UNIVERSITÀ DEGLI STUDI DI FIRENZE
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Magistrale in Informatica

Elaborato d'Esame

MODELLI STATISTICI

MASSIMO NOCENTINI

Professore: *Giovanni Maria Marchetti*

Anno Accademico 2012-2013

INDICE

INTRODUZIONE

Queste note contengono tutto il mio materiale di studio per l'esame di *Modelli Statistici*.

Sintassi esercizi

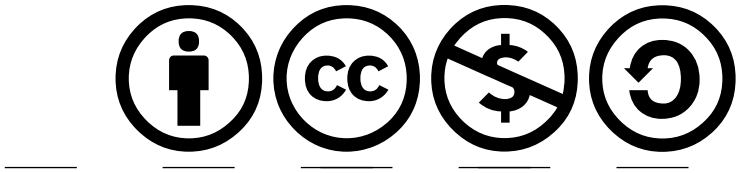
Per gli esercizi utilizziamo la sintassi **Exercise n[(m)]**, dove n rappresenta la numerazione locale all'interno delle sezioni di questo documento, m rappresenta l'identificativo dell'esercizio fissato nel documento
http://www.ds.unifi.it/gmm/resources/capitoli.pdf
Il reference (m) non è sempre presente, come evidenziato dall'uso delle parentesi quadre.

LICENZE

Solo questa sezione dedicata alle licenze verrà scritta completamente in inglese.

Text contents

All the text content is distributed under:
This work is licensed under the Creative Commons Attribution, NonCommercial, ShareAlike 3.0 Unported License. To view a copy of this license, visit
http://creativecommons.org/licenses/by-nc-sa/3.0/
or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.



Sources

All sources are distributed under, where the word "Software" is referred to all of the sources that are present in this work:
Copyright (c) 2011 Massimo Nocentini
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1

STIMA, VERIFICA D'IPOTESI E DISTRIBUZIONI CAMPIONARIE

Exercise 1.0.1. (2.6 nel testo) *Considerate il modello in cui* $Y_1, \ldots, Y_n$ *sono indipendenti e identicamente distribuite come una uniforme* $Y \sim U(0, \theta)$*, dove* θ *è il valore massimo che può assumere* Y*. Determinate per simulazione la distribuzione campionaria di* $T = 2Y$ *considerando* $\theta = 100, n = 20$*. Mostrate con un istogramma che la distribuzione dello stimatore è approssimabile da una normale. Qual è la varianza di* T*? Usate questa varianza per sovrapporre all'istogramma la curva normale che lo approssima (ricordate di disegnare l'istogramma in R con l'opzione freq = FALSE).*

Questo il codice che implementa l'esercizio:

```
twosix <- function(){
  dimension <- 10000
  uniformVector <- rep(0, dimension)
  for (i in 1:dimension) {
    sample <- runif(n=20, min=0, max=100)
    uniformVector[i] <- 2*mean(sample)
  }

  results <- list(estimatorVector=uniformVector,
                  empiricalMean=mean(uniformVector),
                  empiricalVar=var(uniformVector),
                  empiricalVarComputedByHand=sum((
                    uniformVector-mean(uniformVector))^2)/(
                      dimension-1),
                  sd=sqrt(var(uniformVector)))


  postscript("two-six.ps", horizontal = FALSE)  # set
     graphical output file
  hist(twoSixResults$estimatorVector,
       freq=FALSE,
       breaks=20,
       xlab = expression(T=2*bar(Y)),
       ylim=c(0, .04))
  lines(density(twoSixResults$estimatorVector), lty=2, col
     ="blue")
  curve(dnorm(x,
```

```
                mean = twoSixResults$empiricalMean,
                sd = sqrt(twoSixResults$empiricalVar)),
        from=50,
        to=160,
        add=TRUE,
        col="red")
  dev.off()  # close the graphical output file

  return(results)
}
```

Vediamo qualche risultato inserendo in R:

```
> twosix()
$estimatorVector
  [1]  95.96794 105.40497 110.64482  89.14495 107.99978
     116.03221
  94.73158 ...

$empiricalMean
[1] 100.2173

$empiricalVar
[1] 168.8848

$empiricalVarComputedByHand
[1] 168.8848

$sd
[1] 12.99557
```

L'esecuzione della funzione produce la Figura 1, dove la curva in blu rappresenta la densità inferita usando gli algoritmi disponibili in R dello stimatore $2\bar{Y}$, mentre la curva in rosso rappresenta il modello esatto usando come parametri i valori stimati dalla media e dalla varianza campionaria (riportate nel precedente output come `empiricalMean` e `empiricalVar`).

