NBA LASSO + RIDGE

```r
#Read in CSVs
nba <- read.csv("update142.csv")
nba$X <- NULL

#Remove Age, Tm, and Pos columns
mvp <- subset(nba, select = -c(Age, Tm, Pos, DPOY))
dpoy <- subset(nba, select = -c(Age, Tm, Pos, MVP))

roy <- read.csv("roydata142.csv")
roy$X <- NULL
roy <- subset(roy, select = -c(Age, Tm, Pos))

smoy <- read.csv("smoydata142.csv")
smoy$X <- NULL
smoy <- subset(smoy, select = -c(Age, Tm, Pos))

mip <- read.csv("mip_stats.csv")
mip[is.na(mip)] <- 0

#Create Copy for each award
mvpcopy <- mvp
dpoycopy <- dpoy
roycopy <- roy
mipcopy <- mip
mipcopy <- filter(mipcopy, Season > 2010)
smoycopy <- smoy

#Combine Player + Season into single column
mvpcopy$Player <- paste0(mvpcopy$Player,mvpcopy$Season)
mvpcopy <- mvpcopy[!duplicated(mvpcopy$Player), ]
rownames(mvpcopy) <- mvpcopy[,1]
mvpcopy <- mvpcopy[,-1]

dpoycopy$Player <- paste0(dpoycopy$Player,dpoycopy$Season)
dpoycopy <- dpoycopy[!duplicated(dpoycopy$Player), ]
rownames(dpoycopy) <- dpoycopy[,1]
dpoycopy <- dpoycopy[,-1]

roycopy$Player <- paste0(roycopy$Player,roycopy$Season)
roycopy <- roycopy[!duplicated(roycopy$Player), ]
rownames(roycopy) <- roycopy[,1]
roycopy <- roycopy[,-1]

mipcopy$Player <- paste0(mipcopy$Player,mipcopy$Season)
mipcopy <- mipcopy[!duplicated(mipcopy$Player), ]
rownames(mipcopy) <- mipcopy[,1]
```

```r
mipcopy <- mipcopy[,-1]

smoycopy$Player <- paste0(smoycopy$Player,smoycopy$Season)
smoycopy <- smoycopy[!duplicated(smoycopy$Player), ]
rownames(smoycopy) <- smoycopy[,1]
smoycopy <- smoycopy[,-1]
```

```r
#Split Data by Season, 2010-2017 is training, 2018-2020 is testing
trainmvp <- filter(mvpcopy, Season <= 2017)
testmvp <- filter(mvpcopy, Season > 2017)
trainmvp <- subset(trainmvp, select = -c(Season))
testmvp <- subset(testmvp, select = -c(Season))

traindpoy <- filter(dpoycopy, Season <= 2017)
testdpoy <- filter(dpoycopy, Season > 2017)
traindpoy <- subset(traindpoy, select = -c(Season))
testdpoy <- subset(testdpoy, select = -c(Season))

trainroy <- filter(roycopy, Season <= 2017)
testroy <- filter(roycopy, Season > 2017)
trainroy <- subset(trainroy, select = -c(Season))
testroy <- subset(testroy, select = -c(Season))

trainmip <- filter(mipcopy, Season <= 2017)
testmip <- filter(mipcopy, Season > 2017)
trainmip <- subset(trainmip, select = -c(Season))
testmip <- subset(testmip, select = -c(Season))

trainsmoy <- filter(smoycopy, Season <= 2017)
testsmoy <- filter(smoycopy, Season > 2017)
trainsmoy <- subset(trainsmoy, select = -c(Season))
testsmoy <- subset(testsmoy, select = -c(Season))

#I created an R squared function,
rsq <- function (true, predicted)
{
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
}
```

```r
#LASSO for MVP award
xtrainmvp <- model.matrix(MVP ~.^2  , trainmvp)[,-1]
xtestmvp <- model.matrix(MVP ~.^2 , testmvp)[,-1]
ytrainmvp <- trainmvp$MVP
ytestmvp <- testmvp$MVP

lambdas <- 10^seq(2, -3, by = -.1)

#Use cross validation to find best lambda
cvmvp <- cv.glmnet(x = xtrainmvp, y = ytrainmvp, alpha = 1, lambda = lambdas, nfolds = 5, family = "gaus
bestlammvp <- cvmvp$lambda.min
lassomvp <- glmnet(x = xtrainmvp, ytrainmvp, alpha = 1, lambda = bestlammvp)
```

```r
predmvptrain <- predict(lassomvp, s = bestlammvp, newx = xtrainmvp)
predmvptest <- predict(lassomvp, s = bestlammvp, newx = xtestmvp)

#Display the R2, RMSE, and MAE for both the training and testing sets
lassomvptrain <- data.frame(Model = "lassomvptrain", R2 = rsq(ytrainmvp, predmvptrain), RMSE = RMSE(ytra
lassomvptest <- data.frame(Model = "lassomvptest",R2 = rsq(ytestmvp, predmvptest), RMSE = RMSE(ytestmvp

#Create a new dataframe that combines the predicted values with the training/testing set
resultmvptrain <- cbind(trainmvp, "a" = round(predmvptrain,8))
resultmvptest <- cbind(testmvp, "a" = round(predmvptest,8))
resultmvptrain <- subset(resultmvptrain, select = c(MVP,47))
resultmvptest <- subset(resultmvptest, select = c(MVP,47))

names(resultmvptrain)[names(resultmvptrain) == '1'] <- "Predicted"
names(resultmvptest)[names(resultmvptest) == '1'] <- "Predicted"
orderedmvp <- rbind(resultmvptrain,resultmvptest)
orderedmvp <- orderedmvp[order(-orderedmvp$Predicted),]

#Create a variable that takes the absolute difference between actual and predicted vote shares
orderedmvp$diffmvp <- abs(orderedmvp$MVP - orderedmvp$Predicted)

#We now create copies of the results and original dataset to merge.
