Jeff's Variable Analysis

Executive Summary

This is an analysis of the following predictor variables:
$ Filler.Level : num 121 119 120 118 119 ...
$ Filler.Speed : int 4002 3986 4020 4012 4010 4014 0 1004 4014 4028 ...
$ Temperature : num 66 67.6 67 65.6 65.6 66.2 65.8 65.2 65.4 66.6 ...
$ Usage.cont : num 16.2 19.9 17.8 17.4 17.7 ...
$ Carb.Flow : int 2932 3144 2914 3062 3054 2948 30 684 2902 3038 ...
$ Density : num 0.88 0.92 1.58 1.54 1.54 1.52 0.84 0.84 0.9 0.9 ...
$ MFR : num 725 727 735 731 723 ...
$ Balling : num 1.4 1.5 3.14 3.04 3.04 ...

The following activities were performed:
1) Review basic statistics for each variable.
2) Remove rows where response variable value is zero.
3) Determined high number of zeros for predictor variables were actually missing data so replaced zeros with NAs.
4) Generated imputed values for all missing data in predictor variables.
5) Generated boxplots, histograms, and scatter plots for each predictor variable to analyze distributions.
6) Performed Box Cox transformations for variable with poor distributions.
7) Performed 1-on-1 regression analysis for each predictor variable against the response variable for numerous regression model focusing on RMSE optimization and compared models results.
8) Performed a step-wise, both forward and backward, generalize linear model analysis focused on optimizing AIC to identify the most revelant variable relationships.

Variable Analysis Results:
The predictor variable distributions are generally poor, and all but one variable required a Box Cox-selected transformation which, in most cases, yielded only slight improvements.

The 1-on-1 regression modeling yielding generally poor results with poor RMSE values and terrible R-squared values resulting in the conlusion that no single predictor variable from the set has a significant influence or effect on the response variable.

The step-wise regression modeling resulting in generally poor results as well, but the resulting model did eliminate the variable Filler.Speed when arriving an the final model.

Conclusions:
None of the predictor variables in this set have a significant influence on PH, and all have troubled data distributions. A comparision of the distribution anomallies should be made across all of the data set variables to determine if there is a systemic cause. Pending that analysis, the variables in this set, minus the Filler.Speed variable, should be combined with other candidate predictor variables for further modeling analysis.

```
suppressWarnings(suppressMessages(library(knitr)))
suppressWarnings(suppressMessages(library(mice)))
suppressWarnings(suppressMessages(library(fBasics)))
suppressWarnings(suppressMessages(library(nnet)))
suppressWarnings(suppressMessages(library(kernlab)))
suppressWarnings(suppressMessages(library(caret)))
suppressWarnings(suppressMessages(library(randomForest)))
suppressWarnings(suppressMessages(library(mlbench)))
suppressWarnings(suppressMessages(library(MASS)))
suppressWarnings(suppressMessages(library(rpart)))
```

```r
suppressWarnings(suppressMessages(library(party)))
suppressWarnings(suppressMessages(library(partykit)))
suppressWarnings(suppressMessages(library(gbm)))
suppressWarnings(suppressMessages(library(ipred)))
suppressWarnings(suppressMessages(library(forecast)))

#suppressMessages(libraries("tidyverse", "nnet", "kernlab", "caret", "randomForest", "mlbench","MASS",

# read in the data locally
ph.data <- read.csv("/Users/JeffAtLaptop/Dropbox/School/DATA624-PredictiveAnalytics/Project2/StudentData
summary(ph.data)
```

```
##   Brand.Code          Carb.Volume     Fill.Ounces      PC.Volume
##  Length:2571        Min.   :0.000   Min.   : 0.00   Min.   :0.0000
##  Class :character   1st Qu.:5.293   1st Qu.:23.92   1st Qu.:0.2370
##  Mode  :character   Median :5.347   Median :23.97   Median :0.2700
##                     Mean   :5.349   Mean   :23.62   Mean   :0.2729
##                     3rd Qu.:5.453   3rd Qu.:24.03   3rd Qu.:0.3110
##                     Max.   :5.700   Max.   :24.32   Max.   :0.4780
##  Carb.Pressure     Carb.Temp          PSC             PSC.Fill
##  Min.   : 0.00   Min.   :  0.0   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:65.60   1st Qu.:138.4   1st Qu.:0.04800   1st Qu.:0.1000
##  Median :68.00   Median :140.8   Median :0.07600   Median :0.1800
##  Mean   :67.47   Mean   :139.7   Mean   :0.08349   Mean   :0.1936
##  3rd Qu.:70.60   3rd Qu.:143.8   3rd Qu.:0.11200   3rd Qu.:0.2600
##  Max.   :79.40   Max.   :154.0   Max.   :0.27000   Max.   :0.6200
##     PSC.CO2           Mnf.Flow       Carb.Pressure1  Fill.Pressure
##  Min.   :0.00000   Min.   :-100.20   Min.   :  0.0   Min.   : 0.00
##  1st Qu.:0.02000   1st Qu.:-100.00   1st Qu.:118.8   1st Qu.:46.00
##  Median :0.04000   Median :  64.80   Median :123.0   Median :46.40
##  Mean   :0.05556   Mean   :  24.55   Mean   :121.1   Mean   :47.51
##  3rd Qu.:0.08000   3rd Qu.: 140.80   3rd Qu.:125.4   3rd Qu.:50.00
##  Max.   :0.24000   Max.   : 229.40   Max.   :140.2   Max.   :60.40
##  Hyd.Pressure1   Hyd.Pressure2   Hyd.Pressure3   Hyd.Pressure4
##  Min.   :-0.80   Min.   : 0.00   Min.   :-1.20   Min.   :  0.00
##  1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.: 86.00
##  Median :11.40   Median :28.60   Median :27.40   Median : 96.00
##  Mean   :12.38   Mean   :20.84   Mean   :20.34   Mean   : 95.17
##  3rd Qu.:20.20   3rd Qu.:34.60   3rd Qu.:33.20   3rd Qu.:102.00
##  Max.   :58.00   Max.   :59.40   Max.   :50.00   Max.   :142.00
##   Filler.Level    Filler.Speed   Temperature      Usage.cont
##  Min.   :  0.0   Min.   :   0   Min.   : 0.00   Min.   : 0.00
##  1st Qu.: 96.2   1st Qu.:3815   1st Qu.:65.20   1st Qu.:18.35
##  Median :118.2   Median :3980   Median :65.60   Median :21.78
##  Mean   :108.4   Mean   :3605   Mean   :65.61   Mean   :20.95
##  3rd Qu.:120.0   3rd Qu.:3996   3rd Qu.:66.40   3rd Qu.:23.74
##  Max.   :161.2   Max.   :4030   Max.   :76.20   Max.   :25.90
##    Carb.Flow       Density           MFR           Balling
##  Min.   :   0   Min.   :0.000   Min.   :  0.0   Min.   :-0.170
##  1st Qu.:1133   1st Qu.:0.900   1st Qu.:694.9   1st Qu.: 1.496
##  Median :3028   Median :0.980   Median :721.4   Median : 1.648
##  Mean   :2466   Mean   :1.173   Mean   :646.0   Mean   : 2.197
##  3rd Qu.:3186   3rd Qu.:1.620   3rd Qu.:730.4   3rd Qu.: 3.292
##  Max.   :5104   Max.   :1.920   Max.   :868.6   Max.   : 4.012
```

```
##  Pressure.Vacuum        PH        Oxygen.Filler     Bowl.Setpoint
##  Min.   :-6.600   Min.   :0.000   Min.   :0.00000   Min.   :  0.0
##  1st Qu.:-5.600   1st Qu.:8.440   1st Qu.:0.02200   1st Qu.:100.0
##  Median :-5.400   Median :8.540   Median :0.03340   Median :120.0
##  Mean   :-5.216   Mean   :8.532   Mean   :0.04662   Mean   :109.2
##  3rd Qu.:-5.000   3rd Qu.:8.680   3rd Qu.:0.05960   3rd Qu.:120.0
##  Max.   :-3.600   Max.   :9.360   Max.   :0.40000   Max.   :140.0
##  Pressure.Setpoint Air.Pressurer      Alch.Rel        Carb.Rel
##  Min.   : 0.00     Min.   :140.8   Min.   :0.000   Min.   :0.000
##  1st Qu.:46.00     1st Qu.:142.2   1st Qu.:6.540   1st Qu.:5.340
##  Median :46.00     Median :142.6   Median :6.560   Median :5.400
##  Mean   :47.39     Mean   :142.8   Mean   :6.873   Mean   :5.416
##  3rd Qu.:50.00     3rd Qu.:143.0   3rd Qu.:7.220   3rd Qu.:5.540
##  Max.   :52.00     Max.   :148.2   Max.   :8.620   Max.   :6.060
##   Balling.Lvl
##  Min.   :0.000
##  1st Qu.:1.380
##  Median :1.480
##  Mean   :2.049
##  3rd Qu.:3.140
##  Max.   :3.660
```

```
# setup the dataset with just jeffs variables
jeffsList <- c("PH","Filler.Level", "Filler.Speed","Temperature","Usage.cont","Carb.Flow","Density","MF

jeffsVars <- data.frame(ph.data[,jeffsList])
summary(jeffsVars)
```

```
##        PH         Filler.Level    Filler.Speed   Temperature
##  Min.   :0.000   Min.   :  0.0   Min.   :   0   Min.   : 0.00
##  1st Qu.:8.440   1st Qu.: 96.2   1st Qu.:3815   1st Qu.:65.20
##  Median :8.540   Median :118.2   Median :3980   Median :65.60
##  Mean   :8.532   Mean   :108.4   Mean   :3605   Mean   :65.61
##  3rd Qu.:8.680   3rd Qu.:120.0   3rd Qu.:3996   3rd Qu.:66.40
##  Max.   :9.360   Max.   :161.2   Max.   :4030   Max.   :76.20
##    Usage.cont      Carb.Flow       Density           MFR
##  Min.   : 0.00   Min.   :   0   Min.   :0.000   Min.   :  0.0
##  1st Qu.:18.35   1st Qu.:1133   1st Qu.:0.900   1st Qu.:694.9
##  Median :21.78   Median :3028   Median :0.980   Median :721.4
##  Mean   :20.95   Mean   :2466   Mean   :1.173   Mean   :646.0
##  3rd Qu.:23.74   3rd Qu.:3186   3rd Qu.:1.620   3rd Qu.:730.4
##  Max.   :25.90   Max.   :5104   Max.   :1.920   Max.   :868.6
##     Balling
##  Min.   :-0.170
##  1st Qu.: 1.496
##  Median : 1.648
##  Mean   : 2.197
##  3rd Qu.: 3.292
##  Max.   : 4.012
```

```
# run the basic stats on the variables
# Let's start by exploring the type of each variable
types <- sapply(1:length(jeffsVars),function(x) typeof(jeffsVars[,x]))
types.df <- data.frame(VAR=names(jeffsVars),TYPE=types)
kable(types.df)
```

VAR	TYPE
PH	double
Filler.Level	double
Filler.Speed	integer
Temperature	double
Usage.cont	double
Carb.Flow	integer
Density	double
MFR	double
Balling	double

```
# Show a statistical summary of the data
kable(summary(jeffsVars[,1:5]))
```

