IS 804 Advanced Quantitative Methods in IS Research (Spring 2022)

Project Title:

Electrical Power prediction of a wind Turbine using Statistical learning Approach

-presented by

Md Azim Khan - DM55018

submitted to

Prof. Dr. Shimei Pan

Department of Information Systems
University of Maryland Baltimore County Baltimore, USA

2022-05-23

Table of contents

Chapter-1 Introduction

- Abstract
- Dataset Description
- Loading dataset
- Data Statistics
- Pair plot
- Distribution of response variable

Abstract

Most Electrical Power stations are Fossil fuel based that emit gigaton carbon dioxide in the air.For a Sustainable world ,the use of fossil fuels to produce electricity needs to change. Renewable energy is the only alternative to fossil fuels but it’s not reliable, affordable and desirable because of its unpredictable nature. If we are able to predict the nature of Renewable energy like wind energy using statistical learning algorithms, it can bring a revolution in the production of electricity. There are many factors involved in wind energy production like wind direction, speed, air temperature, and air density that need to be considered to have the highest electric power from a wind turbine. Statistical learning can help to fix those criteria to harness maximum power from a wind turbine. The main **objective** of this project is to use statistical learning algorithms to predict electrical power of a wind turbine.

Turbine Dataset Description

The dataset for this research work is obtained from the ENGIE OpenData Website[*].The wind- farm consists of 4 wind turbines namely R80711, R80736, R80790 and R80721and contain data from 2013 to 2020 for each turbine.I read this site [2] to select the most important features that are related to power production of a wind turbine. Finally, My turbine dataset contains 12000 observations with 13 features.

- Number of features - 13

- Size (12000, 13)

Input variable/ features

1. Ws_avg: Average wind speed (m/s)
2. Ws_min: minimum wind speed (m/s)
3. Ws_max: maximum wind speed (m/s)
4. Ws_std: standard deviation of wind speed
5. Wa_avg: average wind direction in degree
6. Wa_min: minimum wind direction
7. Wa_max: maximum wind direction
8. Wa_std: standard deviation value of wind direction
9. Ot_avg: average outdoor temperature
10. Ot_min: minimum outdoor temperature
11. Ot_max: maximum outdoor temperature
12. Ot-std: standard deviation of outdoor temperature

Output/ target variable

13. P_avg: average power generated by the wind turbine

Loading dataset

I read the data from a csv file and attach it to the project. The R function attach() is used to attach data to the project for ease of accessibility. At first , I check the names of the features, few observations, dimension and mission value of the turbine dataset.

```
turbine <-
read.csv("/Users/azimkhan22/Documents/r_doc/final_code/turbine_std.csv",
na.strings = "?", stringsAsFactors = T)
attach(turbine) # dataset name is turbine
```

Below procedure displays the data’s features and observations for better understanding.

```
names(turbine) # displays features names
```

```
##  [1] "Ws_avg" "Ws_min" "Ws_max" "Ws_std" "Wa_avg" "Wa_min" "Wa_max" "Wa_std"
##  [9] "P_avg"  "Ot_avg" "Ot_min" "Ot_max" "Ot_std"
```

```
head(turbine) # displays 10 observations
```

```
##   Ws_avg Ws_min Ws_max Ws_std Wa_avg Wa_min Wa_max    Wa_std   P_avg Ot_avg
## 1   7.25   5.18  10.39   1.01 163.16 127.80 196.29 10.530000  679.69   5.72
## 2   7.64   5.46  12.12   1.04 286.19 236.71 317.75 10.460000  801.23   5.44
## 3   8.96   6.48  11.26   0.81 165.25 126.95 186.97  7.310000 1072.65   5.09
## 4   8.56   6.44  10.41   0.79 175.05 150.94 215.13  8.229999 1091.82   5.48
## 5   8.89   5.78  11.65   1.01 168.29 137.85 189.89  7.560000 1061.43   5.26
## 6   8.91   6.50  11.18   0.94 178.00 146.44 210.35  8.210000 1113.45   5.69
##   Ot_min Ot_max Ot_std
## 1    5.6   5.90   0.06
## 2    5.3   5.80   0.12
## 3    5.0   5.20   0.09
## 4    5.3   5.75   0.09
## 5    5.0   5.50   0.11
## 6    5.5   5.80   0.12
```

```
dim(turbine)
```

```
## [1] 12000    13
```

The dim()function shows the data has 12000 observations and 13 columns.

```
sum(is.na(turbine)) # no missing value
```

```
## [1] 0
```

Data statistics

The summary() function in R shows a numerical summary of each variable in the turbine data set. We display the summary statistics of the turbine dataset.

```
summary(turbine)
```

```
##      Ws_avg          Ws_min          Ws_max           Ws_std      
##  Min.   : 0.00   Min.   :0.000   Min.   : 0.000   Min.   :0.0000  
##  1st Qu.: 2.49   1st Qu.:1.330   1st Qu.: 3.540   1st Qu.:0.4100  
##  Median : 4.30   Median :3.050   Median : 5.800   Median :0.5400  
##  Mean   : 4.12   Mean   :2.685   Mean   : 5.714   Mean   :0.5778  
##  3rd Qu.: 5.62   3rd Qu.:4.010   3rd Qu.: 7.503   3rd Qu.:0.7000  
##  Max.   :11.87   Max.   :8.150   Max.   :17.710   Max.   :2.4500  
##      Wa_avg           Wa_min          Wa_max           Wa_std        
##  Min.   :  0.02   Min.   :  0.0   Min.   :  0.04   Min.   :   1.64  
##  1st Qu.:113.37   1st Qu.:128.5   1st Qu.: 88.39   1st Qu.:   9.85  
##  Median :209.41   Median :188.8   Median :214.93   Median :  12.05  
##  Mean   :196.10   Mean   :190.8   Mean   :189.68   Mean   :  20.93  
##  3rd Qu.:272.48   3rd Qu.:270.4   3rd Qu.:281.03   3rd Qu.:  14.79  
##  Max.   :359.99   Max.   :360.0   Max.   :359.97   Max.   :1104.79  
##      P_avg             Ot_avg           Ot_min           Ot_max      
##  Min.   : -14.01   Min.   :-6.180   Min.   :-6.300   Min.   :-6.000  
##  1st Qu.:  -0.88   1st Qu.:-1.680   1st Qu.:-1.800   1st Qu.:-1.600  
##  Median :  59.52   Median : 1.680   Median : 1.575   Median : 1.800  
##  Mean   : 166.30   Mean   : 1.499   Mean   : 1.386   Mean   : 1.613  
##  3rd Qu.: 244.61   3rd Qu.: 4.850   3rd Qu.: 4.700   3rd Qu.: 5.000  
##  Max.   :1860.91   Max.   : 8.290   Max.   : 8.200   Max.   : 8.400  
##      Ot_std       
##  Min.   :0.00000  
##  1st Qu.:0.03000  
##  Median :0.05000  
##  Mean   :0.05776  
##  3rd Qu.:0.08000  
##  Max.   :0.98000  
```

Next, I examine the correlation state of our data. I display correlation with cor() R function.

```
cor(turbine)
```

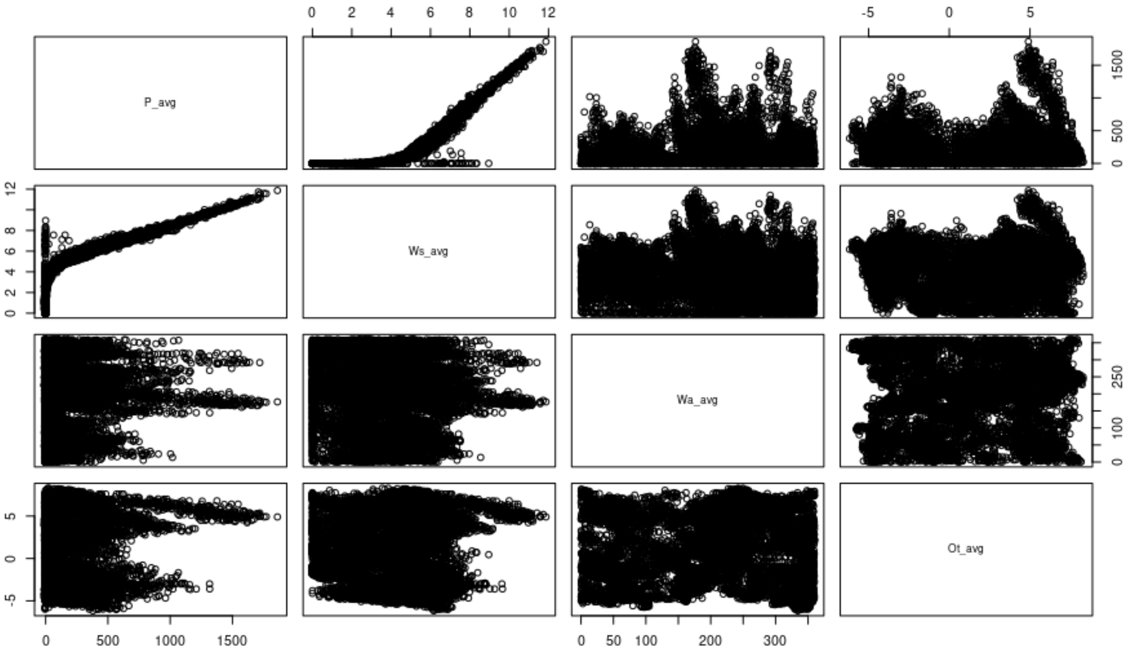
```
##              Ws_avg      Ws_min       Ws_max        Ws_std       Wa_avg
## Ws_avg  1.000000000  0.95039830  0.978354260  0.5434819233 -0.008277598
## Ws_min  0.950398297  1.00000000  0.894648804  0.3103932800 -0.038369318
## Ws_max  0.978354260  0.89464880  1.000000000  0.6742131855  0.007268745
## Ws_std  0.543481923  0.31039328  0.674213186  1.0000000000  0.079235299
## Wa_avg -0.008277598 -0.03836932  0.007268745  0.0792352994  1.000000000
## Wa_min -0.007285600 -0.02004878 -0.004501731  0.0193014350  0.283966516
## Wa_max  0.121353031  0.10086182  0.131568744  0.1209572635  0.260585799
## Wa_std -0.228146478 -0.24474134 -0.190380042  0.0004079327  0.046863053
## P_avg   0.828895493  0.75384446  0.842842061  0.5928181079  0.038656689
## Ot_avg  0.066369225  0.03869983  0.076037776  0.1094559441  0.137821379
## Ot_min  0.068755653  0.04112404  0.078145683  0.1096628983  0.138467319
## Ot_max  0.064014411  0.03630642  0.073836435  0.1088730038  0.136702882
## Ot_std -0.122481080 -0.12532334 -0.111768895 -0.0218778910 -0.031808867
##              Wa_min      Wa_max        Wa_std       P_avg       Ot_avg
## Ws_avg -0.007285600  0.12135303 -0.2281464780  0.82889549  0.066369225
## Ws_min -0.020048778  0.10086182 -0.2447413406  0.75384446  0.038699832
## Ws_max -0.004501731  0.13156874 -0.1903800420  0.84284206  0.076037776
## Ws_std  0.019301435  0.12095726  0.0004079327  0.59281811  0.109455944
## Wa_avg  0.283966516  0.26058580  0.0468630531  0.03865669  0.137821379
## Wa_min  1.000000000 -0.16077421 -0.0146476483 -0.03650932  0.065600228
## Wa_max -0.160774213  1.00000000 -0.0339592436  0.13739103  0.271734615
## Wa_std -0.014647648 -0.03395924  1.0000000000 -0.11734688 -0.009653563
## P_avg  -0.036509320  0.13739103 -0.1173468846  1.00000000  0.075833180
## Ot_avg  0.065600228  0.27173462 -0.0096535625  0.07583318  1.000000000
## Ot_min  0.066292633  0.27334324 -0.0114156173  0.07672250  0.999667741
## Ot_max  0.064414517  0.27005074 -0.0079703679  0.07466679  0.999599759
## Ot_std -0.031539543 -0.06615145  0.0747325028 -0.06049997  0.009711649
##             Ot_min       Ot_max       Ot_std
## Ws_avg  0.06875565  0.064014411 -0.122481080
## Ws_min  0.04112404  0.036306421 -0.125323342
## Ws_max  0.07814568  0.073836435 -0.111768895
## Ws_std  0.10966290  0.108873004 -0.021877891
## Wa_avg  0.13846732  0.136702882 -0.031808867
## Wa_min  0.06629263  0.064414517 -0.031539543
## Wa_max  0.27334324  0.270050737 -0.066151451
## Wa_std -0.01141562 -0.007970368  0.074732503
## P_avg   0.07672250  0.074666785 -0.060499971
## Ot_avg  0.99966774  0.999599759  0.009711649
## Ot_min  1.00000000  0.998884572 -0.010090223
## Ot_max  0.99888457  1.000000000  0.032000898
## Ot_std -0.01009022  0.032000898  1.000000000
```

From the above analysis, wind power is highly correlated with the wind speed (0.8288) , whereas medium correlation can be seen with wind direction and wind direction. Also, I use a graphical plot to show the correlation state of the data set.

Pair plot

To illustrate the effect of correlation further, we have plotted the scatterplots between all the pairs of attributes and power output.First, I use the pair() function to create a scatterplot matrix for all the variables. I see a scatterplot for every pair of variables in the turbine set.

```
pairs(turbine)
```

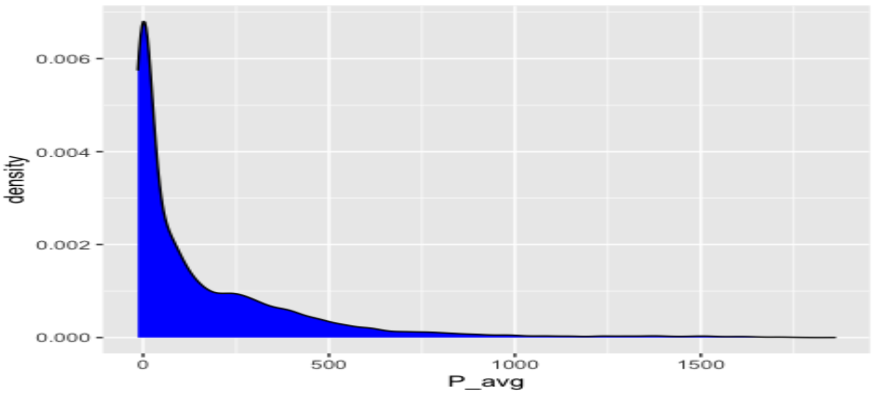


From the pair plot above, I observe some non-linear relationships in these plots and they appear to differ according to wind conditions. Overall, the relationship between predictors and response are not linear.

Distribution of response variables

I used ggplot for exploring the distribution of target variable (P_avg)

```
library(ggplot2) # Data visualization
ggplot(turbine,aes(P_avg)) + geom_density(fill='blue')
```



The Above figure shows the distribution of the response variable (P_avg) closer to normal.

Chapter 2- Research Question

I am going to answer the following research questions in this studies-

1. Is there a relationship between the wind speed (predictor) and the power (response)?
2. How strong is the relationship?
3. Is the relationship between the predictor (wind speed) and the response (power) positive or negative?
4. What is the predicted power associated with a wind speed of 12? What are the associated 95% confidence and prediction intervals?
5. Is there any correlation between features (wind speed, rotor speed, winddirection, outdoor temperature)?
6. Which predictors appear to have a statistically significant relationship to the response?
7. Is this model fit?
8. Is there evidence of non-linear association between any of the predictors and the response?
9. What is the condition of model fit,if some non-significant variables are dropped from the model?
10. Do any interactions appear to be statistically significant?
11. Is there any change after adding Interactions terms to the model?
12. Can we reduce the test error using the validation set approach, LOOCV, k-fold cross validation?
13. What is the value of k (k-fold) for minimum cross validation error?
14. Which are the most important and the least significant variables (features) in best subset methods?
15. Using adjusted R2, what is the number of variables that best subset methods choose?
16. Using Cp, what is the number of variables that best subset methods choose?
17. Using BIC, what is the number of variables that best subset methods choose?
18. What are the 7 variables that best subset methods choose?
19. Which are the most important and the least significant variables (features) in forward stepwise selection methods?
20. Which are the most important and the least significant variables (features) in backward stepwise selection methods?
21. Do all methods (best, forward, backward) select the same features (say first 7 features)?
22. What is the lowest cross validation error for among all models?
23. Why do I select the best 7 features?
24. What does coefficient path display in ridge and lasso regression?

25. how do you choose the tuning parameter and what is the value of tunning parameter in ridge regression?

26. how many components should i select for modeling stage in PCR and PLS?

27. Is polynomial regression of wind speed with all degrees statistically significant?

28. how do one decide the degree of freedom in polynomial regression?

29. Does model with 4 degree show lower test RMSE than model with 2 degrees of freedom?

30. Are all knots statistically significant in spline analysis?

31. How do you select tunning parameter and what is the value of it in smoothing spline model?

32. Does GAM model with natural spline show statistically significant for higher degree of freedom?

33. which is beter, GAM with smoothing spline or GAM with natural spline?

34. What is the tree size in regression tree?

35. how do we know how far back to prune the tree?

36. how many predictors are used in bagging?

37. Is test rmse in random forest lower than the result from bagging?

38. how do we decide which variables are most useful in predicting the response?

39. What is the minimum threshold value of wind speed to produce electrical power?

40. what is the effect of outdoor temperature on producing wind turbine power?

41. Does Boosting model performance improve after tuning the shrinkage parameter?

42. Which kernel function i used in the turbine dataset?

43. how do i select the regularization parameter or budget controls parameter?

44. what is the value of k in knn regression model?

45. Which model is the best for above analysis?

Chapter 3- Linear regression Analyses

- Why linear analysis (not Logistic regression, LDA, QDA)
- Single variable Linear regression
- Multivariate linear regression
- Diagnostic Plots (Fitted vs Residual, Q-Q Plot,Residuals vs Leverage)

- Interactions term effect

- Research Question-

Single variable Linear regression

1. Is there a relationship between the wind speed (predictor) and the power (response)?
2. How strong is the relationship?
3. Is the relationship between the predictor (wind speed) and the response (power) positive or negative?
4. What is the predicted power associated with a wind speed of 12? What are the associated 95% confidence and prediction intervals?

Multivariate Linear regression

5. Is there a relationship between all predictors and the response?
6. Which predictors appear to have a statistically significant relationship to the response?
7. Is this model fit?
8. Is there evidence of non-linear association between any of the predictors and the response?
9. What is the Test RMSE value for the Multivariate Linear model?

Interactions term effect

10. Do any interactions appear to be statistically significant?
11. Is there any change after adding Interactions terms to the model?

Why Linear Regression

Linear Regression is a supervised modeling technique for continuous data. The model fits a line that is closest to all observations in the dataset. The basic assumption here is that functional form is the line and it is possible to fit the line that will be closest to all observation in the dataset. Logistic Regression is another supervised Machine Learning algorithm that helps fundamentally in separating discreet values. As my response variable is continues, i will apply all regression analysis throughout the study.Moreover, Regression analysis allows to understand the strength of relationships between variables,to know what predictors in a model are statistically significant and which are not, to give a confidence interval for each regression coefficient that it estimates, to comunicatie how simple it and how interpretable it is. Here , i run a simple linear regression using a single precitor, then run the multiple linear regression. I extended the linear regression by adding interactions effect to know the other variables has an effect.

Data split:

At first i splitted the dataset (80% 20%), 80% data will be used as a training set and 20% as a test set. I maintained this ratio for all the machine learning analyses that I applied in this study. So, at the end, I compared all the analysis results on 20% test data(compare test RMSE).

```
#Split the turbine data set (80% as a training data), 20% as a testing data
index <- sample(nrow(turbine), nrow(turbine) * 0.80)
turbine.train <- turbine[index, ]
turbine.test <- turbine[-index, ]
dim(turbine.train) # check the dimension of the train dataset

## [1] 9600   13

dim(turbine.test) # dimension of the test dataset

## [1] 2400   13
```

Single variable Linear regression

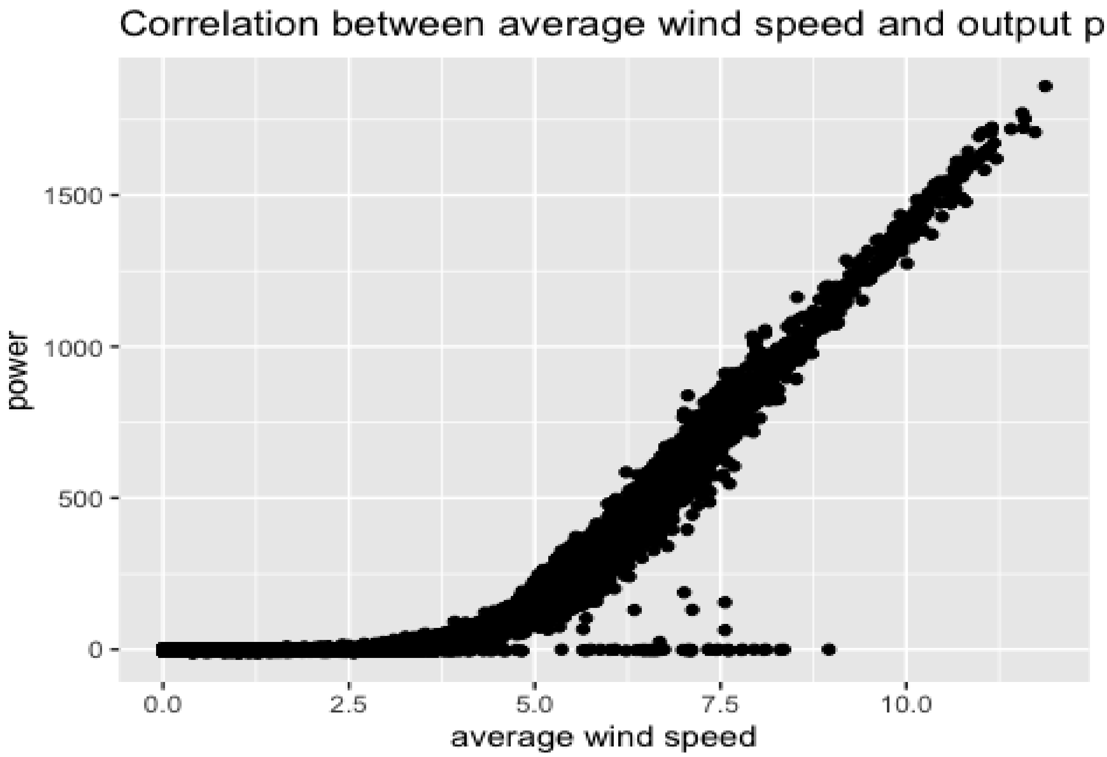
Wind energy produces power from wind speed and other factors. From the initial analysis of the data set (like, correlation and pair plot), wind speed is the highly correlated feature with response variable. The correlation result is

```
cor(Ws_avg, P_avg) # correlation between wind speed and power

## [1] 0.8288955
```

The correlation result shows that the relationship between the single predictor (Ws_avg) in the dataset with the output variable (P_avg) is highly positive (0.82889).

plot Correlation between average wind speed (Ws_avg) and output power (P_avg)

```
ggplot(turbine, aes(x = Ws_avg, y = P_avg)) +
  geom_point() +
  theme(legend.position = "none") +
  scale_x_continuous(labels = scales::comma_format()) +
  labs(x = "average wind speed",
       y = "power",
       title = "Correlation between average wind speed and output power")
```



Now train a linear model with a single predictor, here (average wind speed (Ws_avg).

