Clustering Analysis To find College Majors That Pay Back

omprasad shee

12/31/2019

Table of Contents

Introduction/Overview 1

Methods/analysis Approach 2

Data Manipulation and Exploration........ 2

Data Collection 2

METHOD1. The elbow method........ 5

METHOD2. The silhouette method........ 6

METHOD3. The gap statistic method 7

K-means Clustering Algorithm........ 8

Visualizing the clusters 8

1. Humanities-liberal cluster........ 9

2. The Challenger cluster........ 10

3. The Leader cluster........ 11

Results........ 12

Conclusion........ 12

References........ 13

Introduction/Overview

Choosing a college Major is a complex Decision.The below analysis uses k-means cluster analysis to explore the salary potential of college majors.

In this project, the dataset used is data collected from an year-long survey of 1.2 million people with only a bachelor's degree by PayScale Inc., made available at the following link:

http://online.wsj.com/public/resources/documents/info-Degrees_that_Pay_you_Back-sort.html?mod=article_inline

Methods/analysis Approach

Determining the optimal number of clusters in a data set is a fundamental issue in partitioning clustering, such as k-means clustering, which requires the user to specify the number of clusters k to be generated.

Unfortunately, there is no definitive answer to this question. The optimal number of clusters is somehow subjective and depends on the method used for measuring similarities and the parameters used for partitioning. A simple and popular solution consists of inspecting the dendrogram produced using hierarchical clustering to see if it suggests a particular number of clusters. Unfortunately, this approach is also subjective. These methods include direct methods and statistical testing methods:

Direct methods: consists of optimizing a criterion, such as the within cluster sums of squares or the average silhouette. The corresponding methods are named elbow and silhouette methods, respectively. Statistical testing methods: consists of comparing evidence against null hypothesis. An example is the gap statistic.

Post Data Manipulation and Exploration , we'll compare the recommendations from three different methods 1) *Elbow method 2)*Silhouette method*, and 3)* Gap Statistics method*, apply a* k-means clustering* analysis, and visualize the results.

Data Manipulation and Exploration

Data Collection

We need to *scrape* the data from The Wall Street Journal article at the aforementioned link.

We can scrape the *Salary Increase By Major* data from the web page using the below code:

raw_data object contains salary data scraped from the WSJ article,after cleansing the data *degrees* had 50 observations (for 50 majors).

