

Testing Psicologico

Lezione 4

Filippo Gambarota

@Università di Padova

Inferenza, domande?

Inferenza - Punti fondamentali

- Campione e Popolazione --> Statistiche e Parametri
- Approccio Fisher
 - H_0
 - Statistica test
 - Distribuzione campionaria
 - p value
- Test z
- Covarianza, Correlazione

Neyman e Pearson

Per maggiore intuitività dei concetti di *type-1/2 errors*, *potenza*, etc. c'è una web-app molto interessante <https://shiny.rit.albany.edu/stat/betaprob/>

Visualize Type I/II errors: One-sample Test of Means (Z test)

Choose Tail of the Test

☐ One Tail, Upper Tail
☐ One Tail, Lower Tail
☒ Two Tail

Choose Plot to Display

☒ Show Null Hypothesis Sampling Distribution
☒ Show Alternative Hypothesis Sampling Distribution

Choose alpha control via slider or menu

☒ Choose among several fixed alpha levels
☐ Use a slider for alpha choice

Alpha, the Type I Error rate

☐ .15 ☐ .10 ☒ .05 ☐ .025 ☐ .020 ☐ .01 ☐ .005 ☐ .001 ☐ .0005

Null Hypothesis Mean

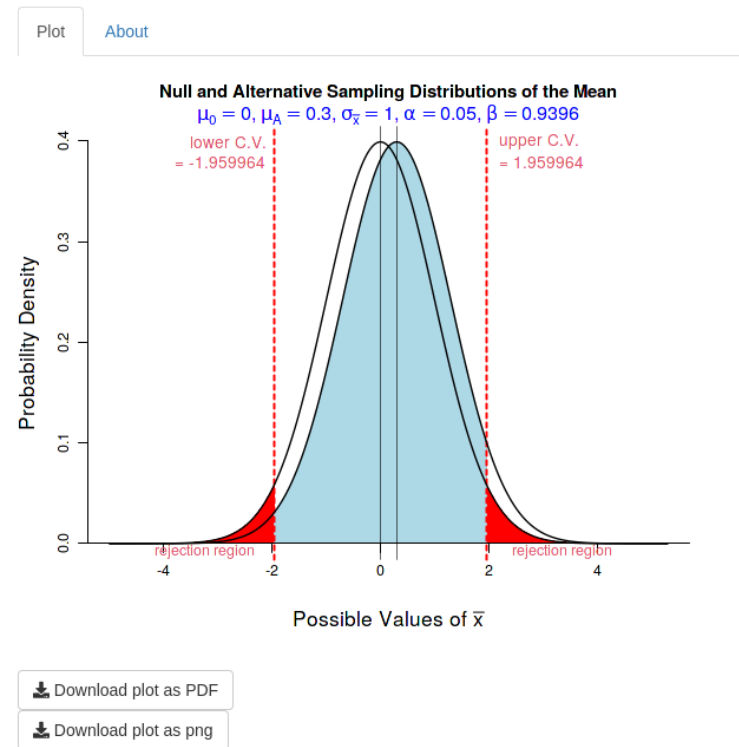
0

Alternative Hypothesis Mean

0.3

Standard Error of the Mean

1



Test Z Esercizi

Esercizio

Il manuale di un test per la **depressione** indica che la popolazione *normativa* ha valori $\mu = 30$ e $\sigma = 8$:

- Generare diversi campioni di $n = 30$ soggetti da una distribuzione normale con:
 - x_1 : $\mu = 30$ e $\sigma = 15$
 - x_2 : $\mu = 40$ e $\sigma = 10$
 - x_3 : $\mu = 30$ e $\sigma = 5$
- Calcolare *media* e *deviazione standard* dei campioni generati
- Rappresentare graficamente i 3 campioni
- Assumendo $\alpha = 0.05$ eseguire uno z test di ognuno dei campioni rispetto alla popolazione. Per il primo calcolate *a mano* lo z test, per gli altri usate la funzione `Ztest`
- Interpretare e descrivere i risultati

