Introduction

In this assignment, a general attitudinal post-hoc segmentation analysis was completed using a sample survey dataset consisting of 1,800 observations. The intent of this analysis is to help App Happy develop a deeper understanding of the market for a new social entertainment app. Since App Happy is not yet in the consumer app business, the hope is to provide a comprehensive understanding of distinctive groups that the organization can leverage using segment specific marketing strategies.

Attitudinal questions around the respondents' level of agreement on technology, personal characteristics, and purchasing behavior were formed as the basis variables. The responses were foundational to the construction of three different clustering models (i.e., schemas): K-Means, PAM (partition around mediods), and Hierarchical. Once the appropriate method was identified, it was used to profile the derived cluster groups to provide greater insight into each segment using demographics or personality characteristics. Furthermore, a brief discussion has been provided to highlight the potential techniques that could be used as a ”typing tool” to classify customers for whom no data exists.

Data Overview

The sample survey dataset (provided by the Consumer Spy Corporation) contains 1,800 observations across 89 variables. However, upon exploration of the data dictionary, it appears that there are 16 distinct questions - some of which have 'sub-questions'. The information was provided through an R data file and consisted of two dataframes: a numerically coded response data frame and a data frame with the character values of the survey responses. From the data dictionary provided, it appears that there may have been 57 questions on the survey (compared to the 16 made available). Of the 16 primary questions, three of them are attitudinal and the responses to these three questions make up approximately 41 of the 89 variables. The other questions within the survey dataset provide both demographic and personality characteristics.

A comprehensive exploratory data analysis was completed and a small sample of the histograms generated are provided in the Appendix. An interesting note to make is the skewness of the respondents' age. The majority of observations seem to be for individuals that are 25 and younger. Furthermore, it also appears that there are more female respondents than males.

Market Segmentation

Attitudinal Variables

Questions 24, 25, and 26 are prime candidates for attitudinal segmentation. All three questions use a Likert scale (a general 6-point scale) where 1 means ”Strongly Agree” and 6 means ”Strongly Disagree”. Question 24 appears to focus on technology; question 25 appears to focus on an individual's personal characteristic & point of view; and question 26 appears to focus on an individual's purchasing preferences. These three questions seem best suited to ascertain an individual's perspective or attitude.

Due to the large number of potential attitudinal variables, a principal component analysis (PCA) was initially done to understand the amount of variation explained. It was discovered that in order to explain at least 70% of the variation, 14 principal components would be required (see Table 9.1 in the Appendix). The attitudinal variables were then regrouped to create a more concise grouping in an effort to reduce the number of principal components. Table 9.2 in the Appendix summarizes how the responses were grouped (i.e., consolidated). In other words, the consolidated values were essentially the average of the response values for each group. Another PCA was completed using this new grouping structure and it was discovered that three principal components were needed to explain at least 70% of the variation (see Table 9.3 in the Appendix). Thus, this new grouping dataset is used as foundation for the different clustering methods.

Clustering Methodology

This analysis focuses on three different clustering methods: K-Means, PAM, and Hierarchical. Two of the methods (K-Means and PAM) are non-hierarchical methods that determine a centroid or mediod, respectively, and effectively try to minimize the Euclidean distance between the centroid/mediod and the observation. The hierarchical method attempts to take each observation as its own cluster and then combine it with neighboring observations or clusters to create a tree-like structure. Eventually, the algorithm ends when all of the nodes are linked (Chapman & Feit, 2015).

K-Means Clustering

K-Means typically requires a pre-defined number of clusters. Two methods were employed to help determine the optimal number of clusters: a scree plot and the average silhouette width. A scree plot essentially summarizes the minimum within-cluster sum of square for each number of clusters. Identifying where the ’bend’ takes place on the plot would imply an optimum number of clusters.

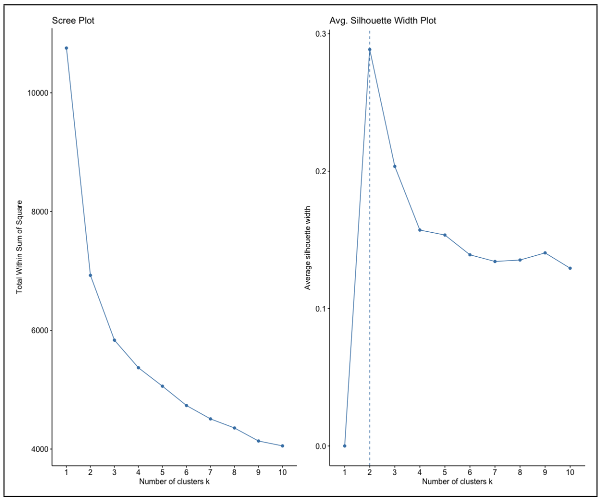

Figure 3.1: Scree Plot & Average Silhouette Width Plot

Figure 3.1 illustrates both the scree plot and the average silhouette widths for multiple clusters. From the scree plot, it appears that the ’bend’ occurs at either two or three number of clusters. However, from the average silhouette width plot, it is clear that the optimal number of clusters is two. Note that it is not sufficient to say that two clusters is sufficient without comprehensively profiling the clusters.

Assuming that two clusters are chosen, the following plot (Figure 3.2) illustrates the first two principal components (which explain almost 70% of the variation - see Table 9.3 in the 9). Note how the two clusters are not very well separated. In fact, there are several overlapping points within this plot. This suggests that some of the responses to the survey overlap in terms of attitudinal responses.

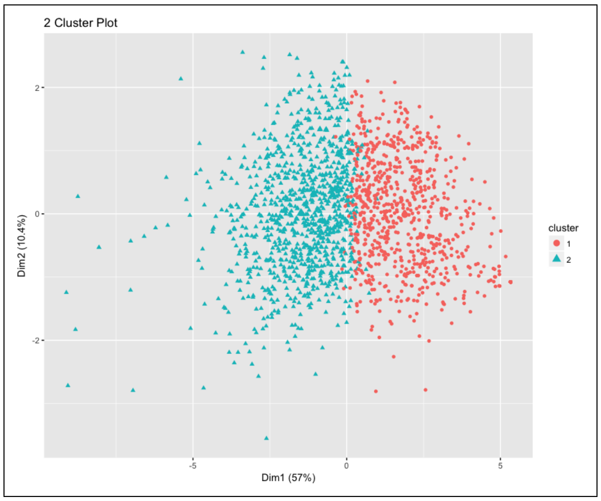

Figure 3.2: Two Cluster PCA Plot

PAM Clustering

PAM (Partitioning Around Mediods) clustering is similar to K-Means, however, the two key differences are that the mediods are used instead of centroids and the dissimilarity-based distance is used instead of the squared-Euclidean distance (Izenman, 2013). This enables the algorithm to be a bit more resistant to the influence of outliers. Similar to Figure 3.1, Figure 3.3 illustrates the scree plot and the average silhouette width for this method.

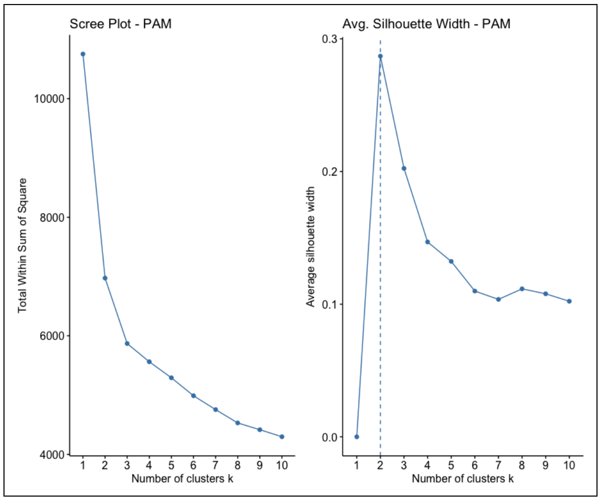

Figure 3.3: Scree Plot & Average Silhouette Width (PAM Clustering)

Note how there is a distinctive bend (when looking at the scree plot) at three clusters. However, the average silhouette width plot continues to show that two clusters is optimal. Figure 3.4 shows the two clusters superimposed on an XY plot of the first two principal components. This figure is very similar to Figure 3.2.

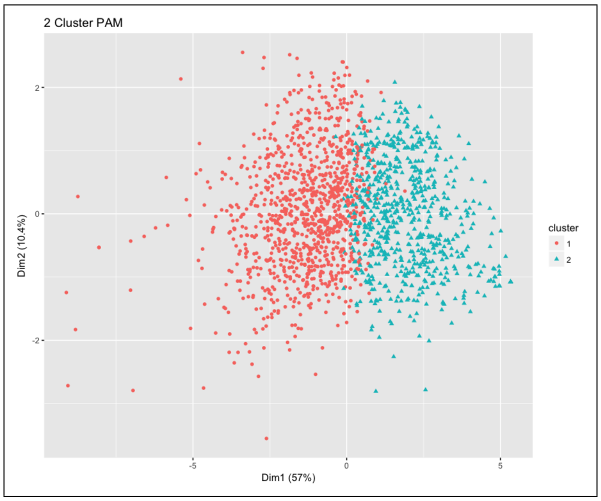

Figure 3.4: PAM Two Cluster PCA Plot

Hierarchical Clustering

Similar to the procedures for the K-Means Clustering and PAM Clustering, the scree and the average silhouette width plots are constructed.

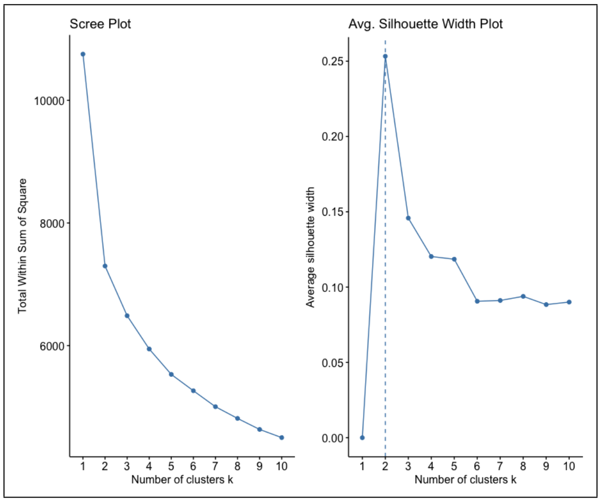

Figure 3.5: Scree Plot & Average Silhouette Width (Hierarchical Clustering)

It is not especially clear where the ’bend’ is in the scree plot (Figure 3.5), however, it appears that two clusters would be optimal. This assumption is confirmed in the average silhouette width plot (Figure 3.5). For the hierarchical method, we leveraged the agglomerative technique. Essentially, this technique assumes that each observation is its own cluster and then links them together to form the upper parts of a tree diagram - known as a dendrogram.

The dissimilarity matrix was used with the Euclidean distance method, however, the linkages were investigated to ensure optimum dissimilarity between two clusters of observations (Analytics, n.d.). This approach led to the investigation of multiple linkage methods (e.g., complete, single, average, and Ward). Therefore, the agglomerative coefficient, which is used to determine which measuring method, can determine the strongest clustering structure. Table 3.1 summarizes the results of this analysis. Note the closer the coefficients are to 1 the stronger the clustering structure.

Table 3.1: Agglomerative Coefficient Analysis

Average	**Single**	**Complete**	**Ward**
0.874	0.714	0.929	0.989

Note how the Ward method was very close to 1. The Ward method attempts to minimize the total within cluster variance (Analytics, n.d.). Hence, the Ward linkage method is used for the dendrogram creation. Figure 3.6 is the dendrogram using the agglomerative approach with the Ward linkage.

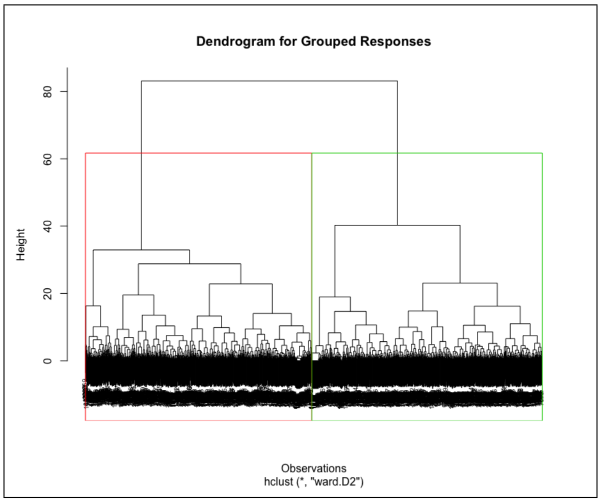

Figure 3.6: Dendrogram

Recall that from the scree and average silhouette width plots (Figure 3.5), two clusters were determined to be optimal. In Figure 3.6, the two clusters are ’encircled’ with two rectangles. Figure 3.7 is the XY plot of the first two principal components with two clusters highlighting the observations. Note the similarity between this method and the previous methods.

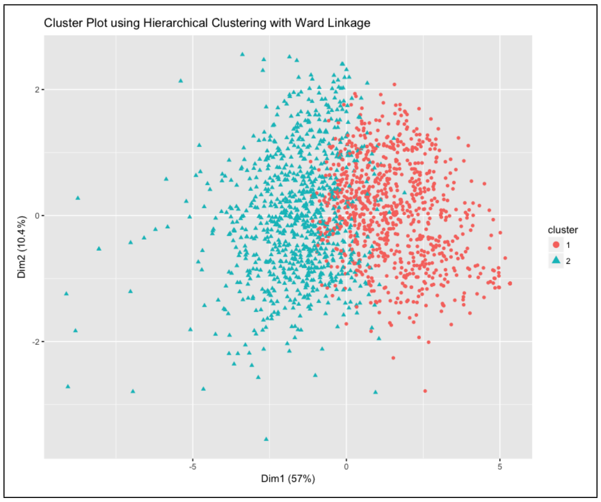

Figure 3.7: Hierarchial Two Cluster PCA Plot

Clustering Method Review & Conclusions

From the three different clustering methods analyzed for this analysis, it is not readily apparent which method may be of most value. Furthermore, the different scree plots and average silhouette width plots for all three methods suggested either two or three number of clusters. Reflecting on the different XY plots constructed, it is clear that there are many points that are overlapping. Ultimately, the profiling of each of the clusters will help determine the contextual value. Therefore, we have selected to proceed forward with two clusters using the K-Means method.

Profiling with K-Means

In order to provide context around the chosen number of clusters, both visual histograms and tabular data of the histograms were leveraged. The contextual profiles have been separated into two features: demographic and preferences. Many of the figures that are referenced in the next parts can be found in the Appendix along with tables that describe the observations proportionally.