Exploratory Analysis of *degrees* data frame and some *summary* statistics were done as below.

```
#############################################Data Loading and Preparation
#############

if(!require(rvest)) install.packages("rvest", repos = "http://cran.us.r-
project.org")

## Loading required package: rvest

## Loading required package: xml2

# SOURCE-from the wall street journal article
url <- "http://online.wsj.com/public/resources/documents/info-
Degrees_that_Pay_you_Back-sort.html?mod=article_inline#top"
```

```r
h <- read_html(url)
nodes <- h %>% html_nodes('table')

# locate table of interest from nodes "xml_nodeset"
tab <- nodes[[7]]
raw_data <- html_table(tab)
head(raw_data)
```

```
##                      X1                     X2                       X3
## 1   Undergraduate Major Starting Median Salary Mid-Career Median Salary
## 2            Accounting             $46,000.00               $77,100.00
## 3 Aerospace Engineering             $57,700.00              $101,000.00
## 4           Agriculture             $42,600.00               $71,900.00
## 5          Anthropology             $36,800.00               $61,500.00
## 6          Architecture             $41,600.00               $76,800.00
##                                                  X4
## 1 Percent change from Starting to Mid-Career Salary
## 2                                              67.6
## 3                                              75.0
## 4                                              68.8
## 5                                              67.1
## 6                                              84.6
##                                  X5                                X6
## 1 Mid-Career 10th Percentile Salary Mid-Career 25th Percentile Salary
## 2                        $42,200.00                        $56,100.00
## 3                        $64,300.00                        $82,100.00
## 4                        $36,300.00                        $52,100.00
## 5                        $33,800.00                        $45,500.00
## 6                        $50,600.00                        $62,200.00
##                                  X7                                X8
## 1 Mid-Career 75th Percentile Salary Mid-Career 90th Percentile Salary
## 2                       $108,000.00                       $152,000.00
## 3                       $127,000.00                       $161,000.00
## 4                        $96,300.00                       $150,000.00
## 5                        $89,300.00                       $138,000.00
## 6                        $97,000.00                       $136,000.00
```

```r
rm(h, nodes, tab, url)

##########################################Data
Manipulation######################################


if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-
project.org")
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

#Massage the data
colnames(raw_data) <- c("College.Major", "Starting.Median.Salaries",
"Mid.Career.Median.Salaries", "Career.Percent.Growth", "Percentile.10",
"Percentile.25", "Percentile.75", "Percentile.90" )
raw_data <- raw_data[-1,]
rownames(raw_data) <- 1:nrow(raw_data)

# Data Cleansing
degrees <- raw_data %>%
  mutate_at(vars(Starting.Median.Salaries: Percentile.90), function(x)
as.numeric(gsub('[\\$,]',"",x))) %>%
  mutate(Career.Percent.Growth = Career.Percent.Growth / 100)

rm(raw_data)

##########################################Exploratory Analysis###############
############

# Load packages

if(!require(tidyverse)) install.packages("tidyverse", repos =
"http://cran.us.r-project.org")

## Loading required package: tidyverse

## -- Attaching packages ----------------------------------------------------
---------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v readr   1.3.1
## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   0.8.3     v stringr 1.4.0
## v ggplot2 3.2.1     v forcats 0.4.0

## -- Conflicts -------------------------------------------------------------
---------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
## x purrr::pluck()          masks rvest::pluck()

if(!require(cluster)) install.packages("cluster", repos = "http://cran.us.r-
project.org")
```

```
## Loading required package: cluster

if(!require(factoextra)) install.packages("factoextra", repos =
"http://cran.us.r-project.org")

## Loading required package: factoextra

## Warning: package 'factoextra' was built under R version 3.6.2

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

if(!require(ggthemes)) install.packages("ggthemes", repos =
"http://cran.us.r-project.org")

## Loading required package: ggthemes
```

METHOD1. The elbow method

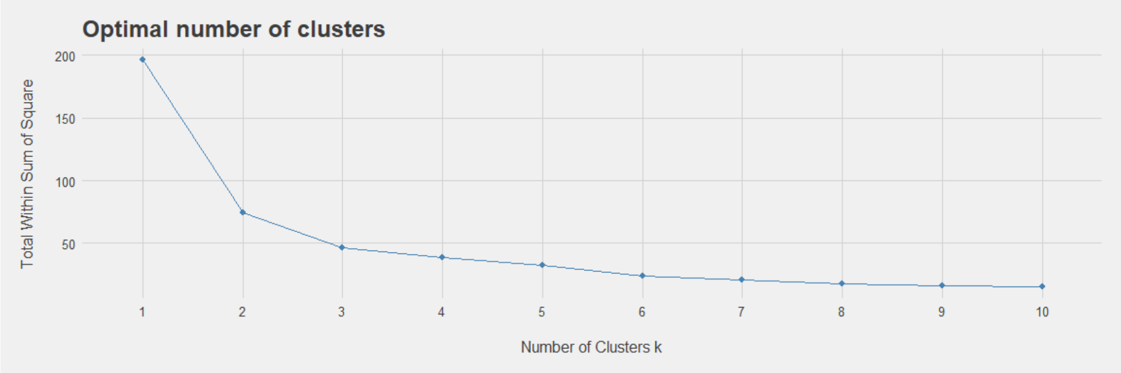
The Elbow method looks at the total WSS as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't improve much better the total WSS. The optimal number of clusters can be defined as follow: 1.Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters. 2.For each k, calculate the total within-cluster sum of square (wss). 3.Plot the curve of wss according to the number of clusters k. 4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

