Programming Summary

Jose Dixon

June 9, 2022

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE, error = TRUE)
```

```
library(tidyverse)
library(reshape)
library(ggplot2)
library(modelr)
library(tinytex)
options(na.action = na.warn)
```

Hadley Wickman Intro to Data Science This is the website for "R for Data Science".https://r4ds.had.co.nz

```
getwd()
```

```
## [1] "C:/Users/z3696/Documents/Document-Classification/classifier/Output"
```

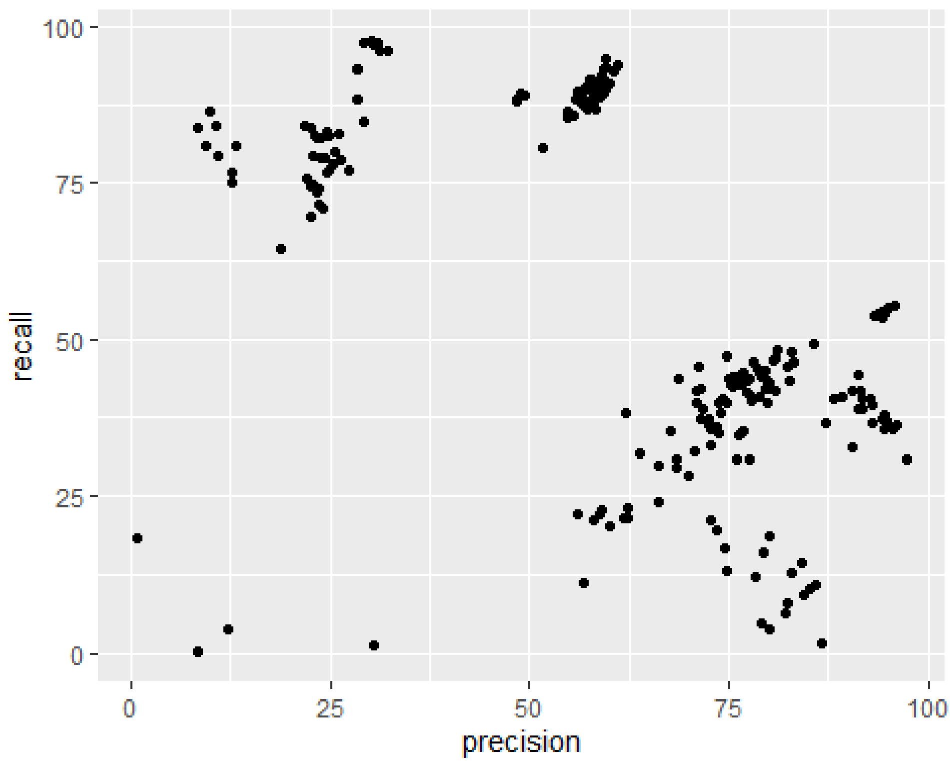
```
table <- read.csv("~/Document-Classification/classifier/Output/Table.csv")
head(table)
```

```
##   Year      Sampling Technique    Classifier Precision Recall
## 1 2010    Imbalanced       N/A   Naive Bayes     74.49  16.70
## 2 2010    Imbalanced       N/A  Logistic Reg     72.82  21.18
## 3 2010    Imbalanced       N/A       XGBoost     12.66  75.05
## 4 2010    Imbalanced       N/A  DecisionTree     59.00  22.77
## 5 2010    Imbalanced       N/A Random Forest     22.65  69.45
## 6 2010 Undersampling  NearMiss   Naive Bayes     76.67  35.38
```

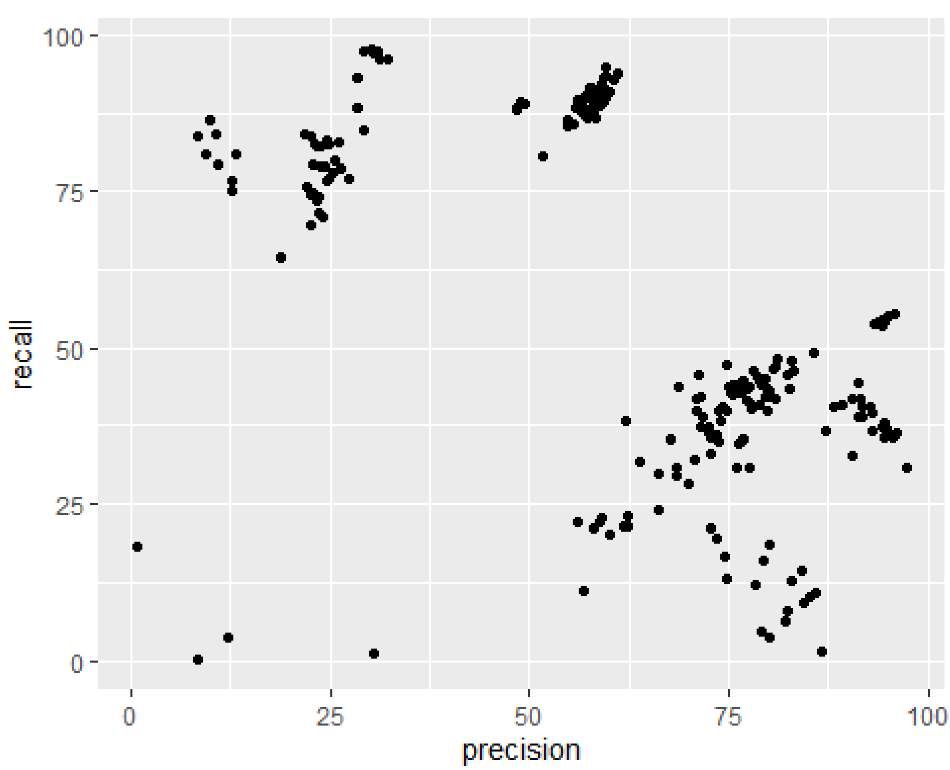
```
precision = table[, 5]
recall = table[, 6]
classifier = table[, 4]
sampling = table[, 2]
technique = table[, 3]
year = table[, 1]
```

Chapter 3 Data Visualization

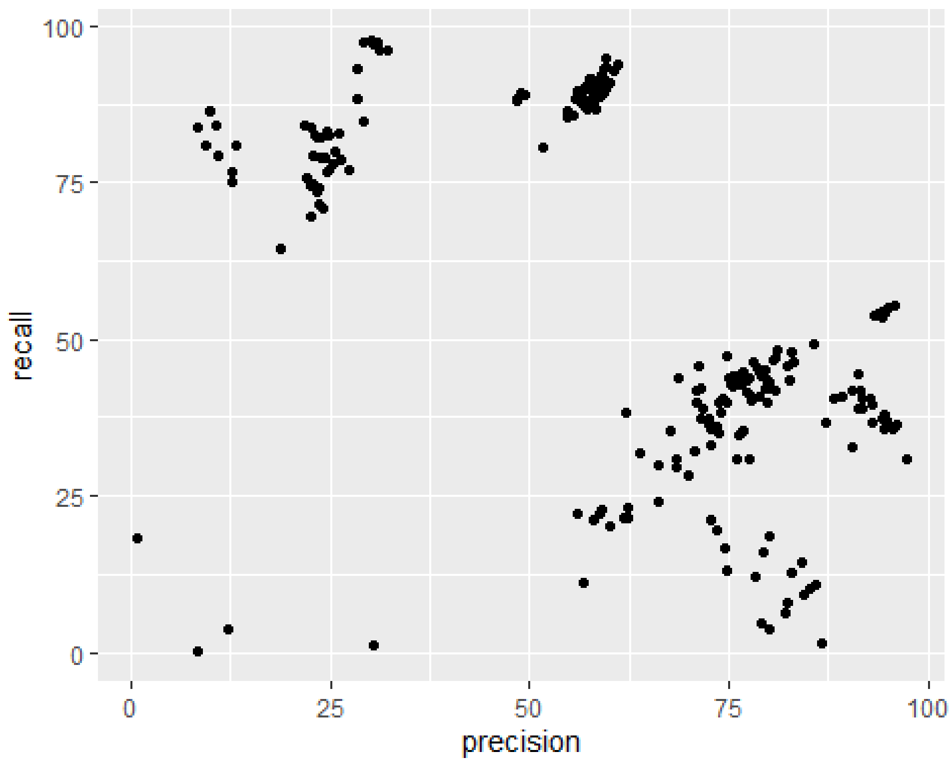
```
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall))
```



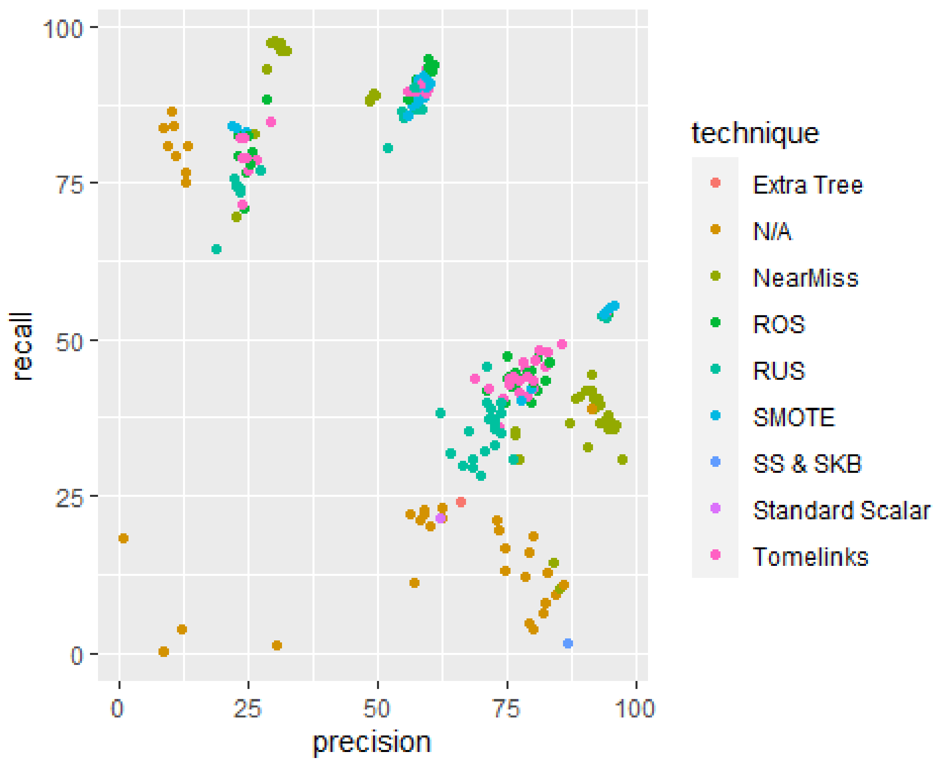

```r
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall))
```



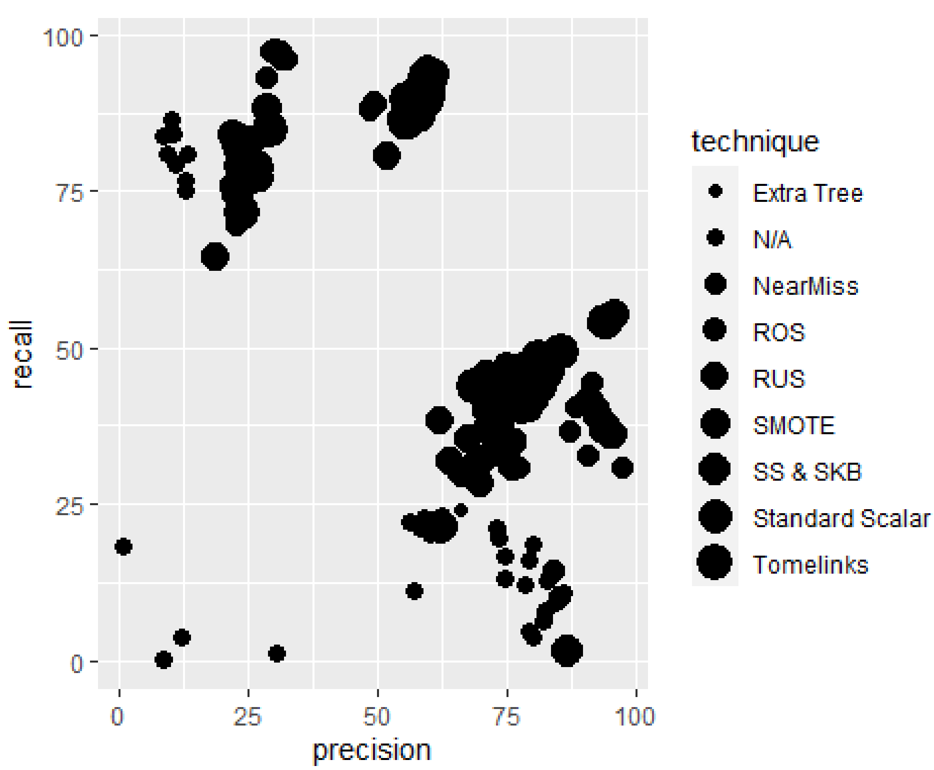
```
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall))
```



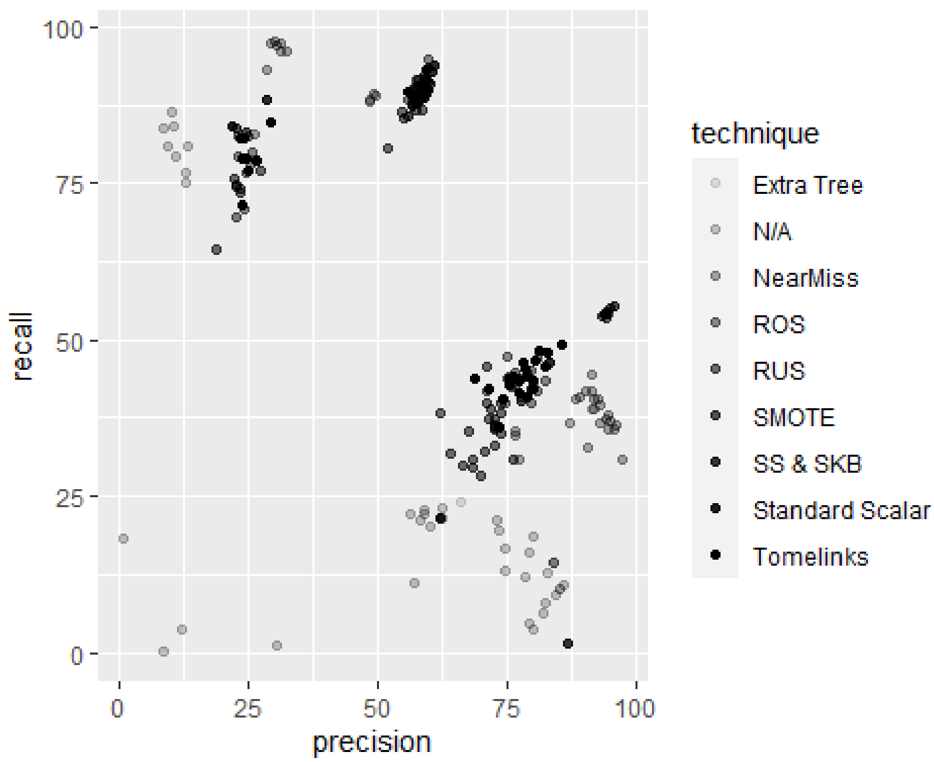

```
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall, color = technique))
```



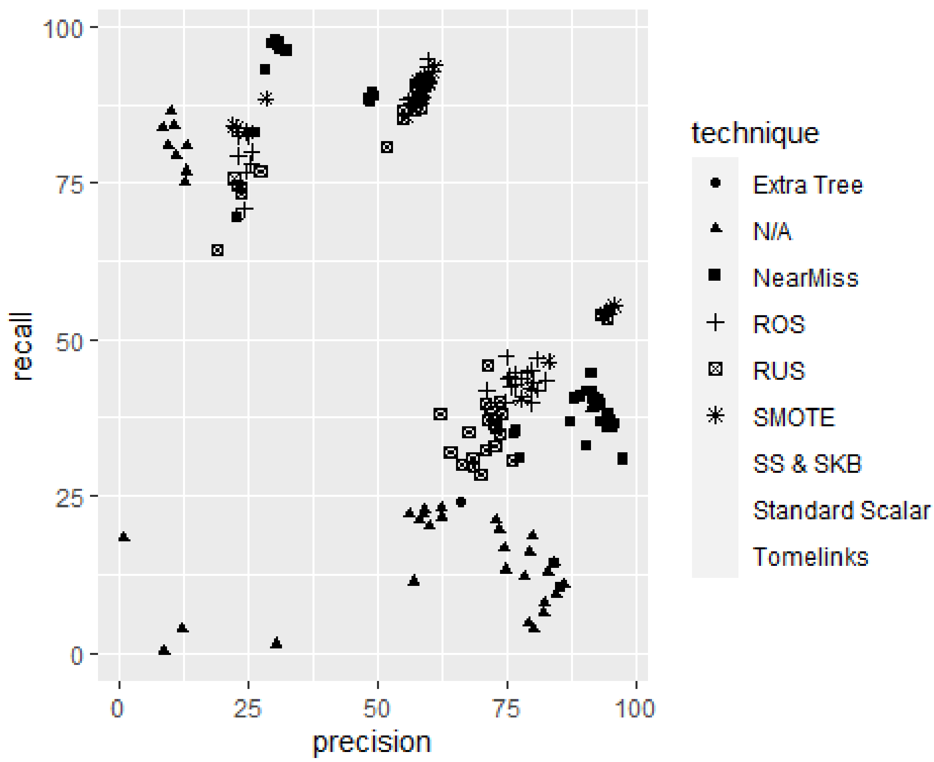

```
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall, size = technique))
```



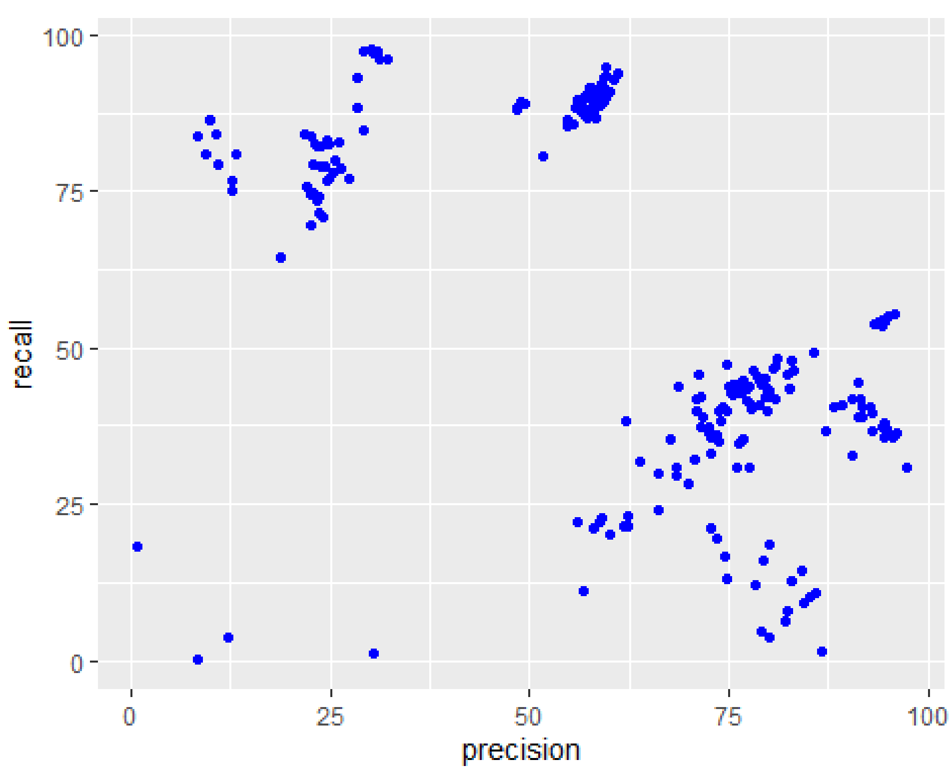
```r
# Left
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall, alpha = technique))
```



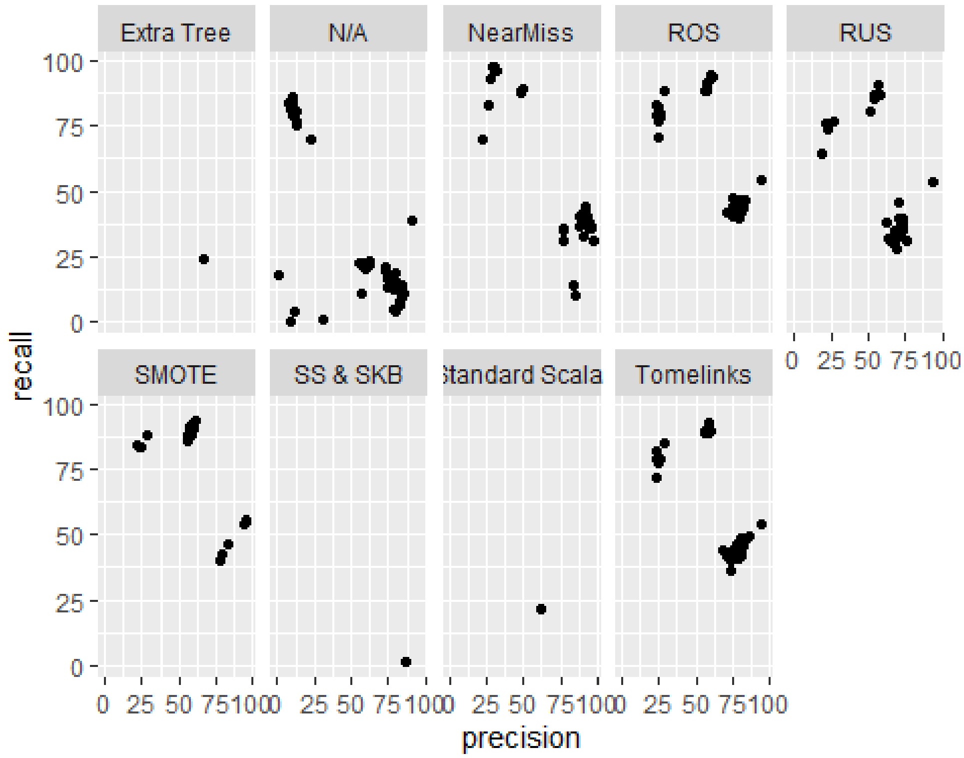

```r
# Right
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall, shape = technique))
```



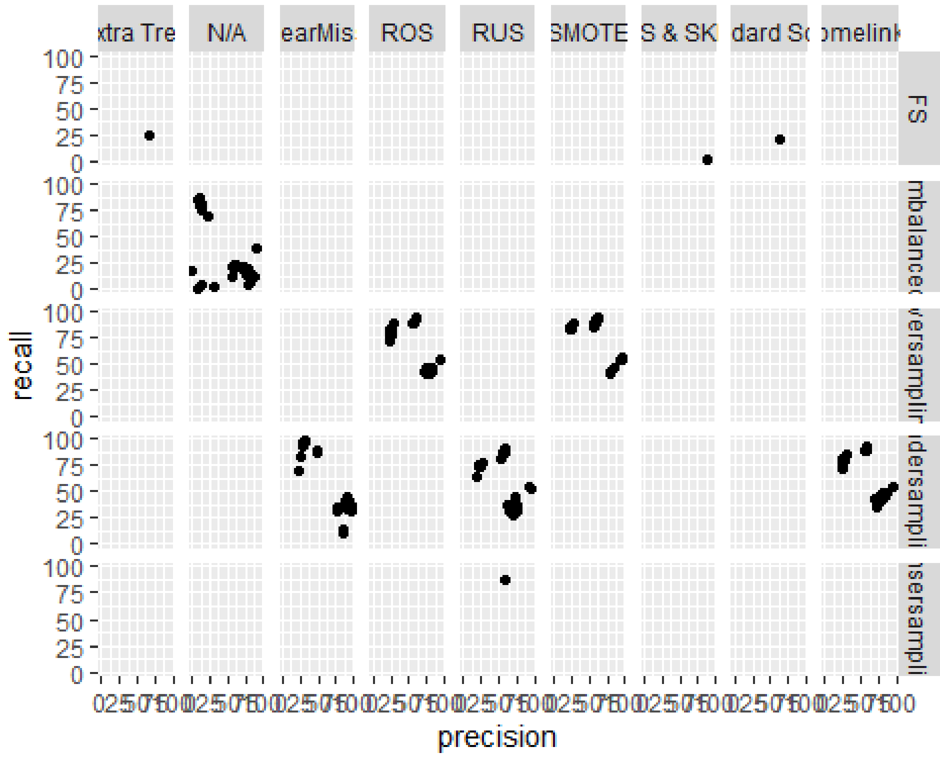

```r
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall), color = "blue")
```



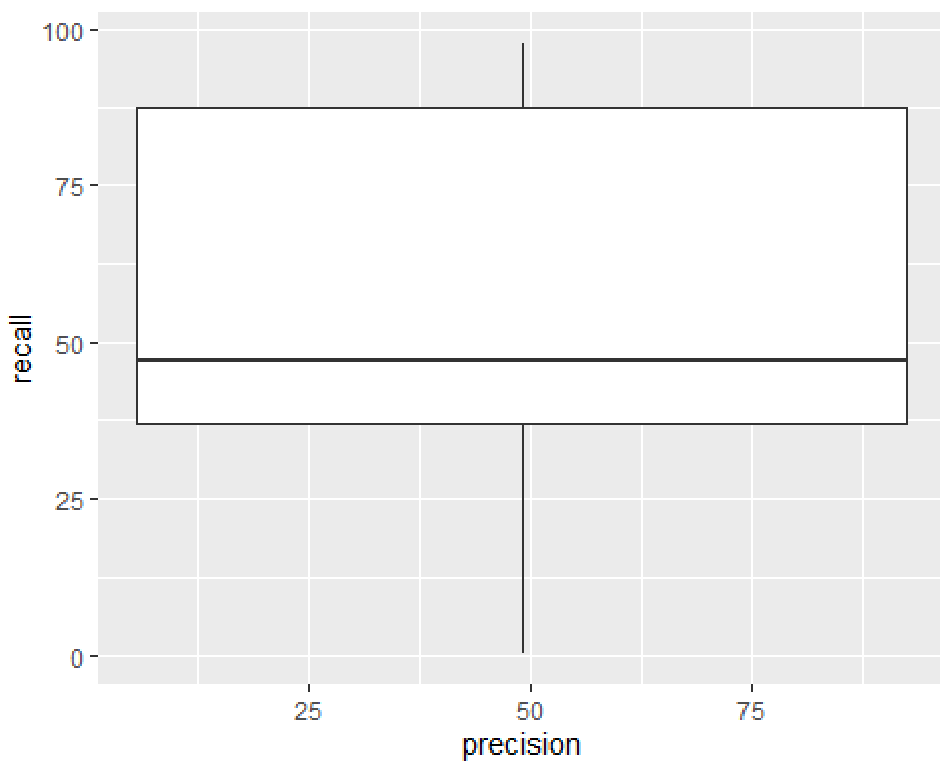
```
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall)) + facet_wrap(~
Technique, nrow = 2)
```



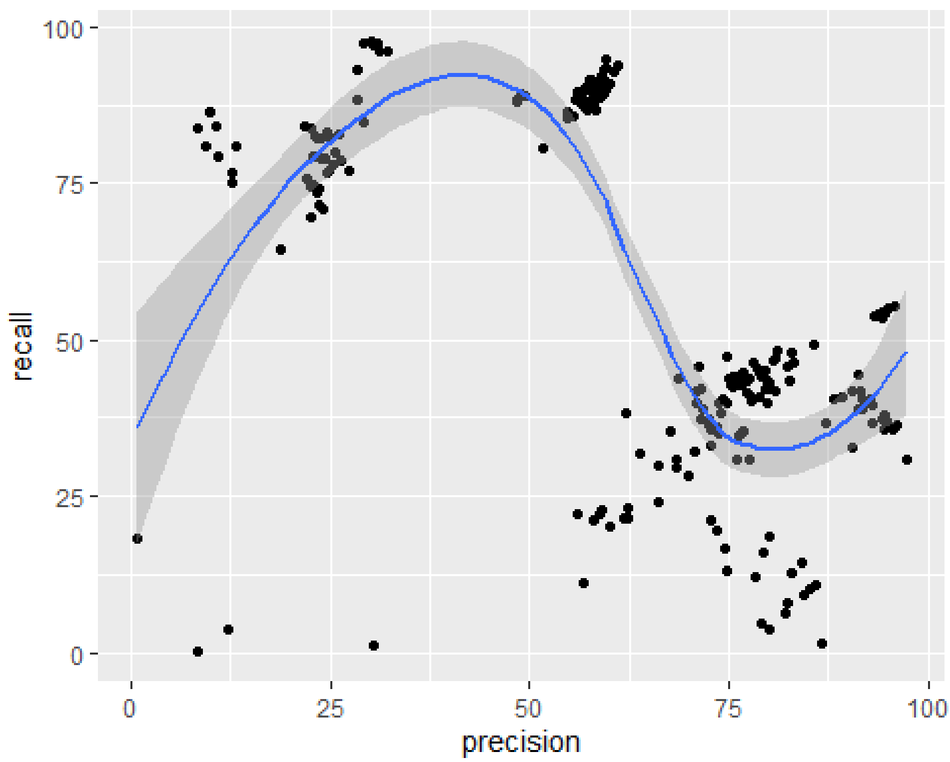

```
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall)) +
  facet_grid(Sampling ~ Technique)
```



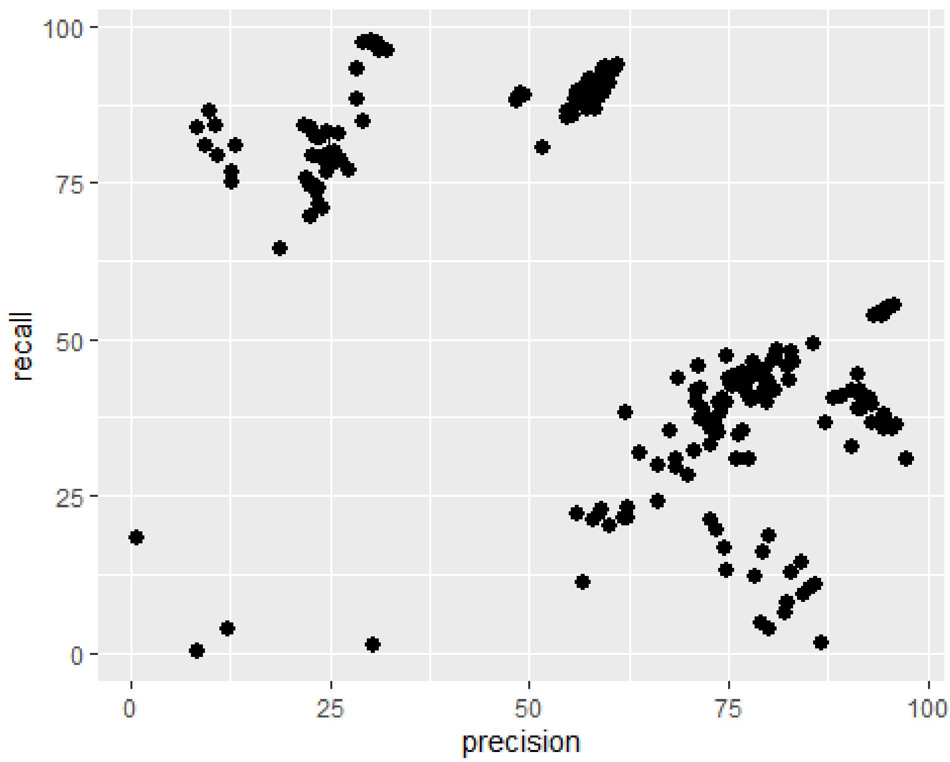

```
ggplot(table, aes(precision, recall)) + geom_boxplot()
```



```r
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall)) +
  geom_smooth(mapping = aes(x = precision, y = recall))
```

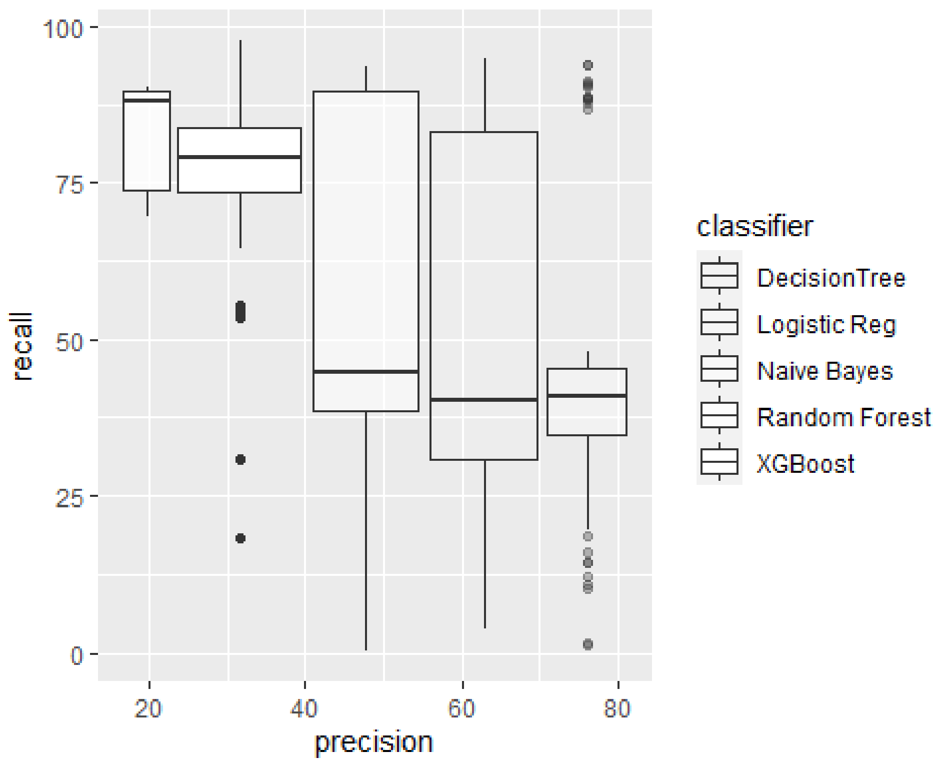



```r
ggplot(data = table) +
  stat_summary(
    mapping = aes(x = precision, y = recall),
    fun.min = min,
    fun.max = max,
    fun = median
  )
```

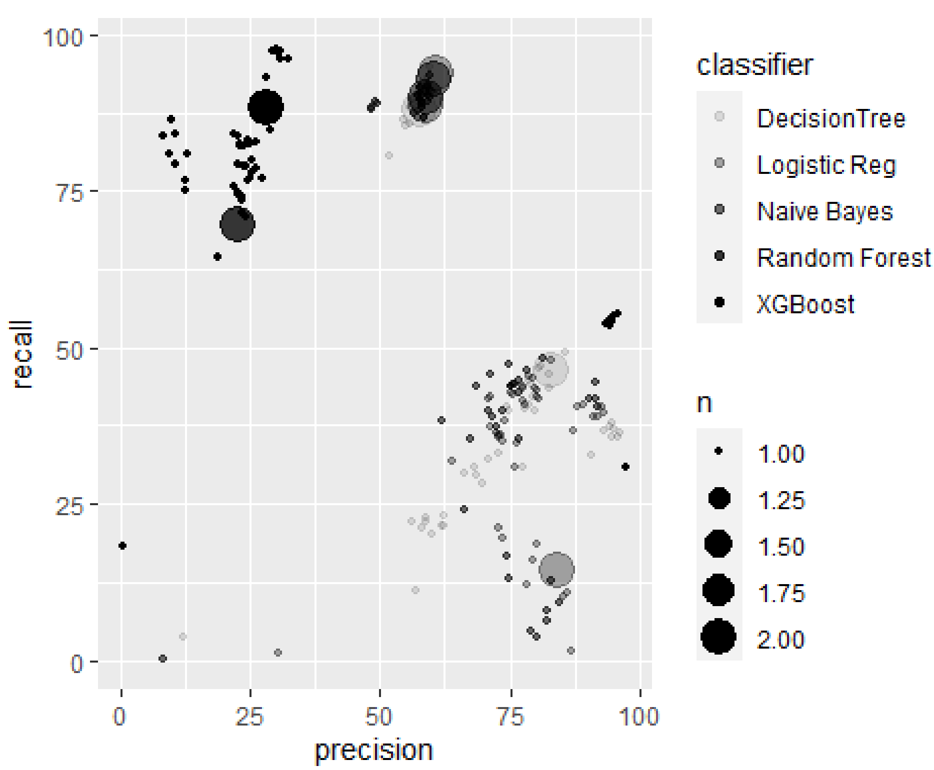



```
# Problematic Code
# ggplot(table, aes(x = precision, y = recall)) +
#   geom_point(size = 2, colour = "grey30") +
#   geom_abline(
#     aes(intercept = a1, slope = a2, colour = -dist),
#     data = table(models, rank(dist) <= 10)
#   )
# Error in x[!nas] : object of type 'closure' is not subsettable
```

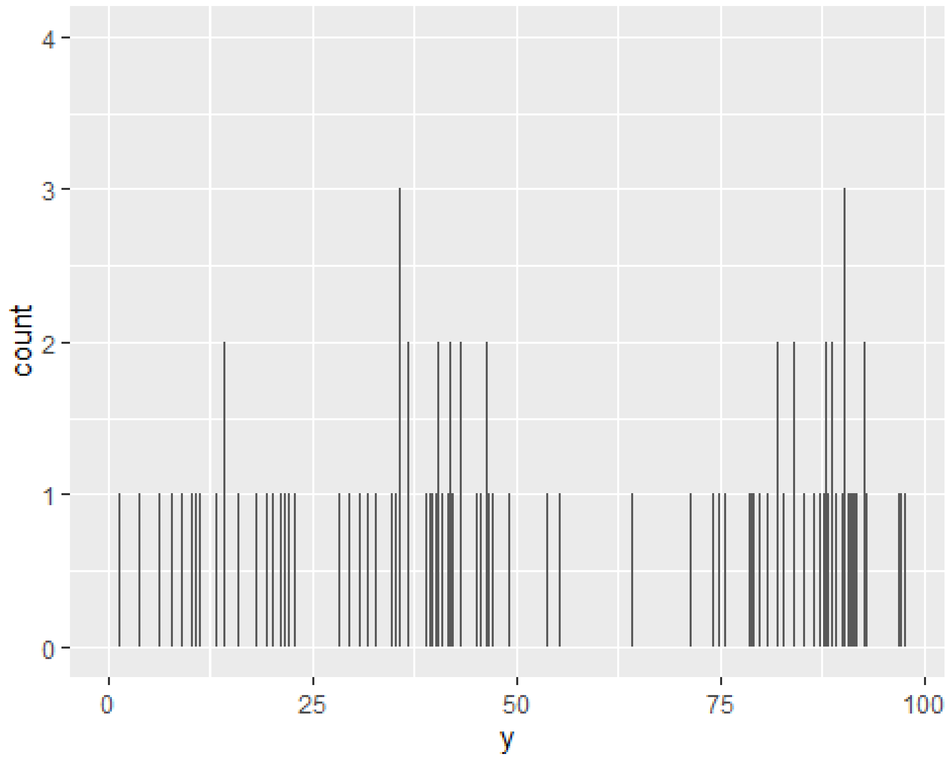
```
ggplot(data = table) +
   geom_boxplot(mapping = aes(x=precision, y=recall, alpha = classifier))
```



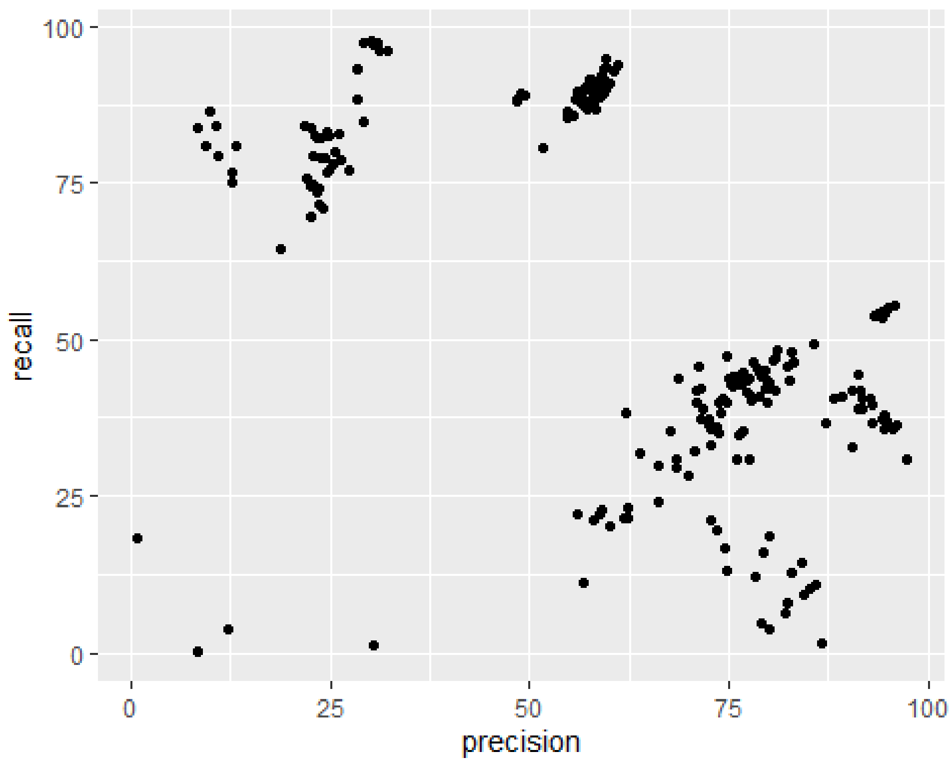

