AOH.R

US16120

Wed Sep 26 14:16:26 2018

library(rcdk)

## Loading required package: rcdklibs

## Loading required package: rJava

library(tidyverse)

## -- Attaching packages ------------------------------------------------------ tidyverse 1.2.1 --

## v ggplot2 3.0.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.6  
## v tidyr 0.8.1 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts --------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::matches() masks rcdk::matches()

library(magrittr)

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

library(purrr)  
library(stringr)  
library(caret)

## Loading required package: lattice

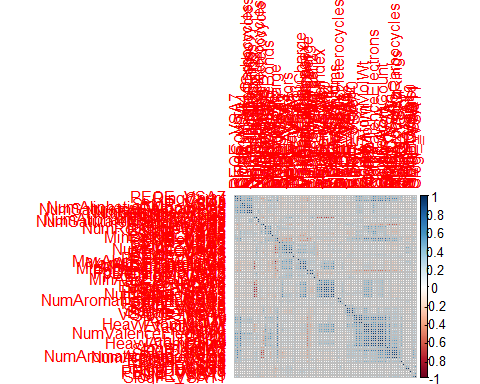
##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

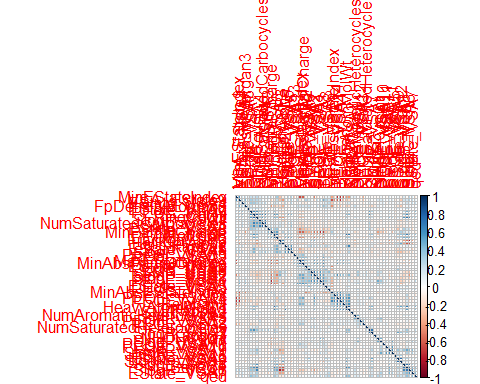
library(corrplot)

## corrplot 0.84 loaded

library(ggplot2)  
library(ggthemes)  
  
# read data  
  
## training data  
train <-  
 read.csv('cache/TR\_AOH\_516\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogOH, everything()) %>%  
 na.omit()  
  
X\_train <- train %>%  
 select(-LogOH)  
y\_train <- train %>%  
 select(LogOH) %>%  
 data.frame()  
  
## test data  
test <-  
 read.csv('cache/TST\_AOH\_176\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogOH, everything()) %>%  
 na.omit()  
  
X\_test <- test %>%  
 select(-LogOH)  
y\_test <- test %>%  
 select(LogOH) %>%  
 data.frame()  
  
# curate data  
  
## near-zero variance descriptors  
  
nzv <- nearZeroVar(X\_train, freqCut = 100/0)  
X\_train <- X\_train[ , -nzv]  
### and  
X\_test <- X\_test[ , -nzv]  
  
## highly correlated descriptors  
  
correlations <- cor(X\_train)  
corrplot::corrplot(correlations, order = 'hclust')



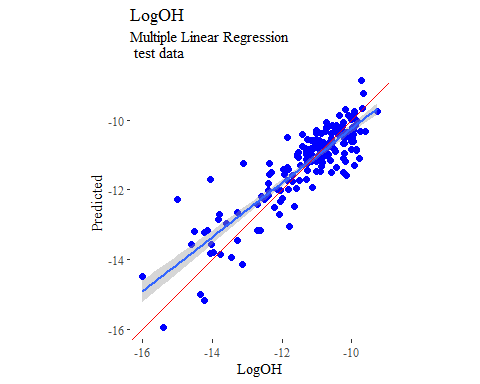
highCorr <- findCorrelation(correlations, cutoff = 0.85)  
X\_train <- X\_train[ , -highCorr]  
### and  
X\_test <- X\_test[ , -highCorr]  
  
correlations <- cor(X\_train)  
corrplot::corrplot(correlations, order = 'hclust')



## linear combinations  
  
comboInfo <- findLinearCombos(X\_train) # returns NULL  
# X\_train <- X\_train[ , -comboInfo$remove]  
# ### and  
# X\_test <- X\_test[ , -nzv]  
  
## center & scale descriptors  
  
preProcValues <- preProcess(X\_train, method = c("center", "scale"))  
  
