444 project code

Zizhou Wang

April 6, 2018

Motivation and introduction of the problem

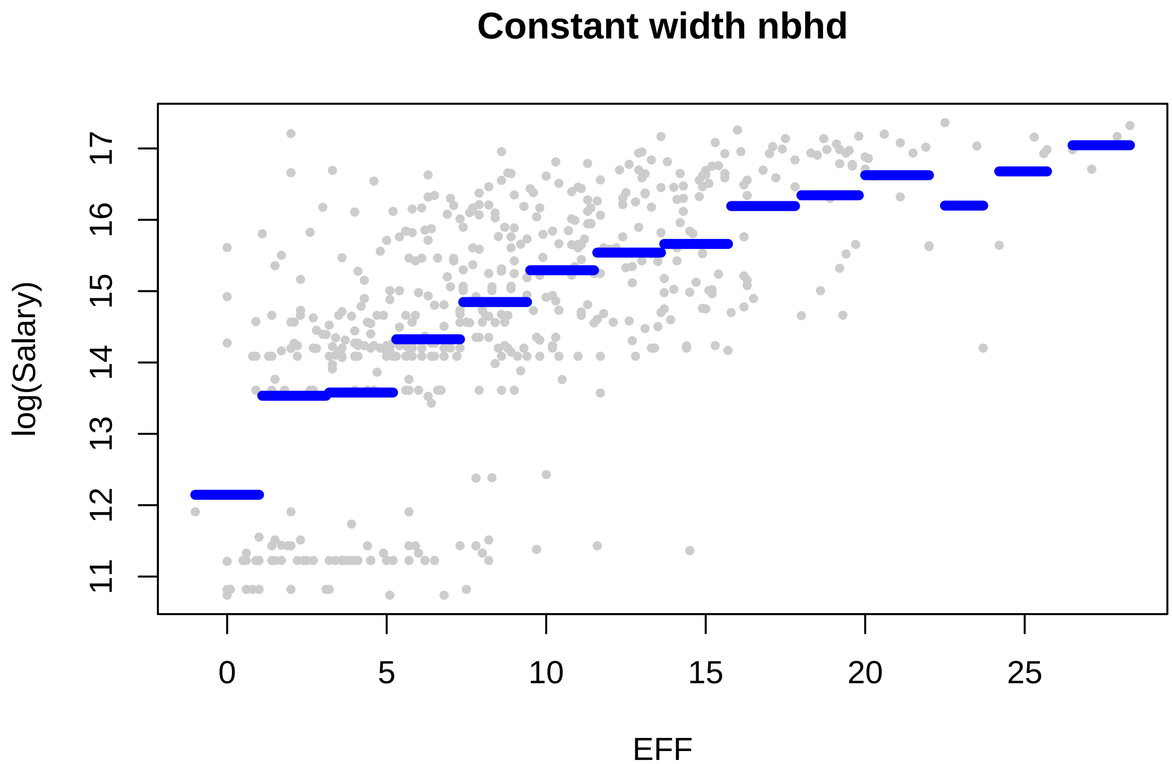
After seeing our final dataset, we raised the question of if it is possible to predict an NBA player's salary based on his previous year's performance. Our project consists of three major parts, smoothing spline, random foresets, and boosting. We used smoothing spline to model our data, by testing different models constructed by different combinations of explanatory variates, we were able to get the best model for our data on hand to predict players' salary. We used random forest to find the importance of our explanatory variates, and we were able to find the most important variates to minimize the error. We were also able to find the importance of each variable using the gradient boosting method.

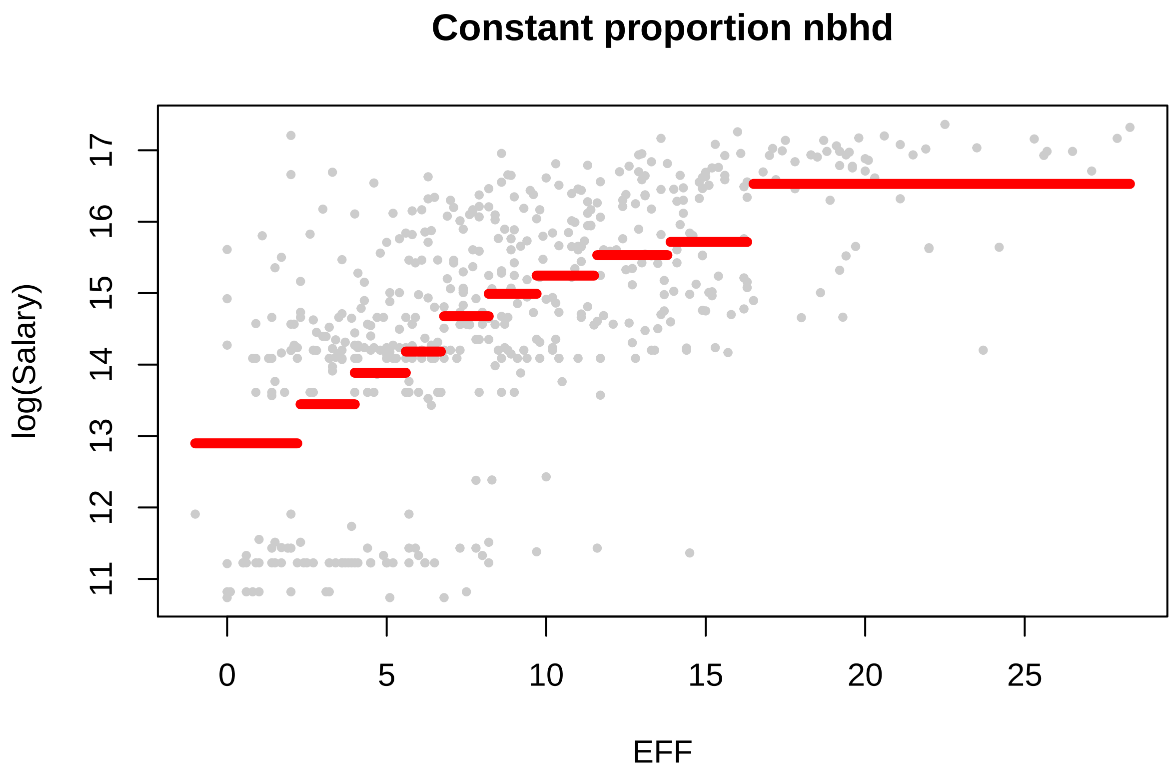
Data

```r
data <- read.csv("./combined.csv", header=TRUE)
data$WR = data$W/data$GP
head(data)
```

```
##   ID           Name   Salary Team AGE GP  W  L  MIN  PTS  FGM  FGA FGPER
## 1  1  Stephen Curry 34682550  GSW  30 51 41 10 32.0 26.4  8.4 16.9  49.5
## 2  2   LeBron James 33285709  CLE  33 76 46 30 37.1 27.6 10.6 19.4  54.7
## 3  3   Paul Millsap 31269231  DEN  33 32 17 15 29.3 14.8  5.4 11.2  48.2
## 4  4 Gordon Hayward 29727900  BOS  28  1  0  1  5.3  2.0  1.0  2.0  50.0
## 5  5  Blake Griffin 29512900  DET  29 58 28 30 33.8 21.3  7.5 17.0  43.8
## 6  6     Kyle Lowry 28703704  TOR  32 71 53 18 32.3 16.6  5.2 12.1  43.3
##   TPM TPA TPPER FTM FTA FTPER OREB DREB REB AST TOV STL BLK  PF   FP DD2
## 1 4.2 9.8  42.3 5.5 5.9  92.1  0.7  4.4 5.1 6.1 3.0 1.6 0.2 2.2 43.8   5
## 2 1.8 4.9  36.1 4.6 6.3  73.0  1.2  7.4 8.6 9.1 4.2 1.5 0.9 1.7 54.5  47
## 3 1.1 2.9  36.6 2.9 4.2  70.7  1.4  4.8 6.3 2.8 1.9 1.2 1.1 2.6 31.3   1
## 4 0.0 1.0   0.0 0.0 0.0   0.0  0.0  1.0 1.0 0.0 0.0 0.0 0.0 1.0  3.2   0
## 5 1.9 5.5  34.8 4.4 5.6  78.6  1.3  6.1 7.3 5.7 2.8 0.7 0.3 2.4 38.8  16
## 6 3.1 7.6  40.9 3.0 3.5  85.9  0.9  4.7 5.6 6.8 2.3 1.1 0.2 2.5 35.3  22
##   TD3 PLUSMINUS Position Country Draft.Round Draft.Number SGap        WR
## 1   0       9.5        G     USA           1            7    1 0.8039216
## 2  16       0.6        F     USA           1            1    1 0.6052632
## 3   0       2.3        F     USA           2           47    1 0.5312500
## 4   0       3.0        F     USA           1            9    1 0.0000000
## 5   3       1.1        F     USA           1            1    1 0.4827586
## 6   3       5.0        G     USA           1           24    1 0.7464789
```

One of the most straight forward way to evaluate the performance of an NBA player is to look at his "Points per Game", "Assists per Game", and "Rebounds per Game", which are the 3 most mentioned statistics when NBA analysts and fans make comparison to players. We initially tried to find a relationship between the PRA(Points + Rebounds + Assists per game) and the Salary of an NBA player. However, we realized that it will almost always introduce a bias, because it does not tell us the full image of the player's ability. For example, Points are usually easier to get compared to Assists and Rebounds. When a player scores, they will either get two points or three points, potentially earning an extra Free Throw, which counts as one more point. When a player gets an Assist or a Rebound, the count only goes up by 1. Having 10 Rebounds or 10 Assists after a game is considered a good performance, but having 10 Points for a game is usually average. The PRA also introduces a heavier weight on the player's offensive ability than his defensive ability on the court, since Points, Assists, and Offensive Rebounds all happen at the front court. Therefore, we found a better way to determine the efficiency of an NBA player, which is to look at his EFF, calculated by EFF = PTS + REB + AST + STL + BLK - FGM - FTM - TOV, where all variates are averaged per game. The EFF takes Steal (STL), Block (BLK), Field Goal Missed (FGM), Free Throw Missed (FTM), and Turn Over (TOV) into account, which adds the defensive ability (STL and BLK) and inefficiency (FGM, FTM, TO) into the equation.





