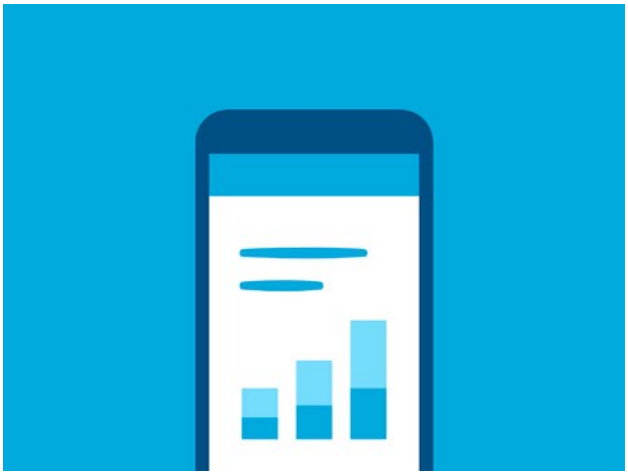
Solo 1: Market Segmentation Analysis



Name: Young, Brent

MSDS 450 Section #: 55

Quarter: Winter 2019

Introduction

Problem

The purpose of Solo 1 Assignment is to analyze survey data using general attitudinal post hoc segmentation analysis (e.g., Hierarchical, K-means, PAM, and density based clustering techniques). The goal of this market segmentation analysis is to help The App Happy Company better understand the market for a new *social entertainment* app that they are thinking of developing within the consumer entertainment app category. For instance, our end goal is to use this data to create an app that can grow organically. Over time, we will be able to use the data that we gathered to make modifications/additions to the app and improve user experience. As a result, in order to accomplish this, we will develop and evaluate a segmentation scheme, profile the segments in the scheme, interpret the results, and make recommendations about product opportunities or additional research. Descriptions on analytic methodology, summary of assumptions, caveats, and limitations will also be provided to App Happy. Lastly, classification models (aka: "typing tools") will be recommended so that App Happy can put future consumers that it doesn't currently have data on into the segments that we will define for the company *(dependent upon the company obtaining data on these consumers)*.

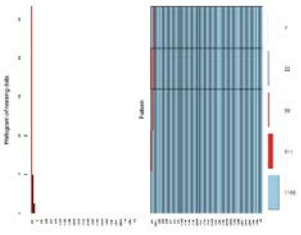
Exploratory Data Analysis

Structure and Description of Datasets & Variables

The structure of the dataset includes 1,800 rows (respondents) and 89 columns (includes caseID variable and specific question coding) and was collected from a sample of consumers in the market by Consumer Spy Corporate (CSC). The original 57 question survey questionnaire was based on qualitative research that included focus groups and one-on-one interviews. However, the survey subset that was used for general attitudinal post hoc segmentation analysis includes only questions 24 through 26. These questions served as our basis variables, are attitudinal in nature, and relate to consumer and purchasing behavior, personality, and attitudes on shopping, technology, and leadership. As a result, the rest of the questions are related to respondent demographics and technology/app usage. The data is stored in an R data file called apphappyData.Rdata, within two R data frames called apphappy.3.num.frame and apphappy.3.labs.frame. The first data frame includes numerically coded survey response data, while the second data frame includes response data coded in the character strings of the questionnaire's value labels.

Data Preparation/Cleaning/Imputation

Overall, there are 656 NA's in the dataset. The histogram on the right shows the total number of missing values for all the variables in the dataset using the VIM package. For instance, the following questions had missing values: q5r1 (533, 30%), q12 (24, 1%), and q57 (99, 5.5%). To address the missing values, I used the MICE package (predictive mean matching) to fill in the NA's. After conducting the imputation, my summary statistics showed that there were no more NA's. This is important since we can now use this data during the profiling phase.



Respondent Demographics

Observations: The barplots of respondent demographics in appendix A were created so that we can obtain a better idea of our survey respondent population and check for distributions prior to conducting cluster analysis. The data shows a right skew for age range with 65% of respondents falling in-between the 18 to 39 age group. Additionally, the data shows that 65% of respondents have some college or college graduate education (normal distribution), 41% are married, 36% are single, 74% are White or Caucasian, and 52% are female. Furthermore, in regards to household income, majority of respondents fell in-between $30,000 to $59,999, with the $150,000 and

over range having the most respondents (normal distribution). Overall, a possible *limitation* of the survey data is how skewed the survey sample is in regards to respondent demographics and may not reflect a representative sample that is similar to what we see in the U.S. and/or international market.

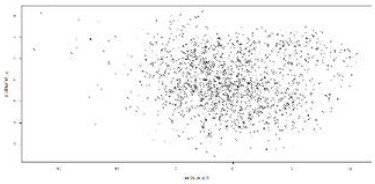
Respondent Technology/App Usage

Observations: The barplots in appendix B show respondent technology and app usage according to the survey results. The data shows that 50% of respondents own an iPhone, 37% own an Android smartphone, 23% own an iPod touch, 19% own a BlackBerry, and 19% own a Tablet. Additionally, 81% of respondents use social apps, 75% use gaming apps, 70% of respondents use music apps, 52% use general news apps, 47% use shopping apps, and 45% use enterainment apps, respectively. Furthermore, 34% of respondents have between 11-30 apps, while 37% of respondents have over 31 apps (appendix C). Additionally, 70% of respondents reported that at least 51% of the apps that they've downloaded were free (majority of respondents falling in the 76% - 99% range) (appendix C). Lastly, according to the survey results, Facebook, YouTube, Pandora, and Netflix were visited most frequently on a weekly basis (appendix D).

Attitudinal Survey Results & Correlation Matrix

Appendix E shows a divergent stacked bar chart of questions 24 to 26. Based on favorability, respondents are technologically saavy, like being in control, and have a strong affinity for: web tools/apps that help them save time, learning about their favorite TV shows, browsing Facebook, listening to music, and bargains/discounts/deals. The correlation matrix of questions 24 to 26 (appendix F), allows us to see which questions may be correlated with each other so that I can gleam interesting insights. The plot shows that a lot of the sub questions within each question have strong positive correlations with each other, which indicates that they grouped questions according to technological savviness, leadership, and shopping. For example, there is a strong positive correlation between q26r18 and q26r7 (0.70). The more attracted someone is to luxury brands, the more that someone prefers to buy designer brands (vice versa). Additionally, there is a strong positive correlation between q26r10 and q26r8 (0.60). The more that people love showing off their new apps, the more they are willing to spend a few extra dollars to get extra app features (vice versa). Interestingly, there are also questions that are negatively correlated (e.g., questions q24r8 and q24r9). This is due to the fact that some of the questions are measuring "negative" aspects of technology or personality.

Principal Component Analysis & Outlier Detection

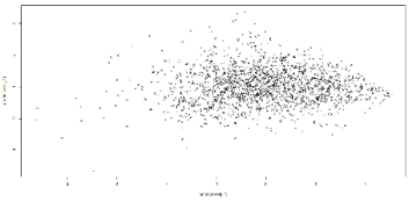


Observations: The PCA plot on the right shows that the first 2 principal components only explain 36% of the total variation in the data. Additionally, when analyzing the PCA scores, there are noticeable outliers in the data. For instance, there are 33 respondents who selected "Agree Strongly" for all questions 24 through 26. Additionally, there is 1 respondent who selected "Disagree" and 2 respondents who selected "Strongly Disagree" for all questions 24 through 26. This illustrates that these respondents most likely did not read the question. However, I decided not to handle/truncate the outliers because the outliers were minor, it wasn't due to incorrectly entered or measured data (e.g., response was outside the Likert scale), and removing them actually made my models worse. Instead, I decided to conduct featuring engineering by grouping similar questions.

Feature Engineering

In order to reduce the noise and increase the signal in the data (e.g., increase the signal/noise ratio), I decided to group similar questions together for questions 24 to 26 and then computed their means. For instance, for question 24 (technology savviness), I grouped the questions accordingly so that 12 variables are reduced to 4

features: r1, r2, r3, r5, r6 -> positive attitude towards technology; r7,r8 -> music/TV; r10, r11 -> Internet/Social Media; and r4, r9, r12 -> negative aspects of technology. For question 25 (leadership), I grouped the questions accordingly so that 12 variables are reduced to 4 features: r1, r2, r3, r4, r5 -> Leadership; r7, r8 -> Control; r9, r10, r11 -> Drive; and r6, r12 -> negative. Lastly, for question 26 (Shopping), I grouped the questions accordingly so that 12 variables are reduced to 5 features: r3, r4, r5, r6, r7 -> bargain; r8, r9, r10 -> show off; r11 -> children; r12, r13, and r14 -> hot; and r15, r16, r17, and r18 -> brand. *Caveat: I then removed all negative questions for consistency purposes and 26r11 since the question in regards to children did not make sense and therefore should be terminated.* As a result, after feature creation, the PCA plot on the right shows that the first 2 principal components now explain 66% of the total variation in the data, which is two times higher prior to feature creation. Additionally, the negative correlations are now no longer present in the correlation plot in appendix G.



Part 1: Market Segmentation

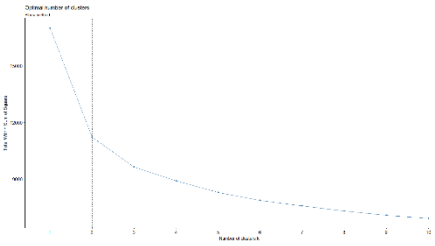
Methodology, Assumptions, Caveats & Limitations

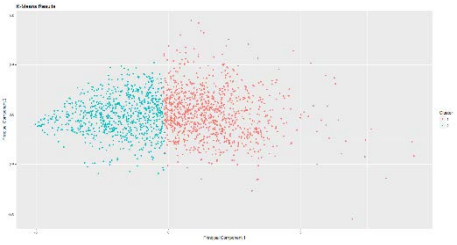
Cluster Analysis Strategy & Assumptions

Cluster analysis is an unsupervised learning technique that seeks to organize a data set into homogenous subgroups or "clusters" (Izenman, 2008). Rows that have high similarity are grouped together and rows outside the grouping have low similarity (Lander, 2014). As a result, given that the goal of this assignment is to analyze survey data using general attitudinal post hoc segmentation analysis, I decided to use distance-based clustering methods such as k-means, hierarchical, and partitioning around medoids (pam) for my analysis. Distance-based methods attempt to find groups that "minimize the distance between members within the group, while maximizing the distance of members from other groups" (Chapman & Feit, 2015). Additionally, given that we are using distance-based clustering methods, common statistical *assumptions* such as assuming a multivariate normal, heteroscedasticity, multi-collinearity, linearity, etc. do not apply.

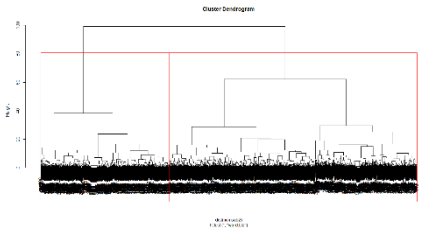
Application of Clustering Methods

K-Means Clustering: K-means clustering is a nonhierarchical or partition based method that splits the data into "predetermined" number clusters so that items within each cluster are similar to each other and items from other clusters are dissimilar (Izenman, 2008). In other words, there is no hierarchical relationship between the K-cluster solution and the (K+1)-cluster solution (Izenman, 2008). K-means begins with a random number of K points that correspond to the K clusters and then assigns items to each (initial K centroids) based on Euclidean distance. Euclidean distance measures the closeness/distance between two points in the high dimensional space (Izenman, 2008). As a result, in order to help determine the number of predetermined clusters, I created a scree plot that swept through clusters 1 to 15. However, according to the scree plot results (see graph to the right), there is no clear elbow (although k=2 seemed to be the clearest). As a result, as a *caveat*, I decided to try k-means with 5, 3, and 2 clusters and then compared the average silhouette widths accordingly. The results showed the following in regards to average silhouette widths: k=5 (0.25), k=3 (0.32), and k= 2(0.45). As a result, I decided to move forward with k-means with 2 clusters given that the average silhouette width: 0.45 was the highest compared to the others. I also validated the proper number of clusters utilizing NbClust, which provides 30 indexes for determining the optimal number of clusters. The package confirmed that according to majority rule, the best number of clusters is 2. The sizes of each cluster were also well-balanced, there was no overlap in the clusters, and had respectable silhouette

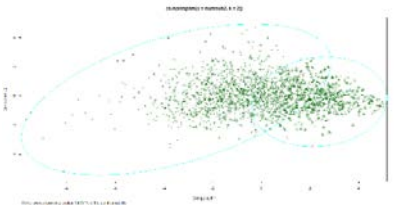




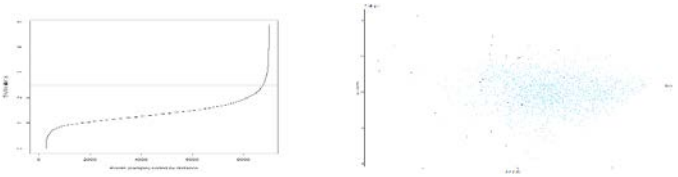
widths as well (cluster 1: 991 observations with a silhouette width of 0.36 and cluster 2: 809 observations with a silhouette width of 0.55). The r-square was 0.34039. See page 4 for the cluster plot.



Hierarchical Clustering (agglomerative): Hierarchical clustering (agglomerative) (aka: "bottom up" methods), builds clusters within clusters since each observation begins in its own individual cluster (bottom) and then clusters are sequentially joined with neighboring observations one at a time according to their distances where there are varying levels of clustering, until only one cluster remains that contains all the observations that are linked (top) (Izenman, 2008). This is illustrated in a dendrogram where items similar to each other are combined at lower heights, while dissimilar items are combined higher up. Therefore, it's the difference in heights that determines how close items are with one another (Izenman, 2008). Additionally, since hierarchical clustering does not require a predetermined number of clusters, a line is cut on the tree to determine the number of clusters (Izenman, 2008). Euclidean distance is also often used as the distance, but three linkages are used to measure the distance between clusters (e.g., single = minimum distance, complete = maximum distance, and average linkage = average distance). Given this context, I experimented with different linkage measures (k=2, same as k-means) and then compared the cluster sizes and average silhouette widths accordingly. For instance, the single linkage produced an average silhouette width of 0.57, but the cluster sizes were extremely imbalanced (1,799 observations were classified into 1 cluster, while the other cluster had only 1 observation). Next, I used an average linkage, which produced an average silhouette width of 0.44, but the cluster sizes were also extremely imbalanced (1,772 observations classified into cluster 1 and only 28 observations were classified in cluster 2). Next, I used a complete linkage which produced an average silhouette width of 0.20 and two clusters with silhouette width of 0.20 (1547 observations) and 0.21 (253 observations), respectively. Lastly, I used ward.D2 which produced better, but still skewed cluster sizes and average silhouette width of 0.24 (higher than complete linkage), r-square of 0.29 and two clusters with silhouette width of 0.18 (1184 observations) and 0.35 (616 observations). See plot to the right. However, the *limitation* of ward.D2 method is that it did not have a better r-square or average silhouette width and produced imbalanced cluster sizes compared to k-means clustering.



Partitioning Around Medoids (pam): Partitioning Around Medoids (pam) is a partition based method that is similar to k-means clustering, but instead of the center of a cluster being the mean of a cluster, it uses an actual observation (similar to median) for the center of the cluster (Lander, 2014). As a result, it's often robust to anomalies and outliers in the data (Izenman, 2008). As a result, in order to help determine the number of predetermined clusters, I created a function in R that sweeps through clusters 2 to 8 using average silhouette width. The results showed that the optimal number is 2 clusters given that the average silhouette width is 0.27 compared to an average silhouette width of 0.19 for 3 clusters and so forth. Pam produced an average silhouette width of 0.27, with skewed cluster sizes and silhouette widths of 1154 (0.21) and 646 (0.39), respectively. However, although pam produced a slightly better average silhouette width of 0.27 compared to ward.D2, the *limitation* of pam was that there's a lot of overlap in the clusters and the cluster sizes were skewed (similar to ward.D2), which isn't good. As a result, k-means still wins out since it had a higher average silhouette width, contained no cluster overlap, and had balanced cluster sizes compared to both methods.



