Stat 245 – Vermeer Project

Adham Rishmawi, Aubrey Williams, Daniel Kwik, Misgana Dinberu

December 14, 2022

Project Question

Background Vermeer Corporation has provided us datasets to help answer statistical questions that might be useful for them. The project question is open and the team was given autonomy on what to do with the data. There are two types of datasets provided to us. The first was Telematics data, machine codes that machines would send that contained information about any machine failure. The second is repair data, data about repair jobs being done on machines when it is sent in for repair.

Question we chose Because the Vermeer Corporation was interested more in the business case, we decided to choose a response variable that we think would be most useful to help reduce cost, namely the number of repairs of a given machine. The logic we started with is if we can figure out what variables best predict higher counts of repairs, Vermeer can take steps to fix those things and deliver value to their customers and dealers.

Question we chose: Which variables are most strongly associated with the number of repairs of a given machine?

We were able to wrangle a final dataset (`summary_by_machine`) with the available variables to test for, given the constraints of time and resource in data wrangling:

-The length of time between a machine's failure, and when it was repaired.

-The total count of indicator lamps turned on (notification, amber, red, malfunction)

-The model of machine

-The number of hours machine was running for before it broke down

Hypothesis: Out of all these variables, we hypothesize that the number of machine hours, length of time between breakdown and repair, and count of 'malfunction' indicator lights are the predictors that are most strongly associated with the number of repairs on a given machine. In our final dataset, these predictors are `max_machine_hours`, `down_time`, and `n_malfunction` respectively, and our response variable is `n_work_orders` - the number of repairs of a given machine.

Logic: -Machines hours: The longer the machine runs for, the greater the depreciation of machine parts, the more often in needs to be repaired. -Down time: The longer we wait before repairing a machine, the more mechanical problems can accumulate, leading to higher number of reapirs -Malfunction lamp: The most severe lamp status (malfunction), as opposed to notification, amber, and red will likely predict higher repairs.

Wrangling:

These are the six data sets we will be using. We have loaded up the data and included a glimpse into them to see what there is.

```
vinpin_data <- read_csv("VINPINdata.csv")
glimpse(vinpin_data)
```

```
## Rows: 18,183
## Columns: 3
## $ assetId   <dbl> 216049, 216050, 216051, 216052, 216053, 216054, 216055, 2160~
## $ name      <chr> "D220X300: 122", "D24X40II: 4132", "HG4000: 181", "D36X50II:~
## $ `VIN/PIN` <chr> "1VRD370C5E1000122", "1VRZ19037E1004132", "1VRC312H8E1000181~
```

```
# This loads in all the different drills D20X-60X.
azure_D20X_60X <- read_csv("Azure_D20-60X.csv")
glimpse(azure_D20X_60X)
```

```
## Rows: 867,889
## Columns: 18
## $ AssetId                              <dbl> 279594, 312266, 307135, 291266, 29~
## $ name                                 <chr> "D20X22III: 2684", "D20X22III: 318~
## $ TroubleCodeTimeStamp                 <dttm> 2022-09-20 12:03:38, 2022-09-20 1~
## $ RedStopFlashLampStatus               <dbl> 3, 3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0~
## $ RedStopLampStatus                    <dbl> 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0~
## $ AmberWarningLampStatus               <dbl> 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1~
## $ AmberWarningFlashLampStatus          <dbl> 0, 0, 3, 3, 3, 3, 3, 3, 3, 0, 3, 3~
## $ ProtectLampStatus                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ ProtectFlashLampStatus               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ MalfunctionIndicatorLampStatus       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ MalfunctionIndicatorFlashLampStatus <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ CanBus                               <dbl> 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0~
## $ SourceAddress                        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ Spn                                  <dbl> 523698, 523698, 523613, 523470, 52~
## $ Fmi                                  <dbl> 11, 11, 1, 14, 14, 14, 6, 3, 4, 12~
## $ LAMPSTATUS                           <chr> "RED", "RED", "AMBER", "AMBER", "A~
## $ ReportType                           <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2~
## $ OccurenceCount                       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

```
work_order1 <- read_csv("Work-order org/work_order.csv")
glimpse(work_order1)
```

```
## Rows: 122,904
## Columns: 7
## $ IDENTIFICATION <chr> "1VRN20074F1000115", "1VRA170V7G1000507", "1VRN20075H10~
## $ MODEL          <chr> "D23X30III", "D20X22III", "D23X30III", "D36X50II", "D23~
## $ WO_SEQ_ID      <dbl> 178340, 179512, 178290, 179853, 178577, 177075, 177358,~
## $ MACHINE_HOURS  <dbl> 1770, 835, 990, 2080, 767, 2146, 177, 598, 672, 86, 0, ~
## $ CREATED_DATE   <chr> "11-Jan-18", "19-Jan-18", "11-Jan-18", "23-Jan-18", "13~
## $ FAILURE_DATE   <chr> "4-Jan-18", "4-Dec-17", "4-Dec-17", "29-Nov-17", "31-Oc~
## $ REPAIR_DATE    <chr> "11-Jan-18", "18-Jan-18", "11-Jan-18", "22-Jan-18", "12~
```

```
work_order2 <- read_csv("Work-order org/work_order2.csv")
glimpse(work_order2)
```

```
## Rows: 425,252
## Columns: 7
## $ IDENTIFICATION    <chr> "1VRA170V8H1000937", "1VRN20073F1000106", "1VRN20073~
## $ MODEL             <chr> "D20X22III", "D23X30III", "D23X30III", "D23X30III", ~
## $ WO_SEQ_ID         <dbl> 180651, 180666, 180666, 180669, 179357, 183040, 1830~
## $ CREATED_DATE      <chr> "25-Jan-18", "25-Jan-18", "25-Jan-18", "25-Jan-18", ~
## $ PART_NUMBER       <chr> "296438394", "296285658", "296411788", "296454327", ~
## $ PART_NAME         <chr> "PLANETARY-MOTOR ASSEMBLY", "CYLINDER 1 X 3-1/2", "P~
## $ `SUM(B.QUANTITY)` <dbl> 1, 1, 1, 1, 4, 1, 1, 1, 2, 1, 1, 1, 17, 3, 1, 1, 2, ~
```

```
work_order3 <- read_csv("Work-order org/work_order3.csv")
glimpse(work_order3)
```

```
## Rows: 174,179
## Columns: 7
## $ IDENTIFICATION  <chr> "1VRN2007XH1000655", "1VR4230D1B1000831", "1VRA170V2H1~
## $ MODEL           <chr> "D23X30III", "D36X50II", "D20X22III", "D24X40III", "D2~
## $ WO_SEQ_ID       <dbl> 183707, 183943, 184163, 184166, 184166, 184167, 184755~
## $ CREATED_DATE    <chr> "7-Feb-18", "8-Feb-18", "8-Feb-18", "8-Feb-18", "8-Feb~
## $ JOB_CODE_NUMBER <chr> "9045V", "8068V", "SB16-135", "A006V-000", "M005V-001"~
## $ JOB_CODE_DESC   <chr> "9045 - Water/Mud Pump Casting/Housing R&R CMP", "8068~
## $ ACTUAL_HOURS    <dbl> 3.54, 1.72, 0.10, 1.25, 4.00, 0.25, 1.72, 1.66, 4.50, ~
```

Prepare and Merge

There is a need to merge these datasets together, and we will do this using the machine VIN as a unique identifier between datasets.

Steps:

- Append VIN to the Azure (telematics) datasets using AssetID as the connection
- Group Azure datasets by VIN and summarize the variables (all the indicator lights) by count
- Group WorkOrder by VIN and summarize number of repairs by number of observations (i.e number of repair jobs done)
- Inner join the grouped Azure dataset onto the grouped WorkOrder dataset

Append VIN to the Azure (telematics) datasets using AssetID as the connection

```
azure_with_vin <- azure_D20X_60X |>
  left_join(vinpin_data |> select(assetId, `VIN/PIN`),
            by = c('AssetId' = 'assetId')) |>
    separate(name, into = c('model', NA), sep = ':', remove = FALSE)
```

Group Azure datasets by VIN and summarize the variables (all the indicator lights) by count

```
azure_summary <- azure_with_vin |>
  group_by(`VIN/PIN`) |>
  summarise(n_red = sum(LAMPSTATUS == "RED", na.rm = TRUE),
            n_amber = sum(LAMPSTATUS == 'AMBER', na.rm = TRUE),
            n_notification = sum(LAMPSTATUS == 'NOTIFICATION', na.rm = TRUE),
            n_malfunction = sum(LAMPSTATUS == 'MALFUNCTION', na.rm = TRUE),
            model = first(model)

  ) |>
  ungroup()
```

Group WorkOrder by VIN and summarize number of repairs by number of observations (i.e number of repair jobs done)

```
all_work_orders <- bind_rows(work_order1, work_order2, work_order3) |>
  mutate(down_time = difftime(lubridate::dmy(REPAIR_DATE),
                              lubridate::dmy(FAILURE_DATE),
                              units = 'days'),
         down_time = as.numeric(down_time))

work_order_summary <- all_work_orders |>
  group_by(IDENTIFICATION) |>
  summarise(n_work_orders = n(),
            median_down_time = median(down_time, na.rm = TRUE),
            max_machine_hours = max(MACHINE_HOURS, na.rm = TRUE))
```

Inner join the grouped Azure dataset onto the grouped WorkOrder dataset

```
summary_by_machine <-
  inner_join(work_order_summary, azure_summary,
             by = c("IDENTIFICATION" = "VIN/PIN"))
glimpse(summary_by_machine)
```

