```
from google.colab import drive
drive.mount('/gdrive')
```

```
Mounted at /gdrive
```

```
!ls
```

```
sample_data
```

```
%cd /gdrive/MyDrive/TC1002S/
#Importar el Modulo para leer JSON
import json

# Lectura del archivo
with open('credentials.json', 'r') as myfile:
    data = myfile.read()

# Leer el formato del archivo
obj = json.loads(data)

# Vamos a guardar los datos en estas variables
GIT_USERNAME = obj['user']

# token
GIT_TOKEN = obj['token']

# Repo
GIT_REPO = obj['repo']

# Creamos la ruta al repositorio de nuestra cuenta
GIT_PATH = "https://" + GIT_USERNAME + ":" + GIT_TOKEN + "@github.com/" +\
            GIT_USERNAME + "/" + GIT_REPO + ".git"

print(GIT_PATH)
```

```
/gdrive/MyDrive/TC1002S
https://DanyGuti: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@github.com/DanyGuti/SemanaTecTC100
```

```
!ls
```

```
credentials.json  README.md  SemanaTecTC1002S
```

```
%cd SemanaTecTC1002S/
```

```
/gdrive/MyDrive/TC1002S/SemanaTecTC1002S
```

```
!ls
```

```
Actividad5_A01068056.ipynb  credentials.json  datasets  README.md
```

```
!git remote -v
```

```
cursoFuente     https://github.com/DanyGuti/SemanaTecTC1002S.git (fetch)
cursoFuente     https://github.com/DanyGuti/SemanaTecTC1002S.git (push)
origin  https://DanyGuti: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@github.com/DanyGuti/Semana
origin  https://DanyGuti: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@github.com/DanyGuti/Semana
```

```
!git status
```

```
Refresh index: 100% (18/18), done.
On branch main
Your branch is up to date with 'cursoFuente/main'.

nothing to commit, working tree clean
```

```
!git pull
```

```
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 17 (delta 7), reused 12 (delta 4), pack-reused 0
Unpacking objects: 100% (17/17), 1.14 MiB | 3.55 MiB/s, done.
From https://github.com/DanyGuti/SemanaTecTC1002S
   223707b..4668d8f  main       -> cursoFuente/main
Updating 223707b..4668d8f
Fast-forward
 Act6COLLAB.ipynb      | 1986 ++++++++++++++++++++++++++++++++++++++++++++++++++
 Act7_A01068056.ipynb  |  662 ++++++++++++++++++
 Actividad6.ipynb      |  990 +++++++++++++++++++++++++
 Actividad6_Collab.pdf |  Bin 0 -> 818807 bytes
 4 files changed, 3638 insertions(+)
 create mode 100644 Act6COLLAB.ipynb
 create mode 100644 Act7_A01068056.ipynb
 create mode 100644 Actividad6.ipynb
 create mode 100644 Actividad6_Collab.pdf
```

```
!ls
```

```
Act6COLLAB.ipynb      Actividad5_A01068056.ipynb  Actividad6.ipynb  datasets
Act7_A01068056.ipynb  Actividad6_Collab.pdf       credentials.json  README.md
```

Actividad Visualización

- **Nombre: Daniel Gutiérrez Gómez**
- **Matrícula: A01068056**
- 03/22/23

```
import pandas as pd
import numpy as np
from scipy import stats
from scipy.stats import pearsonr
import matplotlib.pyplot as plt
import seaborn as sns
```

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.

Crea una tabla resumen con los estadísticas generales de las variables

```
amazon_books = pd.read_csv('./datasets/bestsellers with categories.csv')
display(amazon_books.iloc[: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
4	5,000 Awesome Facts (About Everything!) (Natio...	National Geographic Kids	4.8	7665	12	2019	Non Fiction
5	A Dance with Dragons (A Song of Ice and Fire)	George R. R. Martin	4.4	12643	11	2011	Fiction

¿Cuál es el género con más publicaciones? Muéstralo en un gráfico.

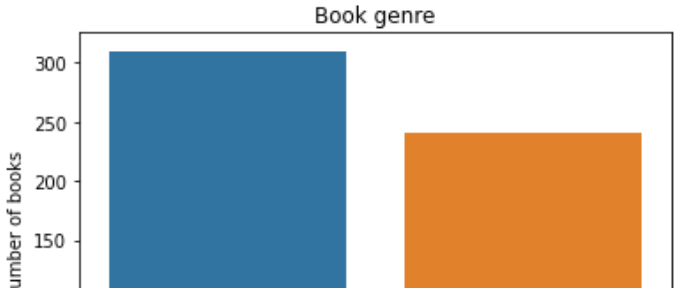
```
display(amazon_books.groupby([amazon_books["Genre"]]).agg(genres=("Genre","count")))
```

Genre	genres
Fiction	240
Non Fiction	310

Vemos que el género con más publicaciones del DataSet es el de no ficción, a continuación la gráfica