Demographics

In terms of age, it appeared that cluster two was made up older consumers (41% of the observations were 40 and older) whereas cluster one was more geared towards a younger (48% were younger than 35) audience (see Figure 9.4 and Table 9.4 in the Appendix). In terms of education, cluster two seemed to consist more of observations (12% for cluster 2 versus 9% for cluster 1) with a post-graduate degree. Cluster two also seems to consist of more separated, widowed, and/or divorced respondents (11% vs 6% - see Table 9.6). However, both clusters have approximately equal number of married and single consumers. Interestingly, cluster two consumers seem to have older children (ages 18+) whereas cluster one seems to be mostly no children or younger children (younger than age 6). Both clusters are predominantly White/Caucasian and do not identify as Latino/Hispanic (Figure 9.8). Both clusters are also predominately female (Figure 9.9). Finally, both clusters seem to have a large portion of observations between $30,000 and $80,000 (Figure 9.10). However, cluster two also has a larger number of observations at the higher end of the income level (although proportionally, they are equal).

From a demographics point of view, cluster two seems to be made up of consumers that are slightly older, more likely to possess a post-graduate degree, more likelihood of being separated/divorced/widowed, and have a higher income. On the other hand, cluster one seems to consist of a younger audience that is still educated (college graduate), have younger or no children, single or married, and a higher likelihood of having a more modest income.

Preferences

In both clusters, the prevalent device owned was the iPhone (see Figure 9.11) followed by Android (which is an OS[1], and not a device). In addition, both clusters had substantial tablet ownership, however, it is difficult to determine what kind of tablet users (e.g., brand and OS type). Both clusters were pretty even (see Figure 9.12) when it came to general news apps, gaming, and shopping apps. Cluster 2 had a higher proportion preferring music identification apps (see Table 9.13). However, cluster one seemed to have a higher preference for TV check-in, TV show apps, and entertainment like apps. Cluster two, however, was more inclined to prefer social networking apps.

Both clusters seem to have an even concentration of consumers with 11 to 30 plus apps on their device. However, cluster two seems to also have a larger number of consumers with fewer than ten apps (see Figure

[1] Operating System

9.13). Cluster one also had a larger preference for fewer free apps as compared to cluster 2 observations (see Figure 9.14). For many of the specific websites identified in the survey, it seems that consumers in cluster 2 do not visit them (see Figure 9.15). Antithetically, consumers in cluster one appear to visit some of the pre-defined websites more frequently.

From a consumer preferences point of view, cluster one appears to have preference for TV related apps and tend to have numerous apps installed with a strong inclination to have paid apps. Consumers in cluster two seem to prefer more social networking and music identification apps. They also tend to have more free apps and do not visit certain websites at all.

Product Opportunity Recommendations

App Happy is interested in developing a social entertainment app, however, that definition could be broad. One cluster of consumers prefers entertainment like apps while the other cluster of consumers prefers more social media apps. If App Happy were to develop an app that catered to both tastes, then there is a good opportunity for App Happy to realize incredible results. An example app that App Happy may wish to investigate further would be a social media app that revolves around a popular TV show. This would enable App Happy to enter a niche market with ample opportunity to create delightful solutions with potentially strong adoption.

Another avenue that App Happy will have to explore is the platform of their social entertainment app. From the results, it is clear to see that both clusters have a strong preference for either an iOS based device (e.g., iPhone, iPod Touch) or Android. Focusing on these two platforms will encourage disciplined expenditures and increased adoption. However, it is not clear what kind of tablet OS is preferred for either cluster. From a demographics point of view, App Happy may want to consider using a website to drive traffic to the app as well since younger consumers may prefer a website over an app while older consumers may prefer an app to a website.

App Happy may want to consider conducting another survey that is more focused on social media and the type of devices used. In modern times, the distinction between social media networks and entertainment apps is virtually non-existent. For instance, Facebook hosts videos, live feeds, and more. However, Facebook is first and foremost a social networking site, but it is also considered an entertainment portal. A similar argument could be made for YouTube. Another avenue of research would be to determine the functionality of a website and an app. For instance, a website could be used to provide information and act as a landing page. However, the main focus could be on the app - especially since mobile device usage is on the rise. Finally, App Happy should consider more finely defining what the term social entertainment means. In some regards, LinkedIn is not necessarily a social entertainment platform. It is more a professional networking site and is not recommended to be used as a personal platform as compared to Facebook.

Limitations & Assumptions

Recall that the chosen clustering method for this analysis was K-Means. One key assumption that was made in order to leverage this approach was that the response values were continuous for the attitudinal questions. Since K-Means is working with distances between observations and centroids, categorical values are incompatible.

One limitation of the analysis is its focus on a subset of the information on the overall survey. It appears that the survey consisted of 57 questions, however, only a small portion of them were available for analysis. There is potential that other missing questions may have provided deeper insights when conducting the profiling. Another limitation found was around some of the responses in the survey. For instance, in question 2, the respondents could pick from an iPhone, iPod Touch, Android, etc. However, Android is not a type of device. Rather, it's an operating system such as iOS (which powers both iPhones and iPod Touches). This could have resulted in confused responses since some respondents may not have known that (for example) a Samsung smartphone may have been running Android. Thus, they may have marked Android or other since Samsung wasn't listed as an option. Furthermore, one of the choices for this question was ”Tablet” which lumped both iPads (powered by iOS) and multiple Android powered tablets into one. A suggestion here would be to reword the question to separate tablet and mobile users and then further separate them by type of OS. Similarly, question 50 (regarding children of certain age groups) could be expanded to let the respondent indicate how many children in each age group they have. This could also prove to beneficial in terms of demographic profiling. In other words, certain questions should either be rewritten or expanded upon to help create a more comprehensive insight.

Another limitation was the decision to not remove outliers in responses or potentially false responses. For instance, a respondent may have answered (on the Likert scale) all 1's or all 6's (specifically on questions 24 - 26). These may be accurate responses, but the way the questions are structured and written, some of the responses are contrary and it would be unlikely that the agreement level would be the same. Removing these values may prove to improve the overall clustering schema.

It is also not very clear from the survey the distinction between visiting a social media website versus using the social media website's app. For instance, it is not very clear to distinguish users who use Facebook's app more than visiting the website since much of that will depend on the usage of a mobile device and a personal computer.

Classification Models Discussion

Another requirement from App Happy is the ability to classify consumers that it does not currently have information on. This can be accomplished by leveraging the aforementioned clustering technique. For instance, the samples used all have a cluster number attached to them and now a predictive model could be built on top of that to facilitate this task. Two potential predictive models that could be leveraged are Random Forest and Naive Bayes. For both approaches, the response (or the dependent variable) would be the cluster number and the predictors would be any of the demographic, attitudinal, or consumer preferences that a consumer may supply or be available.

Random Forest

Random Forest is essentially an advanced decision tree method that enables complex relationships to be modeled quite well. What separates Random Forest from other tree-based models is that it can sample both the observations and the predictors (Bruce & Bruce, 2017). Furthermore, this method constructs hundreds of trees and the ability to only sample a subset of the observations and predictors enables the trees to be decorrelated (James, Witten, Hastie, & Tibshirani, 2015). One key benefit of the Random Forest approach is its ability to reduce overfitting. A major disadvantage of this approach is the inability to visually illustrate the different decision trees that are constructed.

Naive Bayes

The Naive Bayes approach is another interesting approach that essentially assumes that the pair of predictors are independent. This enables the method to take into account both discrete and continuous features (Hastie, Tibshirani, & Friedman, 2009). From a classification standpoint, Naive Bayes could perform as well as decision trees and be cheaper to run. Another way of thinking about this approach is that it tries to determine the most probable predictor for each response variable and then uses that probability to estimate the probability of the response variable within the new dataset (Bruce & Bruce, 2017). One disadvantage of this method is that if a new predictor is introduced, the method will assume zero probability and the results could be misleading.

Classification Model Recommendation

Based on the information provided for this analysis, it is recommended that the Random Forest method be used as a starting point. There are several benefits to using a tree-based modeling approach. One of the perks is the ability to take into account complex nuanced relationships that are not readily apparent. Furthermore, tree-based models are not as sensitive to outliers, which may be of value especially with limited information regarding potential consumers. However, once a model is constructed, it should be continually evaluated to ensure that its usefulness continues.

Conclusion

Three key clustering methods were investigated for this analysis: K-Means, PAM, and Hierarchical Clustering. Using the scree plot and the average silhouette width assessment, it was deemed that two clusters would be optimal. The K-Means clustering method was chosen for in-depth consumer profiling using consumer demographic information and consumer preferences. It was found that consumers preferring social media were in one cluster and consumers preferring entertainment apps were in another. Both clusters showed strong preference for iOS mobile devices (specifically iPhone and iPod Touch) as well as Android (as an OS). Finally, a brief overview of two different classification models (Random Forest and Naive Bayes) was provided to cultivate insight into how consumers could be classified with incomplete information sets. Ultimately, App Happy may want to conduct another survey that is more updated with better questions that help segment the consumers better in addition to more clearly defining what is ”social entertainment” to them.

References

Analytics, U. B. (n.d.). *Hierarchical cluster analysis.* `http://uc-r.github.io/hc_clustering`. (Accessed: 2018-01-20)

Bruce, P., & Bruce, A. (2017). *Practical statistics for data scientists*. Boston: O'Reilly.

Chapman, C., & Feit, E. M. (2015). *R for marketing research and analytics*. New York: Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. New York: Springer.

Izenman, A. J. (2013). *Modern multivariate statistical techniques regression, classification, and manifold learning*. New York: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). *An introduction to statistical learning with applications in r*. New York: Springer.

Appendix

EDA

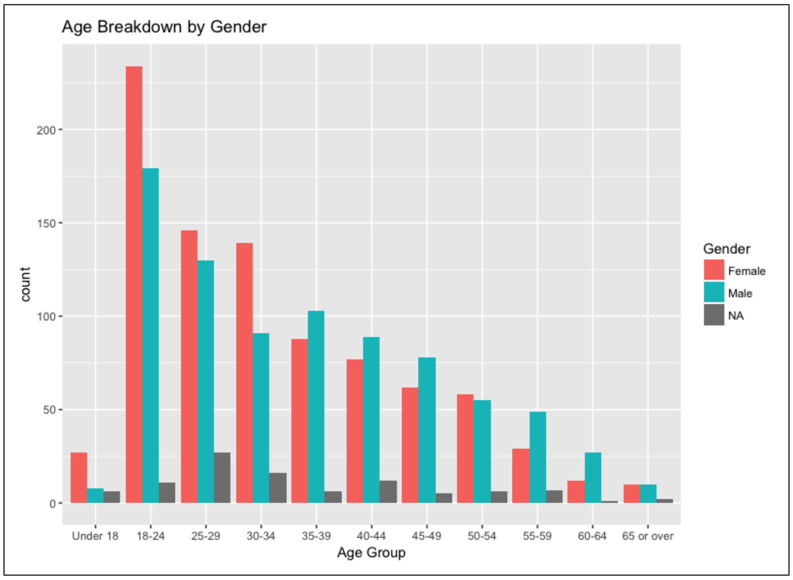

Figure 9.1: Histogram of Age by Gender

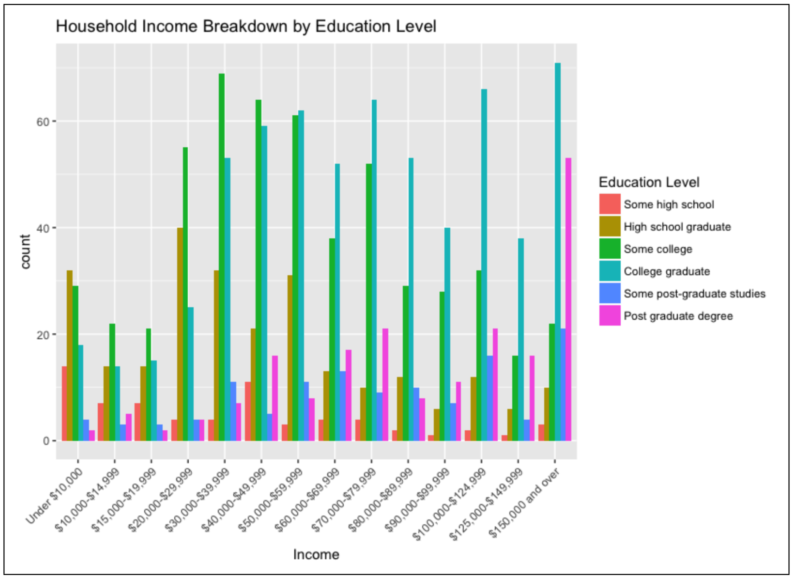

Figure 9.2: Histogram of Income by Education Level

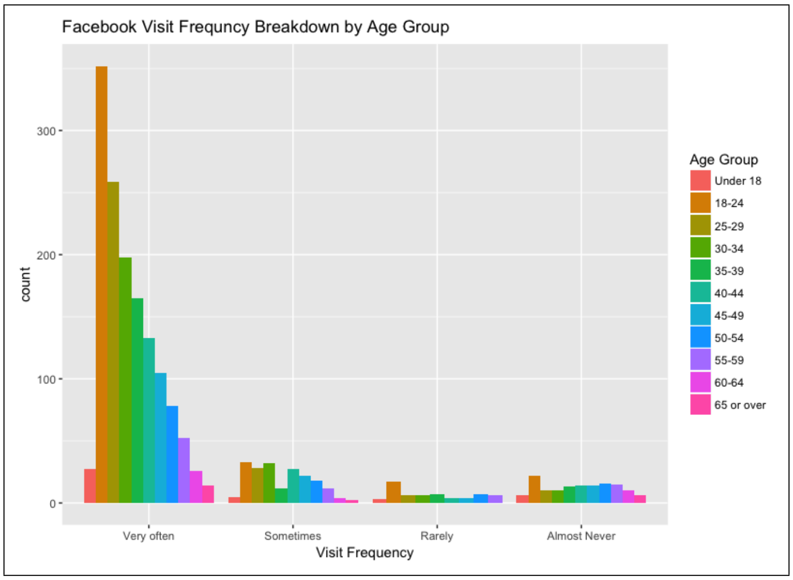

Figure 9.3: Histogram of Visiting Facebook Website by Age

PCA

Table 9.1: Principal Component Analysis Results - Top 15 Principal Components

principal_component	**std_dev**	**percent**	**cumulative_pct**
1	4.334	0.269	0.269
2	2.543	0.093	0.361
3	1.923	0.053	0.414
4	1.669	0.04	0.454
5	1.603	0.037	0.491
6	1.437	0.03	0.521
7	1.393	0.028	0.548
8	1.373	0.027	0.575
9	1.308	0.024	0.6
10	1.283	0.024	0.623
11	1.263	0.023	0.646
12	1.244	0.022	0.668
13	1.217	0.021	0.69
14	1.137	0.019	0.708
15	1.1	0.017	0.726