We first want to see what our data look like when EFF is plotted aginst log(Salary), even though our data look to be bimodel, we can still observe an increasing trend, according to the piece wise fitting using neighbourhood.

Data Preprocessing

We initially started looking at the data for salary of NBA players at https://www.basketball-reference.com/contracts/players.html (updates constantly), which had 582 records of player salaries for year 2017-2018 at the time. However, we had to remove some duplicated records for players with different salaries on different teams. This is because some players could get cut by teams half way through the season, and sometimes they would get picked up by another team, which resulted in having multiple player contracts in a year. An example for this is Rajon Rondo, who was waived by the Chicago Bulls, signed a contract with New Orleans Pelicans right after.

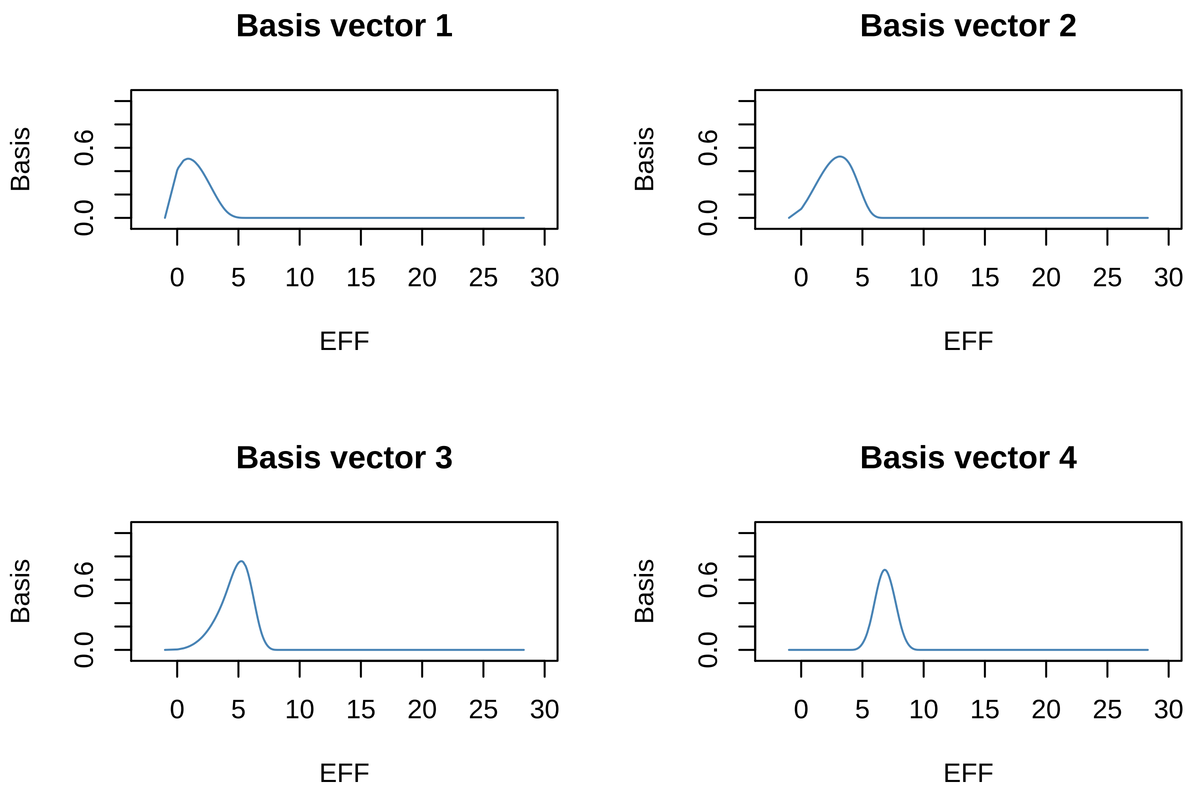
During the process of matching players' statistics with their salaries, we encountered some cases where some the player information for a couple of players listed in our salary could not be found. An example is Walt Lemon, Jr., who is initially listed in our salary data. We were not able to find his player information on https://stats.nba.com/players/bio/, which contains data that we thought could be important in our analysis. Therefore, we removed these records.

For the "Position" categorical variable in our dataset, we stated in our proposal that we would be using 5 values, PG (Point Guard), SG (Shooting Guard), SF (Small Forward), PF (Power Forward), and C (Center). It turns out that many guards in the NBA today are "combo guards", which means they can both play at the Point Guard and Shooting Guard position (e.g. James Harden). There are also many forwards in the NBA who can both play at the Small Forward and Power Forward position (e.g. Lebron James). We reduced the number of values to 3, grouping PG and SG as G (Guard), SF and PF as F (Forward). In addition, there are some players who are "swingman", meaning they can both play at the SG and SF position (e.g. Jimmy Butler). Since this is not a frequent case, we chose a position for each of them based on which position they had mostly been playing at this season (2017-2018) and our knowledge to the players.

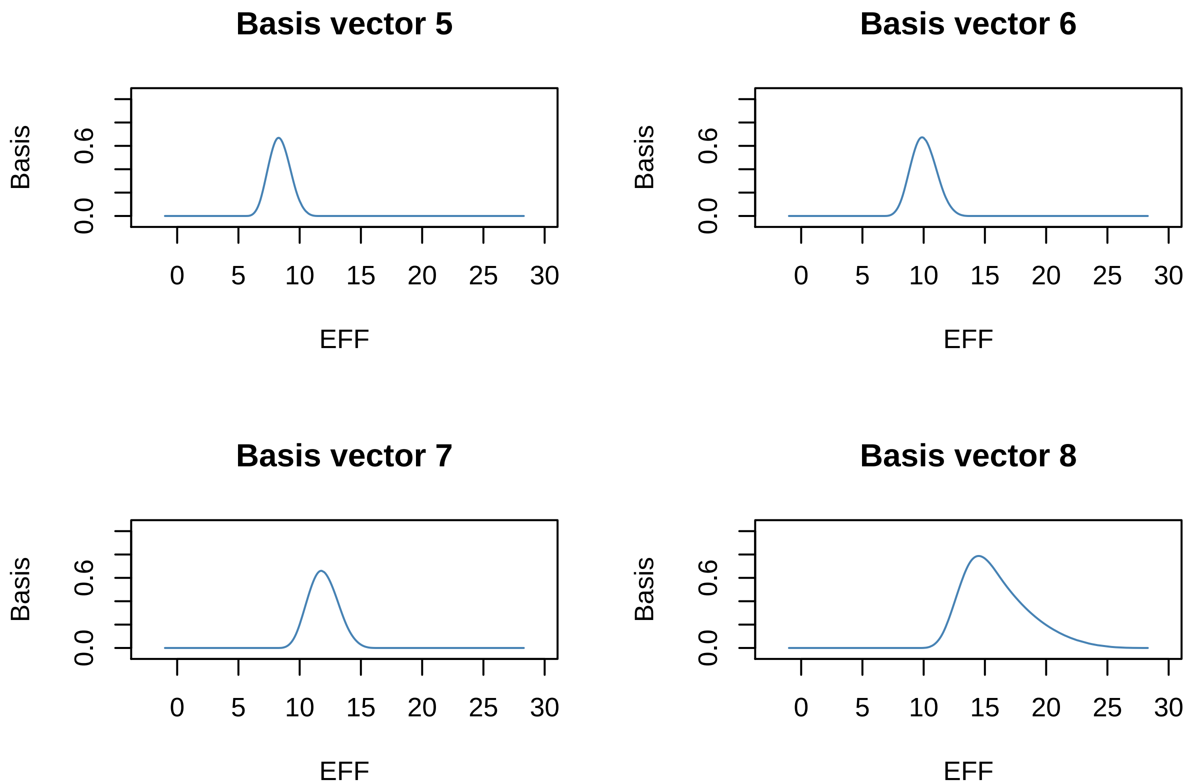
Our eventually obtained our final dataset, which contains 515 records and does not contain any N/A's.

