Capstone\_Project\_Scipt

Justin Carver

November 8, 2018

## R Markdown

#Load Player Data  
library(readr)

## Warning: package 'readr' was built under R version 3.4.4

dfn\_2017\_11\_29 <- read\_csv("projections/dailyfantasynerd/dfn\_2017-11-29.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )

## See spec(...) for full column specifications.

dfn\_2017\_12\_25 <- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-25.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Likes = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## PS = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_character(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_23 <- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-23.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_22 <- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-22.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_20 <- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-20.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## Rest = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_19<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-19.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_16<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-16.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_14<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-14.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_13<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-13.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_12<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-12.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_06<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-06.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_05<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-05.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_04<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-04.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_02<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-02.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_12\_01<- read\_csv("projections/dailyfantasynerd/dfn\_2017-12-01.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_11\_30<- read\_csv("projections/dailyfantasynerd/dfn\_2017-11-30.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_11\_28<- read\_csv("projections/dailyfantasynerd/dfn\_2017-11-28.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

dfn\_2017\_11\_27<- read\_csv("projections/dailyfantasynerd/dfn\_2017-11-27.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_integer(),  
## `Player Name` = col\_character(),  
## Inj = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Opp DvP` = col\_character(),  
## `L5 FP` = col\_double(),  
## `S FP` = col\_double(),  
## `Floor FP` = col\_double(),  
## `Ceil FP` = col\_double(),  
## `Proj Min` = col\_double(),  
## `Proj FP` = col\_double(),  
## `Proj Val` = col\_double(),  
## `Actual Min` = col\_double(),  
## `Actual FP` = col\_double(),  
## `Actual Val` = col\_double()  
## )  
## See spec(...) for full column specifications.

## Including Plots

You can also embed plots, for example:

#More data loading  
player\_results <- read\_csv("2017-11-27/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )

## See spec(...) for full column specifications.

player\_results11\_27<-player\_results  
vector<-rep("11-27",nrow(player\_results11\_27))  
player\_results11\_27<-cbind(player\_results11\_27,vector)  
player\_results11\_28<- read\_csv("2017-11-28/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("11-28",nrow(player\_results11\_28))  
player\_results11\_28<-cbind(player\_results11\_28,vector)  
player\_results11\_29<- read\_csv("2017-11-29/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("11-29",nrow(player\_results11\_29))  
player\_results11\_29<-cbind(player\_results11\_29,vector)  
player\_results11\_30<- read\_csv("2017-11-30/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("11-30",nrow(player\_results11\_30))  
player\_results11\_30<-cbind(player\_results11\_30,vector)  
player\_results12\_01<- read\_csv("2017-12-01/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-01",nrow(player\_results12\_01))  
player\_results12\_01<-cbind(player\_results12\_01,vector)  
player\_results12\_02<- read\_csv("2017-12-02/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-02",nrow(player\_results12\_02))  
player\_results12\_02<-cbind(player\_results12\_02,vector)  
player\_results12\_04<- read\_csv("2017-12-04/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-04",nrow(player\_results12\_04))  
player\_results12\_04<-cbind(player\_results12\_04,vector)  
player\_results12\_05<- read\_csv("2017-12-05/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-05",nrow(player\_results12\_05))  
player\_results12\_05<-cbind(player\_results12\_05,vector)  
player\_results12\_06<- read\_csv("2017-12-06/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-06",nrow(player\_results12\_06))  
player\_results12\_06<-cbind(player\_results12\_06,vector)  
player\_results12\_12<- read\_csv("2017-12-12/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-12",nrow(player\_results12\_12))  
player\_results12\_12<-cbind(player\_results12\_12,vector)  
player\_results12\_13<- read\_csv("2017-12-13/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-13",nrow(player\_results12\_13))  
player\_results12\_13<-cbind(player\_results12\_13,vector)  
player\_results12\_14<- read\_csv("2017-12-14/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-14",nrow(player\_results12\_14))  
player\_results12\_14<-cbind(player\_results12\_14,vector)  
player\_results12\_16<- read\_csv("2017-12-16/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-16",nrow(player\_results12\_16))  
player\_results12\_16<-cbind(player\_results12\_16,vector)  
player\_results12\_19<- read\_csv("2017-12-19/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## `Proj Mins` = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-19",nrow(player\_results12\_19))  
player\_results12\_19<-cbind(player\_results12\_19,vector)  
player\_results12\_20<- read\_csv("2017-12-20/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## `Proj Mins` = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-20",nrow(player\_results12\_20))  
player\_results12\_20<-cbind(player\_results12\_20,vector)  
player\_results12\_22<- read\_csv("2017-12-22/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-22",nrow(player\_results12\_22))  
player\_results12\_22<-cbind(player\_results12\_22,vector)  
player\_results12\_23<- read\_csv("2017-12-23/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-23",nrow(player\_results12\_23))  
player\_results12\_23<-cbind(player\_results12\_23,vector)  
player\_results12\_25<- read\_csv("2017-12-25/player\_results.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## Player = col\_character(),  
## Pos = col\_character(),  
## Team = col\_character(),  
## Opp = col\_character(),  
## `Def v Pos` = col\_character(),  
## Salary = col\_integer(),  
## Exposure = col\_character(),  
## Con. = col\_integer(),  
## Mins = col\_integer()  
## )  
## See spec(...) for full column specifications.

vector<-rep("12-25",nrow(player\_results12\_25))  
player\_results12\_25<-cbind(player\_results12\_25,vector)  
player\_results<- rbind(player\_results11\_27,player\_results11\_28,player\_results11\_29,player\_results11\_30,player\_results12\_01,player\_results12\_02,player\_results12\_04,player\_results12\_05,player\_results12\_06, player\_results12\_12,player\_results12\_13,player\_results12\_14,player\_results12\_16,player\_results12\_19,player\_results12\_20,player\_results12\_22,player\_results12\_23,player\_results12\_25,player\_results12\_25)

final\_player\_results\_11\_27<-merge(player\_results11\_27,dfn\_2017\_11\_27, by=1)  
final\_player\_results\_11\_28<-merge(player\_results11\_28,dfn\_2017\_11\_28, by=1)  
final\_player\_results\_11\_29<-merge(player\_results11\_29,dfn\_2017\_11\_29, by=1)  
final\_player\_results\_11\_30<-merge(player\_results11\_30,dfn\_2017\_11\_30, by=1)  
final\_player\_results\_12\_01<-merge(player\_results12\_01,dfn\_2017\_12\_01, by=1)  
final\_player\_results\_12\_02<-merge(player\_results12\_02,dfn\_2017\_12\_02, by=1)  
final\_player\_results\_12\_04<-merge(player\_results12\_04,dfn\_2017\_12\_04, by=1)  
final\_player\_results\_12\_05<-merge(player\_results12\_05,dfn\_2017\_12\_05, by=1)  
final\_player\_results\_12\_06<-merge(player\_results12\_06,dfn\_2017\_12\_06, by=1)  
final\_player\_results\_12\_12<-merge(player\_results12\_12,dfn\_2017\_12\_12, by=1)  
final\_player\_results\_12\_13<-merge(player\_results12\_13,dfn\_2017\_12\_13, by=1)  
final\_player\_results\_12\_14<-merge(player\_results12\_14,dfn\_2017\_12\_14, by=1)  
final\_player\_results\_12\_16<-merge(player\_results12\_16,dfn\_2017\_12\_16, by=1)  
final\_player\_results\_12\_19<-merge(player\_results12\_19,dfn\_2017\_12\_19, by=1)  
final\_player\_results\_12\_20<-merge(player\_results12\_20,dfn\_2017\_12\_20, by=1)  
final\_player\_results\_12\_22<-merge(player\_results12\_22,dfn\_2017\_12\_22, by=1)  
final\_player\_results\_12\_23<-merge(player\_results12\_23,dfn\_2017\_12\_23, by=1)  
final\_player\_results\_12\_25<-merge(player\_results12\_25,dfn\_2017\_12\_25, by=1)

contest\_standings11\_28 <-read\_csv("2017-11-28/$8.00entry\_NBA $450K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings11\_29 <- read\_csv("2017-11-29/$8.88entry\_NBA $600K Mini Backboard Breaker [$100K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings11\_30 <- read\_csv("2017-11-30/$0.25entry\_NBA $10K Quarter Jukebox [Just $0.25!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_01 <- read\_csv("2017-12-01/$8.00entry\_NBA $475K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_02 <- read\_csv("2017-12-02/$8.00entry\_NBA $350K Excellent 8s [$50K to 1st]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_04 <- read\_csv("2017-12-04/$8.00entry\_NBA $425K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_05 <- read\_csv("2017-12-05/$8.00entry\_NBA $450K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_06 <- read\_csv("2017-12-06/$0.25entry\_NBA $10K Quarter Jukebox [Just $0.25!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_12<- read\_csv("2017-12-12/$0.25entry\_NBA $10K Quarter Jukebox [Just $0.25!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_13<- read\_csv("2017-12-13/$8.00entry\_NBA $425K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_14 <- read\_csv("2017-12-14/$8.00entry\_NBA $450K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_16 <- read\_csv("2017-12-16/$8.00entry\_NBA $350K Excellent 8s [$50K to 1st]/contest-standings.csv")

## Warning: Missing column names filled in: 'X7' [7]

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X7 = col\_character(),  
## Player = col\_character(),  
## `Roster Position` = col\_character(),  
## `%Drafted` = col\_character(),  
## FPTS = col\_double()  
## )

## Warning in rbind(names(probs), probs\_f): number of columns of result is not  
## a multiple of vector length (arg 1)

## Warning: 50128 parsing failures.  
## row # A tibble: 5 x 5 col row col expected actual file expected <int> <chr> <chr> <chr> <chr> actual 1 216 <NA> 11 columns 9 colu~ '2017-12-16/$8.00entry\_NBA $350K Excelle~ file 2 217 <NA> 11 columns 9 colu~ '2017-12-16/$8.00entry\_NBA $350K Excelle~ row 3 218 <NA> 11 columns 9 colu~ '2017-12-16/$8.00entry\_NBA $350K Excelle~ col 4 219 <NA> 11 columns 9 colu~ '2017-12-16/$8.00entry\_NBA $350K Excelle~ expected 5 220 <NA> 11 columns 9 colu~ '2017-12-16/$8.00entry\_NBA $350K Excelle~  
## ... ................. ... .......................................................................... ........ .......................................................................... ...... .......................................................................... .... .......................................................................... ... .......................................................................... ... .......................................................................... ........ ..........................................................................  
## See problems(...) for more details.

contest\_standings12\_19 <- read\_csv("2017-12-19/$8.00entry\_NBA $425K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_20 <- read\_csv("2017-12-20/$0.25entry\_NBA $10K Quarter Jukebox [Just $0.25!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_22 <- read\_csv("2017-12-22/$8.00entry\_NBA $350K Excellent 8s [$50K to 1st]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_23 <- read\_csv("2017-12-23/$8.00entry\_NBA $300K Excellent 8ÔÇÖs [$50K to 1st]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_25 <- read\_csv("2017-12-25/$8.00entry\_NBA $700K Holiday Classic [$100K to 1st!]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_26 <- read\_csv("2017-12-26/$8.00entry\_NBA $1M Big FREEZE [$150K to 1st]/contest-standings.csv")