PH	Filler.Level	Filler.Speed	Temperature	Usage.cont
Min. :0.000	Min. : 0.0	Min. : 0	Min. : 0.00	Min. : 0.00
1st Qu.:8.440	1st Qu.: 96.2	1st Qu.:3815	1st Qu.:65.20	1st Qu.:18.35
Median :8.540	Median :118.2	Median :3980	Median :65.60	Median :21.78
Mean :8.532	Mean :108.4	Mean :3605	Mean :65.61	Mean :20.95
3rd Qu.:8.680	3rd Qu.:120.0	3rd Qu.:3996	3rd Qu.:66.40	3rd Qu.:23.74
Max. :9.360	Max. :161.2	Max. :4030	Max. :76.20	Max. :25.90

```
kable(summary(jeffsVars[,6:9]))
```

Carb.Flow	Density	MFR	Balling
Min. : 0	Min. :0.000	Min. : 0.0	Min. :-0.170
1st Qu.:1133	1st Qu.:0.900	1st Qu.:694.9	1st Qu.: 1.496
Median :3028	Median :0.980	Median :721.4	Median : 1.648
Mean :2466	Mean :1.173	Mean :646.0	Mean : 2.197
3rd Qu.:3186	3rd Qu.:1.620	3rd Qu.:730.4	3rd Qu.: 3.292
Max. :5104	Max. :1.920	Max. :868.6	Max. : 4.012

```
# based on the summary, some of the PH values are 0.0, these rows should be removed since
# we cannot calculate for a 0.0 PH
jeffsVars <- jeffsVars[jeffsVars$PH>0.0,]

# now we'll check how many variables have values of zero
# show the frequency of zeros in the data for each variable
apply(jeffsVars,2,function(x){sum(abs(x-0.0)<=1e-6)})
```

```
##           PH Filler.Level Filler.Speed  Temperature   Usage.cont
##            0           16           54           12            5
##    Carb.Flow      Density          MFR      Balling
##            2            0          208            0
```

```
# based on these counts, we'll replace zeros with NAs for the following variables
index <- which(jeffsVars$Filler.Level <= 0.0)
is.na(jeffsVars$Filler.Level) <- index
index <- which(jeffsVars$Filler.Speed <= 0.0)
is.na(jeffsVars$Filler.Speed) <- index
```

```
index <- which(jeffsVars$Temperature <= 0.0)
is.na(jeffsVars$Temperature) <- index
index <- which(jeffsVars$Usage.cont <= 0.0)
is.na(jeffsVars$Usage.cont) <- index
index <- which(jeffsVars$Carb.Flow <= 0.0)
is.na(jeffsVars$Carb.Flow) <- index
index <- which(jeffsVars$MFR<= 0.0)
is.na(jeffsVars$MFR) <- index
# now we'll impute the NA values which represent missing values
#uses Predictive Mean Matching.
jeffsVars.imp <- complete(mice(jeffsVars, m = 3, print=F))
#jeffsVars.imp <- complete(jeffsVars.tmp,1)
# check that the NAs have all been resolved with imputed data
any(is.na(jeffsVars.imp))
```

```
## [1] FALSE
```

```
# generate the basic stats for all variables, including the imputed values
kable(basicStats(jeffsVars.imp[,1:5]))
```

	PH	Filler.Level	Filler.Speed	Temperature	Usage.cont
nobs	2567.000000	2567.000000	2.567000e+03	2.567000e+03	2567.000000
NAs	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000
Minimum	7.880000	55.800000	9.980000e+02	6.360000e+01	12.080000
Maximum	9.360000	161.200000	4.030000e+03	7.620000e+01	25.900000
1. Quartile	8.440000	97.700000	3.851000e+03	6.520000e+01	18.380000
3. Quartile	8.680000	120.000000	3.997000e+03	6.640000e+01	23.750000
Mean	8.545649	109.242929	3.651046e+03	6.596619e+01	20.995014
Median	8.540000	118.400000	3.980000e+03	6.560000e+01	21.780000
Sum	21936.680000	280426.600000	9.372236e+06	1.693352e+05	53894.200000
SE Mean	0.003405	0.309647	1.614337e+01	2.720100e-02	0.058705
LCL Mean	8.538972	108.635745	3.619391e+03	6.591285e+01	20.879899
UCL Mean	8.552325	109.850114	3.682702e+03	6.601953e+01	21.110129
Variance	0.029762	246.127868	6.689818e+05	1.899339e+00	8.846730
Stdev	0.172516	15.688463	8.179131e+02	1.378165e+00	2.974345
Skewness	-0.290644	-0.846172	-2.633553e+00	2.385686e+00	-0.535497
Kurtosis	0.064429	0.038673	5.338732e+00	1.022225e+01	-1.014588

```
kable(basicStats(jeffsVars.imp[,6:9]))
```

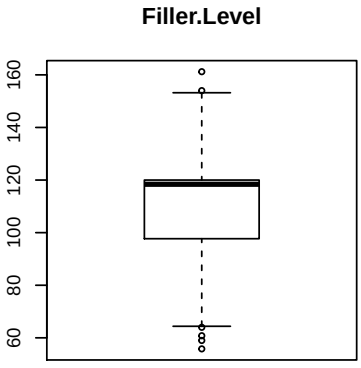
	Carb.Flow	Density	MFR	Balling
nobs	2.567000e+03	2567.000000	2.567000e+03	2567.000000
NAs	0.000000e+00	0.000000	0.000000e+00	0.000000
Minimum	2.600000e+01	0.240000	3.140000e+01	0.160000
Maximum	5.104000e+03	1.920000	8.686000e+02	4.012000
1. Quartile	1.166000e+03	0.900000	6.950000e+02	1.496000
3. Quartile	3.187000e+03	1.620000	7.304000e+02	3.292000
Mean	2.471698e+03	1.174453	6.732807e+02	2.199842
Median	3.028000e+03	0.980000	7.214000e+02	1.648000
Sum	6.344848e+06	3014.820000	1.728312e+06	5646.994000
SE Mean	2.112710e+01	0.007440	2.590541e+00	0.018347
LCL Mean	2.430270e+03	1.159863	6.682010e+02	2.163866

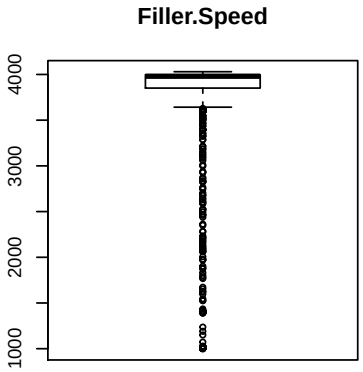
	Carb.Flow	Density	MFR	Balling
UCL Mean	2.513126e+03	1.189042	6.783605e+02	2.235818
Variance	1.145792e+06	0.142105	1.722689e+04	0.864058
Stdev	1.070417e+03	0.376968	1.312513e+02	0.929547
Skewness	-9.907760e-01	0.531113	-2.772375e+00	0.600459
Kurtosis	-5.754890e-01	-1.209505	6.841615e+00	-1.399653
Now we'll proc	eed with our ana	lysis of the p	redictor variabl	es.

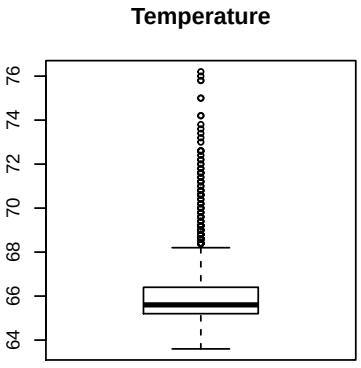
```
# set up the data combinations needed for the analysis
jeffsVars.pred <- jeffsVars.imp[,-c(1)]
head(jeffsVars.pred)
```

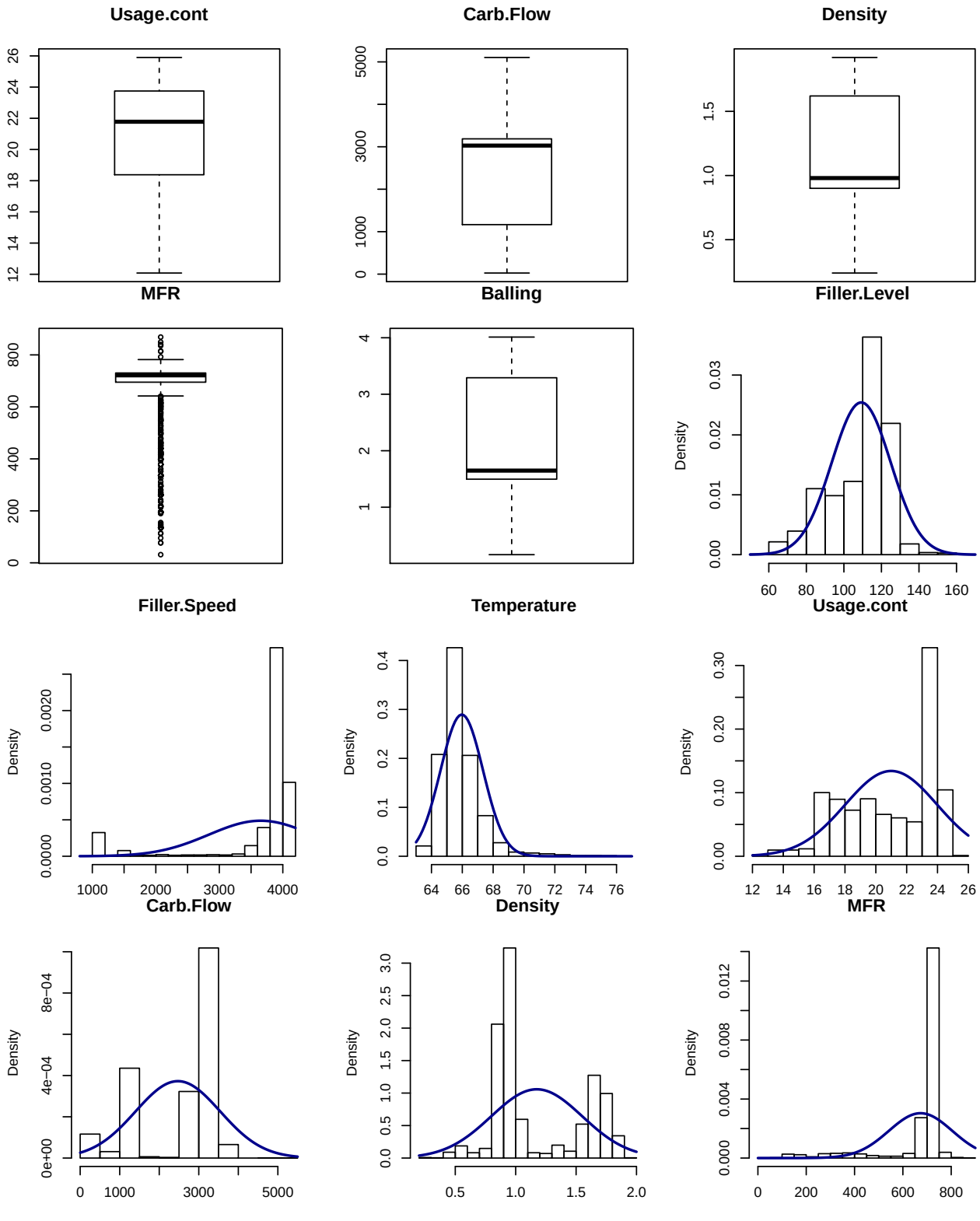
```
##   Filler.Level Filler.Speed Temperature Usage.cont Carb.Flow Density   MFR
## 1        121.2         4002        66.0      16.18      2932    0.88 725.0
## 2        118.6         3986        67.6      19.90      3144    0.92 726.8
## 3        120.0         4020        67.0      17.76      2914    1.58 735.0
## 4        117.8         4012        65.6      17.42      3062    1.54 730.6
## 5        118.6         4010        65.6      17.68      3054    1.54 722.8
## 6        120.2         4014        66.2      23.82      2948    1.52 738.8
##   Balling
## 1   1.398
## 2   1.498
## 3   3.142
## 4   3.042
## 5   3.042
## 6   2.992
```

