MGTA601 Assignment 2 Appendix

Sarah Mansoor

2022-12-13

Contents

Appendix A: Data

Load Data

```
setwd("~/Desktop/wlu/Fall 2022 MMA/MGTA 601/a2")
FordDem <- read.csv("FordKa_demographic.csv")
FordPsy <- read.csv("FordKa_psychographic.csv")
FordKa <- readxl::read_xls("FordKa.xls")
```

Appendix A.1: Merge Data

```
dta_all <- merge(FordDem, FordKa, by = c("respondent","preference", "Gender",
                                         "Age", "MaritalStatus", "Children",
                                         "FirstPurchase", "AgeCat", "ChildCat",
                                         "IncomeCat"))
names(FordPsy)[1] <- "respondent"
dta_all <- merge(dta_all, FordPsy, by = "respondent")
```

Missing values

```
summary(dta_all)
```

```
##    respondent       preference        Gender          Age
##  Min.   :  1.00   Min.   :1.000   Min.   :1.00   Min.   :20.00
##  1st Qu.: 63.25   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:29.00
##  Median :125.50   Median :2.000   Median :1.00   Median :36.00
##  Mean   :125.50   Mean   :1.784   Mean   :1.48   Mean   :36.36
##  3rd Qu.:187.75   3rd Qu.:2.000   3rd Qu.:2.00   3rd Qu.:43.00
##  Max.   :250.00   Max.   :3.000   Max.   :2.00   Max.   :58.00
##  MaritalStatus      Children     FirstPurchase       AgeCat
##  Min.   :1.000   Min.   :0.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:0.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :1.000   Median :0.000   Median :2.000   Median :4.000
##  Mean   :1.872   Mean   :0.728   Mean   :1.852   Mean   :3.768
##  3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:5.000
##  Max.   :3.000   Max.   :4.000   Max.   :2.000   Max.   :6.000
##     ChildCat       IncomeCat          Q1             Q2             Q3
##  Min.   :0.000   Min.   :1.00   Min.   :1.0   Min.   :1.00   Min.   :1.000
##  1st Qu.:0.000   1st Qu.:2.00   1st Qu.:4.0   1st Qu.:2.00   1st Qu.:4.000
##  Median :0.000   Median :4.00   Median :5.0   Median :4.00   Median :4.000
##  Mean   :0.624   Mean   :3.68   Mean   :5.1   Mean   :4.06   Mean   :4.444
##  3rd Qu.:1.000   3rd Qu.:5.00   3rd Qu.:6.0   3rd Qu.:6.00   3rd Qu.:5.000
##  Max.   :2.000   Max.   :6.00   Max.   :7.0   Max.   :7.00   Max.   :7.000
##        Q4              Q5              Q6              Q7             Q8
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :2.00   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:3.00   1st Qu.:3.000
##  Median :4.000   Median :4.000   Median :4.000   Median :4.00   Median :4.000
##  Mean   :4.236   Mean   :3.848   Mean   :3.992   Mean   :3.88   Mean   :3.916
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.00   3rd Qu.:5.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :6.00   Max.   :7.000
##        Q9             Q10             Q11             Q12
##  Min.   :1.000   Min.   :1.000   Min.   :2.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:3.000
##  Median :4.000   Median :4.000   Median :4.000   Median :4.000
##  Mean   :3.904   Mean   :3.916   Mean   :3.984   Mean   :4.072
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##       Q13             Q14             Q15             Q16
##  Min.   :1.000   Min.   :1.000   Min.   :2.000   Min.   :1.000
```

```
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:4.000   1st Qu.:3.000
##  Median :4.000   Median :5.000   Median :5.000   Median :5.000
##  Mean   :3.988   Mean   :4.132   Mean   :4.972   Mean   :4.512
##  3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.000
##  Max.   :6.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##       Q17             Q18             Q19             Q20
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:2.000
##  Median :5.000   Median :5.000   Median :5.000   Median :4.000
##  Mean   :4.444   Mean   :4.532   Mean   :4.688   Mean   :3.832
##  3rd Qu.:6.000   3rd Qu.:5.750   3rd Qu.:6.000   3rd Qu.:5.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##       Q21             Q22             Q23             Q24             Q25
##  Min.   :2.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:4.000   1st Qu.:3.00   1st Qu.:1.000   1st Qu.:2.000
##  Median :5.000   Median :5.000   Median :4.00   Median :2.000   Median :3.000
##  Mean   :4.912   Mean   :4.992   Mean   :4.12   Mean   :2.376   Mean   :3.148
##  3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.00   3rd Qu.:3.000   3rd Qu.:4.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.00   Max.   :6.000   Max.   :7.000
##       Q26             Q27             Q28             Q29             Q30
##  Min.   :1.000   Min.   :1.00   Min.   :1.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.00   1st Qu.:2.00   1st Qu.:3.000   1st Qu.:2.000
##  Median :3.000   Median :4.00   Median :3.00   Median :3.000   Median :3.000
##  Mean   :3.012   Mean   :3.46   Mean   :3.12   Mean   :3.448   Mean   :3.344
##  3rd Qu.:4.000   3rd Qu.:4.00   3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :7.000   Max.   :7.00   Max.   :7.00   Max.   :7.000   Max.   :6.000
##       Q31             Q32             Q33             Q34
##  Min.   :1.000   Min.   :2.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:4.000
##  Median :4.000   Median :5.000   Median :5.000   Median :5.000
##  Mean   :4.056   Mean   :4.604   Mean   :4.564   Mean   :4.496
##  3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:5.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##       Q35             Q36             Q37             Q38
##  Min.   :1.000   Min.   :2.000   Min.   :1.000   Min.   :2.000
##  1st Qu.:4.000   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:4.000
##  Median :5.000   Median :4.000   Median :5.000   Median :5.000
##  Mean   :4.584   Mean   :4.452   Mean   :4.836   Mean   :4.616
##  3rd Qu.:6.000   3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:6.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##       Q39             Q40             Q41             Q42
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.250   1st Qu.:2.000   1st Qu.:2.000
