SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur-603203

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR: 2024-2025

ODD SEMESTER

LAB MANUAL

(REGULATION - 2023)

DS3265 - DATA VISUALIZATION LABORATORY

SECOND SEMESTER

M.Tech-Data Science

Prepared By

Mrs.G.Santhiya, AP(Sr.G)/IT

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

- To understand the need of different datasets visualization
- Can perform descriptive and inferential analysis on data sets
- Can create visualization using modern tool
- Can develop analytical products
- Identify opportunities for application of data visualization in various domains

PROGRAMME OUTCOMES (POs)

After going through the four years of study, Information Technology Graduates will exhibit ability to:

PO#	Graduate Attribute	Programme Outcome
1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.
2	Problem analysis	Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4	Conduct investigations of complex problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling to complex engineering activities, with an understanding of the limitations.
6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10	Communication	Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations,and give and receive clear instructions
11	Project management and finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one‟s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments
12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES (PSOs)

At the end of the course, the student should be able to:

- Design and create data visualizations.
- Conduct exploratory data analysis using visualization.
- Craft visual presentations of data for effective communication.
- Use knowledge of perception and cognition to evaluate visualization design alternatives.
- Use JavaScript with D3.js to develop interactive visualizations for the Web.

OBJECTIVES

- To understand the need of different datasets visualization
- Can perform descriptive and inferential analysis on data sets
- Can create visualization using modern tool
- Can develop analytical products
- Identify opportunities for application of data visualization in various domains

LIST OF EXPERIMENTS

1. Identification of different types of data on the dataset
2. Visualization of datasets in terms of Line Chat, Area Chart, Bar Chart, Pie Chart, histogram, scatterplot, violin plot etc.
3. Representation of datasets and performing various statistical operation
4. Interactive Development of dashboard
5. Creating Visualization with Tableau
6. Presenting the working of dataset with Pivot Tables and Pivot Chart
7. Making World Map interaction with D3.js and SVG
8. Analysis of Variance (ANOVA)
9. Analysis of Covariance(ANCOVA)
10. Statistics associated with cluster Analysis
11. Design and development of Analytical products

TOTAL: 45 PERIODS

OUTCOME

At the end of the course, learners will be able to:

- ❖ Design and create data visualizations.
- ❖ Conduct exploratory data analysis using visualization.
- ❖ Craft visual presentations of data for effective communication.
- ❖ Use knowledge of perception and cognition to evaluate visualization design alternatives.
- ❖ Use JavaScript with D3.js to develop interactive visualizations for the Web.

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

HARDWARE

- ✓ Standalone desktops 30 Nos.

SOFTWARE

- ✓ **Openstack**
- ✓ **Hadoop**
- ✓ **Eucalyptus or Open Nebula or equivalent**

COURSE OUTCOMES

1908705.1	Configure various virtualization tools such as Virtual Box, VMware workstation
1908705.2	Design and deploy a web application in a PaaS environment.
1908705.3	Learn how to simulate a cloud environment to implement new schedulers.
1908705.4	Install and use a generic cloud environment that can be used as a private cloud.
1908705.5	Install and use Hadoop

CO-PO MATRIX

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1908705.1	3				3							
1908705.2	2		3		3							
1908705.3	1			2		2						
1908705.4			3		3							
1908705.5	1				2	2						
Average	**2**		**3**	**2**	**3**	**2**						

CO-PSO MATRIX

	PSO1	PSO2	PSO3	PSO4
1908705.1		2		
1908705.2		3	2	
1908705.3			2	
1908705.4		2	2	
1908705.5		2		
Average		**2**	**2**	

EVALUATION PROCEDURE FOR EACH EXPERIMENTS

S.No	Description	Mark
1.	Aim & Pre-Lab discussion	20
2.	Observation	20
3.	Conduction and Execution	30
4.	Output & Result	10
5.	Viva	20
Total		**100**

INTERNAL ASSESSMENT FOR LABORATORY

S.No	Description	Mark
1.	Observation	10
2.	Record	5
3.	Model Test	5
Total		**20**

SOFTWARE DESCRIPTION

- ✓ **Openstack**

OpenStack software controls large pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API. OpenStack works with popular enterprise and open source technologies making it ideal for heterogeneousinfrastructure.

- ✓ **Hadoop**

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

- ✓ **Eucalyptus or Open Nebula orequivalent**

OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures. The OpenNebula platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service.

HARDWARE

- ✓ Standalone desktops 30Nos

Ex No: 1. IDENTIFICATION OF DIFFERENT TYPES OF DATA ON THE DATASET

```
import pandas as pd import numpy as np

df=pd.read_csv("C:\\Users\\SRMVEC\Desktop\sales_data_types.csv")
   df['2016']+df['Jan Units']

df.dtypes

df.info()

df['Customer Number'].astype('int')

df["Customer Number"]=df['Customer Number'].astype('int')

df.dtypes

df['2016'].astype('float')

df['Jan Units'].astype('int')

df['Active'].astype('bool')


#DEFINITION VARIABLE

def convert_currency(val) : new_val=val.replace(',',").replace('$',") return
   float(new_val)

df['2016'].apply(convert_currency)

df['2016'].apply(lambda x:x.replace('$',").replace(',',")).astype('float')


df['2016']=df['2016'].apply(convert_currency)
   df['2017']=df['2017'].apply(convert_currency) df.dtypes

df['Percent Growth'].apply(lambda x:x.replace('%',")).astype('float')/100
```

df["Active"]=np.where(df["Active"]=="Y", True,False) df.dtypes

OUTPUT:

0 $125,000.00500

1 $920,000.00700

2 $50,000.00125

3 $350,000.0075

4 $15,000.00

Closed dtype: object

Customer Number int64

Customer Name object

2016 object

2017 object

Percent Growth object

Jan Units object

Month int64

Day int64

Year int64

Active object

dtype: object

<class 'pandas.core.frame.DataFrame'> RangeIndex: 5 entries, 0 to 4

Data columns (total 10 columns):

Column Non-Null Count Dtype

0 Customer Number 5 non-null int64

1 Customer Name 5 non-null object

dtypes: int64(4), object(6) memory usage: 528.0+ bytes

0 10002

1 552278

2 23477

3 24900

4 651029

Name: Customer Number, dtype: int32

Customer Number int32

Customer Name object

2016 object

2017 object

Percent Growth object

Jan Units object

Month int64

Day int64

Year int64

Active object dtype: object

STRING TO FLOAT

could not convert string to float: '$125,000.00'

INTEGER

invalid literal for int() with base 10: 'Closed'

0 True

1 True

2 True

3 True

4 True

Name: Active, dtype: bool

0 125000.0

1 920000.0

2 50000.0

3 350000.0

4 15000.0

Name: 2016, dtype: float64

Customer Number int32

Customer Name object

2016 float64

2017 float64

Percent Growth object

Jan Units object

Month int64

Day int64

Year int64

Active object

dtype: object

0 0.30

1 0.10

2 0.25

3 0.04

4 -0.15

Name: Percent Growth, dtype: float64

Customer Number int32

Customer Name object

2016 float64

2017 float64

Percent Growth object

Jan Units object

Month int64

Day int64

Year int64

Active bool

dtype: object

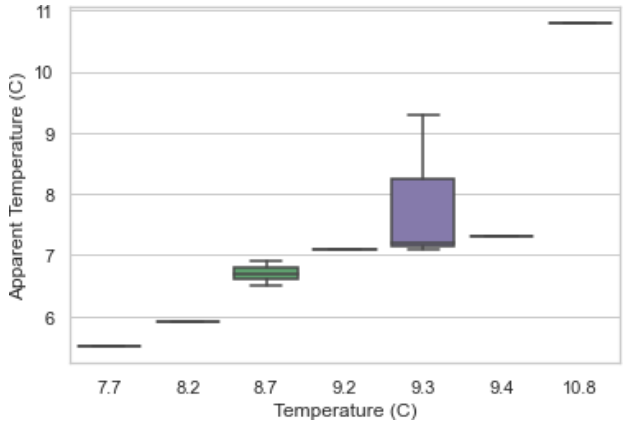
Ex No: 2. VISUALIZATION OF DATASETS IN TERMS OF LINE CHAT, AREACHART, BAR CHART, PIE CHART, HISTOGRAM, SCATTERPLOT,VIOLIN PLOT ETC

VIOLIN PLOT():

import pandas as pd import seaborn

seaborn.set(style='whitegrid')

ds = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
seaborn.boxplot(x="Temperature (C)", y="Apparent Temperature (C)", data=ds)



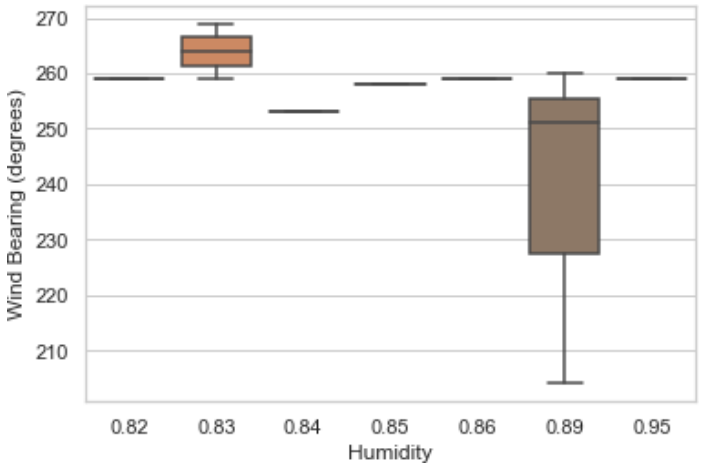
BOXPLOT():

import pandas as pd import seaborn

seaborn.set(style='whitegrid')

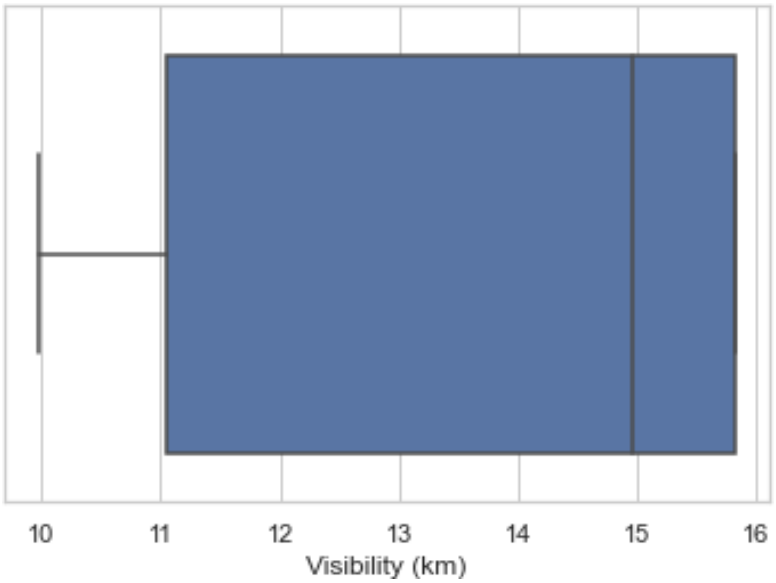
tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')

seaborn.boxplot(x='Humidity', y='Wind Bearing (degrees)', data=tip)



GROUPING VARIABLES IN SEABORN BOXPLOT WITH DIFFERENT ATTRIBUTES:

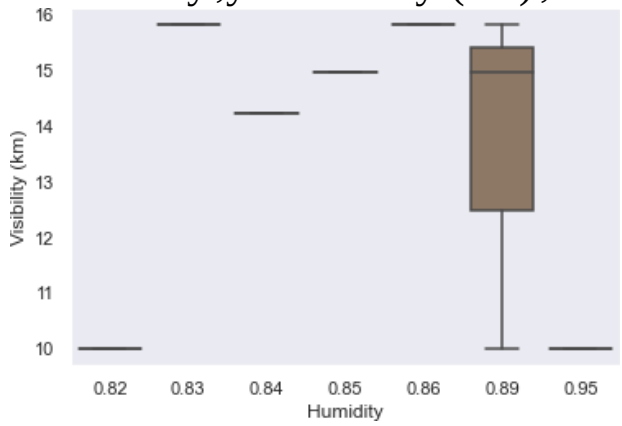
```
import pandas as pd import seaborn

seaborn.set(style="whitegrid")

tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
seaborn.boxplot(x =tip['Visibility (km)'])
```



DRAW HORIZONTAL BOXPLOT:

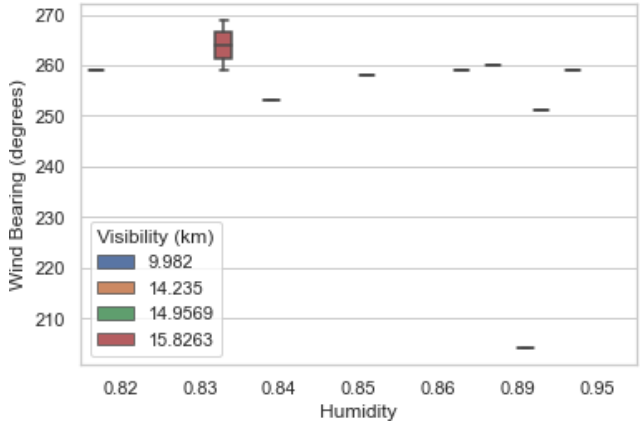
```
import pandas as pd import seaborn seaborn.set(style="dark")
tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
```

seaborn.boxplot(x='Humidity',y='Visibility (km)',data=tip)



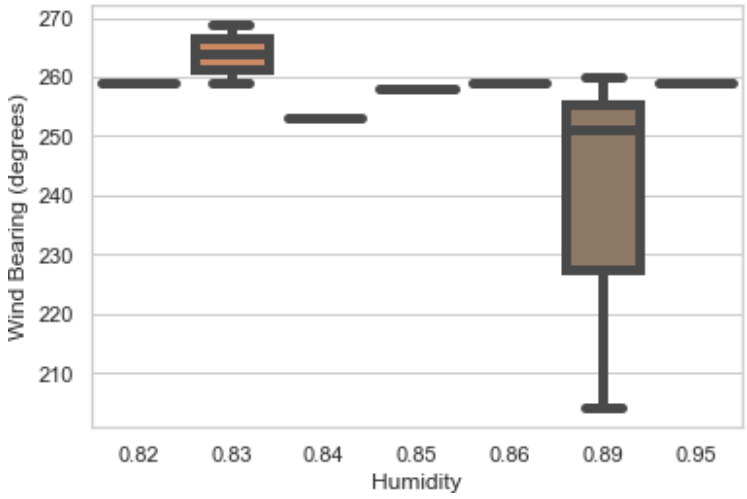
USING HUE PARAMETER:

import pandas as pd import seaborn
seaborn.set(style="whitegrid")
tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
seaborn.boxplot(x ="Humidity", y ="Wind Bearing (degrees)", hue ="Visibility (km)", data = tip)



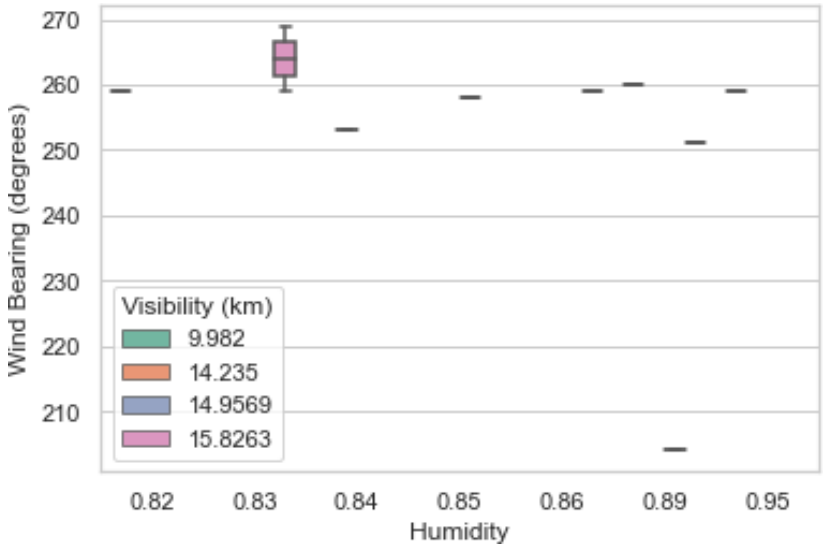
DRAW OUTLINES AROUND THE DATA POINTS USING LINEWIDTH:

import seaborn seaborn.set(style="whitegrid")
tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
seaborn.boxplot(x = 'Humidity', y = 'Wind Bearing (degrees)', data = tip, linewidth=5)



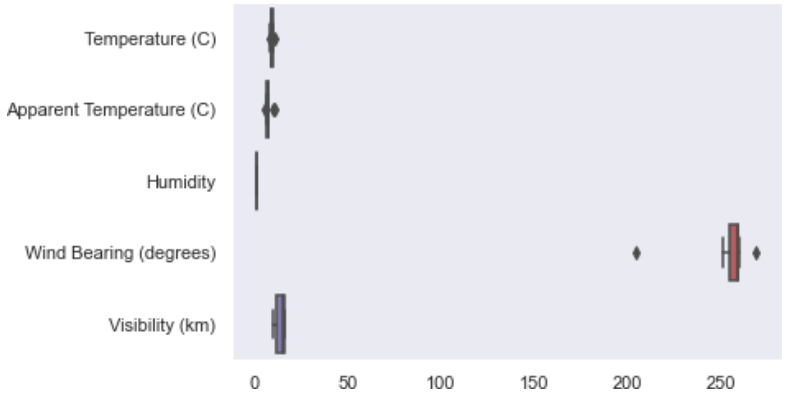
DRAW EACH LEVEL OF THE HUE VARIABLE AT DIFFERENT LOCATIONS ON THE MAJOR CATEGORICAL AXIS:

```
import pandas as pd import seaborn
seaborn.set(style="whitegrid")
tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
seaborn.boxplot(x="Humidity", y="Wind Bearing (degrees)", hue="Visibility (km)",
data=tip, palette="Set2",dodge=True)
```



