**DATA VISUALIZATION LAB MANUAL**

**MR23-1CS0150**

**Index**

|  |  |
| --- | --- |
| **S.No** | **Task** |
| Week1 | Import a sales dataset and perform different data manipulation techniques. |
| Week 2 | Perform different data pre-processing techniques on the sales dataset. |
| Week 3 | Conduct a complete data analysis on a given student results dataset and derive insights using the ggplot2 package in R. |
| Week 4 | Perform a data analysis on the weather dataset and extract insights using the ggplot2 package in R. Utilize Histograms, Box plots, Bar charts, Scatter plots, and Line charts to visualize the data. |
| Week 5 | Merge two DataFrames and apply various data manipulation techniques using the Pandas library in Python. |
| Week 6 | Use the Python ‘Matplotlib’ to perform a thorough data analysis and extract insights from a given Housing dataset. |
| Week 7 | List 5 findings from the state\_wise\_covid dataset by making an in-depth data analysis with the help of the ‘Matplotlib’ library. |
| Week 8 | Customize Pair Plots, subplots and Joint plots with different color palettes using the Seaborn library on any dataset. |
| Week 9 | Introduction to Tableau Desktop and Installation.  Connecting to Data and preparing data for visualization in Tableau. |
| Week 10 | Data transformation, Data aggregation and statistical functions in Tableau. |
| Week 11 | Perform various visualization operations on data in Tableau. |
| Week 12 | Case study: Create a dashboard that gives in-depth insights into sales data with a minimum of six worksheets. |

**WEEK-1**

**Import a sales dataset and perform below data manipulation techniques.**

1.Add new rows

2.Create new column “total\_revenue” by multiplying quantity sold by the price.

3.Delete rows.

4.Delete column.

5.Rename “Quantity” column to “Quantity\_sold”.

6.Create new columns for “day”,“month” and “year” from “Order Date”.

7.Add +2 to “Quantity” variable of South Region.

# load sales dataset  
**data = read.csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/sales\_data.csv",fileEncoding = "UTF-8-BOM")  
#examin the data  
head(data)**

## Row.ID Order.ID Order.Date Ship.Date Country Region  
## 1 1 CA-2016-152156 08-11-2016 11-11-2016 United States South  
## 2 2 CA-2016-152156 08-11-2016 11-11-2016 United States South  
## 3 3 CA-2016-138688 12-06-2016 16-06-2016 United States West  
## 4 4 US-2015-108966 11-10-2015 18-10-2015 United States South  
## 5 5 US-2015-108966 11-10-2015 18-10-2015 United States South  
## 6 6 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## Category Sales Quantity  
## 1 Furniture 261.9600 2  
## 2 Furniture 731.9400 3  
## 3 Office Supplies 14.6200 2  
## 4 Furniture 957.5775 5  
## 5 Office Supplies 22.3680 2  
## 6 Furniture 48.8600 7

# Check the dimentions  
**dim(data)**

## [1] 690 9

#check the structure of the data  
**str(data)**

## 'data.frame': 690 obs. of 9 variables:  
## $ Row.ID : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Order.ID : Factor w/ 321 levels "CA-2014-101476",..: 157 157 146 273 273 12 12 12 12 12 ...  
## $ Order.Date: Factor w/ 262 levels "01-02-2014","01-03-2014",..: 60 60 91 86 86 66 66 66 66 66 ...  
## $ Ship.Date : Factor w/ 279 levels "01-05-2016","01-06-2016",..: 99 99 143 167 167 126 126 126 126 126 ...  
## $ Country : Factor w/ 1 level "United States": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Region : Factor w/ 4 levels "Central","East",..: 3 3 4 3 3 4 4 4 4 4 ...  
## $ Category : Factor w/ 3 levels "Furniture","Office Supplies",..: 1 1 2 1 2 1 2 3 2 2 ...  
## $ Sales : num 262 731.9 14.6 957.6 22.4 ...  
## $ Quantity : int 2 3 2 5 2 7 4 6 3 5 ...

**tail(data)**

## Row.ID Order.ID Order.Date Ship.Date Country Region  
## 685 685 US-2017-168116 04-11-2017 04-11-2017 United States South  
## 686 686 CA-2014-157784 05-07-2014 08-07-2014 United States South  
## 687 687 CA-2014-157784 05-07-2014 08-07-2014 United States South  
## 688 688 CA-2014-157784 05-07-2014 08-07-2014 United States South  
## 689 689 CA-2017-161480 25-12-2017 29-12-2017 United States East  
## 690 690 US-2014-117135 21-06-2014 23-06-2014 United States South  
## Category Sales Quantity  
## 685 Office Supplies 167.440 2  
## 686 Technology 479.970 3  
## 687 Office Supplies 14.620 2  
## 688 Office Supplies 19.440 3  
## 689 Furniture 191.984 2  
## 690 Furniture 104.010 1

**1.ADD rows**

**df <- data.frame(  
 Row.ID = c(693L, 694L),  
 Order.ID = c("CA-2016-789123", "US-2018-987654"),  
 Order.Date = c("05-11-2015", "14-12-2016"),  
 Ship.Date = c("12-11-2015", "20-12-2016"),  
 Country = c("United States", "United States"),  
 Region = c("West", "Central"),  
 Category = c("Technology", "Furniture"),  
 Sales = c(543.8, 789.6),  
 Quantity = c(2L, 7L)  
)  
  
data = rbind(data,df)  
head(data)**

## Row.ID Order.ID Order.Date Ship.Date Country Region  
## 1 1 CA-2016-152156 08-11-2016 11-11-2016 United States South  
## 2 2 CA-2016-152156 08-11-2016 11-11-2016 United States South  
## 3 3 CA-2016-138688 12-06-2016 16-06-2016 United States West  
## 4 4 US-2015-108966 11-10-2015 18-10-2015 United States South  
## 5 5 US-2015-108966 11-10-2015 18-10-2015 United States South  
## 6 6 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## Category Sales Quantity  
## 1 Furniture 261.9600 2  
## 2 Furniture 731.9400 3  
## 3 Office Supplies 14.6200 2  
## 4 Furniture 957.5775 5  
## 5 Office Supplies 22.3680 2  
## 6 Furniture 48.8600 7

dim(data)

## [1] 692 9

print(data[data$Row.ID==693, ])

## Row.ID Order.ID Order.Date Ship.Date Country Region  
## 691 693 CA-2016-789123 05-11-2015 12-11-2015 United States West  
## Category Sales Quantity  
## 691 Technology 543.8 2

**2.Create new column “Total\_revenue” by multiplying quantity sold by the price.**

**library(dplyr)**

**data = mutate(data, Total\_revenue=Sales\*Quantity)  
head(data)**

## Row.ID Order.ID Order.Date Ship.Date Country Region  
## 1 1 CA-2016-152156 08-11-2016 11-11-2016 United States South  
## 2 2 CA-2016-152156 08-11-2016 11-11-2016 United States South  
## 3 3 CA-2016-138688 12-06-2016 16-06-2016 United States West  
## 4 4 US-2015-108966 11-10-2015 18-10-2015 United States South  
## 5 5 US-2015-108966 11-10-2015 18-10-2015 United States South  
## 6 6 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## Category Sales Quantity Total\_revenue  
## 1 Furniture 261.9600 2 523.920  
## 2 Furniture 731.9400 3 2195.820  
## 3 Office Supplies 14.6200 2 29.240  
## 4 Furniture 957.5775 5 4787.887  
## 5 Office Supplies 22.3680 2 44.736  
## 6 Furniture 48.8600 7 342.020

**3.Delete first 5 rows.**

**data = data[-1:-5, ]  
dim(data)**

## [1] 687 10

**4.Delete “Row.ID” column.**

**data$Row.ID = NULL  
head(data)**

## Order.ID Order.Date Ship.Date Country Region  
## 6 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 7 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 8 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 9 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 10 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 11 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## Category Sales Quantity Total\_revenue  
## 6 Furniture 48.860 7 342.020  
## 7 Office Supplies 7.280 4 29.120  
## 8 Technology 907.152 6 5442.912  
## 9 Office Supplies 18.504 3 55.512  
## 10 Office Supplies 114.900 5 574.500  
## 11 Furniture 1706.184 9 15355.656

**5.Reaname “Quantity” column to “Quantity\_sold”.**

**data = rename(data, Quantity\_sold=Quantity)  
head(data)**

## Order.ID Order.Date Ship.Date Country Region  
## 6 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 7 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 8 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 9 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 10 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## 11 CA-2014-115812 09-06-2014 14-06-2014 United States West  
## Category Sales Quantity\_sold Total\_revenue  
## 6 Furniture 48.860 7 342.020  
## 7 Office Supplies 7.280 4 29.120  
## 8 Technology 907.152 6 5442.912  
## 9 Office Supplies 18.504 3 55.512  
## 10 Office Supplies 114.900 5 574.500  
## 11 Furniture 1706.184 9 15355.656

**6.Create new columns for “Order\_day”,“Order\_month” and “Order\_year” from “Order.Date”.**

**library(tidyr)  
data = data %>% separate(Order.Date, into=c("Order\_day","Order\_month","Order\_year"), sep='-')  
head(data)**