After performing simple linear regression, I am going to answer the following research questions-

1. Is there a relationship between the wind speed (predictor) and the power (response)?
2. How strong is the relationship?
3. Is the relationship between the predictor (wind speed) and the response (power) positive or negative?
4. What is the predicted power associated with a wind speed of 12? What are the associated 95% confidence and prediction intervals?

```
set.seed(1)
library(ISLR)
attach (turbine.train)

## The following objects are masked from turbine:
##
##     Ot_avg, Ot_max, Ot_min, Ot_std, P_avg, Wa_avg, Wa_max, Wa_min,
##     Wa_std, Ws_avg, Ws_max, Ws_min, Ws_std

lm.fit=lm(P_avg~Ws_avg,data=turbine.train)
summary(lm.fit)

##
## Call:
## lm(formula = P_avg ~ Ws_avg, data = turbine.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -593.60 -102.79  -34.08   58.67  853.63
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -252.1917     3.2276  -78.14   <2e-16 ***
## Ws_avg       101.2207     0.6991  144.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.2 on 9598 degrees of freedom
## Multiple R-squared:  0.6859, Adjusted R-squared:  0.6859
## F-statistic: 2.096e+04 on 1 and 9598 DF,  p-value: < 2.2e-16
```

1. The p-value for the simple linear model is very small (<<0.05), so there is strong evidence to believe that average wind speed (Ws_avg) is associated with output power (P_avg). Therefore, there is a relationship between the predictor and response.
2. Residual standard error is 145, that means the 145 amount that the response (power) will deviate from the true regression line. and From the R^2 of the model which is 0.6864 (68%). It explained "the percentage of variability in the response that is explained by the predictor", here only wind speed (Ws_avg) explains 68% of the variance in target variable (P_avg)
3. The relationship is positive,as we can see from the parameter estimate by-

```
coefficients(lm.fit)[2]

##   Ws_avg
## 101.2207
```

It means that, if a wind turbine has higher wind speed, it will generally produce higher power.

4. What is the predicted power associated with a wind speed of 12? What are the associated 95% confidence and prediction intervals?

The confidence interval:

```
predict(lm.fit, data.frame(Ws_avg = 12), interval = "confidence", level = 0.95)

##       fit      lwr      upr
## 1 962.457 951.2945 973.6196
```

The 95% confidence interval shows that fitted value is 974 and lower interval is 962 and upper interval is 985.

The prediction interval:

```
predict(lm.fit, data.frame(Ws_avg = 12), interval = "prediction", level = 0.95)

##       fit      lwr     upr
## 1 962.457 683.4642 1241.45
```

The 95% prediction interval shows that fitted value is 974 and lower interval is 689 and upper interval is 1258. The prediction interval is wider than the confidence interval as we would expect.

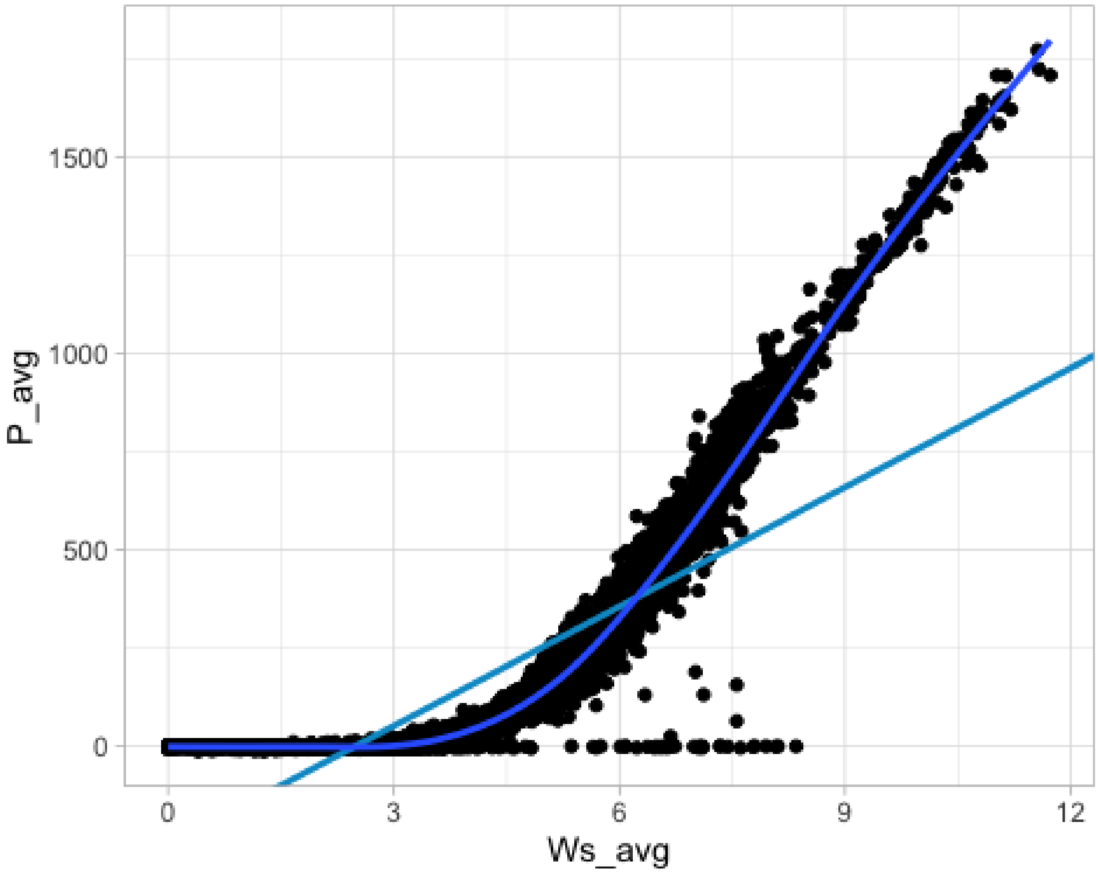
```
#prediction test data
pred1 <- predict(lm.fit, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##        RMSE          R2
## 153.3105454   0.6926686
```

Test rmse is 153 and R2 value is 0.69266 (69%). Test RMSE explains on an average how much of the predicted value will be off from the actual value. Based on test RMSE=153, we can conclude that on an average predicted value will be off by 153 from the actual value, if we just use single predictor (only wind speed and when model is linear)

#plot response (P_avg) vs Predictor (Ws_avg)

```
theme_set(theme_light())
```

```
ggplot(turbine.train, aes(x = Ws_avg, y = P_avg)) +
  geom_point() +
  geom_abline(intercept = coef(lm.fit)[1], slope = coef(lm.fit)[2],
              col = "deepskyblue3",
              size = 1) +
  geom_smooth(se = F)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Deepblue line indicates the actual output line and skyblue line indicates the predicted line by using single predictor (Ws_avg)

Multivariate linear regression

If we want to know the effect of other variables from the dataset, then we apply a multivariate linear model. I am going to answer the following question after running a multivariate linear model.

1. Is there a relationship between the predictors and the response?
2. Which predictors appear to have a statistically significant relationship to the response?
3. Is this model fit?
4. Is there evidence of non-linear association between any of the predictors and the response?
5. What is the condition of model fit,if some non-significant variables are dropped from the model?
6. What is the Test RMSE value for the Multivariate Linear model?

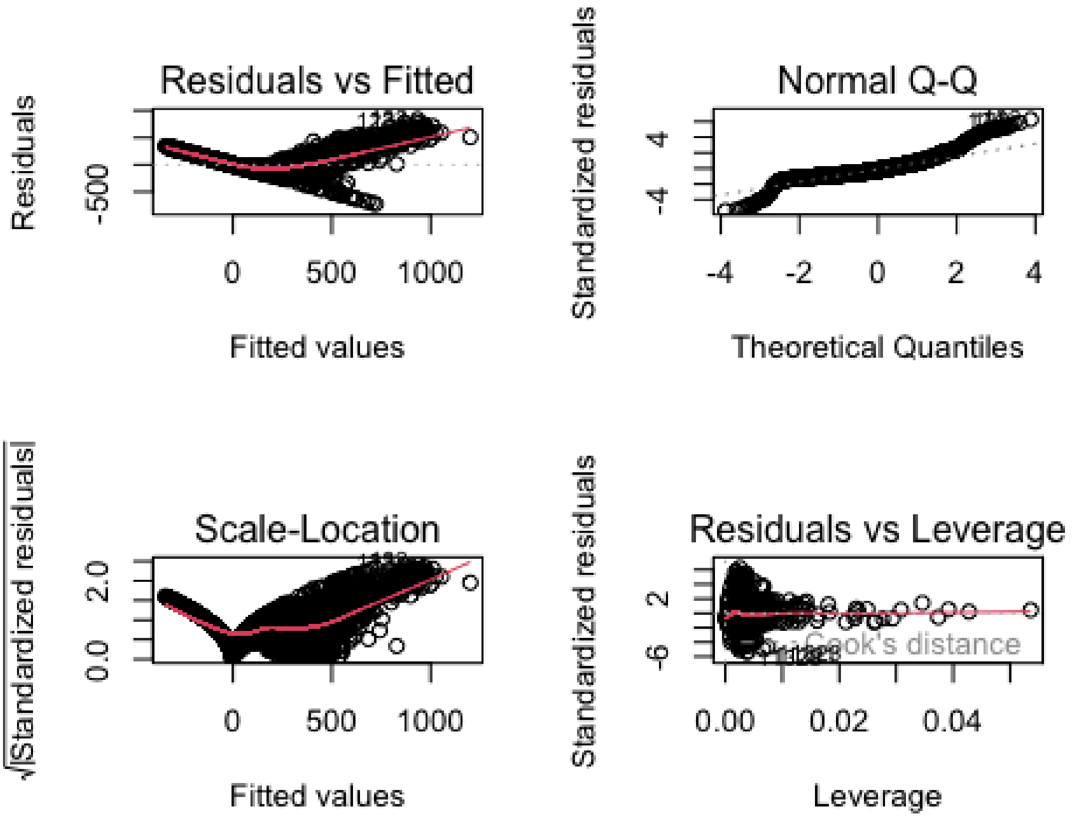
```
lm.fit=lm(P_avg~.,data=turbine.train)
summary(lm.fit)
```

```
## 
## Call:
## lm(formula = P_avg ~ ., data = turbine.train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -722.81  -89.79  -28.64   58.04  836.58 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -326.45881    5.79998 -56.286  < 2e-16 ***
## Ws_avg        51.81730    5.58581   9.277  < 2e-16 ***
## Ws_min        27.18389    4.44633   6.114 1.01e-09 ***
## Ws_max        12.49736    4.44501   2.812  0.00494 ** 
## Ws_std       197.66535   14.73067  13.419  < 2e-16 ***
## Wa_avg         0.10101    0.01505   6.711 2.05e-11 ***
## Wa_min        -0.10827    0.01559  -6.946 4.00e-12 ***
## Wa_max         0.02717    0.01419   1.915  0.05557 .  
## Wa_std         0.22748    0.02869   7.930 2.44e-15 ***
## Ot_avg        51.37308   27.58362   1.862  0.06257 .  
## Ot_min       -55.41841   22.40837  -2.473  0.01341 *  
## Ot_max         3.96604   21.24744   0.187  0.85193    
## Ot_std        59.60721   56.78664   1.050  0.29390    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 133.7 on 9587 degrees of freedom
## Multiple R-squared:  0.7228, Adjusted R-squared:  0.7224 
## F-statistic:  2083 on 12 and 9587 DF,  p-value: < 2.2e-16
```

1. The p-value is given at the bottom of the model summary (p-value: < 2.2e-16), so it's clear that the probability of the null hypothesis being true (given our data) is practically zero. We reject the null hypothesis (and hence conclude that there is a relationship between the predictors and output power (P_avg).
2. Using the coefficient p-values in the model output, and p = 0.05 as my threshold for significance, all variables except Ws_max, Wa_max, Ot_min & Ot_max have a statistically significant relationship with the response.
3. R-squared measures the goodness of fit of a regression model. Hence, a higher R-squared indicates the model is a good fit while a lower R-squared indicates the model is not a good fit. Here, R2 =0.7224 is closer to 1 and so this model is a good fit.

<u>Diagnostic Plots</u>

```
par(mfrow=c(2,2))
plot(lm.fit)
```



4. Is there evidence of non-linear association between any of the predictors and the response?

Answer from the diagnostic plot:

Fitted vs Residual graph: Residuals plots should be random in nature and there should not be any pattern in the graph. The average of the residual plot should be close to zero. From the above plot, we can see that the red trend line is not zero.

Normal Q-Q Plot: Q-Q plot shows whether the residuals are normally distributed. Ideally, the plot should be on the dotted line. If the Q-Q plot is not on the line then models need to be reworked to make the residual normal. In the above plot, we see that most of the plots are on the line except at the end.

Scale-Location This shows how the residuals are spread and whether the residuals have an equal variance or not.

Residuals vs Leverage The plot helps to find influential observations. Here we need to check for points that are outside the dashed line. A point outside the dashed line will be an influential point and removal of that will affect the regression coefficients.

Overall, some pattern in the residuals might suggest some non-linearity in the relationships between the predictors and response that the model is not currently capturing. In nonlinear model analysis, it will be more confirmed.

```
# remove the less significant feature from the model based on p value
lm.fit2 = update(lm.fit, ~.-Ws_max-Wa_max-Ot_min-Ot_max-Ot_avg)
summary(lm.fit2)

##
## Call:
## lm(formula = P_avg ~ Ws_avg + Ws_min + Ws_std + Wa_avg + Wa_min +
##     Wa_std + Ot_std, data = turbine.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -726.35  -89.84  -29.40   57.52  830.64
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -321.79735    5.22120 -61.633  < 2e-16 ***
## Ws_avg        62.74905    3.93474  15.947  < 2e-16 ***
## Ws_min        30.96531    4.27814   7.238 4.90e-13 ***
## Ws_std       229.86584    9.77834  23.508  < 2e-16 ***
## Wa_avg         0.10961    0.01425   7.689 1.62e-14 ***
## Wa_min        -0.11676    0.01500  -7.786 7.61e-15 ***
## Wa_std         0.23088    0.02863   8.064 8.25e-16 ***
## Ot_std       138.74421   26.28783   5.278 1.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 133.8 on 9592 degrees of freedom
## Multiple R-squared:  0.7222, Adjusted R-squared:  0.722
## F-statistic:  3563 on 7 and 9592 DF,  p-value: < 2.2e-16
```

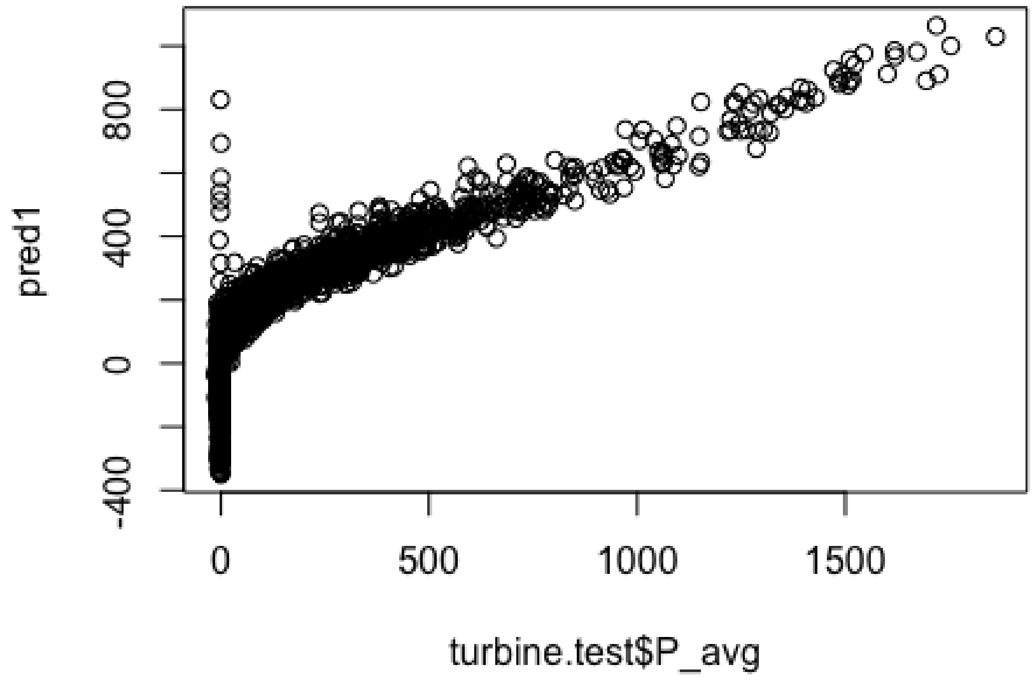
Now in this model, all the predictors are significant.

5. Now adjusted R2 =0.722 is closer to 1 and so this model is a good fit. Please note that this value has decreased a little from the first model but this should be fine as removing five predictors caused a drop from 0.7224 to 0.722 and this is a small drop. In other words, the contribution of three predictors towards explaining the variance is an only small value(0.0004) and hence it is better to drop the predictor.

```
#prediction on test data
pred1 <- predict(lm.fit2, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##        RMSE          R2
## 144.2957393   0.7278593
```

6. The test RMSE is 144 and R^2 value is 0.727. Test RMSE explains on an average how much of the predicted value will be off from the actual value. Based on RMSE=144, we can conclude that on an average predicted value will be off by 144 from the actual value.

```
plot(turbine.test$P_avg, pred1)
```



The graph shows the predicted value with the actual value's relationship.

Interaction Terms

I am going to answer the following question after adding the interactions with the linear model.

1. Do any interactions appear to be statistically significant?
2. Is there any change after adding Interactions terms to the model?

```
summary(lm(formula = P_avg ~ . * .-Ws_max-Wa_max-Ot_min-Ot_max-Ot_avg, data =
turbine.train))