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X = col\_character(),  
## Player = col\_character(),  
## Roster.Position = col\_character(),  
## X.Drafted = col\_character(),  
## FPTS = col\_double()  
## )

contest\_standings12\_28 <- read\_csv("2017-12-28/$8.00entry\_NBA $425K Excellent 8ÔÇÖs [$50K to 1st!]/contest-standings.csv")

## Warning: Missing column names filled in: 'X7' [7]

## Parsed with column specification:  
## cols(  
## Rank = col\_integer(),  
## EntryId = col\_integer(),  
## EntryName = col\_character(),  
## TimeRemaining = col\_integer(),  
## Points = col\_double(),  
## Lineup = col\_character(),  
## X7 = col\_character(),  
## Player = col\_character(),  
## `Roster Position` = col\_character(),  
## `%Drafted` = col\_character(),  
## FPTS = col\_double()  
## )

## Warning in rbind(names(probs), probs\_f): number of columns of result is not  
## a multiple of vector length (arg 1)

## Warning: 62354 parsing failures.  
## row # A tibble: 5 x 5 col row col expected actual file expected <int> <chr> <chr> <chr> <chr> actual 1 147 <NA> 11 columns 9 colu~ '2017-12-28/$8.00entry\_NBA $425K Excelle~ file 2 148 <NA> 11 columns 9 colu~ '2017-12-28/$8.00entry\_NBA $425K Excelle~ row 3 149 <NA> 11 columns 9 colu~ '2017-12-28/$8.00entry\_NBA $425K Excelle~ col 4 150 <NA> 11 columns 9 colu~ '2017-12-28/$8.00entry\_NBA $425K Excelle~ expected 5 151 <NA> 11 columns 9 colu~ '2017-12-28/$8.00entry\_NBA $425K Excelle~  
## ... ................. ... .......................................................................... ........ .......................................................................... ...... .......................................................................... .... .......................................................................... ... .......................................................................... ... .......................................................................... ........ ..........................................................................  
## See problems(...) for more details.

contest\_standings11\_28<-contest\_standings11\_28[,8:11]  
contest\_standings11\_29<-contest\_standings11\_29[,8:11]  
contest\_standings11\_30<-contest\_standings11\_30[,8:11]  
contest\_standings12\_01<-contest\_standings12\_01[,8:11]  
contest\_standings12\_02<-contest\_standings12\_02[,8:11]  
contest\_standings12\_04<-contest\_standings12\_04[,8:11]  
contest\_standings12\_05<-contest\_standings12\_05[,8:11]  
contest\_standings12\_06<-contest\_standings12\_06[,8:11]  
contest\_standings12\_12<-contest\_standings12\_12[,8:11]  
contest\_standings12\_13<-contest\_standings12\_13[,8:11]  
contest\_standings12\_14<-contest\_standings12\_14[,8:11]  
contest\_standings12\_16<-contest\_standings12\_16[,8:11]  
contest\_standings12\_19<-contest\_standings12\_19[,8:11]  
contest\_standings12\_20<-contest\_standings12\_20[,8:11]  
contest\_standings12\_22<-contest\_standings12\_22[,8:11]  
contest\_standings12\_23<-contest\_standings12\_23[,8:11]  
contest\_standings12\_25<-contest\_standings12\_25[,8:11]  
contest\_standings12\_26<-contest\_standings12\_26[,8:11]  
contest\_standings12\_28<-contest\_standings12\_28[,8:11]

#Merge data  
players\_with\_ownership11\_28<-merge(final\_player\_results\_11\_28,contest\_standings11\_28,1)  
players\_with\_ownership11\_29<-merge(final\_player\_results\_11\_29,contest\_standings11\_29,1)  
players\_with\_ownership11\_30<-merge(final\_player\_results\_11\_30,contest\_standings11\_30,1)  
players\_with\_ownership12\_01<-merge(final\_player\_results\_12\_01,contest\_standings12\_01,1)  
players\_with\_ownership12\_02<-merge(final\_player\_results\_12\_02,contest\_standings12\_02,1)  
players\_with\_ownership12\_04<-merge(final\_player\_results\_12\_04,contest\_standings12\_04,1)  
players\_with\_ownership12\_05<-merge(final\_player\_results\_12\_05,contest\_standings12\_05,1)  
players\_with\_ownership12\_06<-merge(final\_player\_results\_12\_06,contest\_standings12\_06,1)  
players\_with\_ownership12\_12<-merge(final\_player\_results\_12\_12,contest\_standings12\_12,1)  
players\_with\_ownership12\_13<-merge(final\_player\_results\_12\_13,contest\_standings12\_13,1)  
players\_with\_ownership12\_14<-merge(final\_player\_results\_12\_14,contest\_standings12\_14,1)  
players\_with\_ownership12\_16<-merge(final\_player\_results\_12\_16,contest\_standings12\_16,1)  
names(players\_with\_ownership12\_16)<-names(players\_with\_ownership12\_14)  
players\_with\_ownership12\_19<-merge(final\_player\_results\_12\_19,contest\_standings12\_19,1)  
players\_with\_ownership12\_20<-merge(final\_player\_results\_12\_20,contest\_standings12\_20,1)  
players\_with\_ownership12\_22<-merge(final\_player\_results\_12\_22,contest\_standings12\_22,1)  
players\_with\_ownership12\_23<-merge(final\_player\_results\_12\_23,contest\_standings12\_23,1)  
players\_with\_ownership12\_25<-merge(final\_player\_results\_12\_25,contest\_standings12\_25,1)

#Setting up variables  
  
players\_with\_ownership\_final<-rbind(players\_with\_ownership11\_28,players\_with\_ownership11\_29,players\_with\_ownership11\_30,players\_with\_ownership12\_01,players\_with\_ownership12\_02,players\_with\_ownership12\_04,players\_with\_ownership12\_05,players\_with\_ownership12\_06,players\_with\_ownership12\_12,players\_with\_ownership12\_13,players\_with\_ownership12\_14,players\_with\_ownership12\_16,players\_with\_ownership12\_19,players\_with\_ownership12\_20,players\_with\_ownership12\_22,players\_with\_ownership12\_23,players\_with\_ownership12\_25)  
  
players\_with\_ownership\_final2<-players\_with\_ownership\_final  
  
players\_with\_ownership\_final2$X.Drafted<-gsub("%","",players\_with\_ownership\_final2$X.Drafted)  
players\_with\_ownership\_final2$`Opp DvP`<-gsub("%","",players\_with\_ownership\_final2$`Opp DvP`)  
  
players\_with\_ownership\_final2$X.Drafted<-as.numeric(players\_with\_ownership\_final2$X.Drafted)  
players\_with\_ownership\_final2$`Opp DvP`<-as.numeric(players\_with\_ownership\_final2$`Opp DvP`)  
  
players\_with\_ownership\_final2<-players\_with\_ownership\_final2[,-c(16,17,18,27,28,29,30,33,48,49,50,51,52,53,54,56)]  
  
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.4

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(plyr)

## Warning: package 'plyr' was built under R version 3.4.4

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

players\_with\_ownership\_final2<- rename(players\_with\_ownership\_final2, c("Pos.x" = "Pos"))  
players\_with\_ownership\_final2<- rename(players\_with\_ownership\_final2, c("Team.x" = "Team"))  
players\_with\_ownership\_final2<- rename(players\_with\_ownership\_final2, c("Opp.x" = "Opp"))  
players\_with\_ownership\_final2<- rename(players\_with\_ownership\_final2, c("Salary.x" = "Salary"))  
players\_with\_ownership\_final2<- rename(players\_with\_ownership\_final2, c("USG.x" = "USG"))  
players\_with\_ownership\_final2<- rename(players\_with\_ownership\_final2, c("vector" = "Date" ))  
players\_with\_ownership\_final2<- rename(players\_with\_ownership\_final2, c("X.Drafted" = "Draft\_Percentage"))  
  
library(tidyr)

## Warning: package 'tidyr' was built under R version 3.4.4

players\_with\_ownership\_final2<-separate(players\_with\_ownership\_final2,Opp,c("Location", "Opp"), sep=" ")  
players\_with\_ownership\_final2$Location<-ifelse(players\_with\_ownership\_final2$Location =="@","Road","Home")  
  
players\_with\_ownership\_final2$Location<-as.factor(players\_with\_ownership\_final2$Location)  
players\_with\_ownership\_final2$`Def v Pos`<-as.factor(players\_with\_ownership\_final2$`Def v Pos`)  
  
players\_with\_ownership\_final2<-players\_with\_ownership\_final2[,-18]  
players\_with\_ownership\_final2$Rest<-as.factor(players\_with\_ownership\_final2$Rest)  
  
outwithinjuries<-which(players\_with\_ownership\_final2$Inj=="O")  
players\_with\_ownership\_final2<-players\_with\_ownership\_final2[-outwithinjuries,]  
  
players\_with\_ownership\_final2<-players\_with\_ownership\_final2[,-23]  
players\_with\_ownership\_final2$PS<-as.numeric(players\_with\_ownership\_final2$PS)  
  
players\_with\_ownership\_final2<-players\_with\_ownership\_final2[,-22]  
  
players\_with\_ownership\_final2<-players\_with\_ownership\_final2[complete.cases(players\_with\_ownership\_final2),]  
  
GPPViable<-ifelse(players\_with\_ownership\_final2$Val >= 6, "Yes", "No")  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2,GPPViable)

#Dummy Variables  
library(psych)

