

Ejercicio 2

Keyner Martinez Miranda, Jairo Enrique Alba.

5/3/2023

Segundo Punto Validación de resultados.

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 diferencia de las proporción muestrales a la distribución normal.

A continuación se presentan los siguientes pasos para su verificación:

```
#Generar la población
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( E ..E S..S).
  print(paste("Lote Generado de tamaño: ", Tamaño, " Enfermos: ",E,"Sanos: ",S))
  #combinan varias entradas en una cadena de caracteres, luego imprime los resultados
  return(Poblacion)
  #R devolverá el resultado de cualquier comando que se ejecute en la última línea
  #de la función
}
```

```
Lote = GenerarPob(Tamaño = 1000, Enfermos = 0.5, Sanos = 0.5)
#mostar el tamaño de población y cuantos hay en cada categoria
```

```
#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, dando las condiciones respectivas
```

```
print(paste("Para una muestra de tamaño:", n, "se obtuvo un  $\hat{P}$  =", muestra))  
#imprimir los resultados
```

La comparación de tratamientos es una práctica fundamental en las ciencias agropecuarias y para ello a nivel estadístico se cuenta con herramientas para apoyar el proceso de toma de decisiones y así poder lograr concluir con un alto grado de confianza sobre los resultados observados en una muestra. A través de la simulación es posible evaluar estimadores y sus propiedades, que nos permitan usarlos con toda tranquilidad.

Suponga un escenario en el cual se aplicó tratamientos diferentes a dos lotes de una misma plantas y se desea analizar si alguno de los dos tratamientos presenta un mejor desempeño en el control de una plaga presente en ambos al momento inicial. Para ello utilizará como criterio de desempeño el tratamiento, el menor porcentaje de plantas enfermas presente después de un tiempo de aplicación (es decir, si se presentan o no diferencias en las proporciones de enfermos p_1 y p_2 - proporciones poblacionales-).

- Realice una simulación en la cual genere dos poblaciones de $N_1 = 1000$ (Lote 1) y $N_2 = 1500$ (Lote 2), para los cuales se asume que el porcentaje de individuos (plantas) enfermas en ambos lotes es del 10% (es decir, sin diferencias entre los tratamientos).

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

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

```
LoteB = GenerarPob(Tamaño = 1500, Enfermos = 0.1, Sanos = 0.9)
```

```
## [1] "Lote Generado de tamaño: 1500 Enfermos: 150 Sanos: 1350"
```

- Genere una función que permita obtener una muestra aleatoria de los lotes y calcule el estimador de la proporción muestral para cada lote \hat{p}_1 y \hat{p}_2 para un tamaño de muestra dado $n_1 = n_2$. Calcule la diferencia entre los estimadores $\hat{p}_1 - \hat{p}_2$.

```
GenerarMuestra_P1P2 = function(Poblacion_1, Poblacion_2, Tamaño, Condicion) {  
  if (Tamaño <= length(Poblacion_1) || Tamaño <= length(Poblacion_2)) {  
    muestra_1 = sample(x = Poblacion_1, size = Tamaño)  
    P_muestra_1 = sum(muestra_1 == Condicion)/Tamaño  
  
    muestra_2 = sample(x = Poblacion_2, size = Tamaño)  
    P_muestra_2 = sum(muestra_2 == Condicion)/Tamaño  
  
    return(P_muestra_1 - P_muestra_2)  
  }  
  else {  
    print("Error: el tamaño de la muestra supera el tamaño de la población.")  
    return(-1)  
  }  
}  
  
n = 400  
P1P2 = GenerarMuestra_P1P2(Poblacion_1 = LoteA, Poblacion_2 = LoteB, n,  
                           Condicion = "Enfermo")  
print(paste("Para una muestra de tamaño:", n, "de ambos lotes se tiene un  $P_1 - P_2$  =", P1P2))
```

```
## [1] "Para una muestra de tamaño: 400 de ambos lotes se tiene un P1-P2 = 0"
```

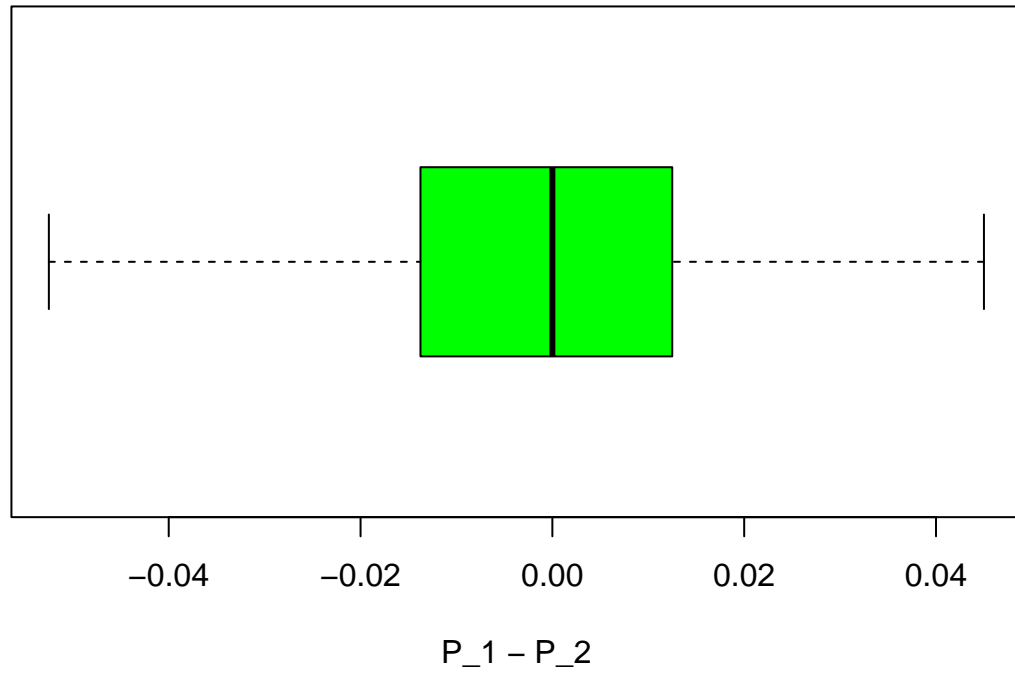
- Repita el escenario anterior (b) 500 veces y analice los resultados en cuanto al comportamiento de los 500 estimadores (diferencias $p_1 - p_2$). ¿Qué tan simétricos son los datos?, ¿Son siempre cero las diferencias?

```
GenerarMuestras_Iter_P1P2 = function(Poblacion_1, Poblacion_2, Tamaño, Condicion, Iteraciones)
{ if (Tamaño<=length(Poblacion_1)||Tamaño<=length(Poblacion_2)) {
  muestras_iter = array(NA,Iteraciones)
  for (i in 1:Iteraciones) {
    muestras_iter[i] = GenerarMuestra_P1P2(Poblacion_1, Poblacion_2, Tamaño, Condicion)
  }
  return(muestras_iter)
}
else {
  print("Error: el tamaño de la muestra supera el tamaño de la población.")
  return(-1)
}
}

iter = 500
P1P2_muestras = GenerarMuestras_Iter_P1P2 (Poblacion_1 = LoteA,
                                             Poblacion_2 = LoteB, Tamaño = n,
                                             Condicion = "Enfermo",Iteraciones = iter)

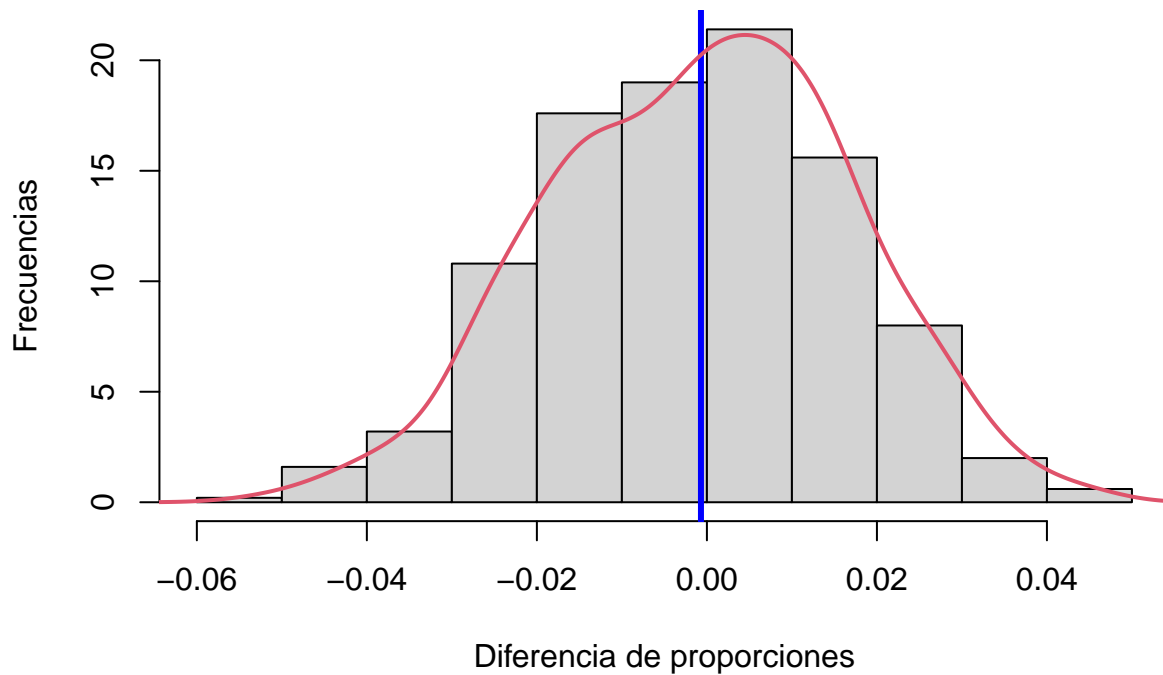
boxplot(P1P2_muestras,prob = TRUE, main = "Diferencia de proporciones",
        xlab = "P_1 - P_2", col = "green",horizontal=T)
```

Diferencia de proporciones



```
hist(P1P2_muestras,prob = TRUE, main = "Histograma con curva de densidad",  
     xlab="Diferencia de proporciones",ylab="Frecuencias")  
line = mean(P1P2_muestras)  
abline(v=line, col="blue", lwd=3)  
lines(density(P1P2_muestras), col = 2, lwd = 2)
```

Histograma con curva de densidad



```
Resultado4=data.frame("ID"=0, "Tamaño_muestra"=n, "Media"=mean(P1P2_muestras), "Mediana"=median(P1P2_muestras),
                      "Varianza"=var(P1P2_muestras), "Mín."=min(P1P2_muestras), "Máx"=max(P1P2_muestras),
                      "Curtosis"= kurtosis(P1P2_muestras))
```

Resultado4

```
##   ID Tamaño_muestra Media Mediana Desvest Varianza Mín. Máx
## 1  0             400 -7e-04      0 0.01757135 0.0003087525 -0.0525 0.045
##   Asimetría Curtosis
## 1 -0.1067814 2.666311
```

Respuestas:

- ¿Qué tan simétricos son los datos?

Para la distribución de las diferencias la asimetría es muy baja al igual que la curtosis, lo cual se puede evidenciar haciendo uso del histograma y del diagrama de cajas y bigotes.

- ¿Son siempre cero las diferencias?

