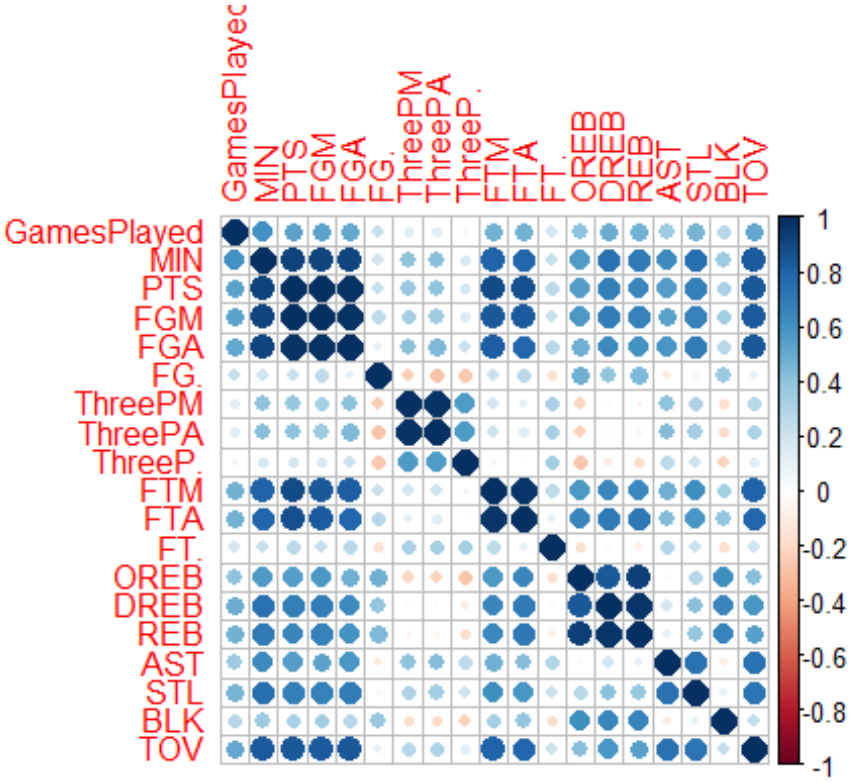
Homework5

Ritwik Bhriguvanshi

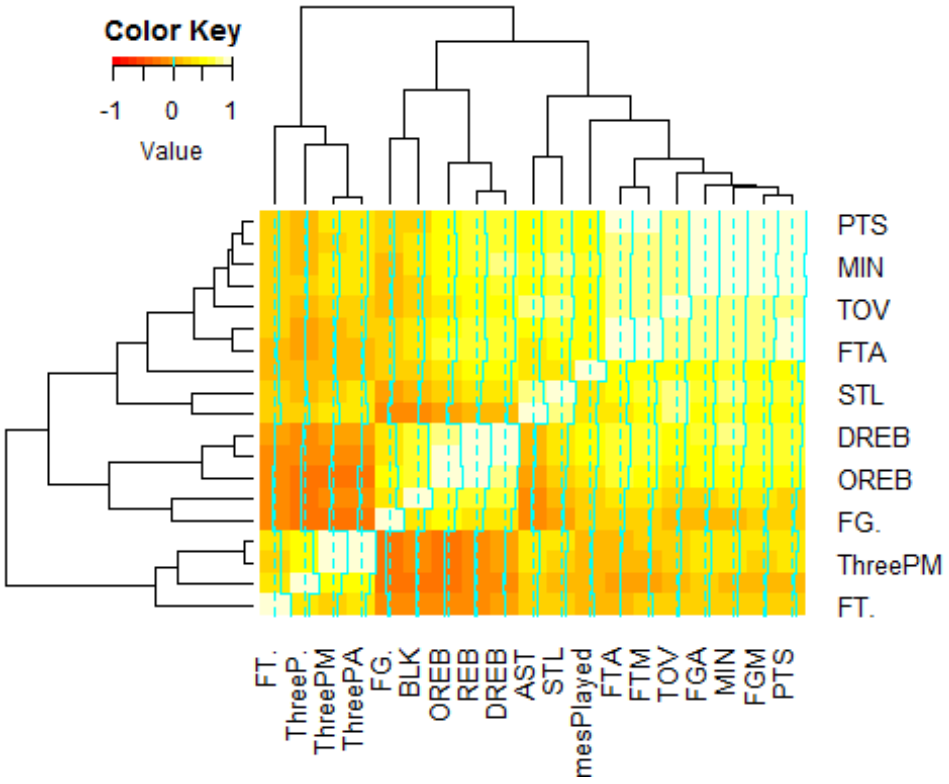
R Markdown

```
## Loading in the dataset

data <- read.csv("nba_merged_subset.csv", header = TRUE)
dim(data)

## [1] 1230   22

## Convert the dataset into a table to check the number of entries
player_names <- table(data$Name)

## Choose players whose name appears only ONCE
player_names_ones <- names(player_names)[(player_names==1)==TRUE]

## Subset the selected players back into the original datset
data <- data[which(data$Name %in% player_names_ones),]

## Number of missing values in the dataset
sum(is.na(data))

## [1] 6

## Column names in which missing values exists
colnames(data)[colSums(is.na(data)) > 0]  ## Column "ThreeP." consists missin
g values

## [1] "ThreeP."

## Rows and Column indices of Missing Values
which(is.na(data), arr.ind=TRUE)

##       row col
## 578   505  10
## 600   527  10
## 601   528  10
## 724   649  10
## 951   848  10
## 1161 1054  10

## Impute Missing Values with Median
data$ThreeP.[which(is.na(data$ThreeP.))] <- median(data$ThreeP., na.rm = T)

## Random Sampling data and splitting data into train and test after using Se
t Seed
```