## Order.ID Order\_day Order\_month Order\_year Ship.Date  
## 6 CA-2014-115812 09 06 2014 14-06-2014  
## 7 CA-2014-115812 09 06 2014 14-06-2014  
## 8 CA-2014-115812 09 06 2014 14-06-2014  
## 9 CA-2014-115812 09 06 2014 14-06-2014  
## 10 CA-2014-115812 09 06 2014 14-06-2014  
## 11 CA-2014-115812 09 06 2014 14-06-2014  
## Country Region Category Sales Quantity\_sold  
## 6 United States West Furniture 48.860 7  
## 7 United States West Office Supplies 7.280 4  
## 8 United States West Technology 907.152 6  
## 9 United States West Office Supplies 18.504 3  
## 10 United States West Office Supplies 114.900 5  
## 11 United States West Furniture 1706.184 9  
## Total\_revenue  
## 6 342.020  
## 7 29.120  
## 8 5442.912  
## 9 55.512  
## 10 574.500  
## 11 15355.656

**7.Add +2 to “Quantity” variable of South Region.**

**head(data[data$Region=="South", ])**

## Order.ID Order\_day Order\_month Order\_year Ship.Date  
## 13 CA-2017-114412 15 04 2017 20-04-2017  
## 44 CA-2017-139619 19 09 2017 23-09-2017  
## 70 CA-2016-119823 04 06 2016 06-06-2016  
## 73 US-2015-134026 26 04 2015 02-05-2015  
## 74 US-2015-134026 26 04 2015 02-05-2015  
## 75 US-2015-134026 26 04 2015 02-05-2015  
## Country Region Category Sales Quantity\_sold  
## 13 United States South Office Supplies 15.552 3  
## 44 United States South Office Supplies 95.616 2  
## 70 United States South Office Supplies 75.880 2  
## 73 United States South Furniture 831.936 8  
## 74 United States South Furniture 97.040 2  
## 75 United States South Office Supplies 72.784 1  
## Total\_revenue  
## 13 46.656  
## 44 191.232  
## 70 151.760  
## 73 6655.488  
## 74 194.080  
## 75 72.784

**data$Quantity\_sold[data$Region == "South"] <- data$Quantity\_sold[data$Region == "South"] + 2  
head(data[data$Region=="South", ])**

## Order.ID Order\_day Order\_month Order\_year Ship.Date  
## 13 CA-2017-114412 15 04 2017 20-04-2017  
## 44 CA-2017-139619 19 09 2017 23-09-2017  
## 70 CA-2016-119823 04 06 2016 06-06-2016  
## 73 US-2015-134026 26 04 2015 02-05-2015  
## 74 US-2015-134026 26 04 2015 02-05-2015  
## 75 US-2015-134026 26 04 2015 02-05-2015  
## Country Region Category Sales Quantity\_sold  
## 13 United States South Office Supplies 15.552 5  
## 44 United States South Office Supplies 95.616 4  
## 70 United States South Office Supplies 75.880 4  
## 73 United States South Furniture 831.936 10  
## 74 United States South Furniture 97.040 4  
## 75 United States South Office Supplies 72.784 3  
## Total\_revenue  
## 13 46.656  
## 44 191.232  
## 70 151.760  
## 73 6655.488  
## 74 194.080  
## 75 72.784

**WEEK-2**

**Perform below data pre-processing techniques on the sales dataset.**

1. Delete Unnecessary columns

2. Handle missing values

3. Remove duplicate data

4. Create Country, Order\_year and Order\_Id from Order\_Id variable

5. Remove outliers from sales column

# load sales dataset  
**data = read.csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/sales\_data\_preprocess.csv",fileEncoding = "UTF-8-BOM")**#examin the data  
**head(data)**

## Row.ID Order.ID Order.Date Ship.Date Region Category  
## 1 1 CA-2016-152156 08-11-2016 11-11-2016 South Furniture  
## 2 2 CA-2016-152156 08-11-2016 11-11-2016   
## 3 3 CA-2016-138688 12-06-2016 16-06-2016 West Office Supplies  
## 4 4 US-2015-108966 11-10-2015 18-10-2015 South Furniture  
## 5 5 US-2015-108966 11-10-2015 18-10-2015 South Office Supplies  
## 6 6 CA-2014-115812 09-06-2014 14-06-2014 West   
## Sales Quantity  
## 1 261.9600 2  
## 2 731.9400 3  
## 3 14.6200 2  
## 4 957.5775 5  
## 5 22.3680 2  
## 6 48.8600 7

**1. Delete Unnecessary columns**

# Row.ID not required for analysis.Delete Row.ID  
**data$Row.ID = NULL  
head(data)**

## Order.ID Order.Date Ship.Date Region Category Sales  
## 1 CA-2016-152156 08-11-2016 11-11-2016 South Furniture 261.9600  
## 2 CA-2016-152156 08-11-2016 11-11-2016 731.9400  
## 3 CA-2016-138688 12-06-2016 16-06-2016 West Office Supplies 14.6200  
## 4 US-2015-108966 11-10-2015 18-10-2015 South Furniture 957.5775  
## 5 US-2015-108966 11-10-2015 18-10-2015 South Office Supplies 22.3680  
## 6 CA-2014-115812 09-06-2014 14-06-2014 West 48.8600  
## Quantity  
## 1 2  
## 2 3  
## 3 2  
## 4 5  
## 5 2  
## 6 7

**2. Handle missing values**

#replace blank values with NA  
**data[data == ""] = NA**  
head(data)

## Order.ID Order.Date Ship.Date Region Category Sales  
## 1 CA-2016-152156 08-11-2016 11-11-2016 South Furniture 261.9600  
## 2 CA-2016-152156 08-11-2016 11-11-2016 <NA> <NA> 731.9400  
## 3 CA-2016-138688 12-06-2016 16-06-2016 West Office Supplies 14.6200  
## 4 US-2015-108966 11-10-2015 18-10-2015 South Furniture 957.5775  
## 5 US-2015-108966 11-10-2015 18-10-2015 South Office Supplies 22.3680  
## 6 CA-2014-115812 09-06-2014 14-06-2014 West <NA> 48.8600  
## Quantity  
## 1 2  
## 2 3  
## 3 2  
## 4 5  
## 5 2  
## 6 7

# find the percentage of missing values column wise  
**missing\_percentage = colSums(is.na(data))/nrow(data)\*100**  
print(missing\_percentage)

## Order.ID Order.Date Ship.Date Region Category Sales   
## 0.000000 0.000000 0.000000 3.890490 5.187320 1.873199   
## Quantity   
## 0.000000

# replace Sales missing values by mean()  
  
#calculate mean of sales  
**mean\_sales = mean(data$Sales, na.rm = TRUE)**  
  
#replace by mean  
**data$Sales = replace(data$Sales, is.na(data$Sales), mean\_sales)**

**Mode =** function**(x){  
 a = table(x)  
 mode\_value = names(a[which.max(a)])  
 return(mode\_value)  
}**

# replace Region and Category missing values by mode  
  
# find the mode of Region and replace  
**region\_mode = Mode(data$Region)**  
print(region\_mode)

## [1] "West"

**data$Region = replace(data$Region, is.na(data$Region), region\_mode)**  
#find the mode of Category and replace  
**category\_mode = Mode(data$Category)**print(category\_mode)

## [1] "Office Supplies"

**data$Category = replace(data$Category, is.na(data$Category), category\_mode)**

**3. Remove duplicate data**

# Using unique() in Base R  
dim(data)

## [1] 694 7

data = unique(data)  
dim(data)

## [1] 690 7

**4. Create Country, Order\_year and Id from Order\_Id variable**

library(tidyr)  
  
data = data %>% separate(Order.ID, into = c("Country","Order\_year","Id"), sep = "-")  
data$Order.ID = NULL  
head(data)

## Country Order\_year Id Order.Date Ship.Date Region Category  
## 1 CA 2016 152156 08-11-2016 11-11-2016 South Furniture  
## 2 CA 2016 152156 08-11-2016 11-11-2016 West Office Supplies  
## 3 CA 2016 138688 12-06-2016 16-06-2016 West Office Supplies  
## 4 US 2015 108966 11-10-2015 18-10-2015 South Furniture  
## 5 US 2015 108966 11-10-2015 18-10-2015 South Office Supplies  
## 6 CA 2014 115812 09-06-2014 14-06-2014 West Office Supplies  
## Sales Quantity  
## 1 261.9600 2  
## 2 731.9400 3  
## 3 14.6200 2  
## 4 957.5775 5  
## 5 22.3680 2  
## 6 48.8600 7

**5. Remove outliers from sales column**

dim(data)

## [1] 690 9

**Q1 = quantile(data$Sales, 0.25)  
Q3 = quantile(data$Sales, 0.75)  
  
IQR = Q3-Q1  
  
lower\_bound = Q1 - 1.5\*IQR  
upper\_bound = Q3 + 1.5\*IQR  
  
outliers = data$Sales < lower\_bound | data$Sales > upper\_bound**  
print(dim(data[outliers, ]))

#remove outlier rows  
**data = data[!outliers, ]**  
dim(data)

## [1] 690 9

## [1] 80 9

## [1] 610 9

**WEEK-3**

**Conduct a complete data analysis on a given student results dataset and derive insights using the ggplot2 package in R.**

# load sales dataset  
**data = read.csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/students\_marks.csv",fileEncoding = "UTF-8-BOM")**  
#examin the data  
**head(data)**

## id Name Gender Age Section Science English History Maths  
## 1 1 Bronnie Female 13 C 21 81 62 49  
## 2 2 Lemmie Male 15 B 29 41 17 40  
## 3 3 Danya Female 14 C 12 87 16 96  
## 4 4 Denna Female 14 B 15 53 82 33  
## 5 5 Jocelin Male 14 A 43 6 3 21  
## 6 6 Malissa Female 14 C 98 51 85 76

**str(data)**

## 'data.frame': 250 obs. of 9 variables:  
## $ id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Name : Factor w/ 247 levels "Abel","Adah",..: 47 148 68 73 132 157 117 36 62 228 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 1 2 1 1 2 1 1 2 2 2 ...  
## $ Age : int 13 15 14 14 14 14 14 14 15 15 ...  
## $ Section: Factor w/ 3 levels "A","B","C": 3 2 3 2 1 3 2 2 1 3 ...  
## $ Science: int 21 29 12 15 43 98 38 25 39 35 ...  
## $ English: int 81 41 87 53 6 51 74 51 16 25 ...  
## $ History: int 62 17 16 82 3 85 54 41 22 37 ...  
## $ Maths : int 49 40 96 33 21 76 60 80 49 27 ...

