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

```
Mounted at /gdrive
```

```
%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
```

```
%cd SemanaTecTC1002S/
```

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

```
!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 pull
```

```
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 5), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (9/9), 257.12 KiB | 1.12 MiB/s, done.
From https://github.com/DanyGuti/SemanaTecTC1002S
   c46a893..6235e3c  main       -> cursoFuente/main
```

```
Updating c46a893..6235e3c
Fast-forward
 Act8Collab.ipynb                  | 894 ++++++++++++++++++++++++++++++++++++
 Act8Collab.ipynb_Colaboratory.pdf | Bin 0 -> 269092 bytes
 2 files changed, 894 insertions(+)
 create mode 100644 Act8Collab.ipynb
 create mode 100644 Act8Collab.ipynb_Colaboratory.pdf
```

```
!git status
```

```
On branch main
Your branch is up to date with 'cursoFuente/main'.

nothing to commit, working tree clean
```

Actividad 9 - K means

- **Nombre: Daniel Gutiérrez Gómez**
- **Matrícula: A01068056**
- 03/24/23

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.

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns; sns.set_theme()
import numpy as np
from sklearn.linear_model import LinearRegression
```

```
amazon_books = pd.read_csv('./datasets/bestsellers with categories.csv')
```

```
display(amazon_books.iloc[:6])
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
	10 Day Green Smoothie						Non

Análisis estadístico

	12 Rules for Life: An	Jordan B.					Non

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

3	1984 (Signet Classics)	George Orwell	4.7	21424	6	2017	Fiction

```
amazon_books["Name"].count()
```

```
550
```

	A Dance with Dragons	George R. R.					

550 Datos

- Primer columna tiene el nombre del libro y es tipo "string"
- Segunda columna tiene el autor del libro y es tipo "string"
- Tercera columna tiene el rating que los usuarios le dan al libro y es tipo "int"
- Cuarta columna tiene el número de reseñas que ha recibido el libro y es tipo "int"
- Quinta columna tiene el precio del libro como tipo "int"
- Sexta columna tiene el año de pub. del libro como tipo "int"
- Séptima columna tiene el género del libro como tipo "string"

¿Qué hay más libros de ciencia ficción o no ficción?

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

Genre	byGenre
Fiction	240
Non Fiction	310

Rangos de máximo y mínimo de rating del usuario

```
display("máximo de user rating: ", amazon_books["User Rating"].max())
display("mínimo de user rating: ", amazon_books["User Rating"].min())
```

```
'máximo de user rating: '
4.9
'mínimo de user rating: '
3.3
```

Rangos de máximo y mínimo del número de reviews por libro

```
display("máximo de número de reviews: ", amazon_books["Reviews"].max())
display("mínimo de número de reviews: ", amazon_books["Reviews"].min())
```

```
'máximo de número de reviews: '
87841
'mínimo de número de reviews: '
37
```

Años en los que se encuentran guardados en la base de datos los libros

```
display("año más antiguo: ", amazon_books["Year"].min())
display("año más reciente: ", amazon_books["Year"].max())
```

```
'año más antiguo: '
2009
'año más reciente: '
2019
```

Rango de precios

```
display("precio más elevado: ", amazon_books["Price"].max())
display("precio más barato: ", amazon_books["Price"].min())
```

```
'precio más elevado: '
105
'precio más barato: '
0
```

Medias, mediana y desvest

```
display(amazon_books.groupby([amazon_books["Year"]])
.agg(bymeanPrice=("Price","mean"),
      bymeanRating=("User Rating", "mean"),
     bymeanReviews=("Reviews","mean")))
```

Year	bymeanPrice	bymeanRating	bymeanReviews
2009	15.40	4.584	4710.12

```
display("media de precio de libros", amazon_books["Price"].mean())
display("media de número de reviews", amazon_books["Reviews"].mean())
display("media de número de User Rating", amazon_books["User Rating"].mean())

print("\n")
display("mediana de precio de libros", amazon_books["Price"].median())
display("mediana de número de reviews", amazon_books["Reviews"].median())
display("mediana de número de User Rating", amazon_books["User Rating"].median())


print("\n")
display("varianza de precio de libros", amazon_books["Price"].var())
display("desvest de precio de libros", amazon_books["Price"].std())
display("varianza de número de reviews", amazon_books["Reviews"].var())
display("desvest de número de reviews", amazon_books["Reviews"].std())
display("varianza de número de User Rating", amazon_books["User Rating"].var())
display("desvest de número de User Rating", amazon_books["User Rating"].std())
```

```
'media de precio de libros'
13.1
'media de número de reviews'
11953.281818181818
'media de número de User Rating'
4.618363636363637


'mediana de precio de libros'
11.0
'mediana de número de reviews'
8580.0
'mediana de número de User Rating'
4.7


'varianza de precio de libros'
117.55464480874316
'desvest de precio de libros'
10.84226197842236
'varianza de número de reviews'
137619458.41041565
'desvest de número de reviews'
11731.132017431892
'varianza de número de User Rating'
0.05152008610697136
'desvest de número de User Rating'
0.2269803650251963
```

