EP07 - Métodos No Paramétricos

Equipo 8

2024-10-29

Introducción

Como equipo número 8, se nos pidió realizar un ejercicio sobre métodos no paramétricos para enfrentar datos numéricos problemáticos.

Enunciado

En el trabajo de título de una estudiante del DIINF se reportan tiempos de ejecución (en milisegundos) y la cercanía con la solución óptima (en porcentaje) de la mejor solución encontrada con tres versiones de un algoritmo genético para resolver instancias del problema del vendedor viajero disponibles en repositorios públicos. Ahora debe enfrentar el análisis de estos datos, por lo que está solicitando ayuda de las y los estudiantes de Estadística Inferencial.

Obtenemos los datos proporcionados mediante el archivo CSV.

```r
# Leemos los datos del CSV

datos <- read.csv("EP07 Datos.csv")

# Mostramos los datos iniciales
head(datos)
```

```
##   instancia n.nodos n.aristas tiempo.A tiempo.B tiempo.C mejor.A mejor.B
## 1         1      50       631   107534   452595   257485   98.72   98.25
## 2         2      50       521    74808   364061   207297   98.99   99.17
## 3         3      50       588    94072   417798   237793   99.10   99.23
## 4         4      50       653   114830   470701   267598   98.69   99.23
## 5         5      50       597    96720   425233   241770   99.80   99.22
## 6         6      50       564    86688   398448   226833   99.19   99.15
##   mejor.C
## 1   99.34
## 2   99.48
## 3   99.10
## 4   97.82
## 5   98.14
## 6   98.04
```

Para este enunciado, junto con los datos entregados, se nos entregaron las siguientes preguntas a responder:

Pregunta 1

Observando los datos, la memorista sospecha que hay diferencias significativas en el tiempo de ejecución entre las versiones A y B del algoritmo cuando las instancias tienen 70 o más nodos. ¿Los datos respaldan la intuición de la memorista? Para responder, filtren los datos para tener las instancias con 70 o más nodos y seleccionen las columnas de los tiempos de ejecución de las versiones A y B (en formato ancho). Usando como semilla el valor 73, obtengan muestras aleatorias independientes de 24 tiempos registrados por la versión A y 20 tiempos registrados por la versión B del algoritmo. Realicen un análisis estadístico pertinente (enunciar hipótesis, revisar condiciones, seleccionar pruebas ómnibus y post-hoc según corresponda) para responder la pregunta planteada, utilizando pruebas no paramétricas de ser necesario.

```r
# Cargamos las funciones necesarias
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Filtramos los datos para tener las instancias con 70 o más nodos

datos_filtradosEJ1 <- datos %>% filter(n.nodos >= 70)

# Mostramos los primeros datos
head(datos_filtradosEJ1)
```

```
##   instancia n.nodos n.aristas tiempo.A tiempo.B tiempo.C mejor.A mejor.B
## 1        41      70      1212   382143   947924   536924   99.36   97.38
## 2        42      70      1232   394395   965334   547148   98.77   97.18
## 3        43      70      1105   318794   854179   484068   98.58   98.01
## 4        44      70      1260   411999   990491   560973   98.25   98.71
## 5        45      70      1254   408354   985163   557996   98.99   98.79
## 6        46      70      1147   343080   891077   505090   99.42   96.36
##   mejor.C
## 1   97.43
## 2   97.89
## 3   96.20
## 4   97.64
## 5   98.95
## 6   98.95
```

Ahora debemos obtener solamente los datos de tiempo de ejecución de las versiones A y B, para luego realizar el análisis estadístico pertinente.

```r
# Seleccionamos los tiempos de ejecución A y B de las instancias que tienen 70 o más nodos

TiempoA <- datos_filtradosEJ1$tiempo.A
TiempoB <- datos_filtradosEJ1$tiempo.B

# Mostramos los datos filtrados
Tiempos_1 <- data.frame(TiempoA, TiempoB)
head(Tiempos_1)
```

```
##   TiempoA TiempoB
## 1  382143  947924
## 2  394395  965334
## 3  318794  854179
## 4  411999  990491
## 5  408354  985163
## 6  343080  891077
```

Formulación de Hipótesis

H0: No existe una diferencia significativa en el tiempo de ejecución entre las versiones A y B del algoritmo cuando las instancias tienen 70 o más nodos.

Ha: Existe una diferencia significativa en el tiempo de ejecución entre las versiones A y B del algoritmo cuando las instancias tienen 70 o más nodos.

Revisión de Condiciones

Como queremos comparar dos muestras independientes, debemos verificar que las muestras provengan de una población con distribución normal. Para esto, utilizaremos el test de Shapiro-Wilk junto a un gráfico Q-Q para verificar la normalidad de los datos.

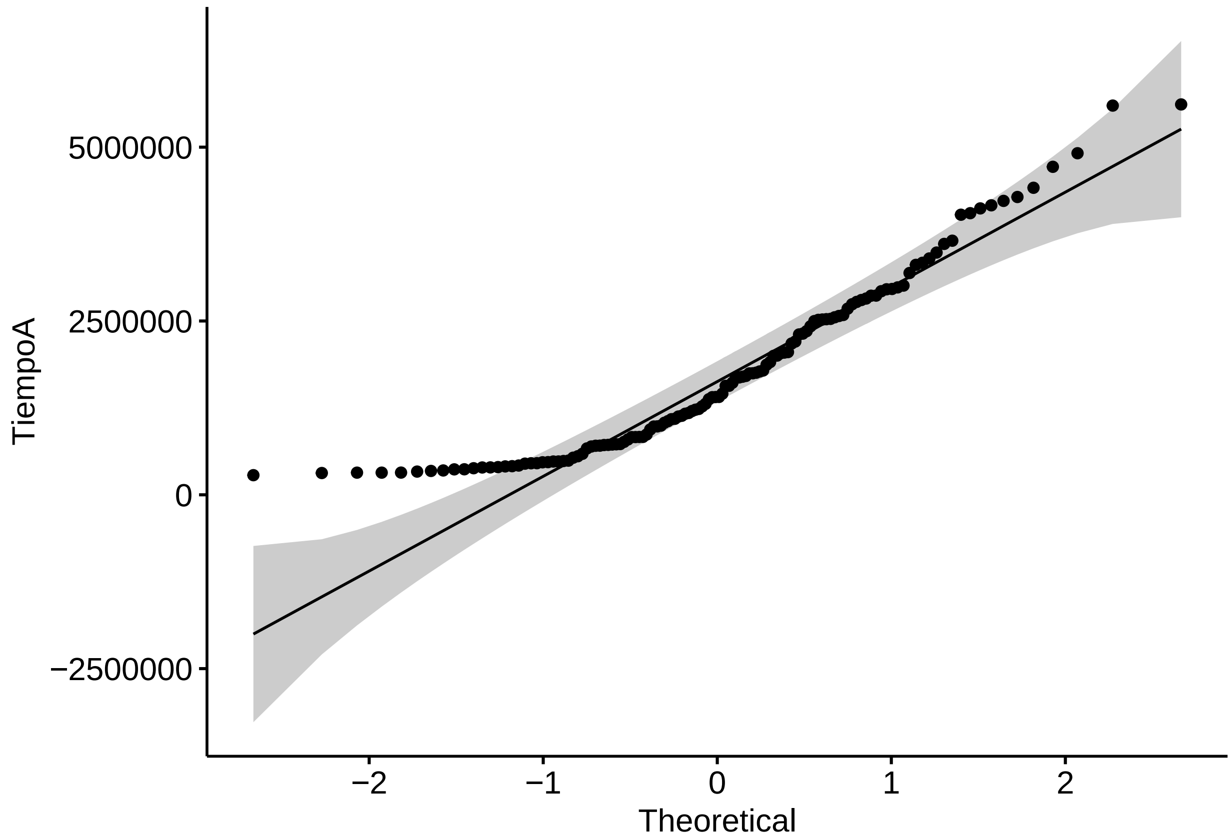
```r
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```r
# Test de Shapiro-Wilk para TiempoA

shapiro.test(TiempoA)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  TiempoA
## W = 0.9055, p-value = 1.53e-07
```

