SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

Course Code: DJ19DSL306 Date: 28/01/2022

Course: Programming with Python Laboratory Semester: III

Miniproject

CO/LO:

CO5. Apply various advance modules of Python for data analysis.

Kresha Shah – 60009220080
Heta Shah – 60009220081
Durva Patel – 60009220088

Problem Statement:
Perform Analysis on Amazon Best Selling Books
Code and output:

```
[1]
import numpy as np
import pandas as pd

from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import plotly.express as px
import plotly.graph_objs as go
import plotly.offline as pyo
from plotly.subplots import make_subplots


# Special Visualization
import wordcloud, missingno
from wordcloud import WordCloud # wordcloud
import missingno as msno # check missing value
import networkx as nx
```

```
[5] df=pd.read_csv("/content/drive/MyDrive/Dataset/bestsellers with categories.csv")
    df.head()
```

	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

```
[6]  df.isnull().sum()
```

```
Name           0
Author         0
User Rating    0
Reviews        0
Price          0
Year           0
Genre          0
dtype: int64
```

```
[7]
    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
```

```
[8]  df.shape
```

```
(550, 7)
```

```
[9]  df.sort_values('Reviews',ascending=False).head(10)
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
534	Where the Crawdads Sing	Delia Owens	4.8	87841	15	2019	Fiction
382	The Girl on the Train	Paula Hawkins	4.1	79446	18	2015	Fiction
383	The Girl on the Train	Paula Hawkins	4.1	79446	7	2016	Fiction
32	Becoming	Michelle Obama	4.8	61133	11	2018	Non Fiction
33	Becoming	Michelle Obama	4.8	61133	11	2019	Non Fiction
137	Gone Girl	Gillian Flynn	4.0	57271	9	2014	Fiction
135	Gone Girl	Gillian Flynn	4.0	57271	10	2012	Fiction
136	Gone Girl	Gillian Flynn	4.0	57271	10	2013	Fiction
368	The Fault in Our Stars	John Green	4.7	50482	13	2014	Fiction
367	The Fault in Our Stars	John Green	4.7	50482	7	2014	Fiction

[10]

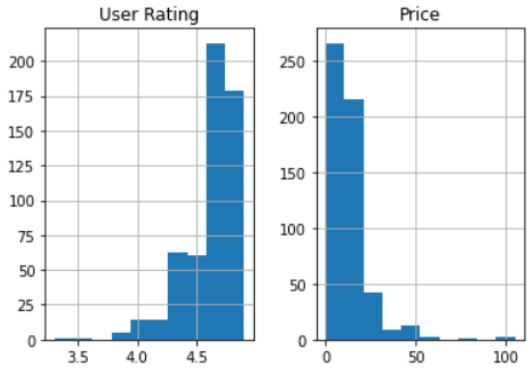
```
#there are 351 number of unique books
len(df.Name.value_counts())
```

```
351
```

[11]