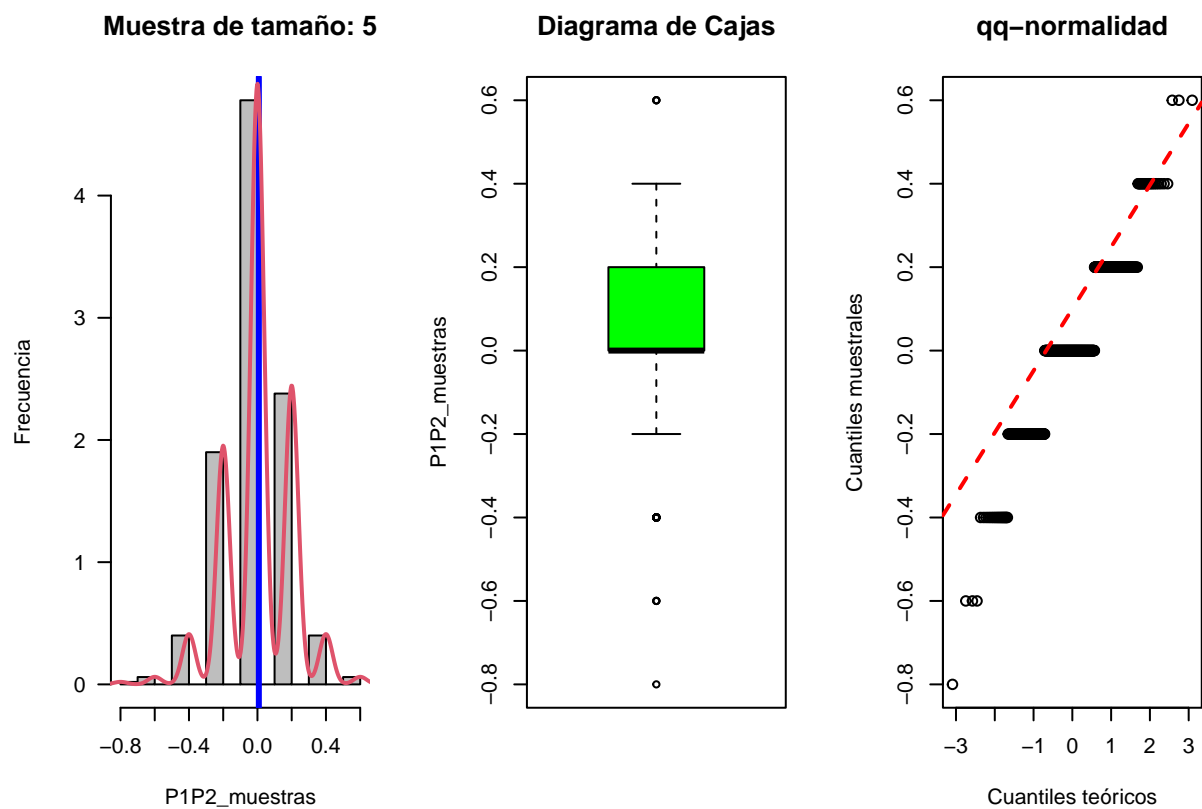
Para muestras de tamaño 400 de ambos lotes la media de las diferencias proporcionales obtenidas $\mu_{\hat{p}_1 - \hat{p}_2}$, son aproximadamente cero, sin embargo no todas las diferencias son cero y también se puede observar que la mediana de las diferencias si es cero.

- Realice los puntos b y c para tamaños de muestra $n_1 = n_2 = 5, 10, 15, 20, 30, 50, 60, 100, 200, 500$. Compare los resultados de los estimadores ($\hat{p}_1 - \hat{p}_2$) en cuanto a la normalidad. También analice el comportamiento de las diferencias y evalúe. ¿Considera que es más probable concluir que existen diferencias entre los tratamientos con muestras grandes que pequeñas, es decir, cuál considera usted que es el efecto del tamaño de muestra en el caso de la comparación de proporciones?

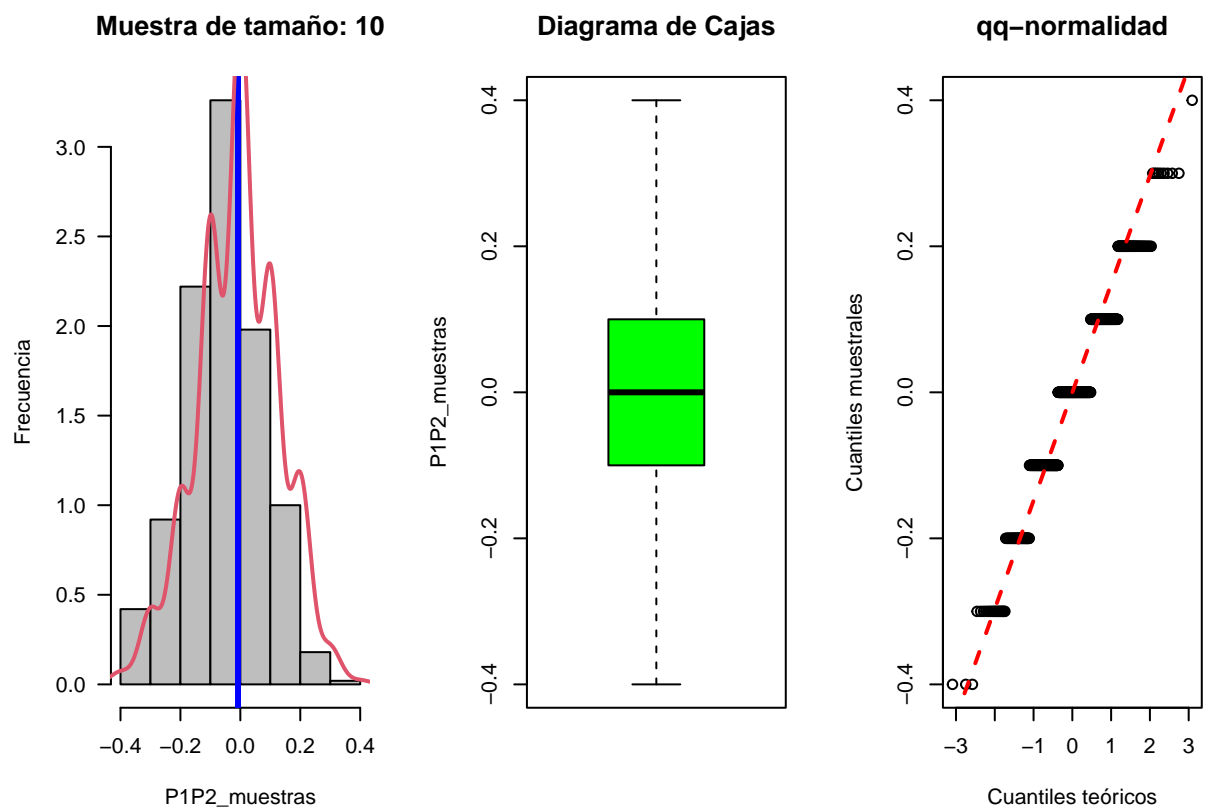
```
t_Muestras = c(5,10,15,20,30,50,60,100,200,500)
iter = 500

for (i in 1:length(t_Muestras)) {
  P1P2_muestras = GenerarMuestras_Iter_P1P2 (Poblacion_1 = LoteA, Poblacion_2 = LoteB,
                                              Tamaño = t_Muestras[i], Condicion = "Enfermo",
                                              Iteraciones = iter)

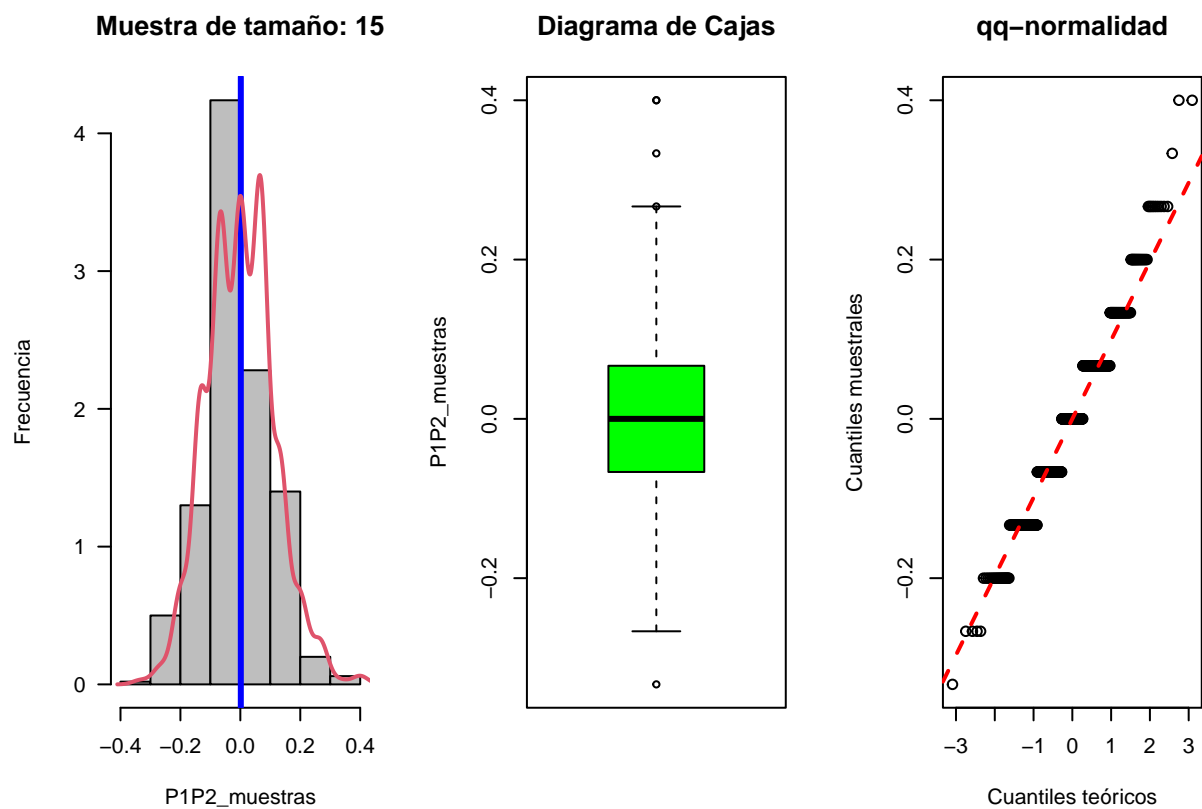
  par(mfrow=c(1,3))
  hist(P1P2_muestras, las=1, ylab = "Frecuencia", title = t_Muestras[i],
       main = paste("Muestra de tamaño:", t_Muestras[i]), col = "gray", prob = TRUE)
  abline(v=mean(P1P2_muestras), col="blue", lwd=3)
  lines(density(P1P2_muestras), col = 2, lwd = 2)
  boxplot(P1P2_muestras, xlab="", ylab="P1P2_muestras", main="Diagrama de Cajas", col = "green")
  qqnorm(P1P2_muestras, xlab="Cuantiles teóricos", ylab="Cuantiles muestrales",
         main="qq-normalidad")
  qqline(P1P2_muestras, col = 'red', lwd = 2, lty = 2)
  print(paste('Para una muestra de tamaño: ', t_Muestras[i]))
  print(shapiro.test(P1P2_muestras))
  x=data.frame("ID"=i, "Tamaño_muestra"=t_Muestras[i],
               "Media"=mean(P1P2_muestras),
               "Mediana"=median(P1P2_muestras),
               "Desvest"=sd(P1P2_muestras),
               "Varianza"=var(P1P2_muestras),
               "Mín."=min(P1P2_muestras),
               "Máx"=max(P1P2_muestras),
               "Asimetría"=skewness(P1P2_muestras),
               "Curtosis"= kurtosis(P1P2_muestras)
               )
  Resultado4=rbind(Resultado4, x)
}
```



