CS 1675 PPG Final Project

Sameera Boppana

3/26/2022

Load Packages

```r
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.6     ✓ dplyr   1.0.8
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

Part 1: Exploratory Data Analysis

Read Data

Read in the data from CSV file

```
df <- readr::read_csv('final_project_train.csv', col_names = TRUE)
```

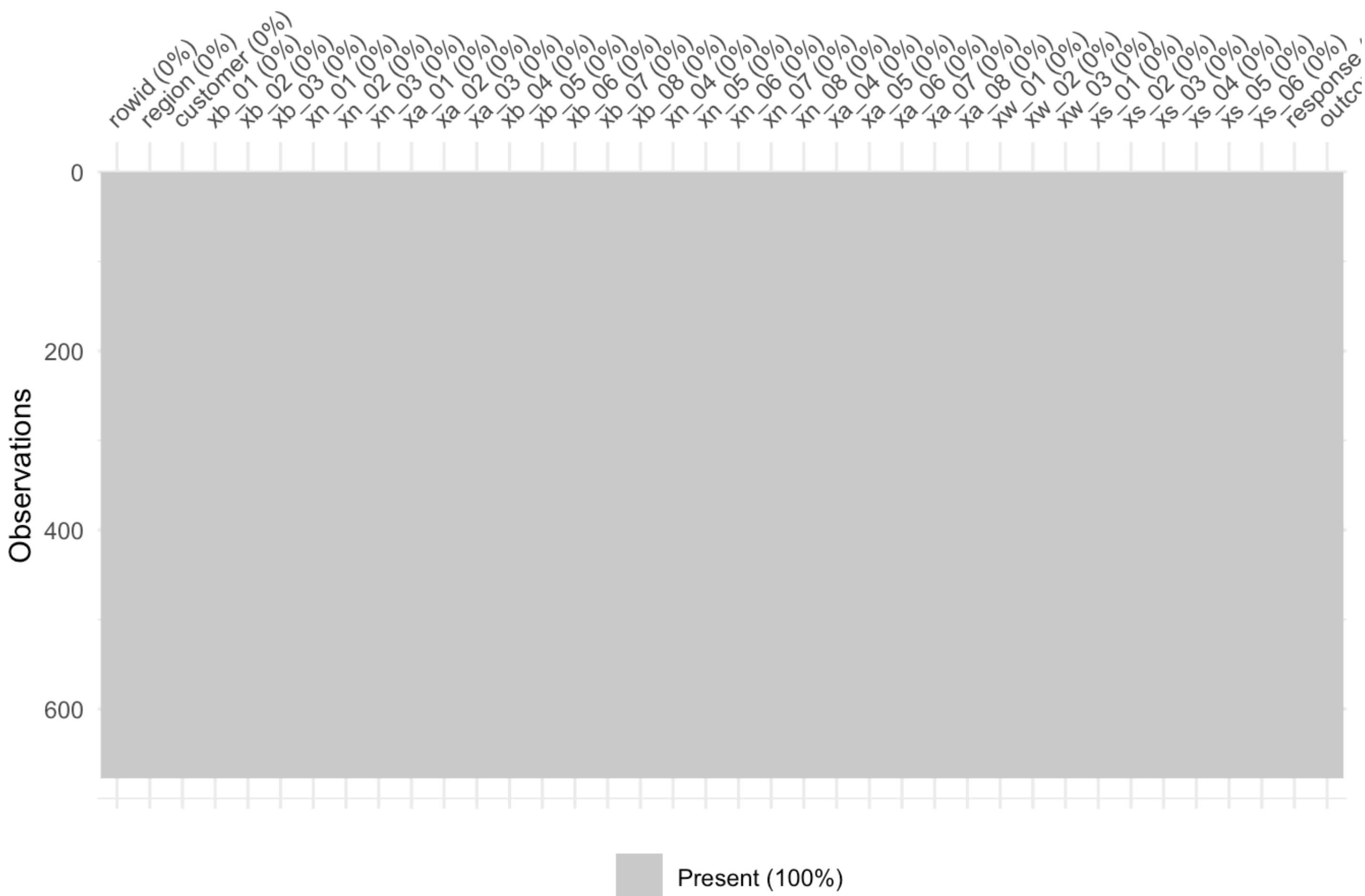
```
## Rows: 677 Columns: 38
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): region, customer, outcome
## dbl (35): rowid, xb_01, xb_02, xb_03, xn_01, xn_02, xn_03, xa_01, xa_02, xa_...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df %>% glimpse()
```

```
## Rows: 677
## Columns: 38
## $ rowid    <dbl> 1, 3, 4, 5, 8, 9, 11, 14, 15, 16, 17, 18, 19, 22, 24, 25, 27,…
## $ region   <chr> "XX", "XX", "XX", "XX", "XX", "XX", "XX", "XX", "XX", "XX", "…
## $ customer <chr> "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "…
## $ xb_01    <dbl> 4.000000, 1.000000, 2.000000, 2.520000, 2.548387, 3.071429, 3…
## $ xb_02    <dbl> 4, 1, 2, 11, 6, 6, 10, 12, 9, 10, 8, 10, 10, 8, 6, 10, 13, 10…
## $ xb_03    <dbl> 4, 1, 2, -6, -1, 1, -4, -4, -2, -4, -2, -2, -2, -4, 1, -4, -3…
## $ xn_01    <dbl> 3.0000000, 2.0000000, 2.0000000, 1.5333333, 0.8387097, 1.8571…
## $ xn_02    <dbl> 3, 2, 4, 9, 3, 8, 6, 10, 10, 4, 6, 8, 9, 5, 7, 12, 12, 6, 6, …
## $ xn_03    <dbl> 3, 2, 0, -3, -4, -2, -5, -6, -3, -5, -3, -6, -4, -3, 0, -5, -…
## $ xa_01    <dbl> 12.000000, 3.000000, 9.000000, 7.080000, 6.451613, 6.857143, …
## $ xa_02    <dbl> 12, 3, 9, 29, 17, 18, 24, 27, 20, 19, 15, 24, 24, 15, 14, 26,…
## $ xa_03    <dbl> 12, 3, 9, -7, -2, 2, -9, -5, -3, -3, -1, 1, -2, -3, 3, -4, -5…
## $ xb_04    <dbl> 1.3333333, 1.0000000, 1.0000000, 0.8950476, 1.2247312, 1.1857…
## $ xb_05    <dbl> 1.3333333, 1.0000000, 1.0000000, -2.0000000, -0.5000000, 0.00…
## $ xb_06    <dbl> 1.333333, 1.000000, 1.000000, 4.000000, 4.000000, 3.000000, 6…
## $ xb_07    <dbl> 4.000000, 1.000000, 2.000000, 1.933333, 1.967742, 1.714286, 1…
## $ xb_08    <dbl> -1.00000000, 1.00000000, 0.00000000, -0.08000000, 0.35483871,…
## $ xn_04    <dbl> 1.0000000, 2.0000000, 1.0000000, 0.5268889, 0.4688172, 0.5607…
## $ xn_05    <dbl> 1.0000000, 2.0000000, 0.0000000, -1.0000000, -1.3333333, -1.0…
## $ xn_06    <dbl> 1.0, 2.0, 2.0, 2.5, 3.0, 2.0, 4.0, 4.0, 3.0, 2.0, 2.0, 2.5, 2…
## $ xn_07    <dbl> 3.000000, 2.000000, 2.500000, 1.493333, 1.225806, 1.642857, 1…
## $ xn_08    <dbl> -1.0000000, 2.0000000, -1.0000000, -0.4400000, -0.4516129, -0…
## $ xa_04    <dbl> 6.000000, 3.000000, 6.750000, 2.425333, 3.023656, 2.685714, 2…
## $ xa_05    <dbl> 6.0000000, 3.0000000, 4.5000000, -3.5000000, -0.6666667, 0.40…
## $ xa_06    <dbl> 6.000000, 3.000000, 9.000000, 9.000000, 13.000000, 6.000000, …
## $ xa_07    <dbl> 9.000000, 3.000000, 7.500000, 4.466667, 4.612903, 4.071429, 4…
## $ xa_08    <dbl> 3.0000000, 3.0000000, 6.0000000, 0.7066667, 1.3225806, 1.3571…
## $ xw_01    <dbl> 23.00000, 17.00000, 52.50000, 64.52564, 54.75758, 58.33333, 6…
## $ xw_02    <dbl> 23, 17, 48, 0, 12, 15, 0, 0, 0, 7, 14, 0, 0, 0, 8, 8, 0, 4, 2…
## $ xw_03    <dbl> 23, 17, 57, 106, 105, 101, 107, 109, 109, 104, 109, 99, 103, …
## $ xs_01    <dbl> 0.262073307, 0.330804757, 0.239795763, 0.142106837, 0.2442957…
## $ xs_02    <dbl> 0.26207331, 0.33080476, 0.19049123, -0.73321509, -0.12204299,…
## $ xs_03    <dbl> 0.2620733, 0.3308048, 0.2891003, 0.5500723, 1.3134719, 0.6540…
## $ xs_04    <dbl> 0.5375576, 0.4286607, 0.3676937, 0.2865445, 0.2375470, 0.2594…
## $ xs_05    <dbl> 0.5375575604, 0.4286607050, 0.2485001680, 0.0000000000, 0.043…
## $ xs_06    <dbl> 0.5375576, 0.4286607, 0.4868872, 0.6357541, 0.4327004, 0.8672…
## $ response <dbl> 2.617991, 1.184632, 2.216626, 2.726715, 1.483323, 2.039279, 1…
## $ outcome  <chr> "non_event", "non_event", "event", "non_event", "non_event", …
```

```
visdat::vis_miss(df)
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generate
d.
```



There are no missing values present in the data.

```
visdat::vis_dat(df)
```

The data types present within the data set are either character or numeric.

```
df %>% purrr::map_dbl(n_distinct)
```

```
##     rowid    region  customer     xb_01     xb_02     xb_03     xn_01     xn_02
##       677         3         9       229        19        21       225        18
##     xn_03     xa_01     xa_02     xa_03     xb_04     xb_05     xb_06     xb_07
##        18       257        38        35       364        59        51       181
##     xb_08     xn_04     xn_05     xn_06     xn_07     xn_08     xa_04     xa_05
##       187       360        51        47       174       174       411        87
##     xa_06     xa_07     xa_08     xw_01     xw_02     xw_03     xs_01     xs_02
##        87       213       212       396       102       103       676       644
##     xs_03     xs_04     xs_05     xs_06  response   outcome
##       672       676       663       676       677         2
```

There are a wide range of distinct values.

```
df %>% count(outcome)
```

```
## # A tibble: 2 × 2
##   outcome       n
##   <chr>     <int>
## 1 event       127
## 2 non_event   550
```

```
df %>% count(region)
```

```
## # A tibble: 3 × 2
##   region     n
##   <chr>  <int>
## 1 XX       161
## 2 YY       222
## 3 ZZ       294
```

```
df %>% count(customer)
```

```
## # A tibble: 9 × 2
##   customer     n
##   <chr>    <int>
## 1 A           55
## 2 B           52
## 3 D           32
## 4 E           35
## 5 G          113
## 6 K           38
## 7 M           71
## 8 Other      245
## 9 Q           36
```

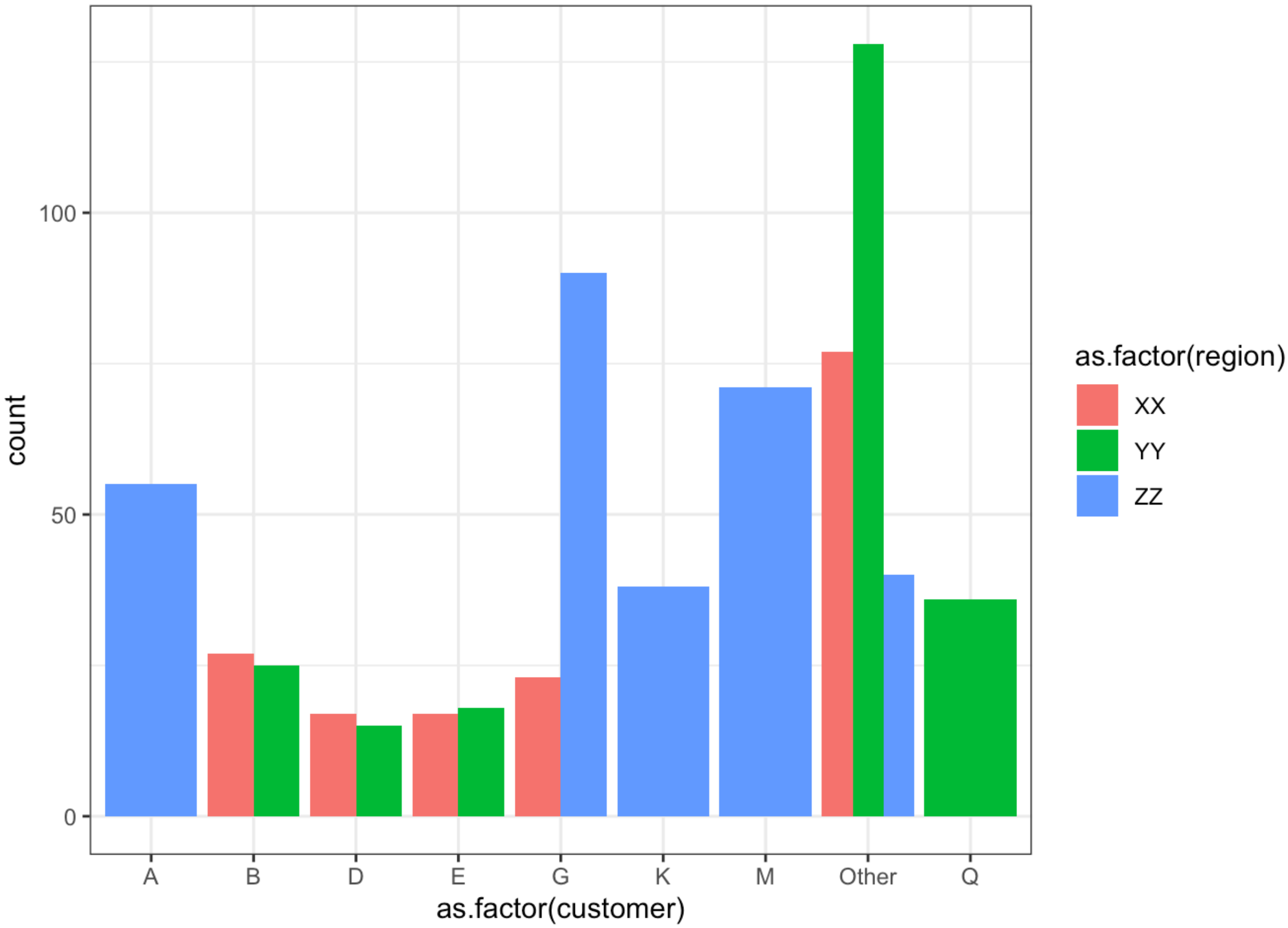
```
df %>% count(customer, region)
```

```
## # A tibble: 15 × 3
##    customer region     n
##    <chr>    <chr>  <int>
##  1 A        ZZ        55
##  2 B        XX        27
##  3 B        YY        25
##  4 D        XX        17
##  5 D        YY        15
##  6 E        XX        17
##  7 E        YY        18
##  8 G        XX        23
##  9 G        ZZ        90
## 10 K        ZZ        38
## 11 M        ZZ        71
## 12 Other    XX        77
## 13 Other    YY       128
## 14 Other    ZZ        40
## 15 Q        YY        36
```

All A customers are from region ZZ. All K customers are from region ZZ.

Visualizing the relationship between customer and region

```
df %>%
  ggplot(mapping = aes(x = as.factor(customer))) +
  geom_bar(mapping = aes(fill = as.factor(region)), position = "dodge") +
  theme_bw()
```



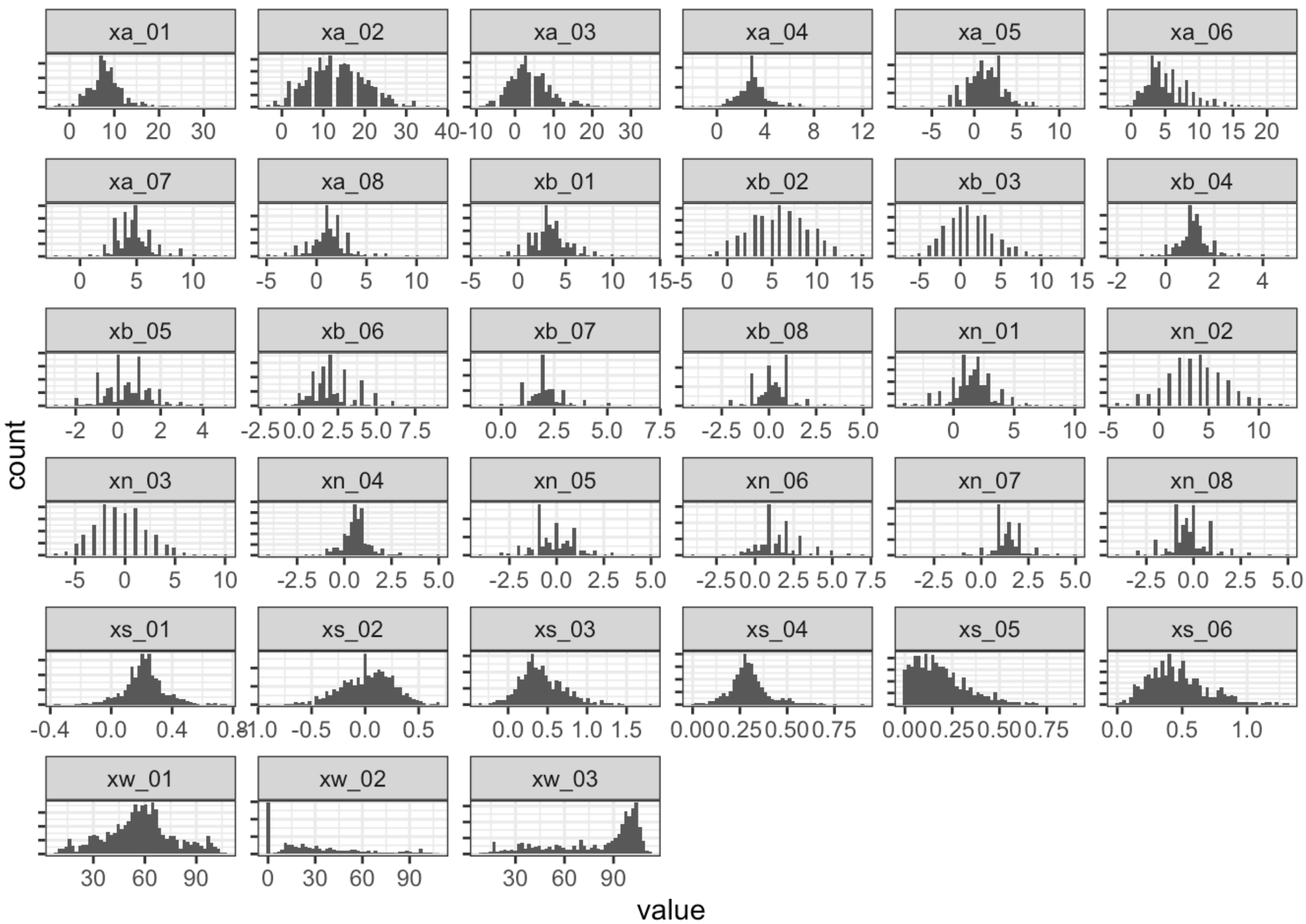
Continous Input Distributions

Examining the distributions of the continous inputs

```
continuous_vars <- colnames(df)
continuous_vars <- continuous_vars[-1:-3]
continuous_vars <- continuous_vars[-length(continuous_vars)]
continuous_vars <- continuous_vars[-length(continuous_vars)]
continuous_vars
```

```
##  [1] "xb_01" "xb_02" "xb_03" "xn_01" "xn_02" "xn_03" "xa_01" "xa_02" "xa_03"
## [10] "xb_04" "xb_05" "xb_06" "xb_07" "xb_08" "xn_04" "xn_05" "xn_06" "xn_07"
## [19] "xn_08" "xa_04" "xa_05" "xa_06" "xa_07" "xa_08" "xw_01" "xw_02" "xw_03"
## [28] "xs_01" "xs_02" "xs_03" "xs_04" "xs_05" "xs_06"
```

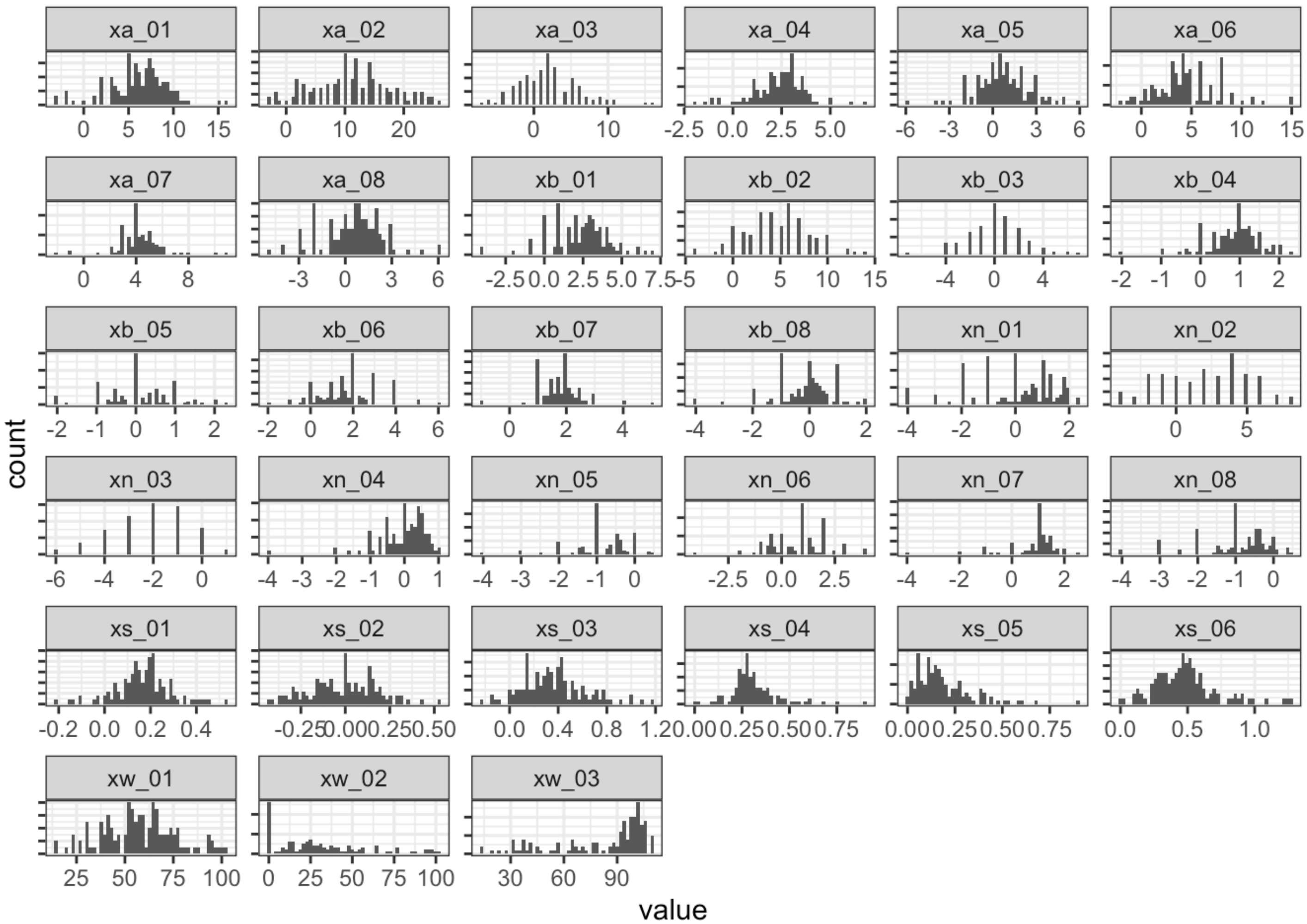
```
df %>%
  select(all_of(continuous_vars)) %>%
  tibble::rowid_to_column() %>%
  pivot_longer(!c("rowid")) %>%
  ggplot(mapping = aes(x = value)) +
  geom_histogram(bins = 50) +
  facet_wrap(~name, scales = "free") +
  theme_bw() +
  theme(axis.text.y = element_blank())
```



Examining the distribution of the continuous inputs filtering by outcome type

Outcome = Event

```
df %>%
  filter(outcome == "event") %>%
  select(all_of(continuous_vars)) %>%
  tibble::rowid_to_column() %>%
  pivot_longer(!c("rowid")) %>%
  ggplot(mapping = aes(x = value)) +
  geom_histogram(bins = 50) +
  facet_wrap(~name, scales = "free") +
  theme_bw() +
  theme(axis.text.y = element_blank())
```




```
df %>%
  filter(outcome == "event") %>%
  select(all_of(continuous_vars)) %>% summary()
```

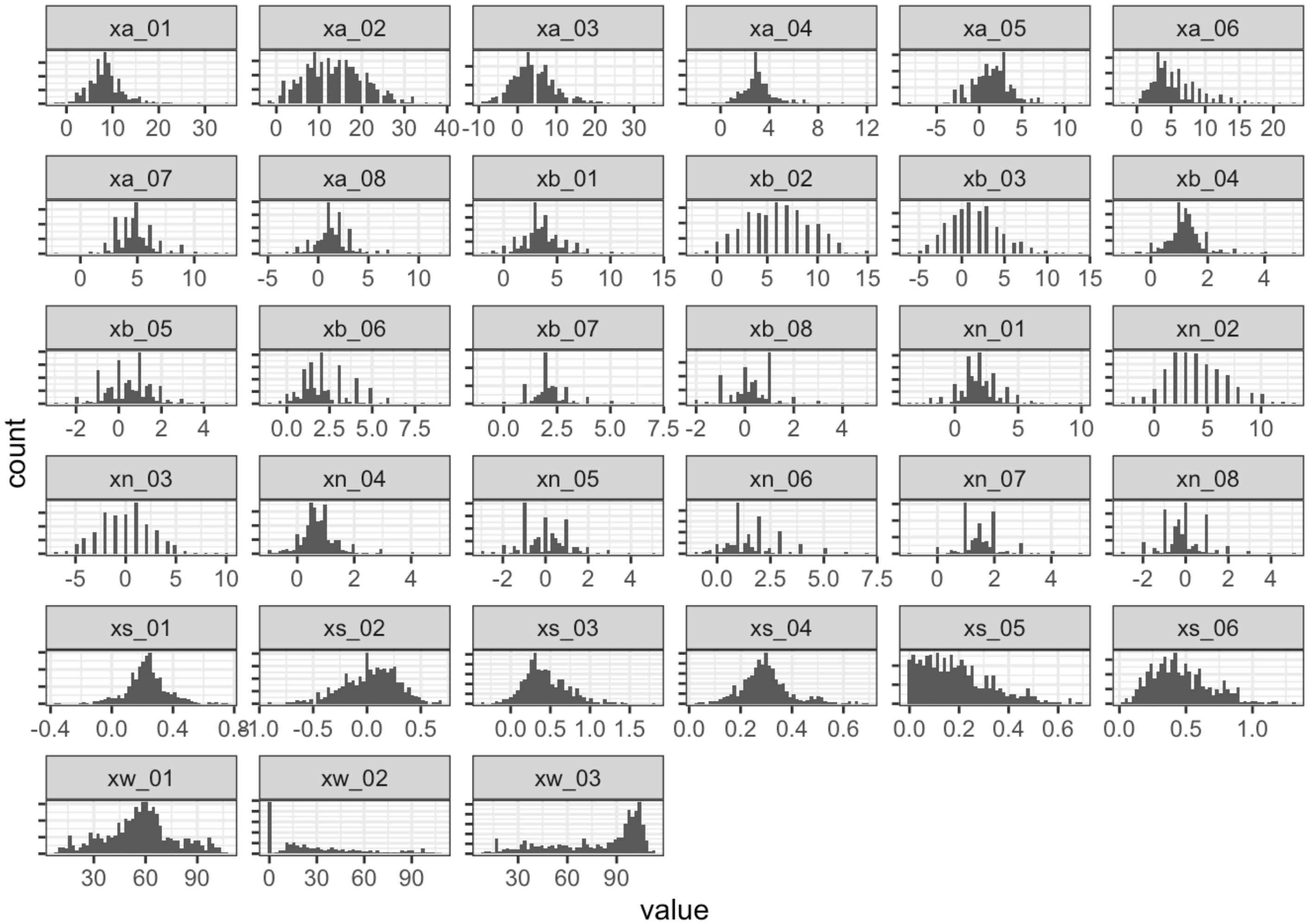
```
##      xb_01            xb_02            xb_03              xn_01
##  Min.   :-4.000   Min.   :-4.000   Min.   :-7.0000   Min.   :-4.00000
##  1st Qu.: 1.050   1st Qu.: 2.500   1st Qu.:-1.0000   1st Qu.:-1.00000
```

```
##  Median : 2.571   Median : 5.000   Median : 0.0000   Median : 0.33333  
##  Mean   : 2.336   Mean   : 4.669   Mean   : 0.1811   Mean   : 0.01976  
##  3rd Qu.: 3.400   3rd Qu.: 7.000   3rd Qu.: 1.5000   3rd Qu.: 1.10238  
##  Max.   : 7.000   Max.   :14.000   Max.   : 7.0000   Max.   : 2.33333  
##      xn_02            xn_03            xa_01            xa_02      
##  Min.   :-4.000   Min.   :-6.000   Min.   :-3.000   Min.   :-3.00  
##  1st Qu.:-0.500   1st Qu.:-3.000   1st Qu.: 4.817   1st Qu.: 7.00  
##  Median : 2.000   Median :-2.000   Median : 6.400   Median :11.00  
##  Mean   : 2.071   Mean   :-2.126   Mean   : 6.161   Mean   :11.17  
##  3rd Qu.: 4.000   3rd Qu.:-1.000   3rd Qu.: 8.000   3rd Qu.:15.00  
##  Max.   : 8.000   Max.   : 1.000   Max.   :16.000   Max.   :26.00  
##      xa_03            xb_04             xb_05             xb_06       
##  Min.   :-7.000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.000  
##  1st Qu.:-1.000   1st Qu.: 0.5000   1st Qu.:-0.5000   1st Qu.: 1.000  
##  Median : 2.000   Median : 0.9722   Median : 0.0000   Median : 1.667  
##  Mean   : 1.953   Mean   : 0.8315   Mean   : 0.0790   Mean   : 1.750  
##  3rd Qu.: 4.000   3rd Qu.: 1.2069   3rd Qu.: 0.6333   3rd Qu.: 2.500  
##  Max.   :16.000   Max.   : 2.3333   Max.   : 2.3333   Max.   : 6.000  
##      xb_07            xb_08             xn_04               xn_05        
##  Min.   :-1.000   Min.   :-4.0000   Min.   :-4.00000   Min.   :-4.0000  
##  1st Qu.: 1.396   1st Qu.:-0.7667   1st Qu.:-0.35417   1st Qu.:-1.0000  
##  Median : 1.812   Median : 0.0000   Median : 0.12500   Median :-1.0000  
##  Mean   : 1.794   Mean   :-0.1639   Mean   :-0.01121   Mean   :-0.8656  
##  3rd Qu.: 2.099   3rd Qu.: 0.3661   3rd Qu.: 0.44097   3rd Qu.:-0.3667  
##  Max.   : 5.000   Max.   : 2.0000   Max.   : 1.00000   Max.   : 0.5000  
##      xn_06             xn_07             xn_08             xa_04       
##  Min.   :-4.0000   Min.   :-4.0000   Min.   :-4.0000   Min.   :-2.000  
##  1st Qu.:-0.1000   1st Qu.: 0.8333   1st Qu.:-1.1944   1st Qu.: 1.736  
##  Median : 1.0000   Median : 1.0000   Median :-0.8000   Median : 2.581  
##  Mean   : 0.8053   Mean   : 0.8639   Mean   :-0.9634   Mean   : 2.375  
##  3rd Qu.: 1.7083   3rd Qu.: 1.3333   3rd Qu.:-0.3632   3rd Qu.: 3.099  
##  Max.   : 4.0000   Max.   : 2.5000   Max.   : 0.5000   Max.   : 6.750  
##      xa_05             xa_06            xa_07            xa_08        
##  Min.   :-6.0000   Min.   :-2.000   Min.   :-2.000   Min.   :-5.0000  
##  1st Qu.:-0.2917   1st Qu.: 2.875   1st Qu.: 3.500   1st Qu.:-0.5000  
##  Median : 0.6667   Median : 4.000   Median : 4.088   Median : 0.7500  
##  Mean   : 0.7251   Mean   : 4.464   Mean   : 4.215   Mean   : 0.5234  
##  3rd Qu.: 1.8286   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 1.6753  
##  Max.   : 6.0000   Max.   :15.000   Max.   :11.000   Max.   : 6.0000  
##      xw_01            xw_02            xw_03           xs_01        
##  Min.   : 14.00   Min.   :  0.00   Min.   : 14.0   Min.   :-0.2177  
##  1st Qu.: 43.85   1st Qu.: 10.50   1st Qu.: 65.5   1st Qu.: 0.1073  
##  Median : 56.85   Median : 25.00   Median : 94.0   Median : 0.1715  
##  Mean   : 57.59   Mean   : 30.93   Mean   : 82.1   Mean   : 0.1660  
##  3rd Qu.: 68.69   3rd Qu.: 45.00   3rd Qu.:101.0   3rd Qu.: 0.2250  
##  Max.   :102.00   Max.   :102.00   Max.   :110.0   Max.   : 0.5247  
##      xs_02             xs_03             xs_04             xs_05        
```

```
##  Min.   :-0.43347   Min.   :-0.2177   Min.   :0.0000   Min.   :0.00000  
##  1st Qu.:-0.14100   1st Qu.: 0.1759   1st Qu.:0.2455   1st Qu.:0.08714  
##  Median : 0.00000   Median : 0.3402   Median :0.2859   Median :0.15195  
##  Mean   :-0.01581   Mean   : 0.3603   Mean   :0.3103   Mean   :0.18920  
##  3rd Qu.: 0.12984   3rd Qu.: 0.4968   3rd Qu.:0.3501   3rd Qu.:0.26305  
##  Max.   : 0.52468   Max.   : 1.1833   Max.   :0.8988   Max.   :0.89883  
##      xs_06       
##  Min.   :0.0000  
##  1st Qu.:0.3334  
##  Median :0.4727  
##  Mean   :0.4815  
##  3rd Qu.:0.5708  
##  Max.   :1.2703
```

Outcome = Non_Event

```
df %>%
  filter(outcome == "non_event") %>%
  select(all_of(continuous_vars)) %>%
  tibble::rowid_to_column() %>%
  pivot_longer(!c("rowid")) %>%
  ggplot(mapping = aes(x = value)) +
  geom_histogram(bins = 50) +
  facet_wrap(~name, scales = "free") +
  theme_bw() +
  theme(axis.text.y = element_blank())
```




```
df %>%
  filter(outcome == "non_event") %>%
  select(all_of(continuous_vars)) %>%
  summary()
```

```
##      xb_01            xb_02            xb_03            xn_01
##  Min.   :-2.000   Min.   :-2.000   Min.   :-6.000   Min.   :-3.500
##  1st Qu.: 2.667   1st Qu.: 4.000   1st Qu.:-1.000   1st Qu.: 1.000
##  Median : 3.429   Median : 6.000   Median : 1.000   Median : 1.857
##  Mean   : 3.618   Mean   : 5.998   Mean   : 1.456   Mean   : 1.913
##  3rd Qu.: 4.500   3rd Qu.: 8.000   3rd Qu.: 3.000   3rd Qu.: 2.650
##  Max.   :14.000   Max.   :15.000   Max.   :14.000   Max.   :10.000
##      xn_02            xn_03                xa_01            xa_02
##  Min.   :-3.000   Min.   :-7.000000   Min.   :-2.000   Min.   :-2.00
##  1st Qu.: 2.000   1st Qu.:-2.000000   1st Qu.: 6.500   1st Qu.: 8.25
##  Median : 4.000   Median : 0.000000   Median : 8.156   Median :13.50
##  Mean   : 4.033   Mean   :-0.003636   Mean   : 8.515   Mean   :13.72
##  3rd Qu.: 6.000   3rd Qu.: 2.000000   3rd Qu.:10.036   3rd Qu.:18.00
##  Max.   :13.000   Max.   :10.000000   Max.   :35.000   Max.   :38.00
```

```
##      xa_03            xb_04              xb_05              xb_06       
##  Min.   :-9.000   Min.   :-1.0000   Min.   :-3.0000   Min.   :-1.000  
##  1st Qu.: 0.000   1st Qu.: 0.9383   1st Qu.:-0.2500   1st Qu.: 1.250  
##  Median : 4.000   Median : 1.1974   Median : 0.5000   Median : 2.000  
##  Mean   : 4.271   Mean   : 1.2273   Mean   : 0.4839   Mean   : 2.189  
##  3rd Qu.: 7.000   3rd Qu.: 1.5000   3rd Qu.: 1.0000   3rd Qu.: 3.000  
##  Max.   :35.000   Max.   : 5.0000   Max.   : 5.0000   Max.   : 9.000  
##      xb_07            xb_08              xn_04              xn_05          
##  Min.   :-1.000   Min.   :-2.0000   Min.   :-1.0000   Min.   :-3.000000  
##  1st Qu.: 1.802   1st Qu.:-0.1667   1st Qu.: 0.4333   1st Qu.:-1.000000  
##  Median : 2.000   Median : 0.2824   Median : 0.6752   Median : 0.000000  
##  Mean   : 2.167   Mean   : 0.2993   Mean   : 0.7458   Mean   : 0.004892  
##  3rd Qu.: 2.500   3rd Qu.: 1.0000   3rd Qu.: 1.0000   3rd Qu.: 0.750000  
##  Max.   : 7.000   Max.   : 5.0000   Max.   : 5.0000   Max.   : 5.000000  
##      xn_06            xn_07            xn_08              xa_04       
##  Min.   :-1.000   Min.   :-1.000   Min.   :-3.0000   Min.   :-2.000  
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:-0.6667   1st Qu.: 2.388  
##  Median : 1.367   Median : 1.500   Median :-0.1854   Median : 3.000  
##  Mean   : 1.634   Mean   : 1.532   Mean   :-0.1063   Mean   : 3.076  
##  3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 0.3333   3rd Qu.: 3.577  
##  Max.   : 7.000   Max.   : 5.000   Max.   : 5.0000   Max.   :12.000  
##      xa_05                xa_06            xa_07            xa_08        
##  Min.   :-8.00000   Min.   :-2.000   Min.   :-2.000   Min.   :-5.0000  
##  1st Qu.: 0.04167   1st Qu.: 3.000   1st Qu.: 4.000   1st Qu.: 0.6254  
##  Median : 1.50000   Median : 4.500   Median : 4.718   Median : 1.2386  
##  Mean   : 1.53068   Mean   : 5.308   Mean   : 4.810   Mean   : 1.3823  
##  3rd Qu.: 3.00000   3rd Qu.: 7.000   3rd Qu.: 5.500   3rd Qu.: 2.0000  
##  Max.   :12.00000   Max.   :23.000   Max.   :13.000   Max.   :12.0000  
##      xw_01            xw_02            xw_03            xs_01        
##  Min.   :  9.00   Min.   :  0.00   Min.   :  9.00   Min.   :-0.3612  
##  1st Qu.: 44.50   1st Qu.:  9.00   1st Qu.: 57.00   1st Qu.: 0.1589  
##  Median : 57.79   Median : 24.00   Median : 91.50   Median : 0.2254  
##  Mean   : 56.88   Mean   : 32.08   Mean   : 78.37   Mean   : 0.2261  
##  3rd Qu.: 67.46   3rd Qu.: 51.00   3rd Qu.:101.00   3rd Qu.: 0.2945  
##  Max.   :108.00   Max.   :108.00   Max.   :113.00   Max.   : 0.7548  
##      xs_02                xs_03              xs_04              xs_05        
##  Min.   :-0.89585   Min.   :-0.3612   Min.   :0.02511   Min.   :0.00000  
##  1st Qu.:-0.14308   1st Qu.: 0.2578   1st Qu.:0.24250   1st Qu.:0.07771  
##  Median : 0.04786   Median : 0.3976   Median :0.29147   Median :0.16581  
##  Mean   : 0.03107   Mean   : 0.4388   Mean   :0.29900   Mean   :0.18850  
##  3rd Qu.: 0.22046   3rd Qu.: 0.6097   3rd Qu.:0.34113   3rd Qu.:0.26384  
##  Max.   : 0.69105   Max.   : 1.7907   Max.   :0.68960   Max.   :0.68960  
##      xs_06        
##  Min.   :0.02511  
##  1st Qu.:0.29971  
##  Median :0.42122  
##  Mean   :0.46320  
```

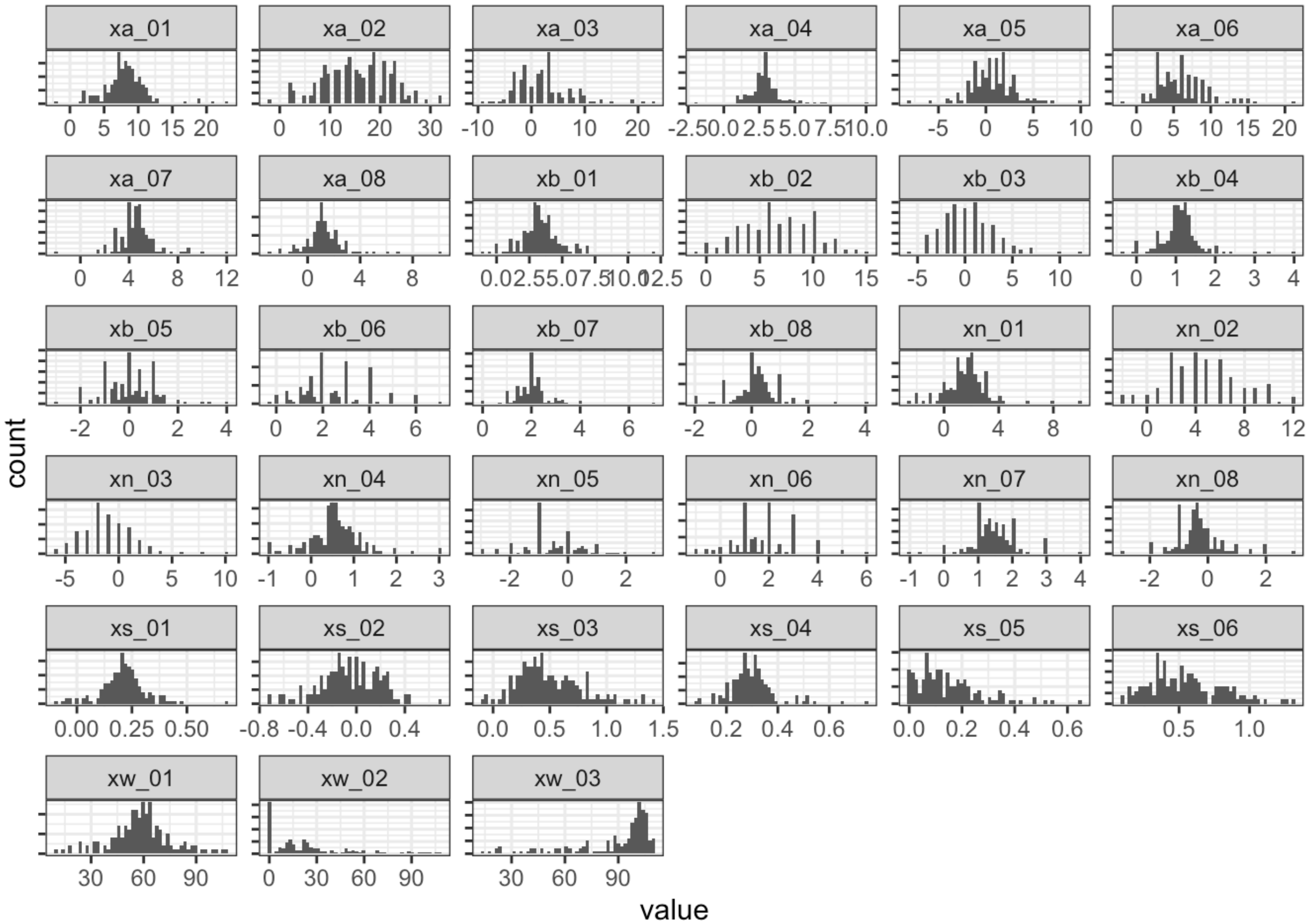
```
##  3rd Qu.:0.60198
##  Max.   :1.30883
```

For most of the sentiment derived features, each product sold to a customer has a greater sentiment value when the outcome is classified as an event.

Examining the impact on region on the continuous variables

Region = XX

```
df %>%
  filter(region == "XX") %>%
  select(all_of(continuous_vars)) %>%
  tibble::rowid_to_column() %>%
  pivot_longer(!c("rowid")) %>%
  ggplot(mapping = aes(x = value)) +
  geom_histogram(bins = 50) +
  facet_wrap(~name, scales = "free") +
  theme_bw() +
  theme(axis.text.y = element_blank())
```




```
df %>%
  filter(region == "XX") %>%
  select(all_of(continuous_vars)) %>%
  summary()
```

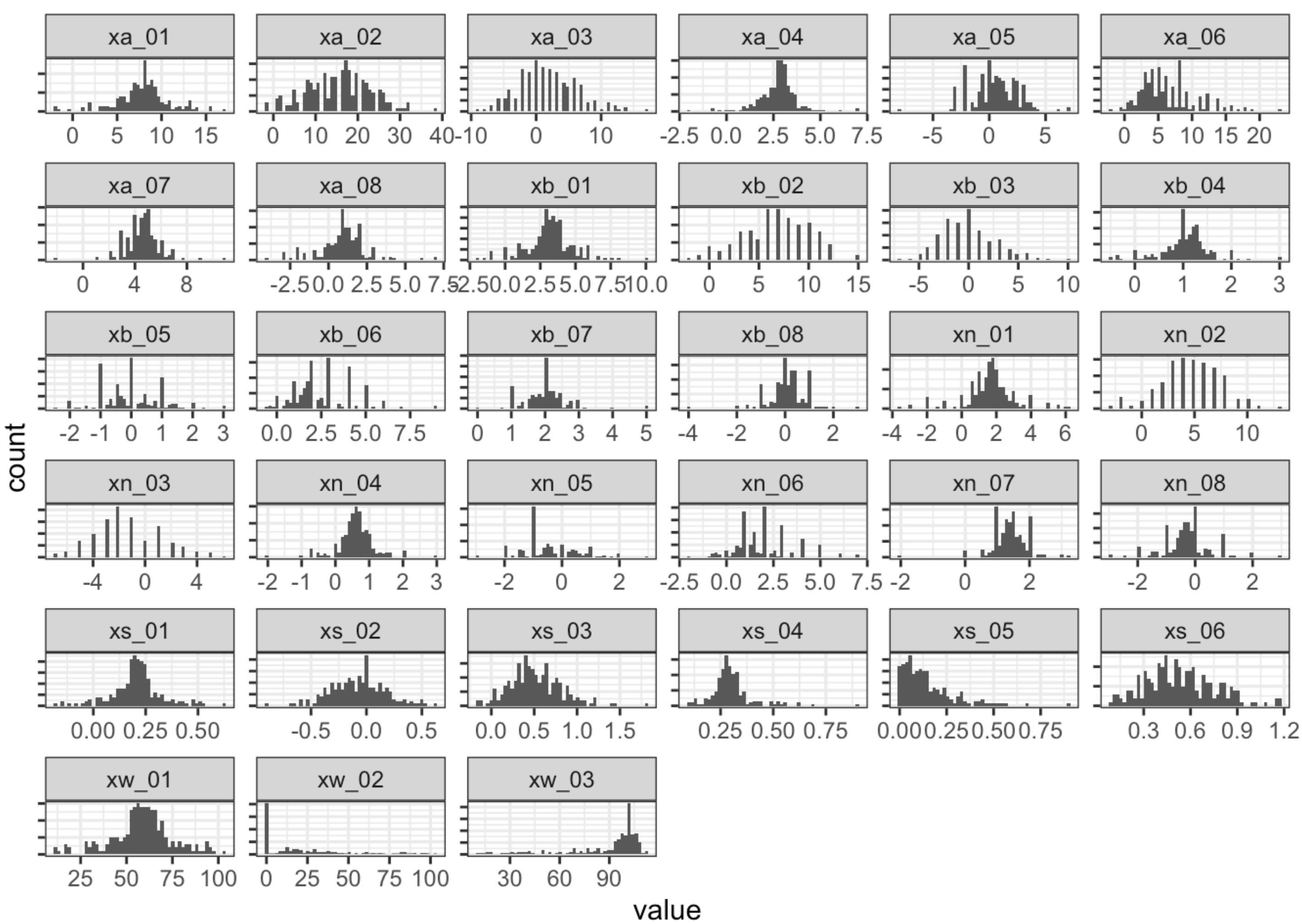
```
##      xb_01            xb_02            xb_03             xn_01       
##  Min.   :-1.000   Min.   :-1.000   Min.   :-6.0000   Min.   :-2.500  
##  1st Qu.: 2.625   1st Qu.: 4.000   1st Qu.:-2.0000   1st Qu.: 1.000  
##  Median : 3.250   Median : 7.000   Median : 0.0000   Median : 1.667  
##  Mean   : 3.354   Mean   : 6.708   Mean   : 0.3851   Mean   : 1.588  
##  3rd Qu.: 4.000   3rd Qu.: 9.000   3rd Qu.: 2.0000   3rd Qu.: 2.286  
##  Max.   :12.000   Max.   :15.000   Max.   :12.0000   Max.   :10.000  
##      xn_02            xn_03            xa_01            xa_02      
##  Min.   :-2.000   Min.   :-6.000   Min.   :-2.000   Min.   :-2.00  
##  1st Qu.: 2.000   1st Qu.:-3.000   1st Qu.: 6.677   1st Qu.:11.00  
##  Median : 4.000   Median :-1.000   Median : 8.000   Median :15.00  
##  Mean   : 4.621   Mean   :-1.087   Mean   : 8.093   Mean   :15.26  
##  3rd Qu.: 6.000   3rd Qu.: 0.000   3rd Qu.: 9.667   3rd Qu.:21.00  
##  Max.   :12.000   Max.   :10.000   Max.   :23.000   Max.   :32.00  
```

```
##      xa_03            xb_04             xb_05              xb_06        
##  Min.   :-9.000   Min.   :-0.3333   Min.   :-3.00000   Min.   :-0.3333  
##  1st Qu.:-1.000   1st Qu.: 0.9056   1st Qu.:-0.66667   1st Qu.: 1.5000  
##  Median : 2.000   Median : 1.1039   Median : 0.00000   Median : 2.0000  
##  Mean   : 2.199   Mean   : 1.1081   Mean   : 0.04583   Mean   : 2.4198  
##  3rd Qu.: 4.000   3rd Qu.: 1.2884   3rd Qu.: 0.75000   3rd Qu.: 3.0000  
##  Max.   :23.000   Max.   : 4.0000   Max.   : 4.00000   Max.   : 7.0000  
##      xb_07           xb_08             xn_04             xn_05        
##  Min.   :0.000   Min.   :-2.0000   Min.   :-1.0000   Min.   :-3.0000  
##  1st Qu.:1.714   1st Qu.:-0.1000   1st Qu.: 0.3312   1st Qu.:-1.0000  
##  Median :2.000   Median : 0.1606   Median : 0.5655   Median :-0.5000  
##  Mean   :2.030   Mean   : 0.1593   Mean   : 0.5772   Mean   :-0.5039  
##  3rd Qu.:2.250   3rd Qu.: 0.5000   3rd Qu.: 0.8600   3rd Qu.: 0.0000  
##  Max.   :7.000   Max.   : 4.0000   Max.   : 3.0000   Max.   : 3.0000  
##      xn_06            xn_07             xn_08             xa_04       
##  Min.   :-1.000   Min.   :-1.000   Min.   :-3.0000   Min.   :-2.000  
##  1st Qu.: 1.000   1st Qu.: 1.067   1st Qu.:-0.7500   1st Qu.: 2.448  
##  Median : 1.667   Median : 1.400   Median :-0.3750   Median : 2.866  
##  Mean   : 1.736   Mean   : 1.440   Mean   :-0.3311   Mean   : 2.958  
##  3rd Qu.: 2.500   3rd Qu.: 1.765   3rd Qu.: 0.0000   3rd Qu.: 3.302  
##  Max.   : 6.000   Max.   : 4.000   Max.   : 3.0000   Max.   :10.000  
##      xa_05              xa_06            xa_07            xa_08        
##  Min.   :-8.0000   Min.   :-2.000   Min.   :-2.000   Min.   :-3.0000  
##  1st Qu.:-0.6667   1st Qu.: 3.500   1st Qu.: 4.000   1st Qu.: 0.6429  
##  Median : 0.6667   Median : 5.500   Median : 4.590   Median : 1.0667  
##  Mean   : 0.7295   Mean   : 5.934   Mean   : 4.666   Mean   : 1.2323  
##  3rd Qu.: 2.0000   3rd Qu.: 8.000   3rd Qu.: 5.167   3rd Qu.: 1.7857  
##  Max.   :10.0000   Max.   :21.000   Max.   :12.000   Max.   :10.0000  
##      xw_01            xw_02            xw_03            xs_01         
##  Min.   : 10.50   Min.   :  0.00   Min.   : 14.00   Min.   :-0.09905  
##  1st Qu.: 50.37   1st Qu.:  0.00   1st Qu.: 82.00   1st Qu.: 0.16001  
##  Median : 58.55   Median : 16.00   Median : 98.00   Median : 0.21005  
##  Mean   : 58.31   Mean   : 24.11   Mean   : 87.91   Mean   : 0.20989  
##  3rd Qu.: 65.92   3rd Qu.: 31.00   3rd Qu.:103.00   3rd Qu.: 0.25971  
##  Max.   :108.00   Max.   :108.00   Max.   :110.00   Max.   : 0.67685  
##      xs_02              xs_03              xs_04              xs_05        
##  Min.   :-0.73322   Min.   :-0.09905   Min.   :0.08563   Min.   :0.00000  
##  1st Qu.:-0.19427   1st Qu.: 0.30751   1st Qu.:0.25467   1st Qu.:0.06645  
##  Median :-0.05646   Median : 0.43543   Median :0.28865   Median :0.11691  
##  Mean   :-0.05512   Mean   : 0.50137   Mean   :0.29742   Mean   :0.14660  
##  3rd Qu.: 0.14512   3rd Qu.: 0.66385   3rd Qu.:0.32932   3rd Qu.:0.20407  
##  Max.   : 0.67685   Max.   : 1.40500   Max.   :0.74984   Max.   :0.64732  
##      xs_06        
##  Min.   :0.09744  
##  1st Qu.:0.35745  
##  Median :0.51720  
##  Mean   :0.54494  
```

```
## 3rd Qu.:0.69807
## Max.   :1.30883
```

Region = YY

```
df %>%
  filter(region == "YY") %>%
  select(all_of(continuous_vars)) %>%
  tibble::rowid_to_column() %>%
  pivot_longer(!c("rowid")) %>%
  ggplot(mapping = aes(x = value)) +
  geom_histogram(bins = 50) +
  facet_wrap(~name, scales = "free") +
  theme_bw() +
  theme(axis.text.y = element_blank())
```



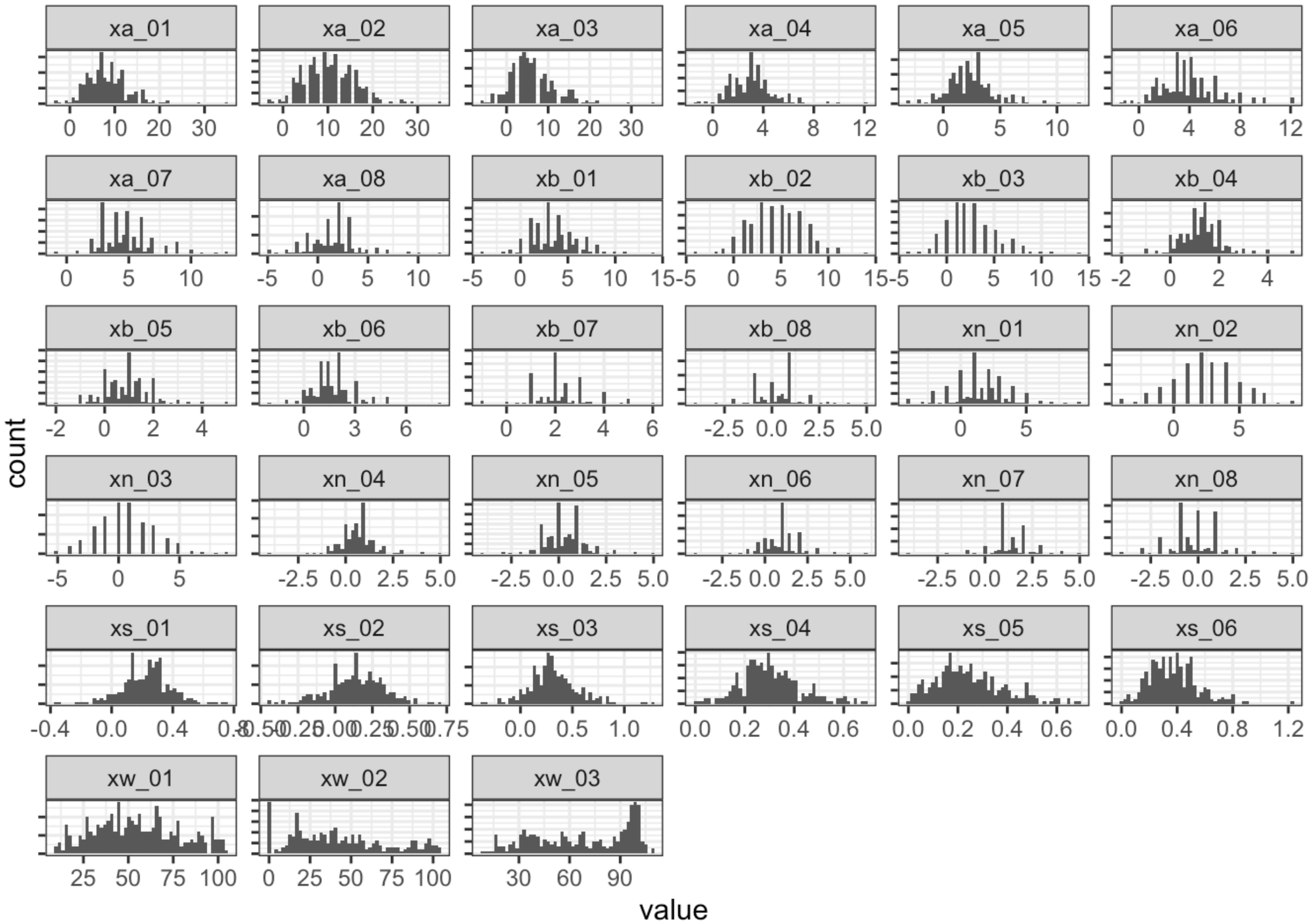
```
df %>%
  filter(region == "YY") %>%
  select(all_of(continuous_vars)) %>%
  summary()
```

```
##      xb_01            xb_02            xb_03              xn_01
##  Min.   :-2.000   Min.   :-2.000   Min.   :-7.00000   Min.   :-3.500
##  1st Qu.: 2.557   1st Qu.: 4.000   1st Qu.:-2.00000   1st Qu.: 1.000
##  Median : 3.231   Median : 7.000   Median : 0.00000   Median : 1.667
##  Mean   : 3.194   Mean   : 6.676   Mean   :-0.01351   Mean   : 1.605
##  3rd Qu.: 3.980   3rd Qu.: 9.000   3rd Qu.: 1.75000   3rd Qu.: 2.231
##  Max.   :10.000   Max.   :15.000   Max.   :10.00000   Max.   : 6.250
##      xn_02            xn_03            xa_01            xa_02
##  Min.   :-3.000   Min.   :-7.000   Min.   :-2.000   Min.   :-2.00
##  1st Qu.: 3.000   1st Qu.:-3.000   1st Qu.: 6.600   1st Qu.:10.00
##  Median : 5.000   Median :-2.000   Median : 8.000   Median :16.00
##  Mean   : 4.662   Mean   :-1.324   Mean   : 7.813   Mean   :15.51
##  3rd Qu.: 7.000   3rd Qu.: 1.000   3rd Qu.: 9.200   3rd Qu.:21.00
##  Max.   :13.000   Max.   : 6.000   Max.   :17.000   Max.   :38.00
##      xa_03            xb_04             xb_05              xb_06
##  Min.   :-9.000   Min.   :-0.5000   Min.   :-2.50000   Min.   :-0.500
##  1st Qu.:-1.000   1st Qu.: 0.8622   1st Qu.:-0.66667   1st Qu.: 1.500
##  Median : 1.000   Median : 1.0528   Median : 0.00000   Median : 2.000
##  Mean   : 1.802   Mean   : 1.0448   Mean   :-0.01128   Mean   : 2.503
##  3rd Qu.: 5.000   3rd Qu.: 1.2881   3rd Qu.: 0.65000   3rd Qu.: 3.500
##  Max.   :17.000   Max.   : 3.0000   Max.   : 3.00000   Max.   : 9.000
##      xb_07           xb_08              xn_04              xn_05
##  Min.   :0.000   Min.   :-4.00000   Min.   :-2.0000   Min.   :-3.0000
##  1st Qu.:1.719   1st Qu.:-0.20625   1st Qu.: 0.3864   1st Qu.:-1.0000
##  Median :2.000   Median : 0.09091   Median : 0.6122   Median :-0.6667
##  Mean   :2.005   Mean   : 0.10428   Mean   : 0.6022   Mean   :-0.4687
##  3rd Qu.:2.250   3rd Qu.: 0.50000   3rd Qu.: 0.8504   3rd Qu.: 0.2375
##  Max.   :5.000   Max.   : 3.00000   Max.   : 3.0000   Max.   : 3.0000
##      xn_06            xn_07            xn_08             xa_04
##  Min.   :-2.000   Min.   :-2.000   Min.   :-3.0000   Min.   :-2.000
##  1st Qu.: 1.000   1st Qu.: 1.118   1st Qu.:-0.6667   1st Qu.: 2.337
##  Median : 1.750   Median : 1.434   Median :-0.2967   Median : 2.845
##  Mean   : 1.958   Mean   : 1.414   Mean   :-0.2939   Mean   : 2.740
##  3rd Qu.: 3.000   3rd Qu.: 1.667   3rd Qu.: 0.0000   3rd Qu.: 3.193
##  Max.   : 7.000   Max.   : 3.250   Max.   : 3.0000   Max.   : 7.000
##      xa_05             xa_06            xa_07            xa_08
##  Min.   :-8.0000   Min.   :-2.000   Min.   :-2.000   Min.   :-4.0000
##  1st Qu.:-0.5000   1st Qu.: 3.375   1st Qu.: 4.000   1st Qu.: 0.3438
##  Median : 0.5000   Median : 5.292   Median : 4.648   Median : 1.0000
##  Mean   : 0.5413   Mean   : 6.210   Mean   : 4.571   Mean   : 0.9398
##  3rd Qu.: 2.0000   3rd Qu.: 8.000   3rd Qu.: 5.162   3rd Qu.: 1.7362
```