# find the percentage of missing values column wise  
**missing\_percentage = colSums(is.na(data))/nrow(data)\*100  
print(missing\_percentage)**

## id Name Gender Age Section Science English History Maths   
## 0 0 0 0 0 0 0 0 0

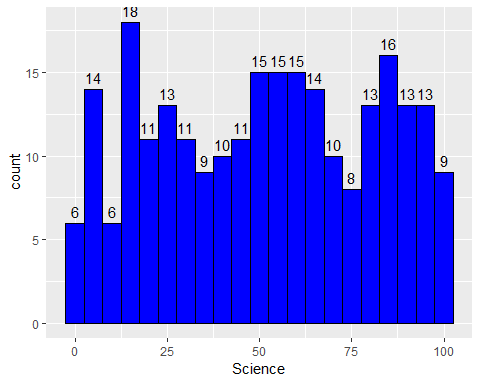
**1. Distribution of Science and English Marks**

library(ggplot2)

# Assuming 'data' is your dataframe  
**ggplot(data, aes(x = Science)) +   
 geom\_histogram(binwidth = 5, fill = 'blue', color = 'black') +   
 stat\_bin(binwidth = 5, geom = "text", aes(label = ..count..), vjust = -0.5) +**

**ggtitle("Distribution of Science Marks")**

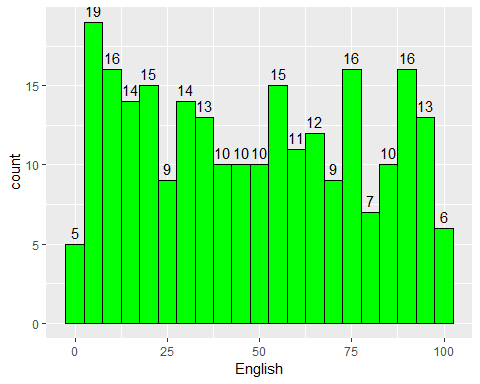
**"Distribution of Science Marks"**



**ggplot(data, aes(x = English)) +   
 geom\_histogram(binwidth = 5, fill = 'green', color = 'black') +   
 stat\_bin(binwidth = 5, geom = "text", aes(label = ..count..), vjust = -0.5) +**

**ggtitle("Distribution of English Marks")**

**Distribution of English Marks**



##

**Answer the below questions from the above histogram plots:**

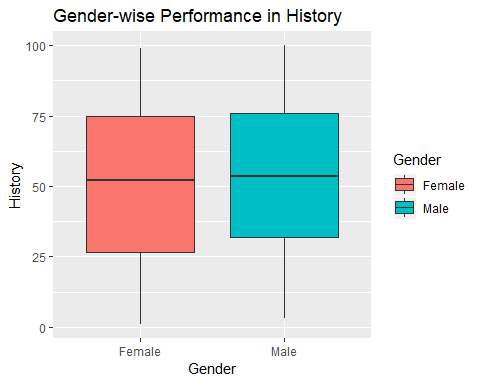
1. How many students are there with science marks > 75 (approximately)?

2. How many students are there with English marks > 75 (approximately)?

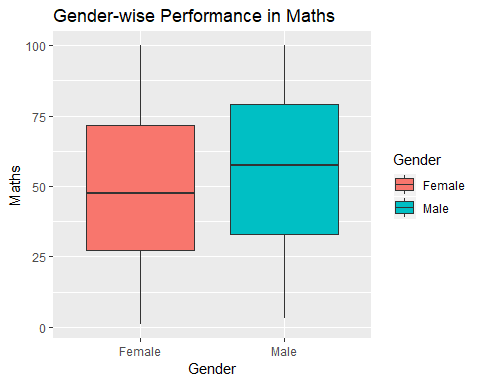
3. How many students are there with science marks < 35 (approximately)?

**2. Gender-wise Performance of Maths and History marks**

**ggplot(data, aes(x = Gender, y = History, fill = Gender)) +  
 geom\_boxplot() +   
 ggtitle("Gender-wise Performance in History")**



**ggplot(data, aes(x = Gender, y = Maths, fill = Gender)) +  
 geom\_boxplot() +   
 ggtitle("Gender-wise Performance in Maths")**

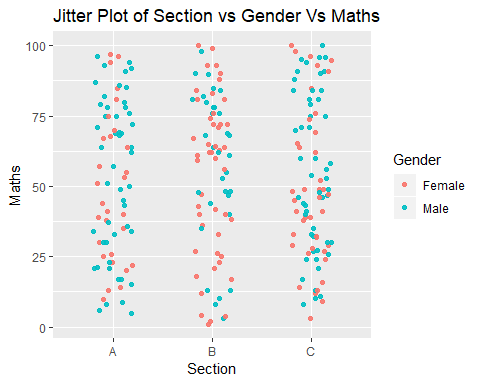


**Answer the below questions from above box plots:**

1. Which gender has the highest average math score?
2. Are there any outliers in the math marks?
3. Which gender performed well in the math exam?

**3. Section and gender wise Performance of maths subject**

**ggplot(data, aes(x = Section, y = Maths, color = Gender)) +  
 geom\_jitter(width = 0.2, height = 0.1, alpha = 0.9) +  
 labs(title = "Jitter Plot of Section vs Gender Vs Maths",  
 x = "Section",  
 y = "Maths")**



**Answer the below questions from above jitter plot:**

1. Draw jitter plot for remaining subjects also.

2. Which gender from what section performed well in the math,science,english and History exams?

1. **Calculate total marks and analyze them with id,section and gender**

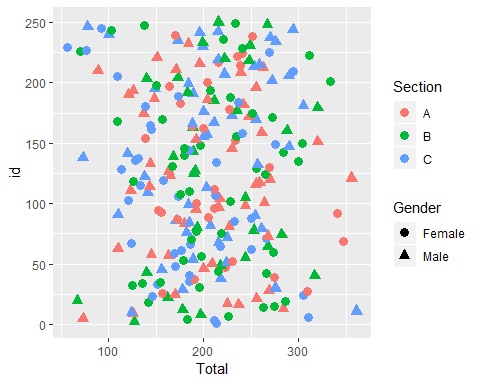
**library(dplyr)**

# create total column

**data = mutate(data, Total = Maths + Science + English + History)  
head(data)**

## id Name Gender Age Section Science English History Maths Total  
## 1 1 Bronnie Female 13 C 21 81 62 49 213  
## 2 2 Lemmie Male 15 B 29 41 17 40 127  
## 3 3 Danya Female 14 C 12 87 16 96 211  
## 4 4 Denna Female 14 B 15 53 82 33 183  
## 5 5 Jocelin Male 14 A 43 6 3 21 73  
## 6 6 Malissa Female 14 C 98 51 85 76 310

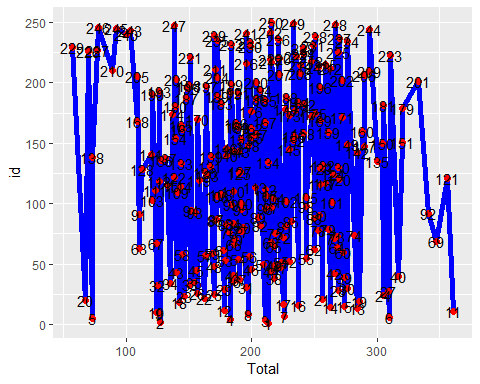
**ggplot(data, aes(x = Total,y = id, shape = Gender, color = Section)) +   
 geom\_point(size = 3)**



**Answer the below questions from above scatter plot:**

1. student from which section and gender got the highest total marks.
2. student from which section and gender got the least total marks.
3. **Line plot between id and total marks**

**ggplot(data, aes(x = Total, y = id)) +   
 geom\_line(size = 2, color = "blue") +  
 geom\_point(color = "red", size = 2) +  
 geom\_text(aes(label=id))**



**Answer the below questions from above line plot:**

What is the ID of the student who got the highest marks?

What is the ID of the student who got the least marks?