mvpcopy2 <- mvpcopy
mvpcopy2$Player <- rownames(mvpcopy2)
orderedmvp2 <- orderedmvp
orderedmvp2$Player <- rownames(orderedmvp2)
mergemvp <- merge(x = mvpcopy2, y = orderedmvp2[ , c("Player", "Predicted", "diffmvp")], by = "Player")
orderedtop <- mergemvp[order(-mergemvp$MVP),]
orderedtop <- subset(orderedtop, select = c(Player, Season, MVP, Predicted, diffmvp))

#Filter by season and take the absolute difference of each season
diff2010 <- head(filter(orderedtop, Season == 2010),10)
mean(diff2010$diffmvp)
```

```
## [1] 0.1405747
```

```r
diff2011 <- head(filter(orderedtop, Season == 2011),10)
mean(diff2011$diffmvp)
```

```
## [1] 0.1549307
```

```r
diff2012 <- head(filter(orderedtop, Season == 2012),10)
mean(diff2012$diffmvp)
```

```
## [1] 0.1626386
```

```r
diff2013 <- head(filter(orderedtop, Season == 2013),10)
mean(diff2013$diffmvp)
```

```
## [1] 0.1034745
```

```r
diff2014 <- head(filter(orderedtop, Season == 2014),10)
mean(diff2014$diffmvp)
```

```
## [1] 0.1608601
```

```r
diff2015 <- head(filter(orderedtop, Season == 2015),10)
mean(diff2015$diffmvp)
```

```
## [1] 0.1784783
```

```r
diff2016 <- head(filter(orderedtop, Season == 2016),10)
mean(diff2016$diffmvp)
```

```
## [1] 0.1751332
```

```r
diff2017 <- head(filter(orderedtop, Season == 2017),10)
mean(diff2017$diffmvp)
```

```
## [1] 0.1445609
```

```r
diff2018 <- head(filter(orderedtop, Season == 2018),10)
mean(diff2018$diffmvp)
```

```
## [1] 0.1411646
```

```r
diff2019 <- head(filter(orderedtop, Season == 2019),10)
mean(diff2019$diffmvp)
```

```
## [1] 0.1493529
```

```r
diff2020 <- head(filter(orderedtop, Season == 2020),10)
mean(diff2020$diffmvp)
```

```
## [1] 0.1621506
```

```r
#LASSO for DPOY
#The steps are identical to the previous

xtraindpoy <- model.matrix(DPOY ~.^2 , traindpoy)[,-1]
xtestdpoy <- model.matrix(DPOY ~.^2 , testdpoy)[,-1]
ytraindpoy <- traindpoy$DPOY
ytestdpoy <- testdpoy$DPOY

lambdas <- 10^seq(2, -3, by = -.1)

cvdpoy <- cv.glmnet(x = xtraindpoy, y = ytraindpoy, alpha = 1, lambda = lambdas, nfolds = 5, family = "g
bestlamdpoy <- cvdpoy$lambda.min
lassodpoy <- glmnet(x = xtraindpoy, ytraindpoy, alpha = 1, lambda = bestlamdpoy)
```

```
preddpoytrain <- predict(lassodpoy, s = bestlamdpoy, newx = xtraindpoy)
preddpoytest <- predict(lassodpoy, s = bestlamdpoy, newx = xtestdpoy)


lassodpoytrain <- data.frame(Model = "lassodpoytrain", R2 = rsq(ytraindpoy, preddpoytrain), RMSE = RMSE
lassodpoytest <- data.frame(Model = "lassodpoytest",R2 = rsq(ytestdpoy, preddpoytest), RMSE = RMSE(ytes

resultdpoytrain <- cbind(traindpoy, round(preddpoytrain,8))
resultdpoytest <- cbind(testdpoy, "a" = round(preddpoytest,8))
resultdpoytrain <- subset(resultdpoytrain, select = c(DPOY,47))
resultdpoytest <- subset(resultdpoytest, select = c(DPOY,47))

names(resultdpoytrain)[names(resultdpoytrain) == '1'] <- "Predicted"
names(resultdpoytest)[names(resultdpoytest) == '1'] <- "Predicted"
ordereddpoy <- rbind(resultdpoytrain,resultdpoytest)
ordereddpoy <- ordereddpoy[order(-ordereddpoy$Predicted),]

ordereddpoy$diffdpoy <- abs(ordereddpoy$DPOY - orderedmvp$Predicted)
dpoycopy2 <- dpoycopy
dpoycopy2$Player <- rownames(dpoycopy2)
ordereddpoy2 <- ordereddpoy
ordereddpoy2$Player <- rownames(ordereddpoy2)
mergedpoy <- merge(x = dpoycopy2, y = ordereddpoy2[ , c("Player", "Predicted", "diffdpoy")], by = "Play
orderedd <- mergedpoy[order(-mergedpoy$DPOY),]
orderedd <- subset(orderedd, select = c(Player, Season, DPOY, Predicted, diffdpoy))

diff2010 <- head(filter(orderedd, Season == 2010),10)
mean(diff2010$diffdpoy)
```

```
## [1] 0.1450771
```

```
diff2011 <- head(filter(orderedd, Season == 2011),10)
mean(diff2011$diffdpoy)
```

```
## [1] 0.09088664
```

```
diff2012 <- head(filter(orderedd, Season == 2012),10)
mean(diff2012$diffdpoy)
```

```
## [1] 0.1082697
```

```
diff2013 <- head(filter(orderedd, Season == 2013),10)
mean(diff2013$diffdpoy)
```

```
## [1] 0.1014607
```

```
diff2014 <- head(filter(orderedd, Season == 2014),10)
mean(diff2014$diffdpoy)
```

```
## [1] 0.1324279
```

```r
diff2015 <- head(filter(orderedd, Season == 2015),10)
mean(diff2015$diffdpoy)
```

```
## [1] 0.1454069
```

```r
diff2016 <- head(filter(orderedd, Season == 2016),10)
mean(diff2016$diffdpoy)
```

```
## [1] 0.2525272
```

```r
diff2017 <- head(filter(orderedd, Season == 2017),10)
mean(diff2017$diffdpoy)
```

```
## [1] 0.08748164
```

```r
diff2018 <- head(filter(orderedd, Season == 2018),10)
mean(diff2018$diffdpoy)
```

```
## [1] 0.1071578
```

```r
diff2019 <- head(filter(orderedd, Season == 2019),10)
mean(diff2019$diffdpoy)
```

```
## [1] 0.09919849
```

```r
diff2020 <- head(filter(orderedd, Season == 2020),10)
