Atmospheric Ozone Concentration and Meteorology in LA Basin, 1976 - A Regression Study

Arkajyoti Bhattacharjee (201277)
Vishweshwar Tyagi (191173)
Saurab Jain (170642)
Apoorva Singh (17816140)

Indian Institute of Technology, Kanpur

Contents

Acknowledgement

We would like to express our gratitude to *Dr. Sharmishtha Mitra* for giving us the opportunity to do this project. It has been a great learning experience and has also provided us with a hands-on practical insight of the theoretical knowledge gathered during the course MTH416A: Regression Analysis. It has also urged us to explore new concepts and apply them in our project.

We would like to thank our friends who have helped in this project. Finally, we would like to thank IIT Kanpur to make all of this possible in these unprecedented times.

Introduction

Although it represents only a tiny fraction of the atmosphere, ozone is crucial for life on Earth. With a weakening of this shield, we would be more susceptible to skin cancer, cataracts and impaired immune systems. Again. closer to Earth in the troposphere (the atmospheric layer from the surface up to about 10 km), ozone is a harmful pollutant that causes damage to lung tissue and plants.

In this project, we aim to understand the relationship between **Ozone concentration** and meteorological variables like **temperature**, **pressure**, **humidity**, etc. and develop **parametric** and **non-parametric** models to be able to **predict** ozone concentration based on given values of the meteorological variables.

We have fitted various regression models while detecting and taking remedial measures for the problems of **multi-collinearity**, **heteroscedasticity** and **auto-correlation**. After that, we compared the predictive power of the models developed in the process by compairing the Root Mean Square Error(**RMSE**) of the model.

The entire project is available in the Github link : *https://github.com/ArkaB-DS/Modelling-linear-relationship-between-Ozone-Concentration-and-Meteorology-LA-Basin-1976*

About the Data

Data Description

We will make use of the **Ozone in Los Angeles Basin in 1976** dataset for this project. It is a historical time-series data. It has **330** observations and **10** variables.

The variables associated with this dataset are as follows -

O3: Ozone conc., ppm, at Sandbug AFB.
vh: a numeric vector
wind: wind speed
humidity: a numeric vector
temp: temperature
ibh: inversion base height
dpg: Daggett pressure gradient
ibt: a numeric vector
vis: visibility
doy: day of the year

Here, **O3** is the response variable and the remaining are potential regressors.

Source :

Breiman, L. and J. H. Friedman (1985). Estimating optimal transformations for multiple regression and correlation. Journal of the American Statistical Association 80, 580-598.

Link to the Data File :

https://github.com/ArkaB-DS/Modelling-linear-relationship-between-Ozone-Concentration-and-Meteorology-LA-Basin-1976/blob/main/Ozone2.csv

Parametric Setup : Model Assumptions

We usually use parametric models for the ease of interpretability of the model and its parameters. It is useful when the goal is inference.

For a preliminary analysis, we fit a multiple linear regression model to the data, with **O3** as the response and all other variables as regressors.

The model is given by :

$$O_3 = \beta_0 + \beta_1 vh + \beta_2 humidity + \beta_3 wind + \beta_4 temp + \beta_5 dpg + \beta_6 ibt + \beta_7 ibh + \beta_8 doy + \beta_9 vis + \epsilon$$

We assume a Gauss-Markov model i.e. we make the following assumptions:

1. $E(\epsilon) = 0$
2. $var(\epsilon) = \sigma^2 I$ i.e.
    2.1. $var(\epsilon_i) = \sigma^2 \; \forall \; i$
    2.2. $cov(\epsilon_i, \epsilon_j) = 0 \; \forall \; i \neq j$

In addition, for testing purposes, we assume

3. $\epsilon \sim N(0, \sigma^2 I)$

Preliminary Analysis - Data Structure, Summary and Exploratory Analysis

We first load the data-set in **R**

```
install.packages("faraway")
data(ozone,package="faraway")
```

We look into the first 6 rows of the dataset to get an idea what values each variable is taking.

```
head(ozone)
```

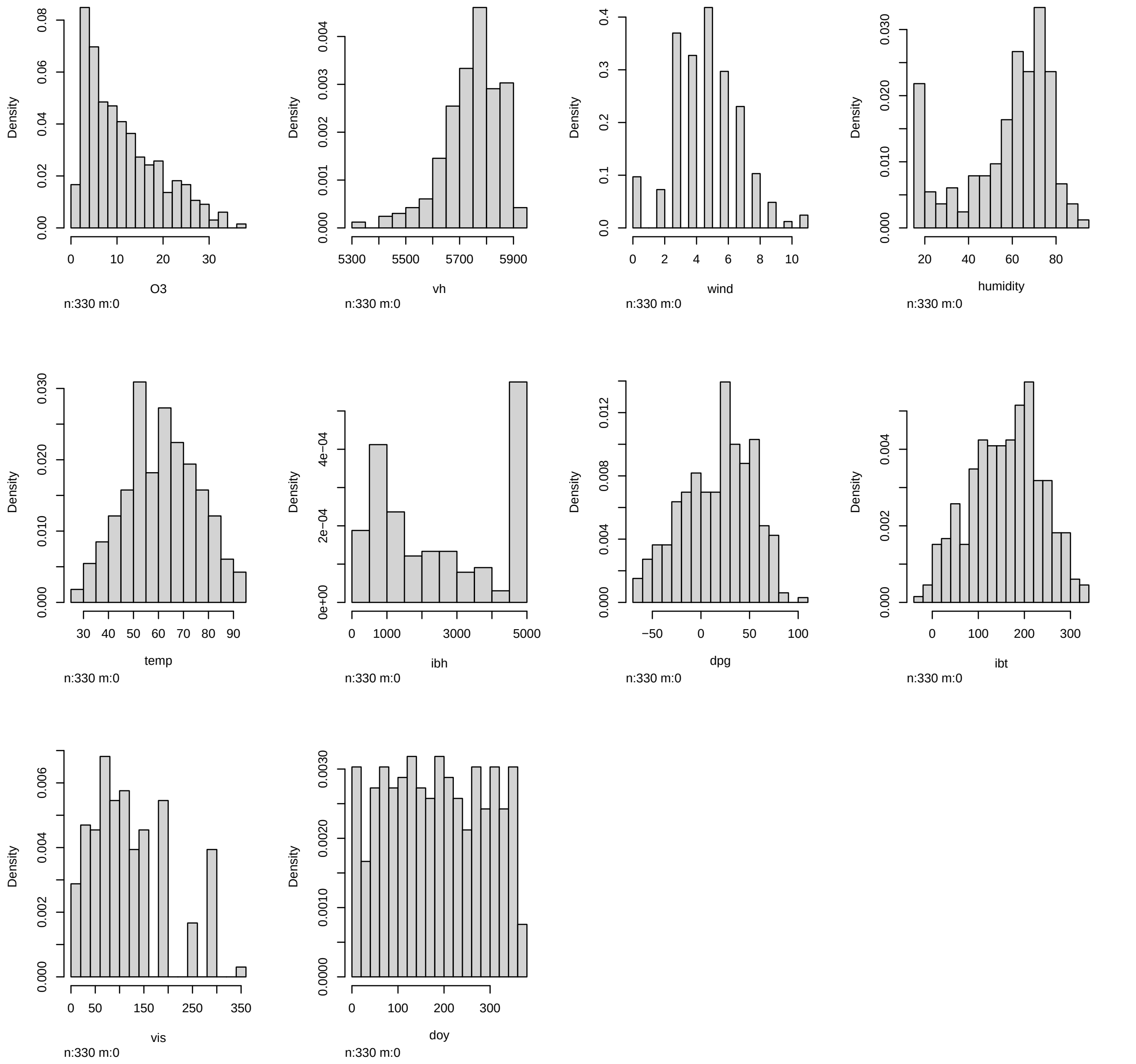
```
##   O3   vh wind humidity temp  ibh dpg ibt vis doy
## 1  3 5710    4       28   40 2693 -25  87 250  33
## 2  5 5700    3       37   45  590 -24 128 100  34
## 3  5 5760    3       51   54 1450  25 139  60  35
## 4  6 5720    4       69   35 1568  15 121  60  36
## 5  4 5790    6       19   45 2631 -33 123 100  37
## 6  4 5790    3       25   55  554 -28 182 250  38
```

We take the **doy** variable and compute it modulo 365 and then add 1 to it to make it in the range 1-365. We then look into the **structure** of the data and compute basic **summary statistics** of the data. We plot the **histograms** of the variables as well.

```
ozone<-data.frame(ozone[,-10],"doy"=ozone[,10]%%365+1)
str(ozone)
summary(ozone)
library(Hmisc)
par(mfrow=c(3,3))
hist.data.frame(ozone,freq=FALSE)
```

```
## 'data.frame':    330 obs. of  10 variables:
##  $ O3      : num  6 5 3 4 7 5 5 4 3 2 ...
##  $ vh      : num  5680 5780 5810 5760 5680 5750 5790 5770 5750 5720 ...
##  $ wind    : num  0 4 3 0 0 0 5 3 0 0 ...
##  $ humidity: num  52 19 19 32 58 26 19 19 19 19 ...
##  $ temp    : num  50 48 51 62 40 44 49 53 53 53 ...
##  $ ibh     : num  1154 2933 3064 826 5000 ...
##  $ dpg     : num  -22 -40 -33 -16 2 -52 -48 -37 -26 -31 ...
##  $ ibt     : num  164 155 171 182 61 201 126 131 106 108 ...
##  $ vis     : num  60 300 200 300 50 40 70 150 150 70 ...
##  $ doy     : num  1 2 3 4 5 6 7 8 9 10 ...
```

```
##        O3              vh            wind           humidity
##  Min.   : 1.00   Min.   :5320   Min.   : 0.000   Min.   :19.00
##  1st Qu.: 5.00   1st Qu.:5690   1st Qu.: 3.000   1st Qu.:47.00
##  Median :10.00   Median :5760   Median : 5.000   Median :64.00
##  Mean   :11.78   Mean   :5750   Mean   : 4.848   Mean   :58.13  
##  3rd Qu.:17.00   3rd Qu.:5830   3rd Qu.: 6.000   3rd Qu.:73.00  
##  Max.   :38.00   Max.   :5950   Max.   :11.000   Max.   :93.00  
##       temp            ibh              dpg              ibt       
##  Min.   :25.00   Min.   : 111.0   Min.   :-69.00   Min.   :-25.0  
##  1st Qu.:51.00   1st Qu.: 877.5   1st Qu.: -9.00   1st Qu.:107.0  
##  Median :62.00   Median :2112.5   Median : 24.00   Median :167.5  
##  Mean   :61.75   Mean   :2572.9   Mean   : 17.37   Mean   :161.2  
##  3rd Qu.:72.00   3rd Qu.:5000.0   3rd Qu.: 44.75   3rd Qu.:214.0  
##  Max.   :93.00   Max.   :5000.0   Max.   :107.00   Max.   :332.0  
##       vis             doy        
##  Min.   :  0.0   Min.   :  1.00  
##  1st Qu.: 70.0   1st Qu.: 96.25  
##  Median :120.0   Median :182.50  
##  Mean   :124.5   Mean   :183.88  
##  3rd Qu.:150.0   3rd Qu.:273.75  
##  Max.   :350.0   Max.   :365.00
```

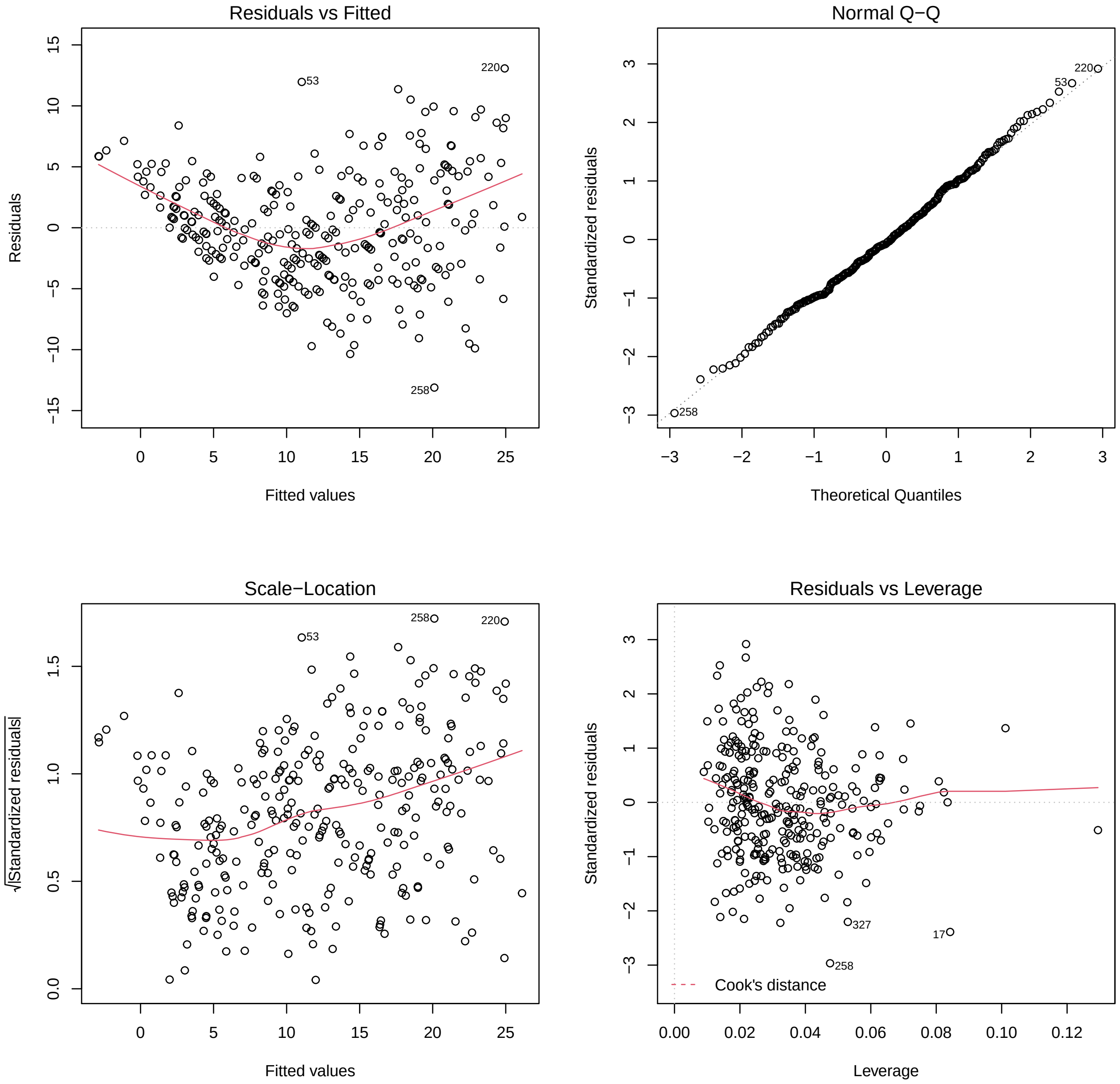


As evident above, the data contains no NA values. All the variables take numeric values.

We divide the data into 80% for training and 20% for validation.

Now, we fit a **multiple linear regression** model with **O3** as the response and all other variables as regressors. We plot the basic summary plots based on the fitted model, **lmod0**, say, to get more idea about the data.

```
lmod0<-lm(O3~.,data=ozone[1:300,])
par(mfrow=c(2,2))
plot(lmod0)
summary(lmod0)
```



```
##
## Call:
## lm(formula = O3 ~ ., data = ozone[1:300, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1115  -2.9906  -0.2988   2.9341  13.0716
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.5006544 32.4565235   0.755 0.450936
## vh          -0.0062400  0.0059171  -1.055 0.292495
## wind         0.0328400  0.1491718   0.220 0.825910
## humidity     0.0771142  0.0213435   3.613 0.000357 ***
## temp         0.2647941  0.0520989   5.083 6.69e-07 ***
## ibh         -0.0004993  0.0003108  -1.607 0.109232
## dpg          0.0009924  0.0119021   0.083 0.933604
## ibt          0.0294090  0.0144697   2.032 0.043018 *
## vis         -0.0060750  0.0039846  -1.525 0.128450
## doy         -0.0023407  0.0041495  -0.564 0.573123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.53 on 290 degrees of freedom
## Multiple R-squared:  0.6986, Adjusted R-squared:  0.6892
## F-statistic: 74.68 on 9 and 290 DF,  p-value: < 2.2e-16
```

Based on the above graphs, we observe the following -

- There is curvature in the **residual vs fitted plot** indicating a **non-linear** relationship in the data-set.
- There is **heteroscedasticity** in the data as the residuals do not form a constant band.
- The **normal Q-Q** plot shows a fairly straight line, indicating the errors are more-or-less **normally distributed**.
- 17, 53, 258 and 220^{th} observations may need special attention.

