HOMEWORK 2 (Intro to ML - Demo 2 / Data Pre-Processing & Regression)

Packages

For the Homework we will need the packages:`caret`, `skimr`,`mice`

```
library(caret)

## Lade nötiges Paket: ggplot2

## Lade nötiges Paket: lattice

library(skimr)
library(mice)

##
## Attache Paket: 'mice'

## Das folgende Objekt ist maskiert 'package:stats':
##
##     filter

## Die folgenden Objekte sind maskiert von 'package:base':
##
##     cbind, rbind
```

MICE (Multivariate Imputation via Chained Equations) creates multiple imputations as compared to a single imputation (such as mean) and takes care of uncertainty in missing values.

1. Airquality dataset

Variables Ozone: Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island Solar.R: Solar radiation in Langleys in the frequency band 4000–7700 Angstroms from 0800 to 1200 hours at Central Park Wind: Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport Temp: Maximum daily temperature in degrees Fahrenheit at La Guardia Airport. Month Day

Questions/Tasks to be done

1. Remove the outliers, delete all missing values & don't normalize the data. Keep the data splitting with 70% / 30%.

    - compare the results to the one obtained in today's class
    - what is you conclusion?

Descriptive statistics

You may look at

1. Dimensions of the dataset
2. Types of variables
3. Statistical summary of all attributes

```
# Structure of the dataframe
str(airquality)

## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

We have a dataset with 153 observations, where we see also some missing values, and 6 variables.

Which ML model do we use

We use Regression, because input and output variables are numeric.

The `skimr::skim()` shows us a Dataframe including descriptive stats of each of the columns.

```
skimmed <- skim(airquality)
skimmed
```

Data summary

Name	airquality
Number of rows	153
Number of columns	6
_______________________	
Column type frequency:	
numeric	6
________________________	
Group variables	None

Variable type: numeric

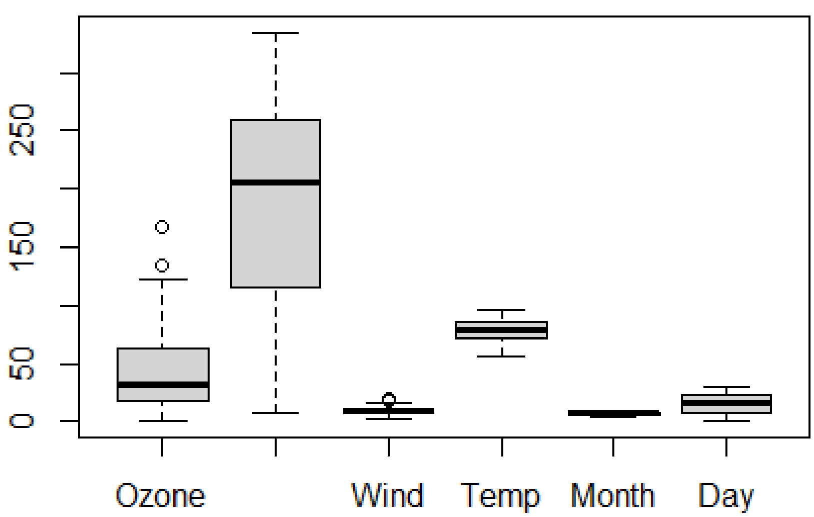
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Ozone	37	0.76	42.13	32.99	1.0	18.00	31.5	63.25	168.00	
Solar.R	7	0.95	185.	90.0	7.0	115.	205.	258.	334.	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
			93	6		75	0	75	0	
Wind	0	1.00	9.96	3.52	1.7	7.40	9.7	11.50	20.7	
Temp	0	1.00	77.88	9.47	56.0	72.00	79.0	85.00	97.0	
Month	0	1.00	6.99	1.42	5.0	6.00	7.0	8.00	9.0	
Day	0	1.00	15.80	8.86	1.0	8.00	16.0	23.00	31.0	

Here we see that there are 37 missing values for Ozone and 7 missing values for Solar.R and how the data is roughly distributed

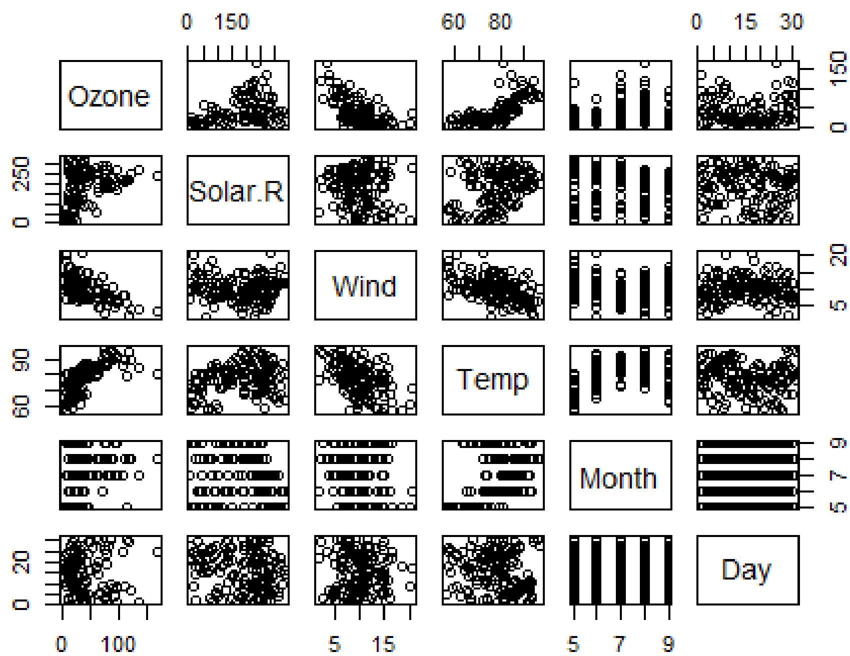
The boxplot function (see below) shows us also some outliers within the ozone and wind columns

```
boxplot(airquality)
```



Here we can see the interaction in the scatter plot between the different variables:

```
pairs(airquality)
```



especially for: - `Temp` with `Wind` -> neg. correlated - `Ozone` with `Wind` => non-linear -> neg. correlated - `Ozone` with `Temp` => non-linear -> pos. correlation - `Temp` with `Month`

Data pre-processing

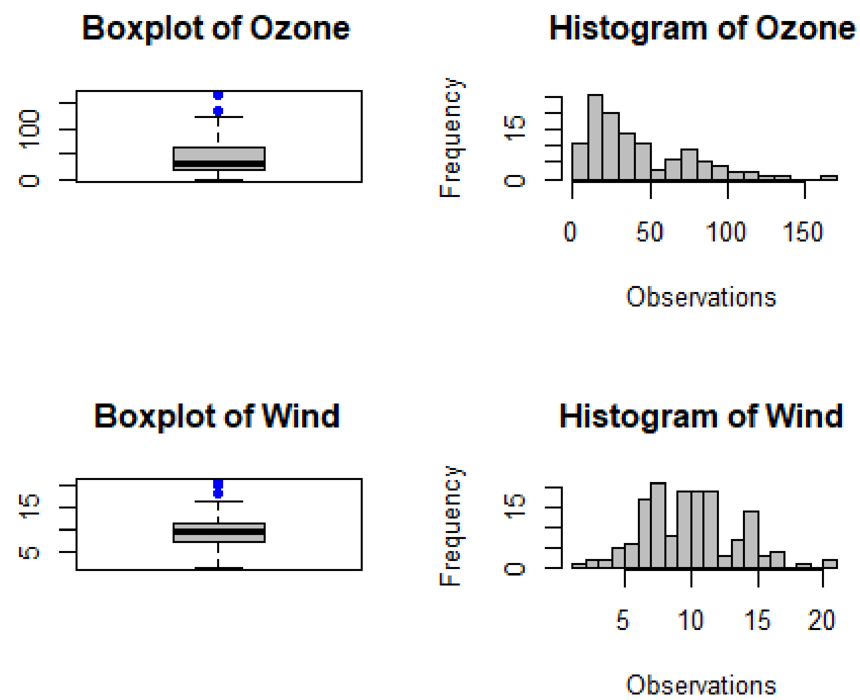
Now for this task 1 we need to:

1. Outlier detection -> and removing
2. Missing value treatment -> deleting all missing values
3. Normalization -> no normalization

Outlier detection

outliers lie outside 1.5 * IQR, which we can see in the boxplot.