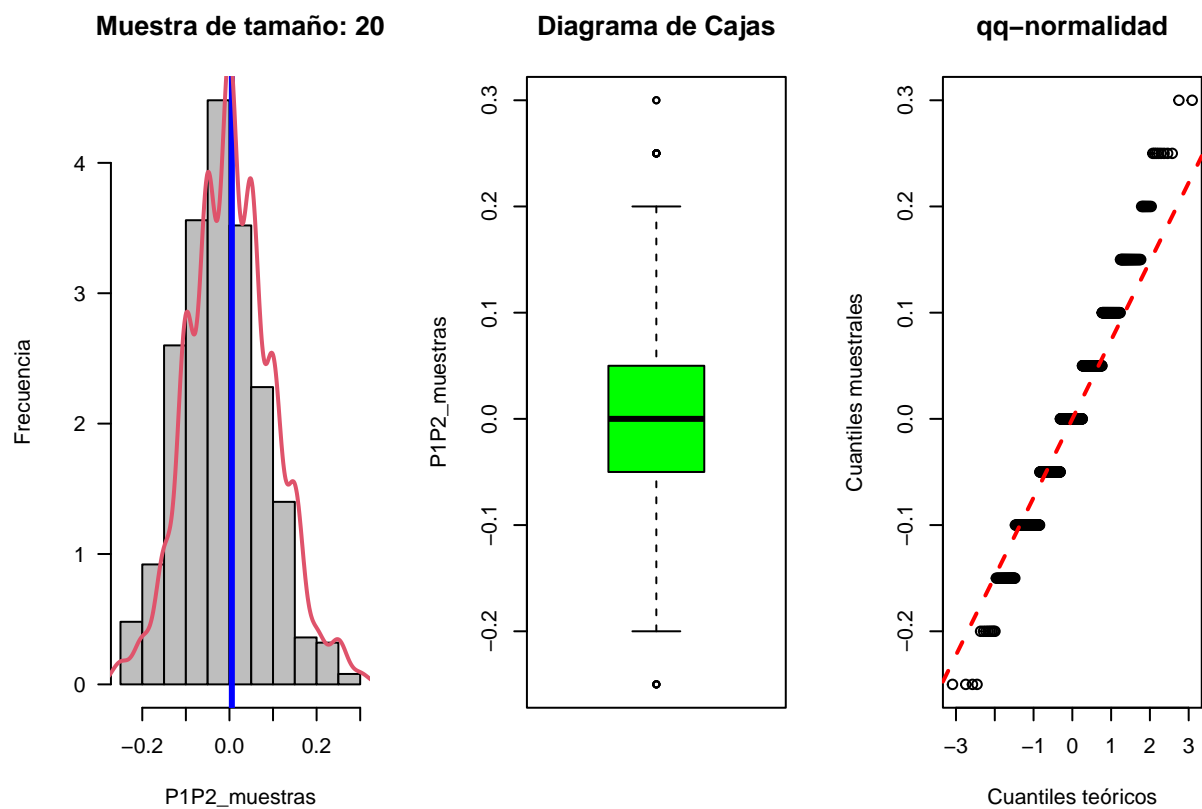
```
## [1] "Para una muestra de tamaño: 5"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.89407, p-value < 2.2e-16
```



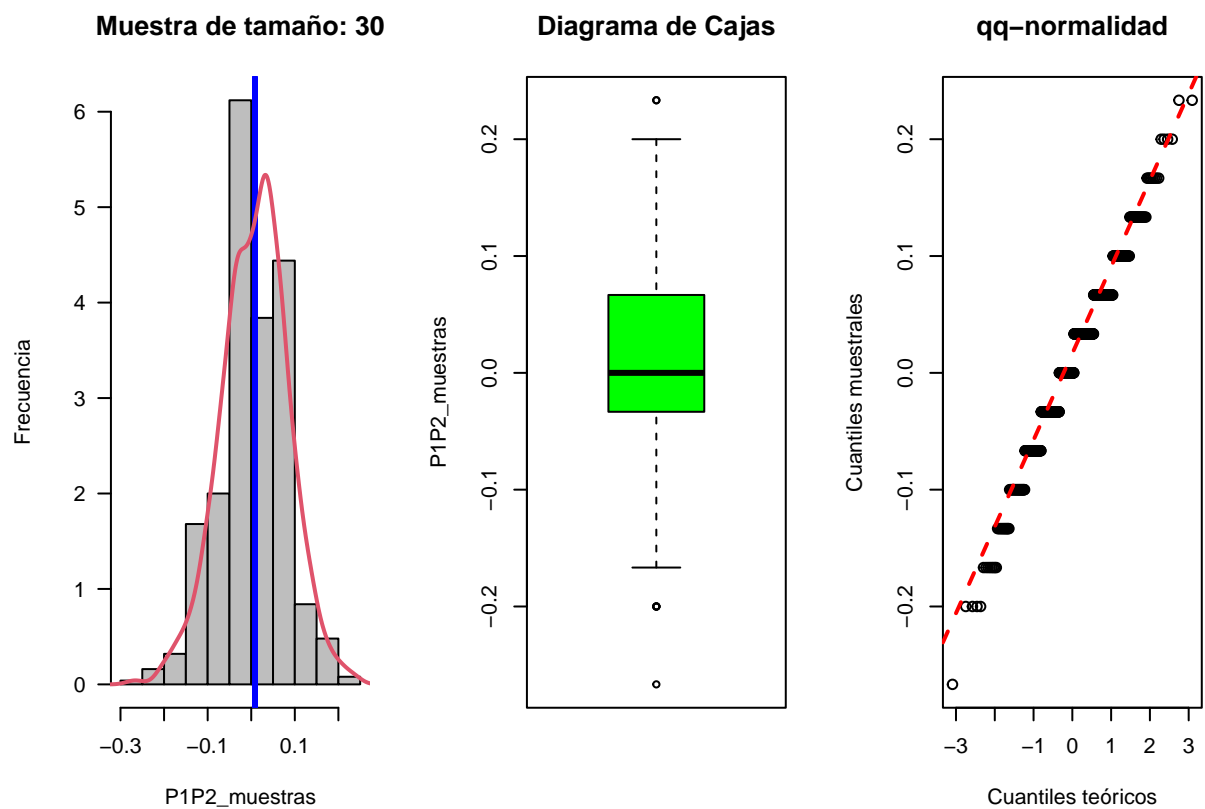
```
## [1] "Para una muestra de tamaño: 10"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.95111, p-value = 8.417e-12
```

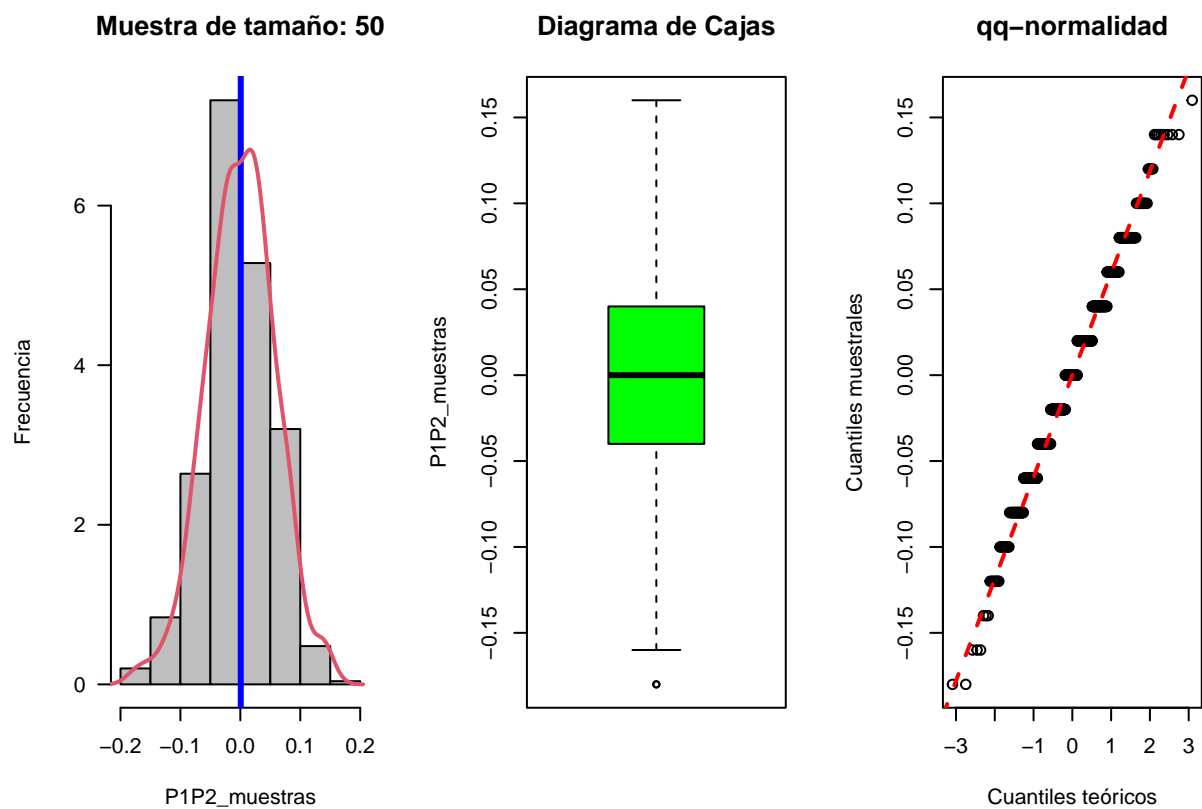
```
## [1] "Para una muestra de tamaño: 15"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.96459, p-value = 1.296e-09
```



