**DAV EXPERIMENT 1**

Plotly

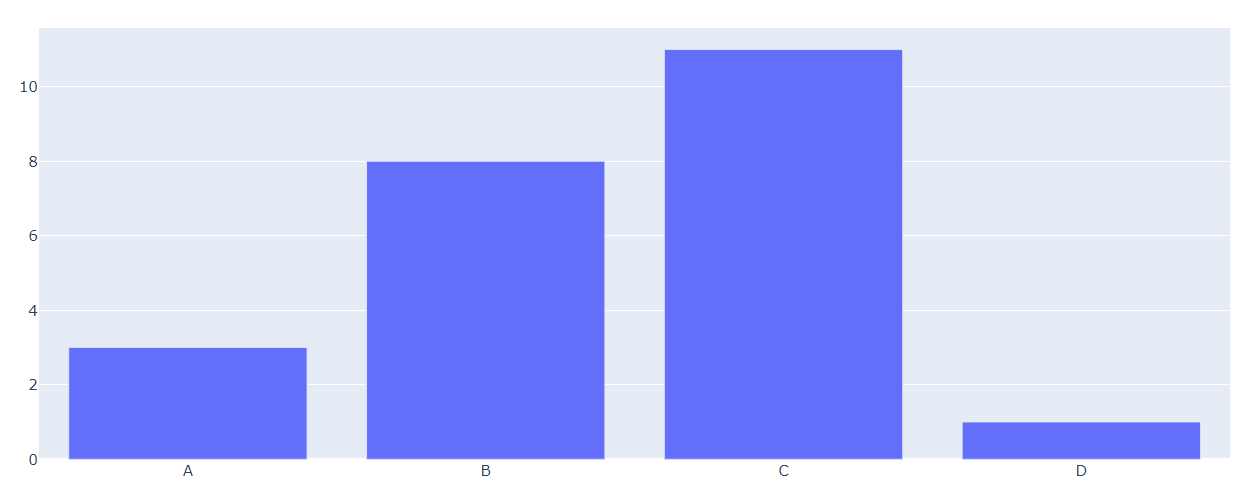
import plotly.graph\_objects as go

x = ['A', 'B', 'C', 'D']

y = [3, 8, 11, 1]

fig = go.Figure(data=go.Bar(x=x, y=y))

fig.show()



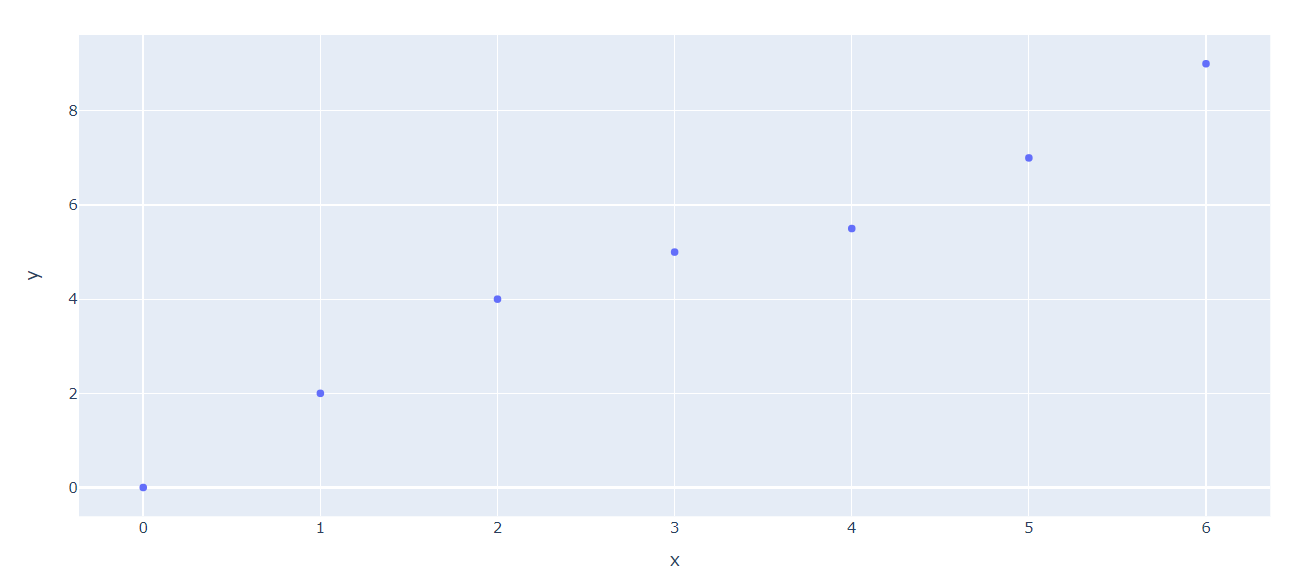
import plotly.express as px

x = [0, 1, 2, 3, 4, 5, 6]

y = [0, 2, 4, 5, 5.5, 7, 9]

fig = px.scatter(x=x, y=y)

fig.show()