```
par(mfrow=c(2,2))
boxplot(airquality$Ozone,col = "grey",main = "Boxplot of
Ozone",outcol="Blue",outpch=19,boxwex=0.7,range = 1.5)
hist(airquality$Ozone,col = "grey",main = "Histogram of Ozone", xlab =
"Observations",breaks = 15)
boxplot(airquality$Wind,col = "grey",main = "Boxplot of
Wind",outcol="Blue",outpch=19,boxwex=0.7,range = 1.5)
hist(airquality$Wind,col = "grey",main = "Histogram of Wind", xlab =
"Observations",breaks = 15)
```



Save the row numbers in a vector:

```
# get the values of the outliers
outliers_ozone <- boxplot(airquality$Ozone, plot = F)$out
# find the row numbers of the outliers
index_out <- match(outliers_ozone, airquality$Ozone)
```

`Wind` contains also outliers (see previous boxplots). Add the row number of these outliers to the vector `index_out`

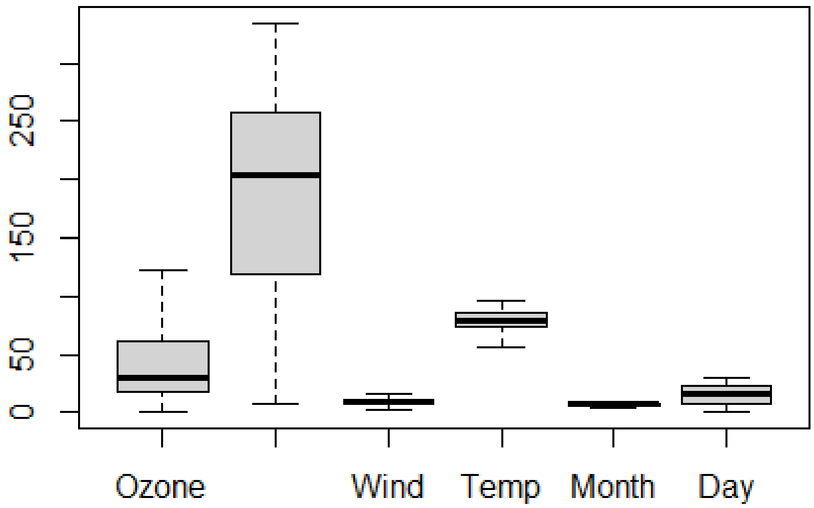
```
# get the values of the outliers
outliers_wind <- boxplot(airquality$Wind, plot = F)$out
# find the row numbers of the outliers & add them to the vector "index_out"
index_out <- c(index_out, match(outliers_wind, airquality$Wind))
index_out

## [1]  62 117   9  18  48
```

Now we remove all the outliers,because they have an effect on the mean and the general distribution:

```
# remove outliers
dataset <- airquality[-index_out,]
```

Check if outliers have been deleted

```
boxplot(dataset)
```



There are no outliers anymore in our dataset.

Missing values

Before we treat the missing data, it is good to check the amount of missing data.

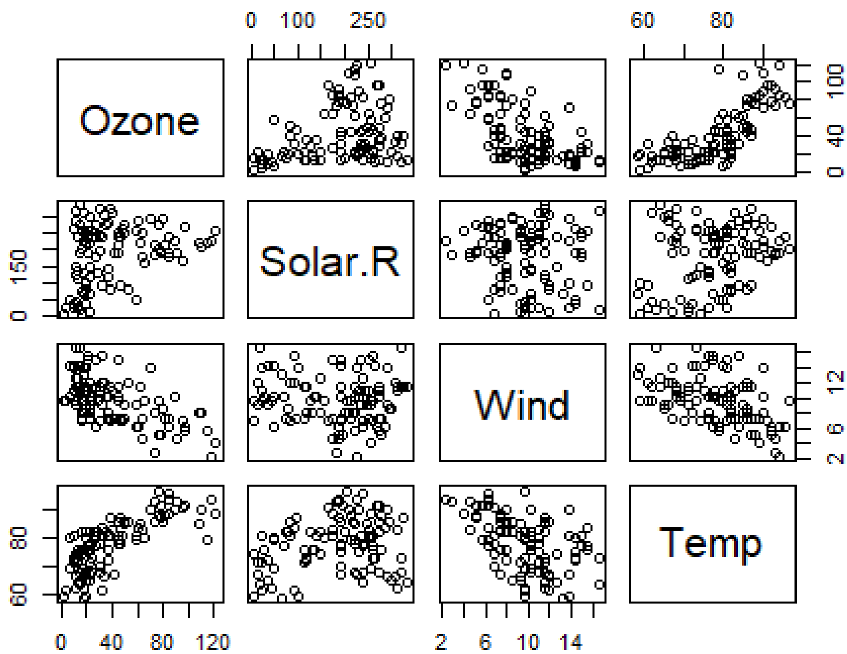
```
colSums(is.na(dataset))

##   Ozone Solar.R    Wind    Temp   Month     Day
##      37       7       0       0       0       0
```

Now we need to delete all the missing data.

```
alldata <- na.omit(dataset)
md.pattern(alldata, plot=F)

##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'

##     Ozone Solar.R Wind Temp Month Day
## 106     1       1    1    1     1   1 0
##         0       0    0    0     0   0 0
```

With the function above we deleted all missing values. And we see no missing values in the pattern function. Now we continue with the 106 remaining values.

```
pairs(alldata[,1:4])
```



Within the scatter-plot with the missing values deleted we see a less values but no bad spread like we had seen with the median imputation.

Because we don't need to normalize the data in this task we continue with developing a linear regression model.

4. Split your data: create a training and a test data set

```
# Create a list of 70% of the rows in the original dataset we can use for training
train_index<-createDataPartition(alldata$Ozone, p =0.70, list = FALSE)

# Select 30% of the data for testing
testing<-alldata[-train_index, ]

# Use the remaining 70% of data to train and validate the models
training <-alldata[train_index, ]
```

With the code above we again did a 70:30 split for training and testing the model.

5. Choose & evaluate ML models

```
# Linear Regression model
set.seed(7)
fit.lm <- train(Ozone~., data=training, method="lm", metric="Rsquared")
```

```
fit.lm

## Linear Regression
##
## 76 samples
##  5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 76, 76, 76, 76, 76, 76, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   17.58635  0.6662155  13.5508
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Here, with the same seed for reproduce ability, we do have already a higher Rsquared coefficient than with the median imputation, so deleting the missing values is a better method.