En base a la desviación estándar de 10.84 del precio de los libros, puedo decir que la variación entre los datos no es mucha, por otra parte, se puede decir que un libro podría costar entre 11 y 13 dolares como promedio con una varianza de 117.55

En base a la desviación estándar de 11,731.132 del número de reviews, puedo decir que la variación entre los datos es bastante, por otra parte, se puede decir que el número de reviews de un libro podría oscilar entre 8,580 y 11,731.132 como promedio con una varianza de 137619458.41

En base a la desviación estándar de 0.0515 del rating de los libros, puedo decir que la variación entre los datos no es casi nada, por otra parte, se puede decir que el rating de un libro podría oscilar entre 4.61 y 4.7 con una desviación estándar de 0.227, lo cual es bastante bueno

Puedo decir que lo que nos puede servir para medición y un análisis puede llegar a ser el rating de los libros, así como el costo, el número de reviews posiblemente no nos serviría de análisis

Correlación de las variables relevantes

```
print('Correlación Pearson (Price y User rating): ', amazon_books['Price'].corr(amazon_books['Use
print('Correlación spearman (Price y User rating): ', amazon_books['Price'].corr(amazon_books['Us
print('Correlación kendall (Price y User rating): ', amazon_books['Price'].corr(amazon_books['Use
```

```
Correlación Pearson (Price y User rating):  -0.13308628728087976
Correlación spearman (Price y User rating):  -0.23106979558156984
Correlación kendall (Price y User rating):  -0.169169022203182
```

```
print('Correlación Pearson (Reviews y User Rating): ', amazon_books['Reviews'].corr(amazon_books[
print('Correlación spearman (Reviews y User Rating): ', amazon_books['Reviews'].corr(amazon_books
print('Correlación kendall (Reviews y User Rating): ', amazon_books['Reviews'].corr(amazon_books[
```

```
Correlación Pearson (Reviews y User Rating):  -0.0017290142555501193
Correlación spearman (Reviews y User Rating):  0.20045803777248955
Correlación kendall (Reviews y User Rating):  0.14392769373560893
```

```
(Reviews y Price): ', amazon_books['Reviews'].corr(amazon_books['Price'], method='pearson'))
(Reviews y Price): ', amazon_books['Reviews'].corr(amazon_books['Price'], method='spearman'))
(Reviews y Price): ', amazon_books['Reviews'].corr(amazon_books['Price'], method='kendall'))
```

```
Correlación Pearson (Reviews y Price):  -0.10918188342780516
Correlación spearman (Reviews y Price):  -0.15026314273004007
Correlación kendall (Reviews y Price):  -0.09471660449212715
```

En base a los resultados generados, hay muy poca correlación entre las variables, las dos que más tienen son las del precio y el rating del usuario, lo cual era lo que había dicho en la parte de arriba, que las reviews no servirían de mucho. Confirmo que las relevantes son: el año, el precio del libro y el rating de los usuario de cada libro, el número de reviews puede o no tomarse, no afectaría mucho en el análisis predictivo de los libros de amazon.

