Actividad - Estadística básica

- **Nombre:**
- **Matrícula:**

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 pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Carga el conjunto de datos al ambiente de Google Colab y muestra los primeros
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])))

# 6 renglones.

```

Choose Files bestsellers …tegories.csv

- **bestsellers with categories.csv**(text/csv) - 51161 bytes, last modified: 3/21/2023 - 100% done

```
Saving bestsellers with categories.csv to bestsellers with categories.csv
User uploaded file "bestsellers with categories.csv" with length 51161 bytes
```

```
# 6 renglones.

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

	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
.	5.000 Awesome Facts (About	National					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).

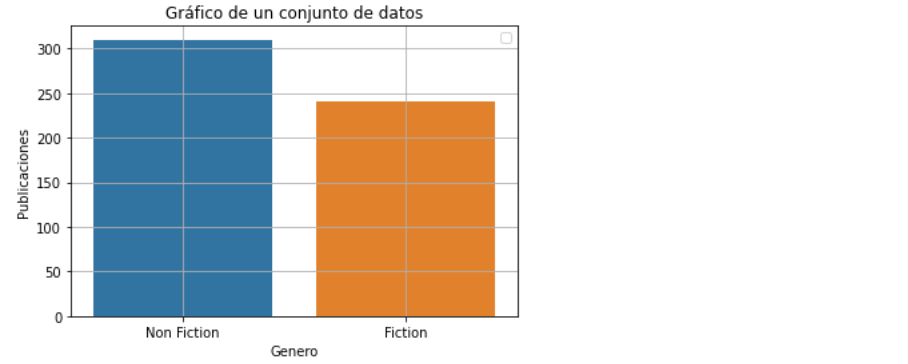
```
# Crea una tabla resumen con las estadísticas generales de las variables
```

```
# Crea una tabla resumen con los estadísticas generales de las variables
# numéricas.
df.describe()
```

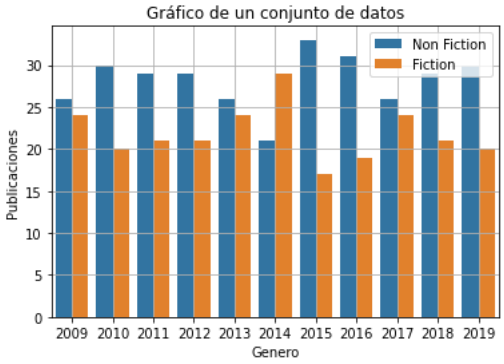
	User Rating	Reviews	Price	Year
count	550.000000	550.000000	550.000000	550.000000
mean	4.618364	11953.281818	13.100000	2014.000000
std	0.226980	11731.132017	10.842262	3.165156
min	3.300000	37.000000	0.000000	2009.000000
25%	4.500000	4058.000000	7.000000	2011.000000
50%	4.700000	8580.000000	11.000000	2014.000000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

```
## ¿Cuál es el género con más publicaciones? Muéstralo en un gráfico.
# Configuramos el tamaño de imagen
fig = plt.figure(figsize=(6,4))

# Graficamos
sns.countplot(data=df, x = 'Genre')

# Agregamos títulos a los ejes y al gráfico
plt.xlabel('Genero')
plt.ylabel('Publicaciones')
plt.title('Gráfico de un conjunto de datos')

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

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

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that arti
```

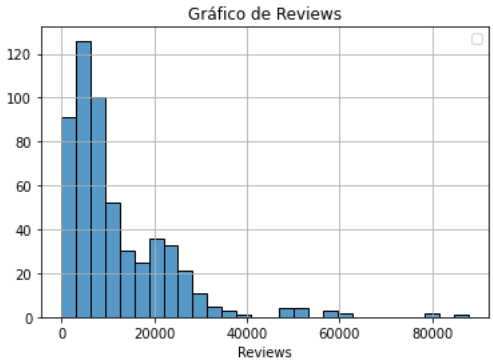



```
# ¿Cuántos libros del top 50 se publicaron por género en cada año? ¿Hay algún
# año donde hubo más libros de ficción en el top 50?. Muéstralo en un gráfico.
# Configuramos el tamaño de imagen
fig = plt.figure(figsize=(6,4))

# Graficamos
sns.countplot(data=df, x = 'Year', hue = 'Genre')

# Agregamos títulos a los ejes y al gráfico
plt.xlabel('Genero')
plt.ylabel('Publicaciones')
plt.title('Gráfico de un conjunto de datos')

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

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



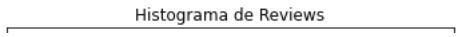

```
# ¿Cómo se distribuye la variable Review? Muéstra el histografa.

# Configuramos el tamaño de imagen
fig = plt.figure(figsize=(6,4))

# Graficamos
sns.histplot(data=df, x='Reviews')

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

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

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

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that arti
```



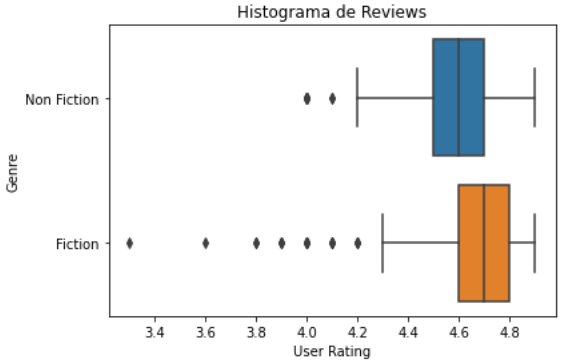

