

TAREA # 3

Teoría de colas

Natalia Berenice Pérez López

15 de septiembre de 2021

1. Objetivo

El objetivo de esta práctica es examinar cómo las diferencias en los tiempos de ejecución de los diferentes ordenamientos cambian cuando se varía el número de núcleos asignados al cluster, utilizando como datos de entrada un vector que contiene primos grandes y no primos chicos, aplicando pruebas estadísticas adecuadas y visualización científica clara e informativa.

2. Desarrollo

Para generar el código objetivo de esta práctica se realizaron diversos códigos prueba, los cuales se encuentran en [mi repositorio](#) en GitHub. Para obtener los datos de entrada de mis pruebas estadísticas se utilizó el código revisado en clase para Python [6], el cual genera números primos grandes y números no primos chicos, designados en el código como números **difíciles** y números **fáciles** respectivamente. Además, el código mide los tiempos para obtener estos números en diferente orden, se tiene el orden de fáciles primero, difíciles primero y el orden aleatorio. Estos ordenamientos se realizan de forma paralela variando el número de núcleos asignados al cluster.

El cuadro 1 contiene las principales variables utilizadas en el código.

Cuadro 1: Variables del experimento.

NOMBRE	VARIABLE
difíciles	números primos grandes
fáciles	números no primos chicos
FP	orden fáciles primero
DP	orden difíciles primero
OA	orden aleatorio
Trabajador 1	1 núcleo asignado en el cluster
Trabajador 2	2 núcleos asignados en el cluster
Trabajador 3	3 núcleos asignados en el cluster

Las modificaciones que se le realizaron al código de Python revisado en clase fueron; cambiar el número de **trabajadores** a 3 debido a que mi computadora tiene 4 núcleos físicos, agregar la librería de **pandas** para hacer un **pd.DataFrame** que almacenará los datos generados y agregar al final la instrucción de exportar en formato **.csv** dichos datos para así poder manipularlos en **RStudio**. A continuación se muestra el código modificado:

```
1 from time import time
2 from math import sqrt, ceil
3 from random import randint, random, shuffle
4 import pandas as pd
5
6 def primo(n):
7     if n < 4: # 1 2 3
8         return True
9     if n % 2 == 0:
10        return False
11    for d in range(3, int(ceil(sqrt(n)))):
12        if n % d == 0:
```

```

13         return False
14     return True
15
16 df = pd.DataFrame()
17 difciles = []
18 meta = 10
19 faciles = [randint(1000, 15000) for i in range(meta)]
20 while len(difciles) < meta:
21     n = randint(50000000, 100000000)
22     if n % 2 == 0:
23         n += 1
24     if primo(n):
25         difciles.append(n)
26
27 from multiprocessing import Pool
28
29 if __name__ == "__main__":
30     c1 = faciles + difciles
31     c2 = difciles + faciles
32     cr = c2.copy()
33     shuffle(cr)
34     ordenes = {'FP': c1, 'DP': c2, 'OA': cr}
35     for trabajadores in range(1, 4):
36         with Pool(trabajadores) as p:
37             for o in ordenes:
38                 label = o
39                 datos = ordenes[o]
40                 for replica in range(10):
41                     start = time()
42                     p.map(primo, datos)
43                     tiempo = 1000 * (time() - start)
44                     if replica > 0:
45                         print(f'{replica},{trabajadores},{label},{tiempo}')
46                         df = df.append(pd.DataFrame(data = {f'{replica},{trabajadores},{label},
47                             {tiempo}'))))
48
49 print(f'{df}')
50 df.to_csv('misdatoos.csv')

```

Listing 1: Código para exportar datos en formato .csv.

Los datos que se obtuvieron del código anterior se muestran en el cuadro 2.

Cuadro 2: Tiempo de los trabajadores de acuerdo al orden de trabajo.

Trabajador	FP				DP				OA			
1	15,77	23,81	15,46	24,11	13,17	16,50	15,71	19,74	15,51	12,68	20,90	27,42
	26,92	22,63	17,25	18,84	16,60	20,60	14,33	22,16	19,84	17,07	19,62	23,20
	16,11				16,80				13,72			
2	14,75	10,27	07,96	08,82	10,67	16,54	13,50	13,98	12,76	11,74	15,04	12,60
	06,35	07,60	07,81	12,74	12,95	12,32	11,01	14,78	12,66	14,00	11,24	12,44
	15,53				13,51				14,35			
3	06,69	06,13	04,87	06,91	13,08	15,43	16,84	12,91	10,54	15,89	09,59	09,44
	05,89	06,55	08,79	17,59	08,64	12,15	13,50	10,75	18,75	14,53	10,31	21,65
	19,47				16,41				16,08			

El archivo .csv generado con Python que contiene los datos del cuadro 2, se utilizó para realizar un diagrama caja-bigote en RStudio (Ver figura 1).

En este diagrama caja-bigote podemos notar que existe una disminución en el tiempo cuando las tareas se realizan con el Trabajador 2 y el Trabajador 3, lo cual tiene sentido ya que estos trabajadores tienen más núcleos asignados al cluster. Además podemos notar que el orden en que se realizan las trabajos afecta de manera diferente a cada trabajador. Para comprobar si las medias de cada trabajador son iguales o diferentes se realizó una prueba estadística.

