DATA624 - Project 2

Glen Dale Davis & Tora Mullings

2023-12-16

Packages:

```
library(tidyverse)
library(httr)
library(readxl)
library(DataExplorer)
library(psych)
library(knitr)
library(snakecase)
library(RColorBrewer)
library(VIM)
library(ggcorrplot)
library(caret)
library(randomForest)
library(cowplot)
library(car)
library(MASS)
select <- dplyr::select
library(earth)
```

```
cur_theme <- theme_set(theme_classic())
palette <- brewer.pal(n = 12, name = "Paired")
greys <- brewer.pal(n = 9, name = "Greys")
```

Introduction:

New regulations require ABC Beverage to understand our manufacturing process and the predictive factors. We need to be able to report to leadership our predictive model of `PH`.

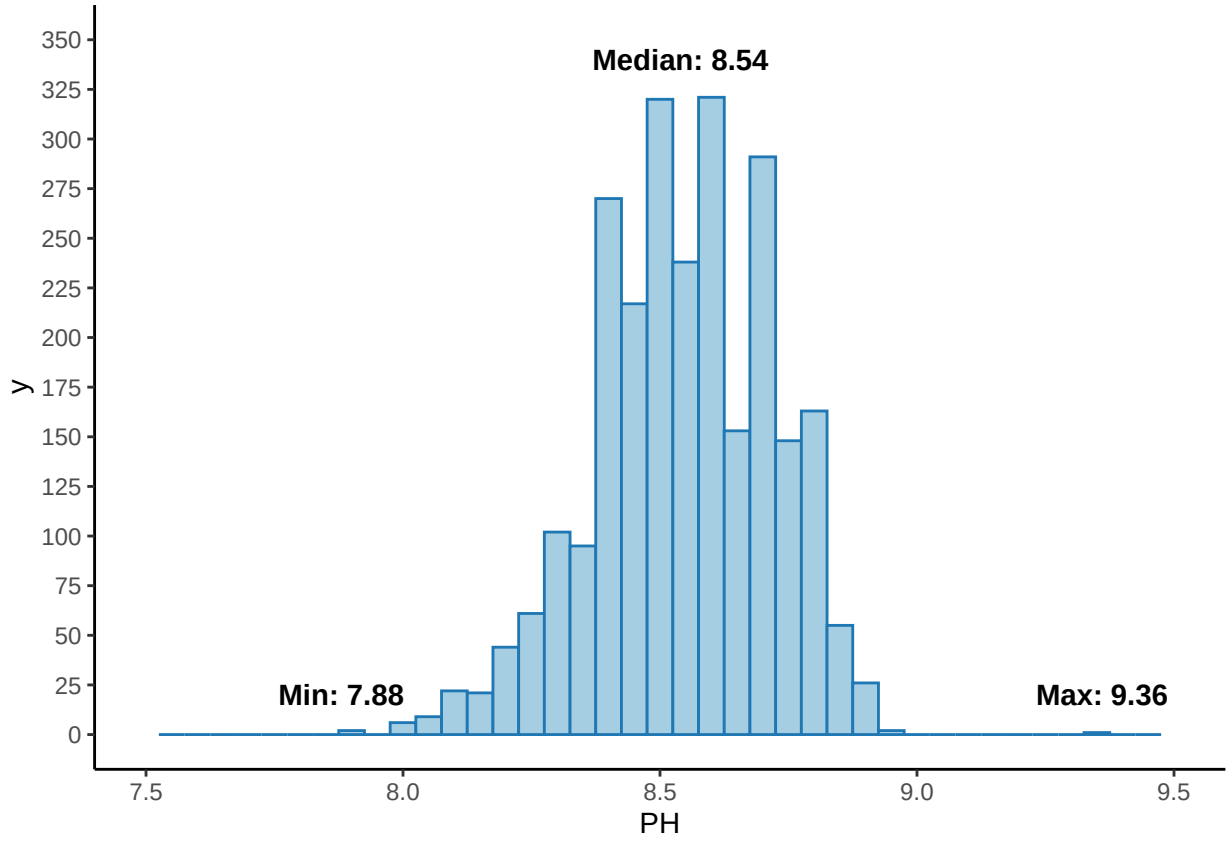
We load the historical dataset provided.

```
my_url1 <- "https://github.com/geedoubledee/data624_project2/raw/main/StudentData.xlsx"
temp <- tempfile(fileext = ".xlsx")
req <- GET(my_url1, authenticate(Sys.getenv("GITHUB_PAT"), ""),
           write_disk(path = temp))
main_df <- readxl::read_excel(temp)
colnames(main_df) <- to_screaming_snake_case(colnames(main_df))
```

Exploratory Data Analysis:

We take a look at the distribution for the response variable and a summary of it.

```
annotations <- data.frame(x = c(min(main_df$PH, na.rm = TRUE),
                                round(median(main_df$PH, na.rm = TRUE), 2),
                                max(main_df$PH, na.rm = TRUE)),
                          y = c(20, 340, 20),
                          label = c("Min:", "Median:", "Max:"))
p0 <- main_df |>
    ggplot(aes(x = PH)) +
    geom_histogram(binwidth = 0.05, color = palette[2], fill = palette[1]) +
    geom_text(data = annotations,
              aes(x = x, y = y, label = paste(label, x)),
              size = 4, fontface = "bold") +
    scale_x_continuous(limits = c(7.5, 9.5), breaks = seq(7.5, 9.5, 0.5)) +
    scale_y_continuous(limits = c(0, 350), breaks = seq(0, 350, 25))
p0
```




```
summary(main_df$PH)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   7.880   8.440   8.540   8.546   8.680   9.360       4
```

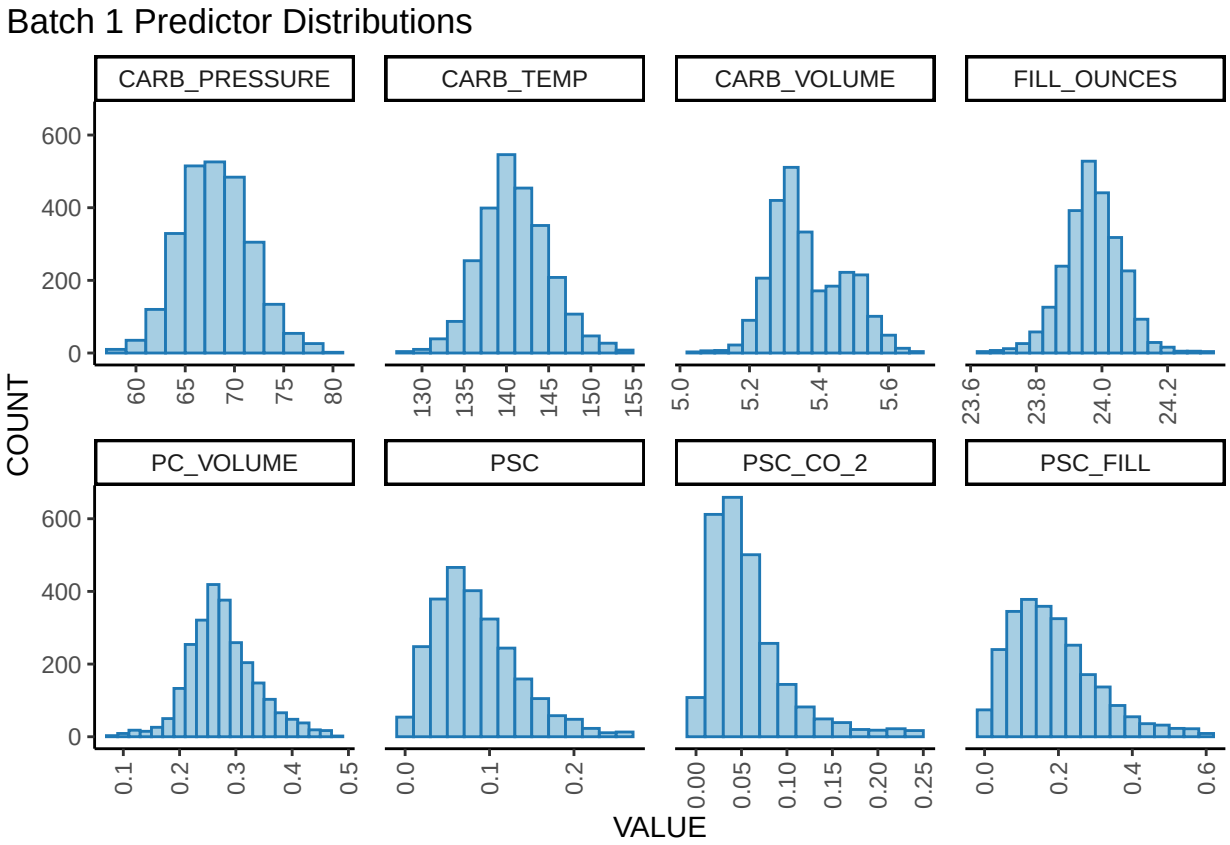
The median PH value is 8.54 and ranges between 7.88 and 9.36. 50 percent of observations have values between 8.44 and 8.68 though. There are 4 observations with missing PH values. This is a small enough

percentage of our total observations to justify simple list-wise deletion. We lose little by removing these observations, and we would gain little by imputing them.

```
main_df <- main_df |>
    filter(!is.na(PH))
```

