Predictive Modeling

Chapter 6: Linear Regression and Its Cousins

STA 6543

The University of Texas at San Antonio

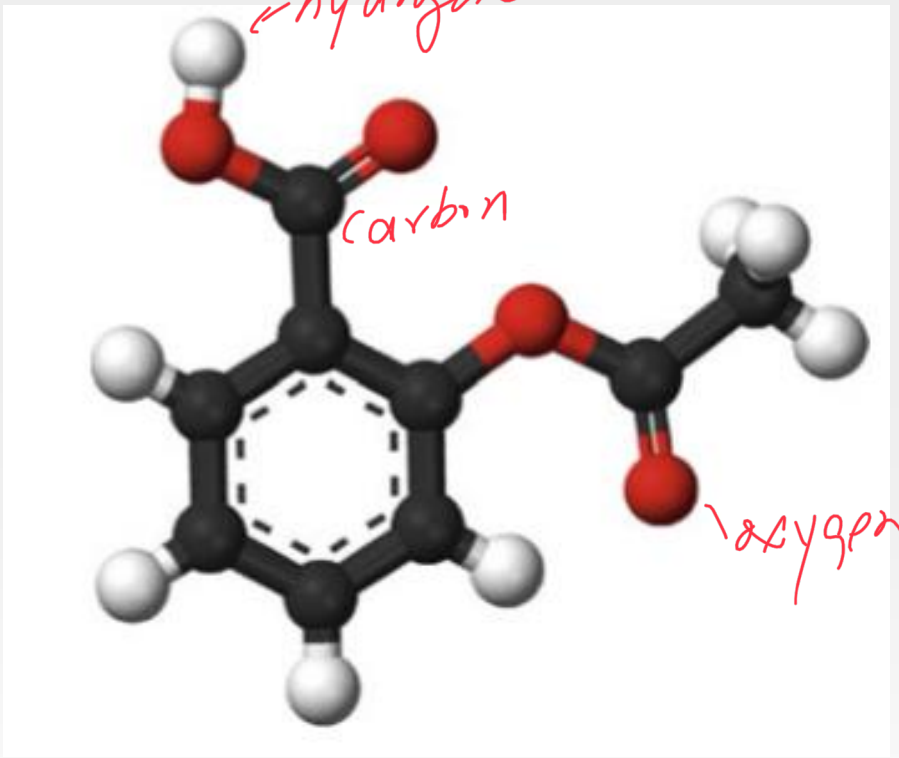
Overview

- Part I: General Strategies
- Part II: Regression Models
  - Chapter 6: Linear Regression and Its Cousins
  - Chapter 7: Nonlinear Regression Models
  - Chapter 8: Regression Trees and Rule-Based Models
- Part III: Classification Models
  - Chapter 12: Discriminant Analysis and Other Linear Classification Models
  - Chapter 13: Nonlinear Classification Models
  - Chapter 14: Classification Trees and Rule-Based Models

Linear Regression and Its Cousins

- A motivating example about solubility data
- Linear regression
- Principal component regression(PCR)
- Partial least squares (PLS)
- Penalized regression models
  - Ridge regression
  - Lasso
  - Elastic net (ENET)
- R demonstrations

A motivating example about solubility data



- A representation of aspirin, which contains carbon atoms (black balls) and hydrogen (white) and oxygen atoms (red)

A motivating example about solubility data

- We collect several quantitative measurements, called as *chemical descriptors.*
- We may obtain many characteristics of molecules empirically using experiments.
- The data set consists of 1267 compounds and a set of more understandable descriptors that fall into one of three groups:
  - Two hundred and eight binary (208) "fingerprints" that indicate the presence or absence of a particular chemical substructure. qualitative
  - Sixteen count descriptors (16), such as the number of bonds or the number of bromine atoms. quantitative (discrete)
  - Four continuous descriptors (4), such as molecular weight or surface area.

Continous

A motivating example about solubility data

- Researchers are interested in investigating a set of compounds with corresponding experimental solubility values using complex sets of descriptors.
- They used *linear regression (parametric model)* and *neural network (nonparametric model) models* to estimate the relationship between chemical structures (predictors) and solubility (response).

