# Data624 - Project2

Amanda Arce, Jatin Jain, Amit Kapoor

5/2/2021

## Contents



## Overview

ABC Beverage has new regulations in place and the leadership team requires the data scientists team to understand the manufacturing process, the predictive factors and be able to report to them predictive model of PH. The selection of model depends upon various factors like model accuracy, data relevance, cross validation etc.

# R packages

We will use `r` for data modeling. All packages used for data exploration, visualization, preparation and modeling are listed in Code Appendix.

# Data Exploration

We will first get the historical dataset, provided in excel and use it to analyze and eventually predict the PH of beverages.

## Data summary

There are 31 predictor variables that are numeric and 1 predictor variable `Brand Code` which is factor. The training dataset has 2,571 observations.

```
## Rows: 2,571
## Columns: 33
## $ `Brand Code`        <fct> B, A, B, A, A, A, A, B, B, B, B, B, B, B, B, C,~
## $ `Carb Volume`       <dbl> 5.340000, 5.426667, 5.286667, 5.440000, 5.486667, ~
## $ `Fill Ounces`       <dbl> 23.96667, 24.00667, 24.06000, 24.00667, 24.31333, ~
## $ `PC Volume`         <dbl> 0.2633333, 0.2386667, 0.2633333, 0.2933333, 0.1113~
## $ `Carb Pressure`     <dbl> 68.2, 68.4, 70.8, 63.0, 67.2, 66.6, 64.2, 67.6, 64~
## $ `Carb Temp`         <dbl> 141.2, 139.6, 144.8, 132.6, 136.8, 138.4, 136.8, 1~
## $ PSC                 <dbl> 0.104, 0.124, 0.090, NA, 0.026, 0.090, 0.128, 0.15~
## $ `PSC Fill`          <dbl> 0.26, 0.22, 0.34, 0.42, 0.16, 0.24, 0.40, 0.34, 0.~
## $ `PSC CO2`           <dbl> 0.04, 0.04, 0.16, 0.04, 0.12, 0.04, 0.04, 0.04, 0.~
## $ `Mnf Flow`          <dbl> -100, -100, -100, -100, -100, -100, -100, -100, -1~
## $ `Carb Pressure1`    <dbl> 118.8, 121.6, 120.2, 115.2, 118.4, 119.6, 122.2, 1~
## $ `Fill Pressure`     <dbl> 46.0, 46.0, 46.0, 46.4, 45.8, 45.6, 51.8, 46.8, 46~
## $ `Hyd Pressure1`     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ `Hyd Pressure2`     <dbl> NA, NA, NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ `Hyd Pressure3`     <dbl> NA, NA, NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ `Hyd Pressure4`     <dbl> 118, 106, 82, 92, 92, 116, 124, 132, 90, 108, 94, ~
## $ `Filler Level`      <dbl> 121.2, 118.6, 120.0, 117.8, 118.6, 120.2, 123.4, 1~
## $ `Filler Speed`      <dbl> 4002, 3986, 4020, 4012, 4010, 4014, NA, 1004, 4014~
## $ Temperature         <dbl> 66.0, 67.6, 67.0, 65.6, 65.6, 66.2, 65.8, 65.2, 65~
## $ `Usage cont`        <dbl> 16.18, 19.90, 17.76, 17.42, 17.68, 23.82, 20.74, 1~
## $ `Carb Flow`         <dbl> 2932, 3144, 2914, 3062, 3054, 2948, 30, 684, 2902,~
## $ Density             <dbl> 0.88, 0.92, 1.58, 1.54, 1.54, 1.52, 0.84, 0.84, 0.~
## $ MFR                 <dbl> 725.0, 726.8, 735.0, 730.6, 722.8, 738.8, NA, NA, ~
## $ Balling             <dbl> 1.398, 1.498, 3.142, 3.042, 3.042, 2.992, 1.298, 1~
## $ `Pressure Vacuum`   <dbl> -4.0, -4.0, -3.8, -4.4, -4.4, -4.4, -4.4, -4.4, -4~
## $ PH                  <dbl> 8.36, 8.26, 8.94, 8.24, 8.26, 8.32, 8.40, 8.38, 8.~
## $ `Oxygen Filler`     <dbl> 0.022, 0.026, 0.024, 0.030, 0.030, 0.024, 0.066, 0~
## $ `Bowl Setpoint`     <dbl> 120, 120, 120, 120, 120, 120, 120, 120, 120, 120, ~
## $ `Pressure Setpoint` <dbl> 46.4, 46.8, 46.6, 46.0, 46.0, 46.0, 46.0, 46.0, 46~
## $ `Air Pressurer`     <dbl> 142.6, 143.0, 142.0, 146.2, 146.2, 146.6, 146.2, 1~
## $ `Alch Rel`          <dbl> 6.58, 6.56, 7.66, 7.14, 7.14, 7.16, 6.54, 6.52, 6.~
## $ `Carb Rel`          <dbl> 5.32, 5.30, 5.84, 5.42, 5.44, 5.44, 5.38, 5.34, 5.~
## $ `Balling Lvl`       <dbl> 1.48, 1.56, 3.28, 3.04, 3.04, 3.02, 1.44, 1.44, 1.~
```

```
##                      n    mean      sd  median     min     max   range  skew
## Brand Code*       2451    2.51    1.00    2.00    1.00    4.00    3.00  0.38
## Carb Volume       2561    5.37    0.11    5.35    5.04    5.70    0.66  0.39
## Fill Ounces       2533   23.97    0.09   23.97   23.63   24.32    0.69 -0.02
```

