Happiness Around the World.Rmd

2023-04-17

Happiness Around the World

Group members: Avery Zuckerman, Kaitlyn Rouse, and Bella Crain

```
# Load packages
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.1     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/ conflicted package]8;; to force all confl
icts to become errors
```

```
library(tidytext)
library(textdata)
library(ggplot2)
library(readr)
library(ade4)
library(ggcorrplot)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WB
a
```

```
library(plotROC)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

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

1.) Introduction

a. Description of Dataset

The "Happiness and Corruption Globally" dataset measures happiness on a global scale by country but was renamed to "Happy" upon importing. We intend to use the variable relating to 'happiness_score' as the outcome variable and 'freedom', 'Year', 'family', 'health', and 'social_support' as the predictor variables. The 'freedom' variable represents "The extent to which Freedom contributed to the calculation of the Happiness Score." The 'Year' variable represents which year 2015-2020 the data is from. The 'health' variable is "the extent to which Life expectancy contributed to the calculation of the Happiness Score." The 'social_support' variable represents " the perception and actuality that one is cared for, and has assistance available from other people." The 'happiness_score' variable represents "average of responses to the primary life evaluation question from the Gallup World Poll (GWP). 0-10."(Kaggle, 2020) This dataset was obtained through the website Kaggle. Each row in the dataset represents a distinct county in the world and the years range from 2015 to 2020. An expected trend within the dataset is that 'family', 'health', 'social_support', and 'freedom' all have a direct relationship with 'happiness_score'. The dataset was already tidy because each variable has its own respective column and each observation has its own row. This dataset was intriguing to our group because happiness is something that is so highly desired throughout the world, so we wanted to investigate how factors such as freedom, year, family, health, and social_support could potentially impact the happiness of individuals around the world.

```
# Loading the dataset
library(readr)
Happy <- read_csv("~/SDS 322E/Project/Happy.csv")
```

```
## Rows: 792 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (2): Country, continent
## dbl (11): happiness_score, gdp_per_capita, family, health, freedom, generosi...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

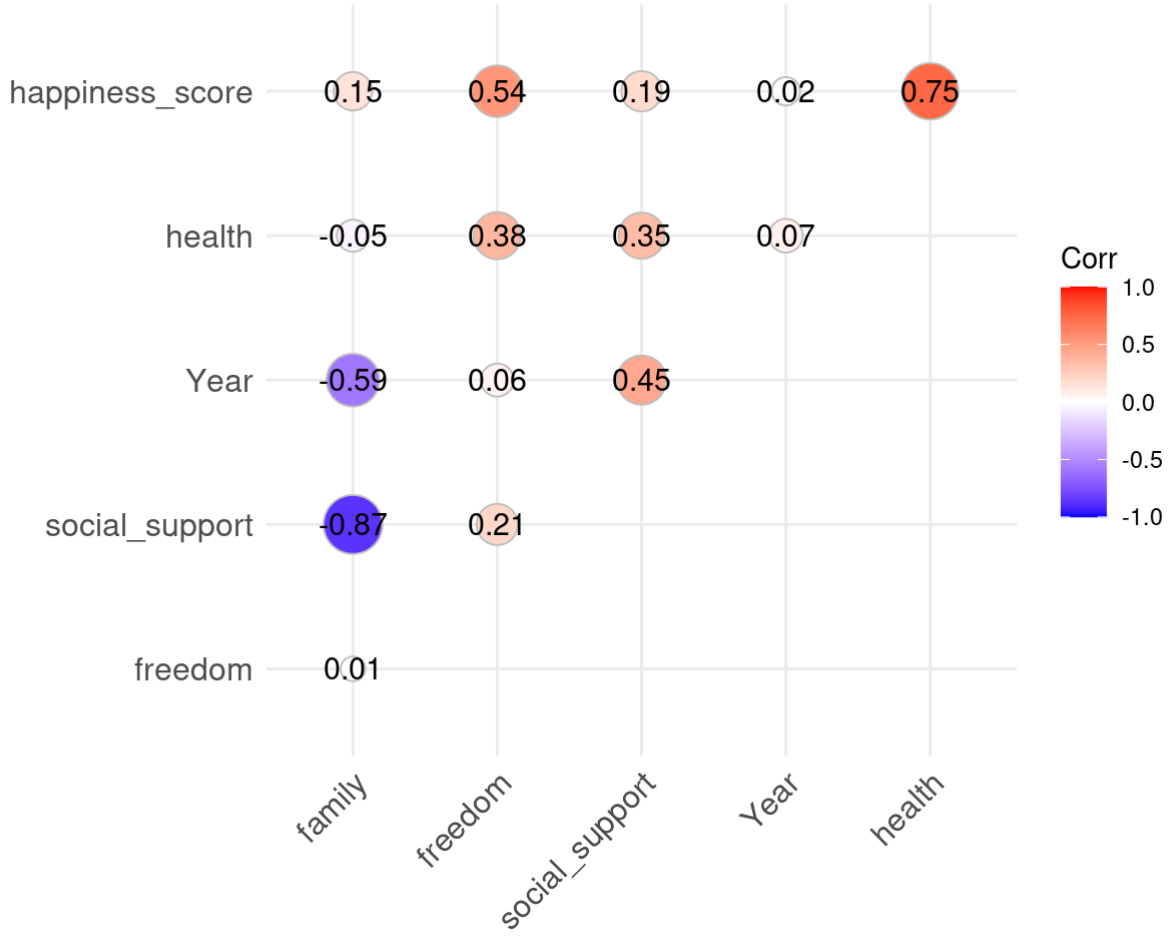
b. Defining Research Questions

The first research question aims to discover "How does freedom and health impact the happiness score globally?". We might expect to see that freedom and health both have a positive correlation with happiness score. The second research question tackles the question of "How does family and social support impact the happiness score?". We might expect to see that family and social support have a positive correlation with happiness score. Finally, the third research question examines "How does family and year impact happiness score globally?" We might expect to see that happiness scores increase with higher family contribution to happiness score and increasing years. ## 2.) Creating a Correlation Matrix

```
# Selecting for Desired Variables
Happy <- Happy%>%
  select(family, freedom, social_support, Year, health, happiness_score) %>%
  na.omit
# Creating a Correlation Matrix
cor(Happy)
```

```
##                      family    freedom social_support        Year      health
## family           1.00000000 0.01383279     -0.8698845 -0.58862993 -0.05468332
## freedom          0.01383279 1.00000000      0.2087090  0.05819511  0.38186939
## social_support  -0.86988453 0.20870902      1.0000000  0.44861592  0.34743827
## Year            -0.58862993 0.05819511      0.4486159  1.00000000  0.07017157
## health          -0.05468332 0.38186939      0.3474383  0.07017157  1.00000000
## happiness_score  0.15494606 0.54428441      0.1926331  0.02349506  0.75353435
##                 happiness_score
## family               0.15494606
## freedom              0.54428441
## social_support       0.19263307
## Year                 0.02349506
## health               0.75353435
## happiness_score      1.00000000
```

```
# Visualizing the Correlation Matrix
ggcorrplot(cor(Happy),
           type = "upper", # upper diagonal
           lab = TRUE, # print values
           method = "circle")
```



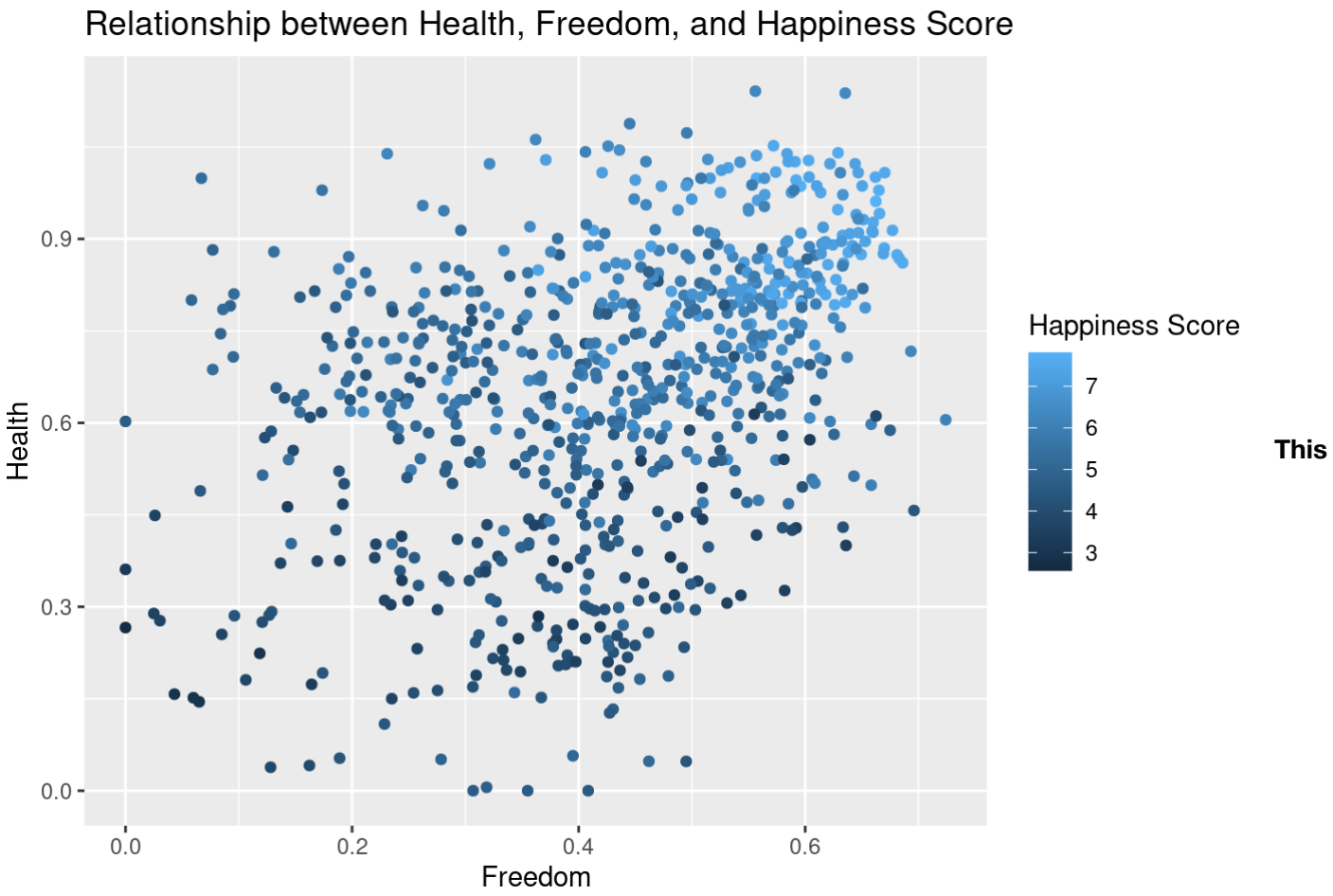
The

Correlation Matrix indicates that the variables with the highest correlation within the "Happy" dataset are 'family' and 'social_support' and 'happiness_score' and 'health' which have a correlation of around 0.87 and 0.75 respectively. The variables with the lowest correlation within the "Happy" dataset are 'family' and 'freedom' and 'Year' and 'happiness_score' which has a correlation of around 0.014 and 0.023 respectively.

3.) Research Question 1: "How does freedom and health impact the happiness score globally?"

Visualizations

```
#Visualization 1 for Research Question 1
Happy%>%
  ggplot(aes(x=freedom, y=health, color = happiness_score))+geom_point()+
  labs(title = "Relationship between Health, Freedom, and Happiness Score", x= "Freedo
m", y= "Health", color= "Happiness Score")
```