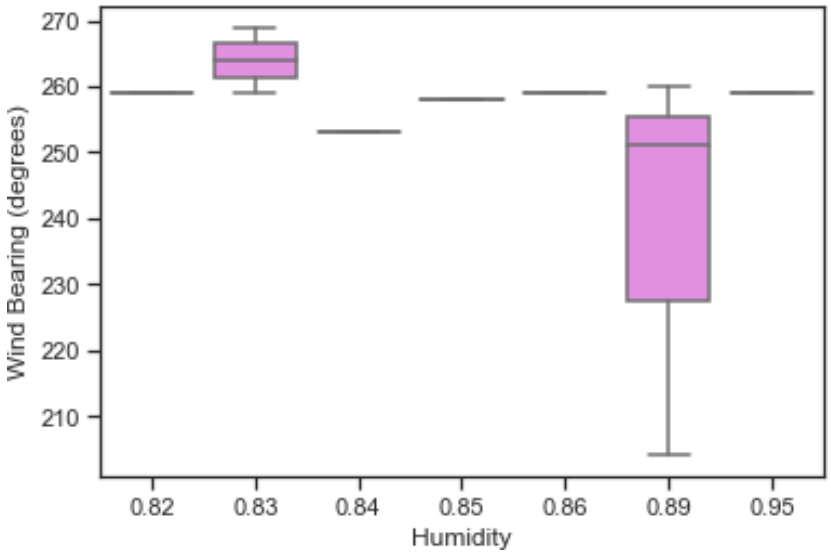
CONTROL ORIENTATION OF THE PLOT (VERTICAL OR HORIZONTAL):

```
import pandas as pd
import seaborn
seaborn.set(style="dark")
tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
seaborn.boxplot(data = tip, orient="h")
```



USING COLOR ATTRIBUTES FOR COLOR FOR ALL THE ELEMENTS:

```
import pandas as pd
import seaborn
seaborn.set(style="ticks")
tip = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
seaborn.boxplot(x = 'Humidity', y = 'Wind Bearing (degrees)', data = tip, color =
"violet")
```




```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df =pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
print(df.head())
plt.figure(figsize=(10, 8))
sns.boxplot(x='Humidity', y='Wind Bearing (degrees)', data=df)
plt.ylabel("Wind Bearing (degrees)", size=14)
plt.xlabel("Humidity", size=14)
plt.title("weather", size=18)
plt.figure(figsize=(10, 8))
sns.boxplot(x='Wind Bearing (degrees)', y='Humidity', data=df, showmeans=True)
plt.ylabel("Humidity", size=30)
plt.xlabel("Wind Bearing (degrees)", size=50)
plt.title("weather", size=40)
```

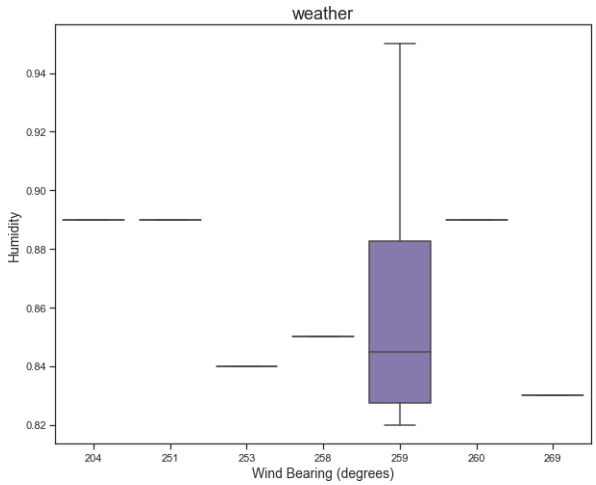
OUTPUT:

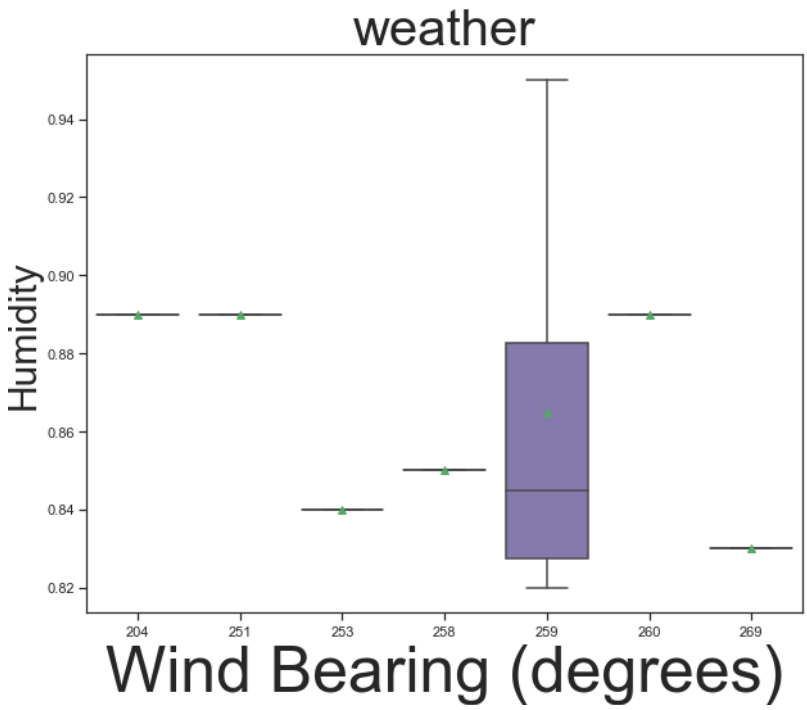
```
Text(0.5, 1.0, 'weather')
Date  Summary Precip Type Temperature (C) \
```

0	01-04-2006	Partly Cloudy	rain	9.4
1	02-04-2006	Partly Cloudy	rain	9.3
2	03-04-2006	Mostly Cloudy	rain	9.3
3	04-04-2006	Partly Cloudy	rain	8.2
4	05-04-2006	Mostly Cloudy	rain	8.7

	Apparent Temperature (C)	Humidity	Wind Bearing (degrees)	Visibility (km)
0	7.3	0.89	251	15.8263
1	7.2	0.86	259	15.8263
2	9.3	0.89	204	14.9569
3	5.9	0.83	269	15.8263
4	6.9	0.83	259	15.8263

Text(0.5, 1.0, 'weather')




weather
Humidity
0.94
0.92
0.90
0.88
0.86
0.84
0.82
204
251
253
258
259
260
269
Wind Bearing (degrees)

Ex No: 3. REPRESENTATION OF DATASETS AND PERFORMING VARIOUS STATISTICAL OPERATION

```
import pandas as pd

dataset = pd.read_csv('C:\\Users\\SRMVEC\Desktop\weather.csv')
dataset.head()

# TO FIND MEAN

mean=dataset['Humidity'].mean() print(mean)

# TO FIND MEDIAN

median=dataset['Temperature (C)'].median() print(median)

# TO FIND MODE

mode=dataset['Precip Type'].mode() print(mode)

# TO FIND COUNT

count=dataset['Summary'].count() print(count)

# TO FIND STANDARD DEVIATION

std=dataset['Wind Speed (km/h)'].std() print(std)

# TO FIND MAXIMUM AND MINIMUM VALUE

maxvalue=dataset['Wind Speed (km/h)'].max() print(maxvalue)

minvalue=dataset['Wind Speed (km/h)'].min() print(minvalue)

# TO DESCRIBE THE DATASET

dataset.describe()
```

OUTPUT:

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)	Daily Summary
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251	15.8263	0	1015.13	Partly cloudy throughout the day.
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259	15.8263	0	1015.63	Partly cloudy throughout the day.
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	204	14.9569	0	1015.94	Partly cloudy throughout the day.
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	269	15.8263	0	1016.41	Partly cloudy throughout the day.
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	259	15.8263	0	1016.51	Partly cloudy throughout the day.

MEAN

0.7348989663358906

MEDIAN

12.0

MODE

0 rain

1 dtype: object

COUNT

96453

MAXIMUM VALUE

63.8526

MINIMUM VALUE

0.0

STANDARD DEVIATION

6.913571012592151

DESCRIBE

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
count	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.0	96453.000000
mean	11.932678	10.855029	0.734899	10.810640	187.509232	10.347325	0.0	1003.235956
std	9.551546	10.696847	0.195473	6.913571	107.383428	4.192123	0.0	116.969906
min	-21.822222	-27.716667	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
25%	4.688889	2.311111	0.600000	5.828200	116.000000	8.339800	0.0	1011.900000
50%	12.000000	12.000000	0.780000	9.965900	180.000000	10.046400	0.0	1016.450000
75%	18.838889	18.838889	0.890000	14.135800	290.000000	14.812000	0.0	1021.090000
max	39.905556	39.344444	1.000000	63.852600	359.000000	16.100000	0.0	1046.380000

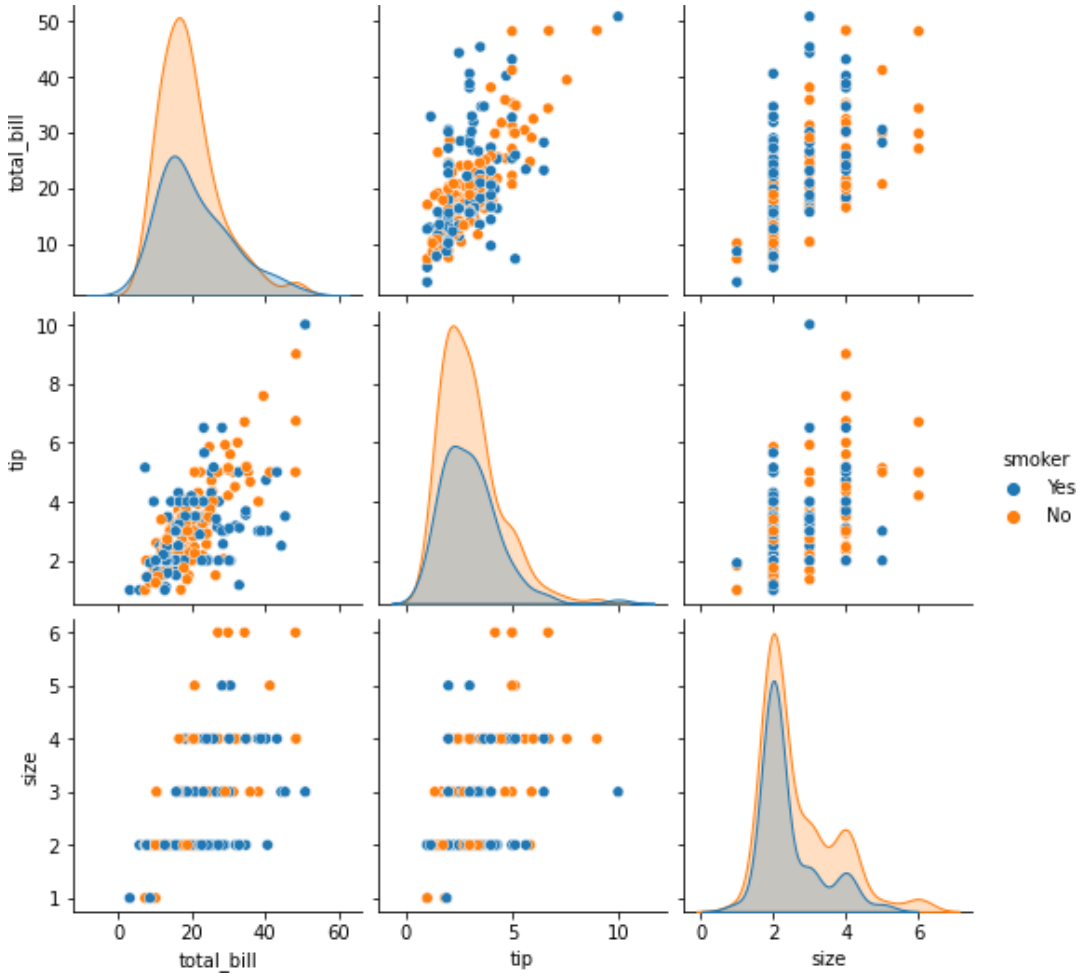
EX NO: 4. INTERACTIVE DEVELOPMENT OF DASHBOARD

```
%matplotlib inline
import seaborn as sns
from ipywidgets
import interact
tips=sns.load_dataset('tips')
tips.head()
@interact(hue=['smoker','sex','time','day'])
def plot(hue):
    a_=sns.pairplot(tips,hue=hue)
```

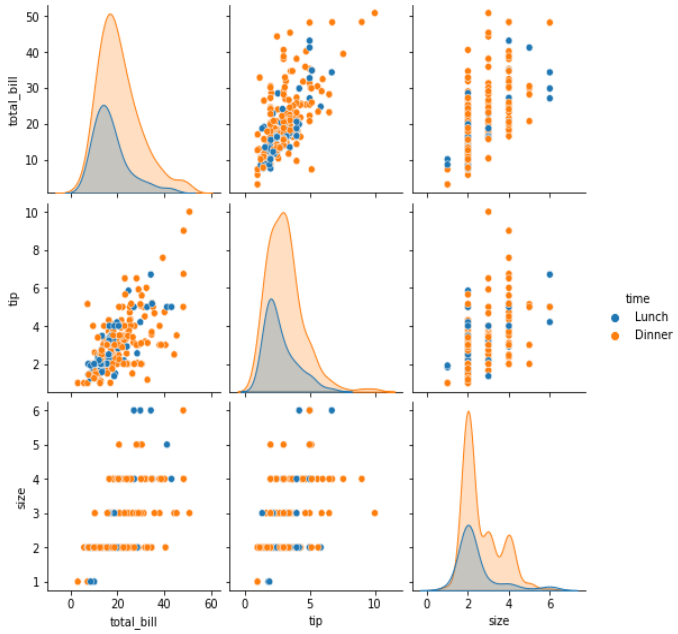
OUTPUT:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