Based on the summary of the fitted model, we make the following observations -

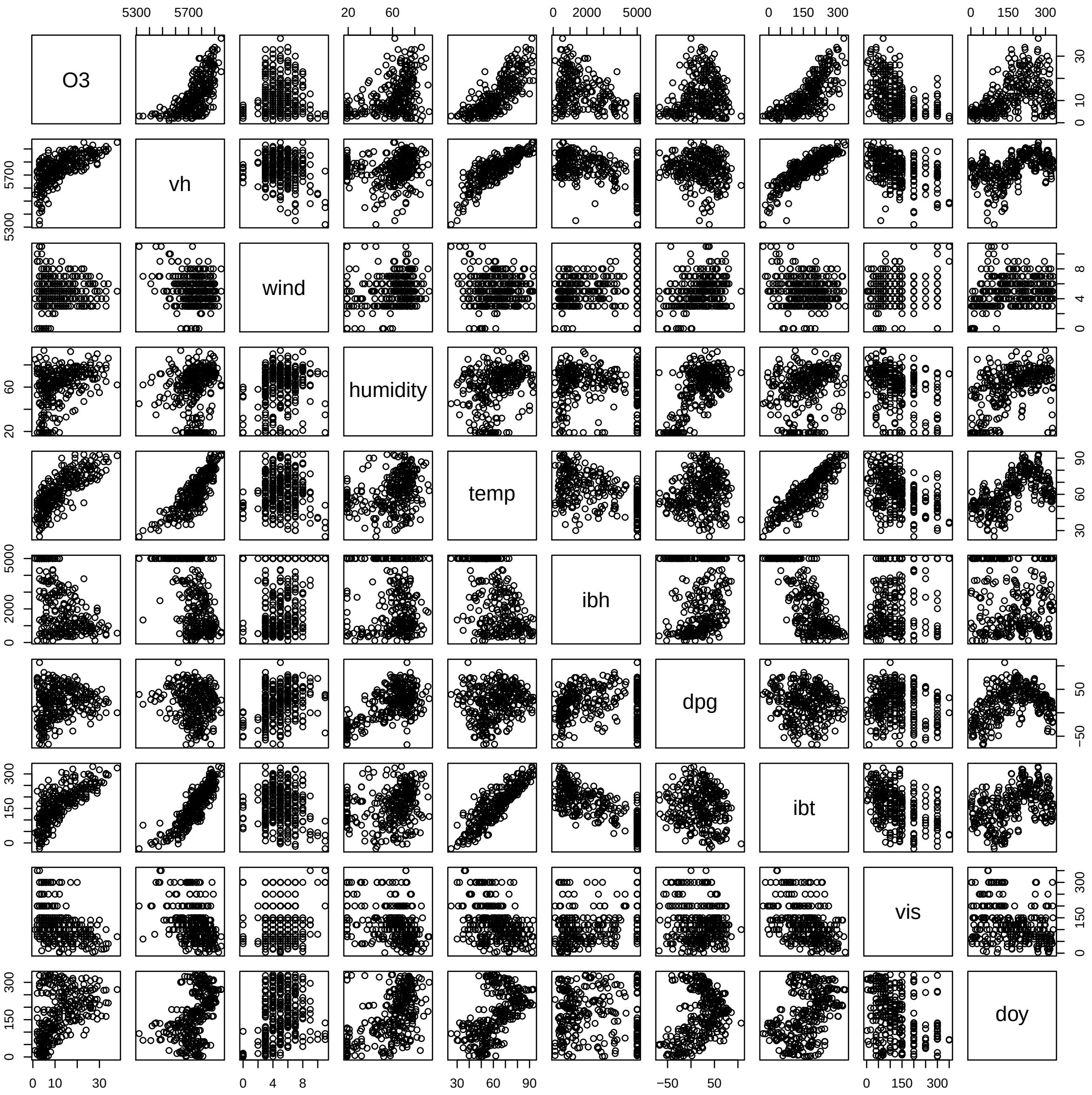
- The **Multiple R-squared** of the model is: **0.6986** and the **Adjusted R-squared** is: **0.6892**.
- The absolute value of the estimate of the regression coefficient of **wind**, **dpg** and **doy** is less than its standard error; it implies that we can drop those variables.
- Since the errors seem to follow normal distribution based on **Q-Q** plot, so taking level of significance to be 0.01, only **humidity** and **temperature** seem to be *statistically significant* based on their p-values.

We now get into a deeper analysis of the data.

Multicollinearity

We first look at the **scatterplot matrix**, a grid (or matrix) of scatter plots used to visualize bivariate relationships between combinations of variables, to get a better idea as to how the variables are related to each other.

```
pairs(O3~.,data=ozone[1:300,])
```



Based on the above **scatterplot matrix**, we make the following observations -

- **vh** and **temp** seem to be almost perfectly **positively correlated**
- **temp** and **ibt** seem to be almost perfectly **positively correlated**
- As expected from the above two points, **vh** and **temp** seem to be almost perfectly **positively**

correlated
- **dpg** and **doy** have a somewhat quadratic relationship
- **temp** and **doy** have a somewhat quadratic relationship

Next, we use the **eigen-decompostion proportion**, to find out which regressors are responsible for multicollinearity. The proportions are given by -

$$\pi_{kj} = \frac{v_{kj}^2/l_k}{\sum_{k=1} v_{kj}^2/l_k}$$

, where $\sum_k \pi_{kj} = 1 \forall j$. Here, $l_1, ...l_p$ are the eigenvalues of $X'X$ with corresponding eigenvectors $v_1, ..v_p$. High values of π_{kj} within the corresponding row indicates that the regressors are involved in multicollinearity.

```r
install.packages("mctest")
library(mctest)
eigprop(lmod0)
```

```
##
## Call:
## eigprop(mod = lmod0)
##
##    Eigenvalues       CI (Intercept)     vh   wind humidity   temp    ibh    dpg
## 1       8.1452   1.0000      0.0000 0.0000 0.0014   0.0005 0.0001 0.0007 0.0016
## 2       0.7577   3.2787      0.0000 0.0000 0.0000   0.0011 0.0000 0.0054 0.2883
## 3       0.6121   3.6478      0.0000 0.0000 0.0017   0.0005 0.0005 0.0413 0.0577
## 4       0.1975   6.4221      0.0000 0.0000 0.0012   0.0003 0.0000 0.1150 0.0798
## 5       0.1202   8.2310      0.0000 0.0000 0.0000   0.0164 0.0007 0.0062 0.0063
## 6       0.1004   9.0067      0.0000 0.0000 0.8948   0.0002 0.0014 0.0435 0.0358
## 7       0.0477  13.0701      0.0000 0.0000 0.0362   0.5155 0.0117 0.0693 0.2288
## 8       0.0147  23.5631      0.0013 0.0010 0.0200   0.4438 0.0112 0.3488 0.0452
## 9       0.0044  42.9618      0.0001 0.0001 0.0001   0.0013 0.9418 0.2976 0.2544
## 10      0.0000 512.9518      0.9985 0.9988 0.0445   0.0203 0.0326 0.0721 0.0021
##       ibt    vis    doy
## 1  0.0001 0.0021 0.0014
## 2  0.0000 0.0352 0.0038
## 3  0.0047 0.0407 0.0092
## 4  0.0010 0.4624 0.0815
## 5  0.0005 0.2653 0.5417
## 6  0.0015 0.0250 0.0074
## 7  0.0323 0.0229 0.0276
## 8  0.0947 0.1343 0.3030
## 9  0.6620 0.0012 0.0022
## 10 0.2031 0.0109 0.0220
##
## ===============================
## Row 10==> vh, proportion 0.998842 >= 0.50
## Row 6==> wind, proportion 0.894776 >= 0.50
## Row 7==> humidity, proportion 0.515506 >= 0.50
## Row 9==> temp, proportion 0.941828 >= 0.50
## Row 9==> ibt, proportion 0.662017 >= 0.50
## Row 5==> doy, proportion 0.541742 >= 0.50
```

Clearly, **vh**, **wind**, **temp**, **humidity**, **ibt** and **doy** have variance decompositon proportion greater than 0.50. We, further, look into the **variance inflation factors(VIFs)** of the model for the same purpose.

Note that $VIF = \frac{1}{1-R_j^2}$, where R^2 is the **multiple** R^2 for the regression of X_j on the other covariates (a regression that does not involve the response variable Y).

```
install.packages("car")
library(car)
vif(lmod0)
```

```
##        vh      wind  humidity      temp       ibh       dpg       ibt       vis
##  5.884904  1.282581  2.445097  8.624229  4.492747  2.465877 18.457599  1.426169
##       doy
##  2.266763
```

Clearly, **vh**, **temp** and **ibt** have **VIFs$>$5.**

So, we have the problem of multicollinearity and we use three methods as a remedial measure -
1. **Dropping Variables**(***Model A***)
2. **Ridge Regression**(***Model B***)
3. **Principal Components Regression**(***Model C***)

Dropping of Variables(Model A)

Now, based on the **scatterplot matrix**, we drop the variables **vh** and **ibt** from the model and again fit the data into a new model, say **lmodA**.
We compute the **VIF**s of **lmodA** and compute the R^2 value the new model to see if there is any significant drop due to variables dropped.

```
vif(lm(O3~.-vh-ibt,data=ozone[1:300,])) #all <5
cat("The R^2 value of lmodA is : ",summary(lm(O3~.-vh-ibt,data=ozone[1:300,]))$r.squared)
```

```
##     wind humidity     temp      ibh      dpg      vis      doy
## 1.227943 2.402486 2.367630 1.730002 1.867278 1.392424 2.143054
```

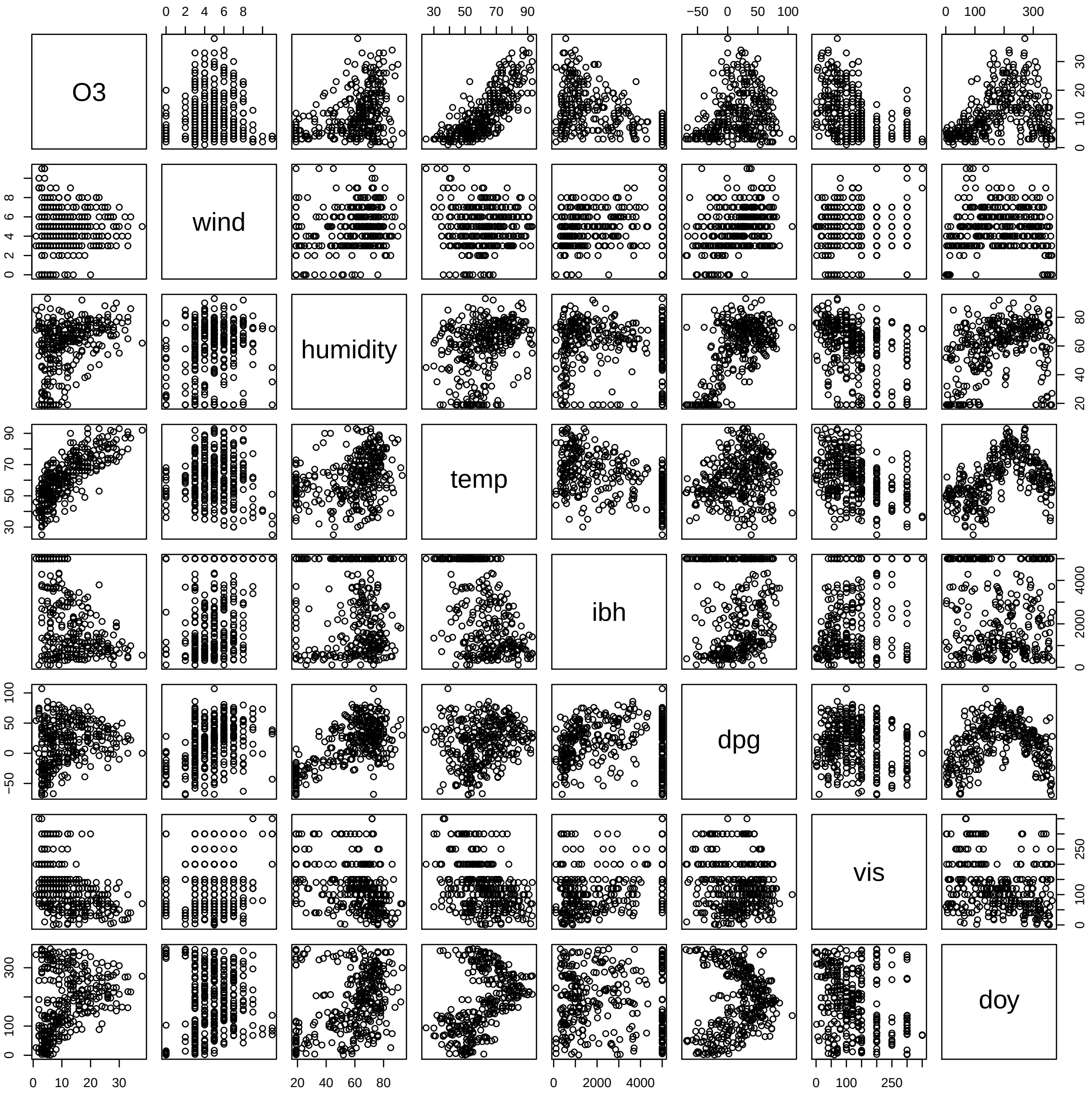
```
## The R^2 value of lmodA is :  0.6942595
```

Recall that the R^2 value of **lmod0** is 0.6986 and that of **lmodA** is 0.6942595 - not significantly lower from the former. Also, the VIFs are all less than 5 and apparently,the multicollinearity problem is solved. Hence, our new model is **lmodA**.

```
lmodA<-lm(O3~.-ibt-vh,data=ozone[1:300,])
```

We, again, look into the new scatterplot matrix, corresponding to **lmodA** to see how the remaining variables are inter-connected.

```r
pairs(ozone[,c(1,3,4,5,6,7,9,10)])
```



We make the following observations based on the above scatterplot matrix -

- There is a quadratic relationship between **temp** and **doy**. This is expected as temperature increases in the middle of the year and is lower elsewhere.
- A similar relationship seems to exist between **dpg** and **doy**

Ridge Regression(Model B)

We employ ridge regression to solve the problem of multicollinearity.

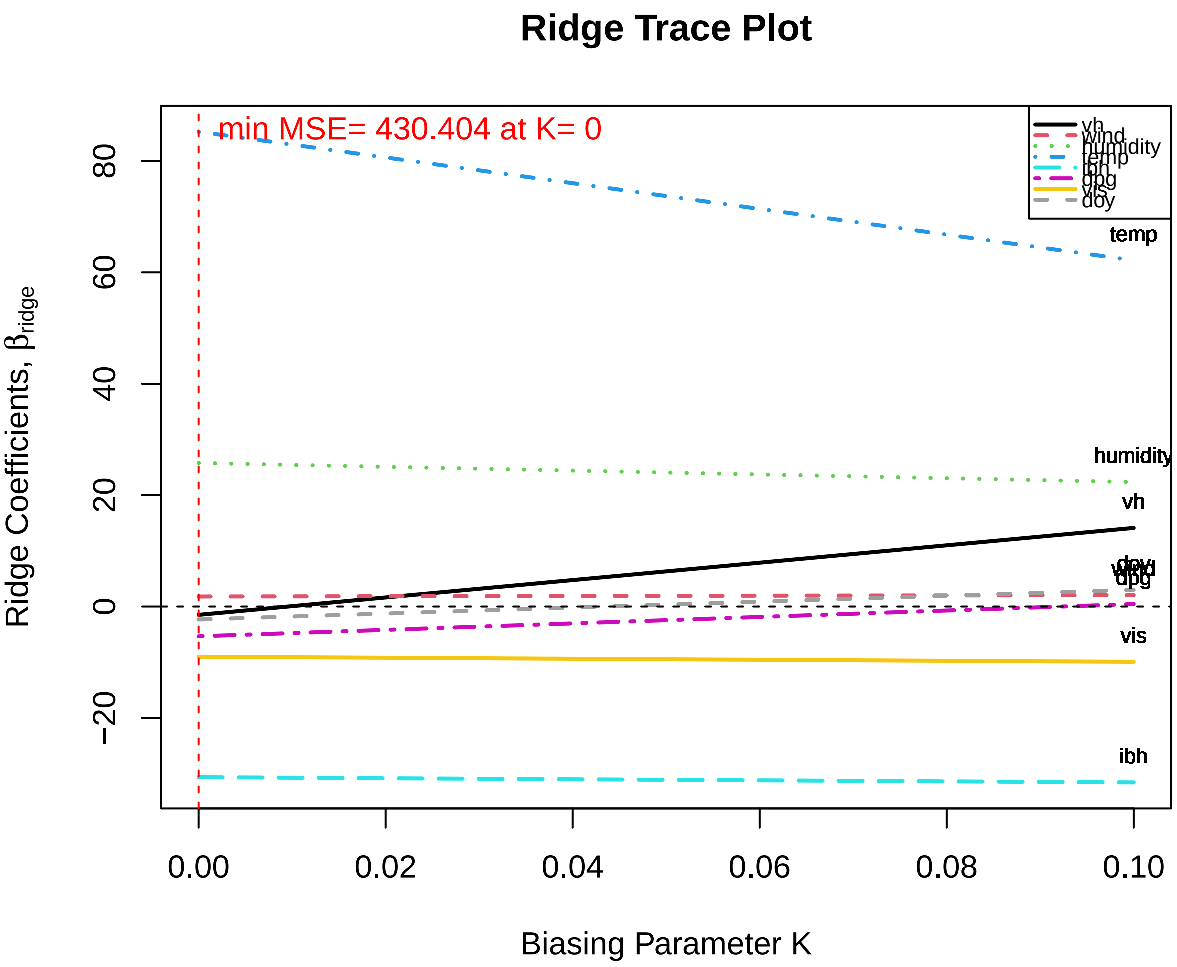
The ridge regression estimator is -

$$\hat{\beta}_{ridge} = (X'X + KI_p)^{-1}X'y$$

, where $K(>0)$ is the **ridge complexity parameter**.

