ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse601032021.pdf
https://doi.org/10.30534/ijatcse/2021/601032021

Systematic Approach to Perform Task Centric Exploratory Data Analysis with Case study

Parvatham Niranjan Kumar[1], Kambhampati Vijay Kumar [2]
[1]Assistant Professor, Anurag Engineering College, Kodad, India,niranjan.1216@gmail.com
[2] Assistant Professor, Anurag Engineering College, Kodad, India,vijay.kambhampati@gmail.com,

ABSTRACT

Exploratory data analysis is a method to summarize main characteristics of data, and also to understand data more deeply using visualization techniques. This paper focuses on defining systematic approach in the form of well-defined sequence of steps to explore data in various aspects. Every organization produces lot of data. Organization needs to analyze this data very carefully to extract hidden patterns in the data. Task Centric EDA [2] produces actionable insights as outcome to improve business process. This uses *Python* programming language and *Jupyter* Notebook for data analysis. Python is an object oriented and interactive programming language, which contains rich sets of libraries like *pandas, MATplotlib, seaborn*[10] etc. We have used different types of charts and various types of parameters to analyze retail dataset and to improve sales using precision marketing.

Keywords: Exploratory data analysis, machine learning, EDA, sea born, matplotlib, precision marketing

1. INTRODUCTION

Data is growing very faster in today's world. Every organization produces and also depends on a lot of data in their everyday processes. It is not easy to process the data manually. Organizations need to understand data carefully, before making assumptions or decisions based on this data.

There are three motivations for analyzing data**.** First to understand what has happened or what is happening, second to predict what is likely to happen in the near future, third to guide us in making decisions.

Data analysis and visualization tools help us to understand data much deeper. Data analytics allow organizations to understand their business efficiency and performance, and also helps in making informed decisions. For example, an e-commerce online store might be interested in analyzing customer attributes to make ads by targeting particular categories of customers for improving sales.
Exploratory Data Analysis (EDA) is an approach which uses both descriptive statistics and graphical tools to have better understanding of data. Especially, "Graphs" are important because humans are much better at seeing patterns in graphs than in large collection of numbers.

EDA is treated as an art of looking at data in an effort to understand the underlying structure of the dataset. It enables data analysts and data scientists to bring right information to the right people. It will be considered as the most important step on which a data-driven organization should focus.

EDA helps to summarize statistical characteristics of dataset by focusing on four key aspects, i.e., Measuring of central tendency (comprising of the mean, mode and median), Measuring of spread (comprising of standard deviation and variance) and Shape of the distribution. Rest of the paper is organized as follows. Section II presents significance of EDA, Section III explains about steps in EDA and also various techniques for the exploratory data analysis. Section IV discusses how to conduct exploratory data analysis using Python and Jupyter Notebook, while Section V presents how to work with datasets to conduct Exploratory Data Analysis with case study. Finally, Section VI presents the concluding remarks.

2. THE SIGNIFICANCE OF EDA

Appropriate and well-established decisions should be made by data driven organizations using huge amount of data collected from various sources. It is highly impossible to sense datasets containing more than a handful of data points without using computing tools. Exploratory Data Analysis is the key, and it is the first step in data mining process.

The Key components of Exploratory Data Analysis includes summarizing data, statistical analysis and visualization of data. Certain insights collected by exploring the data help us to make further decisions.

EDA actually reveals ground truth about the content without making any underlying assumptions. This is the reason why data scientists use this process to actually understand what type of modelling and hypothesis can be created for further analysis [13].

3. STEPS IN EDA

Having understood what EDA is, and its significance, let's understand the various steps involved in data analysis. Basically it involves four different steps[13]. Let's go through each of them to get a brief understanding of each step.

3.1 Problem Definition

Before trying to perform analysis to extract useful insights from the data, it is essential to define the business problem to be solved. The problem definition is the driving force for a data analysis. The following activities are involved in this step.

- Defining the main objective of the analysis
- Obtaining the current status of the data
- Outlining the main roles and responsibilities
- Creating an execution plan

3.2 Data Preparation

This step involves preparing the dataset ready for actual analysis. In this step, we try to digest schema of dataset (Column names and their data types) and main characteristics of the data. Dataset cleaning process will be done by removing non-relevant data, transforming the data, and dividing the data into required chunks for analysis. The following activities are done in this step.

- Identifying shape of dataset.
- Identification of variables and data types
- Analyzing the basic metrics
- Detecting invalid data type of variables.
- Variable transformations
- Missing value treatment
- Outlier treatment
- Dimensionality Reduction

3.3 Data Analysis

This is one of the most crucial step that deals with descriptive statistics and analysis [5] of the data. The following visual methods help in summarizing the data, finding the hidden correlation and relationships among the data.

