STAT 218 - Cluster Analysis

Jasmin Paed

Unsupervised clustering is commonly used in customer segmentation to uncover nontrivial groupings behind a wide range of data. In this study, we'll use the survey result of StackOverflow users conducted last 2018 to create clusters and uncover hidden insights using multivariate unsupervised clustering algorithms. We have the following goals in mind:

1. Find the "best" method in forming clusters

2. Describe and interpret the clusters formed

3. Provide conclusions that can be derived from the results

0.1 Data Exploration

First, we want to have a good look of the raw data that we'll use.

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   Respondent = col_double(),
##   AssessJob1 = col_double(),
##   AssessJob2 = col_double(),
##   AssessJob3 = col_double(),
##   AssessJob4 = col_double(),
##   AssessJob5 = col_double(),
##   AssessJob6 = col_double(),
##   AssessJob7 = col_double(),
##   AssessJob8 = col_double(),
##   AssessJob9 = col_double(),
##   AssessJob10 = col_double(),
##   AssessBenefits1 = col_double(),
##   AssessBenefits2 = col_double(),
##   AssessBenefits3 = col_double(),
##   AssessBenefits4 = col_double(),
##   AssessBenefits5 = col_double(),
##   AssessBenefits6 = col_double(),
##   AssessBenefits7 = col_double(),
##   AssessBenefits8 = col_double(),
##   AssessBenefits9 = col_double()
##   # ... with 23 more columns
## )
## i Use `spec()` for the full column specifications.
## [1] 98855   129
```

From above, we see that we have 129 columns. Most columns will not be included in this study for the ff reasons:
1. Redundant fields (i.e. ConvertedSalary will be retained and Currency, Salary, SalaryType will be removed)
2. Too many possible answers / Too many dummy variables to produce (e.g. Country field)
3. Character fields (e.g. reasons/explanation fields)
4. Questions that can be left blank. Since fill rate tends to be low.

Hence, for the purpose of keeping the study simple, we will only use demographics data with less than 3 categories and delete rows with NA values. We would also focus on student respondents to get an idea of what an early career in coding looks like.

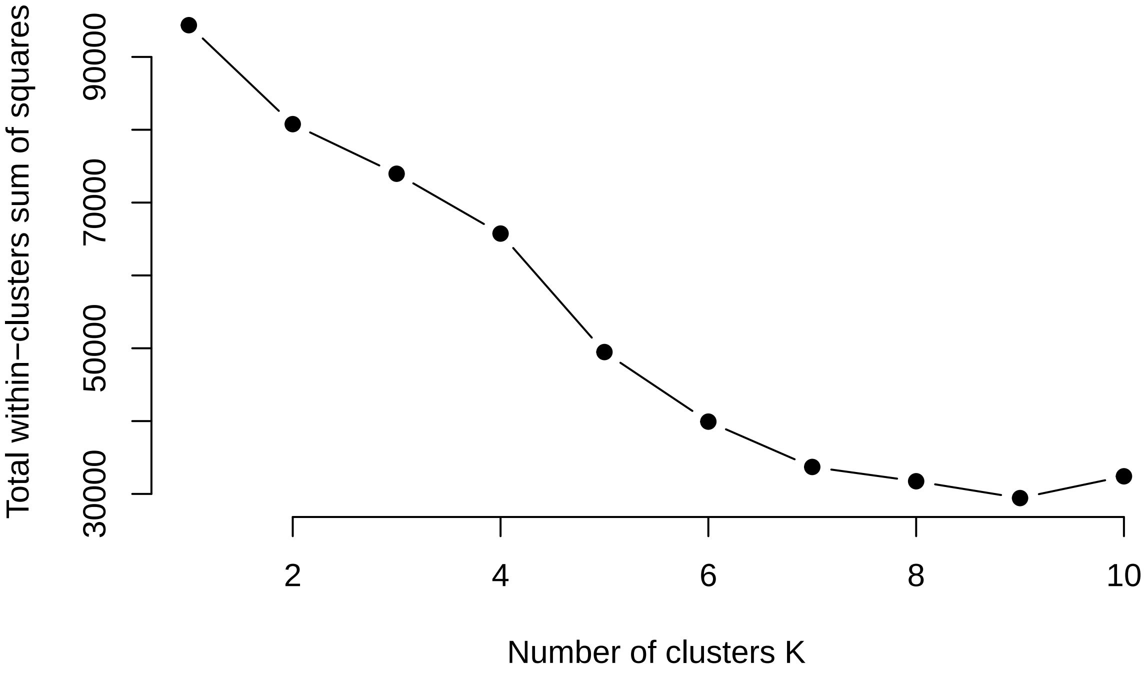
	Hobby	OpenSource	Employment	FormalEducation	YearsCoding	ConvertedSalary	Dependents
7	1	0	1	0	6	21426	1
21	0	0	1	1	1	0	0
41	1	1	1	1	9	51408	1
50	1	1	1	1	9	0	0
56	1	0	1	1	6	0	1
76	1	1	1	1	6	0	0