## Warning: package 'psych' was built under R version 3.4.4

vectorofdifference<-players\_with\_ownership\_final2$`L5 FP`- players\_with\_ownership\_final2$`S FP`  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, vectorofdifference)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("vectorofdifference" = "FPDifferenceL5"))  
  
percentageofdifference<-(players\_with\_ownership\_final2$FPDifferenceL5/players\_with\_ownership\_final2$`S FP`)\*100  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, percentageofdifference)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("percentageofdifference"="FPPercentageL5"))  
  
players\_with\_ownership\_final2$FPDifferenceL5<-scale(players\_with\_ownership\_final2$FPDifferenceL5)  
  
vectorofdifference1<-players\_with\_ownership\_final2$`L5 FGA`-   
players\_with\_ownership\_final2$`S FGA`  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, vectorofdifference1)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("vectorofdifference1"= "FGADifferenceL5"))  
  
percentageofdifference1<-(players\_with\_ownership\_final2$FGADifferenceL5/players\_with\_ownership\_final2$`S FGA`)\*100  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, percentageofdifference1)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("percentageofdifference1"= "FGAPercentageL5"))  
  
players\_with\_ownership\_final2$FGADifferenceL5<-scale(players\_with\_ownership\_final2$FGADifferenceL5)  
  
vectorofdifference2<-players\_with\_ownership\_final2$`L2 FGA`-   
players\_with\_ownership\_final2$`S FGA`  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, vectorofdifference2)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("vectorofdifference2" = "FGADifferenceL2"))  
  
percentageofdifference2<-(players\_with\_ownership\_final2$FGADifferenceL2/players\_with\_ownership\_final2$`S FGA`)\*100  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, percentageofdifference2)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("percentageofdifference2"= "FGAPercentageL2"))  
  
players\_with\_ownership\_final2$FGADifferenceL2<-scale(players\_with\_ownership\_final2$FGADifferenceL2)  
  
vectorofdifference3<-players\_with\_ownership\_final2$`L5 Min`- players\_with\_ownership\_final2$`S Min`  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, vectorofdifference3)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("vectorofdifference3" = "MinDifferenceL5"))  
  
percentageofdifference3<-(players\_with\_ownership\_final2$MinDifferenceL5/players\_with\_ownership\_final2$`S Min`)\*100  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, percentageofdifference3)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("percentageofdifference3" = "MinPercentageL5"))  
  
players\_with\_ownership\_final2$MinDifferenceL5<-scale(players\_with\_ownership\_final2$MinDifferenceL5)  
  
vectorofdifference4<-players\_with\_ownership\_final2$`L2 Min`- players\_with\_ownership\_final2$`S Min`  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, vectorofdifference4)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("vectorofdifference4" = "MinDifferenceL2"))  
  
percentageofdifference4<-(players\_with\_ownership\_final2$MinDifferenceL2/players\_with\_ownership\_final2$`S Min`)\*100  
  
players\_with\_ownership\_final2<-cbind(players\_with\_ownership\_final2, percentageofdifference4)  
  
players\_with\_ownership\_final2<-rename(players\_with\_ownership\_final2, c("percentageofdifference4" = "MinPercentageL2"))  
  
players\_with\_ownership\_final2$MinDifferenceL2<-scale(players\_with\_ownership\_final2$MinDifferenceL2)  
  
draftkingsdata<-players\_with\_ownership\_final2[, -c(1,3,5,8,9,10,16,18,19,21,23,36,37)]  
  
draftkingsdata<-draftkingsdata[draftkingsdata$Draft\_Percentage!=0,]  
  
summary(draftkingsdata)

## Pos Location Def v Pos VegasPts   
## Length:2102 Home:1046 10th : 88 Min. : 93.25   
## Class :character Road:1056 23rd : 88 1st Qu.:100.75   
## Mode :character 12th : 84 Median :104.25   
## 21st : 84 Mean :104.59   
## 30th : 81 3rd Qu.:108.00   
## 11th : 80 Max. :120.50   
## (Other):1597   
## FPG FPPM USG FGA   
## Min. : 5.10 Min. :0.4300 Min. : 6.60 Min. : 1.00   
## 1st Qu.:14.20 1st Qu.:0.7700 1st Qu.:15.90 1st Qu.: 5.10   
## Median :19.80 Median :0.8800 Median :19.20 Median : 7.75   
## Mean :22.03 Mean :0.9027 Mean :19.71 Mean : 8.59   
## 3rd Qu.:27.70 3rd Qu.:1.0100 3rd Qu.:23.20 3rd Qu.:11.10   
## Max. :58.10 Max. :1.6500 Max. :36.80 Max. :21.20   
##   
## MPG Con. Val Rest PER   
## Min. : 7.20 Min. : 0.00 Min. : 0.100 0: 450 Min. : 1.00   
## 1st Qu.:17.40 1st Qu.:48.00 1st Qu.: 3.100 1:1215 1st Qu.:11.00   
## Median :24.05 Median :58.00 Median : 4.400 2: 322 Median :14.00   
## Mean :23.79 Mean :55.43 Mean : 4.522 3: 107 Mean :14.61   
## 3rd Qu.:29.90 3rd Qu.:67.00 3rd Qu.: 5.800 4: 8 3rd Qu.:17.00   
## Max. :37.50 Max. :88.00 Max. :14.800 Max. :32.00   
##   
## Opp Pace Opp DEff Opp DvP L2 FGA   
## Min. : 95.00 Min. : 98.0 Min. :-21.00000 Min. : 0.000   
## 1st Qu.: 98.00 1st Qu.:103.0 1st Qu.: -5.00000 1st Qu.: 6.000   
## Median : 99.00 Median :106.0 Median : 0.00000 Median : 8.000   
## Mean : 99.92 Mean :105.1 Mean : -0.08801 Mean : 9.309   
## 3rd Qu.:101.00 3rd Qu.:108.0 3rd Qu.: 4.00000 3rd Qu.:12.000   
## Max. :106.00 Max. :110.0 Max. : 18.00000 Max. :31.000   
##   
## L5 FGA S FGA L2 Min L5 Min   
## Min. : 1.000 Min. : 1.000 Min. : 3.00 Min. : 4.00   
## 1st Qu.: 6.000 1st Qu.: 5.000 1st Qu.:20.00 1st Qu.:20.00   
## Median : 8.000 Median : 8.000 Median :26.00 Median :26.00   
## Mean : 8.995 Mean : 8.736 Mean :25.57 Mean :25.44   
## 3rd Qu.:12.000 3rd Qu.:11.000 3rd Qu.:32.00 3rd Qu.:32.00   
## Max. :26.000 Max. :21.000 Max. :44.00 Max. :42.00   
##   
## S Min L5 FP S FP Draft\_Percentage  
## Min. : 8.0 Min. : 1.20 Min. : 5.10 Min. : 0.010   
## 1st Qu.:18.0 1st Qu.:14.80 1st Qu.:14.30 1st Qu.: 0.420   
## Median :25.0 Median :21.30 Median :20.15 Median : 2.360   
## Mean :24.6 Mean :23.00 Mean :22.32 Mean : 5.744   
## 3rd Qu.:31.0 3rd Qu.:29.25 3rd Qu.:28.30 3rd Qu.: 7.555   
## Max. :39.0 Max. :67.50 Max. :58.80 Max. :62.350   
##   
## GPPViable FPDifferenceL5.V1 FPPercentageL5 FGADifferenceL5.V1   
## No :1612 Min. :-3.092231 Min. :-84.810 Min. :-3.336983   
## Yes: 490 1st Qu.:-0.674055 1st Qu.:-10.240 1st Qu.:-0.796400   
## Median :-0.087120 Median : 1.355 Median :-0.161255   
## Mean : 0.003238 Mean : 4.376 Mean : 0.003424   
## 3rd Qu.: 0.617203 3rd Qu.: 16.000 3rd Qu.: 0.473891   
## Max. : 3.974475 Max. :149.057 Max. : 4.919911   
##   
## FGAPercentageL5 FGADifferenceL2.V1 FGAPercentageL2   
## Min. :-66.667 Min. :-3.911761 Min. :-100.000   
## 1st Qu.:-10.000 1st Qu.:-0.639146 1st Qu.: -12.500   
## Median : 0.000 Median :-0.230070 Median : 0.000   
## Mean : 4.206 Mean : 0.004245 Mean : 9.375   
## 3rd Qu.: 16.667 3rd Qu.: 0.588084 3rd Qu.: 25.000   
## Max. :150.000 Max. : 4.678852 Max. : 300.000   
##   
## MinDifferenceL5.V1 MinPercentageL5 MinDifferenceL2.V1   
## Min. :-3.935548 Min. :-54.545 Min. :-3.490802   
## 1st Qu.:-0.561953 1st Qu.: -4.167 1st Qu.:-0.645815   
## Median : 0.051428 Median : 2.857 Median : 0.010721   
## Mean : 0.002550 Mean : 4.701 Mean : 0.005203   
## 3rd Qu.: 0.358118 3rd Qu.: 10.714 3rd Qu.: 0.667256   
## Max. : 4.345094 Max. :111.111 Max. : 4.825315   
##   
## MinPercentageL2   
## Min. :-76.471   
## 1st Qu.: -7.692   
## Median : 3.333   
## Mean : 5.682   
## 3rd Qu.: 14.788   
## Max. :187.500   
##

str(draftkingsdata)

## 'data.frame': 2102 obs. of 36 variables:  
## $ Pos : chr "PG/SGPGSG" "PF/CPFC" "SF/PFSFPF" "PG" ...  
## $ Location : Factor w/ 2 levels "Home","Road": 1 2 1 1 2 1 1 2 1 1 ...  
## $ Def v Pos : Factor w/ 30 levels "10th","11th",..: 4 16 19 14 10 17 23 26 23 24 ...  
## $ VegasPts : num 102 106 108 107 102 ...  
## $ FPG : num 13.4 24.1 30.4 12.3 9.5 26.4 17.6 38.6 22.1 10.6 ...  
## $ FPPM : num 0.79 1.07 0.84 0.75 0.79 1.18 0.76 1.11 1 0.92 ...  
## $ USG : num 20.8 16.1 23.8 27 11.6 26.3 20.4 29.3 25.3 18.1 ...  
## $ FGA : num 6.4 5.7 15.5 8 2.6 10.7 8.5 18.4 10.6 3.5 ...  
## $ MPG : num 16.9 22.5 36.2 16.3 12 22.4 23.1 34.8 22.1 11.5 ...  
## $ Con. : int 46 60 76 51 1 67 71 75 53 30 ...  
## $ Val : num 6.3 8.3 3.4 3.5 8.8 3.3 3.2 2.7 3.9 2.8 ...  
## $ Rest : Factor w/ 5 levels "0","1","2","3",..: 3 2 2 2 2 2 1 3 1 1 ...  
## $ PER : int 11 18 15 11 14 19 12 21 17 15 ...  
## $ Opp Pace : int 101 99 100 106 101 106 98 100 98 99 ...  
## $ Opp DEff : int 106 109 104 109 109 109 107 108 107 103 ...  
## $ Opp DvP : num -2 -1 -5 -1 -2 15 -1 9 -1 -2 ...  
## $ L2 FGA : int 8 6 16 11 2 10 8 23 12 4 ...  
## $ L5 FGA : int 9 7 16 10 1 10 10 21 9 4 ...  
## $ S FGA : int 6 6 16 8 3 11 9 19 11 3 ...  
## $ L2 Min : int 24 23 40 17 6 24 22 38 21 19 ...  
## $ L5 Min : int 23 23 40 19 6 22 24 38 19 17 ...  
## $ S Min : int 17 23 37 17 12 23 24 37 23 11 ...  
## $ L5 FP : num 17.7 27.1 31.4 13.8 4.5 23.6 19.3 40.1 18.1 15.1 ...  
## $ S FP : num 13.4 24.1 30.4 12.3 9.5 26.4 17.6 40 22.1 10.1 ...  
## $ Draft\_Percentage: num 0.49 2.99 9.28 0.63 0.01 ...  
## $ GPPViable : Factor w/ 2 levels "No","Yes": 2 2 1 1 2 1 1 1 1 1 ...  
## $ FPDifferenceL5 : num [1:2102, 1] 0.852 0.5468 0.0772 0.1946 -1.3314 ...  
## $ FPPercentageL5 : num 32.09 12.45 3.29 12.2 -52.63 ...  
## $ FGADifferenceL5 : num [1:2102, 1] 1.744 0.474 -0.161 1.109 -1.432 ...  
## $ FGAPercentageL5 : num 50 16.7 0 25 -66.7 ...  
## $ FGADifferenceL2 : num [1:2102, 1] 0.588 -0.23 -0.23 0.997 -0.639 ...  
## $ FGAPercentageL2 : num 33.3 0 0 37.5 -33.3 ...  
## $ MinDifferenceL5 : num [1:2102, 1] 1.585 -0.255 0.665 0.358 -2.095 ...  
## $ MinPercentageL5 : num 35.29 0 8.11 11.76 -50 ...  
## $ MinDifferenceL2 : num [1:2102, 1] 1.324 -0.208 0.448 -0.208 -1.521 ...  
## $ MinPercentageL2 : num 41.18 0 8.11 0 -50 ...

