This homework data set depends on this data set:

https://www.kaggle.com/yersever/500-person-gender-height-weight-bodymassindex?select=500_Person_Gender_Height_Weight_Index.csv

Please submit this homework by creating an RMD file in your project1 git repo. The RMD should run in the project1 docker environment. You may need to install the gbm package.

Problem 1:

Build a glm in R to classifier individuals as either Male or Female based on their weight and height.

```
#Load and clean up
person.data <- read.csv("500_Person_Gender_Height_Weight_Index.csv")
person.data$Gender[person.data$Gender == 'Female'] = 0
person.data$Gender[person.data$Gender == 'Male'] = 1
person.data$Gender = as.integer(person.data$Gender)

#Train/test split
traininds <- sort(sample(nrow(person.data), nrow(person.data)*.7))

train.data <- person.data[traininds,]
test.data <- person.data[-traininds,]

fit.model <- glm(Gender ~ Height + Weight, data = train.data, family=binomial)

summary(fit.model)
```

```
##
## Call:
## glm(formula = Gender ~ Height + Weight, family = binomial, data = train.data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.243  -1.167  -1.099   1.186   1.260
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.796e-01  1.150e+00   0.852    0.394
## Height      -5.877e-03  6.473e-03  -0.908    0.364
## Weight      -5.286e-05  3.254e-03  -0.016    0.987
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 485.16  on 349  degrees of freedom
## Residual deviance: 484.33  on 347  degrees of freedom
## AIC: 490.33
##
## Number of Fisher Scoring iterations: 3
```

```
test.data$glmProb <- predict(fit.model, test.data, type="response")
test.data <- test.data %>% mutate(glmPred = 1*(glmProb > .5) + 0)
confusionMatrix(as.factor(test.data$Gender), as.factor(test.data$glmPred))
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  0  1
##          0 47 31
##          1 45 27
##
##                Accuracy : 0.4933
##                  95% CI : (0.4108, 0.5761)
##     No Information Rate : 0.6133
##     P-Value [Acc > NIR] : 0.9989
##
##                   Kappa : -0.0226
##
##  Mcnemar's Test P-Value : 0.1359
##
##             Sensitivity : 0.5109
##             Specificity : 0.4655
##          Pos Pred Value : 0.6026
##          Neg Pred Value : 0.3750
##              Prevalence : 0.6133
##          Detection Rate : 0.3133
##    Detection Prevalence : 0.5200
##       Balanced Accuracy : 0.4882
##
##        'Positive' Class : 0
##
```

What is the accuracy of the model?

Worse than random! 0.44

Problem 2:

Use the 'gbm' package to train a similar model. Don't worry about hyper parameter tuning for now.

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
#Train/test split
traininds <- sort(sample(nrow(person.data), nrow(person.data)*.7))

fit.model <- gbm(Gender ~ Height + Weight, data = train.data)
```

```
## Distribution not specified, assuming bernoulli ...
```

```
#summary(fit.model)

test.data$gbmProb <- predict(fit.model, test.data, type="response")
```

```
## Using 100 trees...
```

```
test.data <- test.data %>% mutate(gbmPred = 1*(gbmProb > .5) + 0)
caret::confusionMatrix(as.factor(test.data$Gender), as.factor(test.data$gbmPred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 49 29
##          1 36 36
##
##                Accuracy : 0.5667
##                  95% CI : (0.4834, 0.6473)
##     No Information Rate : 0.5667
##     P-Value [Acc > NIR] : 0.5343
##
##                   Kappa : 0.1287
##
##  Mcnemar's Test P-Value : 0.4568
##
##             Sensitivity : 0.5765
##             Specificity : 0.5538
##          Pos Pred Value : 0.6282
##          Neg Pred Value : 0.5000
##              Prevalence : 0.5667
##          Detection Rate : 0.3267
##    Detection Prevalence : 0.5200
##       Balanced Accuracy : 0.5652
##
##        'Positive' Class : 0
##
```

What is the accuracy of the model?

Also bad, 0.5

Problem 3

Filter the data set so that it contains only 50 Male examples and all female examples. Create a new model for this data set. What is the F1 Score of the model?

