MatLab: Statistica Inferenziale Non Parametrica

Centro E. Piaggio
bioengineering and robotics research center

IN SUPREMÆ DIGNITATIS
1343

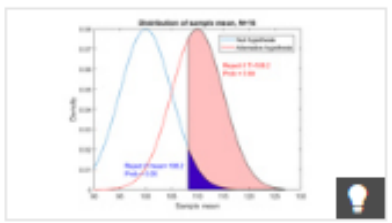
Ing. Vincenzo Catrambone
vincenzo.catrambone@ing.unipi.it

Test delle ipotesi

La variazione casuale può rendere difficile determinare se i campioni prelevati in diverse condizioni sono effettivamente differenti. Il test delle ipotesi è uno strumento efficace per analizzare se le differenze da campione a campione sono significative e se richiedono ulteriore valutazione o se sono coerenti con la casualità e la variazione attesa dei dati.

Statistics and Machine Learning Toolbox supporta ampiamente le procedure di test delle ipotesi parametriche e non parametriche, tra cui:

- *T*-test per uno o due campioni
- test non parametrici per un campione, campioni accoppiati e due campioni indipendenti;
- test di distribuzione (chi-quadrato, Jarque-Bera, Lilliefors e Kolmogorov-Smirnov);
- confronto delle distribuzioni (Kolmogorov-Smirnov per due campioni);
- test per autocorrelazione e casualità;
- test delle ipotesi lineari su coefficienti di regressione.



Selezione della dimensione di un campione (Esempio)
Calcola la dimensione del campione necessaria per un test di ipotesi.

```
>> help kstest
 kstest Single sample Kolmogorov-Smirnov goodness-of-fit hypothesis test.
    H = kstest(X) performs a Kolmogorov-Smirnov (K-S) test to determine if
    a random sample X could have come from a standard normal distribution,
    N(0,1). H indicates the result of the hypothesis test:
       H = 0 => Do not reject the null hypothesis at the 5% significance
       level.
       H = 1 => Reject the null hypothesis at the 5% significance
       level.
Parameter       Value
'alpha'         A value ALPHA between 0 and 1 specifying the
                significance level. Default is 0.05 for 5% significance.
'CDF'           CDF is the c.d.f. under the null hypothesis.  It can
                be specified either as a ProbabilityDistribution object
                or as a two-column matrix. Default is the standard
                normal, N(0,1).
```

Gaussianity test, or in general, Goodness-of-fit test

Other test functions: lillietes; jbtest; adtest…

```
>> help chi2gof
 chi2gof   Chi-square goodness-of-fit test.
    chi2gof performs a chi-square goodness-of-fit test for discrete or
    continuous distributions.  The test is performed by grouping the data into
    bins, calculating the observed and expected counts for those bins, and
    computing the chi-square test statistic SUM((O-E).^2./E), where O is the
    observed counts and E is the expected counts.  This test statistic has an
    approximate chi-square distribution when the counts are sufficiently
    large.
```

signtest Sign test for zero median.

P = **signtest**(X) performs a two-sided sign test of the hypothesis that the data in the vector X come from a distribution whose median is zero, and returns the p-value from the test. P is the probability of observing the given result, or one more extreme, by chance if the null hypothesis ("median is zero") is true. Small values of P cast doubt on the validity of the null hypothesis. The data are assumed to come from an arbitrary continuous distribution. **signtest** omits values where X is zero or NaN.

P = **signtest**(X,M) performs a two-sided test of the hypothesis that the data in the vector X come from a distribution whose median is M. M must be a scalar. **signtest** omits values where X is M or NaN.

```
>> help ranksum
```

ranksum Wilcoxon rank sum test for equal medians.

P = **ranksum**(X,Y) performs a two-sided rank sum test of the hypothesis that two independent samples, in the vectors X and Y, come from distributions with equal medians, and returns the p-value from the test. P is the probability of observing the given result, or one more extreme, by chance if the null hypothesis ("medians are equal") is true. Small values of P cast doubt on the validity of the null hypothesis. The two sets of data are assumed to come from continuous distributions that are identical except possibly for a location shift, but are otherwise arbitrary. X and Y can be different lengths. **ranksum** treats NaNs in X or Y as missing values, and removes them. The two-sided p-value is computed by doubling the most significant one-sided value.

The Wilcoxon rank sum test is equivalent to the Mann-Whitney U test.

P = **signrank**(X,Y) performs a paired, two-sided test of the hypothesis that the difference between the matched samples in the vectors X and Y comes from a distribution whose median is zero. The differences X-Y are assumed to come from a continuous distribution, symmetric about its median. X and Y must be the same length. The two-sided p-value is computed by doubling the most significant one-sided value.

kruskalwallis Nonparametric one-way analysis of variance (ANOVA).