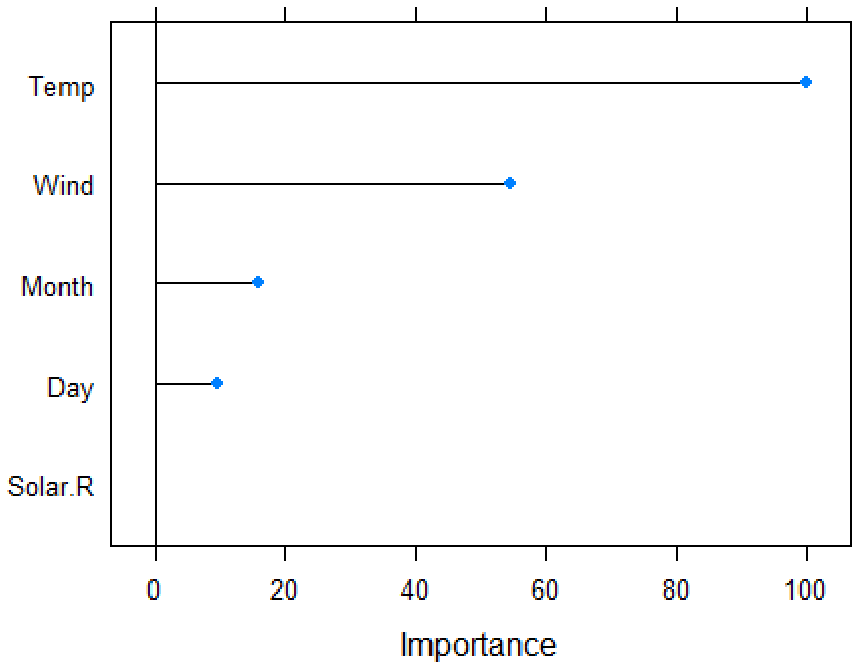
Have a closer look on the results:

```
summary (fit.lm)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.994  -9.616  -1.964   6.395  46.669
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -72.44242   20.76126  -3.489 0.000842 ***
## Solar.R       0.03149    0.02194   1.435 0.155638
## Wind         -3.26684    0.64509  -5.064 3.19e-06 ***
## Temp          1.98129    0.24569   8.064 1.39e-11 ***
## Month        -3.43359    1.37943  -2.489 0.015185 *
## Day           0.43395    0.20860   2.080 0.041162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.57 on 70 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.6944
## F-statistic: 35.09 on 5 and 70 DF,  p-value: < 2.2e-16
```

Only 2 Parameters are significant: `Wind` and `Temp`.

We can see the same order when using the `varImp()` function

```
plot(varImp(fit.lm))
```



Above we see the importance of the variables and reduce the model to the 3 parameters with the main influence:

```
fit.lm1 <- train(Ozone~Solar.R+Wind+Temp, data=training, method="lm",
metric="Rsquared")
summary(fit.lm1)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.971  -9.845  -2.898   7.600  60.688
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -71.08448   21.18383  -3.356  0.00127 **
## Solar.R       0.04908    0.02183   2.248  0.02767 *
## Wind         -3.17150    0.67908  -4.670 1.36e-05 ***
## Temp          1.67760    0.23360   7.182 5.10e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.45 on 72 degrees of freedom
```

```
## Multiple R-squared:  0.6726, Adjusted R-squared:  0.659
## F-statistic: 49.31 on 3 and 72 DF,  p-value: < 2.2e-16
```

We see while reducing the variables the sign. level stayed nearly the same.

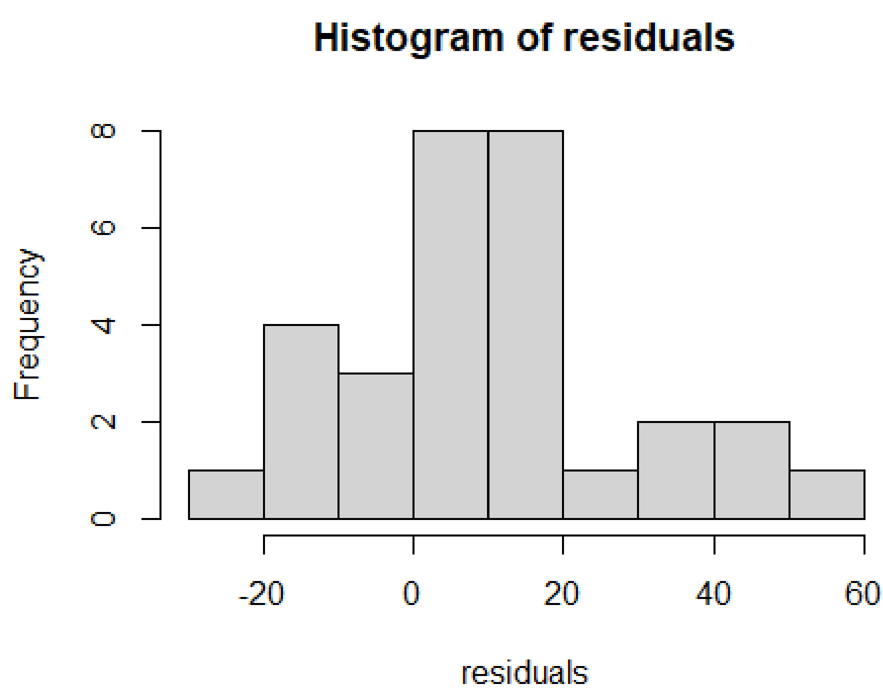
Check the prediction Quality

Make predictions on unseen data

```
predictions<-predict(fit.lm1, testing)
```

Check the distribution of the residuals

```
# density plot of residuals
residuals <- testing$Ozone - predictions

hist(residuals)
```



Coefficient of determination

```
SSE <- sum(residuals^2)
SST <- sum((testing$Ozone-mean(testing$Ozone))^2)
Rsq <-1-SSE/SST
Rsq

## [1] 0.6376463
```

After we make a prediction test with our 30% testing dataset we see that the R-squared coefficient with 52% of the variable variation around its mean is a bit higher than with the median imputed data with 42%.

Questions/Tasks to be done

2. Remove outliers, impute the missing values with predictive mean matching 5 times (m=5). Keep the data splitting with 70% / 30%.

   - compare the 5 imputing results with each other and select the best model (try to use functions from the `mice package`)
   - compare the best model with the one obtained in today's class
   - what is your conclusion?

Data pre-processing

Now for this task 1 we need to:

1. Outlier detection -> and removing
2. Missing value treatment -> PMM
3. Normalization

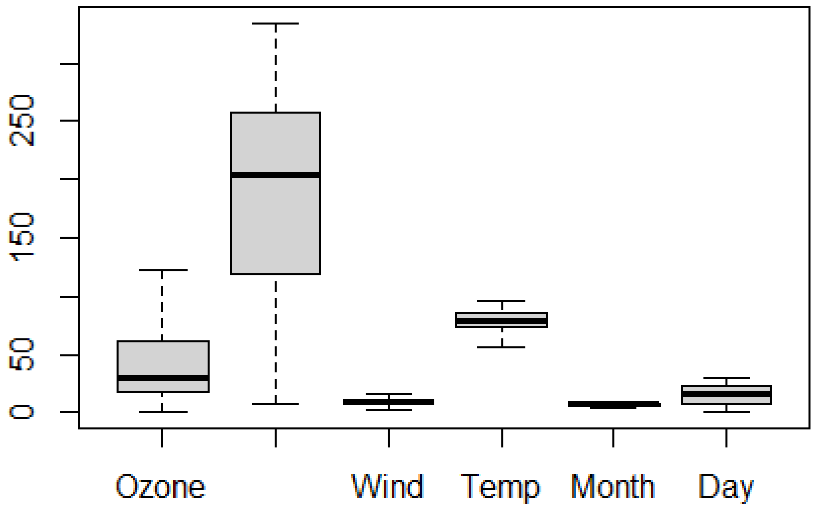
Outlier removing

Now we again remove all the outliers,because they have an effect on the mean and the general distribution:

```
# remove outliers
dataset <- airquality[-index_out,]
```