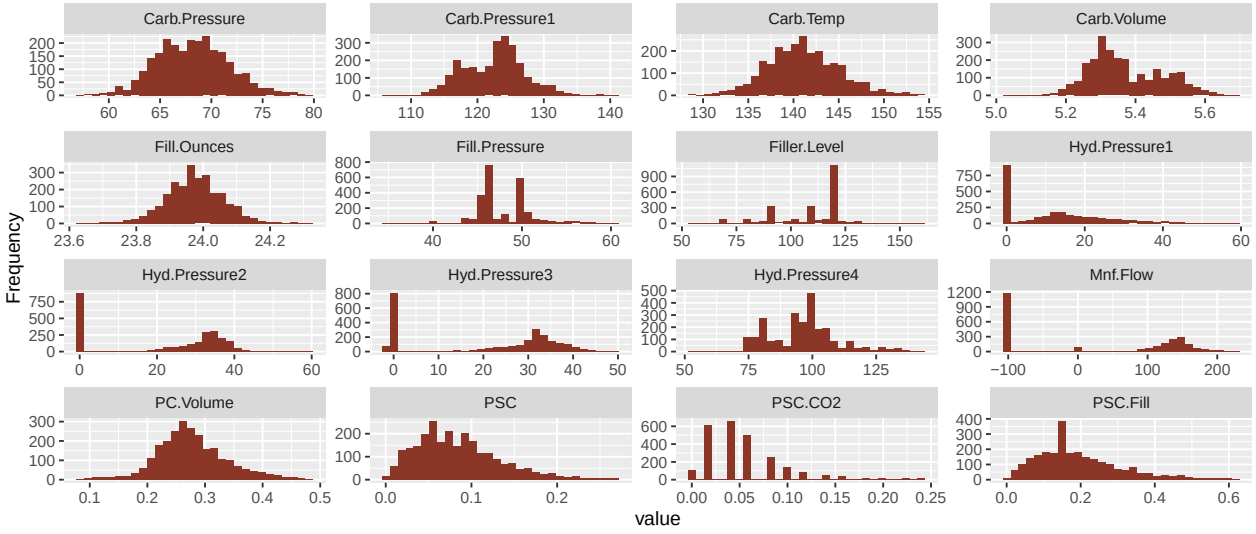
```
## PC Volume         2532    0.28    0.06    0.27    0.08    0.48    0.40  0.34
## Carb Pressure     2544   68.19    3.54   68.20   57.00   79.40   22.40  0.18
## Carb Temp         2545  141.09    4.04  140.80  128.60  154.00   25.40  0.25
## PSC               2538    0.08    0.05    0.08    0.00    0.27    0.27  0.85
## PSC Fill          2548    0.20    0.12    0.18    0.00    0.62    0.62  0.93
## PSC CO2           2532    0.06    0.04    0.04    0.00    0.24    0.24  1.73
## Mnf Flow          2569   24.57  119.48   65.20 -100.20  229.40  329.60  0.00
## Carb Pressure1    2539  122.59    4.74  123.20  105.60  140.20   34.60  0.05
## Fill Pressure     2549   47.92    3.18   46.40   34.60   60.40   25.80  0.55
## Hyd Pressure1     2560   12.44   12.43   11.40   -0.80   58.00   58.80  0.78
## Hyd Pressure2     2556   20.96   16.39   28.60    0.00   59.40   59.40 -0.30
## Hyd Pressure3     2556   20.46   15.98   27.60   -1.20   50.00   51.20 -0.32
## Hyd Pressure4     2541   96.29   13.12   96.00   52.00  142.00   90.00  0.55
## Filler Level      2551  109.25   15.70  118.40   55.80  161.20  105.40 -0.85
## Filler Speed      2514 3687.20  770.82 3982.00  998.00 4030.00 3032.00 -2.87
## Temperature       2557   65.97    1.38   65.60   63.60   76.20   12.60  2.39
## Usage cont        2566   20.99    2.98   21.79   12.08   25.90   13.82 -0.54
## Carb Flow         2569 2468.35 1073.70 3028.00   26.00 5104.00 5078.00 -0.99
## Density           2570    1.17    0.38    0.98    0.24    1.92    1.68  0.53
## MFR               2359  704.05   73.90  724.00   31.40  868.60  837.20 -5.09
## Balling           2570    2.20    0.93    1.65   -0.17    4.01    4.18  0.59
## Pressure Vacuum   2571   -5.22    0.57   -5.40   -6.60   -3.60    3.00  0.53
## PH                2567    8.55    0.17    8.54    7.88    9.36    1.48 -0.29
## Oxygen Filler     2559    0.05    0.05    0.03    0.00    0.40    0.40  2.66
## Bowl Setpoint     2569  109.33   15.30  120.00   70.00  140.00   70.00 -0.97
## Pressure Setpoint 2559   47.62    2.04   46.00   44.00   52.00    8.00  0.20
## Air Pressurer     2571  142.83    1.21  142.60  140.80  148.20    7.40  2.25
## Alch Rel          2562    6.90    0.51    6.56    5.28    8.62    3.34  0.88
## Carb Rel          2561    5.44    0.13    5.40    4.96    6.06    1.10  0.50
## Balling Lvl       2570    2.05    0.87    1.48    0.00    3.66    3.66  0.59
##                   kurtosis
## Brand Code*          -1.06
## Carb Volume          -0.47
## Fill Ounces           0.86
## PC Volume             0.67
## Carb Pressure        -0.01
## Carb Temp             0.24
## PSC                   0.65
## PSC Fill              0.77
## PSC CO2               3.73
## Mnf Flow             -1.87
## Carb Pressure1        0.14
## Fill Pressure         1.41
## Hyd Pressure1        -0.14
## Hyd Pressure2        -1.56
## Hyd Pressure3        -1.57
## Hyd Pressure4         0.63
## Filler Level          0.05
## Filler Speed          6.71
## Temperature          10.16
## Usage cont           -1.02
## Carb Flow            -0.58
## Density              -1.20
## MFR                  30.46
```

