Project Report

Jacob Seedorff

Abstract

We are using a dataset containing many different basketball statistics for all the teams in the NBA from 1980 to 2011. We will attempt to use the data from the previous year to predict how many games a team will win this year. We are also interested in what statistics from the previous season are most useful in predicting wins for this season. We will use many different types of regression to try to predict wins, including lasso, ridge, linear regression, pcr and plsr. The methods we use to predict wins this season will also give us some insight into what statistics from the previous season indicate how good a team is. We find that the linear regression model is good at predicting the number of wins, but we can improve our prediction accuracy by using dimension reduction methods such as partial least squares regression (plsr) and methods to help deal with multicollinearity such as best subset selection.

1.1 Introduction

For this project we have decided to work with the National Basketball Association(NBA) Data set authored by Aman Ajmera. The main goal of this project is to predict how many wins an NBA team will win given their statistics from the previous season. We will use a variety of different methods to try to predict wins and we will test their accuracy by using 10-fold cross-validation. We will use the method deemed the best by 10-fold cross-validation to get an estimate for wins for all teams in the NBA in the 2011 season. The dataset isn't very large originally, we have only 835 observations and 20 predictors. The dataset was originally used by the author to try to understand which statistics from the current year are most indicative of the number of points scored by that team in that season. Since we are trying to predict the number of wins a team gets this season given their statistics from last season, we needed to change the dataset to fit our goal. After modifying the dataset, we now only have 768 observations and only 18 predictors. We have less predictors because we had to get rid of some of the predictors that were perfectly collinear, we also got rid of a few predictors that we felt weren't useful for prediction, and we added additional predictors we thought would be useful. We also have less observations because we had to remove teams that didn't have data from the previous year.

Name	Description	Data Type
Playoffs1	Is 1 if this team made the playoffs last year and 0 otherwise.	Binary
W1	Number of wins by this team last year.	Non-negative float
oppPTS1	Number of points scored against this team last year.	Non-negative float
X2P1	Number of 2 points shots made by this team last year.	Non-negative float
X2PA1	Number of 2 points shots attempted by this team last year.	Non-negative float
X3P1	Number of 3 points shots made by this team last year.	Non-negative float

X3PA1	Number of 3 points shots attempted by this team last year.	Non-negative float
FT1	Number of free throws made by this team last year.	Non-negative float
FTA1	Number of free throws attempted by this team last year.	Non-negative float
ORB1	Number of offensive rebounds by this team last year.	Non-negative float
DRB1	Number of defensive rebounds by this team last year.	Non-negative float
AST1	Number of assists by this team last year.	Non-negative float
STL1	Number of steals by this team last year.	Non-negative float
BLK1	Number of blocks by this team last year.	Non-negative float
TOV1	Number of turnovers by this team last year.	Non-negative float
FG_per1	Field goal percentage by this team last year.	0-100 float
X2P_per1	Two point field goal percentage by this team last year.	0-100 float
X3P_per1	Three point field goal percentage by this team last year.	0-100 float

1.2 Limitations

There is an important limitation that greatly limits the accuracy we have in predicting the win total for a team given only their stats for the previous season. This limitation is that none of these statistics account for players being added to the team and leaving the team in the offseason between the previous year and the current year. This limitation really shows itself when a team either loses some of their best players from the previous season or adds some very talented players to their roster in the offseason. One big example of this is the Cleveland Cavaliers in 2010 had 61 wins and after that season they lost their best player Lebron James, the next season they only managed to win 19 games which is a huge difference from the previous season.