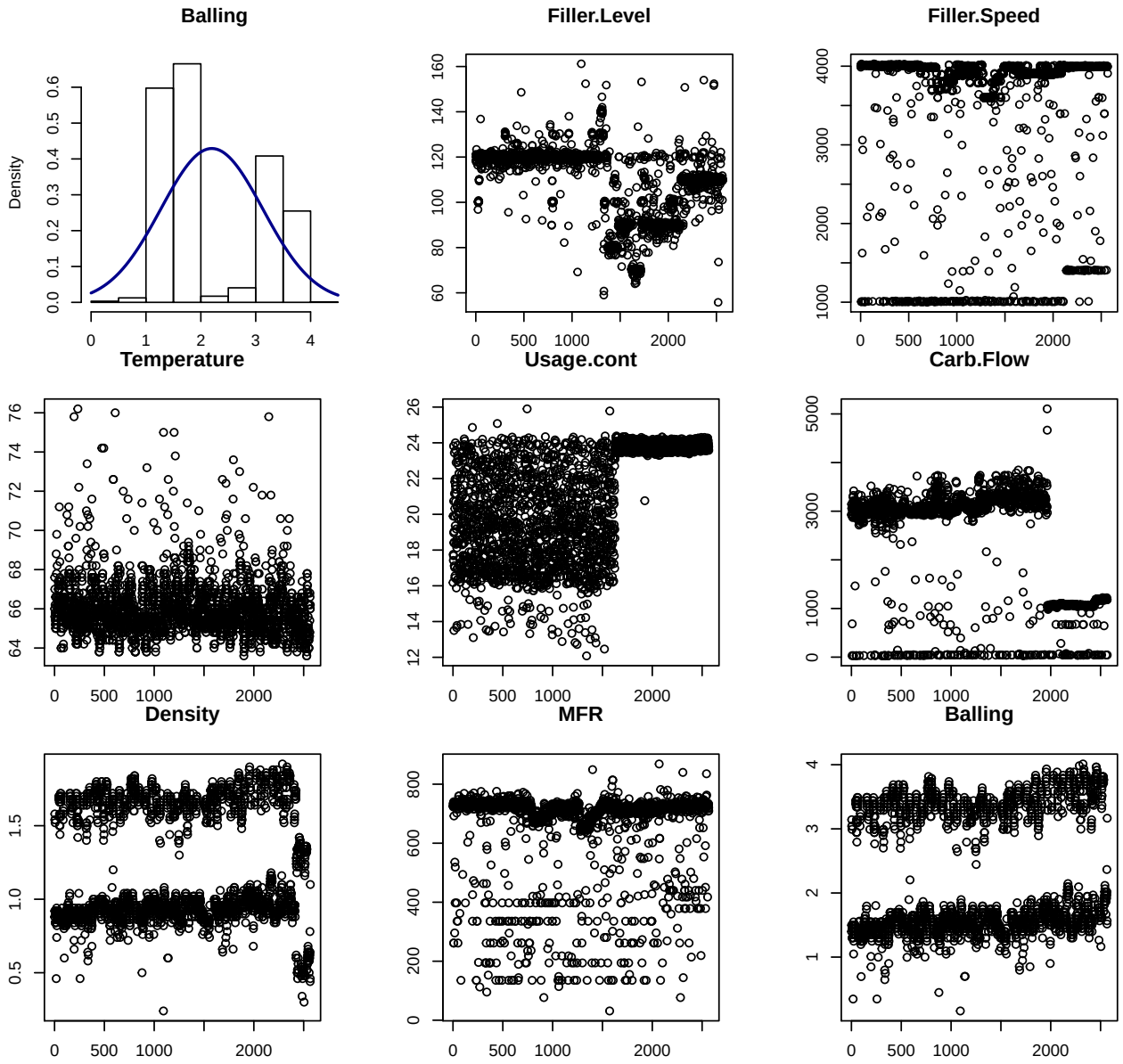
```
# required data sets include a train and test set
# form the training and test data partitions
n = nrow(jeffsVars.imp)
index <- sample(1:n, size = round(.80*n), replace = FALSE)
jeffsVars.train <- jeffsVars.imp[index,]
jeffsVars.test <- jeffsVars.imp[-index,]
jeffsVars.pred.train <- jeffsVars.pred[index,]
jeffsVars.pred.test <- jeffsVars.pred[-index,]
```








Usage.cont
Carb.Flow
Density
MFR
Balling
Filler.Level
Filler.Speed
Temperature
Usage.cont
Carb.Flow
Density
MFR



Based on a quick analysis of the basic statistics and plots, all of the predictor variables except Filler.Level will require a transformation to address issues with the variable's data distribution.

The Filler.Level variable distribution is fairly normal with few outliers so we will not perform any transformations and, instead, perform an analysis of how the variable interacts with the response variable in various regression models.

```
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"

# LM
set.seed(624)
fit.lm <- train(PH~Filler.Level, data=jeffsVars.imp, method="lm", metric=metric,
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
```

```r
fit.glm <- train(PH~Filler.Level, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~Filler.Level, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~Filler.Level, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
fit.knn <- train(PH~Filler.Level, data=jeffsVars.imp, method="knn", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

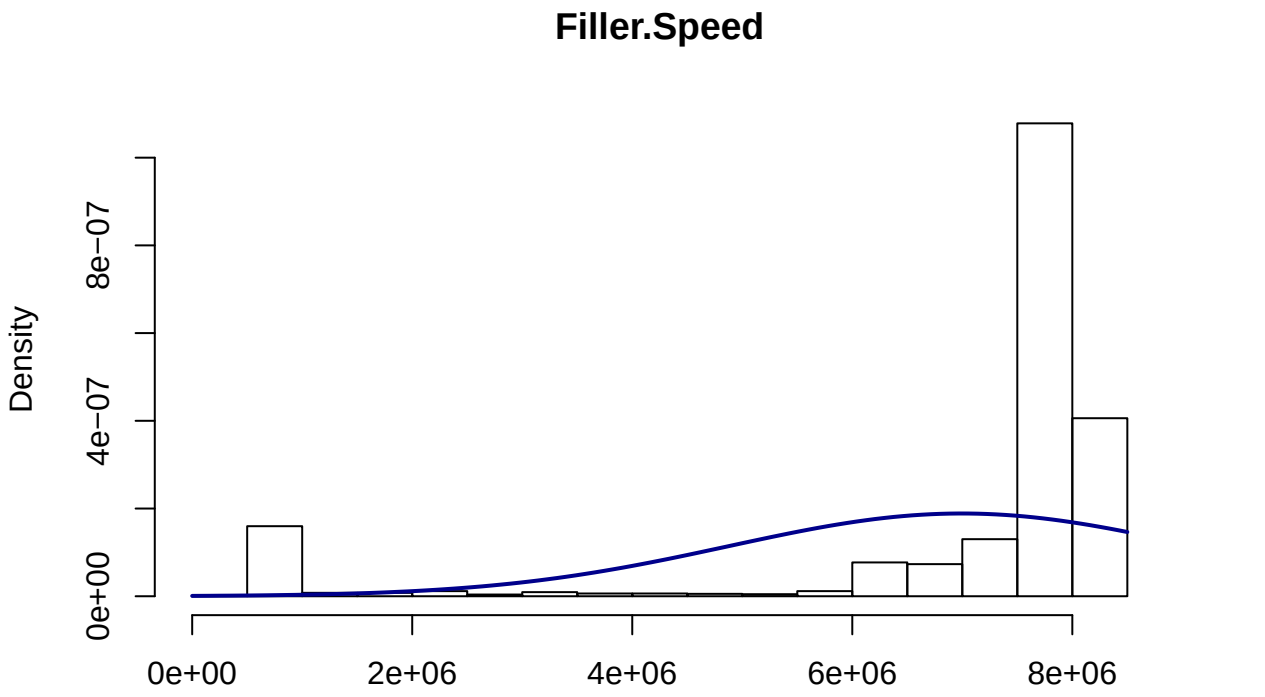
```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1201909 0.1284526 0.1314975 0.1311820 0.1339421 0.1392124    0
## GLM  0.1201909 0.1284526 0.1314975 0.1311820 0.1339421 0.1392124    0
## SVM  0.1133574 0.1221499 0.1236970 0.1240205 0.1254063 0.1344164    0
## CART 0.1136421 0.1239664 0.1259076 0.1263105 0.1289934 0.1330929    0
## KNN  0.1161650 0.1252560 0.1265473 0.1273230 0.1300956 0.1367542    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1453883 0.1593047 0.1629767 0.1632092 0.1672134 0.1767618    0
## GLM  0.1453883 0.1593047 0.1629767 0.1632092 0.1672134 0.1767618    0
## SVM  0.1427099 0.1543673 0.1575158 0.1583896 0.1612493 0.1752402    0
## CART 0.1399061 0.1549897 0.1591494 0.1587623 0.1618148 0.1731220    0
## KNN  0.1445333 0.1568286 0.1601584 0.1605119 0.1634267 0.1780267    0
##
## Rsquared
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
```

```
## LM   0.02986735 0.0854933 0.1077226 0.1079414 0.1244161 0.1848261    0
## GLM  0.02986735 0.0854933 0.1077226 0.1079414 0.1244161 0.1848261    0
## SVM  0.06404251 0.1543401 0.1733811 0.1688094 0.1924288 0.2262500    0
## CART 0.06792278 0.1367975 0.1608383 0.1549523 0.1815269 0.2076281    0
## KNN  0.06679426 0.1263891 0.1454547 0.1408140 0.1602291 0.1798211    0
```

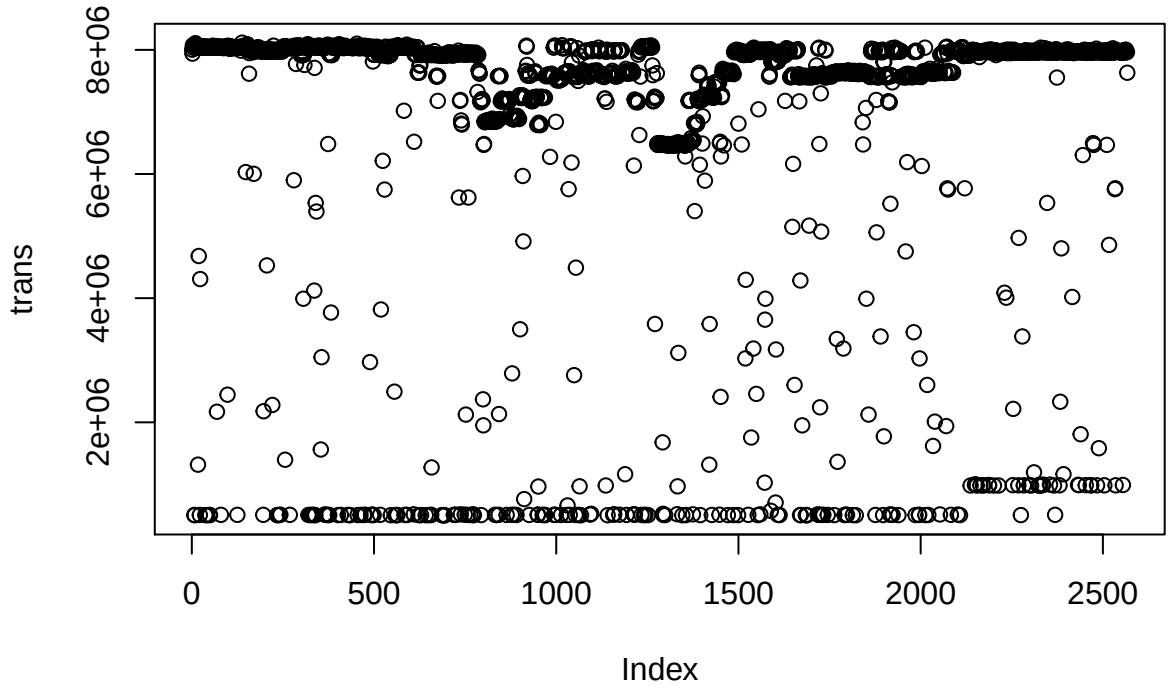
Next, we'll explore the effect of Box Cox transformations on the predictor variables with skewed or non-normal distributions.

We'll start with Filler.Speed first.