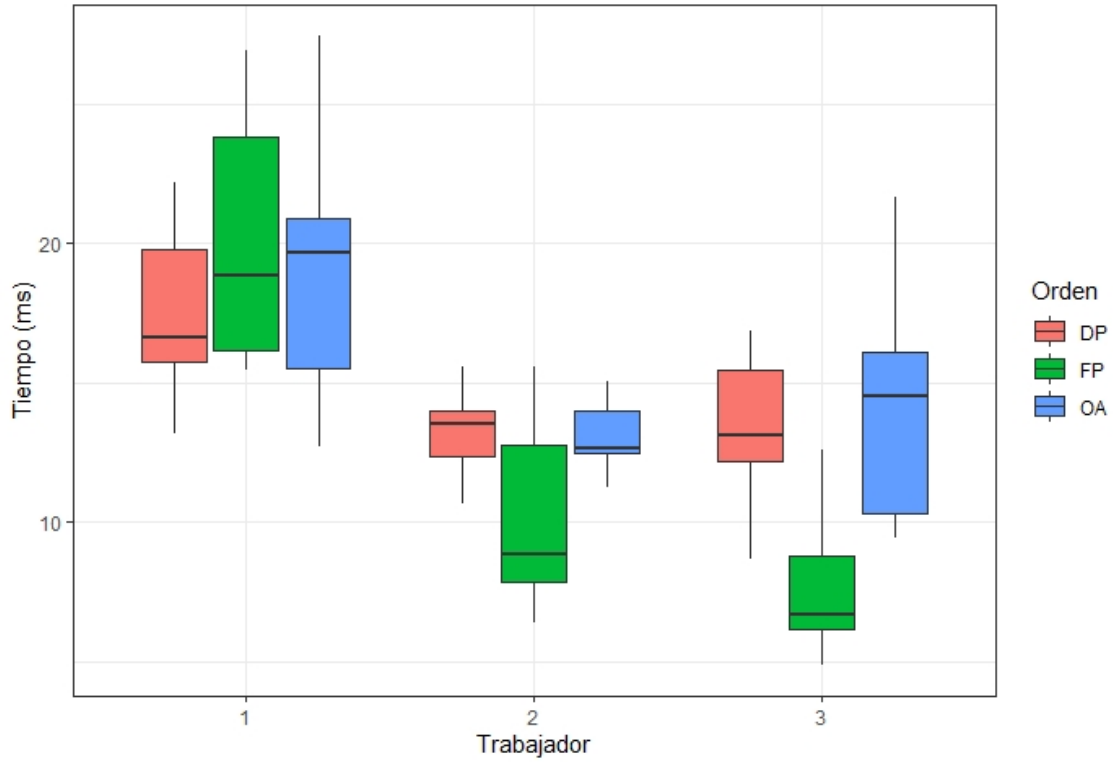


Figura 1: Tiempo que tarda cada trabajador en realizar las tareas en cada orden: difíciles primero (DP), fáciles primero (FP) y orden aleatorio (OA).

Se eligió utilizar la prueba estadística ANOVA de una vía, debido a que se quería realizar un análisis paramétrico para comparar más de 2 grupos. Se revisaron diferentes páginas en internet sobre documentación [7], y video tutoriales [2, 4] para aprender a utilizar la prueba estadística elegida [1].

Se generó el código para aplicar ANOVA de una vía a los datos de cada trabajador y también a los datos de cada tipo de ordenamiento por separado para así analizar sus medias, en la figura 2 se muestran los diagramas caja-bigote para cada **trabajador** y **ordenamiento**.

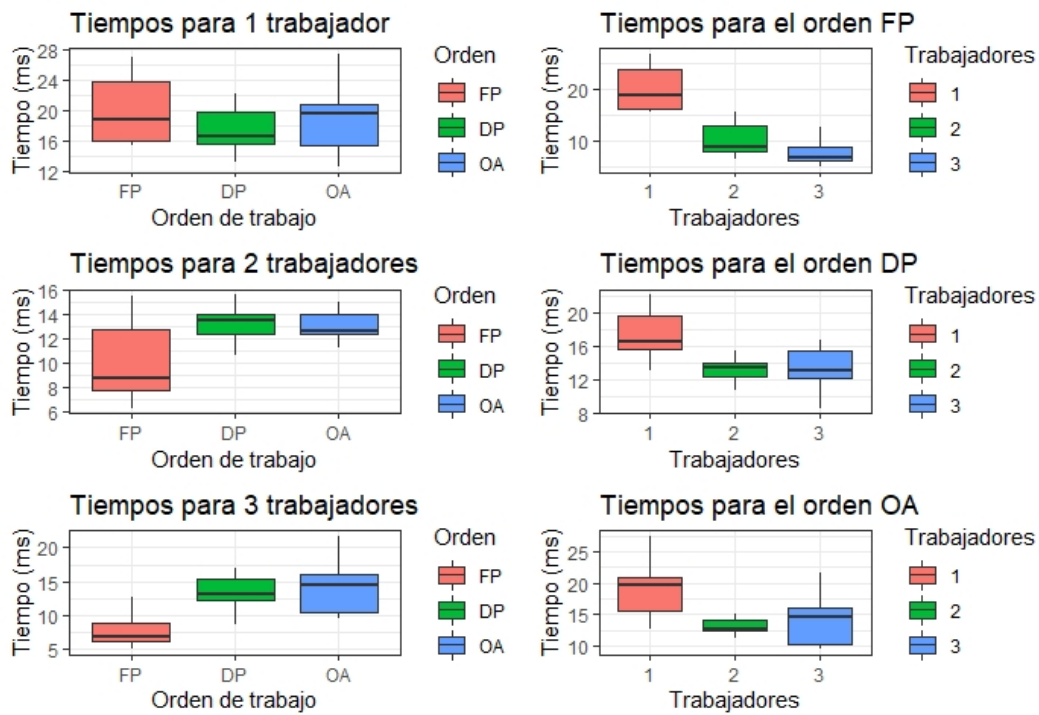
Primeramente, se revisaron los supuestos para poder aplicar la prueba estadística ANOVA. En el cuadro 3 se resumen los resultados. El supuesto outliers se refiere a la cantidad de valores atípicos que existen en los grupos, la normalidad por grupos se obtuvo con la prueba de **Shapiro Wilk**, y la homogeneidad de varianza se obtuvo con la prueba de **Levene**.