Density Based Method (DBSCAN): Density based methods identify dense regions that are measured by the number of objects close to a given point and is similar to K-means. As a result, a potential *limitation* of

this method is that since the clusters are based on density of data points, there is 1 large cluster and not two separate ones. However, on a positive note, this method can be used to help identify outliers. For instance, the plot on the right on page 5 shows 22 black dots that illustrates outliers, which is on par with what we saw in our EDA. The DBSCAN was computed using the fpc package and choosing eps = 2.5 and MinPts = 5 accordingly (see graphs on page 5) by identifying the knee on plot.

Validation & Comparison of Clustering Methods

Method	# of Clusters	R-square	Avg. Silhouette Width	Cluster Sizes
k-means	2	0.34039	0.45	C1: 991; C2: 801
Hierarchical Clustering (ward.d2)	2	0.2899122	0.24	C1: 1184 ; C2: 616
pam	2	N/A	0.27	C1: 1154 ; C2: 646
Density Based (DBSCAN)	1	N/A	N/A	C1: 1778; 22 outliers

Observations: The table on the right shows the performance and evaluation criteria for all the clustering methods that were fit in the dataset. The results show that k-means had the highest r-square, average silhouette width, and the most balanced cluster sizes out of all the clustering methods, while pam and ward.D2 produced similar results. Given that k-means performed the best, I produced a csv file that contains the cluster results assigned to each respondent that will be used for profiling each segment. Additionally, I also computed Rand Index calculations to compare the similarity of two clustering outcomes (value between 0 and 1, 0 = no match and 1 = perfect match). The Rand Index gives us an idea of how robust our clusters are. The results showed that the clusters "kind of" match (k-means vs. pam: 0.6701434; hierarchical vs. k-means: 0.500193, and hierarchical vs. pam: 0.601595).

Segment Profiles

(See appendix 1, 2, and 3 for bar plots and median/mean of survey results that were used for this section)

Segment 1: "Technology Follower" (991; approx. 55% of Respondents): According to appendix 1 & 2, 67% of respondents fell in-between the 18 to 39 age group (average age: 37), 62% of respondents have some college or college graduate education, 42% are married, 34% are single, 76% are White or Caucasian, and 50% are female. Furthermore, in regards to household income, majority of respondents fell in-between $30,000 to $79,999 (median income: $60-69k), with the $150,000 and over range having the most respondents. The data shows that 47% of respondents own an iPhone, 34% own an Android, 18% own a iPod touch, and 17% own a Blackberry. Additionally, 76% use social apps, 69% of respondents use gaming apps, 66% of respondents use music apps, and 45% use general news apps, respectively. Furthermore, 32% of respondents have between 11-30 apps, while 32% of respondents have over 31 apps. Additionally, 75% of respondents reported that at least 51% of the apps that they've downloaded were free (majority of respondents falling in the 76% - 99% range). Moreover, according to the survey results, Facebook, YouTube, Pandora, and Netflix were visited most frequently on a weekly basis. From an attitudinal survey results perspective, appendix 1 shows median and mean survey results for questions 24 to 26 (grouped). The results showed that segment 1 "Agree's Somewhat" with positivie attitudes towards tech, music/tv, internet & social media, leadership, control, drive, and brand. On the other hand, they "Disagree Somewhat" on bargain, showoff, and hot.

Description: This group are passive/indifferent technology users, "followers", and budget conscious. They are afraid to take risks and and like being in control. As a result they approach technology products with a level of skepticism after the majority of people have adopted them. They are also knowledgeable about the latest new web tools/apps and like to give advice, but most likely learn about them through other technology enthusiasts or trend setters. They also have an affinity for gaming, internet, music, TV shows, and social media. Given their budget conscious mindset and reluctance to take risks, they prefer free apps and are unwilling to pay for extra app features. The average age for this group is 37 and the median income is $60-69k.

Segment 2: "Technology Trendsetter" (809; approx. 45% of Respondents): According to appendix 1 & 3, 75% of respondents fell in-between the 18 to 39 age group (average age: 32), 68% of respondents have some college or college graduate education, 41% are married, 38% are single, 71% are White or Caucasian, and 55% are female. Furthermore, in regards to household income, majority of respondents fell in-between $30,000 to $59,999 (median: income $50-59k), with the $150,000 and over range having the most respondents. The data shows that

53% of respondents own an iPhone, 40% own an Android smartphone, 30% own an iPod touch, 25% own a Tablet, and 22% own a Blackberry. Additionally, 87% use social apps, 84% of respondents use gaming apps, 76% of respondents use music apps, and 61% use general news apps, respectively. Furthermore, 36% of respondents have between 11-30 apps, while 43% of respondents have over 31 apps. Additionally, 65% of respondents reported that at least 51% of the apps that they've downloaded were free (majority of respondents falling in the 51% - 75% range). Lastly, according to the survey results, Facebook, Yahoo Ent. & Music, Twitter, Youtube, IMDB, Pandora, and Netflix were visited most frequently on a weekly basis. From an attitudinal survey results perspective, appendix 1 shows median and mean survey results for questions 24 to 26 (grouped). The results showed that segment 2 "Strongly Agree's " with music/tv, internet & social media, leadership, and drive. Additionally, they "Agree" with positive attitude towards technology, control, bargain, showoff, hot, and brand.

Description: This group are "techie"s, technology trendsetters, enthusiasts, and are willing spenders. They have a very strong affinity for social media, internet, TV shows, music, and gaming. They enjoy looking for new technology products, gadgets, appliances, and cool apps because they love standing out and it helps them save time given that they are active and always on the go. Upon trying these products out, they enjoy talking about them, sharing their opinons, and providing buying advice. With that said, they are "smart and knowledgable buyers", are willing to pay for an app as long as it is a good value, and are willing to pay for extra app features. Given that they are always on the go, they think of their mobile phone as a source of entertainment. The average age for this group is 32 and the median income is $50-59k.

Recommended Product Opportunities

According to the survey results, respondents from both segments have a strong affinity towards using gaming, social media, and music apps. However, whereas both social media and music were validated in the attitudinal results and website visits per week (q13), direct attitudinal questions on gaming were not asked on the survey. As a result, my recommendation would be to create a social music app, instead of App Happy's initial thinking of creating a new social entertainment app. However, given the vast attitudinal differences of the two segments, I would recommend offering a free version and a paid version. The target market for the free version would be the "Technology Followers" (aka: budget conscious) segment, while the target market for the paid version would be the "Technology Trendsetters" (aka: willing spenders) segment. The paid version would contain additional app features such as additional features/content, no ads, and ability to customize the look/feel to the user. Additionally, given that the gaming app usage was so high for both segments, I would recommend that App Happy look into adding games to the social music app in the future. However, this would need to be explored further through additional attitudinal gaming questions. Lastly, given that both segments owned Apple and Android products the most, I would recommend creating the app for both Apple and Android first and then look into expanding further to other mobile devices (e.g., Blackberry, Tablet, etc.).

Part 2: Classification

App Happy has requested that the segmentation results be used to create classification models (aka: "typing tools") so that App Happy can put future consumers that it doesn't currently have data on into these segments that we defined for them in part 1 *(dependent upon the company obtaining data on these consumers)*. As a result, there are many classification models that App Happy can apply (e.g., logistic regression, logistic regression GAM, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Naïve Bayes, K-Nearest Neighbors, Neural Network, Decision Trees, Bagging, Random Forest, Boosting, and Support Vector Machines) with varying levels of interpretability, predictive accuracy, robustness to outliers, speed, and scalability. However, in order to accomplish this, they must collect additional data, create train/validation/test datasets (e.g., the training set will be used to fit the models, the validation set will be used to estimate prediction error for model selection, and the test set will be used for assessment of the prediction error of the final model), conduct EDA, clean the data, conduct feature creation, create the model and conduct validation.

As a result, the first model that I would recommend is logistic regression. Logistic regression models the probability that the response variable belongs to a specific category and assumes a linear decision boundary (James, Witten, Hastie, & Tibshirani, 2013). For instance, it models the probabilities of the K classes using linear functions in x, while also ensuring that they sum to 1 and remain in-between 0 and 1 (Hastie, Tibshirani, & Friedman, 2009). The strengths of logistic regression is that it's easy to interpret, implement, and efficient, but it suffers from multicollinearity, is sensitive to outliers, and is inflexible. The second model I would recommend is Random Forest. Random Forest provides an improvement over bagged trees by incorporating a small tweak that decorrelates the trees and then averages them (e.g., forces each split to only consider a subset of predictors and will not consider strong predictors so that other predictors will have more of a chance (James, et al., 2013)). The strengths of Random Forest is that it's very versatile and has high classification accuracy, but can be difficult to interpret and can be prone to overfitting. The last model I would recommend is eXtreme Gradient Boosting (XGBoost). Boosting provides another approach for improving the predictions resulting from a decision tree by fitting each tree on an altered version of the original dataset (James, et al., 2013). In other words, each tree is grown using information from previously grown trees. It also incorporates parallel processing and uses a more regularized model formalization to control over-fitting, which often results in better performance. The strengths of XGBoost is that it's highly flexible, versatile, and has high performance/accuracy, but can be difficult to interpret and is computationally expensive. Overall, if predictive accuracy is top of mind for App Happy, I would recommend XGBoost out of all the models. If interpretation is top of mind I would recommend logistic regression or basic decision trees out of all the models.

Conclusion

Additional Research & Limitations: In regards to future work, there are three primary areas that I would suggest in regards to conducting additional research due to the limitations of the survey data. First, it would be beneficial to obtain additional data such as country, state, city, zip code, and app/user data such as avg. visit time, active users, screen views per visit, and types of content viewed. Second, it was concerning how skewed the survey sample was in regards to respondent demographics and the fact that it was missing data for q5r1, q12, and q57. As a result, it would be beneficial to conduct another survey that contains a representative sample that is similar to their target population (e.g., U.S. and/or international market) (including complete information). For instance, if they sampled the population incorrectly, then App Happy's analysis and results will be skewed. Furthermore, utilizing the proper sampling procedure (e.g., how respondents were chosen), is also important to ensure that the sample is representative. This could be achieved using probability sampling (Kotler & Keller, 2012), stratified random sampling, etc. Lastly, in regards to survey design, the types of survey questions that were asked and "how" they were asked also plays a huge role as well in terms of getting back solid results from its marketing efforts (e.g., potential bias could be at hand). Therefore, revisiting these questions and making slight modifications can be beneficial in the long-run. For instance, adding attitudinal questions around gaming.

Summary: In conclusion, we began this analysis by first conducting EDA to get a better feel of the dataset, respondent demographics, technology/app usage, and attitudinal survey results. We then performed missing value imputation, outlier detection, and feature engineering. We then created and validated 3-4 clustering methods (k-means, hierarchical, pam, and density-based). The results, showed that k-means performed the best based on r-square and average silhouette width. The market was segmented into two segments: Technology Followers and Technology Trendsetters. Our final recommendation was to create a social music app (free and paid version). We then concluded the analysis by suggesting three classification models (aka: "typing tools") so that App Happy can put future consumers that it doesn't currently have data on into these segments that we defined for them in part 1 (dependent upon the company obtaining data on these consumers). These models were Logisitic Regression, Random Forest, and XGBoost.

References

1. Chapman, C. & Feit, E. (2015) R for Marketing Research and Analytics. New York: Springer.

2. Kotler, P., & Keller, K. (2012). *Marketing management* (15th ed.). Boston, MA: Prentice Hall.

3. Izenman, A. J. (2008). Modern multivariate statistical techniques: Regression, classification, and manifold learning. New York: Springer. Chapter 12

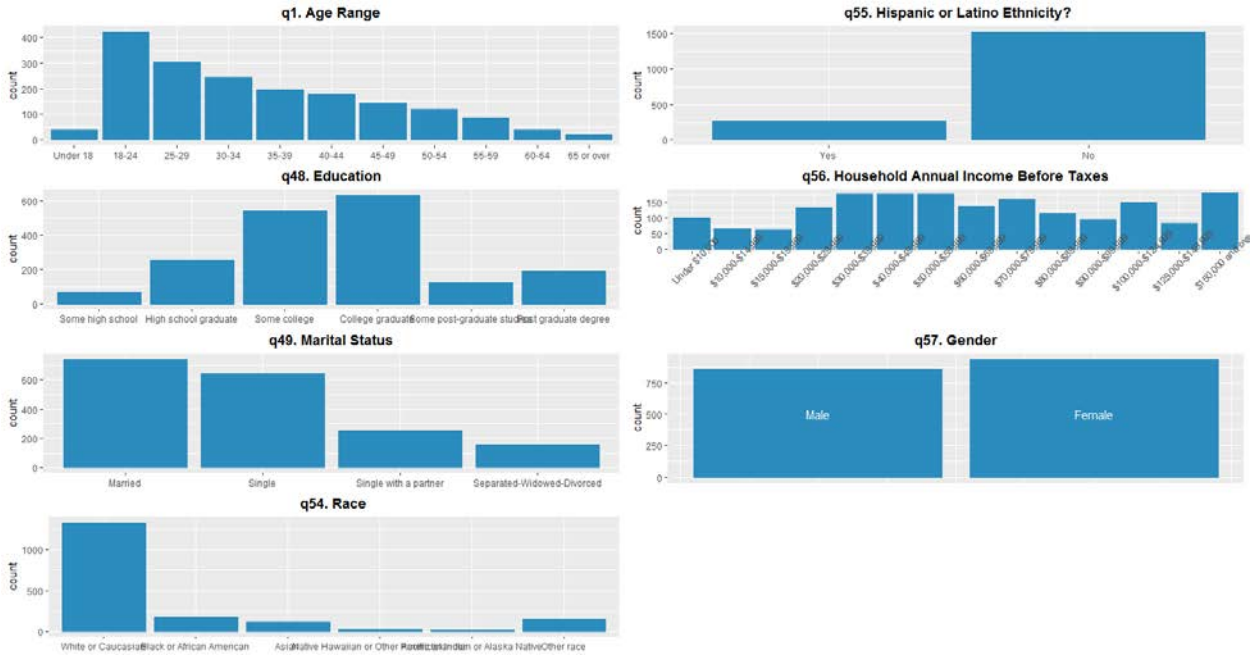
4. Lander, Jared. (2014) R for Everyone. Upper Saddle River NJ: Pearson.

5. James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. New York: Springer Science + Business Media.

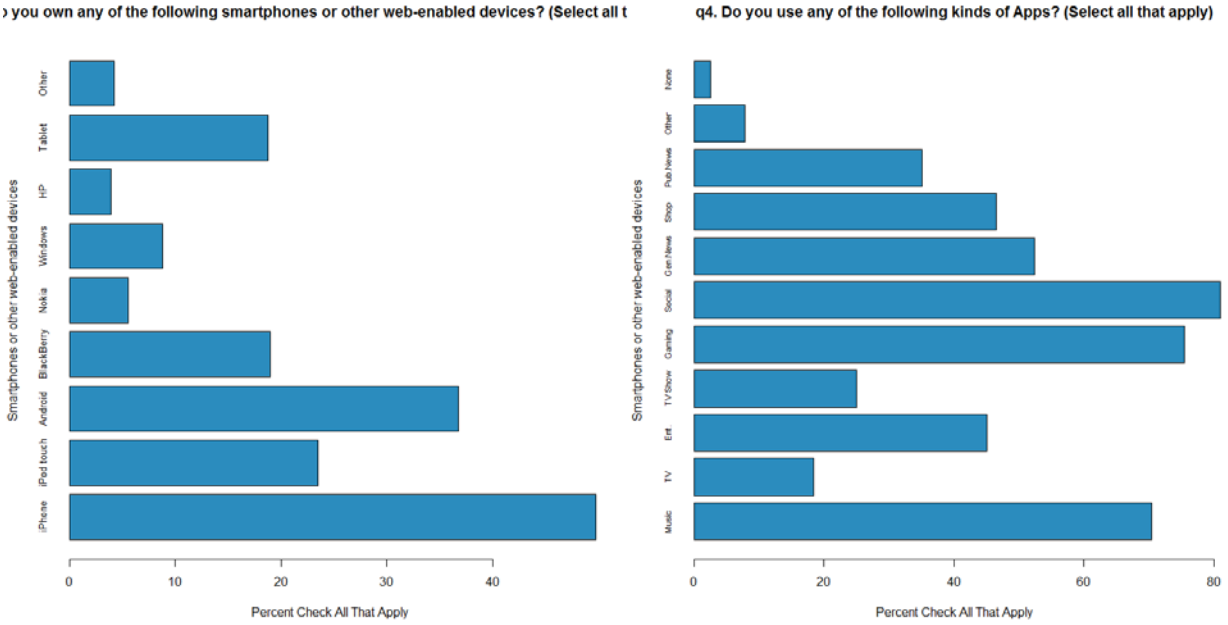
6. Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning. New York: Springer Science + Business Media.

