Diagrama de Barras con Python

Guatemala, 20 de mayo de 2022

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

La gráfica de barras es una representación que utilizamos para mostrar datos categóricos[1] de una forma resumida y con una distribución de frecuencia, frecuencia relativa o de frecuencia porcentual, donde un eje de la gráfica se muestran las etiquetas de las clases o categorías; mientras que en el otro eje de la gráfica se coloca la escala de recurrencia (Anderson *et al.*, 2016; Newbold *et al.*, 2008).

Este tipo de gráfica las podemos emplear para llamar la atención sobre la frecuencia de una categoría, ya que nos ayuda a identificar las principales causas de los problemas o áreas de interés. Si el diagrama de barras se acomoda en orden descendente de altura, mostrando en la primera posición la causa que ocurre con mayor frecuencia, obtenemos el diagrama de Pareto, el cual nos ayuda a identificar estos problemas para corregirlos de una forma rápida, buscando el costo mínimo de la corrección (Anderson *et al.*, 2016; Newbold *et al.*, 2008).

Para elaborar el diagrama de barra con Matplotlib y Plotly utilizando Python, estaremos utilizando los datos del reporte de Excel sobre la pesca marina del 2020 de Canadá, que pueden ser descargado desde el siguiente enlace: ***https://open.canada.ca/data/en/dataset/288b6dc4-16dc-43cc-80a4-2a45b1f93383***. El reporte de Excel y el código elaborado se encontrará disponible en el repositorio de GitHub: ***https://github.com/sdesignca/blog-ps-diagrama-barra-python***

Donde supondremos que el lector está familiarzado con la sintaxis de Python y con el uso de Jupter Notebook; así mismo, como los conocimientos básicos de carga de datos a partir de archivos de Excel en Python utilizando Pandas[2].

Gráfica de Barras con Python

Antes de iniciar con la elaboración de las gráficas importaremos la librería de Pandas con el alias *pd*, de la siguiente forma:

```
import pandas as pd
```

Luego de importar la librería de Pandas, cargaremos los registros del archivo de Excel "*marine-finfish-data-2020.xlsx*" y asignaremos el *DataFrame* a la variable **datos**. Se ha de mencionar que este archivo

1 Las variables categóricas nos ayudan a mostrar respuestas a grupos o categorías, las cuales pueden ser respuestas como: sí/no, totalmente en desacuerdo hasta totalmente de acuerdo; entre otras respuestas (Newbold *et al.*, 2008).

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

de Excel es un reporte con formato predefinido, por lo cual deberemos realizar la manipulación de la información para cargar los datos que deseamos, por medio de la función **read_excel**, donde estaremos omitiendo la lectura específica de filas y columnas, para cargar unicamente los datos del Peróxido de Hidrógeno (*Hydrogen peroxide*) de la provincia de *New Brunswick*, Canadá.

Imagen 1: Reporte de Excel con información de la pesca marina del 2020 de las provincias de Canadá.