Cuadro 3: Resultados de los supuestos para aplicar la prueba estadística ANOVA.

	Trabajador 1	Trabajador 2	Trabajador 3	Orden FP	Orden DP	Orden OA
Outliers	0	0	0	0	0	0
Normalidad por grupo	FP $p = 0,190$	FP $p = 0,182$	FP $p = 0,083$	T1 $p = 0,190$	T1 $p = 0,648$	T1 $p = 0,875$
	DP $p = 0,648$	DP $p = 0,826$	DP $p = 0,807$	T2 $p = 0,182$	T2 $p = 0,826$	T2 $p = 0,610$
	OA $p = 0,875$	OA $p = 0,610$	OA $p = 0,249$	T3 $p = 0,083$	T3 $p = 0,807$	T3 $p = 0,249$
Homogeneidad de varianza	$p = 0,356$	$p = 0,068$	$p = 0,135$	$p = 0,251$	$p = 0,421$	$p = 0,018$

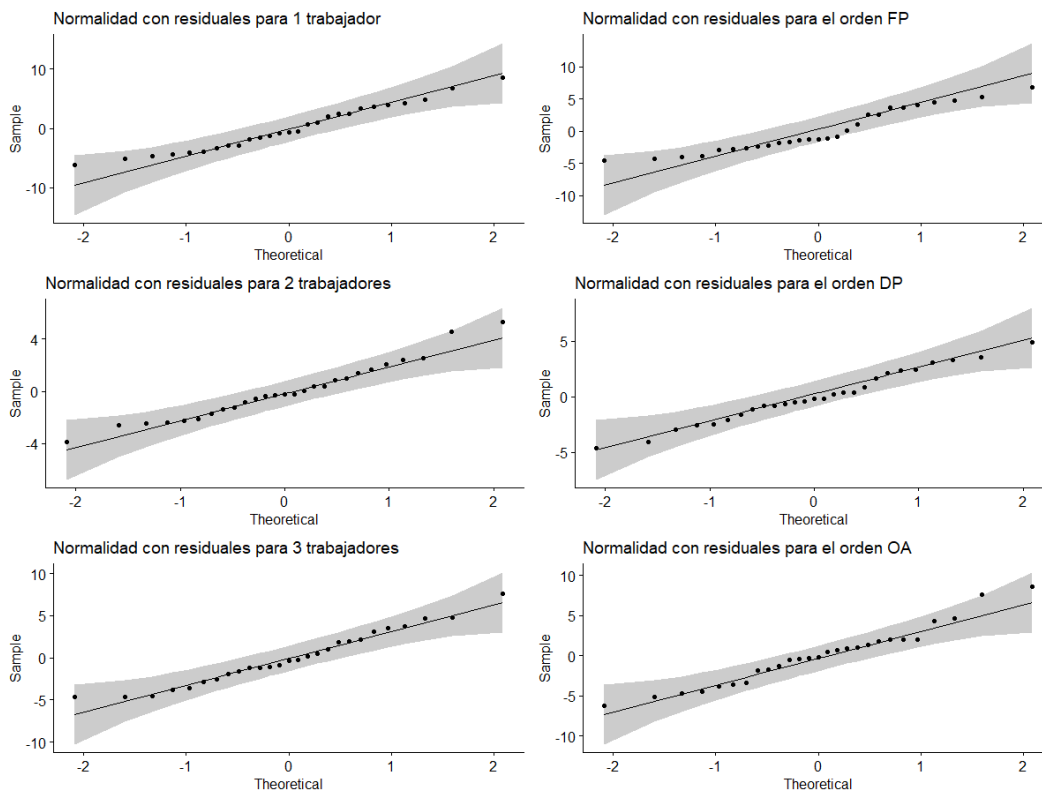
En el cuadro 3 se puede ver que los valores de p son mayores a 0,05 por lo cual se cumplen los supuestos, excepto en la homogeneidad de varianza del **Orden OA**. Para el ordenamiento **OA** se aplicó la prueba estadística ANOVA de Welch, la cual puede utilizarse cuando no se tiene varianza homogénea.