Soluzioni

```
# salviamo i valori normativi
mu <- 30
sigma <- 8
n <- 30

# generiamo i dati
x1 <- rnorm(n, 30, 15)
x2 <- rnorm(n, 40, 10)
x3 <- rnorm(n, 30, 5)

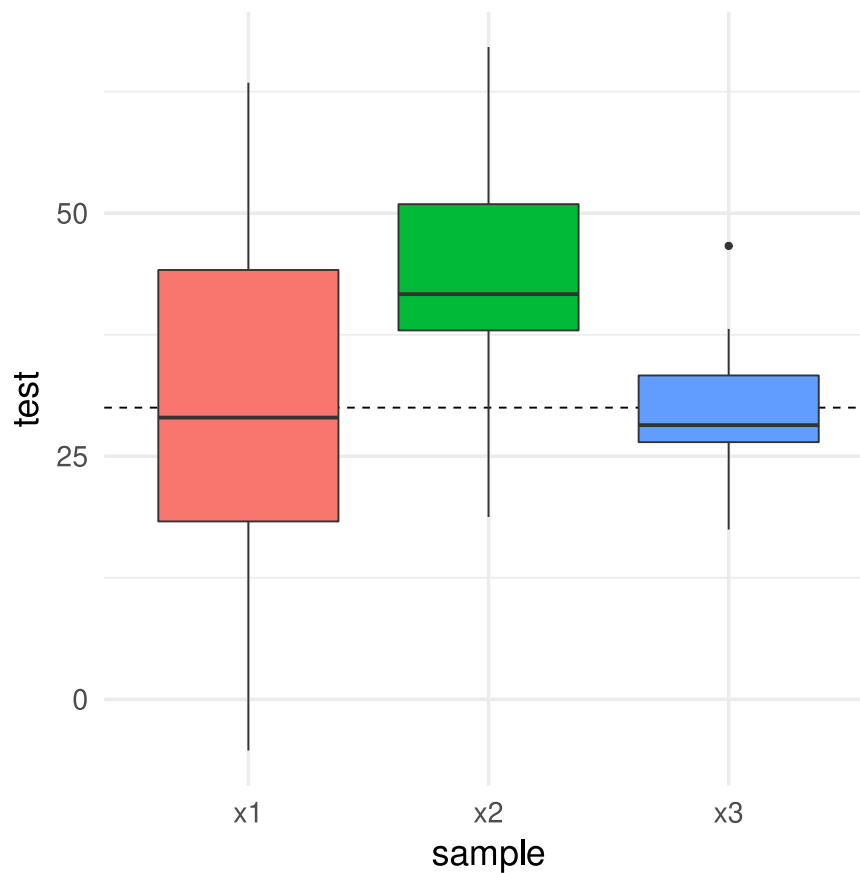
# calcoliamo media e deviazione standard
mean_sd <- function(x){
  c(media = mean(x), sd = sd(x))
}

lapply(list(x1 = x1, x2 = x2, x3 = x3), mean_sd)
```

```
## $x1
##      media      sd
## 29.85367 16.37143
##
## $x2
##      media      sd
## 43.74120 10.68618
##
## $x3
##      media      sd
## 29.366574  6.106123
```

Soluzioni

```
dat <- data.frame(test = c(x1, x2, x3), sample = rep(c("x1", "x2", "x3"), each = n))  
ggplot(dat, aes(x = sample, y = test, fill = sample)) +  
  geom_hline(yintercept = mu, linetype = "dashed") +  
  geom_boxplot(show.legend = FALSE) +  
  theme_minimal(base_size = 20)
```



Soluzioni

```
zoss1 <- (mean(x1) - mu)/(sigma/sqrt(length(x1))) # statistica test
pvalue1 <- pnorm(-abs(zoss1))*2 # p value
pvalue1 # conclusioni?
```

```
## [1] 0.9201963
```

```
(ztest2 <- Ztest(mean(x2), length(x2), mi = mu, sigma = sigma))
```

```
##          z.oss                p.value
## 1 9.407954 0.0000000000000000000000005058874
```