Exercise 1.0.2. (2.7 nel testo) *Una azienda chimica ha prodotto un additivo per la benzina che dovrebbe migliorare il consumo oltre le 25 miglia per gallone. Viene fatto un esperimento con 30 auto uguali su un percorso standard e si ottiene un consumo medio di 26.3 mpg con una deviazione standard campionaria di 2.4 mpg. Trovare un intervallo di confidenza al 95% per il consumo medio supponendo il modello normale.*

Questo il codice che implementa l'esercizio:

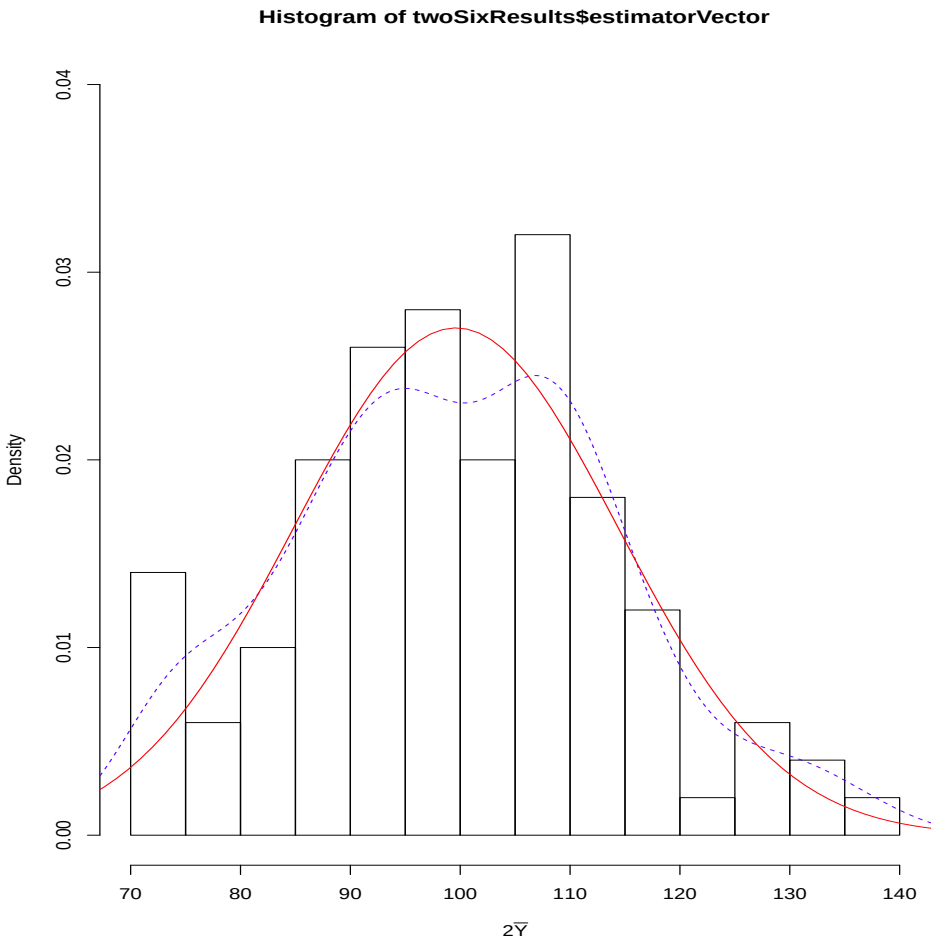


Figura 1: Istogramma esercizio 2.6 del testo

```
twoSeven <- function(){

  nullHp <- 25
  dimension <- 30
  level <- .05
  empiricalMean <- 26.3
  standardError <- 2.4/sqrt(dimension)
  superiorQuart <- qt(df=dimension-1, p=level/2, lower.
      tail=FALSE)
  tOss <- (empiricalMean - nullHp)/standardError

  return(list(
            tOss=tOss,
```

```
            confidenceInterval= c(empiricalMean -
                superiorQuart*standardError,
              empiricalMean + superiorQuart*standardError),
            pValue=pt(df=dimension-1, lower.tail=FALSE, q=
                tOss)
            ))
}
```

Nel campionamento ripetuto l'intervallo di confidenza riportato sotto conterrà il vero valore del parametro (consumo medio) con una probabilità di copertura del 95%.

```
> twoSeven()
$tOss
[1] 2.966831

$confidenceInterval
[1] 25.40383 27.19617

$pValue
[1] 0.002986329
```

Inoltre abbiamo condotto un test di significatività per verificare se l'additivo è efficacie o meno. Per questo mettiamo come ipotesi nulla H_0 :"con un gallone si fanno meno di 25 miglia" (che speriamo di rifiutare) contrapposta a H_1 :"con un gallone si fanno almeno 25 miglia". Dall'esecuzione del nostro codice otteniamo un *p-value* uguale a 0.002, pertanto il test risulta significativo, rifiutiamo H_0, l'additivo è efficacie. È da notare che in questo caso non è stato possibile utilizzare gli algoritmi messi a disposizione in R per il test di significatività in quanto non si ha il campione visibile (nel prossimo esercizio invece sarà possibile effettuare il test sia in modo automatico che manuale).