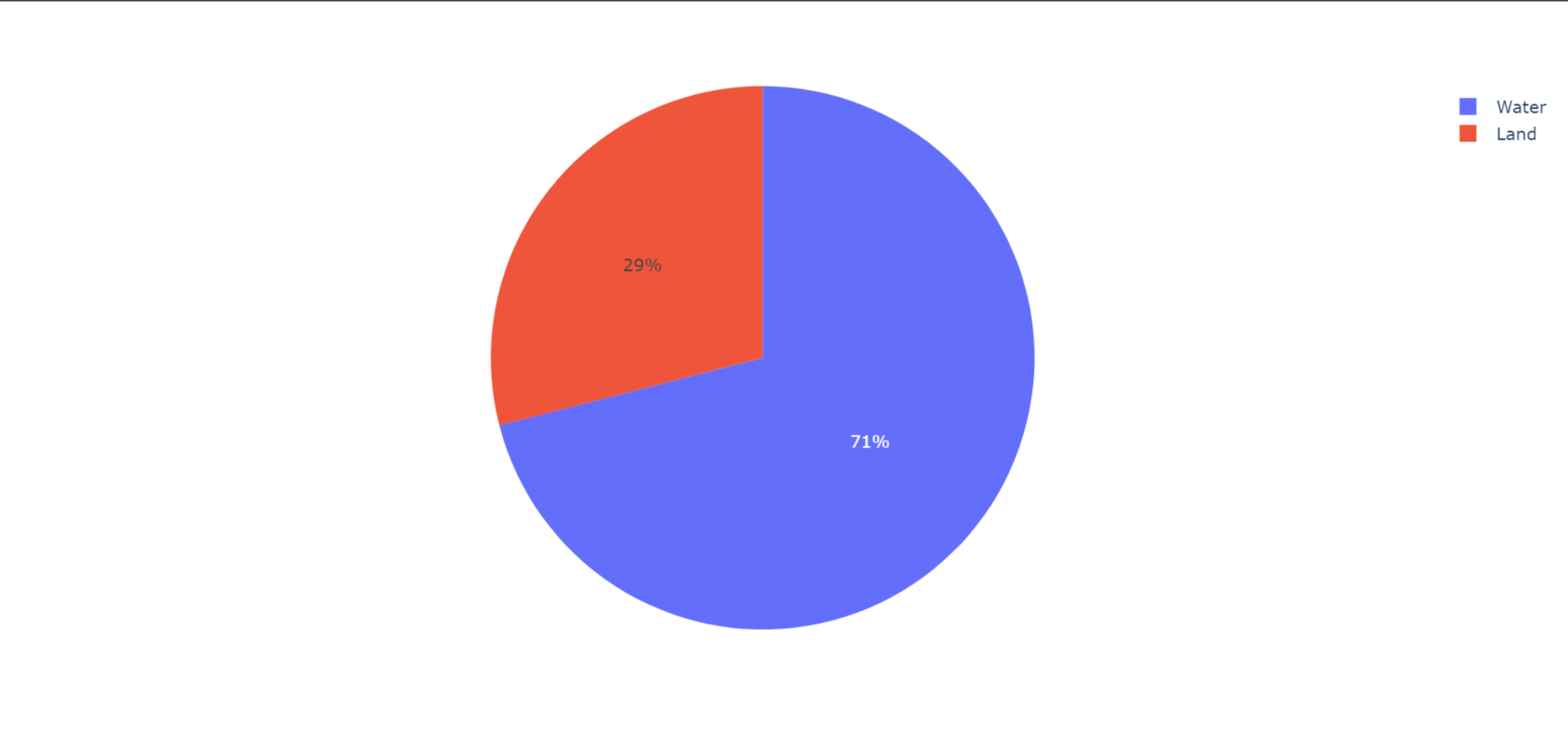
import plotly.express as px

labels = ['Water', 'Land']

values = [71, 29]

fig = px.pie(values=values, names=labels)

fig.show()