describe(draftkingsdata)

## Warning in describe(draftkingsdata): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning  
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning  
## -Inf

## vars n mean sd median trimmed mad min  
## Pos\* 1 2102 NaN NA NA NaN NA Inf  
## Location\* 2 2102 1.50 0.50 2.00 1.50 0.00 1.00  
## Def v Pos\* 3 2102 15.23 8.73 15.00 15.20 11.86 1.00  
## VegasPts 4 2102 104.59 5.53 104.25 104.31 5.19 93.25  
## FPG 5 2102 22.03 10.23 19.80 20.99 9.71 5.10  
## FPPM 6 2102 0.90 0.21 0.88 0.89 0.19 0.43  
## USG 7 2102 19.71 5.43 19.20 19.43 5.19 6.60  
## FGA 8 2102 8.59 4.34 7.75 8.21 4.23 1.00  
## MPG 9 2102 23.79 7.31 24.05 23.88 9.27 7.20  
## Con. 10 2102 55.43 16.14 58.00 57.05 13.34 0.00  
## Val 11 2102 4.52 2.08 4.40 4.45 1.93 0.10  
## Rest\* 12 2102 2.05 0.78 2.00 1.99 0.00 1.00  
## PER 13 2102 14.61 4.92 14.00 14.37 4.45 1.00  
## Opp Pace 14 2102 99.92 2.47 99.00 99.69 1.48 95.00  
## Opp DEff 15 2102 105.14 2.78 106.00 105.25 2.97 98.00  
## Opp DvP 16 2102 -0.09 6.56 0.00 -0.12 7.41 -21.00  
## L2 FGA 17 2102 9.31 4.98 8.00 8.91 4.45 0.00  
## L5 FGA 18 2102 9.00 4.68 8.00 8.62 4.45 1.00  
## S FGA 19 2102 8.74 4.43 8.00 8.34 4.45 1.00  
## L2 Min 20 2102 25.57 8.14 26.00 25.81 8.90 3.00  
## L5 Min 21 2102 25.44 7.68 26.00 25.65 8.90 4.00  
## S Min 22 2102 24.60 7.50 25.00 24.71 8.90 8.00  
## L5 FP 23 2102 23.00 11.08 21.30 22.02 10.38 1.20  
## S FP 24 2102 22.32 10.42 20.15 21.27 10.01 5.10  
## Draft\_Percentage 25 2102 5.74 8.39 2.36 3.89 3.31 0.01  
## GPPViable\* 26 2102 1.23 0.42 1.00 1.17 0.00 1.00  
## FPDifferenceL5 27 2102 0.00 1.00 -0.09 -0.04 0.94 -3.09  
## FPPercentageL5 28 2102 4.38 24.52 1.35 2.66 19.02 -84.81  
## FGADifferenceL5 29 2102 0.00 1.00 -0.16 -0.03 0.94 -3.34  
## FGAPercentageL5 30 2102 4.21 24.06 0.00 2.82 18.53 -66.67  
## FGADifferenceL2 31 2102 0.00 1.00 -0.23 -0.02 1.21 -3.91  
## FGAPercentageL2 32 2102 9.38 37.15 0.00 6.84 29.65 -100.00  
## MinDifferenceL5 33 2102 0.00 1.00 0.05 -0.04 0.91 -3.94  
## MinPercentageL5 34 2102 4.70 18.58 2.86 3.16 11.10 -54.55  
## MinDifferenceL2 35 2102 0.01 1.00 0.01 -0.02 0.97 -3.49  
## MinPercentageL2 36 2102 5.68 25.53 3.33 3.80 16.35 -76.47  
## max range skew kurtosis se  
## Pos\* -Inf -Inf NA NA NA  
## Location\* 2.00 1.00 -0.01 -2.00 0.01  
## Def v Pos\* 30.00 29.00 0.01 -1.20 0.19  
## VegasPts 120.50 27.25 0.46 0.11 0.12  
## FPG 58.10 53.00 0.98 0.95 0.22  
## FPPM 1.65 1.22 0.75 0.85 0.00  
## USG 36.80 30.20 0.48 -0.01 0.12  
## FGA 21.20 20.20 0.72 -0.18 0.09  
## MPG 37.50 30.30 -0.09 -1.06 0.16  
## Con. 88.00 88.00 -1.00 1.03 0.35  
## Val 14.80 14.70 0.40 0.34 0.05  
## Rest\* 5.00 4.00 0.76 0.87 0.02  
## PER 32.00 31.00 0.54 0.69 0.11  
## Opp Pace 106.00 11.00 0.75 -0.26 0.05  
## Opp DEff 110.00 12.00 -0.29 -0.92 0.06  
## Opp DvP 18.00 39.00 0.01 -0.03 0.14  
## L2 FGA 31.00 31.00 0.75 0.33 0.11  
## L5 FGA 26.00 25.00 0.73 0.09 0.10  
## S FGA 21.00 20.00 0.72 -0.16 0.10  
## L2 Min 44.00 41.00 -0.25 -0.50 0.18  
## L5 Min 42.00 38.00 -0.24 -0.67 0.17  
## S Min 39.00 31.00 -0.10 -1.04 0.16  
## L5 FP 67.50 66.30 0.90 0.93 0.24  
## S FP 58.80 53.70 0.97 0.90 0.23  
## Draft\_Percentage 62.35 62.34 2.57 8.21 0.18  
## GPPViable\* 2.00 1.00 1.26 -0.41 0.01  
## FPDifferenceL5 3.97 7.07 0.51 0.65 0.02  
## FPPercentageL5 149.06 233.87 0.94 2.84 0.53  
## FGADifferenceL5 4.92 8.26 0.44 0.73 0.02  
## FGAPercentageL5 150.00 216.67 0.94 3.32 0.52  
## FGADifferenceL2 4.68 8.59 0.30 0.95 0.02  
## FGAPercentageL2 300.00 400.00 1.17 4.64 0.81  
## MinDifferenceL5 4.35 8.28 0.49 1.47 0.02  
## MinPercentageL5 111.11 165.66 1.29 4.78 0.41  
## MinDifferenceL2 4.83 8.32 0.35 0.93 0.02  
## MinPercentageL2 187.50 263.97 1.22 4.69 0.56

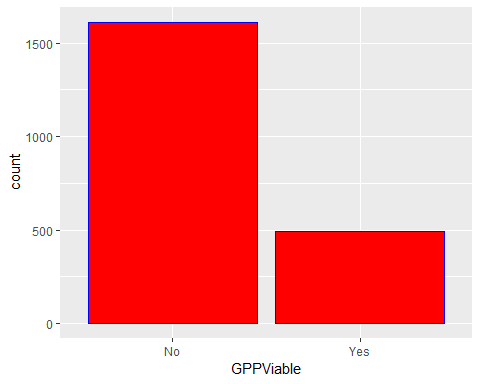
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.4

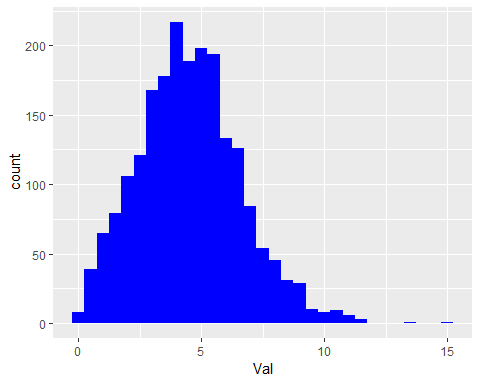
##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

ggplot(data=draftkingsdata, aes(GPPViable))+  
 geom\_bar(aes(GPPViable), colour = "blue", fill= "red")



ggplot(data=draftkingsdata, aes(x=Val))+  
 geom\_histogram(aes(x=Val), binwidth = 0.5, fill="blue")



#Clustering  
  
library(dplyr)  
  
mydata<-model.matrix(~.,data=draftkingsdata)  
cluster\_fit <- kmeans(mydata, 2)  
aggregate(mydata,by=list(cluster\_fit$cluster),FUN=mean)

## Group.1 (Intercept) PosPF PosPF/CPFC PosPG PosPG/SFPGSF  
## 1 1 1 0.04205607 0.2359813 0.06542056 0.018691589  
## 2 2 1 0.01672640 0.1708483 0.11230585 0.008363202  
## PosPG/SGPGSG PosSF PosSF/PFSFPF PosSG PosSG/SFSGSF  
## 1 0.1705607 0.02102804 0.1775701 0.07242991 0.1401869  
## 2 0.1475508 0.01194743 0.1589008 0.05197133 0.2204301  
## LocationRoad `Def v Pos`11th `Def v Pos`12th `Def v Pos`13th  
## 1 0.4929907 0.04205607 0.04439252 0.05841121  
## 2 0.5047790 0.03703704 0.03882915 0.02628435  
## `Def v Pos`14th `Def v Pos`15th `Def v Pos`16th `Def v Pos`17th  
## 1 0.02570093 0.02336449 0.01401869 0.04205607  
## 2 0.03106332 0.03405018 0.03166069 0.02927121  
## `Def v Pos`18th `Def v Pos`19th `Def v Pos`1st `Def v Pos`20th  
## 1 0.01635514 0.02803738 0.03738318 0.02336449  
## 2 0.03703704 0.03763441 0.03106332 0.03046595  
## `Def v Pos`21st `Def v Pos`22nd `Def v Pos`23rd `Def v Pos`24th  
## 1 0.02336449 0.01635514 0.03971963 0.04906542  
## 2 0.04420550 0.02807646 0.04241338 0.02747909  
## `Def v Pos`25th `Def v Pos`26th `Def v Pos`27th `Def v Pos`28th  
## 1 0.03037383 0.03037383 0.03037383 0.04205607  
## 2 0.03524492 0.03405018 0.03046595 0.03524492  
## `Def v Pos`29th `Def v Pos`2nd `Def v Pos`30th `Def v Pos`3rd  
## 1 0.02803738 0.01635514 0.05373832 0.04205607  
## 2 0.03584229 0.02807646 0.03464755 0.02449223  
## `Def v Pos`4th `Def v Pos`5th `Def v Pos`6th `Def v Pos`7th  
## 1 0.03971963 0.01635514 0.04672897 0.02803738  
## 2 0.03166069 0.03464755 0.03524492 0.03106332  
## `Def v Pos`8th `Def v Pos`9th VegasPts FPG FPPM USG  
## 1 0.03971963 0.03504673 104.2868 15.79276 0.8497430 17.62664  
## 2 0.03345281 0.02628435 104.6623 23.63029 0.9161947 20.23883  
## FGA MPG Con. Val Rest1 Rest2 Rest3  
## 1 5.883178 18.50841 42.18692 4.568925 0.5817757 0.1378505 0.04439252  
## 2 9.282497 25.13984 58.81780 4.510275 0.5770609 0.1571087 0.05256870  
## Rest4 PER `Opp Pace` `Opp DEff` `Opp DvP` `L2 FGA` `L5 FGA`  
## 1 0.000000000 13.77804 99.83879 105.1168 -0.37850467 9.009346 7.696262  
## 2 0.004778973 14.82557 99.93489 105.1458 -0.01373955 9.385305 9.327360  
## `S FGA` `L2 Min` `L5 Min` `S Min` `L5 FP` `S FP` Draft\_Percentage  
## 1 5.883178 25.15187 23.45093 19.07710 20.83411 15.84556 5.757547  
## 2 9.465352 25.67742 25.94385 26.00657 23.55800 23.97348 5.740806  
## GPPViableYes FPDifferenceL5 FPPercentageL5 FGADifferenceL5  
## 1 0.2429907 1.0136316 33.722892 0.9903181  
## 2 0.2305854 -0.2550941 -3.127608 -0.2489002  
## FGAPercentageL5 FGADifferenceL2 FGAPercentageL2 MinDifferenceL5  
## 1 33.171358 1.0487733 57.666440 1.0861500  
## 2 -3.199728 -0.2628153 -2.971601 -0.2744992  
## MinPercentageL5 MinDifferenceL2 MinPercentageL2  
## 1 26.1150801 1.1213088 36.292437  
## 2 -0.7738521 -0.2801578 -2.144178