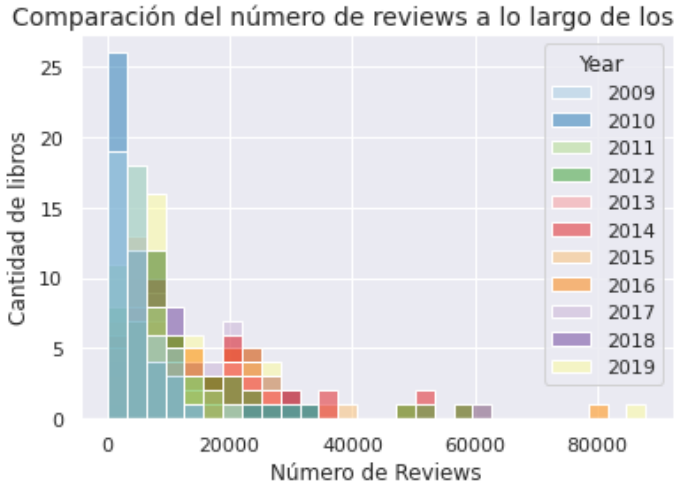
Análisis gráfico

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

Histograma del número de reviews

```
sns.histplot(data=amazon_books, x='Reviews', hue='Year', palette='Paired')
plt.xlabel('Número de Reviews')
plt.ylabel('Cantidad de libros')
plt.title('Comparación del número de reviews a lo largo de los años', fontsize=14)
```

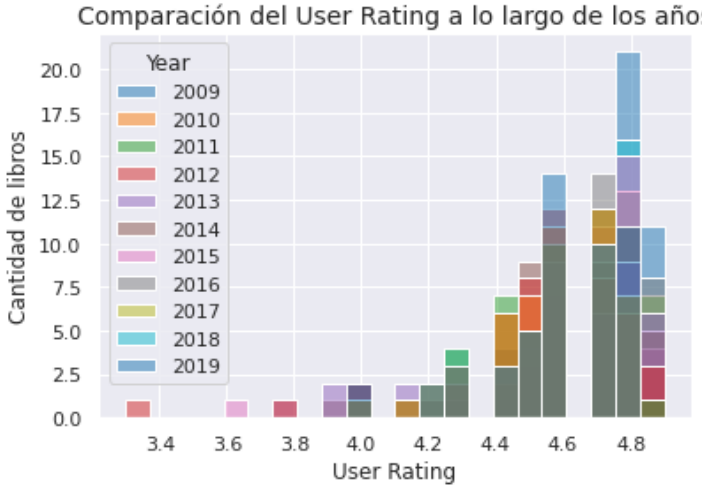
```
Text(0.5, 1.0, 'Comparación del número de reviews a lo largo de los años')
```



Histograma del rating del usuario

```
sns.histplot(data=amazon_books, x='User Rating', hue='Year', palette='tab10')
plt.xlabel('User Rating')
plt.ylabel('Cantidad de libros')
plt.title('Comparación del User Rating a lo largo de los años', fontsize=14)
```

```
Text(0.5, 1.0, 'Comparación del User Rating a lo largo de los años')
```



Histograma del precio del libro

```
sns.histplot(data=amazon_books, x='Price', hue='Year', palette='tab10')
plt.xlabel('Price')
plt.ylabel('Cantidad de libros')
plt.title('Comparación del User Rating a lo largo de los años', fontsize=14)
```

```
Text(0.5, 1.0, 'Comparación del User Rating a lo largo de los años')
```

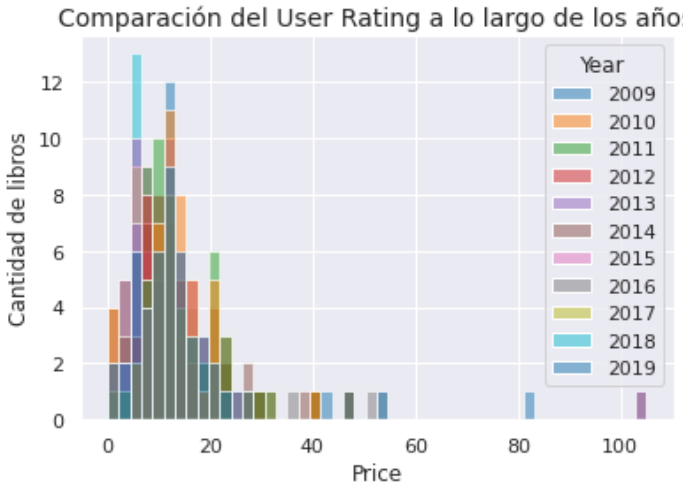


Gráfico de caja y bigote de reviews

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

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

# Ejes y título
plt.title('Caja y bigote del número de reviews', fontsize=14)
```

```
Text(0.5, 1.0, 'Histograma del número de reviews')
```

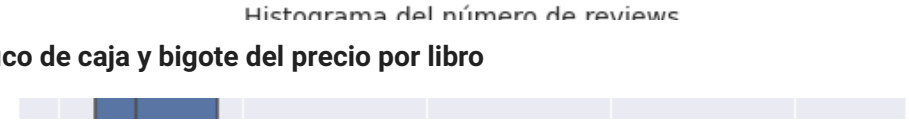


Gráfico de caja y bigote del precio por libro


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

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

# Ejes y título
plt.title('Caja y bigote del precio de libros', fontsize=14)
```

```
Text(0.5, 1.0, 'Caja y bigote del precio de libros')
```