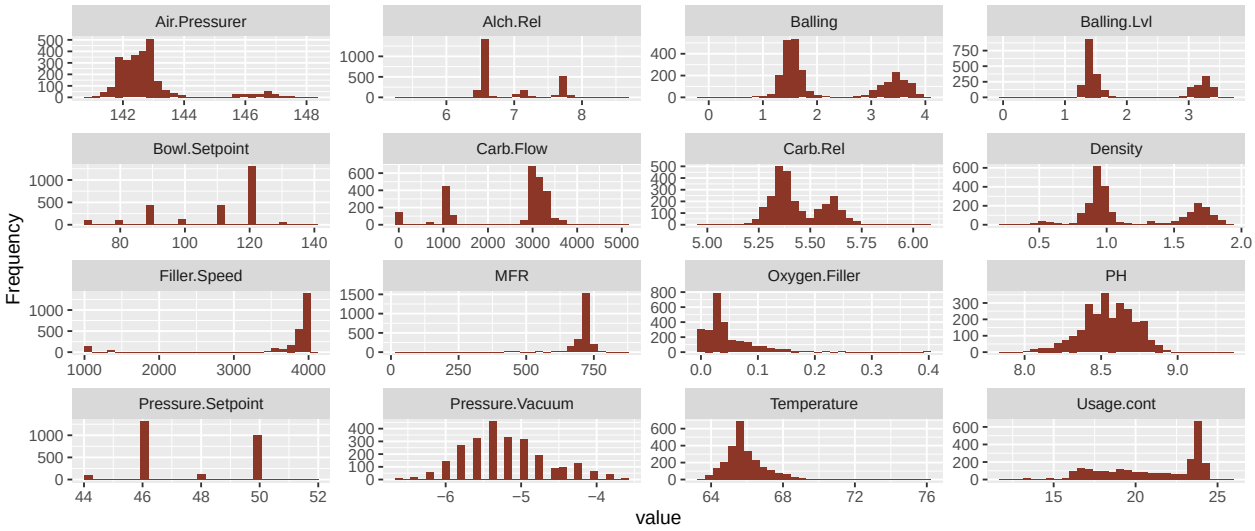
```
## Balling              -1.39
## Pressure Vacuum      -0.03
## PH                    0.06
## Oxygen Filler        11.09
## Bowl Setpoint        -0.06
## Pressure Setpoint    -1.60
## Air Pressurer         4.73
## Alch Rel             -0.85
## Carb Rel             -0.29
## Balling Lvl          -1.49
```

Based of above description, we can see the dataset has missing values so it would need imputation. The predictors `Oxygen Filler`, `MFR`, `Filler Speed` and `Temperature` seems highly skewed and would require transformation. This could be seen in below histogram plots as well.

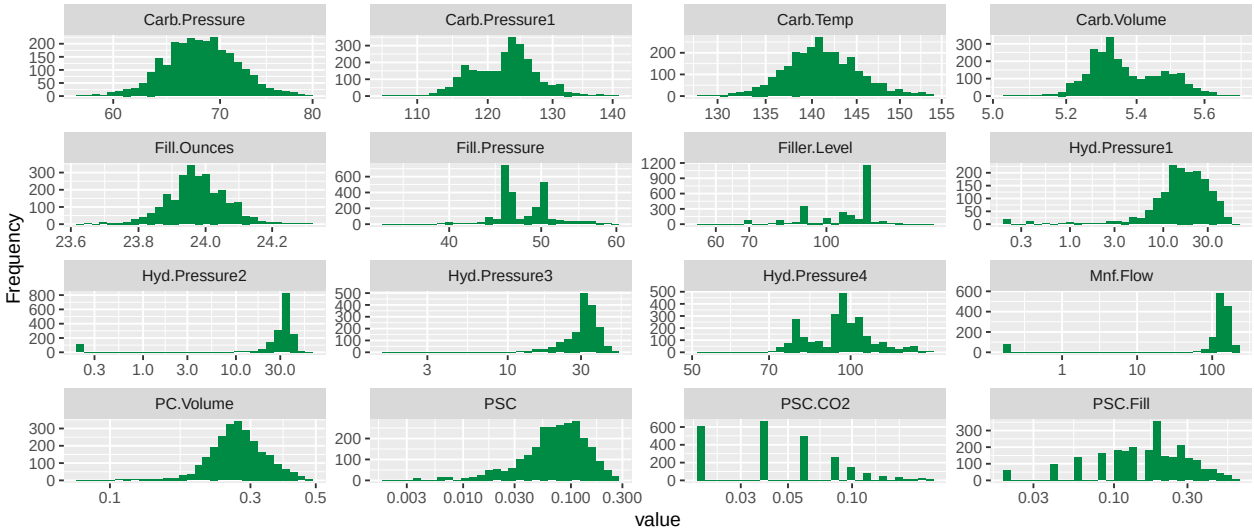
## Variables Distribution

Below we have shown the distribution of dataset variables. There are 2 sets of histograms; the one in red is natural distribution and the ones in green are logarithmic disctribution