```
sns.countplot(data=amazon_books, x='Genre')
plt.title('Book genre')
plt.xlabel('Genres')
plt.ylabel('Number of books')
```

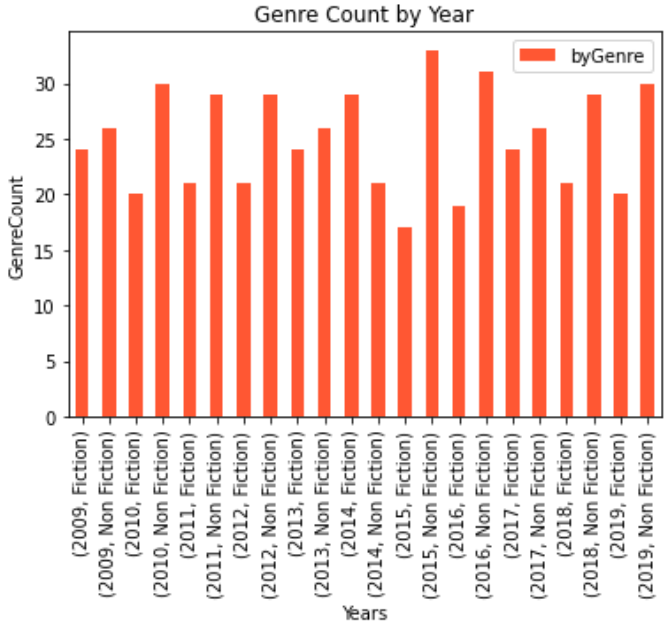
```
Text(0, 0.5, 'Number of books')
```



¿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.


```
N = 22
c = ['#FF5733', '#FF8D33', '#FFDC33', '#FFA533', '#FFD833', '#DFBA1F', '#CCA70C', '#CAA302',
     '#AF8E04', '#9B7F07', '#897003', '#745F03', '#A0B106', '#A0B106', '#909F07', '#818F00',
     '#798413', '#767E25', '#9AA43B', '#87C032', '#8AD21F', '#91EA0C']
amazon_books.groupby(["Year", "Genre"]).agg(byGenre=("Genre", "count")).plot(kind='bar', color=c)
plt.xlabel('Years')
plt.ylabel('GenreCount')
plt.title('Genre Count by Year')
```

```
Text(0.5, 1.0, 'Genre Count by Year')
```



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

