Codice statistica Recap

2°CLEAM

Written by
Alice Oliveri del Castillo

```
library(UBStats) # Per aprire il pacchetto UBStats

# Riordinare i livelli di una variabile qualitativa ordinale

dataframe$x_recode <- factor(dataframe$x, levels=c())
```

Statistica univariata

```
distr.table.x(dataframe$variabile, freq) # Distribuzione di frequenza

distr.table.x(dataframe$variabile, freq='count') # Frequenza assoluta
distr.table.x(dataframe$variabile, freq='prop') # Frequenza relativa
distr.table.x(dataframe$variabile, freq='perc') # Frequenza percentuale
distr.table.x(dataframe$variabile, freq=c('prop','cum')) # Relativa cumulativa
distr.table.x(dataframe$variabile, freq='dens') # Densità di frequenza

distr.table.x(dataframe$variabil, breaks) # Distr. di frequenza di dati
                                            raggruppati in classi di
intervallo

distr.table.x(dataframe$variabile, breaks=n) # n intervalli di uguale ampiezza
distr.table.x(dataframe$variabile, breaks=c(a, b,..., z)) # intervalli con
                                                            estremi
a,b,..., z

distr.plot.x(dataframe$variabile, plot.type) # Grafico

distr.plot.x(dataframe$variabile, plot.type='pie') # Diagramma a torta
distr.plot.x(dataframe$variabile, plot.type='bars') # Diagramma a barre
distr.plot.x(dataframe$variabile, plot.type='spike') # Diagramma ad aste
distr.plot.x(dataframe$variabile, plot.type='cum') # Diagramma a scalini
distr.plot.x(dataframe$variabile, plot.type='hist', breaks) # Istogramma
distr.plot.x(dataframe$variabile, plot.type='cum', breaks) # Ogiva
distr.plot.x(dataframe$variabile, plot.type='boxplot') # Boxplot

distr.summary.x(dataframe$x, stats) # Misure di tendenza e di dispersione

distr.summary.x(dataframe$x, stats='mode') # Moda
distr.summary.x(dataframe$x, stats='median') # Mediana
distr.summary.x(dataframe$x, stats='mean') # Media

distr.summary.x(dataframe$x, stats='central') # Misure tendenza centrale

distr.summary.x(dataframe$x, stats='quartiles') # Quartili
distr.summary.x(dataframe$x, stats='deciles') # Decili
distr.summary.x(dataframe$x, stats='pn') # n-esimo percentile
distr.summary.x(dataframe$x, stats='q1') # Primo quartile
distr.summary.x(dataframe$x, stats='q3') # Terzo quartile

distr.summary.x(dataframe$x, stats='fivenumbers') # 5nr di sintesi

distr.summary.x(dataframe$x, stats='range') # Range
distr.summary.x(dataframe$x, stats='IQrange') # IQR
distr.summary.x(dataframe$x, stats='var') # Varianza
distr.summary.x(dataframe$x, stats='sd') # Deviazione standard
```

```
distr.summary.x(dataframe$x, stats='cv') # Coefficiente di variazione

distr.summary.x(dataframe$x, stats='dispersion') # Misure di dispersione

distr.summary.x(dataframe$x, stats='summary') # 5nr di sintesi, var e sd
```

Statistica bivariata

```
# Distribuzioni di frequenze congiunte

distr.table.xy(dataframe$x, dataframe$y, freq='count', freq.type='joint')

# Distribuzioni di frequenze condizionate

distr.table.xy(dataframe$x, dataframe$y, freq, freq.type='y|x')
distr.table.xy(dataframe$x, dataframe$y, freq, freq.type='x|y')

distr.plot.xy(dataframe$x, dataframe$y, plot.type) # Grafico

distr.plot.xy(dataframe$x, dataframe$y, plot.type='bars') # A barre
distr.plot.xy(dataframe$x, dataframe$y, plot.type='boxplot') # Boxplot
distr.plot.xy(dataframe$x, dataframe$y,plot.type='scatter') # Scatterplot

distr.summary.x(dataframe$x, by1=dataframe$y, stats) # Statistiche

distr.summary.x(dataframe$x, by1=dataframe$y, stats=c('summary','cv')

cov(dataframe$x, dataframe$y, use='complete') # Covarianza
cor(dataframe$x, dataframe$y, use='complete') # Coef. di rel. lineare
```