mean(diff2020$diffdpoy)
```

```
## [1] 0.09132216
```

```r
#LASSO for SMOY
#The steps are identical to the previous

xtrainsmoy <- model.matrix(SMOY ~.^2 , trainsmoy)[,-1]
xtestsmoy <- model.matrix(SMOY ~.^2 , testsmoy)[,-1]
ytrainsmoy <- trainsmoy$SMOY
ytestsmoy <- testsmoy$SMOY

cvsmoy <- cv.glmnet(x = xtrainsmoy, y = ytrainsmoy, alpha = 1, lambda = lambdas, nfolds = 5, family = "g
bestlamsmoy <- cvsmoy$lambda.min
lassosmoy <- glmnet(x = xtrainsmoy, ytrainsmoy, alpha = 1, lambda = bestlamsmoy)
predsmoytrain <- predict(lassosmoy, s = bestlamsmoy, newx = xtrainsmoy)
predsmoytest <- predict(lassosmoy, s = bestlamsmoy, newx = xtestsmoy)

lassosmoytrain <- data.frame(Model = "lassodpoytrain", R2 = rsq(ytrainsmoy, predsmoytrain), RMSE = RMSE
lassosmoytest <- data.frame(Model = "lassodpoytest",R2 = rsq(ytestsmoy, predsmoytest), RMSE = RMSE(ytes

resultsmoytrain <- cbind(trainsmoy, round(predsmoytrain,8))
resultsmoytest <- cbind(testsmoy, "a" = round(predsmoytest,8))
```

```r
resultsmoytrain <- subset(resultsmoytrain, select = c(SMOY,47))
resultsmoytest <- subset(resultsmoytest, select = c(SMOY,47))

names(resultsmoytrain)[names(resultsmoytrain) == '1'] <- "Predicted"
names(resultsmoytest)[names(resultsmoytest) == '1'] <- "Predicted"
orderedsmoy <- rbind(resultsmoytrain,resultsmoytest)
orderedsmoy <- orderedsmoy[order(-orderedsmoy$Predicted),]

orderedsmoy$diffsmoy <- abs(orderedsmoy$SMOY - orderedsmoy$Predicted)
smoycopy2 <- smoycopy
smoycopy2$Player <- rownames(smoycopy2)
orderedsmoy2 <- orderedsmoy
orderedsmoy2$Player <- rownames(orderedsmoy2)
mergesmoy <- merge(x = smoycopy2, y = orderedsmoy2[ , c("Player", "Predicted", "diffsmoy")], by = "Play
ordereds <- mergesmoy[order(-mergesmoy$SMOY),]
ordereds <- subset(ordereds, select = c(Player, Season, SMOY, Predicted, diffsmoy))

diff2010 <- head(filter(ordereds, Season == 2010),10)
mean(diff2010$diffsmoy)
```

```
## [1] 0.1429262
```

```r
diff2011 <- head(filter(ordereds, Season == 2011),10)
mean(diff2011$diffsmoy)
```

```
## [1] 0.1454429
```

```r
diff2012 <- head(filter(ordereds, Season == 2012),10)
mean(diff2012$diffsmoy)
```

```
## [1] 0.1000815
```

```r
diff2013 <- head(filter(ordereds, Season == 2013),10)
mean(diff2013$diffsmoy)
```

```
## [1] 0.1474536
```

```r
diff2014 <- head(filter(ordereds, Season == 2014),10)
mean(diff2014$diffsmoy)
```

```
## [1] 0.1469919
```

```r
diff2015 <- head(filter(ordereds, Season == 2015),10)
mean(diff2015$diffsmoy)
```

```
## [1] 0.1169367
```

```r
diff2016 <- head(filter(ordereds, Season == 2016),10)
mean(diff2016$diffsmoy)
```

```
## [1] 0.1149455
```

```r
diff2017 <- head(filter(ordereds, Season == 2017),10)
mean(diff2017$diffsmoy)
```

```
## [1] 0.1649018
```

```r
diff2018 <- head(filter(ordereds, Season == 2018),10)
mean(diff2018$diffsmoy)
```

```
## [1] 0.1189855
```

```r
diff2019 <- head(filter(ordereds, Season == 2019),10)
mean(diff2019$diffsmoy)
```

```
## [1] 0.1518067
```

```r
diff2020 <- head(filter(ordereds, Season == 2020),10)
mean(diff2020$diffsmoy)
```

```
## [1] 0.1492002
```

```r
#LASSO for ROY
#The steps are the same as the previous
xtrainroy <- model.matrix(ROY ~.^2 , trainroy)[,-1]
xtestroy <- model.matrix(ROY ~.^2 , testroy)[,-1]
ytrainroy <- trainroy$ROY
ytestroy <- testroy$ROY

cvroy <- cv.glmnet(x = xtrainroy, y = ytrainroy, alpha = 1, lambda = lambdas, nfolds = 5, family = "gaus
bestlamroy <- cvroy$lambda.min
lassoroy <- glmnet(x = xtrainroy, ytrainroy, alpha = 1, lambda = bestlamroy)
predroytrain <- predict(lassoroy, s = bestlamroy, newx = xtrainroy)
predroytest <- predict(lassoroy, s = bestlamroy, newx = xtestroy)


lassoroytrain <- data.frame(Model = "lassoroytrain", R2 = rsq(ytrainroy, predroytrain), RMSE = RMSE(ytra
lassoroytest <- data.frame(Model = "lassoroytest",R2 = rsq(ytestroy, predroytest), RMSE = RMSE(ytestroy

resultroytrain <- cbind(trainroy, round(predroytrain,8))
resultroytest <- cbind(testroy, "a" = round(predroytest,8))
resultroytrain <- subset(resultroytrain, select = c(ROY,47))
resultroytest <- subset(resultroytest, select = c(ROY,47))

names(resultroytrain)[names(resultroytrain) == '1'] <- "Predicted"
names(resultroytest)[names(resultroytest) == '1'] <- "Predicted"
```

```r
orderedroy <- rbind(resultroytrain,resultroytest)
orderedroy <- orderedroy[order(-orderedroy$Predicted),]


orderedroy$diffroy <- abs(orderedroy$ROY - orderedroy$Predicted)
roycopy2 <- roycopy
roycopy2$Player <- rownames(roycopy2)
orderedroy2 <- orderedroy
orderedroy2$Player <- rownames(orderedroy2)
mergeroy <- merge(x = roycopy2, y = orderedroy2[ , c("Player", "Predicted", "diffroy")], by = "Player")
