library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.2.0 ──  
## ✔ broom 1.0.7 ✔ rsample 1.2.1  
## ✔ dials 1.3.0 ✔ tune 1.2.1  
## ✔ infer 1.0.7 ✔ workflows 1.1.4  
## ✔ modeldata 1.4.0 ✔ workflowsets 1.1.0  
## ✔ parsnip 1.2.1 ✔ yardstick 1.3.2  
## ✔ recipes 1.1.0   
## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Dig deeper into tidy modeling with R at https://www.tmwr.org

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(naniar)  
library(mice)

##   
## Attaching package: 'mice'  
##   
## The following object is masked from 'package:stats':  
##   
## filter  
##   
## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(dplyr)

train = read\_csv("C:/Users/davef/OneDrive/Desktop/train.csv")

## Rows: 26570 Columns: 26  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (4): product\_code, attribute\_0, attribute\_1, failure  
## dbl (22): id, loading, attribute\_2, attribute\_3, measurement\_0, measurement\_...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

str(train)

## spc\_tbl\_ [26,570 × 26] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ id : num [1:26570] 0 1 2 3 4 5 6 7 8 9 ...  
## $ product\_code : chr [1:26570] "A" "A" "A" "A" ...  
## $ loading : num [1:26570] 80.1 84.9 82.4 101.1 188.1 ...  
## $ attribute\_0 : chr [1:26570] "material\_7" "material\_7" "material\_7" "material\_7" ...  
## $ attribute\_1 : chr [1:26570] "material\_8" "material\_8" "material\_8" "material\_8" ...  
## $ attribute\_2 : num [1:26570] 9 9 9 9 9 9 9 9 9 9 ...  
## $ attribute\_3 : num [1:26570] 5 5 5 5 5 5 5 5 5 5 ...  
## $ measurement\_0 : num [1:26570] 7 14 12 13 9 11 12 4 9 10 ...  
## $ measurement\_1 : num [1:26570] 8 3 1 2 2 4 2 8 6 4 ...  
## $ measurement\_2 : num [1:26570] 4 3 5 6 8 0 4 8 5 7 ...  
## $ measurement\_3 : num [1:26570] 18 18.2 18.1 17.3 19.3 ...  
## $ measurement\_4 : num [1:26570] 12.5 11.5 11.7 11.2 12.9 ...  
## $ measurement\_5 : num [1:26570] 15.7 17.7 16.7 18.6 17 ...  
## $ measurement\_6 : num [1:26570] 19.3 17.9 18.2 18.3 15.7 ...  
## $ measurement\_7 : num [1:26570] 11.7 12.7 12.7 12.6 11.3 ...  
## $ measurement\_8 : num [1:26570] 20.2 17.9 18.3 19.1 18.1 ...  
## $ measurement\_9 : num [1:26570] 10.7 12.4 12.7 12.5 10.3 ...  
## $ measurement\_10: num [1:26570] 15.9 17.9 15.6 16.3 17.1 ...  
## $ measurement\_11: num [1:26570] 17.6 17.9 NA 18.4 19.9 ...  
## $ measurement\_12: num [1:26570] 15.2 11.8 13.8 10 12.4 ...  
## $ measurement\_13: num [1:26570] 15 14.7 16.7 15.2 16.2 ...  
## $ measurement\_14: num [1:26570] NA 15.4 18.6 15.6 12.8 ...  
## $ measurement\_15: num [1:26570] 13 14.4 14.1 16.2 13.2 ...  
## $ measurement\_16: num [1:26570] 14.7 15.6 17.9 17.2 16.4 ...  
## $ measurement\_17: num [1:26570] 764 682 663 826 580 ...  
## $ failure : chr [1:26570] "No" "No" "No" "No" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. id = col\_double(),  
## .. product\_code = col\_character(),  
## .. loading = col\_double(),  
## .. attribute\_0 = col\_character(),  
## .. attribute\_1 = col\_character(),  
## .. attribute\_2 = col\_double(),  
## .. attribute\_3 = col\_double(),  
## .. measurement\_0 = col\_double(),  
## .. measurement\_1 = col\_double(),  
## .. measurement\_2 = col\_double(),  
## .. measurement\_3 = col\_double(),  
## .. measurement\_4 = col\_double(),  
## .. measurement\_5 = col\_double(),  
## .. measurement\_6 = col\_double(),  
## .. measurement\_7 = col\_double(),  
## .. measurement\_8 = col\_double(),  
## .. measurement\_9 = col\_double(),  
## .. measurement\_10 = col\_double(),  
## .. measurement\_11 = col\_double(),  
## .. measurement\_12 = col\_double(),  
## .. measurement\_13 = col\_double(),  
## .. measurement\_14 = col\_double(),  
## .. measurement\_15 = col\_double(),  
## .. measurement\_16 = col\_double(),  
## .. measurement\_17 = col\_double(),  
## .. failure = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