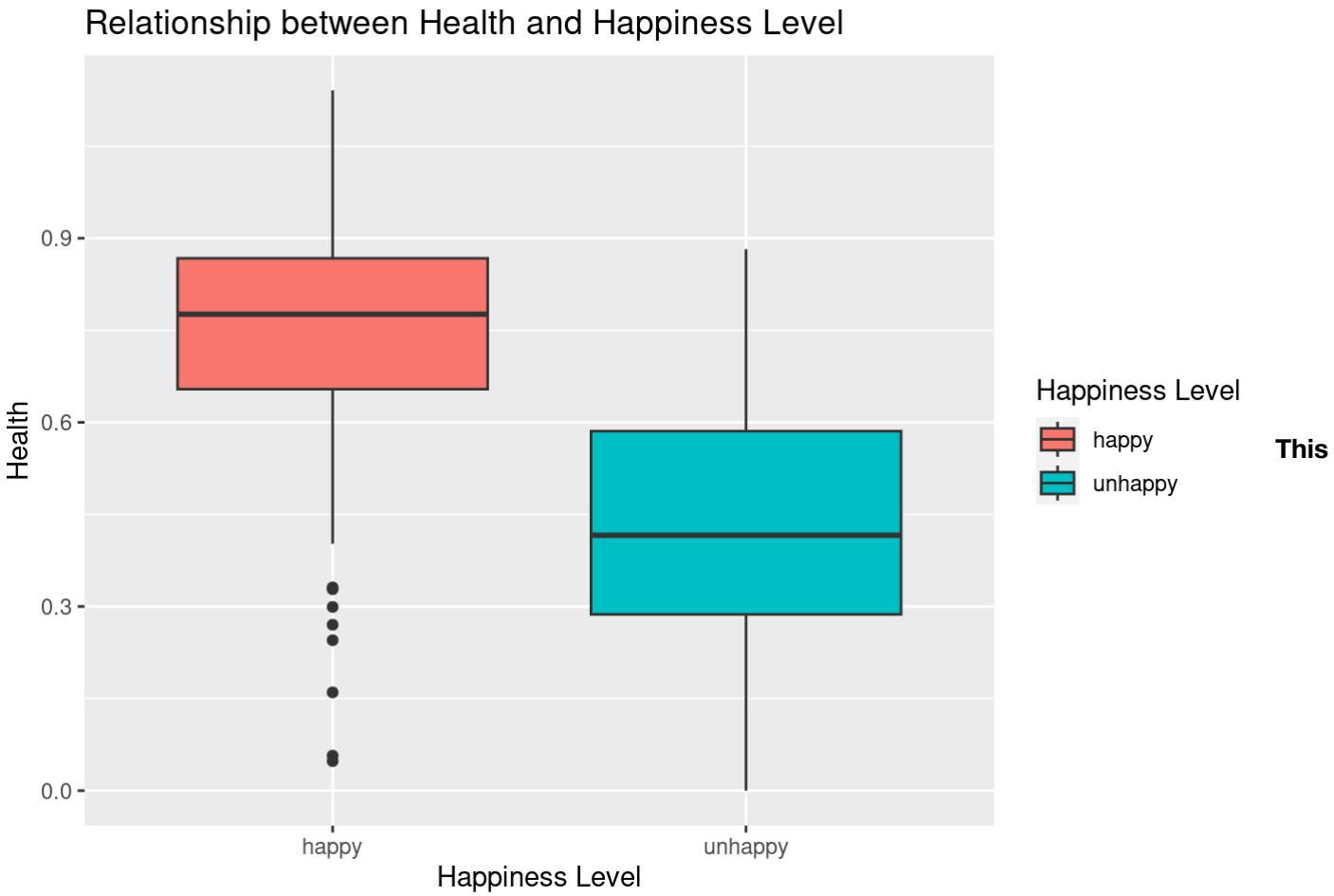


This scatterplot showcases the relastionship between 'freedom', 'health', and 'happiness_score'. Based on the visualization, happiness score is higher when both freedom and health are higher. Also, the scatter plot reveals that freedom and health are positively correlated with each other and increase together.

```
# Make happiness_score "happy" if 1 and "unhappy" if 0
Happy_mut <- Happy %>%
  mutate(happiness_score_mut = ifelse(happiness_score > 5, 1, 0))
head(Happy_mut)
```

```
## # A tibble: 6 × 7
##   family freedom social_support  Year health happiness_score happiness_score_mut
##    <dbl>   <dbl>          <dbl> <dbl>  <dbl>           <dbl>                <dbl>
## 1   1.53   0.635              0  2015  0.797            7.54                    1
## 2   1.55   0.626              0  2015  0.793            7.52                    1
## 3   1.61   0.627              0  2015  0.834            7.50                    1
## 4   1.52   0.620              0  2015  0.858            7.49                    1
## 5   1.54   0.618              0  2015  0.809            7.47                    1
## 6   1.43   0.585              0  2015  0.811            7.38                    1
```

```
#Visualization 2 for Research Question 1
Happy <- Happy_mut%>%
  mutate(Happiness_Level = case_when(happiness_score_mut == 1 ~ 'happy',
                           happiness_score_mut == 0 ~ 'unhappy'))
Happy%>%
  ggplot(aes(x = Happiness_Level, y = health, fill=Happiness_Level)) +
  geom_boxplot() +
  labs(title = "Relationship between Health and Happiness Level", x= "Happiness Level",
y= "Health", fill= "Happiness Level")
```

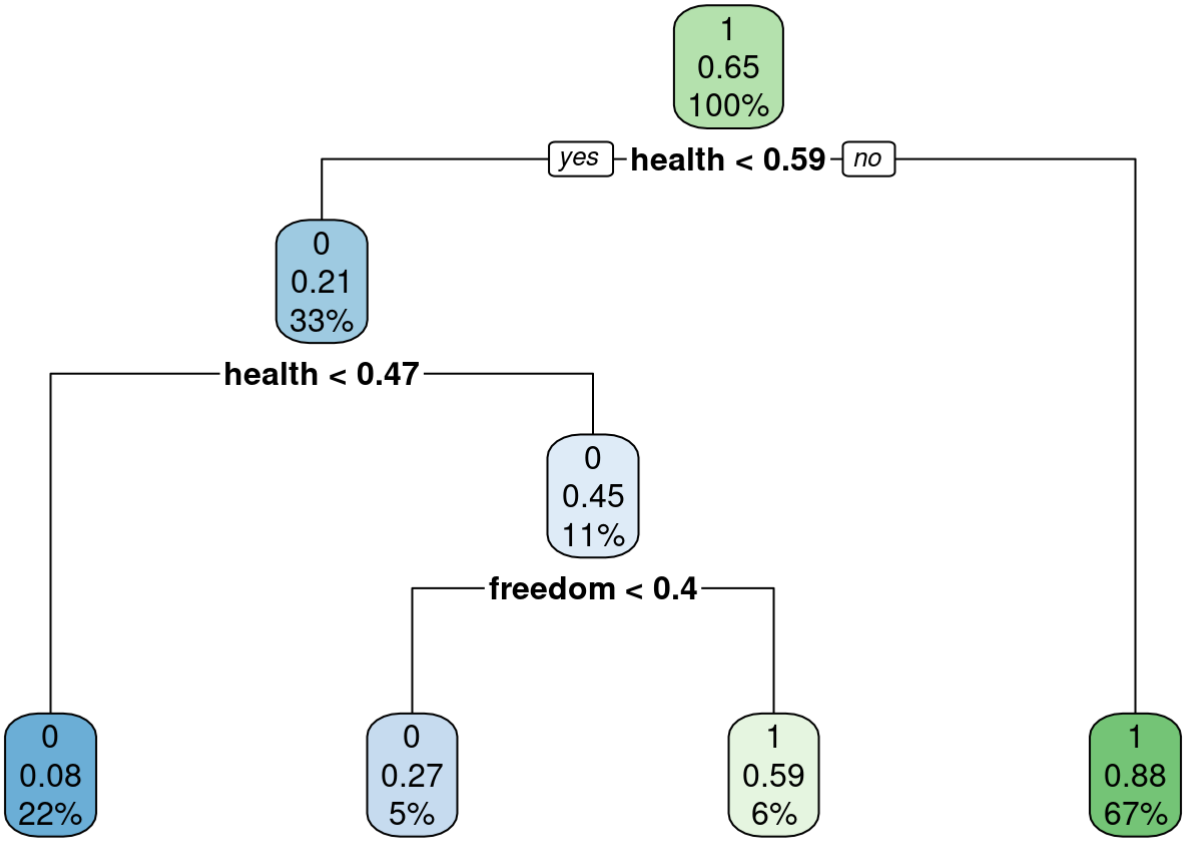


This boxplot shows the relationship between health and happiness level. Based on the visualization, those that are happy have a higher value for health and a smaller range of value for the interquartile range. On the other hand, those that are unhappy have a lower health level and a broader interquartile range. In addition, the interquartile ranges for happy and unhappy values do not overlap, which means the health values for unhappy and happy levels are significantly different from each other. ## Prediction Model (Decision Tree) and Cross Validation for Research Question 1

```
# Make happiness_score "happy" if 1 and "unhappy" if 0
Happy_mut <- Happy %>%
  mutate(happiness_score_mut = ifelse(happiness_score > 5, 1, 0))
head(Happy_mut)
```

```
## # A tibble: 6 × 8
##   family freedom social_support  Year health happiness_score happiness…¹ Happi…²
##    <dbl>   <dbl>          <dbl> <dbl>  <dbl>           <dbl>       <dbl> <chr>
## 1   1.53   0.635              0  2015  0.797            7.54           1 happy
## 2   1.55   0.626              0  2015  0.793            7.52           1 happy
## 3   1.61   0.627              0  2015  0.834            7.50           1 happy
## 4   1.52   0.620              0  2015  0.858            7.49           1 happy
## 5   1.54   0.618              0  2015  0.809            7.47           1 happy
## 6   1.43   0.585              0  2015  0.811            7.38           1 happy
## # … with abbreviated variable names ¹happiness_score_mut, ²Happiness_Level
```

```
# Define a sampling process
sample_process <- sample(c(TRUE, FALSE), # take value TRUE or FALSE
                 nrow(Happy_mut), # for each row in biopsy
                 replace = TRUE, # TRUE or FALSE can repeat
                 prob = c(0.7, 0.3)) # 70% TRUE, 30% FALSE
# Select values for the train set (corresponding to TRUEs in sample_process)
train_happy <- Happy_mut[sample_process, ]
# Select values for the test set (corresponding to FALSEs in sample_process)
test_happy <- Happy_mut[!sample_process, ]
# Decision tree
# Visualize the decision tree
happy_tree <- rpart(happiness_score_mut ~ freedom + health, # model
                    data = Happy_mut, # data
                    method = "class") # classification
rpart.plot(happy_tree)
```




```
  #ree for train
ROC_tree_train <- (ggplot(train_happy) +
  geom_roc(aes(d = happiness_score_mut, m = predict(happy_tree, train_happy)[,2]), n.cut
s = 0))
calc_auc(ROC_tree_train)
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
##   PANEL group     AUC
## 1     1    -1 0.84255
```

```
# tree for test
ROC_tree_test <- (ggplot(test_happy) +
  geom_roc(aes(d = happiness_score_mut, m = predict(happy_tree, test_happy)[,2]), n.cuts
= 0))
calc_auc(ROC_tree_test)
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
##   PANEL group       AUC
## 1     1    -1 0.8656575
```

```
# Cross validation for the decision tree model
tree_cv <- train(happiness_score_mut ~ freedom + health,
      data = Happy_mut,
      method = "rpart",
      trControl = trainControl(method = "cv", number = 10))
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
# Let's take a look at the output
tree_cv
```

```
## CART
##
## 792 samples
##   2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 712, 713, 713, 713, 713, 713, ...
## Resampling results across tuning parameters:
##
##   cp          RMSE       Rsquared   MAE
##   0.03060988  0.3512868  0.4582540  0.2386323
##   0.04563706  0.3633577  0.4185174  0.2610252
##   0.43835162  0.4031629  0.3493519  0.3092542
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.03060988.
```

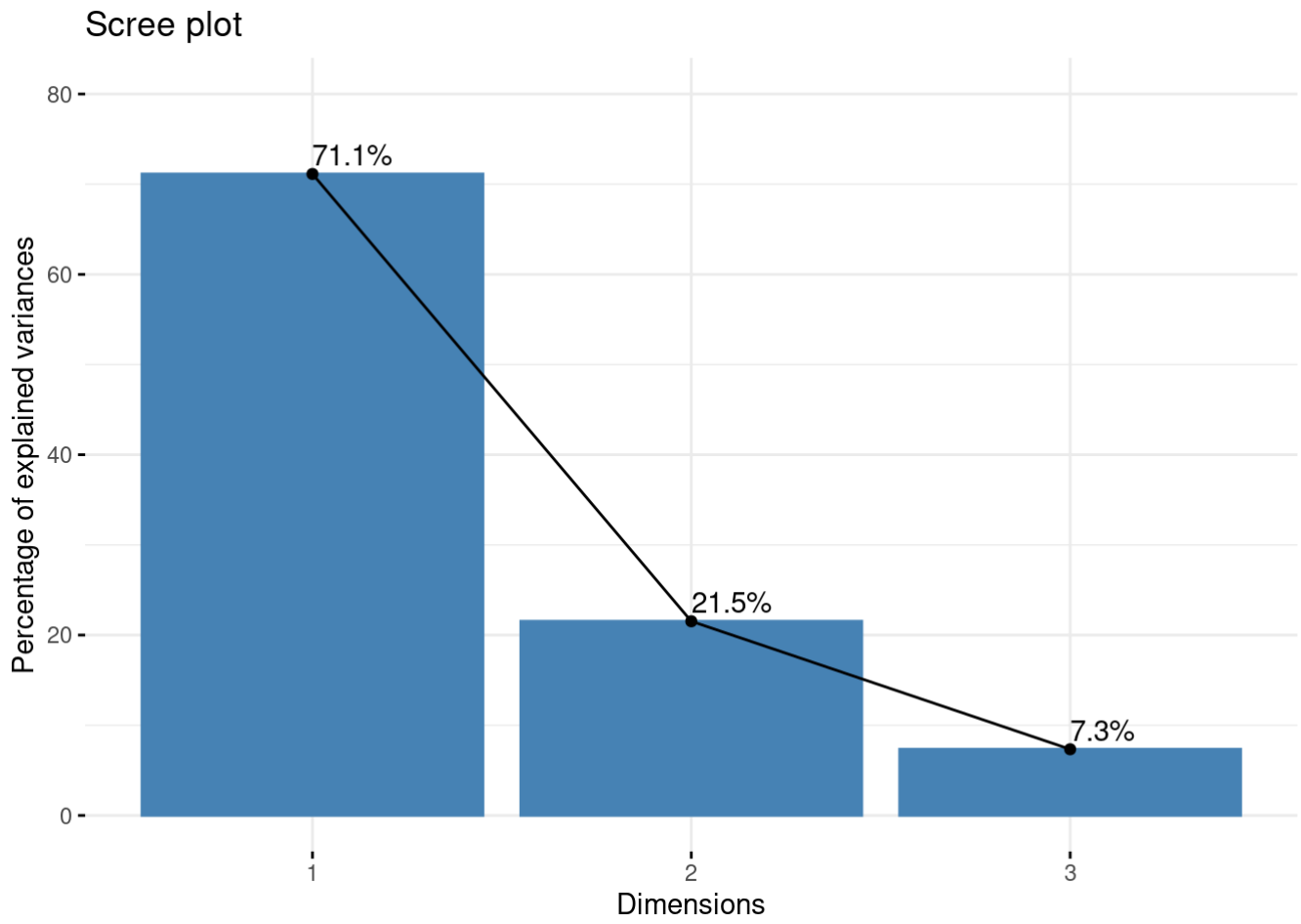
The prediction model for the Decision Tree model for the research question "How does freedom and health impact the happiness score globally?" was accomplished and ROC curves were constructed for both the train data and the test data. The train data for the Decision Tree's ROC curve has an AUC value of