Altro

```
mean(condizione) # Per trovare la % di casi che soddisfano la condizione
sum(condizione) # Per trovare nr di casi che soddisfano la condizione
sum(dataframe$x > q3+1.5*(q3-q1)) # Per trovare nr di outliers superiori
sum(dataframe$x < q1-1.5*(q3-q1)) # Per trovare nr di outliers inferiori
```

pnorm(q, μ=0, σ=1)= $F(q) = Prob(X \leq q)$

qnorm(prob_a_sx, μ=0, σ=1)= x_{1-p} : $F(x_{1-p}) = Prob(X \leq x_{1-p}) = p$

Intervalli di confidenza

qnorm($1-\frac{\alpha}{2}$, 0, 1) # Calcolo di $z_{\frac{\alpha}{2}}$

qt($1-\frac{\alpha}{2}$, n-1) # Calcolo di $t_{n-1,\frac{\alpha}{2}}$

qt($1-\frac{\alpha}{2}$, nx+ny-2) # Calcolo di $t_{nx+ny-2,\frac{\alpha}{2}}$

Media μ

CI.mean(dataframe$x, sigma=$\sigma$, conf.level=$1-\alpha$) # c.i. per μ con σ^2 nota

CI.mean(dataframe$x, conf.level=$1-\alpha$) # c.i. per μ con σ^2 non nota

Proporzione p

CI.prop(dataframe$x, success, conf.level=$1-\alpha$)# c.i. per p

Differenza tra medie $\mu_X - \mu_Y$

CI.diffmean(type='paired')# Campioni appaiati

CI.diffmean(x, y, type, sigma.d, conf.level)

CI.diffmean(type='independent')# Campioni indipendenti

```
# Approccio 1

      # Estrazione dati di interesse


        x <- dataframe$x[dataframe$a=='YES']
        y <- dataframe$y[dataframe$a=='NO]
        CI.diffmean(x, y, type, sigma.x, sigma.y, conf.level)

# Approccio 2

      CI.diffmean(dataframe$x, by, type, sigma.by, conf.level)
```

Differenza tra proporzioni $p_X - p_Y$

```
# Approccio 1

      # Estrazione dati di interesse


        x <- dataframe$x[dataframe$a=='YES']
```

```
        y <- dataframe$y[dataframe$a==‘NO]
        CI.diffprop(x, y, success.x, success.y, conf.level)

# Approccio 2


        CI.diffprop(x, by, success.x, conf.level)
```

Test Statistici

Media μ

```
# Con σ² nota


        TEST.mean(x, sigma, mu0, alternative='greater')# Test μ coda dx
        TEST.mean(x, sigma, mu0, alternative='less')# Test μ coda sx
        TEST.mean(x, sigma, mu0, alternative='two.sided')# Test μ 2 code

# Con σ² NON nota


        TEST.mean(x, mu0, alternative)# Test μ
```

Proporzione p

```
TEST.prop(x, success, p0, alternative)# Test p coda dx
```

Differenza tra medie $\mu_X - \mu_Y$

```
TEST.diffmean(type='paired')# Campioni appaiati

TEST.diffmean(x, y, type, mdiff, sigma.d, alternative)

TEST.diffmean(type='independent')# Campioni indipendenti

# Approccio 1

      # Estrazione dati di interesse


        x <- dataframe$x[dataframe$a=='YES']
        y <- dataframe$y[dataframe$a=='NO]
        TEST.diffmean(x, y, type, mdiff, alternative)

# Approccio 2

      TEST.diffmean(dataframe$x, by, type, mdiff, alternative)

TEST.diffmean(var.test=T) # Levene's test for homogeneity of variance
```

