Tarea 5 Optimización de Flujo en Redes

1985274

30 de abril de 2019

En este trabajo se presenta un análisis en el que se ha seleccionado un algoritmo generador de grafos y a su vez se ha escogido una implementación de algoritmo de flujo máximo. Para esto se ha utilizado el lenguaje **Python** en su versión 3.7 [9], el editor de código Spyder en su versión 3.3.1 y el editor Texmaker 5.0.2 para redactar el documento. Se utilizan además las librerías **networkX** 1.5 [6], **Matplotlib** [4], **Numpy** [7], la librería **Panda** [8], la librería **Scipy** [11], la librería **Seaborn** [5] para graficar, la librería **Researchcpy** [1] para producir pandas `Dataframe` de pruebas estadísticas y la librería **Pingouin** [10] para realizar el análisis de varianza de un factor (ANOVA).

1. Algoritmo generador de grafo

Se utiliza el generador *Watts Strogatz graph* y como algoritmo de flujo máximo: *maximun flow value*.

- Algoritmo *Watts Strogatz graph*: Este es un generador aleatorio que recibe cuatro parámetros que son: n el número de nodos, k cada nodo se une con sus k vecinos en una topología de anillo, p que es la probabilidad de volver a establecer una arista, *seed* que es la semilla o indicador del estado de generación de números aleatorios. Este algoritmo primero crea un anillo sobre n nodos y luego cada nodo en el anillo es conectado con sus vecinos más cercanos. Luego se crean accesos directos reemplazando algunas aristas. Se seleccionó este algoritmo porque es posible representar una red de transporte de una terminal aérea de un país con varias aerolíneas, donde cada una enlaza este país con otros países de ida y vuelta, por lo que se obtiene un grafo no dirigido. En este caso los países son los nodos y las aristas son las aerolíneas.

2. Algoritmo de flujo máximo

- Algoritmo *maximum flow value*: Esta función calcula el valor del flujo máximo a costo mínimo entre la fuente y el sumidero en grafos capacitados. El algoritmo recibe varios parámetros como son: G que es el grafo, s que es el nodo origen o fuente para el flujo, t que es el nodo destino o sumidero

para el flujo, *capacidad* que es el parámetro que indica la cantidad de flujo que puede soportar la arista. Al aplicar este algoritmo se obtiene un diccionario que contiene el valor del flujo que pasó por cada arista. Se generaron 5 grafos de 20 nodos cada uno, y sobre ellos se ejecutó al algoritmo de flujo máximo *maximum flow value*.

2.1. Fragmento del conjunto de datos obtenido

Cuadro 1: Fragmento del conjunto de datos obtenido

Grafo	Fuente	Sumidero	Media	Mediana	Std	FlujMax	Grado	CoefAg	CentCe	CenCar
1	4	0	0.022	0.0041	0.0231	41	4	0.5	0.365	0.045
1	4	1	0.023	0.0065	0.0229	41	4	0.5	0.365	0.045
1	4	2	0.022	0.0065	0.0219	41	4	0.5	0.365	0.045
1	4	3	0.024	0.0068	0.0231	41	4	0.5	0.365	0.045
1	4	5	0.021	0.0041	0.0215	37	4	0.5	0.365	0.045
1	4	6	0.021	0.0066	0.0189	41	4	0.5	0.365	0.045
1	4	7	0.021	0.0051	0.0204	41	4	0.5	0.365	0.045
1	4	8	0.020	0.0054	0.0205	41	4	0.5	0.365	0.045
1	4	9	0.021	0.0085	0.0191	41	4	0.5	0.365	0.045
1	4	10	0.021	0.0059	0.0200	41	4	0.5	0.365	0.045
1	4	11	0.024	0.0044	0.0252	41	4	0.5	0.365	0.045
1	4	12	0.024	0.0060	0.0234	41	4	0.5	0.365	0.045
1	4	13	0.022	0.0072	0.0214	39	4	0.5	0.365	0.045
1	4	14	0.021	0.0074	0.0200	41	4	0.5	0.365	0.045
1	4	15	0.024	0.0091	0.0225	41	4	0.5	0.365	0.045
1	4	16	0.021	0.0058	0.0204	41	4	0.5	0.365	0.045
1	4	17	0.021	0.0062	0.0201	41	4	0.5	0.365	0.045
1	4	18	0.021	0.0070	0.0207	38	4	0.5	0.365	0.045
1	4	19	0.024	0.0073	0.0230	41	4	0.5	0.365	0.045

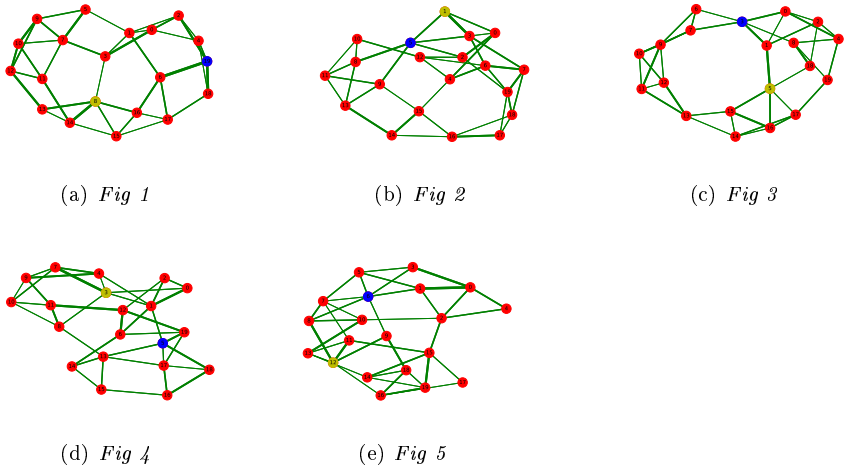
2.2. Código

2.3. Diagrama de los grafos generados

- A continuación se muestran los diagramas de los 5 grafos que se obtuvieron a partir del generador mencionado anteriormente. Todos los grafos se generaron con 20 nodos. En las imágenes del grafo se denota por colores cuál es el mejor par nodo fuente sumidero. El nodo de color amarillo denota que es un nodo fuente y el nodo azul denota que es un nodo destino o sumidero. De igual modo, el flujo se denota por colores en las aristas y la capacidad se denota por el ancho de la arista, en donde a mayor ancho de la arista mayor es la capacidad de la arista. En el caso del flujo, el color verde denota que existe flujo por esa arista, y se presenta desde un tono más claro a un tono más oscuro, para denotar que el flujo va en aumento, el color negro denota que por esa arista no transita ningún flujo.

  Para cada grafo se evaluaron 6 características estructurales, como son: el grado de los nodos, el coeficiente de agrupamiento, el coeficiente de centralidad de cercanía, el coeficiente de centralidad de carga, el coeficiente de excentricidad y el coeficiente *PageRank*.

