Stat 218 - Analytics Project IV

Inigo Benavides and Rommel Bartolome

April 12, 2019

Abstract

In this project, we utilized three techniques for analysis involving two datasets. First, Principal Components Analysis was implemented to a dataset consisting of 500 universities in the United States. Here, we found that at up to `7` components, we are able to achieve around 75% of the cumulative proportion of the variance. Second, we used cluster analysis on the same dataset, using both hierarchical and non-hierarchical methods. We found that the non-hierarchical k-means clustering was the best, producing `8` clusters. Lastly, we used associate rule mining to a dataset containing transactions in the form of financial products. The most interesting association we found is that if you have savings in a bank, you are likely to have a philhealth too. This is actionable by the government wherein they can bundle a savings account as one of our mandatory benefits.

Introduction

For our fourth analytics project, we will be utilizing three techniques using two datasets. The three techniques are Principal Components Analysis (PCA), Clustering, and Associate Rule Mining (ARM).

The first dataset consists of 500 universities in the United States. The variables are:

- ADM_RATE: admission rate defined as number of admitted undergraduates divided by the number of undergraduates who applied
- SATVR25: 25th percentile of the scores in the reading section of the SAT (Scholastic Assessment Test)
- SATVR75: 75th percentile of the scores in the reading section of the SAT (Scholastic Assessment Test)
- SATMT25: 25th percentile of the scores in the math section of the SAT (Scholastic Assessment Test)
- SATMT75: 75th percentile of the scores in the math section of the SAT (Scholastic Assessment Test)
- SATWR25: 25th percentile of the scores in the writing section of the SAT (Scholastic Assessment Test)
- SATWR75: 75th percentile of the scores in the writing section of the SAT (Scholastic Assessment Test)
- PCIP11: percent of degrees awarded in category 11* programs (Computer Science)
- PCIP13: percent of degrees awarded in category 13 programs (Education)
- PCIP24: percent of degrees awarded in category 24 programs (Liberal Arts)
- PCIP38: percent of degrees awarded in category 38 programs (Philosophy)
- PCIP50: percent of degrees awarded in category 50 programs (Visual and Performing Arts)
- PCIP51: percent of degrees awarded in category 51 programs (Health)
- PCIP52: percent of degrees awarded in category 52 programs (Business)
- UGDS: number of degree-seeking undergraduates enrolled
- TUITFTE: net tuition revenue per full-time equivalent student (in USD)
- PFTFAC: proportion of full-time faculty
- RET_FT4: proportion of full-time students who return to the institution after the first year
- PCTFLOAN: proportion of students who received loans
- UG25abv: proportion of students who are ages 25 to 64
- RPY_5YR_RT: proportion of students who are making progress (have not defaulted) in paying their loans five years after graduation

These variables will be used in PCA to extract principal components that can explain a substantial amount of variation in the data. We will aim for the cumulative proportion of variance explained to be at least 75%. Aside from PCA, the same dataset will be used to form clusters and we will be using both hierarchical and non-hierarchical methods in order to find the best balanced clusters.

The second data dataset consists of 350 "transactions" in the form of financial products that a household currently has. A household is considered to have the financial product if the respondent has it personally or

jointly with other household members. The products are `Savings account in bank`, `Savings account in other institution`, `Current account`, `Microfinance loan`, `Business loan`, `Mortgage loan`, `Life insurance`, `Health insurance`, `Philhealth`, `Non-life insurance`, `Mutual funds`, `Stock market shares`, `SSS/GSIS pension`, `Pension fund`, `Credit card`, `Prepaid/debit card`, and `Business franchise`.

For this dataset, we will form association rules to see which products tend to be commonly availed by households.

Data Loading and Cleaning

We load all the libraries we will be using in this project. In addition, similar to our previous projects, we will also clean our data and set our seed for reproducibility.

For the first dataset, we will rename some columns to match the titles above. Fortunately, for the second dataset, we won't need to clean it as it can be already easily analyzed in its given form.

```
library(tidyverse)
library(DT)
library(GGally)
library(cluster)
library(arules)
library(arulesViz)
library(png)
seed <- 99
data1 <- read_csv("data1_BaBe.csv") %>%
  rename(PCIP_COMPSCI=PCIP11,
         PCIP_EDUC=PCIP13,
         PCIP_LIBERAL_ARTS=PCIP24,
         PCIP_PHILO=PCIP38,
         PCIP_VIS_PERF_ARTS=PCIP50,
         PCIP_HEALTH=PCIP51,
         PCIP_BUSINESS=PCIP52)
```

Principal Components Analysis

Below we run an initial PCA fit on the full data set and display the summary. We find that up to 7 components, we achieve around 75% of the cumulative proportion of variance explained:

```
data1_pca <- prcomp(data1, scale = TRUE, center = TRUE)
data1_pca %>% summary
```

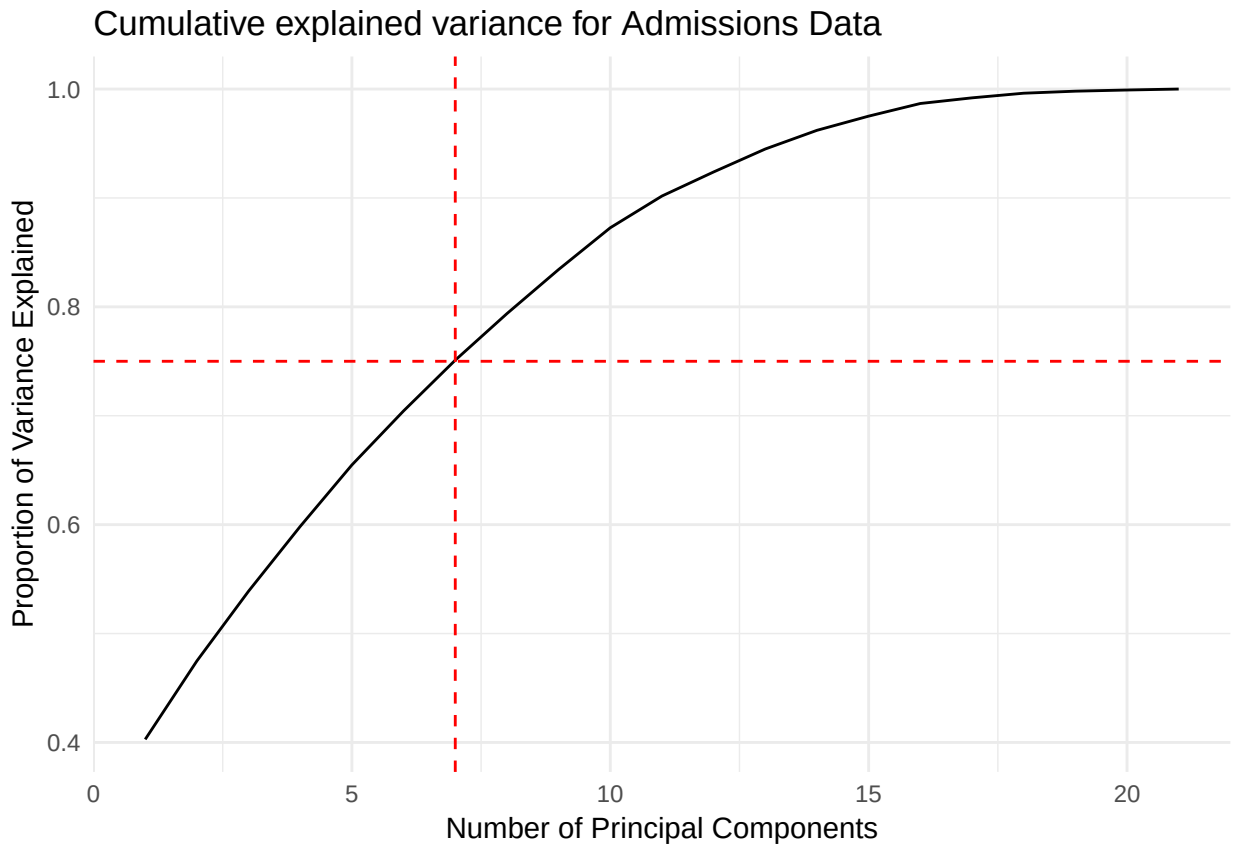
```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.9084 1.23012 1.15961 1.11771 1.08883 1.02076
## Proportion of Variance 0.4028 0.07206 0.06403 0.05949 0.05645 0.04962
## Cumulative Proportion  0.4028 0.47485 0.53888 0.59837 0.65482 0.70444
##                           PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.98917 0.94730 0.92212 0.89745 0.78225 0.68152
## Proportion of Variance 0.04659 0.04273 0.04049 0.03835 0.02914 0.02212
## Cumulative Proportion  0.75103 0.79377 0.83426 0.87261 0.90175 0.92387
##                           PC13    PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.66547 0.59992 0.52279 0.49400 0.32993 0.29901
## Proportion of Variance 0.02109 0.01714 0.01301 0.01162 0.00518 0.00426
## Cumulative Proportion  0.94495 0.96209 0.97511 0.98673 0.99191 0.99617
```

```
##                            PC19    PC20    PC21
## Standard deviation     0.19789 0.15044 0.13657
## Proportion of Variance 0.00186 0.00108 0.00089
## Cumulative Proportion  0.99803 0.99911 1.00000
```