```
## 'data.frame':    13481 obs. of  7 variables:
##  $ Hobby          : num  1 0 1 1 1 1 1 1 1 1 ...
##  $ OpenSource     : num  0 0 1 1 0 1 1 1 0 1 ...
##  $ Employment     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ FormalEducation: num  0 1 1 1 1 1 1 1 1 1 ...
##  $ YearsCoding    : num  6 1 9 9 6 6 6 9 3 6 ...
##  $ ConvertedSalary: num  21426 0 51408 0 0 ...
##  $ Dependents     : num  1 0 1 0 1 0 0 0 0 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:11021] 3 4 5 9 10 15 19 20 21 24 ...
##   ..- attr(*, "names")= chr [1:11021] "3" "4" "5" "9" ...
```

After cleaning and preparing the data to be suitable for clustering, we would now proceed with building the model. Note that we will scale the pre-processed data to avoid unnecessary impact of higher units in the result.

0.2 KMEANS Clustering

This clustering method uses Euclidean distance and starts by choosing certain centroids based on number of clusters (K) defined. Hence, the crucial part is choosing the right number of clusters. One of the method to choose K is plotting the WSS (within cluster sum of squares) and determine the elbow-like curve. This means that after the chosen K, WSS value seems to stabilize.



Using the elbow method, we could select 5, 6, or 7. To further decide on the number of clusters to use, we can compare the results of each K for 5,6, and 7. As seen below, outputs are unbalanced clusters where there are only around 1% of respondents in one cluster.