Table 9.2: Attitudinal Response Groupings

Grouping	Questions Within Grouping
Trendsetter	q25r2, q26r18, q26r7, q26r8, q26r10
Early Adopter	q24r1, q24r3, q25r5, q25r8, q25r9, q25r10, q26r9
Ease of Life	q24r5, q24r6, q25r12, q26r5
Entertainment	q24r7, q24r8
Late Adopter	q24r4, q24r9, q24r12, q25r11
Leader	q25r1, q25r4, q25r7
Shopper	q26r3, q26r4, q26r6, q26r12, q26r13
Social	q24r2, q24r10, q24r11, q25r3, q25r6, q25r11

Table 9.3: Grouped Responses Principal Components Analysis - Top 5 Principal Components

principle_component	std_dev	percent	cumulative_pct
1	2.462	0.411	0.411
2	1.619	0.178	0.589
3	1.141	0.088	0.677
4	1.009	0.069	0.746
5	0.859	0.05	0.796

Profiling

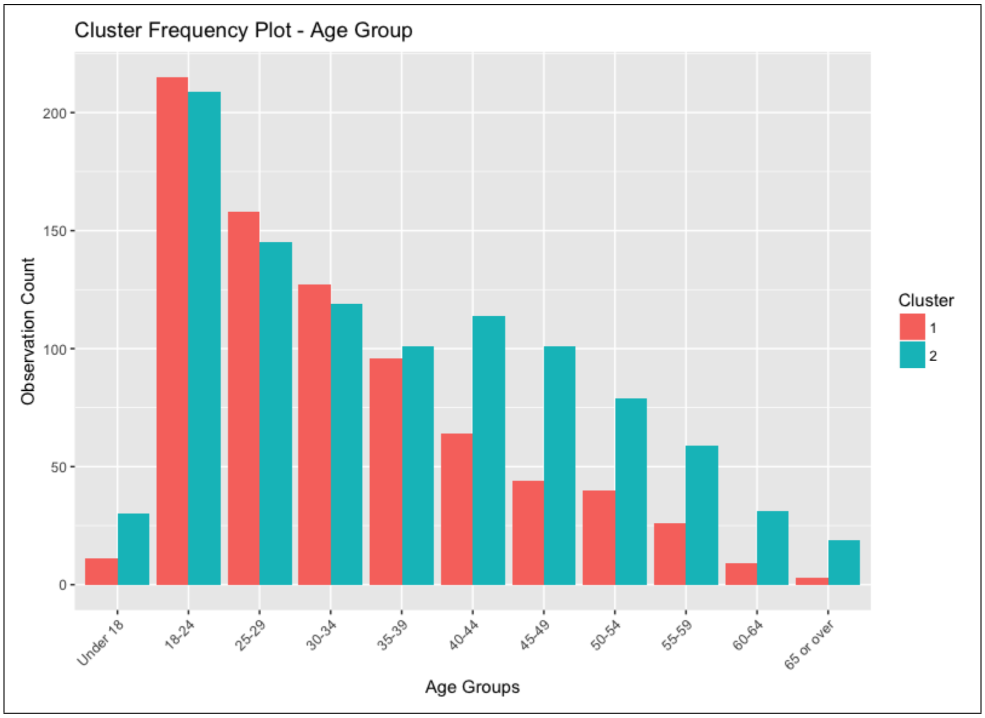

Figure 9.4: Cluster Breakout by Age Group

Table 9.4: Age Group Proportion Table by Cluster

	Cluster 1		**Cluster 2**	
Age Group	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
Under 18	11	1%	30	3%
18-24	215	27%	209	21%
25-29	158	20%	145	14%
30-34	127	16%	119	12%
35-39	96	12%	101	10%
40-44	64	8%	114	11%
45-49	44	6%	101	10%
50-54	40	5%	79	8%
55-59	26	3%	59	6%
60-64	9	1%	31	3%
65 or over	3	0%	19	2%

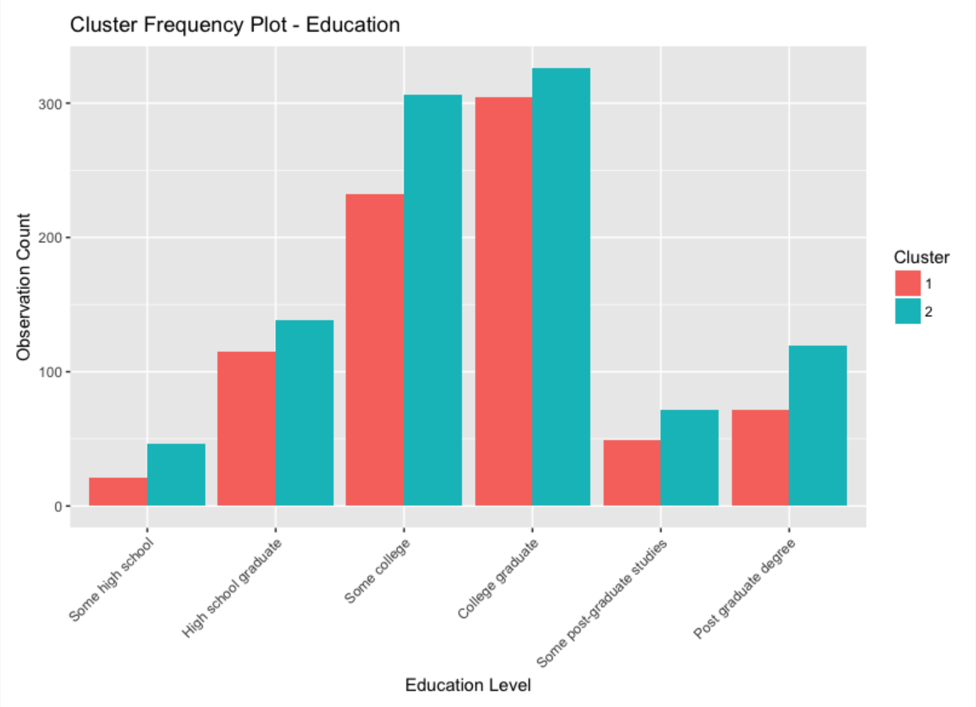

Figure 9.5: Cluster Breakout by Education Level

Table 9.5: Education Level Proportion Table by Cluster

	Cluster 1		**Cluster 2**	
Education Level	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
Some high school	21	3%	46	5%
High school graduate	115	15%	138	14%
Some college	232	29%	306	30%
College graduate	304	38%	326	32%
Some post-graduate studies	49	6%	72	7%
Post graduate degree	72	9%	119	12%

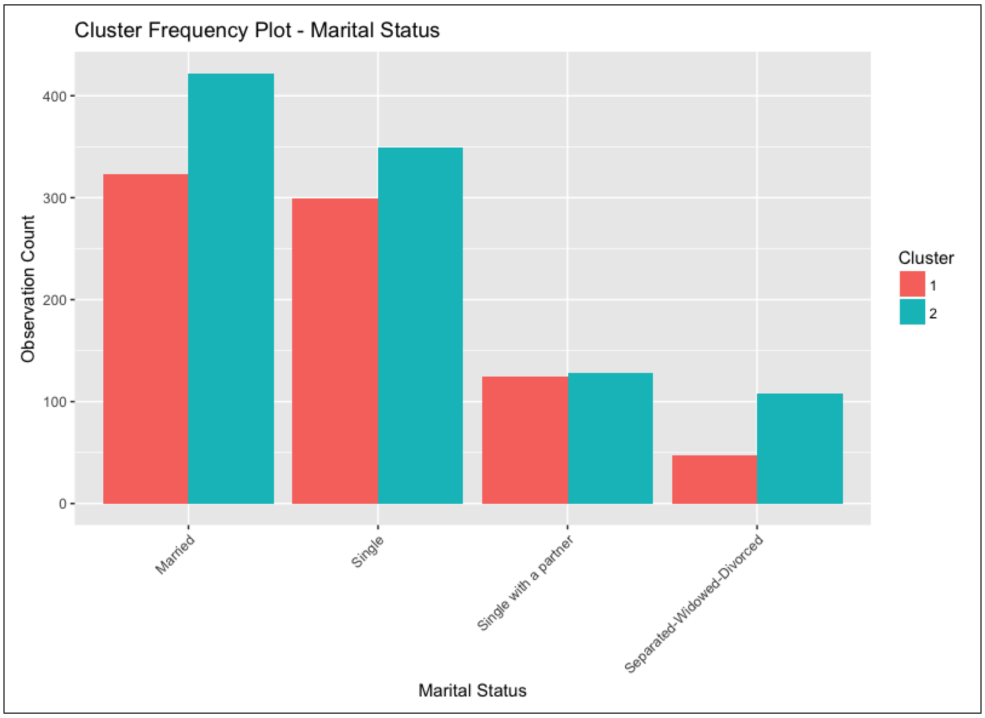

Figure 9.6: Cluster Breakout by Marital Status

Table 9.6: Marital Status Proportion Table by Cluster

	Cluster 1		Cluster 2	
Marital Status	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
Married	323	41%	422	42%
Single	299	38%	349	35%
Single with a partner	124	16%	128	13%
Separated-Widowed-Divorced	47	6%	108	11%

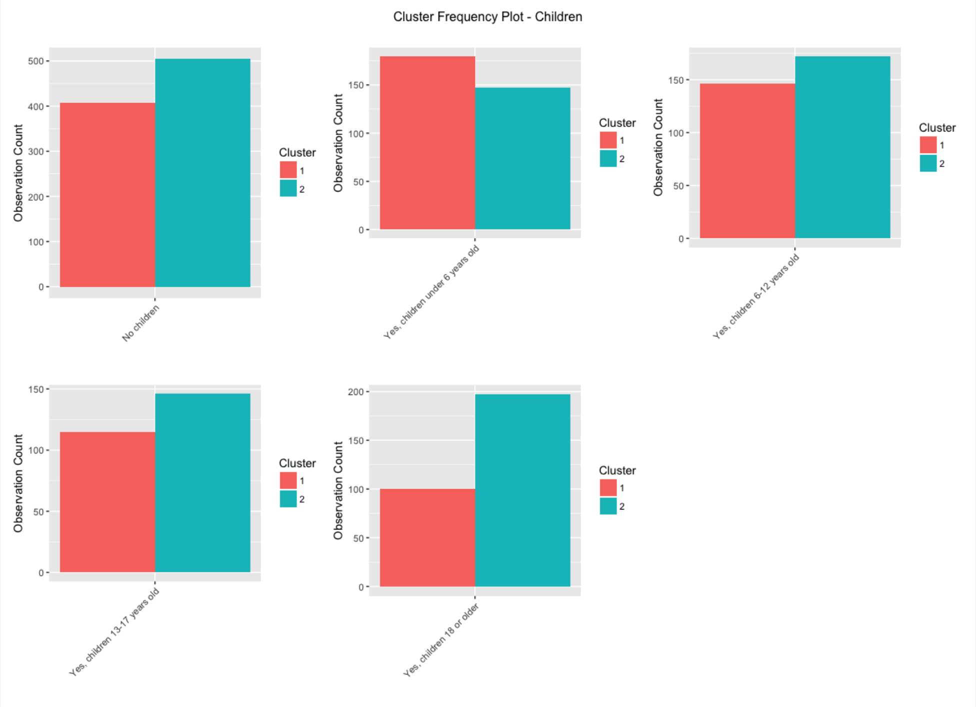

Figure 9.7: Cluster Breakout by Number of Children

Table 9.7: Children Age Group Proportion Table by Cluster

	Cluster 1		**Cluster 2**	
Children	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
No Children	407	43%	505	43%
Yes, children under 6 years old	180	19%	147	13%
Yes, children 6-12 years old	146	15%	172	15%
Yes, children 13-17 years old	115	12%	146	13%
Yes, children 18 or older	100	11%	197	17%

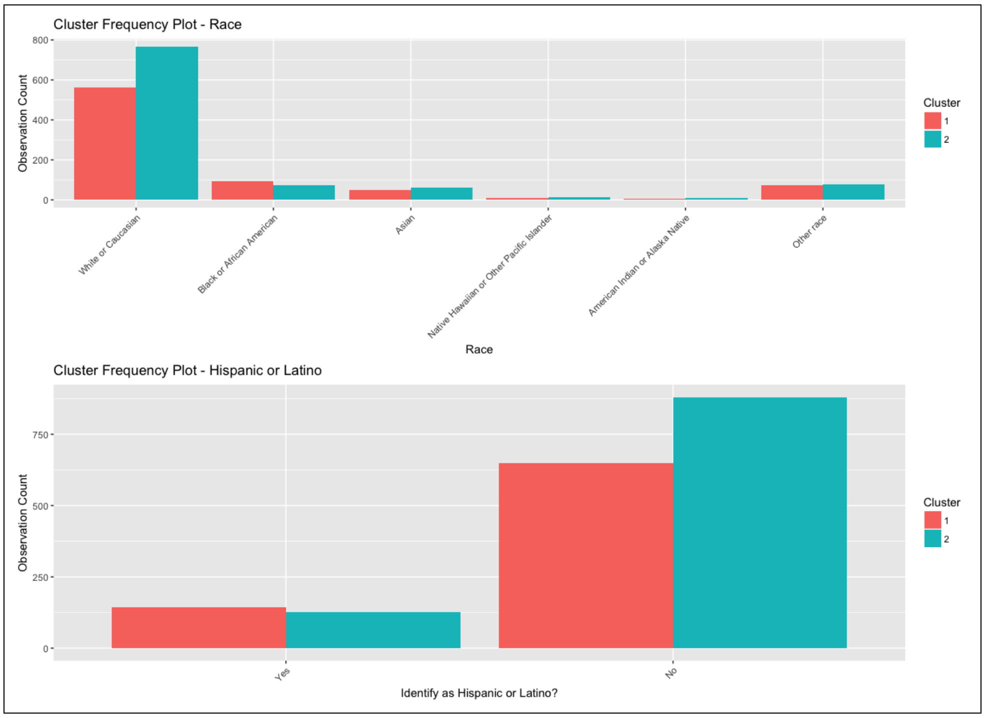

Figure 9.8: Cluster Breakout by Race

Table 9.8: Race Proportion Table by Cluster

	Cluster 1		Cluster 2	
Race	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
White or Caucasian	562	71%	768	76%
Black or African American	93	12%	75	7%
Asian	50	6%	61	6%
Native Hawaiian or Other Pacific Islander	10	1%	14	1%
American Indian or Alaska Native	6	1%	10	1%
Other race	72	9%	79	8%