Diagramas de los 5 grafos con el mejor par fuente - sumidero

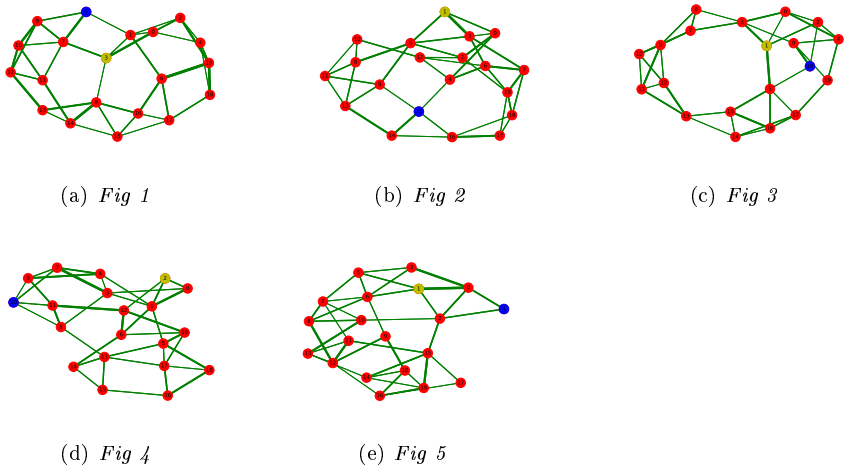


(a) *Fig 1* (b) *Fig 2* (c) *Fig 3*

(d) *Fig 4* (e) *Fig 5*

Figura 1: Ejemplos de grafos

Diagramas de los 5 grafos con el peor par fuente - sumidero



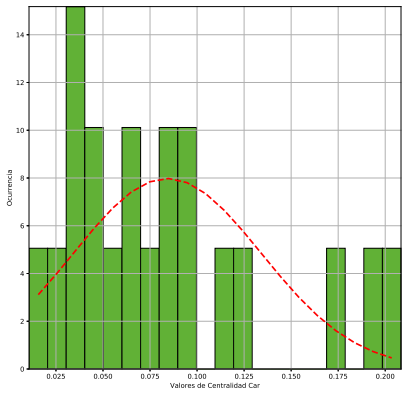
(a) *Fig 1* (b) *Fig 2* (c) *Fig 3*

(d) *Fig 4* (e) *Fig 5*

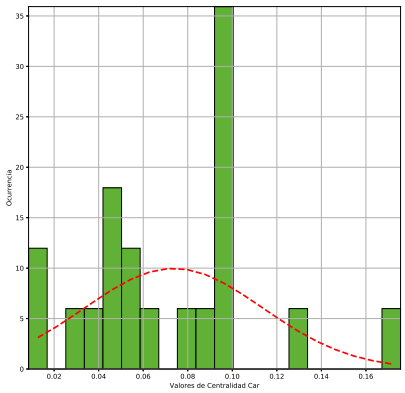
Figura 2: Ejemplos de grafos

3. Histogramas

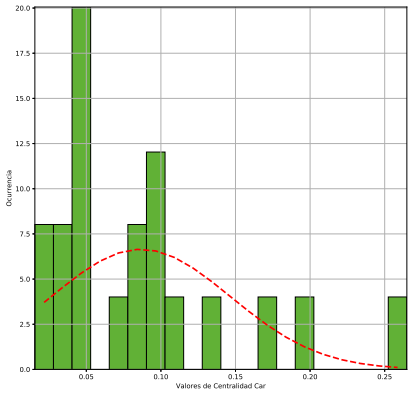
- A continuación se exponen los histogramas de cada uno de los 5 grafos.



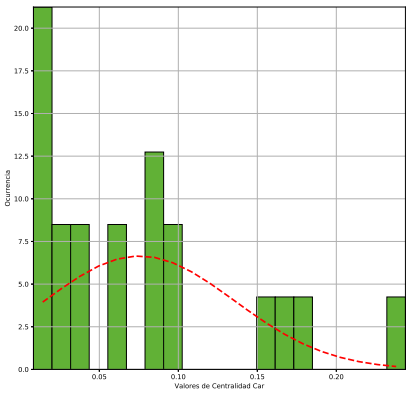
(a) *Fig 1*



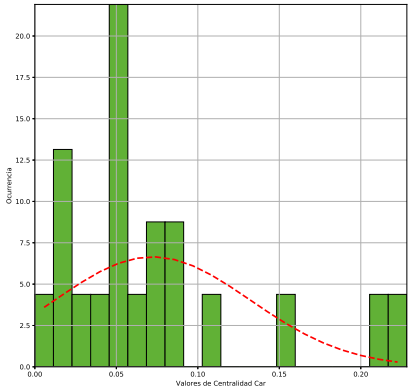
(b) *Fig 2*



(c) *Fig 3*

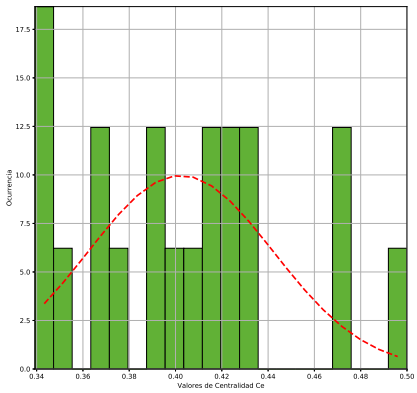


(d) *Fig 4*

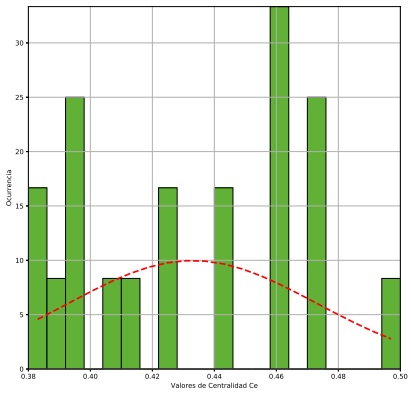


(e) *Fig 5*

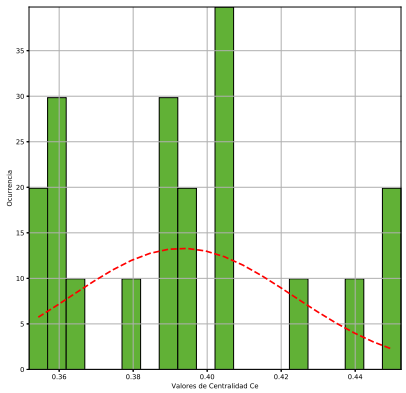
Figura 3: Histogramas que representan la característica Centralidad de carga



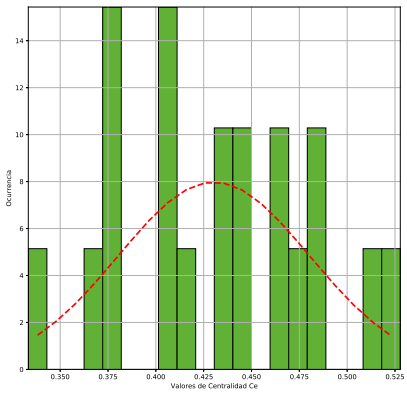
(a) *Fig 1*



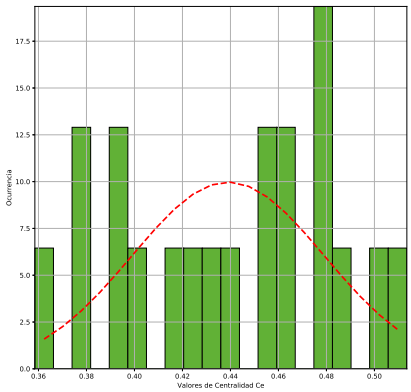
(b) *Fig 2*



(c) *Fig 3*

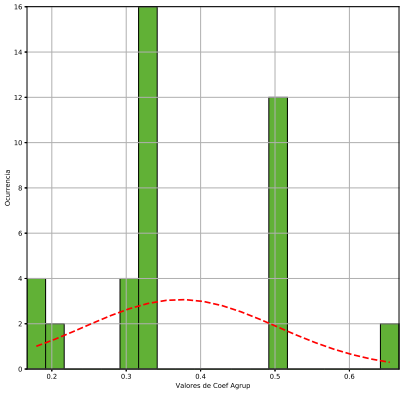


(d) *Fig 4*

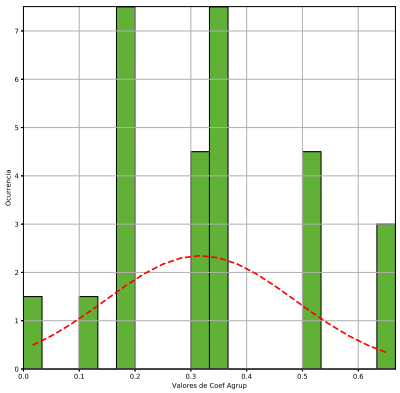


(e) *Fig 5*

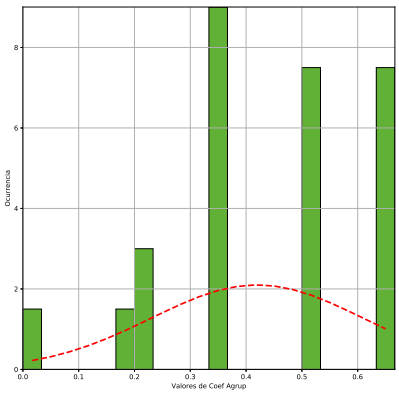
Figura 4: Histogramas que representan la característica Centralidad de cercanía



