(Bio)Statistica con R

Parte I

UNIVERSITÀ DEGLI STUDI DI FOGGIA

Dr. Crescenzio Gallo – Università di Foggia – Dipartimenti di Area Medica

(Bio)Statistica con R — Parte I

- In questa Parte I dell'esercitazione di (bio)statistica con R acquisiremo il dataset di riferimento "Database_Covid_unico_Puglia" che memorizzeremo nella variabile dataframe "dati" di R ed utilizzeremo per lo svolgimento degli esempi ed esercizi.
- Per acquisire il dataset (che si trova in un server remoto) basta eseguire il seguente comando nella sessione R:

```
> dati <- read.csv2(
"https://www.crescenziogallo.it/pub/Database_Covid_unico_Puglia.csv",
header=TRUE, sep = "\t")
```

- L'istruzione read.csv2 legge un file di testo in formato CSV "italiano" (punto decimale ","); il file ha una prima riga di intestazione (*header = TRUE*) e con campi separati dal carattere di tabulazione (*sep = "\t"*).
- Il contenuto del file di testo viene memorizzato nella variabile *dati*, che è di tipo "dataframe". Per verificarlo basta eseguire il comando:

```
> class(dati)
[1] "data.frame"
```

(Bio)Statistica con R — Parte I

- Dopo aver acquisito il dataset, occorre definire esplicitamente le variabili categoriche mediante l'istruzione *factor* e le variabili di tipo data mediante l'istruzione *as.Date*:

```
> dati$Sede = factor(dati$Sede)
> dati$BARI = factor(dati$BARI)
> dati$BRINDISI = factor(dati$BRINDISI)
> dati$FOGGIA = factor(dati$FOGGIA)
> dati$MIULLI = factor(dati$MIULLI)
> dati$Data_Ingresso = as.Date(dati$Data_Ingresso, format("%d/%m/%Y"))
> dati$Age65 = factor(dati$Age65)
> dati$Sesso = factor(dati$Sesso)
> dati$Asma = factor(dati$Asma)
> dati$BPCO = factor(dati$BPCO)
> dati$Diabete = factor(dati$Diabete)
> dati$Mal_Neurologiche = factor(dati$Mal_Neurologiche)
> dati$Cardiopatie = factor(dati$Cardiopatie)
> dati$Neoplasia = factor(dati$Neoplasia)
> dati$Punti_Età = factor(dati$Punti_Età)
> dati$Charlson_Index = factor(dati$Charlson_Index)
> dati$Insuff_Renale = factor(dati$Insuff_Renale)
> dati$Comr_2più = factor(dati$Comr_2più)
> dati$Comr_3più = factor(dati$Comr_3più)
> dati$Deceduto = factor(dati$Deceduto)
> dati$Data_Dimissione_Decesso = as.Date(dati$Data_Dimissione_Decesso, format("%d/%m/%Y"))
> dati$Deceduto_a_30gg = factor(dati$Deceduto_a_30gg)
> dati$Supporto_Respiratorio = factor(dati$Supporto_Respiratorio)
```

(Bio)Statistica con R — Parte I

- Infine, salviamo il dataframe così strutturato:

```
> save(dati, file = "Database_Covid_unico_Puglia.RData")
```

- L'istruzione *save* scrive una rappresentazione esterna di oggetti R nel file specificato.
- Gli oggetti possono essere riletti dal file in un secondo momento usando la funzione *load* o *attach* (o *data* in alcuni casi).
- Oltre a *file*, è possibile specificare anche altri parametri, come "*ascii = TRUE*" (se si desidera una rappresentazione testuale invece che in binario dei dati).
- Nel caso si sia appena aperta una nuova sessione R, è quindi possibile leggere il dataframe *dati* con il seguente comando:

```
> load("Database_Covid_unico_Puglia.RData")
```

(Bio)Statistica con R — Parte I

Statistiche univariate

- Dopo aver caricato il dataframe *dati* nella sessione R ne possiamo esaminare la struttura con il comando *str*:

```
> str(dati)
'data.frame':  521 obs. of  42 variables:
 $ Sede                    : Factor w/ 4 levels "BARI","BRINDISI",..: 1 1 1 1 1 1 1 1 1 1 …
…
 $ Età                     : int  72 88 92 58 64 76 80 43 62 50 ...
```

- Effettuiamo alcune semplici statistiche sulla variabile numerica *Età*:

```
> summary(dati$Età)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13.00   55.00   69.00   67.18   82.00   99.00
```

- Sulla variabile categorica *Sede* ovviamente possiamo solo conteggiare le frequenze assolute e relative:

```
> table(dati$Sede)
BARI BRINDISI       FG   MIULLI
  97       36      161      227
> prop.table(table(dati$Sede))
      BARI   BRINDISI         FG     MIULLI
0.18618042 0.06909789 0.30902111 0.43570058
```

(Bio)Statistica con R — Parte I

Statistiche univariate

- Visualizzo l'istogramma della variabile *Età*:

```
> hist(dati$Età, main="Età pazienti", ylab="frequenza", xlab="classe di età", col = "blue")
```

- Estraggo nel dataframe *df* alcune colonne del dataset:

```
> df = dati[,c("Sede","Età","Sesso","Fumatore","Asma","BPCO","Charlson_Index")]
```

- Esamino la distribuzione della variabile categorica *Sesso*; il risultato presenta le frequenze assolute associate ai due valori che la variabile può assumere:

```
> summary(df$Sesso)
```

- Lo stesso risultato si ottiene con la funzione *table*():

```
> table(df$Sesso)
```

- Si può calcolare la tabella a doppia entrata che incrocia il *Sesso* e la provenienza (*Sede*):

```
> table(df$Sede,df$Sesso)
```

- Per ottenere le frequenze relative è sufficiente utilizzare la funzione di **R prop.table**():

```
> prop.table(table(df$Sesso))
> prop.table(table(df$Sede, df$Sesso))
```

(Bio)Statistica con R — Parte I

Statistiche univariate

- Per mettere in grafico la distribuzione della variabile *Sesso* possiamo usare la funzione barplot():

```
> barplot(prop.table(table(df$Sesso)))
```

- In alternativa all'istogramma, la distribuzione di una variabile categorica può essere visualizzata usando un diagramma a torta, sempre a partire dalla tabella:

```
> pie(table(df$Sesso))
```

- Studiamo la distribuzione della variabile numerica *Età* (in totale e per sede). R fornisce diverse informazioni sulla distribuzione di una variabile numerica; per la precisione, il minimo, il massimo, il primo, il secondo (la mediana) e il terzo quartile della distribuzione.

```
> summary(df$Età)
> EtàBA <- subset(df$Età, df$Sede=="BARI")
> summary(EtàBA)
> EtàBR <- subset(df$Età, df$Sede=="BRINDISI")
> summary(EtàBR)
> EtàFG <- subset(df$Età, df$Sede=="FG")
> summary(EtàFG)
> EtàMIULLI <- subset(df$Età, df$Sede=="MIULLI")
> summary(EtàMIULLI)
```

(Bio)Statistica con R — Parte I

Statistiche univariate

- In un boxplot, la "scatola" (box) centrale marca il 50% centrale della distribuzione (dal primo al terzo quartile), la linea in grassetto al centro rappresenta la mediana, e i due baffi esterni sono il minimo e il massimo della distribuzione. Ci sono molte funzioni di **R** che usano il boxplot. Nel pacchetto di grafica standard, quello che viene caricato automaticamente all'installazione, la funzione si chiama appunto **boxplot**():

```
> boxplot(df$Età, ylab = "Età (anni)")
```