```
## [1] "Cluster Number: 5"
## Cluster Size: [1]  155 2408 2469 6498 1951
## [1] "Cluster Within Sum of Squares: 47889"
## [1] "Cluster Total Sum of Squares: 94360"
## [1] "Cluster Number: 6"
## Cluster Size: [1] 5562 1707 2124 1767 2169  152
## [1] "Cluster Within Sum of Squares: 39363"
## [1] "Cluster Total Sum of Squares: 94360"
## [1] "Cluster Number: 7"
## Cluster Size: [1] 1749 2747 1768  149 2169 2078 2821
## [1] "Cluster Within Sum of Squares: 33697"
## [1] "Cluster Total Sum of Squares: 94360"
```

We will use cluster number 7 since it has lowest WSS value. We can plot the graph to visualize the result more. Figure 1 is a plot of the respondents colored based on Kmeans cluster result using K=7. In this graph, Cluster 7 has the high earners. But since this is just a two-dimensional view, it will not show the groupings clearly. Hence, we may opt to do a Principal Component Analysis graph of the output as shown in Figure 2. To visualize the output of the clusters:

Figure 1. 2D Graph of Kmeans Clustering

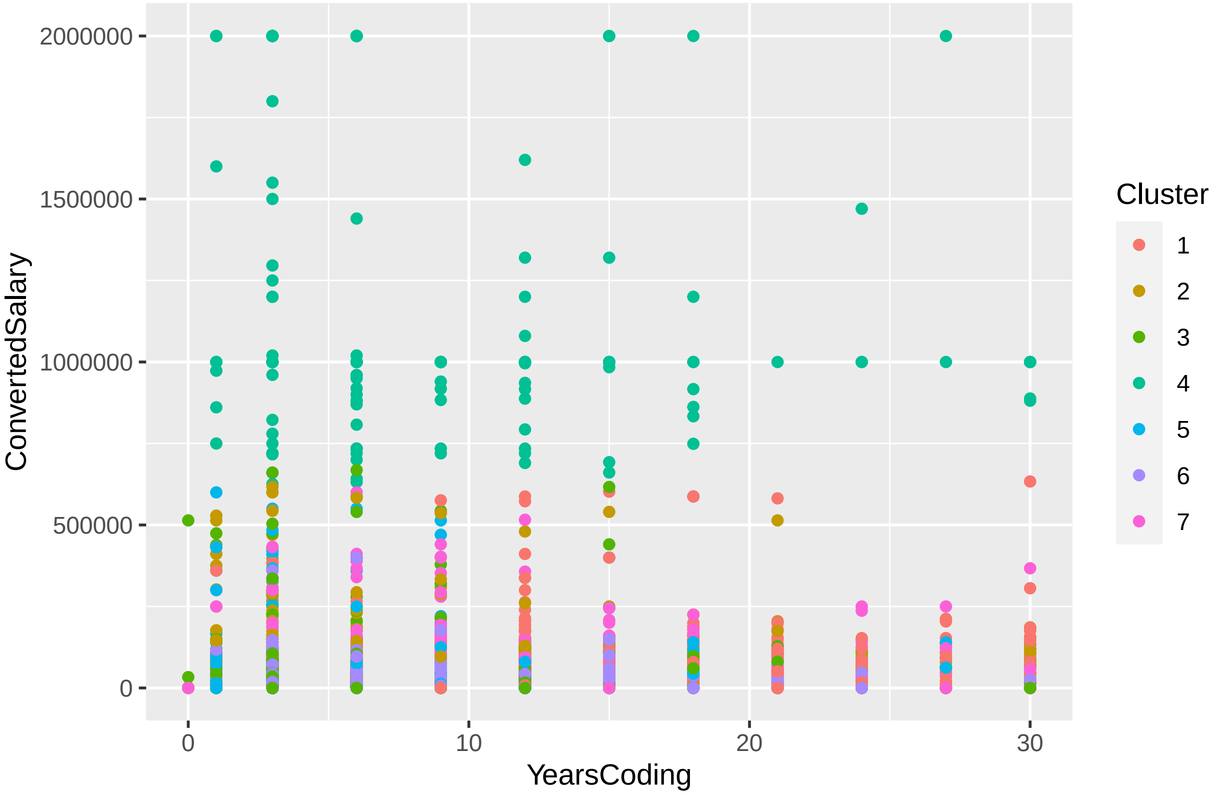
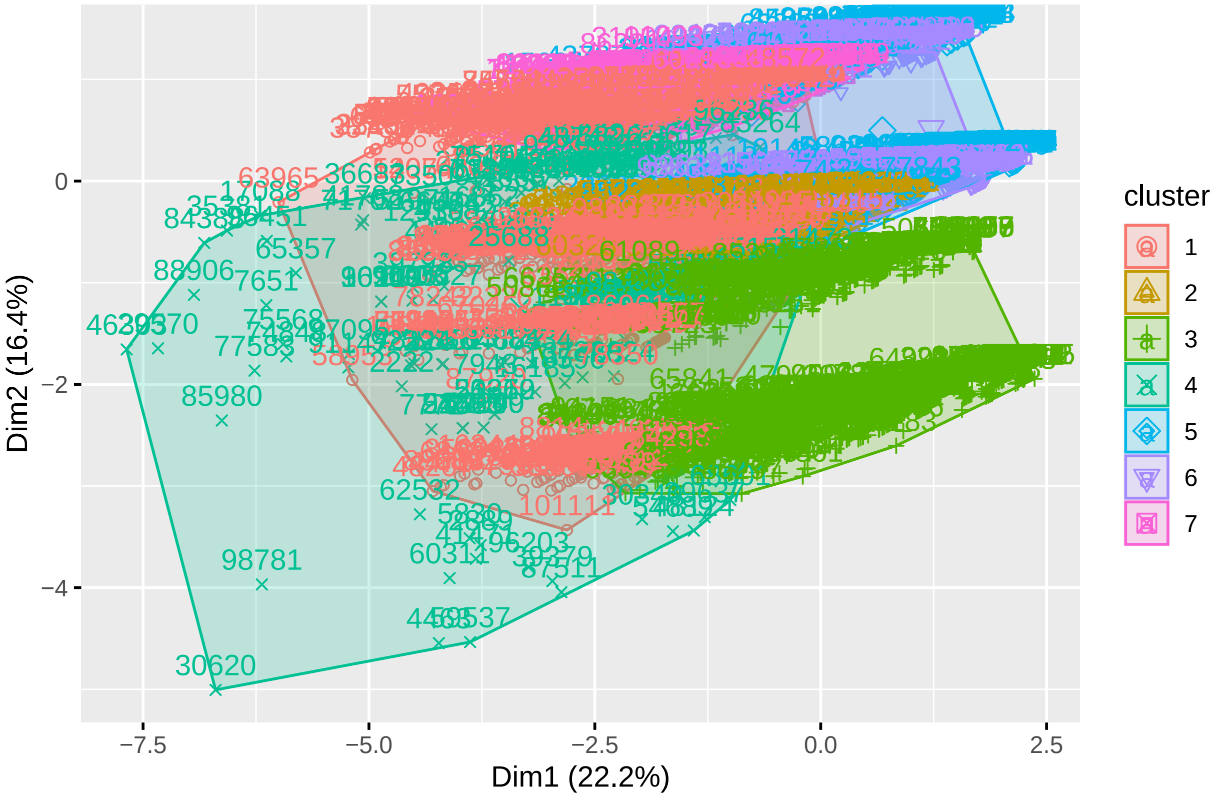


Figure 2. PCA Visualization



Though performance of Kmeans clustering is faster, we should try other clustering methods for better analysis.

0.3 Hierarchical Clustering

First, we'll use Euclidean distance for the dissimilarity measure and look at the three common linkages - complete, single, and average. Since previous elbow method suggests to cut the dendogram at 7 clusters, we'll use that as initial number of clusters. However, as seen below, the number of respondents in a single cluster yields to extreme unbalanced clusters.

