Clustering Stock Market Time Series based on Cumulative Weighted Slopes

CIVE 7100: Applied Time Series and Spatial Statistics

Faculty Advisor: Prof. Auroop Ganguly

Teaching Assistant: Udit Bhatia

Name: Sagar Ghiya

M.S in Operations Research

Contents

Abstract

Clustering time series finds important applications in various domains. One such important application is clustering stocks based on closing prices to identify stocks that behave similar to one another. For this, time series needs to be condensed to one single point. One parameter that can identify similarity between time series is slope of trend. However, for a long time series slope will be different at different intervals. Cumulative weighted slopes method can be used to overcome this problem. This method is based on concept that similar time series will have similar slope at corresponding points. In this method, time series is divided into intervals of same length and cumulative weighted slope is calculated. Now time series is reduced to a single point and k means clustering can be done on it to identify stocks with similarity. One more inference that can be made from this project is that time series belonging to similar industry don't always fall in the same cluster.

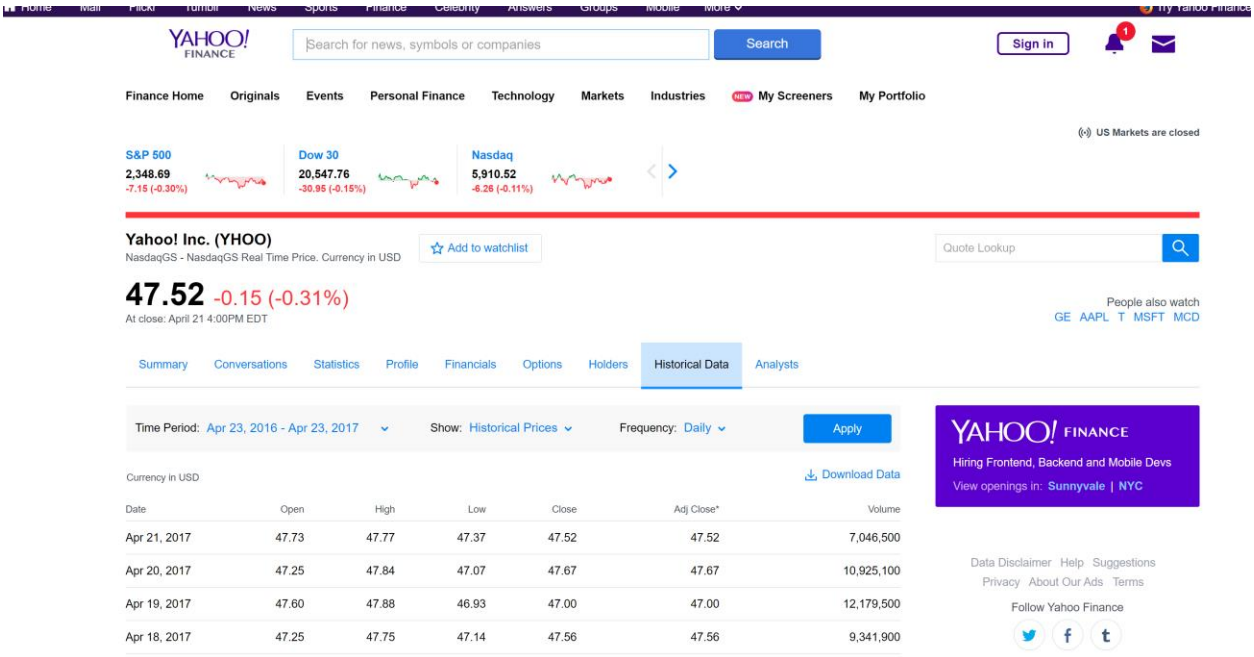
Introduction

The purpose of this project is to cluster stocks based on their similarity using cumulative weighted slopes method. Process is referred to as whole sequence clustering. Cumulative weighted slope can be defined as the sum of the weighted slopes of the given time sequence computed on a point-to-point basis. The parameters representing these cumulative slopes can be clustered using k means to identify similar patterns. One more takeaway from this project is to observe that whether stocks belonging to similar industry fall in same cluster or not. So data is collected for closing price of 25 stocks, 5 industry categories having 5 stocks each. The process of this project is as follows:

1. Data Collection
2. Data Processing and Data shaping
3. Feature Extraction using Cumulative Weighted Slopes
4. K means clustering