- Esaminiamo l'istogramma. Il parametro **freq=TRUE** consente di ottenere le frequenze; per ottenere le densità occorre specificare invece **freq=FALSE**. Il parametro *right* consente di definire l'estremità da considerare nell'intervallo di classificazione: con **right=FALSE** non viene compreso l'estremo destro delle classi.

```
> hist(df$Età, freq = TRUE, right = FALSE, main = "Istrogramma età pazienti",
  xlab = "Età", ylab = "Frequenza")
```

(Bio)Statistica con R — Parte I

Statistiche univariate

- Esaminiamo alcune misure di tendenza centrale:

```
> median(df$Età) # mediana
> t <- table(df$Età); moda <- attr(which(t==max(t)),"names") # moda
> mean(df$Età) # media aritmetica
> sqrt(mean(df$Età^2)) # media quadratica
> exp(mean(log(df$Età))) # media geometrica
> 1/mean(1/df$Età) # media armonica
```

- Per queste ultime medie è possibile installare il pacchetto **psych** e utilizzare le funzioni **geometric.mean**() e **harmonic.mean**():

```
> require(psych)
> geometric.mean(df$Età)
> harmonic.mean(df$Età)
```

(Bio)Statistica con R — Parte I

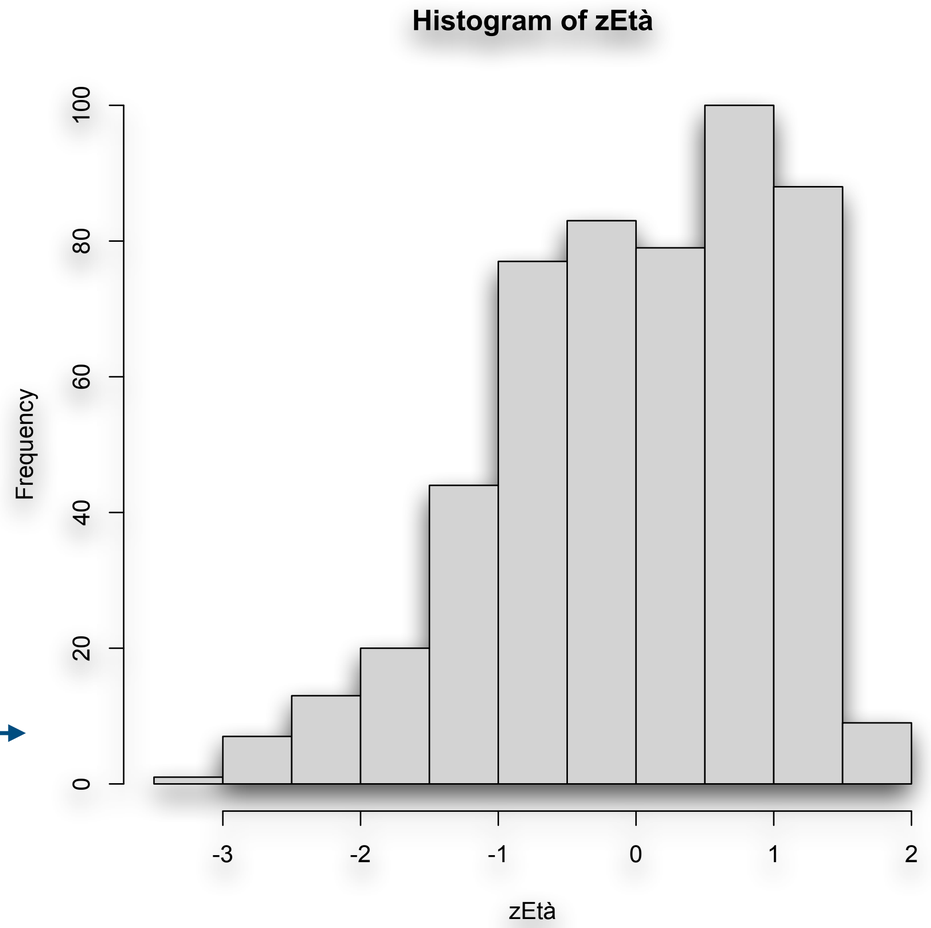
Statistiche univariate

- Misure di dispersione:

```
> range(df$Età)
> IQR(df$Età) # range interquartile
> sd(df$Età) # deviazione standard
> var(df$Età) # varianza
> mad(df$Età) # median absolute deviation – indicata per distribuzioni non normali
```

- Standardizzazione (un'operazione molto utile quando vogliamo confrontare fra loro gruppi di dati raccolti in condizioni diverse):

```
> zEtà <- scale(df$Età)
> hist(zEtà) # media=0 ds=1
```



(Bio)Statistica con R — Parte I

Statistiche univariate

- Esaminiamo la normalità della distribuzione sia graficamente (plot Q-Q normale) che tramite i test ufficiali del pacchetto *normtest*:

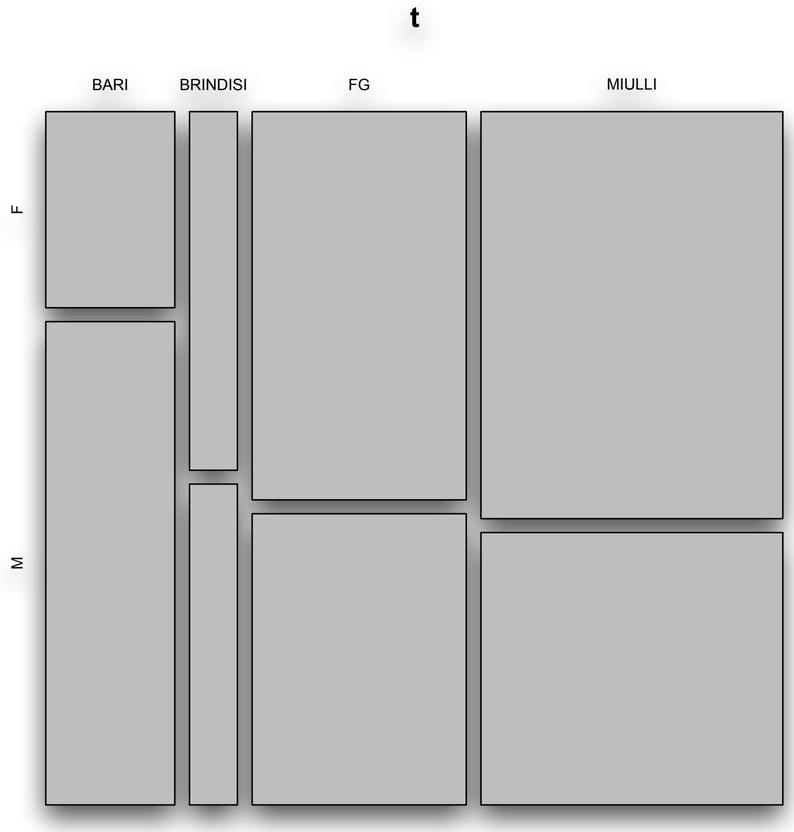
```
> qqnorm(df$Età) # plot Q-Q (grafico quantile-quantile) per una distribuzione normale
> qqline(df$Età) # linea per il plot Q-Q normale, che passa attraverso il primo e il terzo quartile
> require("nortest") # package per effettuare i test di normalità
> ad.test(df$Età) # test di Anderson-Darling: se p>0.05 la distribuzione è normale
> cvm.test(df$Età) # test di Cramer-von Mises: se p>0.05 la distribuzione è normale
> shapiro.test(df$Età) # test di Shapiro-Wilk: se p>0.05 la distribuzione è normale
> pearson.test(df$Età) # test di Pearson chi-Square: se p>0.05 la distribuzione è normale
> ks.test(df$Età,"pnorm") # test di Kolmogorov-Smirnov: se p>0.05 la distribuzione è normale
```

