**EXPERIMENT NO:- 1**

* **Python Libraries:-**

1. **Pandas:-**

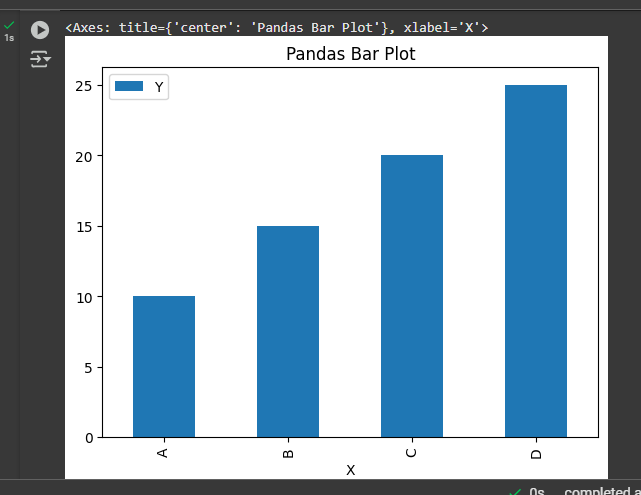
import pandas as pd

data = {'X': ['A', 'B', 'C', 'D'], 'Y': [10, 15, 20, 25]}

df = pd.DataFrame(data)

df.plot(kind='bar', x='X', y='Y', title='Pandas Bar Plot')

**OUTPUT:-**

****

1. **NumPy:-**

import numpy as np

import matplotlib.pyplot as plt

x = np.linspace(0, 10, 100) # 100 evenly spaced values from 0 to 10

y1 = np.sin(x) # Sine function

y2 = np.cos(x) # Cosine function

plt.figure(figsize=(8, 5))

plt.plot(x, y1, label='sin(x)', color='blue', linestyle='--')

plt.plot(x, y2, label='cos(x)', color='red', linestyle='-')

plt.title('Sine and Cosine Functions')

plt.xlabel('X-axis (radians)')

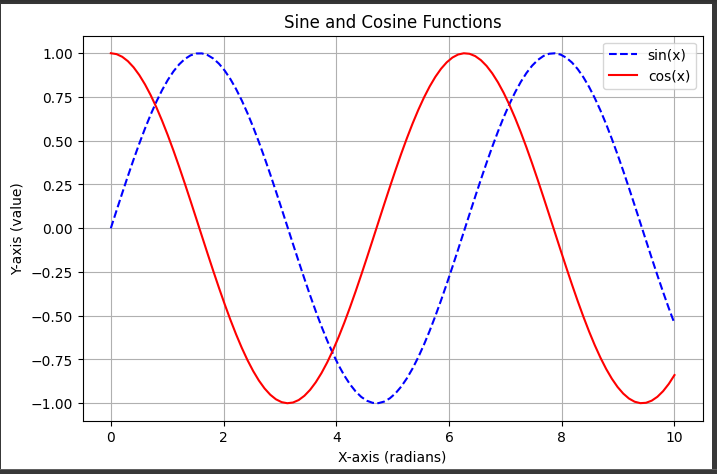
plt.ylabel('Y-axis (value)')

plt.legend()

plt.grid(True)

plt.show()

**OUTPUT:-**

****

1. **Matplotlib:-**

import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]

y = [10, 20, 25, 30, 40]

plt.plot(x, y, marker='o', label='Line Plot')

plt.title('Matplotlib Example')

plt.xlabel('X-axis')

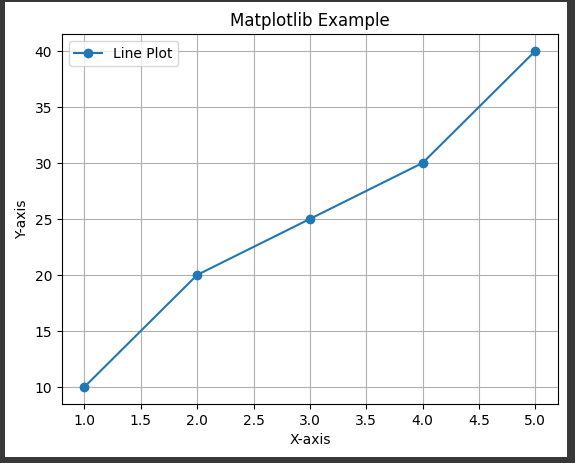
plt.ylabel('Y-axis')

plt.legend()

plt.grid(True)

plt.show()