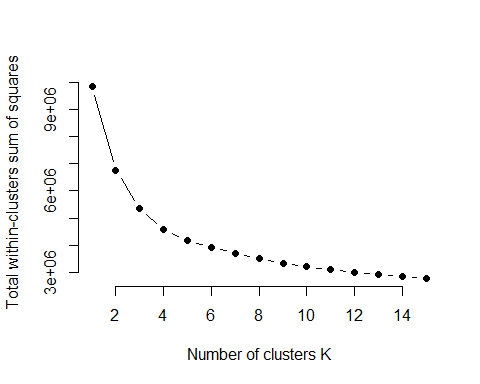
mydata <- data.frame(mydata, cluster\_fit$cluster)  
  
  
k.means.centers<-cluster\_fit$centers  
View(k.means.centers)  
  
mydata.results<-table(mydata$GPPViableYes, cluster\_fit$cluster)  
names(dimnames(mydata.results)) <- c("GPPViable", "Cluster")  
mydata.results

## Cluster  
## GPPViable 1 2  
## 0 324 1288  
## 1 104 386

cluster\_fit$betweenss/cluster\_fit$totss

## [1] 0.3147387

mydata.cluster1<-mydata[mydata$cluster\_fit.cluster==1,]  
mydata.cluster2<-mydata[mydata$cluster\_fit.cluster==2,]  
  
  
k.max<-15  
wss <- sapply(1:k.max,   
 function(k){kmeans(mydata, k, nstart=50,iter.max = 15 )$tot.withinss})  
  
  
plot(1:k.max, wss,  
 type="b", pch = 19, frame = FALSE,   
 xlab="Number of clusters K",  
 ylab="Total within-clusters sum of squares")



kmm = kmeans(mydata[,-77],5,nstart = 50,iter.max = 15)  
kmm.centers<-kmm$centers  
View(kmm.centers)  
  
  
kmm$betweenss/kmm$totss

## [1] 0.577467

nums <- unlist(lapply(draftkingsdata, is.numeric))  
require(dplyr)  
  
  
numericdraftkingsdata<-draftkingsdata[,nums]  
  
head(numericdraftkingsdata)

## VegasPts FPG FPPM USG FGA MPG Con. Val PER Opp Pace Opp DEff Opp DvP  
## 2 102.50 13.4 0.79 20.8 6.4 16.9 46 6.3 11 101 106 -2  
## 3 105.75 24.1 1.07 16.1 5.7 22.5 60 8.3 18 99 109 -1  
## 4 107.75 30.4 0.84 23.8 15.5 36.2 76 3.4 15 100 104 -5  
## 5 106.75 12.3 0.75 27.0 8.0 16.3 51 3.5 11 106 109 -1  
## 6 102.50 9.5 0.79 11.6 2.6 12.0 1 8.8 14 101 109 -2  
## 7 106.75 26.4 1.18 26.3 10.7 22.4 67 3.3 19 106 109 15  
## L2 FGA L5 FGA S FGA L2 Min L5 Min S Min L5 FP S FP Draft\_Percentage  
## 2 8 9 6 24 23 17 17.7 13.4 0.49  
## 3 6 7 6 23 23 23 27.1 24.1 2.99  
## 4 16 16 16 40 40 37 31.4 30.4 9.28  
## 5 11 10 8 17 19 17 13.8 12.3 0.63  
## 6 2 1 3 6 6 12 4.5 9.5 0.01  
## 7 10 10 11 24 22 23 23.6 26.4 4.98  
## FPDifferenceL5 FPPercentageL5 FGADifferenceL5 FGAPercentageL5  
## 2 0.85197743 32.089552 1.7441826 50.000000  
## 3 0.54677085 12.448133 0.4738911 16.666667  
## 4 0.07722227 3.289474 -0.1612546 0.000000  
## 5 0.19460942 12.195122 1.1090369 25.000000  
## 6 -1.33142347 -52.631579 -1.4315461 -66.666667  
## 7 -0.81492003 -10.606061 -0.7964004 -9.090909  
## FGADifferenceL2 FGAPercentageL2 MinDifferenceL5 MinPercentageL5  
## 2 0.5880840 33.333333 1.5848802 35.294118  
## 3 -0.2300696 0.000000 -0.2552624 0.000000  
## 4 -0.2300696 0.000000 0.6648089 8.108108  
## 5 0.9971608 37.500000 0.3581185 11.764706  
## 6 -0.6391463 -33.333333 -2.0954049 -50.000000  
## 7 -0.6391463 -9.090909 -0.5619528 -4.347826  
## MinDifferenceL2 MinPercentageL2  
## 2 1.32379170 41.176471  
## 3 -0.20812455 0.000000  
## 4 0.44841099 8.108108  
## 5 -0.20812455 0.000000  
## 6 -1.52119561 -50.000000  
## 7 0.01072063 4.347826

#nums <- unlist(lapply(train, is.numeric))  
#numerictraindata<-train[,nums]  
#numerictraindata<-numerictraindata[,-22]  
  
Correlations<-cor(numericdraftkingsdata)  
head(cor(numericdraftkingsdata))

## VegasPts FPG FPPM USG FGA MPG  
## VegasPts 1.00000000 0.1177212 0.1618357 0.01009278 0.05919992 0.04772554  
## FPG 0.11772117 1.0000000 0.7745734 0.69801868 0.90316852 0.85255971  
## FPPM 0.16183568 0.7745734 1.0000000 0.70790904 0.59831783 0.36594663  
## USG 0.01009278 0.6980187 0.7079090 1.00000000 0.81123579 0.45906971  
## FGA 0.05919992 0.9031685 0.5983178 0.81123579 1.00000000 0.85655167  
## MPG 0.04772554 0.8525597 0.3659466 0.45906971 0.85655167 1.00000000  
## Con. Val PER Opp Pace Opp DEff  
## VegasPts 0.05927482 0.04863905 0.1634309 0.532954854 0.369734714  
## FPG 0.76767392 0.15764806 0.7262810 0.008792963 -0.002542151  
## FPPM 0.46778098 0.12382468 0.8925496 0.028371871 0.009511191  
## USG 0.48915826 0.11555530 0.5826382 0.010427390 0.014678977  
## FGA 0.72864964 0.15245692 0.5624148 0.006823164 0.003589273  
## MPG 0.81413813 0.16420287 0.3914877 -0.005782479 -0.013514870  
## Opp DvP L2 FGA L5 FGA S FGA L2 Min  
## VegasPts 0.338925228 0.06109196 0.05981149 0.05791017 0.04738682  
## FPG 0.014430012 0.78931519 0.85579585 0.90180234 0.72757530  
## FPPM 0.022358681 0.55165682 0.59236001 0.59793105 0.33574387  
## USG 0.016792709 0.72671383 0.77896637 0.80807921 0.40481709  
## FGA 0.014724566 0.87454179 0.94513087 0.99717598 0.72933058  
## MPG 0.005337456 0.72873386 0.79389623 0.85434869 0.83464535  
## L5 Min S Min L5 FP S FP Draft\_Percentage  
## VegasPts 0.04546596 0.04783081 0.1051823 0.1161640 0.1278371  
## FPG 0.78509196 0.85177463 0.9239065 0.9991058 0.4877953  
## FPPM 0.35304691 0.36695272 0.7420493 0.7713122 0.4232553  
## USG 0.43027855 0.45869242 0.6405972 0.6957977 0.3548503  
## FGA 0.78801577 0.85479115 0.8199365 0.9025424 0.4344312  
## MPG 0.90937664 0.99760758 0.7672816 0.8540734 0.3798252  
## FPDifferenceL5 FPPercentageL5 FGADifferenceL5 FGAPercentageL5  
## VegasPts -0.01068065 -0.035917316 0.014868580 -0.024543600  
## FPG -0.04191255 -0.109480124 0.007427033 -0.074157664  
## FPPM 0.04234200 -0.003271907 0.078823253 0.044978727  
## USG -0.03653162 -0.059961577 0.042554464 -0.007497329  
## FGA -0.07593495 -0.127193838 0.004735796 -0.082062072  
## MPG -0.09422694 -0.167166014 -0.042955238 -0.136690730  
## FGADifferenceL2 FGAPercentageL2 MinDifferenceL5 MinPercentageL5  
## VegasPts 0.01950156 -0.019466729 -0.00290523 -0.0008683313  
## FPG -0.02545250 -0.129072444 -0.10974054 -0.1592046017  
## FPPM 0.04062799 0.001369554 -0.01228553 -0.0237704046  
## USG 0.01666280 -0.052143764 -0.04140069 -0.0612374930  
## FGA -0.02459064 -0.137190916 -0.10979052 -0.1601518283  
## MPG -0.06280268 -0.186939943 -0.15244812 -0.2180942918  
## MinDifferenceL2 MinPercentageL2  
## VegasPts 0.005940165 -0.001155427  
## FPG -0.101774924 -0.151470556  
## FPPM -0.004049489 -0.018310514  
## USG -0.031591189 -0.052502634  
## FGA -0.103601370 -0.152564945  
## MPG -0.150431219 -0.212944426

#Lasso Regression - Ownership  
library(caret)

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

library(glmnet)

