Solo 1: Segmentation Analysis

MSPA PREDICT 450-DL-55 LEC

Darryl Buswell

1 Introduction

This document presents results of the first assignment for the Masters of Science in Predictive Analytics course: PREDICT 450. This assessment required the student to undertake a 'general attitudinal post hoc segmentation analysis' using survey data of over 1,600 respondents. The aim of this segmentation analysis was to identifying distinct groups which a development firm named AppHappy can target using segment-specific marketing strategies and tactics.

Questions from the survey data relating to respondents' technological acceptance, personality characteristics and purchasing behavior were selected as basis variables for the segmentation analysis. These variables were then used in applying hierarchical and non-hierarchical clustering methods with various distance metrics. Upon identifying relevant segements, data exploration techniques were then leveraged in order to identify any demographic or customer behavior characteristics which are able to distinguish the identified clusters. Finally, a classification model was recommended which should enable AppHappy to classify future customers into segments, including those customers whom the the company has no survey data.

2 Data

This assessment leverages survey data of 1,663 respondents conducted by the Consumer Spy Corporation (CSC). The data is stored within a R data file, which contains two R data frames. The first of these has numerically coded survey response data, while the second has response data coded in the character strings of the questionnaire's value labels.

Based on a dictionary of the survey questions, it appears the original survey had at least 57 unique questions, while the survey subset used for this assessment included 16 questions. Questions in the subset are varied, including those related to respondent demographics, purchasing behavior, and online browsing habits.

3 Customer Segmentation

3.1 Basis Variables

App Happy has requested a 'general attitudinal post hoc segmentation analysis'. As such, attitudinal item variables will make up the basis variables to be used for this assessment. Such variables should be able to 'characterize' groups based on attitudes, and therefore allow identification of groups that might comprise of useful attitudinal segments. From a review of the subset of survey data, it appears that Questions 24, 25 and 26 make the best candidates for basis variables. Question 24 relates to the respondents level of technological adoption or technological acceptance, Question 25 relates to personality characteristics, and Question 26 relates to purchasing behavior.

3.2 Clustering Method

Clustering algorithms can be separated into two main classes, non-hierarchical and hierarchical. For non-hierarchical clustering, the desired number of clusters is specified in advance in order to set the number of cluster centers. Each data point is then assigned to its nearest cluster center by minimizing or maximizing a desired criterion, with cluster centroids iteratively recalculated until they remain stable (P. Tan 2006). Under this method, the relationship between clusters remains largely undetermined. That is, the analyst gains a representation of data clusters based on only their predetermined number of centers. In hierarchical clustering however, the number of clusters need not be specified in advance. Instead, pairs of clusters are repetitively

linked until every data point is included in a hierarchy of cluster relationships. The analyst then has the freedom to assess the full array of cluster options and their hierarchy of relationships using a dendrogram.

For this assessment, we employ both non-hierarchical and hierarchical methods in order to segment by our set of attitudinal variables. For non-hierarchical clustering, we employ two R functions. The first is the 'k-means' function as part of the 'stats' R package, which is used in order to perform k-means clustering (STHDA 2016). The second is the 'pam' function as part of the 'cluster' R package, which is used in order to perform partitioning of k-clusters around medoids (Maechler 2014). This is a more robust form of k-means, particularly to outliers, as it minimizes the sum of dissimilarities instead of a sum of squared distance (P. Rousseeuw 1990). Finally, for hierarchical clustering, we employ the 'hcut' function as part of the 'factoextra' R package (F. Mundt 2016). We do this using the 'ward.D' agglomeration technique and use the same package to generate dendograms to aid in deriving an appropriate number of clusters (STHDA 2016).

3.3 Distance Metric

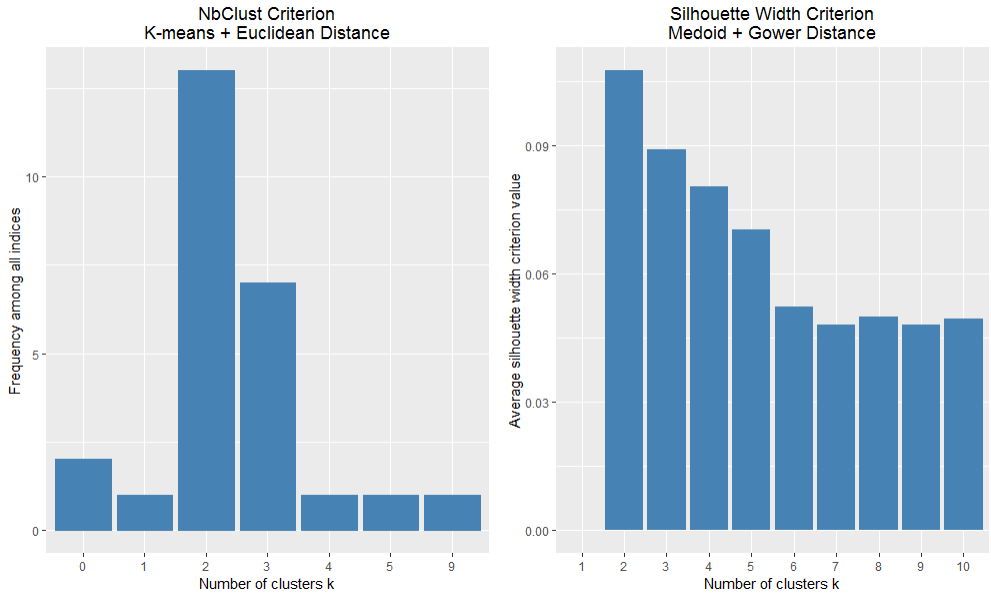
The data used in cluster analysis can be of continuous or categorical type, however the type of data will influence the use of dissimilarity (or distance) metric. For continuous variables, the most common distance measure is the Euclidean distance, while for categorical or mixed type data, a Gower distance metric would be more appropriate (Cornish 2007).

All three of the identified basis variables for this assessment employ a six level Likert scale: Agree Strongly, Agree, Agree Somewhat, Disagree Somewhat, Disagree, and Disagree Strongly. In that case, each variable can be considered to be of categorical type. Although we have the option of using a numeric equivalent of these variables from a numeric equivalent data source, there are still obvious risks in assuming the data to be of continuous type. Fortunately, the 'daisy' function as part of the 'cluster' R package is able to compute distances using the Gower distance metric (A. Struyf 2016). As such, we conduct cluster analysis both on the assumption of continuous data using the Euclidean distance metric and also provide a comparative cluster analysis assuming non-continuous data using the Gower distance metric. This is done using both hierarchical and non-hierarchical methods, resulting in the employment of four combinations of clustering methods and distance metric.

3.4 Non-hierarchical Clustering

As mentioned previously, non-hierarchical clustering methods require the number of clusters be nominated prior to implementation of the clustering routine. There are however, a number of criteria available in order to aid the analyst in determining an appropriate number of clusters prior to employing the clustering technique. Perhaps the most robust of these can be found in the 'NbClust' R package (E. Feit 2015). This package provides the ability to employ up to 30 separate cluster criterion measures as a form of index. We apply the 'NbClust' function to the k-means clustering technique using the Euclidean distance metric and show index frequencies for a range of clusters below. Unfortunately, the 'NbClust' package is unable to accommodate the medoid clustering technique, so we instead leverage the 'pamk' R package to derive a silhouette width criterion in order to assess the optimal number of clusters under this technique.

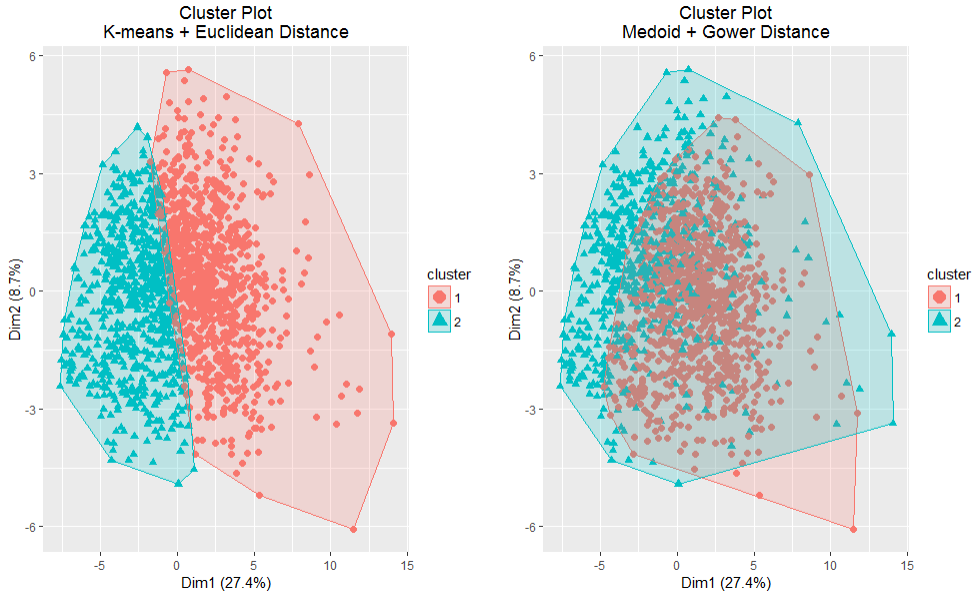
Figure 3.4.1 Non-hierarchical Cluster Criterion



We can see that the use of two clusters is reported to be optimal for both clustering methods, while the use of three clusters is reported to be the next best option.

