26. EDA - amzn books data

October 31, 2022

1 Exploratory Data Analysis (EDA)

- Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Dataset on Amazon's Top 50 bestselling books from 2009 to 2019. Contains 550 books, data has been categorized into fiction and non-fiction

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_csv("../data/bestsellers with categories.csv")
```

2 Observe data in the dataframe

[3]:
```
df.head()
```

[3]:

	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

3 Number of rows and column

[4]:
```
print("Shape of Dataset")
print(df.shape)
```

```
Shape of Dataset
(550, 7)
```

4 Unique elements in columns

[5]:
```
print("Unique elements in Features")
df.nunique()
```

```
Unique elements in Features
```

[5]:
```
Name           351
Author         248
User Rating     14
Reviews        346
Price           40
Year            11
Genre            2
dtype: int64
```

5 Duplicated Rows

[6]:
```
print("Duplicated Series values")
print(df.duplicated().sum())
```

```
Duplicated Series values
0
```

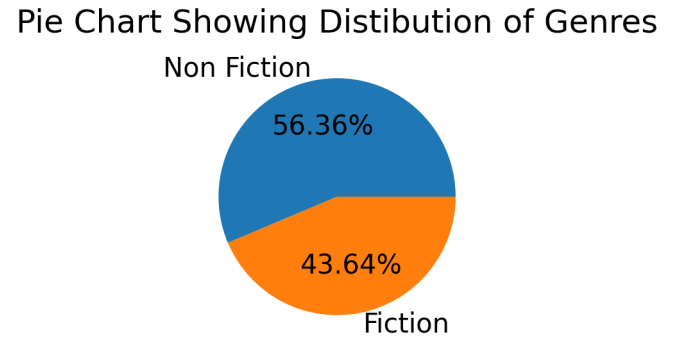
6 Genres Feature

[7]:
```
df['Genre'].value_counts()
```

[7]:
```
Non Fiction    310
Fiction        240
Name: Genre, dtype: int64
```

[8]:
```
fig = plt.figure(figsize=(2,2), dpi=300, facecolor="w")
genres = df['Genre'].value_counts()
plt.pie(genres, labels=genres.index, autopct="%.2f%%")
plt.title("Pie Chart Showing Distibution of Genres")
plt.savefig("genres_pie.png", dpi=300, bbox_inches="tight")
```

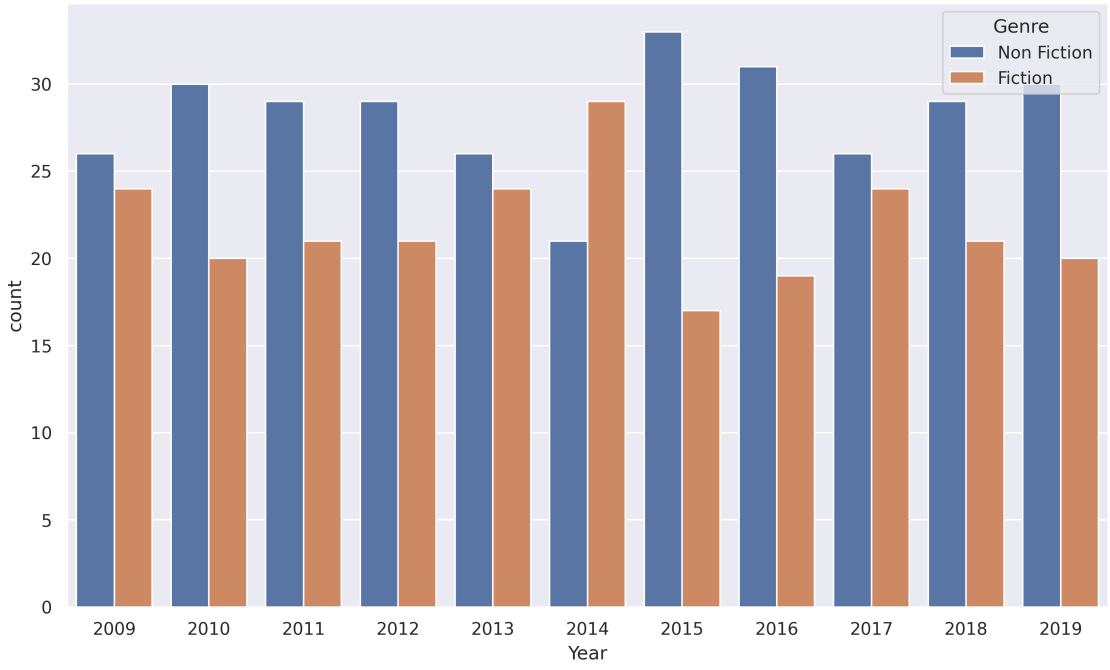
```
plt.show()
```



- Observation: Almost 56% rated as best selling books are Non Fiction

[9]:
```
sns.set_theme(style="darkgrid")
```

[10]:
```
# Below Countplot shows the number of books(Count) that were fiction vs non␣
↪fiction among the best sellers over the years.
plt.figure(figsize=(12,7),dpi = 300)
sns.countplot(x=df['Year'],hue=df['Genre'])
plt.show()
```



- Observations: For all the years except 2014, the number of fiction best sellers have been greater than non fiction best sellers books.

7 User Rating

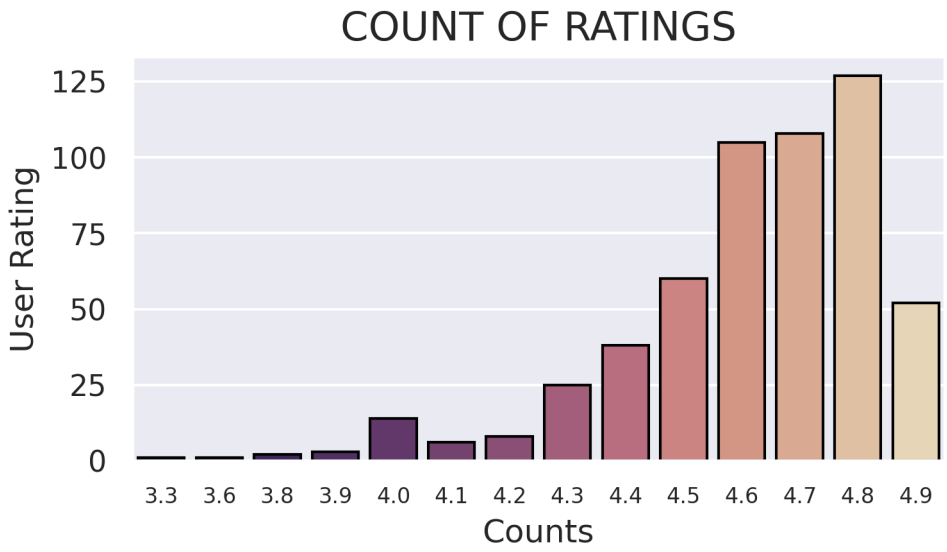
[11]:
```
print("Max User Rating")
print(df['User Rating'].max())
print()
print("Avg User Rating")
print(df['User Rating'].mean())
print()
print("Most Often User Rating")
print(df['User Rating'].mode())
print()
```