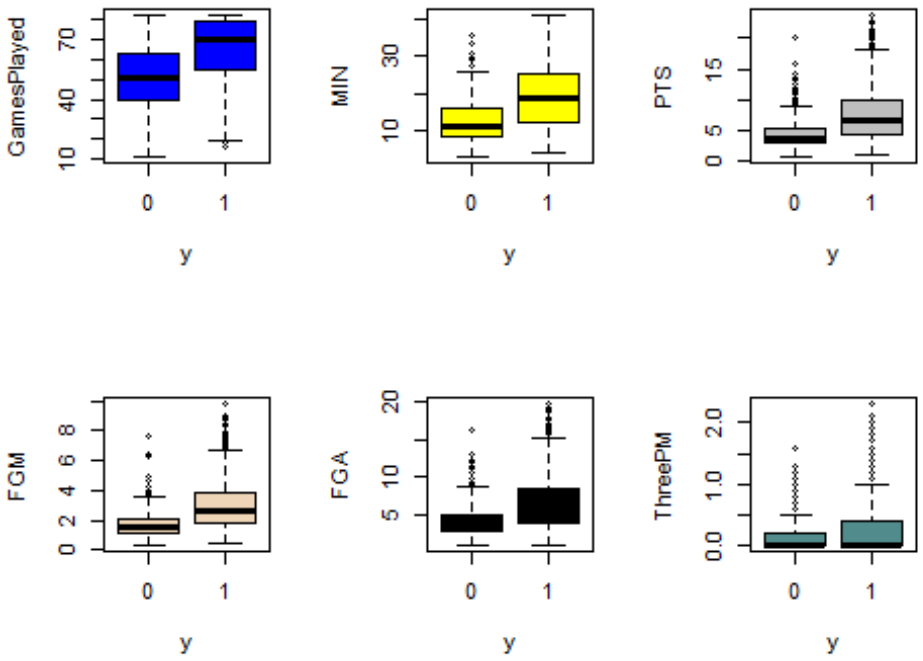
```r
set.seed(540)
index <- sample(1:nrow(data),0.7*nrow(data),replace = FALSE)

train <- data[index,]
dim(train)

## [1] 784  22

test <- data[-index,]
dim(test)

## [1] 337  22

## Basic Visualization Plots with all predictor variables
## Correlation Matrix
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.6.3

## corrplot 0.84 loaded

library(gplots)

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

cormatrix <- cor(train[,-c(1,21,22)])  ## Exclude- name, year drafted and y-v
ariable
corrplot(cormatrix)
```



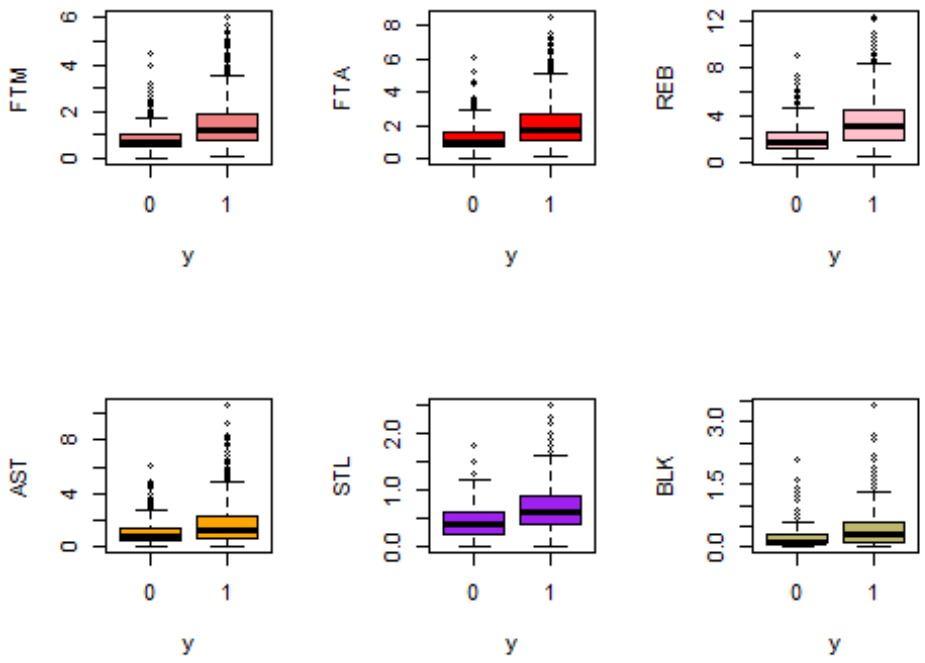

```
## HeatMap is also a good visualization tool here
heatmap.2(cor(train[,-c(1,21,22)]), density.info = "none")
```



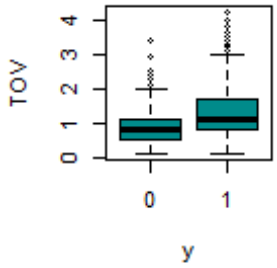
```r
par(mfrow=c(2,3))

boxplot (GamesPlayed ~ y,data = data, ylab = "GamesPlayed", col = "blue")
boxplot(MIN ~ y,data = data, ylab = "MIN", col = "yellow")
boxplot(PTS ~ y,data = data, ylab = "PTS", col = "gray")
boxplot(FGM ~ y,data = data, ylab = "FGM", col = "bisque2")
boxplot(FGA ~ y,data = data, ylab = "FGA", col = "black")
boxplot(ThreePM ~ y,data = data, ylab = "ThreePM", col = "darkslategray4")
```




```r
boxplot(FTM ~ y,data = data, ylab = "FTM", col = "lightcoral")
boxplot(FTA ~ y,data = data, ylab = "FTA", col = "red")
boxplot(REB ~ y,data = data, ylab = "REB", col = "pink")
boxplot(AST ~ y,data = data, ylab = "AST", col = "orange")
boxplot(STL ~ y,data = data, ylab = "STL", col = "purple")
boxplot(BLK ~ y,data = data, ylab = "BLK", col = "darkkhaki")
```




```
boxplot(TOV ~ y,data = data, ylab = "TOV", col = "cyan4")
```



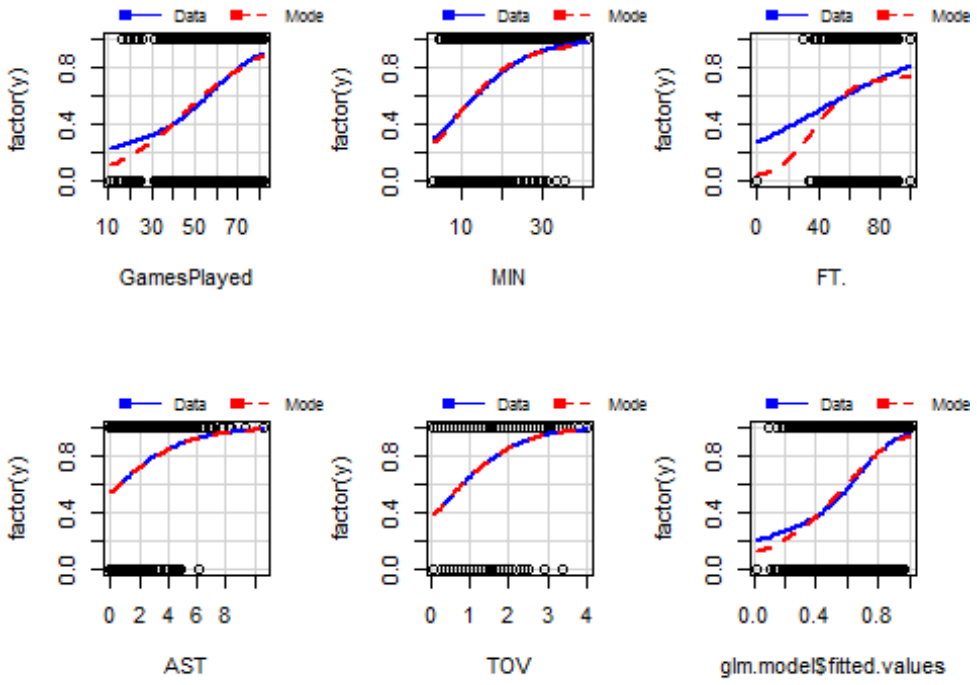
Response:

Observing the corrplot and heatmap, we find that great number of predictor variables have high correlation between themselves. We can observe that MIN, PTS, FGM, FGA have dark blue cicrles when compared with each other depicting high correlation. It does not come as a surpise as these variables would be essential for players spanning their career over 5 years. We can confirm your understanding by plotting boxplot of the response variable with each of the predictor variable separately. The boxplots will depict a higher spread of value for 'y' variables coded as 1(career span is greater than 5 years) as compared to players playing for less than 5 years. We can infer from the boxplots that on an average, rookies who played more games in their rookie year are more likely to play 5 or more years in the league. Similar positive trendscan be seen for average minutes played, points scored per game, average field goals made, average fieldgoals attempted, average three points made average free throws made, average free throws made, average rebounds, average assists, average steals, average blocks, and average turnovers.

```
library(alr3)

## Loading required package: car

## Loading required package: carData

glm.model <- glm(factor(y) ~., data = train[,-c(1,22)], family = "binomial")
summary(glm.model)

##
## Call:
## glm(formula = factor(y) ~ ., family = "binomial", data = train[,
##     -c(1, 22)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5181  -0.9033   0.4404   0.7976   2.1345
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.0052047  1.7321065  -2.312  0.02076 *
## GamesPlayed  0.0374850  0.0068902   5.440 5.32e-08 ***
## MIN         -0.0864351  0.0485433  -1.781  0.07498 .
## PTS          0.2205278  1.2300920   0.179  0.85772
## FGM          0.6344124  2.4329577   0.261  0.79428
## FGA         -0.3691325  0.3696460  -0.999  0.31798
## FG.          0.0007155  0.0307291   0.023  0.98142
## ThreePM      0.0051049  1.8694895   0.003  0.99782
## ThreePA      0.2194958  0.6004086   0.366  0.71468
## ThreeP.     -0.0013202  0.0065675  -0.201  0.84068
## FTM         -0.6374580  1.4619737  -0.436  0.66282
## FTA          0.5703069  0.7252111   0.786  0.43163
## FT.          0.0238625  0.0142031   1.680  0.09294 .
## OREB        -0.0868358  1.7435930  -0.050  0.96028
## DREB        -0.3885932  1.7170476  -0.226  0.82096
## REB          0.7006062  1.7134888   0.409  0.68263
```

```
## AST          0.5681756  0.1752441   3.242  0.00119 **
## STL          0.1281201  0.4610941   0.278  0.78112
## BLK          0.6313444  0.3882680   1.626  0.10394
## TOV         -1.0457556  0.4134013  -2.530  0.01142 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.37  on 783  degrees of freedom
## Residual deviance: 789.81  on 764  degrees of freedom
## AIC: 829.81
##
## Number of Fisher Scoring iterations: 5
```

```r
## For Model Diagnostics
attach(train)
par(mfrow=c(2,3))
mmp(glm.model,GamesPlayed);
mmp(glm.model,MIN);
mmp(glm.model,FT.);
mmp(glm.model,AST);
mmp(glm.model,TOV);
mmp(glm.model,glm.model$fitted.values);
```

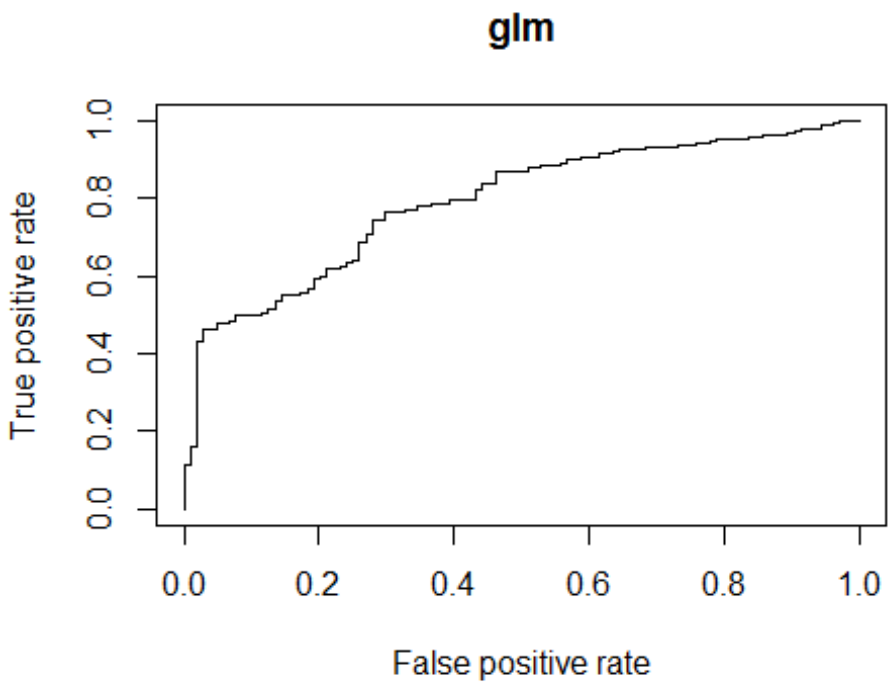



```r
detach(train)
```

Response:

Significant predictors from the model are **'GamesPlayed'**, **'MIN'**, **'FT.'**, **'AST'**, and **'TOV'** for a rookie to have a career span greater than 5 years. For these significant predictor variables, the linear systematic component is adequate as reported and depicted in the marginal model plots for these variables having very small p-values (for 'MIN' and 'FT', it is a little above 5% threshold) For 'FT.' there is a little deviation but nothing major, hence the plots show sufficient evidence of true validity.

```
library(ROCR)

## Warning: package 'ROCR' was built under R version 3.6.3

pred <- predict(glm.model, newdata = test[,-c(1,22)], type = "response")
pred.glm <- prediction(pred,test$y)
performance.glm <- performance(pred.glm, measure = "tpr", x.measure = "fpr")

par(mfrow=c(1,1))
plot(performance.glm,main = "glm")
```




```
(misclassification.rate <- mean(round(pred)!= test$y))

## [1] 0.2670623

## AUC of the model
```

```r
auc.glmnet <- performance(pred.glm, measure = "auc") # compute the auc
auc.glmnet@y.values[[1]] # extract auc of the model
```

```
## [1] 0.791928
```

```r
## By defualt, R reports the coefficients and p-values from the Wald's test
## However we build another model here

model <- glm(factor(y) ~ .,data = train[,-c(1,22)], family = "binomial")
(summary(model))
```

```
##
## Call:
## glm(formula = factor(y) ~ ., family = "binomial", data = train[,
##     -c(1, 22)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5181  -0.9033   0.4404   0.7976   2.1345
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.0052047  1.7321065  -2.312  0.02076 *
## GamesPlayed  0.0374850  0.0068902   5.440 5.32e-08 ***
## MIN         -0.0864351  0.0485433  -1.781  0.07498 .
## PTS          0.2205278  1.2300920   0.179  0.85772
## FGM          0.6344124  2.4329577   0.261  0.79428
## FGA         -0.3691325  0.3696460  -0.999  0.31798
## FG.          0.0007155  0.0307291   0.023  0.98142
## ThreePM      0.0051049  1.8694895   0.003  0.99782
## ThreePA      0.2194958  0.6004086   0.366  0.71468
## ThreeP.     -0.0013202  0.0065675  -0.201  0.84068
## FTM         -0.6374580  1.4619737  -0.436  0.66282
## FTA          0.5703069  0.7252111   0.786  0.43163
## FT.          0.0238625  0.0142031   1.680  0.09294 .
## OREB        -0.0868358  1.7435930  -0.050  0.96028
## DREB        -0.3885932  1.7170476  -0.226  0.82096
## REB          0.7006062  1.7134888   0.409  0.68263
## AST          0.5681756  0.1752441   3.242  0.00119 **
## STL          0.1281201  0.4610941   0.278  0.78112
## BLK          0.6313444  0.3882680   1.626  0.10394
## TOV         -1.0457556  0.4134013  -2.530  0.01142 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.37  on 783  degrees of freedom
## Residual deviance: 789.81  on 764  degrees of freedom
## AIC: 829.81
```

```
##
## Number of Fisher Scoring iterations: 5
```

