-Wbar)^2; %la parentesi va elevata al quadrato
end
S2_w = S2_w/(n-1);
u=Wbar/sqrt(S2_w/n);
t_1menoalphamezzi = tinv(1-alpha/2,n-1);
%siamo nel tipo 1
if abs(u) > t_1menoalphamezzi
    disp("Rifiuto l'ipotesi");
else
    disp("Non rifiuto l’ipotesi");
end
```

Regressione

$$\hat{y} = b_0 + b_1 x$$

$$b_1 = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$b_0 = \frac{\sum_{i=1}^{n} y_i - b_1 \sum_{i=1}^{n} x_i}{n} = \bar{y} - b_1 \bar{x}$$

Abbiamo $\hat{y}_j = \hat{\beta}_0 + \hat{\beta} x_j$

$r_j = y_j - \hat{y}_j = y_j - (\hat{\beta}_0 + \hat{\beta} x_j)$

Introduciamo una varianza

$\hat{v} = \frac{1}{n-2} \sum_j (y_j - \hat{y}_j)^2 = \frac{1}{n-2} \sum_j r_j^2$

polyfit(x,y,n); x e y sono vettori e n il nostro grado,se n è 1, si parla di regressione lineare,quello che è tornato è un vettore di n+1 elementi
y = B1x + B0
Il vettore tornato ha questi coefficienti, B1 e B0 in questo caso.
I coefficienti sono dal grado più alto a quello più basso
Nella cella i c'è il coefficiente del grado n-i

Note:

Ho due popolazione x e y su cui applicare la regressione(stessa cardinalità)

```
Sxx = 0.0;
Syy = 0.0;  E’ detta varianza totale
Sxy = 0.0;
xbar = mean(x);
ybar = mean(y);

for i = 1:1:length(x)
    Sxx = Sxx + (x(i) - xbar)^2;
    Syy = Syy + (y(i) - ybar)^2;
    Sxy = Sxy + (x(i) - xbar)*(y(i) - ybar);
end
```

Varianza residua: `SSres = ((Sxx*Syy)-(Sxy)^2)/Sxx;`
Varianza spiegata: `SSreg = (Sxy^2)/Sxx;`

```
B1 = Sxy/Sxx
B0 = ybar - B1xbar
v^ = SSres/(n-2)

plot(Speedmph,StoppingDistance,'o');
figure;
p=polyfit(Speedmph,StoppingDistance,1);
f=@(x)p(1)*x + p(2); %creiamo la funzione con i nostri coefficienti
%dopo aver importato il file
x=min(Speedmph)-1:0.1:max(Speedmph)+1;
plot(Speedmph,StoppingDistance,'o',x,f(x));
```

Vediamo in pratica 2 grafici, si capisce quanto si comporta bene la nostra retta

Coefficiente di determinazione

ci fa capire quanto la nostra retta approssima bene i dati

si indica con R2, R2=SSreg/Syy = 1 – Ssres/Syy = (Sxy/sqrt(Sxx*Syy))^2

```
clear x;
clear y;
x=Speedmph;
y=StoppingDistance;
xbar=mean(x);
ybar=mean(y);

Sxx=0.0;
Syy=0.0;
Sxy=0.0;
for i=1:1:length(x)
    Sxx=Sxx+(x(i)-xbar)^2;
    Syy=Syy+(y(i)-ybar)^2;
    Sxy=Sxy+(x(i)-xbar)*(y(i)-ybar);
end
R2=Sxy^2/(Sxx*Syy); %coefficiente di determinazione
disp(R2);
NOTA: COEFFICIENTE DI PEARSON E COEFFICIENTE DI CORRELAZIONE SONO LA STESSA COSA E SI
TROVANO CON LA FUNZIONE corrcoef , SONO LA sqrt(R2)
```

Calcolo di SSres, varianza residua

```
clear x;
clear y;
x=Height;
y=Pressure;
n=length(x);
xbar=mean(x);
ybar=mean(y);
plot(x,y);
figure;

p=polyfit(x,y,1);
f=@(a)p(1)*a + p(2);

array=min(x)-1:0.1:max(x)+1;
plot(array,f(array));

Sxx=0.0;
Syy=0.0;
Sxy=0.0;
for i=1:1:length(x)
    Sxx=Sxx+(x(i)-xbar)^2;
    Syy=Syy+(y(i)-ybar)^2;
    Sxy=Sxy+(x(i)-xbar)*(y(i)-ybar);
end
R2=Sxy^2/(Sxx*Syy); %coefficiente di pearson
SSres = (Sxx*Syy-Sxy^2)/Sxx;
R2corr=1-(SSres/(n-2))/(Syy/(n-1)); %coefficiente corretto
disp(R2);
disp(R2corr);
```

Calcolo del coefficiente corretto

R2corr = 1 – (SSres/(n-2))/(Syy/(n-1))

```
clear x;
clear y;
x=EfficiencyhiwayMpg;
y=MSRP;
n=length(x);
xbar=mean(x);
ybar=mean(y);
plot(x,y,'o');
figure;

p=polyfit(x,y,1);
f=@(a)p(1)*a + p(2);

array=min(x)-1:0.1:max(x)+1;
plot(array,f(array));
figure;
plot(x,y,'o',array,f(array));


Sxx=0.0;
Syy=0.0;
Sxy=0.0;

for i=1:1:length(x)
    Sxx=Sxx+(x(i)-xbar)^2;
    Syy=Syy+(y(i)-ybar)^2;
    Sxy=Sxy+(x(i)-xbar)*(y(i)-ybar);
end
R2=Sxy^2/(Sxx*Syy); %coefficiente di pearson
SSres = (Sxx*Syy-Sxy^2)/Sxx;
R2corr=1-(SSres/(n-2))/(Syy/(n-1)); %coefficiente corretto
disp("Il coefficiente è");
disp(R2);
disp("Il coefficiente corretto è");
disp(R2corr);
```

Test sui parametri beta

$$\frac{\hat{\beta}-\beta}{\sqrt{\frac{V}{S_{xx}}}} \sim Nor(\mu=0,\ \sigma^2=1)$$

V è incognita e consideriamo la stima $\hat{V}$

$$\frac{\hat{\beta}-\beta}{\sqrt{\frac{\hat{V}}{S_{xx}}}} \sim T(\nu=n-2)$$

Con beta^ si intende la stima di B1 in
y^ = B1x + B0

Con beta si intende il valore vero di B1

TEST DI IPOTESI:

1. Tipo 1:
   a. H0: B = B*
   b. H1: B =/= B*
2. Tipo 2:
   a. H0: B = B* or B <= B*
   b. H1: B > B*
3. Tipo 3:
   a. H0: B = B* or B >= B*
   b. H1: B < B*

VARIABILE STATISTICA TEST

$$U = \frac{\hat{\beta}-\beta^*}{\sqrt{\frac{\hat{V}}{S_{xx}}}} \overset{H_0}{\sim} T(\nu=n-2)$$

[redacted]

B* intende il valore “vero”