## Warning: package 'glmnet' was built under R version 3.4.4

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':  
##   
## expand

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.4

## Loaded glmnet 2.0-16

set.seed(222)  
  
ind<-sample(2,nrow(draftkingsdata),replace = TRUE, prob = c(0.7,0.3))  
train<-draftkingsdata[ind==1,]  
test<- draftkingsdata[ind==2,]  
  
custom<- trainControl(method = "repeatedcv",  
 number =10,  
 repeats =10)  
  
train<-train[, -c(11,26)]  
test<-test[,-c(11,26)]  
  
lasso<- train(Draft\_Percentage~.,  
 train,  
 method ="glmnet",  
 tuneGrid = expand.grid(alpha=1, lambda= seq(0.0001,.5,length=10)),  
 trControl=custom)  
  
  
train<-model.matrix(~., data=train)  
  
  
test<-model.matrix(~., data=test)  
  
fit.model<-glmnet(x=train[,-64], y= train[ ,64], family = "gaussian", alpha = 1, lambda=lasso$bestTune$lambda)  
fit.model$beta

## 73 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) .   
## PosPF .   
## PosPF/CPFC 1.107552818  
## PosPG 0.587680271  
## PosPG/SFPGSF 2.012324727  
## PosPG/SGPGSG 0.727655884  
## PosSF -1.289128872  
## PosSF/PFSFPF 0.478674851  
## PosSG -0.739905147  
## PosSG/SFSGSF 0.029254186  
## LocationRoad -0.572897893  
## `Def v Pos`11th 0.435054737  
## `Def v Pos`12th 1.059638125  
## `Def v Pos`13th .   
## `Def v Pos`14th .   
## `Def v Pos`15th 0.043711289  
## `Def v Pos`16th -0.385158651  
## `Def v Pos`17th 0.798876794  
## `Def v Pos`18th -0.222229998  
## `Def v Pos`19th 0.926395512  
## `Def v Pos`1st 0.124500794  
## `Def v Pos`20th -0.931619784  
## `Def v Pos`21st -0.489879180  
## `Def v Pos`22nd -0.426905544  
## `Def v Pos`23rd -0.110671774  
## `Def v Pos`24th 0.443138410  
## `Def v Pos`25th 2.132628701  
## `Def v Pos`26th -0.275581235  
## `Def v Pos`27th 0.470013665  
## `Def v Pos`28th .   
## `Def v Pos`29th .   
## `Def v Pos`2nd -0.396257895  
## `Def v Pos`30th -1.119170707  
## `Def v Pos`3rd 1.004762679  
## `Def v Pos`4th 0.359428336  
## `Def v Pos`5th -0.801548663  
## `Def v Pos`6th -0.362326215  
## `Def v Pos`7th .   
## `Def v Pos`8th .   
## `Def v Pos`9th 0.613757890  
## VegasPts .   
## FPG .   
## FPPM 4.085587521  
## USG .   
## FGA .   
## MPG .   
## Con. -0.056635912  
## Rest1 -0.798509012  
## Rest2 -0.300946091  
## Rest3 1.113699195  
## Rest4 7.358953807  
## PER -0.105024095  
## `Opp Pace` 0.295958675  
## `Opp DEff` -0.012391223  
## `Opp DvP` 0.057693453  
## `L2 FGA` 0.103128822  
## `L5 FGA` .   
## `S FGA` .   
## `L2 Min` 0.051442014  
## `L5 Min` .   
## `S Min` .   
## `L5 FP` 0.351825160  
## `S FP` .   
## FPDifferenceL5 .   
## FPPercentageL5 -0.016902441  
## FGADifferenceL5 -0.558467717  
## FGAPercentageL5 -0.002045577  
## FGADifferenceL2 1.199055528  
## FGAPercentageL2 -0.011144875  
## MinDifferenceL5 .   
## MinPercentageL5 -0.016161272  
## MinDifferenceL2 0.661921713  
## MinPercentageL2 .

predictions<-predict(fit.model, test[,-64])  
error<- predictions-test[,64]  
RMSE<-sqrt(mean(error^2))  
  
  
  
RMSE

## [1] 7.138164

#Lasso Logistic- GPP Viable  
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.4.4

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

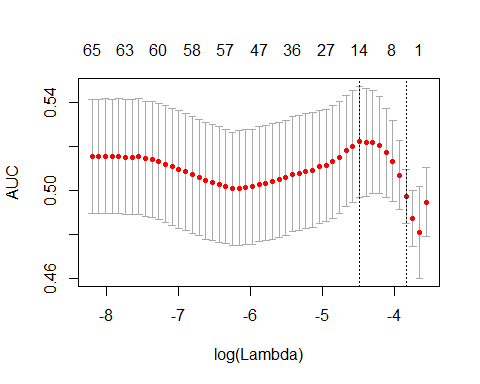
##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

## The following object is masked from 'package:psych':  
##   
## outlier

## The following object is masked from 'package:dplyr':  
##   
## combine

set.seed(224)  
  
draftkingsdatalogviable<-draftkingsdata[,-c(11,25)]  
  
draftkingsdatalogviable$Pos<-as.factor(draftkingsdatalogviable$Pos)  
ind1<-sample(2,nrow(draftkingsdatalogviable),replace = TRUE, prob = c(0.7,0.3))  
train1<-draftkingsdatalogviable[ind1==1,]  
test1<- draftkingsdatalogviable[ind1==2,]  
  
  
  
custom<- trainControl(method = "repeatedcv",  
 number =10,  
 repeats =10)  
  
train3<-model.matrix(~., data=train1)  
  
  
test3<-model.matrix(~., data=test1)  
  
CV=cv.glmnet(x=train3[,-64],y=train3[,64], family="binomial", type.measure = "auc", alpha=1, nlambda=100)  
  
plot(CV)



fit.model1<-glmnet(x=train3[,-64], y= train3[ ,64], family = "gaussian", alpha = 1, lambda=CV$lambda.1se)  
fit.model1$beta

## 73 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) .   
## PosPF .   
## PosPF/CPFC 1.078324e-02  
## PosPG .   
## PosPG/SFPGSF .   
## PosPG/SGPGSG .   
## PosSF .   
## PosSF/PFSFPF .   
## PosSG .   
## PosSG/SFSGSF .   
## LocationRoad .   
## `Def v Pos`11th .   
## `Def v Pos`12th .   
## `Def v Pos`13th .   
## `Def v Pos`14th .   
## `Def v Pos`15th .   
## `Def v Pos`16th .   
## `Def v Pos`17th .   
## `Def v Pos`18th .   
## `Def v Pos`19th .   
## `Def v Pos`1st .   
## `Def v Pos`20th .   
## `Def v Pos`21st .   
## `Def v Pos`22nd .   
## `Def v Pos`23rd .   
## `Def v Pos`24th .   
## `Def v Pos`25th .   
## `Def v Pos`26th .   
## `Def v Pos`27th 3.742240e-02  
## `Def v Pos`28th .   
## `Def v Pos`29th .   
## `Def v Pos`2nd .   
## `Def v Pos`30th .   
## `Def v Pos`3rd .   
## `Def v Pos`4th .   
## `Def v Pos`5th -9.973553e-03  
## `Def v Pos`6th .   
## `Def v Pos`7th -1.438517e-03  
## `Def v Pos`8th .   
## `Def v Pos`9th .   
## VegasPts .   
## FPG .   
## FPPM .   
## USG .   
## FGA .   
## MPG .   
## Con. .   
## Rest1 .   
## Rest2 .   
## Rest3 .   
## Rest4 .   
## PER .   
## `Opp Pace` .   
## `Opp DEff` .   
## `Opp DvP` .   
## `L2 FGA` .   
## `L5 FGA` .   
## `S FGA` .   
## `L2 Min` 7.557632e-05  
## `L5 Min` .   
## `S Min` .   
## `L5 FP` .   
## `S FP` .   
## FPDifferenceL5 .   
## FPPercentageL5 .   
## FGADifferenceL5 .   
## FGAPercentageL5 .   
## FGADifferenceL2 .   
## FGAPercentageL2 .   
## MinDifferenceL5 .   
## MinPercentageL5 .   
## MinDifferenceL2 .   
## MinPercentageL2 .

predictions1<-predict(fit.model1, test3[,-64], type="response")  
  
fitted.results <- ifelse(predictions1 > 0.23,1,0)  
  
confusion.matrix<-table(test3[,64], fitted.results)  
TruePositiveRate<-confusion.matrix[2,2]/(confusion.matrix[1,2] + confusion.matrix[2,2])  
Recall<-confusion.matrix[2,2]/(confusion.matrix[2,1] + confusion.matrix[2,2])  
  
confusion.matrix

## fitted.results  
## 0 1  
## 0 13 477  
## 1 3 131

TruePositiveRate

## [1] 0.2154605

Recall

## [1] 0.9776119

#Random Forest  
#evenoutclasses  
  
draftkingsdatalogviable.df<-randomForest(data=draftkingsforestdata, subset = train1, y=train1[,24], x= train1[,-24], xtest = test1[,-24], ytest = test1[,24], ntree = 500, sampsize = c(300,300))  
  
draftkingsdatalogviable.df$confusion

## No Yes class.error  
## No 687 435 0.3877005  
## Yes 215 141 0.6039326

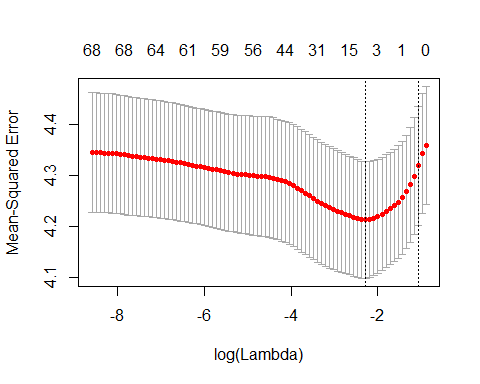
TPP2<- draftkingsdatalogviable.df$confusion[2,2]/(draftkingsdatalogviable.df$confusion[2,2]+draftkingsdatalogviable.df$confusion[1,2])  
  
Recall2<-draftkingsdatalogviable.df$confusion[2,2]/(draftkingsdatalogviable.df$confusion[2,2]+draftkingsdatalogviable.df$confusion[2,1])  
  
TPP2

## [1] 0.2447917

Recall2

## [1] 0.3960674

#Lasso Linear- Value  
  
set.seed(228)  
  
ind3<-sample(2,nrow(draftkingsdata),replace = TRUE, prob = c(0.7,0.3))  
train2<-draftkingsdata[ind3==1,]  
test2<- draftkingsdata[ind3==2,]  
  
train2<-train2[, -c(26,25)]  
test2<-test2[,-c(26,25)]  
  
custom<- trainControl(method = "repeatedcv",  
 number =10,  
 repeats =10)  
  
  
  
train2<-model.matrix(~., data=train2)  
  
  
test2<-model.matrix(~., data=test2)  
  
CV=cv.glmnet(x=train2[,-48],y=train2[,48], family="gaussian", alpha=1, nlambda=100)  
  
plot(CV)



fit.model2<-glmnet(x=train2[,-48], y=(train2[ ,48]), family = "gaussian", alpha = 1, lambda=CV$lambda.1se)  
  
coef(fit.model2)