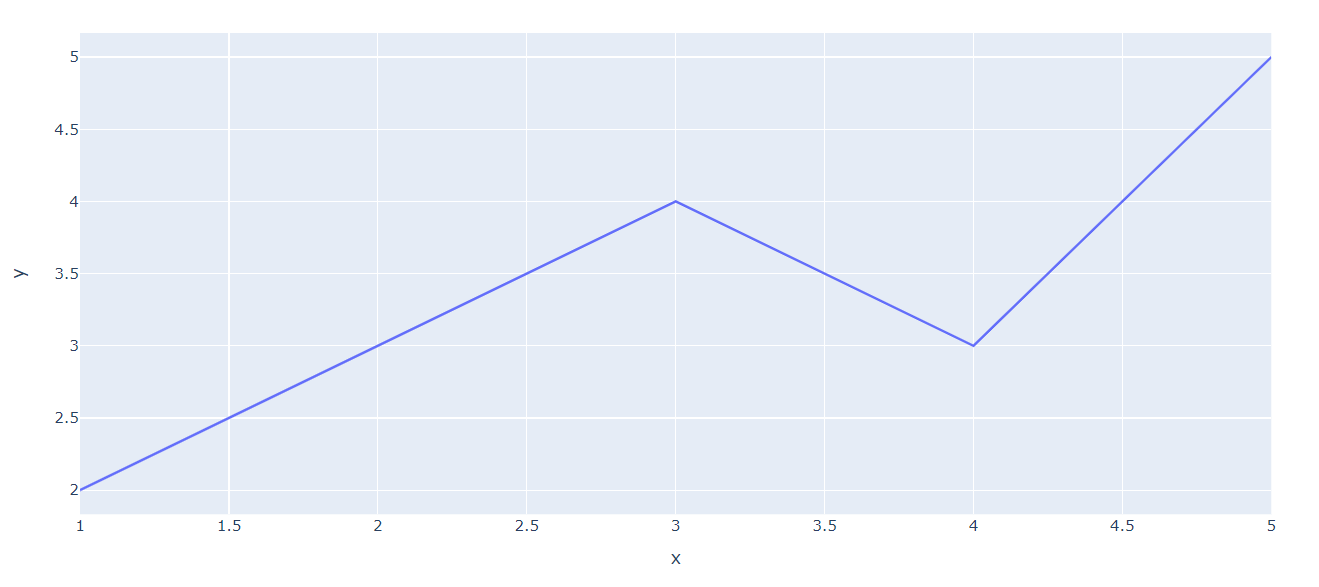
import plotly.express as px

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

y = [2, 3, 4, 3, 5]

fig = px.line(x=x, y=y)

fig.show()



NumPy

import numpy as np

# Create a 1D array

arr1 = np.array([1, 2, 3])

print("1D Array:")

print(arr1)

arr2=np.array([[1, 2, 3], [4, 5, 6]])

print("\n2D Array:")

print(arr2)

#Create third array

arr3=arr2.T

print("\nArray Operations:")

print("Shape of 1D array:", arr1.shape)

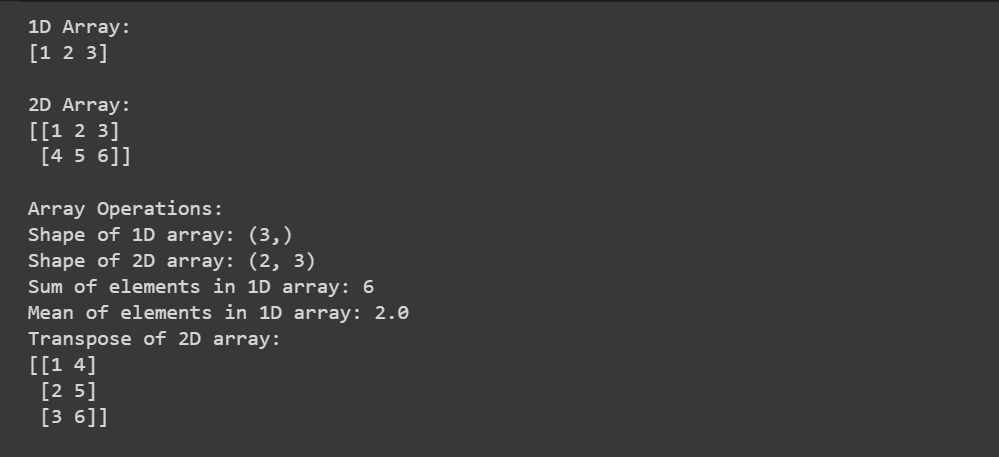
print("Shape of 2D array:", arr2.shape)

