Project

Neeraj Asthana

May 10, 2016

Extra Credit Project

Stat 428: Statistical Computing

Neeraj Asthana (nasthan2)

Introduction

Instead of extending ideas we learned in class, for my project I decided to do something more practical and analyze a real dataset. For my Extra Credit project for STAT 428, I chose to analyze a 2015 Fantasy Football dataset and apply methods we have learned in class to that dataset.

Every summer ESPN.com releases a projections list that forecasts how each NFL player will perform and how many "fantasy" points they will score in the upcoming season. An example list of these projections can be seen at: http://games.espn.go.com/ffl/tools/projections

The goal of my project is to analyze these projections and understand how accurate they compared to how players actually end up performing. I will look at the 2015 Fantasy Football season and compare each player's projected fantasy points to how many points they ended up scoring in the 2015 season. On top of that I will attempt to build and select a regression model using the projection listings as predictors for the actual amount of fantasy football points. I will use the jackknife cross validation method to select features and to compare the models (using the mean squared errors). I will only be looking at the Running Back (RB), Wide Receiver (WR), and TightEnd (TE) positions individually as these are the most relevant positions in fantasy football. Additionally, each of the positions is extremely different and requires a different model.

Setup

All of the data used in this project comes from espn.com and I have included these datasets with the report. They are labelled "2015proj.csv" and "2015data.csv".

The "2015proj.csv" file contains data for how well ESPN believes a specific NFL player will perform in the 2015 season. Each row represents a single player and includes data on the estimated numbers of certain statistics such as touchdowns, receptions, yards, fantasy points, etc. that ESPN forecasts they will score. "2015proj.csv" has 405 rows and 35 columns.

The "2015data.csv" file contains data on how well NFL players actually performed in the 2015 season. Each row represents a single player and includes data on the numbers of certain statistics such as touchdowns, receptions, yards, fantasy points, etc they actually got in the 2015 NFL season. "2015data.csv" has 400 rows and 20 columns. I will only end up using the "PTS" column from this dataset which is the actual number of fantasy football points scored by every player.

I will begin by first reading in the files and then I will manipulate the data so that it can easily be inputted into a linear regression model (easily put into the lm function).

Reading files and partially display data

I will begin by reading in the “2015proj.csv” and “2015data.csv” files using R’s “read.csv” function. I also display the first few rows of each of these files using the head function so that the reader has a better idea of what the data looks like.

```
options(warn=-1)
#setwd("/home/neeraj/Documents/Projects/FantasyFootball/data/formatted_data")

#data for projected number of points in 2015
proj <- read.csv("2015proj.csv")
head(proj)
```

```
##              Name Position Draft  RUSH RUSHYDS RUSHAVG RUSHTD   TAR   REC
## 1    Le'Veon Bell       RB     1 254.0  1090.0     4.3     --    --  62.0
## 2 Adrian Peterson       RB     2 264.0  1231.0     4.7     --    --  46.0
## 3      Eddie Lacy       RB     3 301.0  1332.0     4.4     --    --  45.0
## 4  Jamaal Charles       RB     4 243.0  1118.0     4.6     --    --  55.0
## 5  Marshawn Lynch       RB     5 282.0  1293.0     4.6     --    --  29.0
## 6   Antonio Brown       WR     6   5.0    30.0      --    0.0 170.0 115.0
##   RECYDS RECAVG RECTD  C  A COMPYDS COMPTD INT SCK DINT FR DTD PA YA
## 1  502.0     --   3.0 -- --      --     --  --  --   -- --  -- -- --
## 2  391.0     --   2.0 -- --      --     --  --  --   -- --  -- -- --
## 3  317.0     --   2.0 -- --      --     --  --  --   -- --  -- -- --
## 4  473.0     --   3.0 -- --      --     --  --  --   -- --  -- -- --
## 5  233.0     --   2.0 -- --      --     --  --  --   -- --  -- -- --
## 6 1513.0   13.2    -- -- --      --     --  --  --   -- --  -- -- --
##   X1.39A X1.39C X40.49A X40.49C X50.A X50.C TOTA TOTC XPA XPC   PTS TEAM
## 1     --     --      --      --    --    --   --   --  --  -- 211.5   --
## 2     --     --      --      --    --    --   --   --  --  -- 222.6   --
## 3     --     --      --      --    --    --   --   --  --  -- 232.7   --
## 4     --     --      --      --    --    --   --   --  --  -- 218.7   --
## 5     --     --      --      --    --    --   --   --  --  -- 208.5   --
## 6     --     --      --      --    --    --   --   --  --  -- 209.1   --
```

```
#data for actual 2015 statistics (what players actually scored)
actual <- read.csv("2015stats.csv")
head(actual)
```

```
##             Name Position Rank RUSH RUSHYDS RUSHTD TAR REC RECYDS RECTD
## 1     Cam Newton       QB    1  132     636     10   0   0      0     0
## 2      Tom Brady       QB    2   34      53      3   1   1     36     0
## 3 Russell Wilson       QB    3  103     553      1   0   0      0     0
## 4  Blake Bortles       QB    4   52     310      2   0   0      0     0
## 5  Carson Palmer       QB    5   25      24      1   0   0      0     0
## 6     Drew Brees       QB    6   24      14      1   0   0      0     0
##     C   A COMPYDS COMPTD INT TWOPC FUM MISCTD PTS TEAM
## 1 296 496    3837     35  10     0   4      0 373  Car
## 2 402 624    4770     36   7     0   2      0 335   NE
## 3 329 483    4024     34   8     0   3      0 322  Sea
## 4 355 606    4428     35  18     1   5      0 302  Jax
## 5 342 538    4671     35  11     0   2      0 300  Ari
## 6 428 627    4870     32  11     0   2      0 299   NO
```