(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili categoriche

- Estraggo le colonne *Sede* e *Sesso* in una matrice 4×2 per effettuare i test e ne traccio il grafico a mosaico:

```
> t = as.matrix(table(df$Sede, df$Sesso))
> plot(t)
```



- Esamino l'associazione fra le variabili categoriche

```
> chisq <- chisq.test(t)
> chisq  # visualizzo il risultato della statistica chi-quadrato

Pearson's Chi-squared test
data:  t
X-squared = 28.12, df = 3, p-value = 3.427e-06

# X-squared è la somma degli scarti al quadrato fra le frequenze osservate e le frequenze attese
> str(chisq)  # ne esamino la struttura interna
List of 9
 $ statistic: Named num 28.1
  ..- attr(*, "names")= chr "X-squared"
…
```

(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili categoriche

- Per misurare l'associazione fra le variabili in una tabella come quella creata nella sezione precedente, usiamo la statistica *V* di Cramér che ha valore 0 nel caso di perfetta indipendenza e valore 1 nel caso di perfetta associazione.
- Il *V* di Cramér misura l'associazione utilizzando gli scarti al quadrato fra le frequenze osservate e le frequenze attese, espressi come proporzione delle frequenze attese.
- La somma di questi scarti è la statistica chi-quadrato, mentre *V* è la radice quadrata di chi-quadrato, diviso per il numero totale di osservazioni moltiplicato per il numero di righe o di colonne (scegliere il più piccolo se sono diversi), meno uno:

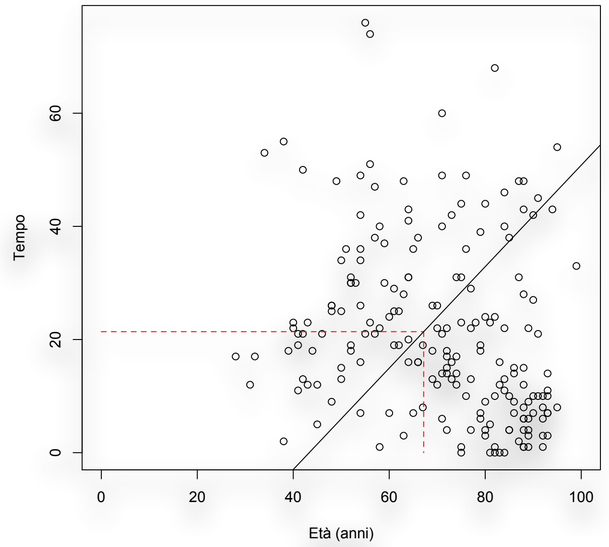
```
> V = sqrt(chisq$statistic[[1]]/sum(t)*(min(dim(t))-1)) # V=0.2323226
```

(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Per studiare la distribuzione bivariata di variabili numeriche lo strumento più utile è il diagramma di dispersione (scatterplot):

```
> plot(dati$Età, dati$Tempo, xlim = c(0,100), ylim = c(0,76), xlab = "Età (anni)",
  ylab = "Tempo")
> segments (mean(dati$Età), mean(dati$Tempo, na.rm = TRUE), mean(dati$Età), 0,
  lty = "dashed", col = "red")
> segments (0, mean(dati$Tempo, na.rm = TRUE), mean(dati$Età),
  mean(dati$Tempo, na.rm = TRUE), lty = "dashed", col = "red")
> x1 <- mean(dati$Età) - 3 * sd(dati$Età)
> x2 <- mean(dati$Età) + 3 * sd(dati$Età)
> y1 <- mean(dati$Tempo, na.rm = TRUE) - 3 * sd(dati$Tempo, na.rm = TRUE)
> y2 <- mean(dati$Tempo, na.rm = TRUE) + 3 * sd(dati$Tempo, na.rm = TRUE)
> segments (x1, y1, x2, y2)
```

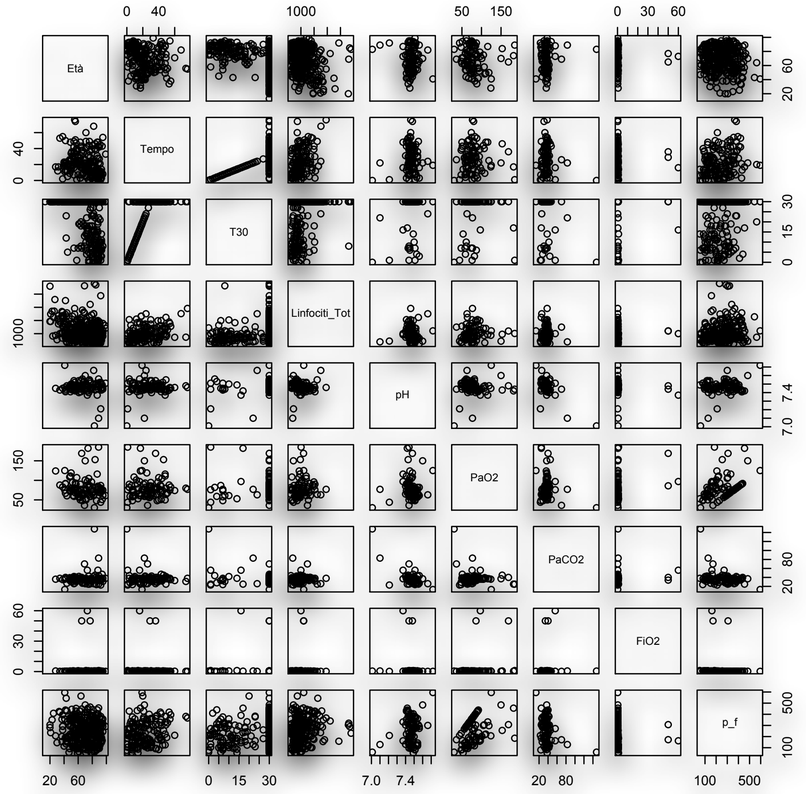


(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Il data set contiene molte altre variabili di cui ci potrebbe interessare la distribuzione bivariata. Per dare un'occhiata rapida usando la funzione **pairs**() possiamo creare una matrice di diagrammi di dispersione (tra le sole varibili numeriche, ovviamente), con tutte le coppie possibili.
- La funzione **pairs**() è un'ottima maniera di visualizzare la struttura complessiva di un set di dati, facendosi un'idea delle relazioni fra le diverse variabili:

```
> pairs(dati[, c("Età", "Tempo", "T30", "Linfociti_Tot",
  "pH", "PaO2", "PaCO2", "FiO2", "p_f")])
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Correlazione

