Code ▾

Project 2: Predicting if donor will give blood in March 2007

Sarah Lee cl45274

1. Introduction

a. Set Up

Hide

```
# Load packages
library(tidyverse)
library(plotROC)
library(caret)
library(rpart)
library(rpart.plot)
library(ggcorrplot)
library(factoextra)
```

b. Quick description of the dataset(s)

https://code.datasciencedojo.com/datasciencedojo/datasets/tree/master/Blood%20Transfusion%20Service%20Center (https://code.datasciencedojo.com/datasciencedojo/datasets/tree/master/Blood%20Transfusion%20Service%20Center)

‘mydata’ comes from the link I haved linked above. This data set consists of 748 rows and 5 columns. While searching through new data sets for this project I thought it would be interesting to research about blood donation as I have gained interest in donating blood after I have turned legal. The five columns display recency, frequency, monetary, time, and whether one donated blood in March 2007, respectively. I plan to choose recency (the number of months since his/her most recent donation), frequency (total number of donations that he/she has made), and monetary (total amount of blood that he/she has donated in cc, cubic cetimeters) variables as my three predictor variables and whether on donated blood in March 2007 as my outcome variable. I predict that recency, frequency, and monetary will provide enough information to predict if a donor will or will not donate blood in March 2007.The data is from a donor database, Blood Transfusion Service Center in Hsin-Chu City, Taiwan.

Hide

```
# Import datasets
library(readr)
mydata <- read_csv("transfusion.data.csv")
```

```
Rows: 748 Columns: 5── Column specification ──────────────────────────────────────────────────────────────────────────────────────────────
Delimiter: ","
dbl (5): Recency, Frequency, Monetary, Time, outcome
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.
```

Hide

```
#take a look
head(mydata)
```

Recency <dbl>	Frequency <dbl>	Monetary <dbl>	Time <dbl>	outcome <dbl>
2	50	12500	98	1
0	13	3250	28	1
1	16	4000	35	1
2	20	5000	45	1
1	24	6000	77	0
4	4	1000	4	0

6 rows

c. Define a research question

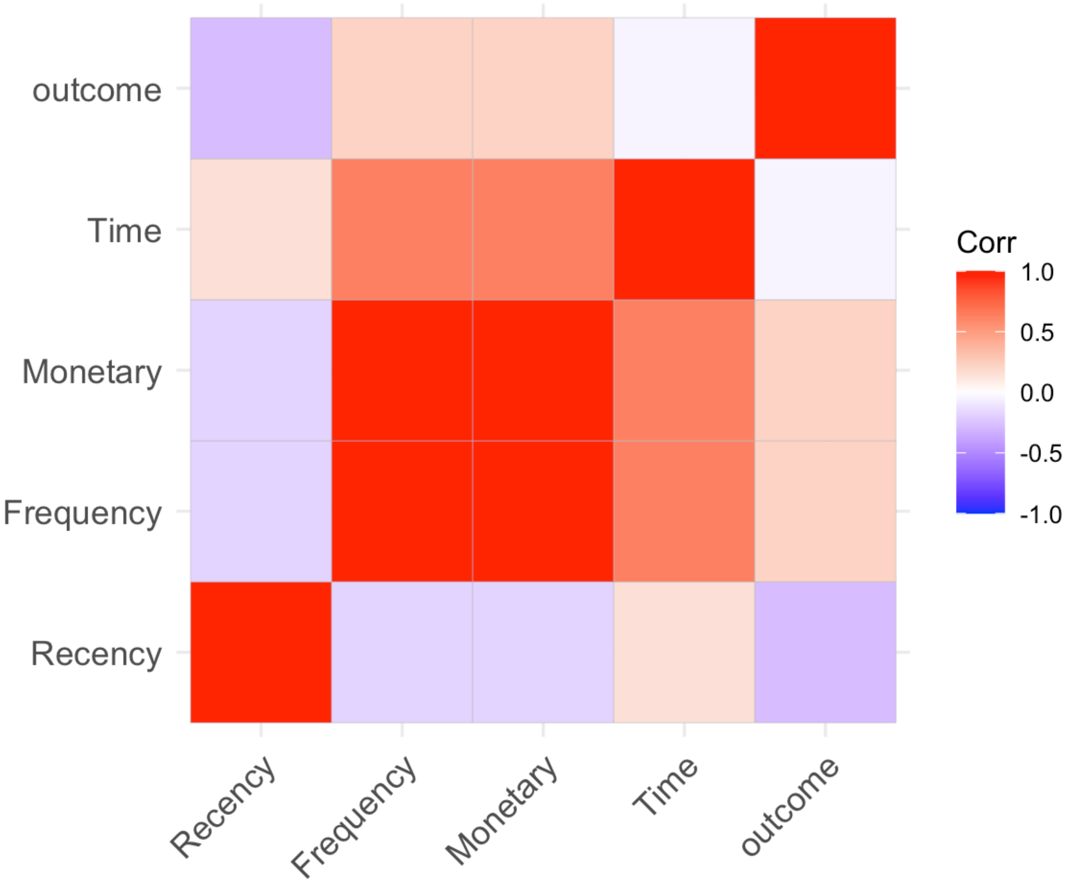
Can Recency, Frequency, and Monetary predict if a donor will donate blood in March of 2007?

