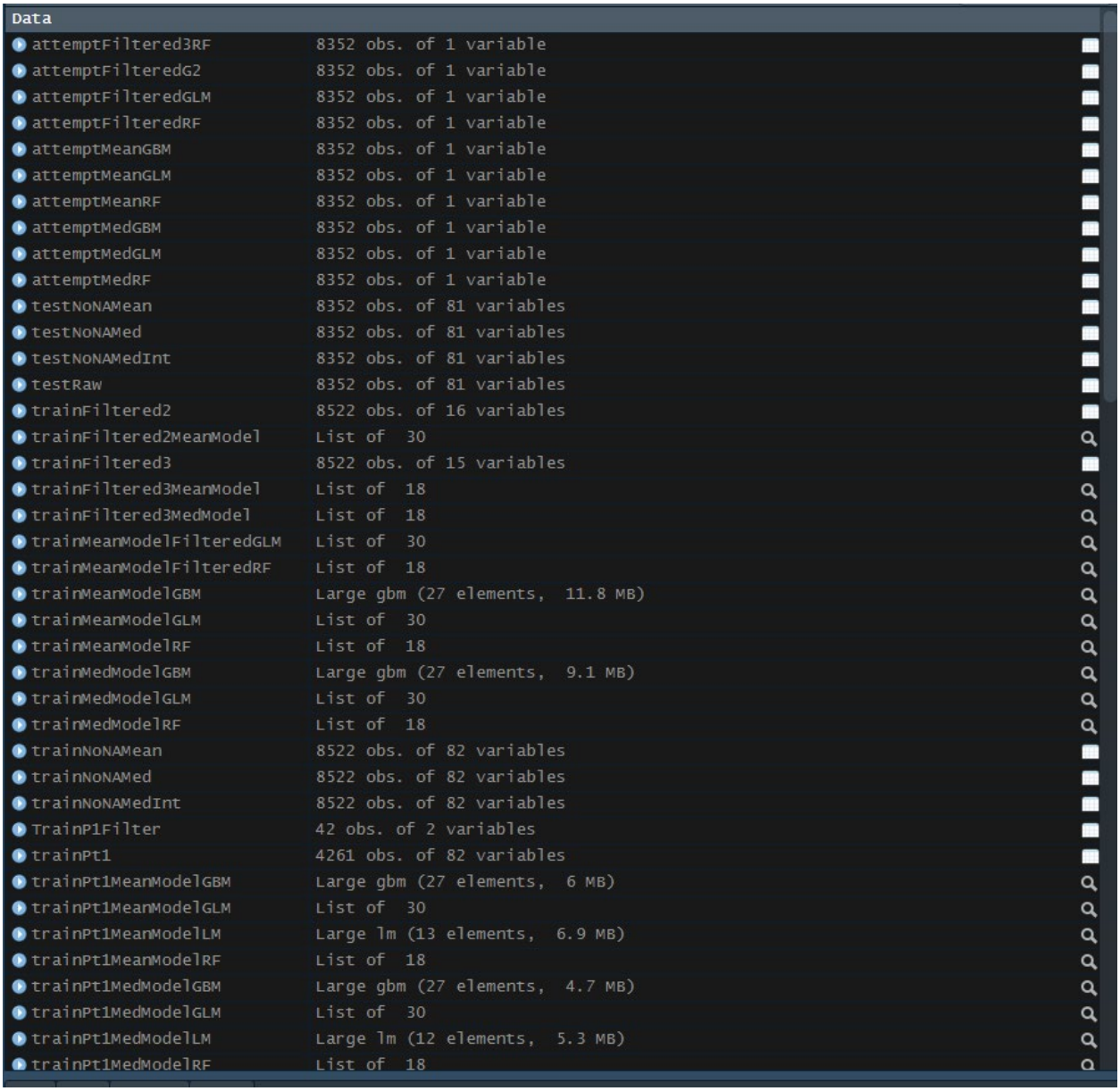
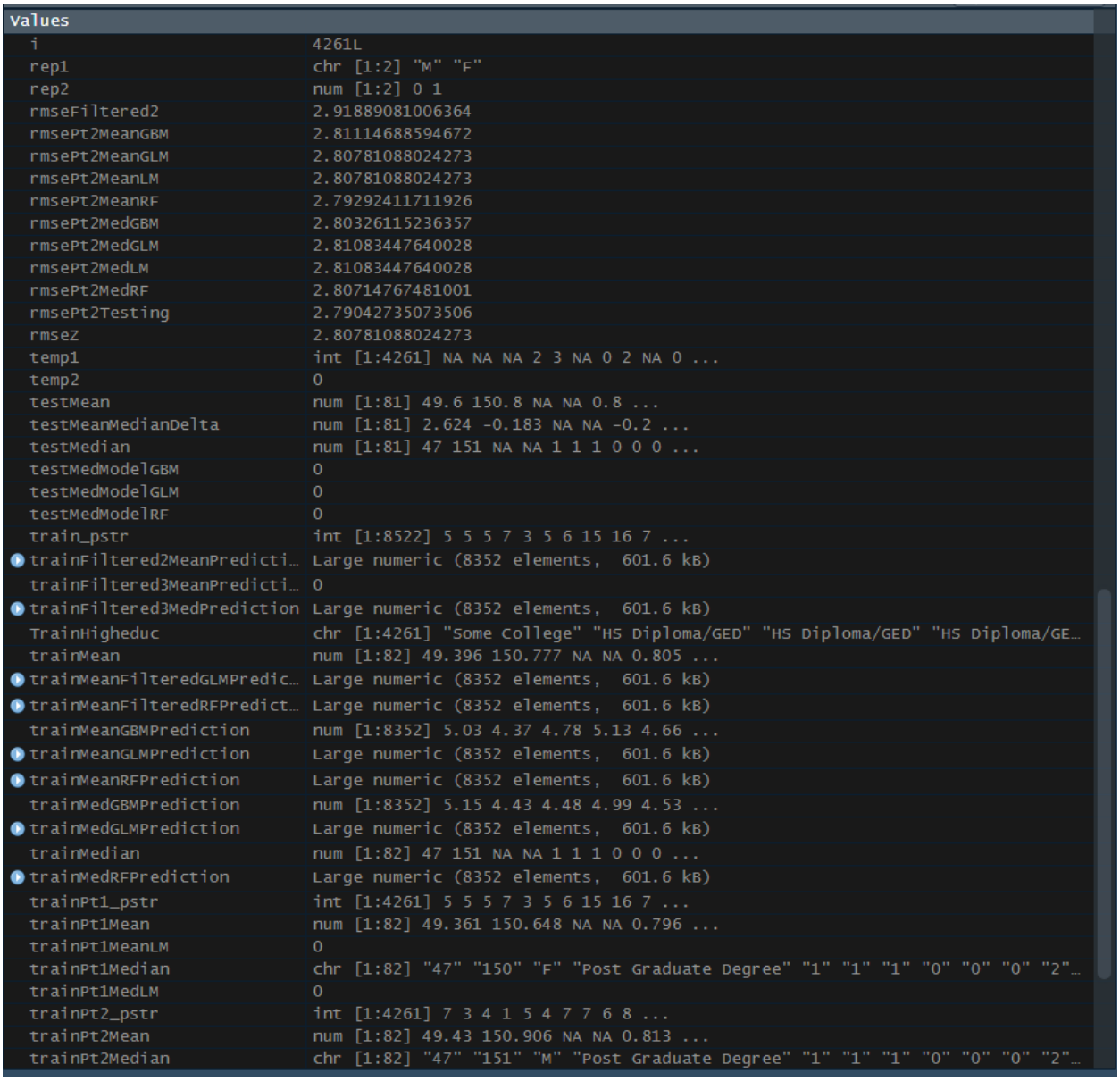
Throughout my submissions to the competition, I tried a variety of different techniques to refine my prediction. Unfortunately, It seems that while I expected R studio to save all of my history to the .Rhistory files that it saves every exit, it seems that it would only save the last 512 entries. Since I did a lot of my model creation at once, A significant amount of my code has been lost. I will provide what code I have, but I unfortunately was unaware of the limit on the history saved.