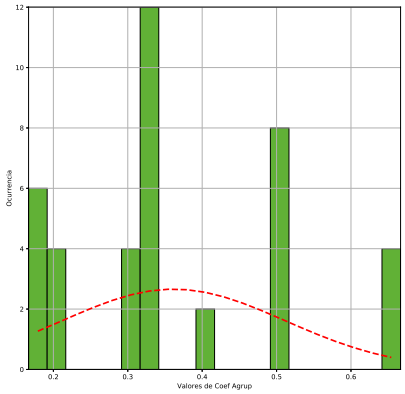
(a) *Fig 1*



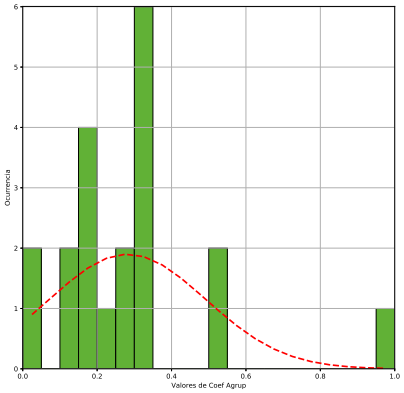
(b) *Fig 2*



(c) *Fig 3*

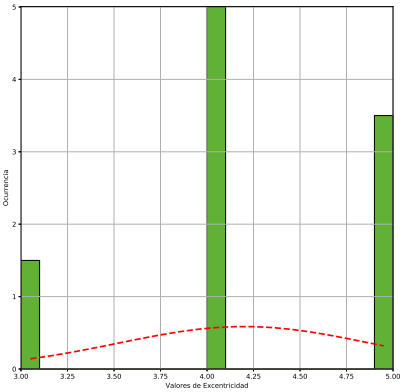


(d) *Fig 4*

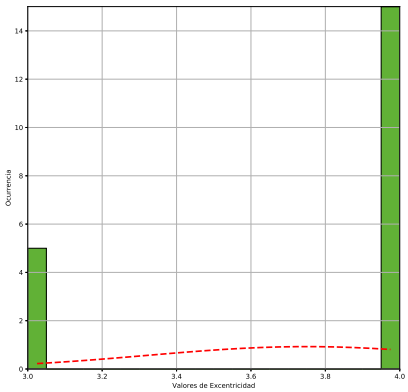


(e) *Fig 5*

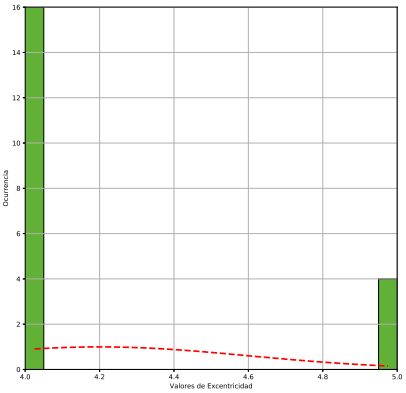
Figura 5: Histogramas que representan la característica Coeficiente de Agrupamiento



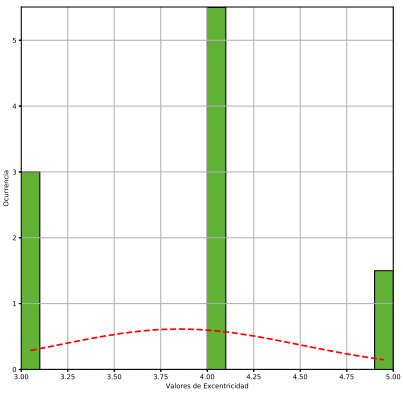
(a) *Fig 1*



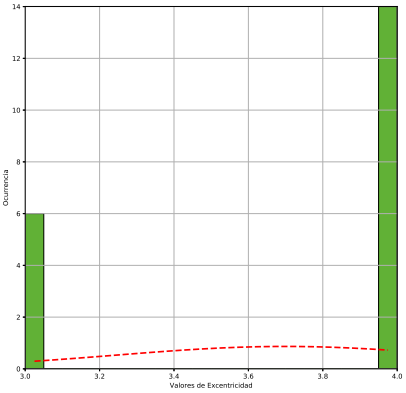
(b) *Fig 2*



(c) *Fig 3*

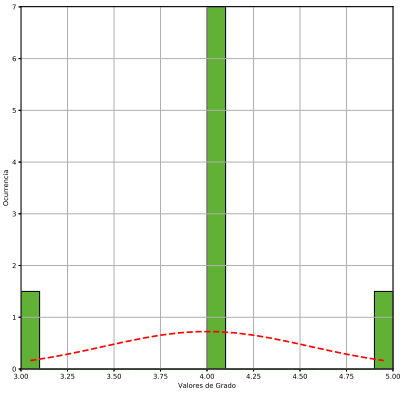


(d) *Fig 4*

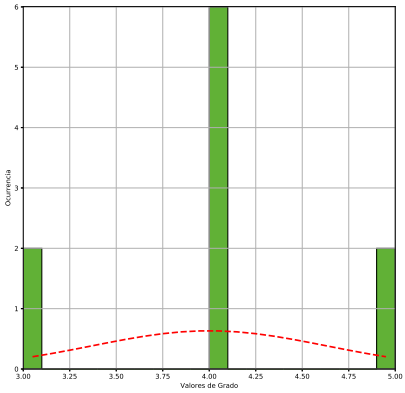


(e) *Fig 5*

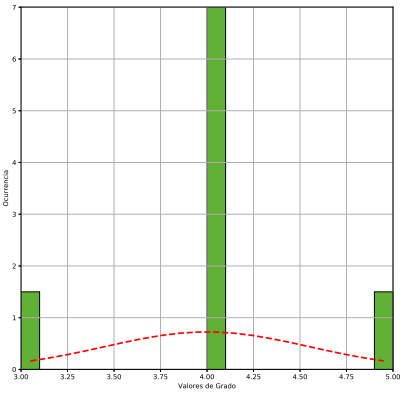
Figura 6: Histogramas que representan la característica Excentricidad



