Guided Project Report

AssociationRuleMiningMarketBasketAnalysis

Name: Shruti Verma
Course: AI and ML
(Batch 4)
Duration: 10 months

Problem Statement: Build a relation between the products bought by the customers using store data and understand the loyalty of the customers.

Prerequisites

What things you need to install the software and how to install them:

Python 3.8 or higher versions This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not- recognized-as-an-internal-or-external- command/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.8 then run below commands in command prompt/terminal to install these packages pip install -U scikit-learn pip install numpy pip install scipy if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages conda install -c scikit-learn conda install -c anaconda numpy conda install -c anaconda scipy . Importing matplotlib.pyplot, Seaborn and datetime modules.Also , installed pip install chart_studio

Video Link

https://drive.google.com/file/d/1yF7Gt2BxeM40db4m_wHbI4vayuF9v7cH/view?usp=sharing

Dataset used

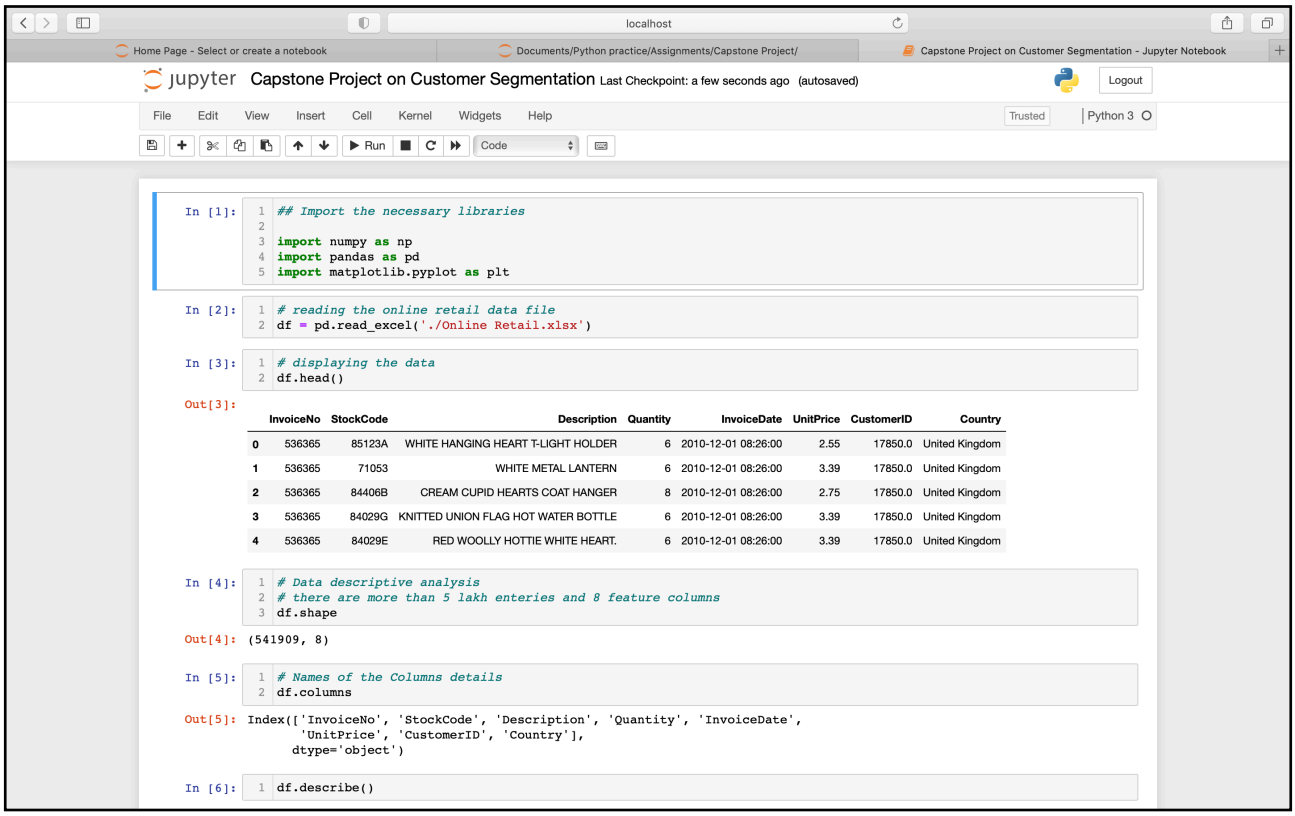
The data source is online retail data having close to 5.4 lakh enteries and 8 column features.The link is (https://archive.ics.uci.edu/ml/datasets/Online+Retail#)

Method used for detection

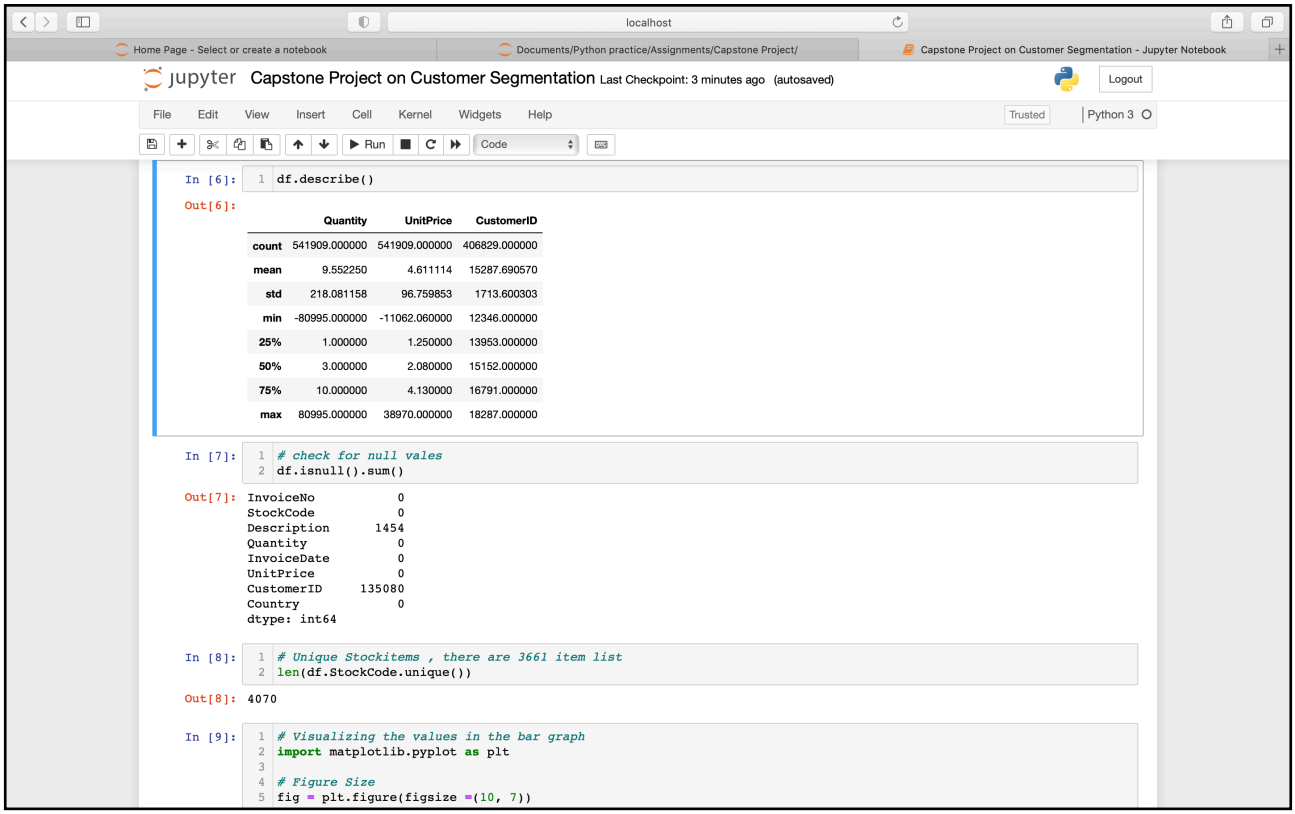
1. Loading and cleaning the data set.
2. Preprocessing the dataset
3. Applying RFM model and then clustering based on customer ids and RFM score
4. Using Elbow bend method finding the optimum number of clusters for dividing the customers into.

Importing the libraries and capturing images:

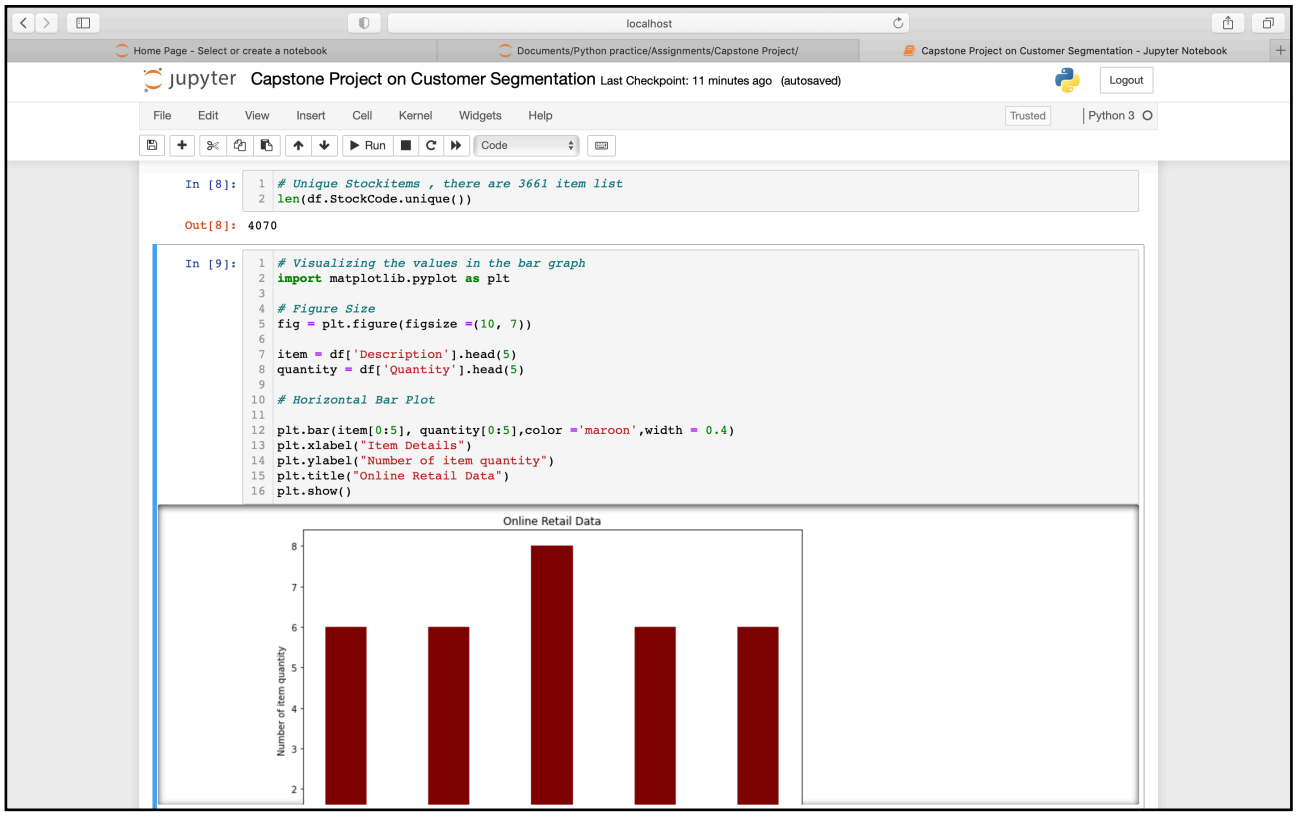
1. Importing the libraries and loading the online retail dataset