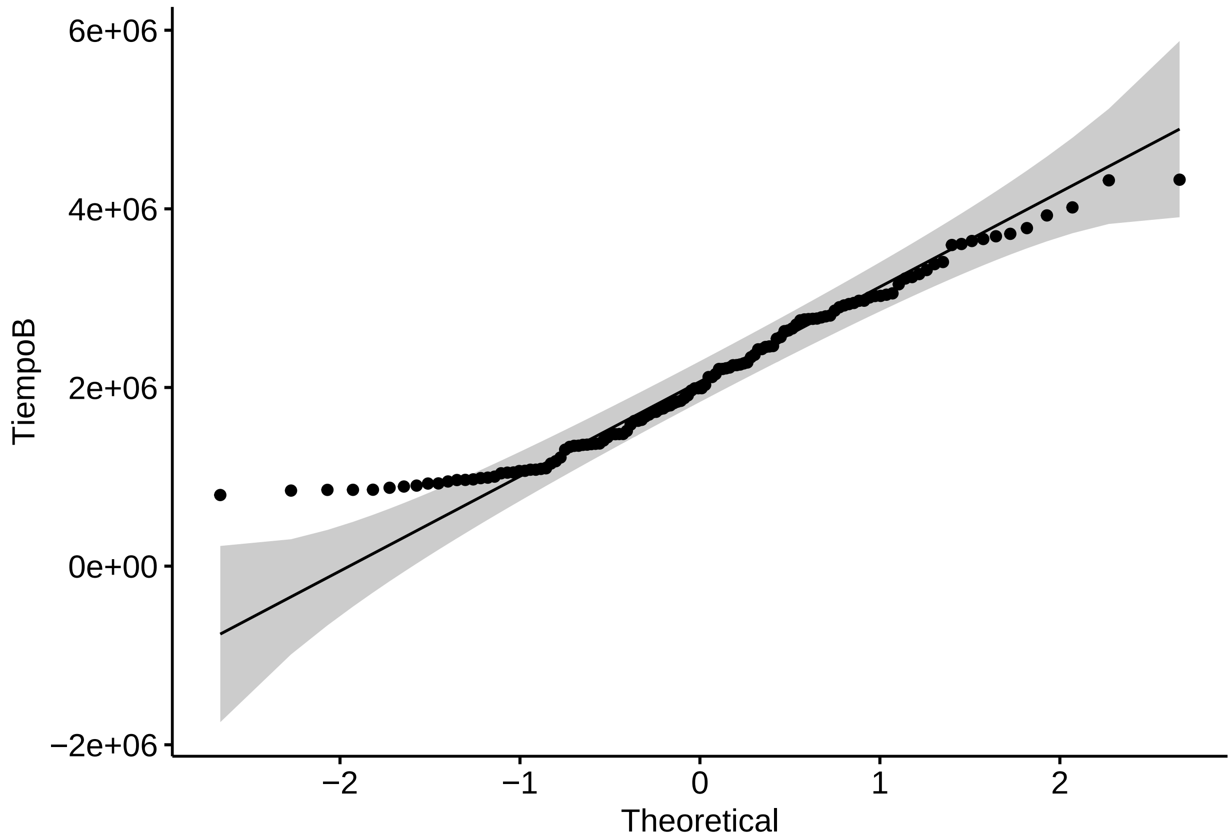
```r
# Realizamos un gráfico Q-Q para TiempoA

g1 <- ggqqplot(TiempoA, ylab = "TiempoA")
print(g1)
```




```r
# Test de Shapiro-Wilk para TiempoB

shapiro.test(TiempoB)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  TiempoB
## W = 0.95206, p-value = 0.0001637
```

```r
# Realizamos un gráfico Q-Q para TiempoB

g2 <- ggqqplot(TiempoB, ylab = "TiempoB")
print(g2)
```



Debido a que los valores obtenidos en las pruebas de Shapiro-Wilk son menores que un valor de significancia del 0.05, y al visualizar el gráfico podemos ver una anormalidad en los datos, rechazamos la hipótesis nula sobre que los datos provienen de una distribución normal. Por lo tanto, no podemos realizar una prueba t de Student, y procederemos a realizar una prueba no paramétrica suma de rangos de Wilcoxon.

Condiciones para realizar la prueba de Wilcoxon

1. Las observaciones de ambas muestras son independientes: debido al enunciado del problema y los datos entregados, podemos asumir que las observaciones son independientes.
2. La escala de medición empleada debe ser a lo menos ordinal: los datos entregados son de tipo numérico, por lo que cumplen con esta condición.

Como se cumplen las condiciones para realizar la prueba de Wilcoxon, procedemos a realizarla.

```
# Prueba de Wilcoxon

# Por enunciado de la pregunta, seteamos la semilla en 73
set.seed(73)

#Filtramos las columnas que necesitamos

datos_filtradosEJ1 <- datos_filtradosEJ1 %>% select(instancia, tiempo.A, tiempo.B)

# Obtenemos muestras aleatorias independientes de 24 tiempos registrados por la versión A y 20 tiempos
```

```r
n_A <- 24
n_B <- 20

#Como las muestras deben ser independientes, obtenemos la totalidad de las muestras aleatorias de las i

Muestras1 <- sample_n(datos_filtradosEJ1, n_A + n_B)

# Obtenemos las muestras de los tiempos de ejecución de las versiones A y B

muestraA <- Muestras1[1:n_A, "tiempo.A"]

muestraB <- Muestras1[(n_A + 1):(n_A + n_B), "tiempo.B"]

# Realizamos la prueba de Wilcoxon

alpha <- 0.05

# Realizamos la prueba de Wilcoxon
Prueba_Wilcoxon <- wilcox.test(muestraA, muestraB, alternative = "two.sided", paired = FALSE, conf.level

print(Prueba_Wilcoxon)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  muestraA and muestraB
## W = 151, p-value = 0.03605
## alternative hypothesis: true location shift is not equal to 0
```

Con los resultados obtenidos, siendo un valor de P de 0.03605, podemos rechazar la hipótesis nula, por lo que existen diferencias significativas en el tiempo de ejecución entre las versiones A y B del algoritmo cuando las instancias tienen 70 o más nodos.

Pregunta 2

La memorista también sospecha que, al comparar las mismas instancias de prueba con iguales características, las mejores soluciones encontradas por las versiones B y C tienen rendimientos distintos. ¿Estará en lo cierto? Para responder, filtren los datos para tener las instancias con 70 o más nodos y seleccionen las columnas con el mejor rendimiento de las versiones B y C en formato ancho. Usando como semilla el valor 71, obtengan una muestra aleatoria de 24 instancias. Realicen un análisis estadístico pertinente (enunciar hipótesis, revisar condiciones, seleccionar pruebas ómnibus y post-hoc según corresponda) para responder la pregunta planteada, utilizando pruebas no paramétricas de ser necesario.

```r
# Filtramos los datos para tener las instancias con 70 o más nodos

datos_filtradosEJ2 <- datos %>% filter(n.nodos >= 70)

# Mostramos los primeros datos
head(datos_filtradosEJ2)
```

```
##   instancia n.nodos n.aristas tiempo.A tiempo.B tiempo.C mejor.A mejor.B
## 1        41      70      1212   382143   947924   536924   99.36   97.38
```

```
## 2         42      70      1232   394395   965334   547148   98.77   97.18
## 3         43      70      1105   318794   854179   484068   98.58   98.01
## 4         44      70      1260   411999   990491   560973   98.25   98.71
## 5         45      70      1254   408354   985163   557996   98.99   98.79
## 6         46      70      1147   343080   891077   505090   99.42   96.36
##   mejor.C
## 1   97.43
## 2   97.89
## 3   96.20
## 4   97.64
## 5   98.95
## 6   98.95
```

Ahora debemos obtener solamente los datos de mejor rendimiento de las versiones B y C, para luego realizar el análisis estadístico pertinente.

```r
# Seleccionamos los mejores rendimientos de las versiones B y C de las instancias que tienen 70 o más n

datos_filtradosEJ2 <- datos_filtradosEJ2 %>% select(instancia, mejor.B, mejor.C)

head(datos_filtradosEJ2)
```

```
##   instancia mejor.B mejor.C
## 1        41   97.38   97.43
## 2        42   97.18   97.89
## 3        43   98.01   96.20
## 4        44   98.71   97.64
## 5        45   98.79   98.95
## 6        46   96.36   98.95
```

Formulación de Hipótesis

H0: No existe una diferencia significativa en el mejor rendimiento entre las versiones B y C del algoritmo cuando las instancias tienen 70 o más nodos.

Ha: Existe una diferencia significativa en el mejor rendimiento entre las versiones B y C del algoritmo cuando las instancias tienen 70 o más nodos.

