

Actividad 2

Jairo Enrique Alba

21/2/2023

```
#Para que funcione como pdf se deben ejecutar estas dos lineas  
#install.packages('tinytex')  
#tinytex::install_tinytex()
```

Teorema del Límite Central

El Teorema del Límite Central es uno de los más importantes en la inferencia estadística y habla sobre la convergencia de los estimadores como la proporción muestral a la distribución normal. Algunos autores afirman que esta aproximación es bastante buena a partir del umbral $n > 30$.

A continuación se presentan los siguientes pasos para su verificación:

Ejercicio 1.

- Realice una simulación en la cual genere una población de $N = 1000$ (Lote), donde el porcentaje de individuos (supongamos plantas) enfermas sea del 50%.

```
#Generar la funcion  
GenerarPob = function(Tamaño = 1000, Enfermos = 0.5, Sanos = 0.5) {  
  E = round(Tamaño * Enfermos, 0)#redondear a cero decimales  
  S = round(Tamaño * Sanos, 0)#redondear a cero decimales  
  Poblacion = c(rep("Enfermo", E), rep("Sana",S))#creo un vector con Enfermo E veces y con Sana S veces  
  print(paste("Lote Generado de tamaño: ", Tamaño, " Enfermos: ",E,"Sanos: ",S))#combinan varias entradas  
  return(Poblacion)#R devolverá el resultado de cualquier comando que se ejecute en la última línea de  
}  
  
Lote = GenerarPob(Tamaño = 1000, Enfermos = 0.5, Sanos = 0.5) #mostar el tamaño de población y cuantos individuos  
  
## [1] "Lote Generado de tamaño: 1000 Enfermos: 500 Sanos: 500"
```

- Genere una función que permita:
 - Obtener una muestra aleatoria de la población y
 - Calcule el estimador de la proporción muestral \hat{p} para un tamaño de muestra dado n .

```

#Funcion para generar la muestra
GenerarMuestra = function(Poblacion, Tamaño, Condicion) {
  if (Tamaño<=length(Poblacion)) { #condición muestra mas pequeña de la población
    muestra = sample(x = Poblacion, size = Tamaño)#toma la muestra
    P_muestra = sum(muestra == Condicion)/Tamaño # para calcular la proporción
    return(P_muestra)#devuelve el resultado
  }
  else { #de lo contrario si el tamaño de la muestra supera la población
    print("Error: el tamaño de la muestra supera el tamaño de la población.")
    return(-1)
  }
}

#Generacion de la muestra de determinado tamaño

n = 200 #tamaño de muestra
muestra = GenerarMuestra (Poblacion = Lote, Tamaño = n, Condicion = "Enfermo") #generación de la muestra

print(paste("Para una muestra de tamaño:", n, "se obtuvo un ^P =",muestra)) #imprimir los resultados

```

```
## [1] "Para una muestra de tamaño: 200 se obtuvo un ^P = 0.475"
```

- Repita el escenario anterior (b) $n = 500$ veces y analice los resultados en cuanto al comportamiento de los 500 resultados del estimador \hat{p} .
 - ¿Qué tan simétricos o sesgados son los resultados obtenidos? y
 - ¿qué se puede observar en cuanto a la variabilidad?.
 - Realice en su informe un comentario sobre los resultados obtenidos.

```

#generación de 500 muestras de la población y mirar las proporciones muestrales de enfermos en cada una
GenerarMuestras_Iter = function(Poblacion, Tamaño, Condicion, Iteraciones) { #condiciones de la funcion
  if (Tamaño<=length(Poblacion)) {
    muestras_iter = array(NA,Iteraciones)#creación del array con NA, de dimensión el numero de iteraciones
    for (i in 1:Iteraciones) {
      muestras_iter[i] = GenerarMuestra (Poblacion, Tamaño, Condicion)
    }
    return(muestras_iter)
  }
  else {
    print("Error: el tamaño de la muestra supera el tamaño de la población.")
    return(-1)
  }
}

iter = 500
P_muestras = GenerarMuestras_Iter (Poblacion = Lote, Tamaño = n,
                                   Condicion = "Enfermo",Iteraciones = iter)

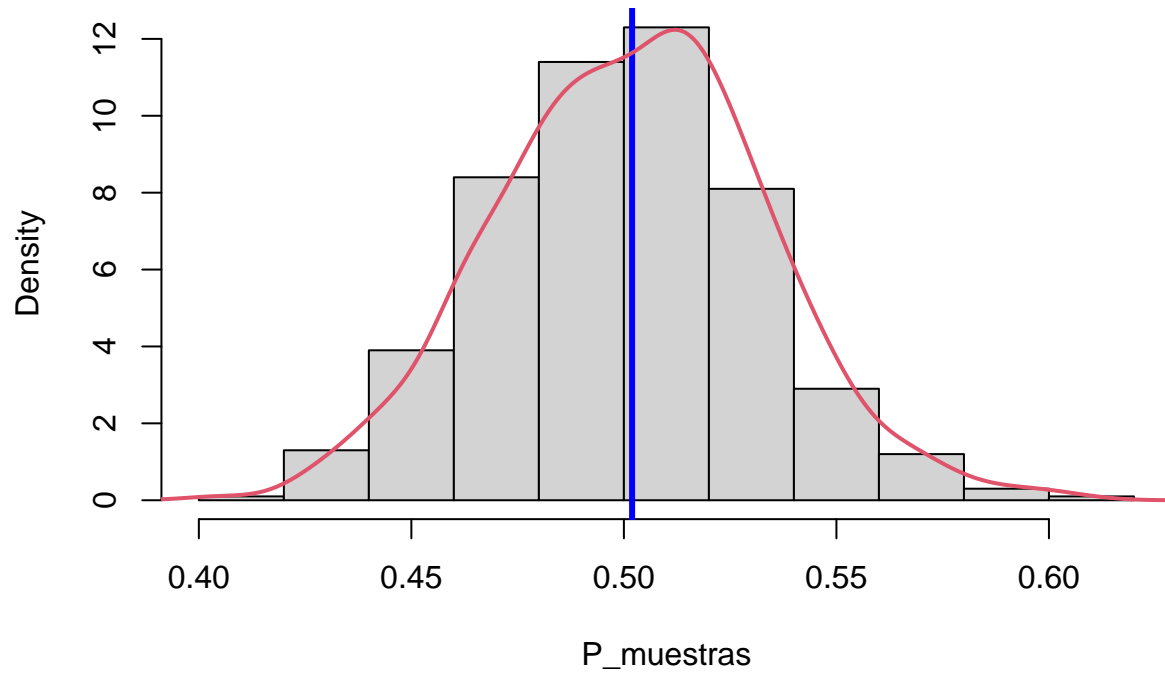
```

```

hist(P_muestras, prob = TRUE, main = "Proporción enfermos en las 500 muestras")
line = mean(P_muestras)
abline(v=line, col="blue", lwd=3)
lines(density(P_muestras), col = 2, lwd = 2)

```

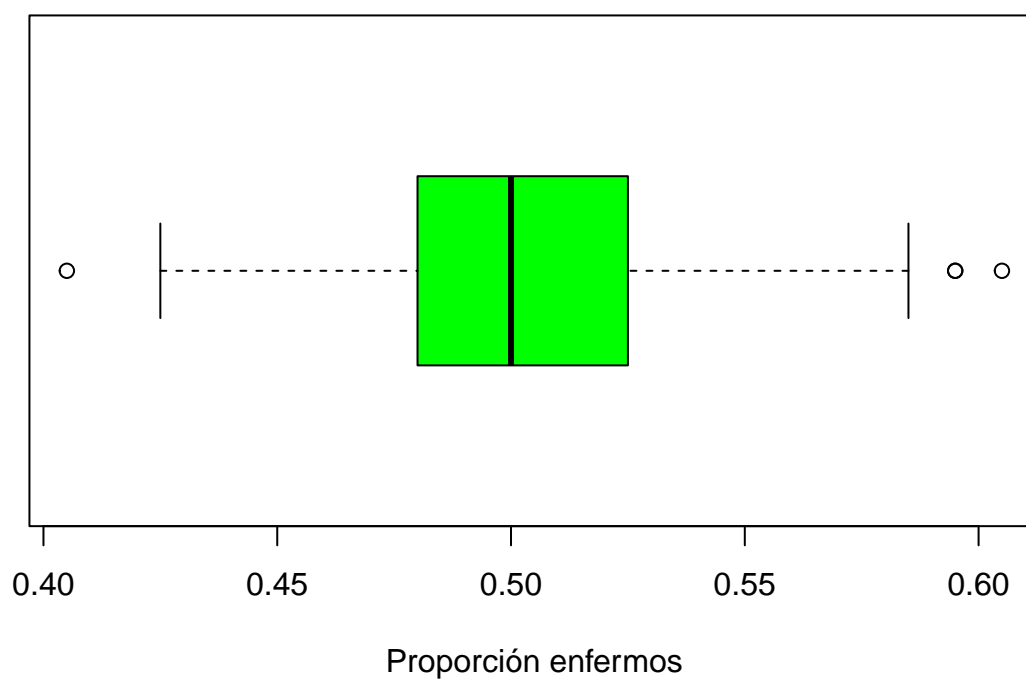
Proporción enfermos en las 500 muestras



```
library(dplyr)

# Diagrama de caja
boxplot(P_muestras, main = "Proporción enfermos en las 500 muestras",
        xlab = "Proporción enfermos", col = "green", horizontal=T)
```