	A	B	C	D	E	F	G	H	I	J
1	*Aquaculture Activities Regulations* Reported Drugs and Pesticides Use (Reporting Year: 2020, Sector: Marine Finfish)									
2	Extracted from AQUIIS-AAR December 1, 2021; revised February 9, 2022. Active ingredient quantities in kilograms.									
3	Facility Reference #	Facility Name	Azamethiphos	Emamectin benzoate	Erythromycin	Florfenicol	Formalin	Hydrogen peroxide	Ivermectin	Oxytetracycline
4	British Columbia									
5	1698	Ahlstrom Point, Jervis Inlet		0.26						
6	1738	Atrevida Point, Hanna Channel				64.4		146817		
7	871	Barnes Bay, Sonora Island		1.166				37299		
8	227	Bawden Point, Herbert Inlet				57.5				
9	1401	Brent Island, Okisollo Channel		2.14						
10	1554	Charlie's Place, E. Pinnace Ch, Kyuquot Sound								1134.02
11	1789	Concepcion Pt., Bligh Island						165679.2		
12	7713	Cougar Bay, Tolmie Channel		0.87						
13	1697	Culloden Point, Jervis Inlet		0.494						
14	458	Cypress Hrbr, Harbour Pt, Sutlej Channel		0.047	1.952	6.696				98
15	234	Dixon Point, Shelter Inlet				45.56				299.6
16	1586	Doctor Islets, Knight Inlet				28.9089				
17	1288	Doyle Island, Gordon Group		0.477		198.5894				
18	1293	Duncan Island, Goletas Channel		0.591		332.88025				
19	1863	Esperanza, Hecate Channel		0.53		252.4		151909.8		
20	540	Fortune Channel, East side Warn Bay		0.537		222.738				
21	7053	Ghi ya, Bull Harbour, Hope Isl		0.5736		122.60145		25320		
22	303	Glacial Creek, near Jervis Inlet		0.0702						
23	1762	Gore Island, King Passage						239958		
24	1581	Hardwicke Is. Site B, Chancellor Channel		1.17						
25	1862	Hecate, Hecate Channel		0.32		241.345		59795.4		
26	1618	Humphrey Rock, Tribune Channel		0.929						
27	1691	Kid Bay, Roderick Island				51.9528				
28	144	Koskimo Bay, Quatsino Sound				56.899				

En la función **read_excel**, necesitamos especificar las filas que deseamos omitir en la carga de datos por medio del parámetro **skiprows**. Como este es un reporte que al descargar desde el sitio obtendremos el mismo formato, utilizaremos el siguiente fragmento de código para agregar a una lista o arreglo los números de filas que omitiremos para cargar únicamente la información de *New Brunswick*:

```
# omitimos el título del  reporte y la información
# de donde se obtuvieron los datos
SKIPROWS = [0,1]

# omitimos la información luego del encabezado hasta New Brunswick
for row in range(3,68):
    SKIPROWS.append(row)


# omitimos la información luego de los datos luego New Brunswick
hasta el final
```

```
for row in range(96,109):
    SKIPROWS.append(row)
```

Teniendo las filas que desamos omitir en la variable **SKIPROWS**, utilizaremos el método read_excel para cargar la información de *New Brunswick*, en el cual cargaremos únicamente las columnas **B** (*Facility Name*) y **H** (*Hydrogen peroxide*). Si existe alguna duda sobre la carga de información desde un archivo de Excel, pueden leer el árticulo "**Cargando Datos de un Excel en Python con Pandas**" en nuestro blog.

```
datos = pd.read_excel("marine-finfish-data-2020.xlsx"
                      ,sheet_name="Marine Finfish 2020 Data"
                      ,skiprows= SKIPROWS
                      ,usecols="B,H")
```

Al ver la información cargados en la variable **datos** con la función ***head()***, podemos observar que para la provincia de *New Brunswick* no hay valores para unas instalaciones.

```
datos.head()
```

Imagen 2: Primeros cinco registros del *DataFrame* contenido en la variable datos.

	Facility Name	Hydrogen peroxide
0	NaN	NaN
1	Adventure	NaN
2	Bancroft Point	NaN
3	Bar Island	29905.0
4	Benson	NaN

Por lo que eliminaremos los registros nulos (NaN) de la información del *DataFrame*, con la función ***dropna***, en la que enviaremos el valor ***True*** en el parámetro ***inplace***, el cual nos servirá para indicar que la eliminación de los datos nulos se aplique al dataframe de la variable, conservando únicamente las instalaciones con registros.

```
datos.dropna(inplace=True)
```

Para poder elaborar la gráfica de Pareto, vamos a ordenar nuestra información por los datos de la columna *Hydrogen peroxide* utilizando el parámetro **by** de la función **sort_value**, donde indicaremos que se ordene de forma descendente, pasando en el parámetro **ascending** el valor **False** y por último indicaremos que los cambios sean aplicados a nuestra variable **datos** por medio del parámetro **inplace** con el valor **True**.

```
datos.sort_values(by='Hydrogen peroxide'
                  ,ascending=False
                  ,inplace=True)
```

Aplicando este ordenamiento, tenemos listos nuestros datos para ser graficados, por esta razón a lo largo de nuestra explicación, ya no se volverán a manipular los datos contenidos en el *DataFrame* de nuestra variable **datos**, por lo cual dicha variable será utilizada para realizar la gráfica con Matplotlib y Plotly.