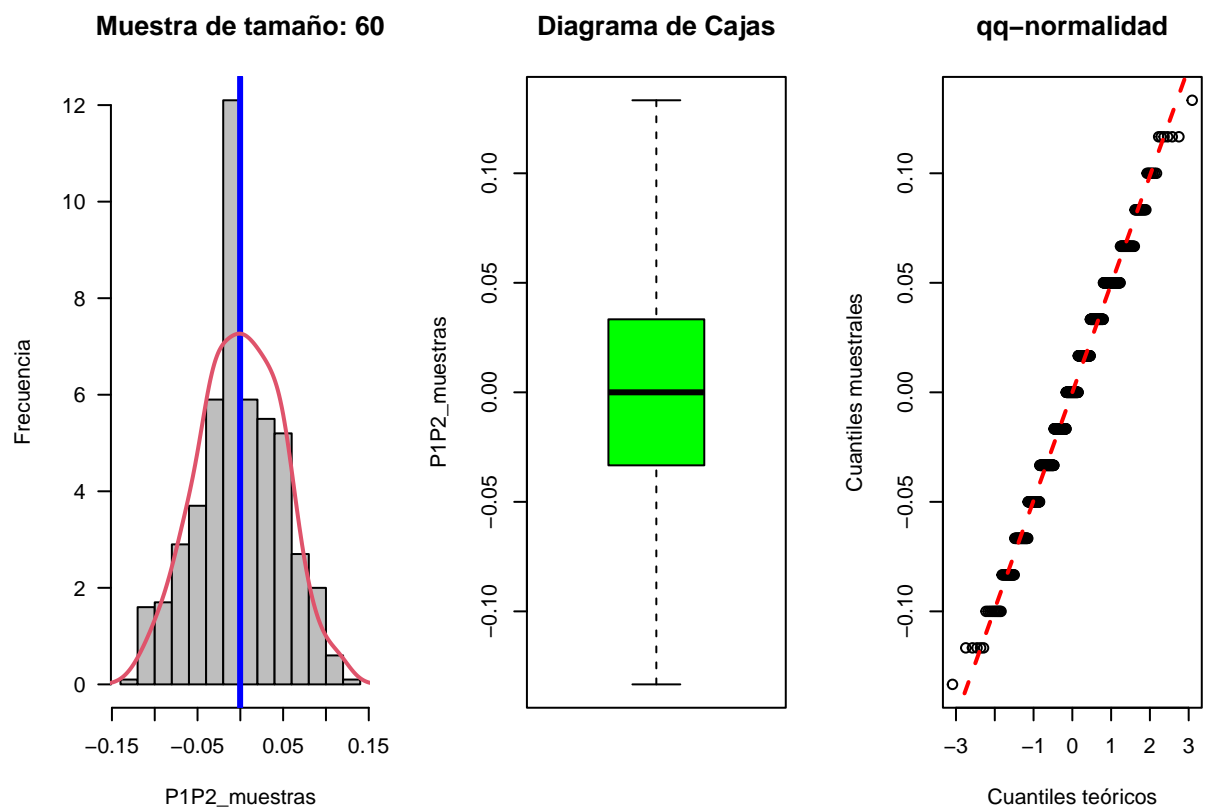
```
## [1] "Para una muestra de tamaño: 20"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.97332, p-value = 6.625e-08
```



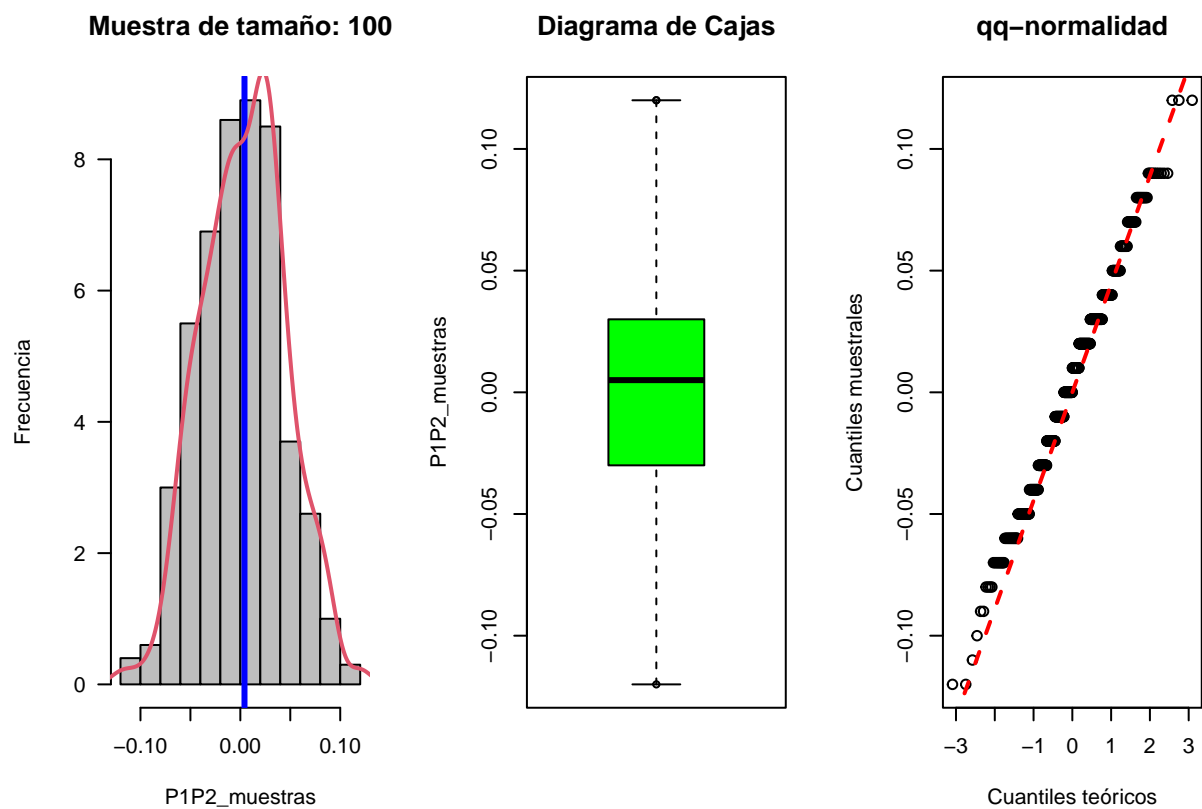
```
## [1] "Para una muestra de tamaño: 30"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.98126, p-value = 4.811e-06
```



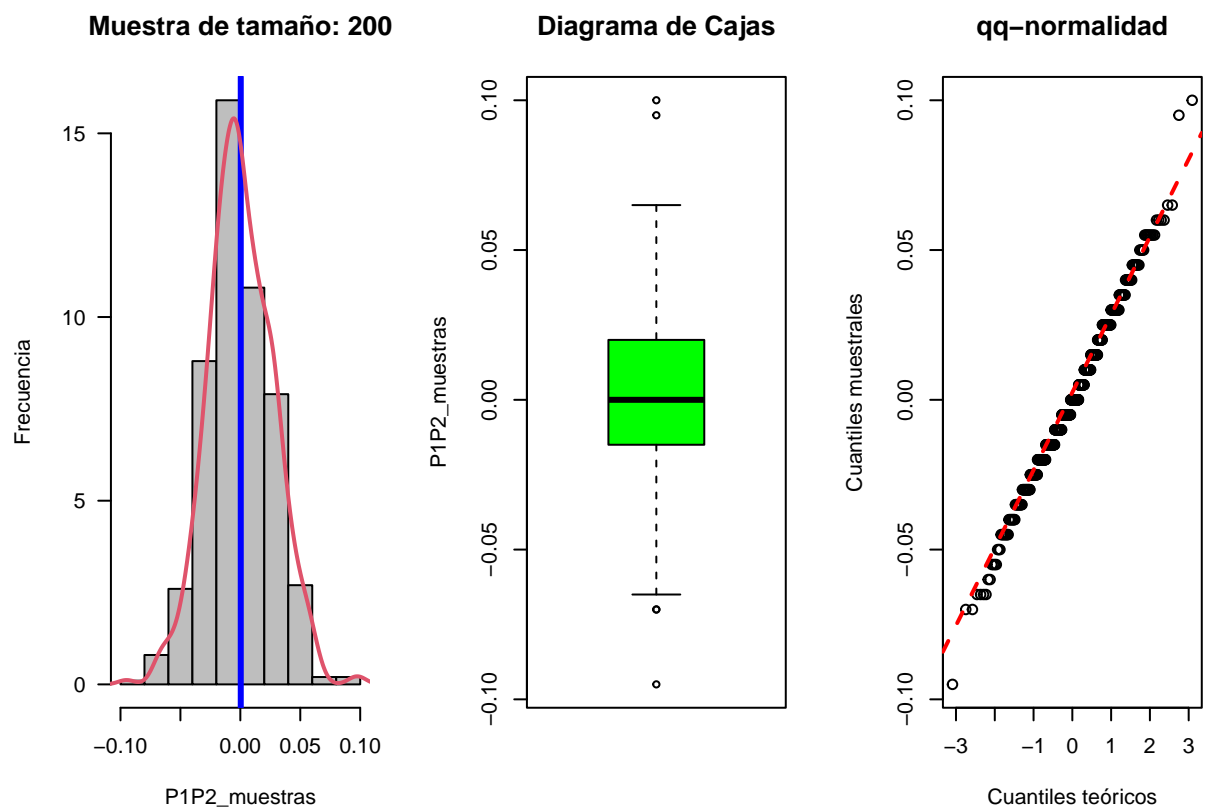
```
## [1] "Para una muestra de tamaño: 50"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.98684, p-value = 0.0001747
```



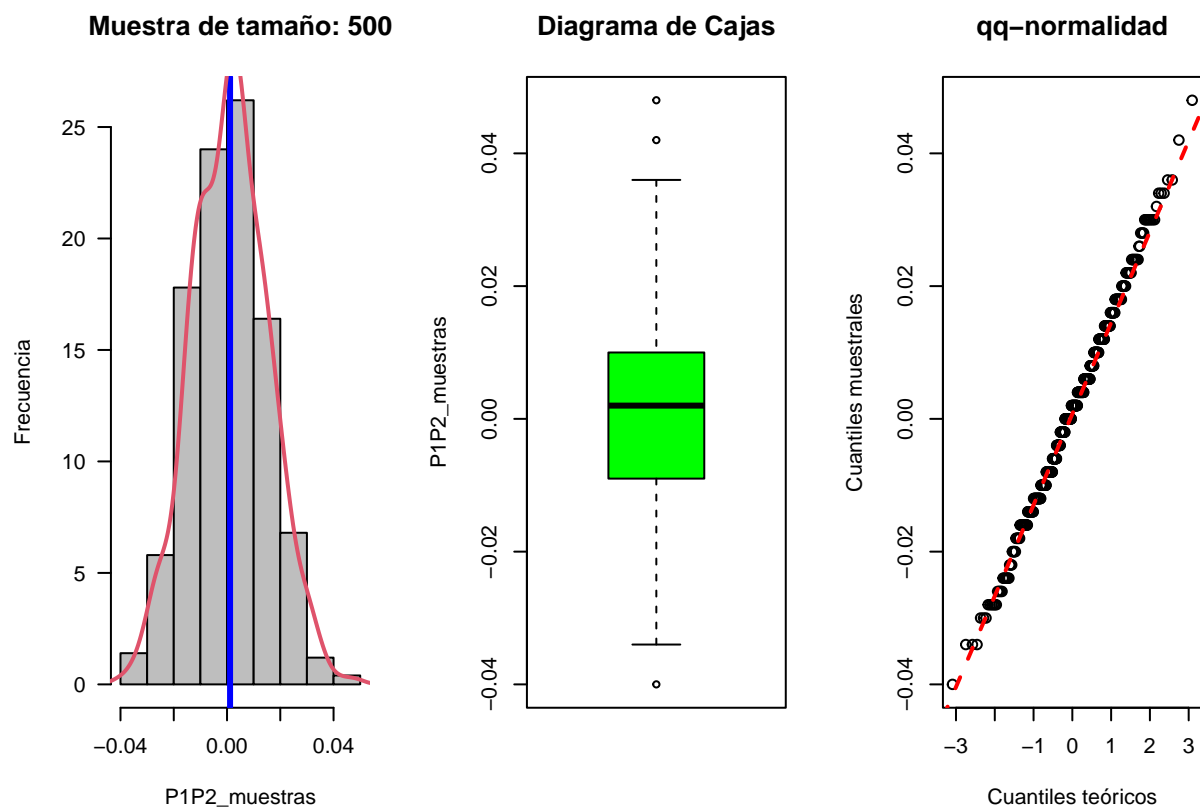
```
## [1] "Para una muestra de tamaño: 60"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.98736, p-value = 0.0002512
```



```
## [1] "Para una muestra de tamaño: 100"
##
## Shapiro-Wilk normality test
##
## data: P1P2_muestras
## W = 0.9914, p-value = 0.005324
```



```
## [1] "Para una muestra de tamaño: 200"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.99323, p-value = 0.02396
```



```
## [1] "Para una muestra de tamaño: 500"
##
## Shapiro-Wilk normality test
##
## data: P1P2_muestras
## W = 0.99669, p-value = 0.3981
```

