Testing Psicologico

Lezione 4

Filippo Gambarota

@Università di Padova

21/11/2022

Inferenza, domande?

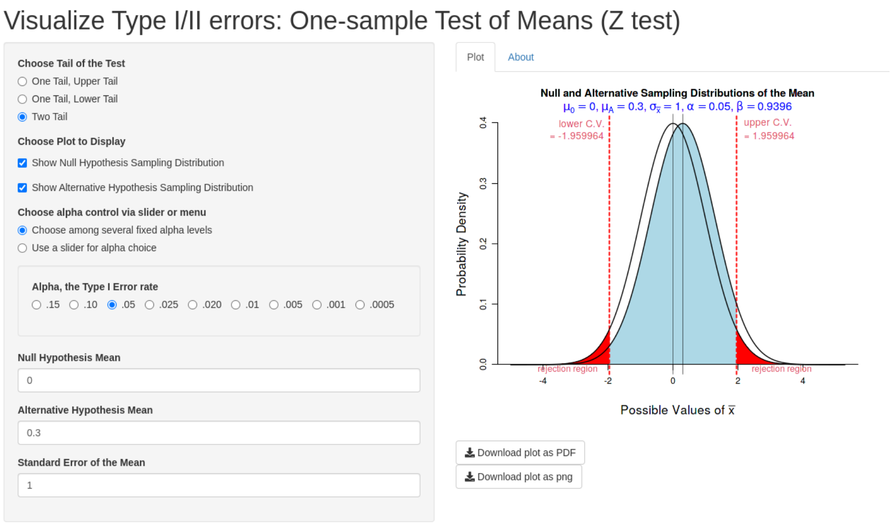
Inferenza - Punti fondamentali

- 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/

```r
knitr::include_graphics("img/shiny-neyman-pearson.png")
```



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:
    - `x1`: $\mu = 30$ e $\sigma = 8$
    - `x2`: $\mu = 40$ e $\sigma = 10$
    - `x3`: $\mu = 30$ e $\sigma = 2$
- 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`
- Intepretare e descrivere i risultati

Soluzioni

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

# generiamo i dati
x1 <- rnorm(n, 30, 8)
x2 <- rnorm(n, 40, 10)
x3 <- rnorm(n, 30, 2)

# 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
## 31.605190  7.224413
##
## $x2
##    media       sd
## 37.63800  8.25706
##
## $x3
##     media        sd
## 29.779007  1.744081
```

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

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

```
## [1] 0.2717686
```

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

```
##     z.oss p.value
## 1 5.22938       0
```

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

```
##        z.oss p.value
## 1 -0.1513036  0.8797
```

Beyond the p value...

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

```
x1 <- rnorm(50, mean = 35, sd = 10)
x2 <- rnorm(50, mean = 45, sd = 10)
dat <- data.frame(test = c(x1, x2), group = rep(c("x1", "x2"), each = 50))
```

Beyond the p value...

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

```
##      z.oss p.value
## 1 5.037906       0
```

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

```
##      z.oss p.value
## 1 12.37547       0
```

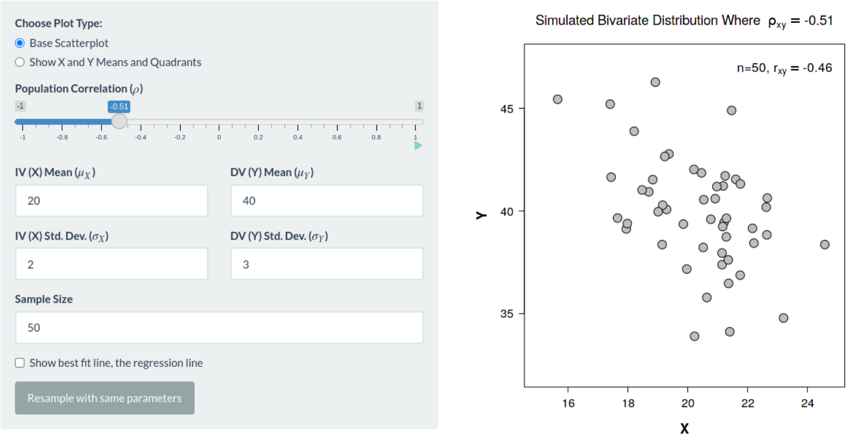
In entrambi i casi il p value è significativo e molto piccolo. Ma la differenza è molto diversa in `x1` rispetto ad `x2`. E' sempre importante quindi guardare i dati e analizzare la grandezza dell'effetto (oltre al p value)

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/