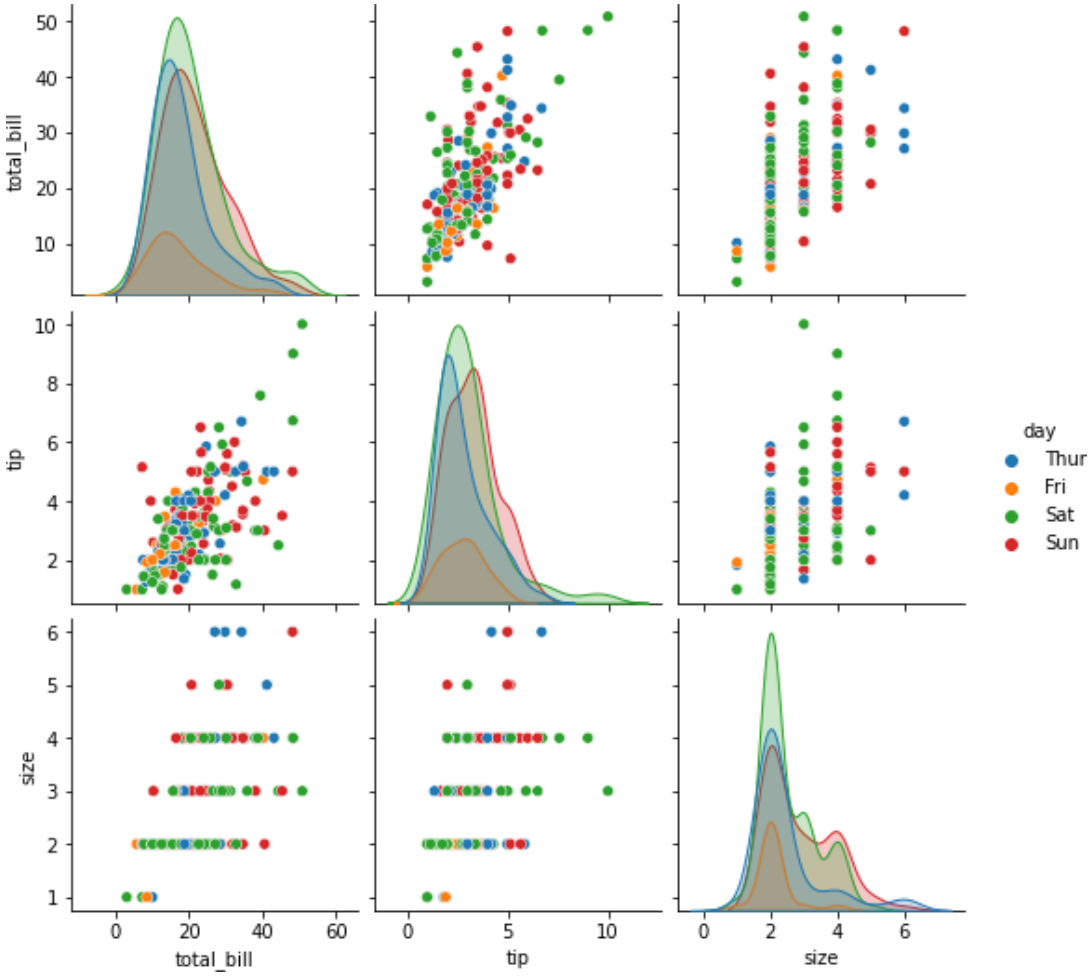
OUTPUT :
SMOKER


total_bill
tip
size
smoker
Yes
No

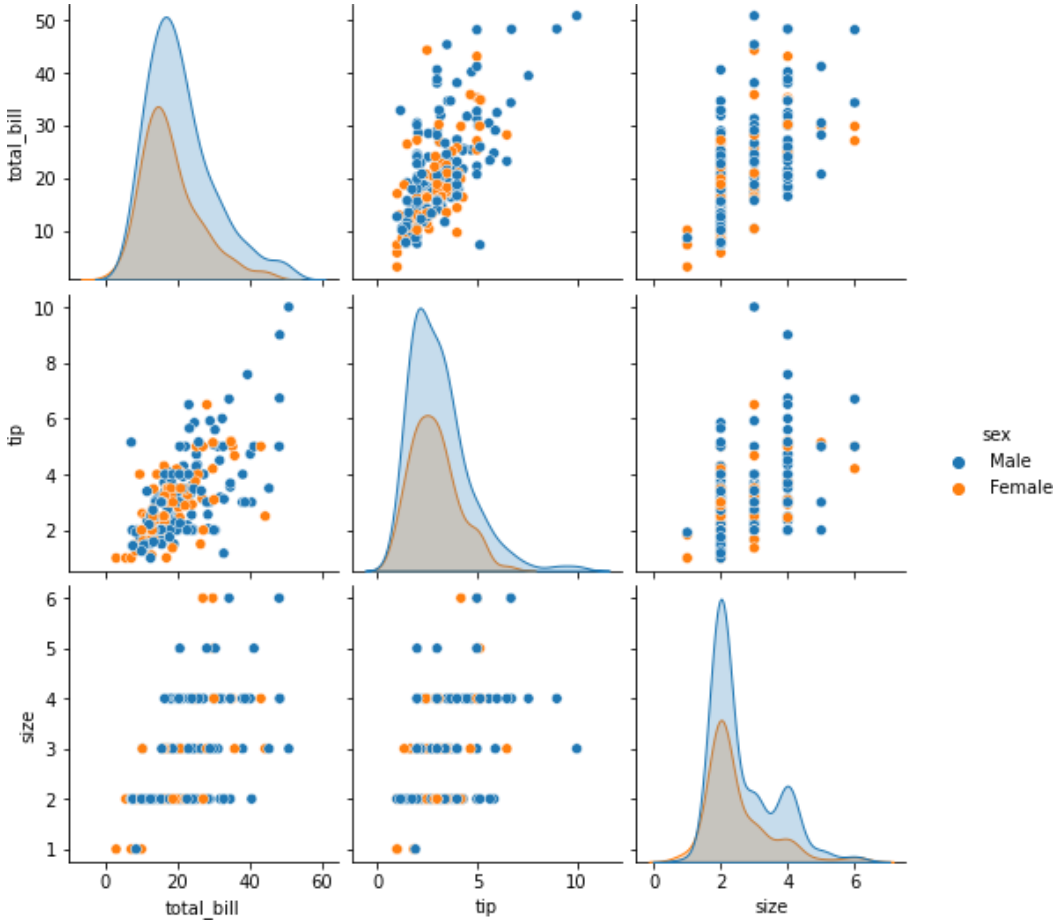
TIME


total_bill
tip
size
time
Lunch
Dinner

DAY



SEX

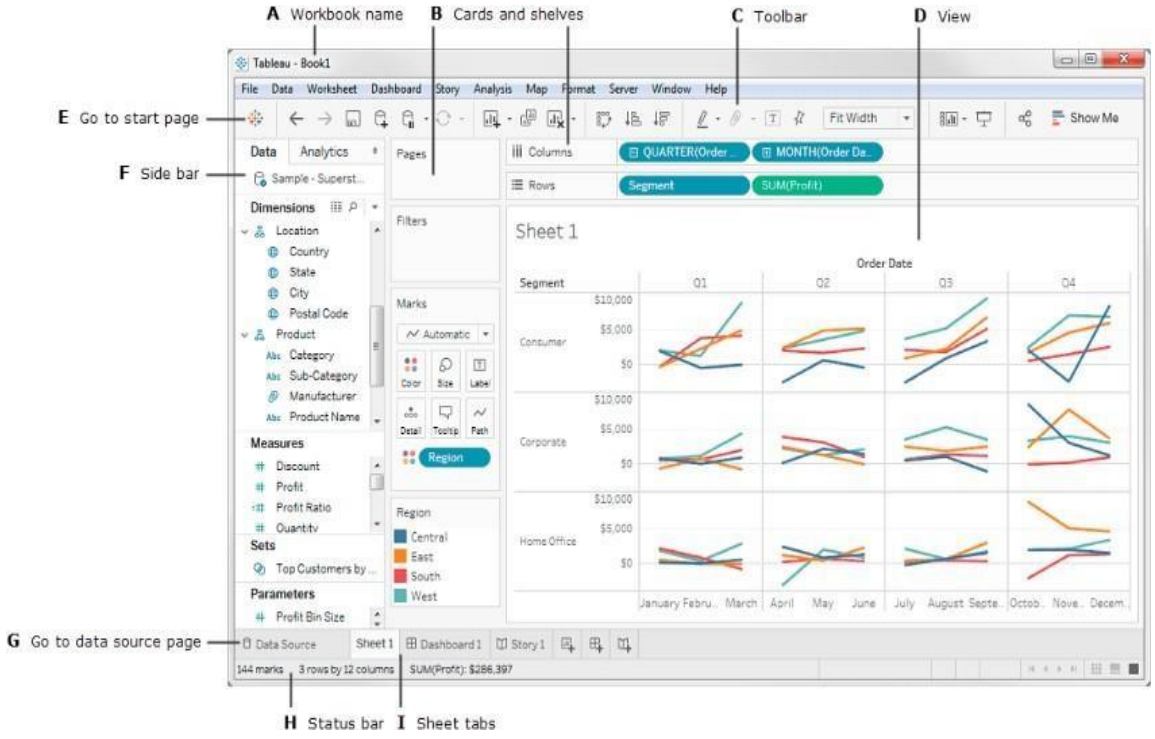


EX NO: 5. CREATING VISUALIZATION WITH TABLEAU

Tableau Software is a software company headquartered in Seattle, Washington that produces interactive data visualization products focused on business intelligence.

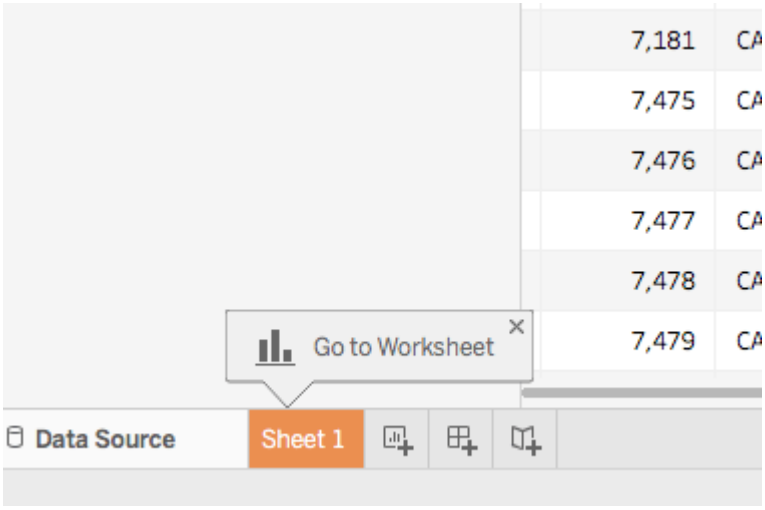
Tableau Workspace

The Tableau workspace is a collection of worksheets, menu bar, toolbar, marks card, shelves and a lot of other elements about which we will learn in sections to come. Sheets can be worksheets, dashboards, or stories. The image below highlights the major components of the workspace. However, more familiarity will be achieved once we work with actual data.



Steps

1. Go to the worksheet. Click on the tab Sheet 1 at the bottom left of the tableau workspace.



2. Once, you are in the worksheet, from Dimensions under the Data pane, drag the Order Date to the Column shelf.

On dragging the Order Date to the columns shelf, a column for each year of Orders is created in the dataset. An 'Abc' indicator is visible under each column which implies that text or numerical or text data can be dragged here. On the other hand, if we pulled Sales here, a cross-tab would be created which would show the total Sales for each year.

3. Similarly, from the Measures tab, drag the Sales field onto the Rows shelf.

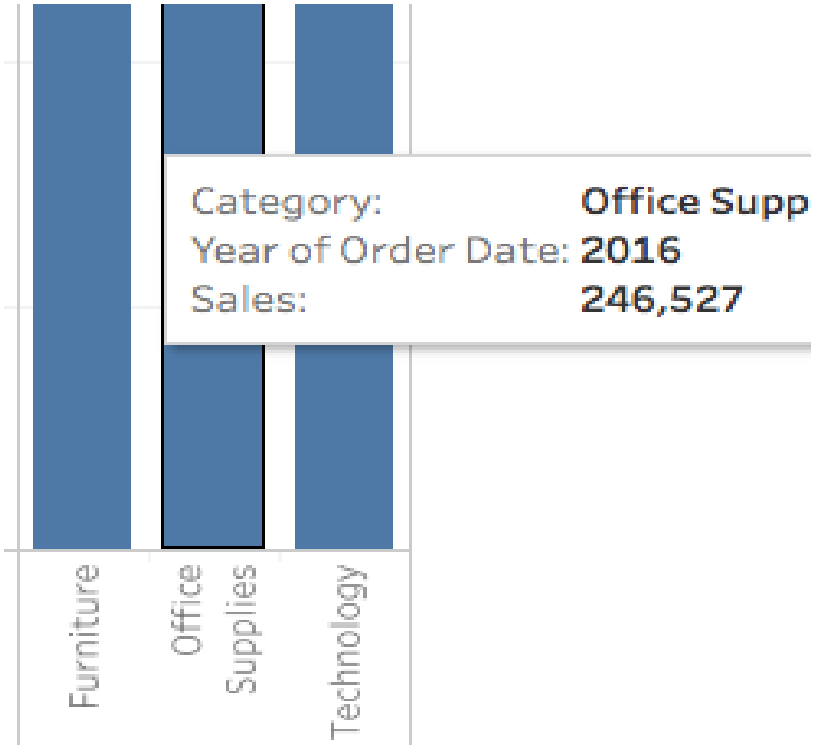
Tableau populates a chart with sales aggregated as a sum. Total aggregated sales for each year by order date is displayed. Tableau always populates a line chart for a view that includes time-field which in this example is Order Date.

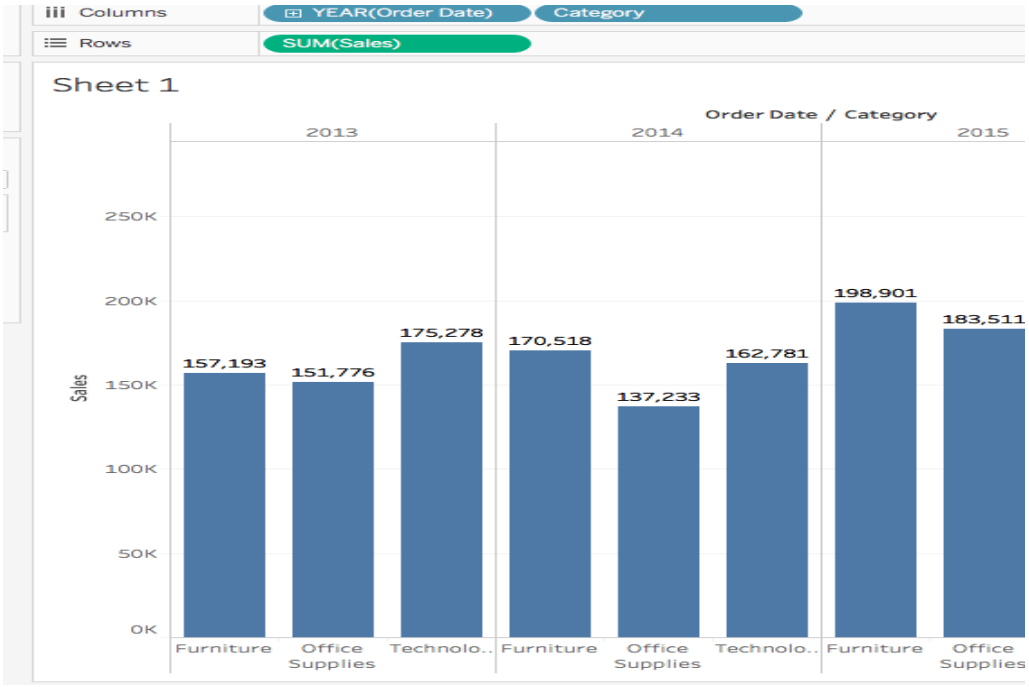
Steps

1. Category is present under the Dimensions pane. Drag it to the columns shelf and place it next to YEAR(Order Date). The Category should be placed to the right of Year. In doing so, the view immediately changes to a bar chart type from a line. The chart shows the overall Sales for every Product by year.

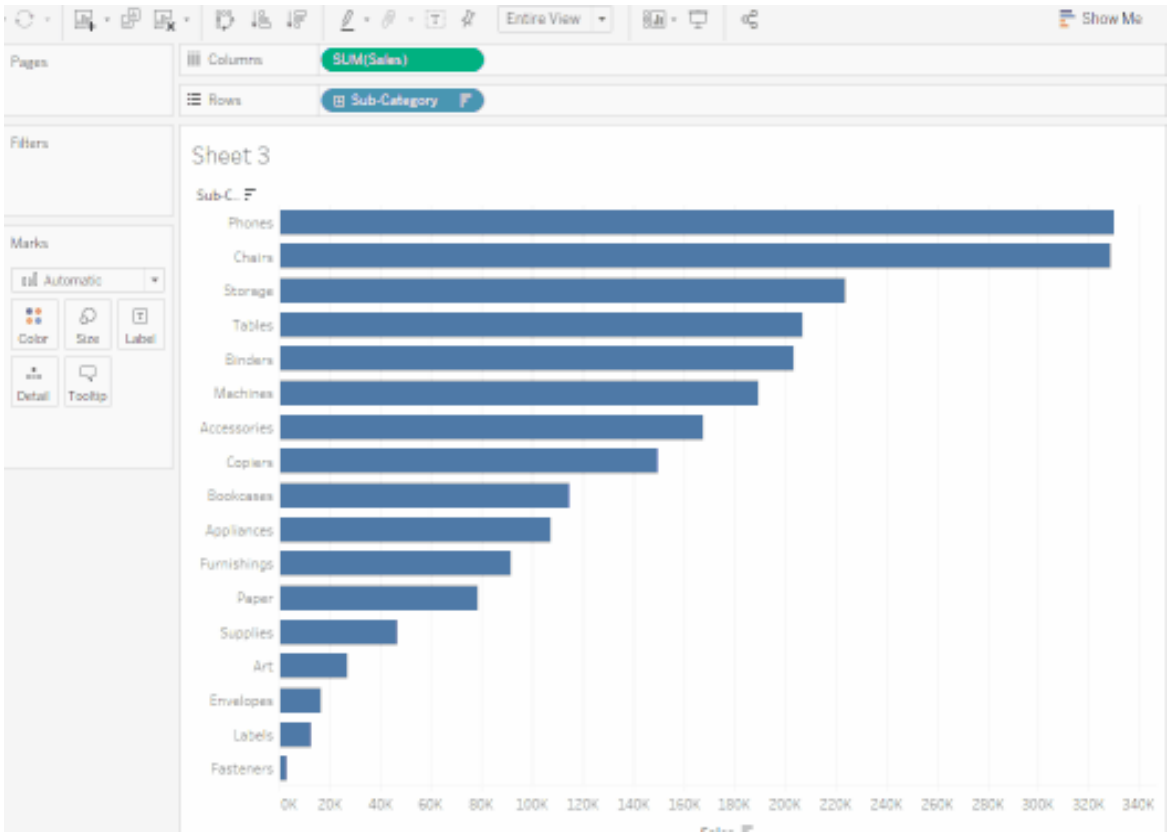
To view information about each data point (that is, mark) in the view, hover over one of the bars to reveal a tooltip. The tooltip displays total sales for that category.

Here is the tooltip for the Office Supplies category for 2016:





The bar chart can be displayed horizontallyinstead of vertically too. Click Swap on the toolbar for the same.



1. The view above nicely shows sales by category, i.e., furniture, office supplies, and technology. We can also infer that furniture sales are growing faster than sales of office supplies except for 2016. Hence it will be wise to focus sales efforts on furniture instead of office supplies. But furniture is a vast category and consists of many different items. How can we identify which furniture item is contributing towards maximum sales?

   To help us answer that question, we decide to look at products by Sub-category to see which items are the big sellers. Let's say for the Furniture category; we want to look at details about only bookcases, chairs, furnishings, and tables. We will Double-click or drag the Sub-Category dimension to the Columns shelf.

   The sub-category is another discrete field. It further dissects the Category and displays a bar for every sub-category broken down by category and year. However, it is a humongous amount of data to make sense of visually. In the next section, we will learn about filters, color and other ways to make the view more comprehensible.