Response:

As reported earlier, the significant variables are 'GamesPlayed', 'AST' and 'TOV' for 5 year longevity of a rookie in the league by Wald's test. 'FT.' and 'MIN' are just above the 5% threshold.

```
coef(summary(model))
```

```
##                   Estimate  Std. Error      z value     Pr(>|z|)
## (Intercept) -4.0052046753 1.732106546 -2.312331585 2.075942e-02
## GamesPlayed  0.0374850346 0.006890183  5.440353610 5.317492e-08
## MIN         -0.0864351170 0.048543269 -1.780578822 7.498128e-02
## PTS          0.2205277848 1.230092003  0.179277472 8.577198e-01
## FGM          0.6344124178 2.432957711  0.260757684 7.942794e-01
## FGA         -0.3691325169 0.369645984 -0.998610921 3.179832e-01
## FG.          0.0007155496 0.030729102  0.023285732 9.814224e-01
## ThreePM      0.0051049192 1.869489545  0.002730649 9.978213e-01
## ThreePA      0.2194958109 0.600408584  0.365577403 7.146804e-01
## ThreeP.     -0.0013202297 0.006567508 -0.201024445 8.406795e-01
## FTM         -0.6374579594 1.461973733 -0.436025590 6.628182e-01
## FTA          0.5703069029 0.725211089  0.786401244 4.316325e-01
## FT.          0.0238624945 0.014203059  1.680095426 9.293875e-02
## OREB        -0.0868357655 1.743593028 -0.049802772 9.602796e-01
## DREB        -0.3885932498 1.717047642 -0.226314774 8.209566e-01
## REB          0.7006062462 1.713488781  0.408877055 6.826299e-01
## AST          0.5681756395 0.175244132  3.242194944 1.186129e-03
## STL          0.1281201011 0.461094065  0.277861093 7.811190e-01
## BLK          0.6313444214 0.388268005  1.626053171 1.039383e-01
## TOV         -1.0457556335 0.413401269 -2.529638181 1.141802e-02
```

```
confint(model)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %       97.5 %
## (Intercept) -7.466612930 -0.673116617
## GamesPlayed  0.024115894  0.051162435
## MIN         -0.182470172  0.008229605
## PTS         -2.186622893  2.641737542
## FGM         -4.146201800  5.403840756
## FGA         -1.100026826  0.352719869
## FG.         -0.059684109  0.061106320
## ThreePM     -3.657415398  3.683183271
## ThreePA     -0.961882874  1.396863228
## ThreeP.     -0.014056655  0.011830191
## FTM         -3.512442914  2.226251129
## FTA         -0.832451984  2.012804224
## FT.         -0.003477483  0.052169994
```

```
## OREB        -3.511852116  3.333891024
## DREB        -3.761201121  2.980370179
## REB         -2.659173553  4.068395376
## AST          0.233709892  0.921316753
## STL         -0.768257426  1.043069336
## BLK         -0.101704685  1.423638506
## TOV         -1.862459258 -0.239484660

## Coefficient estimates along with its confidence interval
(estimates <- (cbind(Estimate = coef(model), confint(model))))

## Waiting for profiling to be done...

##                  Estimate        2.5 %       97.5 %
## (Intercept) -4.0052046753 -7.466612930 -0.673116617
## GamesPlayed  0.0374850346  0.024115894  0.051162435
## MIN         -0.0864351170 -0.182470172  0.008229605
## PTS          0.2205277848 -2.186622893  2.641737542
## FGM          0.6344124178 -4.146201800  5.403840756
## FGA         -0.3691325169 -1.100026826  0.352719869
## FG.          0.0007155496 -0.059684109  0.061106320
## ThreePM      0.0051049192 -3.657415398  3.683183271
## ThreePA      0.2194958109 -0.961882874  1.396863228
## ThreeP.     -0.0013202297 -0.014056655  0.011830191
## FTM         -0.6374579594 -3.512442914  2.226251129
## FTA          0.5703069029 -0.832451984  2.012804224
## FT.          0.0238624945 -0.003477483  0.052169994
## OREB        -0.0868357655 -3.511852116  3.333891024
## DREB        -0.3885932498 -3.761201121  2.980370179
## REB          0.7006062462 -2.659173553  4.068395376
## AST          0.5681756395  0.233709892  0.921316753
## STL          0.1281201011 -0.768257426  1.043069336
## BLK          0.6313444214 -0.101704685  1.423638506
## TOV         -1.0457556335 -1.862459258 -0.239484660

## Exponentiating five significant coefficients from the model

## Exponentiating the coefficient for GamesPlayed
(exp(estimates[2,1]))

## [1] 1.038196

## Exponentiating the coefficient for MIN
(exp(estimates[3,1]))

## [1] 0.9171951

## Exponentiating the coefficient for FT.
(exp(estimates[13,1]))

## [1] 1.024149
```

```
## Exponentiating the coefficient for AST
(exp(estimates[17,1]))

## [1] 1.765044

## Exponentiating the coefficient for TOV
(exp(estimates[20,1]))

## [1] 0.3514262
```

Response:

GamesPlayed and **Assist** are the two most important significant predictors estimated by thhe model.

Keeping all rookie statistics constant, every additional game played is associated with a 3.8196% increase in the odds of the rookie's 5 or more years longevity in the league.