Exercise 1.0.3. (2.8 nel testo) *Un misuratore del tasso di alcool nel sangue viene verificato su un campione di prova in cui la misura dovrebbe essere 12%. Trovare una stima della media e il suo errore standard. Trovare un intervallo di confidenza a livello del 95% per la media col modello normale. Fare un test dell'ipotesi che il misuratore sia correttamente calibrato cioè che* $\mu = 12$ *contro l'alternativa che* $\mu \neq 12$.

Questo il codice che implementa l'esercizio:

```
twoEight <- function(){

  alcoholValues <-c
     (12.3,12.7,13.6,12.7,12.9,12.6,12.6,13.1,12.6,13.1,12.7,12.5,13.2,
```

```
                    12.8,12.4,12.6,12.4,12.4,13.1,12.9,13.3,12.6,12.6,1

                    12.4,13.1,12.4,12.9)
  nullHp <- 12
  dimension <- length(alcoholValues)
  level <- .05
  empiricalMean <- mean(alcoholValues)
  standardError <- sqrt(var(alcoholValues)/dimension)
  superiorQuart <- qt(df=dimension-1, p=level/2, lower.
      tail=FALSE)
  tOss <- (empiricalMean - nullHp)/standardError

  automaticTest <- t.test(alcoholValues, mu = nullHp,
      alternative = "two.sided")
  return(list(
            automaticTest=automaticTest,
            tOss=tOss,
            confidenceInterval= c(empiricalMean -
                superiorQuart*standardError,
              empiricalMean + superiorQuart*standardError),
            pValue=2*pt(df=dimension-1, lower.tail=FALSE, q
                =tOss)
            ))
}
```

Vediamo qualche risultato inserendo in R:

```
> twoEight()
$automaticTest

 One Sample t-test

 data:  alcoholValues
 t = 12.7718, df = 29, p-value = 1.964e-13
 alternative hypothesis: true mean is not equal to 12
 95 percent confidence interval:
 12.63550 12.87784
 sample estimates:
 mean of x
 12.75667


 $tOss
 [1] 12.77184
```

```
18
  $confidenceInterval
  [1] 12.63550 12.87784

  $pValue
23 [1] 1.963566e-13
```

Il test risulta altamente significativo, l'ipotesi nulla $\mu = 12$ viene rifiutata pertanto il misuratore non è correttamente calibrato.

Exercise 1.0.4. (2.10 nel testo) *Si studia un campione di 100 individui e si considera il numero di vegetariani. Si sono osservati* $r = 2$ *vegetariani su* $n = 100$ *prove (supposte Bernoulli indipendenti e identiche). Trovare l'intervallo di confidenza asintotico al 95% per la proporzione di vegetariani nella popolazione. Notare che l'intervallo ha il limite inferiore negativo. Calcolare anche l'intervallo di Agresti e Coull.*

Questo il codice che implementa l'esercizio (abbiamo implementato manualmente il metodo con la correzione Agresti-Coull in quanto nella distribuzione di R con cui sono state svolte queste implementazioni non fornisce la libreria `binom` per poter usare la funzione `binom.confint`):

```
twoTen <- function(){
2
  n <- 100
  r <- 2
  level <- .05
  empiricalMean <- r/n
7 superiorQuart <- qnorm(p=level/2, lower.tail=FALSE)

  empiricalMeanAgrestiCoull <- (r + 2)/(n + 4)
  return(
    list(
12    asymptotic=c(
        empiricalMean - superiorQuart * sqrt(empiricalMean
          *(1-empiricalMean)/n),
        empiricalMean + superiorQuart * sqrt(empiricalMean
          *(1-empiricalMean)/n)),
      AgrestiCoull=c(
        empiricalMeanAgrestiCoull - superiorQuart *
17      sqrt(empiricalMeanAgrestiCoull*(1-
          empiricalMeanAgrestiCoull)/(n+4)),
        empiricalMeanAgrestiCoull + superiorQuart *
        sqrt(empiricalMeanAgrestiCoull*(1-
          empiricalMeanAgrestiCoull)/(n+4)))))
}
```