With the number of clusters pre-determined based on these criteria, we are able to move on to employing each non-hierarchical clustering method. Below we show scatter plots which highlight the non-hierarchical cluster assignments based on a two-dimensional reduction of our original set of attitudinal survey responses.

Figure 3.4.2 Non-hierarchical Cluster Plots

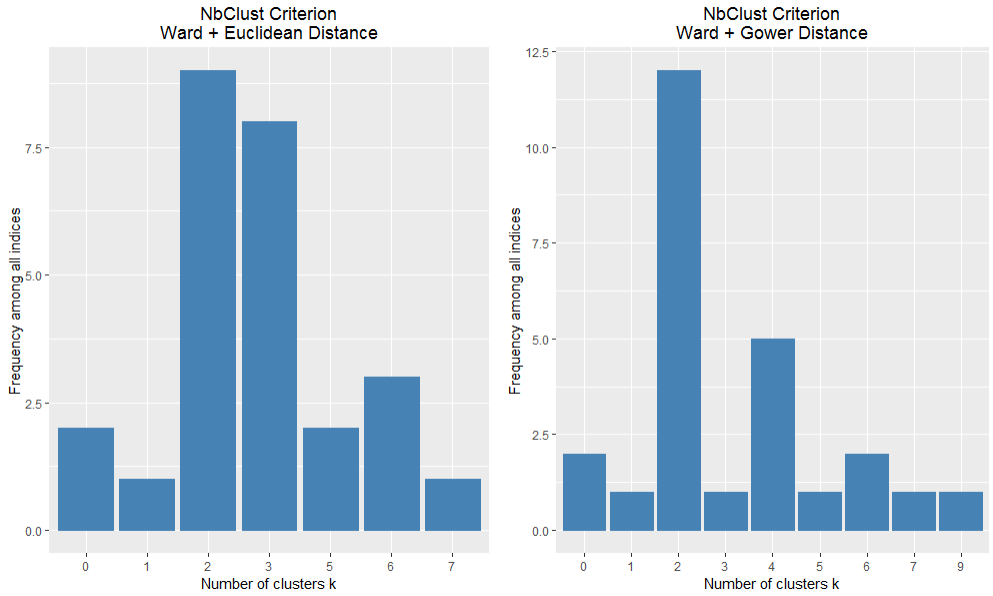


In the above plots, the cluster boundaries are determined by connecting the most extreme data points for each cluster. Interestingly, the medoid cluster technique is shown to have a much greater overlap of clusters than the k-means technique. Clearly, the overlap of cluster points and therefore the inability to distinguish clusters can be seen as a disadvantage for the Medoid cluster technique.

3.5 Hierarchical Clustering

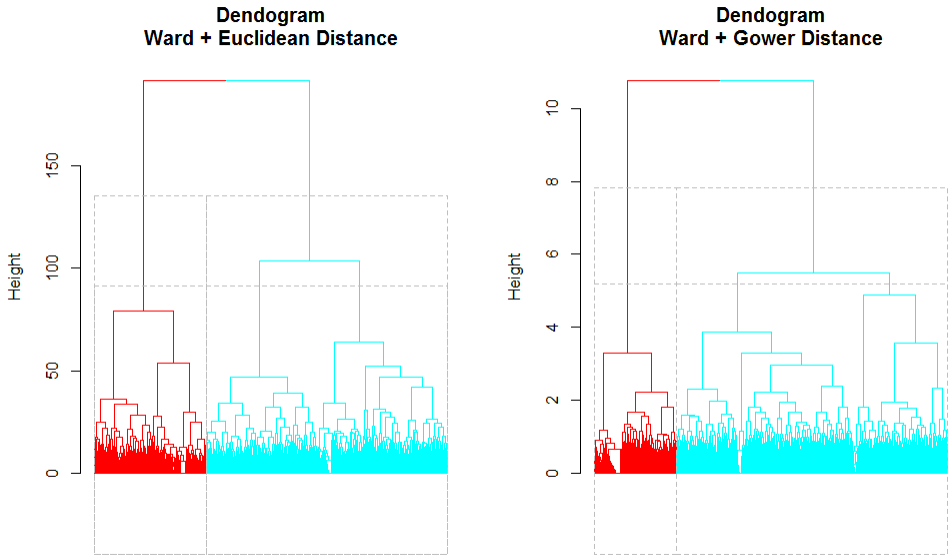
For the hierarchical clustering method, we follow a similar process as above to first determine an appropriate number of clusters. Although we will assess all possible cluster options as part of a review of each dendogram, a pre-determined number of clusters will aid in interpretation of these plots. Again, we apply the 'NbClust' function to the ward clustering technique using both the Euclidean and Gower distance metric.

Figure 3.5.1 Hierarchical Cluster Criterion



We see that the use of three clusters is reported to be optimal when using the Euclidean distance metric. When using the Gower distance metric however, the use of two clusters is reported to be optimal. As mentioned previously, hierarchical clustering methods are also able to provide a dendogram to help assist in determining the number of clusters. Dendograms for both distance metrics are shown below, colored according to the use of their optimal number of clusters as reported by the NbClust criterion.

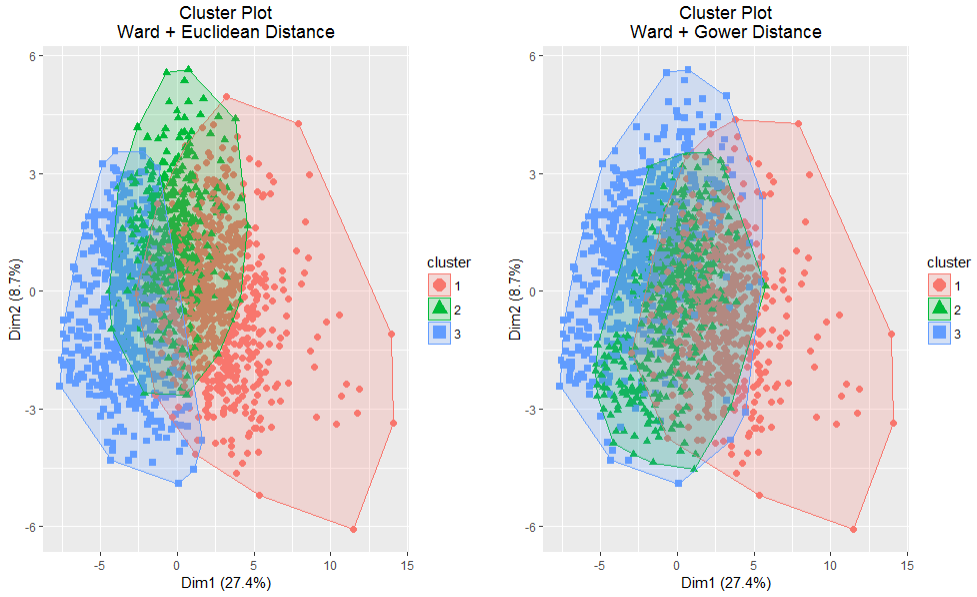
Figure 3.5.2 Hierarchical Cluster Dendogram



Determining the optimum 'cut' using a dendogram is a subjective process. Although there are some rules of thumb available (i.e. clustering from the top-down/bottom-up to include a minimum amount of leaves, or excluding any early made branches with a minimum amount of leaves), the practice is still likely to result in widely varying results between analysts. For the two dendograms shown above, we have made the assessment of three clusters being appropriate under both distance metrics. We base this on an assessment of the amount of data points included within each cluster. That is, making a cut after the second Clade when using either distance metric, results in data points being attributed fairly evenly between three clusters. Note that while the NbClust criterion reported three clusters to be optimum when using the Euclidean distance metric, however, four clusters was reported to be inferior when using the Gower distance metric.