Data Collection

- Downloaded previous 2 years of closing stock price data as .csv file from Yahoo Finance.
- Total 25 datasets downloaded and then imported into R.
- Stocks belonged to 5 industries, Airline, Technology, Retail, Banking and Healthcare with 5 stocks from each.
- Airline: American Airlines, Spirit Airlines, Southwest, Delta, JetBlue
- Technology: Google, Apple, Microsoft, IBM, Intel
- Retail: Walmart, Amazon, Costco, Walgreens, Target
- Banking: Bank of America, Capital One, Goldman Sachs, Morgan Stanley, Wells Fargo
- Healthcare: Aetna, Amerisource Bergen, CVS Health, Johnson & Johnson, Mckesson



Data Processing and Data Shaping

- Scaling data for dates and closing stock prices to avoid any distortions.
- Combining all the data into a single data frame with first column for dates and then 25 more columns containing stock prices for 25 stocks.
- There are a total 505 rows i.e data is collected for stock prices for 505 days.
- Data is divided into 10 strips with each strip composed of 50 days or 50 rows.
- Then cumulative weighted slope can be calculated for each of 10 strips for all 25 stocks.

Filter

	Date	American Airlines	Delta	Jet Blue	Southwest	Spirit Airlines	Bank of America	Capital One	Goldman Sachs	Morgan Stanley	Wells Fargo	Aetna	AmerisourceBergen	CVS Health	Johns & Johns
1	2017-04-07	41.81000	45.17000	21.10	54.54000	51.85	23.16000	83.54000	227.8800	41.95000	54.84000	128.3600	87.62000	77.08000	124.
2	2017-04-06	41.72000	45.26000	20.91	53.38000	51.75	23.26000	84.23000	228.6400	42.09000	55.37000	128.8400	87.64000	77.01000	125.
3	2017-04-05	41.31000	45.08000	20.57	53.35000	50.90	23.17000	84.13000	227.6600	41.74000	54.98000	127.2800	87.18000	77.03000	124.
4	2017-04-04	40.90000	45.11000	20.51	53.07000	51.41	23.44000	85.26000	229.2600	42.51000	55.20000	127.8500	86.93000	77.87000	124.
5	2017-04-03	42.45000	46.32000	20.64	54.00000	53.16	23.59000	85.80000	228.9600	42.71000	55.49000	128.7400	87.33000	78.05000	124.
6	2017-03-31	42.30000	45.96000	20.61	53.76000	53.07	23.59000	86.66000	229.7200	42.84000	55.66000	127.5500	88.50000	78.50000	124.
7	2017-03-30	42.54000	46.27000	20.73	53.83000	52.19	23.87000	87.14000	231.2200	43.43000	56.24000	126.9100	89.14000	78.85000	124.
8	2017-03-29	41.96000	45.95000	20.64	53.48000	51.53	23.35000	84.68000	228.4500	42.80000	55.67000	125.7300	89.51000	78.76000	124.
9	2017-03-28	42.60000	46.53000	20.82	54.19000	52.02	23.48000	84.17000	229.3300	42.49000	55.96000	127.0100	87.50000	78.61000	125.
10	2017-03-27	41.74000	46.10000	20.29	52.88000	50.98	23.03000	82.13000	225.4800	41.58000	55.39000	126.2600	87.48000	78.51000	125.
11	2017-03-24	41.73000	46.00000	19.94	52.61000	50.70	23.12000	83.80000	228.4100	42.46000	55.83000	126.7700	86.56000	78.49000	125.
12	2017-03-23	41.41000	45.92000	19.92	52.37000	50.30	23.07000	83.78000	231.9000	42.59000	55.25000	127.8300	85.99000	78.30000	125.
13	2017-03-22	40.35000	45.74000	19.80	52.46000	50.48	22.94000	83.68000	231.0700	42.21000	55.33000	128.3300	85.88000	78.46000	126.
14	2017-03-21	40.42000	45.52000	19.76	51.88000	50.09	23.02000	83.65000	233.0000	42.66000	55.85000	129.0900	85.91000	78.61000	127.

Feature Extraction using Cumulative Weighted Slope

- After data is split into 10 strips, Cumulative Weighted Slope is calculated for each time series.
- Cumulative Weighted slope is given by:

$$WS_{sq}(X_i) = \sqrt[2]{\sum_{k=1}^{m} S_{ik}^{2} * (k/m)}$$

- Here S represents slope of trend which can be calculated as (y2-y1)/(x2-x1).
- For weighting purpose, slope is squared and multiplied by weight which is (k/m).
- K represents number of strip which is 1 for 1:50, 2 for 51:100 and so on..
- M represents total number of strips =10.
- Weighted slopes of all the strips are added up and then square rooted.
- By this method, time series is condensed into a single point.
- 25 features(Cumulative Weighted Slope) are extracted for 25 different time series.