```
ggplot(data = table) +
  geom_count(mapping = aes(x = precision, y = recall, alpha = classifier))
```



```r
x <- precision
y <- recall
ggplot(table) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.1)
```

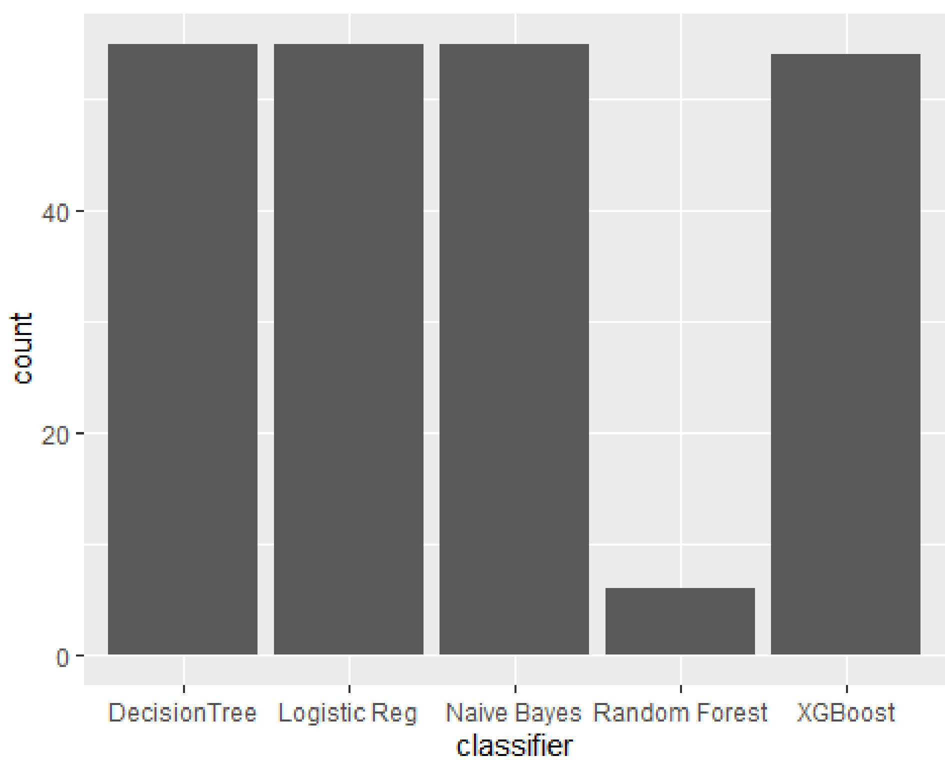



```r
ggplot(data = table) +
  geom_point(mapping = aes(x = precision, y = recall))
```



Chapter 7 EDA

```
ggplot(data = table) +
  geom_bar(mapping = aes(x = classifier))
```



```r
table %>%
  count(cut_width(Precision, 0.5))
```

```
##    cut_width(Precision, 0.5)  n
## 1                [0.25,0.75]  1
## 2                (8.25,8.75]  2
## 3                (9.25,9.75]  1
## 4                (9.75,10.2]  1
## 5                (10.2,10.8]  1
## 6                (10.8,11.2]  1
## 7                (11.8,12.2]  1
## 8                (12.2,12.8]  1
## 9                (12.8,13.2]  2
## 10               (18.2,18.8]  1
## 11               (21.8,22.2]  3
## 12               (22.2,22.8]  5
## 13               (22.8,23.2]  3
## 14               (23.2,23.8]  4
## 15               (23.8,24.2]  2
## 16               (24.2,24.8]  3
## 17               (24.8,25.2]  2
## 18               (25.2,25.8]  2
## 19               (25.8,26.2]  1
## 20               (26.2,26.8]  1
## 21               (26.8,27.2]  1
## 22               (28.2,28.8]  3
## 23               (28.8,29.2]  2
## 24               (29.8,30.2]  1
## 25               (30.2,30.8]  2
## 26               (30.8,31.2]  2
## 27               (32.2,32.8]  1
## 28               (48.2,48.8]  2
## 29               (48.8,49.2]  1
## 30               (49.2,49.8]  1
## 31               (51.2,51.8]  1
## 32               (54.2,54.8]  1
## 33               (54.8,55.2]  1
## 34               (55.2,55.8]  1
## 35               (55.8,56.2]  3
## 36               (56.2,56.8]  4
## 37               (56.8,57.2]  3
## 38               (57.2,57.8]  7
## 39               (57.8,58.2]  9
## 40               (58.2,58.8]  4
## 41               (58.8,59.2] 10
## 42               (59.2,59.8]  6
## 43               (59.8,60.2]  2
## 44               (60.2,60.8]  2
## 45               (60.8,61.2]  2
## 46               (61.8,62.2]  2
```

```
## 47                 (62.2,62.8]  2
## 48                 (63.8,64.2]  1
## 49                 (65.8,66.2]  2
## 50                 (67.2,67.8]  1
## 51                 (68.2,68.8]  3
## 52                 (69.8,70.2]  1
## 53                 (70.2,70.8]  1
## 54                 (70.8,71.2]  3
## 55                 (71.2,71.8]  2
## 56                 (71.8,72.2]  1
## 57                 (72.2,72.8]  4
## 58                 (72.8,73.2]  2
## 59                 (73.2,73.8]  4
## 60                 (73.8,74.2]  1
## 61                 (74.2,74.8]  4
## 62                 (74.8,75.2]  3
## 63                 (75.2,75.8]  3
## 64                 (75.8,76.2]  2
## 65                 (76.2,76.8]  4
## 66                 (77.2,77.8]  5
## 67                 (77.8,78.2]  2
## 68                 (78.2,78.8]  2
## 69                 (78.8,79.2]  4
## 70                 (79.2,79.8]  3
## 71                 (79.8,80.2]  6
## 72                 (80.2,80.8]  1
## 73                 (80.8,81.2]  3
## 74                 (81.8,82.2]  1
## 75                 (82.2,82.8]  3
## 76                 (82.8,83.2]  4
## 77                 (83.8,84.2]  2
## 78                 (84.2,84.8]  1
## 79                 (84.8,85.2]  1
## 80                 (85.2,85.8]  1
## 81                 (85.8,86.2]  1
## 82                 (86.8,87.2]  2
## 83                 (87.8,88.2]  1
## 84                 (88.8,89.2]  1
## 85                 (90.2,90.8]  2
## 86                 (90.8,91.2]  1
## 87                 (91.2,91.8]  2
## 88                 (91.8,92.2]  2
## 89                 (92.2,92.8]  1
## 90                 (92.8,93.2]  2
## 91                 (93.2,93.8]  2
## 92                 (93.8,94.2]  2
## 93                 (94.2,94.8]  5
## 94                 (94.8,95.2]  2
## 95                 (95.2,95.8]  2
```

```
## 96                (95.8,96.2]  1
## 97                (97.2,97.8]  1
```

```
table %>%
  count(Classifier)
```

```
##      Classifier  n
## 1  DecisionTree 55
## 2  Logistic Reg 55
## 3   Naive Bayes 55
## 4 Random Forest  6
## 5       XGBoost 54
```

```
table %>%
  count(cut_width(Recall, 0.5))
```

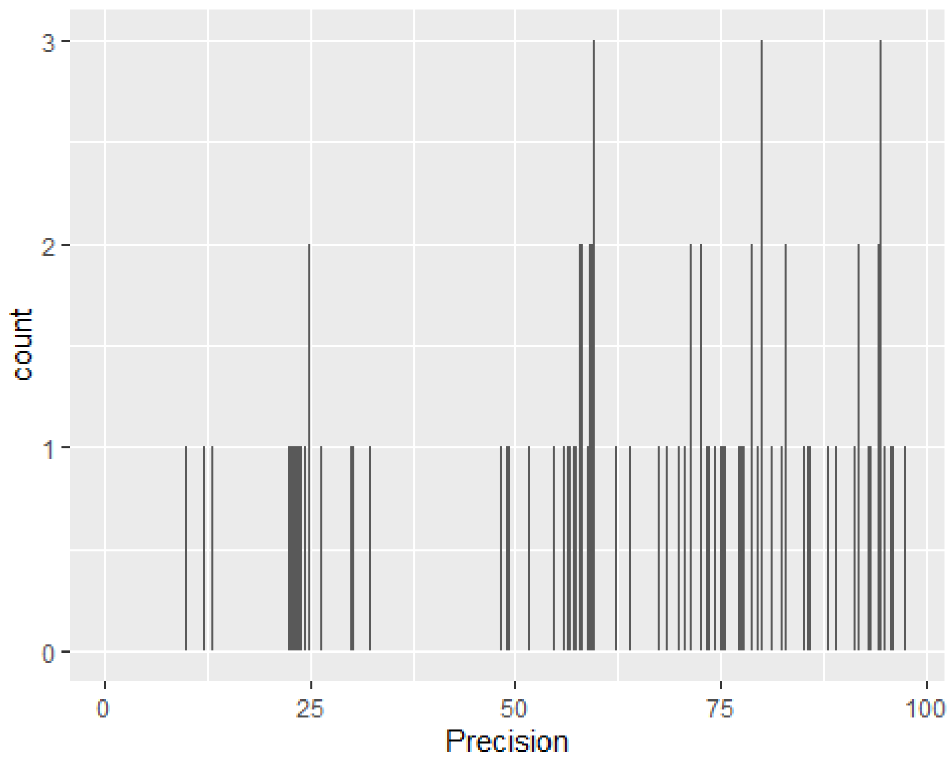
```
##     cut_width(Recall, 0.5) n
## 1              [0.25,0.75] 1
## 2              (1.25,1.75] 2
## 3              (3.75,4.25] 2
## 4              (4.75,5.25] 1
## 5              (6.25,6.75] 1
## 6              (7.75,8.25] 1
## 7              (8.75,9.25] 1
## 8              (10.2,10.8] 1
## 9              (10.8,11.2] 1
## 10             (11.2,11.8] 1
## 11             (11.8,12.2] 1
## 12             (12.2,12.8] 1
## 13             (13.2,13.8] 1
## 14             (14.2,14.8] 2
## 15             (15.8,16.2] 1
## 16             (16.2,16.8] 1
## 17             (18.2,18.8] 2
## 18             (19.2,19.8] 1
## 19             (20.2,20.8] 1
## 20             (20.8,21.2] 2
## 21             (21.2,21.8] 2
## 22             (21.8,22.2] 2
## 23             (22.8,23.2] 2
## 24             (23.8,24.2] 1
## 25             (28.2,28.8] 1
## 26             (29.2,29.8] 1
## 27             (29.8,30.2] 1
## 28             (30.8,31.2] 4
## 29             (31.8,32.2] 1
## 30             (32.2,32.8] 1
## 31             (32.8,33.2] 2
## 32             (34.8,35.2] 3
```

```
## 33               (35.2,35.8] 1
## 34               (35.8,36.2] 5
## 35               (36.2,36.8] 2
## 36               (36.8,37.2] 5
## 37               (37.2,37.8] 1
## 38               (37.8,38.2] 3
## 39               (38.8,39.2] 3
## 40               (39.2,39.8] 1
## 41               (39.8,40.2] 4
## 42               (40.2,40.8] 6
## 43               (40.8,41.2] 2
## 44               (41.2,41.8] 2
## 45               (41.8,42.2] 5
## 46               (42.2,42.8] 4
## 47               (43.2,43.8] 7
## 48               (43.8,44.2] 4
## 49               (44.2,44.8] 3
## 50               (44.8,45.2] 1
## 51               (45.2,45.8] 2
## 52               (45.8,46.2] 1
## 53               (46.2,46.8] 4
## 54               (46.8,47.2] 2
## 55               (47.8,48.2] 1
## 56               (48.2,48.8] 1
## 57               (48.8,49.2] 1
## 58               (53.2,53.8] 1
## 59               (53.8,54.2] 4
## 60               (54.2,54.8] 1
## 61               (54.8,55.2] 1
## 62               (55.2,55.8] 1
## 63               (64.2,64.8] 1
## 64               (69.2,69.8] 2
## 65               (70.2,70.8] 1
## 66               (71.2,71.8] 1
## 67               (73.2,73.8] 1
## 68               (73.8,74.2] 1
## 69               (74.2,74.8] 2
## 70               (74.8,75.2] 1
## 71               (75.2,75.8] 1
## 72               (76.8,77.2] 4
## 73               (77.8,78.2] 1
## 74               (78.2,78.8] 1
## 75               (78.8,79.2] 3
## 76               (79.2,79.8] 1
## 77               (79.8,80.2] 1
## 78               (80.2,80.8] 1
## 79               (80.8,81.2] 2
## 80               (81.8,82.2] 2
## 81               (82.2,82.8] 2
## 82               (82.8,83.2] 2
```

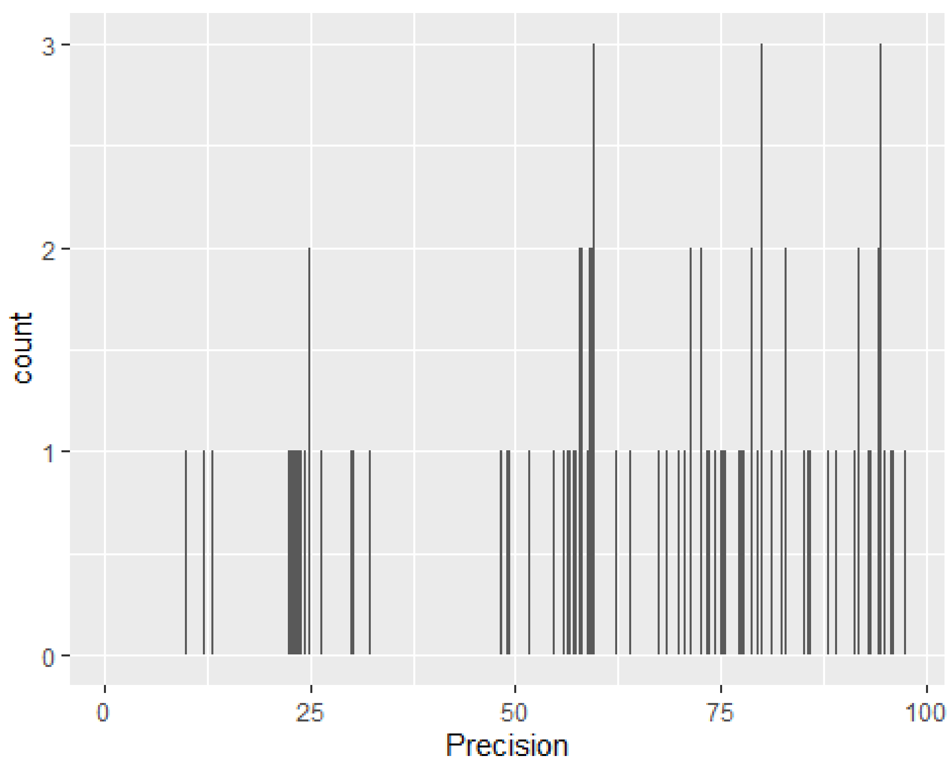
```
## 83              (83.8,84.2] 5
## 84              (84.2,84.8] 1
## 85              (85.2,85.8] 2
## 86              (86.2,86.8] 3
## 87              (86.8,87.2] 1
## 88              (87.2,87.8] 3
## 89              (87.8,88.2] 5
## 90              (88.2,88.8] 8
## 91              (88.8,89.2] 5
## 92              (89.2,89.8] 4
## 93              (89.8,90.2] 4
## 94              (90.2,90.8] 6
## 95              (90.8,91.2] 2
## 96              (91.2,91.8] 4
## 97              (91.8,92.2] 2
## 98              (92.8,93.2] 4
## 99              (93.2,93.8] 3
## 100             (94.2,94.8] 1
## 101             (95.8,96.2] 2
## 102             (96.8,97.2] 1
## 103             (97.2,97.8] 3
```

```
smaller <- table %>%
  filter(Classifier > 60)
```

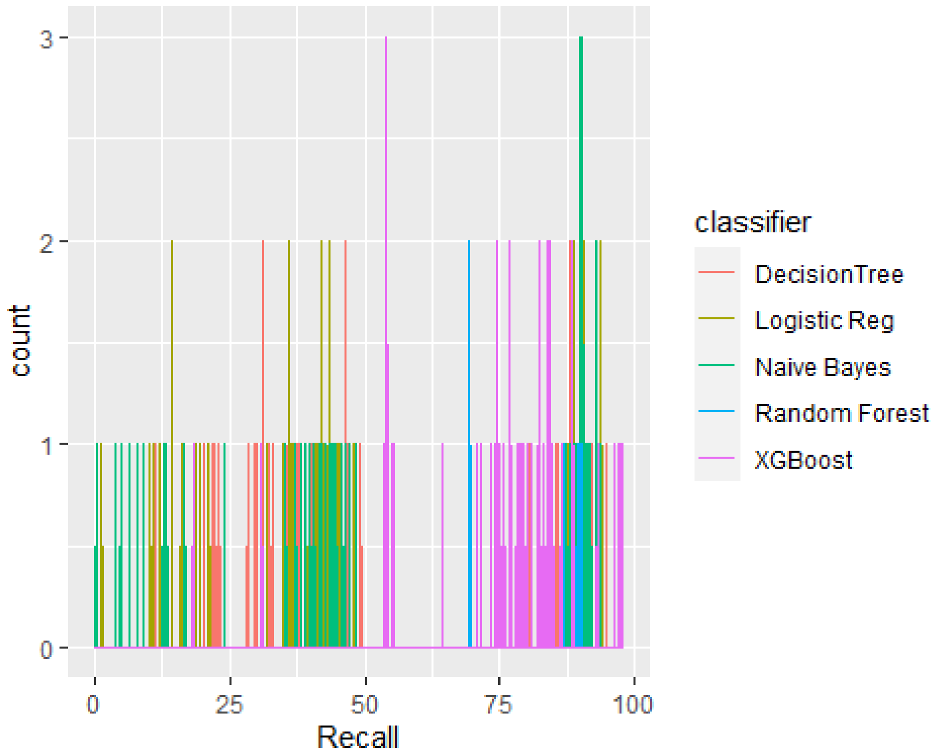
```
ggplot(data = smaller, mapping = aes(x = Precision)) +
  geom_histogram(binwidth = 0.1)
```



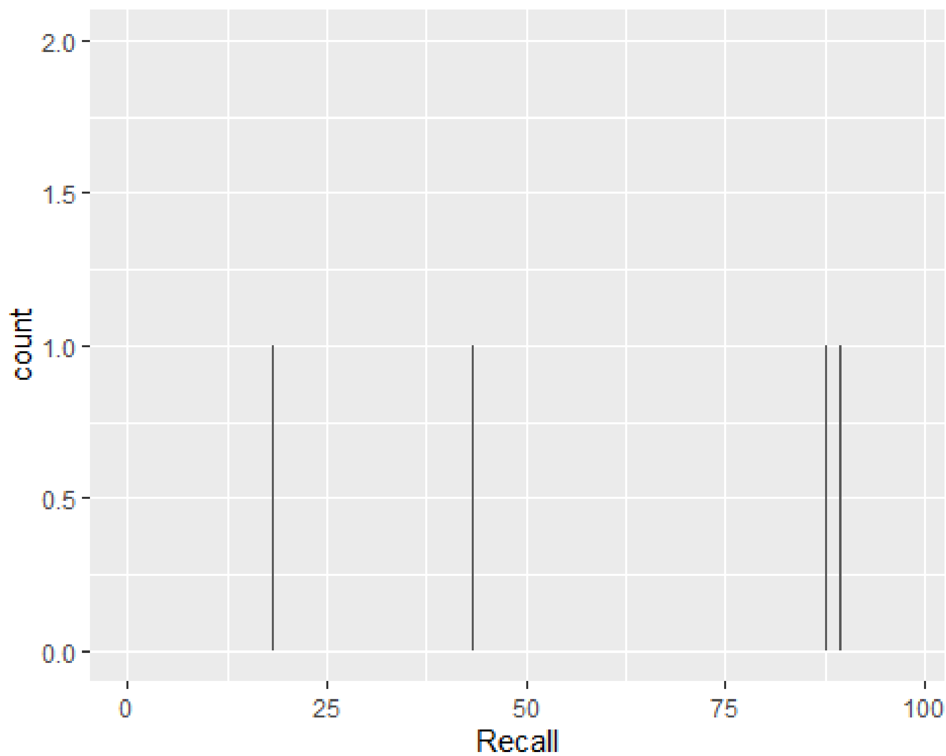

```
ggplot(data = smaller, mapping = aes(x = Precision)) +
  geom_histogram(binwidth = 0.1)
```



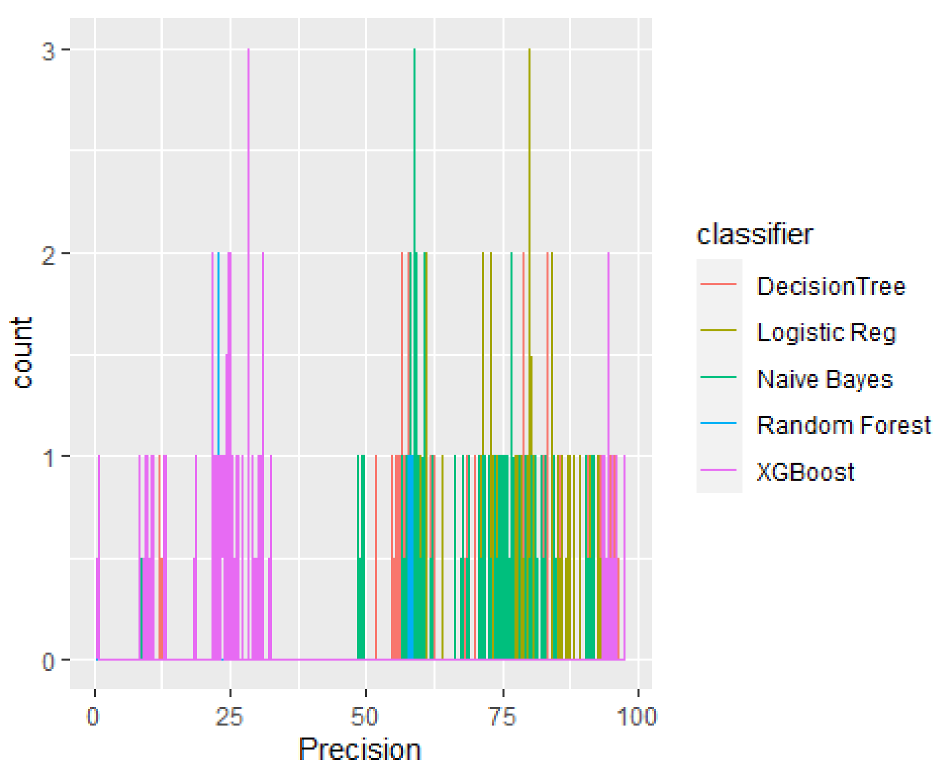
```r
ggplot(data = smaller, mapping = aes(x = Recall, colour = classifier)) +
  geom_freqpoly(binwidth = 0.1)
```



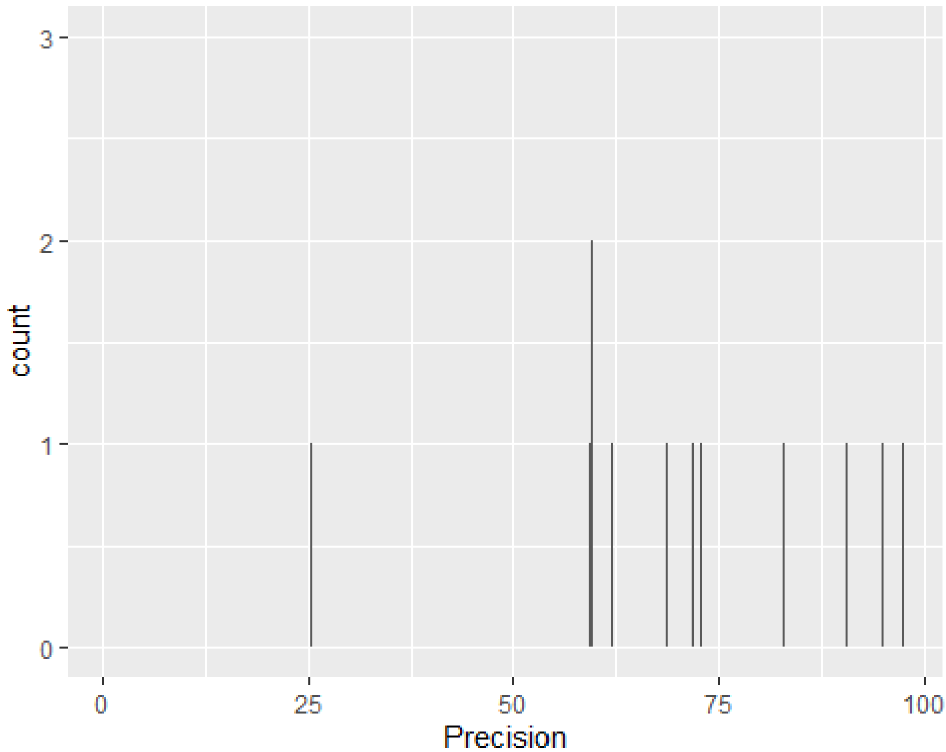

```r
ggplot(data = smaller, mapping = aes(x = Recall)) +
  geom_histogram(binwidth = 0.01)
```



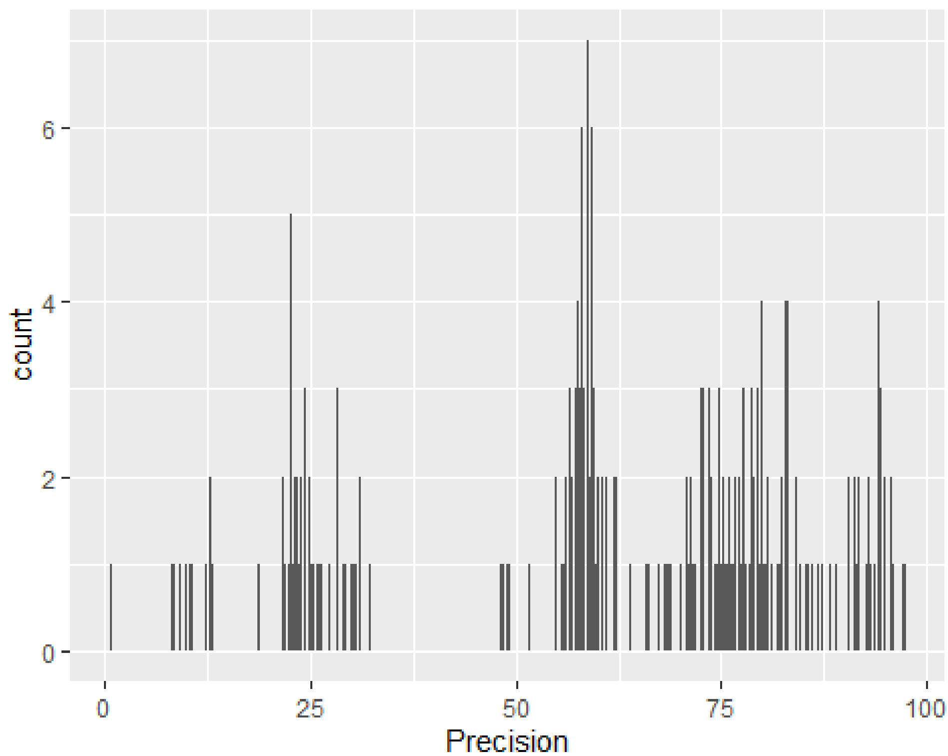

```r
ggplot(data = smaller, mapping = aes(x = Precision, colour = classifier)) +
  geom_freqpoly(binwidth = 0.1)
```



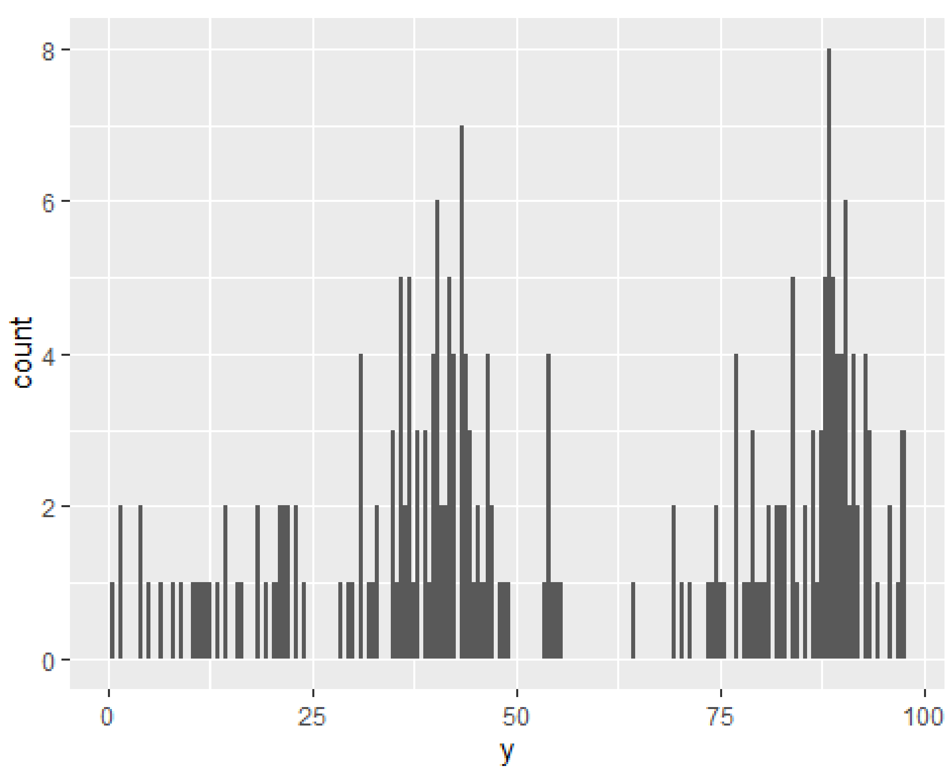
```
ggplot(data = smaller, mapping = aes(x = Precision)) +
  geom_histogram(binwidth = 0.01)
```



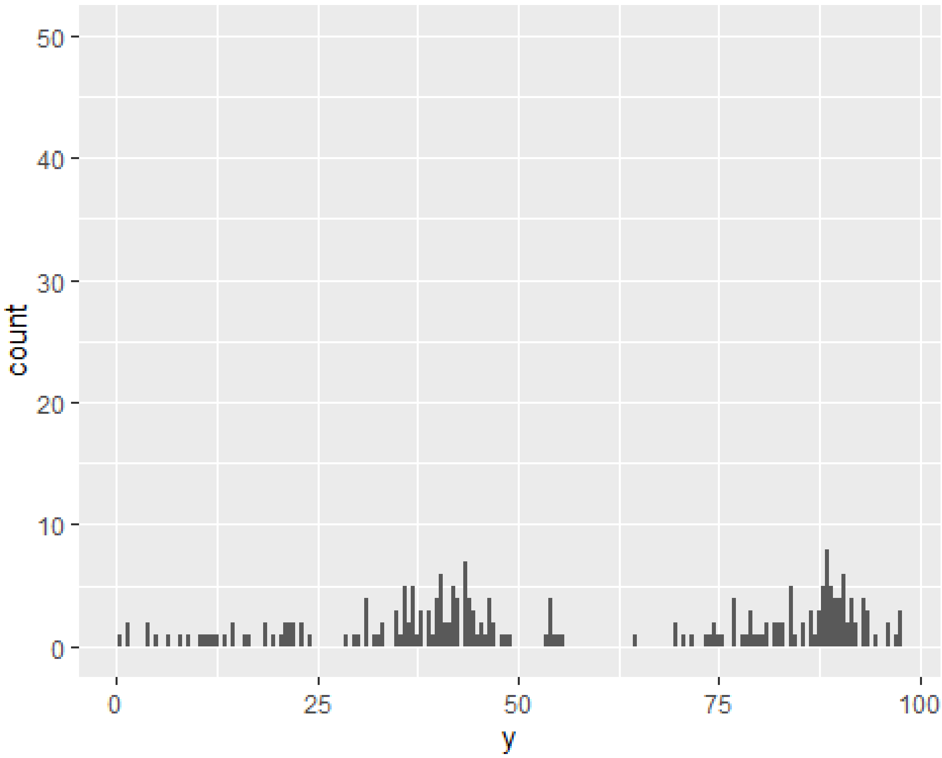

```
ggplot(data = smaller, mapping = aes(x = Precision)) +
  geom_histogram(binwidth = 0.25)
```




```
ggplot(table) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5)
```

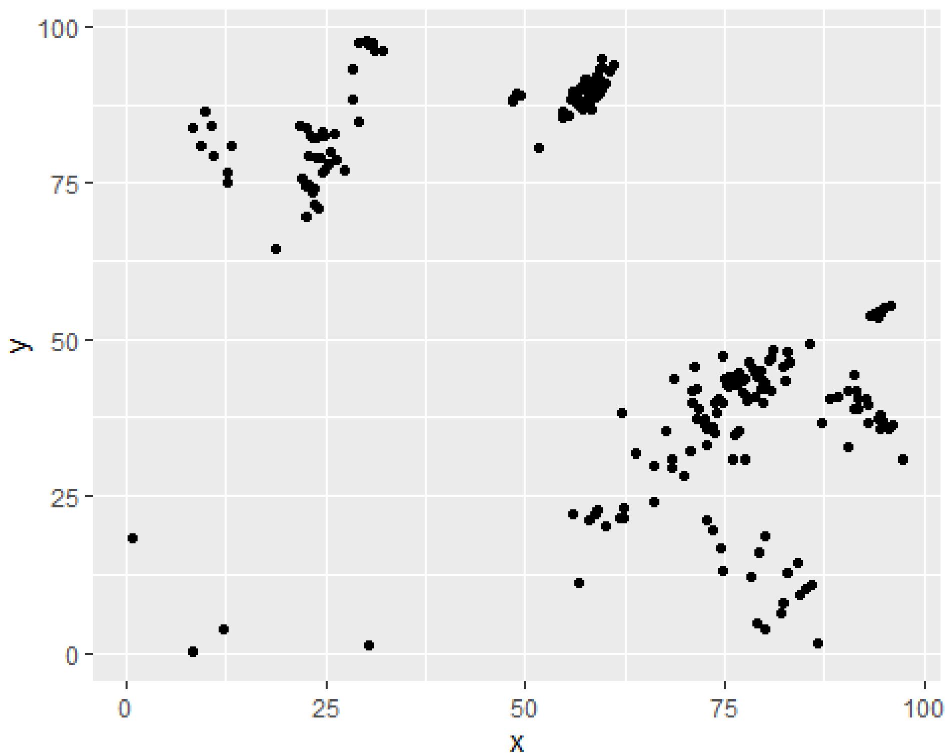


```r
ggplot(table) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  coord_cartesian(ylim = c(0, 50))
```

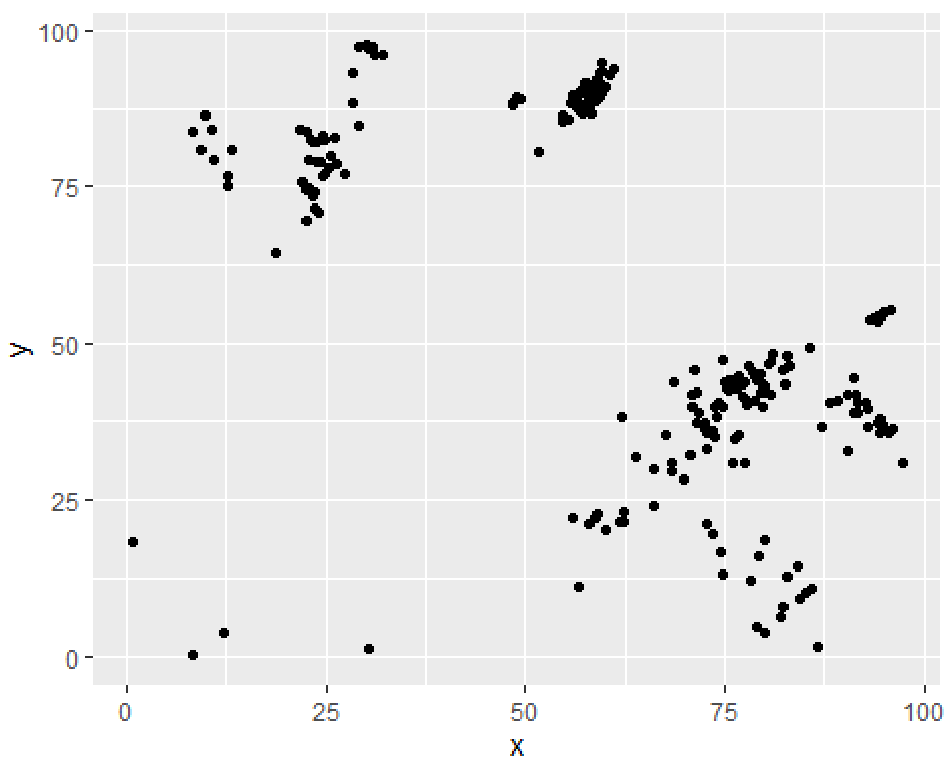


unusual <- table %>% # filter(y < 30 | y > 60) %>% # select(x, y) %>% # arrange(y) # unusual

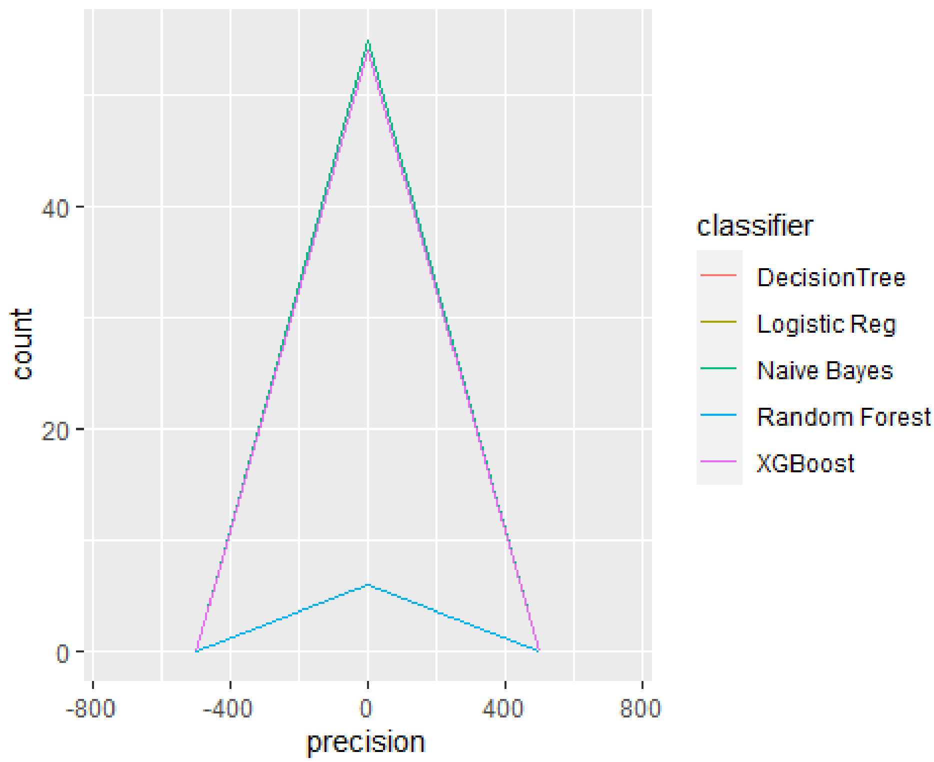
```r
ggplot(data = table, mapping = aes(x = x, y = y)) +
  geom_point()
```



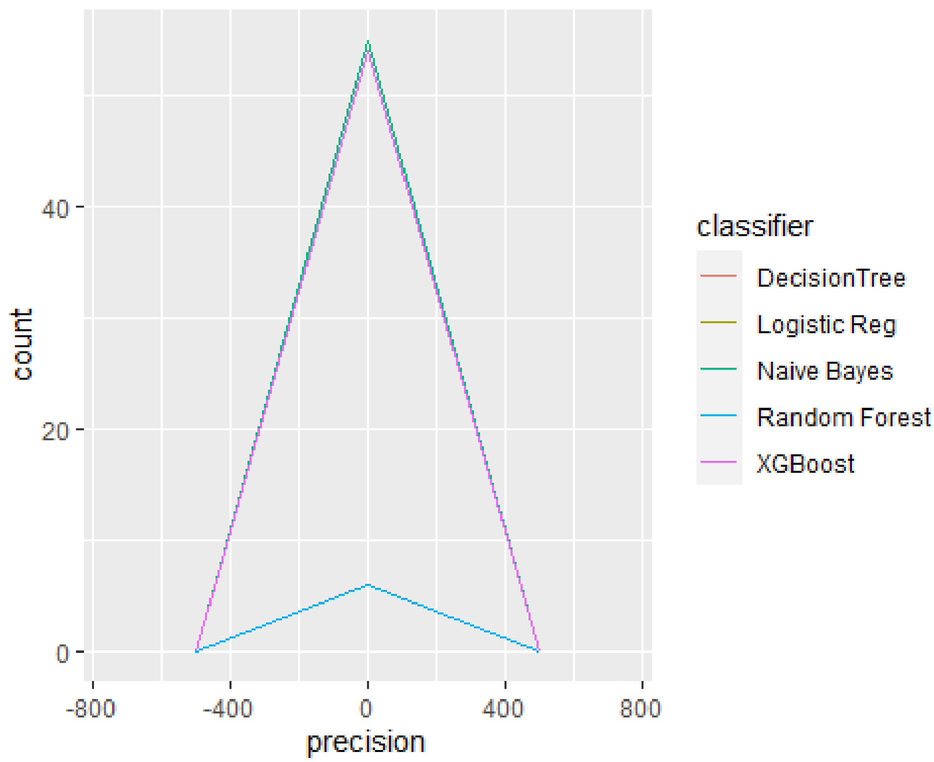