3.3.1Graphical Univariate Analysis

Univariate analysis of data is done using only one variable. Since it’s a single variable, it doesn’t deal with relationships among variables. Univariate analysis describes patterns that exist within the data. Line chart and Histogram are used for performing univariate analysis:

3.3.2 Bivariate Analysis

Bivariate analysis is to understand the relationship between two columns. There are many visualizations to perform bivariate analysis. For example, Scatter plot can be drawn to check linear relationship between year_built and house_price, and hexbin plot to check the distribution of price in different year ranges.

3.3.3 Correlation Analysis

Correlation analysis is commonly used to find important features or to identify redundant features. Heatmap gives correlation matrix which is used to perform correlation analysis, where each cell in the matrix shows the correlation between two columns. It shows which features are strongly correlated with the target variable and which two features are highly correlated with each other.

3.4 Exploring Data

This is the heart of entire EDA process**.** In this step, sequence of relevant questions will be prepared as story telling process to explore the data and also to answer the following questions [14].

- What happened or what is happening?
- To predict what will happen, in the near future
- Actionable insights help organizations to take informed decisions

This questionnaire helps us to extract hidden patterns in data and helps us in finding solution for a given problem.

3.5 Conclusion with Actionable Insights

This step involves presenting analysis results to the target audience in the form of graphs, summary tables, maps, and diagrams. This step suggests Actionable Insights, which are interpreted by the business stakeholders to improve business process.

4. IMPLEMENTATION

4.1 Python

Python is the popular language used for Exploratory Data Analysis. It has rich sets of libraries. Visualization process can make it easier to create the clear report. Pandas is the most powerful package in python to perform data analysis. It is built on the top of the NumPy package. Matplotlib or seaborn can be used to draw plots.

4.2 Jupiter Notebook

Jupyter Notebook is a web-based interactive development environment for creating notebook documents. A Jupyter Notebook contains a list of input and output ordered cells that can contain code, Markdown text, mathematical expressions, plots, charts, and media. Jupiter Notebooks utilize the **.ipynb** file extension. A Jupyter Notebook is a great way to build step-by-step interactive Python programs. The technology is particularly well-suited for data analysis and plotting.

5. WORKING WITH THE DATA SET

It’s time to explore the data and find about it. The data we are using belongs to Retail Dataset. We are going to analyze this data with possible set of options.

5.1 Problem Definition

Now a days, it has been recognized that precision marketing plays a key role in generating profit.

Precision marketing makes the task of providing personalized customer service and customers are better informed about the products that they like. It helps enterprises to gain profits by using high-efficiency marketing.

The accelerated pace of economic globalization and increasing market competition, led enterprise managers to face the problem in choosing the right strategic decision-making policies for selling the right products to the right customers at the right time.

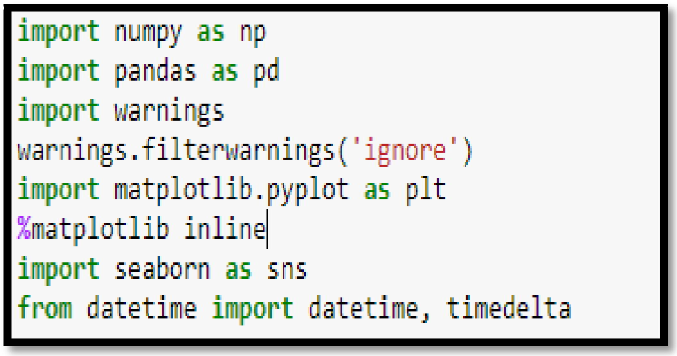
The main objective of the EDA on this dataset is to help managers identify the potential characteristics of different types of customers and put forward appropriate precision marketing strategies, which can greatly optimize inventory for every customer type.

The availability of customer data and their transaction records provide better understanding of customer's consumption behaviors and preferences. The real-world data from a company in UK were collected and used in this case study. Exploratory Data Analysis on this dataset should extract following patterns from data.

- Identify important attributes to distinguish different customer groups.
- Discover transactional patterns of different customers
- Verify the assumption of cancelled orders/invoices that may help in preventing future cancellations.
- To get an overview of the general customers purchase behavior.