```
## Max.   : 7.0000   Max.   :23.000   Max.   :11.000   Max.   : 7.0000
##     xw_01            xw_02            xw_03            xs_01
## Min.   : 11.00   Min.   :  0.00   Min.   : 11.00   Min.   :-0.1789
## 1st Qu.: 51.54   1st Qu.:  0.00   1st Qu.: 83.00   1st Qu.: 0.1562
## Median : 59.01   Median : 14.50   Median : 98.00   Median : 0.2070
## Mean   : 58.58   Mean   : 23.04   Mean   : 88.18   Mean   : 0.2054
## 3rd Qu.: 66.76   3rd Qu.: 35.00   3rd Qu.:103.00   3rd Qu.: 0.2521
## Max.   :103.00   Max.   :103.00   Max.   :113.00   Max.   : 0.6283
##     xs_02              xs_03              xs_04              xs_05
## Min.   :-0.89585   Min.   :-0.1789   Min.   :0.09682   Min.   :0.00000
## 1st Qu.:-0.24315   1st Qu.: 0.3128   1st Qu.:0.25516   1st Qu.:0.04787
## Median :-0.07242   Median : 0.4865   Median :0.28727   Median :0.10632
## Mean   :-0.07333   Mean   : 0.5125   Mean   :0.30026   Mean   :0.14386
## 3rd Qu.: 0.07709   3rd Qu.: 0.6922   3rd Qu.:0.32621   3rd Qu.:0.19564
## Max.   : 0.62832   Max.   : 1.7907   Max.   :0.89883   Max.   :0.89883
##     xs_06
## Min.   :0.09682
## 1st Qu.:0.38598
## Median :0.51745
## Mean   :0.53339
## 3rd Qu.:0.68155
## Max.   :1.17974
```

Region == ZZ

```
df %>%
  filter(region == "ZZ") %>%
  select(all_of(continuous_vars)) %>%
  tibble::rowid_to_column() %>%
  pivot_longer(!c("rowid")) %>%
  ggplot(mapping = aes(x = value)) +
  geom_histogram(bins = 50) +
  facet_wrap(~name, scales = "free") +
  theme_bw() +
  theme(axis.text.y = element_blank())
```




```
df %>%
  filter(region == "ZZ") %>%
  select(all_of(continuous_vars)) %>%
  summary()
```

```
##      xb_01            xb_02            xb_03            xn_01       
##  Min.   :-4.000   Min.   :-4.000   Min.   :-4.000   Min.   :-4.000  
##  1st Qu.: 2.000   1st Qu.: 3.000   1st Qu.: 1.000   1st Qu.: 0.000  
##  Median : 3.208   Median : 4.500   Median : 2.000   Median : 1.333  
##  Mean   : 3.528   Mean   : 4.524   Mean   : 2.602   Mean   : 1.506  
##  3rd Qu.: 5.000   3rd Qu.: 7.000   3rd Qu.: 4.000   3rd Qu.: 2.788  
##  Max.   :14.000   Max.   :14.000   Max.   :14.000   Max.   : 9.000  
##      xn_02            xn_03             xa_01            xa_02      
##  Min.   :-4.000   Min.   :-5.0000   Min.   :-3.000   Min.   :-3.00  
##  1st Qu.: 1.000   1st Qu.:-1.0000   1st Qu.: 5.083   1st Qu.: 6.00  
##  Median : 2.000   Median : 1.0000   Median : 7.583   Median :10.00  
##  Mean   : 2.388   Mean   : 0.6701   Mean   : 8.258   Mean   :10.43  
##  3rd Qu.: 4.000   3rd Qu.: 2.0000   3rd Qu.:11.000   3rd Qu.:14.00  
##  Max.   : 9.000   Max.   : 9.0000   Max.   :35.000   Max.   :35.00  
```

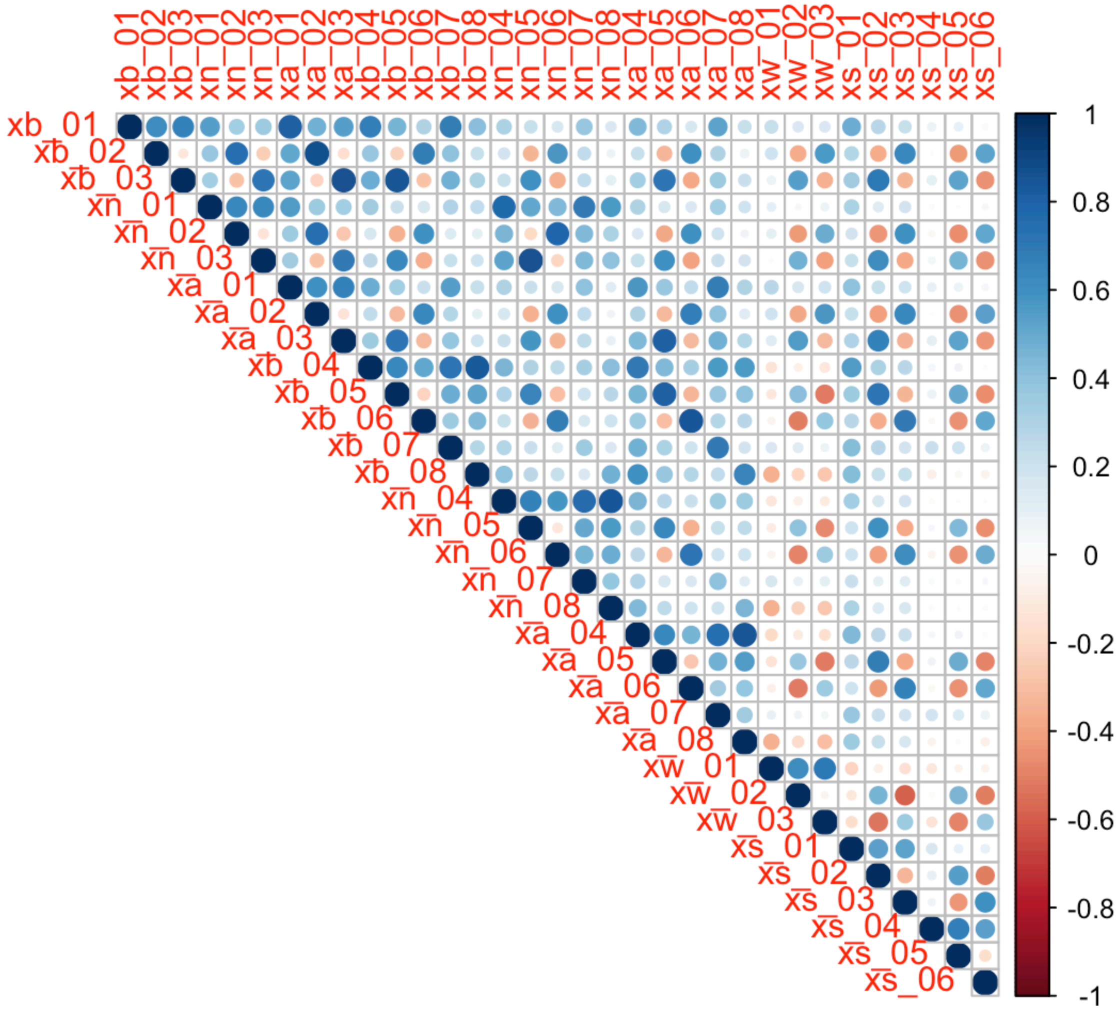
```
##      xa_03            xb_04            xb_05             xb_06
##  Min.   :-6.000   Min.   :-2.000   Min.   :-2.0000   Min.   :-2.000
##  1st Qu.: 3.000   1st Qu.: 0.750   1st Qu.: 0.3333   1st Qu.: 1.000
##  Median : 5.000   Median : 1.292   Median : 1.0000   Median : 1.500
##  Mean   : 6.269   Mean   : 1.259   Mean   : 0.9227   Mean   : 1.636
##  3rd Qu.: 9.000   3rd Qu.: 1.665   3rd Qu.: 1.5000   3rd Qu.: 2.000
##  Max.   :35.000   Max.   : 5.000   Max.   : 5.0000   Max.   : 8.000
##      xb_07            xb_08             xn_04              xn_05
##  Min.   :-1.000   Min.   :-4.0000   Min.   :-4.00000   Min.   :-4.0000
##  1st Qu.: 1.667   1st Qu.:-0.5000   1st Qu.: 0.02708   1st Qu.:-0.3333
##  Median : 2.000   Median : 0.5000   Median : 0.55556   Median : 0.2000
##  Mean   : 2.204   Mean   : 0.3231   Mean   : 0.61960   Mean   : 0.2652
##  3rd Qu.: 3.000   3rd Qu.: 1.0000   3rd Qu.: 1.00000   3rd Qu.: 1.0000
##  Max.   : 6.000   Max.   : 5.0000   Max.   : 5.00000   Max.   : 5.0000
##      xn_06             xn_07            xn_08             xa_04
##  Min.   :-4.0000   Min.   :-4.000   Min.   :-4.0000   Min.   :-1.500
##  1st Qu.: 0.4464   1st Qu.: 1.000   1st Qu.:-1.0000   1st Qu.: 2.000
##  Median : 1.0000   Median : 1.127   Median : 0.0000   Median : 3.000
##  Mean   : 0.9761   Mean   : 1.382   Mean   :-0.2118   Mean   : 3.092
##  3rd Qu.: 1.5000   3rd Qu.: 2.000   3rd Qu.: 0.7292   3rd Qu.: 3.759
##  Max.   : 6.0000   Max.   : 5.000   Max.   : 5.0000   Max.   :12.000
##      xa_05            xa_06            xa_07            xa_08
##  Min.   :-3.000   Min.   :-1.500   Min.   :-1.000   Min.   :-5.000
##  1st Qu.: 1.175   1st Qu.: 2.500   1st Qu.: 3.375   1st Qu.: 0.000
##  Median : 2.250   Median : 3.667   Median : 4.536   Median : 1.667
##  Mean   : 2.369   Mean   : 3.919   Mean   : 4.814   Mean   : 1.428
##  3rd Qu.: 3.237   3rd Qu.: 5.000   3rd Qu.: 6.000   3rd Qu.: 2.500
##  Max.   :12.000   Max.   :12.000   Max.   :13.000   Max.   :12.000
##      xw_01            xw_02            xw_03            xs_01
##  Min.   :  9.00   Min.   :  0.00   Min.   :  9.00   Min.   :-0.3612
##  1st Qu.: 37.50   1st Qu.: 20.00   1st Qu.: 41.25   1st Qu.: 0.1344
##  Median : 53.22   Median : 38.50   Median : 69.00   Median : 0.2393
##  Mean   : 55.13   Mean   : 42.78   Mean   : 67.35   Mean   : 0.2247
##  3rd Qu.: 71.78   3rd Qu.: 62.00   3rd Qu.: 95.00   3rd Qu.: 0.3126
##  Max.   :104.00   Max.   :104.00   Max.   :110.00   Max.   : 0.7548
##      xs_02              xs_03             xs_04             xs_05
##  Min.   :-0.45588   Min.   :-0.3612   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 0.01633   1st Qu.: 0.1679   1st Qu.:0.2268   1st Qu.:0.1516
##  Median : 0.14129   Median : 0.2949   Median :0.2927   Median :0.2240
##  Mean   : 0.13685   Mean   : 0.3150   Mean   :0.3038   Mean   :0.2455
##  3rd Qu.: 0.26076   3rd Qu.: 0.4330   3rd Qu.:0.3692   3rd Qu.:0.3179
##  Max.   : 0.69105   Max.   : 1.2814   Max.   :0.6896   Max.   :0.6896
##      xs_06
##  Min.   :0.0000
##  1st Qu.:0.2448
##  Median :0.3538
##  Mean   :0.3734
```

```
##   3rd Qu.:0.4802
##   Max.   :1.2274
```

In general, it appears that for most of the derived sentiment features, Region ZZ has the highest sentiment values.

Examining the correlation between the continuous inputs

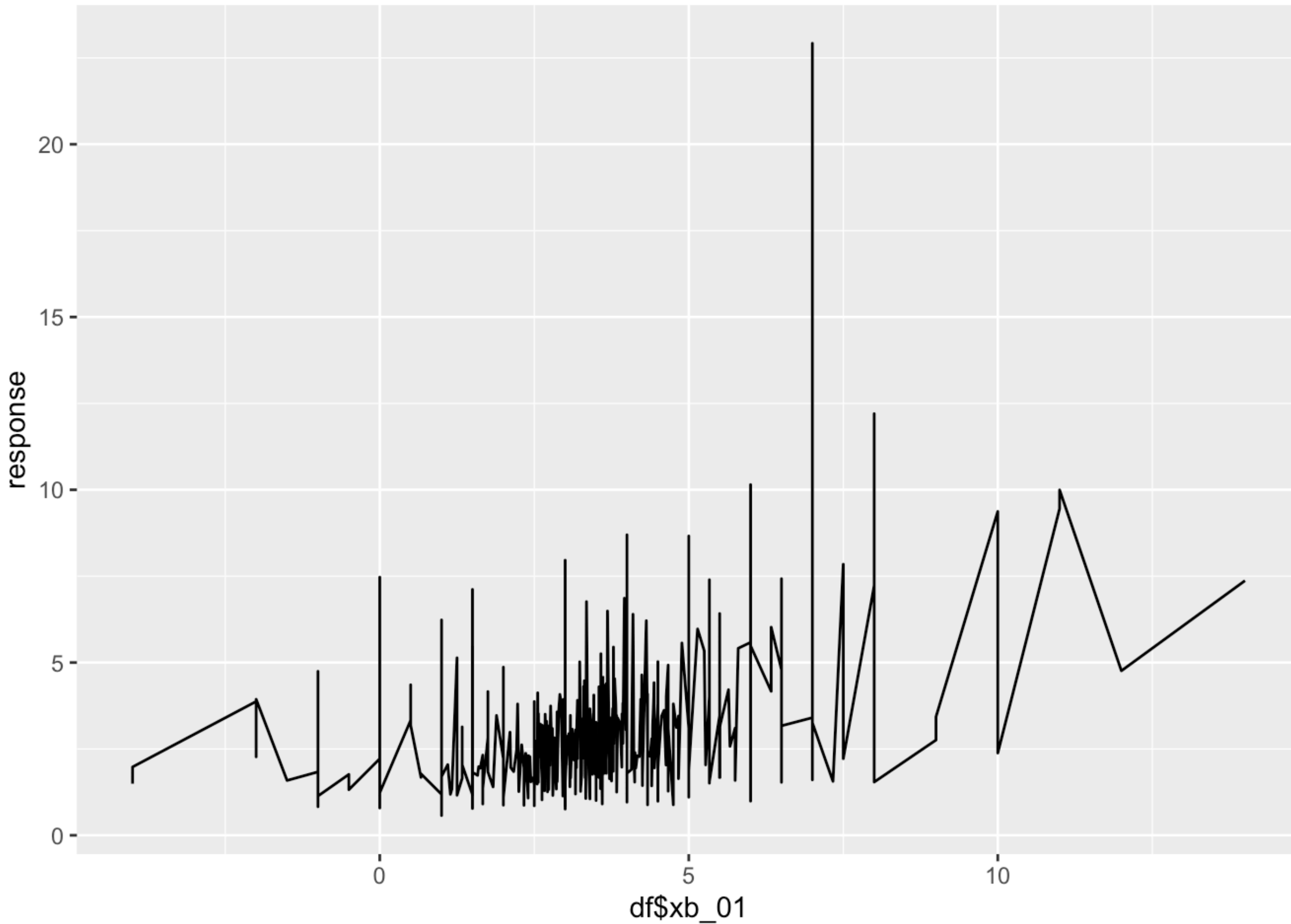
```
df %>%
  select(all_of(continuous_vars)) %>%
  cor() %>%
  corrplot::corrplot( type = 'upper' )
```



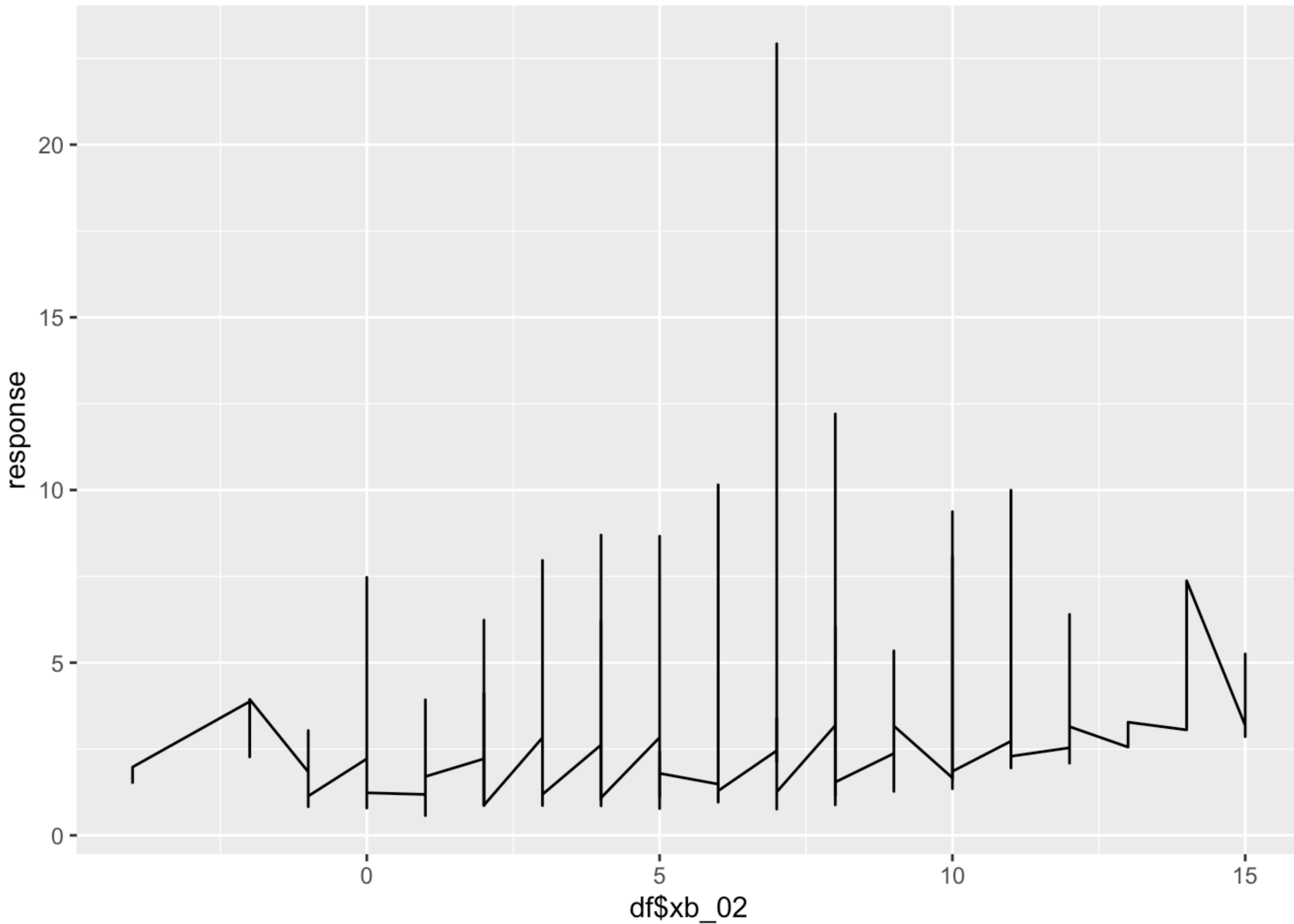
There appears to be many continuous variables that are highly correlated, both positively and negatively.

Examining the relationship between the continuous output (response) with respect to the continous inputs

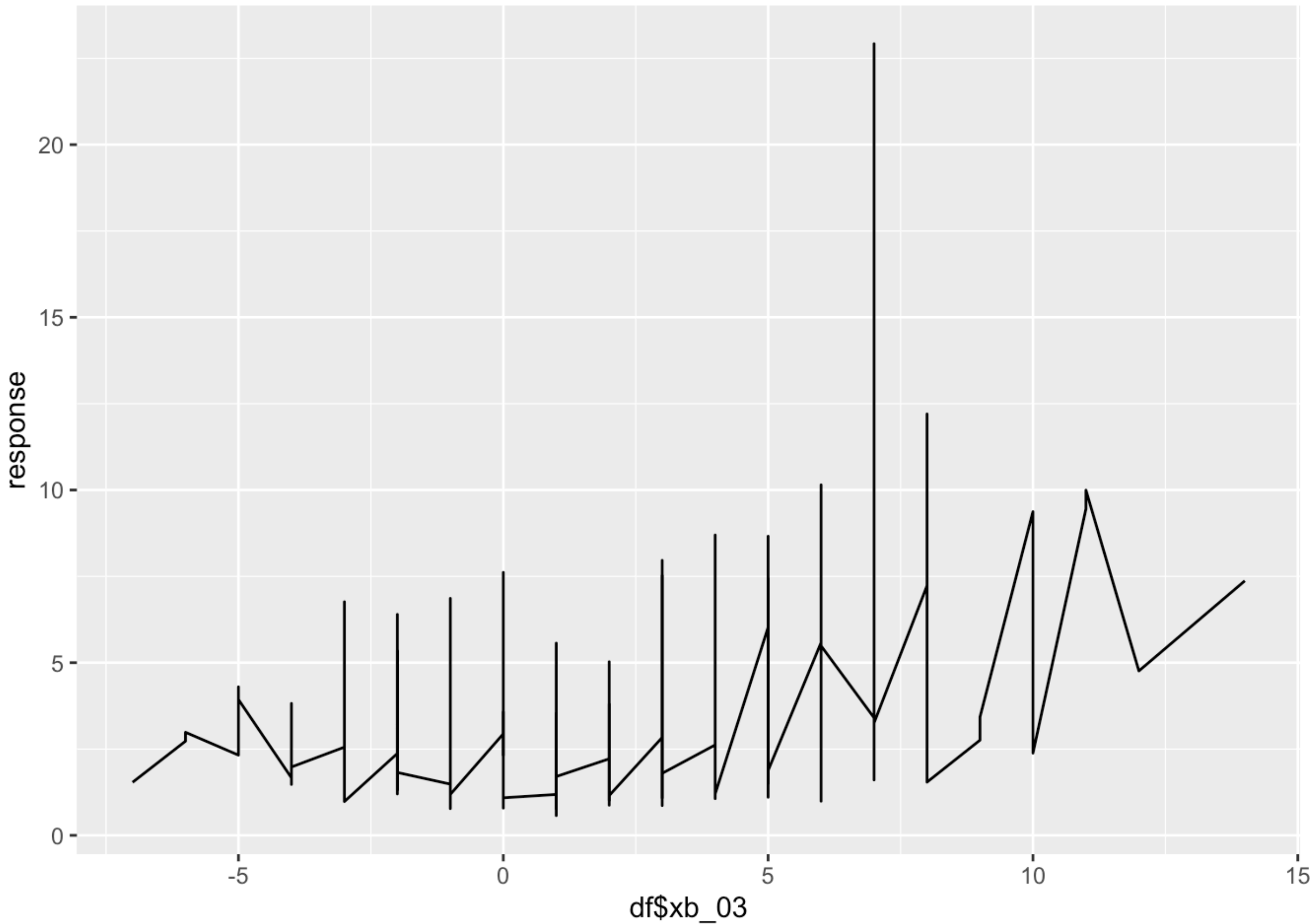
```
df %>%
  ggplot(mapping = aes(df$xb_01, response)) + geom_line()
```



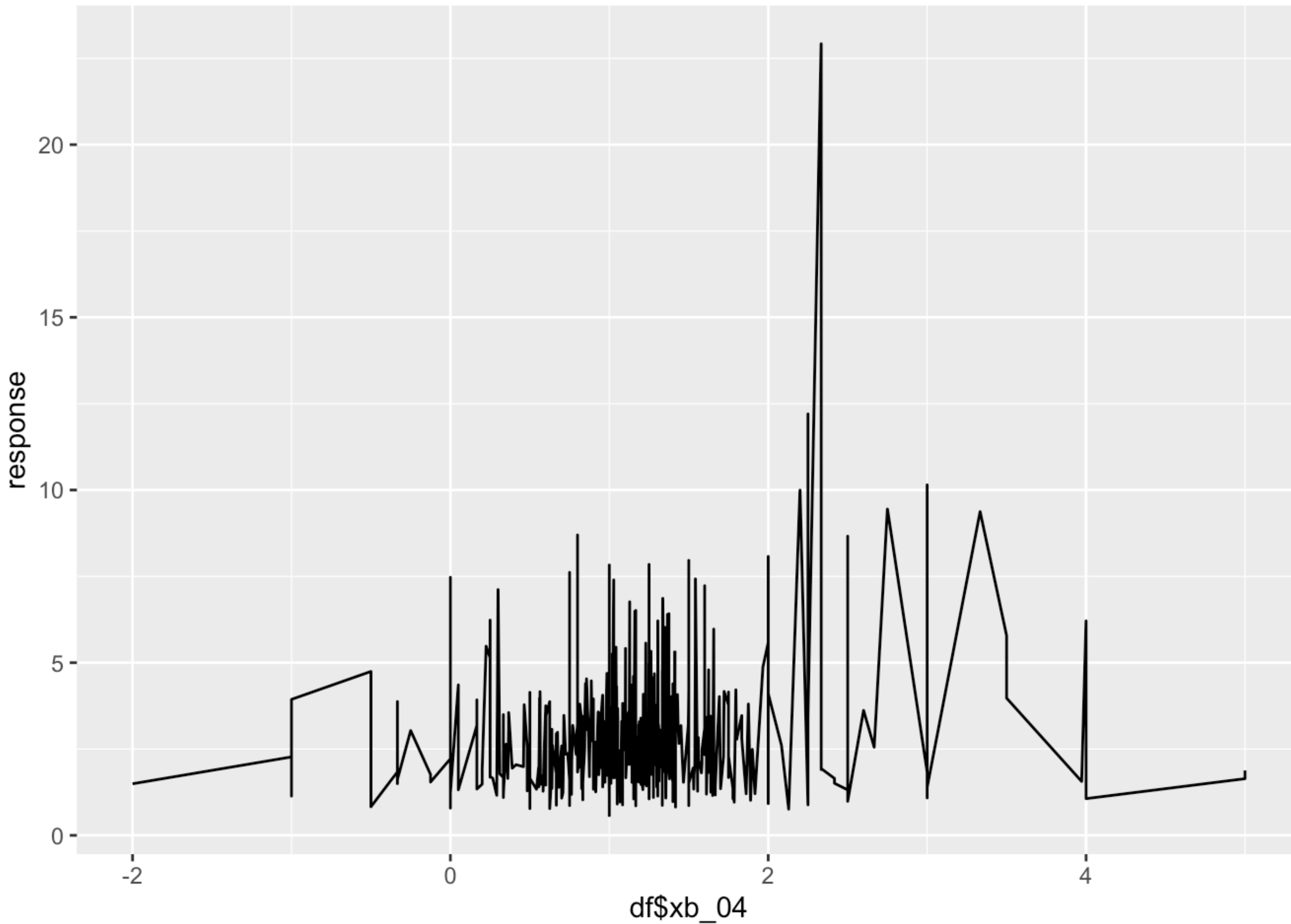

```
df %>%
  ggplot(mapping = aes(df$xb_02, response)) + geom_line()
```



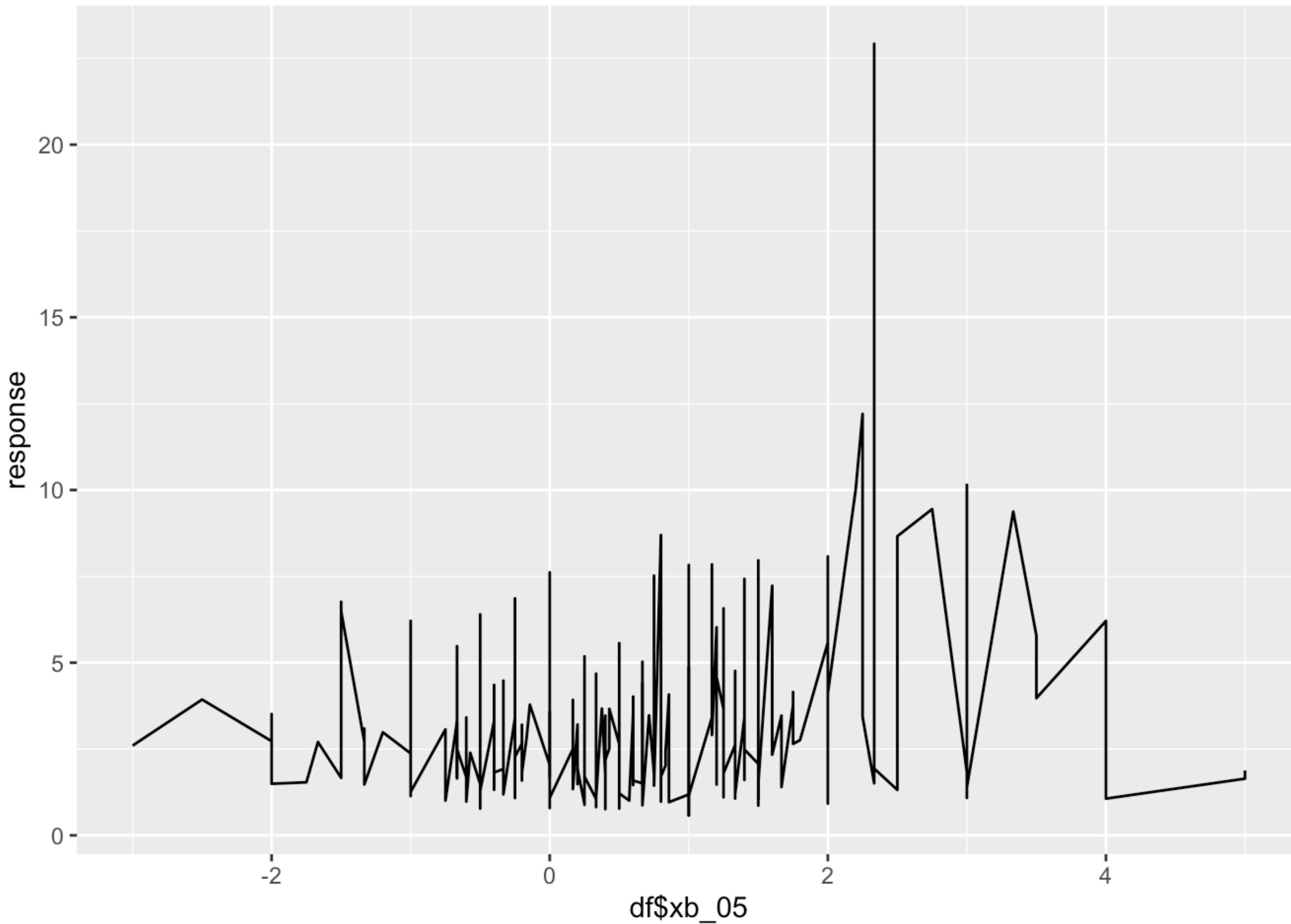

```
df %>%
  ggplot(mapping = aes(df$xb_03, response)) + geom_line()
```



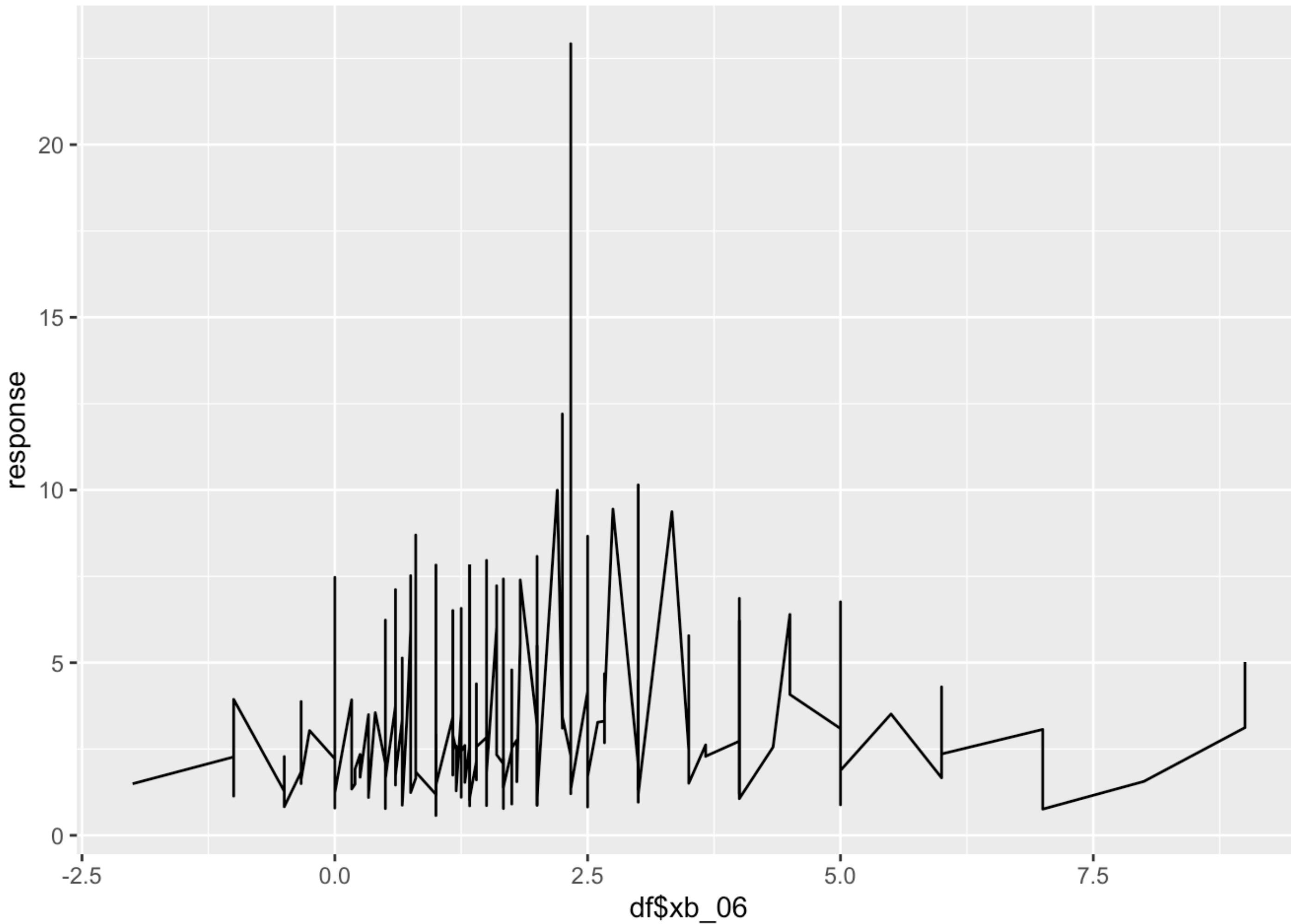

```
df %>%
  ggplot(mapping = aes(df$xb_04, response)) + geom_line()
```



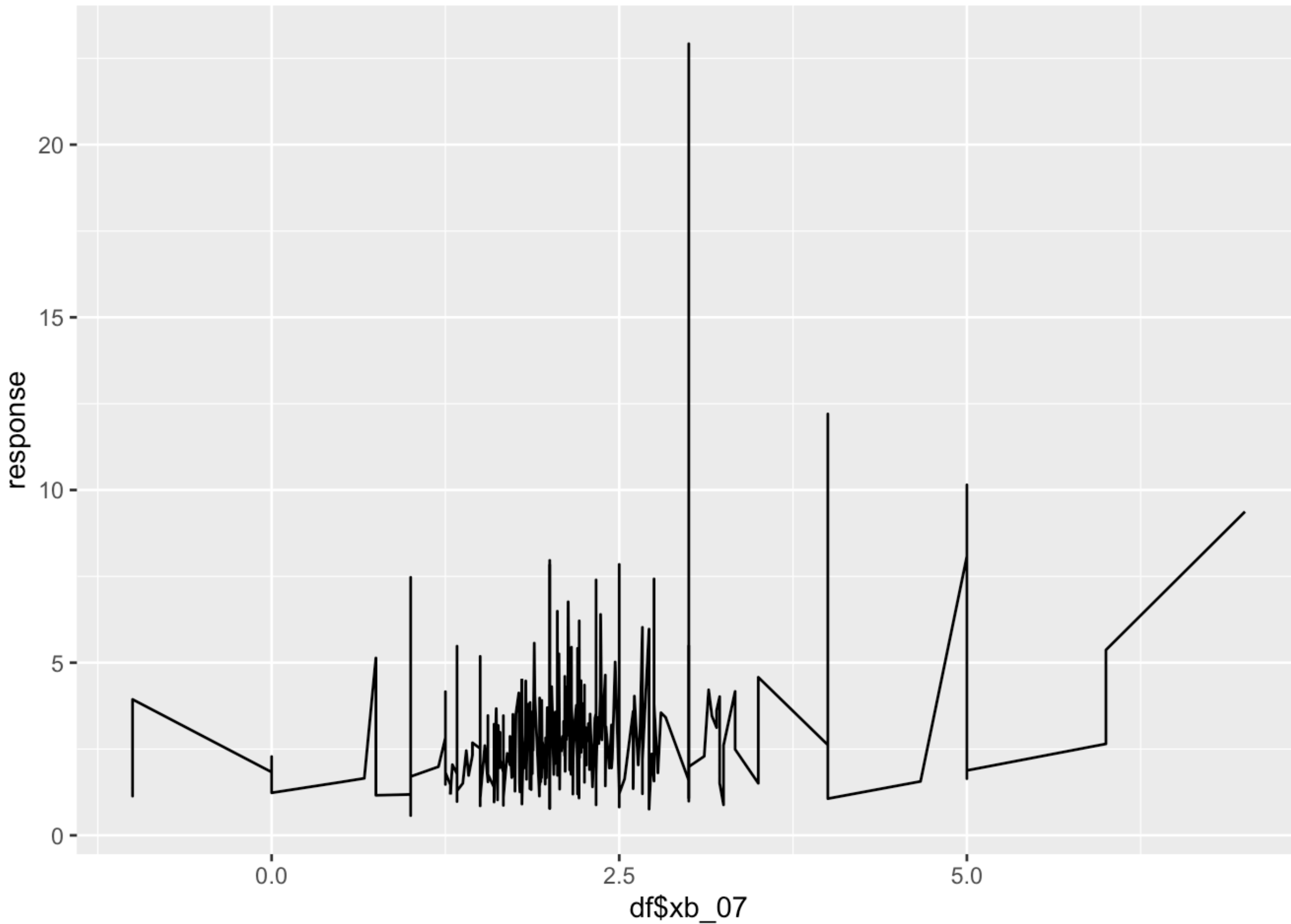

```
df %>%
  ggplot(mapping = aes(df$xb_05, response)) + geom_line()
```



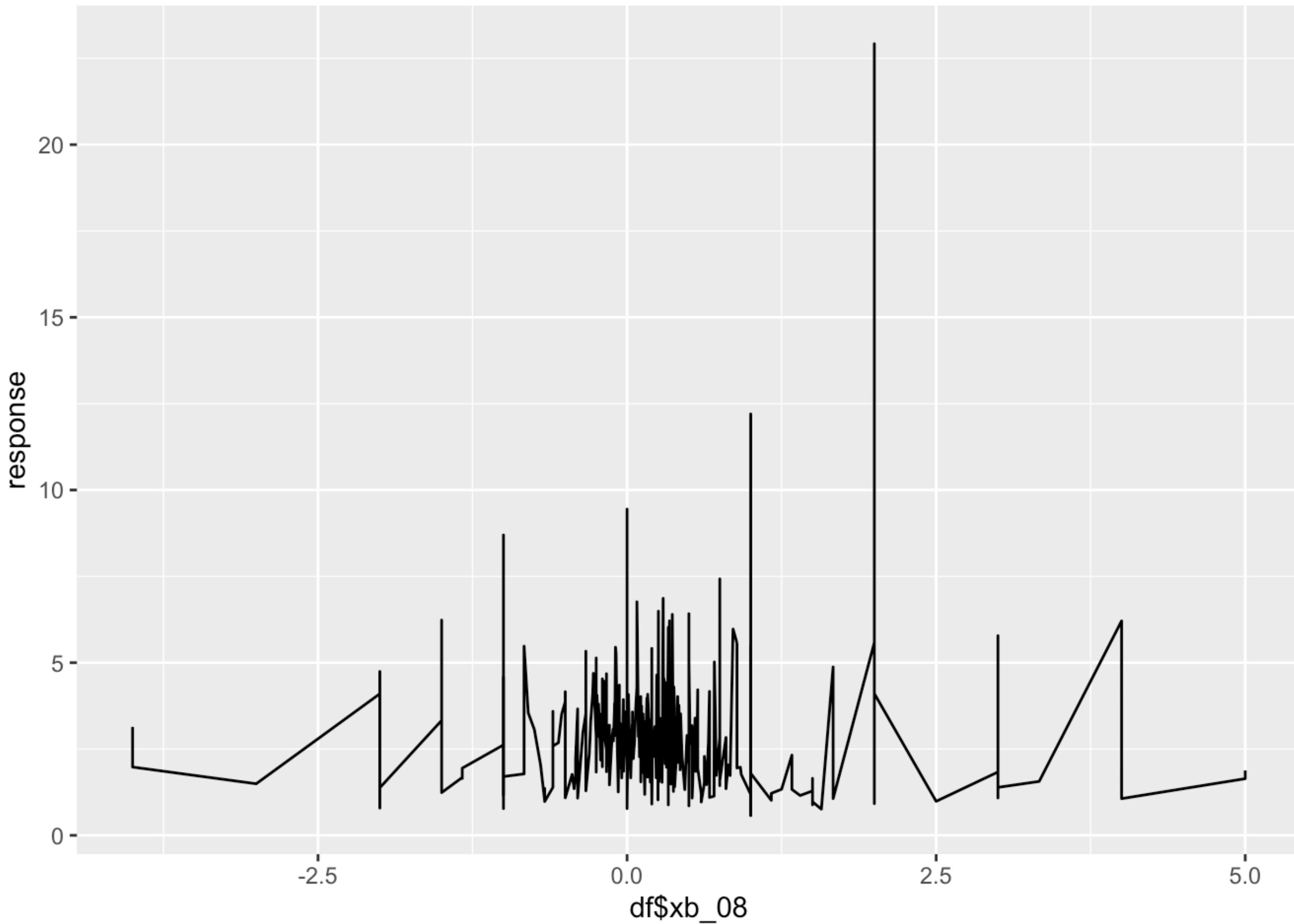

```
df %>%
  ggplot(mapping = aes(df$xb_06, response)) + geom_line()
```



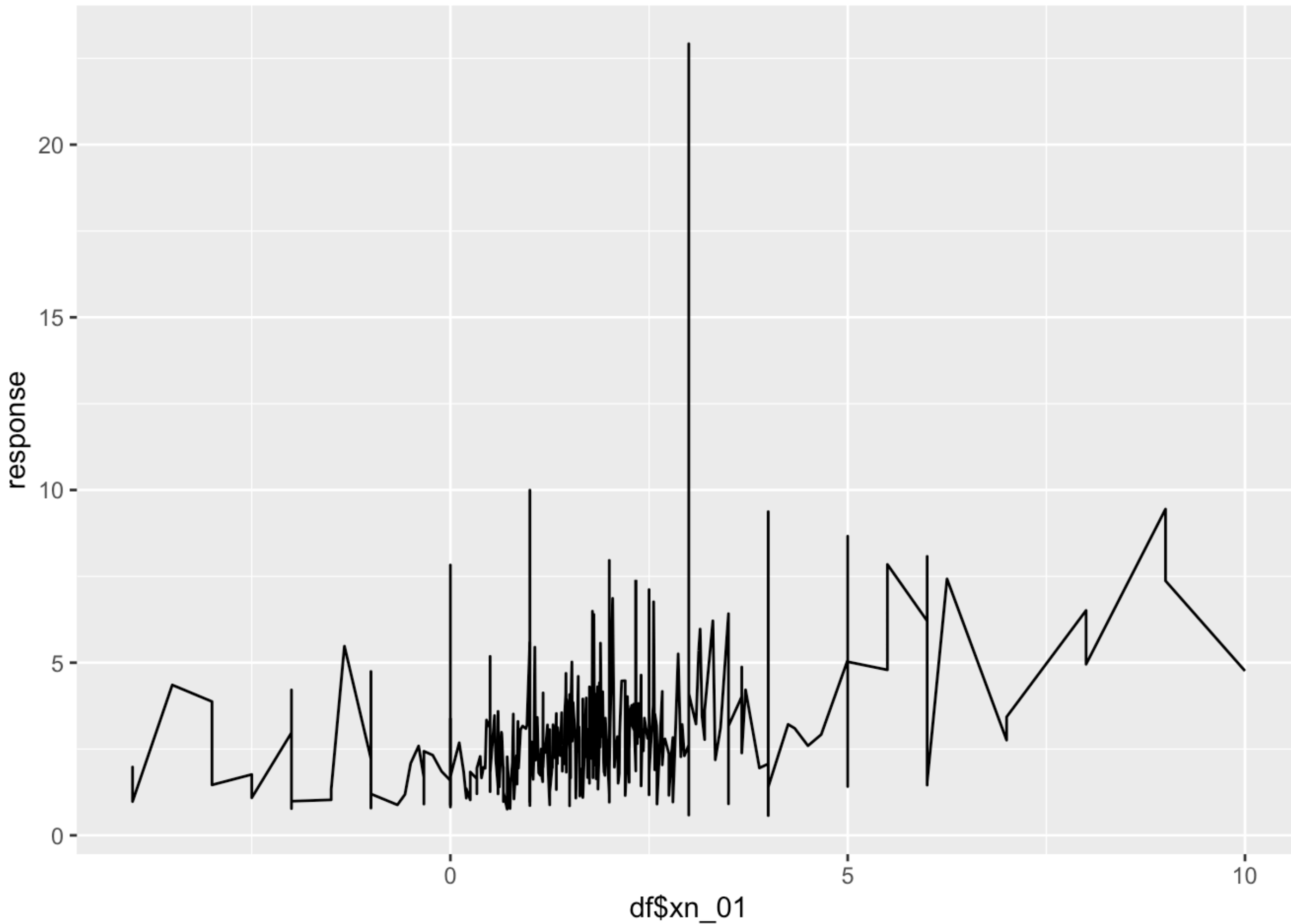

```
df %>%
  ggplot(mapping = aes(df$xb_07, response)) + geom_line()
```



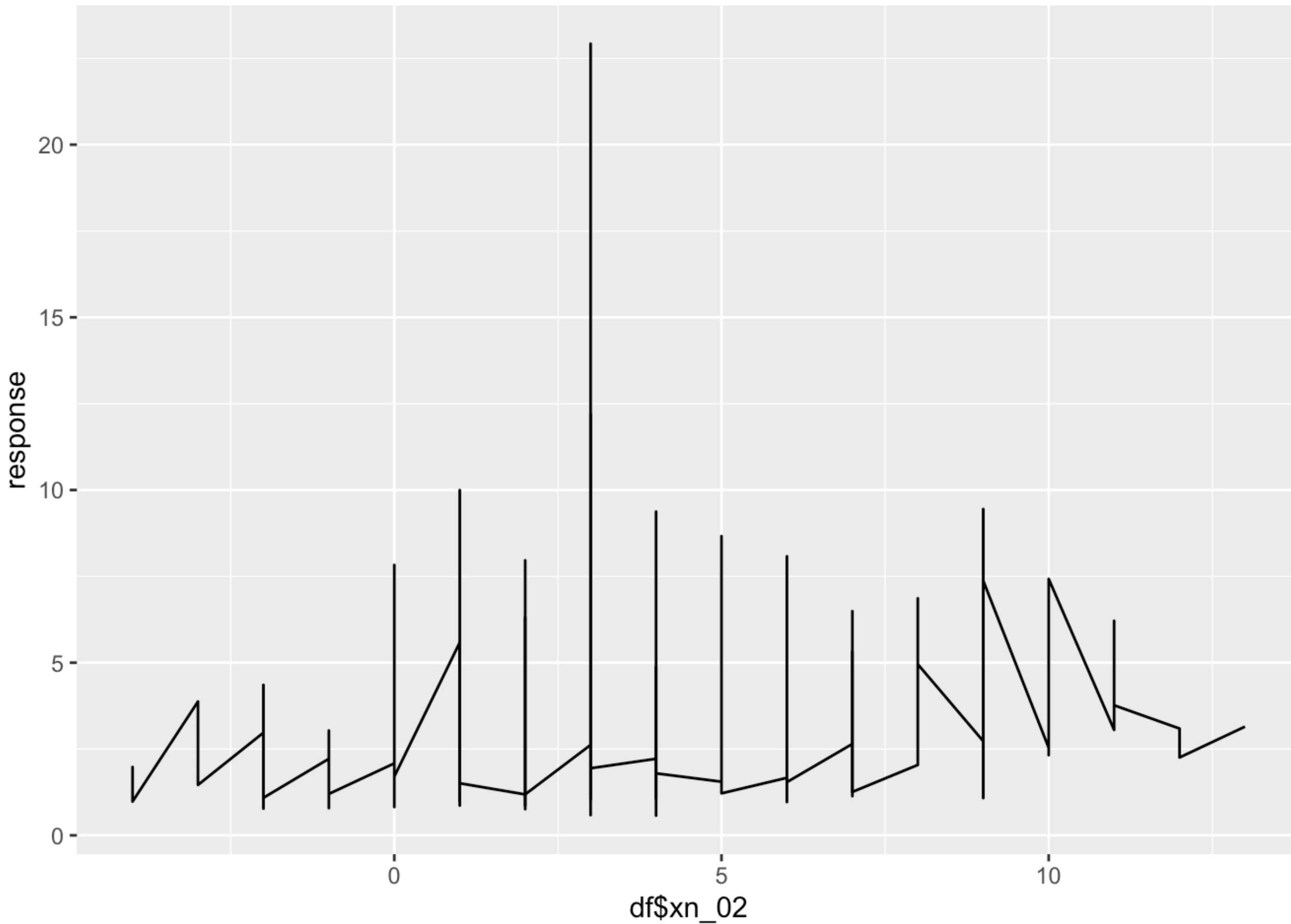

```
df %>%
  ggplot(mapping = aes(df$xb_08, response)) + geom_line()
```



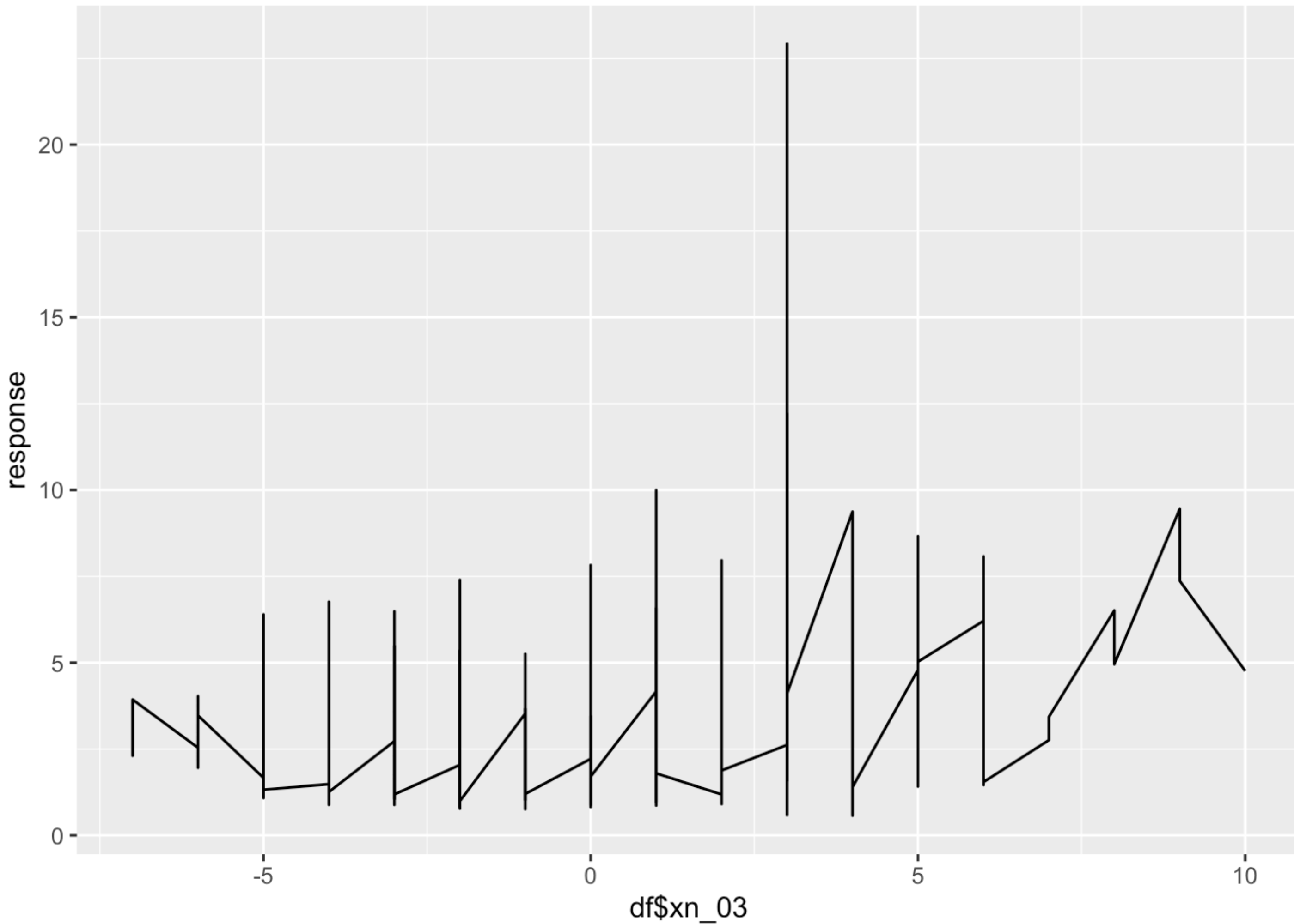

```
df %>%
  ggplot(mapping = aes(df$xn_01, response)) + geom_line()
```



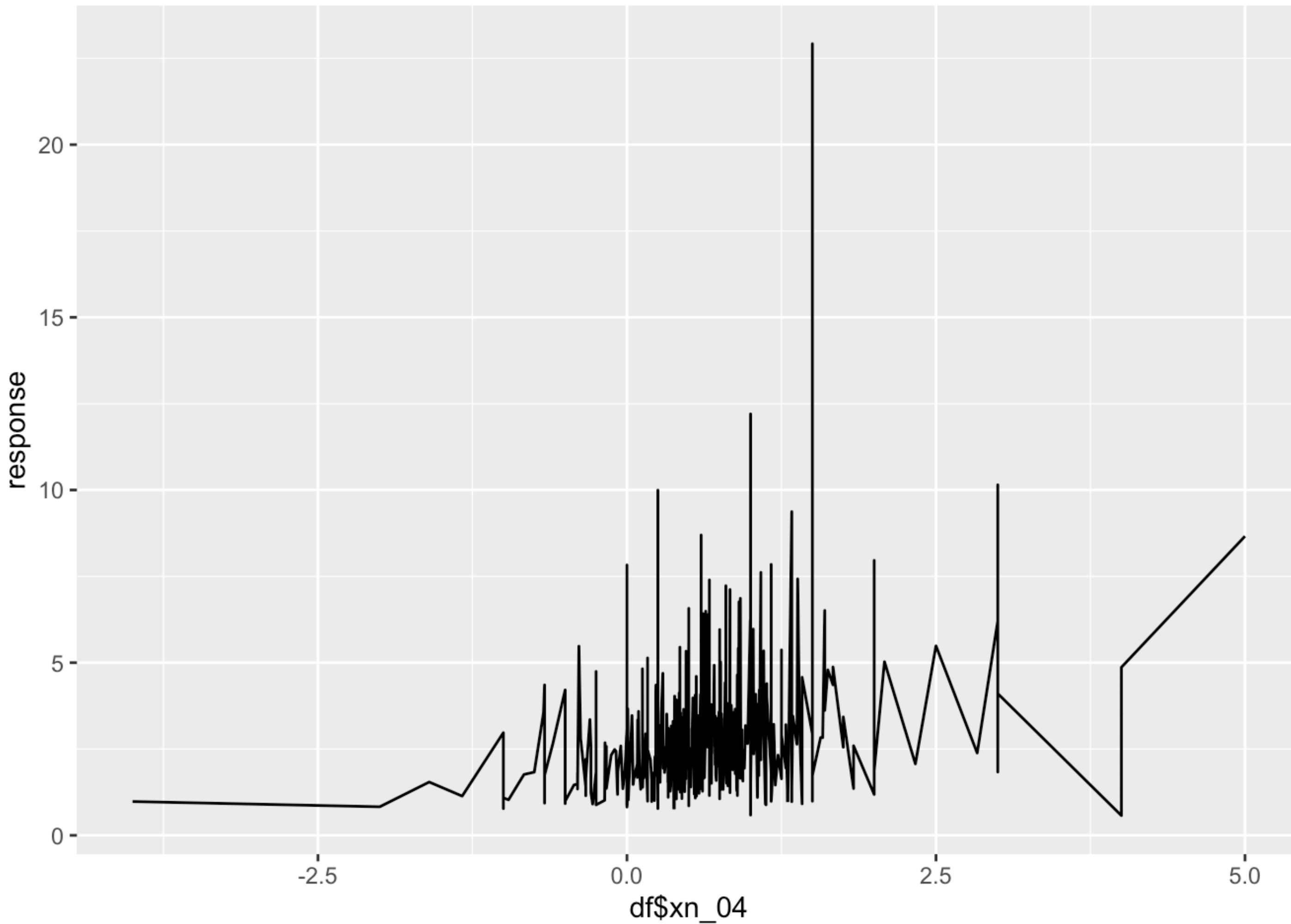

```
df %>%
  ggplot(mapping = aes(df$xn_02, response)) + geom_line()
```



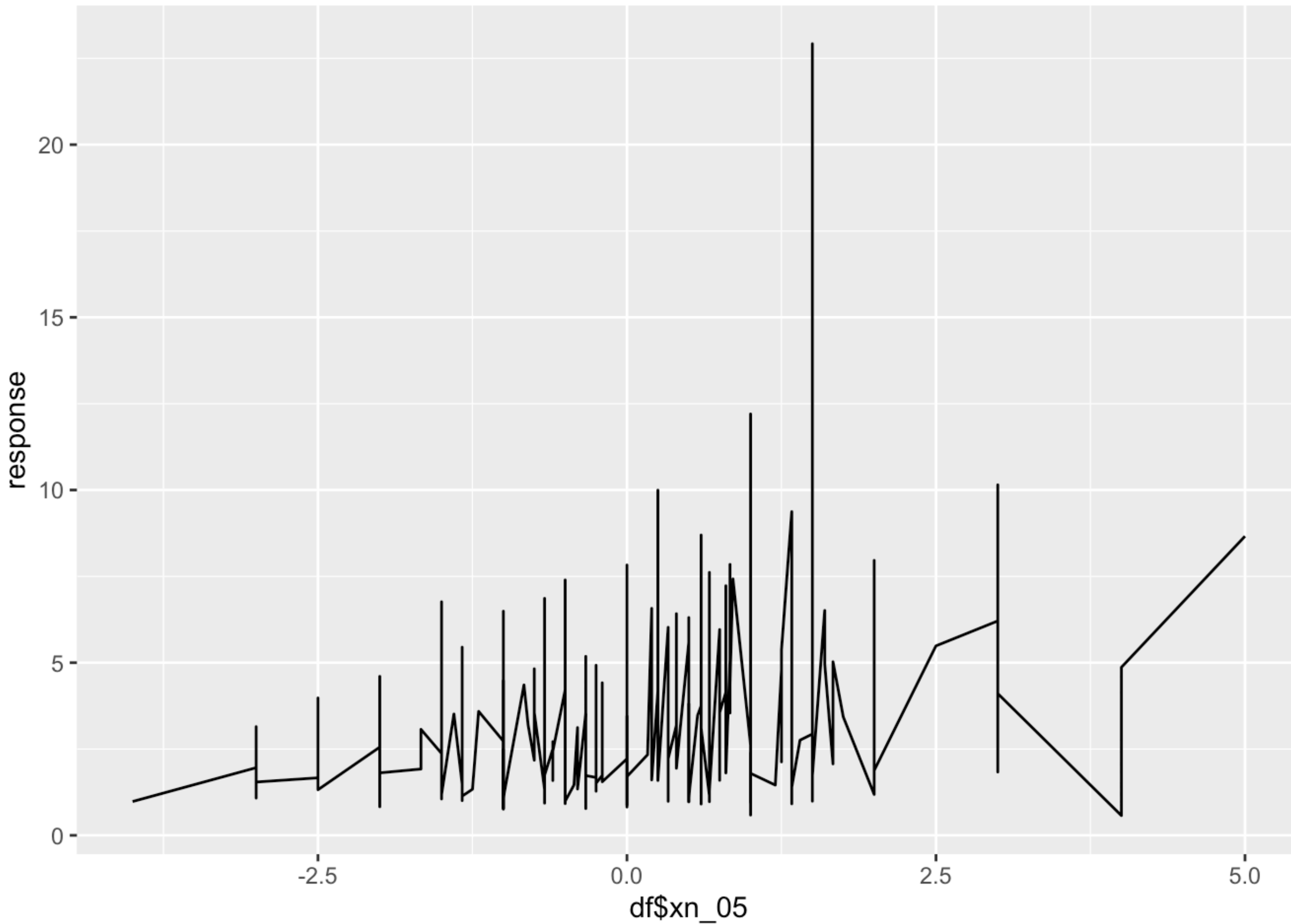

