

LABORATORIO 3 - Statistica Descrittiva Multivariata

**STATISTICA E LABORATORIO (CDL in INTERNET OF THINGS,
BIG DATA, MACHINE LEARNING)**

Anno Accademico 2022-2023

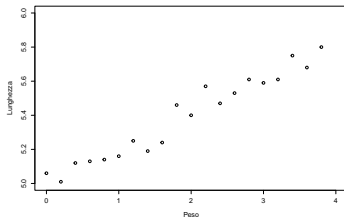
Section 1

Studio della dipendenza

Molla

Si considerano i dati sulla misura in cm di una molla sottoposta a $n = 20$ pesi diversi in Kg.

```
yy <- c(5.06,5.01,5.12,5.13,5.14,5.16,5.25,5.19,5.24,5.46,5.4,
        5.57,5.47,5.53,5.61,5.59,5.61,5.75,5.68,5.8)
xx <- c(0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2, 2.2,2.4,2.6,2.8,3,
        3.2,3.4,3.6,3.8)
plot(xx, yy, xlab = "Peso", ylab = "Lunghezza", lwd = 2,
      cex.axis = 1.2, cex.lab = 1.2, ylim = c(5,6), xlim = c(0,4))
```



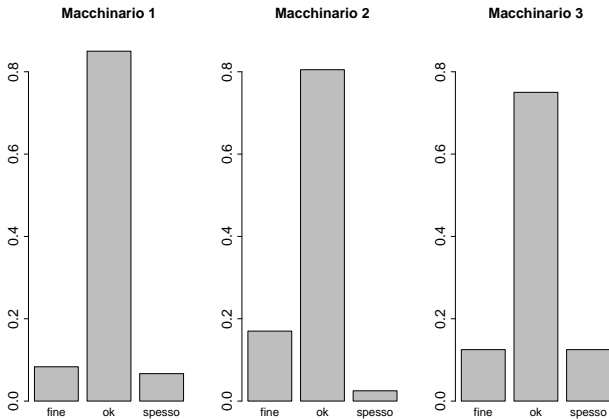
Perni

Considerando i dati riferiti alla produzione dei perni, si determinano le distribuzioni di frequenza relativa riferite al diametro, condizionatamente al macchinario di produzione.

```
perni <- rbind(cbind(rep("M1",10), rep("Fine",10)),
               cbind(rep("M1",102), rep("Ok",102)),
               cbind(rep("M1",8), rep("Spesso",8)),
               cbind(rep("M2",34), rep("Fine",34)),
               cbind(rep("M2",161), rep("Ok",161)),
               cbind(rep("M2",5), rep("Spesso",5)),
               cbind(rep("M3",10), rep("Fine",10)),
               cbind(rep("M3",60), rep("Ok",60)),
               cbind(rep("M3",10), rep("Spesso",10)))
perni <- as.data.frame(perni) # matrice trasformata in data frame
colnames(perni) <- c("Macchinario","Diametro") # nomi delle colonne
# tabella frequenze relative di Diametro per Macchinario
freq.rel <- table(perni$Macchinario,
                  perni$Diametro)/apply(table(perni$Macchinario,
                  perni$Diametro),1,sum)
```

```
par(mfrow=c(1,3))  
barplot(freq.rel[1,],names.arg=c("fine","ok","spesso"),  
        cex.axis=2,cex.names=1.7,main="Macchinario 1", cex.main=2,  
        ylim=c(0,0.9))  
barplot(freq.rel[2,],names.arg=c("fine","ok","spesso"),cex.axis=2,  
        cex.names=1.7,main="Macchinario 2", cex.main=2,  
        ylim=c(0,0.9))  
barplot(freq.rel[3,],names.arg=c("fine","ok","spesso"),cex.axis=2,  
        cex.names=1.7,main="Macchinario 3", cex.main=2,  
        ylim=c(0,0.9))
```