```
> cor(dati$Età, dati$Tempo, use = "na.or.complete") # coeff. di correlazione di Pearson
> plot(scale(dati$Età), scale(dati$Tempo), xlab = "z(Età)", ylab = "z(Tempo)") # scatterplot dei dati standardizzati
> segments(-4, 0, 4, 0, lty = "dashed")
> segments(0, -2, 0, 6, lty = "dashed")
> cor.test(dati$Età, dati$Tempo, method = "spearman", use = "na.or.complete") # coefficiente di correlazione per ranghi di Spearman (cor.test fornisce anche il p-value)
> cor.test(dati$Età, dati$Tempo, method = "kendall", use = "na.or.complete") # coefficiente di correlazione per ranghi tau di Kendall
```

- Regressione lineare

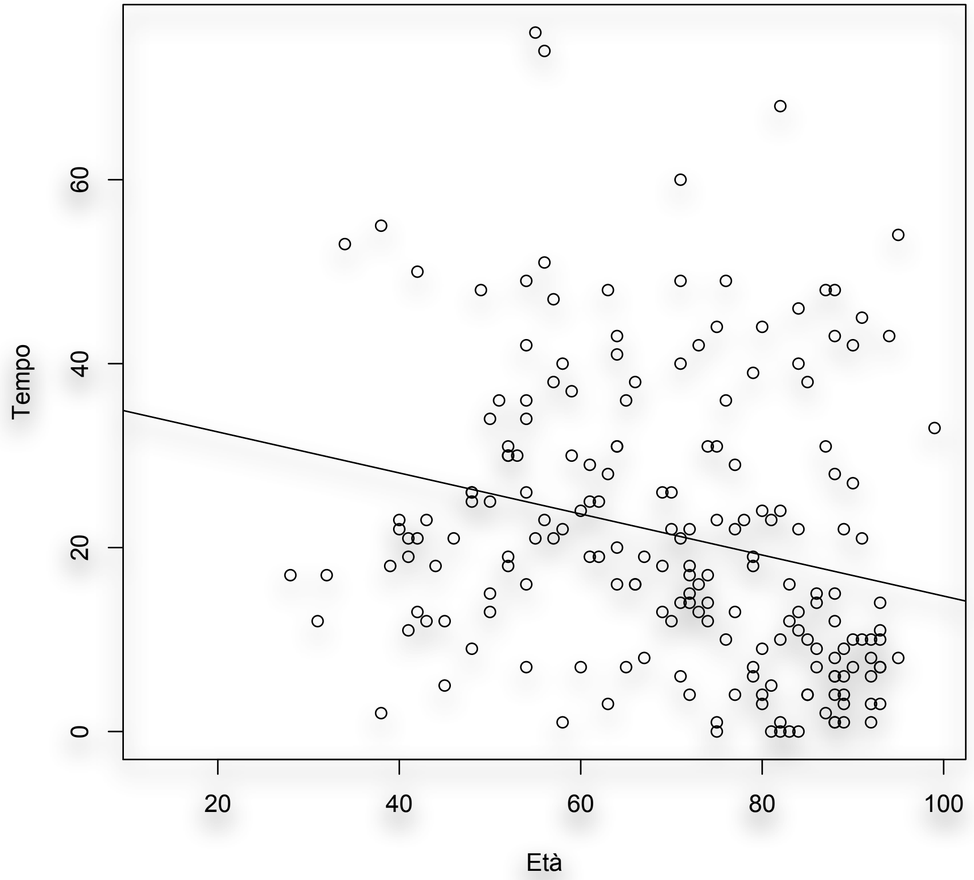
```
> modello <- lm(dati$Tempo ~ dati$Età)
> summary(modello)
```

(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- In forma grafica, la regressione lineare viene presentata sovrapponendo la retta fittata al diagramma di dispersione.
- La sintassi di **R** consente di farlo rapidamente sfruttando la funzione **abline()**, cui è possibile passare i parametri **a** e **b** direttamente da **lm**():

```
> plot(dati$Età, dati$Tempo, xlab = "Età",
  ylab = "Tempo")
> abline(lm(dati$Tempo ~ dati$Età))
```

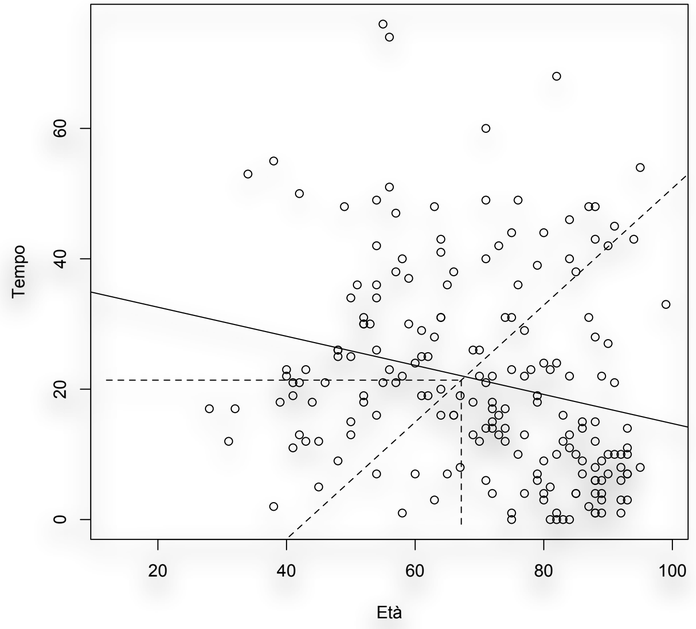


(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Per visualizzare in modo completo la distribuzione bivariata, includendo dei riferimenti sia per il modello che prevede *y* in funzione di *x* (regressione) sia per la visualizzazione dell'associazione lineare fra le due variabili, possiamo sovrapporre al grafico anche il punto delle medie e la retta delle deviazioni standard, come abbiamo fatto in precedenza.
- Per fare questo utilizziamo la funzione **bivd**() – che mostriamo successivamente – che prende in input quattro parametri: due vettori, di cui si vuole fare il diagramma di dispersione, e due stringhe alfanumeriche, che servono per etichettare correttamente gli assi a seconda di cosa si passa alla funzione:

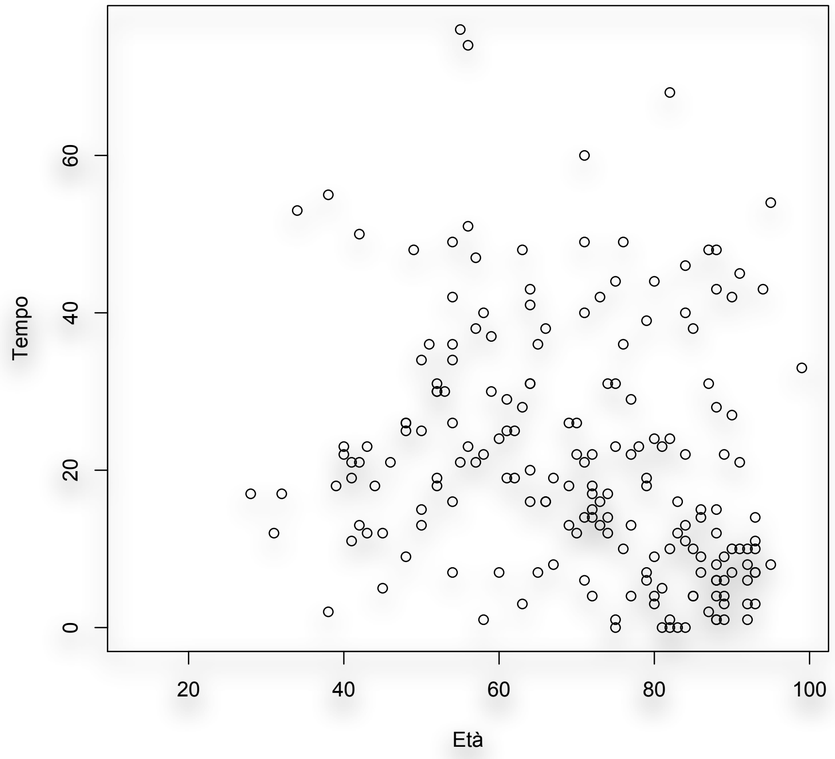
```
> source("bivd.R")
> bivd(dati$Età, dati$Tempo, "Età", "Tempo")
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