summary(train)

## id product\_code loading attribute\_0   
## Min. : 0 Length:26570 Min. : 33.16 Length:26570   
## 1st Qu.: 6642 Class :character 1st Qu.: 99.99 Class :character   
## Median :13284 Mode :character Median :122.39 Mode :character   
## Mean :13284 Mean :127.83   
## 3rd Qu.:19927 3rd Qu.:149.15   
## Max. :26569 Max. :385.86   
## NA's :250   
## attribute\_1 attribute\_2 attribute\_3 measurement\_0   
## Length:26570 Min. :5.000 Min. :5.00 Min. : 0.000   
## Class :character 1st Qu.:6.000 1st Qu.:6.00 1st Qu.: 4.000   
## Mode :character Median :6.000 Median :8.00 Median : 7.000   
## Mean :6.754 Mean :7.24 Mean : 7.416   
## 3rd Qu.:8.000 3rd Qu.:8.00 3rd Qu.:10.000   
## Max. :9.000 Max. :9.00 Max. :29.000   
##   
## measurement\_1 measurement\_2 measurement\_3 measurement\_4   
## Min. : 0.000 Min. : 0.000 Min. :13.97 Min. : 8.008   
## 1st Qu.: 5.000 1st Qu.: 4.000 1st Qu.:17.12 1st Qu.:11.051   
## Median : 8.000 Median : 6.000 Median :17.79 Median :11.733   
## Mean : 8.233 Mean : 6.257 Mean :17.79 Mean :11.732   
## 3rd Qu.:11.000 3rd Qu.: 8.000 3rd Qu.:18.47 3rd Qu.:12.410   
## Max. :29.000 Max. :24.000 Max. :21.50 Max. :16.484   
## NA's :381 NA's :538   
## measurement\_5 measurement\_6 measurement\_7 measurement\_8   
## Min. :12.07 Min. :12.71 Min. : 7.968 Min. :15.22   
## 1st Qu.:16.44 1st Qu.:16.84 1st Qu.:11.045 1st Qu.:18.34   
## Median :17.13 Median :17.52 Median :11.712 Median :19.02   
## Mean :17.13 Mean :17.51 Mean :11.717 Mean :19.02   
## 3rd Qu.:17.80 3rd Qu.:18.18 3rd Qu.:12.391 3rd Qu.:19.71   
## Max. :21.43 Max. :21.54 Max. :15.419 Max. :23.81   
## NA's :676 NA's :796 NA's :937 NA's :1048   
## measurement\_9 measurement\_10 measurement\_11 measurement\_12   
## Min. : 7.537 Min. : 9.323 Min. :12.46 Min. : 5.167   
## 1st Qu.:10.757 1st Qu.:15.209 1st Qu.:18.17 1st Qu.:10.703   
## Median :11.430 Median :16.127 Median :19.21 Median :11.717   
## Mean :11.431 Mean :16.118 Mean :19.17 Mean :11.703   
## 3rd Qu.:12.102 3rd Qu.:17.025 3rd Qu.:20.21 3rd Qu.:12.709   
## Max. :15.412 Max. :22.479 Max. :25.64 Max. :17.663   
## NA's :1227 NA's :1300 NA's :1468 NA's :1601   
## measurement\_13 measurement\_14 measurement\_15 measurement\_16   
## Min. :10.89 Min. : 9.14 Min. : 9.104 Min. : 9.701   
## 1st Qu.:14.89 1st Qu.:15.06 1st Qu.:13.957 1st Qu.:15.268   
## Median :15.63 Median :16.04 Median :14.969 Median :16.436   
## Mean :15.65 Mean :16.05 Mean :14.996 Mean :16.461   
## 3rd Qu.:16.37 3rd Qu.:17.08 3rd Qu.:16.018 3rd Qu.:17.628   
## Max. :22.71 Max. :22.30 Max. :21.626 Max. :24.094   
## NA's :1774 NA's :1874 NA's :2009 NA's :2110   
## measurement\_17 failure   
## Min. : 196.8 Length:26570   
## 1st Qu.: 619.0 Class :character   
## Median : 701.0 Mode :character   
## Mean : 701.3   
## 3rd Qu.: 784.1   
## Max. :1312.8   
## NA's :2284