Revisión de Condiciones

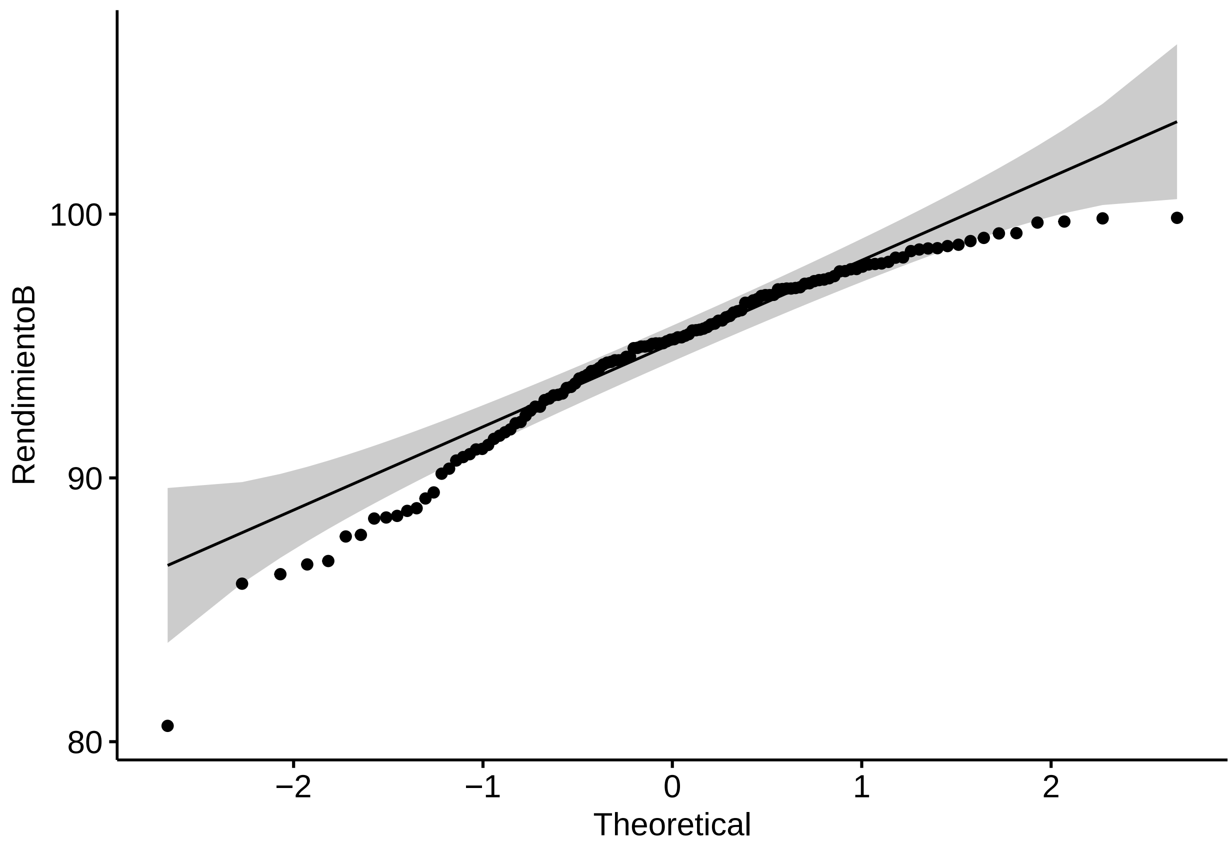
Como queremos comparar dos muestras independientes, debemos verificar que las muestras provengan de una población con distribución normal. Para esto, utilizaremos el test de Shapiro-Wilk junto a un gráfico Q-Q para verificar la normalidad de los datos.

```r
library(ggpubr)
# Test de Shapiro-Wilk para RendimientoB

shapiro.test(datos_filtradosEJ2$mejor.B)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  datos_filtradosEJ2$mejor.B
## W = 0.9322, p-value = 6.138e-06
```

```r
# Gráfico Q-Q para RendimientoB

g1 <- ggqqplot(datos_filtradosEJ2$mejor.B, ylab = "RendimientoB")
print(g1)
```

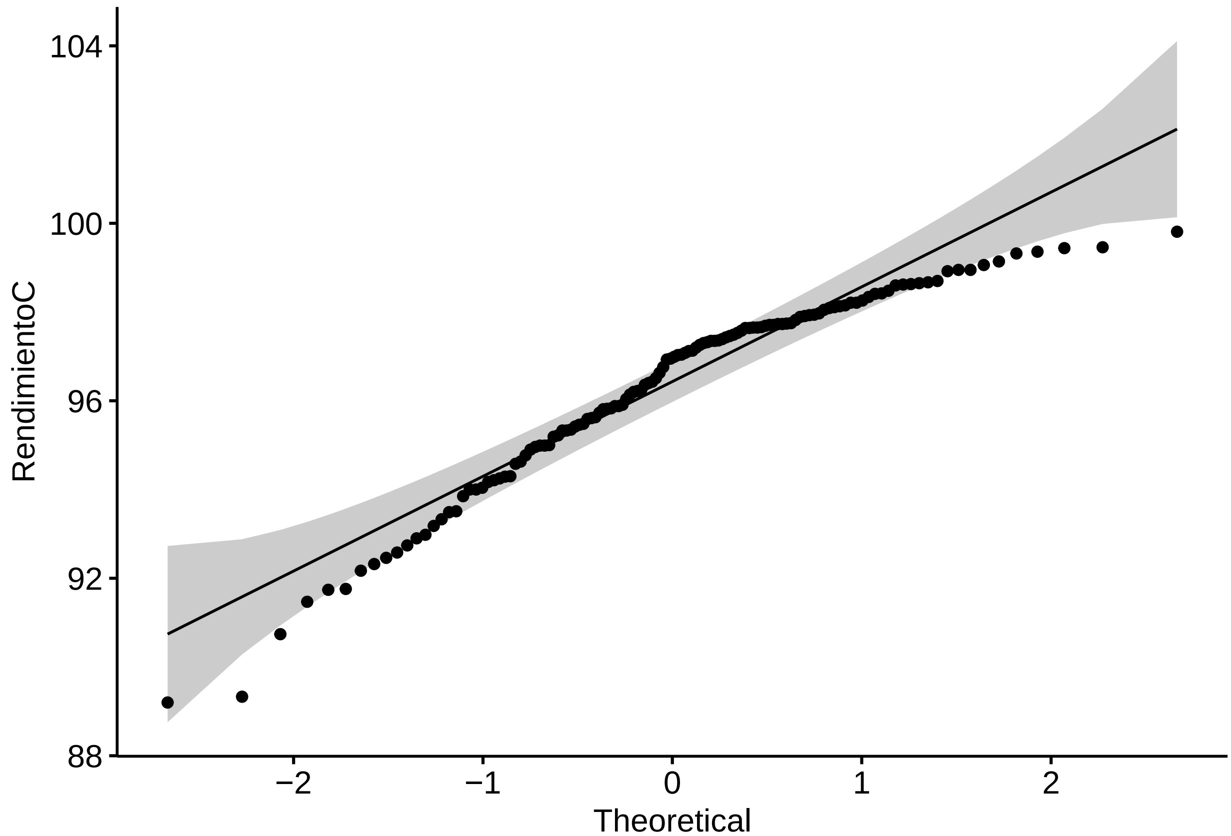


Ahora realizamos el mismo procedimiento para los datos de mejor rendimiento de la versión C.

```r
# Test de Shapiro-Wilk para RendimientoC

shapiro.test(datos_filtradosEJ2$mejor.C)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_filtradosEJ2$mejor.C
## W = 0.93625, p-value = 1.147e-05
```

```r
# Gráfico Q-Q para RendimientoC

g2 <- ggqqplot(datos_filtradosEJ2$mejor.C, ylab = "RendimientoC")

print(g2)
```



Debido a los valores P obtenidos, siendo estos menores a un nivel de significancia del 0.05, y además de la verificación mediante los gráficos Q-Q, podemos apreciar que los datos no provienen de una distribución normal, por lo que no podemos realizar una prueba t de Student, y procederemos a realizar una prueba no paramétrica de Wilcoxon.

Por otro lado, como la memorista realiza la pregunta para comparar mismas instancias, podemos concluir que para esta prueba son datos pareados, por lo que realizaremos una prueba de rangos con signo de Wilcoxon.

