

Università di Bergamo – aa 2004/05

***APPUNTI DI MATLAB PER IL CORSO DI
STATISTICA***

***ANALISI STATISTICA DEI DATI CON MATLAB:
ANALISI DESCRITTIVA, REGRESSIONE LINEARE,
NON LINEARE E POLINOMIALE***

Orietta Nicolis

e-mail: orietta.nicolis@univr.it -

[http://dipinge.unibg.it/download/ nicolis/](http://dipinge.unibg.it/download/nicolis/)

Introduzione

□ Cos'è MATLAB?

Il MATLAB è un programma di calcolo numerico ed è nato sostanzialmente per le applicazioni matematiche. Nel corso degli anni sono state sviluppate una serie di moduli, le toolboxes, soddisfacendo così grande parte delle esigenze professionali dei diversi utenti.

□ Perché MATLAB ?

1. **dispone di una vasta gamma di toolbox:** in questi ultimi anni sono stati sviluppati numerosi pacchetti applicativi non solo nell'ambito matematico, ma anche statistico e ingegneristico che lo hanno reso un programma adatto sia a studenti che a professionisti di qualsiasi campo. Dispone quindi di una vasta gamma di funzioni e comandi che facilitano l'analisi dei dati. Ogni toolbox comprende inoltre una vasta gamma di *demos* che permettono all'utente un apprendimento più rapido dei comandi, delle funzioni e soprattutto delle potenzialità di tale programma.
2. **Semplice linguaggio di programmazione:** nell'ambiente MATLAB è possibile costruire i cosiddetti "m-file". Si chiamano così perché sono file ASCII scritti (con un qualsiasi editor) e hanno estensione *.m*. Quando vengono richiamati in matlab, il codice sorgente scritto esegue l'operazione per cui è stato ideato.
3. **Buona interfaccia grafica:**
 - a) è possibile riprodurre graficamente i dati, mostrando le loro caratteristiche statistiche in modo chiaro e semplice.
 - b) Esiste un linguaggio GUI che permette di 'interfacciare' le funzioni e personalizzare in questo modo il programma.

Per chiarire ulteriormente tali caratteristiche si propone il seguente esempio.

Esempio 1

Il gestore di una catena di negozi tessili ha deciso di studiare le vendite dei classici pullovers blu in dieci periodi. Indicando con X_1 il numero di pullovers venduti, X_2 la variazione del prezzo, X_3 i costi di pubblicità sui giornali locali e con X_4 la presenza di venditori (in ore per periodo), in dieci periodi osserva una seguente matrice $X = (X_1, X_2, X_3, X_4)$.

Determinare se il numero di pullovers venduti dipende dall'andamento dei prezzi.

Soluzioni

Innanzitutto si caricano i dati con la funzione **load** e si selezionano le prime due colonne che rappresentano rispettivamente le vendite e la variazione dei prezzi dei *pullovers blue*.

- `load pullovers .txt -ascii`
- `pullovers`
- `x1=x(:,1); % n° di pullovers venduti`
- `x2=x(:,2); % variaz. nel prezzo tra un periodo e l'altro.`

Ora, si utilizza la toolbox STATISTICS per determinare la media, la deviazione standard e la retta di regressione

- `xm = mean(x)`
- `std(x)`

È inoltre possibile determinare la retta di regressione mediante la funzione **regress** e rappresentarla graficamente.

- `y=x1;`
- `n=length(y);`
- `X=[ones(size(x2)) x2] ;`
- `[B, Bint, Resid, Rint, Stats]=regress(y, X, alpha);`
- `statistiche=Stats`
- `beta=B`
- `yhat=B(1)+B(2).*x2;`
- `subplot(2,1,1)`
- `plot(1:10, yhat,'*- ', 1:10, y,'.-')`
- `legend('dati stimati', 'dati osservati')`
- `subplot(2,1,2)`
- `plot(1:10, Resid,'o-')`
- `legend('Residui')`

La stessa cosa può essere ottenuta costruendo un file **.m**, per esempio

- `pull12_reg`

o con un file **.fig** (mediante il comando **guide**),

- `pull_reg`

DISTRIBUZIONI DI PROBABILITÀ

MATLAB fornisce 5 funzioni per l'analisi di ciascuna distribuzione:

- Funzioni di distribuzione di probabilità (pdf);
- Funzione di distribuzione di probabilità cumulata o di ripartizione (cdf);
- Funzione di distribuzione di probabilità cumulata inversa;
- Generatore di numeri casuali;
- Media e varianza.