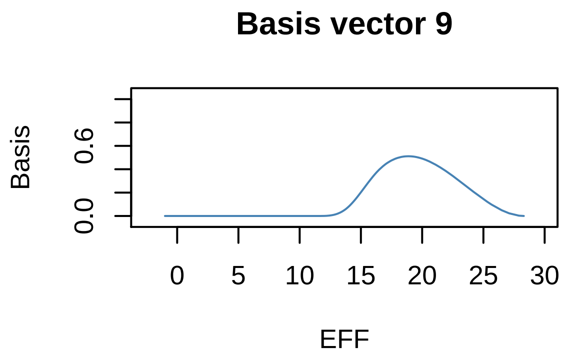
We then realized a couple outliers in our dataset. For example, Gordon Hayward was horribly injured during his very first game at the beginning of the year. He was not able to return for the rest of the season. With the 4th highest salary on our list, he would be an extreme outlier in our models with minimal statistical contribution. However, this does not mean that he is not worth the salary, since he was only able to play for about 5 minutes before the injury. Therefore, we would like to exclude him when building our models, along with several other players in similar conditions.

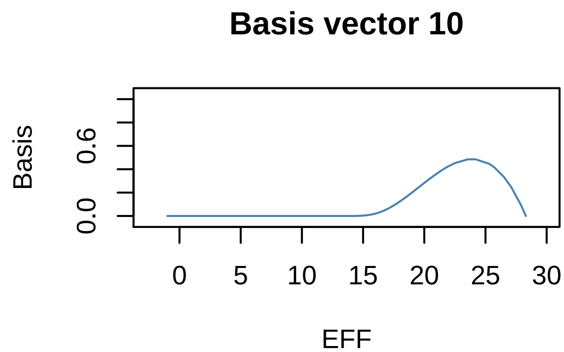
Smoothing

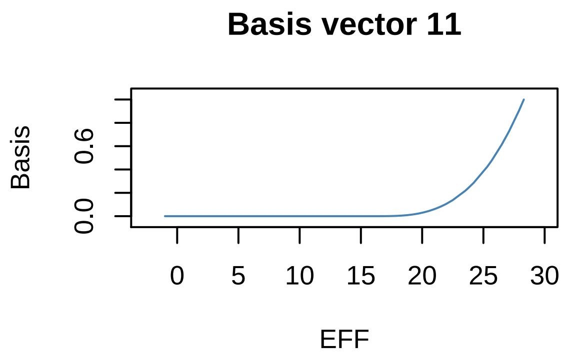


Basis vector 1
Basis vector 2
Basis vector 3
Basis vector 4
Basis
0.0
0.6
0
5
10
15
20
25
30
EFF


Basis vector 5
Basis vector 6
Basis vector 7
Basis vector 8
Basis
0.0
0.6
0
5
10
15
20
25
30
EFF

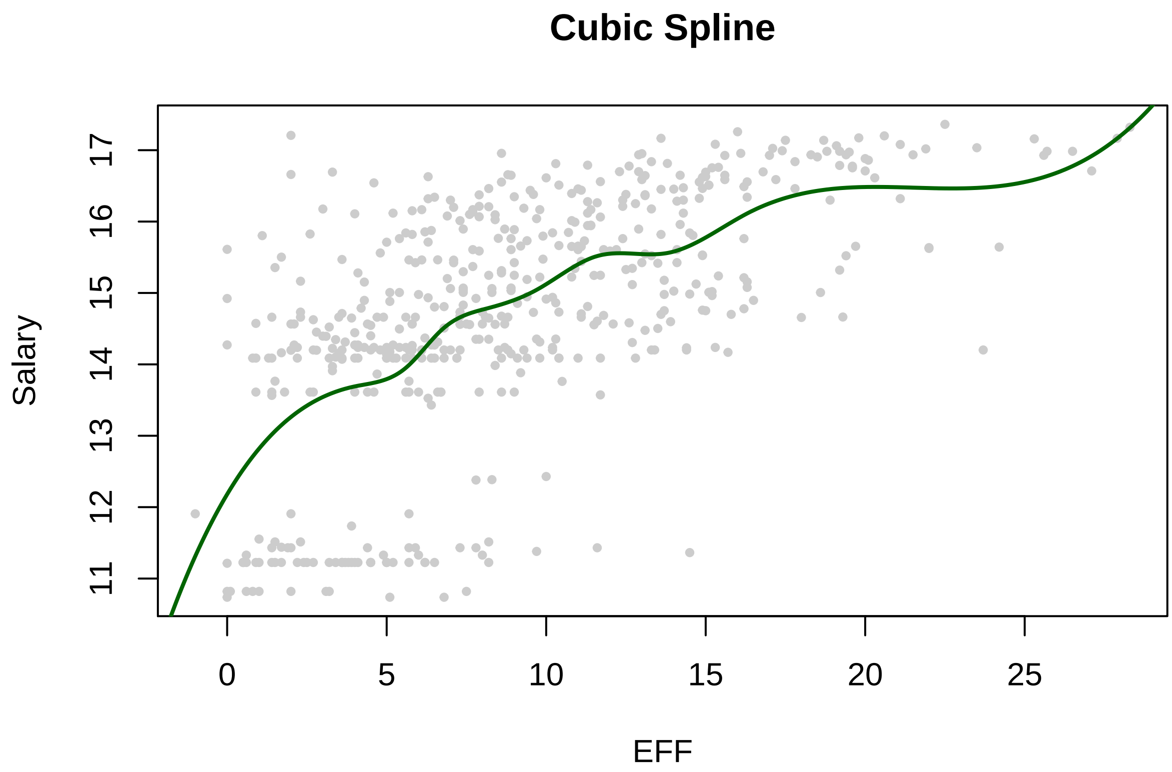






We then try to fit a cubic spline to our data. First we need to get its basis functions for our fitted model, which can be illustrated by plotting them as a function of EFF. The basis functions are clearly not polynomials. The estimated smooth will be a linear combination of these basis functions.

```
## Warning in bs(x, degree = 3L, knots = structure(c(4, 5.6, 6.76, 8.2, 9.7, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases
```