The Ridge complexity parameter is selected based on the iterative method suggested by **Hoerl et al.(1975)**.

```
install.packages("lmridge")
library(lmridge)
lmodB<-lmridge(O3~vh+wind+humidity+temp+ibh+ibt+dpg+vis+doy,
data=ozone[1:300,],K=seq(0,0.1,1e-1))
plot(lmodB)
```



The Ridge complexity parameter turns out to be $K = 0.008$.

So, we define our **Ridge regression model** as **lmodB** and compute the summary of the model. We check out its VIFs as well.

```
lmodB<-lmridge(O3~vh+ibt+wind+humidity+temp+ibh+dpg+vis+doy,
data=ozone[1:300,],K=0.008)
```

```
summary(lmodB)
vif(lmodB)
```

```
##
## Call:
## lmridge.default(formula = O3 ~ vh + wind + humidity + temp +
##     ibh + dpg + vis + doy, data = ozone[1:300, ], K = 0.008)
##
##
## Coefficients: for Ridge parameter K= 0.008
##             Estimate Estimate (Sc) StdErr (Sc) t-value (Sc) Pr(>|t|)
## Intercept   -11.4102    69083.1778  55405.4290       1.2469   0.2134
## vh            0.0004        0.8059      9.2727       0.0869   0.9308
## wind          0.0550        1.8924      5.0719       0.3731   0.7093
## humidity      0.0766       25.4017      6.9113       3.6754   0.0003 ***
## temp          0.3218       82.1650      9.5275       8.6239   <2e-16 ***
## ibh          -0.0010      -30.8637      5.9070      -5.2249   <2e-16 ***
## dpg          -0.0076       -4.5306      6.3312      -0.7156   0.4748
## vis          -0.0067       -9.0767      5.3426      -1.6989   0.0904 .
## doy          -0.0011       -1.7343      6.6325      -0.2615   0.7939
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Ridge Summary
##         R2     adj-R2   DF ridge          F        AIC        BIC
##    0.68730    0.67980    7.83996    82.86510  916.23920 2656.41145
## Ridge minimum MSE= 413.7569 at K= 0.008
## P-value for F-test ( 7.83996 , 292.0061 ) = 7.795083e-70
## -------------------------------------------------------------------
```

```
##            vh    wind humidity    temp     ibh     dpg     vis     doy
## k=0.008 4.159 1.24427  2.31045 4.39076 1.68776 1.93889 1.38067 2.12778
```

Recall that the R^2 value of **lmod0** is 0.6986 and that of **lmodB** is 0.68730 - not significantly lower from the former. Also, all variables have **VIF<5**. So, our multicollinearity probelm is apparently solved, although **temp** and **vh** seem to have a higher VIFs than others.

Principal Components Regression(Model C)

Here, we use PCR to solve the problem of multi-collinearity.

The PCR method may be broadly divided into three major steps:

1. Perform **PCA** on the data matrix for the explanatory variables to obtain the **principal components**, and then select a subset, based on some appropriate criteria, of the principal components so obtained for further use.

2. Regress the observed vector of outcomes on the selected principal components as covariates, using **OLS** regression to get a vector of estimated regression coefficients.
3. Transform this vector back to the scale of the actual covariates, using the selected PCA loadings (the eigenvectors corresponding to the selected principal components) to get the final **PCR estimator** for estimating the regression coefficients characterizing the original model.

```
pcr<-prcomp(ozone[c(1:300),-1],center=TRUE,scale=TRUE)
summary(pcr)
Data<-data.frame("O3"=ozone[1:300,1],pcr$x)
lmodC<-lm(O3~.,data=Data)
beta<-pcr$rotation%*%coef(lmodC)[-1]
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.9906 1.4324 0.9824 0.80988 0.78021 0.60941 0.47795
## Proportion of Variance 0.4403 0.2280 0.1072 0.07288 0.06764 0.04126 0.02538
## Cumulative Proportion  0.4403 0.6683 0.7755 0.84840 0.91604 0.95730 0.98268
##                            PC8     PC9
## Standard deviation     0.34451 0.19278
## Proportion of Variance 0.01319 0.00413
## Cumulative Proportion  0.99587 1.00000
```

The above table gives the standard deviation, proportion of variance and cumulative proportion of variance. We use all the PCs to fit a model. We will use variable selection later to select a smaller number of PCs. The R^2 value of the fitted model, **lmodC**,say, is below along with the **VIF**s

```
lmodC<-lm(ozone[1:300,1]~.,data=data.frame(pcr$x))
cat("The R^2 value of lmodC is: ",summary(lmodC)$r.squared)
vif(lmodC)
```

```
## The R^2 value of lmodC is:  0.6985772
```

```
## PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9
##   1   1   1   1   1   1   1   1   1
```

Recall that the R^2 value of **lmod0** is 0.6986 and that of **lmodC** is 0.6985772- almost equal to the former,which is expected. Also, we see that all PCs have **VIF<5**.

Variable Selection

Now, because of **Occam Razor**'s principle or the **law of parsimony**, we need to do variable selection.

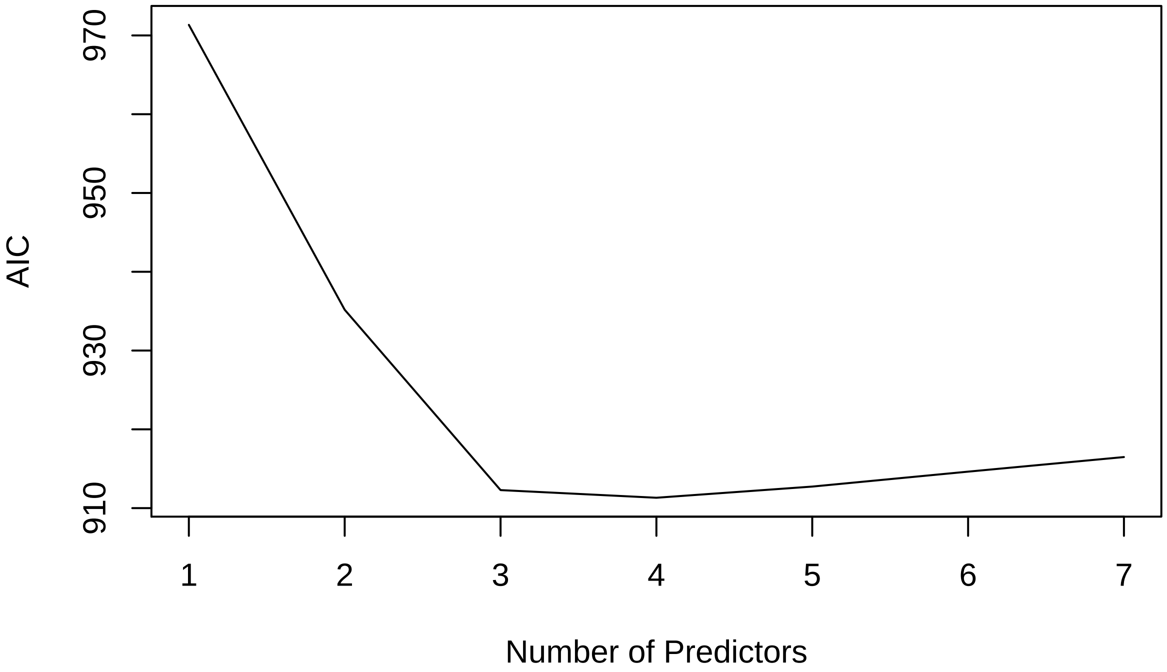
For this, we first plot the **Akaike Information Criterion(AIC)** against the **number of regressors(p)** and see for what **p** is the **AIC** minimum. We then use **stepwise** or **exhaustive** method to find the best subset of regressors which has the **minimum AIC**. We do this for each of the three models **- A**, **B** and **C** respectively.

Note that $AIC = -2log(\hat{L}) + 2p$, where $\hat{L}$ is the **maximum value** of the **likelihood function** L.

Model A

```
install.packages("leaps")
library(leaps)
b <- regsubsets(x=model.matrix(lmodA)[,-1],y=ozone[1:300,1])
rs <- summary(b)
rs$which
AIC <- 300*log(rs$rss/300) + (2:8)*2
plot(AIC ~ I(1:7), ylab="AIC", xlab="Number of Predictors",
type="l",col="red",lwd=2)
```

```
##   (Intercept)  wind humidity temp   ibh   dpg   vis   doy
## 1        TRUE FALSE    FALSE TRUE FALSE FALSE FALSE FALSE
## 2        TRUE FALSE    FALSE TRUE  TRUE FALSE FALSE FALSE
## 3        TRUE FALSE     TRUE TRUE  TRUE FALSE FALSE FALSE
## 4        TRUE FALSE     TRUE TRUE  TRUE FALSE  TRUE FALSE
## 5        TRUE FALSE     TRUE TRUE  TRUE  TRUE  TRUE FALSE
## 6        TRUE  TRUE     TRUE TRUE  TRUE  TRUE  TRUE FALSE
## 7        TRUE  TRUE     TRUE TRUE  TRUE  TRUE  TRUE  TRUE
```



Based on the above plot, we see that for **4** regressors, the **AIC** is minimum. Also, corresponding to **4**, we have **humidity**, **ibh**, **temp** and **vis** as regressors.

We also use stepwise regression to confirm this -

```
step(lmodA)
```

```
## Start:  AIC=916.48
## O3 ~ (vh + wind + humidity + temp + ibh + dpg + ibt + vis + doy) -
##     ibt - vh
##
##            Df Sum of Sq    RSS     AIC
## - doy       1      3.01 6038.3  914.63
## - wind      1      3.10 6038.4  914.63
## - dpg       1     13.68 6049.0  915.16
## <none>                  6035.3  916.48
## - vis       1     56.90 6092.2  917.30
## - humidity  1    279.49 6314.8  928.06
## - ibh       1    538.78 6574.1  940.13
## - temp      1   2989.86 9025.2 1035.20
##
## Step:  AIC=914.63
## O3 ~ wind + humidity + temp + ibh + dpg + vis
```

```
##
##            Df Sum of Sq     RSS     AIC
## - wind      1       2.1  6040.5  912.74
## - dpg       1      13.0  6051.4  913.28
## <none>                   6038.3  914.63
## - vis       1      56.5  6094.8  915.42
## - humidity  1     294.1  6332.4  926.89
## - ibh       1     608.1  6646.5  941.42
## - temp      1    4326.9 10365.3 1074.73
##
## Step:  AIC=912.74
## O3 ~ humidity + temp + ibh + dpg + vis
##
##            Df Sum of Sq     RSS     AIC
## - dpg       1      11.7  6052.2  911.32
## <none>                   6040.5  912.74
## - vis       1      54.5  6095.0  913.43
## - humidity  1     305.7  6346.1  925.54
## - ibh       1     614.5  6655.0  939.80
## - temp      1    4349.3 10389.7 1073.44
##
## Step:  AIC=911.32
## O3 ~ humidity + temp + ibh + vis
##
##            Df Sum of Sq     RSS     AIC
## <none>                   6052.2  911.32
## - vis       1      60.2  6112.4  912.29
## - humidity  1     395.2  6447.3  928.29
## - ibh       1     700.9  6753.0  942.19
## - temp      1    4348.1 10400.3 1071.74
```

```
##
## Call:
## lm(formula = O3 ~ humidity + temp + ibh + vis, data = ozone[1:300,
##     ])
##
## Coefficients:
## (Intercept)     humidity         temp          ibh          vis
##   -8.326321     0.066638     0.322167    -0.001032    -0.006644
```

Hence, the final fitted model is again named **lmodA** and its R^2 value is printed.

```r
lmodA<-lm(O3~humidity+temp+ibh+vis,data=ozone[c(1:300),])
cat("The R^2 value of lmodC is: ",summary(lmodA)$r.squared)
```

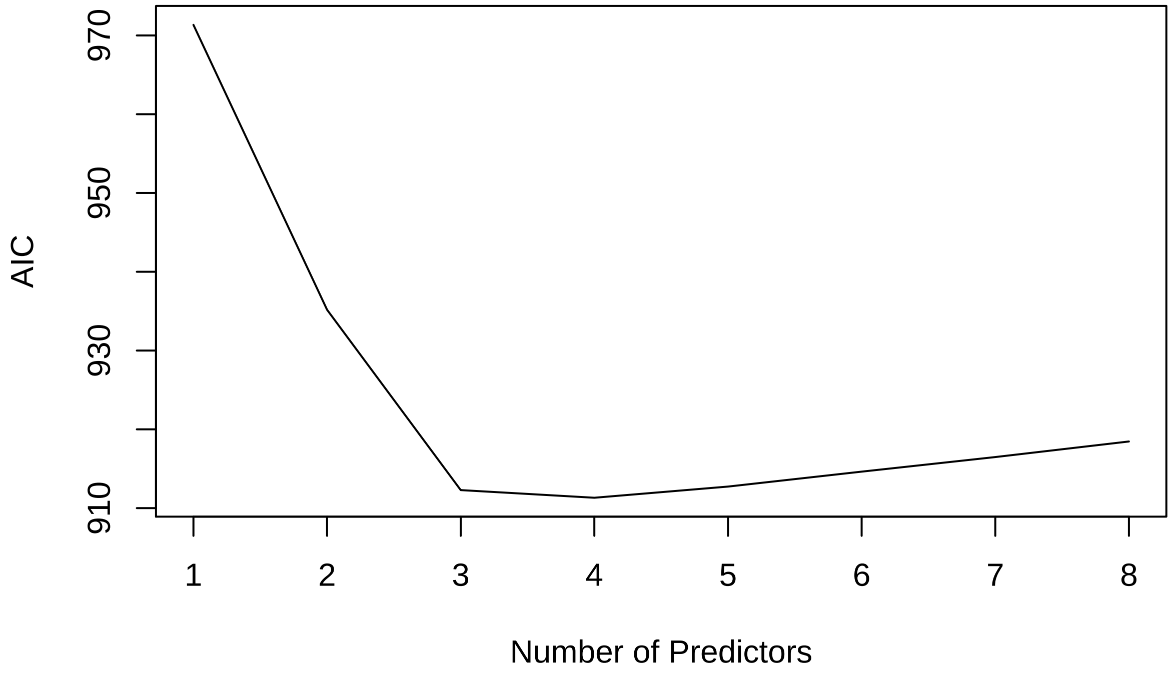
```
## The R^2 value of lmodC is:  0.6913531
```

Recall that the R^2 value of **lmod0** is 0.6986 and that of **lmodA** is 0.6913531 - almost similar values.