Appendix

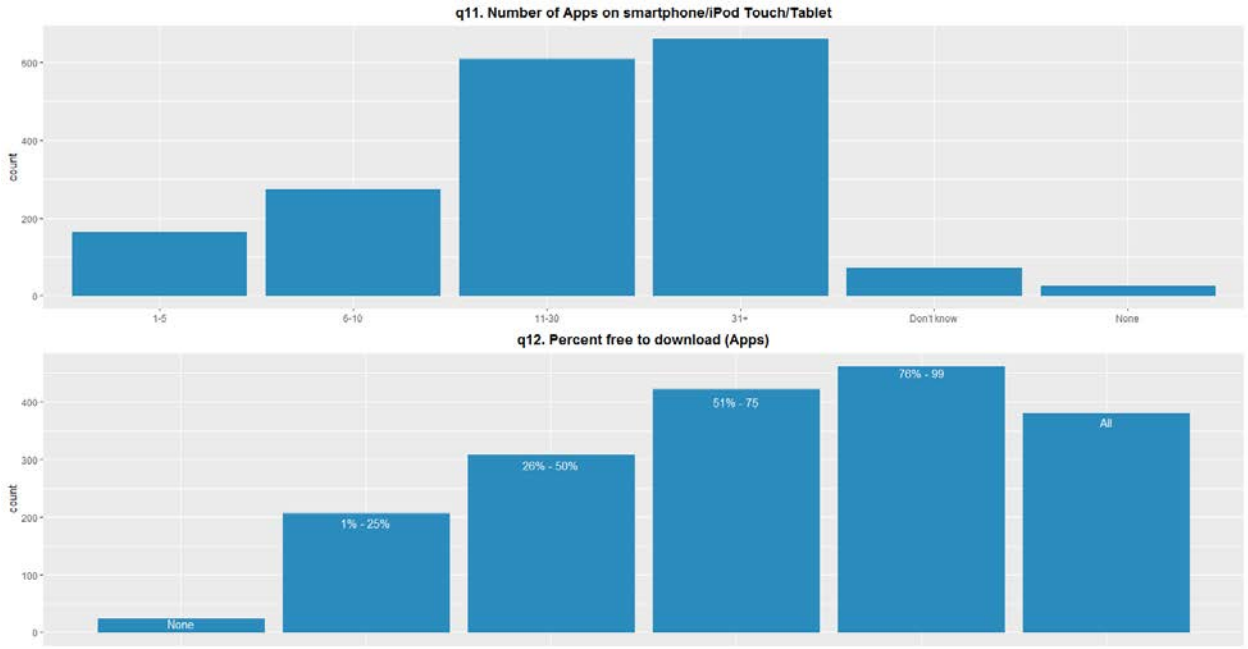
Appendix A



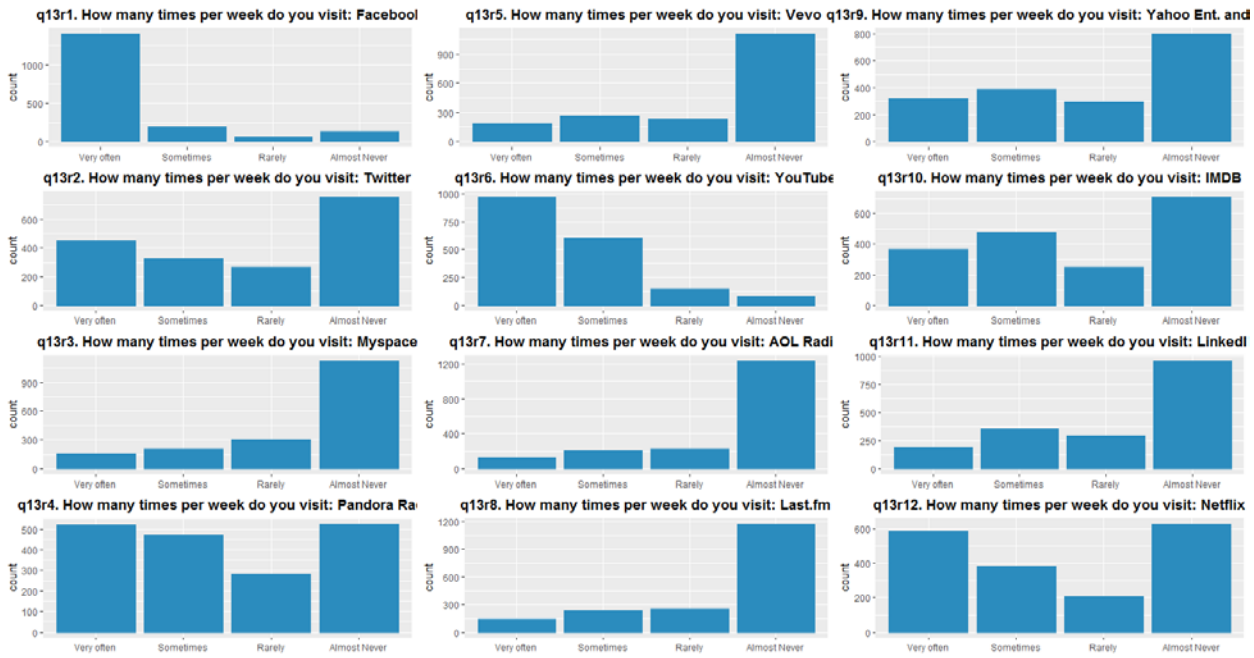
Appendix B



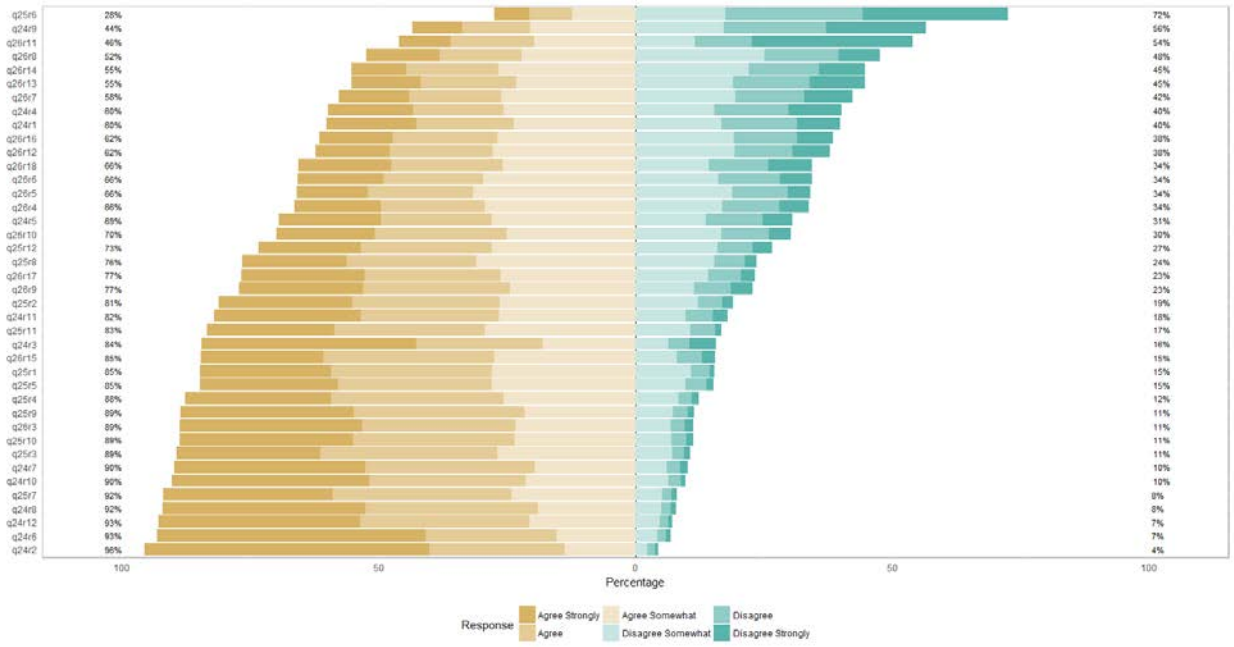
Appendix C



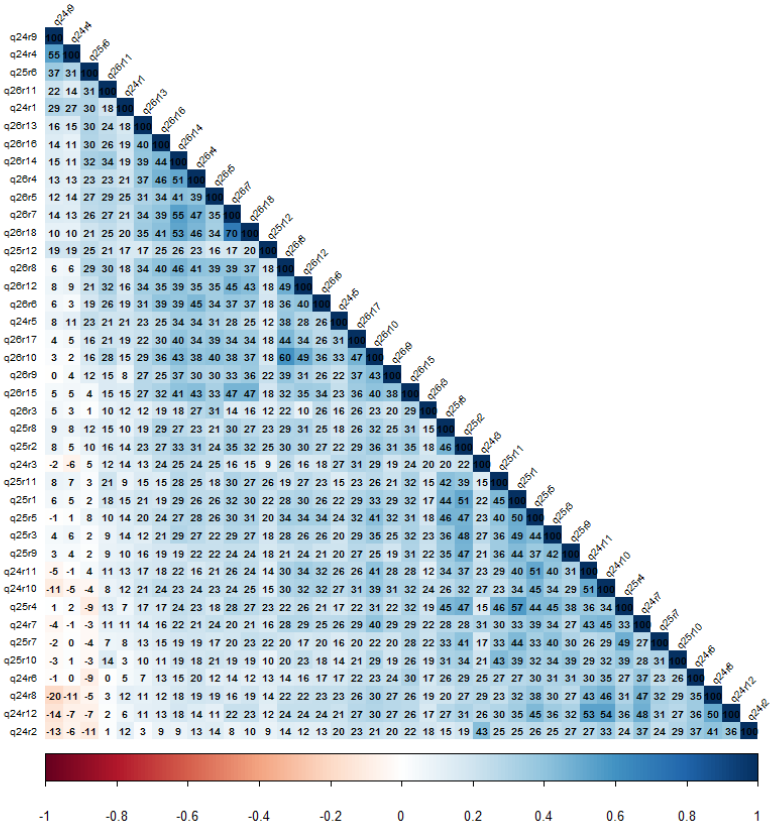
Appendix D



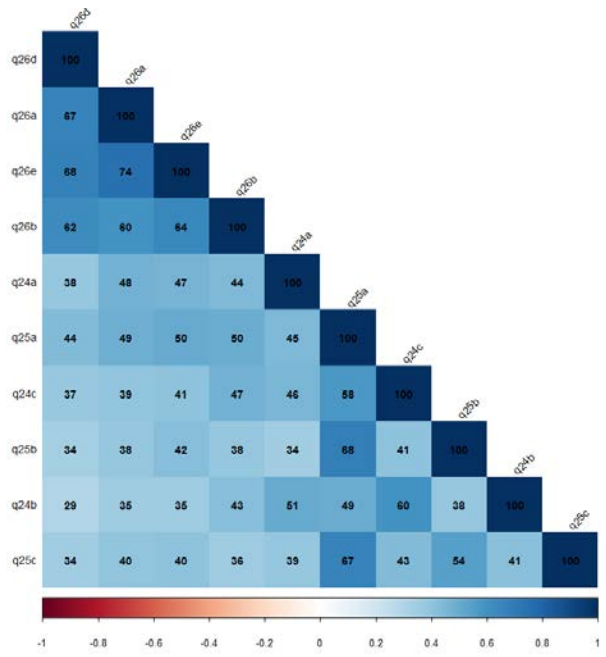
Appendix E



Appendix F



Appendix G



Appendix 1 (Median and Mean Responses for Grouped Questions, etc.)

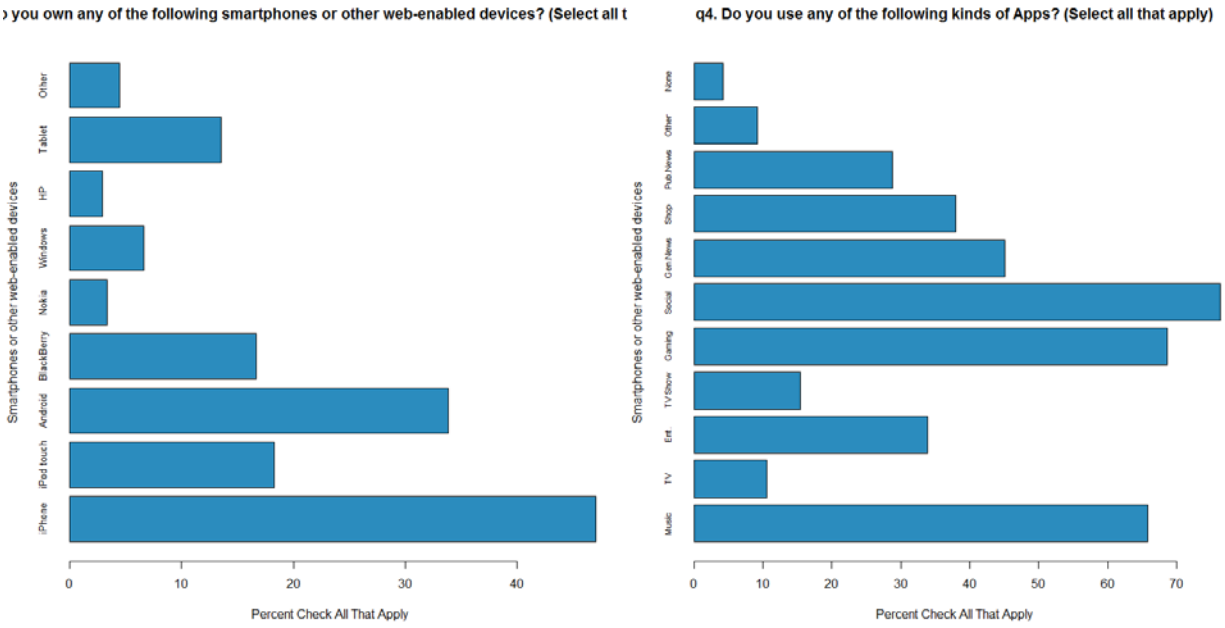
Median

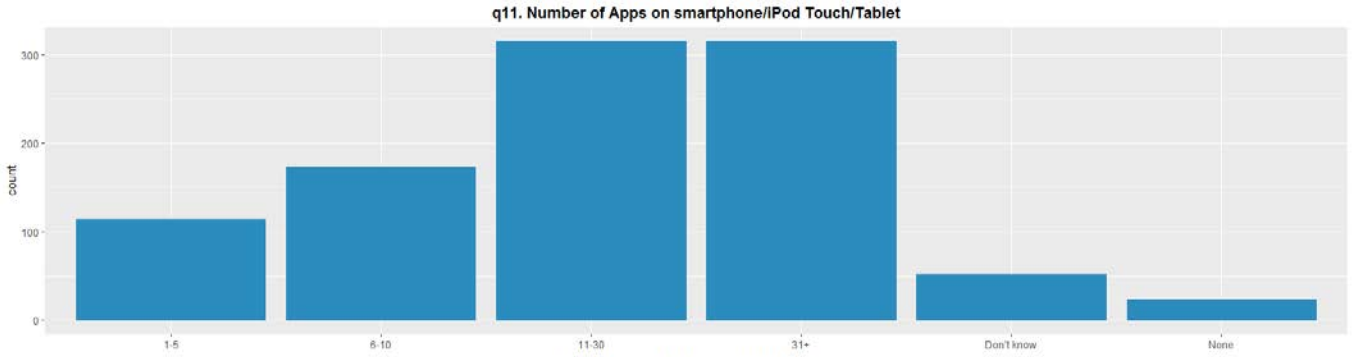
```
  byvar q24a q24b q24c q25a q25b     q25c q26a     q26b     q26d q26e    X cluster numdata$q1 numdata$q48 numdata$q49 numdata$q54
1     1  2.6  2.5  2.5  2.8     3 2.666667  3.4 3.666667 4.000000 3.25 1182       1          4          4          2          1
2     2  2.0  1.5  1.5  1.8     2 1.666667  2.2 2.000000 2.333333 2.00 1185       2          4          4          2          1
  numdata$q55 numdata$q56 numdata$q57
1           2           8           2
2           2           7           2

byvar q24a q24b q24c q25a q25b     q25c q26a     q26b     q26d q26e    X cluster numdata$q11 numdata$q12
    1  2.6  2.5  2.5  2.8     3 2.666667  3.4 3.666667 4.000000 3.25 1182       1           3           5
    2  2.0  1.5  1.5  1.8     2 1.666667  2.2 2.000000 2.333333 2.00 1185       2           3           4
```