1. Emphasizing the Results

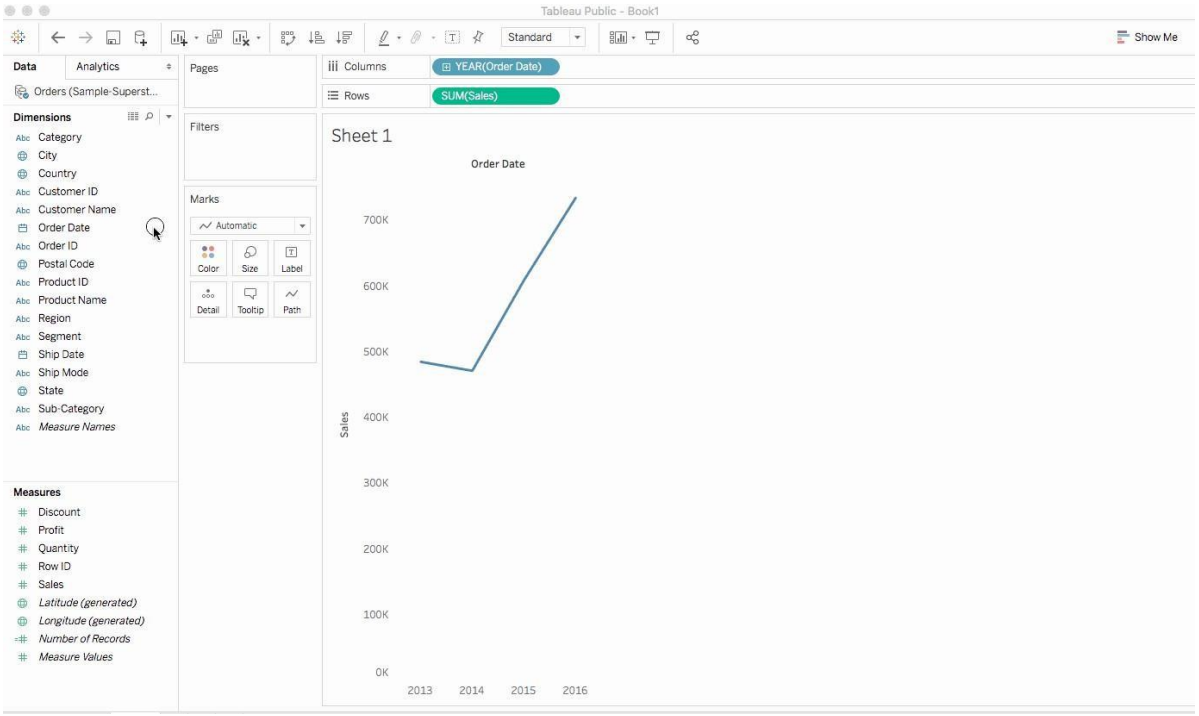
In this section, we will try to focus on specific results. Filters and colors are ways to add more focus to the details that interest us.

Adding filters to the view

Filters can be used to include or exclude values in the view. Here we try to add two simple filters to the worksheet to make it easier to look at product sales by sub-category for a specific year.

Steps

In the Data pane, under Dimensions, right-click Order Date and select Show Filter. Repeat for Sub->category field also.



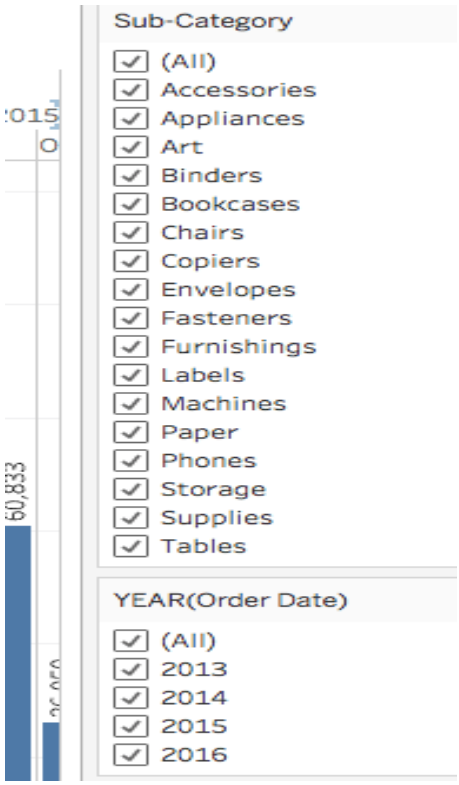
Filters are the type of cards and can be moved around on the worksheet by simple drag and drop

Adding colors to the view

Colors can be helpful in the visual identification of a pattern.

Steps

In the Data pane, under Measures, drag Profit to Color on the Marks card.


Sub-Category
(All)
Accessories
Appliances
Art
Binders
Bookcases
Chairs
Copiers
Envelopes
Fasteners
Furnishings
Labels
Machines
Paper
Phones
Storage
Supplies
Tables
YEAR(Order Date)
(All)
2013
2014
2015
2016

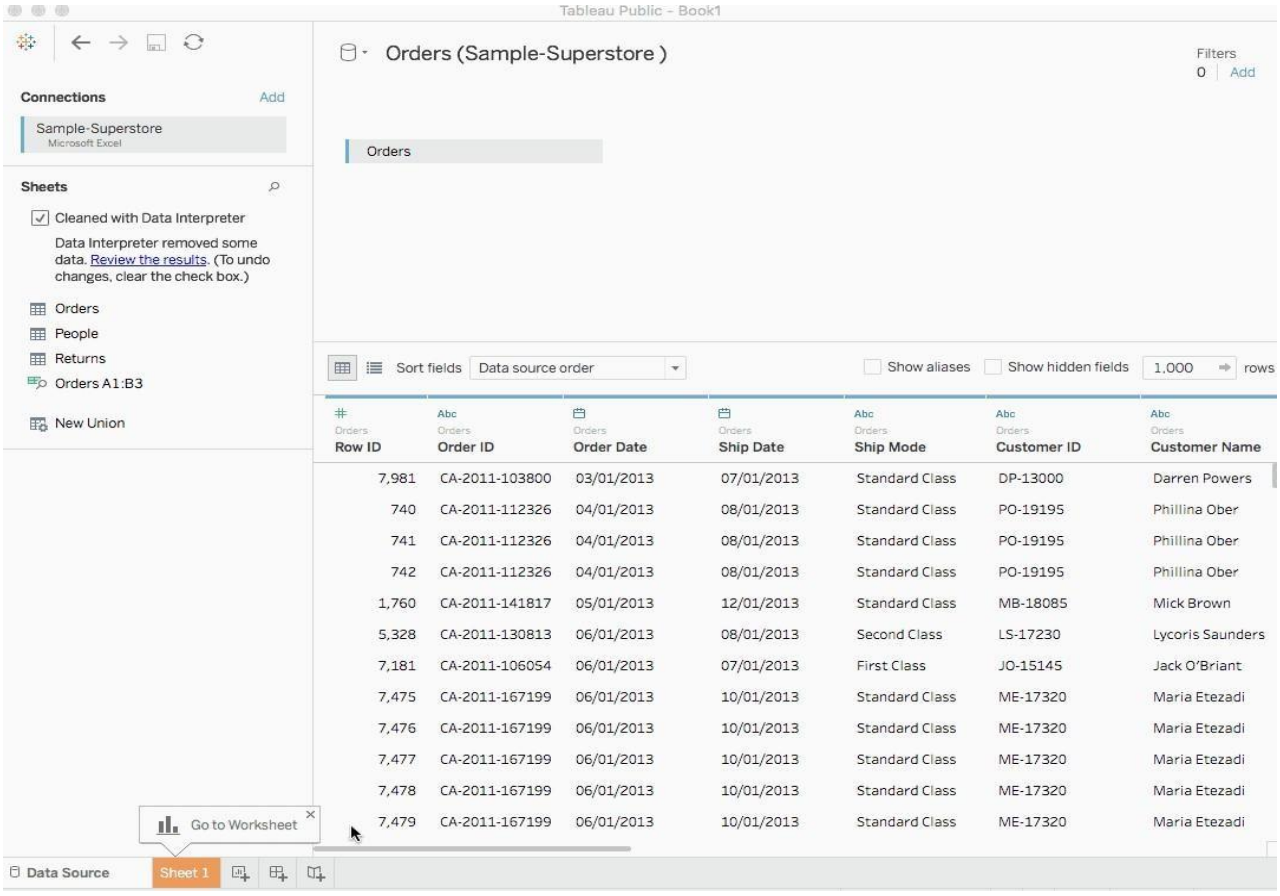


Tableau Public - Book1
Orders (Sample-Superstore)
Filters
0 Add
Connections Add
Sample-Superstore
Microsoft Excel
Sheets
Cleaned with Data Interpreter
Data Interpreter removed some data. Review the results. (To undo changes, clear the check box.)
Orders
People
Returns
Orders A1:B3
New Union
Orders
Sort fields Data source order
Show aliases Show hidden fields 1,000 rows
Row ID | Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name
7,981 | CA-2011-103800 | 03/01/2013 | 07/01/2013 | Standard Class | DP-13000 | Darren Powers
740 | CA-2011-112326 | 04/01/2013 | 08/01/2013 | Standard Class | PO-19195 | Phillina Ober
741 | CA-2011-112326 | 04/01/2013 | 08/01/2013 | Standard Class | PO-19195 | Phillina Ober
742 | CA-2011-112326 | 04/01/2013 | 08/01/2013 | Standard Class | PO-19195 | Phillina Ober
1,760 | CA-2011-141817 | 05/01/2013 | 12/01/2013 | Standard Class | MB-18085 | Mick Brown
5,328 | CA-2011-130813 | 06/01/2013 | 08/01/2013 | Second Class | LS-17230 | Lycoris Saunders
7,181 | CA-2011-106054 | 06/01/2013 | 07/01/2013 | First Class | JO-15145 | Jack O'Briant
7,475 | CA-2011-167199 | 06/01/2013 | 10/01/2013 | Standard Class | ME-17320 | Maria Etezadi
7,476 | CA-2011-167199 | 06/01/2013 | 10/01/2013 | Standard Class | ME-17320 | Maria Etezadi
7,477 | CA-2011-167199 | 06/01/2013 | 10/01/2013 | Standard Class | ME-17320 | Maria Etezadi
7,478 | CA-2011-167199 | 06/01/2013 | 10/01/2013 | Standard Class | ME-17320 | Maria Etezadi
7,479 | CA-2011-167199 | 06/01/2013 | 10/01/2013 | Standard Class | ME-17320 | Maria Etezadi
Go to Worksheet
Data Source
Sheet 1

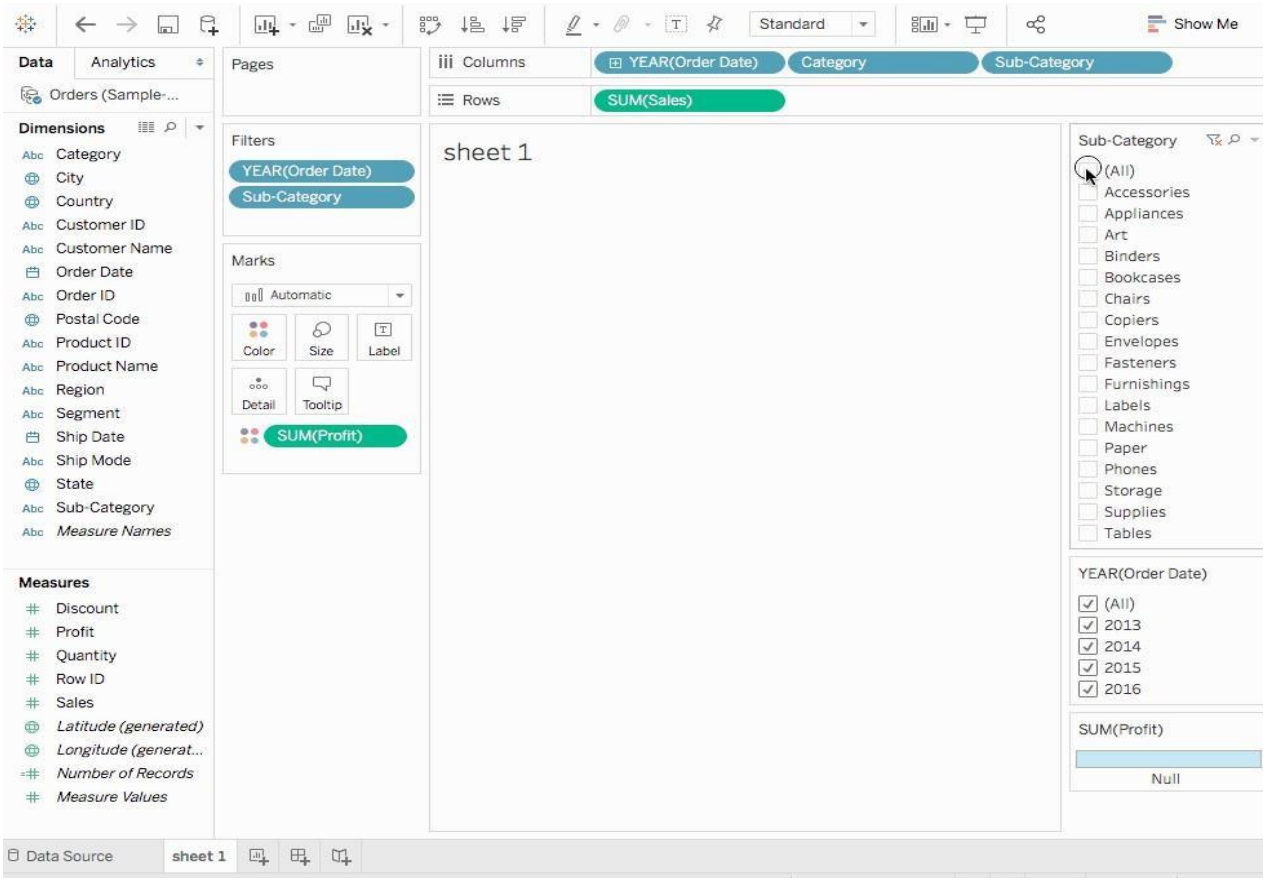
Map View

Creating a Map View

Map views are beneficial when we are looking at geographic data (the Region field). In the current example, Tableau automatically recognizes that the Country, State, City, and Postal Code fields contain geographical information.

Steps

1. Create a new worksheet.
2. Add State and Country under Data pane to Detail on the Marks card. We obtain the map view.
3. Drag Region to the Filters shelf, and then filter down to South only. The map view now zooms in to the South region only, and a mark represents each state.
4. Drag the Sales measure to the Color tab on the Marks card. We obtain a filled map with the colors showing the range of sales in each state.
5. We can change the color scheme by clicking Color on the Marks card and selecting Edit Colors. We can experiment with the available palettes.
6. We observe that **Florida** is performing the best regarding Sales. If we Hover over Florida, it shows a total of 89,474 USD in sales, as compared to South Carolina, for example, which has only 8,482 USD in sales. Let us gauge the performance by Profit now since Profit is a better indicator than Sales alone.
7. Drag Profit to Color on the Marks card. We now see that Tennessee, North Carolina, and Florida have negative profit, even though it appeared they were doing good in Sales. Rename the sheet as Profit Map



Getting into the Details

Maps empower us to visualize the data broadly. In the last step, we discovered that we discovered that Tennessee, North Carolina, and Florida have a negative profit. In this section let us draw a Bar chart to explore the reason for the negative profit.

Steps

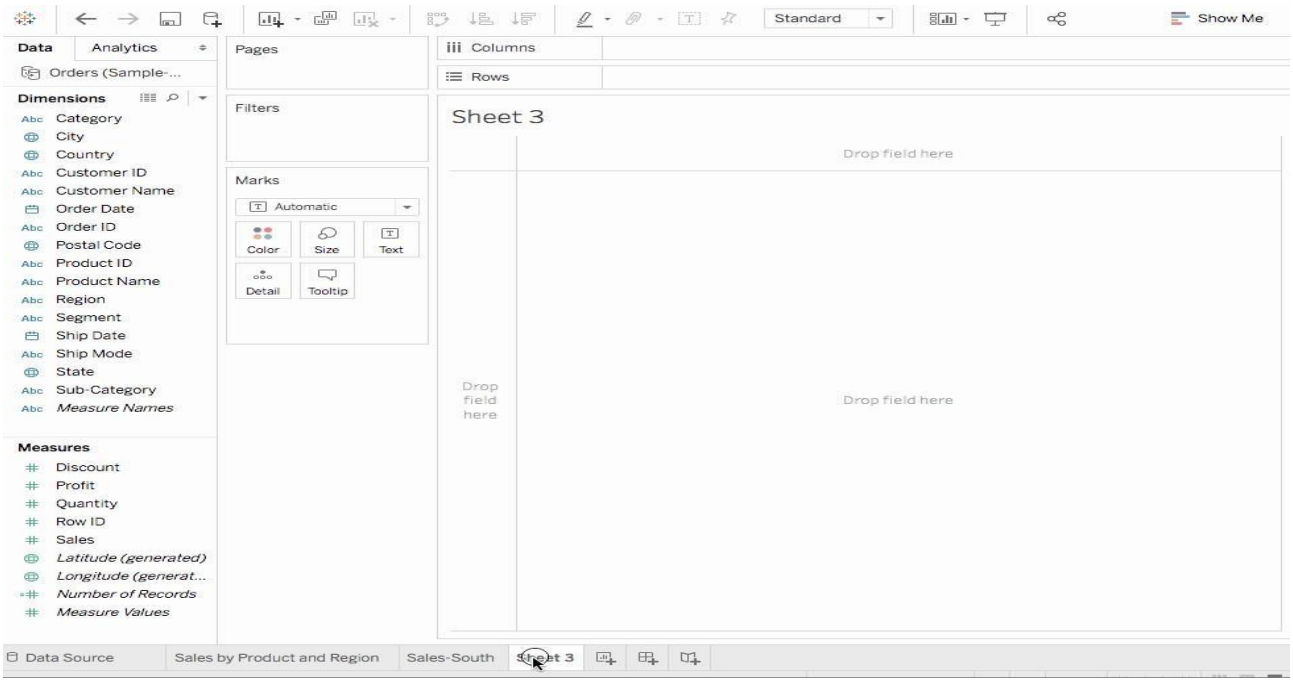
1. Duplicate the Profit Map worksheet and name it Negative Profit Bar Chart.
2. Click Show Me on the **Negative Profit Bar Chart** worksheet. Show Me presents the number of ways in which a graph can be plotted between items mentioned in the worksheet. From Show Me select the horizontal bar option and the view updates to horizontal from vertical bars instantly.
3. We can select more than one bar at a time by simply clicking and dragging the

cursor over them. We want to focus only on the three states, i.e., Tennessee, North Carolina, and Florida. Hence, we will only select the bars pertaining to them.