```r
knitr::include_graphics("img/shiny-correlation.png")
```



`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.
## -2.873314 -0.528048  0.007196 -0.033845  0.491395  2.708188
```

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

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.34091 -0.42742 -0.03502  0.06285  0.64521  2.60697
```

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

```
## [1] 0.676788
```

`rcorr` e `plotcorr`

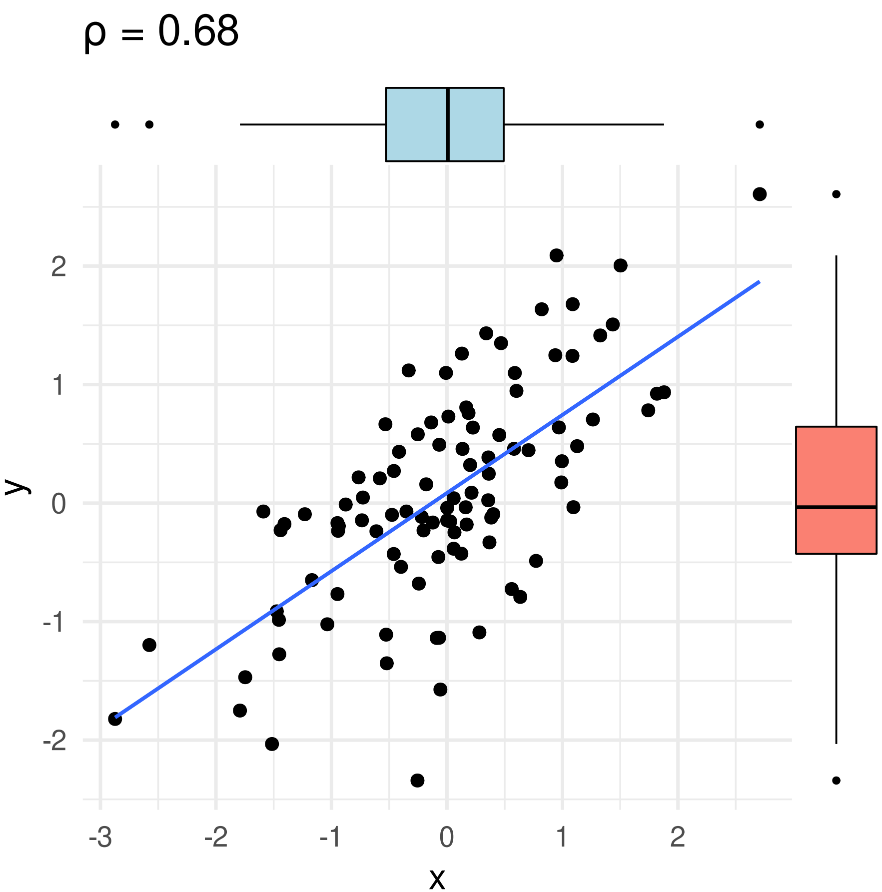
Facciamo i grafici univariati

```
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 intepretare il risultato