The Features used are as follows : *Starting.Median.Salary*, *Mid.Career.Median.Salary*, *Percentile.10*, and *Percentile.90. fviz_nbclust* function from the *factoextra* library is used to to determine and visualize the optimal number of clusters.

```
# Feature Selection for k_means_data
# Elbow method using Median and 10 -90 percentile

k_means_data <- degrees %>%
  select(Starting.Median.Salaries, Mid.Career.Median.Salaries, Percentile.10,
Percentile.90) %>%
  scale()

#METHOD1
# Execute using WSS Method and fviz_nbclust--ELBOW METHOD
elbow_method <- fviz_nbclust(k_means_data, FUNcluster = kmeans, method =
"wss")
elbow_method + theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  xlab('\nNumber of Clusters k') +
  ylab('Total Within Sum of Square\n')
```



METHOD2. The silhouette method

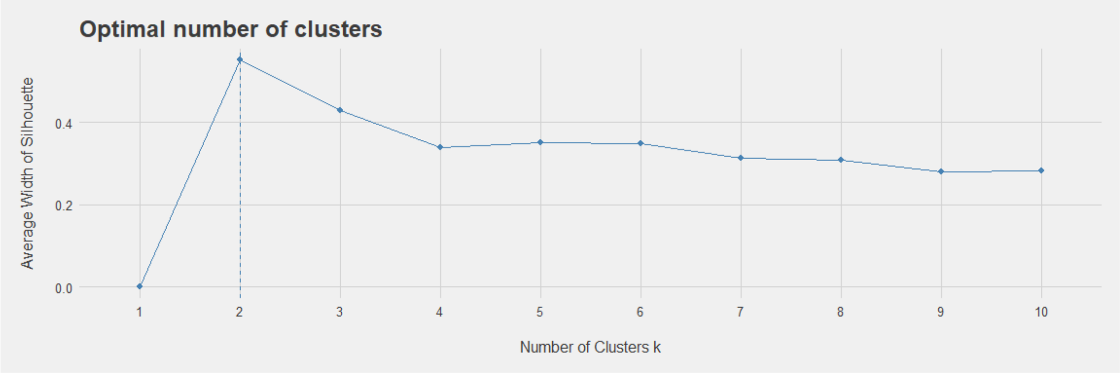
Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k (Kaufman and Rousseeuw 1990).

The algorithm is similar to the elbow method and can be computed as follow 1.Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters. 2.For each k, calculate the average silhouette of observations (avg.sil). 3.Plot the curve of avg.sil according to the number of clusters k. 4.The location of the maximum is considered as the appropriate number of clusters.

fviz_nbclust function is used to avoid "manually" applying the elbow method by running multiple k_means models and plotting the calculated total within cluster sum of squares for each potential value of k.

The Silhouette Method will evaluate the quality of clusters by how well each point fits within a cluster, maximizing average "silhouette" width.

```
#METHOD2
#Execute using the SILHOUETTE Method and the function fviz_nbclust- SILHOUTTE METHOD
silhouette_method <- fviz_nbclust(k_means_data, FUNcluster = kmeans, method = "silhouette")
silhouette_method + theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  xlab('\nNumber of Clusters k') +
  ylab('Average Width of Silhouette\n')
```



METHOD3. The gap statistic method

The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be value that maximize the gap statistic (i.e, that yields the largest gap statistic). This means that the clustering structure is far away from the random uniform distribution of points.

The algorithm works as follow:

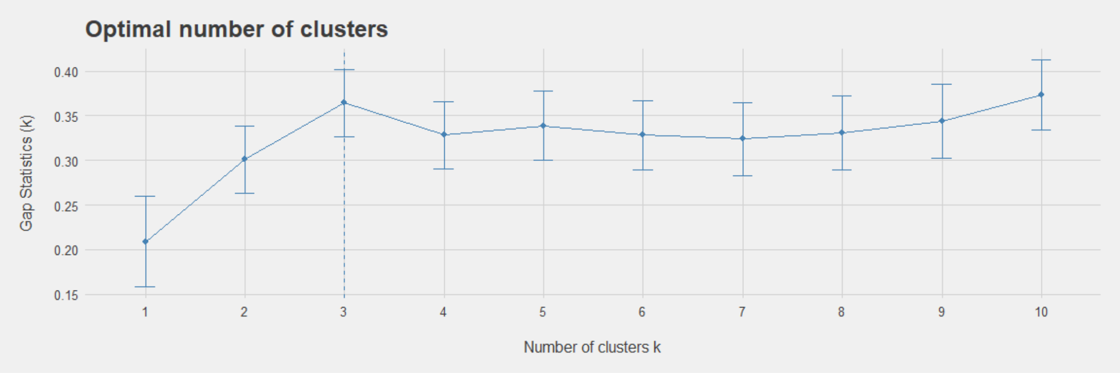
1.Cluster the observed data, varying the number of clusters from k = 1, …, kmax, and compute the corresponding total within intra-cluster variation Wk. 2.Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters k = 1, …, kmax, and compute the corresponding total within intra-cluster variation Wkb. 3.Compute the estimated gap statistic as the deviation of the observed Wk value from its expected value Wkb under the null hypothesis: Gap(k)=1B∑b=1Blog(W∗kb)−log(Wk). Compute also the standard deviation of the statistics. 4.Choose the number of clusters as the smallest value of k such that the gap statistic is within one standard deviation of the gap at k+1: Gap(k)≥Gap(k + 1)−sk + 1.

In other words, how much more variance is explained by k clusters in our dataset than in a faulty dataset where all majors have equal salary potential?

clusGap function calculates this and *fviz_gap_stat* function creates the visualisation for the result set.

```r
#METHOD3

# Execute using GAP STATISTICS Method and clusGap function
gap_stat <- clusGap(k_means_data, FUN = kmeans, nstart = 25, K.max = 10, B = 
50)
# Data Visualization using fviz_gap_stat function
gap_stat_method <- fviz_gap_stat(gap_stat)
gap_stat_method + theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  xlab('\nNumber of clusters k') +
  ylab('Gap Statistics (k)\n')
```



K-means Clustering Algorithm

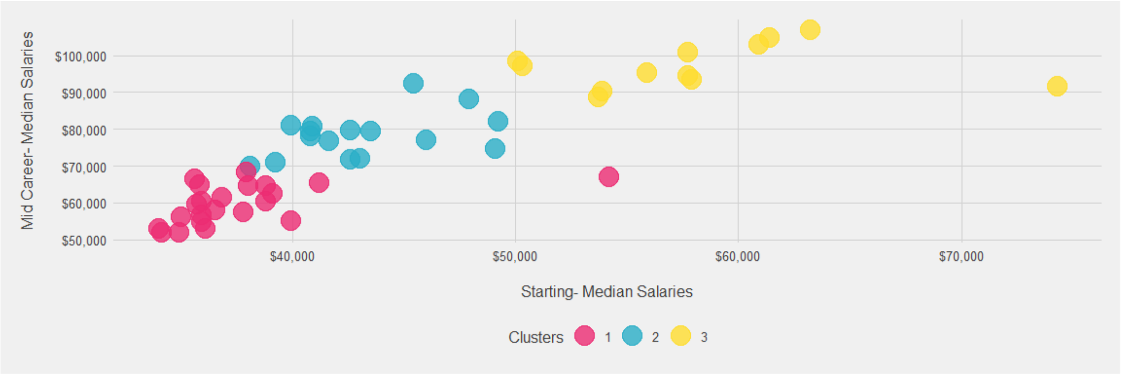
Gap Statistic Method and Elbow Method! produces similiar result. As per majority rule, let's use 3 for our optimal number of clusters. With this information, we can now run our k-means algorithm on the selected data. We will then add the resulting cluster information to label our original dataframe.

