# Diagrama de Caja con Python

Guatemala, 03 de junio de 2022

Pablo Sao Alonzo[1]
1. Departamento de diseño y desarrollo de software, Solution Design of Centroamerica, Guatemala.

La estadística descriptiva nos ayuda a visualizar de forma rápida las características de la distribución en un grupo de datos (*dataset*), por medio de los valores de importancia de una forma simplificada, logrando tener un mejor entendimiento de la información colectada (Potter, 2016).

Por esta razón el diagrama de caja es una de las visualizaciones de datos utilizadas en la estadística descriptiva, permitiendo realizar análisis sin extraer conclusiones, por medio del resumen de los cinco valores que representa este tipo de diagrama (Faraldo y Pateiro, 2012; Anderson *et al.*, 2016).

Anderson *et al.* (2016) hacen mención de que estos cinco valores son:

1. Valor menor del conjunto de datos
2. Primer Cuartil ($Q_1$)
3. Mediana ($Q_2$)
4. Tercer Cuartil ($Q_3$)
5. Valor mayor del conjunto de datos

Para entender un diagrama de caja y su elaboración con Matplotlib y Plotly utilizando Python, estaremos utilizando como datos de prueba la edad de las zarigüeyas extraído de Kaggle desde el siguiente link: ***https://www.kaggle.com/abrambeyer/openintro-possum***, también el archivo de datos CSV se encuentra disponible en el repositorio de GitHub: ***https://github.com/sdesignca/blog-ps-diagrama-caja-python***

## Entendiendo un Diagrama de Caja

Antes de elaborar el diagrama de caja con Matplotlib y Plotly, definiremos de forma breve los conceptos básicos y los cálculos de los valores utilizados para esbozar un diagrama de caja, de una forma práctica. Por lo que estaremos utilizando los siguientes datos que corresponden a la edad de las zarigüeyas hembra:

6, 6, 2, 1, 6, 9, 6, 9, 5, 1, 5, 4, 3, 4, 2, 3, 2, 4, 2, 1, 3, 6,
5, 2, 5, 5, 3, 4, 2, 3, 6, 3, 5, 3, 7, 4, 4, 3, 3, 2, 3, 6, 3

Antes de iniciar a realizar los cálculos de los cinco valores, debemos ordenar los datos de forma ascendente, por lo que los datos nos quedan de la siguiente forma:

1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4,
4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 7, 9, 9

Ahora con los datos ordenados, podemos identificar que el valor mínimo de nuestro set de datos es 1 y el valor máximo es el 9. Por lo que calcularemos ahora la ***mediana*** ($Q_2$). La cual es la posición central de nuestro conjunto de datos (*dataset*).

Si la cantidad de datos es par, la mediana corresponde al promedio de los dos valores centrales, por lo que primero se debe calcular la posición del valor central de la siguiente forma:

$$POSICIÓN_{MEDIANA} = \frac{n}{2}$$

*Donde n es el número de datos con los que contamos.*

Luego de tener la posición, y al tener una cantidad de datos par, se debe calcular la mediana de la siguiente forma:

$$MEDIANA = \frac{POSICIÓN_{MEDIANA} + (POSICIÓN_{MEDIANA} + 1)}{2}$$

Pero si llegáramos a tener un conjunto de datos impar, al realizar el cálculo de la posición deberemos redondear el resultado al entero próximo. En nuestro caso contamos con 43 datos, siendo *n* impar, por que deberemos redondear el resultado que obtendremos, para identificar la posición central de nuestro conjunto de datos:

$$POSICIÓN_{MEDIANA} = \frac{43}{2} = 21.5 = 22$$

Ahora que tenemos la ubicación, debemos identificar el valor que se encuentra en la posición 22, en este caso el valor corresponde al valor 4:

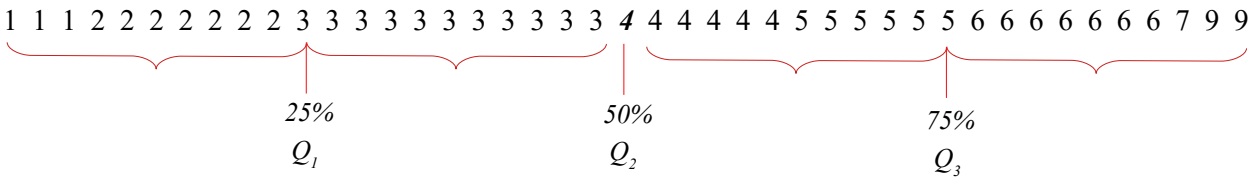
1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ***4***,
4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 7, 9, 9