Firstly, I will go over a general statement about each of my submissions. Please note that all of the scores that are mentioned will be the public score as that is the score that I was seeing at the time of submission.

1. My first submission was done in excel instead of R just so that I could get familiar with the actual numbers in the different features. I ended up only using 7 features for my prediction and received my worst score of 3.41682.
2. My second submission was a GLM using all features and replacing all of the NA values with the mean. This got a score of 2.75938. This would be my best score for a while.
3. Third was a random forest that was made in conjunction with homework 5. It got a slightly worse score of 2.78772. This was my first submission that I did not improve my score.
4. Fourth was a GLM also for homework 4 that got me a score of 2.83371.
5. Fifth was another random forest model, this time however, I used mean replacement of the NA values like I did with my second submission. This got me a very respectable score of 2.75678 and put me in fifth on the leaderboard. This gave me a bias towards the random forest model.
6. On this submission I tried Median replacement and a GBM model. It did not improve my score and got me a 2.76683. Not bad, but GBM models were the most confusing to me so I was dissuaded from using this type.
7. On my seventh submission I started to filter out some of the features as I had a good understanding of how I wanted to proceed model and replacement wise. My first filter used a GLM and filtered out about half of the features. I chose these features based on if removing said feature increased or decreased my local RMSE. Unfortunately, this did not work as well as I hoped and got me a score of 2.76223.
8. This submission was similar to seven only I used RF model instead of GLM. Again, I got an undesirable score of 2.77345. While not a best score, the RF score was worse this time.
9. Submission 9 was a submission of a model I made around the same time as submission 6 and used Median replacement and GLM model. I submitted this older model because I had the prediction made and was not going to submit anything else this day so I figured submitting an older model would be better than no submission. A score of 2.77043 was received.
10. Submission 10 was similar to submission 9 only was Median and RF instead. A score of 2.75537. This was my best score and was kind of upsetting since I had been sitting on this prediction for a few days.
11. I tried removing features again, this time only taking the top 15 features that decreased local RMSE paired with a GLM model. The score of 2.76935 was not encouraging.
12. I tried the same top 15 features as submission 11 with a RF model this time and got an even worse score of 2.82653. I was running out of steam and ideas for submissions at this point.
13. This submission was on the last day and in a last ditch effort to lower my score, I combined the models of my top 3 predictions to make an ensemble model. This worked out much better than expected and produced my lowest score of 2.74304.

14. My last submission was another ensemble, this time with just my top 2 models with a higher weight put of the model that had a lower score. This produced a very good score of 2.74891, but was not my best score.





Values

i	4261L
rep1	chr [1:2] "M" "F"
rep2	num [1:2] 0 1
rmseFiltered2	2.91889081006364
rmsePt2MeanGBM	2.81114688594672
rmsePt2MeanGLM	2.80781088024273
rmsePt2MeanLM	2.80781088024273
rmsePt2MeanRF	2.79292411711926
rmsePt2MedGBM	2.80326115236357
rmsePt2MedGLM	2.81083447640028
rmsePt2MedLM	2.81083447640028
rmsePt2MedRF	2.80714767481001
rmsePt2Testing	2.79042735073506
rmseZ	2.80781088024273
temp1	int [1:4261] NA NA NA 2 3 NA 0 2 NA 0 ...
temp2	0
testMean	num [1:81] 49.6 150.8 NA NA 0.8 ...
testMeanMedianDelta	num [1:81] 2.624 -0.183 NA NA -0.2 ...
testMedian	num [1:81] 47 151 NA NA 1 1 1 0 0 0 ...
testMedModelGBM	0
testMedModelGLM	0
testMedModelRF	0
train_pstr	int [1:8522] 5 5 5 7 3 5 6 15 16 7 ...
trainFiltered2MeanPredicti…	Large numeric (8352 elements, 601.6 kB)
trainFiltered3MeanPredicti…	0
trainFiltered3MedPrediction	Large numeric (8352 elements, 601.6 kB)
TrainHigheduc	chr [1:4261] "Some College" "HS Diploma/GED" "HS Diploma/GED" "HS Diploma/GE…
trainMean	num [1:82] 49.396 150.777 NA NA 0.805 ...
trainMeanFilteredGLMPredic…	Large numeric (8352 elements, 601.6 kB)
trainMeanFilteredRFPredict…	Large numeric (8352 elements, 601.6 kB)
trainMeanGBMPrediction	num [1:8352] 5.03 4.37 4.78 5.13 4.66 ...
trainMeanGLMPrediction	Large numeric (8352 elements, 601.6 kB)
trainMeanRFPrediction	Large numeric (8352 elements, 601.6 kB)
trainMedGBMPrediction	num [1:8352] 5.15 4.43 4.48 4.99 4.53 ...
trainMedGLMPrediction	Large numeric (8352 elements, 601.6 kB)
trainMedian	num [1:82] 47 151 NA NA 1 1 1 0 0 0 ...
trainMedRFPrediction	Large numeric (8352 elements, 601.6 kB)
trainPt1_pstr	int [1:4261] 5 5 5 7 3 5 6 15 16 7 ...
trainPt1Mean	num [1:82] 49.361 150.648 NA NA 0.796 ...
trainPt1MeanLM	0
trainPt1Median	chr [1:82] "47" "150" "F" "Post Graduate Degree" "1" "1" "1" "0" "0" "0" "2"…
trainPt1MedLM	0
trainPt2_pstr	int [1:4261] 7 3 4 1 5 4 7 7 6 8 ...
trainPt2Mean	num [1:82] 49.43 150.906 NA NA 0.813 ...