Visualizing the clusters

Lets visualize our results.Its based on how each cluster compares in Starting vs. Mid Career Median Salaries and what do the clusters say about the relationship between Starting and Mid Career salaries?

```
# Usage of K-means Clustering  algorithm and a random seed setting
suppressWarnings(set.seed(111, sample.kind = 'Rounding'))
# k=optimal number of clusters
num_clusters <- 3
# Executing k-means Clustering algorithm
k_means <- kmeans(k_means_data, centers = num_clusters, iter.max = 15, nstart
= 25)
# Labelling of degrees
degrees_labeled <- degrees %>%
  mutate(clusters = k_means$cluster)

# Data Visualisation of Clusters
# Starting and Mid Career Median Salaries Visualisation
career_growth <- ggplot(degrees_labeled, aes(x = Starting.Median.Salaries, y
= Mid.Career.Median.Salaries, color=factor(clusters))) +
  geom_point(alpha = 4/5, size = 7) +
  scale_x_continuous(labels = scales::dollar) +
  scale_y_continuous(labels = scales::dollar) +
  scale_color_manual(name = "Clusters", values = c("#EC2C73", "#29AEC7",
"#FFDD30"))
career_growth + theme_fivethirtyeight() +
  theme(axis.title = element_text()) +
  xlab('\nStarting- Median Salaries') +
  ylab('Mid Career- Median Salaries\n')
```



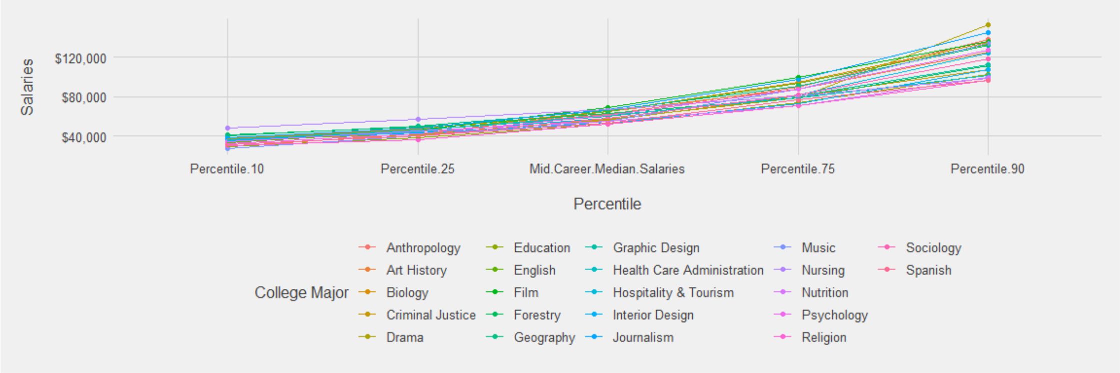
```
#Further detailed Analysis of clusters
# to reshape degrees using gather function and reorder the new percentile
column using mutate() function
degrees_perc <- degrees_labeled %>%
  select(College.Major, Percentile.10, Percentile.25,
Mid.Career.Median.Salaries, Percentile.75, Percentile.90, clusters) %>%
  gather(key=percentile, value=salaries, -c(College.Major, clusters)) %>%
  mutate(percentile = factor(percentile, levels = c("Percentile.10",
"Percentile.25", "Mid.Career.Median.Salaries", "Percentile.75",
"Percentile.90")))
```

Based on the result set , we classify them into 3 Clusters 1)Humanities -liberal cluster 2) The Challenger and 3) The leader

1. Humanities-liberal cluster

Plotting Cluster 1 and analysing the results we call them as “Humanities -liberals”.These majors may represent the lowest percentiles with limited growth opportunity, but there is hope for those who make it! Music is our riskiest major with the lowest 10th percentile salary, but Drama wins the highest growth potential in the 90th percentile for this cluster. Nursing is the outlier culprit of cluster number 1, with a higher safety net in the lowest percentile to the median. Otherwise, this cluster does represent the majors with limited growth opportunity.