Differenza tra proporzioni $p_X - p_Y$

```
# Approccio 1

      # Estrazione dati di interesse
```

```
        x <- dataframe$x[dataframe$a=='YES']
        y <- dataframe$y[dataframe$a=='NO]
        TEST.diffprop(x, y, success.x, success.y, pdiff0, alternative)

# Approccio 2


        TEST.diffprop(x, by, success.x, pdiff0, alternative)




1-pnorm(z) # Calcolo di p-value per test a 1 coda (Normale standard)
2*(1-pnorm(z)) # Calcolo di p-value per test a 2 coda (Normale standard)

1-pt(t, n-1) # Calcolo di p-value per test a 1 coda (t di student)
2*(1-pt(t, n-1)) # Calcolo di p-value per test a 2 coda (t di student)

qnorm(1 − α, 0, 1) # Calcolo di z_α → pnorm(z_α) # Calcolo di 1 − α

qnorm(1 − α/2, 0, 1) # Calcolo di z_{α/2}

qt(1 − α, n-1) # Calcolo di t_{n−1,α}

qt(1 − α/2, n-1) # Calcolo di t_{n−1,α/2}

pnorm(soglia, μ, sqrt(σ²/n)) # Calcolo di β

# TEST CHI-QUADRATO DI ADATTAMENTO

chisq.test(x=c(O1,…,Ok), p=c(p1,…,pk))

1-pchisq(χ̂², k-1) # p-value
qchisq(1 − α, k-1) # χ²_{(k−1),α}

# TEST CHI-QUADRATO DI INDIPENDENZA

chisq.test(x, y)

1-pchisq(χ̂², (r-1)(c-1)) # p-value
qchisq(1 − α, (r-1)(c-1)) # χ²_{(r−1)(c−1),α}
```

Regressione lineare semplice

```
lm1 <- lm(y~x, data=dataframe) # Creazione di un modello su R
summary(lm1)
```

Test di significatività

```
summary(lm1)
```

Oppure

```
2*(1-pt(t, n-2)) # Calcolo di p-value per test a 2 coda (t di student)
```

qt($1-\frac{\alpha}{2}$, n-2) # Calcolo di $t_{n-2,\frac{\alpha}{2}}$

Intervalli di conf/prev

confint(lm1, level=$1-\alpha$) # $c.i._{1-\alpha}$ per β_1 e β_0

```
# Previsione su un valore esatto/singolo valore (Int. di previsione)

z<-data.frame(x=xg)
predict(lm1, z, interval='prediction', level)

# Previsione sulla media stimata (Int. di confidenza)

z<-data.frame(x=xg)
predict(lm1, z, interval='confidence', level)
```

Analisi dei residui

```
distr.plot.xy(x, y, plot.type='scatter', fitline=T) # Scatterplot

plot(lm1, 1) # Residual vs fitted plot

plot(lm1, 2) # Q-q plot

distr.plot.x(rstandard(lm1), plot.type='hist')# Hist residui standar
```

Regressione lineare multipla

```
lm1 <- lm(y~x1+x2+x3, data=dataframe) # Creazione di un modello su R
summary(lm1)
```

Test di significatività globale

```
summary(lm1)
```

Oppure

```
1-pf(f, k, n-k-1) # Calcolo di p-value per test a 2 coda (f)
```

```
qf(1-α, k, n-k-1) # Calcolo di f_{k,n-k-1,1-α}
```

Intervalli di conf/prev

```
confint(lm1, level=1-α) # c.i._{1-α} per β1 e β0
```

```
# Previsione su un valore esatto/singolo valore (Int. di previsione)

z<-data.frame(x1=x1g, x2=x2g, x3=x3g)
predict(lm1, z, interval='prediction', level)

# Previsione sulla media stimata (Int. di confidenza)

z<-data.frame(x1=x1g, x2=x2g, x3=x3g)
predict(lm1, z, interval='confidence', level)

cor(x1, x2) # Per calcolare la correlazione tra le var.
```


ASTRA
BOCCONI