Vi è inoltre un'ulteriore funzione che stima i parametri delle distribuzioni, ma che attualmente non è ancora disponibile per tutte le distribuzioni.

N. B. La demo **disttool** permette un'interazione grafica con le varie distribuzioni di probabilità.

<i>Distribuzioni</i>	<i>Funzioni di densità di probabilità (pdf)</i>	<i>Funzioni cumulate (cdf)</i>	<i>Funzioni cumulate inverse (cdf)</i>	<i>Momenti delle distrib. (media e varianza)</i>	<i>Generatori di numeri casuali</i>	<i>Stima dei parametri</i>
<i>Beta</i>	betapdf	betacdf	betainv	betastat	betarnd	betafit\betalike
<i>Binomiale</i>	binopdf	binocdf	binoinv	binostat	binornd	binofit
<i>Chi-square</i>	chi2pdf	chi2cdf	chi2inv	chi2stat	chi2rnd	
<i>Chi-square non cen.</i>	ncx2pdf	ncx2cdf	ncx2inv	ncx2stat	ncfrnd	
<i>Discreta uniforme</i>	unidpdf	unidcdf	unidinv	unidstat	unidrnd	
<i>Esponenziale</i>	exppdf	expcdf	expinv	expstat	exprnd	expfit
<i>F</i>	fpdf	fcdf	finv	fstat	frnd	
<i>F non centr.</i>	ncfpdf	ncfcdf	ncfinv	ncfstat	ncfrnd	
<i>Gamma</i>	gampdf	gamcdf	gaminv	gamstat	gamrnd	gamfit/gamlike
<i>Geometrica</i>	geopdf	geocdf	geoinv	geopdf	geornd	normfit/normlike
<i>Ipergeometrica</i>	hygepdf	hygecdf	hygeinv	hygestat	hygernd	
<i>Lognormale</i>	lognpdf	logncdf	logninv	lognstat	lognrnd	
<i>Binomiale negativa</i>	nbinopdf	nbinocdf	nbinoinv	nbinostat	nbinrnd	
<i>Normale</i>	normpdf	normcdf	norminv	normstat	normrnd	
<i>Poisson</i>	poisspdf	poisscdf	poissinv	poisstat	poissrnd	poissfit

<i>Reyleigh</i>	raylpdf	raylcdf	raylinv	raylstat	raylrnd	
<i>T di Student</i>	tpdf	tcdf	tinvs	tstat	trnd	
<i>T di Student non c.</i>	nctpdf	nctcdf	nctinv	nctstat	nctrnd	
<i>Uniforme</i>	unifpdf	unifcdf	unifinv	Unifstat	unifrnd	unifit
<i>Weibull</i>	weibpdf	weibcdf	weibinv	weibstat	weibrnd	

1. Funzioni di probabilità (pdf)

Naturalmente, le funzioni di probabilità hanno un significato diverso a seconda che ci si riferisca a variabili casuali discrete o continue: nel discreto la pdf è la probabilità di osservare un dato valore, mentre nel continuo tale probabilità è uguale a zero e quindi rappresenta la probabilità che un dato valore sia contenuto in un certo intervallo (si esegue l'integrale tra due valori).

La funzione pdf di MATLAB ha un formato generale e quindi non distingue il caso discreto dal continuo.

Per esempio, per determinare la funzione di probabilità di una variabile casuale **binomiale**, con parametri $n = 10$ e $p = 0.5$, per tutti i valori da 0 a 10, equivale a determinare i valori della funzione

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x} \quad x = 0, 1, \dots, 10$$

In *MATLAB*:

- `x=0:10;`
- `y=binopdf(x, 10, 0.5);`
- `bar(y)`

Allo stesso modo per determinare la funzione di densità di probabilità di una variabile casuale **normale**, con media con $\mu = 5$ e scarto quadratico medio $\sigma = 0.8$, nell'intervallo $[0;10]$ equivale a determinare i valori della funzione

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

per $0 \leq x \leq 10$.

