# Project 2

Brett Davidoff, Donald Butler, Tyler Brown

2023-12-17

## Contents

# Intoduction

This is role playing. I am your new boss. I am in charge of production at ABC Beverage and you are a team of data scientists reporting to me. My leadership has told me that new regulations are requiring us to understand our manufacturing process, the predictive factors and be able to report to them our predictive model of PH.

Please use the historical data set I am providing. Build and report the factors in BOTH a technical and non-technical report. I like to use Word and Excel. Please provide your non-technical report in a business-friendly readable document and your predictions in an Excel-readable format. The technical report should show clearly the models you tested and how you selected your final approach.

Please submit both Rpubs links and .rmd files or other readable formats for technical and non-technical reports. Also submit the Excel file showing the prediction of your models for pH.

```
# Add libraries for data analysis and visualization
library(tidyverse)
library(fpp3)
library(caret)
library(RANN)
library(psych)
library(DataExplorer)
library(randomForest)
library(Cubist)
library(rpart)
library(openxlsx)
library(corrplot)
library(mice)
library(earth)
```

# Data Exploration

## Loading and Evaluating the Data

Historical data was provided in an Excel document. The data is read into a dataframe and evaluated.

```
# Read data from Excel files
train.df <- read.xlsx('StudentData.xlsx', sheet = 1)
eval.df <- read.xlsx('StudentEvaluation.xlsx', sheet = 1)
glimpse(train.df)
```

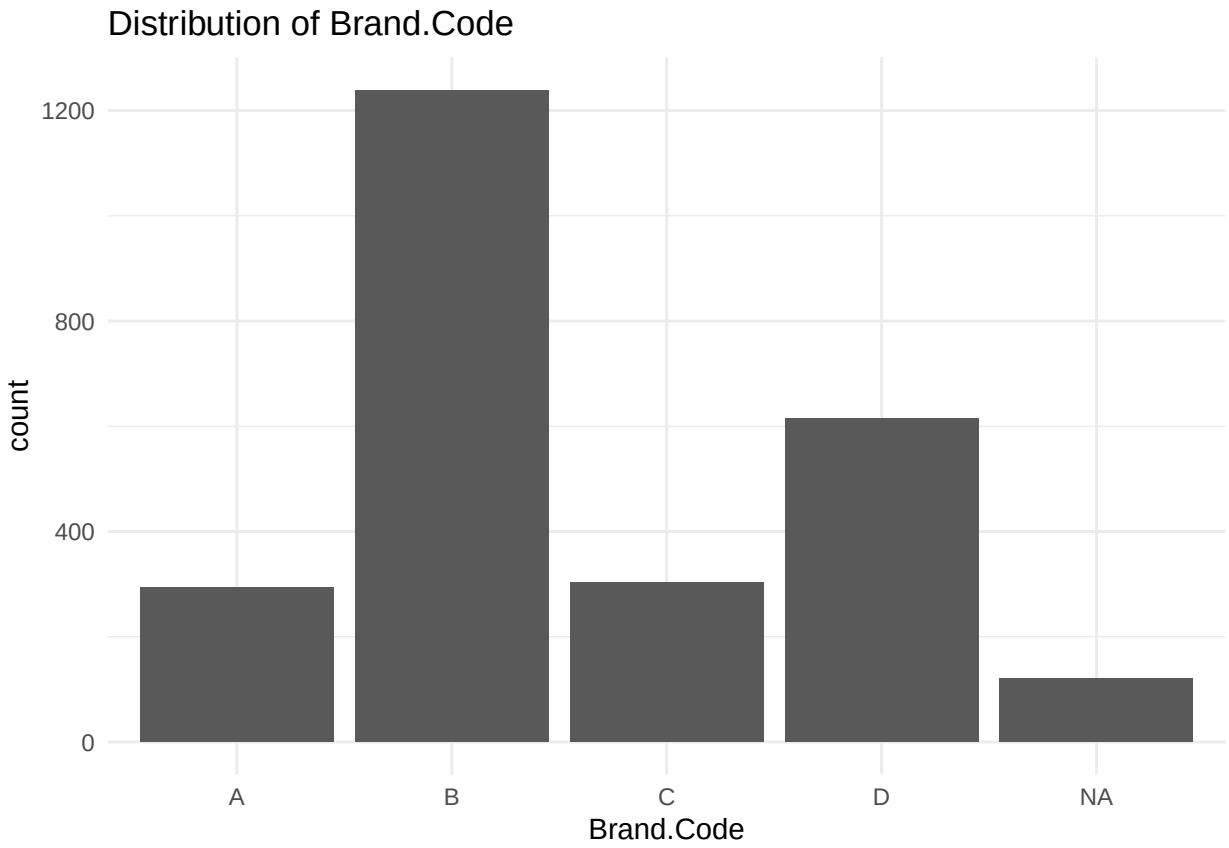
```
## Rows: 2,571
## Columns: 33
## $ Brand.Code        <chr> "B", "A", "B", "A", "A", "A", "A", "B", "B", "B", "B~
```

```
## $ Carb.Volume       <dbl> 5.340000, 5.426667, 5.286667, 5.440000, 5.486667, 5.~
## $ Fill.Ounces       <dbl> 23.96667, 24.00667, 24.06000, 24.00667, 24.31333, 23~
## $ PC.Volume         <dbl> 0.2633333, 0.2386667, 0.2633333, 0.2933333, 0.111333~
## $ Carb.Pressure     <dbl> 68.2, 68.4, 70.8, 63.0, 67.2, 66.6, 64.2, 67.6, 64.2~
## $ Carb.Temp         <dbl> 141.2, 139.6, 144.8, 132.6, 136.8, 138.4, 136.8, 141~
## $ PSC               <dbl> 0.104, 0.124, 0.090, NA, 0.026, 0.090, 0.128, 0.154,~
## $ PSC.Fill          <dbl> 0.26, 0.22, 0.34, 0.42, 0.16, 0.24, 0.40, 0.34, 0.12~
## $ PSC.CO2           <dbl> 0.04, 0.04, 0.16, 0.04, 0.12, 0.04, 0.04, 0.04, 0.14~
## $ Mnf.Flow          <dbl> -100, -100, -100, -100, -100, -100, -100, -100, -100~
## $ Carb.Pressure1    <dbl> 118.8, 121.6, 120.2, 115.2, 118.4, 119.6, 122.2, 124~
## $ Fill.Pressure     <dbl> 46.0, 46.0, 46.0, 46.4, 45.8, 45.6, 51.8, 46.8, 46.0~
## $ Hyd.Pressure1     <dbl> 0, 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, 0~
## $ Hyd.Pressure3     <dbl> NA, NA, NA, 0, 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, 86~
## $ Filler.Level      <dbl> 121.2, 118.6, 120.0, 117.8, 118.6, 120.2, 123.4, 118~
## $ 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, 65.4~
## $ Usage.cont        <dbl> 16.18, 19.90, 17.76, 17.42, 17.68, 23.82, 20.74, 18.~
## $ Carb.Flow         <dbl> 2932, 3144, 2914, 3062, 3054, 2948, 30, 684, 2902, 3~
## $ Density           <dbl> 0.88, 0.92, 1.58, 1.54, 1.54, 1.52, 0.84, 0.84, 0.90~
## $ MFR               <dbl> 725.0, 726.8, 735.0, 730.6, 722.8, 738.8, NA, NA, 74~
## $ Balling           <dbl> 1.398, 1.498, 3.142, 3.042, 3.042, 2.992, 1.298, 1.2~
## $ Pressure.Vacuum   <dbl> -4.0, -4.0, -3.8, -4.4, -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, 8.38~
## $ Oxygen.Filler     <dbl> 0.022, 0.026, 0.024, 0.030, 0.030, 0.024, 0.066, 0.0~
## $ Bowl.Setpoint     <dbl> 120, 120, 120, 120, 120, 120, 120, 120, 120, 120, 12~
## $ Pressure.Setpoint <dbl> 46.4, 46.8, 46.6, 46.0, 46.0, 46.0, 46.0, 46.0, 46.0~
## $ Air.Pressurer     <dbl> 142.6, 143.0, 142.0, 146.2, 146.2, 146.6, 146.2, 146~
## $ Alch.Rel          <dbl> 6.58, 6.56, 7.66, 7.14, 7.14, 7.16, 6.54, 6.52, 6.52~
## $ Carb.Rel          <dbl> 5.32, 5.30, 5.84, 5.42, 5.44, 5.44, 5.38, 5.34, 5.34~
## $ Balling.Lvl       <dbl> 1.48, 1.56, 3.28, 3.04, 3.04, 3.02, 1.44, 1.44, 1.44~
```