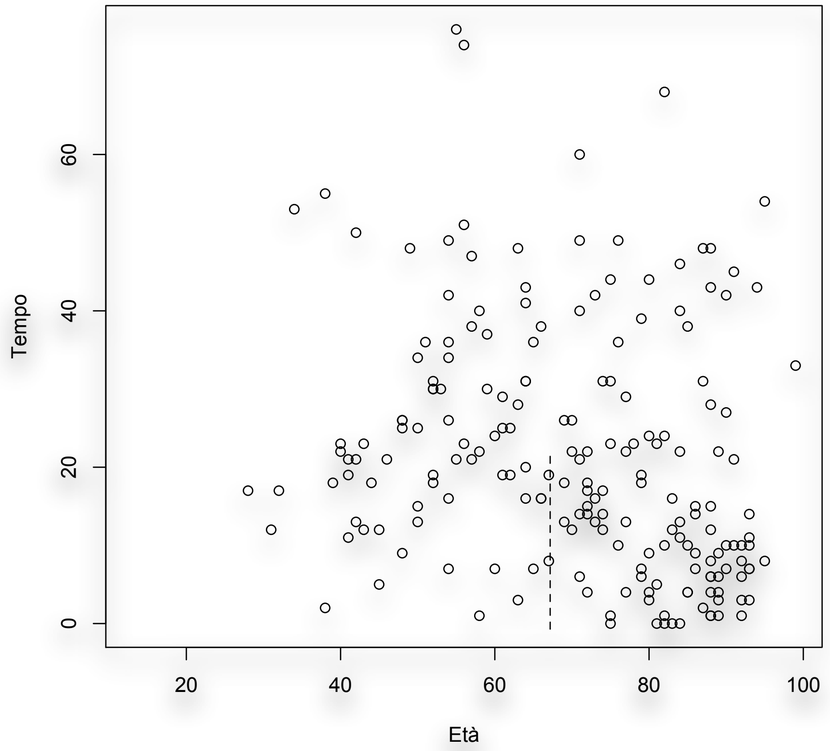
```
bivd <- function(x,y, xname, yname) {
   mx <- mean(x, na.rm = TRUE)
   my <- mean(y, na.rm = TRUE)
   sdx <- sd(x, na.rm = TRUE)
   sdy <- sd(y, na.rm = TRUE)
   plot(y ~ x, xlab = xname, ylab = yname)
   segments (mx, my, mx, min(y, na.rm = TRUE) - 1, lty = "dashed")
   segments (min(x, na.rm = TRUE) - 1, my, mx, my, lty = "dashed")
   x1 <- mx - 3 * sdx
   x2 <- mx + 3 * sdx
   y1 <- my - 3 * sdy
   y2 <- my + 3 * sdy
   segments (x1, y1, x2, y2, lty ="dashed")
   abline(lm(y ~ x))
}
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

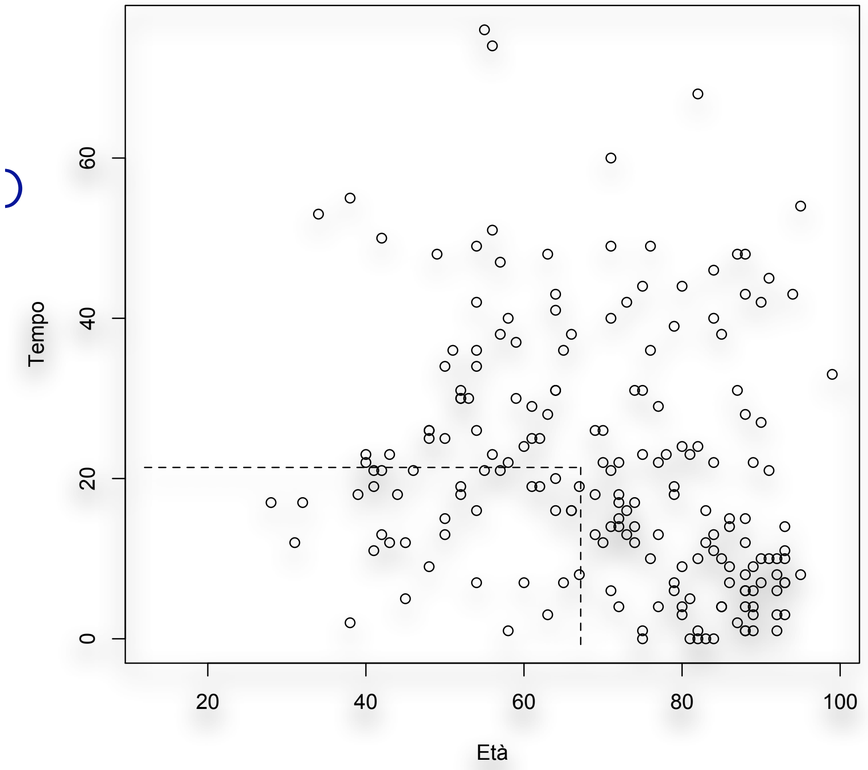
```
bivd <- function(x,y, xname, yname){
mx <- mean(x, na.rm = TRUE)
my <- mean(y, na.rm = TRUE)
sdx <- sd(x, na.rm = TRUE)
sdy <- sd(y, na.rm = TRUE)
plot(y ~ x, xlab = xname, ylab = yname)
segments (mx, my, mx, min(y, na.rm = TRUE) - 1, lty = "dashed")
segments (min(x, na.rm = TRUE) - 1, my, mx, my, lty =
"dashed")
x1 <- mx - 3 * sdx
x2 <- mx + 3 * sdx
y1 <- my - 3 * sdy
y2 <- my + 3 * sdy
segments (x1, y1, x2, y2, lty ="dashed")
abline(lm(y ~ x))
}
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

