**PRUDENT.AI**

**TASK – 01**

**Exploratory Data Analysis:**

**Importing packages:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import os

import warnings

!pip install chart\_studio

from plotly.offline import init\_notebook\_mode, iplot

init\_notebook\_mode(connected=True)

import plotly.graph\_objs as go

import plotly.figure\_factory as ff

from plotly import tools

import chart\_studio.plotly as py

**Importing dataset:**

from google.colab import files

upload = files.upload()

df = pd.read\_csv('books.csv',error\_bad\_lines=False)

df.head(3)

**EDA:**

Before exploring the data, we have to clean the data for further steps. I started by removing the extra spaces, check for null values, duplicates and outliers and check for null values. The bookID is kept as the index.

df.isnull().sum()

bookID 0

title 0

authors 0

average\_rating 0

isbn 0

isbn13 0

language\_code 0

num\_pages 0

ratings\_count 0

text\_reviews\_count 0

publication\_date 0

publisher 0

dup = df.duplicated()

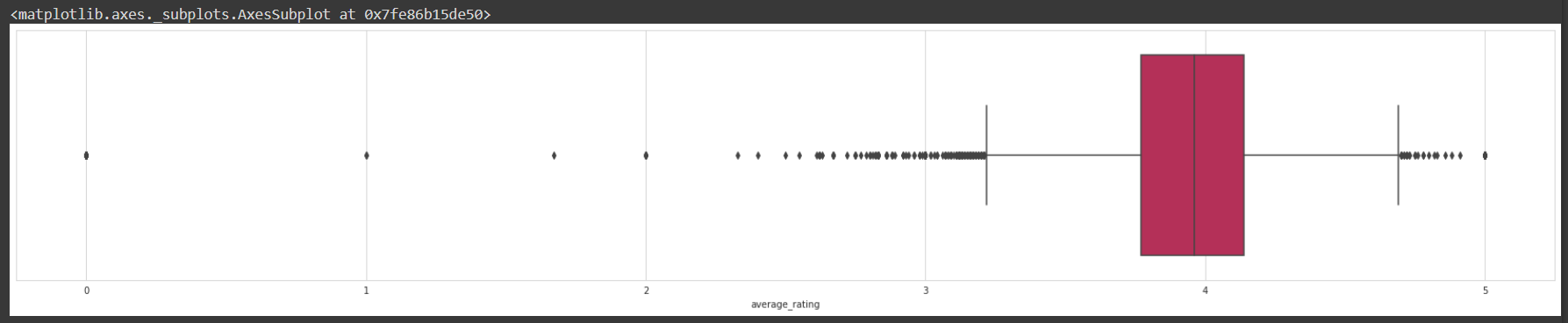
print(dup.sum())

0

From the outlier data we can see that there are no abnormal values. Then exploratory data analysis is done. Firstly unique values in the data is identified. After finding, I started with plots.