unique(train$attribute\_0)

## [1] "material\_7" "material\_5"

unique(train$attribute\_1)

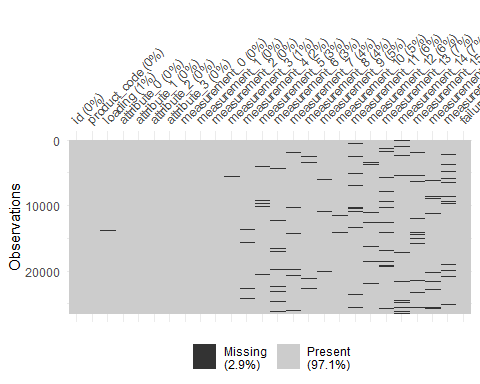
## [1] "material\_8" "material\_5" "material\_6"

train <- train %>%  
 mutate(  
 product\_code = as\_factor(product\_code),  
 product\_code = fct\_recode(product\_code, "A" = "A", "B" = "B", "C" = "C", "D" = "D", "E" = "E"),  
   
 attribute\_0 = as\_factor(attribute\_0),  
 attribute\_0 = fct\_recode(attribute\_0, "7" = "material\_7", "5" = "material\_5"),  
   
 attribute\_1 = as\_factor(attribute\_1),  
 attribute\_1 = fct\_recode(attribute\_1, "8" = "material\_8", "5" = "material\_5", "6" = "material\_6"),  
   
 failure = as.numeric(as\_factor(failure)) - 1,   
 )

str(train)