##
## Call:
## lm(formula = P_avg ~ . * . - Ws_max - Wa_max - Ot_min - Ot_max -
##     Ot_avg, data = turbine.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -699.03  -21.37   -0.70   21.01  275.10
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.662e+01  4.986e+00  17.370  < 2e-16 ***
## Ws_avg         2.906e+00  6.377e+00   0.456 0.648653
## Ws_min        -1.429e+02  7.064e+00 -20.224  < 2e-16 ***
## Ws_std        -7.718e+01  1.581e+01  -4.882 1.07e-06 ***
## Wa_avg        -1.102e-01  2.914e-02  -3.783 0.000156 ***
## Wa_min        -9.119e-02  2.094e-02  -4.356 1.34e-05 ***
## Wa_std        -2.168e-01  4.702e-02  -4.610 4.07e-06 ***
## Ot_std        -4.563e+01  4.079e+01  -1.119 0.263311    
## Ws_avg:Ws_min  2.228e+01  1.035e+00  21.520  < 2e-16 ***
## Ws_avg:Ws_max  1.050e+01  6.103e-01  17.208  < 2e-16 ***
## Ws_avg:Ws_std -8.324e+01  7.224e+00 -11.523  < 2e-16 ***
## Ws_avg:Wa_avg  5.373e-03  2.194e-02   0.245 0.806550    
## Ws_avg:Wa_min  1.370e-03  2.289e-02   0.060 0.952259    
## Ws_avg:Wa_max  9.601e-02  1.965e-02   4.887 1.04e-06 ***
## Ws_avg:Wa_std  3.132e-02  3.243e-02   0.966 0.334185    
## Ws_avg:Ot_avg  6.934e+01  4.170e+01   1.663 0.096340 .  
## Ws_avg:Ot_min -3.880e+01  3.293e+01  -1.178 0.238847    
## Ws_avg:Ot_max -2.644e+01  3.186e+01  -0.830 0.406516    
## Ws_avg:Ot_std  6.426e+01  8.394e+01   0.766 0.443930    
## Ws_min:Ws_max -5.699e+00  1.059e+00  -5.380 7.65e-08 ***
## Ws_min:Ws_std  9.859e+01  4.754e+00  20.736  < 2e-16 ***
## Ws_min:Wa_avg  3.639e-02  1.673e-02   2.175 0.029631 *  
## Ws_min:Wa_min  6.099e-02  1.875e-02   3.253 0.001148 ** 
## Ws_min:Wa_max -2.170e-02  1.593e-02  -1.362 0.173288    
## Ws_min:Wa_std -4.316e-02  3.586e-02  -1.204 0.228729    
## Ws_min:Ot_avg -8.426e+01  3.320e+01  -2.538 0.011157 *  
## Ws_min:Ot_min  4.307e+01  2.731e+01   1.578 0.114707    
## Ws_min:Ot_max  3.838e+01  2.454e+01   1.564 0.117794    
## Ws_min:Ot_std  9.342e+01  6.523e+01   1.432 0.152108    
## Ws_max:Ws_std  9.517e+00  3.522e+00   2.702 0.006899 ** 
## Ws_max:Wa_avg -3.050e-02  1.733e-02  -1.760 0.078490 .  
## Ws_max:Wa_min -4.669e-02  1.652e-02  -2.827 0.004709 ** 
## Ws_max:Wa_max -5.657e-02  1.531e-02  -3.695 0.000221 ***
## Ws_max:Wa_std  7.976e-04  3.337e-02   0.024 0.980934    
## Ws_max:Ot_avg  1.221e+01  3.389e+01   0.360 0.718572    
## Ws_max:Ot_min -6.640e+00  2.631e+01  -0.252 0.800765    
## Ws_max:Ot_max -7.347e+00  2.581e+01  -0.285 0.775859    
## Ws_max:Ot_std -1.305e+02  6.715e+01  -1.944 0.051927 .  
## Ws_std:Wa_avg  5.984e-02  5.475e-02   1.093 0.274440    
## Ws_std:Wa_min  3.542e-01  5.666e-02   6.252 4.24e-10 ***
## Ws_std:Wa_max  1.842e-01  5.041e-02   3.654 0.000260 ***
## Ws_std:Wa_std  1.501e-01  1.008e-01   1.490 0.136210    
## Ws_std:Ot_avg -2.063e+02  1.108e+02  -1.861 0.062740 .  
## Ws_std:Ot_min  7.206e+01  8.701e+01   0.828 0.407597    
## Ws_std:Ot_max  1.330e+02  8.328e+01   1.597 0.110201    
## Ws_std:Ot_std  2.301e+02  2.143e+02   1.074 0.282947    
## Wa_avg:Wa_min  3.533e-04  7.917e-05   4.463 8.18e-06 ***
## Wa_avg:Wa_max -1.883e-04  8.899e-05  -2.116 0.034341 *  
## Wa_avg:Wa_std  2.518e-04  1.140e-04   2.208 0.027234 *  
## Wa_avg:Ot_avg  1.019e-01  1.124e-01   0.907 0.364470    
## Wa_avg:Ot_min -1.354e-01  8.787e-02  -1.540 0.123475    
## Wa_avg:Ot_max  3.642e-02  8.669e-02   0.420 0.674406    
## Wa_avg:Ot_std  3.722e-02  2.229e-01   0.167 0.867388    
## Wa_min:Wa_max -4.245e-05  9.990e-05  -0.425 0.670896    
## Wa_min:Wa_std -5.189e-05  1.393e-04  -0.372 0.709597    
## Wa_min:Ot_avg -8.569e-02  1.033e-01  -0.829 0.407025    
## Wa_min:Ot_min  1.881e-01  8.039e-02   2.340 0.019286 *  
## Wa_min:Ot_max -1.032e-01  8.030e-02  -1.285 0.198688    
## Wa_min:Ot_std  2.858e-01  2.064e-01   1.385 0.166224    
## Wa_max:Wa_std -6.386e-05  1.013e-04  -0.630 0.528605    
## Wa_max:Ot_avg  2.093e-02  9.857e-02   0.212 0.831849    
## Wa_max:Ot_min  1.409e-01  7.930e-02   1.777 0.075563 .  
## Wa_max:Ot_max -1.673e-01  7.510e-02  -2.228 0.025900 *  
## Wa_max:Ot_std  2.072e-01  2.017e-01   1.027 0.304456    
## Wa_std:Ot_avg  1.821e-01  1.792e-01   1.016 0.309507    
## Wa_std:Ot_min -7.671e-02  1.491e-01  -0.515 0.606837    
## Wa_std:Ot_max -1.044e-01  1.492e-01  -0.700 0.484087
## Wa_std:Ot_std  2.850e-01  4.574e-01   0.623 0.533288
## Ot_avg:Ot_min -2.837e-01  2.207e+00  -0.129 0.897734
## Ot_avg:Ot_max  2.273e+00  2.286e+00   0.995 0.319937
## Ot_avg:Ot_std -1.458e+02  7.229e+01  -2.016 0.043782 *
## Ot_min:Ot_max -1.832e+00  2.828e+00  -0.648 0.517142
## Ot_min:Ot_std  7.794e+01  4.439e+01   1.756 0.079178 .
## Ot_max:Ot_std  6.694e+01  3.319e+01   2.017 0.043707 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.86 on 9526 degrees of freedom
## Multiple R-squared:  0.9647, Adjusted R-squared:  0.9644
## F-statistic:  3567 on 73 and 9526 DF,  p-value: < 2.2e-16
```

1. We can see the significant terms (at the 0.05 level) are those with at least one asterisk (*). Using the standard threshold of 0.05, the significant interaction terms are given by:* Ws_avg:Ws_min *Ws_avg:Ws_max* Ws_avg:Ws_std *Ws_avg:Wa_max* Ws_min:Ws_max *Ws_min:Ws_std* Ws_min:Wa_avg *Ws_min:Wa_min* Ws_max:Ws_std
   * Ws_max:Wa_avg * Ws_std:Wa_min * Ws_std:Wa_max * Ot_max:Ot_std the above variable have interaction effect.
2. Yes, after adding the interaction terms to the model, RSE decreases considerably & R2/adjusted R2 increases considerably after adding the interaction terms to the model. New RSE is 47.86 and r-squared value is 0.964

Chapter 3- Resampling Methods

- Why resampling methods?
- validation set approach
- Leave one out cross validation
- K-fold cross validation
- Model fit using cross validation approach
- Research Question-

12. Can we reduce the test error using the validation set approach, LOOCV, k-fold cross validation?
13. What is the value of k (k-fold) for minimum cross validation error?
14. What is the Test RMSE and R2 value after fitting a model with cross validation?

Why resampling methods

Resampling methods are an indispensable tool in modern statistics. They involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model. For example, in order to estimate the variability of a linear regression fit, we can repeatedly draw different samples from the training data, fit a linear regression to each new sample, and then examine the extent to which the resulting fits differ. Such an approach may allow us to obtain information that would not be available from fitting the model only once using the original training sample. In this analysis, I applied validation set approach, leave one out cross validation and k fold cross validation approach. In validation set approach, the available training data is randomly divided into a training set and test (hold-out) set. A model is fit to the training set, and is used to make predictions on the data in the test set. The error from the predictions on the test set serves as the estimate for the test error.Similar to the Validation Set Approach, LOOCV involves splitting a full dataset into separate training and test sets. However, only a single observation is included in the test set, and all of the remaining observations are assigned to the training set.A model is fit to the training set, and a prediction is made for the single excluded observation in the test set. The test error is determined for the prediction. Then, the LOOCV procedure is repeated to individually exclude each observation from the full dataset. Finally, the LOOCV estimate for the test error is simply the average of all of the individual test errors. K-Fold Cross-Validation (K-Fold CV) is an alternative to LOOCV that divides the observations into *K* groups (folds) of approximately the same size.The first fold is treated as the test set, and a model is fit to the remaining folds. The fitted model is used to make predictions on the observations in the first fold, and the test error is measured. The K-Fold CV procedure is repeated *K* times so that a different fold is treated as the test set each time. Finally, the K-Fold CV estimate of the test error is simply the average of the *K* test measures.

Validation set approach

I carry out the Validation set approach on our dataset. I use the set.seed() to randomly generate samples. I will randomly divide the observations into 6000 train and 6000 validation sets.

```
#The Validation Set Approach
#random split into train and test
set.seed(1)
# 50%-%50% split for training and testing (training data half of the data)
train=sample(12000,6000)
lm.fit=lm(P_avg~Ws_avg+Ws_min+Ws_std+Wa_avg+Wa_min+Wa_std+Ot_std,data=turbine,subs
et=train)
summary(lm.fit)

##
## Call:
## lm(formula = P_avg ~ Ws_avg + Ws_min + Ws_std + Wa_avg + Wa_min +
##     Wa_std + Ot_std, data = turbine, subset = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -644.34  -91.29  -29.96   59.68  805.41
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -330.03471    6.63141 -49.768  < 2e-16 ***
## Ws_avg        69.13849    5.05446  13.679  < 2e-16 ***
## Ws_min        26.03650    5.50382   4.731 2.29e-06 ***
## Ws_std       230.95588   12.40364  18.620  < 2e-16 ***
## Wa_avg         0.10328    0.01808   5.712 1.17e-08 ***
## Wa_min        -0.13190    0.01907  -6.915 5.15e-12 ***
## Wa_std         0.23748    0.03598   6.600 4.48e-11 ***
## Ot_std       131.11140   33.27513   3.940 8.23e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 134.4 on 5992 degrees of freedom
## Multiple R-squared:  0.7307, Adjusted R-squared:  0.7304
## F-statistic:  2322 on 7 and 5992 DF,  p-value: < 2.2e-16
```

I run the predict() function using the fit model and observe the validation set error rate.

```
#compute MSE on testing data
mean((P_avg-predict(lm.fit,turbine))[-train]^2)

## [1] 116491.7

#TEST RMSE
sqrt(mean((P_avg-predict(lm.fit,turbine))[-train]^2))



## [1] 135.3088
```

1. Using the validation set approach, I found Test RMSE is 135, which is less than the previous test.

Leave one out cross validation(LOOCV)

I use the cv.glm() function in the boot library in R to carry out Leave-One-Out Cross-Validation.

```
library(boot)
# glm- generalized linear model
glm.fit=glm(P_avg~Ws_avg+Ws_min+Ws_std+Wa_avg+Wa_min+Wa_std+Ot_std,data=turbine)

# The LOOCV estimate using the cv.glm() function
cv.err=cv.glm(turbine,glm.fit)
cv.err$delta

## [1] 18501.87 18501.87

#TEST RMSE using LOOCV method
sqrt(cv.err$delta)

## [1] 136.0216 136.0216
```

using LOOCV, I have found Test RMSE is 136, MSE is 18501

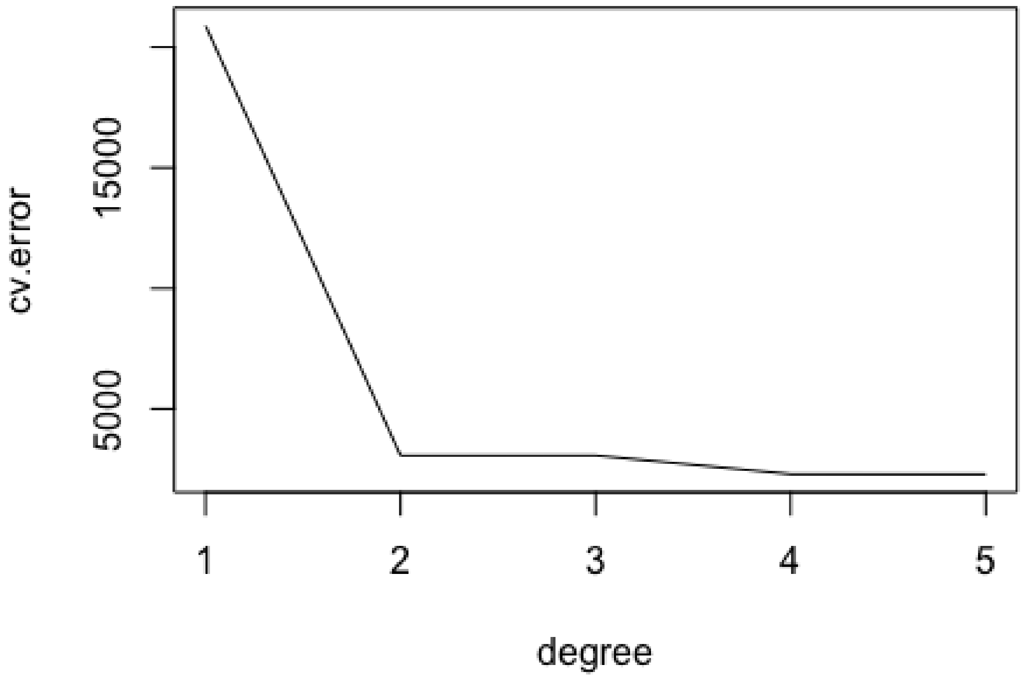
K-fold cross validation

For K-Fold Cross-Validation, I use cv.glm() function to implement k-fold CV. I set K = 5 first and also use k=10.For both procedures, I get CV errors corresponding to polynomial fits for orders 1 to 5.

```
# 5-Fold Cross-Validation
set.seed(17)
cv.error=rep(0,5)
for (i in 1:5){
  glm.fit=glm(P_avg~poly(Ws_avg,i),data=turbine)
  cv.error[i]=cv.glm(turbine,glm.fit)$delta[1]
}
cv.error

## [1] 20887.314  3068.310  3070.050  2295.180  2289.011

degree=1:5
```

plot using k=5 fold cross validation

```
plot(degree,cv.error, type="l")
```

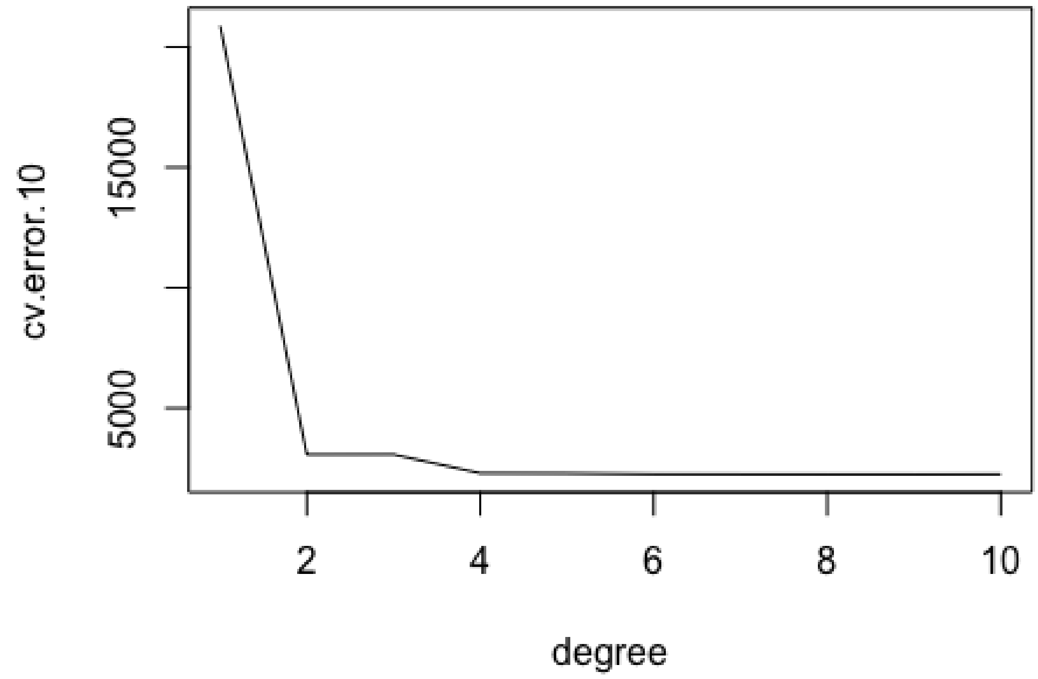


Now compute K=10 fold cross validation

```
# 10-Fold Cross-Validation

set.seed(17)
cv.error.10=rep(0,10) # 10 cross validation
for (i in 1:10){
  glm.fit=glm(P_avg~poly(Ws_avg,i),data=turbine)
  cv.error.10[i]=cv.glm(turbine,glm.fit,K=10)$delta[1]
}
cv.error.10

##  [1] 20880.007  3067.742  3070.736  2293.377  2286.053  2255.511  2250.942
##  [8]  2249.491  2248.899  2252.308

degree=1:10
```

1. using cross validation error, The lowest MSE is 2248 when k=9

```
plot(degree,cv.error.10, type="l")
```



Graph explains the cross validation error and lowest MSE occured when k=9

Model fit using k fold cross validation

```
#model train using k fold cross validation
library(caret)

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

set.seed(1)
ctrl <- trainControl(
  method = "cv",
  number = 10,
)

model1 <- train(
  P_avg ~ .,
  data = turbine.train,
  method = 'lm',
  trControl = ctrl
)
(model1)
```

```
## Linear Regression
##
## 9600 samples
##   12 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 8640, 8641, 8641, 8640, 8640, 8640, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   133.6928  0.7227536  97.6288
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

linear model train with 10 fold cross validation

```
#prediction
pred1 <- predict(model1, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##        RMSE          R2
## 144.2711368   0.7279825
```

2. Prediction result shows that test RMSE is 144.27 with R^2 value is 0.7279

Chapter 4- Subset selection and Regularization

- Why subset selection and Regularization?
- Best subset selection
- Forward and Backward subset selection
- Choosing among models using cross-validation
- Model fit using best 7 features
- Ridge Regression
- Lasso Regression
- Principal Component Regression (PCR)
- Partial Least Squares
- Research Question-

15. Which are the most important and the least significant variables (features) in best subset methods?
16. Using adjusted R2, what is the number of variables that best subset methods choose?
17. Using Cp, what is the number of variables that best subset methods choose?
18. Using BIC, what is the number of variables that best subset methods choose?
19. What are the 7 variables that best subset methods choose?
20. Which are the most important and the least significant variables (features) in forward stepwise selection methods?
21. Which are the most important and the least significant variables (features) in backward stepwise selection methods?
22. Do all methods (best, forward, backward) select the same features (say first 7 features)?
23. What is the lowest cross validation error for among all models?
24. Why do I select the best 7 features?
25. What is the TEST RMSE, after fitting the model using 7 best features?
26. What does coefficient path display in ridge and lasso regression?

27. how do you choose the tuning parameter and what is the value of tunning parameter in ridge and lasso regression?

28. how many components should i select for modeling stage in PCR and PLS?

Why subset selection and Regularization?

The linear model has distinct advantages in terms of inference and, on real-world problems. So if we can improve the linear model by replacing plain least squares with some alternative fitting procedures, it will increase prediction accuracy and model interpretability.There are three classes of alternative methods to least squares: **subset selection, shrinkage**, and **dimension reduction**. **Subset selection** involves identifying a subset of the *p* predictors that are believed to be related to the response, and fitting a least squares model to the reduced set of predictors.**Shrinkage** involves fitting a model involving all *p* predictors. However, the estimated coefficients are shrunken towards zero relative to the least squares estimates. Shrinkage reduces variance, and may perform variable selection.**Dimension reduction** involves projecting all of the *p* predictors into an *M*-dimensional subspace where M < p. *M* different "linear combinations" of all of the *p* predictors are computed, and are used as predictors to fit a linear regression model through least squares.

Best subset selection

I apply the best subset selection approach on the turbine dataset. Although there are other packages to do this, I use the bestglm package in R package to carry out subset selection. The leaps package is used for linear regression .

```
#best subset selection_method
library(leaps)
best_subset=regsubsets(P_avg~.,data=turbine,nvmax=12)
best.summary=summary(best_subset)
best.summary

## Subset selection object
## Call: regsubsets.formula(P_avg ~ ., data = turbine, nvmax = 12)
## 12 Variables  (and intercept)
##        Forced in Forced out
## Ws_avg     FALSE      FALSE
## Ws_min     FALSE      FALSE
## Ws_max     FALSE      FALSE
## Ws_std     FALSE      FALSE
## Wa_avg     FALSE      FALSE
## Wa_min     FALSE      FALSE
## Wa_max     FALSE      FALSE
## Wa_std     FALSE      FALSE
## Ot_avg     FALSE      FALSE
## Ot_min     FALSE      FALSE
## Ot_max     FALSE      FALSE
## Ot_std     FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
##           Ws_avg Ws_min Ws_max Ws_std Wa_avg Wa_min Wa_max Wa_std Ot_avg Ot_min
## 1  ( 1 )  " "    " "    "*"    " "    " "    " "    " "    " "    " "    " "
## 2  ( 1 )  "*"    " "    " "    "*"    " "    " "    " "    " "    " "    " "
## 3  ( 1 )  "*"    " "    " "    "*"    " "    " "    " "    "*"    " "    " "
## 4  ( 1 )  "*"    "*"    " "    "*"    " "    " "    " "    "*"    " "    " "
## 5  ( 1 )  "*"    " "    " "    "*"    "*"    "*"    " "    "*"    " "    " "
## 6  ( 1 )  "*"    "*"    " "    "*"    "*"    "*"    " "    "*"    " "    " "
## 7  ( 1 )  "*"    "*"    " "    "*"    "*"    "*"    " "    "*"    " "    " "
## 8  ( 1 )  "*"    "*"    "*"    "*"    "*"    "*"    " "    "*"    " "    " "
## 9  ( 1 )  "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    " "    " "
## 10  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
## 11  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
## 12  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
##            Ot_max Ot_std
## 1  ( 1 )  " "     " "   
## 2  ( 1 )  " "     " "   
## 3  ( 1 )  " "     " "   
## 4  ( 1 )  " "     " "   
## 5  ( 1 )  " "     " "   
## 6  ( 1 )  " "     " "   
## 7  ( 1 )  " "     "*"   
## 8  ( 1 )  " "     "*"   
## 9  ( 1 )  " "     "*"   
## 10  ( 1 ) " "     " "   
## 11  ( 1 ) " "     "*"   
## 12  ( 1 ) "*"     "*"
```

Asterisk indicates that a given variable is included in the corresponding model.

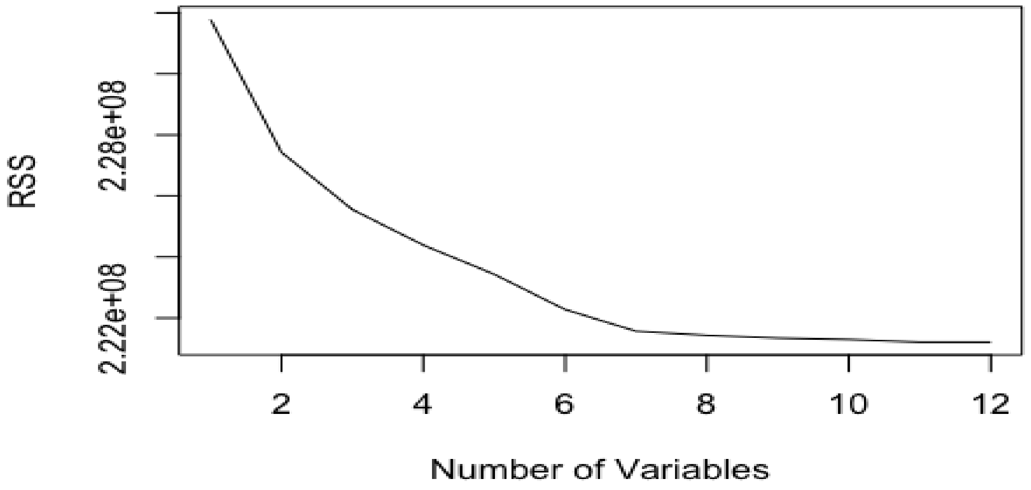
Question: Which are the most important and the least significant variables (features) in best subset methods?

