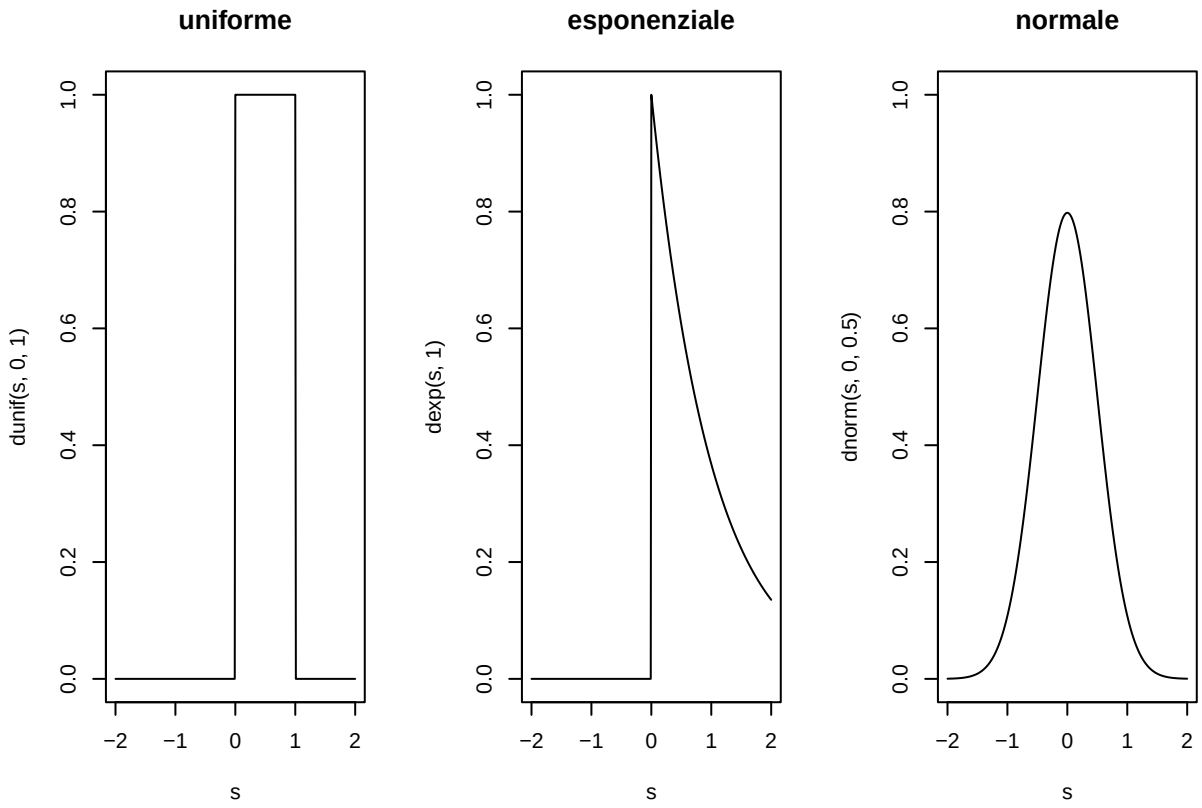
Laboratorio con R - 0

Metodi e Modelli per l'Inferenza Statistica - Ing. Matematica - a.a. 2023-24

Uso e rappresentazione delle principali distribuzioni

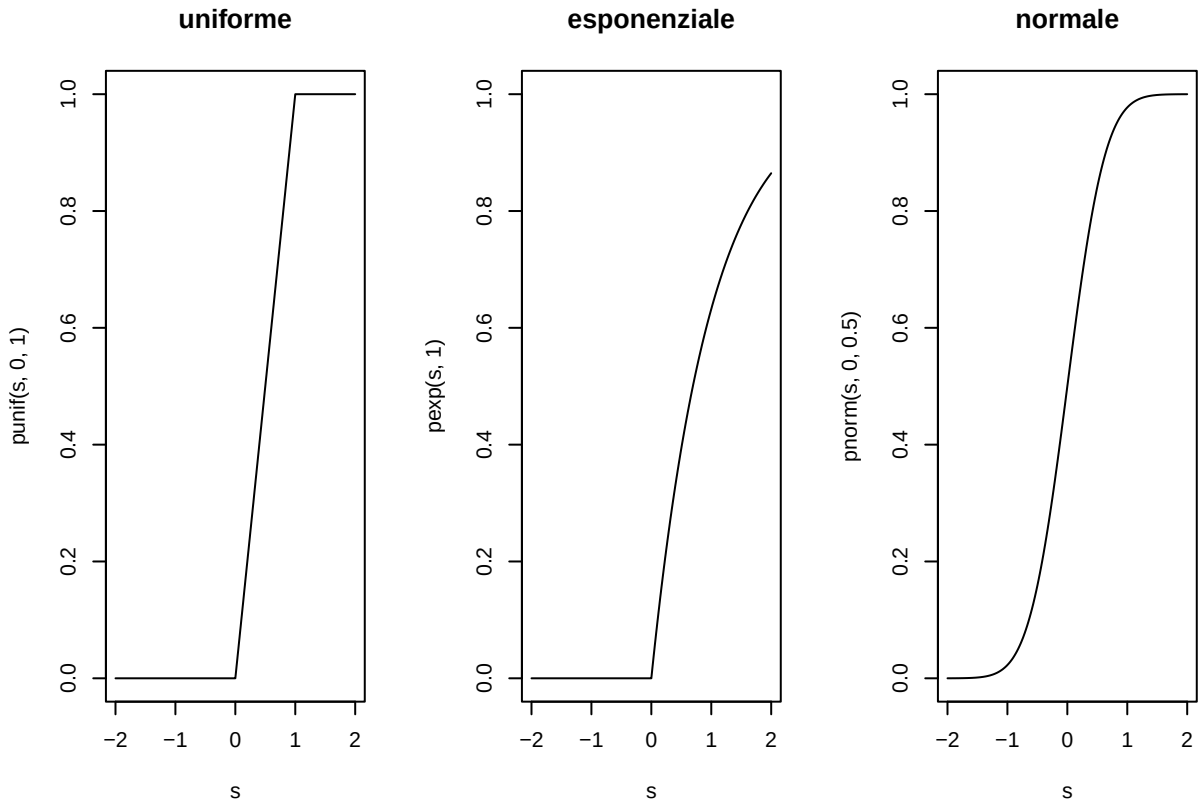
Densità

```
# dividiamo la schermata di output delle figure in tre caselle disposte su una riga
#dev.new()
layout( matrix( c( 1, 2, 3), 1, byrow = T ) )
# o in alternativa
#par( mfrow = c( 3, 1 ) )

s = seq( -2, 2, by = 0.01 )

# tre esempi di densità
plot( s, dunif( s, 0, 1 ), main = "uniforme", type = "l", ylim = c( 0, 1 ) )
plot( s, dexp( s, 1 ), main = "esponenziale", type = "l", ylim = c( 0, 1 ) )
plot( s, dnorm( s, 0, 0.5 ), main = "normale", type = "l", ylim = c( 0, 1 ) )
```



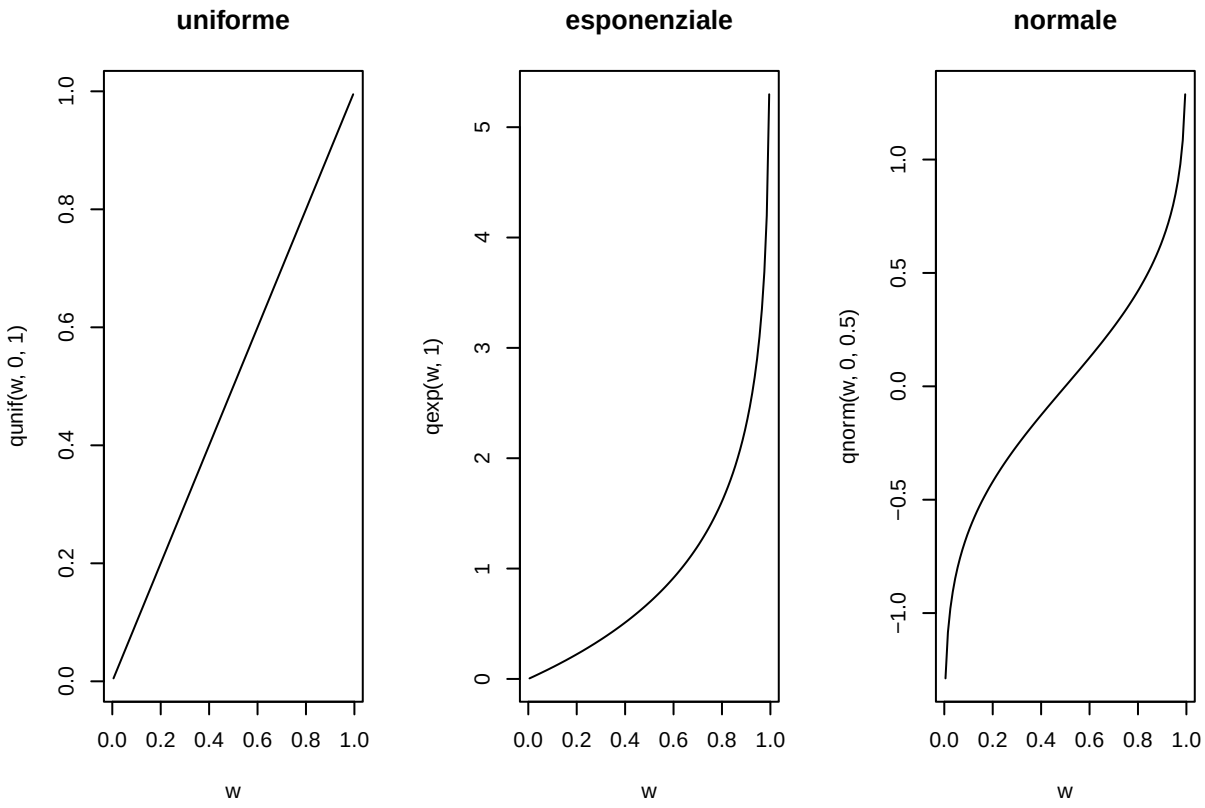
Funzioni di Ripartizione

```r
layout( matrix( c( 1, 2, 3), 1, byrow = T ) )

# tre esempi di funzioni di distribuzione
plot( s, punif( s, 0, 1 ), main = "uniforme", type = "l", ylim = c( 0, 1 ) )
plot( s, pexp( s, 1 ), main = "esponenziale", type = "l", ylim = c( 0, 1 ) )
plot( s, pnorm( s, 0, 0.5 ), main = "normale", type = "l", ylim = c( 0, 1 ) )
```



Inversa della funzione di ripartizione (quantili)