##  Median :4.000   Median :3.000   Median :4.000   Median :3.000
##  Mean   :3.444   Mean   :3.368   Mean   :3.912   Mean   :3.148
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:6.000   3rd Qu.:4.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##       Q43             Q44             Q45             Q46             Q47
##  Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:3.00   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:3.250
##  Median :3.000   Median :4.00   Median :5.000   Median :5.000   Median :5.000
##  Mean   :3.392   Mean   :4.26   Mean   :4.744   Mean   :4.752   Mean   :4.768
##  3rd Qu.:4.000   3rd Qu.:6.00   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.000
##  Max.   :7.000   Max.   :7.00   Max.   :7.000   Max.   :7.000   Max.   :7.000
```

```
##       Q48             Q49             Q50             Q51
##  Min.   :1.000   Min.   :2.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:2.000
##  Median :5.000   Median :5.000   Median :5.000   Median :4.000
##  Mean   :4.776   Mean   :4.776   Mean   :4.812   Mean   :3.308
##  3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:4.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##       Q52             Q53             Q54             Q55             Q56
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.00   1st Qu.:2.000   1st Qu.:2.000
##  Median :4.000   Median :4.000   Median :3.00   Median :3.000   Median :3.000
##  Mean   :3.532   Mean   :3.616   Mean   :3.16   Mean   :3.136   Mean   :3.148
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.00   Max.   :6.000   Max.   :7.000
##       Q57             Q58             Q59             Q60             Q61
##  Min.   :1.000   Min.   :2.000   Min.   :1.00   Min.   :1.000   Min.   :1.00
##  1st Qu.:3.000   1st Qu.:4.000   1st Qu.:3.00   1st Qu.:3.000   1st Qu.:3.00
##  Median :4.000   Median :4.000   Median :4.00   Median :4.000   Median :4.00
##  Mean   :4.316   Mean   :4.384   Mean   :4.32   Mean   :3.772   Mean   :3.68
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.00   3rd Qu.:5.000   3rd Qu.:5.00
##  Max.   :7.000   Max.   :7.000   Max.   :7.00   Max.   :7.000   Max.   :7.00
##       Q62
##  Min.   :1.000
##  1st Qu.:3.000
##  Median :4.000
##  Mean   :3.672
##  3rd Qu.:5.000
##  Max.   :7.000
```

```
str(dta_all)
```

```
## 'data.frame':    250 obs. of  72 variables:
##  $ respondent   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ preference   : int  1 3 2 3 1 1 1 3 1 1 ...
##  $ Gender       : int  2 1 2 1 2 2 1 1 2 2 ...
##  $ Age          : int  44 24 34 44 41 26 33 48 32 34 ...
##  $ MaritalStatus: int  3 2 3 3 1 1 3 3 1 3 ...
##  $ Children     : int  0 1 1 0 2 1 0 0 3 0 ...
##  $ FirstPurchase: int  2 1 2 2 1 1 2 2 2 2 ...
##  $ AgeCat       : int  5 1 3 5 5 2 3 6 3 3 ...
##  $ ChildCat     : int  0 1 1 0 2 1 0 0 2 0 ...
##  $ IncomeCat    : int  6 3 1 3 4 4 6 4 1 4 ...
##  $ Q1           : int  6 7 5 4 5 6 7 6 6 6 ...
##  $ Q2           : int  2 7 4 2 5 6 7 7 4 2 ...
##  $ Q3           : int  4 7 6 5 7 4 3 3 6 4 ...
##  $ Q4           : int  3 5 5 4 6 4 3 4 1 5 ...
##  $ Q5           : int  1 4 7 2 7 5 5 5 3 1 ...
##  $ Q6           : int  5 4 5 4 3 3 4 3 4 5 ...
##  $ Q7           : int  5 5 3 5 4 5 4 4 4 3 ...
##  $ Q8           : int  3 4 5 4 5 2 4 3 4 4 ...
##  $ Q9           : int  4 5 4 3 4 5 4 5 3 5 ...
##  $ Q10          : int  4 5 5 4 2 3 3 3 4 3 ...
##  $ Q11          : int  4 4 5 4 5 4 7 4 4 4 ...
##  $ Q12          : int  5 4 5 4 4 4 4 4 3 5 ...
```

```
##  $ Q13          : int  4 4 6 6 4 5 5 5 6 4 ...
##  $ Q14          : int  7 2 3 5 5 1 1 1 7 5 ...
##  $ Q15          : int  6 3 3 6 4 2 3 4 7 7 ...
##  $ Q16          : int  7 4 4 7 3 4 5 4 5 6 ...
##  $ Q17          : int  6 4 2 6 2 4 3 5 7 6 ...
##  $ Q18          : int  5 3 4 5 5 5 3 4 5 6 ...
##  $ Q19          : int  5 4 3 5 4 4 4 4 2 6 ...
##  $ Q20          : int  6 2 4 5 5 1 2 2 5 7 ...
##  $ Q21          : int  7 4 2 6 5 3 2 4 6 7 ...
##  $ Q22          : int  5 4 3 6 5 5 4 5 7 7 ...
##  $ Q23          : int  2 7 3 1 3 7 7 7 3 2 ...
##  $ Q24          : int  1 1 4 1 4 1 1 1 5 1 ...
##  $ Q25          : int  2 4 2 1 5 5 5 4 1 3 ...
##  $ Q26          : int  3 3 4 2 2 3 3 4 2 3 ...
##  $ Q27          : int  1 4 5 2 2 5 4 5 4 3 ...
##  $ Q28          : int  1 5 3 2 3 4 6 5 2 2 ...
##  $ Q29          : int  2 7 3 3 1 5 4 5 5 1 ...
##  $ Q30          : int  2 4 4 1 4 3 5 5 4 3 ...
##  $ Q31          : int  4 1 7 2 6 2 1 1 7 3 ...
##  $ Q32          : int  4 5 5 5 7 6 3 5 5 2 ...
##  $ Q33          : int  5 5 7 4 7 3 5 4 5 3 ...
##  $ Q34          : int  4 5 5 5 7 4 2 3 4 6 ...
##  $ Q35          : int  3 3 7 4 5 4 3 4 5 4 ...
##  $ Q36          : int  4 3 6 4 7 6 4 5 3 4 ...
##  $ Q37          : int  4 4 5 3 7 4 3 5 7 4 ...
##  $ Q38          : int  3 7 7 3 7 2 5 4 4 5 ...
##  $ Q39          : int  5 4 3 6 2 3 3 6 5 4 ...
##  $ Q40          : int  3 3 2 2 1 3 2 4 5 4 ...
##  $ Q41          : int  5 7 1 4 3 6 7 7 1 4 ...
##  $ Q42          : int  5 6 1 5 2 4 5 4 2 3 ...
##  $ Q43          : int  4 4 1 4 2 3 5 4 3 4 ...
##  $ Q44          : int  3 7 1 2 3 7 6 7 3 3 ...
##  $ Q45          : int  4 6 4 4 4 6 7 7 5 4 ...
##  $ Q46          : int  4 6 3 5 5 6 7 7 5 4 ...
##  $ Q47          : int  4 7 4 4 5 7 6 6 2 4 ...
##  $ Q48          : int  5 6 4 3 2 7 6 7 5 4 ...
##  $ Q49          : int  4 6 3 3 5 6 6 6 3 5 ...
##  $ Q50          : int  4 7 2 4 3 7 6 6 4 4 ...
##  $ Q51          : int  5 1 4 5 4 2 1 1 3 6 ...
##  $ Q52          : int  4 1 4 2 4 2 2 1 7 5 ...
##  $ Q53          : int  2 1 3 3 6 1 2 1 6 5 ...
##  $ Q54          : int  4 1 5 5 4 2 1 2 4 3 ...
##  $ Q55          : int  5 1 6 4 5 1 1 1 3 5 ...
##  $ Q56          : int  4 1 3 4 5 2 1 1 4 5 ...
##  $ Q57          : int  5 5 4 4 4 5 4 5 6 3 ...
##  $ Q58          : int  3 4 4 2 5 4 5 5 7 4 ...
##  $ Q59          : int  4 3 5 5 4 4 4 4 6 4 ...
##  $ Q60          : int  4 5 3 5 3 4 3 6 2 3 ...
##  $ Q61          : int  4 4 4 5 4 4 5 4 2 3 ...
##  $ Q62          : int  2 5 4 3 5 4 4 4 2 4 ...
```

