Code:

->Hive Queries:

1) Create Table:

CREATE TABLE crypto_data (open FLOAT, high FLOAT, low FLOAT, close FLOAT, volume FLOAT, marketCap FLOAT, time_stamp STRING, crypto_name STRING, cdate STRING, percentage_change FLOAT, normalized_marketCap FLOAT, normalized_volume FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;

2) Load data :

LOAD DATA INPATH '/bigdata/preprocessed_Data.csv' INTO TABLE crypto_data;

3) Calculates average open, high, low, close prices, and volume for each cryptocurrency:

SELECT crypto_name, AVG(open) AS avg_open, AVG(high) AS avg_high, AVG(low) AS avg_low, AVG(close) AS avg_close, AVG(volume) AS avg_volume FROM crypto_data GROUP BY crypto_name;

4) Finds the maximum high and minimum low prices for each cryptocurrency:

SELECT crypto_name, MAX(high) AS max_high, MIN(low) AS min_low FROM crypto_data GROUP BY crypto_name;

5) Computes the daily average percentage price change for each cryptocurrency:

SELECT crypto_name, cdate, AVG(percentage_change) AS avg_percentage_change FROM crypto_data GROUP BY crypto_name, cdate ORDER BY crypto_name, cdate limit 10;

6) Average Market Cap & Volume for Top 15 Cryptos:

SELECT crypto_name, AVG(normalized_marketCap) AS avg_normalized_marketCap, AVG(normalized_volume) AS avg_normalized_volume FROM crypto_data GROUP BY crypto_name limit 15;

7) Top 5 Market Caps per Crypto :

SELECT crypto_name, cdate, marketCap FROM crypto_data ORDER BY crypto_name, marketCap DESC LIMIT 5;

8) Calculate Correlation Between Closing Price and Market Cap for Each Cryptocurrency :

SELECT crypto_name, CORR(close, marketCap) AS correlation_close_marketCap FROM crypto_data GROUP BY crypto_name;

9) Correlation between Volume and Market Cap :

SELECT crypto_name, CORR(volume, marketCap) AS correlation_volume_marketCap FROM crypto_data GROUP BY crypto_name;

10) Top 10 Cryptocurrencies by Highest Closing Price :

SELECT crypto_name, MAX(close) AS max_close FROM crypto_data GROUP BY crypto_name ORDER BY max_close DESC LIMIT 10;

11) Top 10 Cryptocurrencies by Lowest Closing Price :

SELECT crypto_name, MIN(close) AS min_close FROM crypto_data GROUP BY crypto_name ORDER BY min_close DESC LIMIT 10;

12) Top 10 Cryptocurrencies by Highest Closing Price :

SELECT crypto_name, MAX(close) AS max_close FROM crypto_data GROUP BY crypto_name ORDER BY max_close DESC LIMIT 10;

13) Top 10 Cryptocurrencies by Market Dominance :