Condiciones para realizar la prueba de Wilcoxon

1. Las observaciones de ambas muestras son independientes: debido al enunciado del problema y los datos entregados, podemos asumir que las observaciones son independientes.
2. La escala de medición empleada debe ser a lo menos ordinal: los datos entregados son de tipo numérico, por lo que cumplen con esta condición.

Como se cumplen las condiciones para realizar la prueba de Wilcoxon, procedemos a realizarla.

```
# Realizamos la prueba de Wilcoxon

# Por enunciado de la pregunta, seteamos la semilla en 71

set.seed(71)

# Obtenemos una muestra aleatoria de 24 instancias
```

```r
muestra <- sample_n(datos_filtradosEJ2, 24)

# Mostramos la muestra

print(muestra)
```

```
##    instancia mejor.B mejor.C
## 1         99   95.00   98.21
## 2         68   95.60   98.48
## 3        119   98.36   98.26
## 4         88   94.60   96.63
## 5        150   88.75   90.74
## 6         41   97.38   97.43
## 7        143   88.46   97.49
## 8         90   92.36   95.19
## 9         48   97.57   97.93
## 10       116   97.15   96.14
## 11        66   98.84   98.42
## 12       117   91.48   91.47
## 13       125   85.99   95.33
## 14        63   96.93   97.35
## 15       130   95.59   95.42
## 16       107   94.93   97.12
## 17       122   94.17   95.88
## 18        60   96.90   99.44
## 19       168   95.10   97.13
## 20        65   95.33   95.48
## 21       120   93.13   97.03
## 22       100   97.46   98.62
## 23        69   97.92   97.36
## 24        87   95.97   95.91
```

```r
# Realizamos la prueba de Wilcoxon

alpha <- 0.05 # Nivel de significancia

Prueba_Wilcoxon_EJ2 <- wilcox.test(muestra$mejor.B, muestra$mejor.C, alternative = "two.sided", paired =
```

```
## Warning in wilcox.test.default(muestra$mejor.B, muestra$mejor.C, alternative =
## "two.sided", : cannot compute exact p-value with ties
```

```r
print(Prueba_Wilcoxon_EJ2)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  muestra$mejor.B and muestra$mejor.C
## V = 44, p-value = 0.002575
## alternative hypothesis: true location shift is not equal to 0
```

Con los resultados obtenidos, siendo un valor de p de 0.002575, podemos rechazar la hipótesis nula, por lo que existen diferencias significativas en el mejor rendimiento entre las versiones B y C del algoritmo cuando las instancias tienen 70 o más nodos.

Pregunta 3

La memorista sospecha que hay diferencias significativas en el tiempo de ejecución entre las versiones del algoritmo cuando las instancias de prueba tienen 50 o más nodos. ¿Los datos respaldan la intuición de la memorista? Para responder, filtren los datos para tener las instancias con 50 o más nodos y seleccionen las columnas con los tiempos de ejecución registrados (en formato ancho). Usando como semilla el valor 43, obtengan muestras aleatorias independientes de 13, 14 y 13 tiempos registrados por las versiones A, B y C, respectivamente. Realicen un análisis estadístico pertinente (enunciar hipótesis, revisar condiciones, seleccionar pruebas ómnibus y post-hoc según corresponda) para responder la pregunta planteada, utilizando pruebas no paramétricas de ser necesario.

```r
library(dplyr)
# Filtramos los datos para tener las instancias con 50 o más nodos

datos_filtradosEJ3 <- datos %>% filter(n.nodos >= 50)

# Mostramos los primeros datos

head(datos_filtradosEJ3)
```

```
##   instancia n.nodos n.aristas tiempo.A tiempo.B tiempo.C mejor.A mejor.B
## 1         1      50       631   107534   452595   257485   98.72   98.25
## 2         2      50       521    74808   364061   207297   98.99   99.17
## 3         3      50       588    94072   417798   237793   99.10   99.23
## 4         4      50       653   114830   470701   267598   98.69   99.23
## 5         5      50       597    96720   425233   241770   99.80   99.22
## 6         6      50       564    86688   398448   226833   99.19   99.15
##   mejor.C
## 1   99.34
## 2   99.48
## 3   99.10
## 4   97.82
## 5   98.14
## 6   98.04
```

Ahora debemos obtener solamente los datos de tiempo de ejecución de las versiones A, B y C, para luego realizar el análisis estadístico pertinente.

```r
# Seleccionamos los tiempos de ejecución de las versiones A, B y C de las instancias que tienen 50 o má

datos_filtradosEJ3 <- datos_filtradosEJ3 %>% select(instancia, tiempo.A, tiempo.B, tiempo.C)

# Obtenemos muestras aleatorias independientes de 13, 14 y 13 tiempos registrados por las versiones A, 

set.seed(43)

nA <- 13; nB <- 14; nC <- 13

nT <- nA + nB + nC

Datos_Muestra3 <- datos_filtradosEJ3[sample(1:nrow(datos_filtradosEJ3), nT),]

head(Datos_Muestra3)
```

```
##     instancia tiempo.A tiempo.B tiempo.C
## 44         44   411999   990491   560973
## 40         40   154137   559792   317852
## 149       149  3608636  3380926  1906000
## 66         66   491683  1095696   620035
## 5           5    96720   425233   241770
## 77         77   554792  1174722   664930
```

Ahora, como nos piden que las muestras deben ser independientes, obtenemos las muestras de 13, 14 y 13 tiempos registrados por las versiones A, B y C, respectivamente de distintas instancias.

```
# Obtenemos las muestras de distintas instancias.

Muestra_A <- Datos_Muestra3[["tiempo.A"]][1:nA] # 13 tiempos registrados por la versión A
Muestra_B <- Datos_Muestra3[["tiempo.B"]][(nA + 1):(nA + nB)] # 14 tiempos registrados por la versión B
Muestra_C <- Datos_Muestra3[["tiempo.C"]][(nA + nB + 1):(nA + nB + nC)] # 13 tiempos registrados por la
```

Formulación de Hipótesis

H0: No existe una diferencia significativa en el tiempo de ejecución entre las versiones del algoritmo cuando las instancias tienen 50 o más nodos. (muA = muB = muC)

Ha: Existe una diferencia significativa en el tiempo de ejecución entre las versiones del algoritmo cuando las instancias tienen 50 o más nodos. (muA != muB != muC)

Como la memorista realiza la pregunta para comparar las versiones del algoritmo A, B y C. Podemos inferir que debemos realizar una prueba ANOVA, debido a comparar más de dos grupos, es por eso que realizaremos las verificaciones de condiciones para realizar esta prueba.