**OUTPUT:-**

****

1. **Seaborn:-**

import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

data = {'Category': ['A', 'B', 'C', 'D'], 'Values': [4, 7, 2, 8]}

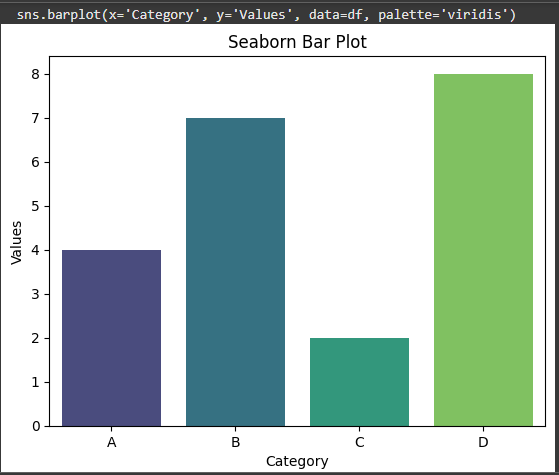
df = pd.DataFrame(data)

sns.barplot(x='Category',y='Values', data=df, palette='viridis')

plt.title('Seaborn Bar Plot')

plt.show()

**OUTPUT:-**

****

1. **BOKEH:-**

from bokeh.plotting import figure, show

from bokeh.io import output\_notebook

output\_notebook()

x = [1, 2, 3, 4, 5]

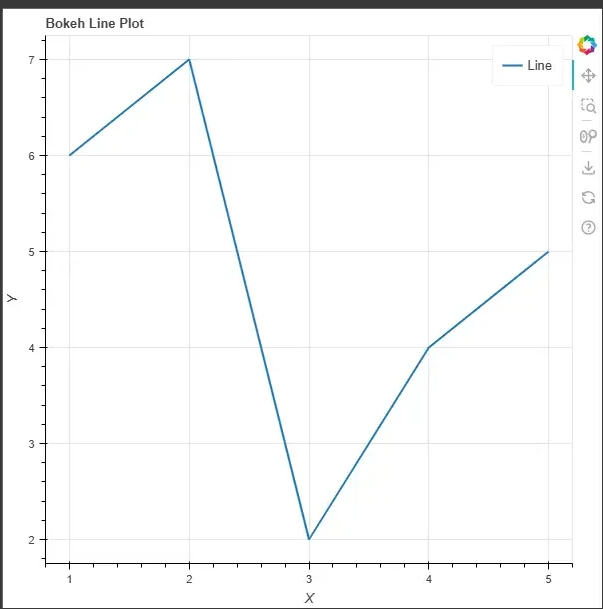
y = [6, 7, 2, 4, 5]

p = figure(title="Bokeh Line Plot",x\_axis\_label='X', y\_axis\_label='Y')

p.line(x, y, legend\_label='Line', line\_width=2)

show(p)

**OUTPUT:-**

****

* **R Libraries:-**

1. **GGPlot 2:-**

!pip install plotnine

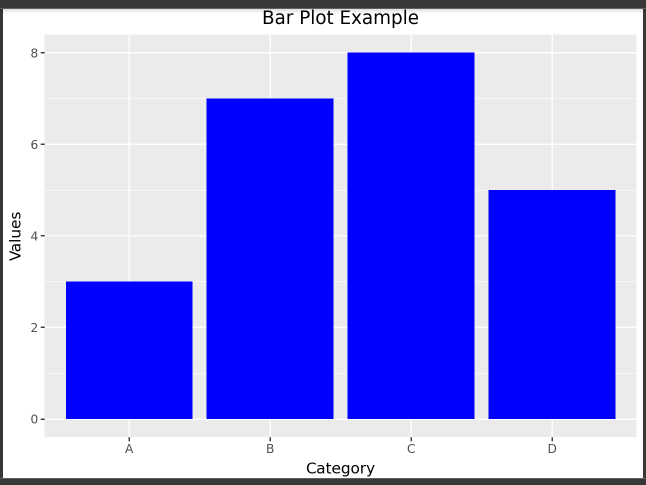
from plotnine import \*

import pandas as pd

data = pd.DataFrame({ 'x': ['A','B','C', 'D'], 'y': [3, 7, 8, 5] })