```
# perform the Box Cox transformation and then look at the distribution
lambda <- BoxCox.lambda(jeffsVars.pred$Filler.Speed)
trans <- BoxCox(jeffsVars.pred$Filler.Speed,lambda)
  m <- mean(trans)
  s <- sd(trans)
  hist(trans,freq=FALSE,main = "Filler.Speed",xlab="")
  curve(dnorm(x,mean=m,sd=s),col="darkblue",lwd=2,add=TRUE)
```




```
  plot(trans)
```




```
jeffsVars.imp <- cbind(jeffsVars.imp,Filler.Speed.Trans=trans)
```

```
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"

# LM
set.seed(624)
fit.lm <- train(PH~Filler.Speed.Trans, data=jeffsVars.imp, method="lm", metric=metric,
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
fit.glm <- train(PH~Filler.Speed.Trans, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~Filler.Speed.Trans, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~Filler.Speed.Trans, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
```

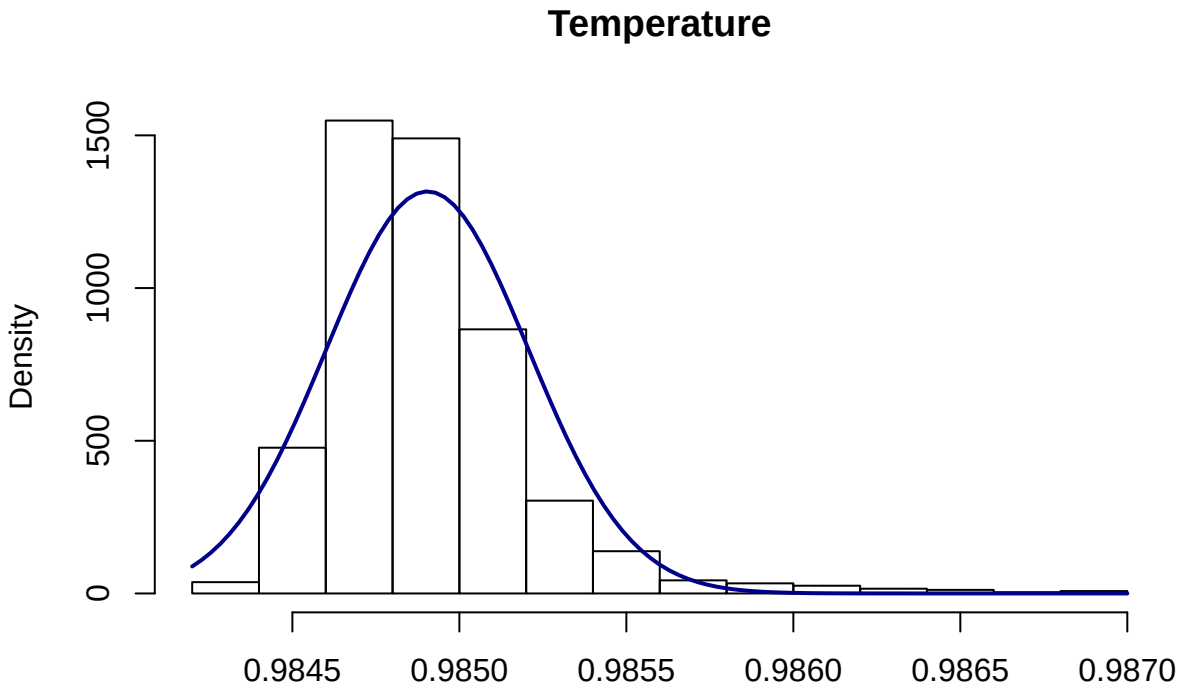
```
fit.knn <- train(PH~Filler.Speed.Trans, data=jeffsVars.imp, method="knn", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1271923 0.1374589 0.1387211 0.1388418 0.1417721 0.1451230    0
## GLM  0.1271923 0.1374589 0.1387211 0.1388418 0.1417721 0.1451230    0
## SVM  0.1268878 0.1365774 0.1383034 0.1382064 0.1407241 0.1454906    0
## CART 0.1162936 0.1272527 0.1310536 0.1296433 0.1323307 0.1384287    0
## KNN  0.1160903 0.1270957 0.1300847 0.1290870 0.1318104 0.1374562    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1547218 0.1695665 0.1715192 0.1724542 0.1774874 0.1842025    0
## GLM  0.1547218 0.1695665 0.1715192 0.1724542 0.1774874 0.1842025    0
## SVM  0.1547329 0.1693693 0.1711177 0.1722107 0.1770796 0.1839634    0
## CART 0.1460320 0.1623519 0.1648594 0.1650456 0.1684791 0.1796409    0
## KNN  0.1471417 0.1609710 0.1641329 0.1641600 0.1679500 0.1781773    0
##
## Rsquared
##              Min.      1st Qu.      Median        Mean     3rd Qu.
## LM   0.0000496871 0.0003409577 0.002479442 0.004181521 0.006748008
## GLM  0.0000496871 0.0003409577 0.002479442 0.004181521 0.006748008
## SVM  0.0001831605 0.0021416531 0.003759548 0.006611637 0.006423930
## CART 0.0469298405 0.0814837427 0.092009728 0.095301140 0.108929678
## KNN  0.0591161985 0.0781351847 0.098736862 0.102030486 0.115897337
##           Max. NA's
## LM   0.02018689    0
## GLM  0.02018689    0
## SVM  0.03372577    0
## CART 0.16259133    0
## KNN  0.18211138    0
```

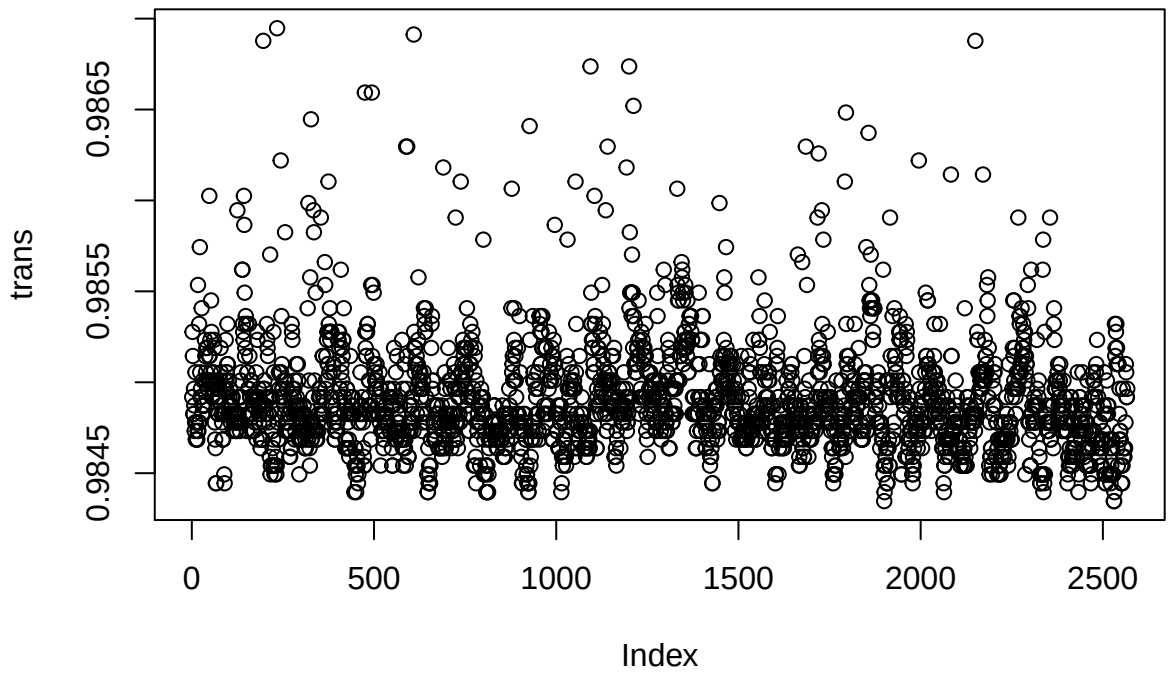
Next, we'll work with the Temperature variable.
We'll start with Filler.Speed first.