```
df[["User Rating", "Price"]].hist()
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f13331406a0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1333113af0>]],
      dtype=object)
```



0s completed at 1:19 PM

[12]

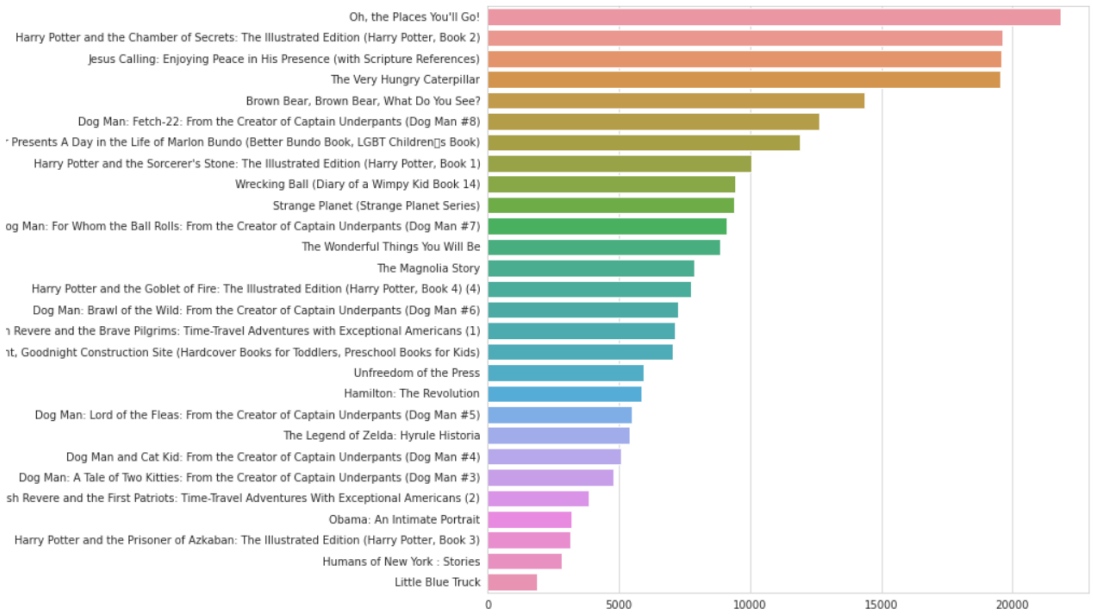
```
x=df[df["User Rating"]==4.9]
x.head(10)
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
40	Brown Bear, Brown Bear, What Do You See?	Bill Martin Jr.	4.9	14344	5	2017	Fiction
41	Brown Bear, Brown Bear, What Do You See?	Bill Martin Jr.	4.9	14344	5	2019	Fiction
81	Dog Man and Cat Kid: From the Creator of Capta...	Dav Pilkey	4.9	5062	6	2018	Fiction
82	Dog Man: A Tale of Two Kitties: From the Creat...	Dav Pilkey	4.9	4786	8	2017	Fiction
83	Dog Man: Brawl of the Wild: From the Creator o...	Dav Pilkey	4.9	7235	4	2018	Fiction
84	Dog Man: Brawl of the Wild: From the Creator o...	Dav Pilkey	4.9	7235	4	2019	Fiction
85	Dog Man: Fetch-22: From the Creator of Captain...	Dav Pilkey	4.9	12619	8	2019	Fiction
86	Dog Man: For Whom the Ball Rolls: From the Cre...	Dav Pilkey	4.9	9089	8	2019	Fiction
87	Dog Man: Lord of the Fleas: From the Creator o...	Dav Pilkey	4.9	5470	6	2018	Fiction
146	Goodnight, Goodnight Construction Site (Hardco...	Sherri Duskey Rinker	4.9	7038	7	2012	Fiction

[13]

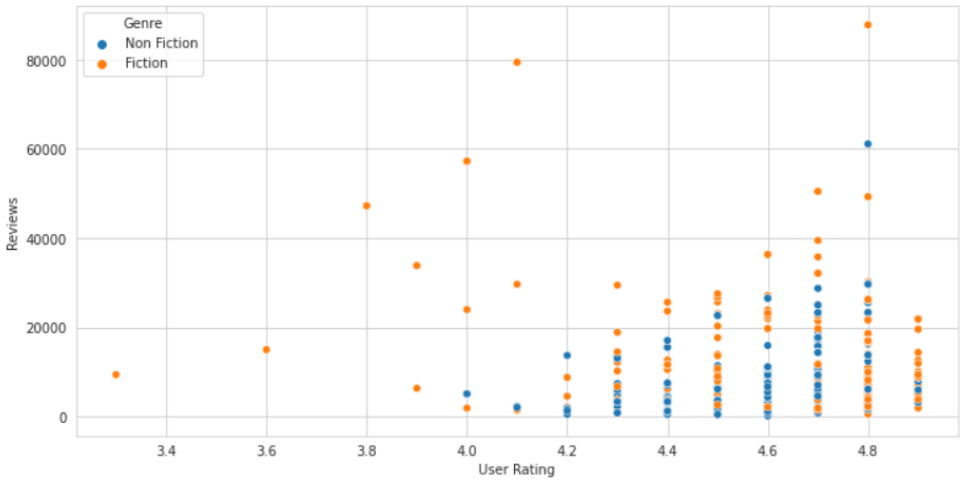
```
y= x.groupby("Name").Reviews.mean().sort_values(ascending= False)
plt.figure(figsize= (10,10))
sns.set_style("whitegrid")
sns.barplot(y.values,y.index)
```