Cleaning the data

The data must be cleaned before it can be modelled or analyzed.

I begin the cleaning process by first merging the the proj and actual datasets to have all of the predictors and the response variable in a single dataset (called *first*). I will merge on the name of the player as this is field is the same for both datasets.

```
#Many of the columns in the proj dataset were read as factors instead of actual numeric values so these
for(i in c(4:17,34)){
  proj[,c(i)] <- as.numeric(as.character(proj[,c(i)]))
}

first <- merge(proj, actual, by.x = "Name", by.y = "Name", suffixes = c("proj","actual"))
dim(first)
```

```
## [1] 335  54
```

The first data structure has 335 rows and 54 columns. Many of the columns are not necessary as they contain statstics for Kicking, Passing, and Defense which irrelevant to the positions I am analyzing. Therefore, many of the columns will dropped as they are not being used as predictors.

There are fewer rows than expected in the first dataset because some players are not in both datasets. Most of the players that are not included are extremely irrelevant and would not be on fantasy footall teams anyways so it is okay that they are not included. However there are a few notable players on the missing players list including “Marlon brown”, “Pierre thomas”, “Charcandrick West”, and “Rishard Matthews” who had very good seasons but only because other players had injuries. However, almost all of the important players are included in the first dataset so I will proceed with the next steps. There a total of 135 missing players which are listed below.

```
#Many of the columns in the proj dataset were read as factors instead of actual numeric values so these
#find which players are missing from the dataset
allnames <- union(proj[,"Name"], actual[,"Name"])
missing <- allnames[!allnames %in% first[,"Name"]]
missing
```

```
##   [1] "Charles Johnson"       "Knile Davis"
##   [3] "Cody Latimer"          "Lorenzo Taliaferro"
##   [5] "Terrance West"         "Roy Helu"
##   [7] "Victor Cruz"           "Josh Robinson"
##   [9] "Dwayne Allen"          "Cody Parkey"
##  [11] "Brian Quick"           "Reggie Bush"
##  [13] "Breshad Perriman"      "Branden Oliver"
##  [15] "Jeff Janis"            "Juwan Thompson"
##  [17] "Boobie Dixon"          "Toby Gerhart"
##  [19] "Dwayne Bowe"           "Cordarrelle Patterson"
##  [21] "Matt Asiata"           "Bryce Brown"
##  [23] "Zac Stacy"             "Jacquizz Rodgers"
##  [25] "Bobby Rainey"          "Montee Ball"
##  [27] "Bernard Pierce"        "Mike Davis"
##  [29] "Zach Zenner"           "Daniel Herron"
##  [31] "Stevan Ridley"         "Pierre Thomas"
##  [33] "Sammie Coates"         "Aaron Dobson"
##  [35] "Kevin White"           "Devin Smith"
##  [37] "Jordan Todman"         "Marlon Brown"
##  [39] "Stepfan Taylor"        "Dri Archer"
##  [41] "Will Johnson"          "Jarryd Hayne"
##  [43] "Brandon Wegher"        "Alonzo Harris"
##  [45] "Garrett Graham"        "Justin Hardy"
##  [47] "Rod Streater"          "Terron Ward"
##  [49] "Corey Grant"           "Vick Ballard"
##  [51] "Preston Parker"        "Antone Smith"
##  [53] "Andrew Quarless"       "Trent Richardson"
##  [55] "Andre Roberts"         "Robert Griffin"
##  [57] "Gavin Escobar"         "Rob Housler"
##  [59] "Dion Sims"             "Paul Richardson"
##  [61] "Jeff Cumberland"       "Timothy Wright"
##  [63] "C.J. Fiedorowicz"      "James Casey"
##  [65] "Jimmy Garoppolo"       "Troy Niklas"
##  [67] "Brandon Pettigrew"     "Tyler Kroft"
##  [69] "Reggie Wayne"          "Mike Glennon"
##  [71] "Chris Boswell"         "Blaine Gabbert"
##  [73] "Dustin Hopkins"        "Willie Snead"
##  [75] "Charcandrick West"     "Connor Barth"
##  [77] "Redskins D/ST"         "Caleb Sturgis"
##  [79] "Raiders D/ST"          "Nick Novak"
##  [81] "Matt Hasselbeck"       "Rishard Matthews"
##  [83] "Antonio Andrews"       "Seth Roberts"
##  [85] "Spencer Ware"          "Tim Hightower"
##  [87] "Dwayne Harris"         "Zach Miller"
##  [89] "Chargers D/ST"         "Matt Cassel"
##  [91] "Chris Hogan"           "Will Tye"
##  [93] "AJ McCarron"           "Lance Moore"
##  [95] "Kyle Juszczyk"         "Shayne Graham"
##  [97] "Mike Gillislee"        "Case Keenum"
##  [99] "Darrius Heyward-Bey"   "J.J. Nelson"
## [101] "Zach Mettenberger"     "Chris Givens"
## [103] "Bryan Walters"         "Rashad Ross"
## [105] "Garrett Celek"         "Jeremy Butler"
## [107] "Kellen Moore"          "Keshawn Martin"
## [109] "Cameron Brate"         "Joshua Bellamy"
## [111] "Adam Humphries"        "Mike Vick"
## [113] "Matt Schaub"           "Ed Dickson"
## [115] "Quincy Enunwa"         "Marc Mariani"
## [117] "DuJuan Harris"         "Robert Turbin"
## [119] "Brice Butler"          "Nick Williams"
## [121] "Zach Line"             "Michael Hoomanawanui"
## [123] "T.J. Yates"            "Craig Stevens"
## [125] "Clay Harbor"           "Landry Jones"
## [127] "Orleans Darkwa"        "Griff Whalen"
## [129] "Jamize Olawale"        "Adam Thielen"
## [131] "Patrick DiMarco"       "John Kuhn"
## [133] "Bruce Miller"          "Austin Davis"
## [135] "Kerwynn Williams"
```