**WEEK-5**

**Merge two Data Frames and apply various data manipulation techniques.**

**Merge two Data Frames**

**import pandas as pd**

# read the files

**data1 = pd.read\_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/salesdata.csv")**

**data1.head()**

**Order**

**ID**

CA-

**Order Date**

08-11-

**Ship Date**

11-

**Customer**

**ID**

CG-

**Country City State Postal Code**

United

**Region Category Sales Quantity Discount Profit**

**0** 2016-

152156

CA-

**1** 2016-

152156

2016

08-11-

2016

11-

2016

11-

11-

2016

12520

CG- 12520

States Henderson Kentucky 42420 South Furniture 261.9600 2.0 0.00 41.9136

United Henderson Kentucky 42420 South Furniture 731.9400 3.0 0.00 219.5820 States

**data2 = pd.read\_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/returnsdata.csv")**

**data2.head()**

**Returned Order ID**

**0** Yes CA-2017-153822

**1** Yes CA-2017-129707

**2** Yes CA-2014-152345

**3** Yes CA-2015-156440

**4** Yes US-2017-155999

# merging two dataframes using inner join

**data = pd.merge(data1, data2, on='Order ID', how='inner')**

**data.head()**

**Customer**

|  |  |  |
| --- | --- | --- |
| **Order** | **Order** | **Ship** |
| **ID** | **Date** | **Date** |
| CA- | 27- | 01- |
| **0** 2014- | 08- | 09- |
| 143336 | 2014 | 2014 |
| CA- | 27- | 01- |
| **1** 2014- | 08- | 09- |
| 143336 | 2014 | 2014 |

**ID**

**Country City State Postal**

**Code**

**Region Category Sales Quantity Discount Profit Returne**

ZD- 21925

United States

San California 94109 West Francisco

Office 8.56 2.0 0.0 2.4824 Ye

Supplies

**data.shape**

(104, 15)

ZD- 21925

United States

San California 94109 West Technology 213.48 3.0 0.2 16.0110 Ye Francisco

**Different data manipulation techniques**

**1. Delete rows**

# Delete 2nd and 41th rows

**data = data.drop([1,40])**

**data.shape**

(102, 15)

**2.Delete columns 'Customer ID', 'Postal Code'.**

**data = data.drop(['Customer ID', 'Postal Code'], axis=1)**

**data.head()**

**Order ID Order Date**

**Ship Date**

**Country City State Region Category Sales Quantity Discount Profit Returned**

**0** CA-2014-

143336

27-08-

2014

01-09-

2014

United States

San Francisco

California West Office Supplies

8.56 2.0 0.0 2.4824 Yes

**2** CA-2014-

143336

27-08-

2014

01-09-

2014

United States

San Francisco

California West NaN 22.72 4.0 0.2 7.3840 Yes

**3** CA-2016- 17-06- 18-06- United Troy New York East Office 208.56 6.0 0.0 52.1400 Yes

**3. Modify the values**

# Round the 'Profit' column to 2 decimal places

**data['Profit'] = data['Profit'].round(2)**

**data.head()**

**Order ID Order Date**

**Ship Date**

**Country City State Region Category Sales Quantity Discount Profit Returned**

**0** CA-2014-

143336

27-08-

2014

01-09-

2014

United States

San Francisco

California West Office Supplies

8.56 2.0 0.0 2.48 Yes

**2** CA-2014-

143336

27-08-

2014

01-09-

2014

United States

San Francisco

California West NaN 22.72 4.0 0.2 7.38 Yes

**3** CA-2016- 17-06- 18-06- United Troy New York East Office 208.56 6.0 0.0 52.14 Yes

**4.Create new column from existing columns**

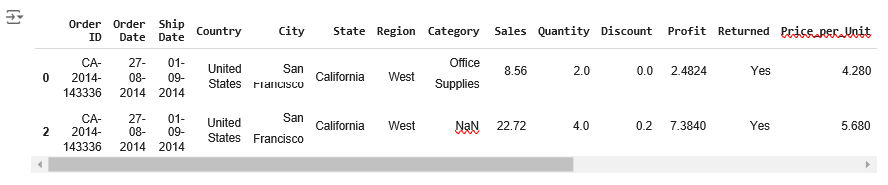
# Create 'Price\_per\_Unit' column

**data['Price\_per\_Unit'] = data['Sales'] / data['Quantity']**

# Extract the year from 'Order ID'

**data['OrYear'] = df['Order ID'].str.split('-').str[1]**

**data.head()**

****

**5. Handle missing data**

**import numpy as np**

# replace blank strings with 'NaN' **data = data.replace('',np.nan)**

# calculate % of missing values columnwise

**missing\_percentage = data.isna().sum()/len(data)\*100 missing\_percentage**

|  |  |
| --- | --- |
| Order ID | 0.000000 |
| Order Date | 0.000000 |
| Ship Date | 0.000000 |
| Country | 0.000000 |
| City | 0.000000 |
| State | 0.000000 |
| Region | 0.000000 |
| Category | 6.862745 |
|  |  |
| Sales | 0.000000 |
| Quantity | 5.882353 |
| Discount | 0.000000 |
| Profit | 0.000000 |
| Returned | 0.000000 |
| Price\_per\_Unit  dtype: float64 | 5.882353 |

# fill the missing values of Category,Quantity and Price\_per\_Unit columns

**data['Category'] = data['Category'].fillna(data['Category'].mode()[0])**

**data['Quantity'] = data['Quantity'].fillna(data['Quantity'].mean())**

**data['Price\_per\_Unit'] = data['Price\_per\_Unit'].fillna(data['Price\_per\_Unit'].mean())**

# calculate % of missing values columnwise

**missing\_percentage = data.isna().sum()/len(data)\*100 missing\_percentage**

 Order ID 0.0

Order Date 0.0

Ship Date 0.0

Country 0.0

City 0.0

State 0.0

Region 0.0

Category 0.0

Sales 0.0

Quantity 0.0

Discount 0.0

Profit 0.0

Returned 0.0

Price\_per\_Unit 0.0

dtype: float64

**data.shape**

(102, 14)

**6. Remove duplicate entries**

**data = data.drop\_duplicates()**

**data.shape**

(102, 14)

No duplicates rows

**WEEK-6**

**Use the Python ‘Matplotlib’ to perform a thorough data analysis and extract insights from a given Housing dataset.**

**import pandas as pd**

**# read the Housing dataset**

**data = pd.read\_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/Housing.csv")**

**data.head()**



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **price** | **area** | **bedrooms** | **bathrooms** | **stories** | **mainroad** | **guestroom** | **basement** | **hotwaterheating** | **airconditioning** | **parking** | **furnishing** |
| **0** 6195000 | 5500 | 3 | 2 | 4 | yes | yes | no | no | yes | 1 | semi-fu |
| **1** 6195000 | 6350 | 3 | 2 | 3 | yes | yes | no | no | yes | 0 | fu |
| **2** 6195000 | 5500 | 3 | 2 | 1 | yes | yes | yes | no | no | 2 | fu |
| **3** 6160000 | 4500 | 3 | 1 | 4 | yes | no | no | no | yes | 0 | unfu |
| **4** 6160000 | 5450 | 4 | 2 | 1 | yes | no | yes | no | yes | 0 | semi-fu |
|  |  |  |  |  |  |  |  |  |  |  |  |

**# check the shape of dataset data.shape**

 (299, 12)

**data.info()**

 <class 'pandas.core.frame.DataFrame'> RangeIndex: 299 entries, 0 to 298

Data columns (total 12 columns):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # |  | Column | Non-Null Count |  | Dtype |
| 0 |  | price | 299 non-null |  | int64 |
| 1 |  | area | 299 non-null |  | int64 |
| 2 |  | bedrooms | 299 non-null |  | int64 |
| 3 |  | bathrooms | 299 non-null |  | int64 |
| 4 |  | stories | 299 non-null |  | int64 |
| 5 |  | mainroad | 299 non-null |  | object |
| 6 |  | guestroom | 299 non-null |  | object |
| 7 |  | basement | 299 non-null |  | object |
| 8 |  | hotwaterheating | 299 non-null |  | object |
| 9 |  | airconditioning | 299 non-null |  | object |
| 10 |  | parking | 299 non-null |  | int64 |
| 11 |  | furnishingstatus | 299 non-null |  | object |

dtypes: int64(6), object(6) memory usage: 28.2+ KB

# check the missing values **data.isnull().sum()**

 price 0

area 0

bedrooms 0

bathrooms 0

stories 0

mainroad 0

guestroom 0

basement 0

hotwaterheating 0

airconditioning 0

parking 0

furnishingstatus 0

dtype: int64

**1. Box plot for price**

**import matplotlib.pyplot as plt**

# Create box plot for the 'price' column **plt.boxplot(data['price'])**

# Add title and labels

**plt.title('Box Plot of price', fontsize=14) plt.ylabel('House price', fontsize=12)**

# Display the plot

**plt.show()**



**Average house price = 4500000**

**There are no outliers**

**range of house price = around 4000000 to 5400000**

**2. Histogram for area**

#Create histogram for area

**plt.hist(data['area'], bins=5, edgecolor='black', color='skyblue')**

# Add labels and title

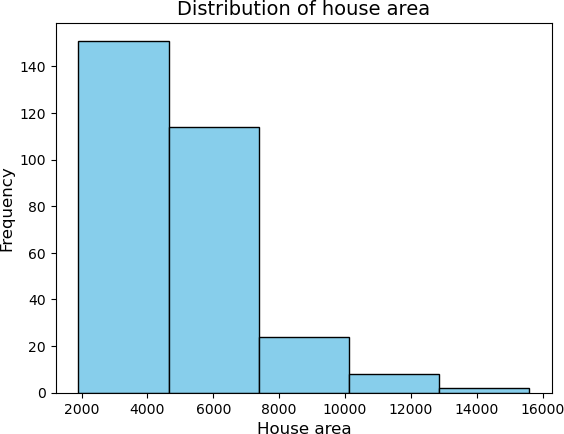
**plt.title('Distribution of house area', fontsize=14) plt.xlabel('House area', fontsize=12)**

