STAT 652 Assignment 1

Dhruv Patel, 301471961

R-code with Answers:

```
## Lecture 4 – Application A ##
set.seed(301471961)

# A) #Loading and Filtering NA values from dataset
data = na.omit(airquality)
filter_data = (data[,1:4])
head(filter_data)

# Computing new columns TWcp and TWrat from Temp and Wind (Interactions)
filter_data$TWcp = filter_data$Temp*filter_data$Wind
filter_data$TWrat = filter_data$Temp/filter_data$Wind

# 1)Reporting Minimum, Maximum, Mean values
#Answer:
min(filter_data$TWcp) # 216.2
max(filter_data$TWcp) # 1490.4
mean(filter_data$TWcp) #756.527

min(filter_data$TWrat) #3.034826
max(filter_data$TWrat) #40.86957
mean(filter_data$TWrat) #9.419117

#2 # a) New model Creation and their Summary

#Temp + Wind + TWcp
lm_twcp = lm(Ozone ~ Temp + Wind + TWcp, data = filter_data)
summary(lm_twcp)
plot(lm_twcp)

#Temp + Wind + TWrat
lm_twrat = lm(Ozone ~ Temp + Wind + TWrat, data = filter_data)
summary(lm_twrat)

t.test(formula=Ozone ~ Temp + Wind + TWrat,filter_data) #==> t = 16.261
t.test(formula=Ozone ~ Temp + Wind + TWcp,filter_data)  #==> t = 16.261
t.test(formula=Ozone ~ Temp + Wind,filter_data)         #==> t = 16.261
confint(lm_twrat)
```

confint(lm_twcp)

#2 # b) Answer: After analyzing the above t.test values, it proves they are not particularly useful. Since, there is not much deviation then before.

#2 # c) Summary for model using Temp and its max and min temp

```
Call:
lm(formula = Ozone ~ Temp, data = AQ)

Residuals:
    Min      1Q  Median      3Q     Max 
-40.922 -17.459  -0.874  10.444 118.078 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -147.6461    18.7553  -7.872 2.76e-12 ***
Temp           2.4391     0.2393  10.192  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.92 on 109 degrees of freedom
Multiple R-squared:  0.488,	Adjusted R-squared:  0.4833 
F-statistic: 103.9 on 1 and 109 DF,  p-value: < 2.2e-16

min(filter_data$Wind)  #2.3
max(filter_data$Wind)  #20.7
```

#3) Model Fitting and computing MSPE for validation data.

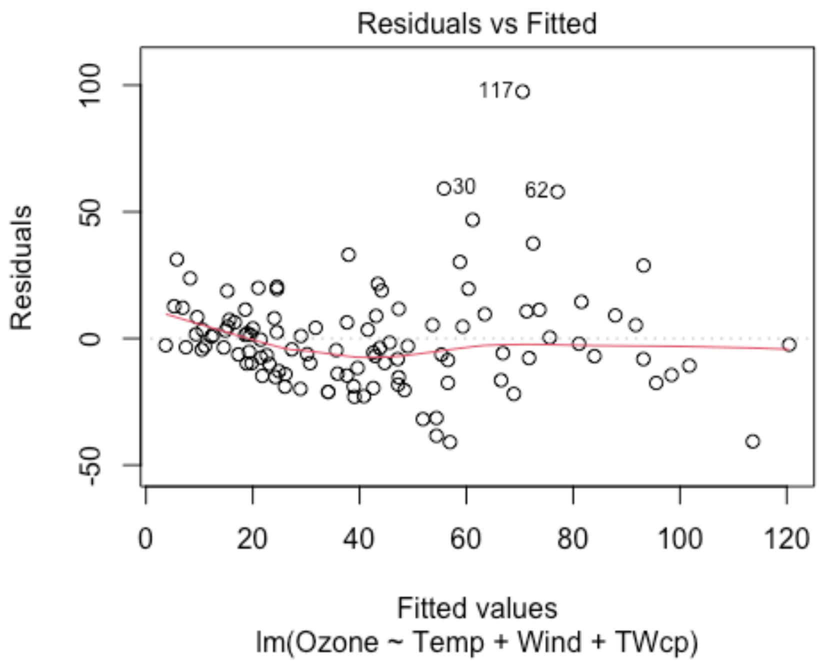
```
# Getting number of rows
rows = nrow(filter_data)

#Splitting data set train data and test data
train_split = 0.75
reorder_col = sample.int(n=rows, size=rows, replace=FALSE)
set = ifelse(test = ((train_split*rows) > reorder_col), yes=1, no=2)

train_data = filter_data[set==1,]
test_data = filter_data[set==2,]
```