(ggplot(data, aes(x = 'x', y = 'y')) + geom\_bar(stat = "identity", fill = "blue") + ggtitle("Bar Plot Example") + xlab("Category") + ylab("Values"))

**OUPUT:-**

****

1. **DPLYR:-**

library(dplyr)

head(mtcars)

selected\_data <- mtcars %>% select(mpg, cyl, hp, wt)

cat("\nSelected Columns:\n")

print(selected\_data)

filtered\_data <- selected\_data %>% filter(mpg > 20)

cat("\nFiltered Rows (mpg > 20):\n")

print(filtered\_data)

mutated\_data <- filtered\_data %>% mutate(Power\_to\_Weight = hp / wt)

cat("\nData with New Column (Power-to-Weight Ratio):\n")

print(mutated\_data)

grouped\_summary <- mutated\_data %>%group\_by(cyl) %>%

summarize( Avg\_MPG = mean(mpg), Avg\_HP = mean(hp), Avg\_Power\_to\_Weight = mean(Power\_to\_Weight))

cat("\nGrouped Summary by Cylinders:\n")

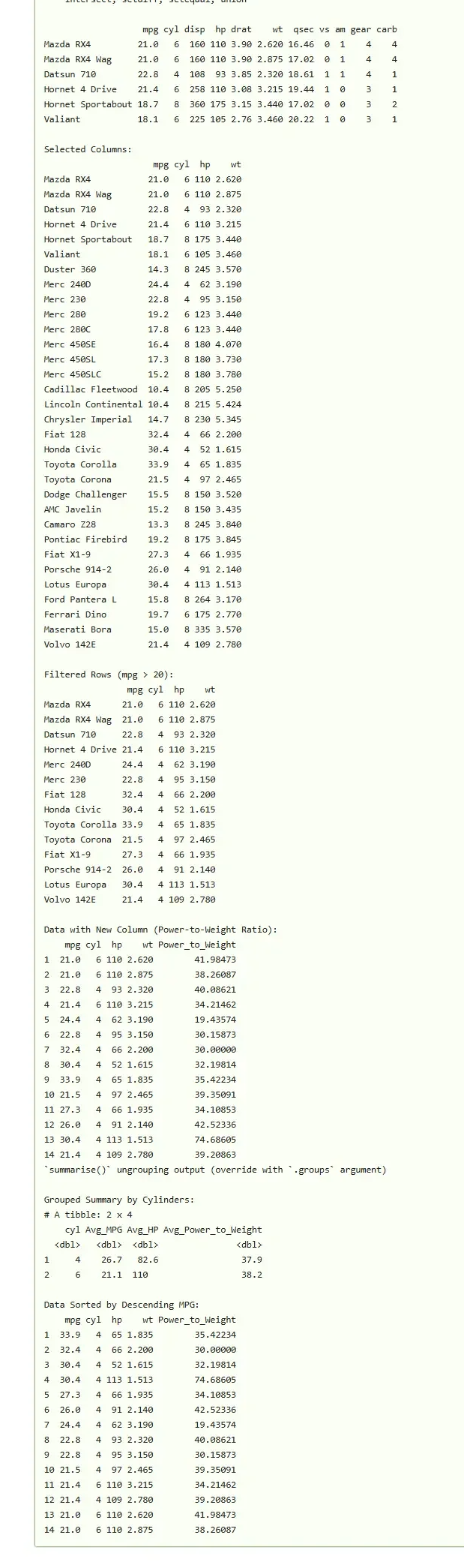
print(grouped\_summary)

sorted\_data <- mutated\_data %>%arrange(desc(mpg))

cat("\nData Sorted by Descending MPG:\n")

print(sorted\_data)

**Output:-**

****

1. **Caret:-**

library(caret)

pred\_values <- factor(c(TRUE,FALSE,FALSE,TRUE,FALSE,TRUE,FALSE))

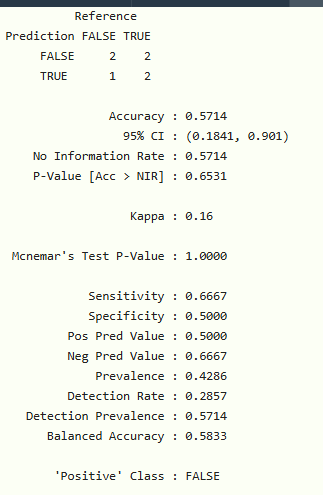
actual\_values<- factor(c(FALSE,FALSE,TRUE,TRUE,FALSE,TRUE,TRUE))