2.1 Feature Exploration

	W	Playoffs1	W1	oppPTS1	X2P1	X2PA1	X3P1	X3PA1	FT1
W	1.00000000	5.396726e-01	0.66882064	-0.20719261	0.07878964	-3.402583e-02	0.05651051	0.032203759	0.156439409
Playoffs1	0.53967257	1.000000e+00	0.79734604	-0.22299241	0.11415946	-3.411725e-05	0.02198522	0.001250445	0.218846867
W1	0.66882064	7.973460e-01	1.00000000	-0.32368404	0.07306206	-7.894089e-02	0.11593078	0.081403153	0.197969225
oppPTS1	-0.20719261	-2.229924e-01	-0.32368404	1.00000000	0.77581011	7.635352e-01	-0.58024284	-0.578744730	0.560730285
X2P1	0.07878964	1.141595e-01	0.07306206	0.77581011	1.00000000	9.663424e-01	-0.88324558	-0.892526883	0.573322041
X2PA1	-0.03402583	-3.411725e-05	-0.07894089	0.76353517	0.96634237	1.000000e+00	-0.92337064	-0.924963468	0.517903027
X3P1	0.05651051	2.198522e-02	0.11593078	-0.58024284	-0.88324558	-9.233706e-01	1.00000000	0.994763364	-0.511861244
X3PA1	0.03220376	1.250445e-03	0.08140315	-0.57874473	-0.89252688	-9.249635e-01	0.99476336	1.000000000	-0.519320206
FT1	0.15643941	2.188469e-01	0.19796922	0.56073028	0.57332204	5.179030e-01	-0.51186124	-0.519320206	1.000000000
FTA1	0.13488645	1.767711e-01	0.15162904	0.55149583	0.57373227	5.256516e-01	-0.53543138	-0.537550591	0.951276005
ORB1	-0.04590375	-2.917582e-02	-0.08488622	0.56631000	0.68803991	7.667388e-01	-0.66164869	-0.645960988	0.395684726
DRB1	0.35026246	3.419086e-01	0.48273674	-0.23110469	-0.10368586	-1.640705e-01	0.22848822	0.218803153	0.000448672
AST1	0.26262594	3.203798e-01	0.31963449	0.54942722	0.77916663	6.822133e-01	-0.57438607	-0.597111503	0.442807737
STL1	0.11552024	1.742540e-01	0.11016318	0.36531320	0.49898884	4.985843e-01	-0.42266876	-0.412673419	0.326308020
BLK1	0.20015482	1.972313e-01	0.21909103	0.02442642	0.22079763	2.072022e-01	-0.23520976	-0.238690921	0.161633885
TOV1	-0.11738929	-1.659166e-01	-0.24045666	0.58359757	0.63430654	6.393500e-01	-0.67886952	-0.675921677	0.435762396
FG_per1	0.34090706	3.645024e-01	0.44022316	0.51004889	0.75588099	5.689613e-01	-0.52799172	-0.567315551	0.532423033
X2P_per1	0.40372291	4.254171e-01	0.53855206	0.33350229	0.47396685	2.338508e-01	-0.17170710	-0.202865152	0.394852937
X3P_per1	0.09972732	5.077949e-02	0.15938196	-0.52689214	-0.71521519	-7.470486e-01	0.78717376	0.758987290	-0.368197714

```
                 FTA1        ORB1         DRB1        AST1       STL1        BLK1       TOV1     FG_per1    X2P_per1    X3P_per1