2. Exploratory Data Analysis

Create a correlation matrix:

Hide

```
## Use the ggcorrplot to visualize the correlation matrix
ggcorrplot(cor(mydata))
```

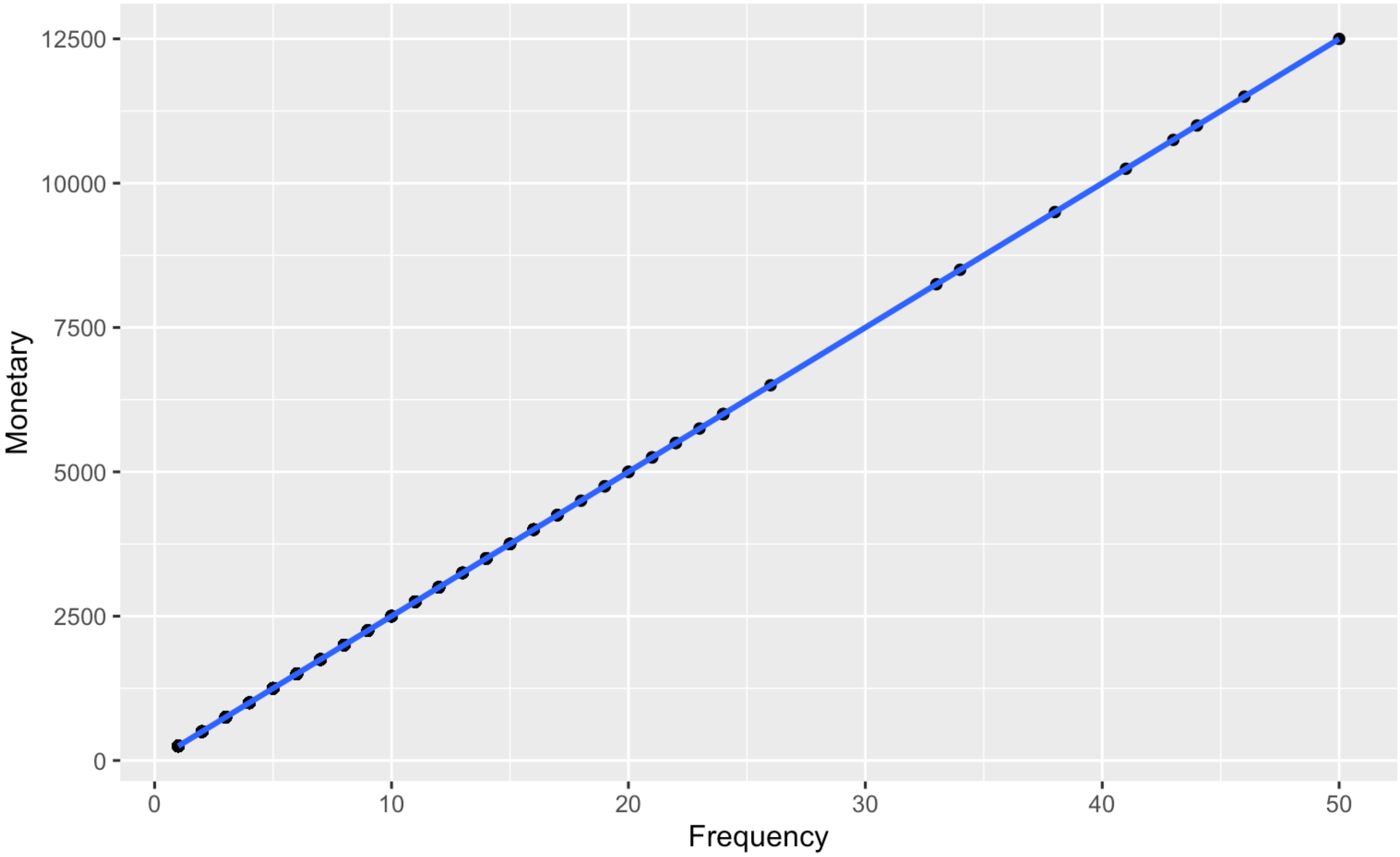


According to the correlation matrix above, ‘Monetary’ and ‘Frequency’ have a strong positive corelation while ‘whether he/she donated blood’ and ‘Time’ has no correlation as its boxes are while with a value of 0 as their correlation coefficient. Overall, the other variables tend to have relatively weak correlations between each other.

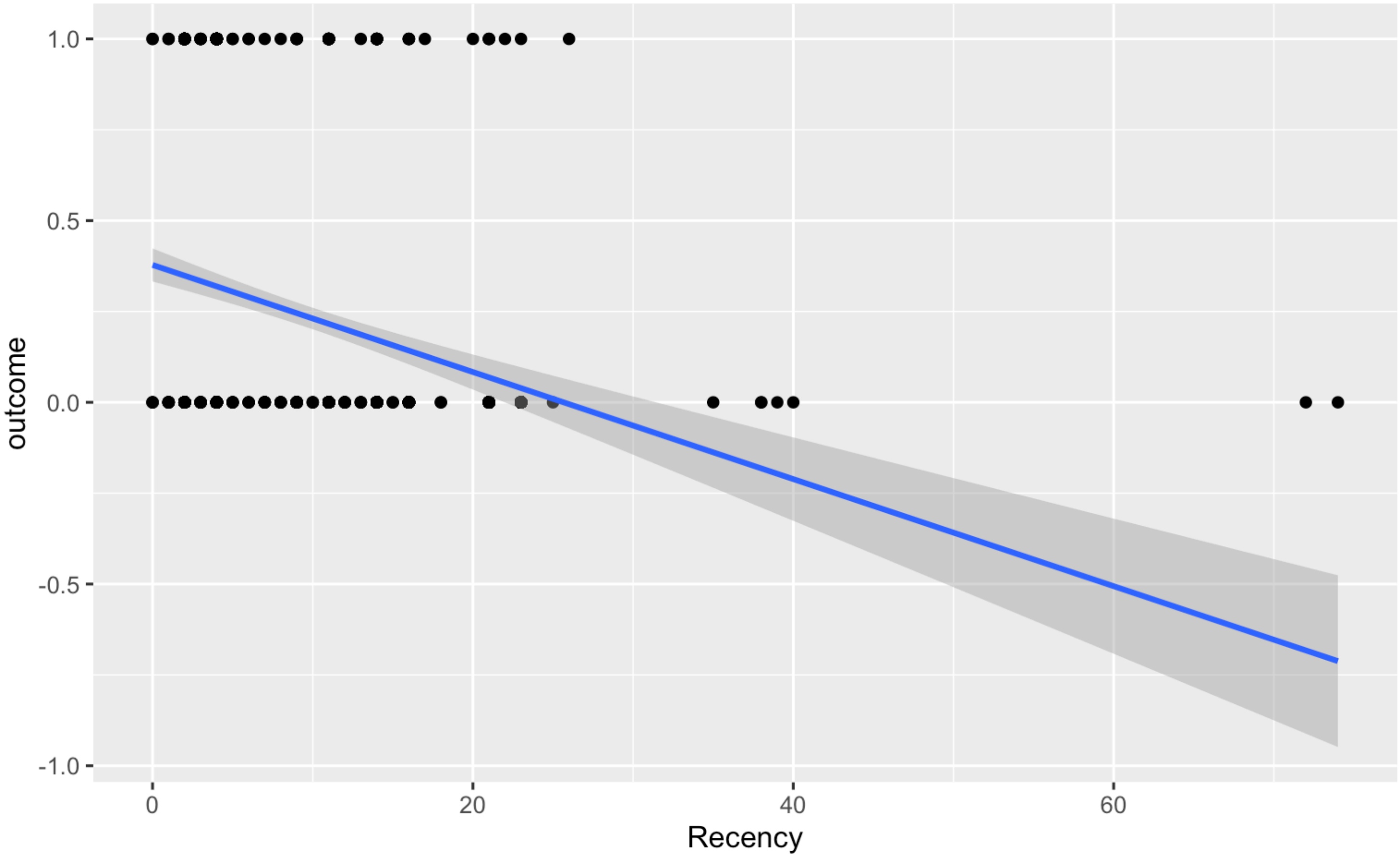
Create visualizations to investigate relationships:

Hide

```
## Use ggplot()
ggplot(mydata, aes(x = Frequency, y = Monetary)) +
  geom_point() +
  geom_smooth(method = "lm")
```



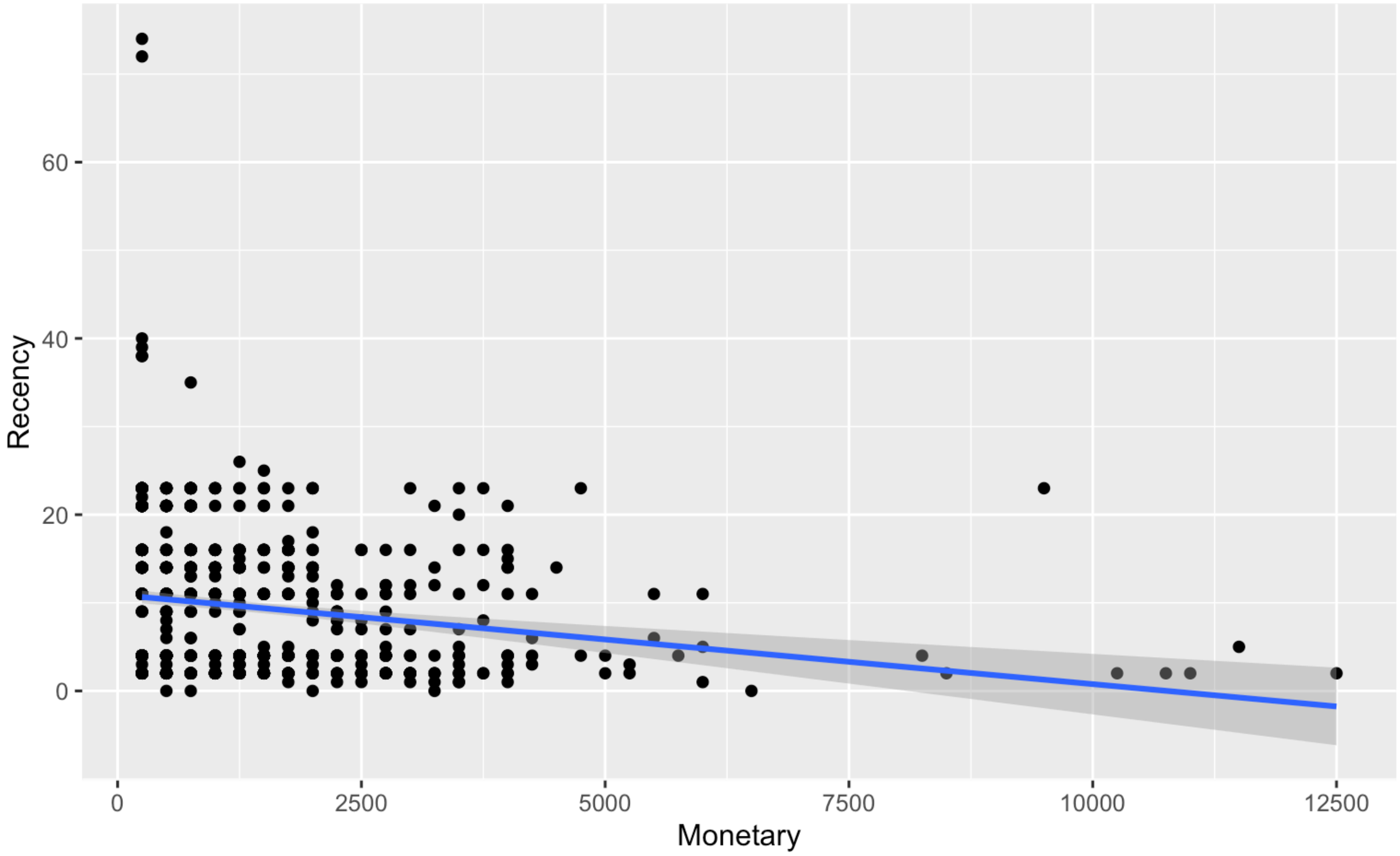
This graph above shows the relationship between ‘Frequency’ and ‘Monetary’ which appeared to have the strongest correlation in the correlation matrix earlier. According to the graph, we are able to conclude that these two variables are actually strongly correlated.



This second graph explores the relationship between ‘outcome’ and ‘Recency’. Through this graph above we are able to conclude that these two variables have a weak negative correlations as shown in the line with negative, relatively flat slope line.

Hide

```
ggplot(mydata, aes(x = Monetary, y = Recency)) +
  geom_point() +
  geom_smooth(method = "lm")
```



This last graph displays the relationship between ‘Recency’ and ‘Monetary’. Based on the correlation matrix earlier, these two variables were said to be less correlated and we are able to see that in this graph as the slope of the graph is relatively flat. It shows a weak negative relationship between ‘Recency’ and ‘Monetary’.