Selection of PCs to be included

We plot cumulative variance of proportion explained:

```
data.frame(
  cev=summary(data1_pca)$importance[3,]
  ) %>% ggplot(aes(x=1:21, y=cev)) +
  geom_line() +
  theme_minimal() +
  labs(title="Cumulative explained variance for Admissions Data") +
  xlab("Number of Principal Components") +
  ylab("Proportion of Variance Explained") +
  geom_hline(aes(yintercept=0.75), color='red', linetype='dashed') +
  geom_vline(aes(xintercept=7), color='red', linetype='dashed')
```



Based on the above analysis, we choose to fit a PCA model using 7 components since it accounts for 75% of the variation.

Description of Final PCs

We display the values of the final PCs:

```
data1_pca_final <- prcomp(data1, scale = TRUE, center = TRUE, rank=7)
data1_pca_final_rotations <- summary(data1_pca_final)$rotation
data1_pca_final_rotations
```

```
##                              PC1          PC2          PC3         PC4
## ADM_RATE            0.16091240 -0.023546421 -0.262280858  0.38799985
## SATVR25            -0.32889710  0.006339800  0.027357565 -0.05054177
## SATVR75            -0.32215951 -0.010073636  0.025515550 -0.03977562
## SATMT25            -0.33067758  0.080381150  0.021413550  0.03703146
## SATMT75            -0.32932089  0.047057910  0.023614242  0.05965200
## SATWR25            -0.33237870 -0.002893333  0.055631958 -0.02934275
## SATWR75            -0.33086056 -0.021346451  0.039478418 -0.02800768
## PCIP_COMPSCI       -0.03263385  0.138613572  0.006778196  0.35335207
## PCIP_EDUC           0.13479715  0.106401098 -0.355231101 -0.12445391
## PCIP_LIBERAL_ARTS   0.08784239  0.143958511  0.375376468 -0.30391777
## PCIP_PHILO         -0.04718315  0.075231066 -0.274055475 -0.35082254
## PCIP_VIS_PERF_ARTS -0.04200767 -0.536848029  0.036582849 -0.27672736
## PCIP_HEALTH         0.08806364 -0.031421366  0.241299909  0.37619690
## PCIP_BUSINESS       0.10510105  0.157404663  0.028076427  0.24085436
## UGDS               -0.12028557  0.455106878  0.126219069  0.07001706
## TUITFTE            -0.19374379 -0.441064714  0.087825975  0.19308609
## PFTFAC             -0.05641591  0.311382905 -0.436402992 -0.05676838
## RET_FT4            -0.28976528 -0.004078979 -0.009946686  0.17782053
## PCTFLOAN            0.21217645 -0.323856080 -0.223210101  0.17137361
## UG25abv             0.18758150  0.105263959  0.457606168  0.09675234
## RPY_5YR_RT         -0.23516263 -0.068386340 -0.219365245  0.30180324
##                             PC5          PC6          PC7
## ADM_RATE           -0.0664985008  0.028909400 -0.158437103
## SATVR25             0.0013636875 -0.065508509 -0.016334047
## SATVR75            -0.0284229508 -0.043420246 -0.055019879
## SATMT25             0.0120278661 -0.014783250  0.024184776
## SATMT75            -0.0093930354 -0.006800372  0.005564817
## SATWR25             0.0100954608 -0.077681293  0.002397794
## SATWR75            -0.0300828058 -0.068889948  0.006530618
## PCIP_COMPSCI        0.2331587478  0.337193312 -0.716060935
## PCIP_EDUC           0.0834338309 -0.082490632  0.148955565
## PCIP_LIBERAL_ARTS  -0.0588307254 -0.161677385 -0.234116015
## PCIP_PHILO         -0.1895298949 -0.418097829 -0.467833578
## PCIP_VIS_PERF_ARTS  0.0566625795  0.450674308 -0.007015668
## PCIP_HEALTH        -0.6662526606 -0.104128925  0.079369283
## PCIP_BUSINESS       0.5809687388 -0.404270464  0.215296342
## UGDS               -0.0009414934  0.414882195  0.259914975
## TUITFTE             0.0647160472 -0.244983967  0.080244277
## PFTFAC             -0.2999840812  0.079041943  0.137152382
## RET_FT4            -0.0301640783 -0.014924237  0.054236981
## PCTFLOAN           -0.0776040021 -0.021471441  0.034879876
## UG25abv            -0.0788975333 -0.167320228 -0.080992979
## RPY_5YR_RT         -0.0292760220 -0.126635556 -0.078133148
```

```
### Use this code snippet below in HTML format for easier visualization
#  %>% formatStyle(
#    columns=colnames(data1_pca_final_rotations),
#    color=styleInterval(c(-1, -0.3, 0, 0.3, 1),
#     c('red', 'orange', 'white', 'grey','orange', 'red'))
```

```
#     )
```

Based on the above results, we explain each component:

1. PC1: Academic Achievement

The first component has high loadings for features relating to SAT scores, with each around -0.33. We can classify this component as reflecting *academic achievement*.