Model B

```
library(leaps)
b <- regsubsets(x=lmodB$xs,y=lmodB$y)
rs <- summary(b)
rs$which
AIC <- 300*log(rs$rss/300) + (2:9)*2
plot(AIC ~ I(1:8), ylab="AIC", xlab="Number of Predictors",
type="l")
```

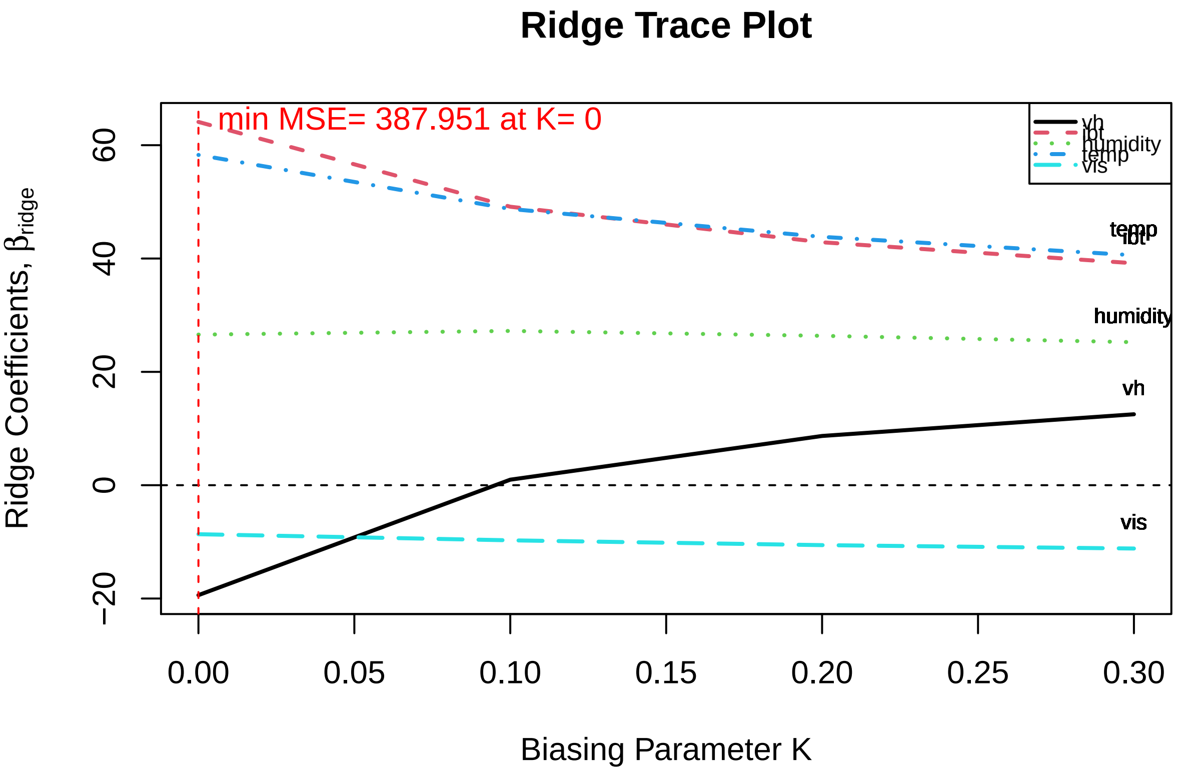
```
##   (Intercept)    vh  wind humidity temp   ibh   dpg   vis   doy
## 1        TRUE FALSE FALSE    FALSE TRUE FALSE FALSE FALSE FALSE
## 2        TRUE FALSE FALSE    FALSE TRUE  TRUE FALSE FALSE FALSE
## 3        TRUE FALSE FALSE     TRUE TRUE  TRUE FALSE FALSE FALSE
## 4        TRUE FALSE FALSE     TRUE TRUE  TRUE FALSE  TRUE FALSE
## 5        TRUE FALSE FALSE     TRUE TRUE  TRUE  TRUE  TRUE FALSE
## 6        TRUE FALSE  TRUE     TRUE TRUE  TRUE  TRUE  TRUE FALSE
## 7        TRUE FALSE  TRUE     TRUE TRUE  TRUE  TRUE  TRUE  TRUE
## 8        TRUE  TRUE  TRUE     TRUE TRUE  TRUE  TRUE  TRUE  TRUE
```



Based on the above plot, we see that for **4** regressors, the **AIC** is minimum. Also, corresponding to **4**, the regressors are **ibh**, **humidty**, **temp** and **vis**.

Hence, the final fitted model is again named **lmodB** and its summary value is printed. here, we again need to find the ridge complexity parameter and using the iterative method, it turns out to be $K = 0.018$

```r
lmodB<-lmridge(O3~vh+ibt+humidity+temp+vis,
data=ozone[1:300,],K=seq(0,0.3,1e-1))
plot(lmodB)
lmodB<-lmridge(O3~ibh+humidity+temp+vis,
data=ozone[1:300,],K=0.018)
summary(lmodB)
```

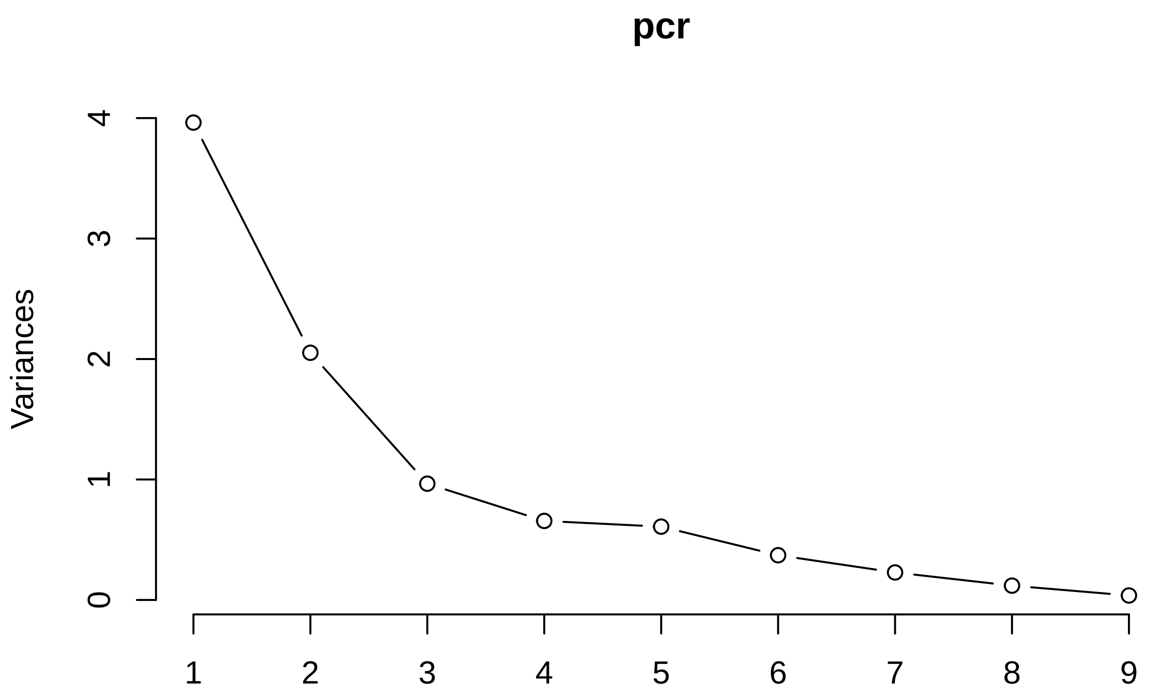


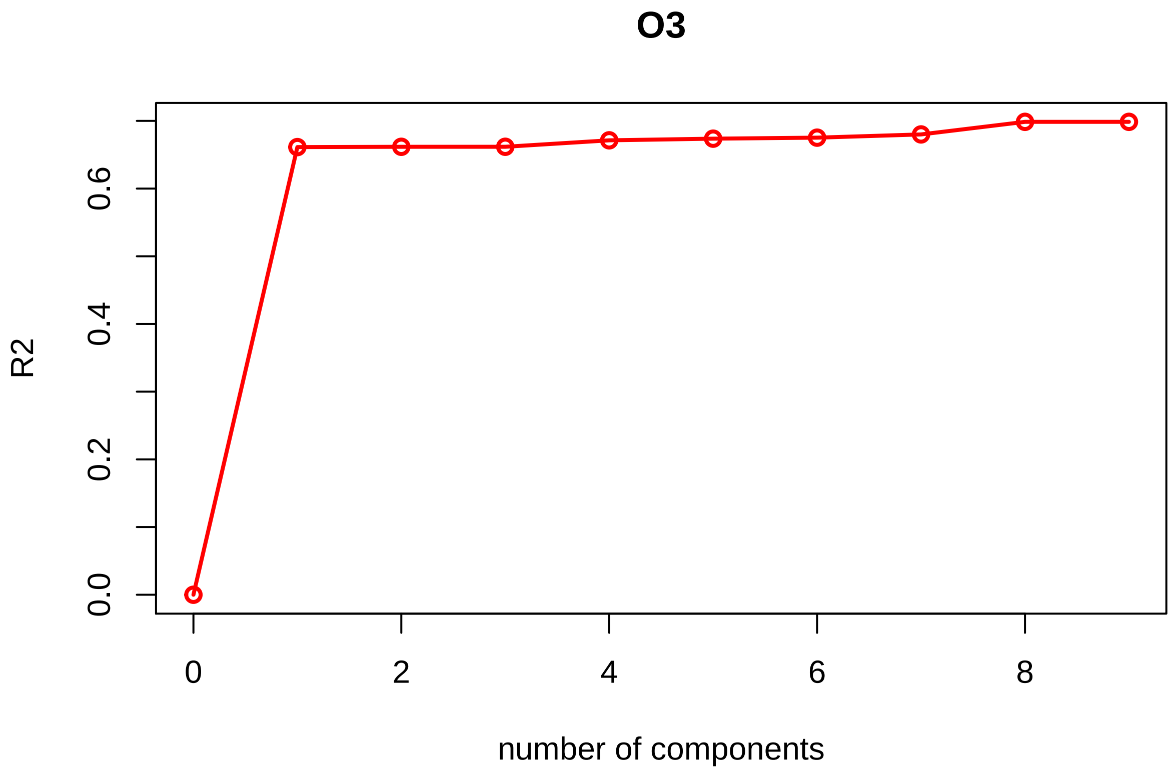
```
##
## Call:
## lmridge.default(formula = O3 ~ humidity + temp + ibh + vis, data = ozone[1:300,
##     ], K = 0.018)
##
##
## Coefficients: for Ridge parameter K= 0.018
##             Estimate Estimate (Sc) StdErr (Sc) t-value (Sc) Pr(>|t|)
## Intercept    -7.8461    76763.2021  13502.8195       5.6850   <2e-16 ***
## humidity      0.0666       22.0888      4.9074       4.5011   <2e-16 ***
## temp          0.3154       80.5392      5.4528      14.7704   <2e-16 ***
## ibh          -0.0010      -32.0672      5.2791      -6.0743   <2e-16 ***
## vis          -0.0070       -9.4865      5.1123      -1.8556   0.0645 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Ridge Summary
##         R2      adj-R2    DF ridge           F         AIC         BIC
##    0.67850     0.67530     3.90232   167.29854   909.23688 2634.82497
## Ridge minimum MSE= 111.0194 at K= 0.018
## P-value for F-test ( 3.90232 , 296.0027 ) = 1.116608e-73
## ----------------------------------------------------------------------
```

Recall that the R^2 value of **lmod0** is 0.6986 and that of **lmodB** is 0.67850 - not significantly lower than the former.

Model C

```
plot(pcr,type="l")
library(pls)
PCR<-pcr(O3~.,data=ozone[1:300,],scale=TRUE)
validationplot(PCR,val.type = "R2",
type="o",col="red",lwd=2)
```





The **scree-plot** gives us the indication of taking the first 4 PCs. We also look at the **validation plot**(validated by R^2) where the cumulative amount of variation in Y explained by the PCs is mostly done by the first PC, with a slight increase with the first 4 PCs. So, we fit a model using the first 4 PCs only.

The final fitted model is again named **lmodC** and its R^2 value is printed.

```
lmodC<-lm(O3~PC1+PC2+PC3+PC4,data=Data)
cat("The value of R^2 is : ",summary(lmodC)$r.squared)
```

```
## The value of R^2 is :  0.6712925
```

Recall that the R^2 value of **lmod0** is 0.6986 and that of **lmodA** is 0.6712925 - not significantly lower than the former.

Heteroscedasticity of Errors

We now look into the homoscedasticity of errors assumption. We use **Breusch-Pagan** test to detect heteroscedasticity and in case of its presence, we wil use **Box-Cox** transformation as a remedy.

The Breusch-Pagan test statistic is asymptotically distributed as χ^2_{p-1}, where p is the number of regressors. It tests whether the variance of the errors from a regression is dependent on the values of the independent variables. In that case, heteroskedasticity is present. If the test statistic has a p-value below the level of significance, $alpha$(=0.01,say), then the null hypothesis of homoscedasticity is rejected and heteroscedasticity is assumed.

The box-cox transformation is given by

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if} \lambda \neq 0, \\ \ln y & \text{if } \lambda = 0 \end{cases}$$

The parameter λ is estimated using the **profile likelihood function** and using **goodness-of-fit tests**.z

Model A

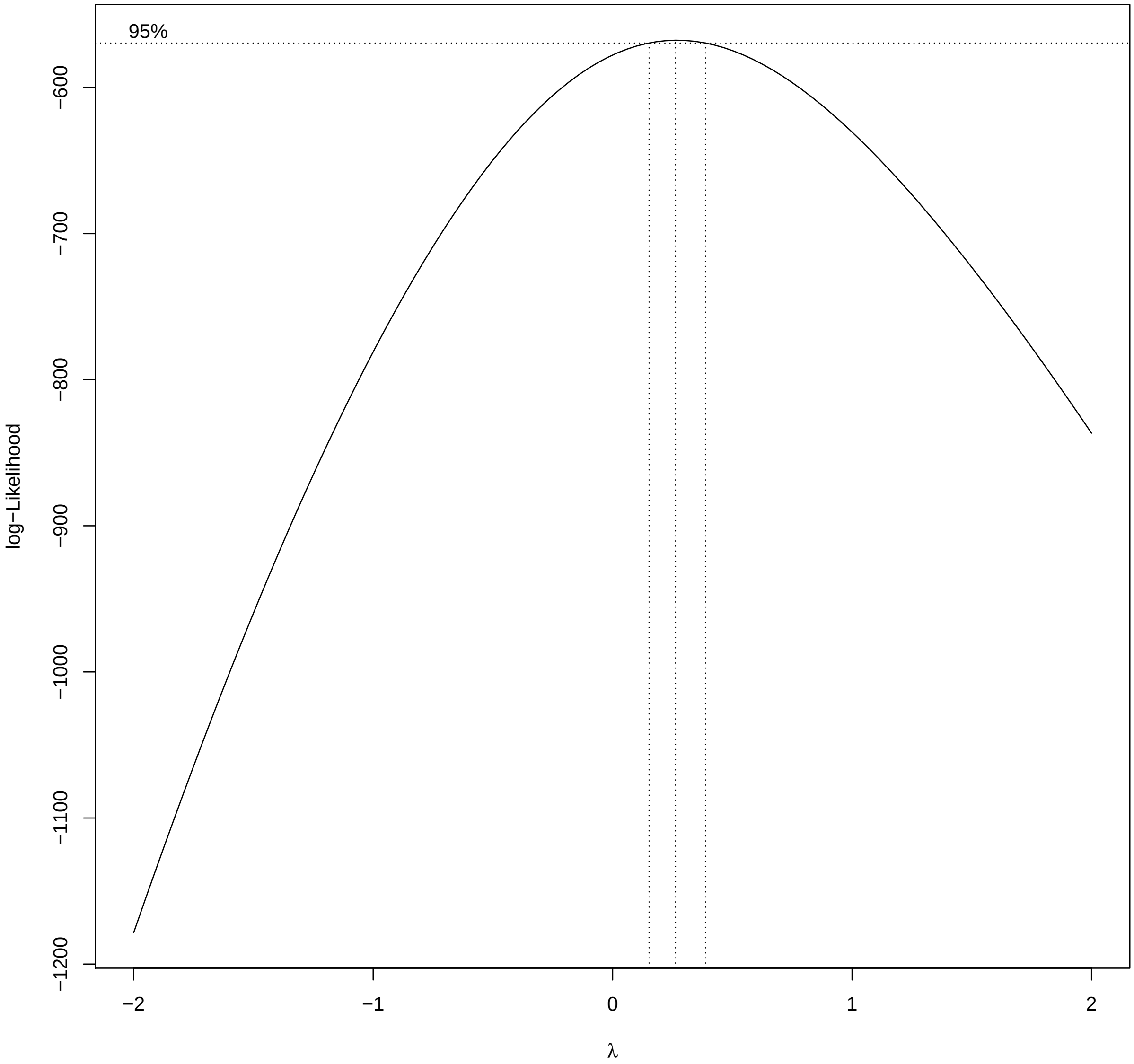
```
install.packages("lmtest")
library(lmtest)
bptest(lmodA)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmodA
## BP = 33.148, df = 5, p-value = 3.517e-06
```

As evident above, the test gets rejected i.e. the *errors are not homoscedastic* based on the data.

We now use the **box-cox** transform as follows -

```
install.packages("MASS")
library(MASS)
ans<-boxcox(lmodA)
lambdaA<-ans$x[which(ans$y==max(ans$y))]
cat("The value of the box-cox paramter is : ",lambdaA)
lmodA<-lm(((O3^lambdaA-1)/lambdaA)~humidity+temp+ibh+vis,data=ozone[1:300,])
```



```
## The value of the box-cox paramter is :  0.2626263
```

Finally, we see if the bp-test gets accepted and see the R^2 value of the new model, say **lmodA**, again.

```
cat("The R^2 value of the transformed model is : ", summary(lmodA)$r.squared)
bptest(lmodA)
```

```
## The R^2 value of the transformed model is :  0.7252028
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmodA
## BP = 8.8891, df = 4, p-value = 0.06393
```

Clearly, the test gets accepted and R^2 value is also significantly better than **lmod0**'s

Model B

```
bptest(lmodB)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmodB
## BP = 30.654, df = 4, p-value = 3.601e-06
```

As evident above, the test gets rejected i.e. the *errors are not homoscedastic* based on the data.