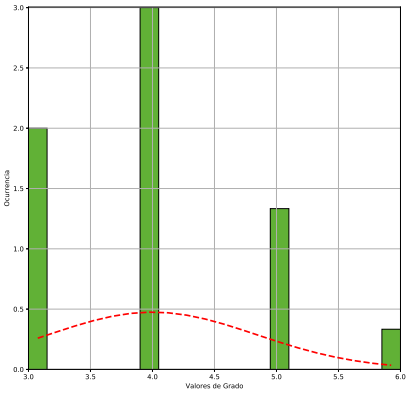
(a) *Fig 1*



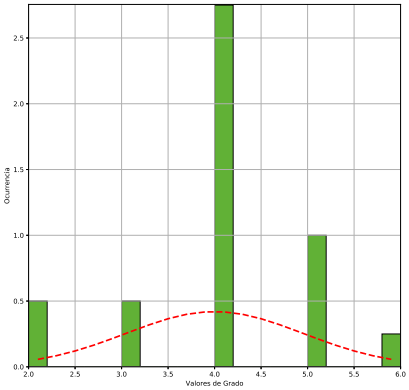
(b) *Fig 2*



(c) *Fig 3*

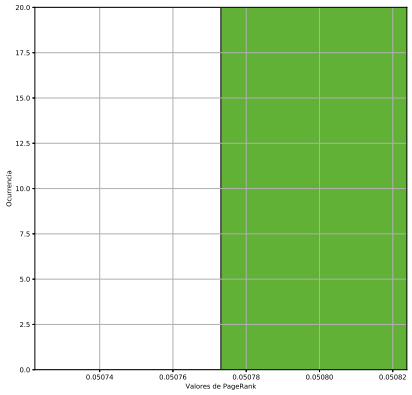


(d) *Fig 4*

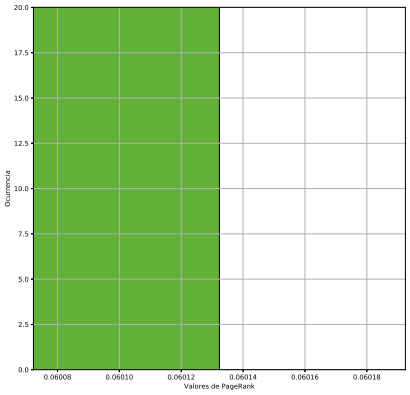


(e) *Fig 5*

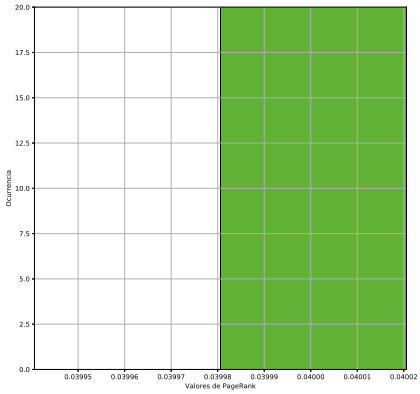
Figura 7: Histogramas que representan la característica Grado



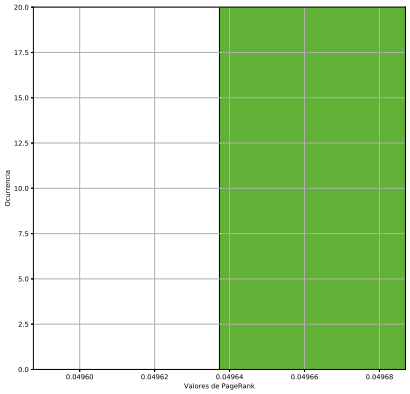
(a) *Fig 1*



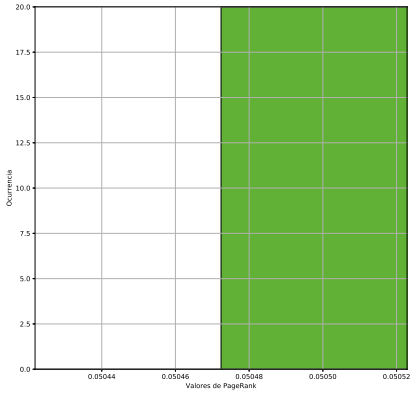
(b) *Fig 2*



(c) *Fig 3*



(d) *Fig 4*



(e) *Fig 5*

Figura 8: Histogramas que representan la característica de *PageRank*

- Como se puede apreciar en las gráfias anteriores los características no siguen una distribución normal, por lo que se utiliza la mediana para hacer el analisis estadístico. Se procedió a realizar una normalización y agrupamiento para categorizar en Bajo, Medio y Alto, obteniéndose 3 grupos.

4. Análisis de varianza ANOVA

Luego se realizó un análisis de varianza (ANOVA) de un factor sobre estos datos, con el objetivo de comparar los tratamientos en cuanto a sus medias poblacionales. Este análisis de varianza (ANOVA) permite analizar a partir de una hipótesis, si el factor tiene o no impacto en la variable tiempo [2].

La estimación se realizó mediante el modelo siguiente:

$$yij = \mu_1 + t_i + \varepsilon_{i,j} \tag{1}$$

en donde:

$$\mu_1 : \quad \text{representa la media de la población} \tag{2}$$

$$t_i : \quad \text{representa el efecto del tratamiento i} \tag{3}$$

$$\varepsilon_{i,j} : \quad \text{representa el error con una distribución normal} \tag{4}$$

Si el p valor obtenido es menor que 0.05 se rechaza la hipótesis y esto quiere decir que hay diferencias entre las medianas, la prueba de *Tukey* se realiza para mostrar las diferencias entre las medianas de los factores de [3].

A continuación se muestran los cuadros que se obtuvieron con los resultados de la prueba estadística de ANOVA.

4.1. Influencia de la característica grado en el tiempo de ejecución

Cuadro 2: Grado

Fuente	**SS**	**DF**	**MS**	**F**	p **valor**	**np2**
Grado	0.001	2	0	22.123	0.003	0.023
within	0.032	1897	0	—	—	—

- A partir del cuadro anterior se observa que existen diferencias entre el parámetro de las medianas de los grupos de factores ya que el p valor es menor que 0.05, por lo que se rechaza la hipótesis de que el grado no influye en el tiempo de ejecución. Esto se puede observar en la Figura 9 de la página 13. Debido a lo anterior se pasa a aplicar la prueba de *Tukey* para mostrar las diferencias entre las medianas, lo que se evidencia en el cuadro 8 de la página 23.

4.2. Influencia de la característica coeficiente de agrupamiento en el tiempo de ejecución

Cuadro 3: Coeficiente de agrupamiento

Fuente	SS	DF	MS	F	p valor	np2
CoefAgrupamiento	0.001	2	0.001	34.972	1.21e-106	0.036
within	0.032	1897	0	—	—	—

- Al observar el cuadro anterior se comprueba que existen diferencias entre el parámetro de las medianas de los grupos de factores ya que el p valor es menor que 0.05, por lo que se rechaza la hipótesis de que el grado no influye en el tiempo de ejecución. Esto se puede observar en la Figura 10 de la página 14. A consecuencia de lo anterior se continua con la aplicación de la prueba de *Tukey* para mostrar las diferencias entre las medianas, lo que se evidencia en el cuadro 9 de la página 24.

4.3. Influencia de la característica coeficiente de centralidad de cercanía en el tiempo de ejecución

Cuadro 4: Coeficiente de centralidad de cercanía

Fuente	SS	DF	MS	F	p valor	np2
Centralidad de cercania	0	2	0	9.501	7.84e-05	0.01
within	0.032	1897	0	—	—	—

- Del cuadro anterior se puede observar que existen diferencias entre el parámetro de las medianas de los grupos de factores ya que el p valor es menor que 0.05, por lo que se rechaza la hipótesis de que el grado no influye en el tiempo de ejecución. Esto es observable en la Figura 11 de la página 15. Como consecuencia de lo anterior, se pasa a aplicar la prueba de *Tukey* para mostrar las diferencias entre las medianas, lo que se evidencia en el cuadro 10 de la página 24.

4.4. Influencia de la característica coeficiente de centralidad de carga en el tiempo de ejecución

Cuadro 5: Coeficiente de centralidad de carga

Fuente	SS	DF	MS	F	p valor	np2
Centralidad de carga	0.002	2	0.001	57.21	7.49e-25	0.057
within	0.031	1897	0	—	—	—

- En el cuadro anterior se muestra que no existen diferencias entre el parámetro de las medianas de los grupos de factores ya que el p valor es menor que 0.05, por lo que se acepta la hipótesis de que el grado influye en el tiempo de ejecución. Esto se puede apreciar en la Figura 12 de la página 16.