```
ggplot(data = table, mapping = aes(x = x, y = y)) +
  geom_point(na.rm = TRUE)
```



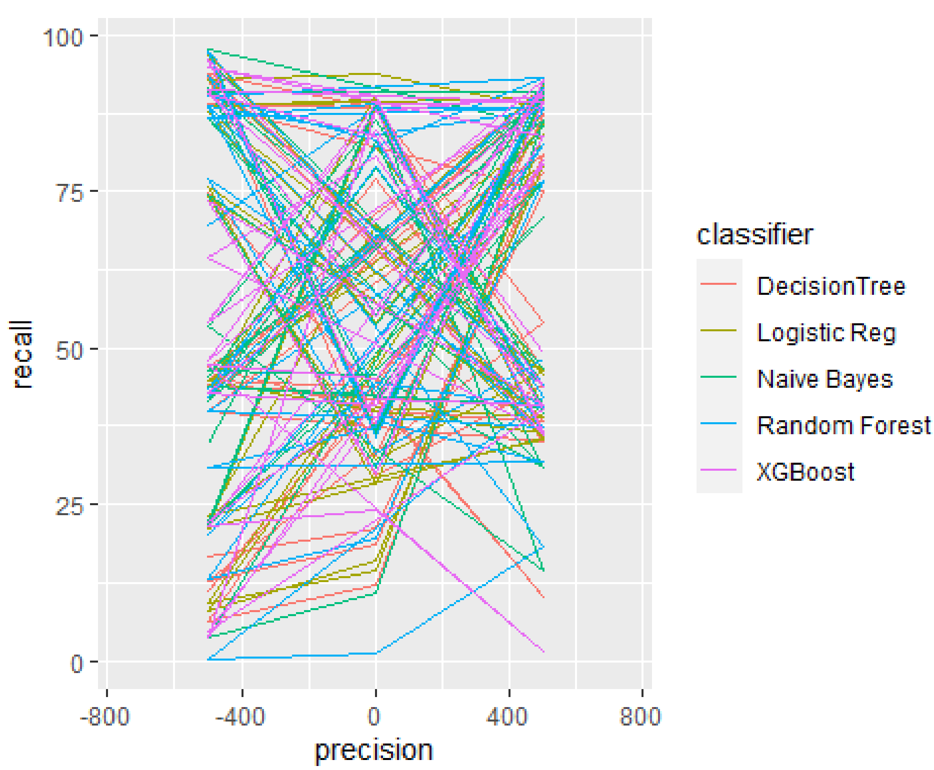
```
ggplot(data = table, mapping = aes(x = precision)) +
  geom_freqpoly(mapping = aes(colour = classifier), binwidth = 500)
```



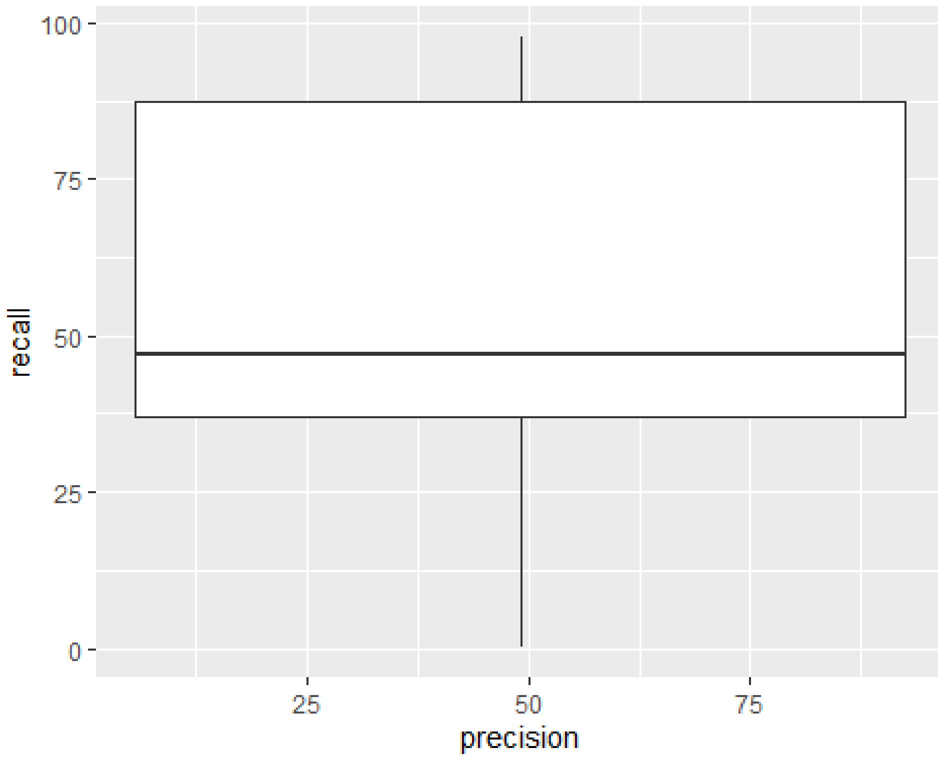

```
ggplot(data = table, mapping = aes(x = precision)) +
  geom_freqpoly(mapping = aes(colour = classifier), binwidth = 500)
```



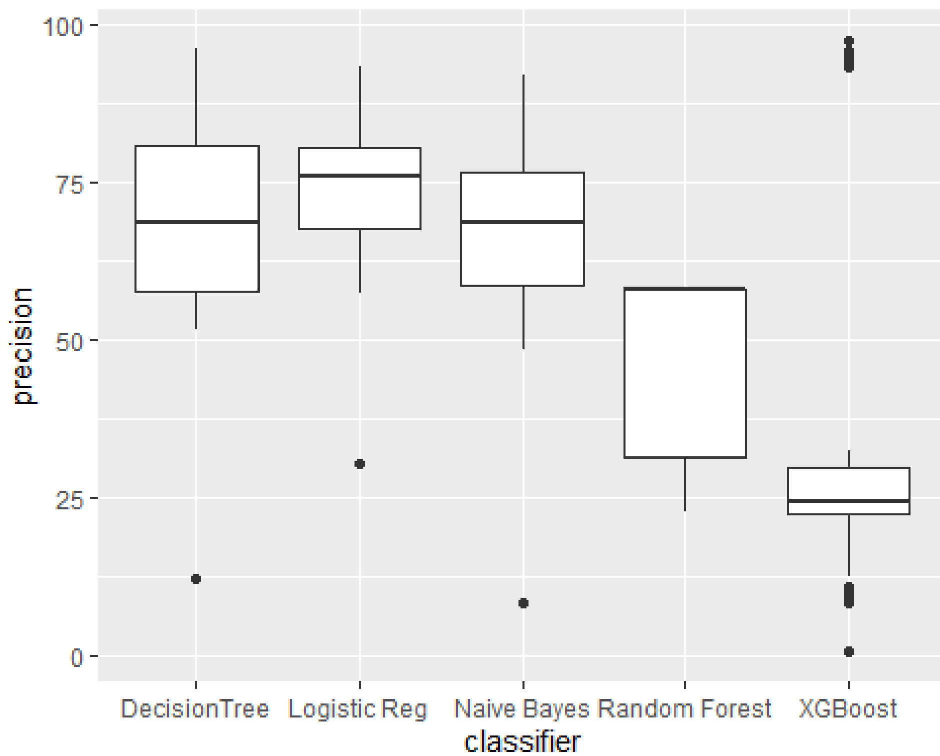

```
ggplot(data = table, mapping = aes(x = precision, y = ..recall..)) +
  geom_freqpoly(mapping = aes(colour = classifier), binwidth = 500)
```



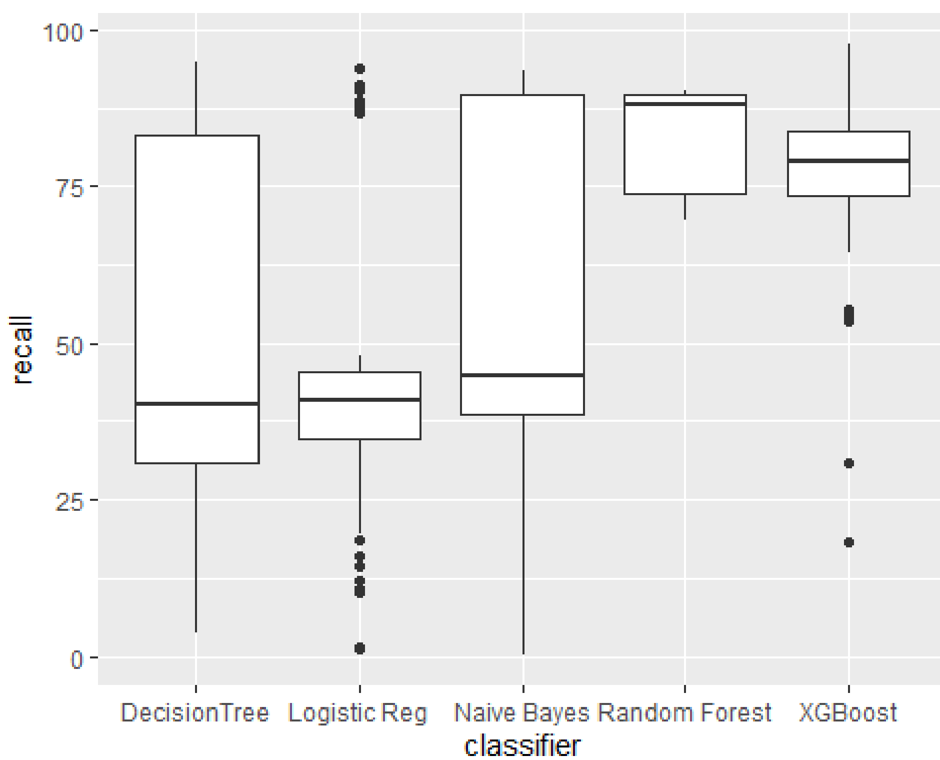
```r
ggplot(data = table, mapping = aes(x = precision, y = recall)) +
  geom_boxplot()
```



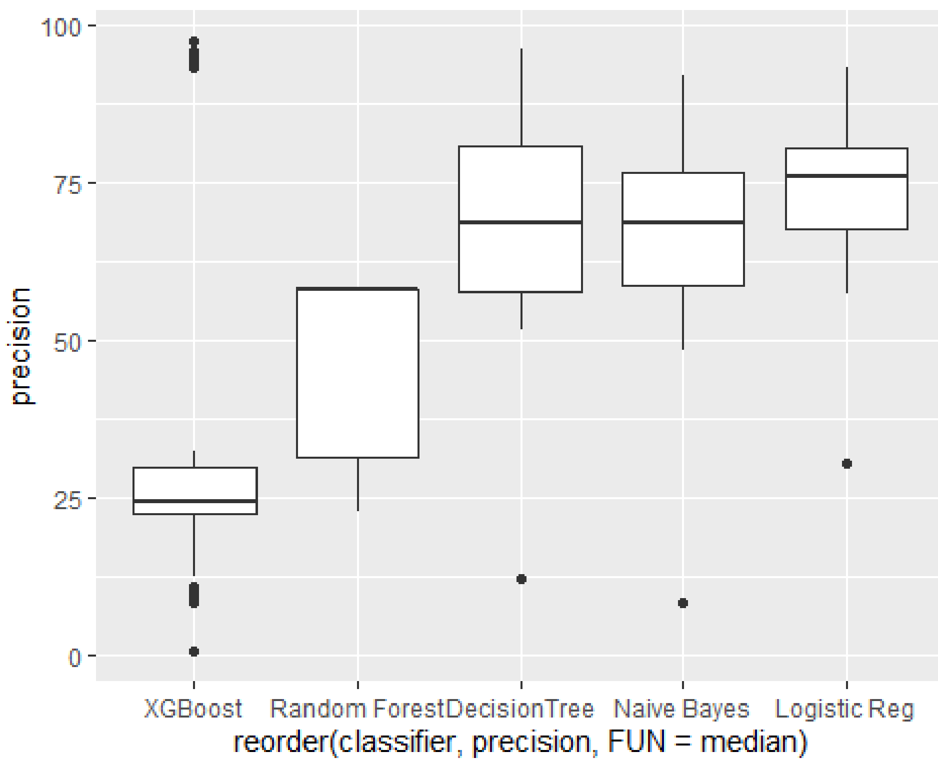

```r
ggplot(data = table, mapping = aes(x = classifier, y = precision)) +
  geom_boxplot()
```



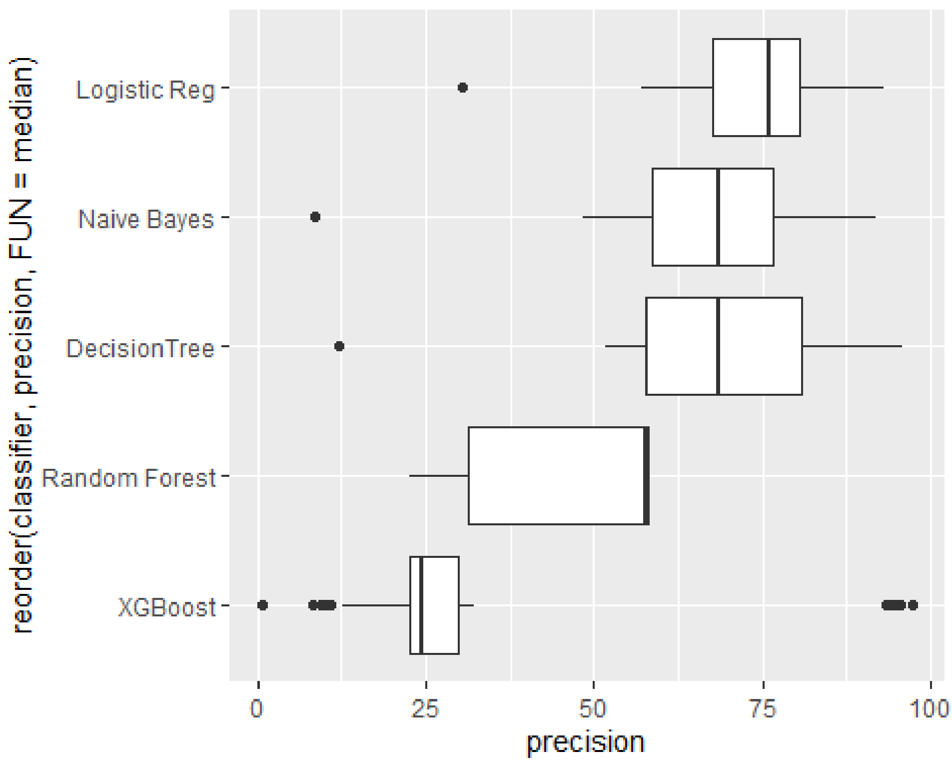

```
ggplot(data = table, mapping = aes(x = classifier, y = recall)) +
  geom_boxplot()
```



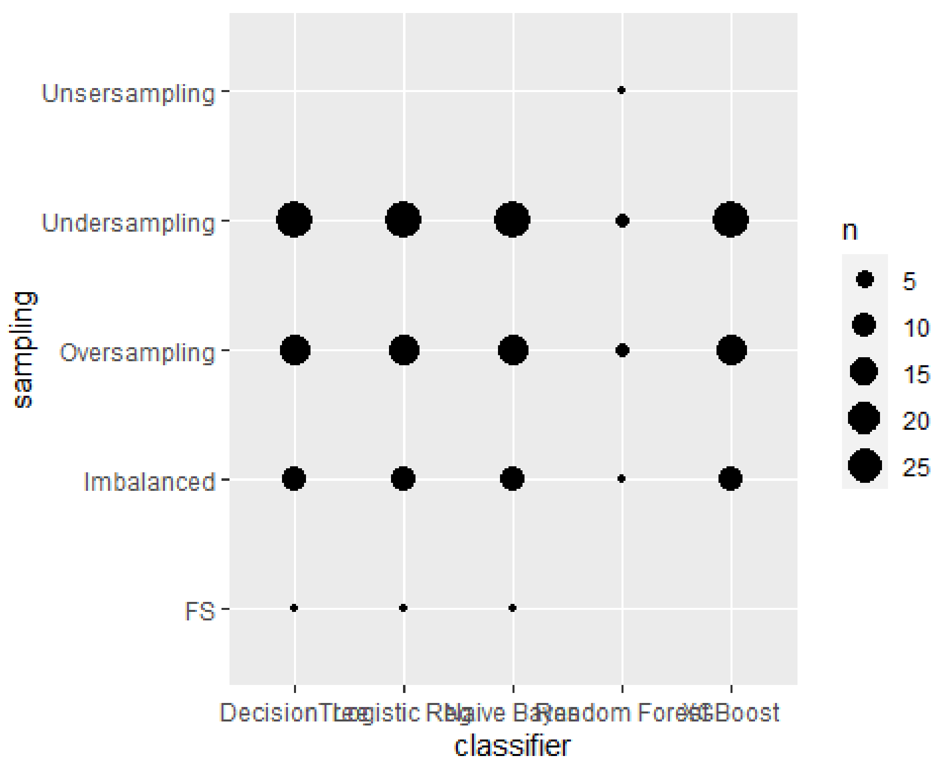
```r
ggplot(data = table) +
  geom_boxplot(mapping = aes(x = reorder(classifier, precision, FUN =
median), y = precision))
```




```r
ggplot(data = table) +
  geom_boxplot(mapping = aes(x = reorder(classifier, precision, FUN =
median), y = precision)) +
  coord_flip()
```



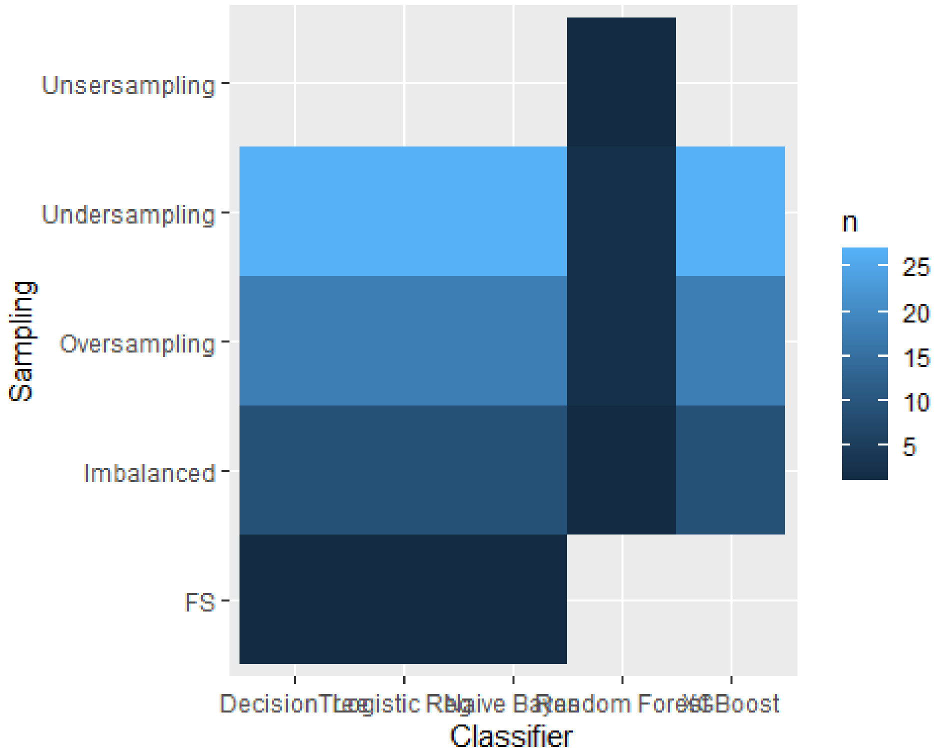

```r
ggplot(data = table) +
  geom_count(mapping = aes(x = classifier, y = sampling))
```



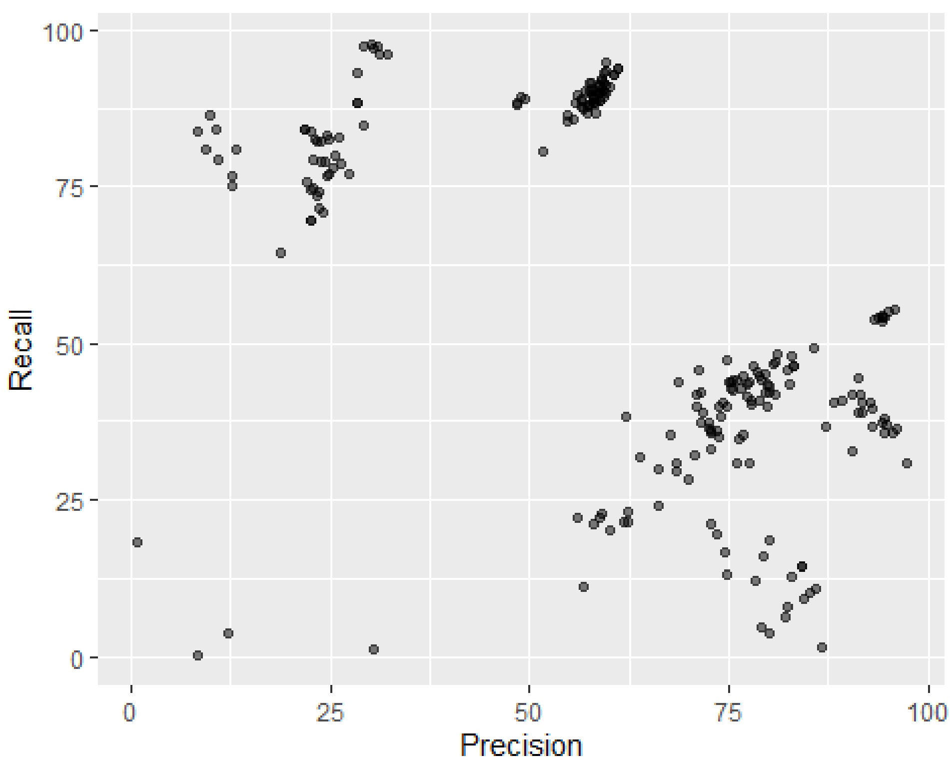
```r
table %>%
  count(Classifier, Sampling)
```

```
##       Classifier      Sampling  n
## 1   DecisionTree            FS  1
## 2   DecisionTree    Imbalanced  9
## 3   DecisionTree  Oversampling 18
## 4   DecisionTree Undersampling 27
## 5   Logistic Reg            FS  1
## 6   Logistic Reg    Imbalanced  9
## 7   Logistic Reg  Oversampling 18
## 8   Logistic Reg Undersampling 27
## 9    Naive Bayes            FS  1
## 10   Naive Bayes    Imbalanced  9
## 11   Naive Bayes  Oversampling 18
## 12   Naive Bayes Undersampling 27
## 13 Random Forest    Imbalanced  1
## 14 Random Forest  Oversampling  2
## 15 Random Forest Undersampling  2
## 16 Random Forest Unsersampling  1
## 17       XGBoost    Imbalanced  9
## 18       XGBoost  Oversampling 18
## 19       XGBoost Undersampling 27
```

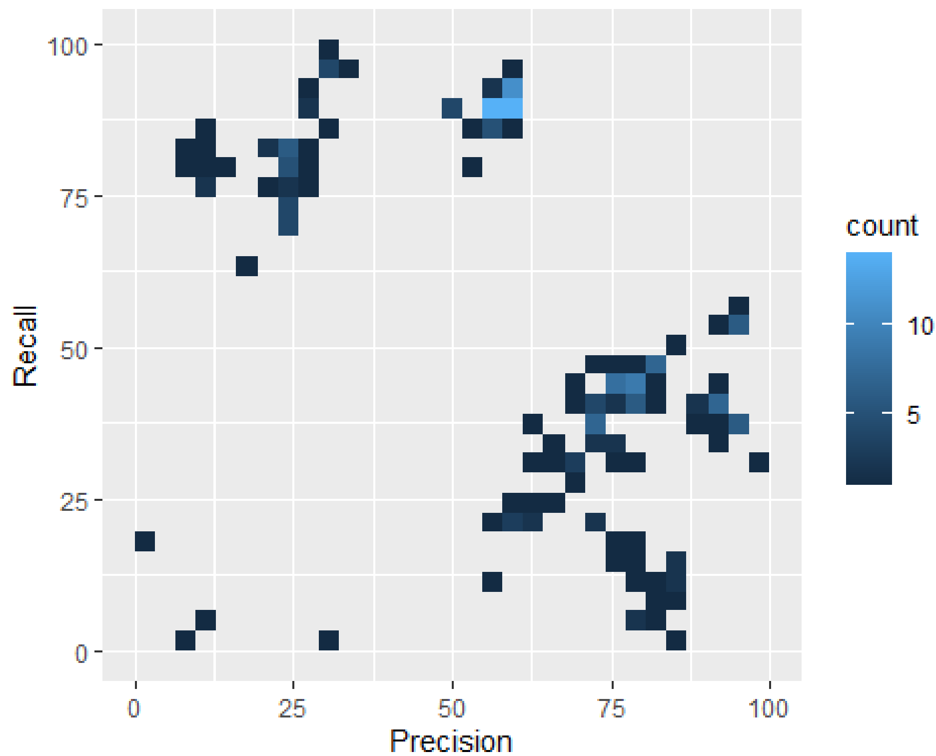
```r
table %>%
  count(Classifier, Sampling) %>%
  ggplot(mapping = aes(x = Classifier, y = Sampling)) +
  geom_tile(mapping = aes(fill = n))
```




```r
ggplot(data = table) +
  geom_point(mapping = aes(x = Precision, y = Recall), alpha = 50 / 100)
```

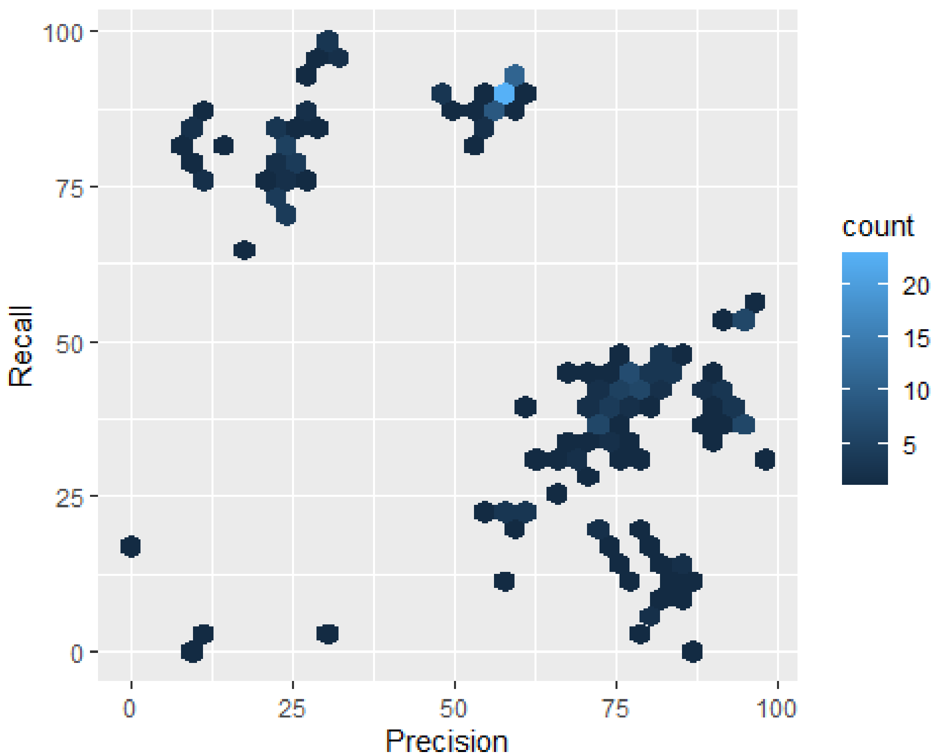


```r
ggplot(data = smaller) +
  geom_bin2d(mapping = aes(x = Precision, y = Recall))
```

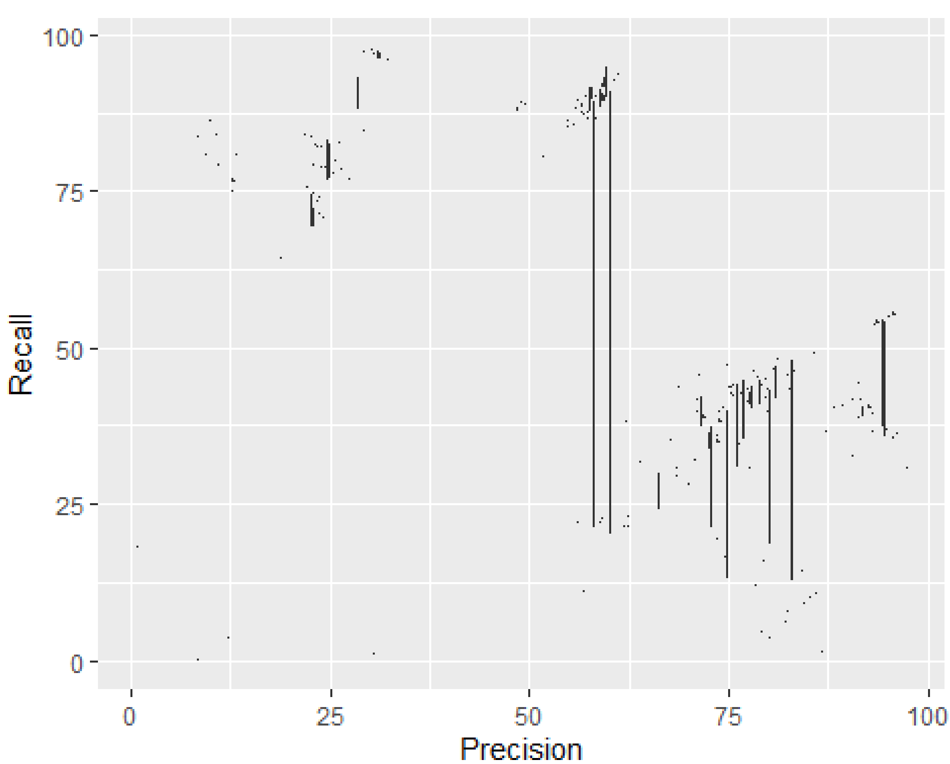



```r
# install.packages("hexbin")
library(hexbin)
```

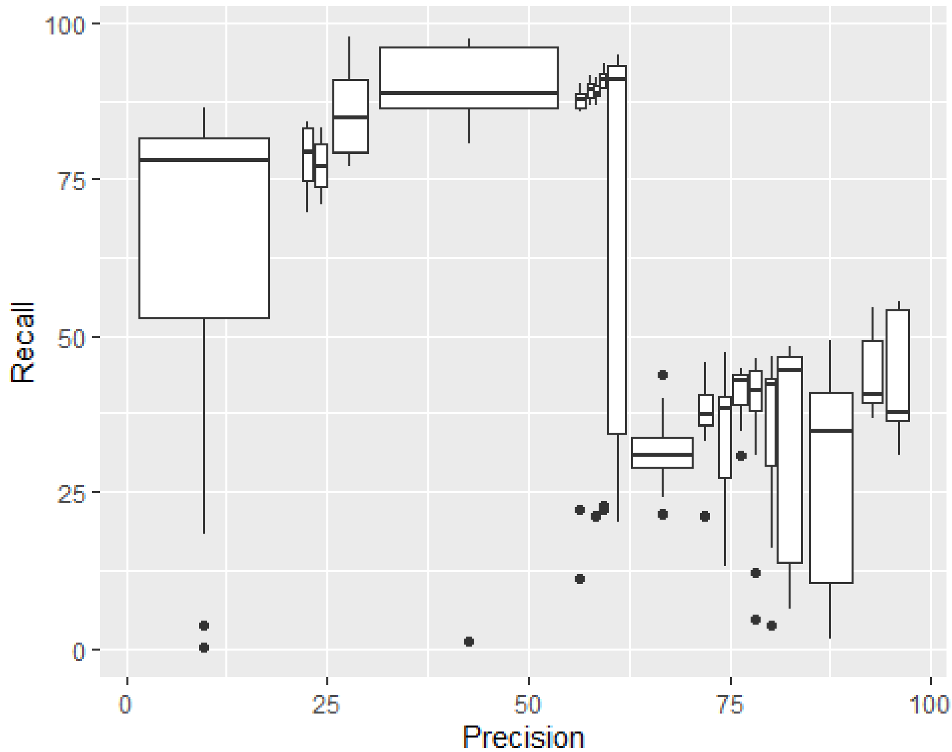
```r
ggplot(data = smaller) +
  geom_hex(mapping = aes(x = Precision, y = Recall))
```



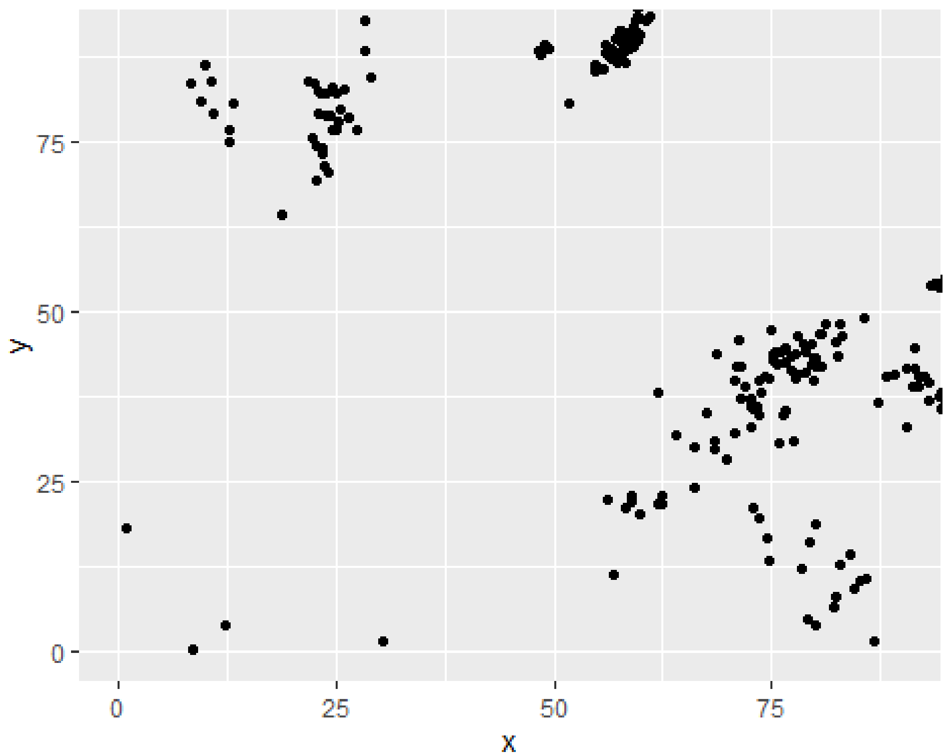

```r
ggplot(data = smaller, mapping = aes(x = Precision, y = Recall)) +
  geom_boxplot(mapping = aes(group = cut_width(Precision, 0.1)))
```



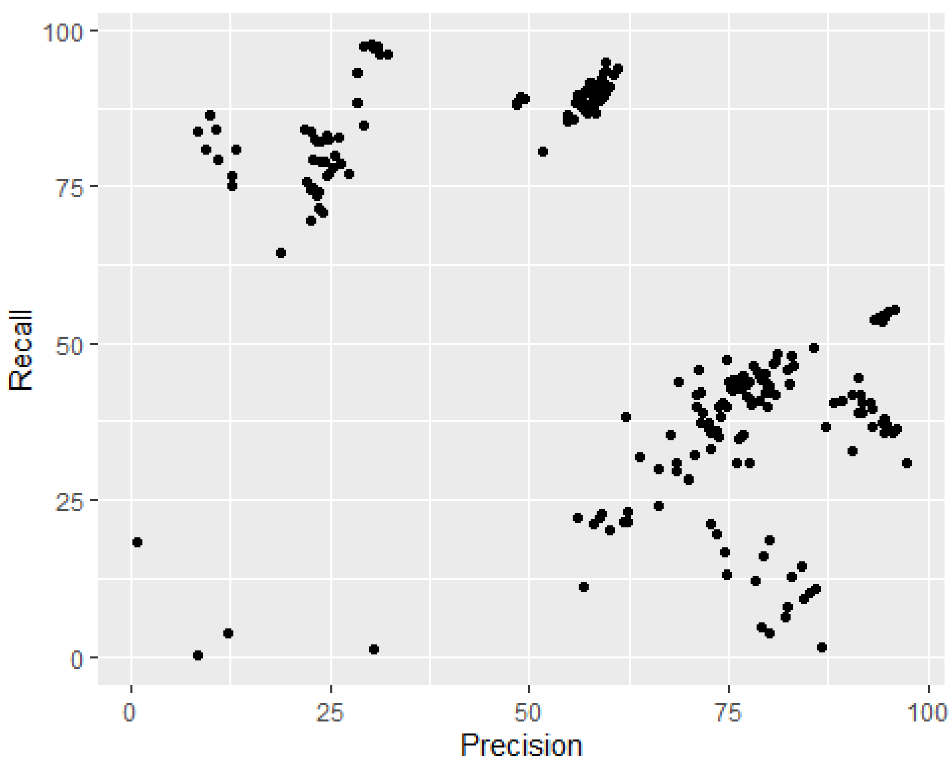
```
ggplot(data = smaller, mapping = aes(x = Precision, y = Recall)) +
  geom_boxplot(mapping = aes(group = cut_number(Precision, 20)))
```