In *MATLAB*:

- `x=0:.1 :10;`
- `y=normpdf(x, 5, 0.8);`
- `plot(x,y)`

Altri esempi :

a) Distribuzione ***chi-quadrato*** con 4 gradi di libertà

- `x=0:0.2:15;`
- `y=chi2pdf(x,4);`
- `plot(x,y)`

b) Distribuzione ***chi-quadrato non centrata*** con 4 gradi di libertà

- `x=0:0.1:10;`
- `p1=ncx2pdf(x,4,2);`
- `p=chi2pdf(x,4);`
- `plot(x,p,'—',x,p1,'-')`

c) Distribuzione ***F***

- `x=0:0.1:10;`
- `y=fpdf(x,5, 3);`
- `plot(x,y)`

d) Distribuzione ***F non centrata***

- `x=(0.01:0.1:10.01);`
- `p1=ncf2pdf(x,5,20,10);`
- `p=fpdf(x,5, 20);`
- `plot(x,p,'—',x,p1,'-')`

e) Distribuzione ***Ipergeometrica***

- `x=0 :10;`
- `y=hygecdf(x, 1000, 50, 20);`
- `stairs(x,y)`

f) Distribuzione ***Lognormale***

- `x=(10 :1000 :125010)' ;`
- `y=lognpdf(x, log(20000), 1.0);`
- `plot(x,y)`
- `set(gca, 'Xtick', [0 30000 60000 90000 120000])`
- `set(gca, 'xticklabel', str2mat('0', '$30.000', '$60.000', '$90.000', '$120.000'))`

g) Distribuzione ***Binomiale Negativa***

- `x=(0 :10);`
- `y=nbinpdf(x, 0.5);`
- `plot(x,y,'+')`
- `set(gca, 'XLim', [-0.5 10.5])`

e) Distribuzione di ***Rayleigh***

- `x=[0:0.01:2];`
- `p=raylpdf(x, 0.5);`
- `plot(x,p)`

f) Distribuzione di ***t-Student***

- `x=-5 :0.1:5;`
- `y=tpdf(x,5);`
- `z=normpdf(x, y,'-', x,z, '-.')`

2. Funzioni di probabilità cumulata o di ripartizione (pdf)

La funzione di ripartizione di una variabile casuale continua X è data da

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t)dt$$

mentre, la funzione di ripartizione di una variabile casuale discreta X è

$$F(x) = P(X \leq x) = \sum_{t \leq x} p(t)$$

Come per la funzione di distribuzione di probabilità, la funzione di MATLAB **cdf** (*cumulative distribution function*) esplora entrambi i casi.

Vediamo ora alcuni esempi di funzioni di ripartizione:

a) Distribuzione ***chi-quadrato*** con 4 gradi di libertà,

- `x=0:10;`
- `y=unidcdf(x, 10);`
- `stairs(x,y)`
- `set(gca, 'xlim',[0 11])`

b) Distribuzione ***geometrica***

- `x=0:25;`
- `y=geocdf(x, 0.03);`
- `stairs(x,y)`

c) Distribuzione ***Ipergeometrica***

- `x=0 :10;`
- `y=hygecdf(x, 1000, 50, 20);`
- `stairs(x,y)`

d) distribuzione di ***Poisson*** per $\lambda=5$.

- `x=0:15;`
- `y=poisscdf(x,5);`
- `plot(x, y,'+')`

e) Distribuzione di ***t-Student*** non centrata

- `x=(-5 :0.1:5)';`
- `p1=nctcdf(x, 10,1);`
- `y=tcdf(x,10);`
- `plot(x, p,'-', x,p1, '-.')`

f) La distribuzione ***Uniforme Discreta*** con 4 gradi di libertà

- `x=0:10;`
- `y=unidcdf(x, 10);`
- `stars(x,y)`
- `set(gca, 'xlim',[0 11])`

3. Funzione di distribuzione di probabilità cumulata inversa

La funzione di distribuzione di probabilità cumulata inversa determina i valori critici per i tests d'ipotesi, data la probabilità significativa.

Per esempio, per determinare il valore critico di una distribuzione normale standard corrispondente ad un livello $\alpha=0.025$, si procede come segue