Similarly, keeping all other rookie statistics constant, every additional assist per game is associated with a 76.5044% increase in the odds of the rookie's 5 or more years longevity in the league.

```
library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-18

library(ROCR)

modelCLASS <- cv.glmnet(x=as.matrix(train[,-c(1,21,22)]),
                        y=as.vector(train$y),
                        family = "binomial",
                        alpha = 1,
                        type.measure = "class")

modelAUC <- cv.glmnet(x=as.matrix(train[,-c(1,21,22)]),
                        y=as.vector(train$y),
                        family = "binomial",
                        alpha = 1,
                        type.measure = "auc")

# Predicted Values

pred_model_CLASS_MIN <- predict(modelCLASS,
                         newx = as.matrix(test[,-c(1,21,22)]),
                         s = "lambda.min",
                         type = "response")
head(pred_model_CLASS_MIN)
```

```
##            1
## 2  0.7457965
## 4  0.5717496
## 5  0.6537273
## 8  0.5185752
## 12 0.4647519
## 13 0.8842547
```

```
pred_model_CLASS_1SE <- predict(modelCLASS,
                            newx = as.matrix(test[,-c(1,21,22)]),
                            s = "lambda.1se",
                            type = "response")
head(pred_model_CLASS_1SE)
```

```
##            1
## 2  0.7480043
## 4  0.5258986
## 5  0.6730347
## 8  0.5406910
## 12 0.5465855
## 13 0.8307453
```

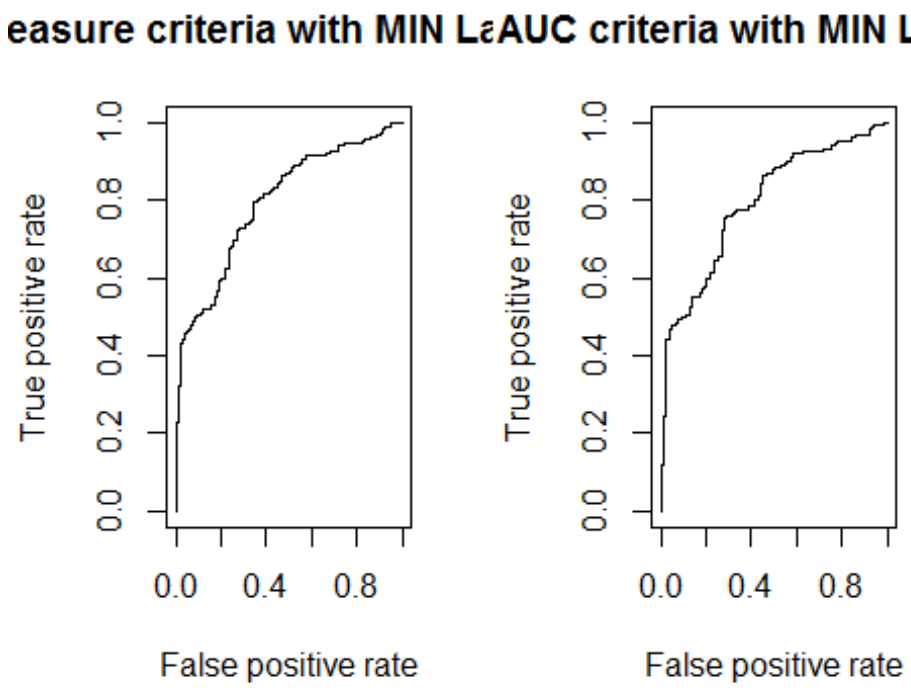
```
pred_model_AUC_MIN <- predict(modelAUC,
                            newx = as.matrix(test[,-c(1,21,22)]),
                            s = "lambda.min",
                            type = "response")
head(pred_model_AUC_MIN)
```

```
##            1
## 2  0.7185235
## 4  0.6361019
## 5  0.6911628
## 8  0.4548094
## 12 0.4185803
## 13 0.8818572
```

```
pred_model_AUC_1SE <- predict(modelAUC,
                            newx = as.matrix(test[,-c(1,21,22)]),
                            s = "lambda.1se",
                            type = "response")
head(pred_model_AUC_1SE)
```

```
##            1
## 2  0.7148733
## 4  0.6004441
## 5  0.6571537
## 8  0.6321731
## 12 0.5954129
## 13 0.7578463
```

```r
## ROC CURVES

# Prediction
pr.model_CLASS_MIN <- prediction(pred_model_CLASS_MIN,test$y)
pr.model_CLASS_1SE <- prediction(pred_model_CLASS_1SE,test$y)
pr.model_AUC_MIN <- prediction(pred_model_AUC_MIN,test$y)
pr.model_AUC_1SE <- prediction(pred_model_AUC_1SE,test$y)

# Performance
prf.model_CLASS_MIN <- performance(pr.model_CLASS_MIN, measure ="tpr", x.meas
ure ="fpr")
prf.model_CLASS_1SE <- performance(pr.model_CLASS_1SE, measure ="tpr", x.meas
ure ="fpr")
prf.model_AUC_MIN <- performance(pr.model_AUC_MIN, measure ="tpr", x.measure
="fpr")
prf.model_AUC_1SE <- performance(pr.model_AUC_1SE, measure ="tpr", x.measure
="fpr")

par(mfrow=c(1,2))
plot(prf.model_CLASS_MIN, main = "Measure criteria with MIN Lambda")
plot(prf.model_AUC_MIN, main = "AUC criteria with MIN Lambda")
```




```r
## MISSCLASSIFICATION RATE
c(MCR_MEASURE_MIN = mean(round(pred_model_CLASS_MIN)!= test$y),
  MCR_MEASURE_1SE = mean(round(pred_model_CLASS_1SE)!= test$y),
```

```
  MCR_AUC_MIN = mean(round(pred_model_AUC_MIN)!= test$y),
  MCR_AUC_1SE = mean(round(pred_model_AUC_1SE)!= test$y))
```

```
## MCR_MEASURE_MIN MCR_MEASURE_1SE     MCR_AUC_MIN     MCR_AUC_1SE
##       0.2492582       0.2581602       0.2640950       0.2789318
```

```
## AUC

AUC_MEASURE_MIN <- performance(pr.model_CLASS_MIN, measure = "auc")
AUC_MEASURE_1SE <- performance(pr.model_CLASS_1SE, measure = "auc")
AUC_AUC_MIN <- performance(pr.model_AUC_MIN, measure = "auc")
AUC_AUC_1SE <- performance(pr.model_AUC_1SE, measure = "auc")

c(AUC_MEASURE_MIN = AUC_MEASURE_MIN@y.values[[1]],
AUC_MEASURE_1SE = AUC_MEASURE_1SE@y.values[[1]],
AUC_AUC_MIN = AUC_AUC_MIN@y.values[[1]],
AUC_AUC_1SE = AUC_AUC_1SE@y.values[[1]])
```

```
## AUC_MEASURE_MIN AUC_MEASURE_1SE     AUC_AUC_MIN     AUC_AUC_1SE
##       0.7965087       0.7887504       0.7945279       0.7824364
```