```
df %>%
  ggplot(mapping = aes(df$xn_03, response)) + geom_line()
```



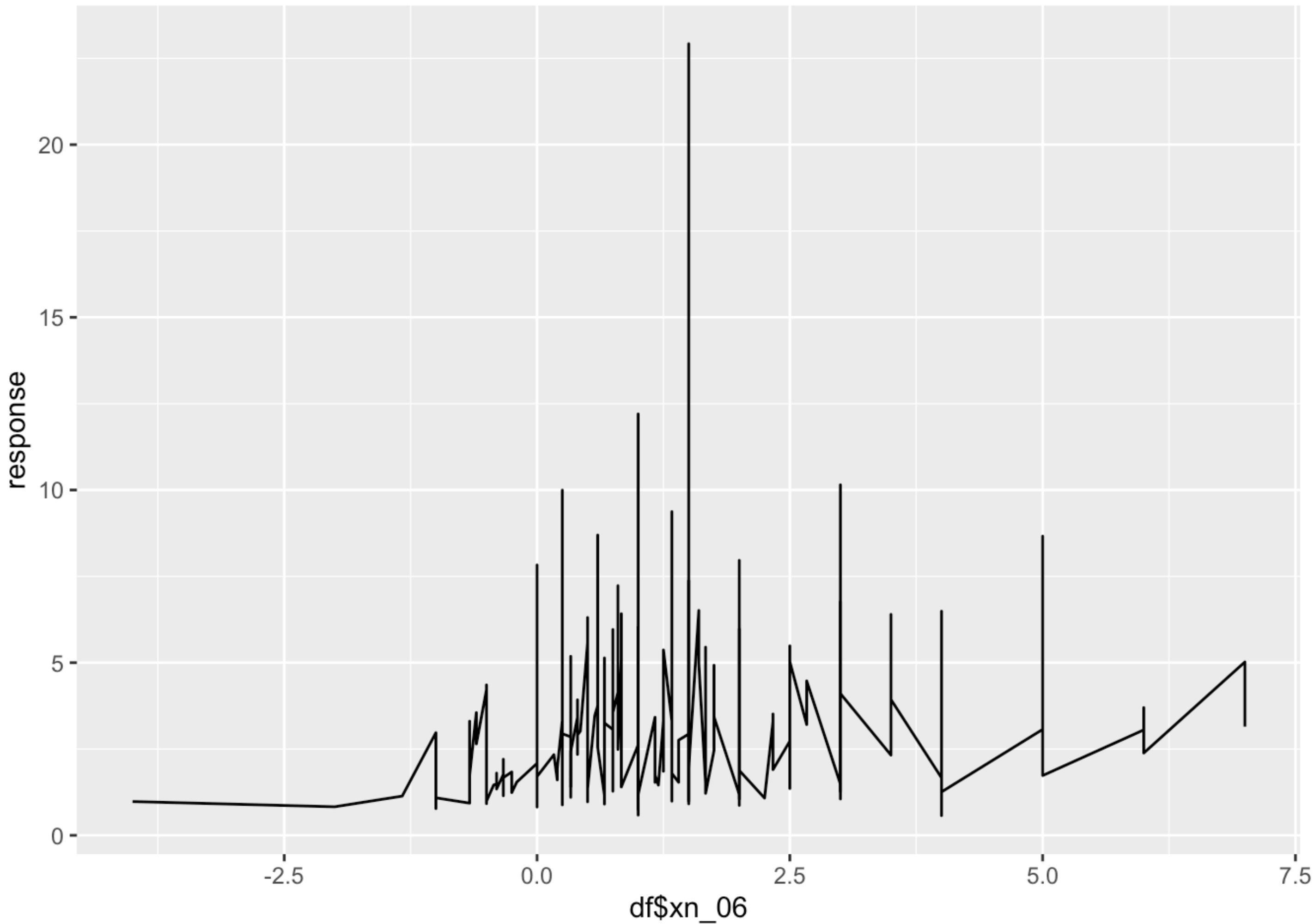

```
df %>%
  ggplot(mapping = aes(df$xn_04, response)) + geom_line()
```



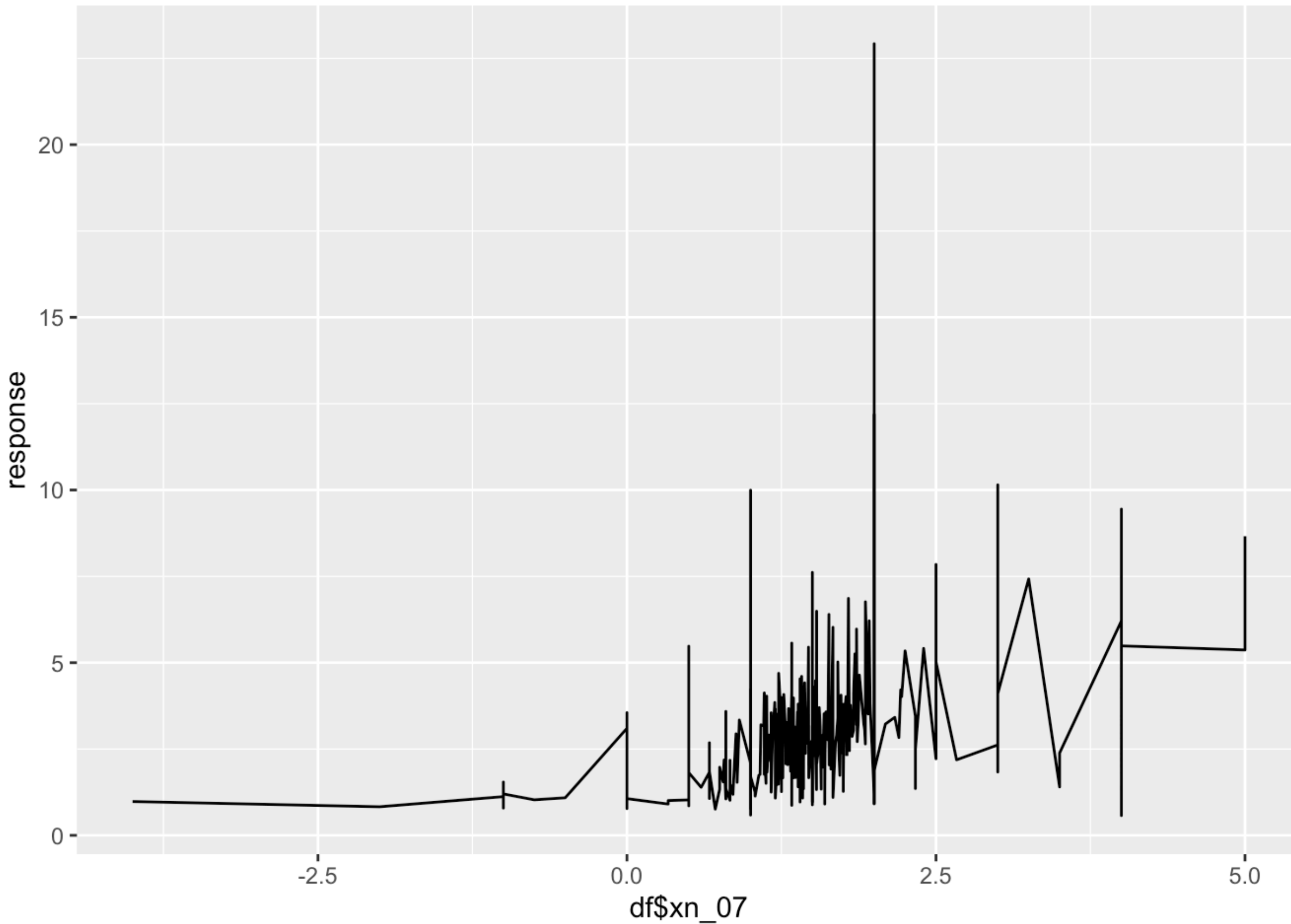

```
df %>%
  ggplot(mapping = aes(df$xn_05, response)) + geom_line()
```



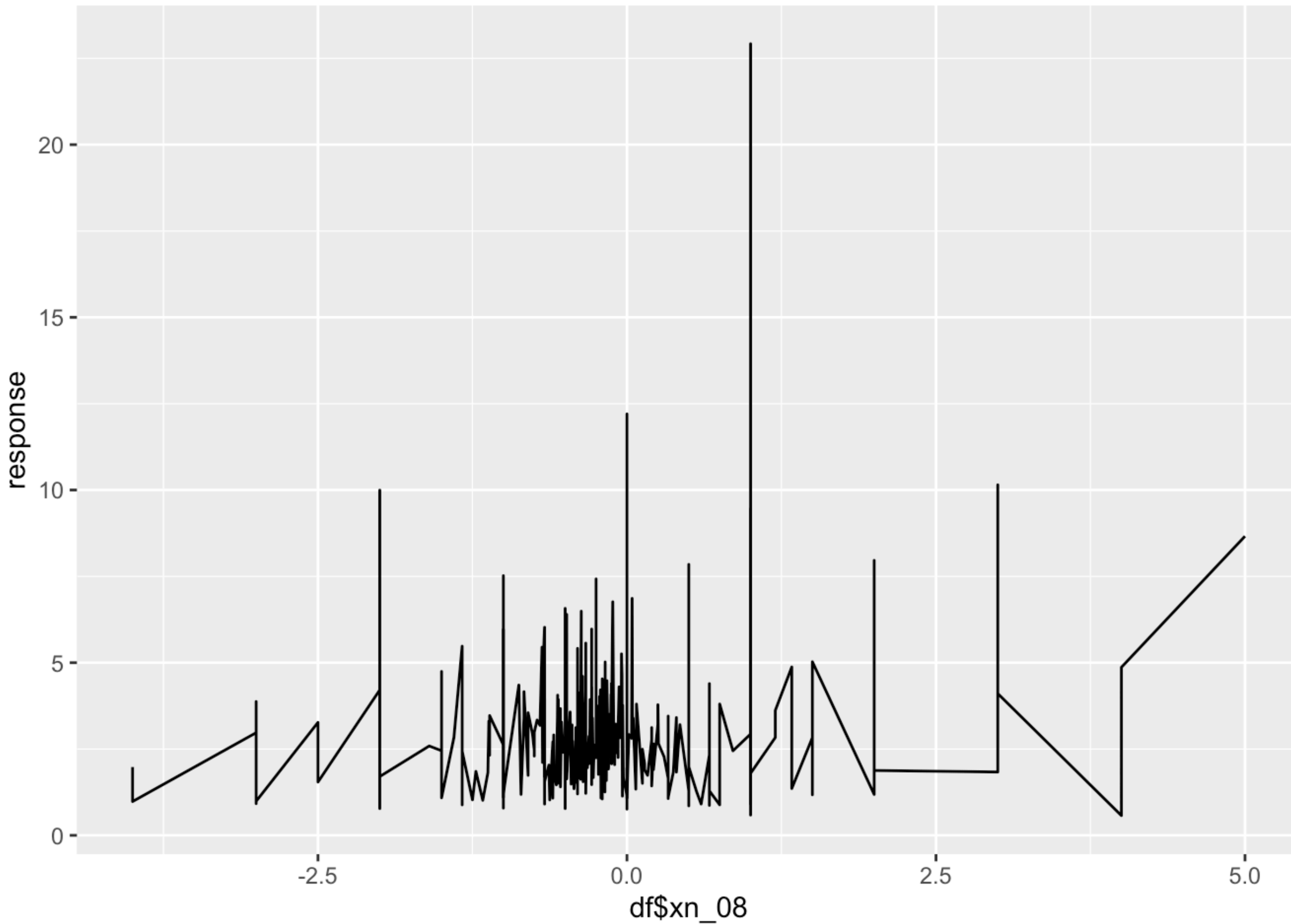

```
df %>%
  ggplot(mapping = aes(df$xn_06, response)) + geom_line()
```



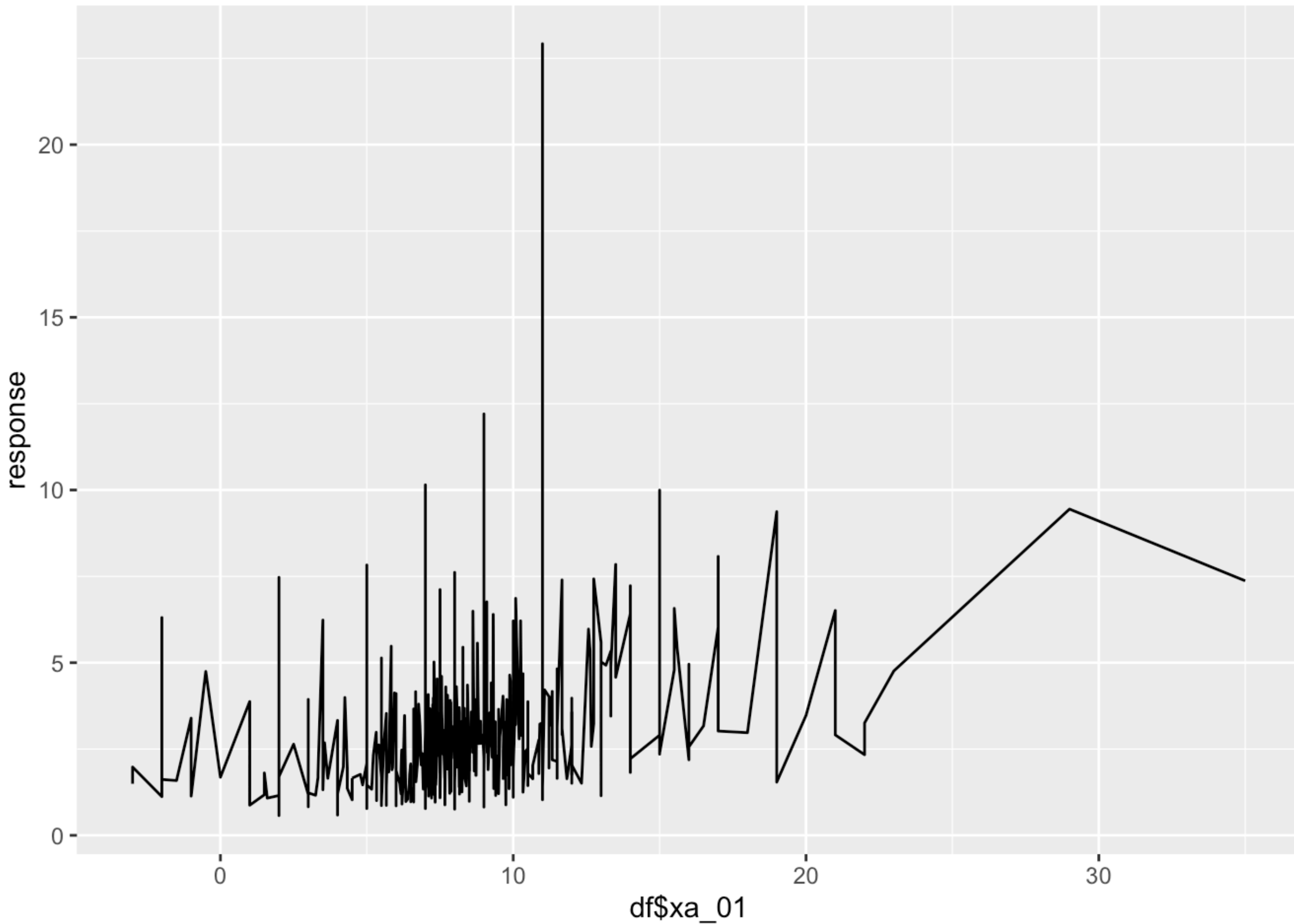

```
df %>%
  ggplot(mapping = aes(df$xn_07, response)) + geom_line()
```



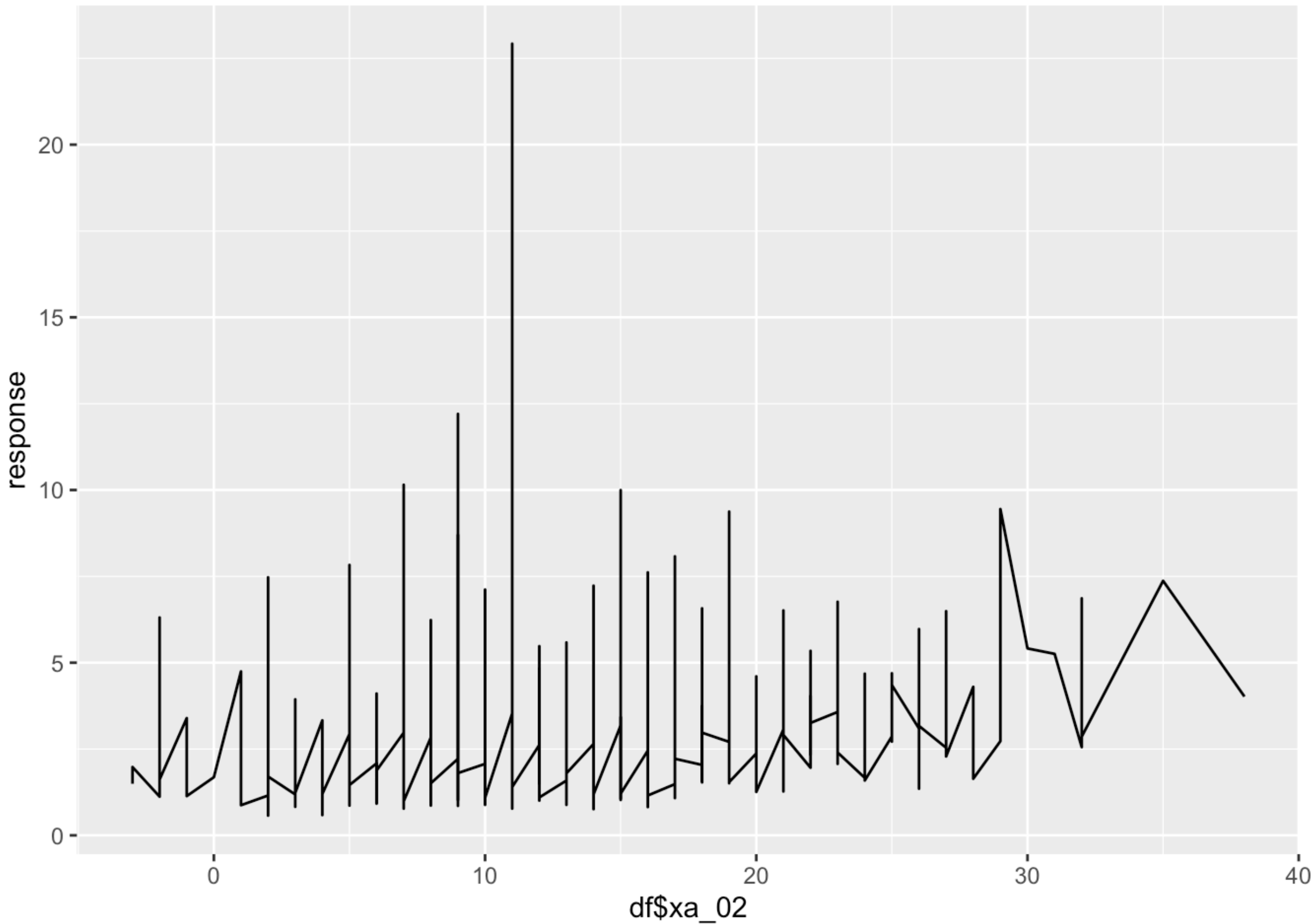

```
df %>%
  ggplot(mapping = aes(df$xn_08, response)) + geom_line()
```



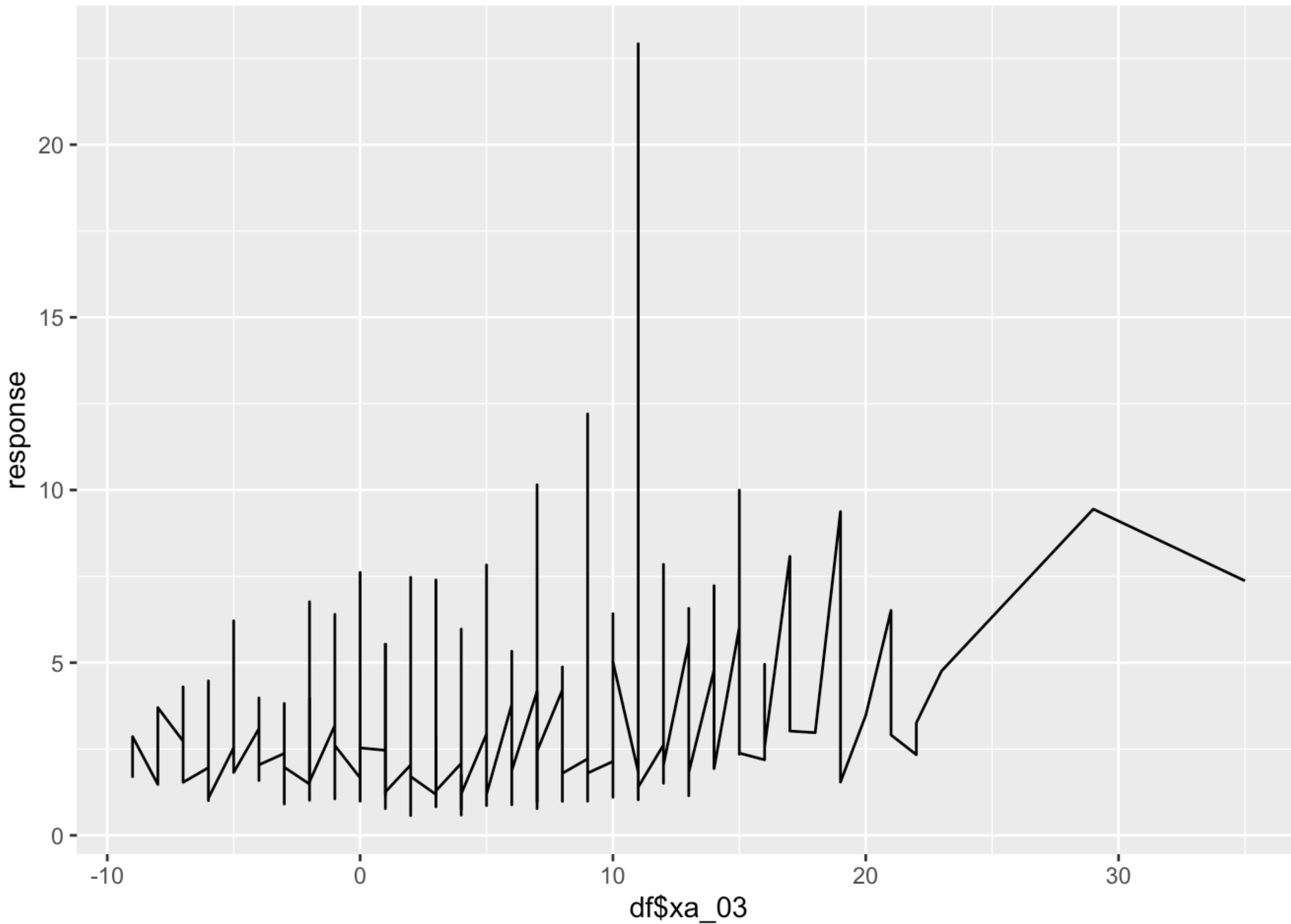

```
df %>%
  ggplot(mapping = aes(df$xa_01, response)) + geom_line()
```



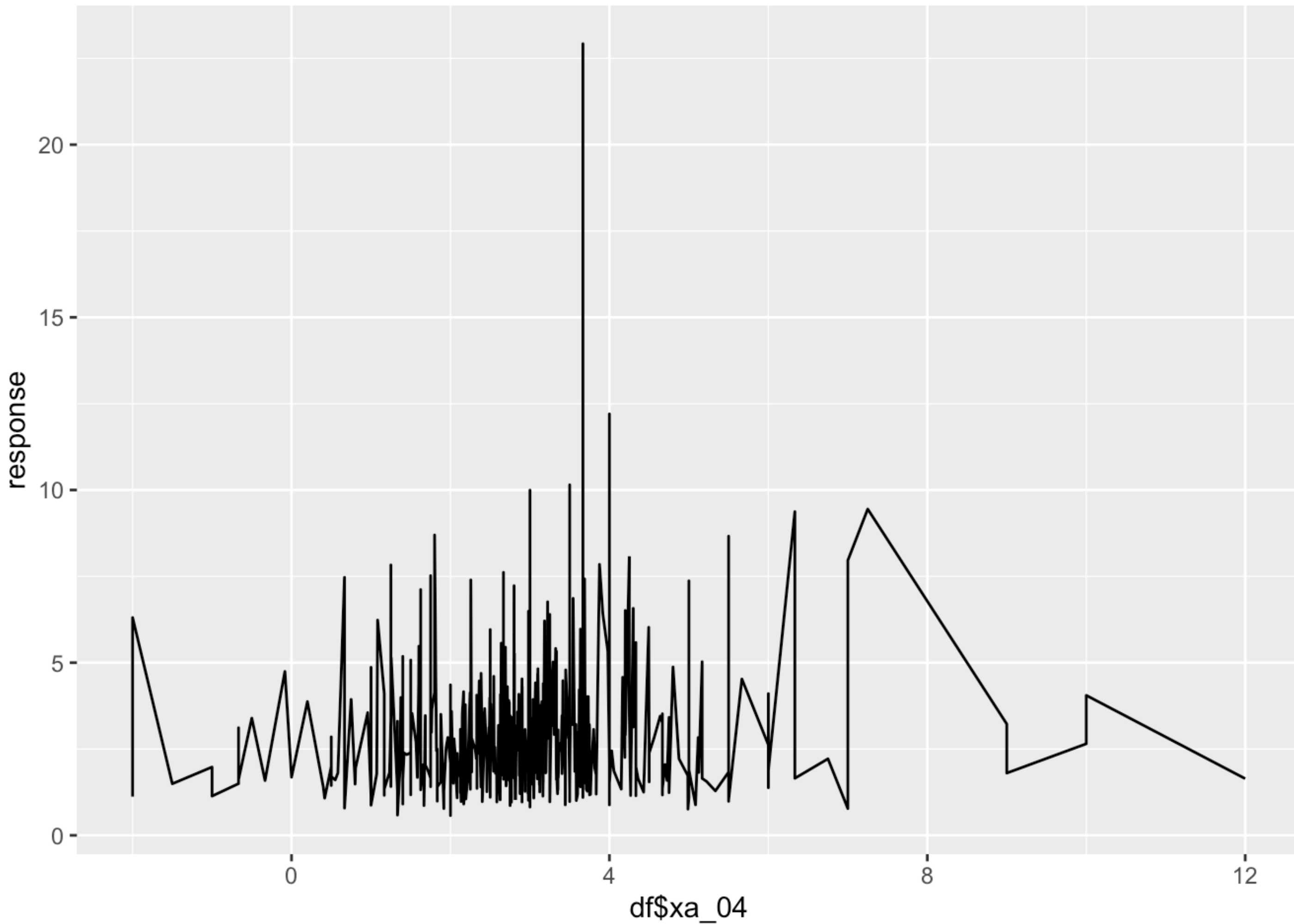

```
df %>%
  ggplot(mapping = aes(df$xa_02, response)) + geom_line()
```



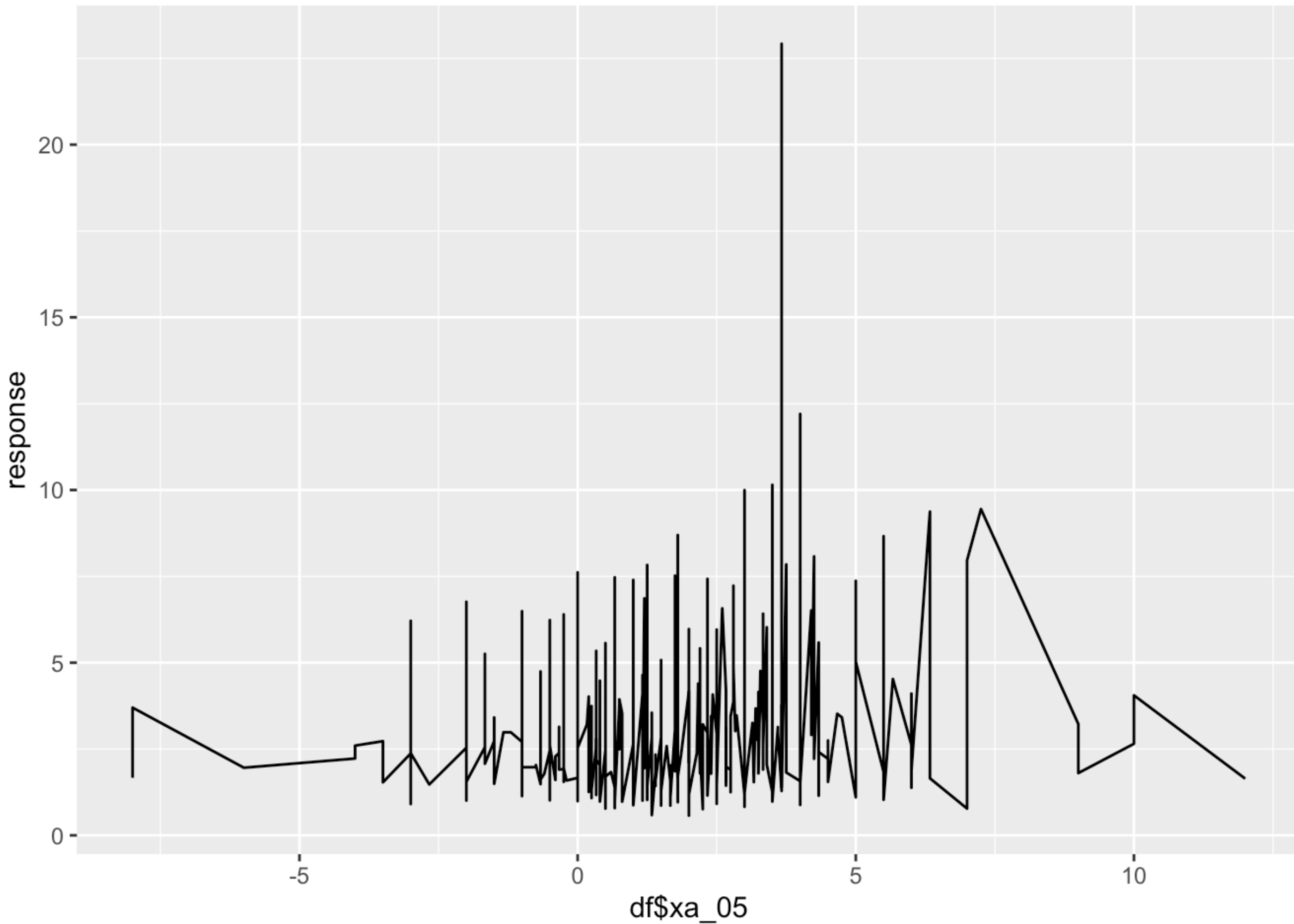

```
df %>%
  ggplot(mapping = aes(df$xa_03, response)) + geom_line()
```



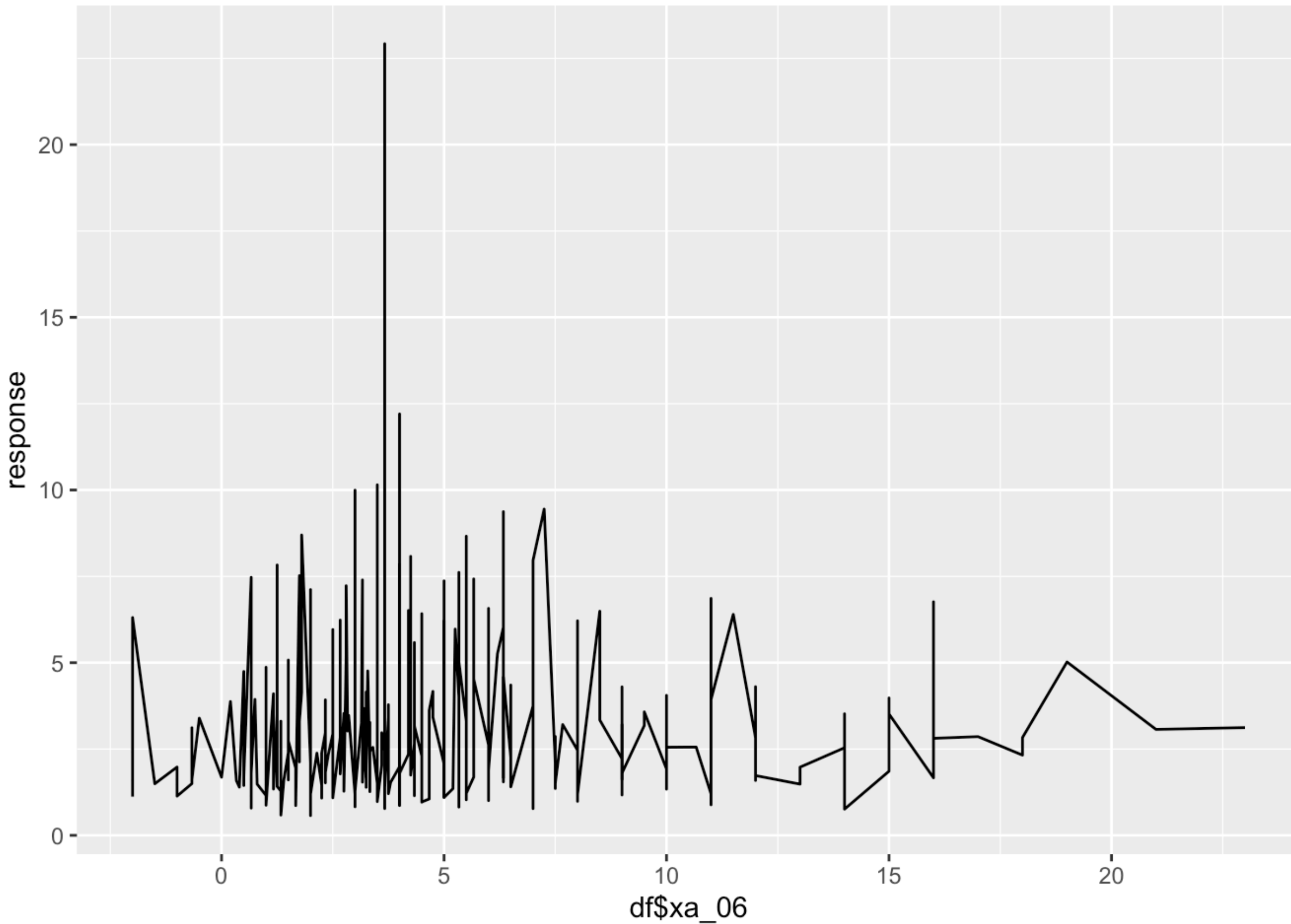

```
df %>%
  ggplot(mapping = aes(df$xa_04, response)) + geom_line()
```



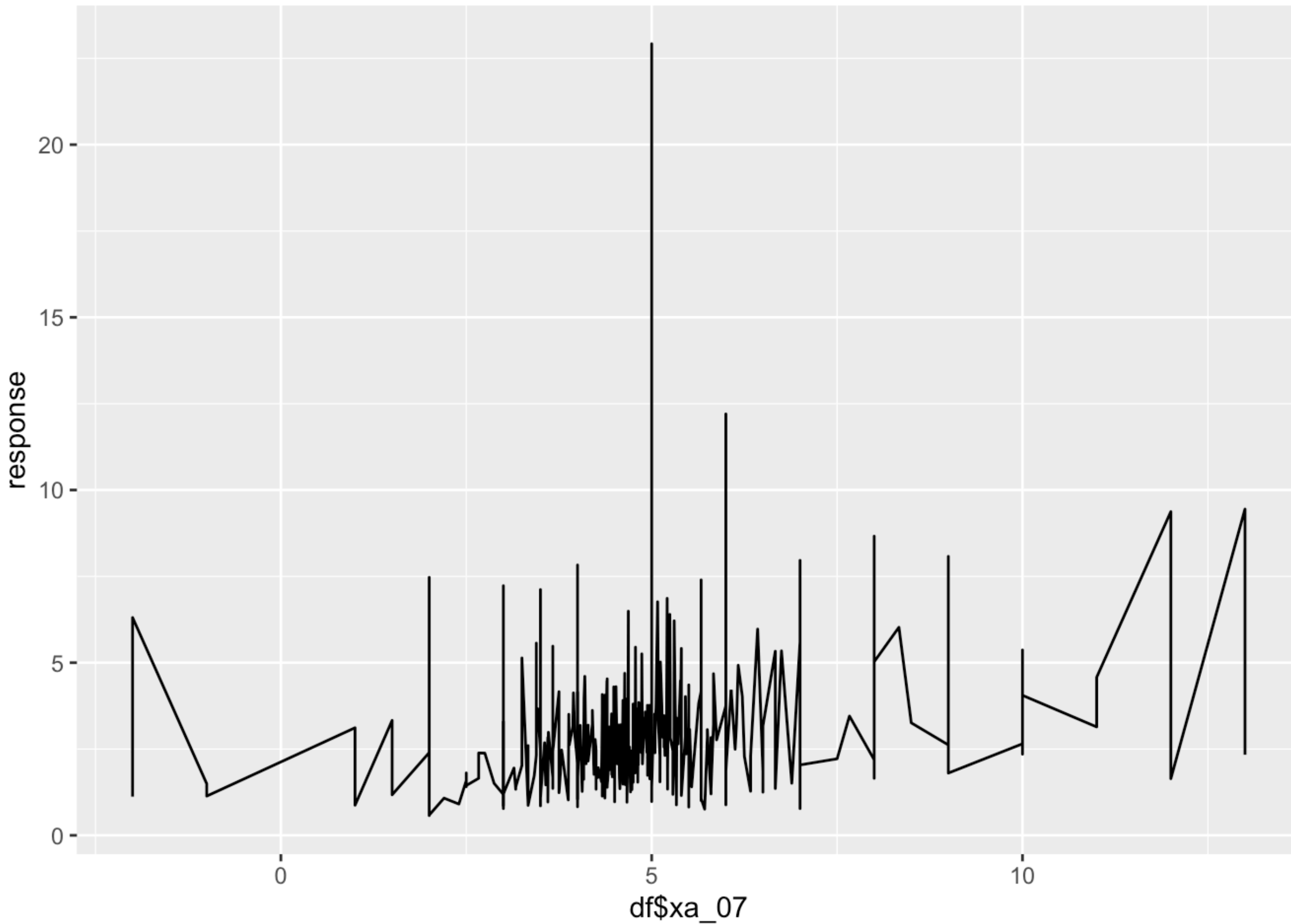

```
df %>%
  ggplot(mapping = aes(df$xa_05, response)) + geom_line()
```



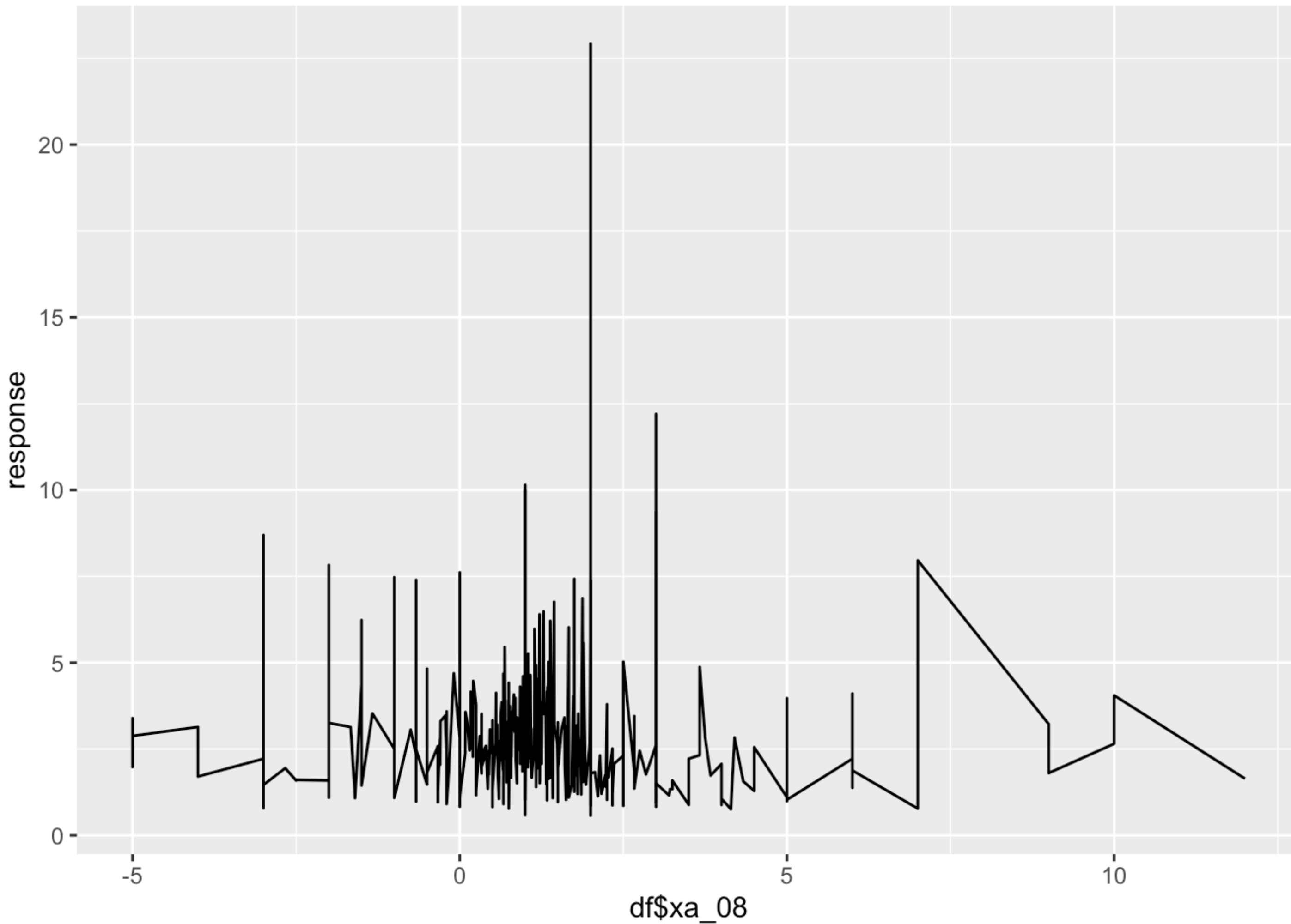

```
df %>%
  ggplot(mapping = aes(df$xa_06, response)) + geom_line()
```



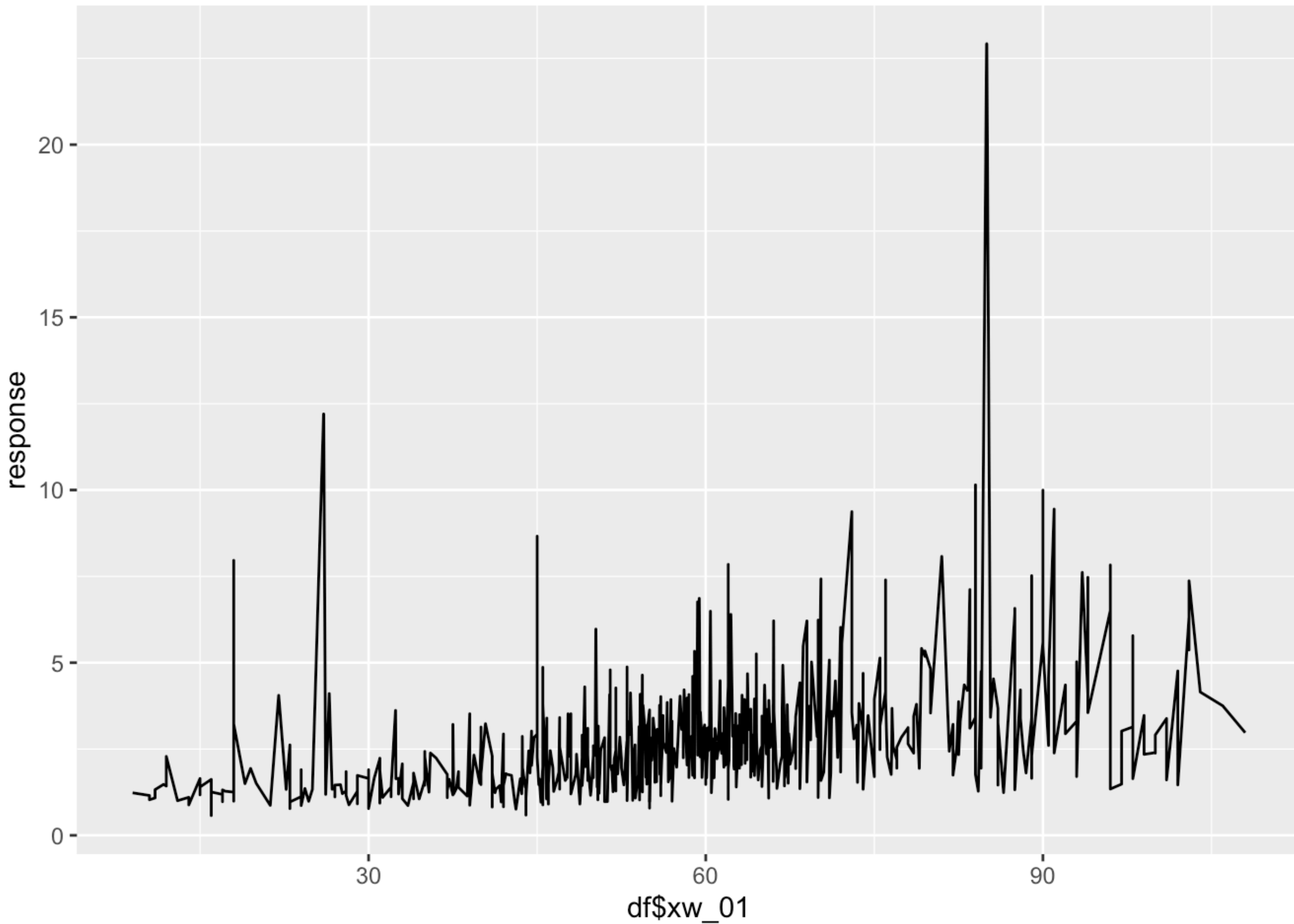

```
df %>%
  ggplot(mapping = aes(df$xa_07, response)) + geom_line()
```



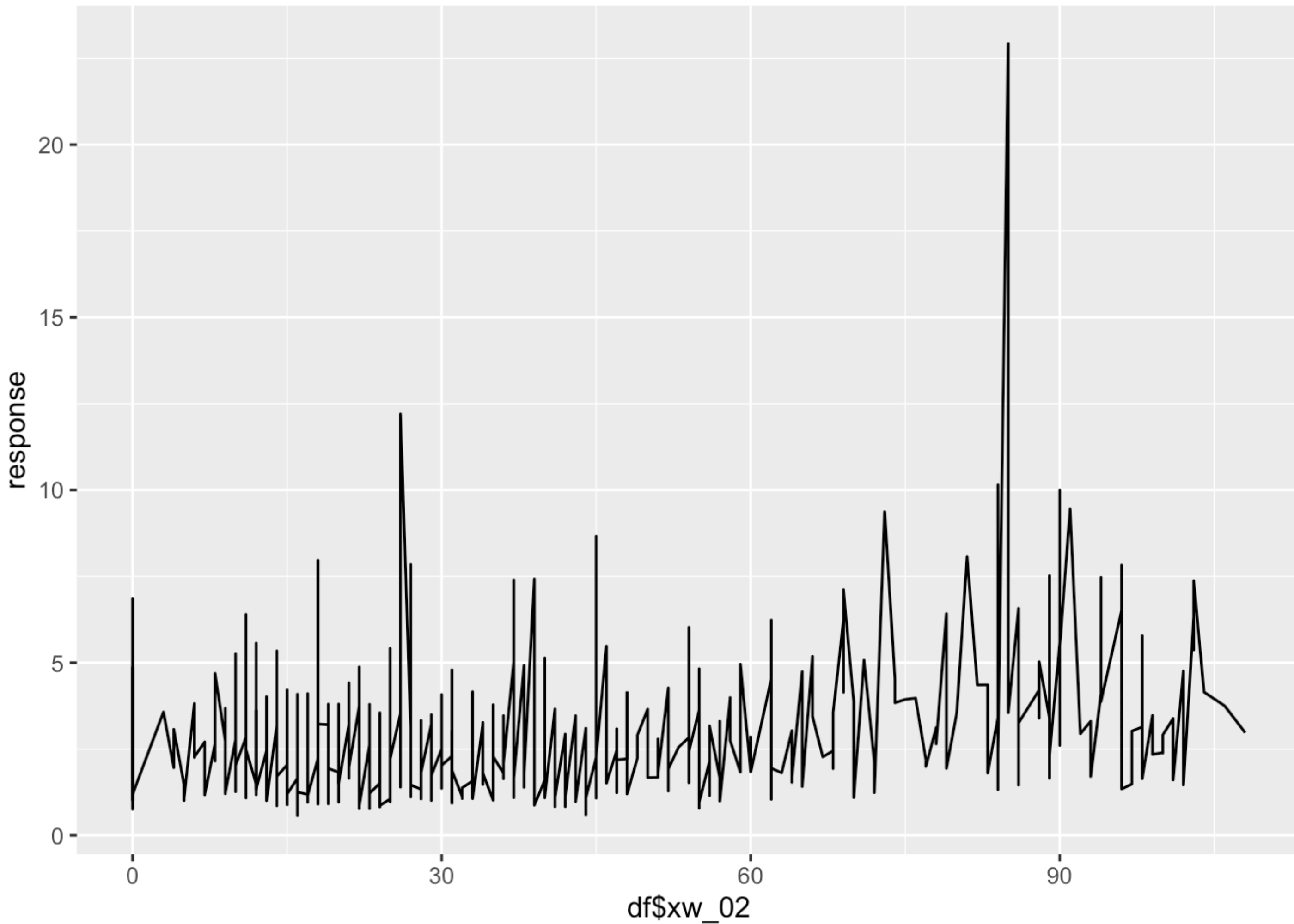

```
df %>%
  ggplot(mapping = aes(df$xa_08, response)) + geom_line()
```



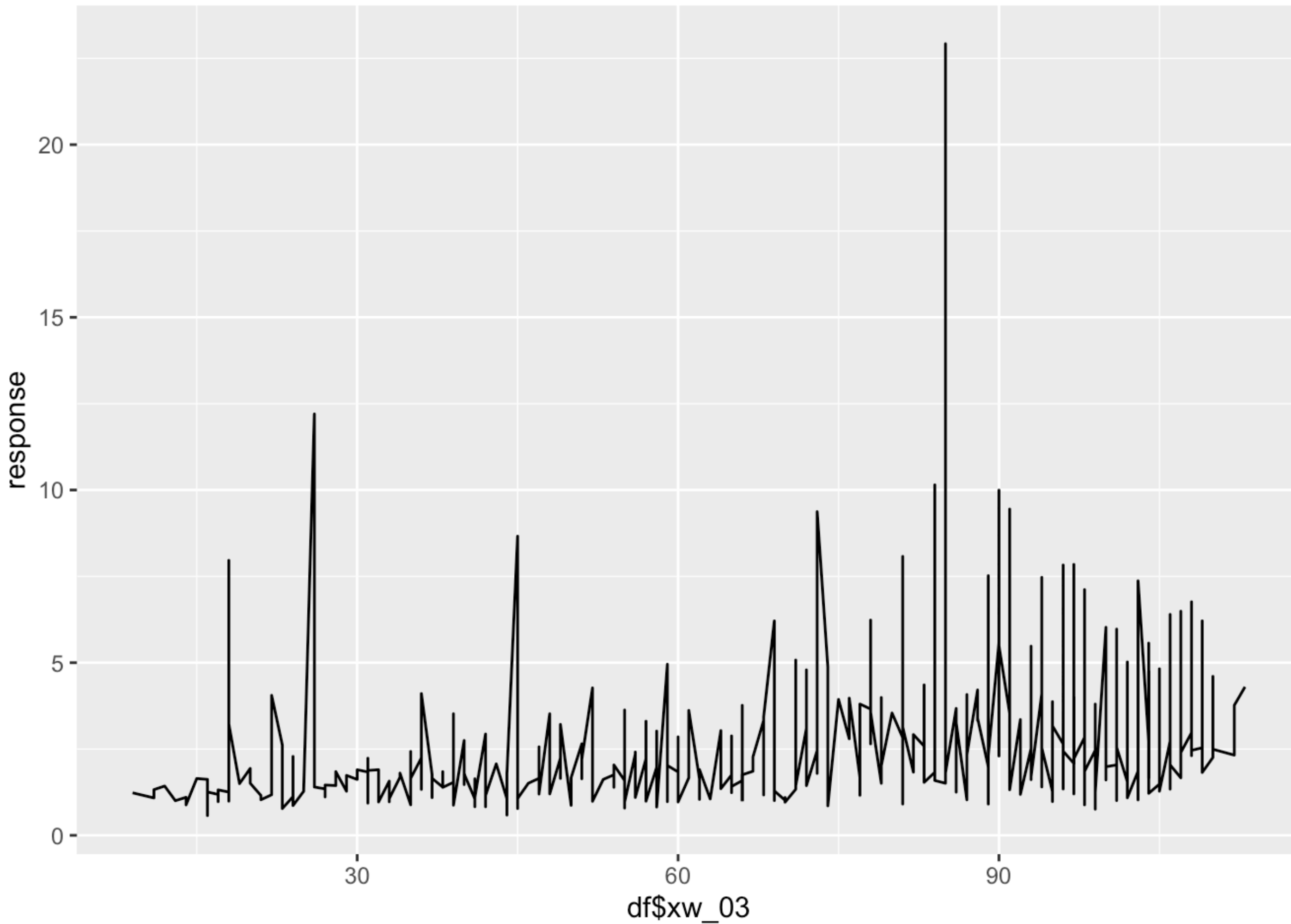

```
df %>%
  ggplot(mapping = aes(df$xw_01, response)) + geom_line()
```



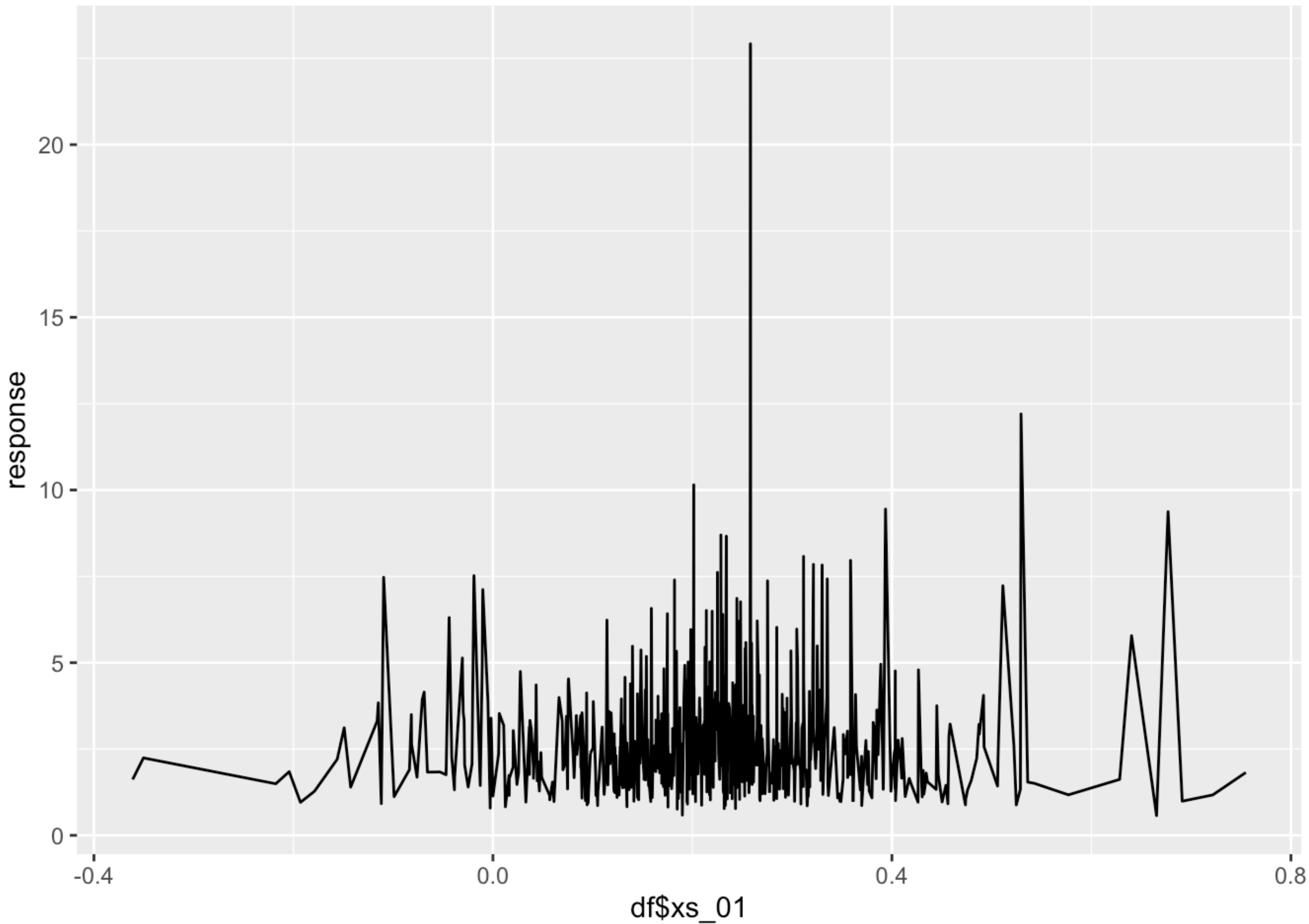

```
df %>%
  ggplot(mapping = aes(df$xw_02, response)) + geom_line()
```



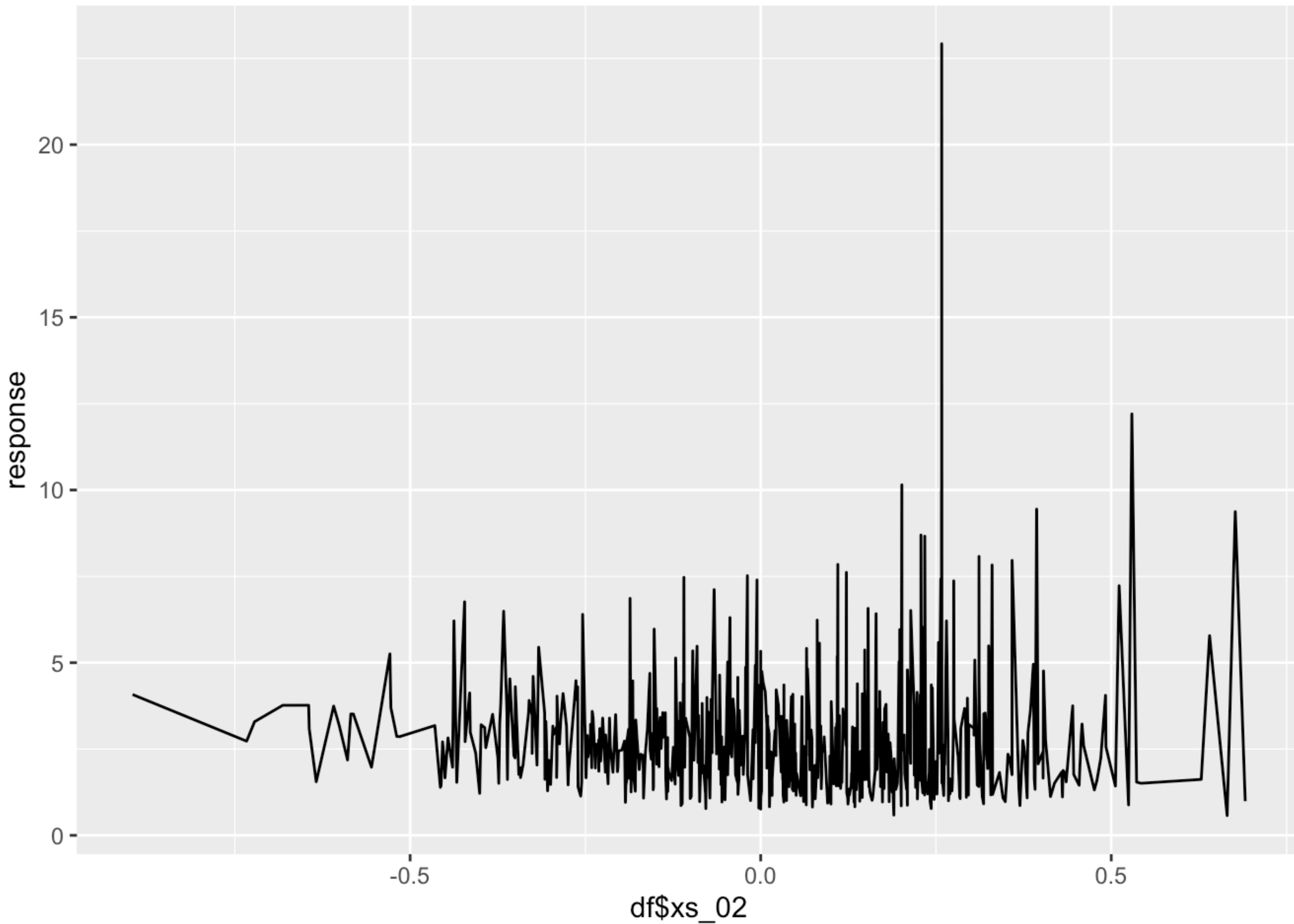

```
df %>%
  ggplot(mapping = aes(df$xw_03, response)) + geom_line()
```



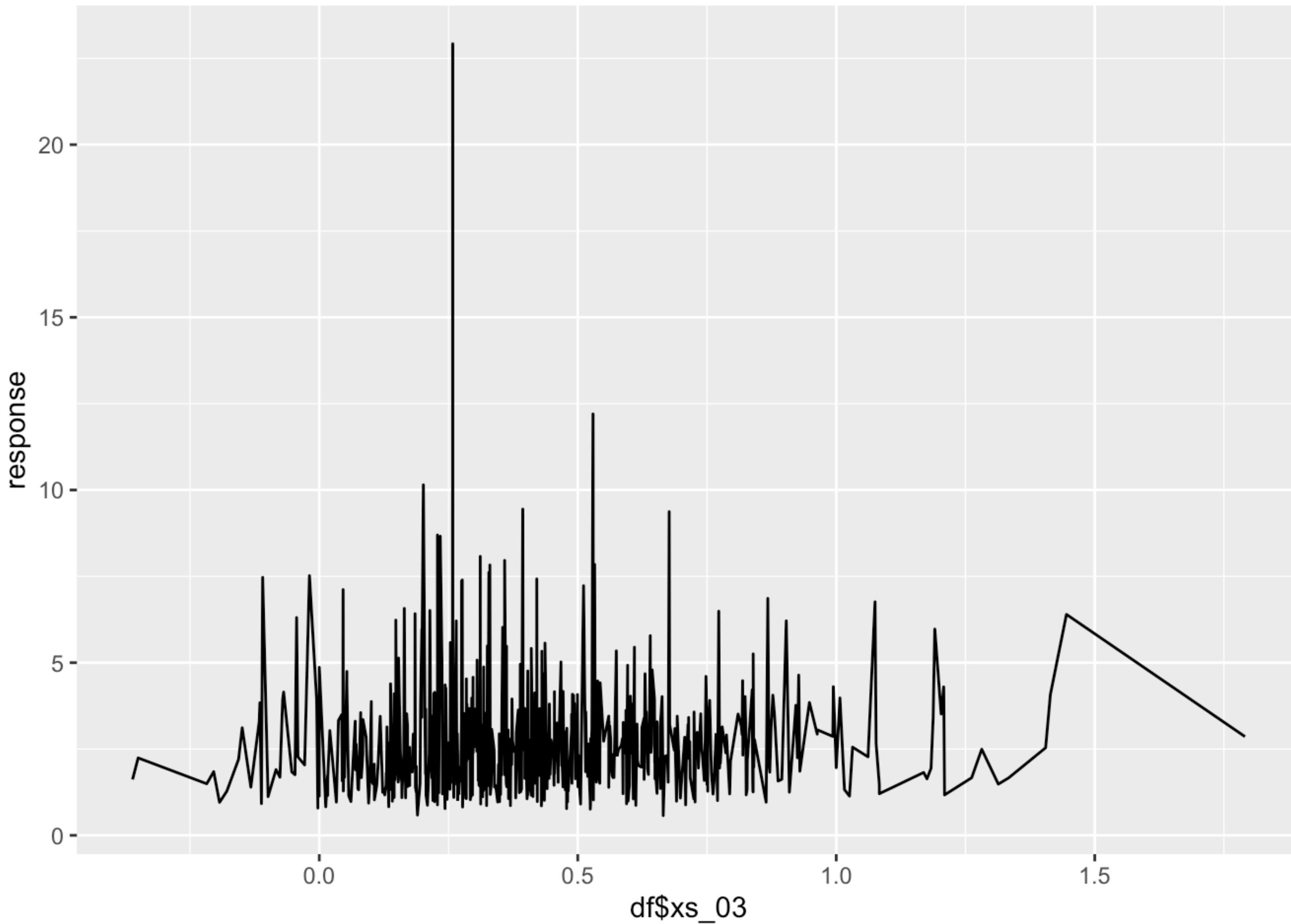