American Airlines 0.16260150
Delta 0.14102179
Jet Blue 0.08614317
Southwest 0.21414005
Spirit Airlines 0.35141625
Bank of America 0.06300926
Capital One 0.26630365
Goldman Sachs 0.68573408
Morgan Stanley 0.14165675
Wells Fargo 0.11358844
Aetna 0.44943351
AmerisourceBergen 0.23153482
CVS Health 0.21860669
Johnson & Johnson 0.18190790
Mckesson 0.66798939
Amazon 2.53980331
Costco 0.28097529
Target 0.20379733
Walgreens 0.12256887
Walmart 0.19840078
Apple 0.40343819
Google 2.03757094
IBM 0.44627394
Intel 0.06338359
Microsoft 0.15936908

k means clustering

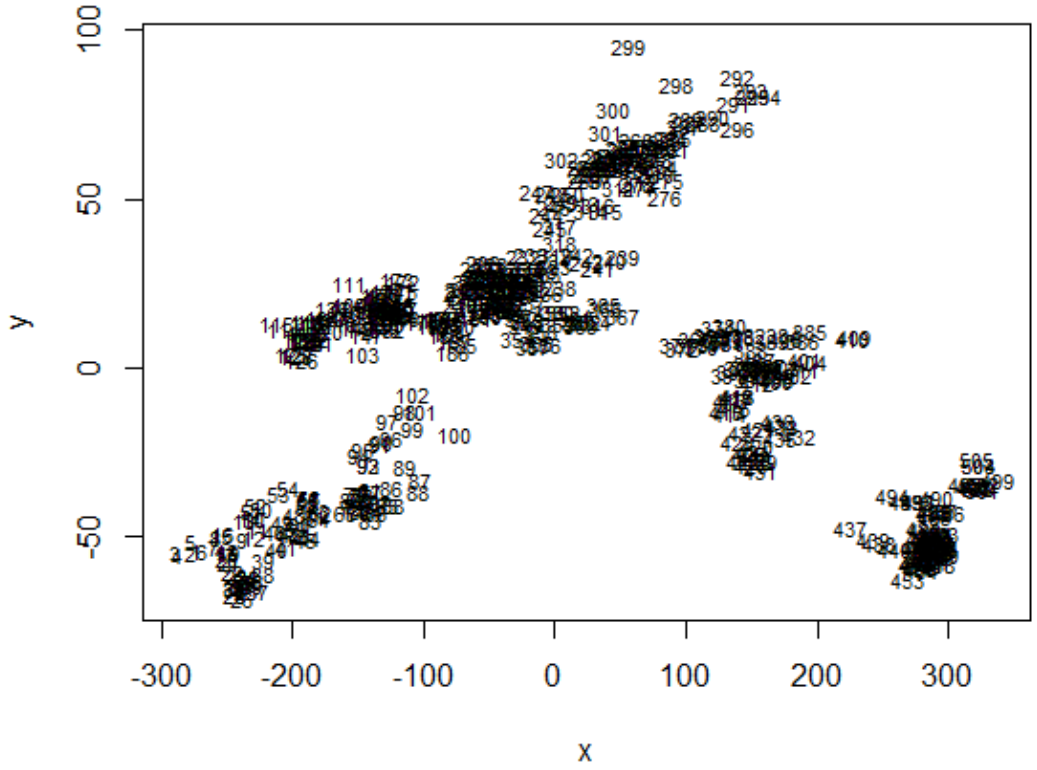
- Features obtained previously are clustered using k means clustering.
- From output, it can be analyzed which stocks fall in same cluster.
- Stocks within same cluster are highly correlated to each other.

```
K-means clustering with 5 clusters of sizes 2, 8, 4, 2, 9

Cluster means:
         50
1 2.2886871
2 0.2244583
3 0.4126405
4 0.6768617
5 0.1170380

Clustering vector:
American Airlines             Delta         Jet Blue         Southwest   Spirit Airlines   Bank of America       Capital One
                5                 5                5                 2                 3                 5                 2
    Goldman Sachs    Morgan Stanley       Wells Fargo            Aetna AmerisourceBergen       CVS Health Johnson & Johnson
                4                 5                5                 3                 2                 2                 2
         Mckesson           Amazon           Costco            Target         Walgreens           Walmart             Apple
                4                 1                2                 2                 5                 2                 3
            Google              IBM            Intel         Microsoft
                1                 3                5                 5
```