Table 9.9: Latino Identification Proportion Table by Cluster

	Cluster 1		Cluster 2	
Identify as Latino	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
Yes	144	18%	127	13%
No	649	82%	880	87%

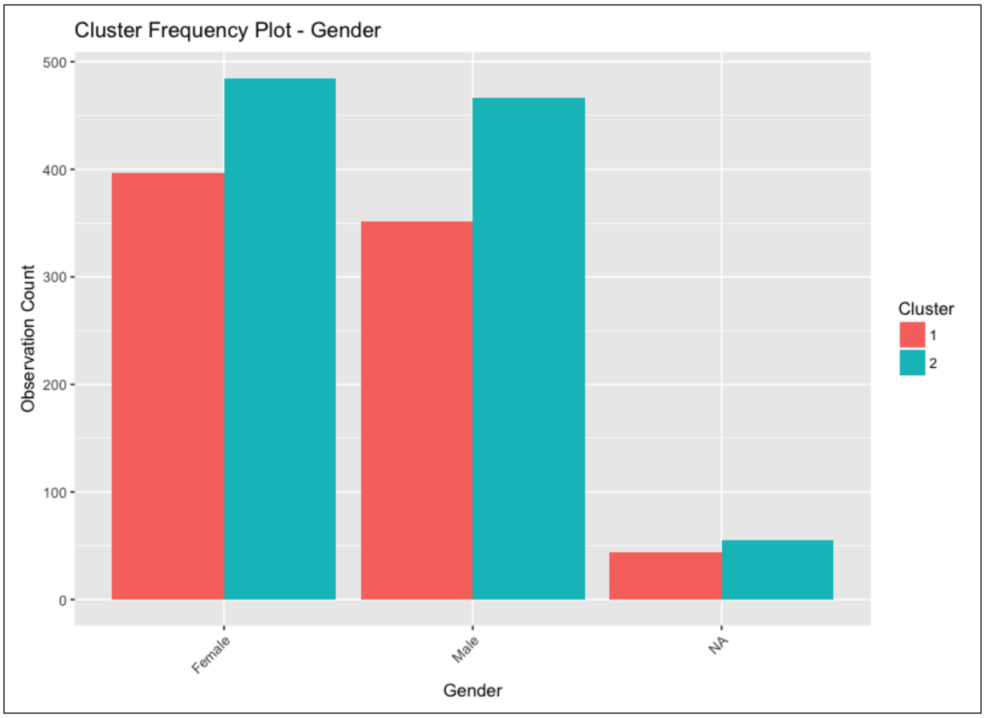

Figure 9.9: Cluster Breakout by Gender

Table 9.10: Gender Proportion Table by Cluster

	Cluster 1		**Cluster 2**	
Gender	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
Female	397	50%	485	48%
Male	352	44%	467	46%
NA	44	6%	55	5%

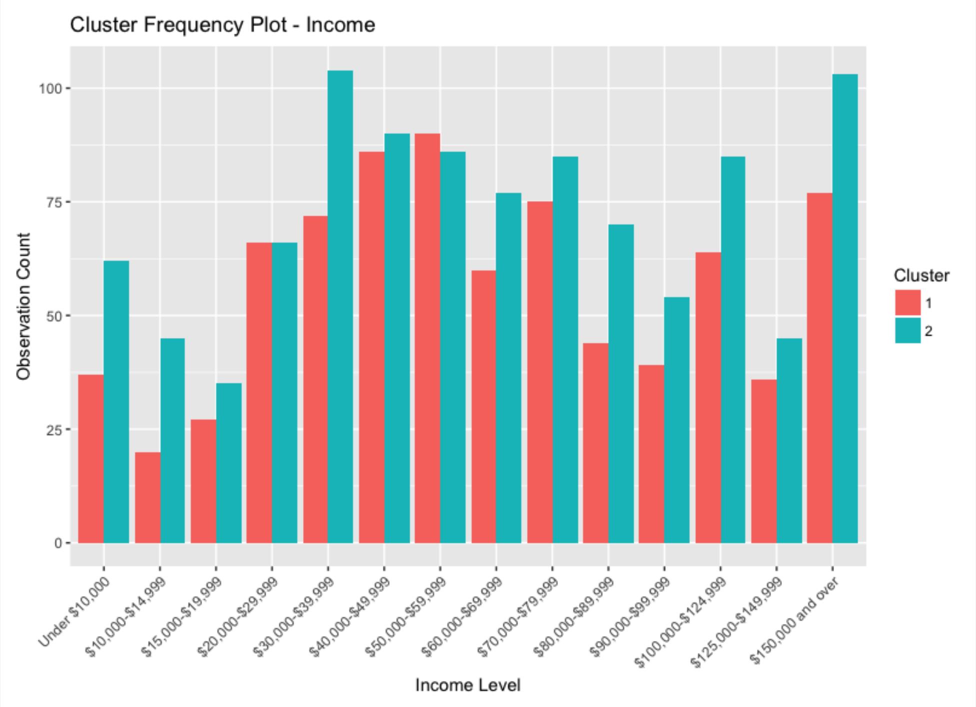

Figure 9.10: Cluster Breakout by Income

Table 9.11: Income Proportion Table by Cluster

	Cluster 1		**Cluster 2**	
Income	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
Under $10,000	37	5%	62	6%
$10,000-$14,999	20	3%	45	4%
$15,000-$19,999	27	3%	35	3%
$20,000-$29,999	66	8%	66	7%
$30,000-$39,999	72	9%	104	10%
$40,000-$49,999	86	11%	90	9%
$50,000-$59,999	90	11%	86	9%
$60,000-$69,999	60	8%	77	8%
$70,000-$79,999	75	9%	85	8%
$80,000-$89,999	44	6%	70	7%
$90,000-$99,999	39	5%	54	5%
$100,000-$124,999	64	8%	85	8%
$125,000-$149,999	36	5%	45	4%
$150,000 and over	77	10%	103	10%

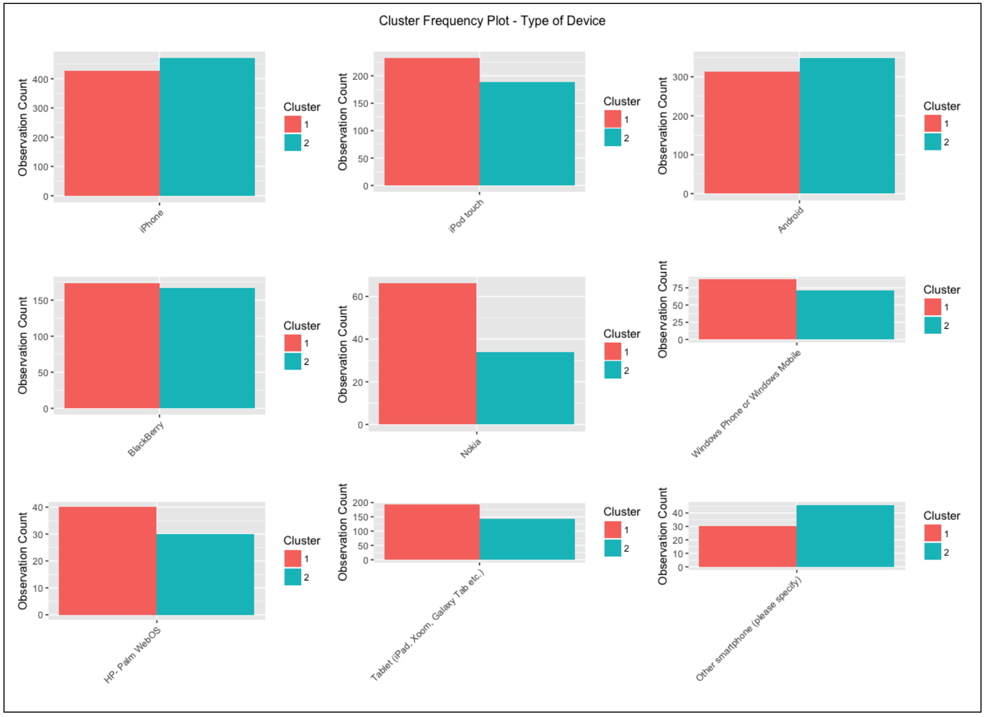

Figure 9.11: Cluster Breakout by Type of Device Owned

Table 9.12: Type of Device Proportion Table

	Cluster 1		Cluster 2	
Type of Device	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
iPhone	426	27%	470	31%
iPod Touch	233	15%	189	13%
Android	314	20%	348	23%
BlackBerry	174	11%	167	11%
Nokia	66	4%	34	2%
Windows Phone/Mobile	87	6%	71	5%
HP/Palm WebOS	40	3%	30	2%
Tablet	193	12%	144	10%
Other	30	2%	46	3%

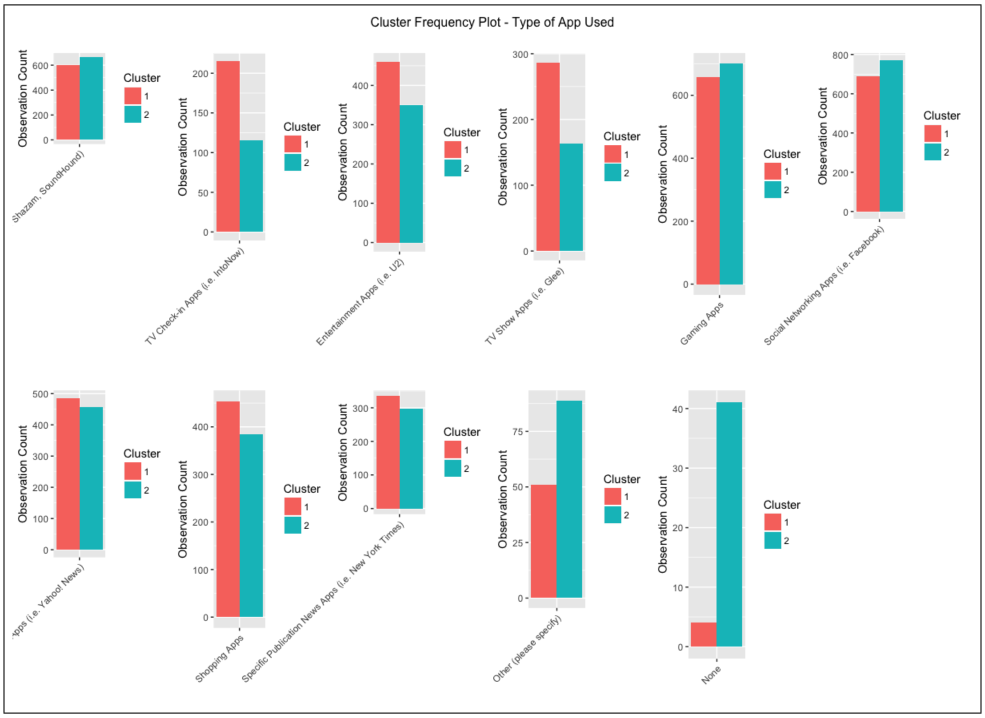

Figure 9.12: Cluster Breakout by Type of App Used

Table 9.13: Type of App Proportion Table by Cluster

	Cluster 1		Cluster 2	
Type of Apps Used	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
Music Identification	601	14%	666	17%
TV Check-In	215	5%	115	3%
Entertainment	460	11%	350	9%
TV Show	287	7%	163	4%
Gaming	658	16%	700	17%
Social Networking	688	16%	770	19%
General News	486	11%	458	11%
Shopping	454	11%	384	10%
Specific Publication News	335	8%	297	7%
Other	51	1%	89	2%
None	4	0%	41	1%

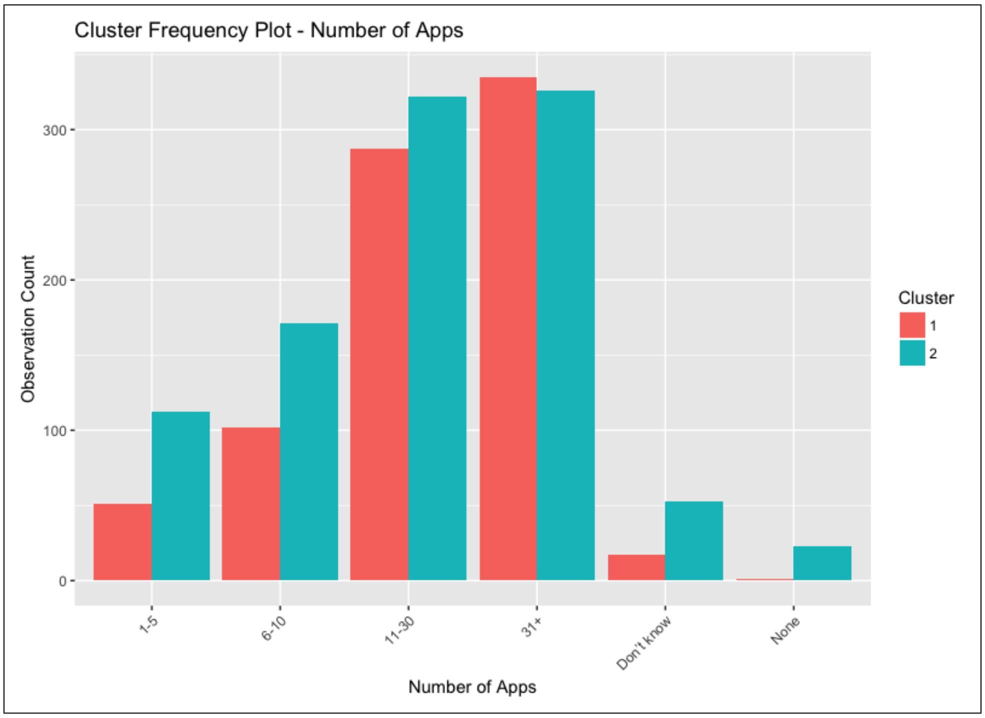

Figure 9.13: Cluster Breakout by Number of Apps

Table 9.14: Number of Apps Proportion Table by Cluster

	Cluster 1		**Cluster 2**	
Number of Apps	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
1-5	51	6%	112	11%
6-10	102	13%	171	17%
11-30	287	36%	322	32%
31+	335	42%	326	32%
Dont know	17	2%	53	5%
None	1	0%	23	2%