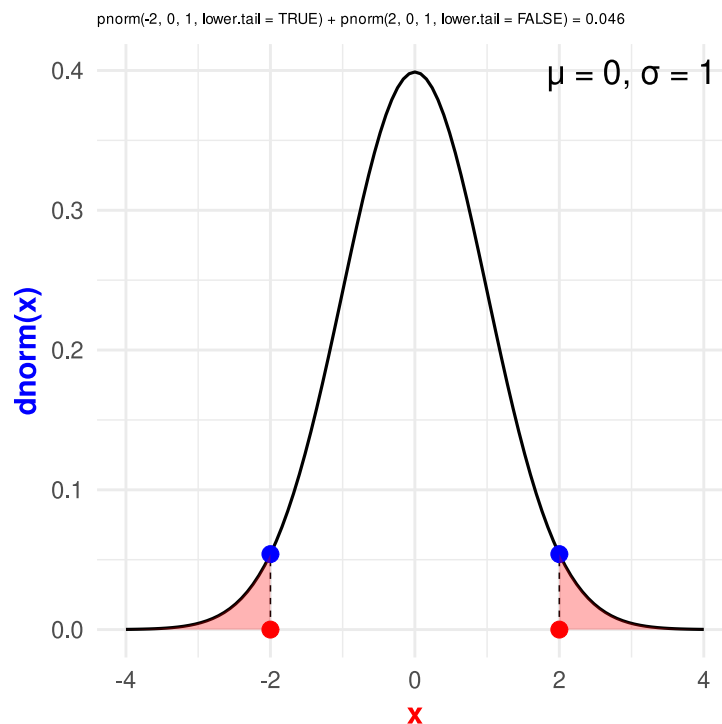
```
(ztest3 <- Ztest(mean(x3), length(x3), mi = mu, sigma = sigma))
```

```
##          z.oss    p.value
## 1 -0.436769 0.6645231
```

Rappresentare il p-value

Possiamo usare la funzione `ggnorm` anche per rappresentare il p-value quando facciamo il test z.

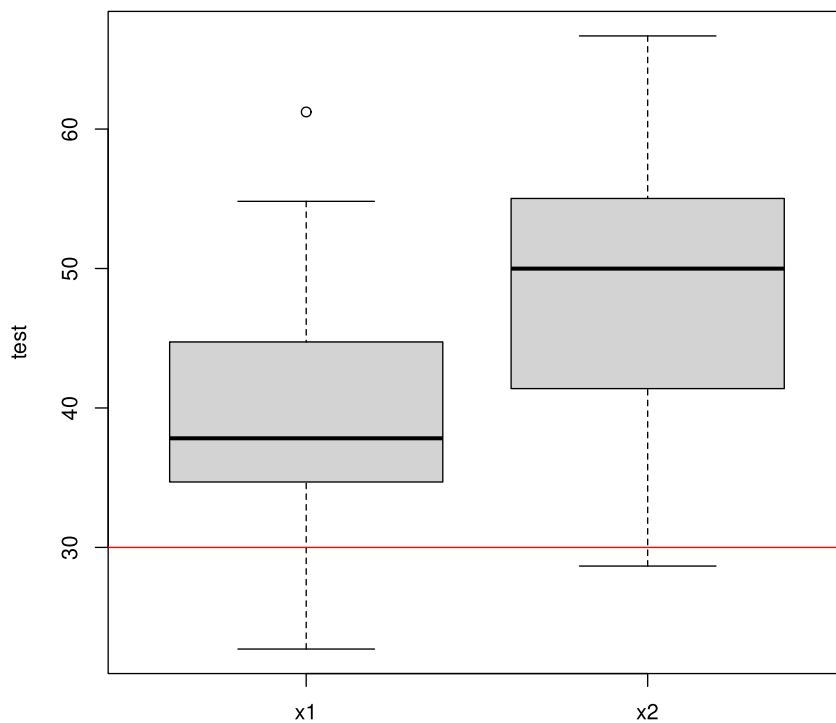
```
zoss <- 2  
ggnorm(c(-zoss, zoss), mean = 0, sd = 1, within = FALSE) +  
  theme(plot.title = element_text(size = 10))
```



Beyond the p value...