Osserviamo dall'output riportato sotto che l'intervallo di confidenza calcolato con il metodo asintotico ha l'intervallo inferiore negativo, mentre non è così per quello calcolato con la correzione Agresti-Coull.

```
> twoTen()
$asymptotic
[1] -0.007439496  0.047439496

$AgrestiCoull
[1] 0.001501869 0.075421208
```

Exercise 1.0.5. (2.13 nel testo) *(Newbold et al.) In un centro di ricerca si vuole condurre uno studio sul costo medio dei biglietti del cinema. Si supponga che* $\sigma = 50$ *centesimi. Quale dimensione del campione dovremmo considerare per avere degli intervalli di confidenza del costo medio di ampiezza uguale a 30 centesimi?*

Supponiamo che gli n dati osservati provengono da un campione di v.a. identiche e identicamente distribuite con media e varianza μ, σ rispettivamente. Possiamo utilizzare il teorema limite centrale e impostare la forma dell'intervallo

$$\bar{Y} \pm z^*_{n,\frac{c}{2}} \frac{\sigma}{\sqrt{n}}$$

dove usiamo il quartile superiore della normale in quanto la deviazione standard è nota e non si deve "spendere" un grado di libertà utilizzando il relativo stimatore. Imponiamo la condizione sulla dimensione dell'intervallo:

$$\bar{Y} + z^*_{n,\frac{c}{2}} \frac{\sigma}{\sqrt{n}} - \bar{Y} + z^*_{n,\frac{c}{2}} \frac{\sigma}{\sqrt{n}} = 30$$

da cui si arriva a

$$n = \left(\frac{z^*_{n,\frac{c}{2}} \sigma}{15} \right)^2$$

aiutandosi con R, supponendo che l'intervallo di confidenza di livello 95%:

```
> (50*qnorm(p=.025, lower.tail=FALSE)/15)^2
[1] 42.68288
```

Exercise 1.0.6. (2.14 nel testo) *Si abbiano due variabili* X = *età della madre (anni),* Y = *età del padre (anni) misurate in una popolazione molto grande di bambini con la sindrome di Down. Si è trovato che la distribuzione congiunta di queste due variabili è normale doppia con medie* $\mu_X = 37.2, \mu_Y = 39.4$, *deviazioni standard* $\sigma_X = 6.8, \sigma_Y = 7.7$ *e coefficiente di correlazione* $\rho = 0.83$. *Trovare la covarianza tra le età. Trovare l'età media di una madre se il padre ha 40 anni. Trovare l'età media di un padre se la madre ha 40 anni. Determinare la retta delle medie condizionate* $E(Y|X = x)$ *nella popolazione.*

La covarianza campionaria si ricava dall'equazione

$$\rho = \frac{S_{XY}}{S_X S_Y}$$

Per i nostri dati abbiamo:

```
> covCampionaria <- .83 * 6.8 * 7.7
> covCampionaria
[1] 43.4588
```

Per le medie condizionate valgono le seguenti:

$$E(Y|X = x) = \bar{Y} + \frac{S_{XY}}{S_{XX}}(x - \bar{X})$$
$$E(X|Y = y) = \bar{X} + \frac{S_{XY}}{S_{YY}}(y - \bar{Y})$$

Di conseguenza le richieste dell'esercizio sono date da:

$$E(Y|X = 40) = 39.4 + \frac{43.4588}{6.8^2}(40 - 37.2)$$
$$E(X|Y = 40) = 37.2 + \frac{43.4588}{7.7^2}(40 - 39.4)$$

Calcolando:

```
> meanYGivenX <- 39.4 + 43.4588/(6.8^2)*(40-37.2)
> meanXGivenY <- 37.2 + 43.4588/(7.7^2)*(40-39.4)
> meanYGivenX
[1] 42.03159
> meanXGivenY
[1] 37.63979
```

Exercise 1.0.7. (2.15 nel testo) *Generare un campione casuale di 1000 osservazioni da una normale doppia con i parametri dell'esercizio precedente. Usare i dati per stimare i parametri in modo opportuno.*