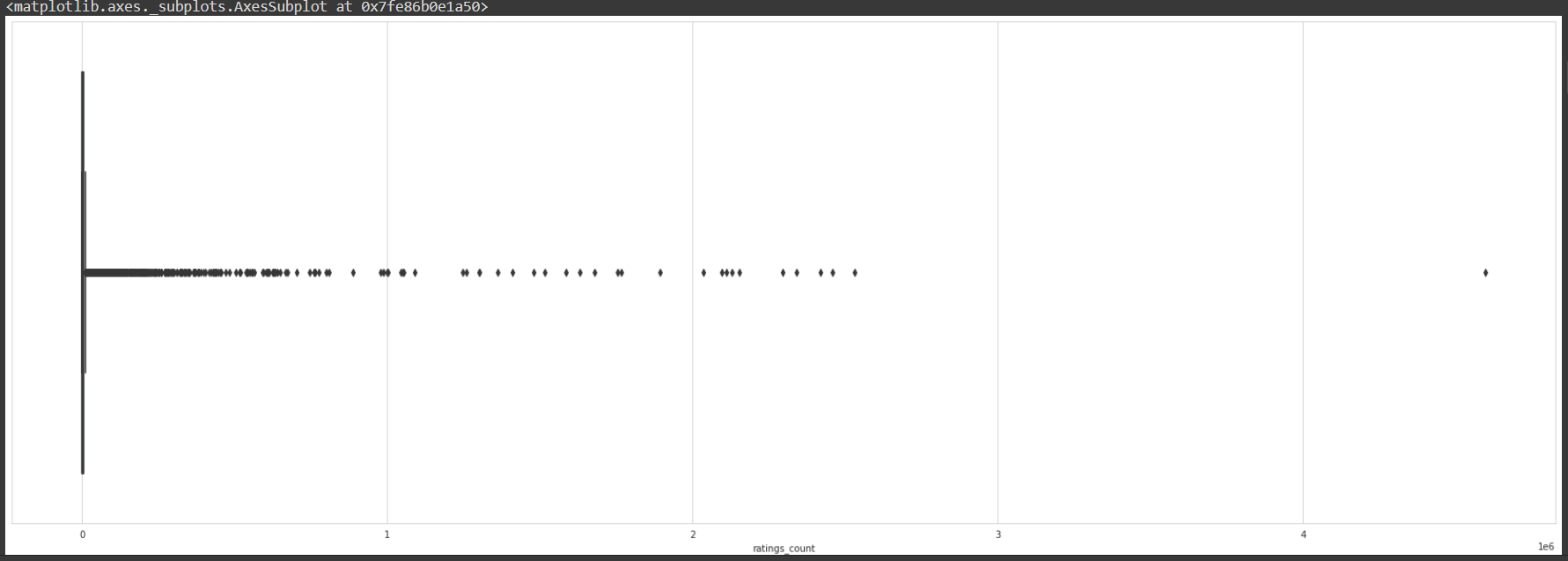
plt.figure(figsize=(30,5))

sns.boxplot(x=df['average\_rating'],palette = 'rocket\_r')



plt.figure(figsize=(30,10))

sns.boxplot(x=df['ratings\_count'],palette = 'colorblind')



The Lliad and The brothers karamazov are the most occurring books. Gullivers travel and Odyssey comes next. The communist manifesto is the least occurring book.

for feature in df.columns:

    uniq = np.unique(df[feature])

    print('{}: {} distinct values\n'.format(feature,len(uniq)))

bookID: 11123 distinct values

title: 10348 distinct values

authors: 6639 distinct values

average\_rating: 209 distinct values

isbn: 11123 distinct values

isbn13: 11123 distinct values

language\_code: 27 distinct values

num\_pages: 997 distinct values

ratings\_count: 5294 distinct values

text\_reviews\_count: 1822 distinct values

publication\_date: 3679 distinct values

publisher: 2290 distinct values

#Taking the first 20:

sns.set\_context('poster')

plt.figure(figsize=(20,15))

book = df['title'].value\_counts()[:20]

rating = df.average\_rating[:20]

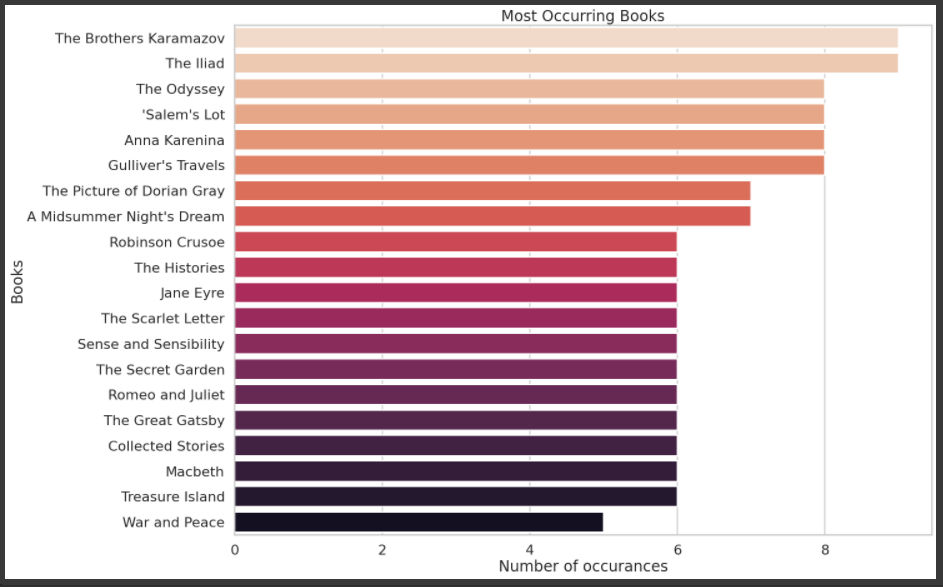
sns.barplot(x = book, y = book.index, palette='rocket\_r')