### Categorical Data

`Brand.Code` is the only categorical predictor in the data set and needs to be converted to a factor.

```
# Convert Brand.Code to a factor
train.df <- train.df |>
  mutate(Brand.Code = as.factor(Brand.Code))

eval.df <- eval.df |>
  mutate(Brand.Code = as.factor(Brand.Code))

# Create a bar plot to visualize the distribution of Brand.Code
train.df |>
  ggplot() +
  geom_bar(aes(x = Brand.Code)) +
  labs(title = 'Distribution of Brand.Code') +
  #plot data
  theme_minimal()
```




### Numerical Distributions

PH is the response variable and the remaining are predcictors for that response.

```
# Display summary statistics for numerical variables
train.df |>
  keep(is.numeric ) |>
  summary()
```

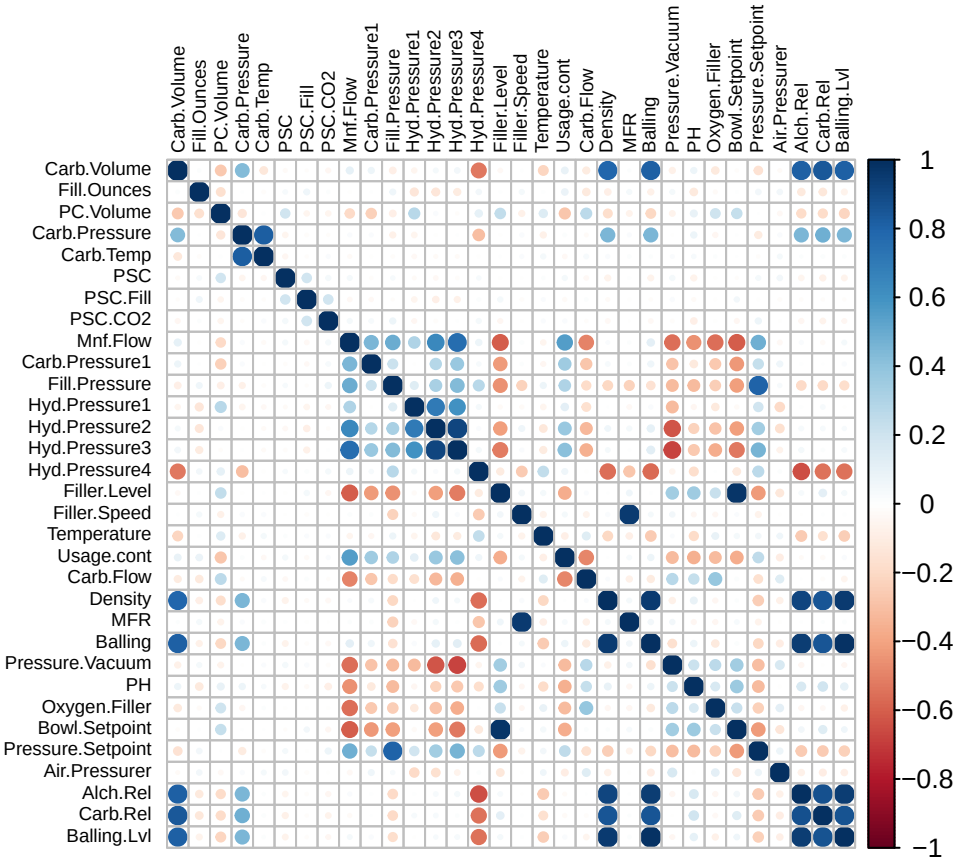
```
##   Carb.Volume     Fill.Ounces      PC.Volume       Carb.Pressure
##  Min.   :5.040   Min.   :23.63   Min.   :0.07933   Min.   :57.00
##  1st Qu.:5.293   1st Qu.:23.92   1st Qu.:0.23917   1st Qu.:65.60
##  Median :5.347   Median :23.97   Median :0.27133   Median :68.20
##  Mean   :5.370   Mean   :23.97   Mean   :0.27712   Mean   :68.19
##  3rd Qu.:5.453   3rd Qu.:24.03   3rd Qu.:0.31200   3rd Qu.:70.60
##  Max.   :5.700   Max.   :24.32   Max.   :0.47800   Max.   :79.40
##  NA's   :10      NA's   :38      NA's   :39        NA's   :27
##    Carb.Temp          PSC             PSC.Fill         PSC.CO2
##  Min.   :128.6   Min.   :0.00200   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:138.4   1st Qu.:0.04800   1st Qu.:0.1000   1st Qu.:0.02000
##  Median :140.8   Median :0.07600   Median :0.1800   Median :0.04000
##  Mean   :141.1   Mean   :0.08457   Mean   :0.1954   Mean   :0.05641
##  3rd Qu.:143.8   3rd Qu.:0.11200   3rd Qu.:0.2600   3rd Qu.:0.08000
##  Max.   :154.0   Max.   :0.27000   Max.   :0.6200   Max.   :0.24000
##  NA's   :26      NA's   :33        NA's   :23       NA's   :39
##     Mnf.Flow       Carb.Pressure1  Fill.Pressure   Hyd.Pressure1
```

```
##  Min.   :-100.20   Min.   :105.6   Min.   :34.60   Min.   :-0.80
##  1st Qu.:-100.00   1st Qu.:119.0   1st Qu.:46.00   1st Qu.: 0.00
##  Median :  65.20   Median :123.2   Median :46.40   Median :11.40
##  Mean   :  24.57   Mean   :122.6   Mean   :47.92   Mean   :12.44
##  3rd Qu.: 140.80   3rd Qu.:125.4   3rd Qu.:50.00   3rd Qu.:20.20
##  Max.   : 229.40   Max.   :140.2   Max.   :60.40   Max.   :58.00
##  NA's   :2         NA's   :32      NA's   :22      NA's   :11
##  Hyd.Pressure2   Hyd.Pressure3   Hyd.Pressure4     Filler.Level
##  Min.   : 0.00   Min.   :-1.20   Min.   : 52.00   Min.   : 55.8
##  1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.: 86.00   1st Qu.: 98.3
##  Median :28.60   Median :27.60   Median : 96.00   Median :118.4
##  Mean   :20.96   Mean   :20.46   Mean   : 96.29   Mean   :109.3
##  3rd Qu.:34.60   3rd Qu.:33.40   3rd Qu.:102.00   3rd Qu.:120.0
##  Max.   :59.40   Max.   :50.00   Max.   :142.00   Max.   :161.2
##  NA's   :15      NA's   :15      NA's   :30       NA's   :20
##   Filler.Speed   Temperature      Usage.cont      Carb.Flow       Density
##  Min.   : 998   Min.   :63.60   Min.   :12.08   Min.   :  26   Min.   :0.240
##  1st Qu.:3888   1st Qu.:65.20   1st Qu.:18.36   1st Qu.:1144   1st Qu.:0.900
##  Median :3982   Median :65.60   Median :21.79   Median :3028   Median :0.980
##  Mean   :3687   Mean   :65.97   Mean   :20.99   Mean   :2468   Mean   :1.174
##  3rd Qu.:3998   3rd Qu.:66.40   3rd Qu.:23.75   3rd Qu.:3186   3rd Qu.:1.620
##  Max.   :4030   Max.   :76.20   Max.   :25.90   Max.   :5104   Max.   :1.920
##  NA's   :57     NA's   :14      NA's   :5       NA's   :2      NA's   :1
##       MFR           Balling       Pressure.Vacuum        PH
##  Min.   : 31.4   Min.   :-0.170   Min.   :-6.600   Min.   :7.880
##  1st Qu.:706.3   1st Qu.: 1.496   1st Qu.:-5.600   1st Qu.:8.440
##  Median :724.0   Median : 1.648   Median :-5.400   Median :8.540
##  Mean   :704.0   Mean   : 2.198   Mean   :-5.216   Mean   :8.546
##  3rd Qu.:731.0   3rd Qu.: 3.292   3rd Qu.:-5.000   3rd Qu.:8.680
##  Max.   :868.6   Max.   : 4.012   Max.   :-3.600   Max.   :9.360
##  NA's   :212     NA's   :1                         NA's   :4
##  Oxygen.Filler     Bowl.Setpoint   Pressure.Setpoint Air.Pressurer
##  Min.   :0.00240   Min.   : 70.0   Min.   :44.00     Min.   :140.8
##  1st Qu.:0.02200   1st Qu.:100.0   1st Qu.:46.00     1st Qu.:142.2
##  Median :0.03340   Median :120.0   Median :46.00     Median :142.6
##  Mean   :0.04684   Mean   :109.3   Mean   :47.62     Mean   :142.8
##  3rd Qu.:0.06000   3rd Qu.:120.0   3rd Qu.:50.00     3rd Qu.:143.0
##  Max.   :0.40000   Max.   :140.0   Max.   :52.00     Max.   :148.2
##  NA's   :12        NA's   :2       NA's   :12
##     Alch.Rel        Carb.Rel      Balling.Lvl
##  Min.   :5.280   Min.   :4.960   Min.   :0.00
##  1st Qu.:6.540   1st Qu.:5.340   1st Qu.:1.38
##  Median :6.560   Median :5.400   Median :1.48
##  Mean   :6.897   Mean   :5.437   Mean   :2.05
##  3rd Qu.:7.240   3rd Qu.:5.540   3rd Qu.:3.14
##  Max.   :8.620   Max.   :6.060   Max.   :3.66
##  NA's   :9       NA's   :10      NA's   :1
```