## tibble [26,570 × 26] (S3: tbl\_df/tbl/data.frame)  
## $ id : num [1:26570] 0 1 2 3 4 5 6 7 8 9 ...  
## $ product\_code : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ loading : num [1:26570] 80.1 84.9 82.4 101.1 188.1 ...  
## $ attribute\_0 : Factor w/ 2 levels "7","5": 1 1 1 1 1 1 1 1 1 1 ...  
## $ attribute\_1 : Factor w/ 3 levels "8","5","6": 1 1 1 1 1 1 1 1 1 1 ...  
## $ attribute\_2 : num [1:26570] 9 9 9 9 9 9 9 9 9 9 ...  
## $ attribute\_3 : num [1:26570] 5 5 5 5 5 5 5 5 5 5 ...  
## $ measurement\_0 : num [1:26570] 7 14 12 13 9 11 12 4 9 10 ...  
## $ measurement\_1 : num [1:26570] 8 3 1 2 2 4 2 8 6 4 ...  
## $ measurement\_2 : num [1:26570] 4 3 5 6 8 0 4 8 5 7 ...  
## $ measurement\_3 : num [1:26570] 18 18.2 18.1 17.3 19.3 ...  
## $ measurement\_4 : num [1:26570] 12.5 11.5 11.7 11.2 12.9 ...  
## $ measurement\_5 : num [1:26570] 15.7 17.7 16.7 18.6 17 ...  
## $ measurement\_6 : num [1:26570] 19.3 17.9 18.2 18.3 15.7 ...  
## $ measurement\_7 : num [1:26570] 11.7 12.7 12.7 12.6 11.3 ...  
## $ measurement\_8 : num [1:26570] 20.2 17.9 18.3 19.1 18.1 ...  
## $ measurement\_9 : num [1:26570] 10.7 12.4 12.7 12.5 10.3 ...  
## $ measurement\_10: num [1:26570] 15.9 17.9 15.6 16.3 17.1 ...  
## $ measurement\_11: num [1:26570] 17.6 17.9 NA 18.4 19.9 ...  
## $ measurement\_12: num [1:26570] 15.2 11.8 13.8 10 12.4 ...  
## $ measurement\_13: num [1:26570] 15 14.7 16.7 15.2 16.2 ...  
## $ measurement\_14: num [1:26570] NA 15.4 18.6 15.6 12.8 ...  
## $ measurement\_15: num [1:26570] 13 14.4 14.1 16.2 13.2 ...  
## $ measurement\_16: num [1:26570] 14.7 15.6 17.9 17.2 16.4 ...  
## $ measurement\_17: num [1:26570] 764 682 663 826 580 ...  
## $ failure : num [1:26570] 0 0 0 0 0 0 1 1 0 0 ...

vis\_miss(train)



train\_clean <- na.omit(train)  
  
# Check how many rows remain  
print(nrow(train\_clean))

## [1] 12183

summary(train\_clean)

## id product\_code loading attribute\_0 attribute\_1  
## Min. : 1 A:2381 Min. : 40.81 7:9763 8:5047   
## 1st Qu.: 6516 B:2420 1st Qu.:100.07 5:2420 5:4744   
## Median :13102 C:2666 Median :122.18 6:2392   
## Mean :13186 D:2324 Mean :127.80   
## 3rd Qu.:19802 E:2392 3rd Qu.:149.19   
## Max. :26569 Max. :385.86   
## attribute\_2 attribute\_3 measurement\_0 measurement\_1   
## Min. :5.000 Min. :5.000 Min. : 0.000 Min. : 0.000   
## 1st Qu.:6.000 1st Qu.:6.000 1st Qu.: 4.000 1st Qu.: 5.000   
## Median :6.000 Median :8.000 Median : 7.000 Median : 8.000   
## Mean :6.765 Mean :7.229 Mean : 7.419 Mean : 8.236   
## 3rd Qu.:8.000 3rd Qu.:8.000 3rd Qu.:10.000 3rd Qu.:11.000   
## Max. :9.000 Max. :9.000 Max. :26.000 Max. :27.000   
## measurement\_2 measurement\_3 measurement\_4 measurement\_5   
## Min. : 0.000 Min. :14.10 Min. : 8.097 Min. :12.07   
## 1st Qu.: 4.000 1st Qu.:17.11 1st Qu.:11.061 1st Qu.:16.43   
## Median : 6.000 Median :17.78 Median :11.733 Median :17.13   
## Mean : 6.216 Mean :17.78 Mean :11.735 Mean :17.13   
## 3rd Qu.: 8.000 3rd Qu.:18.47 3rd Qu.:12.413 3rd Qu.:17.81   
## Max. :24.000 Max. :21.25 Max. :16.484 Max. :21.43   
## measurement\_6 measurement\_7 measurement\_8 measurement\_9   
## Min. :12.71 Min. : 7.973 Min. :15.27 Min. : 7.537   
## 1st Qu.:16.82 1st Qu.:11.050 1st Qu.:18.34 1st Qu.:10.758   
## Median :17.50 Median :11.712 Median :19.01 Median :11.434   
## Mean :17.50 Mean :11.715 Mean :19.02 Mean :11.434   
## 3rd Qu.:18.17 3rd Qu.:12.395 3rd Qu.:19.71 3rd Qu.:12.101   
## Max. :21.08 Max. :15.243 Max. :23.33 Max. :15.045   
## measurement\_10 measurement\_11 measurement\_12 measurement\_13   
## Min. : 9.676 Min. :12.46 Min. : 5.167 Min. :11.04   
## 1st Qu.:15.227 1st Qu.:18.16 1st Qu.:10.698 1st Qu.:14.87   
## Median :16.158 Median :19.23 Median :11.727 Median :15.62   
## Mean :16.137 Mean :19.18 Mean :11.709 Mean :15.64   
## 3rd Qu.:17.034 3rd Qu.:20.23 3rd Qu.:12.732 3rd Qu.:16.38   
## Max. :21.459 Max. :25.43 Max. :17.318 Max. :22.39   
## measurement\_14 measurement\_15 measurement\_16 measurement\_17   
## Min. :10.32 Min. : 9.158 Min. :10.06 Min. : 196.8   
## 1st Qu.:15.05 1st Qu.:13.950 1st Qu.:15.27 1st Qu.: 618.1   
## Median :16.03 Median :14.985 Median :16.44 Median : 700.2   
## Mean :16.05 Mean :15.004 Mean :16.46 Mean : 700.9   
## 3rd Qu.:17.08 3rd Qu.:16.038 3rd Qu.:17.61 3rd Qu.: 782.9   
## Max. :22.30 Max. :20.644 Max. :23.16 Max. :1182.0   
## failure   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.2098   
## 3rd Qu.:0.0000   
## Max. :1.0000