```
df %>%
  ggplot(mapping = aes(df$xs_01, response)) + geom_line()
```



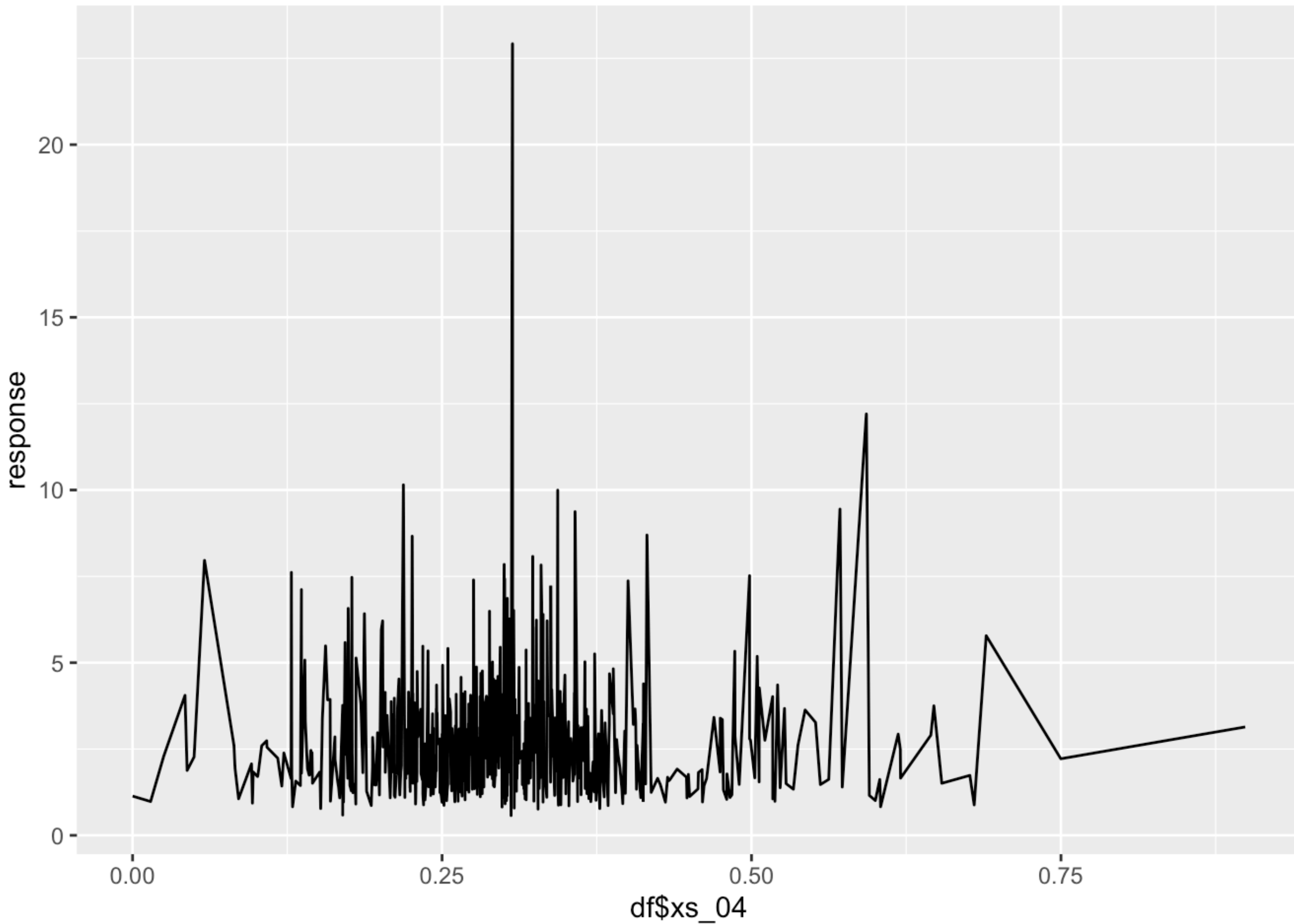

```
df %>%
  ggplot(mapping = aes(df$xs_02, response)) + geom_line()
```



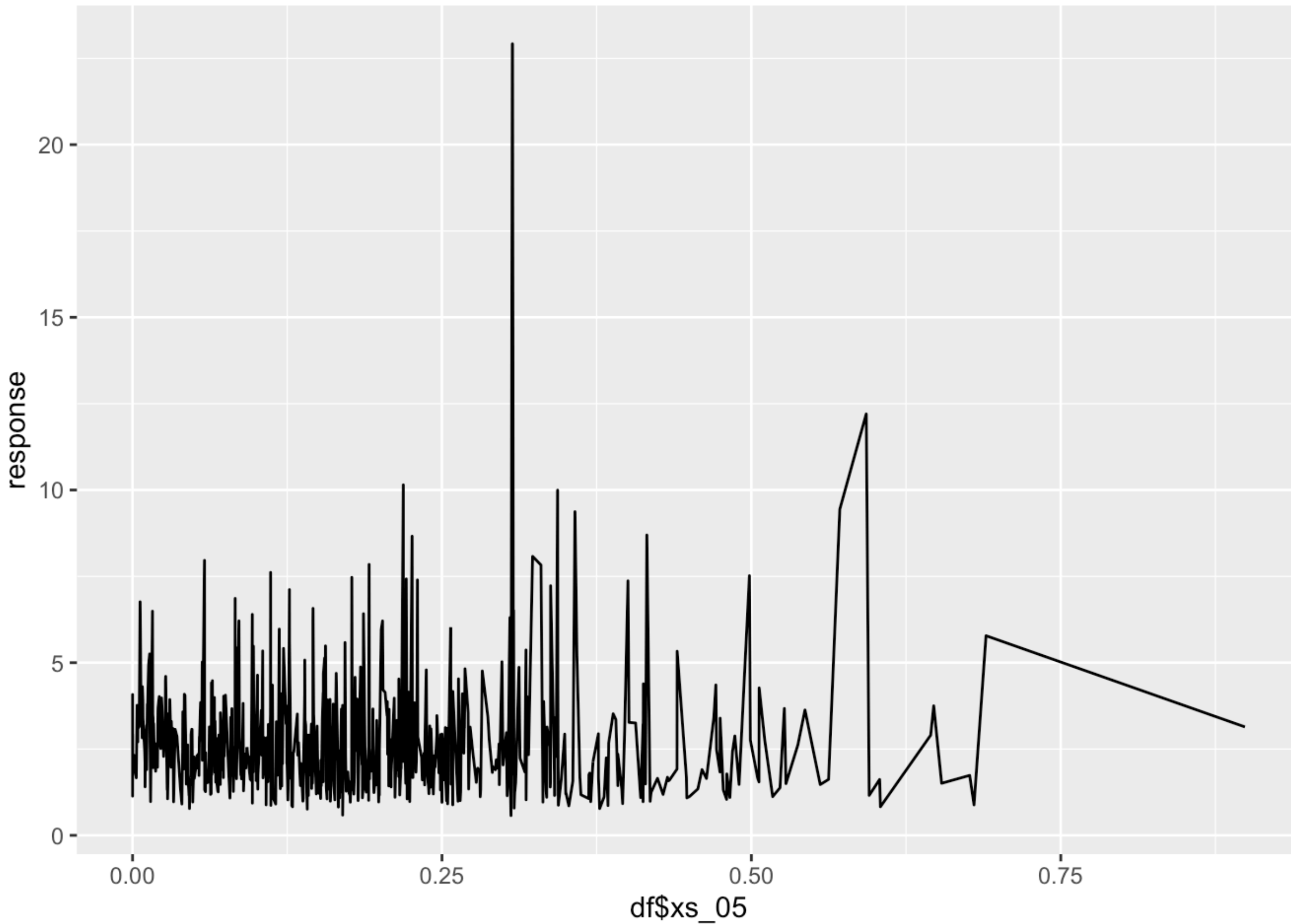

```
df %>%
  ggplot(mapping = aes(df$xs_03, response)) + geom_line()
```



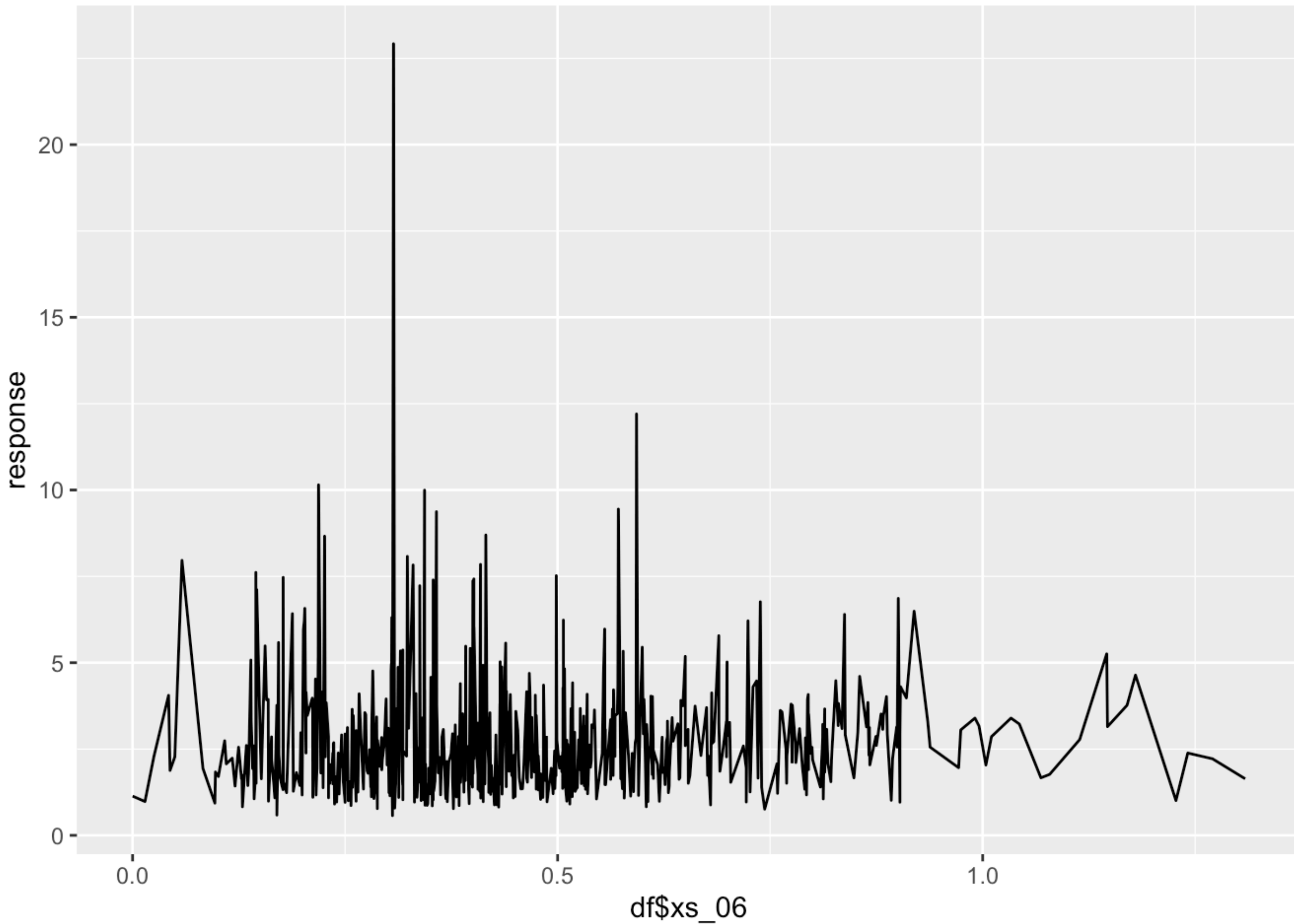

```
df %>%
  ggplot(mapping = aes(df$xs_04, response)) + geom_line()
```




```
df %>%
  ggplot(mapping = aes(df$xs_05, response)) + geom_line()
```




```
df %>%
  ggplot(mapping = aes(df$xs_06, response)) + geom_line()
```

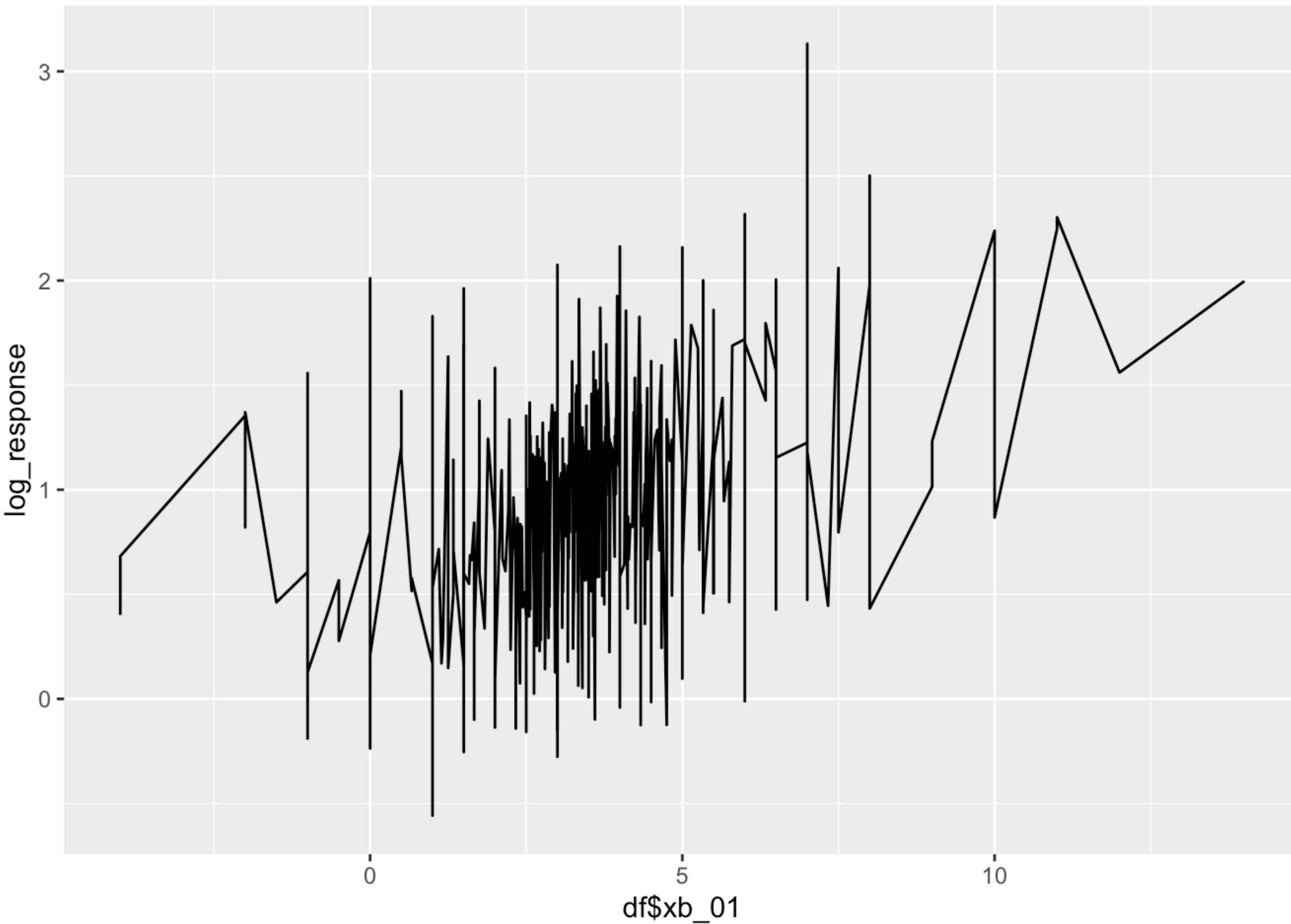


Adding Log-transformed Response

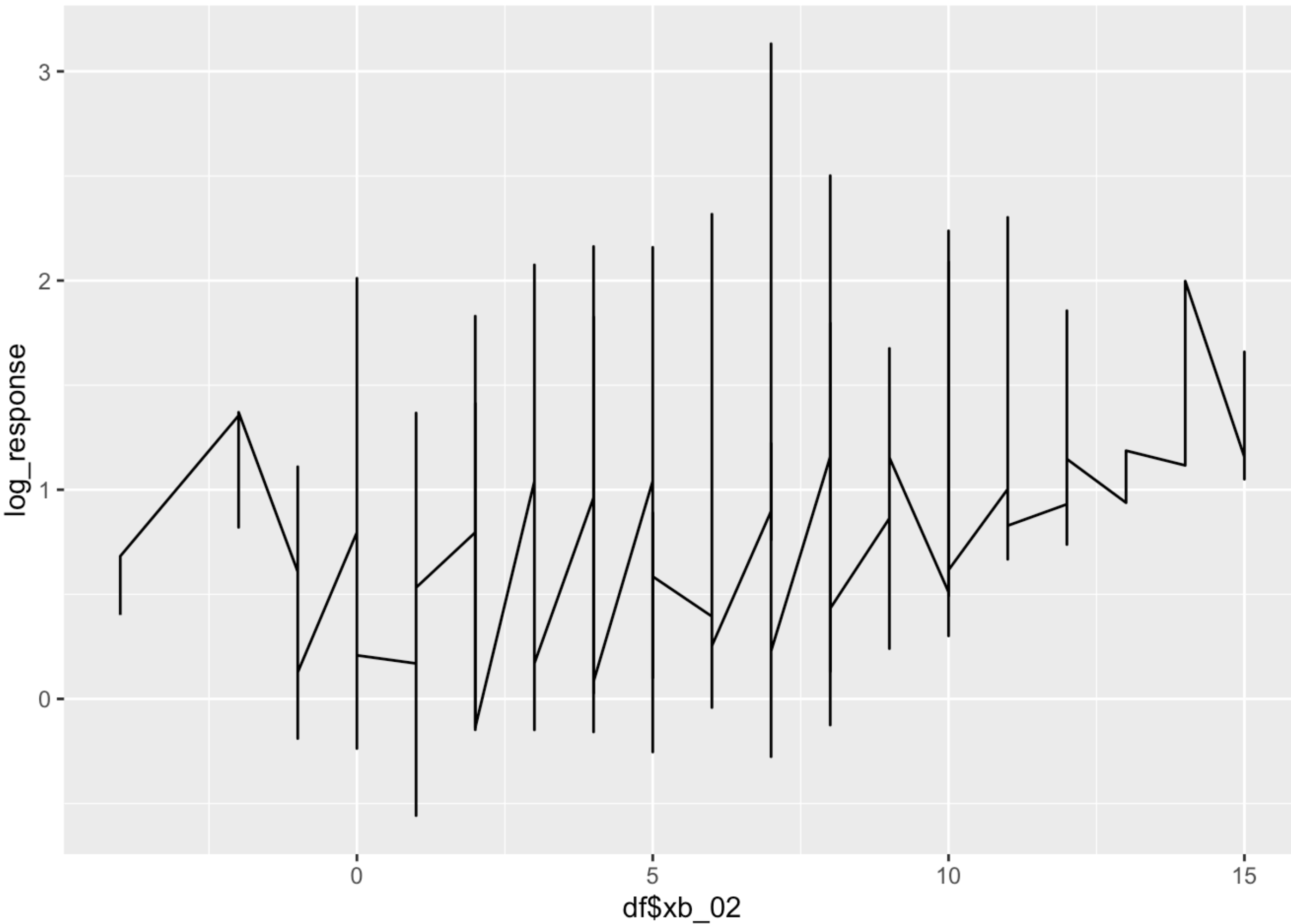
```
df$log_response <- log(df$response)
df %>% glimpse
```

```
## Rows: 677
## Columns: 39
## $ rowid        <dbl> 1, 3, 4, 5, 8, 9, 11, 14, 15, 16, 17, 18, 19, 22, 24, 25,…
## $ region       <chr> "XX", "XX", "XX", "XX", "XX", "XX", "XX", "XX", "XX", "XX…
## $ customer     <chr> "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B…
## $ xb_01        <dbl> 4.000000, 1.000000, 2.000000, 2.520000, 2.548387, 3.07142…
## $ xb_02        <dbl> 4, 1, 2, 11, 6, 6, 10, 12, 9, 10, 8, 10, 10, 8, 6, 10, 13…
## $ xb_03        <dbl> 4, 1, 2, -6, -1, 1, -4, -4, -2, -4, -2, -2, -2, -4, 1, -4…
## $ xn_01        <dbl> 3.0000000, 2.0000000, 2.0000000, 1.5333333, 0.8387097, 1.…
## $ xn_02        <dbl> 3, 2, 4, 9, 3, 8, 6, 10, 10, 4, 6, 8, 9, 5, 7, 12, 12, 6,…
## $ xn_03        <dbl> 3, 2, 0, -3, -4, -2, -5, -6, -3, -5, -3, -6, -4, -3, 0, -…
## $ xa_01        <dbl> 12.000000, 3.000000, 9.000000, 7.080000, 6.451613, 6.8571…
## $ xa_02        <dbl> 12, 3, 9, 29, 17, 18, 24, 27, 20, 19, 15, 24, 24, 15, 14,…
## $ xa_03        <dbl> 12, 3, 9, -7, -2, 2, -9, -5, -3, -3, -1, 1, -2, -3, 3, -4…
## $ xb_04        <dbl> 1.3333333, 1.0000000, 1.0000000, 0.8950476, 1.2247312, 1.…
## $ xb_05        <dbl> 1.3333333, 1.0000000, 1.0000000, -2.0000000, -0.5000000, …
## $ xb_06        <dbl> 1.333333, 1.000000, 1.000000, 4.000000, 4.000000, 3.00000…
## $ xb_07        <dbl> 4.000000, 1.000000, 2.000000, 1.933333, 1.967742, 1.71428…
## $ xb_08        <dbl> -1.00000000, 1.00000000, 0.00000000, -0.08000000, 0.35483…
## $ xn_04        <dbl> 1.0000000, 2.0000000, 1.0000000, 0.5268889, 0.4688172, 0.…
## $ xn_05        <dbl> 1.0000000, 2.0000000, 0.0000000, -1.0000000, -1.3333333, …
## $ xn_06        <dbl> 1.0, 2.0, 2.0, 2.5, 3.0, 2.0, 4.0, 4.0, 3.0, 2.0, 2.0, 2.…
## $ xn_07        <dbl> 3.000000, 2.000000, 2.500000, 1.493333, 1.225806, 1.64285…
## $ xn_08        <dbl> -1.0000000, 2.0000000, -1.0000000, -0.4400000, -0.4516129…
## $ xa_04        <dbl> 6.000000, 3.000000, 6.750000, 2.425333, 3.023656, 2.68571…
## $ xa_05        <dbl> 6.0000000, 3.0000000, 4.5000000, -3.5000000, -0.6666667, …
## $ xa_06        <dbl> 6.000000, 3.000000, 9.000000, 9.000000, 13.000000, 6.0000…
## $ xa_07        <dbl> 9.000000, 3.000000, 7.500000, 4.466667, 4.612903, 4.07142…
## $ xa_08        <dbl> 3.0000000, 3.0000000, 6.0000000, 0.7066667, 1.3225806, 1.…
## $ xw_01        <dbl> 23.00000, 17.00000, 52.50000, 64.52564, 54.75758, 58.3333…
## $ xw_02        <dbl> 23, 17, 48, 0, 12, 15, 0, 0, 0, 7, 14, 0, 0, 0, 8, 8, 0, …
## $ xw_03        <dbl> 23, 17, 57, 106, 105, 101, 107, 109, 109, 104, 109, 99, 1…
## $ xs_01        <dbl> 0.262073307, 0.330804757, 0.239795763, 0.142106837, 0.244…
## $ xs_02        <dbl> 0.26207331, 0.33080476, 0.19049123, -0.73321509, -0.12204…
## $ xs_03        <dbl> 0.2620733, 0.3308048, 0.2891003, 0.5500723, 1.3134719, 0.…
## $ xs_04        <dbl> 0.5375576, 0.4286607, 0.3676937, 0.2865445, 0.2375470, 0.…
## $ xs_05        <dbl> 0.5375575604, 0.4286607050, 0.2485001680, 0.0000000000, 0…
## $ xs_06        <dbl> 0.5375576, 0.4286607, 0.4868872, 0.6357541, 0.4327004, 0.…
## $ response     <dbl> 2.617991, 1.184632, 2.216626, 2.726715, 1.483323, 2.03927…
## $ outcome      <chr> "non_event", "non_event", "event", "non_event", "non_even…
## $ log_response <dbl> 0.9624073, 0.1694321, 0.7959862, 1.0030975, 0.3942847, 0.…
```

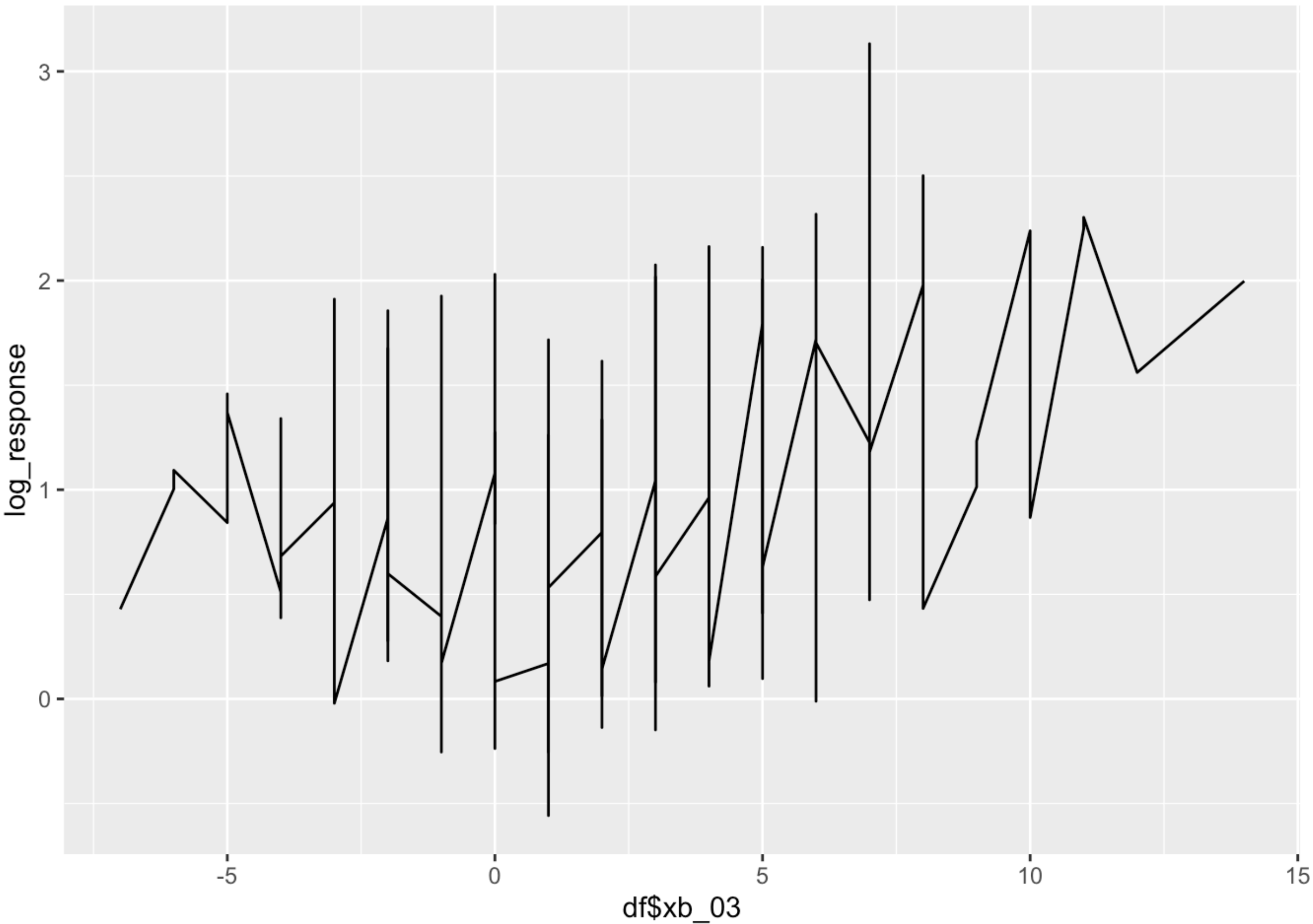
```
df %>%
  ggplot(mapping = aes(df$xb_01, log_response)) + geom_line()
```



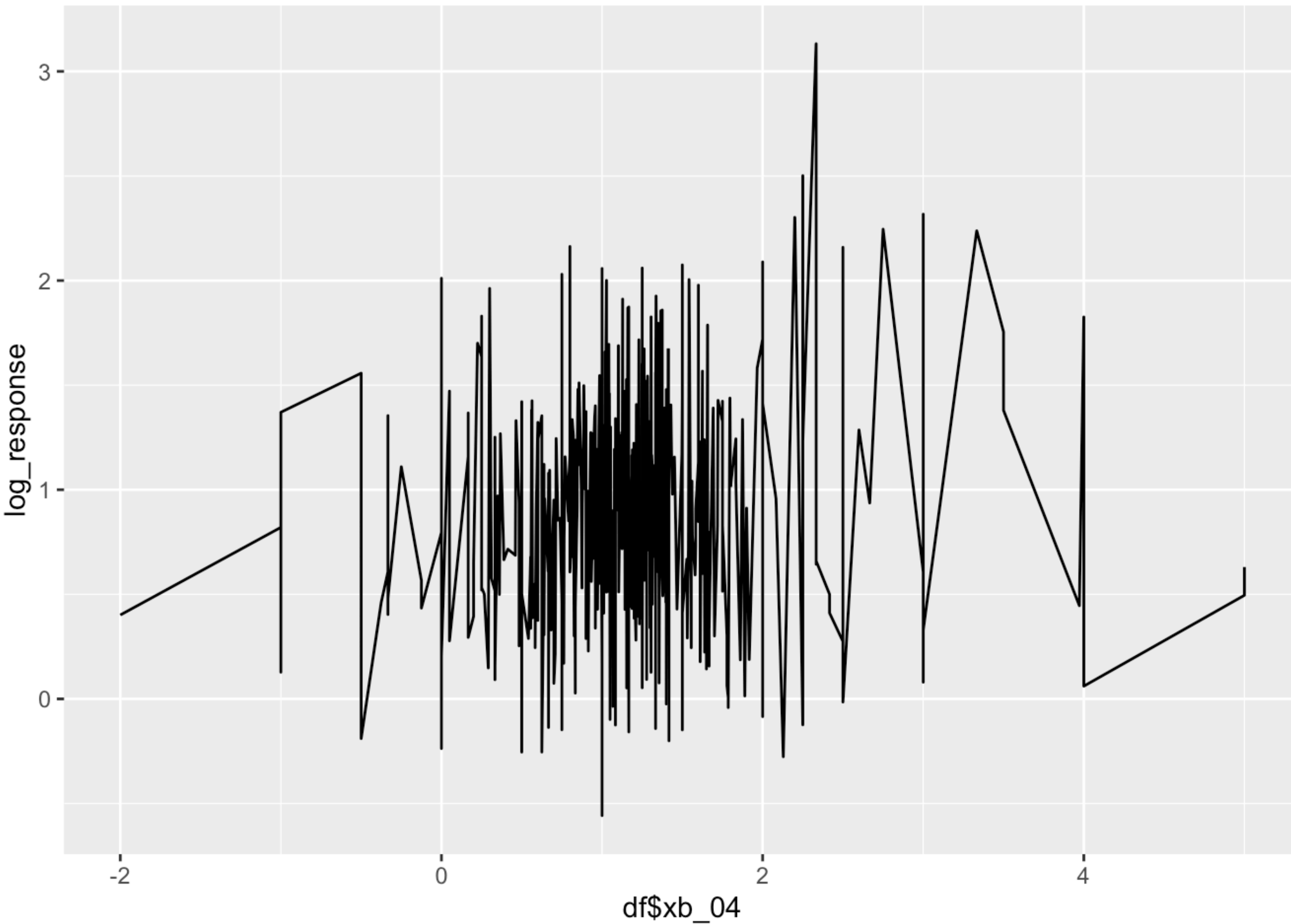

```
df %>%
  ggplot(mapping = aes(df$xb_02, log_response)) + geom_line()
```



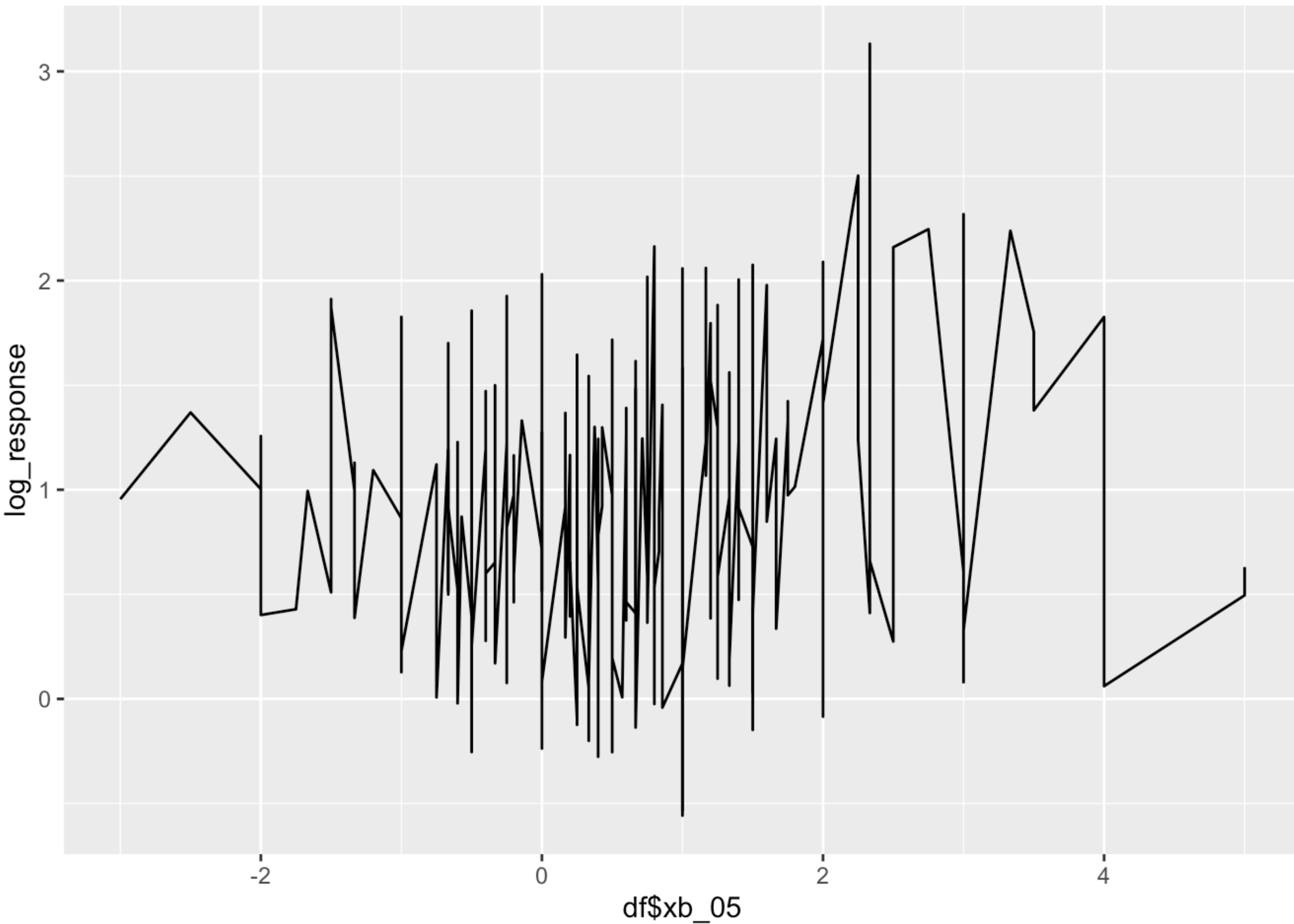

```
df %>%
  ggplot(mapping = aes(df$xb_03, log_response)) + geom_line()
```



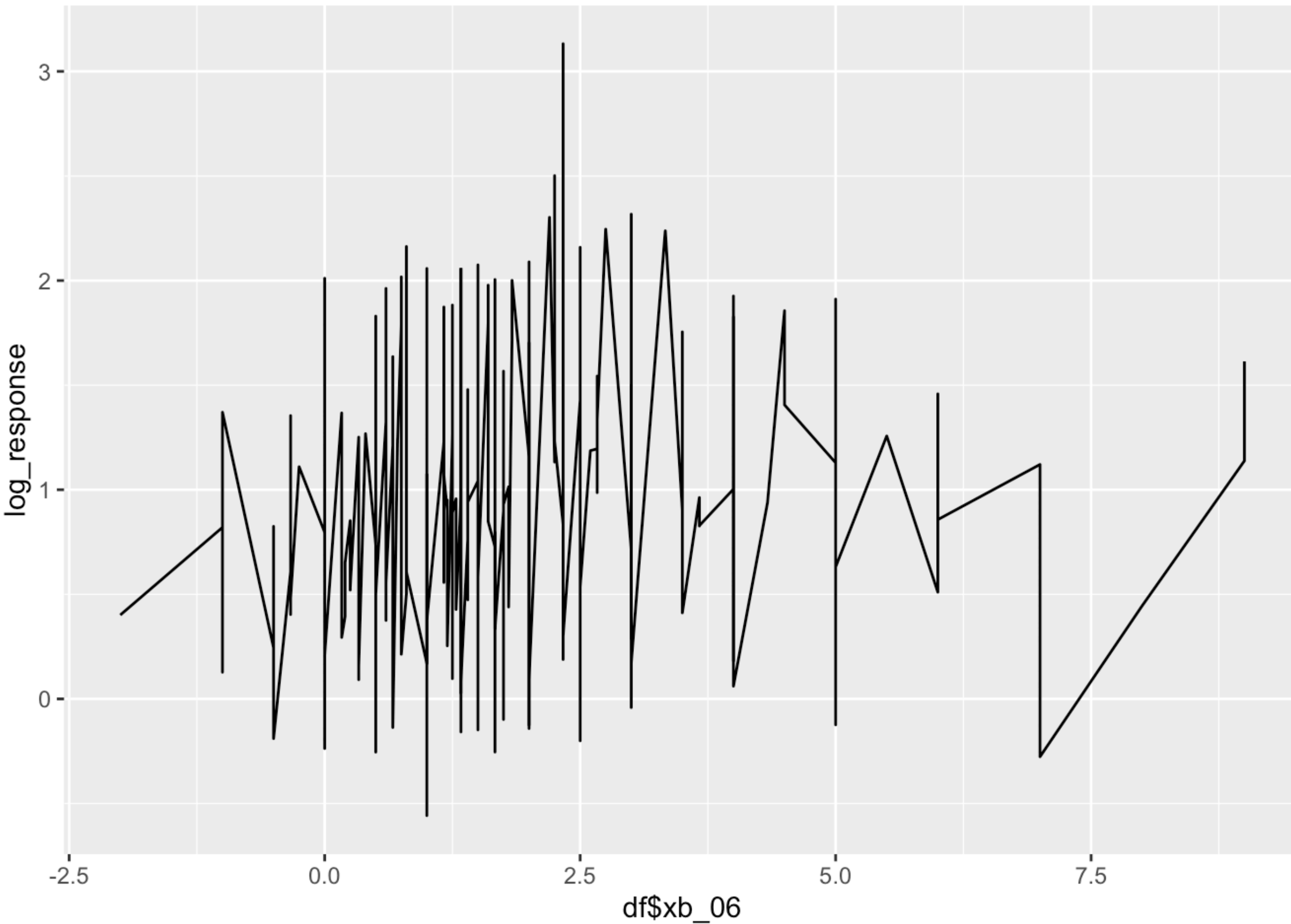

```
df %>%
  ggplot(mapping = aes(df$xb_04, log_response)) + geom_line()
```



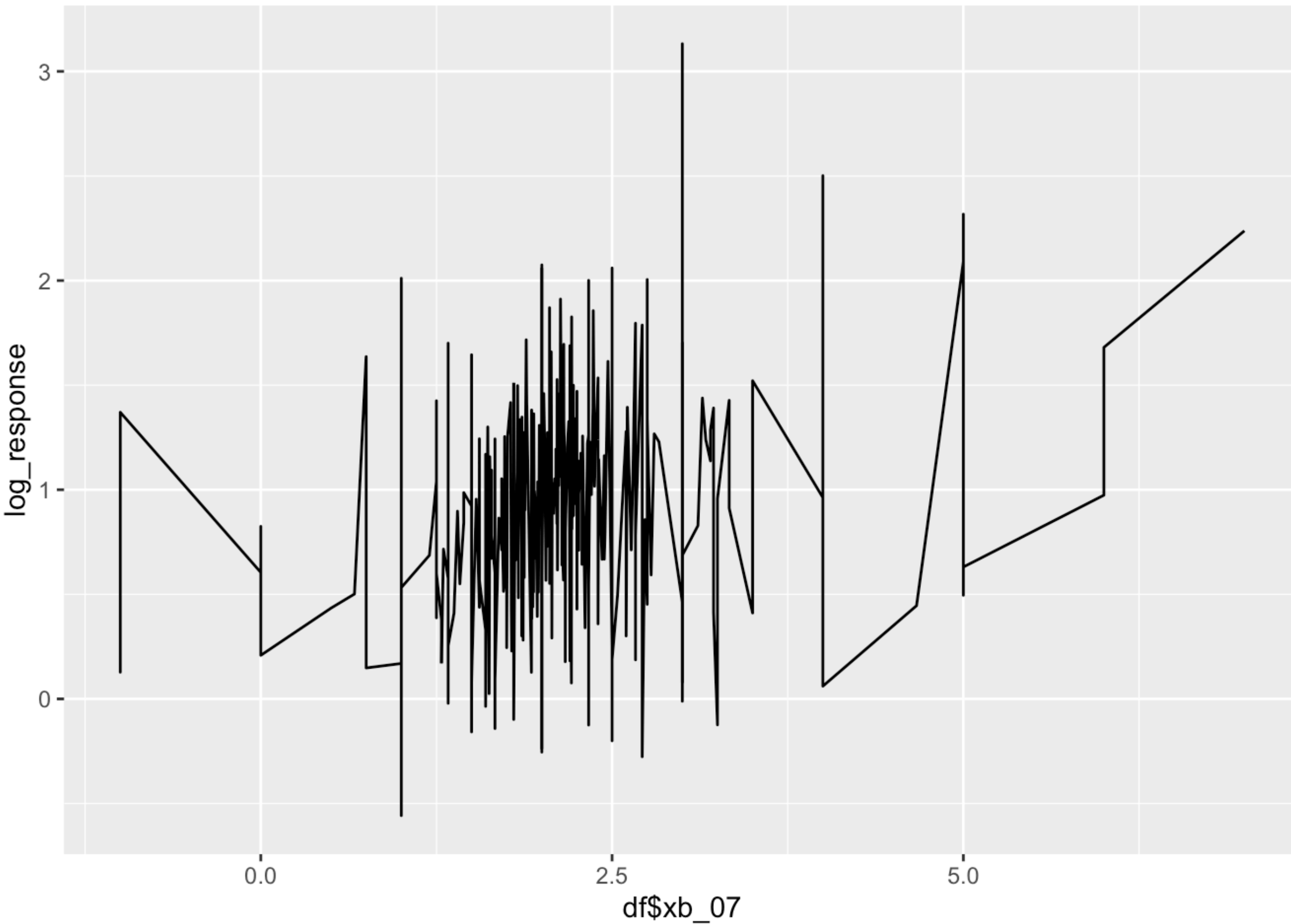

```
df %>%
  ggplot(mapping = aes(df$xb_05, log_response)) + geom_line()
```



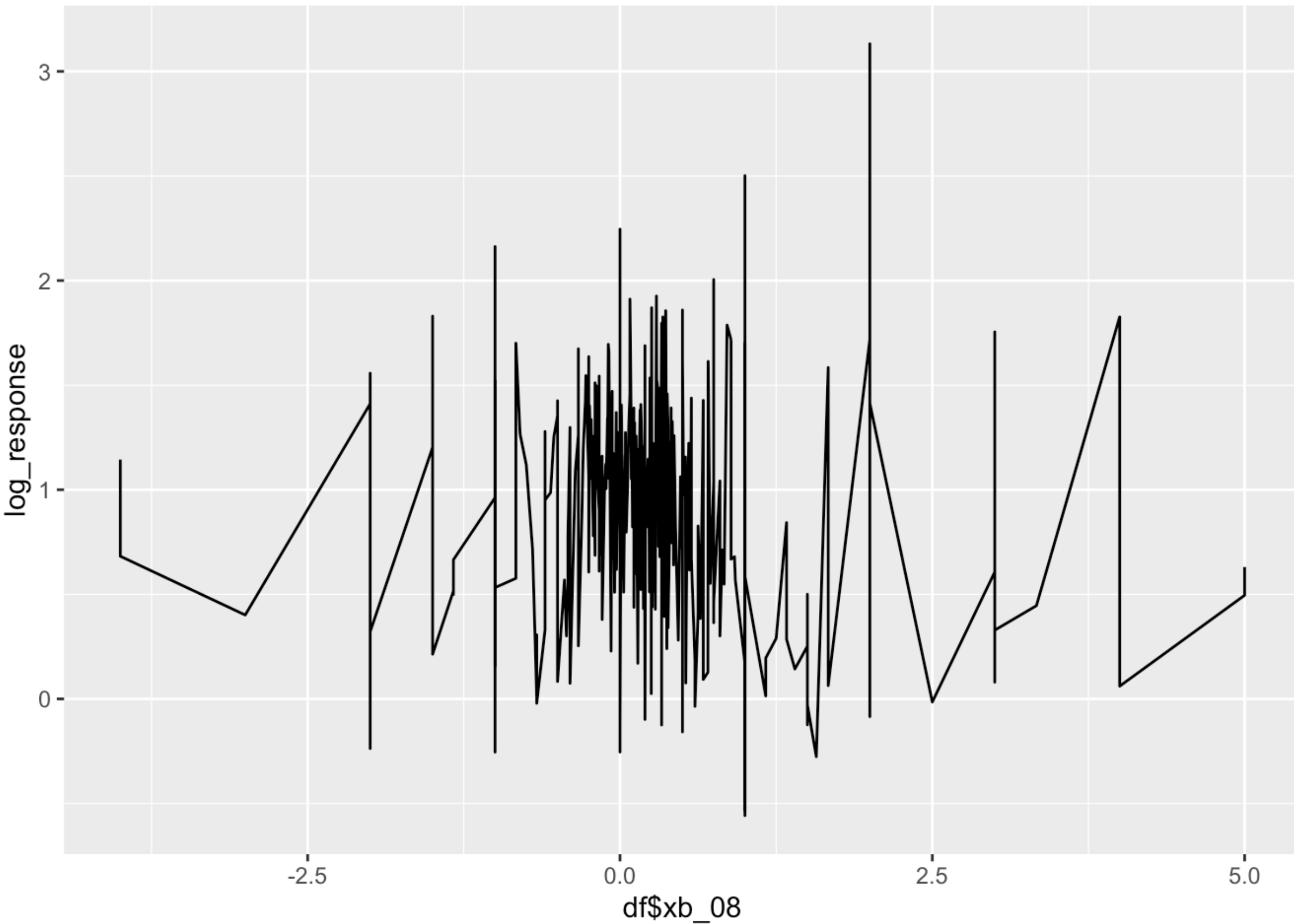

```
df %>%
  ggplot(mapping = aes(df$xb_06, log_response)) + geom_line()
```



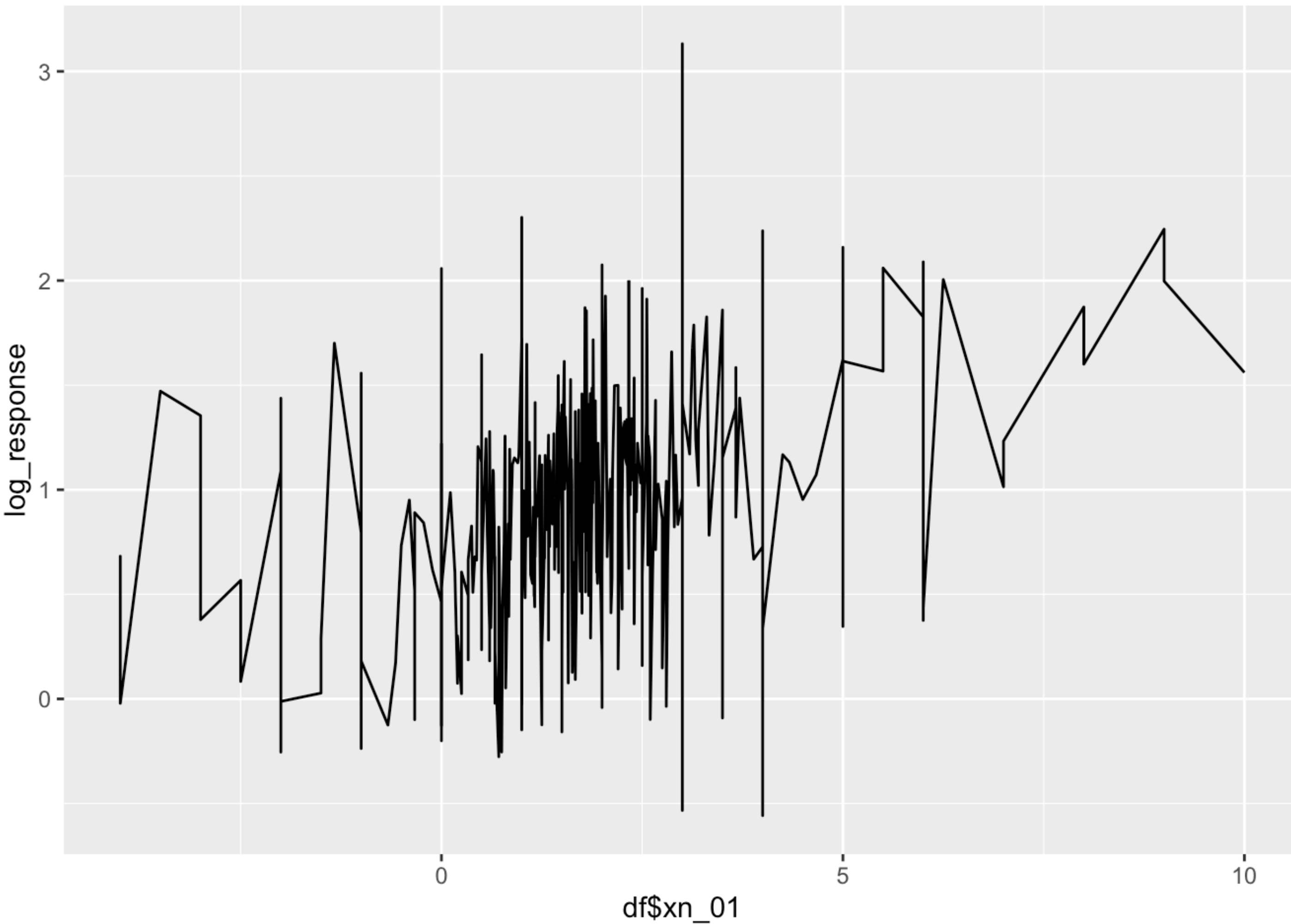

```
df %>%
  ggplot(mapping = aes(df$xb_07, log_response)) + geom_line()
```



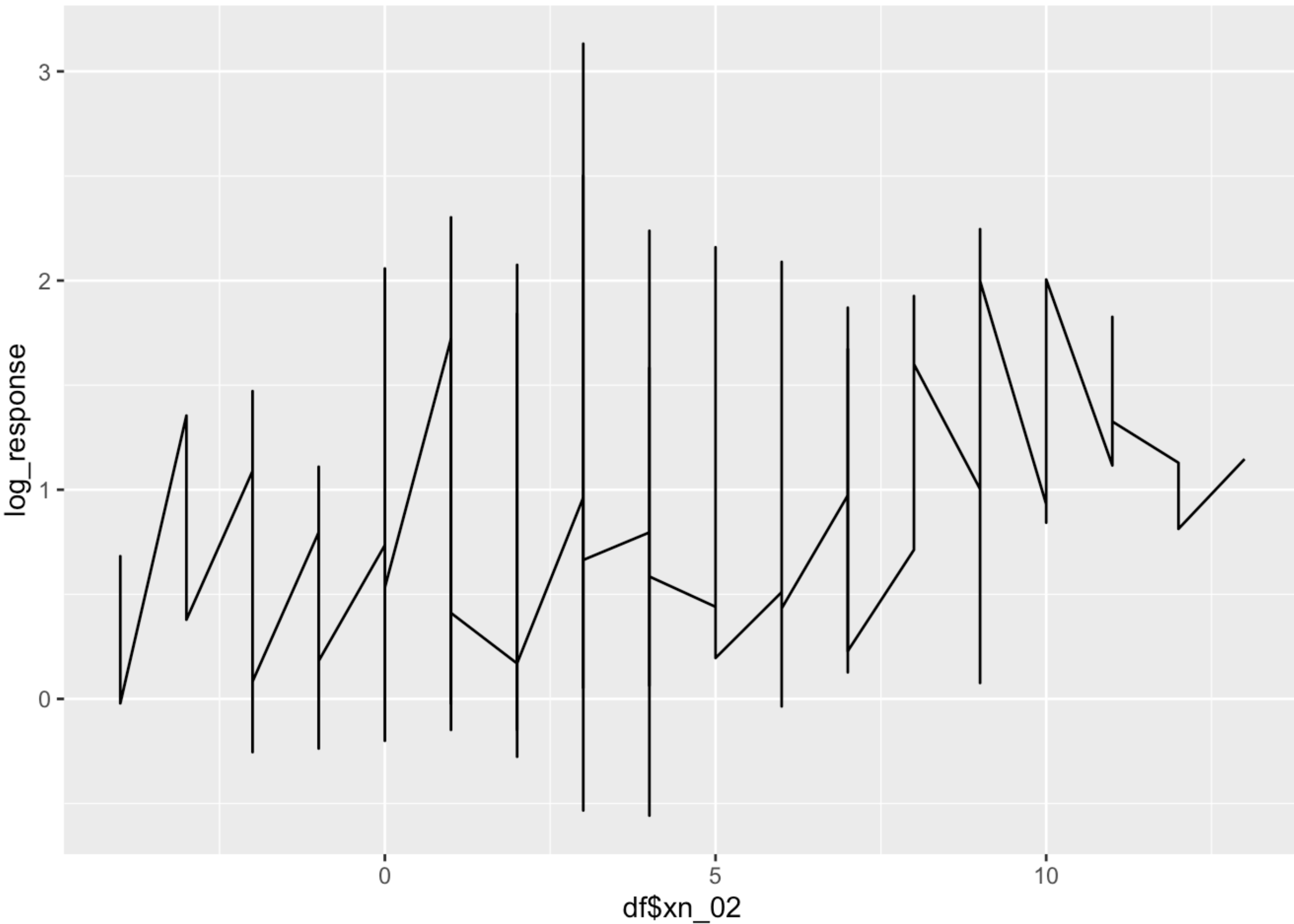

```
df %>%
  ggplot(mapping = aes(df$xb_08, log_response)) + geom_line()
```



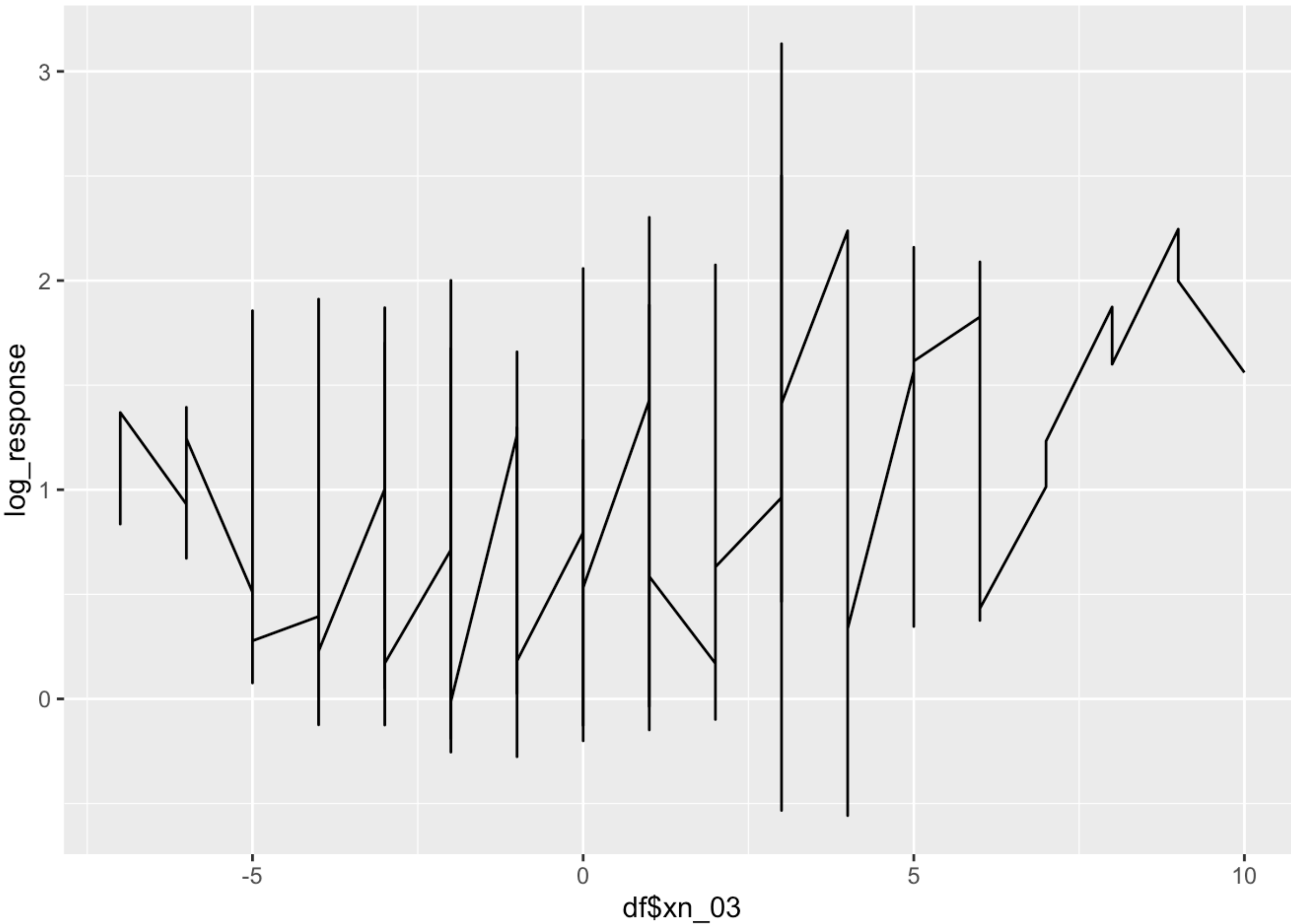

```
df %>%
  ggplot(mapping = aes(df$xn_01, log_response)) + geom_line()
```



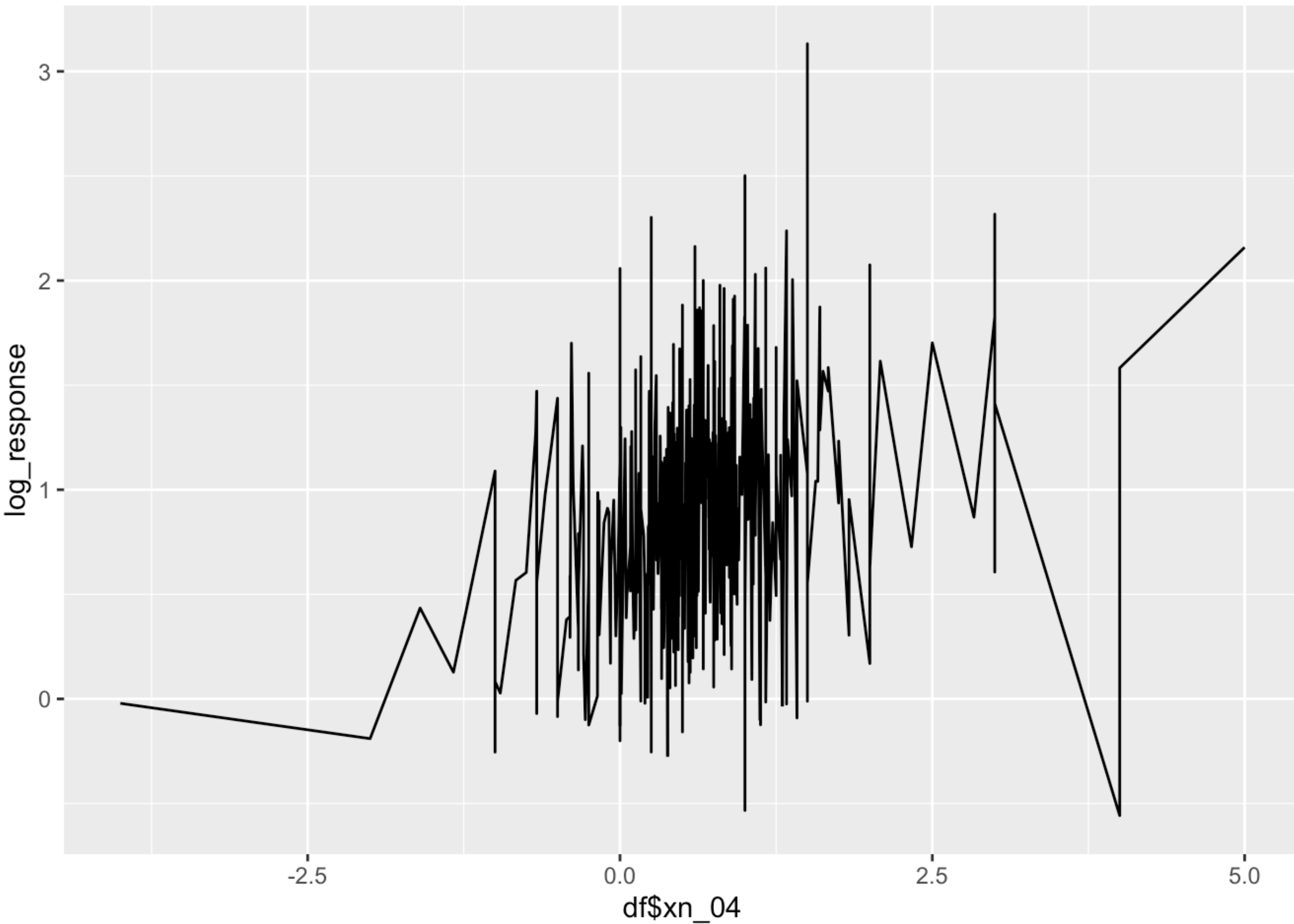