### Predictor Correlations

We constructed a correlation plot to determine which predictors are related to `PH`, and also to determine if there are predictors that are highly related to other variables.

```
# Create a correlation plot for numerical variables
train.df |>
  keep(is.numeric) |>
  na.omit() |>
  cor() |>
  corrplot(tl.col = 'black', tl.cex = .6)
```




Look at the chart, there are several variables that have a very low correlation to `PH`, for example, `Carb.Temp`, `PSC`, `PSC.Fill`, and `PSC.CO2`, seem to have little relationship to `PH`.

### Near-Zero Variance

We want to look for variables that have near-zero variance which indicate that they play little to no role in determining the `PH` of the beverage.

```
# Identify and filter variables with near-zero variance
train.df |>
  nearZeroVar(saveMetrics = TRUE) |>
  filter(nzv == TRUE)
```

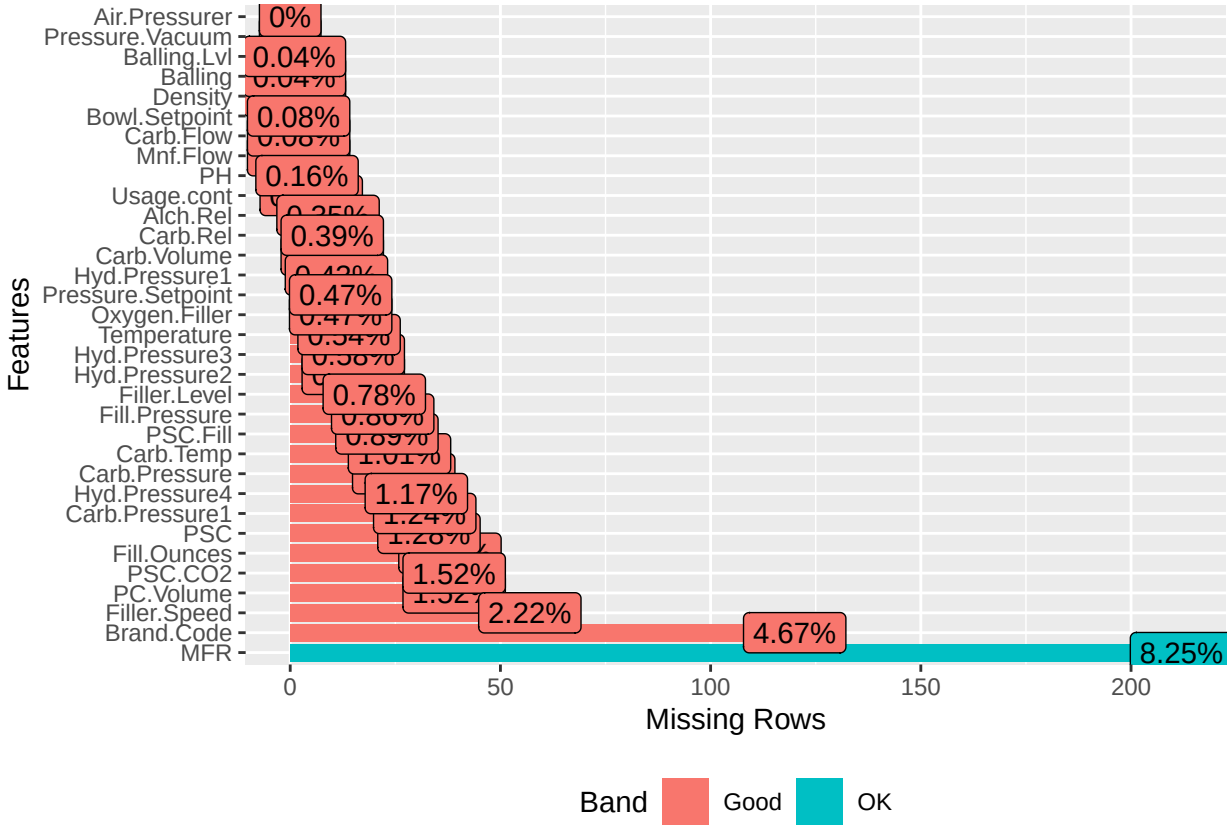
```
##                freqRatio percentUnique zeroVar  nzv
## Hyd.Pressure1   31.11111      9.529366   FALSE TRUE
```

The predictor `Hyd.Pressure1` has near-zero variance and will be removed from the model.