Por lo tanto:

$$Q_2 = 4$$

Teniendo la media identificada, debemos calcular el primer y tercer cuartil. Los ***cuartiles*** son percentiles específicos, siendo recomendable dividir los datos en cuatro partes iguales. La información que nos proporcionan, es la observación de la distribución de los datos menores o iguales al valor del percentil especifico. Donde es utilizado para el primer cuartil ($Q_1$) del 25% y para el tercer cuartil ($Q_3$) del 75%, en los diagramas de caja.

Para poder ejemplificar dicha distribución, mostraremos la división de los datos. Dichos valores los obtendremos con el calculo del cuartil $Q_1$ y $Q_2$:




Ahora procedemos a calcular el cuartil $Q_1$, con la siguiente formula par el calculo del percentil:

$$i=(\frac{p}{100})*n$$

*Donde n es el número de datos con los que contamos*

Dado que nuestra *n* es de 43, y el percentil *p* que deseamos calcular es del 25%, por lo tanto:

$$i=(\frac{25}{100})*43=10.75=11$$

Dado que *i* no es un entero, y se tuvo que aproximar, el cuartil $Q_1$ corresponde al valor de la posición 11 de nuestros datos, por lo tanto.

$$Q_1=3$$

Ahora debemos calcular el cuartil $Q_3$ que corresponde a un percentil *p* del 75%.

$$i=(\frac{75}{100})*43=32.25=33$$

En esta ocasión la *i* no es un entero y se redondeó al entero superior, por lo que percentil $Q_3$ corresponde al valor de la posición 33, por lo tanto.

$$Q_3=5$$

Ahora nos queda calcular el ***rango intercuartílico*** o ***intercuartil*** (RIC), el cual es la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1).

$$RIC = Q_3 - Q_1$$

Además del cálculo del RIC, debemos calcular los limites de nuestro diagrama de caja, por lo que utilizamos la siguiente ecuación para el cálculo del límite inferior:

$$Límite\ Inferior = Q_1 - (1.5 * RIC)$$

Para calcular el límite superior utilizamos la siguiente ecuación:

$$Límite\ Superior = Q_3 + (1.5 * RIC)$$

Por lo que procedemos a iniciar con el calculo de nuestro rango intercuartílico:

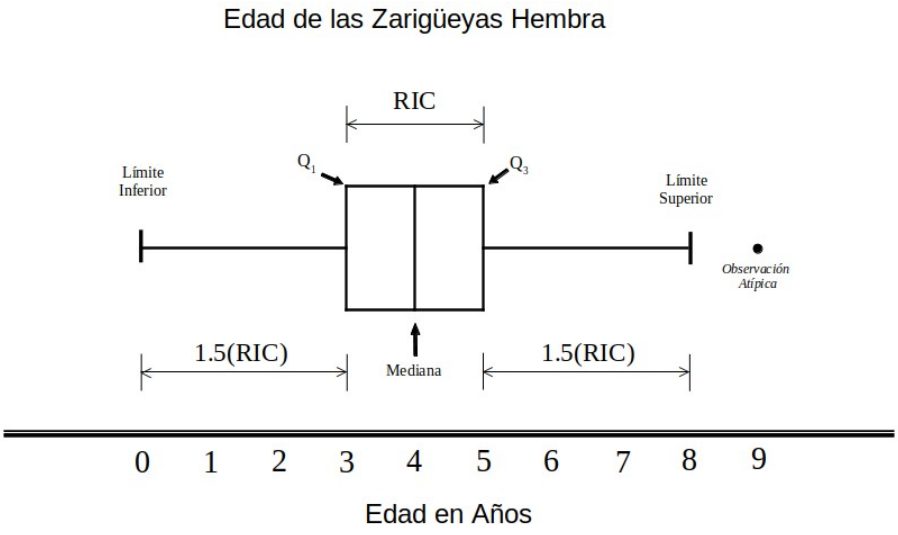
$$RIC = Q_3 - Q_1 = 5 - 3 = 2$$

Obteniendo nuestro RIC = 2, procedemos a calcular el límite inferior y superior:

$$Límite\ Inferior = 3 - (1.5 * 2) = 0$$

$$Límite\ Superior = 5 + (1.5 * RIC) = 8$$

Con los datos calculados, podemos obtener la siguiente gráfica de caja.