```
df %>%
  ggplot(mapping = aes(df$xn_02, log_response)) + geom_line()
```



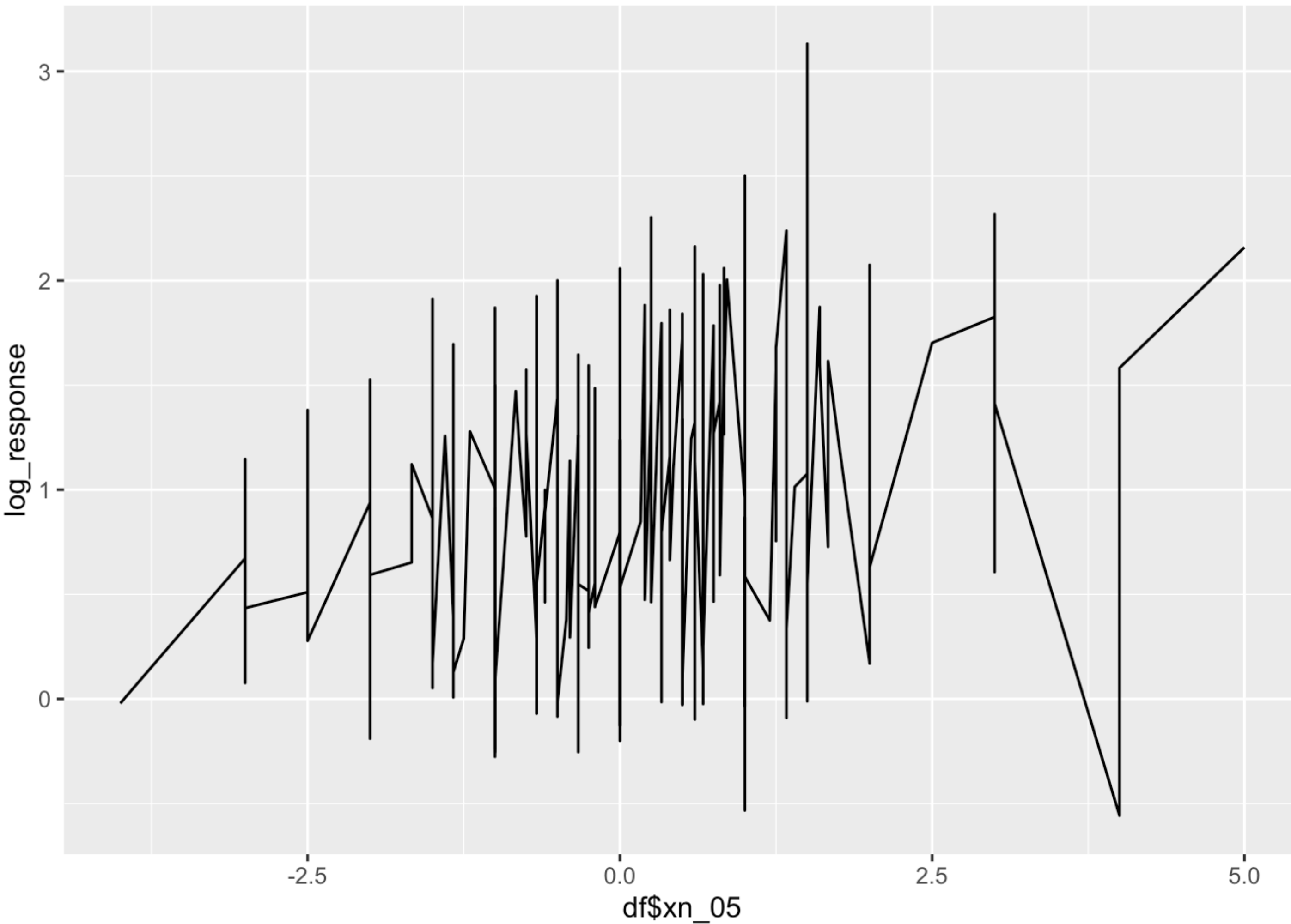

```
df %>%
  ggplot(mapping = aes(df$xn_03, log_response)) + geom_line()
```



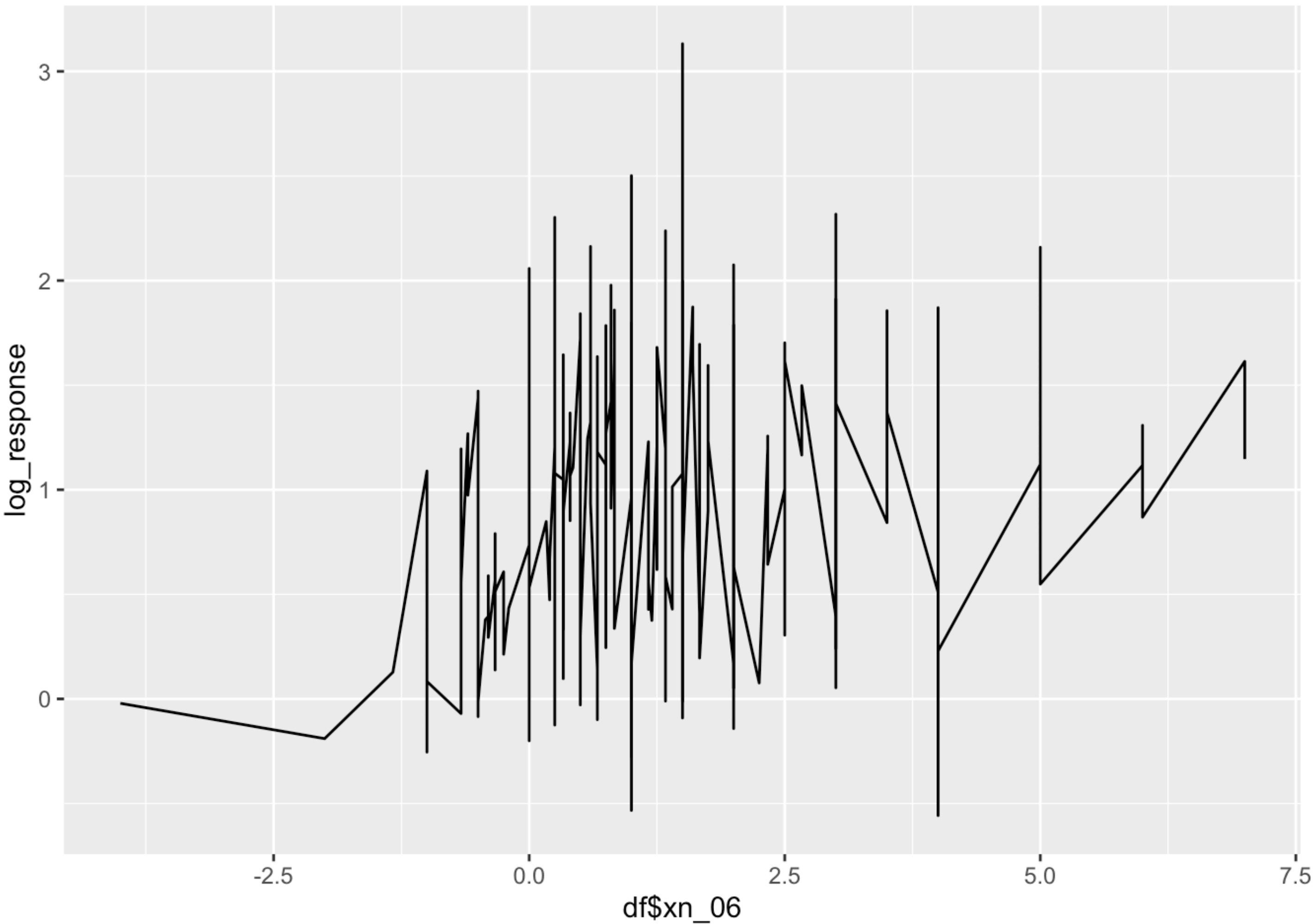

```
df %>%
  ggplot(mapping = aes(df$xn_04, log_response)) + geom_line()
```



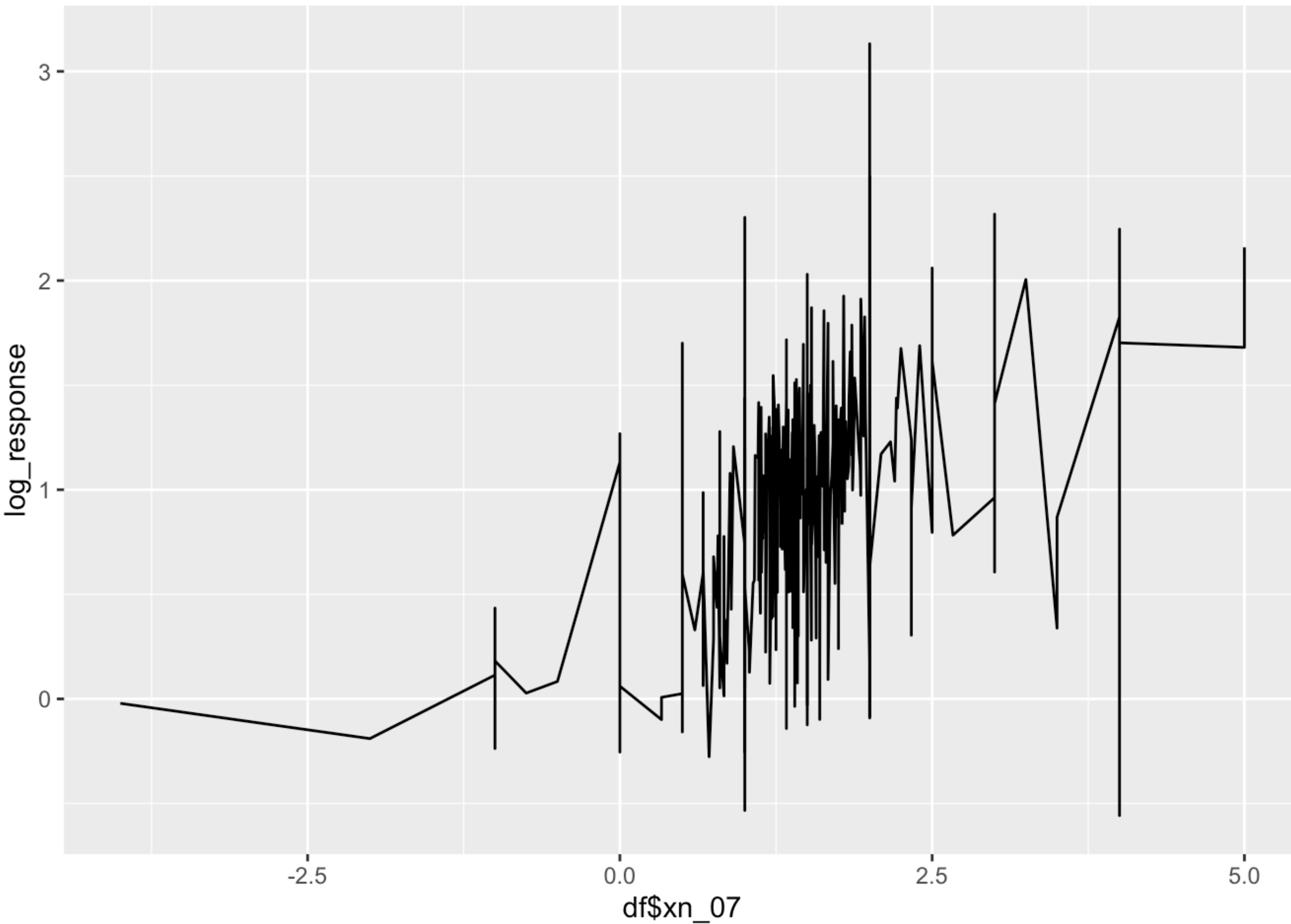

```
df %>%
  ggplot(mapping = aes(df$xn_05, log_response)) + geom_line()
```



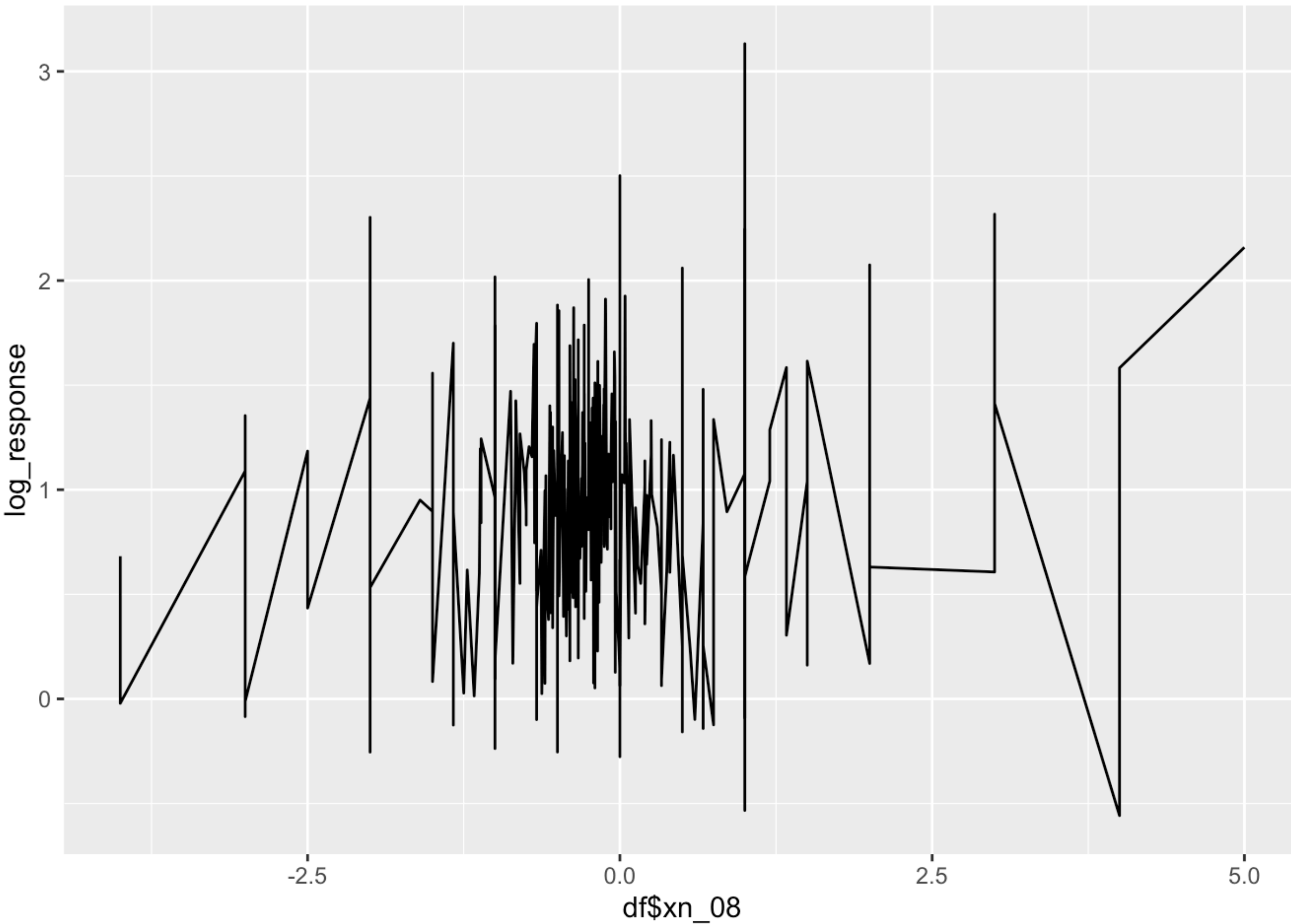

```
df %>%
  ggplot(mapping = aes(df$xn_06, log_response)) + geom_line()
```



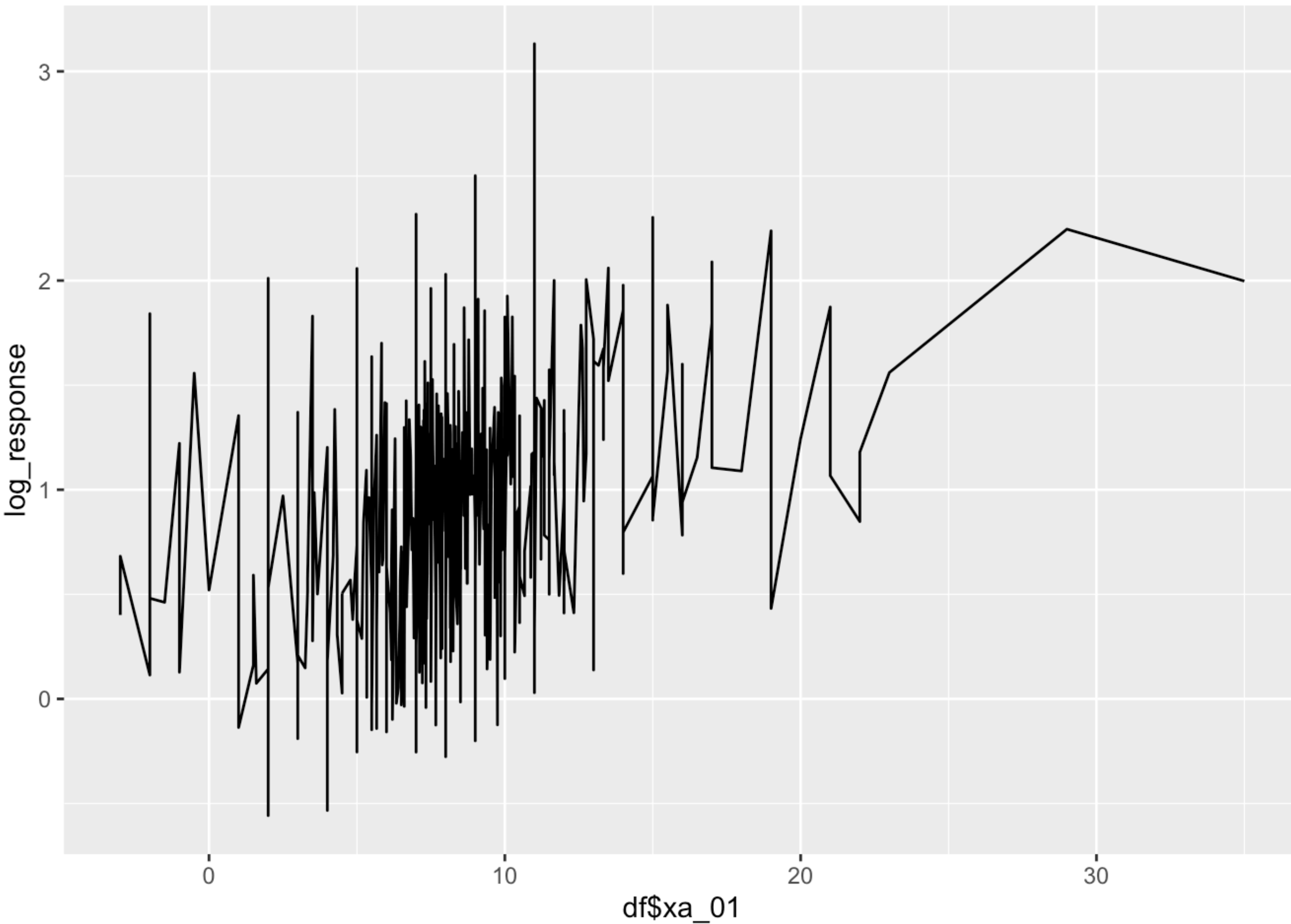

```
df %>%
  ggplot(mapping = aes(df$xn_07, log_response)) + geom_line()
```



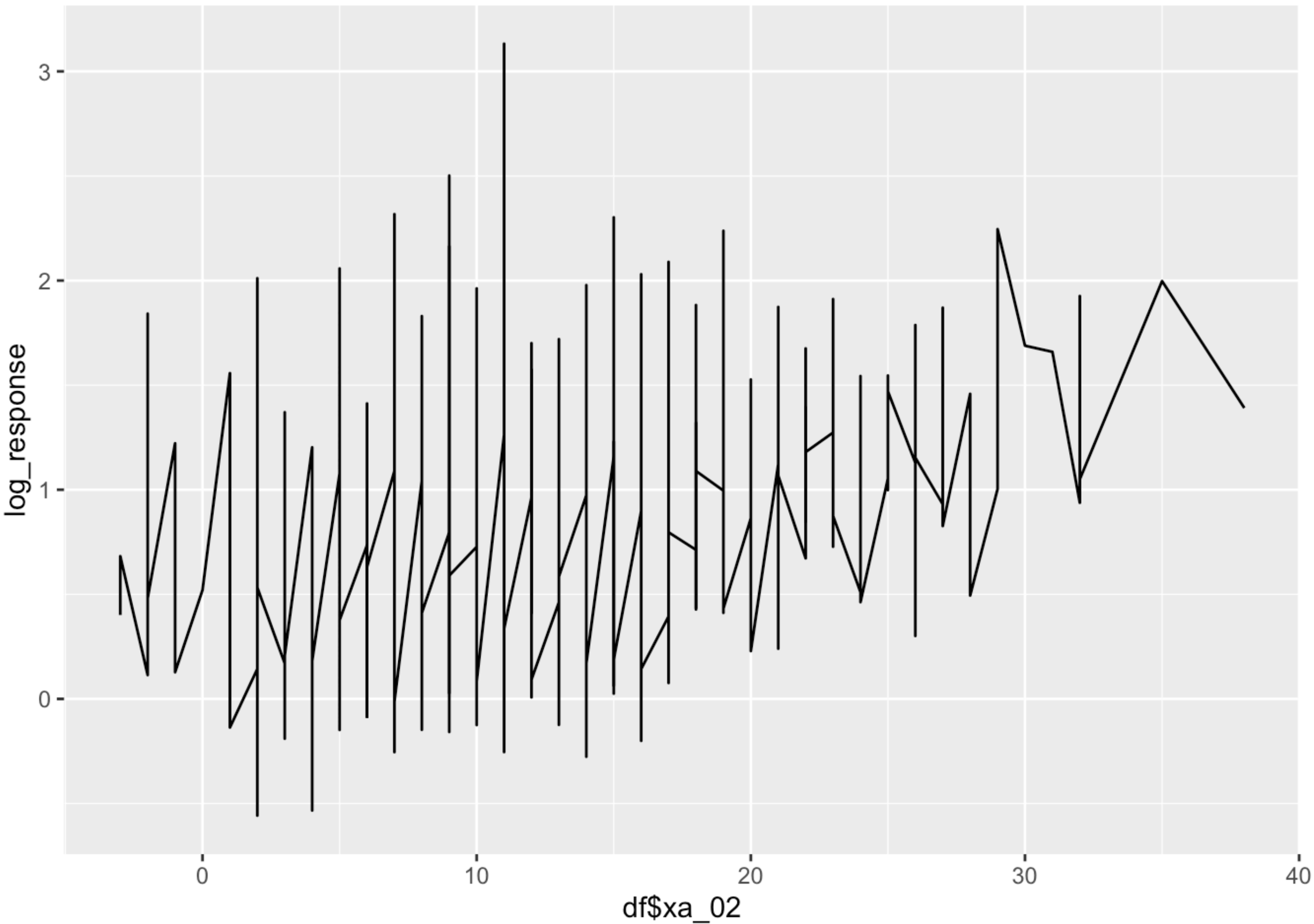

```
df %>%
  ggplot(mapping = aes(df$xn_08, log_response)) + geom_line()
```



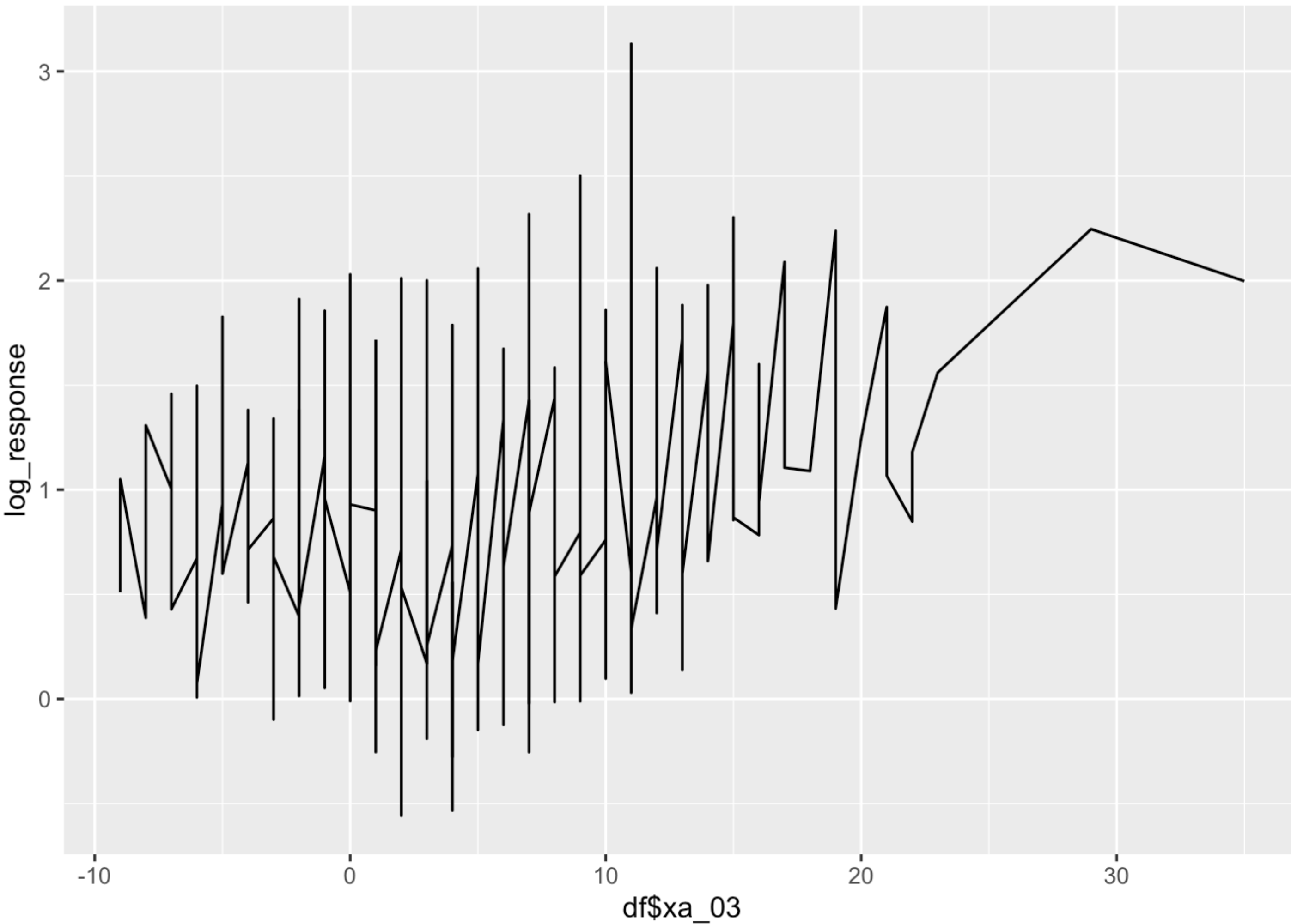

```
df %>%
  ggplot(mapping = aes(df$xa_01, log_response)) + geom_line()
```



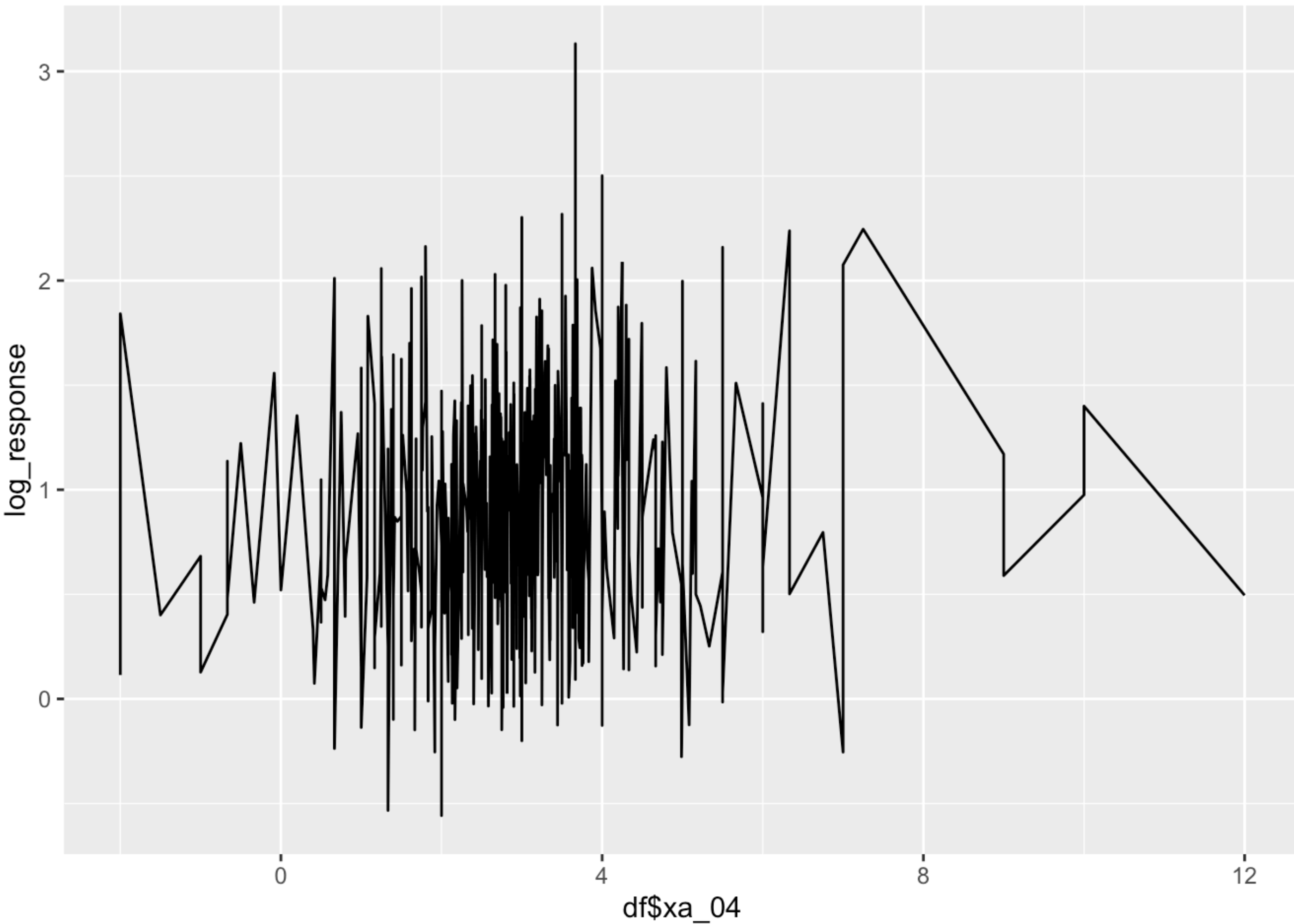

```
df %>%
  ggplot(mapping = aes(df$xa_02, log_response)) + geom_line()
```



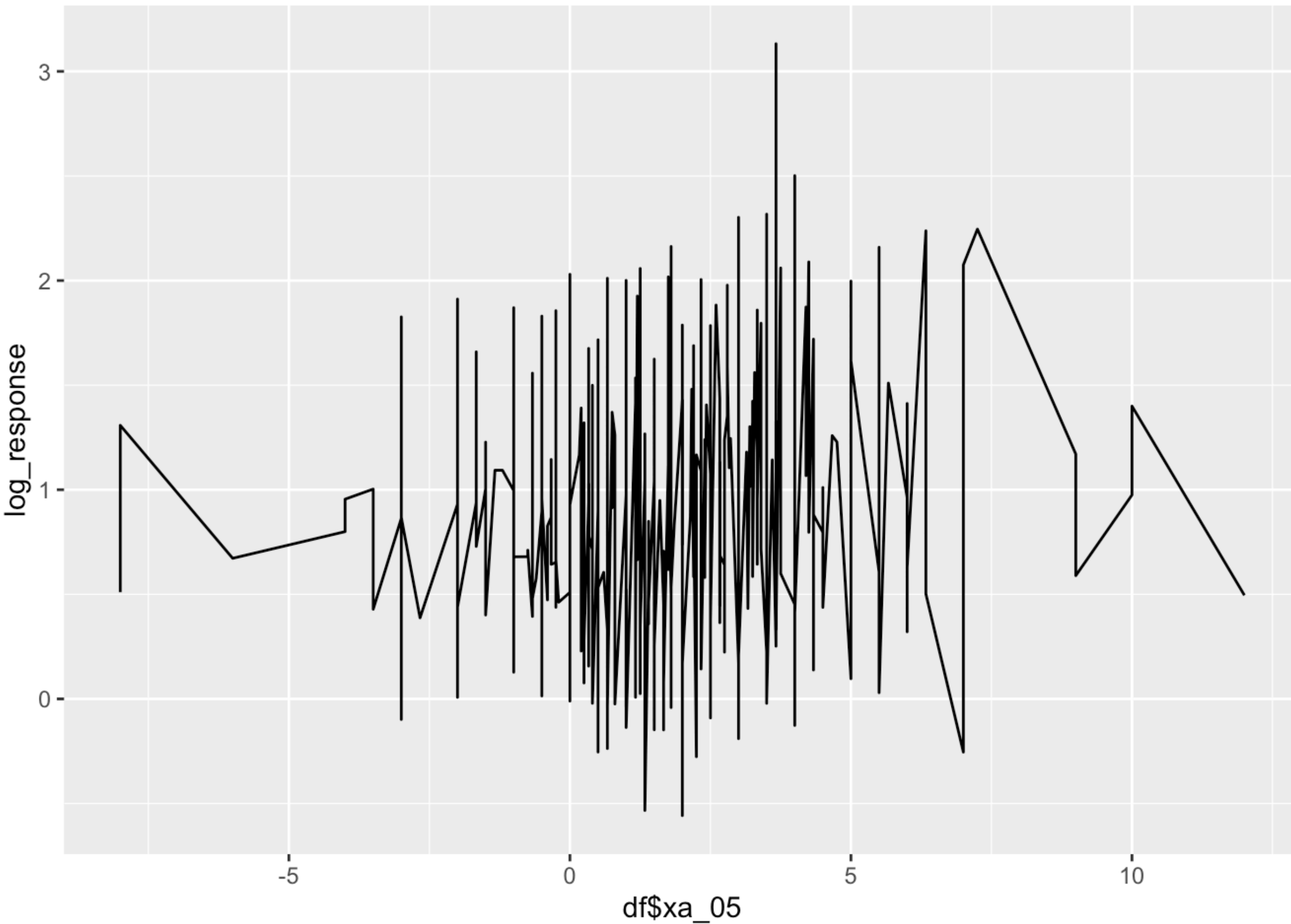

```
df %>%
  ggplot(mapping = aes(df$xa_03, log_response)) + geom_line()
```



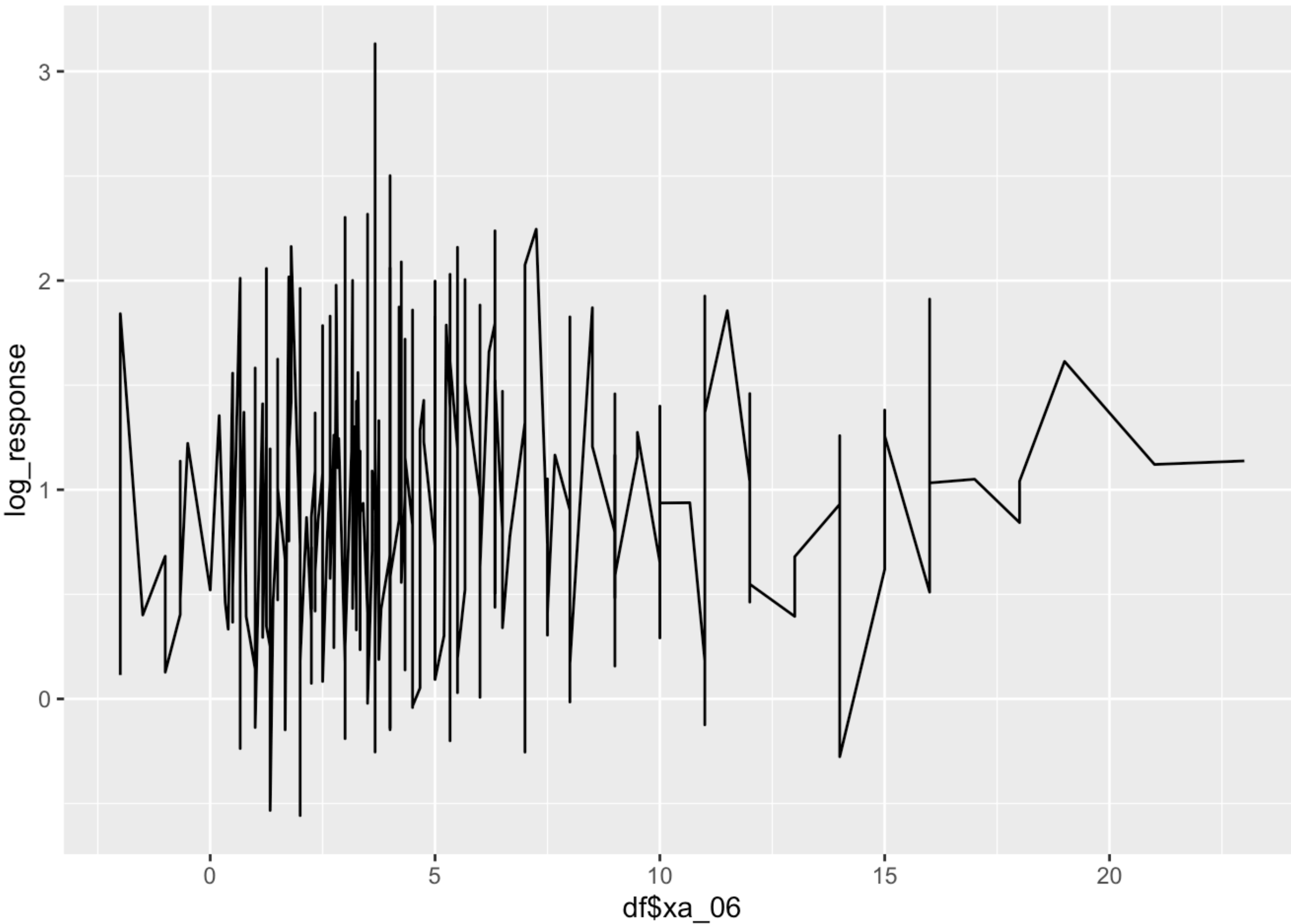

```
df %>%
  ggplot(mapping = aes(df$xa_04, log_response)) + geom_line()
```



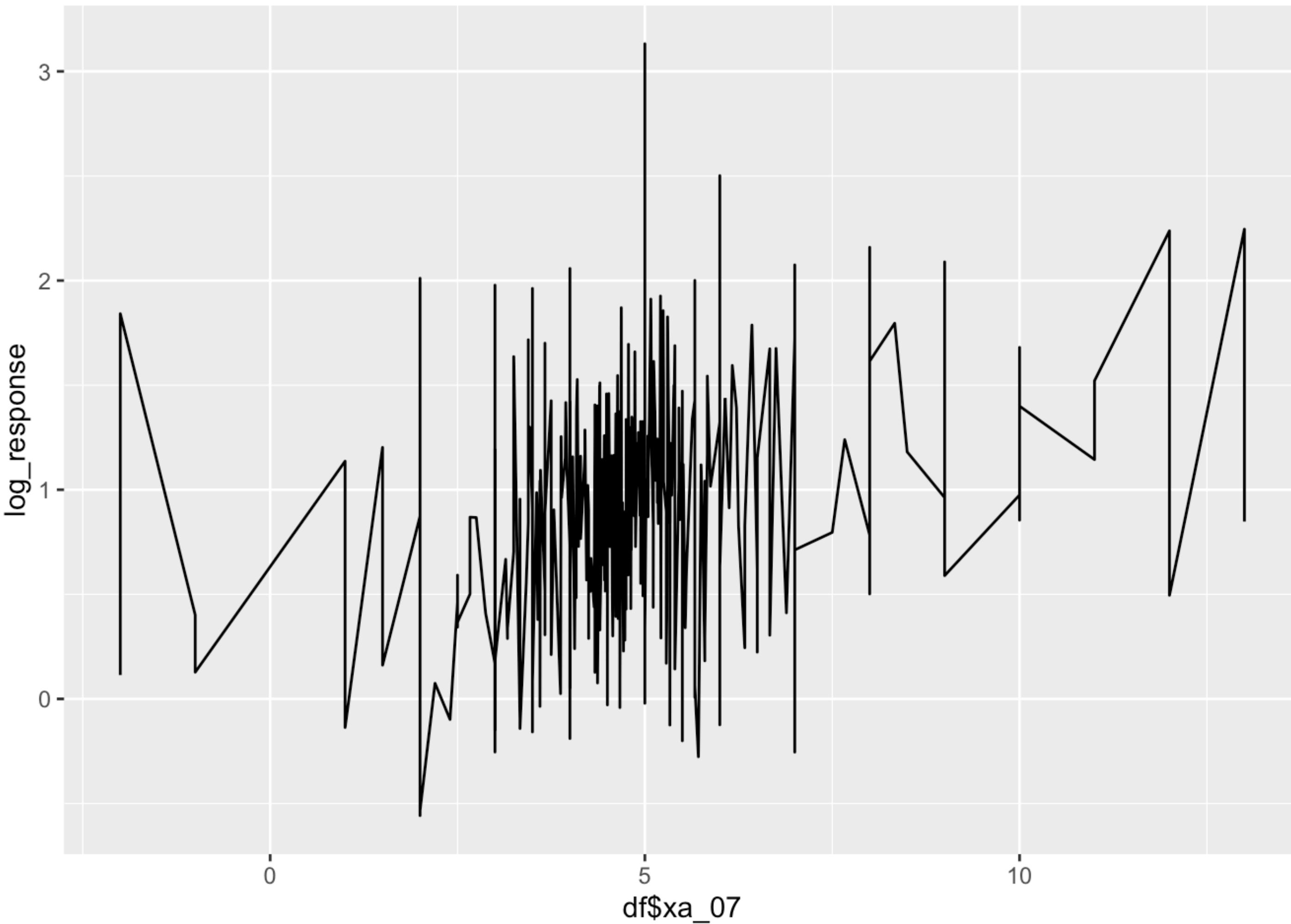

```
df %>%
  ggplot(mapping = aes(df$xa_05, log_response)) + geom_line()
```



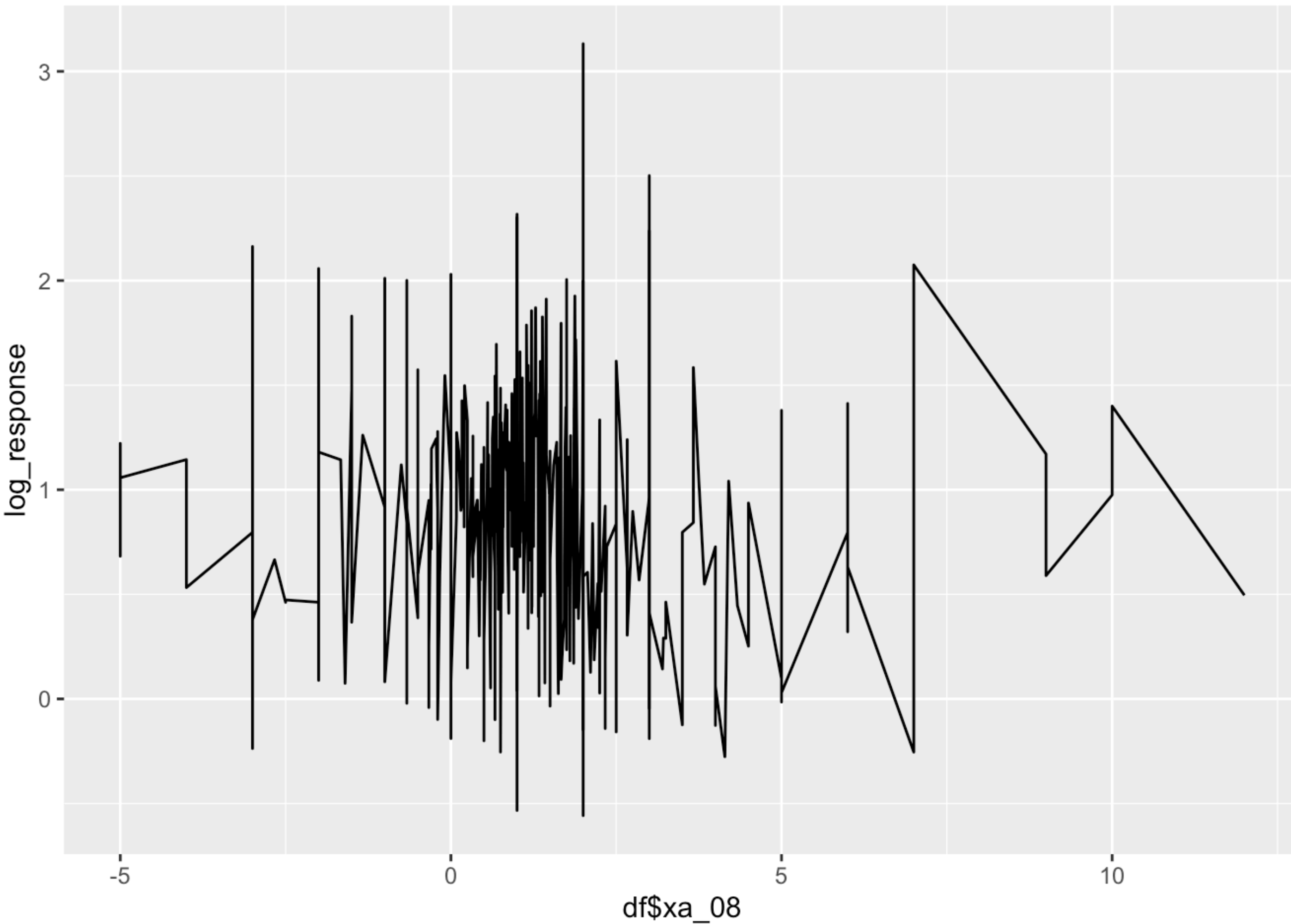

```
df %>%
  ggplot(mapping = aes(df$xa_06, log_response)) + geom_line()
```



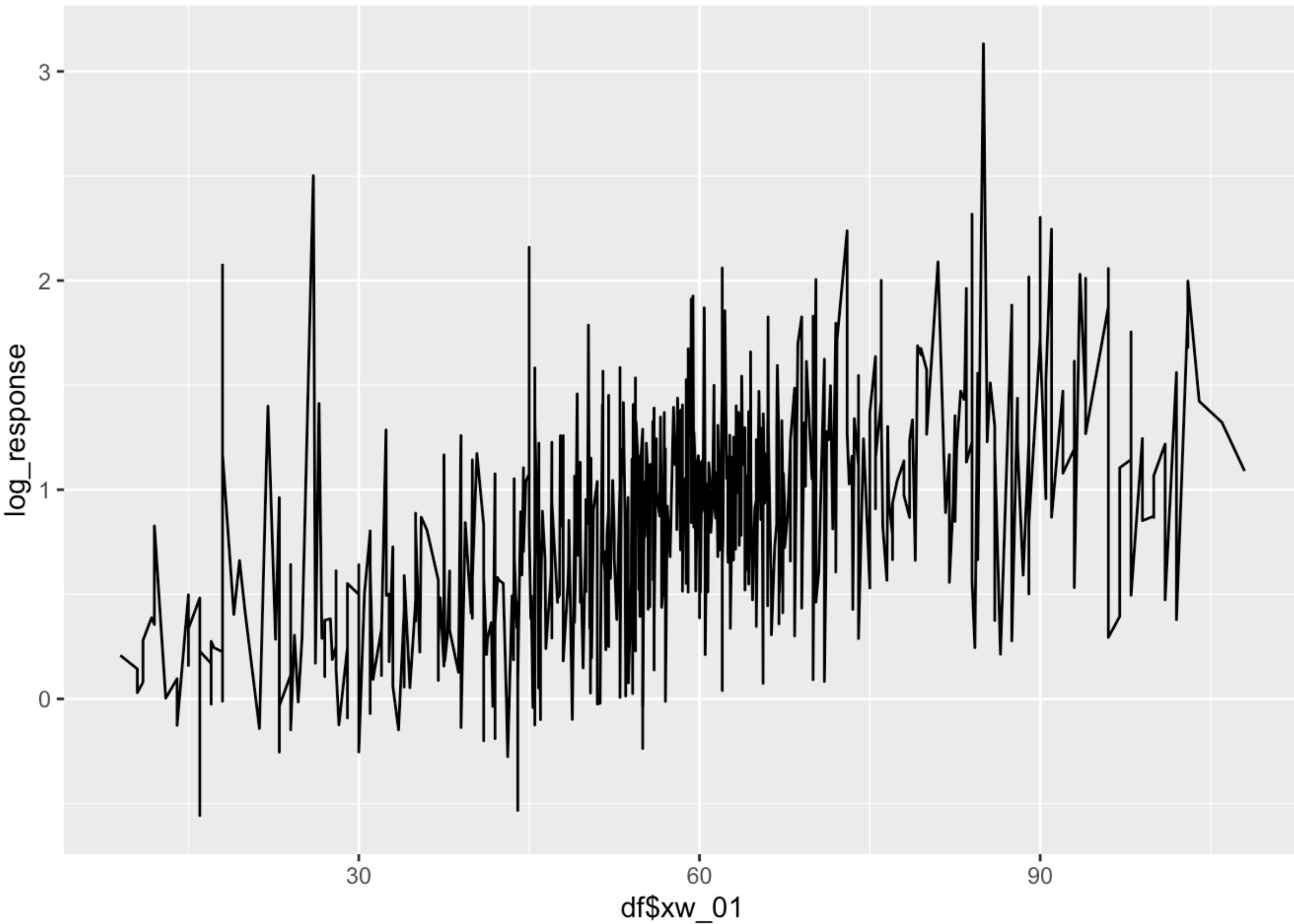

```
df %>%
  ggplot(mapping = aes(df$xa_07, log_response)) + geom_line()
```



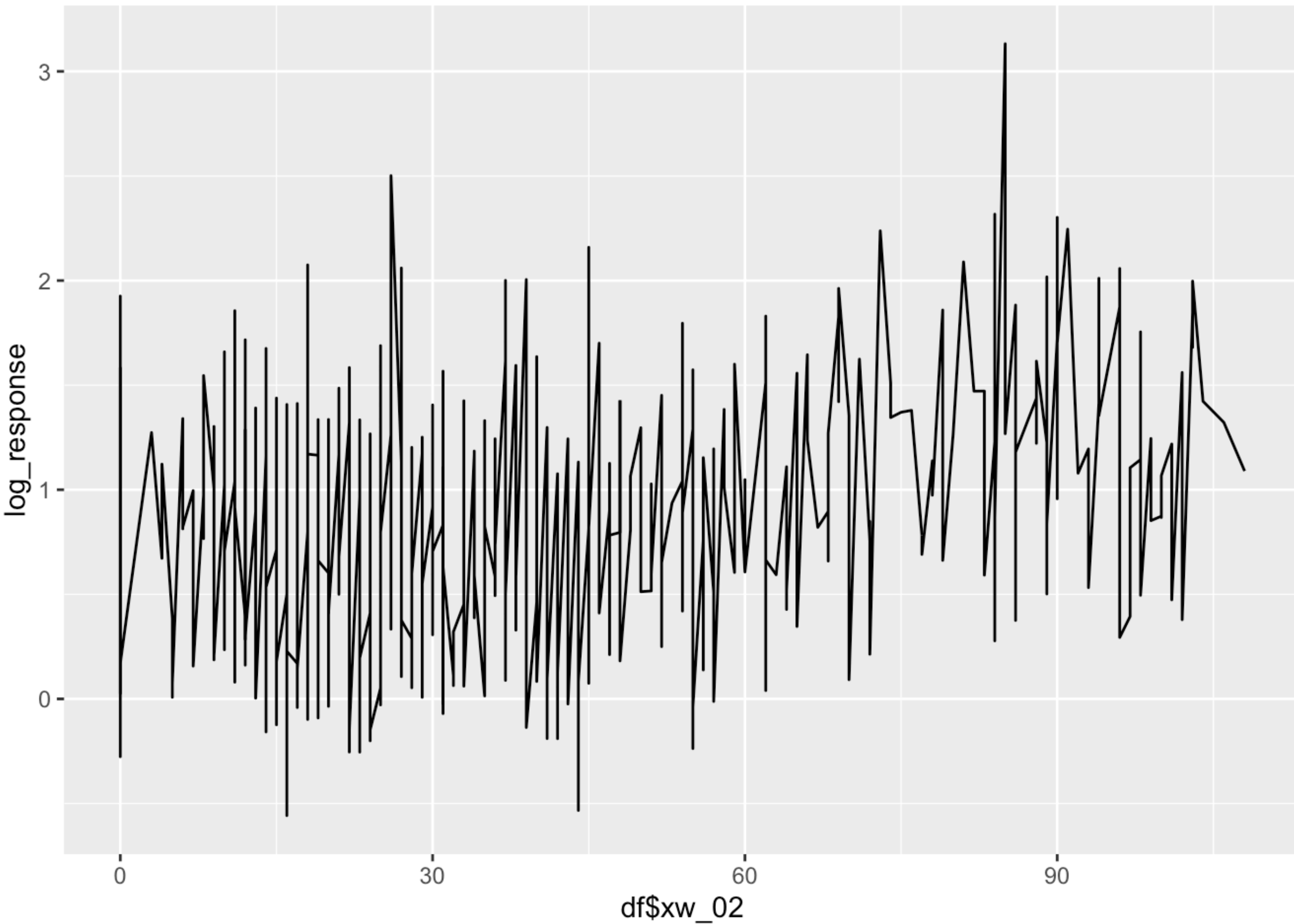

```
df %>%
  ggplot(mapping = aes(df$xa_08, log_response)) + geom_line()
```



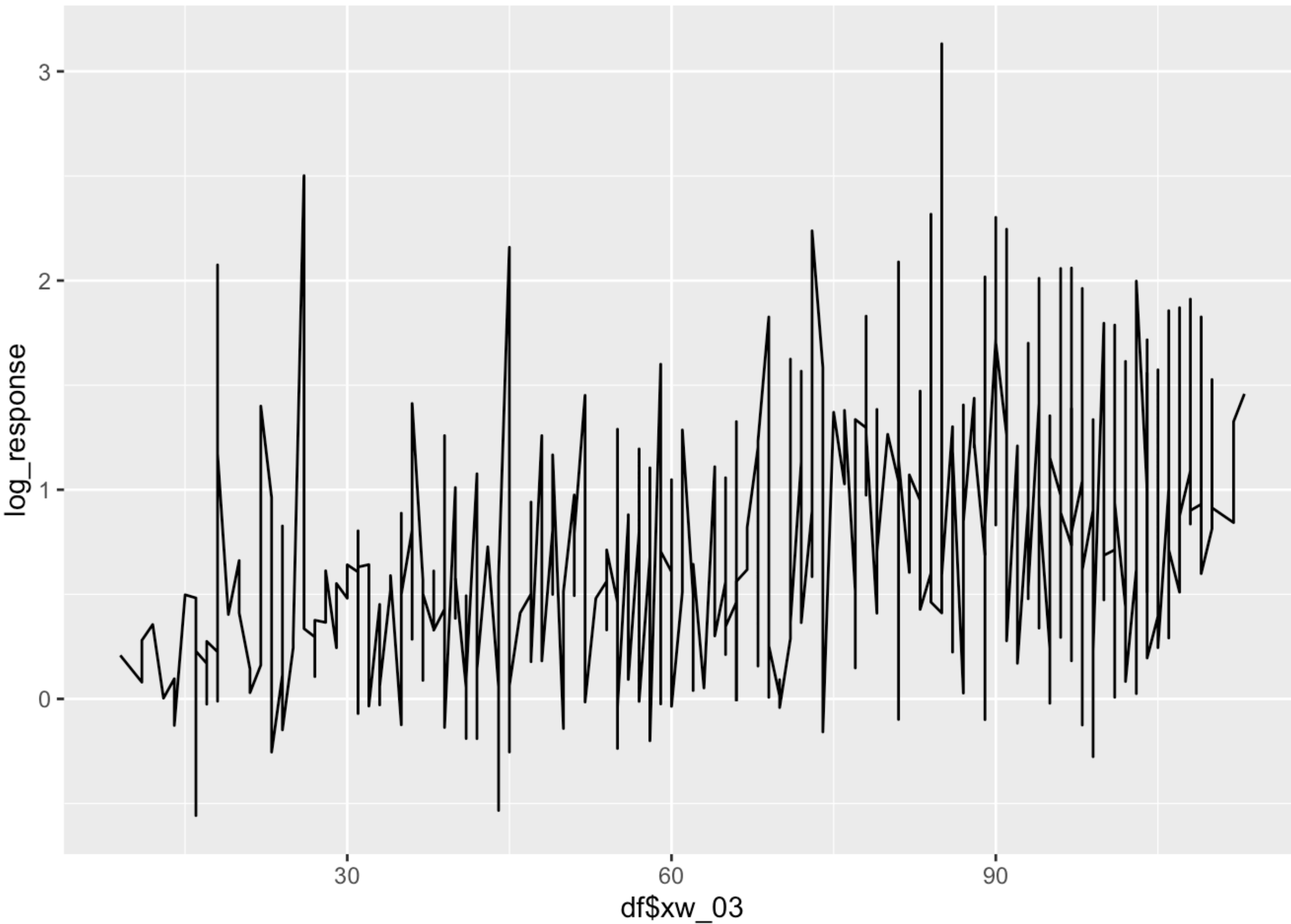

```
df %>%
  ggplot(mapping = aes(df$xw_01, log_response)) + geom_line()
```



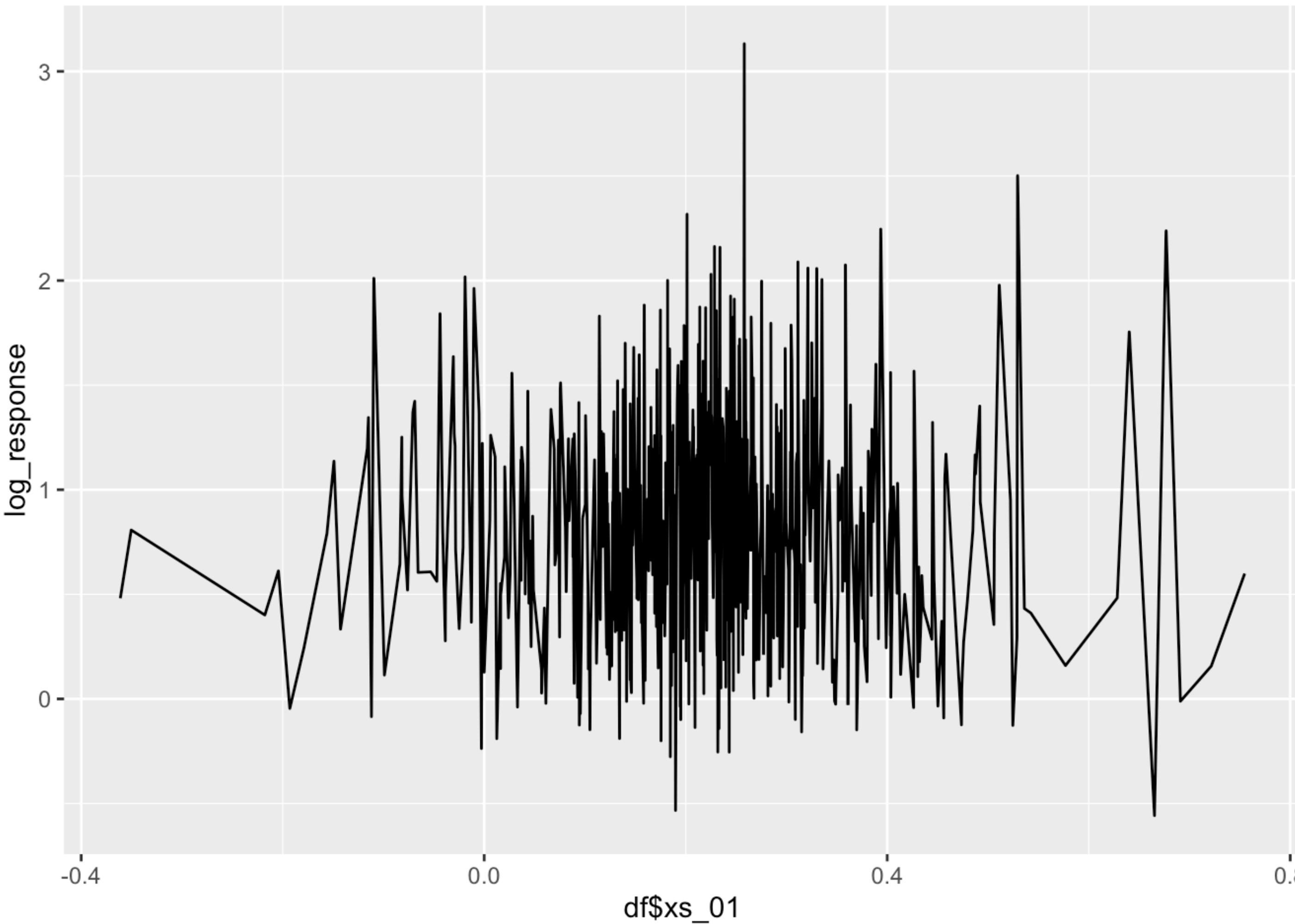

```
df %>%
  ggplot(mapping = aes(df$xw_02, log_response)) + geom_line()
```



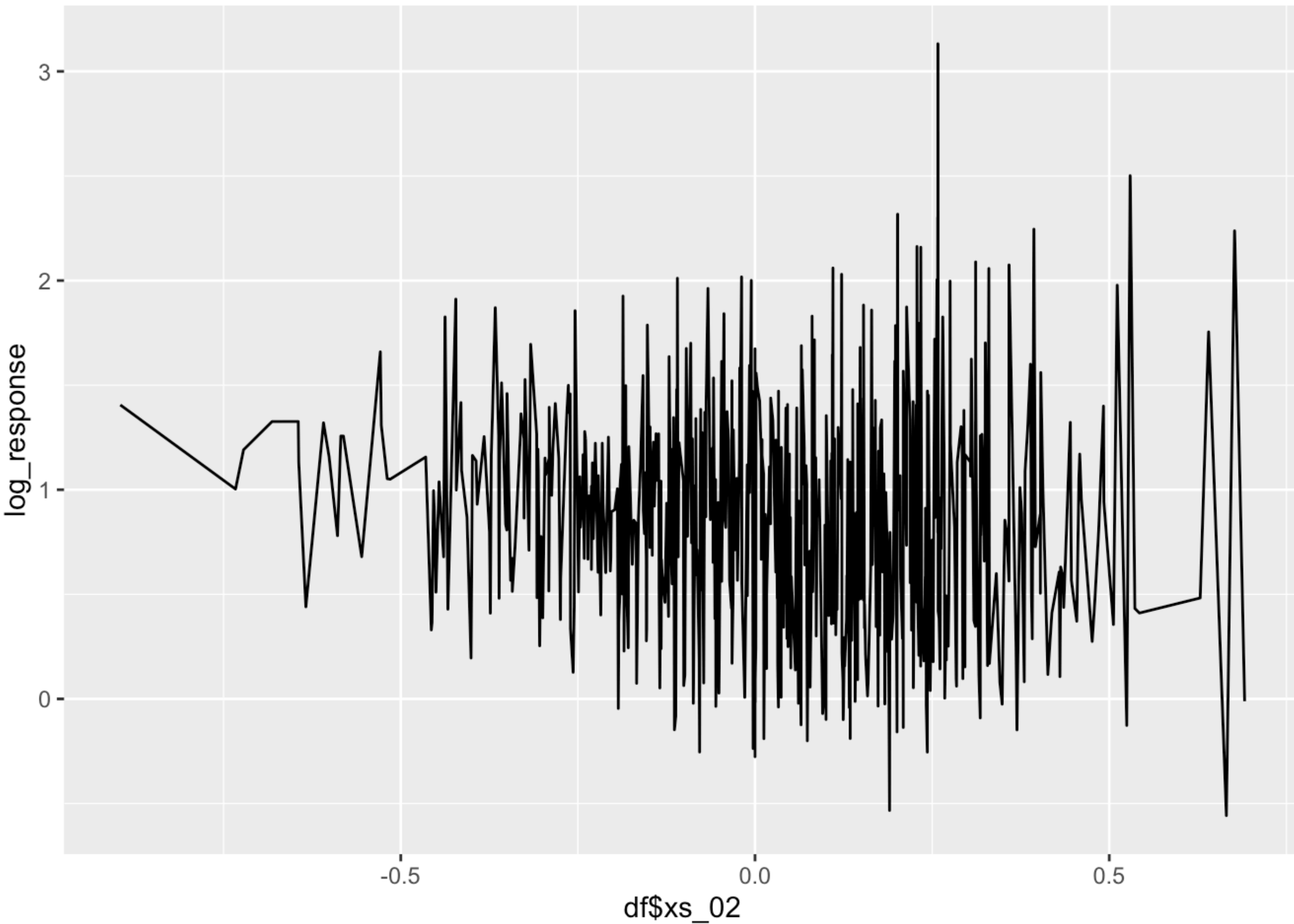