[13]



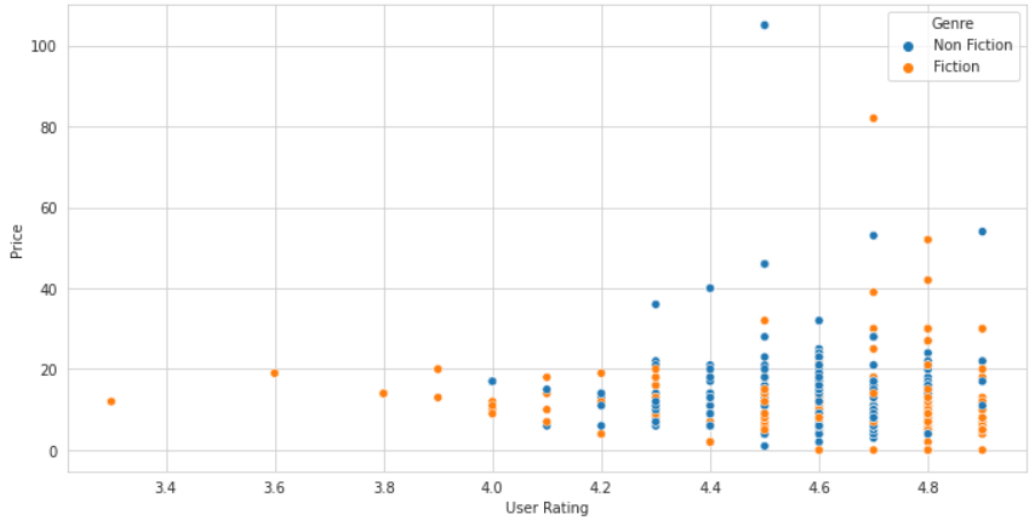
[14]
```
#user ratings and reviews relation
plt.figure(figsize= (12,6))
sns.scatterplot(x="User Rating",y= "Reviews",data= df,hue= "Genre")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f13330eb460>
```



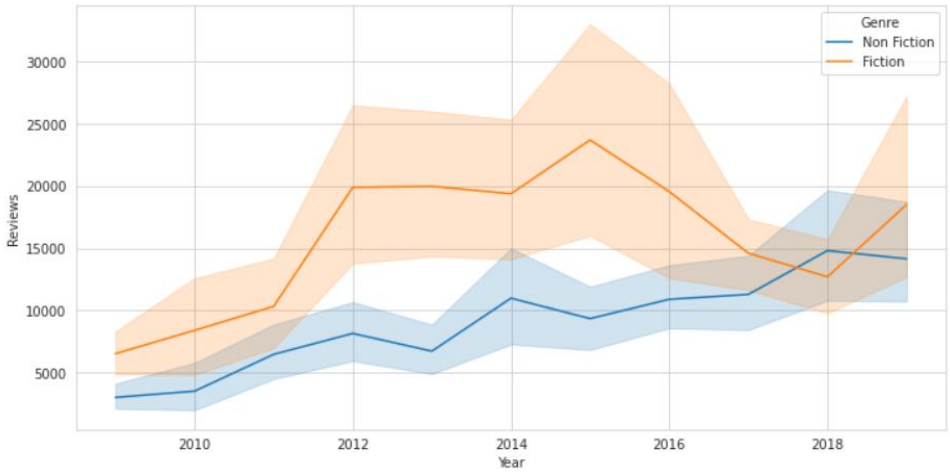
```
[15] plt.figure(figsize= (12,6))
     sns.scatterplot(x="User Rating",y= "Price",data= df,hue= "Genre")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1333032d90>
```