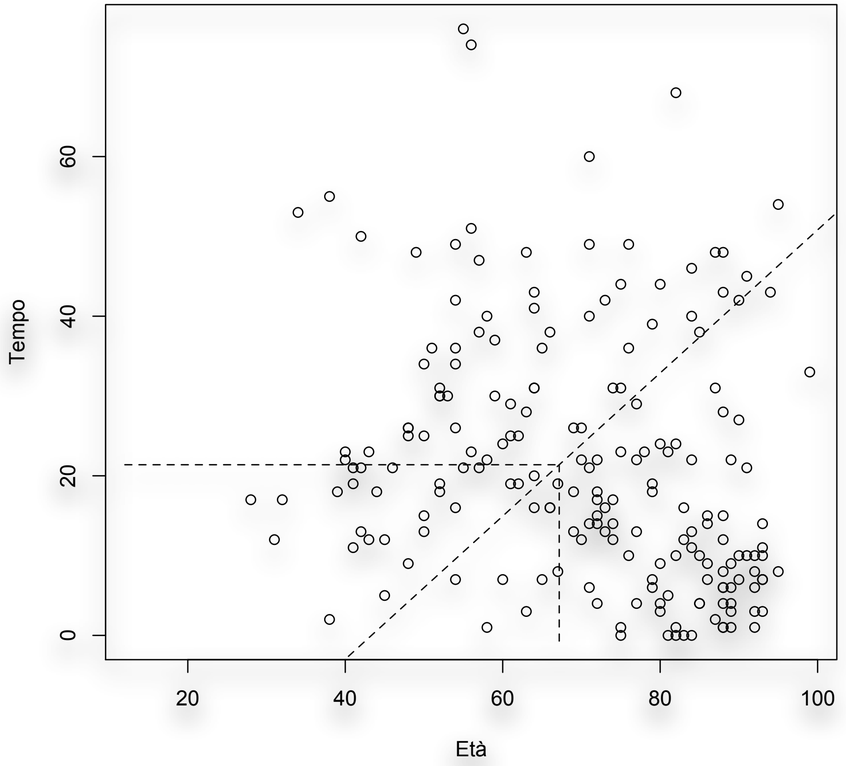
```
bivd <- function(x,y, xname, yname){
mx <- mean(x, na.rm = TRUE)
my <- mean(y, na.rm = TRUE)
sdx <- sd(x, na.rm = TRUE)
sdy <- sd(y, na.rm = TRUE)
plot(y ~ x, xlab = xname, ylab = yname)
segments (mx, my, mx, min(y, na.rm = TRUE) - 1, lty = "dashed")
segments (min(x, na.rm = TRUE) - 1, my, mx, my, lty = "dashed")
x1 <- mx - 3 * sdx
x2 <- mx + 3 * sdx
y1 <- my - 3 * sdy
y2 <- my + 3 * sdy
segments (x1, y1, x2, y2, lty ="dashed")
abline(lm(y ~ x))
}
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

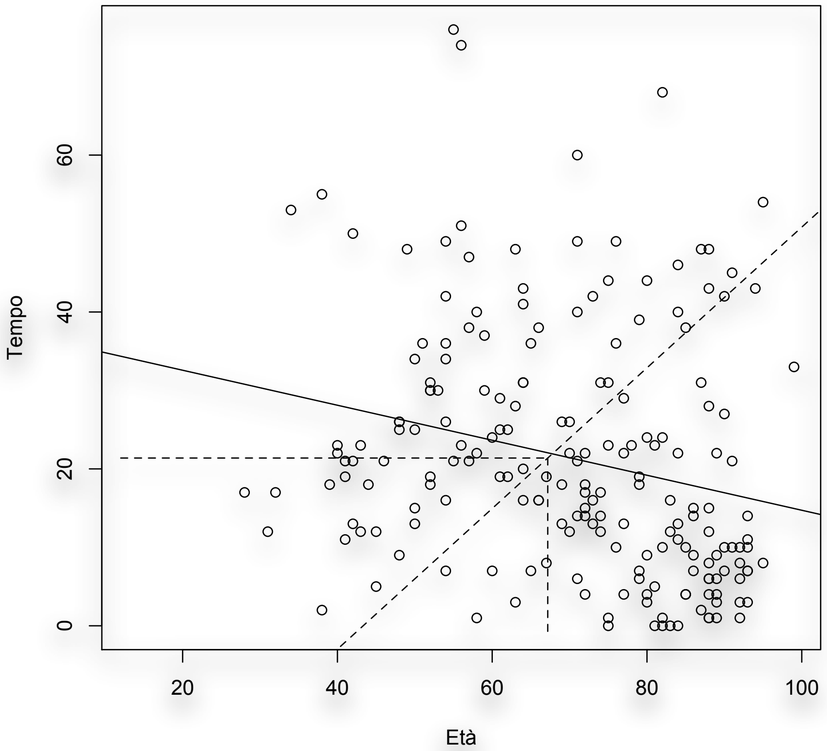
```
bivd <- function(x,y, xname, yname){
mx <- mean(x, na.rm = TRUE)
my <- mean(y, na.rm = TRUE)
sdx <- sd(x, na.rm = TRUE)
sdy <- sd(y, na.rm = TRUE)
plot(y ~ x, xlab = xname, ylab = yname)
segments (mx, my, mx, min(y, na.rm = TRUE) - 1, lty = "dashed")
segments (min(x, na.rm = TRUE) - 1, my, mx, my, lty = "dashed")
x1 <- mx - 3 * sdx
x2 <- mx + 3 * sdx
y1 <- my - 3 * sdy
y2 <- my + 3 * sdy
segments (x1, y1, x2, y2, lty ="dashed")
abline(lm(y ~ x))
}
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

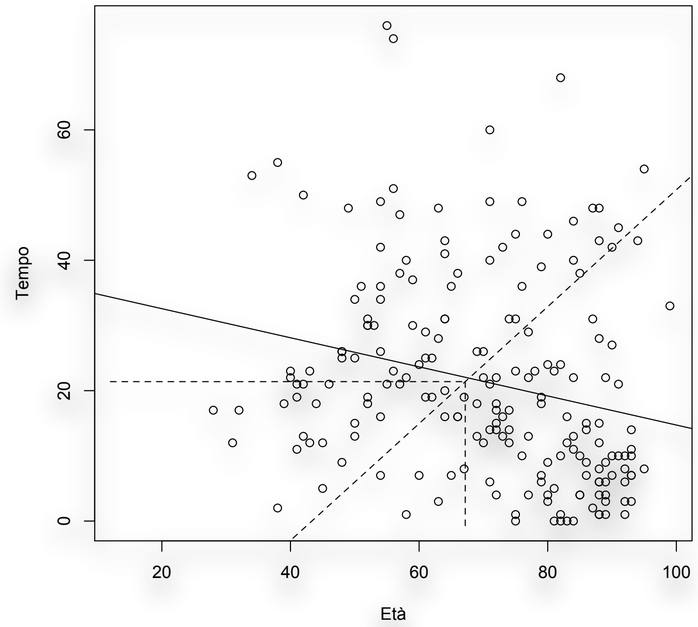
```
bivd <- function(x,y, xname, yname){
mx <- mean(x, na.rm = TRUE)
my <- mean(y, na.rm = TRUE)
sdx <- sd(x, na.rm = TRUE)
sdy <- sd(y, na.rm = TRUE)
plot(y ~ x, xlab = xname, ylab = yname)
segments (mx, my, mx, min(y, na.rm = TRUE) - 1, lty = "dashed")
segments (min(x, na.rm = TRUE) - 1, my, mx, my, lty = "dashed")
x1 <- mx - 3 * sdx
x2 <- mx + 3 * sdx
y1 <- my - 3 * sdy
y2 <- my + 3 * sdy
segments (x1, y1, x2, y2, lty ="dashed")
abline(lm(y ~ x))
}
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Si noti la differenza fra la retta delle deviazioni standard (tratteggiata) e la retta di regressione (continua).
- La prima fa da riferimento per valutare la **forza** dell'associazione lineare: tanto più i dati sono raggruppati attorno a questa retta, maggiore il coefficiente di correlazione.
- La seconda indica i valori previsti di *y* in base al modello lineare.
- Tanto più i dati sono raggruppati attorno a questa seconda retta, tanto minore è l'errore di previsione.

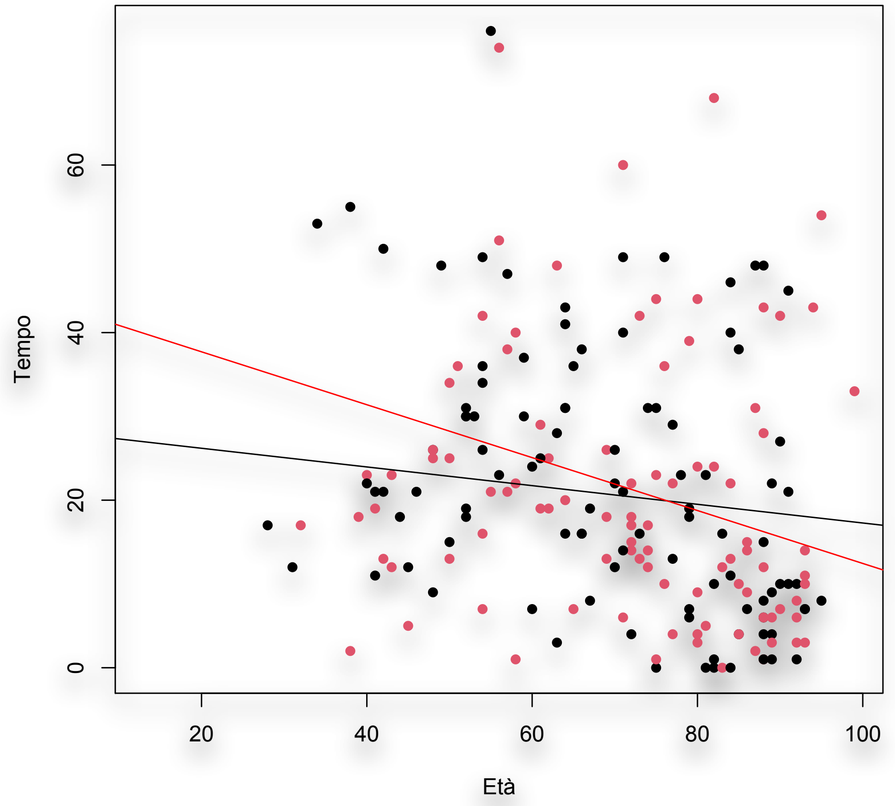


(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Per fittare due modelli lineari separati, dobbiamo suddividere il dataset in due parti (ad es. per *Sesso*) usando la funzione **subset**(). Fatto questo, disegniamo le rette di regressione per ogni sottoinsieme dei dati (nero=maschi, rosso=femmine):