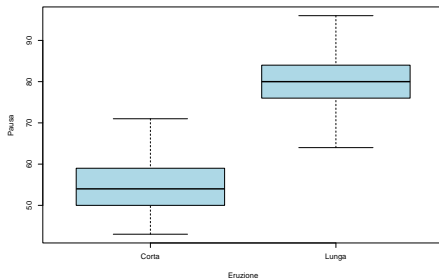
```
par(mfrow=c(1,1))
```



Geyser Old Faithful

Si considerano i dati riferiti alle eruzioni del geyser Old Faithful. Si analizzano le distribuzioni di frequenza relativa riferite alla durata delle pause, condizionatamente alla tipologia dell'eruzione precedente.

```
duration<-ifelse(faithful$eruptions < 3,"Corta", "Lunga")
duration<-factor(duration)
faithful1<-data.frame(Pausa=faithful$waiting,Eruzione=duration)
boxplot(Pausa~Eruzione,data=faithful1,col="lightblue")
```



Section 2

Analisi della connessione

Attitudini

Si analizza l'attitudine musicale X e pittorica Y di $n = 15$ individui con la seguente scala di modalità: sufficiente (S), buona (B), ottima (O). I dati vengono sintetizzati nella seguente tabella di frequenza congiunta, detta tabella di contingenza.

```
attitudine <- rbind(cbind(rep("S",1),rep("S",1)),cbind(rep("S",3),
rep("B",3)),cbind(rep("B",1),rep("S",1)),cbind(rep("B",3),
rep("B",3)),cbind(rep("B",2),rep("O",2)),cbind(rep("O",2),
rep("S",2)),cbind(rep("O",1),rep("B",1)),cbind(rep("O",2),rep("O",2)))
attitudine <- as.data.frame(attitudine) # matrice trasformata
# in data frame
colnames(attitudine) <- c("X","Y") # nomi delle colonne
attitudine$X <- ordered(attitudine$X, levels=c("S","B","O"))
attitudine$Y <- ordered(attitudine$Y, levels=c("S","B","O"))
str(attitudine)
```

```
## 'data.frame':    15 obs. of  2 variables:
## $ X: Ord.factor w/ 3 levels "S"<"B"<"O": 1 1 1 1 2 2 2 2 2 2 ...
## $ Y: Ord.factor w/ 3 levels "S"<"B"<"O": 1 2 2 2 1 2 2 2 3 3 ...
```

```
tab <- table(attitudine$X,attitudine$Y) # tabella di contingenza
#(distribuzione di frequenza assoluta congiunta)
```

```
tab
```

```
##
```

```
##      S B 0
```

```
##     S 1 3 0
```

```
##     B 1 3 2
```

```
##     0 2 1 2
```

```
# distribuzione marginale di X
```

```
# (frequenza assoluta)
```

```
margin.table(tab,1)
```

```
##
```

```
## S B 0
```

```
## 4 6 5
```

```
# distribuzione marginale di Y  
# (frequenza assoluta)  
margin.table(tab,2)
```

```
##  
## S B 0  
## 4 7 4
```

```
# distribuzione condizionata di Y/X=S (frequenza assoluta)  
tab[1,]
```

```
## S B 0  
## 1 3 0
```

```
# distribuzione condizionata di Y/X=B (frequenza assoluta)  
tab[2,]
```

```
## S B 0  
## 1 3 2
```

```
tab[3,] # distribuzione condizionata di  $Y/X=0$  (frequenza assoluta)
```

```
## S B 0
```

```
## 2 1 2
```

```
tab[,1] # distribuzione condizionata di  $X/Y=S$  (frequenza assoluta)
```

```
## S B 0
```

```
## 1 1 2
```

```
tab[,2] # distribuzione condizionata di  $X/Y=B$  (frequenza assoluta)
```

```
## S B 0
```

```
## 3 3 1
```

```
tab[,3] # distribuzione condizionata di  $X/Y=0$  (frequenza assoluta)
```

```
## S B 0
```

```
## 0 2 2
```

```
tab/sum(tab) # distribuzione di frequenza relativa congiunta
```

```
##
##           S           B           0
##  S 0.06666667 0.20000000 0.00000000
##  B 0.06666667 0.20000000 0.13333333
##  0 0.13333333 0.06666667 0.13333333
```

```
# in alternativa, prop.table(tab)
```

```
# distribuzione marginale di X (frequenza relativa)
margin.table(tab,1)/sum(margin.table(tab,1))
```

```
##
##           S           B           0
## 0.26666667 0.40000000 0.33333333
```

```
# distribuzione marginale di Y (frequenza relativa)
margin.table(tab,2)/sum(margin.table(tab,2))
```

```
##
##          S          B          0
## 0.2666667 0.4666667 0.2666667
```

```
# distribuzione condizionata di Y/X=S (frequenza relativa)
tab[1,]/sum(tab[1,])
```

```
##      S      B      0
## 0.25 0.75 0.00
```

```
# distribuzione condizionata di Y/X=B (frequenza relativa)
tab[2,]/sum(tab[2,])
```

```
##          S          B          0
## 0.1666667 0.5000000 0.3333333
```

```
# distribuzione condizionata di  $Y/X=0$  (frequenza relativa)
# in alternativa, prop.table(tab,1)
tab[3,]/sum(tab[3,])
```

```
##      S      B      0
## 0.4 0.2 0.4
```

```
# distribuzione condizionata di  $X/Y=S$  (frequenza relativa)
tab[,1]/sum(tab[,1])
```

```
##      S      B      0
## 0.25 0.25 0.50
```

```
# distribuzione condizionata di  $X/Y=B$  (frequenza relativa)
tab[,2]/sum(tab[,2])
```

```
##              S              B              0
## 0.4285714 0.4285714 0.1428571
```

```
# distribuzione condizionata di  $X/Y=0$  (frequenza assoluta)
tab[,3]/sum(tab[,3])
```

```
##      S      B      0
## 0.0 0.5 0.5
```

```
# in alternativa, prop.table(tab,2)
xtot <- margin.table(tab,1) # distribuzione marginale di X
ytot <- margin.table(tab,2) # distribuzione marginale di Y
xtot <- as.matrix(xtot) # vettore interpretato come matrice
ytot <- as.matrix(ytot) # vettore interpretato come matrice
```