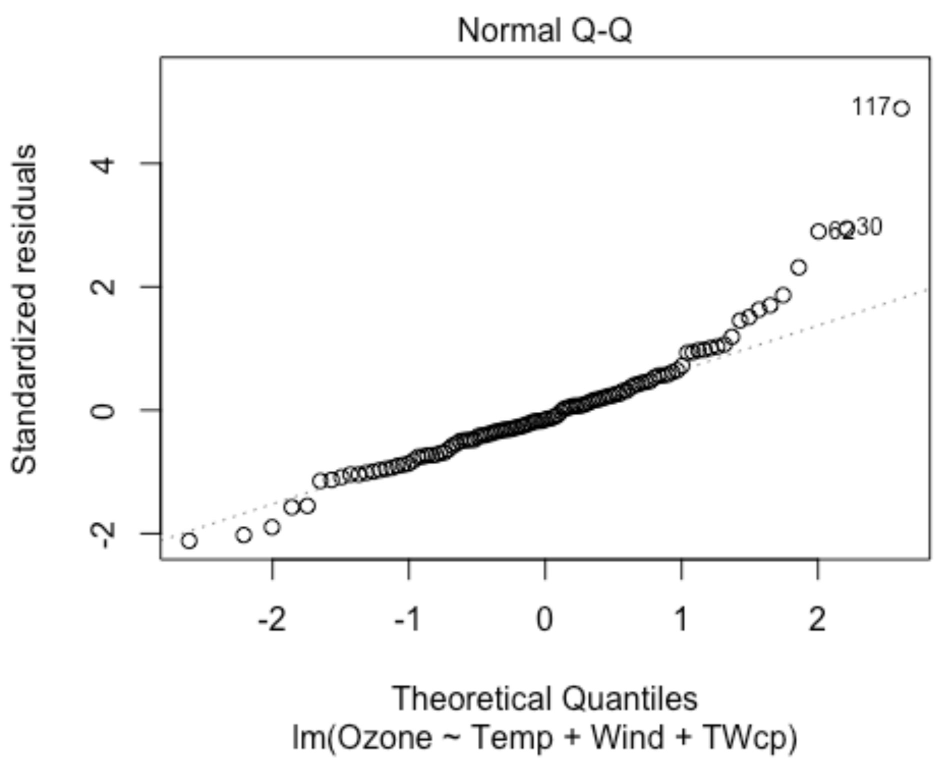
```
#Training model including TWcp
fit.TWcp = lm(Ozone ~ Temp + Wind + TWcp, data = train_data)

#Training model including TWrat
fit.TWrat = lm(Ozone ~ Temp + Wind + TWrat, data = train_data)

# Validating both models
pred.TWcp = predict(fit.TWcp, newdata=test_data)
pred.TWrat = predict(fit.TWrat,newdata=test_data)

#Calculating MSPE w.r.t to both TWcp and TWrat
MSPE.TWcp = mean((test_data$Ozone - pred.TWcp)^2)
MSPE.TWrat = mean((test_data$Ozone - pred.TWrat)^2)

MSPE.TWcp   #582.3652
MSPE.TWrat  #562.1254
# Answer:  From above comparison model with TWrat performs better then model TWcp
#4 ##### Make boxplots of the RMSPE, and narrow focus if necessary to see best models
better.
data$TWcp = data$Temp * data$Wind
data$TWrat = data$Temp / data$Wind

V=7 # No. of Models ["Solar.R", "Wind", "Temp","TWcp","TWrat","All","Custom"]
R=20 # Running CV 20 times

mat_CV = matrix(NA, nrow=V*R, ncol=7)
colnames(mat_CV) = c("Solar.R", "Wind", "Temp","TWcp","TWrat","All","Custom")

for (i in 1:R){

  folds = floor((sample.int(rows)-1)*V/rows) + 1

  for(j in 1:V){

    r = j+V*(i-1)
    # Training Model
    fit.Solar.R = lm(Ozone ~ Solar.R, data = data[folds!=j,])
    fit.Wind = lm(Ozone ~ Wind, data = data[folds!=j,])
    fit.Temp = lm(Ozone ~Temp, data = data[folds!=j,])
    fit.TWcp = lm(Ozone ~ Temp + Wind + TWcp, data = data[folds!=j,])
    fit.TWrat = lm(Ozone ~ Temp + Wind + TWrat, data = data[folds!=j,])
    fit.All = lm(Ozone ~ ., data = data[folds!=j,])
```