Check if outliers have been deleted

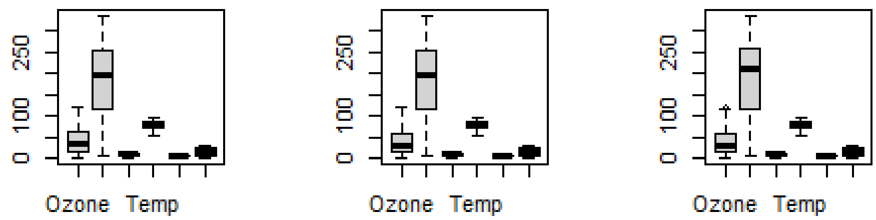
```
boxplot(dataset)
```

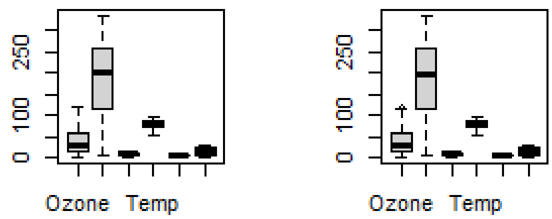


Missing values and PMM (predictive mean matching)

Now we are using PMM to treat our missing value problem

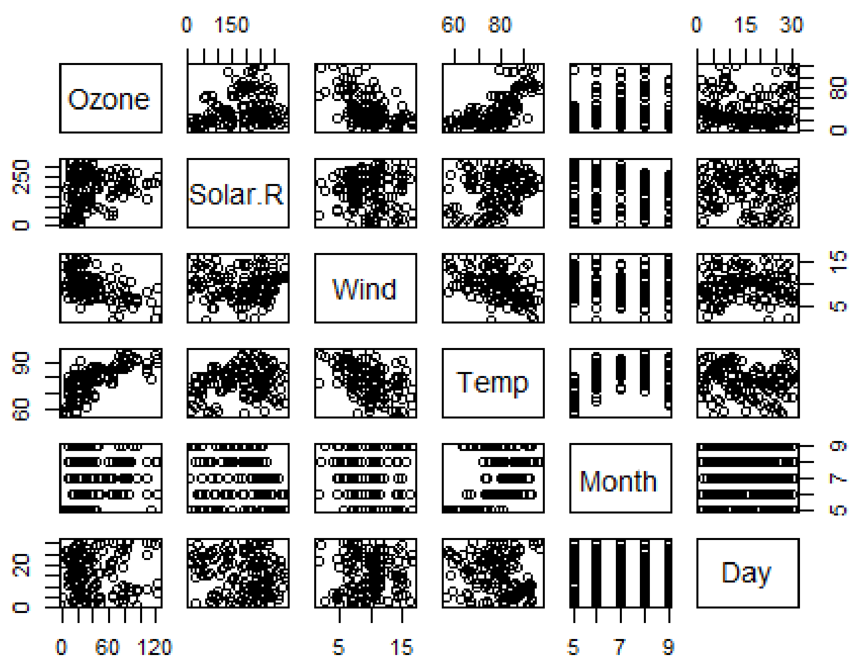
```
imputed_data_pmm <- mice(dataset,m=5,maxit=50,method='pmm', seed=
500,printFlag=F)

par(mfrow=c(2,3))
boxplot(complete(imputed_data_pmm,1))
boxplot(complete(imputed_data_pmm,2))
boxplot(complete(imputed_data_pmm,3))
boxplot(complete(imputed_data_pmm,4))
boxplot(complete(imputed_data_pmm,5))
```



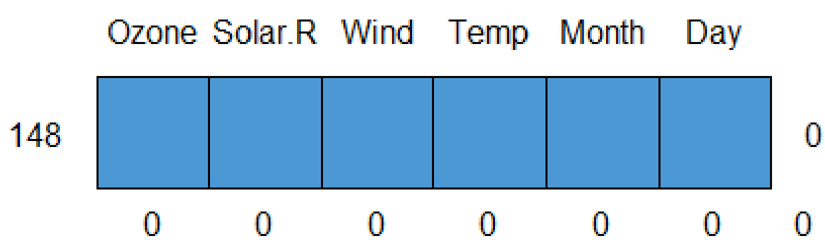


In PMM number 3 and 5 we have some outliers in the boxplot so we dont take these. Number 4 has a slightly lower median than our original boxplot. So i continue with PMM number 2 and also, the data are good presented in the hull in the pairs/scatter plot below.

```
pairs(complete(imputed_data_pmm,2))
```



```
md.pattern((complete(imputed_data_pmm,2)))

##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##     Ozone Solar.R Wind Temp Month Day
## 148     1       1    1    1     1   1 0
##          0       0    0    0     0   0 0
```

Now we have 148 values with which we will continue.

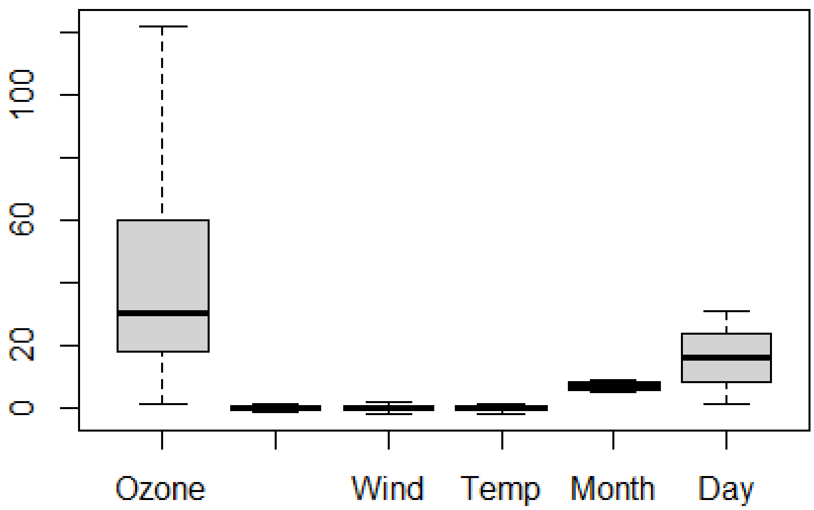
Normalization

We know that `log(Ozone)` may be a good response variable. But if we normalize `Ozone` we cannot do a `log()` transformation afterward.

Also, `Solar.R` is skewed, therefore standardization (z-transform) is not recommended. So we will use median and IQR normalization.

```
IQR <- apply(complete(imputed_data_pmm,2)[,2:4],2, IQR, na.rm=T)
med <- apply(complete(imputed_data_pmm,2)[,2:4],2, median, na.rm=T)

data_normalized <- data.frame("Ozone"=complete(imputed_data_pmm,2)$Ozone,
```

```
                              scale(complete(imputed_data_pmm,2)[,2:4],
center=med, scale = IQR), complete(imputed_data_pmm,2)[,5:6])

boxplot(data_normalized)
```



Split the data: create a training and a test data set

```
# Create a list of 70% of the rows in the original dataset we can use for
training
train_index<-createDataPartition(data_normalized$Ozone, p =0.70, list =
FALSE)

# Select 30% of the data for testing
testing<-data_normalized[-train_index, ]

# Use the remaining 70% of data to train and validate the models
training <-data_normalized[train_index, ]
```

With the code above we again did a 70:30 split for training and testing the model.