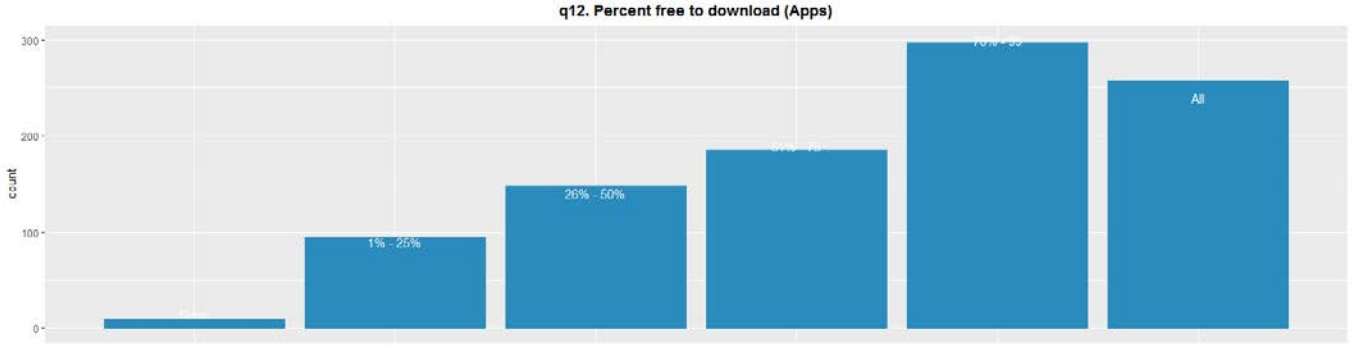
Mean

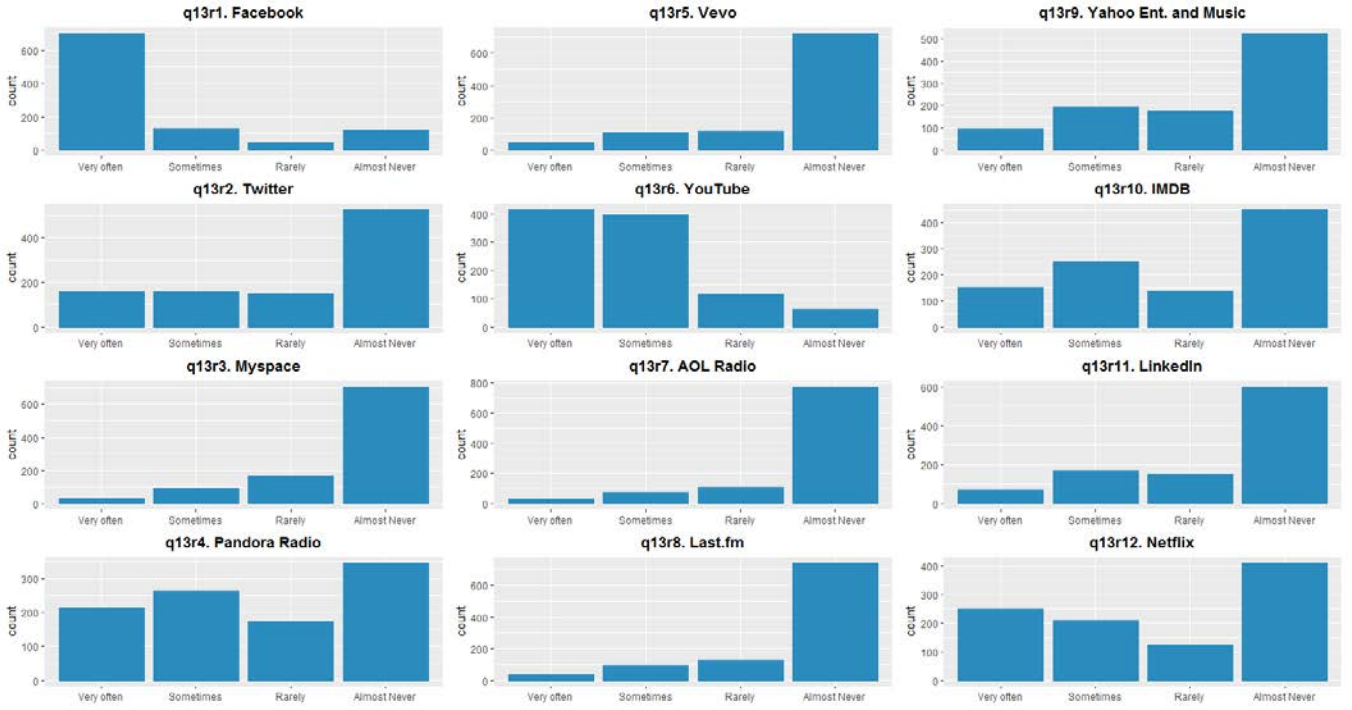
```
  byvar     q24a     q24b     q24c     q25a     q25b     q25c     q26a     q26b     q26d     q26e        X cluster numdata$q1
1     1 2.715641 2.422301 2.758325 2.841372 2.830979 2.676085 3.463774 3.611504 3.925328 3.403128 1150.637       1   4.880928
2     2 1.931026 1.567985 1.652040 1.753028 1.867738 1.756077 2.251916 2.096003 2.512979 2.094870 1224.817       2   4.168109
  numdata$q48 numdata$q49 numdata$q54 numdata$q55 numdata$q56 numdata$q57
1    3.619576    1.928355    1.679112    1.876892    7.855701    1.501514
2    3.548826    1.861557    1.751545    1.815822    7.893696    1.545117

  byvar     q24a     q24b     q24c     q25a     q25b     q25c     q26a     q26b     q26d     q26e        X cluster numdata$q11
1     1 2.715641 2.422301 2.758325 2.841372 2.830979 2.676085 3.463774 3.611504 3.925328 3.403128 1150.637       1    3.088799
2     2 1.931026 1.567985 1.652040 1.753028 1.867738 1.756077 2.251916 2.096003 2.512979 2.094870 1224.817       2    3.229913
  numdata$q12
1    4.450050
2    3.978986
```

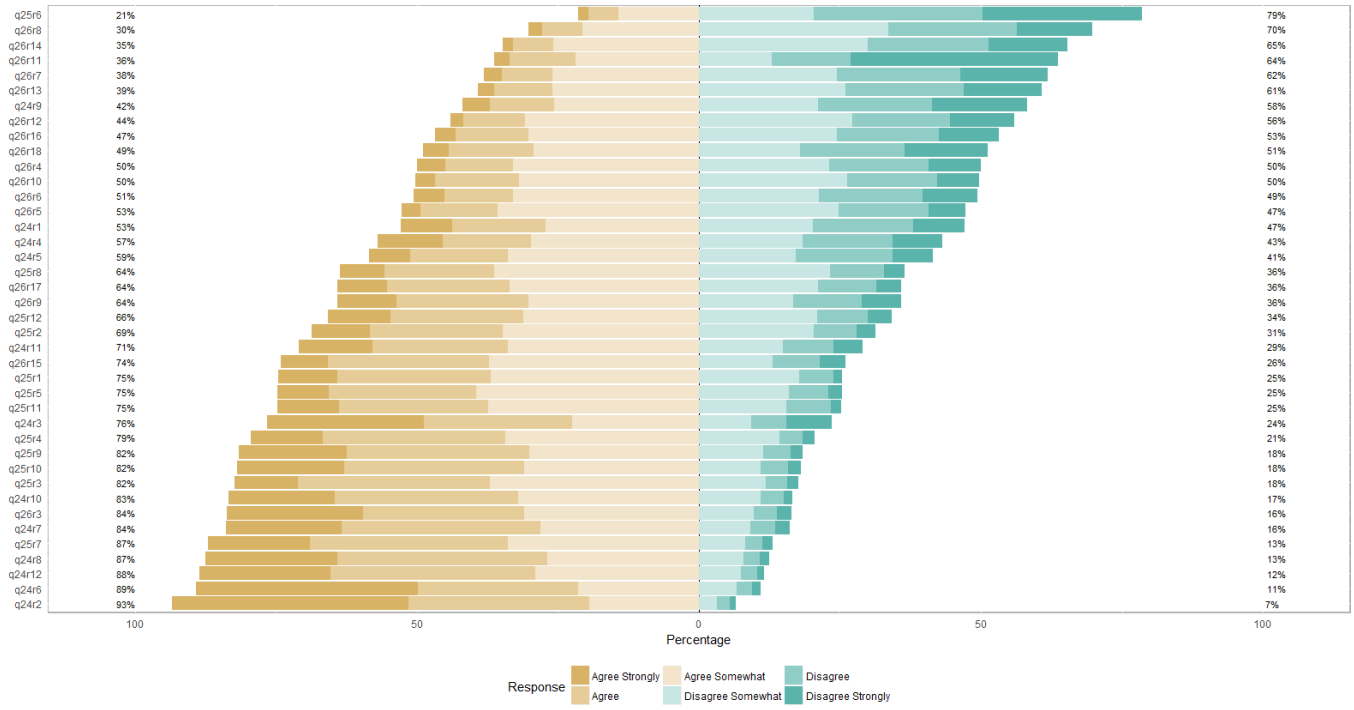
Appendix 2 (Segment 1)



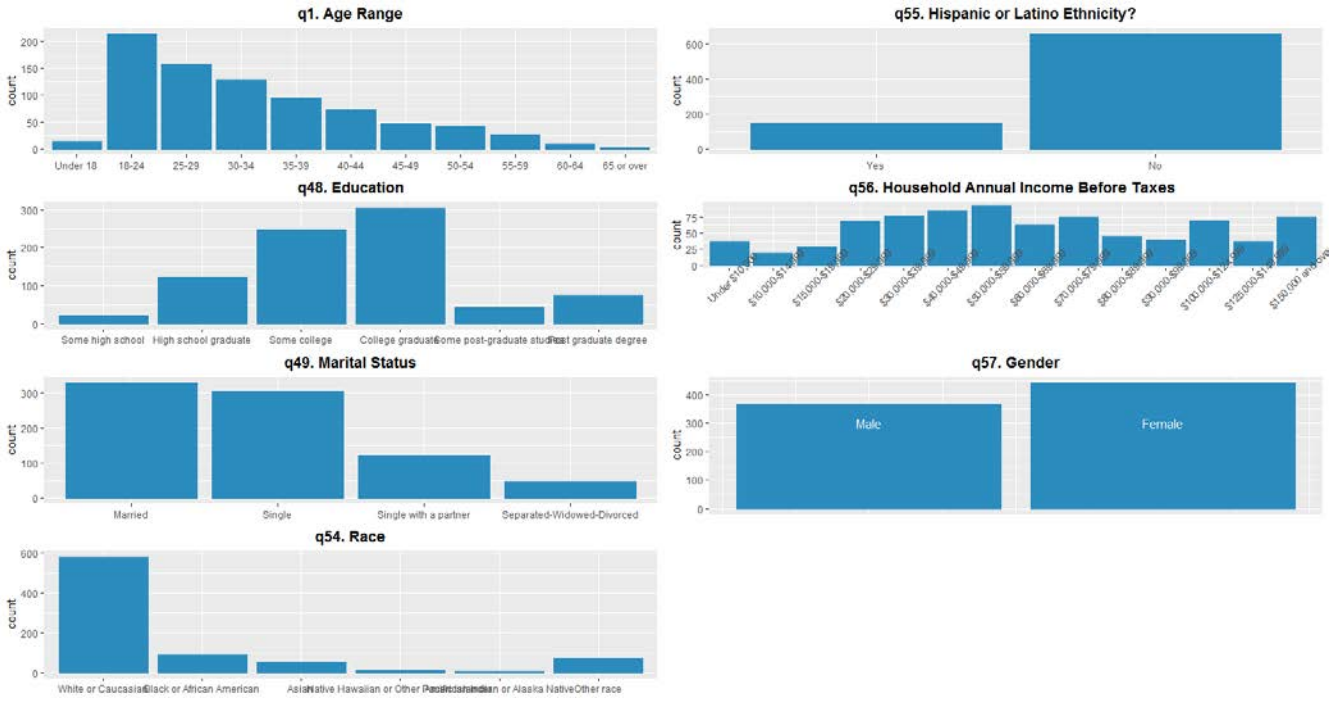

q11. Number of Apps on smartphone/IPod Touch/Tablet
count
1-5
6-10
11-30
31+
Don't know
None

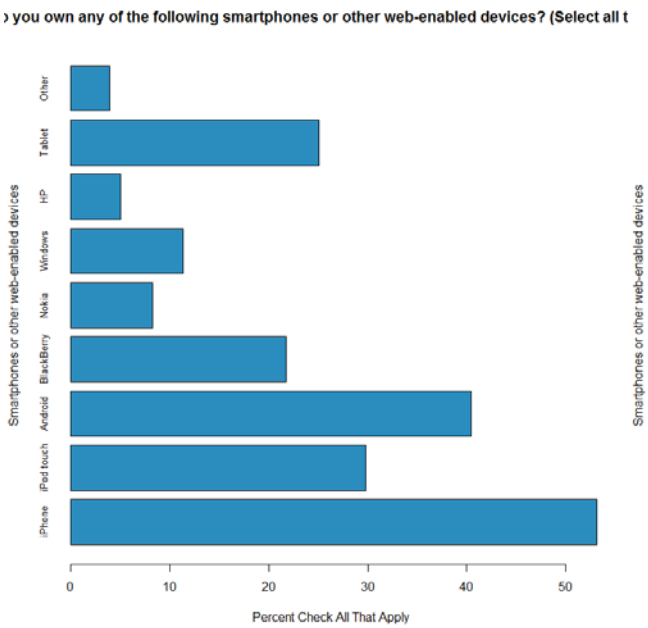

q12. Percent free to download (Apps)
count
1% - 25%
26% - 50%
All

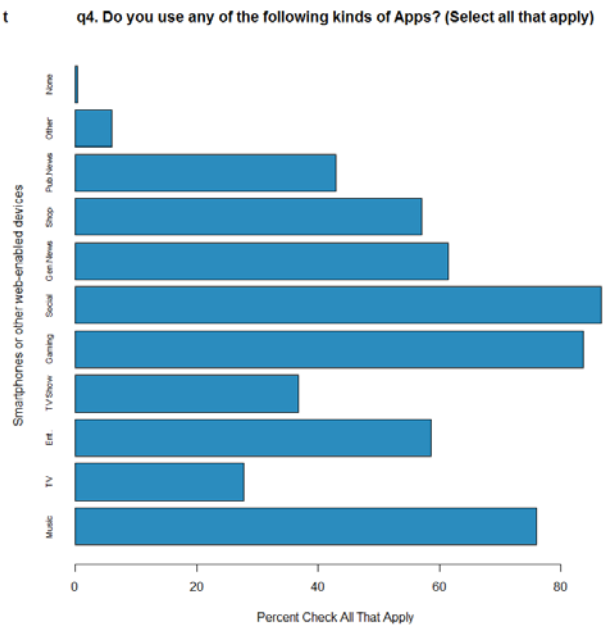

q13r1. Facebook
q13r5. Vevo
q13r9. Yahoo Ent. and Music
q13r2. Twitter
q13r6. YouTube
q13r10. IMDB
q13r3. Myspace
q13r7. AOL Radio
q13r11. LinkedIn
q13r4. Pandora Radio
q13r8. Last.fm
q13r12. Netflix
count
Very often
Sometimes
Rarely
Almost Never

