Actividad - Estadística básica

- **Nombre: Oscar Eduardo Nieto Espitia**
- **Matrícula: A01705090**

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

```
# 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. 550 entradas
- 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.

print('Media:')
```

```
print('User Rating: ', df['User Rating'].mean())
print('Reviews: ', df['Reviews'].mean())
print('Price: ', df['Price'].mean())
print('')
print('Moda:')
moda = pd.Series(df['User Rating'].values.flatten()).mode()[0]
print('User Rating: ', moda)
moda1 = pd.Series(df['Reviews'].values.flatten()).mode()[0]
print('Reviews: ', moda1)
moda2 = pd.Series(df['Price'].values.flatten()).mode()[0]
print('Price: ', moda2)
print('')
print('Mediana:')
print('User Rating:', df['User Rating'].median())
print('Reviews:', df['Reviews'].median())
print('Price:', df['Price'].median())
print('')
print('Varianza y Desviacion estandar:')
print('Varianza User Rating: ', df['User Rating'].var(), 'Desviación estándar: ', df['User Ra
print('Varianza Reviews: ', df['Reviews'].var(), 'Desviación estándar: ', df['Reviews'].std()
print('Varianza Price: ', df['Price'].var(), 'Desviación estándar: ', df['Price'].std())
```

```
Media:
User Rating:  4.618363636363641
Reviews:  11953.281818181818
Price:  13.1

Moda:
User Rating:  4.8
Reviews:  8580
Price:  8

Mediana:
User Rating: 4.7
Reviews: 8580.0
Price: 11.0

Varianza y Desviacion estandar:
Varianza User Rating:  0.05152008610697112 Desviación estándar:  0.22698036502519578
Varianza Reviews:  137619458.4104157 Desviación estándar:  11731.132017431895
Varianza Price:  117.55464480874357 Desviación estándar:  10.84226197842238
```

```
df1 = df
df1 = df1.drop('Name', axis=1)
df1 = df1.drop('Author', axis=1)
df1 = df1.drop('Genre', axis=1)
df1 = df1.drop('Year', axis=1)
df1.corr()
```

```
df['Author'].value_counts()
```

```
Jeff Kinney                          12
Gary Chapman                         11
Rick Riordan                         11
Suzanne Collins                      11
American Psychological Association   10
                                     ..
Keith Richards                        1
Chris Cleave                          1
Alice Schertle                        1
Celeste Ng                            1
Adam Gasiewski                        1
Name: Author, Length: 248, dtype: int64
```

```
df.groupby(['Year', 'Genre']).mean()[['Price']]
```

```
df.groupby(['Year', 'Genre']).agg(['min', 'max'])[['Price', 'User Rating']]
```

```
df.groupby('Author').mean()[['Reviews']]
```

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

Considero que todas las variables recabadas son relevantes, esto se debe a que se puede inferir muchas cosas a traves de un analisis estadistico sin necesidad de solo ocupar las variables numericas, a pesar de que no pude hacer una consulta relevante en donde involucre el nombre del libro, no quiere decir que esta sea irrelevante.

Como mostre anteriormente, pude obtener consultas interesantes como cual es el precio minimo y maximo de los las categorias de libros vendidos a traves del tiempo, tambien pude sacar la media de reviews por cada autor ademas de las veces que ha estado en el top de ventas un autor. Además de esto pude obtener media, mediana, moda, varianza y desviacion estandar de las variables numericas, de las cuales se pueden llegar a inferir varias cosas.