trainPt2Median	chr [1:82] "47" "151" "M" "Post Graduate Degree" "1" "1" "1" "0" "0" "0" "2"…

Here is a list of code commands that I would use often to create different datasets:

```
library(gbm)

library(randomForest)
library(Metrics)

#format for calculating mean and median for data set
for(i in 1:81){
  +     temp1 <- testRaw %>% filter(!is.na(testRaw[,i]))
  +     testMean[i] <- mean(temp1[,i])
  +}
```

```
#format for replacing empty values in data set
trainNoNAMed <- trainRaw
for(i in 1:82){
  +     temp1 <- trainRaw[,i]
  +     trainNoNAMed[,i] <- temp1 %>% replace_na(trainMedian[i])
  + }

#format for creating models from data sets
trainPt1MeanModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMean)
trainPt1MeanModelGBM <- gbm(pstr ~ ., data = trainPt1NoNAMean %>% select(-SEX, -
higheduc))
trainPt1MeanModelRF <- randomForest(pstr ~ ., trainPt1NoNAMean)

#format for creating prediction vectors
trainPt2TestMeanGLMPrediction <- predict.glm(trainPt1MeanModelGLM, trainPt2TestNo
NAMean)
trainPt2TestMeanGBMPrediction <- predict.gbm(trainPt1MeanModelGBM, trainPt2TestNo
NAMean)
trainPt2TestMeanRFPrediction <- predict(trainPt1MeanModelRF, trainPt2TestNoNAMean
)

#format for calculating random mean square error. lower is better
rmsePt2GLM <- rmse(trainPt2$pstr, trainPt2TestMeanGLMPrediction)
rmsePt2GBM <- rmse(trainPt2$pstr, trainPt2TestMeanGBMPrediction)
rmsePt2RF <- rmse(trainPt2$pstr, trainPt2TestMeanRFPrediction)

#format for converting character data sets to integers. will introduce empty valu
es whereever it fails
for(i in 1:82){
  +     trainPt2NoNAMedInt[,i] <- as.integer(trainPt2NoNAMed[,i])
  + }

TrainPt1Testing <- 0
TrainPt1TestingMeanModel <- 0
TrainPt2TestingMeanPrediction <- 0
TrainPt1Testing <- trainPt1NoNAMean
TrainPt1Testing <- TrainPt1Testing %>% select(-SEX)
TrainPt1TestingMeanModel <- glm(pstr ~ .,data = TrainPt1Testing)
TrainPt2TestingMeanPrediction <- predict.glm(TrainPt1TestingMeanModel, trainPt2Te
stNoNAMean)
rmsePt2Testing <- rmse(trainPt2$pstr, TrainPt2TestingMeanPrediction)

#removing data that seems to increase the rmse
```

```
dataset  %>% select(-4, -6, -7, -10, -14, -15, -19, -20, -22, -24, -27, -28, -
29, -35, -36, -42, -43, -44, -45, -47, -50, -53, -54, -55, -56, -57, -60, -62, -
63, -64, -66, -67, -68, -69, -71, -72, -73, -74, -76, -79)

for(i in 1:4261){
  if(TrainSEX[i] == "M"){
    TrainPt1Testing[i,3] <- 0
  }
  if(TrainSEX[i] == "F"){
    TrainPt1Testing[i,3] <- 1
  }
}

for(i in 1:4261){
  if(TrainHigheduc [i] == "HS Diploma/GED"){
    TrainPt1Testing[i,4] <- 0
  }
  if(TrainHigheduc [i] == "Some College"){
    TrainPt1Testing[i,4] <- 1
  }
  if(TrainHigheduc [i] == "Bachelor"){
    TrainPt1Testing[i,4] <- 2
  }
  if(TrainHigheduc [i] == "Post Graduate Degree"){
    TrainPt1Testing[i,4] <- 3
  }
}
```

Here is all of the R history I was able to salvage:

```
temp1 <- testRaw %>% filter(!is.na(testRaw[,i]))

testMedian[i] <- median(temp1[,i])
}
for(i in 1:81){
temp1 <- testRaw %>% filter(!is.na(testRaw[,i]))
testMode[i] <- mode(temp1[,i])
}
for(i in 1:81){
temp1 <- trainRaw %>% filter(!is.na(trainRaw[,i]))
trainMedian[i] <- median(temp1[,i])
}
remove(testMode)
```

```
remove(trainMode)
temp1 <- 0
trainPt1 <- 0
trainPt2 <- 0
trainPt1 <- trainRaw %>% slice(1:4261)
trainPt2 <- trainRaw %>% slice(4261:8522)
trainPt2 <- trainRaw %>% slice(4262:8522)
trainPt2Mean <- 0
trainPt1Mean <- 0
trainPt1Median <- 0
trainPt2Median <- 0
for(i in 1:81){
temp1 <- trainPt1 %>% filter(!is.na(trainPt1[,i]))
trainPt1Median[i] <- median(temp1[,i])
}
for(i in 1:81){
temp1 <- trainPt1 %>% filter(!is.na(trainPt2[,i]))
trainPt2Median[i] <- median(temp1[,i])
}
for(i in 1:82){
temp1 <- trainPt1 %>% filter(!is.na(trainPt2[,i]))
trainPt2Median[i] <- median(temp1[,i])
}
for(i in 1:82){
temp1 <- trainPt1 %>% filter(!is.na(trainPt1[,i]))
trainPt1Median[i] <- median(temp1[,i])
}
for(i in 1:82){
temp1 <- trainRaw %>% filter(!is.na(trainRaw[,i]))
trainMedian[i] <- median(temp1[,i])
}
for(i in 1:82){
temp1 <- trainRaw %>% filter(!is.na(trainRaw[,i]))
trainMean[i] <- mean(temp1[,i])
}
for(i in 1:82){
temp1 <- trainPt2 %>% filter(!is.na(trainPt2[,i]))
trainPt2Median[i] <- median(temp1[,i])
}
for(i in 1:82){
temp1 <- trainPt2 %>% filter(!is.na(trainPt2[,i]))
trainPt2Mean[i] <- mean(temp1[,i])
}
for(i in 1:82){
temp1 <- trainPt1 %>% filter(!is.na(trainPt1[,i]))
```

```
trainPt1Mean[i] <- mean(temp1[,i])
}
trainP1_pstr <- 0
trainP2_pstr <- 0
remove(trainP1_pstr)
remove(trainP2_pstr)
trainPt2_pstr <- 0
trainPt1_pstr <- 0
trainPt1_pstr <- train_pstr %>% slice(1:4261)
trainPt1_pstr <- train_pstr[1:4261]
trainPt1_pstr <- train_pstr[4261:8522]
trainPt1_pstr <- train_pstr[1:4261]
trainPt2_pstr <- train_pstr[4261:8522]
trainPt2_pstr <- train_pstr[4262:8522]
trainPt1Test <- trainPt1 %>% select(-pstr)