Proporción enfermos en las 500 muestras



```
#install.packages("moments") solo una vez  
library(moments)  
mean(P_muestras)
```

```
## [1] 0.50193
```

```
sd(P_muestras)
```

```
## [1] 0.03153433
```

```
min(P_muestras)
```

```
## [1] 0.405
```

```
median(P_muestras)
```

```
## [1] 0.5
```

```
max(P_muestras)
```

```
## [1] 0.605
```

```
skewness(P_muestras)
```

```
## [1] 0.06172948
```

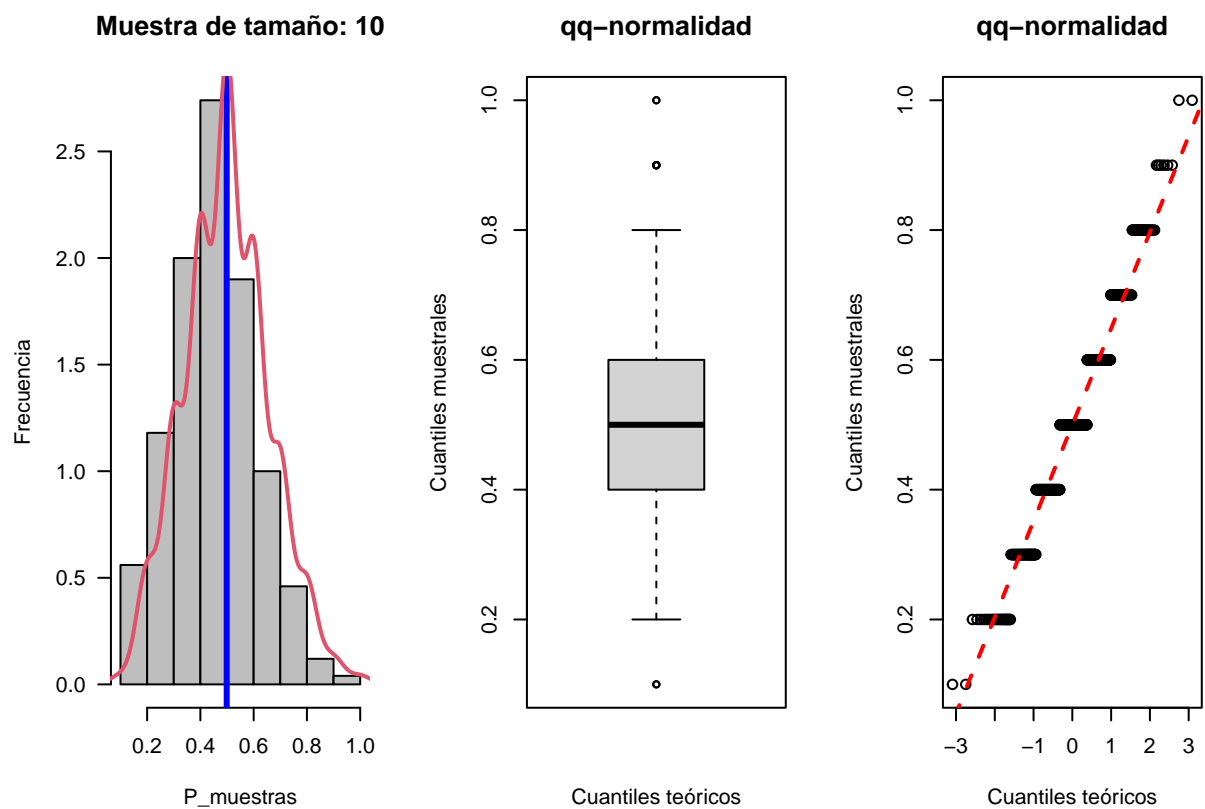
```
kurtosis(P_muestras)
```

```
## [1] 3.053424
```

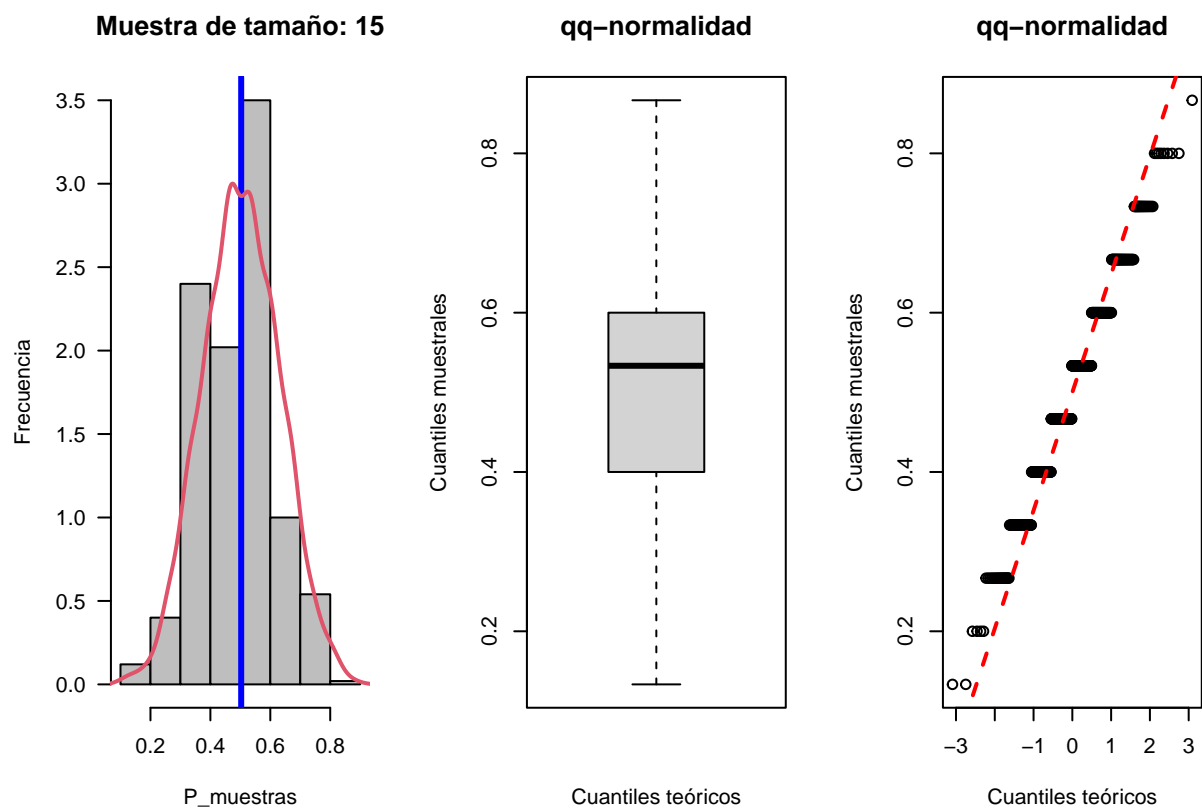
- Repita los puntos b y c para tamaños de muestra $n=5, 10, 15, 20, 30, 50, 60, 100, 200, 500$. Compare los resultados obtenidos para los diferentes tamaños de muestra en cuanto a la normalidad. Utilice pruebas de bondad y ajuste (shapiro wilks :shapiro.test()) y métodos gráficos (grafico de normalidad: qqnorm()). Comente en su informe los resultados obtenidos.

```
t_Muestras = c(10,15,20,30,50,60,100,200,500)
iter = 500
```