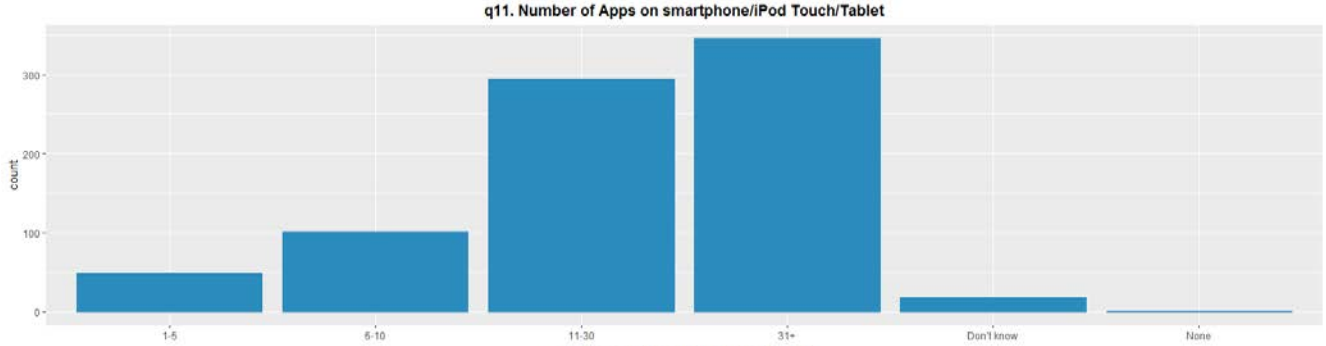


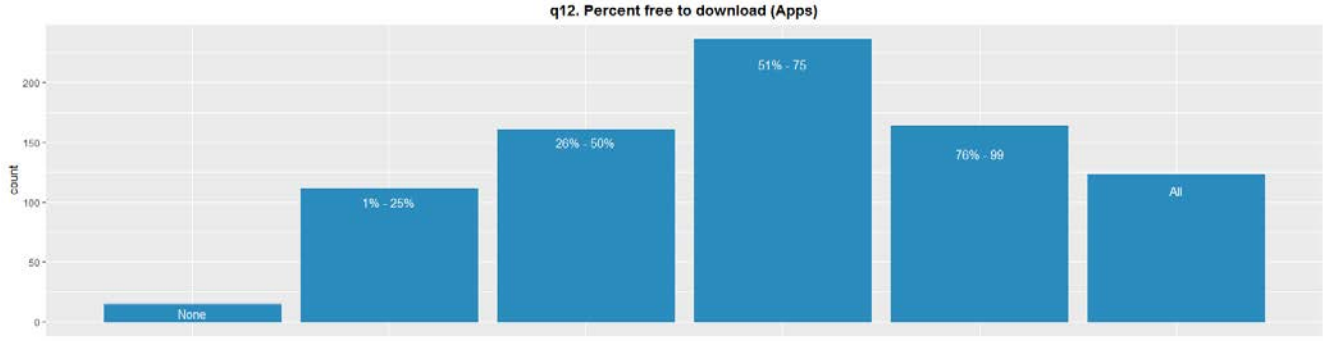
Appendix 3 (Segment 2)

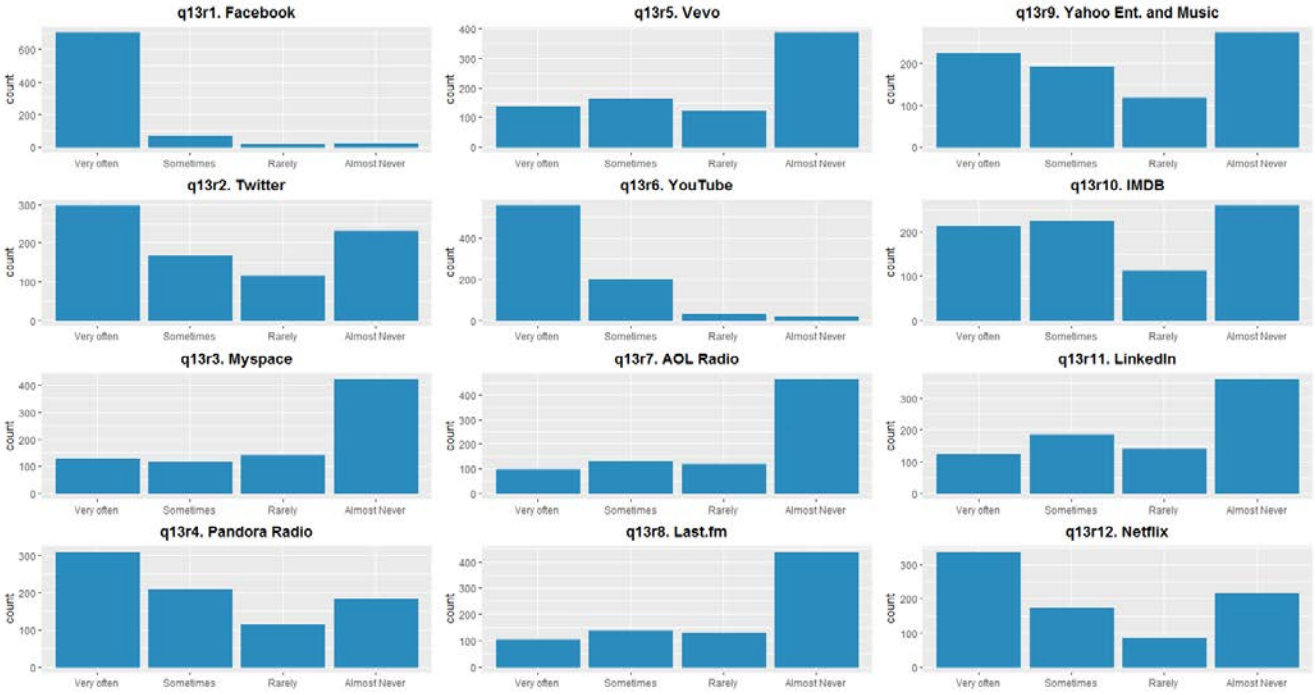


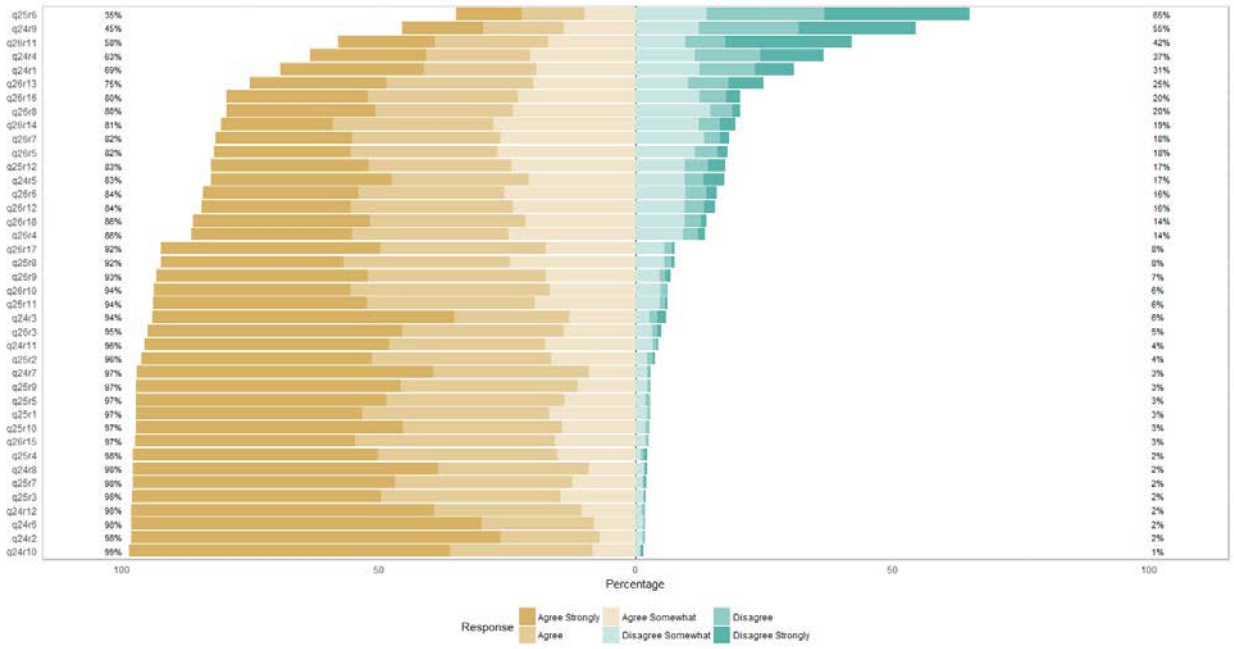

you own any of the following smartphones or other web-enabled devices? (Select all t
Smartphones or other web-enabled devices
Other
Tablet
HP
Windows
Nokia
BlackBerry
Android
iPod touch
iPhone
0
10
20
30
40
50
Percent Check All That Apply


q4. Do you use any of the following kinds of Apps? (Select all that apply)
Smartphones or other web-enabled devices
None
Other
Pub.News
Shop
Gen.News
Social
Gaming
TV Show
Ent.
TV
Music
0
20
40
60
80
Percent Check All That Apply


q11. Number of Apps on smartphone/iPod Touch/Tablet
count
300
200
100
0
1-5
6-10
11-30
31+
Don't know
None


q12. Percent free to download (Apps)
count
200
150
100
50
0
None
1% - 25%
26% - 50%
51% - 75
76% - 99
All


q13r1. Facebook
q13r5. Vevo
q13r9. Yahoo Ent. and Music
q13r2. Twitter
q13r6. YouTube
q13r10. IMDB
q13r3. Myspace
q13r7. AOL Radio
q13r11. LinkedIn
q13r4. Pandora Radio
q13r8. Last.fm
q13r12. Netflix
count
Very often
Sometimes
Rarely
Almost Never


Percentage
Response
Agree Strongly
Agree
Agree Somewhat
Disagree Somewhat
Disagree
Disagree Strongly