A motivating example about solubility data

Here are a few important questions that we might seek to address:

1. Do we need to pre-process the data? (i.e., skewness, near-zero variance predictors, significant between-predictor correlations, etc)
2. How do we split the data into a training and a test set. (random sampling, stratified random sampling)?
3. How do we fit some *parametric* models (i.e., linear regression models, partial least squares, penalized models).
4. Which model has the best predictive ability? Is any model significantly better or worse than the others?

Data Pre-processing

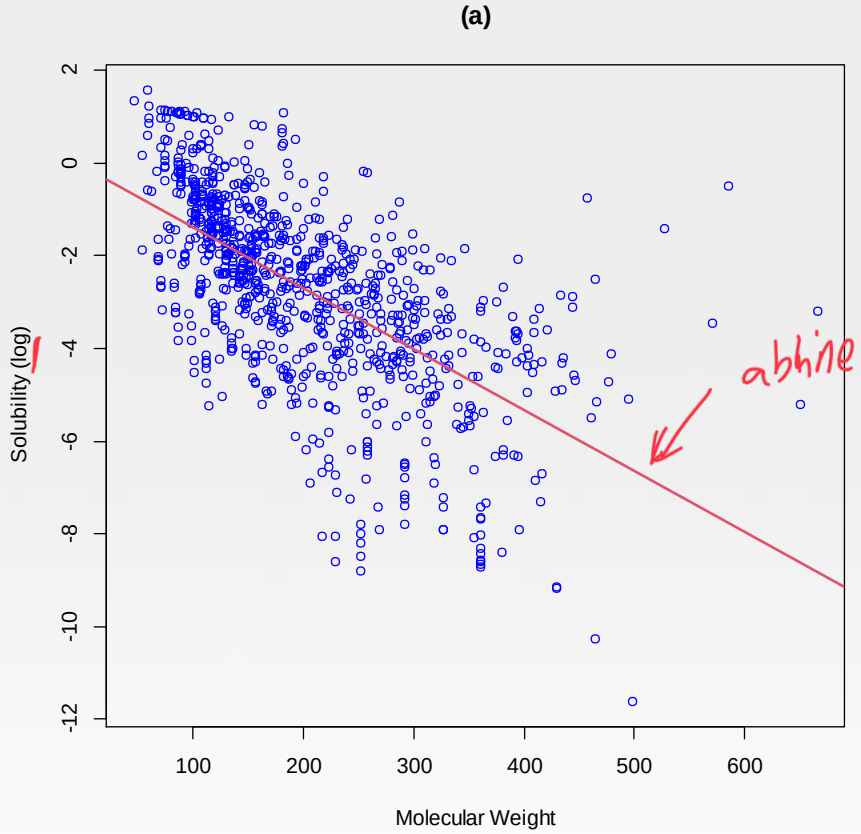
Data pre-processing

```
#Required packages
library(AppliedPredictiveModeling)
library(lattice)
library(caret)
library(corrplot)
library(e1071)
library(pls)
library(elasticnet)

#Access the solubility data
data(solubility)

plot(solTrainY ~ solTrainX$MolWeight, ylab = "Solubility (log)",
     main = "(a)", col='blue', xlab = "Molecular Weight")
fit = lm(solTrainY ~ solTrainX$MolWeight)
abline(fit, col=2, lwd=2)

#correlation test
cor.test(solTrainY,solTrainX$MolWeight)
```

scatter plot Y and X

Note solTrainY has been transformed by take log.

Correlations between descriptors



Negative correlation
linear
Is this relationship statistically significant ?

- Correlations between Molecular Weight and Solubility(log)

Correlation test between descriptors

```
> cor.test(solTrainY, solTrainX$MolWeight)

        Pearson's product-moment correlation

data:  solTrainY and solTrainX$MolWeight
t = -24.936, df = 949, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6660933 -0.5891573
sample estimates:
      cor
-0.6291639

>
```

$< \alpha = 0.05 \rightarrow$ Reject H_0.

which does not include 0.