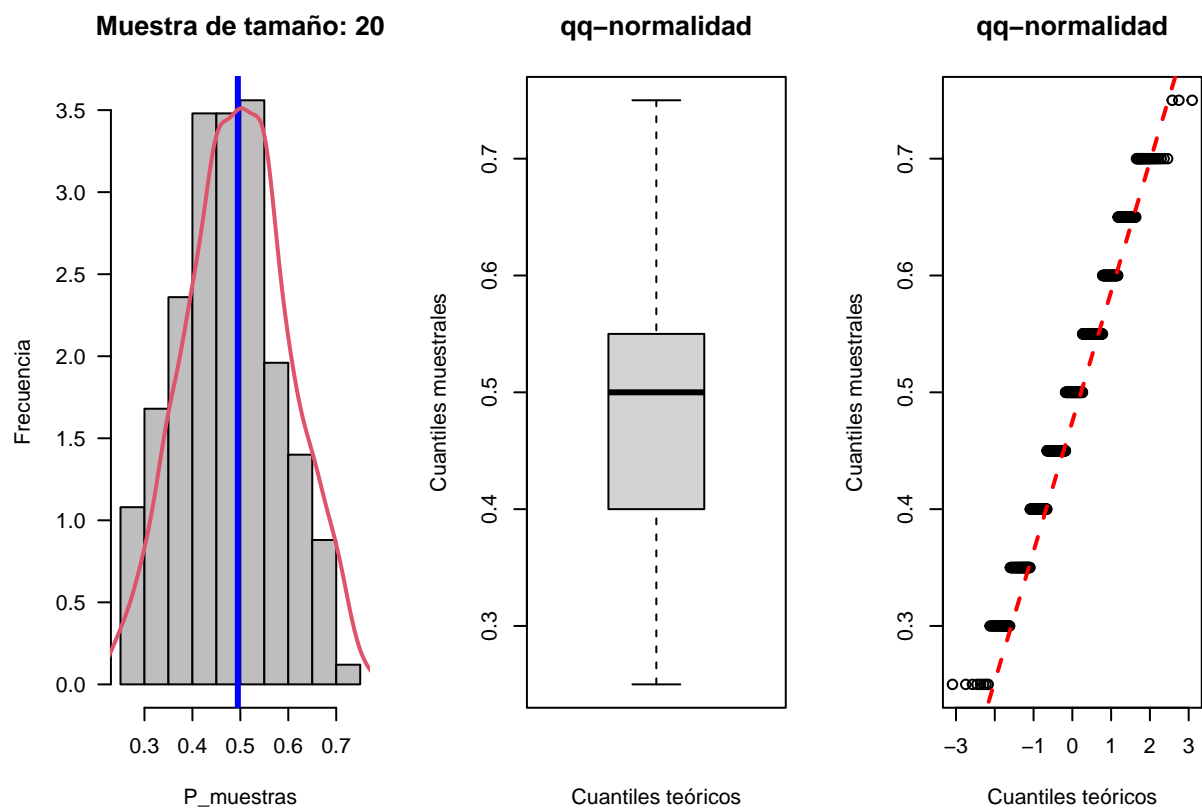
```
for (i in 1:length(t_Muestras)) {
  P_muestras = GenerarMuestras_Iter (Poblacion = Lote, Tamaño = t_Muestras[i], Condicion = "Enfermo"
  par(mfrow=c(1,3))
  hist(P_muestras, las=1, ylab = "Frecuencia", title = t_Muestras[i], main = paste("Muestra de tamaño", t_Muestras[i]))
  abline(v=mean(P_muestras), col="blue", lwd=3)
  lines(density(P_muestras), col = 2, lwd = 2)
  boxplot(P_muestras, xlab="Cuantiles teóricos", ylab="Cuantiles muestrales",main="qq-normalidad")
  qqnorm(P_muestras, xlab="Cuantiles teóricos", ylab="Cuantiles muestrales",main="qq-normalidad")
  qqline(P_muestras,col = 'red', lwd = 2, lty = 2)
  print(paste('Para una muestra de tamaño: ', t_Muestras[i]))
  print(shapiro.test(P_muestras))
  x=data.frame("ID"=i,"Tamaño_muestra"=t_Muestras[i],
               "Media"=mean(P_muestras),
               "Mediana"=median(P_muestras),
               "Desvest"=sd(P_muestras),
               "Varianza"=var(P_muestras),
               "Mín."=min(P_muestras),
               "Máx"=max(P_muestras),
               "Asimetría"=skewness(P_muestras),
               "Curtosis"= kurtosis(P_muestras)
               )
  Resultado = rbind(x)
}
```



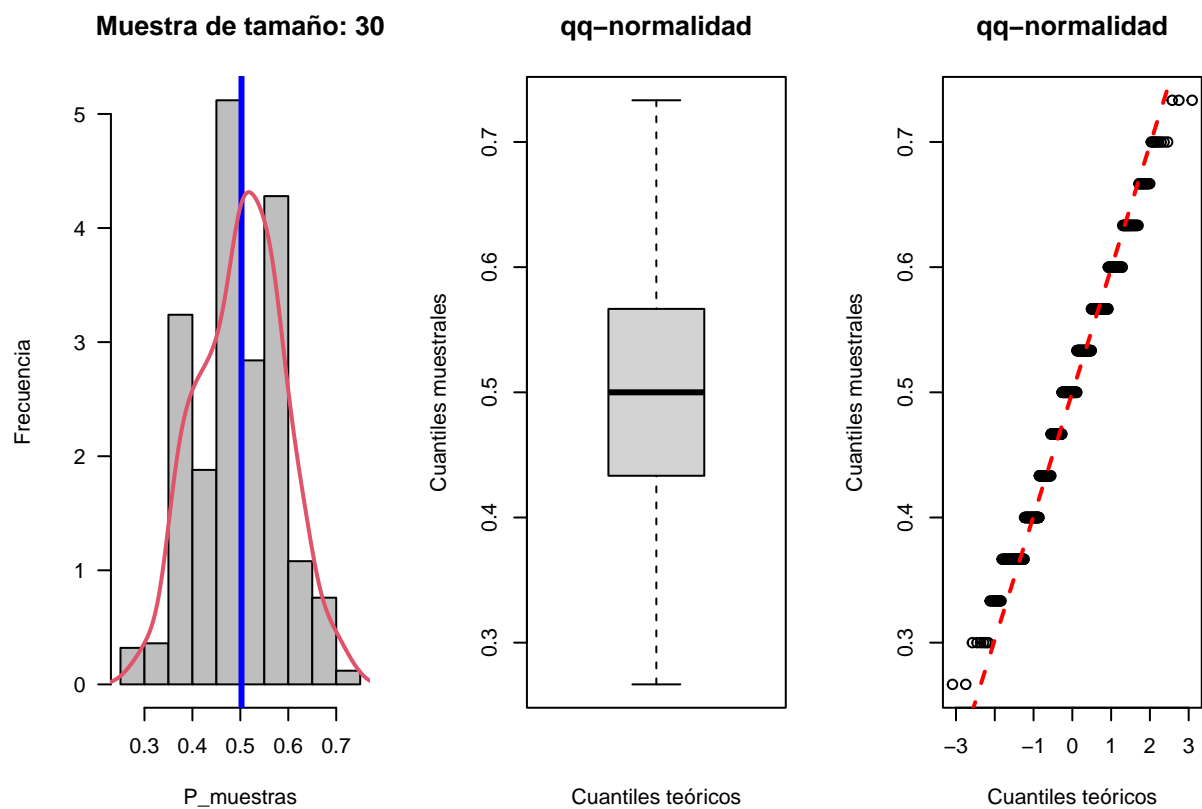
```
## [1] "Para una muestra de tamaño: 10"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.96243, p-value = 5.384e-10
```



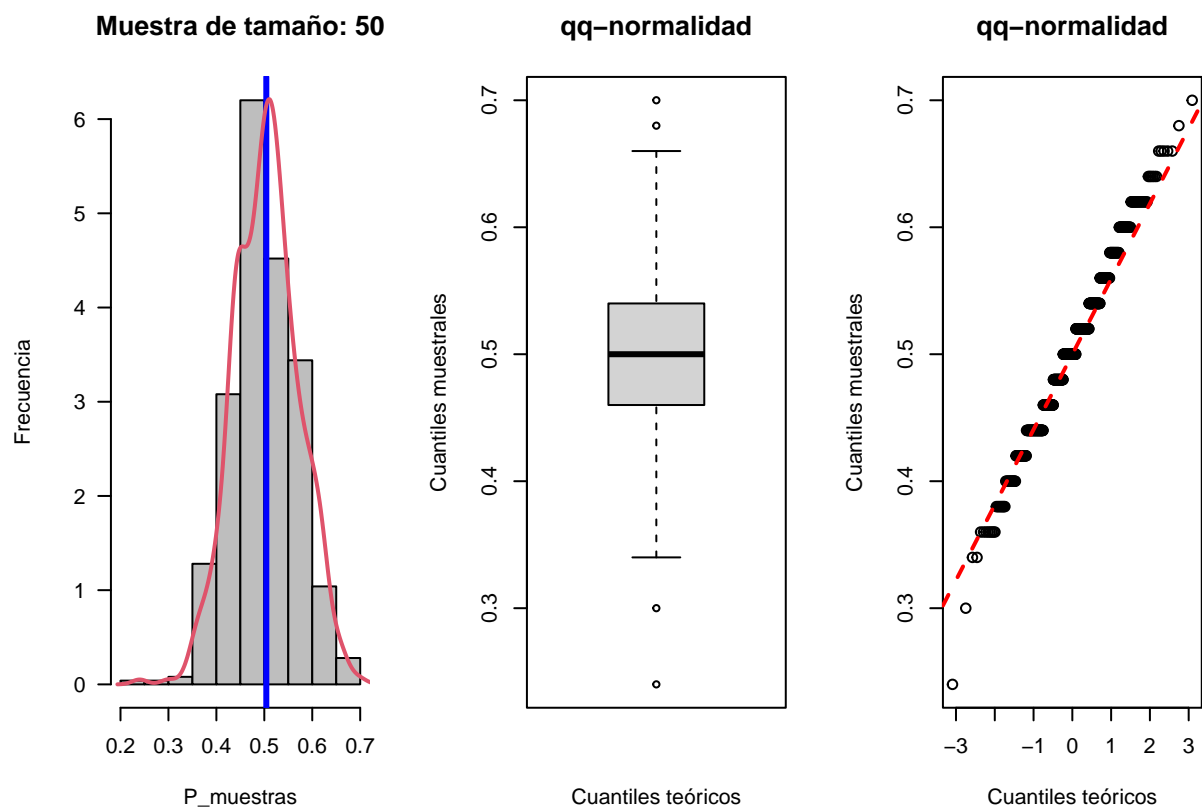
```
## [1] "Para una muestra de tamaño: 15"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.97589, p-value = 2.435e-07
```