orderedr <- mergeroy[order(-mergeroy$ROY),]
orderedr <- subset(orderedr, select = c(Player, Season, ROY, Predicted, diffroy))

diff2010 <- head(filter(orderedr, Season == 2010),10)
mean(diff2010$diffroy)
```

```
## [1] 0.09265858
```

```r
diff2011 <- head(filter(orderedr, Season == 2011),10)
mean(diff2011$diffroy)
```

```
## [1] 0.03961652
```

```r
diff2012 <- head(filter(orderedr, Season == 2012),10)
mean(diff2012$diffroy)
```

```
## [1] 0.1064845
```

```r
diff2013 <- head(filter(orderedr, Season == 2013),10)
mean(diff2013$diffroy)
```

```
## [1] 0.1242702
```

```r
diff2014 <- head(filter(orderedr, Season == 2014),10)
mean(diff2014$diffroy)
```

```
## [1] 0.09498185
```

```r
diff2015 <- head(filter(orderedr, Season == 2015),10)
mean(diff2015$diffroy)
```

```
## [1] 0.1503092
```

```r
diff2016 <- head(filter(orderedr, Season == 2016),10)
mean(diff2016$diffroy)
```

```
## [1] 0.1428654
```

```r
diff2017 <- head(filter(orderedr, Season == 2017),10)
mean(diff2017$diffroy)
```

```
## [1] 0.1528465
```

```r
diff2018 <- head(filter(orderedr, Season == 2018),10)
mean(diff2018$diffroy)
```

```
## [1] 0.08977174
```

```r
diff2019 <- head(filter(orderedr, Season == 2019),10)
mean(diff2019$diffroy)
```

```
## [1] 0.07485164
```

```r
diff2020 <- head(filter(orderedr, Season == 2020),10)
mean(diff2020$diffroy)
```

```
## [1] 0.1224762
```

```r
#LASSO for MIP
#The steps are the same as the previous
xtrainmip <- model.matrix(MIP ~.^2 , trainmip)[,-1]
xtestmip <- model.matrix(MIP ~.^2 , testmip)[,-1]
ytrainmip <- trainmip$MIP
ytestmip <- testmip$MIP

cvmip <- cv.glmnet(x = xtestmip, y = ytestmip, alpha = 1, lambda = lambdas, nfolds = 5, family = "gauss
bestlammip <- cvmip$lambda.min
lassomip <- glmnet(x = xtrainmip, ytrainmip, alpha = 1, lambda = bestlammip)
predmiptrain <- predict(lassomip, s = bestlammip, newx = xtrainmip)
predmiptest <- predict(lassomip, s = bestlammip, newx = xtestmip)


lassomiptrain <- data.frame(Model = "lassomiptrain", R2 = rsq(ytrainmip, predmiptrain), RMSE = RMSE(ytra
lassomiptest <- data.frame(Model = "lassomiptest",R2 = rsq(ytestmip, predmiptest), RMSE = RMSE(ytestmip

resultmiptrain <- cbind(trainmip, round(predmiptrain,8))
resultmiptest <- cbind(testmip, "a" = round(predmiptest,8))
resultmiptrain <- subset(resultmiptrain, select = c(MIP,47))
resultmiptest <- subset(resultmiptest, select = c(MIP,47))

names(resultmiptrain)[names(resultmiptrain) == '1'] <- "Predicted"
names(resultmiptest)[names(resultmiptest) == '1'] <- "Predicted"
orderedmip <- rbind(resultmiptrain,resultmiptest)
orderedmip <- orderedmip[order(-orderedmip$Predicted),]

orderedmip$diffmip <- abs(orderedmip$MIP - orderedmip$Predicted)
mipcopy2 <- mipcopy
mipcopy2$Player <- rownames(mipcopy2)
```

```r
orderedmip2 <- orderedmip
orderedmip2$Player <- rownames(orderedmip2)
mergemip <- merge(x = mipcopy2, y = orderedmip2[ , c("Player", "Predicted", "diffmip")], by = "Player")
orderedm <- mergemip[order(-mergemip$MIP),]
orderedm <- subset(orderedm, select = c(Player, Season, MIP, Predicted, diffmip))

diff2011 <- head(filter(orderedm, Season == 2011),10)
mean(diff2011$diffmip)
```

```
## [1] 0.1471149
```

```r
diff2012 <- head(filter(orderedm, Season == 2012),10)
mean(diff2012$diffmip)
```

```
## [1] 0.1279832
```

```r
diff2013 <- head(filter(orderedm, Season == 2013),10)
mean(diff2013$diffmip)
```

```
## [1] 0.139988
```

```r
diff2014 <- head(filter(orderedm, Season == 2014),10)
mean(diff2014$diffmip)
```

```
## [1] 0.1433748
```

```r
diff2015 <- head(filter(orderedm, Season == 2015),10)
mean(diff2015$diffmip)
```

```
## [1] 0.1507663
```

```r
diff2016 <- head(filter(orderedm, Season == 2016),10)
mean(diff2016$diffmip)
```

```
## [1] 0.1448644
```

```r
diff2017 <- head(filter(orderedm, Season == 2017),10)
mean(diff2017$diffmip)
```

```
## [1] 0.153043
```

```r
diff2018 <- head(filter(orderedm, Season == 2018),10)
mean(diff2018$diffmip)
```

```
## [1] 0.1493857
```

```r
diff2019 <- head(filter(orderedm, Season == 2019),10)
mean(diff2019$diffmip)
```

```
## [1] 0.1636403
```

```r
diff2020 <- head(filter(orderedm, Season == 2020),10)
mean(diff2020$diffmip)
```

```
## [1] 0.1508278
```

```r
#Ridge for MVP
#Identical to Lasso except we use alpha = 0 instead of alpha = 1 in glmnet to indicate ridge
xtrainmvp3 <- model.matrix(MVP ~.^2  , trainmvp)[,-1]
xtestmvp3 <- model.matrix(MVP ~.^2 , testmvp)[,-1]
ytrainmvp3 <- trainmvp$MVP
ytestmvp3 <- testmvp$MVP

cv_ridgemvp <- cv.glmnet(xtrainmvp3, ytrainmvp3, alpha = 0, lambda = lambdas)
optimal_lambdamvp <- cv_ridgemvp$lambda.min
ridge_regmvp = glmnet(xtrainmvp3, ytrainmvp3, alpha = 0, family = 'gaussian', lambda = optimal_lambdamvp