0.8550482 while the test data for the Decision Tree's ROC curve has an AUC value of 0.8389272. This demonstrates that the Decision Tree's model for addressing the research question "How does freedom and health impact the happiness score globally?" has a predictability accuracy of around 85.5% for the train data and 83.9% for the test data which demonstrates that the model is functioning accurately since the AUC score ranges from 0-100%. Additionally, since both the test and train data have similar AUC scores, overfitting is likely not a major issue. A 10-fold cross validation was then executed which presented 3 cp values ranging from 0.03060988 to 0.43835162 demonstrating trees of increasing complexity. In the end, it was determined that the model is more accurate for a less complex tree as the optimal model is for the smallest cp value of 0.03060988.The RMSE value for that decision tree is 0.3527780 which demonstrates that the model is functioning accurately because a low RMSE value demonstrates that there is smaller variation between the predicted values and the actual values. ## PCA for Research Question 1

```
#Applying PCA to the Desired Variables
pca1 <- Happy %>%
  select(happiness_score, freedom, health) %>%
  scale %>%
  prcomp
pca1
```

```
## Standard deviations (1, .., p=3):
## [1] 1.4608423 0.8036426 0.4691464
##
## Rotation (n x k) = (3 x 3):
##                         PC1        PC2        PC3
## happiness_score -0.6327083  0.1794172  0.7533191
## freedom         -0.5021650 -0.8355903 -0.2227537
## health          -0.5895003  0.5192286 -0.6187819
```

```
# Creating a scree plot
fviz_eig(pca1, addlabels = TRUE, ylim = c(0, 80))
```




```
# Visualize the contributions of the variables to the PC1 in a table
get_pca_var(pca1)$coord %>% as.data.frame %>%
  arrange(Dim.1) %>% select(Dim.1)
```

```
##                      Dim.1
## happiness_score -0.9242870
## health          -0.8611670
## freedom         -0.7335839
```

```
# Visualize the contributions of the variables to the PC2 in a table
get_pca_var(pca1)$coord %>% as.data.frame %>%
  arrange(Dim.2) %>% select(Dim.2)
```

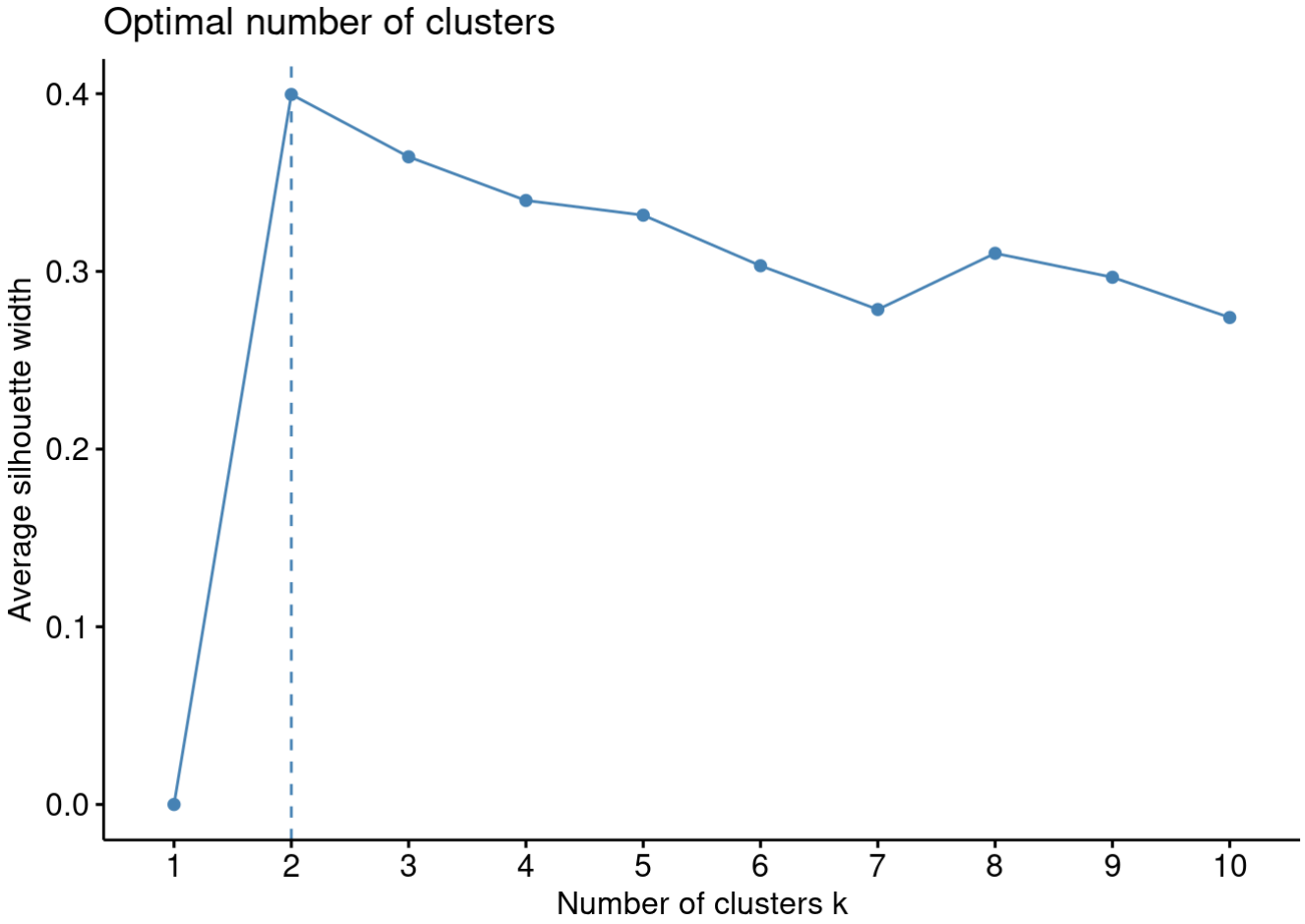
```
##                      Dim.2
## freedom         -0.6715159
## happiness_score  0.1441873
## health           0.4172742
```

The scree plot demonstrates that the percentage of explained variance for the first principal component is 71.1% while the percentage of explained variance for the second principal component is 21.5% ## Kmeans Clustering for Research Question 1

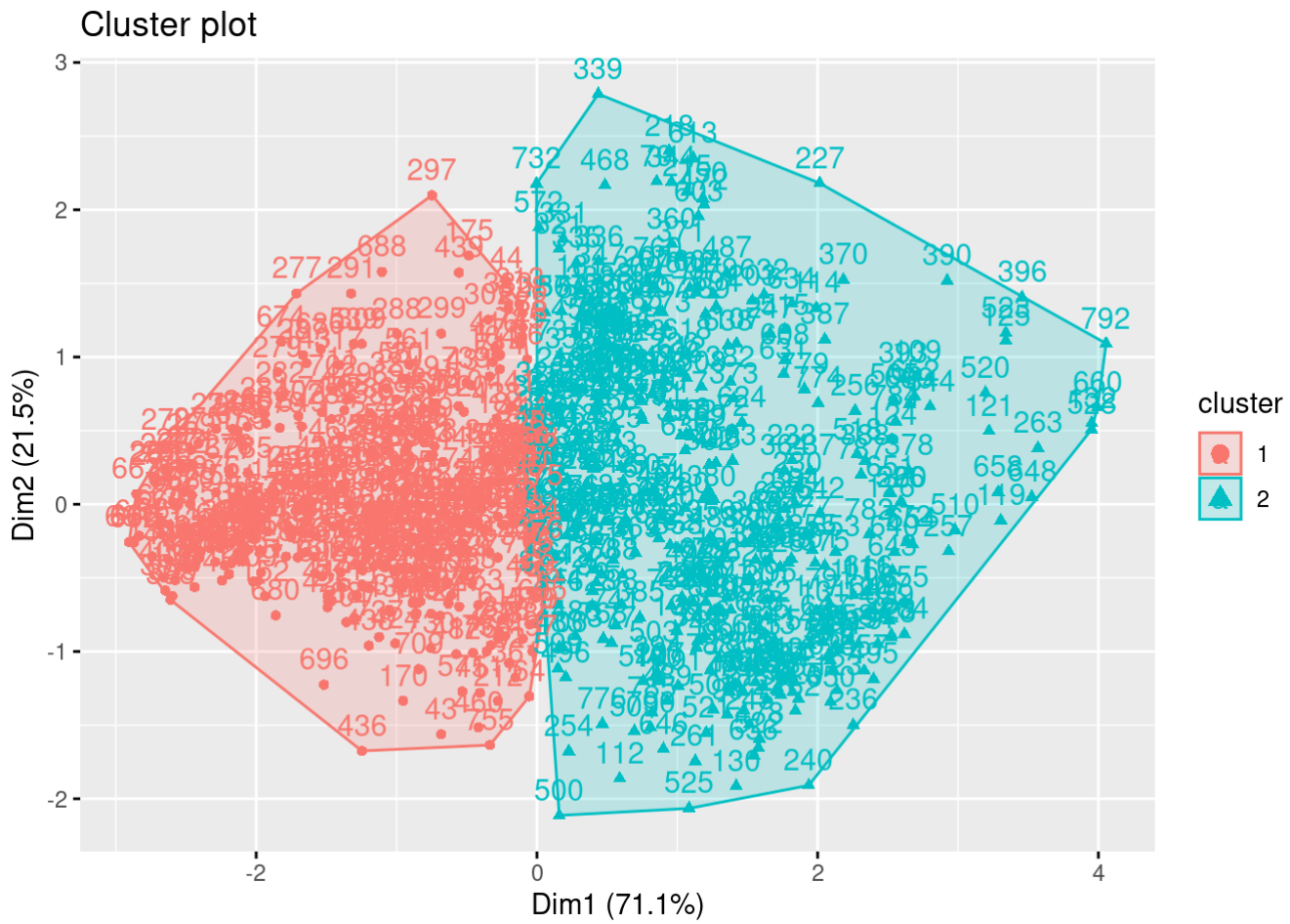
```
# Scaling the Variables
Happy1_scaled <- Happy_mut %>%
  select(happiness_score, freedom, health) %>%
  scale
# Use the function kmeans() to find clusters
kmeans_results <- Happy1_scaled %>%
  kmeans(centers = 2)
kmeans_results
```

```
## K-means clustering with 2 clusters of sizes 402, 390
##
## Cluster means:
##   happiness_score    freedom     health
## 1       0.7600217  0.6333104  0.6564440
## 2      -0.7834070 -0.6527968 -0.6766423
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 2 1 2 2 2 2 1 1 2 1 1 1 1 1 1 2 2 2 1 1 2 2 2 1 2 2 1 2 2 2
##  [75] 2 2 2 1 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1
## [186] 1 2 1 2 1 1 2 2 2 1 1 1 1 2 2 2 2 2 2 2 1 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2 2
## [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [260] 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 2 1 2 1 1
## [334] 1 2 2 1 2 2 2 2 1 2 2 1 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [371] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1
## [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [445] 1 1 1 2 1 1 1 1 1 1 1 2 2 2 1 1 1 2 2 2 1 2 1 2 2 2 1 2 1 1 2 2 2 2 1 2 2
## [482] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [519] 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [556] 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 2 1 1 1 2 2 1 1 2 2 1 2 2 1 2 2 1 1 2 1 2
## [593] 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [630] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1
## [667] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [704] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 2 1
## [741] 2 1 2 1 1 1 1 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [778] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 456.5639 765.6545
##  (between_SS / total_SS =  48.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# Maximize the silhouette while keeping a small number of clusters
fviz_nbclust(Happy1_scaled, kmeans, method = "silhouette")
```




```
fviz_cluster(kmeans_results, data = Happy1_scaled)
```

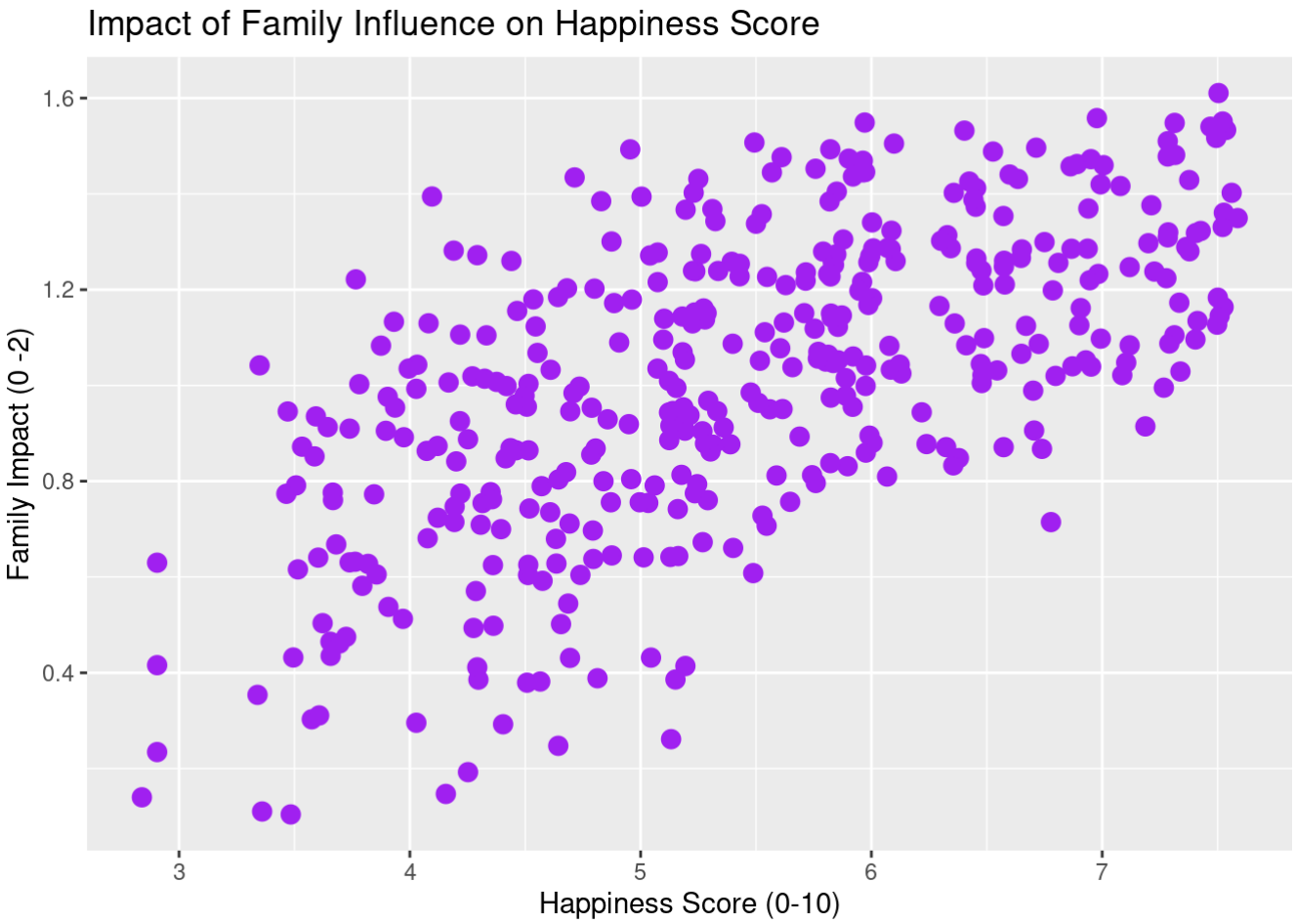



```
# Summary statistics
Happy_mut %>%
  select(happiness_score, freedom, health) %>%
  mutate(cluster = as.factor(kmeans_results$cluster)) %>%
  group_by(cluster) %>%
  summarize_if(is.numeric, mean, na.rm = T)
```

```
## # A tibble: 2 × 4
##   cluster happiness_score freedom health
##   <fct>             <dbl>   <dbl>  <dbl>
## 1 1                  6.33   0.521  0.801
## 2 2                  4.59   0.330  0.492
```