There are no missing values in this dataset. Some of the variables need to be changed to factors.

Appendix A.2: Factors

```
dta_all$preference <- as.factor(dta_all$preference)
dta_all$Gender <- as.factor(dta_all$Gender)
dta_all$MaritalStatus <- as.factor(dta_all$MaritalStatus)
dta_all$FirstPurchase <- as.factor(dta_all$FirstPurchase)
dta_all$AgeCat <- as.factor(dta_all$AgeCat)
dta_all$ChildCat <- as.factor(dta_all$ChildCat)
dta_all$IncomeCat <- as.factor(dta_all$IncomeCat)
```

Reduce preference categories. If preference is 1, keep 1, but if preference is 2 or 3 change to 0.

```
dta_all$preference_new <- ifelse(dta_all$preference == 1, 1, 0)
dta_all$preference_new <- as.factor(dta_all$preference_new)
```

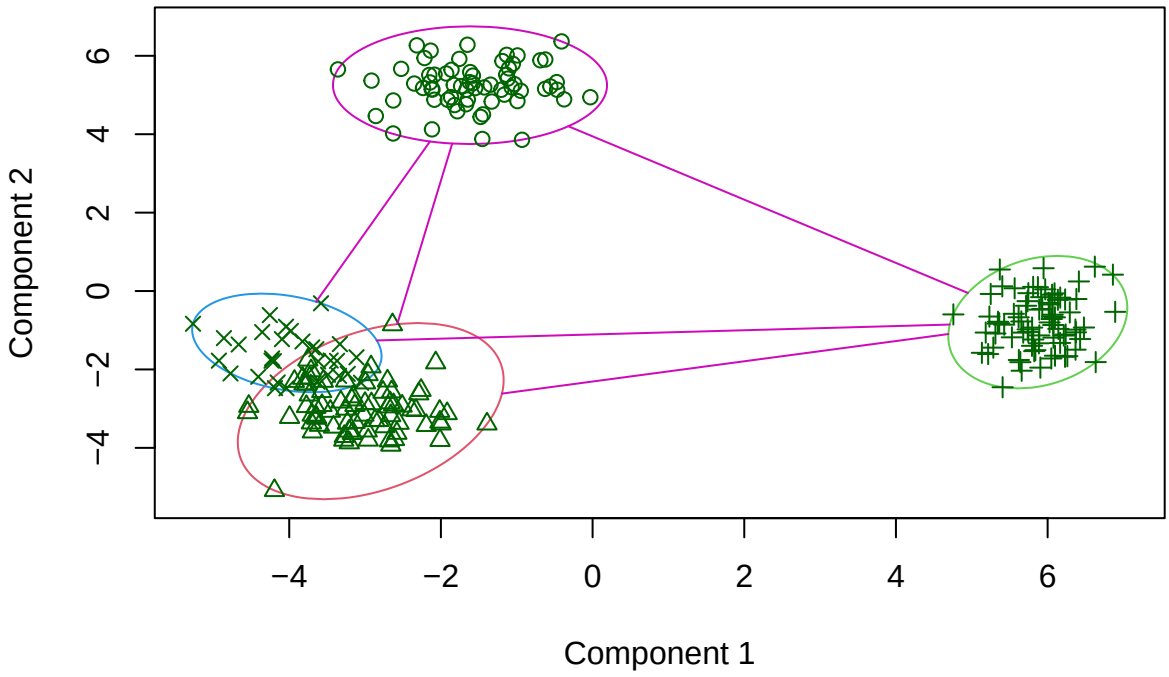
Appendix A.3: Train and Test Sets

```
train.index <- sample(c(1:dim(dta_all)[1]), dim(dta_all)[1]*0.70)
train.df <- dta_all[train.index, ]
test.df <- dta_all[-train.index, ]
```

Appendix A.4: Clustering

```
library(cluster)
#Scale variables
minimums <- apply(dta_all[,c(4, 6, 11:72)],2,min)
ranges <- apply(dta_all[,c(4, 6, 11:72)],2,max)-apply(dta_all[,c(4, 6, 11:72)],2,min)
FordScaled <- scale(dta_all[,c(4, 6, 11:72)],minimums,ranges)
FordScaled <- as.data.frame(FordScaled)

cls <- kmeans(FordScaled,4,iter.max = 1000,nstart = 50)
#rather than plotting all 10 dimensions we use the first 2 principal components
clusplot(FordScaled,cls$cluster,color=TRUE)
```