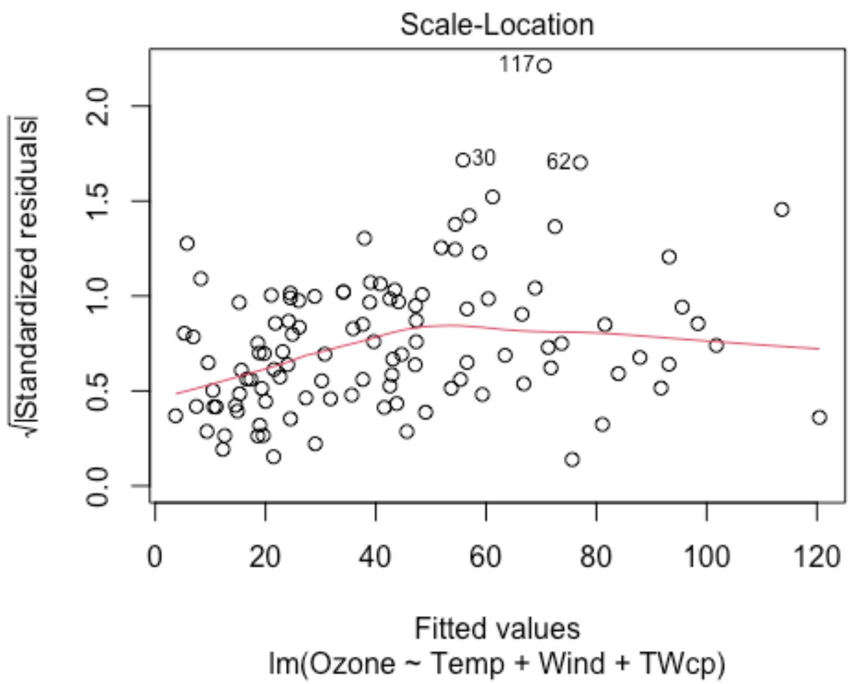
```r
    fit.Custom = lm(Ozone ~ (Temp+Wind+Solar.R):(Temp+Wind+Solar.R), data = data[folds!=j,])

    # Model Prediction
    pred.Solar.R = predict(fit.Solar.R, newdata = data[folds==j,])
    pred.Wind = predict(fit.Wind, newdata = data[folds==j,])
    pred.Temp = predict(fit.Temp, newdata = data[folds==j,])
    pred.TWcp = predict(fit.TWcp, newdata = data[folds==j,])
    pred.TWrat = predict(fit.TWrat,newdata = data[folds==j,])
    pred.All = predict(fit.All, newdata = data[folds==j,])
    pred.Custom = predict(fit.Custom,newdata = data[folds==j,])

    # Calculating MSPE for each attributes
    mat_CV[r,1] = mean((data[folds==j,"Ozone"] - pred.Solar.R)^2)
    mat_CV[r,2] = mean((data[folds==j,"Ozone"] - pred.Wind)^2)
    mat_CV[r,3] = mean((data[folds==j,"Ozone"] - pred.Temp)^2)
    mat_CV[r,4] = mean((data[folds==j,"Ozone"] - pred.TWcp)^2)
    mat_CV[r,5] = mean((data[folds==j,"Ozone"] - pred.TWrat)^2)
    mat_CV[r,6] = mean((data[folds==j,"Ozone"] - pred.All)^2)
    mat_CV[r,7] = mean((data[folds==j,"Ozone"] - pred.Custom)^2)
  }
}
# MSPE Boxplot
boxplot(mat_CV, las=2, ylim=c(0,1200),main="MSPE Cross-Validation")

# Relative MSPE Boxplot (Narrowed Focus)
rel_CV = mat_CV/apply(mat_CV, 1, min)
boxplot(rel_CV, las=2,ylim=c(0,3.5),main="Relative MSPE Cross-Validation")
```