```
ggplot(data = table) +
  geom_point(mapping = aes(x = x, y = y)) +
  coord_cartesian(xlim = c(0, 90), ylim = c(0, 90))
```



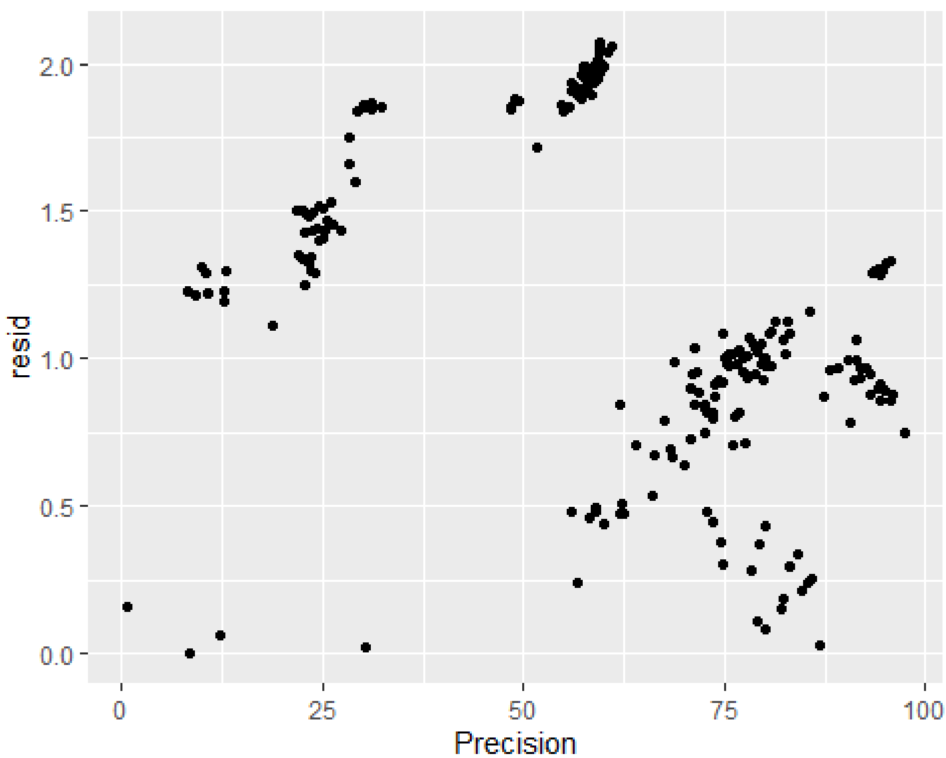

```
ggplot(data = table) +
  geom_point(mapping = aes(x = Precision, y = Recall))
```



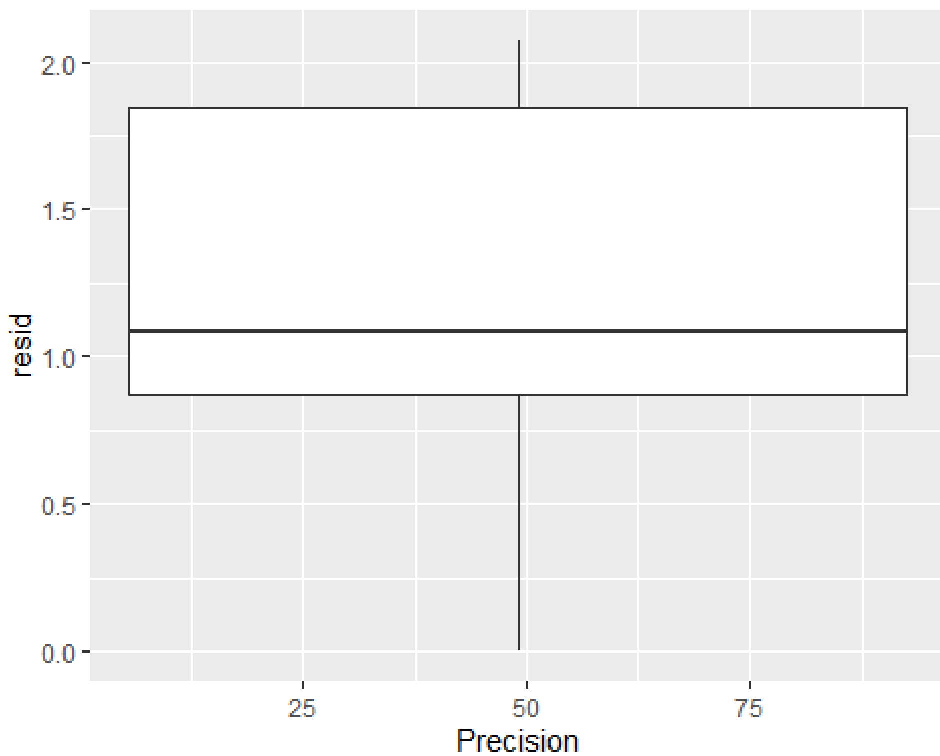
```
library(modelr)
```

```
mod <- lm(log(Recall) ~ log(Precision), data = table)

table1 <- table %>%
  add_residuals(mod) %>%
  mutate(resid = exp(resid))
```

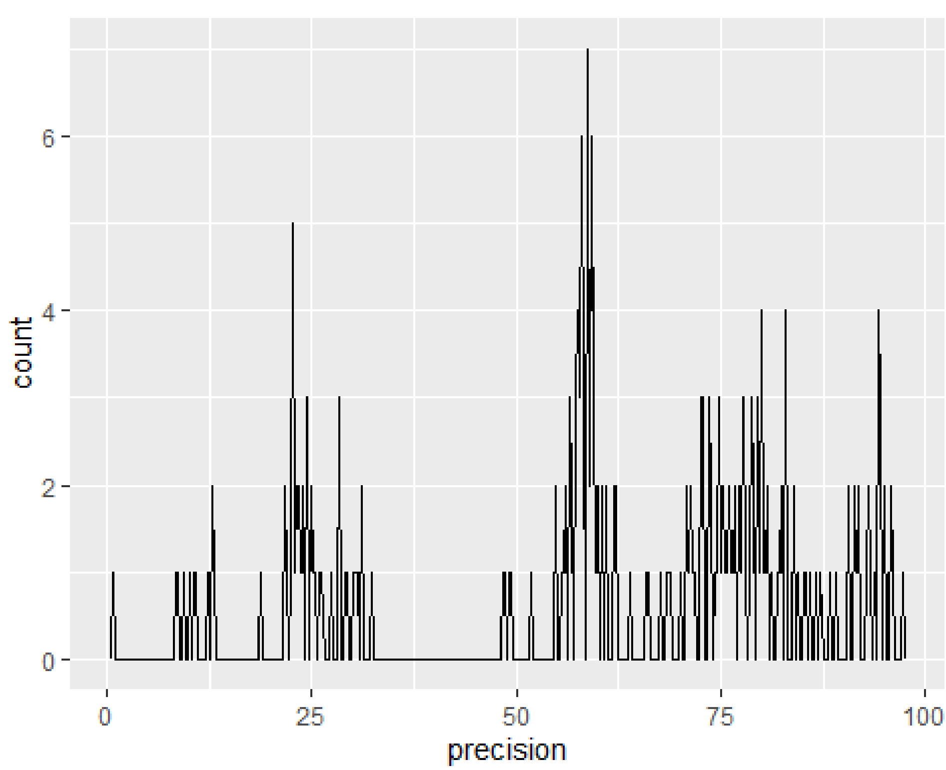
```
ggplot(data = table1) +
  geom_point(mapping = aes(x = Precision, y = resid))
```



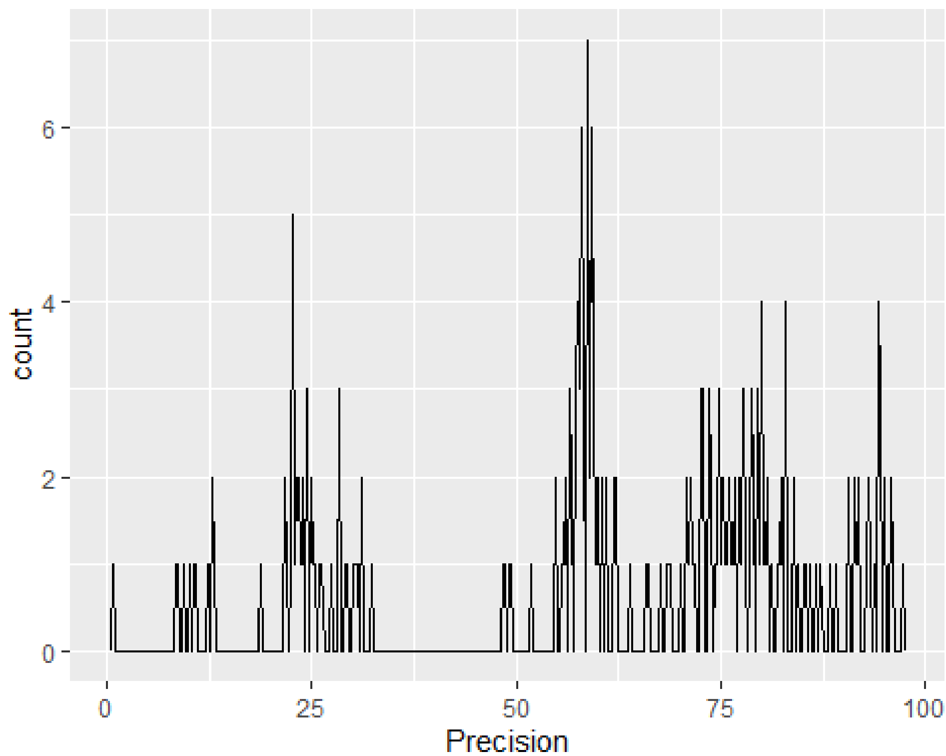

```
ggplot(data = table1) +
  geom_boxplot(mapping = aes(x = Precision, y = resid))
```



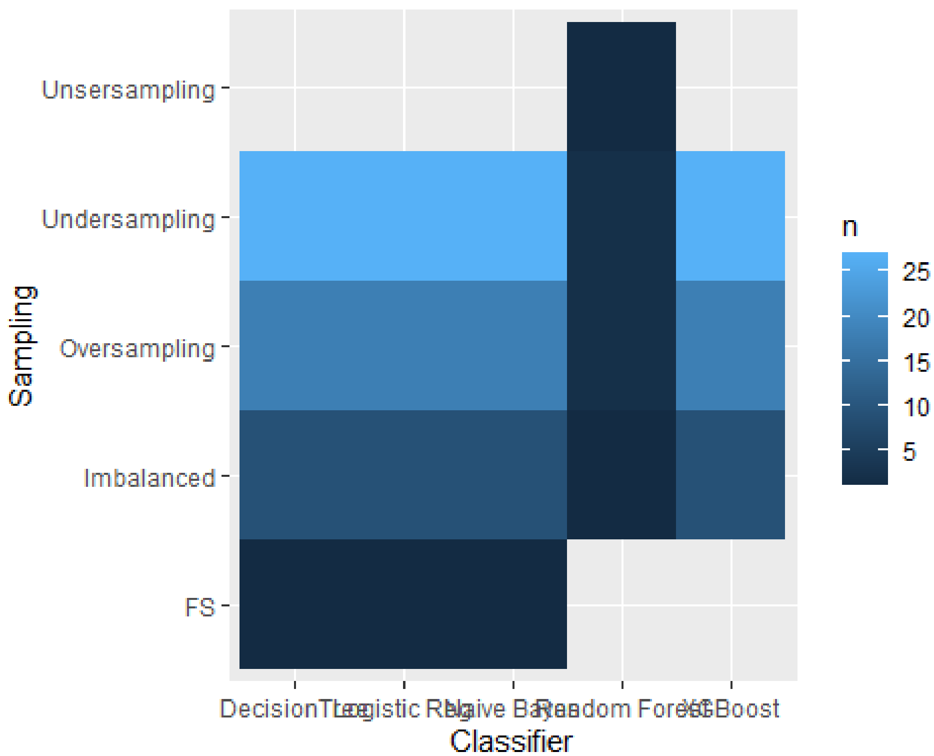

```
ggplot(data = table, mapping = aes(x = precision)) +
  geom_freqpoly(binwidth = 0.25)
```



```
ggplot(table, aes(Precision)) +
  geom_freqpoly(binwidth = 0.25)
```




```
table %>%
  count(Classifier, Sampling) %>%
  ggplot(aes(Classifier, Sampling, fill = n)) +
  geom_tile()
```



Chapter 9 Wrangle

```
table %>%
  count(Precision) %>%
  filter(n > 1)
```

```
##    Precision n
## 1      21.81 2
## 2      22.65 2
## 3      24.91 2
## 4      28.27 2
## 5      57.47 2
## 6      58.71 2
## 7      58.76 3
## 8      59.57 2
## 9      60.53 2
## 10     60.96 2
## 11     83.06 2
## 12     84.04 2
## 13     94.27 2
```

```
table %>%
  count(Recall) %>%
  filter(n > 1)
```

```
##    Recall n
## 1   14.30 2
## 2   46.47 2
## 3   69.45 2
## 4   88.06 2
## 5   88.45 2
## 6   88.66 2
## 7   88.80 2
## 8   89.86 2
## 9   92.82 2
## 10  93.67 2
```

```
table %>%
  count(Precision, Recall) %>%
  filter(n > 1)
```

```
##   Precision Recall n
## 1     22.65  69.45 2
## 2     28.27  88.45 2
## 3     57.47  88.06 2
## 4     58.71  89.86 2
## 5     58.76  88.66 2
## 6     60.53  92.82 2
## 7     60.96  93.67 2
## 8     83.06  46.47 2
## 9     84.04  14.30 2
```

Chapter 23 Model Basics

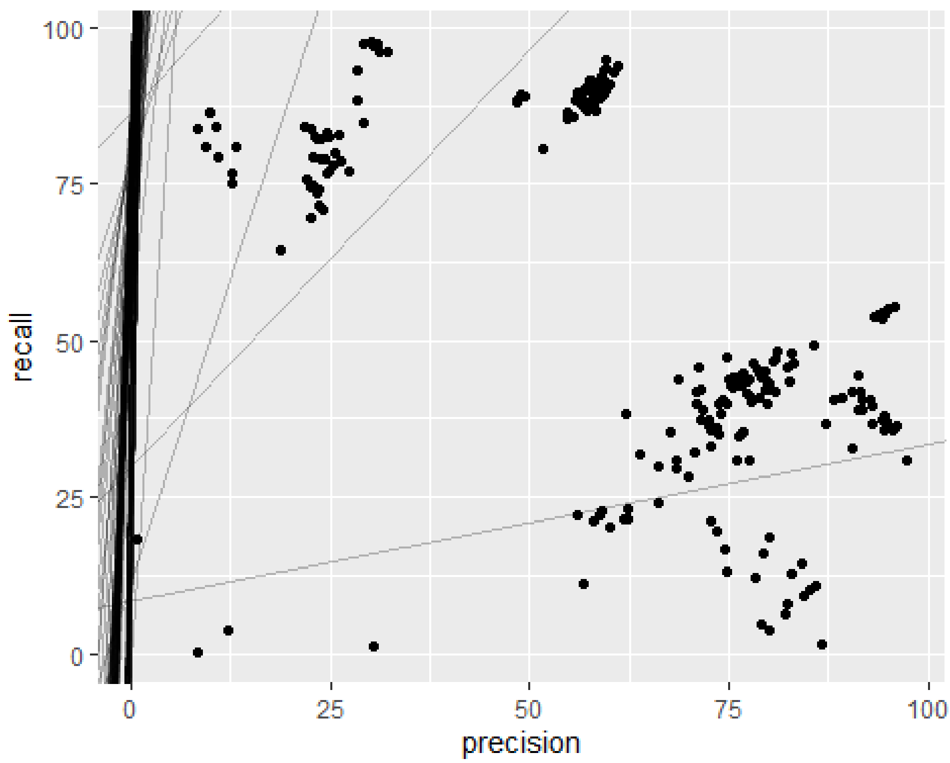
```
x <- precision
y <- recall
# models <- tibble(
#   a1 = c(precision),
#   a2 = c(recall)
# )
```

```
models <- tibble(
  a1 = precision,
  a2 = recall
)
models
```

```
## # A tibble: 225 × 2
##        a1    a2
##     <dbl> <dbl>
##   1  74.5  16.7
```

```
## 2 72.8 21.2
## 3 12.7 75.0
## 4 59 22.8
## 5 22.6 69.4
## 6 76.7 35.4
## 7 76.4 34.8
## 8 26.0 82.9
## 9 77.5 31
## 10 22.6 69.4
## # … with 215 more rows
```

```
ggplot(table, aes(x=precision, y=recall)) +
  geom_abline(aes(intercept = a1, slope = a2), data = models, alpha = 1/4) +
  geom_point()
```



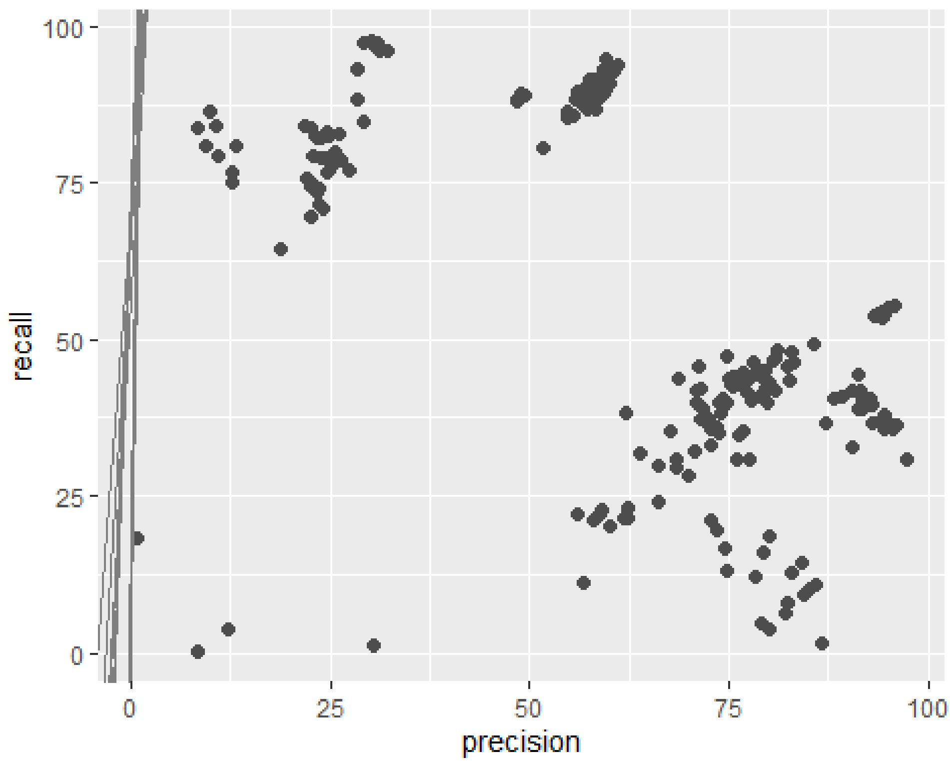

```
model1 <- function(a, data) {
  a[1] + data$x * a[2]
}

# model1(c(7, 1.5), table)
#>  [1]  8.5  8.5  8.5 10.0 10.0 10.0 11.5 11.5 11.5 13.0 13.0 13.0 14.5 14.5
14.5
#> [16] 16.0 16.0 16.0 17.5 17.5 17.5 19.0 19.0 19.0 20.5 20.5 20.5 22.0 22.0
22.0

measure_distance <- function(mod, data) {
  diff <- data$y - model1(mod, data)
```

```
  sqrt(mean(diff ^ 2))
}

table_dist <- function(a1, a2) {
  measure_distance(c(a1, a2), table)
}
# test2 <- c(precision[1],recall[2])
# test2
# measure_distance(c(7.5, 1), table)

models <- models %>%
  mutate(dist = purrr::map2_dbl(a1, a2, table_dist))
models
```

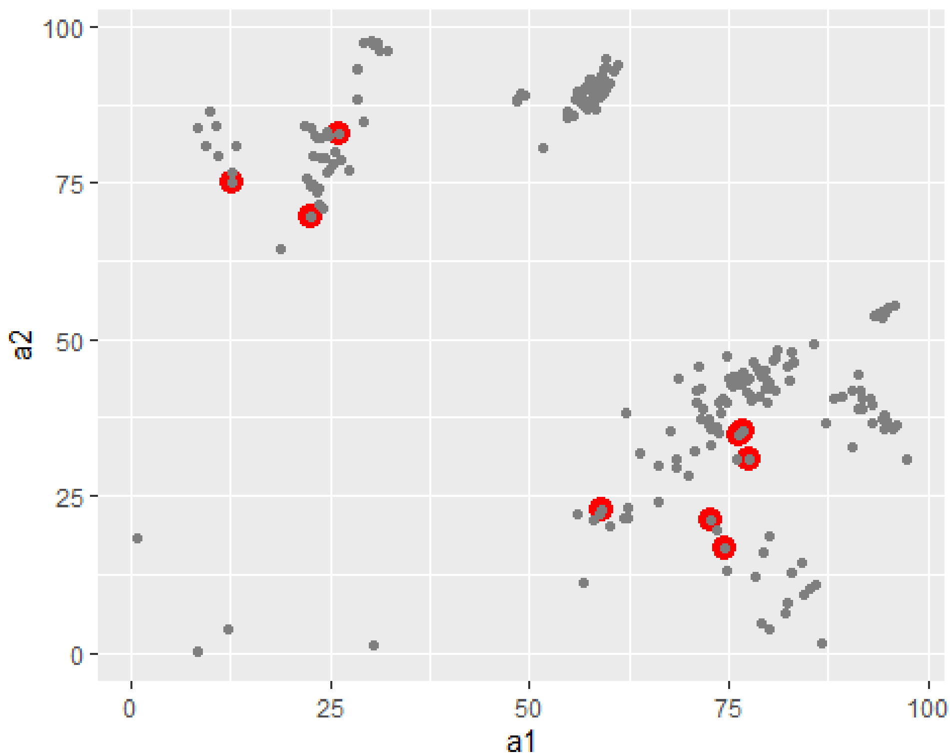
```
## # A tibble: 225 × 3
##       a1    a2  dist
##    <dbl> <dbl> <dbl>
##  1  74.5  16.7   NaN
##  2  72.8  21.2   NaN
##  3  12.7  75.0   NaN
##  4  59    22.8   NaN
##  5  22.6  69.4   NaN
##  6  76.7  35.4   NaN
##  7  76.4  34.8   NaN
##  8  26.0  82.9   NaN
##  9  77.5  31     NaN
## 10  22.6  69.4   NaN
## # … with 215 more rows
```

#> # A tibble: 250 x 3 #> a1 a2 dist #>   #> 1 -15.2 0.0889 30.8 #> 2 30.1 -0.827 13.2 #> 3 16.0 2.27 13.2 #> 4 -10.6 1.38 18.7 #> 5 -19.6 -1.04 41.8 #> 6 7.98 4.59 19.3 #> # . with 244 more rows

```
ggplot(table, aes(x = precision, y = recall)) +
  geom_point(size = 2, colour = "grey30") +
  geom_abline(
    aes(intercept = a1, slope = a2, colour = -dist),
    data = filter(models, rank(dist) <= 10)
  )
```

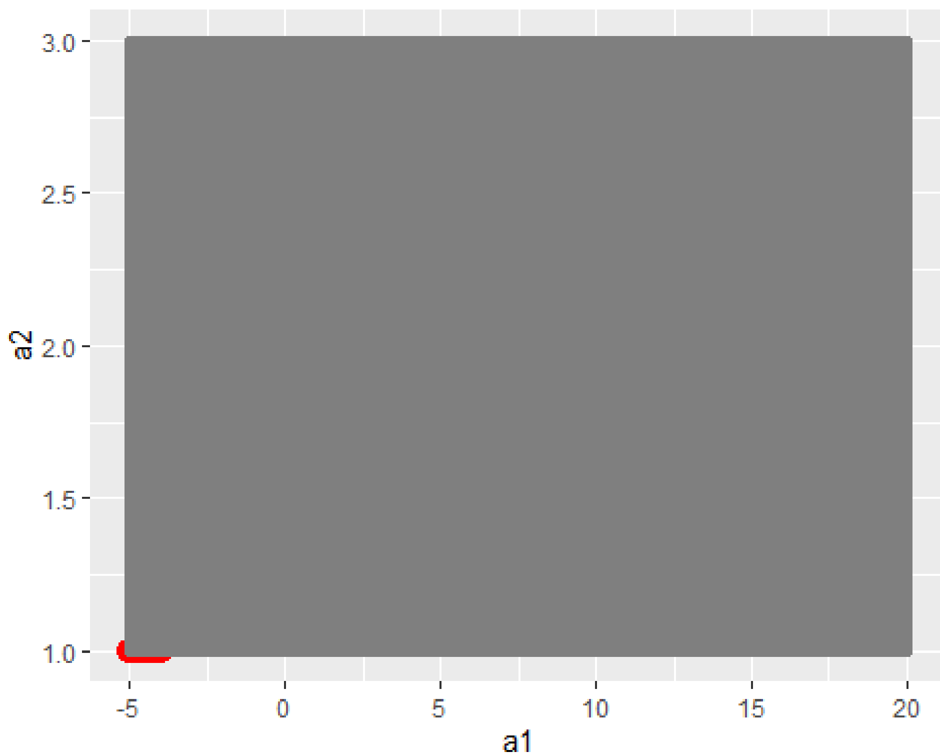



```
ggplot(models, aes(a1, a2)) +
  geom_point(data = filter(models, rank(dist) <= 10), size = 4, colour =
"red") +
  geom_point(aes(colour = -dist))
```

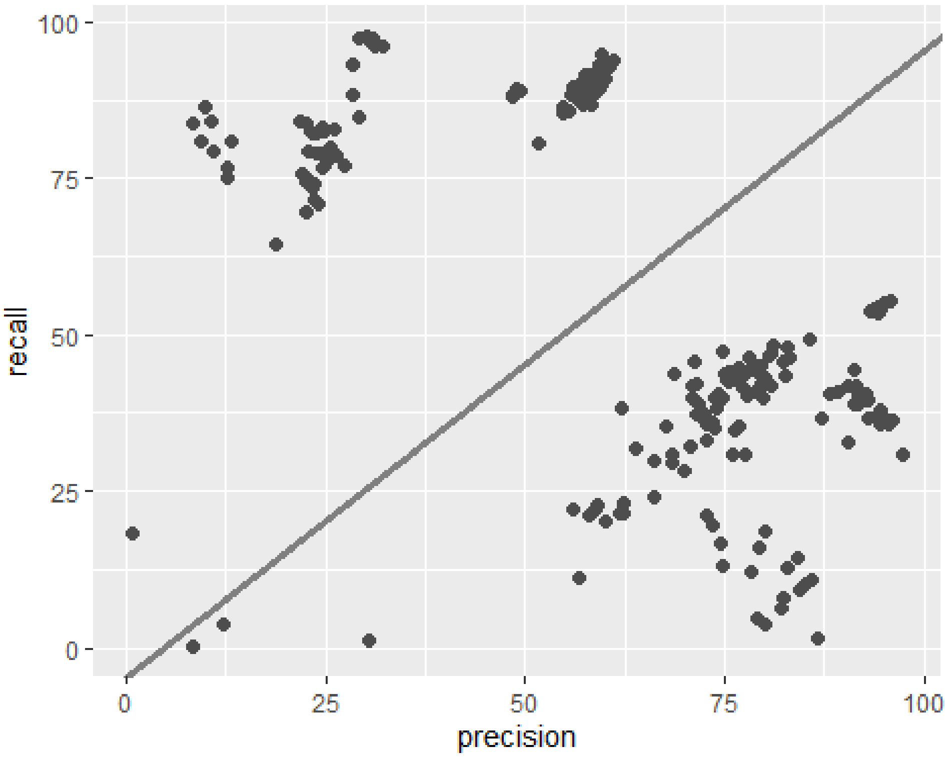



```
grid <- expand.grid(
  a1 = seq(-5, 20, length=25, along.with=precision),
  a2 = seq(1, 3, length = 25, along.with=recall)
) %>%
  mutate(dist = purrr::map2_dbl(a1, a2, table_dist))
```

```
grid %>%
  ggplot(aes(a1, a2)) +
  geom_point(data = filter(grid, rank(dist) <= 10), size = 4, colour = "red")
+
  geom_point(aes(colour = -dist))
```




```
ggplot(table, aes(x=precision, y=recall)) +
  geom_point(size = 2, colour = "grey30") +
  geom_abline(
    aes(intercept = a1, slope = a2, colour = -dist),
    data = filter(grid, rank(dist) <= 10)
  )
```



best <- optim(c(0, 0), measure_distance, data = table) # best$par

ggplot(table, aes(x, y)) +

geom_point(size = 2, colour = “grey30”) +

geom_abline(intercept = best$par[1], slope = best$par[2])

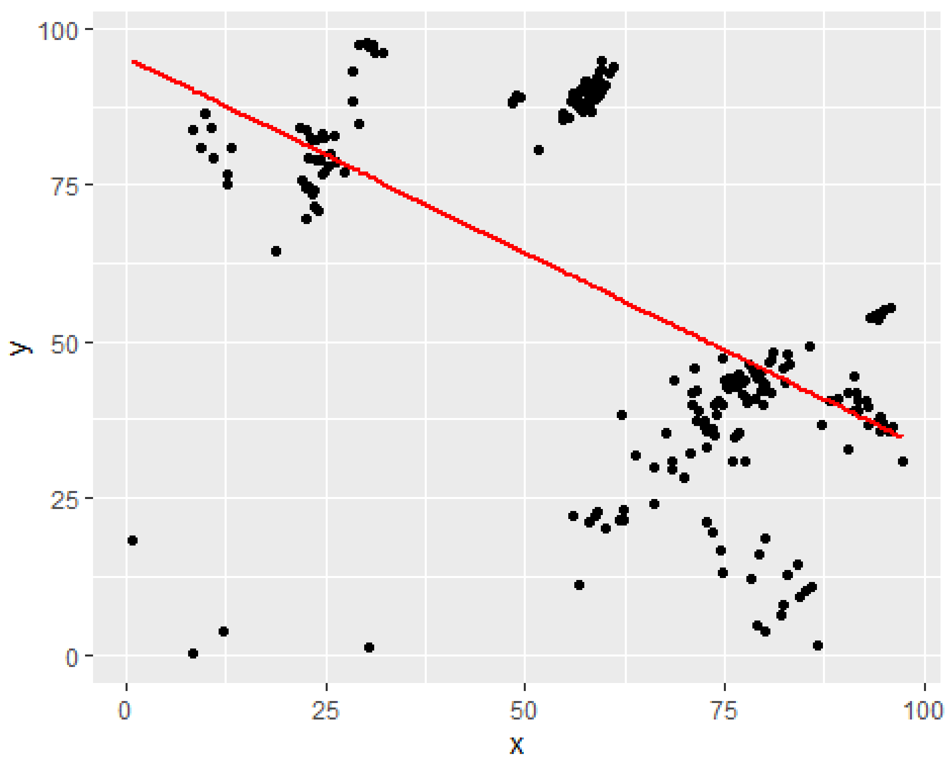
```
table_mod <- lm(y ~ x, data = table)
coef(table_mod)

## (Intercept)           x
##  95.3192201  -0.6232934

tablea <- tibble(
  x = rep(1:10, each = 3),
  y = x * 1.5 + 6 + rt(length(x), df = 2)
)

measure_distance <- function(mod, data) {
  diff <- data$y - model1(mod, data)
  mean(abs(diff))
}
```

```
model1 <- function(a, data) {
  a[1] + data$x * a[2] + a[3]
}

grid <- table %>%
  data_grid(x)
grid

## # A tibble: 211 × 1
##         x
##     <dbl>
##   1  0.74
##   2  8.31
##   3  8.44
##   4  9.32
##   5  9.95
##   6 10.5
##   7 10.8
##   8 12.1
##   9 12.7
##  10 12.8
## # … with 201 more rows

#> # A tibble: 10 x 1
#>        x
#>    <int>
#> 1      1
#> 2      2
#> 3      3
#> 4      4
#> 5      5
#> 6      6
#> # . with 4 more rows

grid <- grid %>%
  add_predictions(table_mod)
grid

## # A tibble: 211 × 2
##         x  pred
##     <dbl> <dbl>
##   1  0.74  94.9
##   2  8.31  90.1
##   3  8.44  90.1
##   4  9.32  89.5
##   5  9.95  89.1
##   6 10.5   88.7
##   7 10.8   88.6
##   8 12.1   87.8
##   9 12.7   87.4
```

```
## 10 12.8   87.4
## # … with 201 more rows
```

```
#> # A tibble: 10 x 2
#>         x  pred
#>     <int> <dbl>
#> 1       1  6.27
#> 2       2  8.32
#> 3       3 10.4
#> 4       4 12.4
#> 5       5 14.5
#> 6       6 16.5
#> # . with 4 more rows
```

```
ggplot(table, aes(x)) +
  geom_point(aes(y = y)) +
  geom_line(aes(y = pred), data = grid, colour = "red", size = 1)
```




```
table <- table %>%
  add_residuals(table_mod)
table
```

```
##       Year       Sampling       Technique     Classifier Precision Recall
## 1     2010     Imbalanced             N/A    Naive Bayes     74.49  16.70
## 2     2010     Imbalanced             N/A   Logistic Reg     72.82  21.18
## 3     2010     Imbalanced             N/A        XGBoost     12.66  75.05
## 4     2010     Imbalanced             N/A   DecisionTree     59.00  22.77
```

```
## 5   2010    Imbalanced            N/A Random Forest     22.65  69.45
## 6   2010 Undersampling       NearMiss   Naive Bayes     76.67  35.38
## 7   2010 Undersampling       NearMiss  Logistic Reg     76.37  34.83
## 8   2010 Undersampling       NearMiss       XGBoost     26.04  82.92
## 9   2010 Undersampling       NearMiss  DecisionTree     77.47  31.00
## 10  2010 Undersampling       NearMiss Random Forest     22.65  69.45
## 11  2010  Oversampling          SMOTE   Naive Bayes     57.98  88.41
## 12  2010  Oversampling          SMOTE  Logistic Reg     58.15  88.40
## 13  2010  Oversampling          SMOTE       XGBoost     94.16  54.34
## 14  2010  Oversampling          SMOTE  DecisionTree     56.43  87.71
## 15  2010  Oversampling          SMOTE Random Forest     58.17  90.11
## 16  2010  Oversampling            ROS   Naive Bayes     57.95  89.05
## 17  2010  Oversampling            ROS  Logistic Reg     57.91  88.23
## 18  2010  Oversampling            ROS       XGBoost     94.52  54.08
## 19  2010  Oversampling            ROS  DecisionTree     56.46  88.55
## 20  2010  Oversampling            ROS Random Forest     57.68  89.70
## 21  2010 Undersampling            RUS   Naive Bayes     67.54  35.25
## 22  2010 Undersampling            RUS  Logistic Reg     57.29  86.59
## 23  2010 Undersampling            RUS       XGBoost     93.31  53.88
## 24  2010 Undersampling            RUS  DecisionTree     54.84  85.36
## 25  2010 Unsersampling            RUS Random Forest     58.32  86.82
## 26  2010 Undersampling      Tomelinks   Naive Bayes     58.00  88.92
## 27  2010 Undersampling      Tomelinks  Logistic Reg     73.39  35.93
## 28  2010 Undersampling      Tomelinks       XGBoost     94.27  54.13
## 29  2010 Undersampling      Tomelinks  DecisionTree     56.52  88.80
## 30  2010 Undersampling      Tomelinks Random Forest     57.96  89.30
## 31  2012    Imbalanced            N/A   Naive Bayes     82.95  12.68
## 32  2012    Imbalanced            N/A  Logistic Reg     80.00  18.73
## 33  2012    Imbalanced            N/A       XGBoost     13.06  80.78
## 34  2012    Imbalanced            N/A  DecisionTree     62.30  23.00
## 35  2012 Undersampling       NearMiss   Naive Bayes     91.79  40.57
## 36  2012 Undersampling       NearMiss  Logistic Reg     91.84  39.06
## 37  2012 Undersampling       NearMiss       XGBoost     30.11  97.70
## 38  2012 Undersampling       NearMiss  DecisionTree     95.96  36.43
## 39  2012  Oversampling          SMOTE   Naive Bayes     58.71  89.86
## 40  2012  Oversampling          SMOTE  Logistic Reg     58.76  88.66
## 41  2012  Oversampling          SMOTE       XGBoost     21.81  84.12
## 42  2012  Oversampling          SMOTE  DecisionTree     57.47  88.06
## 43  2012  Oversampling            ROS   Naive Bayes     76.52  42.70
## 44  2012  Oversampling            ROS  Logistic Reg     77.85  40.72
## 45  2012  Oversampling            ROS       XGBoost     22.85  79.33
## 46  2012  Oversampling            ROS  DecisionTree     79.77  39.89
## 47  2012 Undersampling            RUS   Naive Bayes     72.59  37.17
## 48  2012 Undersampling            RUS  Logistic Reg     73.62  34.90
## 49  2012 Undersampling            RUS       XGBoost     22.10  75.75
## 50  2012 Undersampling            RUS  DecisionTree     69.89  28.39
## 51  2012 Undersampling      Tomelinks   Naive Bayes     59.57  89.88
## 52  2012 Undersampling      Tomelinks  Logistic Reg     77.36  41.44
## 53  2012 Undersampling      Tomelinks       XGBoost     23.72  79.00
## 54  2012 Undersampling      Tomelinks  DecisionTree     78.82  40.96
```

```
## 55  2013    Imbalanced             N/A  Naive Bayes     74.70  13.27
## 56  2013    Imbalanced             N/A Logistic Reg     73.53  19.52
## 57  2013    Imbalanced             N/A      XGBoost     12.76  76.85
## 58  2013    Imbalanced             N/A DecisionTree     56.04  22.23
## 59  2013 Undersampling        NearMiss  Naive Bayes     90.40  41.79
## 60  2013 Undersampling        NearMiss Logistic Reg     89.11  40.89
## 61  2013 Undersampling        NearMiss      XGBoost     30.99  97.43
## 62  2013 Undersampling        NearMiss DecisionTree     94.91  37.03
## 63  2013  Oversampling           SMOTE  Naive Bayes     58.71  89.86
## 64  2013  Oversampling           SMOTE Logistic Reg     58.76  88.66
## 65  2013  Oversampling           SMOTE      XGBoost     21.81  84.13
## 66  2013  Oversampling           SMOTE DecisionTree     57.47  88.06
## 67  2013  Oversampling             ROS  Naive Bayes     75.12  43.64
## 68  2013  Oversampling             ROS Logistic Reg     75.63  42.43
## 69  2013  Oversampling             ROS      XGBoost     24.54  76.83
## 70  2013  Oversampling             ROS DecisionTree     74.68  40.05
## 71  2013 Undersampling             RUS  Naive Bayes     71.84  39.03
## 72  2013 Undersampling             RUS Logistic Reg     71.37  37.24
## 73  2013 Undersampling             RUS      XGBoost     23.37  73.38
## 74  2013 Undersampling             RUS DecisionTree     66.15  30.01
## 75  2013 Undersampling       Tomelinks  Naive Bayes     59.20  89.18
## 76  2013 Undersampling       Tomelinks Logistic Reg     75.19  42.75
## 77  2013 Undersampling       Tomelinks      XGBoost     24.91  76.92
## 78  2013 Undersampling       Tomelinks DecisionTree     74.29  40.53
## 79  2014    Imbalanced             N/A  Naive Bayes     84.51   9.13
## 80  2014    Imbalanced             N/A Logistic Reg     84.04  14.30
## 81  2014    Imbalanced             N/A      XGBoost     10.54  83.97
## 82  2014    Imbalanced             N/A DecisionTree     62.37  21.66
## 83  2014 Undersampling        NearMiss  Naive Bayes     49.33  88.80
## 84  2014 Undersampling        NearMiss Logistic Reg     84.04  14.30
## 85  2014 Undersampling        NearMiss      XGBoost     29.21  97.28
## 86  2014 Undersampling        NearMiss DecisionTree     95.63  35.76
## 87  2014  Oversampling           SMOTE  Naive Bayes     59.31  91.58
## 88  2014  Oversampling           SMOTE Logistic Reg     59.65  91.14
## 89  2014  Oversampling           SMOTE      XGBoost     95.00  54.98
## 90  2014  Oversampling           SMOTE DecisionTree     58.76  91.31
## 91  2014  Oversampling             ROS  Naive Bayes     75.44  44.17
## 92  2014  Oversampling             ROS Logistic Reg     80.77  41.96
## 93  2014  Oversampling             ROS      XGBoost     23.00  82.52
## 94  2014  Oversampling             ROS DecisionTree     82.50  43.45
## 95  2014 Undersampling             RUS  Naive Bayes     57.43  90.28
## 96  2014 Undersampling             RUS Logistic Reg     72.78  35.83
## 97  2014 Undersampling             RUS      XGBoost     94.27  53.47
## 98  2014 Undersampling             RUS DecisionTree     54.68  86.53
## 99  2014 Undersampling       Tomelinks  Naive Bayes     75.27  43.66
## 100 2014 Undersampling       Tomelinks Logistic Reg     80.05  42.08
## 101 2014 Undersampling       Tomelinks      XGBoost     23.23  82.18
## 102 2014 Undersampling       Tomelinks DecisionTree     59.33  93.08
## 103 2014            FS Standard Scalar DecisionTree     61.90  21.62
## 104 2014            FS      Extra Tree  Naive Bayes     66.09  24.14
```