### Missing Values

Many of the predictors are missing values that will need to be evaluated and imputed to develop a consistent model.

```
# Create a missing data plot
plot_missing(train.df)
```




Using the mice package with Predictive Mean Matching to impute the missing values. Then removing `Hyd.Pressure1` which has near-zero variance.

```
# Impute missing values using Predictive Mean Matching
impute.df <- train.df |>
  mice(m = 1, method = 'pmm', print = FALSE) |>
  complete()

# Remove near-zero parameters Hyd.Pressure1
impute.df <- impute.df[,-nearZeroVar(impute.df)]
summary(impute.df)
```

```
##  Brand.Code  Carb.Volume     Fill.Ounces      PC.Volume       Carb.Pressure
##  A: 308     Min.   :5.040   Min.   :23.63   Min.   :0.07933   Min.   :57.00
##  B:1315     1st Qu.:5.293   1st Qu.:23.92   1st Qu.:0.23933   1st Qu.:65.60
##  C: 326     Median :5.347   Median :23.97   Median :0.27133   Median :68.20
##  D: 622     Mean   :5.370   Mean   :23.97   Mean   :0.27773   Mean   :68.21
##             3rd Qu.:5.457   3rd Qu.:24.03   3rd Qu.:0.31267   3rd Qu.:70.60
##             Max.   :5.700   Max.   :24.32   Max.   :0.47800   Max.   :79.40
```

```
##    Carb.Temp          PSC             PSC.Fill         PSC.CO2
##  Min.   :128.6   Min.   :0.00200   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:138.4   1st Qu.:0.04800   1st Qu.:0.1000   1st Qu.:0.02000
##  Median :140.8   Median :0.07600   Median :0.1800   Median :0.04000
##  Mean   :141.1   Mean   :0.08491   Mean   :0.1964   Mean   :0.05661
##  3rd Qu.:143.8   3rd Qu.:0.11200   3rd Qu.:0.2600   3rd Qu.:0.08000
##  Max.   :154.0   Max.   :0.27000   Max.   :0.6200   Max.   :0.24000
##     Mnf.Flow       Carb.Pressure1  Fill.Pressure   Hyd.Pressure2
##  Min.   :-100.20   Min.   :105.6   Min.   :34.60   Min.   : 0.00
##  1st Qu.:-100.00   1st Qu.:119.0   1st Qu.:46.00   1st Qu.: 0.00
##  Median :  65.20   Median :123.2   Median :46.40   Median :28.60
##  Mean   :  24.55   Mean   :122.6   Mean   :47.91   Mean   :20.96
##  3rd Qu.: 140.80   3rd Qu.:125.4   3rd Qu.:50.00   3rd Qu.:34.60
##  Max.   : 229.40   Max.   :140.2   Max.   :60.40   Max.   :59.40
##  Hyd.Pressure3   Hyd.Pressure4     Filler.Level    Filler.Speed
##  Min.   :-1.20   Min.   : 52.00   Min.   : 55.8   Min.   : 998
##  1st Qu.: 0.00   1st Qu.: 86.00   1st Qu.: 97.7   1st Qu.:3819
##  Median :27.40   Median : 96.00   Median :118.4   Median :3980
##  Mean   :20.44   Mean   : 96.55   Mean   :109.2   Mean   :3637
##  3rd Qu.:33.20   3rd Qu.:102.00   3rd Qu.:120.0   3rd Qu.:3996
##  Max.   :50.00   Max.   :142.00   Max.   :161.2   Max.   :4030
##   Temperature      Usage.cont      Carb.Flow       Density           MFR
##  Min.   :63.60   Min.   :12.08   Min.   :  26   Min.   :0.240   Min.   : 31.4
##  1st Qu.:65.20   1st Qu.:18.36   1st Qu.:1142   1st Qu.:0.900   1st Qu.:695.0
##  Median :65.60   Median :21.80   Median :3028   Median :0.980   Median :721.4
##  Mean   :65.98   Mean   :20.99   Mean   :2468   Mean   :1.174   Mean   :671.1
##  3rd Qu.:66.40   3rd Qu.:23.74   3rd Qu.:3186   3rd Qu.:1.620   3rd Qu.:730.4
##  Max.   :76.20   Max.   :25.90   Max.   :5104   Max.   :1.920   Max.   :868.6
##     Balling       Pressure.Vacuum        PH        Oxygen.Filler
##  Min.   :-0.170   Min.   :-6.600   Min.   :7.880   Min.   :0.00240
##  1st Qu.: 1.496   1st Qu.:-5.600   1st Qu.:8.440   1st Qu.:0.02200
##  Median : 1.648   Median :-5.400   Median :8.540   Median :0.03340
##  Mean   : 2.198   Mean   :-5.216   Mean   :8.546   Mean   :0.04713
##  3rd Qu.: 3.292   3rd Qu.:-5.000   3rd Qu.:8.680   3rd Qu.:0.06000
##  Max.   : 4.012   Max.   :-3.600   Max.   :9.360   Max.   :0.40000
##  Bowl.Setpoint   Pressure.Setpoint Air.Pressurer      Alch.Rel
##  Min.   : 70.0   Min.   :44.00     Min.   :140.8   Min.   :5.280
##  1st Qu.:100.0   1st Qu.:46.00     1st Qu.:142.2   1st Qu.:6.540
##  Median :120.0   Median :46.00     Median :142.6   Median :6.560
##  Mean   :109.3   Mean   :47.61     Mean   :142.8   Mean   :6.897
##  3rd Qu.:120.0   3rd Qu.:50.00     3rd Qu.:143.0   3rd Qu.:7.230
##  Max.   :140.0   Max.   :52.00     Max.   :148.2   Max.   :8.620
##     Carb.Rel      Balling.Lvl
##  Min.   :4.960   Min.   :0.00
##  1st Qu.:5.340   1st Qu.:1.38
##  Median :5.400   Median :1.48
##  Mean   :5.436   Mean   :2.05
##  3rd Qu.:5.540   3rd Qu.:3.14
##  Max.   :6.060   Max.   :3.66
```

# Build Models

## Split-Data

We will create an 80/20 split of the data so that we can evaluate the effectiveness of each model.

### LM

Construct a Linear Regression Model with the training set and evaluate against the test set.

```
set.seed(200)

lm.model <- train(train.x, train.y,
                  method = "lm",
                  trControl = trainControl(method = "cv",number = 10))
lm.model
```

```
## Linear Regression
##
## 2058 samples
##   31 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1852, 1851, 1853, 1852, 1853, 1853, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.1352196  0.3941456  0.1048494
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
lm.pred <- predict(lm.model, newdata = test.x)
postResample(pred = lm.pred, obs = test.y)
```

```
##       RMSE   Rsquared        MAE
## 0.13125265 0.40666796 0.09944526
```