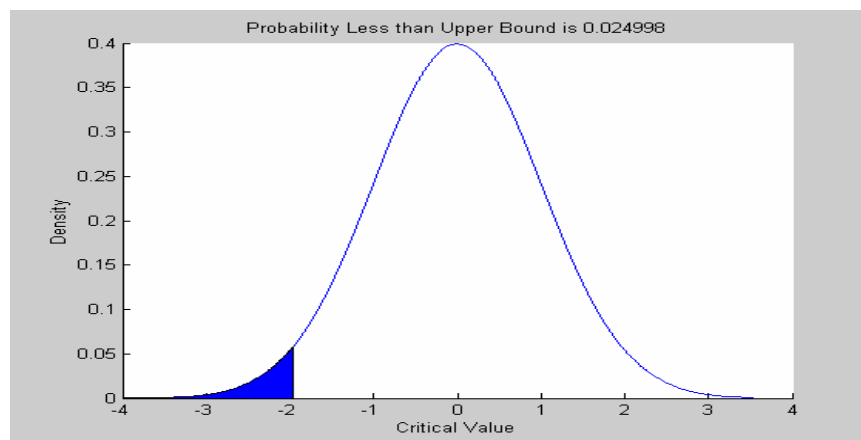
- `xc = norminv(0.025,0,1);`

oppure

- `xc=norminv(normcdf(-1.96,0,1),0,1)`

Il risultato è $xc = 1.96$ che è possibile rappresentare graficamente con la funzione **normspec**:

- `normspec([-Inf -1.96], 0,1)`



Per le

distribuzioni discrete, trovare una relazione tra una cdf e la sua inversa diventa più complicato in quanto si può verificare che non esista alcun valore di x tale che la cdf di x sia la probabilità p . In questo caso la funzione inversa, trova il primo valore di x tale che la cdf di x sia uguale o maggiore

di p . Per esempio, per determinare il valore critico di una distribuzione binomiale con $n = 10$ e $p = 0.5$, corrispondente ad un livello $\alpha=0.025$, si digita

- `xc = binoinv(0.025,10,0.5)`

Il risultato è $xc = 2$ che corrisponde precisamente ad un $\alpha=0.0547$,

4. Generatori di numeri casuali;

Le funzioni che terminino con **rnd** generano matrici di numeri casuali da ciascuna distribuzione.

Per esempio:

- `r = betarnd(5, 0.2, 100, 2);`

genera una matrice (100x2) da una distribuzione **beta** con parametri $a = 5$ e $b = 0.2$.

- `r = binornd(100, 0.9, 20, 3);`

genera una matrice (20x3) da una distribuzione **binomiale** con $n=100$ e $p=0.9$.

- `numbers = unidrnd(250, 1,10)`

genera una matrice (1x10) da una distribuzione **discreta uniforme** con $x = 1,2,\dots,250$.

- `lifetimes=exprnd(700, 100,1)`

genera una matrice (100x1) da una distribuzione **Esponenziale** con $\lambda = 700$.

- `lifetimes=gamrnd(10, 5, 100,1);`

genera una matrice (100x1) da una distribuzione **Gamma** con $a = 10$ e $b = 5$.

- `height=normrnd(50,2,30,1);`

genera una matrice (30x1) da una distribuzione **Normale** con $\mu = 50$ e $\sigma = 2$.

- `strength=weibrnd(0.5, 2, 100,1);`

genera una matrice (100x1) da una distribuzione di **Weibull** con $a=10$ e $b = 5$.

N.B.1. Le funzioni **rand** e **randn** generano matrici rispettivamente da una distribuzione uniforme con valori nell'intervallo $[0; 1]$ e da una distribuzione normale standard con $\mu = 0$ e $\sigma = 1$. Per esempio,

- `u=rand(1000,1);`
- `u=randn(1000,1);`

N. B.2. La demo **randtool** permette di generare numeri casuali da ciascuna distribuzione in modo iterativo.

N. B.3. La funzione **mvnrnd** permette di generare numeri casuali da una distribuzione normale multivariata.

5. Media e varianza

Le funzioni di MATLAB che terminano con **stat** determinano media e varianza delle distribuzioni specificate, dati i parametri. Per esempio,

- `[mu sigma]=normstat(2,1.2)`

determina la media $\mu = 0$ e la varianza $\sigma^2 = 1.44$ di una distribuzione normale con parametri $\mu = 0$ e $\sigma = 1.2$.

6. Stima dei parametri

Le funzioni **betafit/betalike**, **binofit**, **expfit**, **gamfit/gamlike**, **normfit/normlike**, **poissfit** e **unifit** determinano le stime dei parametri e gli intervalli di confidenza per i dati delle derivanti dalle corrispondenti distribuzioni di probabilità. Per esempio:

- `r = binornd(100,0.9);`
- `[phat, pci]=binofit(r, 100)`

genera una distribuzione **binomiale** con $n=100$ e $p=0.9$ e produce le stime MLE e gli intervalli di confidenza dei parametri.