```
## 105 2014             FS         SS & SKB  Logistic Reg     86.79   1.42
## 106 2015     Imbalanced              N/A   Naive Bayes     82.11   6.44
## 107 2015     Imbalanced              N/A  Logistic Reg     78.39  12.22
## 108 2015     Imbalanced              N/A       XGBoost      9.32  80.94
## 109 2015     Imbalanced              N/A  DecisionTree     58.11  21.06
## 110 2015 Undersampling         NearMiss   Naive Bayes     48.34  88.34
## 111 2015 Undersampling         NearMiss  Logistic Reg     88.13  40.58
## 112 2015 Undersampling         NearMiss       XGBoost     31.02  96.18
## 113 2015 Undersampling         NearMiss  DecisionTree     93.00  36.82
## 114 2015  Oversampling            SMOTE   Naive Bayes     59.16  90.34
## 115 2015  Oversampling            SMOTE  Logistic Reg     59.19  90.36
## 116 2015  Oversampling            SMOTE       XGBoost     24.50  83.16
## 117 2015  Oversampling            SMOTE  DecisionTree     79.51  42.27
## 118 2015  Oversampling              ROS   Naive Bayes     74.87  47.24
## 119 2015  Oversampling              ROS  Logistic Reg     79.57  45.13
## 120 2015  Oversampling              ROS       XGBoost     25.46  79.86
## 121 2015  Oversampling              ROS  DecisionTree     80.83  46.89
## 122 2015 Undersampling              RUS   Naive Bayes     73.66  39.95
## 123 2015 Undersampling              RUS  Logistic Reg     73.87  38.19
## 124 2015 Undersampling              RUS       XGBoost     23.48  74.19
## 125 2015 Undersampling              RUS  DecisionTree     70.75  32.26
## 126 2015 Undersampling        Tomelinks   Naive Bayes     78.06  46.37
## 127 2015 Undersampling        Tomelinks  Logistic Reg     78.63  45.33
## 128 2015 Undersampling        Tomelinks       XGBoost     26.37  78.69
## 129 2015 Undersampling        Tomelinks  DecisionTree     80.56  46.69
## 130 2016     Imbalanced              N/A   Naive Bayes      8.44   0.25
## 131 2016     Imbalanced              N/A  Logistic Reg     30.32   1.33
## 132 2016     Imbalanced              N/A       XGBoost      0.74  18.29
## 133 2016     Imbalanced              N/A  DecisionTree     12.13   3.89
## 134 2016 Undersampling         NearMiss   Naive Bayes     48.45  87.85
## 135 2016 Undersampling         NearMiss  Logistic Reg     87.21  36.76
## 136 2016 Undersampling         NearMiss       XGBoost     28.26  93.04
## 137 2016 Undersampling         NearMiss  DecisionTree     90.49  32.91
## 138 2016  Oversampling            SMOTE   Naive Bayes     56.65  87.29
## 139 2016  Oversampling            SMOTE  Logistic Reg     57.18  87.52
## 140 2016  Oversampling            SMOTE       XGBoost     93.65  54.06
## 141 2016  Oversampling            SMOTE  DecisionTree     55.60  85.72
## 142 2016  Oversampling              ROS   Naive Bayes     57.52  91.43
## 143 2016  Oversampling              ROS  Logistic Reg     70.99  41.93
## 144 2016  Oversampling              ROS       XGBoost     24.10  70.74
## 145 2016  Oversampling              ROS  DecisionTree     55.83  88.20
## 146 2016 Undersampling              RUS   Naive Bayes     62.03  38.16
## 147 2016 Undersampling              RUS  Logistic Reg     63.96  31.90
## 148 2016 Undersampling              RUS       XGBoost     18.65  64.27
## 149 2016 Undersampling              RUS  DecisionTree     51.68  80.64
## 150 2016 Undersampling        Tomelinks   Naive Bayes     68.65  43.90
## 151 2016 Undersampling        Tomelinks  Logistic Reg     71.44  42.05
## 152 2016 Undersampling        Tomelinks       XGBoost     23.60  71.46
## 153 2016 Undersampling        Tomelinks  DecisionTree     55.92  89.48
## 154 2017     Imbalanced              N/A   Naive Bayes     82.28   7.92
```

```
## 155 2017    Imbalanced            N/A  Logistic Reg     79.39  15.98
## 156 2017    Imbalanced            N/A       XGBoost     10.83  79.20
## 157 2017    Imbalanced            N/A  DecisionTree     58.87  22.11
## 158 2017 Undersampling       NearMiss   Naive Bayes     48.98  89.36
## 159 2017 Undersampling       NearMiss  Logistic Reg     93.06  39.56
## 160 2017 Undersampling       NearMiss       XGBoost     97.36  30.88
## 161 2017 Undersampling       NearMiss  DecisionTree     94.23  37.48
## 162 2017  Oversampling          SMOTE   Naive Bayes     57.74  91.47
## 163 2017  Oversampling          SMOTE  Logistic Reg     57.82  90.42
## 164 2017  Oversampling          SMOTE       XGBoost     22.55  83.77
## 165 2017  Oversampling          SMOTE  DecisionTree     77.72  40.32
## 166 2017  Oversampling            ROS   Naive Bayes     76.68  44.67
## 167 2017  Oversampling            ROS  Logistic Reg     77.66  43.73
## 168 2017  Oversampling            ROS       XGBoost     25.33  78.13
## 169 2017  Oversampling            ROS  DecisionTree     78.78  44.65
## 170 2017 Undersampling            RUS   Naive Bayes     70.85  39.82
## 171 2017 Undersampling            RUS  Logistic Reg     72.51  36.51
## 172 2017 Undersampling            RUS       XGBoost     22.63  74.59
## 173 2017 Undersampling            RUS  DecisionTree     68.33  30.97
## 174 2017 Undersampling      Tomelinks   Naive Bayes     58.69  90.82
## 175 2017 Undersampling      Tomelinks  Logistic Reg     77.32  43.37
## 176 2017 Undersampling      Tomelinks       XGBoost     24.40  78.95
## 177 2017 Undersampling      Tomelinks  DecisionTree     78.95  44.18
## 178 2018    Imbalanced            N/A   Naive Bayes     79.06   4.85
## 179 2018    Imbalanced            N/A  Logistic Reg     91.19  39.00
## 180 2018    Imbalanced            N/A       XGBoost      8.31  83.80
## 181 2018    Imbalanced            N/A  DecisionTree     56.80  11.33
## 182 2018 Undersampling       NearMiss   Naive Bayes     91.44  41.73
## 183 2018 Undersampling       NearMiss  Logistic Reg     85.23  10.28
## 184 2018 Undersampling       NearMiss       XGBoost     30.41  97.00
## 185 2018 Undersampling       NearMiss  DecisionTree     94.46  35.84
## 186 2018  Oversampling          SMOTE   Naive Bayes     59.13  91.79
## 187 2018  Oversampling          SMOTE  Logistic Reg     59.94  90.74
## 188 2018  Oversampling          SMOTE       XGBoost     95.68  55.37
## 189 2018  Oversampling          SMOTE  DecisionTree     59.07  92.05
## 190 2018  Oversampling            ROS   Naive Bayes     59.51  93.60
## 191 2018  Oversampling            ROS  Logistic Reg     80.01  43.26
## 192 2018  Oversampling            ROS       XGBoost     24.91  82.35
## 193 2018  Oversampling            ROS  DecisionTree     59.57  94.69
## 194 2018 Undersampling            RUS   Naive Bayes     57.13  90.33
## 195 2018 Undersampling            RUS  Logistic Reg     72.71  35.90
## 196 2018 Undersampling            RUS       XGBoost     22.68  74.60
## 197 2018 Undersampling            RUS  DecisionTree     68.43  29.72
## 198 2018 Undersampling      Tomelinks   Naive Bayes     76.05  44.14
## 199 2018 Undersampling      Tomelinks  Logistic Reg     79.88  43.31
## 200 2018 Undersampling      Tomelinks       XGBoost     23.90  82.25
## 201 2018 Undersampling      Tomelinks  DecisionTree     82.42  45.59
## 202 2019    Imbalanced            N/A   Naive Bayes     80.14   3.79
## 203 2019    Imbalanced            N/A  Logistic Reg     85.94  10.82
## 204 2019    Imbalanced            N/A       XGBoost      9.95  86.31
```

```
## 205 2019    Imbalanced             N/A  DecisionTree     59.93  20.26
## 206 2019 Undersampling        NearMiss   Naive Bayes     91.34  44.56
## 207 2019 Undersampling        NearMiss  Logistic Reg     92.64  40.48
## 208 2019 Undersampling        NearMiss       XGBoost     32.26  96.06
## 209 2019 Undersampling        NearMiss  DecisionTree     94.48  37.98
## 210 2019  Oversampling           SMOTE   Naive Bayes     60.53  92.82
## 211 2019  Oversampling           SMOTE  Logistic Reg     60.96  93.67
## 212 2019  Oversampling           SMOTE       XGBoost     28.27  88.45
## 213 2019  Oversampling           SMOTE  DecisionTree     83.06  46.47
## 214 2019  Oversampling             ROS   Naive Bayes     60.53  92.82
## 215 2019  Oversampling             ROS  Logistic Reg     60.96  93.67
## 216 2019  Oversampling             ROS       XGBoost     28.27  88.45
## 217 2019  Oversampling             ROS  DecisionTree     83.06  46.47
## 218 2019 Undersampling             RUS   Naive Bayes     71.22  45.77
## 219 2019 Undersampling             RUS  Logistic Reg     75.98  30.80
## 220 2019 Undersampling             RUS       XGBoost     27.22  76.95
## 221 2019 Undersampling             RUS  DecisionTree     72.62  32.97
## 222 2019 Undersampling       Tomelinks   Naive Bayes     81.20  48.30
## 223 2019 Undersampling       Tomelinks  Logistic Reg     82.89  48.10
## 224 2019 Undersampling       Tomelinks       XGBoost     29.04  84.70
## 225 2019 Undersampling       Tomelinks  DecisionTree     85.60  49.16
##             resid
## 1   -32.19009215
## 2   -28.75099218
## 3   -12.37832520
## 4   -35.77490745
## 5   -11.75162379
## 6   -12.15131246
## 7   -12.88830049
## 8     3.83134096
## 9   -16.03267771
## 10  -11.75162379
## 11   29.22933325
## 12   29.32529313
## 13   17.71008971
## 14   27.56322843
## 15   31.04775900
## 16   29.85063445
## 17   29.00570271
## 18   17.67447534
## 19   28.42192723
## 20   30.33234522
## 21  -17.97198152
## 22   26.97926078
## 23   16.72029029
## 24   24.22219187
## 25   27.85125302
## 26   29.75179912
## 27  -13.64571493
## 28   17.56865198
```

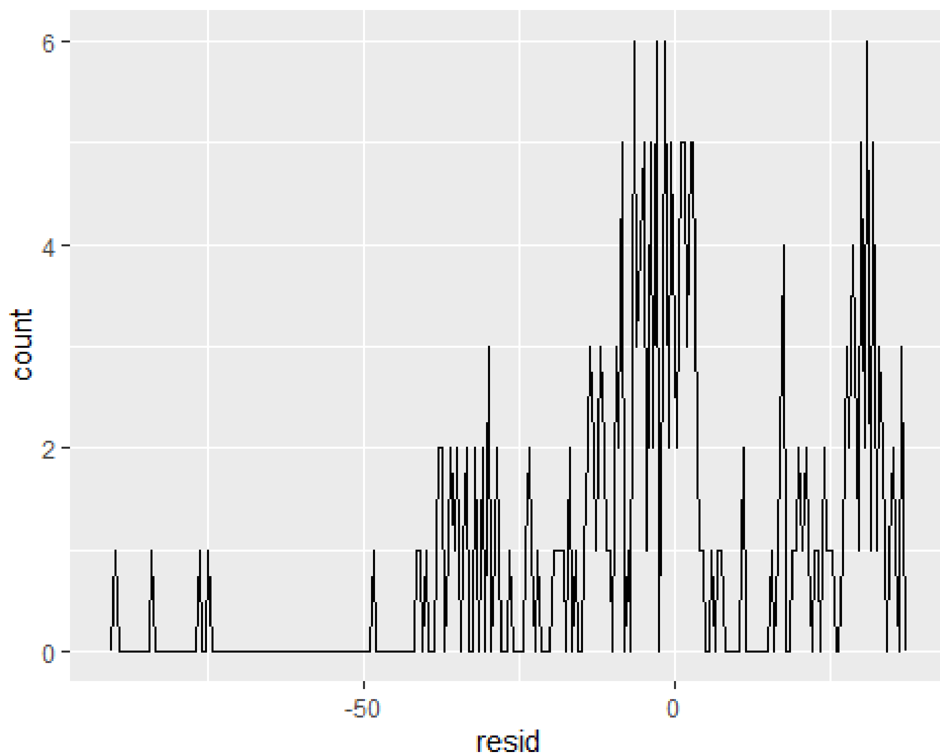
```
## 29   28.70932484
## 30   30.10686738
## 31  -30.93702969
## 32  -26.72574533
## 33   -6.39900782
## 34  -33.48803911
## 35    2.46288427
## 36    0.98404894
## 37   21.14814523
## 38    0.92201789
## 39   31.13433746
## 40   29.96550213
## 41    2.39480973
## 42   28.56145360
## 43   -4.92480648
## 44   -6.07582621
## 45   -1.74696510
## 46   -5.70910282
## 47  -12.90434967
## 48  -14.53235744
## 49   -5.79443518
## 50  -23.36724195
## 51   31.69036981
## 52   -5.66123999
## 53   -1.53469981
## 54   -5.23123158
## 55  -35.48920053
## 56  -29.96845385
## 57  -10.51599585
## 58  -38.15985601
## 59    2.81650639
## 60    1.11245786
## 61   21.42664346
## 62    0.86755978
## 63   31.13433746
## 64   29.96550213
## 65    2.40480973
## 66   28.56145360
## 67   -4.85741728
## 68   -5.74953763
## 69   -3.19359920
## 70   -8.72166640
## 71  -11.51181975
## 72  -13.59476766
## 73   -7.37285251
## 74  -24.07835939
## 75   30.75975124
## 76   -5.70378674
## 77   -2.87298062
## 78   -8.48475084
```

```
## 79  -33.51469194
## 80  -28.63763985
## 81   -4.77970728
## 82  -34.78440857
## 83   24.22784504
## 84  -28.63763985
## 85   20.16718114
## 86    0.04633105
## 87   33.22831352
## 88   33.00023329
## 89   18.87365619
## 90   32.61550213
## 91   -4.12796339
## 92   -3.01580938
## 93    1.53652892
## 94   -0.44751174
## 95   30.75652186
## 96  -14.12592392
## 97   16.90865198
## 98   25.29246492
## 99   -4.74392327
## 100  -3.34458065
## 101   1.33988641
## 102  34.74077939
## 103 -35.11735649
## 104 -29.98575700
## 105 -39.80358291
## 106 -37.70059618
## 107 -34.23924775
## 108  -8.57012527
## 109 -38.03963860
## 110  23.15078454
## 111   0.19163030
## 112  20.19534226
## 113  -0.53293068
## 114  31.89481950
## 115  31.93351831
## 116   3.11146907
## 117  -3.49115911
## 118  -1.41324064
## 119  -0.59376150
## 120   0.40983076
## 121   1.95158823
## 122  -9.45742570
## 123 -11.08653408
## 124  -6.49429024
## 125 -18.96120959
## 126  -0.29493459
## 127  -0.97965733
## 128  -0.19297221
```

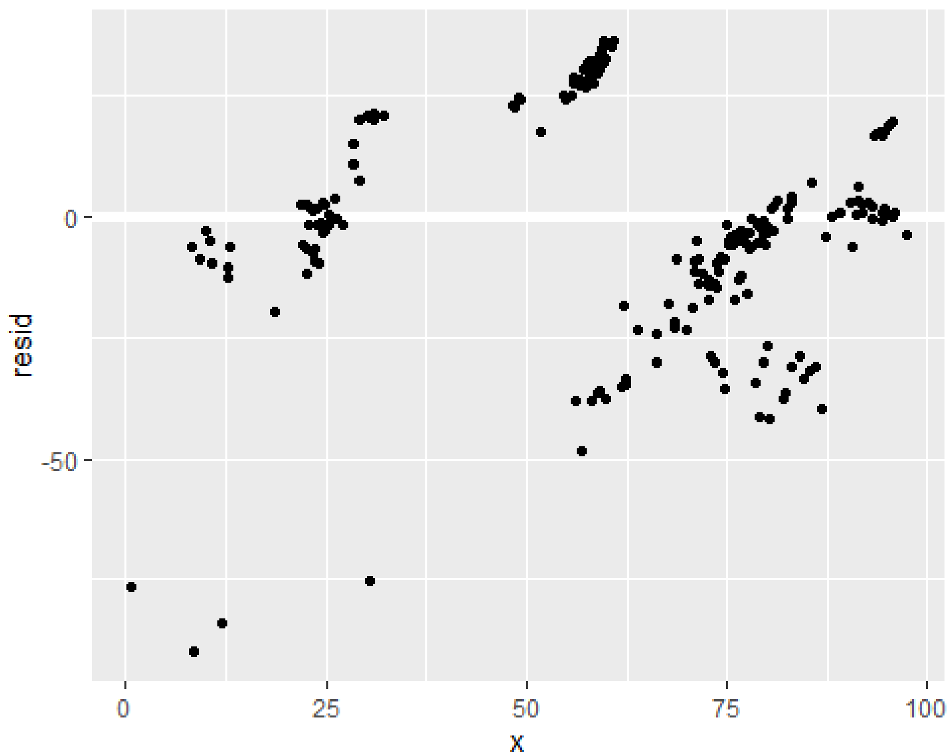
```
## 129   1.58329900
## 130 -89.80862349
## 131 -75.09096314
## 132 -76.56798293
## 133 -83.86867072
## 134  22.72934682
## 135  -4.20179966
## 136  15.33505238
## 137  -6.00739720
## 138  27.28035298
## 139  27.84069850
## 140  17.11221005
## 141  25.05589488
## 142  31.96261827
## 143  -9.14161917
## 144  -9.55784831
## 145  27.67925237
## 146 -18.49632834
## 147 -23.55337201
## 148 -19.42479752
## 149  17.53258461
## 150  -8.63012581
## 151  -8.74113712
## 152  -9.14949502
## 153  29.01534878
## 154 -36.11463629
## 155 -29.85595432
## 156  -9.36895218
## 157 -36.51593559
## 158  24.56969234
## 159   2.24446693
## 160  -3.75537130
## 161   0.89372025
## 162  32.13974283
## 163  31.13960630
## 164   2.50604687
## 165  -6.55685436
## 166  -2.85507953
## 167  -3.18425196
## 168  -1.40119738
## 169  -1.56616332
## 170 -11.33888025
## 171 -13.61421315
## 172  -6.62408966
## 173 -21.75957970
## 174  32.08187159
## 175  -3.75617173
## 176  -1.16086028
## 177  -1.93020343
## 178 -41.19164115
```

```
## 179   0.51890821
## 180  -6.33965164
## 181 -48.58615300
## 182   3.40473156
## 183 -31.91592066
## 184  20.63513326
## 185  -0.60292226
## 186  33.32612070
## 187  32.78098838
## 188  19.68749573
## 189  33.54872309
## 190  35.37297220
## 191  -2.18951239
## 192   2.55701938
## 193  36.50036981
## 194  30.61953383
## 195 -14.09955446
## 196  -6.58292498
## 197 -22.94725036
## 198  -3.77775439
## 199  -2.22054054
## 200   1.82749301
## 201   1.64262479
## 202 -41.57848424
## 203 -30.93338232
## 204  -2.80745040
## 205 -37.70524455
## 206   6.17240222
## 207   2.90268369
## 208  20.84822612
## 209   1.54954360
## 210  35.22873151
## 211  36.34674768
## 212  10.75128531
## 213   2.92153258
## 214  35.22873151
## 215  36.34674768
## 216  10.75128531
## 217   2.92153258
## 218  -5.15826168
## 219 -17.16138493
## 220  -1.40317279
## 221 -17.08565087
## 222   3.59220680
## 223   4.44557270
## 224   7.48122126
## 225   7.19469791

#> # A tibble: 30 x 3
#>       x     y  resid
```

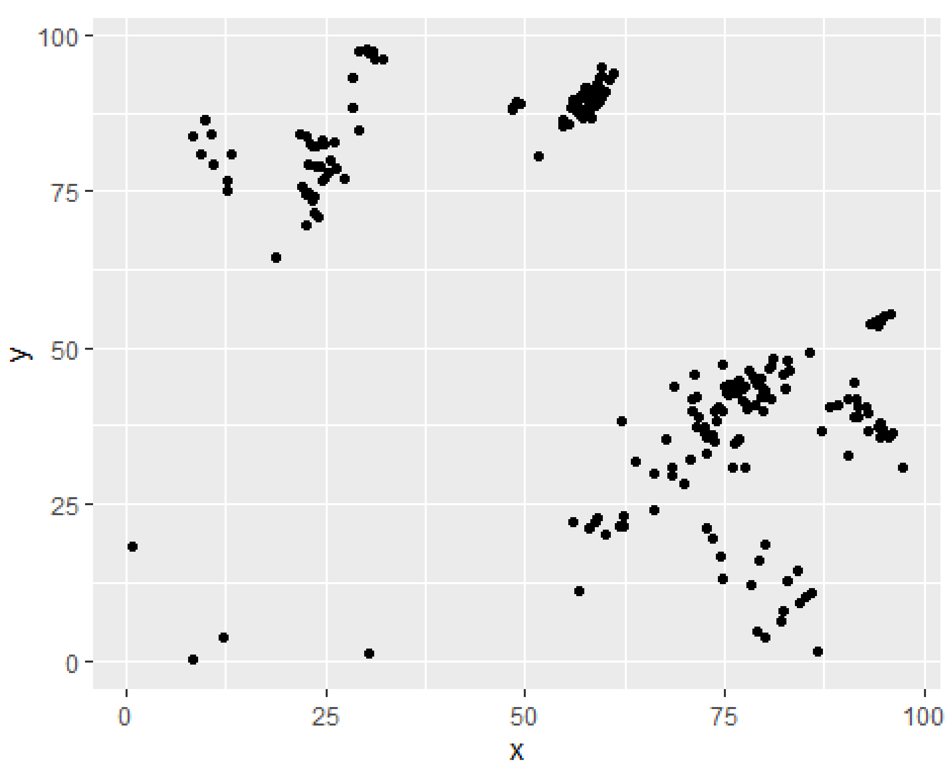
```
#>   <int> <dbl>  <dbl>
#> 1     1  4.20 -2.07
#> 2     1  7.51  1.24
#> 3     1  2.13 -4.15
#> 4     2  8.99  0.665
#> 5     2 10.2   1.92
#> 6     2 11.3   2.97
#> # . with 24 more rows

ggplot(table, aes(resid)) +
  geom_freqpoly(binwidth = 0.5)
```




```
ggplot(table, aes(x, resid)) +
  geom_ref_line(h = 0) +
  geom_point()
```



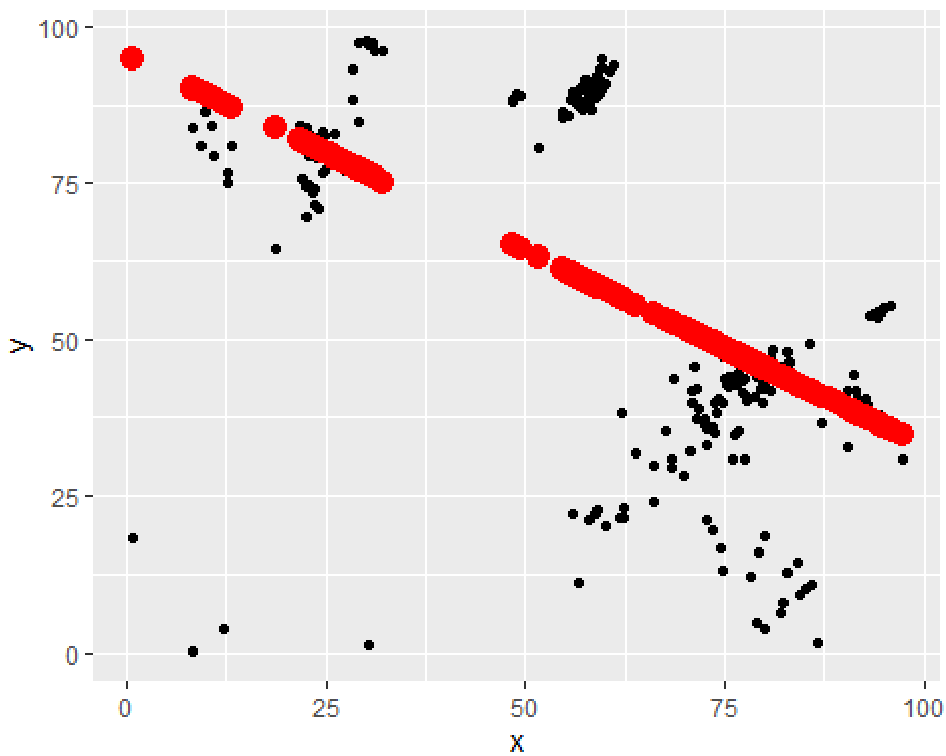

```
ggplot(table) +
  geom_point(aes(x, y))
```



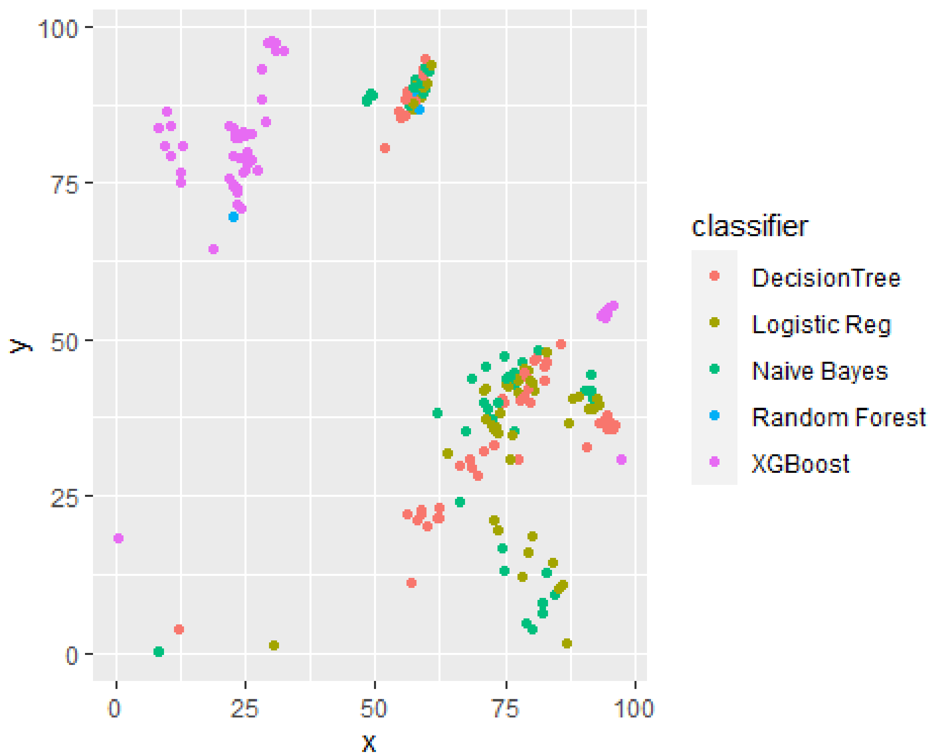
```r
mod2 <- lm(y ~ x, data = table)

grid <- table %>%
  data_grid(x) %>%
  add_predictions(mod2)
grid
```

```
## # A tibble: 211 × 2
##        x  pred
##    <dbl> <dbl>
##  1  0.74  94.9
##  2  8.31  90.1
##  3  8.44  90.1
##  4  9.32  89.5
##  5  9.95  89.1
##  6 10.5   88.7
##  7 10.8   88.6
##  8 12.1   87.8
##  9 12.7   87.4
## 10 12.8   87.4
## # … with 201 more rows
```

```r
#> # A tibble: 4 x 2
#>   x      pred
#>   <chr> <dbl>
#> 1 a      1.15
#> 2 b      8.12
#> 3 c      6.13
#> 4 d      1.91

ggplot(table, aes(x)) +
  geom_point(aes(y = y)) +
  geom_point(data = grid, aes(y = pred), colour = "red", size = 4)
```




```
# tibble(x = "e") %>%
#   add_predictions(mod2)

ggplot(table, aes(x, y)) +
  geom_point(aes(colour = classifier))
```

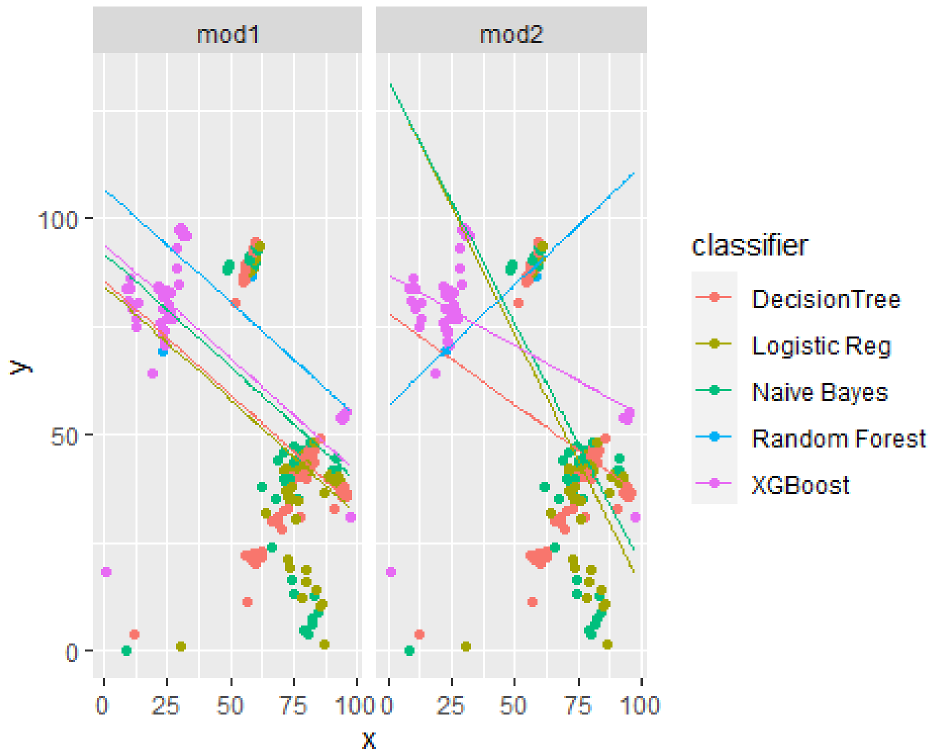



```
#> Error in model.frame.default(Terms, newdata, na.action = na.action, xlev =
object$xlevels): factor x has new level e
```

```
mod1 <- lm(y ~ x + classifier, data = table)
mod2 <- lm(y ~ x * classifier, data = table)

grid <- table %>%
  data_grid(x, classifier) %>%
  gather_predictions(mod1, mod2)
grid

## # A tibble: 2,110 × 4
##    model     x classifier     pred
##    <chr> <dbl> <chr>         <dbl>
##  1 mod1   0.74 DecisionTree   85.1
##  2 mod1   0.74 Logistic Reg   83.9
##  3 mod1   0.74 Naive Bayes    91.5
##  4 mod1   0.74 Random Forest 106.
##  5 mod1   0.74 XGBoost        93.6
##  6 mod1   8.31 DecisionTree   81.1
##  7 mod1   8.31 Logistic Reg   79.9
##  8 mod1   8.31 Naive Bayes    87.5
##  9 mod1   8.31 Random Forest 102.
## 10 mod1   8.31 XGBoost        89.6
## # … with 2,100 more rows
```

Problematic Code

```
foo = tibble(x=precision, y=recall, classifier=classifier, sampling=sampling,
technique=technique, year=year)
ggplot(foo, aes(x, y, colour = classifier)) +
  geom_point() +
  geom_line(data = grid, aes(y = pred)) +
  facet_wrap(~ model)
```

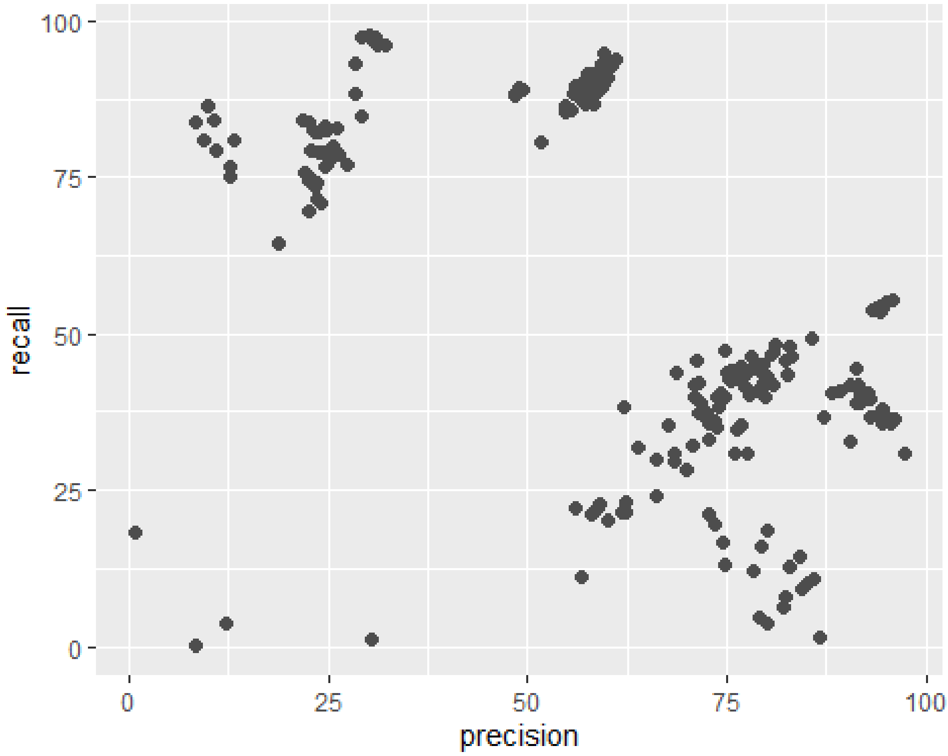


Leftover Code

#> # A tibble: 80 x 4 #> model x1 x2 pred #>   #> 1 mod1 1 a 1.67 #> 2 mod1 1 b 4.56 #> 3 mod1 1 c 6.48 #> 4 mod1 1 d 4.03 #> 5 mod1 2 a 1.48 #> 6 mod1 2 b 4.37 #> # . with 74 more rows

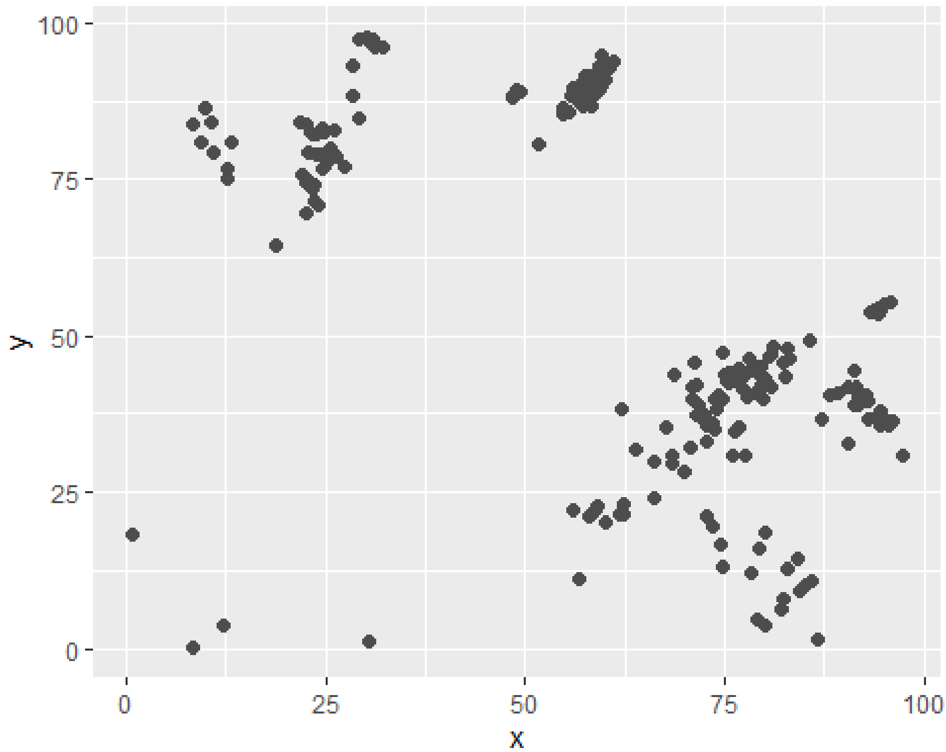
```
# Leftover Code
#> # A tibble: 80 x 4
#>   model    x1 x2     pred
#>   <chr> <int> <fct> <dbl>
#> 1 mod1      1 a      1.67
#> 2 mod1      1 b      4.56
#> 3 mod1      1 c      6.48
#> 4 mod1      1 d      4.03
#> 5 mod1      2 a      1.48
#> 6 mod1      2 b      4.37
#> # . with 74 more rows
```

```r
ggplot(table, aes(x=precision, y=recall)) +
  geom_point(size = 2, colour = "grey30")
```




```r
#   ggplot(foo, aes(x, y)) + geom_boxplot()
# ggplot(classifier, sampling, aes(x, y)) + geom_boxplot()
#