predmvptrain3 <- predict(ridge_regmvp, s = optimal_lambdamvp, newx = xtrainmvp3)
predmvptest3 <- predict(ridge_regmvp, s = optimal_lambdamvp, newx = xtestmvp3)

ridgemvptrain <- data.frame(Model = "ridgemvptrain", R2 = rsq(ytrainmvp3, predmvptrain3), RMSE = RMSE(yt
ridgemvptest <- data.frame(Model = "ridgemvptest", R2 = rsq(ytestmvp3, predmvptest3), RMSE = RMSE(ytestm

resultmvptrain3 <- cbind(trainmvp, "a" = round(predmvptrain3,8))
resultmvptest3 <- cbind(testmvp, "a" = round(predmvptest3,8))
resultmvptrain3 <- subset(resultmvptrain3, select = c(MVP,47))
resultmvptest3 <- subset(resultmvptest3, select = c(MVP,47))

names(resultmvptrain3)[names(resultmvptrain3) == '1'] <- "Predicted"
names(resultmvptest3)[names(resultmvptest3) == '1'] <- "Predicted"
orderedmvp3 <- rbind(resultmvptrain3,resultmvptest3)
orderedmvp3 <- orderedmvp3[order(-orderedmvp3$Predicted),]

orderedmvp3$diffmvp <- abs(orderedmvp3$MVP - orderedmvp3$Predicted)

mvpcopy2 <- mvpcopy
mvpcopy2$Player <- rownames(mvpcopy2)
orderedmvp2 <- orderedmvp
orderedmvp3$Player <- rownames(orderedmvp3)
mergemvp <- merge(x = mvpcopy2, y = orderedmvp3[ , c("Player", "Predicted", "diffmvp")], by = "Player")
orderedtop <- mergemvp[order(-mergemvp$MVP),]
orderedtop <- subset(orderedtop, select = c(Player, Season, MVP, Predicted, diffmvp))

diff2010 <- head(filter(orderedtop, Season == 2010),10)
mean(diff2010$diffmvp)
```

```
## [1] 0.1102125
```

```r
diff2011 <- head(filter(orderedtop, Season == 2011),10)
mean(diff2011$diffmvp)
```

```
## [1] 0.1433106
```

```r
diff2012 <- head(filter(orderedtop, Season == 2012),10)
mean(diff2012$diffmvp)
```

```
## [1] 0.1459586
```

```r
diff2013 <- head(filter(orderedtop, Season == 2013),10)
mean(diff2013$diffmvp)
```

```
## [1] 0.1024702
```

```r
diff2014 <- head(filter(orderedtop, Season == 2014),10)
mean(diff2014$diffmvp)
```

```
## [1] 0.138303
```

```r
diff2015 <- head(filter(orderedtop, Season == 2015),10)
mean(diff2015$diffmvp)
```

```
## [1] 0.144341
```

```r
diff2016 <- head(filter(orderedtop, Season == 2016),10)
mean(diff2016$diffmvp)
```

```
## [1] 0.1390649
```

```r
diff2017 <- head(filter(orderedtop, Season == 2017),10)
mean(diff2017$diffmvp)
```

```
## [1] 0.1301611
```

```r
diff2018 <- head(filter(orderedtop, Season == 2018),10)
mean(diff2018$diffmvp)
```

```
## [1] 0.145024
```

```r
diff2019 <- head(filter(orderedtop, Season == 2019),10)
mean(diff2019$diffmvp)
```

```
## [1] 0.1338642
```

```r
diff2020 <- head(filter(orderedtop, Season == 2020),10)
mean(diff2020$diffmvp)
```

```
## [1] 0.1383936
```

```r
#Ridge for DPOY
#Same as previous
xtraindpoy3 <- model.matrix(DPOY ~.^2  , traindpoy)[,-1]
xtestdpoy3 <- model.matrix(DPOY ~.^2 , testdpoy)[,-1]
ytraindpoy3 <- traindpoy$DPOY
ytestdpoy3 <- testdpoy$DPOY

cv_ridgedpoy <- cv.glmnet(xtraindpoy3, ytraindpoy3, alpha = 0, lambda = lambdas)
optimal_lambdadpoy <- cv_ridgedpoy$lambda.min
ridge_regdpoy = glmnet(xtraindpoy3, ytraindpoy3, alpha = 0, family = 'gaussian', lambda = optimal_lambda
preddpoytrain3 <- predict(ridge_regdpoy, s = optimal_lambdadpoy, newx = xtraindpoy3)
preddpoytest3 <- predict(ridge_regdpoy, s = optimal_lambdadpoy, newx = xtestdpoy3)

ridgedpoytrain <- data.frame(Model = "ridgedpoytrain", R2 = rsq(ytraindpoy3, preddpoytrain3), RMSE = RM
ridgedpoytest <- data.frame(Model = "ridgedpoytest", R2 = rsq(ytestdpoy3, preddpoytest3), RMSE = RMSE(y

resultdpoytrain3 <- cbind(traindpoy, "a" = round(preddpoytrain3,8))
resultdpoytest3 <- cbind(testdpoy, "a" = round(preddpoytest3,8))
resultdpoytrain3 <- subset(resultdpoytrain3, select = c(DPOY,47))
resultdpoytest3 <- subset(resultdpoytest3, select = c(DPOY,47))

names(resultdpoytrain3)[names(resultdpoytrain3) == '1'] <- "Predicted"
names(resultdpoytest3)[names(resultdpoytest3) == '1'] <- "Predicted"
ordereddpoy3 <- rbind(resultdpoytrain3,resultdpoytest3)
ordereddpoy3 <- ordereddpoy3[order(-ordereddpoy3$Predicted),]

ordereddpoy3$diffdpoy <- abs(ordereddpoy3$DPOY - ordereddpoy3$Predicted)
dpoycopy2 <- dpoycopy
dpoycopy2$Player <- rownames(dpoycopy2)
ordereddpoy2 <- ordereddpoy3
ordereddpoy2$Player <- rownames(ordereddpoy3)
mergedpoy <- merge(x = dpoycopy2, y = ordereddpoy2[ , c("Player", "Predicted", "diffdpoy")], by = "Play
orderedd <- mergedpoy[order(-mergedpoy$DPOY),]
orderedd <- subset(orderedd, select = c(Player, Season, DPOY, Predicted, diffdpoy))

diff2010 <- head(filter(orderedd, Season == 2010),10)
mean(diff2010$diffdpoy)
```

```
## [1] 0.09260651
```

```r
diff2011 <- head(filter(orderedd, Season == 2011),10)
mean(diff2011$diffdpoy)
```

```
## [1] 0.09092973
```

```r
diff2012 <- head(filter(orderedd, Season == 2012),10)