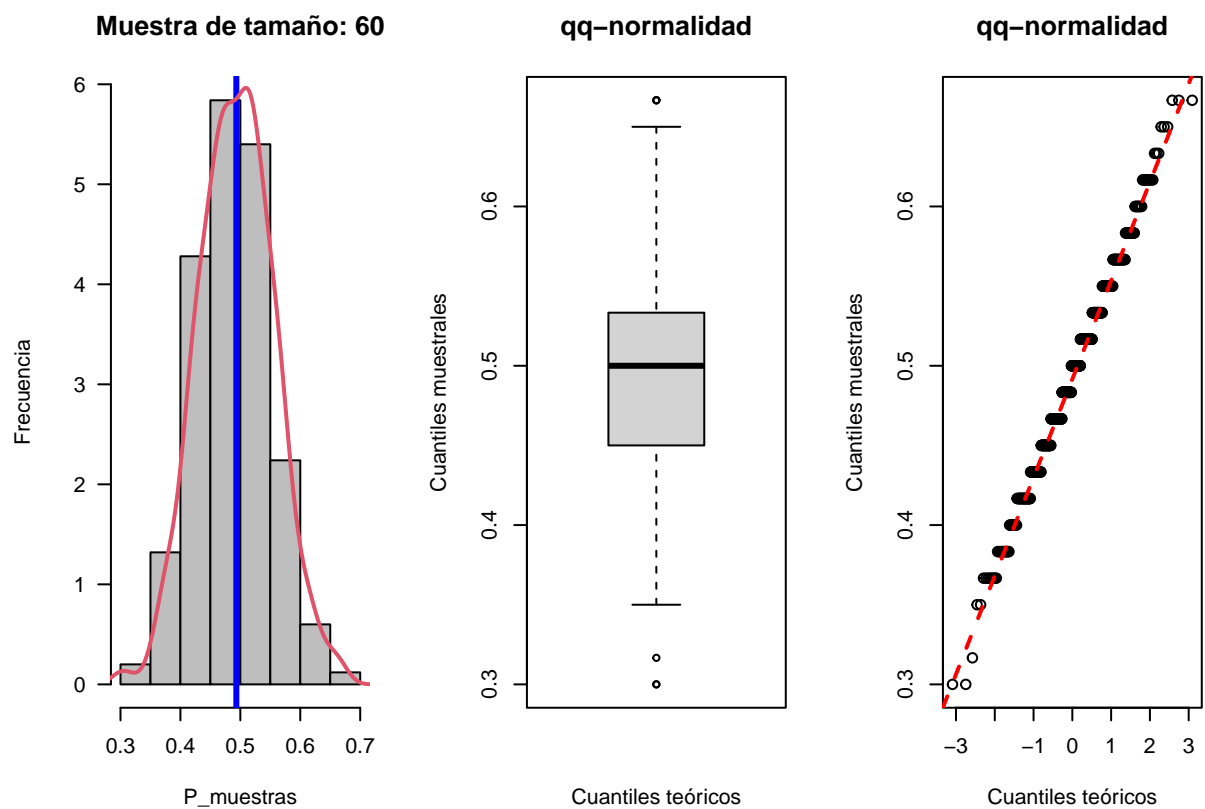
```
## [1] "Para una muestra de tamaño: 20"
##
##  Shapiro-Wilk normality test
##
## data:  P_muestras
## W = 0.97579, p-value = 2.314e-07
```

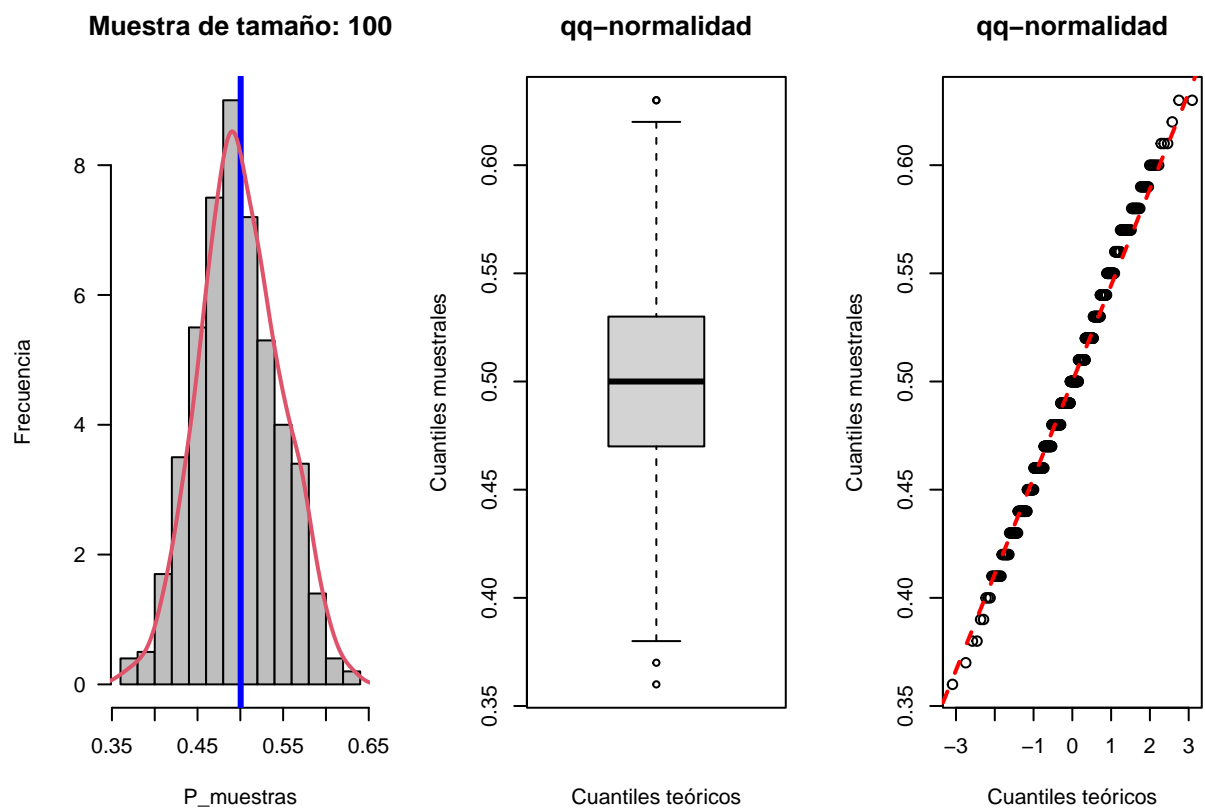
```
## [1] "Para una muestra de tamaño: 30"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.98326, p-value = 1.63e-05
```



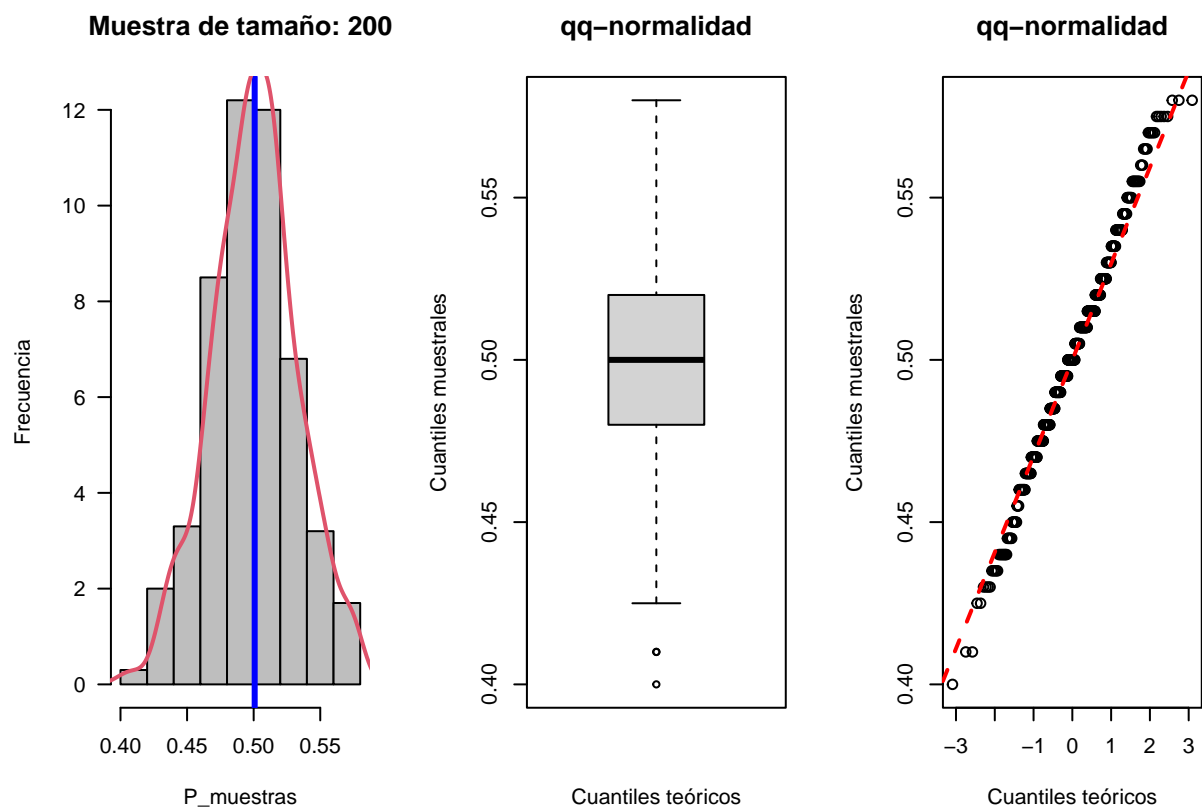
```
## [1] "Para una muestra de tamaño: 50"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.98934, p-value = 0.001074
```