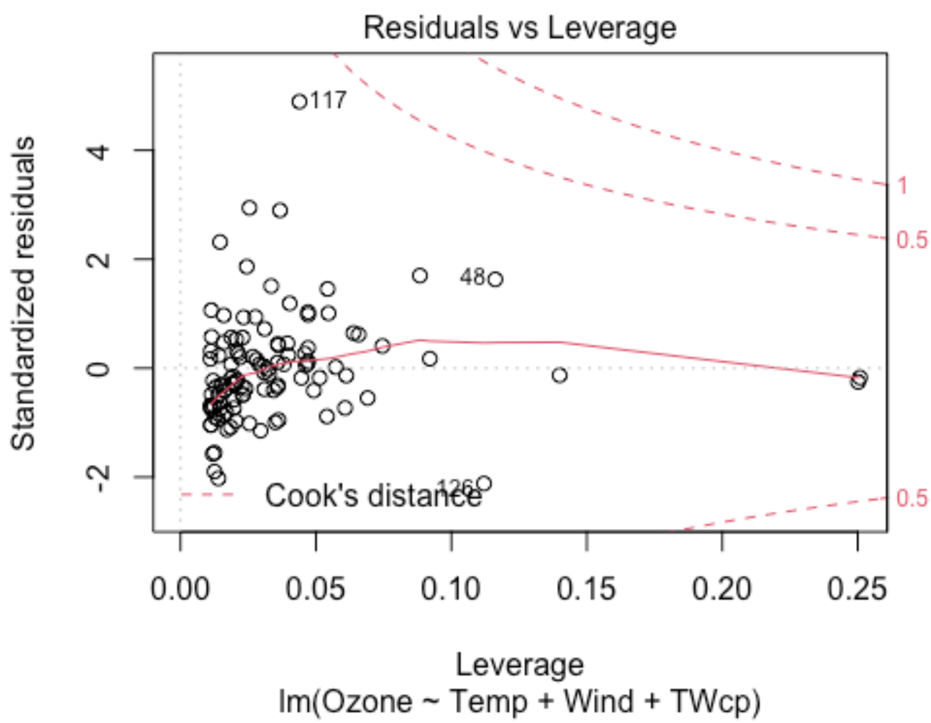
#b) Answer: Custom model with combination of Solar.R, Wind and Temp gives best performance.