5.2 Data Preparation

Importing Packages: Python packages can be imported as shown in Figure1.



```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from datetime import datetime, timedelta
```

Figure 1: Figure Shows on Python Statements to Import Packages

5.2.1 Loading Data

retail_df=pd.read_csv('online_retail.csv')

5.2.2 Identifying Shape of Dataset

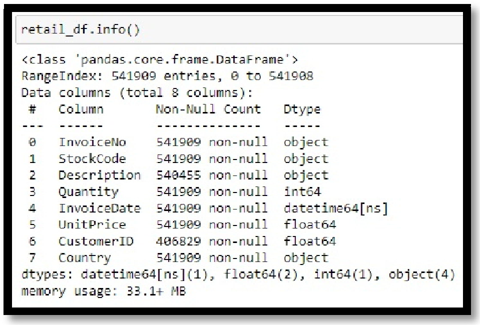
retail_df.shape

(541909, 8)
541909 rows, 8 columns

5.2.3 Identification of Variables and Data Types

The following command shows output as shown in Figure 2.

retail_df.info()



```
retail_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

Figure 2: Columns and their Types in Retail Dataset

5.2.4 Analyzing the Basic Metrics

retail_df.describe()

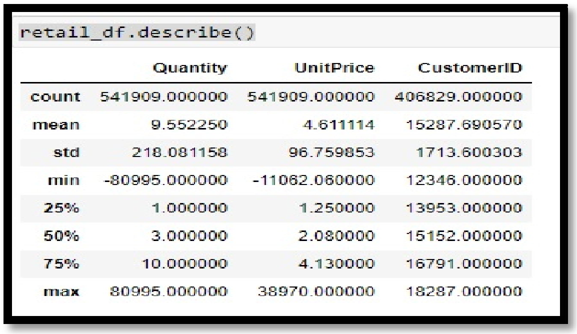
Figure 3 shows output of above command.

5.2.5 Detecting Invalid Data Type of Variables

All columns are assigned with appropriate data types based on value that they contain except CustomerID. It is good to declare CustomerID as int type.

5.2.6 Variable Transformations

CustomerID Column transformed from float type to int type



```
retail_df.describe()

           Quantity      UnitPrice     CustomerID
count  541909.000000  541909.000000  406829.000000
mean        9.552250       4.611114   15287.690570
std       218.081158      96.759853    1713.600303
min    -80995.000000  -11062.060000   12346.000000
25%         1.000000       1.250000   13953.000000
50%         3.000000       2.080000   15152.000000
75%        10.000000       4.130000   16791.000000
max     80995.000000   38970.000000   18287.000000
```

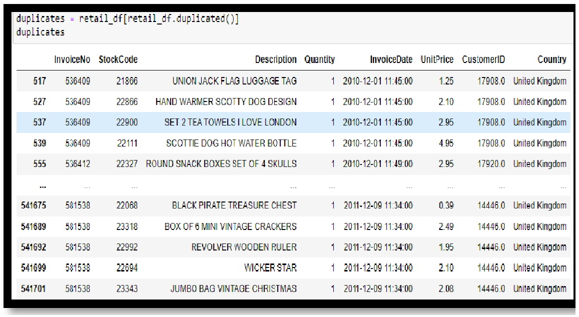
retail_df_ye_cust_analysis2=retail_df_ye_cust_analysis.astype({'CustomerID':'int32'})

Figure 3: Basic Statistics with Retail Dataset

5.2.7Removing Duplicate Rows

We can verify duplicate records using following command.
duplicates = retail_df [retail_df.duplicated()]

duplicates



```
duplicates = retail_df[retail_df.duplicated()]
duplicates
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
517	536409	21866	UNION JACK FLAG LUGGAGE TAG	1	2010-12-01 11:45:00	1.25	17908.0	United Kingdom
527	536409	22866	HAND WARMER SCOTTY DOG DESIGN	1	2010-12-01 11:45:00	2.10	17908.0	United Kingdom
537	536409	22900	SET 2 TEA TOWELS I LOVE LONDON	1	2010-12-01 11:45:00	2.95	17908.0	United Kingdom
539	536409	22111	SCOTTIE DOG HOT WATER BOTTLE	1	2010-12-01 11:45:00	4.95	17908.0	United Kingdom
555	536412	22327	ROUND SNACK BOXES SET OF 4 SKULLS	1	2010-12-01 11:49:00	2.95	17920.0	United Kingdom
...	...	...	...	...	...	...	...	...
541675	581538	22068	BLACK PIRATE TREASURE CHEST	1	2011-12-09 11:34:00	0.39	14446.0	United Kingdom
541689	581538	23318	BOX OF 6 MINI VINTAGE CRACKERS	1	2011-12-09 11:34:00	2.49	14446.0	United Kingdom
541692	581538	22992	REVOLVER WOODEN RULER	1	2011-12-09 11:34:00	1.95	14446.0	United Kingdom
541699	581538	22694	WICKER STAR	1	2011-12-09 11:34:00	2.10	14446.0	United Kingdom
541701	581538	23343	JUMBO BAG VINTAGE CHRISTMAS	1	2011-12-09 11:34:00	2.08	14446.0	United Kingdom