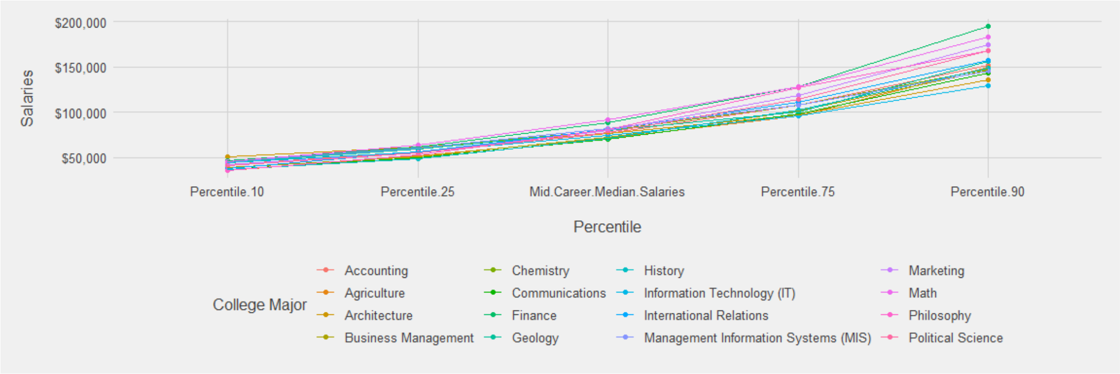
```
# Cluster1
# Cluster 1 Distribution by percentile and plotting of graph
cluster_1 <-  ggplot(degrees_perc %>% filter(clusters == 1),
aes(x=percentile, y=salaries, group=College.Major, color=College.Major)) +
  geom_point() +
  geom_line() +
  theme(axis.text.x = element_text(size=7)) +
  scale_y_continuous(labels = scales::dollar)
cluster_1 + theme_fivethirtyeight() + labs(color = "College Major") +
  theme(axis.title = element_text()) +
  xlab('\nPercentile') +
  ylab('Salaries\n')
```



2. The Challenger cluster

Plotting Cluster 2,and analysing the results we call them as “The Challenger” right in the middle! Accountants are known for having stable job security, but once you’re in the big leagues you may be surprised to find that Marketing or Philosophy can ultimately result in higher salaries. The majors of this cluster are fairly middle of the road in our dataset, starting off not too low and not too high in the lowest percentile. However, this cluster also represents the majors with the greatest differential between the lowest and highest percentiles.

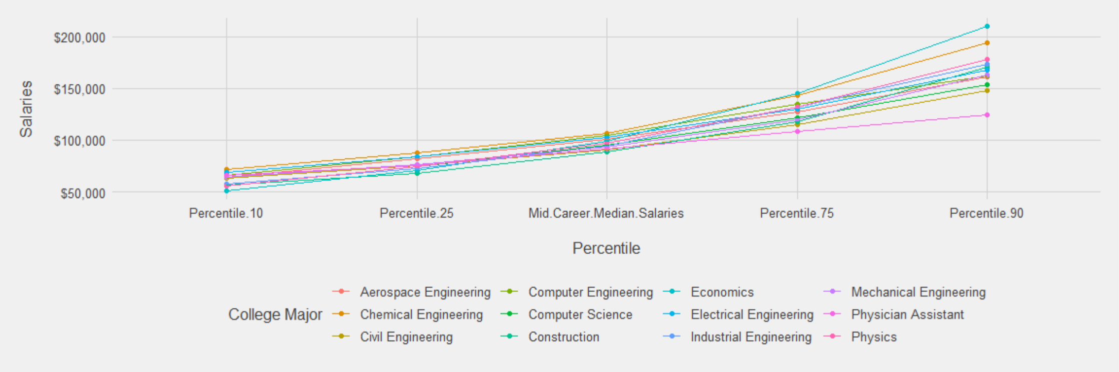
```
# Cluster2
# Cluster 2 Distribution by percentile and plotting of graph
cluster_2 <-  ggplot(degrees_perc %>% filter(clusters == 2),
aes(x=percentile, y=salaries, group=College.Major, color=College.Major)) +
  geom_point() +
  geom_line() +
  theme(axis.text.x = element_text(size=7)) +
  scale_y_continuous(labels = scales::dollar)
cluster_2 + theme_fivethirtyeight() + labs(color = "College Major") +
  theme(axis.title = element_text()) +
  xlab('\nPercentile') +
  ylab('Salaries\n')
```