```
> plot(dati$Tempo ~ dati$Età, xlab = "Età",
    ylab = "Tempo", pch =16, col = dati$Sesso)
> d1 <- subset(dati, dati$Sesso == "M")
> d2 <- subset(dati, dati$Sesso == "F")
> modello1 <- lm(d1$Tempo ~ d1$Età)
> abline(modello1, col = "black") # maschi
> modello2 <- lm(d2$Tempo ~ d2$Età)
> abline(modello2, col = "red") # femmine
> coef(modello1)
> coef(modello2)
```

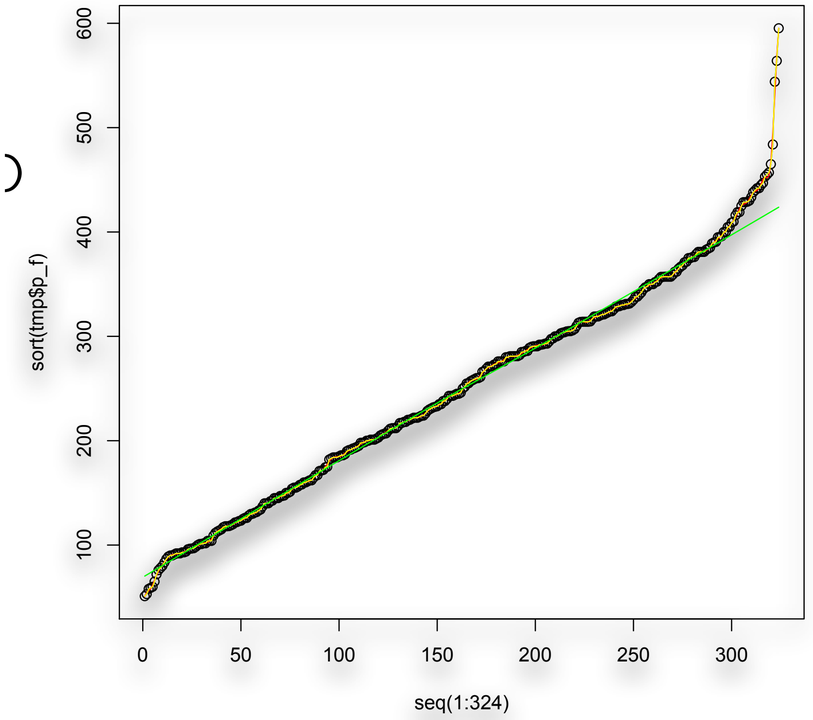


(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Regressione mediante metodi non parametrici (smoothers)

```
# scarto i valori mancanti
> tmp = subset(dati,!is.na(dati$p_f))
# utilizzo i dati p/f come curva crescente
> plot(seq(1:324), sort(tmp$p_f))
# curva di smoothing (in blu)
> lines(seq(1:324), runmed(sort(tmp$p_f), k = 3), col = "blue")
# curva spline (in rosso)
> lines(smooth.spline(sort(tmp$p_f)), col = "red")
# kernel smoothing (in giallo)
> lines(ksmooth(seq(1:324), sort(tmp$p_f)), col = "yellow")
# curva lowess (LOcally WEighted regresSSion, in verde)
> lines(lowess(seq(1:324), sort(tmp$p_f)), col = "green")
```

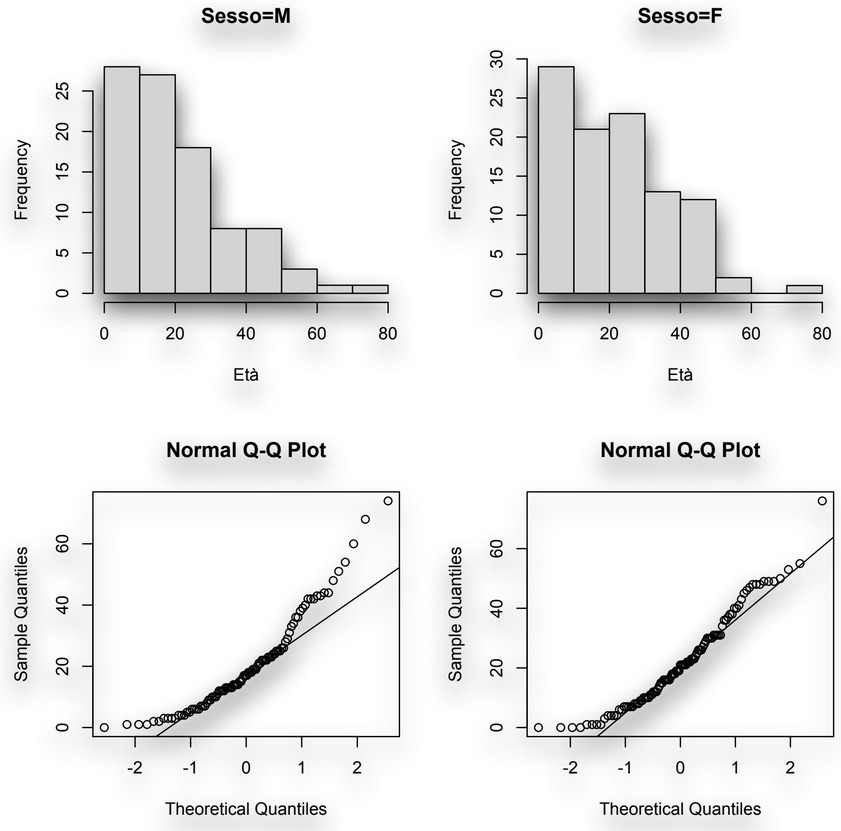


(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Relazione tra una variabile categorica e una numerica

