Actividad - Estadística básica

- **Nombre: Daniel Queijeiro Albo**
- **Matrícula: A01710441**

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.
!curl -O https://raw.githubusercontent.com/Manchas2k4/tc1002S/main/datasets/bestsellers%20with%20categories.csv

df = pd.read_csv('bestsellers%20with%20categories.csv')
df.head(6)
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 51161  100 51161    0     0   601k      0 --:--:-- --:--:-- --:--:--  601k
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
0	10-Day Green Smoothie Cleanse	JJ Smith	4.7	17350	8	2016	Non Fiction
1	11/22/63: A Novel	Stephen King	4.6	2052	22	2011	Fiction
2	12 Rules for Life: An Antidote to Chaos	Jordan B. Peterson	4.7	18979	15	2018	Non Fiction
3	1984 (Signet Classics)	George Orwell	4.7	21424	6	2017	Fiction
4	5,000 Awesome Facts (About Everything!)	National Geographic	4.8	7665	12	2019	Non

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.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()
```

	User Rating	Reviews	Price	Year
count	550.000000	550.000000	550.000000	550.000000
mean	4.618364	11953.281818	13.100000	2014.000000
std	0.226980	11731.132017	10.842262	3.165156
min	3.300000	37.000000	0.000000	2009.000000
25%	4.500000	4058.000000	7.000000	2011.000000
50%	4.700000	8580.000000	11.000000	2014.000000
75%	4.800000	17253.250000	16.000000	2017.000000
max	4.900000	87841.000000	105.000000	2019.000000

```
df.corr()
```

	User Rating	Reviews	Price	Year
User Rating	1.000000	-0.001729	-0.133086	0.242383
Reviews	-0.001729	1.000000	-0.109182	0.263560
Price	-0.133086	-0.109182	1.000000	-0.153979
Year	0.242383	0.263560	-0.153979	1.000000

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

** Escribe la respuesta **

Trás el análisis de correlación sabemos que las variables relevantes son User Rating, Reviews, y Price, mientras que Year resulta irrelevante.

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? Si, eliminariamos las variables de nombre, autor, género y year.
- ¿Existen variables que tengan datos extraños? No
- Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte? No estan en rangos similares y eso si puede llegar a afectar.
- ¿Puedes encontrar grupos qué se parezcan? ¿Qué grupos son estos? Si, Price y User rating

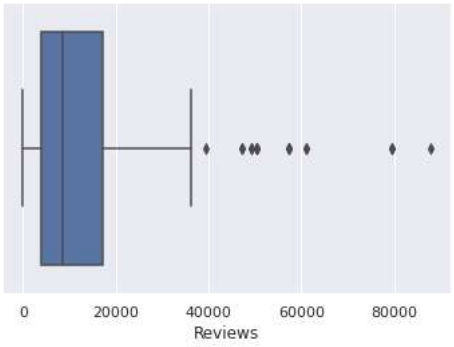
Haz un análisis estadístico de los datos antes de empezar con la segmentación. Debe contener al menos:

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

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

```
#1 gráfico de caja (boxplot)
ndf = df.drop('Name',axis=1)
ndf = ndf.drop('Author',axis=1)
ndf = ndf.drop('Genre',axis=1)
ndf= ndf.drop('Year',axis=1)

sns.boxplot(data=ndf, x='Reviews')
```

```
<Axes: xlabel='Reviews'>
```

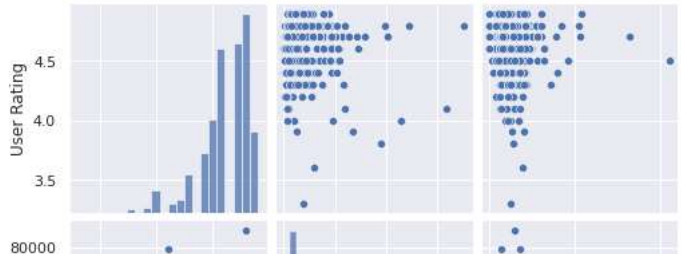



```
#1 mapa de calor
ndf_corr = ndf.corr()

sns.heatmap(data=ndf_corr,vmin=-1, vmax=1, cmap = 'RdBu', annot=True, square = True)
```