Figura 2: Tiempos para cada trabajador y cada orden por separado.



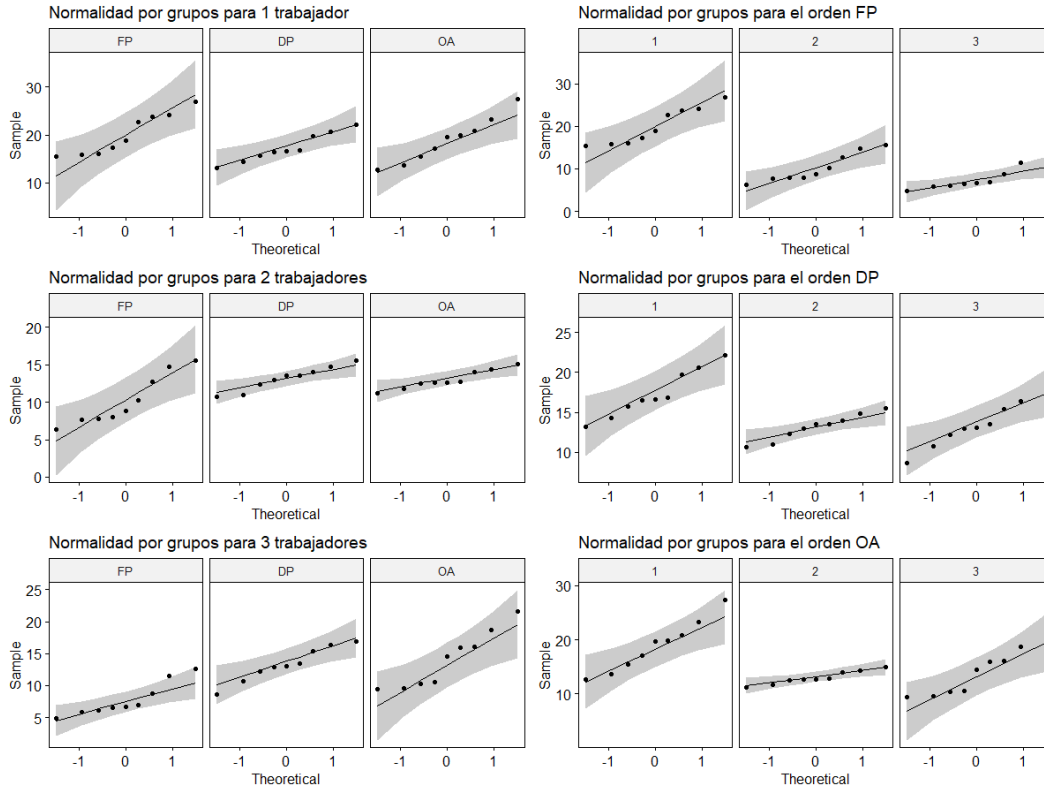
También se calculó la normalidad por residuales y los resultados se muestran en la figura 3.

Figura 3: Normalidad por residuales.



Los resultados de la normalidad por grupos se muestran en la figura 4.

Figura 4: Normalidad por grupos separados.



A continuación, en el cuadro 4 se muestran los resultados de la prueba estadística ANOVA.

Cuadro 4: Resultados de la prueba estadística ANOVA de una vía.

Grupo	Valor de p	Comparaciones múltiples de Tukey de medias (nivel de confianza del 0.95)
Trabajador 1	0,354	DP-FP: $p = 0,323$ / OA-FP: $p = 0,801$ / OA-DP: $p = 0,686$
Trabajador 2	0,0177	DP-FP: $p = 0,028$ / OA-FP: $p = 0,039$ / OA-DP: $p = 0,987$
Trabajador 3	0,0008	DP-FP: $p = 0,004$ / OA-FP: $p = 0,001$ / OA-DP: $p = 0,872$
Orden FP	$2,04e - 07$	T2-T1: $p = 0,000$ / T3-T1: $p = 0,000$ / T3-T2: $p = 0,319$
Orden DP	0,0020	T2-T1: $p = 0,004$ / T3-T1: $p = 0,006$ / T3-T2: $p = 0,989$
Orden OA	0,013	T2-T1: $p = 0,013$ / T3-T1: $p = 0,096$ / T3-T2: $p = 0,755$

Como se mencionó anteriormente para el grupo **Orden OA** se utilizó la prueba estadística ANOVA de Welch.

Si tenemos la hipótesis nula siguiente:

Hipótesis Nula: Las medias de todos los grupos son iguales.

En los resultados mostrados en el cuadro 4 podemos notar que $p < 0,05$ en la mayoría de los grupos, por lo tanto podemos decir que:

Se rechaza la hipótesis nula.

Hipótesis alternativa: Las medias son estadísticamente significativas en los grupos **Trabajador 2**, **Trabajador 3**, **Orden FP**, **Orden DP** y **Orden OA**, es decir que en estos grupos las medias sí son diferentes. Solamente en el grupo **Trabajador 1** se tienen medias no significativas.