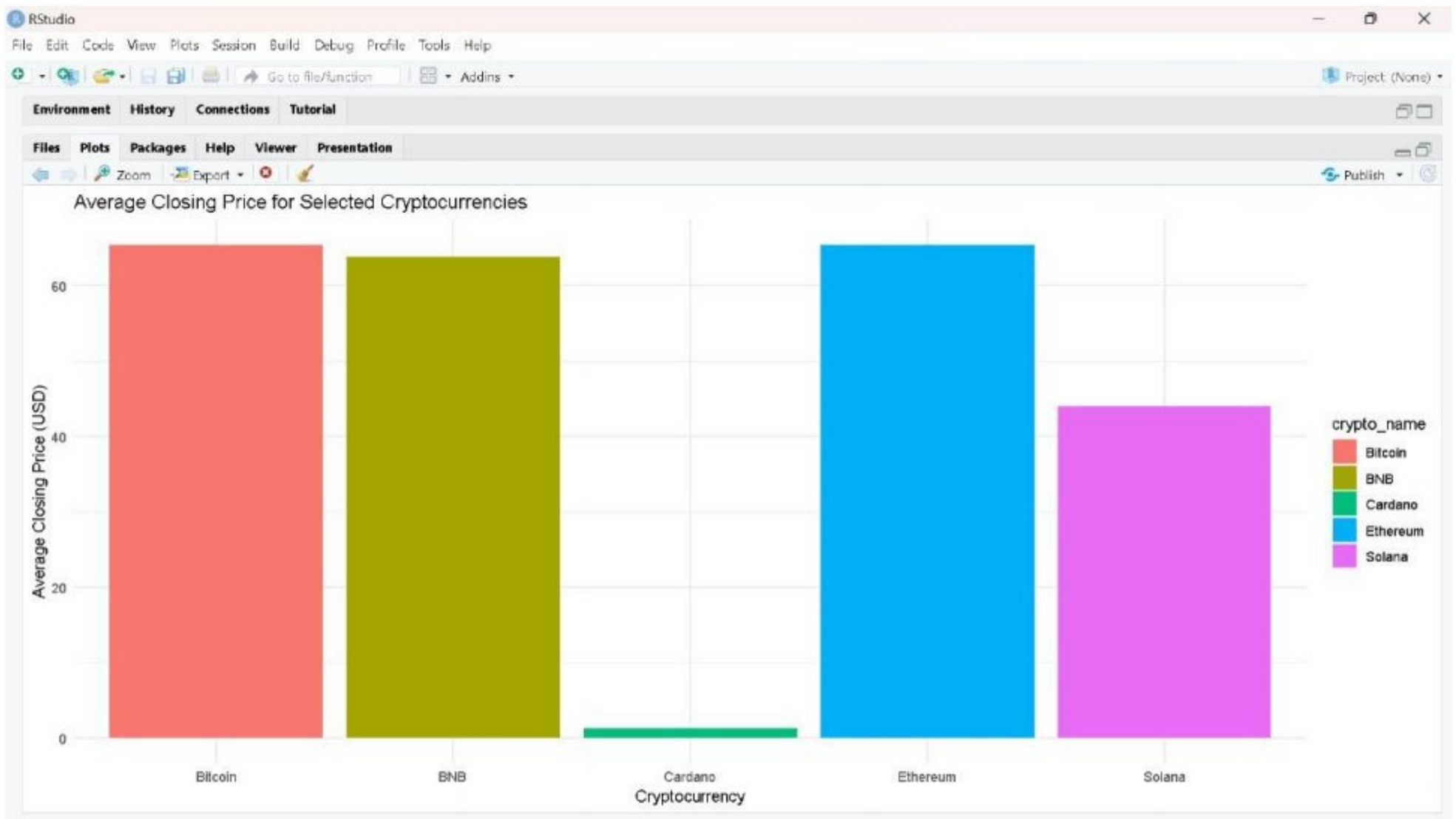
WITH total_market_cap AS (SELECT cdate, SUM(marketCap) AS total_marketCap FROM crypto_data GROUP BY cdate) SELECT a.cdate, a.crypto_name, (a.marketCap / b.total_marketCap) * 100 AS market_dominance FROM crypto_data a JOIN total_market_cap b ON a.cdate = b.cdate ORDER BY a.cdate, market_dominance DESC;

->R Programming:

```
setwd("C:/Users/geeth/OneDrive/Documents")

data <- read.csv('BigData_Project/Data.csv')

getwd()

library(lubridate)

install.packages("ggplot2")

library(ggplot2)

install.packages("dplyr")

library(dplyr)

data$date <- as.Date(data$date, format = "%d-%m-%Y")

library(tidyverse)

install.packages("ggcorrplot")

library(ggcorrplot)

#1)

crypto_data_filtered <- data %>%

 filter(crypto_name %in% c("Bitcoin", "Ethereum", "Cardano", "BNB", "Solana")) %>%

 select(date, crypto_name, close)


# Reshape data to wide format where each cryptocurrency is a column

crypto_wide <- crypto_data_filtered %>%

 pivot_wider(names_from = crypto_name, values_from = close)


# Calculate correlation matrix, using complete observations only

cor_matrix <- cor(crypto_wide[, -1], use = "complete.obs")


# Plot the correlation matrix

ggcorrplot::ggcorrplot(cor_matrix, lab = TRUE, title = "Price Correlations Among
Cryptocurrencies")
```

#2)

