ALY 6070: COMMUNICATION AND VISUALIZATION FOR DATA ANALYTICS

Assignment 1:
Cryptocurrency and Valuations: Bitcoin

Submitted to:

Prof. Fatemeh Ahmadi Abkenari

Submitted by:

Abhilash Dikshit

Milan Prajapati

Shamim Sherafati

Smit Parmar

College of Professional Studies,
Northeastern University
Vancouver, Canada

I. Abstract:

This study aims to demonstrate the application of Principal Component Analysis (PCA) in R for feature reduction and determining the optimal number of components in a given dataset. The dataset used in this study is the Bitcoin cryptocurrency network data from Coin Metrics, which contains multiple variables related to the Bitcoin network. PCA is employed to transform the original variables into a smaller set of uncorrelated variables, providing insights into the underlying structure of the data and simplifying subsequent analyses. The optimal number of components is determined by visualizing the variance explained by each component and selecting the number of components that capture sufficient variance. The results of this study provide a practical example of how PCA can be used in R to perform feature reduction and enhance the efficiency and interpretability of subsequent analyses in large datasets.

II. Introduction

Principal Component Analysis (PCA) is a frequently utilized technique in data analysis and machine learning for dimensionality reduction and feature extraction. PCA is particularly useful in scenarios where a dataset contains many variables, and the objective is to identify a smaller set of variables that explain the majority of the variance in the data.

This study introduces the concept of PCA and its effectiveness in feature reduction. We then provide an overview of the dataset used in this study and describe the variables in the dataset. Subsequently, we demonstrate the implementation of PCA in R and visualize the results to identify the optimal number of components. Finally, we interpret the findings of the PCA and discuss the implications for future analysis. The study concludes by highlighting the potential applications of PCA in data analysis and its usefulness in simplifying complex datasets.

Part A: Dataset Description

The dataset comprises 2991 rows and 10 columns, out of which 10 columns are selected for the analysis, including Date, High, Low, Open, Close, Volume, and Market cap. The dataset was obtained in real-time from Kaggle website.

```
> bitcoin <- read.csv("/Users/abidikshit/R_Projects/Data/coin_Bitcoin.csv", header = T)
> cat("Number of Rows before cleanup:", nrow(bitcoin), "\n") # Printing string and variable row count on the same line
Number of Rows before cleanup: 2991
> cat("Number of Columns before cleanup:", ncol(bitcoin), "\n")
Number of Columns before cleanup: 10
> cat("Blank cells count before cleanup:", sum(!complete.cases(bitcoin))) # Displaying Blank Cells Count for uncleaned data
Blank cells count before cleanup: 0
```

The dataset did not contain any missing or null values; thus, no cleaning or manipulation of the data was required.

```
## Convert date column to Date format
```{r}
bitcoin$Date <- as.Date(bitcoin$Date)
```
```

The class type of the "Date" column was changed from character to Date.

SNo <dbl>	Name <chr>	Symbol <chr>	Date <date>	High <dbl>	Low <dbl>	Open <dbl>	Close <dbl>	Volume <dbl>	Marketcap <dbl>
1	Bitcoin	BTC	2013-04-29	147.49	134.00	134.44	144.54	0	1603768864
2	Bitcoin	BTC	2013-04-30	146.93	134.05	144.00	139.00	0	1542813125
3	Bitcoin	BTC	2013-05-01	139.89	107.72	139.00	116.99	0	1298954594
4	Bitcoin	BTC	2013-05-02	125.60	92.28	116.38	105.21	0	1168517495
2988	Bitcoin	BTC	2021-07-03	34909.26	33402.70	33854.42	34668.55	24383958643	649939701346
2989	Bitcoin	BTC	2021-07-04	35937.57	34396.48	34665.56	35287.78	24924307911	661574836315
2990	Bitcoin	BTC	2021-07-05	35284.34	33213.66	35284.34	33746.00	26721554282	632696207200
2991	Bitcoin	BTC	2021-07-06	35038.54	33599.92	33723.51	34235.19	26501259870	641899161594