- `r = betarnd(5, 0.2, 100, 1);`
- `[phat, pci]=betafit(r);`

genera una distribuzione **Beta** con $a=5$ e $b=0.2$ e produce le stime MLE e gli intervalli di confidenza dei parametri.

- `lifetimes=gamrnd(10, 5, 100,1);`
- `[phat, pci]=gamfit(lifetimes)`

genera una distribuzione **Gamma** con $a=10$, $b=5$ e produce le stime MLE e gli intervalli di confidenza dei parametri.

- `height=normrnd(50,2,30,1);`
- `[mu, s, muc1, sci]=normfit(height)`

genera una distribuzione **Normale** con $\mu = 50$ e $\sigma = 2$ e produce le stime MLE e gli intervalli di confidenza dei parametri.

N.B. La funzione **mle** stima i parametri di ciascuna distribuzione con il metodo della massima verosimiglianza. Per esempio,

- `lifetimes=gamrnd(10, 5, 100,1);`

- `[phat, pci]=mle('gam', lifetimes)`

TEST D'IPOTESI

Test sulla media

Esempio: prezzo della benzina **gas.mat** → ci sono 2 campioni di 20 osservazioni

- `load gas`
- `prices=[price1 price2]`

Verifichiamo che il prezzo medio sia \$1.15, sapendo che la deviazione std. È 0.04.

- `[h, pvalue, ci]=ztest(price1/100, 1.15, 0.04)`

Quando $h = 0$, si accetta l'ipotesi nulla, quando è uguale ad 1 la si rifiuta.

Supponendo di non conoscere la deviazione standard del price2, si applica il **ttest**

- `[h, pvalue, ci]=ttest(price2/100, 1.15)`

La funzione `ttest2` ci consente di verificare se vi è una differenza significativa tra le medie dei due campioni.

- `[h, sig, ci]=ttest(price1, price2)`

ESERCIZI SULLE DISTRIBUZIONI DI PROBABILITÀ

Esercizio 1

Si supponga che un particolare processo produttivo produca pezzi difettosi con probabilità p . Ci si chiede qual è la probabilità che su n pezzi prodotti ce ne siano x difettosi. Dopo aver individuato di quale distribuzione si tratta, costruire una funzione in MATLAB, chiamata **diffettosi.m** che determini per qualsiasi valore di n e p ,

- a) la distribuzione di probabilità di x ;
- b) la funzione di ripartizione di x ;
- c) la media e la varianza della v.c. x ;
- d) la rappresentazione grafica dei punti a) e b).

Esercizio 2

Costruire una funzione in MATLAB che determini i valori di una distribuzione standard bivariata, con argomenti x , y e ρ .

Rappresentare graficamente tale funzione e le ellissi di confidenza.

Esercizio 3

Generare un campione casuale da distribuzione normale bivariata con media $\mu = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ e matrice di

varianza-covarianza $\sigma = \begin{pmatrix} 1 & -1.5 \\ -1.5 & 4 \end{pmatrix}$. Dare la rappresentazione grafica del campione e delle ellissi di confidenza della distribuzione.

FUNZIONI STATISTICHE DI BASE PER L'ANALISI DESCRITTIVA

MISURE DI TENDENZA CENTRALE

Codice	DESCRIZIONE
geomean	Media geometrica
harmmean	Media armonica
mean	Media aritmetica
median	50° percentile
trimmean	Media cubica

MISURE DI DISPERSIONE

Codice	DESCRIZIONE
iqr	Range interquantile
mad	Deviazione media assoluta
range	range
std	Deviazione standard
var	Varianza
corrcoef	Coefficiente di correlazione lineare

Esercizio 1

Calcolare i quartili di un campione formato dalla mistura di due distribuzioni normali con medie e varianze pari a $\mu_1 = 3$ e $\sigma_1^2 = 1$ per la prima distribuzione e $\mu_2 = 5$ e $\sigma_1^2 = 0.5$. Scegliere inoltre una rappresentazione grafica per tali statistiche.

Esercizio 2

Si considerino i rendimenti giornalieri del gruppo UNICREDITO.

- Determinare la media, varianza, la simmetria e la curtosi.
- Eliminare eventuali outliers.
- Rappresentare graficamente la distribuzione di probabilità stimata;

Esercizio 3

Considerare l'esempio 1 (parte introduttiva).