Enseguida se muestra el código generado en **RStudio** para esta práctica [5].

```

1 #PRUEBA ESTADISTICA ANOVA
2 install.packages("tidyverse")
3 install.packages("ggpubr")
4 install.packages("rstatix")
5 install.packages("rapportools")
6
7 library(tidyverse)
8 library(ggpubr)
9 library(rstatix)
10 library(rapportools)
11 library(readr)
12 library(ggplot2)
13 library(gridExtra)
14
15 ruta_csv = "C:\\Users\\beren\\OneDrive\\Escritorio\\Maestr a\\2do semestre\\5. Simulaci n
    computacional de nanomateriales\\Tareas\\Tarea_3\\misdatos.csv"
16
17 datos = read.csv(ruta_csv)
18
19 names(datos) <- c("Replica", "Trabajador", "Orden", "Tiempo")
20
21 datos$Trabajador= as.factor(datos$Trabajador) #crear vector a partir del dataframe
22 datos$Orden = as.factor(datos$Orden) #crear vector a partir del dataframe
23
24 ggplot(data = datos, aes(x = Trabajador, y = Tiempo, fill = Orden)) +
25   geom_boxplot() + theme_bw() + labs(y = "Tiempo (ms)")
26
27
28 library(readxl)
29 ruta_excel = "C:/Users/beren/OneDrive/Escritorio/Maestr a/2do semestre/5. Simulaci n
    computacional de nanomateriales/Tareas/Tarea_3/misdatos.xlsx"
30 misdatos <- read_excel(ruta_excel)
31 excel_sheets(ruta_excel)
32
33 trabajador1 = read_excel(ruta_excel, sheet = "Trabajador1")
34 trabajador2 = read_excel(ruta_excel, sheet = "Trabajador2")
35 trabajador3 = read_excel(ruta_excel, sheet = "Trabajador3")
36 ordenfp = read_excel(ruta_excel, sheet = "FP")
37 ordendp = read_excel(ruta_excel, sheet = "DP")
38 ordenoa = read_excel(ruta_excel, sheet = "OA")
39
40
41 #ANALISIS DESCRIPTIVO
42 #Un trabajador
43 trabajador1 = trabajador1 %>%
44   rstatix::reorder_levels(Orden, order = c("FP", "DP", "OA"))
45
46 trabajador1 %>%
47   group_by(Orden) %>%
48   get_summary_stats(Tiempo, type = "mean_sd")
49 #Dos trabajadores
50 trabajador2 = trabajador2 %>%
51   rstatix::reorder_levels(Orden, order = c("FP", "DP", "OA"))
52
53 trabajador2 %>%
54   group_by(Orden) %>%
55   get_summary_stats(Tiempo, type = "mean_sd")
56 #Tres trabajadores
57 trabajador3 = trabajador3 %>%
58   rstatix::reorder_levels(Orden, order = c("FP", "DP", "OA"))
59
60 trabajador3 %>%
61   group_by(Orden) %>%
62   get_summary_stats(Tiempo, type = "mean_sd")
63 #Orden FP
64 ordenfp = ordenfp %>%
65   rstatix::reorder_levels(Trabajadores, order = c("1", "2", "3"))
66
67 ordenfp %>%
68   group_by(Trabajadores) %>%

```

```

69   get_summary_stats(Tiempo, type = "mean_sd")
70 #Orden DP
71 ordendp = ordendp %>%
72   rstatix::reorder_levels(Trabajadores, order = c("1", "2", "3"))
73
74 ordendp %>%
75   group_by(Trabajadores) %>%
76   get_summary_stats(Tiempo, type = "mean_sd")
77 #Orden OA
78 ordenoa = ordenoa %>%
79   rstatix::reorder_levels(Trabajadores, order = c("1", "2", "3"))
80
81 ordenoa %>%
82   group_by(Trabajadores) %>%
83   get_summary_stats(Tiempo, type = "mean_sd")
84
85 #VISUALIZAR DATOS
86 #Un trabajador
87 g1 = ggplot(data = trabajador1, aes(x = Orden, y = Tiempo, fill = Orden)) +
88   geom_boxplot() + theme_bw() + labs(x = "Orden de trabajo", y = "Tiempo (ms)", title = 'Tiempos
      para 1 trabajador')
89
90 #Dos trabajadores
91 g2 = ggplot(data = trabajador2, aes(x = Orden, y = Tiempo, fill = Orden)) +
92   geom_boxplot() + theme_bw() + labs(x = "Orden de trabajo", y = "Tiempo (ms)", title = 'Tiempos
      para 2 trabajadores')
93
94 #Tres trabajadores
95 g3 = ggplot(data = trabajador3, aes(x = Orden, y = Tiempo, fill = Orden)) +
96   geom_boxplot() + theme_bw() + labs(x = "Orden de trabajo", y = "Tiempo (ms)", title = 'Tiempos
      para 3 trabajadores')
97
98 #Orden FP
99 g4 = ggplot(data = ordenfp, aes(x = Trabajadores, y = Tiempo, fill = Trabajadores)) +
100   geom_boxplot() + theme_bw() + labs(x = "Trabajadores", y = "Tiempo (ms)", title = 'Tiempos para
      el orden FP')
101
102 #Orden DP
103 g5 = ggplot(data = ordendp, aes(x = Trabajadores, y = Tiempo, fill = Trabajadores)) +
104   geom_boxplot() + theme_bw() + labs(x = "Trabajadores", y = "Tiempo (ms)", title = 'Tiempos para
      el orden DP')
105
106 #Orden OA
107 g6 = ggplot(data = ordenoa, aes(x = Trabajadores, y = Tiempo, fill = Trabajadores)) +
108   geom_boxplot() + theme_bw() + labs(x = "Trabajadores", y = "Tiempo (ms)", title = 'Tiempos para
      el orden OA')
109
110 grid.arrange(g1, g4, g2, g5, g3, g6, ncol = 2)
111
112 #SUPUESTOS PARA ANOVA
113
114 #1: OUTLIERS
115 #Un trabajador
116 trabajador1 %>%
117   group_by(Orden) %>%
118   identify_outliers(Tiempo)
119
120 #Dos trabajadores
121 trabajador2 %>%
122   group_by(Orden) %>%
123   identify_outliers(Tiempo)
124
125 #Tres trabajadores
126 trabajador3 %>%
127   group_by(Orden) %>%
128   identify_outliers(Tiempo)
129
130 #Orden FP
131 ordenfp %>%
132   group_by(Trabajadores) %>%
133   identify_outliers(Tiempo)
134
135 #Orden DP