W          0.13488645 -0.04590375  0.350262457  0.26262594  0.1155202  0.20015482 -0.1173893  0.34090706  0.40372291  0.09972732
Playoffs1  0.17677114 -0.02917582  0.341908633  0.32037980  0.1742540  0.19723132 -0.1659166  0.36450242  0.42541708  0.05077949
W1         0.15162904 -0.08488622  0.482736741  0.31963449  0.1101632  0.21909103 -0.2404567  0.44022316  0.53855206  0.15938196
oppPTS1    0.55149583  0.56631000 -0.231104695  0.54942722  0.3653132  0.02442642  0.5835976  0.51004889  0.33350229 -0.52689214
X2P1       0.57373227  0.68803991 -0.103685857  0.77916663  0.4989888  0.22079763  0.6343065  0.75588099  0.47396685 -0.71521519
X2PA1      0.52565164  0.76673876 -0.164070511  0.68221332  0.4985843  0.20720221  0.6393500  0.56896129  0.23385083 -0.74704864
X3P1      -0.53543138 -0.66164869  0.228488222 -0.57438607 -0.4226688 -0.23520976 -0.6788695 -0.52799172 -0.17170710  0.78717376
X3PA1     -0.53755059 -0.64596099  0.218803153 -0.59711150 -0.4126734 -0.23869092 -0.6759217 -0.56731555 -0.20286515  0.75898729
FT1        0.95127601  0.39568473  0.000448672  0.44280774  0.3263080  0.16163388  0.4357624  0.53242303  0.39485294 -0.36819771
FTA1       1.00000000  0.47419109 -0.046070952  0.42210020  0.3736586  0.21030096  0.5244512  0.51308285  0.37405748 -0.41464321
ORB1       0.47419109  1.00000000 -0.254712493  0.41439073  0.4938598  0.19635514  0.5377621  0.24257990 -0.01286809 -0.55971368
DRB1      -0.04607095 -0.25471249  1.000000000  0.05866228 -0.3111556  0.25599417 -0.1932622  0.06145949  0.16467759  0.17612823
AST1       0.42210020  0.41439073  0.058662283  1.00000000  0.4516361  0.21515978  0.4277412  0.76329808  0.60776946 -0.40757168
STL1       0.37365861  0.49385977 -0.311155561  0.45163610  1.0000000  0.11637618  0.4738010  0.30722043  0.18093310 -0.38780548
BLK1       0.21030096  0.19635514  0.255994168  0.21515978  0.1163762  1.00000000  0.2411646  0.19839227  0.12539353 -0.18970618
TOV1       0.52445125  0.53776213 -0.193262228  0.42774116  0.4738010  0.24116462  1.0000000  0.42861935  0.20242194 -0.64353844
FG_per1    0.51308285  0.24257990  0.061459489  0.76329808  0.3072204  0.19839227  0.4286193  1.00000000  0.90989219 -0.37940419
X2P_per1   0.37405748 -0.01286809  0.164677593  0.60776946  0.1809331  0.12539353  0.2024219  0.90989219  1.00000000 -0.13739659
X3P_per1  -0.41464321 -0.55971368  0.176128231 -0.40757168 -0.3878055 -0.18970618 -0.6435384 -0.37940419 -0.13739659  1.00000000
```