```
Max User Rating
4.9

Avg User Rating
4.618363636363637

Most Often User Rating
0    4.8
Name: User Rating, dtype: float64
```

[12]:
```
plt.figure(figsize=(12,6), dpi=300)
# plt.style.use("seaborn")
# plt.figure(figsize=(20,20))
plt.subplot(221)
fund= sns.countplot(x=df["User Rating"],␣
 ↪palette="magma",edgecolor='black',saturation=0.50)
fund.set_xticklabels(fund.get_xticklabels(),fontsize=8)
plt.title("COUNT OF RATINGS",fontsize=15)
fund.set_xlabel("Counts", fontsize=12)
fund.set_ylabel("User Rating", fontsize=12)
plt.show()
```



8 Authors

[13]:
```
df['Author'].value_counts() # How many books each author have written (acc to this dataset)
```

[13]:
```
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
```

[14]:
```
#Author's Books having rating: 4.9
maxrating = df[df['User Rating']==4.9]
aumax = maxrating.groupby(['Author']).size().reset_index(name="Count")
aumax.sort_values(by='Count',ascending=False).head(20)
```

[14]:

	Author	Count
5	Dr. Seuss	8
4	Dav Pilkey	7
7	Eric Carle	7

```
18            Sarah Young      6
6   Emily Winfield Martin      4
9             J.K. Rowling     3
19   Sherri Duskey Rinker      2
17          Rush Limbaugh      2
1         Bill Martin Jr.      2
13          Mark R. Levin      1
16             Pete Souza      1
15         Patrick Thorpe      1
14         Nathan W. Pyle      1
0          Alice Schertle      1
12     Lin-Manuel Miranda      1
11              Jill Twiss     1
8            J. K. Rowling     1
3              Chip Gaines     1
2         Brandon Stanton      1
10             Jeff Kinney     1
```

[15]:
```
maxrating.groupby(['Author']).size()
```

[15]:
```
Author
Alice Schertle           1
Bill Martin Jr.          2
Brandon Stanton          1
Chip Gaines              1
Dav Pilkey               7
Dr. Seuss                8
Emily Winfield Martin    4
Eric Carle               7
J. K. Rowling            1
J.K. Rowling             3
Jeff Kinney              1
Jill Twiss               1
Lin-Manuel Miranda       1
Mark R. Levin            1
Nathan W. Pyle           1
Patrick Thorpe           1
Pete Souza               1
Rush Limbaugh            2
Sarah Young              6
Sherri Duskey Rinker     2
dtype: int64
```

[16]:
```
#'Where the Crawdads sing' Book of Delia Owens has maximum user reviews (87841).
df[df['Reviews']==df['Reviews'].max()]
```

[16]:
```
                         Name       Author  User Rating  Reviews  Price  Year  \
534  Where the Crawdads Sing  Delia Owens          4.8    87841     15  2019