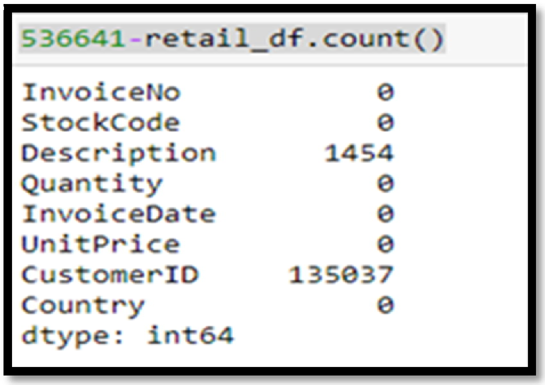
Figure 4: Basic Statistics with Retail Dataset

There are 5268 duplicates records out of 541909 rows. Wecan remove them using the following command.
retail_df = retail_df.drop_duplicates ()

5.2.8Missing Value Treatment

536641-retail_df.count ()

Figure 5 shows output of above command.



```
536641-retail_df.count()

InvoiceNo            0
StockCode            0
Description       1454
Quantity             0
InvoiceDate          0
UnitPrice            0
CustomerID      135037
Country              0
dtype: int64
```

Figure 5: Figure Shows Missing Values in DescriptionandCustomerIDColumn

Observation: Only two variables Description and CustomerID have missing values. We can delete the missing values for Description Column because there are no corresponding CustomerID and UnitPrice values for them. The following commands removes all missing values.

retail_df.dropna (subset = ['Description'], inplace=True)

5.2.9Checking Null Values

CustomerID column has null values. The following command displays records with CustomerID as null value

retail_df [retail_df ['CustomerID'].isnull ()]

5.2.10 Outlier Treatment

We have outliers in **UnitPrice** and **Quantity** column Boxplot[3] can display outliers in a column. Figure 6 shows outliers in Quantity Column. These outliers can be eliminated using z-score (threshold value).Eliminating outliers gives more accurate results in analysis.

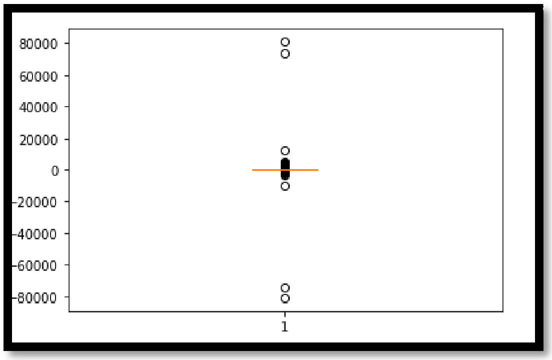


Figure 6: Boxplot Shows Outliers in Quantity Column

5.2.11Dimensionality Reduction

In this step two copies of retail_dataset are created. They are

retail_df_cust_analysis_1: This dataset contains data of customers whose data is available completely. Records with CustomerID as null are removed.
retail_df_cust_analysis = retail_df.copy ()
retail_df_cust_analysis.dropna (subset = ['CustomerID'], inplace=True)
retail_df_cust_analysis.shape
(401604, 8)

retail_df_cust_analysis_2: This dataset contains all customers' data, including records with CustomerID as null. This dataset is used if CustomerID is not used in the analysis.

retail_df_cust_analysis_2= retail_df.copy()
retail_df_cust_analysis_2.shape

(535187, 8)

5.3 Data Analysis

5.3.1 Univariate Analysis

Univariate analysis shows distribution of data points in the column. We have various visualization techniques [4] to perform univariate analysis. Figure 7 shows Histogram on Quantity column.

bins = np.linspace(0,2000,100)
groupby_inv=pd.DataFrame(retail_df_cust_analysis_2[~(retail_df_cust_analysis_2['InvoiceNo'].astype('str').str.contains('C'))]
.groupby('InvoiceNo')['StockCode'].
groupby_inv['Number of products'].plot.hist(bins = bins, figsize=(15, 7), color='brown', xticks=bins)

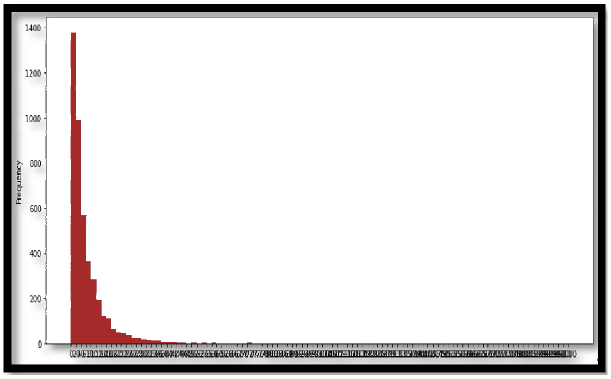


Figure 7: Figure Shows Histogram on QuantityColumn

Observation: Histogram on Quantity column shows that most of the Customers buy less than 20 items.