2. PC2: University Size and Financial Capability

The second component has high loadings for `UGDS` (+), `TUITFTE` (-), `PFTFAC` (+), and `PCTFLOAN` (-), corresponding to the number of undergraduate students enrolled, tuition fee per student, percent of full time faculty, and percent of students with loans, respectively. We can term this component as reflecting *university size and financial capability*. Interestingly, `PCIP_VIS_PERF_ARTS` has high negative loadings, perhaps due to such performing arts schools known for scholarships.

3. PC3: Student Demographics

The third component has high loadings for `PCIP_EDUC` (-), `PCIP_LIBERAL_ARTS` (+), `PFTFAC` (-), and `UG25abv` (+). We can term this component *student demographics*, since the strongest descriptor for this component comes from the proportion of students aged 25 to 64. It seems that universities with older undergraduate students typically have less full time faculty and have more liberal arts graduates.

4. PC4: Science vs. Liberal Arts

The fourth component has high loadings for `ADM_RATE` (+), `PCIP_COMPSCI` (+), `PCIP_LIBERAL_ARTS` (-), `PCIP_PHILO` (-), `PCIP_HEALTH` (+), and `RPY_5YR_RT` (+). We can term this as a sort of *science vs. liberal arts* feature, possibly reflecting the increase in admissions and demand for computer science and medicine degrees over liberal arts and philosophy degrees. Interestingly, the positive inclusion of `RPY_5YR_RT` may indicate that those in science oriented degrees are better able to pay back loans, arguably because the demand for these is higher and have better prospects in industry.

5. PC5: Medicine vs. Business Schools

The fifth component has `PCIP_HEALTH` (-) and `PCIP_BUSINESS` (+). We can term this component as *medicine vs. business schools*.

6. PC6: Other Degree Differences

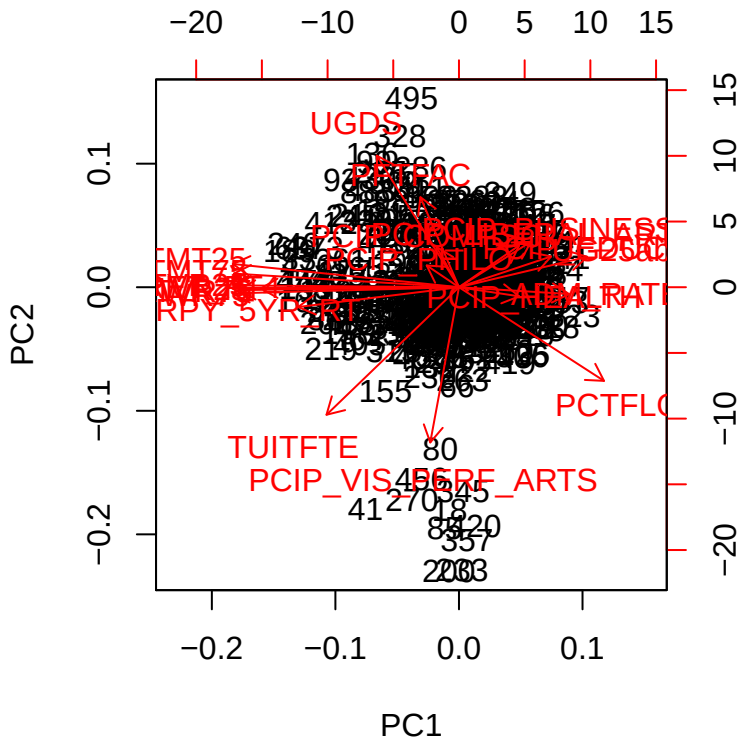
The sixth component has `PCIP_COMPSCI` (+), `PCIP_PHILO` (-), `PCIP_VIS_PERF_ARTS` (+), `PCIP_BUSINESS` (-), and `UGDS` (+). Here we see a similar variation on the contrasts between universities that are known for certain fields – in this case, we see computer science and visual performing arts contrasting with philosophy and business oriented schools; the former appear to be more correlated with larger student populations. We feel that this component tells the same stories as `PC4` and `PC5`.

7. PC7: CompSci and Philo

The seventh component only has `PCIP_COMPSCI` (-) and `PCIP_PHILO` (-), though we feel that this is more or less the same story for `PC4`.

Below we attempt to visualize `PC1` and `PC2` as a bi-plot:

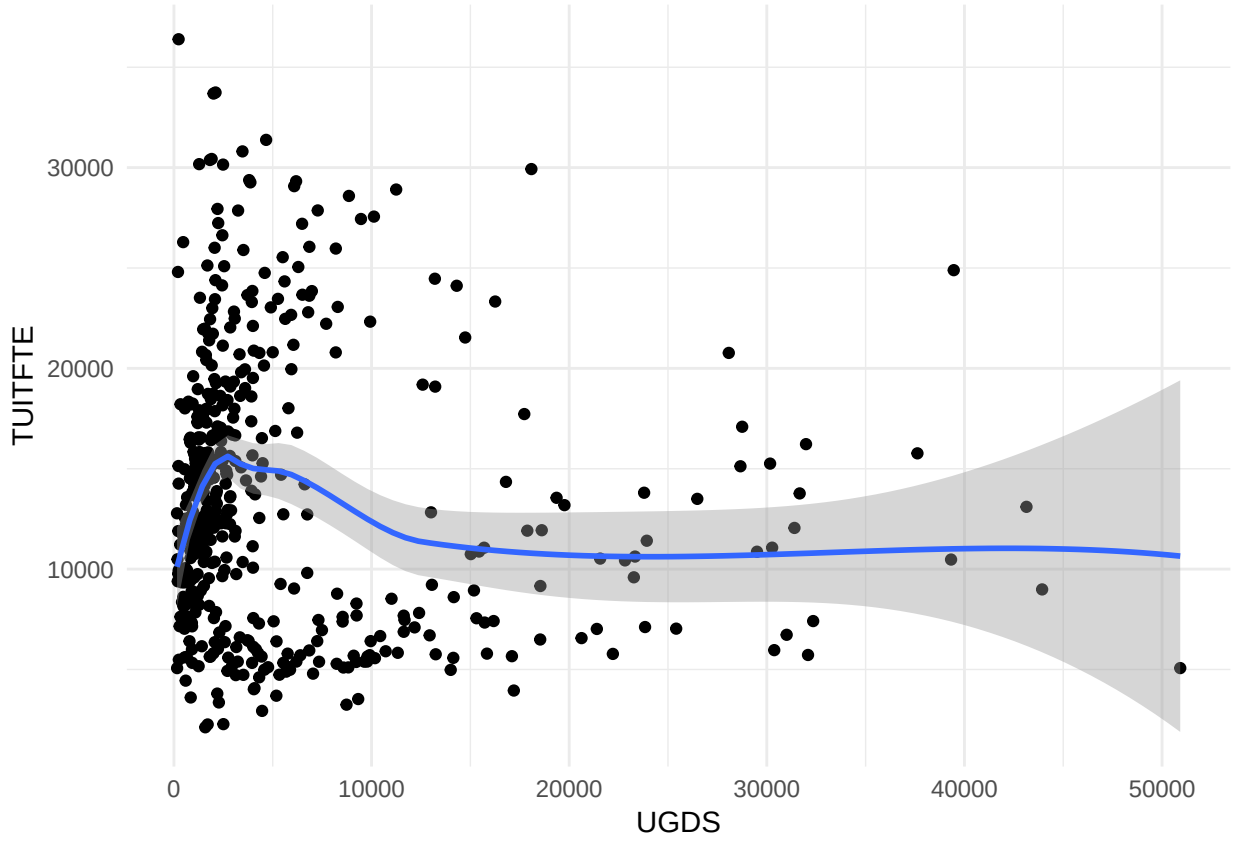
```
data1_pca_final %>% biplot
```



To further understand PC2, we plot UGDS against TUTFTE:

```
data1 %>% ggplot(aes(x=UGDS, y=TUITFTE)) + geom_point() + geom_smooth() + theme_minimal()
```

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



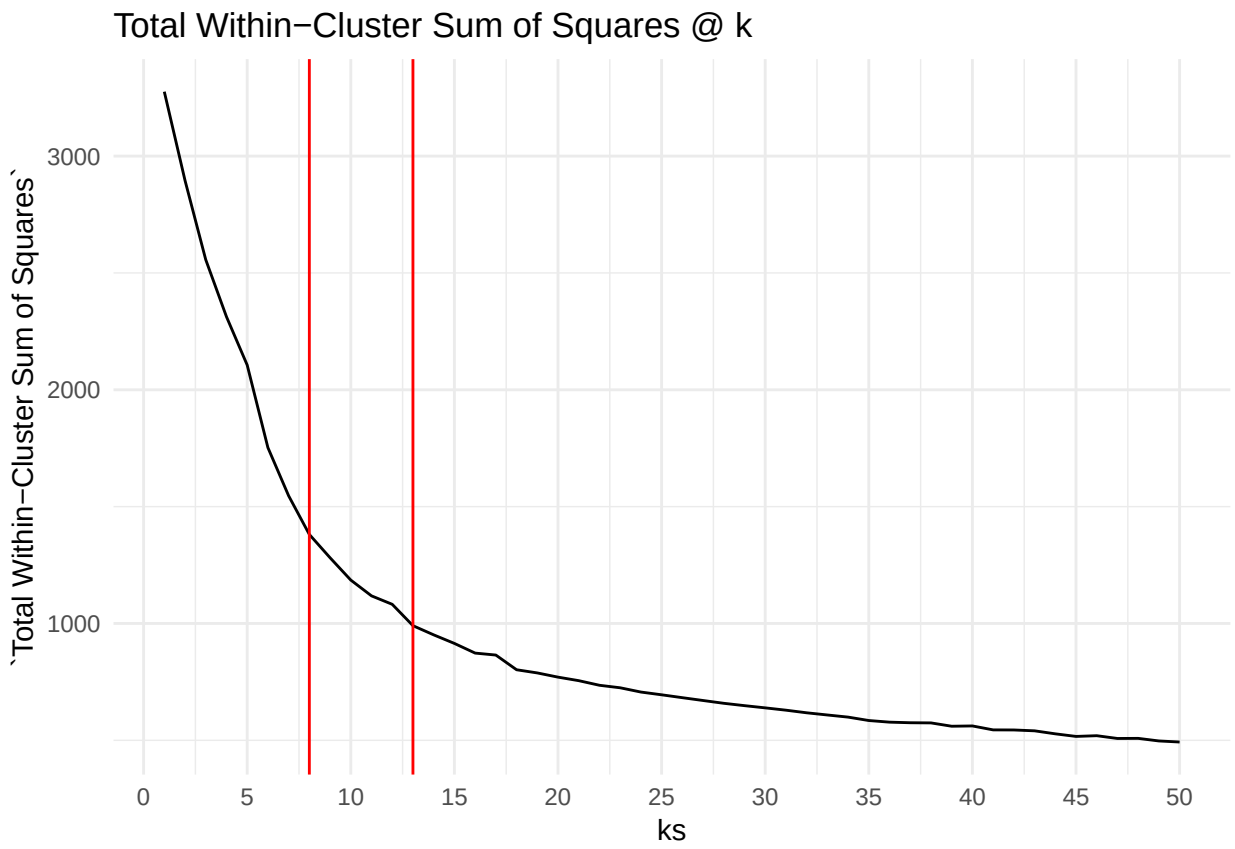
Interestingly, if we plot `UGDS` vs. `TUITFTE` to validate PC2, we find that there does seem to be a split between expensive schools with fewer students (likely private universities), and cheaper schools with many students (likely public universities).

Cluster Analysis

Choosing the Best Clustering Model: K-Means vs Hierarchical Clustering

First, we fit a K-Means clustering model on the data set:

```
set.seed(seed)
ks <- 1:50
ks %>% Map(function(x) {
  kms <- data1 %>% scale(F, T)
  km <- kmeans(kms, x, nstart=20)
  tot_within_ss <- km$tot.withinss
}, .) %>% unlist %>%
  as.data.frame() %>%
  rename("Total Within-Cluster Sum of Squares"=1) %>%
  ggplot(aes(x=ks, y=`Total Within-Cluster Sum of Squares`)) +
  geom_line() + labs(title="Total Within-Cluster Sum of Squares @ k") + theme_minimal() +
  scale_x_continuous(breaks=seq(0, 50, 5)) + geom_vline(aes(xintercept=8), color='red') +
  geom_vline(aes(xintercept=13), color='red')
```



By inspection, it seems that we find an elbow point between k = 8 and k = 13.

We check now the distribution of the cluster members using different cluster numbers:

```
sd.data <- scale(data1, F, T)
for (i in 8:13){
  set.seed(seed)
  km.out_n <- kmeans(sd.data, i)
  table(km.out_n$cluster) %>% print()
}
```

```
##
##   1   2   3   4   5   6   7   8
##  35  76  54 106  12 149  11  57
##
##   1   2   3   4   5   6   7   8   9
##  34  71  51 120  12 148  11  52   1
##
##   1   2   3   4   5   6   7   8   9  10
##  15  36  22 119  12 124  11  48   1 112
##
##   1   2   3   4   5   6   7   8   9  10  11
##  44  37  75 117  12 130  11  52   5  16   1
##
##   1   2   3   4   5   6   7   8   9  10  11  12
##  45  35  76 115   1 128  11  52   5  16   1  15
##
```