X\_trainTransformed <- predict(preProcValues, X\_train)  
### and  
X\_testTransformed <- predict(preProcValues, X\_test)  
  
### PCA  
  
# pca <- preProcess(X\_trainTransformed, method = c('pca'))  
# X\_train\_pca <- predict(pca, X\_trainTransformed)  
# X\_test\_pca <- predict(pca, X\_testTransformed)  
#   
# train\_pca <- X\_train\_pca %>%  
# select(PC1, PC2) %>%  
# mutate(dataset = 'train')  
# test\_pca <- X\_test\_pca %>%  
# select(PC1, PC2) %>%  
# mutate(dataset = 'test')  
# pcaPts <- rbind(train\_pca, test\_pca)  
#   
# p <-  
# ggplot(pcaPts, aes(PC1, PC2)) +  
# geom\_point(aes(colour = factor(dataset), shape = factor(dataset))) +  
# ggthemes::theme\_tufte()  
# p  
  
# models  
  
fitControl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 repeats = 5)  
  
set.seed(350)  
  
## multiple linear regression  
  
trainSet <- cbind(y\_train, X\_trainTransformed)  
  
mlr <- train(LogOH ~ .,  
 data = trainSet,  
 method = 'lm',  
 trControl = fitControl)  
  
y\_predict <- predict(mlr, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.63823 -0.27939 0.01151 0.35837 1.87441   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.46845 0.37430 -6.595 5.04e-10 \*\*\*  
## LogOH 0.77766 0.03287 23.657 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5661 on 172 degrees of freedom  
## Multiple R-squared: 0.7649, Adjusted R-squared: 0.7636   
## F-statistic: 559.7 on 1 and 172 DF, p-value: < 2.2e-16

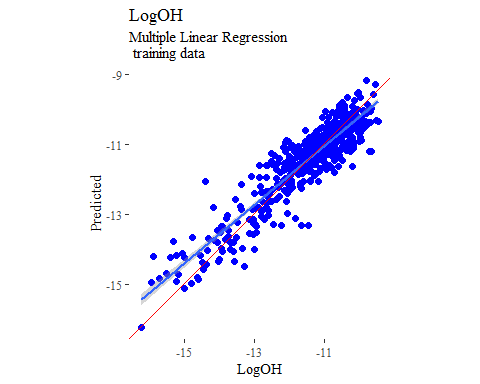
p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Multiple Linear Regression\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(mlr, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
mlrPR <- postResample(pred = y\_predict, obs = X\_trainTransformed)  
rmse\_train = c(mlrPR[1])  
r2\_train = c(mlrPR[2])  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.87509 -0.26401 0.01511 0.28206 1.83948   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.88237 0.18857 -9.982 <2e-16 \*\*\*  
## LogOH 0.83567 0.01636 51.076 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4722 on 513 degrees of freedom  
## Multiple R-squared: 0.8357, Adjusted R-squared: 0.8354   
## F-statistic: 2609 on 1 and 513 DF, p-value: < 2.2e-16

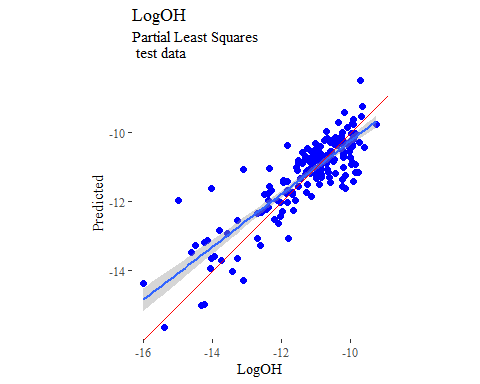
p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Multiple Linear Regression\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## partial least squares  
  
plsModel <- train(  
 LogOH ~ .,  
 data = trainSet,  
 method = 'pls',  
 tuneLength = 20,  
 trControl = fitControl  
)  
  
y\_predict <- predict(plsModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.65962 -0.31109 0.04547 0.39027 2.09410   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.58119 0.39664 -6.508 8.03e-10 \*\*\*  
## LogOH 0.76622 0.03483 21.997 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5999 on 172 degrees of freedom  
## Multiple R-squared: 0.7377, Adjusted R-squared: 0.7362   
## F-statistic: 483.8 on 1 and 172 DF, p-value: < 2.2e-16