5.3.2 Bivariate Analysis

Bivariate Analysis shows relationship among columns in the dataset.

```
r_c=retail_df_cust_analysis_1.groupby ('Country')
r_c_r=r_c['Revenue'].sum()
r_c_r2=r_c_r.reset_index().sort_values(by=['Revenue'],ascending=False)
plt.scatter(r_c_r2['Country'],r_c_r2['Revenue'])
plt.xticks(r_c_r2['Country'], rotation='vertical')
plt.show()
```

Observation: Scatter plot between Revenue and Country columns is shown in Figure 8.It shows that most of the Revenue generated from customers in United Kingdom.

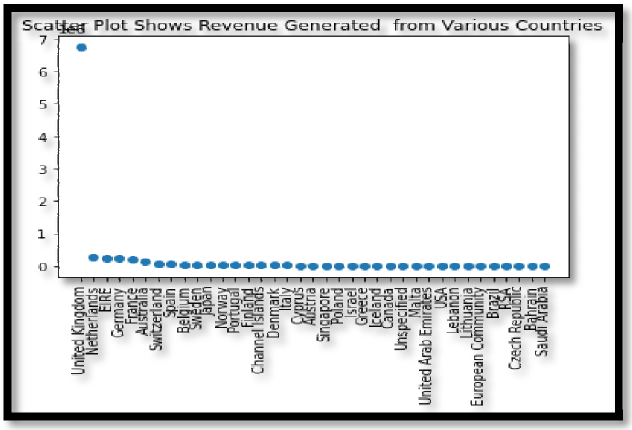


Figure 8: ScatterPlot Shows Revenue from Various Countries

5.3.3 Correlation Analysis

Heat map gives correlation matrix, where each cell in the matrix shows the correlation between two columns.

Figure 9 shows Heat map.

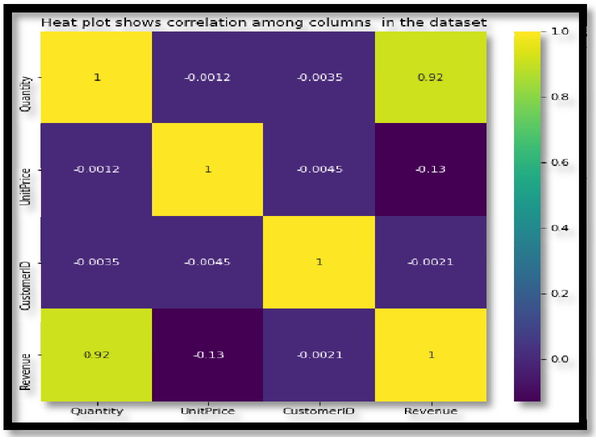


Figure 9: Figure Shows Heat Map on Retail_dataset.

```
corr_mat=retail_df_cust_analysis_1.corr()
plt.figure(figsize=(8,8))
sns.heatmap(corr_mat,annot=True,cmap='viridis')
plt.title("Heat plot shows correlation among columns  in the dataset").
```

Observation: Heat map shows strong correlation between Quantity and Revenue.

5.4 Data exploration

This is the heart of entire EDA process. Data exploration isa story telling processofasking relevant sequence of questions. These questions reveal patterns related to customer's consumption behaviors and preferences.

5.4.1 Analysis Based on Customer Transactions

*i. **How many Cancelled Orders do we have?***

InvoiceNo starting with the letter "C" is treated as Cancelled Order.

```
cancelled_orders=retail_df_cust_analysis_1
[retail_df_cust_analysis_1
['InvoiceNo'].astype('str').str.contains('C')]
```

```
cancelled_orders.shape
(8872, 9)
```

We have 8872 cancelled orders

*ii. **What is the Percentage of Cancelled Orders***

```
total_orders = retail_df_cust_analysis_1['InvoiceNo'].nunique()
can_orders = cancelled_orders['InvoiceNo'].nunique()
can_orders*100/total_orders
```

```
16.466876971608833
```

Percentage of cancelled orders are: **16.466876971608833**

Observations: We have almost 16% Cancelled orders which is a pretty big number for online retailer. Studying these cancelled orders may help in preventing future cancellation.