Resultado4

##	ID	Tamaño_muestra	Media	Mediana	Desvest	Varianza	Mín.
## 1	0	400	-0.0007000000	0.000	0.01757135	0.0003087525	-0.0525000
## 2	1	5	0.0080000000	0.000	0.18848622	0.0355270541	-0.8000000
## 3	2	10	-0.0078000000	0.000	0.13414600	0.0179951503	-0.4000000
## 4	3	15	0.0016000000	0.000	0.11141534	0.0124133779	-0.3333333
## 5	4	20	0.0055000000	0.000	0.09561922	0.0091430361	-0.2500000
## 6	5	30	0.0086000000	0.000	0.07660867	0.0058688889	-0.2666667
## 7	6	50	0.0006800000	0.000	0.05751307	0.0033077531	-0.1800000
## 8	7	60	-0.0002666667	0.000	0.04912886	0.0024136451	-0.1333333
## 9	8	100	0.0041600000	0.005	0.04174422	0.0017425796	-0.1200000
## 10	9	200	0.0003400000	0.000	0.02703417	0.0007308461	-0.0950000
## 11	10	500	0.0011440000	0.002	0.01424729	0.0002029852	-0.0400000
##			Máx	Asimetría	Curtosis		
## 1			0.0450000	-0.10678142	2.666311		
## 2			0.6000000	-0.22339623	4.089725		
## 3			0.4000000	-0.11687228	3.026262		


```
## 4  0.4000000  0.24824876 3.280867
## 5  0.3000000  0.18703523 3.130442
## 6  0.2333333 -0.14866260 3.273246
## 7  0.1600000 -0.11606895 3.160487
## 8  0.1333333 -0.04447581 2.662929
## 9  0.1200000 -0.02382491 2.795456
## 10 0.1000000  0.10338256 3.375651
## 11 0.0480000  0.06219043 2.919284
```

Conclusión: Para analizar la normalidad nos fijaremos en el test de bondad de ajuste de Shapiro-Wilks el cual plantea lo siguiente:

Hipótesis Nula

$H_0 : F(x) = F_0(x)$ las diferencias se ajustan a una distribución norma.

Hipótesis Alternativa

$H_a : F(x) \neq F_0(x)$ las diferencias **NO** se ajustan a una distribución normal.

Eligiendo un nivel de significanza, de 0.05, cuando los tamaños de muestra son grandes, los resultados obtenidos nos arrojan P-valores mayores a 0.05, lo cual nos conduce a no rechazar la hipótesis nula (Distribución normal), en cuanto al sesgamiento y apuntamiento de la distribución de las diferencias de las proporciones también se observa bastante mejoría a nivel que los tamaños de las muestras aumentan y se puede visualizar a través de los diagramas de cajas y bigotes, histogramas y gráficos qq de normalidad.

- Ahora realice nuevamente los puntos a-d bajo un escenario con dos lotes, pero de proporciones de enfermos diferentes ($p_1 = 0.1$ y $p_2 = 0.15$), es decir, el tratamiento del lote 1 si presentó un mejor desempeño reduciendo en un 5% el porcentaje de enfermos. Bajo este nuevo escenario compare la distribución de estas diferencias ($\hat{p}_1 - \hat{p}_2$) con las observadas bajo igualdad de condiciones en los lotes. ¿Qué puede concluir? ¿Existen puntos en los cuales es posible que se observen diferencias de $(p_1 - p_2)$. bajo ambos escenarios (escenario 1: sin diferencias entre p_1 y p_2 , escenario 2: diferencia de 5%)?

```
LoteA2 = GenerarPob(Tamaño = 1000, Enfermos = 0.1, Sanos = 0.9)
```

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

```
LoteB2 = GenerarPob(Tamaño = 1500, Enfermos = 0.15, Sanos = 0.85)
```

```
## [1] "Lote Generado de tamaño: 1500 Enfermos: 225 Sanos: 1275"
```

```
n = 300
P1P2_2 = GenerarMuestra_P1P2(Poblacion_1 = LoteA2,
                              Poblacion_2 = LoteB2, n, Condicion = "Enfermo")
print(paste("Para una muestra de tamaño:", n, "de ambos lotes se tiene un P1-P2 =", P1P2_2))
```

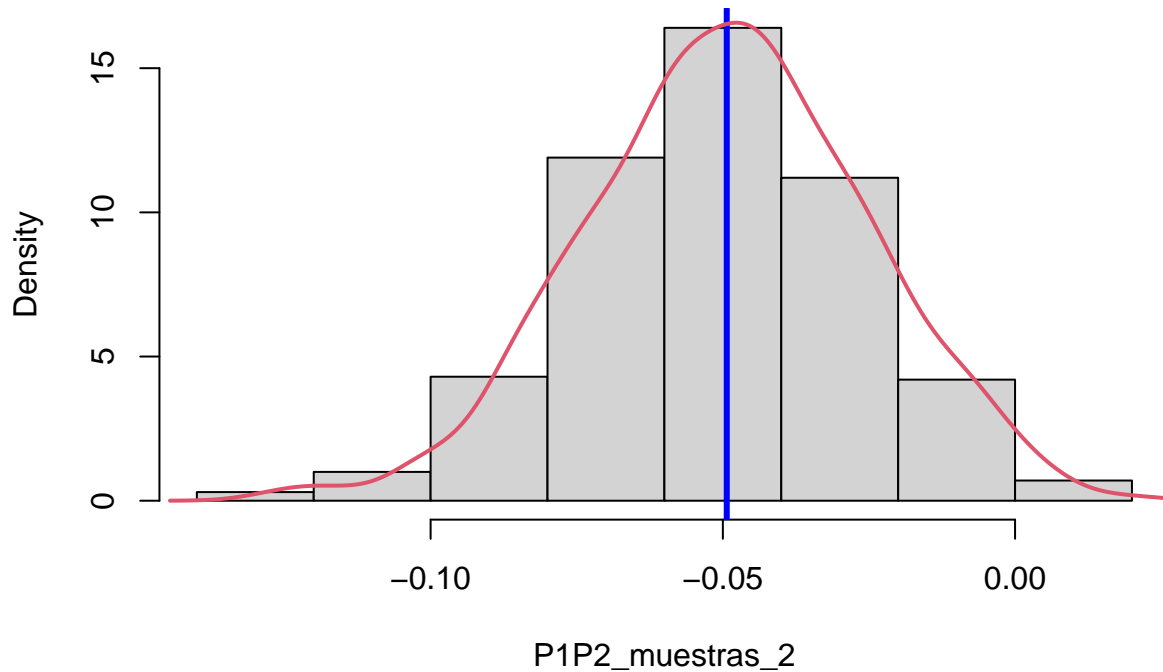
```
## [1] "Para una muestra de tamaño: 300 de ambos lotes se tiene un P1-P2 = -0.07666666666666667"
```

```
iter = 500
```

```
P1P2_muestras_2 = GenerarMuestras_Iter_P1P2 (Poblacion_1 = LoteA2, Poblacion_2 = LoteB2,
Tamaño = n, Condicion = "Enfermo", Iteraciones = iter)
```

```
hist(P1P2_muestras_2, prob= TRUE)
line = mean(P1P2_muestras_2)
abline(v=line, col="blue", lwd=3)
lines(density(P1P2_muestras_2), col = 2, lwd = 2)
```

Histogram of P1P2_muestras_2



```
Resultado5=data.frame("ID"=0,"Tamaño_muestra"=n, "Media"=mean(P1P2_muestras_2),"Mediana"=median(P1P2_mu
"Desvest"=sd(P1P2_muestras_2),
"Varianza"=var(P1P2_muestras_2),
"Mín."=min(P1P2_muestras_2), "Máx"=max(P1P2_muestras_2),"Asimetría"=skewness(P1P2
"Curtosis"= kurtosis(P1P2_muestras_2))
```

Resultado5

```
##   ID Tamaño_muestra      Media Mediana  Desvest    Varianza      Mín. Máx
## 1  0              300 -0.04933333  -0.05 0.02406108 0.0005789356 -0.1266667 0.02
##   Asimetría Curtosis
## 1 -0.1290638 3.020472
```

```
t_Muestras = c(5,10,15,20,30,50,60,100,200,500)
iter = 500

for (i in 1:length(t_Muestras)) {
  P1P2_muestras = GenerarMuestras_Iter_P1P2 (Poblacion_1 = LoteA2,
                                              Poblacion_2 = LoteB2, Tamaño = t_Muestras[i],
```

