```
--------------------------------------------------------------------------------------
Laboratorio 01
--------------------------------------------------------------------------------------
> Ciliegi<-read.table("Cherry.dat",col.names=c("diam","alt","vol"))
> Ciliegi
   diam alt  vol
1   8.3  70 10.3
2   8.6  65 10.3
3   8.8  63 10.2
4  10.5  72 16.4
5  10.7  81 18.8
6  10.8  83 19.7
7  11.0  66 15.6
8  11.0  75 18.2
9  11.1  80 22.6
10 11.2  75 19.9
11 11.3  79 24.2
12 11.4  76 21.0
13 11.4  76 21.4
14 11.7  69 21.3
15 12.0  75 19.1
16 12.9  74 22.2
17 12.9  85 33.8
18 13.3  86 27.4
19 13.7  71 25.7
20 13.8  64 24.9
21 14.0  78 34.5
22 14.2  80 31.7
23 14.5  74 36.3
24 16.0  72 38.3
25 16.3  77 42.6
26 17.3  81 55.4
27 17.5  82 55.7
28 17.9  80 58.3
29 18.0  80 51.5
30 18.0  80 51.0
31 20.6  87 77.0
> attach(Ciliegi)
> vol
 [1] 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 24.2 21.0 21.4 21.3 19.1
[16] 22.2 33.8 27.4 25.7 24.9 34.5 31.7 36.3 38.3 42.6 55.4 55.7 58.3 51.5 51.0
[31] 77.0
> dim(Ciliegi)
[1] 31  3
> dim(Ciliegi)[1]
[1] 31
> Ciliegi[7,1]
[1] 11
> Ciliegi[7,]
  diam alt  vol
7   11  66 15.6
> Ciliegi[,2]
 [1] 70 65 63 72 81 83 66 75 80 75 79 76 76 69 75 74 85 86 71 64 78 80 74 72 77
[26] 81 82 80 80 80 87
> Ciliegi$alt
 [1] 70 65 63 72 81 83 66 75 80 75 79 76 76 69 75 74 85 86 71 64 78 80 74 72 77
[26] 81 82 80 80 80 87
> summary(alt)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
     63      72      76      76      80      87 
> var(alt)
[1] 40.6
> hist(alt)
> help(hist)
> boxplot(alt)
> # Analisi grafica
>
> plot(diam,vol)
> cor(diam,vol)
[1] 0.9671194
> # Y = alpha + beta X + epsilon, epsilon : E(epsilon)=0, Var(epsilon)=sigma^2
>
> # Stima del modello
>
> # --> Metodo dei minimi quadrati
> # ^beta=cov(X,Y)/Var(X)
> # ^alpha=mean(Y)-^beta*mean(X)
>
```

```
> beta<-(sum(diam*vol)/length(diam)-mean(diam)*mean(vol))/(mean(diam^2)-mean(diam)^2)
> n<-length(diam)
> n
[1] 31
> beta<-(sum(diam*vol)/n-mean(diam)*mean(vol))/(mean(diam^2)-mean(diam)^2)
> beta
[1] 5.065856
> beta<-cov(diam,vol)/((n-1)*var(diam)/n)
> beta
[1] 5.065856
> alpha<-mean(vol)-beta*mean(diam)
> alpha
[1] -36.94346
> abline(alpha,beta)
> abline(h=40)
> abline(v=14,lty="dashed")
>
>
> # Step 3: Analisi dei residui
> # res = y - y.stimati
>
> y.stimati<-alpha+beta*diam
> y.stimati
 [1]  5.103149  6.622906  7.636077 16.248033 17.261205 17.767790 18.780962 18.780962
 [9] 19.287547 19.794133 20.300718 20.807304 20.807304 22.327061 23.846818 28.406089
[17] 28.406089 30.432431 32.458774 32.965360 33.978531 34.991702 36.511459 44.110244
[25] 45.630001 50.695857 51.709028 53.735371 54.241956 54.241956 67.413183
> points(diam,y.stimati,pch="X")
> residui<-vol-y.stimati
> residui
 [1]  5.1968508  3.6770939  2.5639226  0.1519667  1.5387954  1.9322098 -3.1809615
 [8] -0.5809615  3.3124528  0.1058672  3.8992815  0.1926959  0.5926959 -1.0270610
[15] -4.7468179 -6.2060887  5.3939113 -3.0324313 -6.7587739 -8.0653595  0.5214692
[22] -3.2917021 -0.2114590 -5.8102436 -3.0300006  4.7041430  3.9909717  4.5646292
[29] -2.7419565 -3.2419565  9.5868168
> plot(residui)
> abline(h=0)
> summary(residui)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-8.065e+00 -3.107e+00  1.520e-01  3.209e-15  3.495e+00  9.587e+00
> boxplot(residui)
> hist(residui)
>
> # R2 = 1 - Var(residui)/Var(Y)
>
> r2<-1-var(residui)/var(vol)
> r2
[1] 0.9353199
>
> # ipotesi di normalita' y ~ N(alpha + beta*X,sigma^2)
> # epsilon ~ N(0,sigma^2)
>
> alpha
[1] -36.94346
> beta
[1] 5.065856
> sigma2<-sum(residui^2)/n
> sigma2
[1] 16.91299
> s2<-n*sigma2/(n-2)
> s2
[1] 18.07940
> # NB: s2 e' la varianza corretta
>
> # (^alpha-alpha)/sqrt(var(^alpha)) ~ t_(n-2)
>
> var.alpha<-s2*(1/n+mean(diam)^2/sum((diam-mean(diam))^2))
> var.alpha
[1] 11.3242
> var.beta<-s2/(sum((diam-mean(diam))^2))
> var.beta
[1] 0.06119536
> # ^alpha +- t_(n-2) * sqrt(Var(^alpha))
> a1<-alpha-qt(.975,n-2)*sqrt(var.alpha)
> a1
[1] -43.82595
> a2<-alpha+qt(.975,n-2)*sqrt(var.alpha)
> a2
```

```
[1] -30.06096
> # ^beta +- t_(n-2) * sqrt(Var(^beta))
> b1<-beta-qt(.975,n-2)*sqrt(var.beta)
> b1
[1] 4.559914
> b2<-beta+qt(.975,n-2)*sqrt(var.beta)
> b2
[1] 5.571799
>
> # test
> # H0 : beta = 0 , H1 : beta != 0
> # t = (^beta-0)/sqrt(var(beta)) ~H0 t_(n-2)
>
> test.t<-(beta-0)/sqrt(var.beta)
> test.t
[1] 20.47829
> qt(.975,n-2)
[1] 2.045230
> # a_oss = 2*min(P(T>t_oss),P(T<t_oss))
> 2*min(pt(test.t,n-2),1-pt(test.t,n-2))
[1] 0
> # rifiuto H0
>
> # test
> # H0 : alpha = 0 , H1 : alpha != 0
> # t = (^alpha-0)/sqrt(var(alpha)) ~H0 t_(n-2)
>
> test.t<-(alpha-0)/sqrt(var.alpha)
> test.t
[1] -10.97827
> 2*min(pt(test.t,n-2),1-pt(test.t,n-2))
[1] 7.621449e-12
> # rifiuto H0
>
> # Rifai l'esercizio con la trasformazione logaritmica dei dati
> # ln(vol)=alpha+beta*ln(diam)
> #        =ln(e^alpha * diam^beta)
> # vol=e^alpha * diam^beta
>
> x <- log(diam)
> y <- log(vol)
> plot(x,y)
> beta<-cov(x,y)/var(x)
> alpha<-mean(y)-beta*mean(x)
> alpha
[1] -2.353325
> beta
[1] 2.19997
>
> plot(diam,vol)
> abline(alpha,beta)
> val.predetti<-exp(alpha)*diam^beta
> lines(diam,val.predetti)
> res<-vol-val.predetti
> var(res)
[1] 10.56006
> var(residui)
[1] 0.01277479
>
> detach(Ciliegi)
------------------------------------------------------------------------------------------
Laboratorio 02
------------------------------------------------------------------------------------------
> brainbod<-read.table('brainbod.dat',header=T)
> brainbod
      species  bodywt brainwt
1     afeleph 6654.00 5712.00
2         cow  465.00  423.00
3      donkey  187.00  419.00
4         man   62.00 1320.00
5    graywolf   36.33  119.50
6      redfox    4.24   50.40
7  narmadillo    3.50   10.80
8     echidna    3.00   25.00
9   phalanger    1.62   11.40
10  guineapig    1.04    5.50
11 eurhedghog    0.79    3.50
12 chinchilla    0.43    4.00
```

```
13   ghamster    0.12    1.00
14     snmole    0.06    1.00
15      lbbat    0.01    0.25
>
> attach(brainbod)
>
> # Step1: Analisi preliminare dei dati
>
> summary(brainwt)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.25    3.75   11.40  540.40  269.30 5712.00
> summary(bodywt)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.01    0.61    3.00  494.60   49.17 6654.00
> par(mfrow=c(2,2))
> hist(bodywt)
> boxplot(bodywt)
> hist(brainwt)
> boxplot(brainwt)
>
>
> cor(bodywt,brainwt)
[1] 0.975509
> plot(bodywt,brainwt)
> identify(bodywt,brainwt,species)
 [1]  1  2  3  4  5  6  7  8  9 12 15
>
> x<-log(bodywt)
> y<-log(brainwt)
>
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.6050 -0.5398  1.0990  1.4400  3.8600  8.8030
> summary(y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -1.386   1.320   2.434   3.174   5.411   8.650
> hist(x)
> hist(y)
> boxplot(x)
> boxplot(y)
> cor(x,y)
[1] 0.9732277
> # y = alpha + beta x + epsilon?????????
>
> # Stima dei parametri
>
> n<-length(x)
> beta<-cov(x,y)/((n-1)*var(x)/n)
> beta
[1] 0.838046
> alpha<-mean(y)-beta*mean(x)
> alpha
[1] 1.96775
> plot(x,y)
> abline(alpha,beta)
>
> # Analisi dei residui
> val.pred<-alpha+beta*x
> res<-y-val.pred
> summary(res)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-1.068e+00 -4.156e-01 -1.909e-01  7.253e-16  3.602e-01  1.759e+00
> boxplot(res)
> max(res)
[1] 1.758908
> brainbod[4,]
  species bodywt brainwt
4     man     62    1320
> plot(x,y)
> r2<-1-var(res)/var(y)
> r2
[1] 0.9423396
>
> qqnorm(res)
> qqline(res)
>
> # intervallo di confidenza
> sigma2<-sum(res^2)/n
```

```
> sigma2
[1] 0.4492044
> s2<-sum(res^2)/(n-2)
> s2
[1] 0.5183127
> var.alpha<-s2*(1/n+mean(x)^2/sum((x-mean(x))^2))
> var.beta<-s2/sum((x-mean(x))^2)
> var.alpha
[1] 0.07924495
> var.beta
[1] 0.004435418
> alpha.lower<-alpha-qt(0.975,n-2)*sqrt(var.alpha)
> alpha.upper<-alpha+qt(0.975,n-2)*sqrt(var.alpha)
> alpha.lower
[1] 1.359596
> alpha.upper
[1] 2.575904
> beta.lower<-beta-qt(0.975,n-2)*sqrt(var.beta)
> beta.upper<-beta+qt(0.975,n-2)*sqrt(var.beta)
> beta.lower
[1] 0.6941678
> beta.upper
[1] 0.9819243
> # verifica di ipotesi: (beta-0)/sqrt(var.beta)
> beta/sqrt(var.beta)
[1] 15.65707
> qt(.975,n-2)
[1] 2.160369
> # 15 > 2 --> accetto l'ipotesi nulla
>
>
> ##################################
> # UOMO
> # elimino il dato scomodo
> x1<-x[-4]
> x1
 [1]  8.80297346  6.14203741  5.23110862  3.59264385  1.44456327  1.25276297  1.09861229
 [8]  0.48242615  0.03922071 -0.23572233 -0.84397007 -2.12026354 -2.81341072 -4.60517019
> y1<-y[-4]
> y1
 [1]  8.650325  6.047372  6.037871  4.783316  3.919991  2.379546  3.218876  2.433613
 [9]  1.704748  1.252763  1.386294  0.000000  0.000000 -1.386294
>
> cor(x,y)
[1] 0.9732277
>
> cor(x1,y1)
[1] 0.989434
>
> n1<-length(x1)
> beta1<-cov(x1,y1)/((n1-1)*var(x1)/14)
> beta1
[1] 0.80816
> alpha1<-mean(y1)-beta1*mean(x1)
> alpha1
[1] 1.879402
>
> plot(x,y)
> abline(alpha,beta)
> abline(alpha1,beta1,lty='dashed')
>
> res1<-y1-alpha1-beta1*x1
> par(mfrow=c(1,2))
> boxplot(res1)
> qqnorm(res1)
> qqline(res1)
>
-------------------------------------------------------------------------------------------
Laboratorio 03
-------------------------------------------------------------------------------------------
# Y = alpha + beta X + epsilon , epsilon ~ N(0,sigma^2)
> # costruiamo il nostro campione
>
> n<-30
> x<-1:30
> x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
[29] 29 30
```

```
> error<-rnorm(n,0,4)  # sigma^2=16
> error
 [1]  0.04406192 -2.78513488 -3.82533130 -7.47414116 -2.55828588 -8.81128405 10.57306139
 [8] -1.00157483 -3.13954695  6.84054944  1.73122246 -2.20881372 -2.77174140 -5.27932086
[15] -1.87458175  1.09956605 -6.85715244 -1.71112577 -0.06596067  4.91313049 -4.76435622
[22] -1.04205041 -1.11789092 -3.23884442  1.96330506  0.43448044  4.65260502  0.84634824
[29]  0.33454760  2.94081540
> y<-5+3*x+error  # alpha=5, beta=3
> y
 [1]  8.044062  8.214865 10.174669  9.525859 17.441714 14.188716 36.573061 27.998425
 [8] 28.860453 41.840549 39.731222 38.791186 41.228259 41.720679 48.125418 54.099566
[15] 49.142848 57.288874 61.934039 69.913130 63.235644 69.957950 72.882109 73.761156
[22] 81.963305 83.434480 90.652605 89.846348 92.334548 97.940815
>
> # step 1
> cor(x,y)
[1] 0.989372
> plot(x,y)
>
>
> # step 2
> beta<-cov(x,y)/var(x)
> beta
[1] 3.120524
> alpha<-mean(y)-beta*mean(x)
> alpha
[1] 2.326758
>
> abline(alpha,beta)
>
> # step 3 analisi modello
>
> res<- y-alpha-beta*x
> boxplot(res)
> qqnorm(res)
> qqline(res)
>
>
> # aumento sigma^2
> error<-rnorm(n,0,10)  # sigma^2=100
> y<-5+3*x+error  # alpha=5, beta=3
> cor(x,y)
[1] 0.940608
> plot(x,y)
> beta<-cov(x,y)/var(x)
> beta
[1] 2.946698
> alpha<-mean(y)-beta*mean(x)
> alpha
[1] 7.626219
>
>
> # diminuisco sigma^2
> error<-rnorm(n,0,.25)  # sigma^2=0.0625
> y<-5+3*x+error  # alpha=5, beta=3
> cor(x,y)
[1] 0.9999632
> plot(x,y)
> beta<-cov(x,y)/var(x)
> beta
[1] 3.003701
> alpha<-mean(y)-beta*mean(x)
> alpha
[1] 4.899634
>
>
> #-------------------------------------------------------#
> # Studio di simulazione                                 #
> #-------------------------------------------------------#
>
> n<-30
> alpha<-5
> beta<-3
> sigma2<-16
>
> x<-1:30
>
> error<-rnorm(n,0,sqrt(sigma2))
```

```
> y<-5+3*x+error
>
> beta <-cov(x,y)/var(x)
> beta
[1] 3.017296
> error<-rnorm(n,0,sqrt(sigma2))
> y<-5+3*x+error
> beta <-cov(x,y)/var(x)
> beta
[1] 2.948699
>
> # numero tentativi = 1000
>
> beta.sim<-rep(0,1000)
> for (i in 1:1000) {
+     error<-rnorm(n,0,sqrt(sigma2))
+     y<-5+3*x+error
+     beta.sim[i]<-cov(x,y)/var(x)
+ }
>
> mean(beta.sim)
[1] 2.997770
>
> # numero tentativi = 10000
>
> beta.sim<-rep(0,10000)
> for (i in 1:10000) {
+      error<-rnorm(n,0,sqrt(sigma2))
+      y<-5+3*x+error
+      beta.sim[i]<-cov(x,y)/var(x)
+ }
> mean(beta.sim) # NB: vicinissima al vero valore del parametro
[1] 3.00149
>
> hist(beta.sim)
> boxplot(beta.sim)
>
> # sigma^2 / sum(x_i - var(x))^2
>
> var.beta<-sigma2/sum((x-mean(x))^2)
> var.beta
[1] 0.007119021
>
> # stima della densita'
> hist(beta.sim,prob=T)
> lines(density(beta.sim))
> lines(seq(2.7,3.3,.01),dnorm(seq(2.7,3.3,.01),3,sqrt(var.beta)))
> lines(seq(2.7,3.3,.01),dnorm(seq(2.7,3.3,.01),3,sqrt(var.beta)),col=2)
>
>
> #----------------------------------------------------#
> # Livello di copertura delgli IC per beta            #
> #----------------------------------------------------#
>
> n<-30
> x<-1:30
> error<-rnorm(n,0,4)
> y<-5+3*x+error
>
> # ^beta +- t_(n-2) * sqrt(var(beta))
>
> beta<-cov(x,y)/var(x)
> alpha<-mean(y)-beta*mean(x)
> res<-y-alpha-beta*x
> s2<-sum(res^2)/(n-2)
> beta.inf<-beta-qt(.975,n-2)*sqrt(var.beta)
> beta.inf
[1] 2.875947
> beta.sup<-beta+qt(.975,n-2)*sqrt(var.beta)
> beta.sup
[1] 3.21878
> # il 3 è compreso tra i due estremi
>
>
> #----------------------------------------------------#
> # studio simulazione su livello copertura per beta   #
> #----------------------------------------------------#
>
```

```
> n<-30
> x<-1:30
>
> nt<-1000
>
> #valore iniziale
> beta.ic<-matrix(NA,ncol=2,nrow=nt) # NA sta per matrice vuota
>
> # simulazione
> for (i in 1:nt) {
+      error<-rnorm(n,0,4)
+      y<-5+3*x+error
+      beta<-cov(x,y)/var(x)
+      alpha<-mean(y)-beta*mean(x)
+      res<-y-alpha-beta*x
+      s2<-sum(res^2)/(n-2)
+      var.beta<-s2/sum((x-mean(x))^2)
+      beta.ic[i,1]<-beta-qt(.975,n-2)*sqrt(var.beta)
+      beta.ic[i,2]<-beta+qt(.975,n-2)*sqrt(var.beta)
+ }
>
> # numero(beta.i<beta<beta.s)/nt
>
> sum( (beta.ic[,1]<=3)&(3<=beta.ic[,2]) )
[1] 948
> sum( (beta.ic[,1]<=3)&(3<=beta.ic[,2]) )/nt
[1] 0.948
>
>
> #-----------------------------------------------------#
> # studio simulazione su livello copertura per alpha     #
> #-----------------------------------------------------#
>
> n<-30
> x<-1:30
>
> nt<-1000
>
> #valore iniziale
> alpha.ic<-matrix(NA,ncol=2,nrow=nt) # NA sta per matrice vuota
>
> # simulazione
>
>
> for (i in 1:nt) {
+      error<-rnorm(n,0,4)
+      y<-5+3*x+error
+      beta<-cov(x,y)/var(x)
+      alpha<-mean(y)-beta*mean(x)
+      res<-y-alpha-beta*x
+      s2<-sum(res^2)/(n-2)
+      var.alpha<-s2*(1/n+(mean(x)^2)/sum((x-mean(x))^2))
+      alpha.ic[i,1]<-alpha-qt(.975,n-2)*sqrt(var.alpha)
+      alpha.ic[i,2]<-alpha+qt(.975,n-2)*sqrt(var.alpha)
+ }
>
> sum( (alpha.ic[,1]<=5)&(5<=alpha.ic[,2]) )
[1] 953
> sum( (alpha.ic[,1]<=5)&(5<=alpha.ic[,2]) )/nt
[1] 0.953
>
-----------------------------------------------------------------------------------------
Laboratorio 04
-----------------------------------------------------------------------------------------
> cement <- read.table("cement.dat",col.names=c("tempo", "resist"))
> attach(cement)
> cement
   tempo resist
1      1   13.0
2      1   13.3
3      1   11.8
4      2   21.9
5      2   24.5
6      2   24.7
7      3   29.8
8      3   28.0
9      3   24.1
10     3   24.2
```

```
11     3   26.2
12     7   32.4
13     7   30.4
14     7   34.5
15     7   33.1
16     7   35.7
17    28   41.8
18    28   42.6
19    28   40.3
20    28   35.7
21    28   37.3
> # Analisi preliminare
>
> plot(tempo,resist)
> cor(tempo,resist)
[1] 0.7862442
>
> # la relazione non sembra lineare
>
> # trasformazione
> # variabile esplicativa
>
> # log(x), exp(x), x^r, x^(-r)
>
> par(mfrow=c(2,2))
>
> plot(log(tempo),resist)
> cor(log(tempo),resist)
[1] 0.9396814
>
> plot(1/tempo,resist)
> cor(1/tempo,resist)
[1] -0.9499735
>
> plot(1/sqrt(tempo),resist)
> cor(1/sqrt(tempo),resist)
[1] -0.9732841
>
> plot(sqrt(tempo),resist)
> cor(sqrt(tempo),resist)
[1] 0.8613284
>
> x<-1/sqrt(tempo)
>
> # Stima del modello lineare
> # resist = beta_1 + beta_2*x + epsilon
> # var esplicativa: x, var risposta y
>
> fit <- lm(resist~x)  # il nome "fit" in inglese significa "stima"
>
> summary(fit)

Call:
lm(formula = resist ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-3.79469 -1.25666 -0.05666  1.89544  3.10531

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   45.655      1.023   44.63  < 2e-16 ***
x            -32.599      1.764  -18.48 1.34e-13 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 2.133 on 19 degrees of freedom
Multiple R-Squared: 0.9473,     Adjusted R-squared: 0.9445
F-statistic: 341.4 on 1 and 19 DF,  p-value: 1.337e-13
>
> # NB: ----------------------------------------
> # Stima del modello lineare senza intercetta
> # resist = beta_2*x + epsilon
> # var esplicativa: x, var risposta y
> # fit <- lm(resist~x-1)
> # ------------------------------------------------
>
> # resist = 45.655 - 32.599 * X
```

```
> # H_0: alpha=0 vs H_1
> # sqrt(S^2)=2.133
> # R^2 = 1 - SOMMA_QUADRATI(res)/SOMMA_QUADRATI(y)
> # R^2 adjusted = 1 - (S_Q(res)/(n-2)) / (S_Q(y)/(n-1))
>
> # H_0: beta=0
>
> # coef, residuals, fitted, summmary, deviance, predict, plot
>
> fit.val<-fit$fitted  # valori stimati
> fit.val
       1        2        3        4        5        6        7        8 
13.05666 13.05666 13.05666 22.60456 22.60456 22.60456 26.83444 26.83444 
       9       10       11       12       13       14       15       16 
26.83444 26.83444 26.83444 33.33414 33.33414 33.33414 33.33414 33.33414 
      17       18       19       20       21 
39.49469 39.49469 39.49469 39.49469 39.49469 
> par(mfrow=c(1,1))
> plot(resist,fit.val)  # plot tra valori originali e valori stimati
> abline(0,1)  # piu' sono vicini alla bisettrice e piu' il modello e' buono
>
>
> # valutazione della bonta' del modello
> # analisi dei residui
>
> res<-fit$residuals  # residui standardizzati (media=0 e var=1)
> res
          1           2           3           4           5           6 
-0.05665701  0.24334299 -1.25665701 -0.70456225  1.89543775  2.09543775 
          7           8           9          10          11          12 
 2.96555873  1.16555873 -2.73444127 -2.63444127 -0.63444127 -0.93413660 
         13          14          15          16          17          18 
-2.93413660  1.16586340 -0.23413660  2.36586340  2.30530942  3.10530942 
         19          20          21 
 0.80530942 -3.79469058 -2.19469058 
> #  altro comando
> res<-rstandard(fit) # residui standardizzati (media=0 e var=1)
> res
          1           2           3           4           5           6 
-0.02984778  0.12819679 -0.66202607 -0.34305540  0.92289950  1.02028065 
          7           8           9          10          11          12 
 1.42684868  0.56079683 -1.31564884 -1.26753485 -0.30525502 -0.45196818 
         13          14          15          16          17          18 
-1.41963860  0.56408577 -0.11328353  1.14468802  1.15297119  1.55308101 
         19          20          21 
 0.40276526 -1.89786623 -1.09764658 
>
> # summary, boxplot, hist
> boxplot(res)
> # c'e' un po' di asimmetria dei residui
>
> hist(res)
> # non molto campanulare
>
> # grafico(res)
> plot(res)
> abline(h=0)
>
> # grafico(res,stimati)
> plot(res,fit.val)
>
> # grafico(x,res)
> plot(x,res)
>
> # varianza sigma^2 non sembra costante
>
> qqnorm(res)
> qqline(res)
>
> # -----------------------------------------
> # R^2, qqnorm/hist(res), grafici(res)
>
> help(lm)
> lm(resist~I(x^2)) # la "I" serve per specificare la formula del modello

Call:
lm(formula = resist ~ I(x^2))