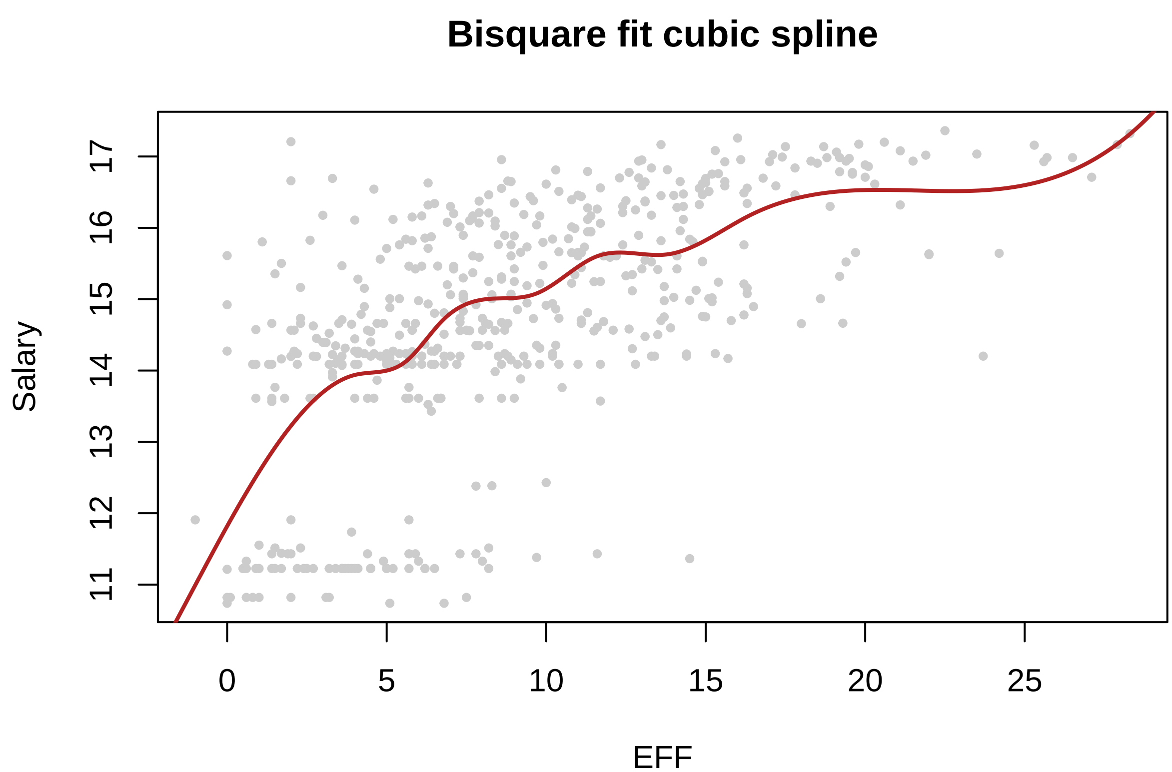


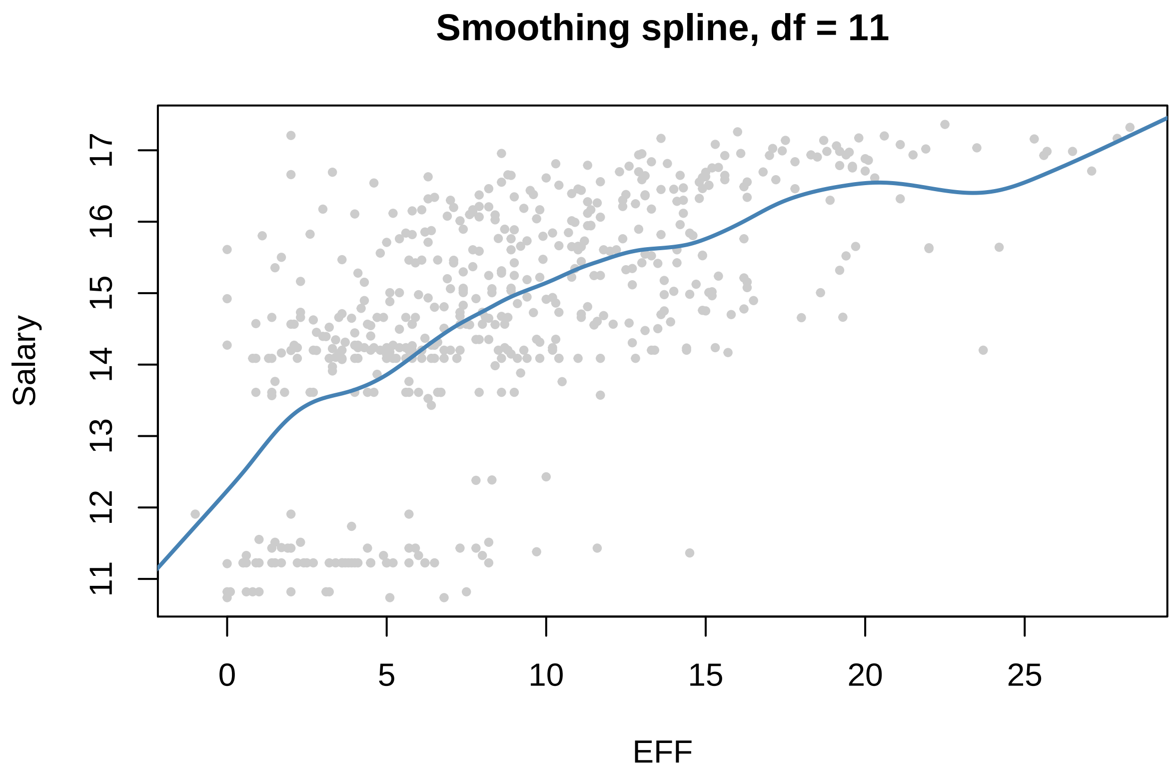

```
##
## Call:
## lm(formula = y ~ bs(x, degree = p, knots = knots_p))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2986 -0.8279  0.2941  0.8438  3.9449
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          11.3123     0.9780  11.567  < 2e-16
## bs(x, degree = p, knots = knots_p)1   1.6553     1.6813   0.985 0.325340
## bs(x, degree = p, knots = knots_p)2   2.4031     0.8984   2.675 0.007720
## bs(x, degree = p, knots = knots_p)3   2.4432     1.1134   2.194 0.028670
## bs(x, degree = p, knots = knots_p)4   3.3388     1.0029   3.329 0.000936
## bs(x, degree = p, knots = knots_p)5   3.4678     1.0755   3.224 0.001345
## bs(x, degree = p, knots = knots_p)6   3.6983     1.0523   3.514 0.000481
## bs(x, degree = p, knots = knots_p)7   4.3980     1.0675   4.120 4.43e-05
## bs(x, degree = p, knots = knots_p)8   4.0580     1.0308   3.937 9.42e-05
## bs(x, degree = p, knots = knots_p)9   6.0080     1.2834   4.681 3.67e-06
## bs(x, degree = p, knots = knots_p)10  4.4082     1.6011   2.753 0.006115
## bs(x, degree = p, knots = knots_p)11  6.0017     1.3079   4.589 5.63e-06
##
## (Intercept)                         ***
## bs(x, degree = p, knots = knots_p)1
## bs(x, degree = p, knots = knots_p)2 **
```

```
## bs(x, degree = p, knots = knots_p)3  *
## bs(x, degree = p, knots = knots_p)4  ***
## bs(x, degree = p, knots = knots_p)5  **
## bs(x, degree = p, knots = knots_p)6  ***
## bs(x, degree = p, knots = knots_p)7  ***
## bs(x, degree = p, knots = knots_p)8  ***
## bs(x, degree = p, knots = knots_p)9  ***
## bs(x, degree = p, knots = knots_p)10 **
## bs(x, degree = p, knots = knots_p)11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.34 on 503 degrees of freedom
## Multiple R-squared:  0.3992, Adjusted R-squared:  0.386
## F-statistic: 30.38 on 11 and 503 DF,  p-value: < 2.2e-16
```

We then fitted the cubic spline to the data.

```
## Warning in bs(x, degree = 3L, knots = structure(c(4, 5.6, 6.76, 8.2, 9.7, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases
```






```
## Warning in bs(x, degree = 3L, knots = structure(c(4, 5.6, 6.76, 8.2, 9.7, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(x, degree = 3L, knots = structure(c(4, 5.6, 6.76, 8.2, 9.7, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## [1] 1.84606
```

Random Forest

We would like to utilize random forest to determine the importance of explanatory variates.

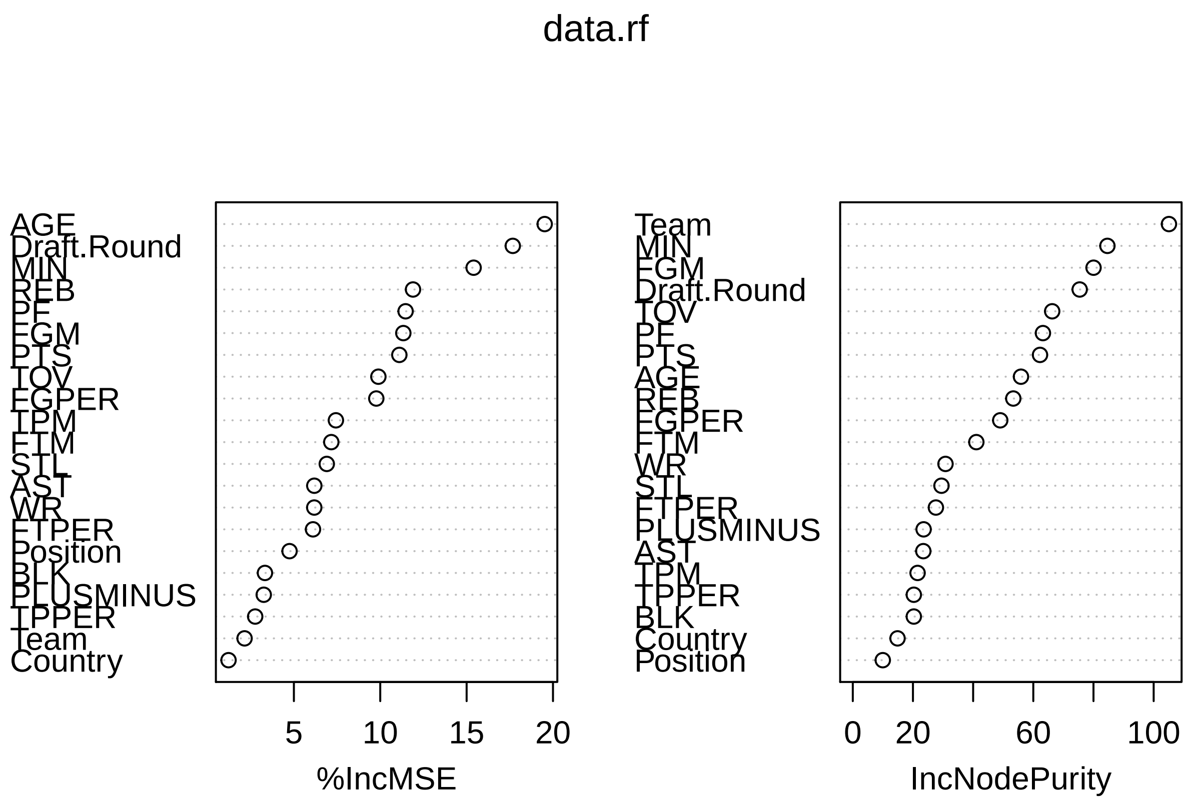
```
## Warning: package 'randomForest' was built under R version 3.4.4
```

```
## randomForest 4.6-14
```