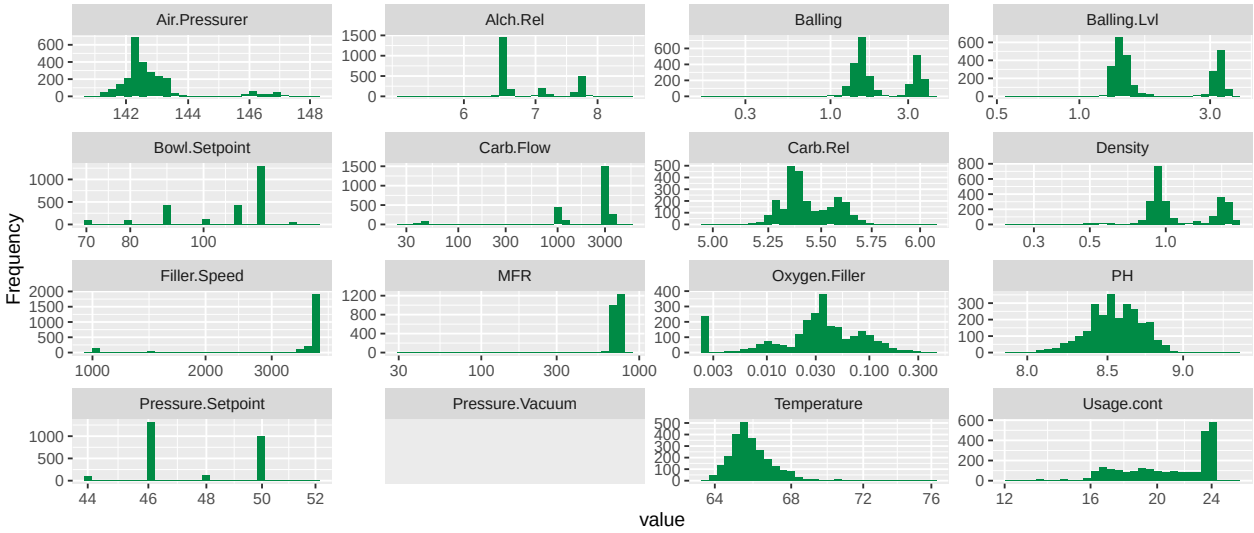





Page 2




Page 1

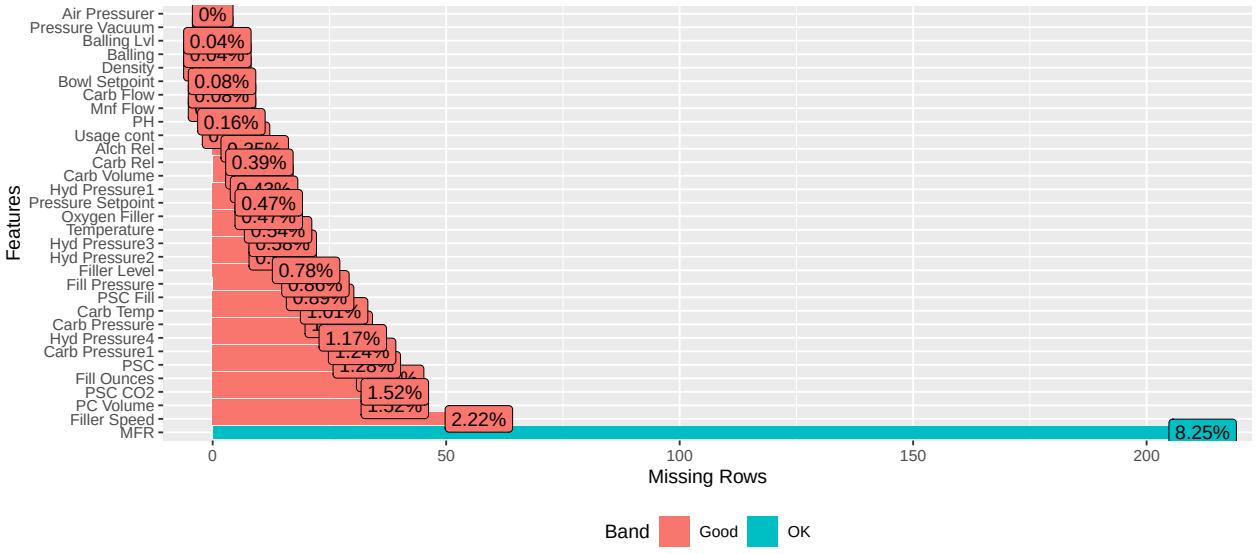



Page 2

## Missing Data

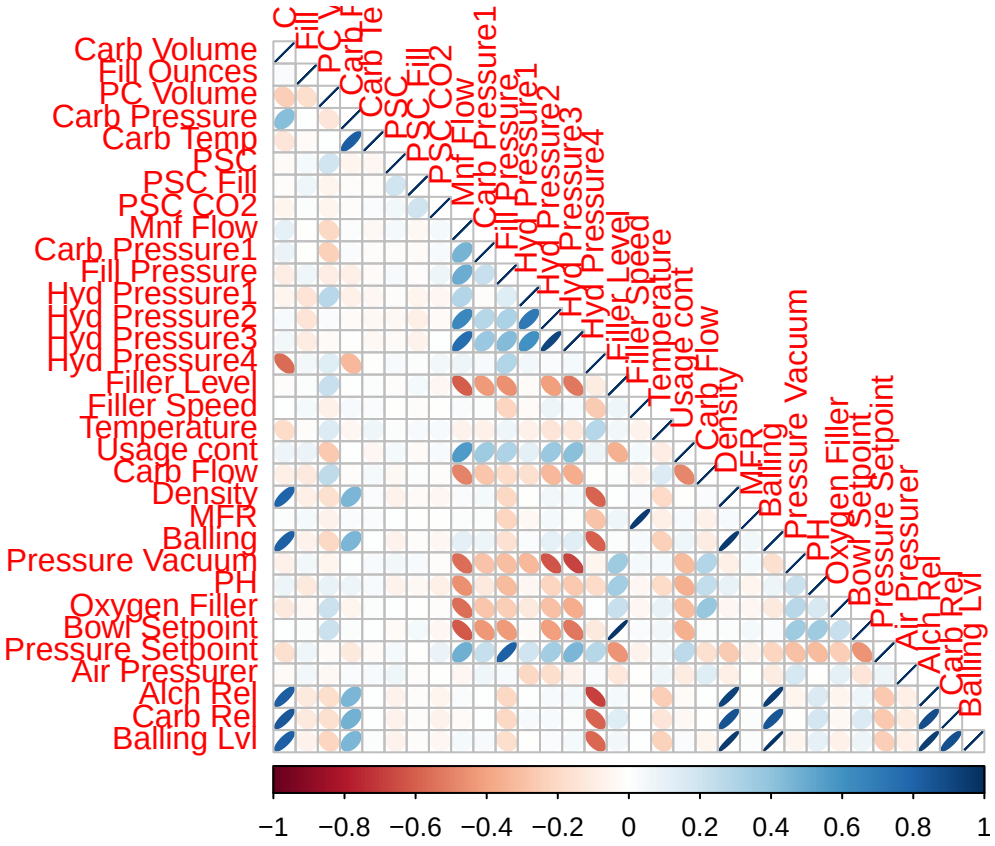
The summary and following graphs show the missing data in training dataset. The plot below shows more than 8% data is missing for `MFR` variable. Next feature that has missing data is `Filler Speed` which shows more than 2% missing data. The missing data will be handled through imputation.

```
##        Brand Code       Carb Volume       Fill Ounces        PC Volume
##               120                10                38               39
##     Carb Pressure         Carb Temp               PSC         PSC Fill
##               27                 26                33               23
##           PSC CO2         Mnf Flow    Carb Pressure1     Fill Pressure
##               39                 2                 32               22
##     Hyd Pressure1     Hyd Pressure2     Hyd Pressure3     Hyd Pressure4
##               11                 15                15               30
##      Filler Level      Filler Speed       Temperature       Usage cont
##               20                 57                14                5
##         Carb Flow           Density               MFR          Balling
##                2                  1               212                1
##   Pressure Vacuum                 PH     Oxygen Filler     Bowl Setpoint
##                0                  4                 12                2
## Pressure Setpoint     Air Pressurer          Alch Rel         Carb Rel
##               12                 0                 9                10
##       Balling Lvl
##                1
```