We start by splitting our data into a training set which includes all teams from all seasons from 1981 to 2010, and a test set which includes all teams from the 2011 season. We will use the training set to find the best model and we will perform a final prediction for the 2011 season with the model that performs the best. We will try to estimate MSE_{test} by using 10-fold cross-validation and we will choose our final model by choosing the model that has the lowest estimated MSE_{test}. We will then use that model to try to predict the win total for all teams in the 2011 season.

Looking at the correlations between all the predictors and wins we see that many of the predictors are correlated with wins and many predictors are also correlated with other predictors. Which indicates we probably have quite a bit of multicollinearity in our predictors. We see that wins from last season and if the team made the playoffs last season are both heavily correlated with wins this season. We can also see that attempts and makes are heavily correlated for 2 pointers, 3 pointers, and free throws. A naïve way to predict the wins for this season is to just guess they will win the same number of games that they won last year. Using this method, we get an estimated MSE_{test} = 103.555, we hope that the regression methods we use will be significantly better than this.

2.2.1 Linear Regression with all predictors

In Linear regression we assume there is a linear relationship between the predictors and the response, this assumption is almost never met. Even if this assumption isn't met linear regression is a good way to explore relationships in the data. The model is of this form:

$$\boldsymbol{Wins = \beta_0 + \beta_1 * Playoffs1 + \beta_2 * W1 + \cdots + \beta_{18} * X3P\_per1 + \varepsilon}$$

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.914e+02  1.627e+02   1.791   0.0737 .  
Playoffs1    3.867e-01  1.160e+00   0.333   0.7389    
W1           1.226e-02  1.170e-01   0.105   0.9166    
oppPTS1     -1.850e-02  4.083e-03  -4.531 6.86e-06 ***
X2P1         1.263e-01  4.983e-02   2.534   0.0115 *  
X2PA1       -4.412e-02  2.359e-02  -1.871   0.0618 .  
X3P1         1.595e-01  7.409e-02   2.153   0.0317 *  
X3PA1       -4.909e-02  3.477e-02  -1.412   0.1584    
FT1          1.101e-02  7.368e-03   1.495   0.1354    
FTA1         6.561e-03  5.476e-03   1.198   0.2313    
ORB1         3.607e-03  5.638e-03   0.640   0.5225    
DRB1         6.582e-03  4.587e-03   1.435   0.1517    
AST1         3.128e-03  2.959e-03   1.057   0.2908    
STL1         7.169e-03  6.247e-03   1.148   0.2516    
BLK1         6.879e-03  4.563e-03   1.507   0.1321    
TOV1        -9.332e-04  4.679e-03  -0.199   0.8420    
FG_per1     -5.991e+00  4.883e+00  -1.227   0.2202    
X2P_per1     1.808e-01  2.183e+00   0.083   0.9340    
X3P_per1     6.621e-02  1.225e-01   0.540   0.5891    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.106 on 719 degrees of freedom
Multiple R-squared:  0.4897,	Adjusted R-squared:  0.4769 
F-statistic: 38.33 on 18 and 719 DF,  p-value: < 2.2e-16
```

Here we create a linear regression model with all of our predictors included in it and our response is wins for this season. From the summary output we can see that oppPTS1, X2P1, and X3P1 are all significant predictors for wins this season and X3PA1 is a marginally significant predictor for wins this season. We can also see that most of the predictors are not significant predictors for wins, which suggests some kind of variable selection may be beneficial. We used 10-fold cross-validation to get an estimate of MSE_{test} and we get an estimated MSE_{test} = 84.8637 for this model which is already quite a bit better than the naïve method of just predicting how many wins the team got the previous season. We also see that the R^2 = .4897 which is pretty good. One of the betas that surprised us is FG_per1 because this should definitely be positive because as you make a larger percentage of your shots you should definitely win more. Incorrect signs on your betas can be caused by large amounts of multicollinearity among your predictors. Variance inflation factor(VIF) is a measure of multicollinearity for a linear model, as VIF increases there is more multicollinearity with that predictor and the other predictors. From the VIF values below we can see that there is a large amount of multicollinearity in the model, so we may want to try some methods that can handle multicollinearity.