3. Prediction and Cross-Validation

a. Train the model

Hide

```
# k-nearest-neighbor:

my_knn=knn3(outcome ~ Recency + Monetary + Frequency, data = mydata, k = 5)

ROC <- ggplot(mydata) +
  geom_roc(aes(d = outcome, m = predict(my_knn, mydata)[,2]), n.cuts=0)

calc_auc(ROC)
```

```
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 fa
ctor?
```

	PANEL <fctr>	group <int>	AUC <dbl>
1	1	-1	0.7819338

1 row

The AUC value is .7819338 which means the performance is relatively ok as it is not as close to 1.

b. Perform cross-validation

Hide

```
# 10 fold cross validation
k = 10

knn_cv=train(as.factor(outcome) ~ Recency + Monetary + Frequency,
             data= mydata,
             method="knn",
             trControl=trainControl(method="cv",number=10))
knn_cv
```

```
k-Nearest Neighbors

748 samples
  3 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 673, 673, 673, 674, 673, 673, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.7608108  0.2146082
  7  0.7527027  0.1889482
  9  0.7460180  0.1327803

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

The average performance was around .76 which is similar to the previous AUC value which means that this classifier is working pretty well. Since the performance of the training data is consistent with the performance of cross validation, there is no signs of overfitting.

4 & 5. Dimensionality reduction and Clustering

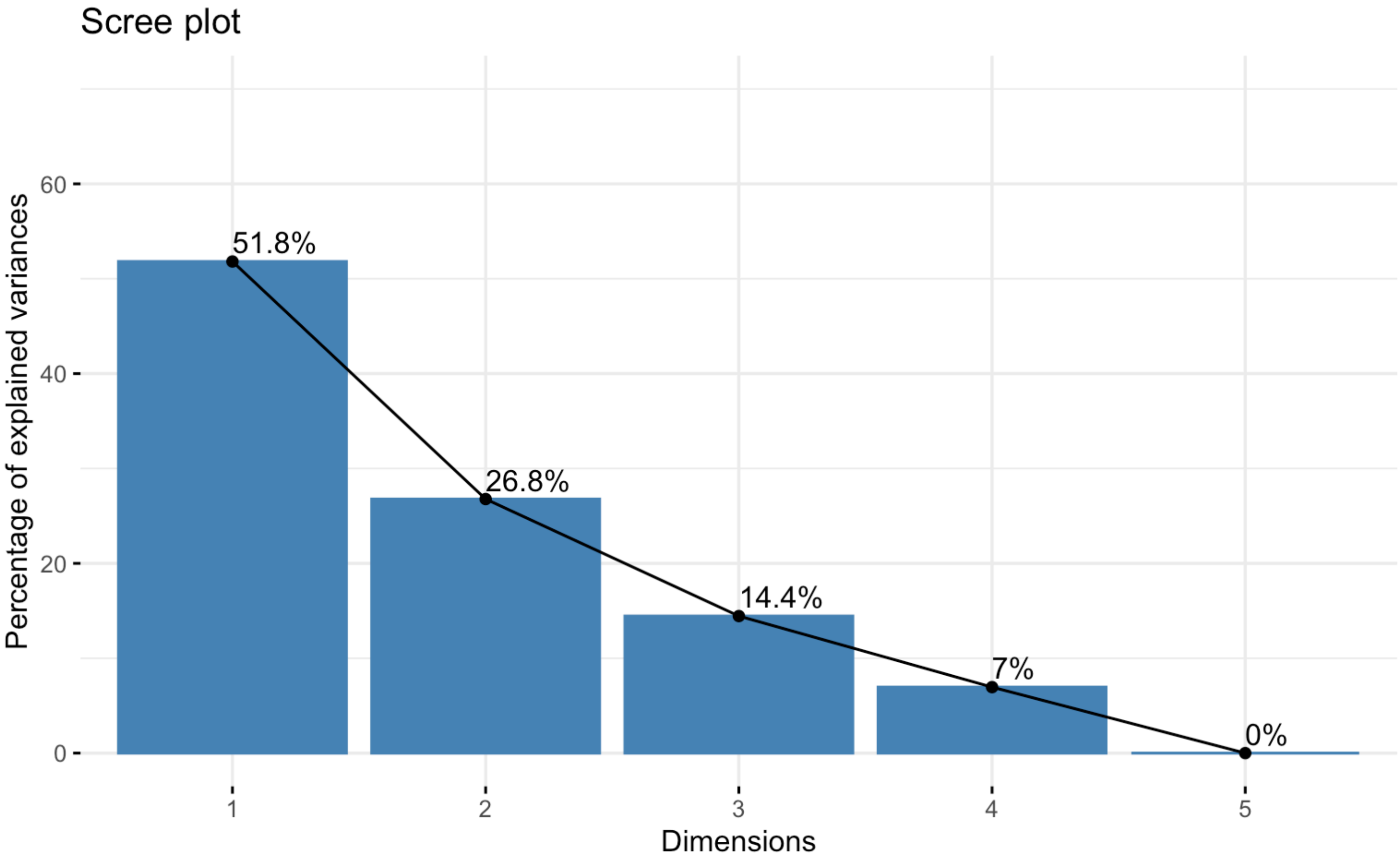
Hide

```
# Perform PCA with prcomp()
mydata_scaled <- mydata %>%
  select_if(is.numeric)%>%
  na.omit%>%
  # Scale the variables
  scale %>%
  # Save as a data frame
  as.data.frame

pca <- mydata_scaled %>%
  prcomp
names(pca)
```

```
[1] "sdev"     "rotation" "center"   "scale"    "x"
```

Hide

```
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 70))
```



Since we usually keep the first 80% of the variance with the first few pcs, the first two components will add up to around 78.6%.

Hide