### PLS

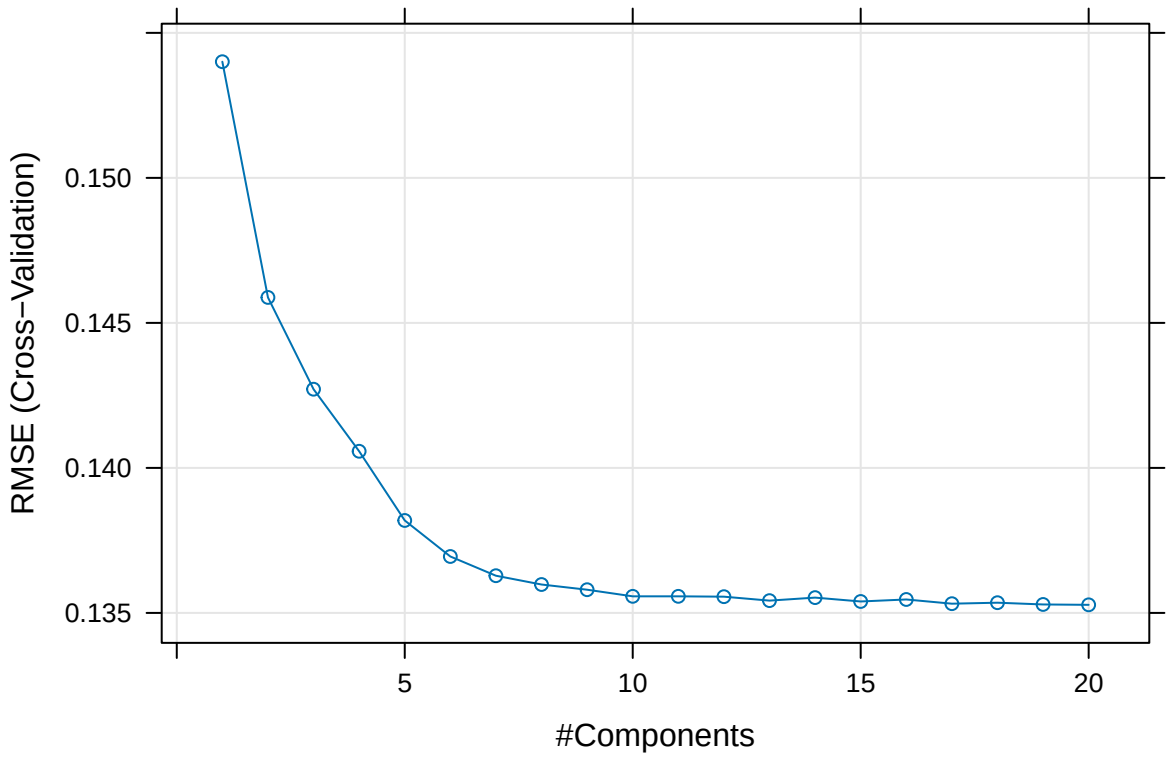
Construct a Partial Least Squares Model with the training set and evaluate against the test set.

```
set.seed(200)

pls.model <- train(train.x, train.y,
                   method='pls',
                   tuneLength=20,
                   trControl=trainControl(method='cv', number=10),
```

```
                  preProc=c('center','scale'))

pls.model
```

```
## Partial Least Squares
##
## 2058 samples
##   31 predictor
##
## Pre-processing: centered (30), scaled (30), ignore (1)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1852, 1851, 1853, 1852, 1853, 1853, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE       Rsquared   MAE
##    1     0.1540058  0.2085488  0.1224548
##    2     0.1458785  0.2907300  0.1142264
##    3     0.1427139  0.3211356  0.1119362
##    4     0.1405736  0.3421281  0.1104976
##    5     0.1381878  0.3652074  0.1078053
##    6     0.1369463  0.3767096  0.1073114
##    7     0.1362826  0.3832350  0.1063576
##    8     0.1359799  0.3859379  0.1060411
##    9     0.1357999  0.3875963  0.1056483
##   10     0.1355717  0.3896280  0.1054913
##   11     0.1355720  0.3897737  0.1055457
##   12     0.1355600  0.3899355  0.1053695
##   13     0.1354246  0.3912435  0.1053533
##   14     0.1355278  0.3907548  0.1052795
##   15     0.1353943  0.3922146  0.1051748
##   16     0.1354640  0.3917222  0.1052796
##   17     0.1353180  0.3931961  0.1051281
##   18     0.1353527  0.3929297  0.1051131
##   19     0.1352917  0.3934865  0.1050694
##   20     0.1352790  0.3936302  0.1050448
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 20.
```

```
plot(pls.model)
```




```
pls.pred <- predict(pls.model, newdata = test.x)
postResample(pred = pls.pred, obs = test.y)
```

```
##       RMSE   Rsquared        MAE
## 0.13122563 0.40680163 0.09967673
```

### MARS

Construct a Multivariate Adaptive Regression Spline Model with the training set and evaluate against the test set.

```
set.seed(200)

mars.model <- train(x = train.x,
                    y = train.y,
                    method='earth',
                    tuneGrid = expand.grid(.degree = 1:2, .nprune = 2:15),
                    preProcess = c('center','scale'),
                    tuneLength = 10,
                    trControl = trainControl(method='cv'))

mars.model
```

```
## Multivariate Adaptive Regression Spline
##
## 2058 samples
##   31 predictor
##
```

```
## Pre-processing: centered (30), scaled (30), ignore (1)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1852, 1851, 1853, 1852, 1853, 1853, ...
## Resampling results across tuning parameters:
##
##   degree  nprune  RMSE       Rsquared   MAE
##   1        2      0.1533707  0.2160827  0.12041325
##   1        3      0.1452316  0.2984651  0.11406093
##   1        4      0.1433105  0.3167566  0.11185935
##   1        5      0.1412967  0.3356402  0.11074189
##   1        6      0.1403053  0.3457627  0.10910620
##   1        7      0.1379904  0.3677464  0.10714223
##   1        8      0.1362781  0.3830074  0.10571799
##   1        9      0.1347267  0.3964502  0.10391368
##   1       10      0.1332380  0.4094487  0.10321580
##   1       11      0.1327546  0.4135390  0.10241921
##   1       12      0.1323425  0.4170303  0.10215519
##   1       13      0.1322686  0.4180857  0.10187043
##   1       14      0.1319259  0.4213700  0.10164560
##   1       15      0.1317546  0.4234226  0.10146405
##   2        2      0.1533707  0.2160827  0.12041325
##   2        3      0.1463621  0.2877279  0.11555681
##   2        4      0.1441228  0.3093653  0.11272207
##   2        5      0.1428376  0.3221746  0.11156166
##   2        6      0.1403087  0.3455610  0.10924430
##   2        7      0.1387740  0.3591495  0.10779423
##   2        8      0.1366048  0.3789081  0.10590635
##   2        9      0.1350471  0.3931003  0.10360121
##   2       10      0.1338721  0.4053752  0.10225736
##   2       11      0.1332930  0.4113214  0.10195356
##   2       12      0.1310075  0.4309044  0.09999013
##   2       13      0.1295508  0.4433647  0.09829696
##   2       14      0.1298500  0.4418995  0.09819203
##   2       15      0.1280382  0.4563576  0.09720005
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nprune = 15 and degree = 2.
```

```
mars.pred <- predict(mars.model, newdata = test.x)
postResample(pred = mars.pred, obs = test.y)
```

```
##       RMSE   Rsquared        MAE
## 0.12884058 0.43486127 0.09654309
```