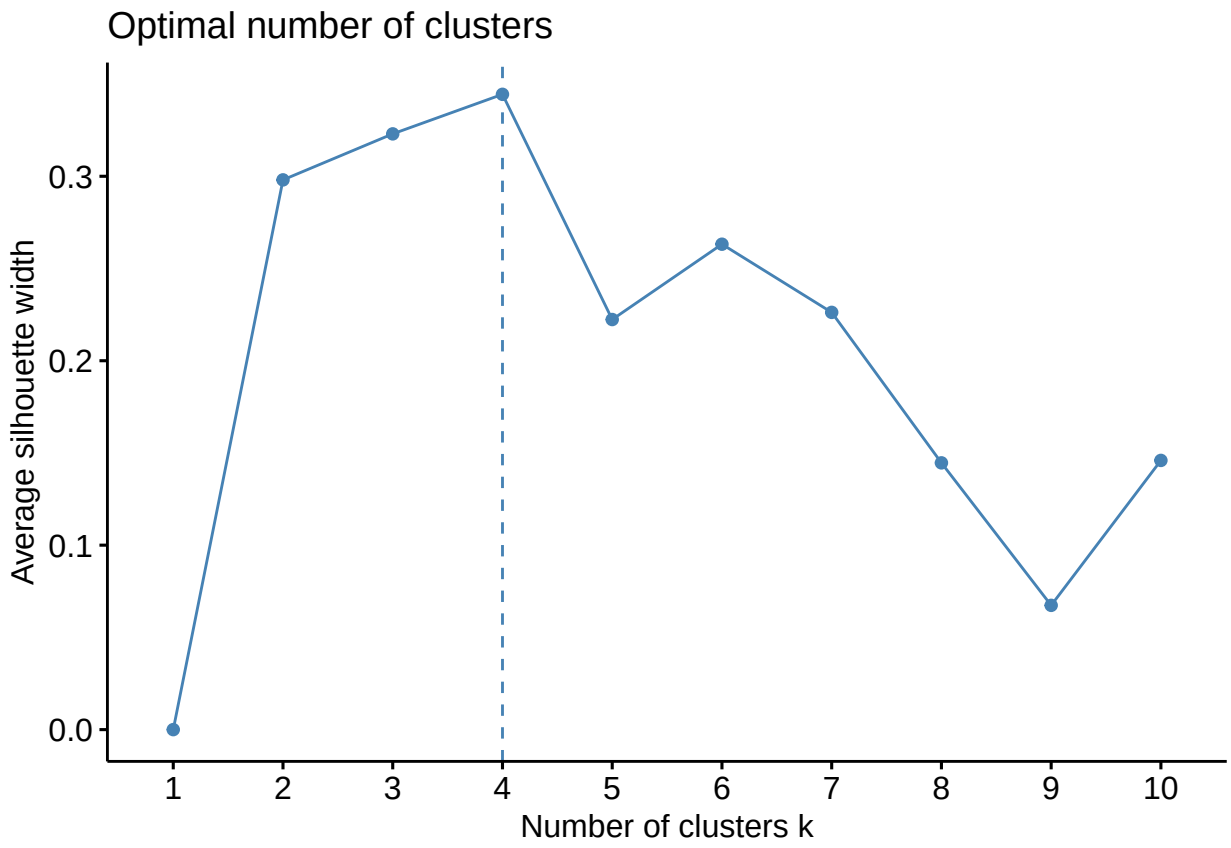



```
library(factoextra)
```

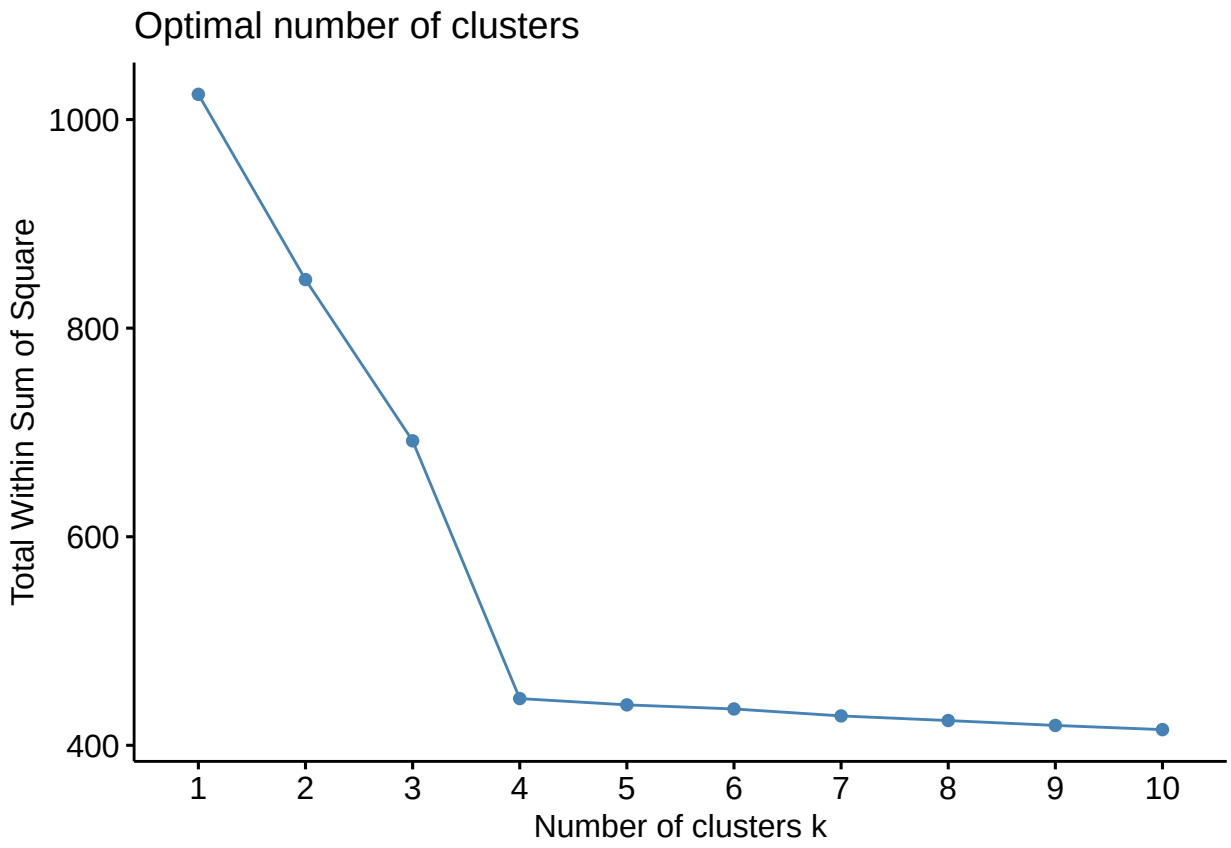
```
## Loading required package: ggplot2
```