```
> boxplot(dati$Tempo ~ dati$Sesso, xlab = "Sesso", ylab = "Tempo")
> par(mfrow = c(2,2)) # quattro grafici su una pagina
> hist(dati$Tempo[dati$Sesso == "M"], freq = TRUE,
  main = "Sesso=M", xlab ="Età")
> hist(dati$Tempo[dati$Sesso == "F"], freq = TRUE,
  main = "Sesso=F", xlab ="Età")
> qqnorm(dati$Tempo[dati$Sesso == "M"])
> qqline(dati$Tempo[dati$Sesso == "M"])
> qqnorm(dati$Tempo[dati$Sesso == "F"])
> qqline(dati$Tempo[dati$Sesso == "F"])
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Quanto è forte l'associazione? Calcoliamo…

```
> t1 <- dati$Tempo[dati$Sesso == "M"]; n1 <- length(t1)
> t2 <- dati$Tempo[dati$Sesso == "F"]; n2 <- length(t2)
```

- Differenza fra le medie:

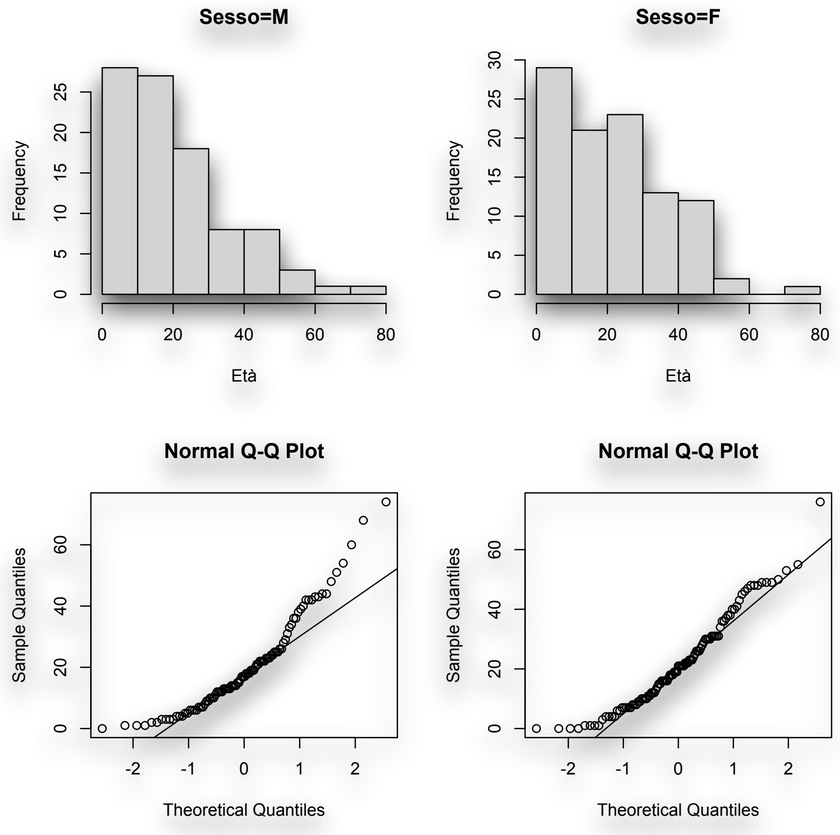
```
>mdif <- (mean(t2, na.rm=T) - mean(t1, na.rm=T))
```

- *d* di Cohen (effect size):

```
> d <- mdif/sqrt(((n1-1)*var(t1,na.rm=T) +
  (n2-1)*var(t2,na.rm=T))/(n1+n2-2))
```

- Pendenza della retta di regressione = differenza fra le medie

```
> dati$dc <- c(rep("M",n1),rep("F",n2))
> mod <- lm(dati$Tempo ~ dati$dc)
> cf <- coef(mod)
```



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Altro esempio di associazione tra una variabile categorica e una numerica.

```
> library(MASS)
```

Il dataset contiene i risultati di una ricerca su due trattamenti per l'anoressia:

$Treat = FT (terapia familiare), CBT (terapia cognitivo-comportamentale), Cont (Controllo); $prewt = peso pre terapia; $Postwt = peso post terapia.

```
> data(anorexia)
> str(anorexia)
'data.frame': 72 obs. of  3 variables:
 $ Treat : Factor w/ 3 levels "CBT","Cont","FT": 2 2 2 2 2 2 2 2 2 2 ...
 $ Prewt : num  80.7 89.4 91.8 74 78.1 88.3 87.3 75.1 80.6 78.4 ...
 $ Postwt: num  80.2 80.1 86.4 86.3 76.1 78.1 75.1 86.7 73.5 84.6 …
```

(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Come prima cosa esprimiamo i dati come variazione percentuale:

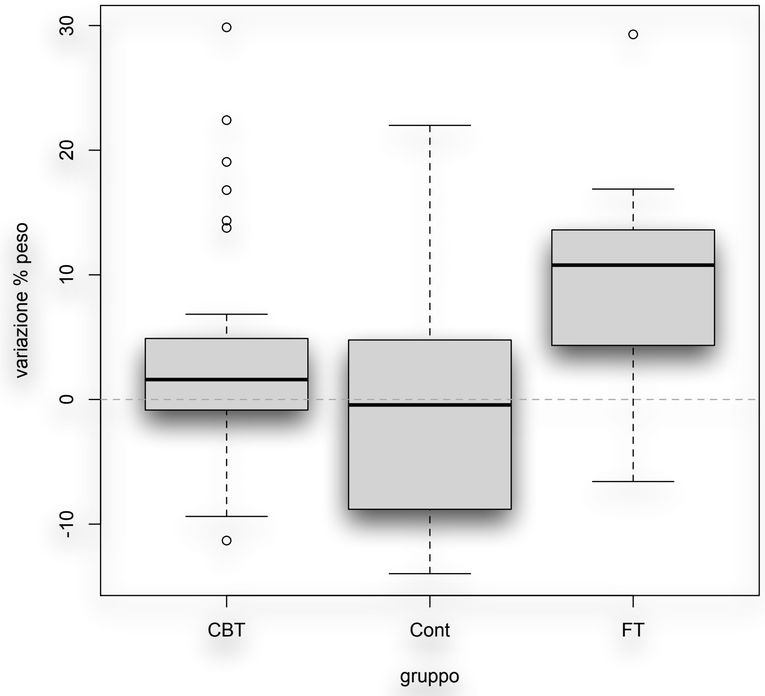
```
> anorexia$difw <- 100 * (anorexia$Postwt - anorexia$Prewt) / anorexia$Prewt
```

- Quindi visualizziamo l'associazione fra variazione percentuale e gruppo:

```
> boxplot(difw ~ Treat, data = anorexia,
     ylab = "variazione % peso", xlab = "gruppo")
> abline(h = 0, col = "dark grey", lty = "dashed")
```

La distribuzione della variazione percentuale nei tre gruppi appare ragionevolmente simmetrica e indica che la terapia familiare produce mediamente un aumento di peso attorno al 10-15%, mentre la terapia cognitivo comportamentale è sostanzialmente simile al gruppo di controllo, dove non si nota in media alcun aumento di peso.

L'incremento proporzionale può essere considerato una misura della grandezza dell'effetto (effect size), in questo caso più significativo delle differenze standardizzate, quindi l'associazione fra le due variabili è ben rappresentato dalle medie dei tre gruppi.



(Bio)Statistica con R — Parte I

Statistiche bivariate tra due variabili numeriche

- Calcoliamo le medie usando la funzione **tapply**(), che applica una funzione a un vettore in base ai livelli di un fattore. La sua sintassi è *tapply(x, index, fun)* dove *x* è un vettore, *index* è il fattore, è *fun* la funzione da applicare:

```
> mns <- tapply(anorexia$difw,
  anorexia$Treat, mean)
```

- L'oggetto **mns** contiene le medie, che confermano l'andamento che avevamo visto nel boxplot.
- Lo stesso approccio può essere utilizzato per calcolare le deviazioni standard nei tre gruppi, o qualsiasi altra statistica.