Questo il codice che implementa l'esercizio:

```
twoFifteen <- function(){
  ## we generate a vector with 1000 items in order to
     build the plane
  generatingVector <- seq(from=-4, to=4, length.out=100)
  n <- length(generatingVector)^2
  matrix <- matrix(nrow=n, ncol=3)
  ro <- .83
  i <- 1
```

```
  for (u in generatingVector){
    for (v in generatingVector){
      matrix[i, 1] <- 37.2 + 6.8*u
      matrix[i, 2] <-  39.4 + 7.7*v
      matrix[i, 3] <- multivariateStandardNormal(u, v, ro)
      i <- i+1
    }
  }

  rowIndicesForSample <- sample(x=1:n, size=1000, prob=
      matrix[,3])
  sample <- matrix[rowIndicesForSample,]

  postscript("two-fifteen.ps", horizontal = FALSE)
  plot(sample, xlab="X = 37.2 + 6.8*U", ylab="Y = 39.4 +
      7.7*V")
  lines(sample[,1], 39.4 + 43.4588/(6.8^2)*(sample
      [,1]-37.2), col="red")
  ## here simply we swap x and y datas in order to draw
      the curve
  ## correctly (that is we have to switch the two axis in
      order to
  ## have a meaningful plot)
  lines(37.2 + 43.4588/(7.7^2)*(sample[,2]-39.4), sample
      [,2], col="blue")
  dev.off()

  return(list(matrix=matrix,
              sample=sample))
}

multivariateStandardNormal <- function(u, v, ro){
  coeff <- (2*pi*sqrt(1-ro^2))^(-1)
  return(coeff * exp(-((u^2 -2*ro*u*v +v^2)/(2*(1-ro^2)))
      )
}
```

L'esecuzione della funzione produce la Figura 2 con una rappresentazione del campione di dimensione 1000 della variabile aleatoria $(X = 37.2 + 6.8U, Y = 39.4 + 7.7V)$, dove (U, V) è una normale doppia standard. Nella figura riportiamo lo scatter delle coppie generate, una curva in rosso per la retta delle medie $E(Y|X = x)$ e una curva in blu per la retta delle medie $E(X|Y = y)$.

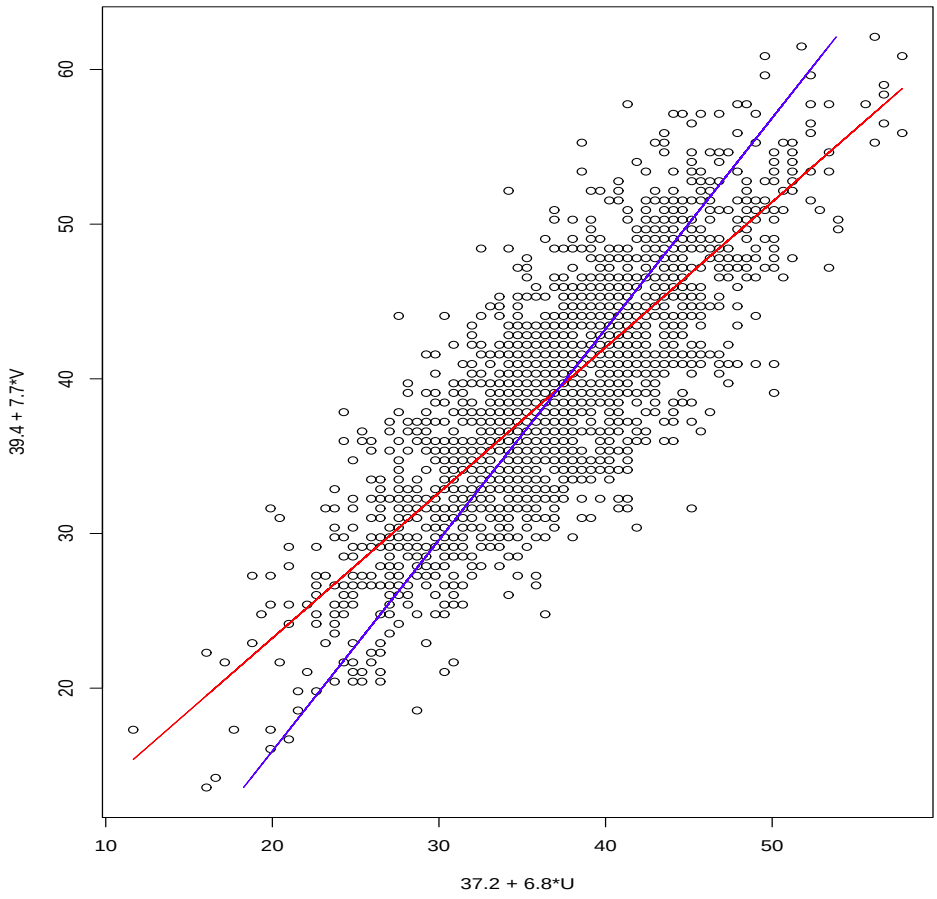


Figura 2: Campione normale doppia (X, Y), esercizio 2.15 del testo