```
df %>%
  ggplot(mapping = aes(df$xw_03, log_response)) + geom_line()
```



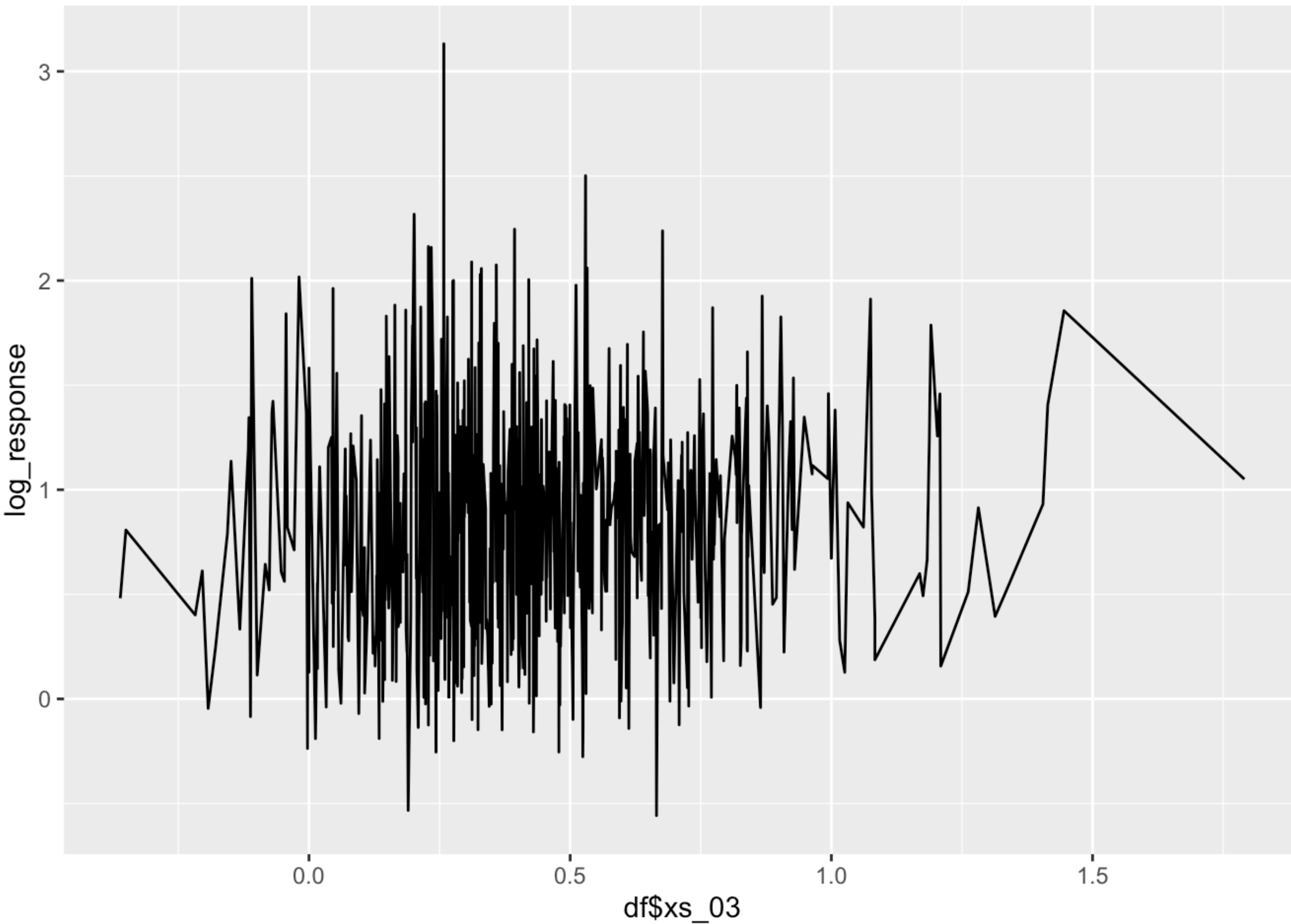

```
df %>%
  ggplot(mapping = aes(df$xs_01, log_response)) + geom_line()
```



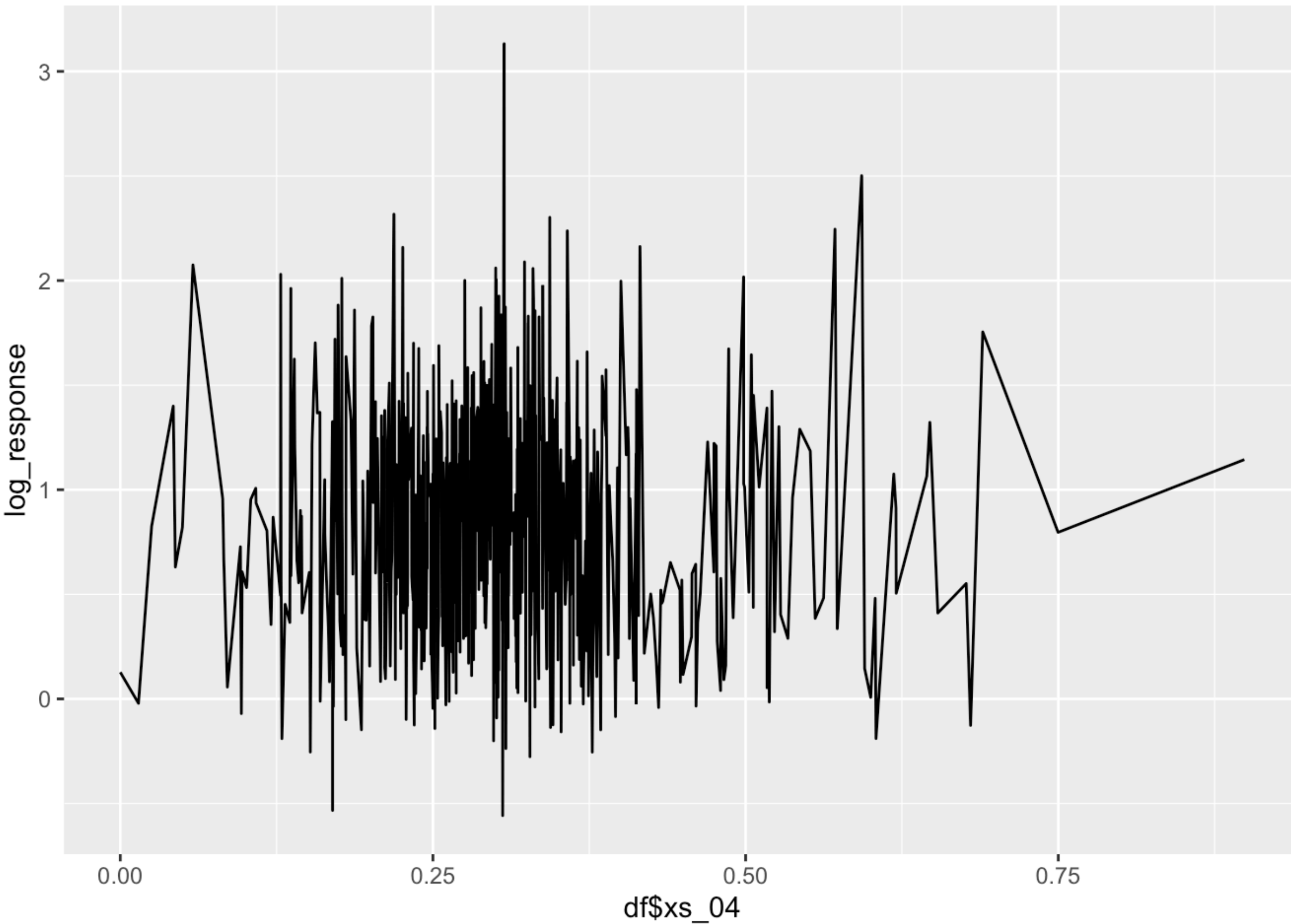

```
df %>%
  ggplot(mapping = aes(df$xs_02, log_response)) + geom_line()
```



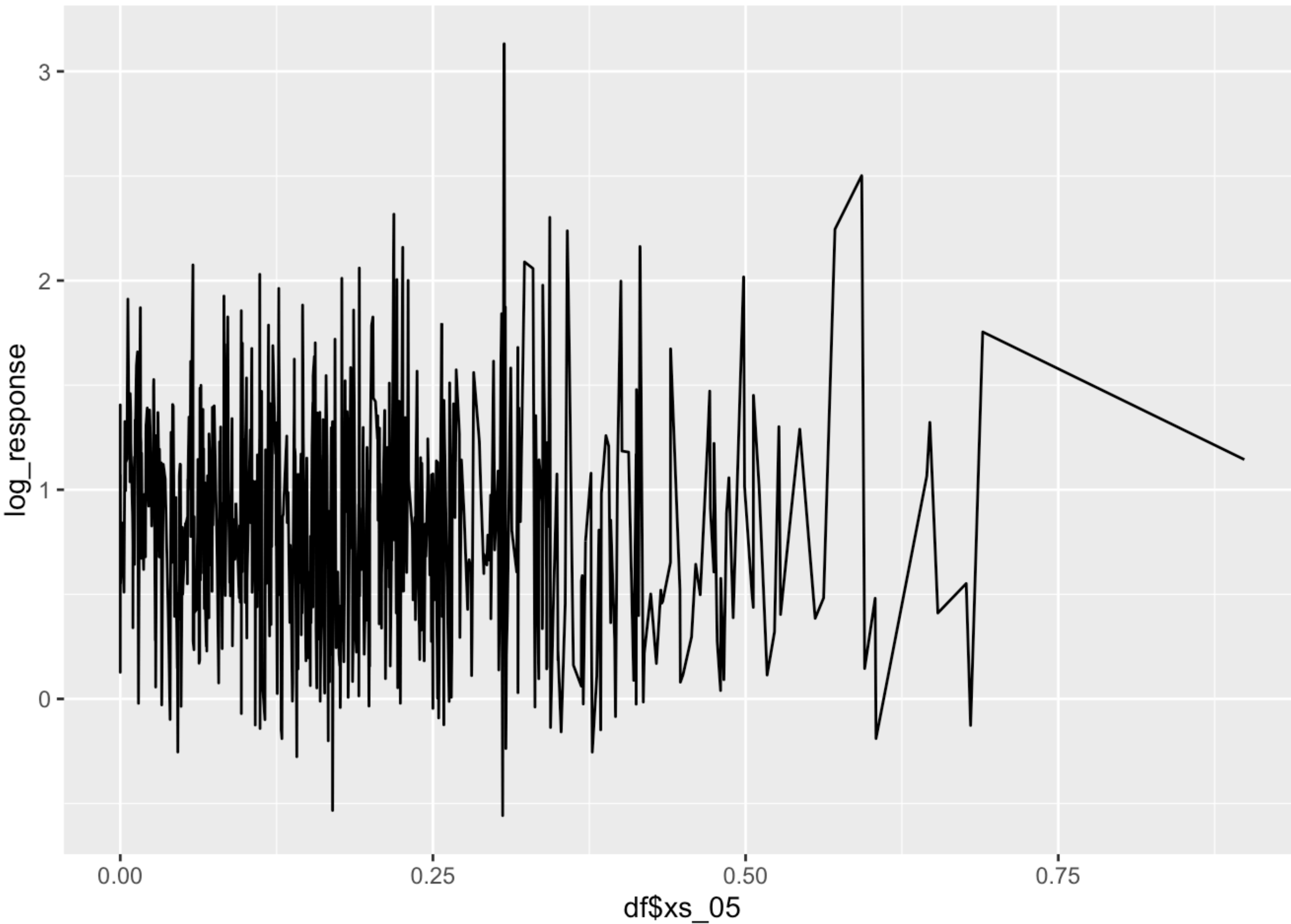

```
df %>%
  ggplot(mapping = aes(df$xs_03, log_response)) + geom_line()
```



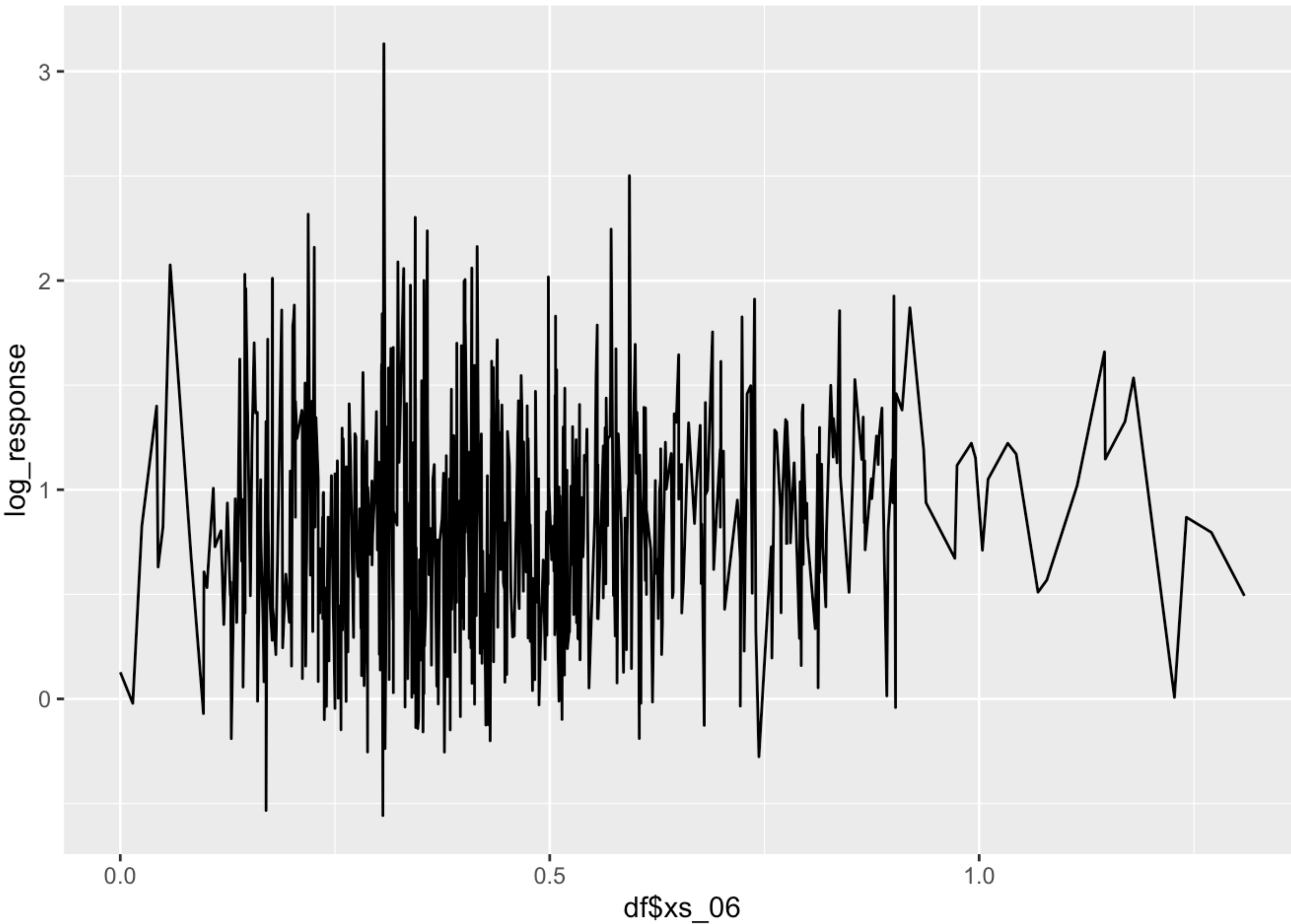

```
df %>%
  ggplot(mapping = aes(df$xs_04, log_response)) + geom_line()
```




```
df %>%
  ggplot(mapping = aes(df$xs_05, log_response)) + geom_line()
```




```
df %>%
  ggplot(mapping = aes(df$xs_06, log_response)) + geom_line()
```



Part 2: Regression - Part A

Categorical Variables Only - Linear Additive

```
mod_categorical <- lm(log_response ~ region + customer + outcome , data = df)
```

Continous Variables Only - Linear Additive

```
mod_continous <- lm(log_response ~ xb_01 + xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + xa
_01 + xa_02 + xa_03 +xb_04 + xb_05
                    + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 +
xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df)
```

All Categorical + Continous inputs - Linear Additive

```
mod_cat_cont <- lm(log_response ~ region + customer + outcome +  xb_01 + xb_02 + xb_0
3 + xn_01 + xn_02 + xn_03 + xa_01 + xa_02 + xa_03 +xb_04 + xb_05 + xb_06 + xb_07 + xb
_08 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df)
```

Interacting Categorical Input ‘Region’ with all continious inputs

```
mod_interact_region <- lm(log_response ~ region*xb_01 + region*xb_02 + region*xb_03 +
region*xn_01 + region*xn_02 + region*xn_03 +
                      region*xa_01 + region*xa_02 + region*xa_03 +region*xb_04 + regi
on*xb_05+
                      region*xb_06 + region*xb_07 + region*xb_08 + region*xn_04 + reg
ion*xn_05 + region*xn_06 + region*xn_07 +
                      region*xn_08 + region*xa_04 + region*xa_05 + region*xa_06 + reg
ion*xa_07 + region*xa_08 +
                      region*xw_01 + region*xw_02 + region*xw_03 + region*xs_01 + reg
ion*xs_02 + region*xs_03 + region*xs_04 +
                      region*xs_05 + region*xs_06, data = df)
```

Interacting Categorical Input ‘Customer’ with all continuous inputs

```
mod_interact_customer <- lm(log_response ~ customer*xb_01 + customer*xb_02 + customer
*xb_03 + customer*xn_01 + customer*xn_02 + customer*xn_03 +
                      customer*xa_01 + customer*xa_02 + customer*xa_03 +customer*xb_0
4 + customer*xb_05+
                      customer*xb_06 + customer*xb_07 + customer*xb_08 + customer*xn_
04 + customer*xn_05 + customer*xn_06 + customer*xn_07 +
                      customer*xn_08 + customer*xa_04 + customer*xa_05 + customer*xa_
06 + customer*xa_07 + customer*xa_08 +
                      customer*xw_01 + customer*xw_02 + customer*xw_03 + customer*xs_
01 + customer*xs_02 + customer*xs_03 + customer*xs_04 +
                      customer*xs_05 + customer*xs_06, data = df)
```

Examining all pairwise interactions terms of the continuous inputs

```
df_continuous <- df[continuous_vars]
df_continuous$log_response <- df$log_response

mod_pairwise_cont <- lm (log_response ~ (.)^2, data = df_continuous)
```

Fitting three other basis functions

Fitting a spline model with continious xb_07 of 15 df

```
mod_lin_quad <- lm(log_response ~ splines::ns(xb_07, 15), data = df_continuous )
```

The input xb_07 seemed be to statistically significant in the previous linear models.

Fitting linear model with continuous inputs and their qudratic features.

```
mod_quadratic <- lm(log_response ~  xb_01 + xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + x
a_01 + xa_02 + xa_03 +xb_04 + xb_05 + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 +
xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 + xw_01 + xw_02 + xw_03 + xs_01
+ xs_02 + xs_03 + xs_04 + xs_05 + xs_06 + I(xb_01^2) + I(xb_02^2) + I(xb_03^2) + I(xn
_01^2) + I(xn_02^2) + I(xn_03^2) + I(xa_01^2) + I(xa_02^2) + I(xa_03^2) + I(xb_04^2)
+ I(xb_05^2) + I(xb_06^2) + I(xb_07^2) + I(xb_08^2) + I(xn_04^2) + I(xn_05^2) + I(xn_
06^2) + I(xn_07^2) + I(xn_08^2) + I(xa_04^2) + I(xa_05^2) + I(xa_06^2) + I(xa_07^2) +
I(xa_08^2) + I(xw_01^2) + I(xw_02^2) + I(xw_03^2) + I(xs_01^2) + I(xs_02^2) + I(xs_03
^2) + I(xs_04^2) + I(xs_05^2) + I(xs_06), data = df_continuous)

mod_quadratic %>% summary()
```

```
##
## Call:
## lm(formula = log_response ~ xb_01 + xb_02 + xb_03 + xn_01 + xn_02 +
##     xn_03 + xa_01 + xa_02 + xa_03 + xb_04 + xb_05 + xb_06 + xb_07 +
##     xb_08 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 + xa_04 + xa_05 +
##     xa_06 + xa_07 + xa_08 + xw_01 + xw_02 + xw_03 + xs_01 + xs_02 +
##     xs_03 + xs_04 + xs_05 + xs_06 + I(xb_01^2) + I(xb_02^2) +
##     I(xb_03^2) + I(xn_01^2) + I(xn_02^2) + I(xn_03^2) + I(xa_01^2) +
##     I(xa_02^2) + I(xa_03^2) + I(xb_04^2) + I(xb_05^2) + I(xb_06^2) +
##     I(xb_07^2) + I(xb_08^2) + I(xn_04^2) + I(xn_05^2) + I(xn_06^2) +
```

```
##     I(xn_07^2) + I(xn_08^2) + I(xa_04^2) + I(xa_05^2) + I(xa_06^2) +
##     I(xa_07^2) + I(xa_08^2) + I(xw_01^2) + I(xw_02^2) + I(xw_03^2) +
##     I(xs_01^2) + I(xs_02^2) + I(xs_03^2) + I(xs_04^2) + I(xs_05^2) +
##     I(xs_06), data = df_continuous)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.42290 -0.21820 -0.00645  0.18734  1.78412
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.248e-02  1.611e-01  -0.326  0.74465
## xb_01        9.331e-02  7.177e-02   1.300  0.19402
## xb_02       -3.253e-02  4.844e-02  -0.672  0.50206
## xb_03        7.670e-03  2.561e-02   0.299  0.76468
## xn_01        4.245e-02  5.900e-02   0.720  0.47209
## xn_02        2.785e-02  4.193e-02   0.664  0.50685
## xn_03       -3.795e-02  2.903e-02  -1.307  0.19158
## xa_01       -3.482e-02  3.687e-02  -0.944  0.34529
## xa_02        8.431e-03  2.331e-02   0.362  0.71773
## xa_03        1.598e-02  1.542e-02   1.036  0.30074
## xb_04       -6.303e-01  2.005e-01  -3.144  0.00175 **
## xb_05        6.367e-02  6.447e-02   0.988  0.32371
## xb_06        6.292e-02  8.704e-02   0.723  0.47002
## xb_07        1.304e-01  9.451e-02   1.380  0.16814
## xb_08        1.123e-01  5.421e-02   2.071  0.03873 *
## xn_04        8.431e-02  1.651e-01   0.511  0.60981
## xn_05        1.601e-01  6.771e-02   2.364  0.01837 *
## xn_06        2.027e-02  8.793e-02   0.231  0.81778
## xn_07        5.181e-02  8.202e-02   0.632  0.52781
## xn_08       -1.392e-01  5.465e-02  -2.547  0.01110 *
## xa_04        2.164e-02  9.885e-02   0.219  0.82675
## xa_05        8.442e-03  3.104e-02   0.272  0.78573
## xa_06       -1.946e-02  3.971e-02  -0.490  0.62431
## xa_07        8.626e-02  5.786e-02   1.491  0.13655
## xa_08       -1.300e-02  2.935e-02  -0.443  0.65807
## xw_01        2.341e-02  9.282e-03   2.522  0.01191 *
## xw_02        1.849e-03  2.907e-03   0.636  0.52506
## xw_03       -1.857e-02  7.212e-03  -2.575  0.01026 *
## xs_01       -3.936e-02  5.558e-01  -0.071  0.94357
## xs_02       -4.305e-01  2.179e-01  -1.976  0.04861 *
## xs_03       -4.488e-02  3.876e-01  -0.116  0.90788
## xs_04       -6.905e-01  7.780e-01  -0.888  0.37515
## xs_05       -7.635e-02  5.123e-01  -0.149  0.88158
## xs_06       -1.958e-01  1.802e-01  -1.086  0.27777
## I(xb_01^2)  -5.065e-03  6.313e-03  -0.802  0.42263
## I(xb_02^2)   3.204e-03  2.740e-03   1.170  0.24263
```

```
## I(xb_03^2)   3.259e-03  3.650e-03   0.893  0.37234
## I(xn_01^2)   2.125e-03  6.385e-03   0.333  0.73939
## I(xn_02^2)  -2.360e-03  2.846e-03  -0.829  0.40734
## I(xn_03^2)  -1.643e-03  3.745e-03  -0.439  0.66108
## I(xa_01^2)   1.096e-03  1.503e-03   0.730  0.46594
## I(xa_02^2)   3.061e-04  5.552e-04   0.551  0.58157
## I(xa_03^2)  -1.692e-03  1.045e-03  -1.619  0.10604
## I(xb_04^2)   1.888e-02  4.306e-02   0.439  0.66117
## I(xb_05^2)   4.055e-02  1.992e-02   2.036  0.04220 *
## I(xb_06^2)  -3.457e-03  9.904e-03  -0.349  0.72717
## I(xb_07^2)   5.385e-03  1.570e-02   0.343  0.73176
## I(xb_08^2)  -2.156e-02  1.856e-02  -1.162  0.24581
## I(xn_04^2)   5.545e-02  4.203e-02   1.319  0.18756
## I(xn_05^2)   6.353e-03  2.020e-02   0.315  0.75324
## I(xn_06^2)   4.552e-03  1.200e-02   0.379  0.70454
## I(xn_07^2)  -2.530e-02  1.995e-02  -1.268  0.20532
## I(xn_08^2)  -1.814e-02  1.798e-02  -1.009  0.31341
## I(xa_04^2)   2.523e-03  1.040e-02   0.242  0.80850
## I(xa_05^2)   1.144e-03  4.021e-03   0.284  0.77618
## I(xa_06^2)   4.625e-04  1.734e-03   0.267  0.78976
## I(xa_07^2)  -5.621e-03  4.599e-03  -1.222  0.22215
## I(xa_08^2)  -1.431e-03  5.264e-03  -0.272  0.78586
## I(xw_01^2)  -1.079e-04  7.664e-05  -1.408  0.15968
## I(xw_02^2)  -5.202e-06  3.465e-05  -0.150  0.88072
## I(xw_03^2)   1.366e-04  4.721e-05   2.893  0.00395 **
## I(xs_01^2)   4.994e-01  5.338e-01   0.936  0.34987
## I(xs_02^2)  -1.512e-01  2.848e-01  -0.531  0.59568
## I(xs_03^2)  -1.675e-02  2.138e-01  -0.078  0.93758
## I(xs_04^2)   2.033e+00  9.966e-01   2.040  0.04183 *
## I(xs_05^2)  -5.675e-01  8.256e-01  -0.687  0.49208
## I(xs_06)            NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3632 on 611 degrees of freedom
## Multiple R-squared:  0.5832, Adjusted R-squared:  0.5389
## F-statistic: 13.15 on 65 and 611 DF,  p-value: < 2.2e-16
```

Interacting linear and quadratic of three statistically significant variables from continous only linear additive model

```
mod_three_signif <- lm (log_response ~ (xb_04 + I(xb_04^2))*(xb_07 + I(xb_07^2)) * (x
w_01 + I(xw_01^2)), data = df_continuous )
```

Compile Performance Metrics on all 9 models

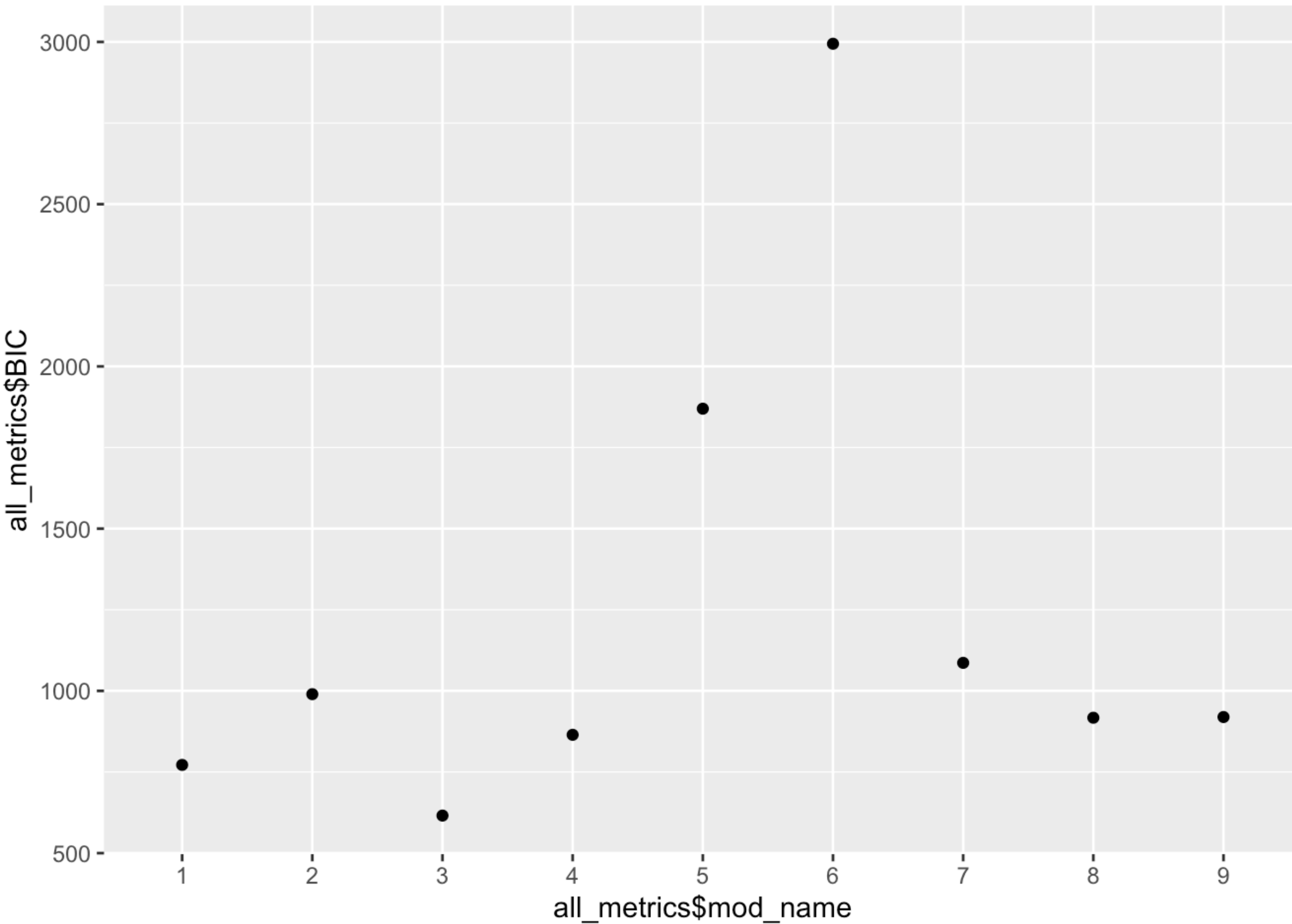
```
extract_metrics <- function (mod, mod_name)
{
  broom::glance(mod) %>% mutate(mod_name = mod_name)
}
```

```
all_metrics <- purrr::map2_dfr(list(mod_continous, mod_categorical,mod_cat_cont, mod_
interact_region, mod_interact_customer,mod_pairwise_cont, mod_lin_quad, mod_quadratic
, mod_three_signif), as.character(1:9), extract_metrics)

all_metrics
```

```
## # A tibble: 9 × 13
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.542         0.519 0.371     23.1  1.16e- 87    33 -272.   614.  772.
## 2     0.220         0.207 0.476     17.0  7.93e- 30    11 -453.   931.  990.
## 3     0.673         0.650 0.316     29.6  1.06e-124    44 -158.   408.  616.
## 4     0.727         0.679 0.303     15.2  6.28e-110   101  -96.7  399.  865.
## 5     0.821         0.679 0.303      5.79 3.41e- 55   299   46.1  510. 1870.
## 6     0.924         0.556 0.357      2.51 6.53e-  9   561  338.   451. 2994.
## 7     0.134         0.114 0.503      6.82 7.23e- 14    15 -488.  1010. 1086.
## 8     0.583         0.539 0.363     13.2  7.27e- 80    65 -240.   615.  917.
## 9     0.391         0.367 0.426     16.1  4.33e- 54    26 -368.   793.  919.
## # … with 4 more variables: deviance <dbl>, df.residual <int>, nobs <int>,
## #   mod_name <chr>
```

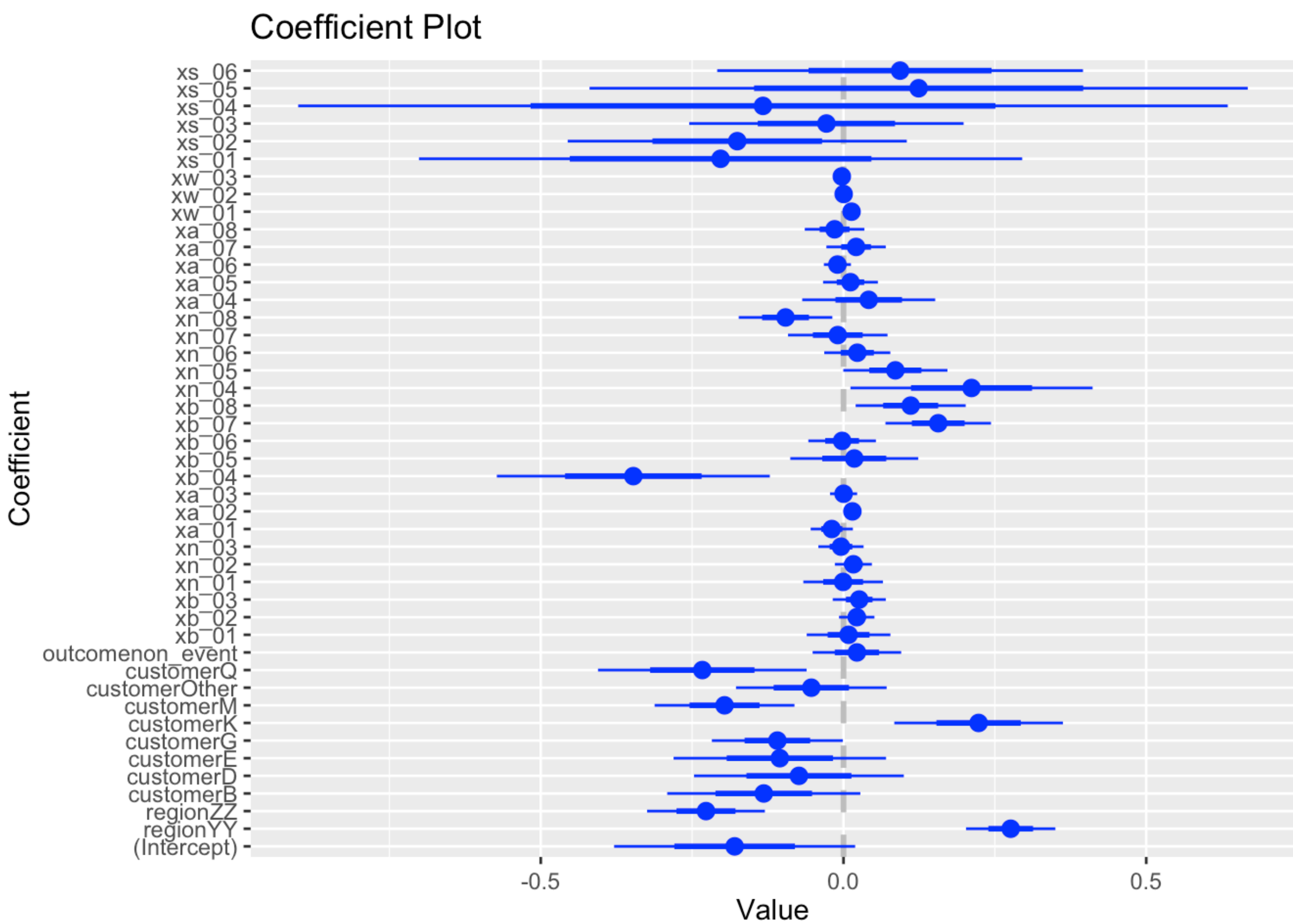
Using the BIC values, the third model (mod_cat_cont) is the best model.

```
ggplot(mapping = aes(y = all_metrics$BIC, x = all_metrics$mod_name)) + geom_point()
```



Examining the Coefficients of the top three models selected using R-squared.

```
mod_cat_cont %>% coefplot::coefplot()
```



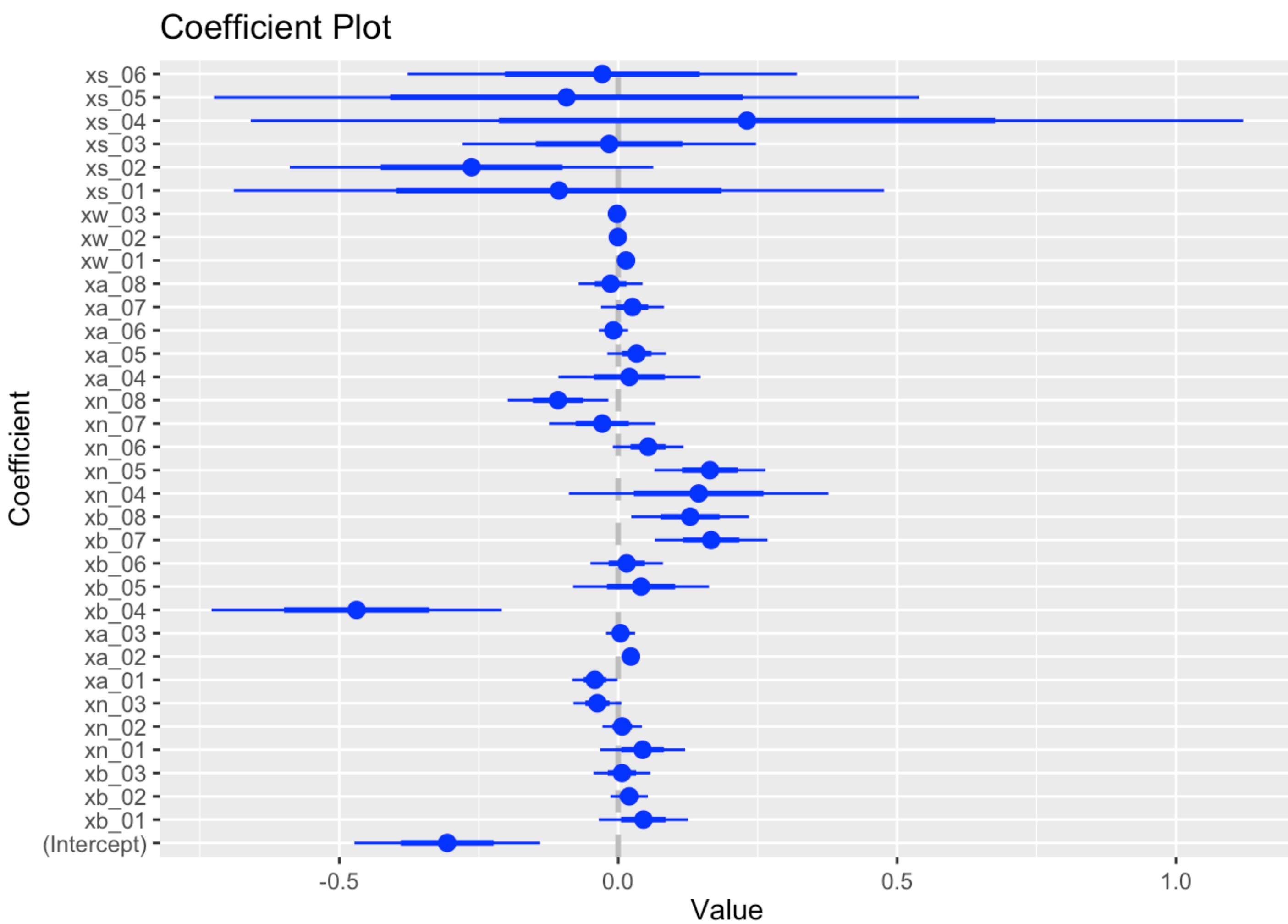

```
mod_cat_cont %>% summary()
```

```
##
## Call:
## lm(formula = log_response ~ region + customer + outcome + xb_01 +
##     xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + xa_01 + xa_02 + xa_03 +
##     xb_04 + xb_05 + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 +
##     xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 + xw_01 +
##     xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 + xs_06,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08667 -0.19713 -0.00152  0.19722  1.55976
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.1797946  0.0995253  -1.807 0.071313 .
## regionYY          0.2761742  0.0368945   7.486 2.40e-13 ***
```

```
## regionZZ         -0.2270989  0.0486347  -4.669 3.69e-06 ***
## customerB        -0.1317048  0.0797339  -1.652 0.099071 .
## customerD        -0.0736332  0.0866413  -0.850 0.395723
## customerE        -0.1051613  0.0877251  -1.199 0.231071
## customerG        -0.1092338  0.0540802  -2.020 0.043821 *
## customerK         0.2231309  0.0696342   3.204 0.001422 **
## customerM        -0.1964893  0.0577596  -3.402 0.000712 ***
## customerOther    -0.0531574  0.0622005  -0.855 0.393089
## customerQ        -0.2331907  0.0861430  -2.707 0.006972 **
## outcomenon_event  0.0221597  0.0365570   0.606 0.544621
## xb_01             0.0083881  0.0345549   0.243 0.808280
## xb_02             0.0218866  0.0146235   1.497 0.134979
## xb_03             0.0259955  0.0219046   1.187 0.235767
## xn_01            -0.0006398  0.0328876  -0.019 0.984486
## xn_02             0.0162909  0.0152735   1.067 0.286555
## xn_03            -0.0041769  0.0186882  -0.224 0.823217
## xa_01            -0.0193491  0.0174309  -1.110 0.267402
## xa_02             0.0146935  0.0070604   2.081 0.037826 *
## xa_03             0.0001389  0.0111830   0.012 0.990094
## xb_04            -0.3471154  0.1127203  -3.079 0.002164 **
## xb_05             0.0178054  0.0528468   0.337 0.736285
## xb_06            -0.0023030  0.0278978  -0.083 0.934234
## xb_07             0.1564532  0.0435929   3.589 0.000358 ***
## xb_08             0.1109475  0.0454388   2.442 0.014892 *
## xn_04             0.2114462  0.0999907   2.115 0.034850 *
## xn_05             0.0856392  0.0430116   1.991 0.046903 *
## xn_06             0.0228028  0.0272780   0.836 0.403503
## xn_07            -0.0094721  0.0410933  -0.231 0.817776
## xn_08            -0.0957456  0.0386609  -2.477 0.013527 *
## xa_04             0.0416379  0.0549285   0.758 0.448711
## xa_05             0.0115047  0.0226307   0.508 0.611372
## xa_06            -0.0100788  0.0111927  -0.900 0.368208
## xa_07             0.0207631  0.0245260   0.847 0.397553
## xa_08            -0.0147295  0.0246192  -0.598 0.549858
## xw_01             0.0132895  0.0029451   4.512 7.64e-06 ***
## xw_02             0.0002824  0.0015227   0.185 0.852916
## xw_03            -0.0027854  0.0016893  -1.649 0.099683 .
## xs_01            -0.2029929  0.2490773  -0.815 0.415391
## xs_02            -0.1755021  0.1400204  -1.253 0.210522
## xs_03            -0.0282183  0.1132876  -0.249 0.803375
## xs_04            -0.1329774  0.3838353  -0.346 0.729124
## xs_05             0.1241057  0.2717536   0.457 0.648055
## xs_06             0.0934684  0.1510491   0.619 0.536274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3163 on 632 degrees of freedom
```

```
## Multiple R-squared:  0.6732, Adjusted R-squared:  0.6504
## F-statistic: 29.58 on 44 and 632 DF,  p-value: < 2.2e-16
```

```
mod_continous %>% coefplot::coefplot()
```




```
mod_continous %>% summary()
```

```
##
## Call:
## lm(formula = log_response ~ xb_01 + xb_02 + xb_03 + xn_01 + xn_02 +
##     xn_03 + xa_01 + xa_02 + xa_03 + xb_04 + xb_05 + xb_06 + xb_07 +
##     xb_08 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 + xa_04 + xa_05 +
##     xa_06 + xa_07 + xa_08 + xw_01 + xw_02 + xw_03 + xs_01 + xs_02 +
##     xs_03 + xs_04 + xs_05 + xs_06, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1981 -0.2279 -0.0121  0.2308  1.7563
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.3065731  0.0832408  -3.683 0.000250 ***
## xb_01        0.0451928  0.0399721   1.131 0.258642
## xb_02        0.0197021  0.0167751   1.174 0.240634
## xb_03        0.0067157  0.0253421   0.265 0.791094
## xn_01        0.0436725  0.0380814   1.147 0.251883
## xn_02        0.0070652  0.0176705   0.400 0.689412
## xn_03       -0.0372646  0.0215715  -1.727 0.084559 .
## xa_01       -0.0419267  0.0202378  -2.072 0.038691 *
## xa_02        0.0226529  0.0082193   2.756 0.006016 **
## xa_03        0.0040960  0.0130010   0.315 0.752823
## xb_04       -0.4689904  0.1300432  -3.606 0.000335 ***
## xb_05        0.0408992  0.0609840   0.671 0.502682
## xb_06        0.0151322  0.0324939   0.466 0.641592
## xb_07        0.1664855  0.0505356   3.294 0.001040 **
## xb_08        0.1289615  0.0527916   2.443 0.014840 *
## xn_04        0.1442231  0.1163268   1.240 0.215498
## xn_05        0.1644335  0.0497648   3.304 0.001005 **
## xn_06        0.0536260  0.0316213   1.696 0.090392 .
## xn_07       -0.0286324  0.0475703  -0.602 0.547454
## xn_08       -0.1079277  0.0451179  -2.392 0.017037 *
## xa_04        0.0200902  0.0636645   0.316 0.752436
## xa_05        0.0330188  0.0263357   1.254 0.210383
## xa_06       -0.0085964  0.0130347  -0.660 0.509809
## xa_07        0.0255978  0.0282339   0.907 0.364940
## xa_08       -0.0136748  0.0286572  -0.477 0.633393
## xw_01        0.0139666  0.0033112   4.218 2.82e-05 ***
## xw_02       -0.0007236  0.0017305  -0.418 0.675994
## xw_03       -0.0022127  0.0019083  -1.160 0.246682
## xs_01       -0.1063306  0.2914305  -0.365 0.715338
## xs_02       -0.2629219  0.1628957  -1.614 0.107007
## xs_03       -0.0161401  0.1315986  -0.123 0.902425
## xs_04        0.2308584  0.4448027   0.519 0.603930
## xs_05       -0.0926908  0.3159142  -0.293 0.769307
## xs_06       -0.0286369  0.1745249  -0.164 0.869716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.371 on 643 degrees of freedom
## Multiple R-squared:  0.5424, Adjusted R-squared:  0.5189
## F-statistic:  23.1 on 33 and 643 DF,  p-value: < 2.2e-16
```

I am choosing the simple linear additive model to compare against the model with both the categorical and continuous predictors because I want to see if the added complexity in the categorical and continous model is actually necessary or if the continuous additive model is sufficient.

Regression - Part B

Fitting Bayesian Linear Models

Creating Design Matrix for each of the two models

```
Xmat_continuous <- model.matrix(~ xb_01 + xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + xa_
01 + xa_02 + xa_03 +xb_04 + xb_05
                    + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 +
xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df)

Xmat_cat_cont <- model.matrix( ~region + customer + outcome +  xb_01 + xb_02 + xb_03
+ xn_01 + xn_02 + xn_03 + xa_01 + xa_02 + xa_03 +xb_04 + xb_05 + xb_06 + xb_07 + xb_0
8 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df)
```

Create list of information

```
my_info_cat_cont<- list(
  yobs = df$log_response,
  design_matrix = Xmat_cat_cont,
  mu_beta = 0,
  tau_beta = 1,
  sigma_rate = 1
)

my_info_cont <- list(
  yobs = df$log_response,
  design_matrix = Xmat_continuous,
  mu_beta = 0,
  tau_beta = 1,
  sigma_rate = 1
)
```

We will be assuming a weak prior standard deviation on the regression coefficients. The prior mean will be 0.

Create Log Posterior Function

```
lm_logpost <- function(unknowns, my_info)
{

  length_beta <- ncol(my_info$design_matrix)
  beta_v <- unknowns[1:length_beta]

  lik_varphi <- unknowns[length_beta + 1]
  lik_sigma <- exp(lik_varphi)

  X <- my_info$design_matrix
  mu <- as.vector( X %*% as.matrix(beta_v) )

  log_lik <- sum(dnorm(x = my_info$yobs,
                       mean = mu,
                       sd = lik_sigma,
                       log = TRUE))

  log_prior_beta <- sum(dnorm(x = beta_v,
                              mean = my_info$mu_beta,
                              sd = my_info$tau_beta,
                              log = TRUE))

  log_prior_sigma <- dexp(x = lik_sigma,
                          rate = my_info$sigma_rate,
                          log = TRUE)

  log_prior <- log_prior_beta + log_prior_sigma

  log_derive_adjust <- lik_varphi

  log_lik + log_prior + log_derive_adjust
}
```

Fitting the Bayesian Linear Models Using Laplace’s Approximation

```
my_laplace <- function(start_guess, logpost_func, ...)
{
  # code adapted from the `LearnBayes`` function `laplace()`
  fit <- optim(start_guess,
               logpost_func,
               gr = NULL,
               ...,
               method = "BFGS",
               hessian = TRUE,
               control = list(fnscale = -1, maxit = 1001))


  mode <- fit$par
  post_var_matrix <- -solve(fit$hessian)
  p <- length(mode)
  int <- p/2 * log(2 * pi) + 0.5 * log(det(post_var_matrix)) + logpost_func(mode, ...
)
  # package all of the results into a list
  list(mode = mode,
       var_matrix = post_var_matrix,
       log_evidence = int,
       converge = ifelse(fit$convergence == 0,
                         "YES",
                         "NO"),
       iter_counts = as.numeric(fit$counts[1]))


}
```

```
laplace_cont <- my_laplace(rep(0, ncol(Xmat_continuous)+1), lm_logpost, my_info_cont)
laplace_cat_cont <- my_laplace(rep(0, ncol(Xmat_cat_cont)+1), lm_logpost, my_info_cat
_cont)
```

Using Bayes Factor to identify the better of the two models

```
exp(laplace_cat_cont$log_evidence) / exp(laplace_cont$log_evidence)
```

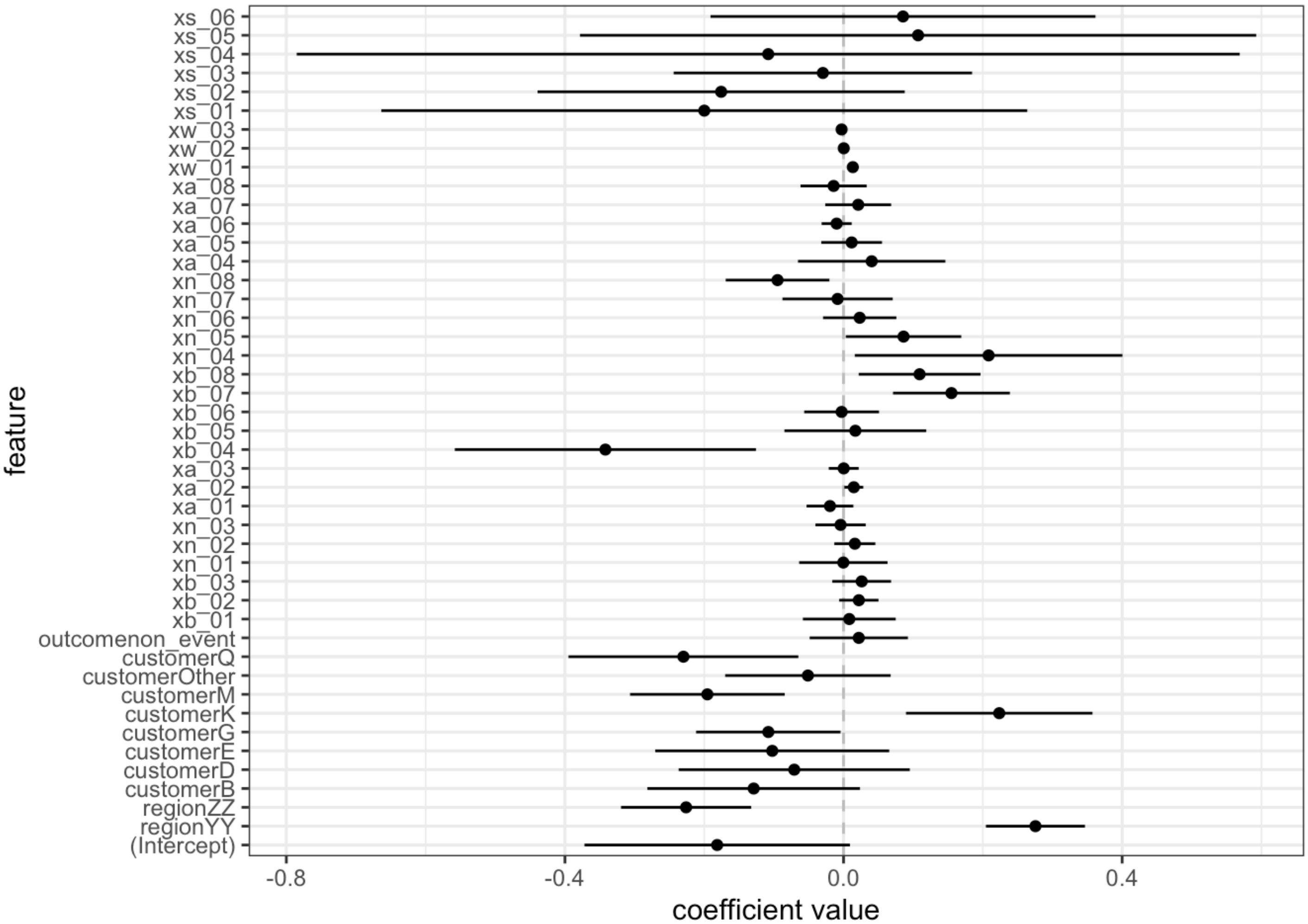
```
## [1] 1.670778e+32
```

The result of the performance metric, Bayes Factor is much greater than 1. This indicates that there is more evidence for the model with both categorical and continuous inputs than just the additive linear model with only continuous inputs.

Visualizing the regression coefficient posterior summary statistics for the best model.

