Hoja 4 (a): Inferencia de una muestra con R

Estadística Computacional I. Grado en Estadística

Departamento de Estadística e Investigación Operativa. Universidad de Sevilla

Introducción

Estudiar la Normalidad

$$H_0 \leadsto Normal$$

```
$$
H_0 \leadsto Normal
$$
```

Test de Shapiro-Wilk

```r
set.seed(12345)
x = rnorm(15,0,1)
x
```

```
##  [1]  0.5855288  0.7094660 -0.1093033 -0.4534972  0.6058875 -1.8179560
##  [7]  0.6300986 -0.2761841 -0.2841597 -0.9193220 -0.1162478  1.8173120
## [13]  0.3706279  0.5202165 -0.7505320
```

```r
shapiro.test(x)
```

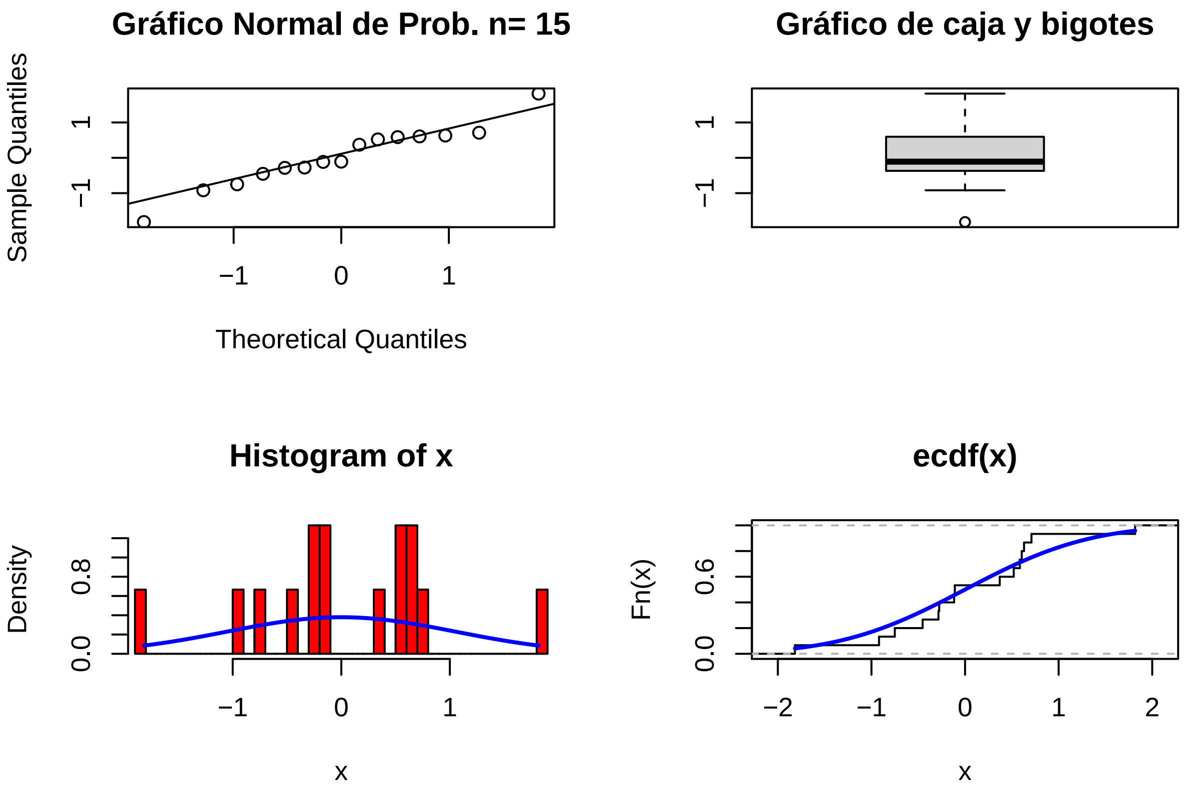
```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.96152, p-value = 0.7189
```

Estudiar gráficamente la Normalidad

```r
## Análisis de la normalidad una muestra
ananor<-function(x)
{
        par(mfrow=c(2,2))
        n<-length(x)
        qqnorm(x,main=paste("Gráfico Normal de Prob. n=",n))
        qqline(x)
        boxplot(x,main="Gráfico de caja y bigotes")
        hist(x,col="red",xlim=c(min(x),max(x)),br=30,freq=FALSE)
        curve(dnorm(x,mean(x),sd(x)),min(x),max(x),1000,col="blue",
              add=TRUE,lwd=2)
        plot(ecdf(x), do.points=FALSE, verticals=TRUE)
        curve(pnorm(x,mean(x),sd(x)),min(x),max(x),1000,col="blue",
```

```
                add=TRUE,lwd=2)
        par(mfrow=c(1,1))
        shapiro.test(x)
}
```

```
ananor(x)
```




```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.96152, p-value = 0.7189
```

```
library(tidyverse)
```