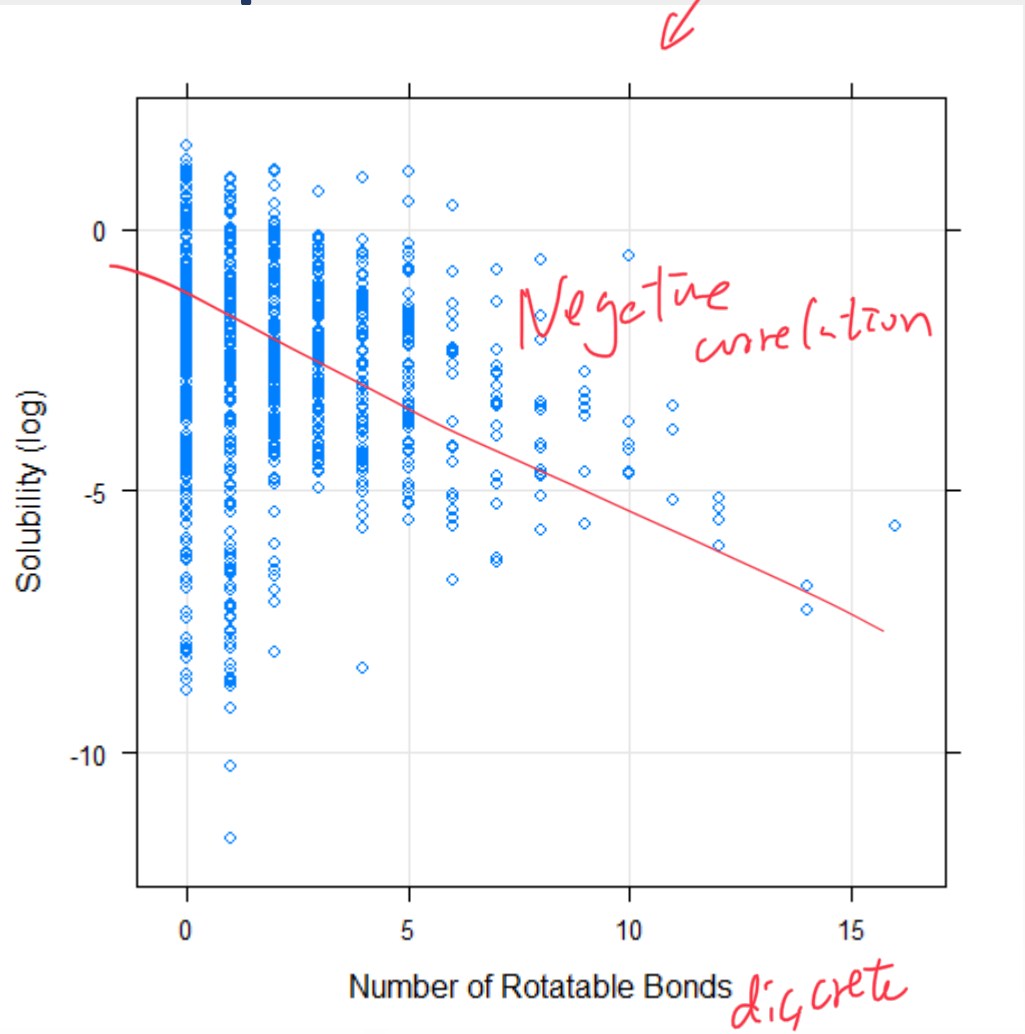
$H_0: \rho = 0$ vs $H_1: \rho \neq 0$

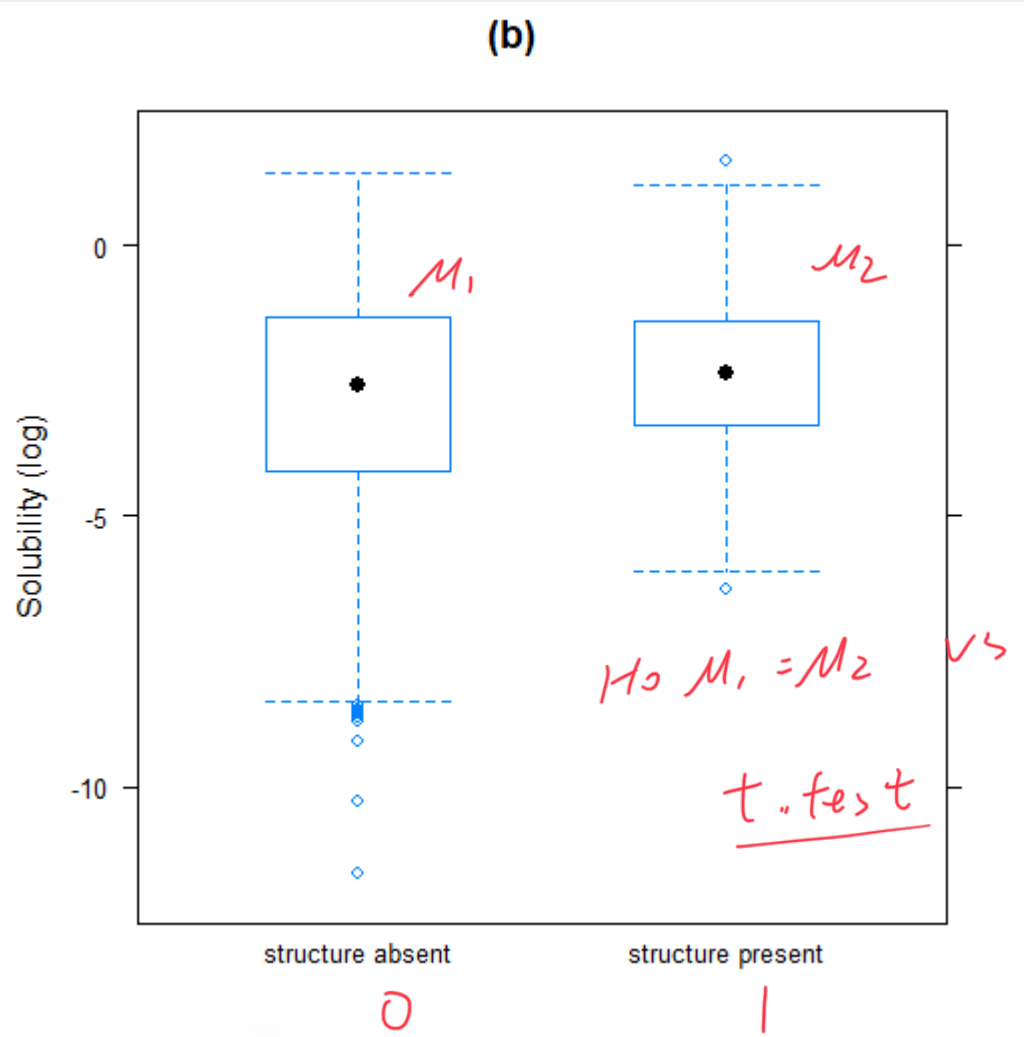
where ρ is the population linear correlation between Molecular weight and Solubility (log)

Scatter plot

```
xyplot(solTrainY ~ solTrainX$NumRotBonds, type = c("p", "g"),
       ylab = "Solubility (log)",
       xlab = "Number of Rotatable Bonds")

#The function bwplot() makes box-and-whisker plots for numerical variables
bwplot(solTrainY ~ ifelse(solTrainX[,100] == 1,
                          "structure present",
                          "structure absent"),
       ylab = "Solubility (log)",
       main = "(b)",
       horizontal = FALSE)
```

Scatter plot





Correlations between descriptors

The above examples showed that there are strong correlations among predictors, so how do we deal with significant correlations between the continuous predictors?

- Key step: Find the columns that are not fingerprints (FP) (i.e. numerical predictors). The R function *grep* will return a list of integers corresponding to column names that contain the pattern "FP".
- Idea: Use the R function *featurePlot* to produce lattice graphs or use the R function *corrplot* to get a graphical display of a correlation matrix