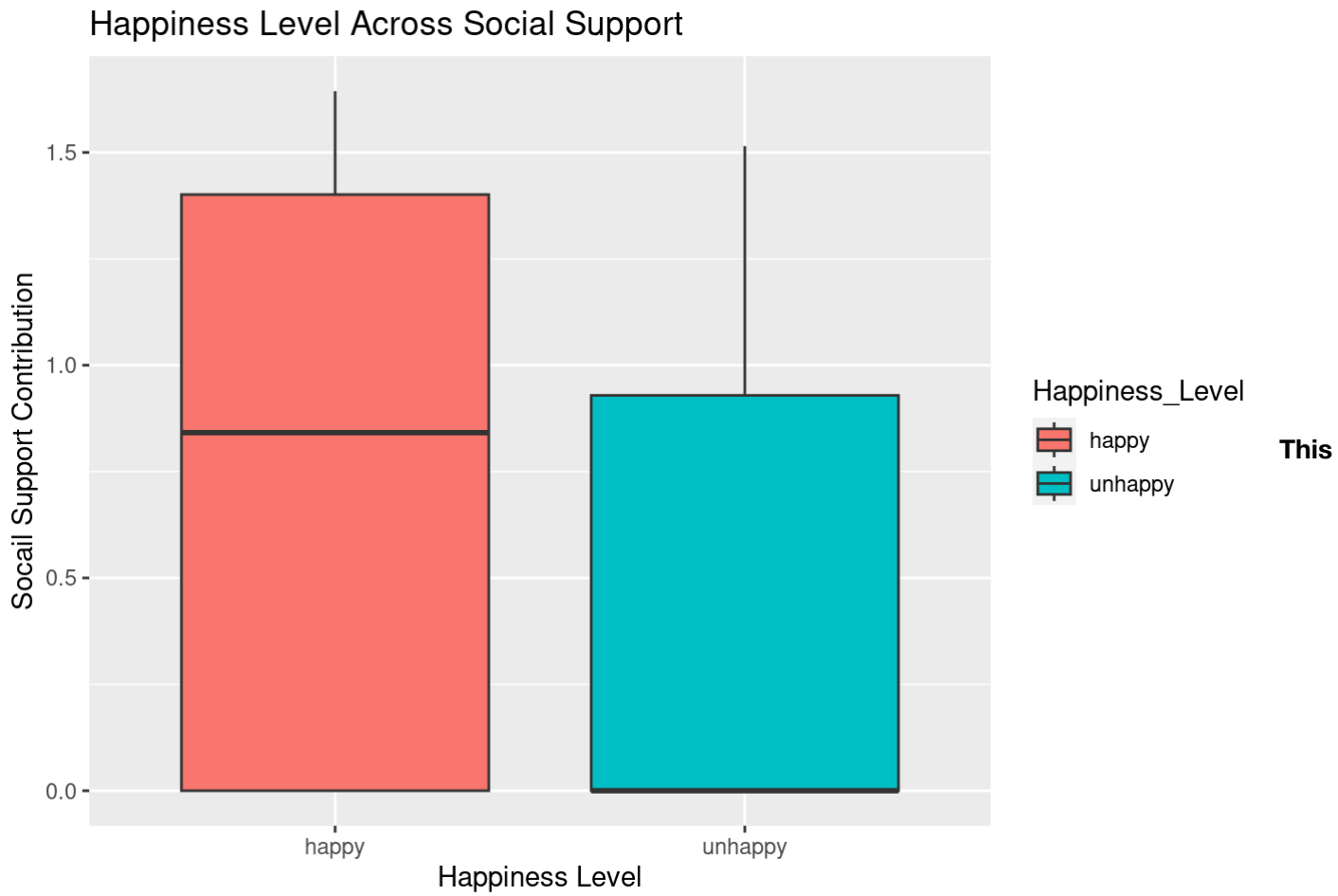
The average silhouette width indicates that 2 clusters maximize the average width silhouette. Additionally, through visualizing the clusters, it is evident that the two clusters and distinguishable from one another. Summary statistics were run on the clusters to determine the differences in the means for the selected variables. This demonstrated that cluster 1 has a higher average "happiness_score", "freedom", and "health" scores than cluster 2 with average values of 6.328126, 0.5207485, 0.8011574 respectively.This is indicative of cluster 1 representing the observations with higher average scores for all of the selected variables while cluster 2 represents the observations with lower average scores for all of the selected variables. ## 4.) Research Question 2: "How does family and social support impact the happiness score globally?" ## Visualizations

```
#Setting up dataset for visualization
Happy_nm <- Happy[!is.na(Happy$family), ]
Happy_vis <- Happy_nm %>%
  filter(family>0)
# Visualization 1 (Family and Happiness Score Scatterplot) for Research Question 2
ggplot(Happy_vis, aes(x = happiness_score, y = family)) +
  geom_point(size = 3, color = "purple")+
  labs(x= "Happiness Score (0-10)", y = "Family Impact (0 -2)", title = "Impact of Famil
y Influence on Happiness Score")
```



This scatterplot demonstrates the relationship between the variables 'happiness_score' and 'family' impact and reveals the correlation between the variables. Based on the visualization, the more a family contributes to the happiness score, the higher the happiness score will be. This positive relationship shows that family plays a vital role in determining happiness score.

```
#Visualization 2 (Social Support and Happiness Score Box Plot) for Research Question 2
Happy_mut %>%
  mutate(Happiness_Level = case_when(happiness_score_mut == 1 ~ 'happy',
                             happiness_score_mut == 0 ~ 'unhappy')) %>%
ggplot(aes(x = Happiness_Level, y = social_support, fill = Happiness_Level)) +
geom_boxplot()+
labs(x= "Happiness Level", y = "Socail Support Contribution", title = "Happiness Level A
cross Social Support")
```



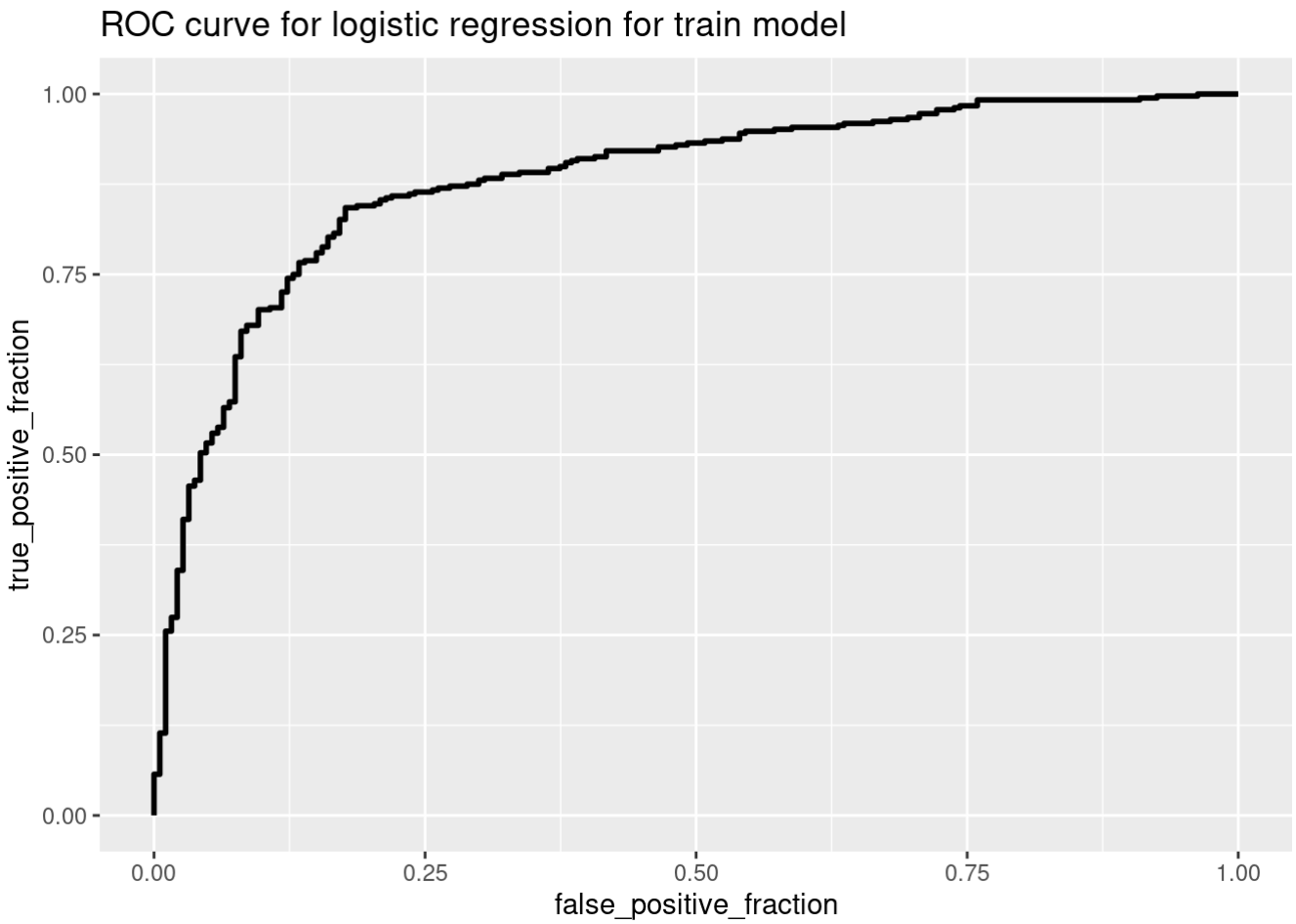
This boxplot demonstrates the relationship between the variables 'Happiness Level' and 'social_support' and reveals the correlation between the variables. Based on the visualization, there is significant overlap between the two Happiness Levels when placed across social support. Low social support can lead to either Happy or Unhappy levels, but high social support (>1) is connected with only the Happy level. ## Prediction Model (Logistic Regression) and Cross Validation

```
sample_process <- sample(c(TRUE, FALSE), # take value TRUE or FALSE
                 nrow(Happy_mut), # for each row in biopsy
                 replace = TRUE, # TRUE or FALSE can repeat
                 prob = c(0.7, 0.3)) # 70% TRUE, 30% FALSE
# Select values for the train set (corresponding to TRUEs in sample_process)
train_Happy2 <- Happy_mut[sample_process, ]
# Select values for the test set (corresponding to FALSEs in sample_process)
test_Happy2 <- Happy_mut[!sample_process, ]
# Fitting a Logistic Regression Model
fit_log <- glm(happiness_score_mut ~ family + social_support, data = Happy_mut)
summary(fit_log)
```

```
##
## Call:
## glm(formula = happiness_score_mut ~ family + social_support,
##     data = Happy_mut)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.10705  -0.26998   0.08715   0.25884   1.13159
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.39441    0.05122   -7.70 4.06e-14 ***
## family          1.00566    0.04958   20.28  < 2e-16 ***
## social_support  0.88724    0.04277   20.75  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1445272)
##
##     Null deviance: 179.21  on 791  degrees of freedom
## Residual deviance: 114.03  on 789  degrees of freedom
## AIC: 720.64
##
## Number of Fisher Scoring iterations: 2
```

```
# Results in a data frame for train data
predict_train2 <- data.frame(
  predictions = predict(fit_log, newdata = train_Happy2, type = "response"),
  outcome = train_Happy2$happiness_score_mut,
  name = "train")
# Results in a data frame for test data
predict_test2 <- data.frame(
  predictions = predict(fit_log, newdata = test_Happy2, type = "response"),
  outcome = test_Happy2$happiness_score_mut,
  name = "test")
#ROC plot for Logistic Regression for the train model
ROC2_train <- ggplot(predict_train2) +
geom_roc(aes(d = outcome, m = predictions), n.cuts = 0) +
labs(title = "ROC curve for logistic regression for train model")
ROC2_train
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



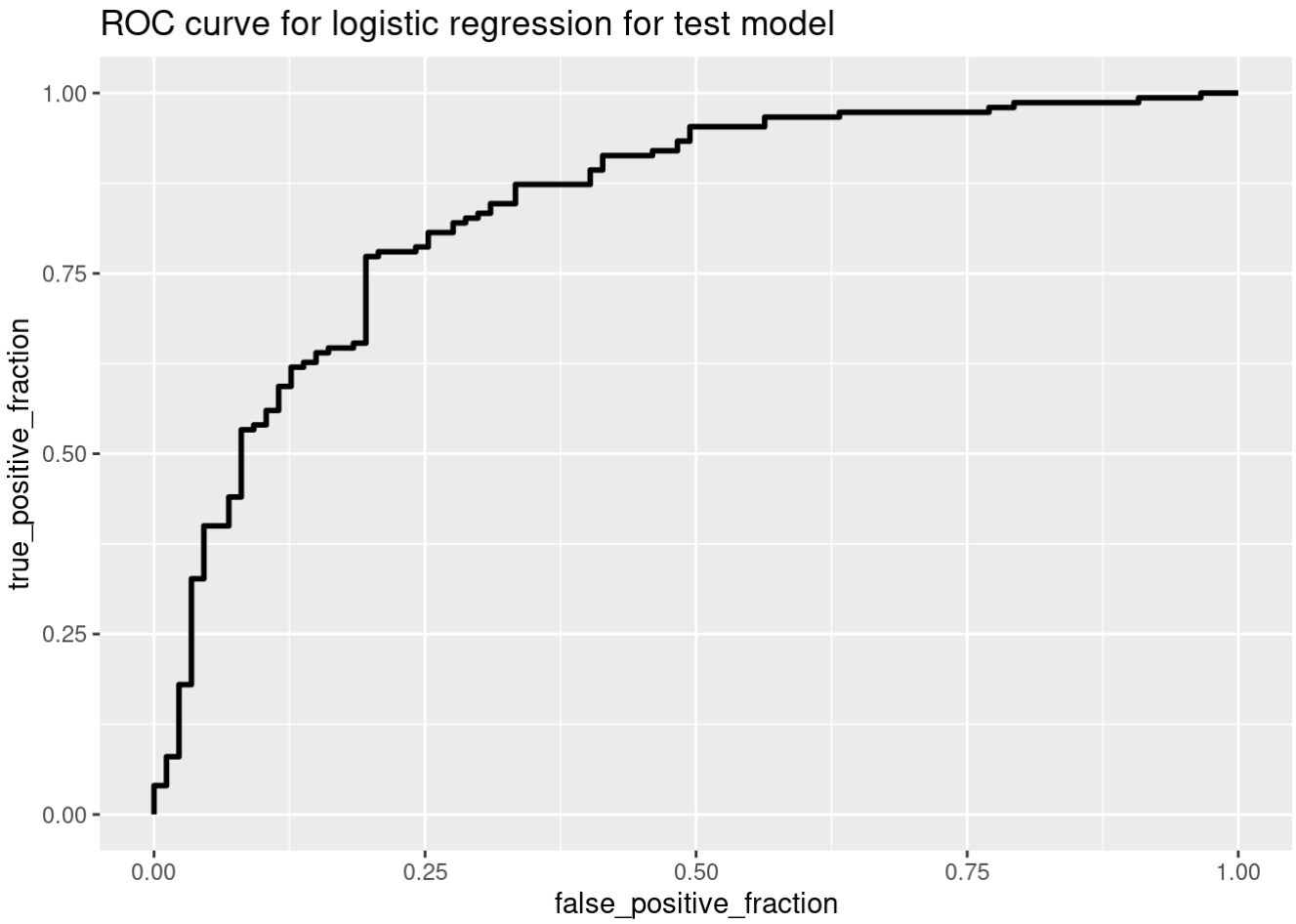