mean(diff2012$diffdpoy)
```

```
## [1] 0.1122588
```

```r
diff2013 <- head(filter(orderedd, Season == 2013),10)
mean(diff2013$diffdpoy)
```

```
## [1] 0.0745736
```

```r
diff2014 <- head(filter(orderedd, Season == 2014),10)
mean(diff2014$diffdpoy)
```

```
## [1] 0.1228502
```

```r
diff2015 <- head(filter(orderedd, Season == 2015),10)
mean(diff2015$diffdpoy)
```

```
## [1] 0.1413407
```

```r
diff2016 <- head(filter(orderedd, Season == 2016),10)
mean(diff2016$diffdpoy)
```

```
## [1] 0.178187
```

```r
diff2017 <- head(filter(orderedd, Season == 2017),10)
mean(diff2017$diffdpoy)
```

```
## [1] 0.1419672
```

```r
diff2018 <- head(filter(orderedd, Season == 2018),10)
mean(diff2018$diffdpoy)
```

```
## [1] 0.132617
```

```r
diff2019 <- head(filter(orderedd, Season == 2019),10)
mean(diff2019$diffdpoy)
```

```
## [1] 0.1406266
```

```r
diff2020 <- head(filter(orderedd, Season == 2020),10)
mean(diff2020$diffdpoy)
```

```
## [1] 0.1414978
```

```r
#Ridge for ROY
#Same as previous
xtrainroy3 <- model.matrix(ROY ~.^2 , trainroy)[,-1]
xtestroy3 <- model.matrix(ROY ~.^2 , testroy)[,-1]
ytrainroy3 <- trainroy$ROY
ytestroy3 <- testroy$ROY

cv_ridgeroy <- cv.glmnet(xtrainroy3, ytrainroy3, alpha = 0, lambda = lambdas)
optimal_lambdaroy <- cv_ridgeroy$lambda.min
ridge_regdroy = glmnet(xtrainroy3, ytrainroy3, alpha = 0, family = 'gaussian', lambda = optimal_lambdaro
predroytrain3 <- predict(ridge_regdroy, s = optimal_lambdaroy, newx = xtrainroy3)
predroytest3 <- predict(ridge_regdroy, s = optimal_lambdaroy, newx = xtestroy3)

ridgeroytrain <- data.frame(Model = "ridgeroytrain", R2 = rsq(ytrainroy3, predroytrain3), RMSE = RMSE(y
ridgeroytest <- data.frame(Model = "ridgeroytest",R2 = rsq(ytestroy3, predroytest3), RMSE = RMSE(ytestr

resultroytrain3 <- cbind(trainroy, round(predroytrain3,8))
resultroytest3 <- cbind(testroy, "a" = round(predroytest3,8))
resultroytrain3 <- subset(resultroytrain3, select = c(ROY,47))
resultroytest3 <- subset(resultroytest3, select = c(ROY,47))

names(resultroytrain3)[names(resultroytrain3) == '1'] <- "Predicted"
names(resultroytest3)[names(resultroytest3) == '1'] <- "Predicted"
orderedroy3 <- rbind(resultroytrain3,resultroytest3)
orderedroy3 <- orderedroy3[order(-orderedroy3$Predicted),]

orderedroy3$diffroy <- abs(orderedroy3$ROY - orderedroy3$Predicted)
roycopy2 <- roycopy
roycopy2$Player <- rownames(roycopy2)
orderedroy2 <- orderedroy3
orderedroy2$Player <- rownames(orderedroy2)
mergeroy <- merge(x = roycopy2, y = orderedroy2[ , c("Player", "Predicted", "diffroy")], by = "Player")
orderedr <- mergeroy[order(-mergeroy$ROY),]
orderedr <- subset(orderedr, select = c(Player, Season, ROY, Predicted, diffroy))

diff2010 <- head(filter(orderedr, Season == 2010),10)
mean(diff2010$diffroy)
```

```
## [1] 0.09389829
```

```r
diff2011 <- head(filter(orderedr, Season == 2011),10)
mean(diff2011$diffroy)
```

```
## [1] 0.04224485
```

```r
diff2012 <- head(filter(orderedr, Season == 2012),10)
mean(diff2012$diffroy)
```

```
## [1] 0.1198806
```

```r
diff2013 <- head(filter(orderedr, Season == 2013),10)
mean(diff2013$diffroy)
```

```
## [1] 0.1111614
```

```r
diff2014 <- head(filter(orderedr, Season == 2014),10)
mean(diff2014$diffroy)
```

```
## [1] 0.09913177
```

```r
diff2015 <- head(filter(orderedr, Season == 2015),10)
mean(diff2015$diffroy)
```

```
## [1] 0.1344766
```

```r
diff2016 <- head(filter(orderedr, Season == 2016),10)
mean(diff2016$diffroy)
```

```
## [1] 0.1417649
```

```r
diff2017 <- head(filter(orderedr, Season == 2017),10)
mean(diff2017$diffroy)
```

```
## [1] 0.1532215
```

```r
diff2018 <- head(filter(orderedr, Season == 2018),10)
mean(diff2018$diffroy)
```

```
## [1] 0.09546244
```

```r
diff2019 <- head(filter(orderedr, Season == 2019),10)
mean(diff2019$diffroy)
```

```
## [1] 0.08595051
```

```r
diff2020 <- head(filter(orderedr, Season == 2020),10)
mean(diff2020$diffroy)
```

```
## [1] 0.123573
```

```r
#Ridge for SMOY
#Same as previous
xtrainsmoy3 <- model.matrix(SMOY ~.^2 , trainsmoy)[,-1]
xtestsmoy3 <- model.matrix(SMOY ~.^2 , testsmoy)[,-1]
ytrainsmoy3 <- trainsmoy$SMOY
ytestsmoy3 <- testsmoy$SMOY
```

```
cv_ridgesmoy <- cv.glmnet(xtrainsmoy3, ytrainsmoy3, alpha = 0, lambda = lambdas)
optimal_lambdasmoy <- cv_ridgesmoy$lambda.min
ridge_regsmoy = glmnet(xtrainsmoy3, ytrainsmoy3, alpha = 0, family = 'gaussian', lambda = optimal_lambda
predsmooytrain3 <- predict(ridge_regsmoy, s = optimal_lambdasmoy, newx = xtrainsmoy3)
predsmooytest3 <- predict(ridge_regsmoy, s = optimal_lambdasmoy, newx = xtestsmoy3)


lassosmoytrain3 <- data.frame(Model = "ridgesmoytrain", R2 = rsq(ytrainsmoy3, predsmooytrain3), RMSE = R