Performance and Modelling

Each position is extremely different so I will subset different positions: RB, WR, TE

Removal of unnessary columns -> Defensive, and Kicking and repeated columns (Teams, Positions)

Jackknife Cross Validation Function

"crossvalidation" is a helper function that takes as input a formula for a lm function and a dataset and returns the mean squared error for fitting that model using jackknife crossvalidation. The function will fit the model on all but one point and use the last point as a test point and compare the expected result to the actual result to get an error value. The error values are then squared and summed across all points. This sum along with the average AIC for each model is returned.

```r
crossvalidation <- function(fit, dataset, labelcol){
  n <- dim(dataset)[1]
  totalMSE <- numeric(n)
  AICs <- numeric(n)
  for(i in 1:n){
    x <- dataset[-i,]
    y <- dataset[i,-labelcol]
    truey <- dataset[i,labelcol]

    model <- lm(fit, data = x)

    #estimated value from model of left out point
    est <- predict(model, y, type = "response")
    mse <- (truey - est)^2
    totalMSE[i] <- mse

    #AIC of model
    AICs[i] <- AIC(model)
  }
  return (c(mean(totalMSE), mean(AICs)))
}
```

Running backs

I will begin my analysis by modelling Running Back performance. I first subset the data to ensure that I only have running backs in the dataset (rbs). I will then remove any column that does not correspond to running back performance. The columns I include are Name, Position, Draft Position, Projected number of Rushes, Porject number of Rushing Yards, Projected Rushing Average, Projected Number of Receptions, Projected Number of Receiving Yards, Projected Number of Receiving Touchdowns, Projected Number of Fantasy Points, and Actual Number of fantasy points scored (response). The dataset has 85 rows and 11 columns. A sampling of the dataset is provided below.