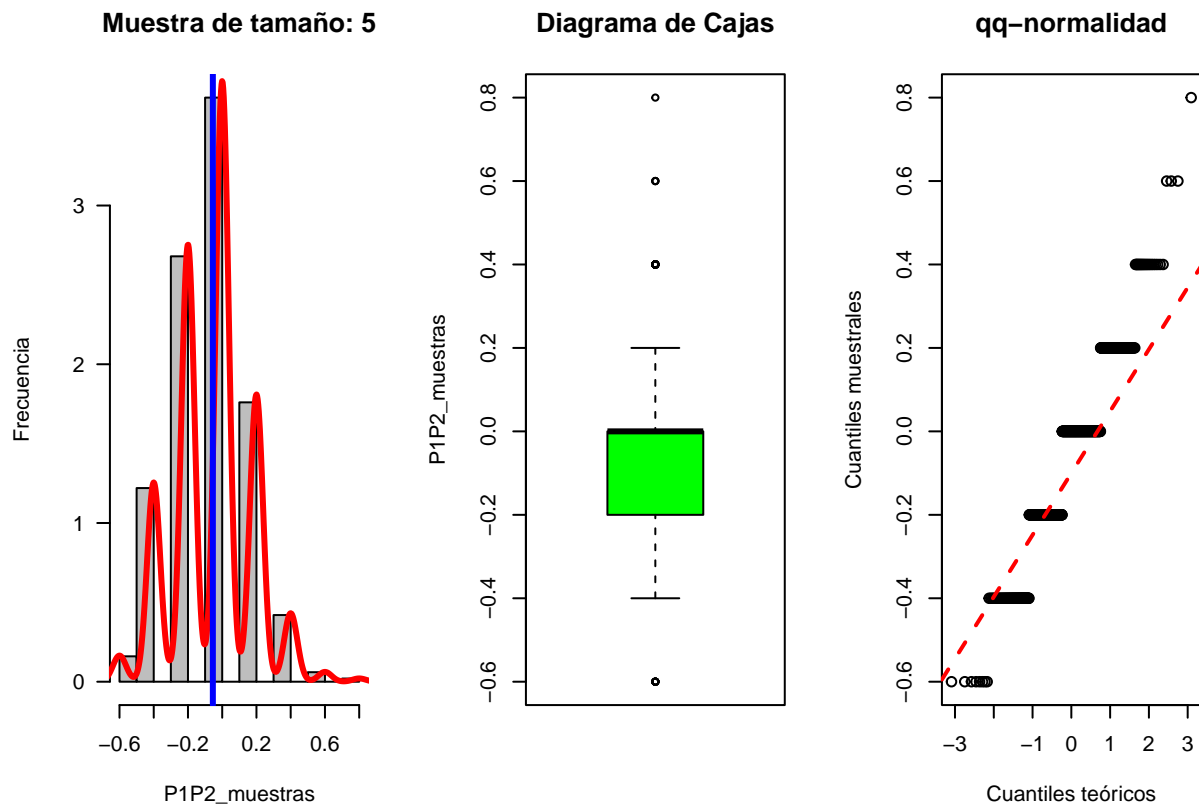
```

Condicion = "Enfermo", Iteraciones = iter)

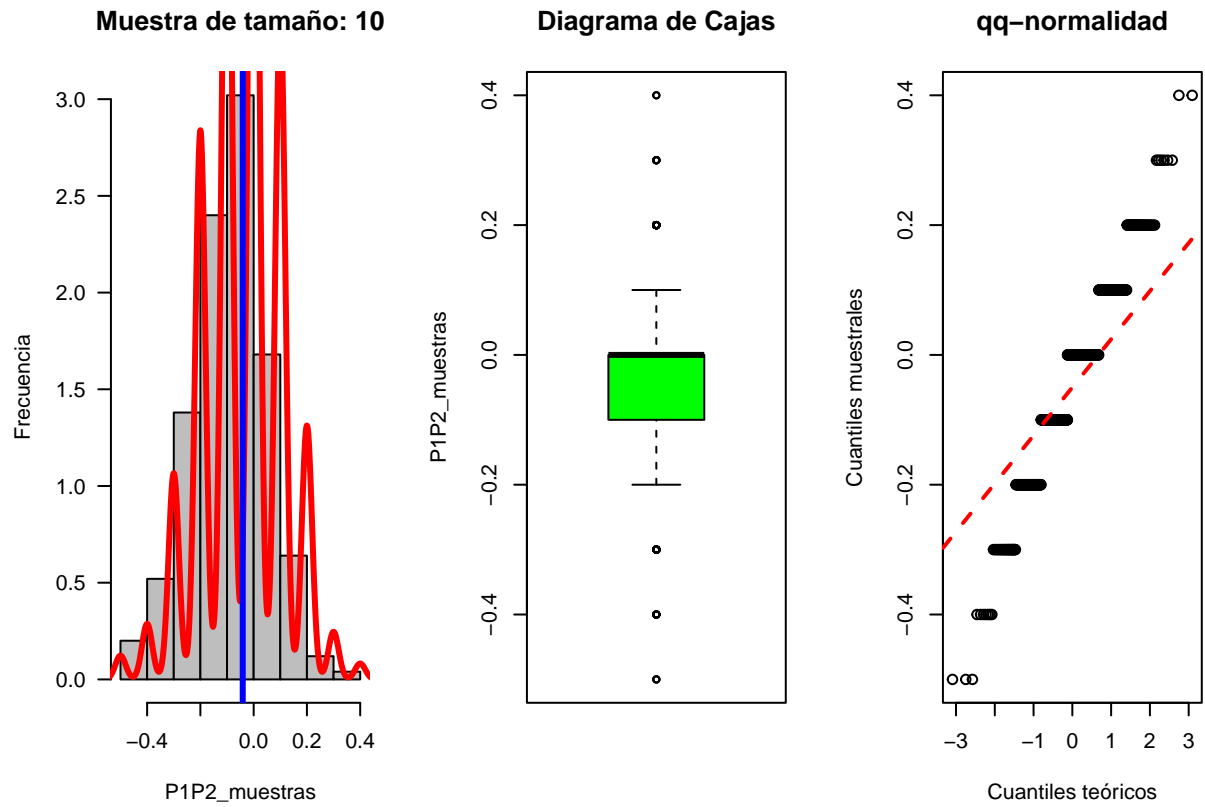
par(mfrow=c(1,3))
hist(P1P2_muestras, las=1, ylab = "Frecuencia", title = t_Muestras[i],
     prob = TRUE, main = paste("Muestra de tamaño:", t_Muestras[i]), col = "gray")
lines(density(P1P2_muestras), col="red", lwd=3)
abline(v=mean(P1P2_muestras), col="blue", lwd=3)
boxplot(P1P2_muestras, xlab="", ylab="P1P2_muestras", main="Diagrama de Cajas", col = "green")
qqnorm(P1P2_muestras, xlab="Cuantiles teóricos", ylab="Cuantiles muestrales", main="qq-normalidad")
qqline(P1P2_muestras, col = 'red', lwd = 2, lty = 2)
print(paste('Para una muestra de tamaño: ', t_Muestras[i]))
print(shapiro.test(P1P2_muestras))

x=data.frame("ID"=i, "Tamaño_muestra"=t_Muestras[i],
             "Media"=mean(P1P2_muestras),
             "Mediana"=median(P1P2_muestras),
             "Desvest"=sd(P1P2_muestras),
             "Varianza"=var(P1P2_muestras),
             "Mín."=min(P1P2_muestras),
             "Máx"=max(P1P2_muestras),
             "Asimetría"=skewness(P1P2_muestras),
             "Curtosis"= kurtosis(P1P2_muestras)
            )
Resultado5=rbind(RResultado5,x)
}

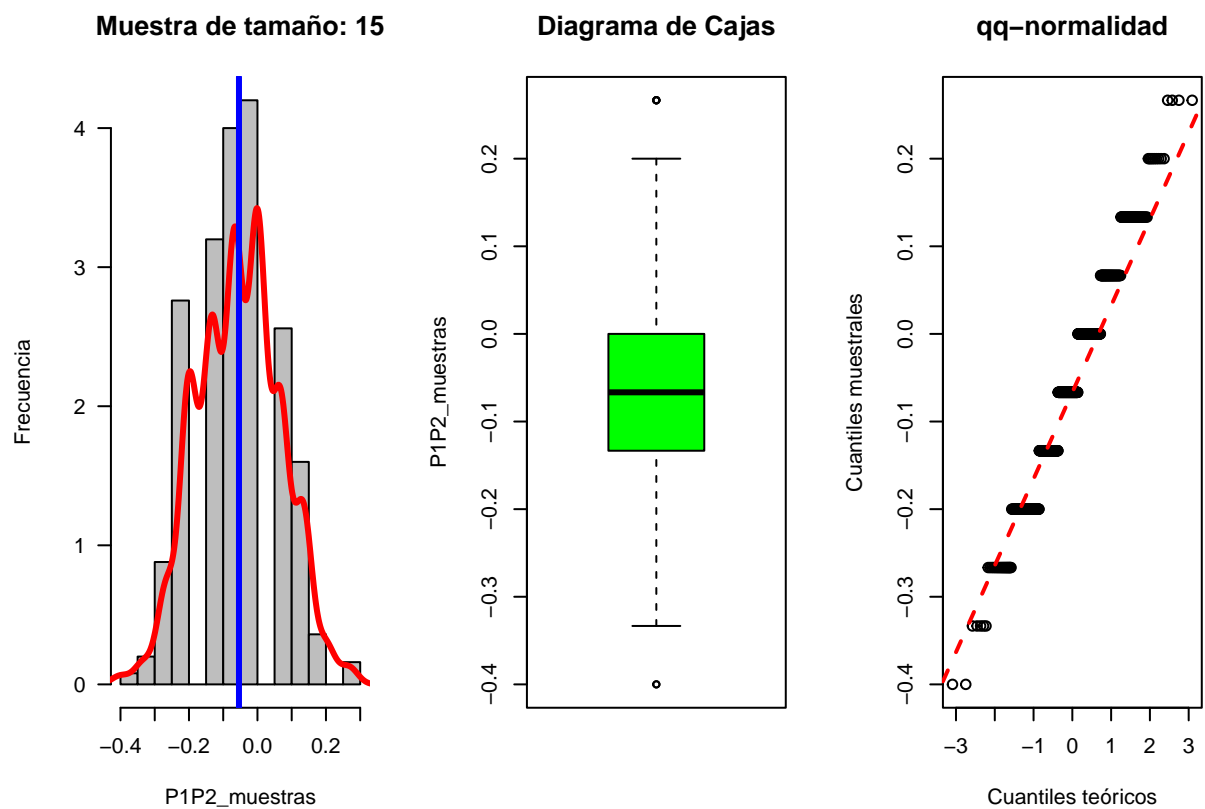
```



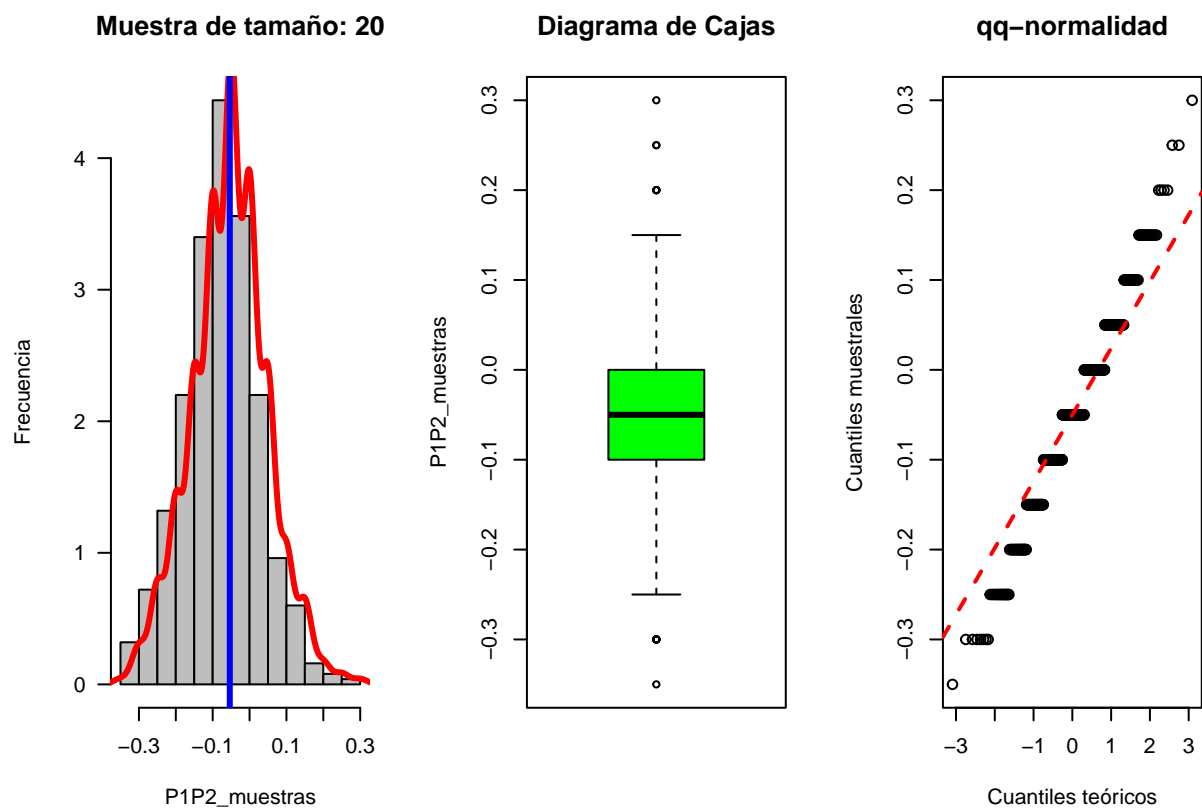
```
## [1] "Para una muestra de tamaño: 5"
##
## Shapiro-Wilk normality test
##
## data: P1P2_muestras
## W = 0.93029, p-value = 1.679e-14
```



