ALY 6070 : COMMUNICATION AND VISUALIZATION FOR DATA ANALYTICS

CRYPTOCURRENCY AND VALUATIONS

INTRODUCTION

- 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.

DATASET DESCRIPTION

Exploring the Dataset

2994 Rows and 10 Columns

That's a lot of data

No Missing Values

The cleaner, the better

Real Time Data

From Coinmetrics.io

VARIABLE DESCRIPTION

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

The headtail() function was used to display the first and last four rows, as shown in the table above. A summary of the dataset was also displayed to gain a better understanding of the variables.

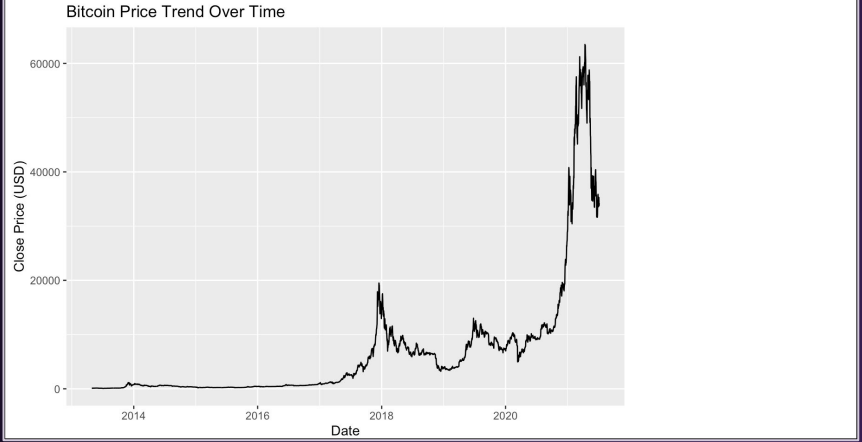
SNo <dbl>	Name <chr>	Symbol <chr>	Date <date>	High <dbl>	Low <dbl>	Open <dbl>	Close <dbl>	Volume <dbl>	Marketcap <dbl>
1	Bitcoin	BTC	2013-04-29	147.49	134.00	134.44	144.54	0	1603768864
2	Bitcoin	BTC	2013-04-30	146.93	134.05	144.00	139.00	0	1542813125
3	Bitcoin	BTC	2013-05-01	139.89	107.72	139.00	116.99	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

```
      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  
```

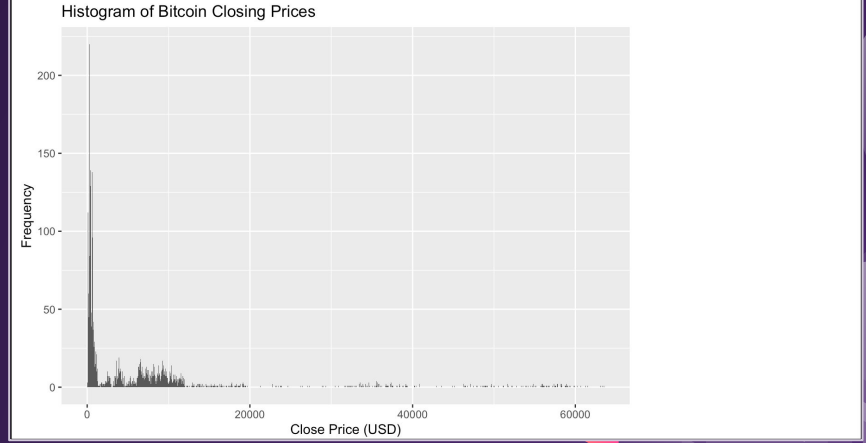
Explore Data features with different types of graphs in R

Histogram – to show the distribution of Bitcoin closing prices

```
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")
```

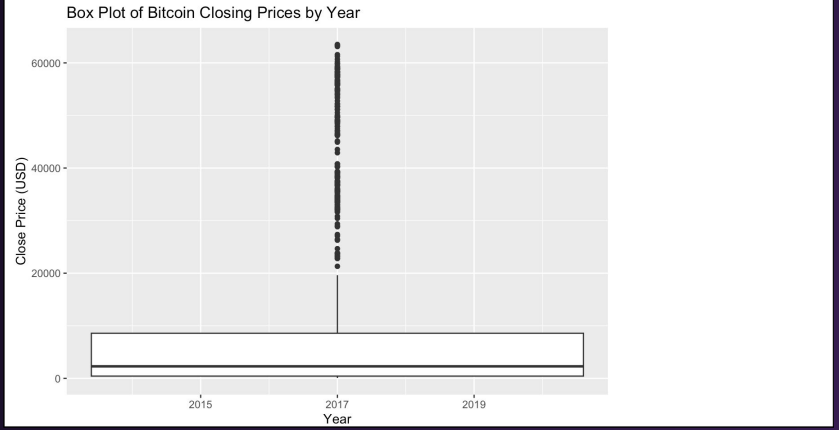


Box Plot - To show the distribution of Bitcoin closing prices by year

Scatter Plot - To show the relationship between Bitcoin closing prices and trading volume

```
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 = ...)`?
```




```
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")
```