## 74 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 4.305603081  
## (Intercept) .   
## PosPF .   
## PosPF/CPFC .   
## PosPG .   
## PosPG/SFPGSF .   
## PosPG/SGPGSG .   
## PosSF .   
## PosSF/PFSFPF .   
## PosSG .   
## PosSG/SFSGSF .   
## LocationRoad .   
## `Def v Pos`11th .   
## `Def v Pos`12th .   
## `Def v Pos`13th .   
## `Def v Pos`14th .   
## `Def v Pos`15th .   
## `Def v Pos`16th .   
## `Def v Pos`17th .   
## `Def v Pos`18th .   
## `Def v Pos`19th .   
## `Def v Pos`1st .   
## `Def v Pos`20th .   
## `Def v Pos`21st .   
## `Def v Pos`22nd .   
## `Def v Pos`23rd .   
## `Def v Pos`24th .   
## `Def v Pos`25th .   
## `Def v Pos`26th .   
## `Def v Pos`27th .   
## `Def v Pos`28th .   
## `Def v Pos`29th .   
## `Def v Pos`2nd .   
## `Def v Pos`30th .   
## `Def v Pos`3rd .   
## `Def v Pos`4th .   
## `Def v Pos`5th .   
## `Def v Pos`6th .   
## `Def v Pos`7th .   
## `Def v Pos`8th .   
## `Def v Pos`9th .   
## VegasPts .   
## FPG .   
## FPPM .   
## USG .   
## FGA .   
## MPG .   
## Con. .   
## Rest1 .   
## Rest2 .   
## Rest3 .   
## Rest4 .   
## PER .   
## `Opp Pace` .   
## `Opp DEff` .   
## `Opp DvP` .   
## `L2 FGA` .   
## `L5 FGA` .   
## `S FGA` .   
## `L2 Min` 0.008649938  
## `L5 Min` .   
## `S Min` .   
## `L5 FP` .   
## `S FP` .   
## FPDifferenceL5 .   
## FPPercentageL5 .   
## FGADifferenceL5 .   
## FGAPercentageL5 .   
## FGADifferenceL2 .   
## FGAPercentageL2 .   
## MinDifferenceL5 .   
## MinPercentageL5 .   
## MinDifferenceL2 .   
## MinPercentageL2 .

fit.model2.predict<-predict(fit.model2, test2[,-48], type="link")  
error33<- fit.model2.predict-test2[,48]  
RMSE33<-sqrt(mean(error33^2))  
RMSE33

## [1] 2.061721

#Multivariate Regression - Ownership  
#install.packages("mlbench", repos="http://cran.us.r-project.org")  
#library(mlbench)  
install.packages("psych", repos="http://cran.us.r-project.org")

## Installing package into 'C:/Users/John/Documents/R/win-library/3.4'  
## (as 'lib' is unspecified)

## Warning: package 'psych' is in use and will not be installed

library(psych)  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(caret)  
library(randomForest)  
  
  
set.seed(232)  
  
  
ind4<-sample(2,nrow(draftkingsdata),replace = TRUE, prob = c(0.7,0.3))  
train4<-draftkingsdata[ind==1,]  
test4<- draftkingsdata[ind==2,]  
  
  
  
train4<-train4[, -c(11,26)]  
  
  
test4<-test4[,-c(11,26)]  
  
nums <- unlist(lapply(train4, is.numeric))  
numerictraindata<-train4[,nums]  
numerictraindata<-numerictraindata[,-22]  
  
  
numerictestdata<-test4[,nums]  
numerictestdata<-numerictestdata[,-22]  
  
  
  
correlatedcols<-findCorrelation(cor(numerictraindata), cutoff = 0.85)  
trainlinear<-train4[,-correlatedcols]  
testlinear<-test4[, -correlatedcols]  
  
nums <- unlist(lapply(draftkingsdata, is.numeric))  
draftkingsdatalinear<-draftkingsdata[,nums]  
draftkingsdatalinear<-draftkingsdatalinear[,-22]  
correlatedcols<-findCorrelation(cor(draftkingsdatalinear), cutoff = 0.85)  
draftkingsdatalinear<- draftkingsdatalinear[, -correlatedcols]  
  
  
fullmodel<- lm(Draft\_Percentage~ ., data=trainlinear)  
stepmodel<- stepAIC( fullmodel, direction = "both", trace=0)  
valueprediction<-predict(stepmodel, testlinear)  
AIC(stepmodel)

## [1] 9732.895

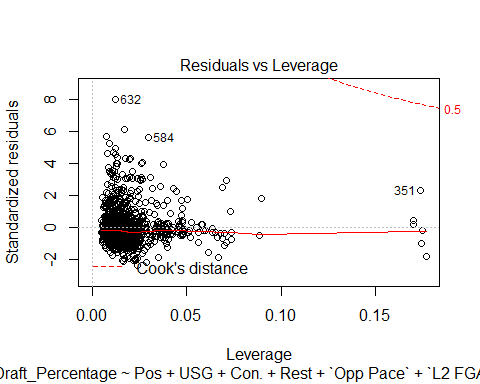
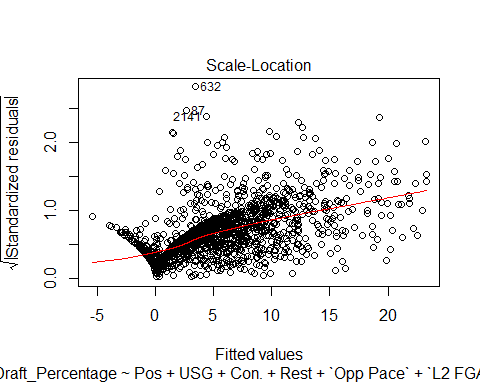
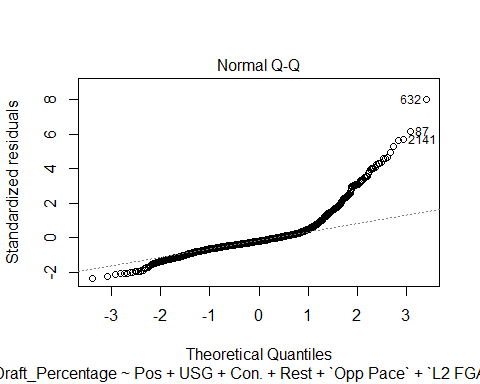
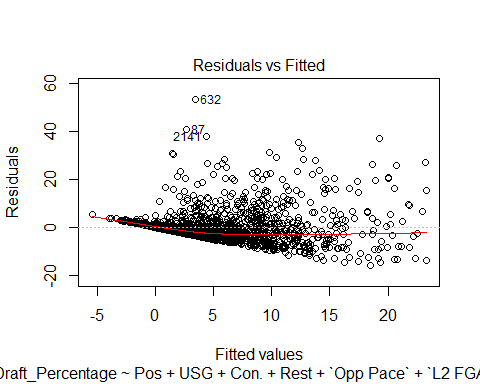
error1<- valueprediction-testlinear[,14]  
RMSE1<-sqrt(mean(error1^2))  
RMSE1

## [1] 7.281681

summary(stepmodel)

##   
## Call:  
## lm(formula = Draft\_Percentage ~ Pos + USG + Con. + Rest + `Opp Pace` +   
## `L2 FGA` + `S Min` + `L5 FP` + FPDifferenceL5 + FGADifferenceL5 +   
## MinPercentageL5 + MinDifferenceL2 + MinPercentageL2, data = trainlinear)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15.582 -3.260 -1.312 1.178 53.232   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -34.82691 7.28374 -4.781 1.92e-06 \*\*\*  
## PosPF 0.62172 1.33910 0.464 0.642514   
## PosPF/CPFC 2.03637 0.71157 2.862 0.004274 \*\*   
## PosPG 1.63316 0.82783 1.973 0.048707 \*   
## PosPG/SFPGSF 2.99654 1.75465 1.708 0.087895 .   
## PosPG/SGPGSG 1.61295 0.78578 2.053 0.040287 \*   
## PosSF -1.51246 1.55246 -0.974 0.330104   
## PosSF/PFSFPF 1.33317 0.75742 1.760 0.078595 .   
## PosSG -0.48470 1.01381 -0.478 0.632652   
## PosSG/SFSGSF 0.78133 0.78344 0.997 0.318783   
## USG -0.14793 0.06615 -2.236 0.025485 \*   
## Con. -0.06406 0.02118 -3.025 0.002531 \*\*   
## Rest1 -0.88429 0.44609 -1.982 0.047638 \*   
## Rest2 -0.40614 0.58924 -0.689 0.490763   
## Rest3 1.27242 0.91635 1.389 0.165177   
## Rest4 8.52853 2.77025 3.079 0.002119 \*\*   
## `Opp Pace` 0.35978 0.07140 5.039 5.28e-07 \*\*\*  
## `L2 FGA` 0.43352 0.09682 4.478 8.14e-06 \*\*\*  
## `S Min` -0.13571 0.07650 -1.774 0.076295 .   
## `L5 FP` 0.43752 0.05171 8.461 < 2e-16 \*\*\*  
## FPDifferenceL5 -0.56254 0.37162 -1.514 0.130307   
## FGADifferenceL5 -0.65986 0.29128 -2.265 0.023637 \*   
## MinPercentageL5 -0.02821 0.01775 -1.589 0.112287   
## MinDifferenceL2 2.17762 0.56190 3.875 0.000111 \*\*\*  
## MinPercentageL2 -0.05058 0.02369 -2.135 0.032930 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.705 on 1436 degrees of freedom  
## Multiple R-squared: 0.3477, Adjusted R-squared: 0.3368   
## F-statistic: 31.9 on 24 and 1436 DF, p-value: < 2.2e-16

plot(stepmodel)