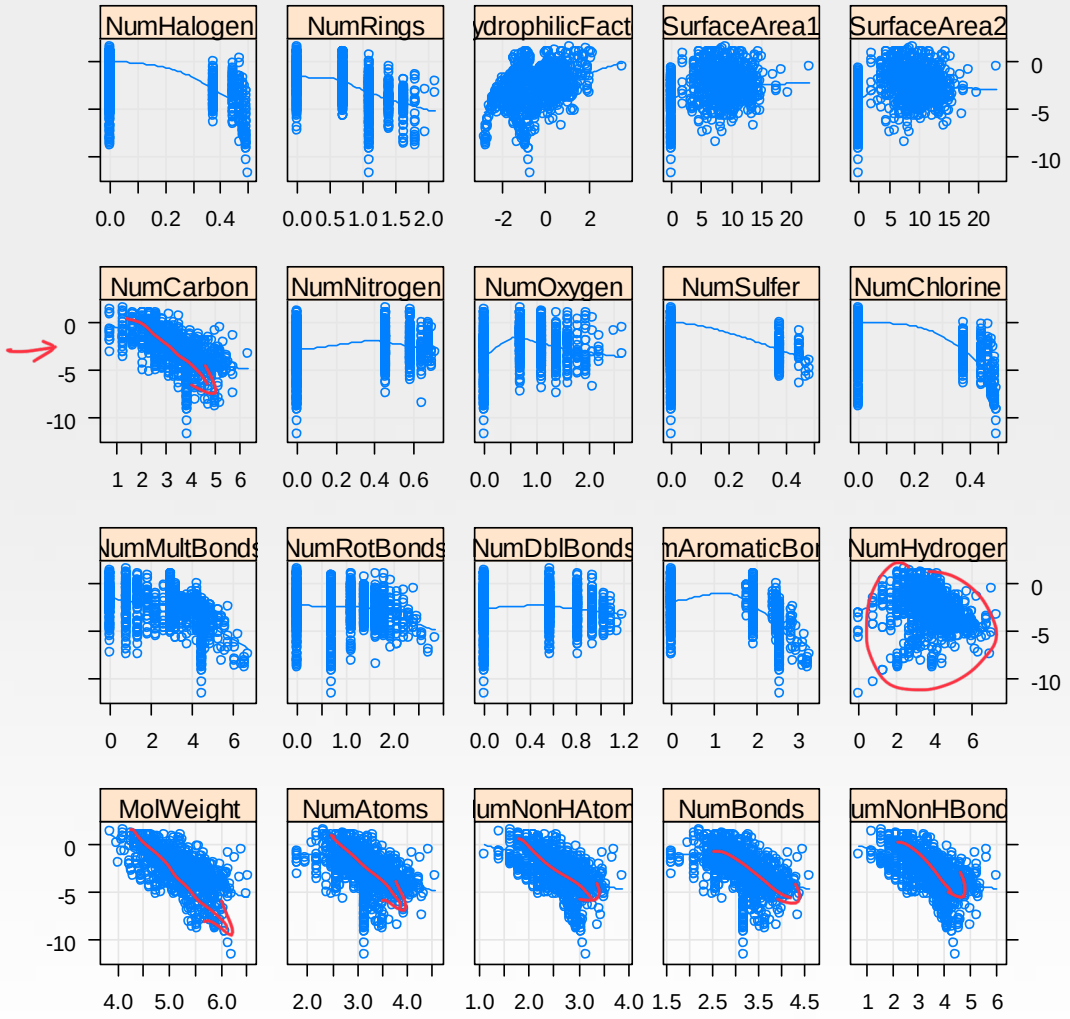
Correlations between descriptors

```
### Find the columns that are not fingerprints (i.e. numerical
### predictors). grep will return a list of integers corresponding to
### column names that contain the pattern "FP".

notFingerprints <- grep("FP", names(solTrainXtrans))

library(caret)
featurePlot(solTrainXtrans[, -notFingerprints],
            solTrainY,
            between = list(x = 1, y = 1),
            type = c("g", "p", "smooth"),
            labels = rep("", 2))

library(corrplot)
corrplot::corrplot(cor(solTrainXtrans[, -notFingerprints]),
                   order = "hclust",  tl.cex = .8)
```