```
 Playoffs1          W1     oppPTS1        X2P1       X2PA1        X3P1       X3PA1         FT1        FTA1        ORB1        DRB1
  2.910853   19.291583   52.358601 4551.787443 3464.395578 1918.231107 2904.254001   19.070094   15.894879    6.186282    3.091500
      AST1        STL1        BLK1        TOV1      FG_per1    X2P_per1    X3P_per1
  3.878047    3.001253    1.309150    4.688884 1042.764219  170.310903    3.800326
```

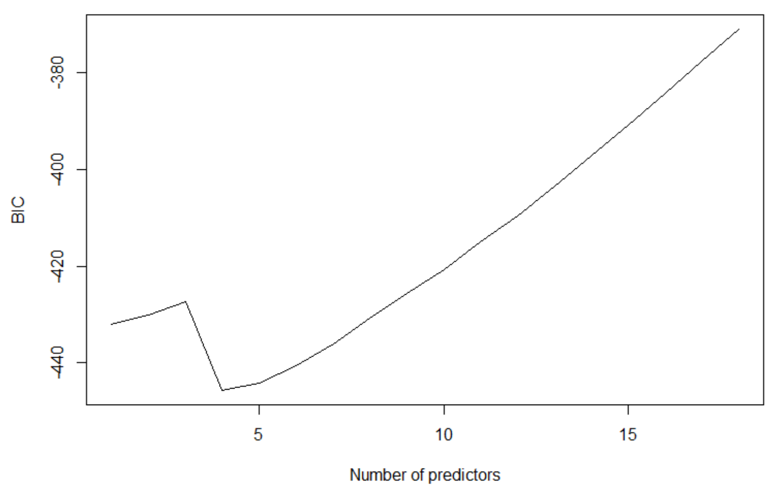
2.2.2 Linear Regression with BIC best subset selection

$$Wins = \beta_0 + \beta_1 * oppPTS1 + \beta_2 * X2P1 + \beta_3 * X3P1 + \beta_4 * FT1 + \varepsilon$$

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 37.966923   6.655252   5.705 1.69e-08 ***
oppPTS1     -0.022923   0.001000 -22.920  < 2e-16 ***
X2P1         0.046750   0.002188  21.362  < 2e-16 ***
X3P1         0.069414   0.003889  17.849  < 2e-16 ***
FT1          0.023176   0.002130  10.880  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.128 on 733 degrees of freedom
Multiple R-squared:  0.4772,	Adjusted R-squared:  0.4744 
F-statistic: 167.3 on 4 and 733 DF,  p-value: < 2.2e-16
```

Best subset selection is a method that does variable selection and can also lead to a decrease in multicollinearity. Best subset selection is done by fitting a linear regression model for every possible combination of predictors and choosing the model that gives the best estimated MSE_{test}. There are many values that are used to approximate MSE_{test}, but we chose to use BIC because it adds a higher penalty for more predictors, and we want to get rid of many of the predictors that aren’t useful. From this graph of the BIC we can see that the number of predictors that minimizes BIC is 4, so we fit a linear regression model with the 4 predictors chosen by best subset selection. An interesting observation is that all of the variables that were found statistically significant in the full model were included in this model.We get an estimated MSE_{test} = 83.7267, which is slightly better than the estimated MSE_{test} was for linear regression with all the predictors. We can also see that the R^2 = .4772 which is about as good as the R^2 was for linear regression with all the predictors, but we are only including 4 of the 18 predictors which is very good.



From the VIF values below we can see that the VIF values for our predictors are much lower now, so we have gotten rid of pretty much all the multicollinearity we had in the full model. We will now look at performing shrinkage and dimension reduction methods to try to deal with multicollinearity and perform variable selection.

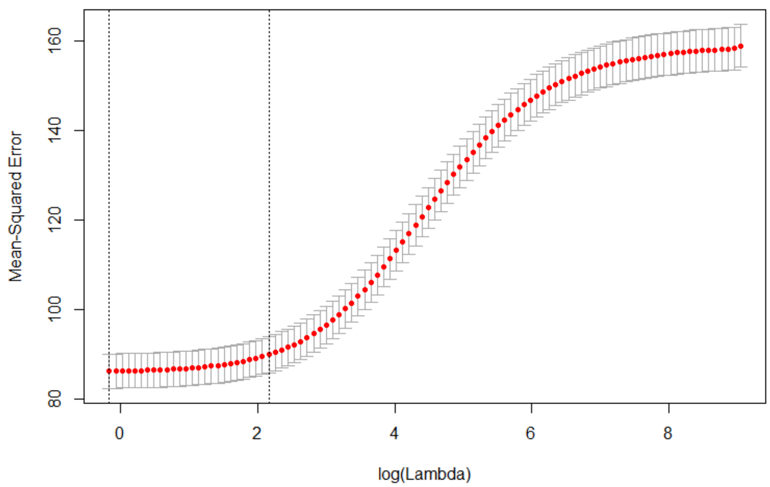
```
 oppPTS1     X2P1     X3P1      FT1 
3.125677 8.737389 5.260230 1.586156 
```