CreatingHierarchies

Hierarchies come in handy when we want to group similar fields so that we can quickly drill down between levels in the viz.

1. In the Data pane, drag a field and drop it directly on top of another field or right-click the field and select
2. Drag any additional fields into the hierarchy. Fields can also be re- ordered in the hierarchy by simply dragging them to a new position. In the current viz. we will create the following hierarchies: Location, Order,andProduct.



4. On the Rows Shelf, click the plus-shaped icon on the State Field to drill- down to the City level.

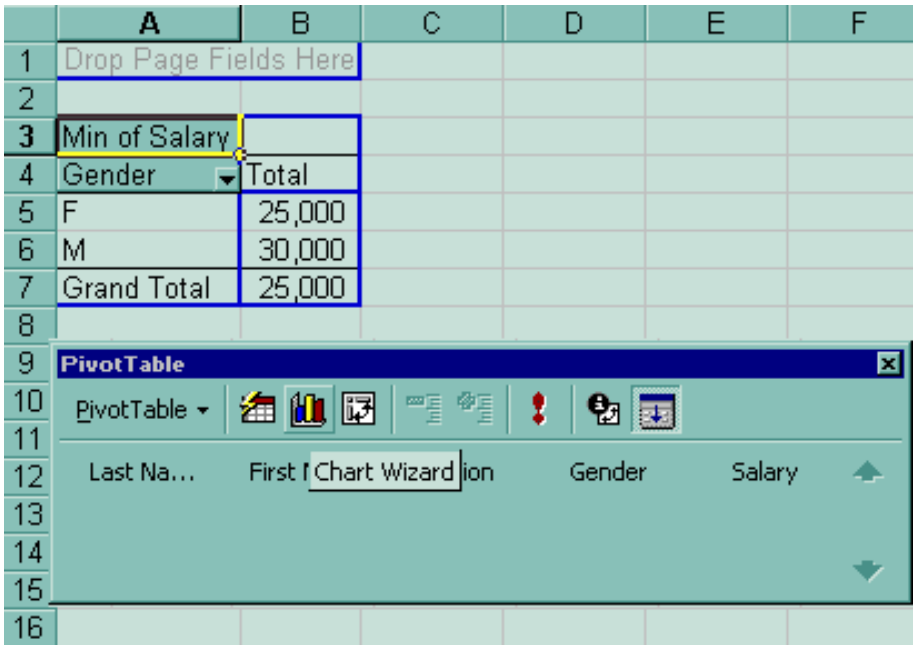
EX NO: 6. PRESENTING THE WORKING OF DATASET WITH PIVOT

TABLES AND PIVOT CHART

Pivot tables can be created and used to manipulate data. Although pivot tables allow us to analyze our data, they‟re not the most appealing way of displaying theresults of this analysis. The best way to present our findings is through the use ofa chart, specifically a pivot chart. With pivot charts, we can perform the same operations we did with pivot tables, but display the results in a manner that provides more impact.

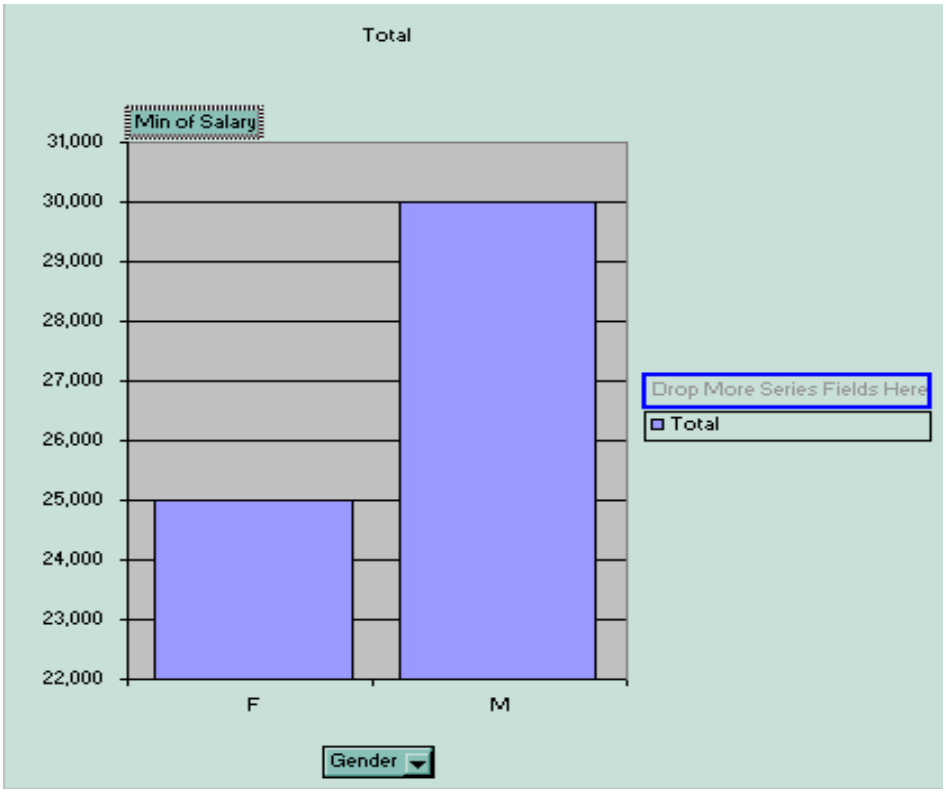
Creating a pivot table and pivot chart

First, select the Chart Wizard button in the PivotTable toolbar.



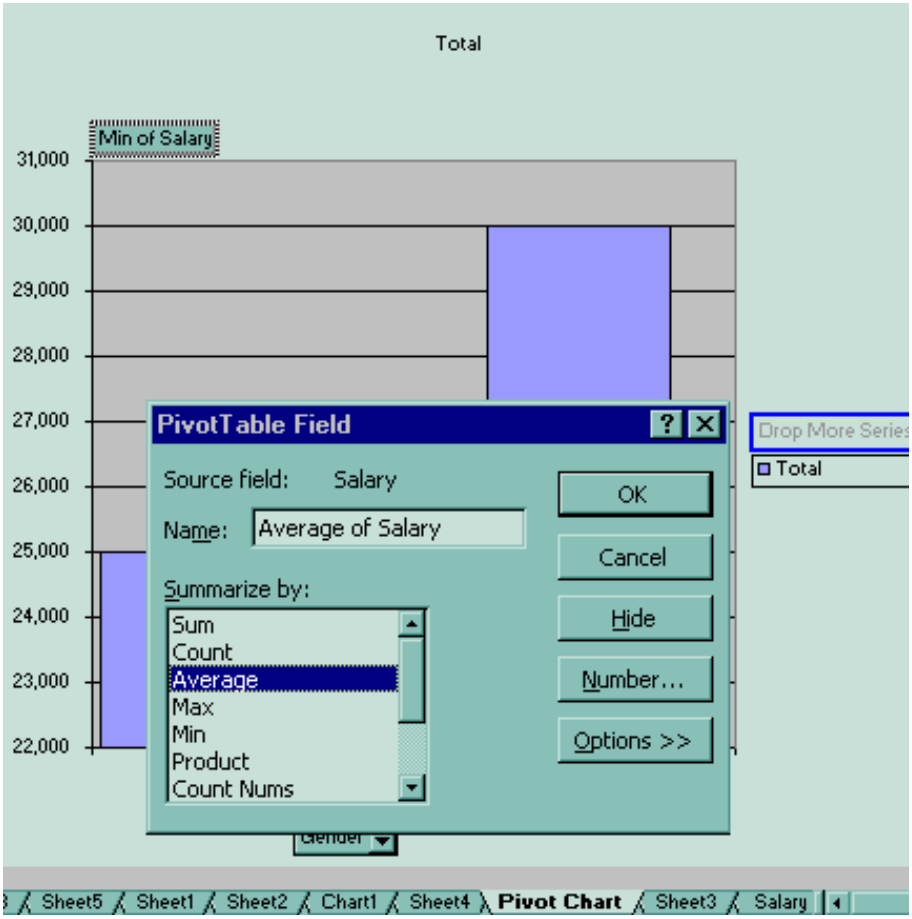
Select the Chart Wizard button.

Doing so will immediately bring up the pivot chart on a new worksheet.



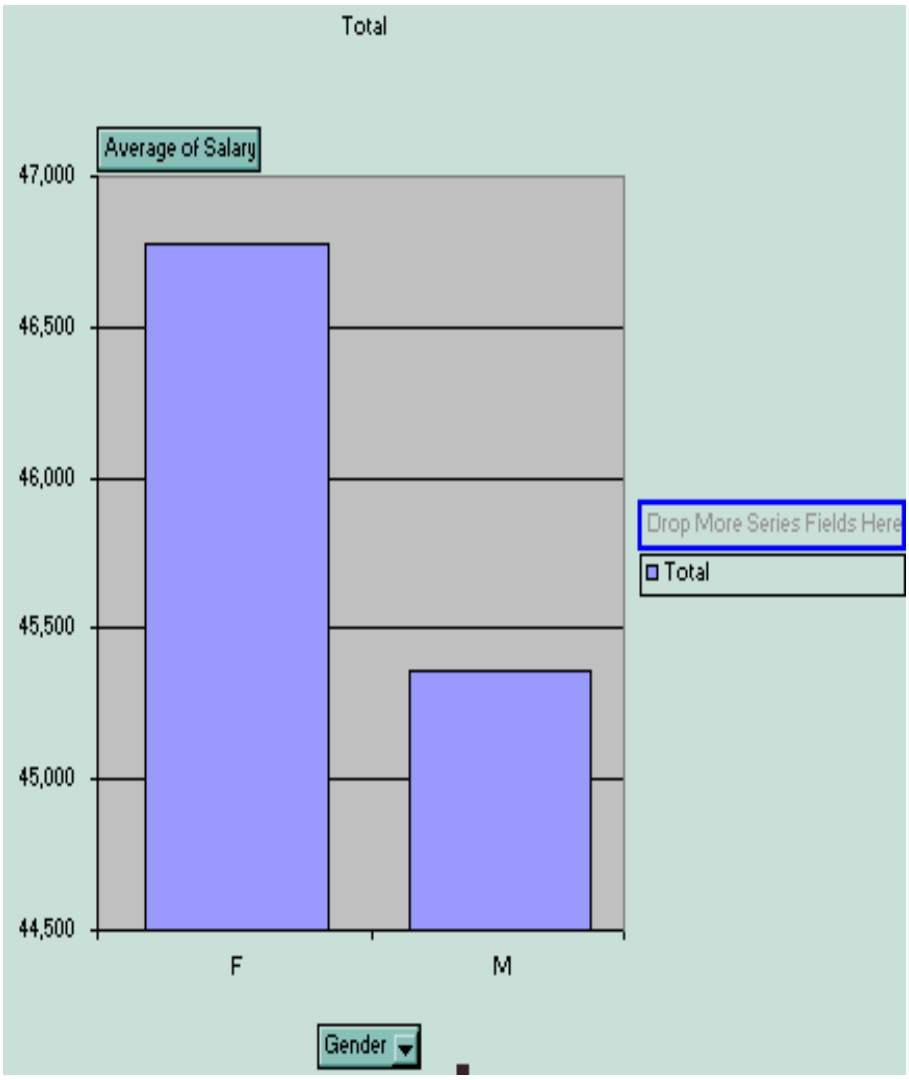
A pivot chart is worth a thousand words.

We can now manipulate the pivot chart in the same manner as we did with the pivot table, only now the results will be shown graphically. For example, the pivotchart currently shows the minimum salary for both groups, as reflected in the pivot table we created in the previous article. To see what the average salary is, we first double-click the Min of Salary button in the upper left corner. This will bring up the PivotTable Field dialog box. We then select Average from the Summarize by: scroll box just as we did when we worked with the pivot table.(After clicking OK, the chart will now display the average salary for both groups.



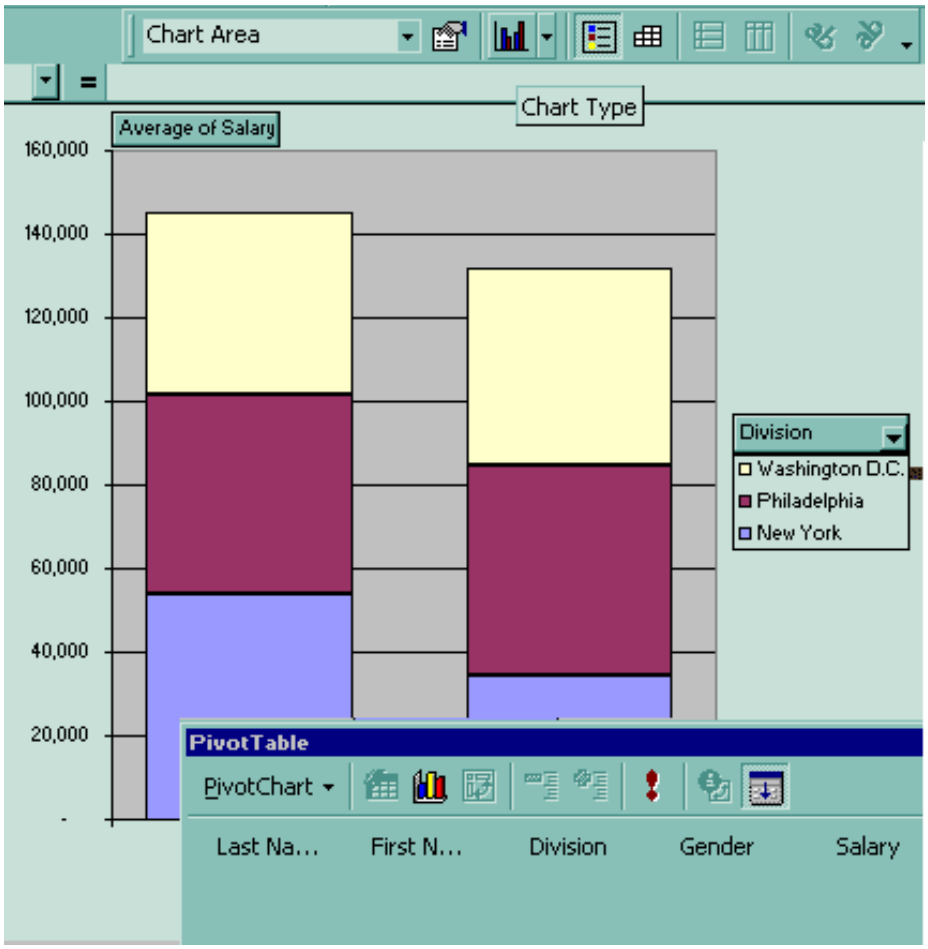
Selecting Average will make the pivot chart display the average salary for both groups of employees.

In addition to changing how the data is calculated, we can also add more fields to the chart by dragging and dropping them to the Drop More Series Fields Herebox located above the legend on the pivot chart.



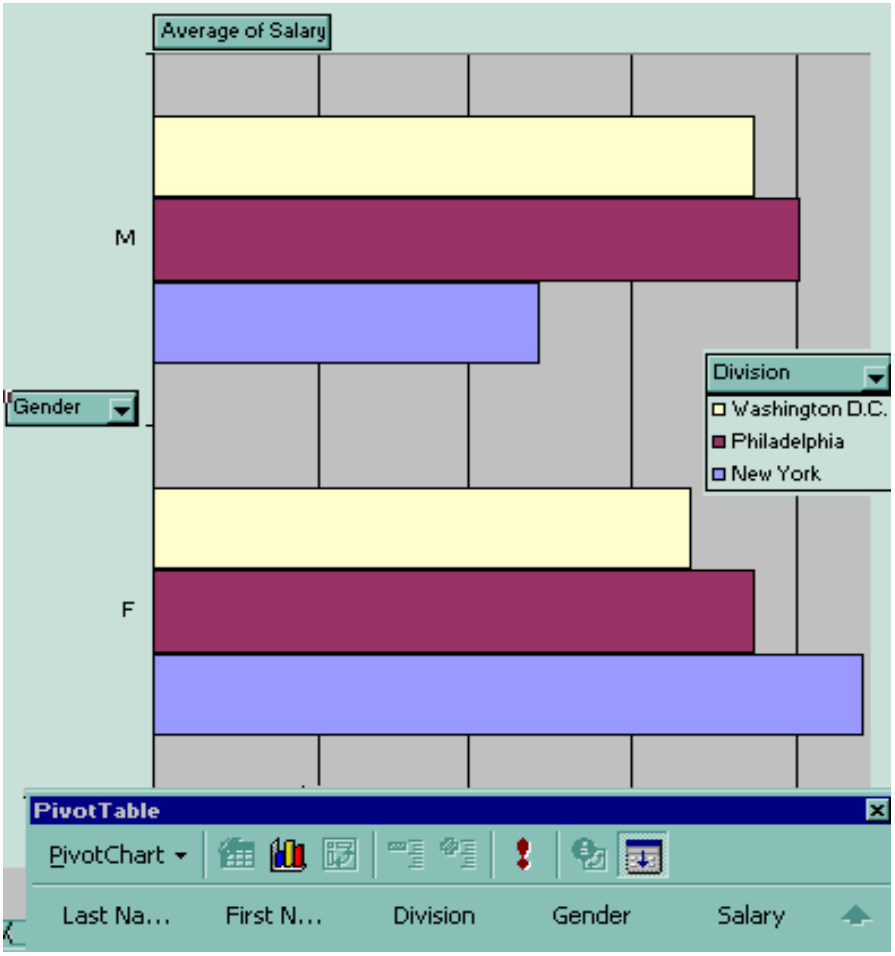
Add more fields by using Drop More Series Fields Here.