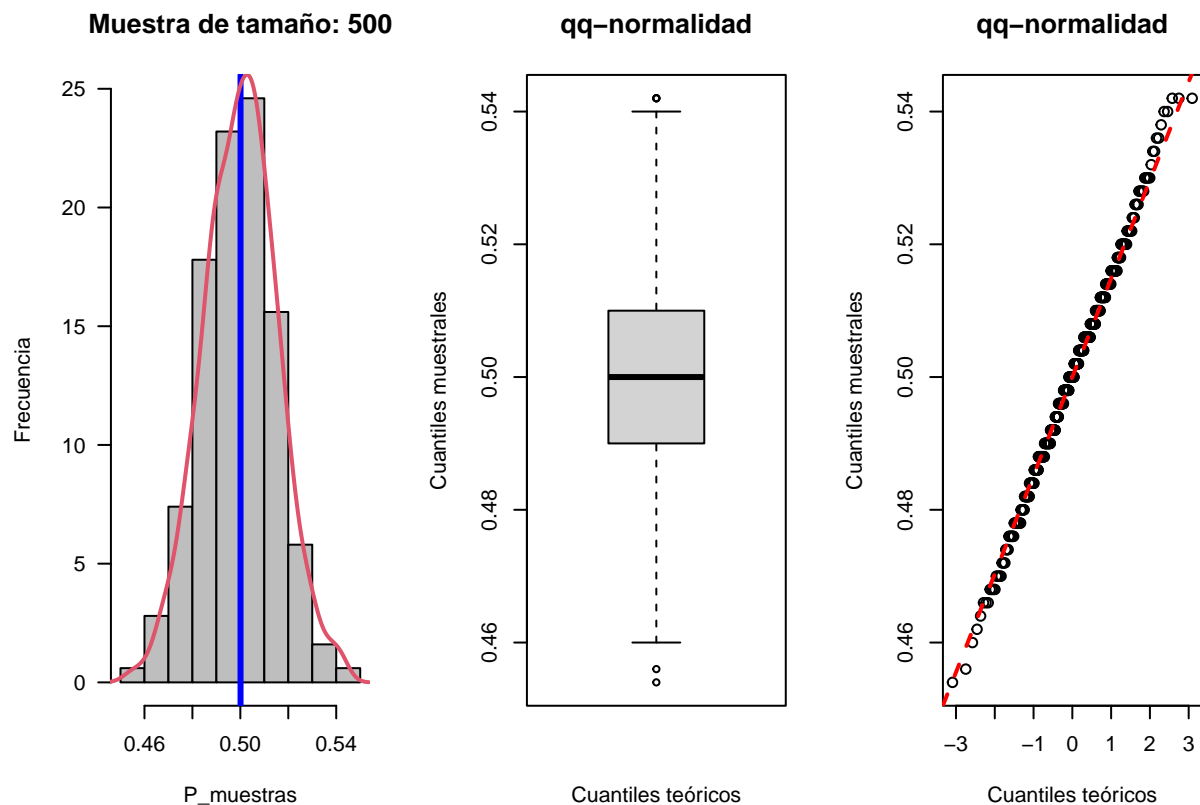
```
## [1] "Para una muestra de tamaño: 60"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.99217, p-value = 0.009978
```



```
## [1] "Para una muestra de tamaño: 100"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.99372, p-value = 0.03619
```



```
## [1] "Para una muestra de tamaño: 200"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.99448, p-value = 0.0682
```



```
## [1] "Para una muestra de tamaño: 500"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.99691, p-value = 0.4632
```

- Repita toda la simulación (puntos a – d), pero ahora para lotes con 10% de plantas enfermas y de nuevo para lotes con un 90% de plantas enfermas. Concluya sobre los resultados del ejercicio.

Para el 90% de plantas enfermas

```
Lote3 = GenerarPob(Tamaño = 1000, Enfermos = 0.9, Sanos = 0.1)
```

```
## [1] "Lote Generado de tamaño: 1000 Enfermos: 900 Sanos: 100"
```

```
n3 = 300
muestra3 = GenerarMuestra (Poblacion = Lote3, Tamaño = n, Condicion = "Enfermo")

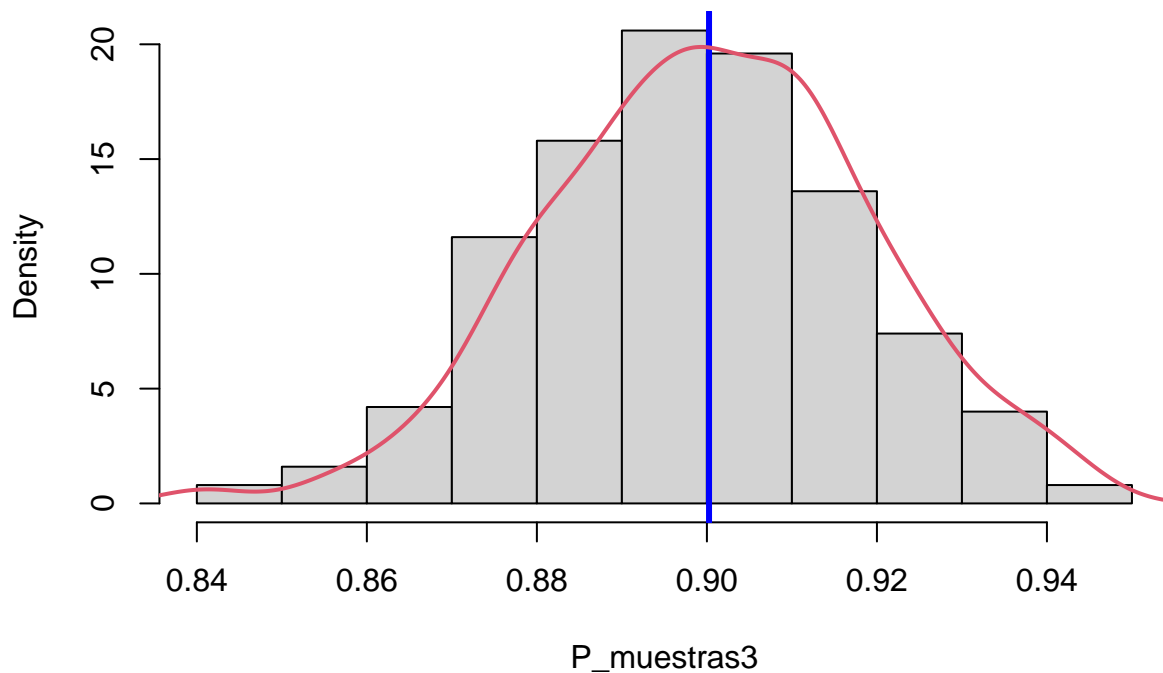
print(paste("Para una muestra de tamaño:", n3, "se obtuvo un ^P =", muestra3))
```

```
## [1] "Para una muestra de tamaño: 300 se obtuvo un ^P = 0.905"
```

```
P_muestras3 = GenerarMuestras_Iter (Poblacion = Lote3, Tamaño = n, Condicion = "Enfermo", Iteraciones = 1000)

hist(P_muestras3, prob = TRUE, main = "Histogram with density curve")
line = mean(P_muestras3)
abline(v=line, col="blue", lwd=3)
lines(density(P_muestras3), col = 2, lwd = 2)
```

Histogram with density curve



```
Resultado3=data.frame("ID"=0, "Tamaño_muestra"=n3, "Media"=mean(P_muestras3), "Mediana"=median(P_muestras3),
"Desvest"=sd(P_muestras3), "Varianza"=var(P_muestras3), "Mín."=min(P_muestras3), "Máx"=max(P_muestras3),
"Asimetría"=skewness(P_muestras3), "Curtosis"=kurtosis(P_muestras3))

Resultado3
```