### BT

Construct a Boosted Tree Model with the training set and evaluate against the test set.

```
set.seed(200)

gbm.model <- train(train.x, train.y,
                   method="gbm",
```

```
                trControl=trainControl(method="cv",n=10),
                tuneGrid=expand.grid(interaction.depth = seq(1,7,by=2),
                                     n.trees=seq(100, 1000, by=50),
                                     shrinkage=c(0.01,0.1,by=0.01),
                                     n.minobsinnode=10),
                verbose=FALSE)
gbm.model
```

```
## Stochastic Gradient Boosting
##
## 2058 samples
##   31 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1852, 1851, 1853, 1852, 1853, 1853, ...
## Resampling results across tuning parameters:
##
##   shrinkage  interaction.depth  n.trees  RMSE       Rsquared   MAE
##   0.01       1                   100     0.1537235  0.3075931  0.12239535
##   0.01       1                   150     0.1487208  0.3398308  0.11803981
##   0.01       1                   200     0.1451050  0.3581143  0.11492483
##   0.01       1                   250     0.1424130  0.3719026  0.11273195
##   0.01       1                   300     0.1404158  0.3792639  0.11108052
##   0.01       1                   350     0.1389300  0.3856153  0.10983503
##   0.01       1                   400     0.1377807  0.3908270  0.10889645
##   0.01       1                   450     0.1369796  0.3945591  0.10825232
##   0.01       1                   500     0.1362914  0.3985708  0.10769350
##   0.01       1                   550     0.1357394  0.4015318  0.10720519
##   0.01       1                   600     0.1352390  0.4046710  0.10677133
##   0.01       1                   650     0.1347176  0.4084274  0.10630111
##   0.01       1                   700     0.1342776  0.4117412  0.10596201
##   0.01       1                   750     0.1339362  0.4138995  0.10565985
##   0.01       1                   800     0.1335529  0.4167397  0.10529934
##   0.01       1                   850     0.1332257  0.4191981  0.10497739
##   0.01       1                   900     0.1329470  0.4210485  0.10472413
##   0.01       1                   950     0.1326579  0.4231420  0.10441341
##   0.01       1                  1000     0.1323536  0.4254666  0.10410476
##   0.01       3                   100     0.1428645  0.4184584  0.11370335
##   0.01       3                   150     0.1365704  0.4397653  0.10829629
##   0.01       3                   200     0.1325372  0.4560128  0.10485200
##   0.01       3                   250     0.1299695  0.4677654  0.10257065
##   0.01       3                   300     0.1281753  0.4770620  0.10090888
##   0.01       3                   350     0.1267679  0.4850809  0.09955898
##   0.01       3                   400     0.1256249  0.4922095  0.09851906
##   0.01       3                   450     0.1245627  0.4991335  0.09747428
##   0.01       3                   500     0.1236002  0.5055310  0.09660279
##   0.01       3                   550     0.1228346  0.5105346  0.09583778
##   0.01       3                   600     0.1221905  0.5146680  0.09517987
##   0.01       3                   650     0.1217085  0.5174910  0.09470052
##   0.01       3                   700     0.1212114  0.5206238  0.09417103
##   0.01       3                   750     0.1207272  0.5237041  0.09364663
##   0.01       3                   800     0.1202882  0.5266871  0.09318134
```

```
##   0.01       3                850     0.1199552  0.5287745  0.09280640
##   0.01       3                900     0.1195940  0.5312018  0.09240964
##   0.01       3                950     0.1192742  0.5332720  0.09205388
##   0.01       3               1000     0.1189594  0.5353642  0.09170827
##   0.01       5                100     0.1382064  0.4669824  0.10965969
##   0.01       5                150     0.1314876  0.4856918  0.10381258
##   0.01       5                200     0.1274044  0.4992757  0.10017170
##   0.01       5                250     0.1246975  0.5112085  0.09767872
##   0.01       5                300     0.1226941  0.5212520  0.09581106
##   0.01       5                350     0.1211842  0.5297126  0.09438570
##   0.01       5                400     0.1198672  0.5374367  0.09316006
##   0.01       5                450     0.1188977  0.5428710  0.09216198
##   0.01       5                500     0.1179751  0.5486321  0.09128248
##   0.01       5                550     0.1172794  0.5527858  0.09058082
##   0.01       5                600     0.1166755  0.5564456  0.08997166
##   0.01       5                650     0.1160549  0.5604347  0.08931508
##   0.01       5                700     0.1155110  0.5639872  0.08877699
##   0.01       5                750     0.1150246  0.5670044  0.08831542
##   0.01       5                800     0.1146154  0.5695319  0.08788976
##   0.01       5                850     0.1142548  0.5718665  0.08758301
##   0.01       5                900     0.1138786  0.5742268  0.08719719
##   0.01       5                950     0.1135450  0.5763246  0.08684885
##   0.01       5               1000     0.1132573  0.5780668  0.08656140
##   0.01       7                100     0.1353408  0.5000220  0.10716041
##   0.01       7                150     0.1281044  0.5176988  0.10072693
##   0.01       7                200     0.1237060  0.5313898  0.09674495
##   0.01       7                250     0.1209143  0.5422302  0.09420634
##   0.01       7                300     0.1189298  0.5508470  0.09233507
##   0.01       7                350     0.1173871  0.5585467  0.09076888
##   0.01       7                400     0.1162197  0.5648365  0.08964980
##   0.01       7                450     0.1151230  0.5713468  0.08859245
##   0.01       7                500     0.1142769  0.5759968  0.08768628
##   0.01       7                550     0.1137055  0.5788567  0.08705071
##   0.01       7                600     0.1130962  0.5823536  0.08639379
##   0.01       7                650     0.1125502  0.5855708  0.08580856
##   0.01       7                700     0.1120482  0.5886257  0.08528053
##   0.01       7                750     0.1116469  0.5911169  0.08487731
##   0.01       7                800     0.1113245  0.5928310  0.08450475
##   0.01       7                850     0.1109373  0.5953033  0.08412665
##   0.01       7                900     0.1107134  0.5965246  0.08389570
##   0.01       7                950     0.1104075  0.5984336  0.08360973
##   0.01       7               1000     0.1101234  0.6002318  0.08333252
##   0.10       1                100     0.1322164  0.4253368  0.10406739
##   0.10       1                150     0.1302634  0.4398300  0.10198463
##   0.10       1                200     0.1291215  0.4478103  0.10071906
##   0.10       1                250     0.1289392  0.4484538  0.10008812
##   0.10       1                300     0.1288439  0.4484341  0.09964895
##   0.10       1                350     0.1286845  0.4499254  0.09929088
##   0.10       1                400     0.1286789  0.4497017  0.09909571
##   0.10       1                450     0.1289266  0.4475093  0.09923390
##   0.10       1                500     0.1287553  0.4489600  0.09899560
##   0.10       1                550     0.1290692  0.4465942  0.09910943
##   0.10       1                600     0.1288035  0.4489534  0.09868688
##   0.10       1                650     0.1287327  0.4493413  0.09871292
```

```
##   0.10       1                  700     0.1286718  0.4498847  0.09862409
##   0.10       1                  750     0.1287102  0.4496151  0.09872327
##   0.10       1                  800     0.1287998  0.4491095  0.09877647
##   0.10       1                  850     0.1287462  0.4499116  0.09860587
##   0.10       1                  900     0.1286772  0.4501600  0.09854641
##   0.10       1                  950     0.1287199  0.4501066  0.09858044
##   0.10       1                 1000     0.1286571  0.4509356  0.09840865
##   0.10       3                  100     0.1198656  0.5261794  0.09220562
##   0.10       3                  150     0.1179415  0.5391569  0.09015794
##   0.10       3                  200     0.1170774  0.5444598  0.08929045
##   0.10       3                  250     0.1164501  0.5487582  0.08836430
##   0.10       3                  300     0.1156613  0.5545877  0.08773574
##   0.10       3                  350     0.1150682  0.5589348  0.08716933
##   0.10       3                  400     0.1147967  0.5610690  0.08679774
##   0.10       3                  450     0.1142793  0.5652642  0.08636143
##   0.10       3                  500     0.1142597  0.5654968  0.08635085
##   0.10       3                  550     0.1136683  0.5700004  0.08584676
##   0.10       3                  600     0.1135432  0.5710838  0.08576367
##   0.10       3                  650     0.1135952  0.5708854  0.08585674
##   0.10       3                  700     0.1134137  0.5726888  0.08555375
##   0.10       3                  750     0.1132246  0.5740430  0.08545183
##   0.10       3                  800     0.1131451  0.5749375  0.08545132
##   0.10       3                  850     0.1130579  0.5758191  0.08536547
##   0.10       3                  900     0.1128520  0.5776030  0.08529829
##   0.10       3                  950     0.1126825  0.5788755  0.08513729
##   0.10       3                 1000     0.1126488  0.5792853  0.08522702
##   0.10       5                  100     0.1158571  0.5562155  0.08832091
##   0.10       5                  150     0.1142724  0.5666240  0.08654126
##   0.10       5                  200     0.1133352  0.5730285  0.08546615
##   0.10       5                  250     0.1125171  0.5791450  0.08448343
##   0.10       5                  300     0.1116749  0.5853145  0.08393292
##   0.10       5                  350     0.1111076  0.5895105  0.08357490
##   0.10       5                  400     0.1105503  0.5937659  0.08298818
##   0.10       5                  450     0.1105132  0.5941345  0.08293359
##   0.10       5                  500     0.1103080  0.5957203  0.08280426
##   0.10       5                  550     0.1101021  0.5976634  0.08266069
##   0.10       5                  600     0.1098930  0.5994052  0.08236939
##   0.10       5                  650     0.1096135  0.6014892  0.08204848
##   0.10       5                  700     0.1093896  0.6032442  0.08191065
##   0.10       5                  750     0.1095460  0.6022820  0.08208919
##   0.10       5                  800     0.1092855  0.6043049  0.08190358
##   0.10       5                  850     0.1091180  0.6055403  0.08167412
##   0.10       5                  900     0.1090556  0.6060659  0.08161721
##   0.10       5                  950     0.1089364  0.6070822  0.08151585
##   0.10       5                 1000     0.1089594  0.6071372  0.08156343
##   0.10       7                  100     0.1121450  0.5831856  0.08464719
##   0.10       7                  150     0.1105493  0.5936413  0.08300443
##   0.10       7                  200     0.1096291  0.6000574  0.08215171
##   0.10       7                  250     0.1092740  0.6023610  0.08175681
##   0.10       7                  300     0.1089540  0.6047874  0.08141465
##   0.10       7                  350     0.1087793  0.6061403  0.08130852
##   0.10       7                  400     0.1086225  0.6075561  0.08111967
##   0.10       7                  450     0.1078956  0.6128713  0.08061720
##   0.10       7                  500     0.1076605  0.6144798  0.08039116
```

```
##   0.10       7                  550     0.1074360  0.6163904  0.08012399
##   0.10       7                  600     0.1072559  0.6178429  0.07999936
##   0.10       7                  650     0.1073463  0.6175871  0.08001988
##   0.10       7                  700     0.1074992  0.6168575  0.08013695
##   0.10       7                  750     0.1072991  0.6183181  0.07993840
##   0.10       7                  800     0.1072779  0.6186341  0.07988318
##   0.10       7                  850     0.1072262  0.6191597  0.07981415
##   0.10       7                  900     0.1071978  0.6193997  0.07982380
##   0.10       7                  950     0.1072594  0.6190637  0.07985797
##   0.10       7                 1000     0.1073232  0.6186890  0.07990850
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 900, interaction.depth =
##  7, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
gbm.pred <- predict(gbm.model, newdata = test.x)
postResample(pred = gbm.pred, obs = test.y)
```

```
##       RMSE   Rsquared        MAE
## 0.10842965 0.59667510 0.07620956
```