Choose & evaluate ML models

```
# Linear Regression model
set.seed(7)
fit.lm <- train(Ozone~., data=training, method="lm", metric="Rsquared")

fit.lm
```

```
## Linear Regression
##
## 104 samples
##   5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 104, 104, 104, 104, 104, 104, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   18.71576  0.5832572  15.19673
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

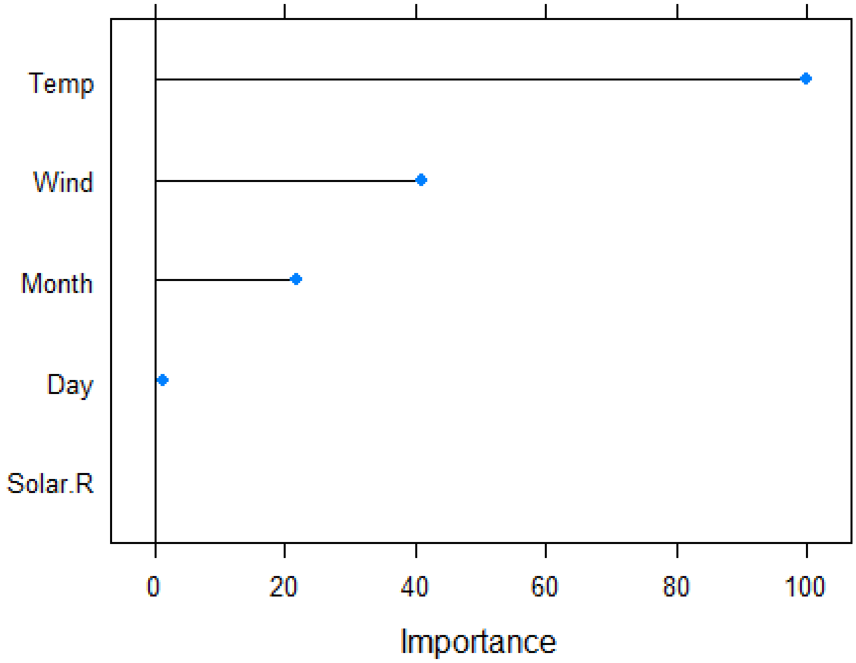
Here we see a not so optimal Rsquared coeff. of 0.58.

```
summary(fit.lm)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.839 -12.887  -0.855  11.416  49.087
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  61.6529    10.4031   5.926 4.62e-08 ***
## Solar.R       1.2065     2.8291   0.426  0.67071
## Wind         -9.4290     2.7094  -3.480  0.00075 ***
## Temp         24.2539     3.0753   7.887 4.42e-12 ***
## Month        -2.8765     1.3988  -2.056  0.04241 *
## Day           0.1019     0.2002   0.509  0.61178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.52 on 98 degrees of freedom
## Multiple R-squared:  0.6371, Adjusted R-squared:  0.6186
## F-statistic: 34.41 on 5 and 98 DF,  p-value: < 2.2e-16
```

Only wind and temp and month are sign. for the model, this is also seen in the plot below.

```
plot(varImp(fit.lm))
```



So we take these variables for furhter improvement.

```
fit.lm1 <- train(Ozone~Wind+Temp+Month, data=training, method="lm",
metric="Rsquared")

summary(fit.lm1)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.293 -12.613  -1.909  12.241  49.971
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   63.925      9.708   6.585 2.14e-09 ***
## Wind          -9.444      2.650  -3.563 0.000563 ***
## Temp          24.344      2.788   8.732 5.90e-14 ***
## Month         -2.988      1.323  -2.258 0.026119 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.39 on 100 degrees of freedom
## Multiple R-squared:  0.6355, Adjusted R-squared:  0.6246
## F-statistic: 58.12 on 3 and 100 DF,  p-value: < 2.2e-16
```

Check the prediction Quality

Make predictions on unseen data

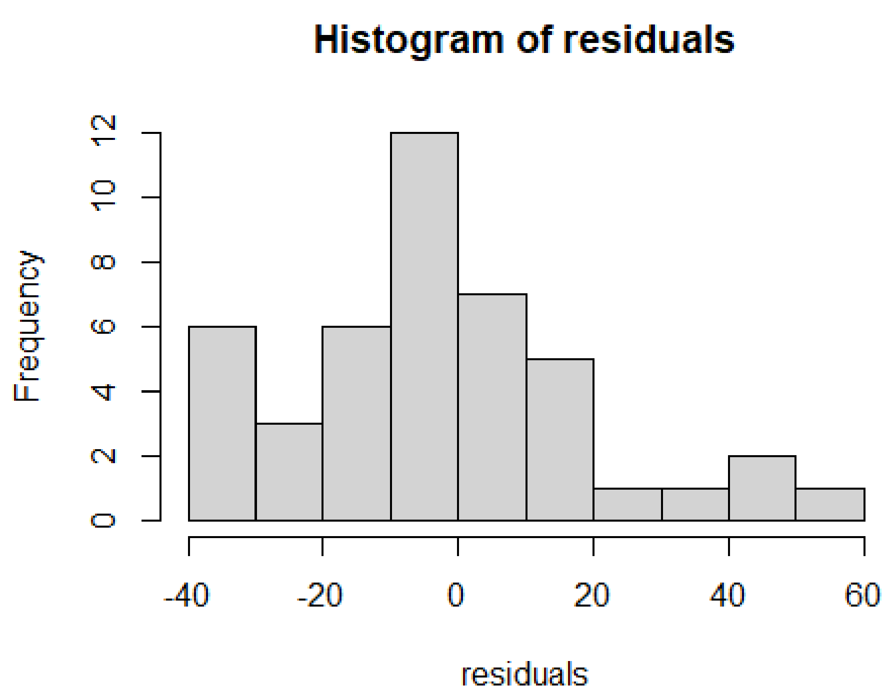
```
predictions<-predict(fit.lm1, testing)
```

Check the distribution of the residuals

```
# density plot of residuals
residuals <- testing$Ozone - predictions
```

Now we want to see how the model behaves with our testing data. And it shows a not so optimal normal distribution. More a skewed distribution.

```
hist(residuals)
```



Coefficient of determination

```
SSE <- sum(residuals^2)
SST <- sum((testing$Ozone-mean(testing$Ozone))^2)
Rsq <-1-SSE/SST
Rsq