```
## # A tibble: 7 x 2
##   complete     n
##      <int> <int>
## 1        1  9434
## 2        2  3780
## 3        3   112
## 4        4    29
## 5        5    82
## 6        6    37
## 7        7     7
```

```
## # A tibble: 7 x 2
##   average     n
##     <int> <int>
## 1       1 13292
## 2       2    28
## 3       3   135
## 4       4     6
## 5       5    12
## 6       6     7
## 7       7     1
```

```
## # A tibble: 7 x 2
##   single     n
##    <int> <int>
## 1      1 13474
## 2      2     1
## 3      3     2
## 4      4     1
## 5      5     1
## 6      6     1
## 7      7     1
```

We'll now use correlation-based distance and see if it can produce more balanced clusters. The clusters produced by the correlation-based distance using complete linkage produced a more desirable result as seen below. We can cut the dendrogram at the height that will yield seven clusters:

```
## # A tibble: 5 x 2
##   corr_complete     n
##           <int> <int>
## 1             1  3453
## 2             2  4358
## 3             3  2446
## 4             4  2223
## 5             5  1001
## # A tibble: 6 x 2
##   corr_complete     n
##           <int> <int>
## 1             1  1417
## 2             2  2036
## 3             3  4358
## 4             4  2446
## 5             5  2223
## 6             6  1001
```