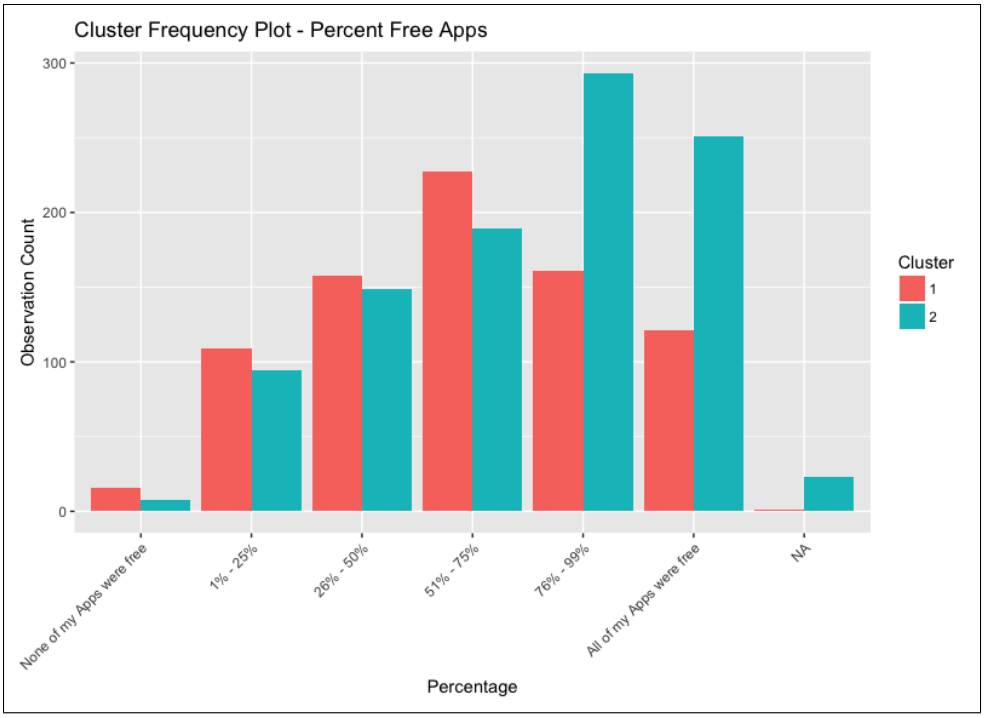

Figure 9.14: Cluster Breakout by Percentage of Free Apps

Table 9.15: Percentage of Free Apps Proprtion Table by Cluster

	Cluster 1		**Cluster 2**	
Percentage of Free Apps	**Number of Observations**	**Percentage**	**Number of Observations**	**Percentage**
None of my Apps were free	16	2%	8	1%
1% - 25%	109	14%	94	9%
26% - 50%	158	20%	149	15%
51% - 75%	227	29%	189	19%
76% - 99%	161	20%	293	29%
All of my Apps were free	121	15%	251	25%
NA	1	0%	23	2%