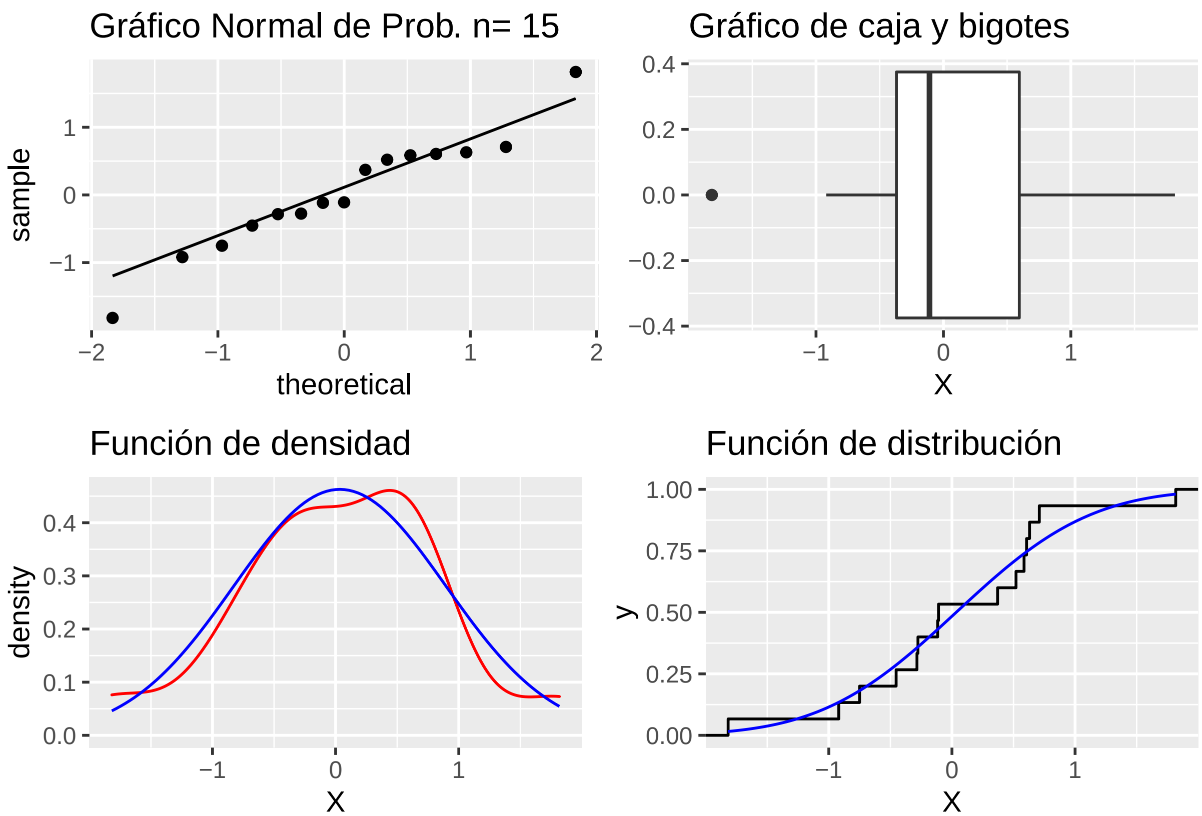
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(patchwork)

ananor_tidy<-function(x)
{
        n<-length(x)
```

```
        datos = data.frame(X = x)
        p1 = ggplot(datos, aes(sample = X)) +
                geom_qq() +
                geom_qq_line() +
                labs(
                        title = paste("Gráfico Normal de Prob. n=",n)
                )

        p2 = ggplot(datos, aes(x = X)) +
                geom_boxplot() +
                labs(
                        title = "Gráfico de caja y bigotes"
                )

        p3 = ggplot(datos, aes(x = X)) +
                #geom_histogram(aes(y = ..density..), fill = "red",col="black", bins = 30) +
                geom_density(col="red") +
                xlim(min(datos$X),max(datos$X)) +
                stat_function(aes(x=seq(min(X),max(X),length = length(X))),
                              fun = dnorm, args = list(mean = mean(datos$X),
                                                       sd = sd(datos$X)),color
                              = "blue") +
                labs(
                        title = "Función de densidad"
                )

        p4 = ggplot(datos, aes(x = X)) +
                stat_ecdf(geom = "step") +
                xlim(min(datos$X),max(datos$X)) +
                stat_function(aes(x=seq(min(X),max(X),length = length(X))),
                              fun = pnorm, args = list(mean = mean(datos$X),
                                                       sd = sd(datos$X)),color
                              = "blue") +
                labs(
                        title = "Función de distribución"
                )

        print(shapiro.test(x))

        (p1 | p2) / (p3 | p4)