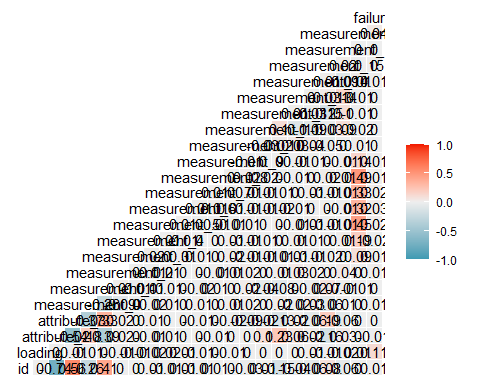
view(train\_clean)

table(train$failure) / nrow(train\_clean) # Check proportion of failure cases

##   
## 0 1   
## 1.7172289 0.4636789

ggcorr(train\_clean,label = TRUE,label\_round = 2)

## Warning in ggcorr(train\_clean, label = TRUE, label\_round = 2): data in  
## column(s) 'product\_code', 'attribute\_0', 'attribute\_1' are not numeric and were  
## ignored



numeric\_cols <- train\_clean %>% select\_if(is.numeric)  
  
# Compute correlation matrix  
cor\_matrix <- cor(numeric\_cols, use = "complete.obs")  
  
# Extract correlations with failure  
failure\_correlation <- cor\_matrix["failure", ]  
  
# Sort and print the correlations (absolute values for strongest predictors first)  
sorted\_correlations <- sort(abs(failure\_correlation), decreasing = TRUE)  
print(sorted\_correlations)

## failure loading measurement\_17 measurement\_6 measurement\_5   
## 1.0000000000 0.1133340834 0.0372130222 0.0271860098 0.0220809232   
## measurement\_4 measurement\_7 id measurement\_14 measurement\_8   
## 0.0172508918 0.0165630968 0.0099752260 0.0083531202 0.0070598907   
## measurement\_2 measurement\_0 measurement\_3 attribute\_2 measurement\_9   
## 0.0068379609 0.0066065106 0.0061993854 0.0060943168 0.0051655448   
## attribute\_3 measurement\_15 measurement\_13 measurement\_11 measurement\_12   
## 0.0031938354 0.0030306373 0.0026109128 0.0015136670 0.0015128762   
## measurement\_1 measurement\_16 measurement\_10   
## 0.0009651279 0.0004861654 0.0001709006

