VSN Sai Krishna Mohan Kocherlakota

Title: Module 3 R Practice

Submission Date: 21st November 2023

INTRODUCTION:

This R script performs comprehensive analysis and statistical testing on Bitcoin data. It includes data cleaning, summary statistics calculation, graphical representation of statistics, and hypothesis testing using one-sample and two-sample t-tests.

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

```
> 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)
+   )
>
> # Display the summary statistics in a table using kable
> kable(open_summary, caption = "Summary Statistics of 'open' column", format = "markdown")

> #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|
> #Close column
Table: Summary Statistics of 'close' column

| min_open| max_open| mean_open| median_open|  sd_open|
|--------:|--------:|---------:|-----------:|--------:|
|     68.5|  4901.42|  693.4974|      446.89| 797.3651|

Table: Summary Statistics of 'close' column

| min_open| max_open| mean_open| median_open|  sd_open|
|--------:|--------:|---------:|-----------:|--------:|
|    68.43|  4892.01|  695.5634|      447.53| 800.5576|
> #High column
Table: Summary Statistics of 'high' column

| min_open| max_open| mean_open| median_open|  sd_open|
|--------:|--------:|---------:|-----------:|--------:|
|    74.56|  4975.04|  712.7766|      452.48| 825.6228|
> #Low column
Table: Summary Statistics of 'low' column

| min_open| max_open| mean_open| median_open|  sd_open|
|--------:|--------:|---------:|-----------:|--------:|
|    65.53|  4678.53|  674.3655|       440.5| 768.1094|
> 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. Data Visualizations

a. Multiple Bar graph: -

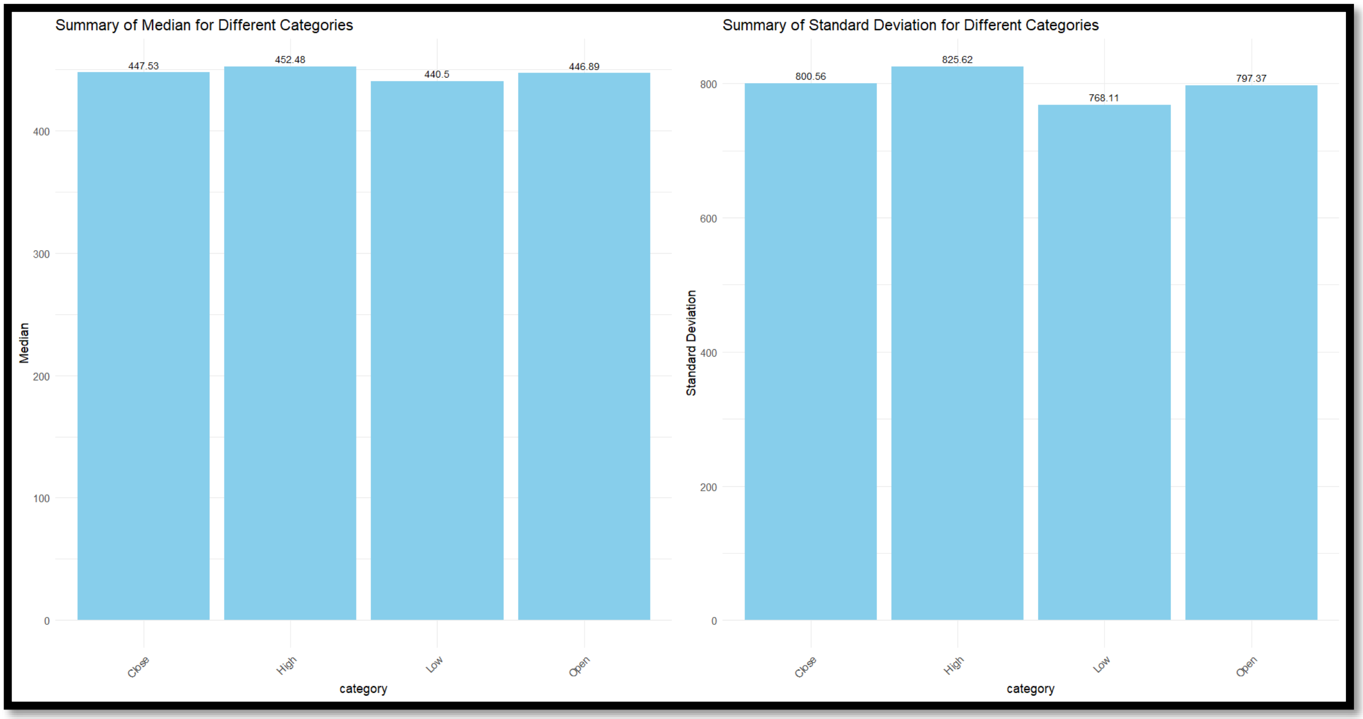

Figure 1: - Bar Graph of Median and Standard Deviation

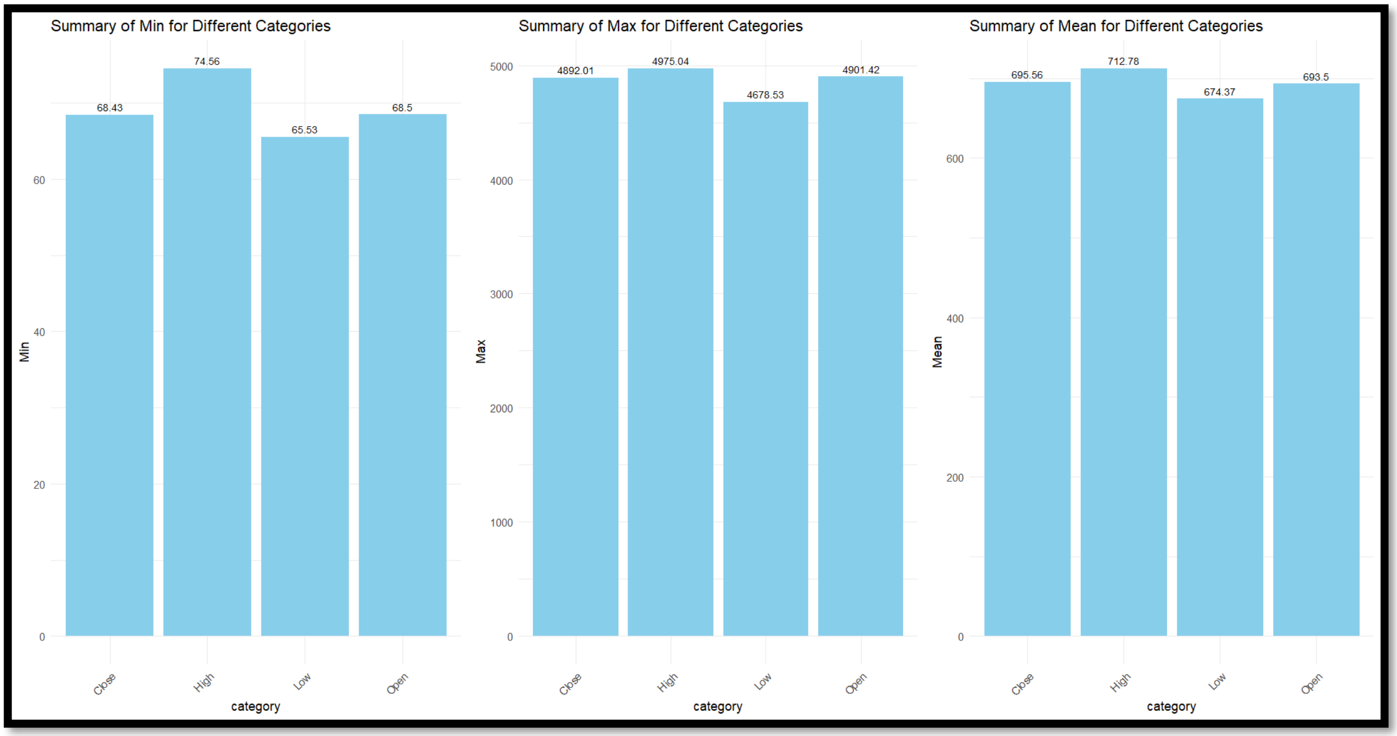

Figure 2: - Bar Graph of Minimum, Maximum and Mean