Con riferimento alla analisi delle attitudini musicale e pittorica, se ci fosse indipendenza tra le due, ferme restando le distribuzioni marginali, si avrebbe la seguente tabella di contingenza:

```
tab_ind <- xtot%*%t(ytot)/sum(xtot) # tabella di contingenza
# in caso di indipendenza
```



```
tab_ind
```

```
##           S           B           O
## S 1.066667 1.866667 1.066667
## B 1.600000 2.800000 1.600000
## O 1.333333 2.333333 1.333333
```

```
chisq_obsstat <- sum((tab-tab_ind)^2/tab_ind) # indice di
# connessione chi-quadrato
chisq_obsstat
```

```
## [1] 3.526786
```

```
# in alternativa si puo' usare il comando
# summary(tab) oppure chisq.test(tab)
chisq_obsstat/(sum(tab)*min(c(length(tab[,1])-1,length(tab[,1])-1)))
```

```
## [1] 0.1175595
```

Casco

La seguente tabella di contingenza illustra i risultati di uno studio sull'efficacia dei caschi protettivi per ciclisti nella prevenzione dei traumi cranici. Si considerano $n = 793$ soggetti coinvolti in incidenti.

```
tab_casco<-matrix(c(17,218,130,428),nrow = 2,byrow = TRUE)
tab_casco<-as.table(tab_casco) # per trasformare la matrice
# in una tabella di contingenza
xtot_casco<-margin.table(tab_casco,1)
ytot_casco<-margin.table(tab_casco,2)
xtot_casco <- as.matrix(xtot_casco) # vettore interpretato
# come matrice
ytot_casco <- as.matrix(ytot_casco) # vettore interpretato
# come matrice
```

```
# tabella di contingenza in caso di indipendenza
```

```
tab_ind_casco <- xtot_casco%*%t(ytot_casco)/sum(xtot_casco)
tab_ind_casco
```

```
##           A           B
## A  43.56242 191.4376
## B 103.43758 454.5624
```

```
chisq_casco <- sum((tab_casco-tab_ind_casco)^2/tab_ind_casco)
chisq_casco
```

```
## [1] 28.2555
```

```
summary(tab_casco)
```

```
## Number of cases in table: 793
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 28.255, df = 1, p-value = 1.063e-07
```

```
chisq_casco/(sum(tab_casco)*min(c(length(tab_casco[1,])-1,  
                                length(tab_casco[,1])-1)))
```

```
## [1] 0.03563114
```