# best <- optim(c(0, 0), measure_distance, data = table)
# best$par
# #> [1] 4.222248 2.051204
#
# data_dist <- function(x, y) {
#   measure_distance(c(x, y), table)
# }
#
 # models <- foo %>%
 #   mutate(dist = purrr::foo(x, y, data_dist))
 # models
#  ggplot(table, aes(x=precision, y=recall)) +
#     geom_point(size = 2, colour = "grey30") +
#     geom_abline(intercept = best$par[1], slope = best$par[2])
#  ggplot(classifier, aes(color, price)) + geom_boxplot()
#  ggplot(classifier, diamonds, aes(clarity, price)) + geom_boxplot()
#

ggplot(foo, aes(x, y)) +
  geom_point(size = 2, colour = "grey30") +
```

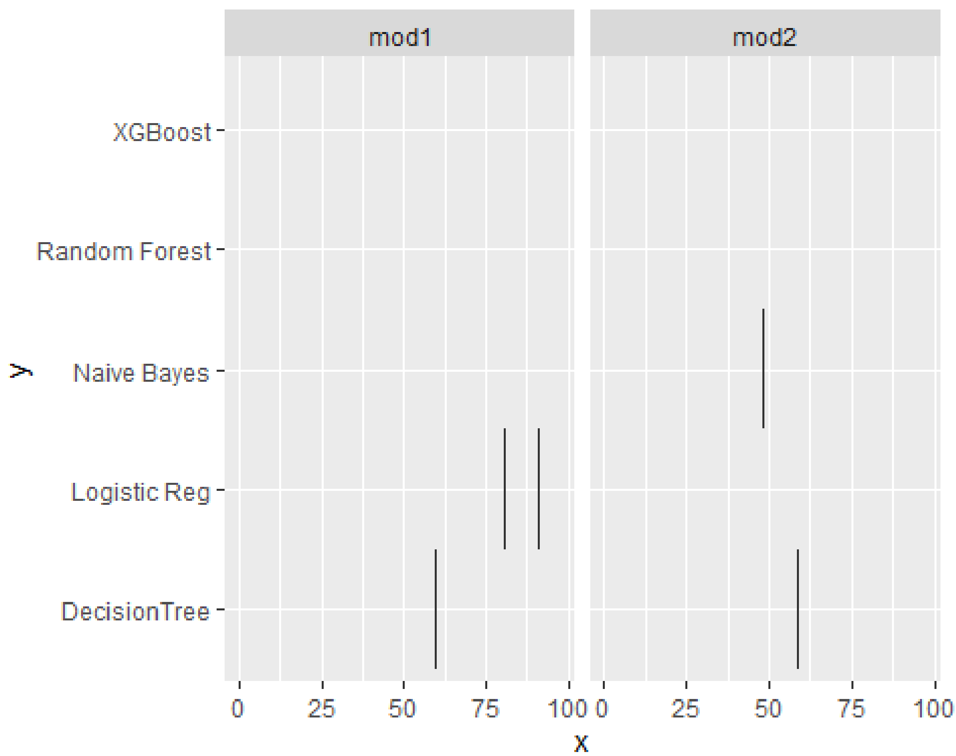
```
  geom_abline(
    aes(intercept = a1, slope = a2, colour = -dist),
    data = filter(models, rank(dist) <= 0)
  )
```



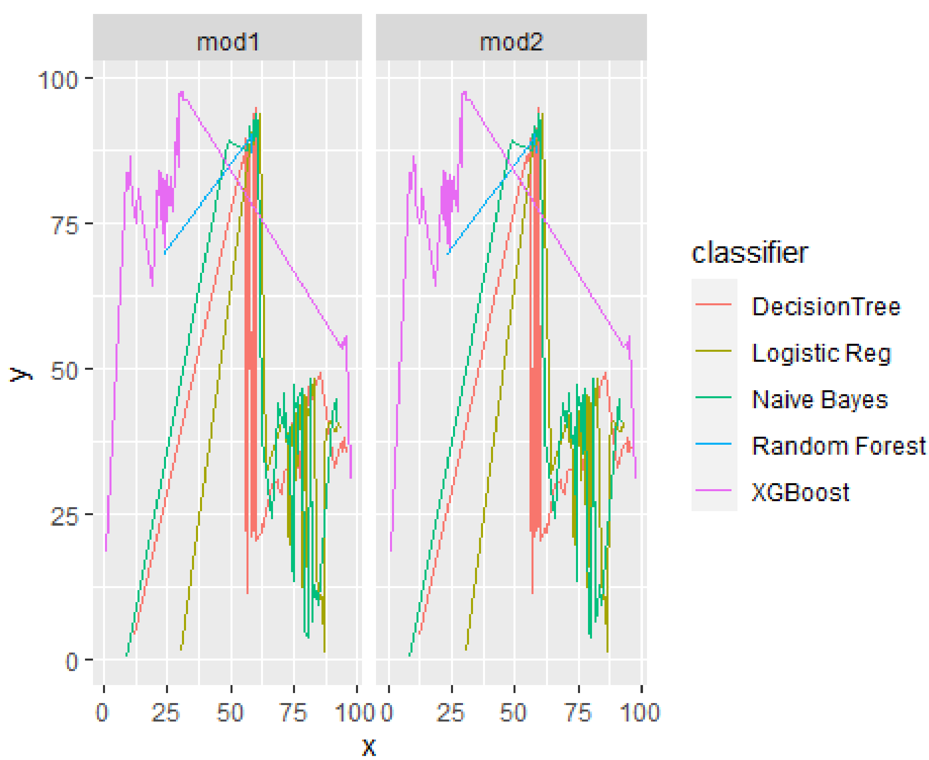

```
# grid <- expand.grid(
#   a1 = seq(-5, 20, length = 25),
#   a2 = seq(1, 3, length = 25)
# ) %>%
#   mutate(dist = purrr::map2_dbl(x, y, data_dist))
#
# grid %>%
#   ggplot(aes(x, y)) +
#   geom_point(data = filter(grid, rank(dist) <= 10), size = 4, colour =
"red") +
#   geom_point(aes(colour = -dist))
#
foo <- foo %>%
  gather_residuals(mod1, mod2)

ggplot(foo, aes(x, y, colour = classifier)) +
  geom_point() +
  facet_grid(model ~ classifier)
```


```r
mod1 <- lm(y ~ x + classifier, data = table)
mod2 <- lm(y ~ x * classifier, data = table)
```

```r
mod1 <- lm(y ~ x + classifier, data = table)
mod2 <- lm(y ~ x * classifier, data = table)

# Problematic Code
# grid <- foo %>%
#   data_grid(
#     x = seq_range(x, 5),
#     y = seq_range(y, 5)
#   ) %>%
#   gather_predictions(mod1, mod2)
# grid


ggplot(foo, aes(x, y)) +
  geom_tile(aes(y = classifier)) +
  facet_wrap(~ model)
```

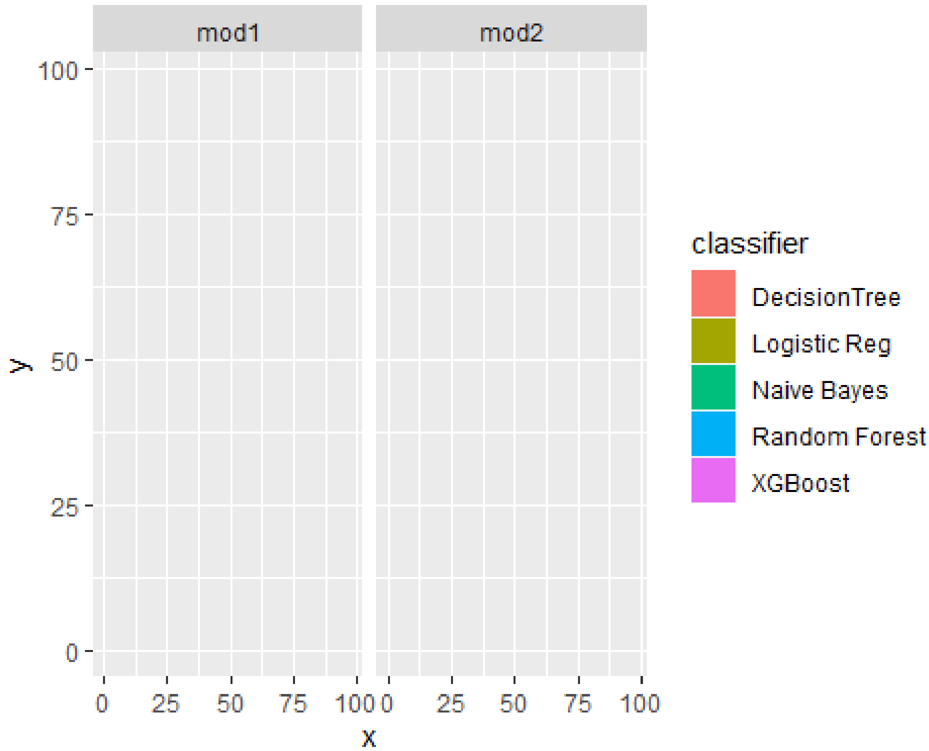



```r
ggplot(foo, aes(x, y, colour= classifier, group = classifier)) +
  geom_line() +
  facet_wrap(~ model)
```



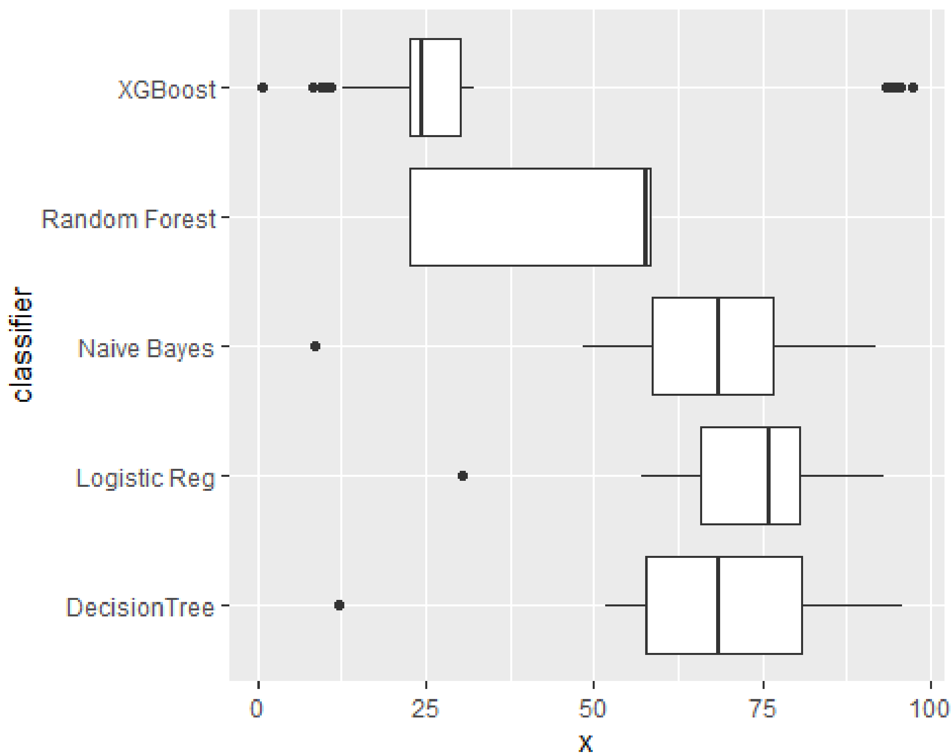
```
ggplot(foo, aes(y, x, colour = classifier, group = classifier)) +
  geom_line() +
  facet_wrap(~ model)
```


```
ggplot(foo, aes(x, y)) +
  geom_tile(aes(fill = classifier)) +
  facet_wrap(~ model)
```

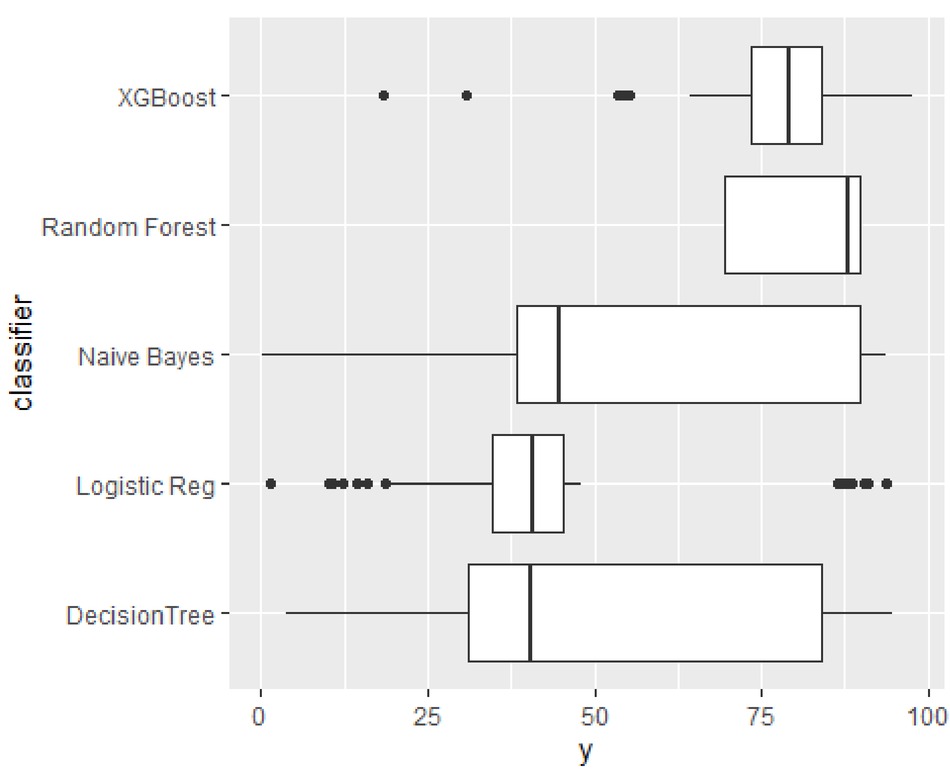


Chapter 24 Model Building

```
library(tidyverse)
library(modelr)
options(na.action = na.warn)

# library(nycflights13)
# library(lubridate)
# 24.2
ggplot(foo, aes(x, classifier)) + geom_boxplot()
```

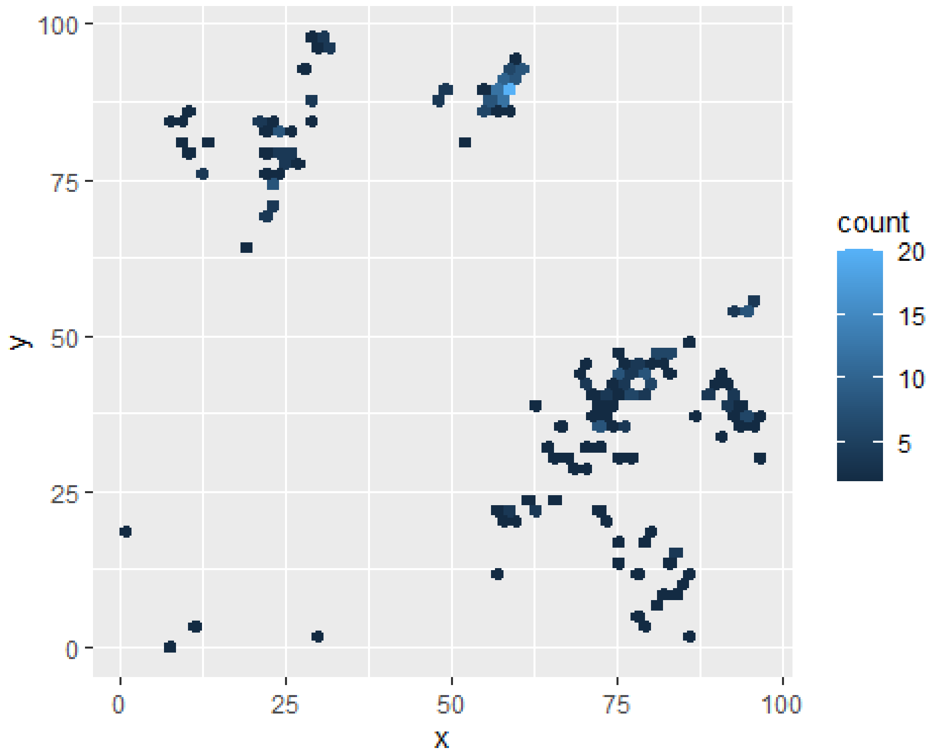



```
ggplot(foo, aes(y, classifier)) + geom_boxplot()
```

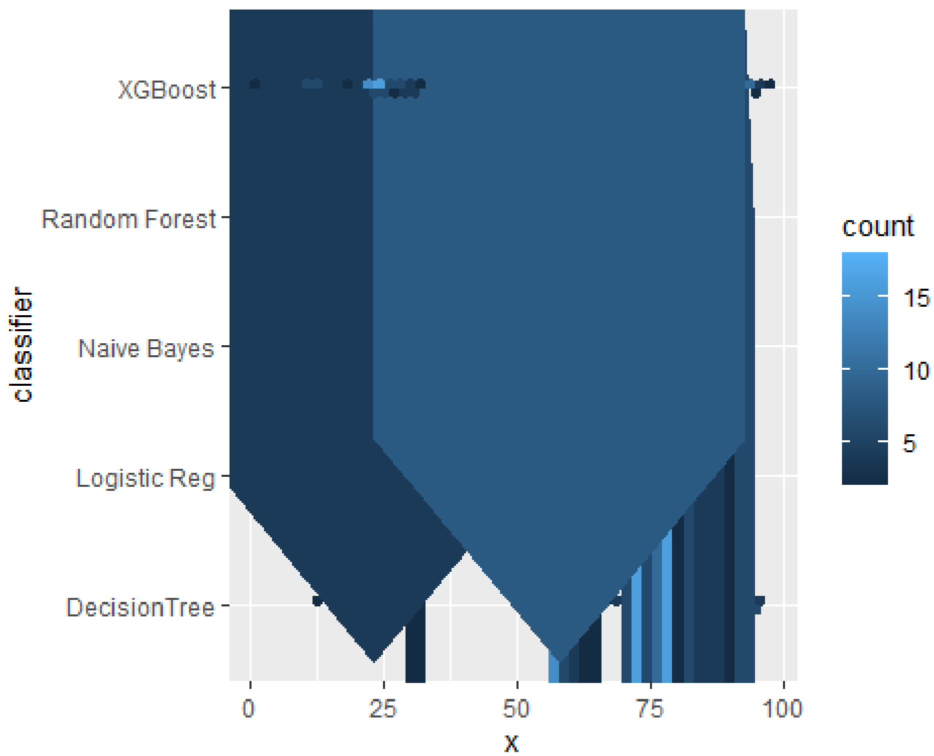


24.2.1

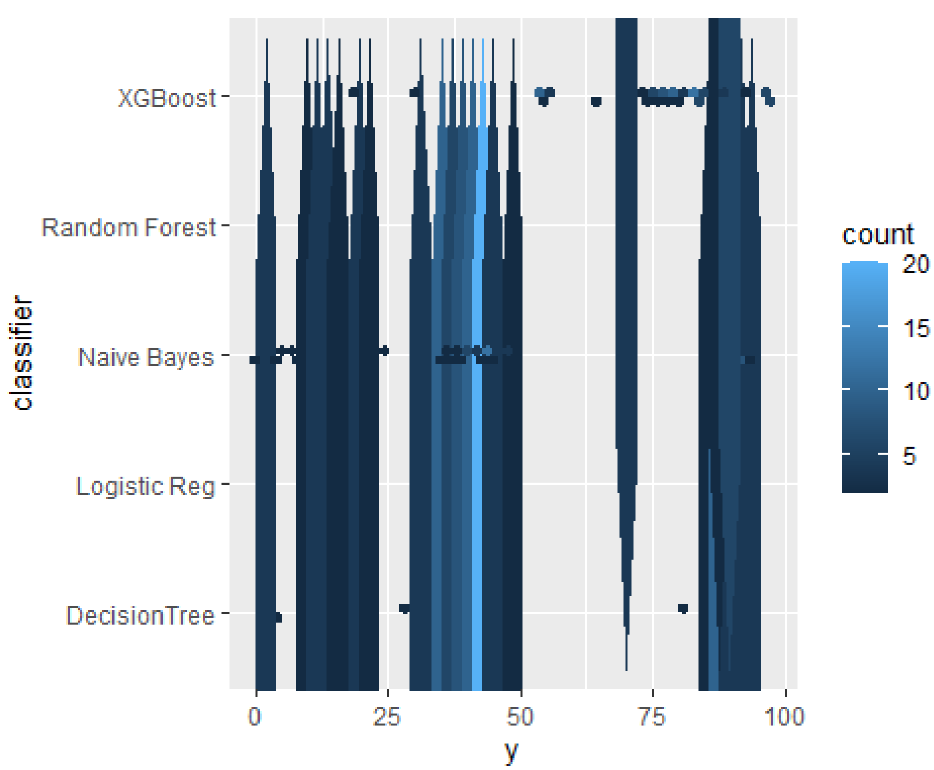
```
ggplot(foo, aes(x, y)) +
  geom_hex(bins = 50)
```




```
ggplot(foo, aes(x, classifier)) +
  geom_hex(bins = 50)
```

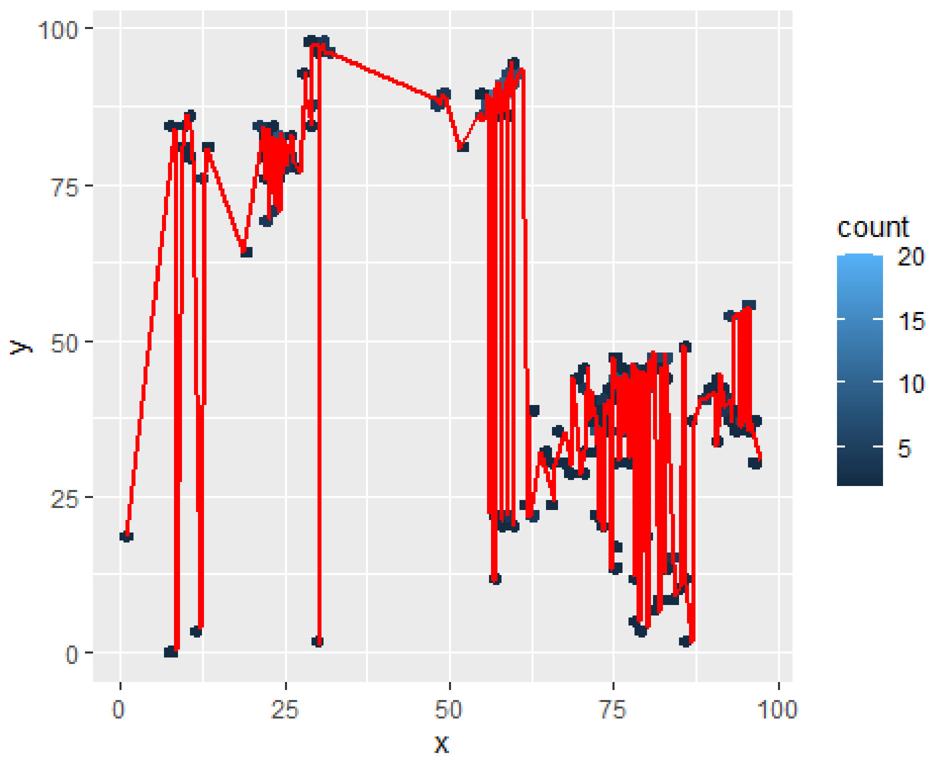



```
ggplot(foo, aes(y, classifier)) +
  geom_hex(bins = 50)
```



```
# Code to ignore
# foo <- foo %>%
#   add_residuals(mod_foo, "lclassifier")
#
# ggplot(foo, aes(x, lclassifier)) +
#   geom_hex(bins = 50)
```

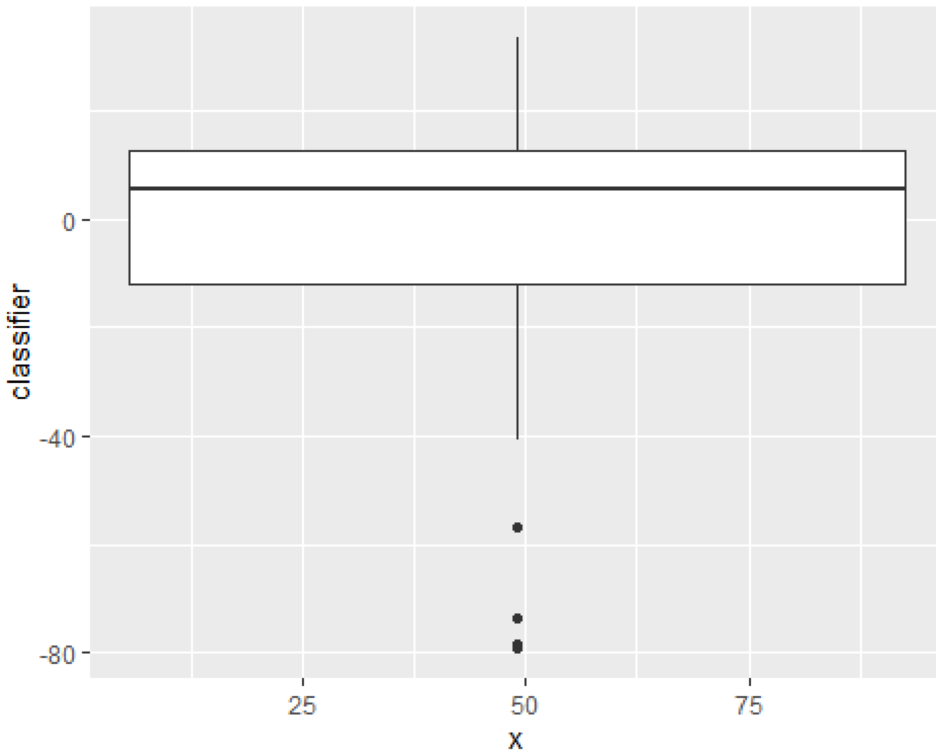
```
# grid <- foo2 %>%
#   data_grid(x = seq_range(x), 20)) %>%
#   mutate(x = log2(x)) %>%
#   add_predictions(mod_foo2, "l_x") %>%
#   mutate(x = 2 ^ x)

lm1 <- lm(y ~ classifier, data=foo)
lm2 <- lm(x ~ classifier, data=foo)

ggplot(foo, aes(x, y)) +
  geom_hex(bins = 50) +
  geom_line(data = foo, colour = "red", size = 1)
```



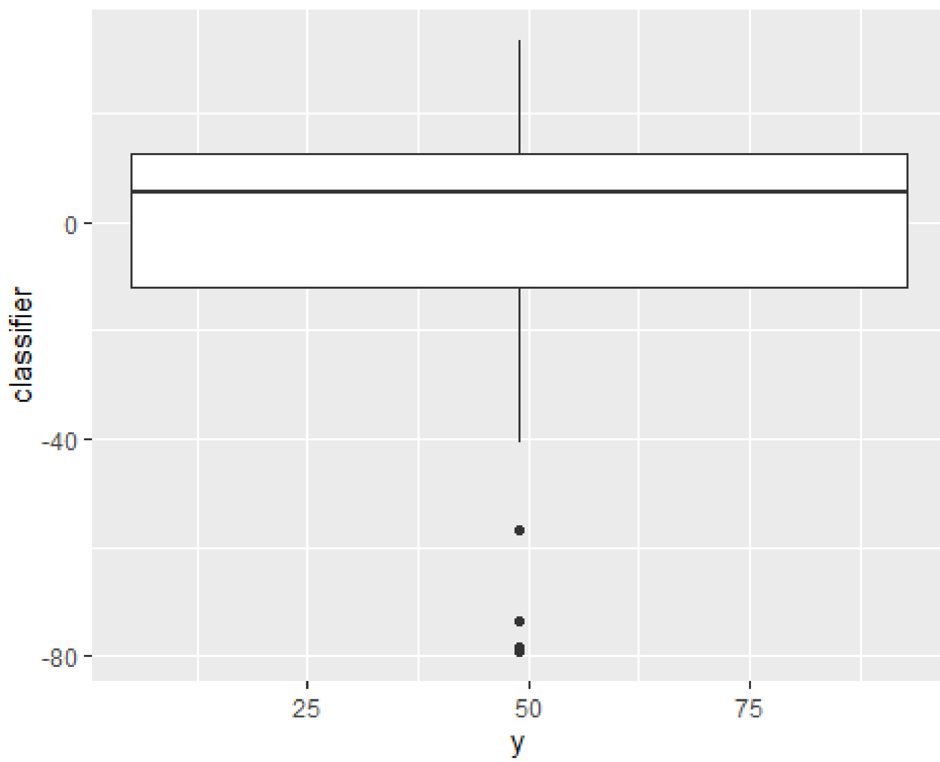

```
# foo <- foo %>%
#   filter(x <= 1) %>%
#   mutate(l_x = log2(l_x), l_y = log2(y))
```

```
# ggplot(foo, aes(l_x, l_y)) +
#   geom_hex(bins = 50)
```

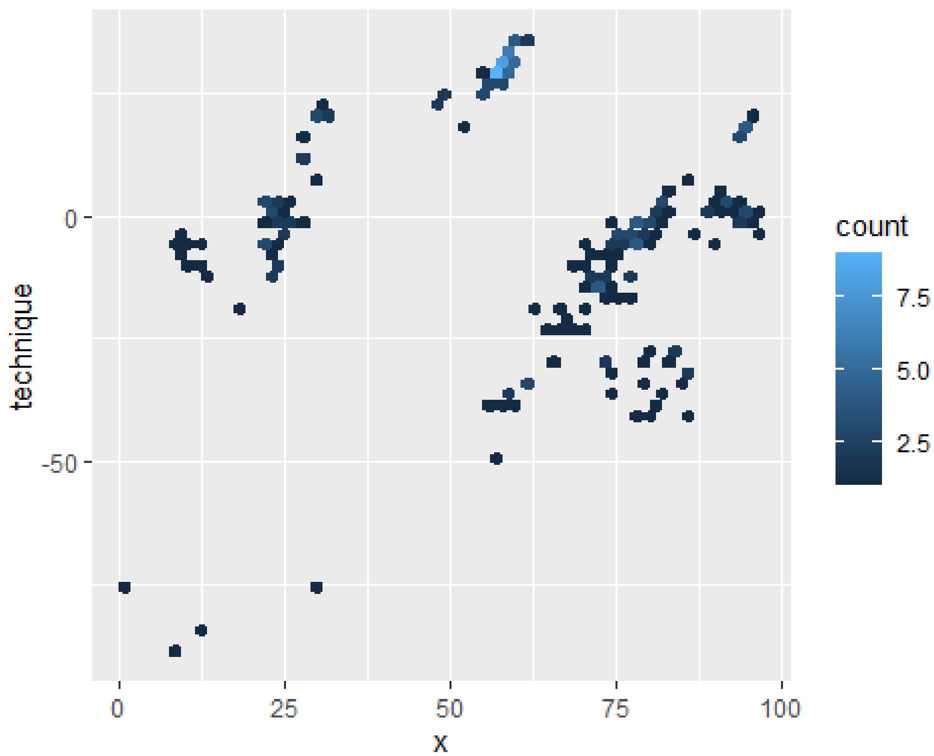
```
mod_foo <- lm(x ~ y, data = foo)
foo <- foo %>%
  add_residuals(mod_foo, "classifier")

ggplot(foo, aes(x, classifier)) + geom_boxplot()
```



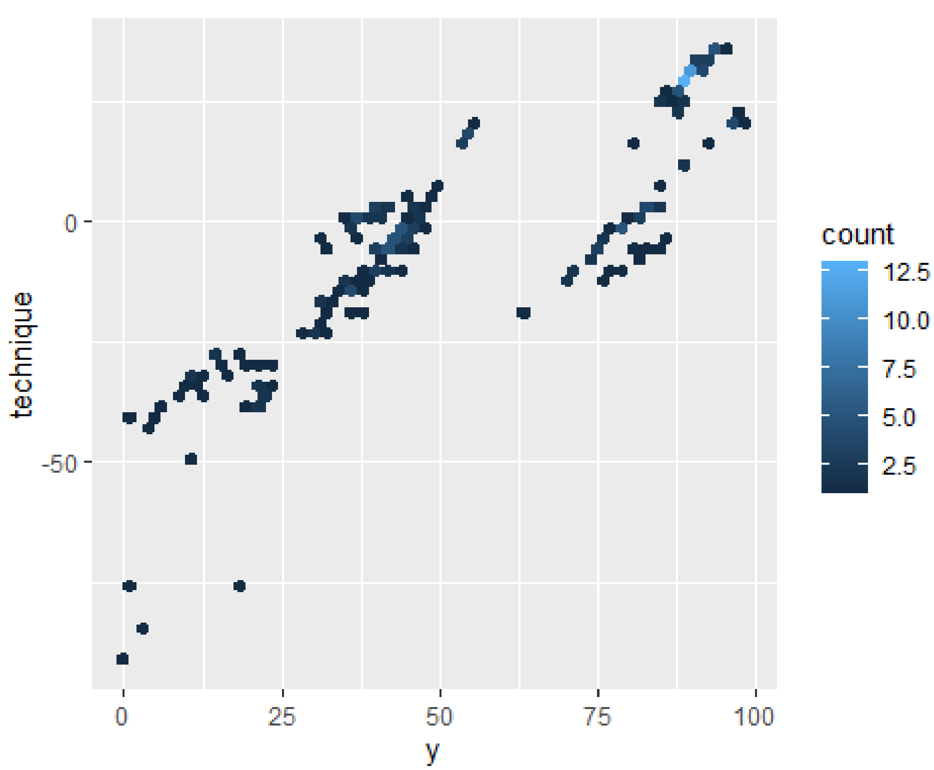

```
ggplot(foo, aes(y, classifier)) + geom_boxplot()
```



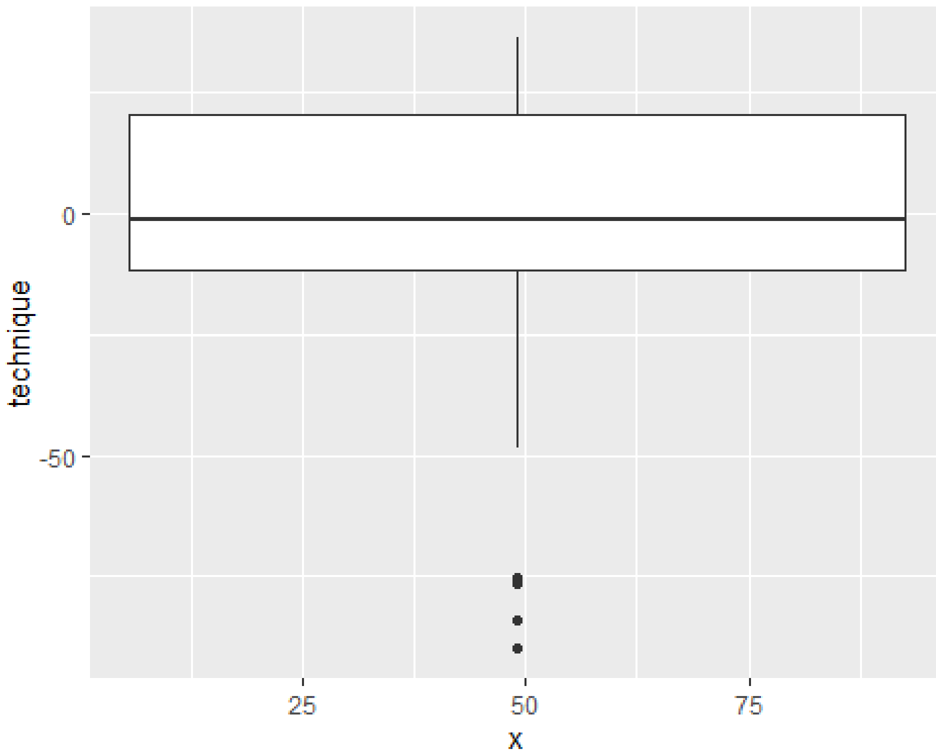

```
foo = tibble(x=precision, y=recall, classifier=classifier, sampling=sampling,
technique=technique, year=year)
mod_foo <- lm(y ~ x + classifier + sampling + technique, data = foo)
mod_foo <- lm(y ~ x, data = foo)
# grid <- foo2 %>%
#   data_grid(x = seq_range(x, 225)) %>%
#   mutate(l_x = log2(x)) %>%
#   add_predictions(mod_foo2, "l_y") %>%
#   mutate(l_y = log2(y))

foo <- foo %>%
    add_residuals(mod_foo, "technique")

ggplot(foo, aes(x, technique)) +
  geom_hex(bins = 50)
```



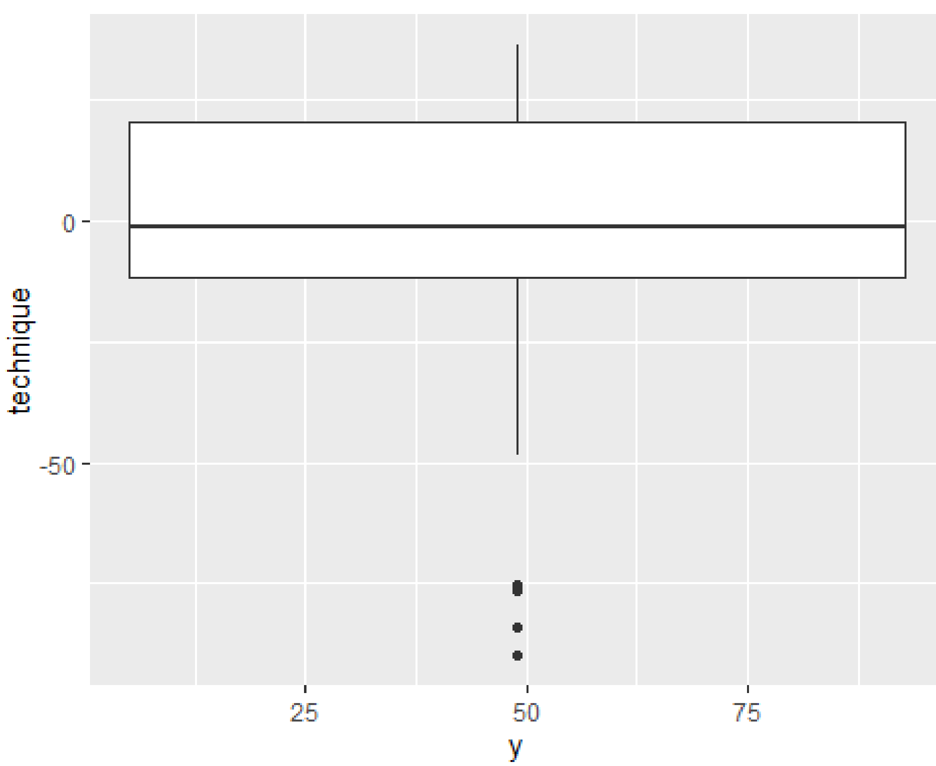

```r
ggplot(foo, aes(y, technique)) +
  geom_hex(bins = 50)
```



```
ggplot(foo, aes(x, technique)) + geom_boxplot()
```

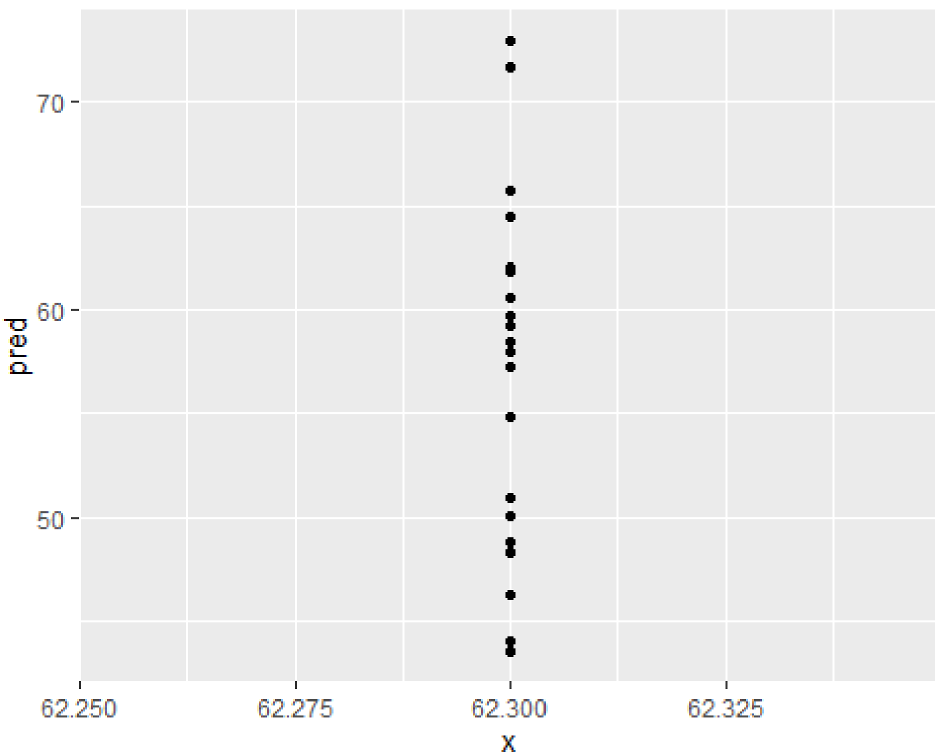



```
ggplot(foo, aes(y, technique)) + geom_boxplot()
```

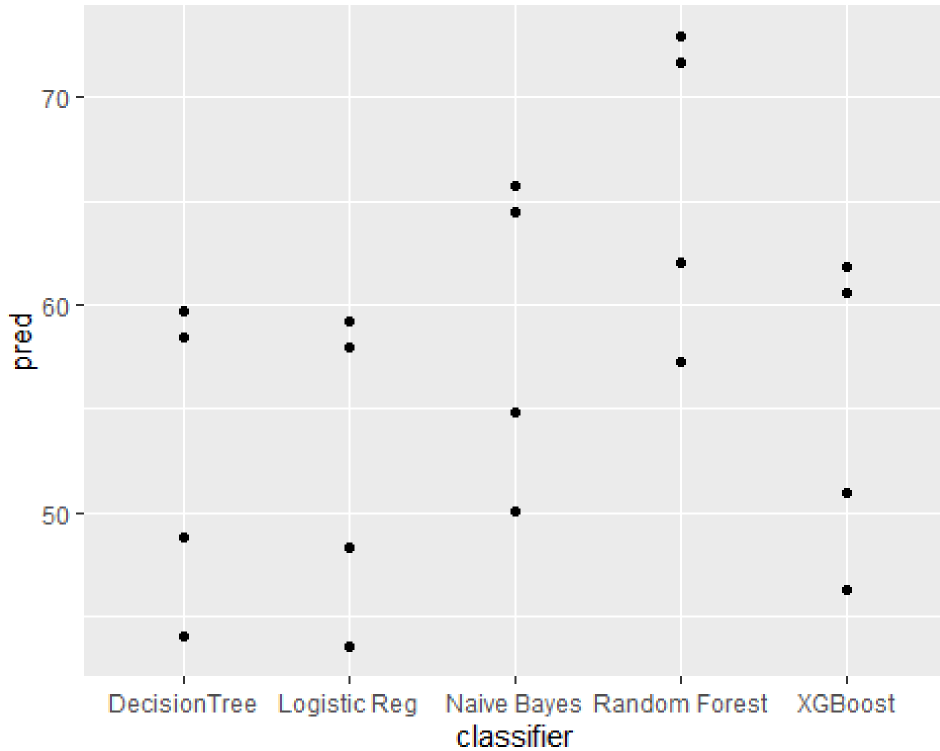


Chapter 24.2.2