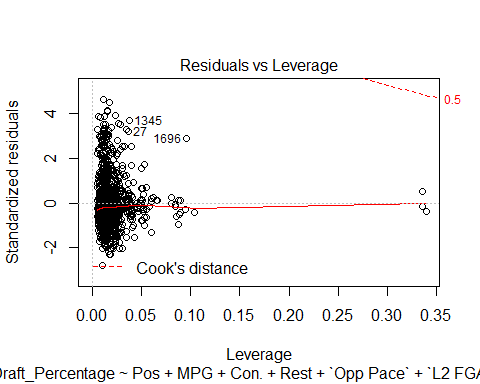
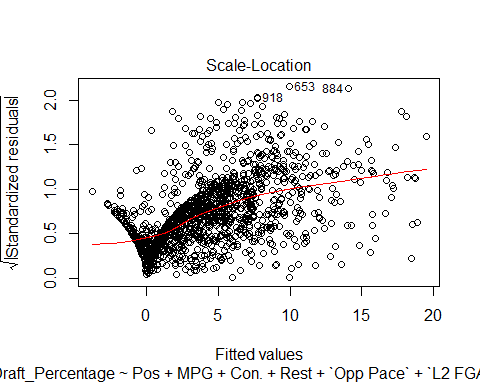
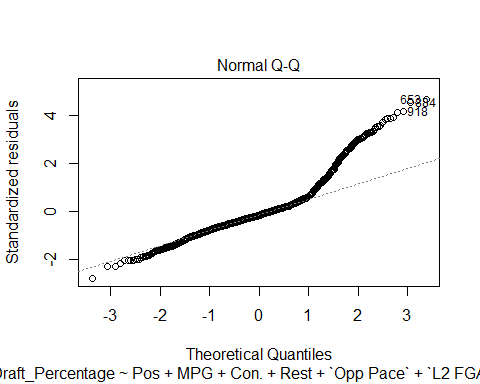
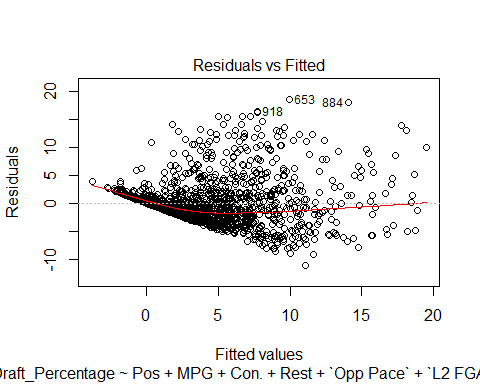
d1<-cooks.distance(fullmodel)  
r <- stdres(fullmodel)  
a<-cbind(trainlinear, d1, r)  
ppp<-a[d1<4/nrow(trainlinear),]  
ppp<-ppp[,-c(22:23)]  
  
  
fullmodelppp<- lm(Draft\_Percentage~ ., data=ppp)  
stepmodelppp<- stepAIC( fullmodelppp, direction = "both", trace=0)  
valuepredictionppp<-predict(stepmodelppp, testlinear)  
AIC(stepmodelppp)

## [1] 7725.076

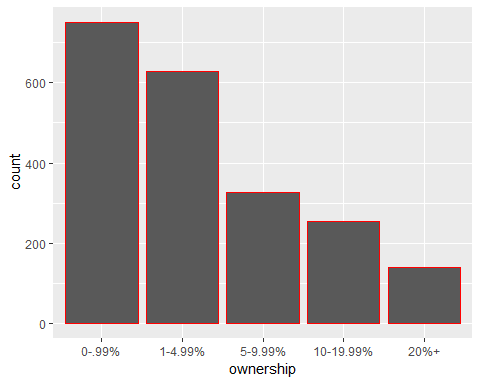
errorppp<- valuepredictionppp-testlinear[,14]  
RMSEppp<-sqrt(mean(errorppp^2))  
summary(stepmodelppp)

##   
## Call:  
## lm(formula = Draft\_Percentage ~ Pos + MPG + Con. + Rest + `Opp Pace` +   
## `L2 FGA` + `L5 Min` + `L5 FP` + FPDifferenceL5 + FGADifferenceL5 +   
## MinDifferenceL2 + MinPercentageL2, data = ppp)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.0810 -2.2540 -0.6541 1.1842 18.4783   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -31.90735 4.43263 -7.198 1.01e-12 \*\*\*  
## PosPF -0.21009 0.82390 -0.255 0.7988   
## PosPF/CPFC 0.89445 0.43942 2.036 0.0420 \*   
## PosPG -0.12059 0.50714 -0.238 0.8121   
## PosPG/SFPGSF 0.46148 1.19295 0.387 0.6989   
## PosPG/SGPGSG 0.57206 0.47438 1.206 0.2281   
## PosSF -1.34060 0.96483 -1.389 0.1649   
## PosSF/PFSFPF 0.49131 0.46784 1.050 0.2938   
## PosSG -1.27502 0.61664 -2.068 0.0389 \*   
## PosSG/SFSGSF 0.07841 0.47365 0.166 0.8685   
## MPG 0.15810 0.07148 2.212 0.0272 \*   
## Con. -0.06528 0.01276 -5.118 3.53e-07 \*\*\*  
## Rest1 0.11728 0.27626 0.425 0.6712   
## Rest2 0.23298 0.36456 0.639 0.5229   
## Rest3 0.47476 0.59429 0.799 0.4245   
## Rest4 10.94466 2.33382 4.690 3.02e-06 \*\*\*  
## `Opp Pace` 0.29910 0.04391 6.812 1.44e-11 \*\*\*  
## `L2 FGA` 0.22131 0.04966 4.457 9.01e-06 \*\*\*  
## `L5 Min` -0.12733 0.05903 -2.157 0.0312 \*   
## `L5 FP` 0.32918 0.02916 11.287 < 2e-16 \*\*\*  
## FPDifferenceL5 -0.40125 0.22080 -1.817 0.0694 .   
## FGADifferenceL5 -0.34553 0.17686 -1.954 0.0510 .   
## MinDifferenceL2 2.05750 0.34476 5.968 3.07e-09 \*\*\*  
## MinPercentageL2 -0.05486 0.01286 -4.265 2.14e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.003 on 1348 degrees of freedom  
## Multiple R-squared: 0.4952, Adjusted R-squared: 0.4865   
## F-statistic: 57.48 on 23 and 1348 DF, p-value: < 2.2e-16

plot(stepmodelppp)



#rlm  
  
  
#Random Forest  
  
ind<-sample(2,nrow(draftkingsdata),replace = TRUE, prob = c(0.7,0.3))  
train5<-draftkingsdatalinear[ind==1,]  
test5<- draftkingsdatalinear[ind==2,]  
  
  
  
train5<-train5[, -c(11,26)]  
  
  
test5<-test5[,-c(11,26)]  
  
  
  
draftkingsdatalinear$ownership<-ifelse(draftkingsdatalinear$Draft\_Percentage<1, draftkingsdatalinear$ownership<-"0-.99%",  
 ifelse(draftkingsdatalinear$Draft\_Percentage<5, draftkingsdatalinear$ownership<-"1-4.99%",  
 ifelse(draftkingsdatalinear$Draft\_Percentage<10,draftkingsdatalinear$ownership<-"5-9.99%",  
 ifelse(draftkingsdatalinear$Draft\_Percentage<20, draftkingsdatalinear$ownership<-"10-19.99%",  
 ifelse(draftkingsdatalinear$Draft\_Percentage<100, draftkingsdatalinear$ownership<-"20%+",0  
 ) ) ) ))  
  
draftkingsdatalinear$ownership<-factor(draftkingsdatalinear$ownership, levels=(c("0-.99%", "1-4.99%", "5-9.99%", "10-19.99%", "20%+")))  
  
ggplot(data = draftkingsdatalinear, aes(x=ownership))+  
 geom\_bar(aes(x=ownership), colour= "red")



set.seed(5050)  
  
  
  
draftkingsforestdata<-draftkingsdatalinear[,-c(11,25,26)]  
  
  
ind21<-sample(2,nrow(draftkingsforestdata),replace = TRUE, prob = c(0.7,0.3))  
train21<-draftkingsforestdata[ind21==1,]  
test21<- draftkingsforestdata[ind21==2,]  
  
  
  
draftkingsdatalinear.df<-randomForest(data=draftkingsforestdata, subset = train21, y=train21[,17], x= train21[,-17], xtest = test21[,-17], ytest = test21[,17], ntree = 500, sampsize = c(90,90,90,90,90), importance = TRUE)  
  
draftkingsdatalinear.df

##   
## Call:  
## randomForest(x = train21[, -17], y = train21[, 17], xtest = test21[, -17], ytest = test21[, 17], ntree = 500, sampsize = c(90, 90, 90, 90, 90), importance = TRUE, data = draftkingsforestdata, subset = train21)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 51.23%  
## Confusion matrix:  
## 0-.99% 1-4.99% 5-9.99% 10-19.99% 20%+ class.error  
## 0-.99% 399 76 9 11 8 0.2067594  
## 1-4.99% 106 150 90 53 27 0.6478873  
## 5-9.99% 14 67 67 49 24 0.6968326  
## 10-19.99% 12 44 46 50 29 0.7237569  
## 20%+ 13 18 13 21 29 0.6914894  
## Test set error rate: 48.15%  
## Confusion matrix:  
## 0-.99% 1-4.99% 5-9.99% 10-19.99% 20%+ class.error  
## 0-.99% 193 41 10 2 2 0.2217742  
## 1-4.99% 46 92 37 19 9 0.5467980  
## 5-9.99% 12 27 31 29 7 0.7075472  
## 10-19.99% 5 17 22 18 12 0.7567568  
## 20%+ 6 2 8 13 17 0.6304348

varImp(draftkingsdatalinear.df)

## 0-.99% 1-4.99% 5-9.99% 10-19.99% 20%+  
## VegasPts 6.407259 2.4677539 -0.65250927 -1.48899145 7.69494049  
## USG 14.624499 -3.8211463 -1.80911168 3.28694915 12.84571426  
## Con. 17.590687 -3.1503034 7.20647626 6.65462261 5.04441773  
## Val 7.644503 -0.1493441 2.63469156 -1.35614052 6.11826965  
## PER 22.710819 -2.2595591 4.80127508 -0.90707200 12.84132137  
## Opp Pace 7.208625 2.3387745 0.46553167 1.99521907 0.09853782  
## Opp DEff 1.285675 1.0374116 4.15675485 -2.34169318 0.75880094  
## Opp DvP 2.494847 0.5641032 0.77471889 2.52534878 0.91901774  
## L5 FGA 24.965527 -6.6714305 4.00748522 11.20090958 11.50307406  
## L5 Min 36.248083 -6.4015392 15.99312014 17.90916821 10.84959667  
## FPPercentageL5 4.748851 -0.2954263 0.02939025 0.02311968 5.86281383  
## FGADifferenceL5 6.110488 2.7508791 1.64638270 0.97374909 -2.27768040  
## FGADifferenceL2 9.212008 1.6379393 0.44960835 1.59694222 5.61787438  
## FGAPercentageL2 4.139836 -0.4207568 2.66736118 2.28799316 7.15863985  
## MinDifferenceL5 5.810691 3.8861844 2.03999928 -1.20473946 2.81755580  
## MinDifferenceL2 8.231427 2.5449358 0.65490982 5.93736917 6.20980781

#Naive Bayes Classification  
library(e1071)

## Warning: package 'e1071' was built under R version 3.4.4

set.seed(1222)  
  
  
ind8<-sample(2,nrow(draftkingsdata),replace = TRUE, prob = c(0.7,0.3))  
train8<-draftkingsdata[ind8==1,]  
test8<- draftkingsdata[ind8==2,]  
  
train8<-train8[, -c(11,25)]  
  
  
test8<-test8[,-c(11,25)]  
  
Naive\_Bayes\_Model<-naiveBayes(GPPViable ~., data=train8)  
NB\_Predictions=predict(Naive\_Bayes\_Model,newdata=test8)  
  
cmatrix<-table(NB\_Predictions,test8$GPPViable)  
  
TPP= cmatrix[2,2]/(cmatrix[1,2]+cmatrix[2,2])  
  
Recall1<- cmatrix[2,2]/(cmatrix[2,1]+cmatrix[2,2])  
  
cmatrix

##   
## NB\_Predictions No Yes  
## No 369 108  
## Yes 108 42

TPP

## [1] 0.28

Recall1

## [1] 0.28