2. Checking for null values and number of items

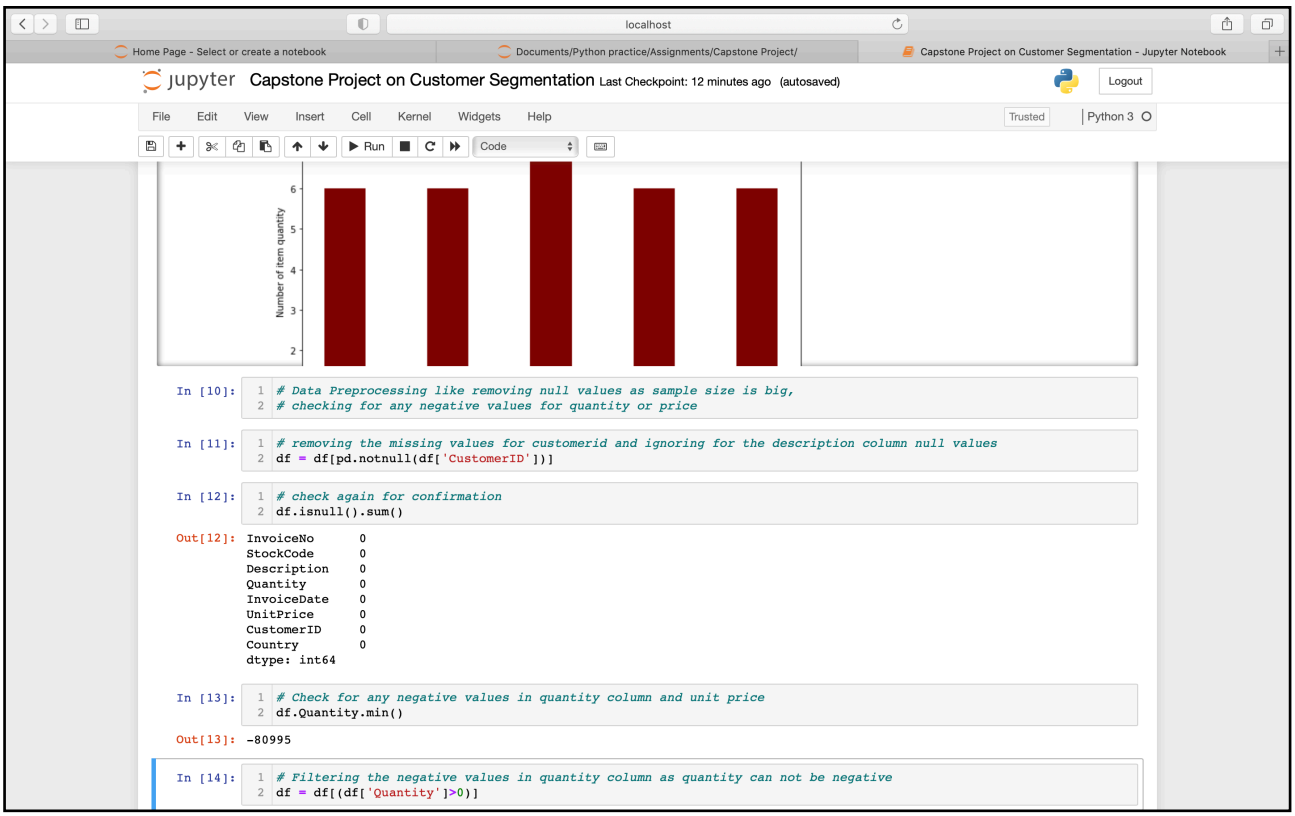


Visualising the data in bar graph

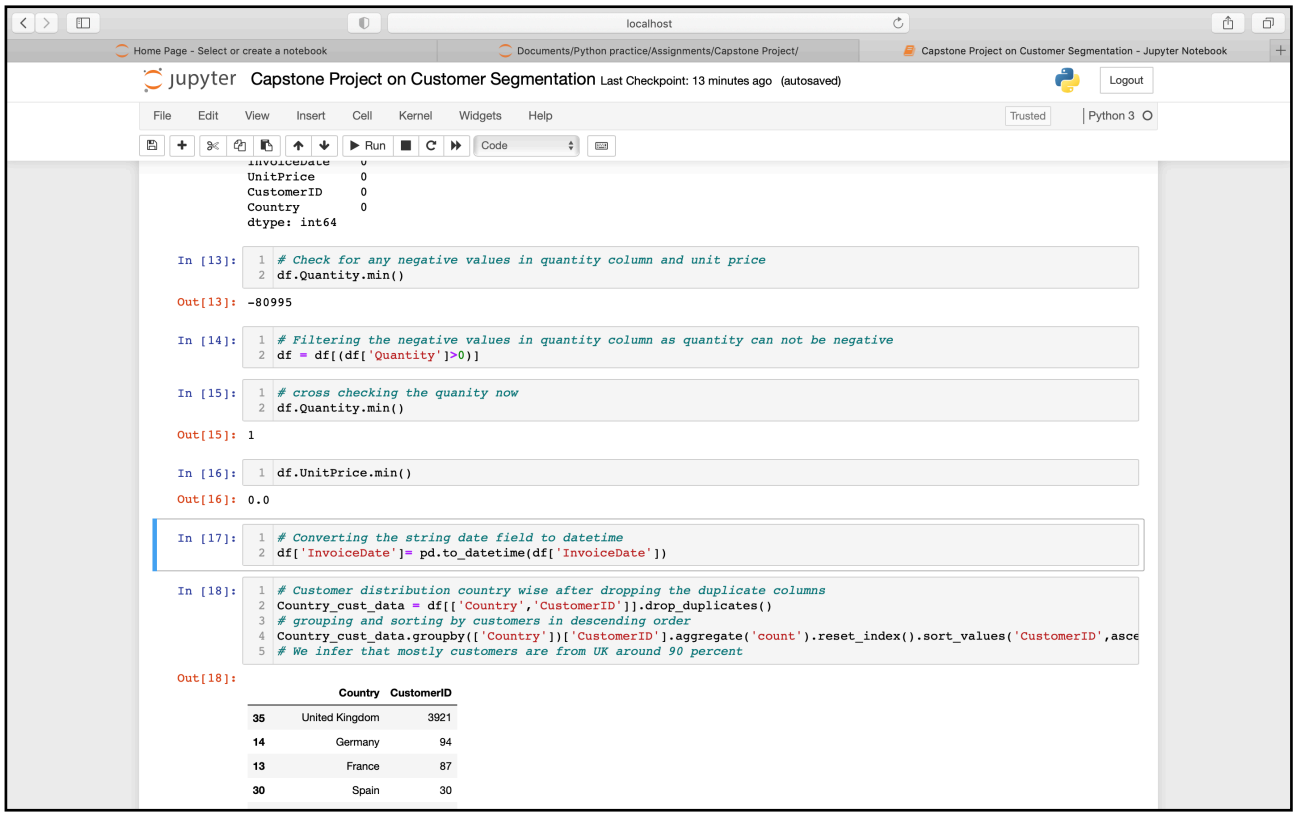


Data processing

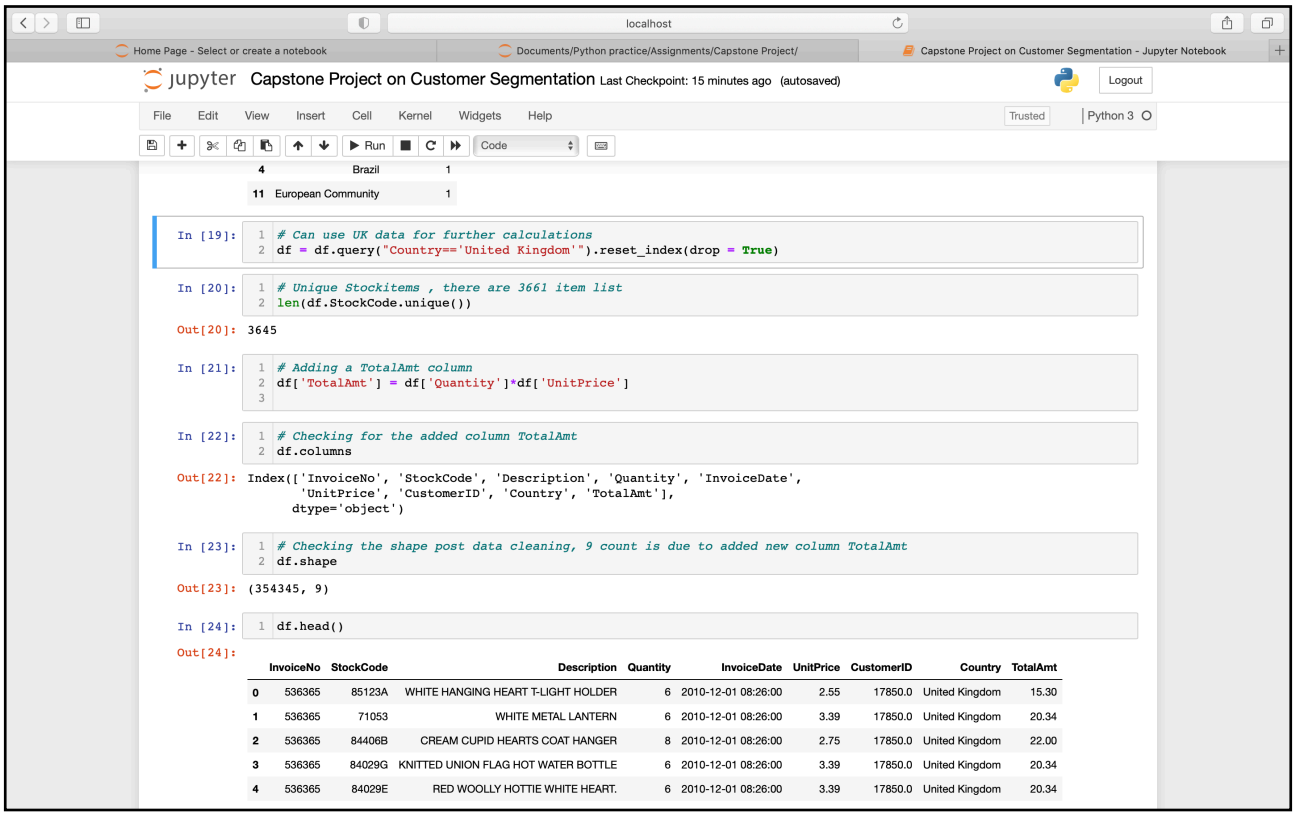