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

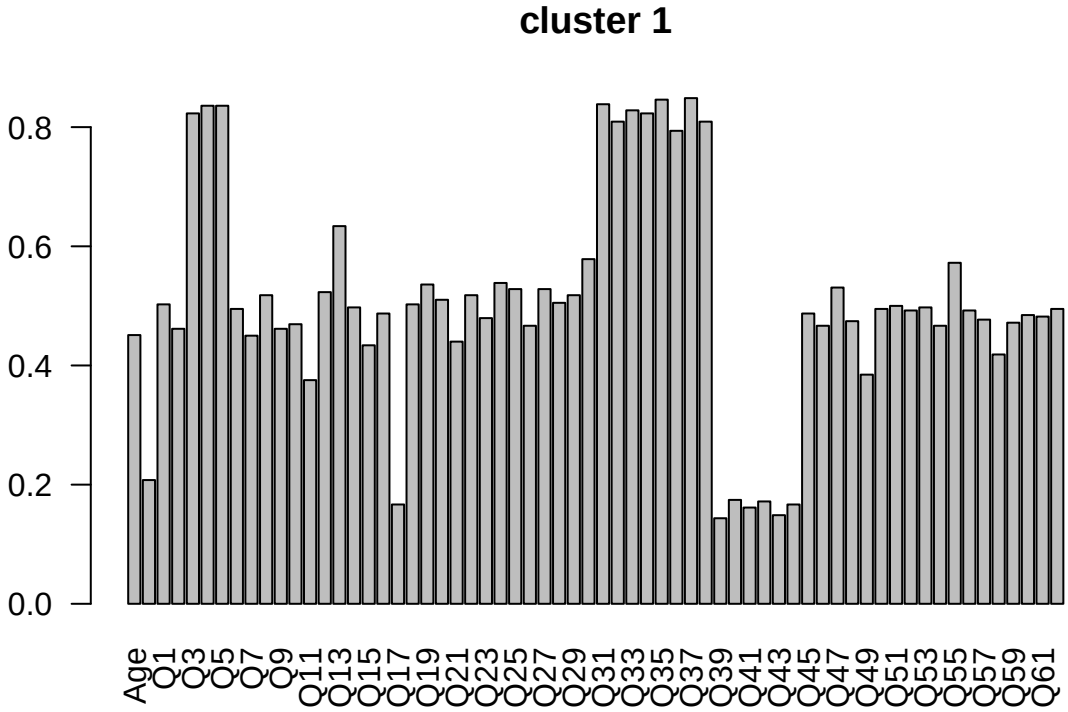
```
fviz_nbclust(FordScaled,kmeans,method="silhouette",k.max = 10)
```



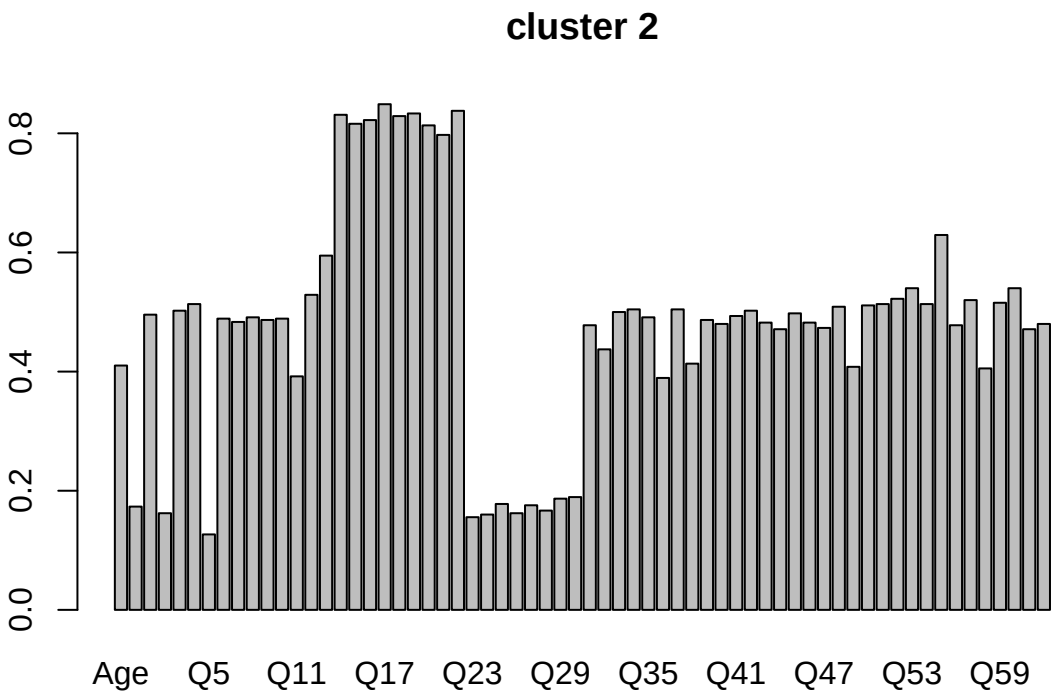

```
fviz_nbclust(FordScaled,kmeans,method="wss",k.max = 10)
```



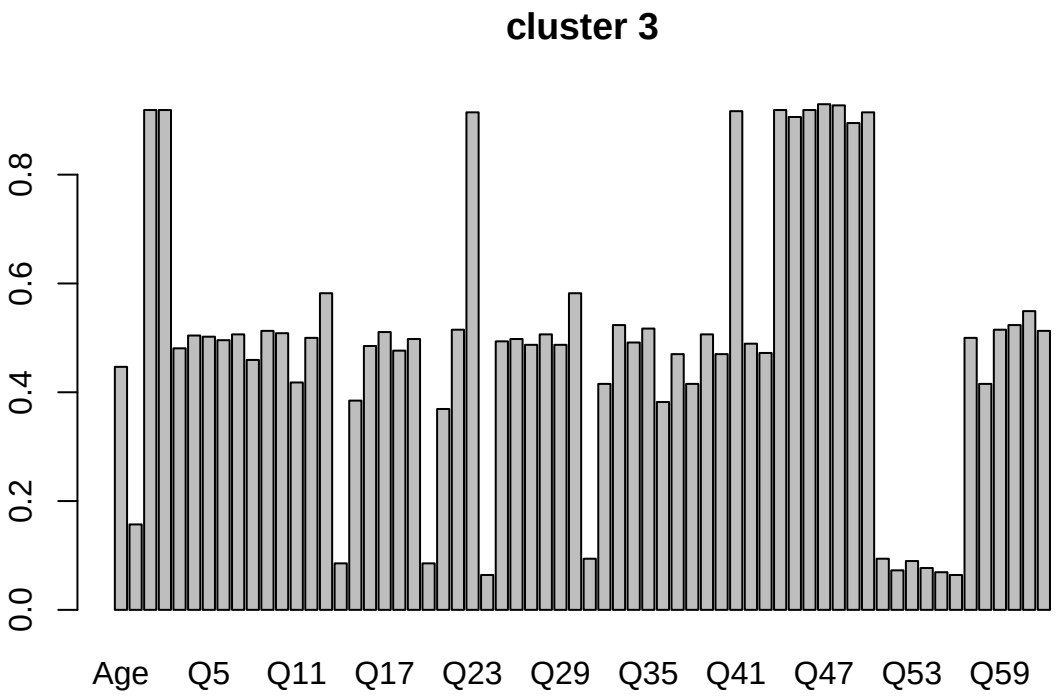

```
barplot(cls$centers[1,],main="cluster 1",las=2)
```