trainPt2Test <- trainPt2 %>% select(-pstr)
testNoNA <- 0
trainNoNA <- 0
trainPt1NoNA <- 0
trainPt2NoNA <- 0
trainPt2TestNoNA <- 0
trainPt1TestNoNA <- 0
remove()
remove(testNoNA)
remove(trainNoNA)
remove(trainPt1NoNA)
remove(trainPt2NoNA)
remove(trainPt2TestNoNA)
remove(trainPt1TestNoNA)
testNoNAMed <- 0
testNoNAMean <- 0
trainNoNAMean <- 0
trainNoNAMed <- 0
trainPt1NoNAMed <- 0
trainPt2NoNAMed <- 0
trainPt1NoNAMean <- 0
trainPt2NoNAMean <- 0
for(i in 1:82){
testNoNAMean[i] <- replace_na(testRaw[,i], testMean[i])
}
testNoNAMean[18] <- replace_na(testRaw[,18], testMean[18])
testNoNAMean <- replace_na(testRaw, testMean)
for(i in 1:82){
testNoNAMean[i] <- replace_na(testRaw[,i], testMean[i])
}
```

```r
for(i in 1:82){
testNoNAMean[i] <- replace_na(testRaw[,i], testMean[i])
}
for(i in 1:81){
testNoNAMean[i] <- replace_na(testRaw[,i], testMean[i])
}
for(i in 1:81){
testNoNAMean[i] <- replace_na(testRaw[,i], testMean[i])
}
temp1 <- 0
for(i in 1:81){
temp1 <- testRaw %>% select(i)
testNoNAMean[i] <- replace_na(temp1, testMean[i])
}
testRaw %>% select(1)
for(i in 1:81){
temp1 <- testRaw %>% select(i)
}
testNoNAMean[81] <- replace_na(temp1, testMean[81])
replace_na(temp1, testMean[81])
replace_na(temp1, testMean)
replace_na(temp1, testMean[1,81])
replace_na(temp1, testMean[81,1])
replace_na(temp1, testMean[81])
testMean[81]
replace_na(temp1, 1)
replace_na(testRaw, 1)
replace_na(temp1, testMean)
replace_na(temp1)
replace_na(temp1, testMean)
replace_na(testRaw, 1)
replace_na(testRaw, testMean)
testNoNAMean <- 0
replace_na()
replace_na(testRaw)
replace_na(testRaw, testMean)
replace_na(testRaw)
replace_na()
testRaw %>% replace_na(testMean)
testRaw %>% replace_na(list(testMean))
list(testMean)
testRaw
temp1
temp1 %>% replace_na(0)
temp1 %>% replace_na("idk")
```

```
as,vector(temp1)
as.vector(temp1)
temp1 <- as.vector(temp1)
temp1 %>% replace_na(0)
is_list(temp1)
is_list(0)
is_list(testMean)
temp2 <- testMean
is_list(temp2)
temp2 <- as.vector(temp2)
is_list(temp2)
temp2 <- c(temp2)
t(temp2)
temp2 <- t(temp2)
is_list(temp2)
temp2 <- as.vector(temp2)
is_list(temp1)
temp2 <- data.frame(temp2)
is_list(temp2)
temp1 <- as.vector(temp1)
temp1 <- t(temp1)
is_list(temp1)
temp1 %>% replace_na(0)
View(temp1)
View(temp1)
temp1 <- temp1 %>% replace_na(0)
temp1 <- testRaw[,18]
is_list(temp1)
temp1 <- temp1 %>% replace_na(testMean[18])
temp1 <- testRaw[,18]
for(i in 1:81){
temp1 <- testRaw[]
temp1 <- testRaw %>% select(i)
}
temp1 <- 0
temp2 <- 0
for(i in 1:81){
temp1 <- testRaw[,i]
testNoNAMean <- temp1 %>% replace_na(testMean[i])
}
for(i in 1:81){
temp1 <- testRaw[,i]
testNoNAMean[i] <- temp1 %>% replace_na(testMean[i])
}
warnings()
```

```
testNoNAMean <- 0
testNoNAMean[81] <- 0
testNoNAMean[8381, 81] <- 0
testNoNAMean <- 0
testNoNAMean[8381, 81] <- 0
testNoNAMean <- testRaw
for(i in 1:81){
temp1 <- testRaw[,i]
testNoNAMean[,i] <- temp1 %>% replace_na(testMean[i])
}
View(testNoNAMean)
View(testNoNAMean)
testNoNAMed <- testRaw
for(i in 1:81){
temp1 <- testRaw[,i]
testNoNAMed[,i] <- temp1 %>% replace_na(testMedian[i])
}
trainNoNAMed <- trainRaw
trainNoNAMean <- testRaw
for(i in 1:81){
temp1 <- trainRaw[,i]
trainNoNAMed[,i] <- temp1 %>% replace_na(trainMedian[i])
}
for(i in 1:81){
temp1 <- trainRaw[,i]
trainNoNAMean[,i] <- temp1 %>% replace_na(trainMean[i])
}
trainNoNAMean <- 0
trainNoNAMean <- trainRaw
for(i in 1:81){
temp1 <- trainRaw[,i]
trainNoNAMean[,i] <- temp1 %>% replace_na(trainMean[i])
}
trainPt1Mean <- trainPt1
for(i in 1:82){
temp1 <- trainPt1 %>% filter(!is.na(trainPt1[,i]))
trainPt1Mean[i] <- mean(temp1[,i])
}
trainPt1Mean <- 0
for(i in 1:82){
temp1 <- trainPt1 %>% filter(!is.na(trainPt1[,i]))
trainPt1Mean[i] <- mean(temp1[,i])
}
for(i in 1:82){
temp1 <- trainRaw[,i]
```

```
trainPt1NoNAMean[,i] <- temp1 %>% replace_na(trainMean[i])
}
for(i in 1:82){
temp1 <- trainRaw[,i]
trainNoNAMean[,i] <- temp1 %>% replace_na(trainMean[i])
}
for(i in 1:82){
temp1 <- trainRaw[,i]
trainNoNAMed[,i] <- temp1 %>% replace_na(trainMedian[i])
}
temp1 <- 0
trainPt1NoNAMean <- trainPt1
trainPt1NoNAMed <- trainPt1
trainPt2NoNAMed <- trainPt1
trainPt2NoNAMean <- trainPt1
trainPt2NoNAMed <- trainPt2
trainPt2NoNAMean <- trainPt2
for(i in 1:82){
temp1 <- trainPt1[,i]
trainPt1NoNAMed[,i] <- temp1 %>% replace_na(trainPt1Median[i])
}
for(i in 1:82){
temp1 <- trainPt1[,i]
trainPt1NoNAMean[,i] <- temp1 %>% replace_na(trainPt1Mean[i])
}
for(i in 1:82){
temp1 <- trainPt2[,i]
trainPt2NoNAMean[,i] <- temp1 %>% replace_na(trainPt2Mean[i])
}
for(i in 1:82){
temp1 <- trainPt2[,i]
trainPt2NoNAMed[,i] <- temp1 %>% replace_na(trainPt2Median[i])
}
trainPt2TestNoNAMean <- trainPt2Test
trainPt1TestNoNAMean <- trainPt1Test
trainPt1TestNoNAMed <- trainPt1Test
trainPt2TestNoNAMed <- trainPt2Test
for(i in 1:82){
temp1 <- trainPt2Test[,i]