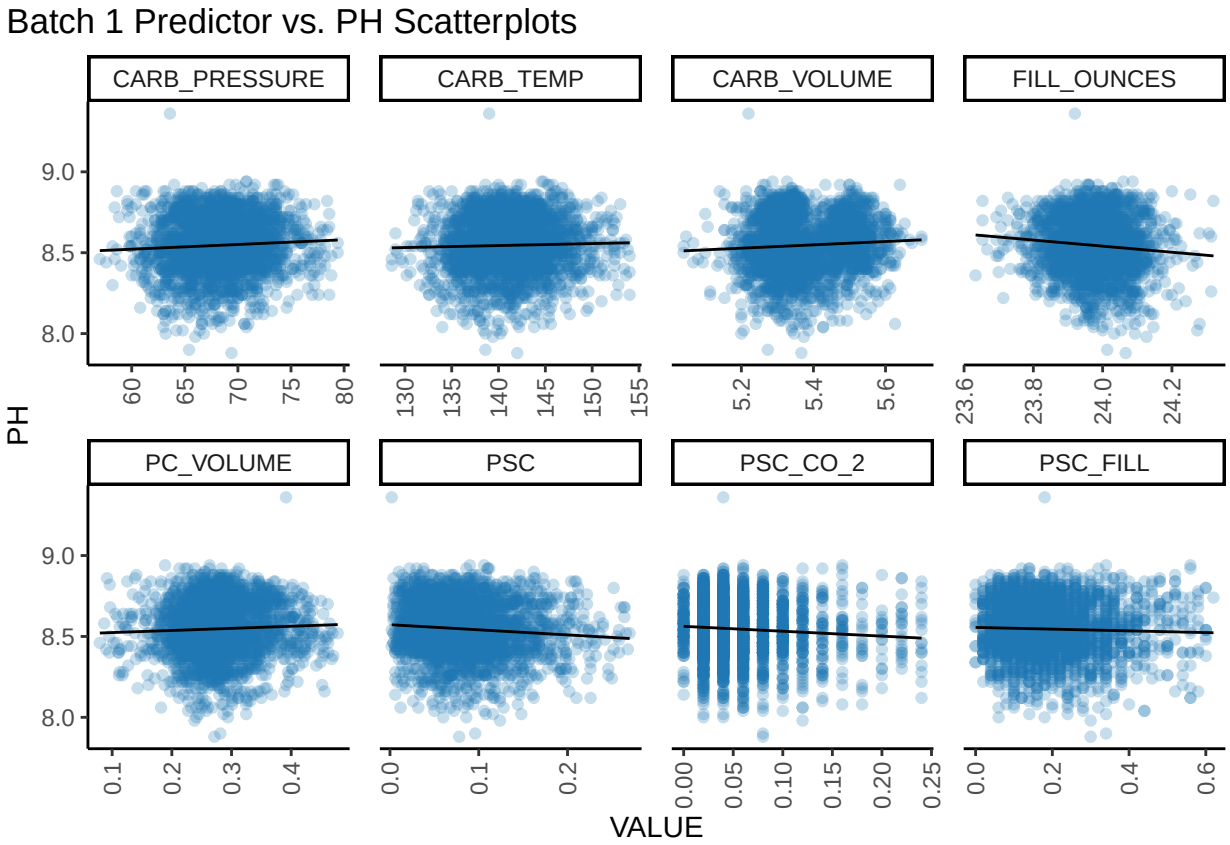
We take a look at histograms for the numeric predictor variables, as well scatterplots of each numeric predictor and the response, in batches since there are so many of them.

```
non_numeric <- c("BRAND_CODE")
all_numeric <- colnames(main_df |> select(-all_of(c("PH", non_numeric))))
n = 8
all_numeric_chunks <- split(all_numeric, ceiling(seq_along(all_numeric)/n))
remove <- c("PH", non_numeric)
pivot_df <- main_df |>
    select(-all_of(remove)) |>
    pivot_longer(cols = all_of(all_numeric), names_to = "PREDICTOR",
                 values_to = "VALUE")
p1a <- pivot_df |>
    filter(PREDICTOR %in% all_numeric_chunks[[1]]) |>
    ggplot(aes(x = VALUE)) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "CARB_PRESSURE"),
                   binwidth = 2) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "CARB_TEMP"),
                   binwidth = 2) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "CARB_VOLUME"),
                   binwidth = 0.04) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "FILL_OUNCES"),
                   binwidth = 0.04) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "PC_VOLUME"),
                   binwidth = 0.02) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "PSC"),
                   binwidth = 0.02) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "PSC_CO_2"),
                   binwidth = 0.02) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "PSC_FILL"),
                   binwidth = 0.04) +
    facet_wrap(vars(PREDICTOR), ncol = 4, scales = "free_x") +
    labs(y = "COUNT",
         title = "Batch 1 Predictor Distributions") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p1a
```



In the first batch of numeric predictors, we see that `PSC`, `PSC_CO_2`, and `PSC_FILL` are all right-skewed, and the distribution for `CARB_VOLUME` is multimodal. The distributions for the rest of the variables are nearly normal.

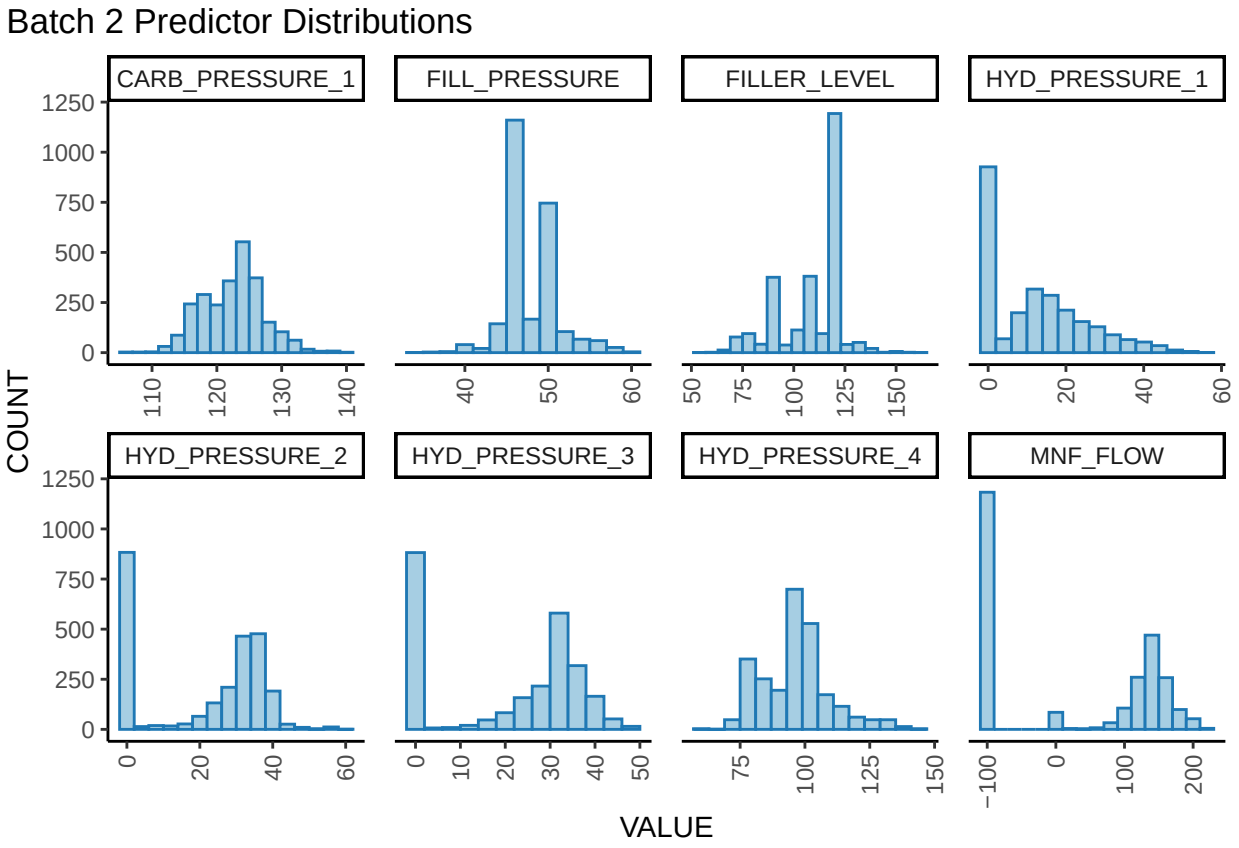
```
sel <- c("PH", all_numeric_chunks[[1]])
p1b <- main_df |>
    select(all_of(sel)) |>
    pivot_longer(cols = all_of(all_numeric_chunks[[1]]), names_to = "PREDICTOR",
                 values_to = "VALUE") |>
    ggplot(aes(x = VALUE, y = PH)) +
    geom_point(color = palette[2], fill = palette[1], alpha = 0.25) +
    geom_smooth(method = "lm", color = "black", linewidth = 0.5, se = FALSE) +
    facet_wrap(~PREDICTOR, ncol = 4, scales = "free_x") +
    labs(title = "Batch 1 Predictor vs. PH Scatterplots") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p1b
```



There are no linear relationships discernable from these scatterplots. There may be two clusters in `CARB_VOLUME`.

```
p2a <- pivot_df |>
    filter(PREDICTOR %in% all_numeric_chunks[[2]]) |>
    ggplot(aes(x = VALUE)) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "CARB_PRESSURE_1"),
                   binwidth = 2) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "FILL_PRESSURE"),
                   binwidth = 2) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "FILLER_LEVEL"),
                   binwidth = 6) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "HYD_PRESSURE_1"),
                   binwidth = 4) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "HYD_PRESSURE_2"),
                   binwidth = 4) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "HYD_PRESSURE_3"),
                   binwidth = 4) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "HYD_PRESSURE_4"),
```

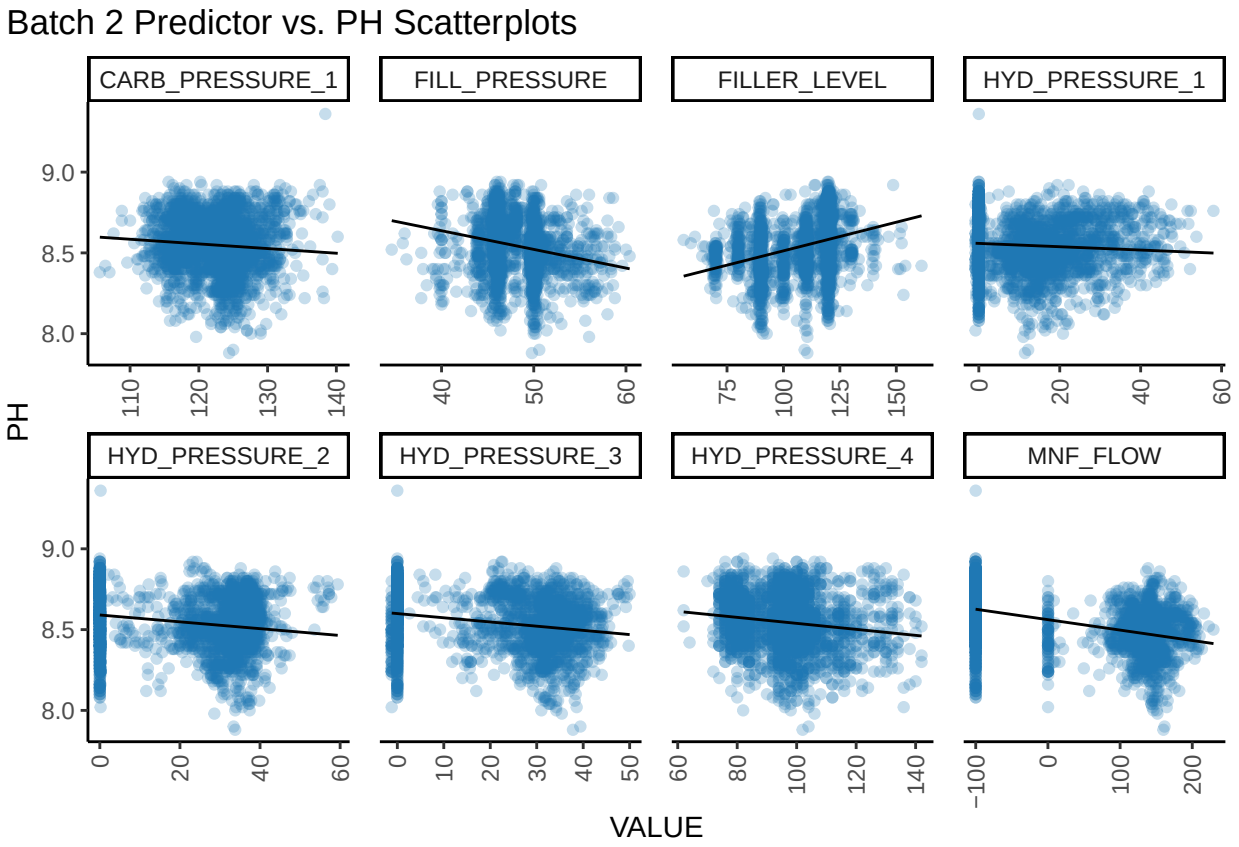
```
                  binwidth = 6) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "MNF_FLOW"),
                   binwidth = 20) +
    facet_wrap(vars(PREDICTOR), ncol = 4, scales = "free_x") +
    labs(y = "COUNT",
         title = "Batch 2 Predictor Distributions") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p2a
```