**RF**

Construct a Random Forest Model with the training set and evaluate against the test set.

```
set.seed(200)

rf.model <- randomForest(train.x, train.y,
                         importance=TRUE,
                         ntree=2000)
rf.model
```

```
##
## Call:
##  randomForest(x = train.x, y = train.y, ntree = 2000, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 2000
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 0.009524937
##                     % Var explained: 68.2
```

```
rf.pred <- predict(rf.model, newdata = test.x)
postResample(pred = rf.pred, obs = test.y)
```

```
##       RMSE   Rsquared        MAE
## 0.10104635 0.66359092 0.07272722
```

### CART

Construct a Classification and Regression Tree Model with the training set and evaluate against the test set.

```
set.seed(200)

cart.model <- train(train.x, train.y,
                    method="rpart",
                    tuneLength = 10,
                    trControl=trainControl(method="cv"))
cart.model
```

```
## CART
##
## 2058 samples
##   31 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1852, 1851, 1853, 1852, 1853, 1853, ...
## Resampling results across tuning parameters:
##
##   cp          RMSE       Rsquared   MAE
##   0.01178492  0.1304242  0.4357046  0.1017862
##   0.01261162  0.1316097  0.4254410  0.1024183
##   0.01357267  0.1322932  0.4196458  0.1026836
##   0.01462094  0.1344206  0.3996851  0.1042461
##   0.01753657  0.1366527  0.3787315  0.1065699
##   0.02288435  0.1385708  0.3608027  0.1088106
##   0.03401464  0.1418522  0.3298281  0.1118829
##   0.04077565  0.1451570  0.2978453  0.1148074
##   0.06832760  0.1506664  0.2444004  0.1182787
##   0.21419108  0.1645887  0.1886571  0.1313111
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.01178492.
```

```
cart.pred <- predict(cart.model, newdata = test.x)
postResample(pred = cart.pred, obs = test.y)
```

```
##      RMSE  Rsquared       MAE
## 0.1344379 0.3803752 0.1059301
```

### SVM

Construct an SVM Model with the training set and evaluate against the test set. This model requires that the categorical predictor, `Brand.Code`, be removed.

```
set.seed(200)

svm.model <- train(train.x[,-1], train.y,
                   method = 'svmRadial',
```

```
                  preProcess = c('center','scale'),
                  tuneLength = 10,
                  trControl = trainControl(method = 'cv'))
svm.model
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 2058 samples
##   30 predictor
##
## Pre-processing: centered (30), scaled (30)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1852, 1851, 1853, 1852, 1853, 1853, ...
## Resampling results across tuning parameters:
##
##   C       RMSE       Rsquared   MAE
##     0.25  0.1328343  0.4212365  0.09864717
##     0.50  0.1299413  0.4440479  0.09510811
##     1.00  0.1276511  0.4619984  0.09243759
##     2.00  0.1259638  0.4751521  0.09078941
##     4.00  0.1247593  0.4858002  0.09011008
##     8.00  0.1252656  0.4859906  0.09106006
##    16.00  0.1275881  0.4759767  0.09310897
##    32.00  0.1312668  0.4613458  0.09631876
##    64.00  0.1351590  0.4487193  0.10020363
##   128.00  0.1400998  0.4336947  0.10471068
##
## Tuning parameter 'sigma' was held constant at a value of 0.0244548
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.0244548 and C = 4.
```

```
svm.pred = predict(svm.model, newdata = test.x[,-1])
postResample(pred = svm.pred, obs = test.y)
```

```
##       RMSE   Rsquared        MAE
## 0.12213413 0.48841986 0.08880445
```

## Model Results

```
results <- data.frame(as.list(postResample(pred = lm.pred, obs = test.y))) |> mutate(model = 'LM') |>
  union(data.frame(as.list(postResample(pred = pls.pred, obs = test.y))) |> mutate(model = 'PLS')) |>
  union(data.frame(as.list(postResample(pred = mars.pred, obs = test.y))) |> mutate(model = 'MARS')) |>
  union(data.frame(as.list(postResample(pred = gbm.pred, obs = test.y))) |> mutate(model = 'TREE')) |>
  union(data.frame(as.list(postResample(pred = rf.pred, obs = test.y))) |> mutate(model = 'RF')) |>
  union(data.frame(as.list(postResample(pred = cart.pred, obs = test.y))) |> mutate(model = 'CART')) |>
  union(data.frame(as.list(postResample(pred = svm.pred, obs = test.y))) |> mutate(model = 'SVM')) |>
  relocate(model, RMSE, Rsquared, MAE)

results |>
  arrange(desc(Rsquared))
```

```
##   model      RMSE  Rsquared        MAE
## 1    RF 0.1010464 0.6635909 0.07272722
## 2  TREE 0.1084296 0.5966751 0.07620956
## 3   SVM 0.1221341 0.4884199 0.08880445
## 4  MARS 0.1288406 0.4348613 0.09654309
## 5   PLS 0.1312256 0.4068016 0.09967673
## 6    LM 0.1312526 0.4066680 0.09944526
## 7  CART 0.1344379 0.3803752 0.10593006
```