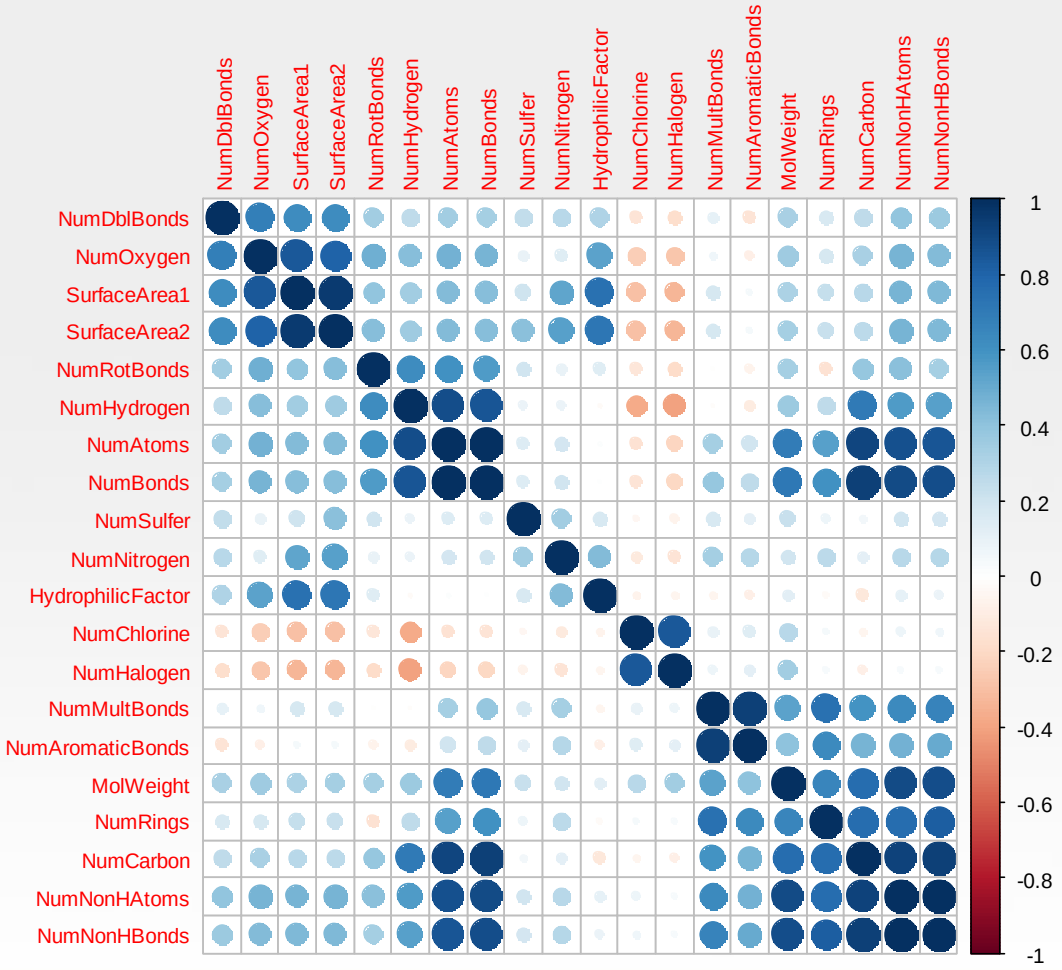
index of the column's name of solTrainXtrans that contain 'FP'

remove these binary random variables.

Correlations between descriptors and Y.



Correlations between descriptors



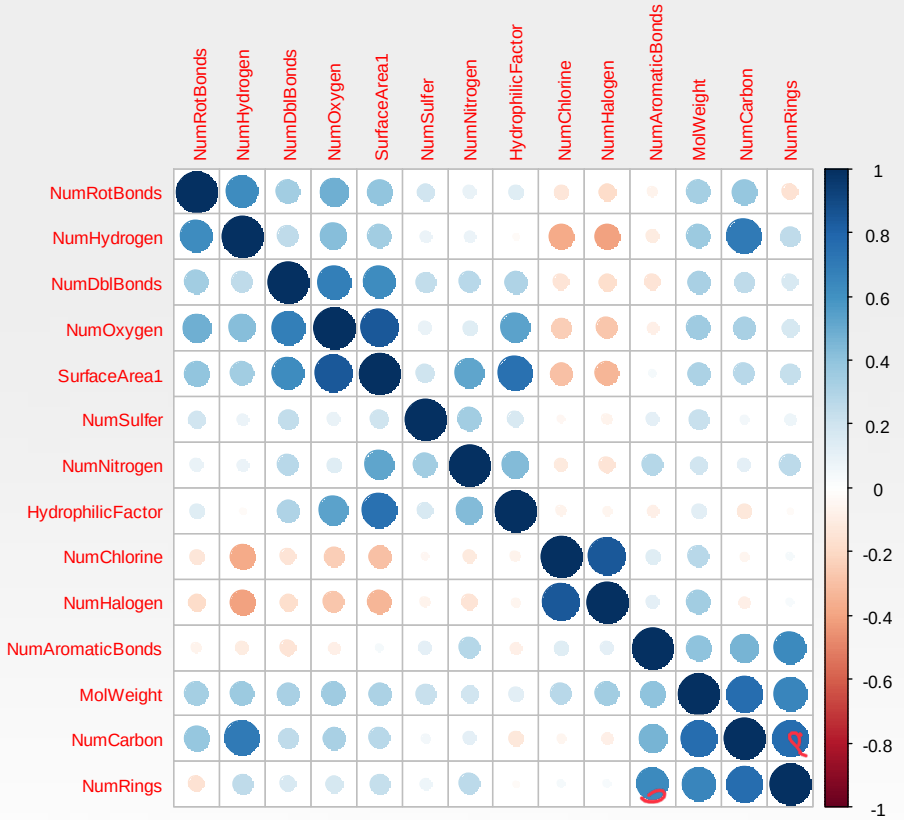
Many predictors (descriptors) are highly correlated!