Section 3

Dipendenza in media

Old Faithful Geyser Data

Si considerano i dati riferiti alle durate delle eruzioni del geyser Old Faithful e si indica con X il tipo di eruzione e con Y la durata della pausa.

```
duration <- ifelse(faithful$eruptions < 3,"Corta", "Lunga")
duration <- factor(duration)
faithful1 <- data.frame(Pausa=faithful$waiting,Eruzione=duration)
summary(faithful1$Pausa[faithful1$Eruzione=="Corta"])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  43.00   50.00   54.00   54.49   59.00   71.00
```

```
summary(faithful1$Pausa[faithful1$Eruzione=="Lunga"])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  64.00   76.00   80.00   79.99   84.00   96.00
```

```
var(faithful1$Pausa[faithful1$Eruzione=="Corta"])
```

```
## [1] 34.10674
```

```
var(faithful1$Pausa[faithful1$Eruzione=="Lunga"])
```

```
## [1] 35.9309
```

```
library("moments")
```

```
## Warning: il pacchetto 'moments' è stato creato con R versione 4.1
```

```
skewness(faithful1$Pausa[faithful1$Eruzione=="Corta"])
```

```
## [1] 0.3027638
```

```
skewness(faithful1$Pausa[faithful1$Eruzione=="Lunga"])
```

```
## [1] -0.04280721
```

```
kurtosis(faithful1$Pausa[faithful1$Eruzione=="Corta"])
```

```
## [1] 2.557099
```

```
kurtosis(faithful1$Pausa[faithful1$Eruzione=="Lunga"])
```

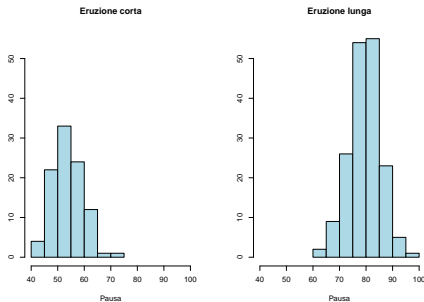
```
## [1] 3.054457
```



```

par(mfrow=c(1,2))
hist(faithful1$Pausa[faithful1$Eruzione=="Corta"],xlim=c(40,100),
     ylim=c(0,55),xlab="Pausa",ylab=" ",col="lightblue",
     main="Eruzione corta")
hist(faithful1$Pausa[faithful1$Eruzione=="Lunga"],xlim=c(40,100),
     ylim=c(0,55),xlab="Pausa",ylab=" ",col="lightblue",
     main="Eruzione lunga")

```



```

par(mfrow=c(1,1))

```

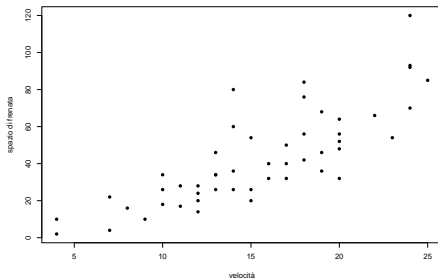
Section 4

Analisi di correlazione

Velocita'

Si considerano i dati sulla velocità X e sullo spazio di frenata Y di automobili degli anni 20.

```
plot(cars$speed,cars$dist,main=" ",xlab="velocità",  
      ylab="spazio di frenata",pch=16)
```



```
# covarianza, tenendo conto che cov() divide per n-1 invece  
# che per n  
cov(cars$speed, cars$dist)*(length(cars[,1])-1)/length(cars[,1])
```

```
## [1] 107.748
```

```
# regola di calcolo  
mean(cars$speed*cars$dist)-mean(cars$speed)*mean(cars$dist)
```

```
## [1] 107.748
```

```
# coefficiente di correlazione lineare,  
# in alternativa cov(cars$speed, cars$dist)/  
#sqrt(var(cars$speed)*var(cars$dist))  
cor(cars$speed, cars$dist)
```

```
## [1] 0.8068949
```