```

```
Coefficients:
(Intercept)       I(x^2)
      37.87       -26.92

>
> fit$coeff
(Intercept)           x
   45.65524   -32.59859
> plot(x,resist)
> abline(fit$coeff)
>
>
> #-----------------------------------------------
> #-----------------------------------------------
>
>
> windmill<-read.table("windmill.dat",header=T)
> windmill
    wind    dc
1   5.00 1.582
2   6.00 1.822
3   3.40 1.057
4   2.70 0.500
5  10.00 2.236
6   9.70 2.386
7   9.55 2.294
8   3.05 0.558
9   8.15 2.166
10  6.20 1.866
11  2.90 0.653
12  6.35 1.930
13  4.60 1.562
14  5.80 1.737
15  7.40 2.088
16  3.60 1.137
17  7.85 2.179
18  8.80 2.112
19  7.00 1.800
20  5.45 1.501
21  9.10 2.303
22 10.20 2.310
23  4.10 1.194
24  3.95 1.144
25  2.45 0.123
>
> attach(windmill)
>
> # analisi preliminare
>
> plot(wind,dc)
> cor(wind,dc)
[1] 0.9351434
> # mmmh... sembra esserci un legame lineare!
>
> # stima del modello
> fit<-lm(dc~wind)
>
> summary(fit)

Call:
lm(formula = dc ~ wind)

Residuals:
     Min       1Q   Median       3Q      Max
-0.59869 -0.14099  0.06059  0.17262  0.32184

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.13088    0.12599   1.039     0.31
wind         0.24115    0.01905  12.659 7.55e-12 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.2361 on 23 degrees of freedom
Multiple R-Squared: 0.8745,     Adjusted R-squared: 0.869
F-statistic: 160.3 on 1 and 23 DF,  p-value: 7.546e-12

>
```

```
> # H_0: alpha=0
> qt(.975,23)
[1] 2.068658
>
> # dc = 0.24115*wind
> abline(fit$coef)
>
>
> # --------------------------------
> # senza intercetta (come esercizio)
>
> # bonta del modello
> fit<-lm(dc~wind-1)
> summary(fit)

Call:
lm(formula = dc ~ wind - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.51276 -0.17156  0.08675  0.20282  0.36832

Coefficients:
     Estimate Std. Error t value Pr(>|t|)
wind  0.25949    0.00715   36.29   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.2364 on 24 degrees of freedom
Multiple R-Squared: 0.9821,     Adjusted R-squared: 0.9814
F-statistic:  1317 on 1 and 24 DF,  p-value: < 2.2e-16

>
> res<-fit$res
> qqnorm(res)
> qqline(res)
>
> plot(wind,res)
>
> # R^2 elevato, qqnorm soddisfacente, plot(x,res) ha un andamento poco
> # soddisfacente
> # --> bisogna riformulare il modello e ristimarlo
>
> # trasformazione della variabile x
>
> par(mfrow=c(2,2))
>
> plot(log(wind),dc)
> plot(exp(wind),dc)
> plot(wind^2,dc)
> plot(1/wind,dc)
>
> cor(log(wind),dc)
[1] 0.978454
> cor(1/wind,dc)
[1] -0.989962
>
> fit1<-lm(dc~I(1/wind))
> summary(fit1)

Call:
lm(formula = dc ~ I(1/wind))

Residuals:
     Min       1Q   Median       3Q      Max
-0.20547 -0.04941  0.01100  0.08352  0.12204

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.9789     0.0449   66.34   <2e-16 ***
I(1/wind)    -6.9345     0.2064  -33.59   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.09417 on 23 degrees of freedom
Multiple R-Squared:  0.98,      Adjusted R-squared: 0.9792
F-statistic:  1128 on 1 and 23 DF,  p-value: < 2.2e-16