```
#interpreting the pcs

get_pca_var(pca)$coord %>% as.data.frame %>%
  arrange(Dim.1) %>% select(Dim.1) # for PC1
```

	Dim.1 <dbl>
Frequency	-0.9756257
Monetary	-0.9756257
Time	-0.7522732
outcome	-0.2861281
Recency	0.1985256

5 rows

Hide

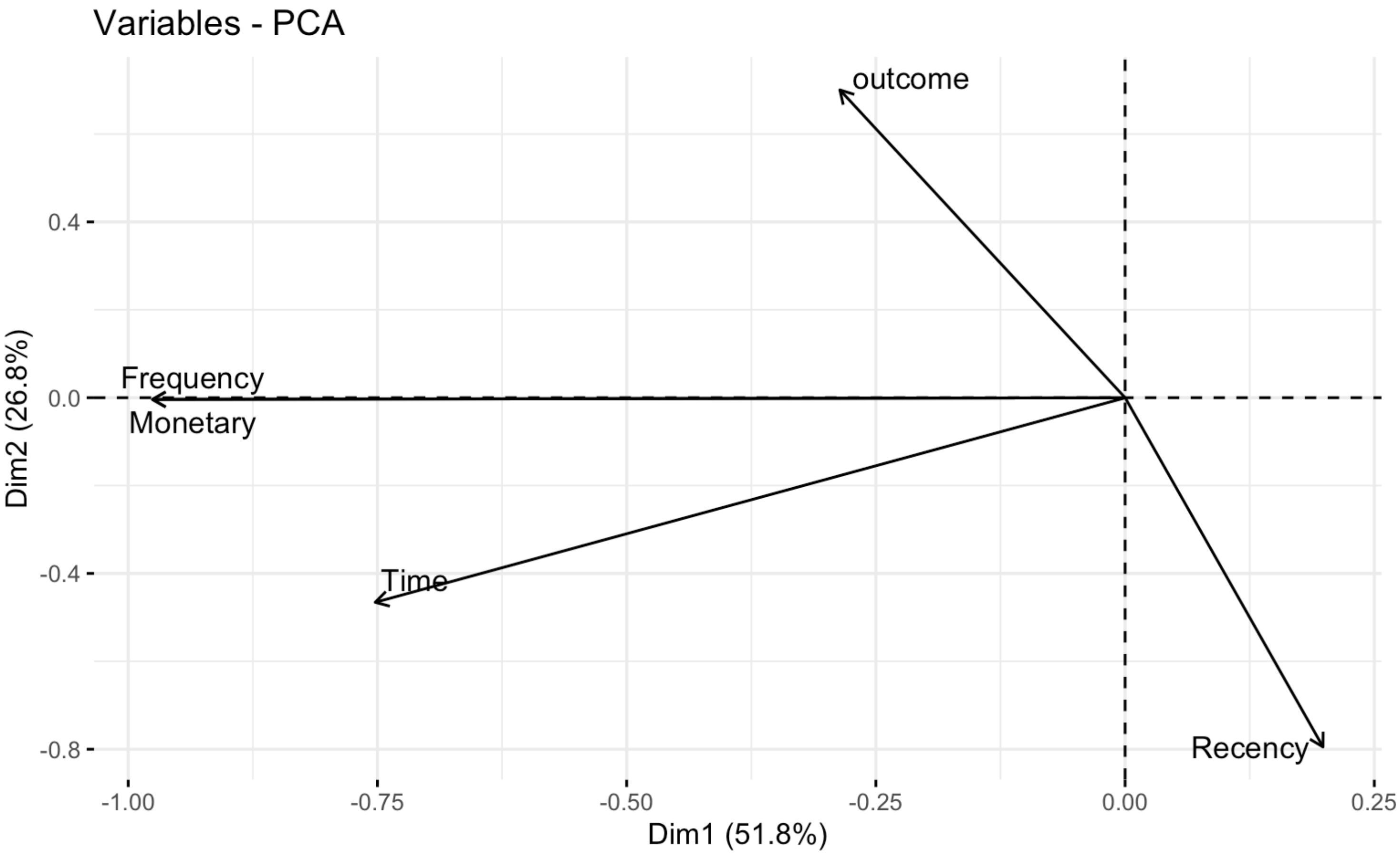
```
get_pca_var(pca)$coord %>% as.data.frame %>%
  arrange(Dim.2) %>% select(Dim.2) # for PC2
```

	Dim.2 <dbl>
Recency	-0.794695194
Time	-0.465706510
Monetary	-0.004004389
Frequency	-0.004004389
outcome	0.700332437

5 rows

Hide

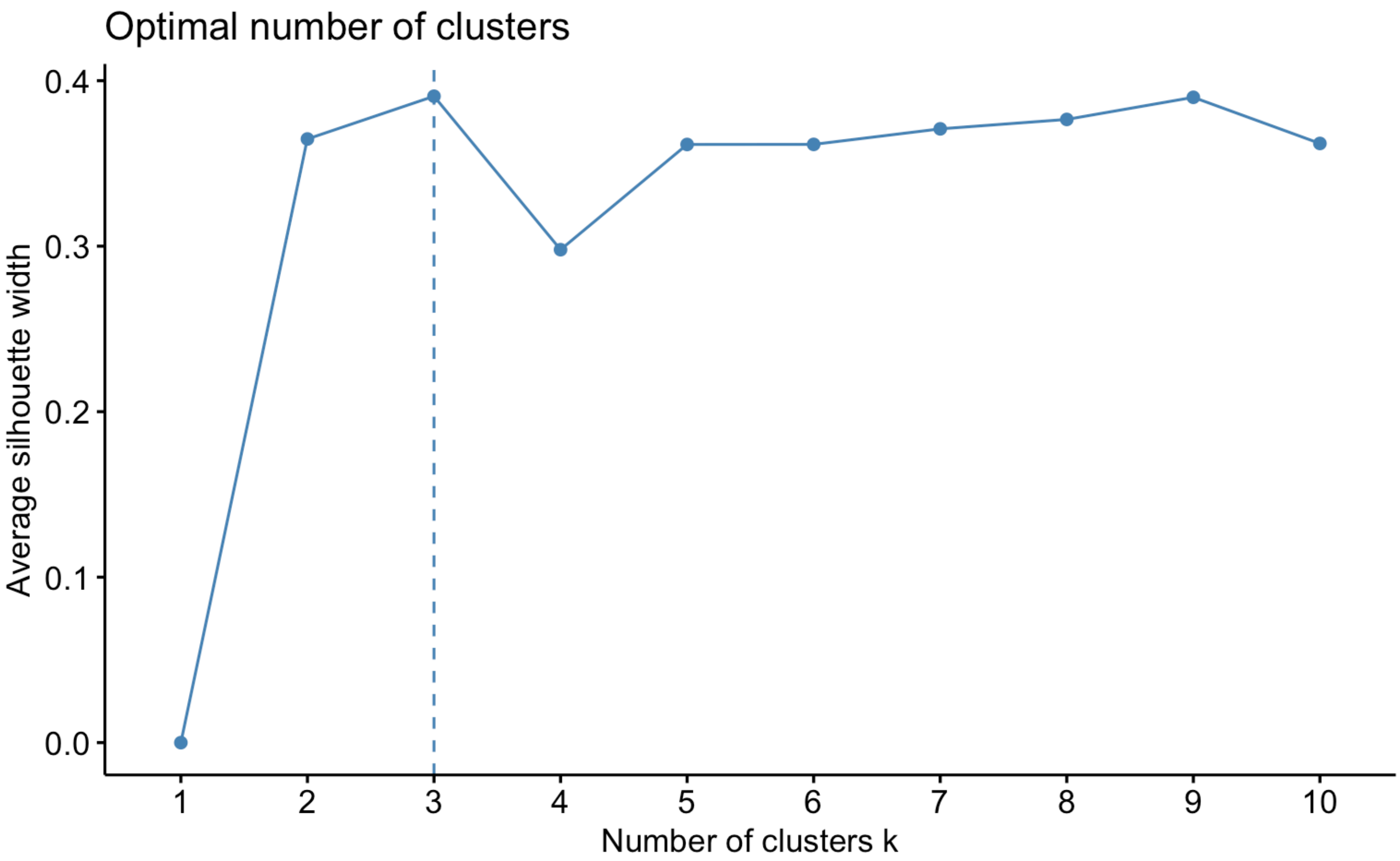
```
fviz_pca_var(pca, col.var = "black",
             repel = TRUE)
```



Based on the interpretations above, we are able to conclude that for dim 1, frequency, monetary, and recency show a strong negative influence which means that donors with high values for these variables will have high scores for pc 1. For dim 2, we see that recency is the highest which means that as donors with higher scores are associated with higher values of recency.

Hide