2.3 Ridge Regression

In Ridge regression we find the betas by solving this equation

$$\underset{\beta}{argmin}\ (\frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\left\|\beta\right\|^2)$$

```
  (Intercept)     Playoffs1            W1       oppPTS1          X2P1         X2PA1          X3P1         X3PA1           FT1 
-4.940938e+01  1.155923e+00  3.115008e-01 -5.216043e-03  1.314016e-03  1.039937e-04  4.654810e-03  5.556191e-04  1.856516e-03 
         FTA1          ORB1          DRB1          AST1          STL1          BLK1          TOV1       FG_per1      X2P_per1 
 2.969019e-03  5.197491e-03  1.007357e-02  3.763596e-03  1.051181e-02  5.699418e-03 -2.860801e-03  7.836017e-01  4.649202e-01 
     X3P_per1 
 9.565397e-02 
```



We are using this modeling strategy because we believe that it will be able to deal with some of the multicollinearity that exists between the predictors. This model is fit by applying a L2 penalty (λ) to the beta vector. No matter how much penalty we add none of the betas will be predicted as zero, only very close to zero, so this method will not help us with variable selection. We choose the λ by selecting the λ that minimizes the MSE_{test} as estimated by 10-fold cross-validation. We end up choosing λ = 0.8468328 which is fairly large, so we should get much different estimates from linear regression. We can see that many of the betas are quite different from what we got in linear regression. Unfortunately, our estimated MSE_{test} = 86.24637, which is worse than what we got from linear regression. We will continue with a method that will perform variable selection.

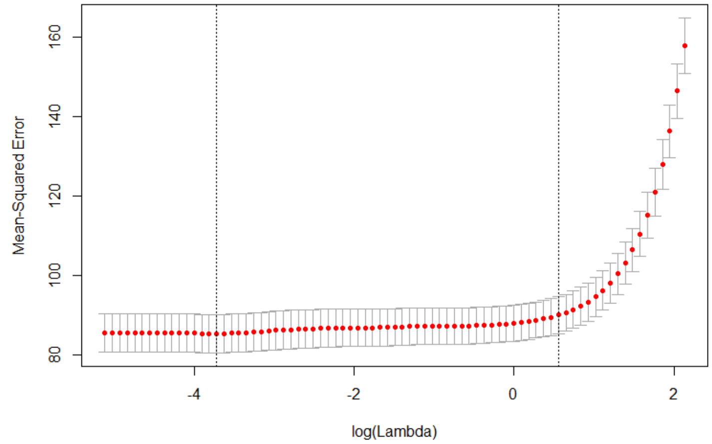
2.4 Lasso Regression

In Lasso regression we find the betas by solving this equation

$$\underset{\beta}{argmin}\ (\frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda|\beta|)$$

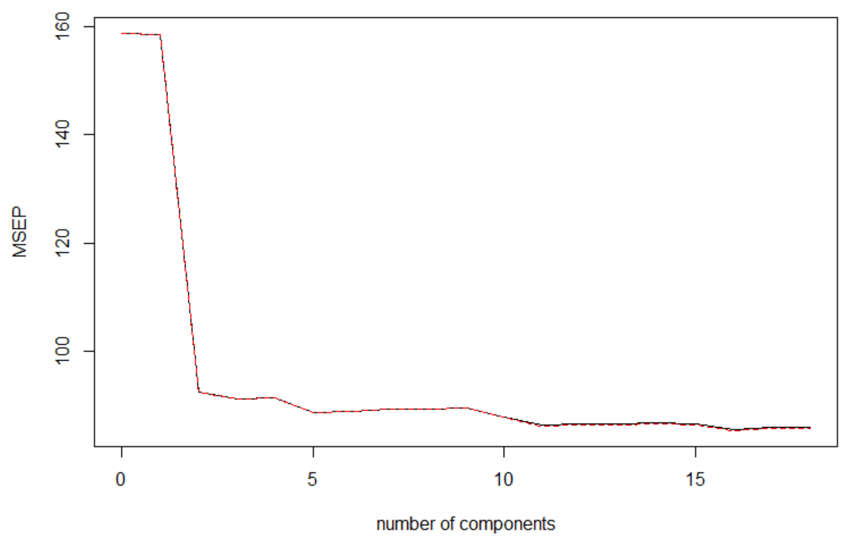
```
  (Intercept)     Playoffs1            W1       oppPTS1          X2P1         X2PA1          X3P1         X3PA1           FT1 
-8.5586429578  0.2885298786  0.0816801807 -0.0170898949  0.0295716993  0.0000000000  0.0482436903  0.0000000000  0.0099731628 
         FTA1          ORB1          DRB1          AST1          STL1          BLK1          TOV1       FG_per1      X2P_per1 
 0.0060463503  0.0012427244  0.0057232258  0.0031594561  0.0057882297  0.0060859490  0.0005238088  0.6647507835 -0.0415588255 
     X3P_per1 
 0.0524976010 
```