```
## Rows: 2,421
## Columns: 9
## $ IDENTIFICATION    <chr> "1VR3240M0J1000108", "1VR3240M1L1000203", "1VR3240M2~
## $ n_work_orders     <int> 433, 85, 171, 628, 388, 166, 70, 270, 133, 269, 156,~
## $ median_down_time  <dbl> 48.0, 22.5, 10.5, 35.0, 14.0, 14.0, 22.0, 20.0, 14.0~
## $ max_machine_hours <dbl> 3910, 516, 1401, 3854, 1884, 1681, 3836, 5000, 1500,~
## $ n_red             <int> 179, 15, 239, 222, 458, 475, 459, 1303, 484, 226, 11~
## $ n_amber           <int> 414, 28, 45, 317, 102, 210, 113, 1148, 346, 546, 1, ~
## $ n_notification    <int> 223, 22, 30, 656, 124, 124, 111, 1610, 230, 1713, 4,~
## $ n_malfunction     <int> 5, 0, 0, 0, 0, 0, 0, 0, 0, 65, 0, 8, 0, 0, 0, 20, 0,~
## $ model             <chr> "D40X55DRS3", "D40X55DRS3", "D40X55DRS3", "D40X55DRS~
```

Now: There aren't *any* NAs in `work_order_summary` so NAs in the `n_...` columns should actually be 0s: no downtime, no work order at all. So to fill those in:

```
summary_by_machine <- summary_by_machine |>
  # in columns from n_work_orders to median_down_time,
  mutate(across(c(n_work_orders : median_down_time),
                # replace NAs with  0s
                ~replace_na(.x, replace = 0)))

glimpse(summary_by_machine)
```

```
## Rows: 2,421
## Columns: 9
## $ IDENTIFICATION    <chr> "1VR3240M0J1000108", "1VR3240M1L1000203", "1VR3240M2~
## $ n_work_orders     <int> 433, 85, 171, 628, 388, 166, 70, 270, 133, 269, 156,~
## $ median_down_time  <dbl> 48.0, 22.5, 10.5, 35.0, 14.0, 14.0, 22.0, 20.0, 14.0~
## $ max_machine_hours <dbl> 3910, 516, 1401, 3854, 1884, 1681, 3836, 5000, 1500,~
## $ n_red             <int> 179, 15, 239, 222, 458, 475, 459, 1303, 484, 226, 11~
## $ n_amber           <int> 414, 28, 45, 317, 102, 210, 113, 1148, 346, 546, 1, ~
## $ n_notification    <int> 223, 22, 30, 656, 124, 124, 111, 1610, 230, 1713, 4,~
```

```
## $ n_malfunction     <int> 5, 0, 0, 0, 0, 0, 0, 0, 0, 65, 0, 8, 0, 0, 0, 20, 0,~
## $ model             <chr> "D40X55DRS3", "D40X55DRS3", "D40X55DRS3", "D40X55DRS~
```

```
#filter out anomalous cases
summary_by_machine <- summary_by_machine|>
 filter(max_machine_hours < 99999)
```

Interpretation: This our final data set. The final data set holds the columns: `Identification`: vin of the specific machine

`n_work_orders`: count of number of repairs done (our response variable)

`max_machine_hours`: The total number of hours the machine has been running for since its commissioning.

`median_down_time`: median of the downtime (difference of repair date and failure date)

`n_red`: this is the count of the red lamp status

`n_amber`: this is the count of the amber lamp status.

`n_notification`: this is the count of the notification lamp status.

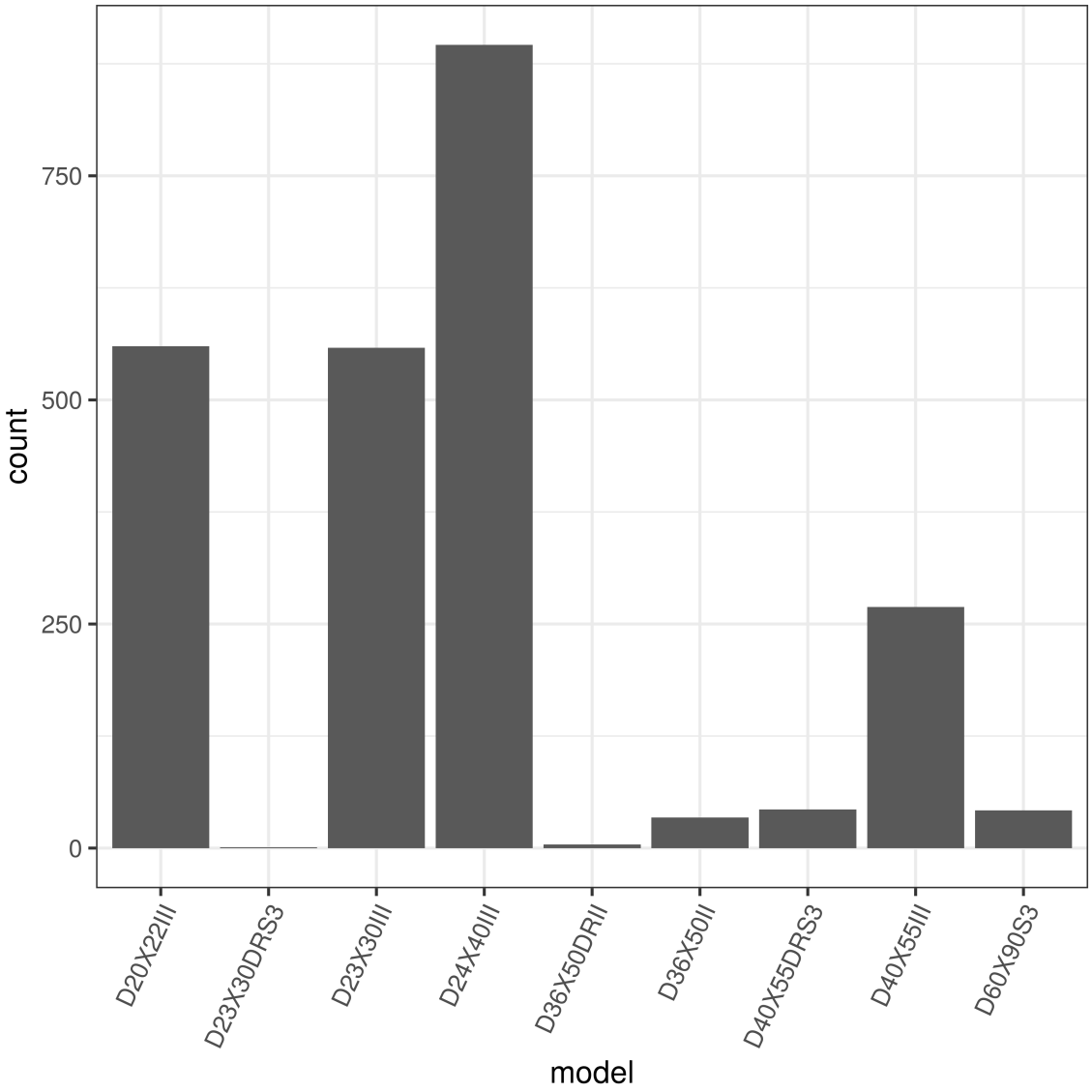
`n_malfunction`: this is the count of the malfunction lamp status.

`model`: the model name of the machine

Exploratory Graphs

Let's check out the number of observations for each type of model

```
summary_by_machine |>
  group_by(model) |>
  summarize(count = n()) |>
  ggplot(aes(x=model, y=count))+
  geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 90))|>
    gf_theme(axis.text.x=element_text(angle=65, hjust=1))
```



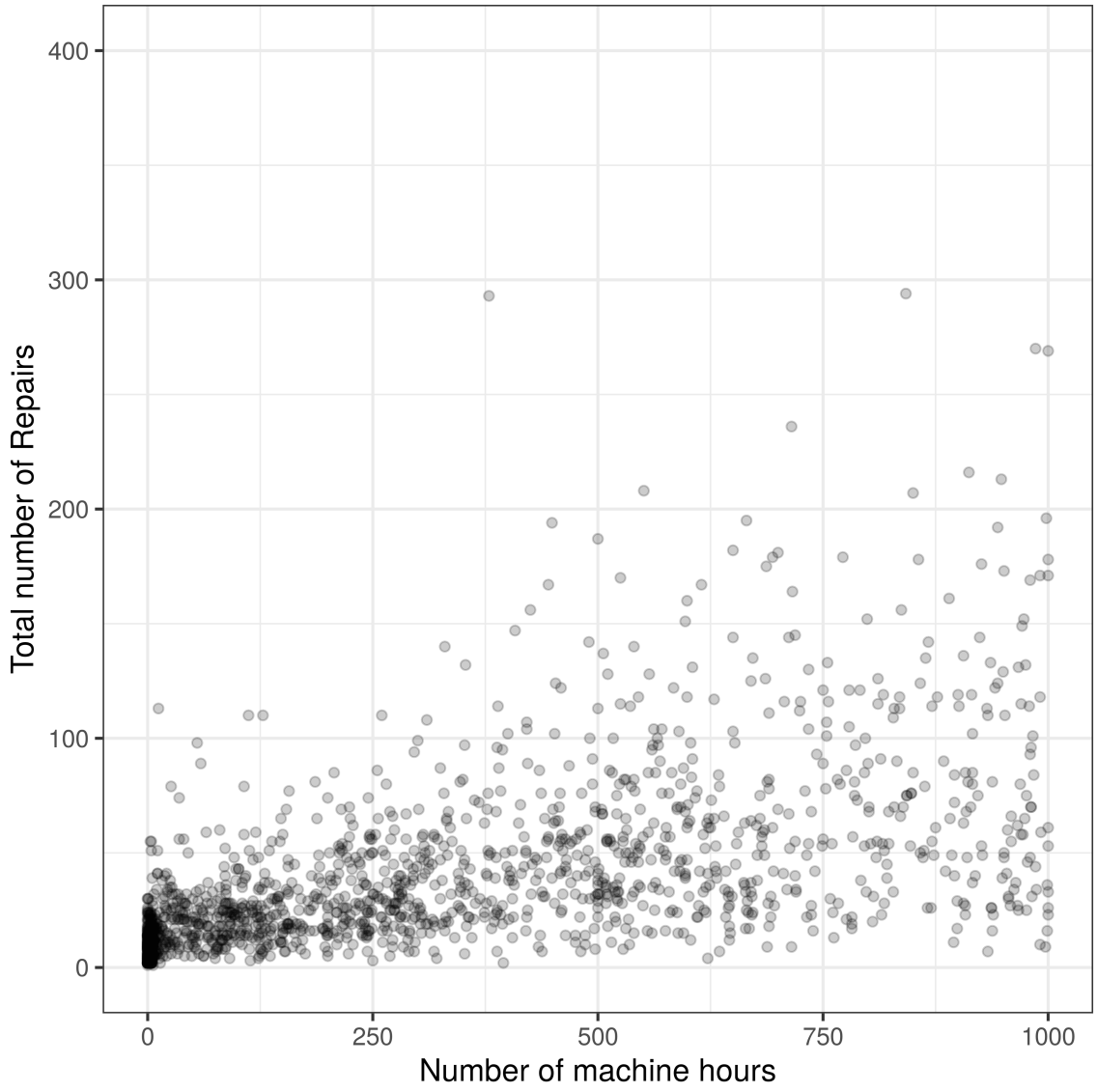
graph one-1.png

Right away we see that we have the most data for the models D24X40III, D23X30III, D20X22III, and D40X55III, and the other models have few observations comparatively.

Next, let's see the relationship between number of machine hours and the number of repairs.