We now use the **box-cox** transform as follows -

```
ans<-boxcox(lmodB)
lambdaB<-ans$x[which(ans$y==max(ans$y))]
lmodB<-lmridge(((O3^lambdaB-1)/lambdaB)~humidity+temp+vis+ibh,
data=ozone[1:300,],K=0.014)
```

Finally, we see if the bp-test gets accepted and see the summary of the new model, say **lmodB**, again.

```
bptest(lmodB)
summary(lmodB)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmodB
## BP = 7.9005, df = 4, p-value = 0.09529
```

```
##
## Call:
## lmridge.default(formula = ((O3^lambdaB - 1)/lambdaB) ~ vis +
##     humidity + temp + ibh, data = ozone[1:300, ], K = 0.007)
##
##
## Coefficients: for Ridge parameter K= 0.007
##             Estimate Estimate (Sc) StdErr (Sc) t-value (Sc) Pr(>|t|)
## Intercept    -0.1044    16157.6186   2308.4514       6.9993   <2e-16 ***
## vis          -0.0011       -1.4521      0.8718      -1.6656   0.0969 .
## humidity      0.0110        3.6527      0.8355       4.3717   <2e-16 ***
## temp          0.0566       14.4549      0.9335      15.4843   <2e-16 ***
## ibh          -0.0002       -6.6921      0.9025      -7.4149   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Ridge Summary
##         R2     adj-R2   DF ridge          F        AIC        BIC
##    0.72020    0.71730    3.96136  196.29516 -161.98770 1563.81906
## Ridge minimum MSE= 3.159141 at K= 0.007
## P-value for F-test ( 3.96136 , 296.0003 ) = 1.53134e-81
## -------------------------------------------------------------------
```

Clearly, the test gets accepted and R^2 value is also significantly better than **lmod0**.

Model C

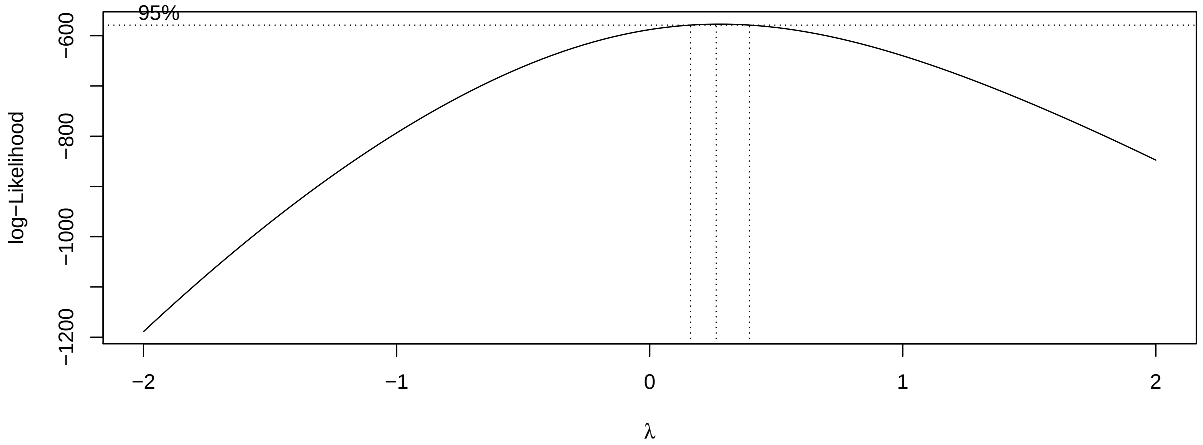
```
bptest(lmodC)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmodC
## BP = 30.719, df = 4, p-value = 3.494e-06
```

As evident above, the test gets rejected i.e. the errors are not homoscedastic based on the data.

We now use the **box-cox** transform as follows -

```
ans<-boxcox(lmodC)
lambdaC<-ans$x[which(ans$y==max(ans$y))]
lmodC<-lm(((ozone[1:300,]$O3^lambdaC-1)/lambdaC)~PC1+PC2+PC3+PC4,data=Data)
```



```
## The value of the box-cox parameter  is :  0.2626263
```

Finally, we see if the bp-test gets accepted and see the R^2 value of the new model, say **lmodC**, again.

```
bptest(lmodC)
cat("The R^2 value of the transformed model is : ", summary(lmodC)$r.squared)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmodC
## BP = 1.6405, df = 4, p-value = 0.8015
```

```
## The R^2 value of the transformed model is :  0.707672
```

Clearly, the test gets accepted and R^2 value is also significantly better.

Normality of Errors

We first see the **normal Q-Q** plot of the residuals and then use the **Shapiro-Wilks** test to confirm whether the errors follow normality or not. The null hypothesis of the **Shapiro-Wilks** test is that the concerned sample is from a normal distribution and the alternative is that the null is false.
So, here, $H_0 : e_1, ...e_n \overset{iid}{\sim} Normal$ vs $H_1 : H_0$ is false.
We reject H_0 if p-value is less than the level of significance, $alpha (= 0.01, say)$ and accept H_0 otherwise.

The test statistic is

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}(x_i - \bar{x})^2}$$

where $x_{(i)}$ is the i^{th} order statistic corresponding to the sample $x_1, ...x_n$,
$\bar{x}$ is the sample mean
The coefficients a_i are given by -

$$(a_1, ..., a_n) = \frac{m^T V^{-1}}{C}$$

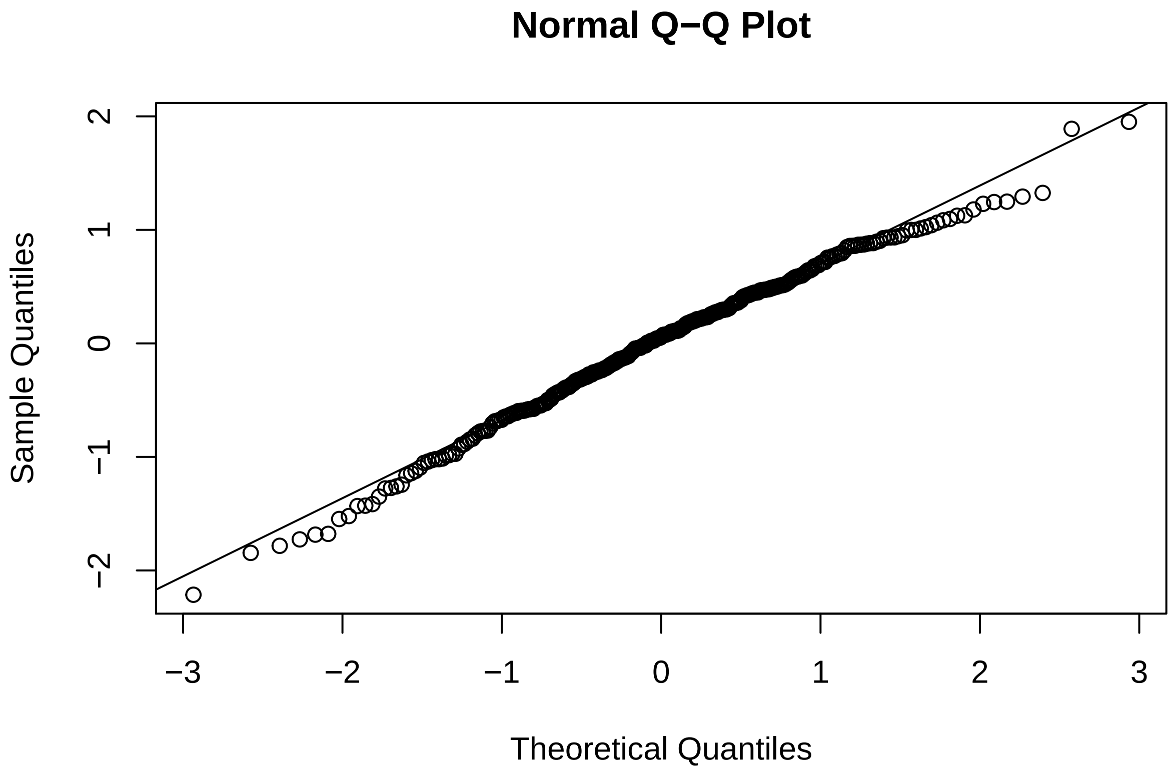
, where C is a vector norm:

$$C = ||V^{-1}m|| = (m^T V^{-1} V^{-1} m)^{1/2}$$

and the vector $m = (m_1, ..., m_n)^T$ is made of the expected values of the order statistics of independent and identically distributed random variables sampled from the standard normal distribution
V is the covariance matrix of those normal order statistics

Model A

```
qqnorm(residuals(lmodA))
qqline(residuals(lmodA))
shapiro.test(residuals(lmodA))
```

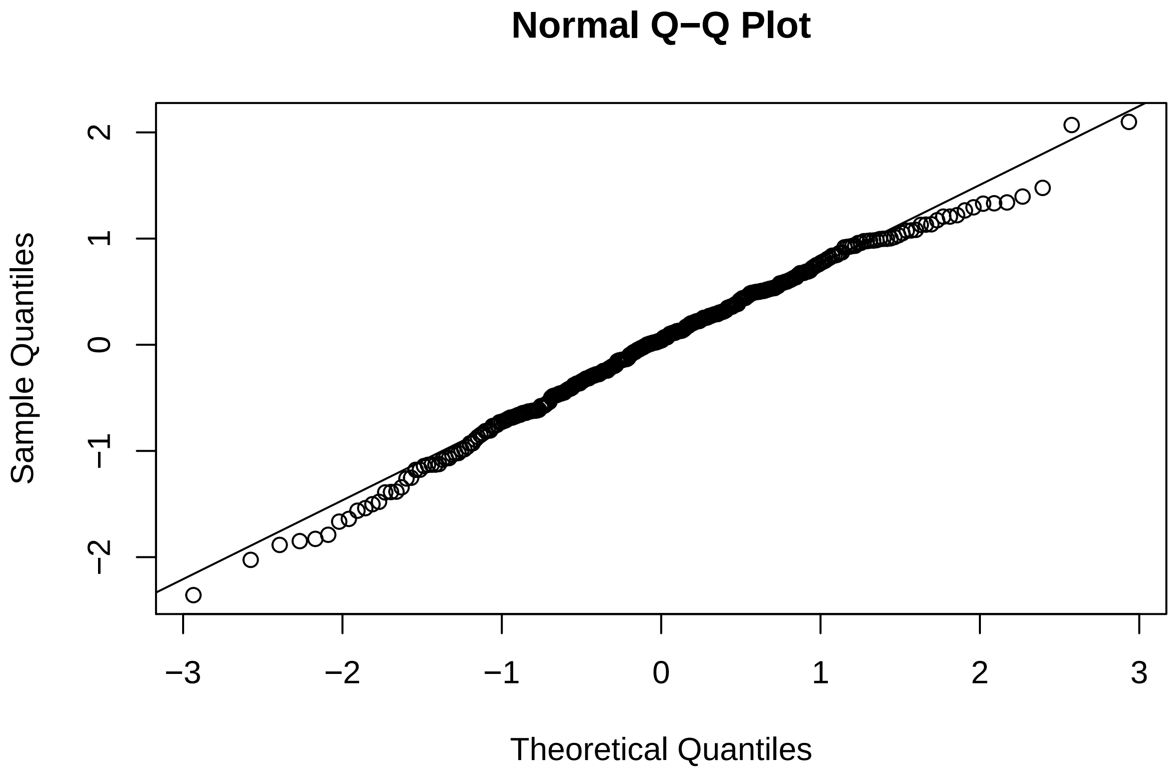


```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmodA)
## W = 0.99121, p-value = 0.07045
```

It is evident from the graph that the *errors follow normality* and it is also *confirmed by the Shapiro-wilks test* as H_0 is accepted.

Model B

```
qqnorm(residuals(lmodB))
qqline(residuals(lmodB))
shapiro.test(residuals(lmodB))
```

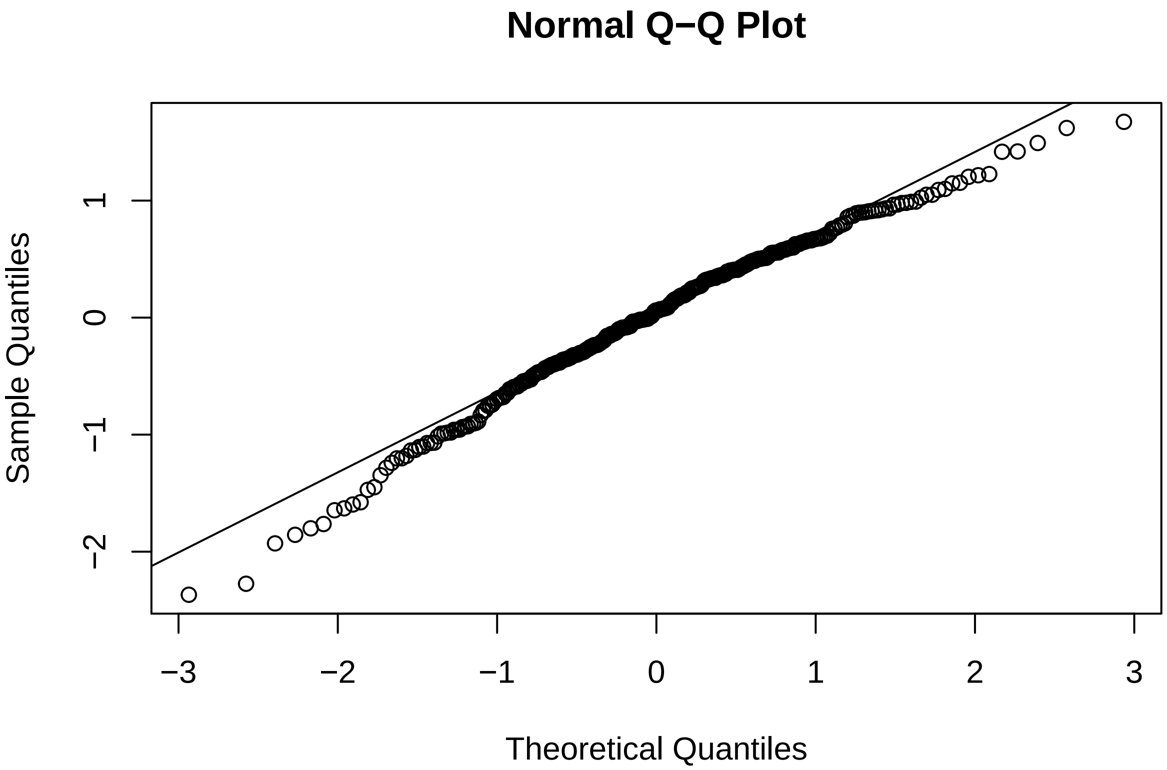


```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmodB)
## W = 0.99211, p-value = 0.1114
```

It is evident from the graph that the *errors follow normality* and it is also *confirmed by the Shapiro-wilks test* as H_0 is accepted.

Model C

```
qqnorm(residuals(lmodC))
qqline(residuals(lmodC))
shapiro.test(residuals(lmodC))
```



```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmodC)
## W = 0.98392, p-value = 0.001924
```

Although the middle part of the $Q-Q$ plot falls in the straight line, the endings are significantly far from the theroretical quantiles and hence, graphically, the errors do not follow normality. This is confirmed by the **Shapiro-Wilks** test as well.

Autocorrelation

We first look at the plots of $\epsilon_t\ vs.\ \epsilon_{t-1}$ and see if there's a high correlation between them. Then we will use **Durbin-Watson** test to confirm the presence of autocorrelation.