Es importante mencionar que los datos que se encuentren fuera de los límites calculados, serán observaciones atípicas en nuestros datos.

# Gráfica de Caja con Python

Para poder iniciar la explicación, supondremos que el lector está familiarizado con la sintaxis de Python y con Jupyter Notebook, además de tener conocimiento sobre la carga de archivos CSV en Python con Pandas[1], donde los archivos de datos que estaremos utilizando; así como, el archivo de Jupyter con el código, puede ser descargados desde el siguiente repositorio en GitHub: https://github.com/sdesignca/blog-ps-diagrama-caja-python

Por lo que estaremos graficando las edades de las zarigüeyas por sexo (macho y hembra), cargando los datos de la columna "***sex***" (sexo) y "***age***" (edad) del archivo "***possum.csv***". Iniciando por la importación de la librería Pandas con la siguiente instrucción:

```
import pandas as pd
```

Luego de importar la librería, asignaremos a nuestra variable "**datos**" el *DataFrame* con las columnas deseadas ("*sex*" y "*age*") de nuestro archivo "***possum.csv***", el cual se encuentra en el mismo directorio que nuestro archivo de *Jupyter Notebook*. Si en dado caso nuestro archivo se encuentra en otra ubicación, debemos colocar la ruta (*path*) de nuestro archivo de datos junto con el nombre del archivo CSV.

```
datos = pd.read_csv( "possum.csv"
                    ,delimiter=','
                    ,usecols=['sex','age']
                   )
```

A lo largo de nuestra explicación, no se manipularán los datos contenidos en el *DataFrame*, por lo cual esta misma variable será utilizada para realizar la gráfica; tanto con Matplolib, como con Plotly.

## Gráfica con Pandas y Matplotlib

*Matplotlib* es una librería utilizada para realizar gráficas estáticas, animadas y visualizaciones interactivas a partir de datos contenidos en listas, vectores y en su extensión matemática, la cual es *NumPy (*Potter*, 2006)*.

1 Si se desea conocer más sobre el método para cargar datos de archivos CSV con Python, pueden leer sobre el tema en el siguiente enlace: https://www.solutiondesign.tech/cargando-datos-de-un-csv-en-python-con-pandas/

La librería de Pandas trabaja en conjunto con *NumPy* y *Matplotlib*, por esta razón es posible utilizar una función propia del objeto creado con Pandas, agilizando el proceso de creación de gráficas, en la que si deseamos realizar cambios en la gráfica, debemos manipular el objeto de *Matplotlib.*

Para poder realizar la gráfica de caja con *Matplotlib*, necesitamos tener instalado el paquete, si no lo hemos instalado, podemos hacerlo por medio del comando *pip* o *pip3*. En nuestro caso haremos uso de pip3, ejecutando el siguiente comando desde la linea de comandos o terminal, para su instalación:

```
pip3 install matplotlib
```

Teniendo instalada la librería de *Matplotlib*, importamos el paquete en nuestro archivo de *Jupyter Notebook* de la siguiente forma, colocándole como alias ***plt***:

```
import matplotlib.pyplot as plt
```

Haciendo uso de la variable **datos**, luego de cargar la información del archivo CSV. Utilizaremos el siguiente código para realizar la gráfica de caja, el cual deberemos ejecutar de forma conjunta en Jupyter Notebook,  donde luego explicaremos linea por linea su funcionamiento.

```
ax = datos.boxplot(by='sex',grid=False,rot=45)
ax.set_title('Edad de Zarigüeyas por Sexo');
plt.suptitle('')
ax.set_xlabel("Sexo");
ax.set_ylabel("Edad (años)");
plt.xticks([1,2], ['Hembra', 'Macho'])
plt.show()
```

Si ejecutamos de forma individual cada instrucción en Jupyter Notebook, primero nos mostrará el diagrama de caja con sus parámetros por defecto (imagen 1), luego nos mostrara por separado el cambio a los ejes que realizamos (imagen 2).