R Code

```
#Solo 1 Assignment Skeleton Code
#MSDS 450, Winter 2019

#Load Data
setwd("~/R/MSDS 450/Solo 1")

load("apphappyData.RData") #Load data
ls() #Load dataframe
## [1] "apphappy.3.labs.frame" "apphappy.3.num.frame" #

#Library will load the existing loaded package.
#Require will install or update when the package is not in our repository

require(cluster)
require(useful)
require(Hmisc)
library(HSAUR)
library(MVA)
library(HSAUR2)
library(fpc)
library(mclust)
library(lattice)
library(car)
library(proxy)
library(VIM) #Missingness Map
library(mice)
library(plyr)
library(likert) #Visualize Likert Scale Data
require(ggplot2)
library(factoextra) #Density Clustering
library(ggpubr) #Density Clustering
library(dbscan) #Density Clustering
library(fpc) #Density Clustering
library(reshape)
library(NbClust) #Provides 30 indexes for determining the optimal number of clusters in a data set

# Multiple plot function

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
 library(grid)

 # Make a list from the ... arguments and plotlist
 plots <- c(list(...), plotlist)

 numPlots = length(plots)

 # If layout is NULL, then use 'cols' to determine layout
 if (is.null(layout)) {
  # Make the panel
  # ncol: Number of columns of plots
  # nrow: Number of rows needed, calculated from # of cols
  layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                   ncol = cols, nrow = ceiling(numPlots/cols))
 }

 if (numPlots==1) {
  print(plots[[1]])
```

```
 } else {
  # Set up the page
  grid.newpage()
  pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

  # Make each plot, in the correct location
  for (i in 1:numPlots) {
   # Get the i,j matrix positions of the regions that contain this subplot
   matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

   print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                   layout.pos.col = matchidx$col))
  }
 }
}


numdata.df <- apphappy.3.num.frame
labsdata <- apphappy.3.labs.frame

######################################## Exploratory Data Analysis ########################################

#Descriptive Statistics
numdata.df$caseID <- as.numeric(numdata.df$caseID)

str(numdata.df) #1800 rows/people and 89 columns/variables/questions
head(numdata.df)
tail(numdata.df)
summary(numdata.df)
dim(numdata.df)
describe(labsdata) #Obtain percentages (proportions) for questions 24 to 26.
describe(numdata)

#Check for Missingness
sapply(numdata.df, function(x) sum(is.na(x)))
sum(is.na(numdata.df))
aggr_plot <- aggr(numdata.df, col=c('#9ecae1','#de2d26'), numbers=TRUE,prop=FALSE, sortVars=TRUE, labels=names(numdata.df), cex.axis=.5, gap=2, ylab=c
("Histogram of missing data","Pattern"))
#NAs: q5r1, q12, and q57. '

#Check missing data percentage
pMiss <- function(x){sum(is.na(x))/length(x)*100}
apply(numdata.df,2,pMiss)

#Run imputation
numdata.df <- mice(numdata.df,m=5,maxit=50,meth='pmm',seed=500)
summary(numdata.df)

#Check N/A values have been removed
numdata <- complete(numdata.df,1) #Create new data frame of imputed data.
apply(numdata,2,pMiss)
summary(numdata)
sapply(numdata, function(x) sum(is.na(x)))
str(numdata)

#Respondent Demographics

#q1. Which of the following best describes your age?
q1<-ggplot(labsdata) +
```

```
 geom_bar( aes(q1),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q1. Age Range" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q1

#q48. Which of the following best describes the highest level of education you have attained?
q48<-ggplot(labsdata) +
 geom_bar( aes(q48),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q48. Education" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q48


#q49. Which of the following best describe your marital status?
q49<-ggplot(labsdata) +
 geom_bar( aes(q49),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q49. Marital Status" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q49

#q54. Which of the following best describes your race?
q54<-ggplot(labsdata) +
 geom_bar( aes(q54),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q54. Race" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q54

#q55. Do you consider yourself to be of Hispanic or Latino ethnicity?
q55<-ggplot(labsdata) +
 geom_bar( aes(q55),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q55. Hispanic or Latino Ethnicity?" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q55


#q56. Which of the following best describes your household annual income before taxes?
q56<-ggplot(labsdata) +
 geom_bar( aes(q56),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q56. Household Annual Income Before Taxes" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q_56<-q56 + theme(axis.text.x = element_text(angle=45))

#q57. Please indicate your gender.
q57<-ggplot(numdata) +
 geom_bar( aes(q57),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q57. Gender" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank(),
       axis.text.x=element_blank(),
       axis.ticks.x=element_blank())
q57_annotate<-q57 + annotate("text", x = 1, y = 500, label = "Male",color="white") +
 annotate("text", x = 2, y = 500, label = "Female",color="white")
q57_annotate

#Mutiplots
multiplot(q1, q48, q49,q54,q55, q_56, q57_annotate,cols=2)

#Technology/App Usage

#q2. Do you own any of the following smartphones or other web-enabled devices? (Select all that apply)
```

```
#Pct.of.all.Resp = Out of all the responses how many chose it
#Pct.Check.All.That.Apply =  Out of each question, what is the percentage?

par(mfrow=c(1,2))
q2_MyMultResp_labels<-data.frame(numdata[3:11])

names(q2_MyMultResp_labels) <- make.names(names(q2_MyMultResp_labels))
colnames(q2_MyMultResp_labels)[1] <- "iPhone"
colnames(q2_MyMultResp_labels)[2] <- "iPod touch"
colnames(q2_MyMultResp_labels)[3] <- "Android"
colnames(q2_MyMultResp_labels)[4] <- "BlackBerry"
colnames(q2_MyMultResp_labels)[5] <- "Nokia"
colnames(q2_MyMultResp_labels)[6] <- "Windows"
colnames(q2_MyMultResp_labels)[7] <- "HP"
colnames(q2_MyMultResp_labels)[8] <- "Tablet"
colnames(q2_MyMultResp_labels)[9] <- "Other"

str(q2_MyMultResp_labels)

q2_MyMultResp<-data.frame(Freq = colSums(q2_MyMultResp_labels[1:9]),
          Pct.of.all.Resp = (colSums(q2_MyMultResp_labels[1:9])/sum(q2_MyMultResp_labels[1:9]))*100,
          Pct.Check.All.That.Apply = (colSums(q2_MyMultResp_labels[1:9])/nrow(q2_MyMultResp_labels[1:9]))*100)
q2_MyMultResp


barplot(q2_MyMultResp[[3]]
     ,names.arg=row.names(q2_MyMultResp)
     ,main = "q2. Do you own any of the following smartphones or other web-enabled devices? (Select all that apply)"
     ,xlab = "Percent Check All That Apply"
     ,ylab = "Smartphones or other web-enabled devices"
     ,col = "#2b8cbe"
     ,horiz = TRUE
     ,cex.names = 0.8)

#q4. Do you use any of the following kinds of Apps? (Select all that apply)
#Pct.of.all.Resp = Out of all the responses how many chose it
#Pct.Check.All.That.Apply =  Out of each question, what is the percentage?

q4_MyMultResp_labels<-data.frame(numdata[13:23])

names(q4_MyMultResp_labels) <- make.names(names(q4_MyMultResp_labels))
colnames(q4_MyMultResp_labels)[1] <- "Music"
colnames(q4_MyMultResp_labels)[2] <- "TV"
colnames(q4_MyMultResp_labels)[3] <- "Ent."
colnames(q4_MyMultResp_labels)[4] <- "TV Show"
colnames(q4_MyMultResp_labels)[5] <- "Gaming"
colnames(q4_MyMultResp_labels)[6] <- "Social"
colnames(q4_MyMultResp_labels)[7] <- "Gen.News"
colnames(q4_MyMultResp_labels)[8] <- "Shop"
colnames(q4_MyMultResp_labels)[9] <- "Pub.News"
colnames(q4_MyMultResp_labels)[10] <- "Other"
colnames(q4_MyMultResp_labels)[11] <- "None"

str(q4_MyMultResp_labels)

q4_MyMultResp<-data.frame(Freq = colSums(q4_MyMultResp_labels[1:11]),
               Pct.of.all.Resp = (colSums(q4_MyMultResp_labels[1:11])/sum(q4_MyMultResp_labels[1:11]))*100,
               Pct.Check.All.That.Apply = (colSums(q4_MyMultResp_labels[1:11])/nrow(q4_MyMultResp_labels[1:11]))*100)
q4_MyMultResp
```

```
barplot(q4_MyMultResp[[3]]
      ,names.arg=row.names(q4_MyMultResp)
      ,main = "q4. Do you use any of the following kinds of Apps? (Select all that apply)"
      ,xlab = "Percent Check All That Apply"
      ,ylab = "Smartphones or other web-enabled devices"
      ,col = "#2b8cbe"
      ,horiz = TRUE
      ,cex.names = 0.7)

par(mfrow=c(1,1))

#q11. How many Apps do you have on your smartphone/iPod Touch/Tablet? If you have more than of
#these device, please tell us the total number of Apps.
q11<-ggplot(labsdata) +
 geom_bar( aes(q11),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q11. Number of Apps on smartphone/iPod Touch/Tablet" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q11

#q12. Of your Apps, what percent were free to download?
q12<-ggplot(numdata) +
 geom_bar(aes(q12),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q12. Percent free to download (Apps)" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank(),
      axis.text.x=element_blank(),
      axis.ticks.x=element_blank())
q12_annotate<-q12_annotate<-q12 + annotate("text", x = 1, y = 15, label = "None", color="white")+
 annotate("text", x = 2, y = 190, label = "1% - 25%",color="white")+
 annotate("text", x = 3, y = 290, label = "26% - 50%",color="white")+
 annotate("text", x = 4, y = 400, label = "51% - 75",color="white")+
 annotate("text", x = 5, y = 450, label = "76% - 99",color="white")+
 annotate("text", x = 6, y = 365, label = "All",color="white")
q12_annotate

multiplot(q11,q12_annotate, cols=1)

#q13. How many times per week do you visit each of the following websites? (Select all that apply)
#Facebook
q13r1<-ggplot(labsdata) +
 geom_bar(aes(q13r1),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r1. How many times per week do you visit: Facebook" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r1

#Twitter
q13r2<-ggplot(labsdata) +
 geom_bar(aes(q13r2),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r2. How many times per week do you visit: Twitter" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r2

#Myspace
q13r3<-ggplot(labsdata) +
 geom_bar(aes(q13r3),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r3. How many times per week do you visit: Myspace" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r3
```

```
#Pandora Radio
q13r4<-ggplot(labsdata) +
 geom_bar(aes(q13r4),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r4. How many times per week do you visit: Pandora Radio" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r4

#Vevo
q13r5<-ggplot(labsdata) +
 geom_bar(aes(q13r5),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r5. How many times per week do you visit: Vevo" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r5

#YouTube
q13r6<-ggplot(labsdata) +
 geom_bar(aes(q13r6),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r6. How many times per week do you visit: YouTube" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r6

#AOL Radio
q13r7<-ggplot(labsdata) +
 geom_bar(aes(q13r7),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r7. How many times per week do you visit: AOL Radio" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r7

#Last.fm
q13r8<-ggplot(labsdata) +
 geom_bar(aes(q13r8),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r8. How many times per week do you visit: Last.fm" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r8

#Yahoo Entertainment and Music
q13r9<-ggplot(labsdata) +
 geom_bar(aes(q13r9),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r9. How many times per week do you visit: Yahoo Ent. and Music" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r9

#IMDB
q13r10<-ggplot(labsdata) +
 geom_bar(aes(q13r10),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r10. How many times per week do you visit: IMDB" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r10

#LinkedIn
q13r11<-ggplot(labsdata) +
 geom_bar(aes(q13r11),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r11. How many times per week do you visit: LinkedIn" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r11

#Netflix
q13r12<-ggplot(labsdata) +
 geom_bar(aes(q13r12),colour="#2b8cbe",fill="#2b8cbe") +
```

```
 ggtitle("q13r12. How many times per week do you visit: Netflix" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r12

multiplot(q13r1, q13r2, q13r3,q13r4, q13r5, q13r6,q13r7, q13r8, q13r9,q13r10, q13r11, q13r12, cols=3)

#Age Range vs. iPhone
a=table(numdata$q1,numdata$q2r1)
a
barplot(a, main = "Age Range vs. iPhone")

#Age Range vs. Android
b=table(numdata$q1,numdata$q2r3)
b
barplot(b, main = "Age Range vs. Android")

#Age Range vs. Music
c=table(numdata$q1,numdata$q4r1)
c
barplot(c, main = "Age Range vs. Music")

#Age Range vs. Gaming
d=table(numdata$q1,numdata$q4r5)
d
barplot(d, main = "Age Range vs. Gaming")

#Age Range vs. Social
e=table(numdata$q1,numdata$q4r6)
e
barplot(e, main = "Age Range vs. Social")

#Age Range vs. Entertainment
f=table(numdata$q1,numdata$q4r3)
f
barplot(f, main = "Age Range vs. Entertainment")

######

#Income vs. iPhone
a=table(numdata$q56,numdata$q2r1)
a
barplot(a, main = "Income vs. iPhone")

#Income vs. Android
b=table(numdata$q56,numdata$q2r3)
b
barplot(b, main = "Income vs. Android")

#Income vs. Music
c=table(numdata$q56,numdata$q4r1)
c
barplot(c, main = "Income vs. Music")

#Income vs. Gaming
d=table(numdata$q56,numdata$q4r5)
d
barplot(d, main = "Income vs. Gaming")

#Income vs. Social
```

```
e=table(numdata$q56,numdata$q4r6)
e
barplot(e, main = "Income vs. Social")

#Income vs. Entertainment
f=table(numdata$q56,numdata$q4r3)
f
barplot(f, main = "Income vs. Entertainment")

######

#Gender vs. iPhone
a=table(numdata$q57,numdata$q2r1)
a
barplot(a, main = "Gender vs. iPhone")

#Gender vs. Android
b=table(numdata$q57,numdata$q2r3)
b
barplot(b, main = "Gender vs. Android")

#Gender vs. Music
c=table(numdata$q57,numdata$q4r1)
c
barplot(c, main = "Gender vs. Music")

#Gender vs. Gaming
d=table(numdata$q57,numdata$q4r5)
d
barplot(d, main = "Gender vs. Gaming")

#Gender vs. Social
e=table(numdata$q57,numdata$q4r6)
e
barplot(e, main = "Gender vs. Social")

#Gender vs. Entertainment
f=table(numdata$q57,numdata$q4r3)
f
barplot(f, main = "Gender vs. Entertainment")

######

#Education vs. iPhone
a=table(numdata$q48,numdata$q2r1)
a
barplot(a, main = "Education vs. iPhone")

#Education vs. Android
b=table(numdata$q48,numdata$q2r3)
b
barplot(b, main = "Education vs. Android")

#Education vs. Music
c=table(numdata$q48,numdata$q4r1)
c
barplot(c, main = "Education vs. Music")

#Education vs. Gaming
```

```
d=table(numdata$q48,numdata$q4r5)
d
barplot(d, main = "Education vs. Gaming")

#Education vs. Social
e=table(numdata$q48,numdata$q4r6)
e
barplot(e, main = "Education vs. Social")

#Education vs. Entertainment
f=table(numdata$q48,numdata$q4r3)
f
barplot(f, main = "Education vs. Entertainment")

######

#Ethnicity vs. iPhone
a=table(numdata$q54,numdata$q2r1)
a
barplot(a, main = "Ethnicity vs. iPhone")

#Ethnicity vs. Android
b=table(numdata$q54,numdata$q2r3)
b
barplot(b, main = "Ethnicity vs. Android")

#Ethnicity vs. Music
c=table(numdata$q54,numdata$q4r1)
c
barplot(c, main = "Ethnicity vs. Music")

#Ethnicity vs. Gaming
d=table(numdata$q54,numdata$q4r5)
d
barplot(d, main = "Ethnicity vs. Gaming")

#Ethnicity vs. Social
e=table(numdata$q54,numdata$q4r6)
e
barplot(e, main = "Ethnicity vs. Social")

#Ethnicity vs. Entertainment
f=table(numdata$q54,numdata$q4r3)
f
barplot(f, main = "Ethnicity vs. Entertainment")

######################################## Create subsets ########################################

#Subset of questions 24 to 26 (attitudinal questions) for numdata
numsub <- subset(numdata, select=c("q24r1","q24r2","q24r3","q24r4","q24r5","q24r6","q24r7","q24r8","q24r9","q24r10","q24r11",
                                   "q24r12","q25r1","q25r2","q25r3","q25r4","q25r5","q25r6","q25r7","q25r8","q25r9","q25r10","q25r11",
                                   "q25r12","q26r3","q26r4","q26r5","q26r6","q26r7","q26r8","q26r9","q26r10","q26r11","q26r12",
                                   "q26r13","q26r14","q26r15","q26r16","q26r17","q26r18"))

str(numsub) #1800 observations and 40 variables
summary(numsub) #Likert scale, min is 1 and max is 6.


#Subset of questions 24 to 26 (attitudinal questions) for labsdata
```

```
labsub <- subset(labsdata, select=c("q24r1","q24r2","q24r3","q24r4","q24r5","q24r6","q24r7","q24r8","q24r9","q24r10","q24r11",
                        "q24r12","q25r1","q25r2","q25r3","q25r4","q25r5","q25r6","q25r7","q25r8","q25r9","q25r10","q25r11",
                        "q25r12","q26r3","q26r4","q26r5","q26r6","q26r7","q26r8","q26r9","q26r10","q26r11","q26r12",
                        "q26r13","q26r14","q26r15","q26r16","q26r17","q26r18"))

str(labsub) #1800 observations and 40 variables
summary(labsub) #Likert scale, min is 1 and max is 6.

#Questions 24 to 26 (Visualizing Likert Scale Data)
library(likert)

likert(labsub)
Result = likert(labsub)
plot(Result,type="bar")

### Correlation Matrix of subset ###

require(corrplot)
numsubcorrelation <- cor(numsub)

##Correlation Plot 3 w/ Numbers
corrplot(numsubcorrelation, method="shade", addCoef.col="black",
addCoefasPercent=TRUE ,type="lower", shade.col=NA, tl.col="black",
tl.srt=45,number.cex = 0.6,tl.cex = 0.6, addcolorlabel="no", order="AOE",insig = "p-value")

### PCA Plots ###

pca <-princomp(numsub)
plot(pca$scores[,1],pca$scores[,2]) #First 2 principal components only explain 0.3614 of the variation in the data.

names(pca)
str(pca)
summary(pca)
head(pca$scores)

#Find outliers where respondents selected all 1's
sortapca<-sort(pca$scores[,1], decreasing = TRUE)
sortapca
head(sortapca)

numsub["431",]
numsub["2176",]
numsub["1083",]
numsub["230",]
numsub["1870",]
numsub["1185",]

#Find outliers where respondents selected all 5's or 6's
sortapca2<-sort(pca$scores[,1], decreasing = FALSE)
sortapca2
head(sortapca2)

numsub["243",]
numsub["2391",]
numsub["858",]

#Remove outliers? #No
#numsub <- numsub[-c(27, 111, 156, 224, 259, 287, 380, 545,
          # 625, 647, 728, 960, 1046, 1122, 1153, 1224, 1227, 1315, 1336, 1359,
```

```
            #1478, 1534, 1573, 1597, 1791, 1466, 1001, 141, 1419, 844, 1614, 1035, 1282, 247, 801,
            #1040), ]