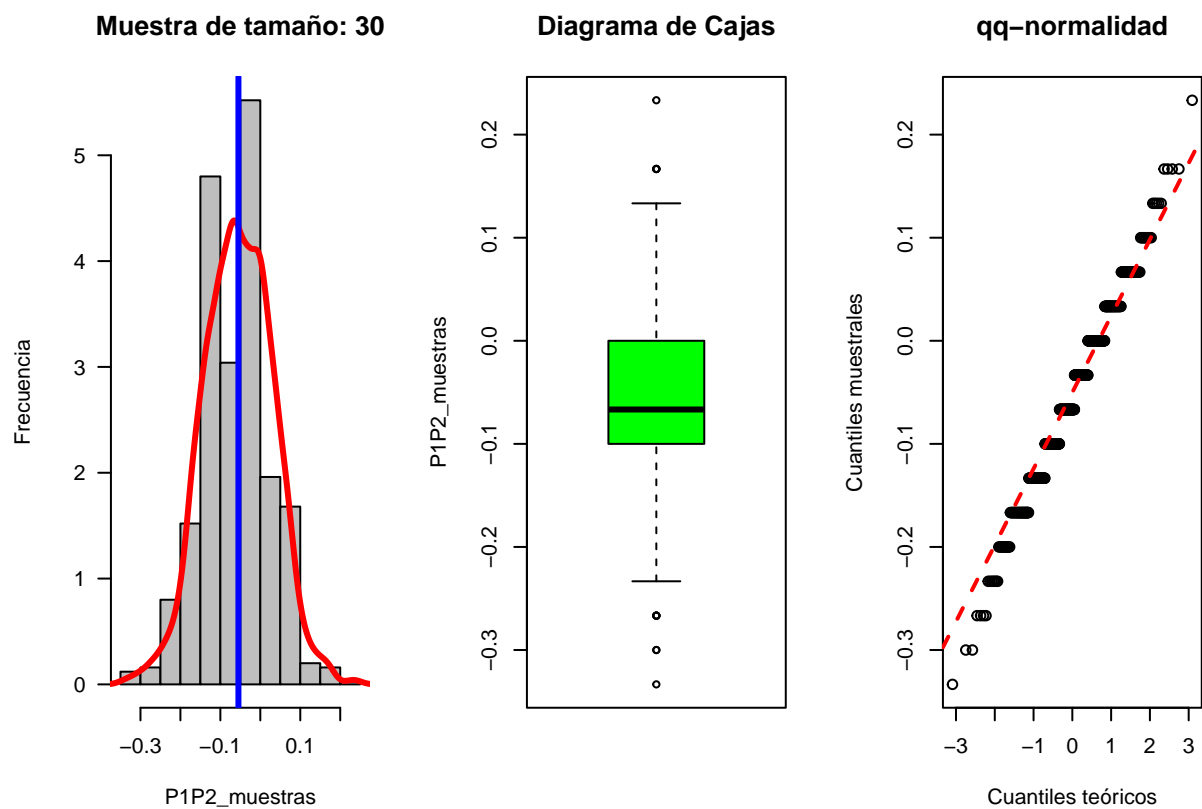
```
## [1] "Para una muestra de tamaño: 10"
##
## Shapiro-Wilk normality test
##
## data: P1P2_muestras
## W = 0.95573, p-value = 4.229e-11
```



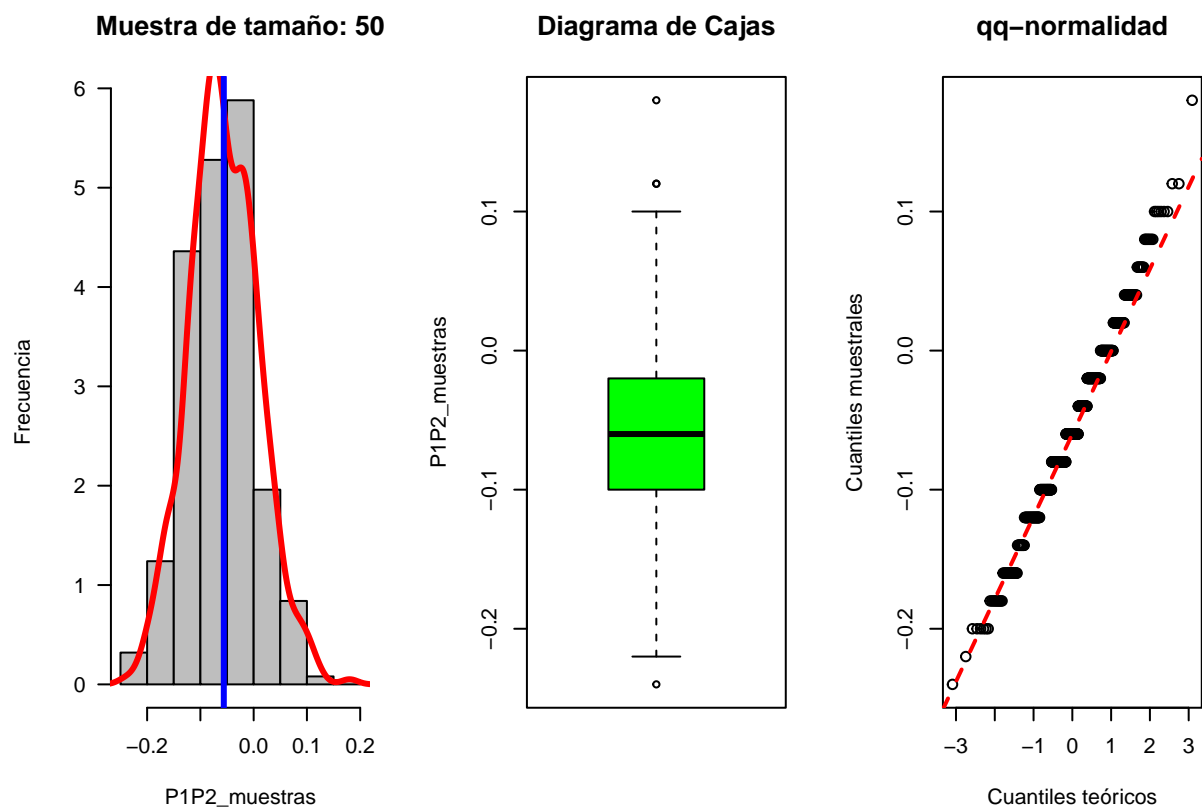
```
## [1] "Para una muestra de tamaño: 15"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.9712, p-value = 2.399e-08
```



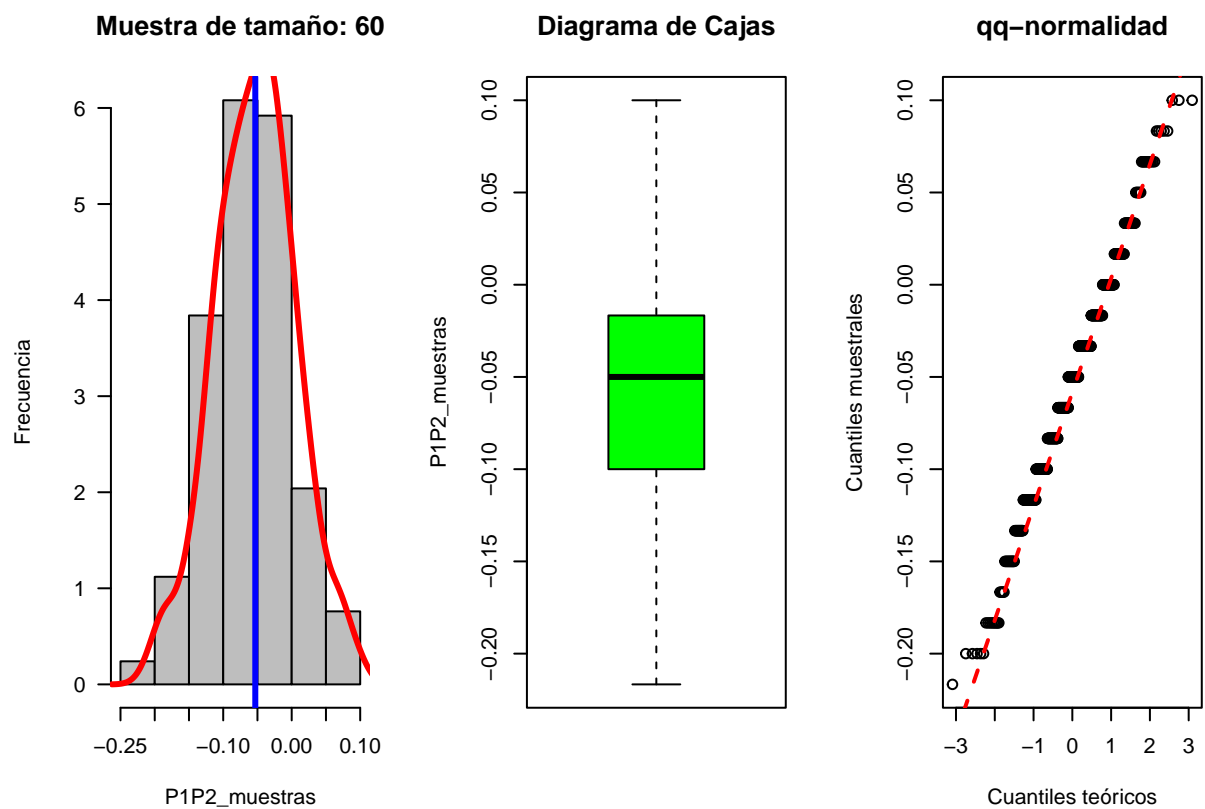
```
## [1] "Para una muestra de tamaño: 20"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.97721, p-value = 4.881e-07
```



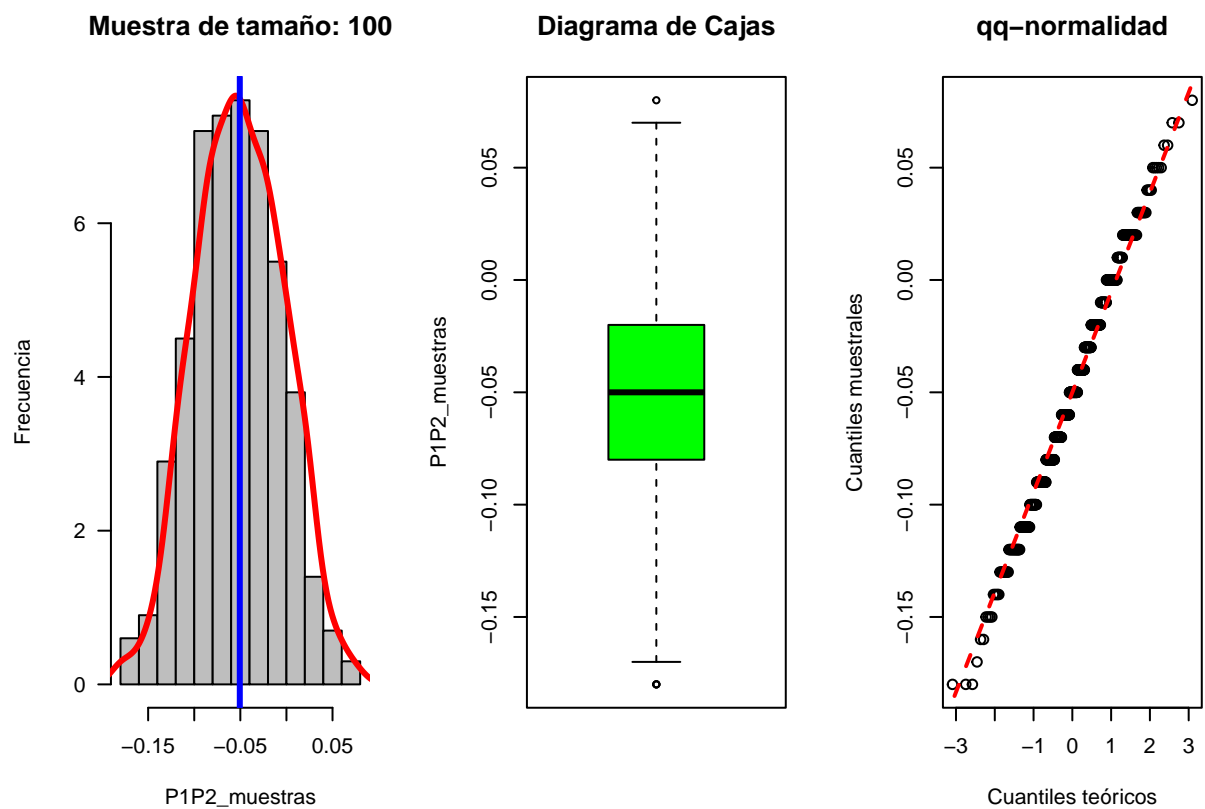
```
## [1] "Para una muestra de tamaño: 30"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.98544, p-value = 6.706e-05
```