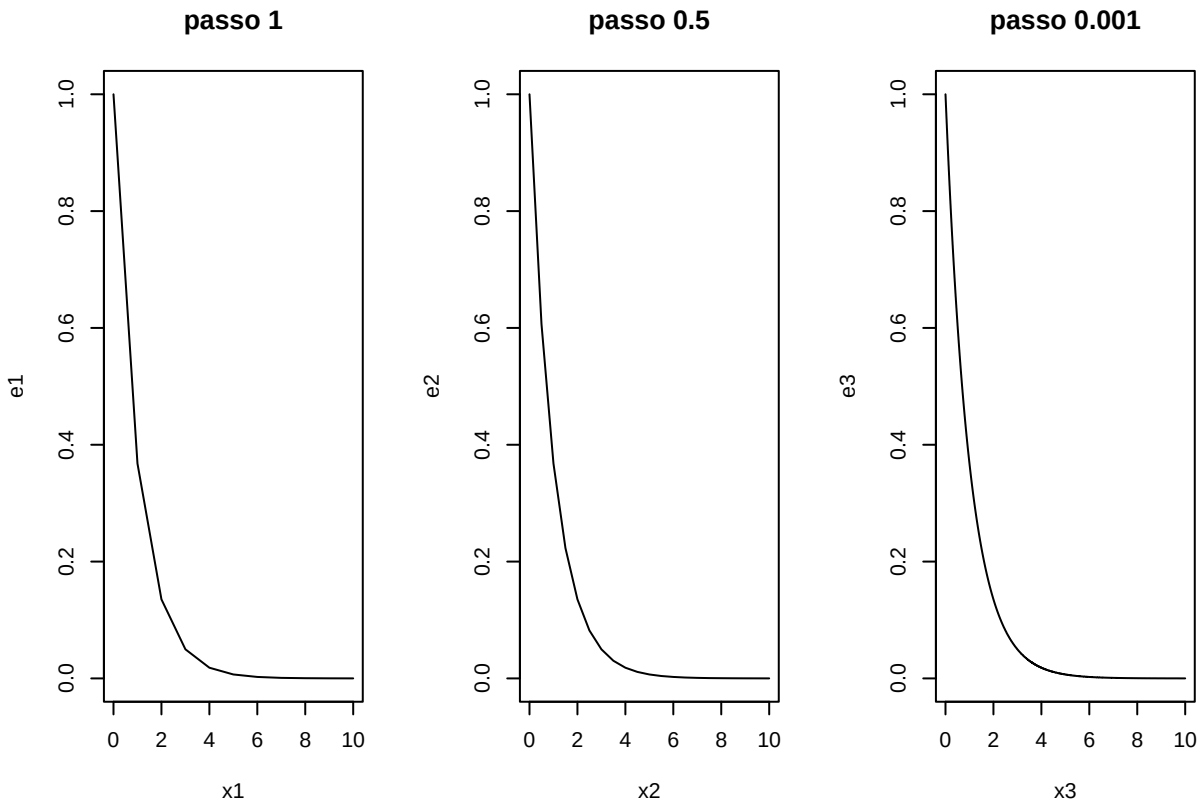
```r
# ricordiamoci di fornire i quantili, escludendo 0 e 1 dal supporto.
# su tali punti la funzione andrebbe a + e - infinito
layout( matrix( c( 1, 2, 3), 1, byrow = T ) )

w = seq( 0.01 / 2, 1 - 0.01 / 2, by = 0.01 )
plot( w, qunif( w, 0, 1 ), main = "uniforme", type = "l" )
plot( w, qexp( w, 1 ), main = "esponenziale", type = "l" )
plot( w, qnorm( w, 0, 0.5 ), main = "normale", type = "l" )
```



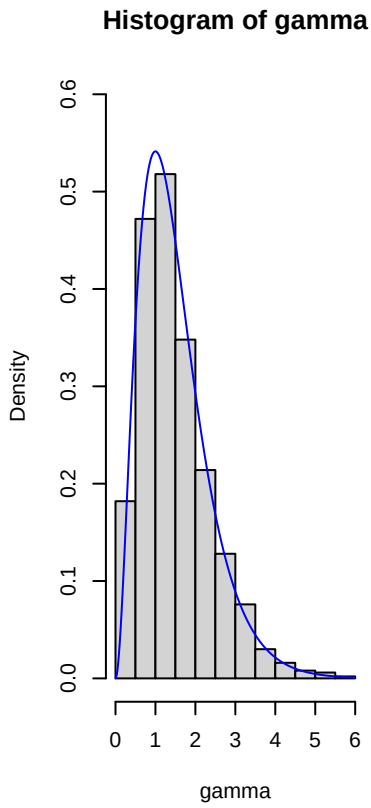
Distribuzioni note

Somma di Esponenziali è una Gamma.

```
# Raffinamento
x1 = seq( 0, 10, 1 )
x2 = seq( 0, 10, 0.5 )
x3 = seq( 0, 10, 0.001 )
e1 = dexp( x1 )
e2 = dexp( x2 )
e3 = dexp( x3 )

par( mfrow = c( 1, 3 ) )
plot( x1, e1, type = "l", main = "passo 1" )
plot( x2, e2, type = "l", main = "passo 0.5" )
plot( x3, e3, type = "l", main = "passo 0.001" )
```



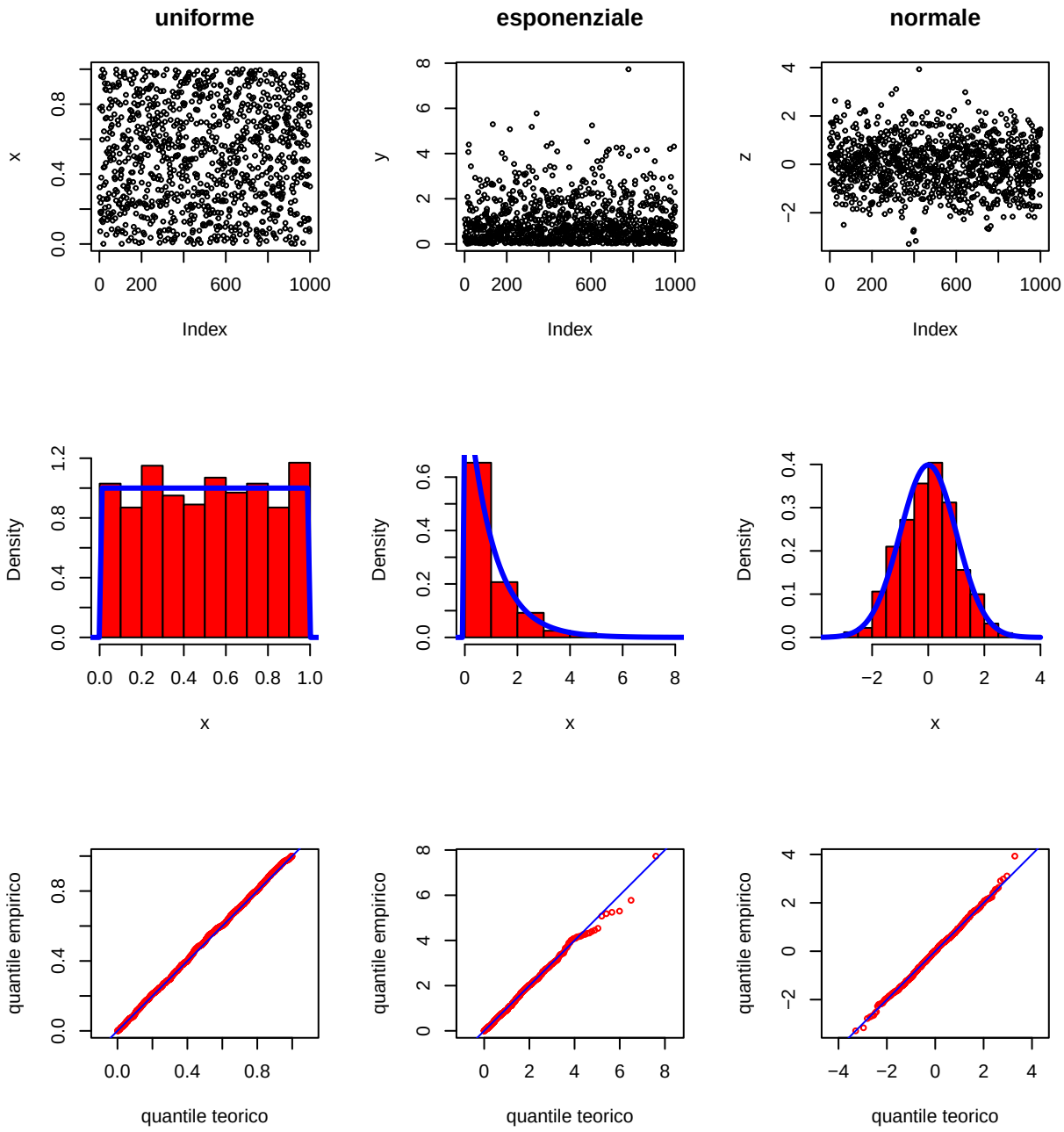

```
# Somma di esponenziali
exp1 = rexp( 1000, 2 )
exp2 = rexp( 1000, 2 )
exp3 = rexp( 1000, 2 )
gamma = exp1 + exp2 + exp3

#dev.new()
hist( gamma, prob = T, ylim = c( 0, 0.6 ) )
grid = seq( 0, 6, 0.01 )
y = dgamma( grid, 3, 2 )
lines( grid, y, col = "blue" )
```



Generazione di campioni casuali

```
x = runif( n = 1000, min = 0, max = 1 )
y = rexp( n = 1000, rate = 1 )
z = rnorm( n = 1000, mean = 0, sd = 1 )

# dividiamo la schermata di output delle figure in 9 caselle disposte su tre righe
#dev.new()
mat = matrix( c( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), 3, byrow = T )
layout(mat, widths = rep.int(1, ncol(mat)),
       heights = rep.int(4, nrow(mat)))

plot( x, main = "uniforme", cex = .5)
plot( y, main = "esponenziale", cex = .5)
plot( z, main = "normale", cex = .5)

hist( x, main = "", col = "red", xlab = "x", prob = T )
lines( seq( -0.2, 1.2, length = 100 ), dunif( seq( -0.2, 1.2, length = 100 ) ),
       col = "blue", lty = 1, lwd = 3 )

hist( y, main = "", col = "red", xlab = "x", prob = T )
lines( seq( -1, 9, length = 100 ), dexp( seq( -1, 9, length = 100 ) ),
       col = "blue", lty = 1, lwd = 3 )

hist( z, main = "", col = "red", xlab = "x", prob = T )
lines( seq( -4, 4, length = 100 ), dnorm( seq( -4, 4, length = 100 ) ),
       col = "blue", lty = 1, lwd = 3 )
```

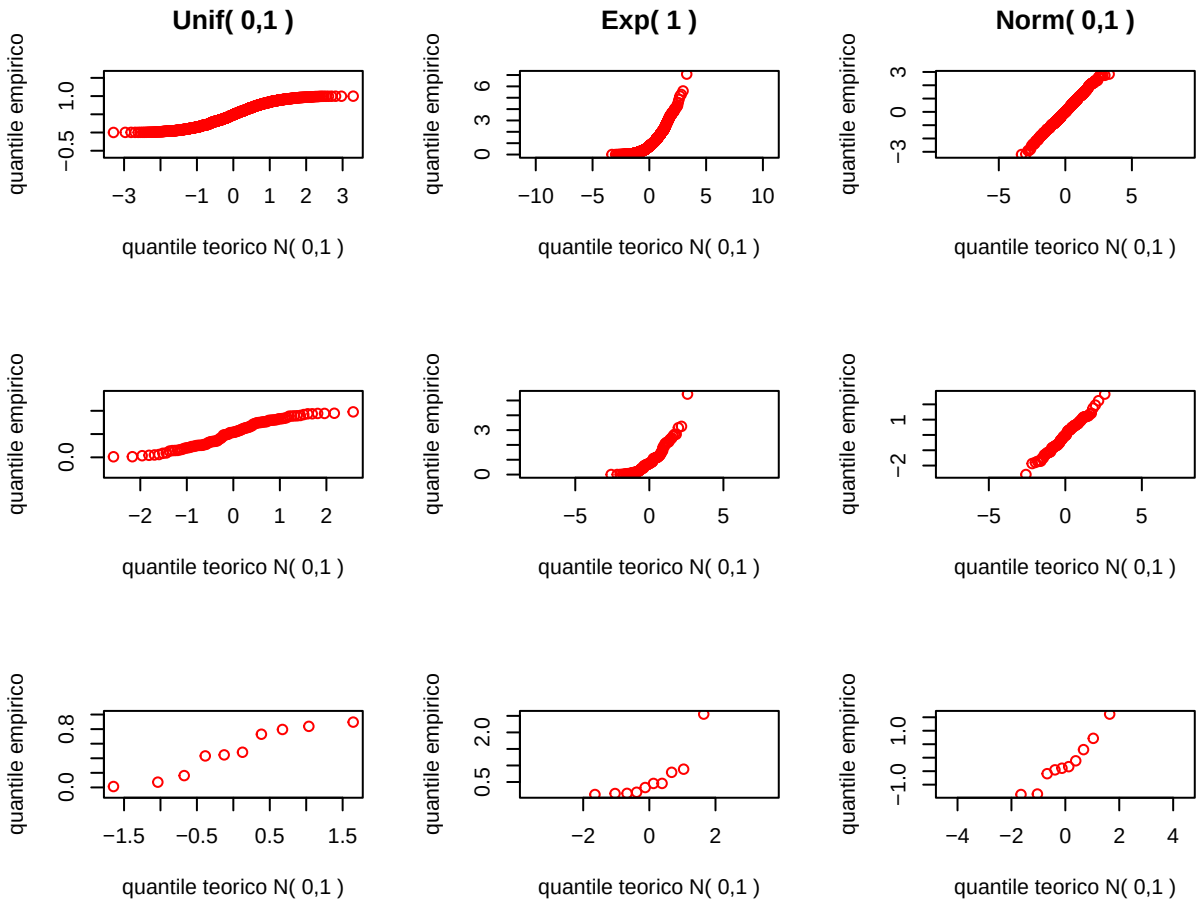
```r
qqplot( qunif( ( 1:1000 / 1000 - 0.5 / 1000 ) ), x, col = "red",
        xlab = "quantile teorico", ylab = "quantile empirico", asp = 1, cex = .6  )
abline( 0, 1, col = "blue" )

qqplot( qexp( ( 1:1000 / 1000 - 0.5 / 1000 ) ), y, col = "red",
        xlab = "quantile teorico", ylab = "quantile empirico", asp = 1, cex = .6  )
abline( 0, 1, col = "blue" )

qqplot( qnorm( ( 1:1000 / 1000 - 0.5 / 1000 ) ), z, col = "red",
        xlab = "quantile teorico", ylab = "quantile empirico", asp = 1, cex = .6  )
abline( 0, 1, col = "blue" )
```