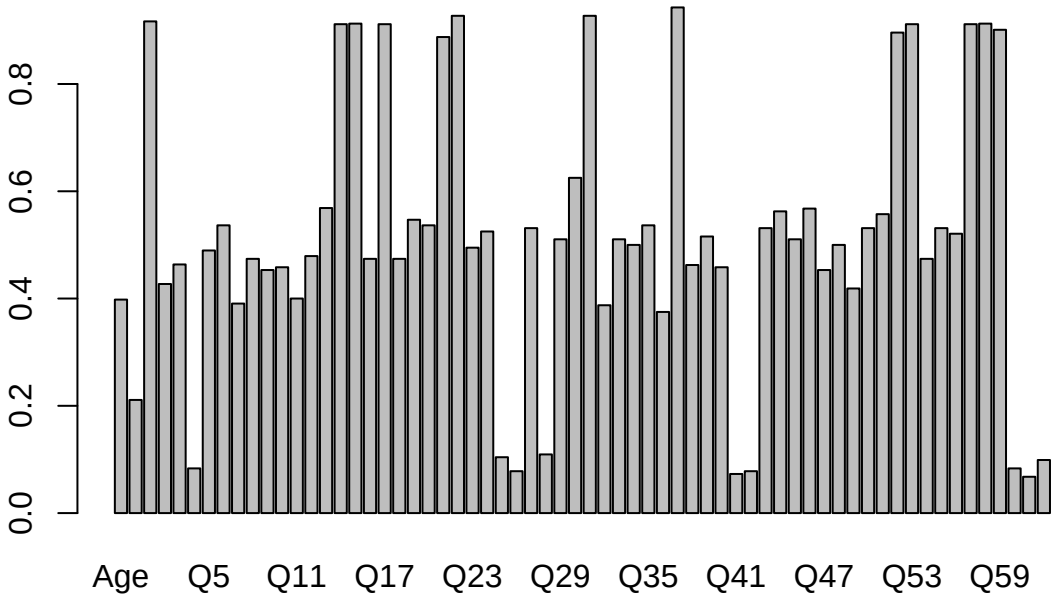
```
barplot(cls$centers[2,],main="cluster 2")
```




```
barplot(cls$centers[3,],main="cluster 3")
```




```
barplot(cls$centers[4,],main="cluster 4")
```



Appendix B: Logistic Regression

Appendix B.1

```
options(scipen=999)
logit.reg1 <- glm(preference_new ~ . - preference - respondent,
                  data = train.df, family = "binomial")
summary(logit.reg1)
```

```
##
## Call:
## glm(formula = preference_new ~ . - preference - respondent, family = "binomial",
##     data = train.df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.99369  -0.53095  -0.00196   0.49694   2.21171
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -25.46839   15.94588  -1.597  0.11023
## Gender2          2.55875    1.25828   2.034  0.04200 *
## Age              0.14461    0.17732   0.816  0.41478
## MaritalStatus2   0.10842    1.43160   0.076  0.93963
## MaritalStatus3  -0.46414    0.87418  -0.531  0.59545
## Children        -1.71611    1.61050  -1.066  0.28662
## FirstPurchase2  -0.10583    1.56787  -0.068  0.94618
## AgeCat2         -2.83641    1.98278  -1.431  0.15257
```

```
## AgeCat3         -3.69999    2.53925  -1.457  0.14508
## AgeCat4         -4.26888    3.22005  -1.326  0.18493
## AgeCat5         -1.44515    3.82635  -0.378  0.70567
## AgeCat6         -4.06094    5.57877  -0.728  0.46666
## ChildCat1        5.08344    2.32278   2.189  0.02863 *
## ChildCat2        5.21927    4.31096   1.211  0.22601
## IncomeCat2      -1.66283    1.49780  -1.110  0.26692
## IncomeCat3      -1.70003    1.63899  -1.037  0.29962
## IncomeCat4       1.37823    1.67482   0.823  0.41056
## IncomeCat5      -2.14559    1.75442  -1.223  0.22134
## IncomeCat6       1.28705    1.75788   0.732  0.46407
## Q1               1.02311    0.60470   1.692  0.09066 .
## Q2               0.21890    0.52136   0.420  0.67459
## Q3               1.89764    0.67488   2.812  0.00493 **
## Q4              -0.48669    0.50163  -0.970  0.33193
## Q5              -0.75757    0.46985  -1.612  0.10688
## Q6              -0.19390    0.50285  -0.386  0.69978
## Q7               1.56365    0.73982   2.114  0.03455 *
## Q8              -0.45528    0.42488  -1.072  0.28393
## Q9               0.83419    0.43680   1.910  0.05616 .
## Q10              0.42885    0.35691   1.202  0.22954
## Q11              0.02391    0.36547   0.065  0.94785
## Q12             -1.57510    0.56682  -2.779  0.00546 **
## Q13              0.59291    0.38144   1.554  0.12009
## Q14             -0.22856    0.49392  -0.463  0.64355
## Q15             -0.96468    0.54895  -1.757  0.07886 .
## Q16              0.28762    0.43903   0.655  0.51240
## Q17              0.77973    0.61699   1.264  0.20632
## Q18              0.98591    0.59576   1.655  0.09795 .
## Q19             -0.07147    0.35697  -0.200  0.84131
## Q20             -0.82739    0.48580  -1.703  0.08854 .
## Q21              0.44306    0.46736   0.948  0.34312
## Q22             -0.75521    0.47735  -1.582  0.11363
## Q23             -0.66553    0.54132  -1.229  0.21890
## Q24              1.94056    0.65924   2.944  0.00324 **
## Q25              0.31609    0.52101   0.607  0.54406
## Q26             -0.01331    0.51269  -0.026  0.97928
## Q27              0.32066    0.57284   0.560  0.57564
## Q28              0.49584    0.48466   1.023  0.30628
## Q29             -0.36624    0.43917  -0.834  0.40432
## Q30              1.41012    0.57810   2.439  0.01472 *
## Q31              0.60493    0.59484   1.017  0.30917
## Q32              0.21174    0.48934   0.433  0.66523
## Q33              0.02316    0.43054   0.054  0.95709
## Q34             -0.89979    0.47816  -1.882  0.05987 .
## Q35             -1.54253    0.58751  -2.626  0.00865 **
## Q36             -1.24183    0.56488  -2.198  0.02792 *
## Q37             -1.51645    0.57766  -2.625  0.00866 **
## Q38              0.50359    0.52806   0.954  0.34026
## Q39             -1.51528    0.59617  -2.542  0.01103 *
## Q40              1.68457    0.64113   2.627  0.00860 **
## Q41             -0.05340    0.52259  -0.102  0.91862
## Q42              0.85842    0.51100   1.680  0.09298 .
## Q43             -0.45889    0.45180  -1.016  0.30978
```

```
## Q44             -0.78972    0.72000  -1.097  0.27272
## Q45              1.92625    0.70210   2.744  0.00608 **
## Q46              1.15159    0.56725   2.030  0.04234 *
## Q47             -0.39711    0.43012  -0.923  0.35587
## Q48             -0.14910    0.42652  -0.350  0.72667
## Q49             -0.99780    0.54062  -1.846  0.06494 .
## Q50              0.15256    0.37268   0.409  0.68229
## Q51              0.30881    0.43658   0.707  0.47936
## Q52              2.42651    0.81619   2.973  0.00295 **
## Q53             -0.45267    0.47781  -0.947  0.34345
## Q54             -0.23714    0.47595  -0.498  0.61830
## Q55             -0.85039    0.49712  -1.711  0.08715 .
## Q56              1.64964    0.59819   2.758  0.00582 **
## Q57              2.08906    0.68583   3.046  0.00232 **
## Q58             -2.17313    0.75197  -2.890  0.00385 **
## Q59             -0.30191    0.41213  -0.733  0.46382
## Q60              0.96724    0.51985   1.861  0.06280 .
## Q61              0.23294    0.45835   0.508  0.61130
## Q62             -0.08108    0.40958  -0.198  0.84307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 241.31  on 174  degrees of freedom
## Residual deviance: 115.83  on  94  degrees of freedom
## AIC: 277.83
##
## Number of Fisher Scoring iterations: 8
```