If e_t is the residual given by $e_t = \rho e_{t-1} + \nu_t$, the **Durbin-Watson** test tests $H_0 : \rho = 0$ vs. $H_1 : \rho \neq 0$

The test statistic is -

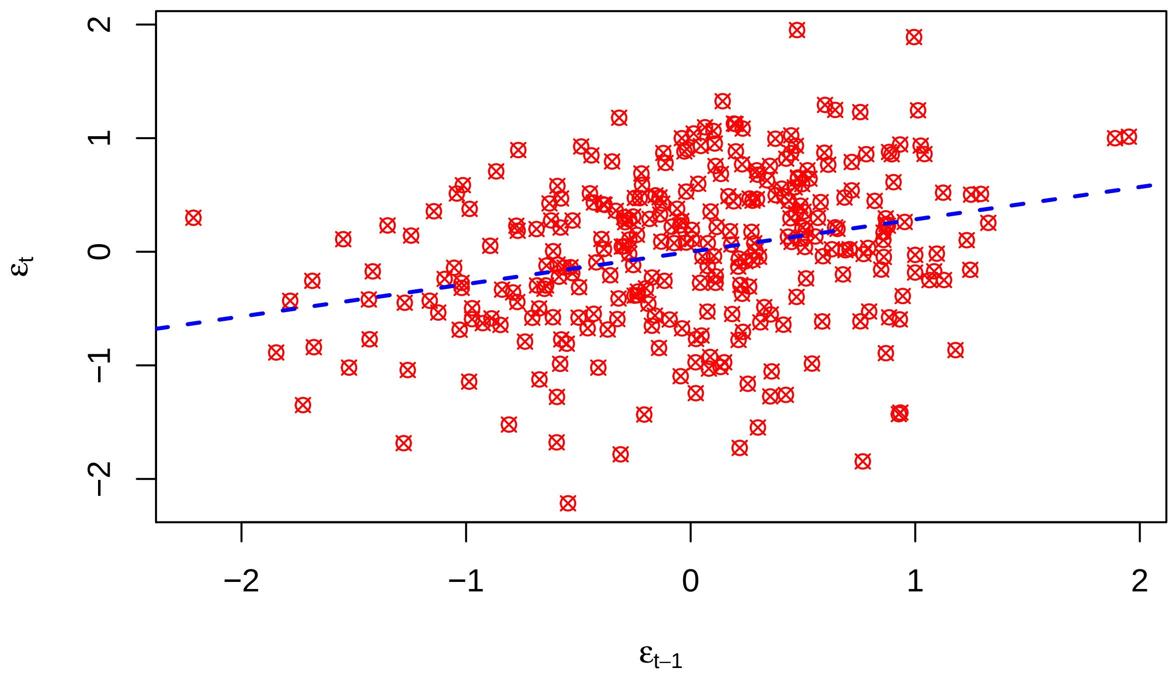
$$d = \frac{\sum_{t=2}^{T}(e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2}$$

where T is the number of observations.

We reject H_0 if p-value is less than the level of significance, $\alpha(= 0.01, say)$ and accept H_0 otherwise.

Model A

```
plot(residuals(lmodA)[-1],residuals(lmodA)[-length(residuals(lmodA))],
xlab=expression(epsilon[t-1]),ylab=expression(epsilon[t]),
pch=13,col="red")
abline(lm(residuals(lmodA)[-length(residuals(lmodA))]~
residuals(lmodA)[-1]),lty=2,lwd=2,col="blue")
dwtest(lmodA)
```

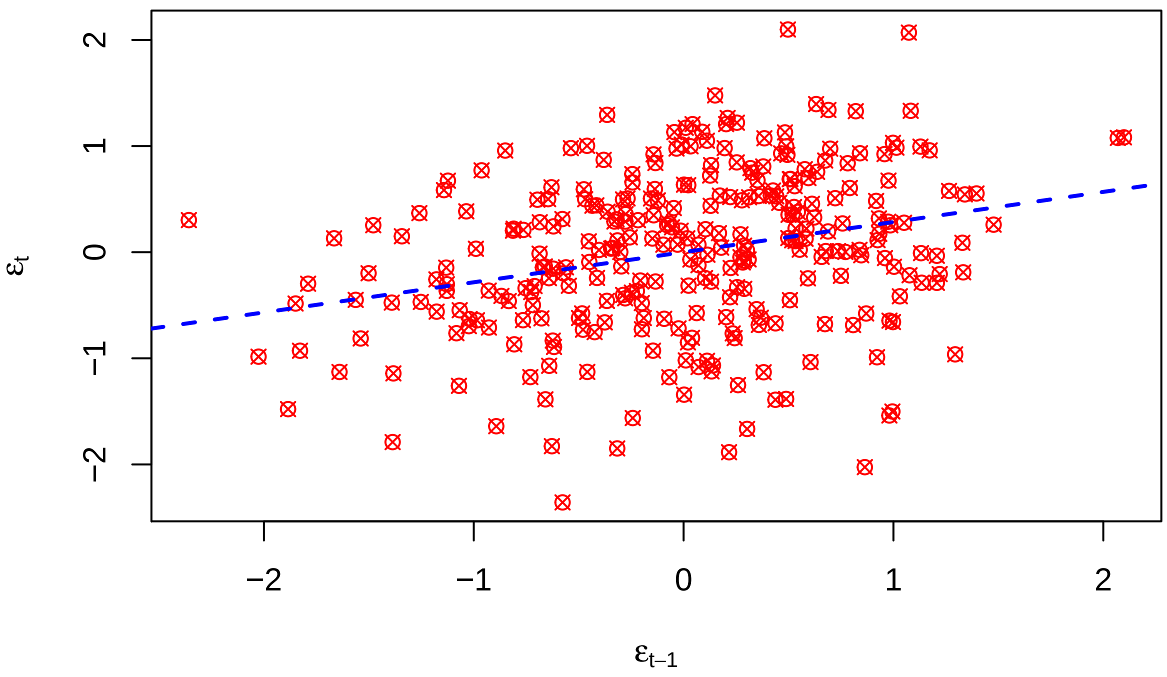


##

```
##  Durbin-Watson test
##
## data:  lmodA
## DW = 1.4288, p-value = 1.824e-07
## alternative hypothesis: true autocorrelation is greater than 0
```

Model B

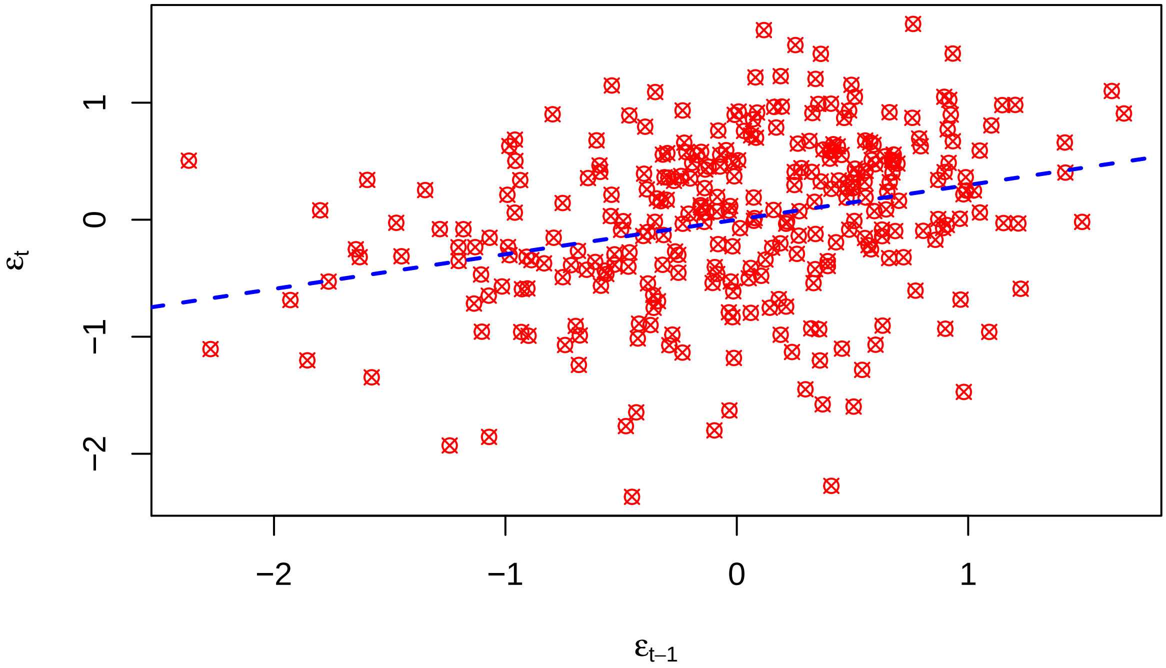
```
plot(residuals(lmodB)[-1],residuals(lmodB)[-length(residuals(lmodB))],
xlab=expression(epsilon[t-1]),ylab=expression(epsilon[t]),
pch=13,col="red")
abline(lm(residuals(lmodB)[-length(residuals(lmodB))]~
residuals(lmodB)[-1]),lty=2,lwd=2,col="blue")
dwtest(lmodB)
```



```
##
##  Durbin-Watson test
##
## data:  lmodB
## DW = 1.4314, p-value = 2.054e-07
## alternative hypothesis: true autocorrelation is greater than 0
```

Model C

```
plot(residuals(lmodC)[-1],residuals(lmodC)[-length(residuals(lmodC))],
xlab=expression(epsilon[t-1]),ylab=expression(epsilon[t]),
pch=13,col="red")
abline(lm(residuals(lmodC)[-length(residuals(lmodC))]~
residuals(lmodC)[-1]),lty=2,lwd=2,col="blue")
dwtest(lmodC)
```

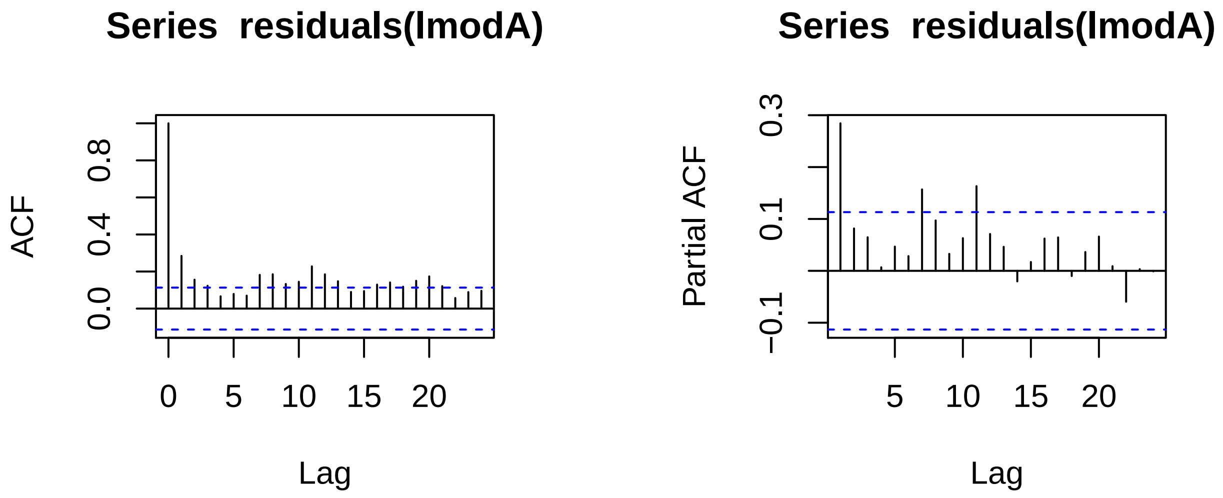


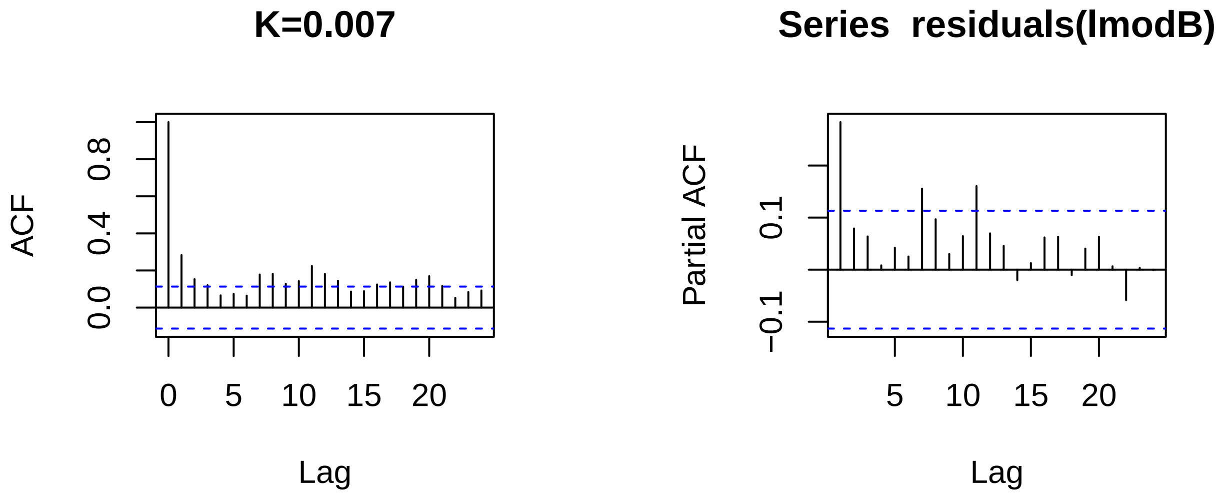
```
##
##  Durbin-Watson test
##
## data:  lmodC
## DW = 1.407, p-value = 5.955e-08
## alternative hypothesis: true autocorrelation is greater than 0
```

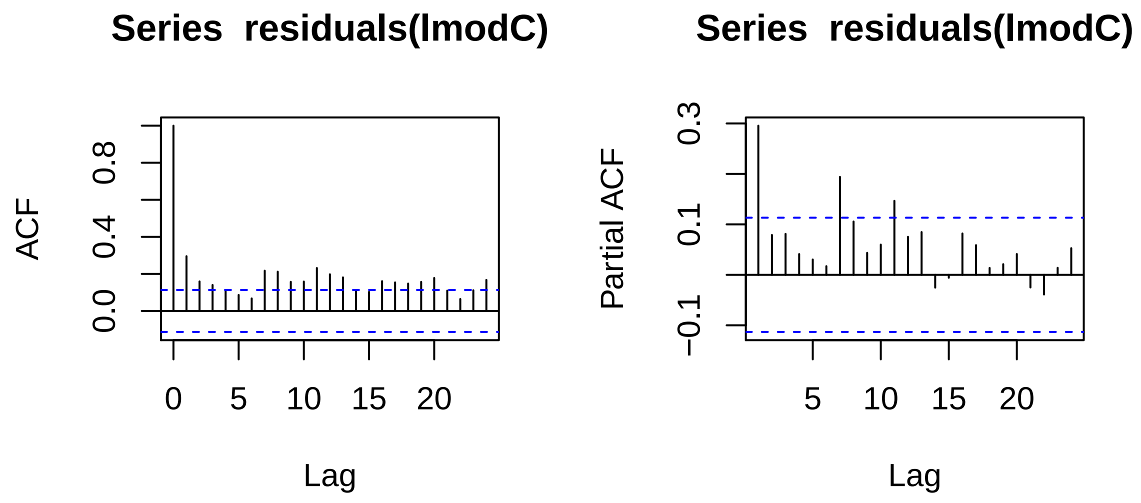
All the models have auto-correlated residuals. Assuming **AR(p)** model for the errors, we fitted models for $p=$1-20. None performed satisfactorily i.e. none achieved stationarity.

We look at the **acf** and the **pacf** plots of the residuals of each model to see if $AR(p)$ is indeed a good model or not for the errors.

```
par(mfrow=c(1,2))
acf(residuals(lmodA));pacf(residuals(lmodA))
par(mfrow=c(1,2))
acf(residuals(lmodB));pacf(residuals(lmodB))
par(mfrow=c(1,2))
acf(residuals(lmodC));pacf(residuals(lmodC))
```







Clearly, $AR(p)$ model does not seem to be a good model for the erros.

Instead, we used the **auto.arima** function in the **forecast** package in **R** that automatically fits an **ARIMA(p,d,q)** process by taking that value of **d** such that stationarity is achieved and **p** and **q** are chosen so that minimum **AIC** is achieved.