Gráfica con Pandas y Matplotlib

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

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

```
pip3 install matplotlib
```

Teniendo instalada la librería de *Matplotlib*, importamos el paquete en nuestro archivo de *Jupyter Notebook* con el alias ***plt***:

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

Haciendo uso de la variable **datos**, utilizaremos el siguiente código para realizar la gráfica de barras, el cual deberemos ejecutar de forma conjunta en *Jupyter Notebook,* o de lo contrario la manipulación posterior a la creación de la gráfica será mostrada por separado en nuestro *Jupyter Notebook*:

```
ax = datos.plot(x="Facility Name", y="Hydrogen peroxide", kind="bar",
grid=False)
```

```
ax.set_title('Presencia de Peróxido de Hidrógeno en New Brunswick,
Canadá.')
ax.set_xlabel("Instalaciones")
ax.set_ylabel("Peróxido de Hidrógeno")
plt.show()
```

Revisando el funcionamiento del código, podemos ver en la primera linea que le asignamos a la variable ax la gráfica de Matplolib, en la que el parametro "**x**" corresponde al eje *X* de la gráfica, donde estaremos utilizando como dato la columna que contiene el nombre de la instalación ("*Facility Name*"). El parámetro "**y**" es utilizado para indicar los datos del eje *Y*, donde estaremos utilizando la información de la columna que contiene los valores del peróxido de hidrógeno ("Hydrogen peroxide"), en el parámetro **kind**, indicaremos que será una gráfica de barras, por lo que enviamos la cadena o *string* **"bar"** y nuestra gráfica será generada sin un grid, por lo que enviamos en el parámetro **grid** el valor *False*.

```
ax = datos.plot(x="Facility Name", y="Hydrogen peroxide", kind="bar",
grid=False)
```

Ahora manipulamos el objeto de nuestra gráfica utilizando la variable ax, donde le colocaremos un título descriptivo a nuestra grafica.

```
ax.set_title('Presencia de Peróxido de Hidrógeno en New Brunswick,
Canadá.')
```

Con la siguiente función podemos agregar a nuestra gráfica un identificador para el eje *X*.

```
ax.set_xlabel("Instalaciones")
```

Con la siguiente función podemos agregar a nuestra gráfica un identificador para el eje *Y*.

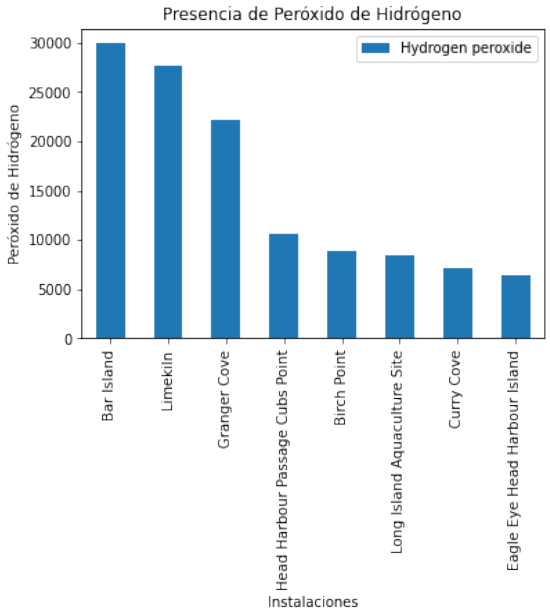
```
ax.set_ylabel("Peróxido de Hidrógeno")
```

Y con la siguiente instrucción mostramos la gráfica. Es importante mencionar que en Jupyter Notebook no es necesario agregar esta instrucción, ya que el interprete mostrará la gráfica generada.

```
plt.show()
```

Por lo que obtendremos como resultado la gráfica de la imagen 3.