Let's first get an overview of the general customers purchase behavior and then dig deeper.

iii. What's the average number of orders per customer?

```
groupby_cust=
pd.DataFrame(retail_df_cust_analysis_1.groupby('CustomerID'
)['InvoiceNo'].nunique()).groupby_cust['InvoiceNo'].mean()
```

5.07548032936871

Observations: The average number of orders per customer is 5. As we found in descriptive statistics, customers buy an average (mean) quantity of 5.

Are they from the same product? Let's examine how many products are purchased.

5.4.2 Analysis Based On Products/Items

i. ***What's the average number of unique items per order?***

```
groupby_inv=pd.DataFrame(retail_df_cust_analysis_2[~(retail
_df_cust_analysis_2['InvoiceNo'].astype('str').str.contains('C'))]
.groupby('InvoiceNo')['StockCode'].nunique())
groupby_inv.median()
```

Observation: Number of unique items per order: 15.0

ii. How many products does a customer buy on an average?

Observations: Figure 7 shows skewed distribution of products. It shows that most people buy less than 20 items.

5.4.3 Analysis Based on Revenue

i. What is the total revenue generated by the online retailer?

```
retail_df_cust_analysis_1['Revenue']=
retail_df_cust_analysis_1['Quantity']*retail_df_cust_analysis_1
['UnitPrice']
retail_df_cust_analysis_1['Revenue'].sum()
8278519.4240000015
```

ii. What is the average revenue per customer?

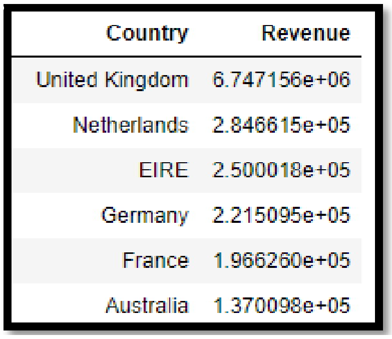
```
groupby_cust =
pd.DataFrame(retail_df_ye_cust_analysis.groupby('CustomerI
D')['Revenue'].sum())

groupby_cust['Revenue'].mean()
1893.5314327538888
```

Observations: In average, 1893.5 revenue is generated per customer

iii. What is the total revenue per country?

```
groupby_country=
pd.DataFrame(retail_df_cust_analysis_2.groupby('Country')['R
evenue'].sum())
groupby_country
```

Figure 10 shows output of above command



Country	Revenue
United Kingdom	6.747156e+06
Netherlands	2.846615e+05
EIRE	2.500018e+05
Germany	2.215095e+05
France	1.966260e+05
Australia	1.370098e+05

Figure 10: Figure Shows Revenue from Various Countries.

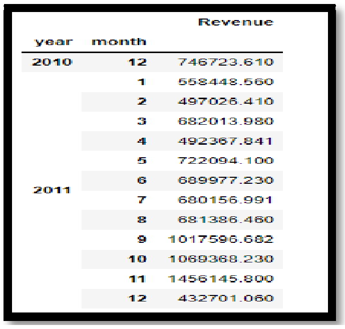
Observations: As we can see, the largest market is the one located in **UK.**So, we can conclude not only most sales revenues are achieved in UK, but also most customers are located there too. We can explore this to find more about what products the customers buy together and what possible future opportunities are there in the UK Market.

iv. What is the monthly revenue of the online store?

Revenue (monthly) = Monthly Invoice Count * Quantity * Unit Price

```
groupby_month=                                                =
pd.DataFrame(retail_df_cust_analysis_2.groupby(['year','mont
h'])['Revenue'].sum())
groupby_month
```

Figure 11 shows output of above command.



		Revenue
year	month	
2010	12	746723.610
2011	1	558448.560
	2	497026.410
	3	682013.980
	4	492367.841
	5	722094.100
	6	689977.230
	7	680156.991
	8	681386.460
	9	1017596.682
	10	1069368.230
	11	1456145.800
	12	432701.060

Figure 11: Figure Shows Year- Month wise Revenue

Observations:

- Monthly revenue for the month of **September**, **October**, and **November** are pretty good.
- The reason being, as we have **Halloween**, **Black Friday**, and **Thanksgiving** sales coming up around these months so the customers tend to buy more sort of gifts.

v. What is the monthly growth rate for the online retail store?

```
groupby_month['Growth']=
groupby_month['Revenue'].pct_change()*100

groupby_month[['Revenue']].plot.line(figsize=(15,7),
color='green', fontsize=13, linestyle='-.')
plt.xlabel('time')
plt.ylabel('Revenue')
plt.title('Line chart')
```

Figure 12 shows output of above command.

