Actividad - Estadística básica

- **Nombre:** Sebastian Flores Lemus
- **Matrícula:** A01709229

Entregar: Archivo PDF de la actividad, así como el archivo .ipynb en tu repositorio. **Nota:** Recuerda habrá una penalización de **50** puntos si la actividad fue entregada fuera de la fecha límite.

Carga el conjunto de datos `bestsellers with categories.csv` (se encuentra en el repositorio de la clase) y realiza un análisis estadístico de las variables.

```
# Carga las librerías necesarias.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
```

```
# Carga el conjunto de datos al ambiente de Google Colab y muestra los primeros
# 6 renglones.
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))

df = pd.read_csv('bestsellers with categories.csv')
df.head(6)
```

El conjunto de datos es una tabla que contiene el top 50 de los libros más vendidos por Amazon por año desde 2009 hasta 2019. Cada libro está clasificado como Ficción o No ficción.

Las variables que contiene son:

- **Name**: Nombre del libro.
- **Author**: Autor.
- **User Rating**: Calificación promedio que los usuarios asignaron al libro (1-5).
- **Reviews**: Número de reseñas.
- **Price**: Precio del libro.
- **Year**: Año de publicación.
- **Genre**: Género literario (ficción/no ficción).

Análisis estadístico

1. Carga la tabla de datos y haz un análisis estadístico de las variables.

- Verifica la cantidad de datos que tienes, las variables que contiene cada vector de datos e identifica el tipo de variables.
- Analiza las variables para saber que representa cada una y en que rangos se encuentran. Si la descripción del problema no te lo indica, utiliza el máximo y el mínimo para encontrarlo.
- Basándote en la media, mediana y desviación estándar de cada variable, ¿qué conclusiones puedes entregar de los datos?
- Calcula la correlación de las variables que consideres relevantes.

```
# Escribe el código necesario para realizar el análisis estadístico descrito
# anteriorment.

df.shape

df.size

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550 entries, 0 to 549
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Name         550 non-null    object
 1   Author       550 non-null    object
```

```
 2   User Rating  550 non-null    float64
 3   Reviews      550 non-null    int64
 4   Price        550 non-null    int64
 5   Year         550 non-null    int64
 6   Genre        550 non-null    object
dtypes: float64(1), int64(3), object(3)
memory usage: 30.2+ KB
```

```
df.describe()
```

¿Cuáles son las variables relevantes e irrelevantes para el análisis?

Para trabajar con datos y gráficas solamente podemos trabajar con valores numéricos y no nos interesan los textos en si. Sin duda los valores irrelevantes serian el nombre del libro y el del autor ya que no se pueden hacer análisis sobre eso. Otra variable que se nos recomendó ignorar es el ano ya que aunque es un valor numérico y podemos obtener información y detalles de el, no tiene tanto impacto en el libro, una persona no compra un libro solamente por el ano en que se escribió. Si podemos rescatar información pero no es el valor mas importante. Sin duda los valores que considero mas importantes son los reviews el precio y el genre. El 'genre' no es un valor numérico pero habla de que es el libro y eso tiene mucho impacto. EL precio de un libro es esencial para que la gente lo pueda comprar y los reviews son esenciales ya que nos dicen si el libro es recomendable o no. Mucha gente se basa en los reviews del usuario y el precio para comprar un libro.

Análisis gráfico

Realiza el análisis de las variables usando diagramas de cajas y bigotes, histogramas y mapas de calor.

Responde las siguientes preguntas:

- ¿Hay alguna variable que no aporta información? Si tuvieras que eliminar variables, ¿cuáles quitarías y por qué?
- ¿Existen variables que tengan datos extraños?
- Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte?
- ¿Puedes encontrar grupos qué se parezcan? ¿Qué grupos son estos?

Haz un análisis estadístico de los datos antes de empezar con la segmentación. Debe contener al menos:

- 1 gráfico de caja (boxplot)
- 1 mapa de calor
- 1 gráfico de dispersión

Describe brevemente las conclusiones que se pueden obtener con las gráficas.

```
# heatmap

iris_corr = df.corr()
sns.heatmap(data=iris_corr,annot=True)
```

```
# boxplot

fig, axs = plt.subplots(1, 3, figsize=(12, 4))

# Graficamos los tres boxplot en una sola imagen.
sns.boxplot(data=df, y = 'Genre', x='Reviews', ax=axs[0])
sns.boxplot(data=df, y = 'Genre', ax=axs[1], x='User Rating')
```

```
sns.boxplot(data=df, y = 'Genre', ax=axs[2], x='Price')

# Esta opción es para que se ajusten las imágenes a la cuadrícula
plt.tight_layout()

# Esta opción es para poner un título general para las tres gráficas
plt.suptitle('Distribución de las variables numéricas por género', y=1.05)
```

```
# dispersion

# Gráfico scatterplot.

sns.scatterplot(data=df, x ='Reviews', y='Price', hue='Year')
sns.color_palette(['red', '#33AAFF','#FFDD33', 'blue','black'])


# Ejes y título. Colocamos la etiqueta correcta de acuerdo a la orientación.
plt.title('Relación entre el número de reseñas, precio y año de publicación')
plt.xlabel('Reviews')
plt.ylabel('Price')
```