print("Sum of elements in 1D array:", np.sum(arr1))

print("Mean of elements in 1D array:", np.mean(arr1))

print("Transpose of 2D array:")

print(arr3)



Matplot

import matplotlib.pyplot as plt

# Sample data

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

y = [2, 3, 4, 3, 5]

# Create a line plot

plt.plot(x, y)

# Add labels and title

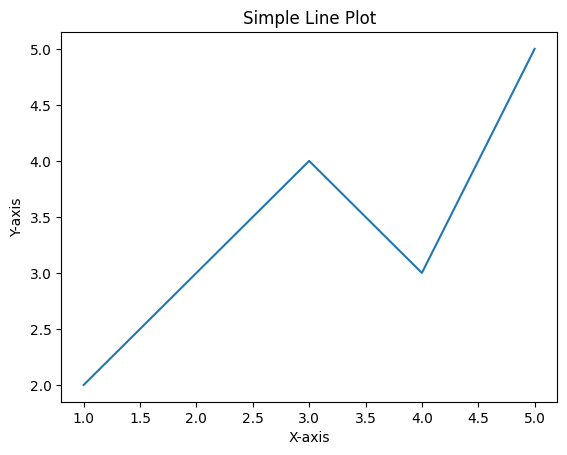
plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Simple Line Plot')

# Display the plot

plt.show()



import matplotlib.pyplot as plt

a = ['A', 'B', 'C', 'D']

b = [3, 8, 11, 1]

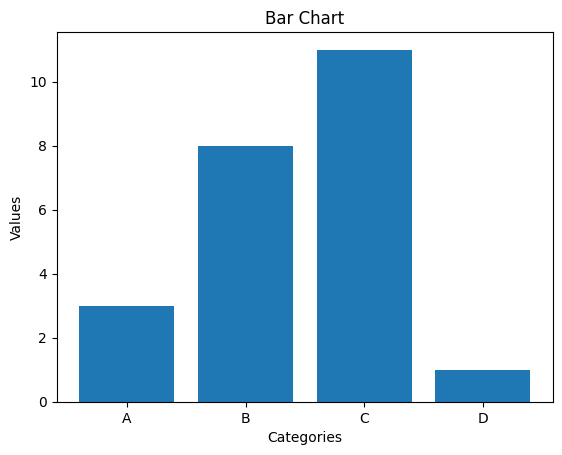
plt.bar(a, b)

plt.xlabel('Categories')

plt.ylabel('Values')

plt.title('Bar Chart')

plt.show()



Seaborn

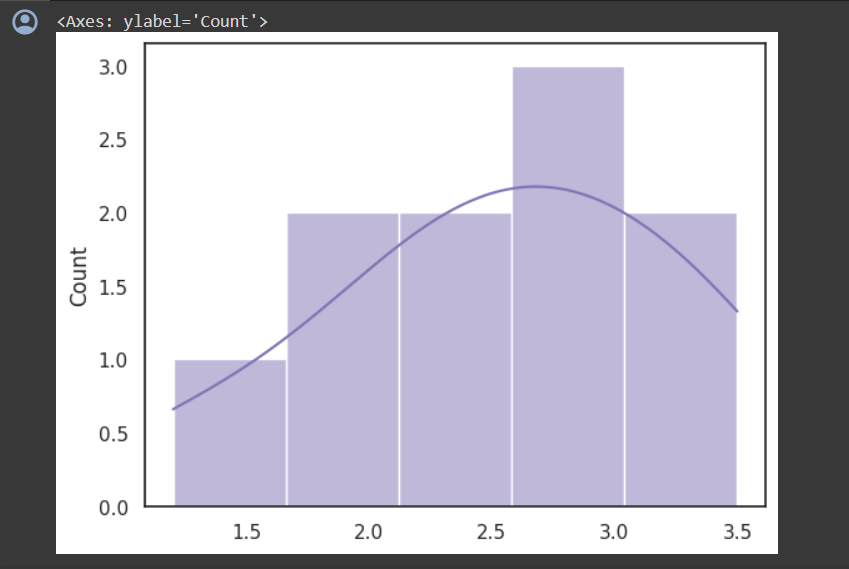
import numpy as np

import seaborn as sb

sb.set(style="white")

d = np.array([1.2, 2.3, 3.5, 2.7, 3.0, 1.8, 2.5, 2.9, 3.4, 2.1])

sb.histplot(d, kde=True, color="m")



dplyr

install.packages("dplyr")

library(dplyr)