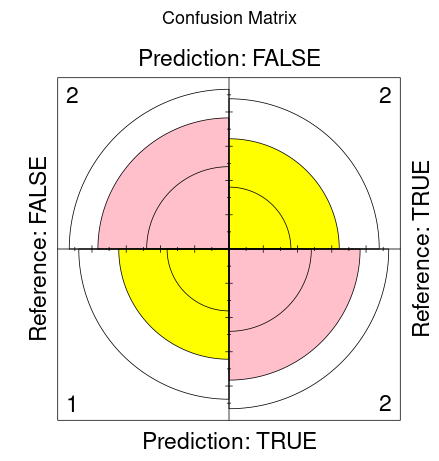
cf <- caret::confusionMatrix(data=pred\_values, reference=actual\_values)

print(cf)

fourfoldplot(as.table(cf),color=c("yellow","pink"),main = "Confusion Matrix")

**OUTPUT:-**

****

****

1. **TIDYR:-**

library(tidyr)

n = 10

tidy\_dataframe = data.frame(

S.No = c(1:n),

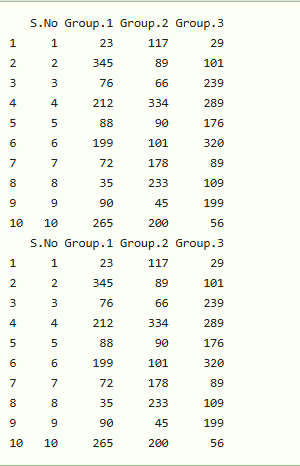
Group.1 = c(23, 345, 76, 212, 88, 199, 72, 35, 90, 265),

Group.2 = c(117, 89, 66, 334, 90, 101, 178, 233, 45, 200),

Group.3 = c(29, 101, 239, 289, 176, 320, 89, 109, 199, 56))

tidy\_dataframe

**OUTPUT:-**

****

1. **SHINY:-**

library(shiny)

ui <- fluidPage(

sliderInput(inputId = "num", label = "Choose a number", value = 10, min = 1, max = 1000),

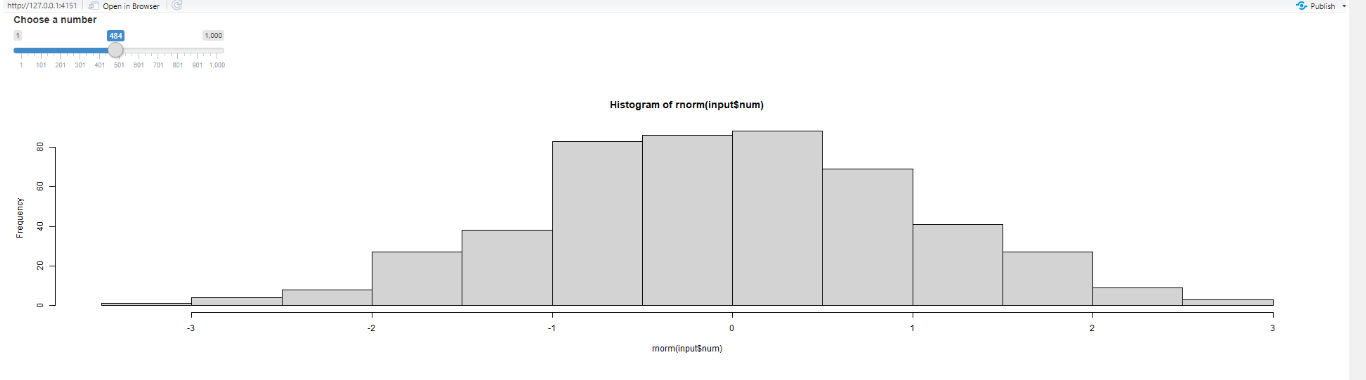
plotOutput("hist"))

server <- function(input, output){

output$hist <- renderPlot({hist(rnorm(input$num))})}

shinyApp(ui = ui, server = server)

**OUTPUT:-**

****