```

```
>
> # dc = 2.47 - 6.93/wind # nuovo modello
> # bonta' del modello
> res1<-fit1$residuals
>
> par(mfrow=c(1,1))
> boxplot(res1)
>
> qqnorm(res1)
> qqline(res1)
>
> plot(1/wind,res1)
>
> # R^2, qqnorm
> plot(res1,fit1$fitted)
>
---------------------------------------------------------------------------------------------
Laboratorio 05
---------------------------------------------------------------------------------------------
> fruit<-read.table("Fruitfly.dat",col.names=c("RS","SS","NS"))
> fruit
     RS   SS   NS
1  12.8 38.4 35.4
2  21.6 32.9 27.4
3  14.8 48.5 19.3
4  23.1 20.9 41.8
5  34.6 11.6 20.3
6  19.7 22.3 37.6
7  22.6 30.2 36.9
8  29.6 33.4 37.3
9  16.4 26.7 28.2
10 20.3 39.0 23.4
11 29.3 12.8 33.7
12 14.9 14.6 29.2
13 27.3 12.2 41.7
14 22.4 23.1 22.6
15 27.5 29.4 40.4
16 20.3 16.0 34.4
17 38.7 20.1 30.4
18 26.4 23.3 14.9
19 23.7 22.9 51.8
20 26.1 22.5 33.8
21 29.5 15.1 37.9
22 38.6 31.0 29.5
23 44.4 16.9 42.4
24 23.2 16.1 36.6
25 23.6 10.8 47.4
> attach(fruit)
> names(fruit)
[1] "RS" "SS" "NS"
>
> # Analisi preliminare (statistiche descrittivi/grafiche)
> summary(RS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  12.80   20.30   23.60   25.26   29.30   44.40
> summary(SS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.80   16.00   22.50   23.63   30.20   48.50
> summary(NS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  14.90   28.20   34.40   33.37   37.90   51.80
>
> boxplot(RS,SS,NS)
>
> # differenza (RS,SS) e NS
> # differenza RS e SS
>
> ## Se Y1~F(theta1) e Y2~F(theta2), devo confrontare theta1 con theta1
> ## H0: mu1=mu2
>
> RSS<-c(RS,SS)
> RSS
 [1] 12.8 21.6 14.8 23.1 34.6 19.7 22.6 29.6 16.4 20.3 29.3 14.9 27.3 22.4 27.5
[16] 20.3 38.7 26.4 23.7 26.1 29.5 38.6 44.4 23.2 23.6 38.4 32.9 48.5 20.9 11.6
[31] 22.3 30.2 33.4 26.7 39.0 12.8 14.6 12.2 23.1 29.4 16.0 20.1 23.3 22.9 22.5
[46] 15.1 31.0 16.9 16.1 10.8
> # t-student mu(RSS) = mu(NS)
> # normalita', sigma^2 omoschedasticita'
```

```
>
> boxplot(RSS,NS)
> hist(RSS,nclass=8)
> hist(NS,nclass=8)
>
> qqnorm(RSS)
> qqline(RSS)
>
> qqnorm(NS)
> qqline(NS)
>
> var(RSS)
[1] 77.00249
> var(NS)
[1] 79.9596
>
> # TRV (test rapporto di verosim. W(theta))
> #     H0: sigma^2(RSS) = sigma^2(NS)
> #   '--> H0: sigma^2(RSS)/sigma^2(NS)=1
> #   '--> H0: tau=1
> #   '--> H0: tau=S^2_1/S^2_2~F(n-1,m-1)=1
> help(var.test)
> var.test(RSS,NS)

        F test to compare two variances

data:  RSS and NS
F = 0.963, num df = 49, denom df = 24, p-value = 0.8833
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4564086 1.8658070
sample estimates:
ratio of variances
         0.9630174

>
> # test t H0: mu(RSS) = mu(NS)
> # test t H0: mu(RSS) - mu(NS) = 0
>
> # t = (mean1 - mean2) - (mu1 - mu2) / sqrt(stima varianza)
>
> help(t.test)
> t.test(RSS,NS,var.equal=T)

        Two Sample t-test

data:  RSS and NS
t = -4.1286, df = 73, p-value = 9.587e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -13.240812  -4.619188
sample estimates:
mean of x mean of y
   24.442    33.372

> # dal p-value --> rifiuto H0 --> le medie sono differenti
>
> boxplot(RS,SS,NS)
>
> # RS e NS sono uguali?
>
> qqnorm(RS);qqline(RS)
>
> var(RS)
[1] 60.41007
> var(NS)
[1] 79.9596
>
> var.test(RS,NS)

        F test to compare two variances

data:  RS and NS
F = 0.7555, num df = 24, denom df = 24, p-value = 0.4974
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.3329286 1.7144557
sample estimates:
```

```
ratio of variances
         0.7555074

>
> t.test(RS,NS,var.equal=T)

        Two Sample t-test

data:  RS and NS
t = -3.4251, df = 48, p-value = 0.001268
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -12.880308  -3.351692
sample estimates:
mean of x mean of y
   25.256    33.372

>
> # esercizio: SS-NS
>
> t.test(RS,NS)

        Welch Two Sample t-test

data:  RS and NS
t = -3.4251, df = 47.087, p-value = 0.001283
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -12.882696  -3.349304
sample estimates:
mean of x mean of y
   25.256    33.372

>
> boxplot(RS,SS,NS)
>
> # normalita'
> qqnorm(SS);qqline(SS)
>
> # omoschedasticita'
> var(RS)
[1] 60.41007
> var(SS)
[1] 95.42293
>
> var.test(RS,SS)

        F test to compare two variances

data:  RS and SS
F = 0.6331, num df = 24, denom df = 24, p-value = 0.2698
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.2789774 1.4366273
sample estimates:
ratio of variances
          0.633077

>
> t.test(RS,SS,var.equal=T)

        Two Sample t-test

data:  RS and SS
t = 0.6521, df = 48, p-value = 0.5175
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.391875  6.647875
sample estimates:
mean of x mean of y
   25.256    23.628

>
> boxplot(RS,SS,NS)
> boxplot(log(RS),log(SS),log(NS))
> qqnorm(log(RS));qqline(log(RS))
>
> detach()
```

```
>
> # ------------------------------------------------------------------
> capto<-read.table("Capto.dat",header=T)
> attach(capto)
> capto
    Sp  Sd  Dp  Dd
1  210 201 130 125
2  169 165 122 121
3  187 166 124 121
4  160 157 104 106
5  167 147 112 101
6  176 145 101  85
7  185 168 121  98
8  206 180 124 105
9  173 147 115 103
10 146 136 102  98
11 174 151  98  90
12 201 168 119  98
13 198 179 106 110
14 148 129 107 103
15 154 131 100  82
>
> # Y1 ~ N(m1,s1)
> # Y2 ~ N(m2,s2)
>
> # D = Y1 - Y2 ~ N(m1-m2,s1+s2)                  # prima
> # D = Y1 - Y2 ~ N(m1-m2,s1+s2-2*cov(Y1,Y2))    # adesso!
> # D = ~ N(m,s) , m=m1-m2 , s=s1+s2-2*cov(Y1,Y2)
> # H0: m=0
>
> # mean(D)-m / stima(varianza)
>
> SD<-Sd-Sp
> DD<-Dd-Dp
>
> boxplot(SD)
> qqnorm(SD);qqline(SD)
> qqnorm(DD);qqline(DD)
>
> t.test(SD)

        One Sample t-test

data:  SD
t = -8.1228, df = 14, p-value = 1.146e-06
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -23.93258 -13.93409
sample estimates:
mean of x
-18.93333

>
> t.test(Sd,Sp,paired=T)       # fa la stessa cosa che ho fatto prima

        Paired t-test

data:  Sd and Sp
t = -8.1228, df = 14, p-value = 1.146e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -23.93258 -13.93409
sample estimates:
mean of the differences
              -18.93333

>
> # H0: m >= 0
> # H1: m <  0
>
> # H0: m <= 0
> # H1: m >  0
>
> t.test(Sd,Sp,paired=T,alternative="g")

        Paired t-test

data:  Sd and Sp
```

```
t = -8.1228, df = 14, p-value = 1
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -23.03874       Inf
sample estimates:
mean of the differences
              -18.93333


>
> # esercizio: test per la D
> t.test(Dd,Dp,paired=T)

        Paired t-test

data:  Dd and Dp
t = -4.1662, df = 14, p-value = 0.000951
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -14.037215  -4.496118
sample estimates:
mean of the differences
              -9.266667


>
> t.test(Dd,Dp,paired=T,alternative="g")

        Paired t-test

data:  Dd and Dp
t = -4.1662, df = 14, p-value = 0.9995
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -13.18427       Inf
sample estimates:
mean of the differences
              -9.266667


-------------------------------------------------------------------------------------------
Laboratorio 06
-------------------------------------------------------------------------------------------
> # caricamento dati
> hook<-read.table('hook.dat',col.names=c('temp','press'))
> hook
    temp  press
1  210.8 29.211
2  210.2 28.559
3  208.4 27.972
4  202.5 24.697
5  200.6 23.726
6  200.1 23.369
7  199.5 23.030
8  197.0 21.892
9  196.4 21.928
10 196.3 21.654
11 195.6 21.605
12 193.4 20.480
13 193.6 20.212
14 191.4 19.758
15 191.1 19.490
16 190.6 19.386
17 189.5 18.869
18 188.8 18.356
19 188.5 18.507
20 185.7 17.267
21 186.0 17.221
22 185.6 17.062
23 184.1 16.959
24 184.6 16.881
25 184.1 16.817
26 183.2 16.385
27 182.4 16.235
28 181.9 16.106
29 181.9 15.928
30 181.0 15.919
31 180.6 15.376
> attach(hook)
>
> # analisi preliminare: grafici
```

```
>
> plot(press,temp)
> cor(press,temp)
[1] 0.9958803
>
> # stima del modello
> fit<-lm(temp~press) # y in funzione di x --> y~x
> summary(fit)

Call:
lm(formula = temp ~ press)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6735 -0.6805  0.2203  0.5296  1.3976

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 146.67290    0.77641  188.91   <2e-16 ***
press         2.25260    0.03809   59.14   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.806 on 29 degrees of freedom
Multiple R-Squared: 0.9918,     Adjusted R-squared: 0.9915
F-statistic:  3498 on 1 and 29 DF,  p-value: < 2.2e-16

>
> # temp = 146.6 + 2.25*press    ## modello dei minimi quadrati
>
> # analisi dei residui
>
> # riassunti
> # analisi grafiche
> res<-fit$res
>
> # temp = 146.6 + 2.25*press    ## modello dei minimi quadrati
>
> # analisi dei residui
>
> # riassunti
> # analisi grafiche
> ren<-fit$res
> rm(ren)
> res<-fit$res
> boxplot(res)
> plot(res)
>
> plot(press,res)
>
> par(mfrow=c(2,2))
> plot(fit)
>
> # lieve asimmetria dei residui
> # x vs residui mostra un andamento di tipo parabolico
>
> # rimedio: riformulare il modello con una trasformata della x
>
> plot(press,res)
>
> # y= a + bx + cx^2
> fit1<-lm(temp~press+I(press^2),x=T)  # I: funz. indicatrice
> fit1$x
   (Intercept)  press I(press^2)
1            1 29.211   853.2825
2            1 28.559   815.6165
3            1 27.972   782.4328
4            1 24.697   609.9418
5            1 23.726   562.9231
6            1 23.369   546.1102
7            1 23.030   530.3809
8            1 21.892   479.2597
9            1 21.928   480.8372
10           1 21.654   468.8957
11           1 21.605   466.7760
12           1 20.480   419.4304
13           1 20.212   408.5249
14           1 19.758   390.3786
```

```
15           1 19.490   379.8601
16           1 19.386   375.8170
17           1 18.869   356.0392
18           1 18.356   336.9427
19           1 18.507   342.5090
20           1 17.267   298.1493
21           1 17.221   296.5628
22           1 17.062   291.1118
23           1 16.959   287.6077
24           1 16.881   284.9682
25           1 16.817   282.8115
26           1 16.385   268.4682
27           1 16.235   263.5752
28           1 16.106   259.4032
29           1 15.928   253.7012
30           1 15.919   253.4146
31           1 15.376   236.4214
attr(,"assign")
[1] 0 1 2
>
> summary(fit1)

Call:
lm(formula = temp ~ press + I(press^2), x = T)

Residuals:
     Min       1Q   Median       3Q      Max
-0.79906 -0.26314 -0.01578  0.25139  0.73891

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 126.701623   2.112363  59.981  < 2e-16 ***
press         4.157627   0.199069  20.885  < 2e-16 ***
I(press^2)   -0.043754   0.004552  -9.612 2.29e-10 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.3956 on 28 degrees of freedom
Multiple R-Squared: 0.9981,     Adjusted R-squared: 0.998
F-statistic:  7307 on 2 and 28 DF,  p-value: < 2.2e-16

> plot(temp,fit1$fitted)
> abline(0,1)
>
> # analisi dei residui
>
> res1<-fit1$res
> plot(res1)
>
> plot(press,res1)
> plot(I(press^2),res1)
>
> qqnorm(res1);qqline(res1)
>
> detach()
>
>
> # -------------------------------------------------------
> # cherry.dat
> # caricamento dati
> cherry<-read.table('cherry.dat',col.names=c('diam','alt','vol'))
> attach(cherry)
>
> # analisi preliminare
> par(mfrow=c(2,2))
> plot(vol~diam)
> plot(vol~alt)
> cor(vol,diam)
[1] 0.9671194
> cor(vol,alt)
[1] 0.5982497
> plot(diam,alt)
> cor(diam,alt)
[1] 0.5192801
> pairs(cbind(vol,diam,alt))  # riassume tutti i grafici di dispersione
>
> # volume ~ diametro * altezza
> # log(volume) ~ log(diametro) + log(altezza)
```

```
> pairs(cbind(log(vol),log(diam),log(alt)))
> cor(log(vol),log(diam))
[1] 0.976665
> cor(log(vol),log(alt))
[1] 0.6486377
>
> # ---------- stima del modello -----------
> fit<-lm(log(vol)~log(diam)+log(alt))
> summary(fit)

Call:
lm(formula = log(vol) ~ log(diam) + log(alt))

Residuals:
      Min        1Q    Median        3Q       Max
-0.168561 -0.048488  0.002431  0.063637  0.129223

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.63162    0.79979  -8.292 5.06e-09 ***
log(diam)    1.98265    0.07501  26.432  < 2e-16 ***
log(alt)     1.11712    0.20444   5.464 7.81e-06 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.08139 on 28 degrees of freedom
Multiple R-Squared: 0.9777,     Adjusted R-squared: 0.9761
F-statistic: 613.2 on 2 and 28 DF,  p-value: < 2.2e-16

>
> # log(volume) ~ -6.63 + 1.98*log(diametro) + 1.11*log(altezza)
> plot(log(vol),fit$fitt)
> abline(0,1)
>
> # analisi dei residui
> res<-fit$res
> boxplot(res)
>
> plot(res,log(diam))
> plot(log(diam),res)
> plot(log(alt),res)
>
> par(mfrow=c(2,2))
> plot(fit)
>
>
> ynew<-log(vol)[-18]
> ynew
 [1] 2.332144 2.332144 2.322388 2.797281 2.933857 2.980619 2.747271 2.901422
 [9] 3.117950 2.990720 3.186353 3.044522 3.063391 3.058707 2.949688 3.100092
[17] 3.520461 3.246491 3.214868 3.540959 3.456317 3.591818 3.645450 3.751854
[25] 4.014580 4.019980 4.065602 3.941582 3.931826 4.343805
> length(ynew)
[1] 30
> x1new<-log(diam)[-18]
> x2new<-log(alt)[-18]
>
> fit1<-lm(ynew ~ x1new + x2new)
> summary(fit1)

Call:
lm(formula = ynew ~ x1new + x2new)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17500 -0.05706  0.00624  0.05940  0.11383

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.18549    0.78061  -9.205 8.14e-10 ***
x1new        1.95816    0.07051  27.770  < 2e-16 ***
x2new        1.26100    0.19984   6.310 9.39e-07 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.07565 on 27 degrees of freedom
Multiple R-Squared: 0.9814,     Adjusted R-squared:  0.98
F-statistic: 712.3 on 2 and 27 DF,  p-value: < 2.2e-16
```

```
>
> plot(fit)
>
> # altro modo senza ridefinire le variabili
> fit1<-lm(log(vol)~log(diam)+log(alt),subset=-c(18))
> summary(fit1)

Call:
lm(formula = log(vol) ~ log(diam) + log(alt), subset = -c(18))

Residuals:
     Min       1Q   Median       3Q      Max
-0.17500 -0.05706  0.00624  0.05940  0.11383

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.18549    0.78061  -9.205 8.14e-10 ***
log(diam)    1.95816    0.07051  27.770  < 2e-16 ***
log(alt)     1.26100    0.19984   6.310 9.39e-07 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.07565 on 27 degrees of freedom
Multiple R-Squared: 0.9814,     Adjusted R-squared:  0.98
F-statistic: 712.3 on 2 and 27 DF,  p-value: < 2.2e-16

>
> fit1<-lm(log(vol)~log(diam)+log(alt),subset=-c(11,15,16,17,18))
> plot(fit1)
> summary(fit1)

Call:
lm(formula = log(vol) ~ log(diam) + log(alt), subset = -c(11,
    15, 16, 17, 18))

Residuals:
      Min        1Q    Median        3Q       Max
-0.083857 -0.047953 -0.003806  0.039704  0.103097

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.55062    0.63479 -10.319 4.21e-10 ***
log(diam)    1.99231    0.05569  35.773  < 2e-16 ***
log(alt)     1.09469    0.16394   6.678 8.21e-07 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.05729 on 23 degrees of freedom
Multiple R-Squared: 0.9907,     Adjusted R-squared: 0.9899
F-statistic:  1223 on 2 and 23 DF,  p-value: < 2.2e-16

>
> par(mfrow=c(1,1))
> qqnorm(fit1$res)
>
--------------------------------------------------------------------------------
Laboratorio 07
--------------------------------------------------------------------------------
> hills<-read.table('hills.dat',header=T)
> hills
                 dist climb    time
Greenmantle       2.5   650  16.083
Carnethy          6.0  2500  48.350
Craig Dunain      6.0   900  33.650
Ben Rha           7.5   800  45.600
Ben Lomond        8.0  3070  62.267
Goatfell          8.0  2866  73.217
Bens of Jura     16.0  7500 204.617
Cairnpapple       6.0   800  36.367
Scolty            5.0   800  29.750
Traprain          6.0   650  39.750
Lairig Ghru      28.0  2100 192.667
Dollar            5.0  2000  43.050
Lomonds           9.5  2200  65.000
Cairn Table       6.0   500  44.133
Eildon Two        4.5  1500  26.933
Cairngorm        10.0  3000  72.250
```

```
Seven Hills      14.0  2200  98.417
Knock Hill        3.0   350  78.650
Black Hill        4.5  1000  17.417
Creag Beag        5.5   600  32.567
Kildcon Hill      3.0   300  15.950
Meall Ant-Suidhe  3.5  1500  27.900
Half Ben Nevis    6.0  2200  47.633
Cow Hill          2.0   900  17.933
N Berwick Law     3.0   600  18.683
Creag Dubh        4.0  2000  26.217
Burnswark         6.0   800  34.433
Largo Law         5.0   950  28.567
Criffel           6.5  1750  50.500
Acmony            5.0   500  20.950
Ben Nevis        10.0  4400  85.583
Knockfarrel       6.0   600  32.383
Two Breweries    18.0  5200 170.250
Cockleroi         4.5   850  28.100
Moffat Chase     20.0  5000 159.833
> attach(hills)
>
> # ---------------------------------------------------
> # time = ??
>
> # ----- Analisi preliminare -----
> pairs(hills)
>
> # Y = X*beta + epsilon
> # ^beta = (Xt X)^(-1) Xt y
> # ^beta ~ N(0,sigma^2 * (Xt X)^(-1))
> # Se le colonne di (Xt X) sono esprimibili tramite le sue righe
> # il rango della matrice non e' piu' pieno --> cioa' si sballa tutto
> # se ci sono delle variabili in relazione tra di loro
>
> # R2=1  -->  ^beta_i ~ 0
>
> cor(hills)
           dist     climb      time
dist  1.0000000 0.6523461 0.9195892
climb 0.6523461 1.0000000 0.8052392
time  0.9195892 0.8052392 1.0000000
>
> # Stima del modello
>
> fit<-lm(time~dist)
> summary(fit)

Call:
lm(formula = time ~ dist)

Residuals:
    Min      1Q  Median      3Q     Max
-35.745  -9.037  -4.201   2.849  76.170

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.8407     5.7562  -0.841    0.406
dist          8.3305     0.6196  13.446 6.08e-15 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 19.96 on 33 degrees of freedom
Multiple R-Squared: 0.8456,     Adjusted R-squared: 0.841
F-statistic: 180.8 on 1 and 33 DF,  p-value: 6.084e-15

> # il modello non sara' piu':  time = alpha + beta * dist
> # ma: time = beta * dist
>
> fit1<-lm(time~dist -1) # tolgo l'intercetta
> summary(fit1)

Call:
lm(formula = time ~ dist - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-28.7646 -11.0285  -6.8327   0.5607  78.0847