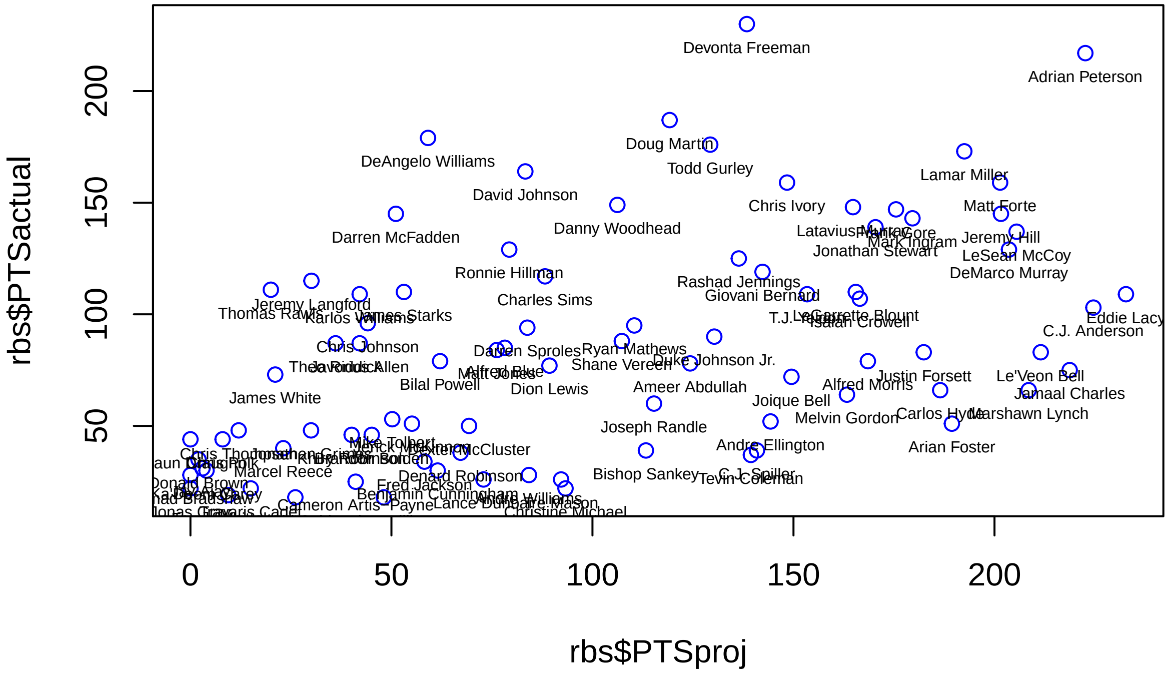
```r
raw_rbs <- first[,"Positionproj"] == "RB"

#include only the necessary columns for running back
rbs <- first[raw_rbs,c(1:6,9,10,12,34,53)]
head(rbs)
```

```
##                Name Positionproj Draft RUSHproj RUSHYDSproj RUSHAVG RECproj
## 4  Adrian Peterson            RB     2      264        1231     4.7      46
## 5   Ahmad Bradshaw            RB   330        0           0     0.0       0
## 9      Alfred Blue            RB    99      127         496     3.9      14
## 10   Alfred Morris            RB    28      250        1064     4.3      22
## 15  Ameer Abdullah            RB    90      125         560     4.5      49
## 16 Andre Ellington            RB    51      172         676     3.9      48
##    RECYDSproj RECTDproj PTSproj PTSactual
## 4         391         2   222.6       217
## 5           0         0     0.0        28
## 9         114         0    78.2        85
## 10        168         1   168.5        79
## 15        402         2   124.3        78
## 16        438         2   144.3        52
```

I have plotted the projected fantasy points by the actual number of fantasy points scored. The plot demonstrates that the ESPN projections for running backs is not very accurate as the data is extremely scattered and there is no general trend.

```
plot(rbs$PTSproj, rbs$PTSactual, col = "blue")
with(rbs, text(PTSactual ~ PTSproj, labels = Name, pos = 1, cex = .5))
```



I will now attempt to fit many different models for the running backs and select the best one using the jackknife cross validated AIC and mean squared error values. The results are held in the rbresults matrix and will be displayed after modelling all 8 of the fits.

```
rbresults <- matrix(0, 8, 3)
colnames(rbresults) <- c("terms", "MSE", "AIC")

#different possible fits
fit1 <- formula(PTSactual ~ PTSproj)
rbresults[1,] <- c(1, crossvalidation(fit1, rbs, 11))
```

```
fit2 <- formula(PTSactual ~ PTSproj + Draft)
rbresults[2,] <- c(2, crossvalidation(fit2, rbs, 11))

fit3 <- formula(PTSactual ~ PTSproj + Draft + RUSHAVG)
rbresults[3,] <- c(3, crossvalidation(fit3, rbs, 11))

fit4 <- formula(PTSactual ~ PTSproj + Draft + RUSHAVG + RECproj)
rbresults[4,] <- c(4, crossvalidation(fit4, rbs, 11))

fit5 <- formula(PTSactual ~ Draft + RUSHproj + RUSHYDSproj + RUSHAVG + RECproj + RECYDSproj + RECTDproj
rbresults[5,] <- c(7, crossvalidation(fit5, rbs, 11))

fit6 <- formula(PTSactual ~ Draft)
rbresults[6,] <- c(1, crossvalidation(fit6, rbs, 11))

fit7 <- formula(PTSactual ~ Draft + RUSHproj + RUSHYDSproj + RUSHAVG + RECproj + RECYDSproj + RECTDproj
rbresults[7,] <- c(8, crossvalidation(fit7, rbs, 11))

fit8 <- formula(PTSactual ~ Draft + RUSHproj + RUSHYDSproj + RECproj + RECYDSproj + RECTDproj + PTSproj
rbresults[8,] <- c(7, crossvalidation(fit8, rbs, 11))

rbresults
```

```
##      terms      MSE      AIC
## [1,]     1 1999.504 879.1526
## [2,]     2 1960.955 877.5088
## [3,]     3 1970.849 879.5059
## [4,]     4 2014.185 881.3931
## [5,]     7 2141.642 883.8632
## [6,]     1 1918.914 875.8967
## [7,]     8 2192.313 885.6611
## [8,]     7 2170.422 883.8264
```