For example, to break down the average salaries for both groups by division, we would drag and drop the Division field from the PivotTable toolbar to get the results.



The average salaries by division for both groups of employees are now displayed.

Once a pivot chart is created, it can be changed and formatted like any other chart. For example, you can change the chart type by clicking the Chart Type button in the Chart toolbar. You can also use the Chart toolbar to make any other formattingchanges to your pivot chart.



You can customize your pivot chart to meet your specifications.

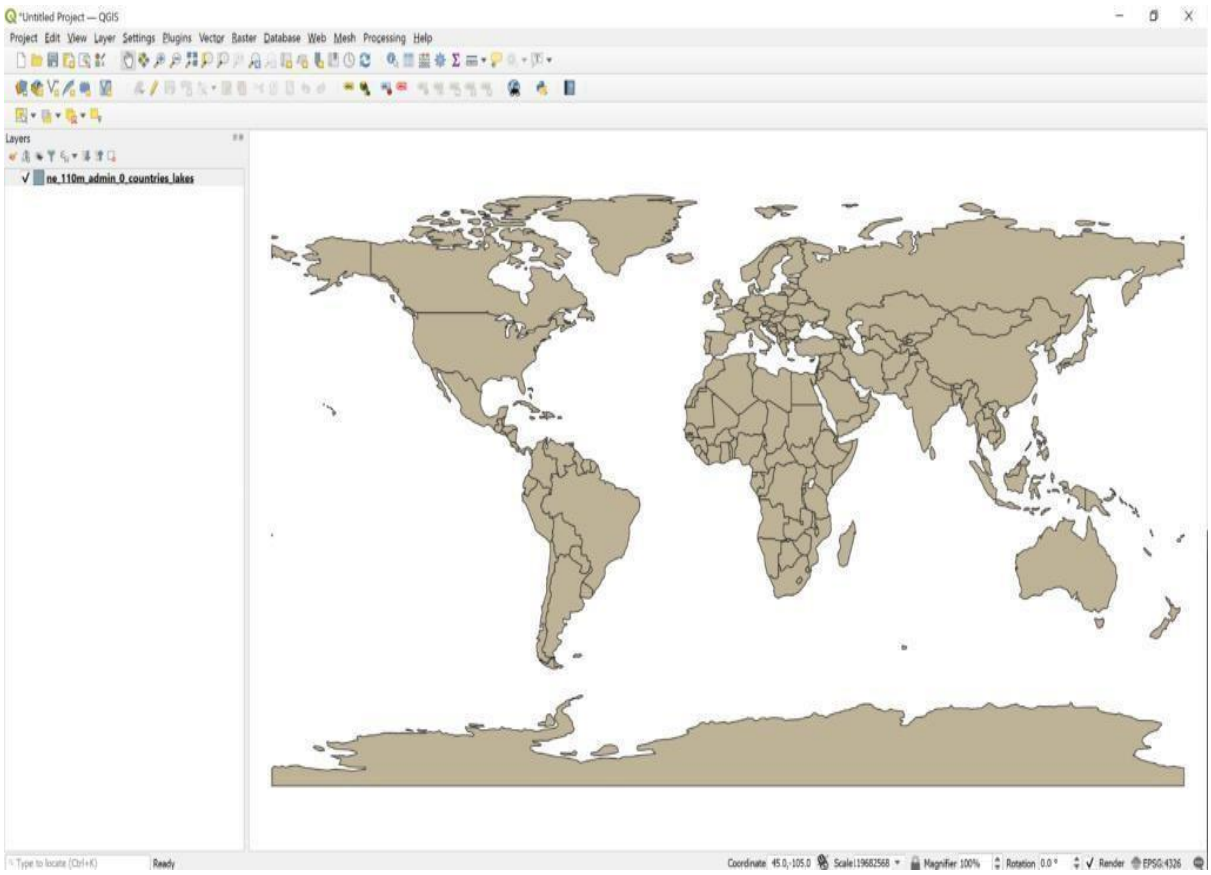
EX NO: 7.MAKING WORLD MAP INTERACTION WITH D3.JS AND SVG

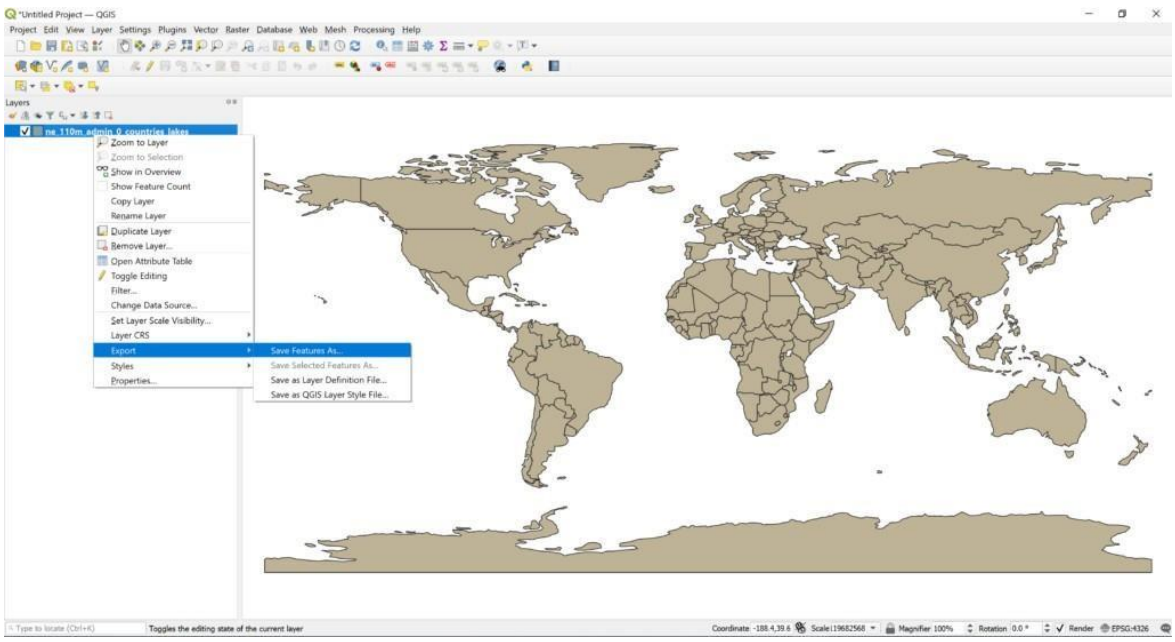
```
import pandas as pd

import requests

from bs4 import BeautifulSoup

url='https://en.wikipedia.org/wiki/List_of_countries_by_coffee_production'

r = requests.get(url)
html = r.text

soup = BeautifulSoup(html)
table = soup.find('table', {"class": "wikitable"})

rows = table.find_all('tr')
data = []
for row in rows[1:]:
 cols = row.find_all('td')

 currentList = []
 for idx, ele in enumerate(cols):
    # Remove comma from the numbers

    x = ele.text.strip().replace(',', '')
    # Clean the country name with unwanted text
    if idx == 1:
```

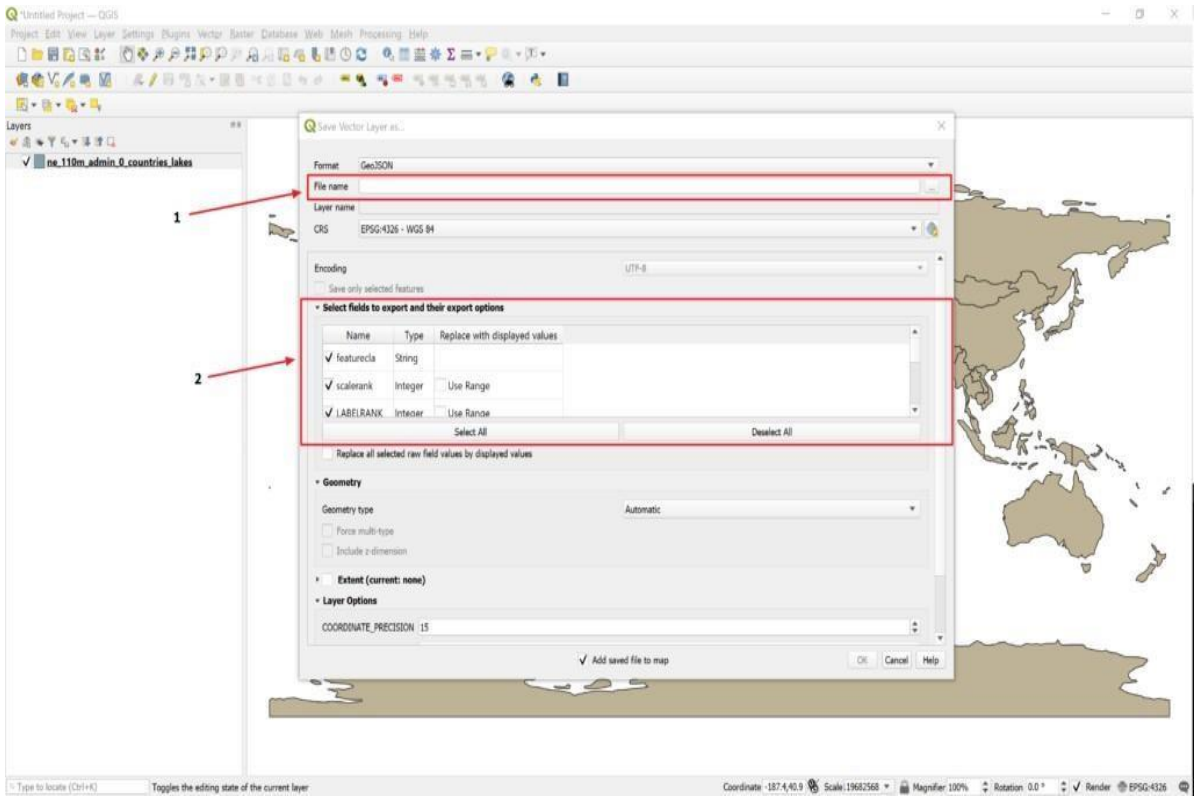
```
        oddCharacterIndex = x.find('(')
        if oddCharacterIndex != -1:
           x = x[0: oddCharacterIndex]
           currentList.append(x)
 data.append([item for item in currentList if item])

result = pd.DataFrame(
 data, columns=['Rank', 'Country', 'Bags', 'MetricTons', 'Pounds'])

with open(r'c:\your-directory-path     \temp.json', 'w') as f:
 f.write(result.to_json(orient='records'))

import { geoEquirectangular, geoPath } from 'd3-geo';
 const projection = geoEquirectangular().fitSize(mapSize, geoJson as any);
const geoPathGenerator = geoPath().projection(projection);

 let svgProps: SVGProps<SVGPathElement> = {
   d: geoPathGenerator(feature as any) || '',
   stroke: defaultColor,fill:
defaultColor,
  }
return (
 <div className="WorldMap">
  <div ref={tooltip} style={{ position: 'absolute', display: 'none' }}>
    <Tooltip>
      {tooltipContent}
    </Tooltip>
```

```
  </div>
  <svg
   className="WorldMap--svg"
   width={mapSize[0]}
   height={mapSize[1]}
  >
   {mapCountries.map(country => {return (
      <path
       id={country.countryName}
       key={country.countryName}
        {...country.svg as any}
       onMouseMove={(e) => handleMouseOverCountry(e, country)}
       onMouseLeave={() => handleMouseLeaveCountry(country)}
      />
    )
   })}
  </svg>
 </div>
 );
};
```

OUTPUT:


*Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Layers
ne_110m_admin_0_countries_lakes
Type to locate (Ctrl+K)
Ready
Coordinate 45.0,-105.0
Scale 119682568
Magnifier 100%
Rotation 0.0 °
Render
EPSG:4326


*Untitled Project — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Layers
ne_110m_admin_0_countries_lakes
Zoom to Layer
Zoom to Selection
Show in Overview
Show Feature Count
Copy Layer
Rename Layer
Duplicate Layer
Remove Layer...
Open Attribute Table
Toggle Editing
Filter...
Change Data Source...
Set Layer Scale Visibility...
Layer CRS
Export
Styles
Properties...
Save Features As...
Save Selected Features As...
Save as Layer Definition File...
Save as QGIS Layer Style File...
Type to locate (Ctrl+K)
Toggles the editing state of the current layer
Coordinate -188.4,39.6
Scale 119682568
Magnifier 100%
Rotation 0.0 °
Render
EPSG:4326


Layers
ne_110m_admin_0_countries_lakes
Save Vector Layer as...
Format
GeoJSON
File name
Layer name
CRS
EPSG:4326 - WGS 84
Encoding
UTF-8
Save only selected features
Select fields to export and their export options
Name
Type
Replace with displayed values
featurecla
String
scalerank
Integer
Use Range
LABELRANK
Integer
Use Range
Select All
Deselect All
Replace all selected raw field values by displayed values
Geometry
Geometry type
Automatic
Force multi-type
Include z-dimension
Extent (current: none)
Layer Options
COORDINATE_PRECISION
15
Add saved file to map
OK
Cancel
Help
1
2
Type to locate (Ctrl+K)
Toggles the editing state of the current layer
Coordinate -187.4,40.9
Scale 19682568
Magnifier 100%
Rotation 0.0 °
Render
EPSG:4326

EX NO: 8.ANALYSIS OF VARIANCE (ANOVA)

ANOVA

Analysis of variance (ANOVA) is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples.

Terminologies related to ANOVA

Before we get started with the applications of ANOVA, I would like to introducesome common terminologies used in the technique.

Grand Mean

Mean is a simple or arithmetic average of a range of values. There are two kinds of means that we use in ANOVA calculations, which are separate sample means $(\mu_1, \mu_2 \& \mu_3)$ and the grand mean (μ). The grand mean is the mean of samplemeans or the mean of all observations combined, irrespective of the sample.

Hypothesis

$$H_o : \mu_1 = \mu_2 = \cdots = \mu_L \qquad \textit{Null hypothesis}$$

$$H_1 : \mu_l \neq \mu_m \qquad \textit{Alternate hypothesis}$$

Anova: Single Factor

	A	B	C	D	E	F	G	H
1	Anova: Single Factor							
2								
3	SUMMARY							
4	*Groups*	*Count*	*Sum*	*Average*	*Variance*			
5	Atlanta	9	118.75	13.19444	0.199653			
6	Chicago	9	113.5	12.61111	0.504236			
7	Houston	9	119.76	13.30667	0.309525			
8	Memphis	9	119.2	13.24444	0.935103			
9	New York	9	121.35	13.48333	0.516875			
10	Philadelphia	9	109.8	12.2	0.254375			
11								
12								
13	ANOVA							
14	*Source of Variation*	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*	
15	Between Groups	10.94567	5	2.189133	4.829385	0.001175	2.408514	
16	Within Groups	21.75813	48	0.453294				
17								
18	Total	32.7038	53					
19								

t-Test:Two-Sample Assuming Equal Variance

	A	B	C	D
1	t-Test: Two-Sample Assuming Equal Variances			
2				
3		*Atlanta*	*Chicago*	
4	Mean	13.19444	12.61111	
5	Variance	0.199653	0.504236	
6	Observations	9	9	
7	Pooled Variance	0.351944		
8	Hypothesized Mean Diffe	0		
9	df	16		
10	t Stat	2.085864		
11	P(T<=t) one-tail	0.02668		
12	t Critical one-tail	1.745884		
13	P(T<=t) two-tail	0.05336		
14	t Critical two-tail	2.119905		
15				

	A	B	C	D	E	F	G	H
1		*Atlanta*	*Chicago*	*Houston*	*Memphis*	*New York*	*Philadelphia*	
2	Atlanta	1						
3	Chicago	0.814737	1					
4	Houston	0.92689	0.837158	1				
5	Memphis	0.935793	0.939693	0.941073	1			
6	New York	0.925773	0.856159	0.965199	0.967889	1		
7	Philadelph	0.644805	0.953709	0.635027	0.815921	0.684291	1	
8								

Castle Bakery Example

Two-Factor Study

Castle Bakery Example
Two-Factor Study

Wrapped Italian Bread Sales (in cases)

Display Height	Display Width: Regular	Display Width: Wide
Bottom	47	46
	43	40
Middle	62	67
	68	71
Top	41	42
	39	46

The Castle Bakery Company supplies wrapped Italian bread to a large number of supermarkets in a metropolitan area. An experimental study was made of the effects height of the shelf display (bottom, middle, top) and width of the shelf display (regular, wide) on sales of this bakery's bread during the experimental period. Twelve supermarkets, similar in terms of sales volume and clientele, were utilized in the study. The six treatments were assigned at random to two stores each according to a completely randomized design, and the display of bread in each store followed the treatment specifications for that store.