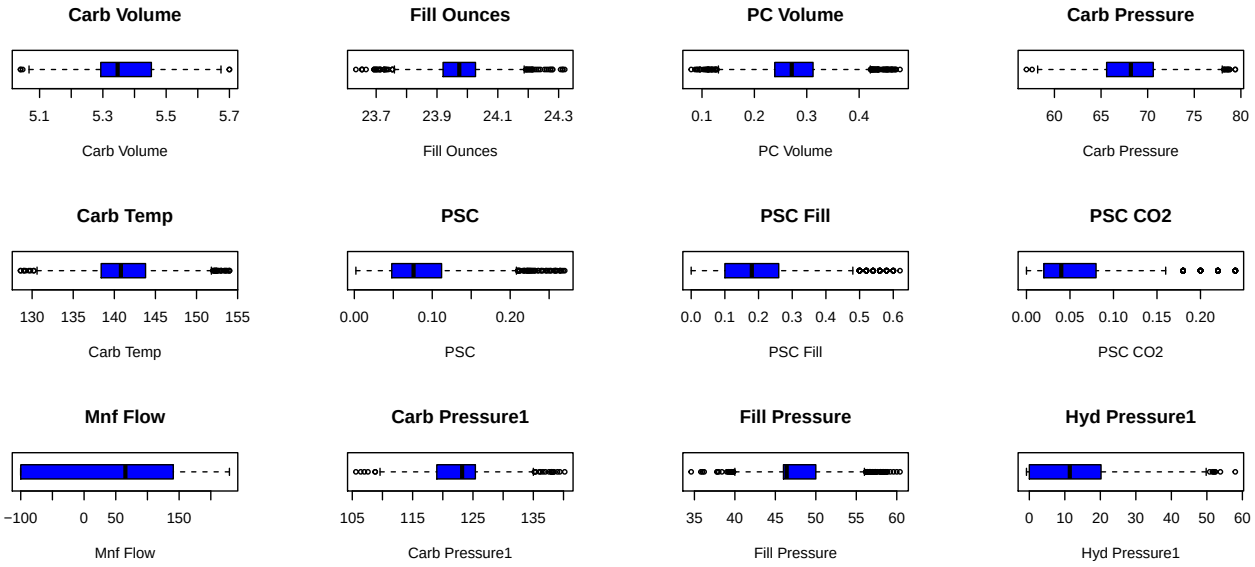

## Correlation

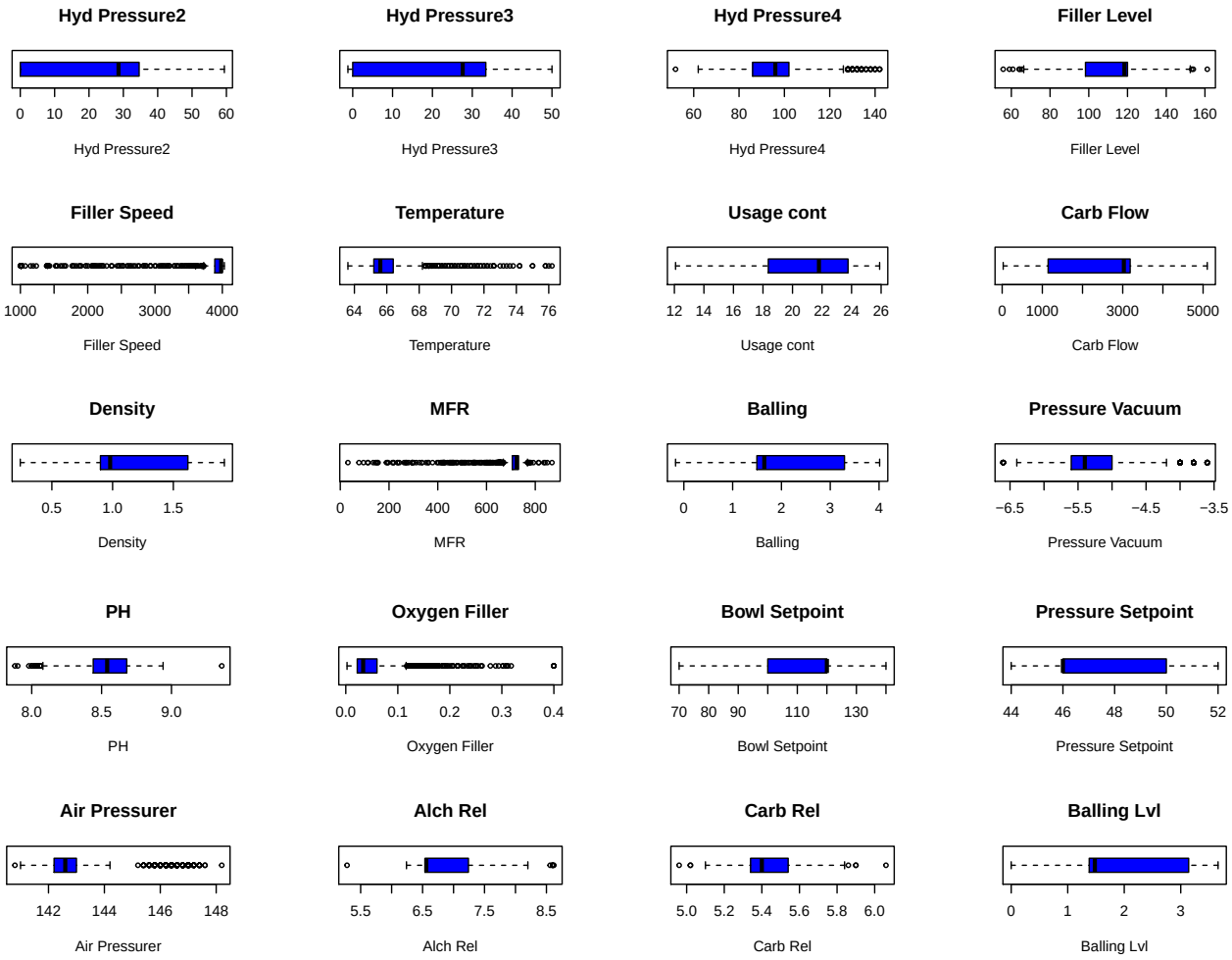
below plot shows the correlation among numeric variables in the dataset. We can see few variables are highly correlated. We will handle the pairwise predictors that has correlation above 0.90 in data preparation section.




## Outliers

In this section we will check the outliers in the data. An outlier is an observation that lies an unusual distance from other values in a random sample.These outlier could impact predictions so will be handled through imputation






# Data Preparation

## Handling missing and outliers

The very first in data preparation we will perform is handling missing data and outliers through imputation. We will use mice package to perform imputation here. MICE (Multivariate Imputation via Chained Equations) is one of the commonly used package for this activity. It creates multiple imputations for multivariate missing data. Also we will perform `nearZeroVar` to see if a variable has very little change or variation and not useful for prediction. If we founf any predictor variable satisfying this condition we would remove it.

## Create Dummy Variables

The variable Brand Code is a categorical variable, having 4 classes (A, B, C, and D). For modeling, we got to convert into set of dummy variables. We will use `dummyVars` function for this purpose that creates a full set of dummy variables.

## Correlation

Next step is to remove highly correlated predictor variables. we will use the cutoff as 0.90 here.