```
## # A tibble: 7 x 2
##   corr_complete     n
##           <int> <int>
## 1             1  1417
## 2             2  2036
## 3             3  4358
## 4             4  2246
## 5             5  2223
## 6             6  1001
## 7             7   200
```

The clusters produced by the correlation-based distance using complete linkage produced more desirable results. We will confirm the ideal number of clusters even further using a dendogram:

Figure 3. Complete Dendogram with Ideal Cut

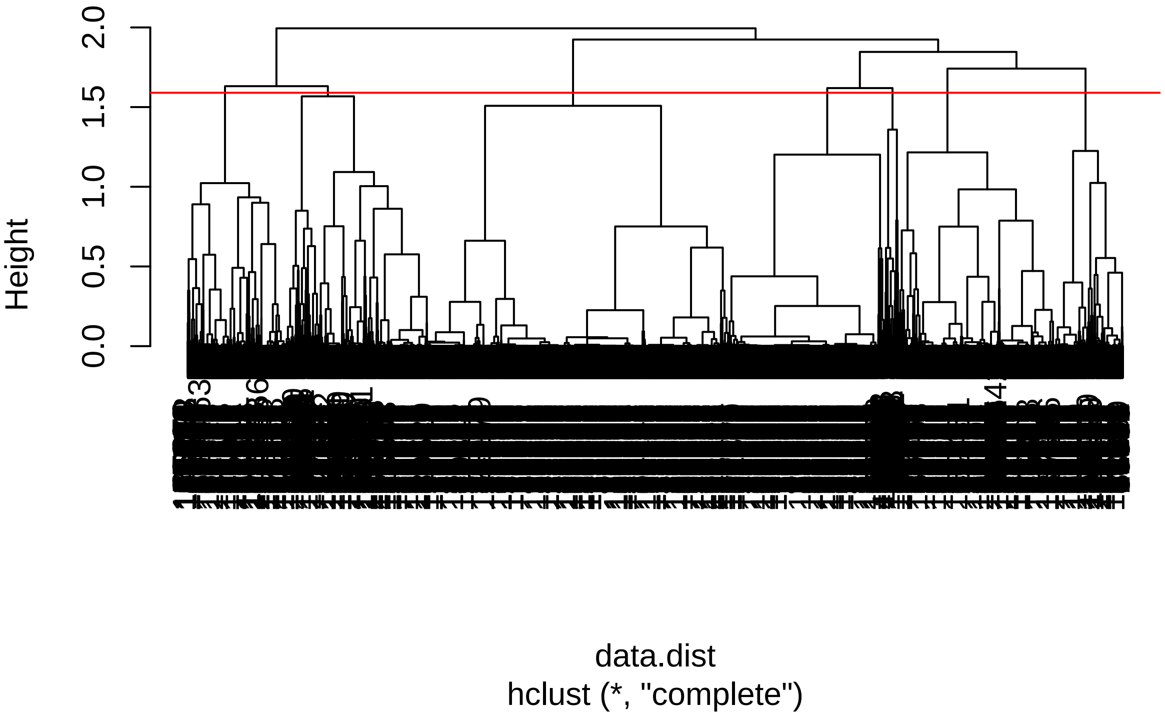
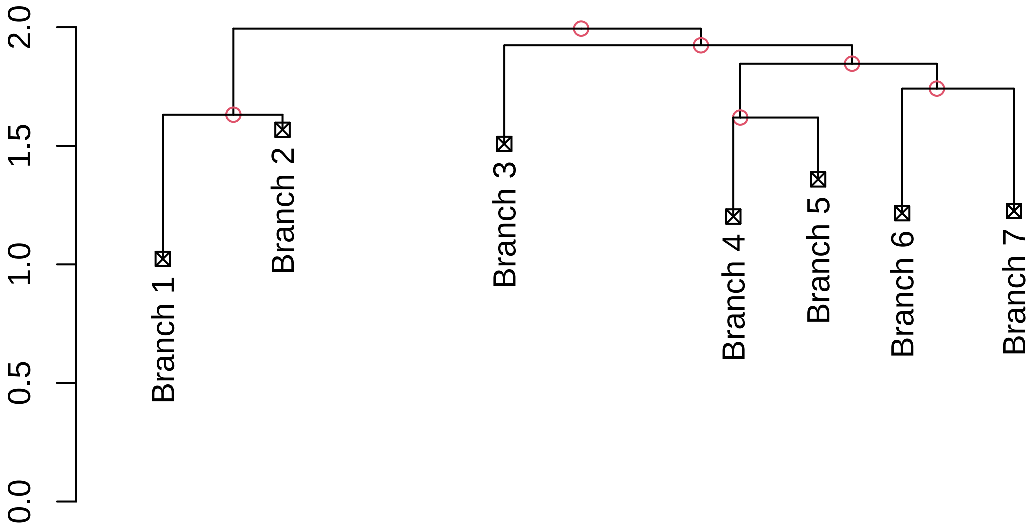
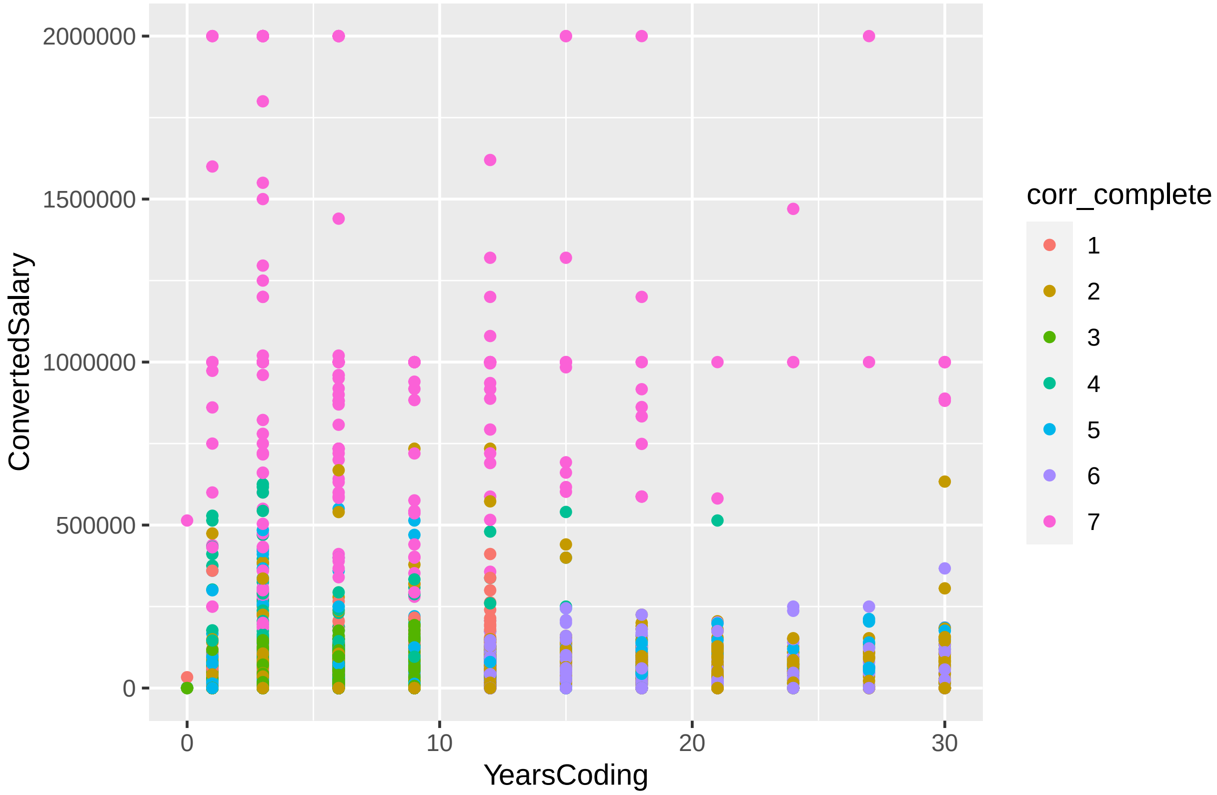


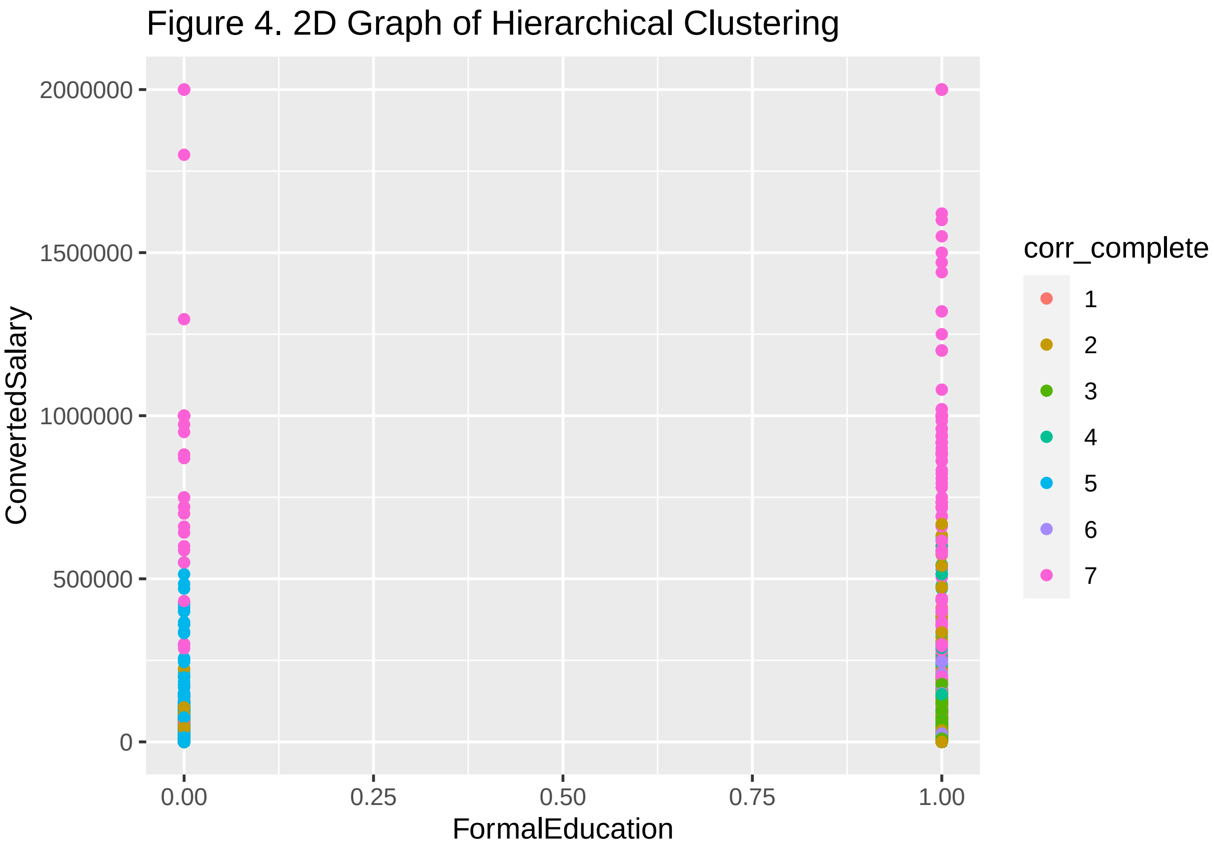
Figure 4. Dendogram Focusing the Seven Branches



To vizualize the correlation-based distance using complete linkage clusters:

Figure 3. 2D Graph of Hierarchical Clustering





0.4 Cluster Goals

0.4.1 Best Method

For this study, we will use the correlation-based distance using complete linkage clustering since it focuses on the shapes of observation profiles rather than their magnitudes. We will use seven clusters for our analysis based on the findings above. Also, since this needs only to be run once then the performance of the model is not an issue.