```
viz_post_coefs <- function(post_means, post_sds, xnames)
{
  tibble::tibble(
    mu = post_means,
    sd = post_sds,
    x = xnames
  ) %>%
    mutate(x = factor(x, levels = xnames)) %>%
    ggplot(mapping = aes(x = x)) +
    geom_hline(yintercept = 0, color = 'grey', linetype = 'dashed') +
    geom_point(mapping = aes(y = mu)) +
    geom_linerange(mapping = aes(ymin = mu - 2 * sd,
                                 ymax = mu + 2 * sd,
                                 group = x)) +
    labs(x = 'feature', y = 'coefficient value') +
    coord_flip() +
    theme_bw()
}
```

```
viz_post_coefs(laplace_cat_cont$mode[1:ncol(Xmat_cat_cont)],
               sqrt(diag(laplace_cat_cont$var_matrix)[1:ncol(Xmat_cat_cont)]),
               colnames(Xmat_cat_cont))
```




```
mod_cat_cont %>% summary()
```

```
##
## Call:
## lm(formula = log_response ~ region + customer + outcome + xb_01 +
##     xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + xa_01 + xa_02 + xa_03 +
##     xb_04 + xb_05 + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 +
##     xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 + xw_01 +
##     xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 + xs_06,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08667 -0.19713 -0.00152  0.19722  1.55976
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.1797946  0.0995253  -1.807 0.071313 .
## regionYY          0.2761742  0.0368945   7.486 2.40e-13 ***
```

```
## regionZZ         -0.2270989  0.0486347  -4.669 3.69e-06 ***
## customerB        -0.1317048  0.0797339  -1.652 0.099071 .
## customerD        -0.0736332  0.0866413  -0.850 0.395723
## customerE        -0.1051613  0.0877251  -1.199 0.231071
## customerG        -0.1092338  0.0540802  -2.020 0.043821 *
## customerK         0.2231309  0.0696342   3.204 0.001422 **
## customerM        -0.1964893  0.0577596  -3.402 0.000712 ***
## customerOther    -0.0531574  0.0622005  -0.855 0.393089
## customerQ        -0.2331907  0.0861430  -2.707 0.006972 **
## outcomenon_event  0.0221597  0.0365570   0.606 0.544621
## xb_01             0.0083881  0.0345549   0.243 0.808280
## xb_02             0.0218866  0.0146235   1.497 0.134979
## xb_03             0.0259955  0.0219046   1.187 0.235767
## xn_01            -0.0006398  0.0328876  -0.019 0.984486
## xn_02             0.0162909  0.0152735   1.067 0.286555
## xn_03            -0.0041769  0.0186882  -0.224 0.823217
## xa_01            -0.0193491  0.0174309  -1.110 0.267402
## xa_02             0.0146935  0.0070604   2.081 0.037826 *
## xa_03             0.0001389  0.0111830   0.012 0.990094
## xb_04            -0.3471154  0.1127203  -3.079 0.002164 **
## xb_05             0.0178054  0.0528468   0.337 0.736285
## xb_06            -0.0023030  0.0278978  -0.083 0.934234
## xb_07             0.1564532  0.0435929   3.589 0.000358 ***
## xb_08             0.1109475  0.0454388   2.442 0.014892 *
## xn_04             0.2114462  0.0999907   2.115 0.034850 *
## xn_05             0.0856392  0.0430116   1.991 0.046903 *
## xn_06             0.0228028  0.0272780   0.836 0.403503
## xn_07            -0.0094721  0.0410933  -0.231 0.817776
## xn_08            -0.0957456  0.0386609  -2.477 0.013527 *
## xa_04             0.0416379  0.0549285   0.758 0.448711
## xa_05             0.0115047  0.0226307   0.508 0.611372
## xa_06            -0.0100788  0.0111927  -0.900 0.368208
## xa_07             0.0207631  0.0245260   0.847 0.397553
## xa_08            -0.0147295  0.0246192  -0.598 0.549858
## xw_01             0.0132895  0.0029451   4.512 7.64e-06 ***
## xw_02             0.0002824  0.0015227   0.185 0.852916
## xw_03            -0.0027854  0.0016893  -1.649 0.099683 .
## xs_01            -0.2029929  0.2490773  -0.815 0.415391
## xs_02            -0.1755021  0.1400204  -1.253 0.210522
## xs_03            -0.0282183  0.1132876  -0.249 0.803375
## xs_04            -0.1329774  0.3838353  -0.346 0.729124
## xs_05             0.1241057  0.2717536   0.457 0.648055
## xs_06             0.0934684  0.1510491   0.619 0.536274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3163 on 632 degrees of freedom
```

```
## Multiple R-squared:  0.6732, Adjusted R-squared:  0.6504
## F-statistic: 29.58 on 44 and 632 DF,  p-value: < 2.2e-16
```

POSTERIOR UNCERTAINITY ON SIGMA

Regression - Part C

Making predictions

```
generate_lm_post_samples <- function(mvn_result, length_beta, num_samples)
{
  MASS::mvrnorm(n = num_samples,
                mu = mvn_result$mode,
                Sigma = mvn_result$var_matrix) %>%
    as.data.frame() %>% tibble::as_tibble() %>%
    purrr::set_names(c(sprintf("beta_%02d", 0:(length_beta-1)), "varphi")) %>%
    mutate(sigma = exp(varphi))
}
```

```
set.seed(87123)
post_samples_cont <- generate_lm_post_samples(laplace_cont, ncol(Xmat_continuous), 25
00)
post_samples_cat_cont <- generate_lm_post_samples(laplace_cat_cont, ncol(Xmat_cat_con
t), 2500)
```

```
post_lm_pred_samples <- function(Xnew, Bmat, sigma_vector)
{
  # number of new prediction locations
  M <- nrow(Xnew)
  # number of posterior samples
  S <- nrow(Bmat)

  # matrix of linear predictors
  Umat <- Xnew %*% t(Bmat)

  # assmeble matrix of sigma samples
  Rmat <- matrix(rep(sigma_vector, M), M, byrow = TRUE)

  # generate standard normal and assemble into matrix
  Zmat <- matrix(rnorm(M*S), M, byrow = TRUE)

  # calculate the random observation predictions
  Ymat <- Umat + Rmat * Zmat

  # package together
  list(Umat = Umat, Ymat = Ymat)
}
```

```
make_post_lm_pred <- function(Xnew, post)
{
  Bmat <- post %>% select(starts_with("beta_")) %>% as.matrix()

  sigma_vector <- post %>% pull(sigma)

  post_lm_pred_samples(Xnew, Bmat, sigma_vector)
}
```

Generate Posterior predictions on the linear additive model with only continous inputs

```
post_pred_samples_cont <- make_post_lm_pred(Xmat_continuous,
                                            post_samples_cont)
```

Generate Posterior predictions on the linear additive model with categorical and continous inputs

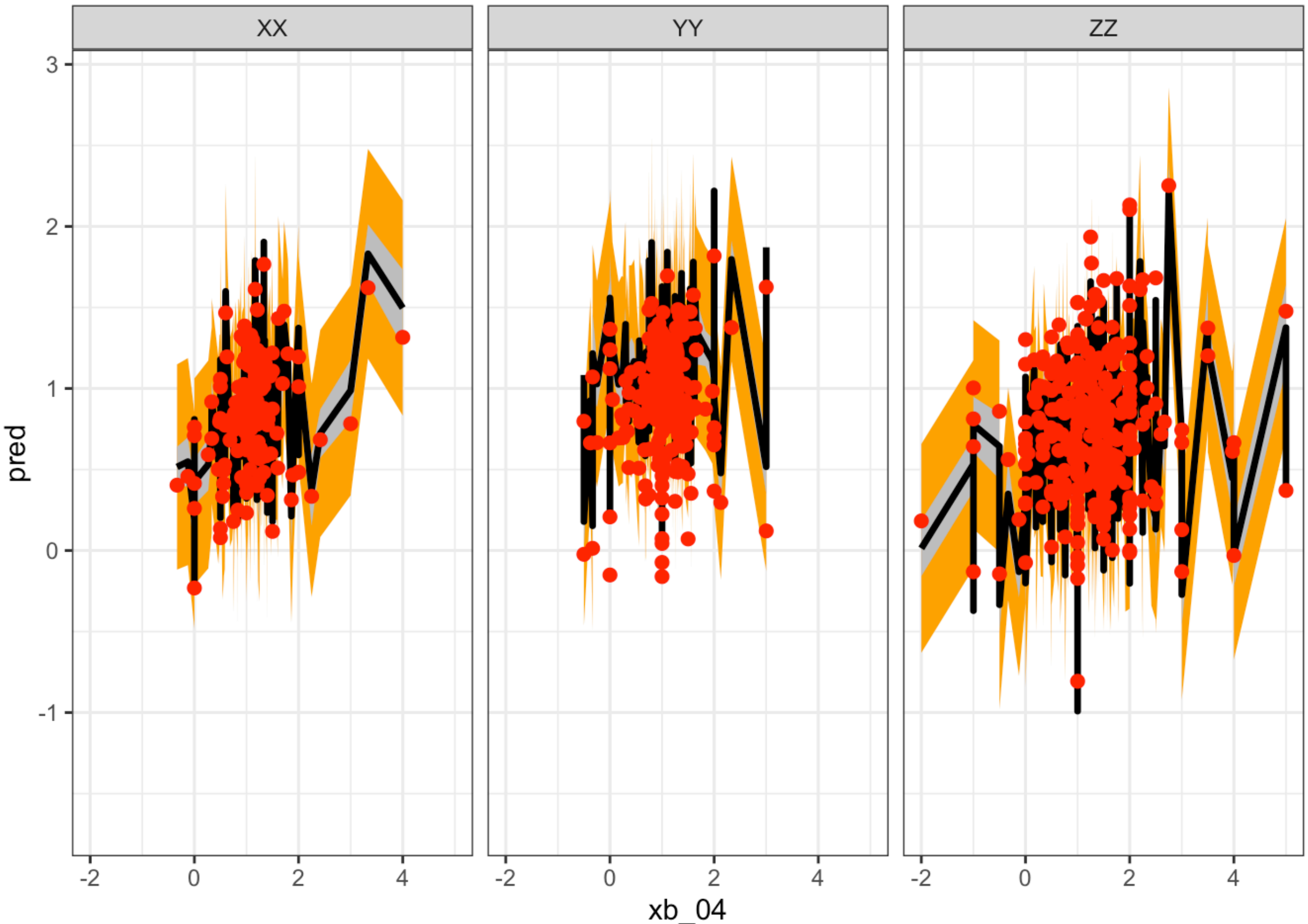
```
post_pred_samples_cont <- make_post_lm_pred(Xmat_cat_cont,
                                            post_samples_cat_cont)
```

```
tidy_predict <- function(mod, xnew)
{
  pred_df <- predict(mod, xnew, interval = "confidence") %>%
    as.data.frame() %>% tibble::as_tibble() %>%
    dplyr::select(pred = fit, ci_lwr = lwr, ci_upr = upr) %>%
    bind_cols(predict(mod, xnew, interval = 'prediction') %>%
                as.data.frame() %>% tibble::as_tibble() %>%
                dplyr::select(pred_lwr = lwr, pred_upr = upr))

  xnew %>% bind_cols(pred_df)
}
```

```
pred_cont <- tidy_predict(mod_continous, df)
pred_cat_cont <- tidy_predict(mod_cat_cont, df)
```

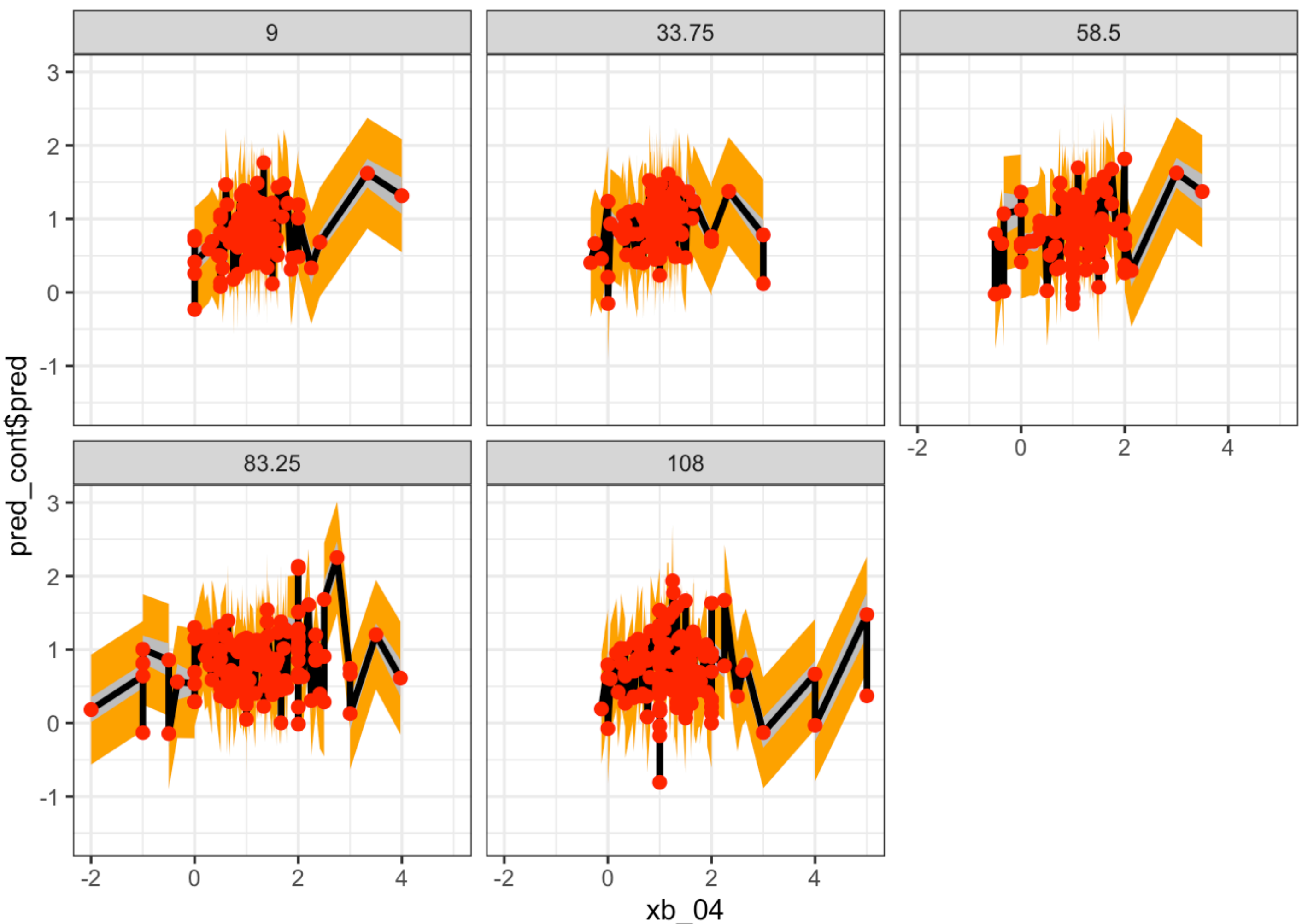
Visualizing Predictive Trends

```
pred_cat_cont %>%
  ggplot(mapping = aes(x = xb_04)) +
  geom_ribbon(mapping = aes(ymin = pred_lwr,
                            ymax = pred_upr),
              fill = 'orange') +
  geom_ribbon(mapping = aes(ymin = ci_lwr,
                            ymax = ci_upr),
              fill = 'grey') +
  geom_line(mapping = aes(y = pred),
            color = 'black', size = 1.2) +
  geom_point(data = df,
             mapping = aes(x = xb_04, y = pred_cont$pred),
             color = 'red', size = 2) +
  facet_wrap(~region) +
  theme_bw()
```




```
xw_01_vals <- pred_cont['xw_01']
min <- min(xw_01_vals)
max <- max(xw_01_vals)
mean <- (min + max) /2


med_0.25 <- (min + mean) /2
med_0.75 <- (mean + max) /2
levels <- c(min, med_0.25, mean, med_0.75, max)
levels_spaced <- rep(levels, each = nrow(df)/5)
levels_spaced[length(levels_spaced) + 1]  = levels_spaced[length(levels_spaced) ]
levels_spaced[length(levels_spaced) + 1]  = levels_spaced[length(levels_spaced) ]
pred_cont['levels'] = as.factor(levels_spaced)
```

```
pred_cont %>%
  ggplot(mapping = aes(x = xb_04)) +
  geom_ribbon(mapping = aes(ymin = pred_lwr,
                            ymax = pred_upr,
                            group = pred_cont$levels),
              fill = 'orange') +
  geom_ribbon(mapping = aes(ymin = ci_lwr,
                            ymax = ci_upr,
                            group = pred_cont$levels),
              fill = 'grey') +
  geom_line(mapping = aes(y = pred_cont$pred,
                            group = pred_cont$levels),
            color = 'black', size = 1.2) +
  geom_point(data = df,
             mapping = aes(x = xb_04, y = pred_cont$pred),
             color = 'red', size = 2) + facet_wrap(~pred_cont$levels) +
  theme_bw()
```



In the second plot, the plot is faceted with the variable xw_01. The predictions look pretty consistent. They majority of the data points are between 0 and 2.

Regresiion - Part D

Training Linear Models

```
df_reg <- df %>%
  select(region, customer, starts_with('x'), log_response)
```

```
my_metric <- "RMSE"
my_ctrl <- trainControl(method = "repeatedcv" , number = 5, repeats = 5, my_metric)
```

All categorical and continuous inputs (also model selected from Part A)

```
cat_cont_tune <- caret::train(log_response ~  ., data = df_reg, method = 'lm' , metri
c = my_metric,
                        preProcess = c('center','scale'), trControl = my_ctrl)
```

All pairwise interactions of continuous inputs and additive categorical features

```
pairwise_tune <- caret::train(log_response ~ (.)^2 + region + customer ,data = df_reg
, method = 'lm', metric = my_metric,
                        preProcess = c('center','scale'), trControl = my_ctrl)
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

Training the second model from Regression Part A (mod_continuous)

```
continuous_tune <- caret::train(log_response ~ . ,data = df_continuous, method = 'lm'
, metric = my_metric,
                        preProcess = c('center','scale'), trControl = my_ctrl)

continuous_tune
```

```
## Linear Regression
##
## 677 samples
##  33 predictor
##
## Pre-processing: centered (33), scaled (33)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 543, 542, 541, 541, 541, 542, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.3827837  0.4958512  0.2958764
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Regularized Regression with Elastic Net - Using glmnet

All pairwise interactions of continuous inputs including additive categorical features

```
pairwise_enet <-  caret::train(log_response ~ (.)^2 + region + customer ,data = df_re
g, method = 'glmnet', metric = my_metric,
                        preProcess = c('center','scale'), trControl = my_ctrl)
```

```
## Warning in preProcess.default(method = c("center", "scale"), x =
## structure(c(0, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

Regularized Regression with Elastic Net - mod_cat_cont

```
mod_cat_cont_enet <-  caret::train(log_response ~  .,data = df_reg, method = 'glmnet'
, metric = my_metric,
                        preProcess = c('center','scale'), trControl = my_ctrl)
```

Using Neural Network to Train, Tune and Evaluate the mod_cat_cont model

```
my_metric <- "RMSE"
my_ctrl <- trainControl(method = "repeatedcv" , number = 5, repeats = 5, my_metric)
```

```
set.seed(4321)
fit_nnet<- train(log_response ~ .,
                        data = df_reg,  method = "nnet",  metric = my_metric, trContr
ol = my_ctrl,
                        preProcess = c('center', 'scale'),
                        trace = FALSE)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

Using Random Forest to Train, evaluate, and tune the mod_cat_cont model.

```
set.seed(4321)

fit_rf <- train(log_response ~ .,
                        data = df_reg,  method = "rf",  metric = my_metric, trControl
= my_ctrl,
                trace = FALSE)
```

Using Extrem Gradient Boosted Tree to Train, evaluate, and tune the mod_cat_cont model.

```
set.seed(4321)
fit_xgb <- train(log_response ~. ,
                        data = df_reg, method = 'xgbTree', metric = my_metric, trCont
rol = my_ctrl,  preProcess = c('center', 'scale'), verbosity = 0)
```

Using Partial Least Squares to Train, evaluate, and tune the mod_cat_cont model.

```
set.seed(4321)
fit_pls <- train(log_response ~ .,
                        data = df_reg,  method = "pls",  metric = my_metric, trContro
l = my_ctrl,
                        preProcess = c('center', 'scale'), importance = TRUE,
                        trace = FALSE)
```

Using KNN to Train, evaluate, and tune the mod_cat_cont model.

```
fit_knn <- train(log_response ~ .,
                        data = df_reg,  method = "knn",  metric = my_metric, trContro
l = my_ctrl,
                        preProcess = c('center', 'scale'))
```

Using the preprocessing options of centering and scaling, using RMSE for the performance metric, and 5 fold CV repeated 5 times, the best model

```
cat_cont_tune
```

```
## Linear Regression
##
## 677 samples
##  35 predictor
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 543, 542, 541, 541, 541, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.3316742  0.6191067  0.2571752
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
pairwise_tune
```

```
## Linear Regression
##
## 677 samples
##  35 predictor
##
## Pre-processing: centered (917), scaled (917)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 543, 542, 541, 541, 541, 541, ...
## Resampling results:
##
##   RMSE          Rsquared     MAE
##   4.384736e+13  0.007700463  5.934113e+12
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
pairwise_enet
```

```
## glmnet
##
## 677 samples
##  35 predictor
##
## Pre-processing: centered (917), scaled (917)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 542, 541, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda      RMSE       Rsquared   MAE
##   0.10   0.01986111  0.2895734  0.7115773  0.2209406
##   0.10   0.06280634  0.2892408  0.7104730  0.2224893
##   0.10   0.19861107  0.3057961  0.6821587  0.2377119
##   0.55   0.01986111  0.2930417  0.7025400  0.2256930
##   0.55   0.06280634  0.3165188  0.6630571  0.2456847
##   0.55   0.19861107  0.3851537  0.5762559  0.3009650
##   1.00   0.01986111  0.3028425  0.6840027  0.2343825
##   1.00   0.06280634  0.3398100  0.6286261  0.2629006
##   1.00   0.19861107  0.4496350  0.5268355  0.3593552
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.1 and lambda = 0.06280634.
```

```
fit_nnet
```

```
## Neural Network
##
## 677 samples
##  35 predictor
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 544, 541, 541, 541, 541, ...
## Resampling results across tuning parameters:
##
##   size  decay  RMSE       Rsquared   MAE
##   1     0e+00  0.5527321  0.5116873  0.4470228
##   1     1e-04  0.4491423  0.3967188  0.3495732
##   1     1e-01  0.3981355  0.5065941  0.3007541
##   3     0e+00  0.5329273  0.4171356  0.4269987
##   3     1e-04  0.4296851  0.4239912  0.3305346
##   3     1e-01  0.3894848  0.5367057  0.2901814
##   5     0e+00  0.5115434  0.4140724  0.4083407
##   5     1e-04  0.4153886  0.4471864  0.3167712
##   5     1e-01  0.3867758  0.5465026  0.2885699
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were size = 5 and decay = 0.1.
```

```
fit_rf
```

```
## Random Forest
##
## 677 samples
##  35 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 544, 541, 541, 541, 541, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE       Rsquared   MAE
##    2    0.3558164  0.6208213  0.2753076
##   22    0.3169718  0.6593186  0.2415222
##   43    0.3194658  0.6478606  0.2439380
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 22.
```

```
fit_xgb
```

```
## eXtreme Gradient Boosting
##
## 677 samples
##  35 predictor
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 544, 541, 541, 541, 541, ...
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  subsample  nrounds  RMSE       Rsquared
##   0.3  1          0.6               0.50        50      0.3419299  0.5921881
##   0.3  1          0.6               0.50       100      0.3404880  0.5980447
##   0.3  1          0.6               0.50       150      0.3404908  0.5997819
##   0.3  1          0.6               0.75        50      0.3388544  0.6006702
##   0.3  1          0.6               0.75       100      0.3348654  0.6092459
##   0.3  1          0.6               0.75       150      0.3361678  0.6069911
##   0.3  1          0.6               1.00        50      0.3403684  0.5998813
##   0.3  1          0.6               1.00       100      0.3332592  0.6134024
##   0.3  1          0.6               1.00       150      0.3320072  0.6162209
##   0.3  1          0.8               0.50        50      0.3394429  0.5970773
##   0.3  1          0.8               0.50       100      0.3373015  0.6039766
##   0.3  1          0.8               0.50       150      0.3376955  0.6052552
##   0.3  1          0.8               0.75        50      0.3388239  0.6005342
##   0.3  1          0.8               0.75       100      0.3354910  0.6080990
##   0.3  1          0.8               0.75       150      0.3367270  0.6058532
##   0.3  1          0.8               1.00        50      0.3387704  0.6040822
##   0.3  1          0.8               1.00       100      0.3325038  0.6150634
##   0.3  1          0.8               1.00       150      0.3320288  0.6159957
##   0.3  2          0.6               0.50        50      0.3262382  0.6301831
##   0.3  2          0.6               0.50       100      0.3259037  0.6331146
##   0.3  2          0.6               0.50       150      0.3280969  0.6302533
##   0.3  2          0.6               0.75        50      0.3145717  0.6544190
##   0.3  2          0.6               0.75       100      0.3144640  0.6567671
##   0.3  2          0.6               0.75       150      0.3185565  0.6497880
##   0.3  2          0.6               1.00        50      0.3126478  0.6592319
##   0.3  2          0.6               1.00       100      0.3108312  0.6632143
##   0.3  2          0.6               1.00       150      0.3129642  0.6590925
##   0.3  2          0.8               0.50        50      0.3242988  0.6347042
##   0.3  2          0.8               0.50       100      0.3288370  0.6282793
##   0.3  2          0.8               0.50       150      0.3349609  0.6171941
##   0.3  2          0.8               0.75        50      0.3155080  0.6537373
##   0.3  2          0.8               0.75       100      0.3156270  0.6551699
##   0.3  2          0.8               0.75       150      0.3184164  0.6495158
```

```
##    0.3  2          0.8               1.00       50       0.3128396  0.6595172
##    0.3  2          0.8               1.00      100       0.3114758  0.6626752
##    0.3  2          0.8               1.00      150       0.3141036  0.6578625
##    0.3  3          0.6               0.50       50       0.3273894  0.6297015
##    0.3  3          0.6               0.50      100       0.3323138  0.6224203
##    0.3  3          0.6               0.50      150       0.3352044  0.6173099
##    0.3  3          0.6               0.75       50       0.3156065  0.6533186
##    0.3  3          0.6               0.75      100       0.3181004  0.6493618
##    0.3  3          0.6               0.75      150       0.3196355  0.6469989
##    0.3  3          0.6               1.00       50       0.3106563  0.6646802
##    0.3  3          0.6               1.00      100       0.3117320  0.6626549
##    0.3  3          0.6               1.00      150       0.3132199  0.6598145
##    0.3  3          0.8               0.50       50       0.3268448  0.6321885
##    0.3  3          0.8               0.50      100       0.3330008  0.6227453
##    0.3  3          0.8               0.50      150       0.3370168  0.6146264
##    0.3  3          0.8               0.75       50       0.3161041  0.6523706
##    0.3  3          0.8               0.75      100       0.3189362  0.6480517
##    0.3  3          0.8               0.75      150       0.3217450  0.6426814
##    0.3  3          0.8               1.00       50       0.3152196  0.6532242
##    0.3  3          0.8               1.00      100       0.3177724  0.6491622
##    0.3  3          0.8               1.00      150       0.3197928  0.6454561
##    0.4  1          0.6               0.50       50       0.3466394  0.5822971
##    0.4  1          0.6               0.50      100       0.3465491  0.5862537
##    0.4  1          0.6               0.50      150       0.3510794  0.5792733
##    0.4  1          0.6               0.75       50       0.3392946  0.5982466
##    0.4  1          0.6               0.75      100       0.3358930  0.6073641
##    0.4  1          0.6               0.75      150       0.3383143  0.6039517
##    0.4  1          0.6               1.00       50       0.3419257  0.5928643
##    0.4  1          0.6               1.00      100       0.3379763  0.6026294
##    0.4  1          0.6               1.00      150       0.3376006  0.6038548
##    0.4  1          0.8               0.50       50       0.3424399  0.5929237
##    0.4  1          0.8               0.50      100       0.3440242  0.5923853
##    0.4  1          0.8               0.50      150       0.3519267  0.5787778
##    0.4  1          0.8               0.75       50       0.3387846  0.5995815
##    0.4  1          0.8               0.75      100       0.3384034  0.6033303
##    0.4  1          0.8               0.75      150       0.3413484  0.5976856
##    0.4  1          0.8               1.00       50       0.3392098  0.5989187
##    0.4  1          0.8               1.00      100       0.3350487  0.6088006
##    0.4  1          0.8               1.00      150       0.3349958  0.6097014
##    0.4  2          0.6               0.50       50       0.3402169  0.6033320
##    0.4  2          0.6               0.50      100       0.3452006  0.5970327
##    0.4  2          0.6               0.50      150       0.3525542  0.5853051
##    0.4  2          0.6               0.75       50       0.3294095  0.6252427
##    0.4  2          0.6               0.75      100       0.3343112  0.6180474
##    0.4  2          0.6               0.75      150       0.3366650  0.6145304
##    0.4  2          0.6               1.00       50       0.3229860  0.6360581
##    0.4  2          0.6               1.00      100       0.3224750  0.6384832
```

```
##    0.4  2          0.6                 1.00       150       0.3252853  0.6332374
##    0.4  2          0.8                 0.50        50       0.3327715  0.6205480
##    0.4  2          0.8                 0.50       100       0.3405982  0.6085293
##    0.4  2          0.8                 0.50       150       0.3460331  0.6002101
##    0.4  2          0.8                 0.75        50       0.3270486  0.6310318
##    0.4  2          0.8                 0.75       100       0.3300480  0.6273127
##    0.4  2          0.8                 0.75       150       0.3329577  0.6228018
##    0.4  2          0.8                 1.00        50       0.3227387  0.6377379
##    0.4  2          0.8                 1.00       100       0.3251971  0.6340727
##    0.4  2          0.8                 1.00       150       0.3275663  0.6299797
##    0.4  3          0.6                 0.50        50       0.3458639  0.5949823
##    0.4  3          0.6                 0.50       100       0.3562958  0.5790781
##    0.4  3          0.6                 0.50       150       0.3574546  0.5788341
##    0.4  3          0.6                 0.75        50       0.3289164  0.6291671
##    0.4  3          0.6                 0.75       100       0.3350389  0.6175949
##    0.4  3          0.6                 0.75       150       0.3373865  0.6132860
##    0.4  3          0.6                 1.00        50       0.3226524  0.6395880
##    0.4  3          0.6                 1.00       100       0.3251702  0.6357800
##    0.4  3          0.6                 1.00       150       0.3267640  0.6326968
##    0.4  3          0.8                 0.50        50       0.3530398  0.5808084
##    0.4  3          0.8                 0.50       100       0.3588508  0.5720844
##    0.4  3          0.8                 0.50       150       0.3598958  0.5713289
##    0.4  3          0.8                 0.75        50       0.3299060  0.6280591
##    0.4  3          0.8                 0.75       100       0.3359804  0.6179190
##    0.4  3          0.8                 0.75       150       0.3376207  0.6155715
##    0.4  3          0.8                 1.00        50       0.3221263  0.6399295
##    0.4  3          0.8                 1.00       100       0.3253440  0.6338670
##    0.4  3          0.8                 1.00       150       0.3262925  0.6324997
##    MAE
##    0.2633640
##    0.2603171
##    0.2603845
##    0.2610766
##    0.2590276
##    0.2596427
##    0.2636127
##    0.2584312
##    0.2571132
##    0.2624094
##    0.2616045
##    0.2615168
##    0.2610343
##    0.2591480
##    0.2594892
##    0.2618004
##    0.2575032
##    0.2568376
```

```
## 0.2529613
## 0.2527136
## 0.2552934
## 0.2403881
## 0.2412629
## 0.2442970
## 0.2426018
## 0.2406064
## 0.2417833
## 0.2506950
## 0.2537268
## 0.2594871
## 0.2436880
## 0.2440602
## 0.2461886
## 0.2415202
## 0.2409330
## 0.2423271
## 0.2542300
## 0.2580360
## 0.2600106
## 0.2442238
## 0.2466920
## 0.2475668
## 0.2396174
## 0.2409338
## 0.2420953
## 0.2519150
## 0.2568105
## 0.2608309
## 0.2453160
## 0.2479695
## 0.2502069
## 0.2432317
## 0.2450319
## 0.2468421
## 0.2672052
## 0.2678211
## 0.2701550
## 0.2629090
## 0.2597291
## 0.2611474
## 0.2638241
## 0.2606790
## 0.2601122
## 0.2636715
## 0.2631573
```

```
##   0.2696653
##   0.2625938
##   0.2612785
##   0.2625051
##   0.2618128
##   0.2584927
##   0.2583001
##   0.2626486
##   0.2663857
##   0.2731184
##   0.2545529
##   0.2593342
##   0.2620724
##   0.2482634
##   0.2478844
##   0.2497067
##   0.2582444
##   0.2642135
##   0.2687524
##   0.2518088
##   0.2543809
##   0.2565896
##   0.2486644
##   0.2509991
##   0.2526135
##   0.2687348
##   0.2774802
##   0.2800297
##   0.2554903
##   0.2605910
##   0.2620428
##   0.2483815
##   0.2507025
##   0.2521484
##   0.2739793
##   0.2800020
##   0.2812212
##   0.2540091
##   0.2579277
##   0.2593607
##   0.2497591
##   0.2531979
##   0.2539270
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##  parameter 'min_child_weight' was held constant at a value of 1
```

```
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nrounds = 50, max_depth = 3, eta
##  = 0.3, gamma = 0, colsample_bytree = 0.6, min_child_weight = 1 and subsample
##  = 1.
```

```
fit_pls
```

```
## Partial Least Squares
##
## 677 samples
##  35 predictor
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 544, 541, 541, 541, 541, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE       Rsquared   MAE
##   1      0.4050463  0.4304784  0.3127297
##   2      0.3440625  0.5885271  0.2631066
##   3      0.3392788  0.6003989  0.2611801
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 3.
```

```
fit_knn
```

```
## k-Nearest Neighbors
##
## 677 samples
##  35 predictor
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 541, 543, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   k  RMSE       Rsquared   MAE
##   5  0.3614280  0.5483458  0.2843570
##   7  0.3571253  0.5656164  0.2822834
##   9  0.3567971  0.5738453  0.2825517
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```

When trained, tuned, and evaluated the best model is The pairwise interactions tuned through elastic net.

Part 3 - Classification Part A

Creating binary outcome column based on outcome column

```
df['numeric_outcome'] <- ifelse(df$outcome == 'event', 1, 0)
df_class <- df %>%
  mutate(outcome = factor(outcome, levels = c("event", "non_event"))) %>%
  mutate(numeric_outcome = factor(numeric_outcome, levels = c(1,0)))
```

```
glm_categorical <- glm(outcome ~ region + customer + outcome , data = df_class, famil
y = "binomial")
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 3 in
## model.matrix: no columns are assigned
```

With a model just containing the categorical variables of customer and region, the variables that are statistically significant are regionZZ, customerB, customerD, customerE, customerOther.

Continous Variables Only - Linear Additive

```
glm_continous <- glm(outcome ~ xb_01 + xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + xa_01
+ xa_02 + xa_03 +xb_04 + xb_05
                    + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 +
xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df_class, family = "binomial")
```

Most of the continuous inputs are not statistically significant to the binary outcome. The inputs that are considered to be significant are xn_03, xa_01, xb_07, xn_04, xn_05,xn_07, xn_08, xa_05, xw_03 .The intercept of the model is also significant.

All Categorical + Continous inputs - Linear Additive

```
glm_cat_cont <- glm(outcome ~ region + customer + outcome +  xb_01 + xb_02 + xb_03 +
xn_01 + xn_02 + xn_03 + xa_01 + xa_02 + xa_03 +xb_04 + xb_05 + xb_06 + xb_07 + xb_08
+ xn_04 + xn_05 + xn_06 + xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df_class, family = "binomial")
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 3 in
## model.matrix: no columns are assigned
```

Interacting Categorical Input ‘Region’ with all continious inputs

```
glm_interact_region <- glm(outcome ~ region*xb_01 + region*xb_02 + region*xb_03 + reg
ion*xn_01 + region*xn_02 + region*xn_03 +
                      region*xa_01 + region*xa_02 + region*xa_03 +region*xb_04 + regi
on*xb_05+
                      region*xb_06 + region*xb_07 + region*xb_08 + region*xn_04 + reg
ion*xn_05 + region*xn_06 + region*xn_07 +
                      region*xn_08 + region*xa_04 + region*xa_05 + region*xa_06 + reg
ion*xa_07 + region*xa_08 +
                      region*xw_01 + region*xw_02 + region*xw_03 + region*xs_01 + reg
ion*xs_02 + region*xs_03 + region*xs_04 +
                      region*xs_05 + region*xs_06, data = df_class, family = "binomia
l")
```

Interacting Categorical Input ‘Customer’ with all continuous inputs

```
glm_interact_customer <- glm(outcome ~ customer*xb_01 + customer*xb_02 + customer*xb_
03 + customer*xn_01 + customer*xn_02 + customer*xn_03 +
                      customer*xa_01 + customer*xa_02 + customer*xa_03 +customer*xb_0
4 + customer*xb_05+
                      customer*xb_06 + customer*xb_07 + customer*xb_08 + customer*xn_
04 + customer*xn_05 + customer*xn_06 + customer*xn_07 +
                      customer*xn_08 + customer*xa_04 + customer*xa_05 + customer*xa_
06 + customer*xa_07 + customer*xa_08 +
                      customer*xw_01 + customer*xw_02 + customer*xw_03 + customer*xs_
01 + customer*xs_02 + customer*xs_03 + customer*xs_04 +
                      customer*xs_05 + customer*xs_06, data = df_class, family = "bin
omial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Examining all pairwise interactions terms of the continuous inputs

```
df_continuous <- df_continuous[-length(df_continuous)]
df_continuous['numeric_outcome'] <- df$numeric_outcome
df_continuous['log_response'] <- df$log_response
```

```
glm_pairwise_cont <- glm (numeric_outcome ~ (.)^2, data = df_continuous, family = "bi
nomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Fitting three other basis functions

Fitting a spline model with continious xb_03 of 15 df

```
df_continuous['outcome'] <- df_class$outcome
glm_spline <- glm(outcome ~ splines::ns(xw_03, 15), data = df_continuous, family = "b
inomial" )
```

The input xb_07 seemed be to statistically significant in the previous linear models.

Fitting linear model with continuous inputs and their qudratic features.

```
glm_quadratic <- glm(outcome ~  xb_01 + xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + xa_01
+ xa_02 + xa_03 +xb_04 + xb_05 + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 + xn_0
7 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 + xw_01 + xw_02 + xw_03 + xs_01 + x
s_02 + xs_03 + xs_04 + xs_05 + xs_06 + I(xb_01^2) + I(xb_02^2) + I(xb_03^2) + I(xn_01
^2) + I(xn_02^2) + I(xn_03^2) + I(xa_01^2) + I(xa_02^2) + I(xa_03^2) + I(xb_04^2) + I
(xb_05^2) + I(xb_06^2) + I(xb_07^2) + I(xb_08^2) + I(xn_04^2) + I(xn_05^2) + I(xn_06^
2) + I(xn_07^2) + I(xn_08^2) + I(xa_04^2) + I(xa_05^2) + I(xa_06^2) + I(xa_07^2) + I(
xa_08^2) + I(xw_01^2) + I(xw_02^2) + I(xw_03^2) + I(xs_01^2) + I(xs_02^2) + I(xs_03^2
) + I(xs_04^2) + I(xs_05^2) + I(xs_06), data = df_continuous, family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Interacting linear and quadratic of three statistically significant variables from continous only linear additive model

```
glm_three_signif <- glm (outcome ~ (xw_03 + I(xw_03^2))*(xn_07 + I(xn_07^2)) * (xn_08
+ I(xn_08^2)), data = df_continuous, family = "binomial" )
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

These models are mostly consistent with the regression portion, with some varying differences in variable significance. THere were some warning messages that fitted probabilities were numerically 0 or 1 occurred.

Extracting Metrics from the 9 GLM models to determine the best models.

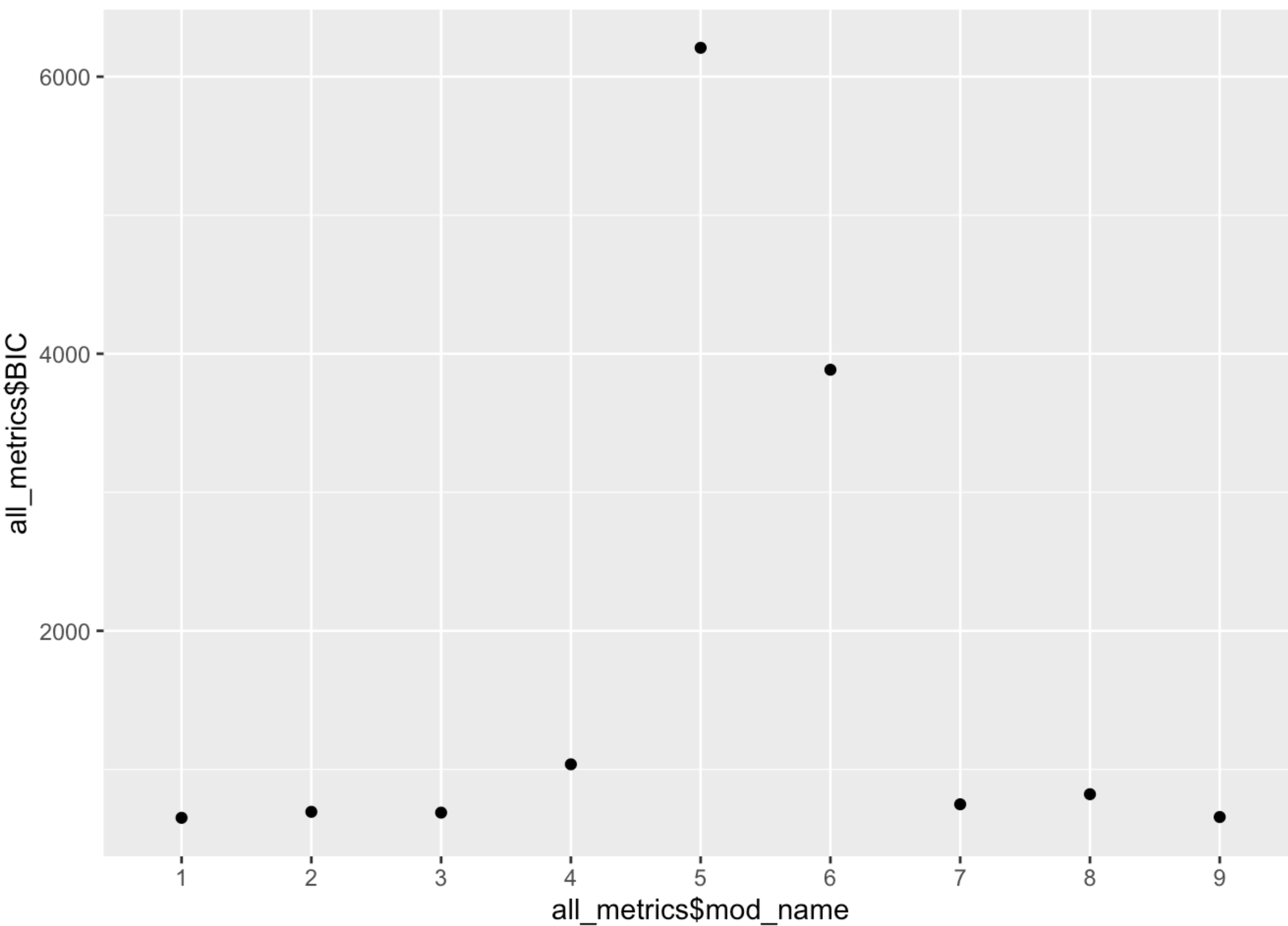
```
extract_metrics <- function (mod, mod_name)
{
  broom::glance(mod) %>% mutate(mod_name = mod_name)
}
```

```
all_metrics <- purrr::map2_dfr(list(glm_continous, glm_categorical, glm_cat_cont, glm
_interact_region, glm_interact_customer,glm_pairwise_cont, glm_spline, glm_quadratic,
glm_three_signif), as.character(1:9), extract_metrics)

all_metrics
```

```
## # A tibble: 9 × 9
##   null.deviance df.null   logLik   AIC   BIC deviance df.residual  nobs mod_name
##           <dbl>   <int>    <dbl> <dbl> <dbl>    <dbl>       <int> <int> <chr>
## 1          654.     676 -2.14e+2  496.  650.  4.28e+2         643   677 1
## 2          654.     676 -3.11e+2  644.  693.  6.22e+2         666   677 2
## 3          654.     676 -2.00e+2  489.  688.  4.01e+2         633   677 3
## 4          654.     676 -1.86e+2  575. 1036.  3.71e+2         575   677 4
## 5          654.     676 -2.13e+3 4853. 6208.  4.25e+3         377   677 5
## 6          654.     676 -2.96e-9 1192. 3885.  5.92e-9          81   677 6
## 7          654.     676 -3.22e+2  675.  748.  6.43e+2         661   677 7
## 8          654.     676 -1.95e+2  522.  821.  3.90e+2         611   677 8
## 9          654.     676 -2.40e+2  534.  656.  4.80e+2         650   677 9
```

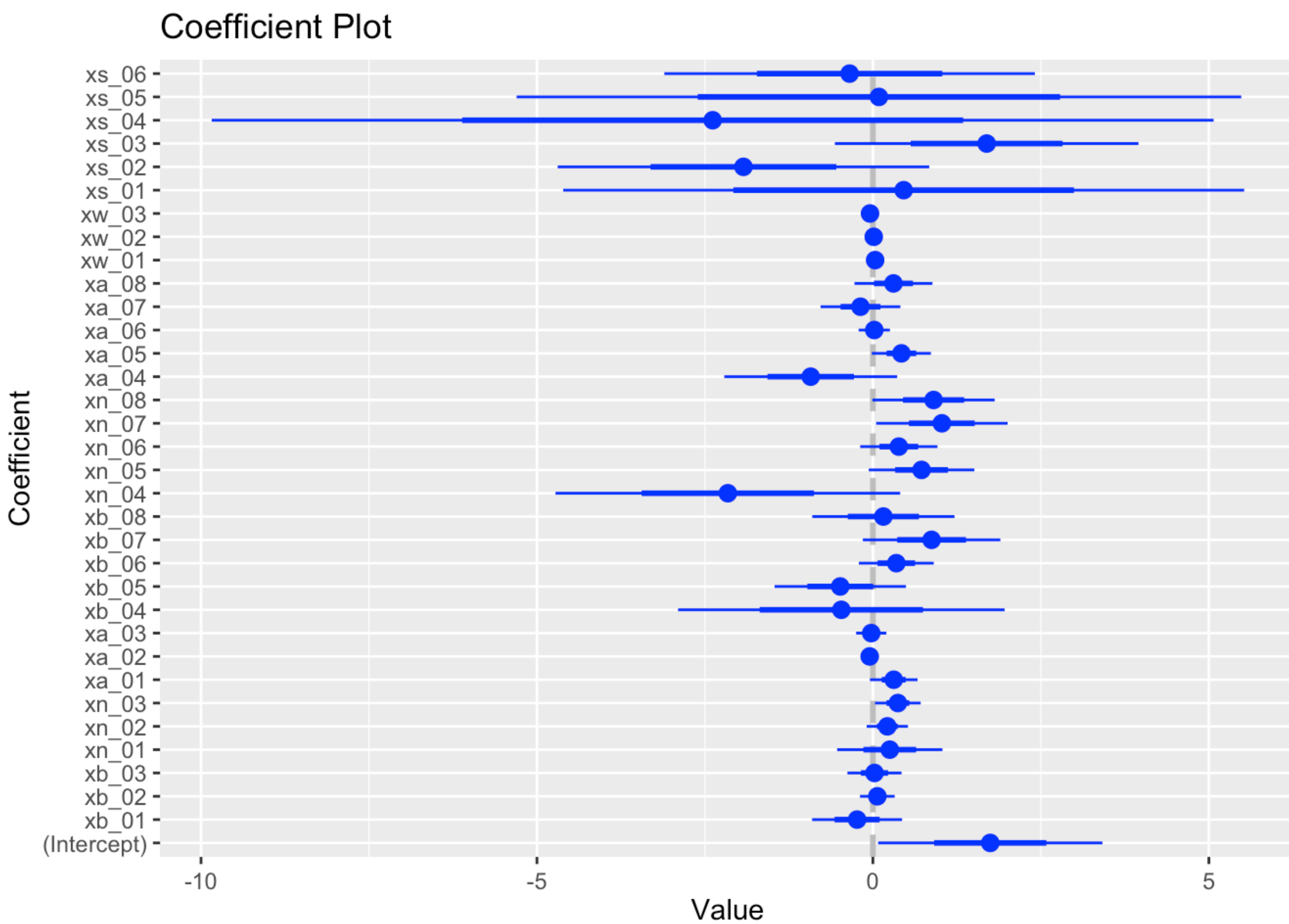
Using BIC again, the best model is model 1 – model with all continuous inputs. The top three models are model 1, 9, 3 (glm_continuous, glm_three_signif, glm_cat_cont)

```
ggplot(mapping = aes(y = all_metrics$BIC, x = all_metrics$mod_name)) + geom_point()
```



Visualizing the coefficient summaries of the top three models.

```
glm_continous %>% coefplot::coefplot()
```

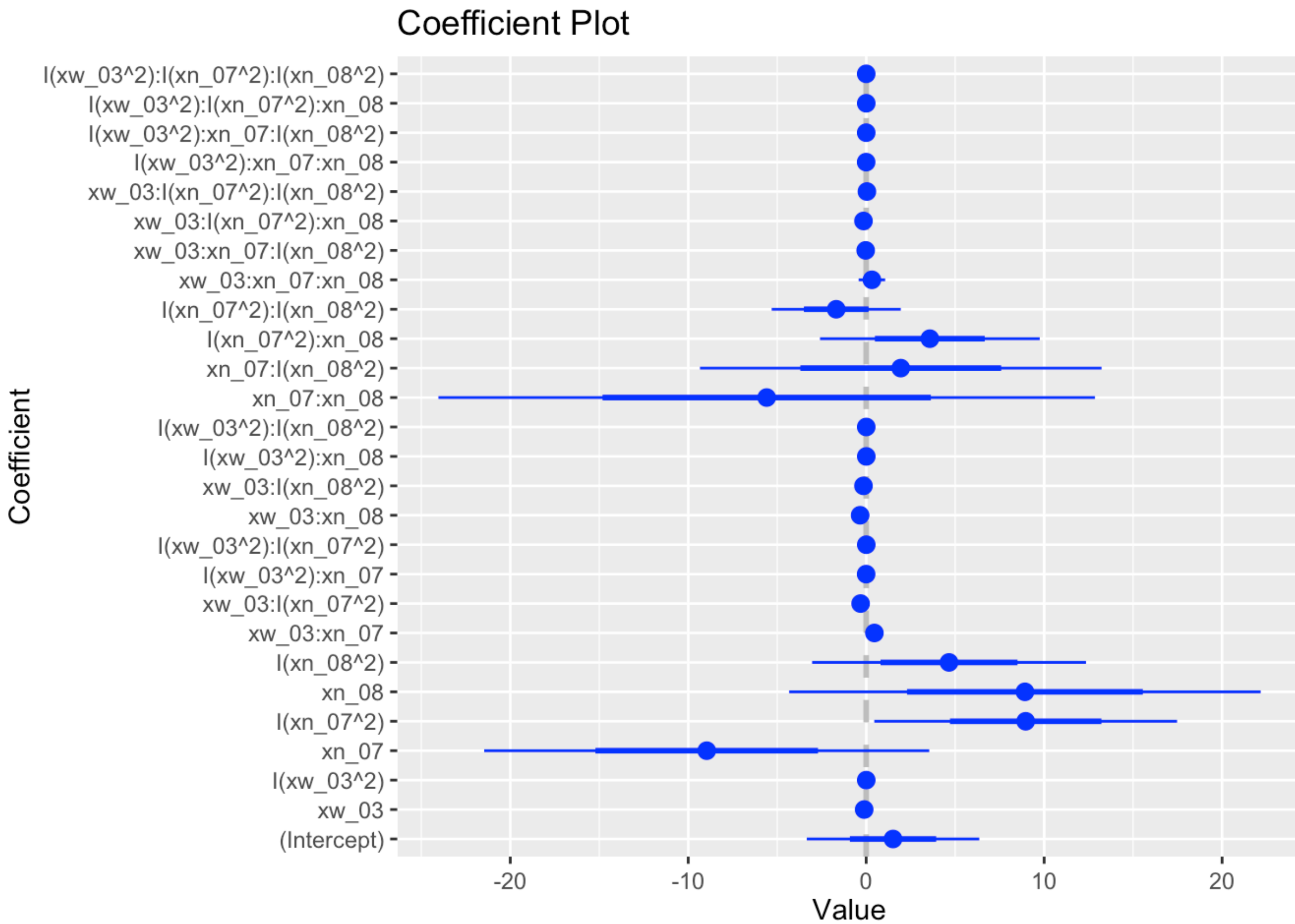



```
glm_continous %>% summary()
```

```
##
## Call:
## glm(formula = outcome ~ xb_01 + xb_02 + xb_03 + xn_01 + xn_02 +
##     xn_03 + xa_01 + xa_02 + xa_03 + xb_04 + xb_05 + xb_06 + xb_07 +
##     xb_08 + xn_04 + xn_05 + xn_06 + xn_07 + xn_08 + xa_04 + xa_05 +
##     xa_06 + xa_07 + xa_08 + xw_01 + xw_02 + xw_03 + xs_01 + xs_02 +
##     xs_03 + xs_04 + xs_05 + xs_06, family = "binomial", data = df_class)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.53568   0.04411   0.25432   0.54178   2.29122
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  1.74664    0.83452   2.093  0.03635 *
## xb_01       -0.23535    0.33484  -0.703  0.48213
## xb_02        0.06593    0.12905   0.511  0.60943
## xb_03        0.02335    0.20140   0.116  0.90770
## xn_01        0.25124    0.39183   0.641  0.52140
## xn_02        0.21489    0.15341   1.401  0.16130
## xn_03        0.37185    0.16961   2.192  0.02835 *
## xa_01        0.30987    0.17743   1.746  0.08074 .
## xa_02       -0.04805    0.06648  -0.723  0.46984
## xa_03       -0.02498    0.11263  -0.222  0.82445
## xb_04       -0.46985    1.21480  -0.387  0.69892
## xb_05       -0.48612    0.48901  -0.994  0.32018
## xb_06        0.34854    0.27853   1.251  0.21080
## xb_07        0.87367    0.51166   1.708  0.08773 .
## xb_08        0.15574    0.52919   0.294  0.76853
## xn_04       -2.15908    1.28253  -1.683  0.09229 .
## xn_05        0.72395    0.39263   1.844  0.06521 .
## xn_06        0.38584    0.28749   1.342  0.17955
## xn_07        1.02626    0.48923   2.098  0.03593 *
## xn_08        0.90325    0.45462   1.987  0.04694 *
## xa_04       -0.92562    0.64324  -1.439  0.15015
## xa_05        0.42427    0.21961   1.932  0.05337 .
## xa_06        0.02034    0.11627   0.175  0.86114
## xa_07       -0.18466    0.29681  -0.622  0.53385
## xa_08        0.30580    0.29036   1.053  0.29227
## xw_01        0.03307    0.02655   1.245  0.21304
## xw_02        0.01233    0.01456   0.847  0.39725
## xw_03       -0.04307    0.01537  -2.802  0.00509 **
## xs_01        0.45776    2.53394   0.181  0.85664
## xs_02       -1.92673    1.38234  -1.394  0.16337
## xs_03        1.69319    1.12978   1.499  0.13395
## xs_04       -2.38492    3.72792  -0.640  0.52234
## xs_05        0.08946    2.69624   0.033  0.97353
## xs_06       -0.34687    1.37884  -0.252  0.80138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 653.59  on 676  degrees of freedom
## Residual deviance: 428.46  on 643  degrees of freedom
## AIC: 496.46
##
## Number of Fisher Scoring iterations: 7
```

```
glm_three_signif %>% coefplot::coefplot()
```

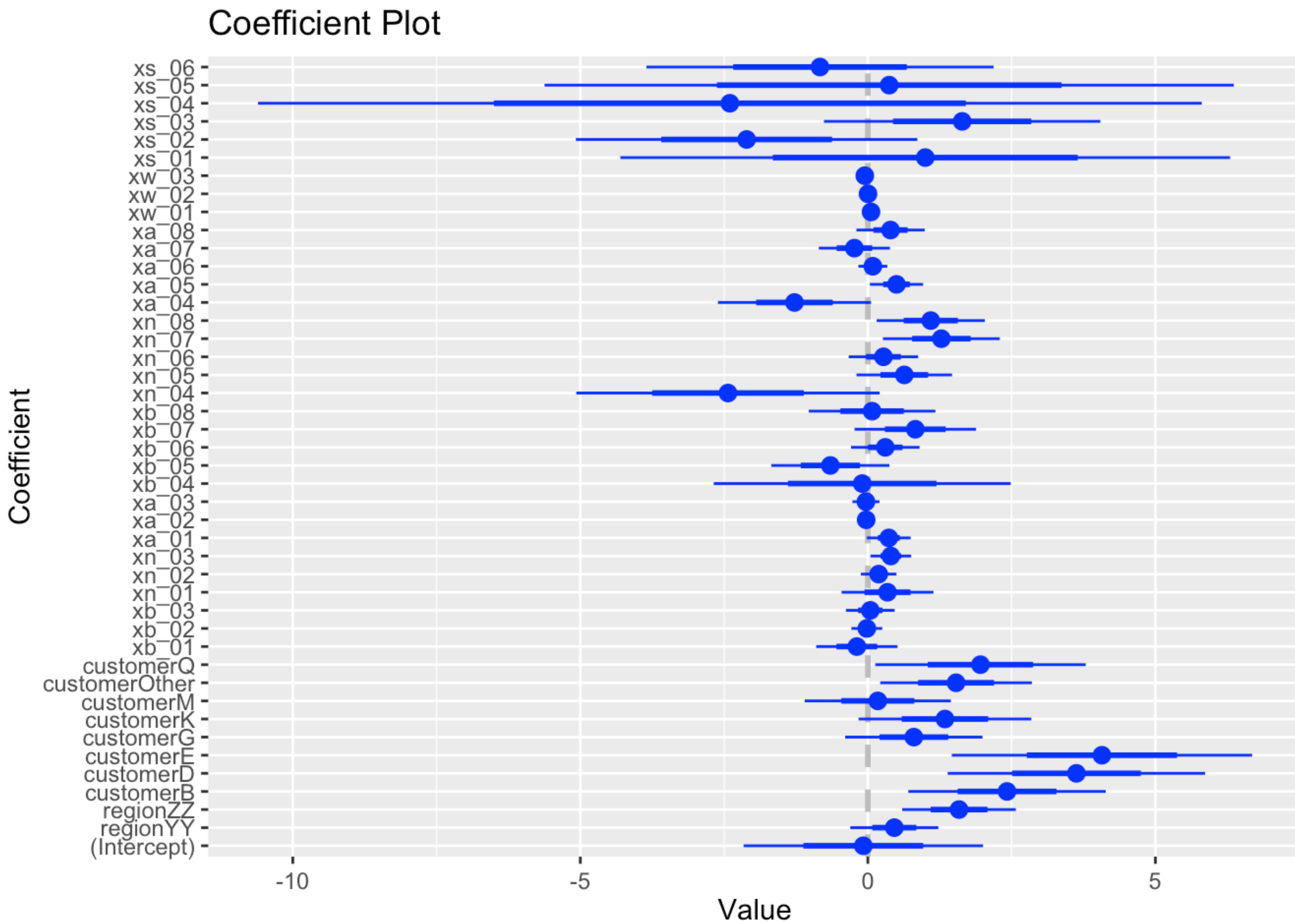



```
glm_three_signif %>% summary()
```

```
##
## Call:
## glm(formula = outcome ~ (xw_03 + I(xw_03^2)) * (xn_07 + I(xn_07^2)) *
##     (xn_08 + I(xn_08^2)), family = "binomial", data = df_continuous)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.45078   0.00251   0.31577   0.64926   1.85648
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       1.5079899  2.4239212   0.622  0.53386
## xw_03                            -0.1133387  0.1183186  -0.958  0.33811
## I(xw_03^2)                        0.0016811  0.0012002   1.401  0.16132
## xn_07                            -8.9612680  6.2522200  -1.433  0.15177
## I(xn_07^2)                        8.9613940  4.2538033   2.107  0.03515 *
## xn_08                             8.9198798  6.6238457   1.347  0.17810
```

```
## I(xn_08^2)                         4.6541940  3.8476501   1.210  0.22642
## xw_03:xn_07                        0.4596920  0.2346506   1.959  0.05011 .
## xw_03:I(xn_07^2)                  -0.3155133  0.1321130  -2.388  0.01693 *
## I(xw_03^2):xn_07                  -0.0048550  0.0021298  -2.280  0.02263 *
## I(xw_03^2):I(xn_07^2)              0.0028131  0.0010516   2.675  0.00747 **
## xw_03:xn_08                       -0.3436373  0.2511866  -1.368  0.17129
## xw_03:I(xn_08^2)                  -0.1533256  0.1394335  -1.100  0.27149
## I(xw_03^2):xn_08                   0.0034004  0.0021444   1.586  0.11280
## I(xw_03^2):I(xn_08^2)              0.0012730  0.0010982   1.159  0.24638
## xn_07:xn_08                       -5.5922877  9.2228273  -0.606  0.54428
## xn_07:I(xn_08^2)                   1.9416192  5.6414443   0.344  0.73072
## I(xn_07^2):xn_08                   3.5760309  3.0865240   1.159  0.24662
## I(xn_07^2):I(xn_08^2)             -1.6844852  1.8139305  -0.929  0.35308
## xw_03:xn_07:xn_08                  0.3202394  0.3727563   0.859  0.39028
## xw_03:xn_07:I(xn_08^2)            -0.0303842  0.2138368  -0.142  0.88701
## xw_03:I(xn_07^2):xn_08            -0.1523356  0.1376059  -1.107  0.26828
## xw_03:I(xn_07^2):I(xn_08^2)        0.0388796  0.0745086   0.522  0.60180
## I(xw_03^2):xn_07:xn_08            -0.0039410  0.0032972  -1.195  0.23198
## I(xw_03^2):xn_07:I(xn_08^2)       -0.0002030  0.0017093  -0.119  0.90549
## I(xw_03^2):I(xn_07^2):xn_08        0.0017026  0.0012913   1.319  0.18731
## I(xw_03^2):I(xn_07^2):I(xn_08^2) -0.0000926  0.0006208  -0.149  0.88142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 653.59  on 676  degrees of freedom
## Residual deviance: 479.88  on 650  degrees of freedom
## AIC: 533.88
##
## Number of Fisher Scoring iterations: 13
```

```
glm_cat_cont %>% coefplot::coefplot()
```




```
glm_cat_cont %>% summary()
```

```
##
## Call:
## glm(formula = outcome ~ region + customer + outcome + xb_01 +
##     xb_02 + xb_03 + xn_01 + xn_02 + xn_03 + xa_01 + xa_02 + xa_03 +
##     xb_04 + xb_05 + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 + xn_06 +
##     xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 + xw_01 +
##     xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 + xs_06,
##     family = "binomial", data = df_class)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.72956   0.02986   0.19193   0.48775   2.51713
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.079176   1.040892  -0.076  0.93937
## regionYY       0.459934   0.382696   1.202  0.22943
```

```
## regionZZ       1.585516   0.493926   3.210  0.00133 **
## customerB      2.420042   0.858364   2.819  0.00481 **
## customerD      3.626300   1.118994   3.241  0.00119 **
## customerE      4.070174   1.305794   3.117  0.00183 **
## customerG      0.799423   0.597087   1.339  0.18061
## customerK      1.340608   0.751047   1.785  0.07426 .
## customerM      0.172952   0.635200   0.272  0.78541
## customerOther  1.533744   0.659114   2.327  0.01997 *
## customerQ      1.958051   0.915096   2.140  0.03238 *
## xb_01         -0.191744   0.353659  -0.542  0.58770
## xb_02         -0.018087   0.134240  -0.135  0.89282
## xb_03          0.042694   0.211803   0.202  0.84025
## xn_01          0.341512   0.399361   0.855  0.39247
## xn_02          0.188469   0.154751   1.218  0.22327
## xn_03          0.399131   0.177306   2.251  0.02438 *
## xa_01          0.364592   0.190021   1.919  0.05502 .
## xa_02         -0.028789   0.071942  -0.400  0.68903
## xa_03         -0.033717   0.117716  -0.286  0.77455
## xb_04         -0.098036   1.291060  -0.076  0.93947
## xb_05         -0.652151   0.513553  -1.270  0.20413
## xb_06          0.302941   0.298214   1.016  0.30970
## xb_07          0.825085   0.528613   1.561  0.11856
## xb_08          0.073412   0.550694   0.133  0.89395
## xn_04         -2.432393   1.318686  -1.845  0.06510 .
## xn_05          0.633605   0.415522   1.525  0.12730
## xn_06          0.271087   0.301647   0.899  0.36882
## xn_07          1.278058   0.508250   2.515  0.01192 *
## xn_08          1.093718   0.470628   2.324  0.02013 *
## xa_04         -1.276047   0.665753  -1.917  0.05528 .
## xa_05          0.497786   0.231415   2.151  0.03147 *
## xa_06          0.086830   0.125676   0.691  0.48963
## xa_07         -0.234222   0.309224  -0.757  0.44878
## xa_08          0.394160   0.296777   1.328  0.18413
## xw_01          0.054832   0.029602   1.852  0.06398 .
## xw_02          0.002573   0.015807   0.163  0.87069
## xw_03         -0.053508   0.016964  -3.154  0.00161 **
## xs_01          0.998168   2.650967   0.377  0.70652
## xs_02         -2.108231   1.484366  -1.420  0.15552
## xs_03          1.639712   1.201712   1.364  0.17242
## xs_04         -2.399347   4.102694  -0.585  0.55867
## xs_05          0.370962   2.996362   0.124  0.90147
## xs_06         -0.832273   1.509090  -0.552  0.58129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 653.59  on 676  degrees of freedom
## Residual deviance: 400.78  on 633  degrees of freedom
## AIC: 488.78
##
## Number of Fisher Scoring iterations: 7
```

In the models with the continuous inputs, xw_03 is significant. Region ZZ, customerB, customerD, customerE ar significant in the models that contain the categorical inputs.

xw_03, Region ZZ, customerB, customerD, customerE seem to be important.

Classification Part B

Fitting Bayesian Genralized Linear Models

Model 1: glm_continuous Model 2: glm_cat_cont The reason for picking the second model is, I want to see if the categorical inputs are necessary when compared to the model with only continuous inputs and I also want to compare it to the linear regression model.

```
logistic_logpost <- function(unknowns, my_info)
{
  X <- my_info$design_matrix

  length_beta <- ncol(my_info$design_matrix)
  beta_v <- unknowns[1:length_beta]


  eta <- as.vector(X %*% as.matrix(beta_v))

  mu <- boot::inv.logit(eta)

  log_lik <- sum(dbinom(x = my_info$yobs,
               size = 1,
               prob = mu,
               log = TRUE))

  log_prior <- sum(dnorm(x =beta_v,
                         mean = my_info$mu_beta,
                         sd = my_info$tau_beta,
                         log = TRUE))

  log_lik + log_prior
}
```

Create Design Matrix for the two models

```
Xmat_cat_cont<-model.matrix( outcome~ region + customer  +  xb_01 + xb_02 + xb_03 + x
n_01 + xn_02 + xn_03 + xa_01 + xa_02 + xa_03 +xb_04 + xb_05 + xb_06 + xb_07 + xb_08 +
xn_04 + xn_05 + xn_06 + xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df)

Xmat_continuous <- model.matrix( outcome ~ xb_01 + xb_02 + xb_03 + xn_01 + xn_02 + xn
_03 + xa_01 + xa_02 + xa_03 +xb_04 + xb_05 + xb_06 + xb_07 + xb_08 + xn_04 + xn_05 +
xn_06 + xn_07 + xn_08 + xa_04 + xa_05 + xa_06 + xa_07 + xa_08 +
                      xw_01 + xw_02 + xw_03 + xs_01 + xs_02 + xs_03 + xs_04 + xs_05 +
xs_06, data = df)
```

```
df['numeric_outcome'] <- ifelse(df$outcome == 'event', 1, 0)
cat_cont_info <- list(
  yobs = df$numeric_outcome,
  design_matrix = Xmat_cat_cont,
  mu_beta = 0,
  tau_beta = 1
)

cont_info <- list(
  yobs = df$numeric_outcome,
  design_matrix = Xmat_continuous,
  mu_beta = 0,
  tau_beta = 1
)
```

```
my_laplace <- function(start_guess, logpost_func, ...)
{
  # code adapted from the `LearnBayes`` function `laplace()`
  fit <- optim(start_guess,
               logpost_func,
               gr = NULL,
               ...,
               method = "BFGS",
               hessian = TRUE,
               control = list(fnscale = -1, maxit = 1001))

  mode <- fit$par
  post_var_matrix <- -solve(fit$hessian)
  p <- length(mode)
  int <- p/2 * log(2 * pi) + 0.5 * log(det(post_var_matrix)) + logpost_func(mode, ...
)
  # package all of the results into a list
  list(mode = mode,
       var_matrix = post_var_matrix,
       log_evidence = int,
       converge = ifelse(fit$convergence == 0,
                         "YES",
                         "NO"),
       iter_counts = as.numeric(fit$counts[1]))
}
```

Execute the Laplace Approximation to fit the glm models

```
laplace_cat_cont <- my_laplace(rep(0,ncol(Xmat_cat_cont)), logistic_logpost, cat_cont
_info)
laplace_cont <- my_laplace(rep(0,ncol(Xmat_continuous)), logistic_logpost, cont_info)
```

Determining which model is best using Bayes Factor.