**plt.ylabel('Frequency', fontsize=12)**

# Display the plot

**plt.show()**





* **most of the house area is in the range from 2000 sqft to 7200 sqft**

**3. Bar chart between mainroad and price**

# Group data by 'mainroad' and sum the price

**grouped\_data = data.groupby('mainroad')['price'].sum()**

**plt.bar(grouped\_data.index, grouped\_data.values, color='orange')**

# Add labels and title

**plt.xlabel('mainroad facing', fontsize=12) plt.ylabel('total price', fontsize=12)**

**plt.title('Bar chart between mainroad and price', fontsize=14)**

# Add data labels on top of the bars

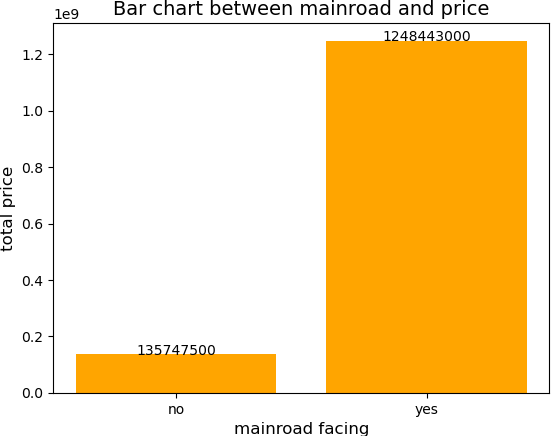
**for i, value in enumerate(grouped\_data.values):**

**plt.text(i, value, str(value), ha='center', fontsize=10)**

# Display the plot

**plt.show()**





* **The houses facing the main road are the most expensive.**

**4. box plot for parking vs price**

# Create box plot for Sales grouped by Region

**data.boxplot(column='price', by='parking', grid=False, patch\_artist=True)**

# Add title and labels

**plt.title('box plot for parking vs price', fontsize=14)**

# Remove default 'Boxplot grouped by Region'

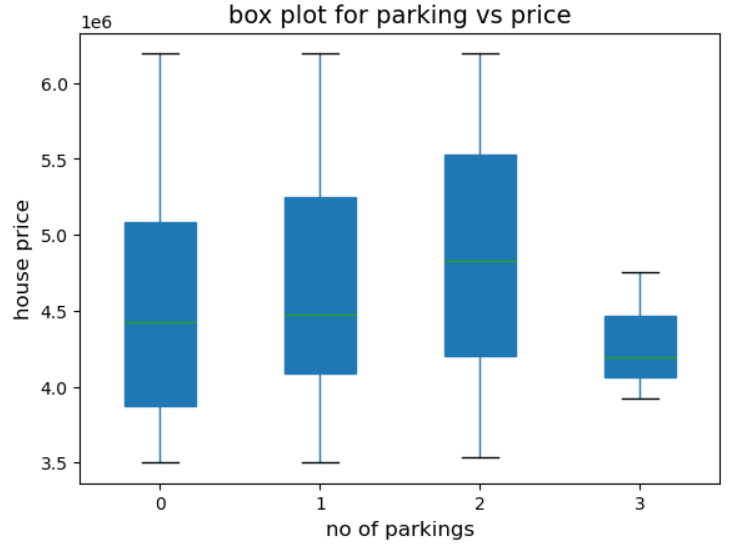
**plt.suptitle('')**

**plt.xlabel('no of parkings', fontsize=12)**

**plt.ylabel('house price', fontsize=12)**

# Display the plot

**plt.show()**



* **The houses with 2 parking spaces are the most expensive.**

**5. jitter plot for furnishingstatus vs price**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

# Create a jitter plot Region vs Sales

**sns.stripplot(x=data['furnishingstatus'], y=data['price'], jitter=0.3)**

# Add labels and title

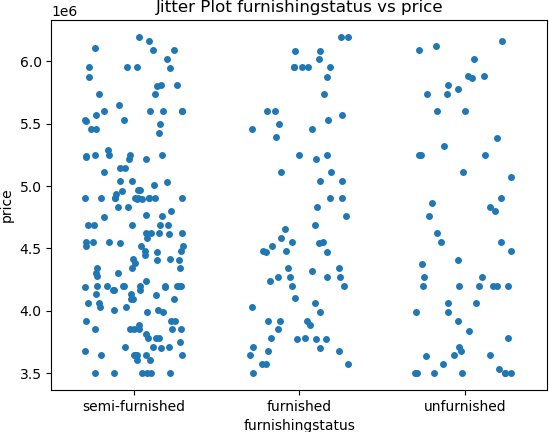
**plt.xlabel('furnishingstatus') plt.ylabel('price')**

**plt.title('Jitter Plot furnishingstatus vs price')**

# Show the plot

**plt.show()**





**No insights**

**6. scatter plot between area and price**

# Create scatter plot

**plt.scatter(data['area'], data['price'], color='blue')**

# Add labels and title

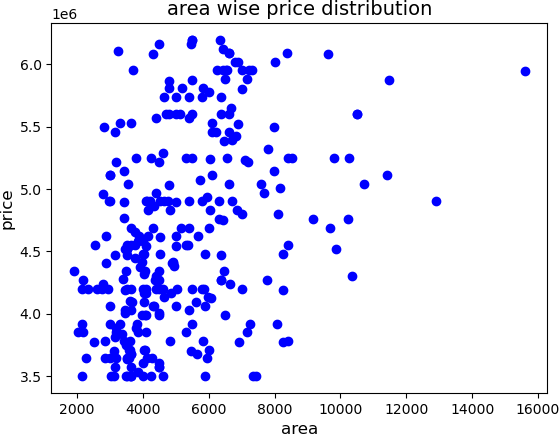
**plt.title('area wise price distribution', fontsize=14) plt.xlabel('area', fontsize=12)**

**plt.ylabel('price', fontsize=12)**

# Display the plot

**plt.show()**





* **There exists a bit positive relation between area and price**

**7. subplots among guestroom vs basement vs price**

**data.guestroom.unique()**

 array(['yes', 'no'], dtype=object)

**data.basement.unique()**

 array(['no', 'yes'], dtype=object)

**import matplotlib.pyplot as plt**

# Create a figure with four subplots sharing both x and y axes

**fig, axes = plt.subplots(2, 2, sharex=True, sharey=True, figsize=(10, 10))**

# Get the unique regions from the data

**guestrooms = data['guestroom'].unique()**

**basements = data['basement'].unique()**

# Plot sales by country for each region

**for i, x in enumerate(guestrooms):**

**for j, y in enumerate(basements):**

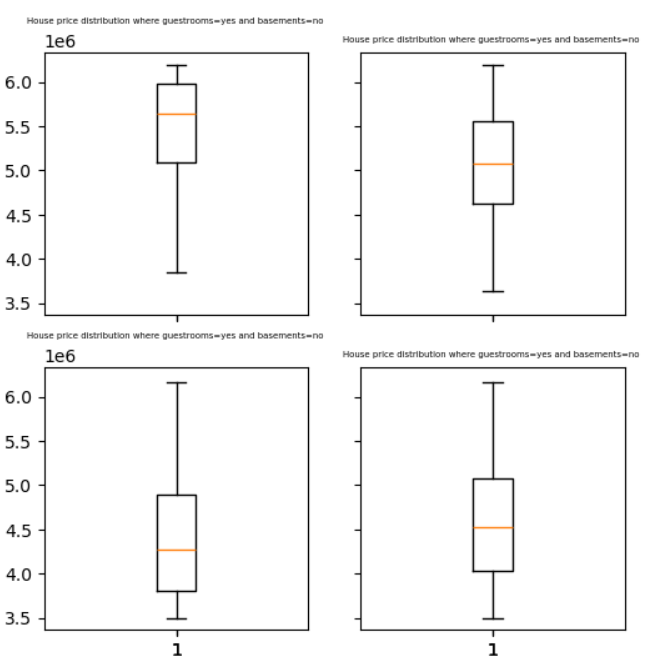
**g\_data = data[(data['guestroom'] == x) & (data['basement'] == y)]**

**axes[i,j].boxplot(g\_data['price'])**

**axes[i][j].set\_title(f'House price distribution where guestrooms={x} and basements={y} ',size = 8)**

# Display the plots

**plt.show()**



**WEEK-7**

**List 5 findings from the state\_wise\_covid dataset by making an in-depth data analysis with the help of the ‘Matplotlib’ library.**

**import pandas as pd**

# read the Housing dataset

**data = pd.read\_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/state\_wise\_covid\_data.csv")**

**data.head()**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **State** | **Confirmed** | **Recovered** | **Deaths** | **Active** | **State\_code** |
| **0** Total | 34285612 | 33661339 | 458470 | 152606 | TT |
| **1** Andaman and Nicobar Islands | 7651 | 7518 | 129 | 4 | AN |
| **2** Andhra Pradesh | 2066450 | 2047722 | 14373 | 4355 | AP |
| **3** Arunachal Pradesh | 55155 | 54774 | 280 | 101 | AR |
| **4** Assam | 610645 | 600974 | 5997 | 2327 | AS |

* **The first row in the dataset is the summary row, which is not required, so remove it.**
* **Row 31 has unassigned state, so remove**

**data = data.drop([0,31])**

**data.head()**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **State** | **Confirmed** | **Recovered** | **Deaths** | **Active** | **State\_code** |
| **1** Andaman and Nicobar Islands | 7651 | 7518 | 129 | 4 | AN |
| **2** Andhra Pradesh | 2066450 | 2047722 | 14373 | 4355 | AP |
| **3** Arunachal Pradesh | 55155 | 54774 | 280 | 101 | AR |
| **4** Assam | 610645 | 600974 | 5997 | 2327 | AS |
| **5** Bihar | 726098 | 716390 | 9661 | 46 | BR |