In the second batch of numeric predictors, we see that `HYD_PRESSURE_1`, `HYD_PRESSURE_2`, and `HYD_PRESSURE_3` are heavy with zero value observations, skewing their distributions. Most observations for `MNF_FLOW` are around -100, and its distribution might be degenerate. We'll check for degeneracy for this variable and any others shortly. `FILL_PRESSURE` and `FILLER_LEVEL` are multimodal. `HYD_PRESSURE_4` is right-skewed. `CARB_PRESSURE_1` has the only nearly normal distribution here.

```
sel <- c("PH", all_numeric_chunks[[2]])
p2b <- main_df |>
    select(all_of(sel)) |>
    pivot_longer(cols = all_of(all_numeric_chunks[[2]]), names_to = "PREDICTOR",
                 values_to = "VALUE") |>
    ggplot(aes(x = VALUE, y = PH)) +
    geom_point(color = palette[2], fill = palette[1], alpha = 0.25) +
    geom_smooth(method = "lm", color = "black", linewidth = 0.5, se = FALSE) +
```

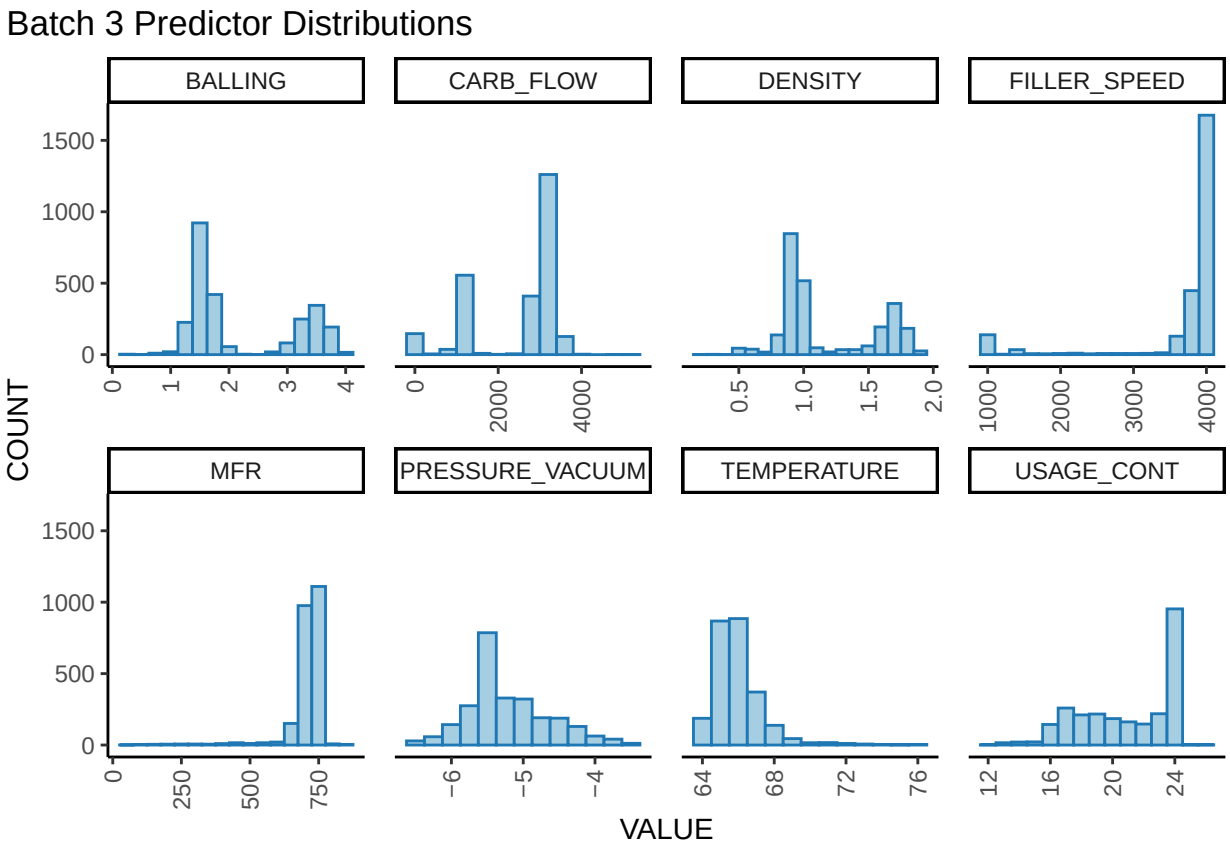
```
    facet_wrap(~PREDICTOR, ncol = 4, scales = "free_x") +
    labs(title = "Batch 2 Predictor vs. PH Scatterplots") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p2b
```



Again, linear relationships are hard to discern from these scatterplots, but there may be a negative relationship between `FILL_PRESSURE` and `PH` and a positive relationship between `FILLER_LEVEL` and `PH`. There's clustering in both variables.

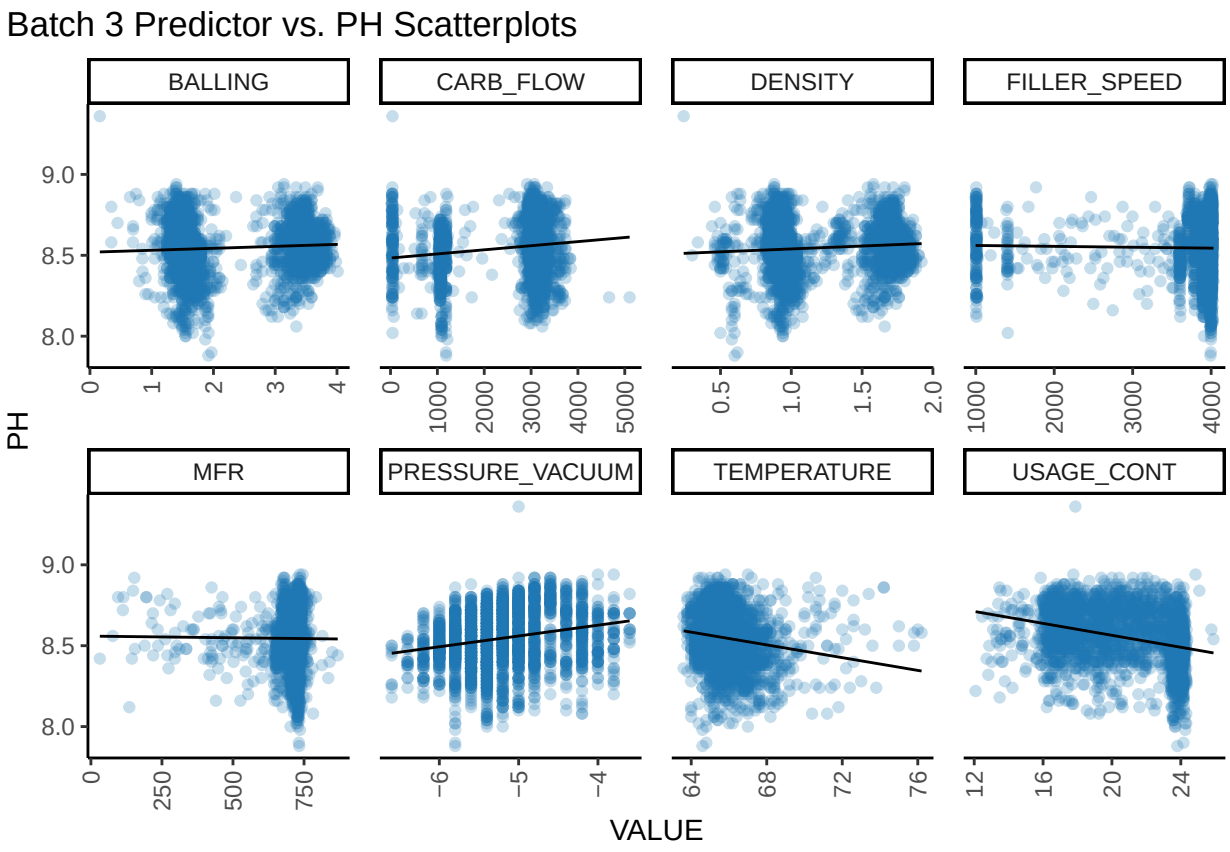
```
p3a <- pivot_df |>
    filter(PREDICTOR %in% all_numeric_chunks[[3]]) |>
    ggplot(aes(x = VALUE)) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "BALLING"),
                   binwidth = 0.25) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "CARB_FLOW"),
                   binwidth = 400) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "DENSITY"),
                   binwidth = 0.1) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "FILLER_SPEED"),
                   binwidth = 200) +
```

```
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "MFR"),
                   binwidth = 50) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "PRESSURE_VACUUM"),
                   binwidth = 0.25) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "TEMPERATURE"),
                   binwidth = 1) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "USAGE_CONT"),
                   binwidth = 1) +
    facet_wrap(vars(PREDICTOR), ncol = 4, scales = "free_x") +
    labs(y = "COUNT",
         title = "Batch 3 Predictor Distributions") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p3a
```



In the third batch of numeric predictors, we see multimodal distributions for `BALLING`, `CARB_FLOW`, and `DENSITY`. `FILLER_SPEED`, `MFR`, and `USAGE_CONT` are left-skewed, and `TEMPERATURE` and `PRESSURE_VACUUM` are right-skewed.