- Stimare il coefficiente di correlazione lineare tra la variabile X_1 e la variabile X_2 .
- Utilizzare la funzione **bootstrap** per ricampionare i dati e determinare la distribuzione del coefficiente di correlazione

MODELLI LINEARI

$$y = X\beta + \varepsilon$$

- y = vettore delle osservazioni $n \times 1$
- X = matrice di disegno per il modello $n \times p$
- β = vettore dei parametri $p \times 1$
- ε = vettore dei disturbi casuali $n \times 1$

Casi specifici del modello lineare sono:

- 1) Analisi della varianza ad 1 via (ANOVA);
- 2) ANOVA a 2 vie;
- 3) Regressione polinomiale
- 4) Regressione lineare multipla.

1. Analisi della varianza ad 1 via (ANOVA);

Lo scopo è di trovare se i dati di diversi gruppi hanno una media comune, cioè determinare se i gruppi sono effettivamente diversi nelle caratteristiche misurate.

L'ANOVA ad una via è un caso particolare del modello lineare

$$y_{ij} = \alpha_{.j} + \varepsilon_{ij}$$

dove

y_{ij} è la matrice delle osservazioni;

$\alpha_{.j}$ è la matrice le cui colonne sono le medie di gruppo

ε_{ij} è la matrice dei disturbi casuali

Il modello stabilisce che le colonne di y sono una costante più un disturbo casuale. Si vuole sapere se le costanti sono tutte uguali.

Es. Le colonne della matrice **hogg** rappresentano il numero di batteri nelle spedizioni di latte. Le righe sono il numero di batteri da cartoni di latte scelti casualmente da ciascun da ciascuna spedizione. Alcune spedizioni hanno un numero di batteri più alto di altre?

- `load hogg`
- `p=anova1(hogg)`

è possibile utilizzare la statistica F per eseguire un test d'ipotesi al fine di trovare se il numero di batteri è lo stesso. **anova1** riporta il p-value.

In questo caso il p-value è circa 0.0001, molto piccolo. Questa è una forte indicazione che il numero di batteri non è lo stesso nelle diverse spedizioni.

2. ANOVA a 2 vie;

Supponiamo che ci siano due imprese automobilistiche che producono entrambe 3 tipi di auto. Verifichiamo se il consumo di carburante nelle auto varia da fabbrica a fabbrica. C'è inoltre un consumo che dipende dal modello (indipendentemente dalla fabbrica) dovuto a differenze nella specificazione del disegno. Inoltre, una fabbrica potrebbe avere auto con un alto consumo (forse perché appartenente ad una linea di produzione superiore) in un modello, ma non essere diversa dall'altra fabbrica per gli altri modelli. Questo effetto è chiamato interazione. Il modello è

$$y_{ijk} = \mu + \alpha_{.j} + \beta_{i.} + \gamma_{ij} + \varepsilon_{ijk}$$

dove

y_{ijk} è la matrice delle osservazioni;

μ è una matrice costante di tutte le medie

$\alpha_{.j}$ è la matrice le cui colonne sono le medie di gruppo

$\beta_{i.}$ è la matrice le cui righe sono le medie di gruppo

γ_{ij} è una matrice delle interazioni (la somma per riga e colonna da 0)

ε_{ijk} è la matrice dei disturbi casuali

Lo scopo di questo esempio è di determinare l'effetto del modello di auto e l'effetto fabbrica sul consumo.

- load mileage
- cars=3;
- p=anova2(mileage, cars)

3. Regressione lineare multipla

Lo scopo è di stabilire una relazione quantitativa tra un gruppo di variabili predittive e la risposta y.

Questo modello è utile per:

- capire quali predittori hanno più effetto;
- Conoscere la direzione dell'effetto (crescente o decrescente con y);

- Usare il modello per prevedere i valori futuri della risposta quando si conoscono solo i predittori.

Il modello lineare ha la seguente forma

$$y = X\beta + \varepsilon$$

- y = vettore delle osservazioni $n \times 1$
- X = matrice dei regressori $n \times p$
- β = vettore dei parametri $p \times 1$
- ε = vettore dei disturbi casuali $n \times 1$

Es. Il dataset *moore* ha 5 variabili previsive ed 1 risposta

- `load moore`
- `X=[ones(size(moore,1),1) moore(:,1:5)];`
- `y=moore(:,6);`
- `[b, bint, r, rint,stats]=regress(y,X);`
- `stats`
- `rcoplot(r, rint)`

b = parametri ($b(1)$ è l'intercetta);

$stats$ = statistica R^2 dei repressori , statistica F e p-value

Modelli polinomiali (Response Surface Methodology)

Si vuole capire la relazione quantitativa tra fra variabili di input multipla e una variabile di output.