}
```

```
ananor_tidy(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.96152, p-value = 0.7189
```



Test de Normalidad en el paquete fBasics

```
library(fBasics)
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
ksnormTest(x)
```

```
##
## Title:
##  One-sample Kolmogorov-Smirnov test
##
## Test Results:
##   STATISTIC:
##     D: 0.1724
##   P VALUE:
##     Alternative Two-Sided: 0.7025
##     Alternative      Less: 0.5779
##     Alternative   Greater: 0.3683
##
## Description:
##  Fri Apr  9 13:39:01 2021 by user:
```

```
fBasics::shapiroTest(x)
```

```
##
## Title:
##  Shapiro - Wilk Normality Test
```

```
##
## Test Results:
##   STATISTIC:
##     W: 0.9615
##   P VALUE:
##     0.7189
##
## Description:
##  Fri Apr  9 13:39:01 2021 by user:
```

```
jarqueberaTest(x)
```

```
##
## Title:
##  Jarque - Bera Normalality Test
##
## Test Results:
##   STATISTIC:
##     X-squared: 0.1005
##   P VALUE:
##     Asymptotic p Value: 0.951
##
## Description:
##  Fri Apr  9 13:39:01 2021 by user:
```

Test de Normalidad en el paquete nortest

```
library(nortest)
ad.test(x) # Anderson Darling
```

```
##
##  Anderson-Darling normality test
##
## data:  x
## A = 0.33942, p-value = 0.4491
```

```
cvm.test(x)
```

```
##
##  Cramer-von Mises normality test
##
## data:  x
## W = 0.053087, p-value = 0.4423
```

```
lillie.test(x)
```

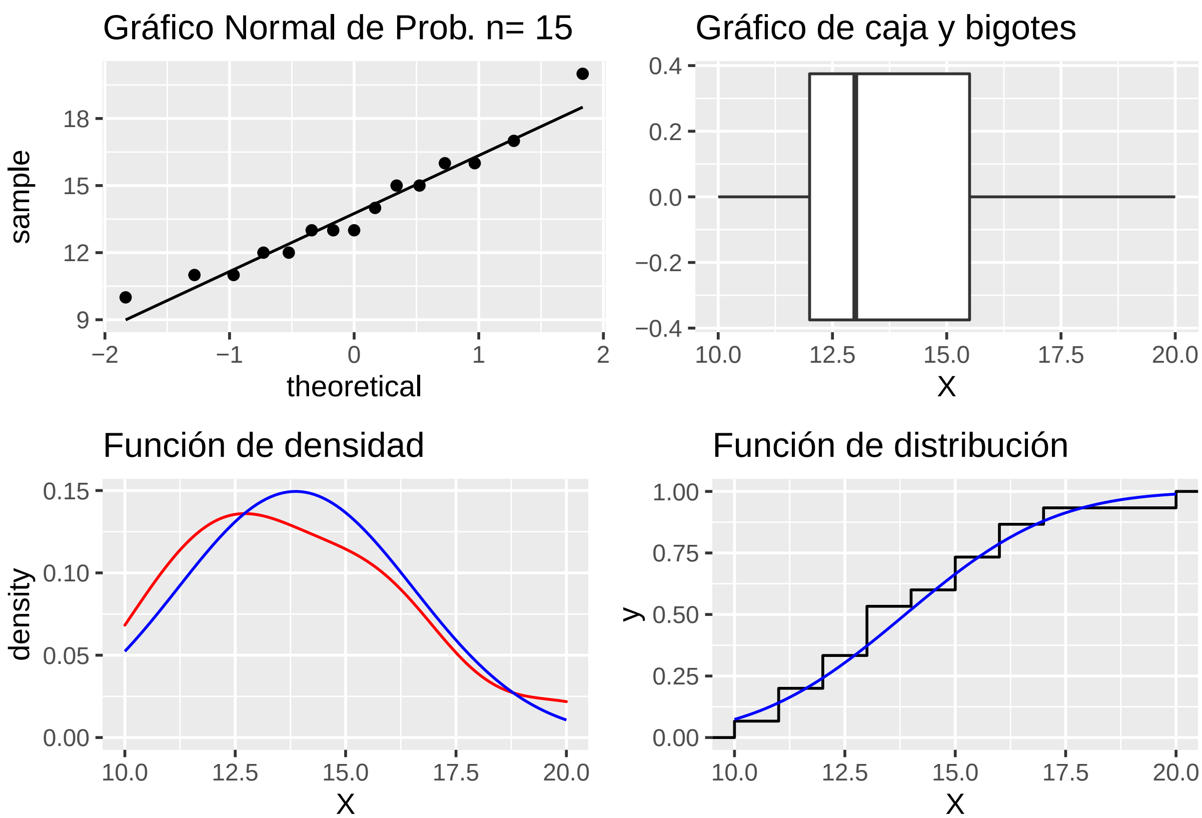
```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  x
## D = 0.15004, p-value = 0.4832
```

Ejemplo: Intervalo de Confianza y Hipótesis EX = 15