The best model appears to be "fit6" which uses only 1 predictor, "Draft". "Draft" is the draft position of the running back and using only this predictor seems ideal to predict fantasy points for the injury prone running back position. "fit6" has a jackknife crossvalidated MSE of 1918.914 and a jackknife crossvalidated AIC of 875.8967, both of which are the lowest of all 8 models.

The worst model appears to be "fit7" which uses all 8 of the predictors. It is interesting that the most informed model (8 predictors) performs the worst in practice, however, there is probably high correlation among the variables. "fit7" has a jackknife crossvalidated MSE of 2192.313 and a jackknife crossvalidated AIC of 885.6611, both of which are the highest of all 8 models.

Wide Receivers

I will now shift my focus and model Wide Reciever performance. I first subset the data to ensure that I only have wide receivers in the dataset (wrs). I will then remove any column that does not correspond to wide receiver performance. The columns I include are Name, Position, Draft Position, Projected number of Rushes, Porject number of Rushing Yards, Projected Number of Targets, Projected Number of Receptions, Projected Number of Receiving Yards, Projected number of average Receiving yards, Projected Number of Fantasy Points, and Actual Number of fantasy points scored (response). The dataset has 106 rows and 11 columns. A sampling of the dataset is provided below.

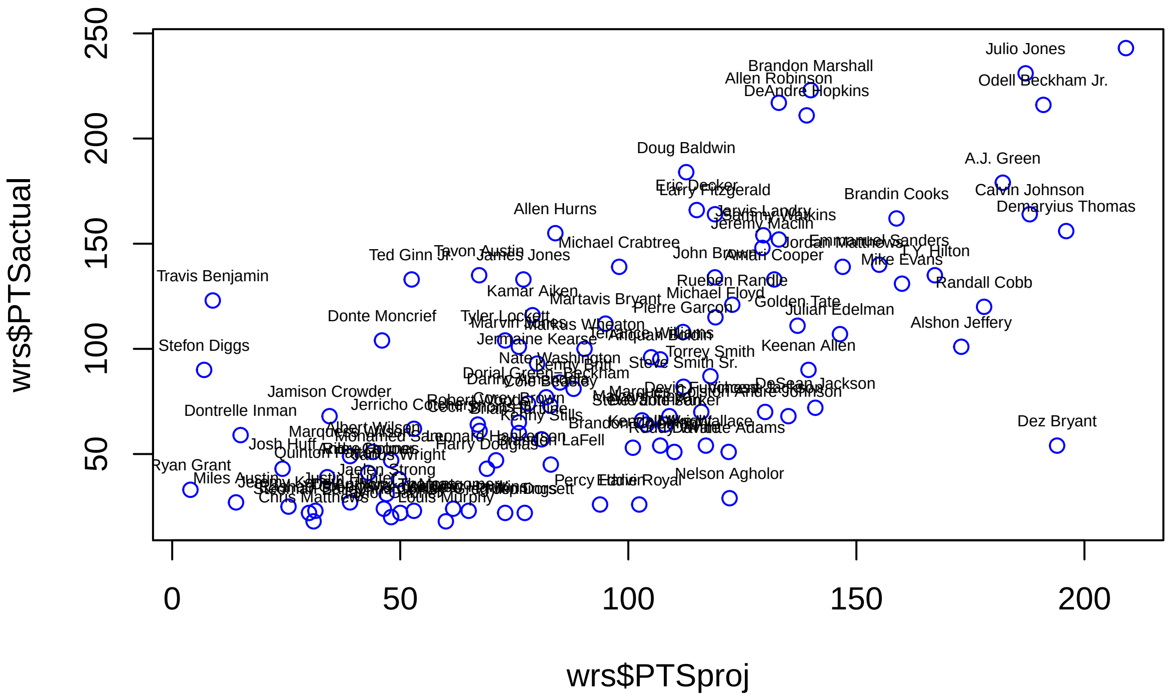
```r
raw_wrs <- first[,"Positionproj"] == "WR"

#include only the necessary columns for wide receivers
wrs <- first[raw_wrs,c(1:5,8:11,34,53)]
wrs[which(is.na(wrs[,"TARproj"])), "TARproj"] = 0 #slight cleaning necessary
head(wrs)
```

```
##               Name Positionproj Draft RUSHproj RUSHYDSproj TARproj RECproj
## 6       A.J. Green           WR    18        5          30   151.0      90
## 7    Albert Wilson           WR   311        0           0    44.1      28
## 11     Allen Hurns           WR   234        4          24    79.7      48
## 12  Allen Robinson           WR    69        0           0   128.3      79
## 13  Alshon Jeffery           WR    21        8          51   147.0      88
## 14    Amari Cooper           WR    48        4          25   123.8      77
##    RECYDSproj RECAVG PTSproj PTSactual
## 6        1274   14.2   182.1       179
## 7         323   11.5    44.0        51
## 11        644   13.4    84.0       155
## 12       1039   13.2   133.0       217
## 13       1222   13.9   173.0       101
## 14       1027   13.3   132.0       133
```