       Genre
534  Fiction
```

[17]:
```
maxrating[maxrating['Reviews']==maxrating['Reviews'].max()]
```

[17]:
```
                          Name     Author  User Rating  Reviews  Price  Year  \
245  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2012
246  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2013
247  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2014
248  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2015
249  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2016
250  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2017
251  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2018
252  Oh, the Places You'll Go!  Dr. Seuss          4.9    21834      8  2019

       Genre
245  Fiction
246  Fiction
247  Fiction
248  Fiction
249  Fiction
250  Fiction
251  Fiction
252  Fiction
```

9 Price

[18]:
```
#Most of books having rating 4.9 have price 8
plt.figure(figsize=(12,6),dpi=300)
sns.histplot(maxrating['Price'])
plt.title('Price Distribution Plot',fontsize=20)
plt.show()
maxrating['Price'].mode()
```

[18]:
```
0    8
Name: Price, dtype: int64
```

[19]:
```
df['Price'].max()
```

[19]:
```
105
```

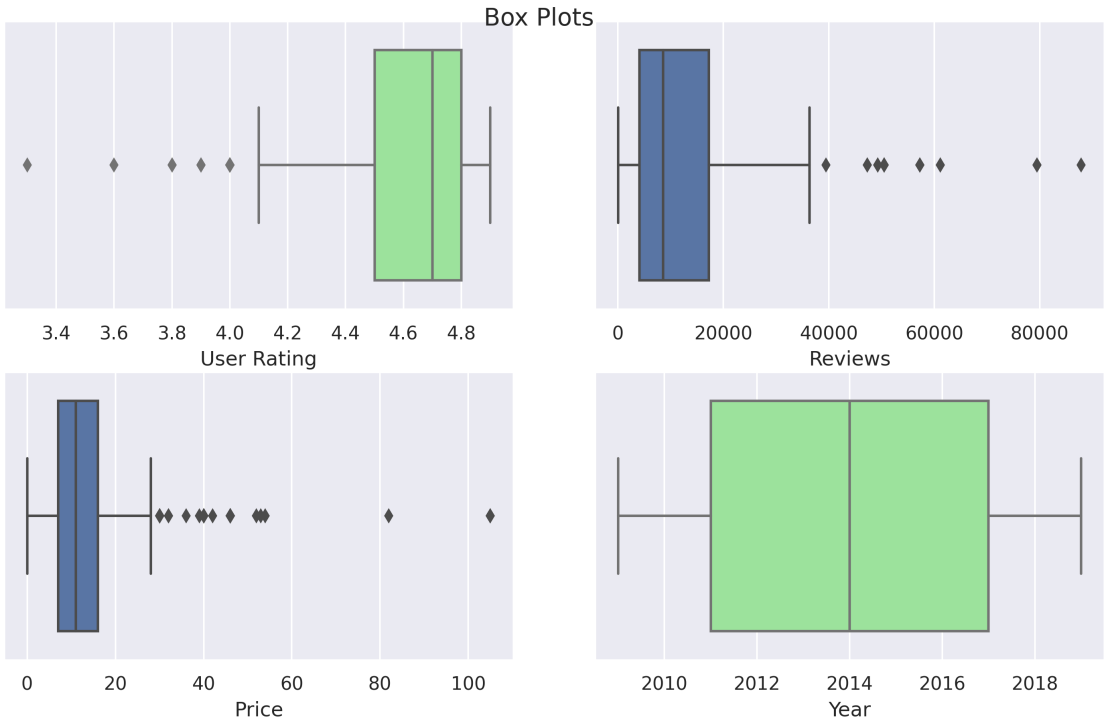
10 General Trend & Outlier

[20]:
```
fig, axes = plt.subplots(2, 2,figsize=(10,6),dpi=300)
fig.tight_layout()
fig.suptitle('Box Plots')
sns.boxplot(x=df["User Rating"], ax=axes[0,0],color="lightgreen")

sns.boxplot(x=df["Reviews"],ax=axes[0,1])

sns.boxplot(x=df["Price"],ax=axes[1,0])
sns.boxplot(x=df["Year"],ax=axes[1,1], color="lightgreen")

plt.show()
```



- When reviewing a box plot, an outlier is defined as a data point that is located outside the whiskers of the box plot. For example, outside 1.5 times the interquartile range above the upper quartile and below the lower quartile (Q1 - 1.5 * IQR or Q3 + 1.5 * IQR).