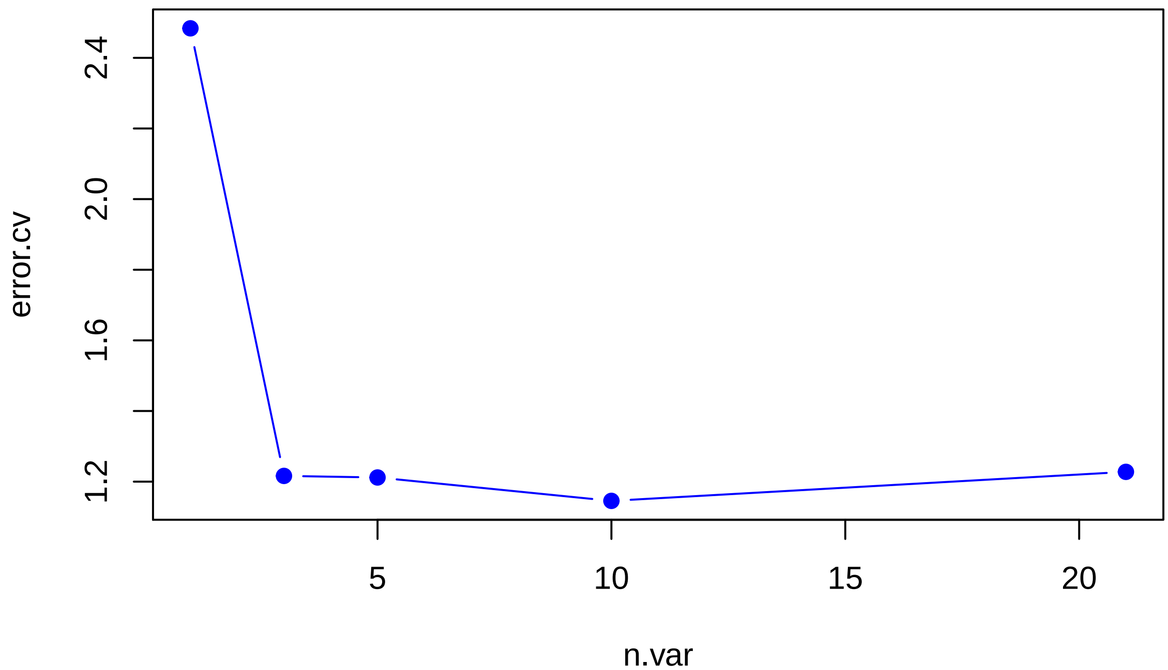
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##             IncNodePurity
## PTS              62.22661
## REB              53.34613
## AST              23.43383
## TOV              66.27728
## STL              29.47236
## BLK              20.27620
## Team            105.08425
## WR               30.81825
## AGE              55.89737
## FGM              80.02194
## FGPER            49.00783
## TPM              21.53979
## TPPER            20.27744
## FTM              41.06172
## FTPER            27.62203
## PF               63.19264
## PLUSMINUS        23.53919
## Position          9.96612
## Country          14.87307
## MIN              84.62544
## Draft.Round      75.41164
```

```
##               %IncMSE
## PTS         11.105887
## REB         11.896329
## AST          6.181612
## TOV          9.891187
## STL          6.904804
## BLK          3.324602
## Team         2.144388
## WR           6.179159
## AGE         19.519539
## FGM         11.336279
## FGPER        9.771315
## TPM          7.432304
## TPPER        2.759934
## FTM          7.165750
## FTPER        6.106386
## PF          11.466293
## PLUSMINUS    3.255276
## Position     4.748315
## Country      1.216483
## MIN         15.405465
## Draft.Round 17.672138
```

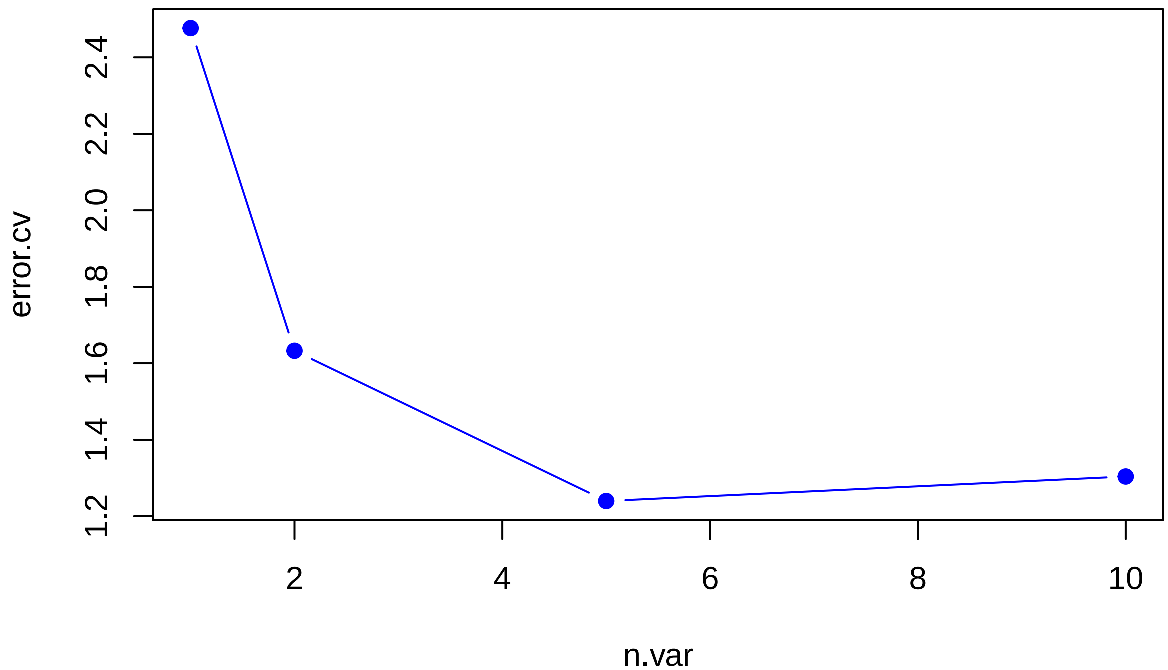


data.rf
AGE
Draft.Round
MIN
REB
PF
FGM
PTS
TOV
FGPER
TPM
FTM
STL
AST
WR
FTPER
Position
BLK
PLUSMINUS
TPPER
Team
Country
5
10
15
20
%IncMSE
Team
MIN
FGM
Draft.Round
TOV
PF
PTS
AGE
REB
FGPER
FTM
WR
STL
FTPER
PLUSMINUS
AST
TPM
TPPER
BLK
Country
Position
0
20
60
100
IncNodePurity

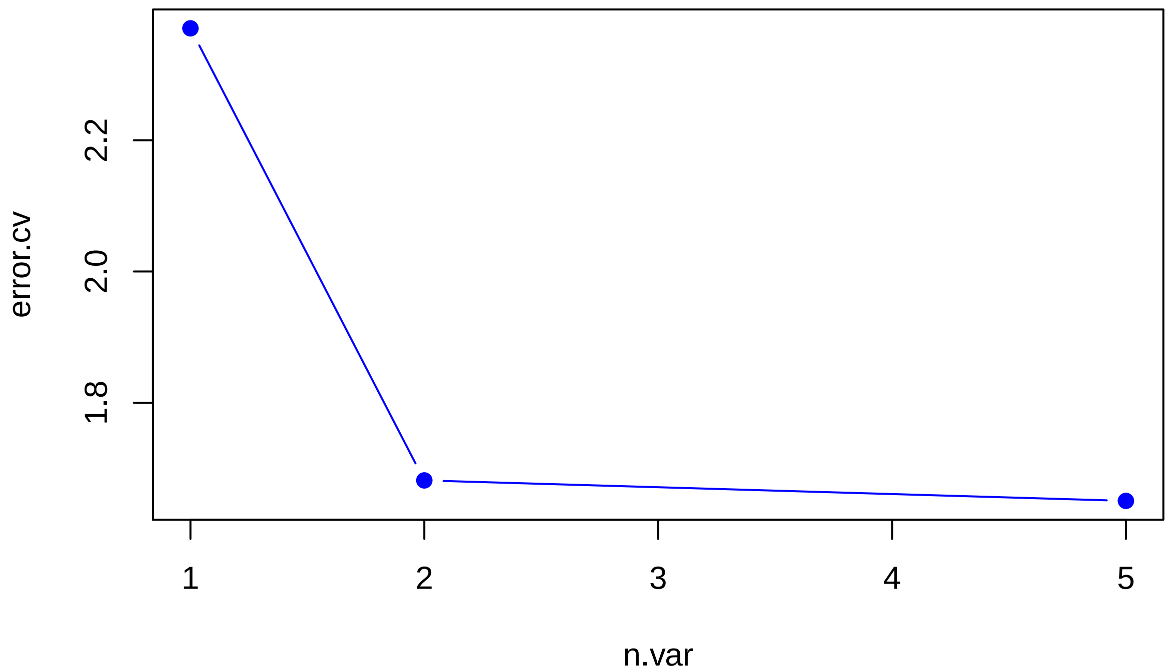


We used PTS, REB, AST, TOV, STL, BLK, Team, WR, AGE, FGM, FGPER, TPM, TPPER, FTM, FTPER, PF, PLUSMINUS, Position, Country, MIN, and Draft.Round against log(Salary) for the random forest. We did not choose to include Draft.Number because it is a categorical variate with 60 different potential values, but random forest does not accept categorical predictors with more than 53 categories.

The result suggests that the error of cross validation is the lowest for 10 explanatory variates, at about 1.18. We then choose the top 10 most important variates based on RSS, Team, MIN, FGM, Draft.Round, TOV, PF, PTS, AGE, REB, and FGPER, and run the process again.



The result from the second run suggests that the error of cross validation is the lowest when there are 5 explanatory variates, at around 1.22. We then choose the top 5 most important variates againbased on RSS, which are Team, MIN, FGM, PTS, and AGE, and run the process again.



The result from the third run suggests that the error of cross validation is the lowest when there are 5 explanatory variates, at around 1.64. We can say that the Team, MIN, FGM, PTS, and AGE are important variates based on cross validation. Also, AGE seems to be the most important for predictive purposes.

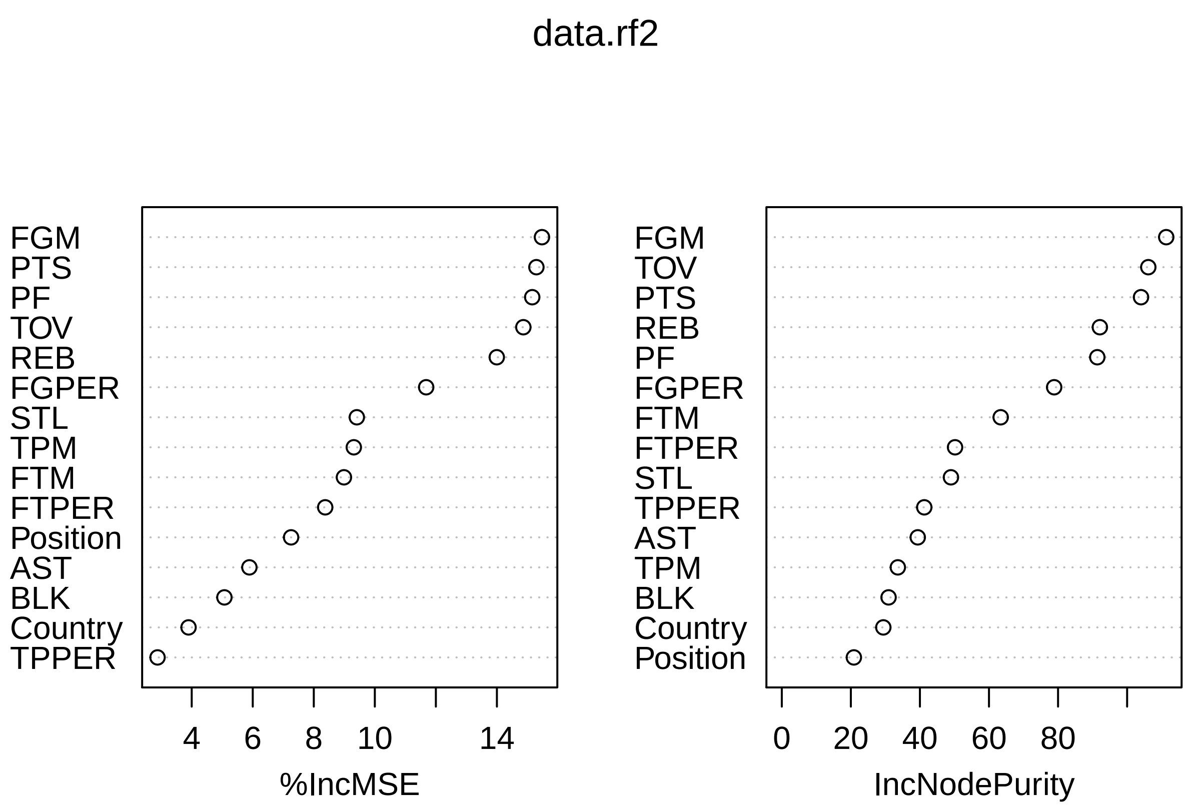
Player self-evaluate and improvements

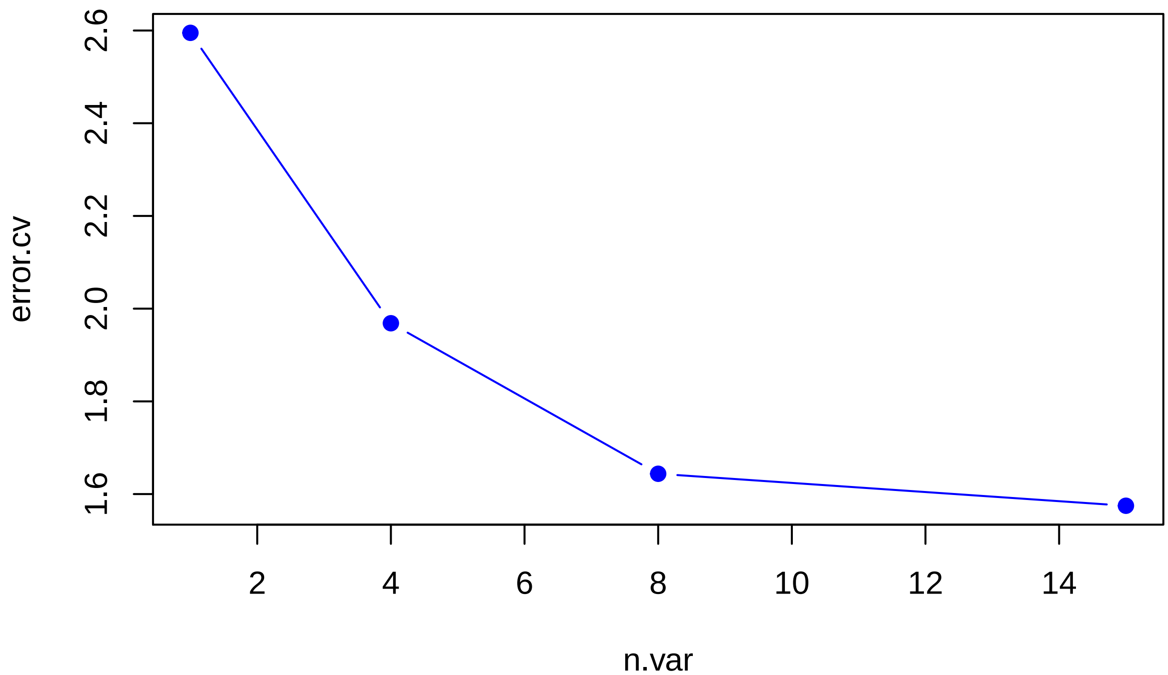
For NBA players who would like to self-evaluate and who are trying to see what they can work on to receive a better contract, we can consider how Salary depends on just those explanatory variates that were under the control of the NBA player. Therefore, we removed Team, MIN, WR, AGE, Draft.Round, and PLUSMINUS. Age is obviously an uncontrollable variate. We think players rarely have control for Team, MIN, and Draft.Round since it does not depend on players' previous NBA performance, instead, these would depend on the decisions from coach and the organization Also, WR and PLUSMINUS have a lot to do with the teammates of the NBA player we are trying to analyze, so we decided to take these out of consideration as well. This move left us with 15 explanatory variates.