From above, the output indicated that the **best one-variable** model contains Ws_max(maximum wind speed). Best two-variable model contains only Ws_avg and Ws_std, the best three-variable model contains Ws_avg,Ws_std and Wa_std etc. The variable Ot_max (maximum outdoor temperature) is **less significant** for the best subset method.

```
names(best.summary)

## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

The summary() function returns R2, RSS, adjusted R2, Cp, and BIC. We can examine these to try to select the best overall model.

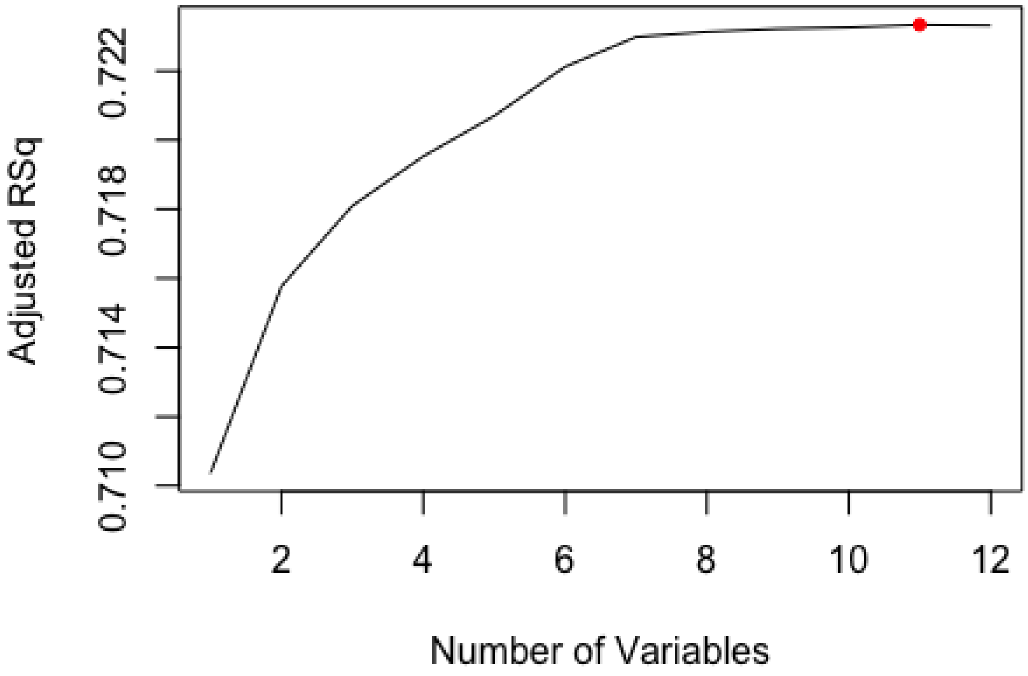
```
plot(best.summary$rss,xlab="Number of Variables",ylab="RSS",type="l")
```



From the Residual Sum of Squares (RSS) plot, I see as the number of variables increases, the RSS decreases.

```
plot(best.summary$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="l")
which.max(best.summary$adjr2)

## [1] 11
```

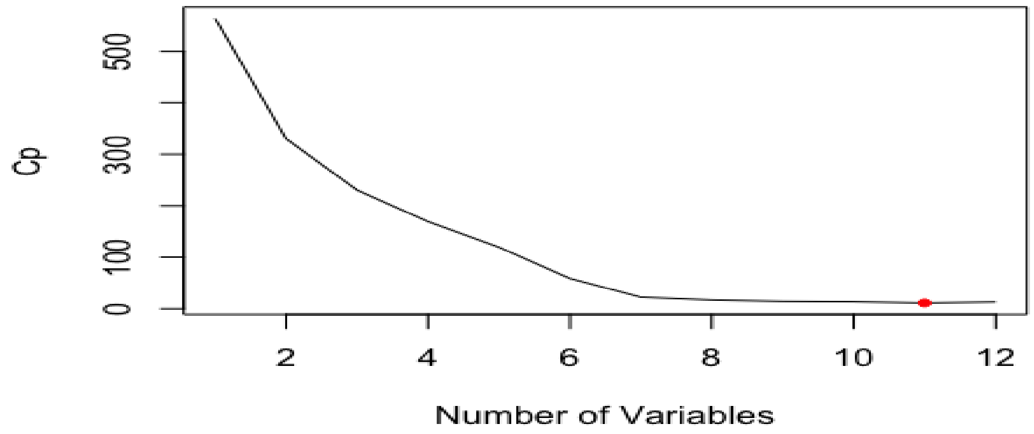
```
points(11,best.summary$adjr2[11], col="red",pch=20)
```



Using **Question:** Using adjusted R2, what is the number of variables that best subset methods choose?

The best-subset algorithm selects the **11-variable model using** Adjusted R2

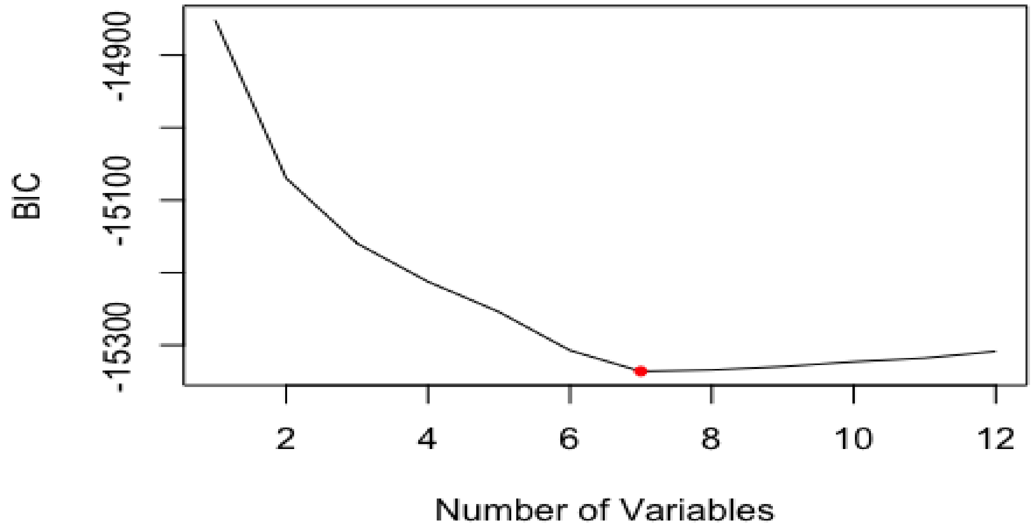
```
plot(best.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
which.min(best.summary$cp)

## [1] 11

points(11,best.summary$cp[11],col="red",pch=20)
```



Question: Using Cp, what is the number of variables that best subset methods choose?

Using Cp, the best-subset algorithm selects the 11-variable model

```
plot(best.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')
which.min(best.summary$bic)

## [1] 7

points(7,best.summary$bic[7],col="red",pch=20)
```



Question: UsingBIC, what is the number of variables that best subset methods choose?

Using BIC, the best-subset algorithm selects the **7-variable model**, with the following coefficients:

```
# coefficients
coef(best_subset,7)

##  (Intercept)       Ws_avg       Ws_min       Ws_std       Wa_avg       Wa_min
## -324.8785076   64.4330753   30.0534410  230.3349762    0.1085308   -0.1191269
##       Wa_std       Ot_std
##    0.2363111  146.4120690
```

Question: what are the 7 variables that best subset methods choose?

The best subset algorithm chooses the (Ws_avg, Ws_min,Ws_std, Wa_avg, Wa_min, Wa_std, Ot_std) variables.

Forward stepwise selection

The regsubsets() function will be used for forward stepwise selection, by adding the parameter method="forward".

```
# Forward subset selection method
subset.fwd=regsubsets(P_avg~.,data=turbine,nvmax=12,method="forward")
summary(subset.fwd)

## Subset selection object
## Call: regsubsets.formula(P_avg ~ ., data = turbine, nvmax = 12, method =
"forward")
## 12 Variables  (and intercept)
##        Forced in Forced out
## Ws_avg     FALSE      FALSE
## Ws_min     FALSE      FALSE
## Ws_max     FALSE      FALSE
## Ws_std     FALSE      FALSE
## Wa_avg     FALSE      FALSE
## Wa_min     FALSE      FALSE
## Wa_max     FALSE      FALSE
## Wa_std     FALSE      FALSE
## Ot_avg     FALSE      FALSE
## Ot_min     FALSE      FALSE
## Ot_max     FALSE      FALSE
## Ot_std     FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: forward
##           Ws_avg Ws_min Ws_max Ws_std Wa_avg Wa_min Wa_max Wa_std Ot_avg Ot_min
## 1  ( 1 )  " "    " "    "*"    " "    " "    " "    " "    " "    " "    " "
## 2  ( 1 )  " "    " "    "*"    " "    " "    " "    " "    "*"    " "    " "
## 3  ( 1 )  " "    " "    "*"    " "    " "    "*"    " "    "*"    " "    " "
## 4  ( 1 )  " "    " "    "*"    " "    "*"    "*"    " "    "*"    " "    " "
## 5  ( 1 )  "*"    " "    "*"    " "    "*"    "*"    " "    "*"    " "    " "
## 6  ( 1 )  "*"    " "    "*"    "*"    "*"    "*"    " "    "*"    " "    " "
## 7  ( 1 )  "*"    "*"    "*"    "*"    "*"    "*"    " "    "*"    " "    " "
## 8  ( 1 )  "*"    "*"    "*"    "*"    "*"    "*"    " "    "*"    " "    " "
## 9  ( 1 )  "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    " "    " "
## 10  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    " "
## 11  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
## 12  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
##           Ot_max Ot_std
## 1  ( 1 )  " "    " "
## 2  ( 1 )  " "    " "
## 3  ( 1 )  " "    " "
## 4  ( 1 )  " "    " "
## 5  ( 1 )  " "    " "
## 6  ( 1 )  " "    " "
## 7  ( 1 )  " "    " "
## 8  ( 1 )  " "    "*"
## 9  ( 1 )  " "    "*"
## 10  ( 1 ) " "    "*"
## 11  ( 1 ) " "    "*"
## 12  ( 1 ) "*"    "*"
```

Question: Which are the most important and the least significant variables (features) in forward stepwise selection methods?

Following the above forward stepwise selection, the best **one variable** model contains only **Ws_max**, and the best two-variable contains Ws_max and Wa_std and so on. The least significant features is Ot_max

Backward subset method

The regsubsets() function can also be used for backward stepwise selection, by adding the parameter method=“backward”.

```
#Backward subset selection method
subset.bwd=regsubsets(P_avg~.,data=turbine,nvmax=12,method="backward")
summary(subset.bwd)

## Subset selection object
## Call: regsubsets.formula(P_avg ~ ., data = turbine, nvmax = 12, method = 
"backward")
## 12 Variables  (and intercept)
##        Forced in Forced out
## Ws_avg     FALSE      FALSE
## Ws_min     FALSE      FALSE
## Ws_max     FALSE      FALSE
## Ws_std     FALSE      FALSE
## Wa_avg     FALSE      FALSE
## Wa_min     FALSE      FALSE
## Wa_max     FALSE      FALSE
## Wa_std     FALSE      FALSE
## Ot_avg     FALSE      FALSE
## Ot_min     FALSE      FALSE
## Ot_max     FALSE      FALSE
## Ot_std     FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: backward
##           Ws_avg Ws_min Ws_max Ws_std Wa_avg Wa_min Wa_max Wa_std Ot_avg Ot_min
## 1  ( 1 )  "*"    " "    " "    " "    " "    " "    " "    " "    " "    " "   
## 2  ( 1 )  "*"    " "    " "    "*"    " "    " "    " "    " "    " "    " "   
## 3  ( 1 )  "*"    " "    " "    "*"    " "    " "    " "    "*"    " "    " "   
## 4  ( 1 )  "*"    " "    " "    "*"    " "    "*"    " "    "*"    " "    " "   
## 5  ( 1 )  "*"    " "    " "    "*"    "*"    "*"    " "    "*"    " "    " "   
## 6  ( 1 )  "*"    "*"    " "    "*"    "*"    "*"    " "    "*"    " "    " "   
## 7  ( 1 )  "*"    "*"    " "    "*"    "*"    "*"    " "    "*"    "*"    " "   
## 8  ( 1 )  "*"    "*"    " "    "*"    "*"    "*"    " "    "*"    "*"    "*"   
## 9  ( 1 )  "*"    "*"    "*"    "*"    "*"    "*"    " "    "*"    "*"    "*"   
## 10  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"   
## 11  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"   
## 12  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"   
##           Ot_max Ot_std
## 1  ( 1 )  " "    " "
## 2  ( 1 )  " "    " "
## 3  ( 1 )  " "    " "
## 4  ( 1 )  " "    " "
## 5  ( 1 )  " "    " "
## 6  ( 1 )  " "    " "
## 7  ( 1 )  " "    " "
## 8  ( 1 )  " "    " "
## 9  ( 1 )  " "    " "
## 10  ( 1 ) " "    " "
## 11  ( 1 ) " "    "*"
## 12  ( 1 ) "*"    "*"
```

Question: Which are the most important and the least significant variables (features) in backward stepwise selection methods?

Following the above backward stepwise selection, the best **one variable** model contains only **Ws_avg**, and the best **two-variabl**e contains **Ws_avg and Ws_std** and so on. The least significant feature is Ot_max.

Now best seven-variable models identified by forward, backward and best subset methods:

```
coef(best_subset,7) # 7 features obtained using best subset

##  (Intercept)       Ws_avg       Ws_min       Ws_std       Wa_avg       Wa_min 
## -324.8785076   64.4330753   30.0534410  230.3349762    0.1085308   -0.1191269 
##       Wa_std       Ot_std 
##    0.2363111  146.4120690

coef(subset.fwd,7) # 7 features obtained using forward subset method

##  (Intercept)       Ws_avg       Ws_min       Ws_max       Ws_std       Wa_avg 
## -313.1746156   54.2386677   27.4848922   10.5261313  206.4988515    0.1061653 
##       Wa_min       Wa_std 
##   -0.1208245    0.2390077

coef(subset.bwd,7) # 7 features obtained using backward subset method

##  (Intercept)       Ws_avg       Ws_min       Ws_std       Wa_avg       Wa_min 
## -314.3327327   63.4570336   30.5257745  232.5068376    0.1045889   -0.1213810 
##       Wa_std       Ot_avg 
##    0.2431685    0.3126287
```

Question: Do all methods (best, forward, backward) select the same features (say first 7 features)?

No. Features identified by the above three approaches are different. Features selected by best subset and backward subset methods are almost similar except Ot_std, Ot_avg, but in forward methods, obtained features are totally different from others.

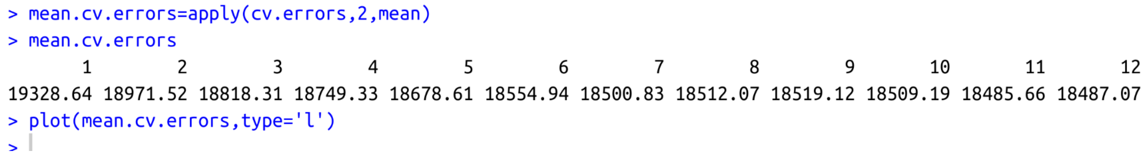
Choosing Among Models using Cross-Validation

R-code:

R-code:

```
#10-fold cross-validation for model selection
regfit.best=regsubsets(P_avg~.,data=turbine,nvmax=12)
k=10
set.seed(1)
folds=sample(1:k,nrow(turbine),replace=TRUE)
folds
```

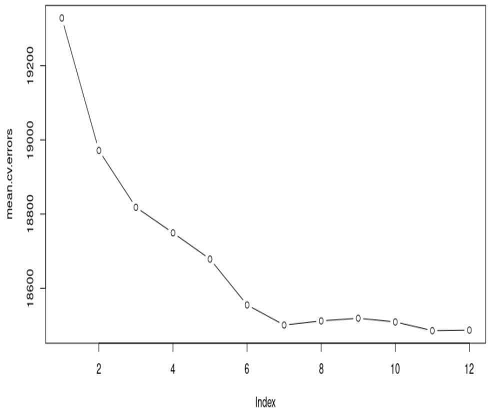
```
cv.errors=matrix(NA,k,12, dimnames=list(NULL, paste(1:12)))
cv.errors
for(j in 1:k){
  best.fit=regsubsets(P_avg~.,data=turbine[folds!=j,],nvmax=12)
  for(i in 1:12){
    pred=predict(best.fit,turbine[folds==j,],id=i)
    cv.errors[j,i]=mean( (turbine$P_avg[folds==j]-pred)^2)
  }
}
cv.errors
#apply function "mean" for each column
mean.cv.errors=apply(cv.errors,2,mean)
mean.cv.errors
plot(mean.cv.errors,type='b')
which.min(mean.cv.errors)
reg.best=regsubsets(P_avg~.,data=turbine, nvmax=12)
coef(reg.best,11)
plot(reg.best)
```



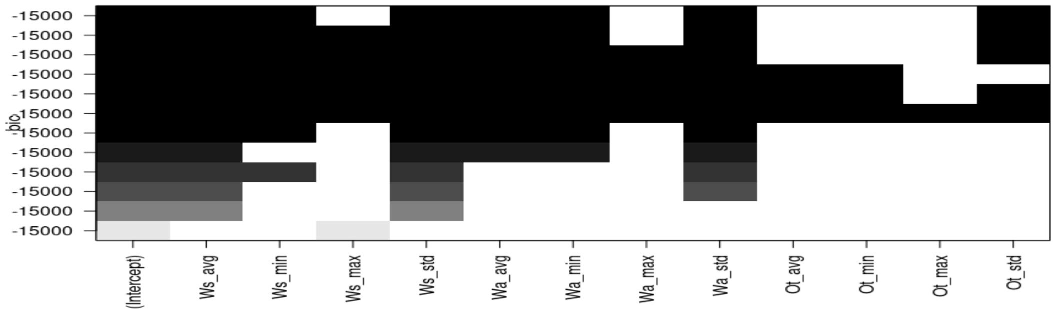
```
> mean.cv.errors=apply(cv.errors,2,mean)
> mean.cv.errors
       1        2        3        4        5        6        7        8        9       10       11       12 
19328.64 18971.52 18818.31 18749.33 18678.61 18554.94 18500.83 18512.07 18519.12 18509.19 18485.66 18487.07 
> plot(mean.cv.errors,type='l')
> 
```

What is the lowest cross validation error for among all models?

The lowest cross validation error is 1845.



Graph shows the model selection using cross validation.



Question: why do I select the best 7 features?
Graphs show the best 7 features. Although the minimum cross validation error is for 11 features, I selected 7 features. Because 7 features cross validation error(MSE=18500) and 11 features cross validation (MSE=18485) almost the same, but if I select 7 features, the model **interpretability** will be easier than 11 features.

Model fit using best 7 features

I will test this by fitting a Linear regression model with only these seven variables and checking the performance of this model.

```
#linear model with best  7 features
subset_fit=lm(P_avg~Ws_avg+Ws_min+Ws_std+Wa_avg+Wa_min+Wa_std+Ot_std,data=turbine.
train)
summary(subset_fit)

##
## Call:
## lm(formula = P_avg ~ Ws_avg + Ws_min + Ws_std + Wa_avg + Wa_min +
##     Wa_std + Ot_std, data = turbine.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -726.35  -89.84  -29.40   57.52  830.64
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -321.79735    5.22120 -61.633  < 2e-16 ***
## Ws_avg        62.74905    3.93474  15.947  < 2e-16 ***
## Ws_min        30.96531    4.27814   7.238 4.90e-13 ***
## Ws_std       229.86584    9.77834  23.508  < 2e-16 ***
## Wa_avg         0.10961    0.01425   7.689 1.62e-14 ***
## Wa_min        -0.11676    0.01500  -7.786 7.61e-15 ***
## Wa_std         0.23088    0.02863   8.064 8.25e-16 ***
## Ot_std       138.74421   26.28783   5.278 1.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 133.8 on 9592 degrees of freedom
## Multiple R-squared:  0.7222, Adjusted R-squared:  0.722
## F-statistic:  3563 on 7 and 9592 DF,  p-value: < 2.2e-16
```

from the summary table, we see all variables are statistically significant. RSE value is 133.8

```
#prediction on test data

pred1 <- predict(subset_fit, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##        RMSE          R2
## 144.2957393   0.7278593
```

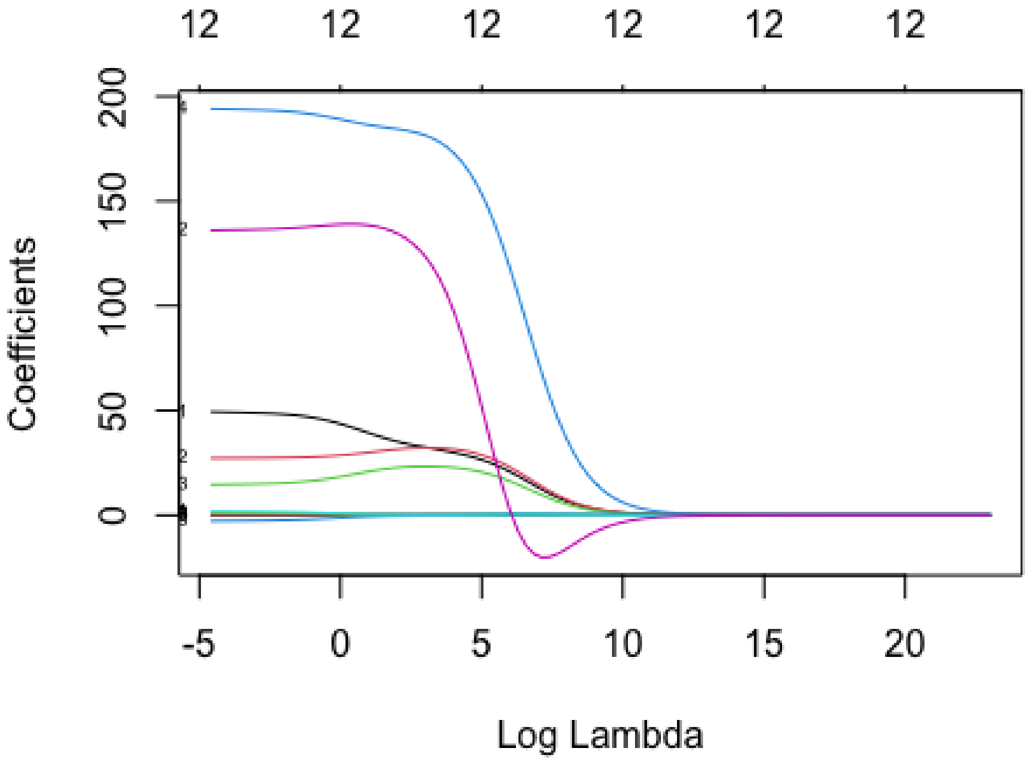
Question: what is the TEST RMSE, after fitting the model using 7 best features?

Test RMSE is **144** and R2 value is 71%

Ridge regression

I use the glmnet() function in R with parameter alpha=0 to fit a ridge regression model. I make use of other R functions to prepare the data for ridge regression.