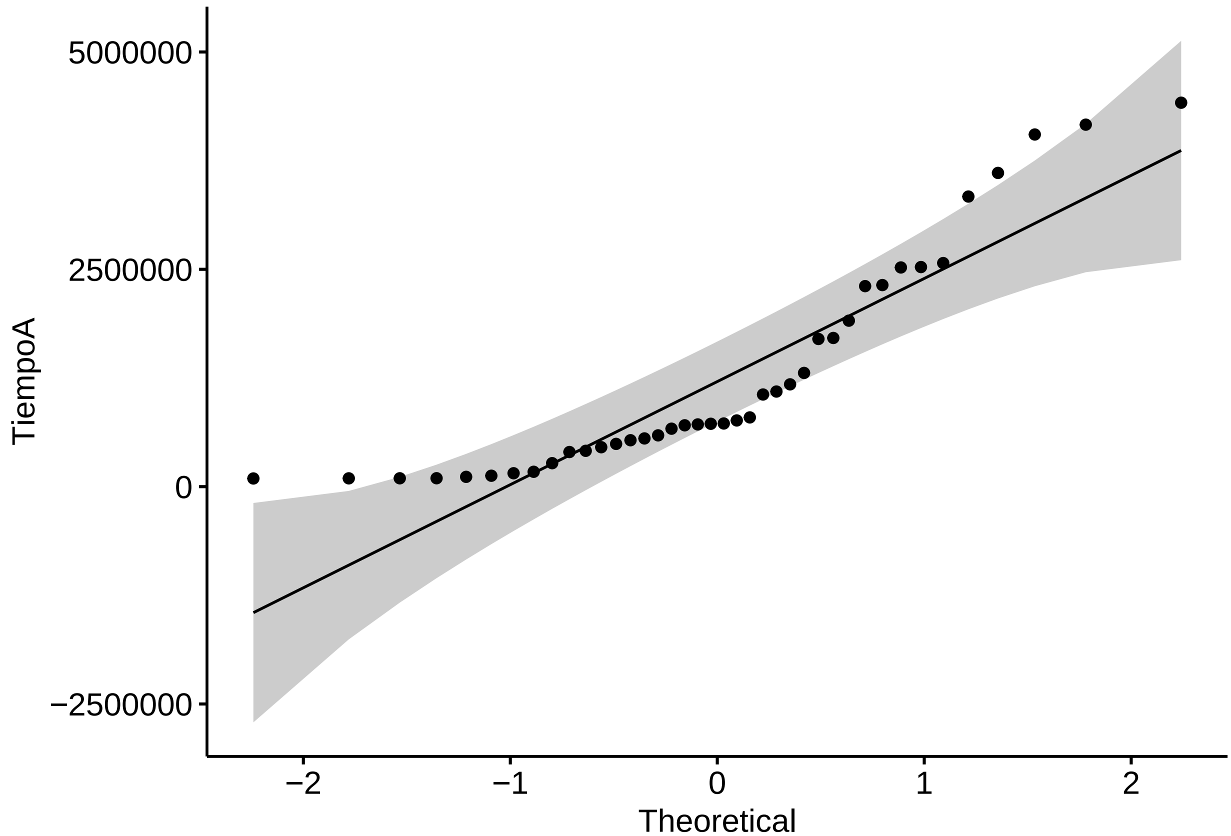
Condiciones

1. La escala con que se mide la variable dependiente tiene las propiedades de una escala de intervalos iguales: los datos entregados son de tipo numérico, por lo que cumplen con esta condición.
2. Las k muestras son obtenidas de manera aleatoria e independiente desde la(s) población(es) de origen: debido al enunciado del problema y los datos entregados, podemos asumir que las observaciones son independientes, además de que se obtienen de manera aleatoria.
3. Se puede suponer razonablemente que la(s) población(es) de origen sigue(n) una distribución normal: Para verificar esta condición, utilizaremos el test de Shapiro-Wilk junto a un gráfico Q-Q para verificar la normalidad de los datos.

```
library(ggpubr)
# Test de Shapiro-Wilk para TiempoA

shapiro.test(Datos_Muestra3$tiempo.A)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Datos_Muestra3$tiempo.A
## W = 0.83491, p-value = 3.95e-05
```

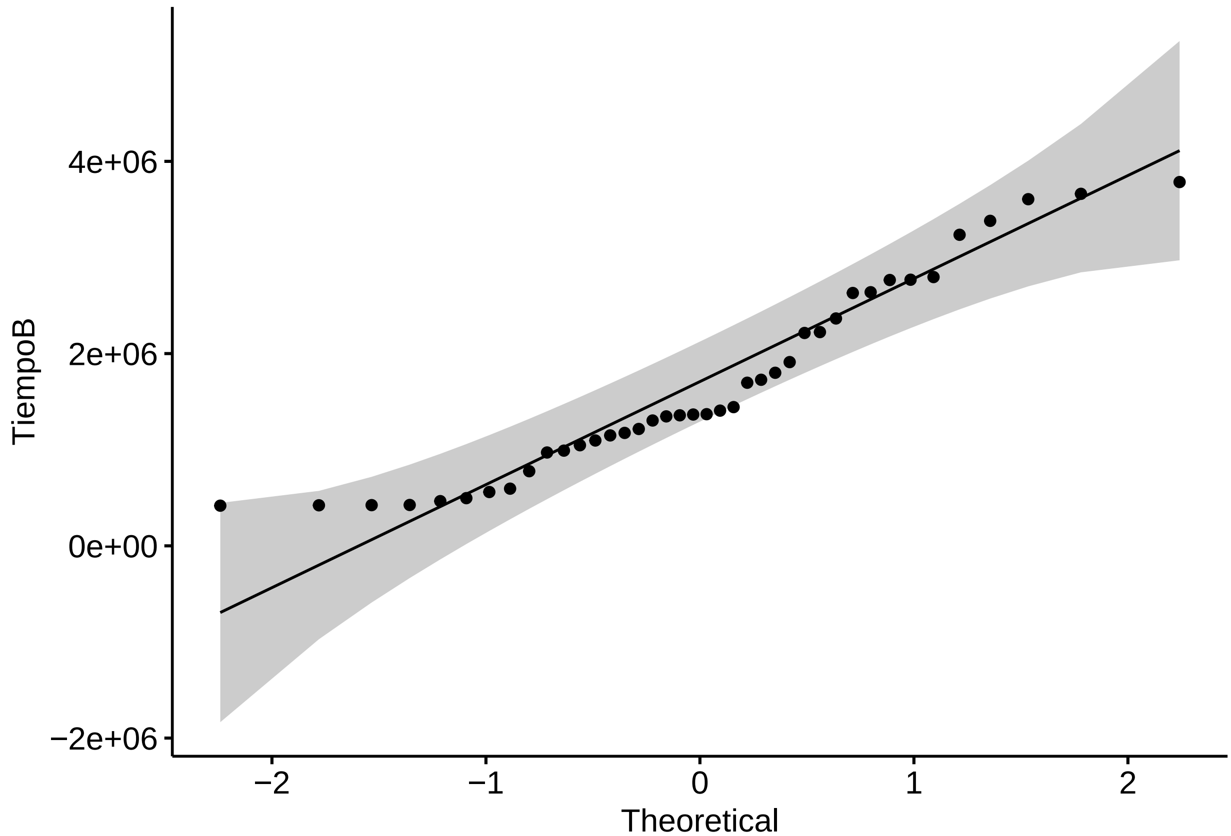
```r
# Realizamos un gráfico Q-Q para TiempoA

g1 <- ggqqplot(Datos_Muestra3$tiempo.A, ylab = "TiempoA")

print(g1)
```




```r
# Test de Shapiro-Wilk para TiempoB

shapiro.test(Datos_Muestra3$tiempo.B)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Datos_Muestra3$tiempo.B
## W = 0.91991, p-value = 0.007654
```

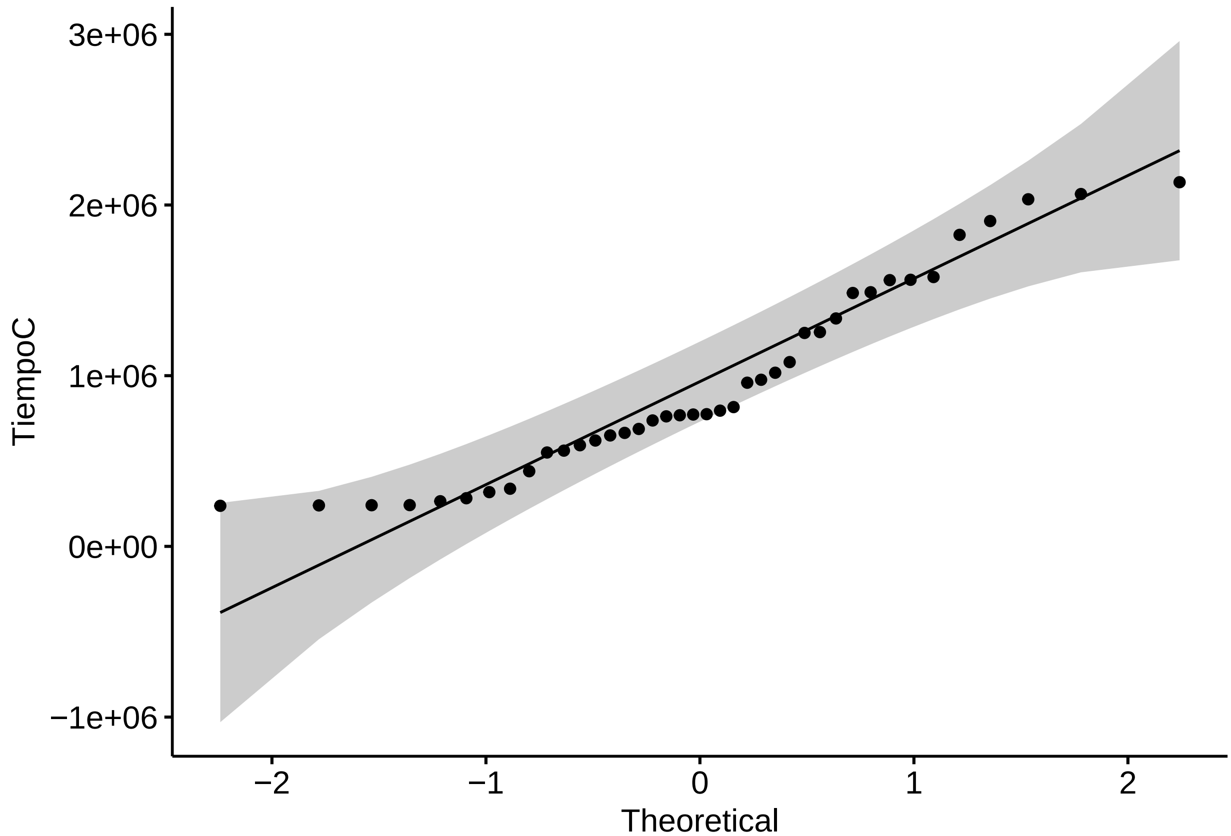
```r
# Realizamos un gráfico Q-Q para TiempoB

g2 <- ggqqplot(Datos_Muestra3$tiempo.B, ylab = "TiempoB")

print(g2)
```




```r
# Test de Shapiro-Wilk para TiempoC

shapiro.test(Datos_Muestra3$tiempo.C)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Datos_Muestra3$tiempo.C
## W = 0.92011, p-value = 0.007761
```

```r
# Realizamos un gráfico Q-Q para TiempoC

g3 <- ggqqplot(Datos_Muestra3$tiempo.C, ylab = "TiempoC")

print(g3)
```