I have plotted the projected fantasy points by the actual number of fantasy points scored for the wide receivers. The plot demonstrates that the ESPN projections for wide receivers is more accurate than the running backs however it is still extremely scattered and there is slight positive correlation to the data (projections somewhat match actual points scored).

```r
plot(wrs$PTSproj, wrs$PTSactual, col = "blue")
with(wrs, text(PTSactual ~ PTSproj, labels = Name, pos = 3, cex= .5))
```



I will now attempt to fit many different models for the wide receivers and select the best one using the jackknife cross validated AIC and mean squared error values. The results are held in the wrresults matrix and will be displayed after modelling all 10 of the fits.

```
wrresults <- matrix(0, 10, 3)
colnames(wrresults) <- c("terms", "MSE", "AIC")

#different possible fits
fit1 <- formula(PTSactual ~ PTSproj)
wrresults[1,] <- c(1, crossvalidation(fit1, wrs, 11))

fit2 <- formula(PTSactual ~ PTSproj + Draft)
wrresults[2,] <- c(2, crossvalidation(fit2, wrs, 11))

fit3 <- formula(PTSactual ~ PTSproj + Draft + RECAVG)
wrresults[3,] <- c(3, crossvalidation(fit3, wrs, 11))

fit4 <- formula(PTSactual ~ PTSproj + Draft + RECAVG + RECproj)
wrresults[4,] <- c(4, crossvalidation(fit4, wrs, 11))

fit5 <- formula(PTSactual ~ Draft + RUSHYDSproj + RUSHproj + RECproj + RECYDSproj)
wrresults[5,] <- c(5, crossvalidation(fit5, wrs, 11))

fit6 <- formula(PTSactual ~ RECproj)
wrresults[6,] <- c(1, crossvalidation(fit6, wrs, 11))

fit7 <- formula(PTSactual ~ Draft + RUSHproj + RUSHYDSproj + TARproj + RECproj + RECYDSproj + RECAVG +
wrresults[7,] <- c(8, crossvalidation(fit7, wrs, 11))

fit8 <- formula(PTSactual ~ Draft + TARproj + RECproj + RECYDSproj + RECAVG + PTSproj)
wrresults[8,] <- c(6, crossvalidation(fit8, wrs, 11))

fit9 <- formula(PTSactual ~ Draft + TARproj + PTSproj)
wrresults[9,] <- c(3, crossvalidation(fit9, wrs, 11))

fit10 <- formula(PTSactual ~ TARproj)
wrresults[10,] <- c(1, crossvalidation(fit10, wrs, 11))

wrresults
```

```
##       terms      MSE      AIC
##  [1,]     1 1811.759 1087.016
##  [2,]     2 1837.130 1088.536
##  [3,]     3 1873.259 1090.521
##  [4,]     4 1811.630 1086.228
##  [5,]     5 2836.776 1089.638
##  [6,]     1 1771.243 1084.728
##  [7,]     8 2995.891 1090.164
##  [8,]     6 1909.937 1089.669
##  [9,]     3 1819.828 1087.191
## [10,]     1 1768.037 1084.585
```

The best model appears to be “fit10” which uses only 1 predictor, “TARproj”. “TARproj” is the ESPN projected number of targets a wide receiver will get and using only this predictor seems ideal to predict fantasy points for this position. The number of targets a wide receiver gets seems to be extremely important to their performance. “fit10” has a jackknife crossvalidated MSE of 1768.037 and a jackknife crossvalidated AIC of 1084.585, both of which are the lowest of all 10 models. This model is closely followed by “fit6” which only uses “RECproj” as a predictor. “RECproj” is the ESPN projected number of receptions a wide receiver will get and this also seems extremely important to a wide receiver’s actual performance.