# check the shape of dataset **data.shape**

 (36, 6)

**data.info()**

 <class 'pandas.core.frame.DataFrame'> Index: 36 entries, 1 to 37

Data columns (total 6 columns):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # |  | Column | Non-Null Count |  | Dtype |
| 0 |  | State | 36 non-null |  | object |
| 1 |  | Confirmed | 36 non-null |  | int64 |
| 2 |  | Recovered | 36 non-null |  | int64 |
| 3 |  | Deaths | 36 non-null |  | int64 |
| 4 |  | Active | 36 non-null |  | int64 |
| 5 |  | State\_code | 36 non-null |  | object |

dtypes: int64(4), object(2) memory usage: 2.0+ KB

# check the missing values **data.isnull().sum()**

 State 0

Confirmed 0

Recovered 0

Deaths 0

Active 0

State\_code 0

dtype: int64

**1. Bar plot showing confirmed cases for each state**

import matplotlib.pyplot as plt

# Plotting a bar chart for confirmed cases by state

**plt.figure(figsize=(15, 8))**

**plt.bar(data['State\_code'], data['Confirmed'], color='skyblue')**

**plt.xlabel('State code')**

**plt.ylabel('No of confirmed cases')**

**plt.title('State-wise Confirmed COVID-19 Cases')**

**plt.tight\_layout()**

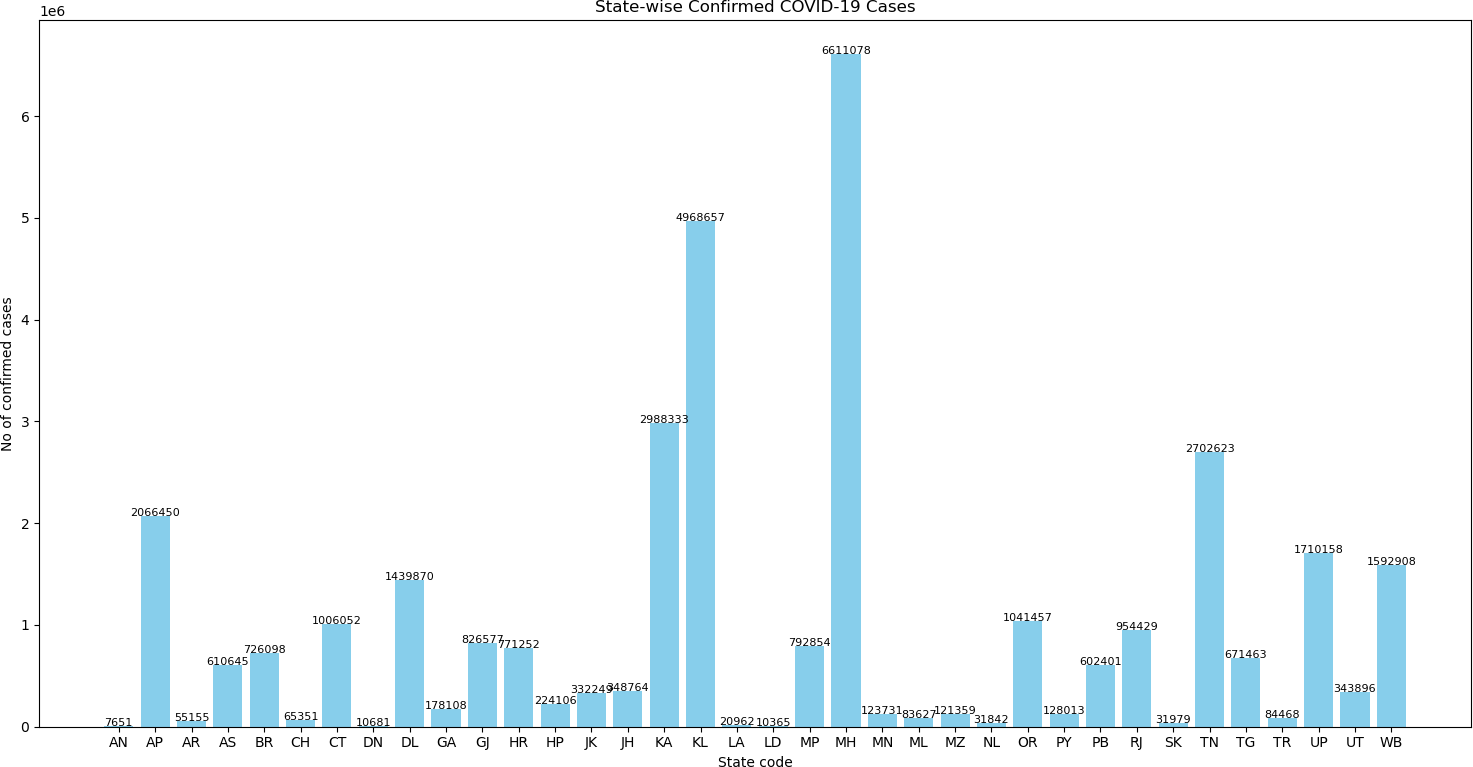
# Add data labels on top of the bars

**for i, value in enumerate(data['Confirmed']):**

**plt.text(i, value, str(value), ha='center', fontsize=8)**

# Display the plot

**plt.show()**



* **Maharashtra state had the highest number of confirmed COVID cases.**
* **Andaman and Nicobar Islands had the lowest number of confirmed COVID cases.**

**2.Bar plot showing recovered % for each state**

# Adding a new column for Recovery Rate

**data['Recovery Rate (%)'] = (data['Recovered'] / data['Confirmed']) \* 100**

# Applying the round function to the 'Recovery Rate (%)' column and limiting it to 1 decimal place

**data['Recovery Rate (%)'] = data['Recovery Rate (%)'].round(1)**

# Plotting a bar chart for confirmed cases by state

**plt.figure(figsize=(15, 8))**

**plt.bar(data['State\_code'], data['Recovery Rate (%)'], color='red') plt.xlabel('State code')**

**plt.ylabel('Recovered %')**

**plt.title('State-wise recovered % of COVID-19 Cases')**

**plt.tight\_layout()**

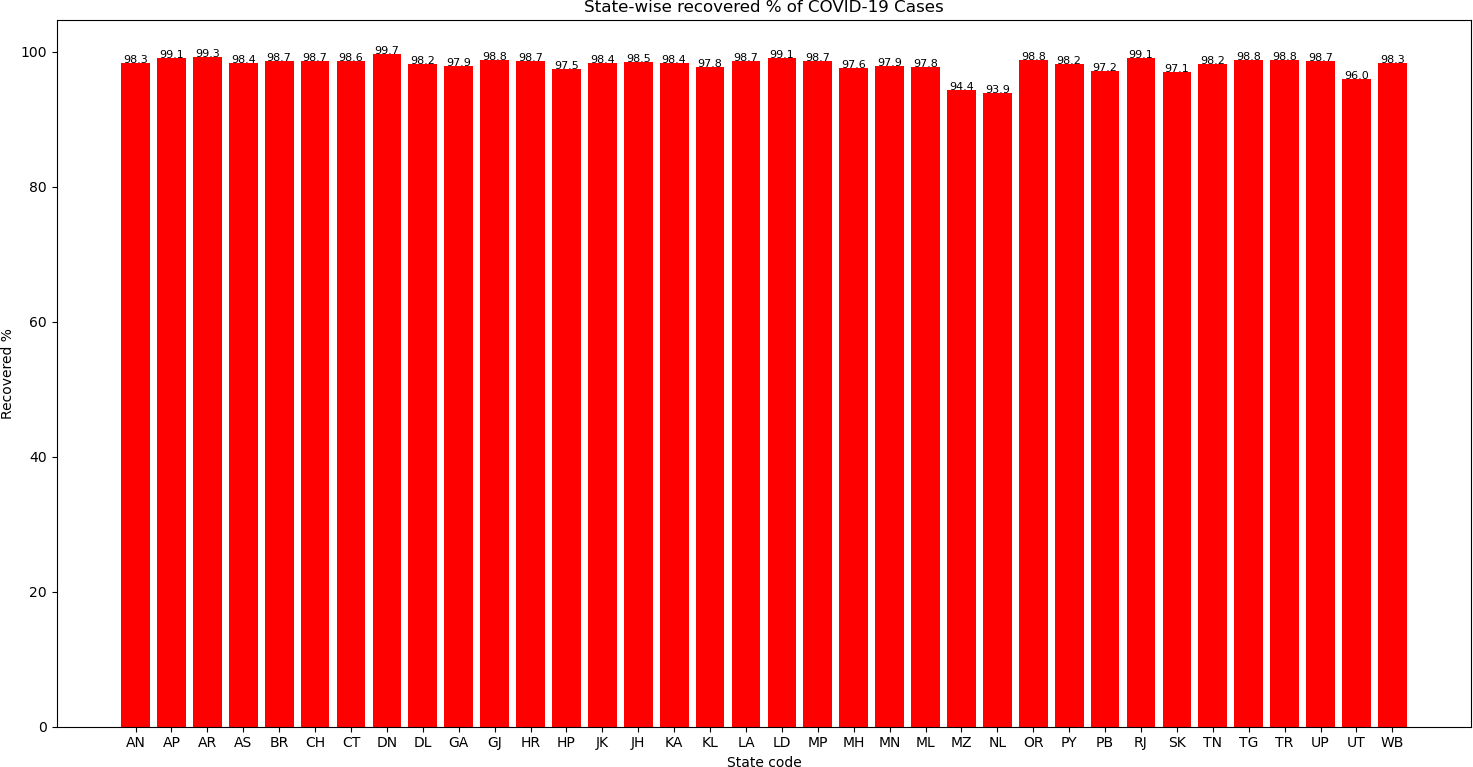
# Add data labels on top of the bars

**for i, value in enumerate(data['Recovery Rate (%)']):**

**plt.text(i, value, str(value), ha='center', fontsize=8)**

# Display the plot

**plt.show()**

* **DN state had a highest recovered rate = 99.7%**
* **NL state had a least recovered rate = 93.9 %**