Below we show scatter plots which highlight the hierarchical cluster assignments based on a two-dimensional reduction of our original set of attitudinal survey responses.

Figure 3.5.3 Hierarchical Cluster Plots



Both are shown to have a large amount of overlap between clusters. In particular, assigning three clusters according to the Euclidean distance metric makes it difficult to distinguish the second cluster from the remaining two clusters. And assigning three clusters according to the Gower distance metric makes it difficult to distinguish the first and third cluster.

3.6 Statistical Confirmation

As a final evaluation of the performance of each clustering method, we applied a generalized linear model to each question based on each clustering method, and measured the models Akaike Information Criterion (AIC) for each. We expect that generally, the lower the AIC value, the more superior that clustering method is in predicting each cluster.

An assessment of the AIC value for each question reveals that each clustering method tends to demonstrate low AIC values for questions two and four, as well as high AIC values for questions one, and 56. This is interesting as questions two and four are focused on the purchase behavior of consumer segments, while questions one and 56 are focused on characterizing the demographic features of each consumer segment.

We are also able to observe the average AIC value for each clustering method in order to understand if a particular method is shown to have superior predictive performance. A summary of the results for each is shown in the table below.

Table 3.6.1: AIC Comparison Summary

	1	2	3	4
Method	Non-Hierarchical	Non-Hierarchical	Hierarchical	Hierarchical
Technique	k-means	Medoid	Ward	Ward
Distance Metric	Euclidean	Gower	Eucildean	Gower
Number of Clusters	2	2	3	3
Mean AIC	4025.706	4131.336	4015.816	4063.146

We can see that the Ward clustering technique with Eucildean distance metric reported the lowest average AIC value, with the k-means clustering technique with Eucildean distance metric reporting the second lowest.

3.7 Optimal Clustering Method

Based on an assessment of the four combinations of clustering methods and distance metric, we find the k-means clustering technique with Euclidean distance to be the most favorable. This is largely based on the technique's ability to distinguish between clusters with minimal overlap over our two-dimensional data representations. We believe that this technique's superior ability to distinguish between clusters should ultimately improve our ability to draw inferences between clusters based on their attitudinal characteristics. We do note the potential shortcoming in leveraging a technique with the Euclidean distance metric, namely that we are to assume the original dataset is of continuous type. However, we feel that the encouraging results and simplicity of this technique is well suited for the scope of this analysis.

4 Segment Profiling

We employ visual methods in order to evaluate both the demographic and consumer behavior characteristics of the two clusters determined by our selected clustering method. We do this by generating a set of bar plots of the total response count and proportion of responses over each available option, for each question.

4.1 Demographic Characteristics

We found a number of similarities over demographic characteristics for both clusters. A similar proportion of respondents from both clusters were either currently married, had no children, or were as likely to be male or female. There were also similarities over the total number of respondents for each cluster. We found for both clusters, there were significantly more respondents who were younger than 35, college graduates, white, married, had no children, and had a household annual income of $30,000 to $70,000.

Fortunately, clusters were also able to be distinguished by a number of demographic characteristics. The first cluster for example, was found to contain a greater amount of younger respondents, who were more educated, were more often white or Caucasian, or had higher incomes. To a lessor extent, we also found this cluster to be more likely to have been previously married but now separated, or to have older children than those respondents from cluster two who also had children.

A number of plots related to the demographic characteristics for each cluster can be found in Appendix A.

4.2 Consumer Preferences

We again noted a number of similarities over consumer preferences for both clusters. The majority of respondents owned an iPhone device, used gaming or social networking applications, had in excess of 10 applications on their chosen device, or acquired at least half of their device applications freely.

In terms of the differences in consumer preferences between clusters, we found that respondents from the first cluster had a greater preference for iPhone and Android devices and less of a preference for iPods and tablet devices. In addition, the first cluster had less of a preference for entertainment or T.V. applications, but a greater bias towards gaming, music and social applications. Finally, respondents from the first cluster were

also found to have less applications, more free applications and less likely to visit the websites designated by the survey.

A number of plots related to the consumer preferences for each cluster can be found in Appendix A.

4.3 Initial Recommendation

Results from segment profiling indicate that the consumers within the first cluster tend to have less of a preference for entertainment and T.V. applications whilst more of a preference for social media applications. Since AppHappy aim to produce a social entertainment application, the results suggest that the social media aspect of the product may have greater appeal to the first cluster, whilst the entertainment aspect of the product may have greater appeal to the second cluster.

The results indicated that consumers within the first cluster have a preference for using iPhone and Android devices. If AppHappy were to target a product at consumers designated by the first cluster, that product would likely have greater adoption if it were made available on these devices. The results also indicated that consumers within the second cluster have a preference for using iPod and tablet devices, as well as a greater tendency to browse those websites designated by the survey (e.g. Facebook and Twitter). If AppHappy were to target a product at consumers designated by the second cluster, that product would likely have greater adoption if it were made available on these devices and advertised on websites designated by the survey.

5 Predictive Classifier

AppHappy has requested that the segmentation results be used to inform a classification model, which can ultimately be applied to future survey data in order to generate meaningful customer segmentations. Such classification models are able to derive predictors from observations that are known, and apply those derived predictors to new observations (E. Feit 2015).

There are many classification models which AppHappy could employ to predict customer membership, with each model generally being categorized according to their underlying algorithm. Three possible algorithms include those based on a Decision Tree (DT), Support Vector Machine (SVM), or K-Nearest Neighbors (KNN) methodology. A summary of the strengths and weaknesses of each are shown below.

5.1 Decision Tree

According to scikit learn, Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression (Scikit-Learn 2014). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Strengths:

- Fast training and testing phases.
- Low memory usage.
- Simple to understand and interpret.
- Implicitly performs variable screening or feature selection.

Weaknesses:

- Has the tendency to overfit data.

5.2 Support Vector Machine

According to scikit learn, Support Vector Machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection (Scikit-Learn 2014). SVMs use a linear hyperplane in order to separate data point, but can also be used as a non-linear classified through the use of kernels.

Strengths:

- Low memory usage, as it only needs to store a subset of the data to make predictions.

- SVM is effective in high dimensional spaces.
- Versatile, as the kernel allows expert knowledge of the problem to be built into the classifier.

Weaknesses:

- Slow training and testing phases.

5.3 K-Nearest Neighbors

According to scikit learn, supervised neighbors-based learning methods come in two flavors: classification for data with discrete labels, and regression for data with continuous labels (Scikit-Learn 2014). The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these.

Strengths:

- Fast training phase.
- Versatile, as it does not make any assumption about the underlying data distribution.

Weaknesses:

- Slow testing phase.
- High memory usage, as it requires storing all training data.

5.4 Initial Recommendation

It is difficult to recommend a classification model without directly assessing the performance of each method. However, it is fair to say that if the model is intended to be applied to large datasets where CPU/memory constraints may be of issue, a DT or SVM based classifier may be superior. On the other hand, if AppHappy intends on capturing data which exhibits varying types, scale and distributions, than a KNeighbors based classifier may be superior.

6 Conclusion

For this assessment, we applied a number combinations of clustering techniques and distance metrics in order to segment survey data according to three basis variables. Combinations involved the use of hierarchical and non-hierarchical clustering methods under k-means, medoid and ward techniques, as well as the use of Euclidean and Gower distance metrics. While there were some noted concerns assuming the data to be of continuous type, results showed the k-means clustering technique with Euclidean distance metric to be the most favorable. This was largely due to this methods ability to distinguish between clusters.