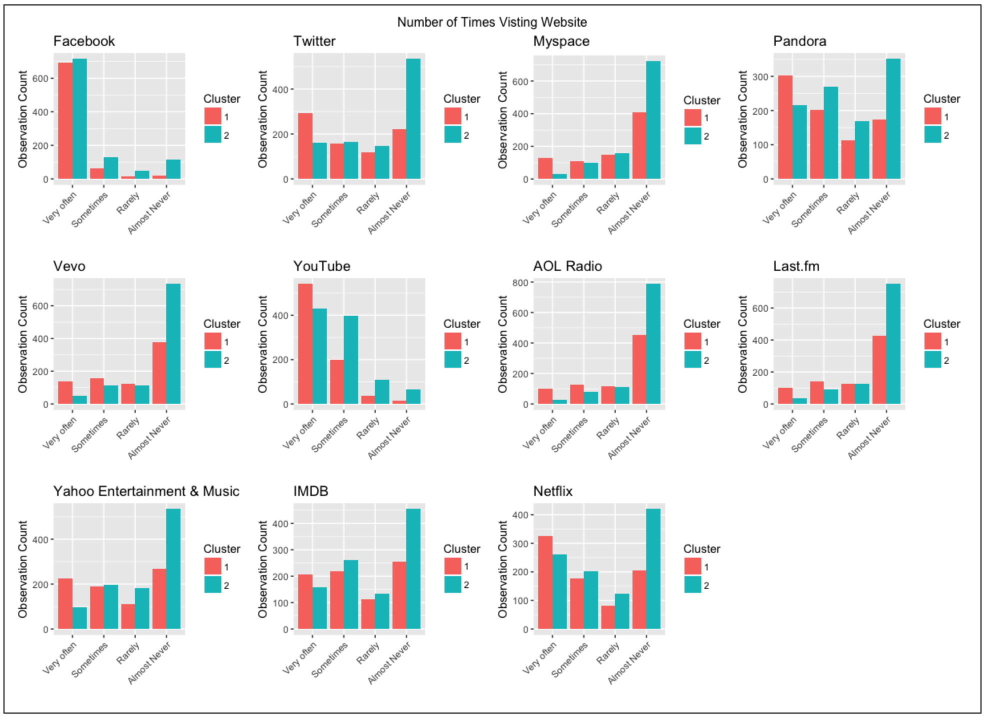

Figure 9.15: Cluster Breakout of Visiting Websites

R Code

```
# load libraries ----
library(tidyverse)
library(skimr) # disables from dplyr: contains, ends_with, everything, matches, num_range,
    one_of, starts_with
library(corrplot)
library(factoextra)
library(cluster)
library(NbClust)
library(mclust)
library(fpc)
library(gridExtra)

# load Rdata file ----

s1 <- load('/Users/nikhil/Documents/Northwestern/PREDICT450/Assignments/Solo1/data/
    apphappyData.RData')

#tblPath <- '/Users/nikhil/Documents/Northwestern/PREDICT450/Assignments/Solo1/paper/tbls/'

# preview data frames ----

glimpse(apphappy.3.labs.frame)
glimpse(apphappy.3.num.frame)

# summary on the data frames ----
summary(apphappy.3.num.frame)
summary(apphappy.3.labs.frame)

skim(apphappy.3.num.frame)

# https://ropensci.org/blog/2017/07/11/skimr/
apphappy.3.num.frame %>%
skim() %>%
filter(stat == ’missing’) %>%
arrange(desc(value)) %>%
print(n=2000)

# missing data check ----
apphappy.3.num.frame %>%
map_df(function(x) sum(is.na(x))) %>%
gather(key = ’variable’, value = ’missing’) %>%
arrange(desc(missing))

apphappy.3.labs.frame %>%
map_df(function(x) sum(is.na(x))) %>%
gather(key = ’variable’, value = ’missing’) %>%
arrange(desc(missing))

# Quality Checks ----

## quality check Q1 (numerical)

apphappy.3.num.frame %>%
select(caseID, q1) %>%
summary

apphappy.3.labs.frame %>%
select(caseID, q1) %>%
filter(str_detect(q1, ’65’))


# create a data frame for attitudinal questions ----
# attitudinal questions seem to be q24, q25, and q26 as they pertain to a person’s
    perspective/attitude

att.num.df <- apphappy.3.num.frame %>%
select(
caseID,
starts_with(’q24’),
starts_with(’q25’),
starts_with(’q26’)
)

att.lbl.df <- apphappy.3.labs.frame %>%
select(
caseID,
starts_with(’q24’),
starts_with(’q25’),
starts_with(’q26’)
)

# custom summary for the attitudinal stuff
att.num.df %>%
select(-caseID) %>%
skim %>%
filter(stat != 'hist') %>%
select(-type, -level, -formatted) %>%
spread(key = stat, value = value) %>%
select(-complete, -n) %>%
transmute(
variable,
missing,
avg = round(mean,3),
sd = round(sd,3),
median,
min = p0,
max = p100,
p25,
p75
) %>%
write_csv(path = '/Users/nikhil/Documents/Northwestern/PREDICT450/Assignments/Solo1/paper/
    tbls/summary_att.csv')

# EDA - all data ----
apphappy.3.labs.frame %>%
ggplot(aes(x = q1)) +
geom_bar(stat = 'count', aes(fill = q57), position = 'dodge') +
scale_fill_discrete(name = 'Gender') +
ggtitle('Age Breakdown by Gender') +
labs(x = 'Age Group')

apphappy.3.labs.frame %>%
ggplot(aes(x = q56)) +
geom_bar(stat = 'count', aes(fill = q57), position = 'dodge') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = 'Household Income') +
scale_fill_discrete(name = "Gender")+
ggtitle('Household Income Frequency Count by Gender')

# breakdown of male vs. female respondents
apphappy.3.labs.frame %>%
ggplot(aes(x = q57)) +
geom_bar(stat = 'count') +
labs(x = 'Gender') +
ggtitle('Histogram of Gender')

# breakdown by race
apphappy.3.labs.frame %>%
ggplot(aes(x = q54)) +
geom_bar(stat = 'count', aes(fill = q57), position = 'dodge') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = 'Race') +
scale_fill_discrete(name = "Gender")+
ggtitle('Race Breakdown by Gender')

apphappy.3.labs.frame %>%
ggplot(aes(x = q55)) +
geom_bar(stat = 'count', aes(fill = q57), position = 'dodge') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = 'Identify as Latino?') +
scale_fill_discrete(name = "Gender")+
ggtitle('Identify as Latino Breakdown by Gender')

apphappy.3.labs.frame %>%
ggplot(aes(x = q54)) +
geom_bar(stat = ’count’, aes(fill = q57), position = ’dodge’) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Race’) +
scale_fill_discrete(name = "Gender")+
ggtitle(’Race Breakdown by Latino Identification & Gender’) +
facet_grid(q55 ~ .)

# age breakdown by race
apphappy.3.labs.frame %>%
ggplot(aes(x = q1)) +
geom_bar(stat = ’count’, aes(fill = q54), position = ’dodge’) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Race’) +
scale_fill_discrete(name = "Race")+
ggtitle(’Age Breakdown by Race’)

# household income breakdown by race
apphappy.3.labs.frame %>%
ggplot(aes(x = q56)) +
geom_bar(stat = ’count’, aes(fill = q54), position = ’dodge’) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Income’) +
scale_fill_discrete(name = "Race")+
ggtitle(’Household Income Breakdown by Race’)

# household income breakdown by marital status
apphappy.3.labs.frame %>%
ggplot(aes(x = q56)) +
geom_bar(stat = ’count’, aes(fill = q49), position = ’dodge’) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Income’) +
scale_fill_discrete(name = "Marital Status")+
ggtitle(’Household Income Breakdown by Marital Status’)

# household income breakdown by marital status & gender
apphappy.3.labs.frame %>%
ggplot(aes(x = q56)) +
geom_bar(stat = ’count’, aes(fill = q49), position = ’dodge’) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Income’) +
scale_fill_discrete(name = "Marital Status")+
ggtitle(’Household Income Breakdown by Marital Status & Gender’) +
facet_grid(q57 ~ .)

# household income breakdown by education level
apphappy.3.labs.frame %>%
ggplot(aes(x = q56)) +
geom_bar(stat = ’count’, aes(fill = q48), position = ’dodge’) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Income’) +
scale_fill_discrete(name = "Education Level")+
ggtitle(’Household Income Breakdown by Education Level’)

# look at the type of apps
apphappy.3.num.frame %>%
select(starts_with(’q4r’)) %>%
map_df(function(x) sum(x)) %>%
gather(key = ’q4’, value = ’count’) %>%
ggplot(aes(x = reorder(as.factor(q4), -count), y = count)) +
geom_col() +
labs(x = 'Question 4 Response Types')
#scale_x_discrete(limits = c('q4r1','q4r2','q4r3','q4r4','q4r5','q4r6','q4r7','q4r8','q4r9
    ','q4r10','q4r11'))

# look at the type of platforms
apphappy.3.num.frame %>%
select(starts_with('q2r')) %>%
map_df(function(x) sum(x)) %>%
gather(key = 'q2', value = 'count') %>%
ggplot(aes(x = q2, y = count)) +
geom_col() +
scale_x_discrete(limits = c('q2r1','q2r2','q2r3','q2r4','q2r5','q2r6','q2r7','q2r8','q2r9','
    q2r10'))


# look at the social sites


apphappy.3.labs.frame %>%
select(q13r1) %>%
group_by(q13r1) %>%
summarize(
obs.count = n()
) %>%
ungroup


apphappy.3.labs.frame %>%
select(q13r1) %>%
group_by(q13r1) %>%
summarize(
obs.count = n()
) %>%
ungroup %>%
ggplot(aes(x = q13r1, y = obs.count)) +
geom_col() +
labs(x = 'Visit Frequency')

getSiteFrequency <- function(df, site, sitename) {
site <- enquo(site)
df %>%
select(!!site) %>%
group_by(!!site) %>%
ggplot(aes_string(x = rlang::quo_text(site))) +
geom_bar(stat = 'count') +
labs(x = 'Visit Frequency') +
ggtitle(paste0(sitename,' Visit Frequency Plot'))
}

getSiteFrequency(apphappy.3.labs.frame, q13r1, 'Facebook')
getSiteFrequency(apphappy.3.labs.frame, q13r2, 'Twitter')
getSiteFrequency(apphappy.3.labs.frame, q13r3, 'Myspace')
getSiteFrequency(apphappy.3.labs.frame, q13r4, 'Pandora')
getSiteFrequency(apphappy.3.labs.frame, q13r5, 'Vevo')
getSiteFrequency(apphappy.3.labs.frame, q13r6, 'YouTube')
getSiteFrequency(apphappy.3.labs.frame, q13r7, 'AOL Radio')
getSiteFrequency(apphappy.3.labs.frame, q13r8, 'Last.fm')
getSiteFrequency(apphappy.3.labs.frame, q13r9, ’Yahoo Music’)
getSiteFrequency(apphappy.3.labs.frame, q13r10, ’IMDB’)
getSiteFrequency(apphappy.3.labs.frame, q13r11, ’LinkedIn’)
getSiteFrequency(apphappy.3.labs.frame, q13r12, ’Netflix’)


apphappy.3.labs.frame %>%
select(q13r1, q57) %>%
ggplot(aes(x = q13r1)) +
geom_bar(stat = ’count’, aes(fill = q57), position = ’dodge’) +
labs(x = ’Visit Frequency’) +
scale_fill_discrete(name = ’Gender’) +
ggtitle(’Facebook Visit Frequency Breakdown by Gender’)


plotSite <- function(df, site, col2, sitename, col2name) {
site <- enquo(site)
col2 <- enquo(col2)
df %>%
select(!!site, !!col2) %>%
ggplot(aes_string(x = rlang::quo_text(site))) +
geom_bar(stat = ’count’, aes_string(fill = rlang::quo_text(col2)), position = ’dodge’) +
labs(x = ’Visit Frequency’) +
ggtitle(paste0(sitename,’ Visit Frequncy Breakdown by ’, col2name)) +
scale_fill_discrete(name = col2name)
}

plotSite(apphappy.3.labs.frame, q13r1, q57, ’Facebook’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r2, q57, ’Twitter’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r3, q57, ’Myspace’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r4, q57, ’Pandora’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r5, q57, ’Vevo’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r6, q57, ’YouTube’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r7, q57, ’AOL Radio’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r8, q57, ’Last.fm’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r9, q57, ’Yahoo Music’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r10, q57, ’IMDB’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r11, q57, ’LinkedIn’, ’Gender’)
plotSite(apphappy.3.labs.frame, q13r12, q57, ’Netflix’, ’Gender’)

plotSite(apphappy.3.labs.frame, q13r1, q1, ’Facebook’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r2, q1, ’Twitter’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r3, q1, ’Myspace’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r4, q1, ’Pandora’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r5, q1, ’Vevo’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r6, q1, ’YouTube’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r7, q1, ’AOL Radio’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r8, q1, ’Last.fm’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r9, q1, ’Yahoo Music’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r10, q1, ’IMDB’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r11, q1, ’LinkedIn’, ’Age Group’)
plotSite(apphappy.3.labs.frame, q13r12, q1, ’Netflix’, ’Age Group’)



# EDA - attitudinal data
apphappy.3.num.frame %>%
select(q1) %>%
group_by(q1) %>%
tally %>%
ggplot(aes(x = q1, y = n)) +
geom_bar(stat = ’identity’) +
ggtitle(’Frequency Histogram for Q1’) +
labs(x = ’Q1’, y = ’Frequency’)


apphappy.3.num.frame %>%
select(q1) %>%
ggplot(aes(x = q1)) +
geom_histogram() +
ggtitle(’Frequency Histogram for Q1 Numerical Responses’) +
labs(x = ’Q1 Response’, y = ’Frequency’)

getHistogram <- function(cname) {
cname <- enquo(cname)
apphappy.3.num.frame %>%
select(!!cname) %>%
ggplot(aes_string(x = rlang::quo_text(cname))) +
geom_histogram(stat = ’count’) +
ggtitle(’Frequency Histogram’) +
labs(x = ’Q1 Response’, y = ’Frequency’)
}
getHistogram(q1)
getHistogram(q5r1)


# EDA on numerical frame only

getGridFreqCounts <- function(cname) {
cname <- enquo(cname)
att.num.df %>%
select(starts_with(!!cname)) %>%
gather(key = ’variable’, value = ’observation’) %>%
group_by(variable, observation) %>%
summarize(cnt = n()) %>%
ungroup %>%
spread(key = variable, value = cnt)
}
getGridFreqCounts(’q24’)
getGridFreqCounts(’q25’)
getGridFreqCounts(’q26’)


att.num.df %>%
select(q24r1, q25r1) %>%
group_by(q24r1, q25r1) %>%
summarize(
cnt.obs = n()
) %>%
ungroup %>%
spread(key = q25r1, value = cnt.obs, fill = 0)

getGroupGrid <- function(df, cname1, cname2) {
cname1 <- enquo(cname1)
cname2 <- enquo(cname2)
df %>%
select(!!cname1, !!cname2) %>%
rename(c1 = !!cname1, c2 = !!cname2) %>%
group_by(c1,c2) %>%
summarize(
cnt.obs = n()
) %>%
ungroup %>%
spread(key = c2, value = cnt.obs, fill = 0)
}
getGroupGrid(att.num.df, 'q24r1','q25r1')
getGroupGrid(att.num.df, 'q24r2','q25r2')



att.num.df %>%
select(q24r1, q25r1) %>%
group_by(q24r1, q25r1) %>%
summarize(
cnt.obs = n()
) %>%
ungroup %>%
ggplot(aes(x = q24r1, y = cnt.obs, fill= q25r1)) +
geom_bar(stat = 'identity')



att.num.df %>%
select(starts_with('q24')) %>%
gather(key = 'variable', value = 'observation') %>%
group_by(variable, observation) %>%
summarize(cnt2 = n()) %>%
ungroup %>%
ggplot(aes(x = observation, y = cnt2, fill = variable)) +
geom_bar(stat = 'identity') +
labs(x = 'Numerical Response Value', y = 'Frequency') +
ggtitle('Stacked Bar Chart for Numerical Response for Q24')

att.num.df %>%
select(starts_with('q25')) %>%
gather(key = 'variable', value = 'observation') %>%
group_by(variable, observation) %>%
summarize(cnt2 = n()) %>%
ungroup %>%
ggplot(aes(x = observation, y = cnt2, fill = variable)) +
geom_bar(stat = 'identity') +
labs(x = 'Numerical Response Value', y = 'Frequency') +
ggtitle('Stacked Bar Chart for Numerical Response for Q25')

att.num.df %>%
select(starts_with('q26')) %>%
gather(key = 'variable', value = 'observation') %>%
group_by(variable, observation) %>%
summarize(cnt2 = n()) %>%
ungroup %>%
ggplot(aes(x = observation, y = cnt2, fill = variable)) +
geom_bar(stat = 'identity') +
labs(x = 'Numerical Response Value', y = 'Frequency') +
ggtitle('Stacked Bar Chart for Numerical Response for Q26')

getStackedBarChart <- function(cname) {
cname2 <- enquo(cname)
```

```
att.num.df %>%
select(starts_with(!!cname2)) %>%
gather(key = 'variable', value = 'observation') %>%
group_by(variable, observation) %>%
summarize(cnt2 = n()) %>%
ungroup %>%
ggplot(aes(x = observation, y = cnt2, fill = variable)) +
geom_bar(stat = 'identity') +
labs(x = 'Numerical Response Value', y = 'Frequency') +
ggtitle(paste0('Stacked Bar Chart for Numerical Response for ',toupper(cname)))
}

getStackedBarChart('q24')
getStackedBarChart('q25')
getStackedBarChart('q26')




# lets do a correlation study ----

numcor <- cor(att.num.df)# %>% select(starts_with('q24'), starts_with('q25')))
corrplot(
numcor,
method = 'color',
order = 'hclust',
addCoef.col = 'red',
tl.col = 'black',
addrect = 5,
tl.cex = 0.6,
number.cex = 0.6
)




# PCA ----
# PCA on the original numerical dataset (cluster size does not matter)
pca <- prcomp(att.num.df %>% select(-caseID))
summary(pca)

tidy(pca, matrix = 'd') %>%
transmute(
principle_component = PC,
std_dev = round(std.dev,3),
percent = round(percent,3),
cumulative_pct = round(cumulative,3)
) %>%
top_n(n=15, wt = percent) %>%
write_csv(path = '/Users/nikhil/Documents/Northwestern/PREDICT450/Assignments/Solo1/paper/
    tbls/pca.csv')
# note how 14 principle components are needed to get about 70% of the variation to be
    explained.

#plot pca
fviz_nbclust(att.num.df %>% select(-caseID), method = 'wss', k.max = 15, FUNcluster = kmeans
    ) #elbow
fviz_eig(pca)
fviz_pca_ind(pca)
```

```
# note the big blob with very little distinction

# next bet is to combine the attitudinal questions
att.num.df2 <- att.num.df %>%
transmute(
caseID,
trendsetter = (q25r2 + q26r18 + q26r7 + q26r8 + q26r10) / 5,
early.adopter = (q24r1 + q24r3 + q25r5 + q25r8 + q25r9 + q25r10 + q26r9) / 7,
ease.of.life = (q24r5 + q24r6 + q25r12 + q26r5) / 4,
entertainment = (q24r7 + q24r8) / 2,
late.adopter = (q24r4 + q24r9 + q24r12 + q25r11) / 4,
leader = (q25r1 + q25r4 + q25r7) / 3,
shopper = (q26r3 + q26r4 + q26r6 + q26r12 + q26r13) / 5,
social = (q24r2 + q24r10 + q24r11 + q25r3 + q25r6 + q25r11) / 6
)

# PCA on the new data frame
pca2 <- prcomp(att.num.df2 %>% select(-caseID))
summary(pca2)

tidy(pca2, matrix = 'd') %>%
transmute(
principle_component = PC,
std_dev = round(std.dev,3),
percent = round(percent,3),
cumulative_pct = round(cumulative,3)
) %>%
top_n(n=5, wt = percent) %>%
write_csv(path = '/Users/nikhil/Documents/Northwestern/PREDICT450/Assignments/Solo1/paper/
    tbls/pca2.csv')

fviz_nbclust(att.num.df2 %>% select(-caseID), method = 'wss', k.max = 15, FUNcluster =
    kmeans) #elbow
fviz_nbclust(att.num.df2 %>% select(-caseID), method = 'silhouette', k.max = 15, FUNcluster
    = kmeans)



#plot pca
fviz_eig(pca2)
fviz_pca_ind(pca2)