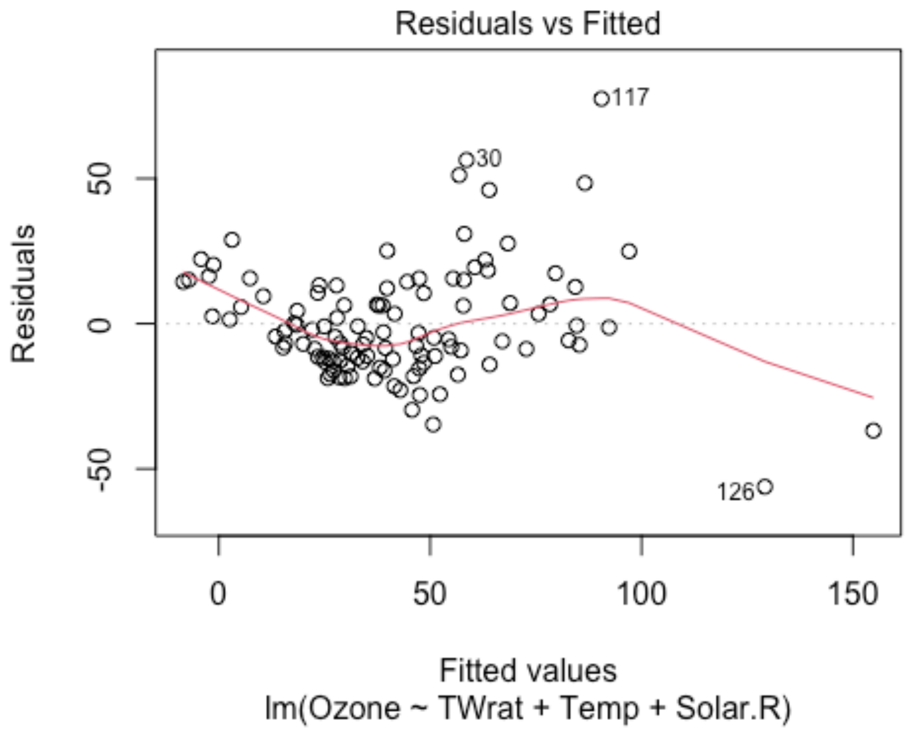
Labels Summary plots

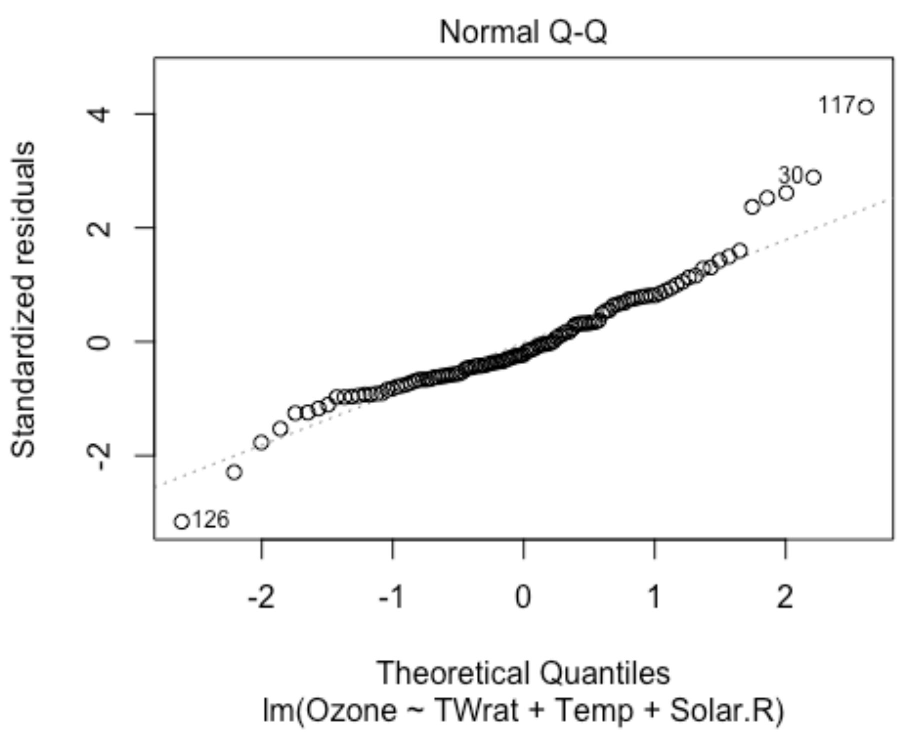


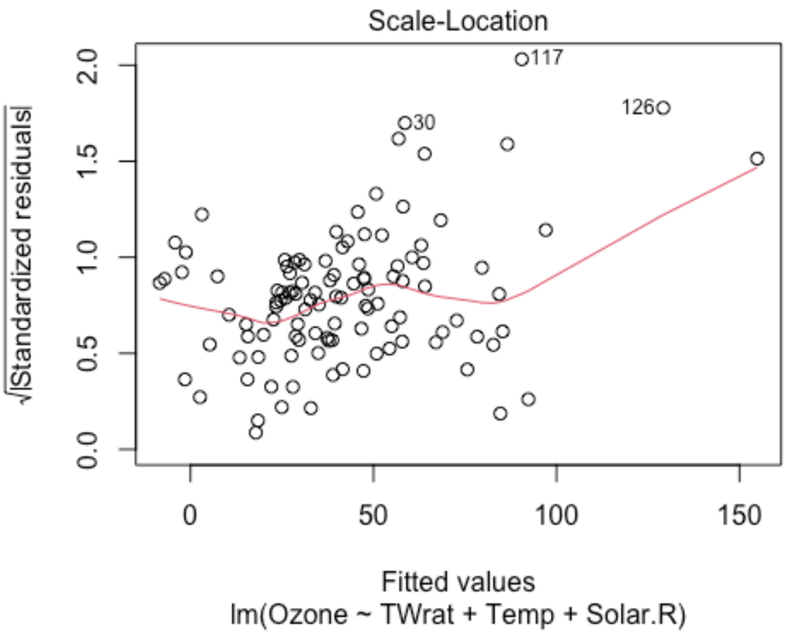