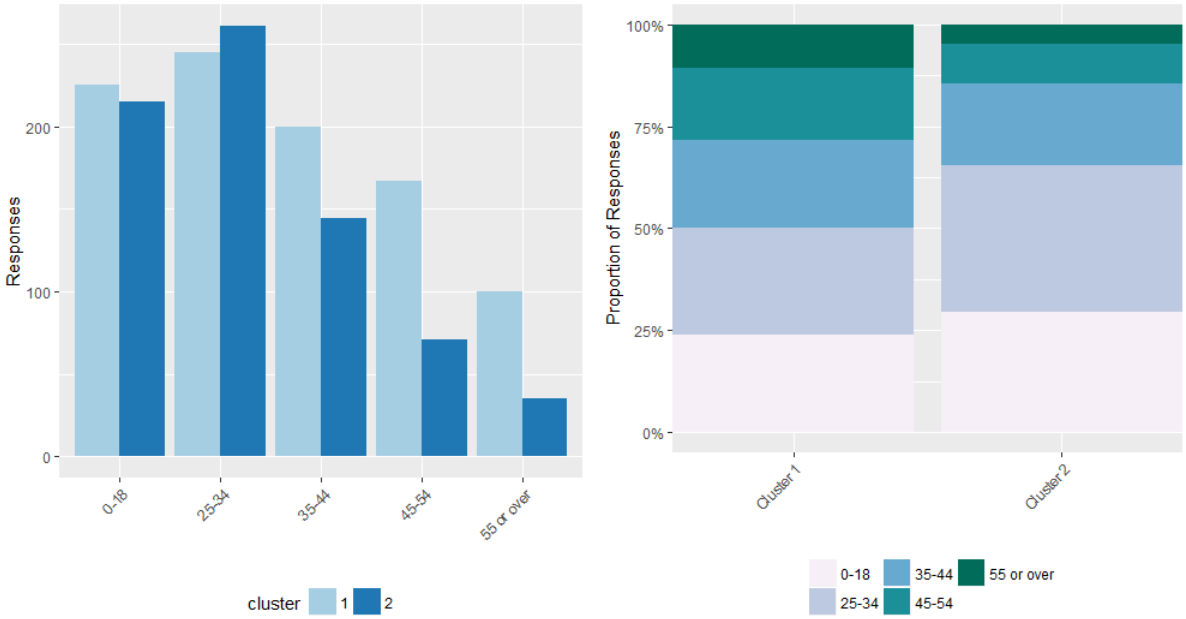
Respondents from both clusters were found to share many demographic and behavioral characteristics, however a number of unique biases were also identified. With respect to demographics, clusters were able to be distinguished according to respondent age, ethnicity, education and incomes. With respect to consumer attitudes, clusters were able to be distinguished according to their preferences for type of electronic device, use of applications and propensity to purchase rather than freely download applications. Most notably, the first cluster was found to have less of a preference for entertainment and T.V. applications, more of a preference for social media applications, a greater preference for using iPhone and Android devices, and less of a tendency to browse those websites designated by the survey (e.g. Facebook and Twitter).

There are many classification models which AppHappy could employ to predict customer membership, with each model generally being categorized according to their underlying algorithm. Three possible algorithms include those based on a Decision Tree (DT), Support Vector Machine (SVM), or K-Nearest Neighbors (KNN) methodology. This assessment presented a summary of advantages and disadvantages associated with each, however we note that a recommendation of classification model would benefit from a direct assessment using the supplied data.

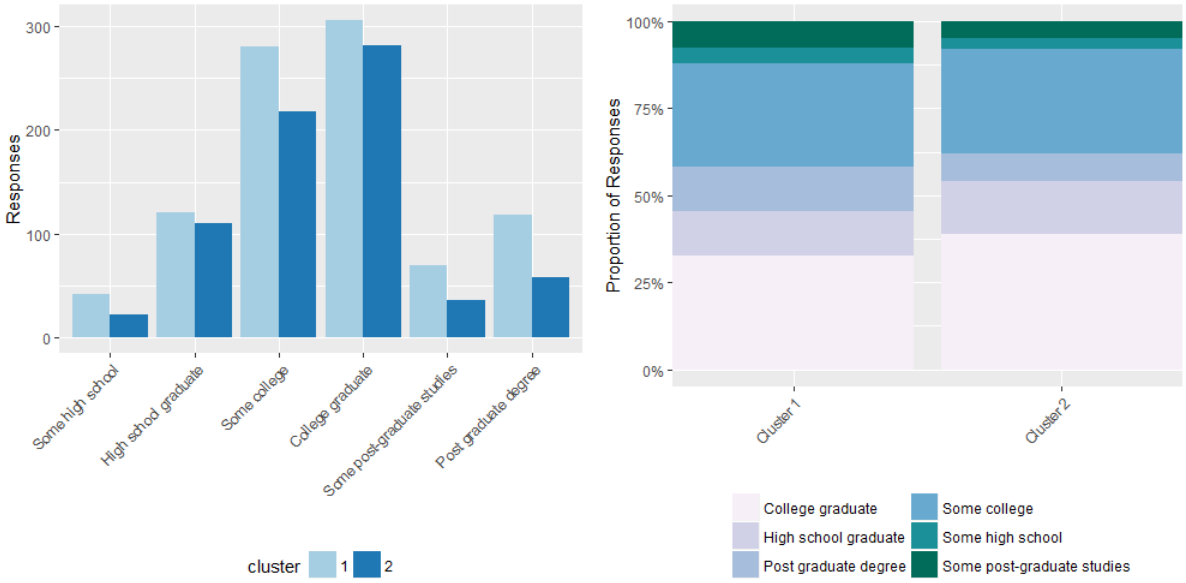
Appendix A Figure Output

Figure Set A1 Demographic Characteristics

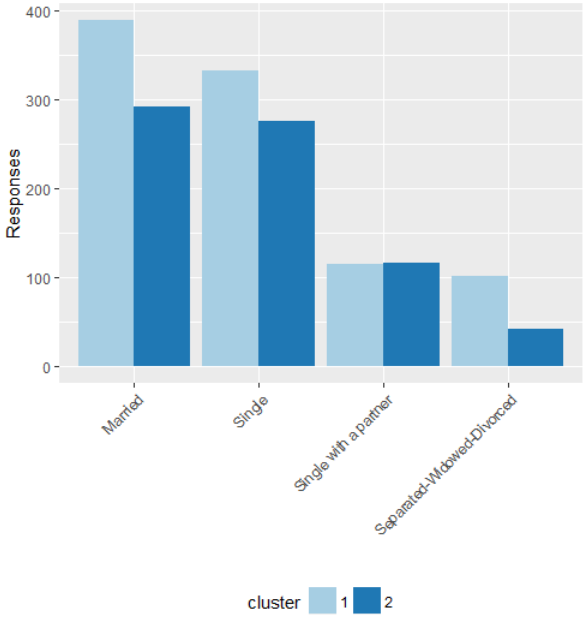
q1. Which of the following best describes your age?

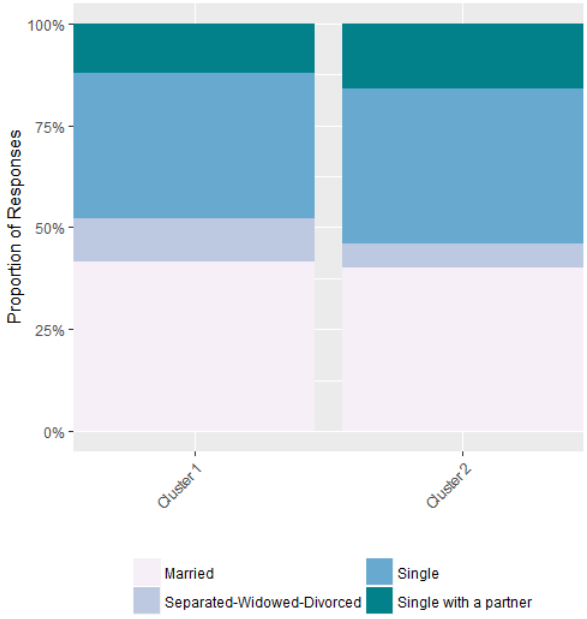


q48. Which of the following best describes the highest level of education you have attained?

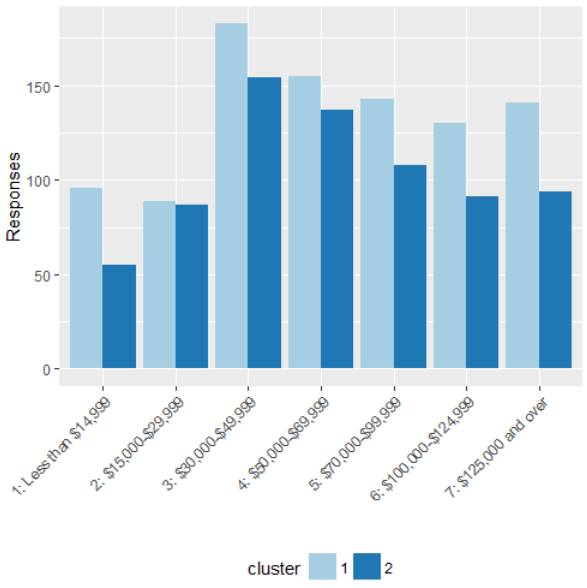


q49. Which of the following best describe your marital status?





q56. Which of the following best describes your household annual income before taxes?