```r
males <- person.data[person.data$Gender==1,]
male.50 <- males[sample(nrow(males), 50),]
filt.data <- rbind(male.50, person.data[person.data$Gender==0,])

#Train/test split
traininds <- sort(sample(nrow(filt.data), nrow(filt.data)*.7))

fit.model <- gbm(Gender ~ Height + Weight, data = train.data)
```

```
## Distribution not specified, assuming bernoulli ...
```

```r
#summary(fit.model)

test.data$gbmProb2 <- predict(fit.model, test.data, type="response")
```

```
## Using 100 trees...
```

```r
test.data <- test.data %>% mutate(gbmPred2 = 1*(gbmProb2 > .5) + 0)
confusionMatrix(as.factor(test.data$Gender), as.factor(test.data$gbmPred2))
```

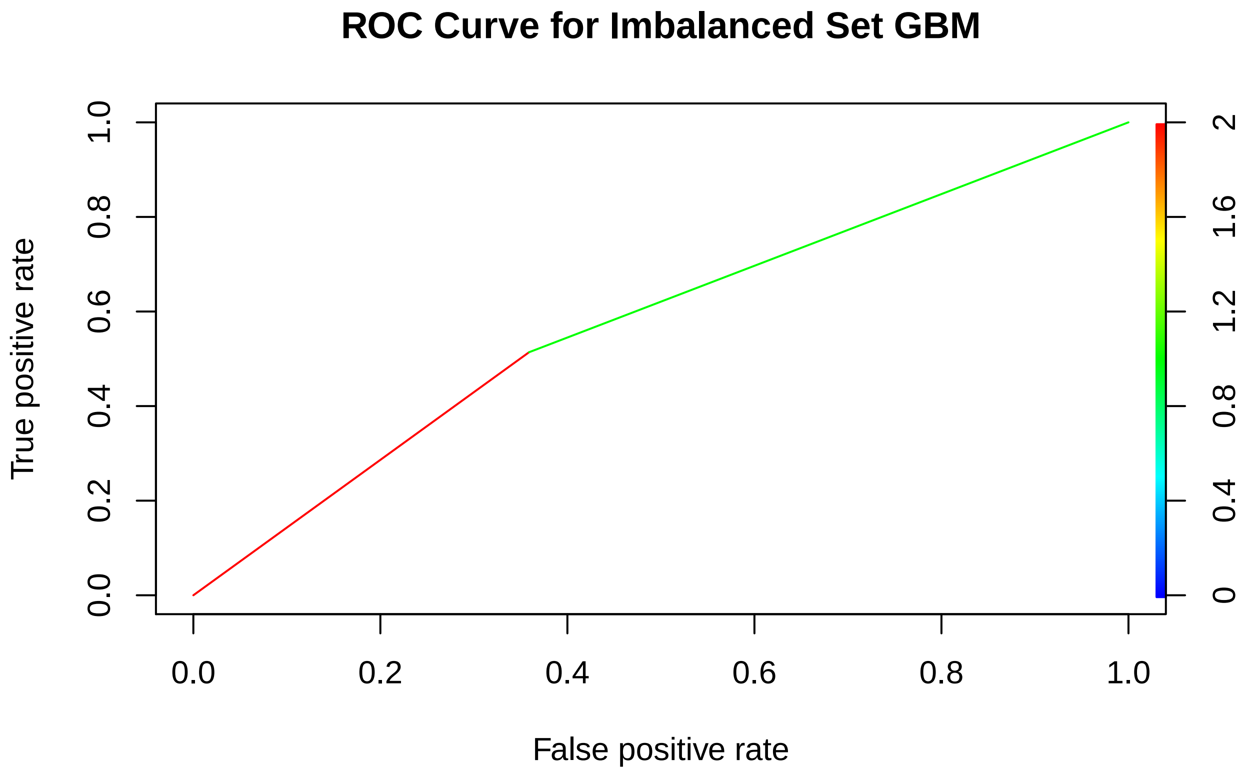
```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  0  1
##          0 50 28
##          1 35 37
##
##                Accuracy : 0.58
##                  95% CI : (0.4968, 0.66)
##     No Information Rate : 0.5667
##     P-Value [Acc > NIR] : 0.4038
##
##                   Kappa : 0.1555
##
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.5882
##             Specificity : 0.5692
##          Pos Pred Value : 0.6410
##          Neg Pred Value : 0.5139
##              Prevalence : 0.5667
##          Detection Rate : 0.3333
##    Detection Prevalence : 0.5200
##       Balanced Accuracy : 0.5787
##
##        'Positive' Class : 0
##
```

Problem 4

For the model in the previous example plot an ROC curve. What does this ROC curve mean?