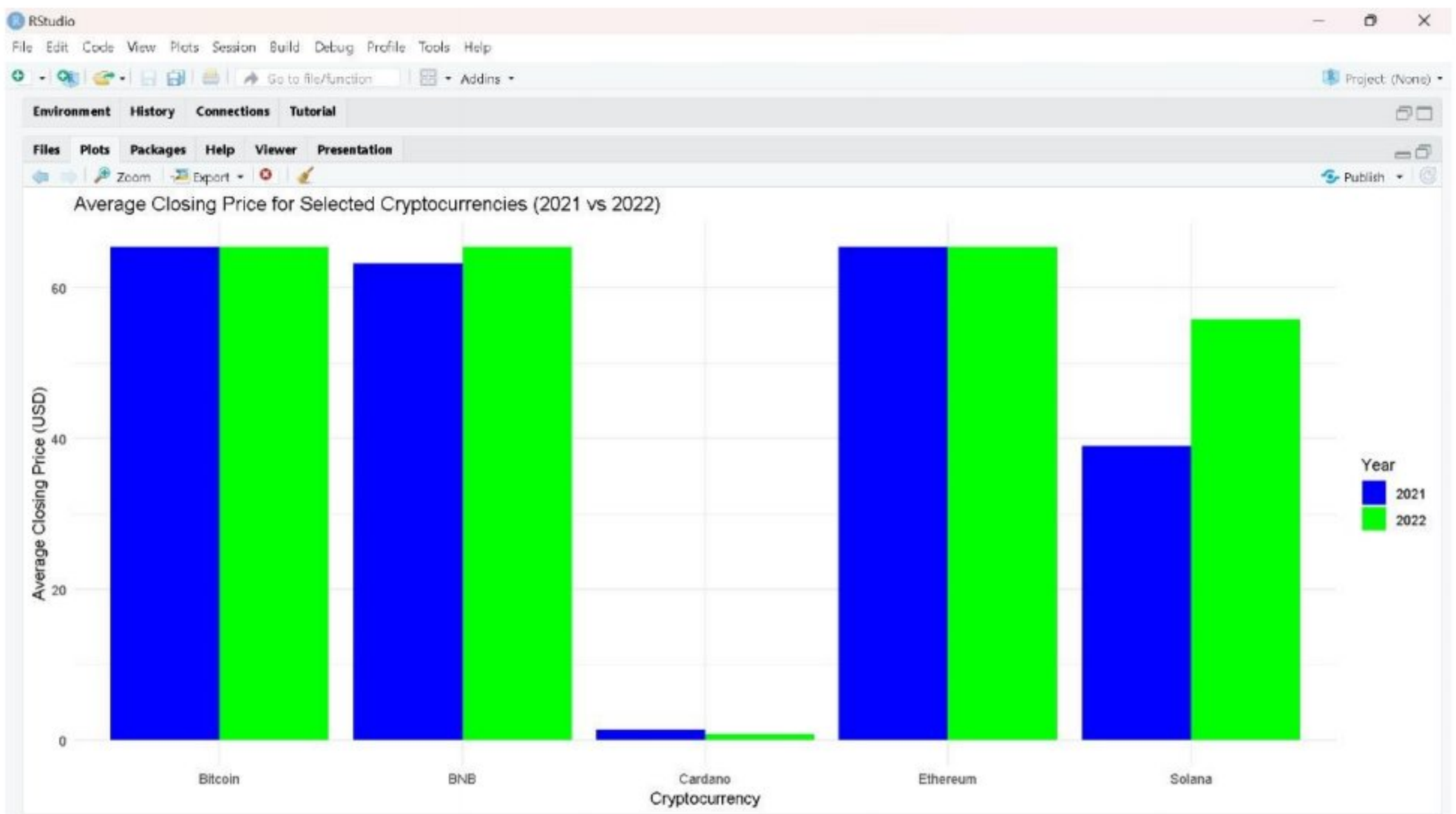
```
# Calculate summary statistics (average closing price for each crypto)
summary_stats <- crypto_data_filtered %>%
 group_by(crypto_name) %>%
 summarise(
  avg_close = mean(close, na.rm = TRUE)
 )

# Step 1: Plot Summary Statistics - Average Closing Prices
ggplot(summary_stats, aes(x = crypto_name, y = avg_close, fill = crypto_name)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Average Closing Price for Selected Cryptocurrencies",
    x = "Cryptocurrency", y = "Average Closing Price (USD)") +
 theme_minimal()
```



#3)

```
# Sample Data (assuming 'data' is your original dataset)
crypto_data <- data %>%
  filter(crypto_name %in% c("Bitcoin", "Ethereum", "Cardano", "BNB", "Solana")) %>%
  select(date, crypto_name, close) %>%
  mutate(year = year(date))

# Calculate summary statistics (average closing price for each crypto by year)
summary_stats_yearly <- crypto_data %>%
  group_by(crypto_name, year) %>%
  summarise(
    avg_close = mean(close, na.rm = TRUE)
  )
```

```
#  Plot Average Closing Prices for 2021 and 2022
ggplot(summary_stats_yearly, aes(x = crypto_name, y = avg_close, fill = factor(year))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Closing Price for Selected Cryptocurrencies (2021 vs 2022)",
       x = "Cryptocurrency", y = "Average Closing Price (USD)",
       fill = "Year") +
  scale_fill_manual(values = c("2021" = "blue", "2022" = "green")) +
  theme_minimal()
```



#4)

```
# Calculate the percentage change from 2021 to 2022
crypto_data_comparison <- summary_stats_yearly %>%
  spread(key = year, value = avg_close) %>%  # Spread the data into 2021 and 2022 columns
  mutate(
    percentage_change = ((`2022` - `2021`) / `2021`) * 100  # Calculate percentage change
  ) %>%
  select(crypto_name, percentage_change)  # Select relevant columns
```

```
# Plot Percentage Change from 2021 to 2022

ggplot(crypto_data_comparison, aes(x = reorder(crypto_name, percentage_change), y =
percentage_change, fill = crypto_name)) +

  geom_bar(stat = "identity", position = "dodge") +

  geom_text(aes(label = round(percentage_change, 2)), vjust = -0.5) +

  labs(title = "Percentage Change in Average Closing Price (2022 vs 2021)",

       x = "Cryptocurrency", y = "Percentage Change (%)") +

theme_minimal() +

scale_fill_brewer(palette = "Set2")
```