Il p value non è l'unica informazione importante quando si interpreta il risultato di un test. Prendiamo due condizioni:

```
x1 <- rnorm(30, mean = 40, sd = 10)
x2 <- rnorm(30, mean = 50, sd = 10)
dat <- data.frame(test = c(x1, x2), group = rep(c("x1", "x2"), each = 30))
```



Beyond the p value...

```
Ztest(mean(x1), length(x1), mi = mu, sigma = sigma)
```

```
##          z.oss          p.value
## 1 6.155452 0.0000000007486383
```

```
Ztest(mean(x2), length(x2), mi = mu, sigma = sigma)
```

[illegible]

In entrambi i casi il p value è significativo e molto piccolo. Ma la differenza è molto diversa in x_1 rispetto ad x_2 . E' sempre importante quindi guardare i dati e analizzare la grandezza dell'effetto (oltre al p value)

Non solo la media...

Provate a fare lo z test per questi due scenari:

```
set.seed(223)
mu <- 35
sigma <- 10

x1 <- rnorm(n = 50, mean = 40, sd = 10)
x2 <- rnorm(n = 500, mean = 40, sd = 10)

mean(x1)
```

```
## [1] 40.42362
```

```
mean(x2)
```

```
## [1] 40.17003
```

Cosa notate? le conclusioni sono le stesse?

Non solo la media...

```
Ztest(mean(x1), length(x1), mu, sigma)
```

```
##          z.oss          p.value
## 1 3.835081 0.0001255231
```

```
Ztest(mean(x2), length(x2), mu, sigma)
```

[illegible]

La formula dello z test dipende da μ_x , μ , σ e n_x . Cambiando alcuni parametri (e.g., n_x) e tenendo fissi altri (e.g., μ_x) abbiamo risultati diversi.

Covarianza e Correlazione

Covarianza e Correlazione

Anche qui una mitica web-app è molto utile per visualizzare la relazione tra due variabili cambiando vari parametri

<https://shiny.rit.albany.edu/stat/rectangles/>

Choose Plot Type:

☒ Base Scatterplot

☐ Show X and Y Means and Quadrants

Population Correlation (ρ)

IV (X) Mean (μ_X)

DV (Y) Mean (μ_Y)

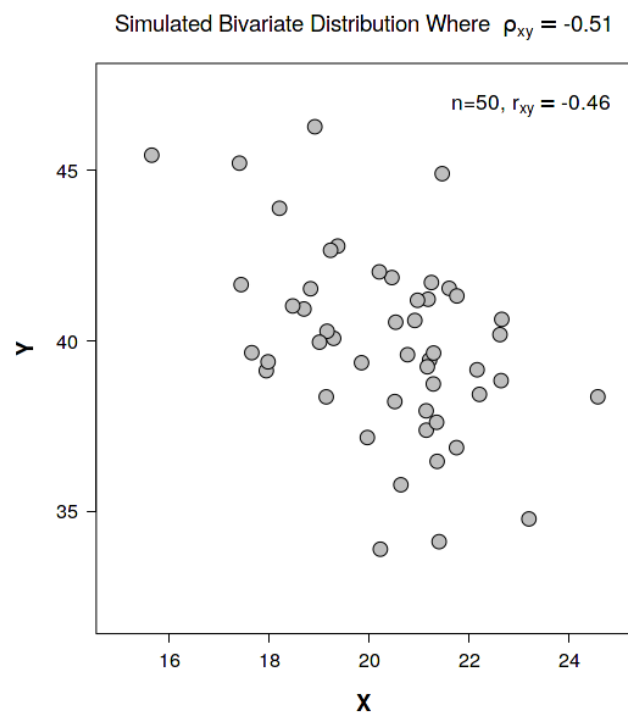
IV (X) Std. Dev. (σ_X)