```

```

136 ordendp %>%
137   group_by(Trabajadores) %>%
138   identify_outliers(Tiempo)
139
140 #Orden OA
141 ordenoa %>%
142   group_by(Trabajadores) %>%
143   identify_outliers(Tiempo)
144
145 #2: NORMALIDAD CON RESIDUALES
146 #Un trabajador
147 normalidad1 = lm(Tiempo ~ Orden, data = trabajador1)
148 head(trabajador1)
149 head(normalidad1$fitted.values)
150 head(normalidad1$residuals)
151
152 n1 = ggqqplot(residuals(normalidad1))+
153   labs(title = 'Normalidad con residuales para 1 trabajador')
154 shapiro_test(residuals(normalidad1))
155
156 trabajador1 %>%
157   group_by(Orden) %>%
158   shapiro_test(Tiempo)
159
160 #Dos trabajadores
161 normalidad2 = lm(Tiempo ~ Orden, data = trabajador2)
162 head(trabajador2)
163 head(normalidad2$fitted.values)
164 head(normalidad2$residuals)
165
166 n2 = ggqqplot(residuals(normalidad2))+
167   labs(title = 'Normalidad con residuales para 2 trabajadores')
168 shapiro_test(residuals(normalidad2))
169
170 trabajador2 %>%
171   group_by(Orden) %>%
172   shapiro_test(Tiempo)
173
174 #Tres trabajadores
175 normalidad3 = lm(Tiempo ~ Orden, data = trabajador3)
176 head(trabajador3)
177 head(normalidad3$fitted.values)
178 head(normalidad3$residuals)
179
180 n3 = ggqqplot(residuals(normalidad3))+
181   labs(title = 'Normalidad con residuales para 3 trabajadores')
182 shapiro_test(residuals(normalidad3))
183
184 trabajador3 %>%
185   group_by(Orden) %>%
186   shapiro_test(Tiempo)
187
188 #Orden FP
189 normalidad4 = lm(Tiempo ~ Trabajadores, data = ordenfp)
190 head(ordenfp)
191 head(normalidad4$fitted.values)
192 head(normalidad4$residuals)
193
194 n4 = ggqqplot(residuals(normalidad4))+
195   labs(title = 'Normalidad con residuales para el orden FP')
196 shapiro_test(residuals(normalidad4))
197
198 ordenfp %>%
199   group_by(Trabajadores) %>%
200   shapiro_test(Tiempo)
201
202 #Orden DP
203 normalidad5 = lm(Tiempo ~ Trabajadores, data = ordendp)
204 head(ordendp)
205 head(normalidad5$fitted.values)
206 head(normalidad5$residuals)
207
208 n5 = ggqqplot(residuals(normalidad5))+