```
sns.histplot(data=amazon_books, x='Reviews')
```

```
<Axes: xlabel='Reviews', ylabel='Count'>
```

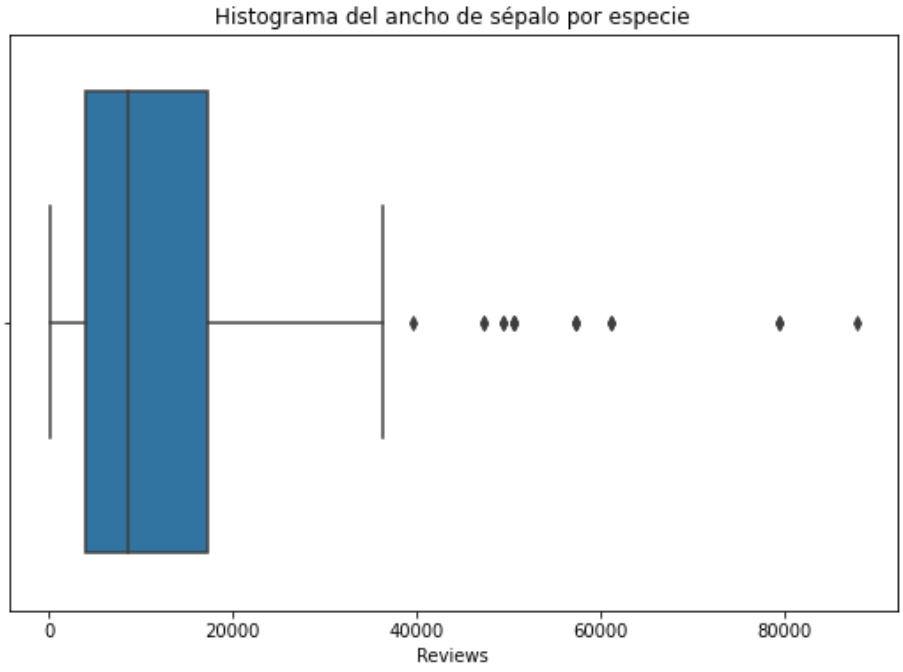
Ahora muéstralo en un gráfico de caja y bigote.


```
# Tamaño de la imagen
fig = plt.figure(figsize=(9, 6))

# Gráfico boxplot
sns.boxplot(data=amazon_books, x='Reviews')

# Ejes y título
plt.title('Histograma del ancho de sépalo por especie')
```

```
Text(0.5, 1.0, 'Histograma del ancho de sépalo por especie')
```



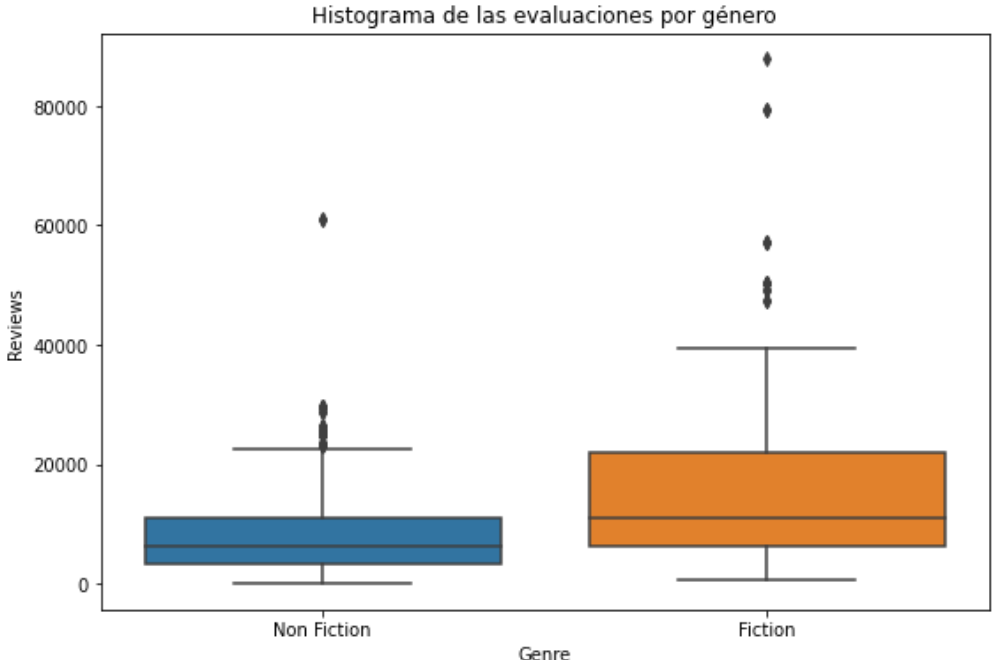
¿Cómo se compara la evaluación del libro por género? ¿Qué genero es mejor evaluado por los lectores? Muéstralo en un solo gráfico de caja y bigote.