```
<Axes: >
```


```
#1 gráfico de dispersión
sns.pairplot(data=ndf)
```

```
<seaborn.axisgrid.PairGrid at 0x7fb3dafa0be0>
```



** Escribe tus conclusiones **

Viendo la grafica pairplot podemos llegar a la conclusión que por las imagenes espejo todas las variables pueden ser empleadas para el análisis.

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 Usaremos 4 como valor de k
- Calcula los centros de los grupos resultantes del algoritmo k-means /Abajo utilizamos codigo para calcular los centros/

Basado en los centros responde las siguientes preguntas

- ¿Crees que estos centros puedan ser representativos de los datos? ¿Por qué? Si, los centros obtenidos del algoritmo K-Means pueden ser representativos de los datos, ya que se calculan utilizando la media de los valores en cada dimensión y corresponden a los puntos más cercanos a cada centroide.
- ¿Cómo obtuviste el valor de k a usar? Para obtener el valor de k a utilizar, podemos observar los datos de los gráficos como el pairplot. En este caso, se puede observar que los datos podrían agruparse en 3 o 4 clusters, dependiendo de la elección de las variables. Utilizaremos k=4.
- ¿Los centros serían más representativos si usaras un valor más alto? ¿Más bajo? Si se utilizara un valor de k más alto, se podrían obtener centros más específicos, pero podría ser más difícil interpretar los resultados. Por otro lado, si se utiliza un valor más bajo, se podría obtener una representación más general de los datos.
- ¿Qué pasaría con los centros si tuviéramos muchos outliers en el análisis de cajas y bigotes? Si hubiera muchos outliers en el análisis de cajas y bigotes, los centros podrían inclinarse más hacia los valores extremos.
- ¿Qué puedes decir de los datos basándose en los centros? Los datos pueden agruparse en tres clusters distintos en función del precio, el User Rating y el número de reseñas. Podemos asumir que estos tres factores son importantes para determinar la popularidad y el éxito de un libro.

```
# Implementa el algoritmo de kmeans y justifica la elección del número de
# clusters. Usa las variables numéricas.
from sklearn.preprocessing import StandardScaler

numeric_cols = ['Price','Reviews','User Rating']
x = ndf.loc[:, numeric_cols]

scaler = StandardScaler()
x_norm = scaler.fit_transform(x)

x_norm = pd.DataFrame(x_norm, columns=numeric_cols)
x_norm.head()
```

	Price	Reviews	User Rating
0	-0.470810	0.460453	0.359990
1	0.821609	-0.844786	-0.080978
2	0.175400	0.599440	0.359990
3	-0.655441	0.808050	0.359990
4	-0.101547	-0.365880	0.800958

```
#Centros de los grupos
# Seleccionar las características a utilizar
for k in range(2,4):
  # Calcular el k-means
  model = KMeans(n_clusters = k)
  # Obtener los grupos o clusters
  groups = model.fit_predict(x)
  # Los centros de los grupos se guardan en cluster_centers_
  centros = model.cluster_centers_
  print(centros)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

[[1.39431280e+01 6.88329621e+03 4.60710900e+00]
 [1.03203125e+01 2.86683906e+04 4.65546875e+00]]
[[1.40202020e+01 6.23506313e+03 4.60479798e+00]
 [1.06231884e+01 2.29664783e+04 4.68115942e+00]
 [1.16875000e+01 5.84903750e+04 4.41250000e+00]]
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

kmax   = 6
grupos = range(2, kmax)
wcss   = []
sil_score = []

for k in grupos:
    model = KMeans(n_clusters=k, random_state = 47)

    clusters = model.fit_predict(x_norm)

    wcss.append(model.inertia_)

    sil_score.append(silhouette_score(x_norm, clusters))

fig, axs = plt.subplots(1, 2, figsize=(15, 6))

axs[0].plot(grupos, wcss)
axs[0].set_title('Método del codo')

axs[1].plot(grupos, sil_score)
axs[1].set_title('Silhouette Score')
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl

/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl

/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
```

Tomamos el punto 4 como total de clusters ya que es el punto en el que se ve un cambio en la grafica de codo y en la silhouette score es el punto mas alto.

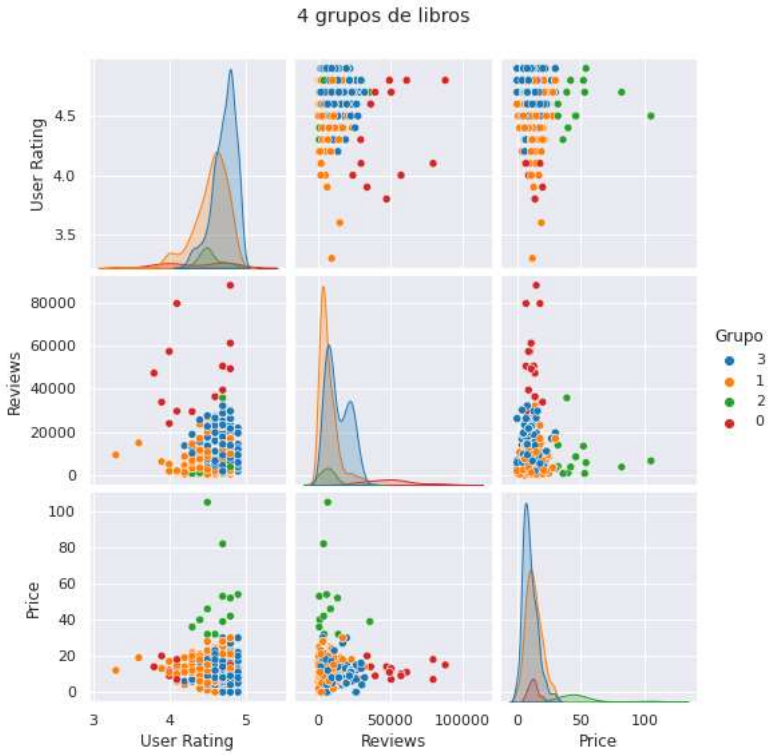
```
# Generamos los 4 grupos
model = KMeans(n_clusters=4, random_state=47)
clusters = model.fit_predict(x_norm)

# Agregamos los clusters a nuestros DATOS ORIGINALES
ndf['Grupo'] = clusters.astype('str')
ndf.head()
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitl
```

	User Rating	Reviews	Price	Grupo
0	4.7	17350	8	0
1	4.6	2052	22	2
2	4.7	18979	15	0
3	4.7	21424	6	0
4	4.8	7665	12	2

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

```
Text(0.5, 1.05, '4 grupos de libros')
```



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

Grupo 0 Este grupo podría llamarse "Bestsellers", porque contiene libros que tienen altas calificaciones y reseñas, lo que sugiere que son muy populares entre los lectores.

Grupo 1 Este grupo podría llamarse "Clásicos", porque contiene libros que tienen calificaciones intermedias, gran cantidad de reseñas, lo que sugiere que son entre sus varios lectores a algunos les gusta y a otros no.

Grupo 2 Este grupo podría llamarse "Podría interesarte", porque contiene libros que tienen altas calificaciones, y una cantidad intermedia de reseñas, lo que sugiere que son del agrado ente sus lectores.

Grupo 3 Este grupo podría llamarse "Poco conocidos", porque contiene libros que tienen altas calificaciones, pero muy baja cantidad de reseñas, lo que sugiere que son buenos libros pero que no tienen tanto reconocimiento.

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

#Las características de cada grupo
ndf.groupby('Grupo').mean()
```

Grupo	User Rating	Reviews	Price
0	4.693846	27444.646154	9.084615
1	4.232143	8631.666667	12.416667
2	4.698065	6753.977419	11.900000
3	4.538462	7219.538462	49.692308

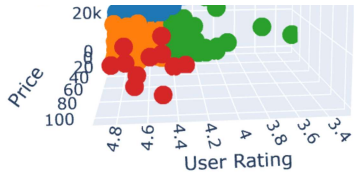
```
#Las dispersiones
ndf.groupby('Grupo').std()
```

Grupo	User Rating	Reviews	Price
0	0.184161	12779.526505	3.833825
1	0.208933	9097.337152	5.013736
2	0.118770	4145.890023	6.819423
3	0.144435	6978.798305	18.750508

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


import plotly.express as px

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

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

4 grupos de libros

El grupo 0 corresponde al tipo de libros con un precio bajo, mayor cantidad de reseñas y un User Rating alto. El grupo 1 corresponde al tipo de libros con un precio bajo, mediana cantidad de reseñas y el User Rating mas bajo. El grupo 2 corresponde al tipo de libros con un precio bajo, baja cantidad de reseñas y un User Rating alto. El grupo 3 corresponde al tipo de libros con un precio alto, menor cantidad de reseñas y el User Rating mas alto.



✓ 0 s se ejecutó 16:05