```
#Calculating the AUC value for ROC2 for the train model
calc_auc(ROC2_train)
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
##   PANEL group       AUC
## 1     1    -1 0.8802023
```

```
#ROC plot for Logistic Regression for the test model
ROC2_test <- ggplot(predict_test2) +
geom_roc(aes(d = outcome, m = predictions), n.cuts = 0) +
labs(title = "ROC curve for logistic regression for test model")
ROC2_test
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
#Calculating the AUC value for ROC2 for the train model
calc_auc(ROC2_test)
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
##   PANEL group       AUC
## 1     1    -1 0.8409962
```

```
#Cross Validation for the Logistic Regression Model
# Choose number of folds
k = 10
# Randomly order rows in the dataset
data <- Happy_mut[sample(nrow(Happy_mut)), ]
# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)
# Initialize a vector to keep track of the performance
perf_k <- NULL
# Use a for loop to get diagnostics for each test set
for(i in 1:k){
  # Create train and test sets
  train_not_i <- data[folds != i, ] # all observations except in fold i
  test_i <- data[folds == i, ]  # observations in fold i
  # Train model on train set (all but fold i)
  happy_log <- glm(happiness_score_mut ~ family + social_support, data = train_not_i)
  # Test model on test set (fold i)
  predict_i <- data.frame(
    predictions = predict(happy_log, newdata = test_i, type = "response"),
    outcome = test_i$happiness_score_mut)
  # Consider the ROC curve for the test dataset
  ROC2_cv <- ggplot(predict_i) +
    geom_roc(aes(d = outcome, m = predictions))
  ROC2_cv
  # Get diagnostics for fold i (AUC)
  perf_k[i] <- calc_auc(ROC2_cv)$AUC
}
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: d, m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
# Average performance
mean(perf_k)
```

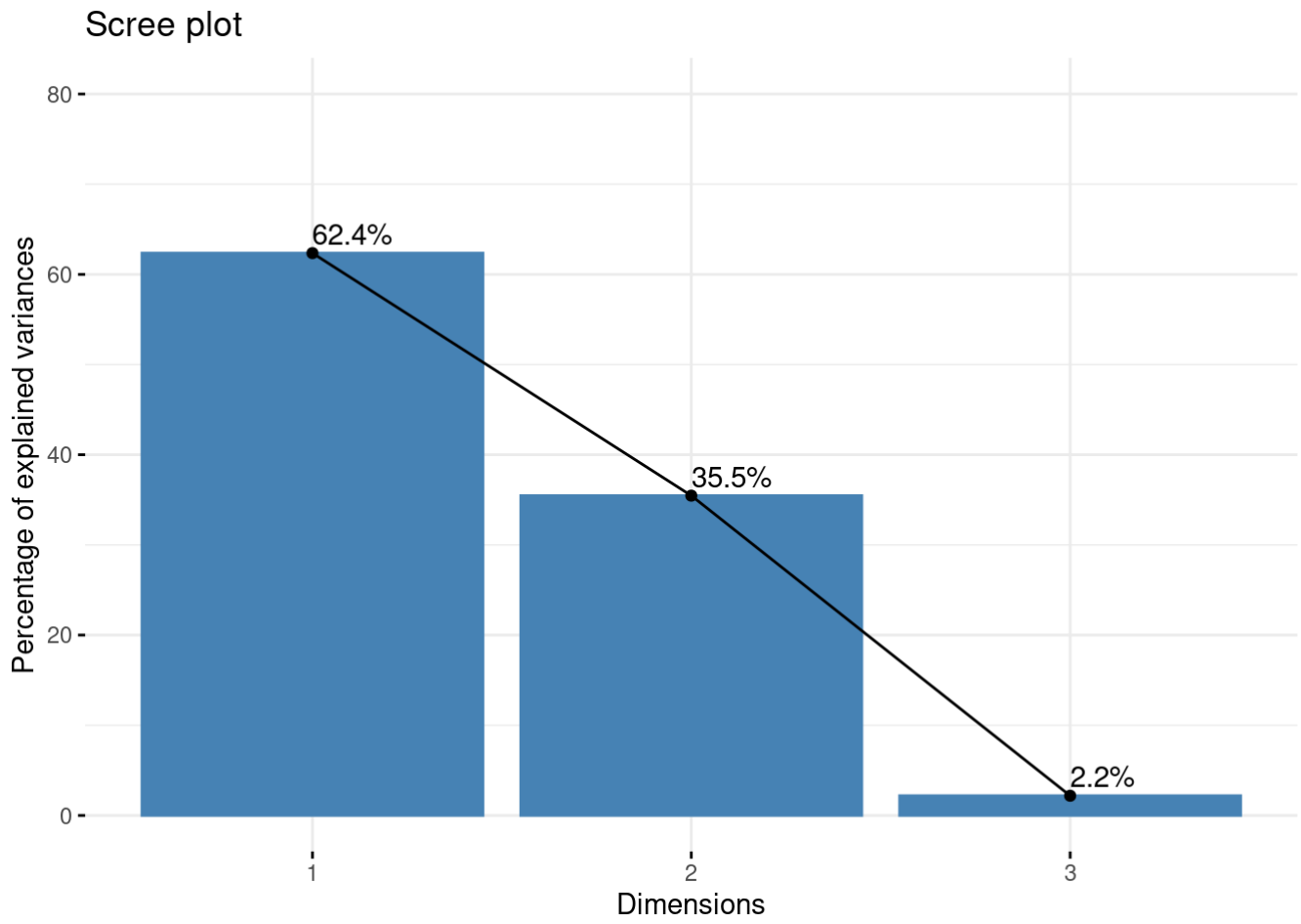
```
## [1] 0.865718
```

The prediction model for the Logistic Regression model for the research question "How does family and social support impact the happiness score globally?" was accomplished and ROC curves were constructed for both the train data and the test data. The train data for the Logistic Regression's ROC curve has an AUC value of 0.8670103 while the test data for the Logistic Regression's ROC curve has an AUC value of 0.8681171. This demonstrates that the Logistic Regression model for addressing the research question "How does family and social support impact the happiness score globally?" has a predictability accuracy of around 86.7% for the train data and around 86.8% for the test data which demonstrates that the model is functioning accurately since the AUC score ranges from 0-100%. Additionally, since both the test and train data have similar AUC scores, overfitting is likely not an issue. A 10-fold cross validation was then executed which presented a mean AUC value of 0.8696584 demonstrating that the Logistic Regression model is still performing very accurately on "new" data. ## PCA for Research Question 2

```
#Applying PCA to the Desired Variables
pca2 <- Happy_mut %>%
  select(happiness_score, family, social_support) %>%
  scale %>%
  prcomp
pca2
```

```
## Standard deviations (1, .., p=3):
## [1] 1.367748 1.031414 0.255836
##
## Rotation (n x k) = (3 x 3):
##                          PC1        PC2        PC3
## happiness_score  0.03185882 -0.9665640  0.2544387
## family          -0.70355978 -0.2025022 -0.6811729
## social_support   0.70992158 -0.1573115 -0.6864870
```

```
# Creating a scree plot
fviz_eig(pca2, addlabels = TRUE, ylim = c(0, 80))
```




```
# Visualize the contributions of the variables to the PC1 in a table
get_pca_var(pca2)$coord %>% as.data.frame %>%
  arrange(Dim.1) %>% select(Dim.1)
```

```
##                      Dim.1
## family          -0.96229233
## happiness_score  0.04357483
## social_support   0.97099366
```

```
# Visualize the contributions of the variables to the PC2 in a table
get_pca_var(pca2)$coord %>% as.data.frame %>%
  arrange(Dim.2) %>% select(Dim.2)
```

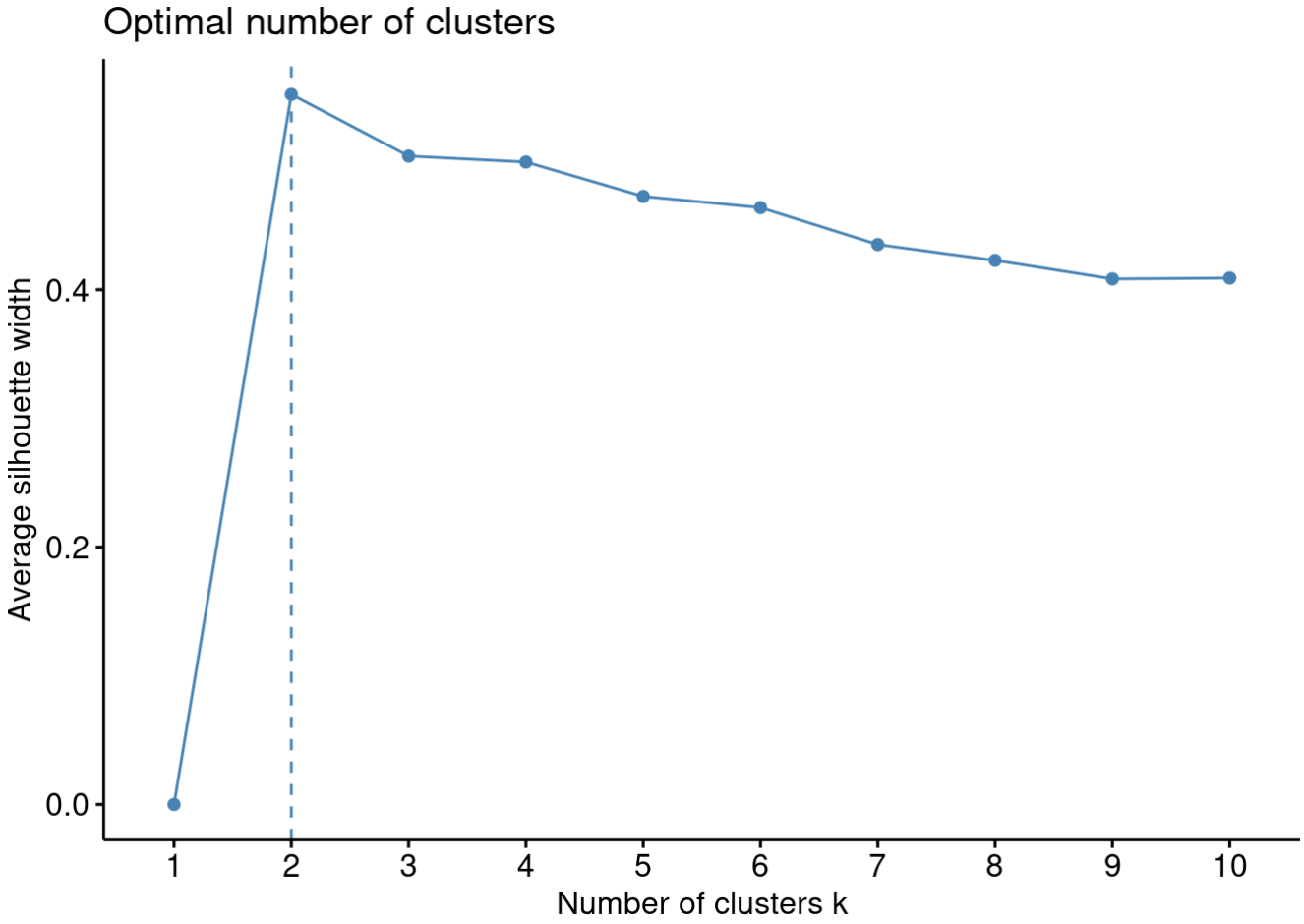
```
##                      Dim.2
## happiness_score -0.9969272
## family          -0.2088635
## social_support  -0.1622532
```

The scree plot demonstrates that the percentage of explained variance for the first principal component is 62.4% while the percentage of explained variance for the second principal component is 35.5% ## Kmeans Clustering for Research Question 2

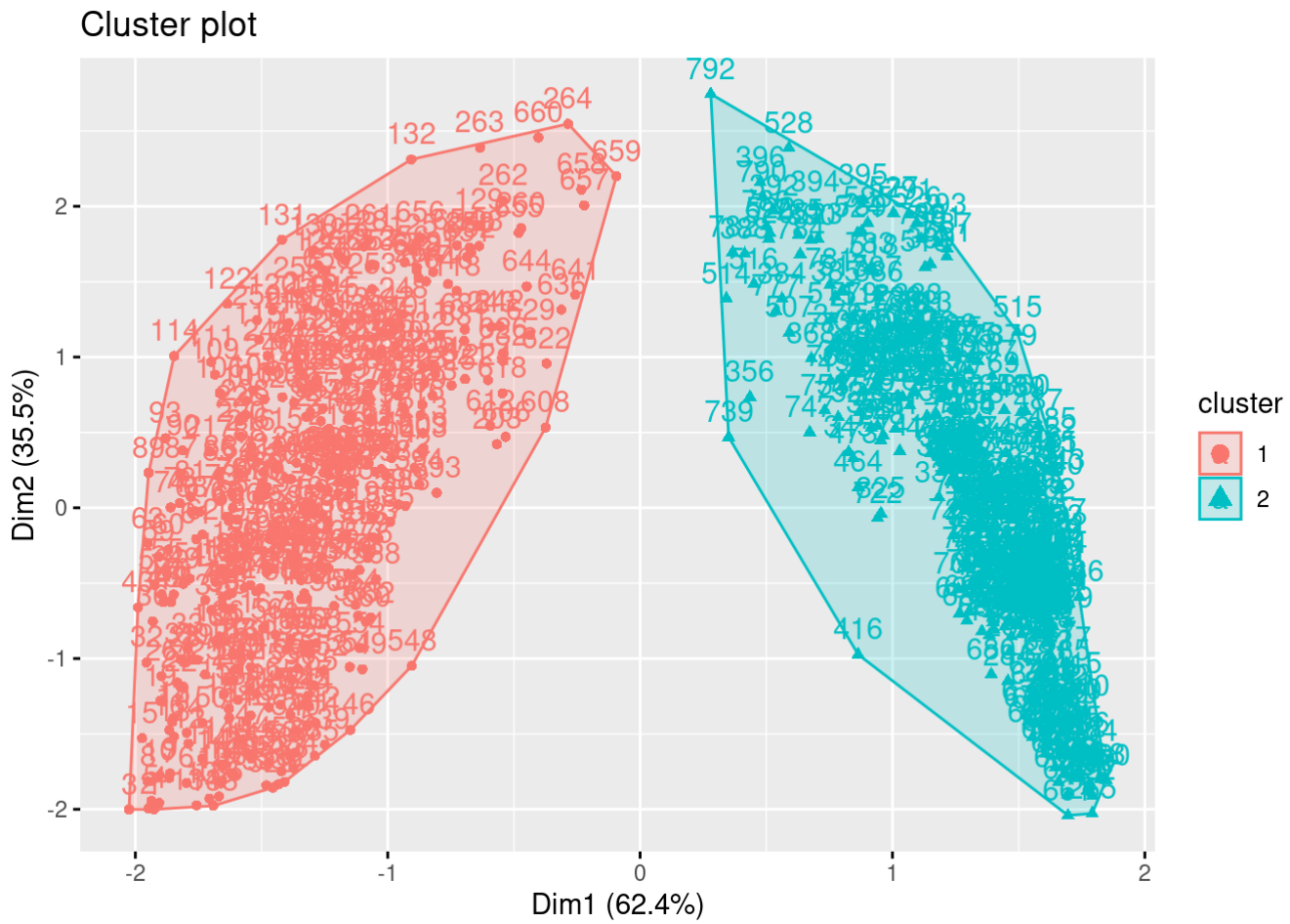
```
# Scaling the Variables
Happy2_scaled <- Happy_mut %>%
  select(happiness_score, family, social_support) %>%
  scale
# Use the function kmeans() to find clusters
kmeans_results <- Happy2_scaled %>%
  kmeans(centers = 2)
kmeans_results
```

