VSN Sai Krishna Mohan Kocherlakota

Title: Module 5 R Practice

Submission Date: 12th December 2023

INTRODUCTION:

The provided code conducts an extensive analysis of Bitcoin's historical data. It begins by loading necessary libraries, reading a CSV file, and cleaning the dataset. Statistical summaries are calculated for columns 'open,' 'close,' 'high,' and 'low,' displaying minimum, maximum, mean, median, and standard deviation values. Graphs illustrate these statistics categorized by type. Additionally, it performs correlation analysis between different attributes and presents a correlation matrix using a heatmap. Finally, it conducts a linear regression model to examine the relationship between 'open' and 'close' Bitcoin prices. The code is organized to facilitate data understanding and exploration through summary statistics, visualizations, and modelling.

TASKS:

1.Data loading and cleaning

```
> #Reading Csv file
> bit_df <-read_csv("BitCoin.csv")
Rows: 1609 Columns: 6
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (1): Date
dbl (5): Id, Open, High, Low, Close

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
> #Cleaning the data set
> bit_df <- clean_names(bit_df)
> #Basic operations to check data set
> str(bit_df)
spc_tbl_ [1,609 × 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ id   : num [1:1609] 1 2 3 4 5 6 7 8 9 10 ...
 $ date : chr [1:1609] "22-09-2017" "21-09-2017" "20-09-2017" "19-09-2017" ...
 $ open : num [1:1609] 3628 3901 3916 4074 3591 ...
 $ high : num [1:1609] 3758 3916 4031 4094 4079 ...
 $ low  : num [1:1609] 3554 3614 3858 3869 3591 ...
 $ close: num [1:1609] 3631 3631 3906 3925 4065 ...
 - attr(*, "spec")=
  .. cols(
  ..   Id = col_double(),
  ..   Date = col_character(),
  ..   Open = col_double(),
  ..   High = col_double(),
  ..   Low = col_double(),
  ..   Close = col_double()
  .. )
 - attr(*, "problems")=<externalptr>
> head(bit_df)
```

The data has been successfully read and cleaned. The dataset contains six columns: 'id', 'date', 'open', 'high', 'low', and 'close'. The 'date' column is in character format, while the others are in numeric format. The 'id' column appears to represent some form of unique identifier, and the 'date' column contains date information in the format 'dd-mm-yyyy'. The 'open', 'high', 'low', and 'close' columns seem to represent financial data related to Bitcoin.
The structure of the dataset has been checked using `str(bit_df)`, confirming the column types and providing an overview of the first few rows using `head(bit_df)`. This dataset appears ready for analysis, with numeric values appropriately formatted for financial calculations and character date values that could be converted to a date format for further temporal analysis if needed.

```
> #Converting data types in data set
> bit_df$date <- dmy(bit_df$date)
> bit_df$year <- year(bit_df$date)
> bit_df$month <- month(bit_df$date)
```

Used the dmy() function from the lubridate package to convert the 'date' column from character to a date format. Then,

you're extracting the year and month information from the 'date' column using the year() and month() functions, respectively, also from the lubridate package.

2. Computing Statistical Equations

```
> #Statistical Analysis
> #Open column
> open_summary <- bit_df %>%
+   summarize(
+     min_open = min(open),
+     max_open = max(open),
+     mean_open = mean(open),
+     median_open = median(open),
+     sd_open = sd(open)
+   )
> #Open column
Table: Summary Statistics of 'open' column

| min_open| max_open| mean_open| median_open|  sd_open|
|--------:|--------:|---------:|-----------:|--------:|
|     68.5|  4901.42|  693.4974|      446.89| 797.3651|
> # Combining summary statistics into a single table
> combined_summary <- bind_rows(
+   high_summary %>% mutate(category = "High"),
+   low_summary %>% mutate(category = "Low"),
+   open_summary %>% mutate(category = "Open"),
+   close_summary %>% mutate(category = "Close")
+ )
>
> combined_summary
# A tibble: 4 × 6
  min_open max_open mean_open median_open sd_open category
     <dbl>    <dbl>     <dbl>       <dbl>   <dbl> <chr>
1     74.6    4975.      713.        452.    826. High
2     65.5    4679.      674.        440.    768. Low
3     68.5    4901.      693.        447.    797. Open
4     68.4    4892.      696.        448.    801. Close
```

The above R code is summary statistics for the 'open', 'close', 'high', and 'low' columns and then combined these statistics into a single summary table named `combined_summary`. Each summary includes minimum, maximum, mean, median, and standard deviation values for their respective columns.