```
# Tamaño de la imagen
fig = plt.figure(figsize=(9, 6))

# Gráfico boxplot
sns.boxplot(data=amazon_books, y='Reviews', x='Genre')

# Ejes y título
plt.title('Histograma de las evaluaciones por género')
```

```
Text(0.5, 1.0, 'Histograma de las evaluaciones por género')
```



Como vemos en el histograma de bigote y caja, el género con más evaluaciones es el de ficción, por otro lado, el que mejores evaluaciones tiene, puede que sea el de Non Fiction, puesot que tiene menos datos que están fuera de los rangos normales (outliers), por lo que pueden afectar a la información verídica

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

```
# Graficaremos la relación entre el número de reseñas y los precios de los libros.

# Tamaño de la imagen.
fig = plt.figure(figsize=(6, 4))

# Gráfico scatterplot.
sns.scatterplot(data=amazon_books, x ='Price', y='Reviews')

# Ejes y título. Colocamos la etiqueta correcta de acuerdo a la orientación.
plt.title('Relación entre el número de reseñas y los precios de los libros')
plt.xlabel('Precios de los libros')
plt.ylabel('Reseñas')
```

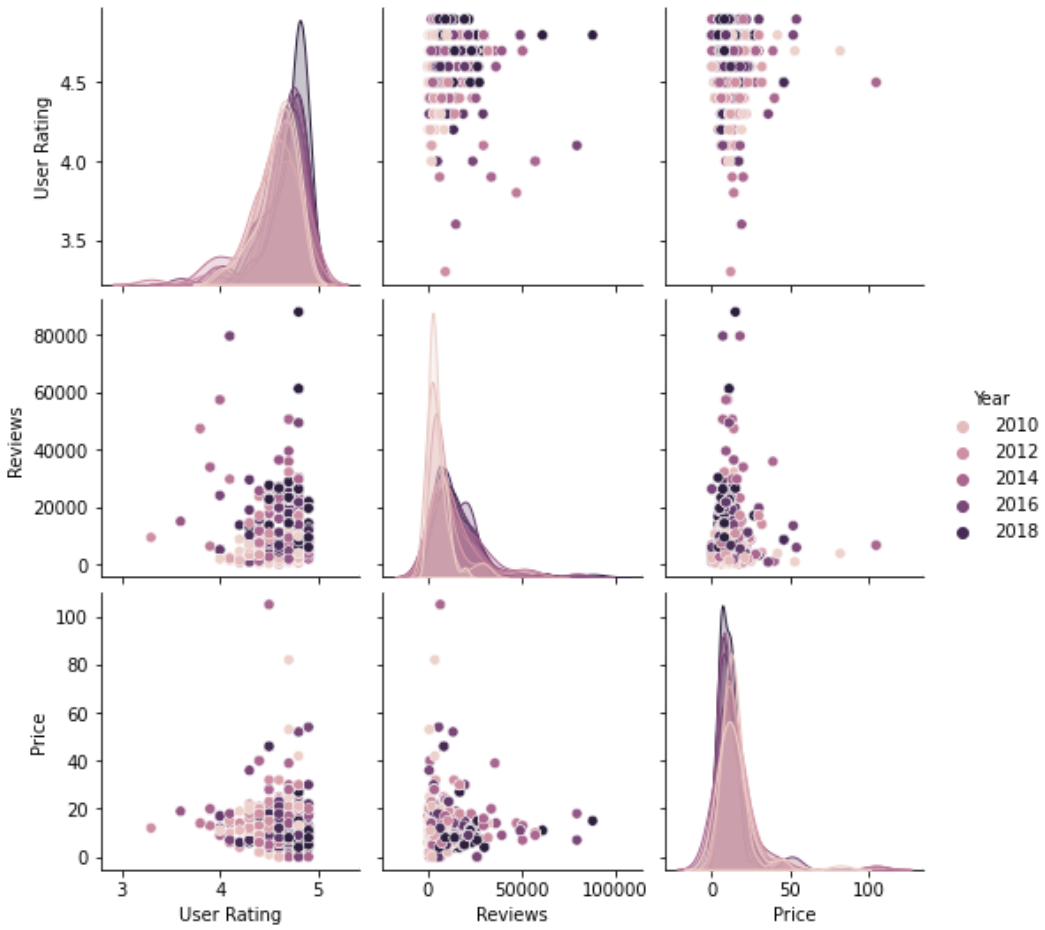
```
Text(0, 0.5, 'Reseñas')
```

Relación entre el número de reseñas y los precios de los libros

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.