Esempio 1 correlazione

```
par(mfrow=c(2,2))  
set.seed(1) # si fissa il seed della procedura di simulazione  
x1 <- rnorm(100) # valori per X: simulazione di 100 valori  
# gaussiani  
y1 <- x1 # valori per Y  
cor(x1,y1) # coefficiente di correlazione lineare
```

```
## [1] 1
```

```
plot(x1,y1,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=1") # scatterplot  
abline(lm(y1~x1),lwd=2,col='red')  
# descrizione del legame lineare tra X e Y
```

```
x2 <- x1  
y2 <- x2*0.8+rnorm(100,0,0.5)  
cor(x2,y2)
```

```
## [1] 0.8319342
```

```
plot(x2,y2,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=0.83")  
abline(lm(y2~x2),lwd=2,col='red')
```

```
x3 <- x1  
y3 <- x3*0.3+rnorm(100,0,0.5)  
cor(x3,y3)
```

```
## [1] 0.4748344
```

```
plot(x3,y3,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=0.49")  
abline(lm(y3~x3),lwd=2,col='red')
```

```
x4 <- x1  
y4 <- x4*0.1+rnorm(100,0,0.5)  
cor(x4,y4)
```

```
## [1] 0.1297707
```

```
plot(x4,y4,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=0.18")  
abline(lm(y4~x4),lwd=2,col='red')
```



```
x5 <- x1  
y5 <- x5*(-0.001)+rnorm(100,0,0.5)  
cor(x5,y5)
```

```
## [1] 0.145548
```

```
plot(x5,y5,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=-0.003")  
abline(lm(y5~x5),lwd=2,col='red')
```

```
x6 <- x1  
y6 <- x6*(-0.3)+rnorm(100,0,0.5)  
cor(x6,y6)
```

```
## [1] -0.5576487
```

```
plot(x6,y6,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=-0.49")  
abline(lm(y6~x6),lwd=2,col='red')
```

```
x7 <- x1  
y7 <- x7*-0.8+rnorm(100,0,0.5)  
cor(x7,y7)
```

```
## [1] -0.7803826
```

```
plot(x7,y7,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=-0.83")  
abline(lm(y7~x7),lwd=2,col='red')
```

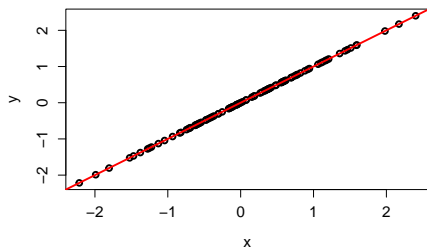
```
x8 <- x1  
y8 <- -x8  
cor(x8,y8)
```

```
## [1] -1
```

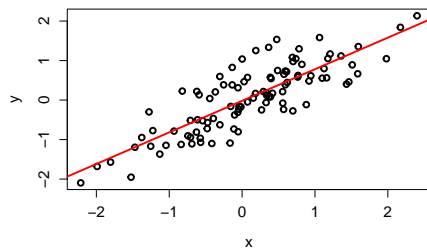
```
plot(x8,y8,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2,  
      cex.lab=1.2,main="r=-1")  
abline(lm(y8~x8),lwd=2,col='red')
```

```
par(mfrow=c(1,1))
```