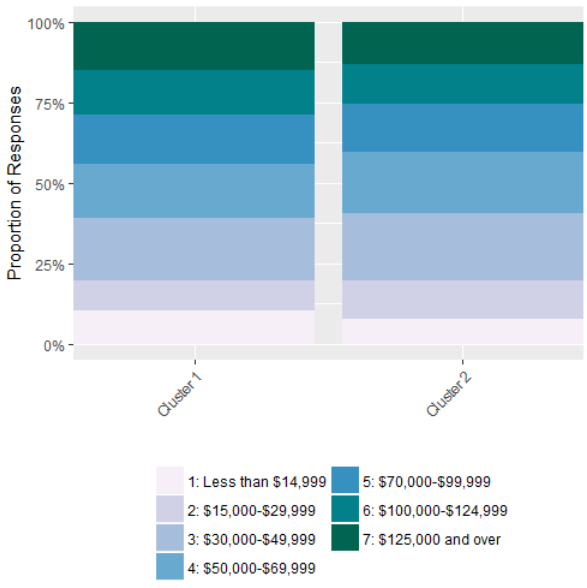
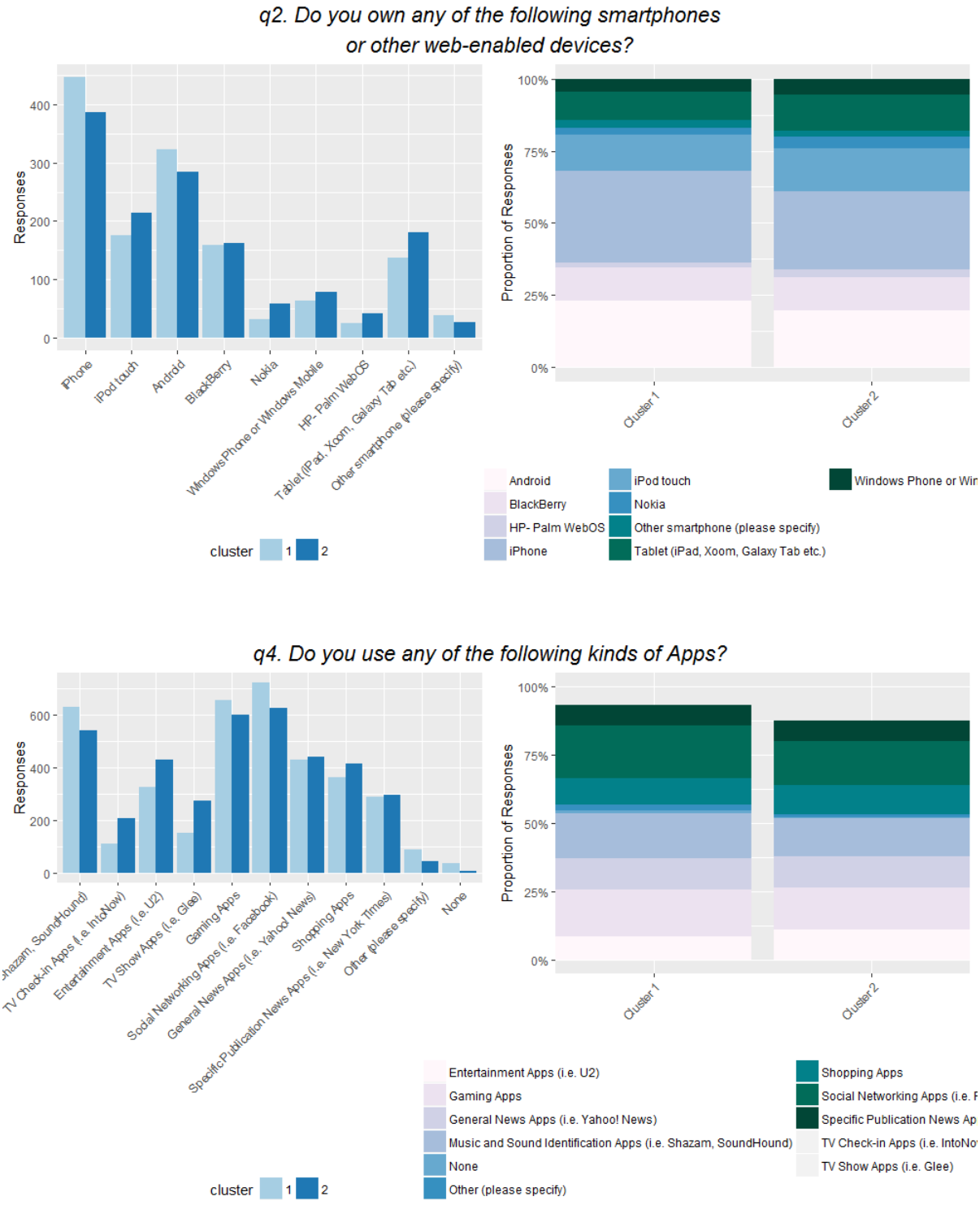
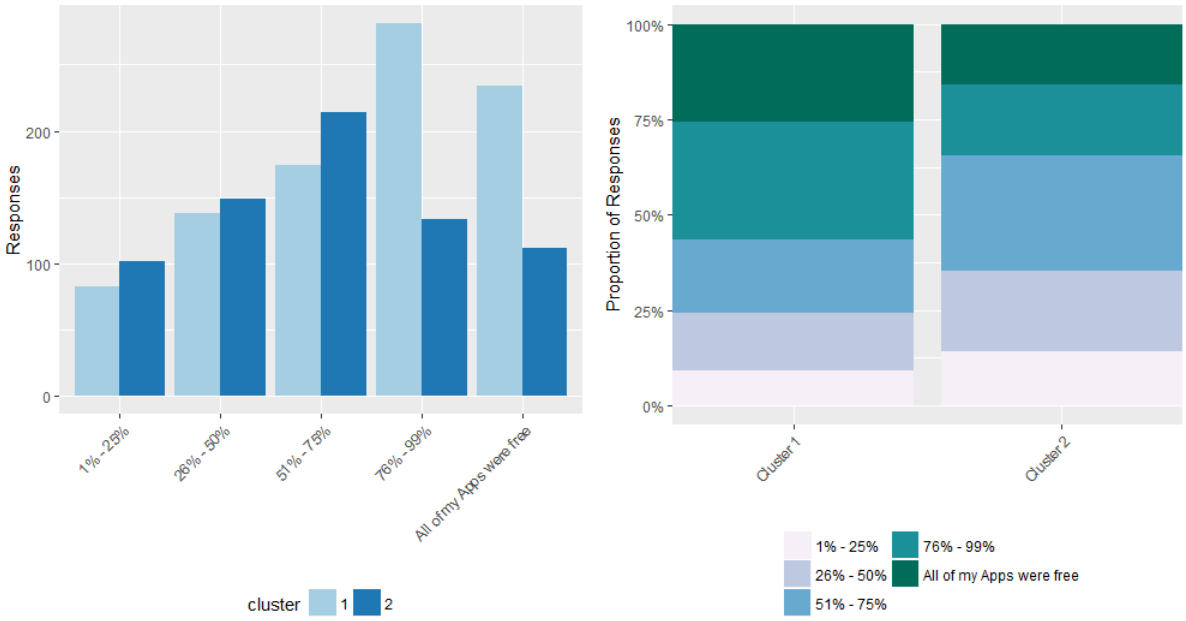