Two Way ANOVA

	Age Group(yrs)		
Class Group	4-8 yrs	8-13 yrs	13-17 yrs
A (Constant sound)	6	4	7
	5	5	6
	5	6	10
	2	9	8
	4	8	9
Sub-total A	22	32	40
B (No sound)	1	4	6
	3	5	8
	2	6	4
	1	7	7
	2	3	5
Sub-total B	9	25	30
Grand Total	31	57	70

		factor 2				
		Age Group(yrs)				
factor 1	Class Group	4-8 yrs	8-13 yrs	13-17 yrs	A_l	factor 1 total
	A (Constant sound)	6,5,5,2,4 (22)	4,5,6,9,8 (32)	7,6,10,8,9 (40)	94	
	B (No sound)	1,3,2,1,2 (9)	4,5,6,7,3 (25)	6,8,4,7,5 (30)	64	
	A_m	31	57	70	316	Grand Total (G)
		subtotal (A_{lm})				
	factor 2 total					

SUMMARY	4-8 yrs	8-13 yrs	13-17 yrs	Total
A (Constant sound)				
Count	5	5	5	15
Sum	22	32	40	94
Average	4.4	6.4	8	6.26666667
Variance	2.3	4.3	2.5	4.92380952
B (No sound)				
Count	5	5	5	15
Sum	9	25	30	64
Average	1.8	5	6	4.26666667
Variance	0.7	2.5	2.5	5.06666667
Total				
Count	10	10	10	
Sum	31	57	70	
Average	3.1	5.7	7	
Variance	3.21111111	3.56666667	3.33333333	

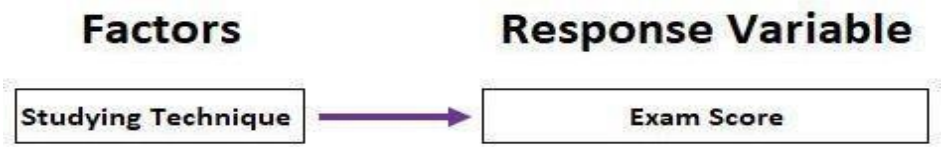
EX NO: 9. ANALYSIS OF COVARIANCE (ANCOVA)

ANCOVA stands for "analysis of covariance." To understand how an ANCOVA works, it helps to first understand the ANOVA.

An ANOVA (analysis of variance) is used to determine whether or not there is a statistically significant difference between the means of three or more independent groups.

For example, suppose we want to know whether or not studying technique has an impact on exam scores for a class of students. We randomly split the class into three groups. Each group uses a different studying technique for one month to prepare for an exam. At the end of the month, all of the students take the same exam.

To find out if studying technique impacts exam scores, we can conduct a one- way ANOVA, which will tell us if if there is a statistically significant difference between the mean scores of the three groups.

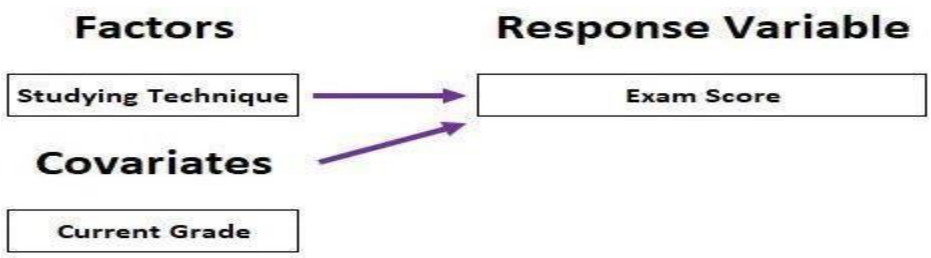


An **ANCOVA** is an extension of an ANOVA in which we"d like to determine if there is a statistically significant difference between three or more independent groups after accounting for one or more covariates.

For example, suppose we want to know whether or not studying technique has an impact on exam scores, *but we want to account for the grade that the student already has in the class*. We can use their current grade as a covariate and conductan ANCOVA to determine if there is a statistically significant difference between the mean exam scores of the three groups.

This allows us to test whether or not studying technique has an impact on exam scores after the influence of the covariate has been removed.

Thus, if we find that there is a statistically significant difference in exam scores between the three studying techniques, we can be sure that this difference exists even after accounting for the students current grade in the class (i.e. if they‟re already doing well or not in the class).



Assumptions of ANCOVA

Before performing an ANCOVA, it‟s important to make sure the following assumptions are met:

- **The covariate(s) and the factor variable(s) are independent** – The covariate and the factor variable should be independent of each other, since adding a covariate term into the model only makes sense if the covariate and the factor variable act independently on the response variable.

- **The covariate(s)are continuousdata.** The covariates should be continuous (i.e. either interval or ratio data).
- **Homogeneity of variances** – The variances among the groups should beroughly equal.
- **Independence** – The observations in each group should be independent.
- **Normality** – The data should be roughly normally distributed in eachgroup.
- **No extreme outliers** – There should be no extreme outliers in any of thegroups that could significantly affect the results of the ANCOVA.

ANCOVA: Example

A teacher wants to know if three different studying techniques have an impact on exam scores, but she wants to account for the current grade that the student already has in the class.

She will perform an ANCOVA using the following variables:

- **Factor variable:** studying technique
- **Covariate:** current grade
- **Response variable:** exam score

The following table shows the dataset for the 15 students that were recruited to participate in the study:

Student	Study Technique	Current Grade	Exam Score
Student 1	A	67	77
Student 2	A	88	89
Student 3	A	75	72
Student 4	A	77	74
Student 5	A	85	69
Student 6	B	92	78
Student 7	B	69	88
Student 8	B	77	93
Student 9	B	74	94
Student 10	B	88	90
Student 11	C	96	85
Student 12	C	91	81
Student 13	C	88	83
Student 14	C	82	88
Student 15	C	80	79

After performing an ANCOVA on the dataset, the teacher ends up with the following results:

Source	SS	df	MS	F	p
Study Technique	390.58	2	195.29	4.81	0.03155
Current Grade	4.19	1	4.19	0.103	0.7539
Residuals	446.61	11	40.6		

The p-value for study technique is **0.03155**. Since this value is less than 0.05, wecan reject the null hypothesis that each of the studying techniques leads to the same average exam score, even after accounting for the student‟s current grade in the class.

EX NO: 10. STATISTICS ASSOCIATED WITH CLUSTER ANALYSIS

Importing Required Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, normalize
from sklearn.metrics import silhouette_score
```

Reading the data and getting top 5 observations to have a look at the data set

```
data = pd.read_csv(r'C:\Users\SHIVAM\Downloads\datasets_531_1056_german_credit_data.csv')
```

```
data.head()
```

	Unnamed: 0	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose
0	0	67	male	2	own	NaN	little	1169	6	radio/TV
1	1	22	female	2	own	little	moderate	5951	48	radio/TV
2	2	49	male	1	own	little	NaN	2096	12	education
3	3	45	male	2	free	little	little	7882	42	furniture/equipment
4	4	53	male	2	free	little	little	4870	24	car

The code for EDA (Exploratory Data Analysis) has not been included, EDA was performed on this data, and outlier analysis was done to clean the data and make it fit for our analysis.

As we know that K-means is performed only on the numerical data so we choose the

numerical columns for our analysis.

```
#K-Means works with Numeric Data
selected_cols = ["Age","Credit amount", "Duration"]
cluster_data = data.loc[:,selected_cols]
```

Now to perform the k-means clustering as discussed earlier in this article we need to find the value of the „k" number of clusters and we can do that using the following code, here we using several values of k for clustering and then selecting using the **Elbow method.**

Finding optimum number of clusters

```
#Plotting Scree Plot to find optimum number of clusters
clusters_range = [2,3,4,5,6,7,8,9,10,11,12,13,14]
inertias =[]

for c in clusters_range:
    kmeans = KMeans(init='k-means++',n_clusters=c,n_init=100, random_state=0).fit(normalized_df)
    inertias.append(kmeans.inertia_)

plt.figure()
plt.plot(clusters_range,inertias, marker='o')
plt.show()
```

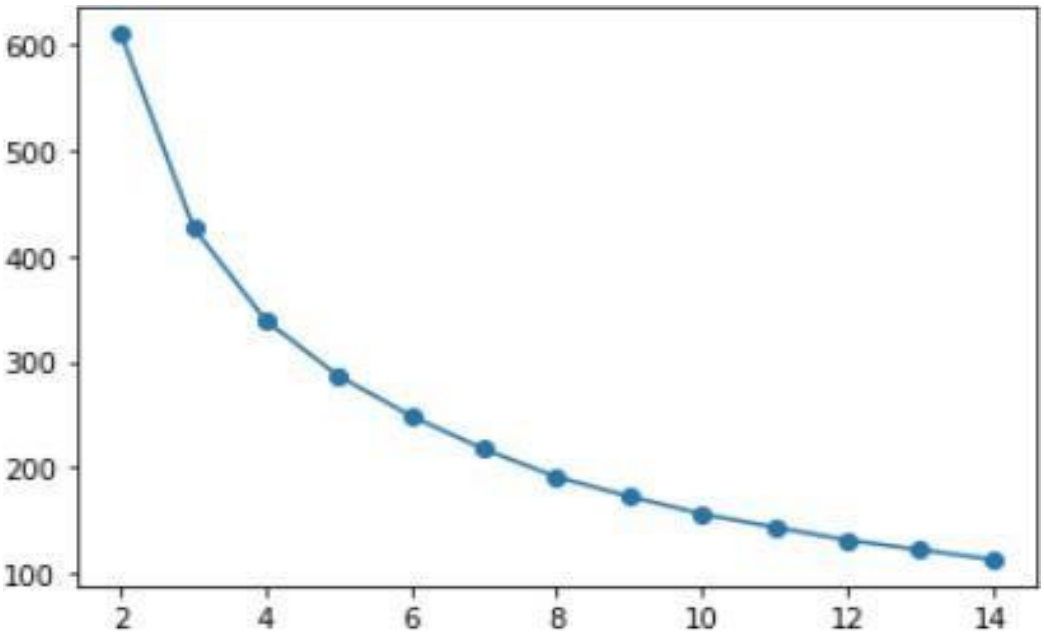
```
from sklearn.metrics import silhouette_samples, silhouette_score

clusters_range = range(2,15)
results =[]
for c in clusters_range:
    clusterer = KMeans(init='k-means++',n_clusters=c,n_init=100, random_state=0)
    cluster_labels = clusterer.fit_predict(normalized_df)
    silhouette_avg = silhouette_score(normalized_df, cluster_labels)
    results.append([c,silhouette_avg])

result = pd.DataFrame(results, columns=["n_clusters","silhouette_score"])
pivot_km = pd.pivot_table(result, index="n_clusters", values="silhouette_score")

plt.figure()
sns.heatmap(pivot_km, annot=True, linewidths=.5, fmt='.3f', cmap=sns.cm.rocket_r)
plt.tight_layout()
```

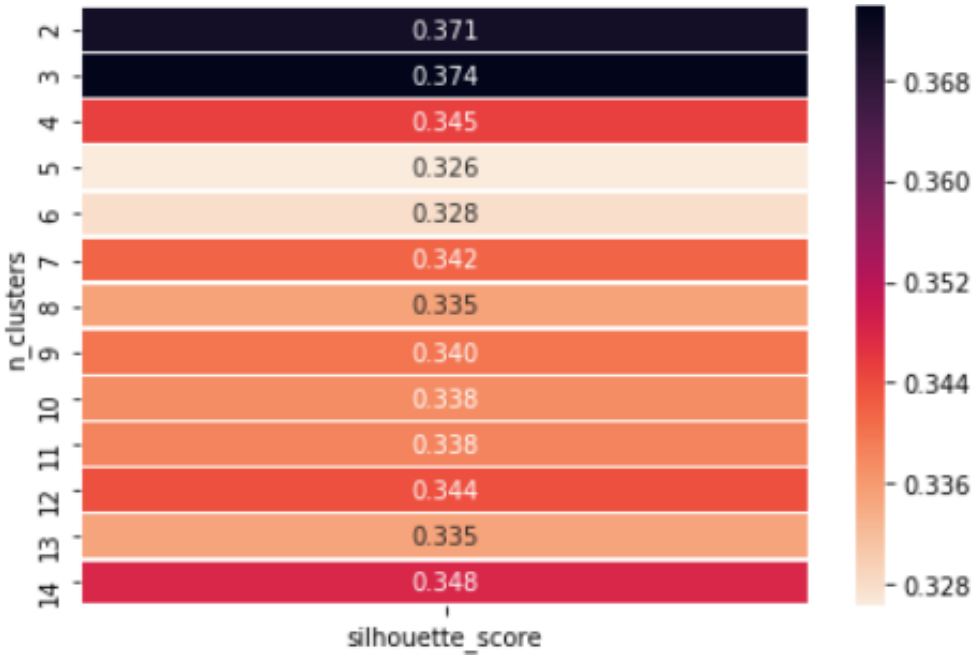
Fitting KMeans algo for 3 clusters