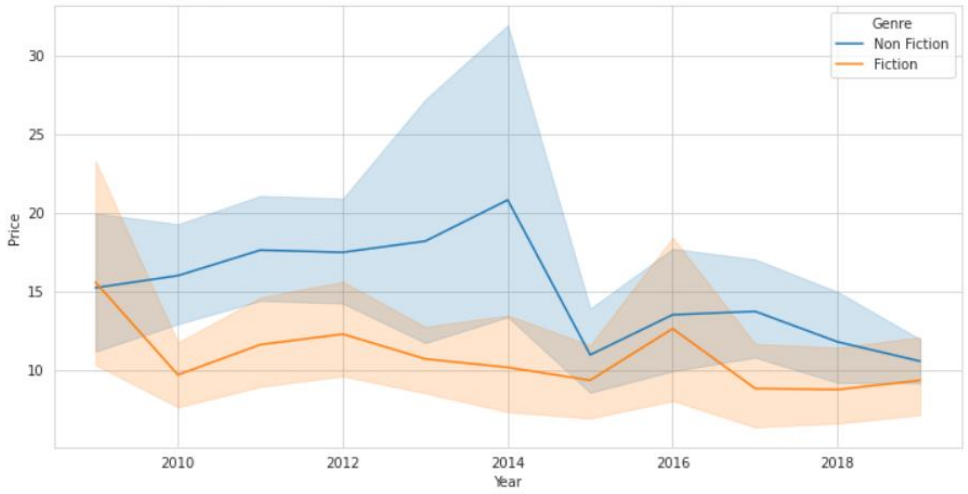

```
[16] plt.figure(figsize= (12,6))
     sns.lineplot(x= "Year",y= "Reviews",data= df,hue="Genre")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f133116ac70>
```



```
[17] plt.figure(figsize= (12,6))
     sns.lineplot(x= "Year",y= "Price",data= df,hue="Genre")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1331152ee0>
```

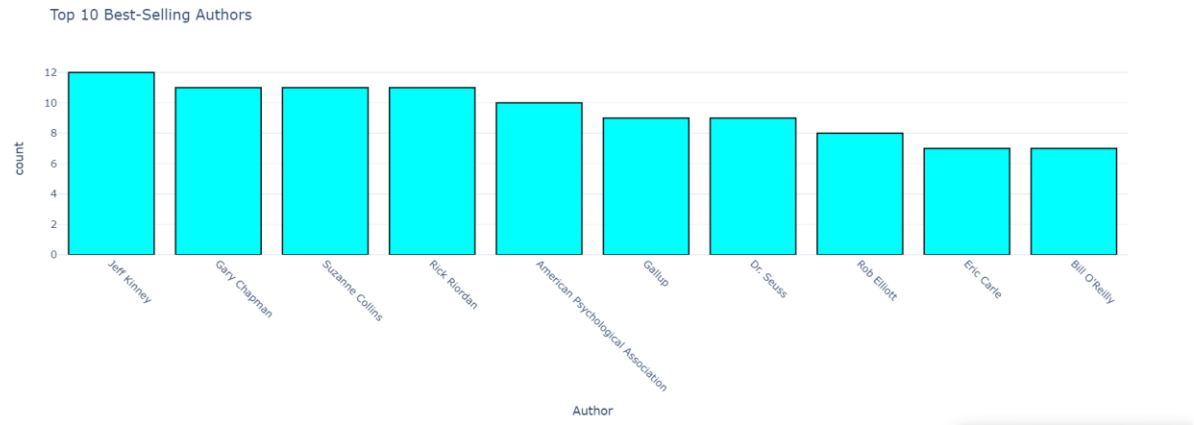



```
[18] df['Author'].value_counts()
```

```
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
```

```
[19]
     #BESTSELLING BOOKS' AUTHORS:
     temp_df1 = df.groupby('Author').count().reset_index().sort_values('Name',ascending=False).head(10)
```