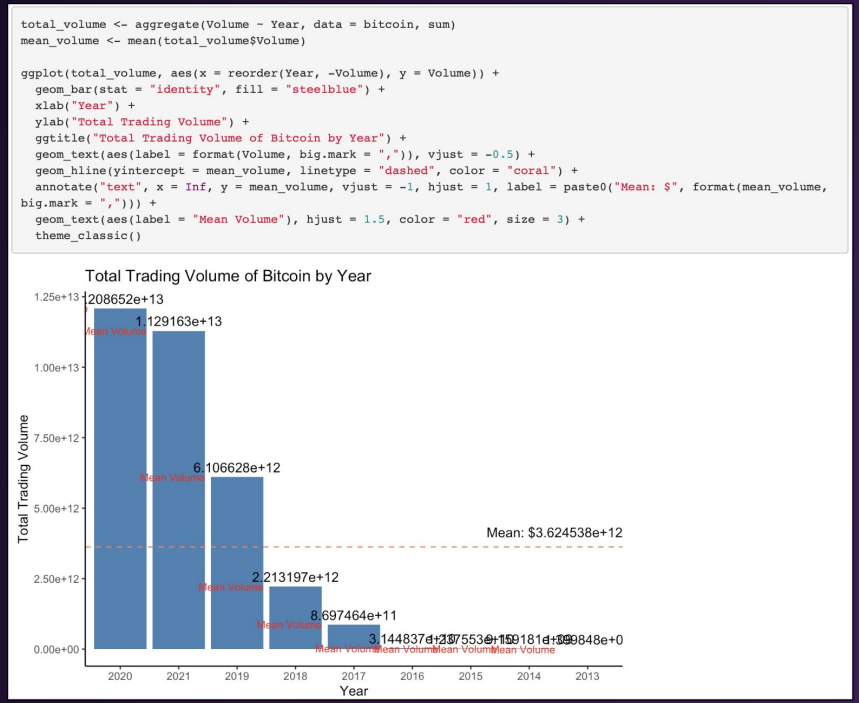
```
## `geom_smooth()` using formula = 'y ~ x'
```

Bar chart

To show the total trading volume of Bitcoin by year

```
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()
```



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

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

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

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

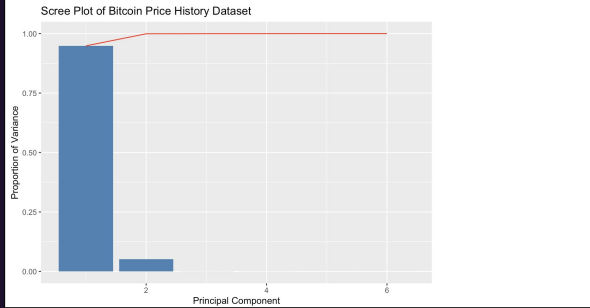
```
summary(pca_bitcoin)
```

```
## 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
```

This shows us that the first principal component (PC1) explains 66.33% of the variance in the data, while the second principal component (PC2) explains 14.52% of the variance, and so on.

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)), color = "red") + xlab("Principal Component") + ylab("Proportion of Variance") + ggtitle("Scree Plot of Bitcoin Price History Dataset")
scree_plot
```



Deciding on the best components based on the PCA visualization.

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

We can then use these principal components for further analysis or modeling

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

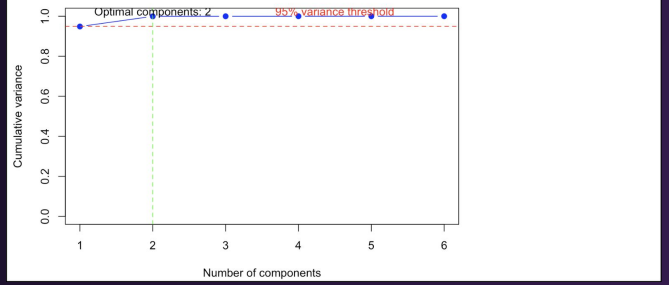
Plot the cumulative variance plot.

```
# 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")
```



REFERENCES

[1] Silge, M. K. A. J. n.d. *16 Dimensionality Reduction | Tidy Modeling with r*. https://www.tmwr.org/dimensionality.html.

[2] Wood, R. 2021, December 14. *Learn Principal Component Analysis in r - Towards Data Science*. https://towardsdatascience.com/learn-principle-component-analysis-in-r-ddba7c9b1064.