DV (Y) Std. Dev. (σ_Y)

Sample Size

☐ Show best fit line, the regression line

Resample with same parameters



rcorr e plotcorr

Con la funzione `rcorr` e `plotcorr` potete generare due vettori numerici con una data correlazione e rappresentarli graficamente. Il principio è lo stesso di `rnorm` ma potete simulare una relazione *bivariata*:

```
cor0.7 <- rcorr(mux = 0, muy = 0, n = 100, sdX = 1, sdY = 1, r = 0.7)
```

```
# analisi univariate  
summary(cor0.7$x)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## -2.15166 -0.59910 -0.14704 -0.05157  0.62173  1.95609
```

```
summary(cor0.7$y)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## -2.49701 -0.82291 -0.05626 -0.09222  0.59159  2.42595
```

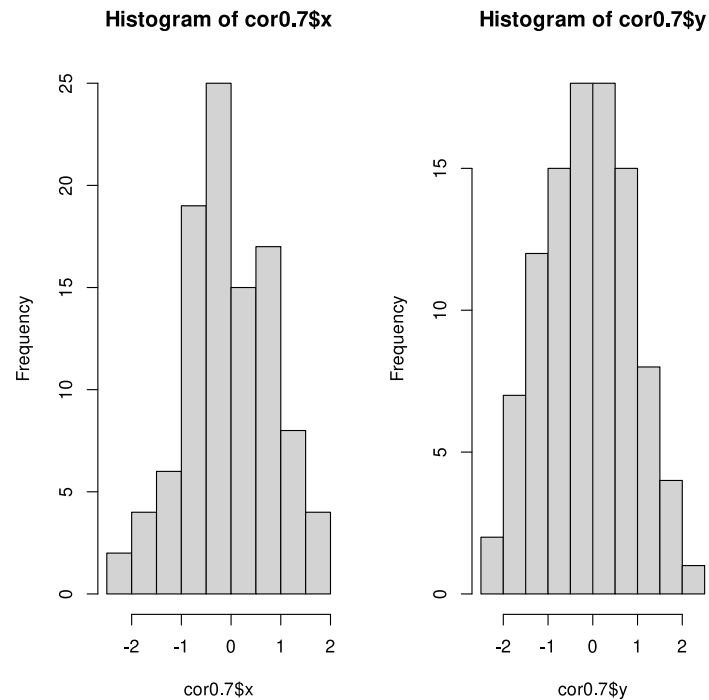
```
# relazione bivariata  
cor(cor0.7$x, cor0.7$y)
```

```
## [1] 0.7722964
```