```
# Perform clustering using kmeans() or pam()
fviz_nbclust(mydata_scaled, kmeans, method = "silhouette")
```



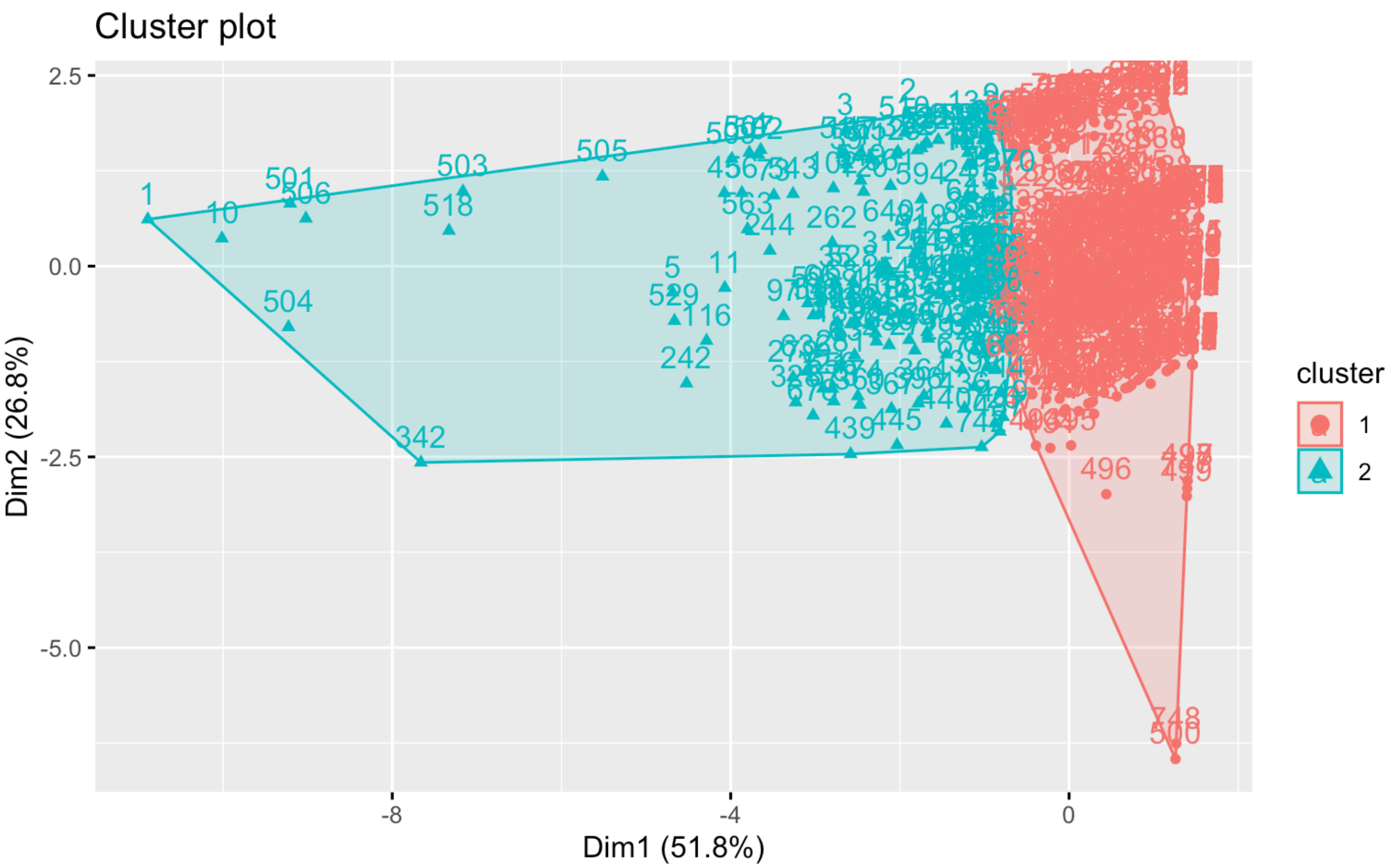
Hide