The headTail() function was used to display the first and last four rows, as shown in the table above.

```
      SNo             Name              Symbol              Date                High              Low        
 Min.   :   1.0   Length:2991        Length:2991        Min.   :2013-04-29   Min.   :   74.56   Min.   :   65.53  
 1st Qu.: 748.5   Class :character   Class :character   1st Qu.:2015-05-16   1st Qu.:  436.18   1st Qu.:  422.88  
 Median :1496.0   Mode  :character   Mode  :character   Median :2017-06-02   Median : 2387.61   Median : 2178.50  
 Mean   :1496.0                                         Mean   :2017-06-02   Mean   : 6893.33   Mean   : 6486.01  
 3rd Qu.:2243.5                                         3rd Qu.:2019-06-19   3rd Qu.: 8733.93   3rd Qu.: 8289.80  
 Max.   :2991.0                                         Max.   :2021-07-06   Max.   :64863.10   Max.   :62208.96  
      Open              Close               Volume              Marketcap        
 Min.   :   68.5   Min.   :   68.43   Min.   :           0   Min.   :    778411179  
 1st Qu.:  430.4   1st Qu.:  430.57   1st Qu.:    30367250   1st Qu.:   6305579329  
 Median : 2269.9   Median : 2286.41   Median :   946035968   Median :  37415031061  
 Mean   : 6700.1   Mean   : 6711.29   Mean   : 10906334005   Mean   : 120876059113  
 3rd Qu.: 8569.7   3rd Qu.: 8576.24   3rd Qu.: 15920149610   3rd Qu.: 149995739946  
 Max.   :63523.8   Max.   :63503.46   Max.   :350967941479   Max.   :1186364044140  
```

A summary of the dataset was also displayed to gain a better understanding of the variables.

Part B: Variable Description

The variables in the dataset include Name, Crypto Name, Symbol, Crypto Symbol, Date, High, Low, Open, Close, Volume, and Market cap. The variables that were considered for the analysis include Date, High, Low, Open, Close, Volume, and Market cap.

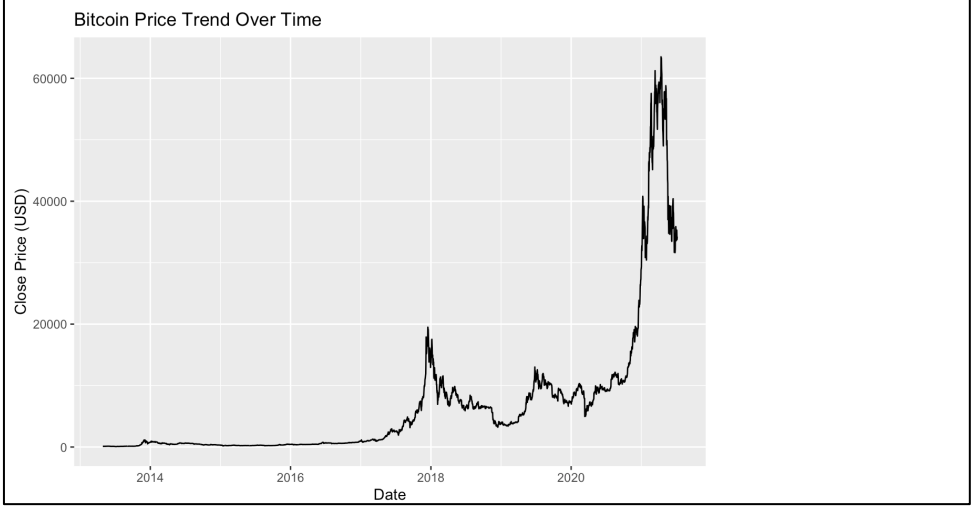
Name	Crypto Name
Symbol	Crypto Symbol
Date	date of observation
High	Highest price on the given day
Low	Lowest price on the given day
Open	Opening price on the given day
Close	Closing price on the given day
Volume	Volume of transactions on the given day
Marketcap	Market capitalization in USD

3. Explore Data features with different types of graphs in R.

Now, let's create different types of graphs to explore the data features:

As we can see, Although, the price of the bitcoin was fluctuating but in overall its price has increased over the time.

```
ggplot(bitcoin, aes(x = Date, y = Close)) + geom_line() + xlab("Date") + ylab("Close Price (USD)") + ggtitle("Bit
coin Price Trend Over Time")
```



Histogram - to show the distribution of Bitcoin closing prices:

The histogram shows the distribution of Bitcoin closing prices. The x-axis represents the closing price of Bitcoin in USD, while the y-axis represents the frequency of that price. The binwidth of the histogram is set to 50 USD, meaning that each bar in the histogram represents a range of 50 USD. From the histogram, we can see that the majority of Bitcoin closing prices fall between 0 to 1000 USD, with a few outliers above 1000 USD. Overall, the histogram provides a visual representation of the distribution of Bitcoin closing prices and can help identify patterns and outliers in the data.