```
x = c(17,12,15,16,15,11,12,13,20,16,14,13,11,10,13)
#length(x)
```

```
ananor_tidy(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.95469, p-value = 0.6011
```




```
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   12.00   13.00   13.87   15.50   20.00
```

Se pueden utilizar test paramétricos

$$
\begin{array}{l}
H_0 : \mu = 15 \\
H_a : \mu \neq 15
\end{array}
$$

```
$$
\begin{array}{l}
H_0: \mu = 15 \\
H_a: \mu \not= 15
\end{array}
$$
```

```
t.test(x,mu = 15,conf.level = 0.99)
```

```
##
##  One Sample t-test
##
```

```
## data:  x
## t = -1.6446, df = 14, p-value = 0.1223
## alternative hypothesis: true mean is not equal to 15
## 99 percent confidence interval:
##  11.81519 15.91814
## sample estimates:
## mean of x
##  13.86667
```

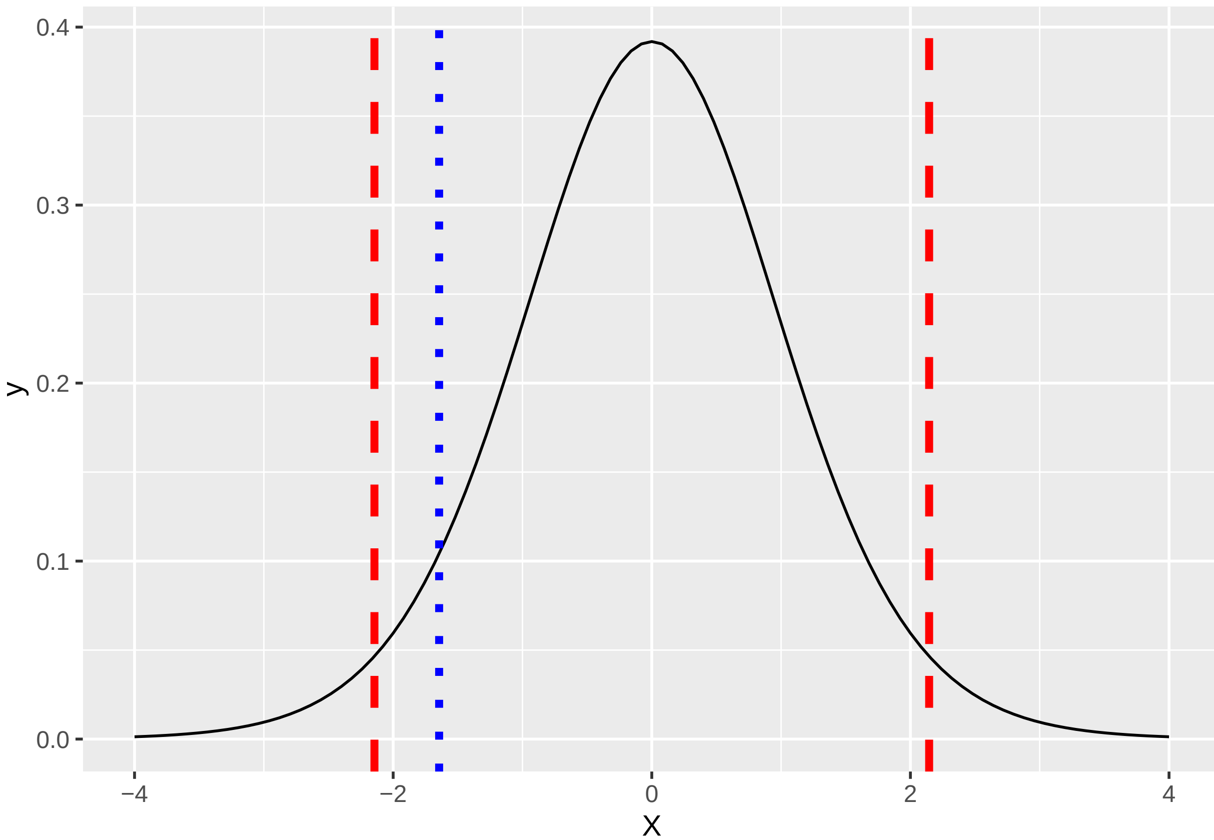
```
# alternative = "greater"
# alternative = "less"
```

Ejercicio 1

Dibujar la densidad de la t-Student bajo H0, los cuantiles que definen los puntos críticos y el valor del estadístico