#############################################################
##create 'derived' variables - means of similar variables ###
#############################################################

attach(numsub)
numsub$q24a <- (q24r1+q24r2+q24r3+q24r5+q24r6)/5
numsub$q24b <- (q24r7+q24r8)/2
numsub$q24c <- (q24r10+q24r11)/2
numsub$q24d <- (q24r4+q24r9+q24r12)/3

numsub$q25a <- (q25r1+q25r2+q25r3+q25r4+q25r5)/5
numsub$q25b <- (q25r7+q25r8)/2
numsub$q25c <- (q25r9+q25r10+q25r11)/3
numsub$q25d <- (q25r6+q25r12)/2

numsub$q26a <- (q26r3+q26r4+q26r5+q26r6+q26r7)/5
numsub$q26b <- (q26r8+q26r9+q26r10)/3
numsub$q26c <- q26r11
numsub$q26d <- (q26r12+q26r13+q26r14)/3
numsub$q26e <- (q26r15+q26r16+q26r17+q26r18)/4

numsub2 <- subset(numsub, select=
            c("q24a","q24b","q24c",
              "q25a","q25b","q25c",
              "q26a","q26b","q26d","q26e"))

pca <-princomp(numsub2)
plot(pca$scores[,1],pca$scores[,2])
names(pca)
head(pca$scores)
str(pca$scores)
summary(pca)

##Correlation Plot 3 w/ Numbers
require(corrplot)
mcor <- cor(numsub2)
corrplot(mcor, method="shade", addCoef.col="black",
      addCoefasPercent=TRUE ,type="lower", shade.col=NA, tl.col="black",
      tl.srt=45,number.cex = 0.9,tl.cex = 0.9, addcolorlabel="no", order="AOE",insig = "p-value")

#####################################################################
################ Kmeans Cluster ###################
#####################################################################

#Create a 'scree' plot to determine the num of clusters
#'Sweep' through 1 to 15 clusters (standard, see slide 22)
wssplot <- function(numsub2, nc=15, seed=1234) {
 wss <- (nrow(numsub2)-1)*sum(apply(numsub2,2,var))
 for (i in 2:nc) {
  set.seed(seed)
  wss[i] <- sum(kmeans(numsub2, centers=i)$withinss)}
 plot(1:nc, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")}

wssplot(numsub2) #Elbow at 2, although no clear elbow. Try 5 for benchmark
```

```
# Elbow method (alternative); #intercept specifies elbow
fviz_nbclust(numsub2, kmeans, method = "wss") +
  geom_vline(xintercept = 2, linetype = 2)+
  labs(subtitle = "Elbow method")

# Silhouette method #Recommends 2
fviz_nbclust(numsub2, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")

# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
#set.seed(123)
#fviz_nbclust(numsub2, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
 # labs(subtitle = "Gap statistic method")

#NbClust: Determining the Best Number of Clusters in a Data Set (#Recommends 2)
#It provides 30 indexes for determining the optimal number of clusters in a data set and offers the best clustering scheme from different results to the user.
NbClust(data = numsub2, diss = NULL, distance = "euclidean",
        min.nc = 2, max.nc = 15, method = "kmeans")

### k-means with 5 clusters ###
clusterresults_5 <- kmeans(numsub2,5) #k-means
names(clusterresults_5) #sub objects of this result file and can access these with $
clusterresults_5$withinss #withinss for each of the clusters (e.g., sitting in cluster centroid, distance with all the people computing)
clusterresults_5$tot.withinss #Total withinss for the clusters
clusterresults_5$totss #Total withins for 1 cluster
clusterresults_5$betweenss #total withins-totss
clusterresults_5$size #Gives the count of people that are sitting in each cluster
rsquare <- clusterresults_5$betweenss/clusterresults_5$totss
rsquare #r-squared: 0.5128982

### Create a PC (Principal Componenet plot) ###

plot(clusterresults_5, data=numsub2) #PCA analysis, plot PC on x and y axis and then will plot the clusters
clusterresults_5$centers
head(clusterresults_5$cluster) #Shows cluster membership

### Create a Silhouette Plot ###
dissE <- daisy(numsub2)
names(dissE)
dE2   <- dissE^2
sk2   <- silhouette(clusterresults_5$cluster, dE2)
str(sk2)
plot(sk2) #Average Silhouette: 0.25

### k-means with 3 clusters ###
clusterresults_3 <- kmeans(numsub2,3) #k-means
names(clusterresults_3) #sub objects of this result file and can access these with $
clusterresults_3$withinss #withinss for each of the clusters (e.g., sitting in cluster centroid, distance with all the people computing)
clusterresults_3$tot.withinss #Total withinss for the clusters
clusterresults_3$totss #Total withins for 1 cluster
clusterresults_3$betweenss #total withins-totss
clusterresults_3$size #Gives the count of people that are sitting in each cluster
rsquare <- clusterresults_3$betweenss/clusterresults_3$totss
rsquare #r-squared: 0.4346976

### Create a PC (Principal Componenet plot) ###
```

```
plot(clusterresults_3, data=numsub2) #PCA analysis, plot PC on x and y axis and then will plot the clusters
clusterresults_3$centers
head(clusterresults_3$cluster) #Shows cluster membership

### Create a Silhouette Plot ###
dissE <- daisy(numsub2)
names(dissE)
dE2   <- dissE^2
sk2   <- silhouette(clusterresults_3$cluster, dE2)
str(sk2)
plot(sk2) #Average Silhouette: 0.32

### k-means with 2 clusters ###
clusterresults <- kmeans(numsub2,2) #k-means
names(clusterresults) #sub objects of this result file and can access these with $
clusterresults$withinss #withinss for each of the clusters (e.g., sitting in cluster centroid, distance with all the people computing)
clusterresults$tot.withinss #Total withinss for the clusters
clusterresults$totss #Total withins for 1 cluster
clusterresults$betweenss #total withins-totss
clusterresults$size #Gives the count of people that are sitting in each cluster
rsquare <- clusterresults$betweenss/clusterresults$totss
rsquare #r-squared: 0.34039

### Create a PC (Principal Componenet plot) ###

plot(clusterresults, data=numsub2) #PCA analysis, plot PC on x and y axis and then will plot the clusters
clusterresults$centers
head(clusterresults$cluster) #Shows cluster membership

### Create a Silhouette Plot ###
dissE <- daisy(numsub2)
names(dissE)
dE2   <- dissE^2
sk2   <- silhouette(clusterresults$cluster, dE2)
str(sk2)
plot(sk2) #Average Silhouette: 0.45, #Best

#Values near one mean that the observation is well placed in its cluster;
#values near 0 mean that it's likely that an observation might really belong in some other cluster.
#Within each cluster, the value for this measure is displayed from smallest to largest.
#If the silhouette plot shows values close to one for each observation, the fit was good;
#if there are many observations closer to zero, it's an indication that the fit was not good.
#The sihouette plot is very useful in locating groups in a cluster analysis that may not be doing a good job;
#in turn this information can be used to help select the proper number of clusters.
#If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.
#1 = Perfect, 0 is on the wall, and -1 is completely wrong.
#Good news, it's on positive side, more than 0 which is good.

### Produce csv Files ###
newdf <- as.data.frame(clusterresults$cluster) #creates dataframe for k-means cluster
write.csv(newdf, file = "clusterresults.csv") #cluster results assigned to each respondent
write.csv(numsub2, file = "numsub2.csv") #File of responses for subset of questions

#####################################################################
################### Hierarchical Clustering #########################
#####################################################################

### Single  Linkage ###
```

```
numsub2.dist = dist(numsub2) #Computes euclidean distances
require(maptree)
hclustmodel <- hclust(dist(numsub2), method = 'single')

#Create Cluster Dendrogram
plot(hclustmodel)

cut.2 <- cutree(hclustmodel, k=2)

clusterresults$centers
head(clusterresults$cluster) #Cluster Membership

#Create Cluster Dendrogram Post Cut
rect.hclust(hclustmodel,k=2, border="red")

#Create a Silhouette Plot
plot(silhouette(cut.2,numsub2.dist)) #Average Silhouette: 0.57, but extremely imbalanced. Not Recommended.
head(cut.2) #Classifys each respondent into a cluster.

### Average  Linkage ###
numsub2.dist = dist(numsub2) #Computes euclidean distances
require(maptree)
hclustmodel <- hclust(dist(numsub2), method = 'average')

#Create Cluster Dendrogram
plot(hclustmodel)

cut.2 <- cutree(hclustmodel, k=2)

clusterresults$centers
head(clusterresults$cluster) #Cluster Membership

#Create Cluster Dendrogram Post Cut
rect.hclust(hclustmodel,k=2, border="red")

#Create a Silhouette Plot
plot(silhouette(cut.2,numsub2.dist)) #Average Silhouette: 0.44, but extremely skewed and imbalanced. Not Recommended.
head(cut.2) #Classifys each respondent into a cluster.

### Complete Linkage ###
numsub2.dist = dist(numsub2) #Computes euclidean distances
require(maptree)
hclustmodel <- hclust(dist(numsub2), method = 'complete')

#Create Cluster Dendrogram
plot(hclustmodel)

cut.2 <- cutree(hclustmodel, k=2)

clusterresults$centers
head(clusterresults$cluster) #Cluster Membership

#Create Cluster Dendrogram Post Cut
rect.hclust(hclustmodel,k=2, border="red")

#Create a Silhouette Plot
plot(silhouette(cut.2,numsub2.dist)) #Average Silhouette: 0.20, somewhat imbalanced 2 to 1. Recommendation: K-means
head(cut.2) #Classifys each respondent into a cluster.
```

```
### ward.D2 ###
numsub2.dist = dist(numsub2) #Computes euclidean distances
require(maptree)
hclustmodel <- hclust(dist(numsub2), method = 'ward.D2')

#Create Cluster Dendrogram
plot(hclustmodel)

cut.2 <- cutree(hclustmodel, k=2)

clusterresults$centers
head(clusterresults$cluster) #Cluster Membership

#Create Cluster Dendrogram Post Cut
rect.hclust(hclustmodel,k=2, border="red")

#Create a Silhouette Plot
plot(silhouette(cut.2,numsub2.dist)) #Average Silhouette: 0.24, somewhat imbalanced. Recommendation: K-means
head(cut.2) #Classifys each respondent into a cluster.

### Produce csv Files ###
write.csv(cut.2, file = "cut2results.csv")

#Between Sum of Squares and Total Sum of Squares
require(proxy)
numsub2mat <- as.matrix(numsub2)
overallmean <- matrix(apply(numsub2mat,2,mean),nrow=1)
overallmean
TSS <- sum(dist(numsub2mat,overallmean)^2)
TSS #Compute TSS 17010.06

#Weighted Sum Statistic
combcutdata <- cbind(numsub2,cut.2)
head(combcutdata)

require(reshape)
combcutdata <- rename(combcutdata, c(cut.2="cluster"))
head(combcutdata)

clust1 <- subset(combcutdata, cluster == 1)
clust1 <- subset(clust1, select=c("q24a","q24b","q24c","q25a","q25b","q25c",
                          "q26a","q26b","q26d","q26e"))
clust1 <- as.matrix(clust1,rowby=T)
dim(clust1)
clust1mean <- matrix(apply(clust1,2,mean),nrow=1)
dim(clust1mean)
dis1 <- sum(dist(clust1mean,clust1)^2)

clust2 <- subset(combcutdata, cluster == 2)
clust2 <- subset(clust2, select=c("q24a","q24b","q24c","q25a","q25b","q25c",
                          "q26a","q26b","q26d","q26e"))
clust2 <- as.matrix(clust2,rowby=T)
clust2mean <- matrix(apply(clust2,2,mean),nrow=1)
dis2 <- sum(dist(clust2mean,clust2)^2)

WSS <- sum(dis1,dis2)
WSS #12078.64

BSS <- TSS - WSS
```

```
BSS #4931.425

### calculating the % of Between SS/ Total SS ###
rsquare <- BSS/TSS
rsquare #0.2899122, compared to k-mean's rsquare of: 0.34039; Note: Use r-square and silouette to compare different methods.

#########################################################
################### PAM Method ##########################
#########################################################

my.k.choices <- 2:8 #Sweep through using average silhouette width (k-means uses WSS)
avg.sil.width <- rep(0, times=length(my.k.choices))
for (ii in (1:length(my.k.choices)) ){
  avg.sil.width[ii] <- pam(numsub2, k=my.k.choices[ii])$silinfo$avg.width
}
print( cbind(my.k.choices, avg.sil.width)) #Optimal number is 2, given average silo of 0.1399

clusterresultsPAM <-pam(numsub2,2)
summary(clusterresultsPAM)

#Create cluster plot of PAM
plot(clusterresultsPAM, which.plots=1) #Different symbols denote various clusters; overlap

#Create a Silhouette Plot
plot(clusterresultsPAM, which.plots=2) #Average Silhouette: 0.27
#Cluster sizes somewhat imbalanced.
#PAM is better than hierarchical, but k-means is still better in terms of average silouette.

#########################################################
######## Density based clustering from Lecture ###
#########################################################

## Find optimal values of 2 parameters epsilon, minpts, knee of the scree plot
dbscan::kNNdistplot(numsub2, k =  5)
abline(h = 2.5, lty = 2)

# Compute DBSCAN (package:fpc)
set.seed(123)
db <- fpc::dbscan(numsub2, eps = 2.5, MinPts = 5) #Use MinPts of 5,6, or 7

# Plot DBSCAN results (package:factoextra)
fviz_cluster(db, data = numsub2, stand = FALSE,
        ellipse = FALSE, show.clust.cent = FALSE,
        geom = "point",palette = "light blue", ggtheme = theme_classic())

print(db) ## 0 means outliers, other values belong to cluster

# Cluster membership. Noise/outlier observations are coded as 0
# A random subset is shown
db$cluster[sample(1:1089, 20)]

#Note:
#Showing how to do this model or mix-model for real-life, but for Solo 1 it's not appropriate.
#As a result, instead, focus on k-means, hierarchical, and PAM. Gives intro idea EDA and clustering.
#Deep dive, take this as a starting point, build upon, and work on it, and then Wednesday for deep
#dive for Solo 1.

####################
## Model based clustering
```

```
##################
library(mclust)
mclust_2 <- Mclust(numsub2,2)
plot(fit,data=numsub2, what="density") # plot results
#plot(mclust_2,data=numsub2, what="BIC") # plot results

summary(mclust_2) # display the best model

dev.off()
dissE <- daisy(numsub2)
names(dissE)
dE2   <- dissE^2
sk2   <- silhouette(mclust_2$classification, dE2)
str(sk2)
plot(sk2)

#Not appropriate given likert scale data.

################################################
## Comparison of cluster results  #############
################################################
##corrected or adjusted rand index lies between 0 & 1
## perfect match between 2 clustering methods means 1, no match means 0
## any number in between represents 'kind of' % of matched pairs

#k-means vs. pam
clstat <- cluster.stats(numsub2.dist, clusterresults$cluster, clusterresultsPAM$cluster)
names(clstat)
clstat$corrected.rand #0.6701434

#Hierarchical vs. k-means
clstat <- cluster.stats(numsub2.dist, clusterresults$cluster, cut.2)
clstat$corrected.rand #0.500193

#Hierarchical vs. pam
clstat <- cluster.stats(numsub2.dist, clusterresultsPAM$cluster, cut.2)
clstat$corrected.rand #0.601595

######################################## Profiling #############################################################
#Create a dataset that combines original data with cluster information, used to create profiles
newdf <- read.csv("clusterresults.csv") #File that contains cluster results assigned to each respondent

#Demographics: Age Range (q1), Education (q48), Marital status (q49), Race (q54), Ethnicity(q55), Income(q56), Gender(q57)
combdata <- cbind(numsub2 ,newdf,numdata$q1,
            numdata$q48,
            numdata$q49,
            numdata$q54,
            numdata$q55,
            numdata$q56,
            numdata$q57)
head(combdata)
require(reshape)
combdata <- rename(combdata, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
aggregate(combdata,by=list(byvar=combdata$cluster), median) #For each cluster, show median response for the subset of q's

#Technology/App Usage
combdata_consump <- cbind(numsub2 ,newdf,numdata$q11,numdata$q12)
head(combdata_consump)
require(reshape)
```

```
combdata_consump <- rename(combdata_consump, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
aggregate(combdata_consump,by=list(byvar=combdata_consump$cluster), median) #For each cluster, show median response for the subset of q's

#By looking at the profile, you will be able to decide what kind of products or services you would want to provide
#for each of these segments.

######################################## Segment 1 ############################################################

#Subsetting on Segment 1 (numdata)
combdata_seg <- cbind(newdf,numdata)
combdata_seg <- rename(combdata_seg, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
combdata_segment1<-combdata_seg[combdata_seg$cluster ==1, ]
str(combdata_segment1)

#Subsetting on Segment 1 (labsdata)
combdata_seg_labs <- cbind(newdf,labsdata)
combdata_seg_labs <- rename(combdata_seg_labs, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
combdata_segment1_labs<-combdata_seg_labs[combdata_seg_labs$cluster ==1, ]
str(combdata_segment1_labs)

#Subsetting on Segment 1 (labsdata) Survey
combdata_seg_labs_survey <- cbind(labsub,newdf)
combdata_seg_labs_survey <- rename(combdata_seg_labs_survey, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
combdata_seg_labs_survey<-combdata_seg_labs_survey[combdata_seg_labs_survey$cluster ==1, ]
combdata_seg_labs_survey <-data.frame(combdata_seg_labs_survey[1:40])
str(combdata_seg_labs_survey)

#Respondent Demographics

#q1. Which of the following best describes your age?
q1<-ggplot(combdata_segment1_labs) +
 geom_bar( aes(q1),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q1. Age Range" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q1

#q48. Which of the following best describes the highest level of education you have attained?
q48<-ggplot(combdata_segment1_labs) +
 geom_bar( aes(q48),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q48. Education" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q48

#q49. Which of the following best describe your marital status?
q49<-ggplot(combdata_segment1_labs) +
 geom_bar( aes(q49),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q49. Marital Status" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q49

#q54. Which of the following best describes your race?
q54<-ggplot(combdata_segment1_labs) +
 geom_bar( aes(q54),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q54. Race" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q54

#q55. Do you consider yourself to be of Hispanic or Latino ethnicity?
q55<-ggplot(combdata_segment1_labs) +
```

```
 geom_bar( aes(q55),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q55. Hispanic or Latino Ethnicity?" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q55

#q56. Which of the following best describes your household annual income before taxes?