Removing the nulls and negative values for non negative value columns in any



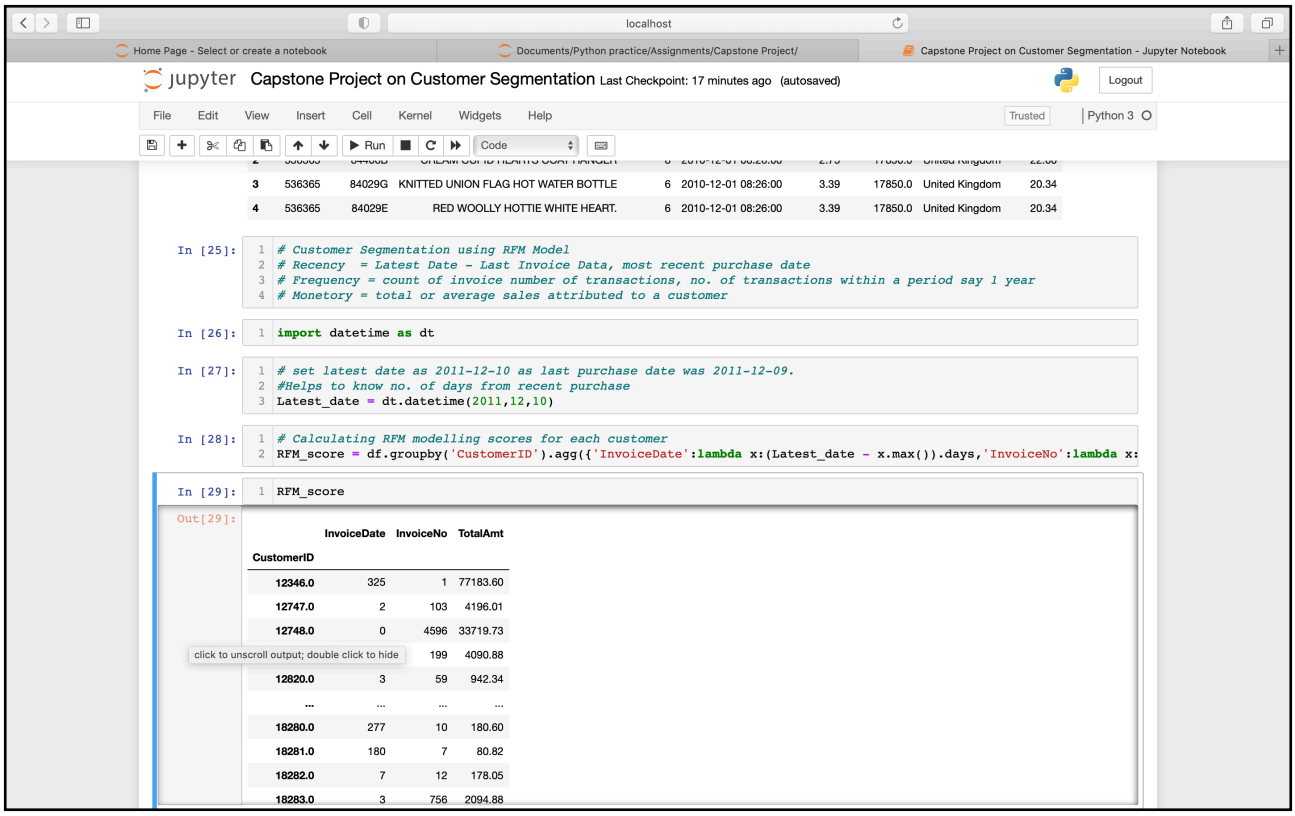
Fixing the datatype as date time and grouping by county



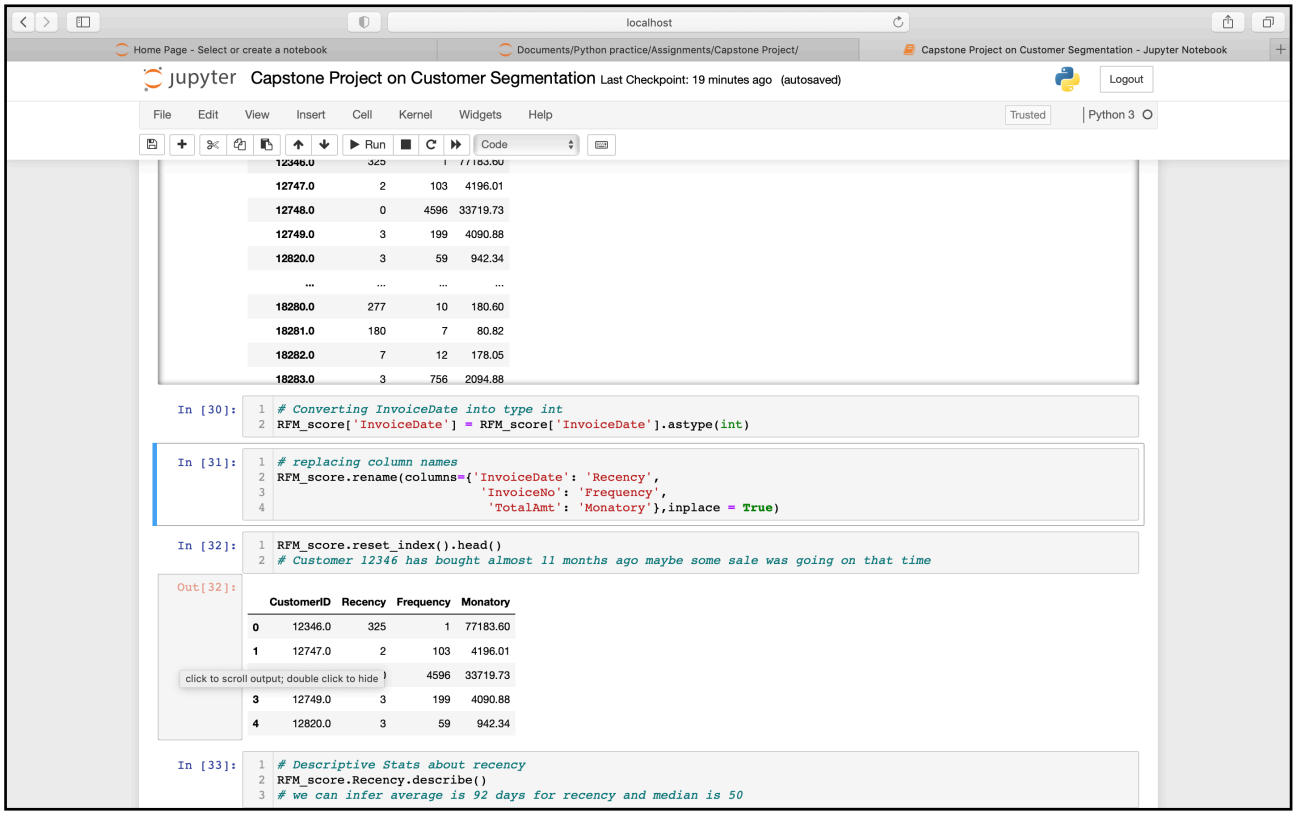
Adding Total Amount column and rechecking specifications of data