```
func_g_dt_test = function(x, mu_ = 15) {
#mu_ = 15
datos = data.frame(X = x)
resul = t.test(datos$X,mu = mu_ )
#str(resul)
ggplot(datos,aes(x = X)) +
  stat_function(aes(x = seq(-4,4,length = length(X))),
                fun = dt, args = list(df = length(datos$X)-1),
                color = "black") +
  geom_vline(aes(xintercept = resul$statistic),color = "blue",
             linetype = "dotted",size = 1.4) +
  geom_vline(aes(xintercept = qt(0.025,df = resul$parameter)),color = "red",
             linetype = "dashed",size = 1.4) +
  geom_vline(aes(xintercept = qt(0.975,df = resul$parameter)),color = "red",
             linetype = "dashed",size = 1.4)
}
```

```
func_g_dt_test(x, mu_ = 15)
```



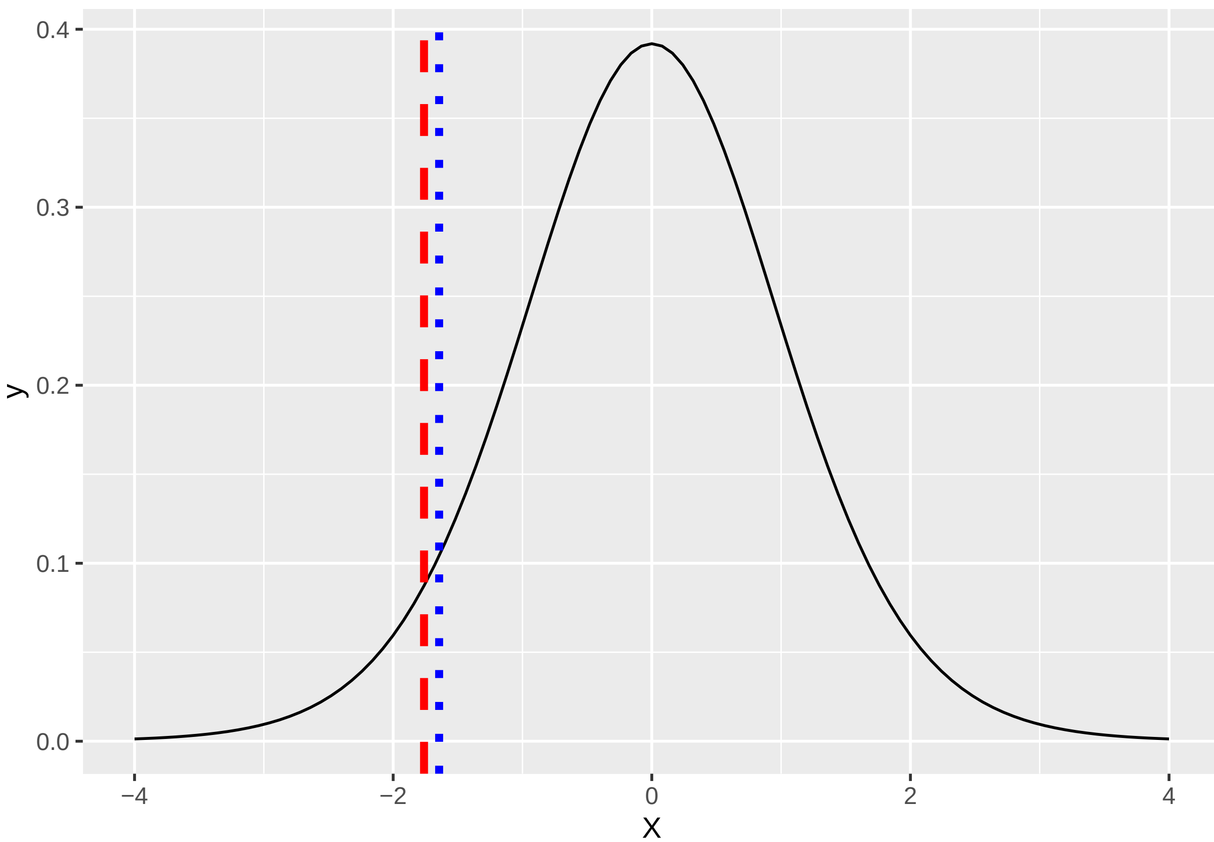
Ejercicio 2

Dibujar un gráfico como el anterior para el contraste unilateral para $\alpha = 0.05$ y $\alpha = 0.1$.

```
func_g_dt_test_uni = function(x, mu_ = 15,
                              alternativa = "less",
                              alpha = 0.05) {
#mu_ = 15
datos = data.frame(X = x)

if (!(alternativa %in% c("less","greater"))) {
  #stop("alternativa Solamente puede tomar los valores: less o greater")
  warning("alternativa Solamente puede tomar los valores: less o greater")
  #message("alternativa Solamente puede tomar los valores: less o greater")