trainPt2TestNoNAMed[,i] <- temp1 %>% replace_na(trainPt2Median[i])
}
for(i in 1:81){
temp1 <- trainPt2Test[,i]
trainPt2TestNoNAMed[,i] <- temp1 %>% replace_na(trainPt2Median[i])
}
```

```
for(i in 1:81){
temp1 <- trainPt1Test[,i]
trainPt1TestNoNAMed[,i] <- temp1 %>% replace_na(trainPt1Median[i])
}
trainPt1MeanModelGLM <- 0
trainPt1MeanModelGBM <- 0
trainPt1MeanModelRF <- 0
trainPt1MedModelRF <- 0
trainPt1MedModelGLM <- 0
trainPt1MedModelGBM <- 0
trainPt1MeanModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMean)
trainPt1MeanModelGLM <- gbm(pstr ~ ., data = trainPt1NoNAMean)
library(gbm)
trainPt1MeanModelGLM <- gbm(pstr ~ ., data = trainPt1NoNAMean)
trainPt1MeanModelGLM <- gbm(pstr ~ ., data = trainPt1NoNAMean %>% select(-SEX, -
higheduc))
trainPt1MeanModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMean)
trainPt1MeanModelGBM <- gbm(pstr ~ ., data = trainPt1NoNAMean %>% select(-SEX, -
higheduc))
trainPt1MeanModelGBM <- randomForest(pstr ~ ., trainPt1NoNAMean)
)
trainPt1MeanModelGBM <- gbm(pstr ~ ., data = trainPt1NoNAMean %>% select(-SEX, -
higheduc))
trainPt1MeanModelRF <- randomForest(pstr ~ ., trainPt1NoNAMean)
trainPt1MeanModelGLM <- glm(pstr ~ ., data = trainPt1TestNoNAMean)
trainPt2TestMeanGLM <- 0
trainPt2TestMeanGBM <- 0
trainPt2TestMeanRF <- 0
remove(trainPt2TestMeanGLM )
remove(trainPt2TestMeanGBM )
remove(trainPt2TestMeanRF )
trainPt2TestMeanGLMPrediction <- 0
trainPt2TestMeanGBMPrediction <- 0
trainPt2TestMeanRFPrediction <- 0
trainPt2TestMeanGLMPrediction <- predict.glm(trainPt1MeanModelGLM, trainPt2TestNo
NAMean)
trainPt2TestMeanGBMPrediction <- predict.gbm(trainPt1MeanModelGBM, trainPt2TestNo
NAMean)
trainPt2TestMeanRFPrediction <- predict(trainPt1MeanModelRF, trainPt2TestNoNAMean
)
trainPt2TestMeanGLMPrediction
trainPt1MeanModelGLM
trainPt2TestNoNAMean
trainPt1TestNoNAMean
trainPt1NoNAMean
```

```r
for(i in 1:81){
temp1 <- trainPt2Test[,i]
trainPt2TestNoNAMed[,i] <- temp1 %>% replace_na(trainPt2Median[i])
}
for(i in 1:81){
temp1 <- trainPt1Test[,i]
trainPt1TestNoNAMed[,i] <- temp1 %>% replace_na(trainPt1Median[i])
}
for(i in 1:81){
temp1 <- trainPt1Test[,i]
trainPt1TestNoNAMean[,i] <- temp1 %>% replace_na(trainPt1Mean[i])
}
for(i in 1:81){
temp1 <- trainPt2Test[,i]
trainPt2TestNoNAMean[,i] <- temp1 %>% replace_na(trainPt2Mean[i])
}
trainPt1TestNoNAMean
trainPt1MeanModelGBM <- gbm(pstr ~ ., data = trainPt1NoNAMean %>% select(-SEX, -
higheduc))
trainPt1MeanModelGLM <- glm(pstr ~ ., data = trainPt1TestNoNAMean)
trainPt1MeanModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMean)
trainPt2TestMeanRFPrediction <- predict(trainPt1MeanModelRF, trainPt2TestNoNAMean
)
trainPt2TestMeanGBMPrediction <- predict.gbm(trainPt1MeanModelGBM, trainPt2TestNo
NAMean)
trainPt2TestMeanGLMPrediction <- predict.glm(trainPt1MeanModelGLM, trainPt2TestNo
NAMean)
trainPt2TestMeanGLMPrediction
rmsePt2GLM <- rmse(trainPt2$pstr, trainPt2TestMeanGLMPrediction)
library(Metrics)
rmsePt2GLM <- rmse(trainPt2$pstr, trainPt2TestMeanGLMPrediction)
rmsePt2GLM
rmsePt2GBM <- rmse(trainPt2$pstr, trainPt2TestMeanGBMPrediction)
rmsePt2GBM
rmsePt2RF <- rmse(trainPt2$pstr, trainPt2TestMeanRFPrediction)
rmsePt2RF
rmse(trainPt2$pstr, trainPt2$pstr)
trainPt1MedModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMed)
trainPt1MedModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMed %>% select(-SEX, -
higheduc))
trainPt1MeanModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMean)
trainPt1NoNAMean
trainPt1NoNAMed
summary(trainPt1NoNAMed)
as.integer(trainPt1NoNAMed)
```

```
as.integer(trainPt1NoNAMed[,1])
trainPt1NoNAMedInt <- 0
trainPt1NoNAMedInt <- 0
trainPt2NoNAMedInt <- 0
trainPt2TestNoNAMedInt <- 0
trainPt1TestNoNAMedInt <- 0
for(i in 1:82){
trainPt2NoNAMedInt[,i] <- as.integer(trainPt1NoNAMed[,1])
}
View(trainPt2TestNoNAMean)
trainPt1TestNoNAMedInt <- trainPt1TestNoNAMean
trainPt2TestNoNAMedInt <- trainPt2TestNoNAMean
trainPt2NoNAMedInt <- trainPt2NoNAMean
trainPt1NoNAMedInt <- trainPt1NoNAMean
for(i in 1:82){
trainPt2NoNAMedInt[,i] <- as.integer(trainPt1NoNAMed[,1])
}
for(i in 1:82){
trainPt2NoNAMedInt[,i] <- as.integer(trainPt2NoNAMed[,1])
}
for(i in 1:82){
trainPt1NoNAMedInt[,i] <- as.integer(trainPt1NoNAMed[,1])
}
for(i in 1:81){
trainPt1TestNoNAMedInt[,i] <- as.integer(trainPt1TestNoNAMed[,1])
}
for(i in 1:81){
trainPt2TestNoNAMedInt[,i] <- as.integer(trainPt2TestNoNAMed[,1])
}
for(i in 1:81){
trainPt2TestNoNAMedInt[,i] <- as.integer(trainPt2TestNoNAMed[,i])
}
for(i in 1:81){
trainPt2TestNoNAMedInt[,i] <- as.integer(trainPt2TestNoNAMed[,i])
}
for(i in 1:81){
trainPt1TestNoNAMedInt[,i] <- as.integer(trainPt1TestNoNAMed[,i])
}
for(i in 1:82){
trainPt1NoNAMedInt[,i] <- as.integer(trainPt1NoNAMed[,i])
}
for(i in 1:82){
trainPt2NoNAMedInt[,i] <- as.integer(trainPt2NoNAMed[,i])
}
```

```