```
# look at a couple of the variables first
lambda <- BoxCox.lambda(jeffsVars.pred$Temperature)
trans <- BoxCox(jeffsVars.pred$Temperature,lambda)
  m <- mean(trans)
  s <- sd(trans)
```

```r
hist(trans,freq=FALSE,main = "Temperature",xlab="")
curve(dnorm(x,mean=m,sd=s),col="darkblue",lwd=2,add=TRUE)
```




```r
plot(trans)
```




```r
jeffsVars.imp <- cbind(jeffsVars.imp,Temperature.Trans=trans)
```

```r
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"
```

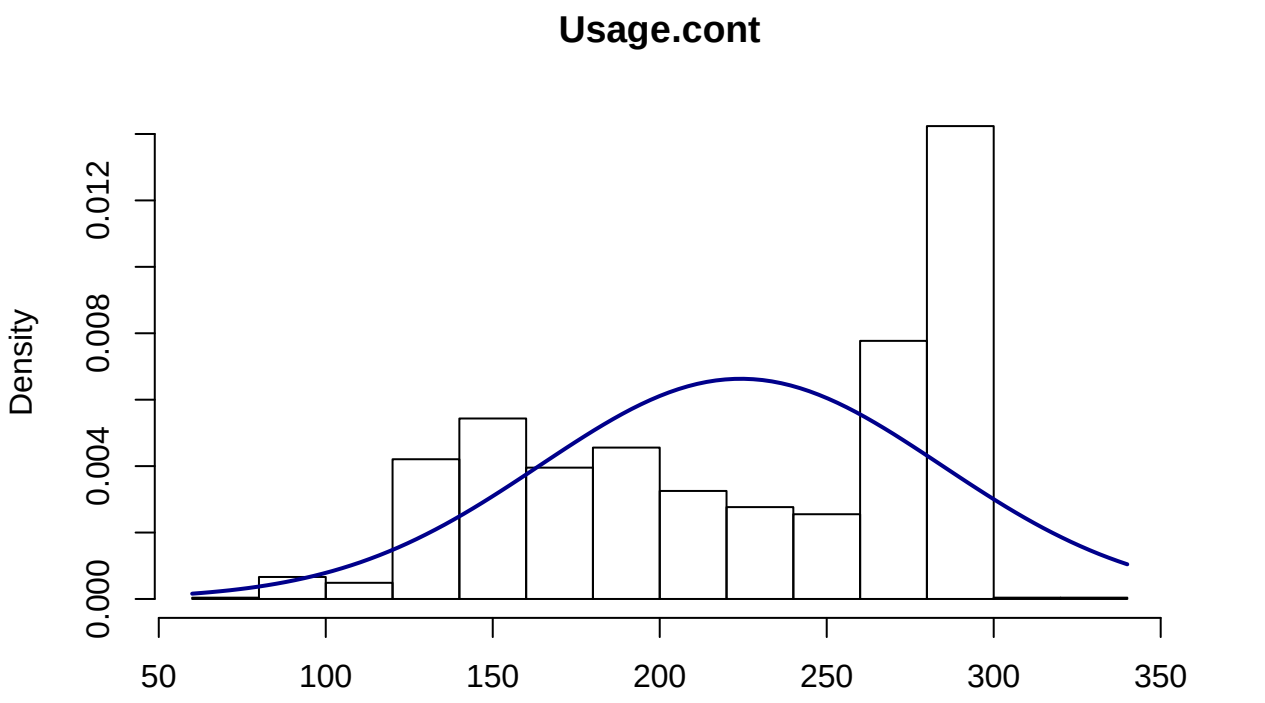
```r
# LM
set.seed(624)
fit.lm <- train(PH~Temperature.Trans, data=jeffsVars.imp, method="lm", metric=metric,
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
fit.glm <- train(PH~Temperature.Trans, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~Temperature.Trans, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~Temperature.Trans, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
fit.knn <- train(PH~Temperature.Trans, data=jeffsVars.imp, method="knn", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1278340 0.1339411 0.1368592 0.1369722 0.1402438 0.1427002    0
## GLM  0.1278340 0.1339411 0.1368592 0.1369722 0.1402438 0.1427002    0
## SVM  0.1260045 0.1314833 0.1366965 0.1352213 0.1393344 0.1413296    0
## CART 0.1264456 0.1319398 0.1364626 0.1357145 0.1393085 0.1444363    0
## KNN  0.1255462 0.1313143 0.1361359 0.1352214 0.1392505 0.1433164    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1554725 0.1661963 0.1700972 0.1701033 0.1754110 0.1818608    0
```

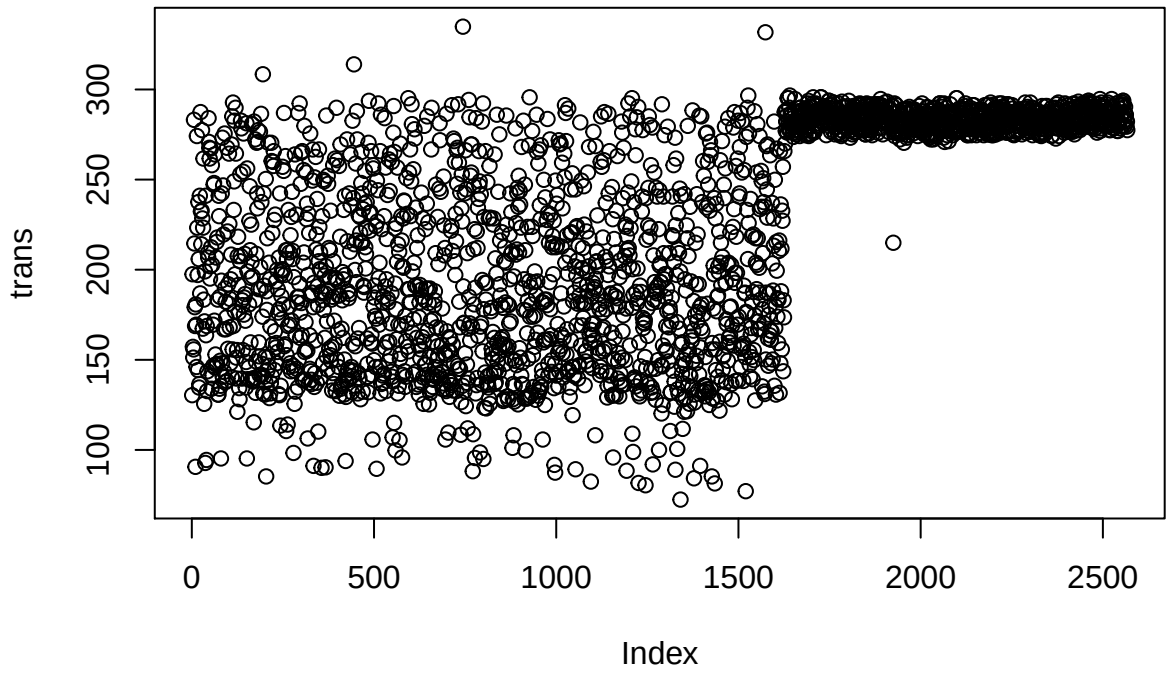
```
## GLM  0.1554725 0.1661963 0.1700972 0.1701033 0.1754110 0.1818608    0
## SVM  0.1549669 0.1650627 0.1676289 0.1682999 0.1736941 0.1795348    0
## CART 0.1555936 0.1649727 0.1680420 0.1689323 0.1734466 0.1818727    0
## KNN  0.1531764 0.1649054 0.1679697 0.1684522 0.1734633 0.1806526    0
##
## Rsquared
##              Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## LM   0.0001357553 0.01486673 0.02882928 0.03146091 0.04211416 0.08112341
## GLM  0.0001357553 0.01486673 0.02882928 0.03146091 0.04211416 0.08112341
## SVM  0.0058105075 0.02962261 0.05327520 0.05333067 0.07889755 0.10772969
## CART 0.0052320741 0.02931945 0.03919873 0.04854575 0.06525373 0.13217649
## KNN  0.0055890244 0.03007223 0.04850929 0.05315718 0.06540948 0.13996601
##      NA's
## LM      0
## GLM     0
## SVM     0
## CART    0
## KNN     0
```

Next, we'll work with the Usage.cont variable.

```
# look at a couple of the variables first
lambda <- BoxCox.lambda(jeffsVars.pred$Usage.cont)
trans <- BoxCox(jeffsVars.pred$Usage.cont,lambda)
  m <- mean(trans)
  s <- sd(trans)
  hist(trans,freq=FALSE,main = "Usage.cont",xlab="")
  curve(dnorm(x,mean=m,sd=s),col="darkblue",lwd=2,add=TRUE)
```




```
  plot(trans)
```




```
jeffsVars.imp <- cbind(jeffsVars.imp,Usage.cont.Trans=trans)
```

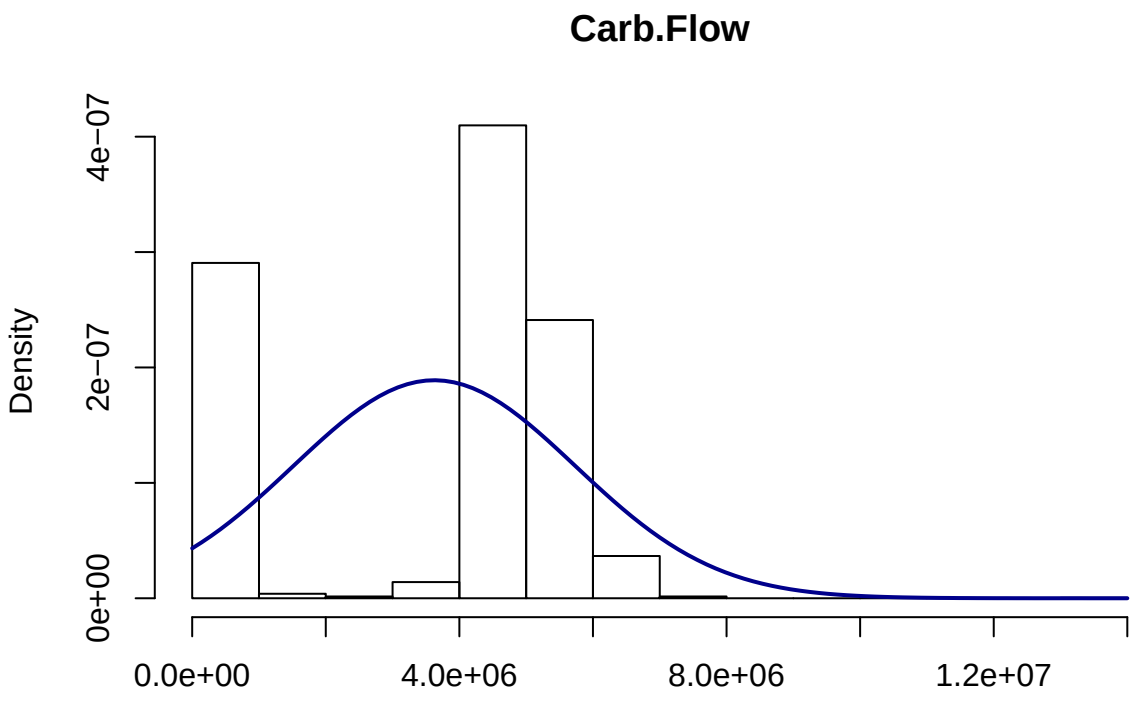
```
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"

# LM
set.seed(624)
fit.lm <- train(PH~Usage.cont.Trans, data=jeffsVars.imp, method="lm", metric=metric,
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
fit.glm <- train(PH~Usage.cont.Trans, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~Usage.cont.Trans, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~Usage.cont.Trans, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
fit.knn <- train(PH~Usage.cont.Trans, data=jeffsVars.imp, method="knn", metric=metric,
```