}

resul = t.test(datos$X,mu = mu_ ,alternative = alternativa)
#str(resul)
p1 = ggplot(datos,aes(x = X)) +
  stat_function(aes(x = seq(-4,4,length = length(X))),
              fun = dt, args = list(df = length(datos$X)-1),
              color = "black") +
  geom_vline(aes(xintercept = resul$statistic),color = "blue",
             linetype = "dotted",size = 1.4)

if (alternativa=="less") {
```

```r
  return(
    p1 +
    geom_vline(aes(xintercept = qt(alpha,df = resul$parameter)),color = "red",
             linetype = "dashed",size = 1.4)
  )
}
if (alternativa=="greater") {
   return(
     p1 +
    geom_vline(aes(xintercept = qt(1-alpha,df = resul$parameter)),color = "red",
             linetype = "dashed",size = 1.4)
   )
}



}
```

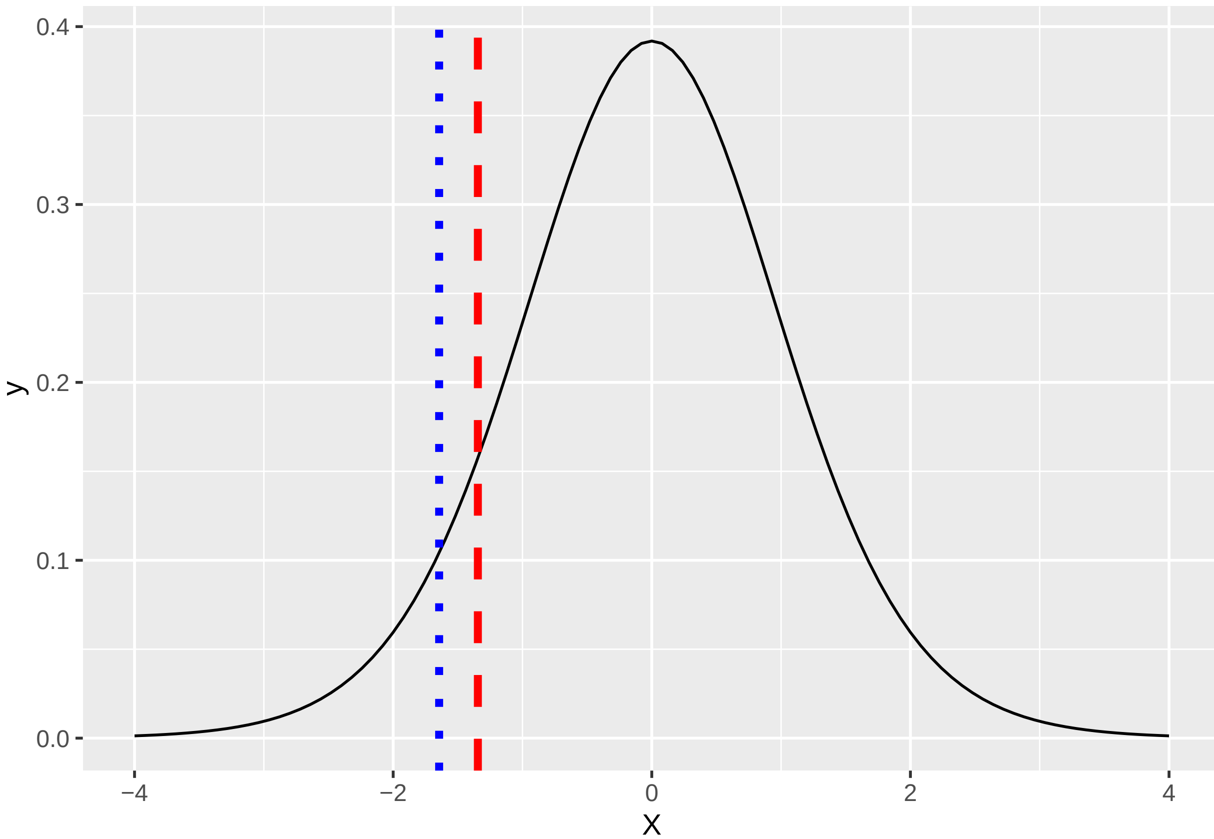
```r
func_g_dt_test_uni(x,mu_ = 15)
```




```r
func_g_dt_test_uni(x,mu_ = 15, alternativa = "two.sided")
```

```
## Warning in func_g_dt_test_uni(x, mu_ = 15, alternativa = "two.sided"):
## alternativa Solamente puede tomar los valores: less o greater
```