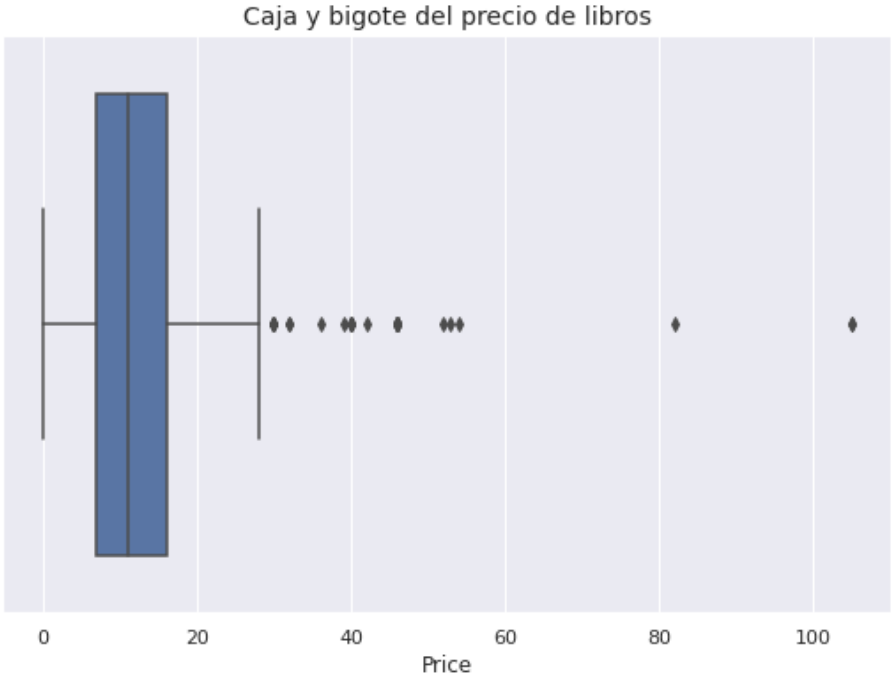


Gráfico de caja y bigote del User Rating

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

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

# Ejes y título
plt.title('Caja y bigote del rating de los libros', fontsize=14)
```

```
Text(0.5, 1.0, 'Caja y bigote del rating de los libros')
```

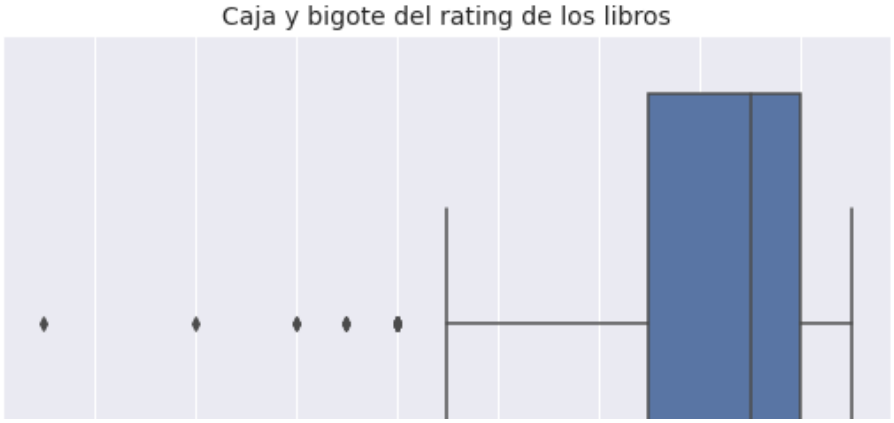


Gráfico de caja y bigote del número de reviews por género de libro

```
# 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('Caja y bigote del número de reviews por número de reviews', fontsize=14)
```