```
# Ahora muéstralo en un gráfico de caja y bigote.
# Configuramos el tamaño de imagen
fig = plt.figure(figsize=(6,4))

# Graficamos
sns.boxplot(data=df, x='Reviews')

# Ejes y título
plt.title('Histograma de Reviews')
```

```
Text(0.5, 1.0, 'Histograma de Reviews')
```



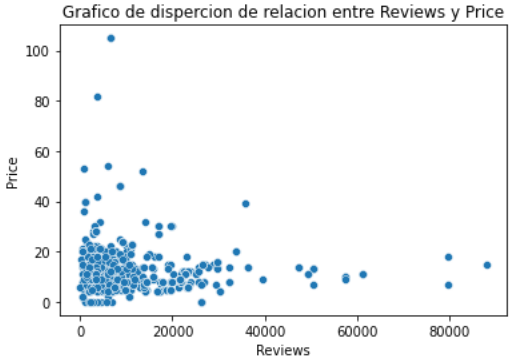

```
# ¿Cómo se compara la evaluación del libro por género? ¿Qué genero es mejor
# evaluado por los lectores? Muéstralo en un solo gráfico de caja y bigote.
fig = plt.figure(figsize=(6,4))

# Graficamos
sns.boxplot(data=df, x='User Rating', y = 'Genre')

# Ejes y título
plt.title('Histograma de Reviews')
```

```
Text(0.5, 1.0, 'Histograma de Reviews')
```

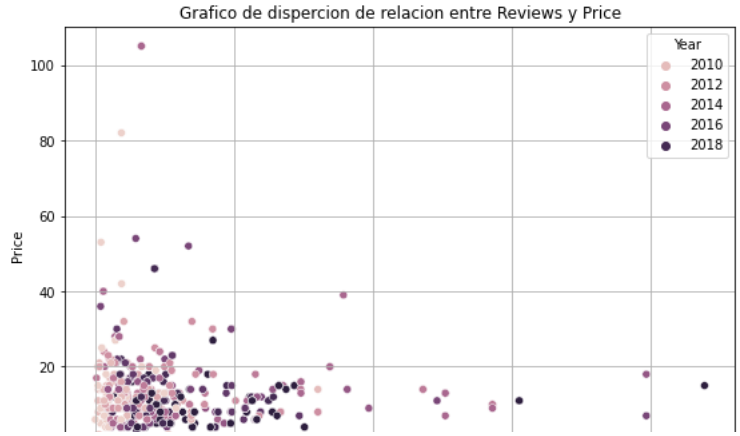



```
# ¿Cuál es la relación entre el número de reseñas y precios? Muéstralo en un
# gráfico de dispersión.

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

# Graficamos
sns.scatterplot(data=df, x='Reviews', y = 'Price')

# Ejes y título
plt.title('Grafico de dispercion de relacion entre Reviews y Price')
```

```
Text(0.5, 1.0, 'Grafico de dispercion de relacion entre Reviews y Price')
```

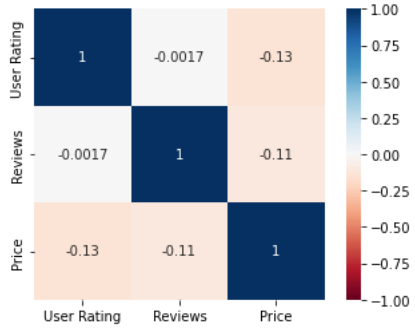



```
# De la pregunta anterior, ¿influye algo el año de publicación? ¿Cuál es la
# relación entre el número de reseñar, el precio y el año de publicación?
# IMPORTANTE: Selecciona una paleta de colores adeacuada.

fig = plt.figure(figsize=(9,6))

# Graficamos
sns.scatterplot(data=df, x='Reviews', y = 'Price', hue = 'Year')

# Ejes y título
plt.title('Grafico de dispercion de relacion entre Reviews y Price')

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




```
# ¿Cuál es la correlación entre las variables numéricas? Muéstralo en un
# gráfico. La variable año, a pesar de ser numérica, la vamos a considerar como
# cualitativa, así que la eliminaremos del análisis.

# Vamos a graficar la matriz de correlación del dataset Iris

iris_corr = df[['User Rating', 'Reviews', 'Price']].corr()

# Gráfico heatmap. Seleccionamos los valores extremos con vmin y vmax.
# El mapa de color que usaremos es de un extremo azul y del otro rojo.
# Con annot podemos desplegar el valor de cada celda
# Con square hacemos que el gráfico sea simétrico en tamaño de ejes
sns.heatmap(data=iris_corr, vmin=-1, vmax=1, cmap = 'RdBu', annot=True, square = True)
```

```
<Axes: >
```



¿Cuáles variables tiene una fuerte relación positiva entre sí y cuáles tienen una fuerte relación negativa? (Esta pregunta no es de código) Responde la pregunta en la siguiente celda de texto.

** Realmente no hay variables que tengan una relacion positiva, solamente entre ellas mismas, sin embargo las que tienen un valor mas acercado al 0 son entre user rating y price, reviews y price. Para aquellas negativas mas alejadas del 0 se encuentran las relaciones entre user ratings y reviews**

```
# Haz una gráfica donde podemos comparar la relación entre las tres variables
# numéricas (User Rating, Reviews y Price) y que, además, podamos ver el efecto
# del libro. La variable año, a pesar de ser numérica, la vamos a considerar como
# cualitativa, así que la eliminaremos del análisis.

sns.pairplot(df[['User Rating', 'Reviews', 'Price']])


```

```
<seaborn.axisgrid.PairGrid at 0x7f31ba6b40a0>
```