```r
library(ROCR)

pred <- prediction(test.data$gbmPred2, test.data$Gender)
perf <- performance(pred, 'tpr', 'fpr')
plot(perf, colorize=TRUE, main="ROC Curve for Imbalanced Set GBM")
```



Problem 5

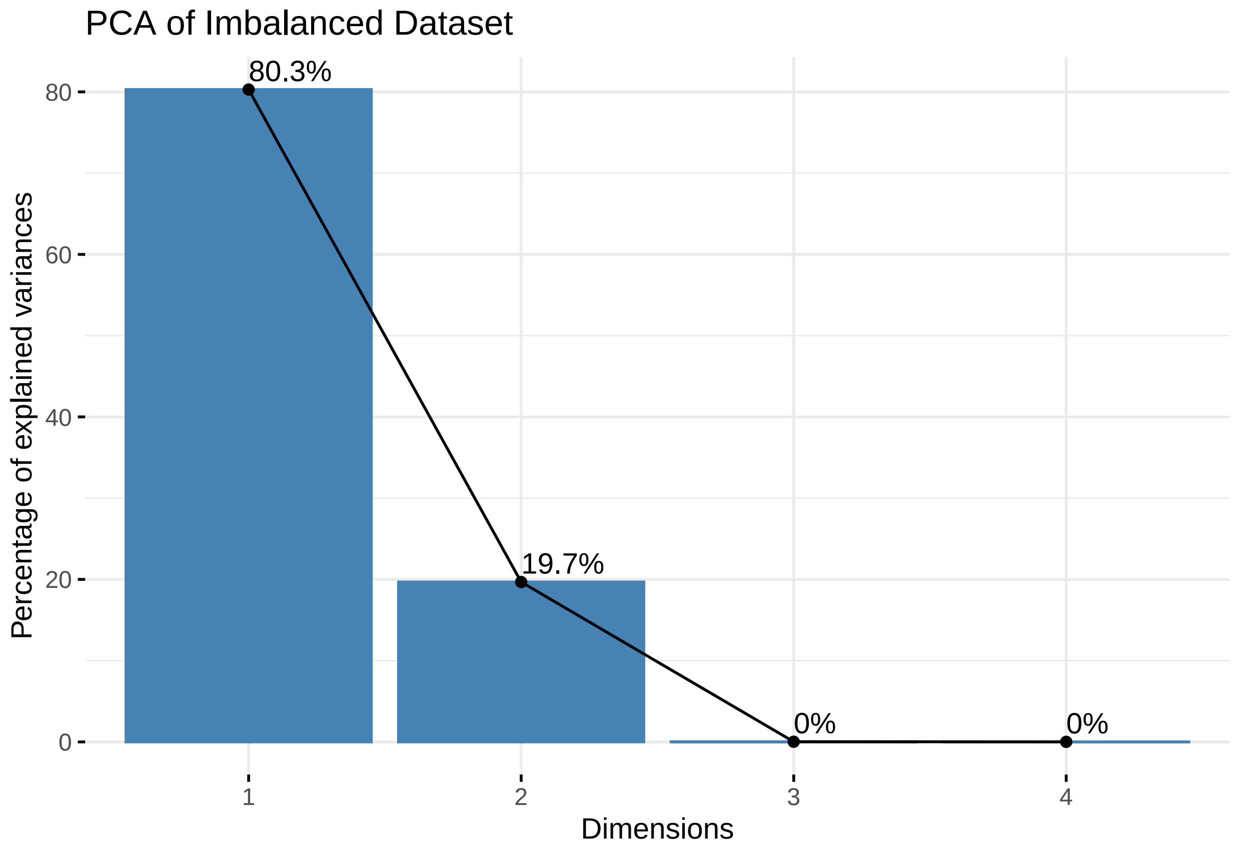
Using K-Means, cluster the same data set. Can you identify the clusters with the known labels? Provide an interpretation of this result.