Appendix B.2: Stepwise Selection

```
summary(logit.step)
```

```
##
## Call:
## glm(formula = preference_new ~ Gender + ChildCat + Q1 + Q3 +
##     Q5 + Q7 + Q8 + Q9 + Q12 + Q15 + Q18 + Q20 + Q23 + Q24 + Q30 +
##     Q31 + Q32 + Q34 + Q35 + Q37 + Q39 + Q40 + Q42 + Q45 + Q46 +
##     Q52 + Q53 + Q56 + Q57 + Q58 + Q60 + Age + Q22, family = "binomial",
##     data = train.df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.16743  -0.72131  -0.05755   0.73385   2.14605
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.68587    6.24988  -3.310 0.000934 ***
## Gender2       0.76390    0.49346   1.548 0.121609
## ChildCat1     1.37467    0.64585   2.128 0.033298 *
## ChildCat2     0.14702    0.51846   0.284 0.776744
```

```
## Q1            0.54599    0.28878   1.891 0.058670 .
## Q3            0.67890    0.25615   2.650 0.008038 **
## Q5           -0.44050    0.22819  -1.930 0.053555 .
## Q7            0.92810    0.31337   2.962 0.003060 **
## Q8           -0.35400    0.24111  -1.468 0.142044
## Q9            0.61939    0.21553   2.874 0.004056 **
## Q12          -0.43563    0.21276  -2.048 0.040606 *
## Q15          -0.42916    0.26787  -1.602 0.109130
## Q18           0.77854    0.24795   3.140 0.001690 **
## Q20          -0.35792    0.23908  -1.497 0.134383
## Q23          -0.34913    0.24855  -1.405 0.160114
## Q24           0.81932    0.31433   2.607 0.009146 **
## Q30           0.57748    0.25716   2.246 0.024732 *
## Q31           0.65951    0.31815   2.073 0.038179 *
## Q32           0.37776    0.24645   1.533 0.125327
## Q34          -0.54849    0.24870  -2.205 0.027423 *
## Q35          -0.70569    0.26167  -2.697 0.007000 **
## Q37          -0.69821    0.26795  -2.606 0.009169 **
## Q39          -1.06617    0.29936  -3.562 0.000369 ***
## Q40           0.83552    0.26515   3.151 0.001627 **
## Q42           0.54904    0.26688   2.057 0.039662 *
## Q45           0.94412    0.31057   3.040 0.002366 **
## Q46           0.35734    0.23519   1.519 0.128672
## Q52           1.13385    0.30389   3.731 0.000191 ***
## Q53          -0.42206    0.26594  -1.587 0.112504
## Q56           0.80691    0.27512   2.933 0.003357 **
## Q57           0.97895    0.27193   3.600 0.000318 ***
## Q58          -0.46533    0.24739  -1.881 0.059970 .
## Q60           0.33358    0.22883   1.458 0.144916
## Age           0.04512    0.02557   1.765 0.077599 .
## Q22          -0.42463    0.24064  -1.765 0.077637 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 241.31  on 174  degrees of freedom
## Residual deviance: 150.09  on 140  degrees of freedom
## AIC: 220.09
##
## Number of Fisher Scoring iterations: 6
```

Appendix B.3: Logistic Error and Accuracy

```
logit.step$fitted.values[1:5]
```

```
##           96          139          147           47          116
## 0.0004382641 0.0732503835 0.5812424043 0.9163910097 0.0003923897
```

```
mean(logit.step$fitted.values)
```

```
## [1] 0.4571429
```

```r
logit.prediction <- predict(logit.step, test.df, type = "response")
predicted.classes <- ifelse(logit.prediction > 0.5, 1, 0)

# error
mean(predicted.classes!=test.df$preference_new)
```

```
## [1] 0.5466667
```

```r
# accuracy
1 - mean(predicted.classes!=test.df$preference_new)
```

```
## [1] 0.4533333
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
confusionMatrix(as.factor(predicted.classes),test.df$preference_new)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 18 20
##          1 21 16
##
##                Accuracy : 0.4533
##                  95% CI : (0.3379, 0.5725)
##     No Information Rate : 0.52
##     P-Value [Acc > NIR] : 0.8982
##
##                   Kappa : -0.0939
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.4615
##             Specificity : 0.4444
##          Pos Pred Value : 0.4737
##          Neg Pred Value : 0.4324
##              Prevalence : 0.5200
##          Detection Rate : 0.2400
##    Detection Prevalence : 0.5067
##       Balanced Accuracy : 0.4530
##
##        'Positive' Class : 0
##
```