```
Text(0.5, 1.0, 'Caja y bigote del número de reviews por número de
reviews')
```

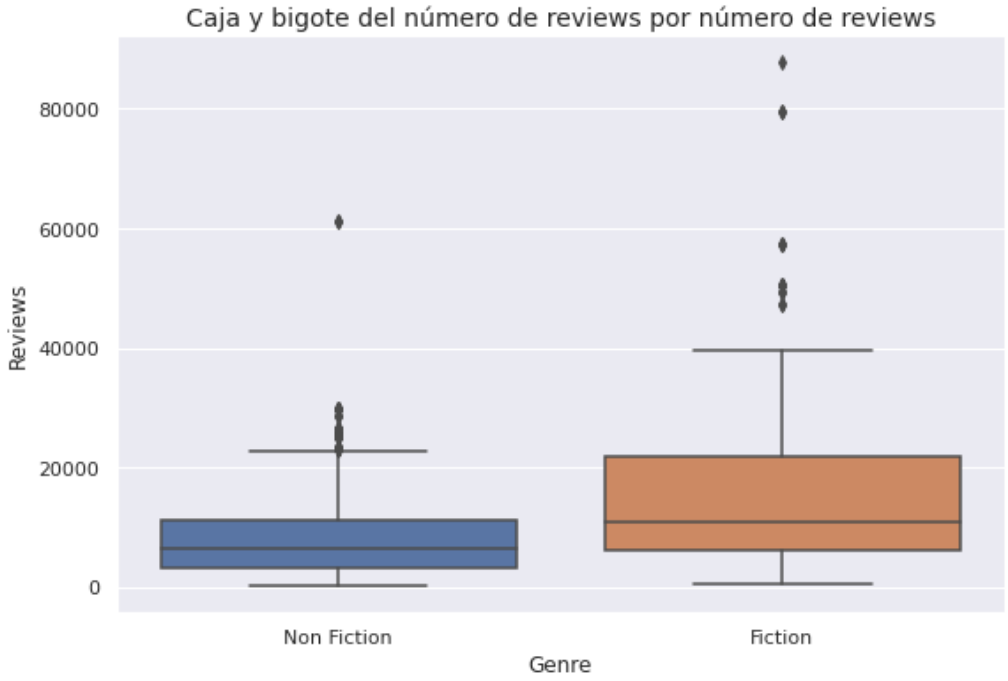


Gráfico de caja y bigote del precio por género de libro

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

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

```
# Ejes y título
plt.title('Caja y bigote del precio por género', fontsize=14)
```

```
Text(0.5, 1.0, 'Caja y bigote del precio por género')
```

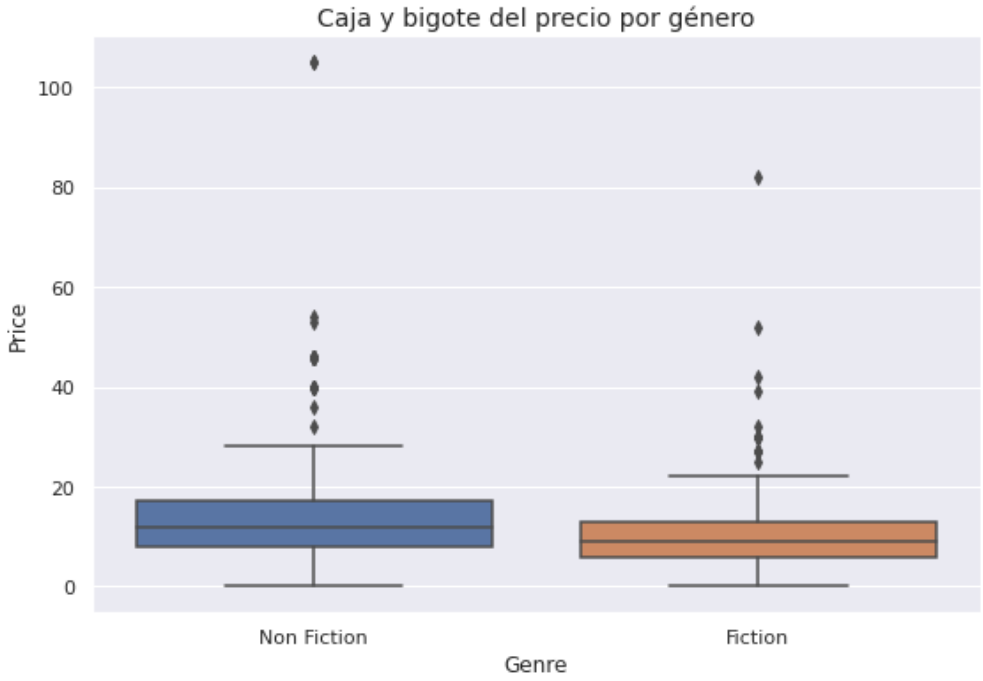
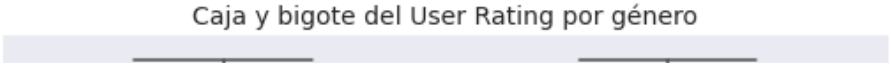


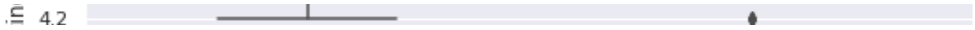
Gráfico de caja y bigote del User rating por género de libro

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

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

# Ejes y título
plt.title('Caja y bigote del User Rating por género', fontsize=14)
```

```
Text(0.5, 1.0, 'Caja y bigote del User Rating por género')
```