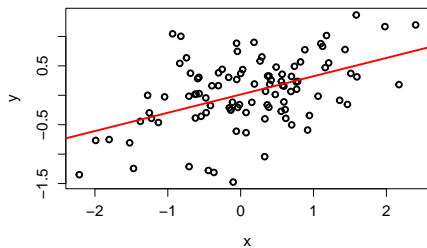
$r=1$



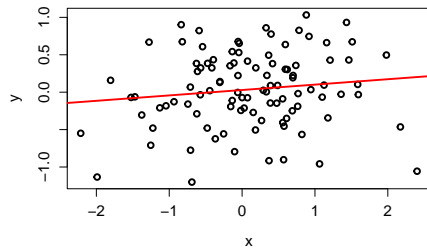
$r=0.83$



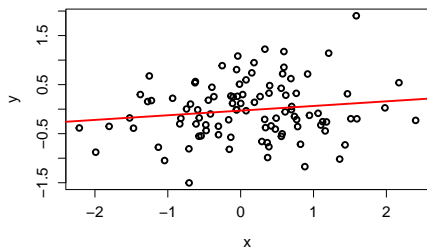
$r=0.49$



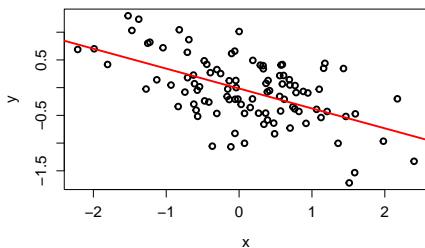
$r=0.18$



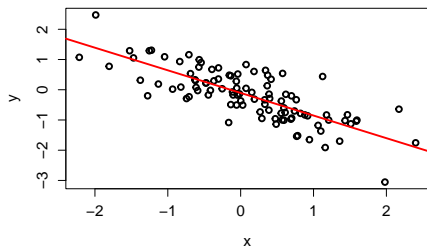
$r = -0.003$



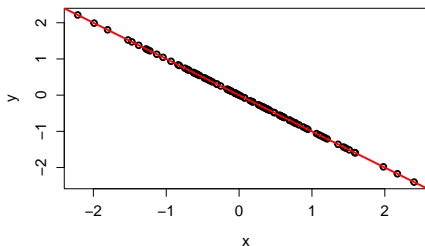
$r = -0.49$



$r = -0.83$



$r = -1$

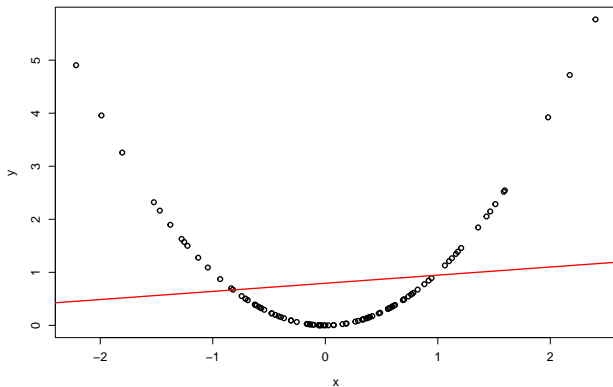


Esempio 2 correlazione

```
set.seed(1)
x <- rnorm(100)
y <- x^2
cor(x,y)
```

```
## [1] 0.1202777
```

```
plot(x,y,pch=1,lwd=2,xlab="x",yla ="y",cex.axis=1.2, cex.lab=1.2)  
abline(lm(y~x),lwd=2,col='red')
```



Molla

Si considerano i dati sulla lunghezza della molla Y e sugli $n = 20$ diversi pesi X a cui viene sottoposta. Si vuole studiare la relazione tra X e Y e, più precisamente, verificare se X spiega Y .

```
yy <- c(5.06,5.01,5.12,5.13,5.14,5.16,5.25,5.19,5.24,5.46,5.4,
        5.57,5.47,5.53,5.61,5.59,5.61,5.75,5.68,5.8)
xx <- c(0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2,2.2,2.4,2.6,
        2.8,3,3.2,3.4,3.6,3.8)
mean(xx)
```

```
## [1] 1.9
```

```
mean(yy)
```

```
## [1] 5.3885
```

```
var(xx)*19/20
```

```
## [1] 1.33
```

```
cov(xx,yy)*19/20
```

```
## [1] 0.27215
```

```
cor(xx,yy) # coefficiente di correlazione lineare
```

```
## [1] 0.9743193
```

```
cor(xx,yy,method = "spearman") # coefficiente di correlazione di Spearman
```

```
## [1] 0.9838286
```

```
cor(xx,yy,method = "kendall") # coefficiente di correlazione di Kendall
```

```
## [1] 0.9129319
```

```
yy[19] <- 5.01 # valore anomalo
```

```
cor(xx,yy)
```

```
## [1] 0.7589276
```

```
cor(xx,yy,method = "spearman")
```

```
## [1] 0.741159
```

```
cor(xx,yy,method = "kendall")
```

```
## [1] 0.7407511
```