```
##          IncNodePurity
## PTS          104.02535
## REB           92.10307
## AST           39.37527
## TOV          106.13224
## STL           48.97887
## BLK           30.92172
## FGM          111.32397
## FGPER         78.87601
## TPM           33.60315
## TPPER         41.25709
```

```
## FTM           63.38671
## FTPER         50.17051
## PF            91.36678
## Position      20.87296
## Country       29.39918
```

```
##             %IncMSE
## PTS       15.293625
## REB       13.995687
## AST        5.889595
## TOV       14.865521
## STL        9.409757
## BLK        5.070758
## FGM       15.475682
## FGPER     11.682218
## TPM        9.311112
## TPPER      2.879952
## FTM        8.987660
## FTPER      8.374388
## PF        15.156821
## Position  7.256454
## Country   3.896222
```





The cross validation suggests that all 15 explanatory variates are important, at an error of around 1.58. The result also suggests that FGM, TOV, REB, and PTS are the most important.

Boosting

We then used the Gradient Boosting method to determine the importance of explanatory variates, and see if it shows a different result compared to Random Forest.

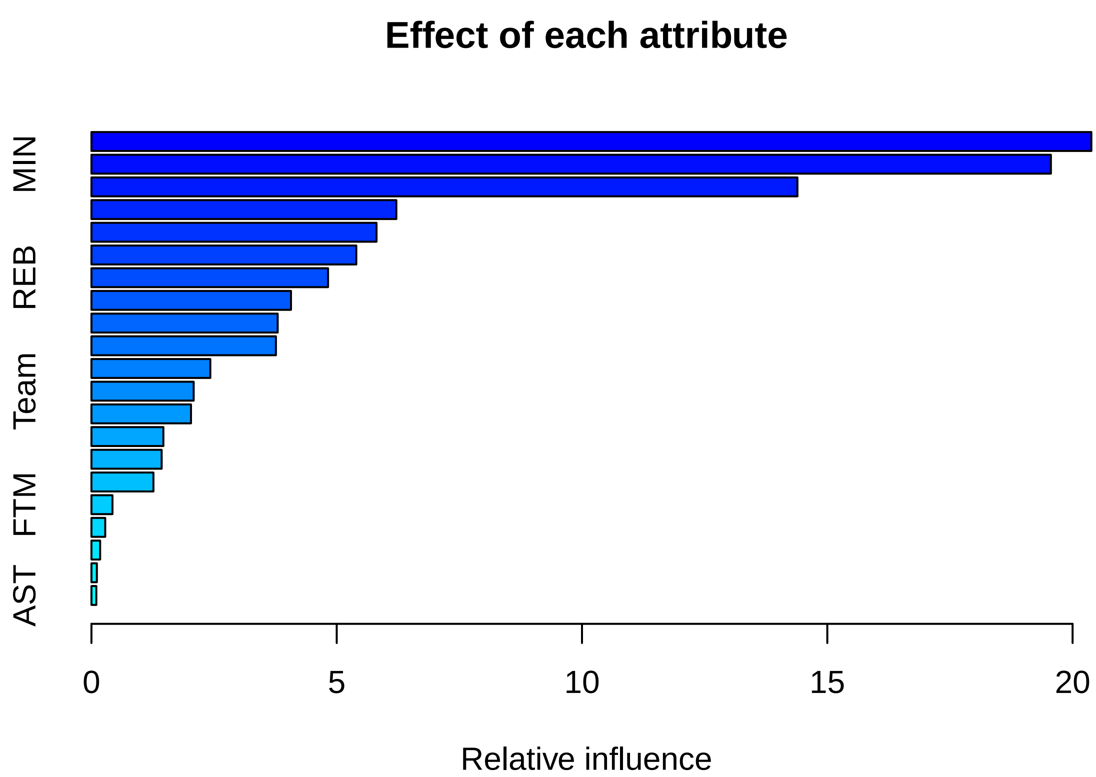
```
## Warning: package 'gbm' was built under R version 3.4.4
```