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? Depende de que quiera analizar, por ejemplo si quisiera hacer un heatmap tendria que tomar las variables numericas
- ¿Existen variables que tengan datos extraños? Nop
- Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte? No, no todas están en rangos similares, de hecho hay demasiada varianza en Reviews, esto afecta mucho a la hora de graficar, se tendría que hacer una normalización
- ¿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.

```
fig = plt.figure(figsize=(6,4))

# Graficamos los incresos contra el límite de crédito
plt.plot(df1['Reviews'], df1['User Rating'], '*')

# Agregamos títulos a los ejes y al gráfico
plt.xlabel('User Rating')
plt.ylabel('Reviews')
plt.title('Gráfico User Rating vs Reviews')

# Aquí la leyenda hace mucho más sentido
plt.legend(loc='best')

# Agregamos la cuadrícula para que se vea mejor
plt.grid(True)
```

```
fig = plt.figure(figsize=(6,4))

# Gráfico countplot para hacer barras con el número de apariciones de cada especie.
sns.countplot(data=df, x = 'Genre')

# Ejes y título. Colocamos la etiqueta correcta de acuerdo a la orientación.
plt.title('Observaciones de cada especie')
plt.xlabel('Especie')
plt.ylabel('Frecuencia')
```

```
df2 = df
df2 = df2.drop('Name', axis=1)
df2 = df2.drop('Author', axis=1)
df2 = df2.drop('Genre', axis=1)
df2 = df2.drop('Reviews', axis=1)
```

```
fig = plt.figure(figsize=(6, 4))

# Gráfico scatterplot.
sns.scatterplot(data=df, x ='User Rating', y='Year', hue='Genre')

# Ejes y título. Colocamos la etiqueta correcta de acuerdo a la orientación.
plt.title('Relacion entre el User Rating y el Año')
plt.xlabel('Año')
plt.ylabel('User Rating')
```

```
df1 = df
df1 = df1.drop('Name', axis=1)
df1 = df1.drop('Author', axis=1)
df1 = df1.drop('Genre', axis=1)
df1 = df1.drop('Year', axis=1)
iris_corr = df1.corr()
sns.heatmap(data=iris_corr, vmin=-1, vmax=1, cmap = 'RdBu', annot=True, square = True)
```

```
fig = plt.figure(figsize=(7,5))
sns.boxplot(data=df, x='User Rating', y = 'Genre')
plt.title('Histograma del User rating por genero')
```

Conclusiones: Están dentro del top 50 el tipo de genero que no es de ciencia Ficción, sin embargo hay que recordar que en proporcion, existen menos libros de ciencia ficción, por lo tanto se puede asumir que el genero de ciencia ficción tiene mayor impacto dentro del top 50 libros más vendidos.

En cuento a la relación entre el User Rating y el año se puede concluir que no ha cambiado a través del tiempo.

Si hablamos de la correlación que existe entre las variables númericas se puede concluir que existe una correlación baja entre las variables, es decir, que una no afecta a la otra

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?

```
from sklearn.preprocessing import StandardScaler

# Seleccionamos las variables a normalizar
numeric_cols = ['User Rating', 'Price']
X = df1.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()
```

```
# Declaramos algunos arreglos. Los usuaremos para guardar los valores de la WCSS
 # y la silhouette score
from sklearn.cluster import KMeans
# Importar la librería para silhouette score
from sklearn.metrics import 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))


    # 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')
```

```
    # Implementa el algoritmo de kmeans y justifica la elección del número de
    # clusters. Usa las variables numéricas.
    # Generamos los 6 grupos
    model = KMeans(n_clusters=4, random_state=47)
    clusters = model.fit_predict(X_norm)

    # Agregamos los clusters a nuestros DATOS ORIGINALES
    df1['Grupo'] = clusters.astype('str')
    df1.head()
```

```
sns.pairplot(data=df1, hue='Grupo', palette='Set2')
plt.suptitle('6 grupos de clientes', y=1.05)
```

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

** Escribe la respuesta **

```
# Haz un análisis por grupo para determinar las características que los hace
# únicos. Ten en cuenta todas las variables numéricas.
df1.groupby('Grupo').mean()
```

Grupo	User Rating	Reviews	Price
0	4.750588	12912.882353	7.231373
1	4.226087	12331.108696	12.228261
2	4.537500	7066.458333	51.166667
3	4.642458	11047.279330	16.804469

```
df1.groupby('Grupo').std()
```

Grupo	User Rating	Reviews	Price
0	0.105693	10046.248224	3.154393
1	0.206419	16132.552289	4.948311
2	0.149819	7108.207968	18.779421
3	0.110599	11636.795993	5.601041

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

Haz doble clic (o pulsa Intro) para editar

✓ 0 s completado a las 13:07