4.5. Influencia de la característica coeficiente de excentricidad en el tiempo de ejecución

Cuadro 6: Coeficiente de excentricidad

Fuente	SS	DF	MS	F	p valor	np2
Excentricidad	0	2	0	10.451	3.06e-05	0.011
within	0.032	1897	0	—	—	—

- A partir del cuadro anterior se observa que existen diferencias entre el parámetro de las medianas de los grupos de factores ya que el p valor es menor que 0.05, por lo que se rechaza la hipótesis de que el grado no influye en el tiempo de ejecución. Esto se puede corroborar al observar la Figura 13 de la página 17. Como consecuencia de lo anterior se continua con el siguiente paso, aplicar la prueba de *Tukey* para mostrar las diferencias entre las medianas, lo que se evidencia en el cuadro 12 de la página 24.

4.6. Influencia de la característica de pagerank en el tiempo de ejecución

Cuadro 7: Coeficiente de *PageRank*

Fuente	SS	DF	MS	F	p valor	np2
PageRank	0.003	1	0.003	217.917	9.11e-47	0.103
within	0.029	1898	0	—	—	—

- Una vez observado el cuadro anterior se observa que no existen diferencias entre el parámetro de las medianas de los grupos de factores, ya que el p valor es menor que 0.05, por lo que se acepta la hipótesis de que el grado influye en el tiempo de ejecución.

5. Prueba estadística

En esta sección se presentan los resultados de la prueba estadística *Tukey* o Método de *Tukey* [3]. Este método se utiliza luego de aplicar la prueba de ANOVA y está basada en una prueba de rango. Una prueba de ANOVA arroja si los resultados son significativos en general, pero no aclara dónde se encuentran esas diferencias. Luego se emplea el método conocido como *Tukey* para averiguar qué medios de grupos específicos son diferentes, o sea para crear intervalos de confianza para todas las diferencias en parejas entre las medias de los niveles de los factores y además controla la tasa de error por familia en un nivel especificado [3].

En esta prueba se construyen intervalos de confianza para todas las posibles comparaciones por parejas que sigue la forma:

$$(\bar{y}_1 - \mu_1), (\bar{y}_2 - \mu_2), ...(\bar{y}_I - \mu_I) \tag{5}$$

Este método de*Tukey* resuelve el contraste:

$$H_0 : \mu_i = \mu_j \quad \text{vs } H_1 : \mu_i \neq \mu_j \tag{6}$$

$$H_0 : \quad \text{Hipótesis a demostrar} \tag{7}$$

$$H_1 : \quad \text{Hipótesis alternativa} \tag{8}$$

Este contraste es la hipótesis que se demostrará si es verdadera o no.
A continuación se presentan los resultados del método de *Tukey*.

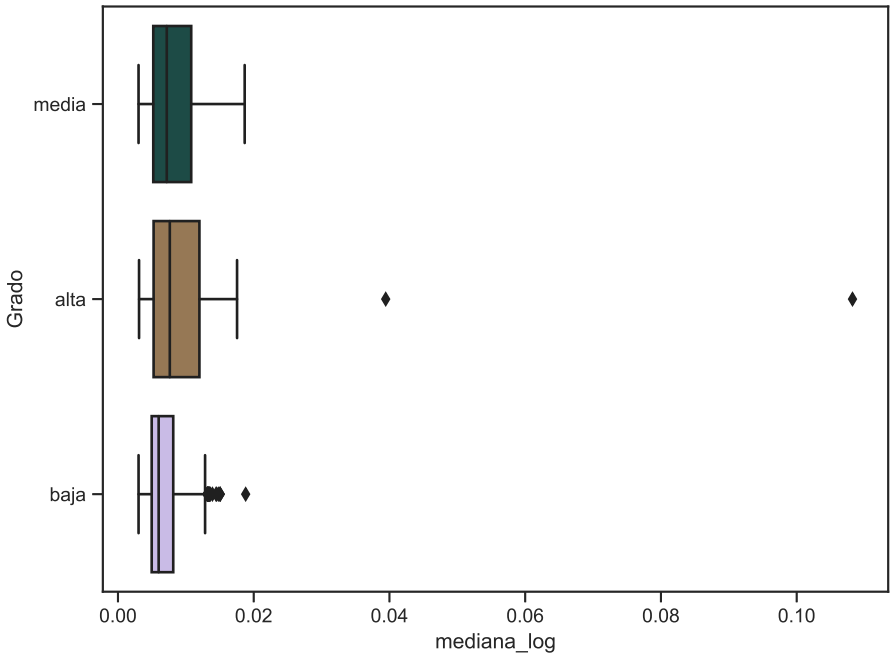


Figura 9: Diagrama de caja del parámetro Grado

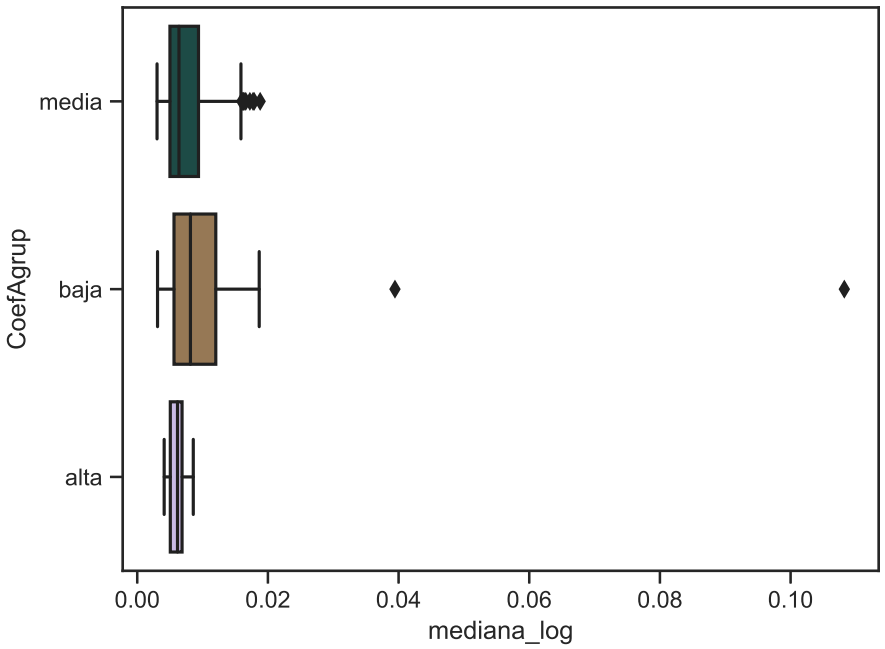


Figura 10: Diagrama de caja de coeficiente de agrupamiento y el tiempo

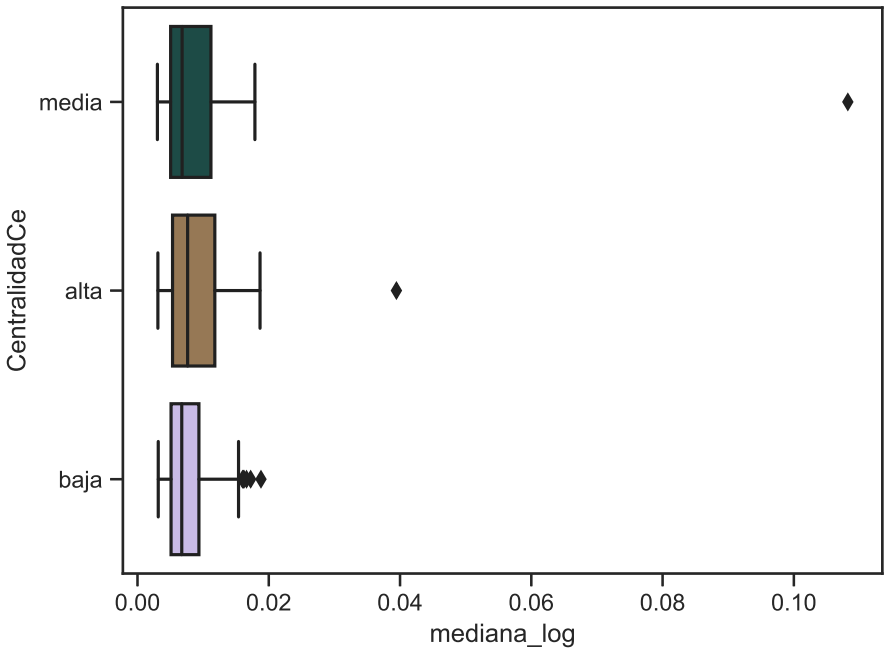


Figura 11: Diagrama de caja del coeficiente de centralidad de cercanía y el tiempo