```
## K-means clustering with 2 clusters of sizes 396, 396
##
## Cluster means:
##   happiness_score     family social_support
## 1     -0.03304441  0.9136332     -0.9509136
## 2      0.03304441 -0.9136332      0.9509136
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [260] 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [334] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [371] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [408] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [445] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [482] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [519] 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [556] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [593] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [630] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
## [667] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [704] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [741] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [778] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 538.2869 456.5902
##  (between_SS / total_SS =  58.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# Maximize the silhouette while keeping a small number of clusters
fviz_nbclust(Happy2_scaled, kmeans, method = "silhouette")
```




```
#Visualizing the Clusters
fviz_cluster(kmeans_results, data = Happy2_scaled)
```



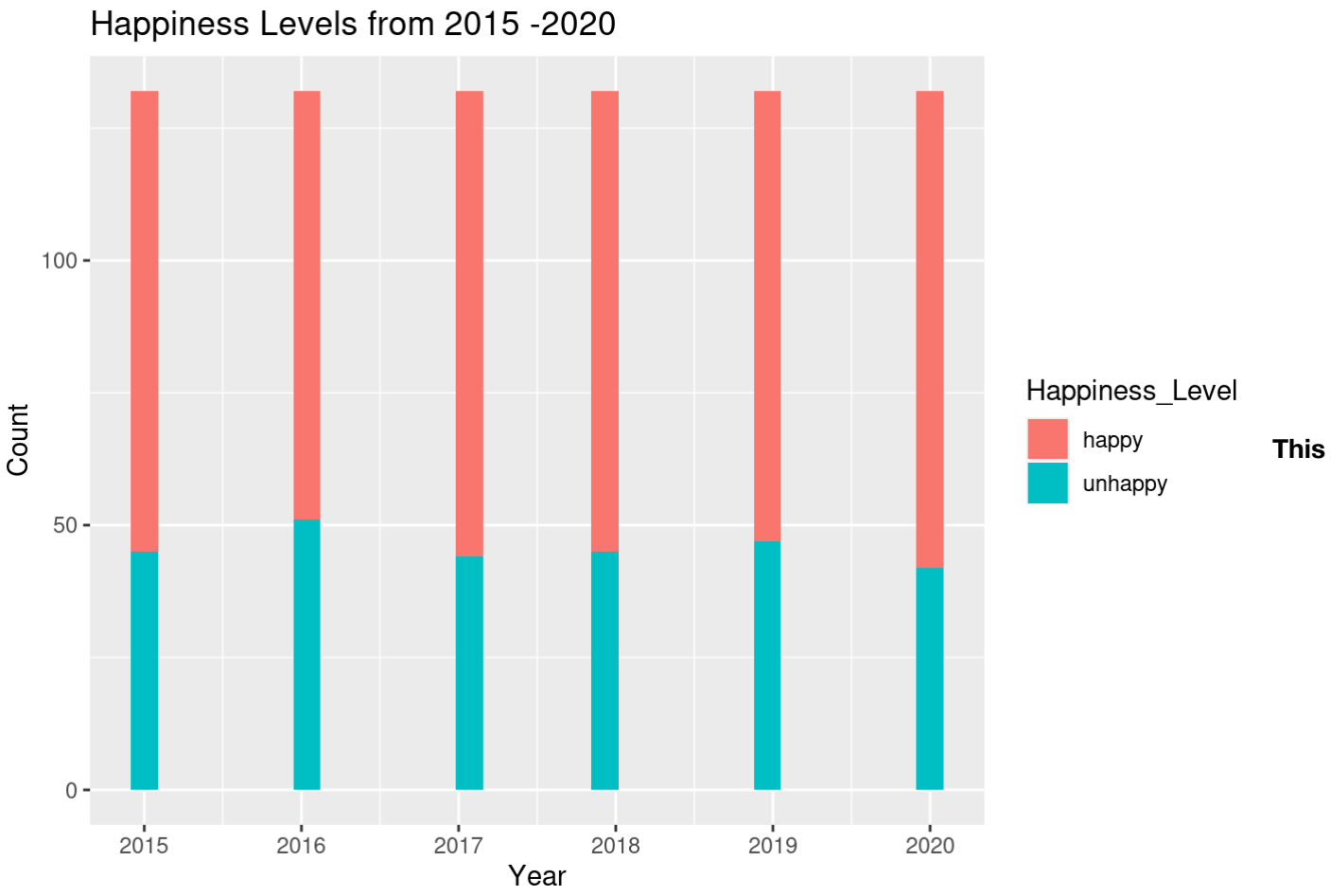

```
# Summary statistics
Happy_mut %>%
  select(happiness_score, family, social_support) %>%
  mutate(cluster = as.factor(kmeans_results$cluster)) %>%
  group_by(cluster) %>%
  summarize_if(is.numeric, mean, na.rm = T)
```

```
## # A tibble: 2 × 4
##   cluster happiness_score family social_support
##   <fct>             <dbl>  <dbl>          <dbl>
## 1 1                  5.44   1.01           0
## 2 2                  5.51   0              1.22
```

The average silhouette width indicates that 2 clusters maximize the average width silhouette. Additionally, through visualizing the clusters, it is evident that the two clusters and distinguishable from one another. Summary statistics were run on the clusters to determine the differences in the means for the selected variables. This demonstrated that cluster 1 has a higher average "happiness_score" and "social_support" than cluster 2 with average values of 5.510476 and 1.218604 respectively. Cluster 2 has a higher average "family" score than cluster 1 with an average value of 1.009995. This is indicative of cluster 1 representing the observations with higher social support and slightly higher happiness scores on average while cluster 2 represents the observations with higher family scores on average. ## 5.) Research Question 3: "How does family and year impact happiness score globally?" ## Visualizations

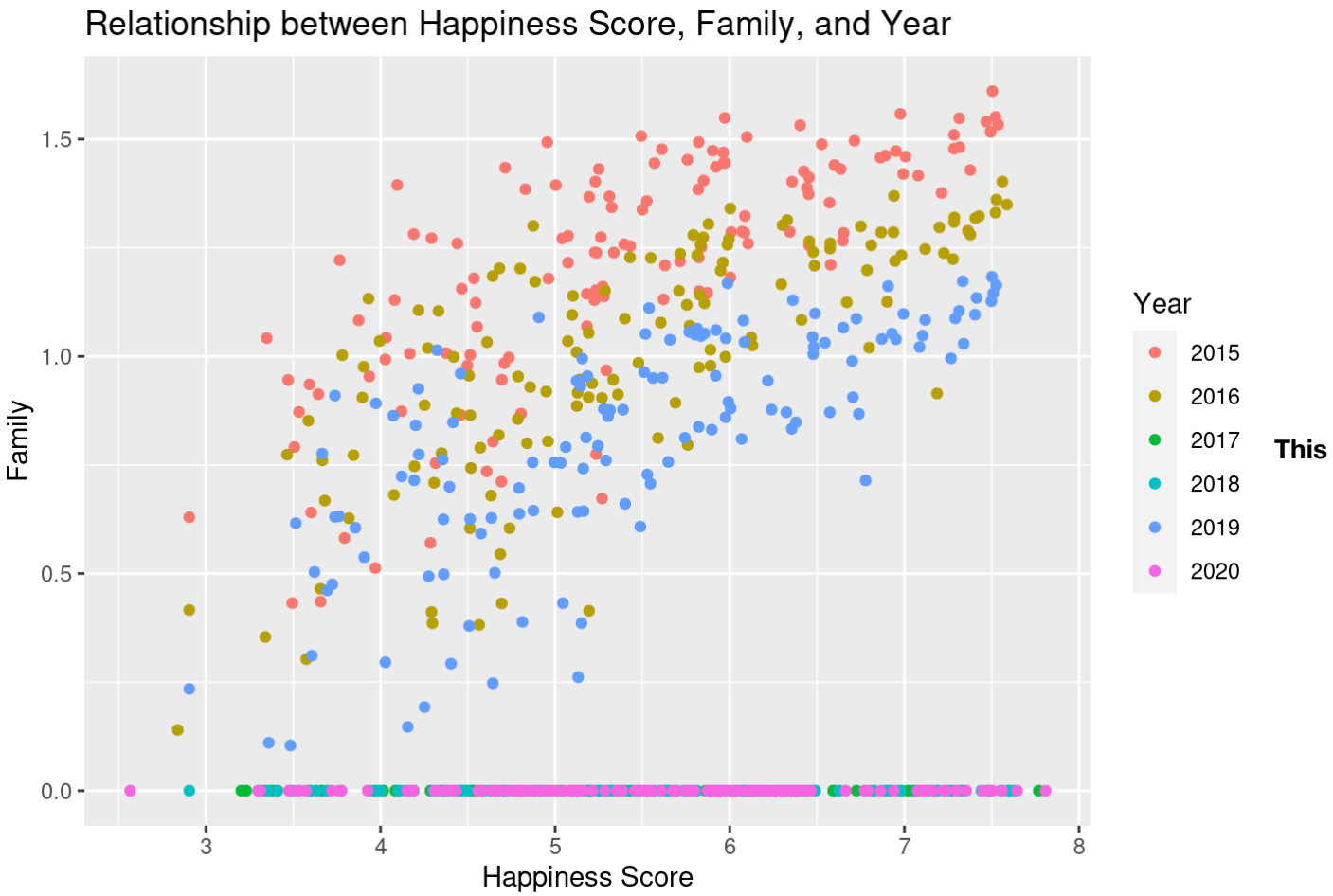
```
#Visualization 1 (Year, and Happiness Score) for Research Question 3
Happy_mut %>%
  mutate(Happiness_Level = case_when(happiness_score_mut == 1 ~ 'happy',
                             happiness_score_mut == 0 ~ 'unhappy')) %>%
ggplot(aes(x= Year, fill = Happiness_Level)) +
  geom_histogram()+
  labs(x = 'Year', y= 'Count', title= "Happiness Levels from 2015 -2020")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



This stacked histogram shows the relationship between year and happiness level. Based on the visualization, the year has little to no impact on the distribution of Happy or Unhappy levels. From 2015 - 2020, the ratio of unhappy to happy counts remains fairly constant, indicating the year has little effect on Happiness Level.