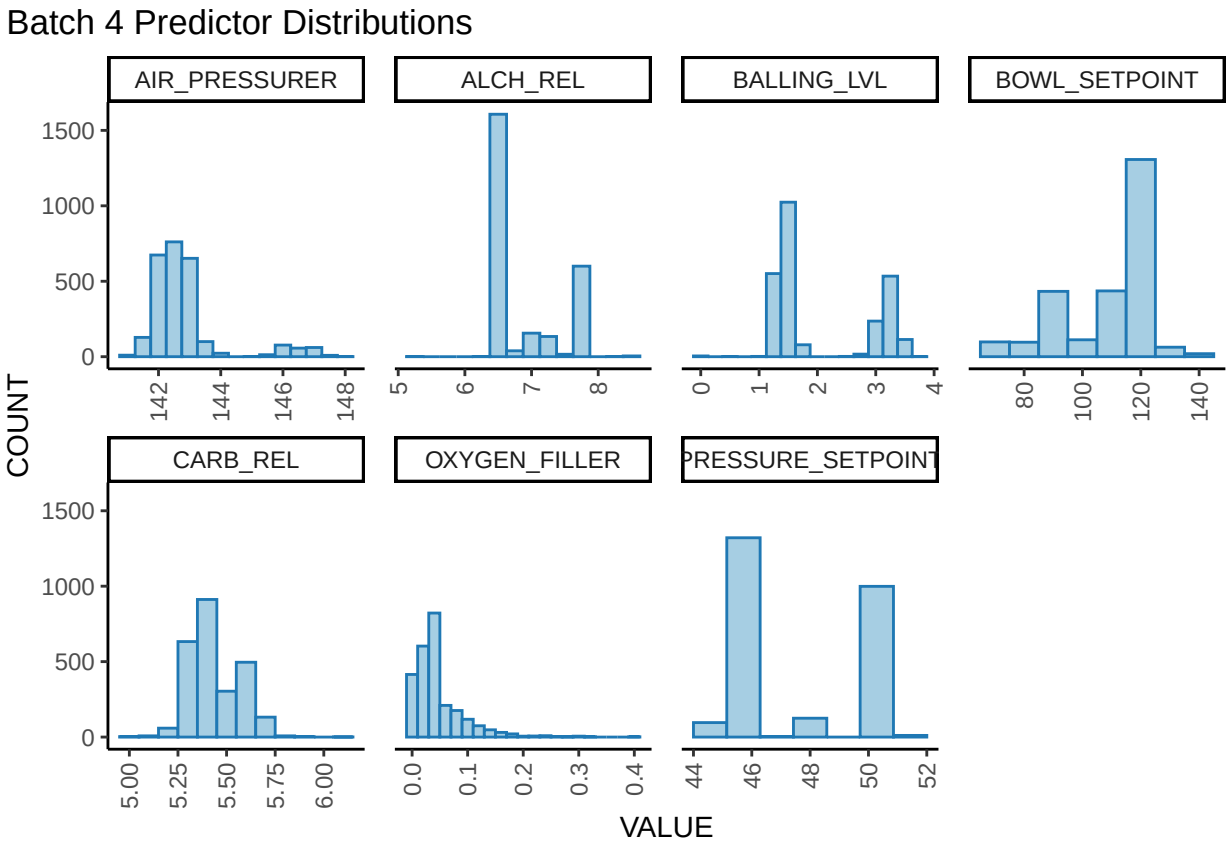
```
sel <- c("PH", all_numeric_chunks[[3]])
p3b <- main_df |>
    select(all_of(sel)) |>
    pivot_longer(cols = all_of(all_numeric_chunks[[3]]), names_to = "PREDICTOR",
                 values_to = "VALUE") |>
    ggplot(aes(x = VALUE, y = PH)) +
    geom_point(color = palette[2], fill = palette[1], alpha = 0.25) +
    geom_smooth(method = "lm", color = "black", linewidth = 0.5, se = FALSE) +
    facet_wrap(~PREDICTOR, ncol = 4, scales = "free_x") +
    labs(title = "Batch 3 Predictor vs. PH Scatterplots") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p3b
```



We see clustering in `BALLING`, `CARB_FLOW`, and `DENSITY`. There may be a somewhat positive relationship between `PRESSURE_VACUUM` and `PH`, as well as a somewhat negative relationship between `TEMPERATURE` and `PH`.

```
p4a <- pivot_df |>
    filter(PREDICTOR %in% all_numeric_chunks[[4]]) |>
    ggplot(aes(x = VALUE)) +
    geom_histogram(color = palette[2], fill = palette[1],
                   data = subset(pivot_df, PREDICTOR == "AIR_PRESSURER"),
                   binwidth = 0.5) +
    geom_histogram(color = palette[2], fill = palette[1],
```

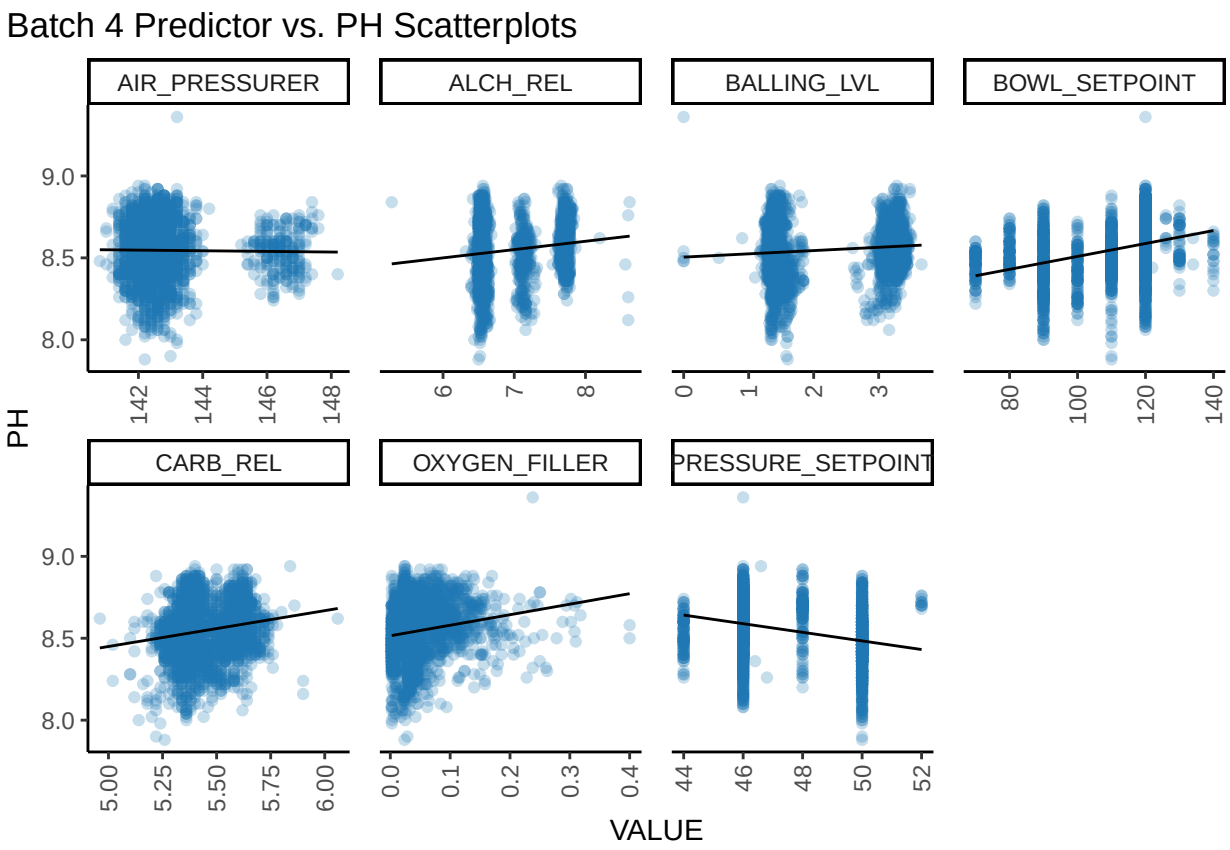
```
                  data = subset(pivot_df, PREDICTOR == "ALCH_REL"),
                  binwidth = 0.25) +
    geom_histogram(color = palette[2], fill = palette[1],
                  data = subset(pivot_df, PREDICTOR == "BALLING_LVL"),
                  binwidth = 0.25) +
    geom_histogram(color = palette[2], fill = palette[1],
                  data = subset(pivot_df, PREDICTOR == "BOWL_SETPOINT"),
                  binwidth = 10) +
    geom_histogram(color = palette[2], fill = palette[1],
                  data = subset(pivot_df, PREDICTOR == "CARB_REL"),
                  binwidth = 0.1) +
    geom_histogram(color = palette[2], fill = palette[1],
                  data = subset(pivot_df, PREDICTOR == "OXYGEN_FILLER"),
                  binwidth = 0.02) +
    geom_histogram(color = palette[2], fill = palette[1],
                  data = subset(pivot_df, PREDICTOR == "PRESSURE_SETPOINT"),
                  bins = 8) +
    facet_wrap(vars(PREDICTOR), ncol = 4, scales = "free_x") +
    labs(y = "COUNT",
         title = "Batch 4 Predictor Distributions") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p4a
```



In the last batch of numeric predictors, we see that `AIR_PRESSURER` and `OXYGEN_FILLER` are right-skewed.

The distributions for ALCH_REL, BALLING_LVL, and PRESSURE_SETPOINT are multimodal. BOWL_SETPOINT is left-skewed. CARB_REL is the only variable for which the distribution is nearly normal.

```
sel <- c("PH", all_numeric_chunks[[4]])
p4b <- main_df |>
    select(all_of(sel)) |>
    pivot_longer(cols = all_of(all_numeric_chunks[[4]]), names_to = "PREDICTOR",
                 values_to = "VALUE") |>
    ggplot(aes(x = VALUE, y = PH)) +
    geom_point(color = palette[2], fill = palette[1], alpha = 0.25) +
    geom_smooth(method = "lm", color = "black", linewidth = 0.5, se = FALSE) +
    facet_wrap(~PREDICTOR, ncol = 4, scales = "free_x") +
    labs(title = "Batch 4 Predictor vs. PH Scatterplots") +
    theme(panel.spacing.x = unit(4, "mm"),
          axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
          plot.title.position = "plot")
p4b
```



We see clustering in AIR_PRESSURER, ALCH_REL, and BALLING_LVL.

Summary statistics for all numeric predictors are below.

```
remove <- c("n", "vars", "trimmed", "mad", "range", "se")
describe <- main_df |>
    select(all_of(all_numeric)) |>
    describe() |>
```

```
    select(-all_of(remove))
knitr::kable(describe, format = "simple")
```

	mean	sd	median	min	max	skew	kurt
CARB_VOLUME	5.3703337	0.1063981	5.3466667	5.0400000	5.700	0.3904059	-0.4682
FILL_OUNCES	23.9749176	0.0874663	23.9733333	23.6333333	24.320	-0.0215410	0.8685
PC_VOLUME	0.2772392	0.0605992	0.2713333	0.0793333	0.478	0.3468176	0.6690
CARB_PRESSURE	68.1902677	3.5386086	68.2000000	57.0000000	79.400	0.1811752	-0.0125
CARB_TEMP	141.0922393	4.0340631	140.8000000	128.6000000	154.000	0.2427101	0.2342
PSC	0.0846433	0.0492487	0.0760000	0.0020000	0.270	0.8504528	0.6503
PSC_FILL	0.1952987	0.1177889	0.1800000	0.0000000	0.620	0.9352821	0.7744
PSC_CO_2	0.0564399	0.0430641	0.0400000	0.0000000	0.240	1.7270393	3.7153
MNF_FLOW	24.6269575	119.5013986	70.2000000	-100.2000000	229.400	0.0031327	-1.8700
CARB_PRESSURE_1	122.5704142	4.7272264	123.2000000	105.6000000	140.200	0.0429942	0.1250
FILL_PRESSURE	47.9221656	3.1775457	46.4000000	34.6000000	60.400	0.5471107	1.4067
HYD_PRESSURE_1	12.4571987	12.4330687	11.4000000	-0.8000000	58.000	0.7779346	-0.1445
HYD_PRESSURE_2	20.9935737	16.3784943	28.6000000	0.0000000	59.400	-0.3056277	-1.5559
HYD_PRESSURE_3	20.4778997	15.9714047	27.6000000	-1.2000000	50.000	-0.3210114	-1.5724
HYD_PRESSURE_4	96.3087830	13.0976498	96.0000000	62.0000000	142.000	0.5602427	0.6096
FILLER_LEVEL	109.2523716	15.6984241	118.4000000	55.8000000	161.200	-0.8482847	0.0460
FILLER_SPEED	3688.1066454	769.6282261	3982.0000000	998.0000000	4030.000	-2.8777117	6.7529
TEMPERATURE	65.9648532	1.3790586	65.6000000	63.6000000	76.200	2.3920389	10.2510
USAGE_CONT	20.9942155	2.9761958	21.7900000	12.0800000	25.900	-0.5351830	-1.0167
CARB_FLOW	2472.0530214	1070.4281545	3030.0000000	26.0000000	5104.000	-0.9916636	-0.5735
DENSITY	1.1744527	0.3769684	0.9800000	0.2400000	1.920	0.5311125	-1.2095
MFR	704.0492582	73.8983094	724.0000000	31.4000000	868.600	-5.0917729	30.4558
BALLING	2.1998418	0.9295470	1.6480000	0.1600000	4.012	0.6004592	-1.3996
PRESSURE_VACUUM	-5.2162057	0.5703665	-5.4000000	-6.6000000	-3.600	0.5258505	-0.0341
OXYGEN_FILLER	0.0464281	0.0450729	0.0334000	0.0024000	0.400	2.4147972	8.8407
BOWL_SETPOINT	109.3450292	15.2891482	120.0000000	70.0000000	140.000	-0.9749161	-0.0544
PRESSURE_SETPOINT	47.6132290	2.0387546	46.0000000	44.0000000	52.000	0.2051072	-1.5997
AIR_PRESSURER	142.8339696	1.2127148	142.6000000	140.8000000	148.200	2.2512354	4.7254
ALCH_REL	6.8978125	0.5052561	6.5600000	5.2800000	8.620	0.8830750	-0.8526
CARB_REL	5.4367956	0.1287629	5.4000000	4.9600000	6.060	0.5028431	-0.2968
BALLING_LVL	2.0516212	0.8688888	1.4800000	0.0000000	3.660	0.5943424	-1.5024

Next we examine the dataset's completeness.