Mi primera observación seria al ver el gráfico de correlaciones. Nos damos cuenta que nuestro valor mas alto positivo es '0.26' esto nos dice que no hay una fuerte relación o correlación entre las variables de nuestro test. DE igual manera, el valor mas alto negativo es '-0.15' esto nos dice que tampoco nos encontramos con una fuerte correlación negativa entre los datos. Como estos valore se mantienen cerca a 0 podemos concluir que la correlación entre los datos no es la mayor y no hay tanto impacto.

Analizando el gráfico de dispersión donde grafique 'Reviews' en el eje x, y Precio en el eje-y si podemos hacer ciertas conclusiones. Nos damos cuenta que los libros con muchos reviews se encuentran muy abajo en el tema de precios (menor a 40), y que los libros mas costosos tiene muchos menos reviews. Esto nos sugiere que los libros mas baratos son los que mas movimiento tienen ya que mas gente esta dispuesto a comprarlos, y que los libros que son mas caros tienen menos ventas. Este analisis puede ser un tanto obvio y no nos describe sobre el contenido del libro en si, pero su podemos hacer conclusiones. De igual manera, siento que la variable del ano no tiene tanto valor ya que la gente no compra un libro por el ano en que fue escrito, hay otras variables que tiene un impacto mucho mas grande.

Viendo en análisis de caja y bigote mi primer análisis seria ver que tenemos muchos datos que son muy grandes comparados a los demás. Esto no es lo mejor ya que esos datos pueden arruinar los resultados estadísticos. Tener un rango tan grande en las variables puede quitar valor a nuestro estudio. Pero, este análisis nos sirve ya que nos ensena los cuartiles de los datos y nos muestra un rango de los valores mas populares lo cual tiene mucho valor.

Clústering

Una vez que hayas realizado un análisis preliminar, haz una segmentación utilizando el método de K-Means. Justifica el número de clusters que elegiste.

- Determina un valor de k
- Calcula los centros de los grupos resultantes del algoritmo k-means

Basado en los centros responde las siguientes preguntas

- ¿Crees que estos centros puedan ser representativos de los datos? ¿Por qué?
- ¿Cómo obtuviste el valor de k a usar?
- ¿Los centros serían más representativos si usaras un valor más alto? ¿Más bajo?
- ¿Qué pasaría con los centros si tuviéramos muchos outliers en el análisis de cajas y bigotes?
- ¿Qué puedes decir de los datos basándose en los centros?

```
# Implementa el algoritmo de kmeans y justifica la elección del número de
# clusters. Usa las variables numéricas.

#from sklearn.preprocessing import StandardScaler
# scaler = StandardScaler()
# X_norm = scaler.fit_transform(X)

# Vamos a escalar las tres variables con StandardScaler, el cual se encuentra
# en SciKitLearn
from sklearn.preprocessing import StandardScaler

# Seleccionamos las variables a normalizar
numeric_cols = ['User Rating', 'Reviews',   'Price', 'Year']
X = df.loc[:, numeric_cols]

# Hacemos el escalamiento.
scaler = StandardScaler()
X_norm = scaler.fit_transform(X)

# El escalador nos genera una matriz de numpy. Vamos a convertirlo en DF
X_norm = pd.DataFrame(X_norm, columns=numeric_cols)
X_norm.head()
```

```
# # Importamos librerias en caso de no haberlo hecho antes
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Declaramos algunos arreglos. Los usuaremos para guardar los valores de la WCSS
 # y la silhouette score
kmax   = 16
grupos = range(2, kmax)
wcss   = []
```

```
sil_score = []

# Ciclo para calcular K-Means para diferentes k
for k in grupos:
    # Clustering
    model = KMeans(n_clusters=k, random_state = 47)

    # Obtener las etiquetas
    clusters = model.fit_predict(X_norm)

    # Guardar WCSS
    wcss.append(model.inertia_)

    # Guardar Silhouette Score
    sil_score.append(silhouette_score(X_norm, clusters))
```

Double-click (or enter) to edit

```
# Graficaremos el codo y silhouette score en la misma gráfica. Recorda que
# subplots nos permite tener más gráficas en la misma figura.
fig, axs = plt.subplots(1, 2, figsize=(15, 6))

# Primera figura es el codo
axs[0].plot(grupos, wcss)
axs[0].set_title('Método del codo')

# La segunda es el Silhouette Score
axs[1].plot(grupos, sil_score)
axs[1].set_title('Silhouette Score')
```

Analiza las características de cada grupo. ¿Qué nombre le pondrías a cada segmento?