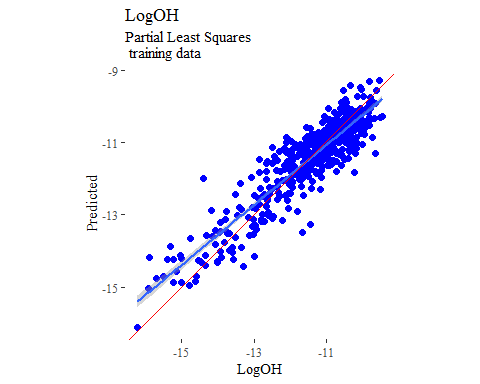
p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Partial Least Squares\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(plsModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.85922 -0.27491 0.03454 0.29723 1.88706   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.97156 0.19208 -10.26 <2e-16 \*\*\*  
## LogOH 0.82788 0.01667 49.67 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.481 on 513 degrees of freedom  
## Multiple R-squared: 0.8279, Adjusted R-squared: 0.8275   
## F-statistic: 2468 on 1 and 513 DF, p-value: < 2.2e-16

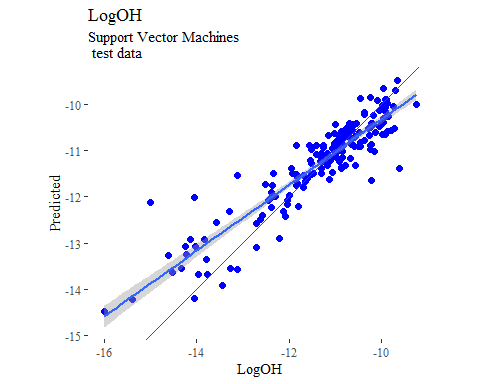
p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Partial Least Squares\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## support vector machines  
  
svmModel <- train(  
 LogOH ~ .,  
 data = trainSet,  
 method = 'svmRadial',  
 # tuneLength = 14,  
 trControl = fitControl  
)  
  
y\_predict <- predict(svmModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.34023 -0.20439 0.01778 0.23473 1.75583   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.21916 0.27158 -11.85 <2e-16 \*\*\*  
## LogOH 0.71055 0.02385 29.79 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4107 on 172 degrees of freedom  
## Multiple R-squared: 0.8377, Adjusted R-squared: 0.8367   
## F-statistic: 887.5 on 1 and 172 DF, p-value: < 2.2e-16

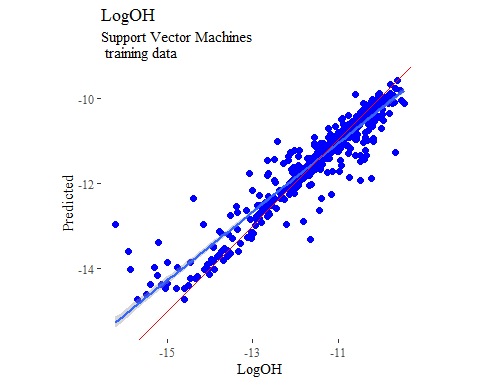
p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Support Vector Machines\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(svmModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.73020 -0.15250 0.03348 0.17459 2.27878   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.23447 0.14264 -15.66 <2e-16 \*\*\*  
## LogOH 0.80293 0.01238 64.88 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3572 on 513 degrees of freedom  
## Multiple R-squared: 0.8914, Adjusted R-squared: 0.8911   
## F-statistic: 4209 on 1 and 513 DF, p-value: < 2.2e-16