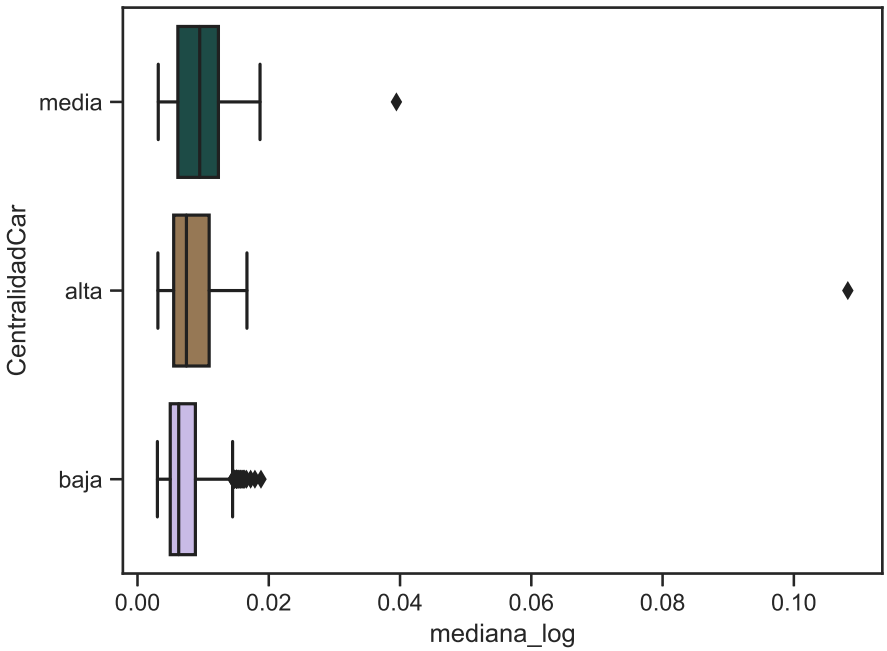


Figura 12: Diagrama de caja del coeficiente de centralidad de carga y el tiempo

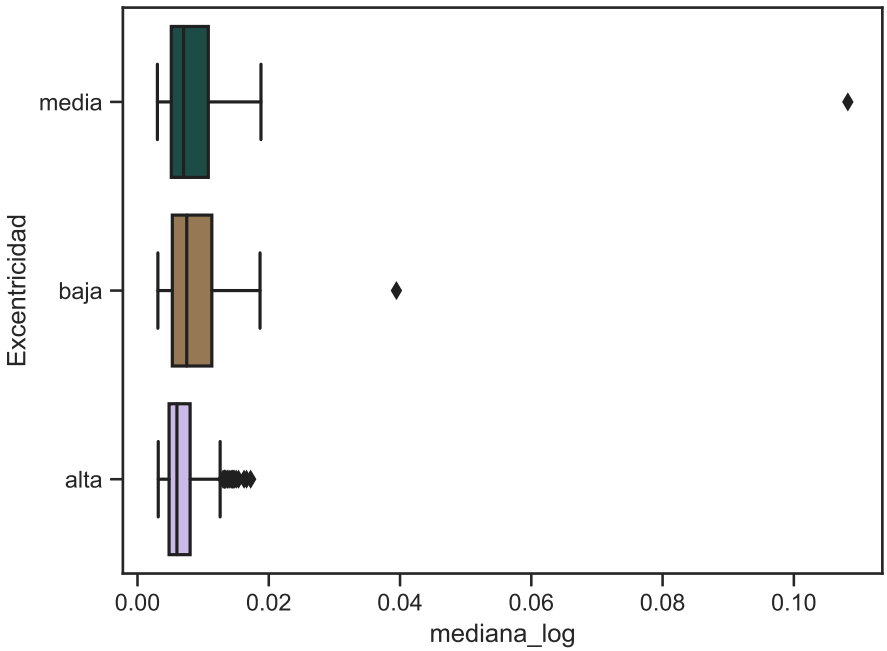


Figura 13: Diagrama de caja del coeficiente de excentricidad y el tiempo

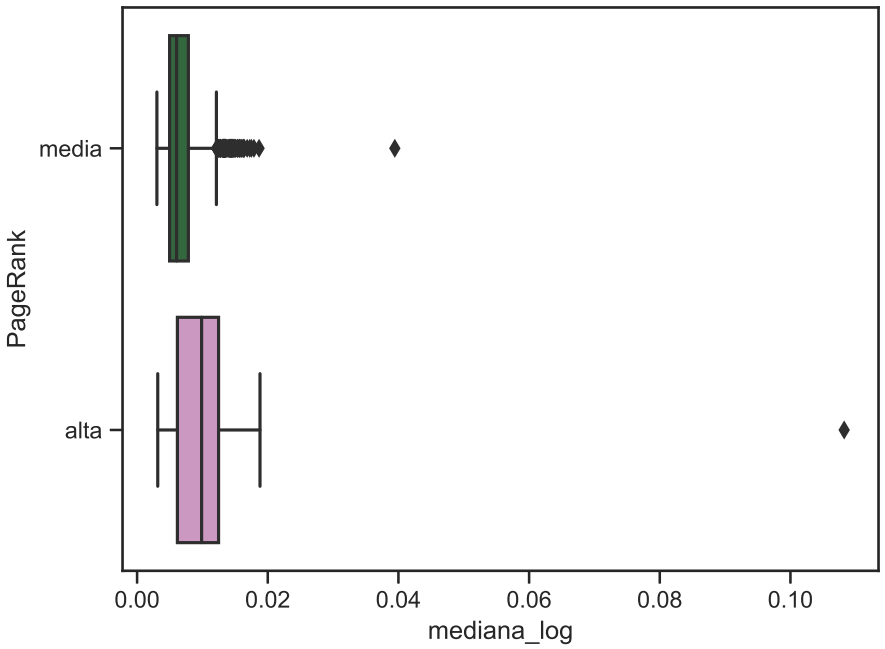


Figura 14: Diagrama de caja del coeficiente *PageRank* y el tiempo

A continuación las gráficas de bigote del método *Tukey*.