# note how 3 principle components can now explain almost 70% of the variance
# still not very clear on the image tho



# k means ----
#### scaling is not necessary since all of the questions analyzed for partitioning are on
    the same scale
# att.num.df <- att.num.df %>%
#   select(-caseID) %>%
#   map_df(function(x) scale(x, center = T, scale = T)) %>%
#   add_column(caseID = att.num.df$caseID)

# kmeans on full dataset ----
## method 1: use scree plot to determine correct (code has been provided by prof)
wssplot <- function(att.num.df, nc=15, seed=1234) {
wss <- (nrow(att.num.df)-1)*sum(apply(att.num.df,2,var))
for (i in 2:nc) {
```

```
set.seed(seed)
wss[i] <- sum(kmeans(att.num.df, centers=i)$withinss)}
plot(1:nc, wss, type="b", xlab="Number of Clusters",
ylab="Within groups sum of squares", main = 'Scree Plot')}

wssplot(att.num.df %>% select(-caseID))

## method 2: custom function that loops through to find best cluster using R2
for (i in 1:15) {
csize = i + 1
set.seed(2018)
kc <- kmeans(att.num.df %>% select(-caseID), centers = csize)
r2 <- kc$betweenss / kc$totss
print(paste0('cluster size: ', csize, ' r2: ',round(r2,3)))
}

## method 3: use avg. silhouette width to determine best cluster size
df <- tibble(x=1,y=0)
for (i in 2:15) {
set.seed(2018)
kc <- kmeans(att.num.df %>% select(-caseID), centers = i)
dissE <- daisy(att.num.df %>% select(-caseID))
dE2   <- dissE^2
sk2   <- silhouette(kc$cluster, dE2)
df <- add_row(df, x=i, y=round(summary(sk2)$avg.width,3))
print(paste0('cluster size: ', i, ' avg. silhouette width: ', round(summary(sk2)$avg.width
    ,3)))
if(i == 15) {
print(
df %>%
ggplot(aes(x = x, y = y)) +
geom_line(col='blue') +
geom_point(col='red') +
ggtitle('Silhouette Average Widths') +
xlab('Cluster Size') +
ylab('Average Width')
)
}
}

## method 4: use NbClust to go through 26 indices to determine which cluster size is best
#### I use Ratkowsky due to the white-paper I found
best.cluster.size <- NbClust(att.num.df %>% select(-caseID), distance = 'euclidean', method
    = 'kmeans', min.nc = 2, max.nc = 10)
fviz_nbclust(best.cluster.size)

# method 5: visual graphs
fviz_nbclust(att.num.df %>% select(-caseID), method = 'wss', k.max = 15, FUNcluster = kmeans
    ) #elbow
fviz_nbclust(att.num.df %>% select(-caseID), method = 'silhouette', k.max = 15, FUNcluster =
     kmeans)
fviz_nbclust(att.num.df %>% select(-caseID), method = 'silhouette', k.max = 15, FUNcluster =
     pam)

# method 6: GAP statistic
clus.gap <- clusGap(att.num.df %>% select(-caseID), FUNcluster = kmeans, K.max = 15, B =
    100)
print(clus.gap, method = 'firstmax')
fviz_gap_stat(clus.gap)
```

```
print(clus.gap, method = 'Tibs2001SEmax')
fviz_gap_stat(clus.gap, maxSE = list(method = "Tibs2001SEmax"))

# choose 2 clusters
set.seed(2018)
kclust.2c <- kmeans(att.num.df %>% select(-caseID), centers = 2)
kclust.2c
kclust.2c$centers
kclust.2c$cluster

fviz_cluster(kclust.2c, data = att.num.df %>% select(-caseID))

plot(silhouette(kclust.2c$cluster, (daisy(att.num.df %>% select(-caseID)))^2), col = 1:2,
    border = NA)
abline(v = summary(silhouette(kclust.2c$cluster, (daisy(att.num.df %>% select(-caseID)))^2))
    $avg.width, col = 'gray', lty = 2, lwd = 4)
#https://stackoverflow.com/questions/32570693/make-silhouette-plot-legible-for-k-means

# choose 4 clusters
set.seed(2018)
kclust.4c <- kmeans(att.num.df %>% select(-caseID), centers = 4)
fviz_cluster(kclust.4c, data = att.num.df %>% select(-caseID), geom = 'point', ellipse = F)

# kmeans on grouped dataset ----

fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = kmeans, method = "wss")
fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = kmeans, method = "silhouette")

a <- fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = kmeans, method = "wss") +
ggtitle('Scree Plot')
b <- fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = kmeans, method = "silhouette") +
ggtitle('Avg. Silhouette Width Plot')
grid.arrange(a,b, ncol=2)

# choose 2 clusters
set.seed(2018)
kclust.2c <- kmeans(att.num.df2%>% select(-caseID), centers = 2)
fviz_cluster(kclust.2c, data = att.num.df2 %>% select(-caseID), geom = 'point', ellipse = F)
     + ggtitle('2 Cluster Plot')
plot(silhouette(kclust.2c$cluster, (daisy(att.num.df2 %>% select(-caseID)))^2), col = 1:2,
    border = NA, main = 'Silhouette Plot - 2 Clusters')

# choose 3 clusters
set.seed(2018)
kclust.3c <- kmeans(att.num.df2%>% select(-caseID), centers = 3)
fviz_cluster(kclust.3c, data = att.num.df2 %>% select(-caseID), geom = 'point', ellipse = T)

# compare clusters
cluster.stats(dist(att.num.df2 %>% select(-caseID)), kclust.2c$cluster, kclust.3c$cluster)
    $corrected.rand

# combine the data with the cluster (K-MEANS 2 cluster) ----
#att.num.df.kcluster <- bind_cols(att.num.df, tibble(kmeans.2cluster = kclust.2c$cluster))
#att.num.df.kcluster <- bind_cols(att.num.df.kcluster, tibble(kmeans.3cluster = kclust.3
    c$cluster))
apphappy.num.kcluster <- bind_cols(apphappy.3.num.frame, tibble(kmeans.2cluster = kclust.2
    c$cluster))
apphappy.num.kcluster <- bind_cols(apphappy.num.kcluster, tibble(kmeans.3cluster = kclust.3
    c$cluster))
```

```
apphappy.num.kcluster <- apphappy.num.kcluster %>%
dplyr::select(
-starts_with('q24'),
-starts_with('q25'),
-starts_with('q26')
)

apphappy.labs.kcluster <- bind_cols(apphappy.3.labs.frame, tibble(kmeans.2cluster = kclust.2
    c$cluster))
apphappy.labs.kcluster <- bind_cols(apphappy.labs.kcluster, tibble(kmeans.3cluster = kclust
    .3c$cluster))
apphappy.labs.kcluster <- apphappy.labs.kcluster %>%
dplyr::select(
-starts_with('q24'),
-starts_with('q25'),
-starts_with('q26')
)

# apphappy.lab.kcluster <- bind_cols(apphappy.3.labs.frame, tibble(cluster = kclust$cluster)
    )

# kmeans profiling with grouped data ----
# deep profiling using kcluster
apphappy.labs.kcluster %>%
select(
q48,
kmeans.2cluster
) %>%
group_by(q48,kmeans.2cluster) %>%
summarize(
obs.count = n()
) %>%
ungroup() %>%
ggplot(aes(x = q48, y = obs.count)) +
geom_bar(stat = 'identity', aes(fill = as.factor(kmeans.2cluster)), position = 'dodge') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
scale_fill_discrete(name = 'cluster')

apphappy.labs.kcluster %>%
select(
q48,
kmeans.2cluster
) %>%
group_by(q48,kmeans.2cluster) %>%
summarize(
obs.count = n()
) %>%
ungroup() %>%
spread(
key = kmeans.2cluster,
value = obs.count,
fill = 0
)

apphappy.labs.kcluster %>%
select(
q48,
kmeans.2cluster
) %>%
mutate(
kmeans.2cluster = as.factor(kmeans.2cluster)
) %>%
group_by(
q48,
kmeans.2cluster
) %>%
summarize(
obs.pct = n()/1800
) %>%
ungroup() %>%
ggplot(aes(x = as.factor(kmeans.2cluster), y = obs.pct)) +
geom_bar(stat = 'identity', aes(fill = q48)) +
scale_fill_brewer(palette = 'YlGnBu')



# to mix aes and aes_string
`+.uneval` <- function(a,b) {
`class<-`(modifyList(a,b), "uneval")
}
DeepCheck <- function(df, col1, cluster, title, xaxis) {
col1 <- enquo(col1)
cluster <- enquo(cluster)
df %>%
select(
!!col1,
!!cluster
) %>%
group_by(!!col1, !!cluster) %>%
mutate(
obs.count = n(),
cluster = as.factor(!!cluster)
) %>%
ungroup() %>%
ggplot(aes_string(x=rlang::quo_text(col1))+aes(y = obs.count)) +
#geom_bar(stat = 'identity')
geom_bar(stat = 'identity', aes_string(fill = 'cluster'), position = 'dodge') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
scale_fill_discrete(name = 'Cluster') +
ggtitle(title) +
labs(x = xaxis, y = 'Observation Count')
}

DeepCheck(apphappy.labs.kcluster, q1, kmeans.2cluster, title = 'Cluster Frequency Plot - Age
     Group', xaxis = 'Age Groups')

# type of device
a <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r1)), q2r1, kmeans.2
    cluster, title = '', xaxis = '') # iphone
b <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r2)), q2r2, kmeans.2
    cluster, title = '', xaxis = '') # ipod touch
c <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r3)), q2r3, kmeans.2
    cluster, title = '', xaxis = '') # android
d <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r4)), q2r4, kmeans.2
    cluster, title = '', xaxis = '') # blackberry
e <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r5)), q2r5, kmeans.2
    cluster, title = '', xaxis = '') # nokia
```

```
f <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q2r6)), q2r6, kmeans.2
    cluster, title = ’’, xaxis = ’’) # windows phone
g <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q2r7)), q2r7, kmeans.2
    cluster, title = ’’, xaxis = ’’) # Palm OS
h <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q2r8)), q2r8, kmeans.2
    cluster, title = ’’, xaxis = ’’) # Tablet
i <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q2r9)), q2r9, kmeans.2
    cluster, title = ’’, xaxis = ’’) # other

grid.arrange(a, b, c, d, e, f, g, h, i, ncol = 3, top = ’Cluster Frequency Plot - Type of
    Device’)

# type of app
a <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r1)), q4r1, kmeans.2
    cluster, title = ’’, xaxis = ’’) # music id app
b <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r2)), q4r2, kmeans.2
    cluster, title = ’’, xaxis = ’’) # tv check in app
c <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r3)), q4r3, kmeans.2
    cluster, title = ’’, xaxis = ’’) # entertainment app
d <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r4)), q4r4, kmeans.2
    cluster, title = ’’, xaxis = ’’) # tv show app
e <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r5)), q4r5, kmeans.2
    cluster, title = ’’, xaxis = ’’) # gaming app
f <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r6)), q4r6, kmeans.2
    cluster, title = ’’, xaxis = ’’) # social app
g <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r7)), q4r7, kmeans.2
    cluster, title = ’’, xaxis = ’’) # general news app
h <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r8)), q4r8, kmeans.2
    cluster, title = ’’, xaxis = ’’) # shopping app
i <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r9)), q4r9, kmeans.2
    cluster, title = ’’, xaxis = ’’) # specific news pub app
j <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r10)), q4r10, kmeans.2
    cluster, title = ’’, xaxis = ’’) # other
k <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl(’NO TO’, q4r11)), q4r11, kmeans.2
    cluster, title = ’’, xaxis = ’’) # none

grid.arrange(a, b, c, d, e, f, g, h, i, j, k, ncol = 6, top = ’Cluster Frequency Plot - Type
     of App Used’)

DeepCheck(apphappy.labs.kcluster, q11, kmeans.2cluster, title = ’Cluster Frequency Plot -
    Number of Apps’, xaxis = ’Number of Apps’)
DeepCheck(apphappy.labs.kcluster, q12, kmeans.2cluster, title = ’Cluster Frequency Plot -
    Percent Free Apps’, xaxis = ’Percentage’)

# frequency of visiting website
a <- DeepCheck(apphappy.labs.kcluster, q13r1, kmeans.2cluster, title = ’Facebook’, xaxis =
    ’’)
b <- DeepCheck(apphappy.labs.kcluster, q13r2, kmeans.2cluster, title = ’Twitter’, xaxis =
    ’’)
c <- DeepCheck(apphappy.labs.kcluster, q13r3, kmeans.2cluster, title = ’Myspace’, xaxis =
    ’’)
d <- DeepCheck(apphappy.labs.kcluster, q13r4, kmeans.2cluster, title = ’Pandora’, xaxis =
    ’’)
e <- DeepCheck(apphappy.labs.kcluster, q13r5, kmeans.2cluster, title = ’Vevo’, xaxis = ’’)
f <- DeepCheck(apphappy.labs.kcluster, q13r6, kmeans.2cluster, title = ’YouTube’, xaxis =
    ’’)
g <- DeepCheck(apphappy.labs.kcluster, q13r7, kmeans.2cluster, title = ’AOL Radio’, xaxis =
    ’’)
h <- DeepCheck(apphappy.labs.kcluster, q13r8, kmeans.2cluster, title = ’Last.fm’, xaxis =
    '')
i <- DeepCheck(apphappy.labs.kcluster, q13r9, kmeans.2cluster, title = 'Yahoo Entertainment
    & Music', xaxis = '')
j <- DeepCheck(apphappy.labs.kcluster, q13r10, kmeans.2cluster, title = 'IMDB', xaxis = '')
k <- DeepCheck(apphappy.labs.kcluster, q13r11, kmeans.2cluster, title = 'LinkedIn', xaxis =
    '')
l <- DeepCheck(apphappy.labs.kcluster, q13r12, kmeans.2cluster, title = 'Netflix', xaxis =
    '')

grid.arrange(a, b, c, d, e, f, g, h, i , j, l, ncol = 4, top = 'Number of Times Visting
    Website')

DeepCheck(apphappy.labs.kcluster, q48, kmeans.2cluster, title = 'Cluster Frequency Plot -
    Education', xaxis = 'Education Level')
DeepCheck(apphappy.labs.kcluster, q49, kmeans.2cluster, title = 'Cluster Frequency Plot -
    Marital Status', xaxis = 'Marital Status')

a <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r1)), q50r1, kmeans.2
    cluster, title ='', xaxis = '')
b <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r2)), q50r2, kmeans.2
    cluster, title = '', xaxis = '')
c <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r3)), q50r3, kmeans.2
    cluster, title = '', xaxis = '')
d <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r4)), q50r4, kmeans.2
    cluster, title = '', xaxis = '')
e <- DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r5)), q50r5, kmeans.2
    cluster, title = '', xaxis = '')

grid.arrange(a,b,c,d,e, ncol = 3, top = 'Cluster Frequency Plot - Children')

a <- DeepCheck(apphappy.labs.kcluster, q54, kmeans.2cluster, title = 'Cluster Frequency Plot
     - Race', xaxis = 'Race')
b <- DeepCheck(apphappy.labs.kcluster, q55, kmeans.2cluster, title = 'Cluster Frequency Plot
     - Hispanic or Latino', xaxis = 'Identify as Hispanic or Latino?')

grid.arrange(a,b)

DeepCheck(apphappy.labs.kcluster, q56, kmeans.2cluster, title = 'Cluster Frequency Plot -
    Income', xaxis = 'Income Level')
DeepCheck(apphappy.labs.kcluster, q57, kmeans.2cluster, title = 'Cluster Frequency Plot -
    Gender', xaxis = 'Gender')

DeepCheck(apphappy.labs.kcluster, q1, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster, q11, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster, q12, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster, q48, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster, q49, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r1)), q50r1, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r2)), q50r2, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r3)), q50r3, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r4)), q50r4, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r5)), q50r5, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster, q54, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster, q56, kmeans.3cluster)
DeepCheck(apphappy.labs.kcluster, q57, kmeans.3cluster)


apphappy.labs.kcluster %>%
select(
q48,
kmeans.2cluster
) %>%
mutate(
kmeans.2cluster = as.factor(kmeans.2cluster)
) %>%
group_by(
q48,
kmeans.2cluster
) %>%
summarize(
obs.pct = n()/1800
) %>%
ungroup() %>%
ggplot(aes(x = as.factor(kmeans.2cluster), y = obs.pct)) +
geom_bar(stat = 'identity', aes(fill = q48))

DeepCheck2 <- function(df, col1, cluster) {
col1 <- enquo(col1)
cluster <- enquo(cluster)
df %>%
select(
!!col1,
!!cluster
) %>%
group_by(
!!col1,
!!cluster
) %>%
summarize(
obs.pct = n()/1800
) %>%
ggplot(aes_string(x = rlang::quo_text(cluster)) + aes(y = obs.pct)) +
geom_bar(stat = 'identity', aes_string(fill = rlang::quo_text(col1))) +
scale_x_continuous(breaks = c(0,1,2,3), name = 'Cluster')
}

DeepCheck2(apphappy.labs.kcluster, q1, kmeans.2cluster)
DeepCheck2(apphappy.labs.kcluster, q11, kmeans.2cluster)
DeepCheck2(apphappy.labs.kcluster, q12, kmeans.2cluster)
DeepCheck2(apphappy.labs.kcluster, q48, kmeans.2cluster)
DeepCheck2(apphappy.labs.kcluster, q49, kmeans.2cluster)
DeepCheck2(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r1)), q50r1, kmeans.3cluster
    )
DeepCheck2(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r2)), q50r2, kmeans.3cluster
    )
DeepCheck2(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r3)), q50r3, kmeans.3cluster
    )
DeepCheck2(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r4)), q50r4, kmeans.3cluster
    )