```
## Loading required package: survival
```

```
## Loading required package: lattice
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

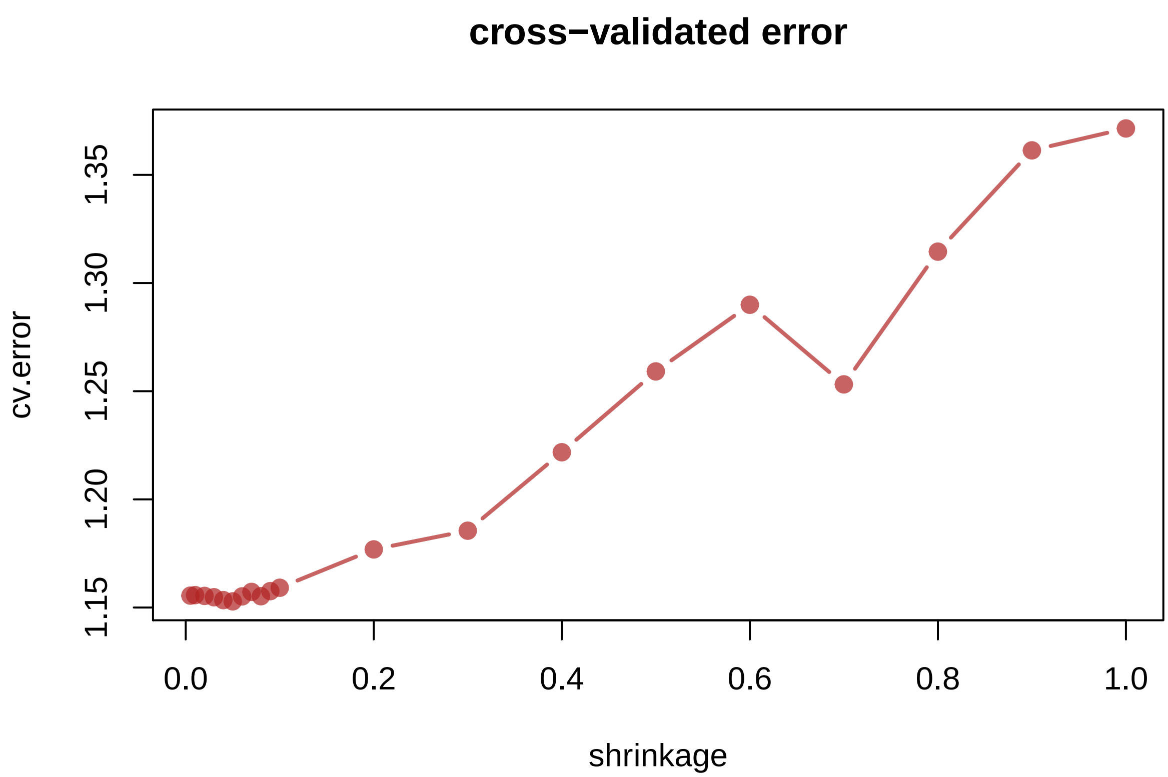



```
##                       var     rel.inf
## Draft.Round Draft.Round 20.38274616
## MIN                   MIN 19.55791333
## AGE                   AGE 14.39111747
## PF                     PF  6.21525093
## TOV                   TOV  5.81122466
## FTPER               FTPER  5.39890328
## REB                   REB  4.82325645
## PTS                   PTS  4.06743224
## Country           Country  3.79626734
## FGPER               FGPER  3.76037604
## FGM                   FGM  2.42323161
## Team                 Team  2.08433641
## PLUSMINUS       PLUSMINUS  2.02944182
```

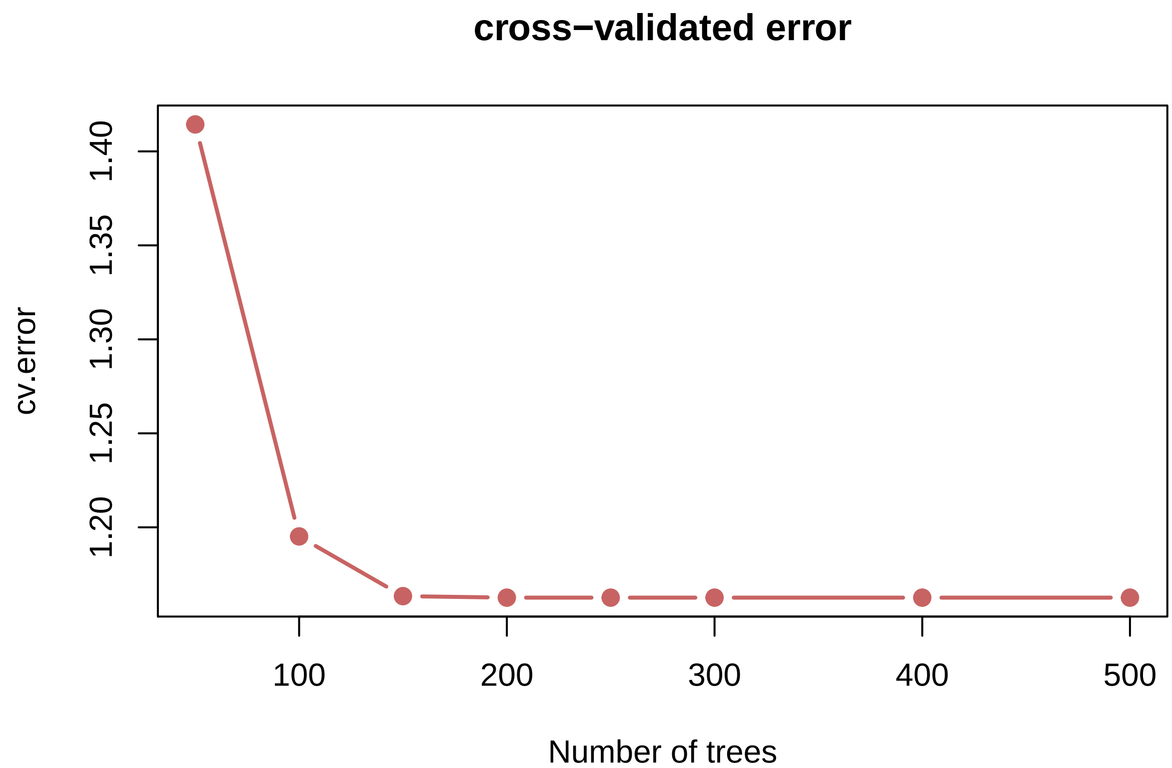
```
## Position       Position  1.46659631
## TPPER             TPPER  1.43128869
## WR                   WR  1.26356778
## FTM                 FTM  0.42841298
## TPM                 TPM  0.28111832
## STL                 STL  0.17711781
## BLK                 BLK  0.11080457
## AST                 AST  0.09959581
```

We used PTS, REB, AST, TOV, STL, BLK, Team, WR, AGE, FGM, FGPER, TPM, TPPER, FTM, FTPER, PF, PLUSMINUS, Position, Country, MIN, and Draft.Round against log(Salary), which is the same as what we used for Random Forest.

The result shows that Draft Round, Minutes played, and Age are the 3 major variates, with much higher influece over others.