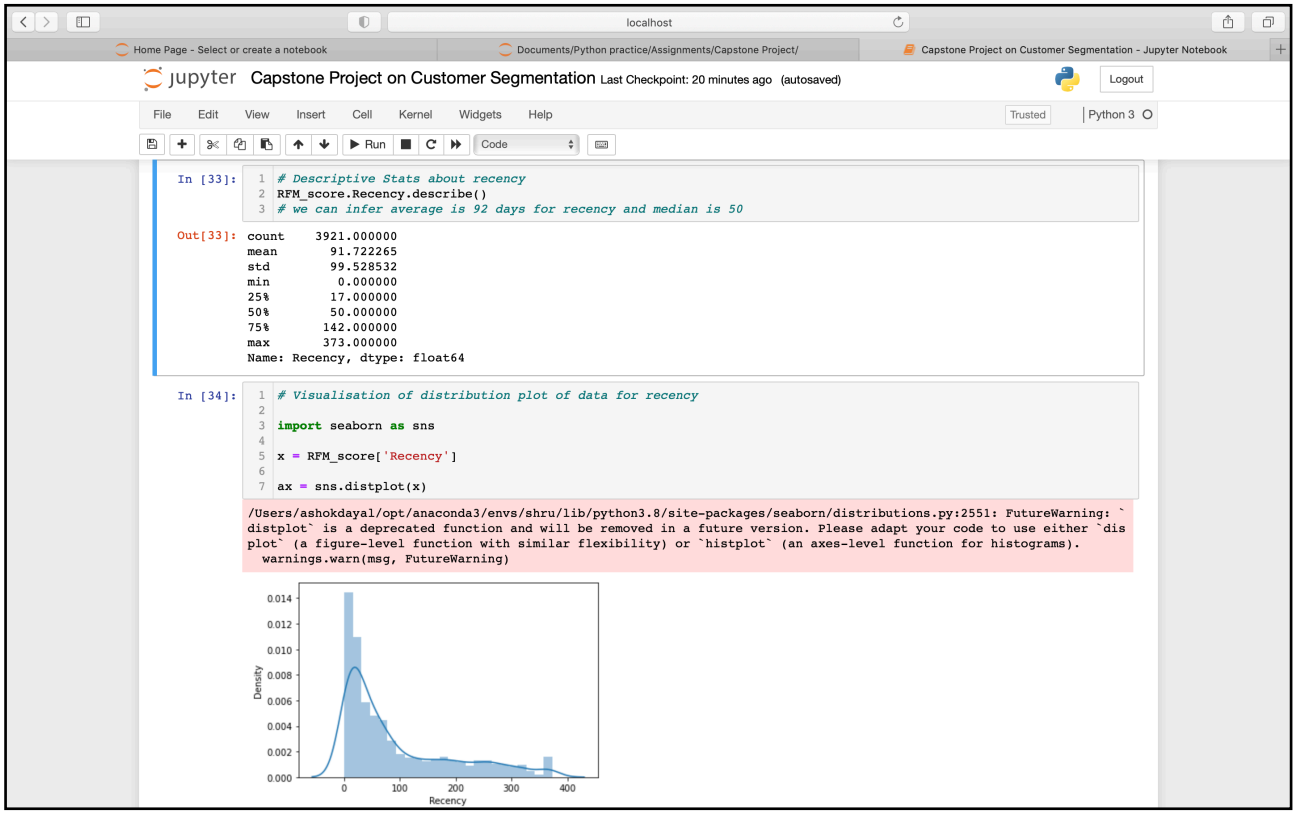
Applying RFM model

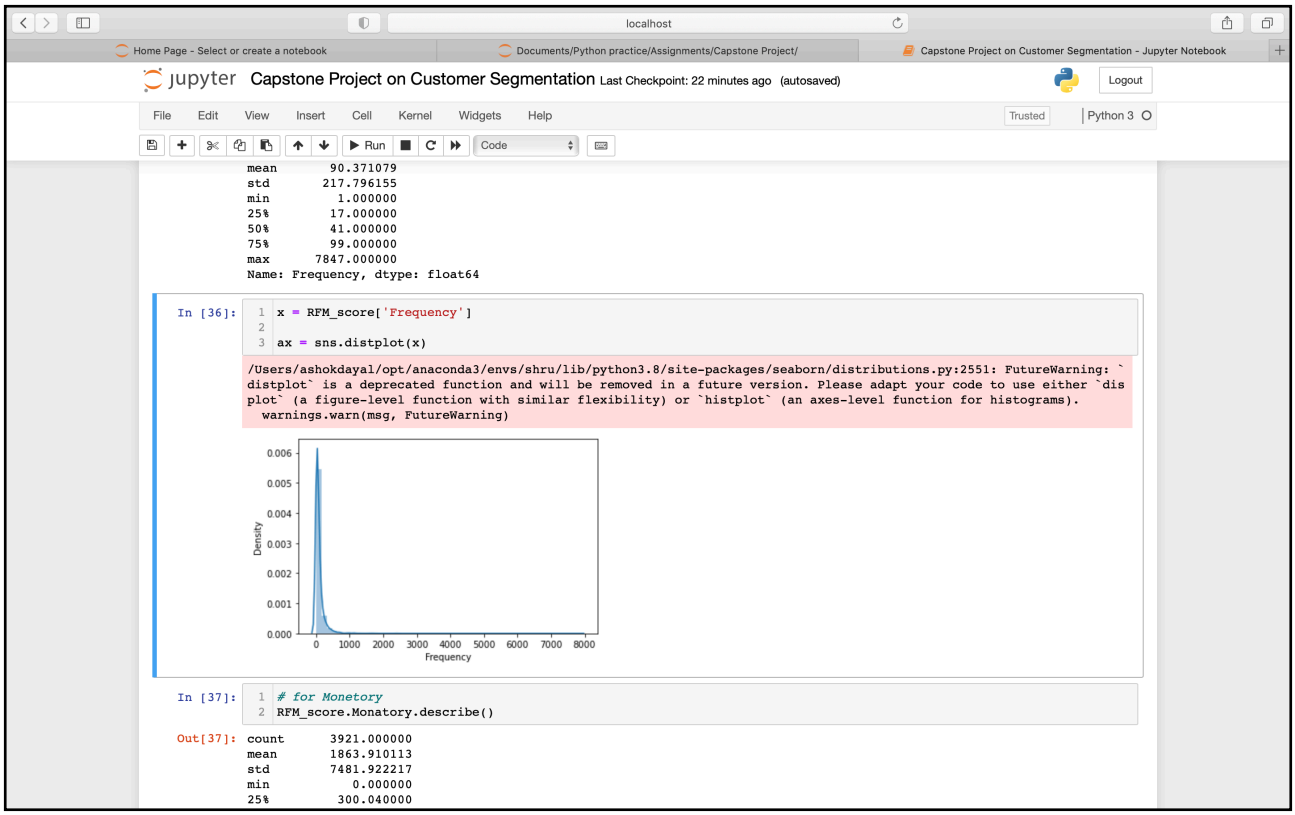


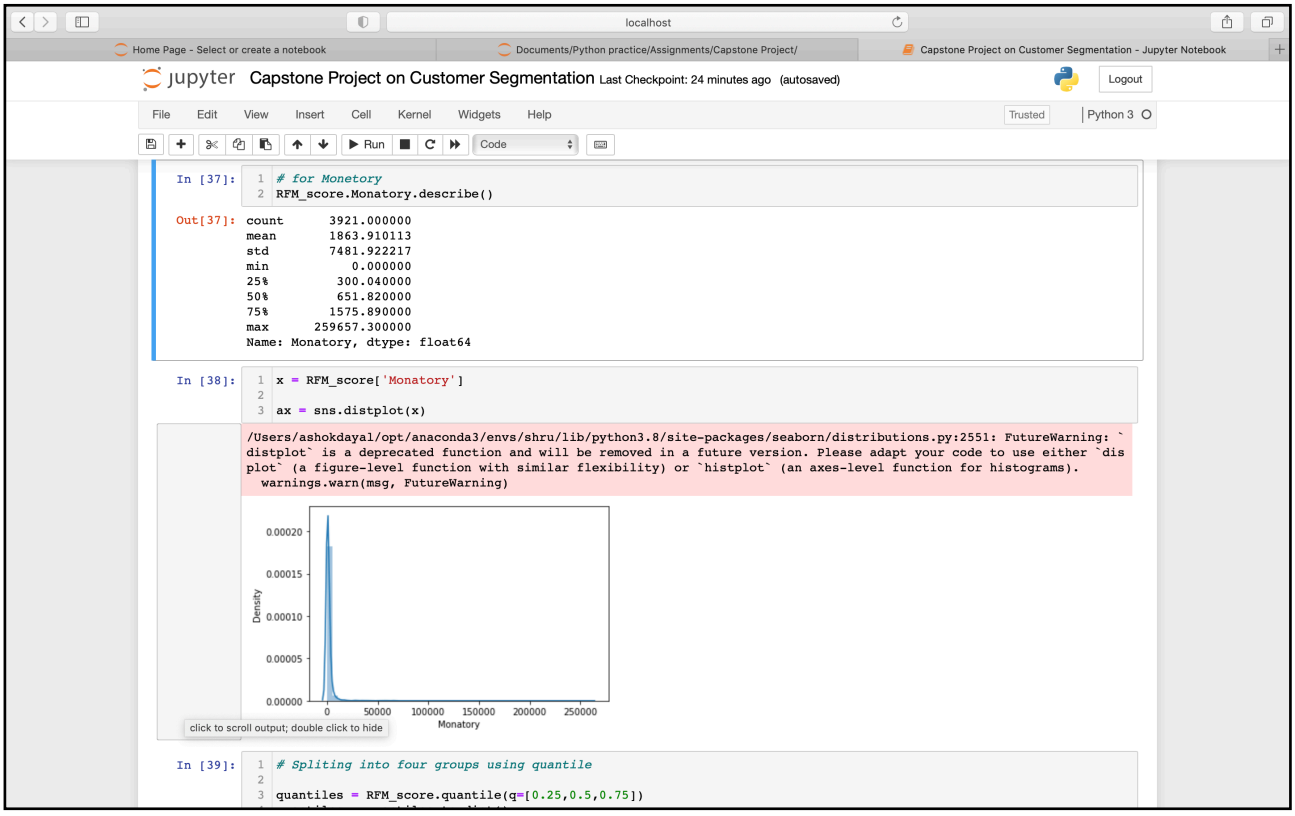
Renaming the columns to RFM



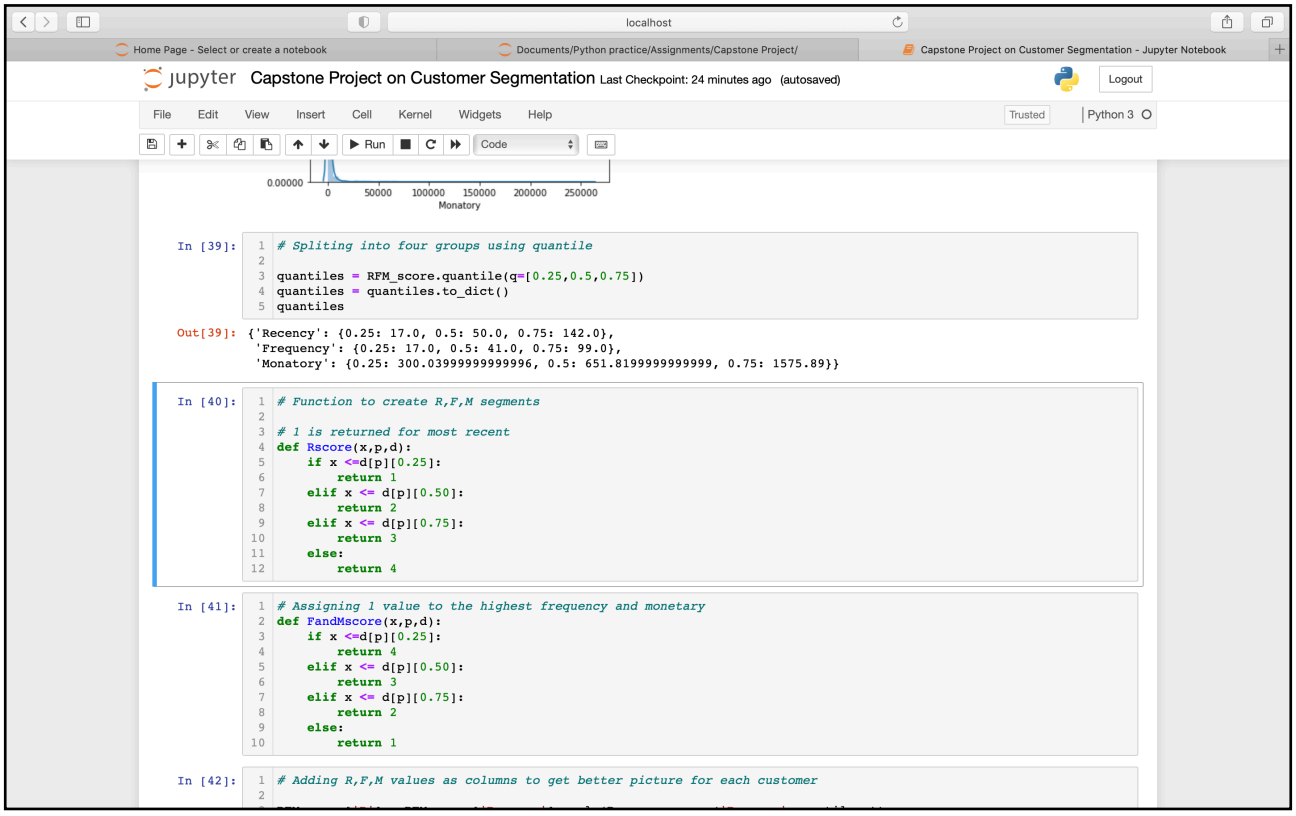
Visualization of distribution