Section 5

Analisi di regressione

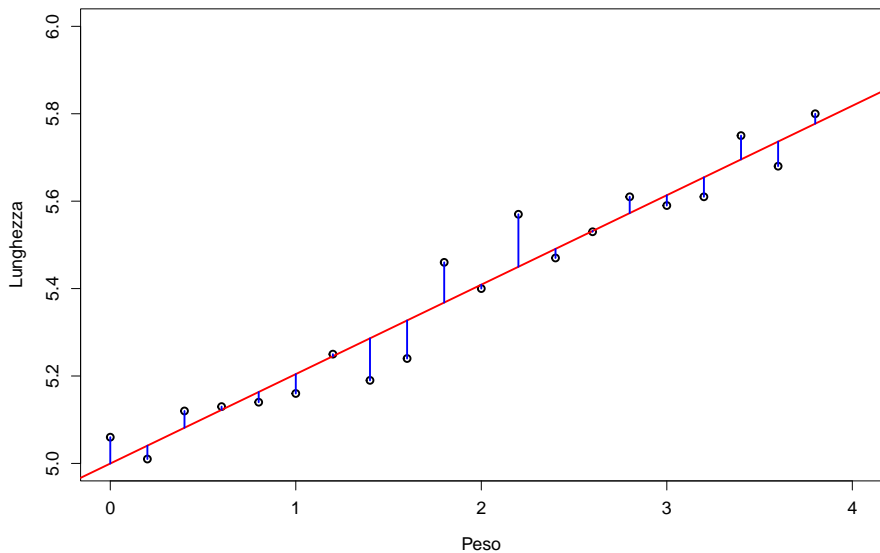
Molla

```
yy <- c(5.06,5.01,5.12,5.13,5.14,5.16,5.25,5.19,5.24,5.46,5.4,
        5.57,5.47,5.53,5.61,5.59,5.61,5.75,5.68,5.8)
xx <- c(0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2,2.2,2.4,2.6,2.8,
        3,3.2,3.4,3.6,3.8)
modello <- lm(yy~xx) # stima del modello lineare
modello # stime dei minimi quadrati, oppure in alternativa
```

```
##
## Call:
## lm(formula = yy ~ xx)
##
## Coefficients:
## (Intercept)          xx
##      4.9997      0.2046

# b <- (cov(xx,yy)*19/20)/(var(xx)*19/20)
# a <- mean(yy)-b*mean(xx)
```

```
plot(xx, yy, xlab = "Peso", ylab = "Lunghezza", lwd = 2,  
      cex.axis = 1.2, cex.lab = 1.2, ylim = c(5,6), xlim = c(0,4))  
# in alternativa abline(a,b,lwd=2,col='red')  
abline(modello, lwd = 2, col = 'red')  
segments(xx, yy, xx, fitted.values(modello), col = "blue", lwd = 2)
```



```
# i valori previsti (stimati) dal modello per
# Y, in corrispondenza ai valori osservati di
# X, in alternativa predict(modello)
fitted.values(modello)
```

```
##           1           2           3           4           5           6           7
## 4.999714 5.040639 5.081564 5.122489 5.163414 5.204338 5.245263 5.
##           9          10          11          12          13          14          15
## 5.327113 5.368038 5.408962 5.449887 5.490812 5.531737 5.572662 5.
##          17          18          19          20
## 5.654511 5.695436 5.736361 5.777286
```

```
residuals(modello) # i residui stimati
```

```
##           1           2           3           4           5
## 0.060285714 -0.030639098 0.038436090 0.007511278 -0.023413534
##           7           8           9          10          11
## 0.004736842 -0.096187970 -0.087112782 0.091962406 -0.008962406
##          13          14          15          16          17
## -0.020812030 -0.001736842 0.037338346 -0.023586466 -0.044511278
##          19          20
```



```
# valore previsto dal modello per x=2.5, in alternativa a+b*2.5  
predict(modello, newdata = data.frame(xx = c(2.5)))
```

```
##          1  
## 5.511274
```

```
# valore previsto dal modello per x=7.5, in alternativa a+b*7.5  
predict(modello, newdata = data.frame(xx = c(7.5)))
```

```
##          1  
## 6.534395
```

```
R2 <- var(fitted.values(modello))*19/(var(yy)*19)
R2
```

```
## [1] 0.9492981
```

```
# in alternativa 1-var(residuals(modello))*19/(var(yy)*19)
# o summary(modello)$r.squared
```

```
cor(xx,yy)^2
```