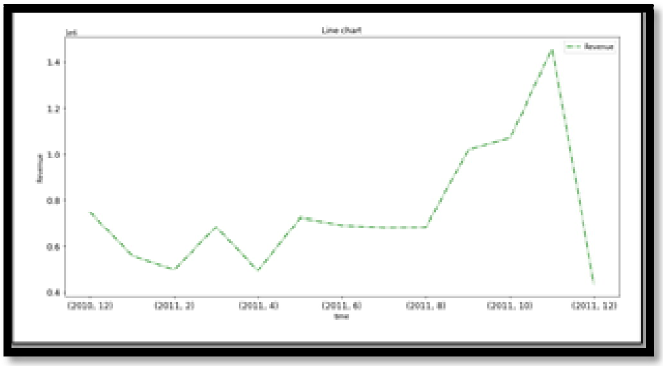


Figure 12: Figure Shows Monthly Growth Rate for the Online Retail Store

Observations: It seems like the growth rate of online store is fluctuating**.** There is no stagnant growth over the months.

5.4.4 Analysis Based on Geographical Location

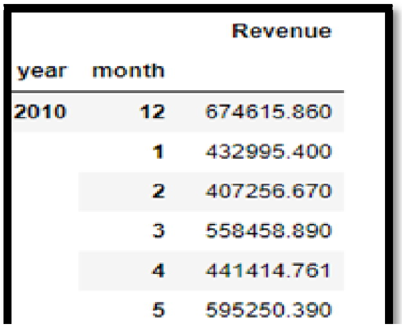
i. What is the average monthly revenue in UK?

```
groupby_month_uk=
pd.DataFrame(retail_df_cust_analysis_2[retail_df_cust_analysis_2['Country']=='United Kingdom'].groupby(['year','month'])['Revenue'].sum())
groupby_month_uk
```

ii. Which products are most bought in UK?

```
retail_uk =
retail_df_cust_analysis_2[retail_df_cust_analysis_2['Country']=='United Kingdom']
```

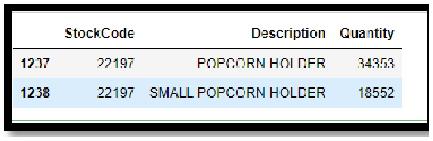
```
group_by_stock_desc =
pd.DataFrame(retail_uk.groupby(['StockCode','Description'],
as_index = False)['Quantity'].sum())
#most bought products in uk
temp_df = group_by_stock_desc.sort_values(by = 'Quantity',
ascending = False)
```



		Revenue
year	month	
2010	12	674615.860
	1	432995.400
	2	407256.670
	3	558458.890
	4	441414.761
	5	595250.390

Figure 13: Figure Shows Monthly Revenue in UK.

```
temp_df[temp_df['StockCode'] == 22197]
```



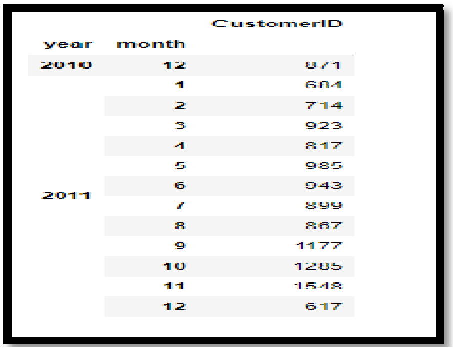
	StockCode	Description	Quantity
1237	22197	POPCORN HOLDER	34353
1238	22197	SMALL POPCORN HOLDER	18552

Figure 14: Figure Shows Product Most People Buy in UK.

Observations: Popcorn holder is the most frequently ordered item.

iii. How many monthly active customers are there from UK?

```
groupby_month_uk =
pd.DataFrame(retail_uk.groupby(['year','month'])['CustomerID'].nunique())
groupby_month_uk
```



		CustomerID
year	month	
2010	12	871
2011	1	684
	2	714
	3	923
	4	817
	5	985
	6	943
	7	899
	8	867
	9	1177
	10	1285
	11	1548
	12	617

Figure 15: Figure Shows Active Customers Year-Moth Wise

Actionable Insights

- By analyzing the data in this way, we can uncover groups of customers that behave in similar ways. This level of customer segmentation is useful in precision marketing.
- A marketing campaign that works for a group of customers that places low value orders frequently may not be suitable for customers who place sporadic high value orders.
- Make relevant product recommendations to the customers using precision marketing.
- Empower your customers to actively share their details, encourage them to share their data with you through conversations, surveys, and other methods. Doing so not only help you to know them better, but it also builds trust.
- Additionally, before performing analysis it would be important to talk with the e-commerce team to understand the business and its customers and its strategic and tactical objectives.