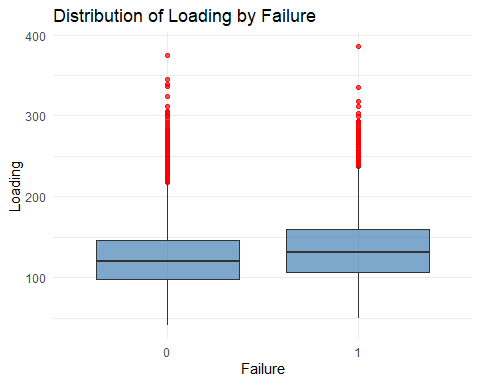
product\_table <- table(train\_clean$product\_code, train\_clean$failure)  
  
# Run Chi-Square test  
chi\_test <- chisq.test(product\_table)  
  
# Print results  
print(chi\_test)

##   
## Pearson's Chi-squared test  
##   
## data: product\_table  
## X-squared = 5.5009, df = 4, p-value = 0.2397

anova\_test <- aov(failure ~ product\_code, data = train\_clean)  
summary(anova\_test)

## Df Sum Sq Mean Sq F value Pr(>F)  
## product\_code 4 0.9 0.2280 1.375 0.24  
## Residuals 12178 2018.8 0.1658

library(ggplot2)  
  
ggplot(train\_clean, aes(x = as.factor(failure), y = loading)) +  
 geom\_boxplot(fill = "steelblue", alpha = 0.7, outlier.color = "red") +  
 labs(title = "Distribution of Loading by Failure", x = "Failure", y = "Loading") +  
 theme\_minimal()

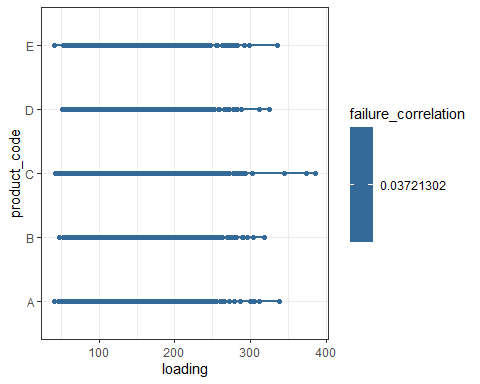


t\_test <- t.test(loading ~ failure, data = train\_clean)  
print(t\_test)

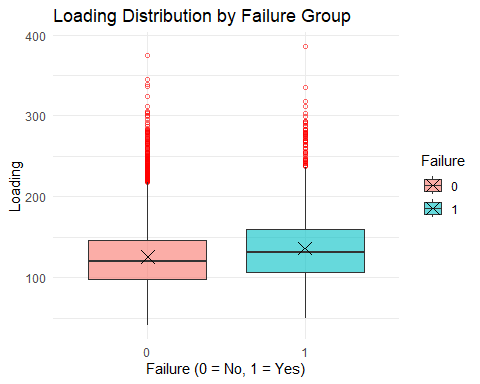
##   
## Welch Two Sample t-test  
##   
## data: loading by failure  
## t = -11.939, df = 3760.7, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0  
## 95 percent confidence interval:  
## -12.645491 -9.078073  
## sample estimates:  
## mean in group 0 mean in group 1   
## 125.5236 136.3853

train\_clean <- train\_clean %>%  
 mutate(failure\_correlation = cor(train\_clean$measurement\_17, train\_clean$failure, use = "complete.obs"))  
  
# Plot with the corrected color aesthetic  
ggplot(train\_clean, aes(x = loading, y = product\_code, color = failure\_correlation)) +  
 geom\_point() +   
 geom\_smooth(method = "lm", se = FALSE) +   
 theme\_bw()