```
library(factoextra)
```

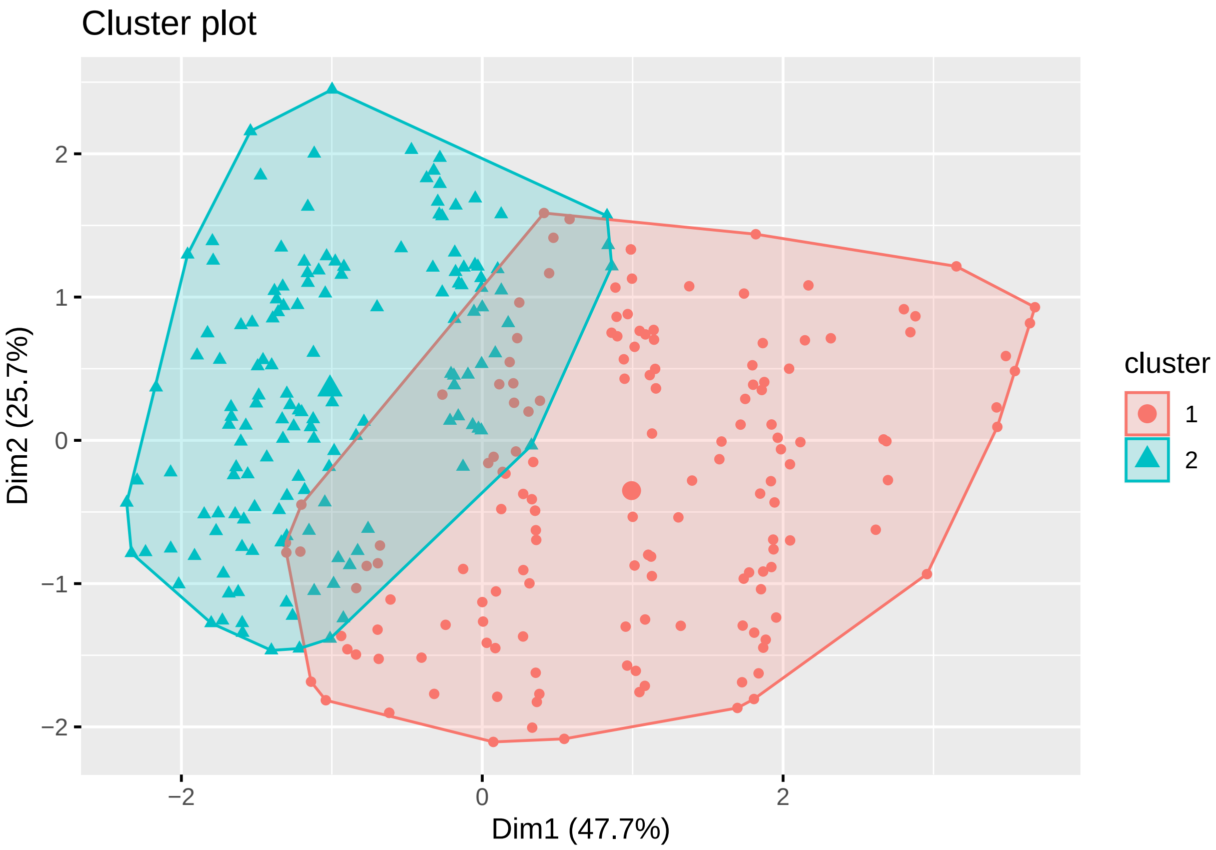
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
pca.fit <- prcomp(filt.data)

fviz_screeplot(pca.fit,
               addlabels=TRUE,
               title="PCA of Imbalanced Dataset")
```



```
# Well... I suppose we could use just one cluster but that wouldn't be very interesting, now would it?
kmeans.fit <- kmeans(filt.data, 2)
fviz_cluster(kmeans.fit, filt.data, geom="point")
```


Cluster plot
Dim2 (25.7%)
Dim1 (47.7%)
2
1
0
−1
−2
−2
0
2
cluster
1
2