Utilizzo del qqplot per vedere qualitativamente se un campione è estratto da una certa popolazione.

```
#dev.new()
mat = matrix( c( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), 3, byrow = T )
layout(mat, widths = rep.int(1, ncol(mat)),
       heights = rep.int(4, nrow(mat)))

# n = 1000
x = runif( n = 1000, min = 0, max = 1 )
y = rexp( n = 1000, rate = 1 )
z = rnorm( n = 1000, mean = 0, sd = 1 )
```

```r
qqplot( qnorm( ( 1:1000 / 1000 - 1 / 2000 ) ), x, col = "red",
        xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1, main = "Unif( 0,1 )" )
qqplot( qnorm( ( 1:1000 / 1000 - 1 / 2000 ) ), y, col = "red",
        xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1, main = "Exp( 1 )" )
qqplot( qnorm( ( 1:1000 / 1000 - 1 / 2000 ) ), z, col = "red",
        xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1, main = "Norm( 0,1 )" )

# n = 100
x = runif( n = 100, min = 0, max = 1 )
y = rexp( n = 100, rate = 1 )
z = rnorm( n = 100, mean = 0, sd = 1 )

qqplot( qnorm( ( 1:100 / 100 - 1 / 200 ) ), x, col = "red", xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1 )
qqplot( qnorm( ( 1:100 / 100 - 1 / 200 ) ), y, col = "red", xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1 )
qqplot( qnorm( ( 1:100 / 100 - 1 / 200 ) ), z, col = "red", xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1 )

# n = 10
x = runif( n = 10, min = 0, max = 1 )
y = rexp( n = 10, rate = 1 )
z = rnorm( n = 10, mean = 0, sd = 1 )

qqplot( qnorm( ( 1:10 / 10 - 1 / 20 ) ), x, col = "red", xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1 )
qqplot( qnorm( ( 1:10 / 10 - 1 / 20 ) ), y, col = "red", xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1 )
qqplot( qnorm( ( 1:10 / 10 - 1 / 20 ) ), z, col = "red", xlab = "quantile teorico N( 0,1 )",
        ylab = "quantile empirico", asp = 1 )
```



Test di normalità univariata

Il test di Shapiro-Wilk è un test per la verifica dell'ipotesi di normalità. Venne introdotto nel 1965 da Samuel Shapiro e Martin Wilk. La verifica della normalità avviene confrontando due stimatori alternativi della varianza σ^2:

- uno stimatore non parametrico basato sulla combinazione lineare ottimale della statistica d'ordine di una variabile aleatoria normale, al numeratore, e
- il consueto stimatore parametrico, ossia la varianza campionaria, al denominatore.

La statistica del test risulta essere:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

dove i coefficienti $a_{(i)}$ sono calcolati a partire dai ranghi di un numero casuale standardizzato.

La statistica W che se ne ricava può assumere valori da 0 a 1. Qualora il valore della statistica W sia troppo piccolo, il test rifiuta l'ipotesi nulla che i valori campionari siano distribuiti come una variabile casuale normale.

$$H_0 : X \sim \mathcal{N}ormal \qquad vs \qquad H_1 : X \sim F \neq \mathcal{N}ormal$$

```
x = rnorm( n = 1000, mean = 0, sd = 1 )
y = rnorm( n = 1000, mean = 2, sd = 5 )
z = rexp( n = 1000, 0.5 )
```

```r
shapiro.test( x )
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.99871, p-value = 0.6976
shapiro.test( y )
##
##  Shapiro-Wilk normality test
##
## data:  y
## W = 0.99832, p-value = 0.4406
shapiro.test( z )
##
##  Shapiro-Wilk normality test
##
## data:  z
## W = 0.83184, p-value < 2.2e-16
```

Probabilità di copertura IC

Si generi un dataset di 100 elementi da una normale di media 4 e varianza 2.

```r
set.seed(1200)
dati.sim = rnorm(  100, 4, sqrt( 2 ) )
```

Eseguiamo il seguente test:

$$H_0 : \mu = 4 \qquad vs \qquad H_1 : \mu \neq 4$$

Ricordiamo la natura duale dei test di verifica di ipotesi e degli intervalli di confidenza. Per ciascun $\theta_0 \in \Theta$, sia $A(\theta_0)$ la Regione di Accettazione di livello α del test $H_0 : \theta = \theta_0$. Per ciascun $x \in X$, si definisca un intervallo $IC(x)$ come:

$$IC(x) = \{\theta_0 : x \in A(\theta_0)\}.$$

Allora $IC(x)$ è un intervallo di confidenza di livello $1 - \alpha$. Alternativamente, possiamo definire per ogni $\theta_0 \in \Theta$ la regione di accettazione di livello α associata al test $H_0 : \theta = \theta_0$ come:

$$A(\theta_0) = \{x : \theta_0 \in IC(x)\}.$$

Caso 1: varianza nota

$$IC = [\bar{x} - z_{1-\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}, \bar{x} + z_{1-\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}]$$

```r
alpha = 0.05
n = length( dati.sim )
sigma = sqrt( 2 )

media = mean( dati.sim )

IC.noto = c( inf = media - sigma / sqrt( n ) * qnorm( 1 - alpha / 2 ),
             center = media, sup = media  +  sigma / sqrt( n ) * qnorm( 1 - alpha / 2 ) )

IC.noto
##      inf   center      sup
## 3.740484 4.017664 4.294845
```

Caso 2: varianza incognita

$$IC = [\bar{x} - t_{1-\alpha/2,n-1} \cdot \frac{s}{\sqrt{n}}, \bar{x} + t_{1-\alpha/2,n-1} \cdot \frac{s}{\sqrt{n}}]$$

dove $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$.

```
alpha = 0.05
n = length( dati.sim )
devst = sd( dati.sim )

IC.inc = c( inf = media - devst / sqrt( n ) * qt( 1 - alpha / 2, n - 1 ),
            center = media, sup = media  +  devst / sqrt( n ) * qt( 1 - alpha / 2, n - 1 ) )
IC.inc
##      inf   center      sup
## 3.713329 4.017664 4.322000
```

Confronto tra IC:

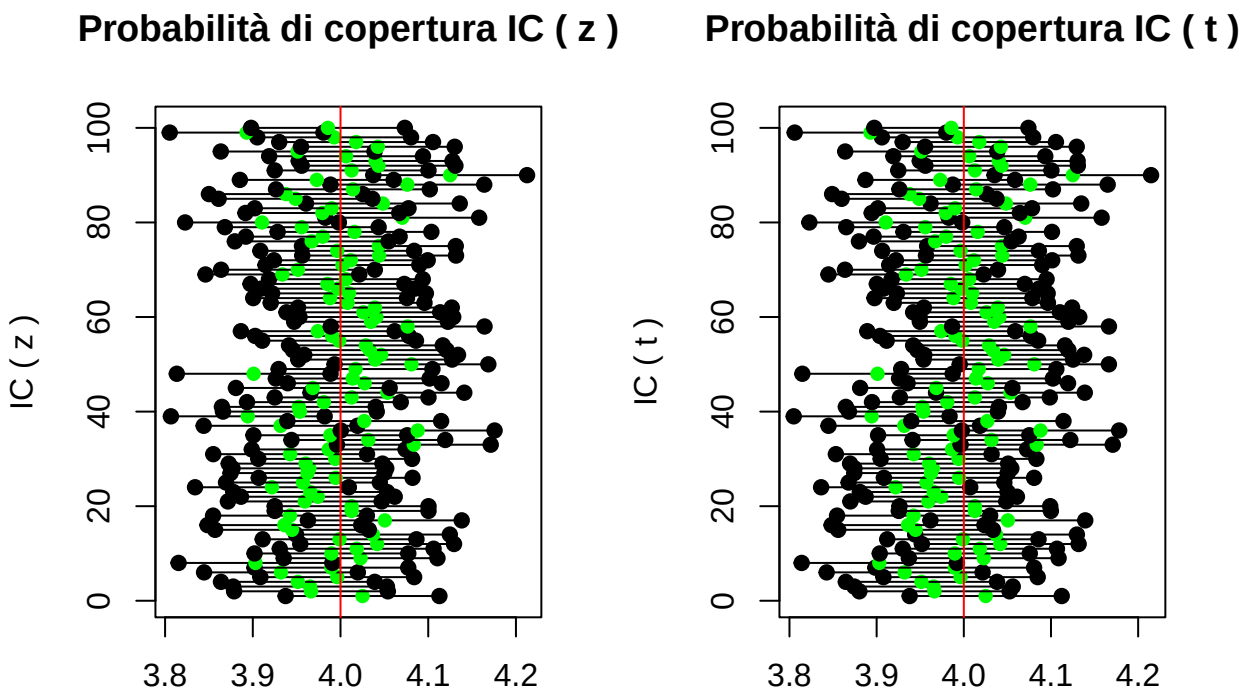
```
rbind( IC.noto, IC.inc )
##              inf   center      sup
## IC.noto 3.740484 4.017664 4.294845
## IC.inc  3.713329 4.017664 4.322000

IC.noto[3] - IC.noto[1]
##       sup
## 0.5543615
IC.inc[3] - IC.inc[1]
##       sup
## 0.6086713
```

REMARK L'IC costruito con i quantili della Normale (caso varianza nota) è più stretto di quello costruito con i quantili della t (la t di Student ha infatti code più pesanti).