```
library(ISLR)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-4

x=model.matrix(P_avg~.-1,data=turbine.train)
y=turbine.train$P_avg
#for test data
x_test=model.matrix(P_avg~.-1,data=turbine.test)
y_test=turbine.test$P_avg
#alpha=0 for Ridge regression
grid =10^ seq (10 , -2 , length =100)
fit.ridge=glmnet(x,y,alpha=0, lambda = grid)
summary(fit.ridge)

##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1200   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric

plot(fit.ridge, xvar="lambda", label=TRUE)
```

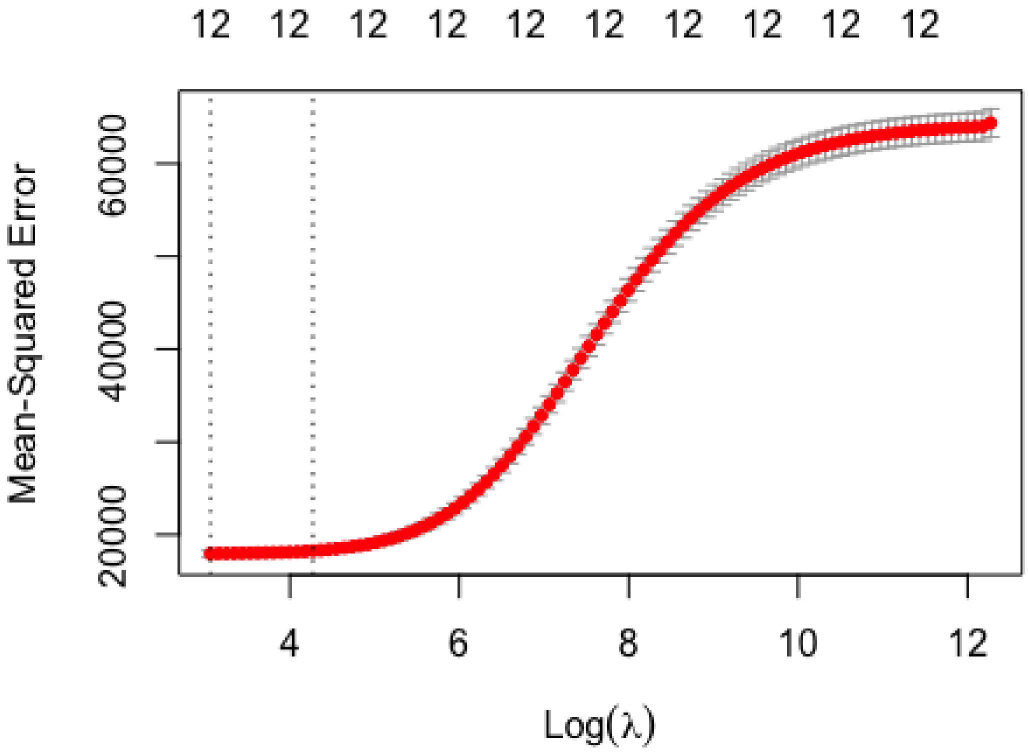


Question:What does coefficient path display in ridge regression?

The coefficient path displays the relationship between λ, the penalty parameter, the number of active variables, and their estimated coefficients.Each point along the coefficient path represents a different lasso model. For example, if log(λ)=8, the model characterized by this penalty contains only 3 variables.

Now,fit a ridge regression model on a training data set.

```
#use cross validation to choose the tuning parameter lambda
cv.out = cv.glmnet(x,y,alpha =0)
plot(cv.out)
```



Question: how do you choose the tuning parameter in ridge regression?

I selected lambda using cross validation.Here i have chosen to implement the function over a grid of values ranging from lambda = 0.01 to lambda = 100, essentially covering the full range of scenarios from the null model containing only the intercept, to the least squares fit.

```
bestlam = cv.out$lambda.min
bestlam

## [1] 21.37684
```

Question: how do you choose the tuning parameter in ridge regression?

The selected value of λ is 21.82054.

I refit the model and produce predictions for this value of lambda on the test data. This gives the resulting test MSE:

```
#fit again using bestlambda
fit.ridge=glmnet(x,y,alpha=0, lambda = bestlam)
# prediction on test data
ridge.pred = predict (fit.ridge , s = bestlam , newx = x_test)
# test RMSE
sqrt(mean (( ridge.pred - y_test ) ^2))

## [1] 145.0549

# test R2
cor(ridge.pred, y_test) ^ 2
```

```
##          [,1]
## s1 0.7270793
```

Question: What is the value of test RMSE after using tuning parameter (best lamda) in the model?

Here the test RMSE is 145 and R2 value is 72%

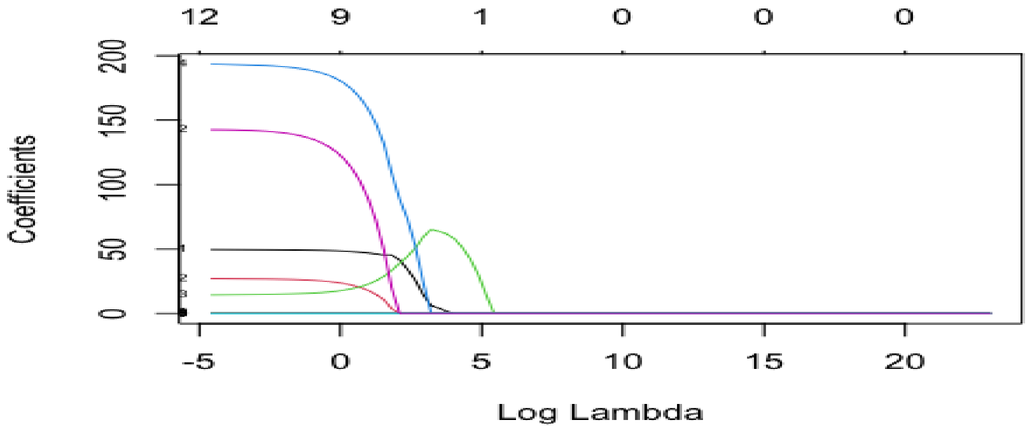
Lasso regression

Lasso works similar to ridge regression with the difference that lasso has a penalty of the sum of the absolute values of the coefficients. The penalty for ridge regression is the sum of the squares of the coefficients. I will use glmnet also for lasso but with alpha parameters set to 1.

```
x=model.matrix(P_avg~.-1,data=turbine.train)
y=turbine.train$P_avg
#for test data
x_test=model.matrix(P_avg~.-1,data=turbine.test)
y_test=turbine.test$P_avg
#alpha=1 for Lasso regression
grid =10^ seq (10 , -2 , length =100)
fit.lasso=glmnet(x,y,alpha=1, lambda = grid)
summary(fit.lasso)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1200   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```

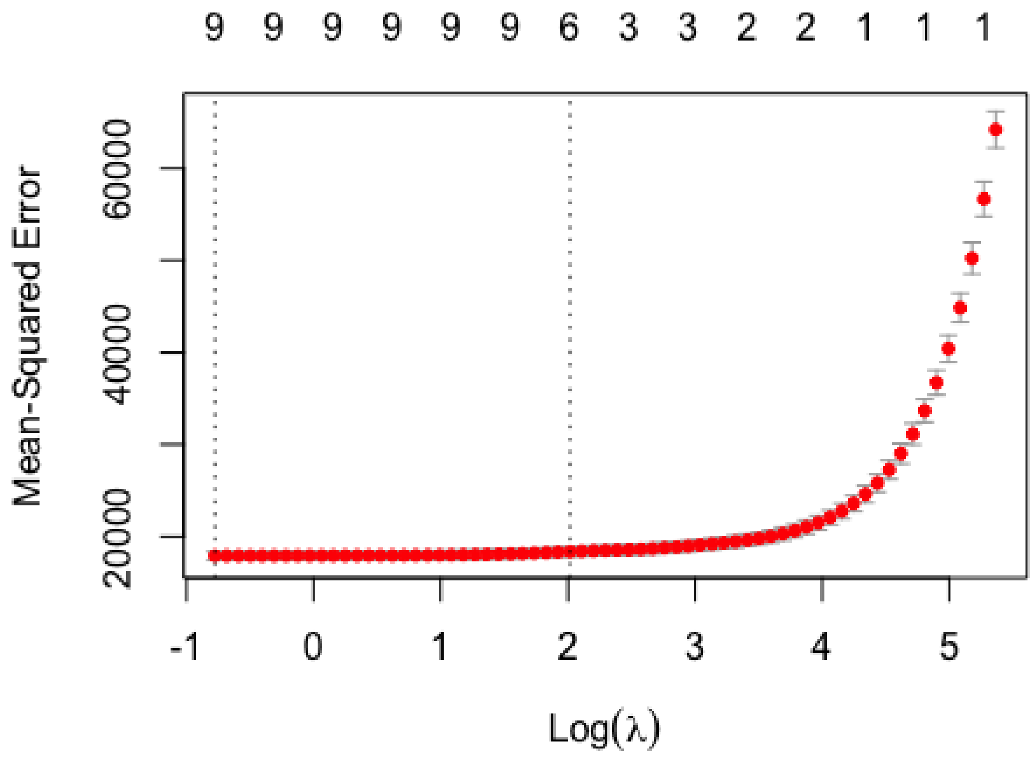
```
plot(fit.lasso, xvar="lambda", label=TRUE)
```



Question: What does coefficient path display in Lasso regression?

The coefficient path displays the relationship between λ, the penalty parameter, the number of active variables, and their estimated coefficients.Each point along the coefficient path represents a different lasso model.

```
#use cross validation to choose the tuning parameter lambda
cv.out = cv.glmnet(x,y,alpha =1)
plot(cv.out)
```



Question: what is the value of tuning parameter in Lasso and how do i select it?

I test varying values of λ (from 0.01 to 100) using cross-validation. The above plot displays the cross-validation error according to the log of lambda. The vertical dash line indictaes that the log of the optimal value of lambda is approximately 1.8, which minimizes the prediction error. This lamdba value will give the most accuract model.

```
bestlam = cv.out$lambda.min
bestlam

## [1] 0.4605501
```

The selected value of λ is 0.460.Fitting the full model, and evaluating the test RMSE:

```
#fit again using bestlambda
fit.lasso=glmnet(x,y,alpha=1, lambda = bestlam)
# prediction on test data
lasso.pred = predict (fit.lasso , s = bestlam , newx = x_test)
# test RMSE
sqrt(mean (( lasso.pred - y_test ) ^2))

## [1] 144.3278
```

```
# test R2
cor(lasso.pred, y_test) ^ 2

##         [,1]
## s1 0.7278969
```

Question: What is the value of Test RMSE for Lasso regression model?

The test rmse for lasso is 144 that is greater than ridge regression model.

Dimension Reduction Models

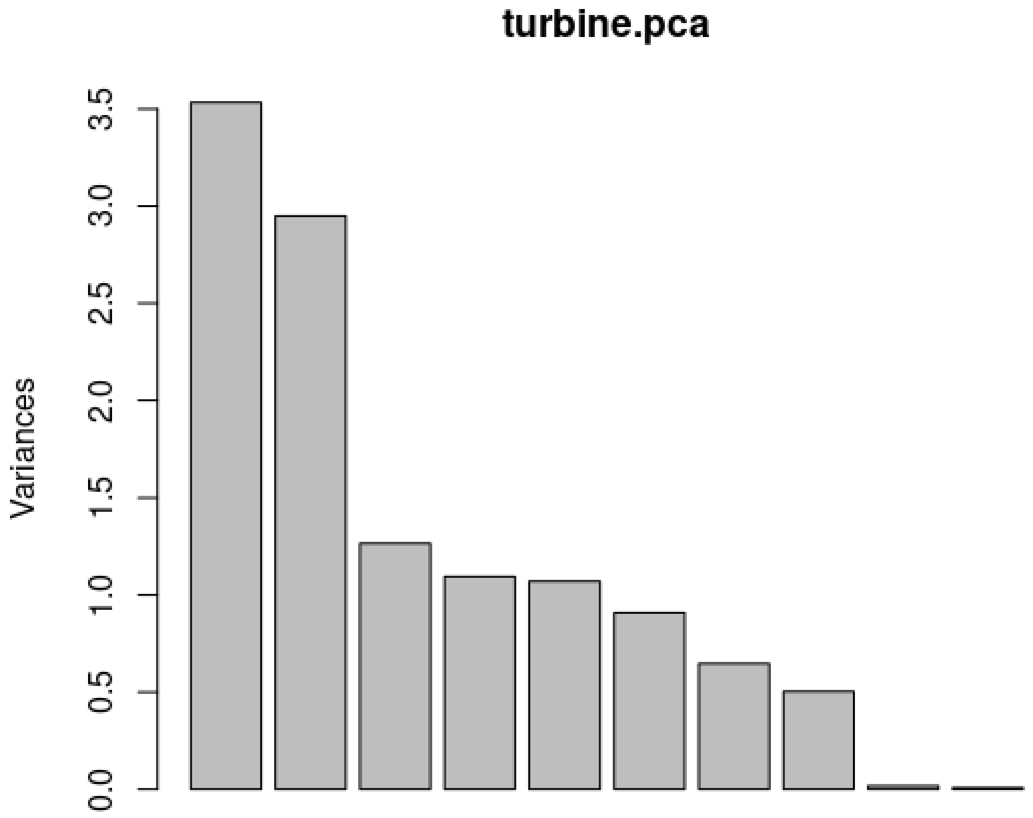
There are two dimension reduction methods we carry out in this session. These methods are principal components regression and partial least squares method.

Principal Component Regression

Principal Components Regression (PCR) can be performed using the pcr() function in R.

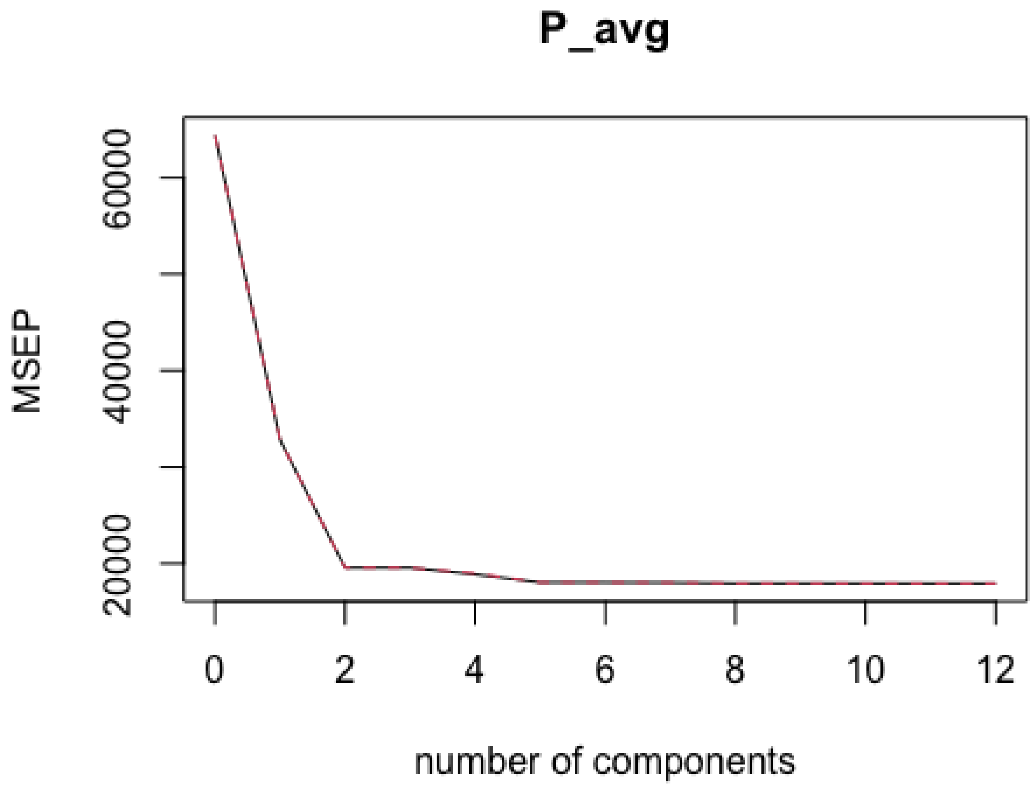
```
# Principal Components Regression
library(pls)

## 
## Attaching package: 'pls'

## The following object is masked from 'package:caret':
## 
##     R2

## The following object is masked from 'package:stats':
## 
##     loadings

set.seed(2)
pcr.fit=pcr(P_avg~., data=turbine.train,scale=TRUE,validation="CV")
summary(pcr.fit)

## Data:    X dimension: 9600 12 
##  Y dimension: 9600 1
## Fit method: svdpc
## Number of components considered: 12
## 
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           253.8    181.1    139.8    139.9    137.4    134.3    134.3
## adjCV        253.8    181.1    139.8    139.8    137.8    134.3    134.3
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
## CV       134.3      134      134       134       134     134.0
## adjCV    134.2      134      134       134       134     133.9
## 
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X        29.31    54.00    64.57    73.63    82.65    90.16    95.59    99.79
## P_avg    49.09    69.68    69.69    70.46    72.05    72.07    72.11    72.22
##        9 comps  10 comps  11 comps  12 comps
## X        99.93    100.00    100.00    100.00
## P_avg    72.22     72.26     72.26     72.28
```

Before doing PCR, Data need to be scaled, by scale = True, i scaled the data.The summary call displays the PCR component coficient and percentage of variance explained by the components.If i select 7 components , that will be expained arroun 95.57% variance of the data. For M =1 only captures 29.36% of all variance or information in the predictors. In contrast, using M=7 increases the value to 95.57%



The following figure shows the plots of projected points on principal components. It is very clear that projected points on PC1 clearly classify the data but the plots of projected points by lower principal components (for PC2, PC3 & PC4) is not able to classify the data as convincingly as PC1

```r
# PCR -M selection using cross validation
validationplot(pcr.fit,val.type="MSEP")
```



Question: how many components should i select for modeling stage ?

Cross-validation selected M= 12 (so we can note that M=p=12) - the dimensionality hasn’t been reduced at all. Because the lowest cross-validation error occurs when M=12 (135.4) But If I select M=7 that will explained 95% variance of the data. Now fit model with M=7 and evaluate the test MSE

```
# fit model using 7 PCA that explained 95% variance of the data
pcr7.fit = pcr (P_avg~., data=turbine.train,scale = TRUE ,ncomp =7)
# Prediction test data
#remove the target variable from turbine test set
test.features = subset(turbine.test, select=-c(P_avg))
#ground truth value
test.target = subset(turbine.test, select=P_avg)[,1]
#prediction
pcr.pred = predict( pcr7.fit , test.features , ncomp =7)
# TEST RMSE
sqrt(mean((test.target - pcr.pred)^2))

## [1] 144.7961

# Test R2
cor(test.target, pcr.pred) ^ 2

## [1] 0.7259731
```

Question: what is the test RMSE for Principal component regression model?

So using PCR test RMSE is 138 and R2 value is 72%

Partial Least Squares(PLS regression)

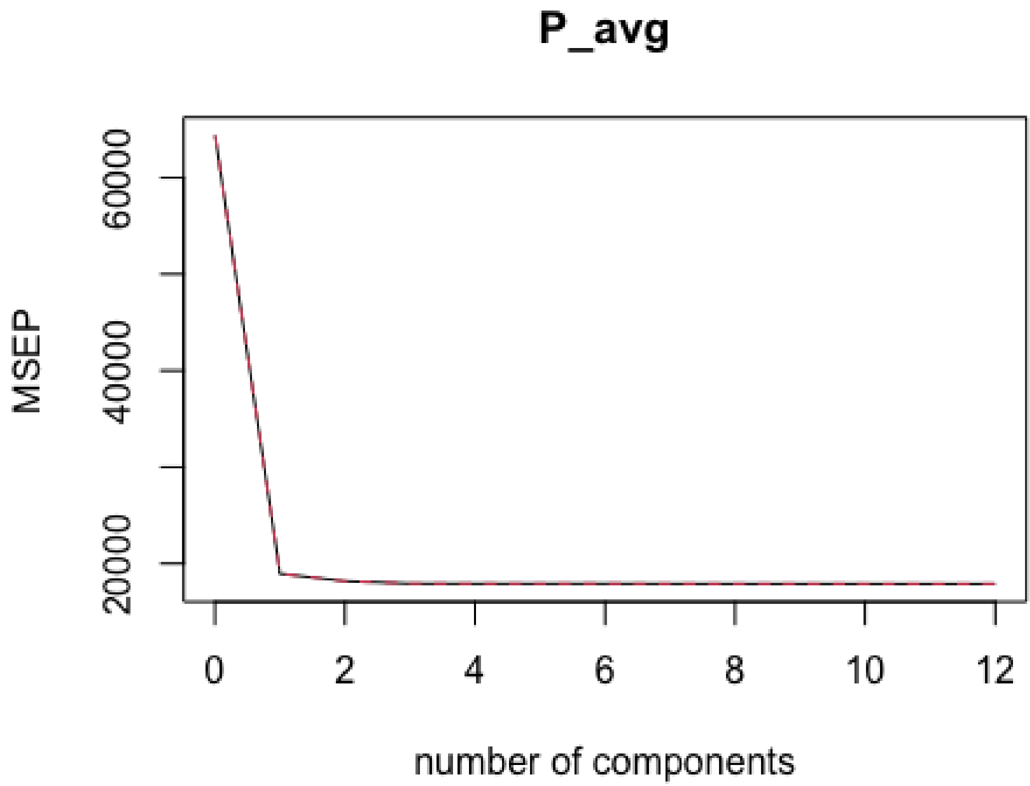
I implement PLS using plsr() function in the R.

```
pls.fit=plsr(P_avg~., data=turbine.train,scale=TRUE,validation="CV")
summary(pls.fit)
```

```
## Data:    X dimension: 9600 12
##  Y dimension: 9600 1
## Fit method: kernelpls
## Number of components considered: 12
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           253.8    137.6    134.8    133.9    133.9    133.9    133.9
## adjCV        253.8    137.6    134.8    133.9    133.9    133.9    133.9
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
## CV       133.8    133.8    133.8     133.8     133.8     133.8
## adjCV    133.8    133.8    133.8     133.8     133.8     133.8
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X        28.18    49.64    62.85    67.65    75.73    82.57    85.23    90.91
## P_avg    70.62    71.82    72.20    72.22    72.22    72.23    72.25    72.25
##        9 comps  10 comps  11 comps  12 comps
## X        99.42    100.00    100.00    100.00
## P_avg    72.26     72.26     72.27     72.28
```

Before doing PLS, Data need to be scaled, by scale = True, i scaled the data.The summary call displays the PCR component coficient and percentage of variance explained by the components.If i select 7 components , that will be expained arroun 84.74% variance of the data that is less than PCR component explained. For M =1 only captures 28.19% of all variance or information in the predictors. In contrast, using M=9 increases the value to 99.78%

```
validationplot(pls.fit,val.type="MSEP")
```



Question: how many components should i select for modeling stage ?