Imagen 3: Diagrama de pareto de la presencia de Peróxido de Hidrógeno en New Brunswick, Canadá.



Gráfica con Plotly

Podemos utilizar Plotly para poder crear aplicaciones web de gráficas interactivas para crear *dashboards* dinámicos, pudiendo aprovechar esta librería en conjunto con *Dash*.

Para poder realizar la gráfica de barras con *Plotly*, necesitamos tener instalado el paquete, si no lo hemos instalado, podemos hacerlo por medio del comando *pip* o *pip3*. En mi caso haré 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 *bar()* de plotly para realizar la gráfica, donde el primer parámetro del método corresponde a la variable que contiene el *DataFrame* con los datos que deseamos gráficar; en nuestro caso es **datos**. Luego por medio del parametro **x**, indicaremos los datos **eje X** de nuestra gráfica, la cual corresponde al nombre de la columna que contiene el nombre de las instalaciones (*Facility Name*). El parámetro **y** nos sirve para indicar que datos del *DataFrame* que corresponderán al **eje Y**, el cual es la cantidad de peróxido de hidrógeno (*Hydrogen peroxide*) .

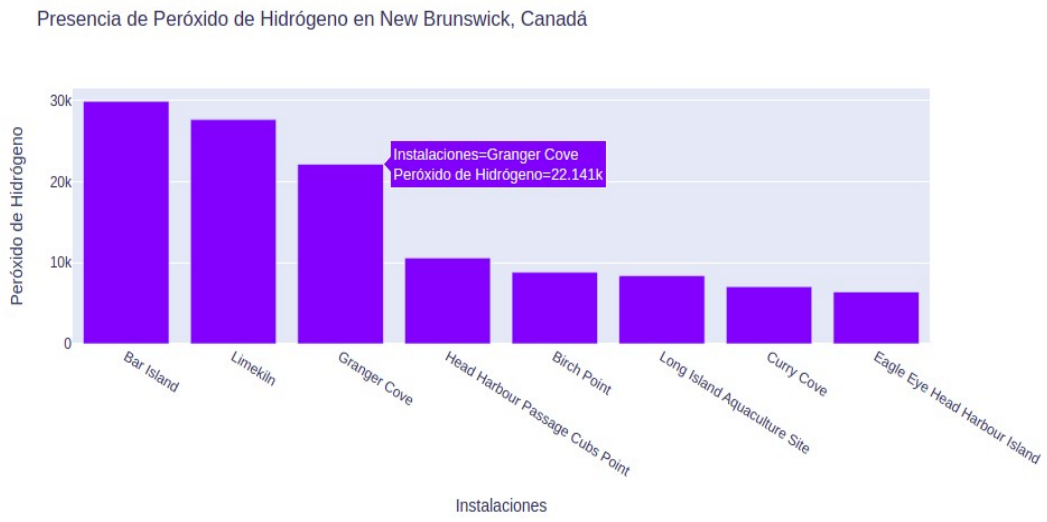
El parámetro **title** nos sive para colocar un nombre descriptivo a nuestra gráfica, y el parámetro **labels** nos sirve para cambiarle el nombre a los ejes de nuestra gráfica. Estos deben ser enviados como un diccionario de Python, por si se desea utilizar una variable para defirnir los nombres de forma separada.

```
fig = px.bar( datos
            ,x="Facility Name"
            ,y="Hydrogen peroxide"
            ,title="Presencia de Peróxido de Hidrógeno"
            ,labels={
                    "Facility Name": "Instalaciones",
                    "Hydrogen peroxide": "Peróxido de Hidrógeno"
                    }
            )
```

Por último, mostramos nuestra gráfica con el siguiente comando:

```
fig.show()
```

Es importante mencionar que en Jupyter Notebook, nuestra gráfica se mostrará 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.

Newbold, P., Carlson, W. y Thorne, B. (2008). *Estadística para administración y economía*. Madrid, España: PEARSON. 10, 14 – 15 pp.