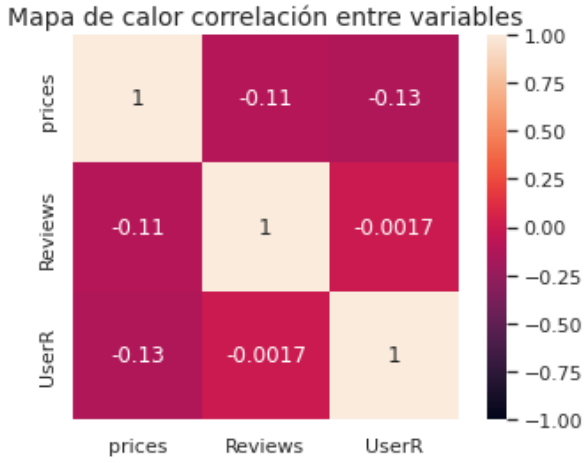


En base a los gráficos de caja y bigote generados, puedo decir que los datos que más nos puedan llegar a interesar es el rating del usuario y el precio del libro, puesto que los reviews van en mucha cantidad y es un número muy grande el que se encuentra en el cuartil de la media por lo que no lo usaremos, antes de eso, un heat map de la correlación entre las variables:




```
correlation = pd.DataFrame().assign(prices=amazon_books["Price"], Reviews=amazon_books["Reviews"]
sns.heatmap(data=correlation, vmin=-1, vmax=1,  annot=True, square = True)
plt.title('Mapa de calor correlación entre variables', fontsize=14)
```

```
Text(0.5, 1.0, 'Mapa de calor correlación entre variables')
```