# Create data frame

data <- data.frame(

id = 1:5,

name = c("Alice", "Bob", "Charlie", "David", "Eve"),

age = c(25, 30, 22, 35, 28),

score = c(88, 75, 92, 67, 80)

)

# Selecting specific columns

selected\_data <- select(data, id, name)

# Filtering rows based on a condition

filtered\_data <- filter(data, age > 25)

# Adding a new column

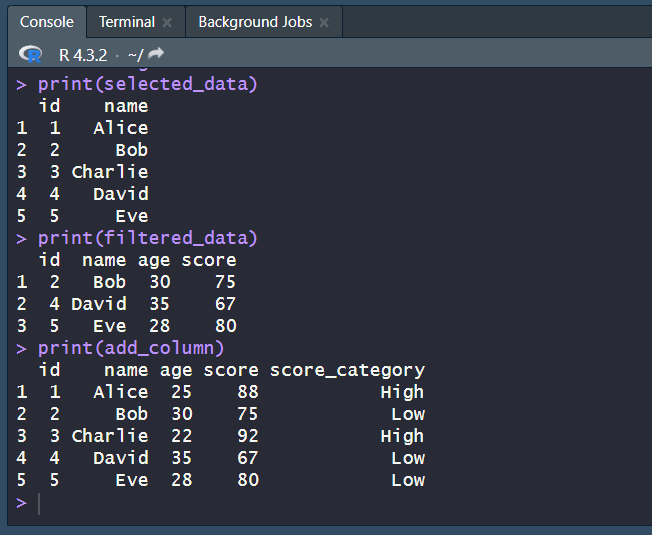
add\_column <- mutate(data, score\_category = ifelse(score > 80, "High", "Low"))

# Printing the results

print(selected\_data)

print(filtered\_data)

print(add\_column)



ggplot2

install.packages("ggplot2")

library(ggplot2)

# Create a sample data frame

sam\_d <- data.frame(

x = c(1, 2, 3, 4, 5),

y = c(2, 3, 5, 7, 11)

)

# Create a scatter plot

ggplot(sam\_d, aes(x = x, y = y)) +

geom\_point() +

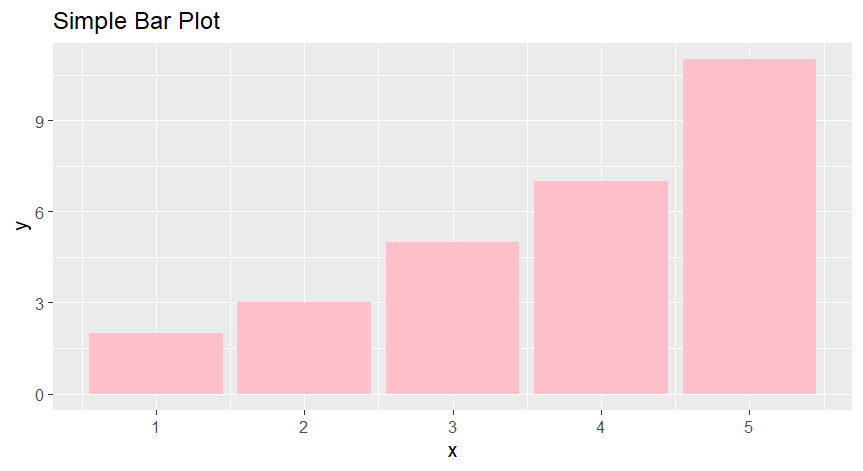
labs(title = "Simple Scatter Plot", x = "X-axis Label", y = "Y-axis Label")

# Create a bar plot

ggplot(sam\_d, aes(x = x, y = y)) +

geom\_bar(stat = "identity", fill = "pink") +

labs(title = "Simple Bar Plot", x = "x", y = "y")



tidyr

# First, install and load the tidyr package

install.packages("tidyr")

library(tidyr)