La funzione **rstool** è utile per stimare modelli di risposta a superficie.

- `load reaction`
- `rstool(reactants, rate, 'quadratic',0.01, xn,yn)`

La variabile risposta è “rate” ossia il grado di reazione che è funzione di tre reagenti, “reactants”: idrogeno, n-pentane, isopentane.

Vedremo un vettore di tre grafici. La variabile dipendente di tutti e tre i grafici è il tasso di reazione (rate). Il primo grafico ha l'idrogeno come variabile indipendente. Il secondo ed il terzo hanno rispettivamente l'n- pentane e l'isopentane.

Ciascun grafico mostra la relazione stimata del tasso di reazione alla variabile indipendente in un valore fisso delle altre due variabili indipendenti. Il valore fisso di ciascuna variabile indipendente è messo in una mascherina che si può cambiare.

Regressione stepwise

È una tecnica per scegliere le variabili da includere nel modello di regressione multipla. La regressione stepwise in avanti (forward) inizia con un modello con nessun termine. Ad ogni passo si aggiunge il termine statisticamente più significativo (quelli con la statistica F più grande o il più basso p-value) finché non ce ne sono più. La regressione stepwise indietro (backward) inizia con tutti i termini nel modello ed elimina quelli meno significativi finché quelli che rimangono sono tutti statisticamente significativi.

Un problema comune nell'analisi di regressione multipla è la *multicollinearità* delle variabili di input. In questo caso il metodo stepwise potrebbe rivelarsi pericoloso.

Esempio,

- `load hald`
- `stepwise(ingredients, heat)`

MODELLI DI REGRESSIONE NON LINEARI

I modelli di regressione non lineare sono più difficili da stimare e richiedono e richiedono metodi iterativi che partono con valori iniziali dei parametri ignoti. Ciascuna iterazione va a variare ciascun parametro finché l'algoritmo converge.

$$y = f(X, \beta) + \varepsilon$$

dove:

- y = vettore delle osservazioni $n \times 1$
- f è una qualsiasi funzione di X e β
- X = matrice $n \times p$ di variabili di input
- β = vettore dei parametri $p \times 1$
- ε = vettore dei disturbi casuali $n \times 1$

Esempio: Prendiamo il modello di Hougen – Watson per stimare il tasso di reazione.

- load reaction
- who
- `betahat=nlinfit(reactants, rate, 'hougen', beta)`

nlinfit ha due output opzionali: i residui e la matrice Jacobiana della soluzione. Questi output sono utili per ottenere degli intervalli di confidenza sulla stima dei parametri.

Intervalli di confidenza sulla stima dei parametri.

Si usa **nlparci** per calcolare intervalli di confidenza di 95% sulla stima dei parametri e la previsione delle risposte.

- `[betahat, f, J]=nlinfit(reactants, rate, 'hougen', beta);`
- `betaci=nlparci(betahat, f, J)`

Intervalli di confidenza sulle risposte previste

Si usa **nlpredci** per calcolare intervalli di confidenza di 95% sulle risposte previste

- `[yhat, delta]=nlpredci('hougen', reactants, betahat, f, J);`
- `opd=[rate yhat delta]`

GUI per la stima non lineare e la previsione

La funzione **nlintool** per modelli non lineari è l'analogo di **rstool** per i modelli polinomiali.

- `nlintool(reactants, rate, 'hougen', beta, 0.01, xn, yn);`

DEMOS

Codice	DESCRIZIONE
aocool	Previsione grafica iterativa delle stime anocova
disttool	Iterazione grafica con distribuzione di probabilità
glmdemo	Modelli lineari generalizzati
nlintool	Fittine iterativo dei modelli non lineari
polytool	Previsione grafica iterativa dei modelli polinomiali
randtool	Controllo iterativo della generazione dei numeri casuali
robustdemo	Confronto iterativo delle stime robuste e ai minimi quadrati
rsmdemo	Disegni degli esperimenti e modelli di regressione
rstool	Esplorazione dei grafici dei polinomi multidimensionali
stepwise	Regressione stepwise iterativa