Formule

$$ cov{x,y} = \frac{\sum{i=1}^{N}(x{i}-\bar{x})(y{i}-\bar{y})}{N-1}

\rho{x,y} = \frac{cov{x,y}}{\sigma_x\sigma_y} $$

Soluzioni

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

rsummary(dat$x)
```

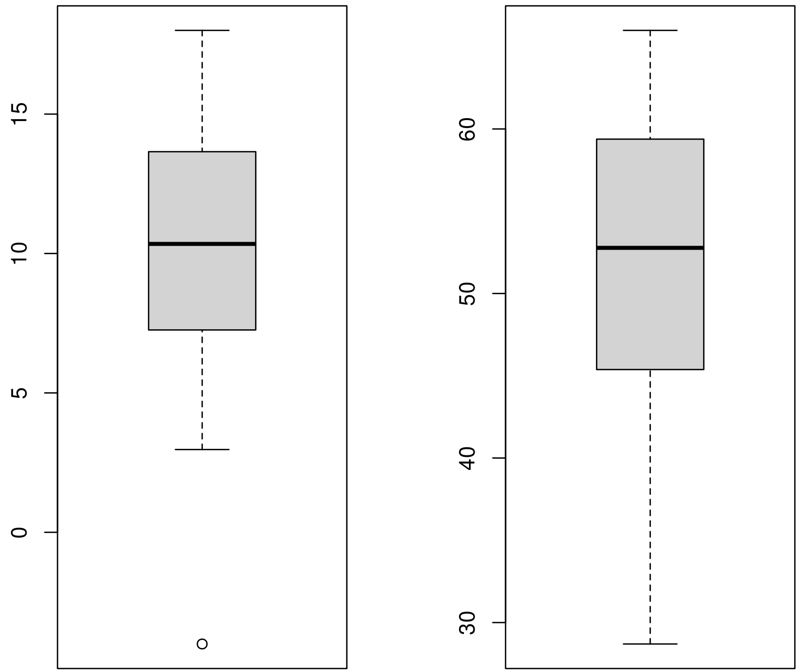
```
##        min        25%       mean     median         sd         cv        75%
## -4.0048629  7.5457800 10.3174020 10.3439231  4.4700952  0.4332578 13.5836867
##        max
## 18.0014689
```

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

```
##       min       25%      mean    median        sd        cv       75%       max
## 28.697406 45.447746 51.854195 52.774339  9.361600  0.180537 59.214827 65.991245
```

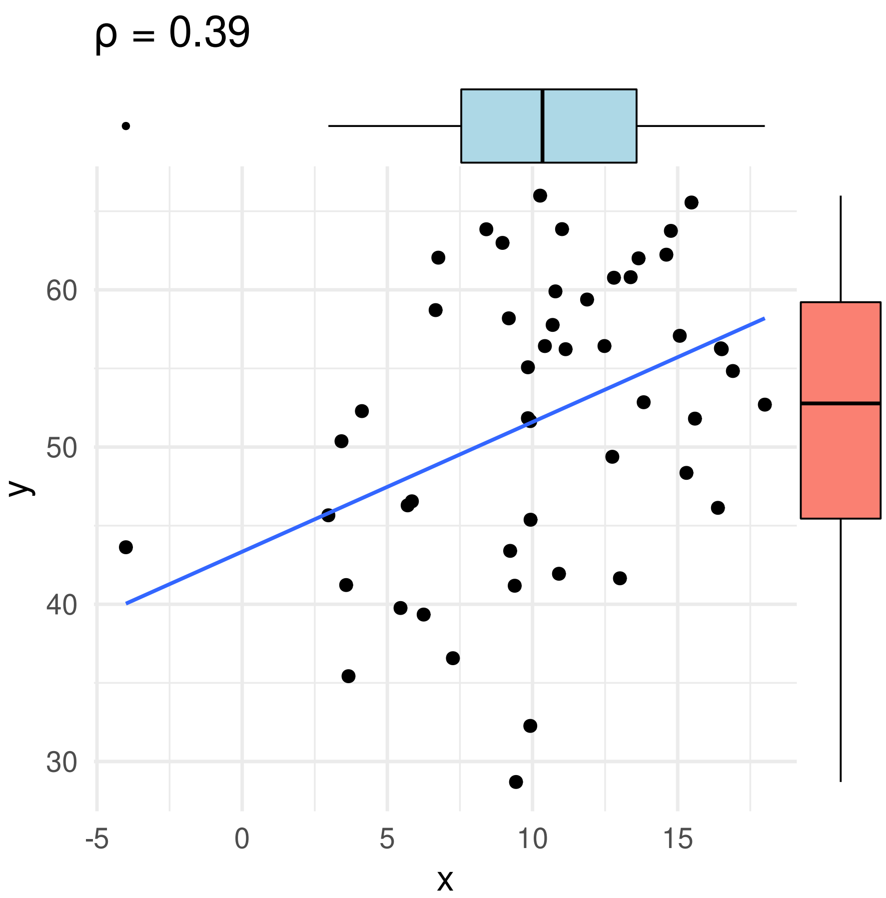
Soluzioni

```
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] 16.48232
```

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

```
## [1] 16.48232
```

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

```
## [1] 0.3938686
```

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

```
## [1] 0.3938686
```

Soluzioni

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

```
##
##     Pearson's product-moment correlation
##
## data:  dat$x and dat$y
## t = 2.9688, df = 48, p-value = 0.004655
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1297452 0.6058009
## sample estimates:
##       cor
## 0.3938686
```