The worst model appears to be “fit7” which uses all 8 of the predictors. It is interesting that the most informed model (8 predictors) performs the worst in practice, however, there is probably high correlation among the variables. “fit7” has a jackknife crossvalidated MSE of 2995.891 and a jackknife crossvalidated AIC of 1090.164. The MSE is the highest among all of the fits, however, the AIC is second highest only to fit3.

In general, the models for wide receivers have lower jackknife crossvalidated mean squared error when compared to the models for running back even though their are more wide recievers than running backs in the dataset. This suggests that the models for wide recievers perform better in practice than the models for running backs.

Tight Ends

I will now shift my focus and model Tight End performance. I first subset the data to ensure that I only have tight ends in the dataset (tes). I will then remove any column that does not correspond to tight end performance. The columns I include are Name, Position, Draft Position, Projected Number of Targets, Projected Number of Receptions, Projected Number of Receiving Yards, Projected Average of Receiving yards, Projected Number of Fantasy Points, and Actual Number of fantasy points scored (response). The dataset has 44 rows and 9 columns. A sampling of the dataset is provided below.

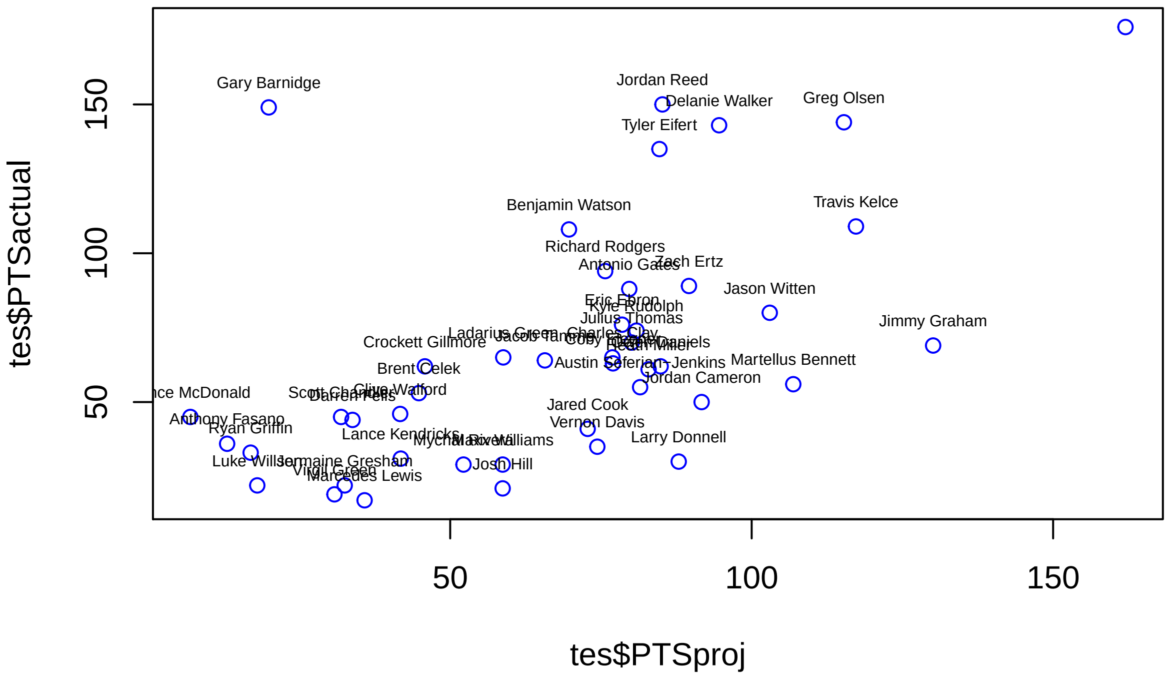
```
raw_tes <- first[,"Positionproj"] == "TE"

#include only the necessary columns for wide receivers
tes <- first[raw_tes,c(1:3,8:11,34,53)]
head(tes)
```

```
##                         Name Positionproj Draft TARproj RECproj RECYDSproj
## 25            Anthony Fasano           TE   385    10.0       7         77
## 27             Antonio Gates           TE   172    62.5      42        509
## 29 Austin Seferian-Jenkins           TE   196    72.3      48        585
## 33            Benjamin Watson           TE   322    63.1      41        462
## 47               Brent Celek           TE   336    36.6      24        270
## 63              Charles Clay           TE   272    76.6      49        531
##    RECAVG PTSproj PTSactual
## 25   11.0    13.0        36
## 27   12.1    79.7        88
## 29   12.2    81.5        55
## 33   11.3    69.7       108
## 47   11.3    44.8        53
## 63   10.8    76.9        65
```