```
kmeans_results <- mydata_scaled %>%
  kmeans(centers = 2)
kmeans_results
```

```
K-means clustering with 2 clusters of sizes 568, 180

Cluster means:
      Recency  Frequency   Monetary       Time    outcome
1  0.09660404 -0.4035162 -0.4035162 -0.3599042 -0.1039742
2 -0.30483941  1.2733179  1.2733179  1.1356978  0.3280964

Clustering vector:
  [1] 2 2 2 2 2 1 1 2 2 2 2 1 2 2 1 1 2 2 1 1 1 2 1 1 1 2 1 2 1 1 2 2 2 2 2 2 2 1 2 1 2 1 1
1 2 2 2 1 1 1 2 2 1 2 1 2 1 1 2 1 1 2 2
 [64] 1 1 2 1 1 1 2 1 1 2 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 1 1 2 2 2 1 1 2
2 1 1 2 1 2 1 2 2 2 2 2 2 2 1 1 1 1 1 1
[127] 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 2 1 1 1 1 1 1 1 1 2 1 2 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 1 1 1 1 2 1 1 1 1 2
[190] 1 1 1 1 2 1 1 2 1 1 2 1 2 1 2 1 1 1 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 2 1 1 1 2 2 2 1 2 1 1 2 2 1 1 1 2
[253] 2 1 2 1 1 1 1 2 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1
1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1
[316] 1 1 1 2 1 2 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1
[379] 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1 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 1 1 2 1 1 2 1 1 2 2 1
[442] 1 1 1 2 1 1 1 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 1 1 1 1
1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2
[505] 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 2 1 2
1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1
[568] 2 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 2 2 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1
1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 2
[631] 2 1 1 2 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 2
2 2 1 1 2 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1
[694] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 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 1 1 1

Within cluster sum of squares by cluster:
[1] 1559.679 1053.386
 (between_SS / total_SS =  30.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
"size"         "iter"
[9] "ifault"
```

Hide

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



Hide

```
#performing clustering using kmeans()
set.seed(123)
kmeans_model=kmeans(mydata_scaled,center=2)#while the average silhouette width indicates tha
t we should consider 3 clusters, there is not much difference with only keeping 2 clusters
kmeans_model$centers #showing cluster centers
```

```
      Recency  Frequency   Monetary       Time    outcome
1  0.09660404 -0.4035162 -0.4035162 -0.3599042 -0.1039742
2 -0.30483941  1.2733179  1.2733179  1.1356978  0.3280964
```

Hide