rcorr e plotcorr

Facciamo i grafici univariati

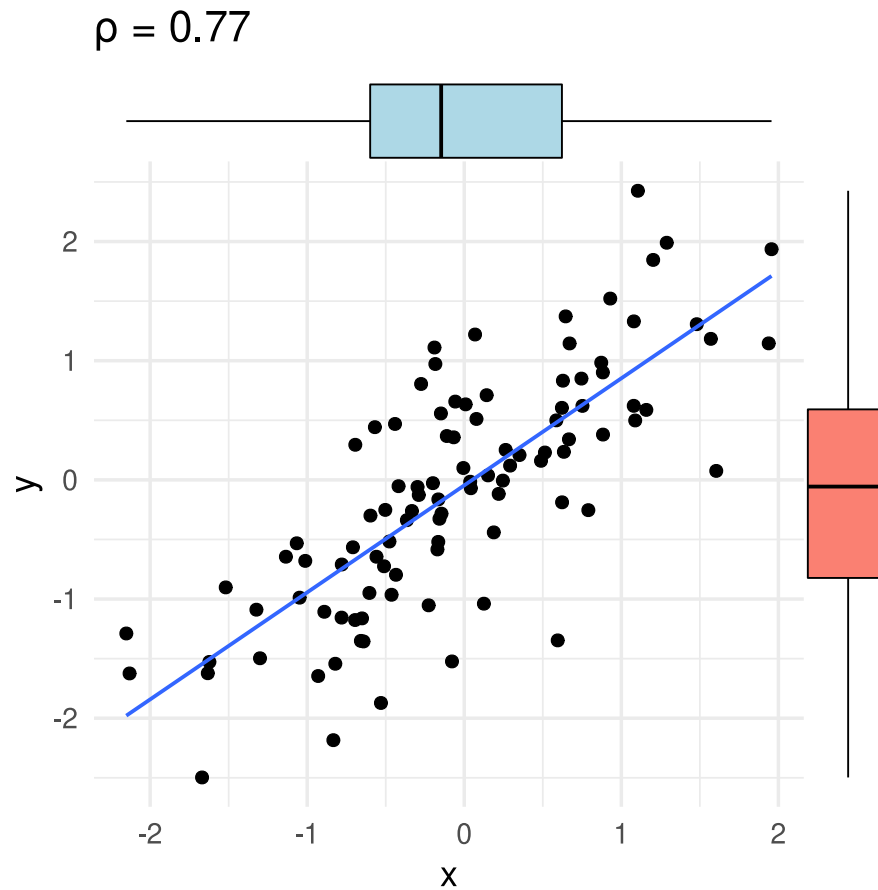
```
par(mfrow = c(1, 2))  
hist(cor0.7$x)  
hist(cor0.7$y)
```



rcorr e plotcorr

Questa è la relazione bivariata

```
plotcorr(cor0.7, marginal = TRUE)
```



Correlazione Esercizi

Correlazione Esercizi

- Generare un dataset (chiamato `dat`) usando la funzione `rcorr` con:
 - $\mu_1 = 10, \mu_2 = 50$
 - $\sigma_1 = 4, \sigma_2 = 10$
 - $\rho = 0.4$
 - $n = 50$
- Calcolare le statistiche descrittive e rappresentare graficamente le variabili in modo univariato
- Rappresentare le variabili in modo bivariato
- Usare le formule per calcolare manualmente la covarianza e la correlazione. Confrontare il risultato con le funzioni `cov` e `cor`
- Applicare il `cor.test` e interpretare il risultato

Formule

Covarianza

$$cov_{x,y} = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{N - 1}$$

Correlazione

$$\rho_{x,y} = \frac{cov_{x,y}}{\sigma_x \sigma_y}$$

Soluzioni

```
dat <- rcorr(mux = 10, muy = 50, sdx = 4, sdy = 10, n = 50, r = 0.4)
rsummary(dat$x)
```

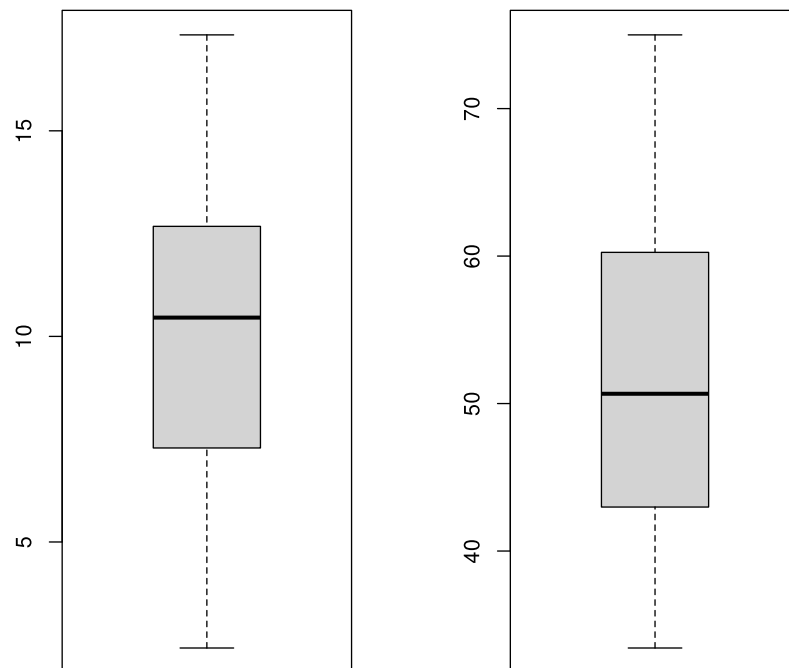
##	min	25%	mean	median	sd	cv	75%
##	2.4214745	7.5720134	10.2182551	10.4572088	3.7524900	0.3672339	12.6114264
##	max						
##	17.3334076						

```
rsummary(dat$y)
```