q56<-ggplot(combdata_segment1_labs) +
 geom_bar( aes(q56),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q56. Household Annual Income Before Taxes" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q_56<-q56 + theme(axis.text.x = element_text(angle=45))
q_56

#q57. Please indicate your gender.
q57<-ggplot(combdata_segment1) +
 geom_bar( aes(q57),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q57. Gender" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank(),
     axis.text.x=element_blank(),
     axis.ticks.x=element_blank())
q57_annotate<-q57 + annotate("text", x = 1, y = 300, label = "Male",color="white") +
 annotate("text", x = 2, y = 300, label = "Female",color="white")
q57_annotate

#Mutiplots
multiplot(q1, q48, q49,q54,q55, q_56, q57_annotate,cols=2)

#Technology/App Usage

par(mfrow=c(1,2))
#q2. Do you own any of the following smartphones or other web-enabled devices? (Select all that apply)
#Pct.of.all.Resp = Out of all the responses how many chose it
#Pct.Check.All.That.Apply =  Out of each question, what is the percentage?

q2_MyMultResp_labels<-data.frame(combdata_segment1[5:13])

names(q2_MyMultResp_labels) <- make.names(names(q2_MyMultResp_labels))
colnames(q2_MyMultResp_labels)[1] <- "iPhone"
colnames(q2_MyMultResp_labels)[2] <- "iPod touch"
colnames(q2_MyMultResp_labels)[3] <- "Android"
colnames(q2_MyMultResp_labels)[4] <- "BlackBerry"
colnames(q2_MyMultResp_labels)[5] <- "Nokia"
colnames(q2_MyMultResp_labels)[6] <- "Windows"
colnames(q2_MyMultResp_labels)[7] <- "HP"
colnames(q2_MyMultResp_labels)[8] <- "Tablet"
colnames(q2_MyMultResp_labels)[9] <- "Other"

str(q2_MyMultResp_labels)

q2_MyMultResp<-data.frame(Freq = colSums(q2_MyMultResp_labels[1:9]),
                Pct.of.all.Resp = (colSums(q2_MyMultResp_labels[1:9])/sum(q2_MyMultResp_labels[1:9]))*100,
                Pct.Check.All.That.Apply = (colSums(q2_MyMultResp_labels[1:9])/nrow(q2_MyMultResp_labels[1:9]))*100)
q2_MyMultResp

barplot(q2_MyMultResp[[3]]
     ,names.arg=row.names(q2_MyMultResp)
     ,main = "q2. Do you own any of the following smartphones or other web-enabled devices? (Select all that apply)"
     ,xlab = "Percent Check All That Apply"
     ,ylab = "Smartphones or other web-enabled devices"
```

```
      ,col = "#2b8cbe"
      ,horiz = TRUE
      ,cex.names = 0.8)

#q4. Do you use any of the following kinds of Apps? (Select all that apply)
#Pct.of.all.Resp = Out of all the responses how many chose it
#Pct.Check.All.That.Apply =  Out of each question, what is the percentage?

q4_MyMultResp_labels<-data.frame(combdata_segment1[15:25])

names(q4_MyMultResp_labels) <- make.names(names(q4_MyMultResp_labels))
colnames(q4_MyMultResp_labels)[1] <- "Music"
colnames(q4_MyMultResp_labels)[2] <- "TV"
colnames(q4_MyMultResp_labels)[3] <- "Ent."
colnames(q4_MyMultResp_labels)[4] <- "TV Show"
colnames(q4_MyMultResp_labels)[5] <- "Gaming"
colnames(q4_MyMultResp_labels)[6] <- "Social"
colnames(q4_MyMultResp_labels)[7] <- "Gen.News"
colnames(q4_MyMultResp_labels)[8] <- "Shop"
colnames(q4_MyMultResp_labels)[9] <- "Pub.News"
colnames(q4_MyMultResp_labels)[10] <- "Other"
colnames(q4_MyMultResp_labels)[11] <- "None"

str(q4_MyMultResp_labels)

q4_MyMultResp<-data.frame(Freq = colSums(q4_MyMultResp_labels[1:11]),
                  Pct.of.all.Resp = (colSums(q4_MyMultResp_labels[1:11])/sum(q4_MyMultResp_labels[1:11]))*100,
                  Pct.Check.All.That.Apply = (colSums(q4_MyMultResp_labels[1:11])/nrow(q4_MyMultResp_labels[1:11]))*100)
q4_MyMultResp

barplot(q4_MyMultResp[[3]]
      ,names.arg=row.names(q4_MyMultResp)
      ,main = "q4. Do you use any of the following kinds of Apps? (Select all that apply)"
      ,xlab = "Percent Check All That Apply"
      ,ylab = "Smartphones or other web-enabled devices"
      ,col = "#2b8cbe"
      ,horiz = TRUE
      ,cex.names = 0.7)

par(mfrow=c(1,1))

#q11. How many Apps do you have on your smartphone/iPod Touch/Tablet? If you have more than of
#these device, please tell us the total number of Apps.
q11<-ggplot(combdata_segment1_labs) +
 geom_bar( aes(q11),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q11. Number of Apps on smartphone/iPod Touch/Tablet" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q11

#q12. Of your Apps, what percent were free to download?
q12<-ggplot(combdata_segment1) +
 geom_bar(aes(q12),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q12. Percent free to download (Apps)" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank(),
      axis.text.x=element_blank(),
      axis.ticks.x=element_blank())
q12_annotate<-q12 + annotate("text", x = 1, y = 15, label = "None", color="white")+
 annotate("text", x = 2, y = 90, label = "1% - 25%",color="white")+
 annotate("text", x = 3, y = 140, label = "26% - 50%",color="white")+
```

```
  annotate("text", x = 4, y = 190, label = "51% - 75",color="white")+
  annotate("text", x = 5, y = 300, label = "76% - 99",color="white")+
  annotate("text", x = 6, y = 240, label = "All",color="white")
q12_annotate

multiplot(q11,q12_annotate, cols=1)

#q13. How many times per week do you visit each of the following websites? (Select all that apply)
#Facebook
q13r1<-ggplot(combdata_segment1_labs) +
  geom_bar(aes(q13r1),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r1. Facebook" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r1

#Twitter
q13r2<-ggplot(combdata_segment1_labs) +
  geom_bar(aes(q13r2),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r2. Twitter" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r2

#Myspace
q13r3<-ggplot(combdata_segment1_labs) +
  geom_bar(aes(q13r3),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r3. Myspace" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r3

#Pandora Radio
q13r4<-ggplot(combdata_segment1_labs) +
  geom_bar(aes(q13r4),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r4. Pandora Radio" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r4

#Vevo
q13r5<-ggplot(combdata_segment1_labs) +
  geom_bar(aes(q13r5),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r5. Vevo" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r5

#YouTube
q13r6<-ggplot(combdata_segment1_labs) +
  geom_bar(aes(q13r6),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r6. YouTube" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r6

#AOL Radio
q13r7<-ggplot(combdata_segment1_labs) +
  geom_bar(aes(q13r7),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r7. AOL Radio" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r7

#Last.fm
q13r8<-ggplot(combdata_segment1_labs) +
```

```
 geom_bar(aes(q13r8),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r8. Last.fm" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r8

#Yahoo Entertainment and Music
q13r9<-ggplot(combdata_segment1_labs) +
 geom_bar(aes(q13r9),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r9. Yahoo Ent. and Music" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r9

#IMDB
q13r10<-ggplot(combdata_segment1_labs) +
 geom_bar(aes(q13r10),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r10. IMDB" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r10

#LinkedIn
q13r11<-ggplot(combdata_segment1_labs) +
 geom_bar(aes(q13r11),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r11. LinkedIn" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r11

#Netflix
q13r12<-ggplot(combdata_segment1_labs) +
 geom_bar(aes(q13r12),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r12. Netflix" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r12

multiplot(q13r1, q13r2, q13r3,q13r4, q13r5, q13r6,q13r7, q13r8, q13r9,q13r10, q13r11, q13r12, cols=3)

#Questions 24 to 26 (Visualizing Likert Scale Data)
library(likert)
likert(combdata_seg_labs_survey)
Result = likert(combdata_seg_labs_survey)
plot(Result,type="bar")

######################################## Segment 2 ############################################################

#Subsetting on Segment 2 (numdata)
combdata_seg <- cbind(newdf,numdata)
combdata_seg <- rename(combdata_seg, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
combdata_segment2<-combdata_seg[combdata_seg$cluster ==2, ]
str(combdata_segment2)

#Subsetting on Segment 2 (labsdata)
combdata_seg_labs <- cbind(newdf,labsdata)
combdata_seg_labs <- rename(combdata_seg_labs, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
combdata_segment2_labs<-combdata_seg_labs[combdata_seg_labs$cluster ==2, ]
str(combdata_segment2_labs)

#Subsetting on Segment 2 (labsdata) Survey
combdata_seg_labs_survey2 <- cbind(labsub,newdf)
combdata_seg_labs_survey2 <- rename(combdata_seg_labs_survey2, c(clusterresults.cluster="cluster")) #rename clusterresults.cluster to cluster
combdata_seg_labs_survey2<-combdata_seg_labs_survey2[combdata_seg_labs_survey2$cluster ==2, ]
```

```
combdata_seg_labs_survey2 <-data.frame(combdata_seg_labs_survey2[1:40])
str(combdata_seg_labs_survey2)

#Respondent Demographics

#q1. Which of the following best describes your age?
q1<-ggplot(combdata_segment2_labs) +
 geom_bar( aes(q1),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q1. Age Range" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q1

#q48. Which of the following best describes the highest level of education you have attained?
q48<-ggplot(combdata_segment2_labs) +
 geom_bar( aes(q48),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q48. Education" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q48

#q49. Which of the following best describe your marital status?
q49<-ggplot(combdata_segment2_labs) +
 geom_bar( aes(q49),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q49. Marital Status" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q49

#q54. Which of the following best describes your race?
q54<-ggplot(combdata_segment2_labs) +
 geom_bar( aes(q54),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q54. Race" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q54

#q55. Do you consider yourself to be of Hispanic or Latino ethnicity?
q55<-ggplot(combdata_segment2_labs) +
 geom_bar( aes(q55),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q55. Hispanic or Latino Ethnicity?" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q55

#q56. Which of the following best describes your household annual income before taxes?
q56<-ggplot(combdata_segment2_labs) +
 geom_bar( aes(q56),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q56. Household Annual Income Before Taxes" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q_56<-q56 + theme(axis.text.x = element_text(angle=45))

#q57. Please indicate your gender.
q57<-ggplot(combdata_segment2) +
 geom_bar( aes(q57),colour="#2b8cbe",fill="#2b8cbe" ) +
 ggtitle("q57. Gender" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank(),
       axis.text.x=element_blank(),
       axis.ticks.x=element_blank())
q57_annotate<-q57 + annotate("text", x = 1, y = 300, label = "Male",color="white") +
 annotate("text", x = 2, y = 300, label = "Female",color="white")
q57_annotate

#Mutiplots
```

```
multiplot(q1, q48, q49,q54,q55, q_56, q57_annotate,cols=2)

par(mfrow=c(1,2))
#Technology/App Usage

#q2. Do you own any of the following smartphones or other web-enabled devices? (Select all that apply)
#Pct.of.all.Resp = Out of all the responses how many chose it
#Pct.Check.All.That.Apply =  Out of each question, what is the percentage?

q2_MyMultResp_labels<-data.frame(combdata_segment2[5:13])

names(q2_MyMultResp_labels) <- make.names(names(q2_MyMultResp_labels))
colnames(q2_MyMultResp_labels)[1] <- "iPhone"
colnames(q2_MyMultResp_labels)[2] <- "iPod touch"
colnames(q2_MyMultResp_labels)[3] <- "Android"
colnames(q2_MyMultResp_labels)[4] <- "BlackBerry"
colnames(q2_MyMultResp_labels)[5] <- "Nokia"
colnames(q2_MyMultResp_labels)[6] <- "Windows"
colnames(q2_MyMultResp_labels)[7] <- "HP"
colnames(q2_MyMultResp_labels)[8] <- "Tablet"
colnames(q2_MyMultResp_labels)[9] <- "Other"

str(q2_MyMultResp_labels)

q2_MyMultResp<-data.frame(Freq = colSums(q2_MyMultResp_labels[1:9]),
                Pct.of.all.Resp = (colSums(q2_MyMultResp_labels[1:9])/sum(q2_MyMultResp_labels[1:9]))*100,
                Pct.Check.All.That.Apply = (colSums(q2_MyMultResp_labels[1:9])/nrow(q2_MyMultResp_labels[1:9]))*100)
q2_MyMultResp


barplot(q2_MyMultResp[[3]]
     ,names.arg=row.names(q2_MyMultResp)
     ,main = "q2. Do you own any of the following smartphones or other web-enabled devices? (Select all that apply)"
     ,xlab = "Percent Check All That Apply"
     ,ylab = "Smartphones or other web-enabled devices"
     ,col = "#2b8cbe"
     ,horiz = TRUE
     ,cex.names = 0.8)

#q4. Do you use any of the following kinds of Apps? (Select all that apply)
#Pct.of.all.Resp = Out of all the responses how many chose it
#Pct.Check.All.That.Apply =  Out of each question, what is the percentage?

q4_MyMultResp_labels<-data.frame(combdata_segment2[15:25])

names(q4_MyMultResp_labels) <- make.names(names(q4_MyMultResp_labels))
colnames(q4_MyMultResp_labels)[1] <- "Music"
colnames(q4_MyMultResp_labels)[2] <- "TV"
colnames(q4_MyMultResp_labels)[3] <- "Ent."
colnames(q4_MyMultResp_labels)[4] <- "TV Show"
colnames(q4_MyMultResp_labels)[5] <- "Gaming"
colnames(q4_MyMultResp_labels)[6] <- "Social"
colnames(q4_MyMultResp_labels)[7] <- "Gen.News"
colnames(q4_MyMultResp_labels)[8] <- "Shop"
colnames(q4_MyMultResp_labels)[9] <- "Pub.News"
colnames(q4_MyMultResp_labels)[10] <- "Other"
colnames(q4_MyMultResp_labels)[11] <- "None"

str(q4_MyMultResp_labels)
```

```
q4_MyMultResp<-data.frame(Freq = colSums(q4_MyMultResp_labels[1:11]),
                  Pct.of.all.Resp = (colSums(q4_MyMultResp_labels[1:11])/sum(q4_MyMultResp_labels[1:11]))*100,
                  Pct.Check.All.That.Apply = (colSums(q4_MyMultResp_labels[1:11])/nrow(q4_MyMultResp_labels[1:11]))*100)
q4_MyMultResp

barplot(q4_MyMultResp[[3]]
      ,names.arg=row.names(q4_MyMultResp)
      ,main = "q4. Do you use any of the following kinds of Apps? (Select all that apply)"
      ,xlab = "Percent Check All That Apply"
      ,ylab = "Smartphones or other web-enabled devices"
      ,col = "#2b8cbe"
      ,horiz = TRUE
      ,cex.names = 0.7)

par(mfrow=c(1,1))

#q11. How many Apps do you have on your smartphone/iPod Touch/Tablet? If you have more than of
#these device, please tell us the total number of Apps.
q11<-ggplot(combdata_segment2_labs) +
 geom_bar( aes(q11),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q11. Number of Apps on smartphone/iPod Touch/Tablet" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q11

#q12. Of your Apps, what percent were free to download?
q12<-ggplot(combdata_segment2) +
 geom_bar(aes(q12),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q12. Percent free to download (Apps)" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank(),
     axis.text.x=element_blank(),
     axis.ticks.x=element_blank())

q12_annotate<-q12 + annotate("text", x = 1, y = 7, label = "None", color="white")+
 annotate("text", x = 2, y = 100, label = "1% - 25%",color="white")+
 annotate("text", x = 3, y = 150, label = "26% - 50%",color="white")+
 annotate("text", x = 4, y = 215, label = "51% - 75",color="white")+
 annotate("text", x = 5, y = 140, label = "76% - 99",color="white")+
 annotate("text", x = 6, y = 110, label = "All",color="white")
q12_annotate
multiplot(q11,q12_annotate, cols=1)


#q13. How many times per week do you visit each of the following websites? (Select all that apply)
#Facebook
q13r1<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r1),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r1. Facebook" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r1

#Twitter
q13r2<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r2),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r2. Twitter" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r2

#Myspace
```

```
q13r3<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r3),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r3. Myspace" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r3

#Pandora Radio
q13r4<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r4),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r4. Pandora Radio" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r4

#Vevo
q13r5<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r5),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r5. Vevo" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r5

#YouTube
q13r6<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r6),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r6. YouTube" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r6

#AOL Radio
q13r7<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r7),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r7. AOL Radio" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r7

#Last.fm
q13r8<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r8),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r8. Last.fm" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r8

#Yahoo Entertainment and Music
q13r9<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r9),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r9. Yahoo Ent. and Music" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r9

#IMDB
q13r10<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r10),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r10. IMDB" ) +
 theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r10

#LinkedIn
q13r11<-ggplot(combdata_segment2_labs) +
 geom_bar(aes(q13r11),colour="#2b8cbe",fill="#2b8cbe") +
 ggtitle("q13r11. LinkedIn" ) +
```

```
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r11

#Netflix
q13r12<-ggplot(combdata_segment2_labs) +
  geom_bar(aes(q13r12),colour="#2b8cbe",fill="#2b8cbe") +
  ggtitle("q13r12. Netflix" ) +
  theme(plot.title=element_text(lineheight=0.8, face="bold", hjust=0.5),axis.title.x=element_blank())
q13r12

multiplot(q13r1, q13r2, q13r3,q13r4, q13r5, q13r6,q13r7, q13r8, q13r9,q13r10, q13r11, q13r12, cols=3)

#Questions 24 to 26 (Visualizing Likert Scale Data)
library(likert)
likert(combdata_seg_labs_survey2)
Result = likert(combdata_seg_labs_survey2)
plot(Result,type="bar")
```