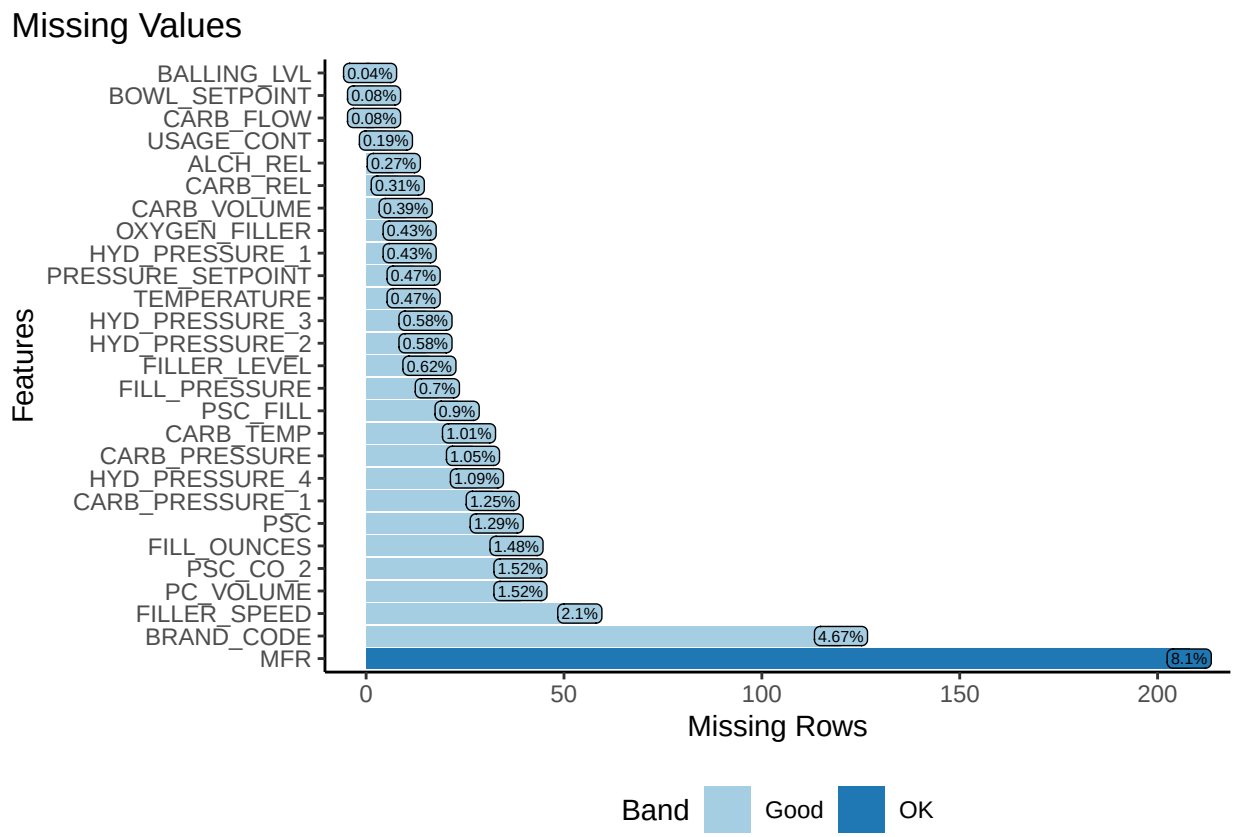
```
remove <- c("discrete_columns", "continuous_columns", "total_observations",
            "memory_usage")
introduce <- main_df |>
    introduce() |>
    select(-all_of(remove))
knitr::kable(t(introduce), format = "simple")
```

rows	2567
columns	33
all_missing_columns	0
total_missing_values	812
complete_rows	2038

Only 2,038 out of 2,571 rows are complete, which is about 79 percent of observations. There are 844 missing values. None of our variables are completely `NA`.

We take a closer look at where the missing values are.

```
p5 <- p5 +
    scale_fill_brewer(palette = "Paired") +
    theme(plot.title.position = "plot")
p5
```

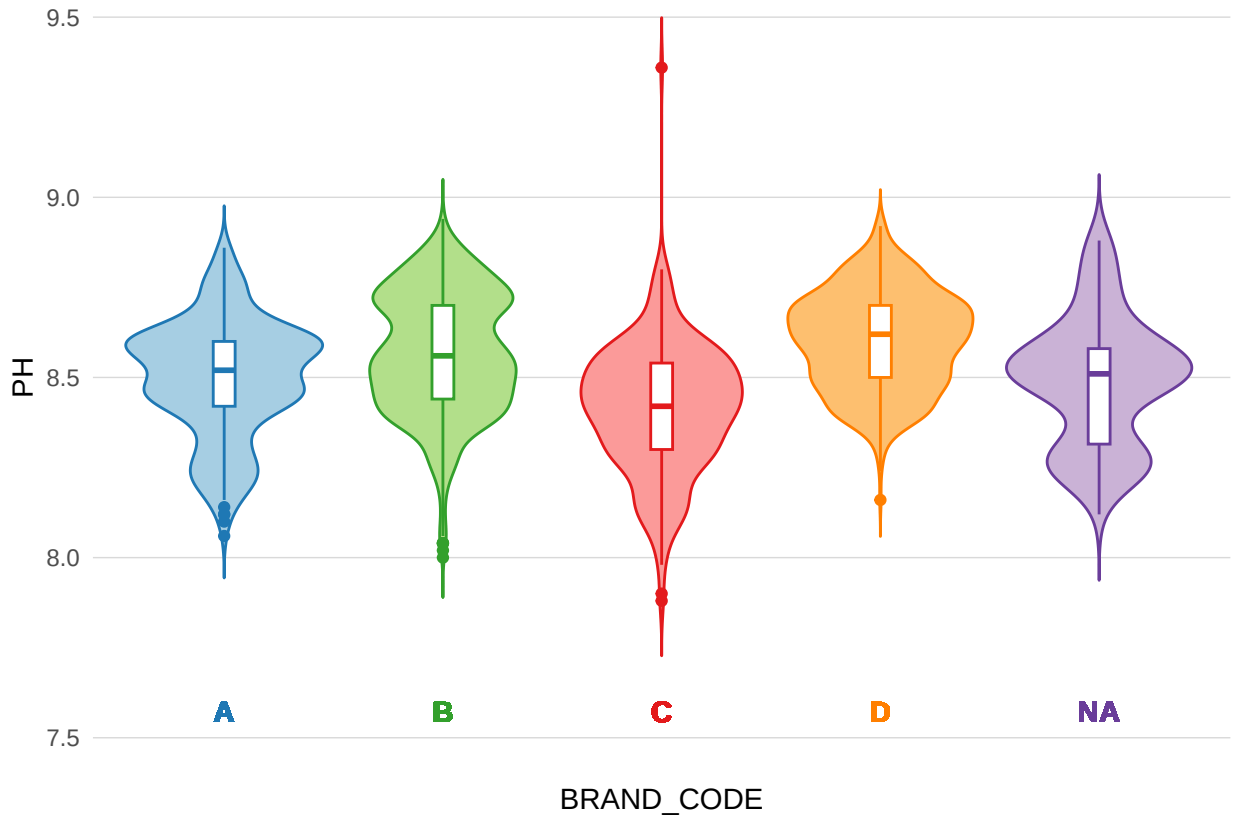


`MFR`, `BRAND_CODE`, and `FILLER_SPEED` are the predictors with the most missing values, but many other predictors are missing values as well. We coerce `BRAND_CODE` to a factor and add a level for `NA` values to handle missingness for this categorical predictor. Then we look at the distribution of `PH` by `BRAND_CODE` level to determine whether there are differences in variation between groups and outliers within groups.

```
main_df <- main_df |>
    mutate(BRAND_CODE = factor(BRAND_CODE, exclude = NULL))
palette <- brewer.pal(n = 12, name = "Paired")
col <- palette[c(2, 4, 6, 8, 10)]
fil <- palette[c(1, 3, 5, 7, 9)]
p6 <- main_df |>
    ggplot(aes(x = BRAND_CODE, y = PH, color = BRAND_CODE, fill = BRAND_CODE)) +
    geom_violin(trim = FALSE) +
    geom_boxplot(width = 0.1, fill = "white") +
    geom_text(aes(label = BRAND_CODE, color = BRAND_CODE), y = 7.5,
              vjust = -0.75, size = 4, fontface = "bold") +
```

```
    scale_y_continuous(limits = c(7.5, 9.5), breaks = seq(7.5, 9.5, 0.5)) +
    scale_color_manual(values = col) +
    scale_fill_manual(values = fil) +
    theme(legend.position = "none",
          axis.ticks = element_blank(),
          axis.text.x = element_blank(),
          axis.line = element_blank(),
          panel.grid.major.y = element_line(color = greys[3], linewidth = 0.25,
                                            linetype = 1))
p6
```