lassosmoytest3 <- data.frame(Model = "ridgesmoytest",R2 = rsq(ytestsmoy3, predsmooytest3), RMSE = RMSE(y

resultsmoytrain3 <- cbind(trainsmoy, round(predsmooytrain3,8))
resultsmoytest3 <- cbind(testsmoy, "a" = round(predsmooytest3,8))
resultsmoytrain3 <- subset(resultsmoytrain3, select = c(SMOY,47))
resultsmoytest3 <- subset(resultsmoytest3, select = c(SMOY,47))

names(resultsmoytrain3)[names(resultsmoytrain3) == '1'] <- "Predicted"
names(resultsmoytest3)[names(resultsmoytest3) == '1'] <- "Predicted"
orderedsmoy3 <- rbind(resultsmoytrain3,resultsmoytest3)
orderedsmoy3 <- orderedsmoy3[order(-orderedsmoy3$Predicted),]

orderedsmoy3$diffsmoy <- abs(orderedsmoy3$SMOY - orderedsmoy3$Predicted)
smoycopy2 <- smoycopy
smoycopy2$Player <- rownames(smoycopy2)
orderedsmoy2 <- orderedsmoy3
orderedsmoy2$Player <- rownames(orderedsmoy2)
mergesmoy <- merge(x = smoycopy2, y = orderedsmoy2[ , c("Player", "Predicted", "diffsmoy")], by = "Playe
ordereds <- mergesmoy[order(-mergesmoy$SMOY),]
ordereds <- subset(ordereds, select = c(Player, Season, SMOY, Predicted, diffsmoy))

diff2010 <- head(filter(ordereds, Season == 2010),10)
mean(diff2010$diffsmoy)
```

```
## [1] 0.1501435
```

```
diff2011 <- head(filter(ordereds, Season == 2011),10)
mean(diff2011$diffsmoy)
```

```
## [1] 0.1460117
```

```
diff2012 <- head(filter(ordereds, Season == 2012),10)
mean(diff2012$diffsmoy)
```

```
## [1] 0.1056165
```

```
diff2013 <- head(filter(ordereds, Season == 2013),10)
mean(diff2013$diffsmoy)
```

```
## [1] 0.1455987
```

```r
diff2014 <- head(filter(ordereds, Season == 2014),10)
mean(diff2014$diffsmoy)
```

```
## [1] 0.1449515
```

```r
diff2015 <- head(filter(ordereds, Season == 2015),10)
mean(diff2015$diffsmoy)
```

```
## [1] 0.1139584
```

```r
diff2016 <- head(filter(ordereds, Season == 2016),10)
mean(diff2016$diffsmoy)
```

```
## [1] 0.1151786
```

```r
diff2017 <- head(filter(ordereds, Season == 2017),10)
mean(diff2017$diffsmoy)
```

```
## [1] 0.1565389
```

```r
diff2018 <- head(filter(ordereds, Season == 2018),10)
mean(diff2018$diffsmoy)
```

```
## [1] 0.119902
```

```r
diff2019 <- head(filter(ordereds, Season == 2019),10)
mean(diff2019$diffsmoy)
```

```
## [1] 0.1614581
```

```r
diff2020 <- head(filter(ordereds, Season == 2020),10)
mean(diff2020$diffsmoy)
```

```
## [1] 0.1474114
```

```r
#Ridge for MIP
#Same as previous
xtrainmip3 <- model.matrix(MIP ~.^2 , trainmip)[,-1]
xtestmip3 <- model.matrix(MIP ~.^2 , testmip)[,-1]
ytrainmip3 <- trainmip$MIP
ytestmip3 <- testmip$MIP

cv_ridgesmip <- cv.glmnet(xtrainmip3, ytrainmip3, alpha = 0, lambda = lambdas)
optimal_lambdasmip <- cv_ridgesmip$lambda.min
ridge_regsmip = glmnet(xtrainsmoy3, ytrainsmoy3, alpha = 0, family = 'gaussian', lambda = optimal_lambda
predmiptrain3 <- predict(ridge_regsmip, s = optimal_lambdasmip, newx = xtrainmip3)
predmiptest3 <- predict(ridge_regsmip, s = optimal_lambdasmip, newx = xtestmip3)
```

```
ridgemiptrain <- data.frame(Model = "ridgemiptrain", R2 = rsq(ytrainmip3, predmiptrain3), RMSE = RMSE(y
ridgemiptest <- data.frame(Model = "ridgemiptest",R2 = rsq(ytestmip3, predmiptest3), RMSE = RMSE(ytestm

resultmiptrain3 <- cbind(trainmip, round(predmiptrain3,8))
resultmiptest3 <- cbind(testmip, "a" = round(predmiptest3,8))
resultmiptrain3 <- subset(resultmiptrain3, select = c(MIP,47))
resultmiptest3 <- subset(resultmiptest3, select = c(MIP,47))

names(resultmiptrain3)[names(resultmiptrain3) == '1'] <- "Predicted"
names(resultmiptest3)[names(resultmiptest3) == '1'] <- "Predicted"
orderedmip3 <- rbind(resultmiptrain3,resultmiptest3)
orderedmip3 <- orderedmip3[order(-orderedmip3$Predicted),]

orderedmip3$diffmip <- abs(orderedmip3$MIP - orderedmip3$Predicted)
mipcopy2 <- mipcopy
mipcopy2$Player <- rownames(mipcopy2)
orderedmip2 <- orderedmip3
orderedmip2$Player <- rownames(orderedmip2)
mergemip <- merge(x = mipcopy2, y = orderedmip2[ , c("Player", "Predicted", "diffmip")], by = "Player")
orderedm <- mergemip[order(-mergemip$MIP),]
orderedm <- subset(orderedm, select = c(Player, Season, MIP, Predicted, diffmip))

diff2011 <- head(filter(orderedm, Season == 2011),10)
mean(diff2011$diffmip)
```

```
## [1] 0.1377614
```

```
diff2012 <- head(filter(orderedm, Season == 2012),10)
mean(diff2012$diffmip)
```

```
## [1] 0.1226397
```

```
diff2013 <- head(filter(orderedm, Season == 2013),10)
mean(diff2013$diffmip)
```

```
## [1] 0.1553169
```

```
diff2014 <- head(filter(orderedm, Season == 2014),10)
mean(diff2014$diffmip)
```

```
## [1] 0.1312629
```

```
diff2015 <- head(filter(orderedm, Season == 2015),10)
mean(diff2015$diffmip)
```

```
## [1] 0.1674176
```

```r