##	min	25%	mean	median	sd	cv	75%
##	33.4252168	43.0256801	51.8407869	50.6636472	10.3800926	0.2002302	60.1238811
##	max						
##	74.9987380						

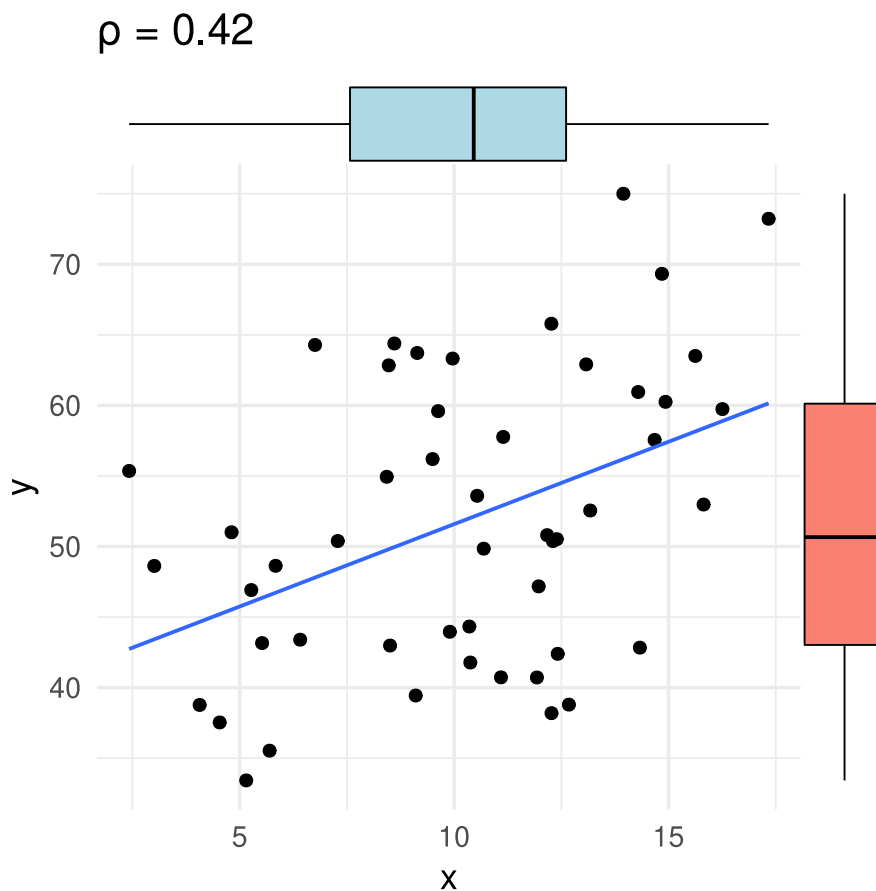
Soluzioni

```
par(mfrow = c(1, 2))  
boxplot(dat$x)  
boxplot(dat$y)
```



Soluzioni

```
plotcorr(dat, marginal = TRUE)
```



Soluzioni

```
# covarianza a mano
cov_xy <- with(dat, sum((x - mean(x)) * (y - mean(y)))/(length(x) - 1))
cov_xy
```

```
## [1] 16.43805
```

```
cov(dat$x, dat$y)
```

```
## [1] 16.43805
```

```
# correlazione
cov_xy / (sd(dat$x) * sd(dat$y))
```

```
## [1] 0.4220167
```

```
cor(dat$x, dat$y)
```

```
## [1] 0.4220167
```

Soluzioni

```
# test  
cor.test(dat$x, dat$y)
```

```
##  
##      Pearson's product-moment correlation  
##  
## data:  dat$x and dat$y  
## t = 3.2251, df = 48, p-value = 0.002269  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
##  0.1627917 0.6267426  
## sample estimates:  
##          cor  
## 0.4220167
```

Che conclusioni possiamo trarre?