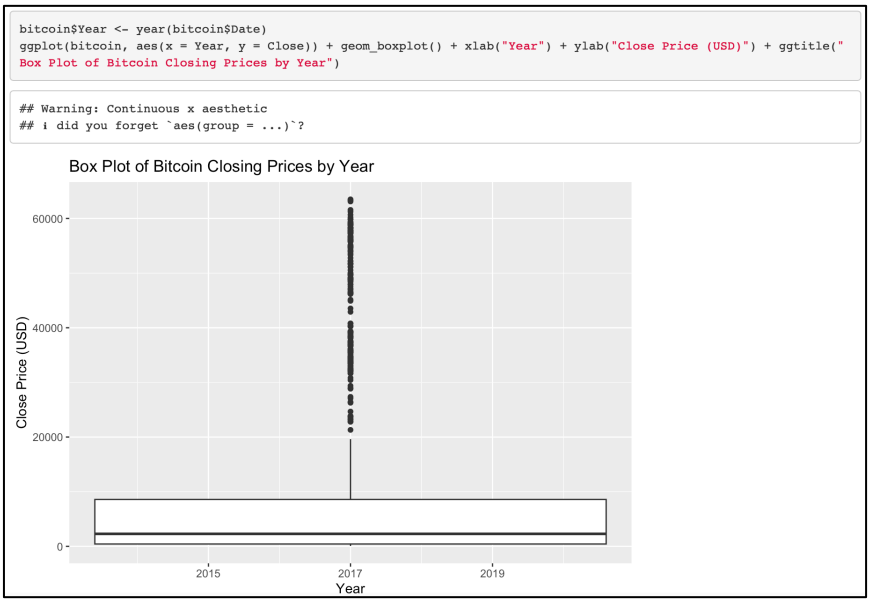
```
ggplot(bitcoin, aes(x = Close)) + geom_histogram(binwidth = 50) + xlab("Close Price (USD)") + ylab("Frequency") +
ggtitle("Histogram of Bitcoin Closing Prices")
```

Box plot - to show the distribution of Bitcoin closing prices by year:

The box plot allows us to compare the distribution of Bitcoin closing prices across different years from 2015 to 2019.

```
bitcoin$Year <- year(bitcoin$Date)
ggplot(bitcoin, aes(x = Year, y = Close)) + geom_boxplot() + xlab("Year") + ylab("Close Price (USD)") + ggtitle("
Box Plot of Bitcoin Closing Prices by Year")
```

```
## Warning: Continuous x aesthetic
## ℹ did you forget `aes(group = ...)`?
```

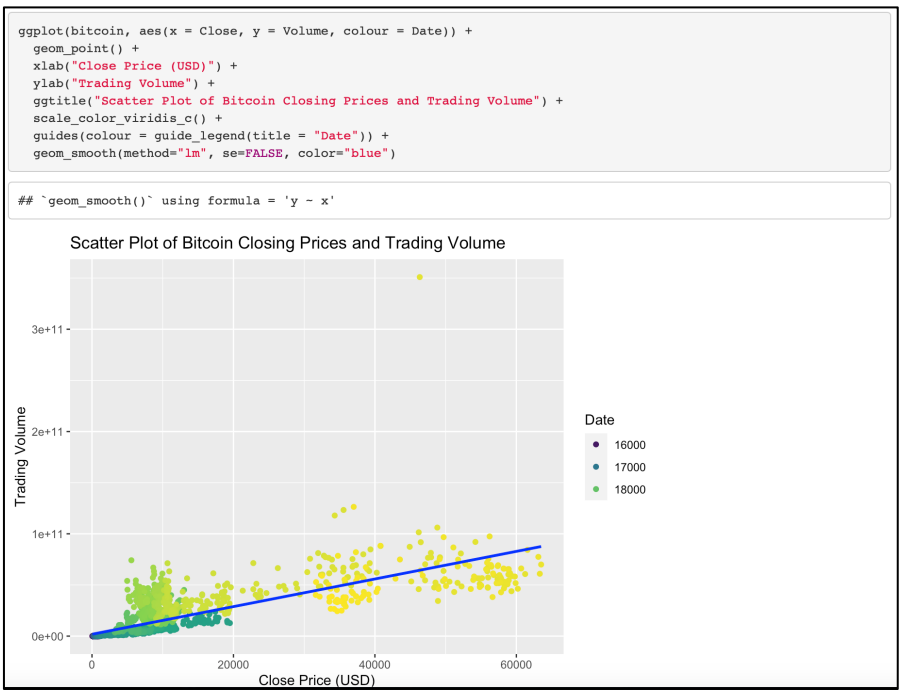


Scatter plot - to show the relationship between Bitcoin closing prices and trading volume:

This scatter plot shows the relationship between Bitcoin closing prices and trading volume. The x-axis represents the closing price of Bitcoin in USD, while the y-axis represents the trading volume. As the blue regression line that shows the trend in the data, we can see that it has a positive trend by increasing the close price. This line is calculated using linear regression, which helps to identify any correlation between the Bitcoin closing price and trading volume.

```
ggplot(bitcoin, aes(x = Close, y = Volume, colour = Date)) +
  geom_point() +
  xlab("Close Price (USD)") +
  ylab("Trading Volume") +
  ggtitle("Scatter Plot of Bitcoin Closing Prices and Trading Volume") +
  scale_color_viridis_c() +
  guides(colour = guide_legend(title = "Date")) +
  geom_smooth(method="lm", se=FALSE, color="blue")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

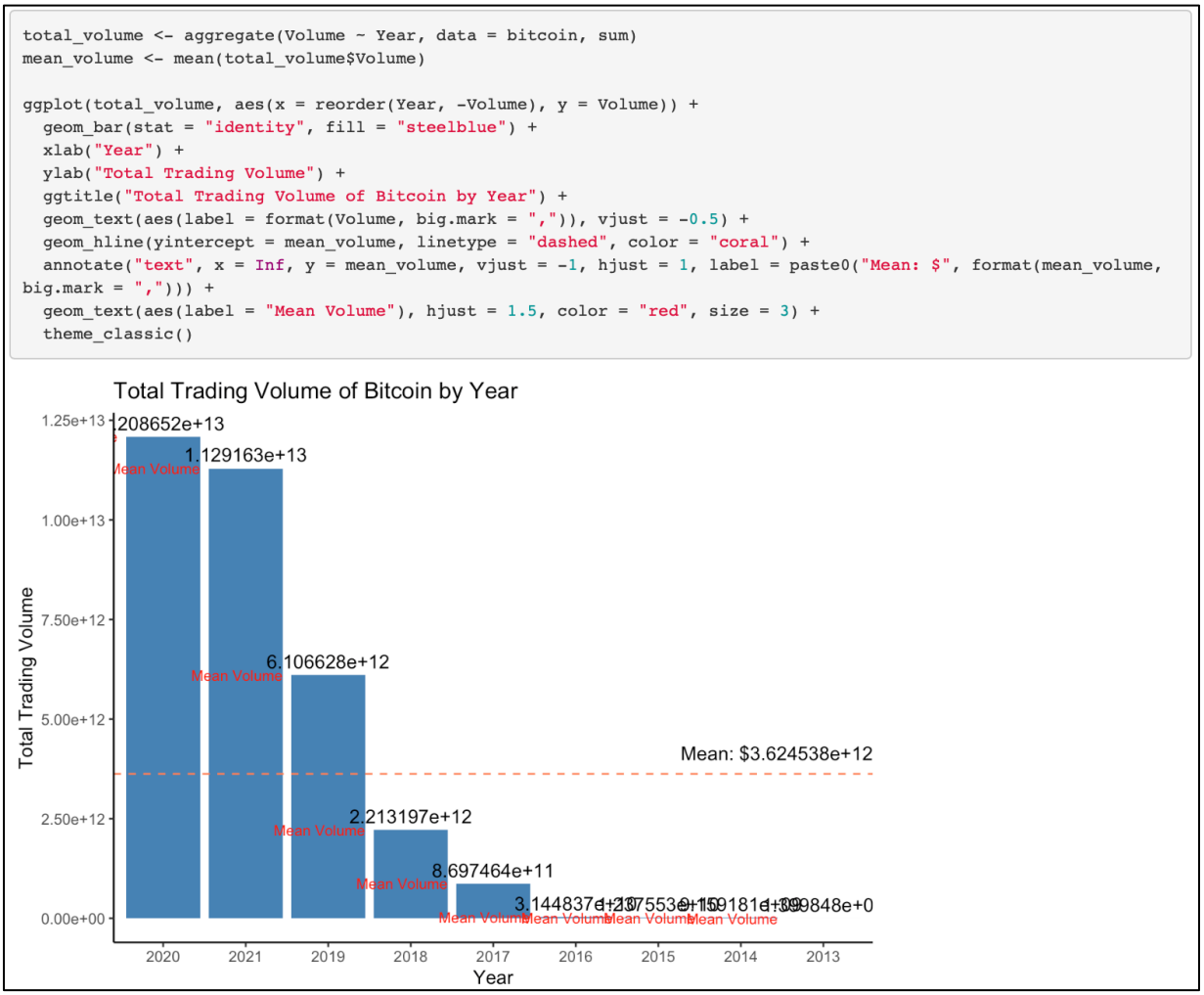


Bar chart - to show the total trading volume of Bitcoin by year:

This bar chart shows the total trading volume of Bitcoin by year. The chart uses data from the "bitcoin" dataset and aggregates the volume by year. The chart also includes a dashed line representing the mean trading volume and a text annotation for the mean value which is 3.62$. The red text indicates the label for the mean volume, and the text above each bar shows the total trading volume for that year. Overall, this chart provides a clear visual representation of the Bitcoin trading volume by year, allowing for easy comparison and analysis.