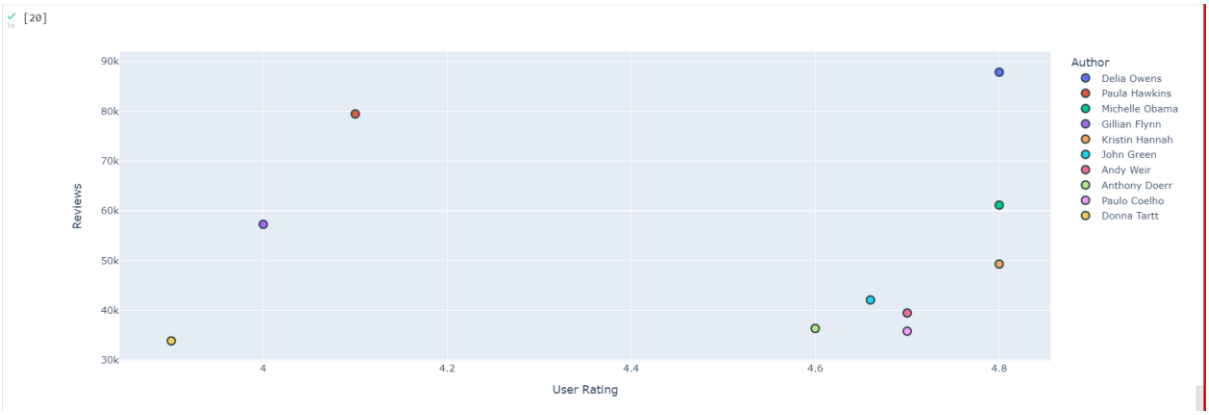
```
[ ]
     top = go.Bar(
                    x = temp_df1['Author'],
                    y = temp_df1['Name'],
                    marker = dict(color = 'cyan',
                                  line=dict(color='rgb(0,0,0)',width=1.5)))
     layout = go.Layout(template= "plotly_white",title = 'Top 10 Best-Selling Authors ' ,
                        xaxis = dict(title = 'Author',tickangle=45), yaxis = dict(title = 'count'))
     fig = go.Figure(data = [top], layout = layout)
     fig.show()
```




```
[20] df1 = df.groupby('Author').mean().sort_values('Reviews',ascending=False).reset_index().head(10)

     fig = px.scatter(df1, x='User Rating', y='Reviews', color='Author')

     fig.update_traces(marker=dict(size=10,
                                   line=dict(width=2,
                                             color='DarkSlateGrey')),
                       selector=dict(mode='markers'))
     fig.show()
```




```
[21] df.head()
```

	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

```
[22] Fiction = df[df["Genre"]=='Fiction']
     Fiction.head(10)
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
1	11/22/63: A Novel	Stephen King	4.6	2052	22	2011	Fiction
3	1984 (Signet Classics)	George Orwell	4.7	21424	6	2017	Fiction
5	A Dance with Dragons (A Song of Ice and Fire)	George R. R. Martin	4.4	12643	11	2011	Fiction
6	A Game of Thrones / A Clash of Kings / A Storm...	George R. R. Martin	4.7	19735	30	2014	Fiction
7	A Gentleman in Moscow: A Novel	Amor Towles	4.7	19699	15	2017	Fiction
9	A Man Called Ove: A Novel	Fredrik Backman	4.6	23848	8	2016	Fiction
10	A Man Called Ove: A Novel	Fredrik Backman	4.6	23848	8	2017	Fiction
13	A Wrinkle in Time (Time Quintet)	Madeleine L'Engle	4.5	5153	5	2018	Fiction
20	All the Light We Cannot See	Anthony Doerr	4.6	36348	14	2014	Fiction
21	All the Light We Cannot See	Anthony Doerr	4.6	36348	14	2015	Fiction