```
# showing the mean of each variable for each cluster
# Create basic summary statistics for each cluster in original units
mydata%>%
  mutate(cluster=as.factor(kmeans_model$cluster))%>%
  group_by(cluster)%>%
  summarize_if(is.numeric,mean, na.rm=T)
```

cluster	Recency	Frequency	Monetary	Time	outcome
<fctr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>

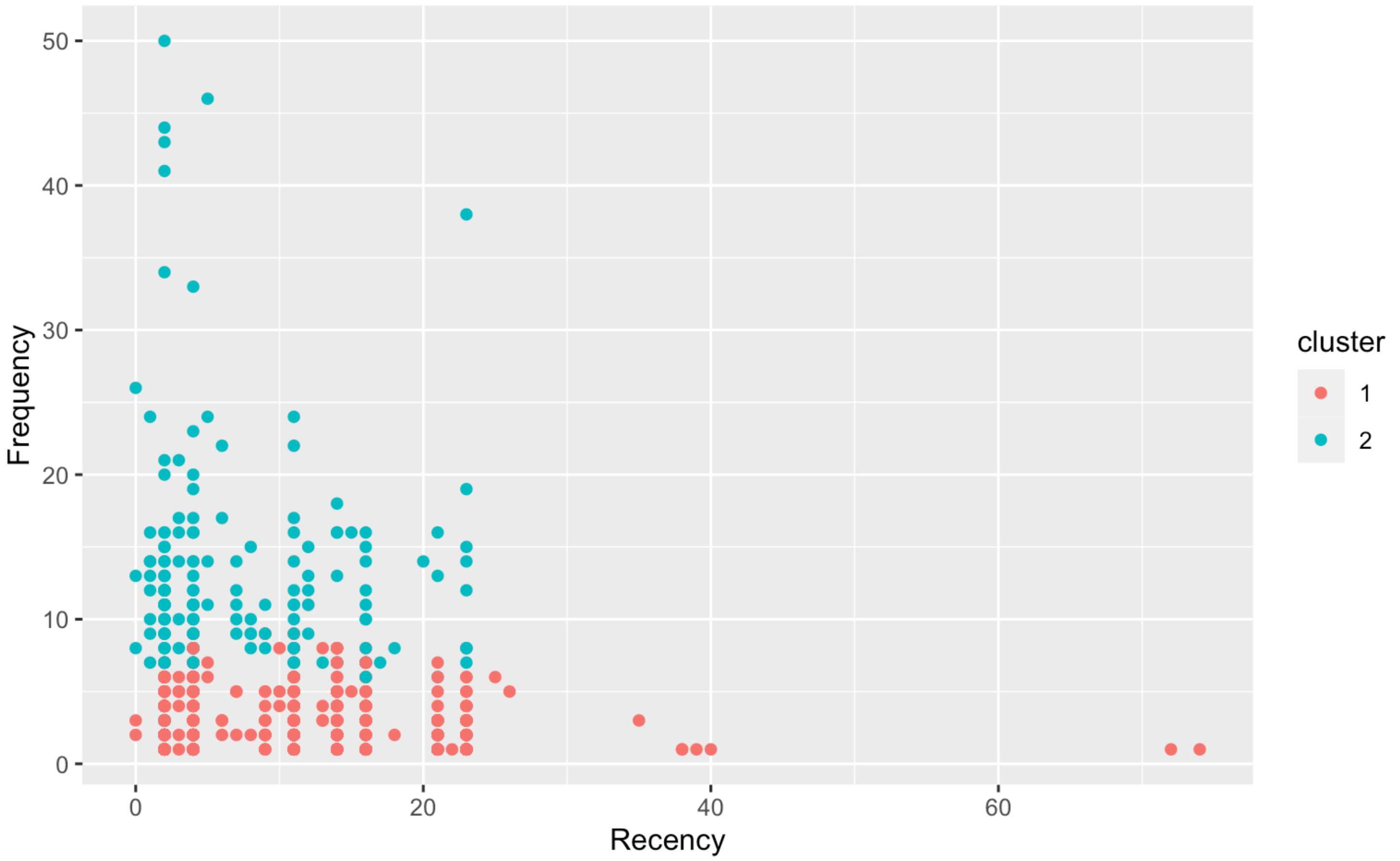
1	10.288732	3.158451	789.6127	25.50880	0.1936620
2	7.038889	12.950000	3237.5000	61.96667	0.3777778

2 rows

Generally, group 2 is higher in all variable except for recency.

Hide

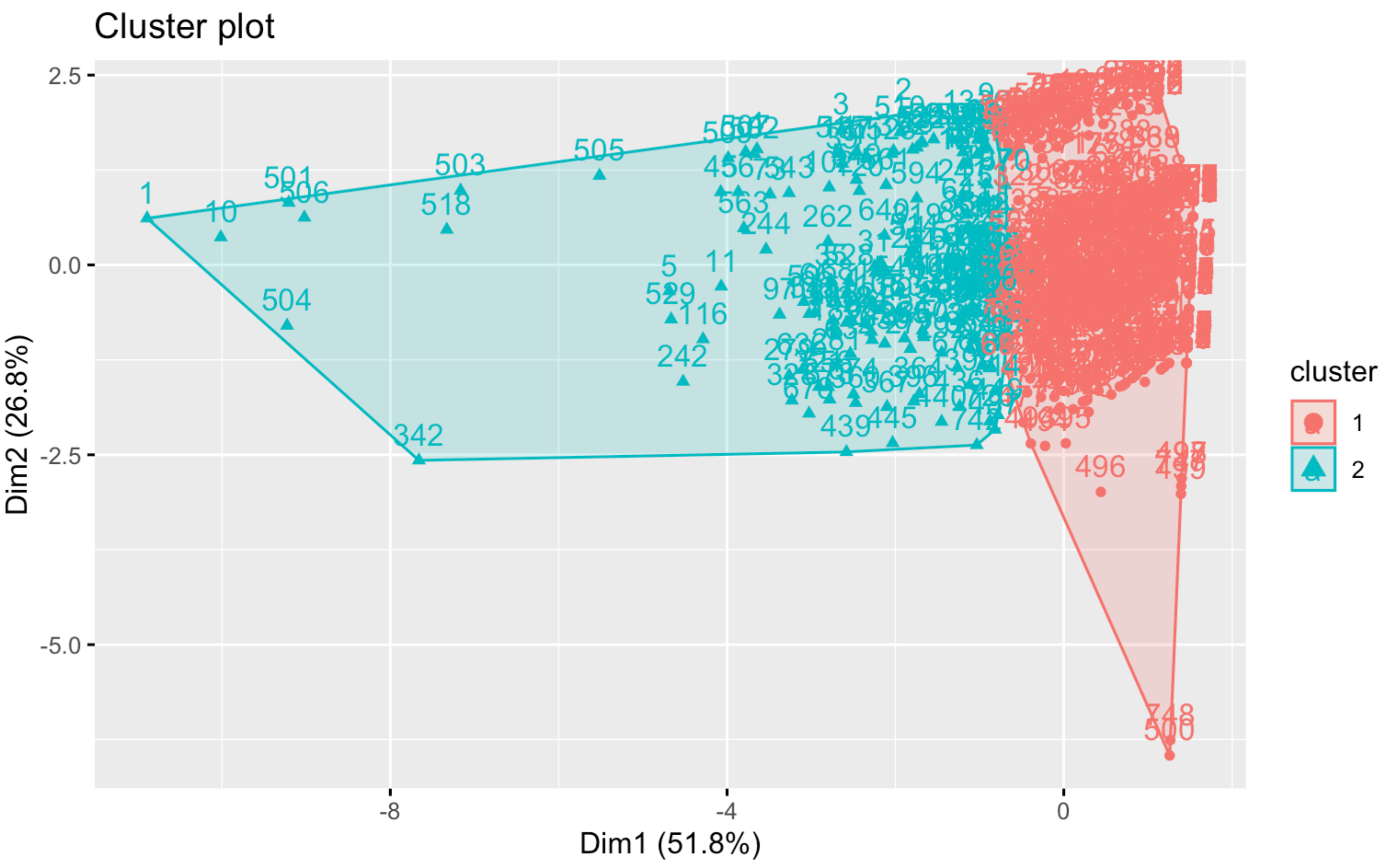
```
#visualizing clusters
mydata%>%
  mutate(cluster=as.factor(kmeans_model$cluster))%>%
  ggplot(aes(x=Recency,y=Frequency, color= cluster))+ geom_point()
```



According to the plot above, we can observe that cluster 2 generally has a higher frequency than cluster 1. We also see that cluster 2 has a few points that are higher in recency than cluster 2.

Hide

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



Hide

```
26.8+51.8
```

```
[1] 78.6
```

The first two components accounts for 78.6% of variation in the response.

6. Discussion

Overall, the research question of ‘Can Recency, Frequency, and Monetary predict if a donor will donate blood in March of 2007?’ was studied through classification and prediction. In this analysis, I used the k-Nearest Neighbor model and PCA to predict if the three variables will predict the outcome or if the donor will make a donation in March 2007. The knn model displayed a performance value of around .76 and pca showed around 78% of the total variation using the first two pcs. Based on these analysis the three variables are valid to use for prediction of blood donation in March 2007. However, in the future there needs to be additional analysis or methods to improve the accuracy of the prediction as the performance values is relatively low.

7. Formatting