```
                preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

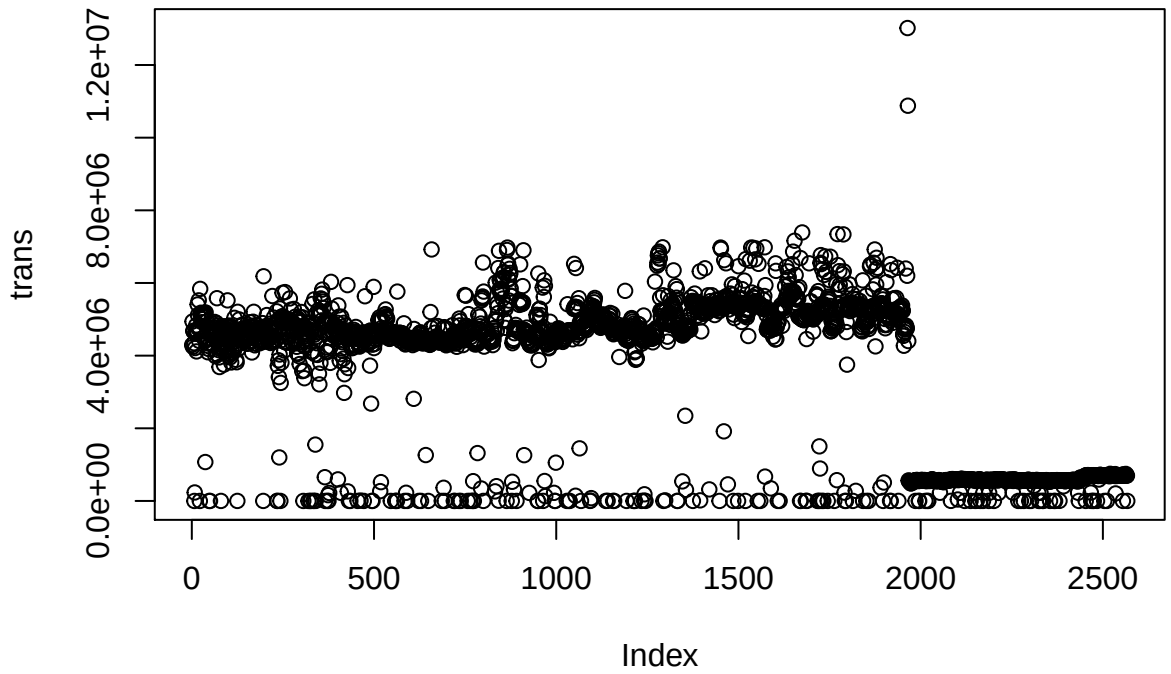
```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1212672 0.1263601 0.1298071 0.1296214 0.1326648 0.1367856    0
## GLM  0.1212672 0.1263601 0.1298071 0.1296214 0.1326648 0.1367856    0
## SVM  0.1184250 0.1222279 0.1250092 0.1254985 0.1275440 0.1368081    0
## CART 0.1165529 0.1217735 0.1245267 0.1251947 0.1278804 0.1356898    0
## KNN  0.1189963 0.1249875 0.1272540 0.1287365 0.1310773 0.1441659    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1474118 0.1583714 0.1635745 0.1628401 0.1667021 0.1773201    0
## GLM  0.1474118 0.1583714 0.1635745 0.1628401 0.1667021 0.1773201    0
## SVM  0.1442978 0.1546044 0.1583267 0.1585363 0.1613678 0.1745178    0
## CART 0.1419906 0.1532369 0.1579076 0.1574653 0.1606953 0.1736164    0
## KNN  0.1464399 0.1595329 0.1619874 0.1622126 0.1639815 0.1805287    0
##
## Rsquared
##            Min.    1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.05316995 0.08853612 0.1124225 0.1118365 0.1312455 0.2085573    0
## GLM  0.05316995 0.08853612 0.1124225 0.1118365 0.1312455 0.2085573    0
## SVM  0.06370052 0.14644607 0.1611852 0.1622991 0.1996278 0.2260597    0
## CART 0.09043158 0.14867365 0.1722759 0.1695019 0.2073174 0.2353797    0
## KNN  0.05925625 0.10764813 0.1410987 0.1360204 0.1652828 0.2024995    0
```

Next, we'll work with the Carb.Flow variable.

```
# look at a couple of the variables first
lambda <- BoxCox.lambda(jeffsVars.pred$Carb.Flow)
trans <- BoxCox(jeffsVars.pred$Carb.Flow,lambda)
  m <- mean(trans)
  s <- sd(trans)
  hist(trans,freq=FALSE,main = "Carb.Flow",xlab="")
  curve(dnorm(x,mean=m,sd=s),col="darkblue",lwd=2,add=TRUE)
```




```
plot(trans)
```




```
jeffsVars.imp <- cbind(jeffsVars.imp,Carb.Flow.Trans=trans)
```

```
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"

# LM
set.seed(624)
fit.lm <- train(PH~Carb.Flow.Trans, data=jeffsVars.imp, method="lm", metric=metric,
```

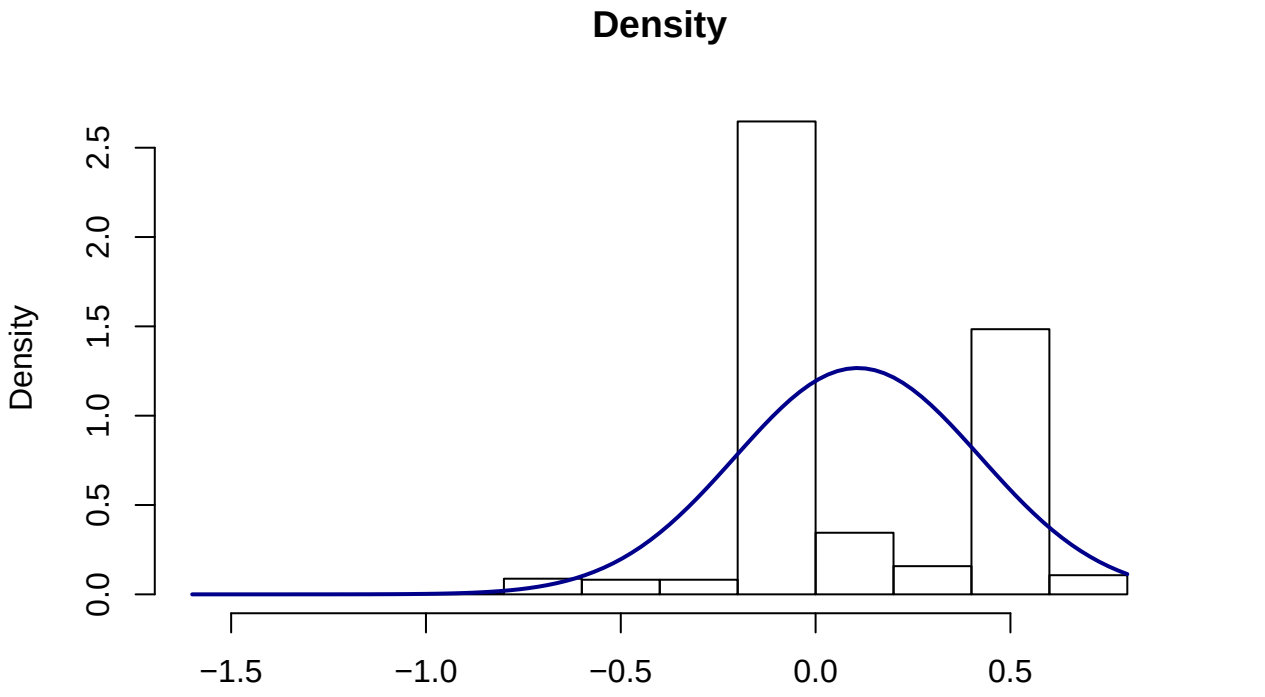
```
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
fit.glm <- train(PH~Carb.Flow.Trans, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~Carb.Flow.Trans, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~Carb.Flow.Trans, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
fit.knn <- train(PH~Carb.Flow.Trans, data=jeffsVars.imp, method="knn", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1241236 0.1349304 0.1372162 0.1370214 0.1398379 0.1451241    0
## GLM  0.1241236 0.1349304 0.1372162 0.1370214 0.1398379 0.1451241    0
## SVM  0.1182172 0.1285998 0.1315154 0.1314406 0.1352706 0.1396238    0
## CART 0.1202369 0.1336060 0.1366733 0.1369843 0.1404501 0.1484637    0
## KNN  0.1182755 0.1320064 0.1341726 0.1347674 0.1377288 0.1460630    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1512740 0.1679135 0.1708299 0.1703785 0.1736246 0.1845862    0
## GLM  0.1512740 0.1679135 0.1708299 0.1703785 0.1736246 0.1845862    0
## SVM  0.1458013 0.1626592 0.1656994 0.1655638 0.1690097 0.1818950    0
## CART 0.1484250 0.1683059 0.1715341 0.1716644 0.1746056 0.1896656    0
```

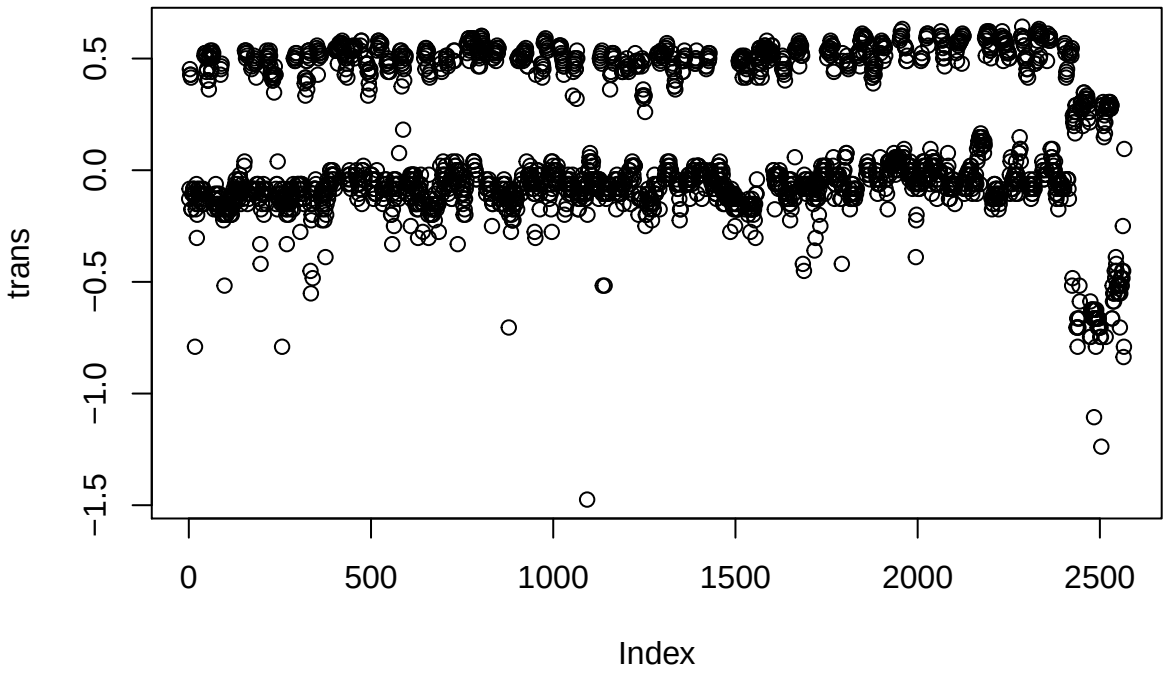
```
## KNN  0.1458705 0.1649942 0.1677449 0.1682364 0.1712889 0.1860165    0
##
## Rsquared
##               Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## LM   1.059423e-07 0.01450212 0.02860950 0.02885432 0.04033276 0.07312242
## GLM  1.059423e-07 0.01450212 0.02860950 0.02885432 0.04033276 0.07312242
## SVM  4.199900e-02 0.05874813 0.08254345 0.08480841 0.10415913 0.15600255
## CART 1.597245e-02 0.04179563 0.05758382 0.06559880 0.09364391 0.13026163
## KNN  3.103466e-02 0.05671126 0.07340516 0.07805263 0.10445217 0.13925275
##      NA's
## LM      0
## GLM     0
## SVM     0
## CART    0
## KNN     0
```