trainPt1MedModelGLM <- glm(pstr ~ ., data = trainPt1NoNAMedInt %>% select(-SEX, -higheduc))
trainPt1MedModelGBM <- gbm(pstr ~ ., data = trainPt1NoNAMedInt %>% select(-SEX, -higheduc))
trainPt1MedModelRF <- randomForest(pstr ~ ., data = trainPt1NoNAMedInt %>% select(-SEX, -higheduc))
trainPt2TestMedGLMPrediction <- predict.glm(trainPt1MedModelGLM, trainPt2TestNoNAMedInt %>% select(-SEX, -higheduc))
trainPt2TestMedGBMPrediction <- predict.gbm(trainPt1MedModelGBM, trainPt2TestNoNAMedInt %>% select(-SEX, -higheduc))
trainPt2TestMedRFPrediction <- predict(trainPt1MedModelRF, trainPt2TestNoNAMedInt %>% select(-SEX, -higheduc))
rmsePt2MedGLM <- rmse(trainPt2$pstr, trainPt2TestMedGLMPrediction)
rmsePt2MedGBM <- rmse(trainPt2$pstr, trainPt2TestMedGBMPrediction)
rmsePt2MedRF <- rmse(trainPt2$pstr, trainPt2TestMedRFPrediction)
rmsePt2MeanGLM <- rmsePt2GLM
rmsePt2MeanGBM <- rmsePt2GBM
rmsePt2MeanRF <- rmsePt2RF
remove(rmsePt2RF)
remove(rmsePt2GLM)
remove(rmsePt2GBM)
testMeanModelGLM <- 0
testMeanModelGBM <- 0
testMeanModelRF <- 0
testMedModelRF <- 0
testMedModelGBM <- 0
testMedModelGLM <- 0
testNoNAMedInt <- testRaw
trainNoNAMedInt <- trainRaw
for(i in 1:82){
+     trainPNoNAMedInt[,i] <- as.integer(trainNoNAMed[,i])
+ }
for(i in 1:82){     trainPNoNAMedInt[,i] <- as.integer(trainNoNAMed[,i])
}
for(i in 1:82){
trainPNoNAMedInt[,i] <- as.integer(trainNoNAMed[,i])
}
for(i in 1:82){
trainNoNAMedInt[,i] <- as.integer(trainNoNAMed[,i])
}
for(i in 1:81){
testNoNAMedInt[,i] <- as.integer(testNoNAMed[,i])
}
remove(testMeanModelGBM)
remove(testMeanModelGLM)
```

```
remove(testMeanModelRF)
trainMeanModelGLM <- 0
trainMeanModelRF <- 0
trainMeanModelGBM <- 0
trainMeanModelGLM <- glm(pstr ~ ., data = trainNoNAMean)
trainPt1MeanModelGBM <- gbm(pstr ~ ., data = trainNoNAMean %>% select(-SEX, -
higheduc))
trainMeanModelRF <- randomForest(pstr ~ ., trainNoNAMean)
trainMedModelGLM <- 0
trainMedModelGBM <- 0
trainMedModelRF <- 0
trainMedModelGBM <- gbm(pstr ~ ., data = trainNoNAMedInt %>% select(-SEX, -
higheduc))
trainMedModelGLM <- glm(pstr ~ ., data = trainNoNAMedInt %>% select(-SEX, -
higheduc))
trainMedModelRF <- randomForest(pstr ~ ., trainNoNAMedInt %>% select(-SEX, -
higheduc))
trainPt1MeanModelGBM <- gbm(pstr ~ ., data = trainPt1NoNAMean %>% select(-SEX, -
higheduc))
trainMeanModelGBM <- gbm(pstr ~ ., data = trainNoNAMean %>% select(-SEX, -
higheduc))
trainPt1MeanLM <- 0
trainPt1MedLM <- 0
trainPt1MeanModelLM <- lm(pstr ~ ., data = trainPt1NoNAMean)
trainPt1MedModelLM <- lm(pstr ~ ., data = trainPt1NoNAMedInt %>% select(-SEX, -
higheduc))
trainPt2TestMedLMPrediction <-
trainPt2TestMeanLMPrediction <- 0
trainPt2TestMedLMPrediction <- 0
trainPt2TestMeanLMPrediction <- predict.lm(trainPt1MeanModelLM, trainPt2TestNoNAM
ean)
trainPt2TestMedLMPrediction <- predict.glm(trainPt1MedModelLM, trainPt2TestNoNAMe
dInt %>% select(-SEX, -higheduc))
rmsePt2LM <- rmse(trainPt2$pstr, trainPt2TestMeanLMPrediction)
remove(rmsePt2LM )
rmsePt2MeanLM <- rmse(trainPt2$pstr, trainPt2TestMeanLMPrediction)
rmsePt2MedLM <- rmse(trainPt2$pstr, trainPt2TestMedLMPrediction)
trainMeanGLMPrediction <- 0
trainMeanGBMPrediction <- 0
trainMeanRFPrediction <- 0
trainMedGLMPrediction <- 0
trainMedGBMPrediction <- 0
trainMedRFPrediction <- 0
trainMeanGLMPrediction <- predict.glm(trainMeanModelGLM, trainNoNAMean)
trainMedGLMPrediction <- predict.glm(trainMedModelGLM, trainNoNAMean)
```

```
trainMeanGBMPrediction <- predict.gbm(trainMeanModelGBM, trainNoNAMean)
trainMedGBMPrediction <- predict.gbm(trainMedModelGBM, trainNoNAMean)
trainMedGLMPrediction <- predict.glm(trainMedModelGLM, trainNoNAMed)
trainMedGBMPrediction <- predict.gbm(trainMedModelGBM, trainNoNAMed)
trainMedGLMPrediction <- predict.glm(trainMedModelGLM, trainNoNAMedInt %>% select
(-SEX, -higheduc))
trainMedGBMPrediction <- predict.gbm(trainMedModelGBM, trainNoNAMedInt %>% select
(-SEX, -higheduc))
trainMedRFPrediction <- predict(trainMedModelRF, trainNoNAMedInt %>% select(-
SEX, -higheduc))
trainMeanRFPrediction <- predict(trainMeanModelRF, trainNoNAMean)
trainMeanGLMPrediction <- predict.glm(trainMeanModelGLM, testNoNAMean)
trainMeanGBMPrediction <- predict.glm(trainMeanModelGBM, testNoNAMean)
trainMeanGBMPrediction <- predict.gbm(trainMeanModelGBM, testNoNAMean)
trainMeanRFPrediction <- predict(trainMeanModelRF, testNoNAMean)
trainMeanGLMPrediction <- predict.glm(trainMeanModelGLM, testNoNAMean)
trainMeanGBMPrediction <- predict.gbm(trainMeanModelGBM, testNoNAMean)
trainMeanRFPrediction <- predict(trainMeanModelRF, testNoNAMean)
trainMedGLMPrediction <- predict.glm(trainMedModelGLM, testNoNAMedInt %>% select(
-SEX, -higheduc))
trainMedGBMPrediction <- predict.gbm(trainMedModelGBM, testNoNAMedInt %>% select(
-SEX, -higheduc))
trainMedRFPrediction <- predict(trainMedModelRF, testNoNAMedInt %>% select(-
SEX, -higheduc))