```

```
Coefficients:
     Estimate Std. Error t value Pr(>|t|)
dist   7.9083     0.3615   21.88   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 19.87 on 34 degrees of freedom
Multiple R-Squared: 0.9337,     Adjusted R-squared: 0.9317
F-statistic: 478.6 on 1 and 34 DF,  p-value: < 2.2e-16

>
> # NB: R2 e' cambiato --> R^2 = sum(y.stimati^2)/sum(y^2)
> # modello: time = 7.90 * dist
>
> # ---- analisi bonta' del modello -----
> # R^2
> # analisi dei residui
> # grafico (residui, variabile esclusa)
>
> res<-fit1$res
> plot(res,climb)
>
> fit2<-lm(time~dist+climb-1)        # oppure:
> fit2<-update(fit1, . ~ . + climb)
>
> summary(fit2)

Call:
lm(formula = time ~ dist + climb - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-18.089 -10.053  -5.539  -3.180  58.235

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
dist  5.605651   0.551046  10.173 1.05e-11 ***
climb 0.010280   0.002118   4.853 2.84e-05 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 15.41 on 33 degrees of freedom
Multiple R-Squared: 0.9613,     Adjusted R-squared: 0.959
F-statistic: 409.8 on 2 and 33 DF,  p-value: < 2.2e-16

>
> # time = 5.60*dist + 0.01*climb
> # R2: 0.93 - 0.96
> # Analisi della variaza # voglio vedere quale modello e' piu' efficace
> # H0 : fit1
> # H1 : fit2
>
> # F = ((SQR1-SQR2)/(n-p1-n+p2)) / (SQR2/(n-p2))
>
> anova(fit1,fit2)
Analysis of Variance Table

Model 1: time ~ dist - 1
Model 2: time ~ dist + climb - 1
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1     34 13423.2
2     33  7832.5  1    5590.7 23.555 2.84e-05 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
> # rifiuto H0, conviene tener conto anche della seconda var esplicativa
>
> # time = 5.60*dist + 0.01*climb
>
> # analisi dei residui
> res2<-fit2$res
> boxplot(res2)
>
> qqnorm(res2);qqline(res2)
>
> plot(res2,fitted(fit2))
> identify(res2,fitted(fit2),dimnames(hills)[[1]])
[1]  7 11 33 35
>
```

```
> fit3<-lm(time~dist+climb-1,subset=-c(7,11,33,35))
> summary(fit3)

Call:
lm(formula = time ~ dist + climb - 1, subset = -c(7, 11, 33,
    35))

Residuals:
    Min      1Q  Median      3Q     Max
-14.161  -4.120  -2.137   1.415  59.386

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
dist  5.762763   0.779834   7.390 3.84e-08 ***
climb 0.005645   0.002886   1.956   0.0602 .
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 12.29 on 29 degrees of freedom
Multiple R-Squared: 0.9364,     Adjusted R-squared: 0.932
F-statistic: 213.6 on 2 and 29 DF,  p-value: < 2.2e-16

>
> fit3<-lm(time~dist+climb-1,subset=-c(7,11,33,35))
> summary(fit3)

Call:
lm(formula = time ~ dist + climb - 1, subset = -c(7, 11, 33,
    35))

Residuals:
    Min      1Q  Median      3Q     Max
-14.161  -4.120  -2.137   1.415  59.386

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
dist  5.762763   0.779834   7.390 3.84e-08 ***
climb 0.005645   0.002886   1.956   0.0602 .
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 12.29 on 29 degrees of freedom
Multiple R-Squared: 0.9364,     Adjusted R-squared: 0.932
F-statistic: 213.6 on 2 and 29 DF,  p-value: < 2.2e-16

> detach()
>
>
>
> # -------------------------------------------------------------
> # dati - gasoline
> gaso<-read.table("gasoline.dat")
> names(gaso)
[1] "V1" "V2" "V3" "V4" "V5"
> names(gaso)<-c('y','x1','x2','x3','x4')
> ls()
[1] "fit"   "fit1"  "fit2"  "fit3"  "gaso"  "hills" "res"   "res2"
> attach(gaso)
> y
 [1]  6.9 14.4  7.4  8.5  8.0  2.8  5.0 12.2 10.0 15.2 26.8 14.0 14.7  6.4 17.6
[16] 22.3 24.8 26.0 34.9 18.2 23.2 18.0 13.1 16.1 32.1 34.7 31.7 33.6 30.4 26.6
[31] 27.8 45.7
>
> # analisi preliminare
> pairs(gaso)
>
> # (x2,x3)
>
> cor(gaso)
            y         x1         x2         x3         x4
y   1.0000000  0.2463260  0.3840706 -0.3150243  0.7115262
x1  0.2463260  1.0000000  0.6205867 -0.7001539 -0.3216782
x2  0.3840706  0.6205867  1.0000000 -0.9062248 -0.2979843
x3 -0.3150243 -0.7001539 -0.9062248  1.0000000  0.4122466
x4  0.7115262 -0.3216782 -0.2979843  0.4122466  1.0000000
> # (x2,x3)
>
> # stima del modello
```

```
> fit1<-lm(y~x4)
> summary(fit1)

Call:
lm(formula = y ~ x4)

Residuals:
      Min        1Q    Median        3Q       Max 
-14.75837  -6.27829   0.05255   5.16243  17.84805 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -16.66206    6.68721  -2.492   0.0185 *  
x4            0.10937    0.01972   5.546 4.98e-06 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1 

Residual standard error: 7.659 on 30 degrees of freedom
Multiple R-Squared: 0.5063,     Adjusted R-squared: 0.4898 
> 
> 
> # y = -16.66 + 0.10*x4
> 
> fit11<-update(fit1, . ~ .+x1)
> fit12<-update(fit1, . ~ .+x2)
> fit13<-update(fit1, . ~ .+x3)
> 
> anova(fit1,fit11)
Analysis of Variance Table

Model 1: y ~ x4
Model 2: y ~ x4 + x1
  Res.Df     RSS Df Sum of Sq      F  Pr(>F)    
1     30 1759.69                                
2     29  861.95  1    897.75 30.204 6.4e-06 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1 
> anova(fit1,fit12)
Analysis of Variance Table

Model 1: y ~ x4
Model 2: y ~ x4 + x2
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)    
1     30 1759.69                                  
2     29  369.87  1   1389.83 108.97 2.468e-11 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1 
> anova(fit1,fit13)
Analysis of Variance Table

Model 1: y ~ x4
Model 2: y ~ x4 + x3
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)    
1     30 1759.69                                  
2     29  170.61  1   1589.08 270.11 3.111e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1 
> 
> # scelgo il trezo fit perche' ha il 'Sum of Sq' maggiore --> maggiore R2
> 
> summary(fit3)

Call:
lm(formula = time ~ dist + climb - 1, subset = -c(7, 11, 33, 
    35))

Residuals:
    Min      1Q  Median      3Q     Max 
-14.161  -4.120  -2.137   1.415  59.386 

Coefficients:
      Estimate Std. Error t value Pr(>|t|)    
dist  5.762763   0.779834   7.390 3.84e-08 ***
climb 0.005645   0.002886   1.956   0.0602 .  
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1 

Residual standard error: 12.29 on 29 degrees of freedom
```

```
Multiple R-Squared: 0.9364,     Adjusted R-squared: 0.932
F-statistic: 213.6 on 2 and 29 DF,  p-value: < 2.2e-16

>
> # --------
>
> # y = -18.46 + 0.155*x4 - 0.209*x3
>
> fit131<-update(fit13, . ~ .+x1)
> fit132<-update(fit13, . ~ .+x2)
>
> anova(fit13,fit131)
Analysis of Variance Table

Model 1: y ~ x4 + x3
Model 2: y ~ x4 + x3 + x1
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     29 170.61
2     28 146.00  1     24.61 4.7198 0.03844 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
> anova(fit13,fit132)
Analysis of Variance Table

Model 1: y ~ x4 + x3
Model 2: y ~ x4 + x3 + x2
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     29 170.612
2     28 160.620  1     9.992 1.7419 0.1976
>
> # scelgo la fit131
>
> summary(fit131)

Call:
lm(formula = y ~ x4 + x3 + x1)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5303 -1.3606 -0.2681  1.3911  4.7658

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.032034   7.223341   0.558   0.5811
x4           0.156527   0.006462  24.224  < 2e-16 ***
x3          -0.186571   0.015922 -11.718 2.61e-12 ***
x1           0.221727   0.102061   2.173   0.0384 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 2.283 on 28 degrees of freedom
Multiple R-Squared: 0.959,      Adjusted R-squared: 0.9546
F-statistic: 218.5 on 3 and 28 DF,  p-value: < 2.2e-16

>
> # --------
>
> fit131bis<-update(fit131, . ~ .-1)
> summary(fit131bis)

Call:
lm(formula = y ~ x4 + x3 + x1 - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6075 -1.3229 -0.3831  1.7549  4.9115

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
x4  0.157168   0.006283  25.017  < 2e-16 ***
x3 -0.179328   0.009116 -19.672  < 2e-16 ***
x1  0.274133   0.039548   6.932 1.28e-07 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 2.256 on 29 degrees of freedom
Multiple R-Squared: 0.9907,     Adjusted R-squared: 0.9898
F-statistic:  1034 on 3 and 29 DF,  p-value: < 2.2e-16
```

```
> # y = 0.15*x4 - 0.17*x3 + 0.27*x1
>
>
>
> # -------------------------------------------------------
> # Altro metodo Hyper veloce!!
> # y = alpha + b1*x1 + b2*x2 + b3*x3 + b4*x4
>
> fit<-lm(y~x1+x2+x3+x4)
> summary(fit)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5804 -1.5223 -0.1098  1.4237  4.6214

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.820774  10.123152  -0.674   0.5062
x1           0.227246   0.099937   2.274   0.0311 *
x2           0.553726   0.369752   1.498   0.1458
x3          -0.149536   0.029229  -5.116 2.23e-05 ***
x4           0.154650   0.006446  23.992  < 2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 2.234 on 27 degrees of freedom
Multiple R-Squared: 0.9622,     Adjusted R-squared: 0.9566
F-statistic: 171.7 on 4 and 27 DF,  p-value: < 2.2e-16

>
> fit1<-lm(y~x1+x2+x3+x4-1)
> summary(fit1)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4 - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6693 -1.2920 -0.1271  1.2348  4.5478

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
x1  0.182249   0.073618   2.476   0.0196 *
x2  0.375377   0.255639   1.468   0.1531
x3 -0.167438   0.012061 -13.882 4.45e-14 ***
x4  0.154725   0.006382  24.245  < 2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 2.213 on 28 degrees of freedom
Multiple R-Squared: 0.9914,     Adjusted R-squared: 0.9902
F-statistic: 806.6 on 4 and 28 DF,  p-value: < 2.2e-16

>
> # --> Elimino la x2 perche' l'alpha oss e' il piu' grande ed e' > 0.05
>
> fit2<-lm(y~x1+x3+x4-1)
> summary(fit2)

Call:
lm(formula = y ~ x1 + x3 + x4 - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6075 -1.3229 -0.3831  1.7549  4.9115

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
x1  0.274133   0.039548   6.932 1.28e-07 ***
x3 -0.179328   0.009116 -19.672  < 2e-16 ***
x4  0.157168   0.006283  25.017  < 2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