Next, we'll work with the Density variable.

```
# look at a couple of the variables first
lambda <- BoxCox.lambda(jeffsVars.pred$Density)
trans <- BoxCox(jeffsVars.pred$Density,lambda)
  m <- mean(trans)
  s <- sd(trans)
  hist(trans,freq=FALSE,main = "Density",xlab="")
  curve(dnorm(x,mean=m,sd=s),col="darkblue",lwd=2,add=TRUE)
```




```
  plot(trans)
```




```
jeffsVars.imp <- cbind(jeffsVars.imp,Density.Trans=trans)
```

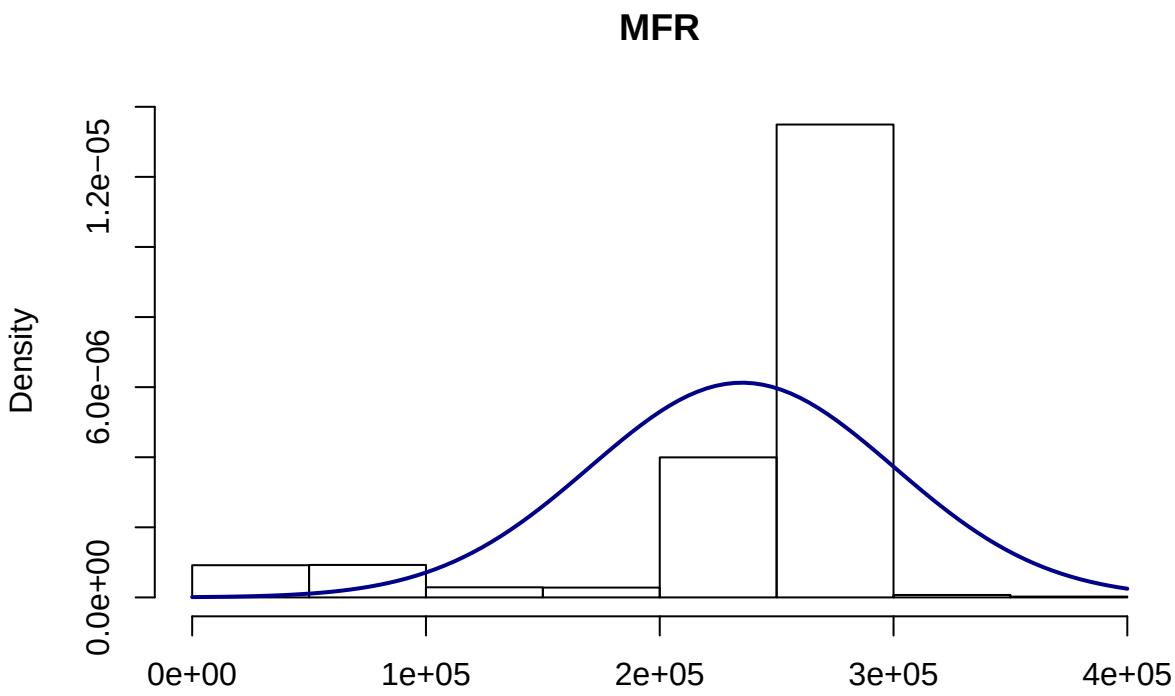
```
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"

# LM
set.seed(624)
fit.lm <- train(PH~Density.Trans, data=jeffsVars.imp, method="lm", metric=metric,
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
fit.glm <- train(PH~Density.Trans, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~Density.Trans, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~Density.Trans, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
fit.knn <- train(PH~Density.Trans, data=jeffsVars.imp, method="knn", metric=metric,
```

```
                preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

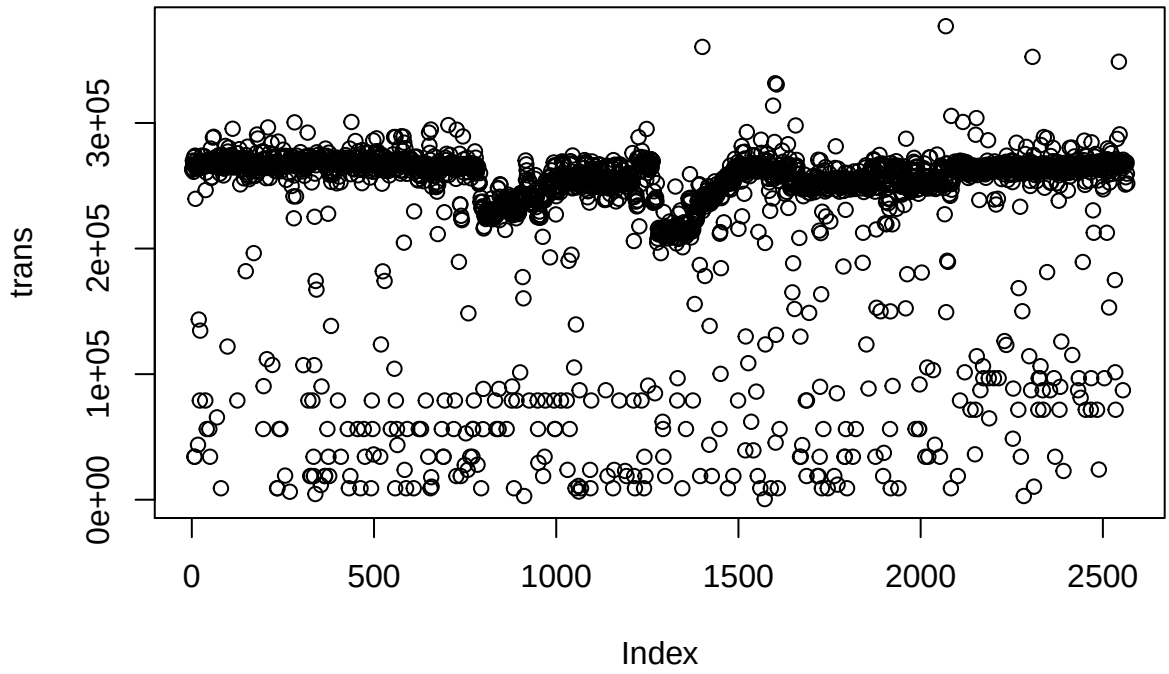
```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1259148 0.1366626 0.1375890 0.1381896 0.1410775 0.1441072    0
## GLM  0.1259148 0.1366626 0.1375890 0.1381896 0.1410775 0.1441072    0
## SVM  0.1206728 0.1294410 0.1328013 0.1324740 0.1355260 0.1388555    0
## CART 0.1222047 0.1296428 0.1328848 0.1323287 0.1351617 0.1376902    0
## KNN  0.1221373 0.1296131 0.1325631 0.1321520 0.1350807 0.1382186    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1543272 0.1691211 0.1718563 0.1720389 0.1768771 0.1830566    0
## GLM  0.1543272 0.1691211 0.1718563 0.1720389 0.1768771 0.1830566    0
## SVM  0.1512251 0.1637221 0.1664892 0.1671117 0.1719010 0.1783252    0
## CART 0.1515065 0.1622659 0.1675605 0.1669726 0.1717553 0.1759668    0
## KNN  0.1508530 0.1625653 0.1671192 0.1666046 0.1715170 0.1757176    0
##
## Rsquared
##              Min.     1st Qu.      Median        Mean    3rd Qu.
## LM   5.830486e-06 0.001985787 0.008314255 0.008010528 0.01125320
## GLM  5.830486e-06 0.001985787 0.008314255 0.008010528 0.01125320
## SVM  2.275359e-02 0.049529010 0.059899344 0.065049448 0.07770287
## CART 1.912344e-02 0.052927012 0.066619049 0.070330750 0.08712817
## KNN  2.675947e-02 0.058805891 0.068280337 0.072360521 0.08987026
##           Max. NA's
## LM   0.02306657    0
## GLM  0.02306657    0
## SVM  0.16028619    0
## CART 0.13153455    0
## KNN  0.13321256    0
```

Next, we'll work with the MFR variable.

```
# look at a couple of the variables first
lambda <- BoxCox.lambda(jeffsVars.pred$MFR)
trans <- BoxCox(jeffsVars.pred$MFR,lambda)
  m <- mean(trans)
  s <- sd(trans)
  hist(trans,freq=FALSE,main = "MFR",xlab="")
  curve(dnorm(x,mean=m,sd=s),col="darkblue",lwd=2,add=TRUE)
```




```
plot(trans)
```




```
jeffsVars.imp <- cbind(jeffsVars.imp,MFR.Trans=trans)
```

```
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"