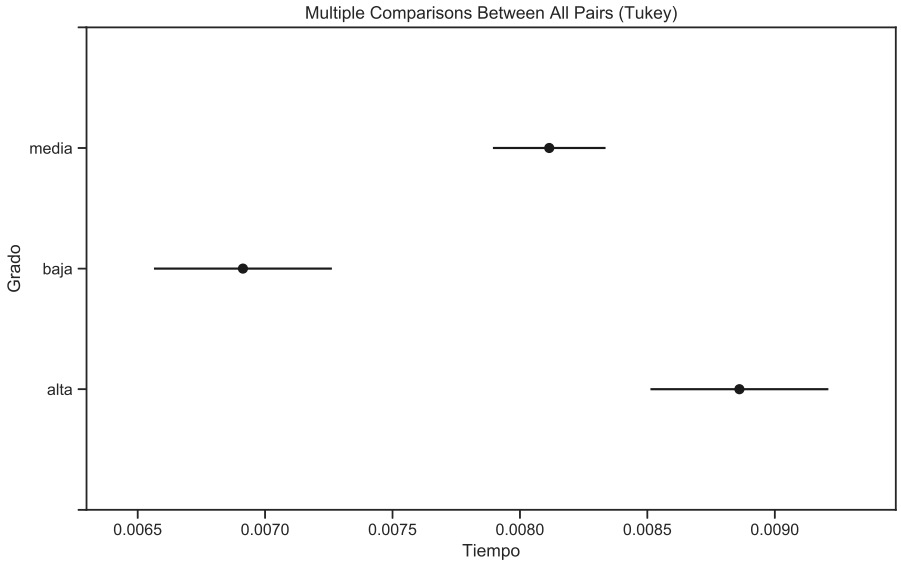


Figura 15: Gráfica del coeficiente grado y el tiempo

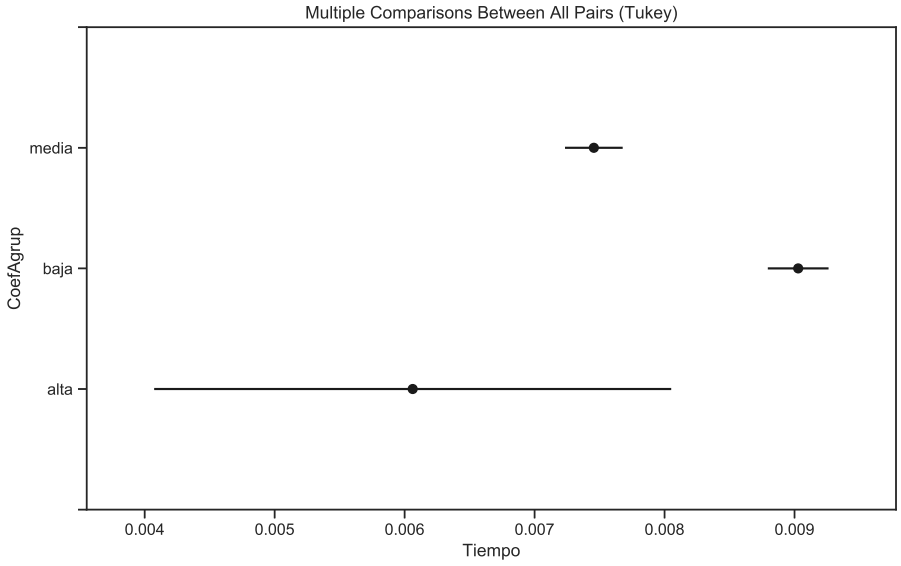


Figura 16: Gráfica del coeficiente de agrupamiento y el tiempo

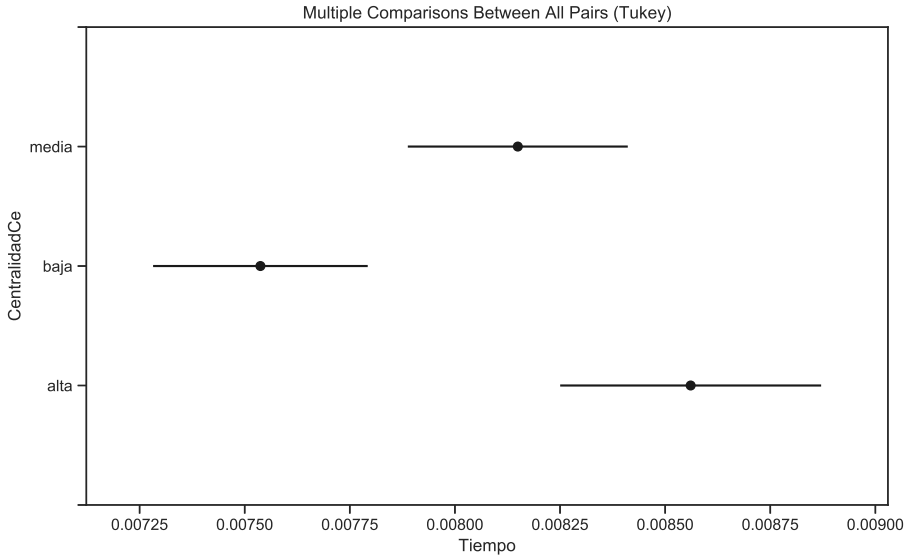


Figura 17: Gráfica del coeficiente de centralidad de cercanía y el tiempo

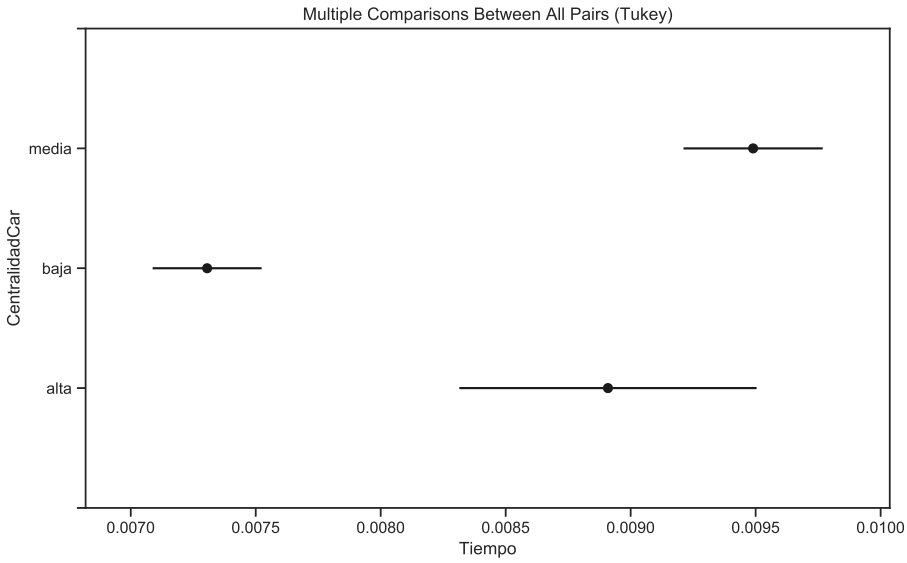


Figura 18: Gráfica del coeficiente de centralidad de carga y el tiempo

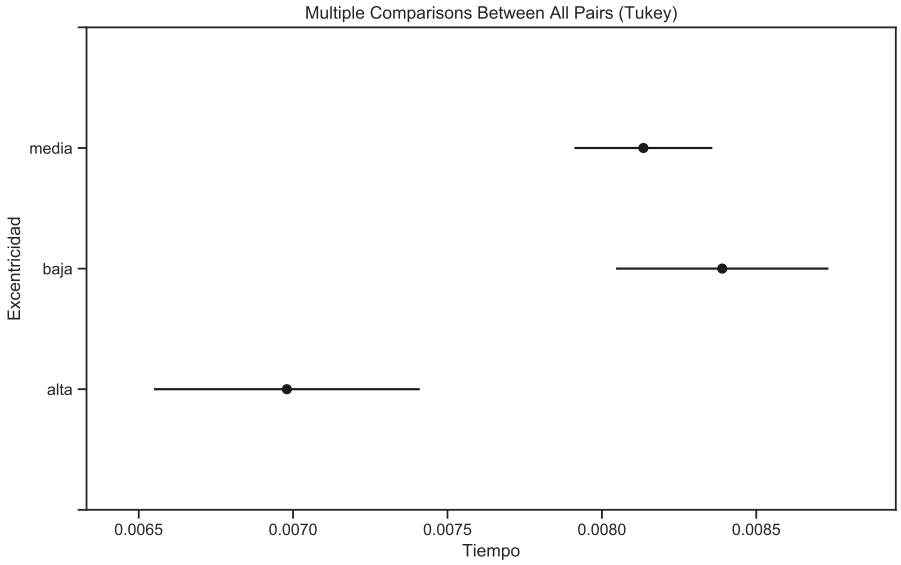


Figura 19: Gráfica del coeficiente de excentricidad y el tiempo

A continuación le sigue el código para aplicar la prueba de ANOVA y luego la prueba estadística de *Tukey*. La variable *tukey* almacena los resultados de la prueba estadística.