Cross-validation selected M= 12 (so we can note that M=p=12) - the dimensionality hasn't been reduced at all. Because the lowest cross-validation error occurs when M=12 (136.5) But If I select M=7 that will explained 84% variance of the data but i will compare the test RMSE with PCR test rmse. Now fit model with M=7 and evaluate the test RMSE.

```
# fit model using 7 PLS that explained 84% variance of the data
pls7.fit = plsr (P_avg~., data=turbine.train,scale = TRUE ,ncomp =7)
# Prediction test data
#remove the target variable from turbine test set
test.features = subset(turbine.test,scale = TRUE, select=-c(P_avg))
#ground truth value
test.target = subset(turbine.test,scale = TRUE, select=P_avg)[,1]
#prediction
pls.pred = predict( pls7.fit , test.features ,scale = TRUE, ncomp =7)
# TEST RMSE
sqrt(mean((test.target - pls.pred)^2))

## [1] 144.3506

# Test R2
cor(test.target, pls.pred) ^ 2

## [1] 0.7276497
```

Question: what is the test rmse for PLS fitted model?

So using PLS, The test RMSE with M=7 component is 133.86 that is less than PCR test RMSE 138

Chapter 5- Non Linear Model

- Why non linear analyses?
- Poly nominal regression
- splines with knot
- smoothing splines
- Local Regression
- Generalized Additive Model
- GAM with smoothing spline
- GAM using natural spline
- Research Question-

29. Is polynomial regression of wind speed with all degrees statistically significant?
30. how do one decide the degree of freedom in polynomial regression?
31. Does model with 4 degree show lower test RMSE than model with 2 degrees of freedom?
32. Are all knots statistically significant in spline analysis?
33. How do you select tunning parameter and what is the value of it in smoothing spline model?
34. What is the test rmse for local regression?
35. Does GAM model with natural spline show statistically significant for higher degree of freedom?
36. which is beter, GAM with smoothing spline or GAM with natural spline?

Why non linear analyses

Linear models are advantageous when it comes to their interpretability. However, their capabilities are limited, especially in scenarios where the linear assumption is poor. Ridge, lasso, and principal components regression improve upon the least squares regression model by reducing the variance of the coefficient estimates. However, these models are still linear, and will perform poorly in nonlinear settings. Here, I will analyse the polynomial regression, splines, local regression, and generalized additive models.Polynomial regression extends the linear model by adding extra pre-dictors, obtained by raising each of the original predictors to a power.For example, a cubic regression uses three variables, X, X 2 , and X 3 ,as predictors. This approach provides a simple way to provide a non-linear fit to data.Spline regression. Fits a smooth curve with a series of polynomial segments. The values delimiting the spline segments are called Knots.Local regression is yet another approach to fitting flexible non-linear functions. It involves determining a fit at some target point using only the nearby training observations.Generalized additive models (GAMs) allow for non-linear functions of multiple predictors, while maintaining additivity. Additivity simply refers to the ability to add together each predictor's contribution.

Polynomial regression

```
cor(Ws_avg, P_avg) # correlation between wind speed and power

## [1] 0.8288955
```

#wind speed with target variable (power) has highest correlation and it’s 0.82, so here as polynomial regression analysis, i will take wind speed as a feature variable

```
# polynomial regression
model_poly = lm(P_avg ~ poly(Ws_avg, 5), data = turbine.train)
summary(model_poly)

##
## Call:
## lm(formula = P_avg ~ poly(Ws_avg, 5), data = turbine.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -949.22   -9.85    1.75    8.48  260.61
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         165.2012     0.4763 346.862  < 2e-16 ***
## poly(Ws_avg, 5)1 20590.4073    46.6650 441.238  < 2e-16 ***
## poly(Ws_avg, 5)2 12870.8219    46.6650 275.813  < 2e-16 ***
## poly(Ws_avg, 5)3   144.2779    46.6650   3.092  0.504
## poly(Ws_avg, 5)4 -2744.2310    46.6650 -58.807  < 2e-16 ***
## poly(Ws_avg, 5)5   126.4963    46.6650   2.711  0.00673 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.67 on 9594 degrees of freedom
## Multiple R-squared:  0.9662, Adjusted R-squared:  0.9662
## F-statistic: 5.485e+04 on 5 and 9594 DF,  p-value: < 2.2e-16
```

Question: Is polynomial regression of wind speed with all degrees statistically significant?

From the summary table, we see polynomial regression with degree 3 is not statistically significant.

Question: how do one decide the degree of freedom in polynomial regression?

For deciding a degree I can use hypothesis tests(e.g. ANOVA) to test nested sequence of models.

```
poly_1 = lm(P_avg~Ws_avg, data = turbine.train)
poly_2 = lm(P_avg~poly(Ws_avg,2), data = turbine.train)
poly_3 = lm(P_avg~poly(Ws_avg,3), data = turbine.train)
poly_4 = lm(P_avg~poly(Ws_avg,4), data = turbine.train)
poly_5 = lm(P_avg~poly(Ws_avg,5), data = turbine.train)
print(anova(poly_1,poly_2,poly_3,poly_4,poly_5))

## Analysis of Variance Table
##
## Model 1: P_avg ~ Ws_avg
## Model 2: P_avg ~ poly(Ws_avg, 2)
## Model 3: P_avg ~ poly(Ws_avg, 3)
## Model 4: P_avg ~ poly(Ws_avg, 4)
## Model 5: P_avg ~ poly(Ws_avg, 5)
##   Res.Df       RSS Df Sum of Sq          F    Pr(>F)
## 1   9598 194117818
## 2   9597  28459763  1 165658055 76072.7837 < 2.2e-16 ***
## 3   9596  28438947  1     20816     9.5591  0.001995 **
## 4   9595  20908143  1   7530804  3458.2634 < 2.2e-16 ***
## 5   9594  20892142  1     16001     7.3481  0.006725 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 2 is statiscially signifcant than model 3 with degree of 3. But For higher degree like (degree, 4,5), both model is also significant. I will take model 2 with degree 2 as the model with higher polynomial degree is more difficult to explain.

```
# fit a polynomial directly using degree 4 that selected by ANOVA
model_poly_fit=lm(P_avg~I(Ws_avg^4), data=turbine.train)

#prediction on test data
pred1 <- predict(model_poly_fit, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##        RMSE          R2
## 103.9565208   0.8630787
```

The model 4 Test RMSE is 103, now see the test RMSE for taking the model 2

```
# fit a polynomial directly using degree 2 that selected by ANOVA
model_poly_fit=lm(P_avg~I(Ws_avg^2), data=turbine.train)

#prediction on test data
pred1 <- predict(model_poly_fit, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##       RMSE         R2
## 83.5027146  0.9084945
```

Question: Does model with 4 degree show lower test RMSE than model with 2 degrees of freedom?

No, Here, i see model 2 Test RMSE (83.50) which is less than test RMSE of model 4(103). It means if i take model4, it will overfit the data as a result we will get the high variance result on test data.

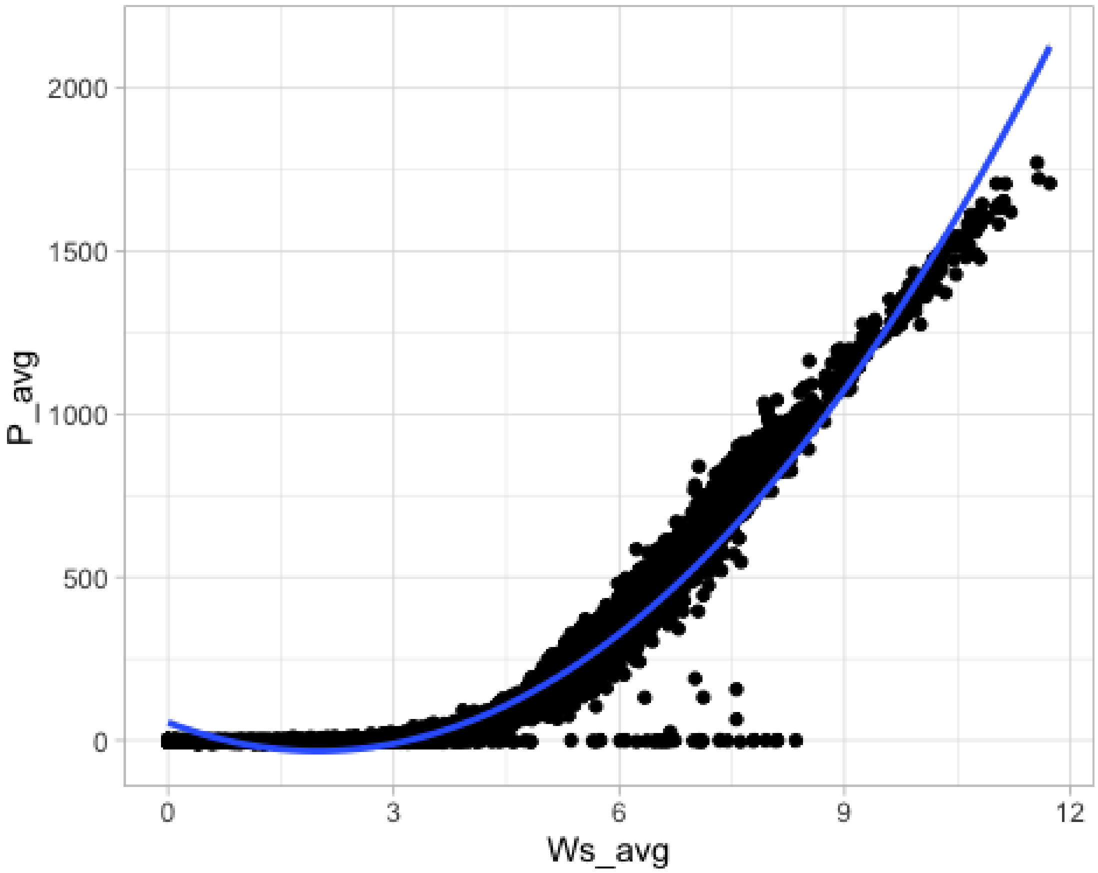
Spline with Knots

```r
library(splines)
library(tidyverse)

knots <- quantile(turbine.train$Ws_avg, p = c(0.25, 0.5, 0.75))
model <- lm (P_avg ~ bs(Ws_avg, knots = knots), data = turbine.train)
summary(model)

##
## Call:
## lm(formula = P_avg ~ bs(Ws_avg, knots = knots), data = turbine.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -946.45   -5.92    0.57    7.67  255.25
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.920      2.561  -0.750   0.4535
## bs(Ws_avg, knots = knots)1     3.228      5.301   0.609   0.5426
## bs(Ws_avg, knots = knots)2    -7.169      3.468  -2.067   0.0388 *
## bs(Ws_avg, knots = knots)3    13.496      3.310   4.077 4.59e-05 ***
## bs(Ws_avg, knots = knots)4   462.912      4.598 100.679  < 2e-16 ***
## bs(Ws_avg, knots = knots)5  1471.884     10.333 142.441  < 2e-16 ***
## bs(Ws_avg, knots = knots)6  1736.508     12.498 138.947  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.32 on 9593 degrees of freedom
## Multiple R-squared:  0.9667, Adjusted R-squared:  0.9667
## F-statistic: 4.641e+04 on 6 and 9593 DF,  p-value: < 2.2e-16
```

Question: Are all knots statistically significant?

From the summary , all knots are not significant, but overall RSE is decresead than any other model.

```r
#plot
ggplot(turbine.train, aes(Ws_avg, P_avg) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ splines::bs(x, df = 2))
```



The plot shows the predicted result with spline using knots

```
#prediction on test data
pred1 <- predict(model, newdata = turbine.test)

## Warning in bs(Ws_avg, degree = 3L, knots = c(`25%` = 2.49, `50%` = 4.3099999, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##      RMSE         R2
## 51.6694398  0.9648638
```

Question: What is the Test rmse for spline with knots?

Test RMSE is 51 for spline with knots

Smoothing spline

```
model_spline = smooth.spline(y = turbine.train$P_avg,
                             x = turbine.train$Ws_avg,
                             lambda = 0.01)
model_spline

## Call:
## smooth.spline(x = turbine.train$Ws_avg, y = turbine.train$P_avg,
##     lambda = 0.01)
##
## Smoothing Parameter  spar= NA  lambda= 0.01
## Equivalent Degrees of Freedom (Df): 10.79028
## Penalized Criterion (RSS): 4447481
## GCV: 2146.537

#model2, different lamda
mod_ss2 = smooth.spline(y = turbine.train$P_avg,
                        x = turbine.train$Ws_avg,
                        lambda = 1)
mod_ss2

## Call:
## smooth.spline(x = turbine.train$Ws_avg, y = turbine.train$P_avg,
##     lambda = 1)
##
## Smoothing Parameter  spar= NA  lambda= 1
## Equivalent Degrees of Freedom (Df): 4.02715
## Penalized Criterion (RSS): 7952839
## GCV: 2508.961
```

Question? what is the effective degree of freedom of smoothing spline model?

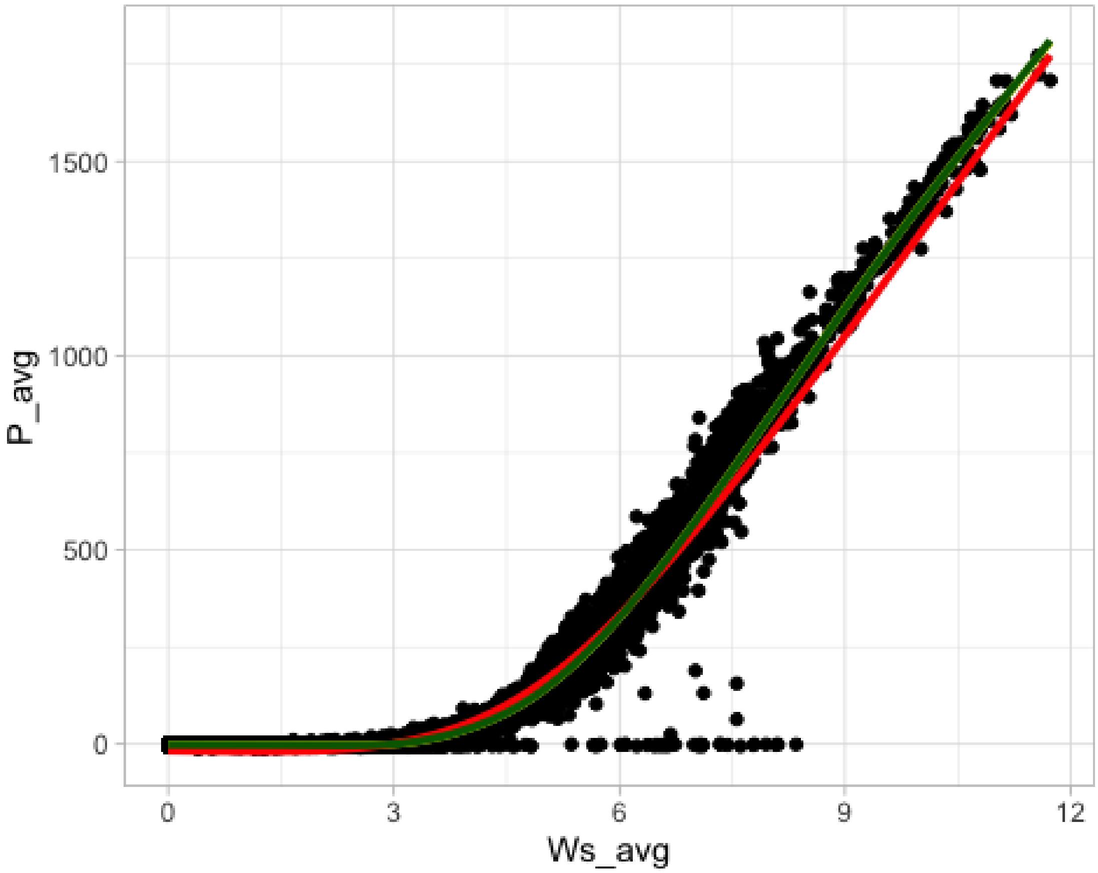
Two smothing spline model with lambda = 0.001, and lambda=1. But effective degree of freedom for model1 is 10.9044 that is higher the model 2 where value is 4.062176. Effective degree of freedom determines the smoothing of wiggle. The higher the effective degree of freedom the more the curve is wiggly in nature.Variables which have effective degree of freedom (edf) close to 1 have linear relationship with power (P_avg).now select tuning parameter using cross validation method.

```
#cross-validation to select the best lamda for model flexibility.
mod_ss3 = smooth.spline(y = turbine.train$P_avg,
                        x = turbine.train$Ws_avg,
                        cv = T)

## Warning in smooth.spline(y = turbine.train$P_avg, x = turbine.train$Ws_avg, :
## cross-validation with non-unique 'x' values seems doubtful

mod_ss3

## Call:
## smooth.spline(x = turbine.train$Ws_avg, y = turbine.train$P_avg,
##     cv = T)
##
## Smoothing Parameter  spar= 0.8413018  lambda= 0.01241127 (11 iterations)
## Equivalent Degrees of Freedom (Df): 10.27191
## Penalized Criterion (RSS): 4451281
## PRESS(l.o.o. CV): 2150.278

mod_ss3$lambda

## [1] 0.01241127
```

QuestionHow do you select tunning parameter and what is the value of it in smoothing spline model?

Tunning parameter is selected by cross validation approach. Here the value of tunning parameter lambda is 0.0124117 where effective degree of freedom is 10.27. Now plot the three model.

```
# plot 3 fitted model

turbine.train %>%
  mutate(pred1 = fitted(model_spline),
         pred2 = fitted(mod_ss2),
         pred3 = fitted(mod_ss3)) %>%
  ggplot(aes(Ws_avg, P_avg)) +
  geom_point() +
  geom_line(aes(y=pred1),col="yellow",lwd=1.2) +
  geom_line(aes(y=pred2),col="red",lwd=1.2) +
  geom_line(aes(y=pred3),col="darkgreen",lwd=1.2)
```



darkgreen indicates the model 3 with tuning parameter lamda = 0.0124117 . Now see on the test data

```
#prediction on test data for best fiited mod_ss3
ss_pred = predict(mod_ss3, turbine.test$Ws_avg)
# Model performance
ss_perf = data.frame(
  RMSE = sqrt(sum((ss_pred$y- turbine.test$P_avg)^2)/length(turbine.test$P_avg)),
  COR = cor(ss_pred$y, turbine.test$P_avg)
)
ss_perf

##       RMSE       COR
## 1 51.44274 0.9824314
```

The test RMSE for smoothing spline is 51.44

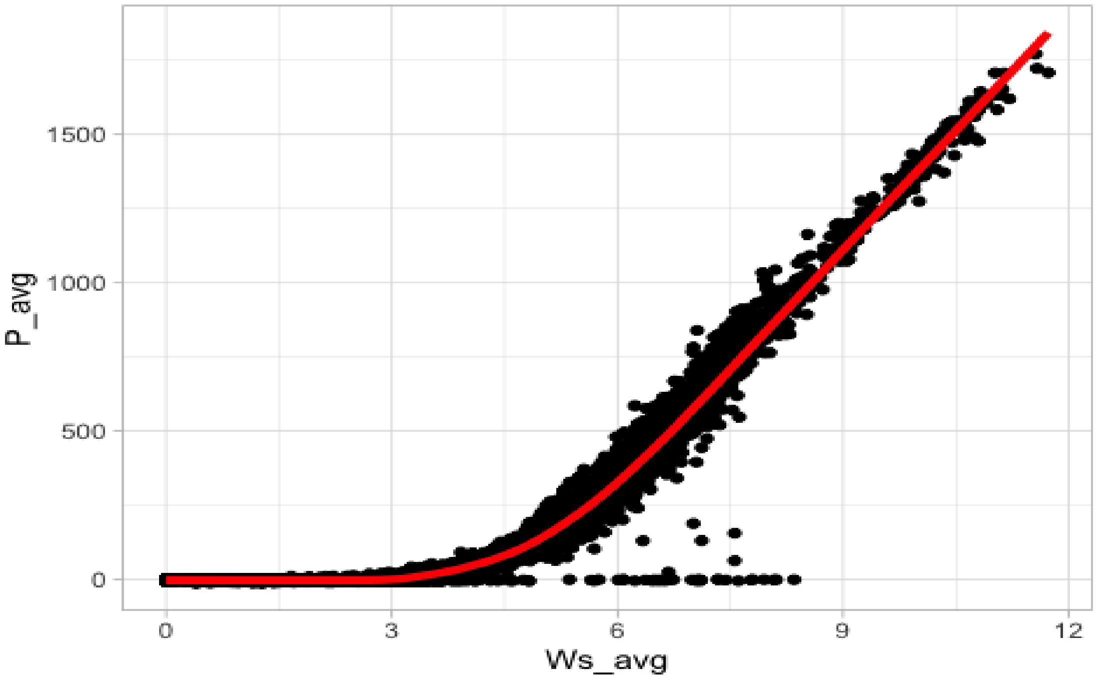
local regression

```
#loess: local regression, span controls the degree of smoothing
mod_loess = loess(P_avg ~ Ws_avg,span = 0.2,degree = 1,data = turbine.train)
summary(mod_loess)
```

```
## Call:
## loess(formula = P_avg ~ Ws_avg, data = turbine.train, span = 0.2,
##     degree = 1)
##
## Number of Observations: 9600
## Equivalent Number of Parameters: 9.47
## Residual Standard Error: 46.37
## Trace of smoother matrix: 11.23  (exact)
##
## Control settings:
##   span     :  0.2
##   degree   :  1
##   family   :  gaussian
##   surface  :  interpolate      cell = 0.2
##   normalize:  TRUE
##  parametric:  FALSE
## drop.square:  FALSE
```

Comment: here span is ised for local regression s =0.2 and for training data residual standard error is 46.37

```
#plot
turbine.train %>%
  mutate(pred = mod_loess$fitted) %>%
  ggplot(aes(Ws_avg, P_avg)) +
  geom_point() +
  geom_line(aes(Ws_avg,pred), col="red", lwd=1.5)
```



The graphs shows the how train data is fitted with local non linear regression model. Now test the model on test data.