Stimiamo la probabilità di copertura degli intervalli.

```
N = 100  # Numero di intervalli
n = 1000  # Numero campioni dalla Normale
alpha = 0.05  # livello di confidenza

mat.IC.z = matrix( NA, N, 3 )
mat.IC.t = matrix( NA, N, 3 )
sigma = sqrt( 2 )
for ( i in 1:N ) {
  sample = rnorm( n, 4, sqrt( 2 ) )

  mat.IC.z[ i, 1 ] = mean( sample ) - sigma / sqrt( n ) * qnorm( 1 - alpha / 2 )
  mat.IC.z[ i, 2 ] = mean( sample )
  mat.IC.z[ i, 3 ] = mean( sample ) +  sigma / sqrt( n ) * qnorm( 1 - alpha / 2 )

  mat.IC.t[ i, 1 ] = mean( sample ) - sd( sample ) / sqrt( n ) * qt( 1 - alpha / 2, n - 1 )
  mat.IC.t[ i, 2 ] = mean( sample )
  mat.IC.t[ i, 3 ] = mean( sample ) +  sd( sample ) / sqrt( n ) * qt( 1 - alpha / 2, n - 1 )
}


par( mfrow = c( 1, 2 ) )
plot( range( mat.IC.z ), c( 0.5, N  +  0.5 ), pch = "",
```

```
      xlab = "", ylab = "IC ( z )", main = "Probabilità di copertura IC ( z )" )
for ( k in 1:N ) {
  lines( c( mat.IC.z[ k, 1 ], mat.IC.z[ k, 3 ] ), c( k, k ) )
  points( mat.IC.z[ k, 1 ], k, pch = 19 )  #95
  points( mat.IC.z[ k, 2 ], k, pch = 16, col = "green" )
  points( mat.IC.z[ k, 3 ], k, pch = 19 )
}
abline( v = 4, col = "red" )

# #dev.new()
plot( range( mat.IC.t ), c( 0.5, N  +  0.5 ), pch = "",
      xlab = "", ylab = "IC ( t )", main = "Probabilità di copertura IC ( t )" )
for ( k in 1:N ) {
  lines( c( mat.IC.t[ k, 1 ], mat.IC.t[ k, 3 ] ), c( k, k ) )
  points( mat.IC.t[ k, 1 ], k, pch = 19 )  #95
  points( mat.IC.t[ k, 2 ], k, pch = 16, col = "green" )
  points( mat.IC.t[ k, 3 ], k, pch = 19 )
}
abline( v = 4, col = "red" )
```




```
test.cop.z = NULL
test.cop.t = NULL
for ( i in 1:N ) {
  test.cop.z[ i ] = 4 < mat.IC.z[ i, 3 ] & 4 > mat.IC.z[ i, 1 ]
```

```
  test.cop.t[ i ] = 4 < mat.IC.t[ i, 3 ] & 4 > mat.IC.t[ i, 1 ]
}
cop.z = as.numeric( test.cop.z )
cop.t = as.numeric( test.cop.t )
sum( cop.z ) / N
## [1] 0.93
sum( cop.t ) / N
## [1] 0.94


# Lunghezza media
l.z = NULL
l.t = NULL
for ( i in 1:N ) {
  l.z[ i ] = mat.IC.z[ i, 3 ] - mat.IC.z[ i, 1 ]
  l.t[ i ] = mat.IC.t[ i, 3 ] - mat.IC.t[ i, 1 ]
}
mean( l.z )
## [1] 0.1753045
mean( l.t )  # maggiore
## [1] 0.1755642
```

Notiamo che la media della lughezza degli intervalli costruiti sotto l'ipotesi di varianza incognita è più grande della media della lughezza degli intervalli costruiti sotto l'ipotesi di varianza nota, ma al crescere di n gli intervalli tenderanno a coincidere perchè la t-student si avvicinerà sempre di più alla normale.

Test di ipotesi per la media su uno o due campioni

Argomenti trattati:

1. test per la media in ipotesi di normalità: verifica della probabilità di errore di I tipo, di II tipo, e della potenza del test, tramite simulazione;
2. esempio di inferenza per la media in ipotesi di normalità su un dataset reale, considerando uno o due campioni.

Esercizi di simulazione

```
numero.casuale = 819260647

set.seed( numero.casuale )
```

Esercizio 1

1.a

Consideriamo un test bilatero per la media μ di una popolazione gaussiana di varianza nota $\sigma^2 = 6.25$. Sia il test :

$$H_0 : \mu = 50 \qquad vs \qquad H_1 : \mu \neq 50$$

La regione critica del test bilatero di livello α è

$$R_\alpha = \left\{ \frac{|\bar{X} - 50|}{\sigma/\sqrt{n}} > z_{(1-\alpha/2)} \right\}$$

Vogliamo ora verificare, tramite simulazione, che l'errore di I tipo venga commesso proprio con probabilità α, come sappiamo dalla teoria per i test di livello α.

Soluzione Simuliamo 100,000 realizzazioni di un campione di 14 variabili aleatorie gaussiane con media $\mu = 50$ e deviazione standard $\sigma = 2.5$ nota. Calcoliamo quindi la percentuale di realizzazioni campionarie che ci portano a rifiutare H_0.

REMARK L'errore del I tipo è la probabilità di rifiutare erroneamente l'ipotesi nulla, e accettare l'ipotesi alternativa (ovvero accettare un falso positivo).

```
N = 1e+5
n = 14
sigma = 2.5
# media vera della popolazione da cui provengono i campioni
mu = 50
# media ipotizzata in H_0!
mu.0 = 50
# mi metto nella situazione in cui H_0 è vera per verificare errore di primo tipo..
# livello teorico del test
alpha = 0.05

# vettore che conterrà il risultato del test ad ogni iterazione
esito = rep( 0, N )

for ( i in 1: N ) { # ripeto il test N volte

  # ad ogni iterazione simulo i dati gaussiani su cui effettuare il test
  dati.sim = rnorm( n, mean = mu, sd = sigma )

  media.camp = mean( dati.sim )

  # calcolo la soglia della regione critica: è il quantile della Normale
  z.alpha = qnorm( 1 - alpha / 2 )

  # calcolo la statistica test
  Z.0 = abs( media.camp - mu.0 ) / ( sigma / sqrt( n ) )

  # effettuo il test:  esito = 1 se rifiuto, 0 se accetto
  esito[ i ] = ifelse( Z.0 > z.alpha, 1, 0)
}

# calcolo una stima della probabilità di errore di primo tipo
alpha.camp = mean( esito )
alpha.camp
## [1] 0.04992
```

La stima di α è molto vicina all'errore di I tipo reale.

ESERCIZIO PER CASA
Provare cosa succede per un numero di tentativi N che va da 10 a 100,000 con passo 100 e rappresentare l'andamento della variabile α.

1.b

Consideriamo ora un test unilatero per la media μ di una popolazione Gaussiana di varianza nota σ^2. Sia il test:

$$H_0 : \mu \leq 0 \qquad vs \qquad H_1 : \mu > 0$$

La regione critica del test unilatero di livello α è:

$$R_\alpha = \{\bar{X} > 0 + z_{(1-\alpha)} \cdot \sigma/\sqrt{n}\}$$

Vogliamo valutare l'andamento dell'errore di II tipo, β, e della POTENZA del test, rispetto alla violazione dell'ipotesi nulla (ovvero all'allontanarsi dal valore 0 della media vera della popolazione, μ).

Soluzione

L'obiettivo è disegnare le funzioni β e la potenza del test. Impostiamo i valori della media vera della popolazione: è un vettore di possibili violazioni di H_0, che vanno da una violazione blanda ($\mu = 0.5$) a una più estrema ($\mu = 5$). Imposto anche il livello α.

REMARK L'errore di II tipo è la probabilità di accettare H_0 quando è falsa, ovvero di accettare un falso negativo (in questo caso, quando è vero ch $\mu > 0$).

```
mu = seq( 0.5, 5, by = 0.01 )

# devo stabilire le caratteristiche del campione che sto considerando
n = 30
sigma = 3
alpha = 0.01
# devo fissare il livello del test per trovare la potenza!

# media ipotizzata sotto H_0
mu.0 = 0
```

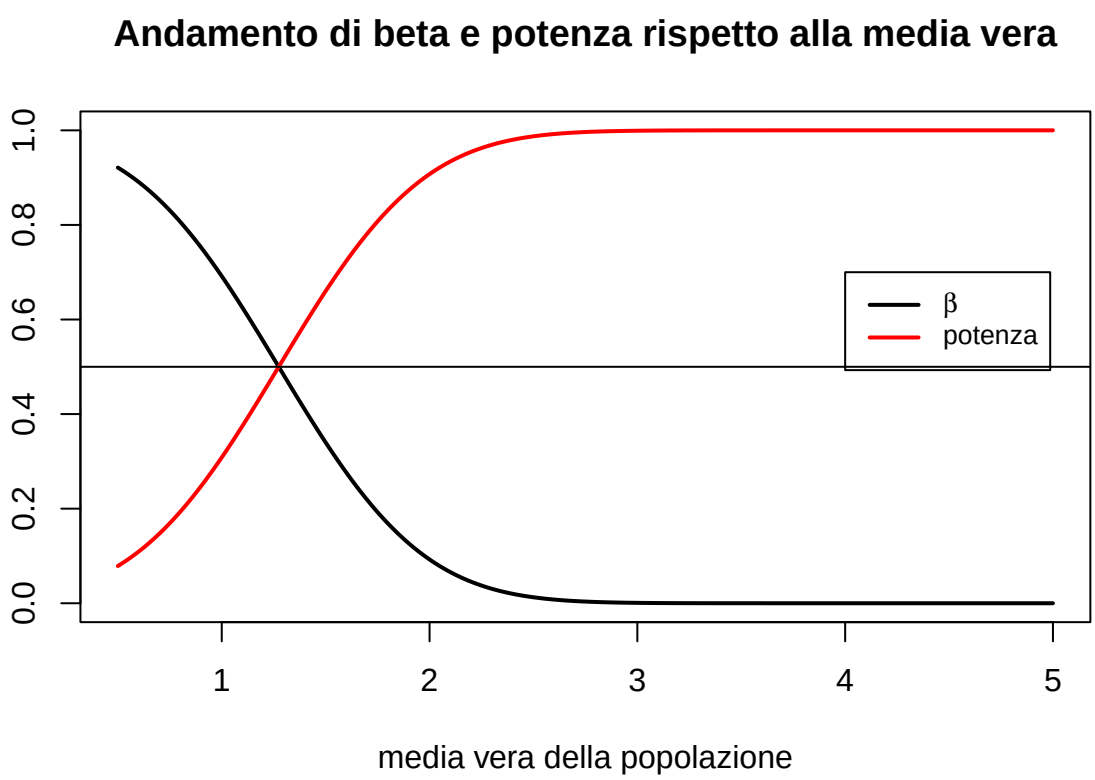
EX Provare a vedere cosa cambia cambiando il livello.

Calcolo il quantile di ordine $1 - \alpha$ della normale standard.

```
z.alpha = qnorm( 1 - alpha )
beta = pnorm( z.alpha - mu / sigma * sqrt( n ) )
potenza = 1 - beta
```

Disegno l'andamento delle funzioni calcolate.

```
plot( mu, beta, type = "l", lwd = 2, ylim = range( cbind( beta, potenza ) ),
      main = "Andamento di beta e potenza rispetto alla media vera",
      xlab = "media vera della popolazione", ylab = "" )
lines( mu, potenza, lwd = 2, col = "red" )
legend( 4, 0.7, legend = c( expression( beta ), "potenza" ), col = c( "black", "red" ),
        lwd = 2, cex = 0.85 )
abline( h = 0.5 )
```



Notiamo che più μ è vicina a 0, più β cresce e la potenza descresce, mentre allontanandosi da 0 (e quindi dall'ipotesi nulla, secondo cui μ è negativa) β tende a 0 e la potenza ad 1.

N.B. questo è l'andamento TEORICO di β e della potenza al variare di μ, vediamo ora cosa succede empiricamente, simulando dei dati di media μ fissata (scegliamo alcuni valori di μ), e valutando il β campionario.