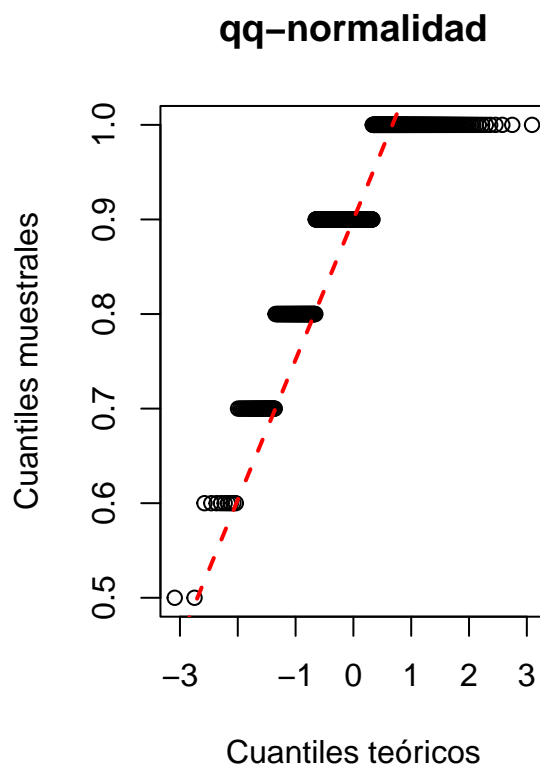
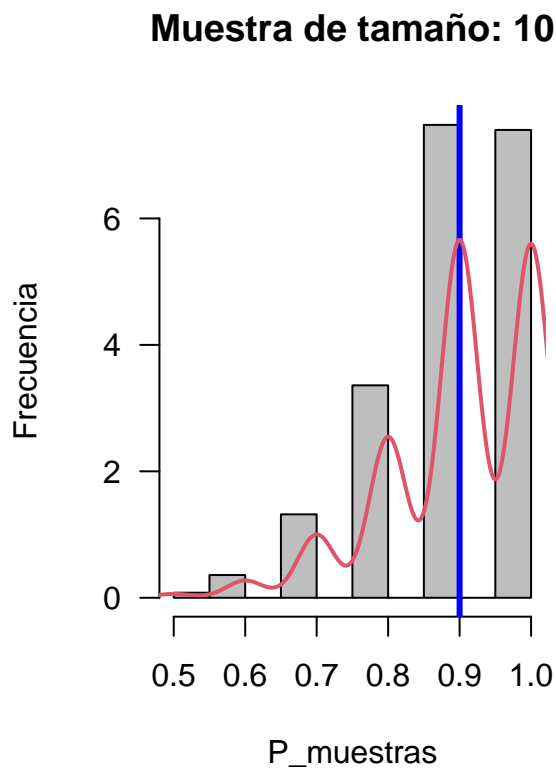
```
## ID Tamaño_muestra Media Mediana Desvest Varianza Mín. Máx
## 1 0 300 0.90025 0.9 0.01907919 0.0003640155 0.84 0.945
## Asimetría Curtosis
## 1 -0.1526121 3.006409
```

```
for (i in 1:length(t_Muestras)) {
  P_muestras = GenerarMuestras_Iter (Poblacion = Lote3, Tamaño = t_Muestras[i], Condicion = "Enfermo", Iteraciones = 1000)
  par(mfrow=c(1,2))
  hist(P_muestras, las=1, ylab = "Frecuencia", title = t_Muestras[i], main = paste("Muestra de tamaño", t_Muestras[i]))
  abline(v=mean(P_muestras), col="blue", lwd=3)
  lines(density(P_muestras), col = 2, lwd = 2)
  qqnorm(P_muestras, xlab="Cuantiles teóricos", ylab="Cuantiles muestrales", main="qq-normalidad")
  qqline(P_muestras, col = 'red', lwd = 2, lty = 2)
```

```

print(paste('Para una muestra de tamaño: ', t_Muestras[i]))
print(shapiro.test(P_muestras))
x=data.frame("ID"=i,"Tamaño_muestra"=t_Muestras[i],
             "Media"=mean(P_muestras),
             "Mediana"=median(P_muestras),
             "Desvest"=sd(P_muestras),
             "Varianza"=var(P_muestras),
             "Mín."=min(P_muestras),
             "Máx"=max(P_muestras),
             "Asimetría"=skewness(P_muestras),
             "Curtosis"= kurtosis(P_muestras)
            )
Resultado3=rbind(Resultado3,x)
}

```

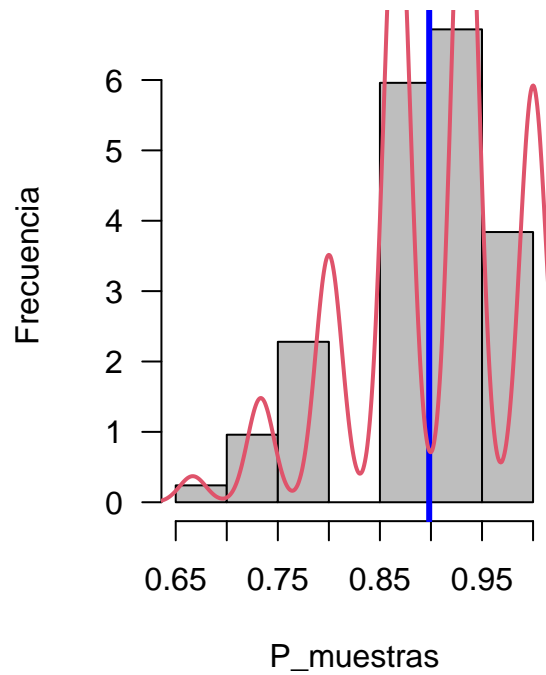


```

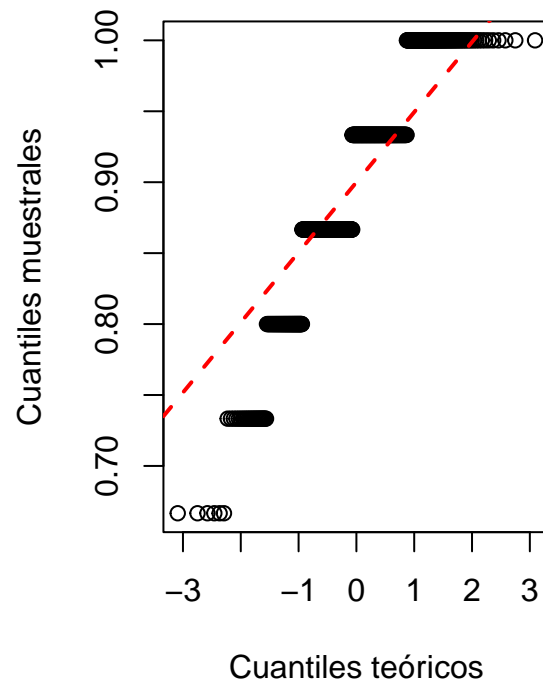
## [1] "Para una muestra de tamaño: 10"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.82953, p-value < 2.2e-16

```


Muestra de tamaño: 15

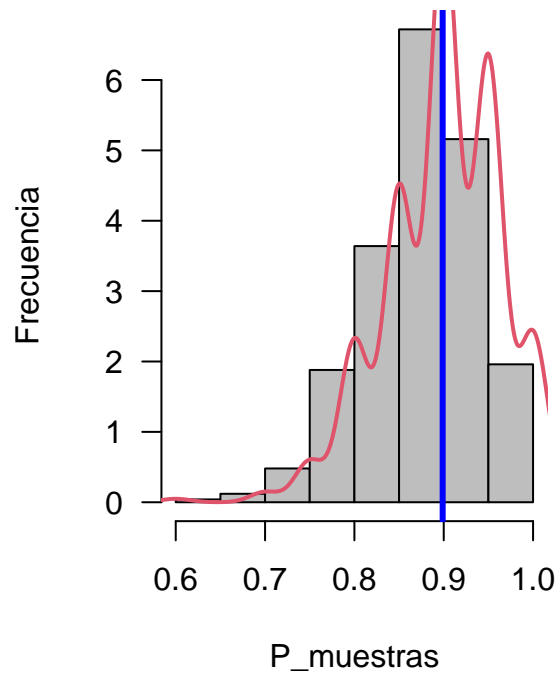


qq-normalidad

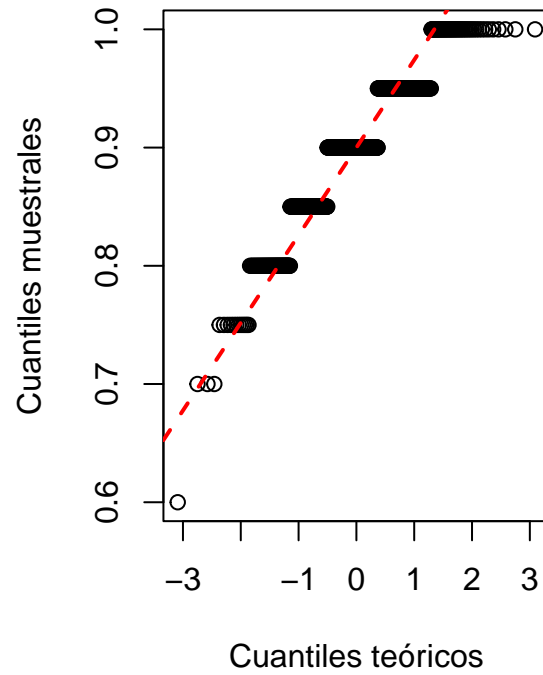


```
## [1] "Para una muestra de tamaño: 15"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.89885, p-value < 2.2e-16
```