```
total_volume <- aggregate(Volume ~ Year, data = bitcoin, sum)
mean_volume <- mean(total_volume$Volume)

ggplot(total_volume, aes(x = reorder(Year, -Volume), y = Volume)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  xlab("Year") +
  ylab("Total Trading Volume") +
  ggtitle("Total Trading Volume of Bitcoin by Year") +
  geom_text(aes(label = format(Volume, big.mark = ",")), vjust = -0.5) +
  geom_hline(yintercept = mean_volume, linetype = "dashed", color = "coral") +
  annotate("text", x = Inf, y = mean_volume, vjust = -1, hjust = 1, label = paste0("Mean: $", format(mean_volume,
big.mark = ","))) +
  geom_text(aes(label = "Mean Volume"), hjust = 1.5, color = "red", size = 3) +
  theme_classic()
```



Run a PCA approach in R to do a feature reduction.

Let's run a Principal Component Analysis (PCA) approach in R to do a feature reduction on the cryptocurrency price history dataset. First, let's load the dataset and prepare it for PCA. We will use the dataset, and select only the Open, High, Low, Close, Volume, and Market cap columns:

```
bitcoin_pca <- read_csv("/Users/abidikshit/R_Projects/Data/coin_Bitcoin.csv", col_types = cols_only(Date = col_da
te(), Open = col_double(), High = col_double(), Low = col_double(), Close = col_double(), Volume = col_double(),
Marketcap = col_double()))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
bitcoin_pca <- bitcoin_pca[,c(2:7)]
```

Next, we will scale the data to have zero mean and unit variance:

```
scaled_bitcoin <- scale(bitcoin_pca)
```

Now, we can run the PCA using the prcomp() function:

```
pca_bitcoin <- prcomp(scaled_bitcoin, scale = TRUE)
```

We can now explore the results of the PCA, starting with the summary of the PCA object:

```
summary(pca_bitcoin)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.3857 0.5521 0.04500 0.03276 0.01892 0.01157
## Proportion of Variance 0.9486 0.0508 0.00034 0.00018 0.00006 0.00002
## Cumulative Proportion  0.9486 0.9994 0.99974 0.99992 0.99998 1.00000
```

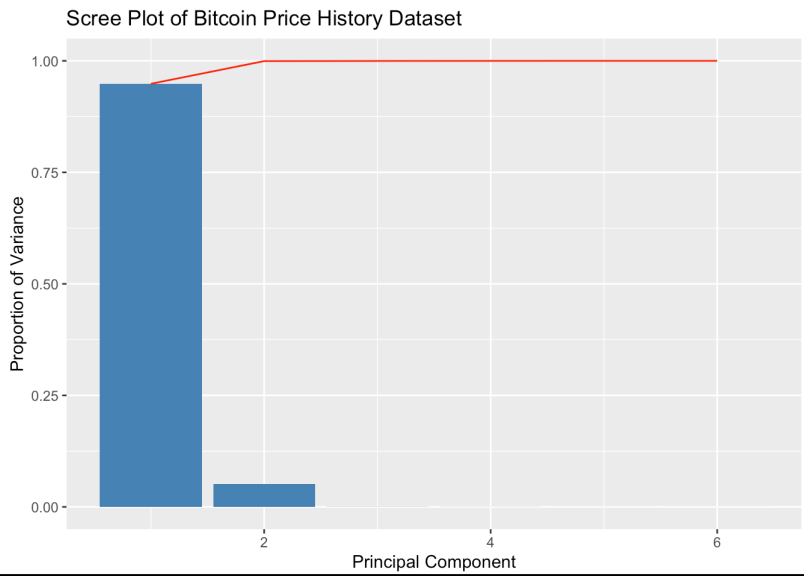
The first PC has a relatively high standard deviation and proportion of variance (0.9486), indicating that it captures a significant amount of the variability in the original dataset. The subsequent PCs have much lower values for both standard deviation and proportion of variance, suggesting that they explain much less of the variability in the data. The cumulative proportion shows that the first PC alone explains almost all (94.86%) of the total variance in the data.

- So, the proportion of variance explained by the first principal component is: 0.9486 x 100 = 94.86%
- And the proportion of variance explained by the second principal component is: 0.0508 x 100 = 5.08%

This indicates the proportion of total variance captured by all the principal components up to that point. The first principal component captures 94.86% of the total variance, and the cumulative proportion increases with each subsequent component until it reaches 100% at PC6.

We can also plot a scree plot to visualize the proportion of variance explained by each principal component:

```
scree_plot <- ggplot(data.frame(PC = 1:6, Variance = pca_bitcoin$sdev^2 / sum(pca_bitcoin$sdev^2)), aes(x = PC, y
= Variance)) + geom_bar(stat = "identity", fill = "steelblue") + geom_line(aes(x = PC, y = cumsum(Variance)), col
or = "red") + xlab("Principal Component") + ylab("Proportion of Variance") + ggtitle("Scree Plot of Bitcoin Price
History Dataset")
scree_plot
```



This scree plot is a graph that shows the proportion of variance explained by each principal component in descending order. It helps us determine the number of principal components to retain for further analysis.

4.Decide on the best number of components based on the PCA visualization.

Finally, we can extract the principal components using the predict() function:

After conducting PCA on the Bitcoin price history dataset, we obtain principal components which are a linear combination of the original variables. These principal components are orthogonal and uncorrelated, which helps reduce the dimensionality of the dataset while still retaining most of its information.

```
pcs_bitcoin <- predict(pca_bitcoin, scaled_bitcoin)
```

The above code uses the predict() function to project the original scaled_bitcoin data set onto the principal components generated from a PCA analysis (pca_bitcoin). The resulting projected data set is stored in a new object called pcs_bitcoin.
The predict() function uses the pca_bitcoin object as the first argument and the scaled_bitcoin data set as the second argument. The scaled_bitcoin data set has the same variables as the original bitcoin data set, but the variables have been scaled to have a mean of zero and a standard deviation of one.
By applying the predict() function to the pca_bitcoin object and the scaled_bitcoin data set, the function uses the weights and loadings calculated during the PCA analysis to transform the original data set into a new set of variables (the principal components). This process is commonly referred to as dimensionality reduction or feature extraction. The resulting pcs_bitcoin object contains the projected data set, which is now represented by the principal

components instead of the original variables.

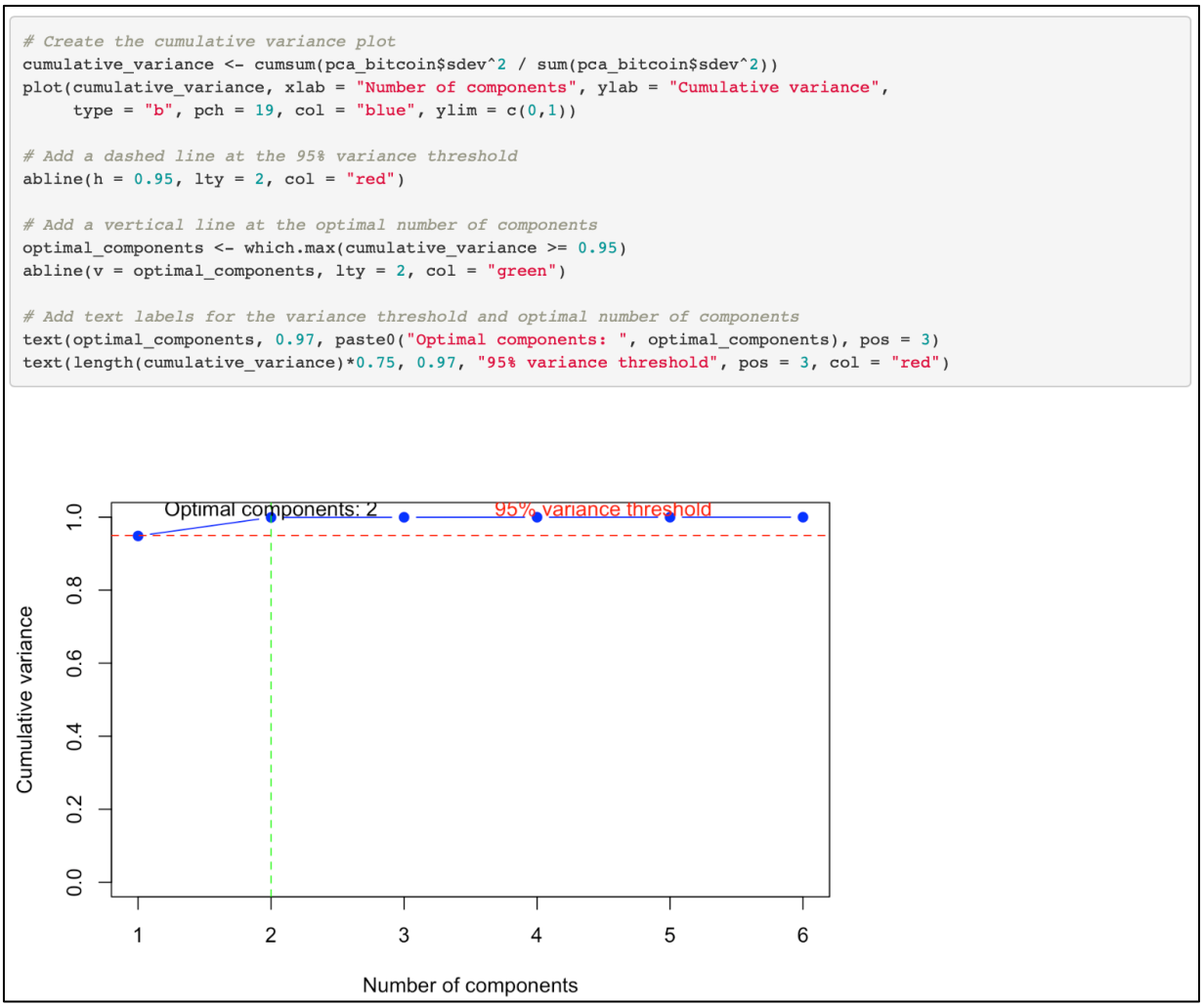
Plot the cumulative variance plot:

```r
# Create the cumulative variance plot
cumulative_variance <- cumsum(pca_bitcoin$sdev^2 / sum(pca_bitcoin$sdev^2))
plot(cumulative_variance, xlab = "Number of components", ylab = "Cumulative variance",
     type = "b", pch = 19, col = "blue", ylim = c(0,1))

# Add a dashed line at the 95% variance threshold
abline(h = 0.95, lty = 2, col = "red")

# Add a vertical line at the optimal number of components
optimal_components <- which.max(cumulative_variance >= 0.95)
abline(v = optimal_components, lty = 2, col = "green")

# Add text labels for the variance threshold and optimal number of components
text(optimal_components, 0.97, paste0("Optimal components: ", optimal_components), pos = 3)
text(length(cumulative_variance)*0.75, 0.97, "95% variance threshold", pos = 3, col = "red")
```



The cumulative variance plot shows the cumulative proportion of variance explained by each principal component in the Bitcoin Price History dataset. The x-axis represents the number of principal components, while the y-axis represents the cumulative proportion of variance.
As we can see from the blue line, which shows the cumulative proportion of variance explained by each additional principal component, the blue line in the cumulative variance plot at first it was increasing and then stays steady after the second component.
The red dashed line indicates the 95% variance threshold, which is commonly used as a cutoff for the number of principal components to retain in a PCA analysis.
The green dashed line shows the optimal number of components, which is the number of components at or just below the 95% variance threshold and intersects with the plot at the second component.
Overall, As the blue line in the cumulative variance plot stays steady after the second component, and that the green dashed line intersects with the plot at the second component.
In this case, choosing two components would explain 95% of the variance in the data, which is a good amount of the total variance. Therefore, using two components may be a reasonable choice for dimensionality reduction or further analysis. So, the optimal number of

components would be 2.

5.References

- Silge, M. K. A. J. n.d. *16 Dimensionality Reduction | Tidy Modeling with r*. https://www.tmwr.org/dimensionality.html.
- Wood, R. 2021, December 14. *Learn Principal Component Analysis in r - Towards Data Science*. https://towardsdatascience.com/learn-principle-component-analysis-in-r-ddba7c9b1064.

6.Appendix

```
my_packages = c("plyr", "plotly", "ggplot2", "psych", "tidyr", "tidyverse","dplyr","lubridate","readr","caret")