Debido a los valores P obtenidos, siendo estos menores al nivel de significancia del 0.05, y al visualizar los gráficos Q-Q, podemos apreciar que los datos no provienen de una distribución normal, por lo que no podemos realizar una prueba ANOVA directamente, es por eso que procederemos a realizar una prueba no paramétrica de Kruskal-Wallis.

Condiciones para realizar la prueba de Kruskal-Wallis

1. La variable independiente debe tener al menos dos niveles: en este caso, la variable independiente es el tiempo de ejecución de las versiones A, B y C del algoritmo.
2. La escala de la variable dependiente debe ser, al menos, ordinal: los datos entregados son de tipo numérico, por lo que cumplen con esta condición.
3. Las observaciones de cada grupo deben ser independientes: debido al enunciado del problema y los datos entregados, podemos asumir que las observaciones son independientes.

Como se cumplen las condiciones para realizar la prueba de Kruskal-Wallis, procedemos a realizarla.

```
# Realizamos la prueba de Kruskal-Wallis

# Definimos el vector de tiempos

Tiempo <- c(Muestra_A, Muestra_B, Muestra_C)

# Definimos el vector de grupos
```

```r
Grupo <- c(rep("A", nA), rep("B", nB), rep("C", nC))
Grupo <- factor(Grupo)

datos_kruskal <- data.frame(Tiempo, Grupo)

print(datos_kruskal)
```

```
##     Tiempo Grupo
## 1   411999     A
## 2   154137     A
## 3  3608636     A
## 4   491683     A
## 5    96720     A
## 6   554792     A
## 7   125645     A
## 8  2571876     A
## 9  1059855     A
## 10  589858     A
## 11   95588     A
## 12  723014     A
## 13  453516     A
## 14 3662204     B
## 15  594740     B
## 16 1357717     B
## 17 1443220     B
## 18  417798     B
## 19 2365495     B
## 20 1369662     B
## 21 2768526     B
## 22 1346778     B
## 23 1911543     B
## 24  776426     B
## 25 1303900     B
## 26  465122     B
## 27 1406599     B
## 28 1559700     C
## 29 1484462     C
## 30  976107     C
## 31 1017315     C
## 32 2033393     C
## 33 2133500     C
## 34 1255338     C
## 35  241141     C
## 36 1250374     C
## 37  549967     C
## 38 1488924     C
## 39 1824954     C
## 40  649933     C
```

```r
# Nivel de significancia

alpha <- 0.05
```

```r
# Realizamos la prueba de Kruskal-Wallis

prueba_kruskal <- kruskal.test(Tiempo ~ Grupo, data = datos_kruskal)

# Mostramos el resultado obtenido

print(prueba_kruskal)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Tiempo by Grupo
## Kruskal-Wallis chi-squared = 7.7242, df = 2, p-value = 0.02102
```

Luego de realizar la prueba de Kruskal-Wallis, y obteniendo un valor de P de 0.02102, podemos rechazar la hipótesis nula, por lo que existen diferencias significativas en el tiempo de ejecución entre las versiones del algoritmo cuando las instancias tienen 50 o más nodos.

Debido a esto, procederemos a realizar una prueba post-hoc de Benjamini-Hochberg para determinar entre cuáles versiones existen diferencias significativas.

```r
# Realizamos la prueba post-hoc de Benjamini-Hochberg

post_hoc_kruskal <- pairwise.wilcox.test(datos_kruskal[["Tiempo"]], datos_kruskal[["Grupo"]], p.adjust.

print(post_hoc_kruskal)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  datos_kruskal[["Tiempo"]] and datos_kruskal[["Grupo"]]
##
##   A     B
## B 0.037 -
## C 0.041 0.716
##
## P value adjustment method: BH
```

Conclusión

Finalmente, luego de seleccionar la prueba de ANOVA como prueba ómnibus inicial, y corroborar que no se cumplen la prueba de normalidad de los datos para realizar esta prueba, procedimos a realizar una prueba no paramétrica de Kruskal-Wallis, concluyendo para esta prueba el rechazar la hipótesis nula, en favor de la alternativa, por lo que existen diferencias significativas en el tiempo de ejecución entre las versiones del algoritmo cuando las instancias tienen 50 o más nodos. Es por eso que procedimos a realizar una prueba post-hoc de Benjamini-Hochberg, concluyendo que existen diferencias significativas entre las versiones A y B, y entre las versiones A y C, pero no entre las versiones B y C.

Pregunta 4

La memorista también sospecha que, al comparar las mismas instancias con iguales características, las mejores soluciones encontradas por las diferentes versiones del algoritmo tienen rendimientos distintos. ¿Estará en lo cierto?

Para responder, filtren los datos para tener las instancias con 50 o más nodos y seleccionen las columnas con los mejores rendimientos registrados. Usando como semilla el valor 16, obtengan una muestra aleatoria de 21 instancias. Realicen un análisis estadístico pertinente (enunciar hipótesis, revisar condiciones, seleccionar pruebas ómnibus y post-hoc según corresponda) para responder la pregunta planteada, utilizando pruebas no paramétricas de ser necesario.

```
# Filtramos los datos para tener las instancias con 50 o más nodos

datos_filtradosEJ4 <- datos %>% filter(n.nodos >= 50)

# Mostramos los primeros datos

head(datos_filtradosEJ4)
```

```
##   instancia n.nodos n.aristas tiempo.A tiempo.B tiempo.C mejor.A mejor.B
## 1         1      50       631   107534   452595   257485   98.72   98.25
## 2         2      50       521    74808   364061   207297   98.99   99.17
## 3         3      50       588    94072   417798   237793   99.10   99.23
## 4         4      50       653   114830   470701   267598   98.69   99.23
## 5         5      50       597    96720   425233   241770   99.80   99.22
## 6         6      50       564    86688   398448   226833   99.19   99.15
##   mejor.C
## 1   99.34
## 2   99.48
## 3   99.10
## 4   97.82
## 5   98.14
## 6   98.04
```

Ahora debemos obtener solamente los datos de mejor rendimiento de las versiones A, B y C, para luego realizar el análisis estadístico pertinente.

```
# Seleccionamos los mejores rendimientos de las versiones A, B y C de las instancias que tienen 50 o má

datos_filtradosEJ4 <- datos_filtradosEJ4 %>% select(instancia, mejor.A, mejor.B, mejor.C)

# Obtenemos una muestra aleatoria de 21 instancias

set.seed(16)

muestra4 <- sample_n(datos_filtradosEJ4, 21)

# Mostramos la muestra

print(muestra4)
```

```
##    instancia mejor.A mejor.B mejor.C
## 1        127   96.66   88.50   98.09
## 2         59   99.05   98.13   97.66
## 3         15   99.51   99.51   99.62
## 4          8   99.79   97.95   99.22
## 5         20   99.55   99.35   99.67
## 6         84   97.59   94.40   97.35
```

```
## 7         64   98.28   98.19   97.08
## 8         81   98.76   96.64   95.88
## 9        125   98.39   85.99   95.33
## 10       126   97.56   92.95   97.71
## 11        67   98.35   98.66   97.65
## 12       124   95.09   96.32   94.63
## 13       170   97.33   88.56   92.46
## 14       105   97.96   96.93   97.91
## 15        62   99.02   97.16   97.30
## 16        83   96.90   93.83   98.92
## 17        21   98.58   98.09   98.98
## 18       141   99.56   97.84   92.98
## 19        88   97.05   94.60   96.63
## 20       142   96.94   90.35   92.58
## 21       165   93.42   99.86   99.06
```

Formulación de Hipótesis

H0: No existe una diferencia significativa en el mejor rendimiento entre las versiones del algoritmo cuando las instancias tienen 50 o más nodos. (muA = muB = muC)

Ha: Existe una diferencia significativa en el mejor rendimiento de al menos una de las versiones del algoritmo cuando las instancias tienen 50 o más nodos.