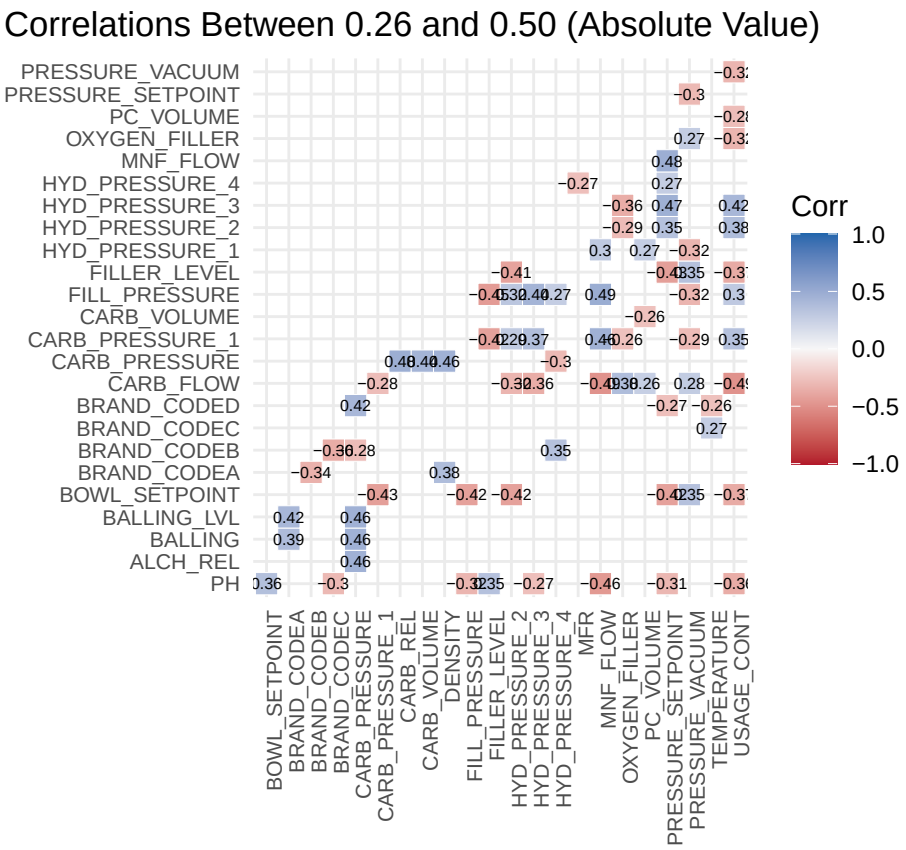
```
## Warning: Removed 4 rows containing missing values (`geom_violin()`).
```



Level "D" has the highest median PH, level "C" has the only outlier on the high end, and all levels except the level representing NA values have outliers on the low end. Level "A" has the narrowest IQR, whereas as the level representing NA values has the widest IQR.

We will perform KNN imputation for the numeric predictors with missing data. We will also create a secondary version of the data where we perform list-wise deletion instead. Before we handle this remaining missing data, we first look at correlations between our predictors and the response variable. Because we have so many variables, it would be difficult to visualize all correlations at the same time without binning them. So we will bin absolute value correlations into four groups: 1) 0.00 to 0.25, 2) 0.26 to 0.50, 3) 0.51 to 0.75, and 4) 0.76 to 1.00. We won't visualize any correlations less than 0.26, but note that this doesn't imply those correlations are insignificant. Limiting what we examine most closely will simply help us hone in on a) the predictor variables we should expect any good model we develop to include and b) the predictor variables that are so highly correlated with one another that they could inhibit certain models' performance.

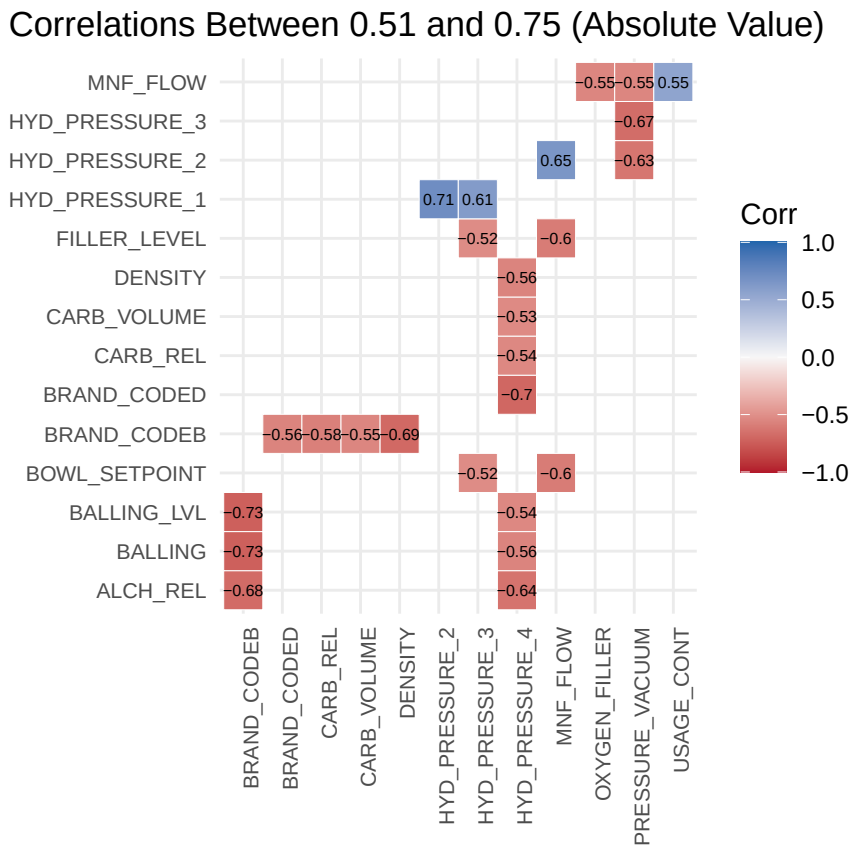
```
incl <- c("PH", sort(colnames(main_df |> select(-PH))))
palette <- brewer.pal(n = 7, name = "RdBu")[c(1, 4, 7)]
r <- model.matrix(~0+., data = main_df |> select(all_of(incl))) |>
    cor(use = "pairwise.complete.obs")
is.na(r) <- abs(r) > 0.5
is.na(r) <- abs(r) < 0.26
p7 <- r |>
    ggcorrplot(show.diag = FALSE, type = "lower", lab = TRUE, lab_size = 2,
               tl.cex = 8, tl.srt = 90,
               colors = palette, outline.color = "white") +
    labs(title = "Correlations Between 0.26 and 0.50 (Absolute Value)") +
    theme(plot.title.position = "plot")
p7
```



Here, we see the predictors that are most positively correlated with `PH` are `BOWL_SETPOINT` and `FILLER_LEVEL`, and the predictors that are most negatively correlated with `PH` are `BRAND_CODE` level "C", `FILL_PRESSURE`, `HYD_PRESSURE_3`, `MNF_FLOW`, `PRESSURE_SETPOINT`, and `USAGE_CONT`. While some of the predictors in this plot are moderately correlated with each other, we will focus on higher/more worrisome predictor-predictor correlation levels in the following two plots.

```
r <- model.matrix(~0+., data = main_df |> select(all_of(incl))) |>
    cor(use = "pairwise.complete.obs")
is.na(r) <- abs(r) > 0.75
is.na(r) <- abs(r) < 0.51
p8 <- r |>
    ggcorrplot(show.diag = FALSE, type = "lower", lab = TRUE, lab_size = 2,
```

```
                tl.cex = 8, tl.srt = 90,
                colors = palette, outline.color = "white") +
    labs(title = "Correlations Between 0.51 and 0.75 (Absolute Value)") +
    theme(plot.title.position = "plot")
p8
```



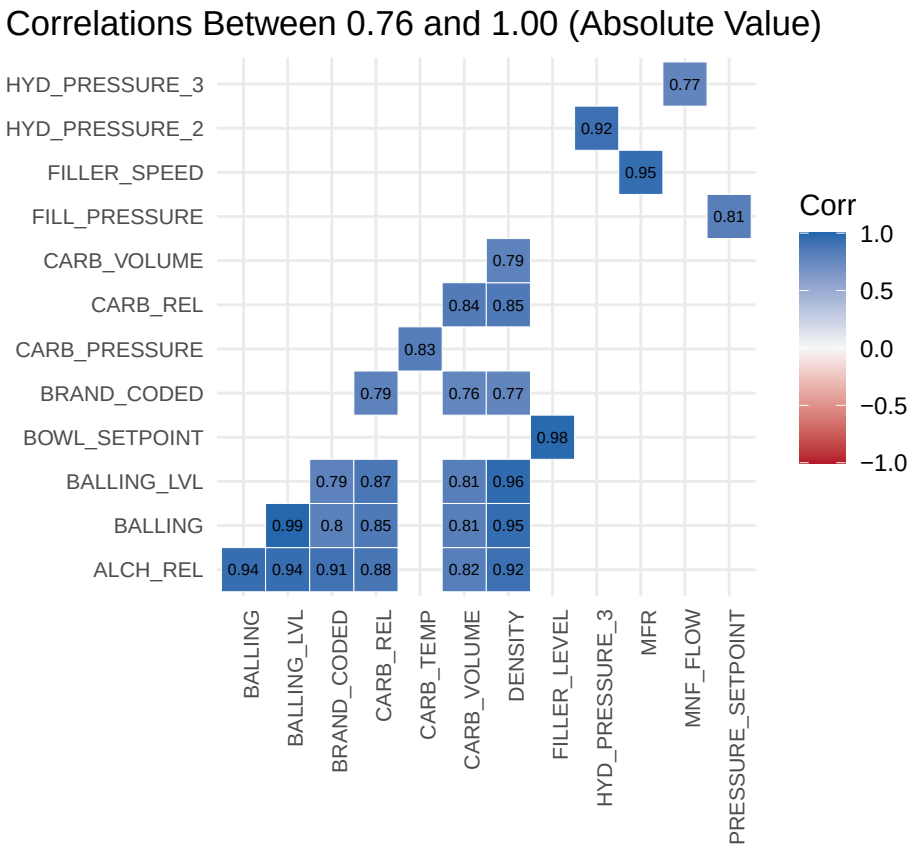
Here, we notice immediately that `PH` is missing from the plot and is therefore not correlated with any predictors at a level between 0.51 and 0.75 in absolute value. Although we could comment on all these correlation levels, we see high (> 0.6) positive predictor-predictor correlations between:

- `MNF_FLOW` and `HYD_PRESSURE_2`
- `HYD_PRESSURE_3` and `HYD_PRESSURE_2`
- `HYD_PRESSURE_2` and `HYD_PRESSURE_1`

We also see high (< -0.6) negative predictor-predictor correlations between:

- `PRESSURE_VACUUM` and `HYD_PRESSURE_3`/`HYD_PRESSURE_2`
- `HYD_PRESSURE_4` and `BRAND_CODE` level "D"/`ALCH_REL`
- `BRAND_CODE` level "B" and `DENSITY`/`BALLING_LVL`/`BALLING`/`ALCH_REL`

```
r <- model.matrix(~0+., data = main_df |> select(all_of(incl))) |>
    cor(use = "pairwise.complete.obs")
is.na(r) <- abs(r) < 0.76
p9 <- r |>
    ggcorrplot(show.diag = FALSE, type = "lower", lab = TRUE, lab_size = 2,
               tl.cex = 8, tl.srt = 90,
               colors = palette, outline.color = "white") +
    labs(title = "Correlations Between 0.76 and 1.00 (Absolute Value)") +
    theme(plot.title.position = "plot")
p9
```



PH is again missing, so it is therefore not correlated with any predictors at a level between 0.76 and 1.00 in absolute value. Although we could again comment on all these correlation levels, we see extremely high (> 0.9) positive predictor-predictor correlations between

- MFR and FILLER_SPEED
- HYD_PRESSURE_3 and HYD_PRESSURE_2
- FILLER_LEVEL and BOWL_SETPOINT
- DENSITY and BALLING_LVL/BALLING/ALCH_REL
- BRAND_CODE level "D" and ALCH_REL
- BALLING_LVL and BALLING/ALCH_REL
- BALLING and ALCH_REL

There are no extremely high (< -0.9) negative predictor-predictor correlations.