plt.title("Most Occurring Books")

plt.xlabel("Number of occurances")

plt.ylabel("Books")

plt.show()



Then we find the most frequently occurring language, which is English.

sns.set\_context('paper')

plt.figure(figsize=(15,10))

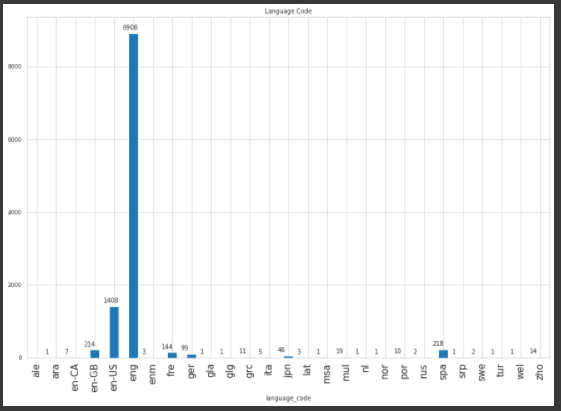
ax = df.groupby('language\_code')['title'].count().plot.bar()

plt.title('Language Code')

plt.xticks(fontsize = 15)

for p in ax.patches:

    ax.annotate(str(p.get\_height()), (p.get\_x()-0.3, p.get\_height()+100))

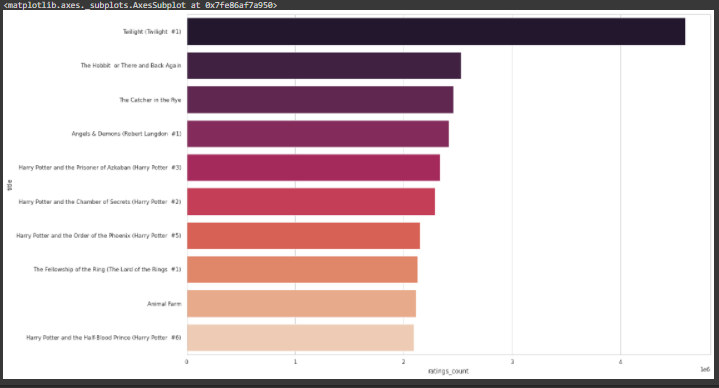


Next we identify the book with the highest rating, which we find it as Twilight.

most\_rated = df.sort\_values('ratings\_count', ascending = False).head(10).set\_index('title')

plt.figure(figsize=(15,10))

sns.barplot(most\_rated['ratings\_count'], most\_rated.index, palette='rocket')



Next we plot a graph for top 10 authors with the most books. From seeing the graph we can easily visualize the top authors.

sns.set\_context('talk')

most\_books = df.groupby('authors')['title'].count().reset\_index().sort\_values('title', ascending=False).head(10).set\_index('authors')

plt.figure(figsize=(15,10))

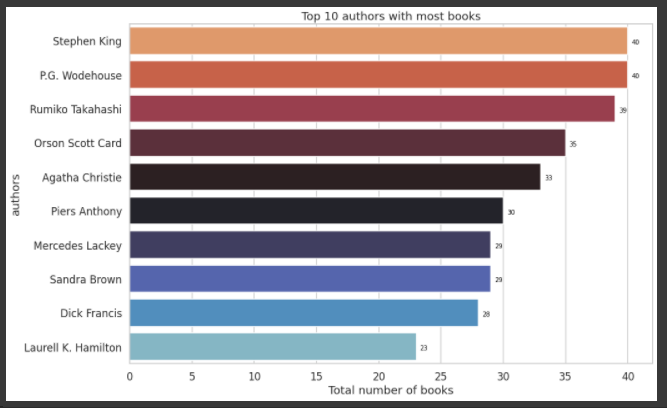
ax = sns.barplot(most\_books['title'], most\_books.index, palette='icefire\_r')