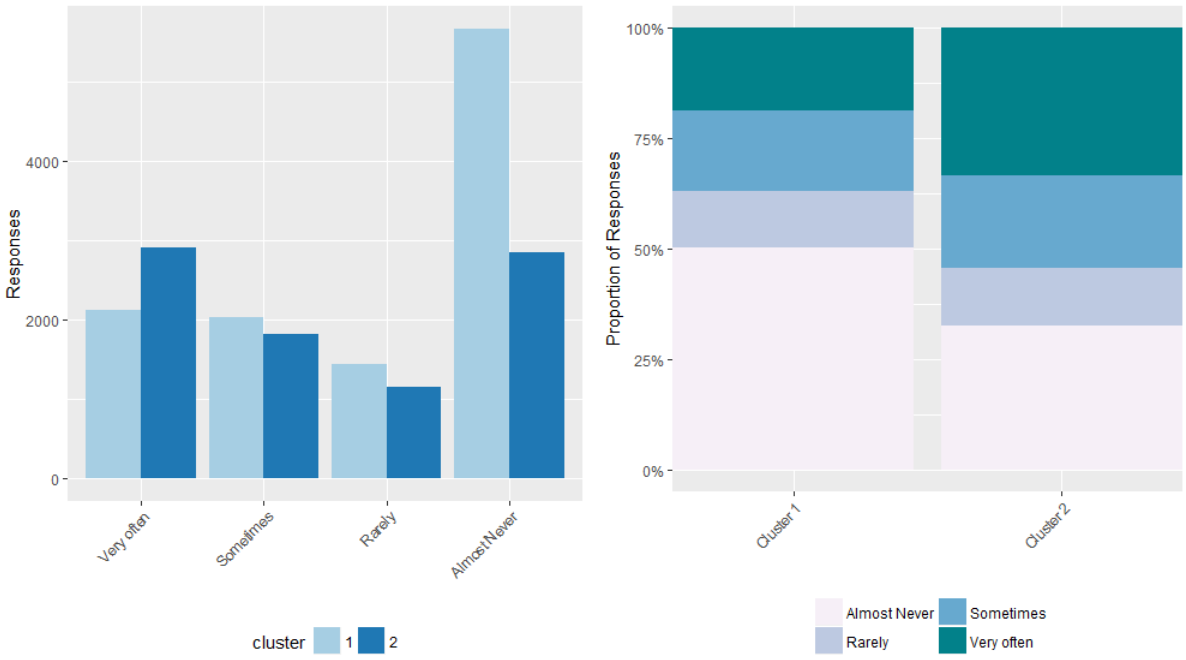
Figure Set A2 Consumer Preferences



q12. Of your Apps, what percent were free to download?



q13. How many times per week do you visit each of the following websites?



Appendix B Code

B1 Data Prep

```
for(package in c('cluster', 'factoextra', 'fpc', 'NbClust',
                 'reshape', 'plyr',
                 'ggplot2', 'scales', 'grid', 'gridExtra')) {
  if(!require(package, character.only=TRUE)) {
    install.packages(package)
    library(package, character.only=TRUE)
  }
}

rm(package)

# Load the dataset
load('data/appHappyData-sp2016.RData')
df_appnum.raw <- apphappy.4.num.frame
df_applab.raw <- apphappy.4.labs.frame

# Subset the data for attitudinal variables
colstart <- which(colnames(df_appnum.raw) == 'q24r1')
colend <- which(colnames(df_appnum.raw) == 'q26r17')
df_appnum.att = df_appnum.raw[,c(colstart:colend)]
df_applab.att = df_applab.raw[,c(colstart:colend)]

# Subset the data for nonattitudinal variables
df_appnum.nonatt = df_appnum.raw[,c(-colstart:-colend)]
df_applab.nonatt = df_applab.raw[,c(-colstart:-colend)]

# Count NA rows for attitudinal variables
inclna <- nrow(df_appnum.att)
exclna <- nrow(na.omit(df_appnum.att))
#print(paste0("Basis observations: ", inclna - exclna)) #0

# Count NA rows for non-attitudinal variables
inclna <- nrow(df_appnum.nonatt)
exclna <- nrow(na.omit(df_appnum.nonatt))
#print(paste0("Non-basis observations: ", inclna - exclna)) #585

# Count NA rows for original dataset
inclna <- nrow(df_appnum.raw)
exclna <- nrow(na.omit(df_appnum.raw))
#print(paste0("Dataset observations: ", inclna - exclna)) #585

rm(apphappy.4.num.frame, apphappy.4.labs.frame, colstart, colend)
```

B2 Non-hierarchical Clustering

```
# Method: kmeans, Distance: euclidean
nbc.eu <- NbClust(df_appnum.att,
                  min.nc=2, max.nc=10, distance='euclidean',
```

```
                  method='kmeans', index='all')
table(nc$Best.n[1,])

plot1 <- fviz_nbclust(nbc.eu) +
  ggtitle('NbClust Criterion\nK-means + Euclidean Distance')

rm(nbc.eu)


# Method: medoid, Distance: gower
dist.daisy <- daisy(df_applab.att, metric='gower', stand=FALSE)
pamk.gow <- pamk(dist.daisy, diss=TRUE, krange=1:10,
                 criterion='asw', critout=TRUE)

df_pamk <- melt(pamk.gow$crit)
df_pamk$index <- rownames(df_pamk)

plot3 <- ggplot(df_pamk, aes(x=index, y=value)) +
  geom_bar(stat='identity', fill='steelblue') +
  scale_x_discrete(limits=c(1:10)) +
  ggtitle('Silhouette Width Criterion\nMedoid + Gower Distance') +
  labs(x='Number of clusters k', y='Average silhouette width criterion value')

rm(dist.daisy, pamk.gow, df_pamk)


png(filename='images/nonhier_kcrit.png',
    width = 1000, height = 600, res = 100)

grid.arrange(plot1, plot3, ncol=2)

dev.off()

rm(plot1, plot2, plot3)

# Method: kmeans, Distance: euclidean
k = 2 # Set cluster center count according to no. clust criterion

set.seed(1)
res.km <- kmeans(df_appnum.att, centers=k)

plot1 <- fviz_cluster(res.km, df_appnum.att,
                      show.clust.cent=TRUE, geom='point',
                      title='Cluster Plot\nK-means + Euclidean Distance')


# Method: medoid, Distance: gower
k = 2 # Set cluster center count according to no. clust criterion

set.seed(1)
dist.daisy <- daisy(df_applab.att, metric='gower', stand=FALSE)
res.pam2 <- pam(dist.daisy, k=k, diss=TRUE,
                metric='euclidean',
```

```
                stand=FALSE, cluster.only=FALSE)

plot3 <- fviz_cluster(list(data=df_appnum.att,
                           cluster=res.pam2$cluster),
                      show.clust.cent=TRUE, geom='point',
                      title='Cluster Plot\nMedoid + Gower Distance')

rm(dist.daisy)


png(filename='images/nonhier_clust.png',
    width = 1000, height = 600, res = 100)

grid.arrange(plot1, plot3, ncol=2)

dev.off()

rm(plot1, plot2, plot3)
rm(k)
```

B3 Hierarchical Clustering

```
# Method: ward.D, Distance: euclidean
nbc.eu <- NbClust(df_appnum.att, diss=NULL,
                  min.nc=2, max.nc=10, distance='euclidean',
                  method='ward.D2', index='all')
table(nbc.eu$Best.n[1,])

plot1 <- fviz_nbclust(nbc.eu) +
  ggtitle('NbClust Criterion\nWard + Euclidean Distance')


# Method: ward.D, Distance: manhattan
nbc.man <- NbClust(df_appnum.att, diss=NULL,
                   min.nc=2, max.nc=10, distance='manhattan',
                   method='ward.D2', index='all')
table(nbc.man$Best.n[1,])

plot2 <- fviz_nbclust(nbc.eu) +
  ggtitle('NbClust Criterion\nWard + Manhattan Distance')


# Method: ward.D, Distance: gower
dist.daisy <- daisy(df_applab.att, metric='gower', stand=FALSE)
nbc.gow <- NbClust(df_appnum.att, diss=dist.daisy,
                   min.nc=2, max.nc=10, distance=NULL,
                   method='ward.D2', index='all')
table(nbc.gow$Best.n[1,])

plot3 <- fviz_nbclust(nbc.gow) +
  ggtitle('NbClust Criterion\nWard + Gower Distance')
```

```
png(filename='images/hier_kcrit.png',
    width = 1000, height = 600, res = 100)

grid.arrange(plot1, plot3, ncol=2)

dev.off()

rm(plot1, plot2, plot3)

# Method: ward.D, Distance: euclidean
k = 3 # Color dendogram according to no. clust criterion

set.seed(1)
res.hcut1 <- hcut(df_appnum.att, k=k, isdiss=FALSE,
                  hc_func='hclust', hc_metric='euclidean',
                  hc_method='ward.D2', stand=FALSE)

plot1a <- fviz_silhouette(res.hcut1) +
  scale_y_continuous(limits = c(-0.25, 0.5)) +
  ggtitle('Silhouette Plot\nWard + Euclidean Distance') +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot1b <- fviz_cluster(res.hcut1, df_appnum.att,
                       show.clust.cent=TRUE, geom='point',
                       title='Cluster Plot\nWard + Euclidean Distance')


# Method: ward.D, Distance: gower
k = 2 # Color dendogram according to no. clust criterion

set.seed(1)
dist.daisy <- daisy(df_applab.att, metric='gower', stand=FALSE)
res.hcut3 <- hcut(dist.daisy, k=k, isdiss=TRUE,
                  hc_func='hclust', hc_metric='euclidean',
                  hc_method='ward.D2', stand=FALSE)

plot3a <- fviz_silhouette(res.hcut3) +
  scale_y_continuous(limits = c(-0.25, 0.5)) +
  ggtitle('Silhouette Plot\nWard + Gower Distance') +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot3b <- fviz_cluster(res.hcut3, df_appnum.att,
                       show.clust.cent=TRUE, geom='point',
                       title='Cluster Plot\nWard + Gower Distance')

rm(dist.daisy)


png(filename='images/hier_den.png',
    width = 1000, height = 600, res = 100)

par(mfrow=c(1,2))
```

```
fviz_dend(res.hcut1, k=3,
          cex=0.5, show_labels=FALSE, rect=TRUE,
          main='Dendogram\nWard + Euclidean Distance')

fviz_dend(res.hcut3, k=2,
          cex=0.5, show_labels=FALSE, rect=TRUE,
          main='Dendogram\nWard + Gower Distance')

par()
dev.off()


# Method: ward.D, Distance: gower
k = 3 # Re-run for four clusters based on interpretation of dendogram

set.seed(1)
dist.daisy <- daisy(df_applab.att, metric='gower', stand=FALSE)
res.hcut3 <- hcut(dist.daisy, k=k, isdiss=TRUE,
                  hc_func='hclust', hc_metric='euclidean',
                  hc_method='ward.D2', stand=FALSE)

plot3a <- fviz_silhouette(res.hcut3) +
  scale_y_continuous(limits = c(-0.25, 0.5)) +
  ggtitle('Silhouette Plot\nWard + Gower Distance') +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot3b <- fviz_cluster(res.hcut3, df_appnum.att,
                       show.clust.cent=TRUE, geom='point',
                       title='Cluster Plot\nWard + Gower Distance')

rm(dist.daisy)


png(filename='images/hier_sil.png',
    width = 1000, height = 600, res = 100)

grid.arrange(plot1a, plot3a, ncol=2)

dev.off()

rm(plot1a, plot2a, plot3a)

png(filename='images/hier_clust.png',
    width = 1000, height = 600, res = 100)

grid.arrange(plot1b, plot3b, ncol=2)

dev.off()

rm(plot1b, plot2b, plot3b)
rm(k)
```