3. The Leader cluster

Lastly,Plotting Cluster 3,and analysing the results we call them as “The Leader”.If you want financial security, these are the majors to choose from. Besides our one previously observed outlier now identifiable as Physician Assistant lagging in the highest percentiles, these hard hitters and tough engineers represent the highest growth potential in the 90th percentile, as well as the best security in the 10th percentile rankings.

```
# Cluster3
# Cluster 3 Distribution by percentile and plotting of graph
cluster_3 <-  ggplot(degrees_perc %>% filter(clusters == 3),
aes(x=percentile, y=salaries, group=College.Major, color=College.Major)) +
  geom_point() +
  geom_line() +
  theme(axis.text.x = element_text(size=7)) +
  scale_y_continuous(labels = scales::dollar)
cluster_3 + theme_fivethirtyeight() + labs(color = "College Major") +
  theme(axis.title = element_text()) +
  xlab('\nPercentile') +
  ylab('Salaries\n')
```



Below is an analysis of career growth wise Majors in descending order.

```
# Career.Percent.Growth -sorting them in this order
degrees_sorted <- degrees_labeled %>% arrange(desc(Career.Percent.Growth))
degrees_sorted %>% as_tibble()

## # A tibble: 50 x 9
##    College.Major Starting.Median~ Mid.Career.Medi~ Career.Percent.~
##    <chr>                    <dbl>            <dbl>            <dbl>
##  1 Math                     45400            92400            1.03
##  2 Philosophy               39900            81200            1.03
##  3 Internationa~            40900            80900            0.978
##  4 Economics                50100            98600            0.968
##  5 Marketing                40800            79600            0.951
##  6 Physics                  50300            97300            0.934
##  7 Political Sc~            40800            78200            0.917
##  8 Chemistry                42600            79900            0.876
##  9 Journalism               35600            66700            0.874
## 10 Architecture             41600           76800           0.846
## # ... with 40 more rows, and 5 more variables: Percentile.10 <dbl>,
## #   Percentile.25 <dbl>, Percentile.75 <dbl>, Percentile.90 <dbl>,
## #   clusters <int>
```

Results

In cluster analysis, since the number of clusters to be modelled, *k* is a hyper-parameter, choosing its value is not a clear-cut answer. To optimize the value *k* we used 3 methods viz. Elbow method, Silhouette method, and Gap Statistic method.

The value of *k* according to each method are as follows:

method	k
Elbow method	3
Silhouette method	2
Gap Statistic method	3

Based on majority rule, running K-means with *k* = 3, we attributed each major to one of the three clusters.Results post teh Visualisation are as below.

- **Cluster 1** majors may represent the lowest percentiles with limited growth opportunity.
  - Music is the riskiest major with lowest 10th percentile salary.
  - Drama has highest growth potential in the 90th percentile for this cluster.
  - Nursing is an outlier for this cluster with higher safety net in the lowest percentile to the median.
- **Cluster 2** majors start off not too low and not too high in the lowest percentile, but majors in this cluster represent greatest differential between the lowest and highest percentiles.
  - Accountants have stable job security.
  - Marketing or Philosophy ultimately result in higher salaries.
- **Cluster 3** majors are characterized by financial security and highest growth potential in the 90th percentile as well as best security in the 10th percentile rankings.
  - Physician Assistant is an outlier in this cluster lagging in the highest percentiles.

Conclusion

This concludes the salary projections by college majors via k-means clustering analysis. This is Unsupervised(unlabelled) data, hence we used the three popular methods to determine the optimal number of clusters. We also used visualizations to interpret the patterns revealed by our three clusters.

From the data, **Math** and **Philosophy** tie for the highest career percent growth. While it's tempting to focus on starting career salaries when choosing a major, it's important to also consider the growth potential down the road. Keep in mind that whether a major falls into the Humanities -liberal cluster,Challenger Cluster and Leader Cluster, one's financial success certainly is influenced by numerous other factor such as the school attended, location, passion or talent for the subject, and of course the actual career(s) pursued.

References

- http://online.wsj.com/public/resources/documents/info-Degrees_that_Pay_you_Back-sort.html?mod=article_inline