```r
func_g_dt_test_uni(x,mu_ = 15, alternativa = "less",alpha = 0.10)
```



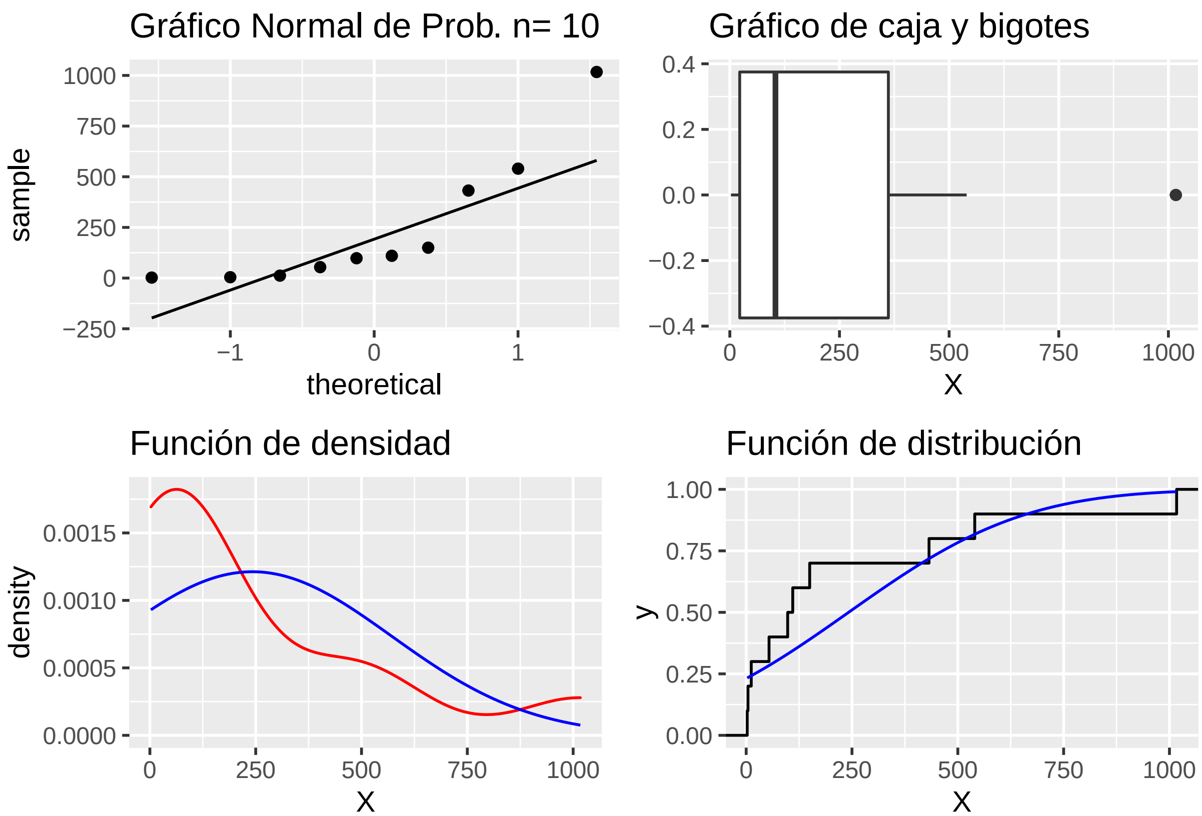
Ejercicio 3

$$110, 12, 2.5, 98, 1017, 540, 54, 4.3, 150, 432$$

Se desea contrastar H0: precio medio= 500.

```r
x = c(110, 12, 2.5, 98, 1017, 540, 54, 4.3, 150, 432)
ananor_tidy(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.76013, p-value = 0.004741
```



No podemos usar la estadística paramétrica para estudiar esta muestra.

```
wilcox.test(x, conf.int = TRUE, mu = 500)
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  x
## V = 11, p-value = 0.1055
## alternative hypothesis: true location is not equal to 500
## 95 percent confidence interval:
##   33.0 514.5
## sample estimates:
## (pseudo)median
##            150
```

Apartado a

Calcular directamente W+ (test de rango-signo de Wilcoxon)

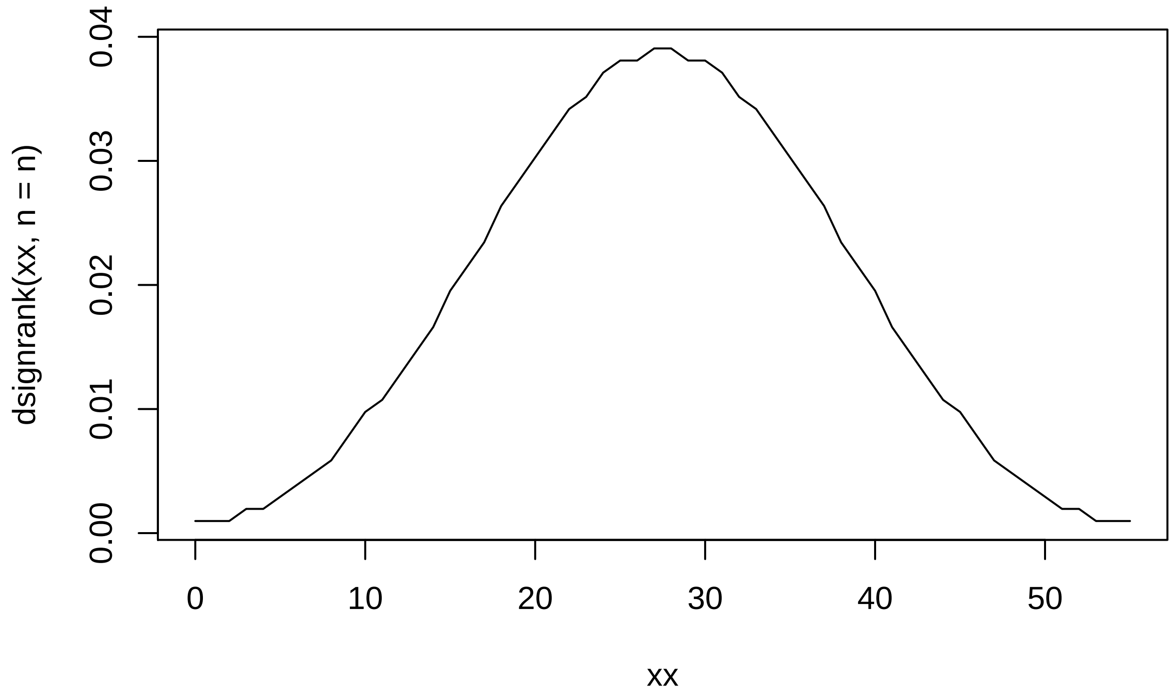
```
# W+ = Suma(rangos(|Xi|), Xi>0)
# H0 = mu=0
mu_ = 500
rangos = rank(abs(x-mu_))
#rangos[(x-mu_)>0]
(est.W =  sum(rangos[(x-mu_)>0]))
```