- Vogliamo calcolare l'errore di II tipo teorico in corrispondenza di alcuni valori fissati di μ, e valutare quindi tramite simulazione che l'errore di II tipo venga commesso proprio con probabilità β, come sappiamo dalla teoria.

```
# valori selezionati della media vera della popolazione sono quelli in base ai quali
#  simulerò i campioni di dati
mu.sel = c( 1, 1.5, 3 )

# valori teorici di beta e potenza in corrispondenza delle scelte fatte per mu
beta.sel = beta[ match( mu.sel, mu ) ]
beta.sel
## [1] 0.6916757861 0.3400726313 0.0008139032

potenza.sel = 1 - beta.sel
potenza.sel
## [1] 0.3083242 0.6599274 0.9991861

# quante simulazioni?
N = 1000

esito = matrix( 0, N, length( mu.sel ) )
```

```
for ( i in 1: N )
{
  # ad ogni iterazione simulo i dati gaussiani su cui effettuare il test
  # poiché ho diversi valori di mu da cui simulare!

  for ( j in 1: length( mu.sel ) )
  {
    dati.sim = rnorm( n, mean = mu.sel[ j ], sd = sigma )

    media.camp = mean( dati.sim )

    # calcolo la statistica test :
    Z_0 = ( media.camp - mu.0 ) / sigma * sqrt( n )

    # effettuo il test:  esito = 1 se rifiuto, 0 se accetto
    esito[ i, j ] = ifelse( Z_0 > z.alpha, 1, 0 )
  }
}

# potenza empirica = proporzione di volte in cui ho effettivamente rifiutato
potenza.camp = colMeans( esito )
potenza.camp
## [1] 0.288 0.634 0.997
potenza.sel
## [1] 0.3083242 0.6599274 0.9991861


beta.camp = 1 - potenza.camp
beta.camp
## [1] 0.712 0.366 0.003
beta.sel
## [1] 0.6916757861 0.3400726313 0.0008139032
```

C'è un ottimo accordo tra valori teorici e valori derivanti dalla simulazione. Il che significa che il test sta effettivamente funzionando come ci aspettiamo in base alla teoria.

Esercizio 2: test sulla media e sulla proporzione

Carichiamo i dati contenuti nel file outcomes.txt [fonte: database clinico Lombardia]. Il database contiene 963 pazienti e l'osservazione di 4 variabili :

- **PRESSIONE**: pressione arteriosa sanguigna;
- **ST_RESOLUTION_70_60**: riduzione dello slivellamento del tratto ECG a 1 ora dall' intervento (angioplastica): 1 = si, 0 = no;
- **CREATININA_INGRESSO**: valori della creatinina in ingresso;
- **CREATININA_USCITA**: valori della creatinina in uscita.

Questi dati possono essere utilizzati per rispondere a diverse domande :

a. sapendo che i pazienti contenuti nel database hanno subito un infarto, è ragionevole supporre che la pressione sanguigna di tale popolazione sia diversa da quella fisiologica (80)?
b. le linee guida regionali per l'intervento di angioplastica indicano come soglia di 'accettabilità' del protocollo che l'intervento produca una effettiva riduzione dello slivellamento (a 1 ora) almeno nel 70% dei casi. In base al campione a disposizione, è possibile affermare che negli ospedali lombardi

l'intervento viene effettuato con un protocollo accettabile?

2.a TEST SULLA MEDIA

Per rispondere alla domanda dell'esercizio devo effettuare un test d'ipotesi: eseguo un test sulla media vera μ della pressione sanguigna dei pazienti affetti da infarto. In base alla richiesta dell'esercizio vorremo verificare:

$$H_0 : \mu = \mu_0 \qquad vs \qquad H_1 : \mu \neq \mu_0$$

dove la varianza della variabile che considero è incognita. La statistica test in questo caso è dunque:

$$T_0 = \frac{|\bar{X} - \mu_0|}{s/\sqrt{n}}.$$

dove μ_0 nel nostro caso è 80, mentre la regione critica del test bilatero di livello α è:

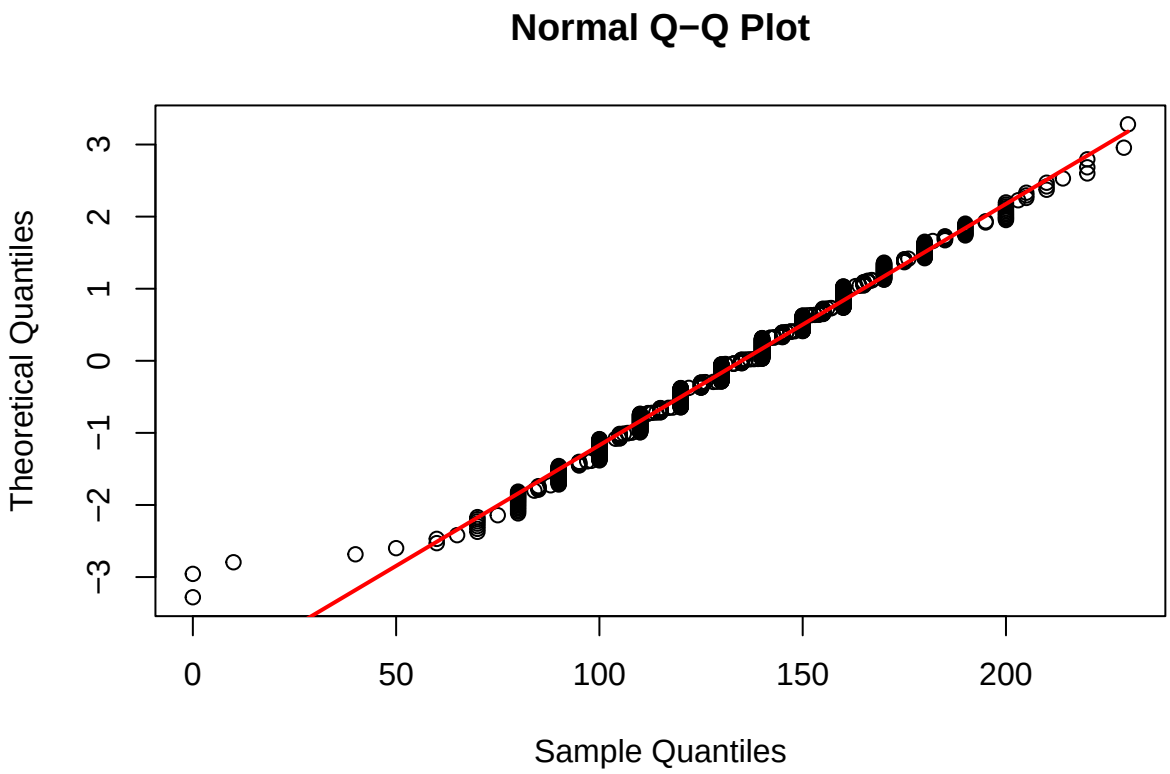
$$R_\alpha = \{T_0 > t_{(1-\alpha/2, n-1)}\}$$

Iniziamo, importando il dataset.

```
dati = read.table( "outcomes.txt", header = T )
head( dati )
##   PRESSIONE ST_RESOLUTION_70_60 CREATININA_INGRESSO CREATININA_USCITA
## 1       170                   1                1.10         0.6291668
## 2        90                  NA                1.26         2.5725009
## 3       150                   1                0.74         0.3895242
## 4       180                   1                0.77         1.7123947
## 5       160                   1                0.70         1.1919551
## 6       145                   1                2.03         2.4732782

dim( dati )
## [1] 963   4
# n è il numero di pazienti ( dimensione del campione )
n = dim( dati )[ 1 ]
names( dati )
## [1] "PRESSIONE"           "ST_RESOLUTION_70_60" "CREATININA_INGRESSO"
## [4] "CREATININA_USCITA"

attach( dati )
```

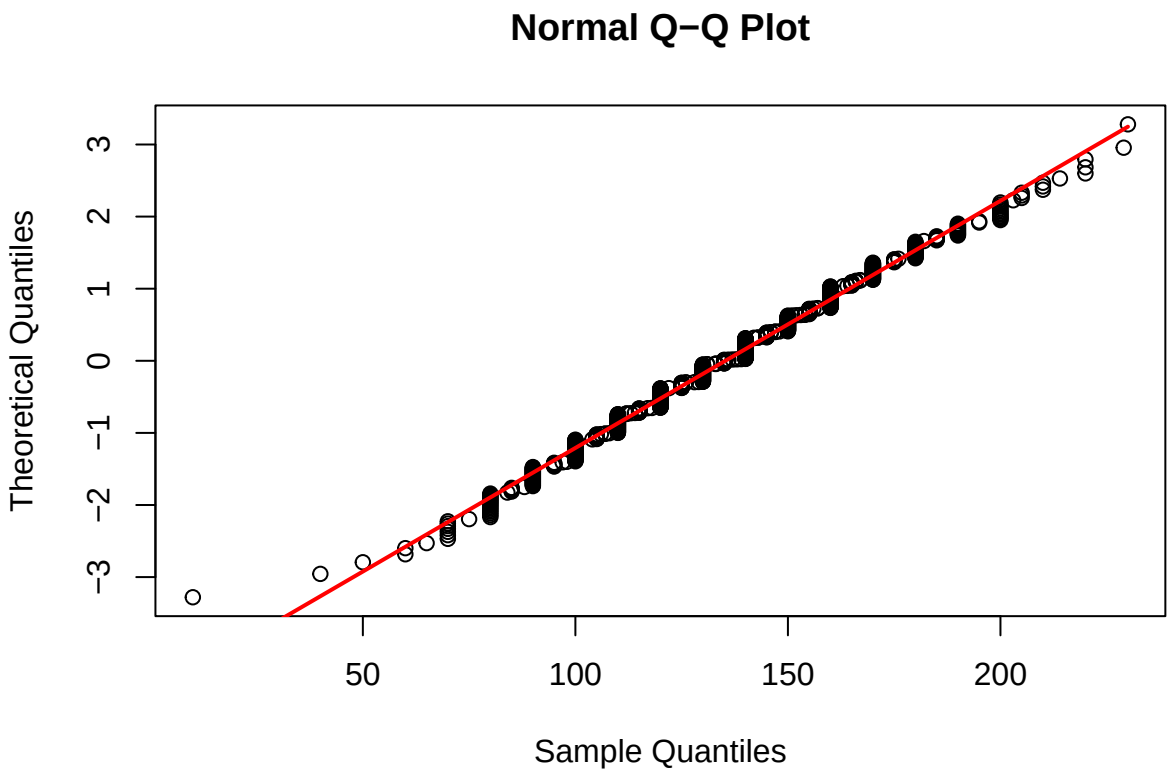
Primo passo: dal momento che la varianza è incognita devo verificare la normalità dei dati.

```
# la variabile che mi interessa è la pressione
n = sum( !is.na( PRESSIONE ) )
n
## [1] 963
# non ci sono dati mancanti!

#dev.new()
qqnorm( PRESSIONE, datax = T )
dati.ord = sort( PRESSIONE )
ranghi = 1: n
F.emp = ( ranghi - 0.5 ) / n
z_j = qnorm( F.emp )
y_j = lm( z_j ~ dati.ord )$fitted.values
lines( dati.ord, y_j, col = "red", lwd = 2 )
```



Ci sono due outlier negativi INVEROSIMILI, ovvero le osservazioni con pressione sanguigna nulla (=0). Rimuovo i due dati sospetti (cambiando anche la dimensione campionaria) e ricontrollo la normalità.

```
PRESSIONE = PRESSIONE[ which( PRESSIONE != 0 ) ]
n = sum( !is.na( PRESSIONE ) )
n
## [1] 961

# e rifaccio il qq-plot
#dev.new()
qqnorm( PRESSIONE, datax = T )
dati.ord = sort( PRESSIONE )
ranghi = 1: n
F.emp = ( ranghi - 0.5 ) / n
z_j = qnorm( F.emp )
y_j = lm( z_j ~ dati.ord )$fitted.values
lines( dati.ord, y_j, col = "red", lwd = 2 )
```



Nonostante la variabile in questione è discreta, ho una grande numerosità campionaria ed un buon adattamento alla distribuzione Normale: possiamo procedere.