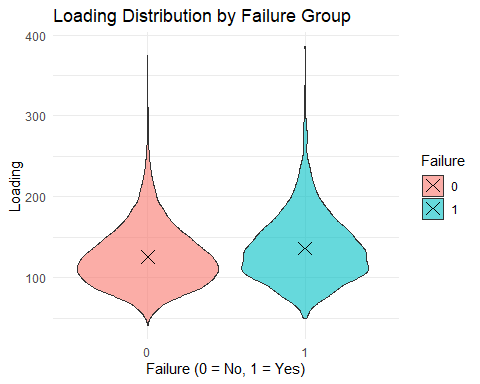
## `geom\_smooth()` using formula = 'y ~ x'



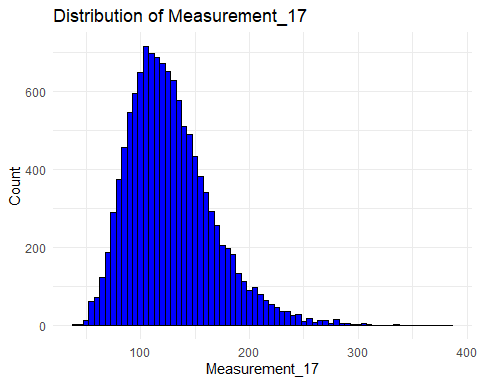
ggplot(train\_clean, aes(x = as.factor(failure), y = loading, fill = as.factor(failure))) +  
 geom\_boxplot(alpha = 0.6, outlier.color = "red", outlier.shape = 1) +  
 stat\_summary(fun = mean, geom = "point", shape = 4, size = 4, color = "black") +  
 labs(title = "Loading Distribution by Failure Group",  
 x = "Failure (0 = No, 1 = Yes)",  
 y = "Loading",  
 fill = "Failure") +  
 theme\_minimal()



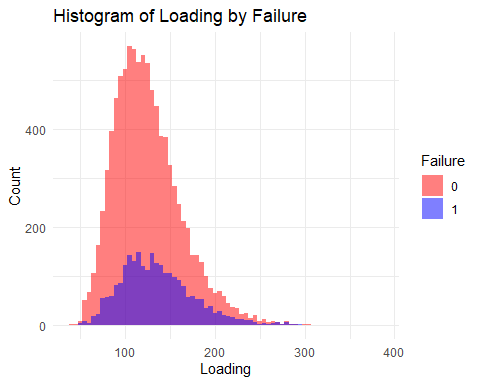
ggplot(train\_clean, aes(x = as.factor(failure), y = loading, fill = as.factor(failure))) +  
 geom\_violin(alpha = 0.6) +  
 stat\_summary(fun = mean, geom = "point", shape = 4, size = 4, color = "black") +  
 labs(title = "Loading Distribution by Failure Group",  
 x = "Failure (0 = No, 1 = Yes)",  
 y = "Loading",  
 fill = "Failure") +  
 theme\_minimal()



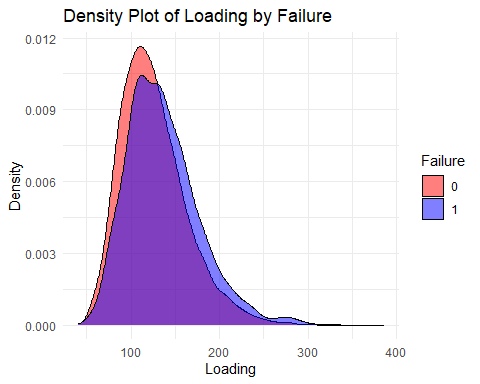
ggplot(train\_clean, aes(x = loading)) +  
 geom\_histogram(binwidth = 5, fill = "blue", color = "black") +  
 theme\_minimal() +  
 labs(title = "Distribution of Measurement\_17", x = "Measurement\_17", y = "Count")