5.1. Código

```
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import researchpy as rp
import statsmodels.api as sm
from statsmodels.formula.api import ols
import numpy as np
import pingouin as pg
import seaborn as sns
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import csv
df = pd.read_csv("timesCopiaKOk.csv", index_col=None, usecols
    =[4,7,8,9,10,11,12],dtype={ 'Mediana': np.float64, 'Grado': np.int,
    'CoefAgrup':np.float64, 'CentralidadCe': np.float64,'CentralidadCar'
    : np.float64,
                                                                    '
    Excentricidad': np.int,'PageRank': np.float64} )

for column in range(0, df["PageRank"].count()):
```

```
    pass
    if  df.iat[column, 6] >=0.0399806  and df.iat[column, 6] < 0.04669785
     :
        df.iat[column, 6] = 1
    elif df.iat[column, 6] >=0.04669785  and df.iat[column, 6] <
    0.0534151:
        df.iat[column, 6] = 2
    else:
        df.iat[column, 6] = 3
df['PageRank'].replace({1:"baja", 2: 'media', 3:'alta' }, inplace= True)
for column in range(0, df["CoefAgrup"].count()):
    pass
    if  df.iat[column, 2] >=0 and df.iat[column, 2] < 0.33333333:
        df.iat[column, 2] = 1
    elif df.iat[column, 2] >=0.33333333 and df.iat[column, 2] <
    0.66666667:
        df.iat[column, 2] = 2
    else:
        df.iat[column, 2] = 3
df['CoefAgrup'].replace({1:"baja", 2: 'media', 3:'alta' }, inplace= True)
for column in range(0, df["CentralidadCe"].count()):
    pass
    if  df.iat[column, 3] >=0.33333333 and df.iat[column, 3] <
    0.39814815:
        df.iat[column, 3] = 1
    elif df.iat[column, 3] >=0.39814815 and df.iat[column, 3] <
    0.46296296:
        df.iat[column, 3] = 2
    else:
        df.iat[column, 3] = 3
df['CentralidadCe'].replace({1:"baja", 2: 'media', 3:'alta' }, inplace=
    True)
for column in range(0, df["CentralidadCar"].count()):
    pass
    if  df.iat[column, 4] >=0 and df.iat[column, 4] < 0.08828785:
        df.iat[column, 4] = 1
    elif df.iat[column, 4] >=0.08828785 and df.iat[column, 4] <
    0.1765757:
        df.iat[column, 4] = 2
    else:
        df.iat[column, 4] = 3
df['CentralidadCar'].replace({1:"baja", 2: 'media', 3:'alta' }, inplace=
    True)
for column in range(0, df["Grado"].count()):
    pass
    if  df.iat[column, 1] >=2 and df.iat[column, 1] < 3.33333333:
        df.iat[column, 1] = 1
    elif df.iat[column, 1] >=3.33333333 and df.iat[column, 1] <
    4.66666667:
        df.iat[column, 1] = 2
```

```
    else:
        df.iat[column, 1] = 3
df['Grado'].replace({1:"baja", 2: 'media', 3:'alta' }, inplace= True)
for column in range(0, df["Excentricidad"].count()):
    pass
    if  df.iat[column, 5] >=3 and df.iat[column, 5] < 3.66666667:
        df.iat[column, 5] = 1
    elif df.iat[column, 5] >=3.66666667 and df.iat[column,5] <
    4.66666667:
        df.iat[column, 5] = 2
    else:
        df.iat[column, 5] = 3
df['Excentricidad'].replace({1:"baja", 2: 'media', 3:'alta' }, inplace=
    True)
logX = np.log1p(df['Mediana'])
df = df.assign(mediana_log=logX.values)
df.drop(['Mediana'], axis= 1, inplace= True)
factores=["Grado","CoefAgrup","CentralidadCe","CentralidadCar","
    Excentricidad","PageRank"]
plt.figure(figsize=(8, 6))
for i in factores:
    anova = pg.anova (dv='mediana_log', between=i, data=df, detailed=True
     , )
    pg._export_table (anova,("ANOVAs"+i+".csv"))
    ax=sns.boxplot(x=df["mediana_log"], y=df[i], data=df, palette="
    cubehelix")
    plt.savefig("boxplot_" + i + ".eps", bbox_inches='tight')
    tukey = pairwise_tukeyhsd(endog = df["mediana_log"], groups= df[i],
    alpha=0.05)
    tukey.plot_simultaneous(xlabel='Tiempo', ylabel=i)
    plt.vlines(x=49.57,ymin=-0.5,ymax=4.5, color="red")
    plt.savefig("simultaneous_tukey" + i + ".eps", bbox_inches='tight')
    t_csv = open("Tukey"+i+".csv", 'w')
```

6. Cuadros de la prueba de Tukey

Cuadro 8: Coeficiente de grado

group1	group2	meandiff	lower	upper	reject
alta	baja	-0.0019	-0.0026	-0.0013	True
alta	media	-0.0007	-0.0013	-0.0002	True
baja	media	0.0012	0.0006	0.0018	True

Cuadro 9: Coeficiente de agrupamiento

group1	group2	meandiff	lower	upper	reject
alta	baja	0.003	0.0007	0.0052	True
alta	media	0.0014	-0.0008	0.0036	False
baja	media	-0.0016	-0.002	-0.0011	True

Cuadro 10: Cuadro del coeficiente de centralidad de cercanía

group1	group2	meandiff	lower	upper	reject
alta	baja	-0.001	-0.0016	-0.0005	True
alta	media	-0.0004	-0.001	0.0002	False
baja	media	0.0006	0.0001	0.0011	True

Cuadro 11: Cuadro del coeficiente de centralidad de carga

group1	group2	meandiff	lower	upper	reject
alta	baja	-0.0016	-0.0024	-0.0008	True
alta	media	0.0006	-0.0003	0.0015	False
baja	media	0.0022	0.0017	0.0027	True

Cuadro 12: Coeficiente de excentricidad

group1	group2	meandiff	lower	upper	reject
alta	baja	0.0014	0.0006	0.0022	True
alta	media	0.0012	0.0005	0.0018	True
baja	media	-0.0003	-0.0008	0.0003	False

7. Conclusión

Se puede concluir luego del análisis del comportamiento de los datos reflejados en los cuadros y en las gráficas del método estadístico *Tukey* que de los factores que son: el grado, el coeficiente de agrupamiento, el coeficiente de Centralidad de cercanía, el coeficiente de centralidad de carga, el coeficiente de excentricidad y el coeficiente PageRank, los factores que más influyen en la variable dependiente (el tiempo de ejecución) son factor de el coeficiente de centralidad de carga y el coeficiente *PageRank*.

Referencias

[1] Corey Bryant. `https://pypi.org/project/researchpy/`.

[2] Jorge Fallas. *Análisis de varianza*. 2012.

[3] Lara Porras. `http://wpd.ugr.es/~bioestad/wp-content/uploads/ComparacionesMultiples.pdf/`.

[4] Matplotlib developers. `https://matplotlib.org`.

[5] Michael Waskom. `https://seaborn.pydata.org/`.

[6] Networkx developers con última actualización el 19 de Septiembre 2018. `https://networkx.github.io/documentation/stable`.

[7] Numpy developers. `https://www.numpy.org/`.

[8] Olivier Pomel. `https://pandas.pydata.org/`.

[9] Python Software Foundation Versión. `https://www.python.org`.

[10] Raphael Vallat. `https://pypi.org/project/pingouin/`.

[11] Travis Oliphant. `https://www.scipy.org/`.