P = **kruskalwallis**(X,GROUP,DISPLAYOPT) performs a non-parametric one-way ANOVA to test the null hypothesis that independent samples from two or more groups come from distributions with equal medians, and returns the p-value for that test. The data are assumed to come from continuous distributions that are identical except possibly for location shifts due to group effects, but are otherwise arbitrary.

If X is a matrix, **kruskalwallis** treats each column as coming from a separate group. This form of input is appropriate when each sample has the same number of elements (balanced). GROUP can be a character array or a cell array of strings, with one row per column of X, containing the group names. Enter an empty array ([]) or omit this argument if you do not want to specify group names.

If X is a vector, GROUP must be a categorical variable, vector, string array, or cell array of strings with one row for each element of X. X values corresponding to the same value of GROUP are placed in the same group.

DISPLAYOPT can be 'on' (the default) to display figures containing a boxplot and the Kruskal-Wallis version of a one-way ANOVA table, or 'off' to omit these displays.

```
friedman Nonparametric two-way analysis of variance.
   P = friedman(X,REPS,DISPLAYOPT) performs Friedman's test, a
   nonparametric version of balanced two-way ANOVA.  friedman compares
   columns of data in X, adjusting for possible row effects, and returns
   the p-value for the null hypothesis that there are no column effects.
   The data are assumed to be independent samples from continuous
   distributions that are identical except possibly for location shifts
   due to row and column effects, but are otherwise arbitrary.  If there
   is more than one observation per row-column "cell", use the scalar
   argument REPS to indicate the number of observations per cell.  Each
   cell corresponds to REPS consecutive rows in one column of X.
   DISPLAYOPT can be 'on' (the default) to display the table, or 'off' to
   skip the display.

      COMPARISON = multcompare(STATS) performs a multiple comparison
      using a STATS structure that is obtained as output from any of
      the following functions:  anova1, anova2, anovan, aoctool,
      kruskalwallis, friedman.  The return value COMPARISON is a matrix
      with one row per comparison and six columns.  Columns 1-2 are the
      indices of the two samples being compared.  Columns 3-5 are a lower
      bound, estimate, and upper bound for their difference. Column 6 is the
      p-value for each individual comparison.
```

```
polyfit Fit polynomial to data.
    P = polyfit(X,Y,N) finds the coefficients of a polynomial P(X) of
    degree N that fits the data Y best in a least-squares sense. P is a
    row vector of length N+1 containing the polynomial coefficients in
    descending powers, P(1)*X^N + P(2)*X^(N-1) +...+ P(N)*X + P(N+1).
    [P,S] = polyfit(X,Y,N) returns the polynomial coefficients P and a
    structure S for use with POLYVAL to obtain error estimates for
    predictions.  S contains fields for the triangular factor (R) from a QR
    decomposition of the Vandermonde matrix of X, the degrees of freedom
    (df), and the norm of the residuals (normr).  If the data Y are random,
    an estimate of the covariance matrix of P is (Rinv*Rinv')*normr^2/df,
    where Rinv is the inverse of R.
```

- Use polyfit to compute a linear regression that predicts y from x:

  p = polyfit(x,y,1)

- Call polyval to use p to predict y, calling the result yfit:

  yfit = polyval(p,x);

  yfit = p(1) * x + p(2);

- Compute the residual values as a vector of signed numbers:

  yresid = y - yfit;

- Square the residuals and total them to obtain the residual sum of squares:

  SSresid = sum(yresid.^2);

- Compute the total sum of squares of y by multiplying the variance of y by the number of observations minus 1:

  SStotal = (length(y)-1) * var(y);

- Compute R2 using the formula:

  rsq = 1 - SSresid/SStotal

```
polyval Evaluate polynomial.
    Y = polyval(P,X) returns the value of a polynomial P evaluated at X. P
    is a vector of length N+1 whose elements are the coefficients of the
    polynomial in descending powers.

        Y = P(1)*X^N + P(2)*X^(N-1) + ... + P(N)*X + P(N+1)

    If X is a matrix or vector, the polynomial is evaluated at all
    points in X.  See POLYVALM for evaluation in a matrix sense.
    [Y,DELTA] = polyval(P,X,S) uses the optional output structure S created
    by POLYFIT to generate prediction error estimates DELTA.  DELTA is an
    estimate of the standard deviation of the error in predicting a future
    observation at X by P(X).
```

Curve Fitting Tool

File Fit View Tools Desktop Window Help

untitled fit 4

Fit name: untitled fit 4

X data: x

Y data: y

Z data: (none)

Weights: (none)

Polynomial

Degree: 1

Robust: Off

Center and scale

Fit Options...