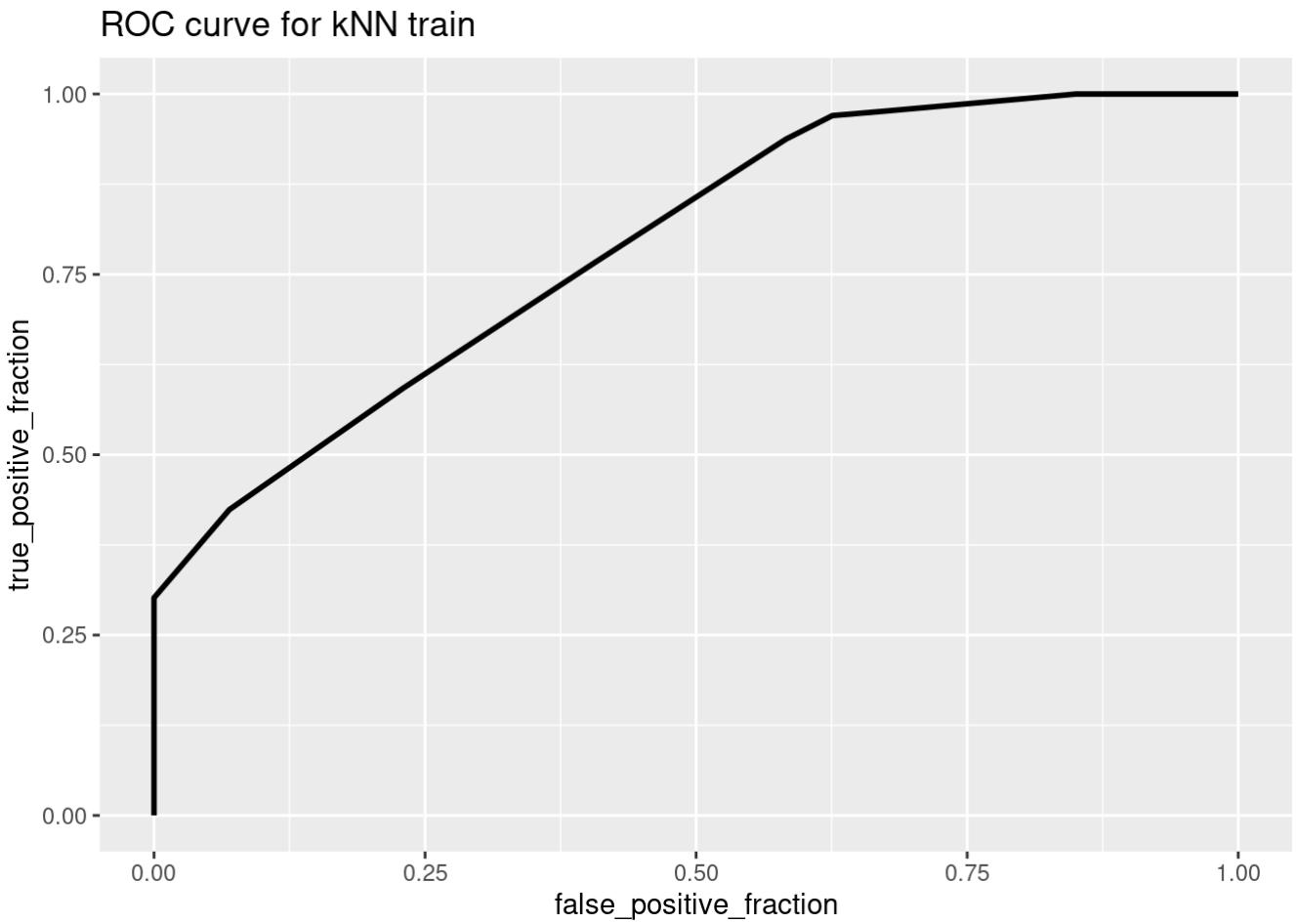
```
#Visualization 2 for Research Question 3
Happy%>%
  ggplot(aes(x=happiness_score, y= family, color=as.factor(Year)))+geom_point()+
  labs(title = "Relationship between Happiness Score, Family, and Year", x= "Happiness S
core", y= "Family", color= "Year")
```



scatter plot shows the relationship between happiness score, family, and year and reveals how the variables are correlated. Based on the visualization, family and happiness score seemed to both be the highest in 2015 and both variables have decreased since 2015. In addition, the relationship between family and happiness score seemed to be positively correlated. ## Prediction Model (kNN) and Cross Validation

```
# Define a sampling process
sample_process <- sample(c(TRUE, FALSE), # take value TRUE or FALSE
                 nrow(Happy_mut), # for each row in biopsy
                 replace = TRUE, # TRUE or FALSE can repeat
                 prob = c(0.7, 0.3)) # 70% TRUE, 30% FALSE
# Select values for the train set (corresponding to TRUEs in sample_process)
train_happy <- Happy_mut[sample_process, ]
# Select values for the test set (corresponding to FALSEs in sample_process)
test_happy <- Happy_mut[!sample_process, ]
# Fit the model
happy_kNN <- knn3(happiness_score_mut ~ family + Year,
                data = train_happy,
                k = 5)
# Results in a data frame for train data
predict_train3 <- data.frame(
  predictions = predict(happy_kNN, train_happy)[,2],
  outcome = train_happy$happiness_score_mut,
  name = "train")
# Results in a data frame for test data
predict_test3 <- data.frame(
  predictions = predict(happy_kNN, test_happy)[,2],
  outcome = test_happy$happiness_score_mut,
  name = "test")
# ROC curve train
ROC_Knn_train <- ggplot(predict_train3) +
  geom_roc(aes(d = outcome, m = predictions), n.cuts = 0) +
  labs(title = "ROC curve for kNN train")
ROC_Knn_train
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



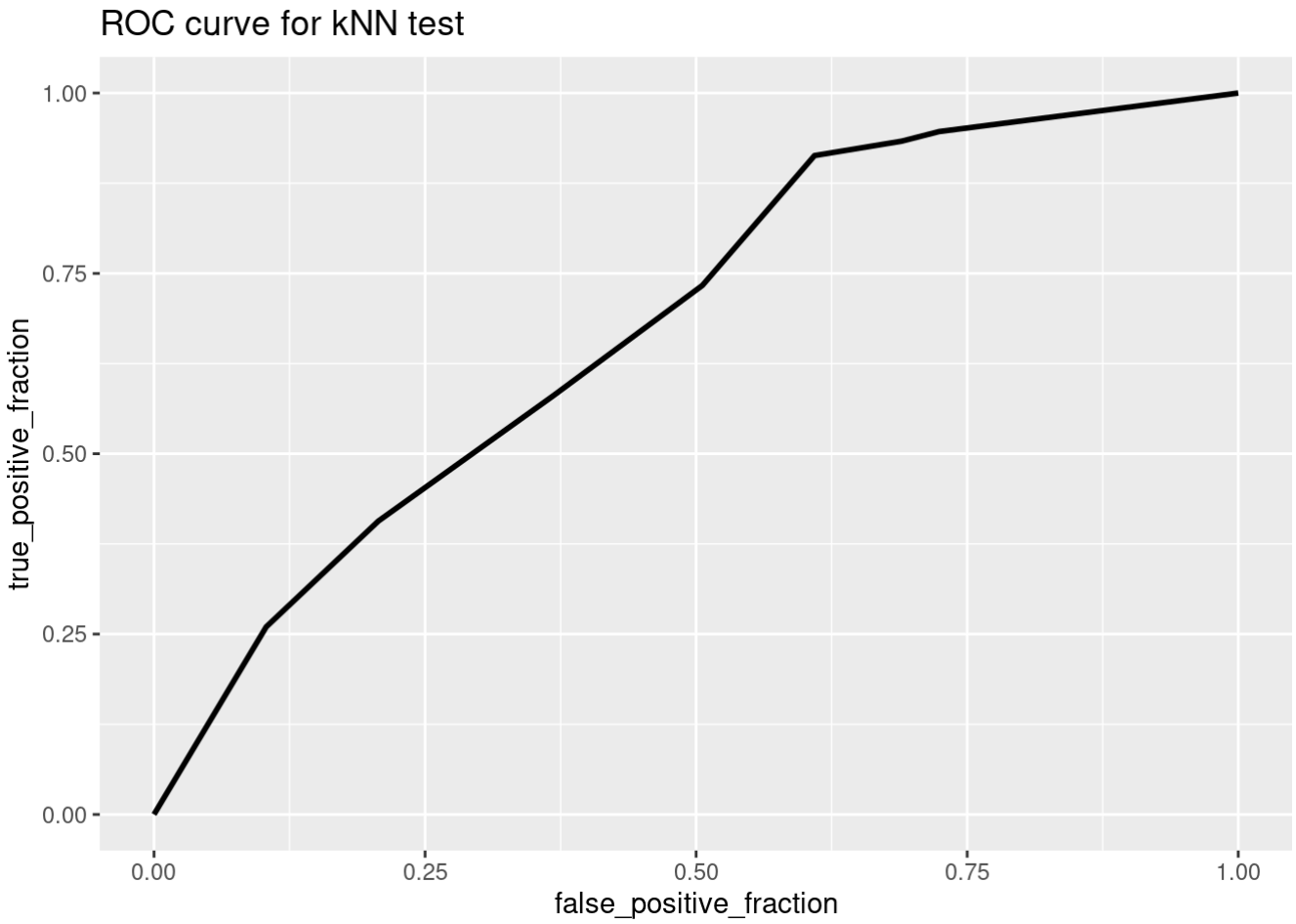

```
#Calculating the AUC value for ROC for the train model
calc_auc(ROC_Knn_train)
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
##   PANEL group       AUC
## 1     1    -1 0.7887119
```

```
# ROC curve test
ROC_Knn_test <- ggplot(predict_test3) +
  geom_roc(aes(d = outcome, m = predictions), n.cuts = 0) +
  labs(title = "ROC curve for kNN test")
ROC_Knn_test
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
#Calculating the AUC value for ROC for the test model
calc_auc(ROC_Knn_test)
```

```
## Warning: The following aesthetics were dropped during statistical transformation: d,
m
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
##   PANEL group       AUC
## 1     1    -1 0.6782759
```

```
# Cross Validation for kNN
kNN_cv <- train(happiness_score_mut ~ family + Year,
      data = Happy_mut,
      method = "knn",
      trControl = trainControl(method = "cv", number = 10)) # 10-fold cv
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
# Let's take a look at the output
kNN_cv
```

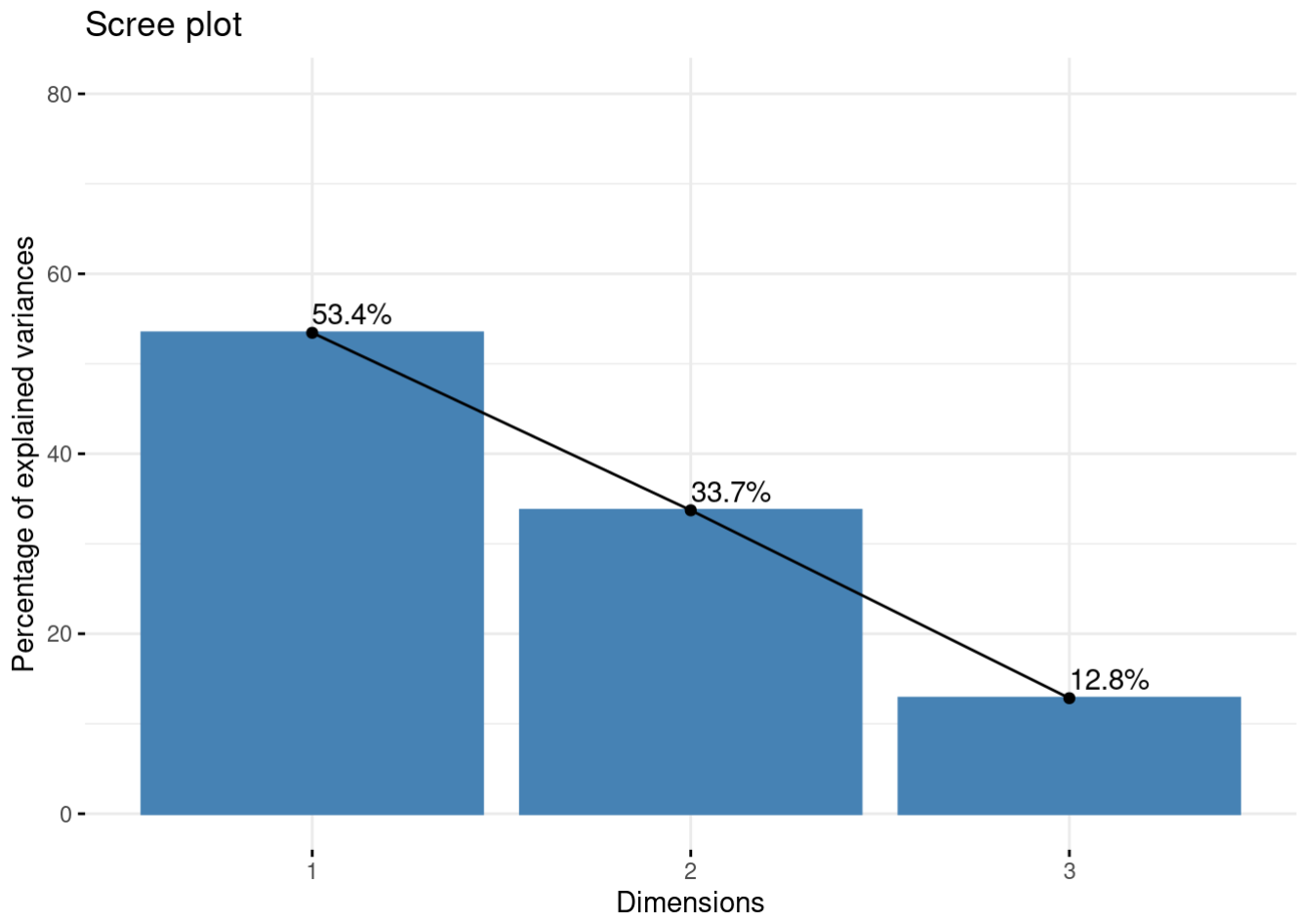
```
## k-Nearest Neighbors
##
## 792 samples
##   2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 712, 713, 713, 713, 713, 713, ...
## Resampling results across tuning parameters:
##
##   k  RMSE       Rsquared   MAE
##   5  0.4368790  0.1756295  0.3537763
##   7  0.4342777  0.1785981  0.3577889
##   9  0.4332327  0.1792744  0.3594468
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```