Como la memorista realiza la pregunta para comparar las versiones del algoritmo A, B y C. Podemos inferir que debemos realizar una prueba ANOVA, debido a comparar más de dos grupos, por otro lado, como nos indican comparar una misma instancia en diferentes versiones, podemos inferir que los datos son pareados, por lo que se debería de realizar una prueba de ANOVA para muestras correlacionadas.

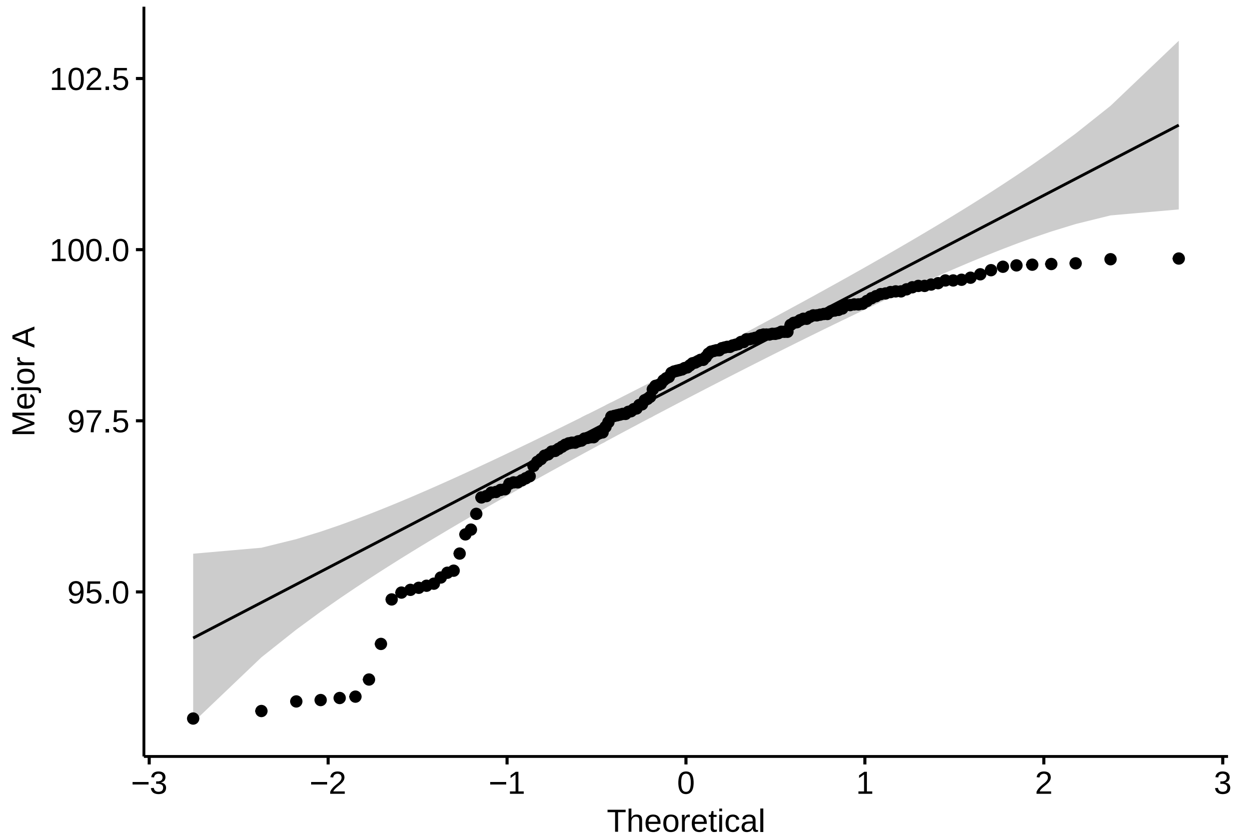
Condiciones para realizar la prueba de ANOVA

1. La escala con que se mide la variable dependiente tiene las propiedades de una escala de intervalos iguales: los datos entregados son de tipo numérico, por lo que cumplen con esta condición.
2. Las mediciones son independientes al interior de cada grupo: debido al enunciado del problema y los datos entregados, podemos asumir que las observaciones son independientes.
3. Se puede suponer razonablemente que la(s) población(es) de origen sigue(n) una distribución normal: Para verificar esta condición, utilizaremos el test de Shapiro-Wilk junto a un gráfico Q-Q para verificar la normalidad de los datos.

```r
# Test de Shapiro-Wilk para muestra A

shapiro.test(datos_filtradosEJ4$mejor.A)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_filtradosEJ4$mejor.A
## W = 0.89525, p-value = 1.329e-09
```

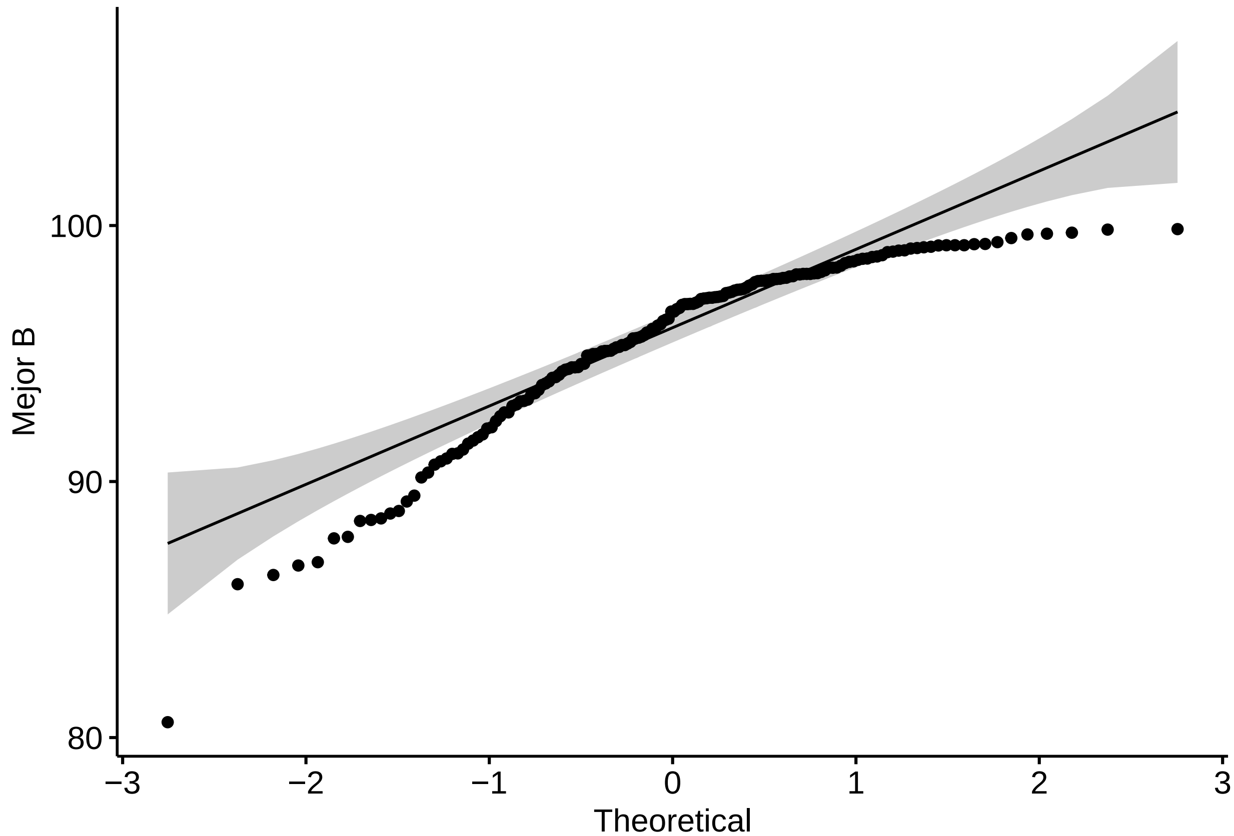
```r
# Gráfico Q-Q para muestra A

g1 <- ggqqplot(datos_filtradosEJ4$mejor.A, ylab = "Mejor A")

print(g1)
```




```r
# Test de Shapiro-Wilk para muestra B

shapiro.test(datos_filtradosEJ4$mejor.B)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_filtradosEJ4$mejor.B
## W = 0.89104, p-value = 7.644e-10
```