The table generated using kable() beautifully presents these statistics in a tabular format, making it easier to comprehend and analyze the distribution and central tendencies within the 'open', 'close', 'high', and 'low' columns of dataset.This combined summary table provides a comparative overview of these statistical measures across different categories ('High', 'Low', 'Open', and 'Close').

3.Co-relation Analysis & Regrression

```
> # Selecting a subset of variables for correlation analysis
> correlation_subset <- bit_df %>%
+   select(open, close, high, low, month)
>
> # Calculating correlation matrix
> correlation_matrix <- cor(correlation_subset)
>
> # Displaying correlation matrix
> print(correlation_matrix)
             open       close       high         low       month
open  1.00000000 0.997858014 0.99900885 0.998605026 0.009373790
close 0.99785801 1.000000000 0.99896084 0.998988299 0.009349972
high  0.99900885 0.998960835 1.00000000 0.998216700 0.010258760
low   0.99860503 0.998988299 0.99821670 1.000000000 0.007739221
month 0.00937379 0.009349972 0.01025876 0.007739221 1.000000000
```

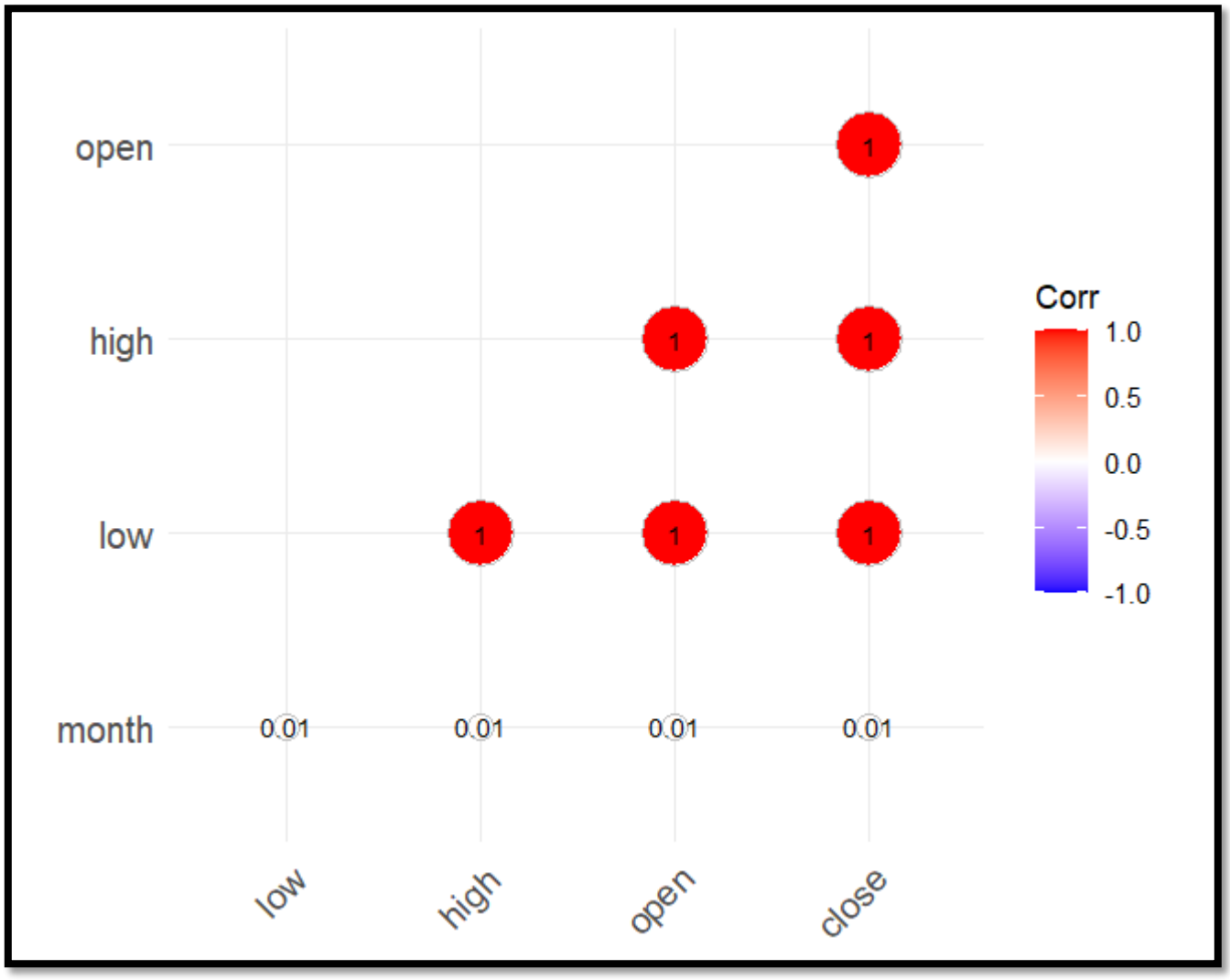

Figure 1: - Co-relation Chart

```
> # Plotting correlation chart
> ggcorrplot(correlation_matrix, hc.order = TRUE, type = "lower",
+            lab = TRUE, lab_size = 3, method = "circle")
```

This snippet calculates the correlation matrix for selected variables ('open,' 'close,' 'high,' 'low,' and 'month') in the Bitcoin dataset. The correlation values range between -1 and 1, indicating the strength and direction of relationships between these variables. The subsequent code generates a correlation chart using ggcorrplot, displaying the matrix with labels and a circular representation.

The correlation matrix showcases the high correlation among the 'open,' 'close,' 'high,' and 'low' variables (nearly 1), signifying strong linear relationships. However, the 'month' variable exhibits notably low correlations with the price-related attributes, hovering around 0.01, suggesting a very weak linear association with these price components.

The ggcorrplot function creates a visual representation of this correlation matrix, aiding in the comprehension of relationships among these variables.

```
> # Perform linear regression
> regression_model <- lm(close ~ open, data = bit_df)
>
> # Display regression summary
> summary(regression_model)
```

```
Call:
lm(formula = close ~ open, data = bit_df)

Residuals:
    Min      1Q  Median      3Q     Max
-728.38   -5.93   -0.83    5.44  542.72

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.780690   1.731072   0.451    0.652
open        1.001853   0.001638 611.485   <2e-16 *
---
Signif. codes:  0 ‘*’ 0.001 ‘’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 52.39 on 1607 degrees of freedom
Multiple R-squared:  0.9957,   Adjusted R-squared:  0.9957
F-statistic: 3.739e+05 on 1 and 1607 DF,  p-value: < 2.2e-16
```

The regression analysis results for predicting the 'close' price based on the 'open' price are as follows:

Model Summary:

- Model Equation: close = 0.780690 + 1.001853 * open
- Residuals: The difference between predicted and actual values ranges from -728.38 to 542.72, with most falling within -5.93 to 5.44.
- Coefficients: The 'open' coefficient is 1.001853, indicating that for every unit increase in the 'open' price, the 'close' price tends to increase by approximately 1.001853 units.

Statistical Significance:

- Intercept: The intercept (0.780690) is not statistically significant (p = 0.652), suggesting that the 'close' price doesn't significantly differ from zero when the 'open' price is zero.
- Open Coefficient: The coefficient for 'open' is highly significant (p < 2e-16), indicating a strong relationship between 'open' and 'close' prices.

Goodness of Fit:

- R-squared: The R-squared value of 0.9957 indicates that approximately 99.57% of the variability in the 'close' price is explained by the 'open' price in this model.

Interpretation:

The model suggests a strong positive linear relationship between the 'open' and 'close' prices. For each unit increase in the 'open' price, the 'close' price tends to increase by approximately 1.001853 units. However, the intercept is not significantly different from zero, implying that the 'close' price might not have a meaningful value when the 'open' price is zero.

CONCLUSION

The code begins by reading and cleaning Bitcoin data, conducting statistical summaries (min, max, mean, median, SD) for 'open', 'close', 'high', and 'low' columns, presenting them in tables and plots. It explores correlations and performs linear regression between 'open' and 'close'. The summary statistics reveal the distribution characteristics, while the correlation analysis and regression help understand relationships and potential predictive patterns between variables. The visualizations aid in comprehending the data distribution and relationships visually.