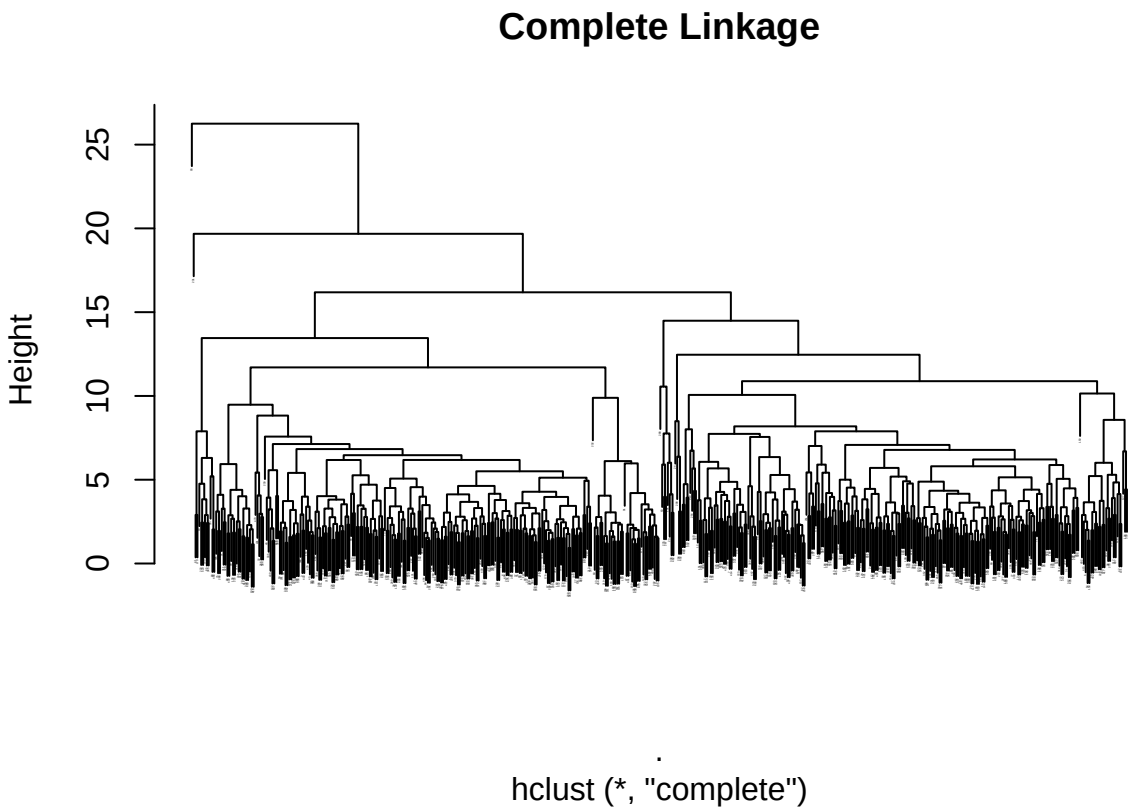
```
##   1   2   3   4   5   6   7   8   9  10  11  12  13
##  44  23  86  98   1 105  11  45   5  53   1  14  14
```

Checking the distribution, we have an extremely small cluster with only 1 member when k is equal to 9 to 11. As we would want to keep the clusters balanced as possible, we will choose k = 8.

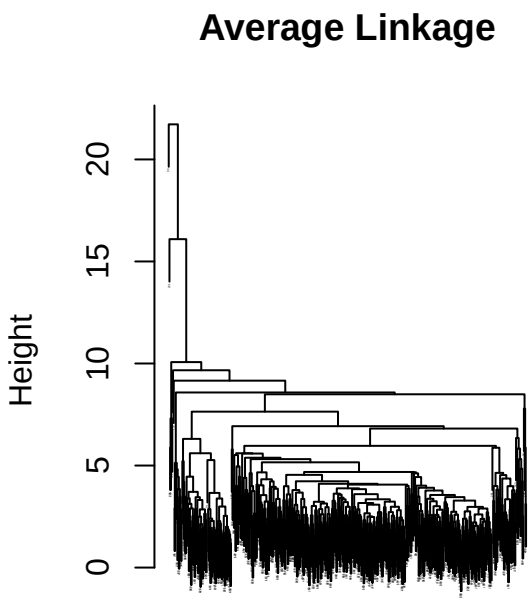
Next, we perform hierarchical clustering with complete, average, and single linkages:

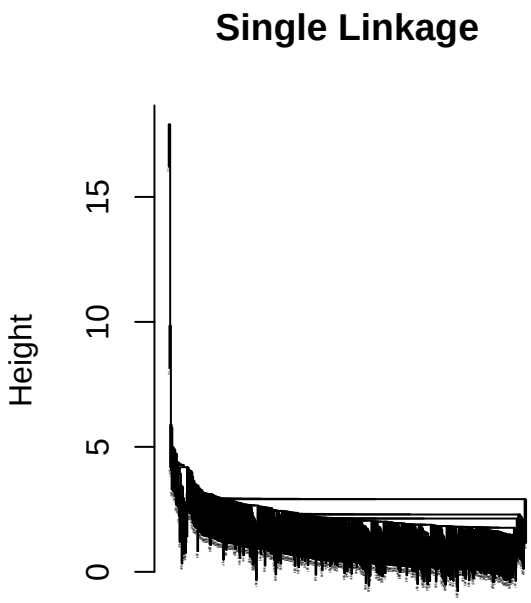
```
hc_complete <- data1 %>% scale %>% dist %>% hclust('complete')
hc_average <- data1 %>% scale %>% dist %>% hclust('average')
hc_single <- data1 %>% scale %>% dist %>% hclust('single')

hc_complete %>% plot(main = 'Complete Linkage', cex = .1)
```




```
par(mfrow = c(1,2))
hc_average %>% plot(main = 'Average Linkage', cex = .1)
hc_single %>% plot(main = 'Single Linkage', cex = .1)
```





It is clear that the complete linkage provides the most balanced segment, with the average and the single linkage having extremely one-sided initial clustering. Now, we want to cut the "tree" to a balanced segment. We check again the cluster member distribution using different k:

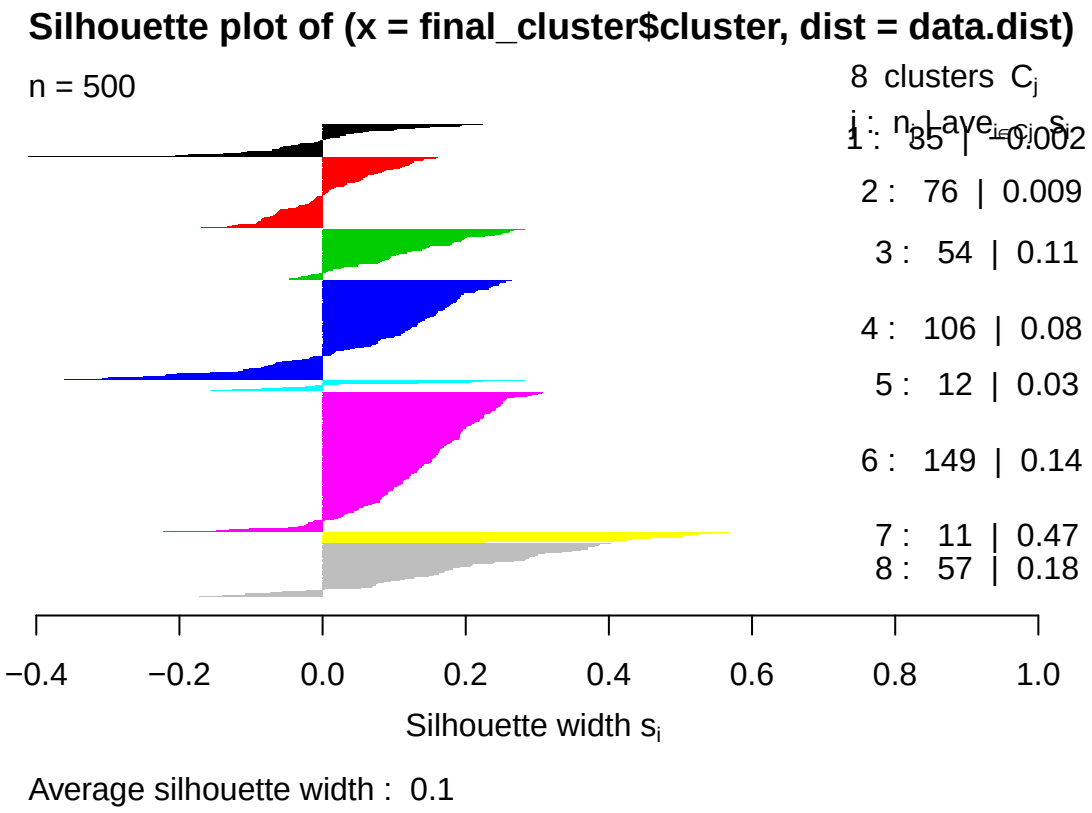
```
for (i in 8:13){
  set.seed(seed)
  hc.out <- data1 %>% scale %>% dist %>% hclust('complete')
  hc.clusters <- cutree(hc.out, i)
  table(hc.clusters) %>% print()
}
```

```
## hc.clusters
##   1   2   3   4   5   6   7   8
## 201 238  36  11   8   4   1   1
## hc.clusters
##   1   2   3   4   5   6   7   8   9
## 201 212  26  36  11   8   4   1   1
## hc.clusters
##   1   2   3   4   5   6   7   8   9  10
## 201 212  26  36  11   7   4   1   1   1
## hc.clusters
##   1   2   3   4   5   6   7   8   9  10  11
## 201 212  25  36  11   7   4   1   1   1   1
## hc.clusters
##   1   2   3   4   5   6   7   8   9  10  11  12
## 201 204  25  36  11   7   4   1   8   1   1   1
## hc.clusters
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13
## 201 204  25  35  11   7   4   1   8   1   1   1   1
```

Here, we see that the its even worse than k-means clustering! Most of the members are just in the first two clusters. As such, we would choose k-means clustering with k = 8. We check the silhoutte value of the said clustering:

```
set.seed(seed)
final_cluster <- kmeans(sd.data, 8)
data.dist <- data1 %>% scale %>% dist
plot(silhouette(final_cluster$cluster, data.dist), col=1:8, border = NA)
```



Here, we can see that most of the clusters are clustered correctly, except for cluster 2 which has quite a small value, bordering to almost zero. However, the average silhoutte width is 0.10, which is still on the positive side which means the similarity of our objects its own cluster (cohesion) is stronger compared to other clusters (separation).

Descriptive Analysis of the Formed Clusters

To allow for more flexibility, we exported the values of the clusters in Excel, so we can easily pinpoint visually the differences of the clusters:

```
pp <- readPNG("clusters2.png")
plot.new()
rasterImage(pp,0,0,1,1)
```

	1	2	3	4	5	6	7	8
ADM_RATE	0.3152	0.1509	-0.1167	-0.3278	0.2861	0.3939	-0.3278	-1.8458
SATVR25	-0.6611	0.4317	-1.1016	0.7666	0.4783	-0.4477	0.1068	2.2451
SATVR75	-0.6624	0.4918	-1.1903	0.7173	0.5323	-0.3999	0.3594	1.9884
SATMT25	-0.6500	0.3885	-1.1119	1.0653	0.7797	-0.4488	-0.2392	2.1504
SATMT75	-0.6169	0.4185	-1.2018	1.0577	0.8647	-0.4229	-0.0139	2.0060
SATWR25	-0.6378	0.4261	-1.0650	0.8139	0.3914	-0.4831	0.0761	2.3107
SATWR75	-0.5972	0.4666	-1.1773	0.7786	0.4043	-0.4376	0.2070	2.1475
PCIP_COMPSCI	-0.1734	-0.1256	-0.1683	0.0684	5.4964	-0.0478	-0.3572	-0.0220
PCIP_EDUC	-0.3925	-0.2626	0.0471	-0.3713	-0.8980	0.6319	-0.8816	-0.8759
PCIP_LIBERAL ARTS	0.0269	-0.2540	1.0165	-0.2624	-0.0036	-0.0706	-0.4737	-0.3184
PCIP_PHILO	-0.1863	0.0561	-0.0853	-0.0809	-0.2119	-0.0625	-0.2288	0.5804
PCIP_VIS_PERF_ARTS	-0.3325	0.0167	-0.3267	-0.1917	-0.1534	-0.1433	6.2036	-0.0396
PCIP_HEALTH	2.5807	-0.2391	-0.1408	-0.2548	-0.4980	-0.1013	-0.6699	-0.5034
PCIP_BUSINESS	-0.3804	-0.0750	0.7029	-0.1885	-0.3895	0.1268	-1.3330	-0.5354
UGDS	-0.3832	-0.2739	-0.3890	2.7205	-0.1648	-0.1448	-0.5328	0.1458
TUITFTE	0.1396	0.4765	-0.5863	-0.3498	0.3366	-0.5151	1.3993	1.4990
PFTFAC	-0.2781	0.1755	-0.1651	0.3777	-0.0601	-0.0642	-1.1388	0.2661
RET_FT4	-0.3131	0.4778	-1.2092	1.0747	0.2762	-0.4065	0.1273	1.5658
PCTFLOAN	0.6185	0.0376	0.4133	-1.0102	-0.1265	0.2746	0.2540	-1.6430
UG25abv	1.1653	-0.5952	0.7839	-0.4878	-0.0910	0.1859	-0.3510	-0.8772
RPY_5YR_RT	-0.1238	0.6014	-1.6254	0.5191	0.5936	-0.1060	-0.2287	0.9951

Here, we can the stark difference of the clusters. The red ones mean they have the lowest mark in that particular variable, green being the highest and yellow being a placeholder to group "average ones". Clusters 1, 2, 3, 6, and 8 appears to be primarily clustered via the scores in the Scholastic Assessment Test. Clusters 4 and 6 on the other hand have distinct high marks on some criteria. Cluster 7 appears to be worst in some areas, but the best in Visual Performance Arts. Cluster 5 appears to be the most "average" one, having high marks on Computer Science.

We describe them in detail and create a persona for each cluster.

1. The Healthcare Professionals Cluster (35 Universities)

This cluster can be characterized by a high-percentage of degrees awarded to health programs. Aside from that, they are usually the students who received loans and are above 25 years old. This is quite logical, as these are likely medical students that needs financing in their studies and given the long time to finish a medical course, are usually older. Surprisingly, they have lower than average SAT scores, which are not intuitive for medical professionals.