```



```

209 labs(title = 'Normalidad con residuales para el orden DP')
210 shapiro_test(residuals(normalidad5))
211
212 ordendp %>%
213   group_by(Trabajadores) %>%
214   shapiro_test(Tiempo)
215
216 #Orden OA
217 normalidad6 = lm(Tiempo ~ Trabajadores, data = ordenoa)
218 head(ordenoa)
219 head(normalidad6$fitted.values)
220 head(normalidad6$residuals)
221
222 n6 = ggqqplot(residuals(normalidad6))+
223   labs(title = 'Normalidad con residuales para el orden OA')
224 shapiro_test(residuals(normalidad6))
225
226 ordenoa %>%
227   group_by(Trabajadores) %>%
228   shapiro_test(Tiempo)
229
230 grid.arrange(n1, n4, n2, n5, n3, n6, ncol = 2)
231
232 #NORMALIDAD PARA CADA GRUPO POR SEPARADO
233 #Un trabajador
234 m1 = ggqqplot(data = trabajador1, x = "Tiempo", facet.by = "Orden")+
235   labs(title = 'Normalidad por grupos para 1 trabajador')
236 trabajador1 %>%
237   group_by(Orden) %>%
238   shapiro_test(Tiempo)
239
240 #Dos trabajadores
241 m2 = ggqqplot(data = trabajador2, x = "Tiempo", facet.by = "Orden")+
242   labs(title = 'Normalidad por grupos para 2 trabajadores')
243 trabajador2 %>%
244   group_by(Orden) %>%
245   shapiro_test(Tiempo)
246
247 #Tres trabajadores
248 m3 = ggqqplot(data = trabajador3, x = "Tiempo", facet.by = "Orden")+
249   labs(title = 'Normalidad por grupos para 3 trabajadores')
250 trabajador3 %>%
251   group_by(Orden) %>%
252   shapiro_test(Tiempo)
253
254 #Orden FP
255 m4 = ggqqplot(data = ordenfp, x = "Tiempo", facet.by = "Trabajadores")+
256   labs(title = 'Normalidad por grupos para el orden FP')
257 ordenfp %>%
258   group_by(Trabajadores) %>%
259   shapiro_test(Tiempo)
260
261 #Orden DP
262 m5 = ggqqplot(data = ordendp, x = "Tiempo", facet.by = "Trabajadores")+
263   labs(title = 'Normalidad por grupos para el orden DP')
264 ordendp %>%
265   group_by(Trabajadores) %>%
266   shapiro_test(Tiempo)
267
268 #Orden OA
269 m6 = ggqqplot(data = ordenoa, x = "Tiempo", facet.by = "Trabajadores")+
270   labs(title = 'Normalidad por grupos para el orden OA')
271 ordenoa %>%
272   group_by(Trabajadores) %>%
273   shapiro_test(Tiempo)
274
275 grid.arrange(m1, m4, m2, m5, m3, m6, ncol = 2)
276
277 #HOMOGENEIDAD DE VARIANZAS CON PRUEBA DE LEVENE
278 #Un trabajador
279 trabajador1 %>%
280   levene_test(Tiempo~Orden)
281

```

```

282 #Dos trabajadores
283 trabajador2 %>%
284   levene_test(Tiempo~Orden)
285
286 #Tres trabajadores
287 trabajador3 %>%
288   levene_test(Tiempo~Orden)
289
290 #Orden FP
291 ordenfp %>%
292   levene_test(Tiempo~Trabajadores)
293
294 #Orden DP
295 ordendp %>%
296   levene_test(Tiempo~Trabajadores)
297
298 #Orden OA
299 ordenoa %>%
300   levene_test(Tiempo~Trabajadores)
301
302 #ANOVA
303 #Un trabajador
304 view(trabajador1)
305 trabajador1
306 attach(trabajador1)
307 names(trabajador1)
308 class(Orden)
309 class(Tiempo)
310 summary(trabajador1)
311 boxplot(Tiempo~Orden)
312 aov(Tiempo~Orden)
313 anova1 = aov(Tiempo~Orden)
314 summary(anova1)
315 TukeyHSD(anova1)
316
317 #Dos trabajadores
318 view(trabajador2)
319 trabajador2
320 attach(trabajador2)
321 names(trabajador2)
322 class(Orden)
323 class(Tiempo)
324 summary(trabajador2)
325 boxplot(Tiempo~Orden)
326 aov(Tiempo~Orden)
327 anova2 = aov(Tiempo~Orden)
328 summary(anova2)
329 TukeyHSD(anova2)
330
331 #Tres trabajadores
332 view(trabajador3)
333 trabajador3
334 attach(trabajador3)
335 names(trabajador3)
336 class(Orden)
337 class(Tiempo)
338 summary(trabajador3)
339 boxplot(Tiempo~Orden)
340 aov(Tiempo~Orden)
341 anova3 = aov(Tiempo~Orden)
342 summary(anova3)
343 TukeyHSD(anova3)
344
345 #Orden FP
346 view(ordenfp)
347 ordenfp
348 attach(ordenfp)
349 names(ordenfp)
350 class(Trabajadores)
351 class(Tiempo)
352 summary(ordenfp)
353 boxplot(Tiempo~Trabajadores)
354 aov(Tiempo~Trabajadores)

```

```

355 anova4 = aov(Tiempo~Trabajadores)
356 summary(anova4)
357 TukeyHSD(anova4)
358
359 #Orden DP
360 view(ordendp)
361 ordendp
362 attach(ordendp)
363 names(ordendp)
364 class(Trabajadores)
365 class(Tiempo)
366 summary(ordendp)
367 boxplot(Tiempo~Trabajadores)
368 aov(Tiempo~Trabajadores)
369 anova5 = aov(Tiempo~Trabajadores)
370 summary(anova5)
371 TukeyHSD(anova5)
372
373 #ANOVA DE WELCH
374 #Orden OA
375 anova6 = ordenoa %>% welch_anova_test(Tiempo~Trabajadores)
376 anova6
377 comp_mult = ordenoa %>% games_howell_test(Tiempo~Trabajadores)
378 comp_mult

```

Listing 2: Código para la prueba estadística ANOVA y ANOVA Welch.

Después de revisar otras páginas de internet y video tutoriales [3], encontré que para analizar el efecto de los trabajadores y el orden de los trabajos en una sola prueba estadística se puede utilizar ANOVA para dos vías o factores. Para esta prueba también se revisó la homogeneidad de varianza con una prueba de **Levene** y la normalidad con una prueba de **Shapiro Wilk**. Los resultados se muestran en el cuadro 5, e indican que ambos resultados no son significativos, es decir que si hay normalidad y varianza homogénea.