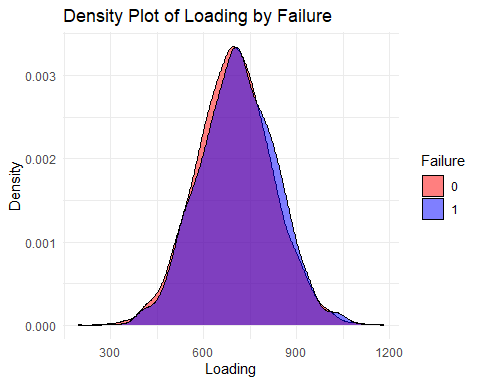
ggplot(train\_clean, aes(x = loading, fill = as.factor(failure))) +  
 geom\_histogram(binwidth = 5, position = "identity", alpha = 0.5) +  
 theme\_minimal() +  
 labs(title = "Histogram of Loading by Failure",  
 x = "Loading",  
 y = "Count",  
 fill = "Failure") +  
 scale\_fill\_manual(values = c("red", "blue"))



ggplot(train\_clean, aes(x = loading, fill = as.factor(failure))) +  
 geom\_density(alpha = 0.5) +  
 theme\_minimal() +  
 labs(title = "Density Plot of Loading by Failure",  
 x = "Loading",  
 y = "Density",  
 fill = "Failure") +  
 scale\_fill\_manual(values = c("red", "blue"))



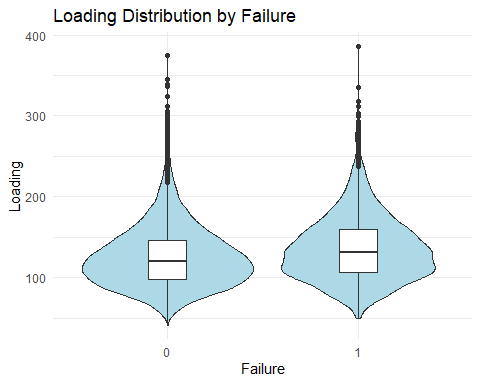
ggplot(train\_clean, aes(x = measurement\_17, fill = as.factor(failure))) +  
 geom\_density(alpha = 0.5) +  
 theme\_minimal() +  
 labs(title = "Density Plot of Loading by Failure",  
 x = "Loading",  
 y = "Density",  
 fill = "Failure") +  
 scale\_fill\_manual(values = c("red", "blue"))



t\_test <- t.test(loading ~ failure, data = train\_clean)  
print(t\_test)

##   
## Welch Two Sample t-test  
##   
## data: loading by failure  
## t = -11.939, df = 3760.7, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0  
## 95 percent confidence interval:  
## -12.645491 -9.078073  
## sample estimates:  
## mean in group 0 mean in group 1   
## 125.5236 136.3853

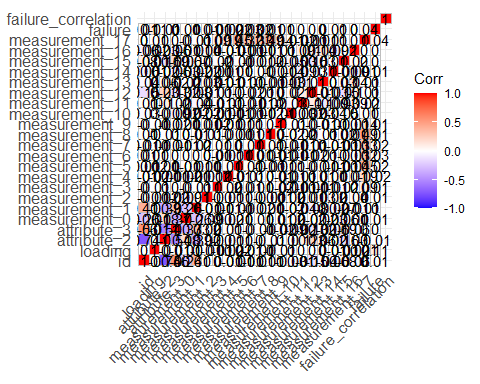
ggplot(train\_clean, aes(x = as.factor(failure), y = loading)) +  
 geom\_violin(fill = "lightblue") +  
 geom\_boxplot(width = 0.2, fill = "white") +  
 theme\_minimal() +  
 labs(title = "Loading Distribution by Failure", x = "Failure", y = "Loading")



library(ggcorrplot)  
cor\_matrix <- cor(select\_if(train\_clean, is.numeric), use = "complete.obs")

## Warning in cor(select\_if(train\_clean, is.numeric), use = "complete.obs"): the  
## standard deviation is zero

ggcorrplot(cor\_matrix, method = "square", lab = TRUE)



ggplot(train\_clean, aes(x = measurement\_13, fill = as.factor(failure))) +  
 geom\_density(alpha = 0.5) +  
 theme\_minimal() +  
 labs(title = "Density Plot of Measurement\_13 by Failure", x = "Measurement\_13", y = "Density")