2. The Above Average SAT Cluster (76 Universities)

This cluster have above average SAT scores. This is across the board, with the reading, math and writing section, as well as the 25th and 75th percentile of scores.

3. The Lowest SAT Cluster (54 Universities)

Objectively, this is the "worst" cluster of all. This cluster have the lowest SATs. They also have the lowest revenue per full-time equivalent student, the highest drop-out rate (proportion of full-time students who does not return to the institution after the first year) and the highest default rate among all clusters. They are mostly Liberal Arts and Business Schools.

4. The Big Universities Cluster (106 Universities)

This cluster have the highest number of degree-seeking undergraduates enrolled and the highest proportion of full-time faculty. This means that these are likely the big schools where there are lots of students and teachers.

5. The Computer Schools Cluster (12 Universities)

This cluster is characterized by having lots of degrees awarded in computer programs.

6. The Education Cluster (149 Universities)

This cluster is characterized by having lots of degrees awarded in education programs. They also have the best admission rate, which is defined to be the number of admitted undergraduates divided by the number of undergraduates who applied.

7. The Visual & Performance Arts Cluster (11 Universities)

This cluster is characterized by having lots of degrees awarded in visual and performance arts. One distinct characteristic is that they solely focus on their programs, offering very little other programs. They are also in clear contrast with Cluster 4, as they have small student and teacher population.

8. The Elite Cluster (57 Universities)

This cluster are called the "elite" ones as they have the lowest admission rate, yet most of their students have very high SATs. They have the lowest drop-out rate and the lowest default rate. Conversely, most of their students do not have loans. They are also the youngest cluster and universities get the most net tuition revenue per full-time equivalent student. They mostly award Philosophy degrees.

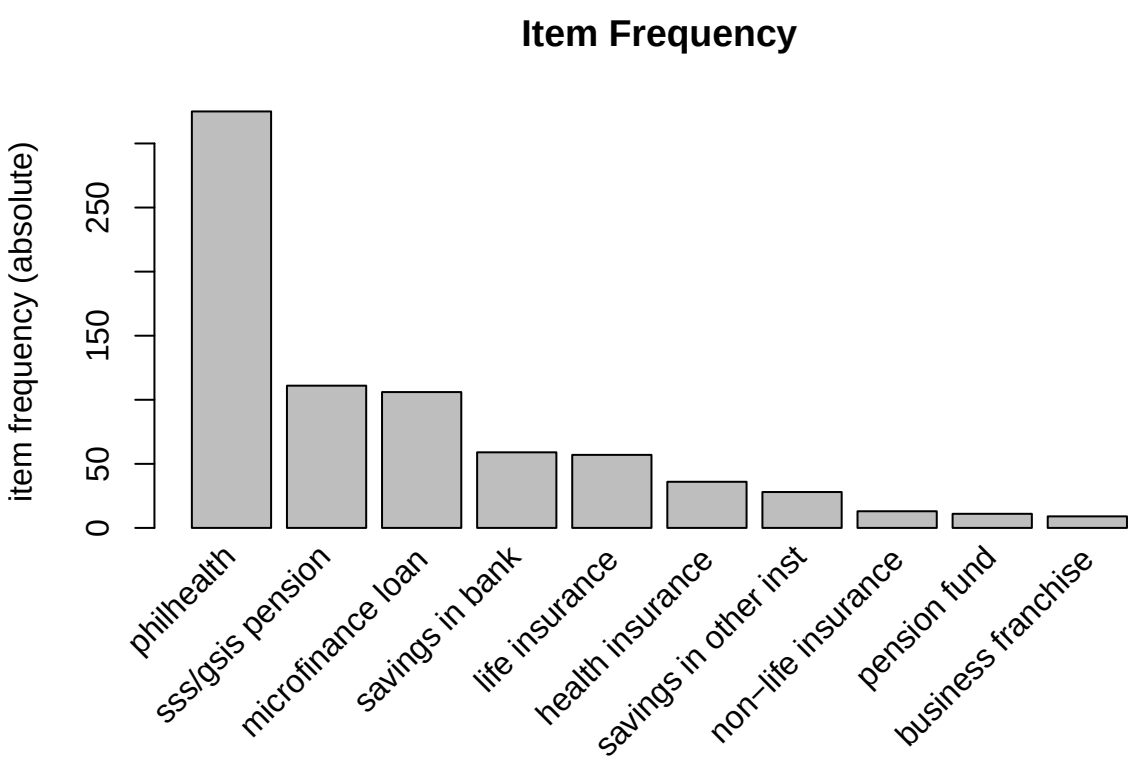
Overall, it appears that the clusters formed were unique enough and is distributed relatively well.

Association Rule Mining

Initial inspection

We first check the transactions and the item frequency:

```
fin <- read.transactions("data2_Babe.csv", format = 'basket', sep = ',')
itemFrequencyPlot(fin, topN=10, type="absolute", main="Item Frequency")
```



Here, we can see that most of the household have philhealth, with sss/gsis pension being the second one. We also check the summary of our data:

```
summary(fin)
```

```
## transactions as itemMatrix in sparse format with
##  350 rows (elements/itemsets/transactions) and
##  17 columns (items) and a density of 0.1331092
##
## most frequent items:
##        philhealth  sss/gsis pension microfinance loan   savings in bank
##               325               111               106                59
##    life insurance           (Other)
##                57               134
##
## element (itemset/transaction) length distribution:
## sizes
##   1   2   3   4   5   6   8  12  13  16
## 125 116  60  27  12   5   2   1   1   1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.263   3.000  16.000
##
## includes extended item information - examples:
##               labels
## 1 business franchise
## 2      business loan
```

```
## 3        credit card
```

Here we see that number of items, density, actual number of items and even the extended items. We shall take note of these when checking finding the rules.

Rule Mining

We set our support to 0.05 and confidence to 0.75. We now check first based on lift:

```
f <- apriori(fin, parameter = list(supp=0.05, conf=0.75, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.75    0.1    1 none FALSE            TRUE       5    0.05      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 17
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[17 item(s), 350 transaction(s)] done [0.00s].
## sorting and recoding items ... [7 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [10 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(f, by = 'lift')[1:10])
```

```
##      lhs                                     rhs          support
## [1]  {life insurance,sss/gsis pension}    => {philhealth} 0.06857143
## [2]  {sss/gsis pension}                   => {philhealth} 0.30285714
## [3]  {health insurance,sss/gsis pension}  => {philhealth} 0.05142857
## [4]  {savings in bank,sss/gsis pension}   => {philhealth} 0.10000000
## [5]  {microfinance loan,sss/gsis pension} => {philhealth} 0.07142857
## [6]  {health insurance}                   => {philhealth} 0.09428571
## [7]  {savings in bank}                    => {philhealth} 0.15428571
## [8]  {microfinance loan}                  => {philhealth} 0.26000000
## [9]  {life insurance}                     => {philhealth} 0.13714286
## [10] {savings in other inst}              => {philhealth} 0.06285714
##      confidence lift      count
## [1]  0.9600000  1.0338462  24
## [2]  0.9549550  1.0284130 106
## [3]  0.9473684  1.0202429  18
## [4]  0.9459459  1.0187110  35
## [5]  0.9259259  0.9971510  25
## [6]  0.9166667  0.9871795  33
## [7]  0.9152542  0.9856584  54
## [8]  0.8584906  0.9245283  91
## [9]  0.8421053  0.9068826  48
```

```
## [10] 0.7857143  0.8461538  22
```