```
Residual standard error: 2.256 on 29 degrees of freedom
Multiple R-Squared: 0.9907,     Adjusted R-squared: 0.9898
F-statistic:  1034 on 3 and 29 DF,  p-value: < 2.2e-16

>
> # ora il modello va bene, viva la Ventura!
>
> # analisi dei residui
> res<-fit2$res
> boxplot(res)
> qqnorm(res);qqline(res)
>
> par(mfrow=c(2,2))
> plot(res,x1)
> plot(res,x2)
> plot(res,x3)
> plot(res,x4)
>
---------------------------------------------------------------------------------------
Laboratorio 08
---------------------------------------------------------------------------------------
> # sturdy.dat
> x1<-scan('sturdy.dat')
Read 26 items
> x1
 [1] 2.34 2.46 2.83 2.04 2.69 2.64 3.00 3.19 3.83 2.61 2.07 2.80 2.58 2.98 2.30
[16] 1.32 1.62 1.92 0.88 1.50 1.30 0.41 0.83 0.58 0.32 1.62
>
> group<-c(rep("A",5),rep("B",4),rep("C",6),rep("D",6),rep("E",5))
> group
 [1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "C" "C" "C" "C" "C" "C" "D" "D" "D" "D"
[20] "D" "D" "E" "E" "E" "E" "E"
>
> group<-factor(group)  # fattorizzazione dei gruppi
> group
 [1] A A A A A B B B B C C C C C C D D D D D D E E E E E
Levels:  A B C D E
>
> # ----- Esistono delle differenze tra le marche ???
>
> # t_i ~ N(mu_i,sigma^2_i)
>
> # normalita', varianze uguali
> # H0: mu_i = mu_j
>
> stu<-data.frame(x1,group)  # creazione del dataframe = read.table ('file.dat')
> stu
     x1 group
1  2.34     A
2  2.46     A
3  2.83     A
4  2.04     A
5  2.69     A
6  2.64     B
7  3.00     B
8  3.19     B
9  3.83     B
10 2.61     C
11 2.07     C
12 2.80     C
13 2.58     C
14 2.98     C
15 2.30     C
16 1.32     D
17 1.62     D
18 1.92     D
19 0.88     D
20 1.50     D
21 1.30     D
22 0.41     E
23 0.83     E
24 0.58     E
25 0.32     E
26 1.62     E
>
> attach(stu)
> # analisi preliminare
> plot(x1~group)
```