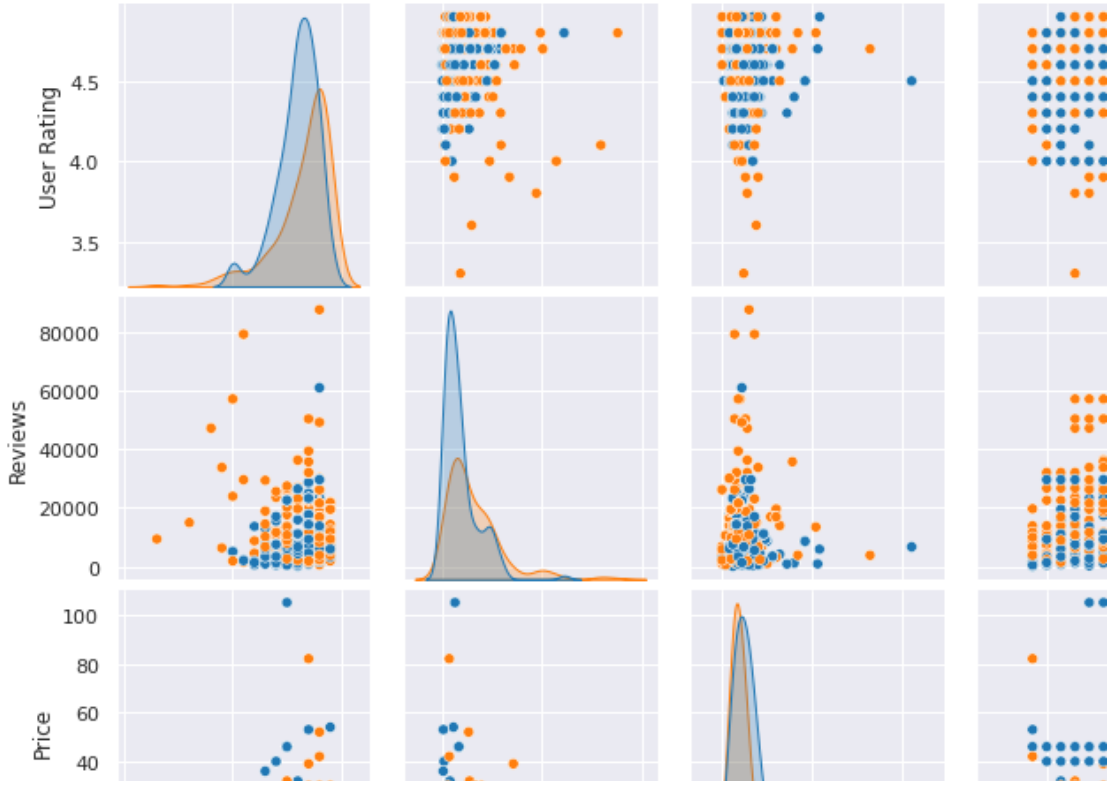


Como vemos la correlación entre las variables es muy baja la más alta siendo entre el rating del usuario y el precio de los libros... como dije anteriormente en el análisis estadístico.

A continuación gráficos de dispersión de todas las variables seguido de un gráfico de dispersión de las variables sin las reseñas de los usuarios

```
sns.pairplot(data=amazon_books, hue="Genre", palette='tab10')
plt.title('Gráficos de dispersión por género', fontsize=14)
```

```
Text(0.5, 1.0, 'Gráficos de dispersión por género')
```



Como podemos ver de manera general los libros de ciencia ficción podrían venderse más porque tienen un costo más barato en comparación con los de no ciencia ficción, además de que el número de reseñas por los de ficción es mayor que el número de reseñas que los de no ficción, afectando así en el rating del libro, siendo los de ciencia ficción los que tienen un mejor rating.

Ahora, gráficos de dispersión con variables que creemos son las mejores

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

```
<seaborn.axisgrid.PairGrid at 0x7ff28fd3d340>
```

De manera general no podemos decir que mientras más caro el libro mejor evaluado está por el cliente, sino que el precio de un buen libro (buen rating) oscilará entre los 10 a 20-22 doláres y que este precio depende realmente tanto de las reseñas, como del rating que tenga el libro

Clustering

1.- Normalización de variables


```
# Vamos a escalar las tres variables con StandardScaler, el cual se encuentra
# en SciKitLearn
from sklearn.preprocessing import StandardScaler
amazon_books = amazon_books.drop('Year', axis=1)

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

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

1. Elemento de la lista
2. Elemento de la lista

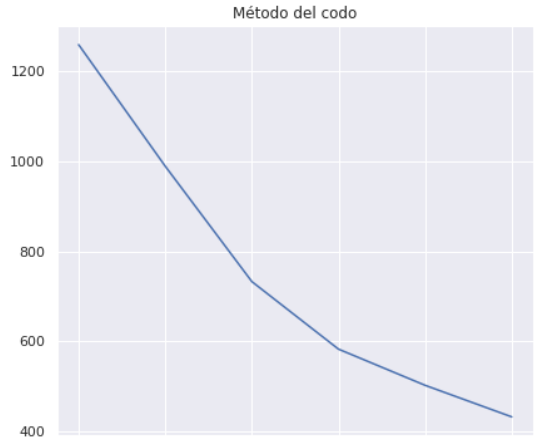
```
# # Importamos librerias en caso de no haberlo hecho antes
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Declaramos algunos arreglos. Los usuaremos para guardar los valores de la WCSS
 # y la silhouette score
kmax   = 8
grupos = range(2, kmax)
wcss   = []
```

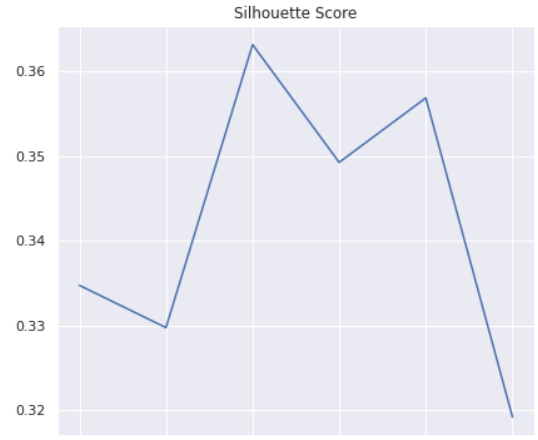
```
sil_score = []