- Above results show the 5 clusters that have been formed and which stocks belong to the same cluster.
- Next, its needs to be checked whether this kind of clustering is reliable.
- This can be inferred by looking at ratio of within sum of square/total sum of square.

```
Within cluster sum of squares by cluster:
[1] 0.126118677 0.008052394 0.006318025 0.000157437 0.011844139
 (between_SS / total_SS =  98.2 %)
```

- Between_ss represents sum of squares between clusters and total_ss is addition of sum of squares between clusters and within clusters.
- High percentage suggests that between clusters sum of square takes large portion of total meaning that there is very little sum of squares between stocks within cluster.
- Thus result is very accurate and clustering is done neatly.
- Clusters of stock prices obtained are reliable and can be trusted upon.



- Above figure shows the clustering plot. 5 clusters obtained can be visualized from the figure.
- Figure gives indication that clustering is done fairly neatly.

```
> table(colnames(f1), cluster$cluster)

                    1 2 3 4 5
  Aetna             0 0 1 0 0
  Amazon            0 0 0 1 0
  American Airlines 1 0 0 0 0
  AmerisourceBergen 0 0 0 0 1
  Apple             0 0 1 0 0
  Bank of America   1 0 0 0 0
  Capital One       0 0 0 0 1
  Costco            0 0 0 0 1
  CVS Health        0 0 0 0 1
  Delta             1 0 0 0 0
  Goldman Sachs     0 1 0 0 0
  Google            0 0 0 1 0
  IBM               0 0 1 0 0
  Intel             1 0 0 0 0
  Jet Blue          1 0 0 0 0
  Johnson & Johnson 0 0 0 0 1
  Mckesson          0 1 0 0 0
  Microsoft         1 0 0 0 0
  Morgan Stanley    1 0 0 0 0
  Southwest         0 0 0 0 1
  Spirit Airlines   0 0 1 0 0
  Target            0 0 0 0 1
  Walgreens         1 0 0 0 0
  Walmart           0 0 0 0 1
  Wells Fargo       1 0 0 0 0
```

- Table to easily visualize which stock belongs to which cluster.

Clustering by Industry???

- One common thought which comes to mind is that usually stocks of an industry are highly correlated depending on demand and functioning of industry.
- Thus they should come under the same cluster.
- However much research is done in this area with most people refuting this claim.
- For instance, a research uses Pearson coefficient to prove that stocks of same industry do not always come under one cluster.
- There have always been doubts whether Pearson coefficient is an appropriate method.
- But with this method having results with 98.2 % accuracy and stocks of same industry not necessarily coming under same cluster, it can be proved.
- The method also seems perfectly appropriate.

References

- Rani, S., & Sikka, G. (n.d.). *Recent Techniques of Clustering of Time Series Data.*
- Toshniwal, D., & Joshi, R. (n.d.). *Using Cumulative Weighted Slopes for Clustering Time Series Data.*
- https://finance.yahoo.com
- www.wikipedia.com