**Imagen 1:** Gráfica de Matplolib sin modificaciones.

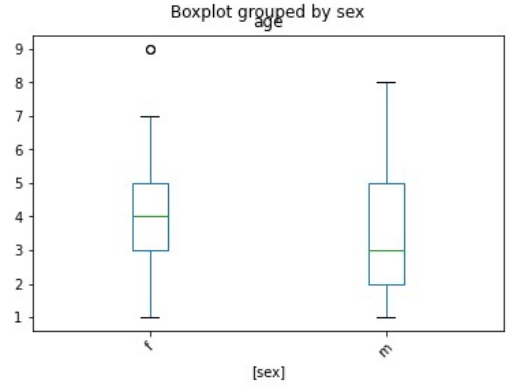
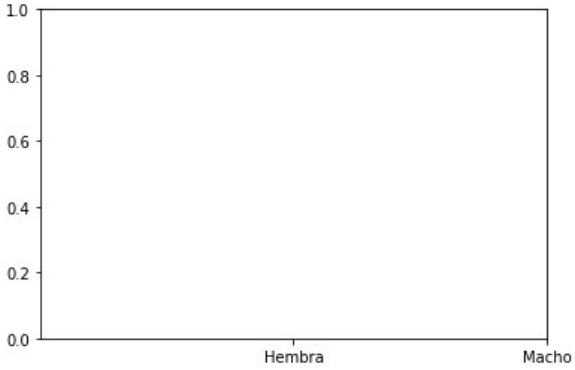
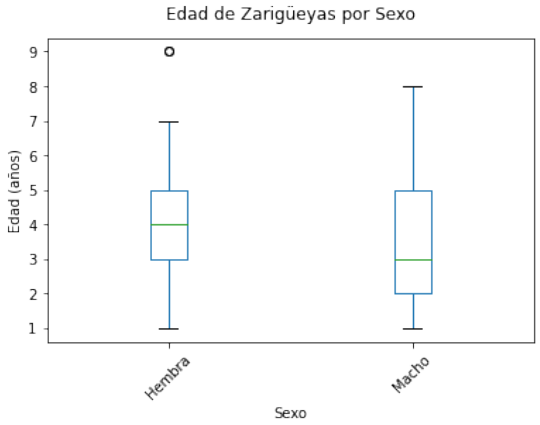



**Imagen 2:** Gráfica de Matplotlib solo con ejes modificados.




Al ejecutar todo el fragmento de código en conjunto, obtenemos la gráfica con las modificaciones que realizamos:



Si revisamos el código con su funcionamiento, en la siguiente linea, utilizamos el método de ***boxplot*** de nuestro *DataFrame*. Donde nuestra segmentación del sexo de las zarigüeyas se encuentra en la columna sexo (*sex*); donde ***f*** es para las hembras y ***m*** es para los machos; por esta razón debemos agrupar los datos por los valores de esta columna. Esto lo hacemos con el parámetro ***by***, donde nos agrupará la información del resto de columnas (en nuestro caso solo tenemos la columna *age*). El parámetro ***grid*** con el valor ***False***, nos quita el grid del fondo de la gráfica, si quisiéramos que el grid apareciera podemos colocar ***True*** o eliminarlo, ya que el valor por defecto es ***True***. El parámetro ***rot***, se refiere a la rotación de la etiqueta de los valores (en nuestro caso ***f*** y ***m***).

```
ax = datos.boxplot(by='sex',grid=False,rot=45)
```

Con la siguiente instrucción le asignamos el título de nuestra gráfica.

```
ax.set_title('Edad de Zarigüeyas por Sexo');
```

Con la siguiente instrucción colocamos el subtitulo de nuestra gráfica como vacía. Si no colocamos el subtitulo como vacío, nos aparecerá el valor generado por defecto, que en nuestro caso es la agrupación por sexo de las zarigüeyas.

```
plt.suptitle('')
```

Ahora colocamos el identificador de nuestro eje x con el valor "*Sexo*":

```
ax.set_xlabel("Sexo");
```