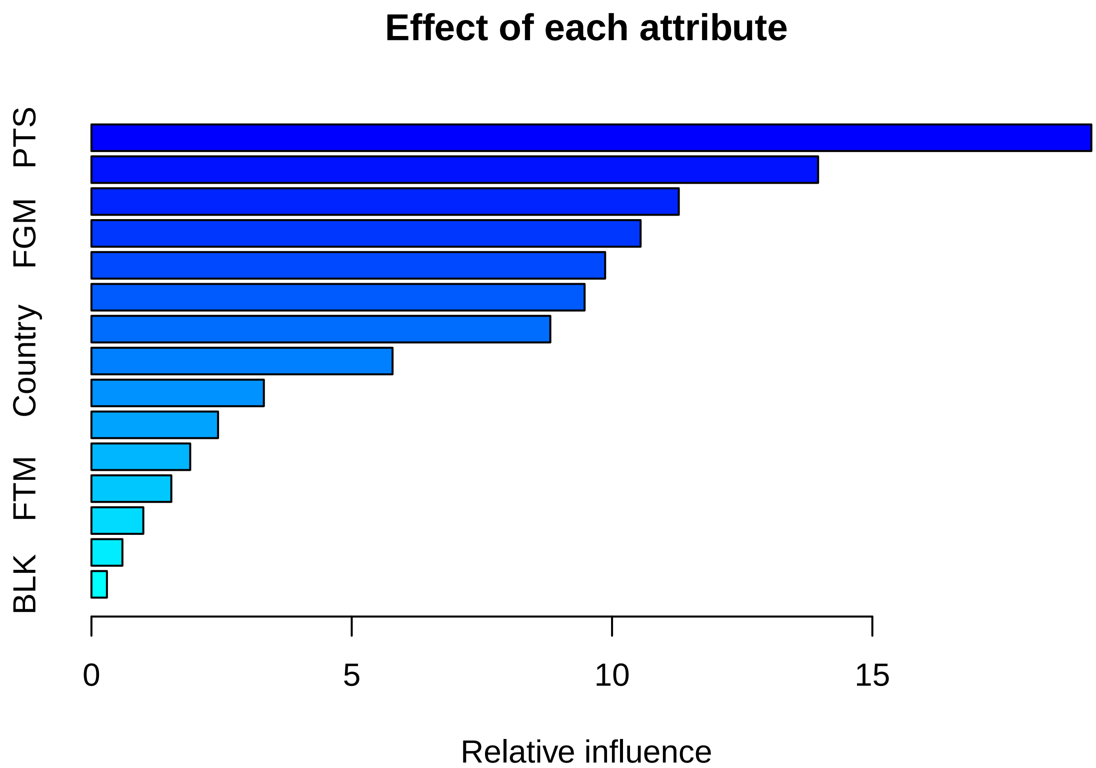
We see that the best learning rate for these 21 explainatory variates is at around 0.06, with cv error around 1.15



We see that the best value for M for more explainatory variables is at about 150, with cv error around 1.16. We can see that the return is very small when M is bigger than 150.

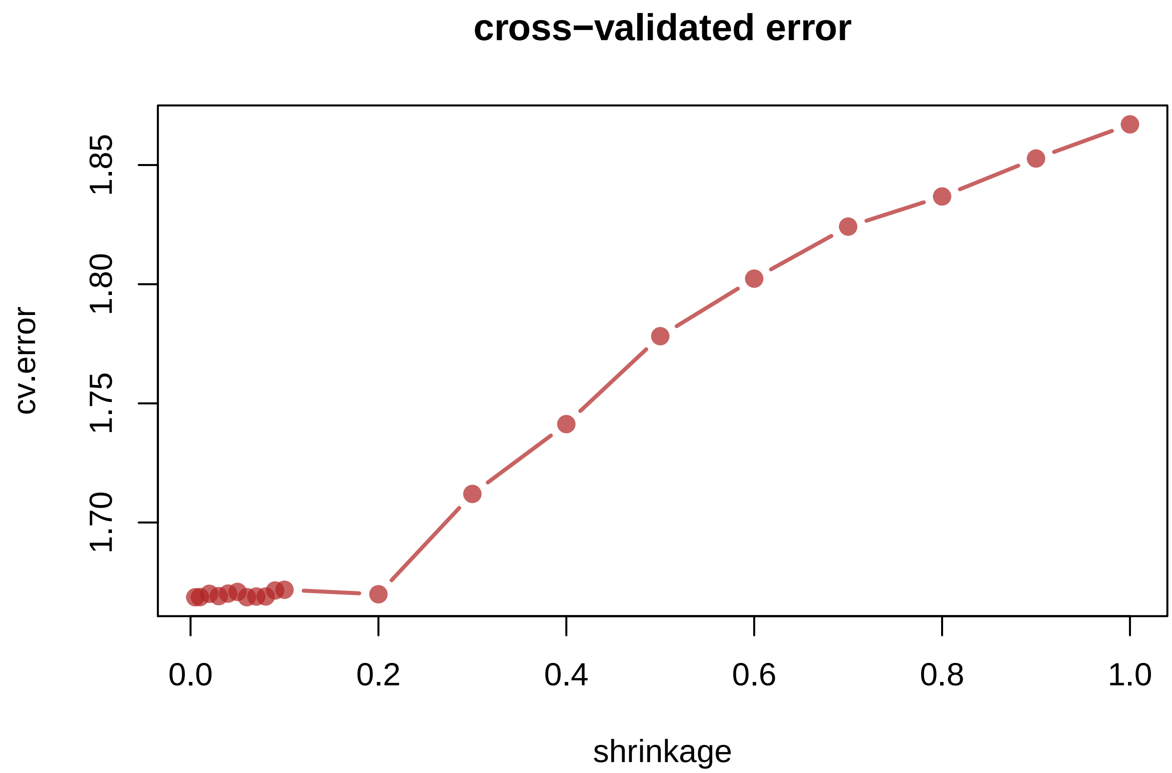
Player self-evaluate and improvements

We want to do the same thing in boosting for what we did in random tree, allowing players to see what they need to improve on the most to get a higher salary. Again, we removed Team, MIN, WR, AGE, Draft.Round, and PLUSMINUS, and repeated the process.

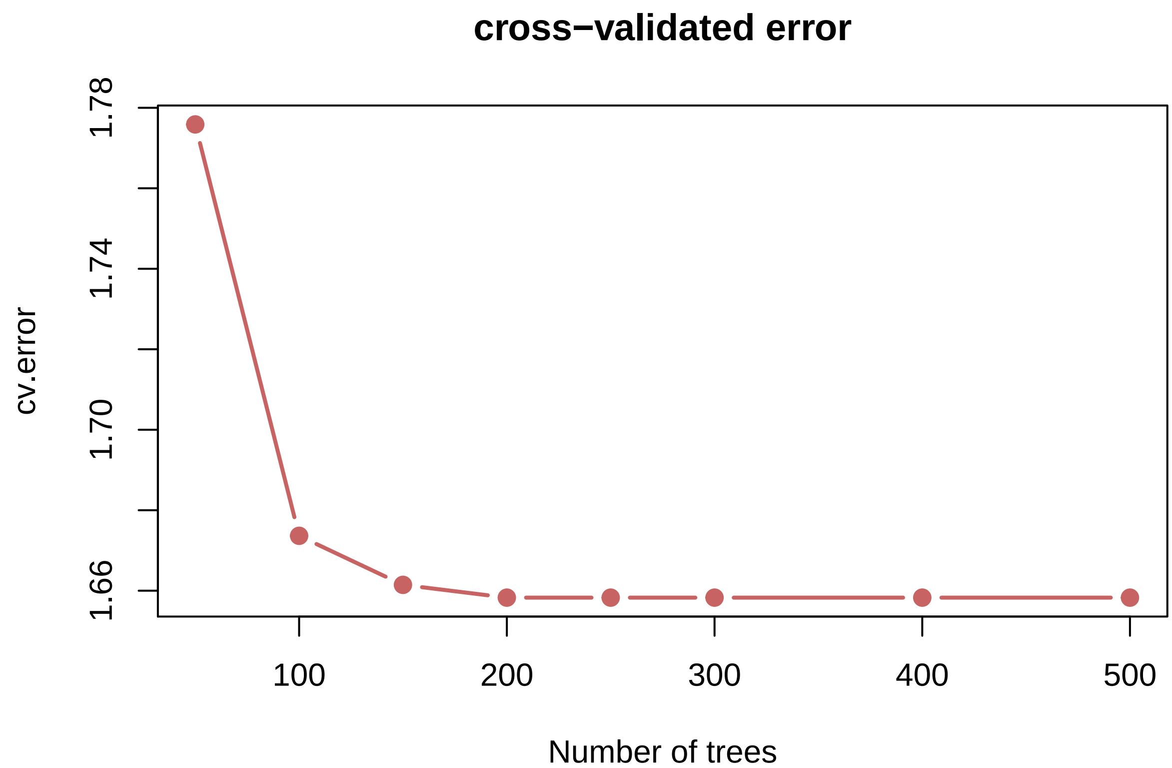



```
##               var    rel.inf
## PTS           PTS 19.2066075
## REB           REB 13.9579523
## TOV           TOV 11.2821537
## FGM           FGM 10.5487276
## FTPER       FTPER  9.8673637
## PF             PF  9.4727863
## FGPER       FGPER  8.8143765
## Country   Country  5.7839050
## Position Position  3.3121806
## TPM           TPM  2.4321169
## TPPER       TPPER  1.8974861
## FTM           FTM  1.5347916
## AST           AST  0.9965880
## STL           STL  0.5951962
## BLK           BLK  0.2977679
```

The result shows that PTS is the most important variable here, REB is the second most important, with TOV, FGM, FTPER, PF, and FGPER at 3rd to 7th, which are very close with each other.



We see that the best learning rate for these 15 explainatory variates is also at around 0.06, with cv error around 1.66.



We see that the best value for M for more explanatory variables is also at about 200, with cv error around 1.66. We can see that the return is very small when M is bigger than 200.