# LM
set.seed(624)
```

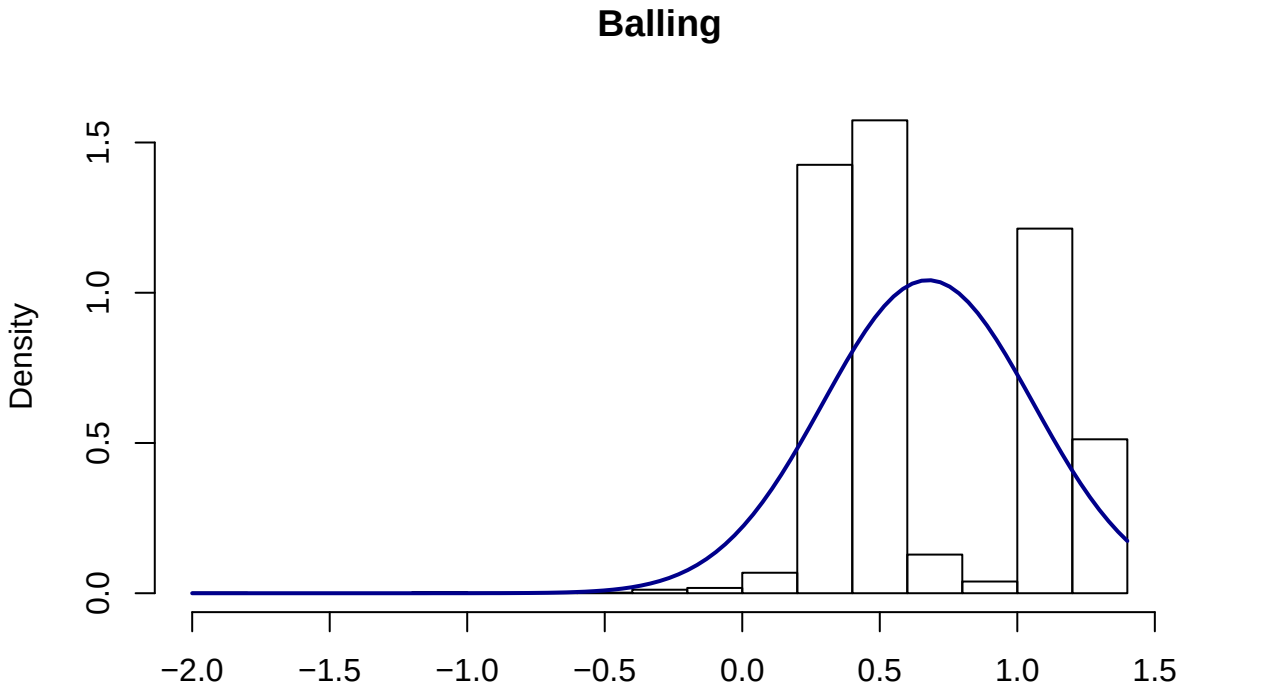
```
fit.lm <- train(PH~MFR.Trans, data=jeffsVars.imp, method="lm", metric=metric,
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
fit.glm <- train(PH~MFR.Trans, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~MFR.Trans, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~MFR.Trans, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
fit.knn <- train(PH~MFR.Trans, data=jeffsVars.imp, method="knn", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1271927 0.1374235 0.1387372 0.1388281 0.1417470 0.1451214    0
## GLM  0.1271927 0.1374235 0.1387372 0.1388281 0.1417470 0.1451214    0
## SVM  0.1267783 0.1366648 0.1381861 0.1384726 0.1414777 0.1462941    0
## CART 0.1272557 0.1372381 0.1387320 0.1387509 0.1417459 0.1451763    0
## KNN  0.1257882 0.1360143 0.1409684 0.1400099 0.1436910 0.1506375    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1547053 0.1695260 0.1714954 0.1724423 0.1774725 0.1842028    0
## GLM  0.1547053 0.1695260 0.1714954 0.1724423 0.1774725 0.1842028    0
## SVM  0.1545132 0.1696239 0.1720508 0.1723559 0.1772828 0.1842605    0
```

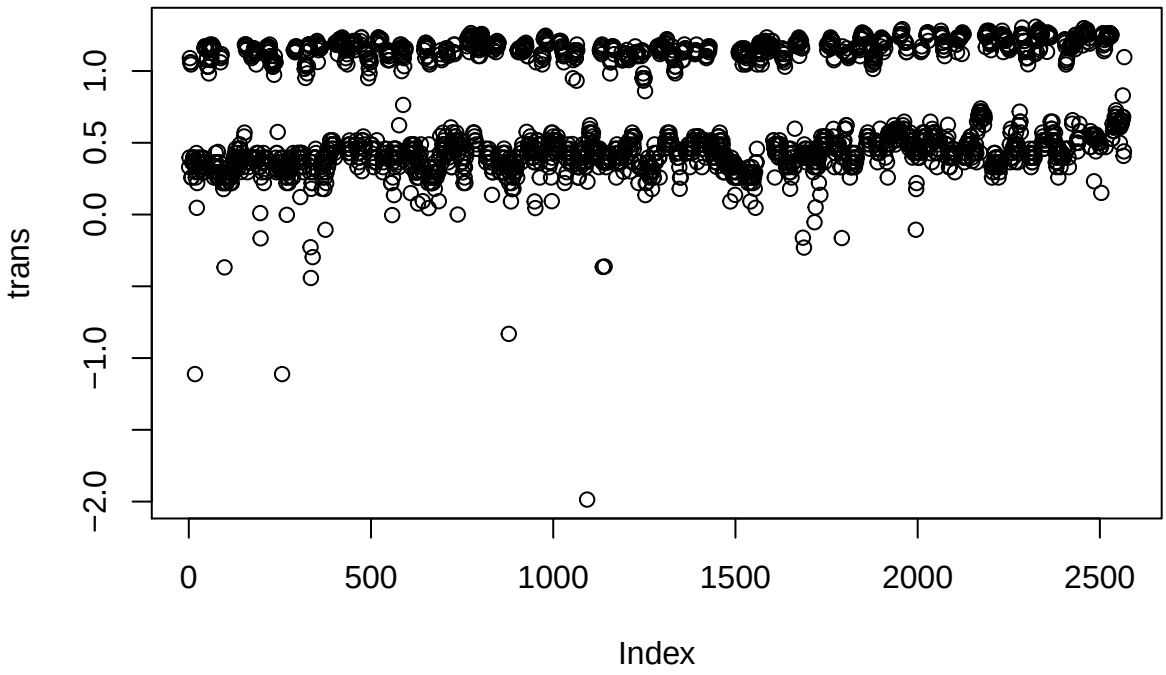
```
## CART 0.1548066 0.1694532 0.1715767 0.1723899 0.1774563 0.1842563    0
## KNN  0.1516632 0.1709094 0.1769157 0.1752070 0.1799176 0.1948919    0
##
## Rsquared
##              Min.      1st Qu.      Median        Mean     3rd Qu.
## LM   1.985897e-07 0.0002442781 0.002253521 0.003767926 0.005522049
## GLM  1.985897e-07 0.0002442781 0.002253521 0.003767926 0.005522049
## SVM  4.045557e-05 0.0010331188 0.002637306 0.006183408 0.007490397
## CART           NA           NA          NA         NaN          NA
## KNN  4.183698e-05 0.0092037323 0.017492110 0.018863525 0.023865447
##            Max. NA's
## LM   0.01595261    0
## GLM  0.01595261    0
## SVM  0.03558619    0
## CART         NA   30
## KNN  0.06038659    0
```

Next, we'll work with the Bslling variable.

```
# look at a couple of the variables first
lambda <- BoxCox.lambda(jeffsVars.pred$Balling)
trans <- BoxCox(jeffsVars.pred$Balling,lambda)
  m <- mean(trans)
  s <- sd(trans)
  hist(trans,freq=FALSE,main = "Balling",xlab="")
  curve(dnorm(x,mean=m,sd=s),col="darkblue",lwd=2,add=TRUE)
```




```
  plot(trans)
```




```
jeffsVars.imp <- cbind(jeffsVars.imp,Balling.Trans=trans)
```

```
# Run algorithms using 10-fold cross-validation
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "RMSE"

# LM
set.seed(624)
fit.lm <- train(PH~Balling.Trans, data=jeffsVars.imp, method="lm", metric=metric,
                preProc=c("center", "scale"), trControl=trainControl)

# GLM
set.seed(624)
fit.glm <- train(PH~Balling.Trans, data=jeffsVars.imp, method="glm", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)
# GLMNET
set.seed(624)
#fit.glmnet <- train(PH~Filler.Speed, data=jeffsVars.imp, method="glmnet", metric=metric,
#                    preProc=c("center", "scale"), trControl=trainControl)
# SVM
set.seed(624)
fit.svm <- train(PH~Balling.Trans, data=jeffsVars.imp, method="svmRadial", metric=metric,
                 preProc=c("center", "scale"), trControl=trainControl)

# CART
set.seed(624)
grid <- expand.grid(.cp=c(0, 0.05, 0.1))
fit.cart <- train(PH~Balling.Trans, data=jeffsVars.imp, method="rpart", metric=metric,
                  tuneGrid=grid, preProc=c("center", "scale"),
                  trControl=trainControl)
# KNN
set.seed(624)
fit.knn <- train(PH~Balling.Trans, data=jeffsVars.imp, method="knn", metric=metric,
```

```
                preProc=c("center", "scale"), trControl=trainControl)

# Compare algorithms
#feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm, GLMNET=fit.glmnet,
#                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
feature_results <- resamples(list(LM=fit.lm, GLM=fit.glm,
                                  SVM=fit.svm, CART=fit.cart, KNN=fit.knn))
summary(feature_results)
```

```
##
## Call:
## summary.resamples(object = feature_results)
##
## Models: LM, GLM, SVM, CART, KNN
## Number of resamples: 30
##
## MAE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1266276 0.1369666 0.1382339 0.1385185 0.1415090 0.1447364    0
## GLM  0.1266276 0.1369666 0.1382339 0.1385185 0.1415090 0.1447364    0
## SVM  0.1210123 0.1306028 0.1331553 0.1323495 0.1350785 0.1387711    0
## CART 0.1210872 0.1284054 0.1324030 0.1314425 0.1351302 0.1364562    0
## KNN  0.1208905 0.1284944 0.1320758 0.1307691 0.1340340 0.1364024    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LM   0.1547101 0.1693703 0.1716753 0.1724054 0.1772578 0.1840733    0
## GLM  0.1547101 0.1693703 0.1716753 0.1724054 0.1772578 0.1840733    0
## SVM  0.1500447 0.1632652 0.1655961 0.1659218 0.1715056 0.1769712    0
## CART 0.1492675 0.1621286 0.1660909 0.1660337 0.1721116 0.1774060    0
## KNN  0.1495694 0.1621816 0.1646886 0.1651819 0.1709270 0.1756947    0
##
## Rsquared
##              Min.     1st Qu.      Median        Mean     3rd Qu.
## LM   6.241192e-06 0.000790578 0.002662098 0.004693808 0.006526726
## GLM  6.241192e-06 0.000790578 0.002662098 0.004693808 0.006526726
## SVM  3.136045e-02 0.060732038 0.069267516 0.077273276 0.090490935
## CART 3.125766e-02 0.061486740 0.083684605 0.085903689 0.102055374
## KNN  3.571639e-02 0.066262307 0.084046226 0.092334003 0.118251132
##           Max. NA's
## LM   0.02629823    0
## GLM  0.02629823    0
## SVM  0.15194152    0
## CART 0.15520604    0
## KNN  0.15846199    0
```

The 1-on-1 modeling results have been so poor that we're going to try a modeling experiment using all variables in the set (in their transformed state if applicable).

```
# generate a generalize linear model with all variables
whole.model <- glm(PH ~ Filler.Level+Filler.Speed.Trans+Temperature.Trans+Usage.cont.Trans+Carb.Flow.Tra
stepwise <- step(whole.model, direction = "both")
```

```
## Start:  AIC=-2344.26
## PH ~ Filler.Level + Filler.Speed.Trans + Temperature.Trans +
```

```
##     Usage.cont.Trans + Carb.Flow.Trans + Density.Trans + MFR.Trans +
##     Balling.Trans
##
##                       Df Deviance     AIC
## - Filler.Speed.Trans  1   59.845 -2345.9
## - MFR.Trans           1   59.845 -2345.9
## <none>                    59.835 -2344.3
## - Balling.Trans       1   59.937 -2341.9
## - Density.Trans       1   60.001 -2339.1
## - Carb.Flow.Trans     1   60.316 -2325.7
## - Usage.cont.Trans    1   62.010 -2254.6
## - Temperature.Trans   1   63.351 -2199.7
## - Filler.Level        1   64.393 -2157.8
##
## Step:  AIC=-2345.86
## PH ~ Filler.Level + Temperature.Trans + Usage.cont.Trans + Carb.Flow.Trans +
##     Density.Trans + MFR.Trans + Balling.Trans
##
##                       Df Deviance     AIC
## <none>                    59.845 -2345.9
## + Filler.Speed.Trans  1   59.835 -2344.3
## - Balling.Trans       1   59.947 -2343.5
## - Density.Trans       1   60.012 -2340.7
## - Carb.Flow.Trans     1   60.317 -2327.7
## - MFR.Trans           1   60.357 -2326.0
## - Usage.cont.Trans    1   62.060 -2254.6
## - Temperature.Trans   1   63.354 -2201.6
## - Filler.Level        1   64.402 -2159.4
```

```
stepwise
```

```
##
## Call:  glm(formula = PH ~ Filler.Level + Temperature.Trans + Usage.cont.Trans +
##     Carb.Flow.Trans + Density.Trans + MFR.Trans + Balling.Trans,
##     family = gaussian(link = "identity"), data = jeffsVars.imp)
##
## Coefficients:
##       (Intercept)       Filler.Level  Temperature.Trans
##         1.406e+02          2.961e-03         -1.342e+02
##  Usage.cont.Trans    Carb.Flow.Trans      Density.Trans
##        -5.940e-04          7.868e-09          6.943e-02
##         MFR.Trans      Balling.Trans
##        -2.484e-07         -4.538e-02
##
## Degrees of Freedom: 2566 Total (i.e. Null);  2559 Residual
## Null Deviance:       76.37
## Residual Deviance: 59.84     AIC: -2346
```