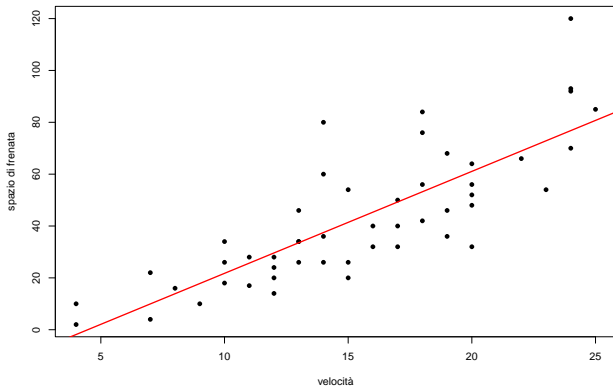
```
## [1] 0.9492981
```

Velocita'

```
modello_vel<-lm(dist~speed,data=cars)
# in alternativa modello_vel <- lm(cars$dist~cars$speed)
modello_vel
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)      speed
##      -17.579       3.932
```

```
plot(cars$speed,cars$dist,main=" ",xlab="velocità",  
      ylab="spazio di frenata",pch=16)  
abline(modello_vel,lwd=2,col='red')
```



```
fitted.values(modello_vel)
```

```
##           1           2           3           4           5           6
## -1.849460 -1.849460  9.947766  9.947766 13.880175 17.812584 21.74
##           9          10          11          12          13          14
## 21.744993 25.677401 25.677401 29.609810 29.609810 29.609810 29.60
##          17          18          19          20          21          22
## 33.542219 33.542219 33.542219 37.474628 37.474628 37.474628 37.47
##          25          26          27          28          29          30
## 41.407036 41.407036 45.339445 45.339445 49.271854 49.271854 49.27
##          33          34          35          36          37          38
## 53.204263 53.204263 53.204263 57.136672 57.136672 57.136672 61.06
##          41          42          43          44          45          46
## 61.069080 61.069080 61.069080 68.933898 72.866307 76.798715 76.79
##          49          50
## 76.798715 80.731124
```

```
residuals(modello_vel)
```

```
##          1          2          3          4          5          6
##  3.849460  11.849460 -5.947766  12.052234   2.119825 -7.812584
##          8          9         10         11         12         13
##  4.255007  12.255007 -8.677401   2.322599 -15.609810 -9.609810
##         15         16         17         18         19         20
## -1.609810 -7.542219   0.457781   0.457781  12.457781 -11.474628
##         22         23         24         25         26         27
##  22.525372  42.525372 -21.407036 -15.407036  12.592964 -13.339445
##         29         30         31         32         33         34
## -17.271854 -9.271854   0.728146 -11.204263   2.795737  22.795737
##         36         37         38         39         40         41
## -21.136672 -11.136672  10.863328 -29.069080 -13.069080 -9.069080
##         43         44         45         46         47         48
##   2.930920 -2.933898 -18.866307  -6.798715  15.201285  16.201285
##         50
##   4.268876
```

```
predict(modello_vel,newdata = data.frame(speed=c(13,23,28)))
```

```
##           1           2           3  
## 33.54222 72.86631 92.52835
```

```
summary(modello_vel)$r.squared
```

```
## [1] 0.6510794
```



```
modello_vel1 <- lm(sqrt(cars$dist)~cars$speed)
# modello con risposta sqrt(Y), in alternativa
# modello_vel1 <- lm(sqrt(dist)~speed,data=cars)
modello_vel1
```

```
##
## Call:
## lm(formula = sqrt(cars$dist) ~ cars$speed)
##
## Coefficients:
## (Intercept)    cars$speed
##          1.2771          0.3224
```

```
summary(modello_vel1)$r.squared
```

```
## [1] 0.7094131
```

```

par(mfrow=c(1,2))
plot(cars$speed,sqrt(cars$dist),main=" ",xlab="velocità",
ylab="radice quadrata spazio di frenata",pch=16)
abline(modello_vel1,lwd=2,col='red')
plot(cars$speed,cars$dist,main=" ",xlab="velocità",
ylab="spazio di frenata",pch=16) # dati nella scala originaria
lines(seq(3,27,0.01),(1.2771+0.3224*seq(3,27,0.01))^2,
lwd=2,col='red') # retta di regressione riportata nella scala

```