Una volta verificata la normalità posso procedere con il test: fisso il livello $\alpha = 0.01$.

Calcolo media e dev. std campionarie.

```
alpha = 0.01

# stima puntuale di media e deviazione standard
media.camp = mean( PRESSIONE )
devstd.camp = sd( PRESSIONE )
```

Calcolo il quantile della corrispondente t-Student (ricordate che la varianza è incognita).

```
# quantile della corrispondente t-Student ( varianza incognita! )
t.alpha = qt( 1 - alpha / 2, n - 1 )
```

Calcolo la statistica test T_0.

```
# calcolo dunque la statistica test T.0
T.0 = abs( media.camp - 80 ) / ( devstd.camp / sqrt( n ) )
T.0
## [1] 58.95984
```

Il valore è molto grande. Qual è l'esito del test? La statistica test cade nella regione critica?

```
T.0 > t.alpha
## [1] TRUE
```

Al livello 1% ho evidenza per rifiutare H_0 ed affermare che la vera media della pressione sanguigna negli

infartati è diversa da 80. Visto però il valore così elevato della statistica test, e per essere maggiormente precisi, calcoliamo il p-value del test bilatero a varianza incognita:

$$p - value = 2 \cdot P(t > T.0)$$

```
pvalue = 2 * ( 1 - pt( T.0, n - 1 ) )
pvalue
## [1] 0
```

Come si può osservare, il p-value è circa 0, per cui l'evidenza per affermare che la media vera della pressione sia diversa da 80 è molto forte.

N.B. Esiste una funzione automatica di R che effettua il test t :

```
t.test( PRESSIONE, alternative = "two.sided", mu = 80, conf.level = 1 - alpha )
##
##  One Sample t-test
##
## data:  PRESSIONE
## t = 58.96, df = 960, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 80
## 99 percent confidence interval:
##  132.8531 137.6922
## sample estimates:
## mean of x
##  135.2726
```

Utilizziamo la funzione per verificare di avere effettuato il test nel modo corretto.

2.b TEST SULLA PROPORZIONE

Per rispondere alla domanda dell'esercizio devo effettuare un test d'ipotesi: eseguo un test sulla proporzione vera p, precentuale di casi in cui l'intervento avviene secondo protocollo (slivellamento ridotto). In particolare in base alla richiesta dell'esercizio vorremo verificare :

$$H_0 : p = 0.7 \qquad vs \qquad H_1 : p > 0.7$$

Ricordiamo che la statistica test in questo caso è:

$$Z_0 = \frac{\bar{p} - p_0}{\sqrt{p_0 \cdot (1 - p_0)/n}}$$

dove p_0 nel nostro caso è 0.7, mentre la regione critica del test unilatero di livello α è:

$$R_\alpha = \{Z_0 > z_{(1-\alpha)}\}$$

Fisso quindi $p_0 = 0.7$ e fisso il livello del test che voglio effettuare ad $\alpha = 0.05$ (poi per completezza calcolerò anche il p-value).

```
p.0 = 0.7
alpha = 0.05

# variabile che voglio considerare:  ST_RESOLUTION_70_60 do un nome più semplice alla variabile
ST = ST_RESOLUTION_70_60
n = sum( ! is.na( ST ) )
# dimensione effettiva del dataset ( escludo i missing! )
```

A questo punto, calcolo il quantile che mi serve da limite inferiore nella regione critica (test unilatero).

```
z.alpha = qnorm( 1 - alpha )
```

Calcolo la stima puntuale della proporzione di casi in cui il trattamento ha successo: conto quante volte vedo un successo (S) rispetto al totale.

```
p.camp = sum( ST, na.rm = TRUE ) / n
p.camp
## [1] 0.7721925
```

La stima puntuale di p è maggiore di 0.7. Già questo è un primo indizio a per il rifiuto di H_0 e per l'accettazione di H_1. Calcolo la statistica test per vedere se è effettivamente così:

```
Z.0 = ( p.camp - p.0 ) / sqrt( p.0 * ( 1 - p.0 ) / n )
Z.0
## [1] 4.817129
```

Qual è l'esito del test? La statistica test cade nella regione critica?

```
Z.0 > z.alpha
## [1] TRUE
```

I dati forniscono quindi evidenza sufficiente per rifiutare l'ipotesi nulla, ed affermare che il protocollo negli ospedali lombardi ha succcesso almeno nel 70% dei casi, ed è dunque accettabile.

La conclusione che abbiamo tratto è forte? Quanto dipende dal livello scelto (α)? Calcoliamo il p-value:

```
pvalue = 1 - pnorm( Z.0 )
pvalue
## [1] 7.281909e-07
```

Il p-value è molto basso, dunque abbiamo forte evidenza per affermare che H_1 è vera.

N.B. Come per il test t, esiste una funzione automatica di R che effettua il test per la proporzione:

```
counts = sum( ST, na.rm = TRUE )
prop.test( counts, n, p = 0.7, alternative = "greater", conf.level = 1 - alpha, correct = FALSE )
##
##  1-sample proportions test without continuity correction
##
## data:  counts out of n, null probability 0.7
## X-squared = 23.205, df = 1, p-value = 7.282e-07
## alternative hypothesis: true p is greater than 0.7
## 95 percent confidence interval:
##  0.7488645 1.0000000
## sample estimates:
##         p
## 0.7721925

detach( dati )
```

Utilizziamo la funzione per verificare di avere effettuato il test nel modo corretto.

Attenzione! Questa funzione è scritta in modo tale da effettuare una correzione per migliorare le prestazioni del test, mentre noi abbiamo eseguito un test classico. Per poter confrontare i risultati, dobbiamo assegnare 'FALSE' all'argomento 'correct'. 'correct' fa riferimento alla correzione di continuità di Yates.

Esercizio 3: inferenza sulla media di una popolazione gaussiana.

Carichiamo i dati contenuti nel file *temperatura.txt*: Il file contiene 130 osservazioni di 3 variabili :

- **Temperatura**: si riferisce alla temperatura corporea (gradi Fahrenheit),
- **Sesso**: si riferisce al sesso del paziente (U = uomo, D = donna);
- **Freq_cardiaca**: si riferisce alla frequenza cardiaca (battiti al minuto).

I dati provengono da un articolo pubblicato sul 'Journal of the American Medical Association' che studia se la vera temperatura media del corpo umano è pari a 98.6 gradi Fahrenheit ($\sim$ 37 gradi centigradi).

[Mackowiak, P. A., Wasserman, S. S., and Levine, M. M. (1992), 'A Critical Appraisal of 98.6 Degrees F, the Upper Limit of the Normal Body Temperature, and Other Legacies of Carl Reinhold August Wunderlich', Journal of the American Medical Association, 268, 1578-1580.]

Le due principali questioni a cui si vuole dare risposta nello studio da cui i dati provengono sono:

a. stabilire se la media reale della temperatura corporea della popolazione sia 98.6 gradi F;
b. stabilire se ci sono differenze nella temperatura corporea dovute al sesso del soggetto, e in particolare se la temperatura corporea delle donne è più alta di quella degli uomini.

3.a Rispondiamo calcolando un intervallo di confidenza per la media, ed effettuando un test d'ipotesi sulla media della popolazione.

Importiamo i dati e concentriamoci sulla variabile Temperatura.

```
# importazione del dataset
dati = read.table( "temperatura.txt", header = T )
head( dati )
##   Temperatura Sesso Freq_cardiaca
## 1        96.3     U            70
## 2        96.7     U            71
## 3        96.9     U            74
## 4        97.0     U            80
## 5        97.1     U            73
## 6        97.1     U            75

dim( dati )
## [1] 130   3
n = dim( dati )[ 1 ]  # n è il numero di pazienti ( dimensione del campione )
names( dati )
## [1] "Temperatura"   "Sesso"         "Freq_cardiaca"

attach( dati )
```

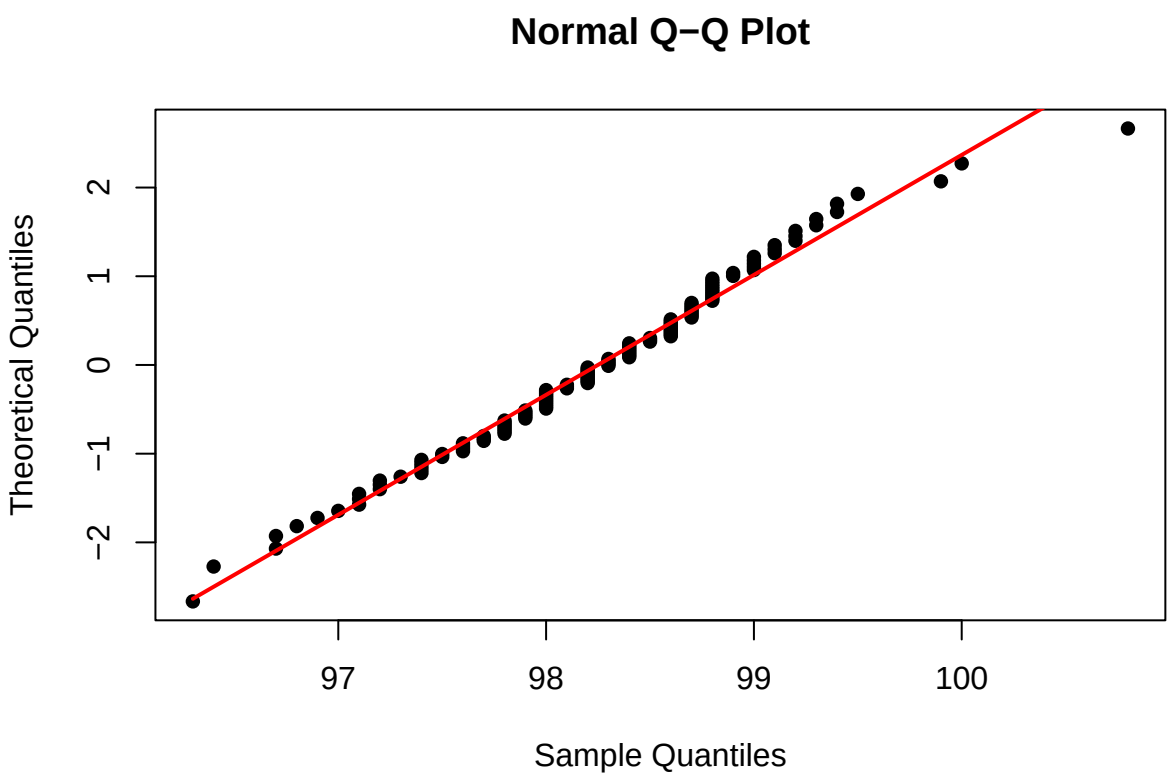
Primo passo: calcolo la stima puntuale della media della popolazione.

```
media.camp = mean( Temperatura )
media.camp
## [1] 98.24923
```

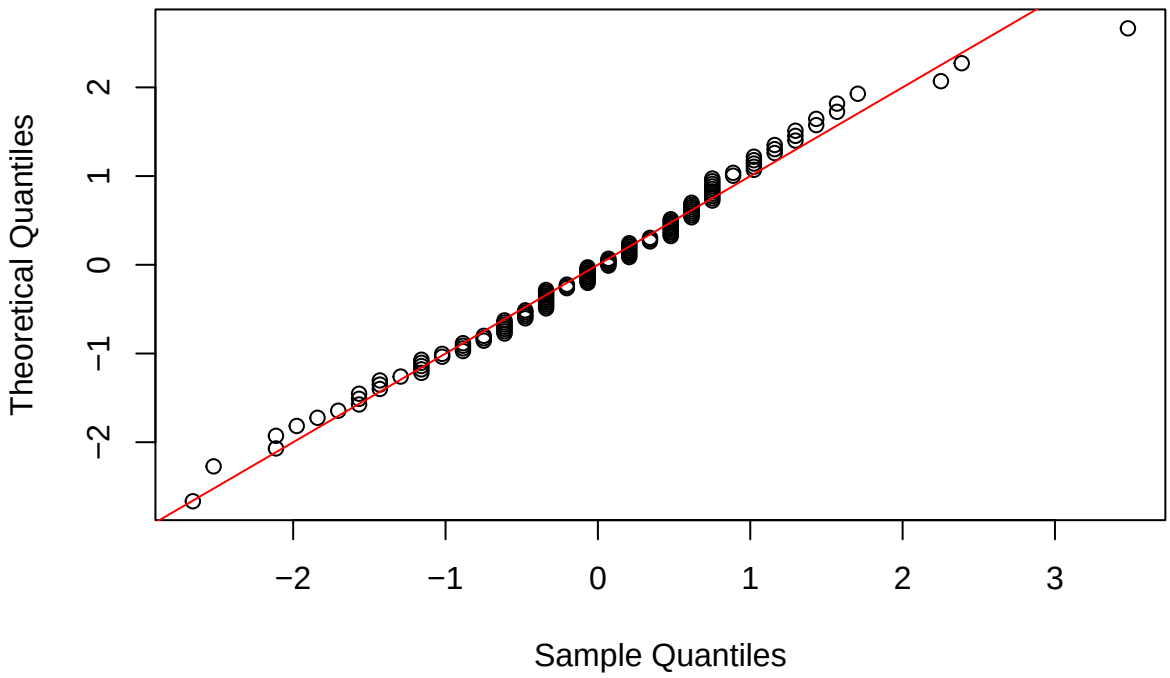
È molto vicina alla media vera ipotizzata nello studio.