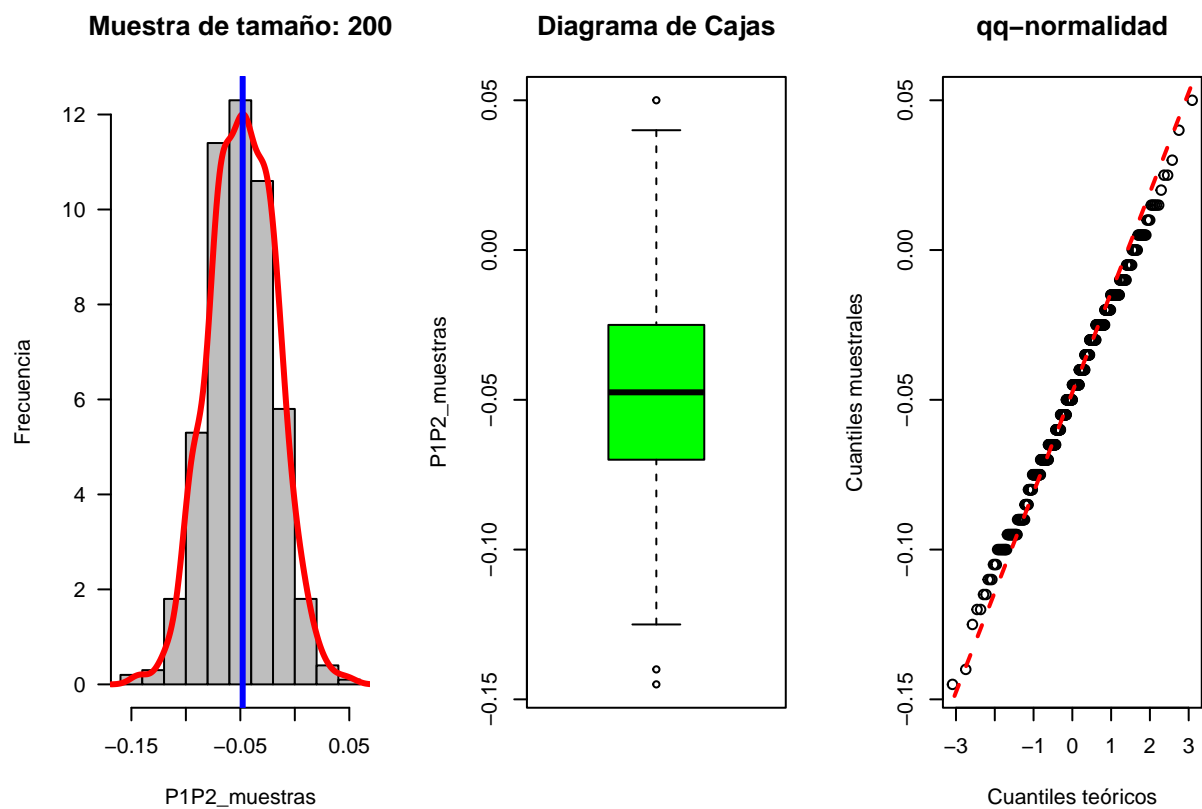
```
## [1] "Para una muestra de tamaño: 50"
##
##  Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.98893, p-value = 0.0007881
```

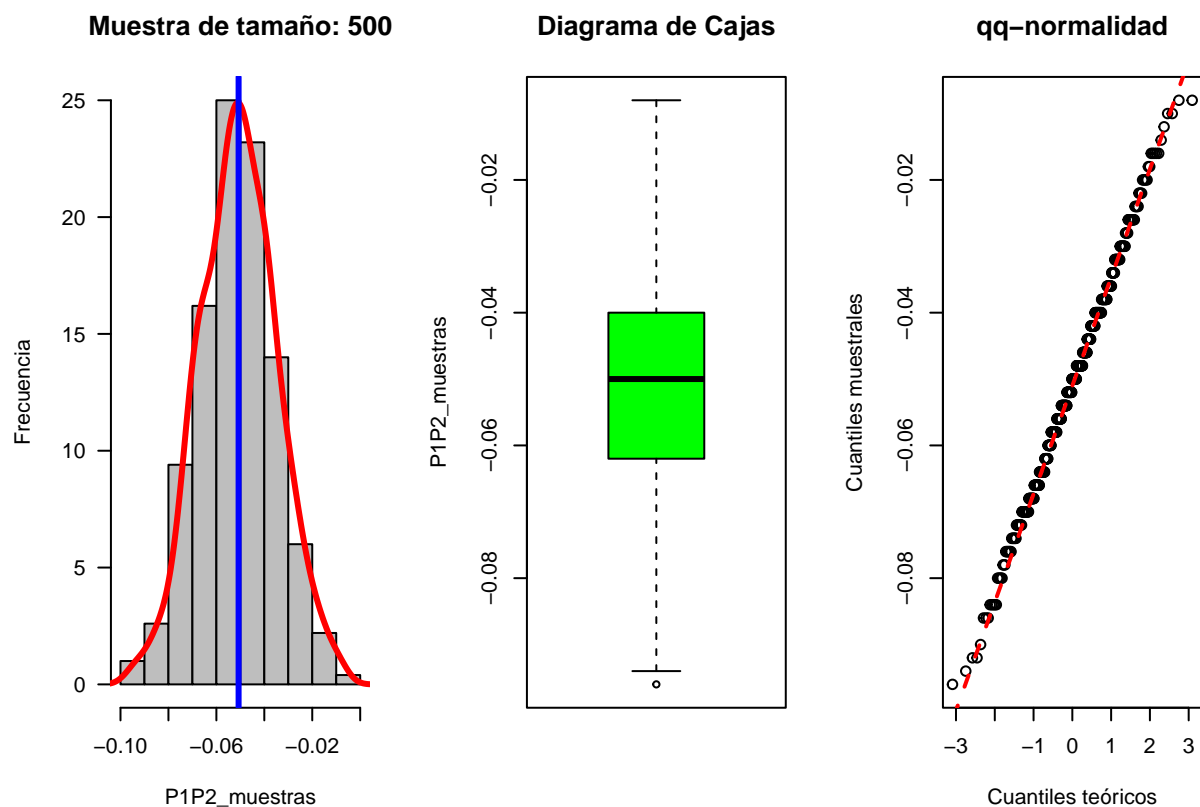
```
## [1] "Para una muestra de tamaño: 60"
##
## Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.99047, p-value = 0.002557
```



```
## [1] "Para una muestra de tamaño: 100"
##
## Shapiro-Wilk normality test
##
## data: P1P2_muestras
## W = 0.99301, p-value = 0.01994
```



```
## [1] "Para una muestra de tamaño: 200"
##
## Shapiro-Wilk normality test
##
## data: P1P2_muestras
## W = 0.99581, p-value = 0.2044
```



```
## [1] "Para una muestra de tamaño: 500"
##
## Shapiro-Wilk normality test
##
## data:  P1P2_muestras
## W = 0.99671, p-value = 0.4032
```

Resultado5

##	ID	Tamaño_muestra	Media	Mediana	Desvest	Varianza	Mín.
## 1	0	300	-0.04933333	-0.05000000	0.02406108	0.0005789356	-0.1266667
## 2	1	5	-0.05480000	0.00000000	0.22427116	0.0502975551	-0.6000000
## 3	2	10	-0.04100000	0.00000000	0.14373698	0.0206603206	-0.5000000
## 4	3	15	-0.05400000	-0.06666667	0.12052546	0.0145263861	-0.4000000
## 5	4	20	-0.05370000	-0.05000000	0.10218706	0.0104421944	-0.3500000
## 6	5	30	-0.05473333	-0.06666667	0.08523767	0.0072654598	-0.3333333
## 7	6	50	-0.05604000	-0.06000000	0.06449681	0.0041598381	-0.2400000
## 8	7	60	-0.05316667	-0.05000000	0.05909278	0.0034919561	-0.2166667
## 9	8	100	-0.05058000	-0.05000000	0.04754771	0.0022607852	-0.1800000
## 10	9	200	-0.04785000	-0.04750000	0.03028633	0.0009172620	-0.1450000
## 11	10	500	-0.05072800	-0.05000000	0.01595930	0.0002546994	-0.0960000

##	Máx	Asimetría	Curtosis
## 1	0.0200000	-0.1290638014	3.020472
## 2	0.8000000	0.1267017258	3.166234
## 3	0.4000000	-0.1786514951	3.341614

```
## 4    0.2666667  0.0149133547 2.684532
## 5    0.3000000  0.0507372392 3.240959
## 6    0.2333333 -0.0468264598 3.104671
## 7    0.1800000  0.1604604870 3.094316
## 8    0.1000000 -0.0416757180 2.919853
## 9    0.0800000 -0.0004096768 2.652110
## 10   0.0500000 -0.0207492333 2.948259
## 11  -0.0080000  0.0251689991 2.878757
```

• Análisis

En ambas situaciones hacen que el $\hat{p}_1 - \hat{p}_2$ converja a $P_1 - P_2$. Es de notar que en el segundo caso como en el escenario B se presenta un mejor desempeño en el lote A del 5%. $\hat{P}_1 - \hat{P}_2$ converjan a esta diferencia -0.05 lo que nos indicaría de manera general que si hay un mejoramiento diferencial X entre las poblaciones, las diferencias en muestras de ambos lotes $P_1 - P_2$ tenderían a ser X.

Con el ejercicio anterior podemos verificar lo siguiente:

- $\mu_{\hat{p}_1 - \hat{p}_2} = E(\hat{p}_1 - \hat{p}_2) = E(\hat{p}_1) - E(\hat{p}_2) = P_1 - P_2$.
- $\sigma_{\hat{p}_1 - \hat{p}_2}^2 = \frac{P_1(1-P_1)}{n_1} + \frac{P_2(1-P_2)}{n_2}$

y la distribución de la diferencia de las proporciones siguen una distribución normal con media $P_1 - P_2$ y desviación estándar $\sqrt{\frac{P_1(1-P_1)}{n_1} + \frac{P_2(1-P_2)}{n_2}}$, por lo tanto $z = \frac{(\hat{p}_1 - \hat{p}_2) - (P_1 - P_2)}{\sqrt{\frac{P_1(1-P_1)}{n_1} + \frac{P_2(1-P_2)}{n_2}}}$ sigue una distribución normal estándar.