**3. Line chart between State and Death %**

# Adding a new column for Death Rate (%)

**data['Death Rate (%)'] = (data['Deaths'] / data['Confirmed']) \* 100**

**data['Death Rate (%)'] = data['Death Rate (%)'].round(1)**

#line plot

**plt.figure(figsize=(15, 8))**

**plt.plot(data['State\_code'], data['Death Rate (%)'], marker='o', color='red') plt.xlabel('State code')**

**plt.ylabel('Death Rate (%)')**

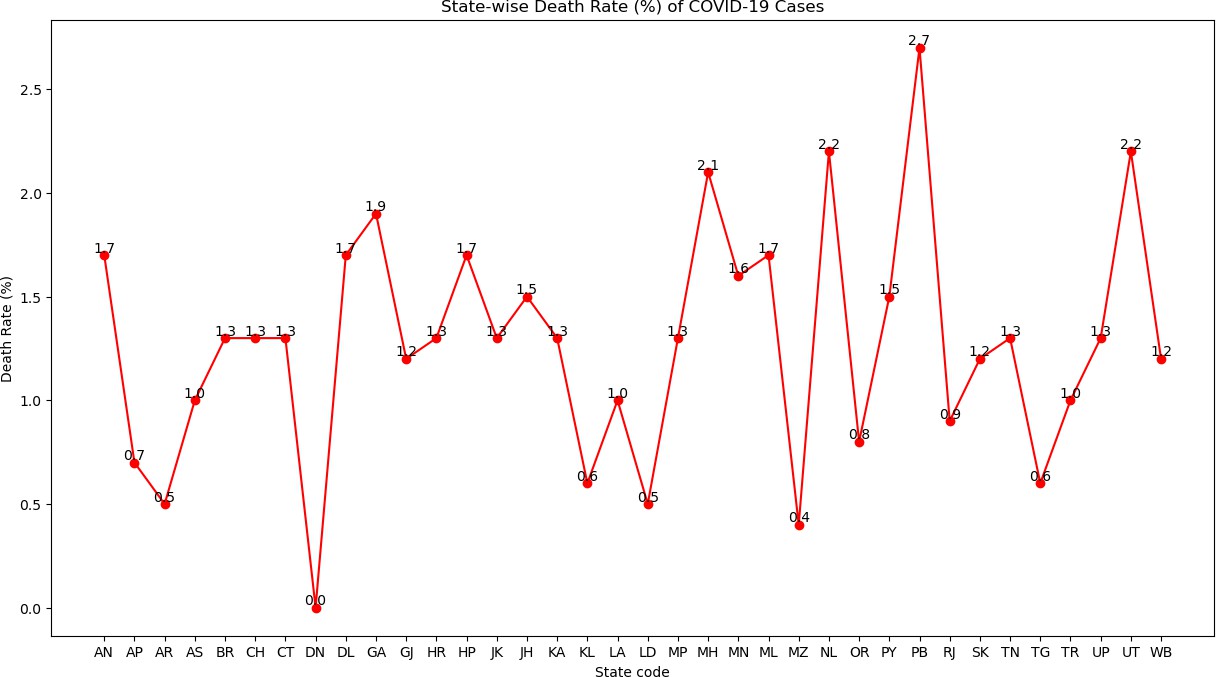
**plt.title('State-wise Death Rate (%) of COVID-19 Cases')**

# Add data labels

**for i, value in enumerate(data['Death Rate (%)']):**

**plt.text(i, value, str(value), ha='center',va='bottom', fontsize=10)**

#display the plot **plt.show()**

* **PB state had a highest death % = 2.7%**
* **DN state had a least death % = 0 %**

**4. Pie plot showing active cases % state wise**

# Creating a pie chart for Active Cases across states **plt.figure(figsize=(8,8))**

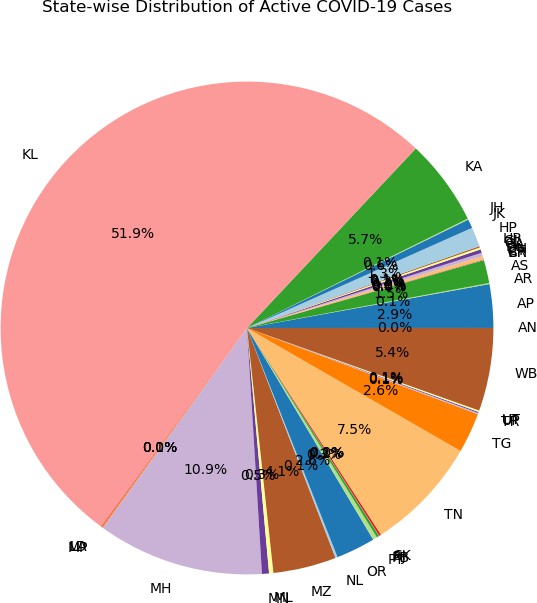
# Creating a pie chart for state wise active cases

**plt.pie(data['Active'], labels=data['State\_code'], autopct='%1.1f%%', colors=plt.cm.Paired.colors)**

# Adding title

**plt.title('State-wise Distribution of Active COVID-19 Cases')**

# Display the pie chart **plt.show()**

* **KL state had a highest active cases**

**5. Scatter plot for confirmed vs recovered**

# Create scatter plot

**plt.figure(figsize=(15,8))**

**plt.scatter(data['Confirmed'], data['Deaths'], color='blue')**

# Add labels and title

**plt.xlabel('confirmed')**

**plt.ylabel('Deaths')**

**plt.title('Scatter plot for confirmed vs Deaths')**

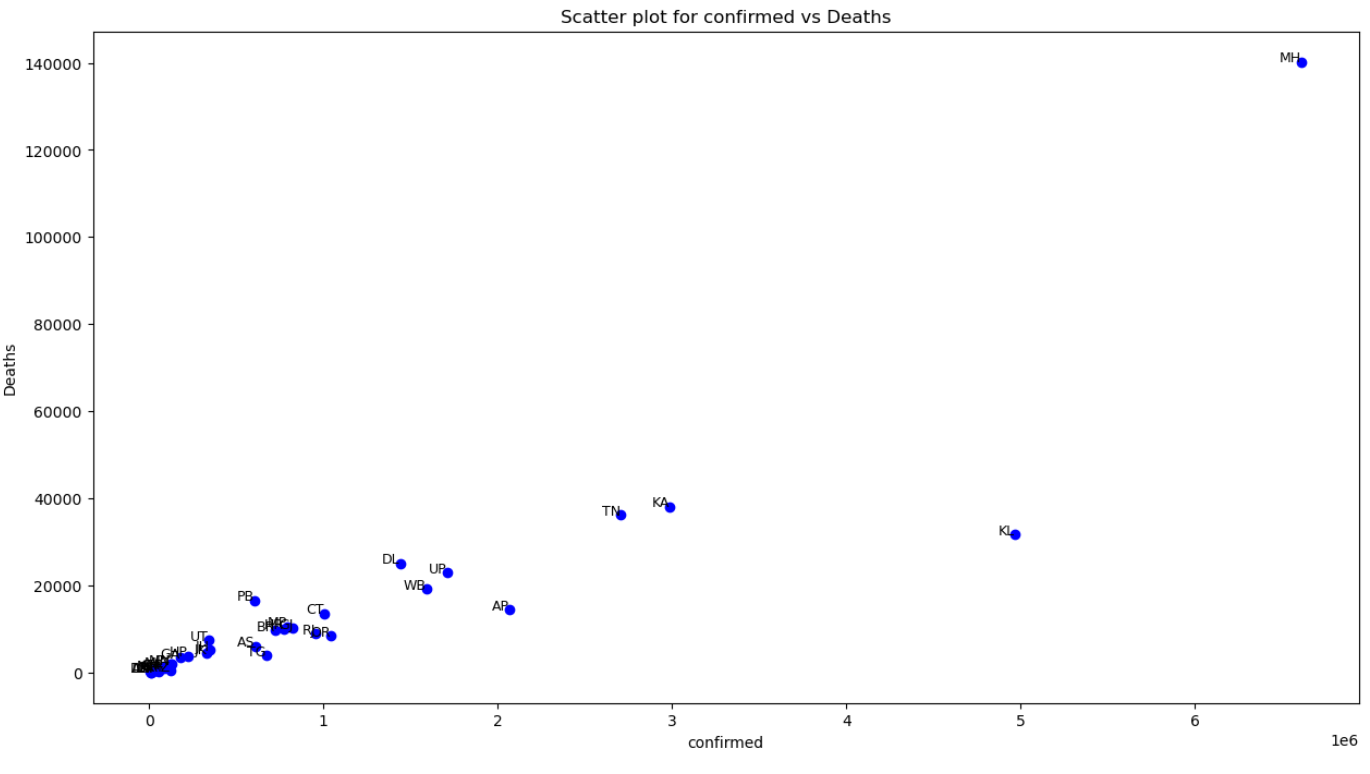
# Annotating state codes

**for i in range(len(data)):**

**plt.text(data['Confirmed'].iloc[i], data['Deaths'].iloc[i], data['State\_code'].iloc[i], fontsize=9, ha='right')**

# Show the plot

**plt.show()**

****

* **There exists a +ve relation between confirmed and deaths variables except for KL state.**
* **From plots 4 and 5 , we can derive that KL state had a less deaths but high active cases.**