The prediction model for the kNN model for the research question "How does family and year impact happiness score globally?" was accomplished and ROC curves were constructed for both the train data and the test data. The train data for the kNN's ROC curve has an AUC value of 0.7776359 while the test data for the kNN's ROC curve has an AUC value of 0.7214414. This demonstrates that the kNN model for addressing the research question "How does family and year impact happiness score globally?" has a predictability accuracy of around 77.8% for the train data and around 72.1% for the test data which demonstrates that the model is functioning accurately but not as accurately as the Decision Tree's accuracy for Research Question 1 or the Logistic Regression's for Research Question 2 since the AUC score ranges from 0-100%. There could have been small effects of overfitting as the accuracy went down by about 5% from the train data to the test data. A 10-fold cross validation was then executed for the kNN model which demonstrated that the optimal model would be for data utilzing 9 nearest neighbors as data using 9 neighbors resulted in the smallest RMSE value of 0.4291881. The smaller the RMSE value, the smaller the difference is between the predicted values and the actual values. An RMSE value of 0.4291881 shows that the kNN model for 9 nearest neighbors functions not highly accurately. ## PCA for Research Question 3

```
#Applying PCA to the Desired Variables
pca3 <- Happy_mut %>%
  select(happiness_score, family, Year) %>%
  scale %>%
  prcomp
pca3
```

```
## Standard deviations (1, .., p=3):
## [1] 1.2662052 1.0057608 0.6206205
##
## Rotation (n x k) = (3 x 3):
##                          PC1         PC2        PC3
## happiness_score -0.1556227 -0.96667303 -0.2032851
## family          -0.7100438 -0.03360254  0.7033552
## Year             0.6867454 -0.25379939  0.6811509
```

```
# Creating a scree plot
fviz_eig(pca3, addlabels = TRUE, ylim = c(0, 80))
```




```
# Visualize the contributions of the variables to the PC1 in a table
get_pca_var(pca3)$coord %>% as.data.frame %>%
  arrange(Dim.1) %>% select(Dim.1)
```

```
##                       Dim.1
## family          -0.8990612
## happiness_score -0.1970503
## Year             0.8695606
```

```
# Visualize the contributions of the variables to the PC2 in a table
get_pca_var(pca3)$coord %>% as.data.frame %>%
  arrange(Dim.2) %>% select(Dim.2)
```

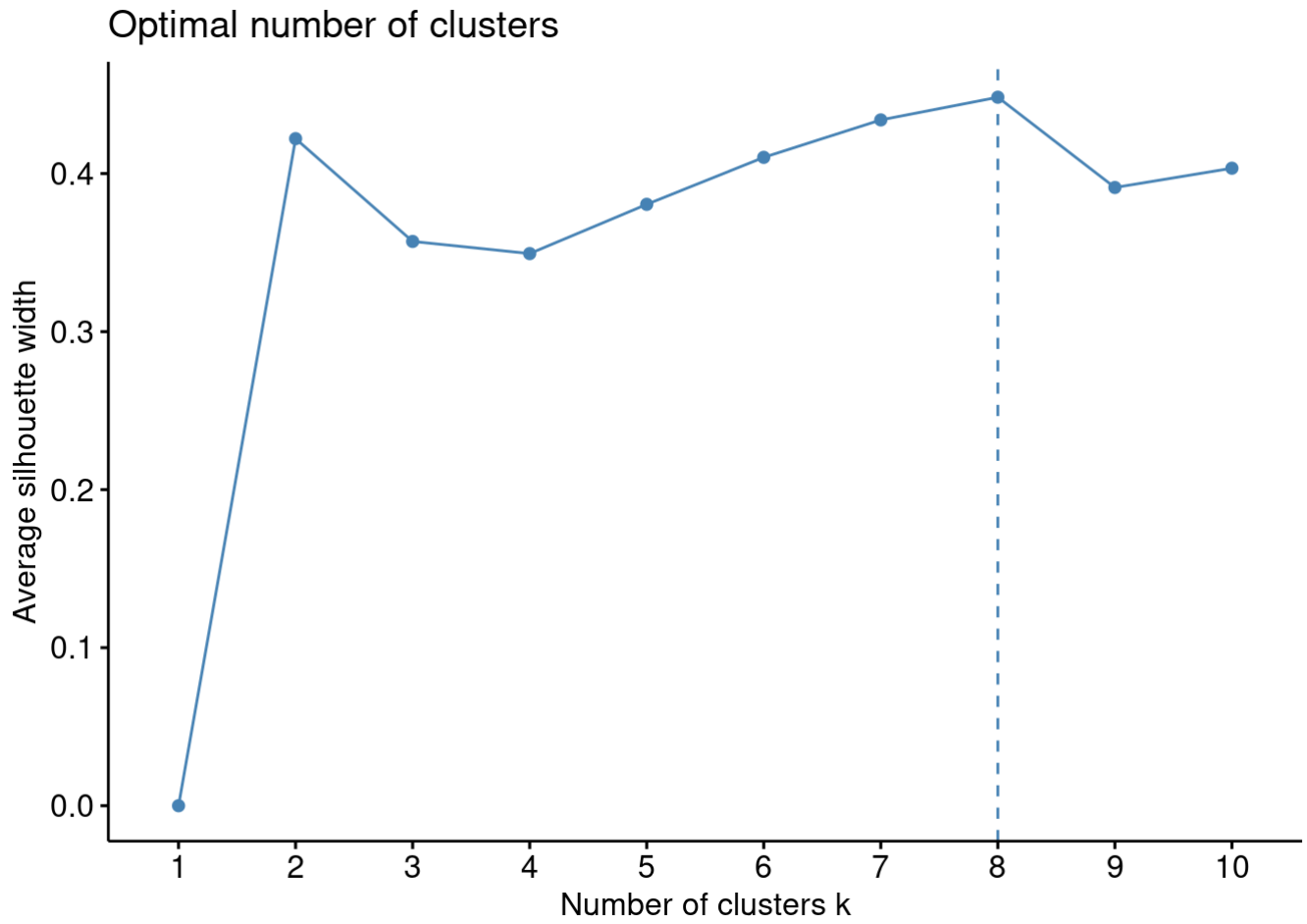
```
##                        Dim.2
## happiness_score -0.97224179
## Year            -0.25526146
## family          -0.03379612
```

The scree plot demonstrates that the percentage of explained variance for the first principal component is 53.4% while the percentage of explained variance for the second principal component is 33.7% ## Kmeans Clustering for Research Question 3

```
# Scaling the Variables
Happy3_scaled <- Happy_mut %>%
  select(happiness_score, family, Year) %>%
  scale
# Use the function kmeans() to find clusters
kmeans_results <- Happy3_scaled %>%
  kmeans(centers = 2)
kmeans_results
```

```
## K-means clustering with 2 clusters of sizes 263, 529
##
## Cluster means:
##   happiness_score     family       Year
## 1     -0.02183714  1.1019997 -1.1714530
## 2      0.01085665 -0.5478751  0.5824048
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [260] 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [334] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [371] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [408] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [445] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [482] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [519] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [556] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [593] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [630] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [667] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [704] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [741] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [778] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 356.6325 997.6533
##  (between_SS / total_SS =  42.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# Maximize the silhouette while keeping a small number of clusters
fviz_nbclust(Happy3_scaled, kmeans, method = "silhouette")
```




```
fviz_cluster(kmeans_results, data = Happy3_scaled)
```


```
# Summary statistics
Happy_mut %>%
  select(happiness_score, family, Year) %>%
  mutate(cluster = as.factor(kmeans_results$cluster)) %>%
  group_by(cluster) %>%
  summarize_if(is.numeric, mean, na.rm = T)
```

```
## # A tibble: 2 × 4
##   cluster happiness_score family  Year
##   <fct>             <dbl>  <dbl> <dbl>
## 1 1                  5.45  1.11  2015.
## 2 2                  5.49  0.202 2018.
```

After examining the "Optimal numer of clusters" visualization we will be using 2 clusters as that is where the average silhouette width changes. Additionally, through visualizing the clusters, it is evident that the two clusters are distinguishable from one another. Summary statistics were run on the clusters to determine the differences in the means for the selected variables. This demonstrated that cluster 1 has a higher average "family" score than cluster 2 with average value of 1.1141120. Cluster 2 has a higher average "happiness_score" and "Year" than cluster 1 with average values of 5.485521 and 2018.495 respectively. This is indicative of cluster 1 representing the observations with higher family scores on average while cluster 2 represents slightly higher Years and happiness scores on average. ## 6.) Discussion
We utilized the dataset "Happiness and Corruption Globally" dataset for this project. 'happiness_score' was utilized as the outcome variable and 'freedom', 'Year', 'family', 'health', and 'social_support' as the predictor variables to assess their relationships to one another through three primary research questions.

The data was already tidy as each variable has its own respective column and each observation has its own row so no tidying was required prior to assessing the research questions. A correlation matrix was executed in order to determine which of these variables are most highly correlated. Through the correlation matrix, it was determined that 'family' and 'social_support' and 'happiness_score' and 'health' had the highest correlation to each other with a correlation of 0.87 and 0.75 respectively. The variables with the lowest correlation were 'family' and 'freedom' and 'Year' and 'happiness_score' with a correlation of around 0.014 and 0.023 respectively. The first research question was "How does freedom and health impact the happiness score globally?". The first visualization for Research Question 1 was a scatterplot and discovered a higher happiness score is associated with higher values of freedom and health. The second visualization for Research Question 1 was a boxplot and revealed the Happy Level is associated with higher health values. The visualizations matched our predictions for Research Question 1 that family and social support have a positive correlation with happiness score. The prediction model utilized to assess Research Question 1 was a Decision Tree model to make predictions and categorize the variables of interest using both train and test data. The train data for the Decision Tree's ROC curve has an AUC value of 0.8550482 while the test data for the Decision Tree's ROC curve has an AUC value of 0.8389272 showing that the Decision Tree is highly accurate in its predictability and likely does not have issues with overfitting since the AUC values for both the train and test data are very similar to each other. A 10-fold cross validation was then run for the Decision Tree model which showed that the optimal model is for the smallest cp value of 0.03060988.The RMSE value for that decision tree is 0.3527780 which demonstrates that the model is functioning accurately because a low RMSE value is good. A PCA was then done to visualize the variance for the principal components and examine how highly each variable contributes to such components. The variance for the first principal component is 71.1% while the percentage of explained variance for the second principal component is 21.5% both visualized through a scree plot. Kmeans clustering was then done and it was determined that 2 clusters would be optimal. Cluster 1 represented the observations with higher average scores for all of the selected variables for Research Question 1 while cluster 2 represented the observations with lower average scores for all of the selected variables for Research Question 1. The second research question was "How does family and social support impact the happiness score?". The first visualization for Research Question 2 was a scatterplot which revealed happiness scores to be positively correlated family impact. The second visualization for Research Question 2 was a boxplot and discovered that the Happy level was associated with higher values of social support and Unhappy level was associated with lower values of social support. The visualizations matched our predictions for Research Question 2 that that freedom and health both have a positive correlation with happiness score. The prediction model utilized to assess Research Question 2 was a Logistic Regression model to make predictions about the data for the variables involved with the research question utilizing both train and test data. The train data for the Logistic Regression's ROC curve has an AUC value of 0.8670103 while the test data for the Logistic Regression's ROC curve has an AUC value of 0.8681171. AUC values range from 0-1 representing accuracy in predictability ranging from 0% to 100%. Considering both sets of data produced AUC values of about 87%, the model is highly accurate in making predictions. Additionally, because of the extremely minimal difference between the two AUC values, overfitting is not likely an issue for the model. A 10-fold cross validation was then run for the Logistic Regression model presented a mean AUC value of 0.8696584 demonstrating that the Logistic Regression model is still performing very accurately on "new" data. A PCA was then accomplished and the variance for the first principal component is 62.4% while the percentage of explained variance for the second principal component is 35.5% both visualized through a scree plot. The Kmeans cluster analysis for Research Question 2 showed that 2 clusters would be optimal. Cluster 1 represents the observations with higher social support and slightly higher happiness scores on average while cluster 2 represents the observations with higher family scores on average. The third research question was "How does family and year impact happiness score globally?". The first visualization for Research Question 3 was a histogram that revealed the year to have no impact on the distribution of Happy or Unhappy levels and the ratio of

happy to unhappy counts were overall fairly constant across the years. The second visualization for Research Question 3 was a scatter plot that the year 2015 had the highest happiness and family scores and family and happiness score were positively correlated. Therefore, our visualizations matched the expected trend of happiness score and family increasing together and being directly correlated. However, happiness and family scores were the highest in the first year in 2015 and seemed to decrease in later years, which is not the trend we expected. For Research Question 3, the k-Nearest Neighbors model was used for prediction of happiness score. For the train and test dataset, an ROC curve was created and the AUC was calculated to be 0.7776359 and 0.7214414, respectively. Therefore, the accuracy for the train and test data was 77.8% and 72.1%, respectively. These values mean the model's accuracy at predicting happiness score is decent but not as highly accurate as the Decision Tree Model and Logistic Regression Model in terms of predictability. Then, a 10-fold cross validation was completed for the kNN model and determined that 9 nearest neighbors would be optimal due to it having the least amount of error with the smallest RMSE value of 0.3783371. Therefore, the 10-fold cross validation model for the kNN model is quite accurate. Next, a Principal Component Analysis was completed and a scree plot was constructed. The scree plot revealed the percentage of variance explained by the first principal component and second principal component to be 53.4% and 33.7%, respectively. Then, K-means clustering was completed and it was discovered that 2 clusters were optimal. Cluster 1 had a higher average family score of 1.1141120 compared to cluster 2. Cluster 2 had a higher average happiness score and year of 5.485521 and 2018.495, respectively, which is higher than cluster 1. Therefore, cluster 1 represents observations with higher family scores on average, while cluster 2 represents observations with later years and higher happiness scores on average. The most challenging aspect of the project was doing the prediction models. Our data kept producing errors when trying to compute the Decision Tree, Logistic Regression, and kNN models. However, these models were integral in determining the predictability and accuracy of our models for assessing the variables in the research questions. ## 7.) Formatting *Acknowledgements: Thanks to Kaggle for providing the datasets. Bella Crain was responsible for Research Question 1 that addressed the question "How does freedom and health impact the happiness score globally?". She constructed visualizations to demonstrate the relationships between such variables, created a prediction model, cross validation, PCA, and performed clustering. Avery Zuckerman was responsible for Research Question 2 that analyzed the question "How does family and social support impact the happiness score?". She constructed visualizations to demonstrate the relationships between such variables, created a prediction model, cross validation, PCA, and performed clustering. Kaitlyn Rouse addressed the question "How does family and year impact happiness score globally?". She constructed visualizations to demonstrate the relationships between such variables, created a prediction model, cross validation, PCA, and performed clustering. Additionally, Avery Zuckerman, Bella Crain, and Kaitlyn Rouse collaborated together equally on the "Introduction", "Creating a Correlation Matrix", and the "Discussion" sections of the report.*