Con la siguiente instrucción colocamos el identificador de nuestro eje Y con el valor "*Edad (años)*":

```
ax.set_ylabel("Edad (años)");
```

Es importante tomar en consideración el siguiente punto, ya que vamos a cambiar con la siguiente instrucción la etiqueta de identificación a nuestras gráficas, por lo que debemos de saber a que valores corresponde el identificador. En nuestro caso, el valor 1 corresponde a la gráfica con los datos de las edades de las zarigüeyas hembra, y el valor 2 corresponde a los datos de las edades de las zarigüeyas macho, por lo que ponemos en el primer parámetro (*ticks*) los identificadores de los datos, y en el

segundo parámetro (*labels*) le asignamos el nombre que tendrán los identificadores de los componentes de la gráfica.

```
plt.xticks([1,2], ['Hembra', 'Macho'])
```

Luego de las transformaciones realizadas a nuestra gráfica, la mostramos con la siguiente instrucción:

```
plt.show()
```

## Gráfica con Plotly

Plotly es un framework open-source de Python, utilizado para poder crear aplicaciones web de gráficas interactivas, donde no es necesario el ser un experto en HTML, CSS y Javascript para crear *dashboards* de gráficas.

Para poder realizar la gráfica de caja con *Plotly*, necesitamos tener instalado el paquete, si no lo hemos instalado, podemos hacerlo por medio del comando *pip* o *pip3*. En nuestro caso haremos uso de *pip3*, ejecutando el siguiente comando para su instalación:

```
pip3 install plotly
```

Luego debemos de importar la librería de plotly con el alias px.

```
import plotly.express as px
```

Teniendo importada la librería, utilizaremos el método box de plotly para realizar la gráfica de caja, donde el primer parámetro del método corresponde a la variable que contiene el *DataFrame* con los datos que deseamos graficar; en nuestro caso dicha variable es **datos**. El siguiente parámetro que utilizaremos sera **x**, el cual nos sirve para indicar que datos se encontraran en el **eje X** de nuestra gráfica, que en nuestro caso es "sex". El parámetro **y** nos sirve para indicar que datos del *DataFrame* estarán en el **eje Y**.

Con el parámetro **points** podemos indicar que información deseamos mostrar en nuestra gráfica, con la opción "**outliers**" (que es el parámetro por defecto) indicamos que nos muestre datos atípicos, con la opción "**none**" indicamos que no nos muestre los valores atípicos, ni la dispersión de los datos en la gráfica de caja. Y por último, la opción "**all**" nos muestra la dispersión de datos al lado del diagrama de caja y los valores atípicos.

El parámetro **title** nos sirve para indicar el nombre que tendrá nuestro diagrama, y el parámetro **labels** nos sirve para definirle el nombre en forma de diccionario a nuestros ejes.

```
fig = px.box( datos
            ,x="sex"
            ,y="age"
            ,points="outliers"
            ,title="Edad de Zarigüeyas por Sexo"
            ,labels={
                        "sex": "Sexo"
                        ,"age": "Edad (años)"
            }
      )
```

Por último, mostramos nuestra gráfica con el siguiente comando:

```
fig.show()
```

Es importante mencionar que en *Jupyter Notebook*, nuestra gráfica se mostrara sin necesidad de utilizar este último comando.

## Referencias

Anderson, D., Sweeney, D., Williams, T., Camm, J. y Cochran, J. (2016). *Estadística para negocios y economía*. 12va edición. Ciudad de México, México. CENGAGE Learning. 104, 108 – 110, 131 – 132 pp.

Faraldo, P. y Pateiro, B: (2012). *Tema 1. Estadística Descriptiva*. Extraído de: http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP-DPTO/MATERIALES/Mat_G2021103104_EstadisticaTema1.pdf

Kathari, S. (2020). Plotly Dash: A beginner's guide to building an analytics dashboard. Extraído de: https://medium.com/analytics-vidhya/plotly-dash-a-beginners-guide-to-building-an-analytics-dashboard-cedf297e01f1

Potter, K. (2006). *Methods for Presenting Statistical Information: The Box Plot*. Extraído de: http://www.sci.utah.edu/~kpotter/publications/potter-2006-MPSI.pdf