Cuadro 5: Resultados de pruebas de varianza y normalidad.

Prueba	p
Levene	0,1648
Shapiro Wilk	0,1069

En el código se realizaron dos modelos para la prueba estadística, el **Modelo 1** de ANOVA que es para solo considerar efectos principales de las variables independientes **Trabajador** y **Orden**, y el **Modelo 2** toma en cuenta las interacciones que existen entre dichas variables. Si el resultado del **Modelo 2** es significativo se debe realizar un análisis de ANOVA **Tipo III**, y de lo contrario se utiliza un **Tipo II**. En los cuadros 6 y 7 se presentan los resultados de los modelos, los cuales indican que la interacción entre las dos variables si es significativa.

Cuadro 6: Resultados del Modelo 1.

Variables	p
Trabajador	$1,758e - 09$
Orden	0,118

Cuadro 7: Resultados del Modelo 2.

Variables	p
Trabajador	$5,845e - 10$
Orden	0,09832
Orden:Trabajador	0,03684

Realizando la prueba estadística de ANOVA de dos factores Tipo III se obtienen los resultados mostrados en el cuadro 8.

Cuadro 8: Resultados de pruebas estadística ANOVA de dos factores

Variables	p	Interpretación
Trabajador	$5,845e - 10$	$p < 0,05$ entonces es significativa
Orden	0,09832	$p > 0,05$ entonces no es significativa
Orden:Trabajador	0,036842	$p < 0,05$ entonces es significativa

Hipótesis nula: Las medias son iguales en todos los grupos.

Hipótesis alternativa: Se rechaza la hipótesis nula. Si existen diferencias en los tiempos de los trabajadores, es decir que el tiempo se ve afectado por el hecho de asignar más núcleos en el cluster. Y en cuanto al orden en que se realizan los trabajos si afecta de manera significativa en los tiempos cuando se tiene la interacción con los diferentes trabajadores.

El código generado para la prueba estadística ANOVA con dos factores se muestra a continuación:

```

1 install.packages("car")
2 install.packages("reshape")
3 install.packages("multcomp")
4 install.packages("pastecs")
5 install.packages("WRS")
6
7 library(ggplot2)
8 library(car)
9 library(reshape)
10 library(multcomp)
11 library(pastecs)
12 library(WRS)
13 library(tidyverse)
14 library(ggpubr)
15 library(rstatix)
16 library(rapportools)
17 library(readr)
18 library(ggplot2)
19 library(gridExtra)
20
21 names(misdatos2)
22
23 misdatos2$Trabajador= as.character(misdatos2$Trabajador) #crear vector a partir del dataframe
24 misdatos2$Orden = as.character(misdatos2$Orden) #crear vector a partir del dataframe
25 misdatos2$Tiempo = as.numeric(misdatos2$Tiempo) #crear vector a partir del dataframe
26
27 attach(misdatos2)
28 names(misdatos2)
29
30 class(Trabajador)
31 class(Orden)
32 class(Tiempo)
33 factor(Trabajador)
34 factor(Orden)
35
36 leveneTest(misdatos2$Tiempo, interaction(misdatos2$Orden, misdatos2$Trabajador), center = median)
37
38 Model1 = aov(Tiempo~Trabajador+Orden, data = misdatos2) #Modelo con efectos principales
39 anova(Model1)
40
41 Model2 = aov(Tiempo~Orden*Trabajador, data = misdatos2) #Modelo con efectos principales con
    interacci n
42 anova(Model2)
43
44 plot(Model1)
45
46 aov_residuals = residuals(object = Model1)
47 shapiro.test(x=aov_residuals)
48
49 Anova(Model2, Type = "III")

```

Listing 3: Código para prueba estadística ANOVA de dos vías.

3. Conclusión

Con base en los resultados obtenidos de las diferentes pruebas estadísticas ANOVA que realicé puedo concluir que cuando se asignan más núcleos en el cluster el trabajo o la tarea se puede realizar más rápidamente. En general el desarrollo de la práctica me aportó conocimiento sobre las diferentes pruebas estadísticas, los requerimientos que tiene cada una de ellas y sus aplicaciones.

Referencias

- [1] José Antonio: Estadística Aplicada. Anova de welch en rstudio, 2020. URL <https://youtu.be/2pf0kUbigr0>.
- [2] Lourdes Cuellar. Cómo hacer un anova en rstudio., 2019. URL <https://youtu.be/3wbuVnKGkxg>.
- [3] Lourdes Cuellar. Anova de dos factores en rstudio two way anova, 2019. URL https://youtu.be/4q_mfYZV7-8.
- [4] Epidemiología e Investigación en Salud. Pruebas estadísticas, 2020. URL <https://youtu.be/M1MQ9yD9rAg>.
- [5] Overleaf. Code listing, 2021. URL https://es.overleaf.com/learn/latex/Code_listing.
- [6] Elisa Schaeffer. Simulación: hilos y primos (p3 ad21), 2021. URL https://www.youtube.com/watch?v=IJpQheI6jSM&list=LL&index=8&t=209s&ab_channel=ElisaSchaeffer.
- [7] Elisa Schaeffer. Probabilidad y estadística, 2021. URL <https://elisa.dyndns-web.com/teaching/prob/prob.html>.