ax.set\_title("Top 10 authors with most books")

ax.set\_xlabel("Total number of books")

for i in ax.patches:

    ax.text(i.get\_width()+.3, i.get\_y()+0.5, str(round(i.get\_width())), fontsize = 10, color = 'k')

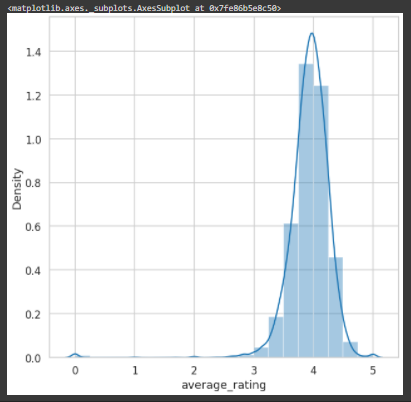


The following plot shows the average rating among the books. The average rating is shown as 4.5

plt.figure(figsize=(10,10))

rating= df.average\_rating.astype(float)

sns.distplot(rating, bins=20)



Next we segregate the percentage of books on various intervals of ratings. We can see that most of the ratings occur between 3 and 4 and 4 and 5.

def segregation(data):

    values = []

    for val in data.average\_rating:

        if val>=0 and val<=1:

            values.append("Between 0 and 1")

        elif val>1 and val<=2:

            values.append("Between 1 and 2")

        elif val>2 and val<=3:

            values.append("Between 2 and 3")

        elif val>3 and val<=4:

            values.append("Between 3 and 4")

        elif val>4 and val<=5:

            values.append("Between 4 and 5")

        else:

            values.append("NaN")

    print(len(values))

    return values

books['Ratings\_Dist'] = segregation(books)

ratings\_pie = books['Ratings\_Dist'].value\_counts().reset\_index()

labels = ratings\_pie['index']

colors = ['lightblue','darkmagenta','coral','bisque', 'black']

percent = 100.\*ratings\_pie['Ratings\_Dist']/ratings\_pie['Ratings\_Dist'].sum()

fig, ax1 = plt.subplots()

ax1.pie(ratings\_pie['Ratings\_Dist'],colors = colors,

        pctdistance=0.85, startangle=90, explode=(0.05, 0.05, 0.05, 0.05, 0.05))

#Draw a circle now:

centre\_circle = plt.Circle((0,0), 0.70, fc ='white')

fig1 = plt.gcf()

fig1.gca().add\_artist(centre\_circle)

#Equal Aspect ratio ensures that pie is drawn as a circle

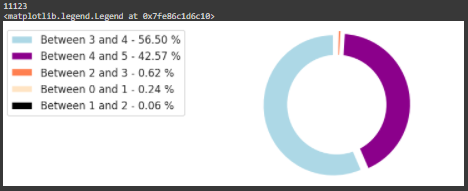
plt.axis('equal')

plt.tight\_layout()

labels = ['{0} - {1:1.2f} %'.format(i,j) for i,j in zip(labels, percent)]

plt.legend( labels, loc = 'best',bbox\_to\_anchor=(-0.1, 1.),)

Next we find if there is any relationship between average rating and number of reviews?



We can see that, most of the reviews were done for ratings around 4. This means that very less number of people give either full or 0 marks,

### 

### Here we find if there is any relationship between number of pages and average rating? Here because of the outliers, we can see that the whole graph is getting positively skewed. For this we will try to plot a graph having books with pages not more than 1000.

### 

### We find if there is any relationship between average rating and the numbers of rating received? Here too we can see that the outliers are affecting the plot. Hence we will take a temporary data frame having number of ratings more than 2000000

### Finally we can see that most of the ratings are 4 from the graph below.

### The final graph shows us which book has the highest ratings.