When creating models that are not robust to collinearity later, we will definitely need to exclude one or more variables in each extremely correlated group, and we may have to do the same for less (but still highly) correlated groups as well.

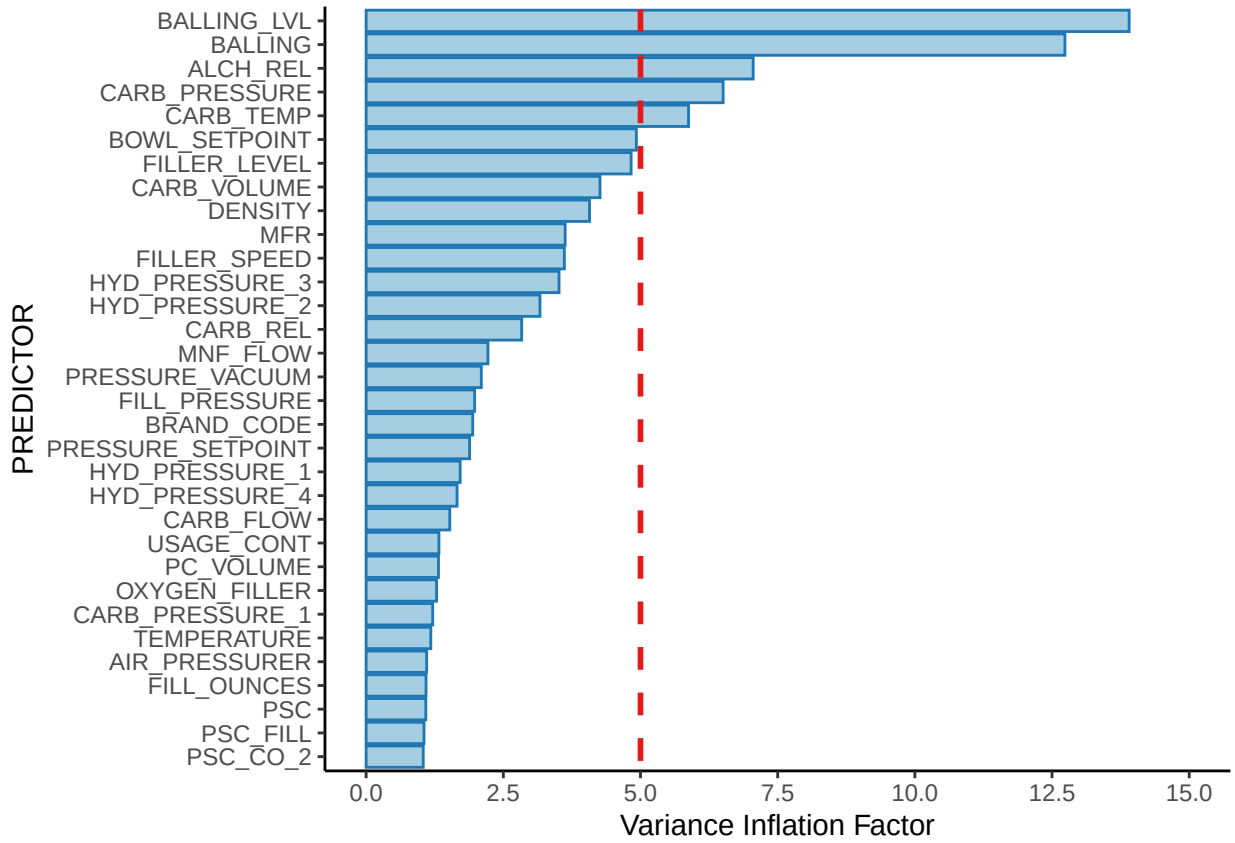
Data Preparation:

To create two versions of the data, one in which missing numeric values have been imputed and one in which observations with missing numeric values have been deleted, we first set a seed and split the data into train and test sets.

```
set.seed(417)
rows <- sample(nrow(main_df))
main_df <- main_df[rows, ]
sample <- sample(c(TRUE, FALSE), nrow(main_df), replace=TRUE,
                 prob=c(0.7,0.3))
train_df <- main_df[sample, ]
test_df <- main_df[!sample, ]
```

Next we perform KNN imputation on the missing numeric values for the primary train and test sets separately. To determine the variables this imputation method will use to calculate distance, and the weight of each distance variable used, we fit a random forest model to the training data, identify the 25 most important variables, and extract their variable importance scores. Since including highly correlated variables in a random forest model reduces all their variable importance scores, however, we're actually going to first fit a full multiple linear regression model, check the variance inflation factors to determine any sources of multicollinearity, and eliminate problematic predictors from consideration.

```
mlr_model1 <- lm(PH ~ ., data = train_df)
mlr_model1_vif <- as.data.frame(vif(mlr_model1)) |>
    rownames_to_column()
cols <- c("PREDICTOR", "GVIF", "DF", "GVIF_ADJ_BY_DF")
colnames(mlr_model1_vif) <- cols
palette <- brewer.pal(n = 12, name = "Paired")
p10 <- mlr_model1_vif |>
    ggplot() +
    geom_col(aes(x = reorder(PREDICTOR, GVIF_ADJ_BY_DF), y = GVIF_ADJ_BY_DF),
             color = palette[2], fill = palette[1]) +
    geom_abline(intercept = 5, slope = 0, linewidth = 1, linetype = 2,
                color = palette[6]) +
    labs(x = "PREDICTOR",
         y = "Variance Inflation Factor") +
    scale_y_continuous(limits = c(0, 15), breaks = seq(0, 15, 2.5)) +
    coord_flip()
p10
```



The variables with variance inflation factors greater than five are `BALLING_LVL`, `BALLING`, `ALCH_REL`, `CARB_PRESSURE`, and `CARB_TEMP`. We remove `ALCH_REL` and `BALLING` from variable importance consideration for imputation because the information these variables provide is largely covered by `BALLING_LVL`, and we remove `CARB_TEMP` from variable importance consideration for imputation in favor of `CARB_PRESSURE`for the same reason.

```
rf_model1 <- randomForest(PH ~ . - ALCH_REL - BALLING - CARB_TEMP, data = train_df,
                          importance = TRUE,
                          ntree = 1000,
                          na.action = na.omit)
rf_imp1 <- varImp(rf_model1, scale = TRUE)
cols <- c("Predictor", "Importance")
rf_imp1 <- rf_imp1 |>
    rownames_to_column()
colnames(rf_imp1) <- cols
rf_imp1 <- rf_imp1 |>
    arrange(desc(Importance)) |>
    top_n(25)
```

```
## Selecting by Importance
```

```
knitr::kable(rf_imp1, format = "simple")
```

Predictor	Importance
BRAND_CODE	61.200440
USAGE_CONT	45.665427
MNF_FLOW	43.158327
PRESSURE_VACUUM	40.102631
OXYGEN_FILLER	39.627032
CARB_REL	35.554152
BALLING_LVL	35.341949
AIR_PRESSURER	32.211929
TEMPERATURE	30.519410
FILLER_SPEED	30.475565
DENSITY	28.328100
CARB_FLOW	27.735542
FILLER_LEVEL	27.295486
BOWL_SETPOINT	24.555389
HYD_PRESSURE_1	22.681445
CARB_VOLUME	22.091540
CARB_PRESSURE_1	21.495410
HYD_PRESSURE_3	21.101332
HYD_PRESSURE_4	21.000142
PC_VOLUME	20.120549
FILL_PRESSURE	19.348660
HYD_PRESSURE_2	18.762196
MFR	18.121053
PRESSURE_SETPOINT	13.805334
FILL_OUNCES	3.732359

The above variables become the distance variables, and their variable importance scores become the weights in our KNN imputation.

```
dist_vars = rf_imp1$Predictor
wts = rf_imp1$Importance
find_cols_na <- function(df){
    col_sums_na <- colSums(is.na(df))
    cols <- names(col_sums_na[col_sums_na > 0])
    cols #returns column names vector
}
missing_val_cols <- find_cols_na(train_df)
primary_train_df <- train_df |>
    VIM::kNN(variable = missing_val_cols, k = 15, dist_var = dist_vars,
             weights = wts, numFun = median, imp_var = FALSE)
missing_val_cols <- find_cols_na(test_df)
primary_test_df <- test_df |>
    VIM::kNN(variable = missing_val_cols, k = 15, dist_var = dist_vars,
             weights = wts, numFun = median, imp_var = FALSE)
```

Then we create secondary train and test sets where all observations with missing numeric values have been deleted.

```
remove_rows_na <- function(df){
    na_row_sums <- rowSums(is.na(df))
    row_has_na <- ifelse(na_row_sums > 0, TRUE, FALSE)
```

```
    copy <- df[!row_has_na, ]
    copy
}
secondary_train_df <- remove_rows_na(train_df)
secondary_test_df <- remove_rows_na(test_df)
```

Model Building:

Linear Regression Models:

Adjusted R-squared for linear model with all predictors and missing values included:

```
lm0 <- lm(PH ~ ., data=primary_train_df)
summary(lm0)$adj.r.squared
```

```
## [1] 0.4120694
```

Adjusted R-squared for linear model with all predictors and no missing values:

```
lm1 <- lm(PH ~ ., data=secondary_train_df)
summary(lm1)$adj.r.squared
```

```
## [1] 0.4310702
```

Step-wise Predictor selection The mode of the step-wise search is "both" (ie we are seraching for the best model adding predictors one by one, as well as removing one by one)."Better" models are quantified as having lower AIC (Akaikon Information Criterion). Missing values are included for this model.

```
lm2 <- stepAIC(lm0, trace=FALSE)
summary(lm2)
```

```
##
## Call:
## lm(formula = PH ~ BRAND_CODE + CARB_VOLUME + PSC + MNF_FLOW +
##     CARB_PRESSURE_1 + HYD_PRESSURE_2 + HYD_PRESSURE_3 + FILLER_LEVEL +
##     TEMPERATURE + USAGE_CONT + CARB_FLOW + DENSITY + BALLING +
##     PRESSURE_VACUUM + OXYGEN_FILLER + BOWL_SETPOINT + PRESSURE_SETPOINT +
##     ALCH_REL + BALLING_LVL, data = primary_train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52113 -0.07784  0.00982  0.08610  0.42610
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8.973e+00  3.903e-01  22.987  < 2e-16 ***
## BRAND_CODEB        1.043e-01  2.653e-02   3.933 8.71e-05 ***
## BRAND_CODEC       -4.523e-02  2.617e-02  -1.728 0.084109 .
## BRAND_CODED        6.641e-02  1.838e-02   3.614 0.000310 ***
## BRAND_CODENA       2.893e-02  2.968e-02   0.975 0.329803
```

```
## CARB_VOLUME       -1.336e-01  5.285e-02  -2.528 0.011573 *
## PSC               -1.347e-01  6.436e-02  -2.093 0.036516 *
## MNF_FLOW          -6.775e-04  5.405e-05 -12.534  < 2e-16 ***
## CARB_PRESSURE_1    6.352e-03  8.181e-04   7.764 1.39e-14 ***
## HYD_PRESSURE_2    -1.095e-03  5.443e-04  -2.012 0.044384 *
## HYD_PRESSURE_3     3.409e-03  6.644e-04   5.131 3.19e-07 ***
## FILLER_LEVEL      -1.041e-03  6.428e-04  -1.620 0.105416
## TEMPERATURE       -1.426e-02  2.670e-03  -5.340 1.05e-07 ***
## USAGE_CONT        -5.349e-03  1.317e-03  -4.062 5.09e-05 ***
## CARB_FLOW          1.431e-05  3.823e-06   3.742 0.000188 ***
## DENSITY           -1.058e-01  3.281e-02  -3.224 0.001287 **
## BALLING           -1.208e-01  2.653e-02  -4.554 5.62e-06 ***
## PRESSURE_VACUUM   -2.681e-02  8.369e-03  -3.203 0.001383 **
## OXYGEN_FILLER     -2.925e-01  8.991e-02  -3.254 0.001161 **
## BOWL_SETPOINT      3.326e-03  6.649e-04   5.002 6.24e-07 ***
## PRESSURE_SETPOINT -7.773e-03  1.906e-03  -4.078 4.76e-05 ***
## ALCH_REL           6.935e-02  2.709e-02   2.560 0.010560 *
## BALLING_LVL        1.728e-01  2.753e-02   6.276 4.35e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1313 on 1754 degrees of freedom
## Multiple R-squared:  0.4204, Adjusted R-squared:  0.4131
## F-statistic: 57.83 on 22 and 1754 DF,  p-value: < 2.2e-16
```