```
#Plot histogram of mean machine hours
ggplot(summary_by_machine, aes(x=max_machine_hours, y=n_work_orders))+
  geom_point(alpha = 0.2) +
  xlim(c(0,1000)) +
  ylim(c(0,400))+
  ylab('Total number of Repairs')+
  xlab('Number of machine hours')
```

```
## Warning: Removed 789 rows containing missing values (`geom_point()`).
```

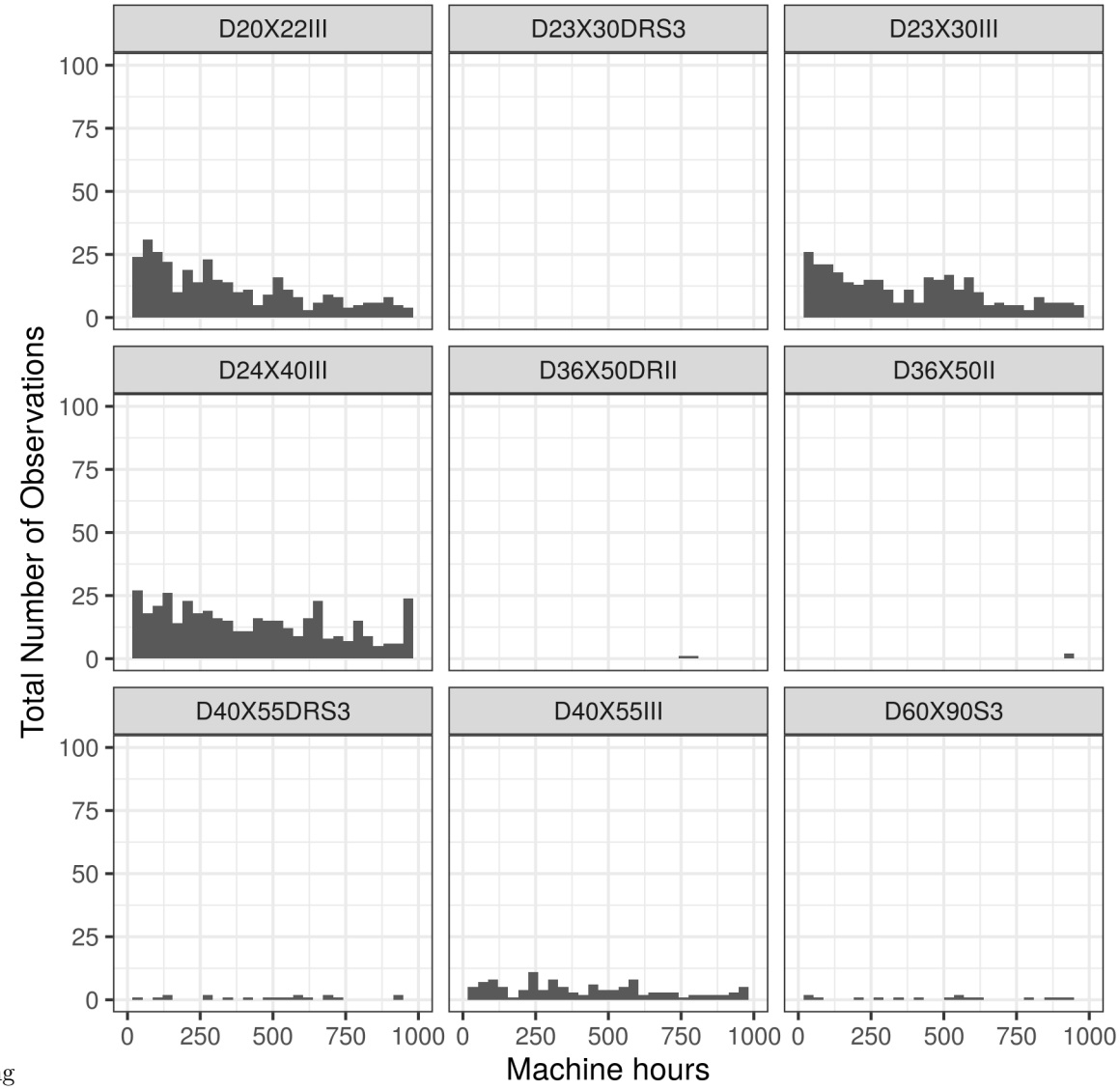


graph two-1.png

We see that generally, the higher the number of machine horus, the higher the total number of repairs. This seems logical, although the relationship does not seem extremely strong (lots of spread). There is also a dense cluster of machines with low machine hours, and low repairs. This means that a lot of our data come from machines that haven't been running for very long.

Next let's see the the number of observations broken down by model and machine hours.

```
ggplot(summary_by_machine)+
  aes(x=max_machine_hours) +
  geom_histogram() +
  xlim(c(0,1000)) +
  ylim(c(0,100))+
  facet_wrap(~model)+
  ylab('Total Number of Observations')+
  xlab('Machine hours')
```

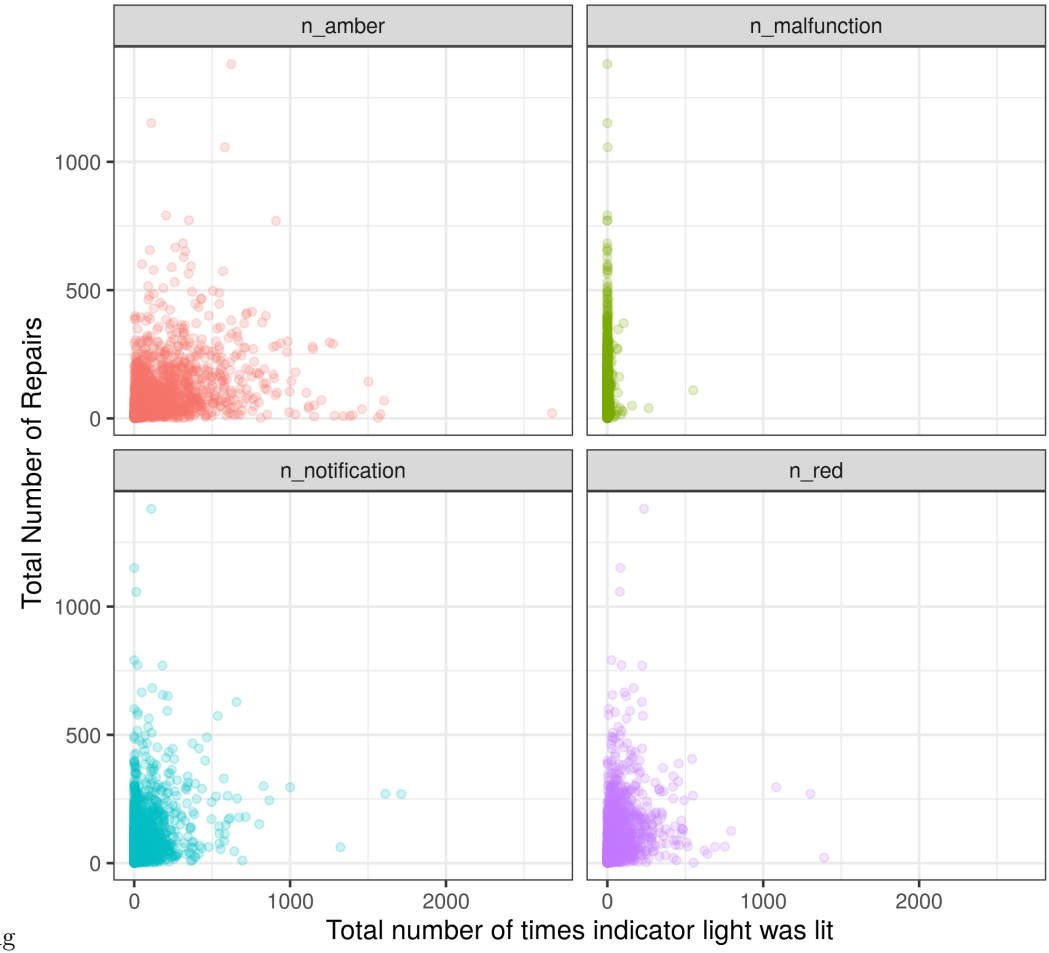


graph three-1.png

We see that what we observed about the dense cluster of machines with low hours and low repairs is explained by the above graph. Generally, across all model types, we have more observations of machines that have less hours, indicated by the downward trend in the histogram.

Next, let's explore how our indicator lamp status (notification, amber, red, malfunction) behaves with the number of repairs.

```
summary_by_machine |>
  select(c('IDENTIFICATION', 'n_work_orders', 'n_red', 'n_amber', 'n_notification', 'n_malfunction', 'm
  pivot_longer(c(n_red,n_amber,n_notification,n_malfunction),
               names_to = 'indicator_light_color',
               values_to = 'indicator_light_count') |>
    ggplot(aes(x= indicator_light_count, y=n_work_orders, color = indicator_light_color))+
  geom_point(alpha = 0.2)+
  facet_wrap(~indicator_light_color)+
  ylab('Total Number of Repairs')+
  labs(colour = "Light Color") +
  xlab('Total number of times indicator light was lit')
```

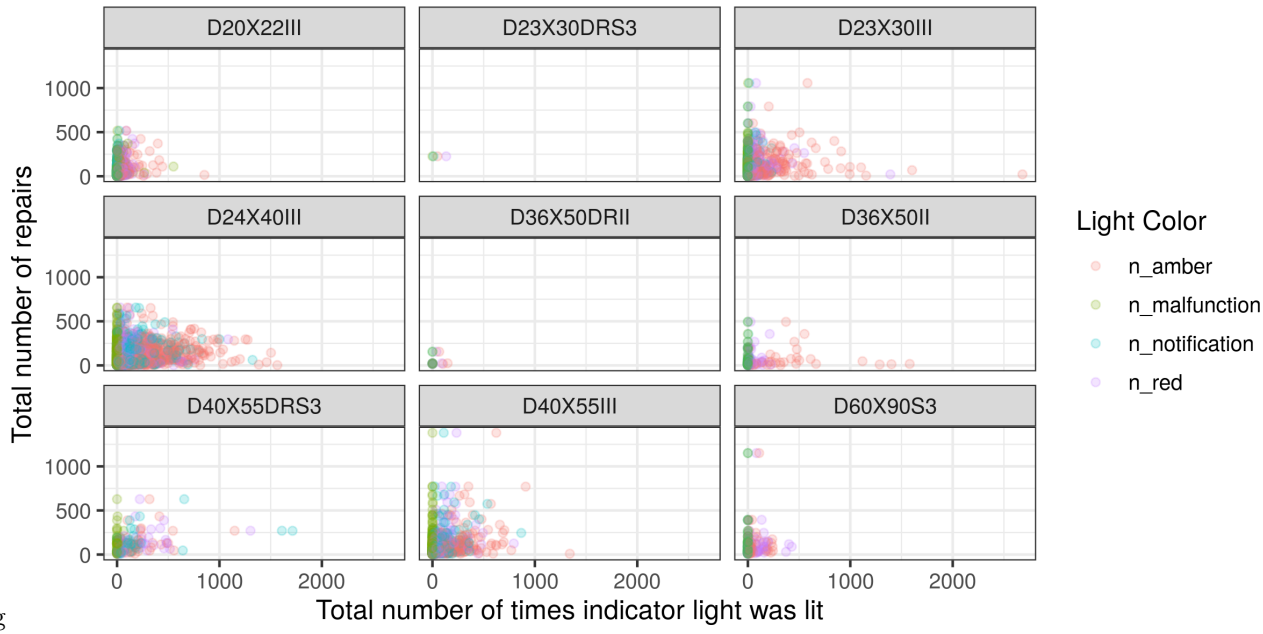


graph four-1.png

From this graph we see that the red, notification, and amber lights behave similarly to one another. However, the malfunction light behaves very differently. It is clear that even when the count of malfunction lights are low, it is associated with high numbers of repairs. This intuitively makes sense since malfunction is the highest severity of indicator lamp status.

Let's plot this graph and break it down by model to see if this is behavior is model-specific:

```
summary_by_machine |>
  select(c('IDENTIFICATION', 'n_work_orders', 'n_red', 'n_amber', 'n_notification', 'n_malfunction', 'm
  pivot_longer(c(n_red,n_amber,n_notification,n_malfunction),
               names_to = 'indicator_light_color',
               values_to = 'indicator_light_count') |>
    ggplot(aes(x= indicator_light_count, y=n_work_orders, color = indicator_light_color))+
  geom_point(alpha = 0.2)+
  facet_wrap(~model) +
  xlab('Total number of times indicator light was lit') +
    labs(colour = "Light Color") +
  ylab('Total number of repairs')
```



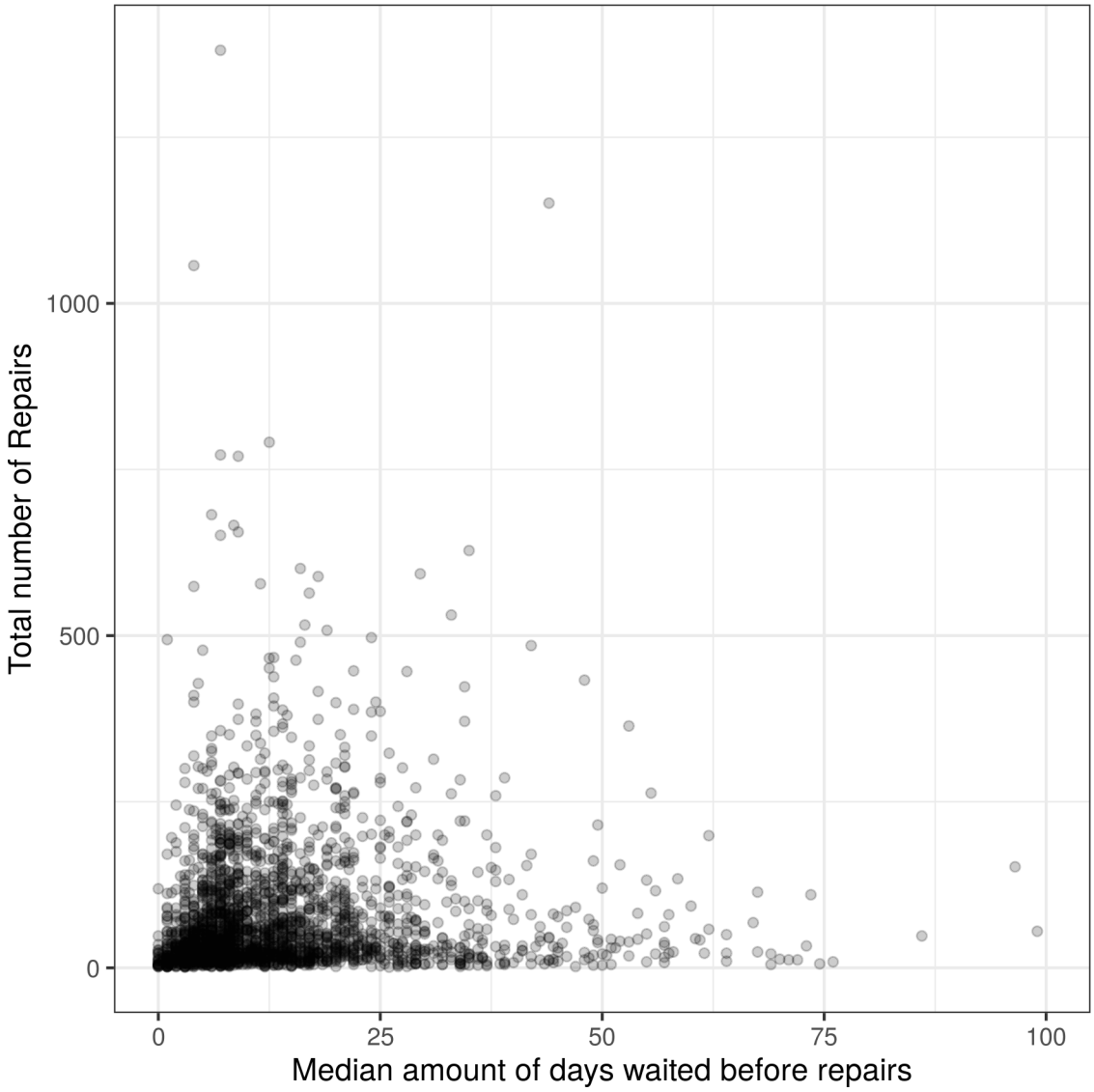
graph five-1.png

From this graph, we see that indicator lights generally behave consistently across all models. I.e there is nothing extremely outof the ordinary we can observe here that shows that a specific model behaves differently than others with regards to how it's indicator light correlates with the number of repairs.

Finally, let's see the relationship of median days waited before repairing and the total number of repairs:

```
#Plot histogram of mean machine hours
ggplot(summary_by_machine, aes(x=median_down_time, y =n_work_orders))+
  geom_point(alpha = 0.2) +
  xlim(c(0,100)) +
  ylab('Total number of Repairs')+
  xlab('Median amount of days waited before repairs')
```

```
## Warning: Removed 8 rows containing missing values (`geom_point()`).
```



This graph shows that there seems to be a an upward trend between 0-12 days, where waiting longer in this bracket leads to significantly more repairs. After the 12 days, the spike mysteriously goes down and tails off. It seems that machines that (on average) are sent into repairs within the bands of 8-15 days are associated with the highest number of repairs.

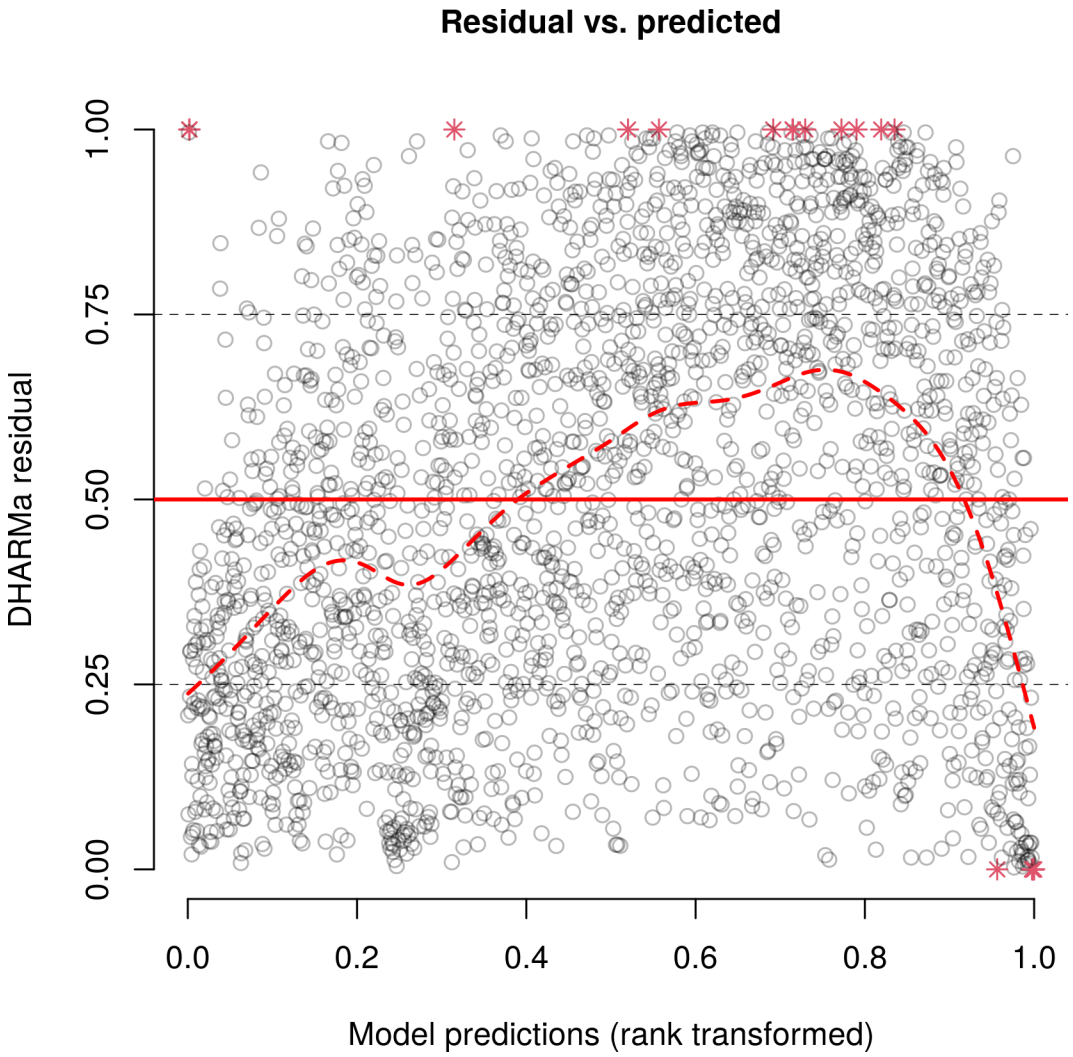
Model Fitting:

We started off with trying to fit a linear negative binomial regression model to model the number of repairs, since it is count data. However, we found that our mean-variance condition and independence of residuals condition fails:

```
#fitting a linear negative binomial regression model
library(glmmTMB)

machine_nb2 <- glmmTMB(n_work_orders ~ max_machine_hours + median_down_time + n_red + n_amber + n_notif
                       data = summary_by_machine,
                       family = nbinom2(link = "log"))
```

Failed Mean-Variance Relationship:

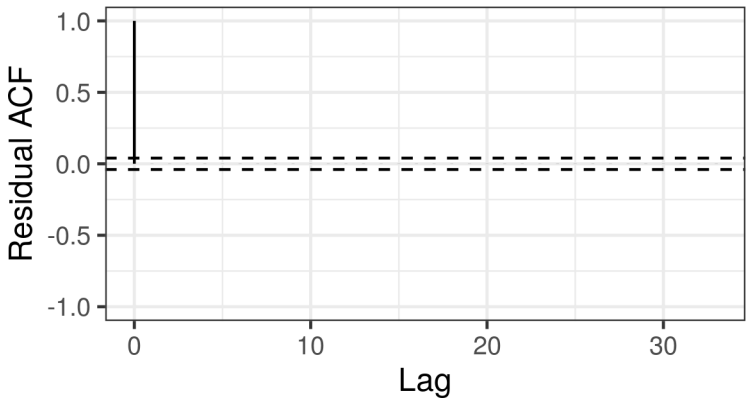
```
library(DHARMa)

nb2_sim <- simulateResiduals(machine_nb2)
plotResiduals(nb2_sim,
              quantreg = FALSE)
```



The scaled-dharma residuals is not uniformly spread vertically, so this condition fails.

Independence of Residuals:

```
s245::gf_acf(~machine_nb2) %>%
  gf_lims(y = c(-1,1))
```



There is a spike at lag 4, so condition fails.

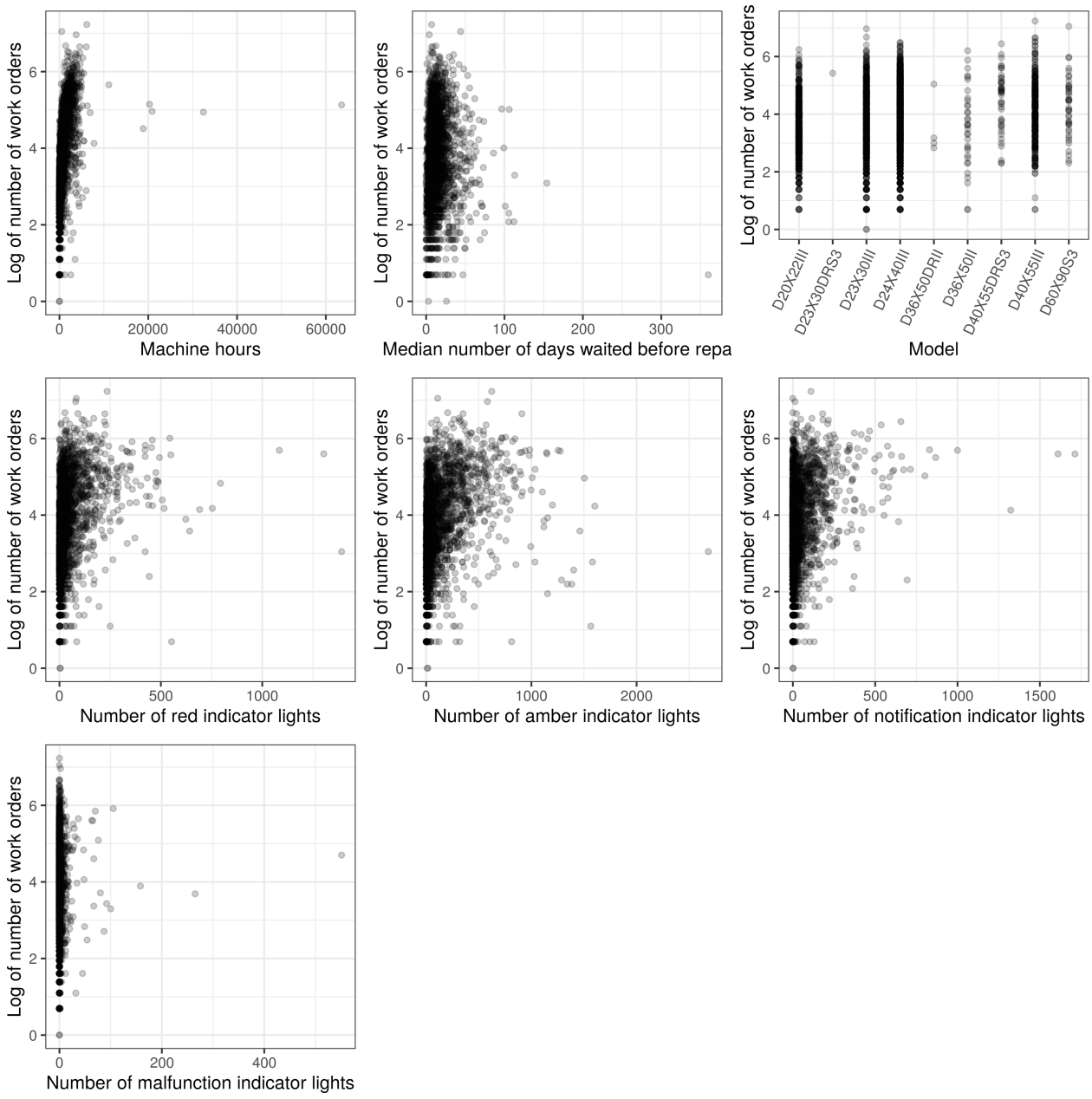
Circling back to our data, we plotted our response and predictors on individual plots and found that it looked like our predictors were behaving non-linearly to our response, as seen in the 'r' like relationship below:

Why we didn't use a linear model

```
loglin1 <- gf_point(log(n_work_orders) ~ max_machine_hours, data = summary_by_machine, alpha = 0.2) |>
  gf_labs(
    x = "Machine hours",
    y = "Log of number of work orders"
  )

loglin2 <- gf_point(log(n_work_orders) ~ median_down_time, data = summary_by_machine, alpha = 0.2) |>
  gf_labs(
    x = "Median number of days waited before repairs",
    y = "Log of number of work orders"
  )

loglin3 <- gf_point(log(n_work_orders) ~ model, data = summary_by_machine, alpha = 0.2) |>
  gf_labs(
    x = "Model",
    y = "Log of number of work orders"
  ) |>
    gf_theme(axis.text.x=element_text(angle=65, hjust=1))

loglin4 <- gf_point(log(n_work_orders) ~ n_red, data = summary_by_machine, alpha = 0.2) |>
  gf_labs(
    x = "Number of red indicator lights",
    y = "Log of number of work orders"
  )

loglin5 <- gf_point(log(n_work_orders) ~ n_amber, data = summary_by_machine, alpha = 0.2) |>
  gf_labs(
    x = "Number of amber indicator lights",
    y = "Log of number of work orders"
  )

loglin6 <- gf_point(log(n_work_orders) ~ n_notification, data = summary_by_machine, alpha = 0.2) |>
  gf_labs(
```

```
    x = "Number of notification indicator lights",
    y = "Log of number of work orders"
  )

loglin7 <- gf_point(log(n_work_orders) ~ n_malfunction, data = summary_by_machine, alpha = 0.2) |>
  gf_labs(
    x = "Number of malfunction indicator lights",
    y = "Log of number of work orders"
  )
```

```
require(ggpubr)
ggarrange(loglin1,loglin2,loglin3,loglin4,loglin5,loglin6,loglin7,
          ncol = 3, nrow = 3
          )
```



Thus, we decided to pivot to model a generalized additive model instead with smooths on all predictors in order to account for the non-linear trend.

Fitting the Model:

```
library(glmmTMB)
require(mgcv)

machine_nb1 <- gam(n_work_orders ~ s(max_machine_hours, k = 6, bs = 'cs') + s(median_down_time, k = 6, b
                       data = summary_by_machine,
                   method = 'ML',
                   select = TRUE,
                   family = nb(link = 'log'))
```

Model summary