#install.packages(my_packages)

lapply(my_packages, require, character.only = T)


#Dataset used from https://coinmetrics.io/community-network-data/

bitcoin <- read.csv("/Users/abidikshit/R_Projects/Data/coin_Bitcoin.csv", header = T)

cat("Number of Rows before cleanup:", nrow(bitcoin), "\n") # Printing string and variable row count on the same line

cat("Number of Columns before cleanup:", ncol(bitcoin), "\n")

cat("Blank cells count before cleanup:", sum(!complete.cases(bitcoin))) # Displaying Blank Cells Count for uncleaned data


bitcoin$Date <- as.Date(bitcoin$Date)


headTail(bitcoin, top = 4, bottom = 4, ellipsis = F)


summary(bitcoin)


ggplot(bitcoin, aes(x = Date, y = Close)) + geom_line() + xlab("Date") + ylab("Close Price (USD)") + ggtitle("Bitcoin Price Trend Over Time")


ggplot(bitcoin, aes(x = Close)) + geom_histogram(binwidth = 50) + xlab("Close Price (USD)") + ylab("Frequency") + ggtitle("Histogram of Bitcoin Closing Prices")


bitcoin$Year <- year(bitcoin$Date)
```

```r
ggplot(bitcoin, aes(x = Year, y = Close)) + geom_boxplot() + xlab("Year") + ylab("Close Price (USD)") + ggtitle("Box Plot of Bitcoin Closing Prices by Year")


ggplot(bitcoin, aes(x = Close, y = Volume, colour = Date)) +
  geom_point() +
  xlab("Close Price (USD)") +
  ylab("Trading Volume") +
  ggtitle("Scatter Plot of Bitcoin Closing Prices and Trading Volume") +
  scale_color_viridis_c() +
  guides(colour = guide_legend(title = "Date")) +
  geom_smooth(method="lm", se=FALSE, color="blue")


total_volume <- aggregate(Volume ~ Year, data = bitcoin, sum)
mean_volume <- mean(total_volume$Volume)


ggplot(total_volume, aes(x = reorder(Year, -Volume), y = Volume)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  xlab("Year") +
  ylab("Total Trading Volume") +
  ggtitle("Total Trading Volume of Bitcoin by Year") +
  geom_text(aes(label = format(Volume, big.mark = ",")), vjust = -0.5) +
  geom_hline(yintercept = mean_volume, linetype = "dashed", color = "coral") +
  annotate("text", x = Inf, y = mean_volume, vjust = -1, hjust = 1, label = paste0("Mean: $", format(mean_volume, big.mark = ","))) +
  geom_text(aes(label = "Mean Volume"), hjust = 1.5, color = "red", size = 3) +
  theme_classic()



bitcoin_pca <- read_csv("/Users/abidikshit/R_Projects/Data/coin_Bitcoin.csv", col_types = cols_only(Date = col_date(), Open = col_double(), High = col_double(), Low = col_double(), Close = col_double(), Volume = col_double(), Marketcap = col_double()))
bitcoin_pca <- bitcoin_pca[,c(2:7)]


scaled_bitcoin <- scale(bitcoin_pca)


pca_bitcoin <- prcomp(scaled_bitcoin, scale = TRUE)


summary(pca_bitcoin)
```

```r
scree_plot <- ggplot(data.frame(PC = 1:6, Variance = pca_bitcoin$sdev^2 / sum(pca_bitcoin$sdev^2)), aes(x = PC, y = Variance)) + geom_bar(stat = "identity", fill = "steelblue") + geom_line(aes(x = PC, y = cumsum(Variance)), color = "red") + xlab("Principal Component") + ylab("Proportion of Variance") + ggtitle("Scree Plot of Bitcoin Price History Dataset")

scree_plot


pcs_bitcoin <- predict(pca_bitcoin, scaled_bitcoin)


plot(cumsum(pca_bitcoin$sdev^2 / sum(pca_bitcoin$sdev^2)), xlab = "Number of components", ylab = "Cumulative variance")


# Choose the number of components

n_components <- length(which(cumsum(pca_bitcoin$sdev^2 / sum(pca_bitcoin$sdev^2)) < 0.8)) + 1

cat("Number of components:", n_components, "\n")


df_pca_components <- predict(pca_bitcoin, scaled_bitcoin) %>%

  as.data.frame() %>%

  rename_all(~ paste0("PC", .))


#df_pca_components$Date <- df$Date


headTail(df_pca_components,top = 3, bottom = 3, ellipsis = 0)


## NA
```