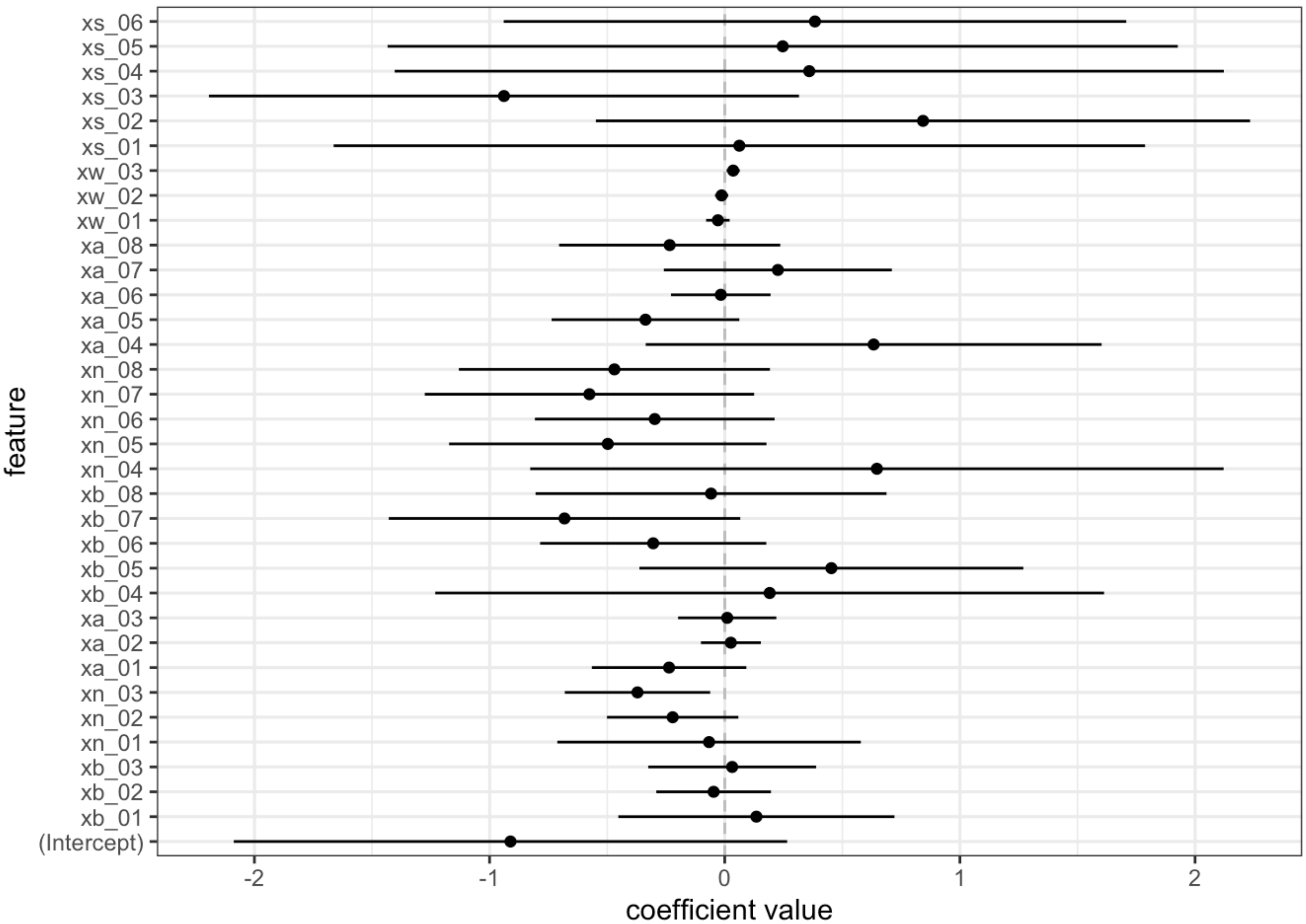
```
exp(laplace_cat_cont$log_evidence) / exp(laplace_cont$log_evidence)
```

```
## [1] 0.723306
```

The result of the Bayes Factor shows that there is more evidence for the continuous model, as opposed to the model with both the categorical and continuous model.

Vizualizing the regession coefficient posterior summary statsitics for the model with continuous additive inputs

```
viz_post_coefs(laplace_cont$mode[1:ncol(Xmat_continuous)],
               sqrt(diag(laplace_cont$var_matrix)[1:ncol(Xmat_continuous)]),
               colnames(Xmat_continuous))
```



Classifcation Part C

```
generate_glm_post_samples <- function(mvn_result, num_samples)
{
  length_beta <- length(mvn_result$mode)

  beta_samples <- MASS::mvrnorm(n = num_samples,
                                mu = mvn_result$mode,
                                Sigma = mvn_result$var_matrix)

  beta_samples %>%
    as.data.frame() %>% tibble::as_tibble() %>%
    purrr::set_names(sprintf("beta_%02d", (1:length_beta) - 1))
}
```

```
post_logistic_pred_samples <- function(Xnew, Bmat)
{
  eta_mat <- Xnew %*% t(Bmat)
  mu_mat <- boot::inv.logit(eta_mat)

  list(eta_mat = eta_mat, mu_mat = mu_mat)
}
```

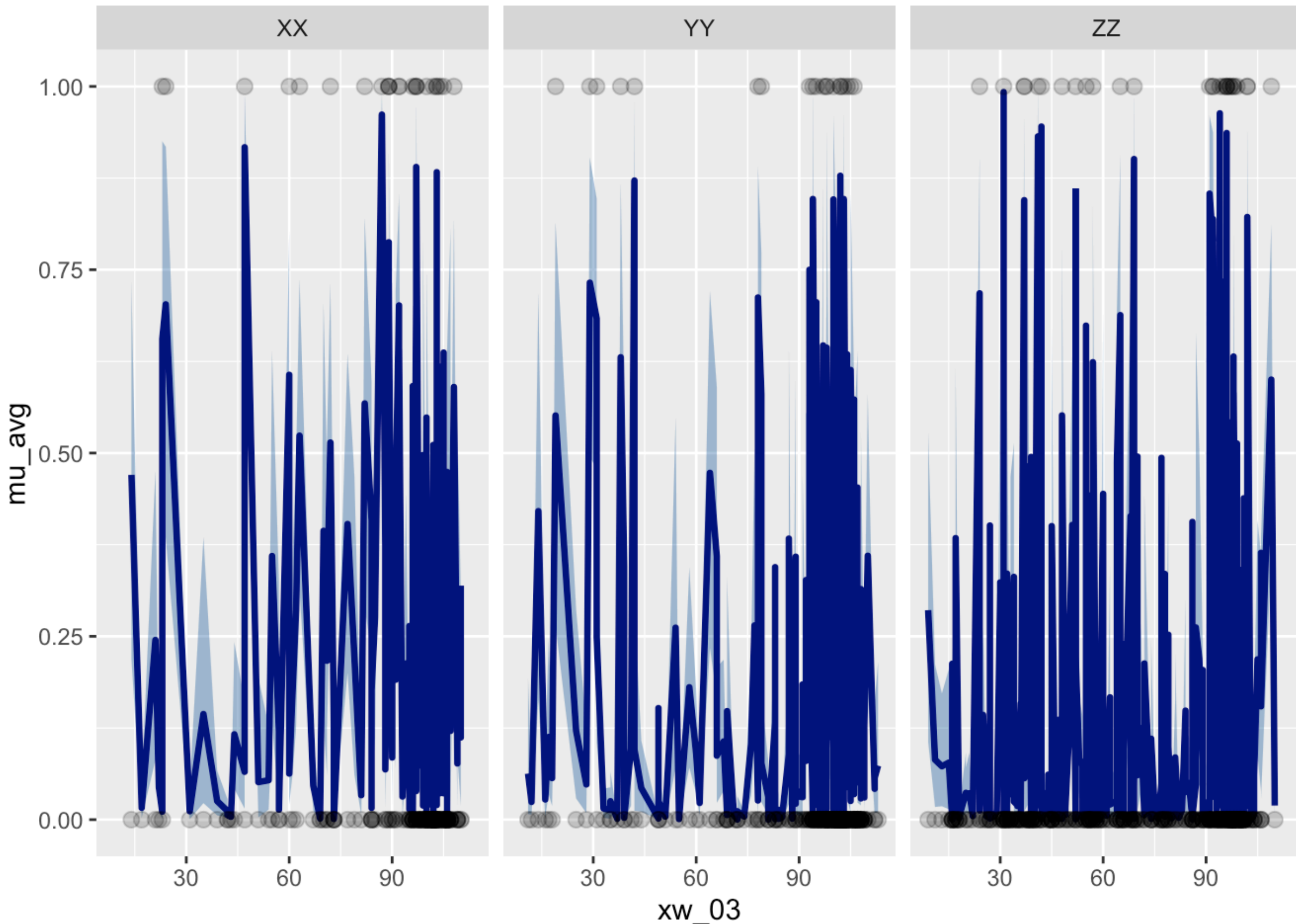
```
summarize_logistic_pred_from_laplace <- function(mvn_result, Xtest, num_samples)
{
  betas <- generate_glm_post_samples(mvn_result, num_samples)

  betas <- as.matrix(betas)

  pred_test <- post_logistic_pred_samples(Xtest, betas)

  mu_avg <- rowMeans(pred_test$mu_mat)

  mu_q05 <- apply(pred_test$mu_mat, 1, stats::quantile, probs = 0.05)
  mu_q95 <- apply(pred_test$mu_mat, 1, stats::quantile, probs = 0.95)

  tibble::tibble(
    mu_avg = mu_avg,
    mu_q05 = mu_q05,
    mu_q95 = mu_q95
  ) %>%
    tibble::rowid_to_column("pred_id")
}
```

Make predictions on generated posterior samples and summarize

```
continuous_post_pred <- summarize_logistic_pred_from_laplace(laplace_cont, Xmat_conti
nuous, 2500)
cat_cont_post_pred <- summarize_logistic_pred_from_laplace(laplace_cat_cont, Xmat_cat
_cont, 2500)
```

Visualize predictive trends

```
cat_cont_post_pred %>%
  left_join(df %>% tibble::rowid_to_column("pred_id"),
            by = "pred_id") %>%
  mutate(event_prob = ifelse(mu_avg > 0.5, 1, 0)) %>%
  ggplot(mapping = aes(x = xw_03)) +
  geom_ribbon(mapping = aes(ymin = mu_q05,
                            ymax = mu_q95),
              fill = "steelblue", alpha = 0.5) +
  geom_line(mapping = aes(y = mu_avg),
            color = "navyblue", size = 1.15) +
  geom_point(mapping = aes(y = event_prob),
             size = 2.5, alpha = 0.2) +
  facet_grid(~region)
```




```
theme_bw()
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
```

```
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : NULL
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
```

```
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
```

```
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
```

```
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                :List of 6
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.
5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
```

```
##  $ legend.spacing.y          : NULL
##  $ legend.key                :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.key.size           : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
```

```
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.border              :List of 5
##   ..$ fill         : logi NA
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : chr "white"
```

```
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

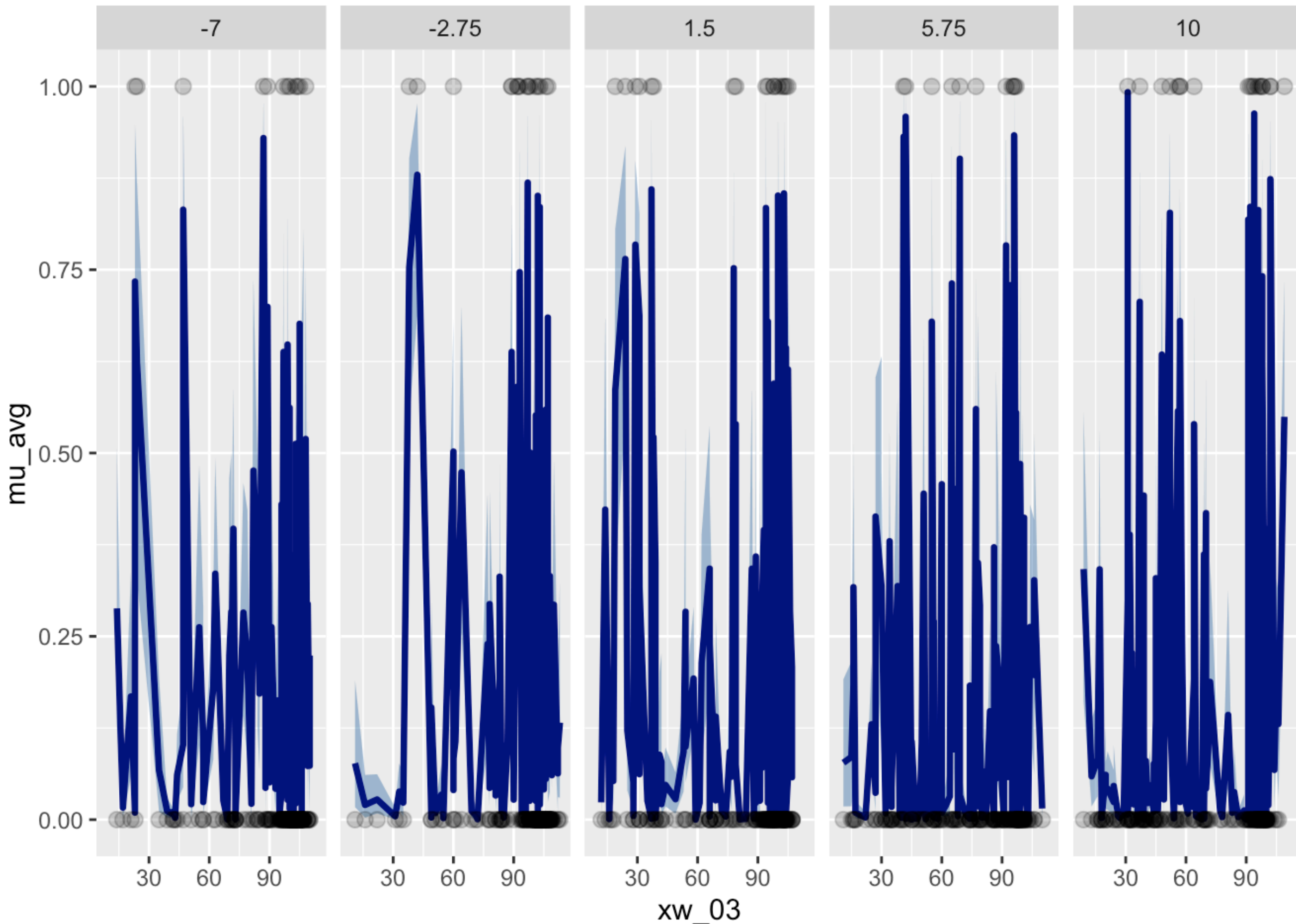
```
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.
5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          :List of 5
##   ..$ fill         : chr "grey85"
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
```

```
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

```
xn_03_vals <- pred_cont['xn_03']
min <- min(xn_03_vals)
max <- max(xn_03_vals)
mean <- (min + max) /2


med_0.25 <- (min + mean) /2
med_0.75 <- (mean + max) /2
levels <- c(min, med_0.25, mean, med_0.75, max)
levels_spaced <- rep(levels, each = nrow(df)/5)
levels_spaced[length(levels_spaced) + 1]  = levels_spaced[length(levels_spaced) ]
levels_spaced[length(levels_spaced) + 1]  = levels_spaced[length(levels_spaced) ]
continuous_post_pred['levels'] = as.factor(levels_spaced)
continuous_post_pred
```

```
## # A tibble: 677 × 5
##    pred_id  mu_avg  mu_q05 mu_q95 levels
##      <int>   <dbl>   <dbl>  <dbl> <fct>
##  1       1 0.00958 0.00139 0.0290 -7
##  2       2 0.0164  0.00334 0.0437 -7
##  3       3 0.110   0.0209  0.289  -7
##  4       4 0.0888  0.0227  0.210  -7
##  5       5 0.369   0.144   0.640  -7
##  6       6 0.143   0.0551  0.274  -7
##  7       7 0.483   0.173   0.805  -7
##  8       8 0.0732  0.0206  0.167  -7
##  9       9 0.236   0.0715  0.483  -7
## 10      10 0.304   0.136   0.522  -7
## # … with 667 more rows
```

```
continuous_post_pred %>%
  left_join(df %>% tibble::rowid_to_column("pred_id"),
            by = "pred_id") %>%
  mutate(event_prob = ifelse(mu_avg > 0.5, 1, 0)) %>%
  ggplot(mapping = aes(x = xw_03)) +
  geom_ribbon(mapping = aes(ymin = mu_q05,
                            ymax = mu_q95,
                            group = continuous_post_pred$levels),
              fill = "steelblue", alpha = 0.5) +
  geom_line(mapping = aes(y = mu_avg,
                            group = continuous_post_pred$levels),
            color = "navyblue", size = 1.15) +
  geom_point(mapping = aes(y = event_prob,
                            group = continuous_post_pred$levels),
             size = 2.5, alpha = 0.2) +
  facet_grid(~continuous_post_pred$levels)
```




```
theme_bw()
```

```
## List of 93
##  $ line                      :List of 6
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
```

```
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                      : NULL
##  $ aspect.ratio               : NULL
##  $ axis.title                 : NULL
##  $ axis.title.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom        : NULL
##  $ axis.title.y               :List of 11
##   ..$ family       : NULL
```

```
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
```

```
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
```

```
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                :List of 6
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.
5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
```

```
##  $ legend.spacing.y          : NULL
##  $ legend.key                :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ legend.key.size           : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
```

```
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : logi NA
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.border              :List of 5
##   ..$ fill         : logi NA
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ panel.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           :List of 5
##   ..$ fill         : NULL
##   ..$ colour       : chr "white"
```

```
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ plot.title                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

```
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.
5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          :List of 5
##   ..$ fill         : chr "grey85"
##   ..$ colour       : chr "grey20"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
```

```
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

The predictive trends are somewhat consistent between the two models. When xw_03 ≥ 60 there is a higher chance that the predicted class is event. When xw_03 is between 30 and 60 it is more likely that the class predicted is a non-event.

Classification - Part D

Training, Tuning, Evaluation Generalized Linear Models

```
df_class <- df %>%
  select(region, customer, starts_with('x'), outcome)
```

```
metric_acc <- "Accuracy"

my_ctrl_acc <- trainControl(method = "repeatedcv" , number = 5, repeats = 5, metric_a
cc, classProbs = TRUE)
my_ctrl_roc <- trainControl(method = "repeatedcv" , number = 5, repeats = 5, summaryF
unction = twoClassSummary, classProbs = TRUE)
```

```
cat_cont_tune_acc <- caret::train(outcome ~  ., data = df_class, method = 'glm' , met
ric = metric_acc,
                                preProcess = c('center','scale'), trControl = my_ctrl
_acc)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
cat_cont_tune_roc <- caret::train(outcome ~  ., data = df_class, method = 'glm' ,
                                preProcess = c('center','scale'), trControl = my_ctrl
_roc)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.

## Warning in train.default(x, y, weights = w, ...): glm.fit: fitted probabilities
## numerically 0 or 1 occurred
```

All pairwise interactions of continuous inputs and additive categorical features

```
pairwise_tune_acc <- caret::train(outcome ~ (.)^2 + region + customer ,data = df_clas
s, method = 'glm', metric = metric_acc,
                        preProcess = c('center','scale'), trControl = my_ctrl_acc)
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
pairwise_tune_roc <- caret::train(outcome ~ (.)^2 + region + customer ,data = df_clas
s, method = 'glm',
                        preProcess = c('center','scale'), trControl = my_ctrl_roc)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Training the second model from Regression Part A (mod_continuous)

Second model from Part A is the same as the cat_cont_tune

```
continuous_tune_acc <- caret::train(outcome ~ .,data = df_class, method = 'glm', metr
ic = metric_acc,
                        preProcess = c('center','scale'), trControl = my_ctrl_acc)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
continuous_tune_roc <- caret::train(outcome ~ .,data = df_class, method = 'glm',
                        preProcess = c('center','scale'), trControl = my_ctrl_roc)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Regularized Logistic Regression with Elastic Net

All pairwise interactions of continuous inputs, including categorical features

```
pairwise_tune_acc_enet <- caret::train(outcome ~ (.)^2 + region + customer ,data = df
_class, method = 'glmnet', metric = metric_acc,
                        preProcess = c('center','scale'), trControl = my_ctrl_acc)
```

```
## Warning in preProcess.default(method = c("center", "scale"), x =
## structure(c(0, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
pairwise_tune_roc_enet <- caret::train(outcome ~ (.)^2 + region + customer ,data = df
_class, method = 'glmnet',
                        preProcess = c('center','scale'), trControl = my_ctrl_roc)
```

```
## Warning in preProcess.default(method = c("center", "scale"), x =
## structure(c(0, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: regionZZ:customerB,
## regionZZ:customerD, regionZZ:customerE, regionYY:customerG, regionYY:customerK,
## regionYY:customerM, regionZZ:customerQ
```

More complex of the top two models - Categorical + Continuous inputs

```
cat_cont_tune_acc_enet <- caret::train(outcome ~  ., data = df_class, method = 'glmne
t' , metric = metric_acc,
                                preProcess = c('center','scale'), trControl = my_ctrl
_acc)

cat_cont_tune_roc_enet <- caret::train(outcome ~  ., data = df_class, method = 'glmne
t' ,
                                preProcess = c('center','scale'), trControl = my_ctrl
_roc)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Using Neural Network to Train, evaluate, and tune the glm_cat_cont model.

```
set.seed(4321)

fit_nnet_acc <- train(outcome ~ .,
                        data = df_class,  method = "nnet", metric = metric_acc,  trCo
ntrol = my_ctrl_roc,
                        preProcess = c('center', 'scale'),
                        trace = FALSE)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```
fit_nnet_roc <- train(outcome ~ .,
                        data = df_class,  method = "nnet",   trControl = my_ctrl_roc,
                        preProcess = c('center', 'scale'),
                        trace = FALSE)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Using Random Forest to Train, evaluate, and tune the glm_cat_cont model.

```
set.seed(4321)

fit_rf_acc <- train(outcome ~ .,
                        data = df_class,  method = "rf",  metric = metric_acc, trCont
rol = my_ctrl_acc ,trace = FALSE)

fit_rf_roc <- train(outcome ~ .,
                        data = df_class,  method = "rf", trControl = my_ctrl_roc ,tra
ce = FALSE)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Using Extrem Gradient Boosted Tree to Train, evaluate, and tune the glm_cat_cont model.

```
set.seed(4321)
fit_xgb_acc <- train(outcome ~ .,
                        data = df_class, method = 'xgbTree', metric = metric_acc, trC
ontrol = my_ctrl_acc,  preProcess = c('center', 'scale'), verbosity = 0)

fit_xgb_roc <- train(outcome ~ .,
                        data = df_class, method = 'xgbTree', trControl = my_ctrl_roc,
preProcess = c('center', 'scale'), verbosity = 0)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Using SVM to Train, evaluate, and tune the glm_cat_cont model.

```
set.seed(4321)
fit_pls_acc <- train(outcome ~ .,  data = df_class,  method = "pls",  metric = metric
_acc,
                     trControl = my_ctrl_acc,
                        preProcess = c('center', 'scale'),
                        trace = FALSE)

fit_pls_roc <- train(outcome ~ .,
                        data = df_class,  method = "pls",   trControl = my_ctrl_roc,
                        preProcess = c('center', 'scale'), importance = TRUE,
                        trace = FALSE)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Using KNN to Train, evaluate, and tune the glm_cat_cont model.

```
fit_knn_acc <- train(outcome ~ .,
                        data = df_class,  method = "knn",  metric = metric_acc, trCon
trol = my_ctrl_acc,
                        preProcess = c('center', 'scale'))

fit_knn_roc <- train(outcome ~ .,
                        data = df_class,  method = "knn",  trControl = my_ctrl_roc,
                        preProcess = c('center', 'scale'))
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```
cat_cont_tune_acc
```

```
## Generalized Linear Model
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 542, 542, 541, 542, 542, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8369368  0.4061143
```

```
pairwise_tune_acc
```

```
## Generalized Linear Model
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (917), scaled (917)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 541, 542, 542, 541, 541, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.5090436  0.01971732
```

```
pairwise_tune_acc_enet
```

```
## glmnet
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (917), scaled (917)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 542, 542, 541, 541, 542, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda      Accuracy   Kappa
##   0.10   0.01039198  0.8088497  0.287327941
##   0.10   0.03286232  0.8221373  0.308150217
##   0.10   0.10391977  0.8259804  0.244211871
##   0.55   0.01039198  0.8197712  0.297273152
##   0.55   0.03286232  0.8271612  0.241654299
##   0.55   0.10391977  0.8189194  0.089376004
##   1.00   0.01039198  0.8242070  0.282578860
##   1.00   0.03286232  0.8230305  0.194994095
##   1.00   0.10391977  0.8133072  0.007630058
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 0.55 and lambda = 0.03286232.
```

```
fit_nnet_acc
```

```
## Neural Network
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 542, 541, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   size  decay  ROC        Sens       Spec
##   1     0e+00  0.7299463  0.4419692  0.8414545
##   1     1e-04  0.7554853  0.6216000  0.7880000
##   1     1e-01  0.8268397  0.4904000  0.8920000
##   3     0e+00  0.7655530  0.4824000  0.8596364
##   3     1e-04  0.7743127  0.4729846  0.8683636
##   3     1e-01  0.7837874  0.4319385  0.8687273
##   5     0e+00  0.7763231  0.4894769  0.8625455
##   5     1e-04  0.7778305  0.4776615  0.8720000
##   5     1e-01  0.7800604  0.4078769  0.8720000
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were size = 1 and decay = 0.1.
```

```
fit_rf_acc
```

```
## Random Forest
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 542, 541, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8203856  0.2130003
##   22    0.8082919  0.2526457
##   43    0.8065185  0.2544518
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
fit_xgb_acc
```

```
## eXtreme Gradient Boosting
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 542, 541, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  subsample  nrounds  Accuracy   Kappa
##   0.3  1          0.6               0.50        50      0.8130000  0.2823817
##   0.3  1          0.6               0.50       100      0.8076863  0.2773012
##   0.3  1          0.6               0.50       150      0.7967538  0.2570461
##   0.3  1          0.6               0.75        50      0.8091656  0.2613077
##   0.3  1          0.6               0.75       100      0.8082658  0.2670792
##   0.3  1          0.6               0.75       150      0.8038235  0.2605973
##   0.3  1          0.6               1.00        50      0.8103529  0.2315027
##   0.3  1          0.6               1.00       100      0.8085708  0.2433223
##   0.3  1          0.6               1.00       150      0.8091503  0.2621768
##   0.3  1          0.8               0.50        50      0.8132919  0.2884634
##   0.3  1          0.8               0.50       100      0.8050044  0.2739623
##   0.3  1          0.8               0.50       150      0.7946797  0.2402142
##   0.3  1          0.8               0.75        50      0.8174444  0.2926027
##   0.3  1          0.8               0.75       100      0.8100414  0.2780364
##   0.3  1          0.8               0.75       150      0.8032462  0.2570141
##   0.3  1          0.8               1.00        50      0.8121285  0.2418932
##   0.3  1          0.8               1.00       100      0.8097538  0.2537657
##   0.3  1          0.8               1.00       150      0.8097429  0.2613991
##   0.3  2          0.6               0.50        50      0.8003007  0.2569425
##   0.3  2          0.6               0.50       100      0.7958824  0.2573299
##   0.3  2          0.6               0.50       150      0.7926318  0.2514879
##   0.3  2          0.6               0.75        50      0.8029586  0.2606482
##   0.3  2          0.6               0.75       100      0.7920174  0.2328071
##   0.3  2          0.6               0.75       150      0.7905468  0.2271942
##   0.3  2          0.6               1.00        50      0.8053159  0.2479008
##   0.3  2          0.6               1.00       100      0.7982135  0.2433879
##   0.3  2          0.6               1.00       150      0.7952658  0.2428139
##   0.3  2          0.8               0.50        50      0.8023638  0.2698115
##   0.3  2          0.8               0.50       100      0.7961699  0.2542804
##   0.3  2          0.8               0.50       150      0.7935011  0.2563295
##   0.3  2          0.8               0.75        50      0.7982353  0.2416338
```

```
##    0.3  2          0.8               0.75        100       0.7946906  0.2490171
##    0.3  2          0.8               0.75        150       0.7887821  0.2368030
##    0.3  2          0.8               1.00         50       0.8014771  0.2335315
##    0.3  2          0.8               1.00        100       0.7955577  0.2328083
##    0.3  2          0.8               1.00        150       0.7896449  0.2258429
##    0.3  3          0.6               0.50         50       0.8050131  0.2670057
##    0.3  3          0.6               0.50        100       0.7946732  0.2375245
##    0.3  3          0.6               0.50        150       0.7934815  0.2369197
##    0.3  3          0.6               0.75         50       0.8005817  0.2531733
##    0.3  3          0.6               0.75        100       0.7952593  0.2463800
##    0.3  3          0.6               0.75        150       0.7984989  0.2667216
##    0.3  3          0.6               1.00         50       0.7946819  0.2381066
##    0.3  3          0.6               1.00        100       0.7937778  0.2299837
##    0.3  3          0.6               1.00        150       0.7926057  0.2336673
##    0.3  3          0.8               0.50         50       0.8044379  0.2715835
##    0.3  3          0.8               0.50        100       0.7946863  0.2480306
##    0.3  3          0.8               0.50        150       0.7949804  0.2457518
##    0.3  3          0.8               0.75         50       0.7926187  0.2216252
##    0.3  3          0.8               0.75        100       0.7929259  0.2308439
##    0.3  3          0.8               0.75        150       0.7935120  0.2332497
##    0.3  3          0.8               1.00         50       0.8023660  0.2559682
##    0.3  3          0.8               1.00        100       0.7988105  0.2468088
##    0.3  3          0.8               1.00        150       0.7979216  0.2513858
##    0.4  1          0.6               0.50         50       0.8067952  0.2750498
##    0.4  1          0.6               0.50        100       0.8038584  0.2589239
##    0.4  1          0.6               0.50        150       0.8038519  0.2834329
##    0.4  1          0.6               0.75         50       0.8112418  0.2776025
##    0.4  1          0.6               0.75        100       0.8020828  0.2621478
##    0.4  1          0.6               0.75        150       0.7973508  0.2458121
##    0.4  1          0.6               1.00         50       0.8109455  0.2508713
##    0.4  1          0.6               1.00        100       0.8094510  0.2615937
##    0.4  1          0.6               1.00        150       0.8070959  0.2655213
##    0.4  1          0.8               0.50         50       0.8124227  0.2999567
##    0.4  1          0.8               0.50        100       0.7970523  0.2584186
##    0.4  1          0.8               0.50        150       0.7973617  0.2529232
##    0.4  1          0.8               0.75         50       0.8056340  0.2440220
##    0.4  1          0.8               0.75        100       0.8017843  0.2430125
##    0.4  1          0.8               0.75        150       0.7979455  0.2397753
##    0.4  1          0.8               1.00         50       0.8130218  0.2587003
##    0.4  1          0.8               1.00        100       0.8088627  0.2615008
##    0.4  1          0.8               1.00        150       0.8056078  0.2582943
##    0.4  2          0.6               0.50         50       0.7949869  0.2465330
##    0.4  2          0.6               0.50        100       0.7870174  0.2312048
##    0.4  2          0.6               0.50        150       0.7896667  0.2442424
##    0.4  2          0.6               0.75         50       0.7967342  0.2580543
##    0.4  2          0.6               0.75        100       0.7970414  0.2687535
##    0.4  2          0.6               0.75        150       0.7940806  0.2472883
```

```
##   0.4  2          0.6                 1.00         50       0.7996885  0.2401947
##   0.4  2          0.6                 1.00        100       0.7949455  0.2440420
##   0.4  2          0.6                 1.00        150       0.7958410  0.2487236
##   0.4  2          0.8                 0.50         50       0.7961634  0.2573260
##   0.4  2          0.8                 0.50        100       0.7952919  0.2586614
##   0.4  2          0.8                 0.50        150       0.7935185  0.2648670
##   0.4  2          0.8                 0.75         50       0.8017647  0.2640843
##   0.4  2          0.8                 0.75        100       0.7925904  0.2527595
##   0.4  2          0.8                 0.75        150       0.7884532  0.2448207
##   0.4  2          0.8                 1.00         50       0.8014662  0.2481301
##   0.4  2          0.8                 1.00        100       0.7899586  0.2290135
##   0.4  2          0.8                 1.00        150       0.7902593  0.2343891
##   0.4  3          0.6                 0.50         50       0.8014967  0.2637866
##   0.4  3          0.6                 0.50        100       0.7979303  0.2583711
##   0.4  3          0.6                 0.50        150       0.7994074  0.2598881
##   0.4  3          0.6                 0.75         50       0.8035599  0.2749628
##   0.4  3          0.6                 0.75        100       0.7946885  0.2454333
##   0.4  3          0.6                 0.75        150       0.7982309  0.2517559
##   0.4  3          0.6                 1.00         50       0.7970370  0.2439013
##   0.4  3          0.6                 1.00        100       0.8002789  0.2552603
##   0.4  3          0.6                 1.00        150       0.7982179  0.2567209
##   0.4  3          0.8                 0.50         50       0.7940980  0.2425109
##   0.4  3          0.8                 0.50        100       0.8011721  0.2776495
##   0.4  3          0.8                 0.50        150       0.8026427  0.2862123
##   0.4  3          0.8                 0.75         50       0.7929129  0.2396934
##   0.4  3          0.8                 0.75        100       0.7935054  0.2445333
##   0.4  3          0.8                 0.75        150       0.7931983  0.2478071
##   0.4  3          0.8                 1.00         50       0.7982092  0.2458159
##   0.4  3          0.8                 1.00        100       0.7988083  0.2548772
##   0.4  3          0.8                 1.00        150       0.7982135  0.2532278
## 
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##  parameter 'min_child_weight' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 50, max_depth = 1, eta
##  = 0.3, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample
##  = 0.75.
```

```
fit_pls_acc
```

```
## Partial Least Squares
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 542, 541, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   ncomp  Accuracy   Kappa
##   1      0.8236558  0.1531122
##   2      0.8263246  0.2107718
##   3      0.8277930  0.1962398
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was ncomp = 3.
```

```
fit_knn_acc
```

```
## k-Nearest Neighbors
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 542, 541, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.8115163  0.2568544
##   7  0.8197865  0.2703851
##   9  0.8194989  0.2505884
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

```
cat_cont_tune_roc
```

```
## Generalized Linear Model
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 541, 542, 542, 542, 542, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.8320599  0.4379692  0.932
```

```
pairwise_tune_roc
```

```
## Generalized Linear Model
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (917), scaled (917)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 541, 542, 542, 541, 541, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.5060948  0.4865231  0.5221818
```

```
pairwise_tune_roc_enet
```

```
## glmnet
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (917), scaled (917)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 541, 542, 541, 542, 541, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda      ROC        Sens       Spec
##   0.10   0.01039198  0.8020699  0.3480615  0.9192727
##   0.10   0.03286232  0.8194154  0.3244923  0.9349091
##   0.10   0.10391977  0.8433331  0.2360615  0.9647273
##   0.55   0.01039198  0.8174316  0.3273846  0.9349091
##   0.55   0.03286232  0.8468319  0.2188923  0.9690909
##   0.55   0.10391977  0.8389866  0.0648000  0.9923636
##   1.00   0.01039198  0.8326730  0.2817231  0.9520000
##   1.00   0.03286232  0.8447748  0.1735385  0.9749091
##   1.00   0.10391977  0.8239122  0.0032000  1.0000000
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 0.55 and lambda = 0.03286232.
```

```
fit_nnet_roc
```

```
## Neural Network
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 542, 542, 541, 542, 541, ...
## Resampling results across tuning parameters:
##
##   size  decay  ROC        Sens       Spec
##   1     0e+00  0.7342764  0.6107077  0.7843636
##   1     1e-04  0.7569891  0.6452923  0.7712727
##   1     1e-01  0.8318870  0.5019077  0.8989091
##   3     0e+00  0.7674638  0.5105231  0.8578182
##   3     1e-04  0.7776050  0.4545231  0.8767273
##   3     1e-01  0.7873594  0.4316923  0.8807273
##   5     0e+00  0.7622722  0.4688615  0.8698182
##   5     1e-04  0.7536008  0.4550769  0.8632727
##   5     1e-01  0.7835933  0.4256000  0.8730909
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were size = 1 and decay = 0.1.
```

```
fit_rf_roc
```

```
## Random Forest
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 541, 542, 542, 542, 542, ...
## Resampling results across tuning parameters:
##
##   mtry  ROC        Sens       Spec
##    2    0.8316719  0.2164308  0.9600000
##   22    0.8213407  0.3072615  0.9287273
##   43    0.8137200  0.3214154  0.9272727
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
fit_xgb_roc
```

```
## eXtreme Gradient Boosting
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 542, 542, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  subsample  nrounds  ROC        Sens
##   0.3  1          0.6               0.50        50      0.8163639  0.3356923
##   0.3  1          0.6               0.50       100      0.8042148  0.3408000
##   0.3  1          0.6               0.50       150      0.7915357  0.3409846
##   0.3  1          0.6               0.75        50      0.8197373  0.3057231
##   0.3  1          0.6               0.75       100      0.8065483  0.3264615
##   0.3  1          0.6               0.75       150      0.7985326  0.3185846
##   0.3  1          0.6               1.00        50      0.8203024  0.2416000
##   0.3  1          0.6               1.00       100      0.8148503  0.2743385
##   0.3  1          0.6               1.00       150      0.8084112  0.3072615
##   0.3  1          0.8               0.50        50      0.8130971  0.3230769
##   0.3  1          0.8               0.50       100      0.8020526  0.3216615
##   0.3  1          0.8               0.50       150      0.7969348  0.3200000
##   0.3  1          0.8               0.75        50      0.8177116  0.3043692
##   0.3  1          0.8               0.75       100      0.8055250  0.3123692
##   0.3  1          0.8               0.75       150      0.7981494  0.3187692
##   0.3  1          0.8               1.00        50      0.8212173  0.2619692
##   0.3  1          0.8               1.00       100      0.8135105  0.2806154
##   0.3  1          0.8               1.00       150      0.8087681  0.2900923
##   0.3  2          0.6               0.50        50      0.7971603  0.3218462
##   0.3  2          0.6               0.50       100      0.7817538  0.3297231
##   0.3  2          0.6               0.50       150      0.7730697  0.3329846
##   0.3  2          0.6               0.75        50      0.8071390  0.3515692
##   0.3  2          0.6               0.75       100      0.7937007  0.3358769
##   0.3  2          0.6               0.75       150      0.7878456  0.3499077
##   0.3  2          0.6               1.00        50      0.8063429  0.3029538
##   0.3  2          0.6               1.00       100      0.7942081  0.3203692
##   0.3  2          0.6               1.00       150      0.7866294  0.3294769
##   0.3  2          0.8               0.50        50      0.8042143  0.3377231
##   0.3  2          0.8               0.50       100      0.7908201  0.3376615
##   0.3  2          0.8               0.50       150      0.7824727  0.3360000
##   0.3  2          0.8               0.75        50      0.8011345  0.3157538
```

```
##    0.3  2          0.8              0.75        100      0.7881029  0.3361231
##    0.3  2          0.8              0.75        150      0.7839787  0.3360615
##    0.3  2          0.8              1.00         50      0.8019323  0.3011077
##    0.3  2          0.8              1.00        100      0.7891262  0.3123692
##    0.3  2          0.8              1.00        150      0.7852929  0.3252308
##    0.3  3          0.6              0.50         50      0.7970478  0.3452308
##    0.3  3          0.6              0.50        100      0.7829975  0.3404308
##    0.3  3          0.6              0.50        150      0.7768895  0.3511385
##    0.3  3          0.6              0.75         50      0.8006786  0.3465231
##    0.3  3          0.6              0.75        100      0.7849180  0.3324923
##    0.3  3          0.6              0.75        150      0.7811905  0.3452308
##    0.3  3          0.6              1.00         50      0.7944369  0.2950154
##    0.3  3          0.6              1.00        100      0.7857499  0.3094769
##    0.3  3          0.6              1.00        150      0.7809706  0.3172923
##    0.3  3          0.8              0.50         50      0.7891653  0.3472615
##    0.3  3          0.8              0.50        100      0.7812319  0.3392615
##    0.3  3          0.8              0.50        150      0.7769768  0.3392000
##    0.3  3          0.8              0.75         50      0.7954982  0.3422769
##    0.3  3          0.8              0.75        100      0.7901572  0.3329231
##    0.3  3          0.8              0.75        150      0.7838014  0.3265846
##    0.3  3          0.8              1.00         50      0.7976207  0.3059077
##    0.3  3          0.8              1.00        100      0.7847972  0.2995692
##    0.3  3          0.8              1.00        150      0.7804973  0.3137846
##    0.4  1          0.6              0.50         50      0.8074408  0.3388308
##    0.4  1          0.6              0.50        100      0.7936481  0.3468923
##    0.4  1          0.6              0.50        150      0.7897617  0.3438769
##    0.4  1          0.6              0.75         50      0.8115804  0.2966154
##    0.4  1          0.6              0.75        100      0.8028906  0.3310154
##    0.4  1          0.6              0.75        150      0.7919748  0.3295385
##    0.4  1          0.6              1.00         50      0.8174490  0.2731077
##    0.4  1          0.6              1.00        100      0.8109941  0.2932308
##    0.4  1          0.6              1.00        150      0.8024593  0.3137846
##    0.4  1          0.8              0.50         50      0.8000761  0.3313231
##    0.4  1          0.8              0.50        100      0.7939866  0.3686769
##    0.4  1          0.8              0.50        150      0.7757427  0.3435077
##    0.4  1          0.8              0.75         50      0.8116543  0.3153231
##    0.4  1          0.8              0.75        100      0.7943575  0.3033231
##    0.4  1          0.8              0.75        150      0.7894042  0.3126154
##    0.4  1          0.8              1.00         50      0.8166498  0.2650462
##    0.4  1          0.8              1.00        100      0.8088453  0.2979077
##    0.4  1          0.8              1.00        150      0.8038971  0.3212308
##    0.4  2          0.6              0.50         50      0.7763021  0.3196923
##    0.4  2          0.6              0.50        100      0.7719894  0.3342769
##    0.4  2          0.6              0.50        150      0.7657650  0.3451692
##    0.4  2          0.6              0.75         50      0.7817248  0.3028308
##    0.4  2          0.6              0.75        100      0.7753264  0.3127385
##    0.4  2          0.6              0.75        150      0.7706853  0.3110769
```

```
##    0.4  2         0.6          1.00     50     0.7922378  0.2916923
##    0.4  2         0.6          1.00    100     0.7821857  0.3120615
##    0.4  2         0.6          1.00    150     0.7763765  0.3199385
##    0.4  2         0.8          0.50     50     0.7859312  0.3582154
##    0.4  2         0.8          0.50    100     0.7732050  0.3438154
##    0.4  2         0.8          0.50    150     0.7687452  0.3473846
##    0.4  2         0.8          0.75     50     0.7935401  0.3362462
##    0.4  2         0.8          0.75    100     0.7869555  0.3392000
##    0.4  2         0.8          0.75    150     0.7784699  0.3455385
##    0.4  2         0.8          1.00     50     0.7963457  0.3123077
##    0.4  2         0.8          1.00    100     0.7843916  0.3139692
##    0.4  2         0.8          1.00    150     0.7761617  0.3315077
##    0.4  3         0.6          0.50     50     0.7807401  0.3563692
##    0.4  3         0.6          0.50    100     0.7770081  0.3648615
##    0.4  3         0.6          0.50    150     0.7727759  0.3617846
##    0.4  3         0.6          0.75     50     0.7854344  0.3392000
##    0.4  3         0.6          0.75    100     0.7789303  0.3344000
##    0.4  3         0.6          0.75    150     0.7762081  0.3296615
##    0.4  3         0.6          1.00     50     0.7851877  0.3391385
##    0.4  3         0.6          1.00    100     0.7768917  0.3172308
##    0.4  3         0.6          1.00    150     0.7743211  0.3171692
##    0.4  3         0.8          0.50     50     0.7796878  0.3373538
##    0.4  3         0.8          0.50    100     0.7719944  0.3563077
##    0.4  3         0.8          0.50    150     0.7686378  0.3438154
##    0.4  3         0.8          0.75     50     0.7856481  0.3249846
##    0.4  3         0.8          0.75    100     0.7787189  0.3315077
##    0.4  3         0.8          0.75    150     0.7756694  0.3331692
##    0.4  3         0.8          1.00     50     0.7922920  0.3247385
##    0.4  3         0.8          1.00    100     0.7859144  0.3281231
##    0.4  3         0.8          1.00    150     0.7792828  0.3374154
##    Spec
##    0.9356364
##    0.9203636
##    0.9123636
##    0.9363636
##    0.9203636
##    0.9160000
##    0.9429091
##    0.9352727
##    0.9312727
##    0.9330909
##    0.9312727
##    0.9261818
##    0.9356364
##    0.9218182
##    0.9152727
##    0.9410909
```

```
##   0.9345455
##   0.9316364
##   0.9170909
##   0.9036364
##   0.9025455
##   0.9221818
##   0.9094545
##   0.9061818
##   0.9280000
##   0.9163636
##   0.9101818
##   0.9225455
##   0.9098182
##   0.9040000
##   0.9218182
##   0.9098182
##   0.9069091
##   0.9287273
##   0.9160000
##   0.9054545
##   0.9149091
##   0.9156364
##   0.9120000
##   0.9178182
##   0.9112727
##   0.9112727
##   0.9218182
##   0.9156364
##   0.9134545
##   0.9174545
##   0.9087273
##   0.9069091
##   0.9120000
##   0.9076364
##   0.9083636
##   0.9200000
##   0.9149091
##   0.9127273
##   0.9272727
##   0.9090909
##   0.9025455
##   0.9298182
##   0.9229091
##   0.9141818
##   0.9385455
##   0.9330909
##   0.9243636
```

```
##   0.9247273
##   0.9080000
##   0.9014545
##   0.9287273
##   0.9127273
##   0.9076364
##   0.9392727
##   0.9305455
##   0.9236364
##   0.9014545
##   0.8978182
##   0.8970909
##   0.9087273
##   0.8967273
##   0.8989091
##   0.9178182
##   0.9116364
##   0.9105455
##   0.9090909
##   0.9058182
##   0.9069091
##   0.9141818
##   0.9080000
##   0.9043636
##   0.9236364
##   0.9112727
##   0.9072727
##   0.9018182
##   0.9036364
##   0.9029091
##   0.9050909
##   0.9083636
##   0.9040000
##   0.9101818
##   0.9105455
##   0.9127273
##   0.9080000
##   0.9072727
##   0.9083636
##   0.9109091
##   0.9061818
##   0.9087273
##   0.9163636
##   0.9101818
##   0.9141818
##
## Tuning parameter 'gamma' was held constant at a value of 0
```

```
## Tuning
##  parameter 'min_child_weight' was held constant at a value of 1
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 50, max_depth = 1, eta
##  = 0.3, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample
##  = 1.
```

```
fit_pls_roc
```

```
## Partial Least Squares
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 542, 541, 542, 542, 541, 541, ...
## Resampling results across tuning parameters:
##
##   ncomp  ROC        Sens       Spec
##   1      0.8094193  0.1257846  0.9836364
##   2      0.8323234  0.1921231  0.9745455
##   3      0.8428850  0.1668923  0.9836364
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was ncomp = 3.
```

```
fit_knn_roc
```

```
## k-Nearest Neighbors
##
## 677 samples
##  35 predictor
##   2 classes: 'event', 'non_event'
##
## Pre-processing: centered (43), scaled (43)
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 541, 542, 542, 542, 541, 542, ...
## Resampling results across tuning parameters:
##
##   k  ROC        Sens       Spec
##   5  0.7182420  0.2774769  0.9363636
##   7  0.7436666  0.2726769  0.9520000
##   9  0.7636523  0.2425846  0.9534545
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

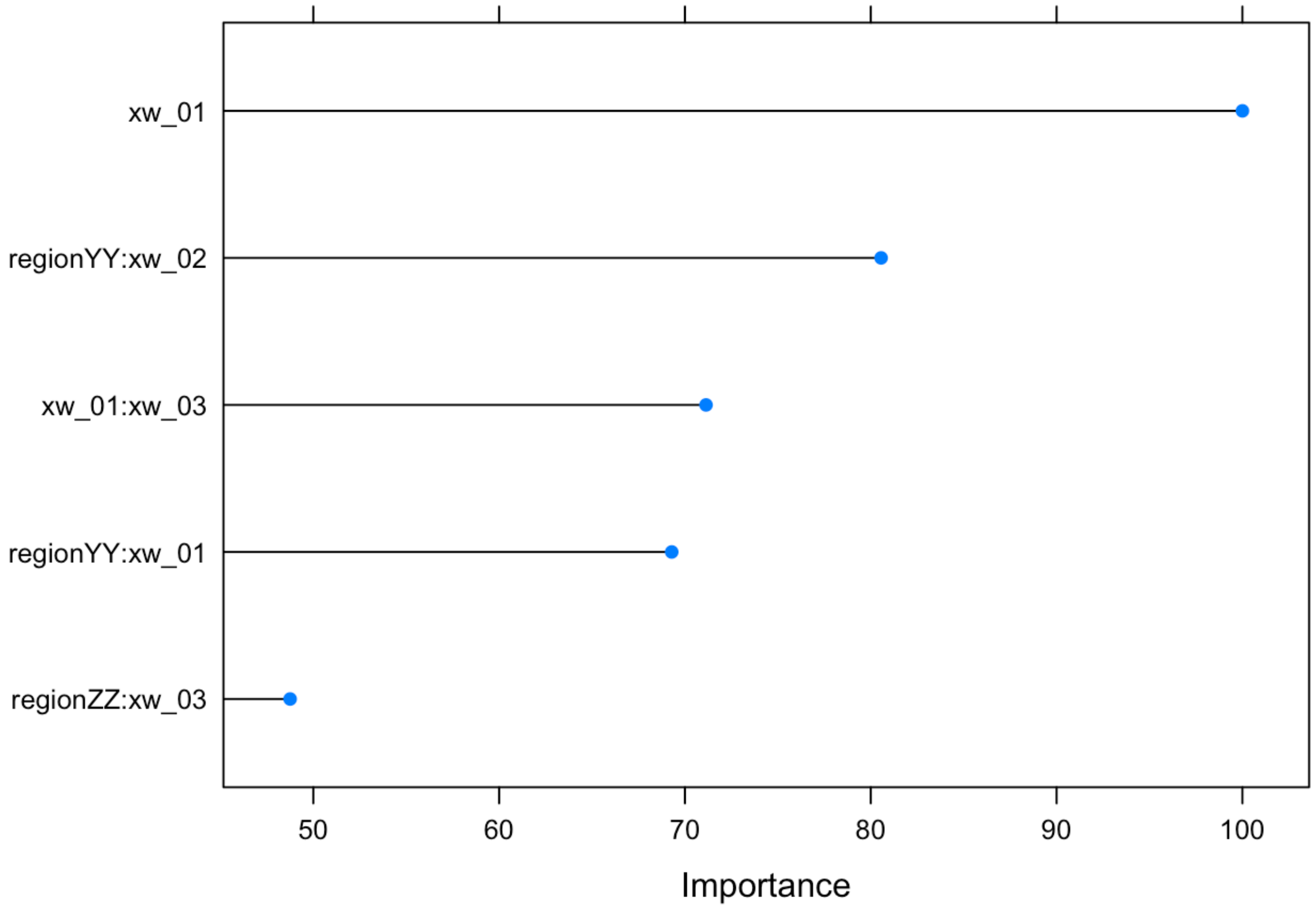
The model that maximizes the accuracy is the categorical and continuous inputs trained through partial least squares. The model that maximizes the ROC is also the categorical and continuous inputs trained through partial least squares.

Interpretation - Part A

- When trained, tuned, and evaluated the best model is The pairwise interactions tuned through elastic net.
- For the classification models, the model that was selected based on Accuracy and AUC ROC is the model with the categoerical and continuous inputs trained via Partial Least Squares.

Identifying the most important variables associated with the best performing regression and classification models

```
plot(varImp(pairwise_enet), top = 5)
```

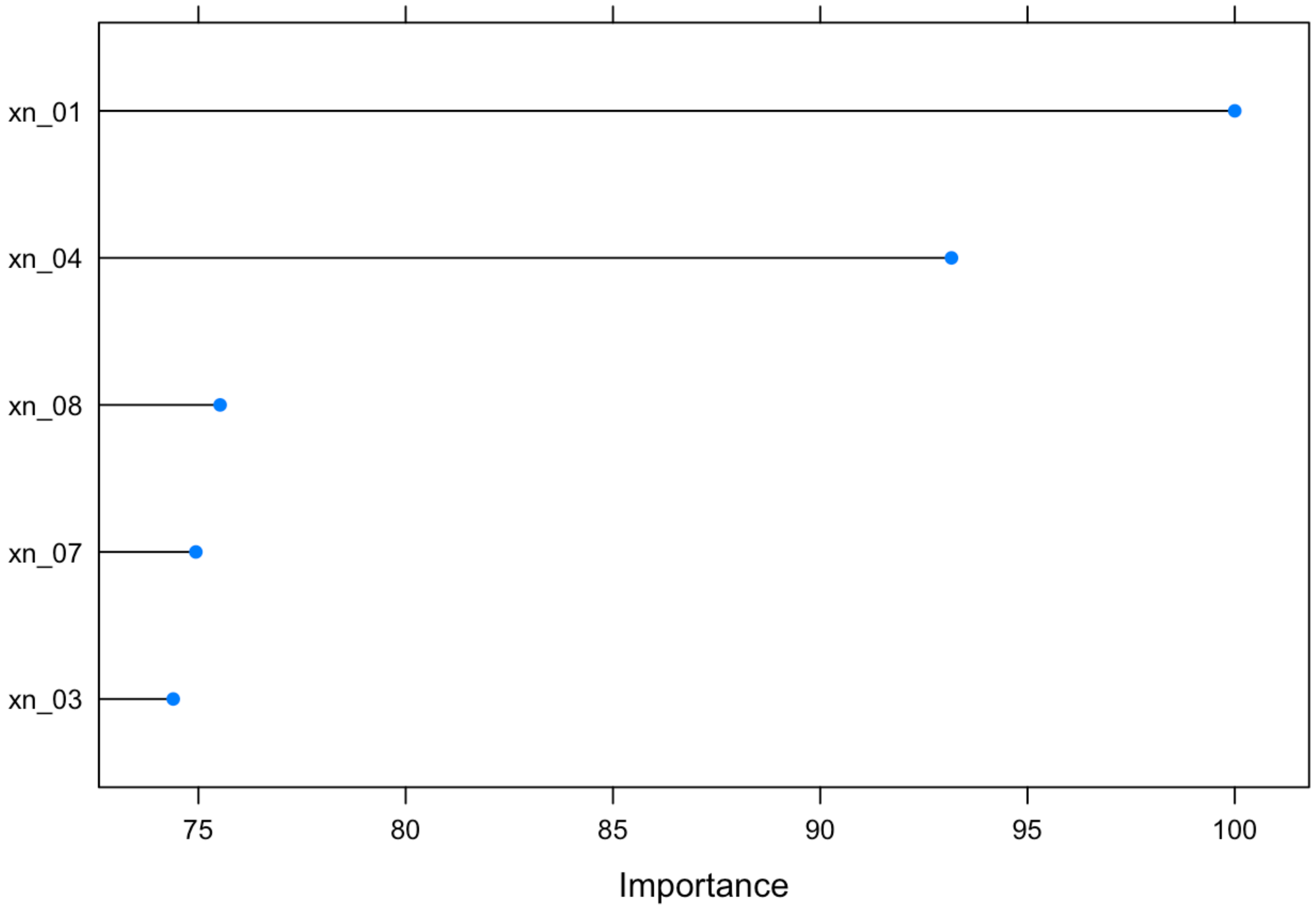



```
plot(varImp(fit_pls_acc), top = 5)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:caret':
##
##     R2
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```



For the regression pairwise model tuned through elastic net xw_01 is the most important variable. For the classification RF, the most important variable is xn_01

The important features between the regression and classification models are not very consistent with each other.

The model results are okay. They had a very high performance on the test metric but I feel as though based on that and the sentiment contributions, that they are not actually that useful. It seems that the most important feature is the response type.

Read in Hold-out set

```
holdout <- readr::read_csv('final_project_holdout_inputs.csv', col_names = TRUE)
```

```
## Rows: 73 Columns: 36
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (2): region, customer
## dbl (34): rowid, xb_01, xb_02, xb_03, xn_01, xn_02, xn_03, xa_01, xa_02, xa_...
##
## 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.
```

```
holdout_inputs <- holdout %>%
  select(-rowid)
```

Predict using Pairwise Elastic Net - Regression

```
predict_reg <- predict(pairwise_enet, holdout_inputs)
```

Predict using Partial Least Squares - Classification

```
predict_class <- predict(fit_pls_acc, holdout_inputs)
predict_class_prob <- predict(fit_pls_acc, holdout_inputs, type = 'prob')
head(predict_class_prob)
```

```
##       event non_event
## 1 0.4209747 0.5790253
## 2 0.3823736 0.6176264
## 3 0.2707591 0.7292409
## 4 0.4532684 0.5467316
## 5 0.3665141 0.6334859
## 6 0.3053693 0.6946307
```

Compile Predictions

```
my_preds <- tibble::tibble(
  y = predict_reg,
  outcome = predict_class
) %>%
  bind_cols(
    predict_class_prob %>%
      select(probability = event)
  ) %>%
  tibble::rowid_to_column('id')

head(my_preds)
```

```
## # A tibble: 6 × 4
##      id     y outcome   probability
##   <int> <dbl> <fct>           <dbl>
## 1     1 0.450 non_event       0.421
## 2     2 0.874 non_event       0.382
## 3     3 1.13  non_event       0.271
## 4     4 0.924 non_event       0.453
## 5     5 0.492 non_event       0.367
## 6     6 0.336 non_event       0.305
```

```
my_preds %>%
  readr::write_csv('Boppana_Sameera_preds.csv', col_names = TRUE)
```

Determining the hardest customer to predict

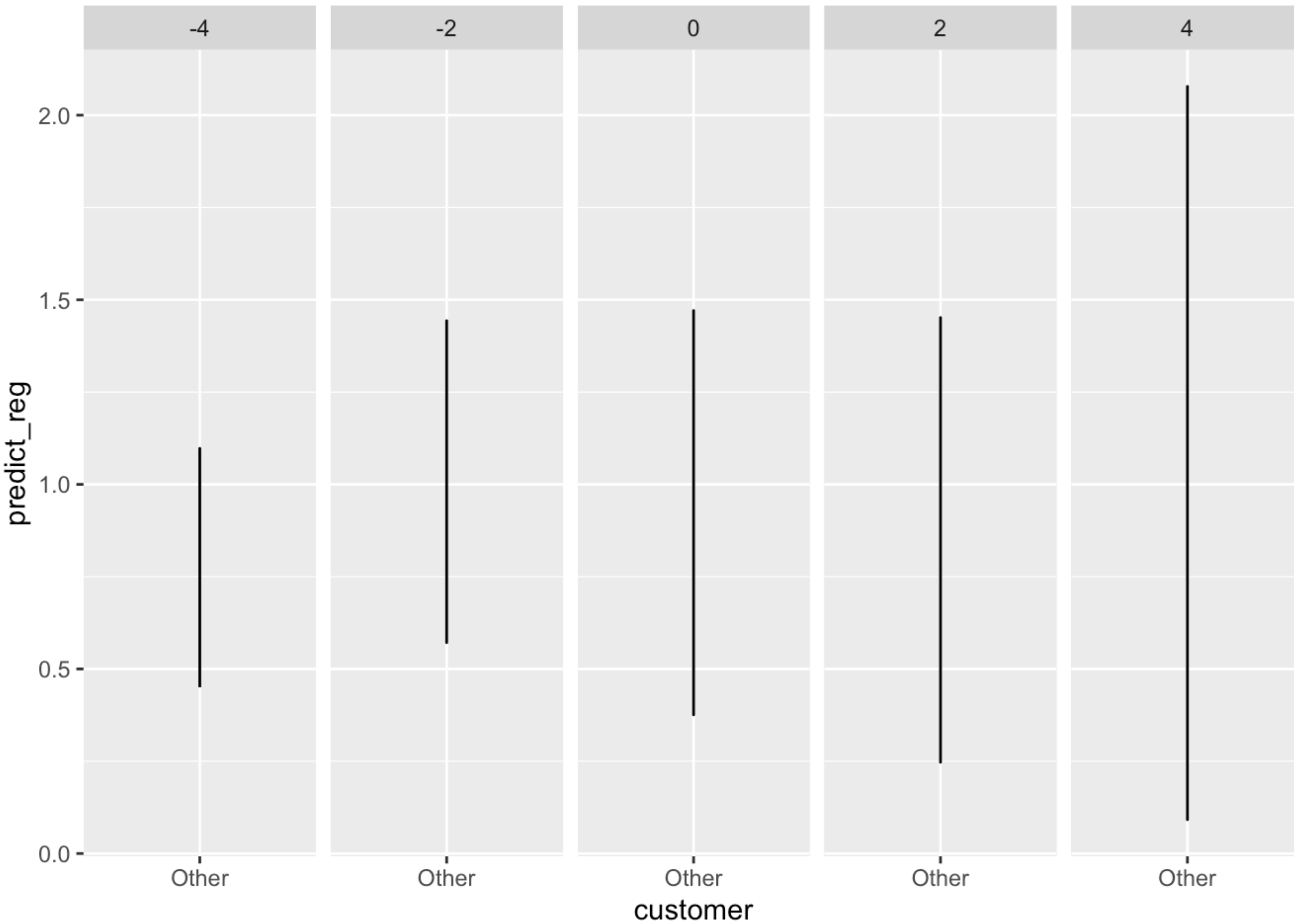
```
holdout$predict_reg <- predict_reg
holdout$predict_class <- predict_class

customer <- data.frame(cbind(holdout$customer, holdout$predict_reg, holdout$predict_c
lass))
colnames(customer) <- c("customer", "pred_reg", "pred_class")
customer <- customer %>% mutate(pred_class = ifelse(pred_class == 2, "non_event", "ev
ent"))
customer$true_class <- ifelse(predict_reg > 0.5, "event", "non_event")
customer %>% group_by(customer) %>%
  count(pred_class == true_class)
```

```
## # A tibble: 13 × 3
## # Groups:   customer [8]
##    customer `pred_class == true_class`     n
##    <chr>    <lgl>                      <int>
##  1 A        FALSE                          8
##  2 A        TRUE                           2
##  3 B        FALSE                          4
##  4 D        FALSE                          2
##  5 E        FALSE                          3
##  6 E        TRUE                           1
##  7 G        FALSE                          7
##  8 G        TRUE                           5
##  9 M        FALSE                          4
## 10 M        TRUE                           3
## 11 Other    FALSE                         25
## 12 Other    TRUE                           6
## 13 Q        FALSE                          3
```

The most amount of missclassifications is from Customer “Other”. Since this customer group had the most amount of missclassications, this is the hardest customer to predict.

Visuzaling prediction trends with the hardest to classify customer group (Other) with the most important derived feature xn_01

```
hardest_customer <- holdout[holdout$customer == 'Other',]

xn_01_vals <- hardest_customer['xn_01']
min <- min(xn_01_vals)
max <- max(xn_01_vals)
mean <- (min + max) /2


med_0.25 <- (min + mean) /2
med_0.75 <- (mean + max) /2
levels <- c(min, med_0.25, mean, med_0.75, max)
levels_spaced <- rep(levels, each = nrow(hardest_customer)/5)
levels_spaced[length(levels_spaced) ]  = levels_spaced[length(levels_spaced) ]
levels_spaced[length(levels_spaced) +1]  = levels_spaced[length(levels_spaced) ]
hardest_customer['levels'] = as.factor(levels_spaced)

hardest_customer %>%
  ggplot(mapping = aes(x = customer)) +
  geom_line(mapping = aes(y = predict_reg)) +
  facet_grid(~ hardest_customer$levels)
```



At higher levels of xn_01, there is a higher chance of predicting an event for Customer Other.