```
sns.pairplot(data=amazon_books, hue="Year")
```

```
<seaborn.axisgrid.PairGrid at 0x7f091b14a8e0>
```



¿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.

```
correlation = pd.DataFrame().assign(prices=amazon_books["Price"], Reviews=amazon_books["Reviews"]
sns.heatmap(data=correlation, vmin=-1, vmax=1,  annot=True, square = True)
```

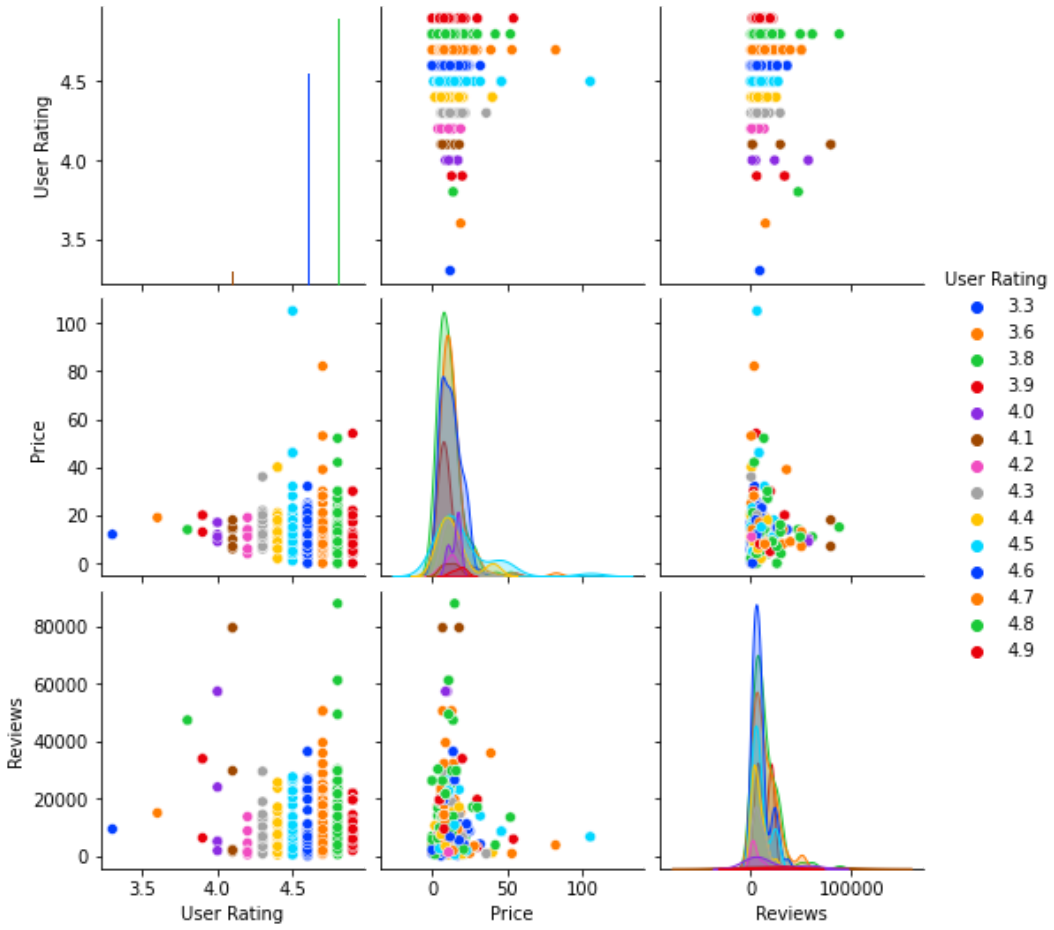
```
<Axes: >
```

¿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. Como vemos en el heatMap, la mayoría tiene un valor de correlación casi de 0, por lo que decimos que casi no hay correlación, por lo que no hay casi relación entre datos.

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(data=amazon_books,
             x_vars= ["User Rating", "Price", "Reviews"],
             y_vars=["User Rating", "Price", "Reviews"], hue="User Rating", palette="bright")
```

```
<seaborn.axisgrid.PairGrid at 0x7f0918176160>
```



✓ 6 s se ejecutó 10:35