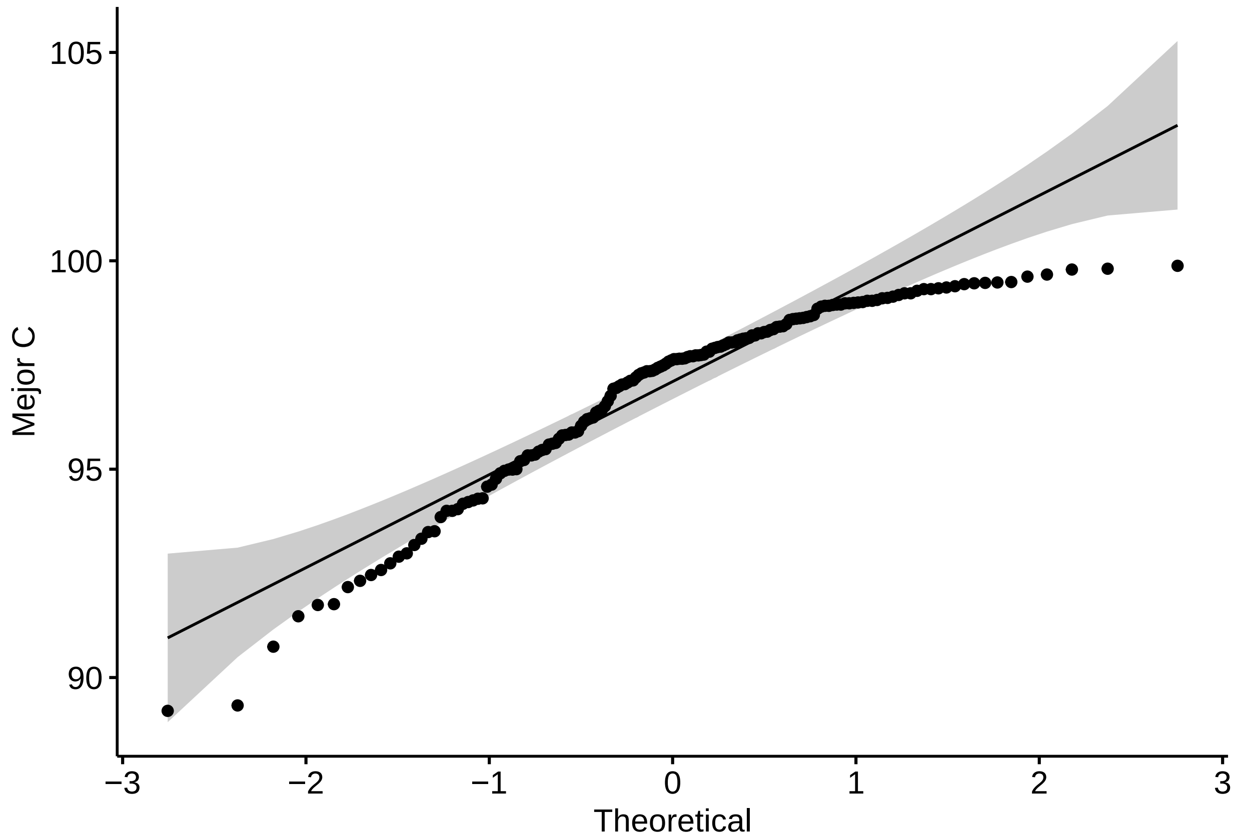
```r
# Gráfico Q-Q para muestra B

g2 <- ggqqplot(datos_filtradosEJ4$mejor.B, ylab = "Mejor B")

print(g2)
```




```r
# Test de Shapiro-Wilk para muestra C

shapiro.test(datos_filtradosEJ4$mejor.C)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos_filtradosEJ4$mejor.C
## W = 0.90792, p-value = 7.691e-09
```

```r
# Gráfico Q-Q para muestra C

g3 <- ggqqplot(datos_filtradosEJ4$mejor.C, ylab = "Mejor C")

print(g3)
```



Debido a que los valores P obtenidos, siendo estos menores al nivel de significancia del 0.05, y al visualizar los gráficos Q-Q, podemos apreciar que los datos no provienen de una distribución normal, por lo que no podemos realizar una prueba ANOVA directamente, es por eso que realizaremos una prueba de Friedman, la cual es una alternativa a la prueba de ANOVA para cuando los datos no provienen de una distribución normal.

Condiciones para realizar la prueba de Friedman

1. La variable dependiente debe ser categórica y tener al menos tres niveles: en este caso, la variable dependiente es el mejor rendimiento de las versiones A, B y C del algoritmo, por lo tanto cumple con esta condición.
2. La escala de la variable dependiente debe ser, al menos, ordinal: los datos entregados son de tipo numérico, por lo que cumplen con esta condición.
3. Las observaciones son una muestra aleatoria e independiente de la población: debido al enunciado del problema y los datos entregados, podemos asumir que las observaciones son independientes.

Realizamos la prueba de Friedman

```
# Realizamos la prueba de Friedman

# Definimos el vector de mejores rendimientos
```

```r
Mejor_Rendimiento <- c(muestra4$mejor.A, muestra4$mejor.B, muestra4$mejor.C)

# Definimos el vector de grupos

Grupo <- c(rep("A", 21), rep("B", 21), rep("C", 21))
Grupo <- factor(Grupo)

# Definimos el vector de casos
Caso <- rep(1:21, 3)

datos_friedman <- data.frame(Caso, Mejor_Rendimiento, Grupo)

# Mostramos el dataframe
head(datos_friedman)
```

```
##   Caso Mejor_Rendimiento Grupo
## 1    1             96.66     A
## 2    2             99.05     A
## 3    3             99.51     A
## 4    4             99.79     A
## 5    5             99.55     A
## 6    6             97.59     A
```

```r
# Establecemos el nivel de significancia

alpha <- 0.05

# Realizamos la prueba de Friedman

prueba_friedman <- friedman.test(Mejor_Rendimiento ~ Grupo | Caso, data = datos_friedman)

print(prueba_friedman)
```

```
##
##  Friedman rank sum test
##
## data:  Mejor_Rendimiento and Grupo and Caso
## Friedman chi-squared = 10.627, df = 2, p-value = 0.004926
```

Luego de realizar la prueba de Friedman, y obteniendo un valor de P de 0.004926, podemos rechazar la hipótesis nula, por lo que existen diferencias significativas en el mejor rendimiento entre las versiones del algoritmo cuando las instancias tienen 50 o más nodos. Debido a esto, procederemos a realizar una prueba post-hoc de Holm o Benjamini-Hochberg para determinar entre cuáles versiones existen diferencias significativas.

```r
# Realizamos la prueba post-hoc de Holm

post_hoc_friedman <- pairwise.wilcox.test(datos_friedman[["Mejor_Rendimiento"]], datos_friedman[["Grupo

# Mostramos el resultado obtenido

print(post_hoc_friedman)
```

```
##
##  Pairwise comparisons using Wilcoxon signed rank test with continuity correction
##
## data:  datos_friedman[["Mejor_Rendimiento"]] and datos_friedman[["Grupo"]]
##
##   A    B
## B 0.01 -
## C 0.14 0.14
##
## P value adjustment method: holm
```

Conclusión

Luego de realizar la prueba de Friedman, y obtener un valor de P de 0.004926, podemos rechazar la hipótesis nula, por lo que existen diferencias significativas en el mejor rendimiento entre las versiones del algoritmo cuando las instancias tienen 50 o más nodos. Por otro lado, al realizar la prueba post-hoc de Holm, podemos concluir que existen diferencias significativas entre el mejor rendimiento de las versiones A y B.