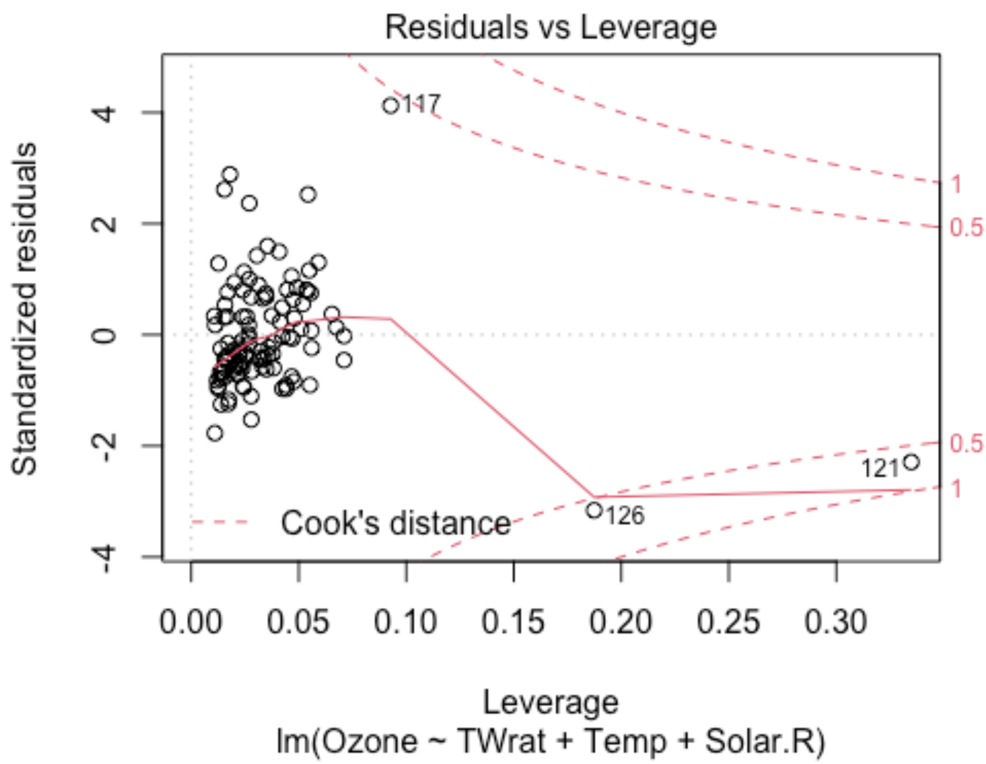

Scale-Location
117
30
62
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
20
40
60
80
100
120
Fitted values
lm(Ozone ~ Temp + Wind + TWcp)