Muestra de tamaño: 20

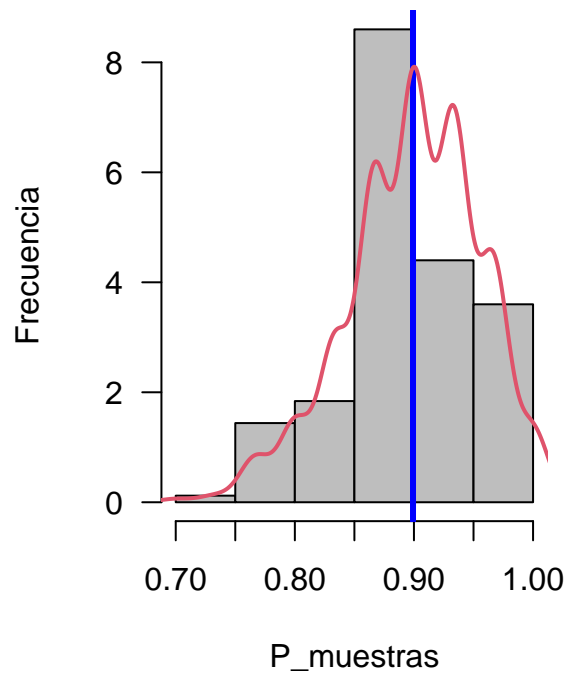


qq-normalidad

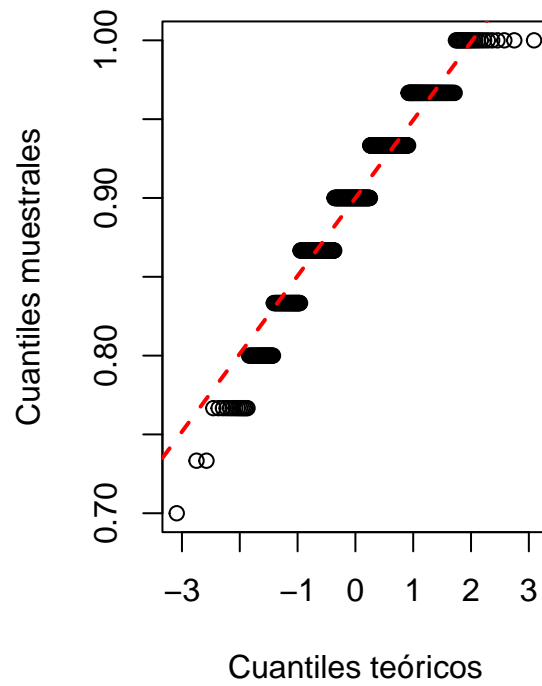


```
## [1] "Para una muestra de tamaño: 20"
##
##  Shapiro-Wilk normality test
##
## data:  P_muestras
## W = 0.92154, p-value = 1.801e-15
```

Muestra de tamaño: 30

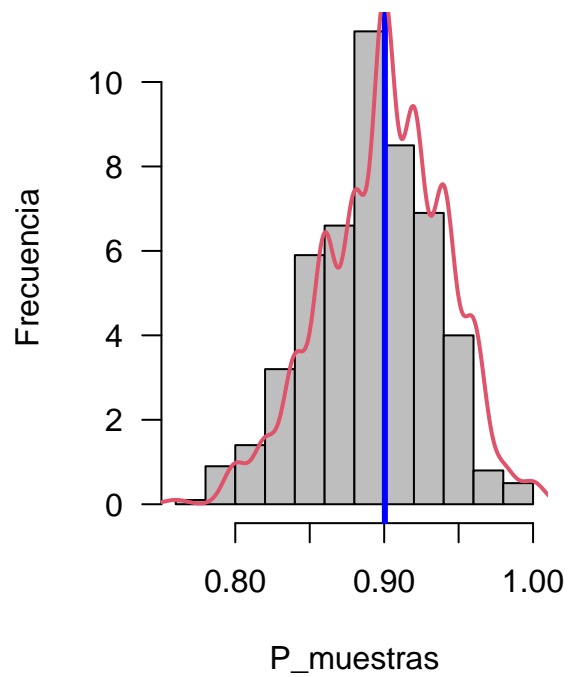


qq-normalidad

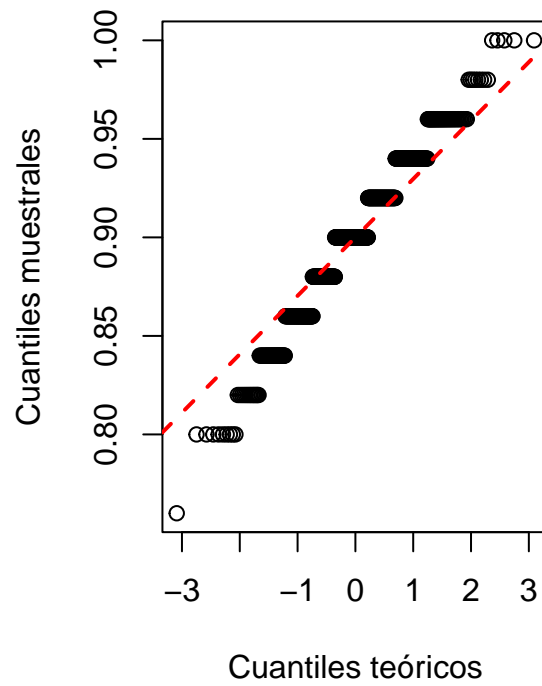


```
## [1] "Para una muestra de tamaño: 30"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.95252, p-value = 1.364e-11
```

Muestra de tamaño: 50

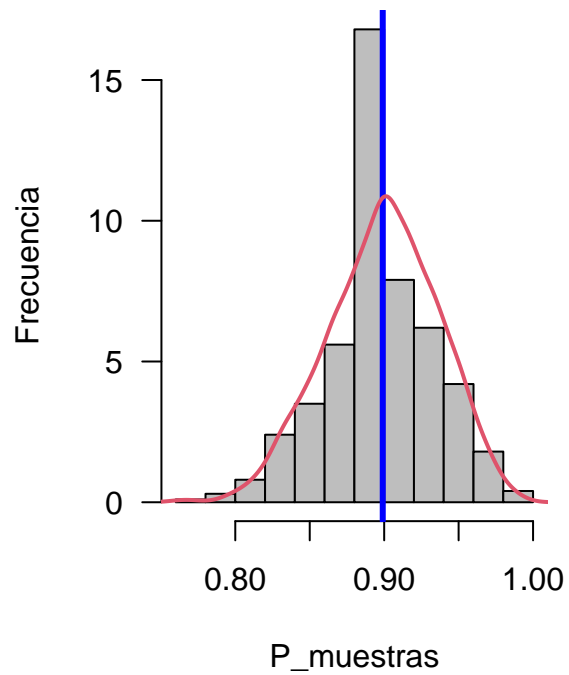


qq-normalidad

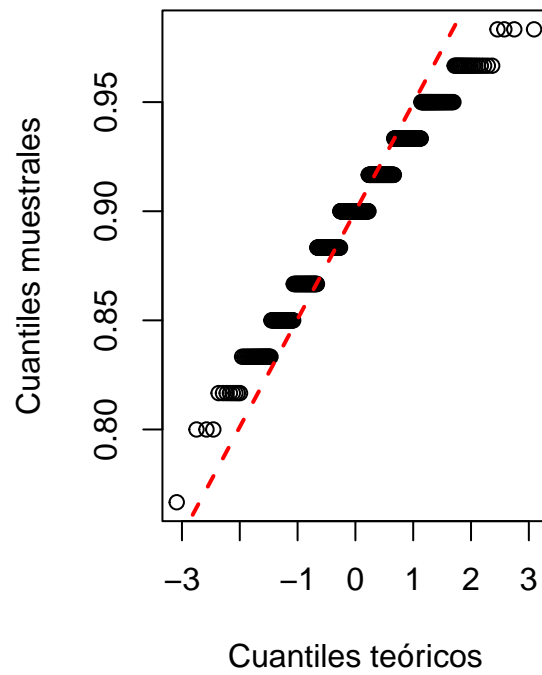


```
## [1] "Para una muestra de tamaño: 50"  
##  
## Shapiro-Wilk normality test  
##  
## data: P_muestras  
## W = 0.97349, p-value = 7.198e-08
```