The missclassification rate and AUC are reported by the R output above.

```
coefficients = cbind(as.matrix(coef(modelCLASS, s="lambda.min"))[-1,],
                     as.matrix(coef(modelCLASS, s="lambda.1se"))[-1,],
                     as.matrix(coef(modelAUC, s="lambda.min"))[-1,],
                     as.matrix(coef(modelAUC, s="lambda.1se"))[-1,])

colnames(coefficients) <- c("LASSO_MEASURE_MIN",
                            "LASSO_MEASURE_1SE",
                            "LASSO_AUC_MIN",
                            "LASSO_AUC_1SE")
(round(coefficients,4))
```

```
##             LASSO_MEASURE_MIN LASSO_MEASURE_1SE LASSO_AUC_MIN
## GamesPlayed            0.0335            0.0321        0.0371
## MIN                    0.0000            0.0015       -0.0809
## PTS                    0.0162            0.0533        0.0000
## FGM                    0.0000            0.0000        0.5802
## FGA                    0.0000            0.0000       -0.1545
## FG.                    0.0294            0.0140        0.0163
## ThreePM                0.2597            0.0000        0.3326
## ThreePA                0.0000            0.0000        0.1389
## ThreeP.                0.0000            0.0000       -0.0007
## FTM                    0.0000            0.0000       -0.0614
## FTA                    0.1015            0.0000        0.3080
## FT.                    0.0163            0.0027        0.0196
## OREB                   0.3378            0.0000        0.3343
## DREB                   0.0000            0.0120        0.0000
## REB                    0.1371            0.1688        0.2983
## AST                    0.2460            0.0000        0.5438
```

```
## STL                    0.0000            0.0000        0.0959
## BLK                    0.4714            0.0138        0.6246
## TOV                   -0.3747            0.0000       -1.0006
##              LASSO_AUC_1SE
## GamesPlayed         0.0201
## MIN                 0.0107
## PTS                 0.0000
## FGM                 0.0000
## FGA                 0.0000
## FG.                 0.0000
## ThreePM             0.0000
## ThreePA             0.0000
## ThreeP.             0.0000
## FTM                 0.0000
## FTA                 0.0000
## FT.                 0.0000
## OREB                0.0000
## DREB                0.0000
## REB                 0.0371
## AST                 0.0000
## STL                 0.0000
## BLK                 0.0000
## TOV                 0.0000
```

Response:

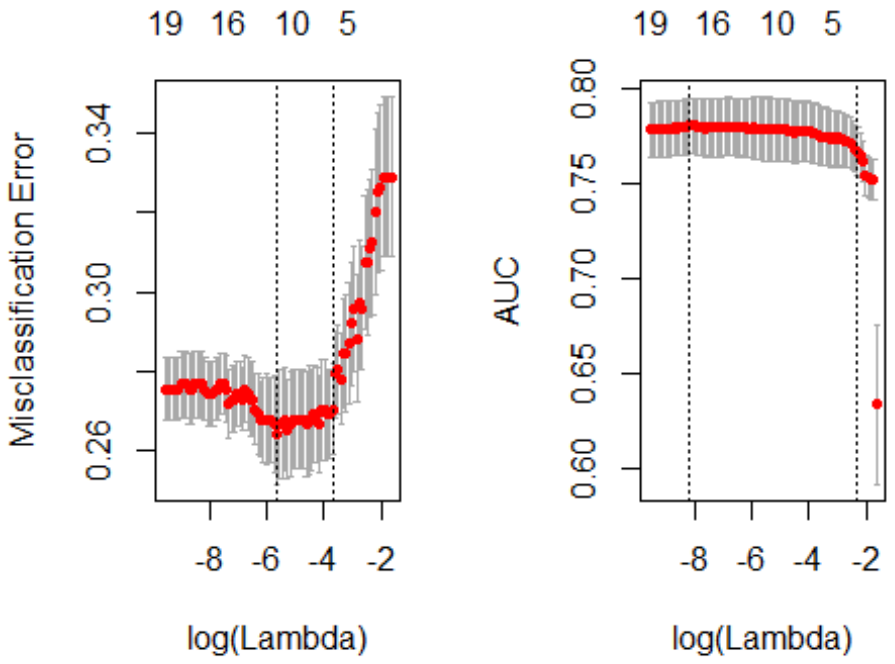
I have combined the coeffificent output of logistic-lasso regression model with bothe the measure and AUC criterion respective to lambda.min and lambda.1se.

We observe that **GamesPlayed** and **REB** have a non-zero estimated coefficient across all the models. Apart from these two predictors, other coefficients with non-zero estimates are across majority of the models are **MIN**, **PTS**, **FG.**. By majoirty, I mean non-zero coefficients for minimum of three out of four models.

There are other predictors present that have non-zero coefficients, but depends on the model on interest.

The prediction performance measured by the AUC for different models is around the same range, so there's not much distinction in model performance for the four different models listed above enabling to point out one single best model.

```
par(mfrow=c(1,2))
plot(modelCLASS)
plot(modelAUC)
```