```
summary(machine_nb1)
```

```
##
## Family: Negative Binomial(2.367)
## Link function: log
##
## Formula:
## n_work_orders ~ s(max_machine_hours, k = 6, bs = "cs") + s(median_down_time,
##     k = 6, bs = "cs") + s(n_red, k = 6, bs = "cs") + s(n_amber,
##     k = 6, bs = "cs") + s(n_notification, k = 6, bs = "cs") +
##     s(n_malfunction, k = 6, bs = "cs") + model
##
## Parametric coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.99045    0.03050 130.851  < 2e-16 ***
## modelD23X30DRS3  0.29223    0.65556   0.446  0.65577
## modelD23X30III   0.08065    0.04103   1.966  0.04934 *
## modelD24X40III  -0.09339    0.04061  -2.300  0.02147 *
## modelD36X50DRII -0.94134    0.33649  -2.798  0.00515 **
## modelD36X50II   -0.83104    0.12454  -6.673 2.51e-11 ***
## modelD40X55DRS3  0.22578    0.10770   2.096  0.03606 *
## modelD40X55III   0.03921    0.05242   0.748  0.45446
## modelD60X90S3    0.46316    0.10681   4.336 1.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                          edf Ref.df   Chi.sq  p-value
## s(max_machine_hours) 4.9342      5 2822.147  < 2e-16 ***
## s(median_down_time)  4.5720      5   35.566 2.16e-06 ***
## s(n_red)             0.1170      5    0.119   0.3102
## s(n_amber)           1.8603      5    4.733   0.0537 .
## s(n_notification)    0.5444      5    1.150   0.1311
## s(n_malfunction)     0.1938      5    0.234   0.2707
## ---
```

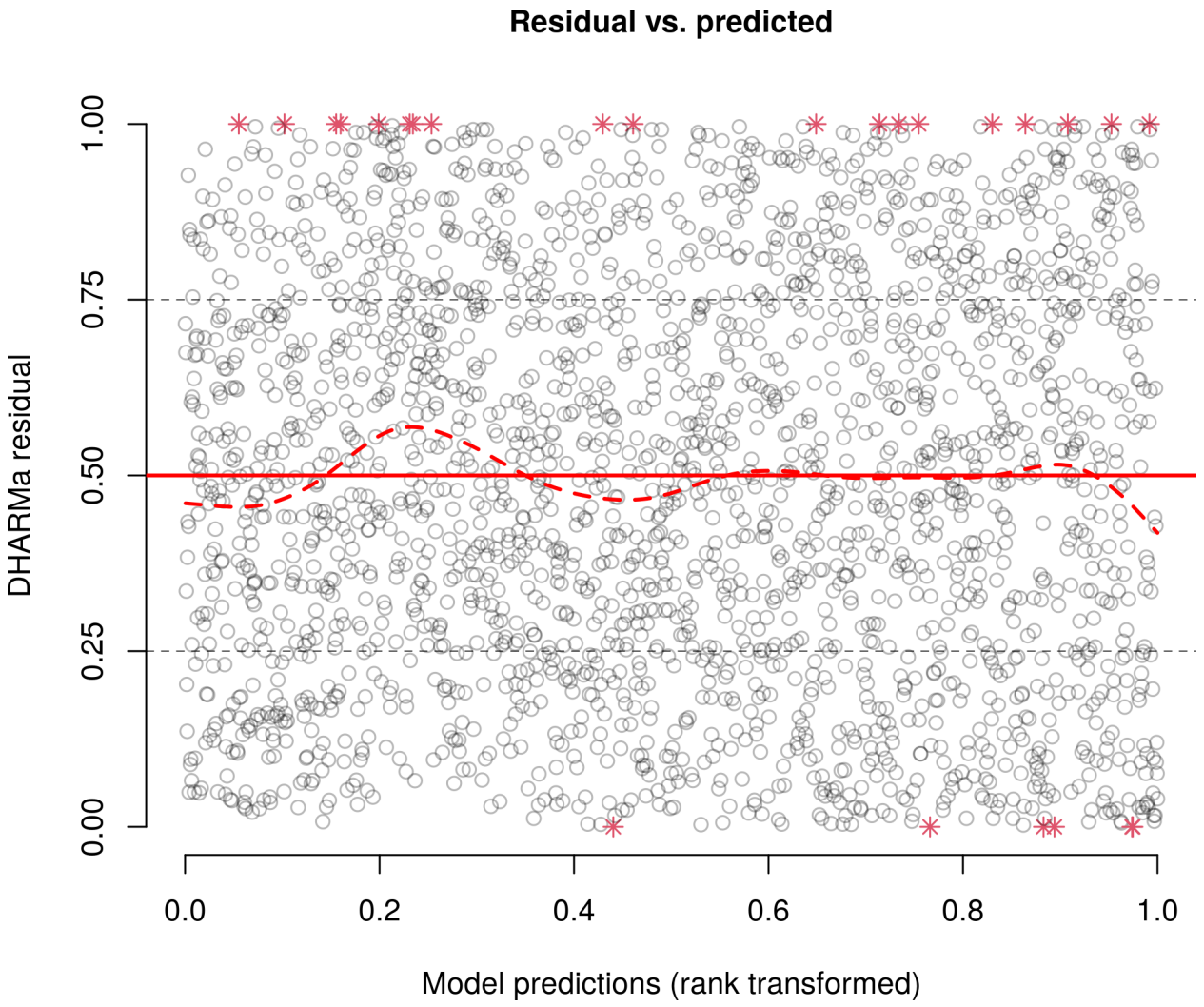
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.381   Deviance explained = 62.8%
## -ML =  11678  Scale est. = 1         n = 2407
```

Model Assessment:

Here, we can see that our GAM model helps solve some of the failed model assessment conditions we had. Both our mean-variance relationship tests and indpendence of residuals tests look good!

Mean-Varince Relationship:

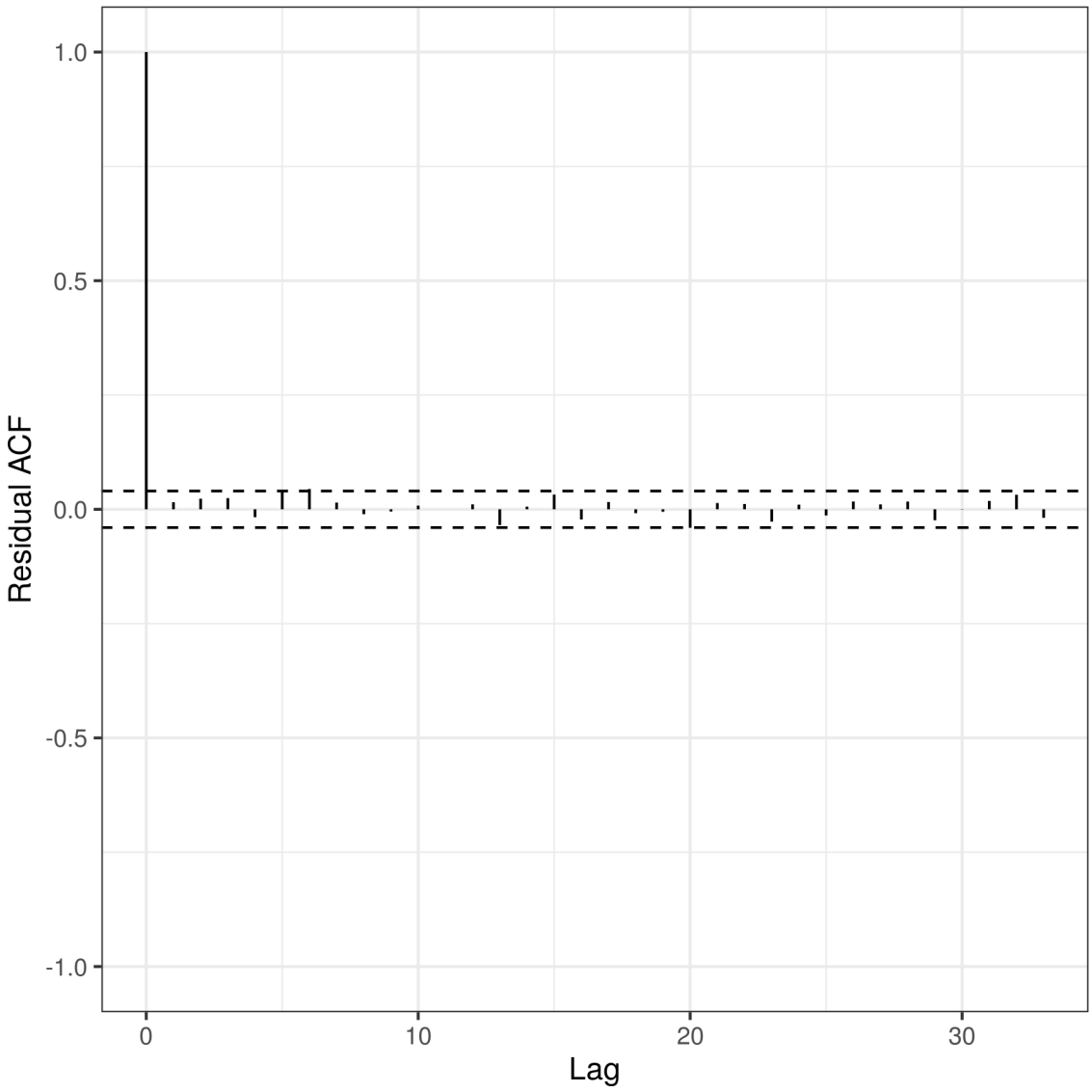
```
library(DHARMa)

nb1_sim <- simulateResiduals(machine_nb1)
plotResiduals(nb1_sim,
              quantreg = FALSE)
```



DHARMA scaled residuals are evenly spread vertically.

Independence of Residuals:

```
s245::gf_acf(~machine_nb1) %>%
  gf_lims(y = c(-1,1))
```



The ACF plot shows that all the lags are within our confidence bounds.