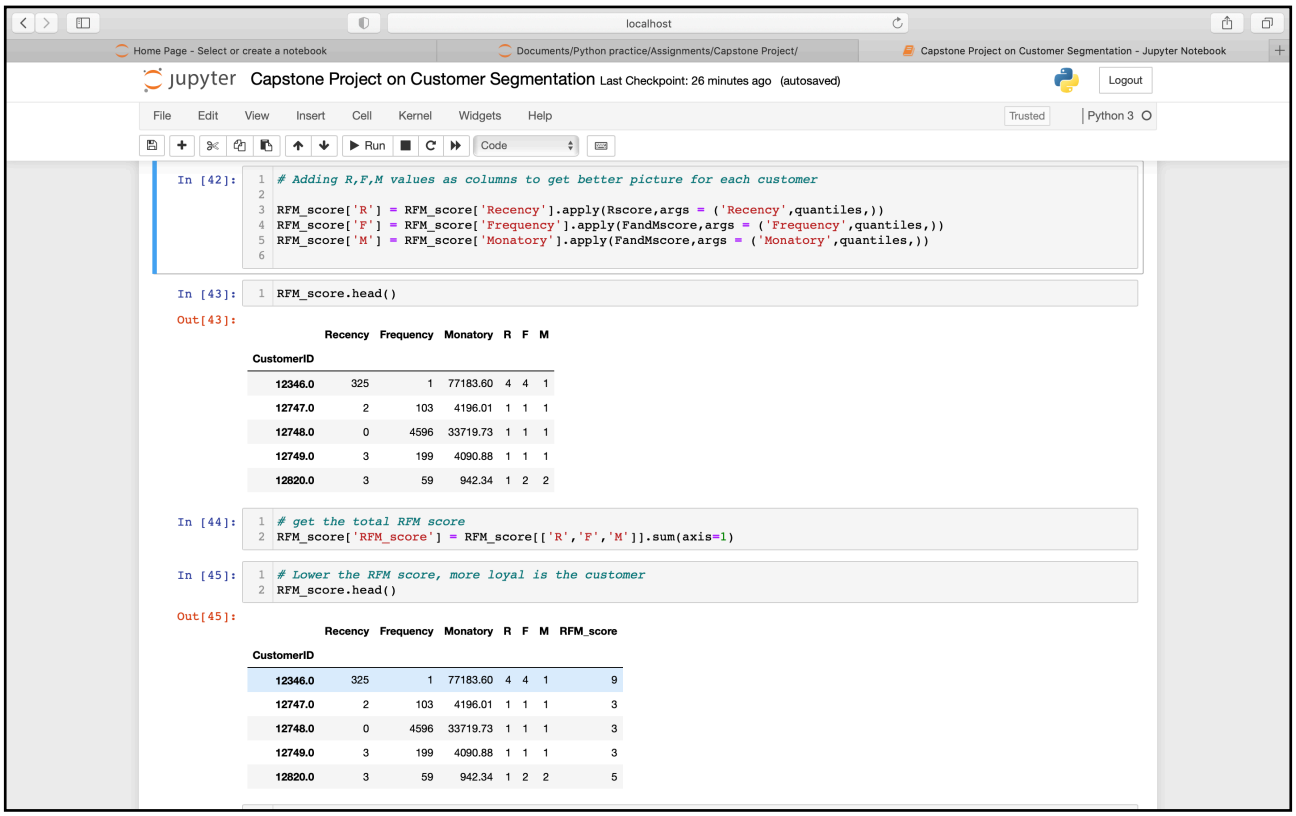




Splitting data for RFM into 4 groups

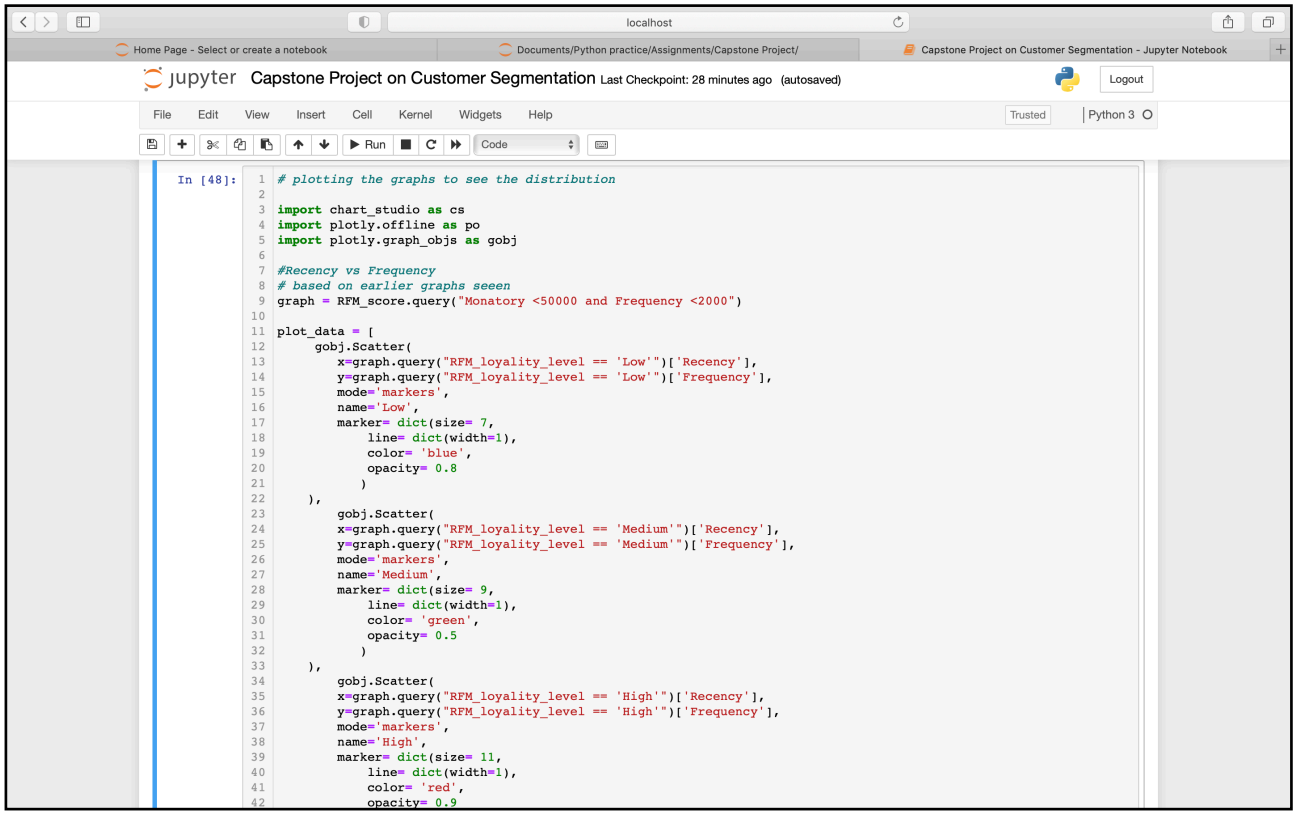


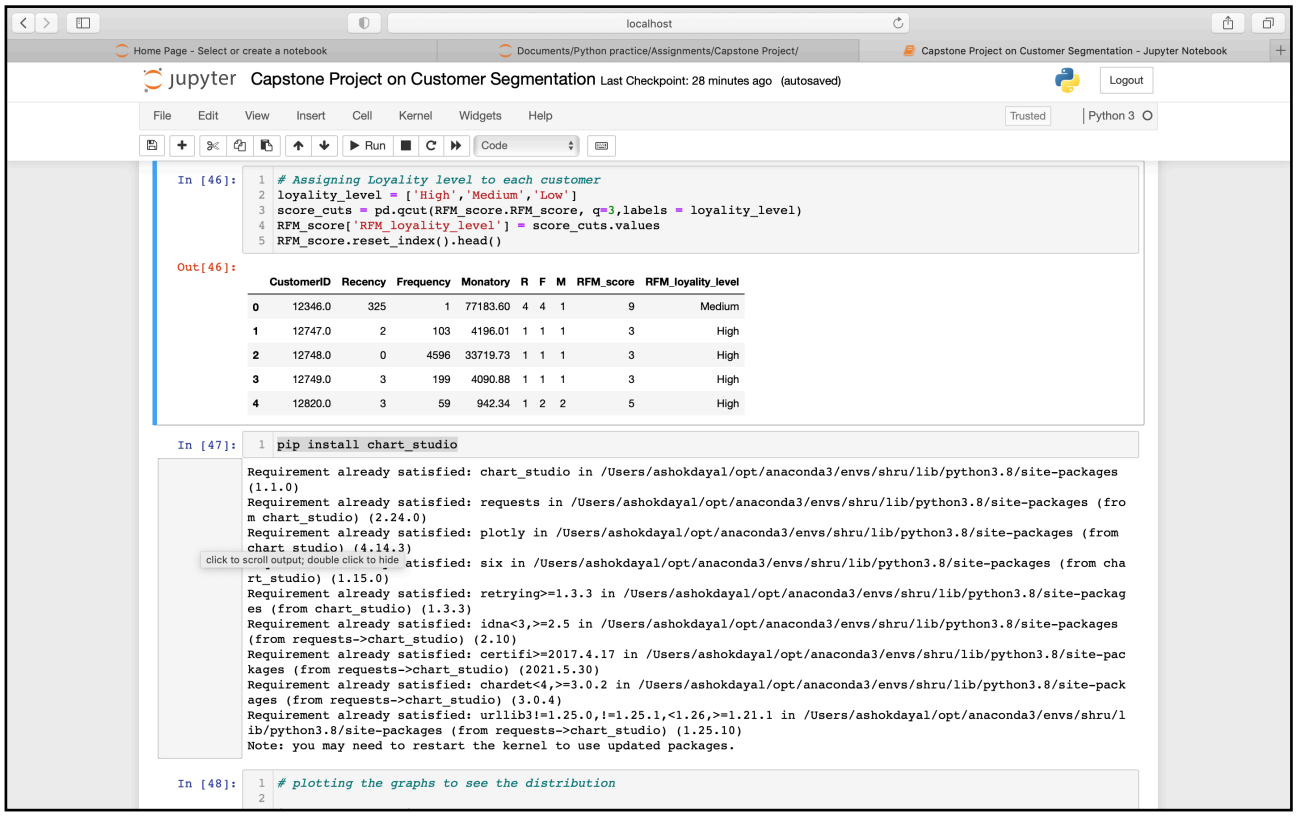
Calculating the total score , one value indicates high score and 4 indicates low score