```
 foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
 mod_foo <- lm(y ~ x + classifier + sampling + technique, data = foo)

grid <- foo %>%
  data_grid(classifier, .model = mod_foo) %>%
  add_predictions(mod_foo)
grid

## # A tibble: 30 × 5
##    classifier       x sampling      technique  pred
##    <chr>        <dbl> <chr>         <chr>     <dbl>
##  1 DecisionTree  62.3 Undersampling N/A        59.7
##  2 DecisionTree  62.3 Undersampling NearMiss   58.5
##  3 DecisionTree  62.3 Undersampling ROS        44.1
##  4 DecisionTree  62.3 Undersampling RUS        48.8
##  5 DecisionTree  62.3 Undersampling SMOTE      59.7
##  6 DecisionTree  62.3 Undersampling Tomelinks  59.7
##  7 Logistic Reg  62.3 Undersampling N/A        59.2
##  8 Logistic Reg  62.3 Undersampling NearMiss   58.0
##  9 Logistic Reg  62.3 Undersampling ROS        43.6
## 10 Logistic Reg  62.3 Undersampling RUS        48.3
## # … with 20 more rows

ggplot(grid, aes(x, pred)) +
  geom_point()
```

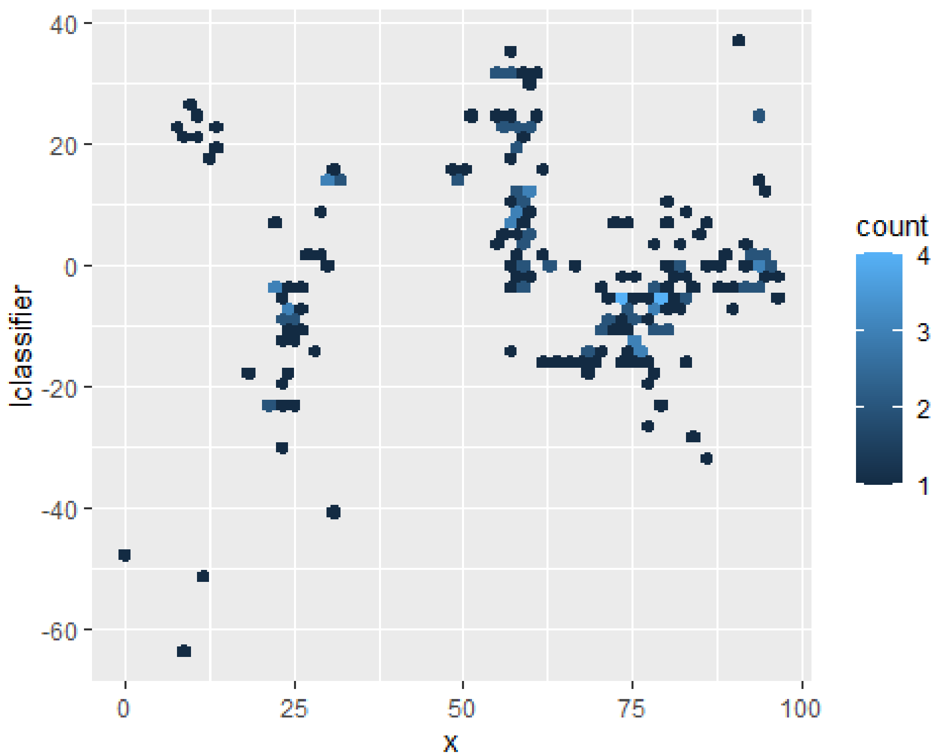



```
#> # A tibble: 5 x 5
#>   cut       lcarat color clarity  pred
#>   <ord>      <dbl> <chr> <chr>   <dbl>
#> 1 Fair      -0.515 G     VS2      11.2
#> 2 Good      -0.515 G     VS2      11.3
#> 3 Very Good -0.515 G     VS2      11.4
#> 4 Premium   -0.515 G     VS2      11.4
#> 5 Ideal     -0.515 G     VS2      11.4

ggplot(grid, aes(classifier, pred)) +
  geom_point()
```

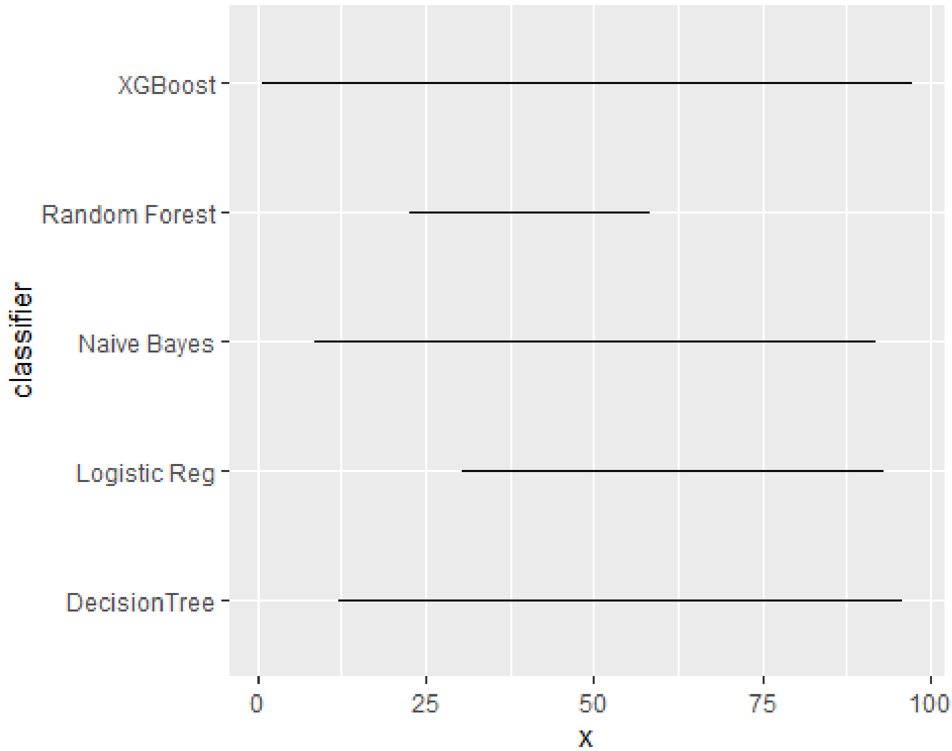


24.2.2

```
foo <- foo %>%
add_residuals(mod_foo, "lclassifier")

ggplot(foo, aes(x, lclassifier)) +
   geom_hex(bins = 50)
```

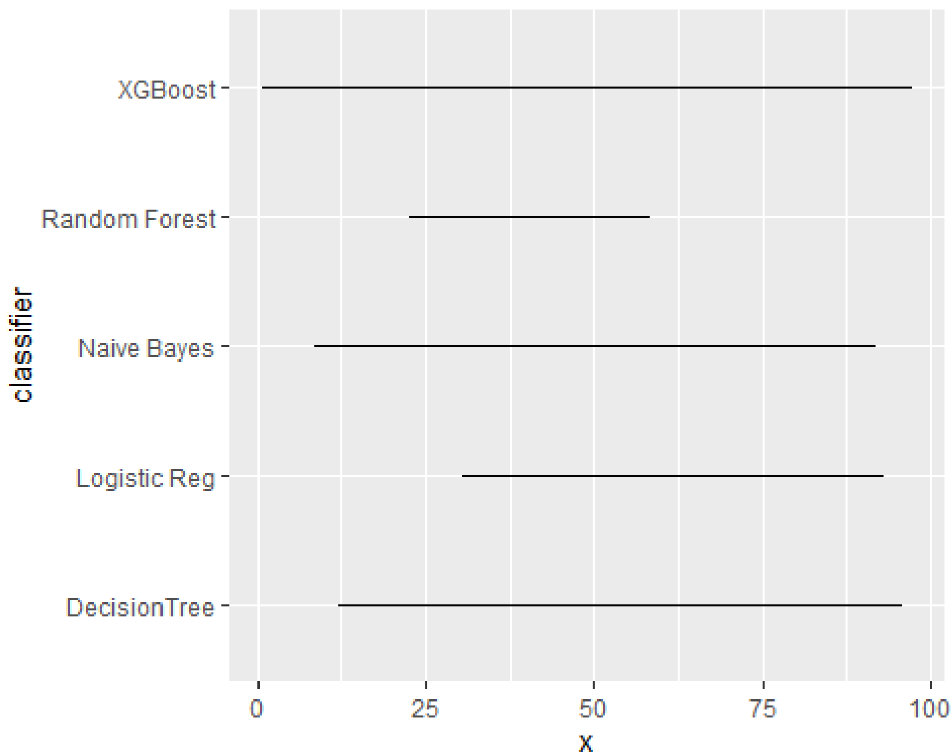



```
foo <- foo %>%
  add_residuals(mod_foo)


ggplot(foo, aes(x, classifier)) +
  geom_line()
```

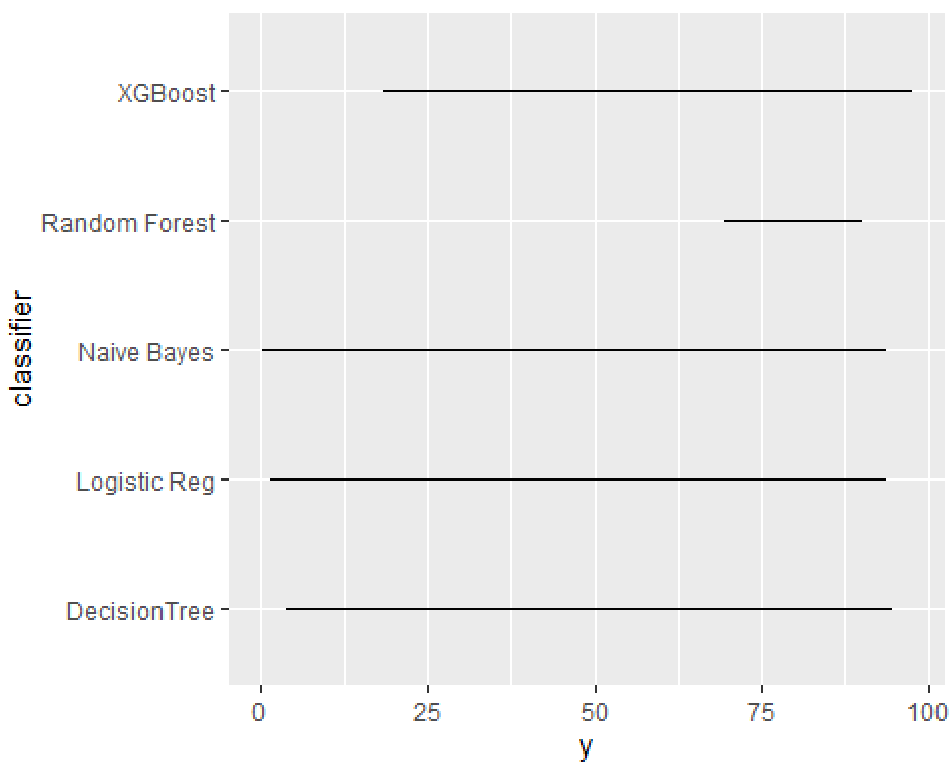


Chapter 24.3

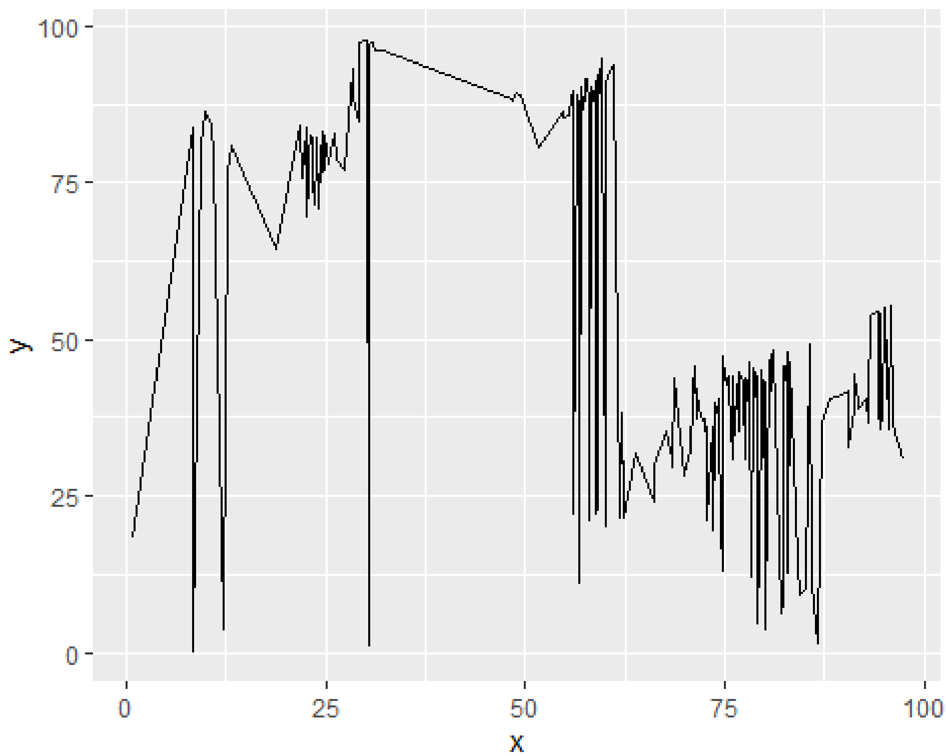
```
# foo2 %>%
# filter(abs(l_x) > 1) %>%
#  add_predictions(mod_foo) %>%
#  mutate(pred = pred) %>%
#  select(l_x, pred, l_y:all_of(foo), x:y) %>%
#  arrange(x)

ggplot(foo, aes(x, classifier)) +
  geom_line()
```




```
ggplot(foo, aes(y, classifier)) +
  geom_line()
```

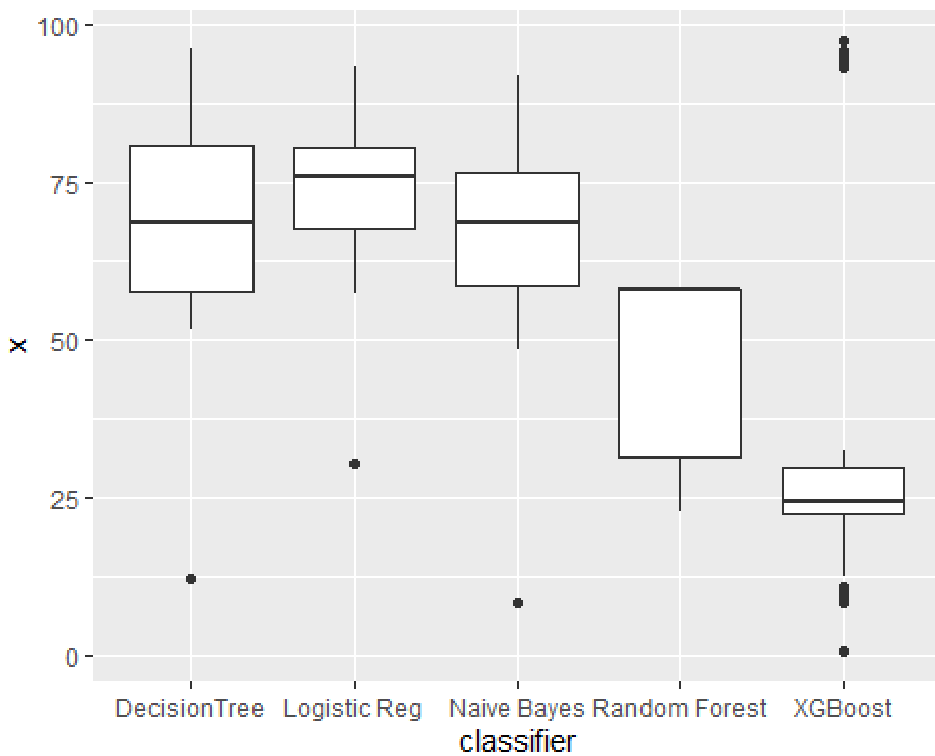


```
ggplot(foo, aes(x, y)) +
  geom_line()
```

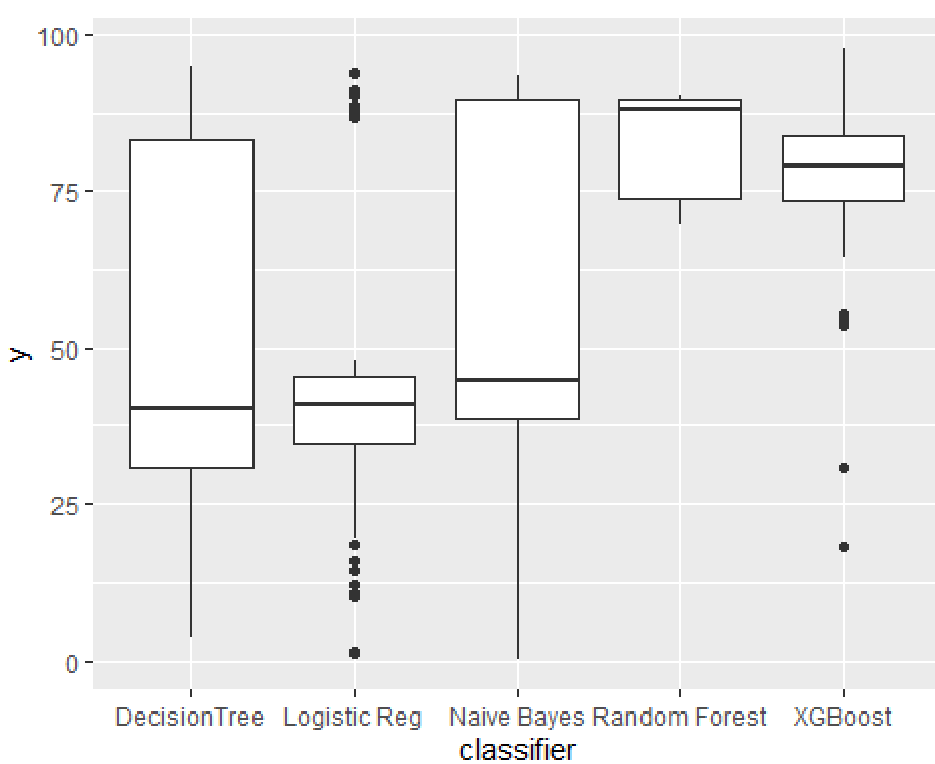


Chapter 24.3.1

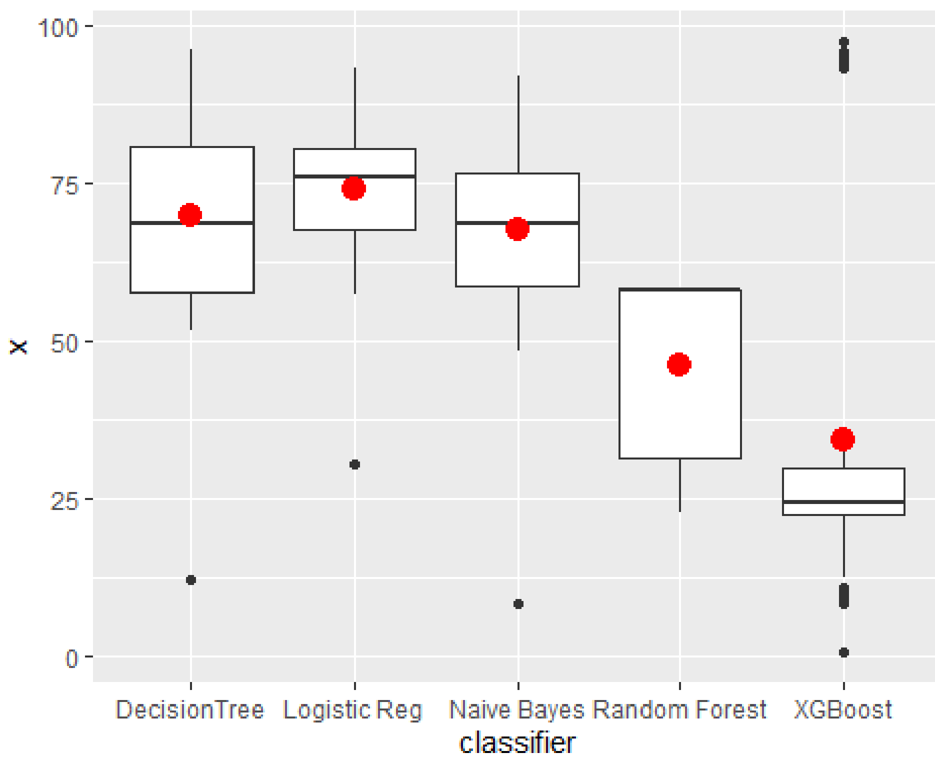
```
ggplot(foo, aes(classifier, x)) +
  geom_boxplot()
```



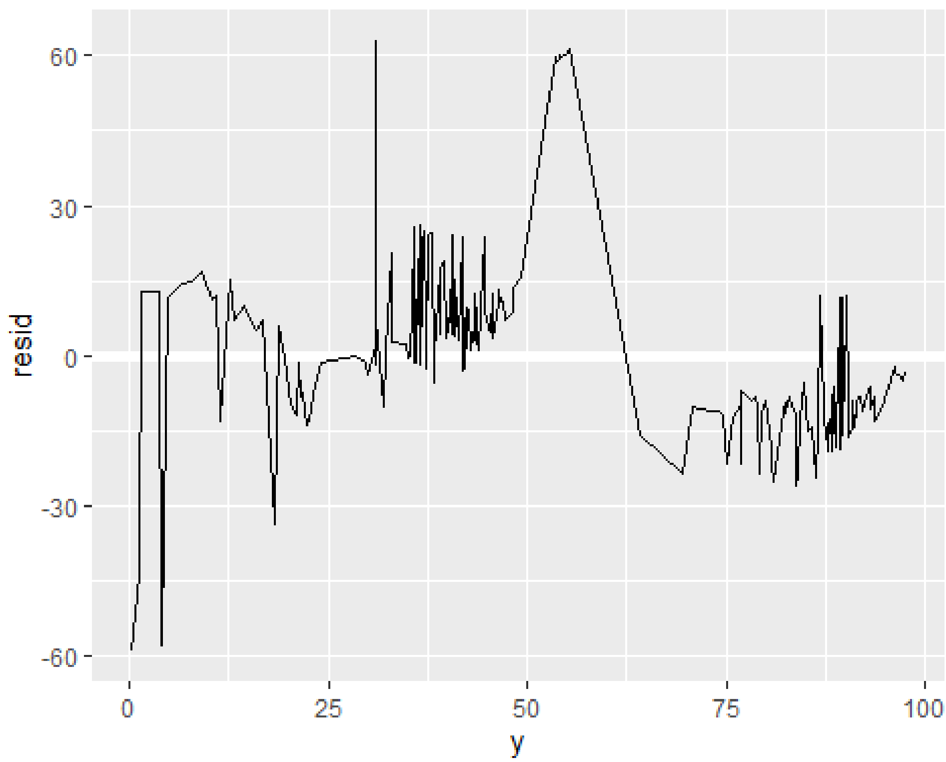

```
ggplot(foo, aes(classifier, y)) +
  geom_boxplot()
```



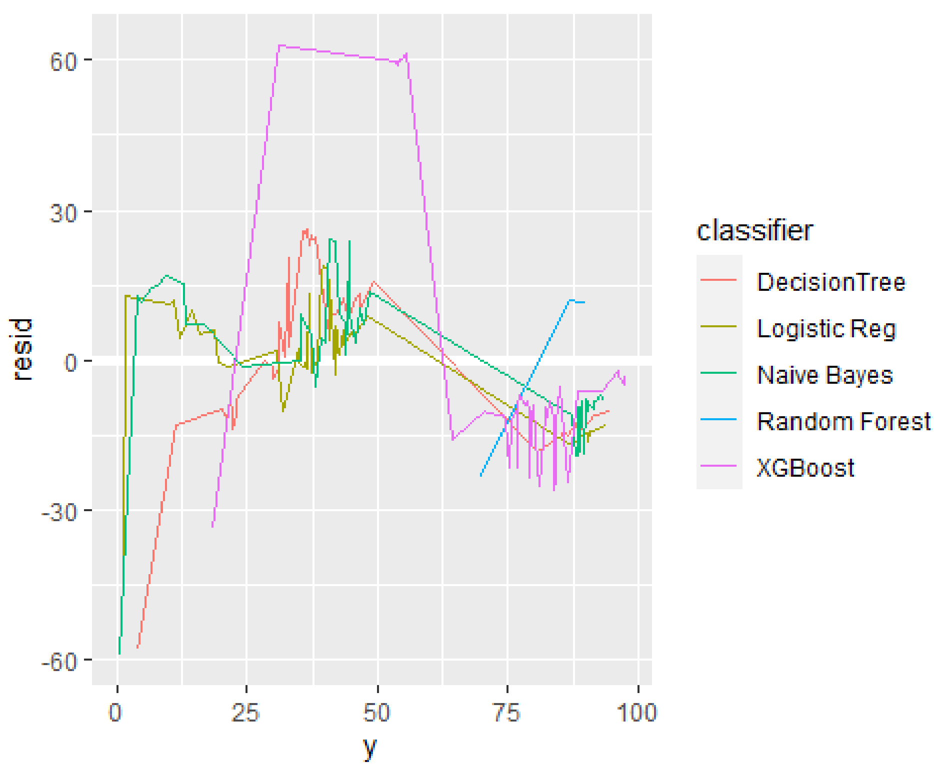
```
mod <- lm(x ~ classifier, data = foo)


grid <- foo %>%
  data_grid(classifier) %>%
  add_predictions(mod, "x")

ggplot(foo, aes(classifier, x)) +
  geom_boxplot() +
  geom_point(data = grid, colour = "red", size = 4)
```



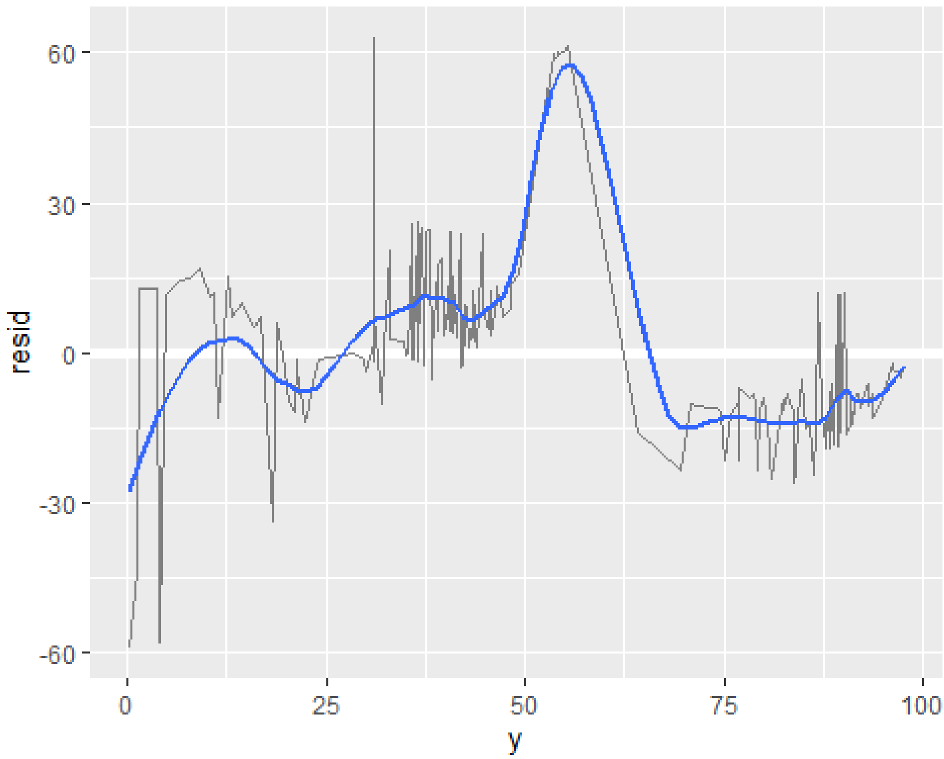

```
foo <- foo %>%
  add_residuals(mod)

foo %>%
  ggplot(aes(y, resid)) +
  geom_ref_line(h = 0) +
  geom_line()
```



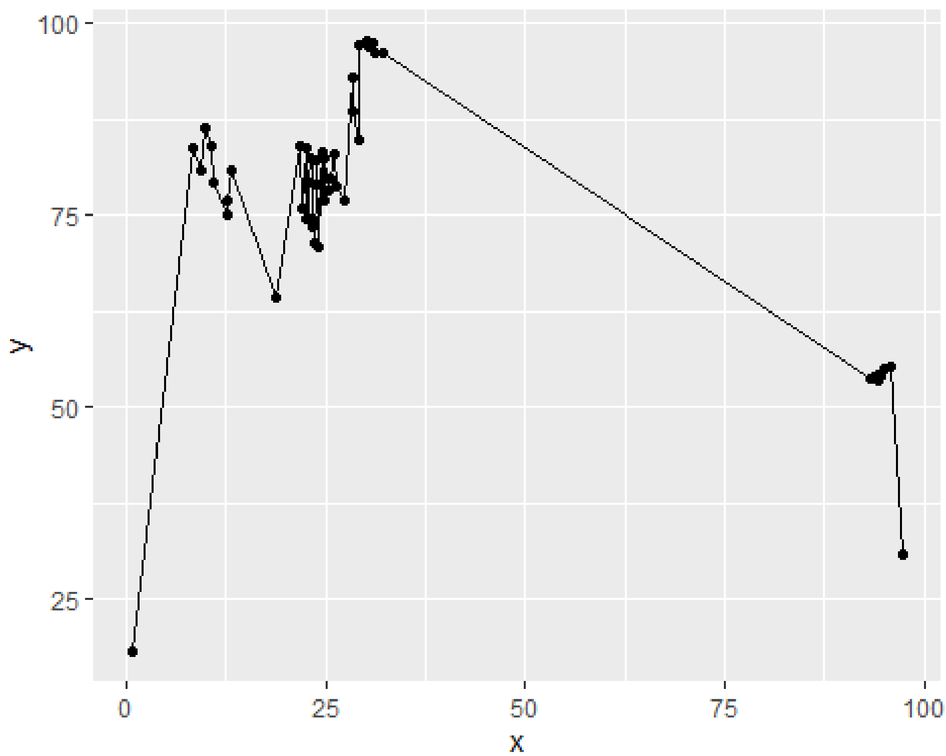

```
ggplot(foo, aes(y, resid, colour = classifier)) +
  geom_ref_line(h = 0) +
  geom_line()
```



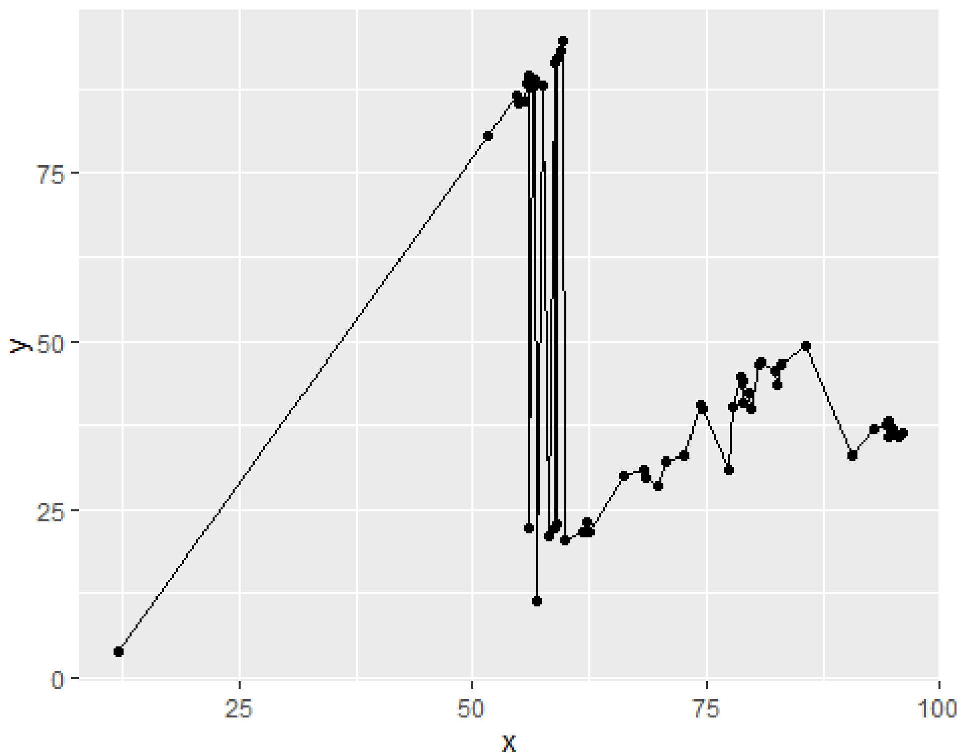
```r
foo %>%
  ggplot(aes(y, resid)) +
  geom_ref_line(h = 0) +
  geom_line(colour = "grey50") +
  geom_smooth(se = FALSE, span = 0.20)
```



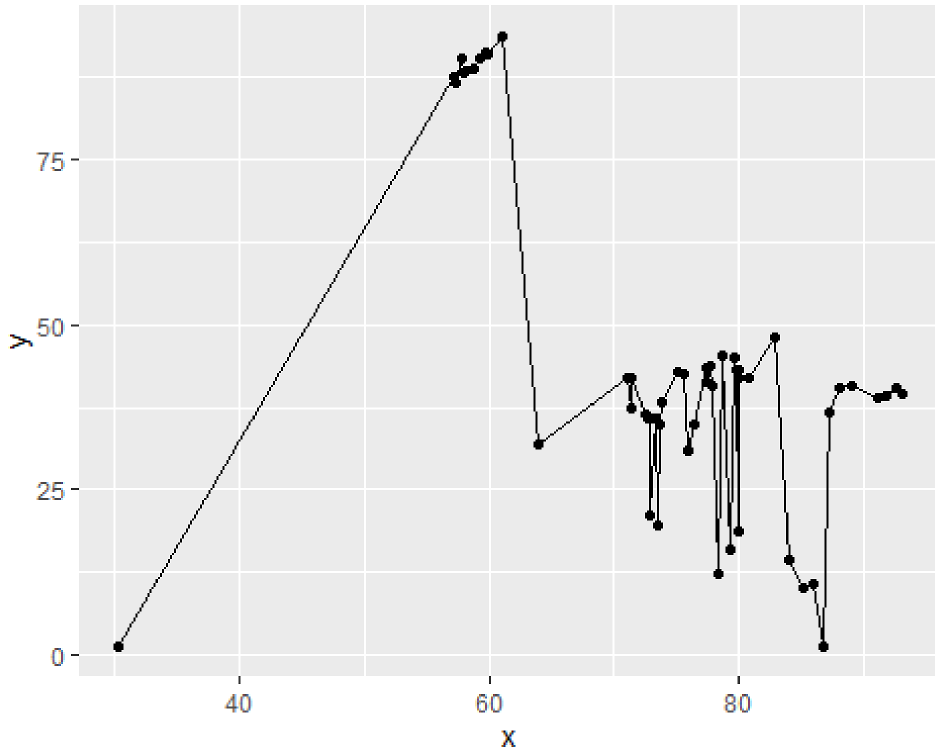

```r
foo %>%
  filter(classifier == "XGBoost") %>%
  ggplot(aes(x, y)) +
  geom_point() +
  geom_line()
```



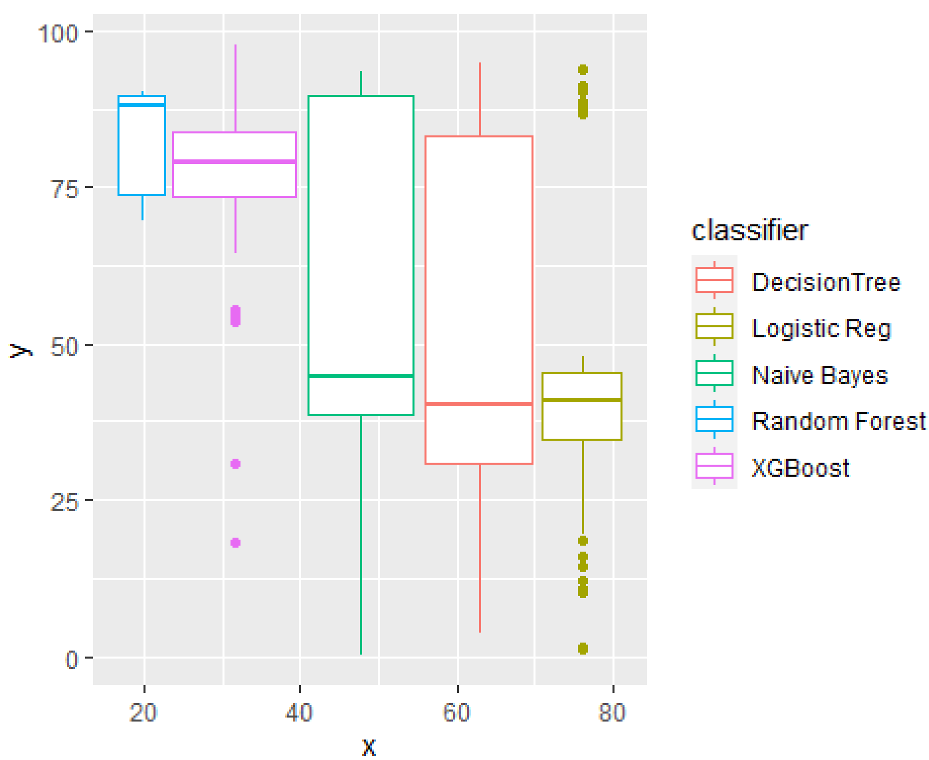

```
foo %>%
  filter(classifier == "DecisionTree") %>%
  ggplot(aes(x, y)) +
  geom_point() +
  geom_line()
```




```
foo %>%
  filter(classifier == "Logistic Reg") %>%
  ggplot(aes(x, y)) +
  geom_point() +
  geom_line()
```



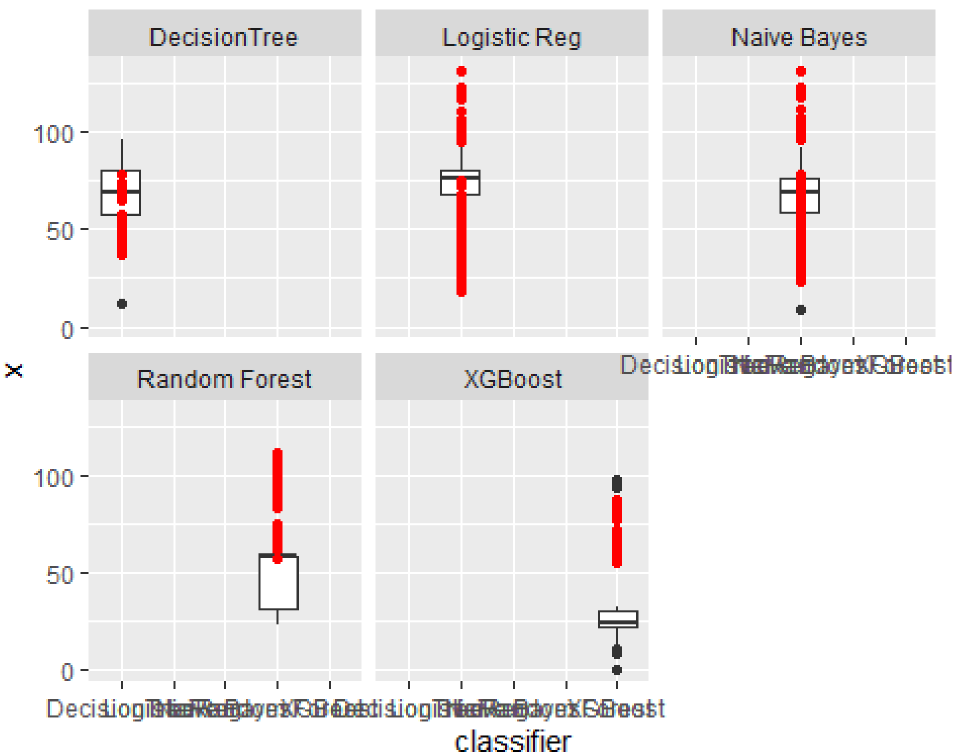