Step-wise with no missing values:

```
lm3 <- stepAIC(lm1, trace=FALSE)
summary(lm3)
```

```
##
## Call:
## lm(formula = PH ~ BRAND_CODE + CARB_VOLUME + FILL_OUNCES + CARB_PRESSURE +
##     CARB_TEMP + MNF_FLOW + CARB_PRESSURE_1 + HYD_PRESSURE_2 +
##     HYD_PRESSURE_3 + HYD_PRESSURE_4 + TEMPERATURE + USAGE_CONT +
##     CARB_FLOW + DENSITY + BALLING + PRESSURE_VACUUM + OXYGEN_FILLER +
##     BOWL_SETPOINT + PRESSURE_SETPOINT + CARB_REL + BALLING_LVL,
##     data = secondary_train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54066 -0.07215  0.00653  0.08647  0.43003
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.305e+01  1.437e+00   9.085  < 2e-16 ***
## BRAND_CODEB        1.792e-01  3.740e-02   4.791 1.83e-06 ***
## BRAND_CODEC        2.057e-02  3.601e-02   0.571 0.568022
## BRAND_CODED        1.079e-01  1.678e-02   6.429 1.74e-10 ***
## BRAND_CODENA       1.160e-01  4.023e-02   2.883 0.004001 **
## CARB_VOLUME       -3.947e-01  1.363e-01  -2.895 0.003844 **
## FILL_OUNCES       -9.681e-02  4.272e-02  -2.266 0.023594 *
## CARB_PRESSURE      1.226e-02  6.420e-03   1.909 0.056421 .
```

```
## CARB_TEMP         -9.288e-03  5.022e-03  -1.849 0.064594 .
## MNF_FLOW          -6.262e-04  5.923e-05 -10.571  < 2e-16 ***
## CARB_PRESSURE_1    5.918e-03  9.018e-04   6.562 7.36e-11 ***
## HYD_PRESSURE_2    -1.463e-03  5.835e-04  -2.507 0.012284 *
## HYD_PRESSURE_3     3.509e-03  7.331e-04   4.787 1.87e-06 ***
## HYD_PRESSURE_4     7.270e-04  4.817e-04   1.509 0.131453
## TEMPERATURE       -1.874e-02  3.763e-03  -4.981 7.09e-07 ***
## USAGE_CONT        -6.878e-03  1.480e-03  -4.647 3.68e-06 ***
## CARB_FLOW          2.013e-05  5.337e-06   3.772 0.000168 ***
## DENSITY           -1.149e-01  3.567e-02  -3.222 0.001303 **
## BALLING           -1.885e-01  4.217e-02  -4.469 8.45e-06 ***
## PRESSURE_VACUUM   -4.973e-02  1.111e-02  -4.476 8.22e-06 ***
## OXYGEN_FILLER     -3.267e-01  1.062e-01  -3.076 0.002139 **
## BOWL_SETPOINT      2.323e-03  3.361e-04   6.911 7.18e-12 ***
## PRESSURE_SETPOINT -7.642e-03  2.080e-03  -3.673 0.000248 ***
## CARB_REL           1.090e-01  7.484e-02   1.456 0.145535
## BALLING_LVL        3.049e-01  4.850e-02   6.288 4.26e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1283 on 1448 degrees of freedom
## Multiple R-squared:  0.4424, Adjusted R-squared:  0.4331
## F-statistic: 47.86 on 24 and 1448 DF,  p-value: < 2.2e-16
```

FILL_OUNCES, PSC, ALCH_REL are among the differences in predictors between the two step-wise linear models. Of the linear models, the step-wise model with no missing values has a higher Adjusted R-squared at 0.43.

Nonlinear Regression Models:

Multivariate Adaptive Regression Splines Generalized R-squared of a MARS model with missing values

```
mars1 <- earth::earth(PH ~ ., data = primary_train_df)
mars1$grsq
```

```
## [1] 0.4248042
```

Generalized R-squared of a MARS model without missing values

```
mars2 <- earth::earth(PH ~ ., data = secondary_train_df)
mars2$grsq
```

```
## [1] 0.4452059
```

The MARS model without missing values performed about 2% better.

K Nearest Neighbors

KNN with missing values

```
ctrl <- trainControl(method="repeatedcv",repeats = 3)
knn1 <- train(PH ~ ., data = primary_train_df, method = "knn", trControl=ctrl, preProcess = c("center",
knn1$results
```

```
##     k      RMSE  Rsquared        MAE      RMSESD RsquaredSD       MAESD
## 1   5 0.1210190 0.5077160 0.09041480 0.007692803 0.05311684 0.005216766
## 2   7 0.1203227 0.5129572 0.09090697 0.007614240 0.05437946 0.005365893
## 3   9 0.1211402 0.5067878 0.09251006 0.007635269 0.05545390 0.005518617
## 4  11 0.1218617 0.5020730 0.09344699 0.008274703 0.06197180 0.005814792
## 5  13 0.1223929 0.4983277 0.09407935 0.008009016 0.06069483 0.005832246
## 6  15 0.1234620 0.4896835 0.09536075 0.007738599 0.05790123 0.005920425
## 7  17 0.1244735 0.4819488 0.09628523 0.007587700 0.05652577 0.005895991
## 8  19 0.1252861 0.4755775 0.09686126 0.007450170 0.05672739 0.005833652
## 9  21 0.1260550 0.4698875 0.09739671 0.007328652 0.05580912 0.005670036
## 10 23 0.1268626 0.4632686 0.09785015 0.007446063 0.05689723 0.005778785
## 11 25 0.1273760 0.4590031 0.09831088 0.007417029 0.05642336 0.005707683
## 12 27 0.1279242 0.4547387 0.09875313 0.007340357 0.05723720 0.005660514
## 13 29 0.1284111 0.4507985 0.09915531 0.007411504 0.05760406 0.005677733
## 14 31 0.1291080 0.4447055 0.09979980 0.007278146 0.05672946 0.005558173
## 15 33 0.1297440 0.4392083 0.10036049 0.007279430 0.05694777 0.005667451
## 16 35 0.1302029 0.4353581 0.10067379 0.007267559 0.05645076 0.005687862
## 17 37 0.1305938 0.4323563 0.10092350 0.007184968 0.05630242 0.005544514
## 18 39 0.1309903 0.4291314 0.10123264 0.007210521 0.05609429 0.005636099
## 19 41 0.1314458 0.4255725 0.10149659 0.007125009 0.05609822 0.005551930
## 20 43 0.1318614 0.4220736 0.10184257 0.007225879 0.05751020 0.005636593
```

KNN without missing values

```
knn2 <- train(PH ~ ., data = secondary_train_df, method = "knn", trControl=ctrl, preProcess = c("center
knn2$results
```

```
##     k      RMSE  Rsquared        MAE      RMSESD RsquaredSD       MAESD
## 1   5 0.1143771 0.5524199 0.08549123 0.009259738 0.05514064 0.006534897
## 2   7 0.1143672 0.5541766 0.08649048 0.008299311 0.05095771 0.005818019
## 3   9 0.1151408 0.5486924 0.08780881 0.008930976 0.05443097 0.006151551
## 4  11 0.1161389 0.5417671 0.08901554 0.009267439 0.05849473 0.006497887
## 5  13 0.1173533 0.5320168 0.09000779 0.008711463 0.05663163 0.006139681
## 6  15 0.1184675 0.5234814 0.09124716 0.008671984 0.05546420 0.005992882
## 7  17 0.1196567 0.5140747 0.09229583 0.008503269 0.05566728 0.005758091
## 8  19 0.1205587 0.5068085 0.09309436 0.008260696 0.05424687 0.005447394
## 9  21 0.1210880 0.5032914 0.09327472 0.008277697 0.05348685 0.005671254
## 10 23 0.1219129 0.4967053 0.09405988 0.008392550 0.05399672 0.005715929
## 11 25 0.1226662 0.4906081 0.09472520 0.008246657 0.05308477 0.005750855
## 12 27 0.1235483 0.4832575 0.09542238 0.008220911 0.05244087 0.005702362
## 13 29 0.1242832 0.4772667 0.09602690 0.007891275 0.05248927 0.005363742
## 14 31 0.1252014 0.4694019 0.09670552 0.007768071 0.05097481 0.005229042
## 15 33 0.1260907 0.4619124 0.09720836 0.007784387 0.05098748 0.005275481
## 16 35 0.1268544 0.4554195 0.09765853 0.007775865 0.04980675 0.005365854
## 17 37 0.1274474 0.4504840 0.09803606 0.007589225 0.04797425 0.005350412
## 18 39 0.1280333 0.4452478 0.09841944 0.007492698 0.04729737 0.005256891
## 19 41 0.1286766 0.4395307 0.09891641 0.007527547 0.04755558 0.005213890
## 20 43 0.1291203 0.4355804 0.09926446 0.007789113 0.04739336 0.005361321
```

For the knn model that includes missing values, the best value of k is 5. The R-squared is 0.51. For the knn model without missing values, the bests value of k is 7. The R-squared is 0.56.

Tree Models:

Final Model Selection:

Final Model Evaluation:

We load the evaluation dataset we will make predictions on.

```
my_url2 <- "https://github.com/geedoubledee/data624_project2/raw/main/StudentEvaluation.xlsx"
temp <- tempfile(fileext = ".xlsx")
req <- GET(my_url2, authenticate(Sys.getenv("GITHUB_PAT"), ""),
           write_disk(path = temp))
eval_df <- readxl::read_excel(temp)
colnames(eval_df) <- to_screaming_snake_case(colnames(eval_df))
```

We build and report the predictive factors in BOTH a technical and non-technical report. The non-technical report will be in a business-friendly readable document, and the predictions will be in an Excel readable format. The technical report will clearly show the models we tested and how we selected our final approach.

Conclusions: