

**UNIVERSIDAD DE EL SALVADOR  
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE  
DEPARTAMENTO DE MATEMÁTICA**



**Licenciatura en Estadística**

**Control Estadístico del Paquete R**

**”UNIDAD CUATRO”**

**Práctica 18 - Estimación por intervalos de confianza (una población)**

**Alumna:  
Martha Yoana Medina Sánchez**

**Fecha de elaboración  
Santa Ana - 27 de noviembre de 2015**

## INTERVALOS DE CONFIANZA PARA UNA MEDIA POBLACIONAL. PRIMER CASO: VARIANZA CONOCIDA

```
# Define una variable n para poder cambiar facilmente el tamaño de la muestra
n = 10

# Define una variable para alfa
alfa = 0.95

# Genera dos conjuntos de n muestras aleatorias
# Ambos siguen una distribución normal de parametros

mu = 174.7
sigma = 1.5
x = rnorm(n, 174.7, 1.5)
x2 = rnorm(n, 174.7, 1.5)

sigma = 1.5
media = 174.7
margenError = qnorm(alfa/2, mean=0, sd=1, lower.tail = FALSE)*sigma/sqrt(n)
extrIzq = media - margenError
extrDer = media + margenError
print(c(extrIzq, extrDer))

## [1] 174.6703 174.7297
```

## CASO DOS: VARIANZA DESCONOCIDA

```
media <- function(x)
{
  n = length(x)
  suma <- 0.0
  for(i in 1:n) suma = suma + x[i]
  media = suma/n
}

save(media, file= "media.RData")
rm(list=ls(all=TRUE))
load("media.RData")

z <-c(3.4, 2.8, 4.4, 2.5, 3.3, 4.0, 4.8, 2.9, 5.6, 5.2, 3.7, 3.0, 3.6, 2.8, 4.8);
(media(z))
```

```
## [1] 3.786667

(sd(z))

## [1] 0.9709102

t.test(z, conf.level = 0.95)

##
## One Sample t-test
##
## data:  z
## t = 15.105, df = 14, p-value = 4.64e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  3.248995 4.324339
## sample estimates:
## mean of x
##  3.786667
```

### INTERVALOS DE CONFIANZA PARA UNA PROPORCIÓN.

```
# Utilizando la funci\on proporcionada por el R.

prop.test(360, 1200, alternative = "two.sided", conf.level=0.95)

##
## 1-sample proportions test with continuity correction
##
## data:  360 out of 1200, null probability 0.5
## X-squared = 191.2, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.2743388 0.3269581
## sample estimates:
##      p
## 0.3
```

```
# Creando nuestra propia funci\on

intervaloProp <- function(x, n, nivel.conf = 0.95)

{
  pe <- x/n
```

```

alfa <- (1 - nivel.conf)
z <- qnorm(1-alfa/2)
SE <- sqrt(pe*(1-pe)/n)
  print(rbind(pe, alfa, z, SE))
LInf <- pe-z*SE
LSup <- pe+z*SE
print(" ")
  print(paste("Intervalo para p es: [", round(LInf, 2),
              ",", round(LSup, 2), "]"))
}
x=360; n=1200; nivel.conf=0.95

intervaloProp(x, n, nivel.conf)

##           [,1]
## pe    0.30000000
## alfa  0.05000000
## z     1.95996398
## SE    0.01322876
## [1] " "
## [1] "Intervalo para p es: [ 0.27 , 0.33 ]"

```

## INTERVALOS DE CONFIANZA PARA LA VARIANZA POBLACIONAL

```

h <- c (46.4, 46.1, 45.8, 47.0, 46.1, 45.9, 45.8, 46.9, 45.2, 46.0);
h

## [1] 46.4 46.1 45.8 47.0 46.1 45.9 45.8 46.9 45.2 46.0

# Creando nuestra propia funci\on

intervalovarpobla <- function(n, nivel.conf=0.95)
{
  varianza = (var(h)) # varianza
  gl = n - 1 # grados de libertad
  alfa = 1 - nivel.conf # valor de alfa
  X_1 <- qnorm(1-(alfa/2), gl)
  X_2 <- qnorm((alfa/2), gl)
  u <- ((n - 1)*varianza)
  print(rbind(varianza, gl, alfa, X_1, X_2, u))
  LInf <- u/X_1
  LSup <- u/X_2
  print(" ")
  print(paste("Intervalo para D es: [", round(LInf, 2), ",", round(LSup, 2), "]"))
}

```

```
}  
n=10; nivel.conf=0.95  
intervalovarpobla ( n, nivel.conf)  
  
##                [,1]  
## varianza  0.2862222  
## gl        9.0000000  
## alfa      0.0500000  
## X_1       10.9599640  
## X_2       7.0400360  
## u         2.5760000  
## [1] " "  
## [1] "Intervalo para D es: [ 0.24 , 0.37 ]"
```