Here we see that most are about philhealth. We also check based on confidence:

```
inspect(sort(f, by = 'confidence')[1:10])
```

```
##      lhs                                      rhs          support
## [1]  {life insurance,sss/gsis pension}    => {philhealth} 0.06857143
## [2]  {sss/gsis pension}                   => {philhealth} 0.30285714
## [3]  {health insurance,sss/gsis pension}  => {philhealth} 0.05142857
## [4]  {savings in bank,sss/gsis pension}   => {philhealth} 0.10000000
## [5]  {microfinance loan,sss/gsis pension} => {philhealth} 0.07142857
## [6]  {health insurance}                   => {philhealth} 0.09428571
## [7]  {savings in bank}                    => {philhealth} 0.15428571
## [8]  {microfinance loan}                  => {philhealth} 0.26000000
## [9]  {life insurance}                     => {philhealth} 0.13714286
## [10] {savings in other inst}              => {philhealth} 0.06285714
##      confidence lift      count
## [1]  0.9600000  1.0338462  24
## [2]  0.9549550  1.0284130 106
## [3]  0.9473684  1.0202429  18
## [4]  0.9459459  1.0187110  35
## [5]  0.9259259  0.9971510  25
## [6]  0.9166667  0.9871795  33
## [7]  0.9152542  0.9856584  54
## [8]  0.8584906  0.9245283  91
## [9]  0.8421053  0.9068826  48
## [10] 0.7857143  0.8461538  22
```

This is also the case. We try to limit the support to 0.15, as we would want to make rules with strong support:

```
f_30 <- apriori(fin, parameter = list(supp=0.15, conf=0.75, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.75    0.1    1 none FALSE            TRUE       5    0.15      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 52
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[17 item(s), 350 transaction(s)] done [0.00s].
## sorting and recoding items ... [5 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [3 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(sort(f_30, by = 'confidence'))
```

```
##     lhs                    rhs          support   confidence lift
## [1] {sss/gsis pension}  => {philhealth} 0.3028571 0.9549550  1.0284130
## [2] {savings in bank}   => {philhealth} 0.1542857 0.9152542  0.9856584
## [3] {microfinance loan} => {philhealth} 0.2600000 0.8584906  0.9245283
##     count
## [1] 106
## [2]  54
## [3]  91
```

For the first one, we see that if you have an sss/gsis pension, you are likely to have a philhealth too. However, this is actually usually the case for most working class Filipinos since both are government mandated. The other one tells us that if you have a microfinance loan you likely have a philhealth too. This is likely connected again to a mandated government benefit called PAG-IBIG where you can have a microfinance loan.

The most interesting rule we saw is that if you have savings in bank, you will likely have a philhealth. Having savings in a bank is not mandated by the government, unlike the components in the first two rules. As such, one actionable insight is having a government mandated savings account for all, bundled with financial products such as sss/gsis pension, philhealth and microfinance loans (PAG-IBIG).

We also tried to check non-Philhealth rules:

```
f_n <- apriori(fin, parameter = list(supp=0.001, conf=0.75, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.75    0.1    1 none FALSE            TRUE       5   0.001      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 0
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[17 item(s), 350 transaction(s)] done [0.00s].
## sorting and recoding items ... [17 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.02s].
## writing ... [419587 rule(s)] done [0.14s].
## creating S4 object  ... done [0.42s].
```

```
not_philhealth <- f_n %>% subset(!rhs %in% "philhealth") %>%
  sort(by="confidence", decreasing = F)
inspect(sort(not_philhealth, by = 'confidence')[1:5])
```

```
##     lhs                      rhs                      support confidence       lift count
## [1] {stock market shares} => {microfinance loan} 0.008571429          1   3.301887     3
## [2] {business loan}       => {microfinance loan} 0.008571429          1   3.301887     3
## [3] {current account}     => {savings in bank}   0.025714286          1   5.932203     9
## [4] {business loan,
##      stock market shares} => {mortgage loan}     0.005714286          1  87.500000     2
## [5] {mortgage loan,
##      stock market shares} => {business loan}     0.005714286          1 116.666667     2
```

Unfortunately, the support is very low. We also check for other possibly interesting rules on microfinance loans:

```
not_philhealth <- f_n %>% subset(rhs %in% "microfinance loan") %>%
  sort(by="confidence", decreasing = F)
inspect(sort(not_philhealth, by = 'confidence')[1:5])
```

```
##     lhs                       rhs                       support confidence     lift count
## [1] {stock market shares} => {microfinance loan} 0.008571429          1 3.301887     3
## [2] {business loan}       => {microfinance loan} 0.008571429          1 3.301887     3
## [3] {business loan,
##      stock market shares} => {microfinance loan} 0.005714286          1 3.301887     2
## [4] {mortgage loan,
##      stock market shares} => {microfinance loan} 0.005714286          1 3.301887     2
## [5] {business franchise,
##      stock market shares} => {microfinance loan} 0.005714286          1 3.301887     2
```

Similarly, the support is also extremely low. We also check for bank savings:

```
not_philhealth <- f_n %>% subset(rhs %in% "savings in bank") %>% sort(by="confidence", decreasing = F)
inspect(sort(not_philhealth, by = 'confidence')[1:5])
```

```
##     lhs                       rhs                     support confidence     lift count
## [1] {current account}     => {savings in bank} 0.025714286          1 5.932203     9
## [2] {business loan,
##      stock market shares} => {savings in bank} 0.005714286          1 5.932203     2
## [3] {mortgage loan,
##      stock market shares} => {savings in bank} 0.005714286          1 5.932203     2
## [4] {business franchise,
##      stock market shares} => {savings in bank} 0.005714286          1 5.932203     2
## [5] {mutual funds,
##      stock market shares} => {savings in bank} 0.005714286          1 5.932203     2
```

Here we see that if you have a current account, you will likely have savings in bank. Again, this is quite obvious as for most Filipinos, the savings account is the your "entry" product in a banking institution and the checking account is quite secondary.

Overall, we would recommended having more data with better frequency distribution so we can check for more rules with higher support.

Conclusions

1. Principal Components Analysis was utilized in analyzing university data. We found that up to 7 components, we are able to achieve around 75% of the cumulative proportion of the variance.
2. Cluster analysis was utilized on the same dataset. We found that k-means clustering was the best clustering method, producing 8 clusters.
3. Associate rule mining was utilized on a new dataset containing transactions of Filipino financial products. The most interesting association we found is that if you have savings in a bank, you are likely to have a philhealth too. One actionable insight from this is the government bundling a new mandatory benefit in the form of a savings account.