# Findings from the state\_wise\_covid dataset:

### 1. Maharashtra state had the highest number of confirmed COVID cases.%%.

### 2. Andaman and Nicobar Islands had the lowest number of confirmed COVID cases.

### 3. DN state had a highest recovered rate = 99.7%

### 4. NL state had a least recovered rate = 93.9 %

### 5. PB state had a highest death % = 2.7%

### 6. DN state had a least death % = 0 %

### 7. KL state had a highest active cases

### 8. There exists a +ve relation between confirmed and deaths variables except for KL state.

### 9. From plots 4 and 5 , we can derive that KL state had a less deaths but high active cases.

**WEEK-8**

**Customize Heat Maps, Pair Plots, Violin plots and Joint plots with different color palettes using the Seaborn library on any dataset.**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

# read the Housing dataset

**data1 = pd.read\_csv("C:/Users/Dell/Desktop/MRU/DV/Datasets/salesdata.csv") data1.head()**

**Order**

**ID**

CA-

**Order Date**

08-11-

**Ship Date**

11-

**Customer**

**ID**

CG-

**Country City State Postal Code**

United

**Region Category Sales Quantity Discount Profit**

**0** 2016-

152156

CA-

**1** 2016-

152156

2016

08-11-

2016

11-

2016

11-

11-

2016

12520

CG- 12520

States Henderson Kentucky 42420 South Furniture 261.9600 2.0 0.00 41.9136

United Henderson Kentucky 42420 South Furniture 731.9400 3.0 0.00 219.5820 States

CA- 16-

**1. Subplots**

# Create a FacetGrid with "Region" as columns and "Category" as hue

**g = sns.FacetGrid(data1, col="Region", hue="Category", height=4, aspect=1, palette="bright")**

# Map a scatter plot to each region with color based on "Category"

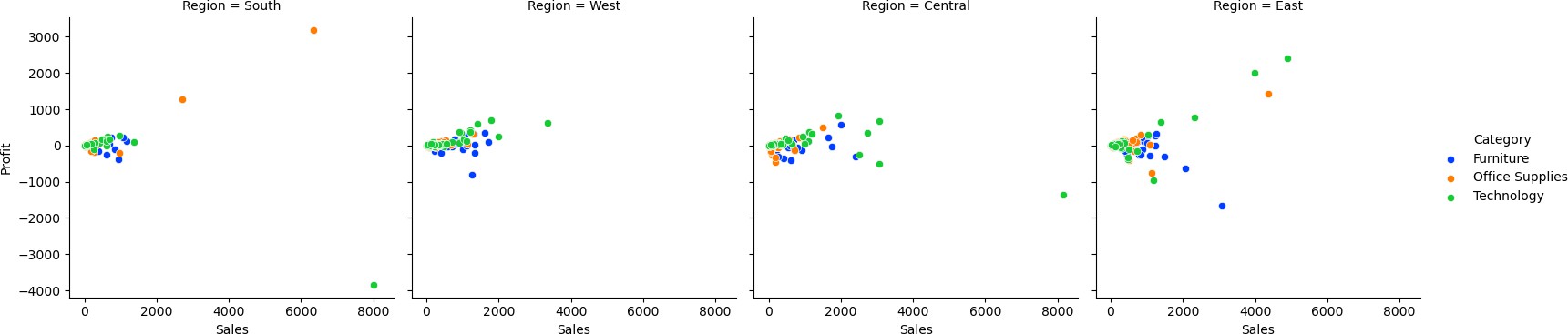
**g.map(sns.scatterplot, "Sales", "Profit")**

# Add a legend

**g.add\_legend()**

# Display the plot

**plt.show()**



# Define a custom palette with named colors

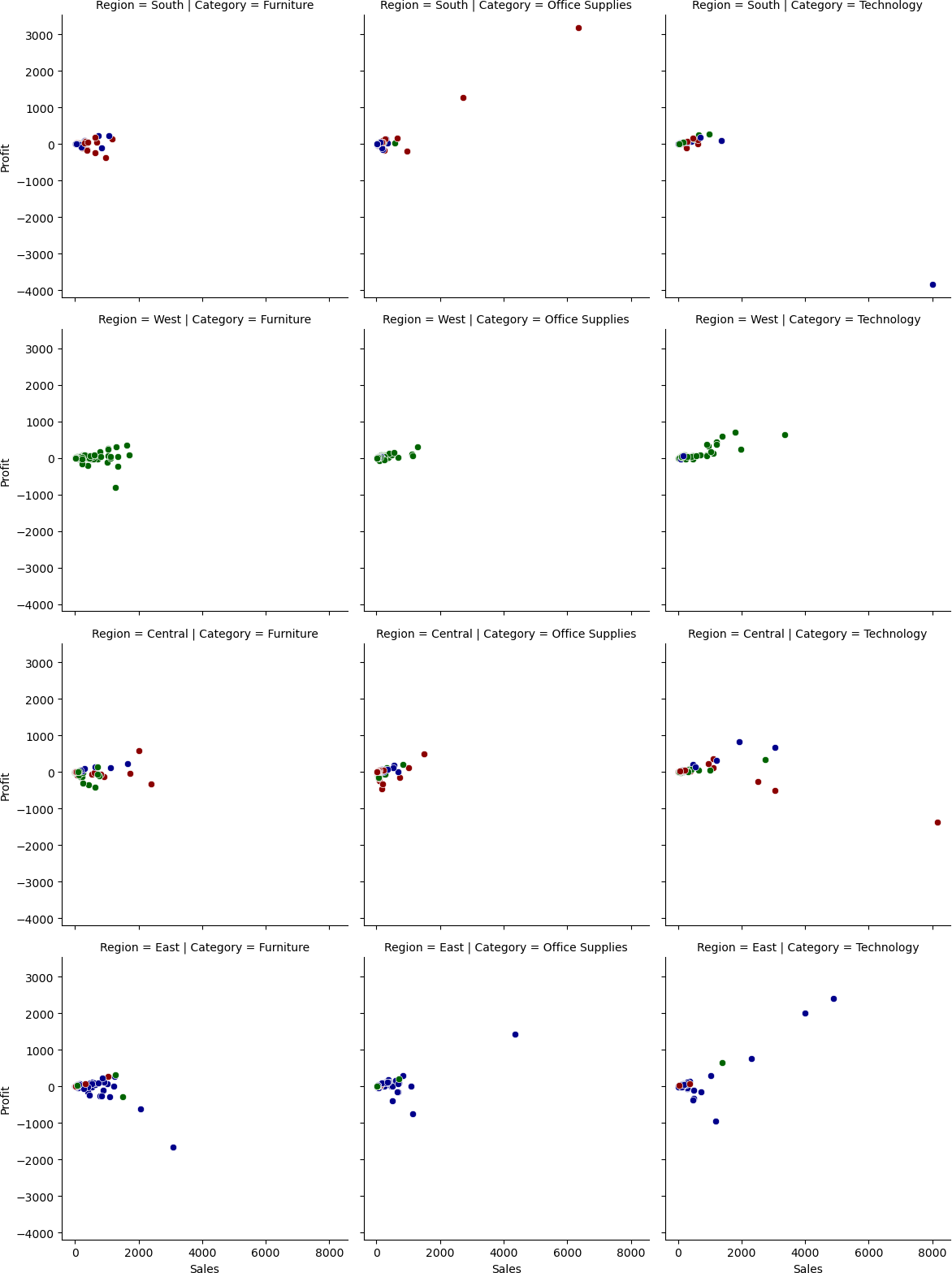
**custom\_palette = sns.color\_palette(["darkblue", "darkgreen", "darkred"])**

# Create a 2D FacetGrid with rows based on "region" and columns based on "category"

**g = sns.FacetGrid(data1, row="Region", col="Category", hue = "State", height=4, aspect=1, palette= custom\_palette)**

# Map a scatter plot with color based on "State" **g.map(sns.scatterplot, "Sales", "Profit")**

# Display the plot **plt.show()**



**2.Joint plots**

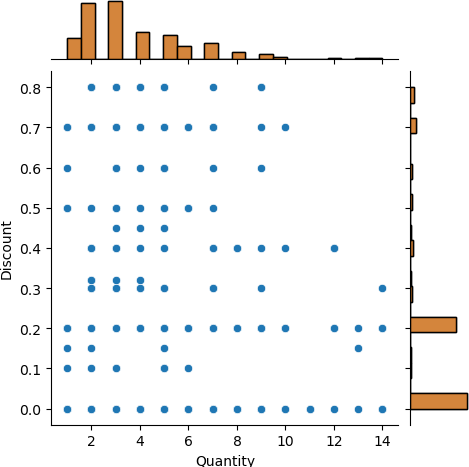
# Create a joint plot grid

**g = sns.JointGrid(data=data1, x="Quantity", y="Discount", height=5) g.plot(sns.scatterplot, sns.histplot)**

**g.plot\_marginals(sns.histplot)**

**plt.show()**





**3. Pair Plots**

# pair plot

**sns.pairplot(data1, hue="Region", palette= 'pastel')**

**plt.show()**