## Preprocess using transformation

In this step we will use caret `preprocess` method using transformation as `YeoJohnson` which applies Yeo-Johnson transformation, like a BoxCox, but values can be negative as well.

```
##  [1] "Brand.Code.A"      "Brand.Code.B"      "Brand.Code.C"
##  [4] "Brand.Code.D"      "Carb.Volume"       "Fill.Ounces"
##  [7] "PC.Volume"         "Carb.Pressure"     "Carb.Temp"
## [10] "PSC"               "PSC.Fill"          "PSC.CO2"
## [13] "Mnf.Flow"          "Carb.Pressure1"    "Fill.Pressure"
## [16] "Hyd.Pressure1"     "Hyd.Pressure2"     "Hyd.Pressure3"
## [19] "Hyd.Pressure4"     "Filler.Level"      "Filler.Speed"
## [22] "Temperature"       "Usage.cont"        "Carb.Flow"
## [25] "Density"           "MFR"               "Balling"
## [28] "Pressure.Vacuum"   "PH"                "Oxygen.Filler"
## [31] "Bowl.Setpoint"     "Pressure.Setpoint" "Air.Pressurer"
## [34] "Alch.Rel"          "Carb.Rel"          "Balling.Lvl"
```

## Training and Test Partition

Finally in this step for data preparation we will partition the training dataset for training and validation using `createDataPartition` method from `caret` package. We will reserve 75% for training and rest 25% for validation purpose.

The variable Brand Code is a categorical variable, having 4 classes (A, B, C, and D). We opt to use the "one-hot" encoding scheme for this variable, creating 5 new variables for the data: BrandCodeA, BrandCodeB, BrandCodeC, BrandCodeD, and BrandCodeNA.

```
##    `Brand Code`A `Brand Code`B `Brand Code`C `Brand Code`D `Brand Code`NA
## 1              0             1             0             0              0
## 2              1             0             0             0              0
## 3              0             1             0             0              0
## 4              1             0             0             0              0
## 5              1             0             0             0              0
## 6              1             0             0             0              0
## 7              1             0             0             0              0
## 8              0             1             0             0              0
## 9              0             1             0             0              0
## 10             0             1             0             0              0
```

```
##  [1] B A B A A A A B B B
## Levels: A B C D <NA>
```

```
##    `Brand Code`A `Brand Code`B `Brand Code`C `Brand Code`D `Brand Code`NA
## 1              0             0             0             1              0
## 2              1             0             0             0              0
## 3              0             1             0             0              0
## 4              0             1             0             0              0
## 5              0             1             0             0              0
## 6              0             1             0             0              0
## 7              1             0             0             0              0
## 8              0             1             0             0              0
## 9              1             0             0             0              0
## 10             0             0             0             1              0
```

```
##  [1] D A B B B B A B A D
## Levels: A B C D <NA>
```

White spaces and special characters in the column names are removed so they does not cause issues in some of the R packages.

There are a few rows with target variable (PH) missing. These rows are removed, since they cannot be used for training.

There is one near-zero-variance variable in the data:

```
## [1] "BrandCodeNA"
```

Below, we remove the near-zero-variance predictor, and separate the predictors and target:

The train function from the caret package is used to tune the models. The 5-fold cross validation scheme is used to estimate the model performance based on their RMSE. Below, we create the folds and set up the train control:

For the missing values, we experiment with three different imputation algorithms provided in the preProcess function:

- KNN imputation
- Bagged trees imputation
- Median imputation

As will be seen in the "Linear Models" section below, the choice of imputation method does not seem to affect the prediction performance much. We opt to use the knnImpute method due to its high efficiency.

For the linear and non-linear models, the pre-processing step also include centering and scaling (standardizing), so that the variables all have a mean of 0 and standard deviation of 1. For the tree-based models, this step is omitted, since tree models work fine without this step.

The caret package supports parallel processing (multi-core training). This capability significantly lowers the training time:

# Build Models

# Select Model

# Prediction

# Conclusion

# References


- https://machinelearningmastery.com/pre-process-your-dataset-in-r/
- https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/
- Applied Predictive Modeling. Max Kuhn and Kjell Johnson