## [1] 0.5414223
```

After we make a prediction test with our 30% testing dataset we see that the R-squared coefficient with 54% of the variable variation around its mean is a bit higher than with the median imputed data with 42%.

###————

Questions/Tasks to be done

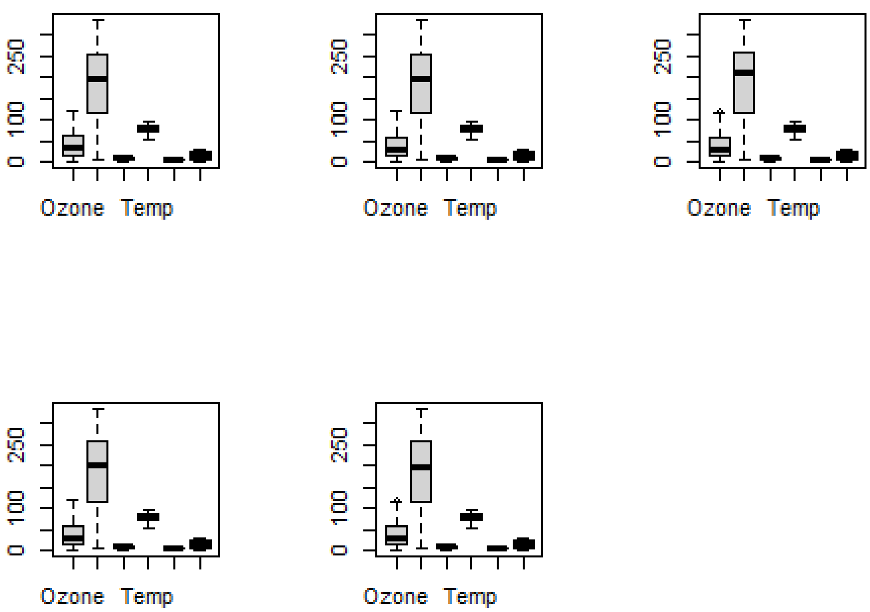
3. Select one of the imputed data sets from b. Keep the data splitting with 70% / 30%.

   - why did you select this data set?
   - play with the predictors and the response parameters and try to find a better model (use diverse transformation, multiplication of predictions, etc.)

Missing values and PMM (predictive mean matching)

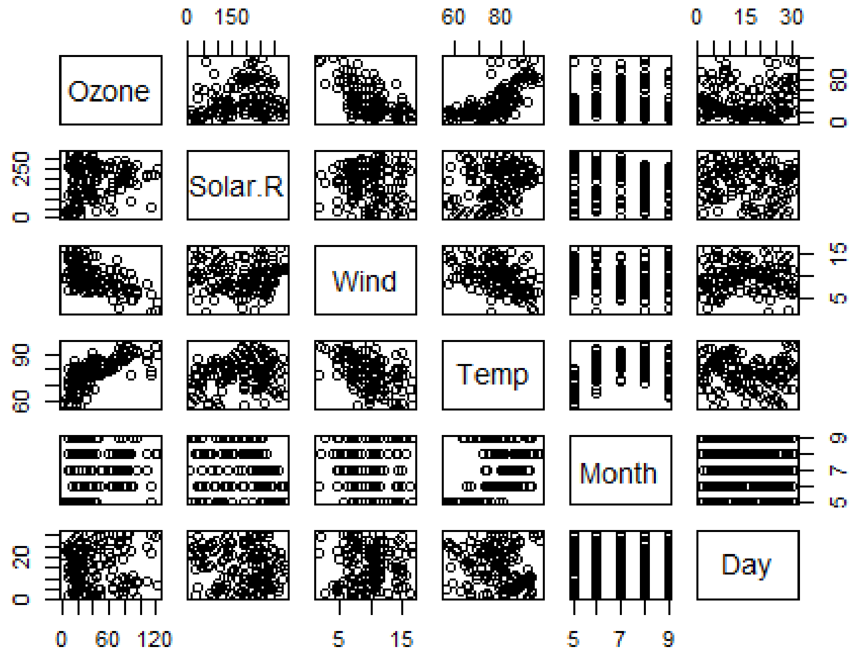
Now we are using PMM to treat our missing value problem

```
imputed_data_pmm <- mice(dataset,m=5,maxit=50,method='pmm', seed=
500,printFlag=F)

par(mfrow=c(2,3))
boxplot(complete(imputed_data_pmm,1))
boxplot(complete(imputed_data_pmm,2))
boxplot(complete(imputed_data_pmm,3))
boxplot(complete(imputed_data_pmm,4))
boxplot(complete(imputed_data_pmm,5))
```



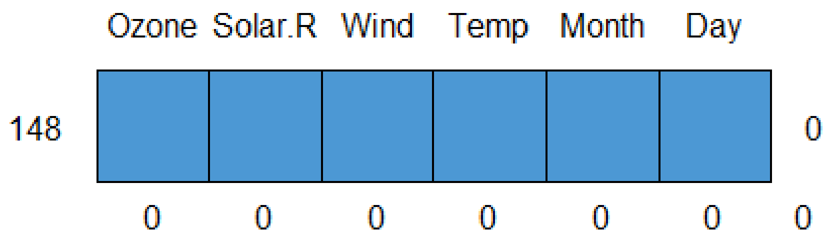
In PMM number 3 and 5 we have some outliers in the boxplot so we dont take these. Number 4 has a slightly lower median than our original boxplot. So this time i continue with PMM number 1 and also, the data are good presented in the hull in the pairs/scatter plot below.

```
pairs(complete(imputed_data_pmm,1))
```




```
md.pattern((complete(imputed_data_pmm,1)))
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##     Ozone Solar.R Wind Temp Month Day
## 148     1       1    1    1     1   1 0
##         0       0    0    0     0   0 0
```

Now we have 148 values with which we will continue.

Normalization

We now use median and IQR normalization. We also will try later not to normalize and see how this alters the model

```
IQR <- apply(complete(imputed_data_pmm,1)[,2:4],2, IQR, na.rm=T)
med <- apply(complete(imputed_data_pmm,1)[,2:4],2, median, na.rm=T)

data_normalized <- data.frame("Ozone"=complete(imputed_data_pmm,1)$Ozone,
                              scale(complete(imputed_data_pmm,1)[,2:4],
center=med, scale = IQR), complete(imputed_data_pmm,1)[,5:6])
```

Split the data: create a training and a test data set

```
# Create a list of 70% of the rows in the original dataset we can use for
training
train_index<-createDataPartition(data_normalized$Ozone, p =0.70, list =
FALSE)

# Select 30% of the data for testing
```

```r
testing<-data_normalized[-train_index, ]

# Use the remaining 70% of data to train and validate the models
training <-data_normalized[train_index, ]
```

With the code above we again did a 70:30 split for training and testing the model.

Choose & evaluate ML models

```r
# Linear Regression model
set.seed(7)
fit.lm <- train(Ozone~., data=training, method="lm", metric="Rsquared")

fit.lm

## Linear Regression
##
## 106 samples
##   5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 106, 106, 106, 106, 106, 106, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   18.90313  0.6103876  14.8832
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

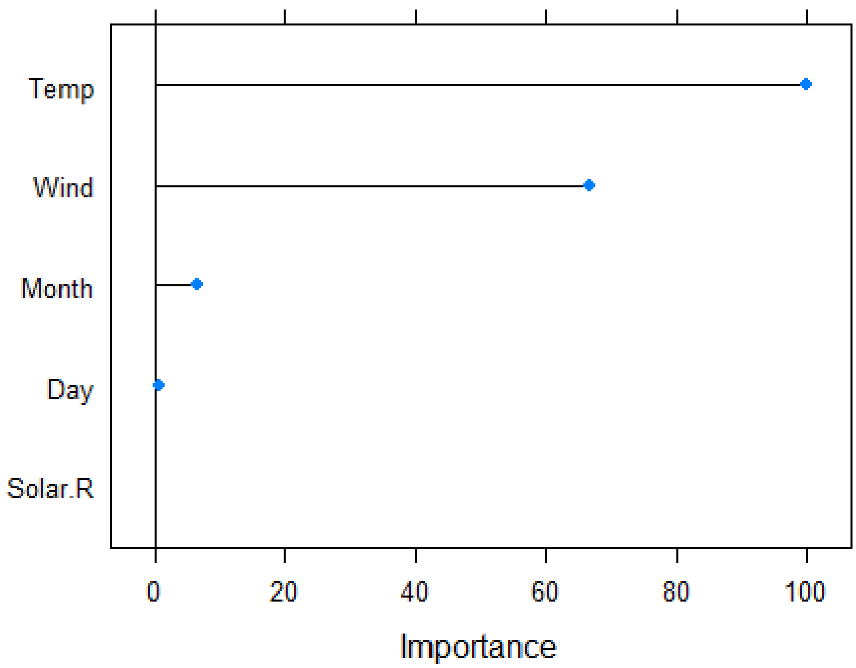
Here we see a not so optimal Rsquared coeff. of 0.61, slightly better than with the PMM number 2 imputation.

```r
summary(fit.lm)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.141 -10.782  -1.991  10.683  49.303
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  57.8173    10.1246   5.711 1.16e-07 ***
## Solar.R       4.1039     2.7965   1.468   0.1454
## Wind        -13.4436     2.3684  -5.676 1.35e-07 ***
## Temp         21.8252     2.8101   7.767 7.19e-12 ***
## Month        -2.5503     1.3633  -1.871   0.0643 .
## Day           0.2939     0.1947   1.510   0.1343
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.54 on 100 degrees of freedom
## Multiple R-squared:  0.6552, Adjusted R-squared:  0.638
## F-statistic: 38.01 on 5 and 100 DF,  p-value: < 2.2e-16
```