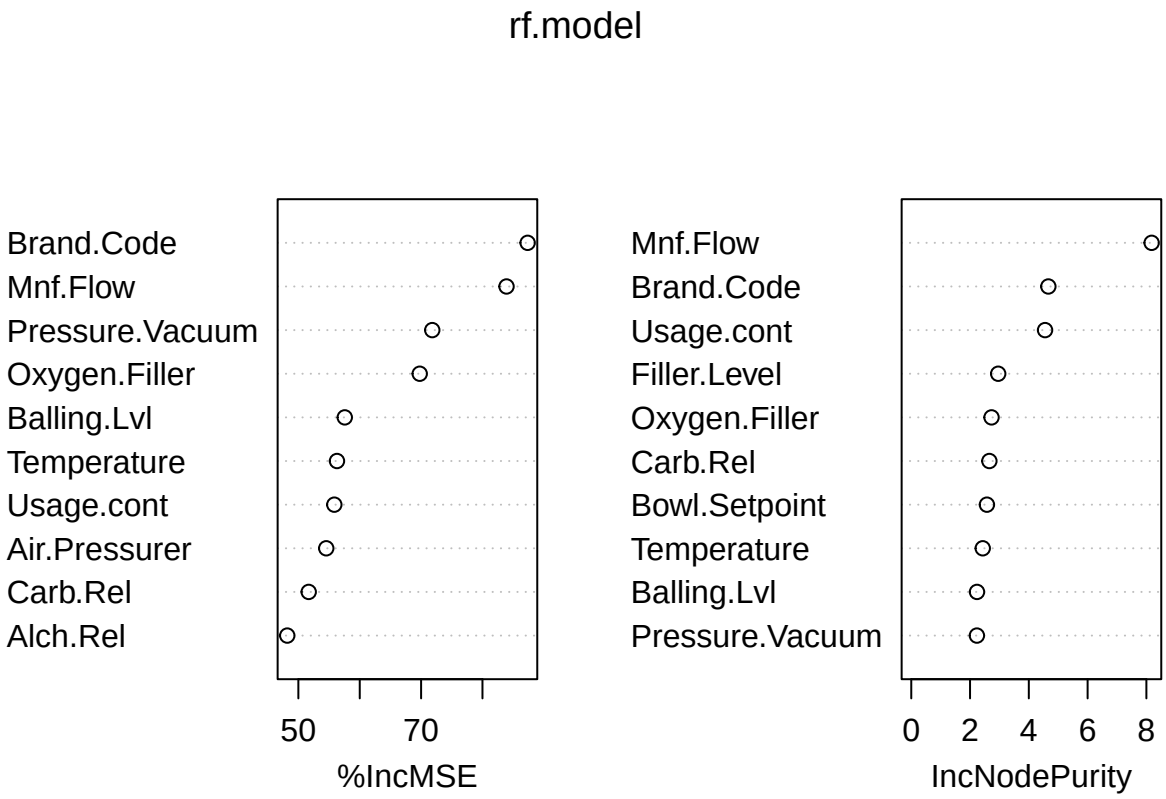
The Random Forest Model has the highest $R^2$ value and will be chosen to model the `PH` values.

# Model Evaluation

## Predictor Importance

Looking at the top 10 predictors we see that `Brand.Code` and `Mnf.Flow` are the two most important predictors for determining the `PH`.

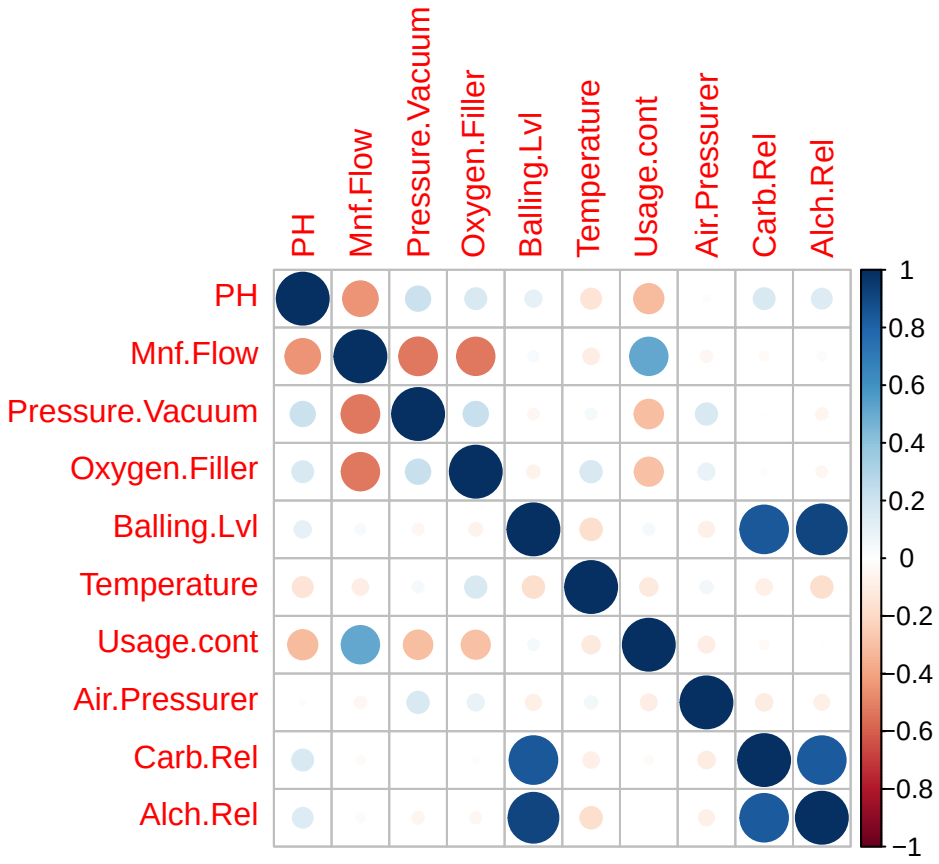
```
top10 <- varImp(rf.model) |>
  arrange(desc(Overall)) |>
  head(10)

# Plotting the variable importance
varImpPlot(rf.model, n.var = 10)
```



### Correlation for Important Predictors

Now that we know which predictors are most important for determining the `PH` of the beverage, we can look at the correlation matrix to see how they relate.

```
impute.df |>
  select(c('PH', row.names(top10))) |>
  keep(is.numeric) |>
  cor() |>
  corrplot()
```




## Forecast PH

### Impute Missing Values

We will impute missing values in the evaluation data set using the same method as the training set. Again we will remove `Hyd.Pressure` because it has near-zero variance.

```
set.seed(200)

eval.impute <- eval.df |>
  select(-PH) |>
  mice(m = 1, method = 'pmm', print = FALSE) |>
  complete() |>
```

```
  mutate(PH = '') |>
  select(-Hyd.Pressure1)
```

## Predict PH

Using the Random Forest Model, we will predict the `PH` in the evaluation data set.

```
eval.pred <- predict(rf.model, newdata = eval.impute)
head(eval.pred, 10)
```

```
##        1        2        3        4        5        6        7        8
## 8.575898 8.468623 8.539325 8.603994 8.519567 8.537066 8.494102 8.572170
##        9       10
## 8.553924 8.629252
```

## Create Output

Insert the computed `PH` into the original evaluation data set and export to Excel.

```
pred.df <- eval.df |>
  mutate(PH = eval.pred)

pred.df |>
  write.xlsx('StudentEvaluationPreds.xlsx')
```