I have plotted the projected fantasy points by the actual number of fantasy points scored for the tight ends. The plot demonstrates that the ESPN projections for tight ends is more accurate than the running backs and wide receivers as the data is not very scattered and there is a positive correlation to the data (projections match actual points scored).

```
plot(tes$PTSproj, tes$PTSactual, col = "blue")
with(tes, text(PTSactual ~ PTSproj, labels = Name, pos = 3, cex = .5))
```



I will now attempt to fit many different models for the tight ends and select the best one using the jackknife cross validated AIC and mean squared error values. The results are held in the teresults matrix and will be displayed after modelling all 10 of the fits.

```
teresults <- matrix(0, 10, 3)
colnames(teresults) <- c("terms", "MSE", "AIC")

#different possible fits
fit1 <- formula(PTSactual ~ PTSproj)
teresults[1,] <- c(1, crossvalidation(fit1, tes, 9))

fit2 <- formula(PTSactual ~ PTSproj + Draft)
teresults[2,] <- c(2, crossvalidation(fit2, tes, 9))

fit3 <- formula(PTSactual ~ PTSproj + Draft + RECAVG)
teresults[3,] <- c(3, crossvalidation(fit3, tes, 9))

fit4 <- formula(PTSactual ~ PTSproj + Draft + RECAVG + RECproj)
teresults[4,] <- c(4, crossvalidation(fit4, tes, 9))

fit5 <- formula(PTSactual ~ Draft + RECproj + RECYDSproj)
teresults[5,] <- c(3, crossvalidation(fit5, tes, 9))

fit6 <- formula(PTSactual ~ RECproj)
teresults[6,] <- c(1, crossvalidation(fit6, tes, 9))

fit7 <- formula(PTSactual ~ TARproj + RECproj)
teresults[7,] <- c(2, crossvalidation(fit7, tes, 9))
```

```r
fit8 <- formula(PTSactual ~ Draft + TARproj + RECproj + RECYDSproj + RECAVG + PTSproj)
teresults[8,] <- c(6, crossvalidation(fit8, tes, 9))

fit9 <- formula(PTSactual ~ Draft + TARproj + PTSproj)
teresults[9,] <- c(3, crossvalidation(fit9, tes, 9))

fit10 <- formula(PTSactual ~ TARproj)
teresults[10,] <- c(1, crossvalidation(fit10, tes, 9))

teresults
```

```
##       terms      MSE      AIC
##  [1,]     1 1213.259 428.1559
##  [2,]     2 1242.150 429.4240
##  [3,]     3 1346.279 430.9301
##  [4,]     4 1387.993 430.6710
##  [5,]     3 1291.089 429.6789
##  [6,]     1 1191.044 427.1958
##  [7,]     2 1235.638 428.9526
##  [8,]     6 1686.413 433.4905
##  [9,]     3 1336.893 431.0324
## [10,]     1 1204.536 427.7606
```

The best model appears to be "fit6" which uses only 1 predictor, "RECproj". "RECproj" is the ESPN projected number of receptions a tight end will get and using only this predictor seems ideal to predict fantasy points for this position. The number of receptions a tight end gets seems to be extremely important to their performance. "fit6" has a jackknife crossvalidated MSE of 1191.044 and a jackknife crossvalidated AIC of 427.1958, both of which are the lowest of all 10 models. This model is closely followed by "fit10" which only uses "TARproj" as a predictor. "TARproj" is the ESPN projected number of targets a tight end will get and this also seems extremely important to a tight end's actual performance. However, the combined model of both "RECproj" and "TARproj" did not perform as well as the other 2 models (fit7).

The worst model appears to be "fit8" which uses all 6 of the predictors. It is interesting that the most informed model (6 predictors) performs the worst in practice, however, there is probably high correlation among the variables. "fit8" has a jackknife crossvalidated MSE of 1686.413 and a jackknife crossvalidated AIC of 433.4905 both of which are the highest among the 10 fits.

In general, the models for tightends have lower jackknife crossvalidated mean squared error and AICs when compared to the models for running backs and wide receivers. This is probably do to the fact that there are less tight ends than the wide receivers and running backs.

Results

In general the jackknife crossvalidation worked pretty well for model evaluation for this dataset. It was interesting to see that the smaller models (with 1 predictor) tended to do much better than the larger models (with all predictors). The larger models tend to have higher jackknife crossvalidated MSEs and AICs. The fits for wide receivers and tightends was much better than the fits for running backs in general.