Only wind and temp are sign. for the model, this is also seen in the plot below.

```
plot(varImp(fit.lm))
```



So we take these variables for further improvement.

```
fit.lm1 <- train(Ozone~Wind+Temp, data=training, method="lm",
metric="Rsquared")

summary(fit.lm1)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.929 -10.008  -2.152  12.316  49.487
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   44.296      1.761  25.160  < 2e-16 ***
```

```
## Wind          -13.474      2.399  -5.616 1.67e-07 ***
## Temp           20.297      2.463   8.242 5.74e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.99 on 103 degrees of freedom
## Multiple R-squared:  0.6263, Adjusted R-squared:  0.619
## F-statistic: 86.31 on 2 and 103 DF,  p-value: < 2.2e-16
```

Check the prediction Quality

Make predictions on unseen data

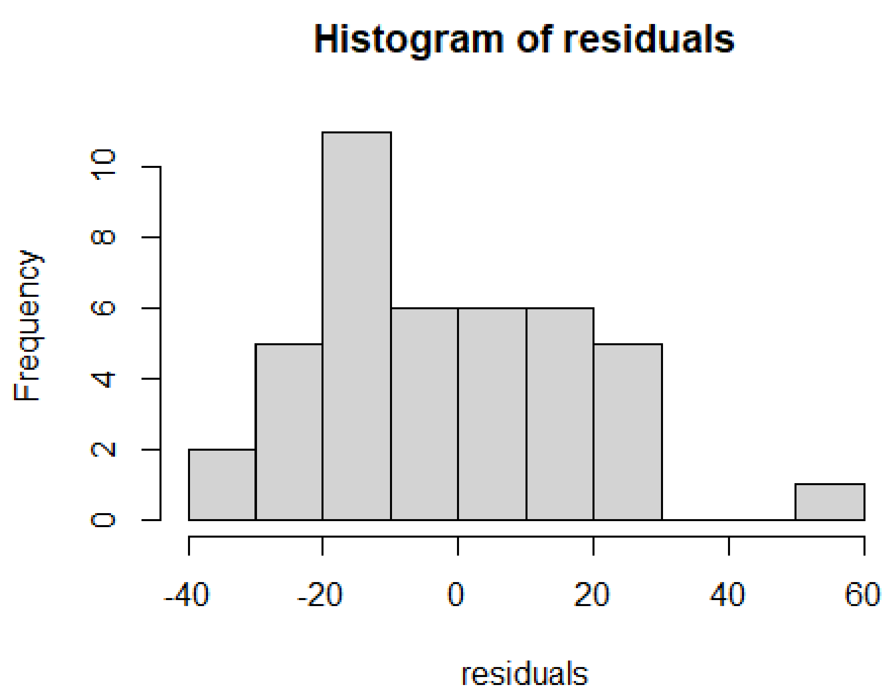
```
predictions<-predict(fit.lm1, testing)
```

Check the distribution of the residuals

```
# density plot of residuals
residuals <- testing$Ozone - predictions
```

Now we want to see how the model behaves with our testing data. And it shows no normal distribution. More a skewed distribution.

```
hist(residuals)
```



Coefficient of determination

```
SSE <- sum(residuals^2)
SST <- sum((testing$Ozone-mean(testing$Ozone))^2)
```

```
Rsq <-1-SSE/SST
Rsq

## [1] 0.5690257
```

After we make a prediction test with our 30% testing dataset we see that the R-squared coefficient with 56% of the variable variation around its mean is slightly higher than with the PMM 2 data with 54%.

New Approach without normalization.

Split the data: create a training and a test data set

```
# Create a list of 70% of the rows in the original dataset we can use for
training
train_index<-createDataPartition(complete(imputed_data_pmm,1)$Ozone, p =0.70,
list = FALSE)

# Select 30% of the data for testing
testing<-complete(imputed_data_pmm,1)[-train_index, ]

# Use the remaining 70% of data to train and validate the models
training <-complete(imputed_data_pmm,1)[train_index, ]
```

With the code above we again did a 70:30 split but with the not normalized dataset for training and testing the model.

Choose & evaluate ML models

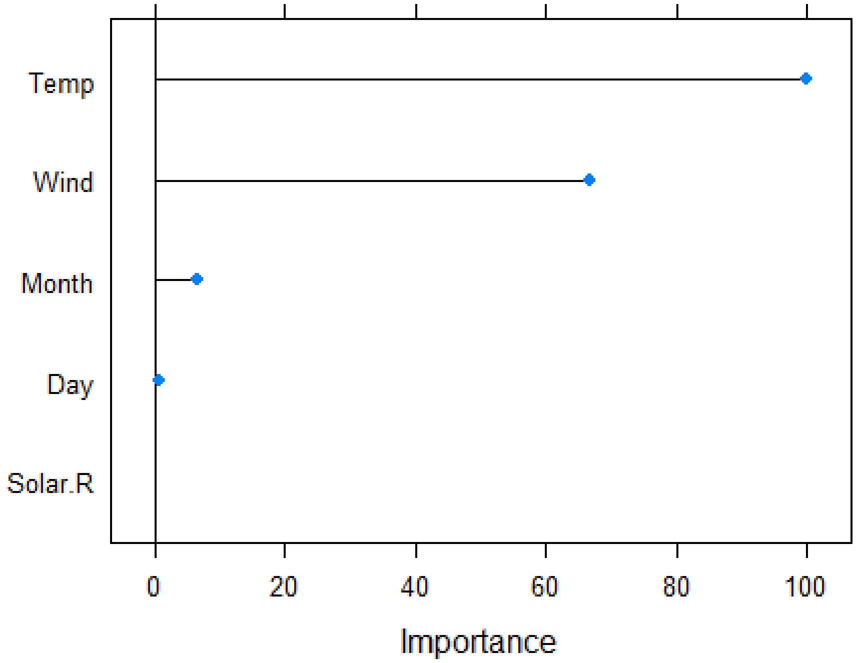
```
# Linear Regression model
set.seed(7)
fit.lm <- train(Ozone~., data=training, method="lm", metric="Rsquared")

fit.lm

## Linear Regression
##
## 106 samples
##   5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 106, 106, 106, 106, 106, 106, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   18.90313  0.6103876  14.8832
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Here we see a slightly better Rsquared coeff. of 0.64, than with the normalized data

```
summary(fit.lm)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.141 -10.782  -1.991  10.683  49.303
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -59.87319   19.65278  -3.047  0.00296 **
## Solar.R       0.02974    0.02026   1.468  0.14538
## Wind         -3.27893    0.57766  -5.676 1.35e-07 ***
## Temp          1.81876    0.23418   7.767 7.19e-12 ***
## Month        -2.55031    1.36330  -1.871  0.06431 .
## Day           0.29390    0.19468   1.510  0.13430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.54 on 100 degrees of freedom
## Multiple R-squared:  0.6552, Adjusted R-squared:  0.638
## F-statistic: 38.01 on 5 and 100 DF,  p-value: < 2.2e-16
```

Now wind, temp, month and day are sign. for the model, this is also seen in the plot below.