B4 Statistical Confirmation

```
ls_nm <- list('res.km', 'res.pam2', 'res.hcut1', 'res.hcut3')
ls_df <- list(res.km, res.pam2, res.hcut1, res.hcut3)

# Calculate the AIC for each clustering method, for each question
apply.glm.f <- function(y,class){
  return(glm(y~class)$aic)
}

df_temp <- data.frame(names = rownames(t(df_appnum.raw)))

for (i in 1:length(ls_nm)){
  df_appnum.raw$cluster <- ls_df[[i]]$cluster
  df_aic <- data.frame(apply(df_appnum.raw, 2,
                             apply.glm.f, class=df_appnum.raw$cluster))
  names(df_aic)[1] <- ls_nm[[i]]
  df_aic$names <- rownames(df_aic)

  df_temp <- merge(x=df_temp, y=df_aic, by.x='names', all=TRUE)
}

rownames(df_temp) <- df_temp[, 1]
df_temp <- df_temp[-c(1:2),-1]

df_aic <- df_temp[is.finite(rowSums(df_temp)), ]
colnames(df_aic) <- c('k-means + Euclidean', 'Medoid + Gower',
                      'Ward + Eucildean', 'Ward + Gower')
write.table(round(df_aic, 3), 'aic.csv')
apply(df_aic, 2, mean)


# Calculate the Chi for each clustering method, for each question
sim.chisq.pval.f <- function(x,class){
  return(chisq.test(x,class,
                    rescale.p=TRUE,
                    simulate.p.value=TRUE)$p.value)
}

df_temp <- data.frame(names = rownames(t(df_appnum.raw)))

for (i in 1:length(ls_nm)){
  df_appnum.raw$cluster <- ls_df[[i]]$cluster
  df_chi <- data.frame(apply(df_appnum.att, 2,
                             sim.chisq.pval.f, class=df_appnum.raw$cluster))
  names(df_chi)[1] <- ls_nm[[i]]
  df_chi$names <- rownames(df_chi)

  df_temp <- merge(x=df_temp, y=df_chi, by.x='names', all=TRUE)
}

rownames(df_temp) <- df_temp[, 1]
df_temp <- df_temp[-c(1:2),-1]
```

```
df_chi <- df_temp[is.finite(rowSums(df_temp)), ]
colnames(df_chi) <- c('k-means + Euclidean', 'Medoid + Gower',
                      'Ward + Eucildean', 'Ward + Gower')
apply(df_chi, 2, mean)


rm(ls_nm, ls_df, df_temp)
```