# Code Appendix

```
knitr::opts_chunk$set(echo=FALSE, error=FALSE, warning=FALSE, message=FALSE, fig.align="center", fig.wi
# Libraries
library(readxl)
library(tidyverse)
library(caret)
library(doParallel)
```

```
library(DataExplorer)
library(psych)
library(mice)

set.seed(624)
# download training data from git repo
temp.file <- tempfile(fileext = ".xlsx")
download.file(url="https://github.com/DATA624-PredictiveAnalytics-Project2/Project2/blob/main/StudentDa
              destfile = temp.file,
              mode = "wb",
              quiet = TRUE)

# read excel for training data
train.df <- read_excel(temp.file, skip=0)

# download testing data from git repo
download.file(url="https://github.com/DATA624-PredictiveAnalytics-Project2/Project2/blob/main/StudentEva
              destfile = temp.file,
              mode = "wb",
              quiet = TRUE)

# read excel for testing data
test.df <- read_excel(temp.file, skip=0)


# transform Brand.code to factor
train.df$`Brand Code` = as.factor(train.df$`Brand Code`)
test.df$`Brand Code` = as.factor(test.df$`Brand Code`)
glimpse(train.df)
describe(train.df) %>% dplyr::select(-vars, -trimmed, -mad, -se)
plot_histogram(train.df, geom_histogram_args = list("fill" = "tomato4"))
# log histograms
plot_histogram(train.df, scale_x = "log10", geom_histogram_args = list("fill" = "springgreen4"))
colSums(is.na(train.df))
plot_missing(train.df[-1])
forcorr <- train.df[complete.cases(train.df),-1]
corrplot::corrplot(cor(forcorr), method = 'ellipse', type = 'lower')

# boxplot
par(mfrow = c(3,4))
for(i in colnames(train.df[-1])){
boxplot(train.df[,i], xlab = names(train.df[i]),
  main = names(train.df[i]), col="blue", horizontal = T)
}
set.seed(317)

# Training set
train.df.clean <- mice(data.frame(train.df), method = 'rf', m=2, maxit = 2, print=FALSE)
train.df.clean <- complete(train.df.clean)

nzv_preds <- nearZeroVar(train.df.clean)
train.df.clean <- train.df.clean[,-nzv_preds]
set.seed(317)
```

```
# Testing set
test.df.clean <- mice(data.frame(test.df), method = 'rf', m=2, maxit = 2, print=FALSE)
test.df.clean <- complete(test.df.clean)
set.seed(317)
dum.brandcode <- dummyVars(PH ~ Brand.Code, data = train.df.clean)
dum.train.predict <- predict(dum.brandcode, train.df.clean)
train.df.clean <- cbind(dum.train.predict, train.df.clean) %>% dplyr::select(-Brand.Code)
set.seed(317)
dum.brandcode <- dummyVars( ~ Brand.Code, data = test.df.clean)
dum.test.predict <- predict(dum.brandcode, test.df.clean)
test.df.clean <- cbind(dum.test.predict, test.df.clean) %>% dplyr::select(-Brand.Code)
highCorr <- findCorrelation(cor(train.df.clean), 0.90)
train.df.clean <- train.df.clean[, -highCorr]
set.seed(317)
preproc_traindf <- preProcess(train.df.clean, method = "YeoJohnson")
train.df.clean <- predict(preproc_traindf, train.df.clean)
set.seed(317)
preproc_testdf <- preProcess(test.df.clean, method = "YeoJohnson")
test.df.clean <- predict(preproc_testdf, test.df.clean)
colnames(test.df.clean)
set.seed(317)

partition <- createDataPartition(train.df.clean$PH, p=0.75, list = FALSE)

# training/validation partition for independent variables
X.train <- train.df.clean[partition, ] %>% dplyr::select(-PH)
X.test <- train.df.clean[-partition, ] %>% dplyr::select(-PH)

# training/validation partition for dependent variable PH
y.train <- train.df.clean$PH[partition]
y.test <- train.df.clean$PH[-partition]
# One-hot encoding the categorical variable `Brand Code`

train.df$`Brand Code` <- addNA(train.df$`Brand Code`)
test.df$`Brand Code` <- addNA(test.df$`Brand Code`)
brandCodeTrain <- predict(dummyVars(~`Brand Code`, data=train.df), train.df)
brandCodeTest <- predict(dummyVars(~`Brand Code`, data=test.df), test.df)
head(brandCodeTrain, 10)
head(train.df$`Brand Code`, 10)
head(brandCodeTest, 10)
head(test.df$`Brand Code`, 10)
train <- cbind(brandCodeTrain, subset(train.df, select=-c(`Brand Code`)))
test <- cbind(brandCodeTest, subset(test.df, select=-c(`Brand Code`)))
# Remove special symbols (white space and `) in names

names(train) <- gsub(patter=c(' |`'), replacement='', names(train))
names(test) <- gsub(patter=c(' |`'), replacement='', names(test))
# Remove rows in training set with missing target variables
#train <- train[complete.cases(train$PH),]
train <- train %>% na.omit()
# Check near-zero-variance variables
nearZeroVar(train, names=T)
# Separate the predictors and target, and remove nzv variable
```

```
xTrain <- subset(train, select=-c(PH,`HydPressure1`)) %>% as.data.frame()
xTest <- subset(test, select=-c(PH,`HydPressure1`)) %>% as.data.frame()
yTrain <- train$PH
set.seed(1)
cvFolds <- createFolds(yTrain, k=5)
trControl <- trainControl(verboseIter=T,
                          method='cv',
                          number=5,
                          index=cvFolds)
# Set up and start multi-core processing
cl <- makePSOCKcluster(5)
registerDoParallel(cl)
library(MASS)
library(caret)
library(AppliedPredictiveModeling)
library(lars)
library(pls)
library(nnet)
library(randomForest)
# Boosted Tree Ensemble via XGBoost
# this section takes 10 min to complete
# XGboost works with using the xgb.DMatrix function
# Creating a cross validation control
xgb_trcontrol = trainControl(
  method = "cv",
  number = 5,
  allowParallel = TRUE,
  verboseIter = FALSE,
  returnData = FALSE
)
# Setting up a grid search for the best parameters
xgbGrid <- expand.grid(nrounds = c(100,200),  # this is n_estimators above
                       max_depth = c(10, 15, 20, 25),
                       colsample_bytree = seq(0.5, 0.9, length.out = 5),
                       ## The values below are default values in the sklearn-api.
                       eta = 0.1,
                       gamma=0,
                       min_child_weight = 1,
                       subsample = 1
                      )

set.seed(123)

#xgb_model = train(
#  xTrain, as.factor(yTrain),
#  trControl = xgb_trcontrol,
#  tuneGrid = xgbGrid,
#  method = "xgbTree"
#)

# Testing against data set.
#predicted <- predict(xgb_model, xTest)
#table(predicted)
```

```
#randomForest
# Testing on data set
#rf_model <- randomForest(x = xTrain, y = as.factor(yTrain), ntree = 500)

#xTest <- xTest %>% na.omit()

#predicted <- predict(rf_model, xTest)
#table(predicted)

#rf_varimp <- varImp(rf_model)
#rf_varimp
#varImpPlot(rf_model)
```