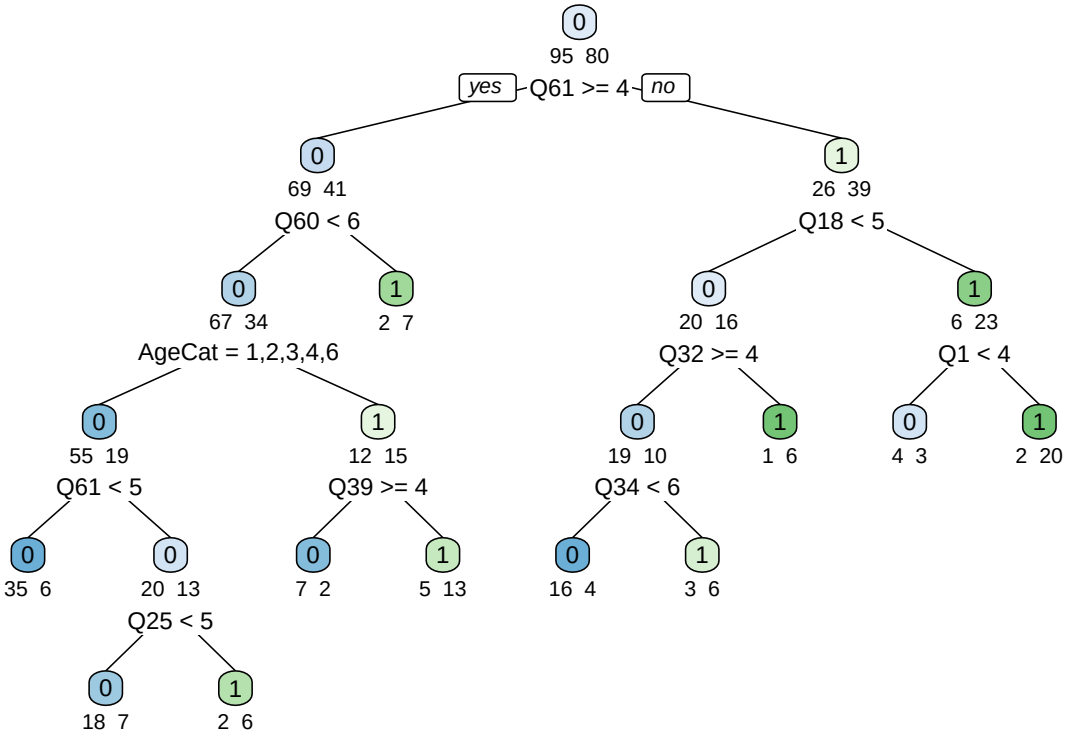
Appendix C: Classification Tree

```
library(rpart)
library(rpart.plot)
```

```
# classification tree
default.ct <- rpart(preference_new ~ . - preference - respondent,
                    data = train.df, method = "class")
length(default.ct$frame$var[default.ct$frame$var == "<leaf>"])
```

```
## [1] 11
```

```
# plot tree
prp(default.ct, type = 2, extra = 1, under = TRUE, split.font = 1, varlen = -10,  box.palette="auto")
```



Appendix C.1: Classification Accuracy

```
default.ct.point.pred <- predict(default.ct,test.df,type = "class")
confusionMatrix(default.ct.point.pred, as.factor(test.df$preference_new))
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  0  1
##          0 22 23
##          1 17 13
##
```

```
##                Accuracy : 0.4667
##                  95% CI : (0.3505, 0.5855)
##     No Information Rate : 0.52
##     P-Value [Acc > NIR] : 0.8508
##
##                   Kappa : -0.0753
##
##  Mcnemar's Test P-Value : 0.4292
##
##             Sensitivity : 0.5641
##             Specificity : 0.3611
##          Pos Pred Value : 0.4889
##          Neg Pred Value : 0.4333
##              Prevalence : 0.5200
##          Detection Rate : 0.2933
##    Detection Prevalence : 0.6000
##       Balanced Accuracy : 0.4626
##
##        'Positive' Class : 0
##
```

Appendix D: Random Forest

```
library(randomForest)
```

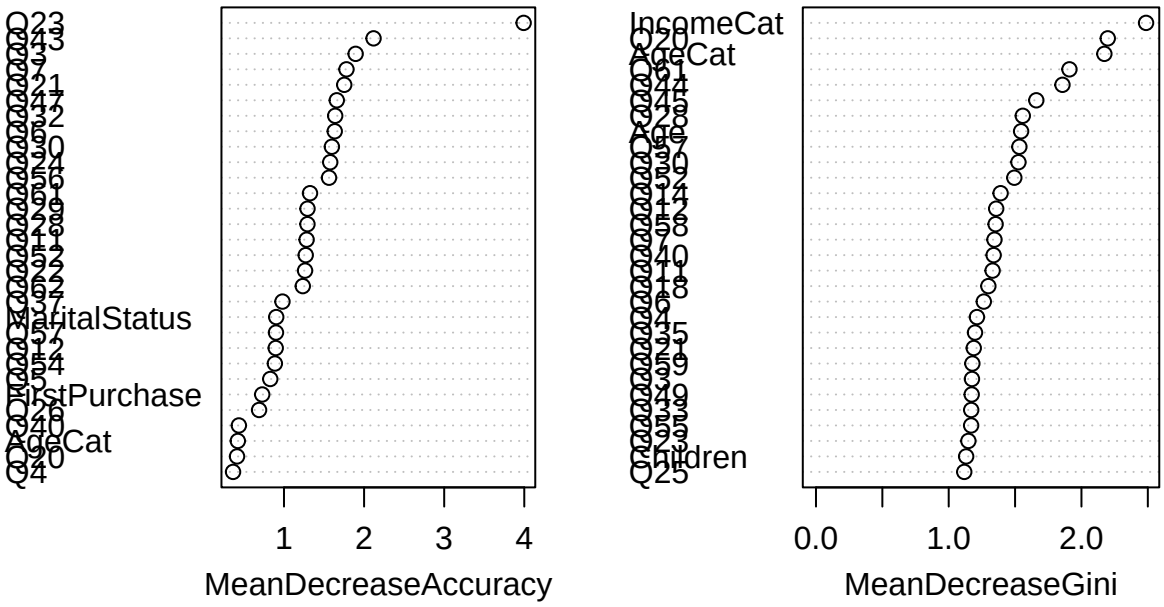
```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rf <- randomForest(as.factor(preference_new) ~ .- preference - respondent,
                   data = train.df, ntree = 50,  mtry = 2, nodesize = 2,
                   importance = TRUE, parms = list(loss = lossmatrix))
varImpPlot(rf)
```



Appendix D.1: Random Forest Accuracy

```
rf.pred <- predict(rf, test.df)
confusionMatrix(rf.pred, as.factor(test.df$preference_new))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 22 24
##          1 17 12
##
##                Accuracy : 0.4533
##                  95% CI : (0.3379, 0.5725)
##     No Information Rate : 0.52
##     P-Value [Acc > NIR] : 0.8982
##
##                   Kappa : -0.1033
##
##  Mcnemar's Test P-Value : 0.3487
##
##             Sensitivity : 0.5641
##             Specificity : 0.3333
##          Pos Pred Value : 0.4783
##          Neg Pred Value : 0.4138
##              Prevalence : 0.5200
```

```
##          Detection Rate : 0.2933
##    Detection Prevalence : 0.6133
##       Balanced Accuracy : 0.4487
##
##        'Positive' Class : 0
##
```