B5 Cluster Exploration

```
# Select clustering method
res.sel <- res.km # res.km, res.pam1, res.pam2, res.hcut1, res.hcut2, res.hcut3
df_appnum.raw$cluster <- res.sel$cluster
df_applab.raw$cluster <- res.sel$cluster

df_means <- ddply(df_appnum.raw,
                  .(df_appnum.raw$cluster), colwise(mean))

# Basic Cluster Profiling
with(df_applab.raw, table(q1, cluster))
with(df_applab.raw, table(q11, cluster))
with(df_applab.raw, table(q12, cluster))
with(df_applab.raw, table(q48, cluster))
with(df_applab.raw, table(q49, cluster))
with(df_applab.raw, table(q54, cluster))
with(df_applab.raw, table(q56, cluster))

plot(df_means$q1, type='b', main='Age by Cluster')
plot(df_means$q11, type='b', main='# of Apps by Cluster')
plot(df_means$q12, type='b', main='% of Free Apps by Cluster')
plot(df_means$q48, type='b', main='Education by Cluster')
plot(df_means$q49, type='b', main='Martial Status by Cluster')
plot(df_means$q54, type='b', main='Ethnicity by Cluster')
plot(df_means$q56, type='b', main='Income by Cluster')

anova(aov(q1~cluster, data=df_appnum.raw))
anova(aov(q11~cluster, data=df_appnum.raw))
anova(aov(q12~cluster, data=df_appnum.raw))
anova(aov(q48~cluster, data=df_appnum.raw))
anova(aov(q49~cluster, data=df_appnum.raw))
anova(aov(q54~cluster, data=df_appnum.raw))
anova(aov(q56~cluster, data=df_appnum.raw))


# Recategorize high level fields
df_applab.raw$q1.recat <- as.character(df_applab.raw$q1)
df_applab.raw$q1.recat[df_applab.raw$q1.recat == 'Under 18' |
                       df_applab.raw$q1.recat == '18-24'] <- '0-18'
df_applab.raw$q1.recat[df_applab.raw$q1.recat == '25-29' |
                       df_applab.raw$q1.recat == '30-34'] <- '25-34'
df_applab.raw$q1.recat[df_applab.raw$q1.recat == '35-39' |
                       df_applab.raw$q1.recat == '40-44'] <- '35-44'
```

```
df_applab.raw$q1.recat[df_applab.raw$q1.recat == '45-49' |
                       df_applab.raw$q1.recat == '50-54'] <- '45-54'
df_applab.raw$q1.recat[df_applab.raw$q1.recat == '55-59' |
                       df_applab.raw$q1.recat == '60-64' |
                       df_applab.raw$q1.recat == '65 or over'] <- '55 or over'
df_applab.raw$q1.recat <- as.factor(df_applab.raw$q1.recat)

df_applab.raw$q11.recat <- df_applab.raw$q11
df_applab.raw$q11.recat <- as.character(df_applab.raw$q11)
df_applab.raw$q11.recat[grepl("Don", df_applab.raw$q11.recat)] <- NA
df_applab.raw$q11.recat[df_applab.raw$q11.recat == 'None' |
                        df_applab.raw$q11.recat == '1-5'] <- '1: 0-5'
df_applab.raw$q11.recat[df_applab.raw$q11.recat == '6-10'] <- '2: 6-10'
df_applab.raw$q11.recat[df_applab.raw$q11.recat == '11-30'] <- '3: 11-30'
df_applab.raw$q11.recat[df_applab.raw$q11.recat == '31+'] <- '4: 31+'
df_applab.raw$q11.recat <- as.factor(df_applab.raw$q11.recat)

df_applab.raw$q12.recat <- df_applab.raw$q12
df_applab.raw$q12.recat[df_applab.raw$q12.recat == 'None of my Apps were free'] <- NA

df_applab.raw$q56.recat <- as.character(df_applab.raw$q56)
df_applab.raw$q56.recat[df_applab.raw$q56.recat == 'Under $10,000' |
                        df_applab.raw$q56.recat == '$10,000-$14,999'] <- '1: Less than $14,999'
df_applab.raw$q56.recat[df_applab.raw$q56.recat == '$15,000-$19,999' |
                        df_applab.raw$q56.recat == '$20,000-$29,999'] <- '2: $15,000-$29,999'
df_applab.raw$q56.recat[df_applab.raw$q56.recat == '$30,000-$39,999' |
                        df_applab.raw$q56.recat == '$40,000-$49,999'] <- '3: $30,000-$49,999'
df_applab.raw$q56.recat[df_applab.raw$q56.recat == '$50,000-$59,999' |
                        df_applab.raw$q56.recat == '$60,000-$69,999'] <- '4: $50,000-$69,999'
df_applab.raw$q56.recat[df_applab.raw$q56.recat == '$70,000-$79,999' |
                        df_applab.raw$q56.recat == '$80,000-$89,999'] <- '5: $70,000-$99,999'
df_applab.raw$q56.recat[df_applab.raw$q56.recat == '$90,000-$99,999' |
                        df_applab.raw$q56.recat == '$100,000-$124,999'] <- '6: $100,000-$124,999'
df_applab.raw$q56.recat[df_applab.raw$q56.recat == '$125,000-$149,999' |
                        df_applab.raw$q56.recat == '$150,000 and over'] <- '7: $125,000 and over'
df_applab.raw$q56.recat <- as.factor(df_applab.raw$q56.recat)

#lapply(df_applab.raw, class)


# Loop through plots
ls_aes.q2 <- list('q2r1', 'q2r2', 'q2r3', 'q2r4', 'q2r5',
                  'q2r6', 'q2r7', 'q2r8', 'q2r9', 'q2r10')
ls_sub.q4 <- list('q4r1', 'q4r2', 'q4r3', 'q4r4', 'q4r5',
                  'q4r6', 'q4r7', 'q4r8', 'q4r9', 'q4r10', 'q4r11')
ls_sub.q13 <- list('q13r1', 'q13r2', 'q13r3', 'q13r4', 'q13r5', 'q13r6',
                   'q13r7', 'q13r8', 'q13r9', 'q13r10', 'q13r11', 'q13r12', 'q13r13')
ls_sub.q24 <- list('q24r1', 'q24r2', 'q24r3', 'q24r4', 'q24r5', 'q24r6',
                   'q24r7', 'q24r8', 'q24r9', 'q24r10', 'q24r11', 'q24r12')
ls_sub.q25 <- list('q25r1', 'q25r2', 'q25r3', 'q25r4', 'q25r5', 'q25r6',
                   'q25r7', 'q25r8', 'q25r9', 'q25r10', 'q25r11', 'q25r12')
ls_sub.q26 <- list('q26r1', 'q26r2', 'q26r3', 'q26r4', 'q26r5', 'q26r6', 'q26r7', 'q26r8',
                   'q26r9', 'q26r10', 'q26r11', 'q26r12', 'q26r13', 'q26r14', 'q26r15', 'q26r16', 'q26r
```

```
ls_sub.q50 <- list('q50r1', 'q50r2', 'q50r3', 'q50r4', 'q50r5')

ls_aes <- list('q1.recat', 'q11.recat', 'q12.recat', 'q48', 'q49', 'q54', 'q55', 'q56.recat', 'q57',
               ls_aes.q2, ls_sub.q4, ls_sub.q13, ls_sub.q24, ls_sub.q25, ls_sub.q26, ls_sub.q50)

ls_title <- c('q1. Which of the following best describes your age?\n',
              'q11. How many Apps do you have on your smartphone/iPod Touch/Tablet?\n',
              'q12. Of your Apps, what percent were free to download?\n',
              'q48. Which of the following best describes the\nhighest level of education you have atta
              'q49. Which of the following best describe your marital status?\n',
              'q54. Which of the following best describes your race?\n',
              'q55. Do you consider yourself to be of Hispanic or Latino ethnicity?\n',
              'q56. Which of the following best describes your\nhousehold annual income before taxes?\n
              'q57. Please indicate your gender',
              'q2. Do you own any of the following smartphones\nor other web-enabled devices?',
              'q4. Do you use any of the following kinds of Apps?',
              'q13. How many times per week do you visit each of the following websites?',
              'q24. Please tell us how much you agree or disagree\nwith each of the following statement
              'q25. And how much do you agree or disagree with each of the following?',
              'q26. And finally how much do you agree or disagree\nwith each of these statements?',
              'q50. Do you currently have any children in the following age groups?')

for (i in 1:length(ls_aes)){

  df_temp <- df_applab.raw[ , names(df_applab.raw) %in% c(ls_aes[[i]], 'cluster')]
  df_temp <- melt(df_temp, id.vars=c('cluster'))
  df_temp <- df_temp[!grepl("NO TO", df_temp$value), ]

  df_temp[, 'cluster'] <- as.factor(df_temp[, 'cluster'])
  df_temp[, 'value'] <- as.factor(df_temp[, 'value'])

  plot1 <- ggplot(na.omit(df_temp), aes_string(x='value', fill='cluster')) +
    geom_bar(position='dodge') +
    scale_fill_brewer(palette='Paired') +
    #scale_y_continuous(limits = c(0, 250)) +
    #ggtitle(ls_title[i]) +
    labs(x='', y='Responses') +
    #guides(fill=guide_legend(reverse=TRUE)) +
    theme(axis.text.x=element_text(angle=45, hjust=1),
          legend.position='bottom')

  df_temp[, 'cluster'] <- as.numeric(df_temp[, 'cluster'])
  df_temp[, 'value'] <- as.character(df_temp[, 'value'])

  legrow <- round(length(unique(df_temp[, 'value'])) / 2, 0)

  plot2 <- ggplot(na.omit(df_temp), aes_string(x='cluster', fill='value')) +
    geom_bar(position='fill') +
    scale_fill_brewer(palette='PuBuGn') +
    scale_y_continuous(labels=percent) +
    scale_x_discrete(limits = c('Cluster 1', 'Cluster 2')) +
    #ggtitle(ls_title[i]) +
    labs(x='', y='Proportion of Responses') +
```

```
    guides(fill=guide_legend(nrow=legrow)) +
    theme(axis.text.x=element_text(angle=45, hjust=1),
          legend.title=element_blank(), legend.position='bottom')

  filenm <- paste0('images/barplot_', ls_aes[[i]][[1]], '.png')

  png(filename=filenm, width = 1000, height = 600, res = 100)

  grid.arrange(plot1, plot2, ncol=2,
               top=textGrob(ls_title[i],
                            gp=gpar(fontsize=16,font=3)))

  dev.off()

  print(paste0('saved: ', filenm))

}

rm(ls_aes.q2, ls_sub.q4, ls_sub.q13, ls_sub.q24, ls_sub.q25, ls_sub.q26, ls_sub.q50, ls_aes, ls_title)
rm(i, df_temp, legrow, filenm, plot1, plot2)
```

Reference

A. Struyf, M Hubert & P. Rousseeuw. 2016. “Dissimilarity Matrix Calculation.” https://stat.ethz.ch/R-manual/R-devel/library/cluster/html/daisy.html.

Cornish, R. 2007. “Statistics - Cluster Analysis.” http://www.statstutor.ac.uk/resources/uploaded/clusteranalysis.pdf.

E. Feit, C. Chapman &. 2015. “R for Marketing Research and Analytics.” Springer International Publishing.

F. Mundt, A. Kassambara &. 2016. “Extract and Visualize the Results of Multivariate Data Analyses.” https://cran.r-project.org/web/packages/factoextra/index.html.

Maechler, M. 2014. “Agglomerative Nesting (Hierarchical Clustering).” http://stat.ethz.ch/R-manual/R-patched/library/cluster/html/agnes.html.

P. Rousseeuw, L. Kaufman &. 1990. “Finding Groups in Data: An Introduction to Cluster Analysis.” Wiley.

P. Tan, M. Steinbach & V. Kumar. 2006. “Introduction to Data Mining.” Addison-Wesley.

Scikit-Learn. 2014. “Machine Learning in Python.” http://scikit-learn.org/.

STHDA. 2016. “Determining the Optimal Number of Cluster - 3 Must Known Methods - Unsupervised Machine Learning.” http://www.sthda.com/english/wiki/determining-the-optimal-number-of-clusters-3-must-known-methods-unsuperv