DeepCheck2(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r5)), q50r5, kmeans.3cluster
    )
DeepCheck2(apphappy.labs.kcluster, q54, kmeans.2cluster)
DeepCheck2(apphappy.labs.kcluster, q55, kmeans.2cluster)
DeepCheck2(apphappy.labs.kcluster, q56, kmeans.2cluster)

DeepCheck2(apphappy.labs.kcluster, q1, kmeans.3cluster)
```

```
deepTbl <- function(df, col1, cluster) {
col1 <- enquo(col1)
cluster <- enquo(cluster)
df %>%
select(
!!col1,
!!cluster
) %>%
group_by(!!col1, !!cluster) %>%
summarize(
obs.count = n()
) %>%
ungroup() %>%
spread(
key = !!cluster,
value = obs.count,
fill = 0
)
}

# using 2 clusters
deepTbl(apphappy.labs.kcluster, q1, kmeans.2cluster) #age
# type of device
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r1)), q2r1, kmeans.2cluster) #
    iphone
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r2)), q2r2, kmeans.2cluster) #
    ipod touch
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r3)), q2r3, kmeans.2cluster) #
    android
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r4)), q2r4, kmeans.2cluster) #
    blackberry
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r5)), q2r5, kmeans.2cluster) #
    nokia
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r6)), q2r6, kmeans.2cluster) #
    windows phone
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r7)), q2r7, kmeans.2cluster) #
    Palm OS
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r8)), q2r8, kmeans.2cluster) #
    Tablet
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r9)), q2r9, kmeans.2cluster) #
    other
# type of app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r1)), q4r1, kmeans.2cluster) #
    music id app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r2)), q4r2, kmeans.2cluster) #
    tv check in app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r3)), q4r3, kmeans.2cluster) #
    entertainment app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r4)), q4r4, kmeans.2cluster) #
    tv show app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r5)), q4r5, kmeans.2cluster) #
    gaming app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r6)), q4r6, kmeans.2cluster) #
    social app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r7)), q4r7, kmeans.2cluster) #
```

```
    general news app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r8)), q4r8, kmeans.2cluster) #
    shopping app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r9)), q4r9, kmeans.2cluster) #
    specific news pub app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r10)), q4r10, kmeans.2cluster) #
     other
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r11)), q4r11, kmeans.2cluster) #
     none
deepTbl(apphappy.labs.kcluster, q11, kmeans.2cluster) #number of apps
# website visit frequency
deepTbl(apphappy.labs.kcluster, q13r1, kmeans.2cluster) # facebook
deepTbl(apphappy.labs.kcluster, q13r2, kmeans.2cluster) # twitter
deepTbl(apphappy.labs.kcluster, q13r3, kmeans.2cluster) # myspace
deepTbl(apphappy.labs.kcluster, q13r4, kmeans.2cluster) # Pandora
deepTbl(apphappy.labs.kcluster, q13r5, kmeans.2cluster) # Vevo
deepTbl(apphappy.labs.kcluster, q13r6, kmeans.2cluster) # YouTube
deepTbl(apphappy.labs.kcluster, q13r7, kmeans.2cluster) # AOL Radio
deepTbl(apphappy.labs.kcluster, q13r8, kmeans.2cluster) # Last.fm
deepTbl(apphappy.labs.kcluster, q13r9, kmeans.2cluster) # Yahoo Entertainment
deepTbl(apphappy.labs.kcluster, q13r10, kmeans.2cluster) # IMDB
deepTbl(apphappy.labs.kcluster, q13r11, kmeans.2cluster) # LinkedIn
deepTbl(apphappy.labs.kcluster, q13r12, kmeans.2cluster) # Netflix

deepTbl(apphappy.labs.kcluster, q12, kmeans.2cluster) # % free apps
deepTbl(apphappy.labs.kcluster, q48, kmeans.2cluster) #education level
deepTbl(apphappy.labs.kcluster, q49, kmeans.2cluster) # marital status
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r1)), q50r1, kmeans.2cluster) #
     no children
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r2)), q50r2, kmeans.2cluster) #
     children under 6
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r3)), q50r3, kmeans.2cluster) #
     kids 6-12
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r4)), q50r4, kmeans.2cluster) #
     kids 13-17
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r5)), q50r5, kmeans.2cluster) #
     kids 18+
deepTbl(apphappy.labs.kcluster, q54, kmeans.2cluster) #race
deepTbl(apphappy.labs.kcluster, q55, kmeans.2cluster) #latino?
deepTbl(apphappy.labs.kcluster, q56, kmeans.2cluster) #income
deepTbl(apphappy.labs.kcluster, q57, kmeans.2cluster) #gender

# using 3 clusters
deepTbl(apphappy.labs.kcluster, q1, kmeans.3cluster) #age
# type of device
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r1)), q2r1, kmeans.3cluster) #
    iphone
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r2)), q2r2, kmeans.3cluster) #
    ipod touch
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r3)), q2r3, kmeans.3cluster) #
    android
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r4)), q2r4, kmeans.3cluster) #
    blackberry
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r5)), q2r5, kmeans.3cluster) #
    nokia
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r6)), q2r6, kmeans.3cluster) #
    windows phone
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r7)), q2r7, kmeans.3cluster) #
    Palm OS
```

```
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r8)), q2r8, kmeans.3cluster) #
    Tablet
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q2r9)), q2r9, kmeans.3cluster) #
    other
# type of app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r1)), q4r1, kmeans.3cluster) #
    music id app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r2)), q4r2, kmeans.3cluster) #
    tv check in app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r3)), q4r3, kmeans.3cluster) #
    entertainment app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r4)), q4r4, kmeans.3cluster) #
    tv show app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r5)), q4r5, kmeans.3cluster) #
    gaming app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r6)), q4r6, kmeans.3cluster) #
    social app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r7)), q4r7, kmeans.3cluster) #
    general news app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r8)), q4r8, kmeans.3cluster) #
    shopping app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r9)), q4r9, kmeans.3cluster) #
    specific news pub app
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r10)), q4r10, kmeans.3cluster) #
     other
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q4r11)), q4r11, kmeans.3cluster) #
     none
deepTbl(apphappy.labs.kcluster, q11, kmeans.3cluster) #number of apps
# visit frequency
deepTbl(apphappy.labs.kcluster, q13r1, kmeans.3cluster) # facebook
deepTbl(apphappy.labs.kcluster, q13r2, kmeans.3cluster) # twitter
deepTbl(apphappy.labs.kcluster, q13r3, kmeans.3cluster) # myspace
deepTbl(apphappy.labs.kcluster, q13r4, kmeans.3cluster) # Pandora
deepTbl(apphappy.labs.kcluster, q13r5, kmeans.3cluster) # Vevo
deepTbl(apphappy.labs.kcluster, q13r6, kmeans.3cluster) # YouTube
deepTbl(apphappy.labs.kcluster, q13r7, kmeans.3cluster) # AOL Radio
deepTbl(apphappy.labs.kcluster, q13r8, kmeans.3cluster) # Last.fm
deepTbl(apphappy.labs.kcluster, q13r9, kmeans.3cluster) # Yahoo Entertainment
deepTbl(apphappy.labs.kcluster, q13r10, kmeans.3cluster) # IMDB
deepTbl(apphappy.labs.kcluster, q13r11, kmeans.3cluster) # LinkeIn
deepTbl(apphappy.labs.kcluster, q13r12, kmeans.3cluster) # Netflix

deepTbl(apphappy.labs.kcluster, q12, kmeans.3cluster) # % free apps
deepTbl(apphappy.labs.kcluster, q48, kmeans.3cluster) #education level
deepTbl(apphappy.labs.kcluster, q49, kmeans.3cluster) # marital status
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r1)), q50r1, kmeans.3cluster) #
     no children
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r2)), q50r2, kmeans.3cluster) #
     children under 6
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r3)), q50r3, kmeans.3cluster) #
     kids 6-12
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r4)), q50r4, kmeans.3cluster) #
     kids 13-17
deepTbl(apphappy.labs.kcluster %>% filter(!grepl('NO TO', q50r5)), q50r5, kmeans.3cluster) #
     kids 18+
deepTbl(apphappy.labs.kcluster, q54, kmeans.3cluster) #race
deepTbl(apphappy.labs.kcluster, q55, kmeans.3cluster) #latino?
deepTbl(apphappy.labs.kcluster, q56, kmeans.3cluster) #income
deepTbl(apphappy.labs.kcluster, q57, kmeans.3cluster) #gender
```

```
apphappy.labs.kcluster %>%
ggplot(aes(x = q48)) +
geom_bar(stat = 'count', aes(fill = as.factor(kmeans.2cluster)), position = 'dodge') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
facet_grid(q57 ~ .) +
scale_fill_discrete(name = 'cluster')


apphappy.labs.kcluster %>%
ggplot(aes(x = q49)) +
geom_bar(stat = 'count', aes(fill = q57), position = 'dodge') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
facet_grid(kmeans.3cluster ~ .) +
scale_fill_discrete(name = 'gender')


# Heirarchical Clustering ----

fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = hcut, method = "wss")
fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = hcut, method = "silhouette")

set.seed
a <- fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = hcut, method = "wss") +
ggtitle('Scree Plot')
b <- fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = hcut, method = "silhouette") +
ggtitle('Avg. Silhouette Width Plot')
grid.arrange(a,b, ncol=2)

hclust1 <- hclust(dist(att.num.df2 %>% select(-caseID)), method = 'ward.D2')
plot(hclust1, xlab = 'Observations', main = 'Dendrogram for Grouped Responses', cex = 0.6)
rect.hclust(hclust1, k=2, border=2:5)

cor(cophenetic(hclust1), dist(att.num.df2 %>% select(-caseID)))
hclust1.cut3 <- cutree(hclust1, k = 3)
table(hclust1.cut3)
plot(silhouette(hclust1.cut3, dist(att.num.df2 %>% select(-caseID))), border = NA)
summary(silhouette(hclust1.cut3, dist(att.num.df2 %>% select(-caseID))))


hclust1.cut2 <- cutree(hclust1, k=2)
table(hclust1.cut2)
plot(silhouette(hclust1.cut2, dist(att.num.df2 %>% select(-caseID))), border = NA)
summary(silhouette(hclust1.cut2, dist(att.num.df2 %>% select(-caseID))))

fviz_cluster(list(data = att.num.df2 %>% select(-caseID), cluster = hclust1.cut2), ellipse =
     F)
fviz_cluster(list(data = att.num.df2 %>% select(-caseID), cluster = hclust1.cut3), ellipse.
    type = 't')

#http://uc-r.github.io/hc_clustering
fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = hcut, method = "wss")
fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = hcut, method = "silhouette")
#set.seed(2018)
```

```
#gap_stat <- clusGap(att.num.df %>% select(-caseID), FUN = hcut, nstart = 25, K.max = 10, B
    = 50)
#fviz_gap_stat(gap_stat)

# methods to assess
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
#the ward above is actually Ward.D2 in method

# function to compute coefficient
ac <- function(x) {
agnes(att.num.df2 %>% select(-caseID), method = x)$ac
}

map_dbl(m, ac)
hc3 <- agnes(att.num.df2 %>% select(-caseID), method = "ward")
pltree(hc3, cex = 0.6, hang = -1, main = "Dendrogram of agnes")
#cutree(as.hclust(hc3), k = 3)
fviz_cluster(list(data = att.num.df2 %>% select(-caseID),cluster = cutree(as.hclust(hc3), k
    = 3)), ellipse = F)
fviz_cluster(list(data = att.num.df2 %>% select(-caseID),cluster = cutree(as.hclust(hc3), k
    = 2)), ellipse = F, geom = 'point', main = 'Cluster Plot using Hierarchical Clustering
    with Ward Linkage')
# loop to determine best # of cuts
hdf <- tibble(x=1,y=0)
hclust1 <- hclust(dist(att.num.df2 %>% select(-caseID)), method = 'complete')
plot(hclust1)
for (i in 2:15) {
hc <- cutree(hclust1, k = i)
sk2  <- silhouette(hc, dist(att.num.df2 %>% select(-caseID)))
hdf <- add_row(hdf, x=i, y=round(summary(sk2)$avg.width,3))
print(paste0('cluster size: ', i, ' avg. silhouette width: ', round(summary(sk2)$avg.width
    ,3)))
if(i == 15) {
print(
hdf %>%
ggplot(aes(x = x, y = y)) +
geom_line(col='blue') +
geom_point(col='red') +
ggtitle('Silhouette Average Widths') +
xlab('Cluster Size') +
ylab('Average Width')
)
}
}

plot(hclust1)
rect.hclust(hclust1, k=2, border="red")

# find WSS, TSS and BSS for heirarchical clustering
{
library(proxy)
hclust1.cut2 <- cutree(hclust1, k=2)
hclust1.combo.df <- bind_cols(att.num.df, tibble(cluster = hclust1.cut2))

# TSS
TSS <- sum(
dist(
hclust1.combo.df %>%
select(-caseID, -cluster) %>%
map_df(function(x) mean(x))
,
hclust1.combo.df %>%
select(-caseID, -cluster)
)^2
)

# WSS
# cluster 1
dis1 <- sum(
dist(
hclust1.combo.df %>%
dplyr::filter(cluster == 1) %>%
select(-caseID, -cluster) %>%
map_df(function(x) mean(x))
,
hclust1.combo.df %>%
dplyr::filter(cluster == 1) %>%
select(-caseID, -cluster)
)^2
)

#cluster 2
dis2 <- sum(
dist(
hclust1.combo.df %>%
dplyr::filter(cluster == 2) %>%
select(-caseID, -cluster) %>%
map_df(function(x) mean(x))
,
hclust1.combo.df %>%
dplyr::filter(cluster == 2) %>%
select(-caseID, -cluster)
)^2
)

WSS <- sum(dis1, dis2)
BSS <- TSS - WSS
rsquare <- BSS/TSS
rsquare
}


# PAM Clustering ----
fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = pam, method = "wss")
fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = pam, method = "silhouette")

a <- fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = pam, method = "wss") + ggtitle('
    Scree Plot - PAM')
b <- fviz_nbclust(att.num.df2 %>% select(-caseID), FUN = pam, method = "silhouette") +
    ggtitle('Avg. Silhouette Width - PAM')
grid.arrange(a,b,ncol=2)

#pam.2clust
set.seed(2018)
pam.2clust <- pam(att.num.df2 %>% select(-caseID), k=2)
fviz_cluster(pam.2clust, data = att.num.df2 %>% select(-caseID), geom = 'point', ellipse.
    type = 't', repel = T)
```

```
fviz_cluster(pam.2clust, data = att.num.df2 %>% select(-caseID), geom = ’point’, ellipse = F
    ) + ggtitle(’2 Cluster PAM’)

#pam.3clust
set.seed(2018)
pam.3clust <- pam(att.num.df2 %>% select(-caseID), k=3)
fviz_cluster(pam.3clust, data = att.num.df2 %>% select(-caseID), geom = ’point’, ellipse.
    type = ’t’, repel = T)
fviz_cluster(pam.3clust, data = att.num.df2 %>% select(-caseID), geom = ’point’, ellipse = F
    )

# compare clusters
cluster.stats(dist(att.num.df2 %>% select(-caseID)), pam.2clust$clustering, pam.3
    clust$clustering)$corrected.rand
plot(silhouette(kclust.2c$cluster, (dist(att.num.df2 %>% select(-caseID)))^2), col = 1:2,
    border = NA)
plot(silhouette(pam.2clust$clustering, (dist(att.num.df2 %>% select(-caseID)))^2), col =
    1:2, border = NA)
plot(silhouette(kclust.3c$cluster, (dist(att.num.df2 %>% select(-caseID)))^2), col = 1:3,
    border = NA)
plot(silhouette(pam.3clust$clustering, (dist(att.num.df2 %>% select(-caseID)))^2), col =
    1:3, border = NA)


# Model Based Clustering ----
set.seed(2018)
mclust.fit <- Mclust(att.num.df2 %>% select(-caseID),2)
plot(mclust.fit,data=numsub, what="density") # plot results (does not work)
summary(mclust.fit) # display the best model

mclustBIC(att.num.df2 %>% select(-caseID))

mclust.fit1 <- Mclust(att.num.df2 %>% select(-caseID))
summary(mclust.fit1)
mclust.fit2 <- Mclust(att.num.df2 %>% select(-caseID), G = 3)
summary(mclust.fit2)

BIC(mclust.fit1, mclust.fit2)

seg.summ <- function(data, groups) {
aggregate(data, list(groups), function(x) mean(as.numeric(x)))
}

seg.summ(att.num.df2 %>% select(-caseID), mclust.fit1$classification)

clusplot(att.num.df2 %>% select(-caseID), mclust.fit1$classification, color=TRUE, shade=TRUE
    , labels=4, lines=0, main="Model-based cluster plot")

fviz_mclust(mclust.fit1, what = ’classification’)
fviz_mclust(mclust.fit2, what = ’classification’)

fviz_mclust_bic(mclust.fit1)
```