Muestra de tamaño: 60

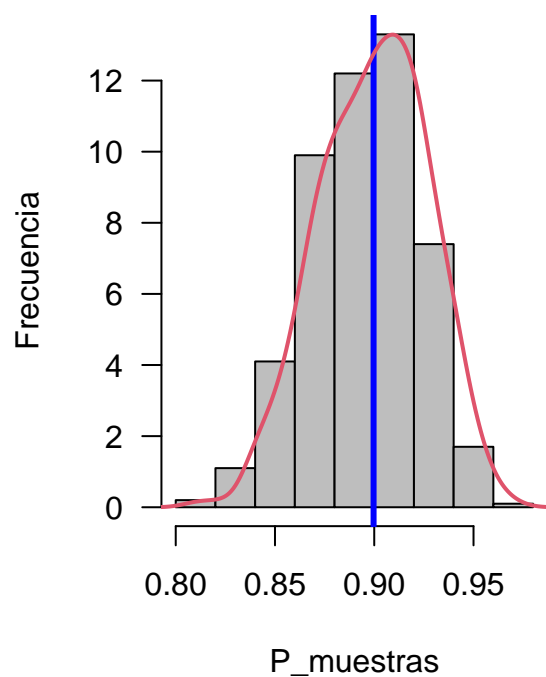


qq-normalidad

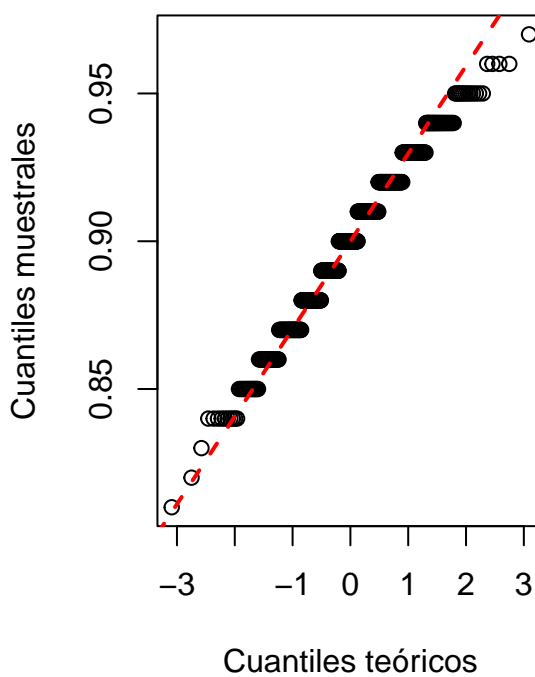


```
## [1] "Para una muestra de tamaño: 60"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.97693, p-value = 4.214e-07
```

Muestra de tamaño: 100

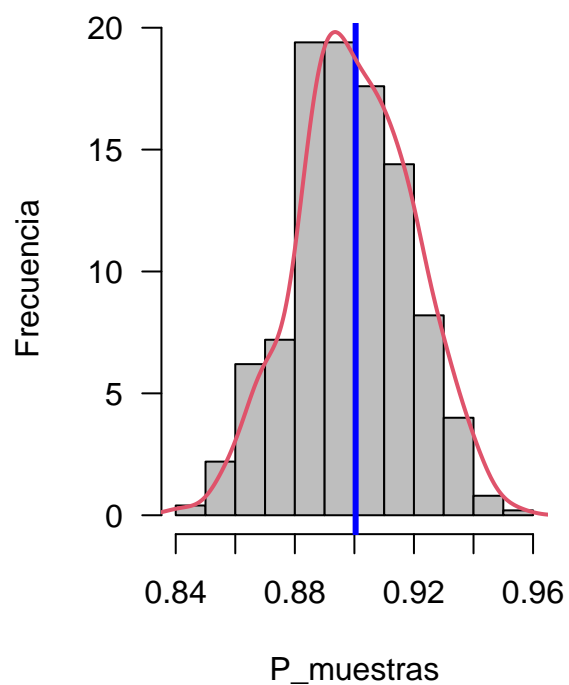


qq-normalidad

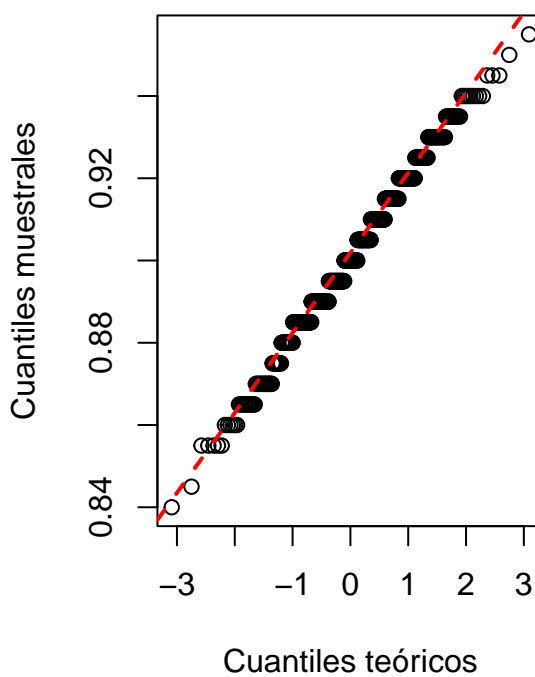


```
## [1] "Para una muestra de tamaño: 100"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.98306, p-value = 1.442e-05
```

Muestra de tamaño: 200

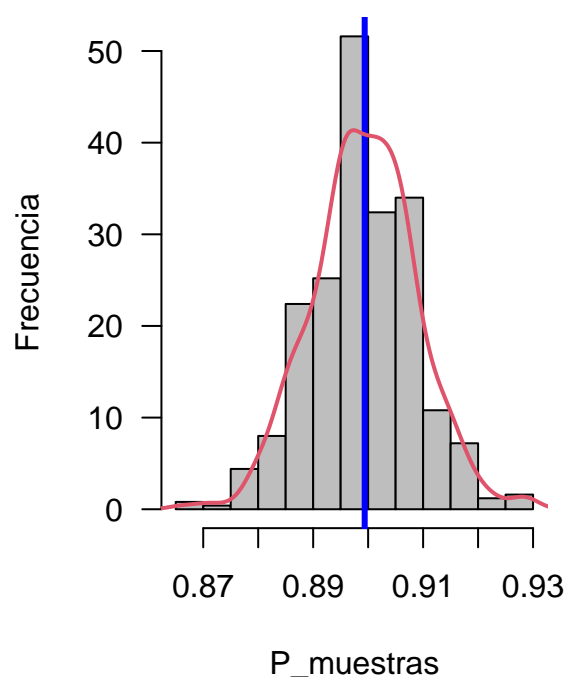


qq-normalidad

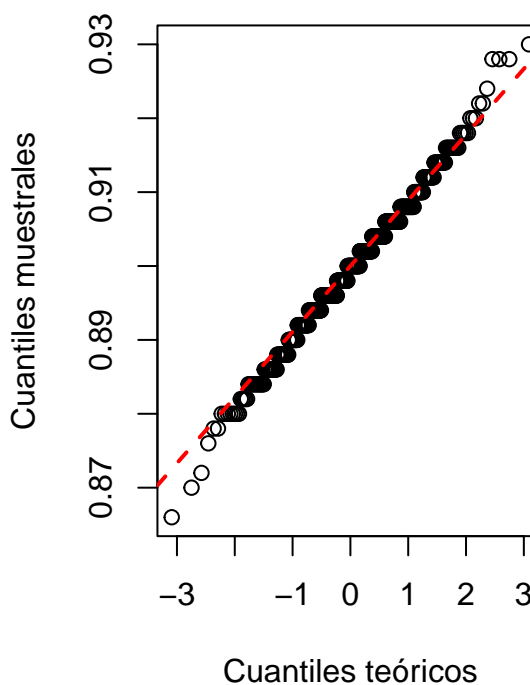


```
## [1] "Para una muestra de tamaño: 200"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.99227, p-value = 0.01079
```

Muestra de tamaño: 500



qq-normalidad



```
## [1] "Para una muestra de tamaño: 500"
##
## Shapiro-Wilk normality test
##
## data: P_muestras
## W = 0.99323, p-value = 0.02396
```

Resultado3

ID	Tamaño_muestra	Media	Mediana	Desvest	Varianza	Mín.	Máx	Asimetría	Curtosis	
1	0	300	0.9002500	0.9000000	0.019079191	3.640155e-04	0.8400000	0.9450000	-0.15261206	3.006409
2	1	10	0.9000000	0.9000000	0.101491878	1.030060e-02	0.5000000	1.0000000	-1.02466424	3.856985
3	2	15	0.8982667	0.9333333	0.075892199	5.759626e-03	0.6666667	1.0000000	-0.60891211	3.088510
4	3	20	0.8988000	0.9000000	0.063138998	3.986533e-03	0.6000000	1.0000000	-0.61834971	3.734980
5	4	30	0.8992000	0.9000000	0.054902436	3.014277e-03	0.7000000			
6	5	50	0.9004800	0.9000000	0.040653105	1.652675e-03	0.7600000			
7	6	60	0.8990333	0.9000000	0.036985301	1.367912e-03	0.7666667			
8	7	100	0.8996600	0.9000000	0.027393159	7.503852e-04	0.8100000			
9	8	200	0.9004100	0.9000000	0.019543433	3.819458e-04	0.8400000			
10	9	500	0.8993720	0.9000000	0.009534118	9.089941e-05	0.8660000			


```
## 5  1.0000000 -0.47342676 3.103919
## 6  1.0000000 -0.25329888 2.929980
## 7  0.9833333 -0.25172981 2.827171
## 8  0.9700000 -0.20270041 2.652099
## 9  0.9550000 -0.04167783 2.791975
## 10 0.9300000  0.01219339 3.381549
```

Para el 10% de plantas enfermas