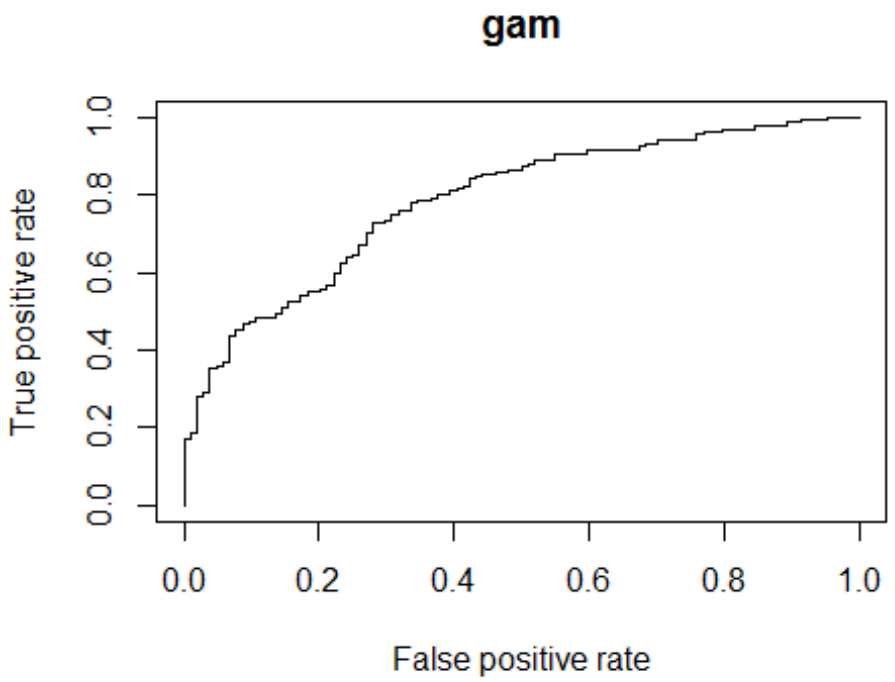

```
library(mgcv)

## Loading required package: nlme

## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.

gam.model <- gam(y ~ s(GamesPlayed) + s(MIN) + s(PTS) + s(FGM) + s(FGA) +
                 s(FG.) + s(ThreePM) + s(ThreePA) + s(ThreeP.) + s(FTM) +
                 s(FTA) + s(FT.) + s(OREB) + s(DREB) + s(REB) + s(AST) +
                 s(STL) + s(BLK) + s(TOV), data = train, family = "binomial")

summary(gam.model)

##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ s(GamesPlayed) + s(MIN) + s(PTS) + s(FGM) + s(FGA) + s(FG.) +
##     s(ThreePM) + s(ThreePA) + s(ThreeP.) + s(FTM) + s(FTA) +
##     s(FT.) + s(OREB) + s(DREB) + s(REB) + s(AST) + s(STL) + s(BLK) +
##     s(TOV)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.0794     0.1138   9.486   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                   edf Ref.df Chi.sq  p-value
## s(GamesPlayed) 1.532  1.901 29.047 4.27e-07 ***
## s(MIN)         1.842  2.344  5.012  0.09776 .
## s(PTS)         1.000  1.000  0.001  0.97890
## s(FGM)         1.000  1.000  0.275  0.60012
## s(FGA)         1.000  1.000  1.594  0.20673
## s(FG.)         6.633  7.622 13.415  0.10275
## s(ThreePM)     3.376  4.162  4.436  0.41232
## s(ThreePA)     1.000  1.000  1.259  0.26197
## s(ThreeP.)     4.953  6.004  6.495  0.37448
## s(FTM)         1.000  1.000  0.221  0.63845
## s(FTA)         1.000  1.000  1.150  0.28348
## s(FT.)         1.719  2.192  4.629  0.13387
## s(OREB)        1.000  1.000  0.001  0.97984
## s(DREB)        1.760  2.230  1.539  0.56329
## s(REB)         1.000  1.000  0.158  0.69056
## s(AST)         1.000  1.000  8.586  0.00339 **
## s(STL)         1.000  1.000  0.152  0.69641
## s(BLK)         1.000  1.000  3.512  0.06092 .
## s(TOV)         1.000  1.000  5.537  0.01862 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.249   Deviance explained = 24.6%
## UBRE = 0.044542  Scale est. = 1         n = 784
```

Response: From the GAM model, we observe that significant predictors for the model are **GamesPlayed**, **AST**, and **TOV**. The edf (effective degrees of freedom) reported for 'AST' and 'TOV' is 1, which ascertains that these two variables have linear association with the response variable.

Much like the GLM model in (a), the significant variables are the same as mentioned above. Comparing the GAM model to the regularized logistic model in (b), we can infer that 'GamesPlayed' is a significant predictor variable. 'AST' and 'TOV' are significant variables with two out of the four models stated in part(b), both the models being with minimum value of lambda with class and AUC attributes.

```
pred_gam.model <- as.vector(predict(gam.model, newdata = test[,-c(1,21,22)],
type = "response"))

pr.gam.model <- prediction(pred_gam.model, test$y)
prf.gam.model <- performance(pr.gam.model, measure = "tpr", x.measure = "fpr"
)

par(mfrow=c(1,1))
plot(prf.gam.model,main = "gam")
```

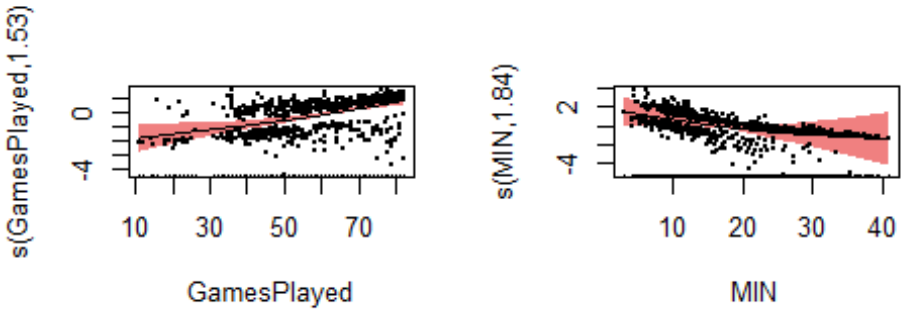



```
## Misclassification

(misclassification.rate.gam <- mean(round(pred_gam.model)!=test$y))

## [1] 0.2522255

## AUC

auc.gam.model <- performance(pr.gam.model, measure = "auc")
(auc.gam.model <- auc.gam.model@y.values[[1]])

## [1] 0.786687
```

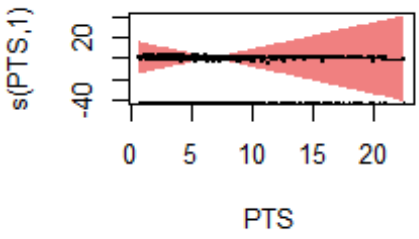
Response:

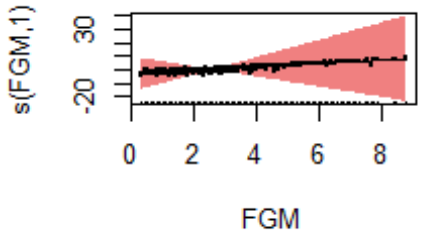
The performance of the GAM model is comparable to the performance of logistic and regularized logistic models developed in earlier parts as the AUC is around the same range for the models.

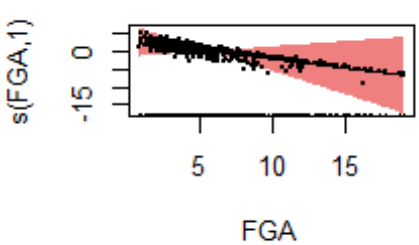
```
plot(gam.model, scale = 0, se = 2, shade = TRUE, resid = TRUE, pages = 5, pch
= 19, cex = 0.25, shade.col='lightcoral')
```