attemptMeanGLM <- data.frame(trainMeanGLMPrediction)
attemptMeanGBM <- data.frame(trainMeanGBMPrediction)
attemptMeanRF <- data.frame(trainMeanRFPrediction)
attemptMedRF <- data.frame(trainMedRFPrediction)
attemptMedGBM <- data.frame(trainMedGBMPrediction)
attemptMedGLM <- data.frame(trainMedGLMPrediction)
setwd("~/IUPUI_Spring_2021/CSCI 48900/competition/Submissions")
write.csv(attemptMeanGLM,"HampshireSubmissionMeanGLM.csv")
write.csv(attemptMeanGBM,"HampshireSubmissionMeanGBM.csv")
write.csv(attemptMeanRF,"HampshireSubmissionMeanRF.csv")
write.csv(attemptMedRF,"HampshireSubmissionMedRF.csv")
write.csv(attemptMedGBM,"HampshireSubmissionMedGBM.csv")
write.csv(attemptMedGLM,"HampshireSubmissionMedGLM.csv")
savehistory("~/IUPUI_Spring_2021/CSCI 48900/competition/Rhistory3_17_21.Rhistory"
)
```

```
library(gbm)
library(randomForest)
library(Metrics)
test <- read.csv(file="trainPt1.csv")
```

```
training <- read.csv(file="trainPt2.csv")
testAve = 0
trainAve = 0
temp1 <- 0
temp2 <- 0
for(i in 1:81){
  +     temp1 <- test %>% filter(!is.na(test[,i]))
  +     temp2 <- temp1[,i]
  +     testAve[i] <- mean(temp2)
  + }
for(i in 1:82){
  +     temp1 <- training %>% filter(!is.na(training[,i]))
  +     temp2 <- temp1[,i]
  +     trainAve[i] <- mean(temp2)
  + }
for(i in 1:82){
  +     training[,i] <- replace_na(training[,i], trainAve[i])
  + }
for(i in 1:81){
  +     test[,i] <- replace_na(test[,i], testAve[i])
  + }
modGLM <- glm(pstr ~. ,data = training)
predGLM <- predict.glm(modGLM, test)

modGBM <- gbm(formula = pstr ~ .,distribution = "gaussian", data = training %>% s
elect(-SEX, -higheduc))
predGBM <- predict.gbm(modGBM, test)

modRF <- randomForest(pstr ~ ., training)
predRF <- predict(modRF, test)

real <- read.csv(file = "trainPt1(true).csv")
real <- real$pstr

rmseGLM <- rmse(real, predGLM)
rmseGBM <- rmse(real, predGBM)
rmseRF <- rmse(real, predRF)

compTestAve = 0
for(i in 1:81){
  +     temp1 <- compTest %>% filter(!is.na(compTest[,i]))
  +     temp2 <- temp1[,i]
  +     compTestAve[i] <- mean(temp2)
  + }
for(i in 1:81){
```

```
  +     compTest[,i] <- replace_na(compTest[,i], compTestAve[i])
  + }

predCompGLM <- predict.glm(modGLM, compTest)
predCompGBM <- predict.gbm(modGBM, compTest)
predCompRF <- predict(modRF, compTest)

attemptGLM <- data.frame(predCompGLM)
attemptGBM <- data.frame(predCompGBM)
attemptRF <- data.frame(predCompRF)

write.csv(attemptGLM,"HampahireSubmissionGLM.csv")
write.csv(attemptGBM,"HampahireSubmissionGBM.csv")
write.csv(attemptRF,"HampahireSubmissionRF.csv")
```

```
TrainP1Filter[27,1] <- 41
TrainP1Filter[28,1] <- 46
TrainP1Filter[29,1] <- 48
TrainP1Filter[30,1] <- 49
TrainP1Filter[31,1] <- 51
TrainP1Filter[32,1] <- 52
TrainP1Filter[33,1] <- 58
TrainP1Filter[34,1] <- 59
TrainP1Filter[35,1] <- 61
TrainP1Filter[36,1] <- 65
TrainP1Filter[37,1] <- 70
TrainP1Filter[38,1] <- 75
TrainP1Filter[39,1] <- 77
TrainP1Filter[40,1] <- 78
TrainP1Filter[41,1] <- 80
TrainP1Filter[42,1] <- 81
TrainP1Filter[1,2] <- -0.002686665
TrainP1Filter[2,2] <- -0.0007619336
TrainP1Filter[3,2] <- -0.007150341
TrainP1Filter[4,2] <- 0.002312843
TrainP1Filter[5,2] <- -0.001057552
TrainP1Filter[6,2] <- -0.0001483416
TrainP1Filter[7,2] <- -5.065629e-06
TrainP1Filter[8,2] <- -0.0009113905
TrainP1Filter[9,2] <- -0.001170666
TrainP1Filter[10,2] <- -0.0003985442
```

```
TrainP1Filter[11,2] <- -7.237327e-05
TrainP1Filter[12,2] <- -4.757804e-05
TrainP1Filter[13,2] <- -0.0001141496
TrainP1Filter[14,2] <- -0.0004216312
TrainP1Filter[15,2] <- -0.0004006316
TrainP1Filter[16,2] <- -0.0007157724
TrainP1Filter[17,2] <- -0.0001453683
TrainP1Filter[18,2] <- -0.0001615187
TrainP1Filter[19,2] <- -0.0002685915
TrainP1Filter[20,2] <- -0.0013096
TrainP1Filter[21,2] <- -0.0006569393
TrainP1Filter[22,2] <- 1.942733e-05
TrainP1Filter[23,2] <- -2.666819e-05
TrainP1Filter[24,2] <- -2.593712e-05
TrainP1Filter[25,2] <- -0.01340357
TrainP1Filter[26,2] <- -8.639328e-05
TrainP1Filter[27,2] <- -0.0004790353
TrainP1Filter[28,2] <- -0.001482289
TrainP1Filter[29,2] <- -0.000404667
TrainP1Filter[30,2] <- -0.001817483
TrainP1Filter[31,2] <- -9.892795e-05
TrainP1Filter[32,2] <- -0.00261045
TrainP1Filter[33,2] <- -0.001990104
TrainP1Filter[34,2] <- -1.217378e-05
TrainP1Filter[35,2] <- -0.0004078777
TrainP1Filter[36,2] <- -0.001552952
TrainP1Filter[37,2] <- -0.0005711461
TrainP1Filter[38,2] <- -0.00493374
TrainP1Filter[39,2] <- -0.002517518
TrainP1Filter[40,2] <- -0.00177647
TrainP1Filter[41,2] <- -0.0003281012
TrainP1Filter[42,2] <- -0.0004154705
View(TrainP1Filter)
TrainPt1Testing <- 0
TrainPt1TestingMeanModel <- 0
TrainPt2TestingMeanPrediction <- 0
TrainPt1Testing <- trainPt1NoNAMean
TrainPt1Testing <- TrainPt1Testing %>% select(39, 3, 75, 1, 52, 77, 58, 49, 78, 6
5, 46, 32, 12, 5)
TrainPt1TestingMeanModel <- glm(pstr ~ .,data = TrainPt1Testing)