Model Selection

Now that our model assessments have passed, we conducted model selection using akaike's information criterion.

```
library(MuMIn)
machine_nb1 <- update(machine_nb1, na.action = 'na.fail')
head(dredge(machine_nb1, rank='AIC'))
```

```
## Global model call: gam(formula = n_work_orders ~ s(max_machine_hours, k = 6, bs = "cs") +
##     s(median_down_time, k = 6, bs = "cs") + s(n_red, k = 6, bs = "cs") +
```

```
##     s(n_amber, k = 6, bs = "cs") + s(n_notification, k = 6, bs = "cs") +
##     s(n_malfunction, k = 6, bs = "cs") + model, family = nb(link = "log"),
##     data = summary_by_machine, na.action = "na.fail", method = "ML",
##     select = TRUE)
## ---
## Model selection table
##     (Int) mdl s(max_mch_hrs,6,"cs") s(mdn_dwn_tim,6,"cs") s(n_amb,6,"cs")
## 16  3.991   +                     +                      +              +
## 80  3.991   +                     +                      +              +
## 48  3.992   +                     +                      +              +
## 112 3.992   +                     +                      +              +
## 32  3.989   +                     +                      +              +
## 96  3.989   +                     +                      +              +
##     s(n_mlf,6,"cs") s(n_ntf,6,"cs") s(n_red,6,"cs") df    logLik     AIC delta
## 16                                                  22 -11639.41 23323.5  0.00
## 80                                                + 22 -11639.40 23323.5  0.03
## 48                                  +               23 -11638.72 23323.6  0.12
## 112                                 +             + 23 -11638.57 23323.8  0.28
## 32                 +                              22 -11639.20 23323.9  0.36
## 96                 +                            + 22 -11639.20 23323.9  0.36
##     weight
## 16   0.183
## 80   0.180
## 48   0.172
## 112  0.159
## 32   0.153
## 96   0.153
## Models ranked by AIC(x)
```

Interpretation: This is our best model based on the predictors of `max_machine_hours`, `model`, & `median_down_time` `n_amber`.

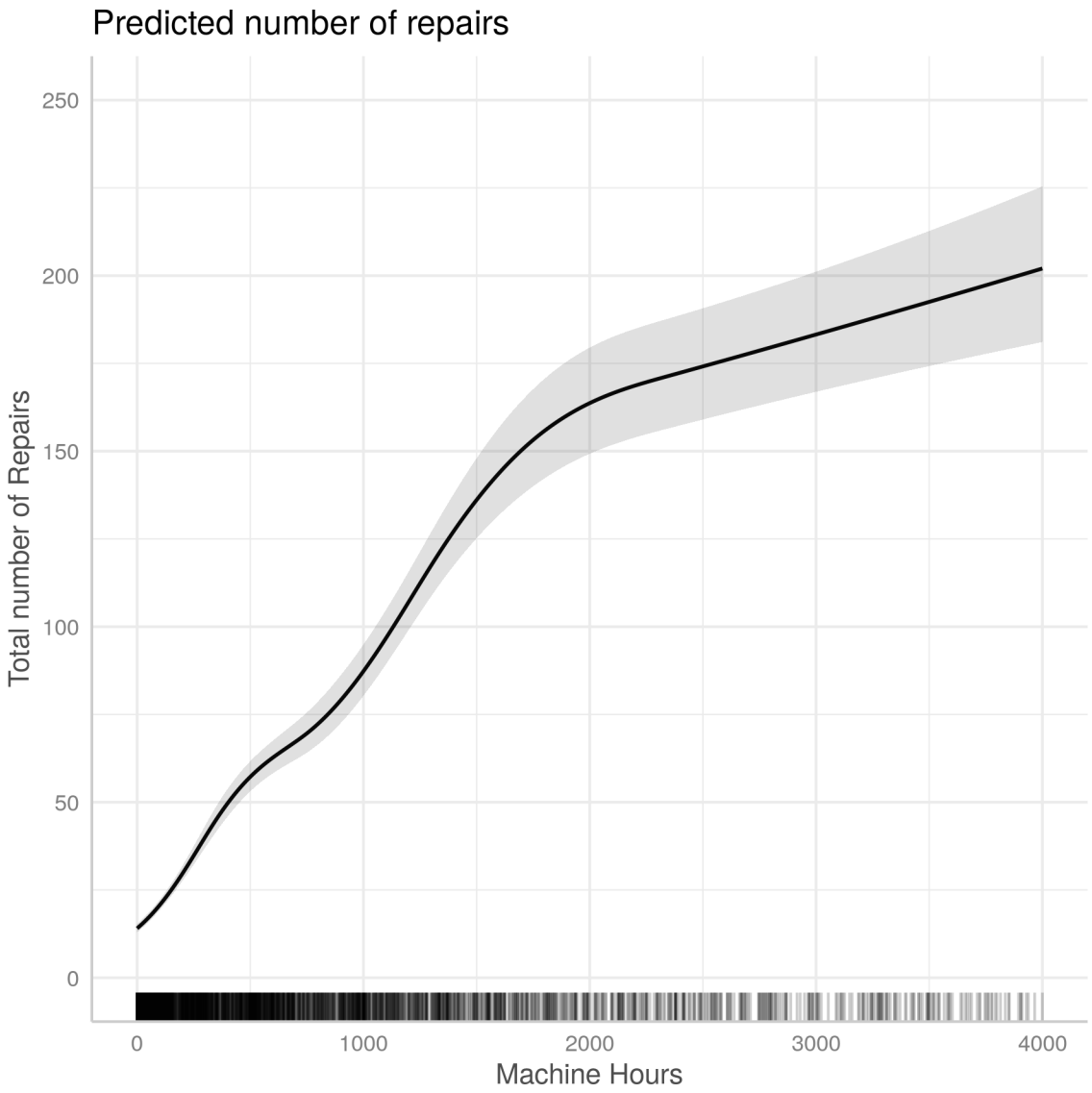
This fit our hypothesis, but we were surprised that it was the amber indicator light, not the malfunction light that is the best predictor of number of repairs.

We plotted some prediction plots for each of these predictors:

Prediction Plots

```
plot(ggeffects::ggpredict(machine_nb1, terms = ('max_machine_hours [n=50]')))|>
  gf_labs(x = "Machine Hours", y = "Total number of Repairs", title = 'Predicted number of repairs')|>
  gf_rugx(~max_machine_hours, data = summary_by_machine, inherit = FALSE, alpha = .2)|>
  gf_lims(x = c(0,4000), y = c(0,250))
```

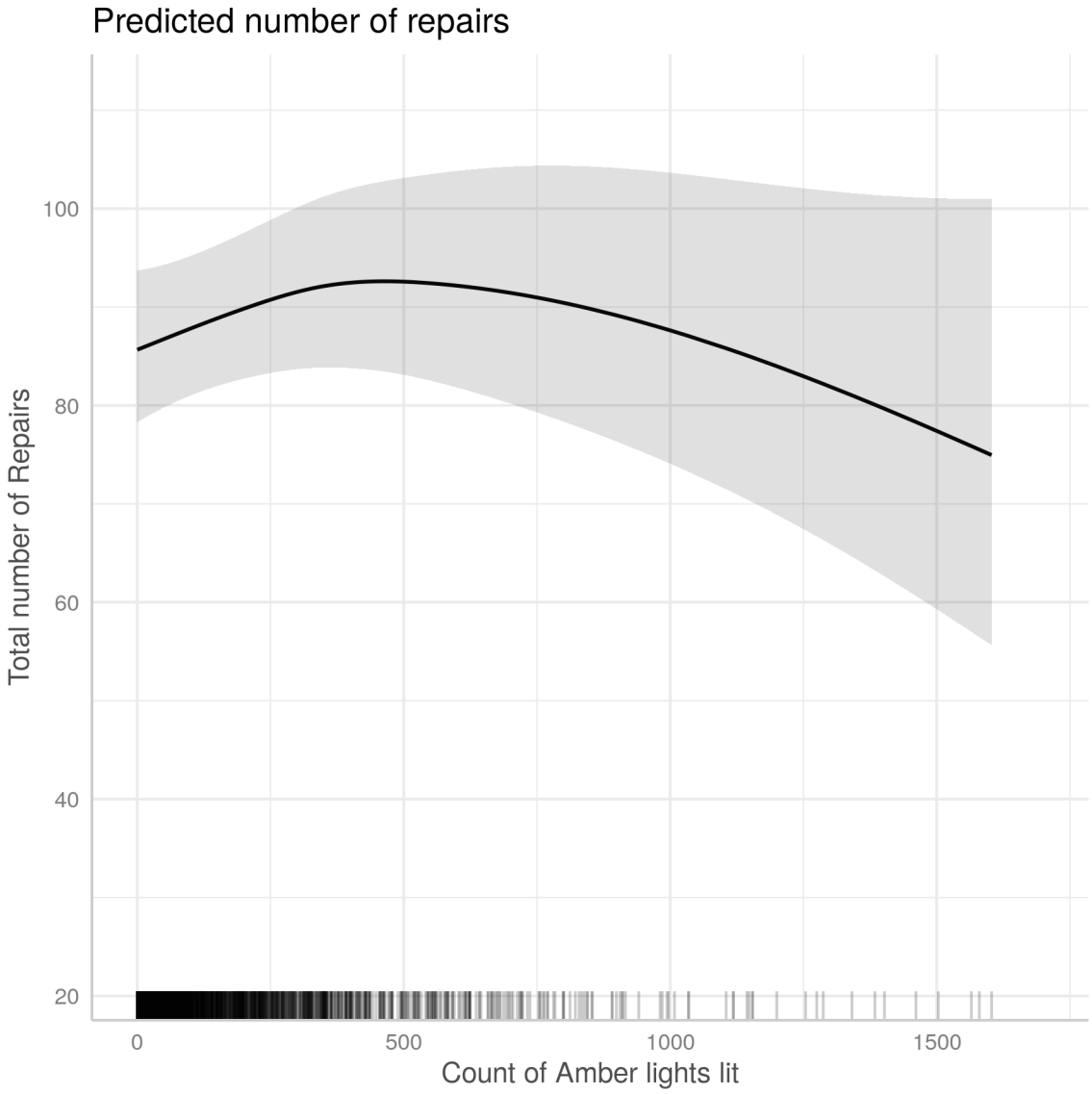
```
## Warning: Removed 119047 rows containing missing values (`geom_line()`).
```



It seems like there is a somewhat linear relationship between number of hours a machine has been running for and the total number of repairs, between 0 to 1800 hours. After which, there seems to be an easing of the rate at which the number of repairs are increasing. However, we have much fewer data points for that range and there is more uncertainty.