Es difícil analizar y ver que variable es la responsable que define en que grupo se encuentra cada libro. Nosotros no sabemos cual es la que tiene mas impacto pero nos podemos dar una idea. Es clave resaltar que de acuerdo al análisis que nosotros hicimos se nos recomendó hacer únicamente 4 grupos para nuestros valores. Analizando 20 datos de la tabla podemos inferir que el valor que mas 'ponderación' o mas efecto puede tener son las reviews, ya que este es el dato que mas cambia y mas rango tiene. El user rating es un valor entre 0 y 5 donde la mayoría se encuentra en 4 entonces si cambia un par de decimales no tiene un impacto significativo, el precio igualmente se mantiene de 90 unidades la cual si tiene mas valor que otras unidades pero comparada con los reviews no observamos una gran diferencia. Para la variable de ano solo vemos un rango de 10 el cual no tiene ningún efecto impactante en nuestro experimento. Por ultimo nuestra variable 'review' tiene un rango o diferencia entre nuestro valor mayor y el valor menor de 87,400 unidades. Tener tanta dispersión en esta variable afecta por completo el análisis, y sin duda la variable que mas importancia tiene para la clasificación de los grupo es esta.

Si en nuestra tabla vemos el numero de reviews que tienen los elementos en el grupo 1 vemos que están cerca de los 2000 reviews. Los que sen encuentran en el grupo 2 tienen cerca de 4,000 reviews, y finalmente los que están en el grupo 3 se encentran arriba de 5,000 reviews. Entonces el grupo 0 lo llamaría '0 - 2000' reviews, el grupo 1 '2000-4000 reviews' El grupo 2 '4000 - 5000' y el ultimo grupo arriba de 5,000. Claro que este análisis no es perfecto ya que hay mas variables que también afectaran el posicionamiento de las variables en los grupos, pero sin duda el mas impactante es el de las reviews.