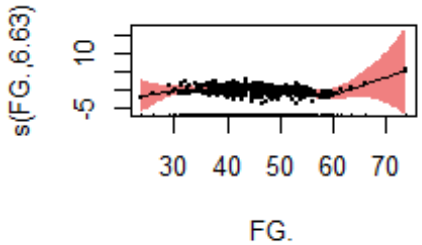

s(GamesPlayed,1.53)
-4
0
10
30
50
70
GamesPlayed

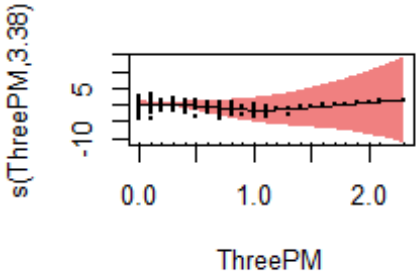
s(MIN,1.84)
-4
2
10
20
30
40
MIN

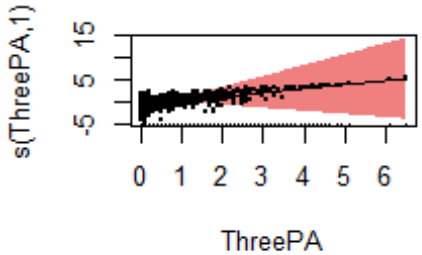

s(PTS,1)
-40
20
0
5
10
15
20
PTS

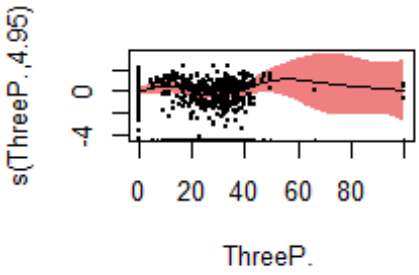

s(FGM,1)
-20
30
0
2
4
6
8
FGM

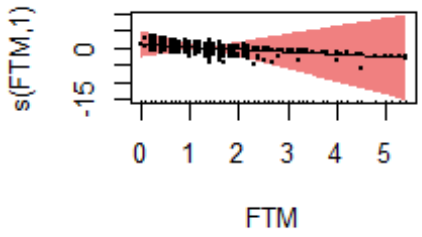

s(FGA,1)
-15
0
5
10
15
FGA

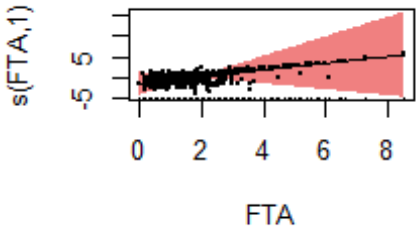

s(FG.,6.63)
-5
10
30
40
50
60
70
FG.

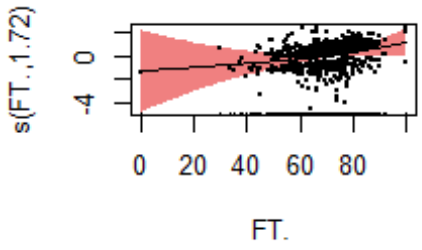

s(ThreePM,3.38)
-10
5
0.0
1.0
2.0
ThreePM

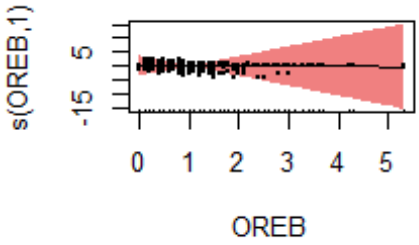

s(ThreePA,1)
-5
5
15
0
1
2
3
4
5
6
ThreePA

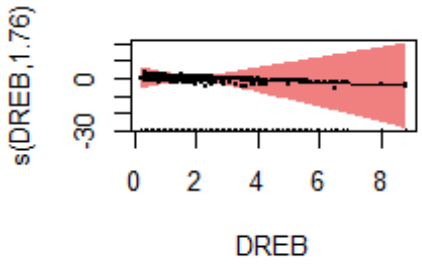

s(ThreeP.,4.95)
-4
0
0
20
40
60
80
ThreeP.

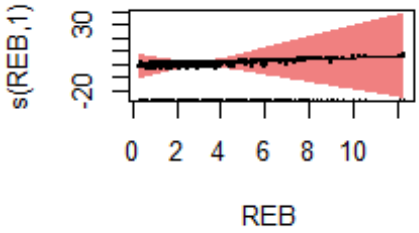

s(FTM,1)
-15
0
0
1
2
3
4
5
FTM

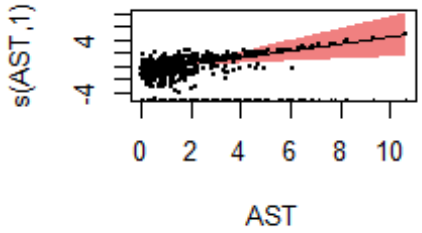

s(FTA,1)
-5
5
0
2
4
6
8
FTA

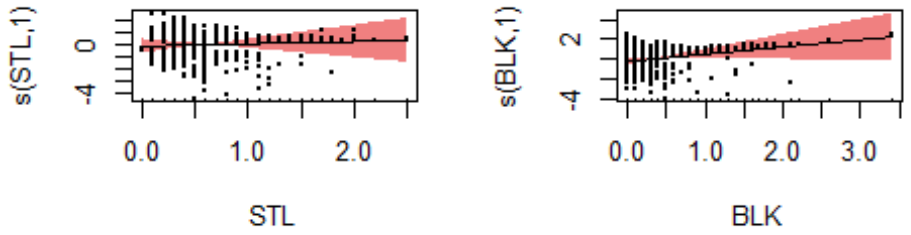

s(FT.,1.72)
-4
0
0
20
40
60
80
FT.

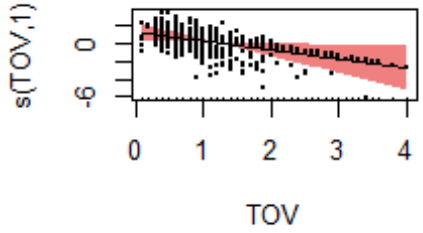

s(OREB,1)
-15
5
0
1
2
3
4
5
OREB


s(DREB,1.76)
-30
0
0
2
4
6
8
DREB


s(REB,1)
-20
30
0
2
4
6
8
10
REB


s(AST,1)
-4
4
0
2
4
6
8
10
AST





Response:

As found in the partial effect plot, our significant predictors are: **GamesPlayed**, **AST**, **TOV** The effect of 'GamesPlayed' on log odds of 5 or more years longetivity of a rookie is very slightly non-linear(confirmed with edf), and positive. The effect for 'AST' on the log odds of 5 or more years longetivity of a rookie is linear, which comprehends that increase is 'AST' is associated with a linear increase in log-odds of a five year longetivity of a rookie in the league. 'TOV' has a downward slope which is understandable, since good rookies with a five year longetivity are expected to have lower 'TOV' as they would not want to lose possession of the ball when in control. We can infer that there is a negative linear association with 'TOV' in this case.

Another interesting observation from the plots is the predictor variable 'FG.' .From the edf, we can observe that it is very non-linear(edf of 6.633) and has low p-value, but it is not present in 5% threshold. The reason behind low p-value for this predictor variable is because a few rookies have extreme value at both the ends, while majority of rookies have consistent values between 35 and 55. However, this predictor is insignificant in our model.

#Question 1d

Response:

To summarize some important observations from the models developed in this homework are:

- ‘GamesPlayed’ is a significant predictor of 5 or more years of longetivity in the league for all the models.
- The variables are highly correlated among themselves. For example, if a rookie has a large number of games played, it is obvious that it would be accompanied by high ‘MIN’ values, thus scoring higher points and demonstrating other better game statistics.
- The model is comparable across all the different models since the predictive performance of 5 year longetivity in the league is around the same range:

Missclassification rate is between: 0.24 and 0.28

AUC performance is between: 0.77 and 0.8

- The logistic lasso regression model performs the best with missclassification rate of 0.25 and AUC of 0.79.