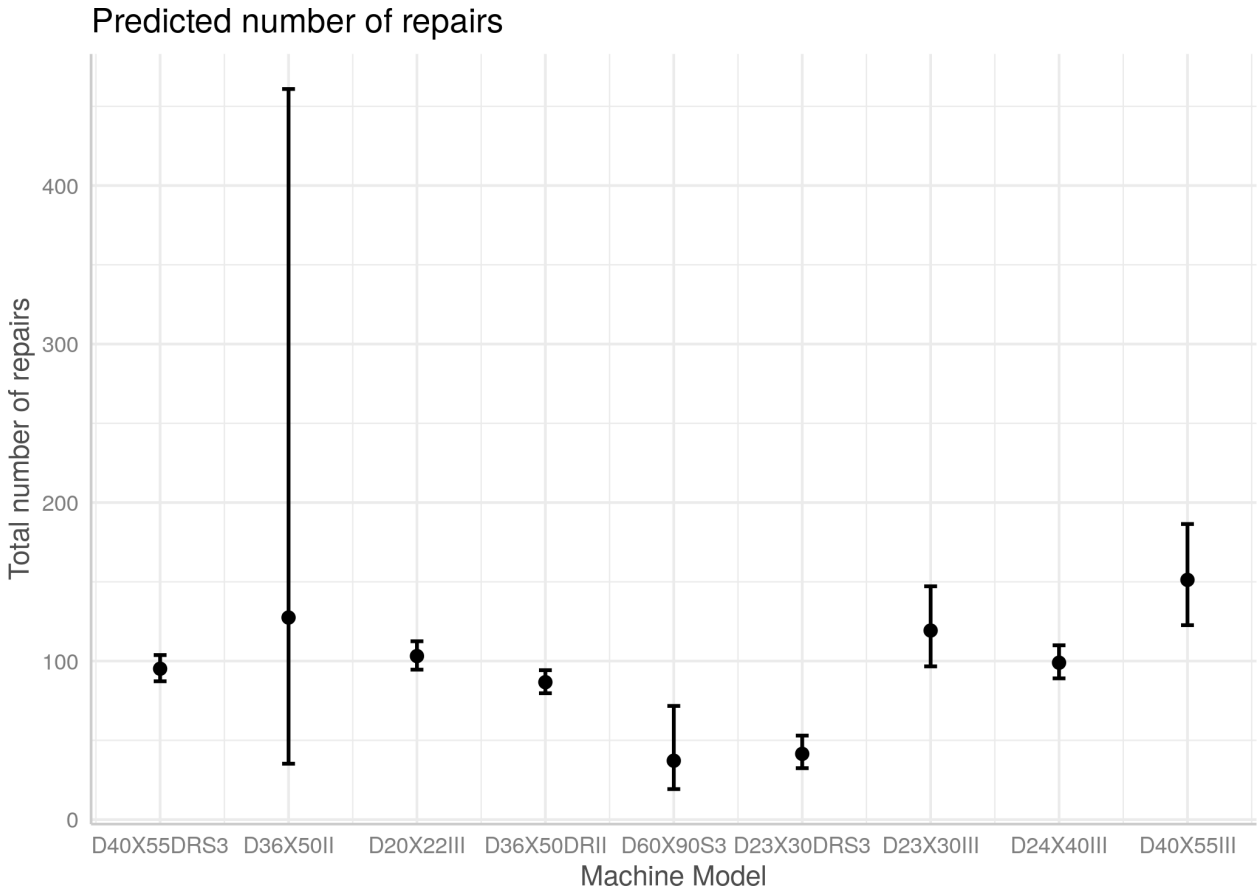
```
plot(ggeffects::ggpredict(machine_nb1, terms = c('n_amber'))) |>
  gf_rugx(~n_amber, data = summary_by_machine, inherit = FALSE, alpha = .2) |>
  gf_labs(x = "Count of Amber lights lit", y = "Total number of Repairs", title = 'Predicted number of
  gf_lims(x = c(0, 1700))
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



The count of amber lights seem to be linearly increasing with the number of repairs from 0 to about 250, after which it dips down. There is a lot of uncertainty in these predictinos, and we have not been able to explain this trend, but it is certainly something to look further into.

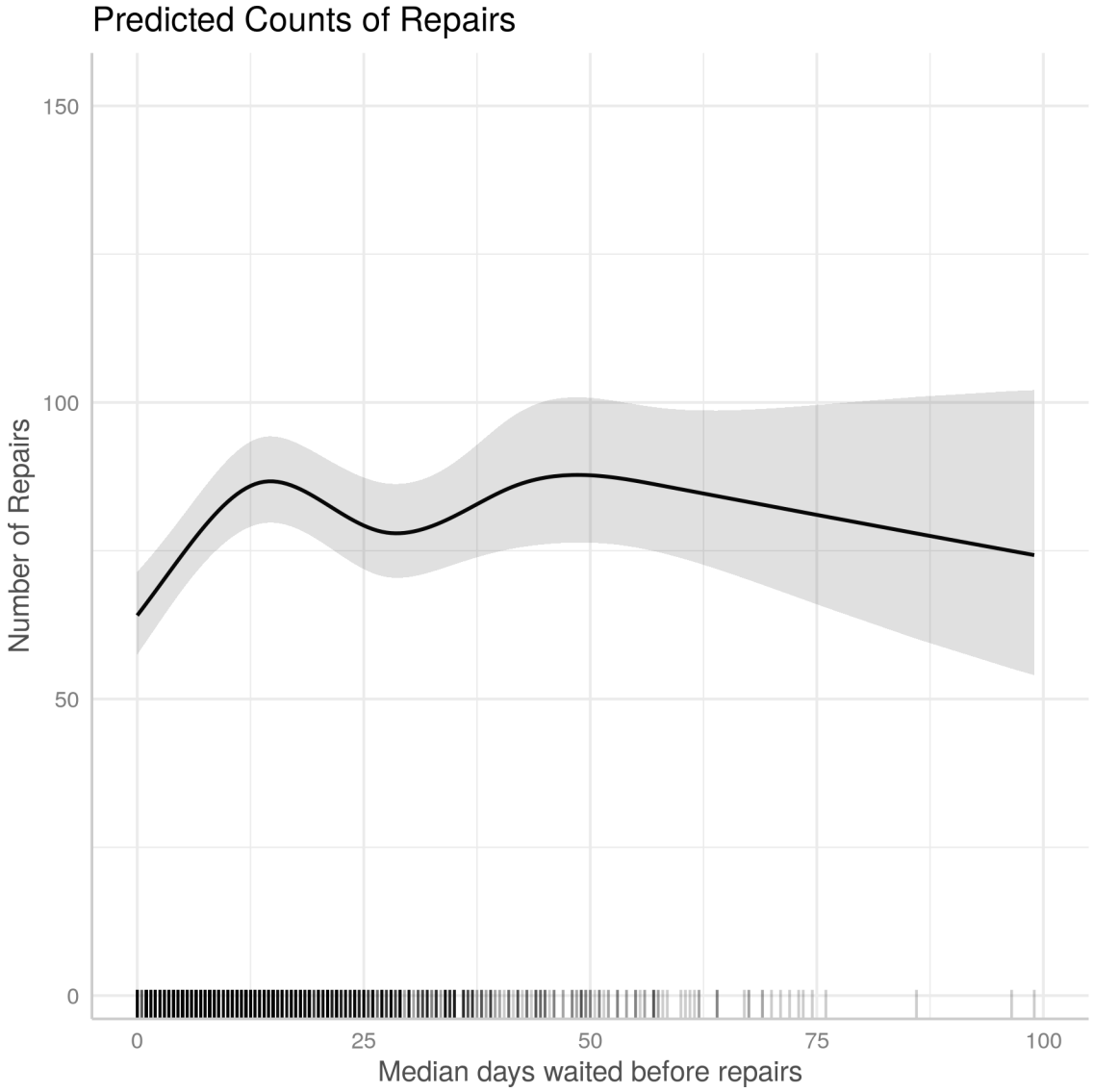
```
plot(ggeffects::ggpredict(machine_nb1, terms = c('model [all]')))+
  xlab('Machine Model')+
  ylab('Total number of repairs')+
  labs(title = 'Predicted number of repairs')
```



It is clear from this plot that the models that best predict the total number of repairs are D40X55III, followed by D23X30III. Vermeer might look into these models to learn more precisely what are causing these models to fail more often than others.

```
plot(ggeffects::ggpredict(machine_nb1, terms = c('median_down_time'))) |>
  gf_rugx(~median_down_time, data = summary_by_machine, inherit = FALSE, alpha = .2) |>
  gf_labs(x = 'Median days waited before repairs', y = 'Number of Repairs', title = 'Predicted Counts o
  gf_lims(x = c(0,100))
```

```
## Warning: Removed 7 rows containing missing values (`geom_line()`).
```



Finally, it seems that waiting between 0 to 12 days before repairing is correlated strongly with a steep increase in the number of repairs, about 25%. While we are painting with broad strokes because of the high-level view of our model, we certainly think it would be interesting to investigate further why this is the case. We also would recommend looking further into why there is a dip after 12 days reducing the number of repairs. Whether these variations are explained through policy of repair procedure, or some other factor, we think that it could be useful to mine the data for more insights, perhaps in search of a maximum down time in which machines can be saved from more repairs in the future.

Conclusion and Future Direction

Overall, our model uses a blunt knife to see how repairs might be correlated to very high level predictors, such as machines, indicator lights, machine hours and down time. We think that doing further investigation into problems in specific model types, and how down time would be a good next step. We also think that the count of indicator lights (amber, red, malfunction and notification) is a weak representation of machine failure, since different combinations of lights can mean different 'categories' of failure (e.g a certain combination of lights can mean an engine failure). Future studies should account for this by taking into account the spn, fmi guides on how combination of lights relate to types of machine failure. This would be a much sharper tool

that can help Vermeer make more tangible business decisions.