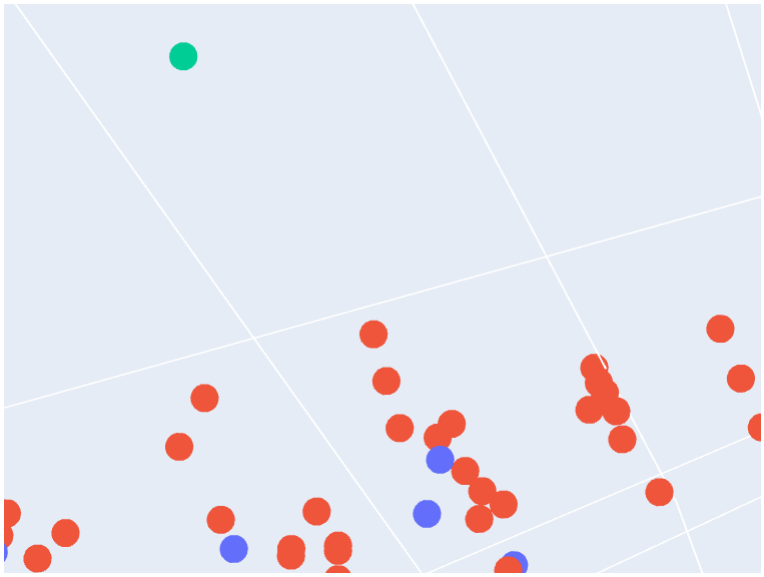
Double-click (or enter) to edit

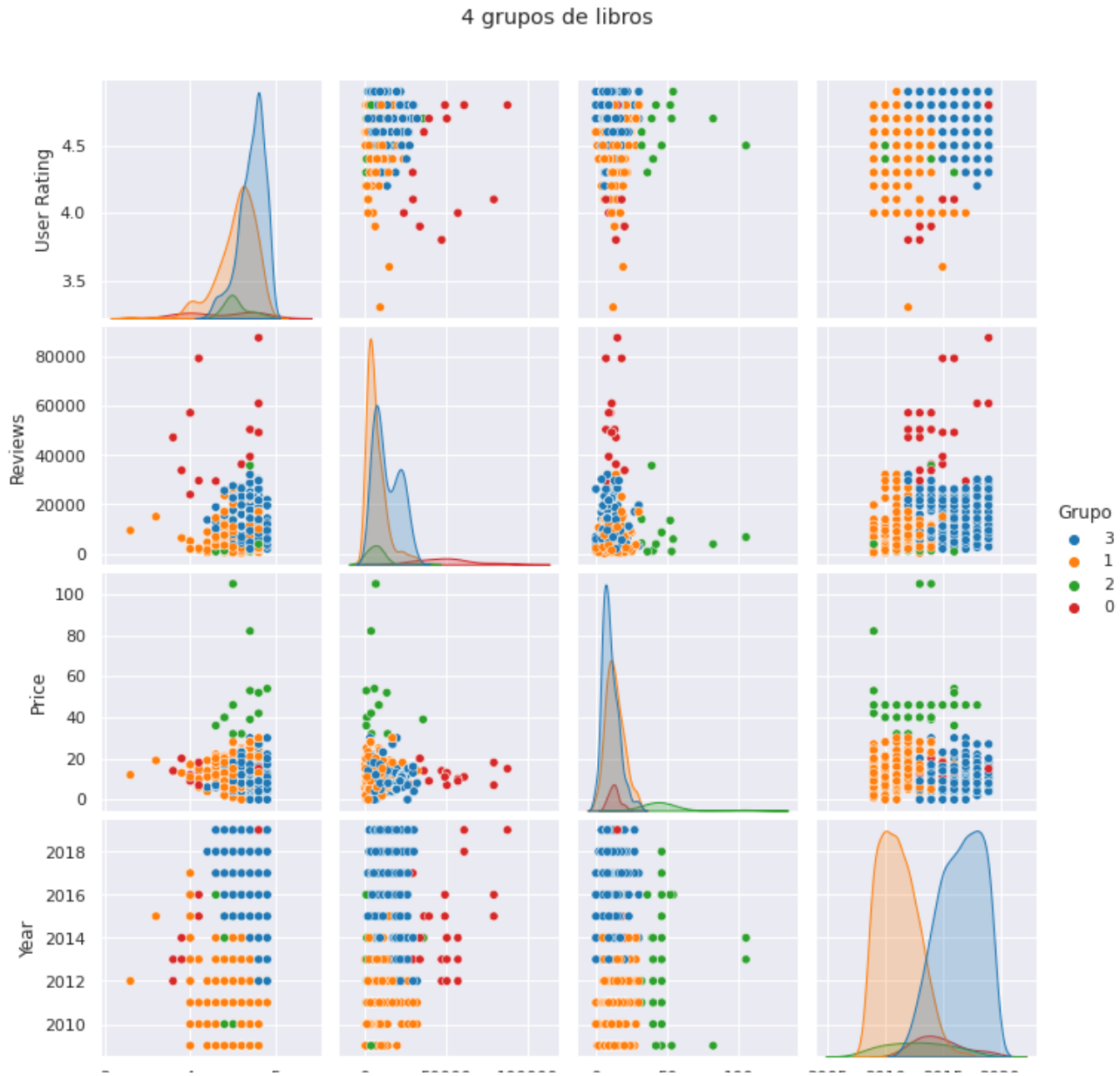
```
# Haz un análisis por grupo para determinar las características que los hace
# únicos. Ten en cuenta todas las variables numéricas.

df.groupby('Grupo').mean()
```

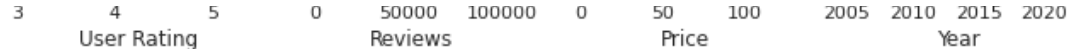
```
# Dispersiones por grupo

df.groupby('Grupo').std()
```

```
# Grafica los grupos con un pairplot y con un scatterplot en 3D
# (si es necesario). Analiza las características de cada grupo.

sns.pairplot(data=df, hue='Grupo', palette='tab10')
plt.suptitle('4 grupos de libros', y=1.05)
```

```
# importar una librería más
import plotly.express as px

# Creamos la figura donde graficaremos
fig = px.scatter_3d(df, x = 'User Rating', y = 'Reviews',
                    z = 'Price',
                    title='4 grupos de libros',
                    color='Grupo')#,
                    #color_discrete_sequence=px.colors.qualitative.D3)

# mostramos la imagen
fig.show()
```


2
0


4 grupos de libros
User Rating
4.5
4.0
3.5
Reviews
80000
60000
40000
20000
0
Price
100
80
60
40
20
0
Year
2018
2016
2014
2012
2010
Grupo
3
1
2
0



✓ 0s completed at 9:25 PM