Lasso regression is a method that will help us get rid of some of the useless predictors. Lasso regression is fit very similarly to ridge regression except the penalty is a L1 penalty instead of a L2 penalty. Another difference between ridge and lasso is that lasso will actually say that some of the betas are zero while ridge won't, this is how lasso helps us do variable selection. We again choose λ by selecting the λ that minimizes the MSE_{test} as estimated by 10-fold cross-validation. We end up choosing λ = 0.01145805. We see that lasso estimates 2 of the betas as being 0, this is promising because if we are using lasso, we hope that it predicts some of the betas to be zero. We see that lasso estimates far fewer betas as zero than we got using best subset selection, but all of the betas that were zero in the lasso model were also removed by the best subset selection model. Unfortunately, we get an estimated MSE_{test} = 85.25006, which is worse than linear regression with best subset selection, but it is slightly better than ridge regression. We will now look at some non-parametric methods to see if we can improve performance since lasso and ridge haven't helped us predict better than linear regression with best subset selection.



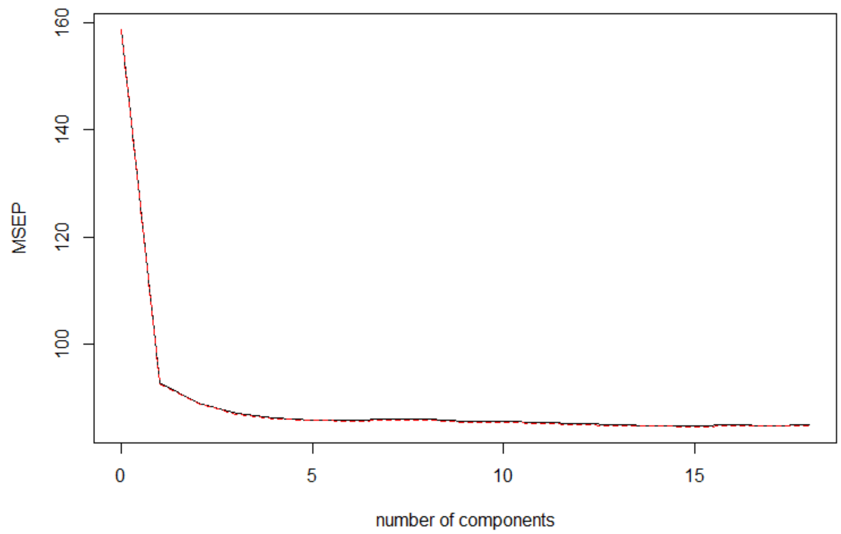
2.5 Principal Components Regression

Principal components regression is another technique that can help with multicollinearity. Principal components regression is fit by creating new predictors where these new predictors are a linear combination of our old predictors and then fitting a linear regression model using these new predictors. We choose these new predictors by creating new predictors which are linear combinations of the old predictors that contain the most amount of variance. From the graph we choose five components because the change in MSE is very small after we get past five components. The estimated MSE_{test} = 91.4907, which is much worse than the other methods we have tried. Principal components regression doesn't take the values of the response into account when creating the principal components, so maybe a method that creates



the new predictors and takes the values of the response into account will perform better. We don't show the beta estimates for the model because they don't have the same interpretability of the other models because the predictors we use in the linear regression model aren't our original predictors, but linear combinations of them.