```
[23] Non_Fiction = df[df["Genre"]=='Non Fiction']
     Non_Fiction.head(10)
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
0	10-Day Green Smoothie Cleanse	JJ Smith	4.7	17350	8	2016	Non Fiction
2	12 Rules for Life: An Antidote to Chaos	Jordan B. Peterson	4.7	18979	15	2018	Non Fiction
4	5,000 Awesome Facts (About Everything!) (Natio...	National Geographic Kids	4.8	7665	12	2019	Non Fiction
8	A Higher Loyalty: Truth, Lies, and Leadership	James Comey	4.7	5983	3	2018	Non Fiction
11	A Patriot's History of the United States: From...	Larry Schweikart	4.6	460	2	2010	Non Fiction
12	A Stolen Life: A Memoir	Jaycee Dugard	4.6	4149	32	2011	Non Fiction
14	Act Like a Lady, Think Like a Man: What Men Re...	Steve Harvey	4.6	5013	17	2009	Non Fiction
15	Adult Coloring Book Designs: Stress Relief Col...	Adult Coloring Book Designs	4.5	2313	4	2016	Non Fiction
16	Adult Coloring Book: Stress Relieving Animal D...	Blue Star Coloring	4.6	2925	6	2015	Non Fiction
17	Adult Coloring Book: Stress Relieving Patterns	Blue Star Coloring	4.4	2951	6	2015	Non Fiction

Code:

```
import numpy as np
import pandas as pd

from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
import seaborn as sns


import plotly.express as px
import plotly.graph_objs as go
import plotly.offline as pyo
from plotly.subplots import make_subplots


# Special Visualization
import wordcloud, missingno
from wordcloud import WordCloud # wordcloud
import missingno as msno # check missing value
```

```
import networkx as nx
df.isnull().sum()



df.info()
df.shape
df.sort_values('Reviews',ascending=False).head(10)
len(df.Name.value_counts())
df[["User Rating", "Price"]].hist()
x=df[df["User Rating"]==4.9]
x.head(10)


y= x.groupby("Name").Reviews.mean().sort_values(ascending= False)
plt.figure(figsize= (10,10))
sns.set_style("whitegrid")
sns.barplot(y.values,y.index)


plt.figure(figsize= (12,6))
sns.scatterplot(x="User Rating",y= "Reviews",data= df,hue= "Genre")


plt.figure(figsize= (12,6))
sns.scatterplot(x="User Rating",y= "Price",data= df,hue= "Genre")


plt.figure(figsize= (12,6))
sns.lineplot(x= "Year",y= "Reviews",data= df,hue="Genre")


plt.figure(figsize= (12,6))
sns.lineplot(x= "Year",y= "Price",data= df,hue="Genre")


df['Author'].value_counts()


temp_df1 = df.groupby('Author').count().reset_index().sort_values('Name
',ascending=False).head(10)


top = go.Bar(
                x = temp_df1['Author'],
                y = temp_df1['Name'],
                marker = dict(color = 'cyan',
                              line=dict(color='rgb(0,0,0)',width=1.5)))
layout = go.Layout(template= "plotly_white",title = 'Top 10 Best-
Selling Authors ' ,
                   xaxis = dict(title = 'Author',tickangle=45), yaxis =
 dict(title = 'count'))
fig = go.Figure(data = [top], layout = layout)
```

```
fig.show()


df1 = df.groupby('Author').mean().sort_values('Reviews',ascending=False
).reset_index().head(10)

fig = px.scatter(df1, x='User Rating', y='Reviews', color='Author')

fig.update_traces(marker=dict(size=10,
                              line=dict(width=2,
                                        color='DarkSlateGrey')),
                  selector=dict(mode='markers'))
fig.show()


df.head()


Fiction = df[df["Genre"]=='Fiction']
Fiction.head(10)


Non_Fiction = df[df["Genre"]=='Non Fiction']
Non_Fiction.head(10)
```