```
> #Graph
> # Function to create individual plots
> create_plot <- function(data, y_var, y_label) {
+   plot <- ggplot(data, aes(x = category, y = !!rlang::sym(y_var))) +
+     geom_bar(stat = "identity", position = position_dodge(width = 0.7), fill = "skyblue")
 +
+     geom_text(aes(label = round(!!rlang::sym(y_var), 2)), vjust = -0.5, size = 3) +
+     labs(title = paste("Summary of", y_label, "for Different Categories"), y = y_label) +
+     theme_minimal() +
+     theme(axis.text.x = element_text(angle = 45, hjust = 1))
+
+   return(plot)
+ }
>
> # Create individual plots
> plot_min_open <- create_plot(combined_summary, "min_open", "Min")
> plot_max_open <- create_plot(combined_summary, "max_open", "Max")
> plot_mean_open <- create_plot(combined_summary, "mean_open", "Mean")
> plot_median_open <- create_plot(combined_summary, "median_open", "Median")
> plot_sd_open <- create_plot(combined_summary, "sd_open", "Standard Deviation")
>
> # Combine plots into a single window
> grid.arrange(plot_min_open, plot_max_open, plot_mean_open, ncol = 3)
> grid.arrange(plot_median_open, plot_sd_open, ncol = 2)
```

The above code shows five individual plots representing various summary statistics ("Min," "Max," "Mean," "Median," and "Standard Deviation") for different categories. Finally, the `grid.arrange` function is used to display these plots in a grid layout with three plots per row. This visual representation provides insights into how these statistics vary across different categories in your data.

4. Hypothesis Testing