Correlations between descriptors

```
#Remove high correlated predictors (cor>0.9)

tooHigh <- findCorrelation(cor(solTrainXtrans[, -notFingerprints]), .9)

corrplot::corrplot(cor(solTrainXtrans[, -notFingerprints][,-tooHigh]),
         order = "hclust", tl.cex = .8)
```

package function from this package.

Correlations between descriptors



Near zero variance predictors

#Remove near zero variance predictors

```
> # Remove near zero variance predictors
> nearZeroVar(solTrainXtrans)
[1] 154 199 200
```

we have three random predictors indexed by 154, 199, 200 that are near zero variance.

Skewness and Box-Cox transformation

```
#Skewness
library(e1071)
apply(solTrainXtrans[, -notFingerprints], 2, skewness)

#Box-Cox transformation
library(caret)
Original = as.matrix(solTrainXtrans[, -notFingerprints])
solTrainXtransBoxCox = BoxCoxTrans(Original)
solTrainXtransBoxCox
```

Skewness

```
> library(e1071)
> apply(solTrainXtrans[, -hotFingerprints], 2, skewness)
       MolWeight          NumAtoms     NumNonHAtoms         NumBonds
   -0.0002162255     -0.0713055864     -0.0555982369     -0.1881635919
    NumNonHBonds      NumMultBonds       NumRotBonds       NumDblBonds
    0.0432437801     -0.0946655258      0.0973568321      0.1496862360
NumAromaticBonds      NumHydrogen        NumCarbon       NumNitrogen
   -0.1463815894     -0.0407015938      0.0616443828      0.4262727429
       NumOxygen        NumSulfer       NumChlorine        NumHalogen
    0.1870352249      2.2707457390      1.4673236666      1.0331764976
        NumRings HydrophilicFactor       SurfaceArea1      SurfaceArea2
    0.0055727736      0.1003040295     -0.1316118434     -0.1681548297
```

Box-Cox transformation

```
> #Box-Cox transformation
> Original = as.matrix(solTrainXtrans[, -notFingerprints])
> solTrainXtransBoxCox = BoxCoxTrans(Original)
> solTrainXtransBoxCox
Box-Cox Transformation

19020 data points used to estimate Lambda

Input data summary:
   MolWeight        NumAtoms      NumNonHAtoms      NumBonds
 Min.   :3.852   Min.   :1.792   Min.   :1.099   Min.   :1.609
 1st Qu.:4.817   1st Qu.:2.890   1st Qu.:2.197   1st Qu.:2.890
 Median :5.194   Median :3.135   Median :2.565   Median :3.178
 Mean   :5.199   Mean   :3.174   Mean   :2.549   Mean   :3.176
 3rd Qu.:5.581   3rd Qu.:3.466   3rd Qu.:2.890   3rd Qu.:3.481
 Max.   :6.503   Max.   :4.554   Max.   :3.871   Max.   :4.585
  NumNonHBonds     NumMultBonds     NumRotBonds      NumDblBonds
 Min.   :0.7435   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:2.7592   1st Qu.:0.7988   1st Qu.:0.0000   1st Qu.:0.0000
 Median :3.3514   Median :2.9448   Median :1.0986   Median :0.5671
```