In model **A**, an **ARIMA(0,1,2)** model is fitted. We do not take any remedial measure for model **B** and **C** as the problem then becomes too complicated.

```
library(forecast)
(modelA<-auto.arima(y=(ozone[c(1:300),1]^lambdaA-1)/lambdaA,
xreg=model.matrix(lmodA)[,-1],
max.p=7,max.q=7,max.d=7))
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Series: (ozone[c(1:300), 1]^lambdaA - 1)/lambdaA
## Regression with ARIMA(0,1,1) errors
##
## Coefficients:
##           ma1   drift  humidity    temp     ibh      vis
##       -0.9154  0.0017    0.0042  0.0528  -2e-04  -0.0018
## s.e.   0.0241  0.0023    0.0024  0.0041   0e+00   0.0006
##
## sigma^2 estimated as 0.4325:  log likelihood=-296.85
## AIC=607.7   AICc=608.08   BIC=633.6
```

Now, we fit the final models as **modA** and retain model B as **lmodB** and model C as **lmodC**. The summary of **modC** is printed below as well.

```r
modA<-arima(x=(ozone[c(1:300),1]^lambdaA-1)/lambdaA,
xreg=model.matrix(lmodA)[,-1],
order=c(0,1,2))
summary(modA)
```

```
##
## Call:
## arima(x = (ozone[c(1:300), 1]^lambdaA - 1)/lambdaA, order = c(0, 1, 2), xreg = model.matrix(lmodA)[,
##     -1])
##
## Coefficients:
##           ma1      ma2  humidity    temp     ibh      vis
##       -0.7510  -0.1701    0.0038  0.0523  -2e-04  -0.0016
## s.e.   0.0599   0.0592    0.0022  0.0021     NaN   0.0005
##
## sigma^2 estimated as 0.4129:  log likelihood = -292.9,  aic = 599.79
##
## Training set error measures:
##                      ME      RMSE       MAE  MPE MAPE      MASE         ACF1
## Training set 0.02050877 0.6414822 0.5091768 -Inf  Inf 0.6694062 -0.001645668
```

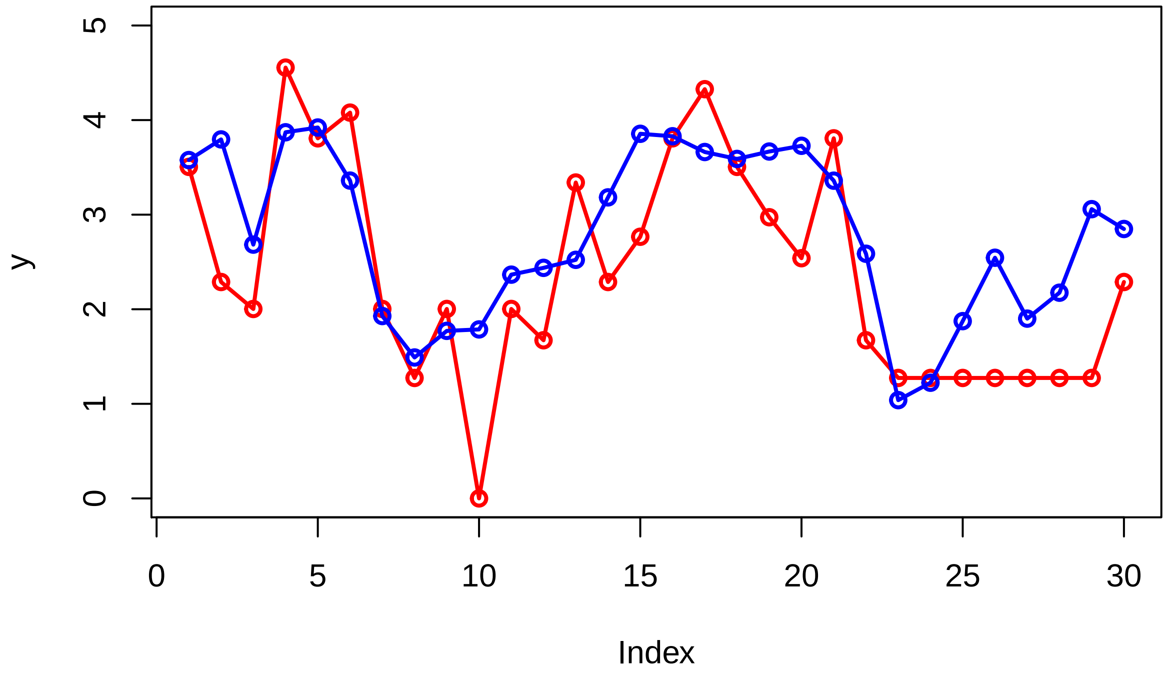
Possibly better models may be fitted after a course on *Time Series Analysis*.

Prediction

We finally predict using our models - **modA**, **lmodB** and **lmodC** with the test data, i.e the last 20% of the **ozone** dataset. We will use $RMSE = \sqrt{\frac{1}{n}\sum_i (y - \hat{y}_i)^2}$ as a metric to compare our models. The best of these three models will be the one with smaller.

Model A

```
y<-ozone[301:330,1]
y<-(y^lambdaA-1)/lambdaA
y_pred<-as.vector(predict(modA,
newxreg=ozone[301:330,c(4,5,6,9)])[[1]])
plot(y,type="o",col="red",ylim=c(0,5),lwd=2)
lines(y_pred,col="blue",type="o",lwd=2)
cat("The RMSE of model A is : ",sqrt(mean((y-y_pred)^2)))
```

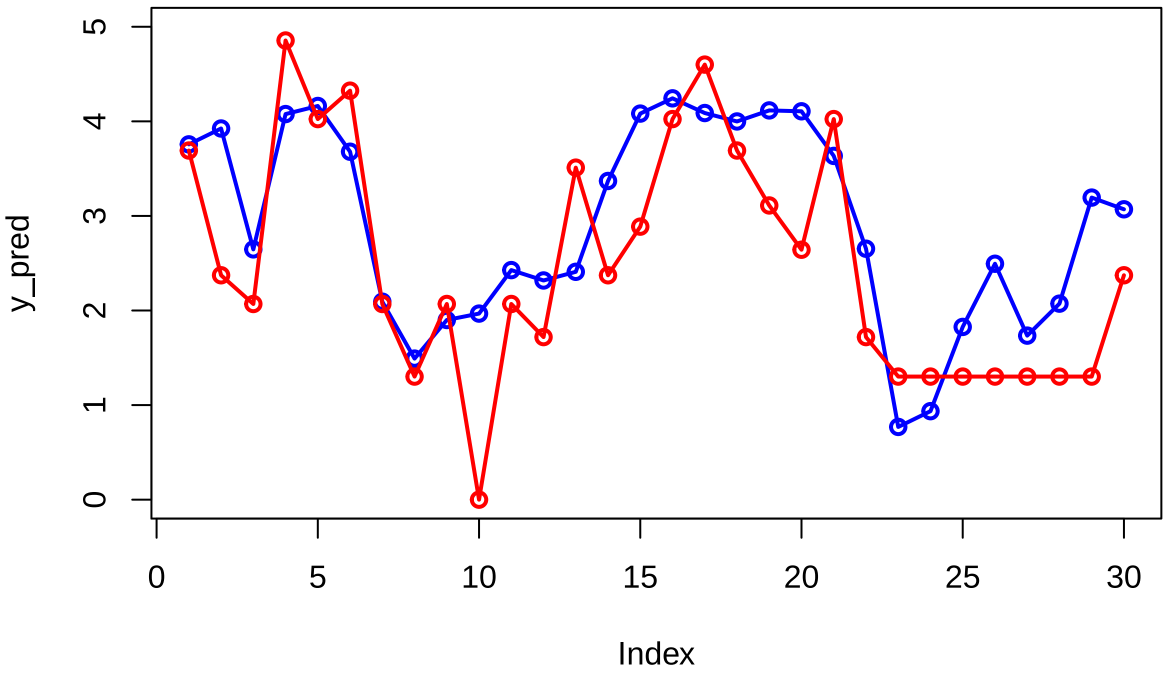


```
## The RMSE of model A is :  0.8272072
```

Model B

```
y<-ozone[301:330,1]
y<-(y^lambdaB-1)/lambdaB
```

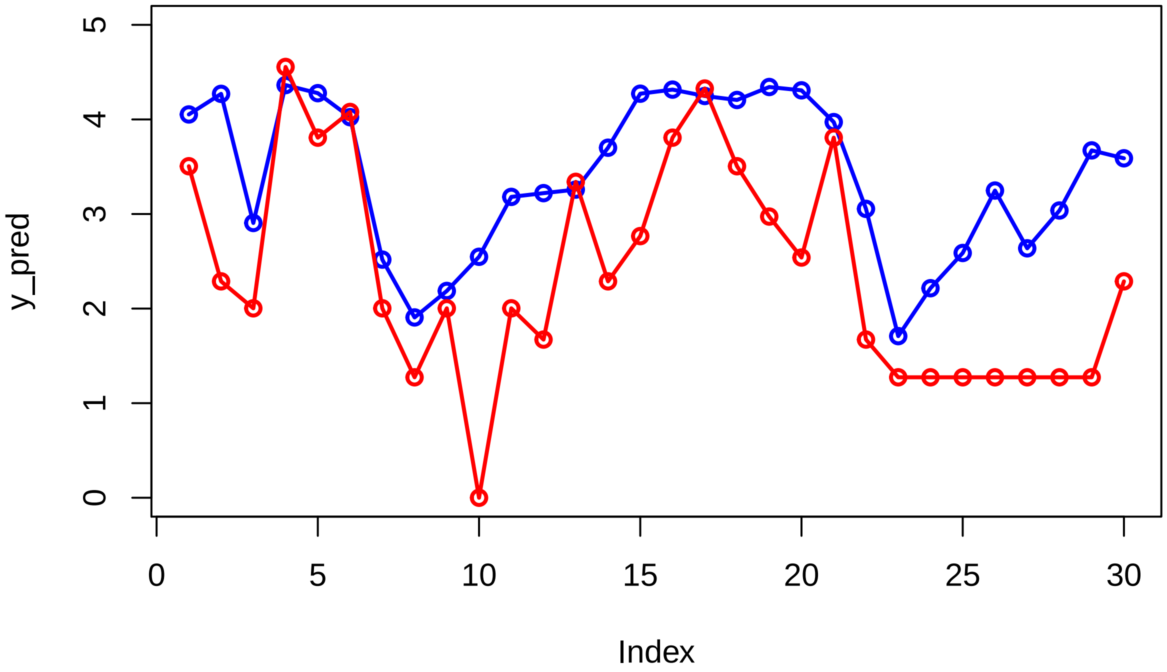
```
y_pred<-predict(lmodB,ozone[301:330,-1],
type="response")
plot(y_pred,type="o",col="blue",ylim=c(0,5),lwd=2)
lines(y,col="red",type="o",lwd=2)
cat("The RMSE of model B is : ",sqrt(mean((y-y_pred)^2)))
```



```
## The RMSE of model B is :  0.883063
```

Model C

```
y<-ozone[301:330,1]
y<-(y^lambdaC-1)/lambdaC
PCR<-pcr((O3^lambdaC-1)/lambdaC~.,data=ozone[1:300,],
scale=TRUE,ncomp=1)
y_pred<-predict(PCR,ozone[301:330,-1])
plot(y_pred,type="o",col="blue",ylim=c(0,5),lwd=2)
lines(y,col="red",type="o",lwd=2)
cat("The RMSE of Model C is: ",sqrt(mean((y-y_pred)^2)))
```



```
## The RMSE of Model C is:  1.25652
```

So, based on the RMSE values, model **B** performs best, even without auto-correlation correction, with model **A** being a close competitor. Model **C** performs comparatively poor, evident from the graph as well as **RMSE** value. A poor model for autocorrelation may be a reason.

Non-parametric Steup : Alternating Conditional Expectation(ACE)

We apply the **ACE** algorithm on the **ozone dataset** and see how it performs in terms of **RMSE**.

The mathematical description of the algorithm is as follows - Suppose we predict Y using $X_1, ..., X_p$. Suppose $\theta(Y), \phi_1(X_1)..., \phi_p(X_p)$ are **zero-mean functions** and with these transformation functions, the fraction of variance of $\theta(Y)$ not explained is -

$$e^2(\theta, \phi_1, .., \phi_p) = \frac{E[\theta(Y) - \sum_{i=1}^{p} \phi_i(X_i)]^2}{E[\theta^2(Y)]}$$

Generally, the optimal transformations that minimize the unexplained part are difficult to compute directly. As an alternative, ACE is an iterative method to calculate the optimal transformations. The procedure of ACE has the following steps:

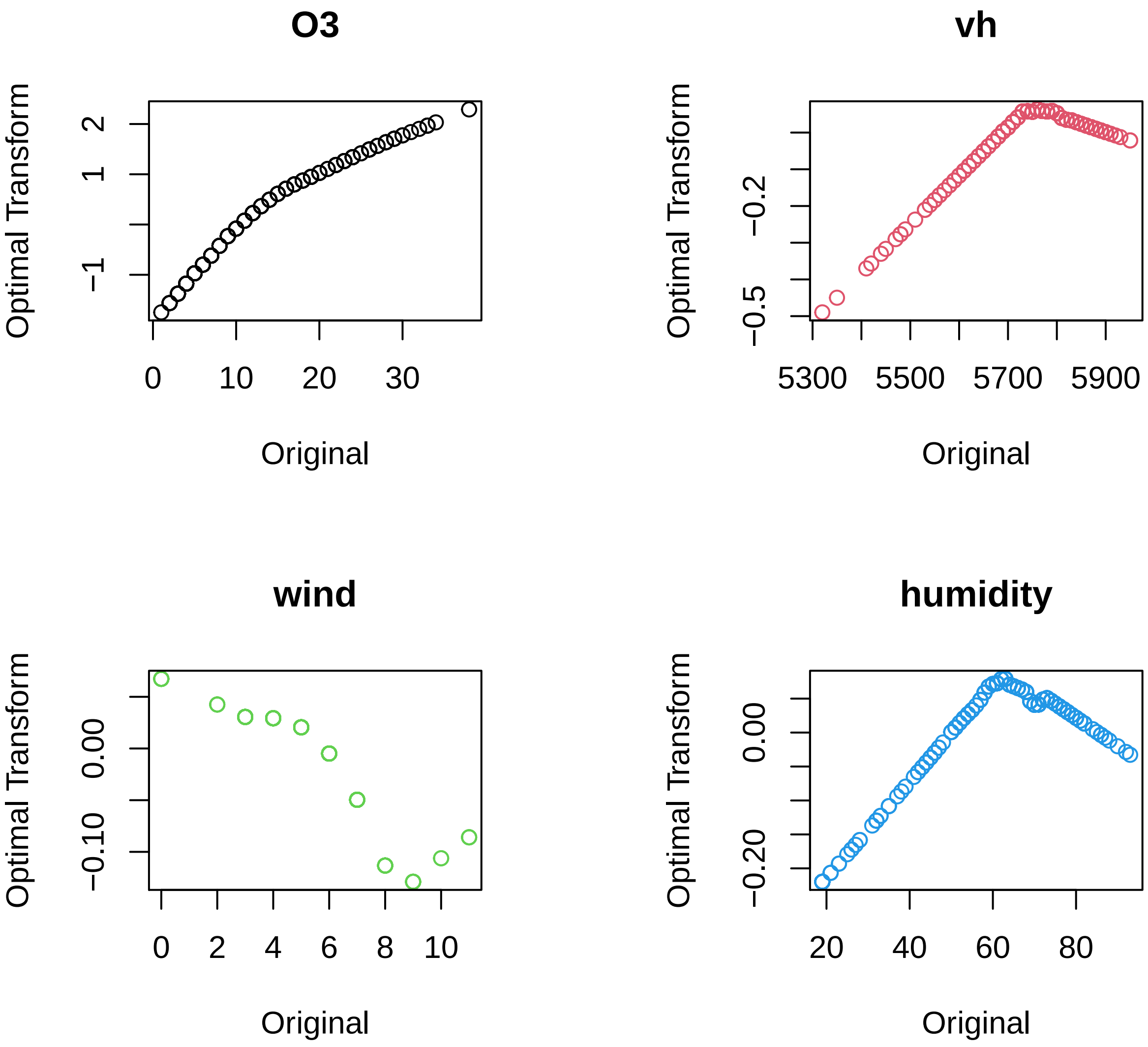
1. Hold $\phi_1(X_1), ..., \phi_p(X_p)$ fixed, minimizing e^2 gives $\theta_1(Y) = E[\sum_{i=1}^{p} \phi_i(X_i)|Y]$
2. Normalize $\theta_1(Y)$ to unit variance.
3. Fix k, fix other $\phi_i(X_i)$ and $\theta(Y)$, minimizing e^2 and the solution is $\tilde{\phi}_k = E[\theta(Y) - \sum_{i \neq k} \phi_i(X_i)|X_k]$
4. Iterate the above three steps until e^2 is within error tolerance.