Auto fit

Fit

Stop

Results

Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bound:
p1 = 5.702 (5.616, 5.788)
p2 = -0.1747 (-2.668, 2.318)

Goodness of fit:
SSE: 1.009e+05
R-square: 0.9712
Adjusted R-square: 0.9712
RMSE: 14.22

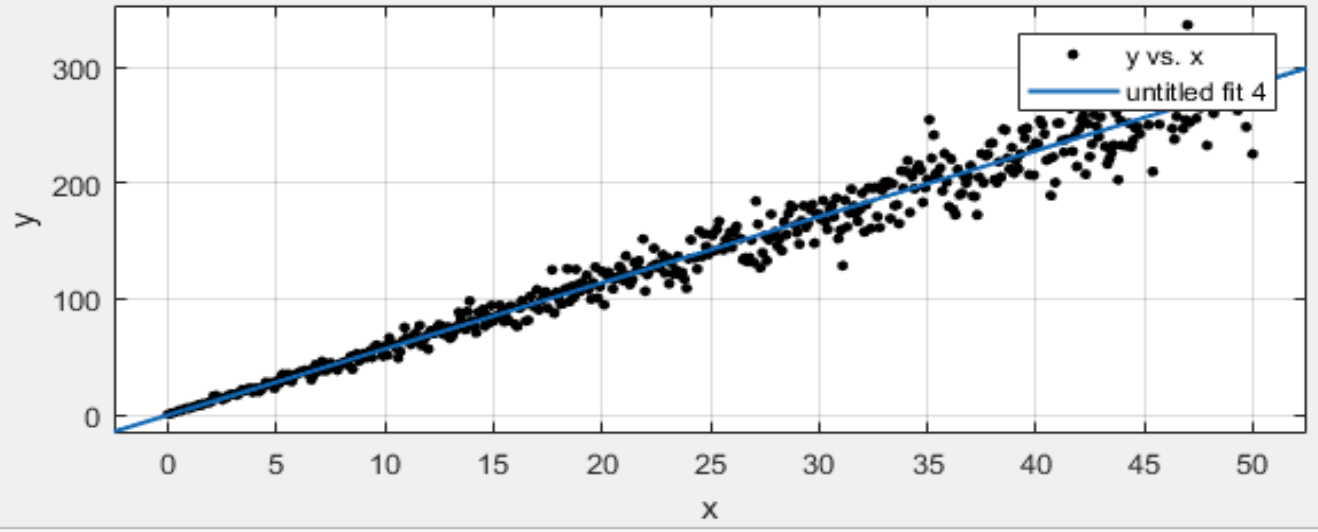


Table of Fits

Fit name	Data	Fit type	SSE	R-square	DFE	Adj R-sq	RMSE	# Coeff	Validation ...	Validation ...	Validation ...
fit 2	y vs. x	poly1	3.8909e-25	1	499	1	2.7924e-14	2			
untitled f...	y vs. x	poly1	7.9339e-25	1	499	1	3.9874e-14	2			
untitled f		linearintern									

Esercizio 1

La seguente tabella riporta probabilità percentuali di rilevare un certo numero di spikes neurali durante una osservazione di durata $T = 4$ secondi data una certa corrente assonale d'ingresso. Di conseguenza, ogni valore è stato calcolato utilizzando la distribuzione di Poisson con un dato momento del primo ordine per un neurone A, un neurone B, ed un neurone C. Verificare che le probabilità dei neuroni A, B, e C siano realizzazioni della stessa variabile aleatoria ipotizzando che i soli dati del neurone A siano risultati del campionamento statistico di una distribuzione F di Fisher con 2 g.d.l al numeratore e 12 g.d.l al denominatore (considerare possibile ogni approssimazione).

A	B	C
2.71	1.75	2.22
2.06	2.19	2.38
2.84	2.09	2.56
2.97	2.75	2.60
2.55		2.72
2.78		

Esercizio 2

I dati riportati nella seguente tabella riguardano due osservazioni nel tempo del peso di nove soggetti a cui è stato somministrato un farmaco anti-depressivo. Usando un'appropriata tecnica di inferenza statistica, si verifichi l'ipotesi per cui non vi stato alcun cambio di peso nel gruppo.

Soggetto	1	2	3	4	5	6	7	8	9
Peso t=1	91	88	111	81	91	88	72	80	69
Peso t=2	76	75	70	92	99	84	72	69	73

	T 1	T 2	T 3	T 4
P 1	4	3	0	2
P 2	4	2	2	2
P 3	3	2	2	1
P 4	5	1	2	2
P 5	3	4	1	2
P 6	5	4	3	3

Esercizio 3

A 6 psicologi vengono richieste le valutazioni (con un punteggio da 0 a 5 in ordine crescente di validità) di 4 terapie. Si stabilisca se vi sono delle differenze di valutazione fra le 4 diverse terapie.