```
#prediction on test data
pred1 <- predict(mod_loess, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)
```

```
## RMSE    R2
## 49.73487  0.982413
```

Question: what is the value of Test rmse for local regression model?

The test RMSE for Local regression is 49.73

Generalized Additive Model

Generalized additive models (GAMs) presents framwork to extending a standard linear model by allowing non-linear functions of each variables while maintaining additivity.I use the gam() function in the gam package in R. One important feature of gam is its ability to allow non-linearity on multiple variables at the sametime. GAM can be used with other non-linear functions.

```
# library gam package
library(gam)
```

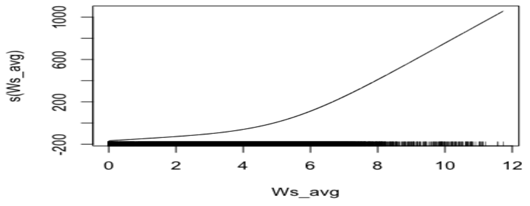
First i fit the model using smoothing splines and predic on test data. Then i fit the model using natural spline and predic on test data.

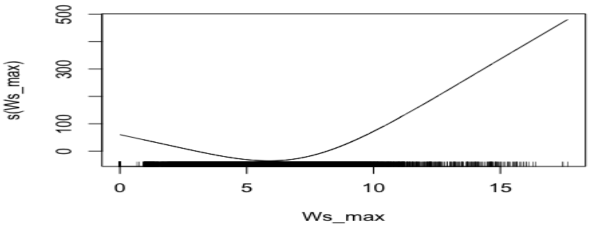
Gam with smoothing spline

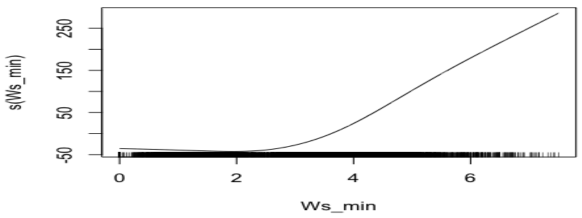
```
#smoothing spline fit with GAM
turbine.gam <- gam(P_avg ~ s(Ws_avg) + s(Ws_max) + s(Ws_min) + s(Ws_std) +
s(Wa_avg) +
                     s(Wa_max) + s(Wa_std) + s(Wa_min) + s(Ot_avg) + s(Ot_max) +
s(Ot_min) + s(Ot_std), data = turbine.train)
summary(turbine.gam)

##
## Call: gam(formula = P_avg ~ s(Ws_avg) + s(Ws_max) + s(Ws_min) + s(Ws_std) +
##     s(Wa_avg) + s(Wa_max) + s(Wa_std) + s(Wa_min) + s(Ot_avg) +
##     s(Ot_max) + s(Ot_min) + s(Ot_std), data = turbine.train)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -774.84  -10.64    1.37   11.29  268.16
##
## (Dispersion Parameter for gaussian family taken to be 1902.66)
##
##     Null Deviance: 618082690 on 9599 degrees of freedom
## Residual Deviance: 18172303 on 9551 degrees of freedom
## AIC: 99784.17
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##             Df    Sum Sq   Mean Sq    F value    Pr(>F)
## s(Ws_avg)    1 424908133 424908133 2.2332e+05 < 2.2e-16 ***
## s(Ws_max)    1      2043      2043 1.0740e+00  0.300067
## s(Ws_min)    1    977656    977656 5.1384e+02 < 2.2e-16 ***
## s(Ws_std)    1     20156     20156 1.0594e+01  0.001139 **
## s(Wa_avg)    1     59672     59672 3.1362e+01 2.200e-08 ***
## s(Wa_max)    1     14894     14894 7.8280e+00  0.005154 **
## s(Wa_std)    1         0         0 0.0000e+00  0.997120
## s(Wa_min)    1     50402     50402 2.6490e+01 2.701e-07 ***
## s(Ot_avg)    1     88188     88188 4.6350e+01 1.049e-11 ***
## s(Ot_max)    1     30965     30965 1.6274e+01 5.522e-05 ***
## s(Ot_min)    1       430       430 2.2580e-01  0.634692
## s(Ot_std)    1      5496      5496 2.8883e+00  0.089255 .
## Residuals 9551  18172303      1903
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##             Npar Df Npar F     Pr(F)
## (Intercept)
## s(Ws_avg)         3 7390.9 < 2.2e-16 ***
## s(Ws_max)         3 3603.3 < 2.2e-16 ***
## s(Ws_min)         3 1494.0 < 2.2e-16 ***
## s(Ws_std)         3  303.6 < 2.2e-16 ***
## s(Wa_avg)         3    8.8 7.929e-06 ***
## s(Wa_max)         3    5.5 0.0008711 ***
## s(Wa_std)         3    2.3 0.0747944 .
## s(Wa_min)         3    7.9 2.973e-05 ***
## s(Ot_avg)         3    5.4 0.0010436 **
## s(Ot_max)         3    4.6 0.0034187 **
## s(Ot_min)         3    7.9 2.794e-05 ***
## s(Ot_std)         3    3.6 0.0122031 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

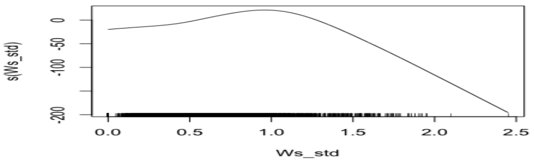
summary indicates the significant value for gam model, all are not statistically significant.

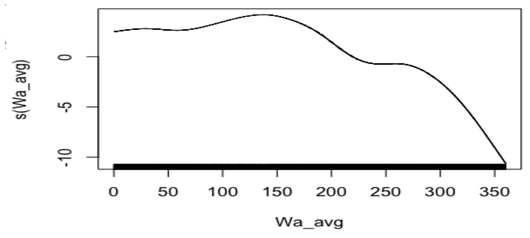
```
#plot
plot(turbine.gam, shade = TRUE, seWithMean = TRUE, scale = 0)
```

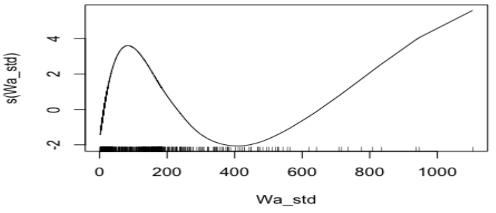


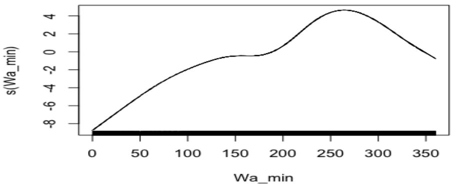


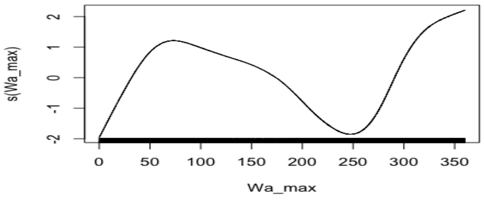


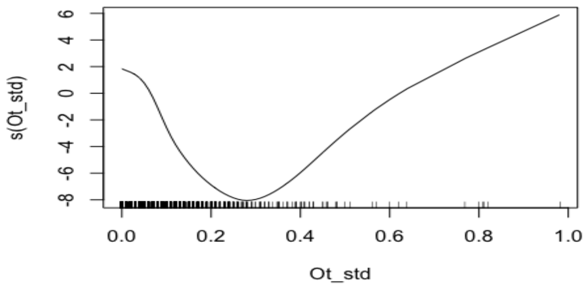


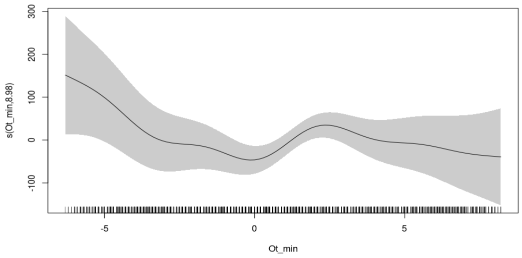


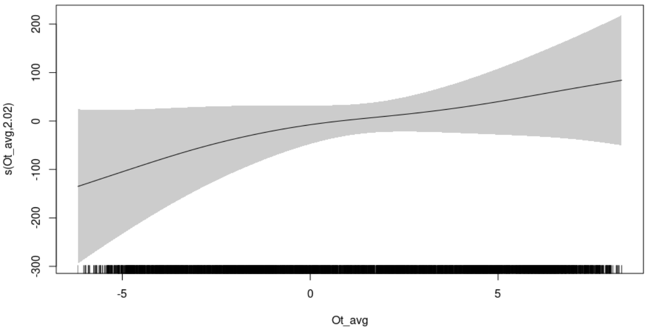












From the plots above, can see that fitting non-linearity on most of the variables is sufficient, but Non-linearity isn't sufficint for all variables.

```
#prediction
pred1 <- predict(turbine.gam, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##       RMSE         R2 
## 48.6159380  0.9688615
```

Question : what is the test RMSE value for GAM model (using smoothing spline)?

Test rmse is 48.61 for Gam model using smoothing spline.

<u>GAM model using natural spline</u>

```
#natural spline
turbine.gam <- lm(P_avg ~ ns(Ws_avg,df=2) + ns(Ws_max,df=2) + ns(Ws_min,df=2) +
ns(Ws_std,df=2) + ns(Wa_avg,df=2) +ns(Wa_max,df=2) + ns(Wa_std,df=2) + ns(Wa_min,
df=2) +
                     ns(Ot_avg, df=2) + ns(Ot_max,df=2) + ns(Ot_min,df=2) +
ns(Ot_std,df=2), data = turbine.train)
summary(turbine.gam)

## 
## Call:
## lm(formula = P_avg ~ ns(Ws_avg, df = 2) + ns(Ws_max, df = 2) + 
##     ns(Ws_min, df = 2) + ns(Ws_std, df = 2) + ns(Wa_avg, df = 2) + 
##     ns(Wa_max, df = 2) + ns(Wa_std, df = 2) + ns(Wa_min, df = 2) + 
##     ns(Ot_avg, df = 2) + ns(Ot_max, df = 2) + ns(Ot_min, df = 2) + 
##     ns(Ot_std, df = 2), data = turbine.train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max
## -775.91  -16.77    1.19   17.59  275.72
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           42.910      4.015  10.686  < 2e-16 ***
## ns(Ws_avg, df = 2)1  948.345     26.132  36.290  < 2e-16 ***
## ns(Ws_avg, df = 2)2 1514.800     29.254  51.781  < 2e-16 ***
## ns(Ws_max, df = 2)1 -123.361     33.674  -3.663  0.00025 ***
## ns(Ws_max, df = 2)2  422.941     27.754  15.239  < 2e-16 ***
## ns(Ws_min, df = 2)1   96.470     13.631   7.077 1.57e-12 ***
## ns(Ws_min, df = 2)2  239.190     13.535  17.672  < 2e-16 ***
## ns(Ws_std, df = 2)1   75.541     13.689   5.518 3.51e-08 ***
## ns(Ws_std, df = 2)2 -236.089     15.315 -15.415  < 2e-16 ***
## ns(Wa_avg, df = 2)1  -15.327      6.753  -2.270  0.02325 *
## ns(Wa_avg, df = 2)2  -10.612      3.322  -3.195  0.00140 **
## ns(Wa_max, df = 2)1   -4.854      5.453  -0.890  0.37340
## ns(Wa_max, df = 2)2    4.431      2.317   1.913  0.05581 .
## ns(Wa_std, df = 2)1  -29.794      9.796  -3.042  0.00236 **
## ns(Wa_std, df = 2)2    7.326     23.235   0.315  0.75256
## ns(Wa_min, df = 2)1   13.519      4.729   2.859  0.00426 **
## ns(Wa_min, df = 2)2    1.356      3.179   0.426  0.66979
## ns(Ot_avg, df = 2)1  248.676    203.725   1.221  0.22225
## ns(Ot_avg, df = 2)2  212.459    109.841   1.934  0.05311 .
## ns(Ot_max, df = 2)1 -218.879    147.043  -1.489  0.13664
## ns(Ot_max, df = 2)2 -101.278     84.014  -1.205  0.22804
## ns(Ot_min, df = 2)1  -64.372    154.841  -0.416  0.67762
## ns(Ot_min, df = 2)2 -117.952     89.030  -1.325  0.18525
## ns(Ot_std, df = 2)1  -22.994     19.179  -1.199  0.23060
## ns(Ot_std, df = 2)2   -1.787     29.079  -0.061  0.95099
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.54 on 9575 degrees of freedom
## Multiple R-squared:  0.9664, Adjusted R-squared:  0.9664
## F-statistic: 1.149e+04 on 24 and 9575 DF,  p-value: < 2.2e-16
```

Question: Does GAM model with natural spline show statistically significant for higher degree of freedom?

All of the variable are not statistically significant for 2 degrees of freedom. Here from summary , it displays the Outdoor temperature (Ot) are not statistically significant for df=2.

```
#prediction
pred1 <- predict(turbine.gam, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)
```

```
##       RMSE         R2
## 51.4001506  0.9653333
```

Question: which is beter, GAM with smoothing spline or GAM with natural spline?

Test rmse is 51.40 for Gam model using natural spline. But Test rmse is 48.61 for Gam model using smoothing spline. so, GAM with smoothing spline is better than other.

Chapter 6- Tree based Model

- Why Tree based Model?
- Regression Trees
- Bagging
- Random Forests
- Boosting
- Research Question-

37. What is the tree size in regression tree?

38. how do we know how far back to prune the tree?

39. how many predictors are used in bagging?

40. Is test rmse in random forest lower than the result from bagging?

41. how do we decide which variables are most useful in predicting the response?

42. What is the minimum threshold value of wind speed to produce electrical power?

43. what is the effect of outdoor temperature on producing wind turbine power?

44. Does Boosting model performance improve after tuning the shrinkage parameter?

Why Tree based Model?

Tree-based methods for regression involve segmenting the predictor space into a number of simple regions. To make a prediction for an observation, we simply use the mean or mode of the training observations in the region that it belongs to. Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these approaches are known as decision tree methods.These methods are simple and useful for interpretation, but not competitive in terms of prediction accuracy when compared with the methods from the chapter on linear model selection and regularization. However, advanced methods such as bagging, random forests, and boosting can result in dramatic improvements at the expense of a loss in interpretation.

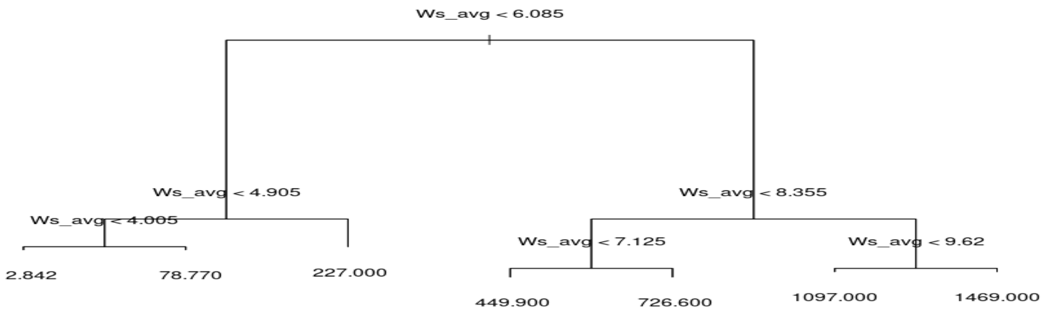
Regression Trees

```
## The tree function is used to fit a decision tree
library(tree)
tree.turbine <- tree(P_avg ~ ., data = turbine.train)
## Use the summary function to see the predictors used, # of terminal nodes, error

summary(tree.turbine)

##
## Regression tree:
## tree(formula = P_avg ~ ., data = turbine.train)
## Variables actually used in tree construction:
## [1] "Ws_avg"
## Number of terminal nodes:  7
## Residual mean deviance:  3994 = 38320000 / 9593
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -741.500   -9.687   -4.183    0.000   18.350  399.900
```

Question: What is the tree size in regression tree?

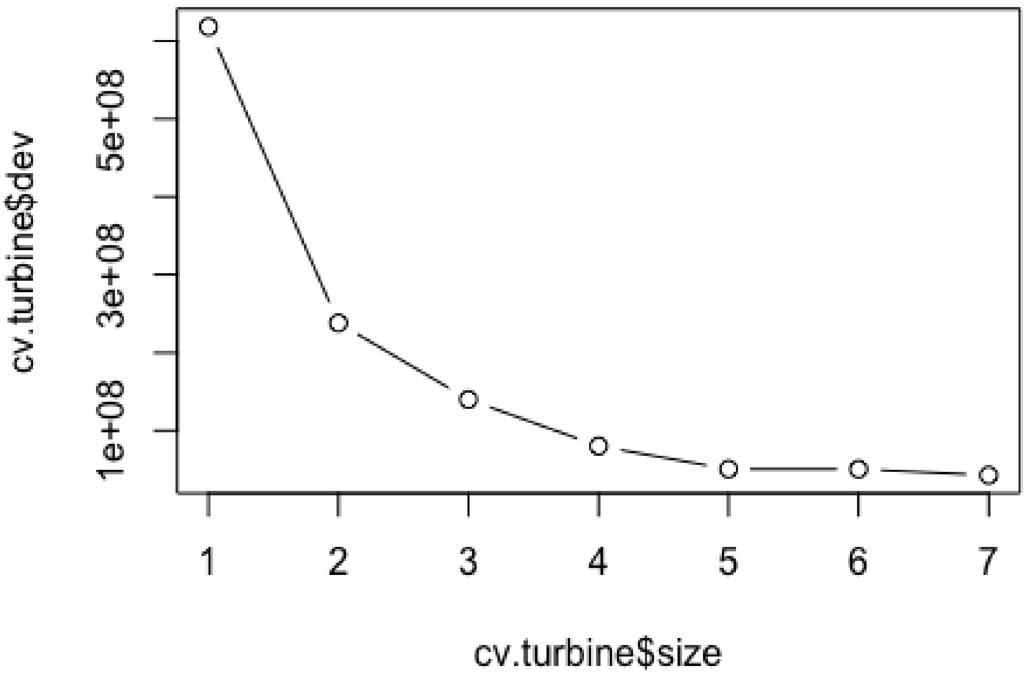
From the summary function, i see Ws_avg variables used to construct tree where terminal nodes are seven.

```
plot(tree.turbine)
text(tree.turbine, pretty = 0)
```



From above plots, the root nodes where tree is splitted at first is Ws_avg . Tree consists of six internal nodes and 7 terminal nodes.

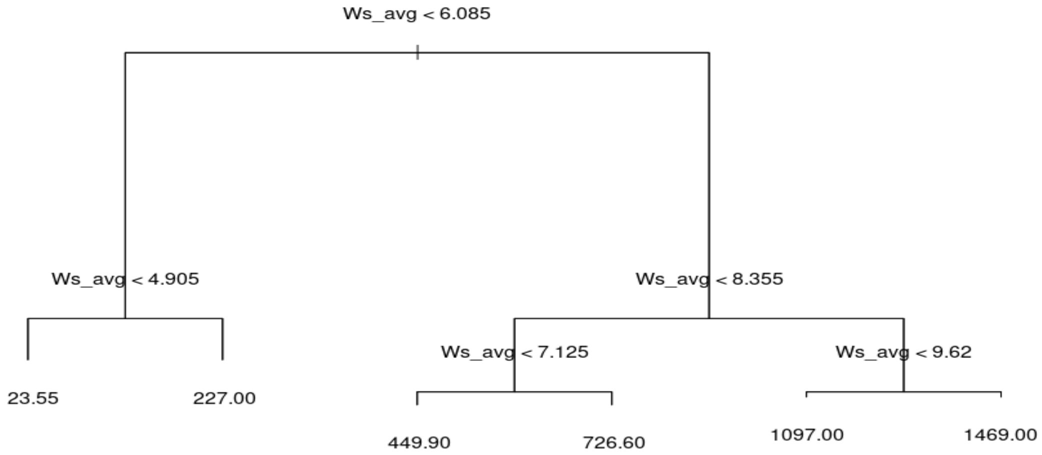
```
cv.turbine <- cv.tree(tree.turbine)
plot(cv.turbine$size, cv.turbine$dev, type = "b")
```



Question: how do we know how far back to prune the tree?

Based on cross validation results. The minimum cross validation error occurs at a tree size 6. so, A tree with 6 terminal nodes results in the lowest error.

```
#pruning the tree
prune.turbine <- prune.tree(tree.turbine, best = 6)

#plot the prune tree
plot(prune.turbine)
text(prune.turbine, pretty = 0)
```



tree pruning with 6 terminal nodes

```
#prediction
pred1 <- predict(tree.turbine, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##       RMSE         R2
## 68.5482900  0.9381233
```

Question : what is the Test rmse for regression tree?

Regression tree rmse is 68.54

Bagging

Bagging is a special case of random forest, where all predictors are used.

```
##  randomForest package is used for Bagging and Random Forests

library(randomForest)

set.seed(1)
bag.turbine <- randomForest(P_avg ~ ., data = turbine.train, mtry = 12, importance
= TRUE)
bag.turbine

##
## Call:
##  randomForest(formula = P_avg ~ ., data = turbine.train, mtry = 12,
importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 12
##
##           Mean of squared residuals: 1162.49
##                     % Var explained: 98.19
```

Question: how many predictors are used in bagging?

From the summary, mty=12 (all) number of variables are used , with a 500 tress.

```
#use the bag.turbine to make prediction on test data
pred1 <- predict(bag.turbine, newdata = turbine.test)


#determine the test rmse

rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##      RMSE        R2
## 41.102880  0.977823
```

Question: Determine the test rmse for bag.tree model?