```
> # --> normalita' ok! (abbiamo pero' pochi dati)
> # boxplot
> var(x1[group=="A"])  # seleziono i valori di x1 = 'A'
[1] 0.09497
> var(x1[group=="B"])
[1] 0.2485667
> var(x1[group=="C"])
[1] 0.1089067
> var(x1[group=="D"])
[1] 0.1224667
> var(x1[group=="E"])
[1] 0.27317
>
> # ----- test TRV per verificare l'omoschedasticita'
> # '--> test di Bartlett
> help(bartlett.test)
>
> # '--> l'ipotesti di omoschedasticita' puo' essere assunta!
>
>
> stu.lm<-lm(x1~group,x=T)
> stu.lm$x
   (Intercept) groupB groupC groupD groupE
1            1      0      0      0      0
2            1      0      0      0      0
3            1      0      0      0      0
4            1      0      0      0      0
5            1      0      0      0      0
6            1      1      0      0      0
7            1      1      0      0      0
8            1      1      0      0      0
9            1      1      0      0      0
10           1      0      1      0      0
11           1      0      1      0      0
12           1      0      1      0      0
13           1      0      1      0      0
14           1      0      1      0      0
15           1      0      1      0      0
16           1      0      0      1      0
17           1      0      0      1      0
18           1      0      0      1      0
19           1      0      0      1      0
20           1      0      0      1      0
21           1      0      0      1      0
22           1      0      0      0      1
23           1      0      0      0      1
24           1      0      0      0      1
25           1      0      0      0      1
26           1      0      0      0      1
attr(,"assign")
[1] 0 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$group
[1] "contr.treatment"

>
> # x_i = alpha + b1*d1 + b2*d2 + b3*d3 + b4*d4
> # alpha      : E(A)
> # alpha + b1 : E(B)
> # alpha + b2 : E(C)
> # alpha + b3 : E(D)
> # alpha + b4 : E(E)
>
> summary(stu.lm)

Call:
lm(formula = x1 ~ group, x = T)

Residuals:
      Min        1Q    Median        3Q       Max
-0.543333 -0.235500  0.005667  0.212667  0.868000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.47200    0.17929  13.788 5.40e-12 ***
groupB       0.69300    0.26893   2.577 0.017583 *
groupC       0.08467    0.24276   0.349 0.730733
groupD      -1.04867    0.24276  -4.320 0.000302 ***
```

```
groupE      -1.72000    0.25355  -6.784 1.04e-06 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4009 on 21 degrees of freedom
Multiple R-Squared: 0.8433,     Adjusted R-squared: 0.8135
F-statistic: 28.26 on 4 and 21 DF,  p-value: 3.475e-08

>
> # E(A)=2.47
> # E(B)=2.47+0.69
> # E(C)=E(A)=2.47  # perche' il coeff non e' significativo
> # E(D)=2.47-1.04
> # E(E)=2.47-1.72
>
> # statistica che verifica l'ipotesi H0: b1=b2=b3=b4=0
> stu.aov<-aov(x1~group)
> summary(stu.aov)
            Df  Sum Sq Mean Sq F value    Pr(>F)
group        4 18.1681  4.5420  28.261 3.475e-08 ***
Residuals   21  3.3751  0.1607
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> detach()
> # -------------------------------------------------------
> grn<-factor(c(rep("G1",15),rep("G2",11)))
> grn
 [1] G1 G1 G1 G1 G1 G1 G1 G1 G1 G1 G1 G1 G1 G1 G1 G2 G2 G2 G2 G2 G2 G2 G2 G2 G2
[26] G2
Levels:  G1 G2
> stu1<-data.frame(x1,grn)
> attach(stu1)
> plot(x1~grn)
>
> # test? --> t di student
> # ipotesi: normalita', omoschedasticita'
> qqnorm(x1[grn=="G1"]);qqline(x1[grn=="G1"])
> qqnorm(x1[grn=="G2"]);qqline(x1[grn=="G2"])
>
> var(x1[grn=="G1"])
[1] 0.2083495
> var(x1[grn=="G2"])
[1] 0.2934164
>
> var.test(x1[grn=="G1"],x1[grn=="G2"])

        F test to compare two variances

data:  x1[grn == "G1"] and x1[grn == "G2"]
F = 0.7101, num df = 14, denom df = 10, p-value = 0.5422
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1999999 2.2345278
sample estimates:
ratio of variances
         0.7100815

> t.test(x1[grn=="G1"],x1[grn=="G2"],var.equal=T,alternative="g")

        Two Sample t-test

data:  x1[grn == "G1"] and x1[grn == "G2"]
t = 8.0229, df = 24, p-value = 1.500e-08
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 1.237152      Inf
sample estimates:
mean of x mean of y
 2.690667  1.118182

>
> detach()
>
> # ------------------------------------------------------------------
>
> topi<-read.table('rats.dat',header=T)
> topi
```

```
   tempo veleno trattamento
1   0.31      I           A
2   0.82      I           B
3   0.43      I           C
4   0.45      I           D
5   0.45      I           A
6   1.10      I           B
7   0.45      I           C
8   0.71      I           D
9   0.46      I           A
10  0.88      I           B
11  0.63      I           C
12  0.66      I           D
13  0.43      I           A
14  0.72      I           B
15  0.76      I           C
16  0.62      I           D
17  0.36     II           A
18  0.92     II           B
19  0.44     II           C
20  0.56     II           D
21  0.29     II           A
22  0.61     II           B
23  0.35     II           C
24  1.02     II           D
25  0.40     II           A
26  0.49     II           B
27  0.31     II           C
28  0.71     II           D
29  0.23     II           A
30  1.24     II           B
31  0.40     II           C
32  0.38     II           D
33  0.22    III           A
34  0.30    III           B
35  0.23    III           C
36  0.30    III           D
37  0.21    III           A
38  0.37    III           B
39  0.25    III           C
40  0.36    III           D
41  0.18    III           A
42  0.38    III           B
43  0.24    III           C
44  0.31    III           D
45  0.23    III           A
46  0.29    III           B
47  0.22    III           C
48  0.33    III           D
>
> topi$veleno<-factor(topi$veleno)
> topi$tratt<-factor(topi$tratt)
> attach(topi)
>
> # effetto del trattamento sul veleno II  # W la Ventura
> topi2<-data.frame(tempo=tempo[veleno=='II'],trattamento=trattamento[veleno=='II'])
> attach(topi2)
>
> plot(tempo,trattamento)
> plot(tempo~trattamento)
>
> bartlett.test(tempo,trattamento)

        Bartlett test for homogeneity of variances

data:  tempo and trattamento
Bartlett's K-squared = 9.5432, df = 3, p-value = 0.02288

>
> # ipotesi non soddisfatte: casa fare????
> # '--> trasformo i dati con la potenza della Ventura
> # log(tempo), exp(tempo), tempo^p, tempo^-p
>
> plot(1/tempo~trattamento)
> bartlett.test(1/tempo~trattamento)

        Bartlett test for homogeneity of variances

```

```
data:  1/tempo by trattamento
Bartlett's K-squared = 1.2807, df = 3, p-value = 0.7337

> # ----- ipotesi di normalita' soddisfatta
> # '--> anche i qqnorm
>
> topi.aov<-aov(1/tempo~trattamento)
> summary(topi.aov)
            Df Sum Sq Mean Sq F value   Pr(>F)
trattamento  3 9.1424  3.0475  7.3913 0.004594 **
Residuals   12 4.9477  0.4123
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> # H0: mu1=mu2=mu3=mu4 , alpha_oss<0.05 --> rifiuto H0
>
> detach()
>
>
> mor<-read.table('Morley.dat',header=T)
> mor$Expt<-factor(mor$Expt)
> attach(mor)
>
> plot(Speed~Expt)
> # c'e' un comportamento anomalo nel gruppo 3, medie simili
>
> # verifica della normalita'
> par(mfrow=c(2,3))
> qqnorm(Speed[Expt==1]);qqline(Speed[Expt==1])
> qqnorm(Speed[Expt==2]);qqline(Speed[Expt==2])
> qqnorm(Speed[Expt==3]);qqline(Speed[Expt==3])
> qqnorm(Speed[Expt==4]);qqline(Speed[Expt==4])
> qqnorm(Speed[Expt==5]);qqline(Speed[Expt==5])
> # risultati poco soddisfacenti
>
> bartlett.test(Speed,Expt)

        Bartlett test for homogeneity of variances

data:  Speed and Expt
Bartlett's K-squared = 11.5518, df = 4, p-value = 0.02102

> bartlett.test(1/Speed,Expt)

        Bartlett test for homogeneity of variances

data:  1/Speed and Expt
Bartlett's K-squared = 13.0758, df = 4, p-value = 0.01091

> bartlett.test(log(Speed),Expt)

        Bartlett test for homogeneity of variances

data:  log(Speed) and Expt
Bartlett's K-squared = 11.7615, df = 4, p-value = 0.01922

> bartlett.test(Speed^2,Expt)

        Bartlett test for homogeneity of variances

data:  Speed^2 and Expt
Bartlett's K-squared = 12.3319, df = 4, p-value = 0.01505

>
> # non trovo una trasformata adatta!!
> # si usa allora (grazie alla Ventura che ce lo ha detto!) un test non
> # parametrico per il confronto delle medie
> # '--> non assumo nessuna distrib per i dati
> # ===> test di Kruskal Wallis
>
> kruskal.test(Speed,Expt)

        Kruskal-Wallis rank sum test

data:  Speed and Expt
Kruskal-Wallis chi-squared = 15.0221, df = 4, p-value = 0.004656

>
```

```

------------------------------------------------------------------------------
Riassunto:
* Controllo le ipotesti: di normalita' e omoschedasticita'
* Se sono verificate faccio i test altrimenti provo delle trafsormate
* se funziona e' tutto ok e faccio i test, altrimenti uso il test Kruskal

----------------------------------------------------------------------------------------------
Laboratorio 09
----------------------------------------------------------------------------------------------
> # ---- penicillin.dat
> pen<-read.table("penicillin.dat",header=T)
> pen
   miscela modo penicillina
1        I    A          89
2        I    B          88
3        I    C          97
4        I    D          94
5       II    A          84
6       II    B          77
7       II    C          92
8       II    D          79
9      III    A          81
10     III    B          87
11     III    C          87
12     III    D          85
13      IV    A          87
14      IV    B          92
15      IV    C          89
16      IV    D          84
17       V    A          79
18       V    B          81
19       V    C          80
20       V    D          88
> # disegno fatto con 5x4 livelli complessivi, con 1 replicaz
> attach(pen)
> miscela
 [1] I   I   I   I   II  II  II  II  III III III III IV  IV  IV  IV  V   V   V  
[20] V  
Levels:  I II III IV V
> modo
 [1] A B C D A B C D A B C D A B C D A B C D
Levels:  A B C D
> is.factor(miscela)
[1] TRUE
> is.factor(modo)
[1] TRUE
> # penicillina = miscela + errore
> # penicillina = modo + errore
> # penicillina = miscela + modo + errore
>
> # anova 1 fattore
> # analisi preliminare
> plot(penicillina~modo)
>
> # penicillina = modo + errore
> # errore ~ N(0, sigma^2)
>
> # H0: mu_1 = mu_2 = mu_3 = mu_4
>
> bartlett.test(penicillina,modo)

        Bartlett test for homogeneity of variances

data:  penicillina and modo
Bartlett's K-squared = 0.6901, df = 3, p-value = 0.8755

> # H0: mu_1 = mu_2 = mu_3 = mu_4
>
> # TRV funzione monotona crescente F
> # F = [SS(tra i campioni)/(J-1)]/[SS(entro i campioni)/(n-J)] # SS e' la somma
>                                                               # dei quadrati
> # F ~ F_(J-1),(n-J)       # F di Schnedecor
>
> # aov
> summary(aov(penicillina~modo))
            Df Sum Sq Mean Sq F value Pr(>F)
modo         3  70.00   23.33  0.7619 0.5318
```

```
Residuals   16 490.00   30.62
>
> # penicillina = modo + errore
> # accettiamo H0
>
> # pen_ij = cost + alpha_j + err_ij
>
> fit<-lm(penicillina~modo,x=T)
> fit$x
   (Intercept) modoB modoC modoD
1            1     0     0     0
2            1     1     0     0
3            1     0     1     0
4            1     0     0     1
5            1     0     0     0
6            1     1     0     0
7            1     0     1     0
8            1     0     0     1
9            1     0     0     0
10           1     1     0     0
11           1     0     1     0
12           1     0     0     1
13           1     0     0     0
14           1     1     0     0
15           1     0     1     0
16           1     0     0     1
17           1     0     0     0
18           1     1     0     0
19           1     0     1     0
20           1     0     0     1
attr(,"assign")
[1] 0 1 1 1
attr(,"contrasts")
attr(,"contrasts")$modo
[1] "contr.treatment"

>
> summary(fit)

Call:
lm(formula = penicillina ~ modo, x = T)

Residuals:
       Min         1Q     Median         3Q        Max
-9.000e+00 -3.250e+00  3.002e-15  3.000e+00  8.000e+00

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   84.000      2.475  33.941 2.45e-16 ***
modoB          1.000      3.500   0.286    0.779
modoC          5.000      3.500   1.429    0.172
modoD          2.000      3.500   0.571    0.576
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 5.534 on 16 degrees of freedom
Multiple R-Squared: 0.125,      Adjusted R-squared: -0.03906
F-statistic: 0.7619 on 3 and 16 DF,  p-value: 0.5318

> # H0: alpha_1 = alpha_2 = alpha_3 = 0
>
> anova(fit) # confronto tra i due modelli di regressione
Analysis of Variance Table