# Create data frame

d\_f <- data.frame(

id = 1:5,

name = c("Alice", "Bob", "Charlie", "David", "Eve"),

age = c(25, 30, 22, 35, 28),

score = c(88, 75, 92, 67, 80)

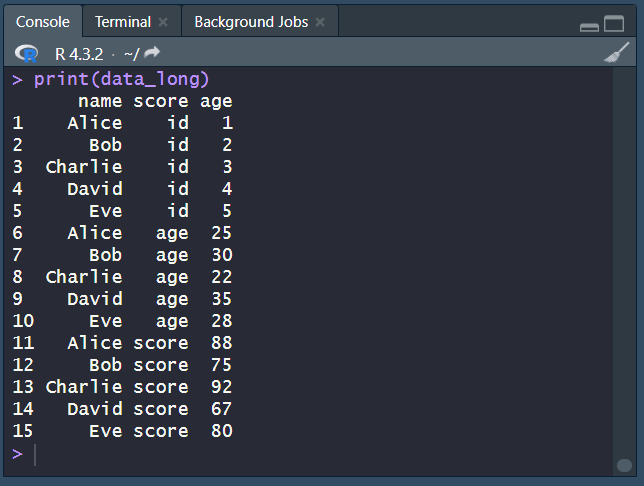
)

# Reshape the data from wide to long format

data\_long <- gather(d\_f, score, age, -name)

# Print the long format data

print(data\_long)



readr

# First, install and load the readr package

install.packages("readr")

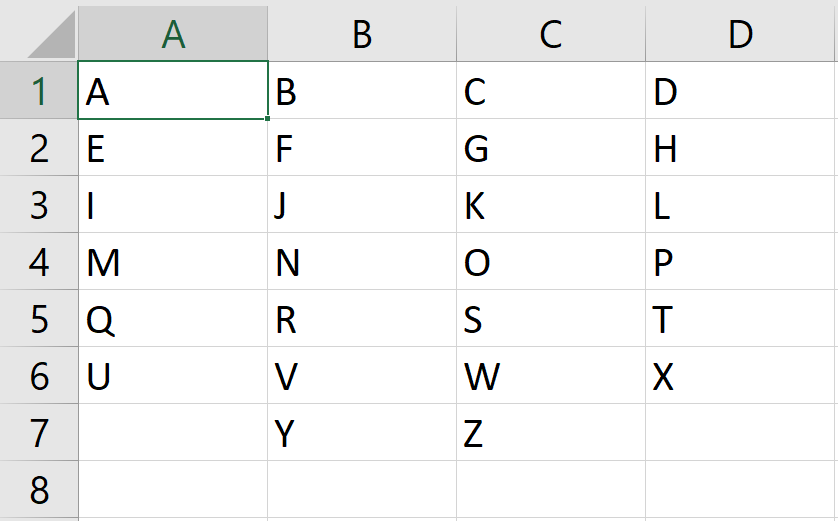
library(readr)

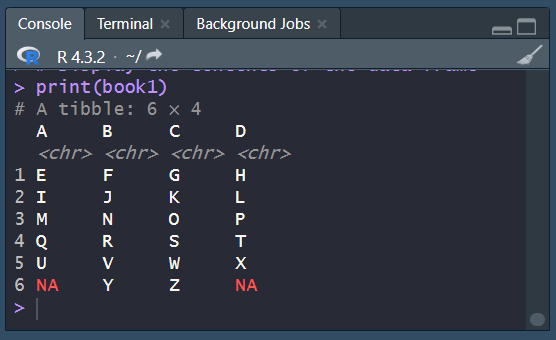
# Read a CSV file into a data frame

book1 <- read\_csv("C:\\Users\\PARTHIVI\\OneDrive\\Desktop\\Book1.csv")

# Display the contents of the data frame

print(book1)





quanteda

library(quanteda)

# Create a corpus of text documents

corpus <- corpus(c(

"This is the first document. It contains some sample text.",

"The second document is also included in the corpus.",

"Finally, the third document completes the corpus."

))

# Print the corpus

print(corpus)

# Tokenize the text

tokens <- tokens(corpus)

# Create a document-feature matrix

dfm <- dfm(tokens)

# Print the document-feature matrix

print(dfm)