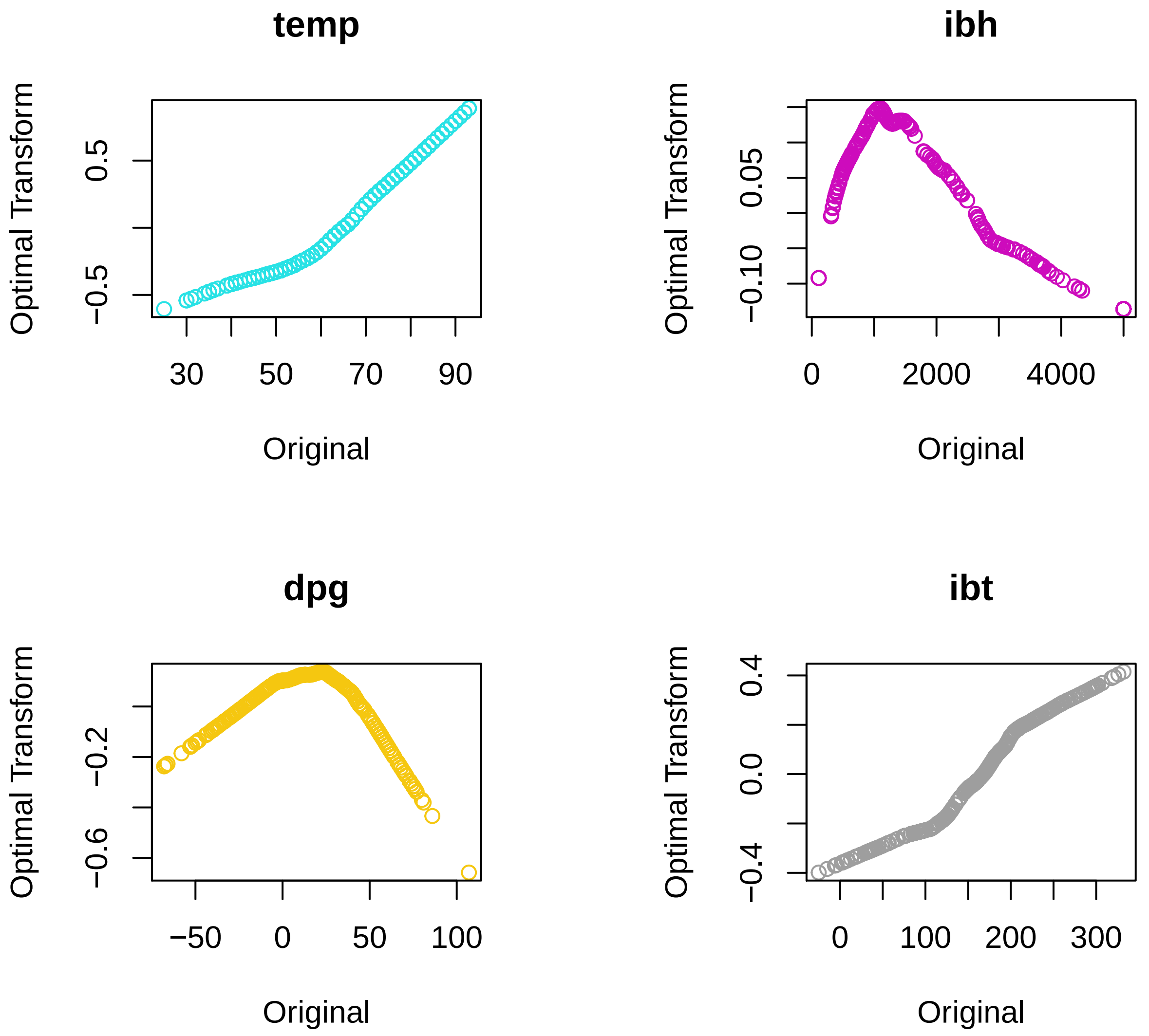
We first store the transformed variables use **ACE** algorithm

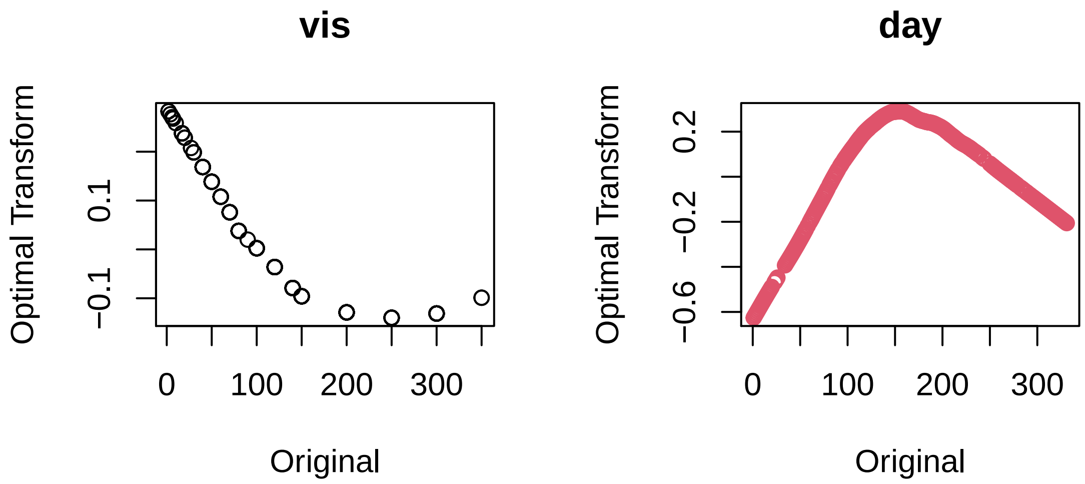
```
library(acepack) # for the ACE algorithm
final<-ace(x=as.matrix(ozone[1:300,-1]),
y=ozone[1:300,1]) # storing the optimal transformed variables
Data<-data.frame(O3=final$ty,final$tx)
```

We look at the resulting transformations plotting the transformed vs original variables.

```
par(mfrow=c(2,5))
for (i in 1:10) plot(ozone[1:300,i],
Data[,i],col=i,xlab="Original",ylab="Optimal Transform",main=names(Data)[i])
```


O3
Optimal Transform
−1
1
2
0
10
20
30
Original
vh
Optimal Transform
−0.5
−0.2
5300
5500
5700
5900
Original
wind
Optimal Transform
−0.10
0.00
0
2
4
6
8
10
Original
humidity
Optimal Transform
−0.20
0.00
20
40
60
80
Original


temp
Optimal Transform
0.5
−0.5
30
50
70
90
Original
ibh
Optimal Transform
0.05
−0.10
0
2000
4000
Original
dpg
Optimal Transform
−0.2
−0.6
−50
0
50
100
Original
ibt
Optimal Transform
0.4
0.0
−0.4
0
100
200
300
Original



Next, we fit a linear model to the optimal transformed dataset and look into the summary of the fitted model **lmod**.

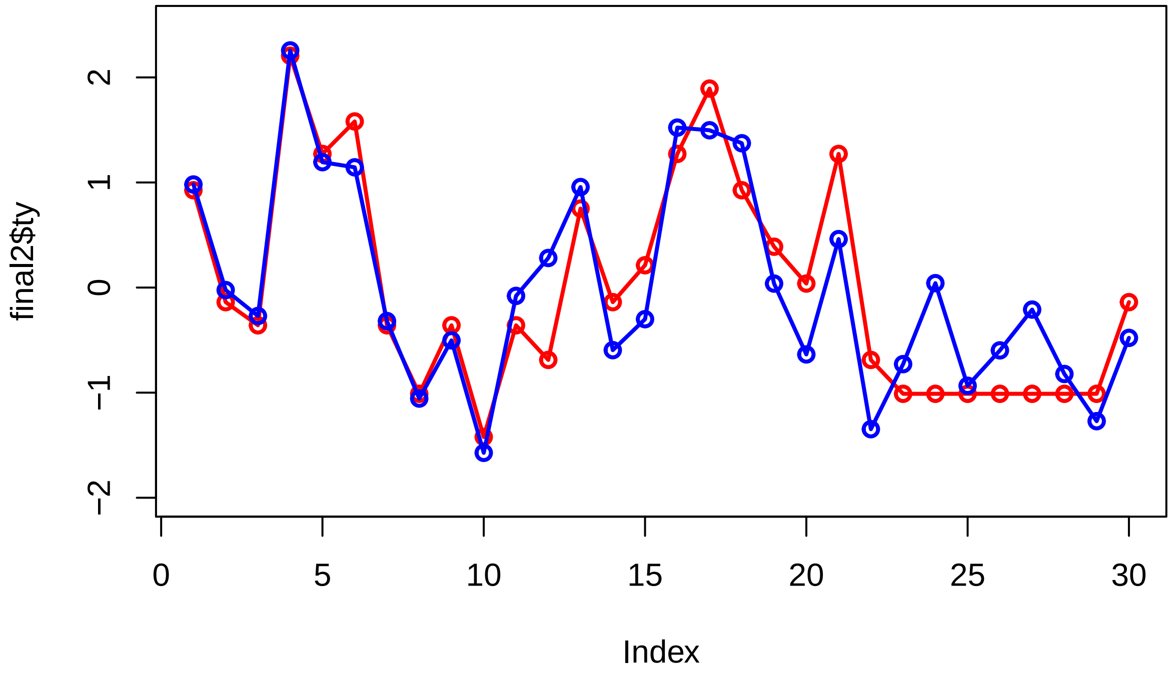
```
lmod<-lm(O3~.,data=Data) # fitting a model after optimal transformations
cat("The R^2 value of lmod is : ",summary(lmod)$r.squared)
```

```
## The R^2 value of lmod is :  0.8406105
```

Clearly, the R^2 value is quite high compared to our previous models.

Now, we finally predict using **lmod** and compute its RMSE.

```
final2 <- ace(x=as.matrix(ozone[301:330,-1]),y=ozone[301:330,1]) # test data
New <- data.frame(final2$tx)
y_pred<-as.vector(predict(lmod,newdata=New,type="response")) # predicted values
plot(final2$ty,type="o",col="red",ylim=c(-2,2.5),lwd=2) # original vs predicted
lines(y_pred,type="o",col="blue",lwd=2)
cat("The RMSE  value of lmod is: ",sqrt(mean((final2$ty-y_pred)^2)))
```



```
## The RMSE  value of lmod is:  0.4516001
```

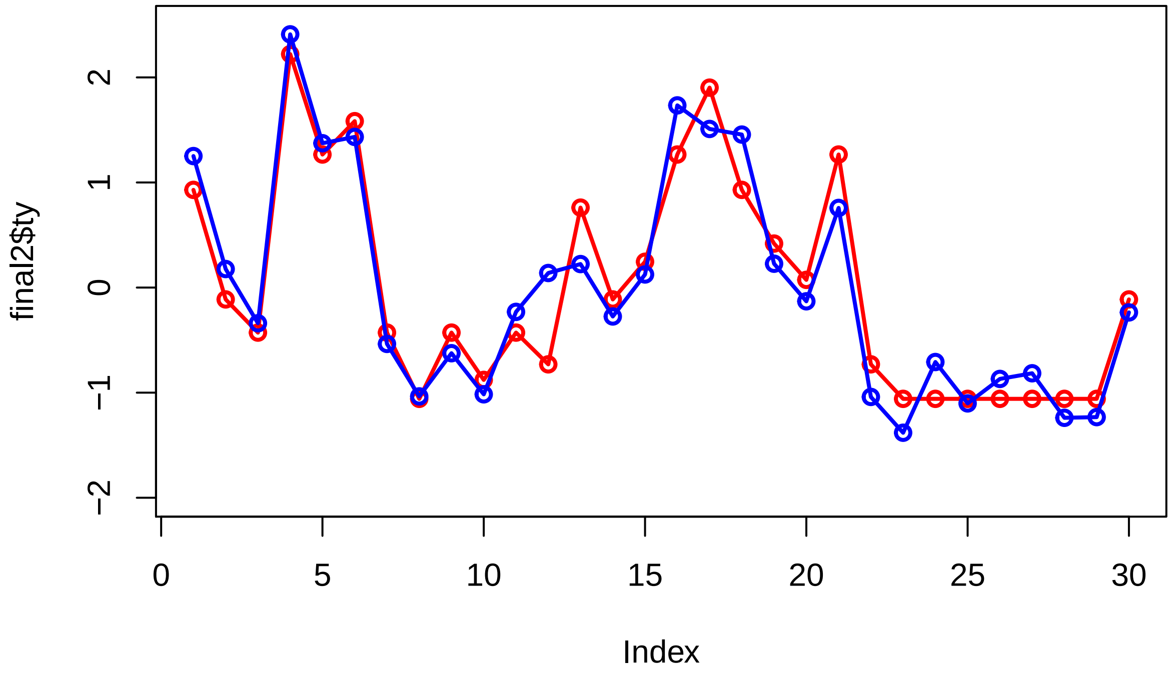
The RMSE value is 0.45 - quite remarkable on comparison with **modA**, **lmodB** and **modC**.

Based on previous experience, we know that **ibt** and **temp** are almost perfectly correlated and **vh** showed a similar relationship with either of them.

We again fit a linear model, **final**, based on the transformed data, removing **ibt** and **vh**.

```
final<-ace(x=as.matrix(ozone[1:300,-c(1,2,8)]),
y=ozone[1:300,1]) # storing the optimal transformed variables
Data<-data.frame(O3=final$ty,final$tx)
lmod<-lm(O3~.,data=Data) # fitting a model after optimal transformations
cat("The R-squared value of the final model is: ",summary(lmod)$r.squared)
final2 <- ace(x=as.matrix(ozone[301:330,-c(1,2,8)]),
y=ozone[301:330,1]) # test data
New <- data.frame(final2$tx)
y_pred<-as.vector(predict(lmod,newdata=New,type="response")) # predicted values
plot(final2$ty,type="o",col="red",ylim=c(-2,2.5),lwd=2) # original vs predicted
lines(y_pred,type="o",col="blue",lwd=2)
cat("The RMSE  value of final is: ",sqrt(mean((final2$ty-y_pred)^2)))
```

```
## The R-squared value of the final model is:  0.8271309
```



```
## The RMSE  value of lmod is:  0.3132212
```

The R^2 value of the **final** model is 0.82 and the RMSE value of the model is 0.31 - both significantly better than our previous models.

As a final check, we see if our *final **model has any problem of** multicollinearity**,** heteroscedasticity **of errors,** non-normality **of errors and** auto-coorelation** of errors.

```
vif(lmod)
shapiro.test(residuals(lmod)) # accepted
bptest(lmod)
dwtest(lmod,alternative="two.sided")
```

```
##     wind humidity     temp      ibh      dpg      vis      day
## 1.144903 1.468402 1.643312 1.750943 1.313939 1.354353 1.703137
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lmod)
## W = 0.99533, p-value = 0.5051
```

```
##
##  studentized Breusch-Pagan test
```

```
##
## data:  lmod
## BP = 15.308, df = 7, p-value = 0.03225
```

```
##
##  Durbin-Watson test
##
## data:  lmod
## DW = 1.7443, p-value = 0.01524
## alternative hypothesis: true autocorrelation is not 0
```

As evident from the above tests, the **final** model accepts all tests and seems to be an ideal model compared to the previous models.

Final Remarks

Bibliography

1. Leo Breiman & Jerome H. Friedman (1985): Estimating Optimal Transformations for Multiple Regression and Correlation, Journal of the American Statistical Association, 80:391, 580-598
2. Jolliffe, Ian T. (1982). “A note on the Use of Principal Components in Regression”. Journal of the Royal Statistical Society, Series C. 31 (3): 300–303. doi:10.2307/2348005. JSTOR 2348005.
3. Sung H. Park (1981). “Collinearity and Optimal Restrictions on Regression Parameters for Estimating Responses”. Technometrics. 23 (3): 289–295. doi:10.2307/1267793.
4. Wilkinson, L., & Dallal, G.E. (1981). Tests of significance in forward selection regression with an F-to enter stopping rule. Technometrics, 23, 377–380
5. Akaike, H. (1973), “Information theory and an extension of the maximum likelihood principle”, in Petrov, B. N.; Csáki, F. (eds.), 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, September 2-8, 1971, Budapest: Akadémiai Kiadó, pp. 267–281. Republished in Kotz, S.; Johnson, N. L., eds. (1992), Breakthroughs in Statistics, I, Springer-Verlag, pp. 610–624.
6. Akaike, H. (1974), “A new look at the statistical model identification”, IEEE Transactions on Automatic Control, 19 (6): 716–723, doi:10.1109/TAC.1974.1100705, MR 0423716.
7. Sugiura, N. (1978), “Further analysis of the data by Akaike’s information criterion and the finite corrections”, Communications in Statistics - Theory and Methods, 7: 13–26, doi:10.1080/03610927808827599.
8. Shapiro, S. S.; Wilk, M. B. (1965). “An analysis of variance test for normality (complete samples)”. Biometrika. 52 (3–4): 591–611. doi:10.1093/biomet/52.3-4.591. JSTOR 2333709. MR 0205384. p. 593
9. Breusch, T. S.; Pagan, A. R. (1979). “A Simple Test for Heteroskedasticity and Random Coefficient Variation”. Econometrica. 47 (5): 1287–1294. doi:10.2307/1911963. JSTOR 1911963. MR 0545960.
10. Box, George E. P.; Cox, D. R. (1964). “An analysis of transformations”. Journal of the Royal Statistical Society, Series B. 26 (2): 211–252. JSTOR 2984418. MR 0192611.
11. Durbin, J.; Watson, G. S. (1950). “Testing for Serial Correlation in Least Squares Regression, I”. Biometrika. 37 (3–4): 409–428. doi:10.1093/biomet/37.3-4.409. JSTOR 2332391
12. Durbin, J.; Watson, G. S. (1951). “Testing for Serial Correlation in Least Squares Regression, II”. Biometrika. 38 (1–2): 159–179. doi:10.1093/biomet/38.1-2.159. JSTOR 2332325
13. Faraway, J.J. (2004). Linear Models with R (1st ed.). Chapman and Hall/CRC. https://doi.org/10.4324/9780203507278
14. Hoerl, A. E., Kennard, R. W. and Baldwin, K. F. (1975). Ridge regression: Some simulations. Communications in Statistics-Theory and Methods, 4(2), 105-123.