2.6 Partial Least Squares Regression



Partial least squares regression is similar to Principal components regression, but it takes the values of the response into account when it is creating the new predictors. Partial least squares regression is better at reducing the dimension of the space of your predictors, but it isn't as good at dealing with multicollinearity as principal components regression. We choose to use five components since the change in MSE after adding additional components is very small. The estimated MSE_{test} = 83.44801 when using five components, this is our best estimated MSE_{test} so we will use it to perform the final prediction. We don't show the beta estimates for the same reason we didn't show them in the principal components regression model.

3.1 Useful Predictors

We performed a few methods that helped us find out which of the predictors were most useful in guessing how good a team is and will most help us predict their number of wins in the next season. These methods were best subset selection and lasso regression. Lasso was more conservative and had only 4 of the betas as being zero while best subset selection only chose four of the betas as being non-zero. The betas that were zero in the lasso model were also zero in the best subset model, so they agree that those four predictors aren't useful. The best subset selection model chose to include all three predictors that tell us how many points are scored by the team in that season and also oppPTS1. Interestingly enough the beta for X3P1 is almost exactly 3 times larger than the beta for FT1 which makes sense because a three point shot is worth 3 times as much as a free throw and the same trend holds true for the beta for X2P1. We can see that this model will be almost exactly the same as a model with just oppPTS1 and points scored by this team last season where the beta for points scored last season will be the almost exactly same as the beta for FT1. We fit a linear regression model with just oppPTS1 and points and we saw that the beta for points was almost exactly the same as the beta for FT1 and we also see the R^2 is almost exactly the same for the two models. We can also note that the betas for pts and oppPTS1 are almost the same except the signs are flipped, so this will be almost exactly the same as the model that contains only point differential in the model, which is PTS – oppPTS1. We see this is also true because the beta for point differential is very close to the beta for PTS and our R^2 decreases only slightly. So, we conclude that the predictor that is most indicative of the skill level of a team is point differential.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 36.5808104  5.0385833    7.26 9.86e-13 ***
oppPTS1     -0.0228100  0.0009371  -24.34  < 2e-16 ***
PTS          0.0233623  0.0009438   24.75  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.116 on 735 degrees of freedom
Multiple R-squared:  0.4771,    Adjusted R-squared:  0.4757 
F-statistic: 335.4 on 2 and 735 DF,  p-value: < 2.2e-16
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.122e+01  3.355e-01  122.85   <2e-16 ***
PTdiff      2.308e-02  8.915e-04   25.88   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.115 on 736 degrees of freedom
Multiple R-squared:  0.4765,    Adjusted R-squared:  0.4758 
F-statistic:   670 on 1 and 736 DF,  p-value: < 2.2e-16
```

3.2 Final Prediction

We now have our best model and we will use it to predict how many wins each NBA team got in 2011 based on their statistics from 2010. We will use our Partial least squares regression model to predict this because it had the lowest estimated MSE_{test}. While Partial least squares regression performed the best of any of the models, it wasn't significantly better than all of the other models except for principal components regression. Our MSE_{test} = 112.86632, this seems quite a bit worse than we expected because our estimated MSE_{test} was about 30 less than this. This is probably because this test set includes that 2011 Cleveland Cavaliers team, I talked about earlier that lost Lebron James and went from 61 wins in 2010 to 19 wins in 2011. If we remove this outlier our MSE_{test} = 76.469, which is way better and is very good. So, our model is good at predicting the number of games a team will win next season given they haven't made any big changes from last season that will greatly change impact how good they are.