diff2016 <- head(filter(orderedm, Season == 2016),10)
mean(diff2016$diffmip)
```

```
## [1] 0.1365493
```

```r
diff2017 <- head(filter(orderedm, Season == 2017),10)
mean(diff2017$diffmip)
```

```
## [1] 0.1401368
```

```r
diff2018 <- head(filter(orderedm, Season == 2018),10)
mean(diff2018$diffmip)
```

```
## [1] 0.1441136
```

```r
diff2019 <- head(filter(orderedm, Season == 2019),10)
mean(diff2019$diffmip)
```

```
## [1] 0.162357
```

```r
diff2020 <- head(filter(orderedm, Season == 2020),10)
mean(diff2020$diffmip)
```

```
## [1] 0.155756
```

```r
#Read in csv of 2021 nba stats (Beginning of the season - May 11, 2021)
nba2021 <- read.csv("2021data.csv")

Player <-  stri_trans_general(str = nba2021$Player, id = "Latin-ASCII")
Player <- gsub("\\\\.*", "", Player)
Player <- gsub("\\*.*", "", Player)
nba2021[["Player"]] <- Player

nba2021[is.na(nba2021)] <- 0
nba2021$MVP <- 0
nba2021$DPOY <- 0

#Create a subset for SMOY
smoy2021 <- nba2021 %>% filter((G/2) > GS)
smoy2021$SMOY <- 0
smoy2021$MVP <- NULL
smoy2021$DPOY <- NULL

#Create a subset for ROY
roylist <- read.csv("roylist2021.csv")
Player <-  stri_trans_general(str = roylist$Player, id = "Latin-ASCII")
Player <- gsub("\\\\.*", "", Player)
Player <- gsub("\\*.*", "", Player)
roylist[["Player"]] <- Player
```

```
roy2021 <- merge(nba2021, roylist, by = "Player")
roy2021$MVP <- NULL
roy2021$DPOY <- NULL
roy2021$ROY <- 0
write.csv(roy2021, "roy2021.csv")
write.csv(smoy2021, "smoy2021.csv")
write.csv(nba2021, "nba2021.csv")

mip21 <- read.csv("mip2021.csv")
mip21 <- subset(mip21, select = -c(Season))
```

LASSO Predictions for 2021

```
#Predicted 2021 MVP
mvpcopy2021 <- nba2021
rownames(mvpcopy2021) <- mvpcopy2021[,1]
mvpcopy2021 <- mvpcopy2021[,-1]
mvpcopy2021 <- filter(mvpcopy2021, G > 20)
mvpcopy2021 <- subset(mvpcopy2021, select = -c(Season, Pos, Age, Tm, DPOY))

xmvp2021 <- model.matrix(MVP ~.^2 , mvpcopy2021)[,-1]
ymvp2021 <- mvpcopy2021$MVP
predmvp2021 <- predict(lassomvp, newx = xmvp2021)
mvpresult2021 <- cbind(mvpcopy2021, "s0" = round(predmvp2021,8))
mvpresult2021 <- mvpresult2021[order(-mvpresult2021$s0),]
mvpresult2021 <- subset(mvpresult2021, select = c(s0))
names(mvpresult2021)[names(mvpresult2021) == 's0'] <- "Predicted"

#Predicted 2021 DPOY
dpoycopy2021 <- nba2021
rownames(dpoycopy2021) <- dpoycopy2021[,1]
dpoycopy2021 <- dpoycopy2021[,-1]
dpoycopy2021 <- filter(dpoycopy2021, G > 20)
dpoycopy2021 <- subset(dpoycopy2021, select = -c(Season, Pos, Age, Tm, MVP))

xdpoy2021 <- model.matrix(DPOY ~.^2 , dpoycopy2021)[,-1]
ydpoy2021 <- dpoycopy2021$DPOY
predpoy2021 <- predict(lassodpoy, newx = xdpoy2021)
dpoyresult2021 <- cbind(dpoycopy2021, "s0" = round(predpoy2021,8))
dpoyresult2021 <- dpoyresult2021[order(-dpoyresult2021$s0),]
dpoyresult2021 <- subset(dpoyresult2021, select = c(s0))
names(dpoyresult2021)[names(dpoyresult2021) == 's0'] <- "Predicted"

#Predicted 2021 SMOY
smoycopy2021 <- smoy2021
rownames(smoycopy2021) <- smoycopy2021[,1]
smoycopy2021 <- smoycopy2021[,-1]
smoycopy2021 <- filter(smoycopy2021, G > 20)
smoycopy2021 <- subset(smoycopy2021, select = -c(Season, Pos, Age, Tm))

xsmoy2021 <- model.matrix(SMOY ~.^2 , smoycopy2021)[,-1]
ysmoy2021 <- smoycopy2021$SMOY
presmoy2021 <- predict(lassosmoy, newx = xsmoy2021)
```

```
smoyresult2021 <- cbind(smoycopy2021, "s0" = round(presmoy2021,8))
smoyresult2021 <- smoyresult2021[order(-smoyresult2021$s0),]
smoyresult2021 <- subset(smoyresult2021, select = c(s0))
names(smoyresult2021)[names(smoyresult2021) == 's0'] <- "Predicted"

write.csv(roy2021, "roy2021.csv")
#Predicted 2021 ROY
roycopy21 <- roy2021
rownames(roycopy21) <- roycopy21[,1]
roycopy21 <- roycopy21[,-1]
roycopy21 <- filter(roycopy21, G > 20)
roycopy21 <- subset(roycopy21, select = -c(Season, Pos, Age, Tm))

xroy2021 <- model.matrix(ROY ~.^2 , roycopy21)[,-1]
yroy2021 <- roycopy21$ROY
predroy2021 <- predict(lassoroy, newx = xroy2021)
royresult2021 <- cbind(roycopy21, "s0" = round(predroy2021,8))
royresult2021 <- royresult2021[order(-royresult2021$s0),]
royresult2021 <- subset(royresult2021, select = c(s0))
names(royresult2021)[names(royresult2021) == 's0'] <- "Predicted"

mip21$MIP <- 0
mipcopy2021 <- mip21
rownames(mipcopy2021) <- mipcopy2021[,1]
mipcopy2021 <- mipcopy2021[,-1]
mipcopy2021 <- filter(mipcopy2021)

xmip2021 <- model.matrix(MIP ~.^2 , mipcopy2021)[,-1]
ymip2021 <- mipcopy2021$MIP
predmip2021 <- predict(ridge_regsmip, newx = xmip2021)
mipresult2021 <- cbind(mipcopy2021, "s0" = round(predmip2021,8))
mipresult2021 <- mipresult2021[order(-mipresult2021$s0),]
mipresult2021 <- subset(mipresult2021, select = c(s0))
names(mipresult2021)[names(mipresult2021) == 's0'] <- "Predicted"
```