0.4.2 Description

Description of each clusters are as follows:

cluster	n	Hobby	OpenSource	Employment	FormalEducation	YearsCoding	ConvertedSalary
1	1417	0.83	0.52	0.89	0.85	5.36	27100.33
2	2036	0.18	0.33	0.79	0.84	8.03	29159.60
3	4358	1.00	0.73	0.55	1.00	4.25	12118.48
4	2246	1.00	0.00	1.00	1.00	3.71	24843.00
5	2223	1.00	0.45	0.71	0.00	5.45	19630.08
6	1001	0.99	0.46	0.91	1.00	13.65	33562.36
7	200	0.86	0.54	0.96	0.80	8.04	971783.85

The Common Ones – The cluster with the most number of respondents is 3 with an average annual salary of 1.2118482×10^4 dollars. Out of all the clusters, they earned the least. All are with formal education but only half are employed. They have an average of 4 years coding experience.

The Above Common Ones – Cluster 1 are very much alike to the common ones except that they are earning more.

The Rich Ones – The cluster with the least number of respondents is 7 but with the highest average annual salary of 9.7178385×10^5 dollars. Almost all are employed but not all has a formal education.They have an average of 8 years coding experience.

The Kind Ones – Cluster 2 has similar description like the rich ones but they are earning less. They use open source software less and code as a hobby less.

The Experienced Coders – Cluster 6 has the highest years of coding experience of `clust_summ$YearsCoding[6]` on average. Most are employed with formal education and they are the second to the highest earners.

The Closed Dudes – Cluster 4 respondents are all employed with formal education. Though they do not like open source software.

The Bonakids – CLuster 5 are earning more than the common ones even without a formal education. Most are employed and code as a hobby.

0.4.3 Conclusion

Almost all coders have used StackOverflow and value the community of sharing and collaborating. Monitoring the users of the website could prevent it from being irrelevant. Furthermore, since it is a free website, ads and other marketing campaigns could use the cluster findings above to target consumers and interested segments efficiently.