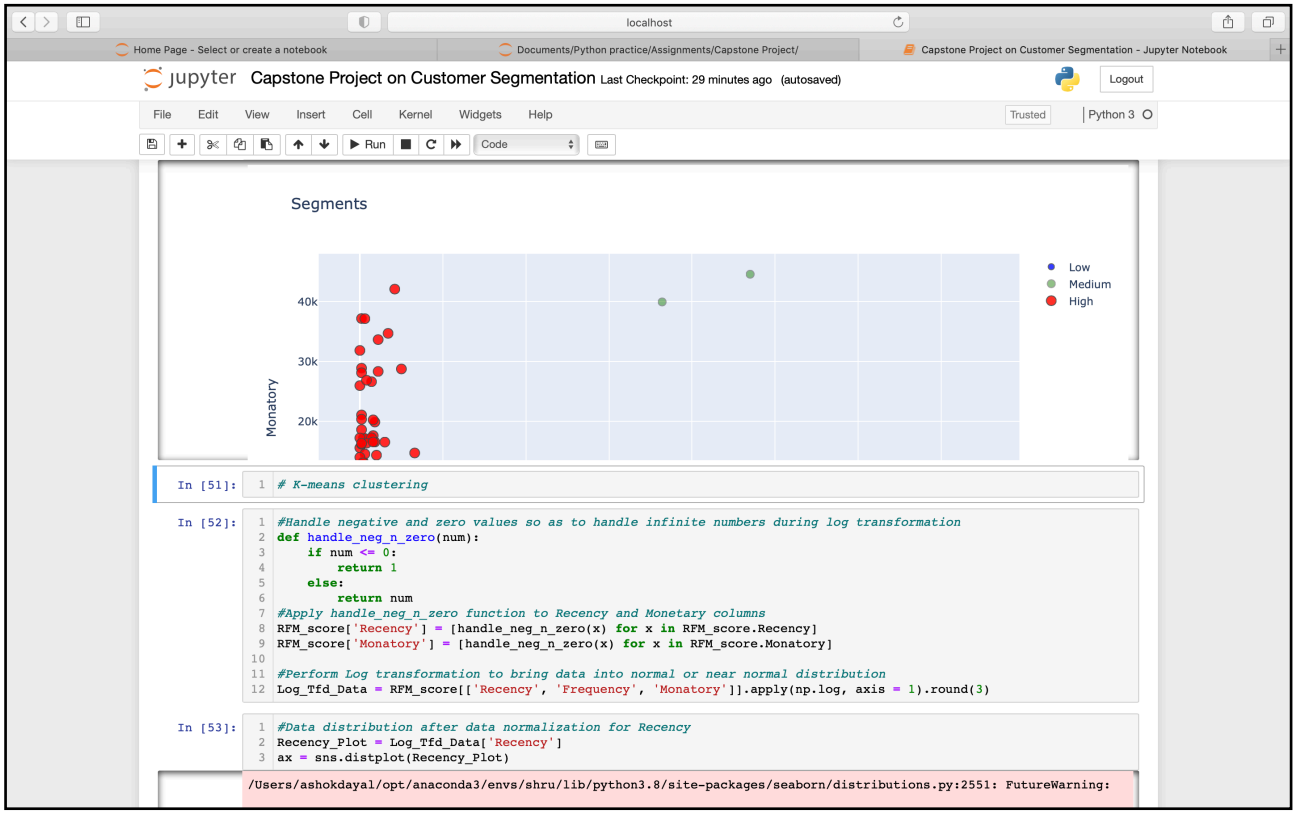
Assigning loyalty level

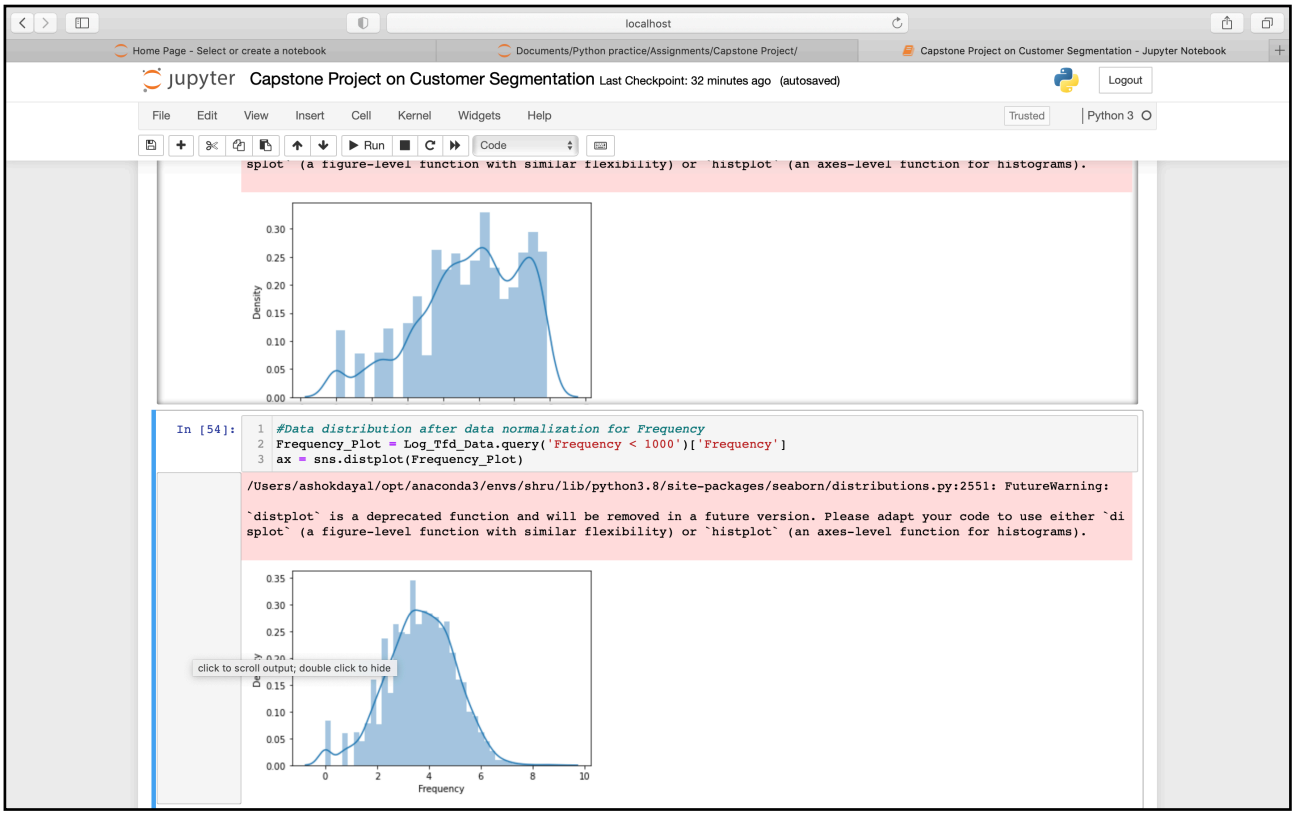
Plot to check the distribution

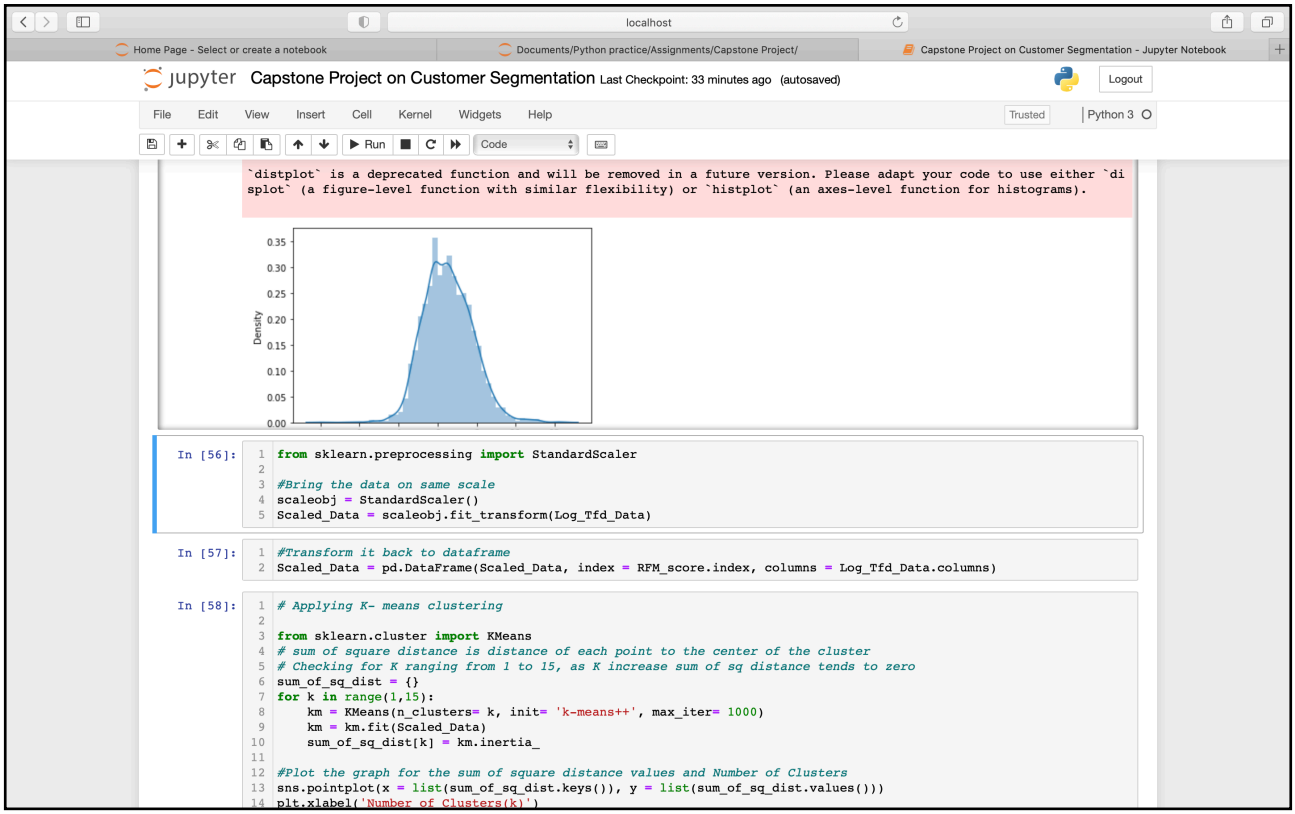


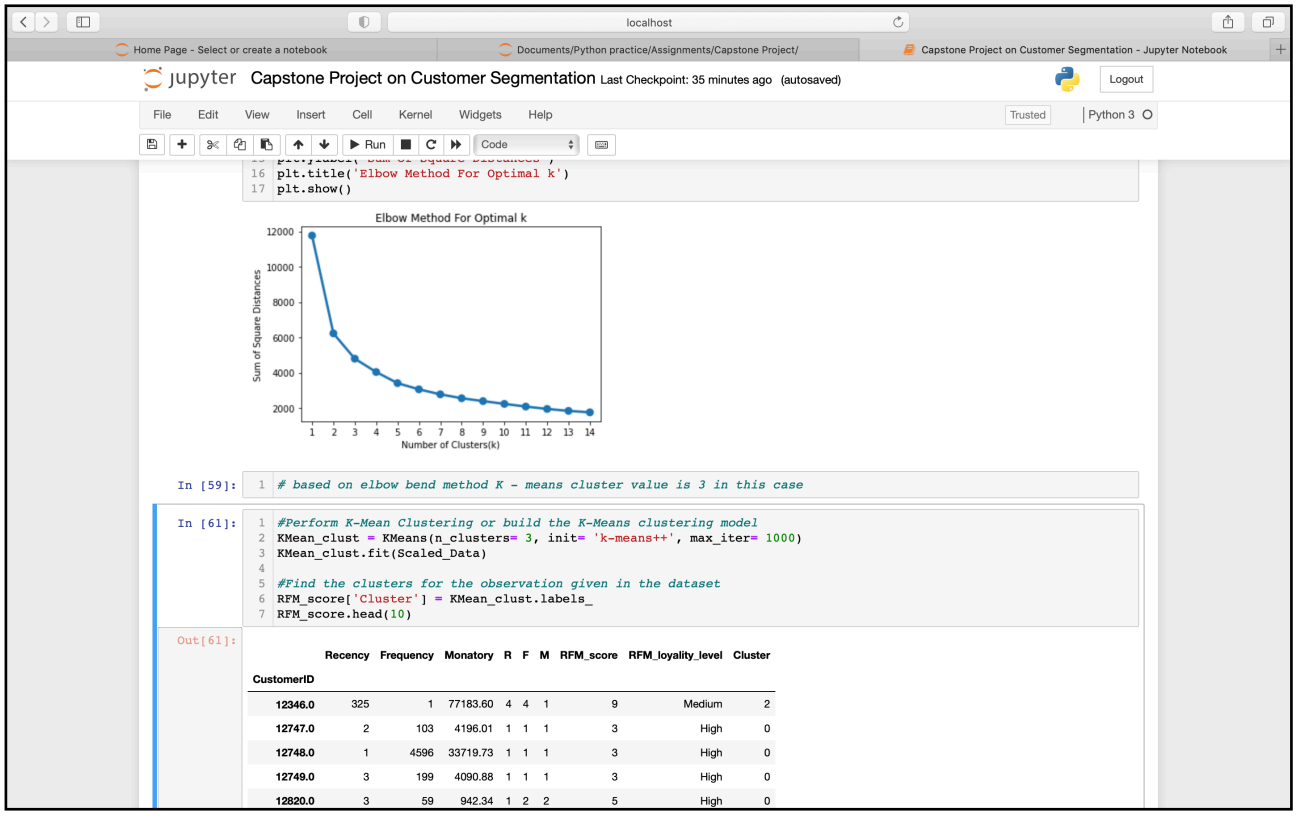


Applying K means clustering


localhost
Home Page - Select or create a notebook
Documents/Python practice/Assignments/Capstone Project/
Capstone Project on Customer Segmentation - Jupyter Notebook
jupyter Capstone Project on Customer Segmentation Last Checkpoint: 29 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Segments
Low
Medium
High
Monatory
40k
30k
20k
In [51]:
K-means clustering
In [52]:
#Handle negative and zero values so as to handle infinite numbers during log transformation
def handle_neg_n_zero(num):
if num <= 0:
return 1
else:
return num
#Apply handle_neg_n_zero function to Recency and Monetary columns
RFM_score['Recency'] = [handle_neg_n_zero(x) for x in RFM_score.Recency]
RFM_score['Monatory'] = [handle_neg_n_zero(x) for x in RFM_score.Monatory]
#Perform Log transformation to bring data into normal or near normal distribution
Log_Tfd_Data = RFM_score[['Recency', 'Frequency', 'Monatory']].apply(np.log, axis = 1).round(3)
In [53]:
#Data distribution after data normalization for Recency
Recency_Plot = Log_Tfd_Data['Recency']
ax = sns.distplot(Recency_Plot)
/Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages/seaborn/distributions.py:2551: FutureWarning:

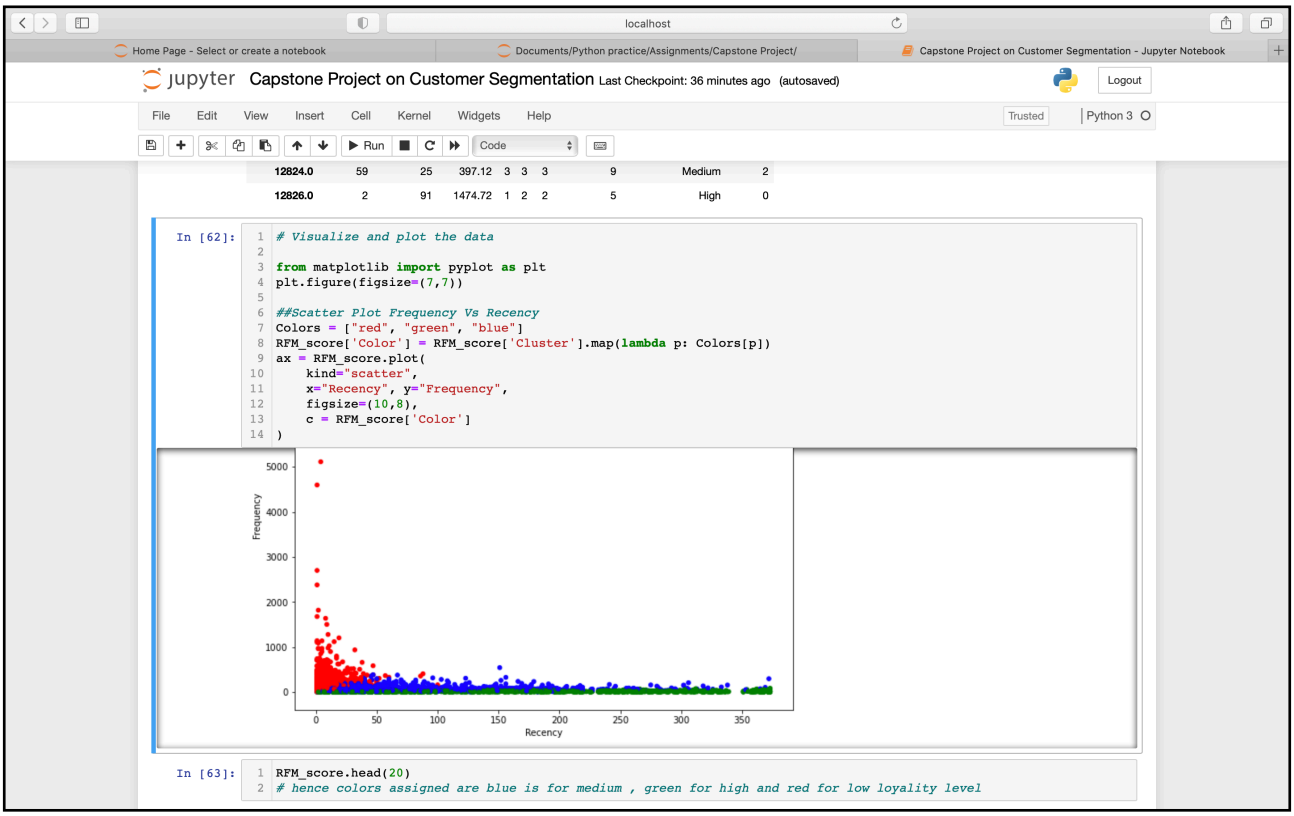

localhost
Home Page - Select or create a notebook
Documents/Python practice/Assignments/Capstone Project/
Capstone Project on Customer Segmentation - Jupyter Notebook
jupyter Capstone Project on Customer Segmentation Last Checkpoint: 32 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
splot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
Density
In [54]:
#Data distribution after data normalization for Frequency
Frequency_Plot = Log_Tfd_Data.query('Frequency < 1000')['Frequency']
ax = sns.distplot(Frequency_Plot)
/Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages/seaborn/distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `di
splot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
click to scroll output; double click to hide
Frequency

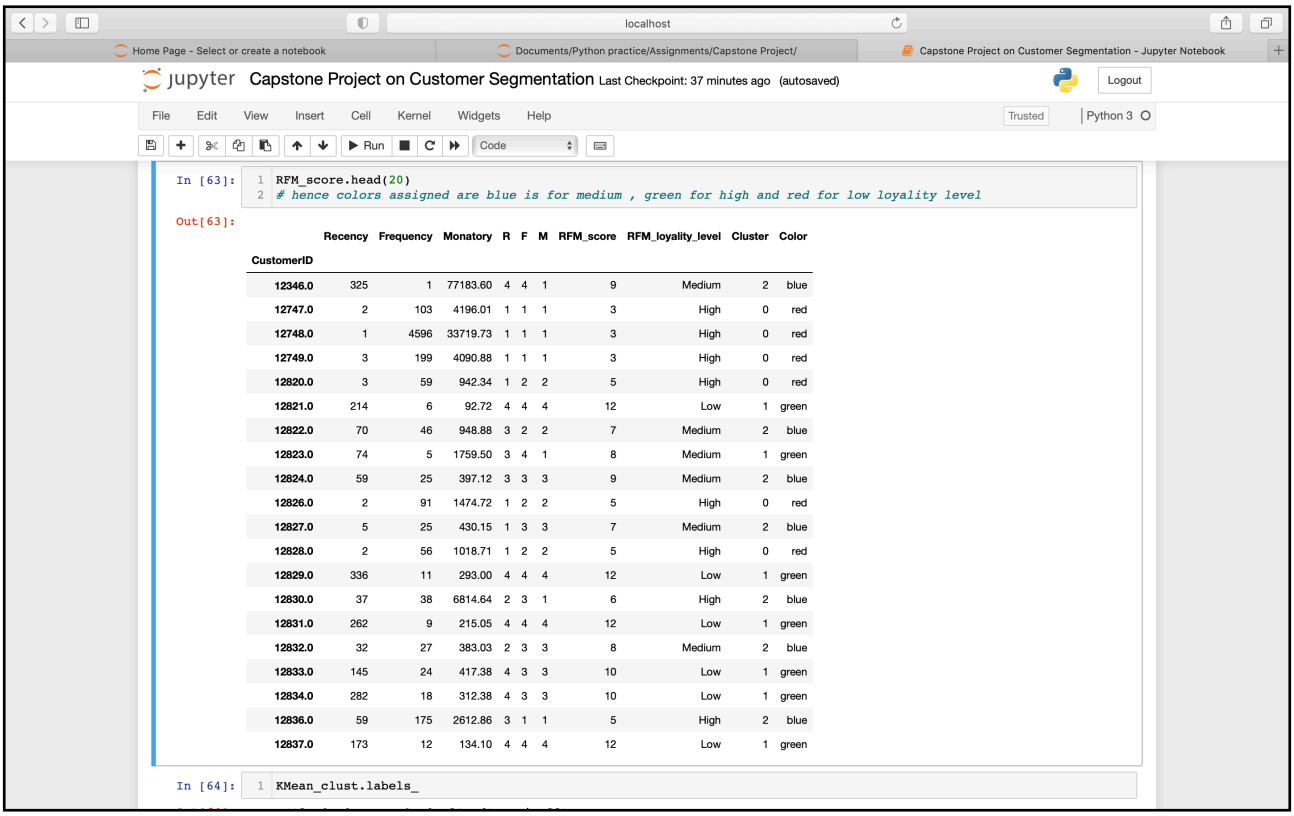




CustomerID	Recency	Frequency	Monatory	R	F	M	RFM_score	RFM_loyality_level	Cluster
12346.0	325	1	77183.60	4	4	1	9	Medium	2
12747.0	2	103	4196.01	1	1	1	3	High	0
12748.0	1	4596	33719.73	1	1	1	3	High	0
12749.0	3	199	4090.88	1	1	1	3	High	0
12820.0	3	59	942.34	1	2	2	5	High	0

Plot the graph to see the loyal customer distribution


localhost
Home Page - Select or create a notebook
Documents/Python practice/Assignments/Capstone Project/
Capstone Project on Customer Segmentation - Jupyter Notebook
jupyter Capstone Project on Customer Segmentation Last Checkpoint: 36 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
12824.0 59 25 397.12 3 3 3 9 Medium 2
12826.0 2 91 1474.72 1 2 2 5 High 0
In [62]:
# Visualize and plot the data
from matplotlib import pyplot as plt
plt.figure(figsize=(7,7))
##Scatter Plot Frequency Vs Recency
Colors = ["red", "green", "blue"]
RFM_score['Color'] = RFM_score['Cluster'].map(lambda p: Colors[p])
ax = RFM_score.plot(
kind="scatter",
x="Recency", y="Frequency",
figsize=(10,8),
c = RFM_score['Color']
)
Frequency
Recency
In [63]:
RFM_score.head(20)
# hence colors assigned are blue is for medium , green for high and red for low loyality level


localhost
Home Page - Select or create a notebook
Documents/Python practice/Assignments/Capstone Project/
Capstone Project on Customer Segmentation - Jupyter Notebook
jupyter Capstone Project on Customer Segmentation Last Checkpoint: 37 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [63]:
RFM_score.head(20)
# hence colors assigned are blue is for medium , green for high and red for low loyality level
Out[63]:
CustomerID Recency Frequency Monatory R F M RFM_score RFM_loyality_level Cluster Color
12346.0 325 1 77183.60 4 4 1 9 Medium 2 blue
12747.0 2 103 4196.01 1 1 1 3 High 0 red
12748.0 1 4596 33719.73 1 1 1 3 High 0 red
12749.0 3 199 4090.88 1 1 1 3 High 0 red
12820.0 3 59 942.34 1 2 2 5 High 0 red
12821.0 214 6 92.72 4 4 4 12 Low 1 green
12822.0 70 46 948.88 3 2 2 7 Medium 2 blue
12823.0 74 5 1759.50 3 4 1 8 Medium 1 green
12824.0 59 25 397.12 3 3 3 9 Medium 2 blue
12826.0 2 91 1474.72 1 2 2 5 High 0 red
12827.0 5 25 430.15 1 3 3 7 Medium 2 blue
12828.0 2 56 1018.71 1 2 2 5 High 0 red
12829.0 336 11 293.00 4 4 4 12 Low 1 green
12830.0 37 38 6814.64 2 3 1 6 High 2 blue
12831.0 262 9 215.05 4 4 4 12 Low 1 green
12832.0 32 27 383.03 2 3 3 8 Medium 2 blue
12833.0 145 24 417.38 4 3 3 10 Low 1 green
12834.0 282 18 312.38 4 3 3 10 Low 1 green
12836.0 59 175 2612.86 3 1 1 5 High 2 blue
12837.0 173 12 134.10 4 4 4 12 Low 1 green
In [64]:
KMean_clust.labels_