Response: penicillina
          Df Sum Sq Mean Sq F value Pr(>F)
modo       3  70.00   23.33  0.7619 0.5318
Residuals 16 490.00   30.62
> # '--> stesso risultato di prima!
>
> plot(penicillina~miscela)
> bartlett.test(penicillina,miscela)

        Bartlett test for homogeneity of variances

data:  penicillina and miscela
Bartlett's K-squared = 2.3859, df = 4, p-value = 0.6652
```

```
> # p-value elevato --> accetto l'ipotesi di omoscedasticita'
> summary(aov(penicillina~miscela))
            Df  Sum Sq Mean Sq F value  Pr(>F)
miscela      4 264.000  66.000  3.3446 0.03801 *
Residuals   15 296.000  19.733
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> # ---------------------------------------------
> # pen_(ijk) = cost + alpha_j + beta_k + errore_(ijk)
>
> summary(aov(penicillina~modo+miscela))
            Df  Sum Sq Mean Sq F value  Pr(>F)
modo         3  70.000  23.333  1.2389 0.33866
miscela      4 264.000  66.000  3.5044 0.04075 *
Residuals   12 226.000  18.833
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
> # e' come se avessimo fatto una scomposizione del tipo:
> # Var(Y) = Var(modo) + Var(miscela) + Var(residua)
>
> fit1<-lm(penicillina~modo+miscela,x=T)
> fit$x
   (Intercept) modoB modoC modoD
1            1     0     0     0
2            1     1     0     0
3            1     0     1     0
4            1     0     0     1
5            1     0     0     0
6            1     1     0     0
7            1     0     1     0
8            1     0     0     1
9            1     0     0     0
10           1     1     0     0
11           1     0     1     0
12           1     0     0     1
13           1     0     0     0
14           1     1     0     0
15           1     0     1     0
16           1     0     0     1
17           1     0     0     0
18           1     1     0     0
19           1     0     1     0
20           1     0     0     1
attr(,"assign")
[1] 0 1 1 1
attr(,"contrasts")
attr(,"contrasts")$modo
[1] "contr.treatment"

>
> summary(fit1)

Call:
lm(formula = penicillina ~ modo + miscela, x = T)

Residuals:
   Min     1Q Median     3Q    Max
 -5.00  -2.25  -0.50   2.25   6.00

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   90.000      2.745  32.791  4.1e-13 ***
modoB          1.000      2.745   0.364  0.72194
modoC          5.000      2.745   1.822  0.09351 .
modoD          2.000      2.745   0.729  0.48018
miscelaII     -9.000      3.069  -2.933  0.01254 *
miscelaIII    -7.000      3.069  -2.281  0.04159 *
miscelaIV     -4.000      3.069  -1.304  0.21686
miscelaV     -10.000      3.069  -3.259  0.00684 **
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 4.34 on 12 degrees of freedom
Multiple R-Squared: 0.5964,     Adjusted R-squared: 0.361
F-statistic: 2.534 on 7 and 12 DF,  p-value: 0.07535

> # il modo non influenza i risultati
```

```
> # miscela I   : Y=90
> # miscela II  : Y=90-9=81
> # miscela III : Y=90-7=83
> # miscela IV  : Y=90=90  # il -4 non e' significativo!
> # miscela V   : Y=90-10=80
>
> # H0: alpha=beta=0
>
> anova(fit1)
Analysis of Variance Table

Response: penicillina
          Df  Sum Sq Mean Sq F value  Pr(>F)
modo       3  70.000  23.333  1.2389 0.33866
miscela    4 264.000  66.000  3.5044 0.04075 *
Residuals 12 226.000  18.833
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> # ------------
> # la F sopra, deriva dal seguente test:
> # F = [(~sigma - ^sigma)/(p-p0)] / [^sigma/(n-p)]
>
> # Var(modo)
> # Y = cost + alpha + beta + errore
> # H0: alpha = 0
> # SQ(modo)
>
> # H0: beta = 0
>
> detach()
> # ------------------------------------------------------------------------
>
> topi<-read.table("rats.dat",header=T)
> topi
   tempo veleno trattamento
1   0.31      I           A
2   0.82      I           B
3   0.43      I           C
4   0.45      I           D
5   0.45      I           A
6   1.10      I           B
7   0.45      I           C
8   0.71      I           D
9   0.46      I           A
10  0.88      I           B
11  0.63      I           C
12  0.66      I           D
13  0.43      I           A
14  0.72      I           B
15  0.76      I           C
16  0.62      I           D
17  0.36     II           A
18  0.92     II           B
19  0.44     II           C
20  0.56     II           D
21  0.29     II           A
22  0.61     II           B
23  0.35     II           C
24  1.02     II           D
25  0.40     II           A
26  0.49     II           B
27  0.31     II           C
28  0.71     II           D
29  0.23     II           A
30  1.24     II           B
31  0.40     II           C
32  0.38     II           D
33  0.22    III           A
34  0.30    III           B
35  0.23    III           C
36  0.30    III           D
37  0.21    III           A
38  0.37    III           B
39  0.25    III           C
40  0.36    III           D
41  0.18    III           A
42  0.38    III           B
```

```
43  0.24    III          C
44  0.31    III          D
45  0.23    III          A
46  0.29    III          B
47  0.22    III          C
48  0.33    III          D
> # disegno fattoriale 3x4, con 4 replicazioni
>
> attach(topi)
> is.factor(veleno)
[1] TRUE
> is.factor(trattamento)
[1] TRUE
>
> # ---------------------
> # analisi preliminare
> plot(tempo~veleno+trattamento)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
> help(interaction.plot)
> interaction.plot(veleno,trattamento,tempo)
> interaction.plot(trattamento,veleno,tempo)
>
> # --------------
> # Risp = Effetto(costante) + Effetto(trattamento) + Effetto(veleno) +
> #        Effetto(trattamento/veleno) + Effetto(residuo)
> summary(aov(tempo~trattamento + veleno + trattamento:veleno))
                   Df  Sum Sq Mean Sq F value    Pr(>F)
trattamento         3 0.92121 0.30707 13.8056 3.777e-06 ***
veleno              2 1.03301 0.51651 23.2217 3.331e-07 ***
trattamento:veleno  6 0.25014 0.04169  1.8743    0.1123
Residuals          36 0.80073 0.02224
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
> # lo stesso e' con:
> summary(aov(tempo~trattamento * veleno))
                   Df  Sum Sq Mean Sq F value    Pr(>F)
trattamento         3 0.92121 0.30707 13.8056 3.777e-06 ***
veleno              2 1.03301 0.51651 23.2217 3.331e-07 ***
trattamento:veleno  6 0.25014 0.04169  1.8743    0.1123
Residuals          36 0.80073 0.02224
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> # H0: trattamento non ha effetto
> # y_ijk ~ N(mu_ijk,sigma^2)
> # H0: mu1_jk = mu2_jk = mu3_jk
> # i: trattamento # j: veleno # k: interazione tra trattamento e veleno
>
> # y_ijk = mu + alpha_i + beta_j + delta_ij + epsilon
>
> g<-lm(tempo~trattamento * veleno)
> summary(g)

Call:
lm(formula = tempo ~ trattamento * veleno)

Residuals:
     Min       1Q   Median       3Q      Max
-0.32500 -0.04875  0.00500  0.04312  0.42500

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              0.41250    0.07457   5.532 2.94e-06 ***
trattamentoB             0.46750    0.10546   4.433 8.37e-05 ***
trattamentoC             0.15500    0.10546   1.470   0.1503
trattamentoD             0.19750    0.10546   1.873   0.0692 .
velenoII                -0.09250    0.10546  -0.877   0.3862
velenoIII               -0.20250    0.10546  -1.920   0.0628 .
trattamentoB:velenoII    0.02750    0.14914   0.184   0.8547
trattamentoC:velenoII   -0.10000    0.14914  -0.671   0.5068
trattamentoD:velenoII    0.15000    0.14914   1.006   0.3212
trattamentoB:velenoIII -0.34250    0.14914  -2.297   0.0276 *
trattamentoC:velenoIII -0.13000    0.14914  -0.872   0.3892
trattamentoD:velenoIII -0.08250    0.14914  -0.553   0.5836
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

```

Residual standard error: 0.1491 on 36 degrees of freedom
Multiple R-Squared: 0.7335,     Adjusted R-squared: 0.6521
F-statistic:  9.01 on 11 and 36 DF,  p-value: 1.986e-07

>
> # il trattamento A e' quello di riferimento
> # C e D si comportano come A --> non sono significativi
> # il veleno significativo e' il II
> # l'interazione significativa e' solo B:III
> anova(g)
Analysis of Variance Table

Response: tempo
                   Df  Sum Sq Mean Sq F value    Pr(>F)
trattamento         3 0.92121 0.30707 13.8056 3.777e-06 ***
veleno              2 1.03301 0.51651 23.2217 3.331e-07 ***
trattamento:veleno  6 0.25014 0.04169  1.8743    0.1123
Residuals          36 0.80073 0.02224
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> # H0: alpha_i = 0
> # H0: beta_i = 0
> # H0: delta_ij = 0
>
> # '--> y_ijk = mu + alpha_i + beta_j + epsilon
> g0<-lm(tempo~trattamento+veleno)
>
> # A, I : Y = 0.45
> # A, II: Y = 0.45  # xche' il veleno II non e' significativo
> # A, III: Y = 0.45 - 0.34 = 0.11
> # B, I: Y = 0.45 + 0.36 = 81
> # ecc...
>
>
> # ----------------------
> # analisi dei residui
>
> qqnorm(g0$res);qqline(g0$res)
> plot(g0$fitted,g0$res)
> # cosa fare wuando i residui non sono soddisfacenti???
> # '--> riformulo il modello!
>
> g1<-lm(1/tempo~veleno * trattamento)
> anova(g1)
Analysis of Variance Table

Response: 1/tempo
                   Df Sum Sq Mean Sq F value    Pr(>F)
veleno              2 34.877  17.439 72.6347 2.310e-13 ***
trattamento         3 20.414   6.805 28.3431 1.376e-09 ***
veleno:trattamento  6  1.571   0.262  1.0904    0.3867
Residuals          36  8.643   0.240
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> qqnorm(g1$res);qqline(g1$res)
> plot(g1$fitted,g1$res)
>
```