Appendix

R Programming Code:

```
library(readr)

# Airline Industry

American_Airlines <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Airline/American Airlines.csv")

Delta <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Airline/Delta.csv")

Jet_Blue <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Airline/Jet Blue.csv")

Southwest <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Airline/Southwest.csv")

Spirit_Airlines <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Airline/Spirit Airlines.csv")


# Banking Industry

Bank_of_America <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Banking/Bank of America.csv")

Capital_One <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Banking/Capital One.csv")

Goldman_Sachs <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Banking/Goldman Sachs.csv")

Morgan_Stanley <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Banking/Morgan Stanley.csv")

Wells_Fargo <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Banking/Wells Fargo.csv")

# Healthcare Industry

Aetna <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Healthcare/Aetna.csv")
```

```
AmerisourceBergen <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Healthcare/AmerisourceBergen.csv")

CVS_Health <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Healthcare/CVS Health.csv")

Johnson_Johnson <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Healthcare/Johnson & Johnson.csv")

Mckesson <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Healthcare/Mckesson.csv")


# Retail Industry

Amazon <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Retail/Amazon.csv")

Costco <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Retail/Costco.csv")

Target <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Retail/Target.csv")

Walgreens <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Retail/Walgreens.csv")

Walmart <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Retail/Walmart.csv")


# Technology Industry

Apple <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Technology/Apple.csv")

Google <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Technology/Google.csv")

IBM <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Technology/IBM.csv")

Intel <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Technology/Intel.csv")

Microsoft <- read_csv("C:/Users/Sagar Ghiya/Desktop/Study/SEM 2/Applied Time Series and Spatial Statistics/Stock Price Data/Technology/Microsoft.csv")
```

```
# Data Shaping

# Converting in form of data frames

df1 <- data.frame(American_Airlines[,c(1,7)], Delta[,7], Jet_Blue[,7], Southwest[,7], Spirit_Airlines[,7])

colnames(df1) <- c("Date", "American Airlines", "Delta", "Jet Blue", "Southwest", "Spirit Airlines")

df2 <- data.frame(Bank_of_America[,7], Capital_One[,7], Goldman_Sachs[,7], Morgan_Stanley[,7], Wells_Fargo[,7])

colnames(df2) <- c("Bank of America", "Capital One", "Goldman Sachs", "Morgan Stanley", "Wells Fargo")

df3 <- data.frame(Aetna[,7], AmerisourceBergen[,7], CVS_Health[,7], Johnson_Johnson[,7], Mckesson[,7])

colnames(df3) <- c("Aetna", "AmerisourceBergen", "CVS Health", "Johnson & Johnson", "Mckesson")

df4 <- data.frame(Amazon[,7], Costco[,7], Target[,7], Walgreens[,7], Walmart[,7])

colnames(df4) <- c("Amazon", "Costco", "Target", "Walgreens", "Walmart")

df5 <- data.frame(Apple[,7], Google[,7], IBM[,7], Intel[,7], Microsoft[,7])

colnames(df5) <- c("Apple", "Google", "IBM", "Intel", "Microsoft")

final_df <- cbind.data.frame(df1, df2, df3, df4, df5)


#set1(1:50)

z1 <- ((final_df[50,2:26] - final_df[1,2:26])/(as.numeric(as.Date(final_df[50,1]) - as.Date(final_df[1,1]))))^2

t1 <- z1*(1/10)


# Set2(51:100)

z2 <- ((final_df[100,2:26] - final_df[51,2:26])/(as.numeric(as.Date(final_df[100,1]) - as.Date(final_df[51,1]))))^2

t2 <- z2*(2/10)
```

```
# Set3(101:150)

z3 <- ((final_df[150,2:26] - final_df[101,2:26])/(as.numeric(as.Date(final_df[150,1]) - as.Date(final_df[101,1]))))^2

t3 <- z3*(3/10)


# Set4(151:200)

z4 <- ((final_df[200,2:26] - final_df[151,2:26])/(as.numeric(as.Date(final_df[200,1]) - as.Date(final_df[151,1]))))^2

t4 <- z4*(4/10)


# Set5(201:250)

z5 <- ((final_df[250,2:26] - final_df[201,2:26])/(as.numeric(as.Date(final_df[250,1]) - as.Date(final_df[201,1]))))^2

t5 <- z5*(5/10)


# Set6(251:300)

z6 <- ((final_df[300,2:26] - final_df[251,2:26])/(as.numeric(as.Date(final_df[300,1]) - as.Date(final_df[251,1]))))^2

t6 <- z6*(6/10)


# Set7(301:350)

z7 <- ((final_df[350,2:26] - final_df[301,2:26])/(as.numeric(as.Date(final_df[350,1]) - as.Date(final_df[301,1]))))^2

t7 <- z7*(7/10)


# Set8(351:400)

z8 <- ((final_df[400,2:26] - final_df[351,2:26])/(as.numeric(as.Date(final_df[400,1]) - as.Date(final_df[351,1]))))^2

t8 <- z8*(8/10)
```

```
# Set9(401:450)
z9 <- ((final_df[450,2:26] - final_df[401,2:26])/(as.numeric(as.Date(final_df[450,1]) - as.Date(final_df[401,1]))))^2
t9 <- z9*(9/10)

# Set10(451:500)
z10 <- ((final_df[500,2:26] - final_df[451,2:26])/(as.numeric(as.Date(final_df[500,1]) - as.Date(final_df[451,1]))))^2
t10 <- z10*(10/10)

# Calculating Cumulative Weighted Slopes
summation <- t1 + t2 + t3 + t4 + t5 + t6 + t7 + t8 + t9 + t10
answer <- sqrt(summation)

# k means clustering
cluster <- kmeans(t(answer),5)
cluster

# Plotting
# n dimensional to 2d
library(cluster)
library(fpc)
d <- dist(final_df[2:26])
fit <- cmdscale(d, eig=TRUE,2)
x <- fit$points[,1]
y <- fit$points[,2]

plot(x, y, type='n', col=cluster$cluster )
text(x, y, labels = row.names(final_df), cex=.7)
```