Valutiamo prima di procedere la Normalità dei dati, dal momento che vorremo sia calcolare un intervallo di confidenza basato sulla distribuzione t di Student, sia effettuare un test sulla media in ipotesi di Normalità.

```
#dev.new()
qqnorm( Temperatura, datax = T, pch = 16 )
temp.ord = sort( Temperatura )
ranghi = 1: n
F.emp = ( ranghi - 0.5 ) / n
z_j = qnorm( F.emp )
y_j = lm( z_j ~ temp.ord )$fitted.values
lines( temp.ord, y_j, col = "red", lwd = 2 )
```




```
#modo alternativo
qqplot( ( temp.ord - mean( temp.ord ) )/sd( temp.ord ), z_j,
        xlab = 'Sample Quantiles', ylab = 'Theoretical Quantiles' )
abline( 0, 1, col='red' )
```



Buon adattamento dei dati alla distribuzione Normale: possiamo procedere!

Eseguiamo quindi un test di INFERENZA SULLA MEDIA DI UNA POPOLAZIONE GAUSSIANA, A VARIANZA INCOGNITA.

Calcolo un intervallo di confidenza per la media di livello 95%.

```
alpha = 0.05
devstd.camp = sd( Temperatura )

t.alpha = qt( 1 - alpha / 2, n - 1 )
IC.alpha = c( media.camp - t.alpha * devstd.camp / sqrt( n ),
              media.camp + t.alpha * devstd.camp / sqrt( n ) )
IC.alpha
## [1] 98.12200 98.37646
```

REMARK Il valore della media vera ipotizzato nello studio non è contenuto nell'intervallo: già questo mi sta dando informazioni sul test d'ipotesi che voglio fare. Quali?

Effettuo ora un test per verificare l'ipotesi:

$$H_0 : \mu = 98.6F \qquad vs \qquad H_1 : \mu \neq 98.6F$$

La regione critica del test bilatero di livello α è:

$$R_\alpha = \left\{ \frac{|\bar{X} - 98.6|}{s/\sqrt{n}} > t_{(1-\alpha/2)(n-1)} \right\}$$

Il quantile della t-Student l'abbiamo appena calcolato. Calcolo dunque la statistica test T_0:

```r
T.0 = abs( media.camp - 98.6 ) / ( devstd.camp / sqrt( n ) )
T.0
## [1] 5.454823
```

Qual è l'esito del test? La statistica test cade nella regione critica?

```r
T.0 > t.alpha
## [1] TRUE
```

Al livello 5% ho evidenza per rifiutare H_0 ed affermare che la vera media della popolazione è diversa da 98.6 F.

Per essere maggiormente precisi, calcoliamo il p-value del test test bilatero a varianza incognita:

$$p - value = 2 \cdot P(t > T_0)$$

```r
p = 2 * ( 1 - pt( T.0, n - 1 ) )
p
## [1] 2.410632e-07
```

Come si può osservare, il p-value è circa 0, per cui ho forte evidenza per affermare che la media vera sia diversa da 98.6.

3.b

(INFERENZA SULLA DIFFERENZA TRA LE MEDIE DI DUE POPOLAZIONI)

Rispondiamo a questa domanda calcolando un intervallo di confidenza, ed effettuando un test d'ipotesi per la differenza tra le medie delle temperature corporee nelle due sottopopolazioni individuate dal sesso. Consideriamo innanzitutto i due campioni distinti per sesso:

```r
names( dati )
## [1] "Temperatura"   "Sesso"         "Freq_cardiaca"

temp.m = Temperatura[ which( Sesso == "U" ) ]
temp.f = Temperatura[ which( Sesso == "D" ) ]
length( temp.m )
## [1] 65
length( temp.f )
## [1] 65
# ora ho due campioni di ampiezza dimezzata!
n = length( temp.m )
```

Primo passo: calcolo la stima puntuale della temperatura media corporea maschile e femminile.

```r
media.m = mean( temp.m )
media.m
## [1] 98.10462
media.f = mean( temp.f )
media.f
## [1] 98.39385
```

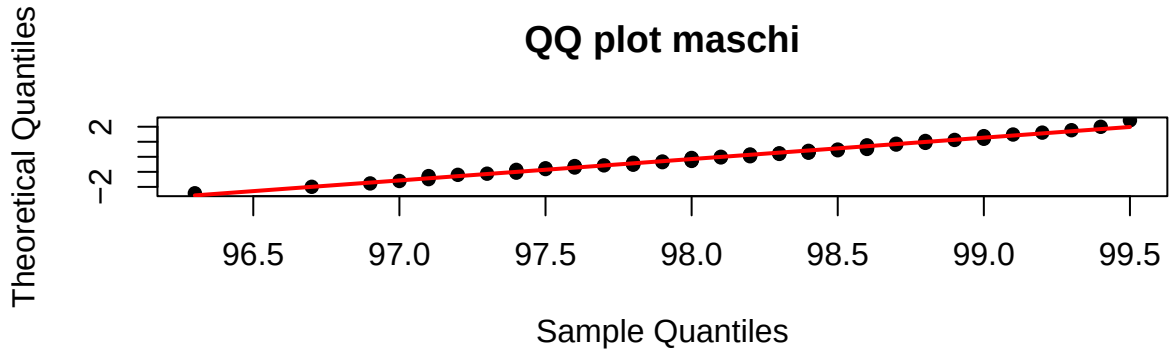
La temperatura nelle donne sembra mediamente più alta.

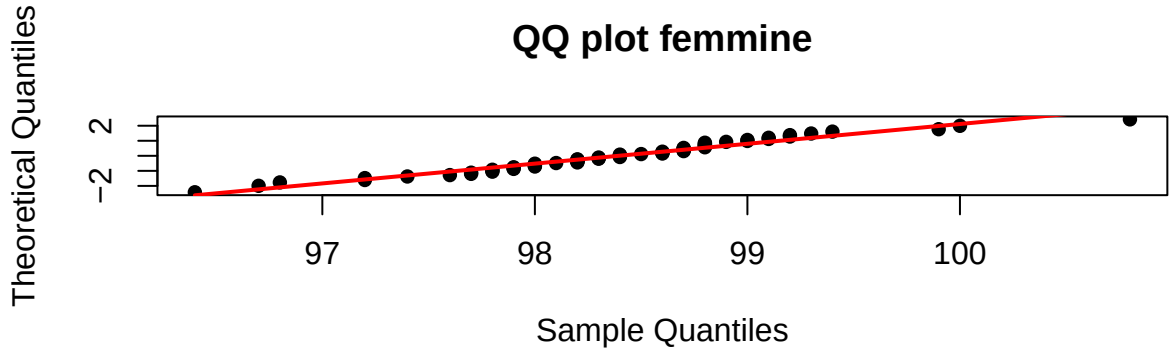
Tramite questi due valori posso calcolare subito la stima puntuale della differenza tra le medie delle temperature corporee maschile e femminile.

```r
diff.camp = media.f - media.m
diff.camp
## [1] 0.2892308
```

Come prima, prima di procedere, valutiamo la Normalità dei dati (separatamente per i due gruppi).

```r
#dev.new()
par( mfrow = c( 2, 1 ) )

qqnorm( temp.m, datax = T, main = "QQ plot maschi", pch = 16 )
temp.ord = sort( temp.m )
ranghi = 1: n
F.emp = ( ranghi - 0.5 ) / n
z_j = qnorm( F.emp )
y_j = lm( z_j ~ temp.ord )$fitted.values
lines( temp.ord, y_j, col = "red", lwd = 2 )

qqnorm( temp.f, datax = T, main = "QQ plot femmine", pch = 16 )
temp.ord = sort( temp.f )
ranghi = 1: n
F.emp = ( ranghi - 0.5 ) / n
z_j = qnorm( F.emp )
y_j = lm( z_j ~ temp.ord )$fitted.values
lines( temp.ord, y_j, col = "red", lwd = 2 )
```





Buon adattamento dei dati alla distribuzione Normale: possiamo procedere.

Calcolo un intervallo di confidenza per la differenza tra le medie di livello 95%.

IPOTESI: le varianze teoriche (incognite) della temperatura nelle due sottopopolazioni sono uguali.

Non conoscendo le vere distribuzioni nè tantomeno le vere varianze delle due popolazioni, quello che posso fare un è *test sul confronto tra le varianze nei due gruppi* per verificare se questa ipotesi è realistica per il nostro dataset.

Calcolo la stima puntuale della deviazione standard:

```
sd( temp.m )
## [1] 0.6987558
sd( temp.f )
## [1] 0.7434878
```

Eseguiamo il test bilatero sulle varianze. Ricordiamo che il rapporto tra due stimatori della varianza di popolazioni gaussiane ha una distribuzione che può essere approssimata dalla distribuzione di Fisher (con parametri $n_X - 1$ e $n_Y - 1$, dove n_X e n_Y sono le numerosità delle sue popolazioni rispettivamente.

```
f.0 = var( temp.m ) / var( temp.f )
alpha = 0.05
f.alpha1 = qf( alpha / 2, n - 1, n - 1 )
f.alpha2 = qf( 1 - alpha / 2, n - 1, n - 1 )

f.0 < f.alpha1 | f.0 > f.alpha2
## [1] FALSE
```

Con un livello di significatività del 5%, non ho evidenza per pensare che le due varianze siano diverse (accetto H_0). Calcolo anche il p-value:

```
p = 2 * min( pf( f.0, n - 1, n - 1 ), 1 - pf( f.0, n - 1, n - 1 ) )
p
## [1] 0.6210837
```

REMARK Come per il test bilatero sulla varianza del singolo campione, visto che la distribuzione F è asimmetrica, per trovare il p-value devo prendere 2 volte il minimo tra la coda destra e la coda sinistra della distribuzione F in corrispondenza della statistica test F_0.