```
## [1] 11
```

Apartado b

Dibujar la función de probabilidad de W+ para esta n, usando `dsignrank`.

```
#dsignrank()

n = length(x)
xx = seq(0,n*(n+1)/2,1)
plot(xx,dsignrank(xx,n = n), type = "l")
```



Apartado c

Calcular E[W+] y Var[W+] directamente.

```
# E[W+]
sum(xx * dsignrank(xx, n=n))
```

```
## [1] 27.5
```

```
n*(n+1)/4
```

```
## [1] 27.5
```

```
# varianza
sum(xx^2 * dsignrank(xx,n=n)) - (n*(n+1)/4)^2
```

```
## [1] 96.25
```

```
n*(n+1)*(2*n+1)/24
```

```
## [1] 96.25
```

Ejercicio 4

En este ejemplo se considera la hipótesis nula de que la progenie de un cruce de plantas produce como resultado plantas de tipo A o B con probabilidades respectivas 1/4 y 3/4.

En un experimento se obtienen 243 de tipo A y 682 de tipo B.

Tomando la clase B como éxito,

```
H0: p=3/4
H1: p!= 3/4
```

```
binom.test(c(682,243), p = 3/4)
```

```
##
##  Exact binomial test
##
## data:  c(682, 243)
## number of successes = 682, number of trials = 925, p-value = 0.3825
## alternative hypothesis: true probability of success is not equal to 0.75
## 95 percent confidence interval:
##  0.7076683 0.7654066
## sample estimates:
## probability of success
##              0.7372973
```

Apartado a

Calcular el estadístico chi-cuadrado y comprobar que no coincide con Z^2.

```
n = 682+243
pg = 682/n
Z = (pg-(3/4))/sqrt(0.75*(1-0.75)/n)
Z^2
```

```
## [1] 0.796036
```

```
E0 = n*(1/4)
E1 = n*(3/4)
Ob0 = 243
Ob1 = 682
((E0-Ob0)^2)/E0 + ((E1-Ob1)^2)/E1
```

```
## [1] 0.796036
```

Apartado b

Calcular el estadístico chi-cuadrado con la corrección de Yates y comprobar que coincide con el estadístico que da prop.test.

```
prop.test(x = 682, n = 682+243, p = 3/4)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  682 out of 682 + 243, null probability 3/4
## X-squared = 0.72973, df = 1, p-value = 0.393
## alternative hypothesis: true p is not equal to 0.75
## 95 percent confidence interval:
##  0.7074391 0.7651554
## sample estimates:
##         p
## 0.7372973
```

Con la corrección de Yates:

```r
((abs(E0-Ob0)-0.5)^2)/E0 + ((abs(E1-Ob1)-0.5)^2)/E1
```

```
## [1] 0.7297297
```

Apartado c

`H0: p=0.4; 35 éxitos de 80 ensayos`

`H0: p=0.5; IC al 90%`

`H0: p=0.8; H1: p<0.8`

```r
prop.test(35,80,p=0.8,alternative = "less",conf.level = 0.95)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  35 out of 80, null probability 0.8
## X-squared = 63.457, df = 1, p-value = 8.195e-16
## alternative hypothesis: true p is less than 0.8
## 95 percent confidence interval:
##  0.0000000 0.5354685
## sample estimates:
##      p
## 0.4375
```