Esercizio 4

Uno studente di ingegneria biomedica della Università di Pisa deve caratterizzare, durante il suo lavoro di tesi, la forza applicata da una mano robotica durante la presa di un oggetto, quando vi è applicato l'algoritmo di controllo A e l'algoritmo B. Sapendo che lo stesso studente ha effettuato 8 prove per ogni condizione ottenendo i seguenti valori:

A	8.26	8.13	8.35	8.07	8.34	9.01	8.44	8.99
B	7.95	7.89	7.90	8.14	7.92	7.11	9.11	7.34

Verificare l'ipotesi che le variazioni medie tra gli elementi di A e B siano dovute alla stocasticità del controllo applicato, con un livello di significatività $\alpha=0.01$ nel caso che:

1) La distribuzioni di probabilità di A e B siano il risultato di una somma di quadrati di funzioni Gaussiane;

2) La distribuzioni di probabilità di A e B siano il risultato di una somma di funzioni Gaussiane.

Esercizio 5

In un corso di Scienze della Nutrizione gli iscritti sono 15 e il Professore decide di fare uno studio statistico sul peso degli studenti per avere un campione di base su cui discutere delle abitudini alimentari fra i ragazzi.

Il campione è il seguente (Peso in Kg):

55.4	67.1	67.5	63.2	88.6
67.7	58.2	63.1	66.5	65.2
66.8	68.3	66.0	61.3	57.9

(1) Verificare se la mediana sia pari o inferiore a 62Kg;

(2) Verificare se la mediana sia pari o diversa da 62Kg;

(3) Riportare un intervallo di confidenza al 95% sulla mediana.

SOLUZIONI

```
%% esercizio 1
A = [ 2.71; 2.06; 2.84; 2.97; 2.55; 2.78];
B = [ 1.75; 2.19; 2.09; 2.75];
C = [ 2.22; 2.38; 2.56; 2.60; 2.72];
M = [A [B; NaN; NaN] [C; NaN]];
[p1, anvtb1, stat1] = kruskalwallis(M);
```

```
%% esercizio 2
Peso_t1 = [ 91 88 111 81 91 88 72 80 69];
Peso_t2 = [ 76 75 70 92 99 84 72 69 73];
[p2, h2, stat2] = signrank(Peso_t1, Peso_t2);
```

```
%% esercizio 3
F = [4 3 0 2; 4 2 2 2; 3 2 2 1;
    5 1 2 2; 3 4 1 2; 5 4 3 3];
[p3, tab3, stat3] = friedman(F);
[COMPARISON,MEANS,H,GNAMES] = multcompare(stat3);
```

```
%% esercizio 4
A4 = [8.26 8.13 8.35 8.07 8.34 9.01 8.44 8.99];
B4 = [7.95 7.89 7.90 8.14 7.92 7.11 9.11 7.34];
[p4_1, h4_1, stat4_1] = ranksum(A,B);
[h4_2, p4_2, stat4_2] = ttest2(A,B);
```

```
%% esercizio 5
Peso = [55.4 67.1 67.5 63.2 88.6 67.7 58.2 63.1 66.5 65.2 66.8 68.3 66.0 61.3 57.9];
[p5_1, h5_1] = signtest(Peso,62,'tail','left');
[p5_2, h5_2] = signtest(Peso,62,'tail','both');
C_alpha = 3; L = C_alpha+1; U = length(Peso)-C_alpha;
CI = [Peso(L) Peso(U)];
```

Esercizio

Si provi a definire una funzione MatLab, del tipo :

function Y = assign_rank(x)

Che dato in ingresso un vettore colonna, o una matrice, dia in uscita la corrispondente trasformazione in ranghi.
Si assuma che per le matrici la trasformazione è da farsi colonna per colonna

Esercizio

Si provi a definire una funzione MatLab, del tipo :

function Y = assign_rank(x)

Che dato in ingresso un vettore colonna, o una matrice, dia in uscita la corrispondente trasformazione in ranghi.
Si assuma che per le matrici la trasformazione è da farsi colonna per colonna

```
function R = assign_rank(A)

R = zeros(size(A));

for c = 1:size(A,2)
B = sort(unique(A(:,c)));
rango = 0;
for i = 1:length(B)
    d = (A(:,c)==B(i));
    R(d,c) = mean(rango+1:rango+sum(d));
    rango = rango+sum(d);
end
end

end
```