6. CONCLUSION

This paper clearly explained in detail about explorative data analysis. This paper mainly focused on significance of EDA and also systematic approach i.e. sequence of steps to be followed to extract interesting hidden pattern in the dataset. Here we have taken retail dataset as case study to implement Task Centric EDA. We applied EDA on this dataset to identify the potential characteristics of different customer categories and put forward appropriate precision marketing strategies to improve sales.Python programming language with Jupyter Note Book are used to analyze data and to draw various charts. At last the outcome of EDA produces conclusion remarks and actionable insights to improve business process.

REFERENCES

7.1 Journal Articles

1. Tristan Langer and Tobias Meisen,**System Design to UtilizeDomain Expertise for Visual Exploratory Data Analysis,** Information 2021.
2. Jinglin Peng,Weiyuan Wu,Brandon Lockhart,Song Bian,Jing Nathan Yan Linghao Xu,Zhixuan Chi,Jeffrey M. Rzeszotarski,Jiannan Wang ,**DataPrep.EDA: Task-Centric Exploratory Data Analysis for Statistical Modeling in Python.**
3. Babangida Ibrahim Babura,Mohd Bakri Adam2,Muhammad Sani3,Usman Waziri4,Felix Yakubu Eguda1,**Construction and Applications of Stairboxplot for Exploratory Data Analysis, Journal of Physics**: Conference Series, ICMSDS 2020.
4. Parvatham Niranjan Kumar, Kambhampati Vijay Kumar,**Comparative Study of Univariate Data Visualization with Case Study Approach**, JAC : A Journal Of Composition Theory, Volume XIV, Issue V, MAY 2021,pp-82-92
5. J.Rajendra, Prasad, S.SaiKumar, B.V.SubbaRao,**Design and Development of Financial Fraud Detection using Machine Learning**,International Journal of Emerging Trends in Engineering Research, Volume 8. No.9, September 2020, pp.5838 – 5843
6. Pujo Hari Saputro,Herlino Nanang,**Exploratory Data Analysis & Booking Cancelation Prediction on Hotel Booking Demands Data,** Journal of Applied Data Sciences Vol. 2, No. 1, January 2021, pp. 40.
7. Behrens, J. T. (1997). **Principles and procedures of exploratory data analysis.** *Psychological Methods*, *2*(2), 131–160. https://doi.org/10.1037/1082-989X.2.2.131.
8. Smith, A. F., & Prentice, D. A. (1993). **Exploratory data analysis.** In G. Keren & C. Lewis (Eds.), A handbook for data analysis in the behavioral sciences: Statistical issues (p. 349–390). Lawrence Erlbaum Associates, Inc.
9. EstelleCamizuli, Emmanuel John Carranza, **Exploratory Data Analysis (EDA),** First published: 26 November,2018,https://doi.org/10.1002/9781119188230.saseas0271
10. Ren Jie Tan,**A Starter Pack to Exploratory Data Analysis with Python,** pandas, seaborn, and scikit-learn
11. Jitendra Pramanik,Abhaya Kumar Samal,Kabita Sahoo,Dr. Subhendu Kumar Pani,Exploratory **Data Analysis using Python,**International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-12, October 2019.
12. I.J. **Good,The Philosophy of Exploratory Data Analysis**, Philosophy of Science Volume 50, Number 2 ,Jun-1983.

7.2 Books

13. **Exploratory Data Analysis with MATLAB**
By Wendy L. Martinez, Angel R. Martinez, Jeffrey Solka.
14. **Exploratory data analysis as a foundation of inductive research**,Andrew T.Jebb,Scott Parrigon,Sang EunWoo,Human Resource Management Review,Volume 27, Issue 2, June 2017, Pages 265-276.
15. **Hands-On Exploratory Data Analysis with Python: Perform EDA**,Suresh Kumar Mukhiya, Usman Ahmed,2020,PACKT publishing Ltd.
16. **Exploratory Data Analysis Using R**,Ronald K. Pearson 2018, CRC Press.
17. **Hands-On Exploratory Data Analysis with R,**Radhika Datar, Harish Garg · May-2019, PACKT publishing Ltd.
18. **Exploratory Data Analysis -** Volume 2 **-** Page 1, John Wilder Tukey • 1977