```
kmeans_sel = KMeans(init='k-means++',n_clusters=3,n_init=100, random_state=1).fit(normalized_df)
labels = pd.DataFrame(kmeans_sel.labels_)
clustered_data = cluster_data.assign(Cluster=labels)
```

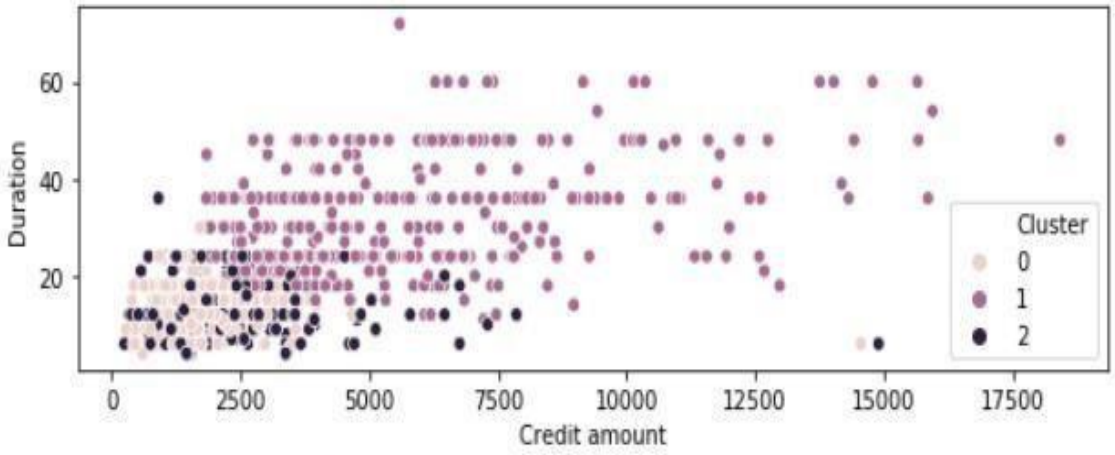
```
scatters(clustered_data, h='Cluster')
```

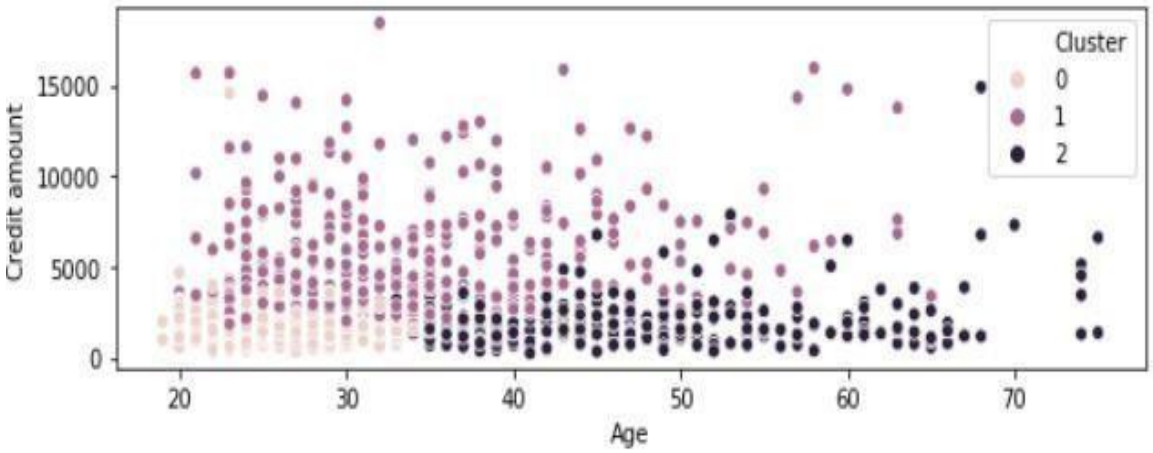
OUTPUT:

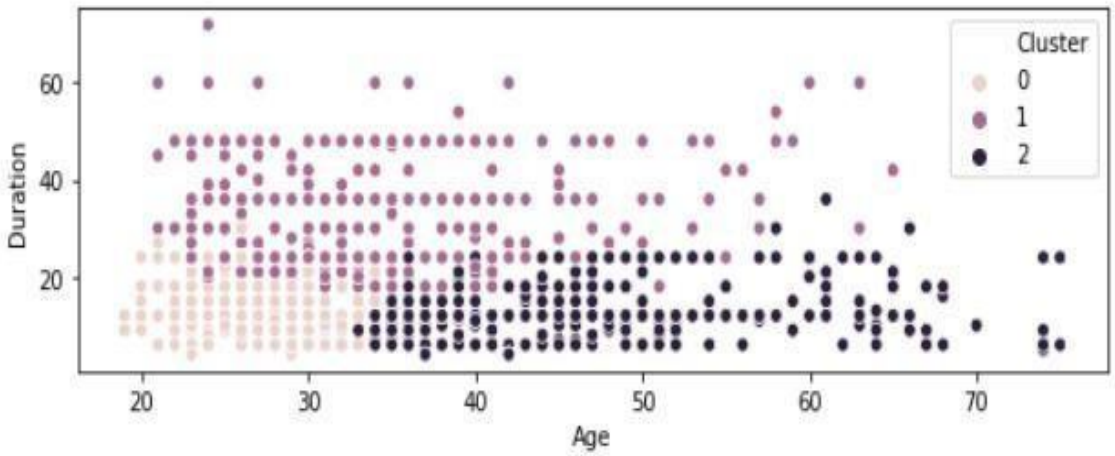

600
500
400
300
200
100
2
4
6
8
10
12
14


n_clusters
2
3
4
5
6
7
8
9
10
11
12
13
14
0.371
0.374
0.345
0.326
0.328
0.342
0.335
0.340
0.338
0.338
0.344
0.335
0.348
silhouette_score
0.368
0.360
0.352
0.344
0.336
0.328

Cluster	Age	Credit amount	Duration
0	26.4	1684.6	14.1
1	33.9	5544.1	31.8
2	46.4	1959.0	13.8







EX NO: 11. DESIGN AND DEVELOPMENT OF ANALYTICAL PRODUCTS

A growing area of research in the engineering community is the use of data and analytics for transforming information into knowledge to design better systems, products, and processes. Data-driven decisions can be made in the early, middle, and late stages in a design process where customer needs are identified and understood, a final concept for a design is chosen, and usage data from the deployed product is captured, respectively.

Design Analytics (DA) is a paradigm for improving the core information-to-knowledge transformations in these stages of a design process resulting in better performing and functioning products that reflect both explicit and implicit customer needs. In this paper, a simulator is used to model usage of a hypothetical refrigerator and generate artificial data driven by four different customer behavior profiles with variation.

The population of customers is randomly divided among the four behavior profiles so that the underlying customer preferences are unknown to the experimenter prior to data analysis.

The purpose of the simulation is to illustrate the use of DA in the late stage of a design process to improve the transition from an existing product to the next generation product. Metrics are developed to analyze the product usage data, and both prevailing and subtle usage trends are identified. After conclusions are made, the study proceeds to the early and middle stages of a subsequent design process where a hypothetical next-generation refrigeratoris conceptualized.

PROCEDURE:

- greater_than_slot_ratio: the preferred shelf location of each shelf inside the refrigerator
- num_food_genesis: the number of food items initially stocked in therefrigerator
- greater_than_area_ratio: the allowable area for a food item
- greater_than_height_ratio: the allowable height for a food item
- num_food_move: the number of food items moved each time therefrigerator is accessed
- num_food_expire: the number of food items expired each time therefrigerator is accessed
- num_food_add: the number of food items added each time therefrigerator is accessed
- lucky_shelf_ratio: the preferred shelf that a customer prefers to putfood on
- use_refrigerator: the number of times the refrigerator is accessed perday.

Metrics for Data Analysis:

As the simulator runs, three metrics quantify product usage and are updated continuously. The first metric is the cumulative moving average of shelf placement and calculated as:

$$Moving\ Average_i = \frac{(Moving\ Average_{i-1})(i-1)+X_i}{i}$$

The maximum frequency for each shelf, shown at the bottom left of each graph, is used to normalize the scale of movement activity. The normalizationis calculated as:

$$Y_{i,0\ to\ 1} = \frac{Y_i - Y_{min}}{Y_{max} - Y_{min}}$$

Customer population distribution by the number of shelves used:

Total Number of Shelves Used	2	3	4	5
Percent of Total Customer Population	26.8%	18%	41.3%	13.9%

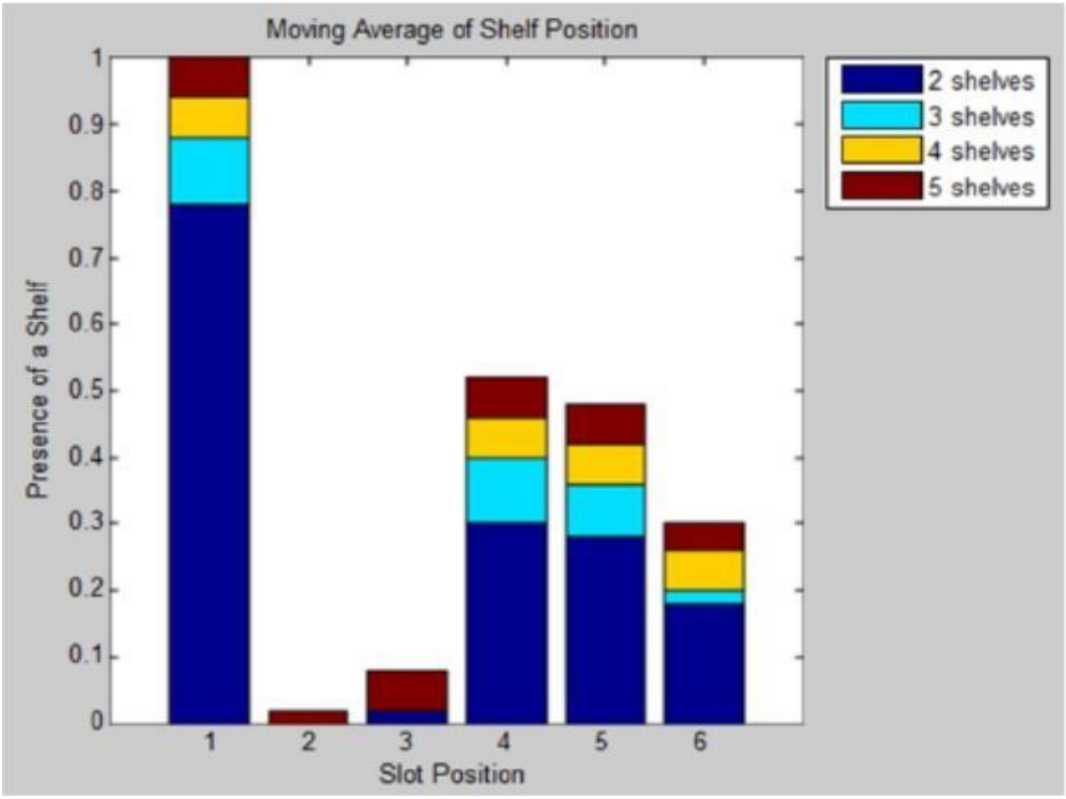
Randomized customer distribution correlated to customer segments according to the number of shelves used by each customer.

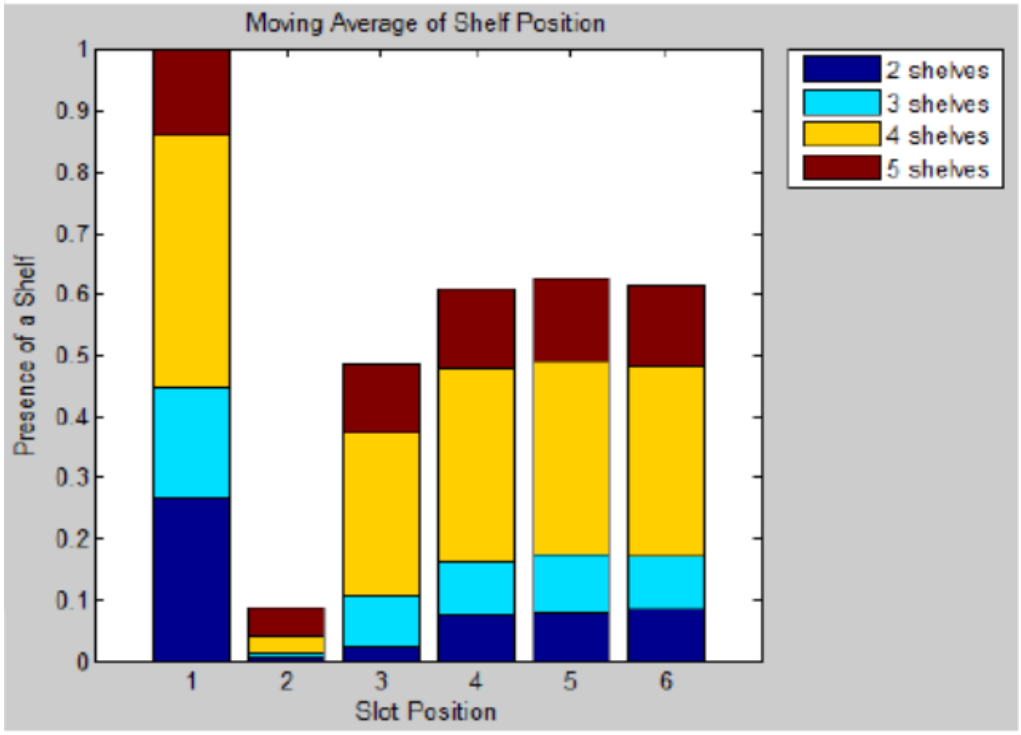
	No. of Customer 1 Profiles	No. of Customer 2 Profiles	No. of Customer 3 Profiles	No. of Customer 4 Profiles
Actual customer population	288 (28.8%)	149 (14.9%)	518 (51.8%)	45 (4.5%)
Customer segments according to no. of shelves used (from Table 1)	26.8%	18.0%	41.3%	13.9%
Correlation coefficient	r = .997 (strong-positive)			

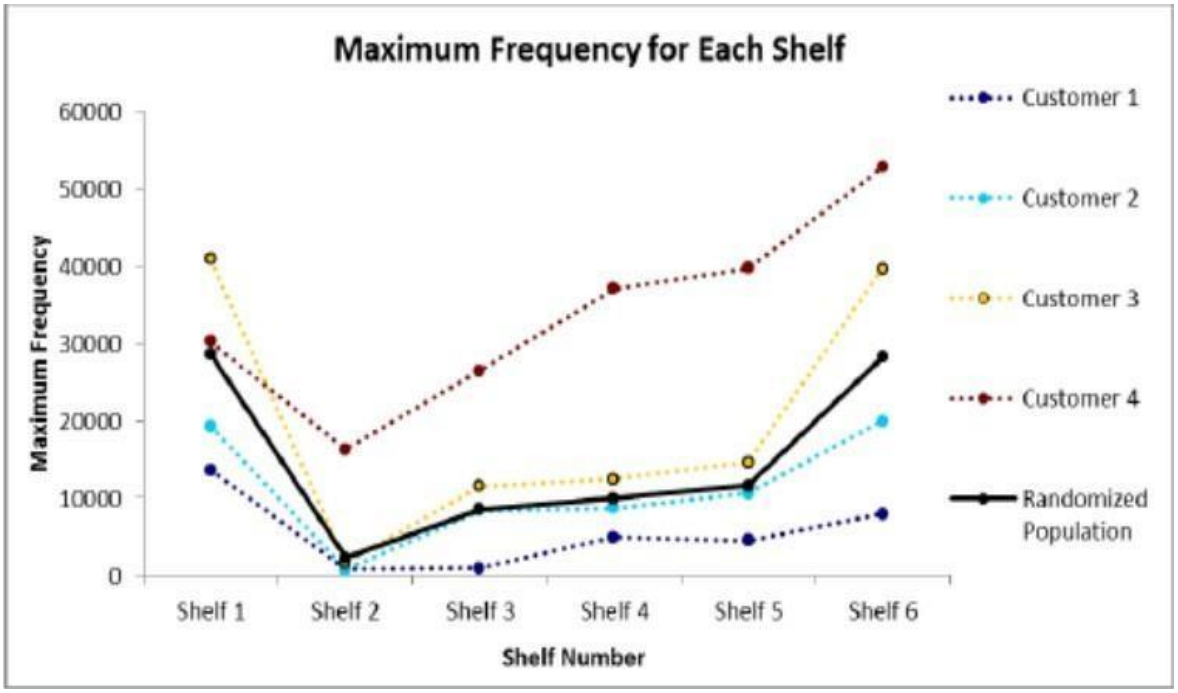
Defines The Correlation Coefficient As:

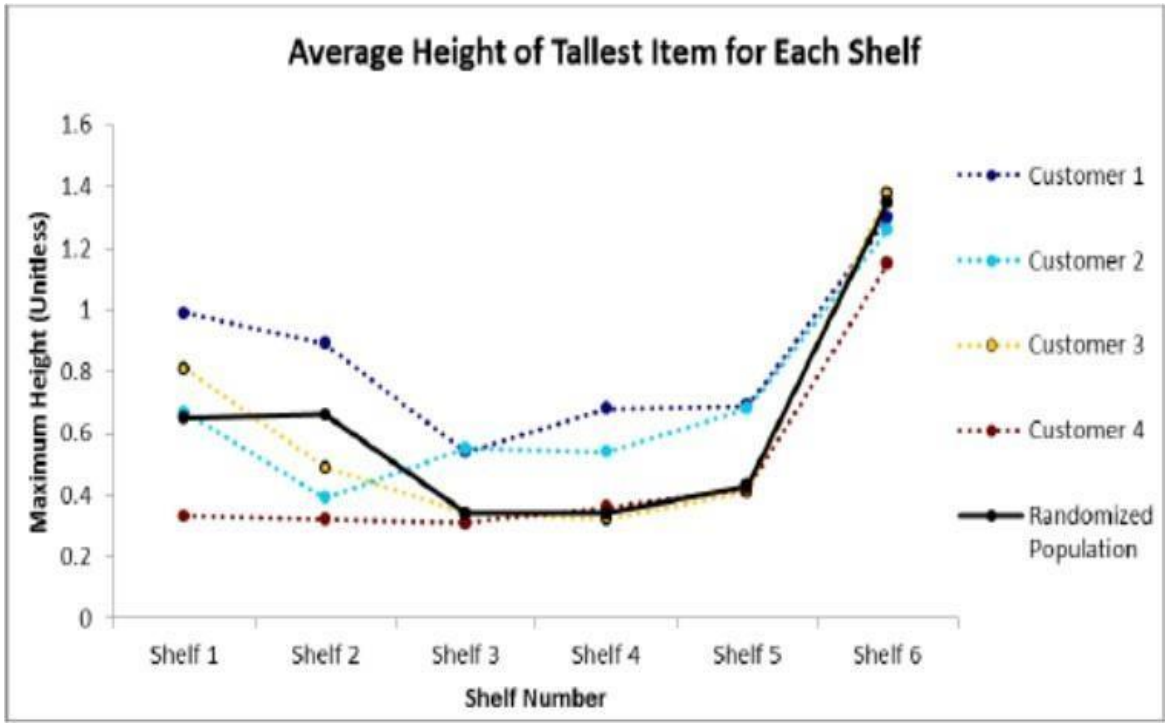
$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(b_i - \bar{b})^2}}$$

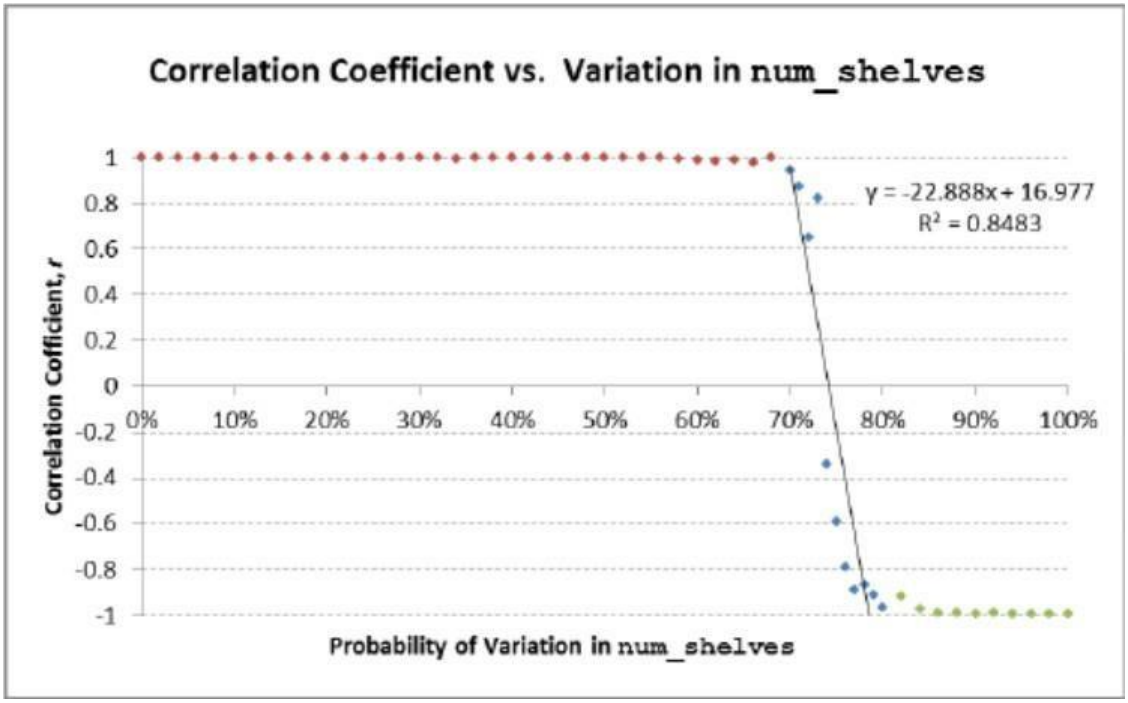
OUTPUT:


Moving Average of Shelf Position
Presence of a Shelf
Slot Position
2 shelves
3 shelves
4 shelves
5 shelves
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
2
3
4
5
6


Moving Average of Shelf Position
Presence of a Shelf
Slot Position
2 shelves
3 shelves
4 shelves
5 shelves
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
2
3
4
5
6


Maximum Frequency for Each Shelf
60000
50000
40000
30000
20000
10000
0
Maximum Frequency
Shelf 1
Shelf 2
Shelf 3
Shelf 4
Shelf 5
Shelf 6
Shelf Number
Customer 1
Customer 2
Customer 3
Customer 4
Randomized Population


Average Height of Tallest Item for Each Shelf
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Maximum Height (Unitless)
Shelf 1
Shelf 2
Shelf 3
Shelf 4
Shelf 5
Shelf 6
Shelf Number
Customer 1
Customer 2
Customer 3
Customer 4
Randomized Population


Correlation Coefficient vs. Variation in num_shelves
Correlation Cofficient, r
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
y = -22.888x + 16.977
R² = 0.8483
Probability of Variation in num_shelves