```
plot(varImp(fit.lm))
```



So we take these variables for further improvement.

```
fit.lm1 <- train(Ozone~Wind+Temp+Month+Day, data=training, method="lm",
metric="Rsquared")

summary(fit.lm1)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.983 -11.328  -2.316  11.623  46.522
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -61.3406    19.7391  -3.108  0.00245 **
## Wind         -3.1657     0.5757  -5.498 2.89e-07 ***
## Temp          1.9242     0.2242   8.584 1.16e-13 ***
## Month        -2.8520     1.3554  -2.104  0.03785 *
## Day           0.2761     0.1954   1.413  0.16081
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.64 on 101 degrees of freedom
## Multiple R-squared:  0.6478, Adjusted R-squared:  0.6338
## F-statistic: 46.44 on 4 and 101 DF,  p-value: < 2.2e-16
```

Check the prediction Quality

Make predictions on unseen data

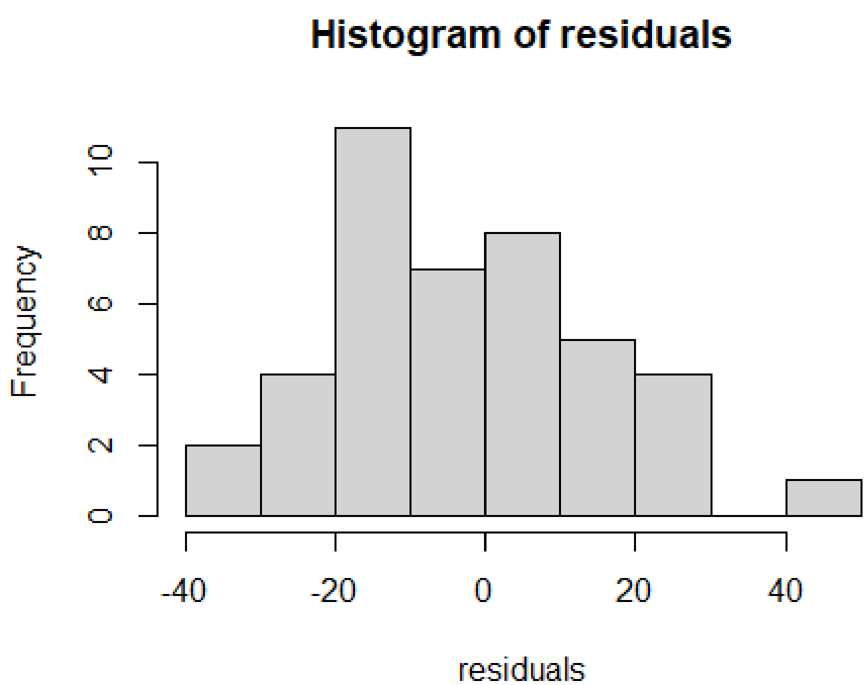
```
predictions<-predict(fit.lm1, testing)
```

Check the distribution of the residuals

```
# density plot of residuals
residuals <- testing$Ozone - predictions
```

Now we want to see how the model behaves with our testing data. And it shows no normal distribution. Again, more a skewed distribution.

```
hist(residuals)
```



Coefficient of determination

```
SSE <- sum(residuals^2)
SST <- sum((testing$Ozone-mean(testing$Ozone))^2)
Rsq <-1-SSE/SST
Rsq

## [1] 0.6391833
```

But we see a higher Rsquared coeff. with the non normalized data.

Exponential Transformation

Now we will try a exponential approach on our data

```
fit.lm2 <- train(Ozone~exp(Wind)+exp(Temp)+exp(Month)+exp(Day),
data=training, method="lm", metric="Rsquared")

summary(fit.lm2)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.878 -22.427  -6.122  15.901  79.456
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.505e+01  3.857e+00  11.678  < 2e-16 ***
## `exp(Wind)`  -3.094e-06  1.054e-06  -2.935  0.00413 **
## `exp(Temp)`   3.236e-41  1.975e-41   1.638  0.10449
## `exp(Month)` -8.447e-04  9.231e-04  -0.915  0.36233
## `exp(Day)`    5.820e-13  5.341e-13   1.090  0.27844
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.81 on 101 degrees of freedom
## Multiple R-squared:  0.1241, Adjusted R-squared:  0.08943
## F-statistic: 3.578 on 4 and 101 DF,  p-value: 0.009014
```

Here we see our model behaving much worse with the exp() transformation.

Check the prediction Quality

Make predictions on unseen data

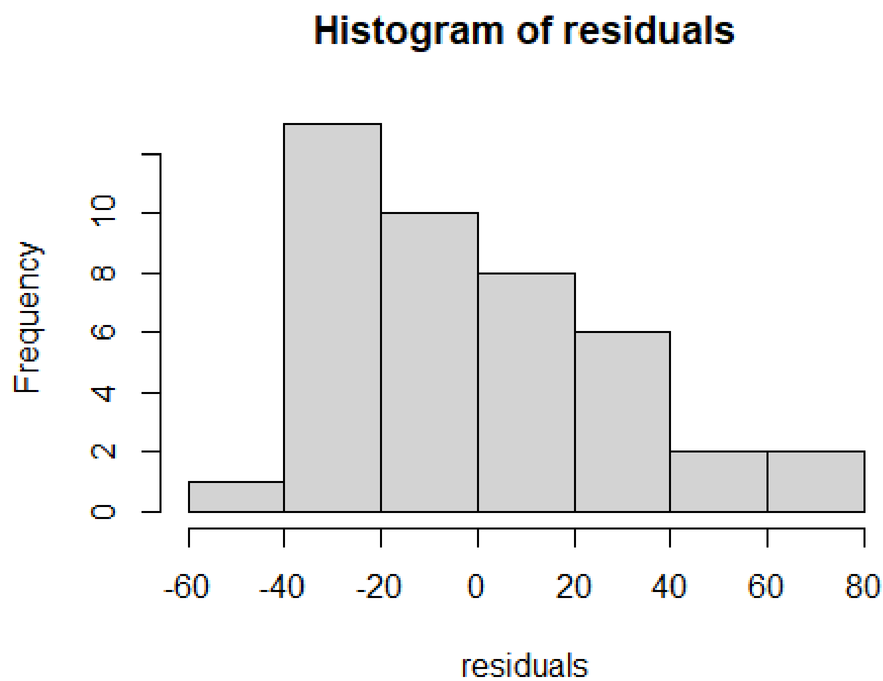
```
predictions<-predict(fit.lm2, testing)
```

Check the distribution of the residuals

```
# density plot of residuals
residuals <- testing$Ozone - predictions
```

Now we want to see how the model behaves with our testing data. And it shows no normal distribution. Again, more a skewed distribution.

```
hist(residuals)
```



Coefficient of determination

```
SSE <- sum(residuals^2)
SST <- sum((testing$Ozone-mean(testing$Ozone))^2)
Rsq <-1-SSE/SST
Rsq

## [1] 0.06794877
```

But we see a much fewer Rsquared coeff. with using the exp() transformation on the non normalized data.

End Summary:

Its obvious in my case that the best Rsquared coeff. of 0.60 can be achieved using the PMM number 1 instead of 2 and not normalized data. Especially the transformation exp() did harm my coeff.