Come per il test t e il test z, la stessa cosa si può fare in automatico con una funzione R:

```
var.test( temp.m, temp.f, alternative = "two.sided" )
##
##  F test to compare two variances
##
## data:  temp.m and temp.f
## F = 0.88329, num df = 64, denom df = 64, p-value = 0.6211
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5387604 1.4481404
## sample estimates:
## ratio of variances
##          0.8832897
```

Posso quindi procedere nell'ipotesi di varianze uguali. Calcolo la deviazione standard campionaria pooled, che tiene conto di entrambi i campioni. Ricordiamo che in generale vale:

$$S_p = \frac{(n_X - 1)S_X^2 + (n_Y - 1)S_Y^2}{n_X + n_Y - 2}$$

```
s.pooled = sqrt( ( ( n - 1 ) * sd( temp.m )^2 + ( n - 1 ) * sd( temp.f )^2 ) / ( 2 * n - 2 ) )
s.pooled
## [1] 0.7214685
```

Calcolo infine l'IC con confidenza 95% grazie agli stimatori che ho calcolato finora:

```
alpha = 0.05
t.alpha = qt( 1 - alpha / 2, 2 * n - 2 )

# realizzazione dell'IC per la differenza tra le medie
IC.alpha = c( diff.camp - t.alpha * s.pooled * sqrt( 2 / n ),
              diff.camp + t.alpha * s.pooled * sqrt( 2 / n ) )
IC.alpha
## [1] 0.03882216 0.53963938
```

L'IC non contiene lo 0. Quindi pare che esista una differenza significativa nella temperatura corporea tra uomini e donne. Inoltre l'intervallo contiene solo valori positivi, dunque la temperatura delle donne sembra effettivamente superiore a quella degli uomini. Verifichiamolo con un test unilatero.

Effettuo ora un test per verificare l'ipotesi:

$$H_0 : \mu_f \leq \mu_m \qquad vs \qquad H_1 : \mu_f > \mu_m$$

La regione critica del test unilatero per due campioni di livello α è;

$$R_\alpha = \left\{ \frac{(\bar{x}_f - \bar{x}_m) - 0}{S_p \cdot \sqrt{2/n}} > t_{(1-\alpha)(2*n-2)} \right\}$$

dove come prima consideriamo la differenza delle medie campionarie:

```
diff.camp = media.f - media.m
```

Calcoliamo il quantile della t-Student:

```
alpha = 0.05
t.alpha = qt( 1 - alpha, 2 * n - 2 )
```

Dunque la statistica test T_0 risulta:

```
T.0 = diff.camp / ( s.pooled * sqrt( 2 / n ) )
T.0
## [1] 2.285435
```

Qual è l'esito del test? La statistica test cade nella regione critica?

```
T.0 > t.alpha
## [1] TRUE
```

Al livello 5% ho evidenza per rifiutare H_0 ed affermare che esiste una differenza nella media della temperatura corporea nelle due sottopopolazioni. Per essere maggiormente precisi, calcoliamo il p-value del test unilatero a varianza incognita: $ p-value = P(t > T_0)$

```
pvalue = ( 1 - pt( T.0, 2 * n - 2 ) )
pvalue
## [1] 0.01196594
```

Il p-value è basso a sufficienza per rifiutare l'ipotesi al livello 5%, ma osserviamo che $p - value > 0.01$. Poichè il p-value rappresenta il più piccolo valore di α per il quale ho evidenza di poter accettare l'ipotesi nulla, allora al livello 1% non avrei rifiutato H_0.

REMARK Posso usare la funzione t.test anche per fare un test di confronto tra due gruppi :

```
t.test( temp.f, temp.m, alternative = "greater", mu = 0, var.equal = TRUE, conf.level = 1 - alpha )
##
##  Two Sample t-test
##
## data:  temp.f and temp.m
## t = 2.2854, df = 128, p-value = 0.01197
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.07955046        Inf
## sample estimates:
## mean of x mean of y
##  98.39385  98.10462

detach( dati )
```

Utilizziamo la funzione per verificare di avere effettuato il test nel modo corretto.

Esercizio 4: test per dati accoppiati.

Consideriamo ancora i dati contenuti nel file *outcomes.txt* e consideriamo le variabili:

- **CREATININA_INGRESSO**: valori della creatinina in ingresso (pre-ricovero);
- **CREATININA_USCITA**: valori della creatinina in uscita (post-infarto).

La misura della concentrazione di creatinina nel plasma è un indicatore della funzione renale, e in particolare un suo aumento è un possibile indice di danno renale; secondo una teoria non ancora accettata dalla comunità scientifica internazionale, le disfunzioni ai reni sono una delle possibili complicanze dell'infarto. Un professore del Policlinico di Milano sta conducendo uno studio sulle complicanze dell'infarto, e vuole dunque utilizzare il campione a disposizione per dimostrare la tesi che nei pazienti infartati si osservi un innalzamento nella concentrazione di creatinina nel plasma.

Importiamo i dati:

```
dati = read.table( "outcomes.txt", header = T )
head( dati )
##   PRESSIONE ST_RESOLUTION_70_60 CREATININA_INGRESSO CREATININA_USCITA
## 1       170                   1                1.10         0.6291668
## 2        90                  NA                1.26         2.5725009
## 3       150                   1                0.74         0.3895242
## 4       180                   1                0.77         1.7123947
## 5       160                   1                0.70         1.1919551
## 6       145                   1                2.03         2.4732782

dim( dati )
## [1] 963   4
n = dim( dati )[ 1 ]  # n è il numero di pazienti ( dimensione del campione )
names( dati )
## [1] "PRESSIONE"           "ST_RESOLUTION_70_60" "CREATININA_INGRESSO"
## [4] "CREATININA_USCITA"

attach( dati )
## Il seguente oggetto è mascherato _da_ .GlobalEnv:
##
##     PRESSIONE
```

Per rispondere alla questione posta dall'esercizio devo provare che il livello di creatinina è significativamente più alto al momento della dimissione rispetto al ricovero. Dal momento che le misurazioni che ho a disposizione riguardano gli stessi pazienti, prima e dopo l'infarto, i due gruppi non sono indipendenti ma accoppiati; considero dunque le differenze.

Creo la nuova variabile differenza:

```
DIFF = CREATININA_USCITA - CREATININA_INGRESSO
```

In questo modo, per provare le complicanze dell'angioplastica, dovrei provare che la media delle differenze è positiva. Per rispondere alla domanda dell'esercizio devo effettuare un test d'ipotesi: eseguo un test per dati accoppiati sulla differenza media (μ_d) tra la creatinina post infarto e quella pre infarto. In base alla richiesta dell'esercizio vorremo verificare:

$$H_0 : \mu_d \leq 0 \qquad vs \qquad H_1 : \mu_d > 0$$

dove la varianza della variabile che considero è incognita. La statistica test in questo caso è dunque:

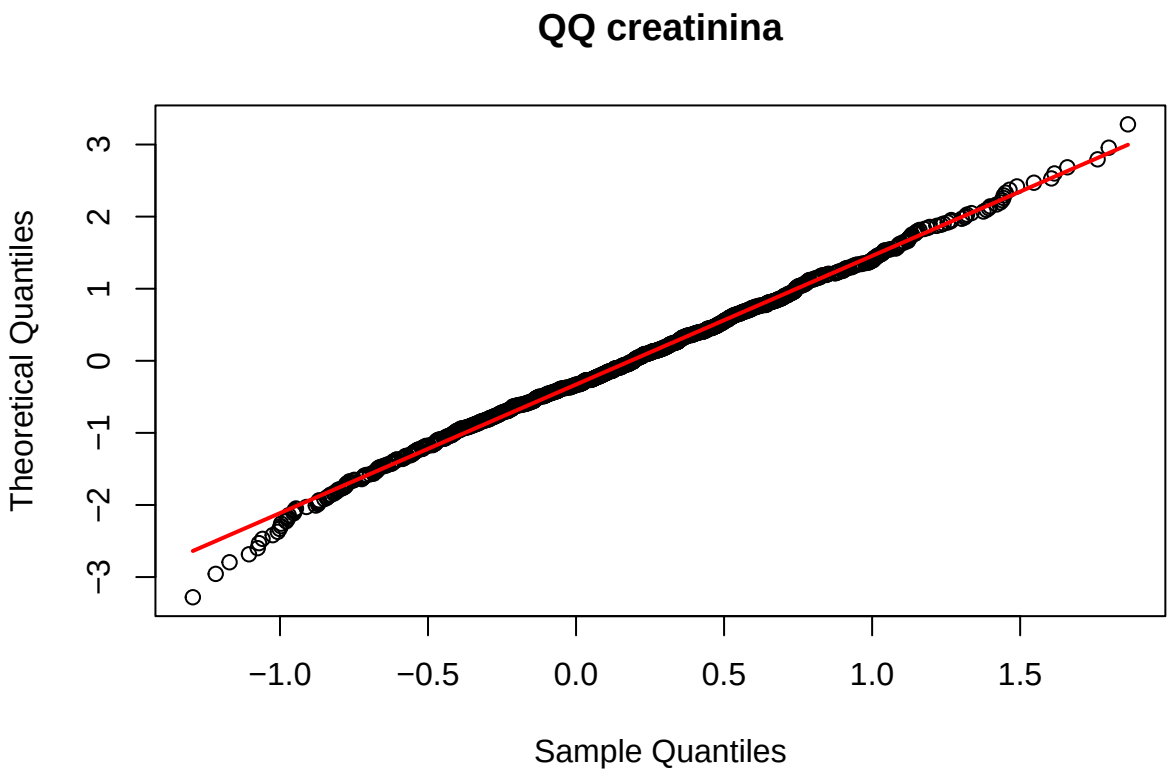
$$T_0 = \frac{\bar{D} - 0}{s/\sqrt{n}}$$

mentre la regione critica del test unilatero di livello α è:

$$R_\alpha = \{T_0 > t_{(1-\alpha,n-1)}\}.$$

La variabile che mi interessa è ora la differenza nei valori di creatinina. Devo verificare la normalità delle differenze per poter procedere con il test.

```
n = sum( !is.na( DIFF ) )
n
## [1] 963

#dev.new()
qqnorm( DIFF, datax = T, main = "QQ creatinina" )
diff.ord = sort( DIFF )
ranghi = 1: n
F.emp = ( ranghi - 0.5 ) / n
z_j = qnorm( F.emp )
y_j = lm( z_j ~ diff.ord )$fitted.values
lines( diff.ord, y_j, col = "red", lwd = 2 )
```



I dati si adattano perfettamente alla distribuzione normale, quindi posso procedere col test.

```
# stima puntuale della media e della varianza
media.diff = mean( DIFF )
media.diff  # la stima della media è superiore a 0...
## [1] 0.1843498
dev.stand.diff = sd( DIFF )
dev.stand.diff
## [1] 0.5596303

# eseguo il test!  quantile della t-Student :
alpha = 0.01
t.alpha = qt( 1 - alpha, n - 1 )

# statistica test T.0 :
T.0 = media.diff / ( dev.stand.diff / sqrt( n ) )
T.0
## [1] 10.22244

# qual è l'esito del test? la statistica test cade nella regione critica?
T.0 > t.alpha
## [1] TRUE
```

Al livello 1% ho evidenza per rifiutare H_0 ed affermare che esiste una differenza nella media della creatinina prima e dopo l'intervento. Per essere maggiormente precisi, calcoliamo il p-value del test unilatero a varianza incognita per dati accoppiati: $p - value = P(t > T_0)$.

```
pvalue = ( 1 - pt( T.0, n - 1 ) )
pvalue
## [1] 0
```

Il p-value è praticamente zero.

N.B. Posso usare la funzione t.test anche per fare un test per dati accoppiati: basta impostare l'argomento 'paired' a 'TRUE'.

```
t.test( CREATININA_USCITA, CREATININA_INGRESSO, alternative = "greater",
        mu = 0, paired = TRUE, conf.level = 1 - alpha )
##
##  Paired t-test
##
## data:  CREATININA_USCITA and CREATININA_INGRESSO
## t = 10.222, df = 962, p-value < 2.2e-16
## alternative hypothesis: true mean difference is greater than 0
## 99 percent confidence interval:
##  0.1423268       Inf
## sample estimates:
## mean difference
##       0.1843498

detach( dati )
```

Utilizziamo la funzione per verificare di avere effettuato il test nel modo corretto.