Matrice di correlazione

Quando abbiamo più variabili possiamo comunque calcolare la correlazione bivariata tra tutte le coppie di variabili:

```
dat <- readRDS("cordat.rds")  
head(dat$datw)
```

##		A	B	C
## 1	-1.37007868	-0.88731141	0.01047664	
## 2	1.34586911	1.01121441	2.04993212	
## 3	-2.30289552	-0.01314173	1.42270331	
## 4	-0.96066337	-0.66562039	1.95803229	
## 5	0.02332091	0.36085584	-0.23760797	
## 6	1.05984479	0.10978431	1.13820181	

Matrice di correlazione

Possiamo usare il comando `cor` ed anche il comando `plot` per calcolare e rappresentare le correlazioni

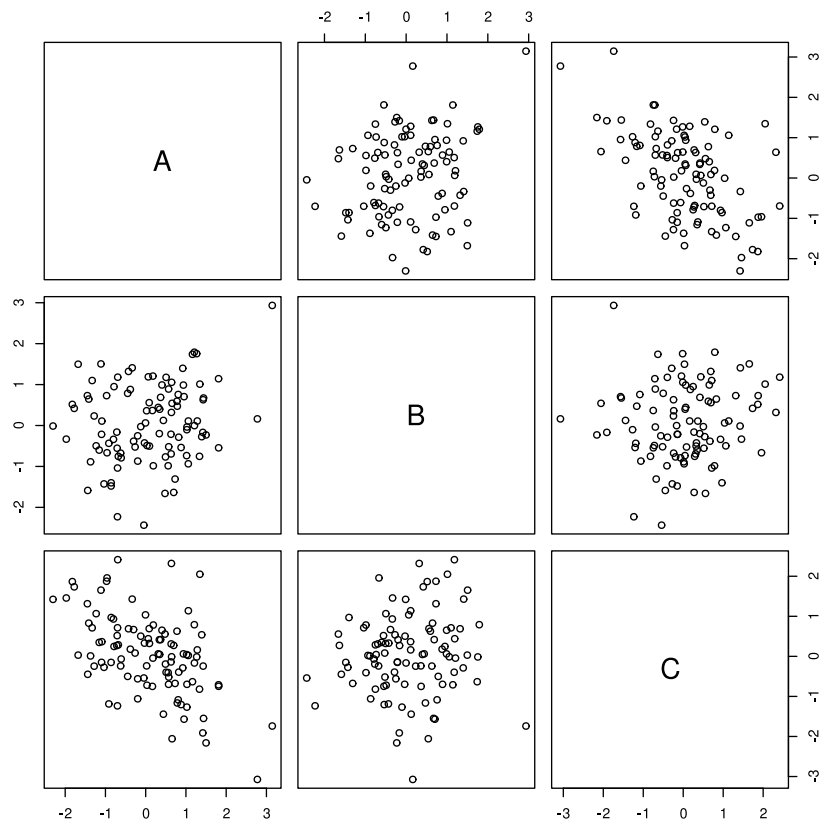
```
round(cor(dat$datw), 3) # matrice di correlazione
```

```
##           A          B          C
## A  1.000  0.209 -0.523
## B  0.209  1.000  0.094
## C -0.523  0.094  1.000
```

Matrice di correlazione

Possiamo usare il comando `cor` ed anche il comando `pairs` per calcolare e rappresentare le correlazioni

```
pairs(dat$datw)
```



... `corrplot`

```
corrplot::corrplot(cor(dat$datw))
```