```
> summary(bit_df)
> #one sample t test
> t_test_open <- t.test(bit_df$open, alternative = c("less"), mu = 500, conf.level = 0.95)
> t_test_close <- t.test(bit_df$close, alternative = c("greater"), mu = 700, conf.level = 0
.95)
> t_test_high <- t.test(bit_df$high, mu = 800, conf.level = 0.95)
> t_test_low <- t.test(bit_df$low,mu = 1000, conf.level = 0.95)
> t_test_open
One Sample t-test
data:  bit_df$open
t = 9.7341, df = 1608, p-value = 1
alternative hypothesis: true mean is less than 500
95 percent confidence interval:
     -Inf 726.2132
sample estimates:
mean of x
 693.4974
> t_test_close
One Sample t-test
data:  bit_df$close
t = -0.2223, df = 1608, p-value = 0.5879
alternative hypothesis: true mean is greater than 700
95 percent confidence interval:
 662.7166      Inf
sample estimates:
mean of x
 695.5634
> t_test_high
One Sample t-test
data:  bit_df$high
t = -4.2377, df = 1608, p-value = 2.386e-05
alternative hypothesis: true mean is not equal to 800
95 percent confidence interval:
 672.4047 753.1484
sample estimates:
```

```
mean of x
 712.7766
> t_test_low
One Sample t-test
data:  bit_df$low
t = -17.005, df = 1608, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 1000
95 percent confidence interval:
 636.8060 711.9251
sample estimates:
mean of x
 674.3655
> #two sample t test
> # paired t test
> # Performing a paired t-test between 'open' and 'close' prices
> t_test_paired <- t.test(bit_df$open, bit_df$close, paired = TRUE, conf.level = 0.95)
> # Display the test results
> print(t_test_paired)
Paired t-test
data:  bit_df$open and bit_df$close
t = -1.5817, df = 1608, p-value = 0.1139
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -4.6277695  0.4959236
sample estimates:
mean difference
     -2.065923
> # independent t test
> t_test_high_low <- t.test(bit_df$high, bit_df$low, paired = FALSE,conf.level = 0.95)
> # Display the test results
> print(t_test_high_low)
Welch Two Sample t-test
data:  bit_df$high and bit_df$low
t = 1.3663, df = 3199.4, p-value = 0.1719
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -16.70998  93.53209
sample estimates:
mean of x mean of y
 712.7766  674.3655
```

The statistical analysis performed using t-tests provides insights into different aspects of your data. Here's a summary:

One-Sample t-tests:

- Open Prices: The t-test suggests that the mean open price is significantly greater than 500 (p-value < 0.05).

- Close Prices: The t-test does not provide evidence to support the claim that the mean close price is greater than 700 (p-value > 0.05).

- High Prices: Indicates that the mean high price is significantly different from 800 (p-value < 0.05).

- Low Prices: The test strongly suggests that the mean low price is significantly different from 1000 (p-value < 0.05).

Two-Sample t-tests:

- Paired t-test (Open vs. Close Prices): There’s no strong evidence to suggest a significant difference in means between open and close prices (p-value > 0.05).

- Independent t-test (High vs. Low Prices): Indicates no strong evidence to support a significant difference in means between high and low prices (p-value > 0.05).

In summary, while there are differences in means for certain variables (open, high, low) compared to specific values (500, 800, 1000), there's no significant difference observed between open and close prices or between high and low prices in your dataset.

CONCLUSION

The analysis encompassed data cleaning, summary statistics generation (including 'open', 'close', 'high', and 'low' columns), and visualizations of these statistics using bar plots. Further, it conducted one-sample t-tests to compare means against specific values and two-sample t-tests to compare means between columns. Paired t-tests scrutinized related measurements, while independent t-tests.In essence, the analysis revealed significant deviations in 'open', 'high', and 'low' columns from certain values, contrasting with the 'close' column. However, it didn't strongly support differences between 'open' and 'close' prices nor between 'high' and 'low' prices.

CITATIONS

1. Banerjee, A., Chitnis, U. B., Jadhav, S., Bhawalkar, J. S., & Chaudhury, S. (2009). Hypothesis testing, type I and type II errors. Industrial Psychiatry Journal, 18(2), 127. https://doi.org/10.4103/0972-6748.62274
2. Auguié, B. (2019, July 13). Laying out multiple plots on a page. https://cran.r-project.org/web/packages/egg/vignettes/Ecosystem.html