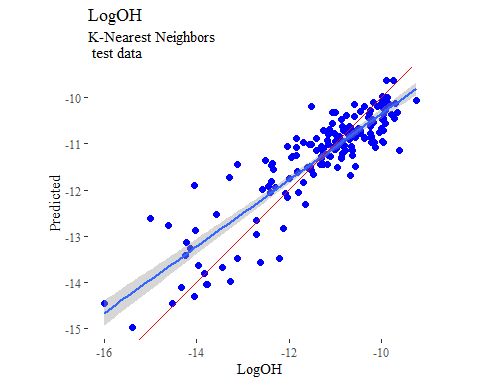
p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Support Vector Machines\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## k-nearest neighbors  
  
knnModel <- train(  
 LogOH ~ .,  
 data = trainSet,  
 method = 'knn',  
 tuneGrid = data.frame(.k = 1:20),  
 trControl = fitControl  
)  
  
y\_predict <- predict(knnModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.54403 -0.25552 0.01104 0.24276 1.35962   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.15007 0.32300 -9.752 <2e-16 \*\*\*  
## LogOH 0.71926 0.02837 25.355 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4885 on 172 degrees of freedom  
## Multiple R-squared: 0.7889, Adjusted R-squared: 0.7877   
## F-statistic: 642.9 on 1 and 172 DF, p-value: < 2.2e-16

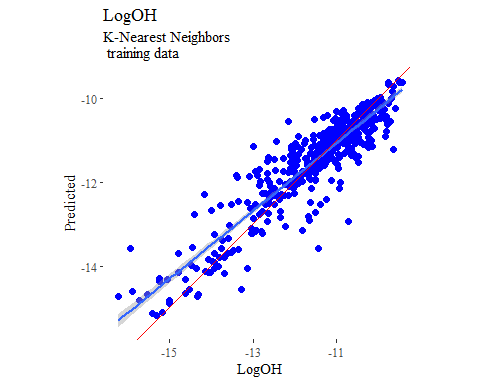
p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogOH',  
 subtitle = 'K-Nearest Neighbors\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(knnModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.16083 -0.19279 0.03843 0.22497 1.54963   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.12324 0.17924 -11.85 <2e-16 \*\*\*  
## LogOH 0.81271 0.01555 52.26 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4489 on 513 degrees of freedom  
## Multiple R-squared: 0.8419, Adjusted R-squared: 0.8415   
## F-statistic: 2731 on 1 and 513 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogOH',  
 subtitle = 'K-Nearest Neighbors\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## boosted trees  
  
gbmGrid <- expand.grid(  
 .interaction.depth = seq(1, 7, by = 2),  
 .n.trees = seq(100, 1000, by = 50),  
 .shrinkage = c(0.001, 0.1),  
 .n.minobsinnode = 3  
)  
  
treeModel <- train(  
 LogOH ~ .,  
 data = trainSet,  
 method = 'gbm',  
 tuneGrid = gbmGrid,  
 verbose = FALSE  
)

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

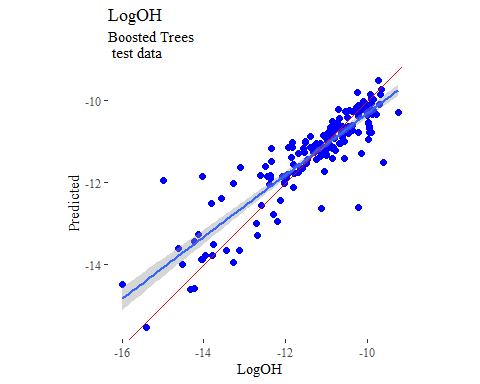
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 46: SMR\_VSA2 has no variation.

y\_predict <- predict(treeModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.14336 -0.19082 0.06032 0.27022 2.12590   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.73629 0.32983 -8.296 3.01e-14 \*\*\*  
## LogOH 0.75597 0.02897 26.098 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4988 on 172 degrees of freedom  
## Multiple R-squared: 0.7984, Adjusted R-squared: 0.7972   
## F-statistic: 681.1 on 1 and 172 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Boosted Trees\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(treeModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogOH, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogOH, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.087839 -0.019501 -0.000518 0.018391 0.074182   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.0395246 0.0108192 -3.653 0.000286 \*\*\*  
## LogOH 0.9965188 0.0009387 1061.558 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0271 on 513 degrees of freedom  
## Multiple R-squared: 0.9995, Adjusted R-squared: 0.9995   
## F-statistic: 1.127e+06 on 1 and 513 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogOH, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogOH',  
 subtitle = 'Boosted Trees\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p