Laboratorio 10

```
> cats<-read.table('cats.dat',col.names=c('B','H','S'))
> cats
     B    H S
1  2.3  9.6 1
2  3.0 10.6 1
3  2.9  9.9 1
4  2.4  8.7 1
5  2.3 10.1 1
6  2.0  7.0 1
7  2.2 11.0 1
8  2.1  8.2 1
9  2.3  9.0 1
10 2.1  7.3 1
11 2.1  8.5 1
12 2.2  9.7 1
```

```
13 2.0  7.4 1
14 2.3  7.3 1
15 2.2  7.1 1
16 2.3  9.0 1
17 2.1  7.6 1
18 2.0  9.5 1
19 2.9 10.1 1
20 2.7 10.2 1
21 2.6 10.1 1
22 2.3  9.5 1
23 2.6  8.7 1
24 2.1  7.2 1
25 2.9  9.4 2
26 2.4  9.3 2
27 2.2  7.2 2
28 2.9 11.3 2
29 2.5  8.8 2
30 3.1  9.9 2
31 3.0 13.3 2
32 2.5 12.7 2
33 3.4 14.4 2
34 3.0 10.0 2
35 2.6 10.5 2
36 2.5  8.6 2
37 2.8 10.0 2
38 3.1 12.1 2
39 3.0 13.8 2
40 2.7 12.0 2
41 2.8 12.0 2
42 2.1 10.1 2
43 3.3 11.5 2
44 3.4 12.2 2
45 2.8 13.5 2
46 2.7 10.4 2
47 3.2 11.6 2
48 3.0 10.6 2
> as.factor(cats$S)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2
Levels:  1 2
> is.factor(cats$S)
[1] FALSE
> cats$S<-factor(cats$S)
>
> attach(cats)
> # ---------------------
> # peso del cuore a seconda del sesso
>
> # analisi preliminare
> plot(H~S)
>
> # confronto tra i due gruppi
> # ipotesi: normalita'
> qqnorm(H[S==1]);qqline(H[S==1])
> qqnorm(H[S==2]);qqline(H[S==2])
>
> # ipotesi: omoschedasticita'/varianze
> var(H[S==1])
[1] 1.552446
> var(H[S==2])
[1] 3.234783
> plot(H~S)
>
> # H0: sigma^2_1/sigma^2_2 = 1
> # F = (S^2_1/(n1_1)) / (s^2_2/(n2-1))
> var.test(H[S==1],H[S==2])

        F test to compare two variances

data:  H[S == 1] and H[S == 2]
F = 0.4799, num df = 23, denom df = 23, p-value = 0.08496
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.2076113 1.1094088
sample estimates:
ratio of variances
         0.4799227
```

```
>
> # t-Student
> # H0: mu_1 = mu_2
> # H1: mu_1 < mu_2
>
> t.test(H[S==1],H[S==2],alternative="less",var.equal=T)

        Two Sample t-test

data:  H[S == 1] and H[S == 2]
t = -4.8419, df = 46, p-value = 7.455e-06
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -1.412780
sample estimates:
mean of x mean of y
   8.8875   11.0500

> # ----------------
> # mi aspetto che ci sia una relazione tra il peso ed il cuore
> # proviamo:
> plot(H~B)
> cor(H,B)
[1] 0.7335559
>
> # evidenziamo sul grafico i maschi e le femmine con colori diversi
> points(B[S==1],H[S==1],col='red')
> points(B[S==2],H[S==2],col='blue')
>
> # ----- H = alpha + alpha_j(S) + beta * B + beta_j * (S:B) + err
> # S:B=interazione tra S e B
>
> # S = (1,0)  # 1=F, 0=M
> # (S:B) = (1,0)
> #
> # H_F = (alpha + alpha_1) + (beta+beta_1)*B
> # H_M = alpha + beta*B
>
> # H0: alpha_1=beta_1=0
> # F = ((~sigma^2 - ^sigma^2)/(p-p0))/(^sigma^2/(n-p))
>
> fit<-lm(H~S + B + S:B,x=T)
> fit$x
   (Intercept) S2   B S2:B
1            1  0 2.3  0.0
2            1  0 3.0  0.0
3            1  0 2.9  0.0
4            1  0 2.4  0.0
5            1  0 2.3  0.0
6            1  0 2.0  0.0
7            1  0 2.2  0.0
8            1  0 2.1  0.0
9            1  0 2.3  0.0
10           1  0 2.1  0.0
11           1  0 2.1  0.0
12           1  0 2.2  0.0
13           1  0 2.0  0.0
14           1  0 2.3  0.0
15           1  0 2.2  0.0
16           1  0 2.3  0.0
17           1  0 2.1  0.0
18           1  0 2.0  0.0
19           1  0 2.9  0.0
20           1  0 2.7  0.0
21           1  0 2.6  0.0
22           1  0 2.3  0.0
23           1  0 2.6  0.0
24           1  0 2.1  0.0
25           1  1 2.9  2.9
26           1  1 2.4  2.4
27           1  1 2.2  2.2
28           1  1 2.9  2.9
29           1  1 2.5  2.5
30           1  1 3.1  3.1
31           1  1 3.0  3.0
32           1  1 2.5  2.5
33           1  1 3.4  3.4
34           1  1 3.0  3.0
```

```
35           1  1 2.6  2.6
36           1  1 2.5  2.5
37           1  1 2.8  2.8
38           1  1 3.1  3.1
39           1  1 3.0  3.0
40           1  1 2.7  2.7
41           1  1 2.8  2.8
42           1  1 2.1  2.1
43           1  1 3.3  3.3
44           1  1 3.4  3.4
45           1  1 2.8  2.8
46           1  1 2.7  2.7
47           1  1 3.2  3.2
48           1  1 3.0  3.0
attr(,"assign")
[1] 0 1 2 3
attr(,"contrasts")
attr(,"contrasts")$S
[1] "contr.treatment"

>
> # -----  H = alpha + alpha_j(S) + beta * B + beta_j * (S:B) + err
> # H_F = alpha + beta * B
> # H_M = alpha + alpha_1*1 + beta * B + beta_1*B
> summary(fit)

Call:
lm(formula = H ~ S + B + S:B, x = T)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9813 -0.9589 -0.1629  0.8573  2.6277

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.9318     2.1105   1.389  0.17178
S2           -0.2849     3.0313  -0.094  0.92554
B             2.5525     0.8975   2.844  0.00674 **
S2:B          0.4177     1.1784   0.354  0.72466
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1.28 on 44 degrees of freedom
Multiple R-Squared: 0.5664,     Adjusted R-squared: 0.5368
F-statistic: 19.16 on 3 and 44 DF,  p-value: 4.269e-08

> fit<-lm(H~S * B,x=T) # stessa cosa
> summary(fit)

Call:
lm(formula = H ~ S * B, x = T)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9813 -0.9589 -0.1629  0.8573  2.6277

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.9318     2.1105   1.389  0.17178
S2           -0.2849     3.0313  -0.094  0.92554
B             2.5525     0.8975   2.844  0.00674 **
S2:B          0.4177     1.1784   0.354  0.72466
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1.28 on 44 degrees of freedom
Multiple R-Squared: 0.5664,     Adjusted R-squared: 0.5368
F-statistic: 19.16 on 3 and 44 DF,  p-value: 4.269e-08

> # '---> H = 2.55*B
> fit1<-lm(H~B - 1)
> summary(fit1)

Call:
lm(formula = H ~ B - 1)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-2.0372 -0.9521 -0.0969  0.8620  3.0732

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
B   3.8507     0.0714   53.93   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1.292 on 47 degrees of freedom
Multiple R-Squared: 0.9841,     Adjusted R-squared: 0.9838
F-statistic:  2909 on 1 and 47 DF,  p-value: < 2.2e-16

> # '---> H = 3.85*B
>
> anova(fit1,fit)
Analysis of Variance Table

Model 1: H ~ B - 1
Model 2: H ~ S + B + S:B
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     47 78.488
2     44 72.073  3     6.415 1.3055 0.2846
>
> qqnorm(fit1$res);qqline(fit1$res)
> plot(fit1$fitted,fit1$res)
>
> detach()
>
> # ---------------------------------------------------------------------
> insu<-read.table('insulate.dat',col.names=c('quando','temp','cons'))
> insu
   quando temp cons
1   prima -0.8  7.2
2   prima -0.7  6.9
3   prima  0.4  6.4
4   prima  2.5  6.0
5   prima  2.9  5.8
6   prima  3.2  5.8
7   prima  3.6  5.6
8   prima  3.9  4.7
9   prima  4.2  5.8
10  prima  4.3  5.2
11  prima  5.4  4.9
12  prima  6.0  4.9
13  prima  6.0  4.3
14  prima  6.0  4.4
15  prima  6.2  4.5
16  prima  6.3  4.6
17  prima  6.9  3.7
18  prima  7.0  3.9
19  prima  7.4  4.2
20  prima  7.5  4.0
21  prima  7.5  3.9
22  prima  7.6  3.5
23  prima  8.0  4.0
24  prima  8.5  3.6
25  prima  9.1  3.1
26  prima 10.2  2.6
27   dopo -0.7  4.8
28   dopo  0.8  4.6
29   dopo  1.0  4.7
30   dopo  1.4  4.0
31   dopo  1.5  4.2
32   dopo  1.6  4.2
33   dopo  2.3  4.1
34   dopo  2.5  4.0
35   dopo  2.5  3.5
36   dopo  3.1  3.2
37   dopo  3.9  3.9
38   dopo  4.0  3.5
39   dopo  4.0  3.7
40   dopo  4.2  3.5
41   dopo  4.3  3.5
42   dopo  4.6  3.7
43   dopo  4.7  3.5
44   dopo  4.9  3.4
45   dopo  4.9  3.7
46   dopo  4.9  4.0
```

```
47   dopo  5.0  3.6
48   dopo  5.3  3.7
49   dopo  6.2  2.8
50   dopo  7.1  3.0
51   dopo  7.2  2.8
52   dopo  7.5  2.6
53   dopo  8.0  2.7
54   dopo  8.7  2.8
55   dopo  8.8  1.3
56   dopo  9.7  1.5
>
> is.factor(insu$quando)
[1] TRUE
> attach(insu)
>
> # analisi preliminare
> plot(cons~temp)
> cor(cons,temp)
[1] -0.6832545
>
> # evidenziamo i punti a seconda del quando:
> points(temp[quando=='prima'],cons[quando=='prima'],col='red')
> points(temp[quando=='dopo'],cons[quando=='dopo'],col='blue')
>
> cor(temp[quando=='prima'],cons[quando=='prima'])
[1] -0.9714978
> cor(temp[quando=='dopo'],cons[quando=='dopo'])
[1] -0.9017078
>
> # ------------------------------------------------------------
> # cons = alpha + alpha_1*quando + beta*temp + beta_1*(quando:temp) + err
> fit<-lm(cons~quando * temp,x=T) # si considerano gli effetti marginali del
>                                 # quando e della temperatura
> fit$x
   (Intercept) quandoprima temp quandoprima:temp
1            1           1 -0.8             -0.8
2            1           1 -0.7             -0.7
3            1           1  0.4              0.4
4            1           1  2.5              2.5
5            1           1  2.9              2.9
6            1           1  3.2              3.2
7            1           1  3.6              3.6
8            1           1  3.9              3.9
9            1           1  4.2              4.2
10           1           1  4.3              4.3
11           1           1  5.4              5.4
12           1           1  6.0              6.0
13           1           1  6.0              6.0
14           1           1  6.0              6.0
15           1           1  6.2              6.2
16           1           1  6.3              6.3
17           1           1  6.9              6.9
18           1           1  7.0              7.0
19           1           1  7.4              7.4
20           1           1  7.5              7.5
21           1           1  7.5              7.5
22           1           1  7.6              7.6
23           1           1  8.0              8.0
24           1           1  8.5              8.5
25           1           1  9.1              9.1
26           1           1 10.2             10.2
27           1           0 -0.7              0.0
28           1           0  0.8              0.0
29           1           0  1.0              0.0
30           1           0  1.4              0.0
31           1           0  1.5              0.0
32           1           0  1.6              0.0
33           1           0  2.3              0.0
34           1           0  2.5              0.0
35           1           0  2.5              0.0
36           1           0  3.1              0.0
37           1           0  3.9              0.0
38           1           0  4.0              0.0
39           1           0  4.0              0.0
40           1           0  4.2              0.0
41           1           0  4.3              0.0
42           1           0  4.6              0.0
43           1           0  4.7              0.0
```

```
44           1           0  4.9              0.0
45           1           0  4.9              0.0
46           1           0  4.9              0.0
47           1           0  5.0              0.0
48           1           0  5.3              0.0
49           1           0  6.2              0.0
50           1           0  7.1              0.0
51           1           0  7.2              0.0
52           1           0  7.5              0.0
53           1           0  8.0              0.0
54           1           0  8.7              0.0
55           1           0  8.8              0.0
56           1           0  9.7              0.0
attr(,"assign")
[1] 0 1 2 3
attr(,"contrasts")
attr(,"contrasts")$quando
[1] "contr.treatment"

>
> # '---> cons_prima = alpha+alpha_1 + (beta+beta_1)*temp
> # '---> cons_dopo = alpha + beta*temp
>
> # H0: alpha_1=beta_1=0
> summary(fit)

Call:
lm(formula = cons ~ quando * temp, x = T)

Residuals:
     Min       1Q   Median       3Q      Max
-0.97802 -0.18011  0.03757  0.20930  0.63803

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       4.72385    0.11810  40.000  < 2e-16 ***
quandoprima       2.12998    0.18009  11.827 2.32e-16 ***
temp             -0.27793    0.02292 -12.124  < 2e-16 ***
quandoprima:temp -0.11530    0.03211  -3.591  0.00073 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.323 on 52 degrees of freedom
Multiple R-Squared: 0.9277,     Adjusted R-squared: 0.9235
F-statistic: 222.3 on 3 and 52 DF,  p-value: < 2.2e-16

>
> # cons_prima = alpha+alpha_1 + (beta+beta_1)*temp = 6.84-0.38*temp
> # cons_dopo = 4.72 - 0.27 * temp
>
> fit1<-lm(cons~temp)
> anova(fit1,fit)
Analysis of Variance Table

Model 1: cons ~ temp
Model 2: cons ~ quando + temp + quando:temp
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     54 39.995
2     52  5.425  2    34.570 165.67 < 2.2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
>
> abline(6.84,-0.38)
> abline(4.72,-0.27)
>
> # analisi dei residui
> qqnorm(fit$res);qqline(fit$res)
> plot(fit$fitted,fit$res)
>
> # -----------------------------------------
> fit2<-lm(cons~temp,subset=(quando=='prima'))
> summary(fit2)

Call:
lm(formula = cons ~ temp, subset = (quando == "prima"))

Residuals:
     Min       1Q   Median       3Q      Max
```

```
-0.62020 -0.19947  0.06068  0.16770  0.59778

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.85383    0.11842   57.88   <2e-16 ***
temp        -0.39324    0.01959  -20.08   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.2813 on 24 degrees of freedom
Multiple R-Squared: 0.9438,     Adjusted R-squared: 0.9415
F-statistic: 403.1 on 1 and 24 DF,  p-value: < 2.2e-16

>
> fit2<-lm(cons~temp,subset=(quando=='dopo'))
> summary(fit2)

Call:
lm(formula = cons ~ temp, subset = (quando == "dopo"))

Residuals:
     Min       1Q   Median       3Q      Max
-0.97802 -0.11082  0.02672  0.25294  0.63803

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.72385    0.12974   36.41  < 2e-16 ***
temp        -0.27793    0.02518  -11.04 1.05e-11 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.3548 on 28 degrees of freedom
Multiple R-Squared: 0.8131,     Adjusted R-squared: 0.8064
F-statistic: 121.8 on 1 and 28 DF,  p-value: 1.046e-11

>
> # H0: beta_1 = beta_2
>
```