# Ciclo para calcular K-Means para diferentes k
for k in grupos:
    # Clustering
    model = KMeans(n_clusters=k, random_state = 66)

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

    # Guardar WCSS (Within Cluster Sum of Squares)
    wcss.append(model.inertia_)

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

```
/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

/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

/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

/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

/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

/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
```

```
# 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')
```

```
Text(0.5, 1.0, 'Silhouette Score')
```





Vemos que el score de Silhueta nos dice que lo mejor es 4 grupos, por lo que recalcularemos kmeans para 4 grupos

```
# 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
amazon_books['Grupo'] = clusters.astype('str')
amazon_books.head()
```

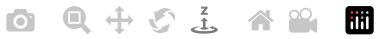
```
st-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

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

Author	User Rating	Reviews	Price	Genre	Grupo
JJ Smith	4.7	17350	8	Non Fiction	0
phen King	4.6	2052	22	Fiction	2
Jordan B. Peterson	4.7	18979	15	Non Fiction	0
rge Orwell	4.7	21424	6	Fiction	0
National eographic Kids	4.8	7665	12	Non Fiction	2

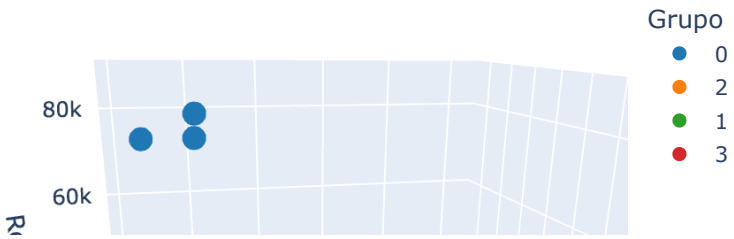
```
#Ahora el pairplot de dispersión de todas las variables con 4 grupos

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

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


```
# Vamos a graficar las 3 variables al mismo tiempo. Para ello, necesitamos
# importar una librería más
import plotly.express as px

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

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



4 grupos de clientes




```
# Veamos las características de cada grupo (i.e. los centros)
amazon_books.groupby('Grupo').mean()
```

Grupo	User Rating	Reviews	Price	Year
0	4.693846	27444.646154	9.084615	2015.307692
1	4.232143	8631.666667	12.416667	2012.857143
2	4.698065	6753.977419	11.900000	2013.858065
3	4.538462	7219.538462	49.692308	2012.846154

```
# También veamos las dispersiones
amazon_books.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

- ¿Crees que estos centros puedan ser representativos de los datos? ¿Por qué? Si lo son, puesto que con la técnica del codo y la silhueta busqué el valor donde se intersecta.
- ¿Cómo obtuviste el valor de a usar? Con la silhueta y la técnica del codo y el algoritmo de Kmeans
- ¿Los centros serían más representativos si usaras un valor más alto? ¿Más bajo? Con un valor más alto y más bajo, los grupos no serían representativos, por lo que no se podrían describir y analizar tan bien que con 4 grupos
- ¿Qué pasaría con los centros si tuviéramos muchos outliers en el análisis de cajas y bigotes? Los centros estarían muy dispersos entre sí y el análisis no podría ser del todo bueno.
- ¿Qué puedes decir de los datos basándose en los centros? Que el valor de los grupos a utilizar es de 4 y que aunque existen outliers, no es tanta la cantidad de outliers.

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

Utilicé un valor de clusters de 4, puesto que fue el resultado que nos arrojó el algoritmo de Kmeans, con un valor de 4 clusters (centroides)

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

1. Grupo 0: Rating del usuario medio alto (4-4.5) con alto número de reseñas y un costo aceptable
2. Grupo 1: Rating del usuario bajo (3.7-4) con bajo número de reseñas con un costo aceptable
3. Grupo 2: Rating del usuario alto (4.5-4.9) con bajo número de reseñas con un costo bajo
4. Grupo 3: Rating del usuario alto (4.5-4.9) con alto número de reseñas y un costo alto

✓ 0 s se ejecutó 13:27