TrainPt2TestingMeanPrediction <- predict.glm(TrainPt1TestingMeanModel, trainPt2Te
stNoNAMean)
rmsePt2Testing <- rmse(trainPt2$pstr, TrainPt2TestingMeanPrediction)
TrainPt1Testing <- 0
TrainPt1TestingMeanModel <- 0
```

```
TrainPt2TestingMeanPrediction <- 0
TrainPt1Testing <- trainPt1NoNAMean
TrainPt1Testing <- TrainPt1Testing %>% select(39, 3, 75, 1, 52, 77, 58, 49, 78, 6
5, 46, 32, 12, 5, 82)
TrainPt1TestingMeanModel <- glm(pstr ~ .,data = TrainPt1Testing)
TrainPt2TestingMeanPrediction <- predict.glm(TrainPt1TestingMeanModel, trainPt2Te
stNoNAMean)
rmsePt2Testing <- rmse(trainPt2$pstr, TrainPt2TestingMeanPrediction)
TrainPt1Testing <- 0
TrainPt1TestingMeanModel <- 0
TrainPt2TestingMeanPrediction <- 0
TrainPt1Testing <- trainPt1NoNAMean
TrainPt1Testing <- TrainPt1Testing %>% select(39, 3, 75, 1, 52, 77, 58, 49, 78, 6
5, 46, 32, 12, 5, 11, 82)
TrainPt1TestingMeanModel <- glm(pstr ~ .,data = TrainPt1Testing)
TrainPt2TestingMeanPrediction <- predict.glm(TrainPt1TestingMeanModel, trainPt2Te
stNoNAMean)
rmsePt2Testing <- rmse(trainPt2$pstr, TrainPt2TestingMeanPrediction)
TrainPt1Testing <- 0
TrainPt1TestingMeanModel <- 0
TrainPt2TestingMeanPrediction <- 0
TrainPt1Testing <- trainPt1NoNAMean
TrainPt1Testing <- TrainPt1Testing %>% select(39, 1, 3, 5, 11, 12, 32, 46, 49, 52
, 58, 65, 75, 77, 78, 82)
TrainPt1TestingMeanModel <- glm(pstr ~ .,data = TrainPt1Testing)
TrainPt2TestingMeanPrediction <- predict.glm(TrainPt1TestingMeanModel, trainPt2Te
stNoNAMean)
rmsePt2Testing <- rmse(trainPt2$pstr, TrainPt2TestingMeanPrediction)
trainFiltered2 <- 0
trainFiltered2MeanModel <- 0
trainFiltered2MeanPrediction <- 0
trainFiltered2 <- trainNoNAMean
trainFiltered2 <- trainFiltered2 %>% select(39, 1, 3, 5, 11, 12, 32, 46, 49, 52,
58, 65, 75, 77, 78, 82)
trainFiltered2MeanModel <- glm(pstr ~ .,data = trainFiltered2)
trainFiltered2MeanPrediction <- predict.glm(trainFiltered2MeanModel, testNoNAMean
)
rmseFiltered2 <- rmse(trainNoNAMean$pstr, trainFiltered2MeanPrediction)
trainFiltered2 <- 0
trainFiltered2MeanModel <- 0
trainFiltered2MeanPrediction <- 0
trainFiltered2 <- trainNoNAMean
trainFiltered2 <- trainFiltered2 %>% select(39, 1, 3, 5, 11, 12, 32, 46, 49, 52,
58, 65, 75, 77, 78, 82)
trainFiltered2MeanModel <- glm(pstr ~ .,data = trainFiltered2)
```

```
trainFiltered2MeanPrediction <- predict.glm(trainFiltered2MeanModel, testNoNAMean
)
rmseFiltered2 <- rmse(trainNoNAMean$pstr, trainFiltered2MeanPrediction)
attemptFilteredG2 <- data.frame(trainFiltered2MeanPrediction)
write.csv(attemptFilteredG2 , "HampshireSubmissionFilteredGLM2")
trainFiltered2 <- 0
trainFiltered2MeanModel <- 0
trainFiltered2MeanPrediction <- 0
trainFiltered2 <- trainNoNAMean
trainFiltered2 <- trainFiltered2 %>% select(39, 1, 3, 5, 11, 12, 32, 46, 49, 52,
58, 65, 75, 77, 78, 82)
trainFiltered2MeanModel <- glm(pstr ~ .,data = trainFiltered2)
trainFiltered2MeanPrediction <- predict.glm(trainFiltered2MeanModel, testNoNAMean
)
rmseFiltered2 <- rmse(trainNoNAMean$pstr, trainFiltered2MeanPrediction)
trainFiltered3 <- 0
trainFiltered3MeanModel <- 0
trainFiltered3MeanPrediction <- 0
trainFiltered3 <- trainNoNAMean
trainFiltered3 <- trainFiltered3 %>% select(39, 1, 3, 5, 11, 12, 32, 46, 49, 52,
58, 65, 75, 77, 78, 82)
trainFiltered3MeanModel <- randomForest(pstr ~ .,data = trainFiltered3)
trainFiltered3MeanPrediction <- predict.glm(trainFiltered3MeanModel, testNoNAMean
)
trainFiltered3 <- 0
trainFiltered3MedModel <- 0
trainFiltered3MedPrediction <- 0
trainFiltered3 <- trainNoNAMedInt
trainFiltered3 <- trainFiltered3 %>% select(39, 1, 3, 5, 11, 12, 32, 46, 49, 52,
58, 65, 75, 77, 78, 82)
trainFiltered3MedModel <- randomForest(pstr ~ .,data = trainFiltered3)
trainFiltered3MedPrediction <- predict.glm(trainFiltered2MeanModel, testNoNAMedIn
t)
trainFiltered3 <- 0
trainFiltered3MedModel <- 0
trainFiltered3MedPrediction <- 0
trainFiltered3 <- trainNoNAMedInt
trainFiltered3 <- trainFiltered3 %>% select(39, 1, 5, 11, 12, 32, 46, 49, 52, 58,
 65, 75, 77, 78, 82)
trainFiltered3MedModel <- randomForest(pstr ~ .,data = trainFiltered3)
trainFiltered3MedPrediction <- predict.glm(trainFiltered2MeanModel, testNoNAMedIn
t)
trainFiltered3MedPrediction <- predict(trainFiltered3MeanModel, testNoNAMedInt)
trainFiltered3 <- 0
trainFiltered3MedModel <- 0
```

```
trainFiltered3MedPrediction <- 0
trainFiltered3 <- trainNoNAMedInt
trainFiltered3 <- trainFiltered3 %>% select(39, 1, 5, 11, 12, 32, 46, 49, 52, 58,
 65, 75, 77, 78, 82)
trainFiltered3MedModel <- randomForest(pstr ~ .,data = trainFiltered3)
trainFiltered3MedPrediction <- predict(trainFiltered3MedModel, testNoNAMedInt)
attemptFiltered3RF <- data.frame(trainFiltered3MedPrediction)
setwd("~/IUPUI_Spring_2021/CSCI 48900/competition/Submissions")
write.csv(attemptFiltered3RF , "HampshireSubmissionFilteredRF2.csv")
```