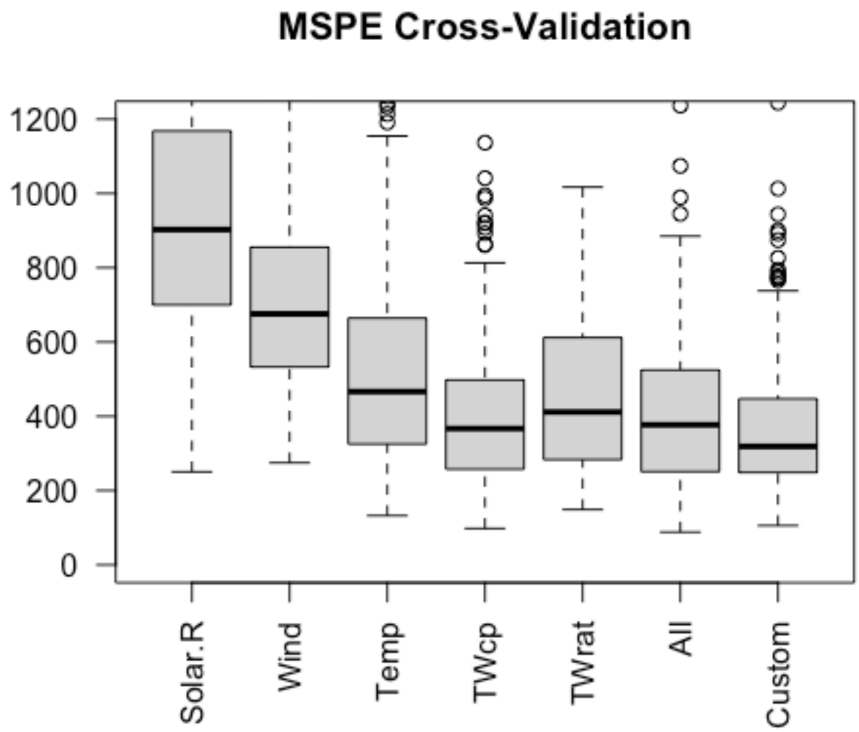

Residuals vs Leverage
117
48
126
Cook's distance
1
0.5
0.5
Standardized residuals
-2
0
2
4
0.00
0.05
0.10
0.15
0.20
0.25
Leverage
lm(Ozone ~ Temp + Wind + TWcp)

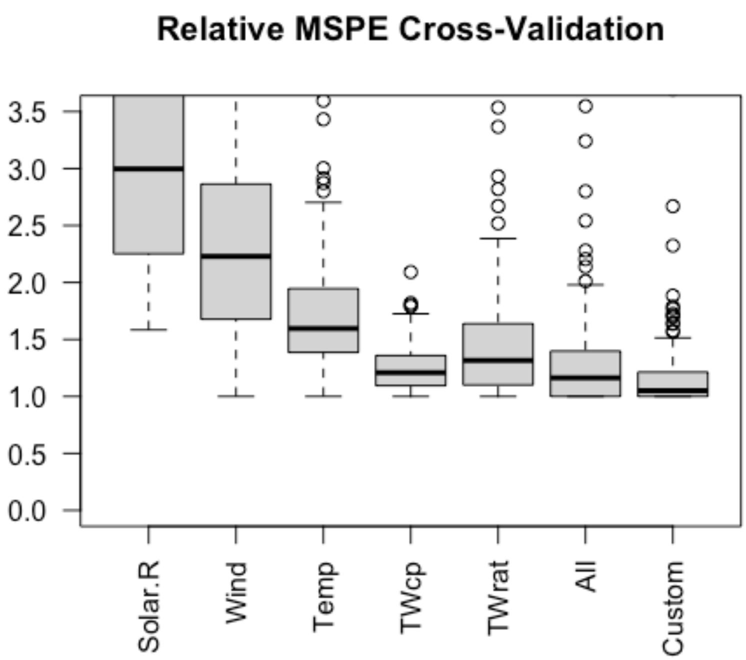

Residuals vs Fitted
Residuals
Fitted values
lm(Ozone ~ TWrat + Temp + Solar.R)
117
30
126
-50
0
50
0
50
100
150


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Ozone ~ TWrat + Temp + Solar.R)
117
30
126
-2
0
2
4
-2
-1
0
1
2


Scale-Location
√|Standardized residuals|
Fitted values
lm(Ozone ~ TWrat + Temp + Solar.R)
117
126
30


Residuals vs Leverage
Standardized residuals
Leverage
lm(Ozone ~ TWrat + Temp + Solar.R)
Cook's distance
117
126
121
1
0.5


MSPE Cross-Validation
0
200
400
600
800
1000
1200
Solar.R
Wind
Temp
TWcp
TWrat
All
Custom


Relative MSPE Cross-Validation
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Solar.R
Wind
Temp
TWcp
TWrat
All
Custom