The bagged tree prediction is 41.10 with 97% R^2 value.

Random Forest

For random forest, i simply specify a smaller number of predictors in mtry.

```
### random forest reducing number of variables(mtry=6)
set.seed(1)
rf.turbine <- randomForest(P_avg ~ ., data = turbine.train, mtry = 6, importance =
TRUE)
#prediction
pred1 <- predict(rf.turbine, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##       RMSE         R2
## 39.6709417  0.9792754
```

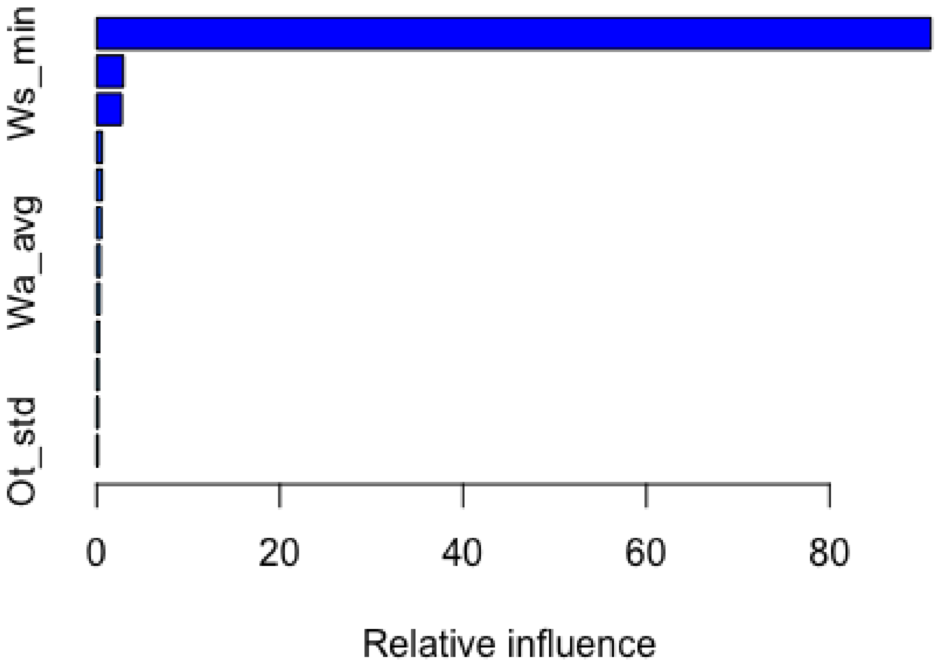
Question: Is test rmse in random forest lower than the result from bagging?

yes, random forest test rmse is 39.67 that is lower than bagging (41.10).

Boosting

The gbm function is used to perform boosting. As i am doing the regression problem,i set the distribution is gaussian and n.trees is used to specify the number of trees. interaction.depth is used to limit the depth of each tree.

```
## Boosting
library(gbm)

set.seed(1)
boost.turbine <- gbm(P_avg ~ ., data = turbine.train,distribution = "gaussian",
n.trees = 5000,interaction.depth = 4)
summary(boost.turbine)
```



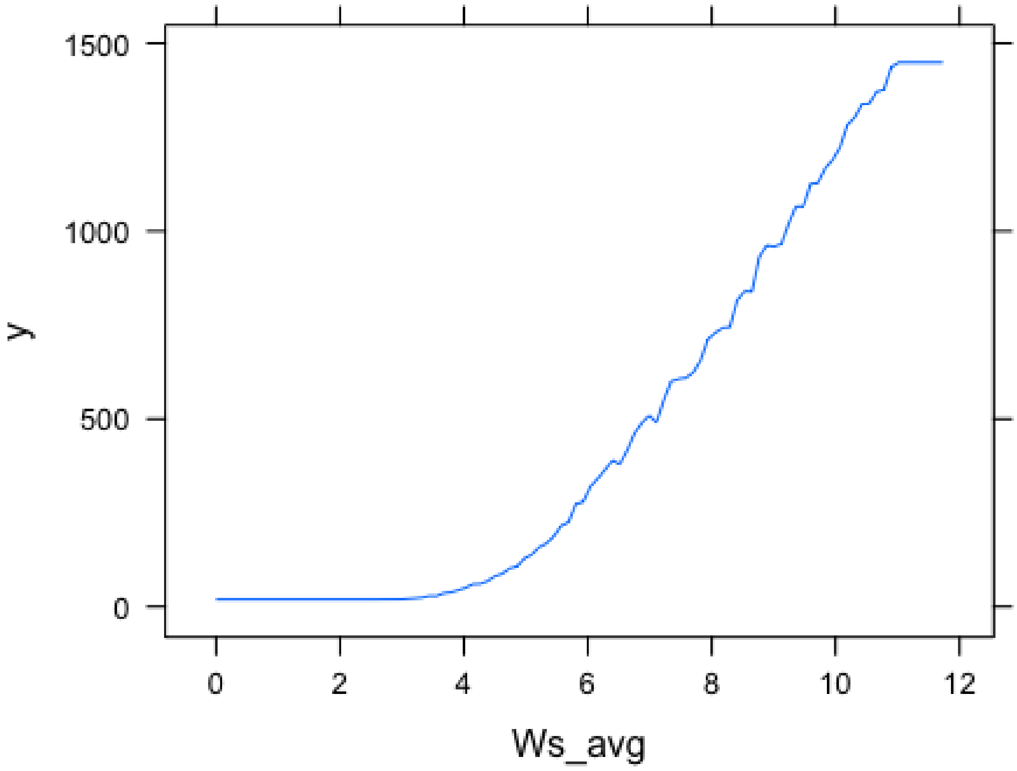
```
##            var    rel.inf
## Ws_avg Ws_avg 91.0421838
## Ws_min Ws_min  2.9113381
## Ws_max Ws_max  2.7401367
## Ot_avg Ot_avg  0.5978446
## Ws_std Ws_std  0.5946239
## Wa_std Wa_std  0.5025368
## Wa_avg Wa_avg  0.4110238
## Wa_min Wa_min  0.3723300
## Wa_max Wa_max  0.2805760
## Ot_min Ot_min  0.2218963
## Ot_max Ot_max  0.1835112
## Ot_std Ot_std  0.1419987
```

The summary functions shows the relative influence statistics for each variable.

Question: how do we decide which variables are most useful in predicting the response?

By computing relative influence plot, we can decide which variables are most useful in predicting the response. The above relative influence plot shows that average wind speed is the most important variable for predicting the response (power).

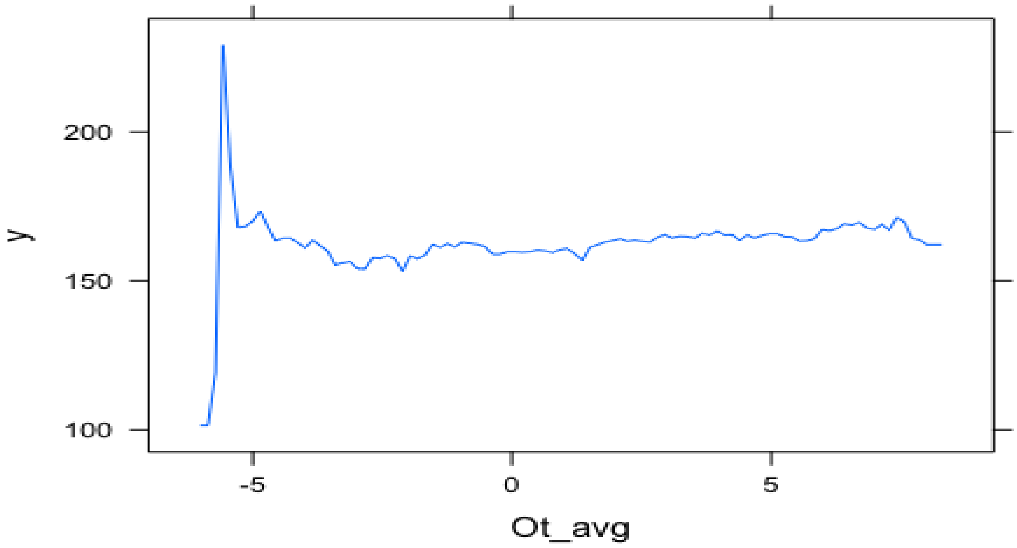
```
###plot the marginal effect of ws_avg variable on the response
plot(boost.turbine, i = "Ws_avg")
```



Question: what is value of minimum wind speed to produce the power?

Plot shows the marginal effect of Ws_avg on the response after integrating out the other variables. After 4m/s of wind speed, with increase in Ws_avg, the output response will be increased.

```
#plot the marginal effect of Ot_avg variable on the response

plot(boost.turbine, i = "Ot_avg")
```



Question: what is the effect of outdoor temperature on producing wind turbine power?

Plot shows the marginal effect of Ot_avg on the response after integrating out the other variables. From the constant flat line indicates that the effect of outdoor temperature on wind_power production is almost constant.

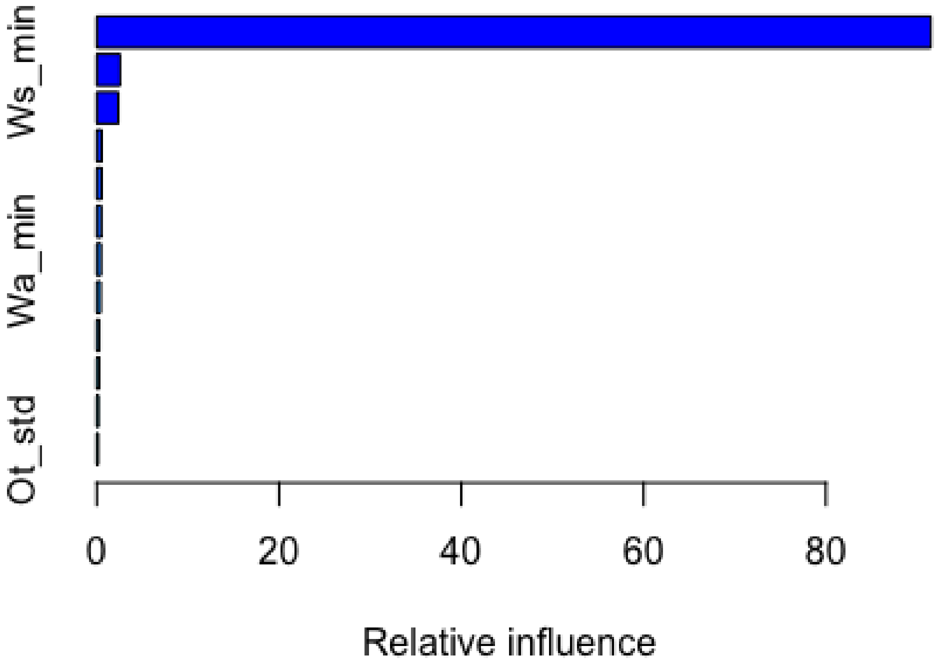
```
#prediction
pred1 <- predict(boost.turbine, newdata = turbine.test, n.trees = 5000)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)
```

```
##       RMSE         R2
## 37.0840074  0.9819546
```

Question: what is the value of test rmse for boosting model?

Boosting shows 37.084 test rmse.

```
### boosting another model
boost.turbine <- gbm(P_avg ~ ., data = turbine.train,distribution = "gaussian",
n.trees = 5000,interaction.depth = 4, shrinkage = 0.2, verbose = F)
summary(boost.turbine)
```



```
##            var    rel.inf
## Ws_avg Ws_avg 91.4865440
## Ws_min Ws_min  2.5732355
## Ws_max Ws_max  2.3713240
## Ot_avg Ot_avg  0.5617786
## Wa_std Wa_std  0.5565183
## Ws_std Ws_std  0.5127244
## Wa_min Wa_min  0.4778179
## Wa_avg Wa_avg  0.4401954
## Wa_max Wa_max  0.3201011
## Ot_min Ot_min  0.3107931
## Ot_max Ot_max  0.2205635
## Ot_std Ot_std  0.1684042
```

here will se the prediction result after tuning the shrinkage parameter lambda.

```
#prediction
pred1 <- predict(boost.turbine, newdata = turbine.test, n.trees = 5000)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##       RMSE         R2
## 36.5779571  0.9824497
```

<u>Question:</u> Does Boosting model performance improve after tuning the shrinkage parameter?

Yes, Now the test rmse is 36.57 that is lower than before.

Chapter 4- SVM and KNN

- Why SVM and KNN?
- SVM with radial kernel
- KNN
- Research Question-

45. Which kernel function i used in this turbine dataset?
46. how do i select the regularization parameter or budget controls parameter?
47. what is the value of k in knn regression model?

Why SVM and KNN

Support vector machine is a supervised machine learning algorithm. It tunes the parameter based on various kernel parameters. As target response is a regression based problem, i used support vector regression that works similar to support vector machine.In SVM, Tries to fit a hyperplane such that error is minimized and margin is maximized, with some error tolerated (depending on how much penalty you impose on errors. The SVM is a generalization of a simple classifier known as the *maximal margin classifier*. The maximal margin classifier is simple and intuitive, but cannot be applied to most datasets because it requires classes to be perfectly separable by a boundary. Another classifier known as the *support vector classifier* is an extension of the maximal margin classifier, which can be applied in a broader range of cases. The support vector machine is a further extension of the support vector classifier, which can accommodate non-linear class boundaries. Other methods KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood.

SVM with radial kernel

There are several options of kernels to use. The library we have used provides options like ‘rbf’ – radial basis function, ‘poly’ – higher degree function, ‘linear’, ‘sigmoid’, etc.

```
#support vector regressor
set.seed(1)
library(caret)
library(kernlab)

set.seed(1)
model3 <- train(P_avg ~ .,data = turbine.train,method = 'svmRadial')
model3

## Support Vector Machines with Radial Basis Function Kernel
##
## 9600 samples
##   12 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 9600, 9600, 9600, 9600, 9600, 9600, ...
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared   MAE
##   0.25  43.63790  0.9701500  22.38112
##   0.50  40.57633  0.9740706  21.19187
##   1.00  38.63702  0.9764529  20.52279
##
## Tuning parameter 'sigma' was held constant at a value of 0.08390127
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.08390127 and C = 1.
```

Question: Which kernel function i used in the turbine dataset?

I use radial basis function kernel for turbine dataset. Radial basis function can fit any data easily than other kernel functions like linear or poly with fastest learning time.

```
# model training using cross validation, 10 cross validation
ctrl <- trainControl(method = "cv",number = 10)
model4 <- train(P_avg ~ .,data = turbine.train,method = 'svmRadial', trCtrl =
ctrl)
model4

## Support Vector Machines with Radial Basis Function Kernel
##
## 9600 samples
##   12 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 9600, 9600, 9600, 9600, 9600, 9600, ...
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared   MAE
##   0.25  43.25826  0.9708694  22.35136
##   0.50  40.12926  0.9748009  21.14710
##   1.00  38.14363  0.9771856  20.53630
##
## Tuning parameter 'sigma' was held constant at a value of 0.08551708
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.08551708 and C = 1.
```

Question: how do i select the regularization parameter or budget controls parameter?

The regularization parameter C ‘s value is 1 for the lowest RMSE (38.14).

```
#prediction
pred1 <- predict(model, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##       RMSE         R2
## 42.8844652  0.9757792

plot(pred1, turbine.test$P_avg)
```

Question: what is the SVM test rmse ?

Test rmse is 42.88 for SVM with radial basis function.

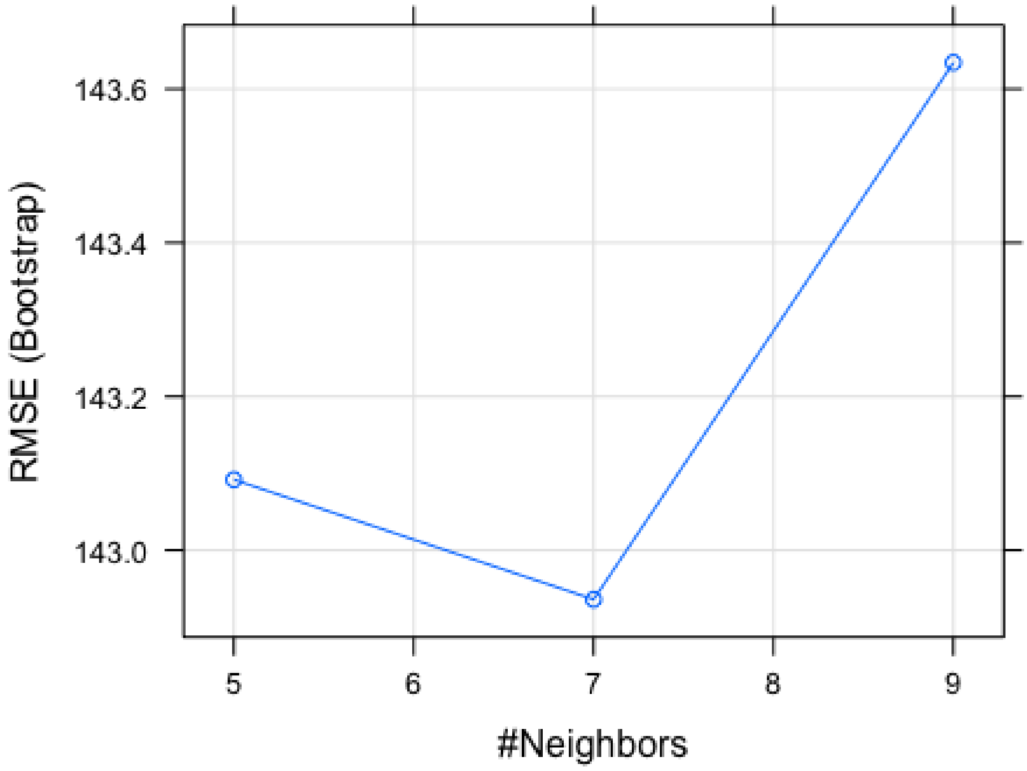
KNN regression

```
library(caret)
set.seed(1)
modelknn <- train(P_avg ~ .,data = turbine.train, method = 'knn')
modelknn

## k-Nearest Neighbors
##
## 9600 samples
##   12 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 9600, 9600, 9600, 9600, 9600, 9600, ...
## Resampling results across tuning parameters:
##
```

```
##   k  RMSE      Rsquared   MAE
##   5  143.0918  0.6788931  88.86204
##   7  142.9360  0.6789520  90.09365
##   9  143.6343  0.6760480  91.47579
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 7.
```

here rmse is selec from lower rmse, k=7

```
plot(modelknn)
```



Question: what is the value of k in knn regression model?

From above plot, the value of k is 7, because this value shows the lowes training RMSE (k=7, rmse=142)

```
#prediction
pred1 <- predict(modelknn, newdata = turbine.test)
rmse <- sqrt(sum((pred1 - turbine.test$P_avg)^2)/length(turbine.test$P_avg))
c(RMSE = rmse, R2=cor(pred1, turbine.test$P_avg) ^ 2)

##        RMSE          R2
## 133.6399967   0.7677916
```

Question: what is the value of Test rmse for KNN regression model?

Test rmse is 133.6399 with R^2 7677% for KNN regression model.

Chapter 8- Model comparison and Conclusion

- Result table
- Model comparison
- Conclusion
- References
- Research Question-

48. Which model is the best for above analysis and why?

Model comparison

Here i summarized the result of all my analysis.

Model	Test RMSE	R^2
Single variable linear model	153.31	0.69
Multivariate linear model	144.29	0.73
Linear model (validation set approach)	135.30	0.73
Linear model (LOOCV)	136.02	0.71
Linear model (k-fold, k=10)	144.27	0.73
Linear model (best subset method)	144.29	0.7278
Ridge Regression (lambda=21.37)	145	0.7270
Lasso Regression(lambda =0.460)	144.32	0.7278
Principal component Regression (M=7)	144.79	0.726
Partial Least Squares (M=7)	144.35	0.727
Polynomial regression	83.50	0.9084
Spline with Knots	51.669	0.964
Smoothing spline (lambda=0.0124)	51.44	0.9824
Local regression (span=0.2)	49.73	0.982
Generalized Additive Model- GAM with smoothing spline	48.61	0.9688
GAM with natural spline	51.40	0.9653
Decising tree	68.54	0.938
Bagging	41.10	0.9778
Random forest	39.67	0.979
Boosting	37.084	0.98
Boosting (shrinkage para. = 0.2)	**36.577**	**0.982**
SVM(radial basis function kernel)	42.88	0.975
KNN regression (k=7)	133.64	0.7677

From Above table, Tree based models perform the best than others methods. Boosting algorithms shows the lowest Test RMSE with R^2 value 98%. So, this is the best model for the turbine data set. Boosting involves building many trees, but each tree is grown sequentially. This means that each tree is grown using information from a previously grown tree. Additionally, boosting does not involve bootstrap sampling. Each tree is fit on a modified version of the original dataset

Conclusion

After applying above predictive modelling techniques and carefully observing the output, I have chosen **Boosting algorithms** to be the best model so far. As discussed earlier in decision trees, it is very much bagging method but a little bit more upgraded .Boosting involves building many trees, but each tree is grown sequentially. This means that each tree is grown using information from a previously grown tree. So Boosting gives better prediction as compared to other model. And Boosting builds tree based on bootstrap, and random forest but each tree is grown sequentially, thus decreasing the variance, so helps improving the prediction accuracy.
Moreover, an important feature of Boosting is its shrinkage parameter that slows the process down even further, allowing more and different shaped trees to attack the residuals.In boosting, unlike in bagging, the construction of each tree depends strongly on the trees already grown. Additionally, boosting could potentially overfit the data if the number of trees B is too large. Therefore, we use cross-validation to select B.Overall, Decision trees are known for showing high variance and low bias. They are often accurate but show a large degree of variability, between different data samples taken from the same data. So, In this turbine dataset, boosting is the best model to predict the output.

References

1. https://opendata-renewables.engie.com/explore/index
2. class slide
3. An introduction to Statistical Learning with Applications in R
4. https://energyeducation.ca/encyclopedia/Wind_power#:~:text=Wind%20speed%20largely%20determines%20the,electrical%20power%20from%20the%20generator
5. R language: www.r-project.org
6. Rstudio: www.rstudio.org
7. https://towardsdatascience.com/3-reasons-why-you-should-use-linear-regression-models-instead-of-neural-networks-16820319d644
8. support vector regression: https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0#:~:text=Support%20Vector%20Regression%20is%20a,the%20maximum%20number%20of%20points.
9. knn regression: https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html