```
foo %>%
  ggplot(aes(x, y, colour = classifier)) +
  geom_boxplot()
```



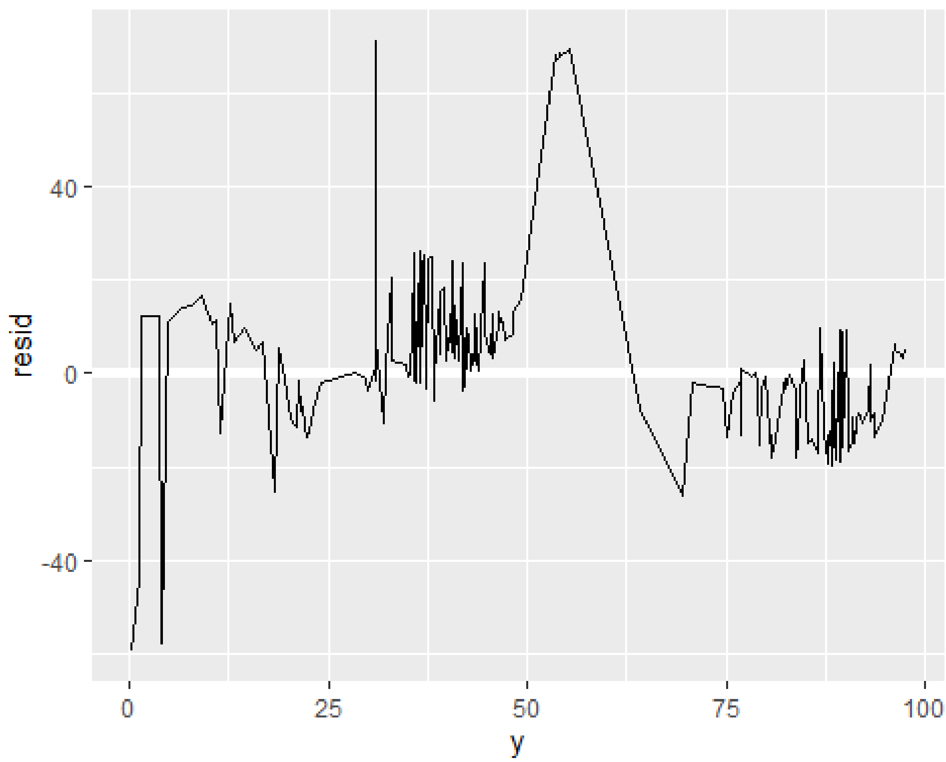
<br

```
mod3 <- lm(x ~ classifier, data = foo)
mod4 <- lm(y ~ classifier, data = foo)

foo %>%
  gather_residuals(precision = mod3, recall = mod4) %>%
  ggplot(aes(x, resid, colour = model)) +
  geom_line(alpha = 0.75)
```

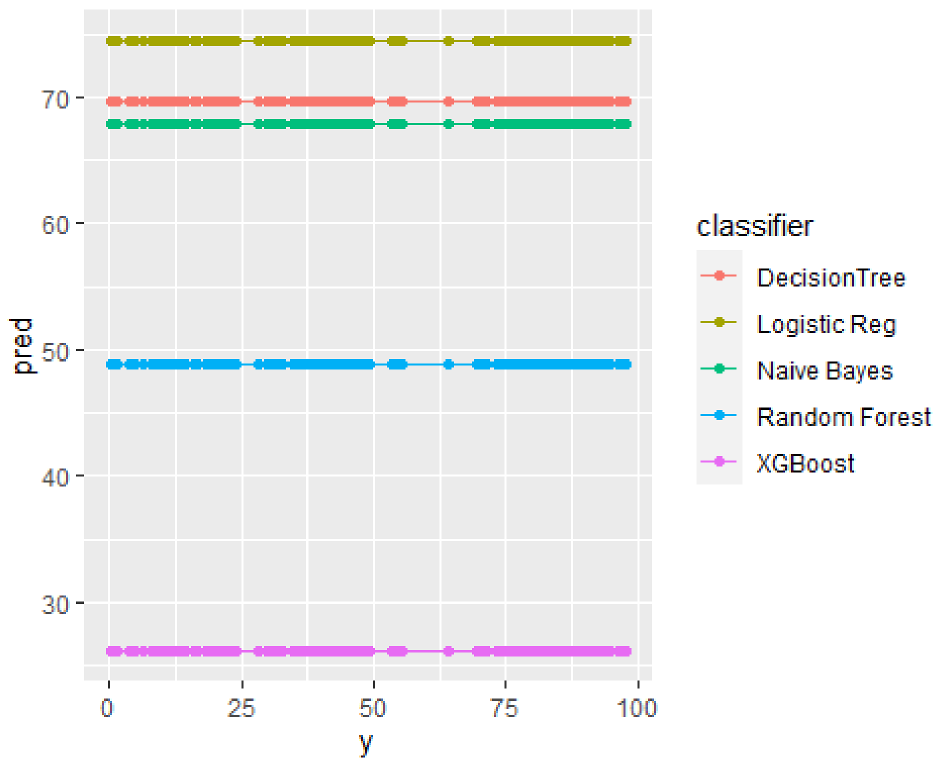

```
grid <- foo %>%
  data_grid(x, classifier) %>%
  add_predictions(mod2, "x")

ggplot(foo, aes(classifier, x)) +
  geom_boxplot() +
  geom_point(data = grid, colour = "red") +
  facet_wrap(~ classifier)
```




```
library(splines)
mod6 <- MASS::rlm(x ~ classifier, data = foo)

foo %>%
  add_residuals(mod6, "resid") %>%
  ggplot(aes(y, resid)) +
  geom_hline(yintercept = 0, size = 2, colour = "white") +
  geom_line()
```

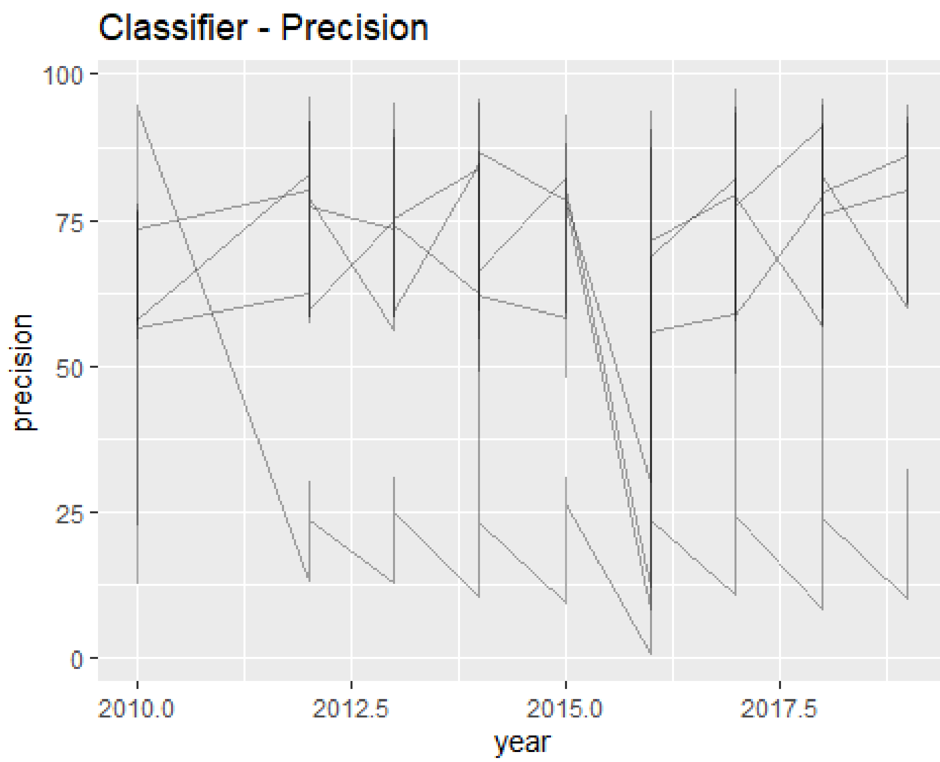



```
foo %>%
  data_grid(y, classifier) %>%
  add_predictions(mod6) %>%
  ggplot(aes(y, pred, colour = classifier)) +
  geom_line() +
  geom_point()
```

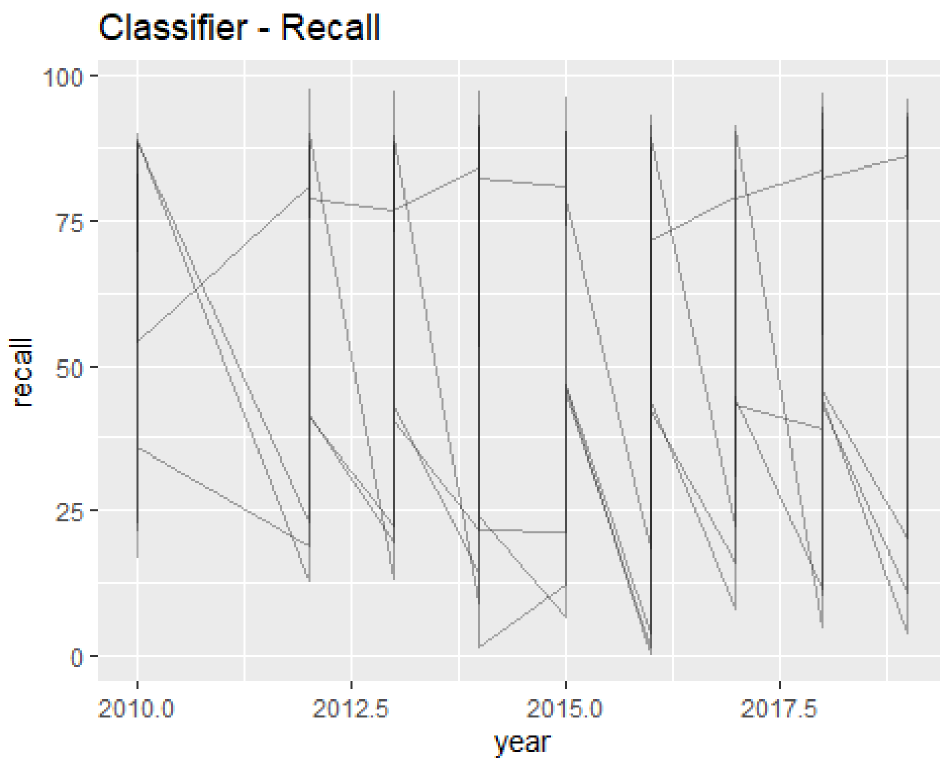


Chapter 25.1 Many Models

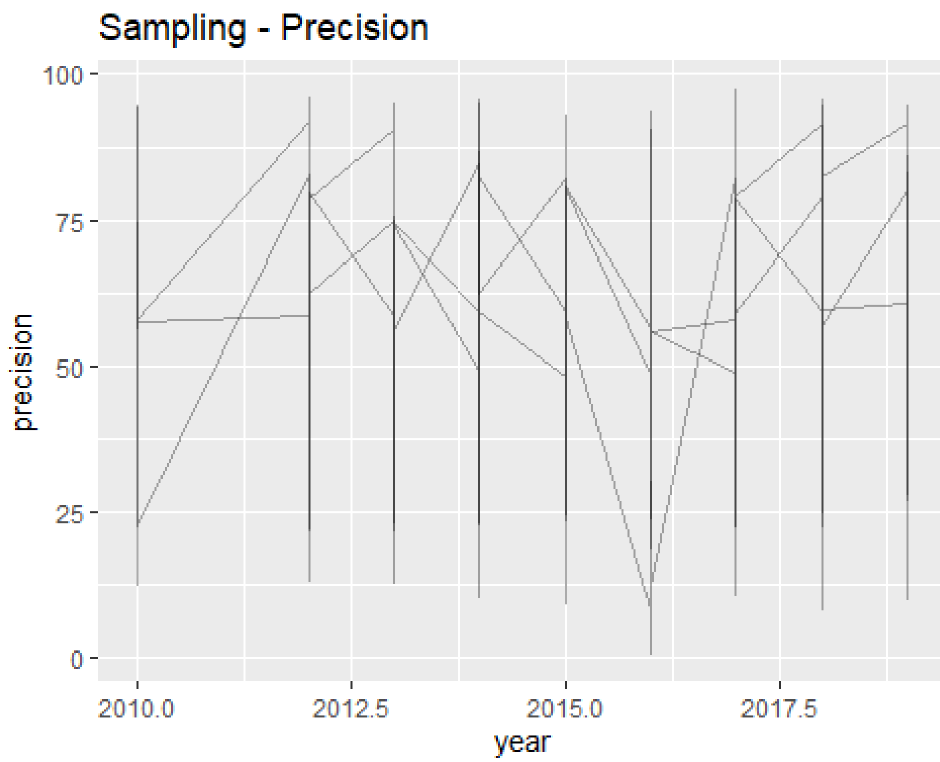
```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
foo %>%
  ggplot(aes(year, precision, group = classifier)) +
  geom_line(alpha = 1/3) +
    ggtitle("Classifier - Precision")
```



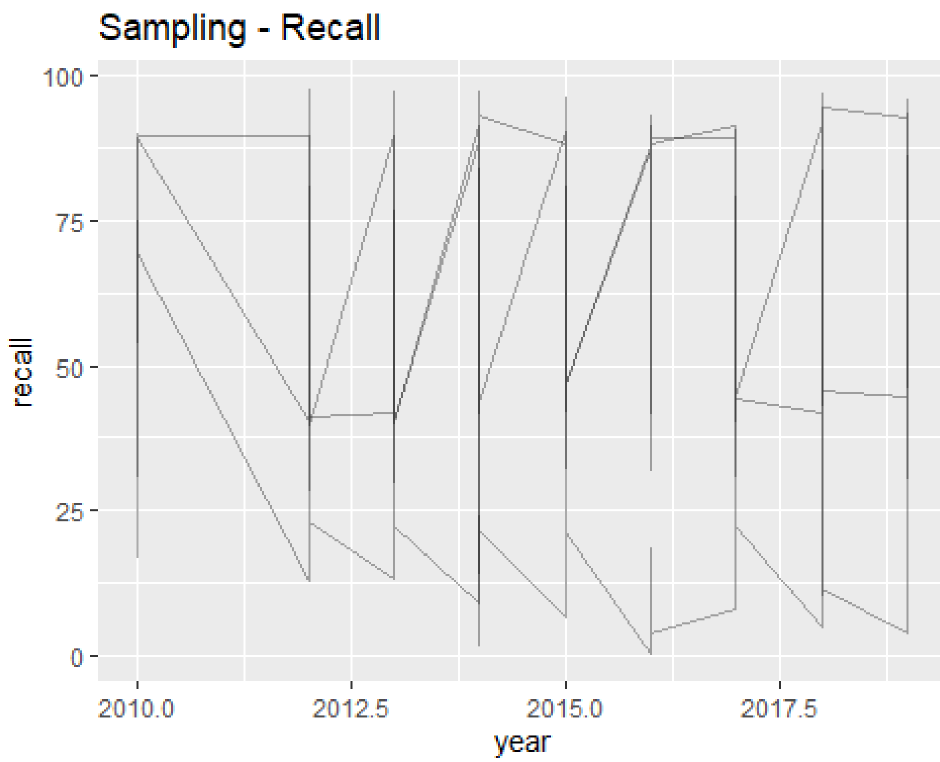

```
foo %>%
  ggplot(aes(year, recall, group = classifier)) +
  geom_line(alpha = 1/3) +
  ggtitle("Classifier - Recall")
```



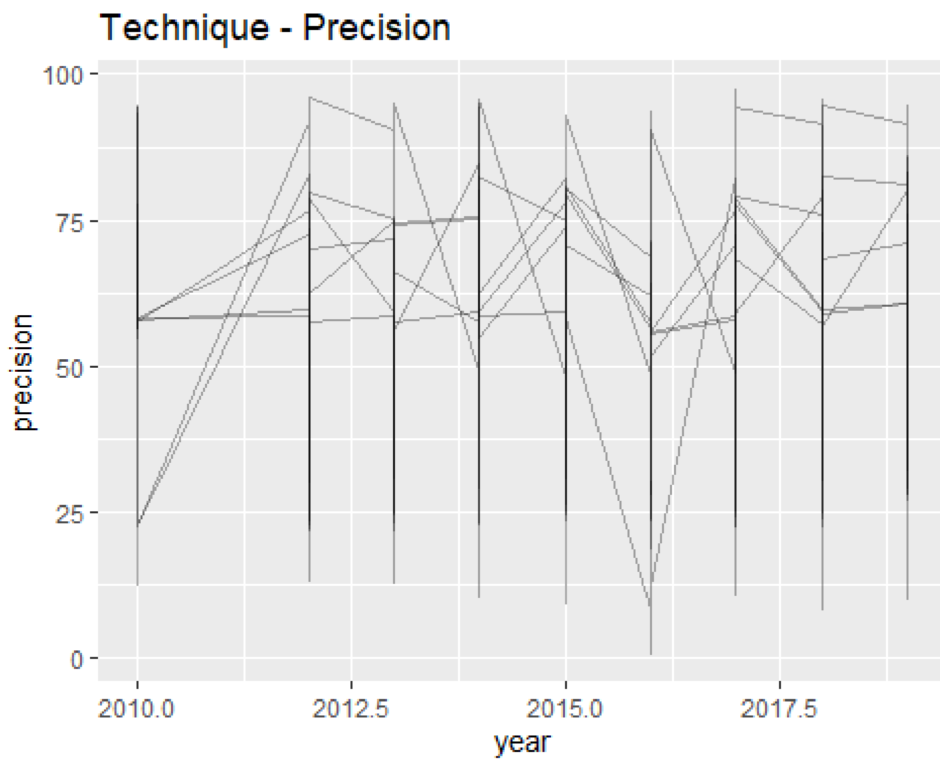

```
foo %>%
  ggplot(aes(year, precision, group = sampling)) +
  geom_line(alpha = 1/3) +
  ggtitle("Sampling - Precision")
```



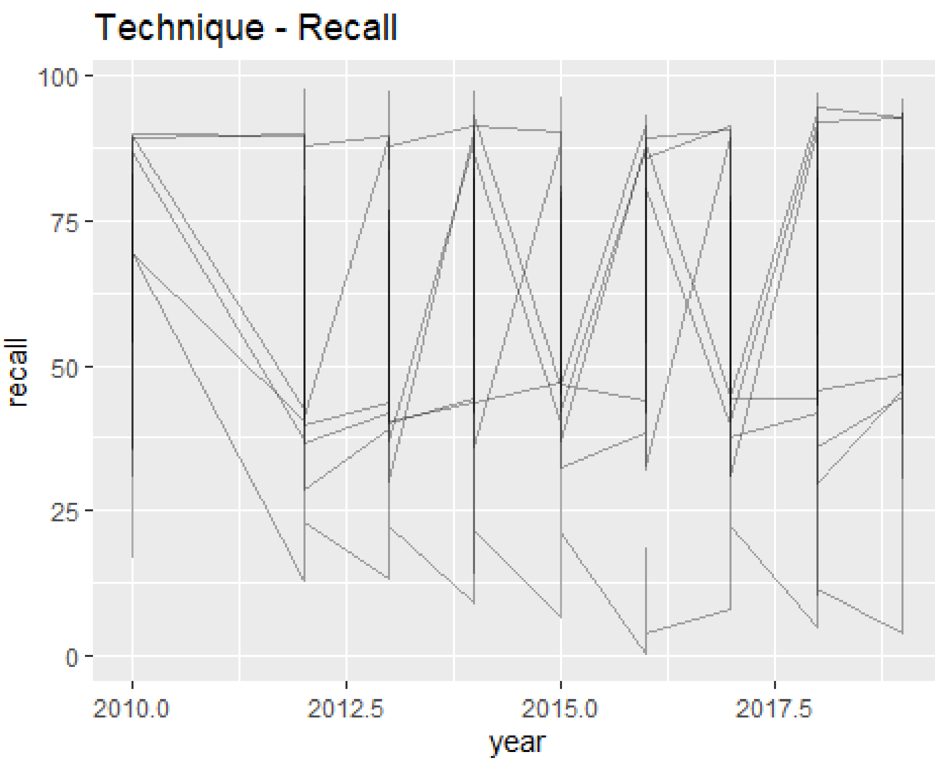

```r
foo %>%
  ggplot(aes(year, recall, group = sampling)) +
  geom_line(alpha = 1/3) +
  ggtitle("Sampling - Recall")
```



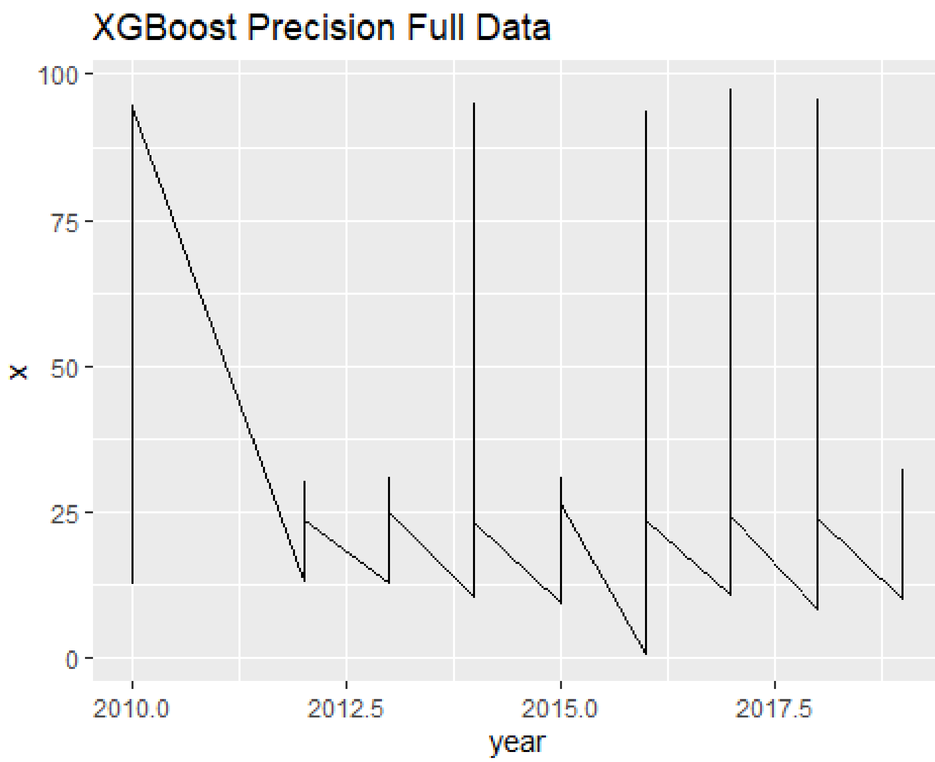

```
foo %>%
  ggplot(aes(year, precision, group = technique)) +
  geom_line(alpha = 1/3) +
  ggtitle("Technique - Precision")
```



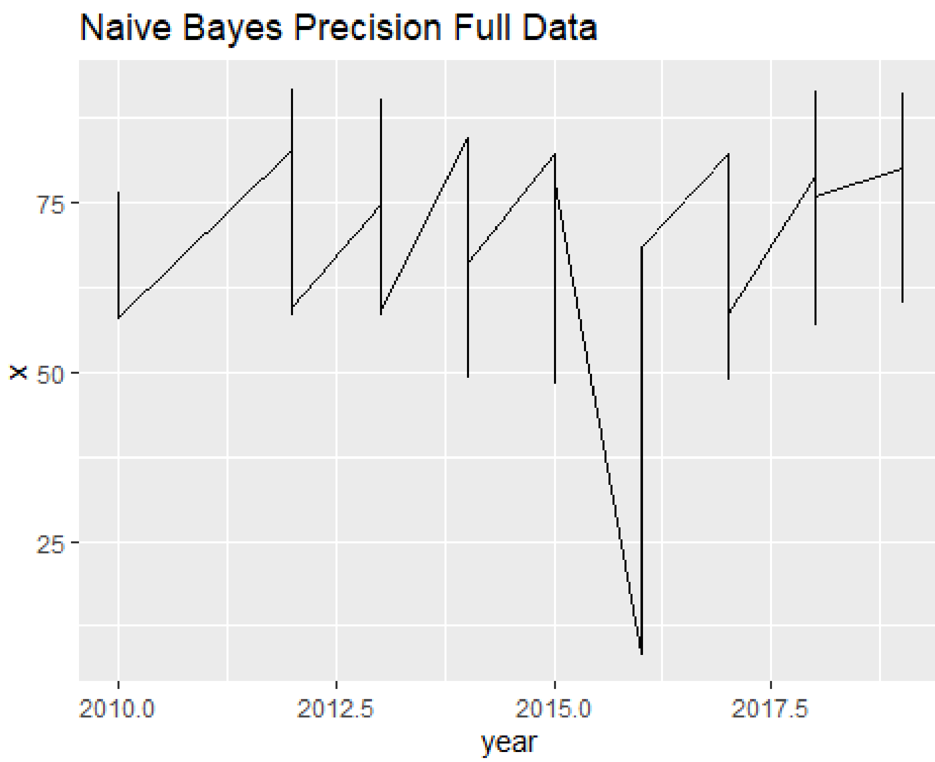

```
foo %>%
  ggplot(aes(year, recall, group = technique)) +
  geom_line(alpha = 1/3) +
  ggtitle("Technique - Recall")
```



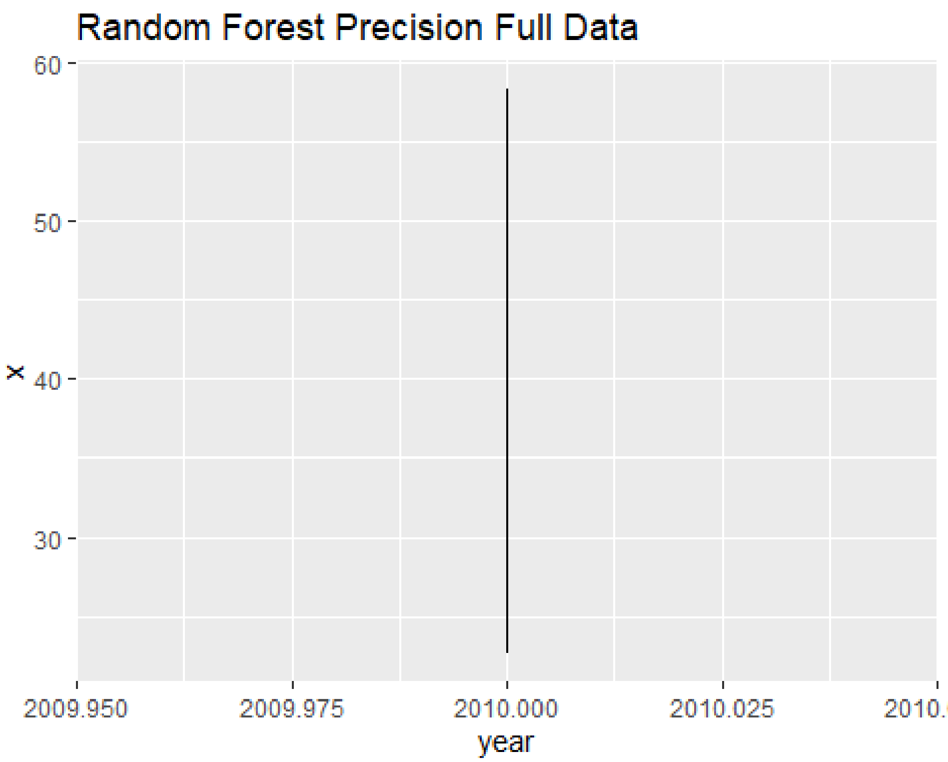

```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
xg <- filter(foo, classifier == "XGBoost")
xg %>%
  ggplot(aes(year, x)) +
  geom_line() +
  ggtitle("XGBoost Precision Full Data ")
```



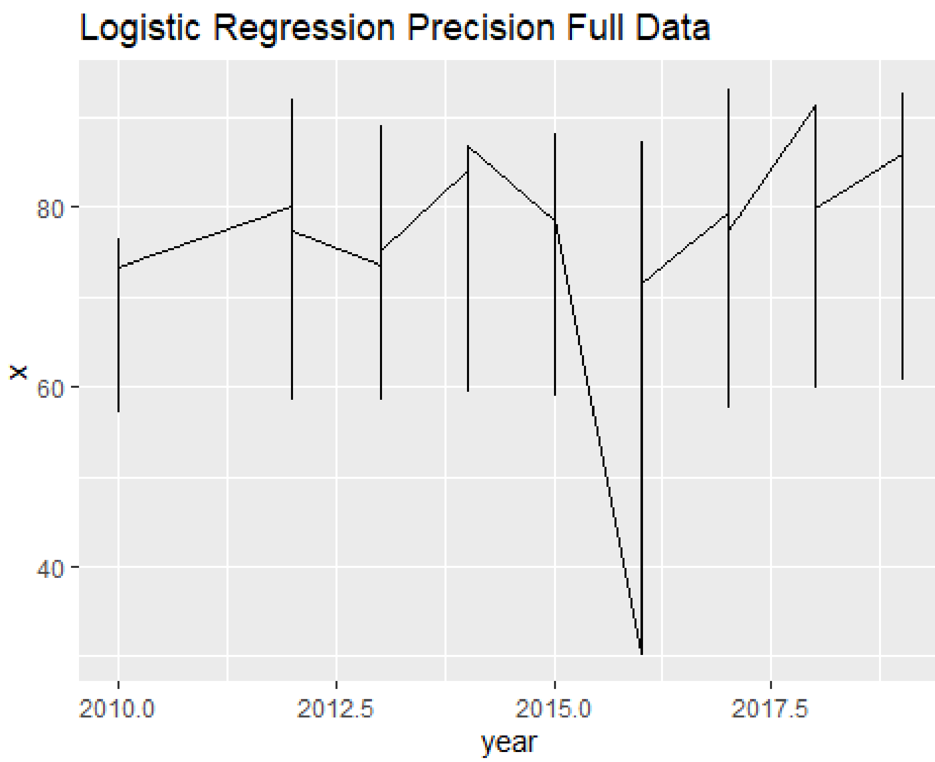

```r
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
nb <- filter(foo, classifier == "Naive Bayes")
nb %>%
  ggplot(aes(year, x)) +
  geom_line() +
  ggtitle("Naive Bayes Precision Full Data ")
```



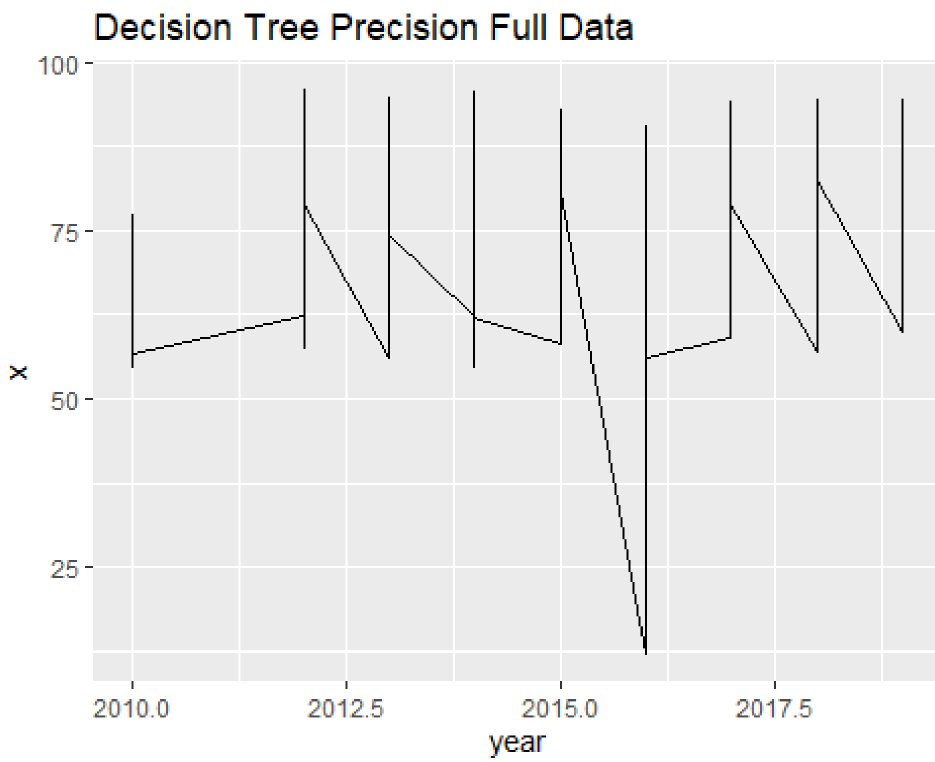

```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
rf <- filter(foo, classifier == "Random Forest")
rf %>%
  ggplot(aes(year, x)) +
  geom_line() +
  ggtitle("Random Forest Precision Full Data ")
```



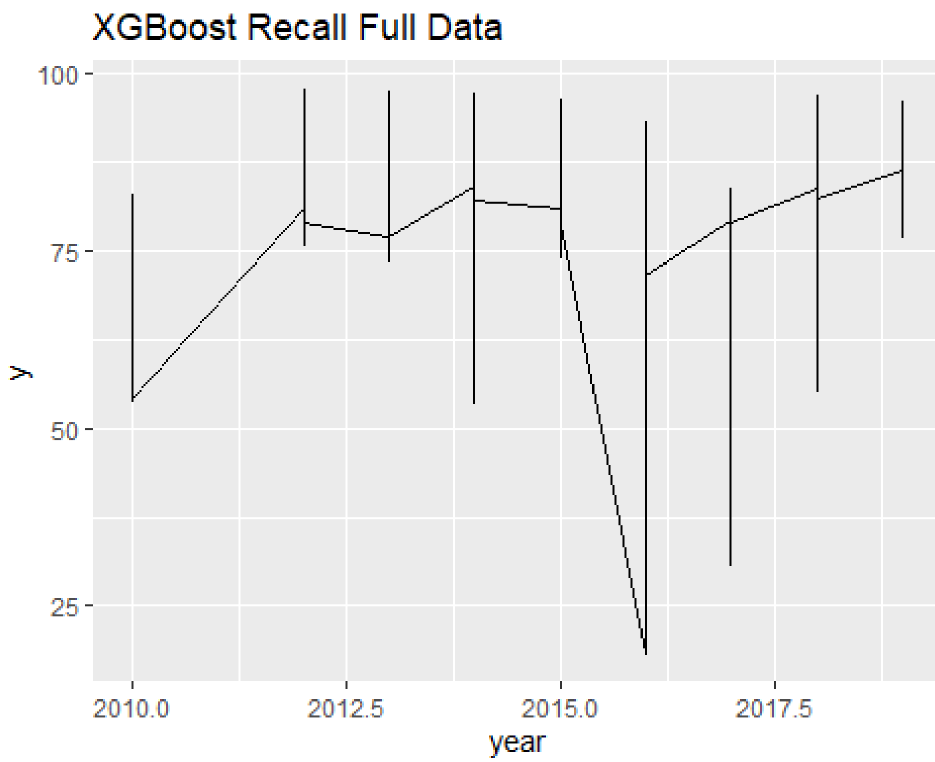

```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
lr <- filter(foo, classifier == "Logistic Reg")
lr %>%
  ggplot(aes(year, x)) +
  geom_line() +
  ggtitle("Logistic Regression Precision Full Data ")
```



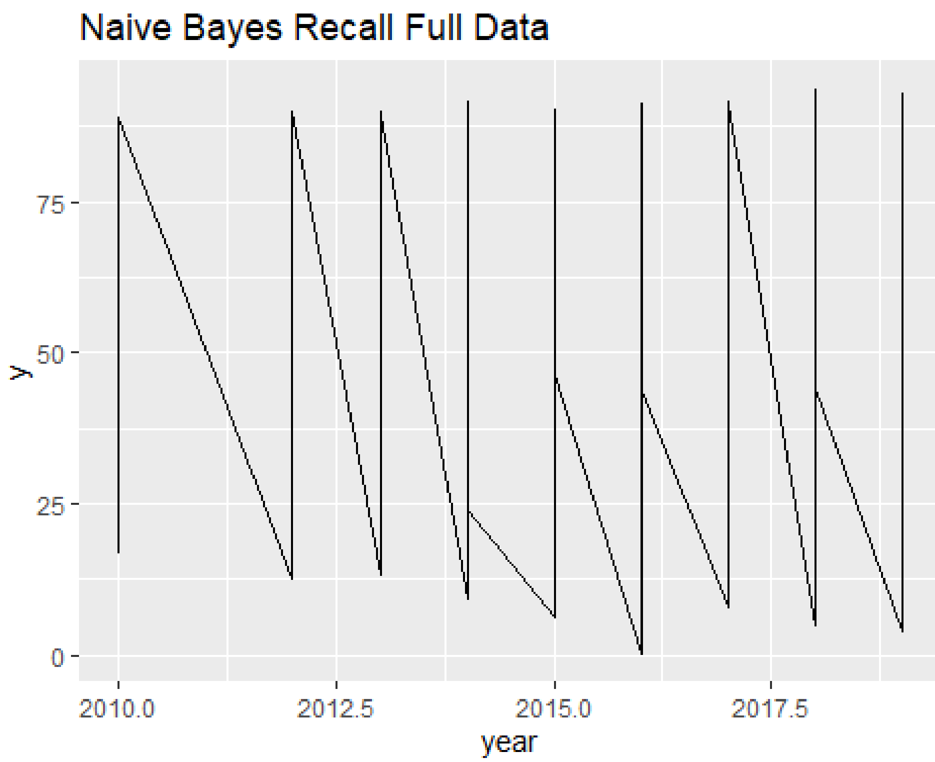

```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
dt <- filter(foo, classifier == "DecisionTree")
dt %>%
  ggplot(aes(year, x)) +
  geom_line() +
  ggtitle("Decision Tree Precision Full Data ")
```



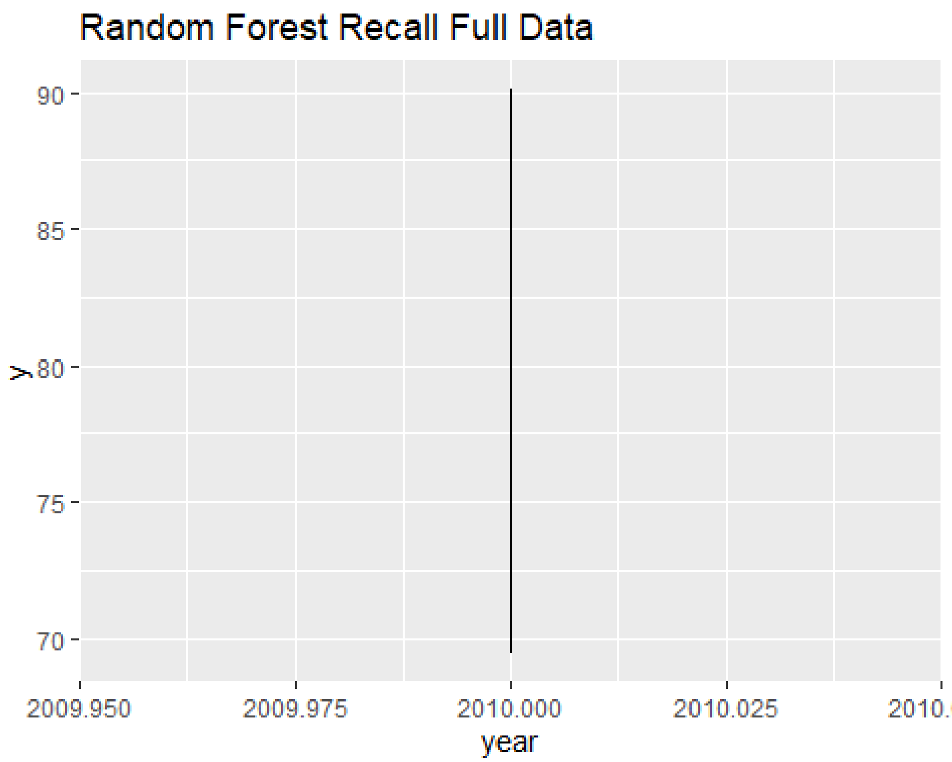

```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
xg <- filter(foo, classifier == "XGBoost")
xg %>%
  ggplot(aes(year, y)) +
  geom_line() +
  ggtitle("XGBoost Recall Full Data ")
```



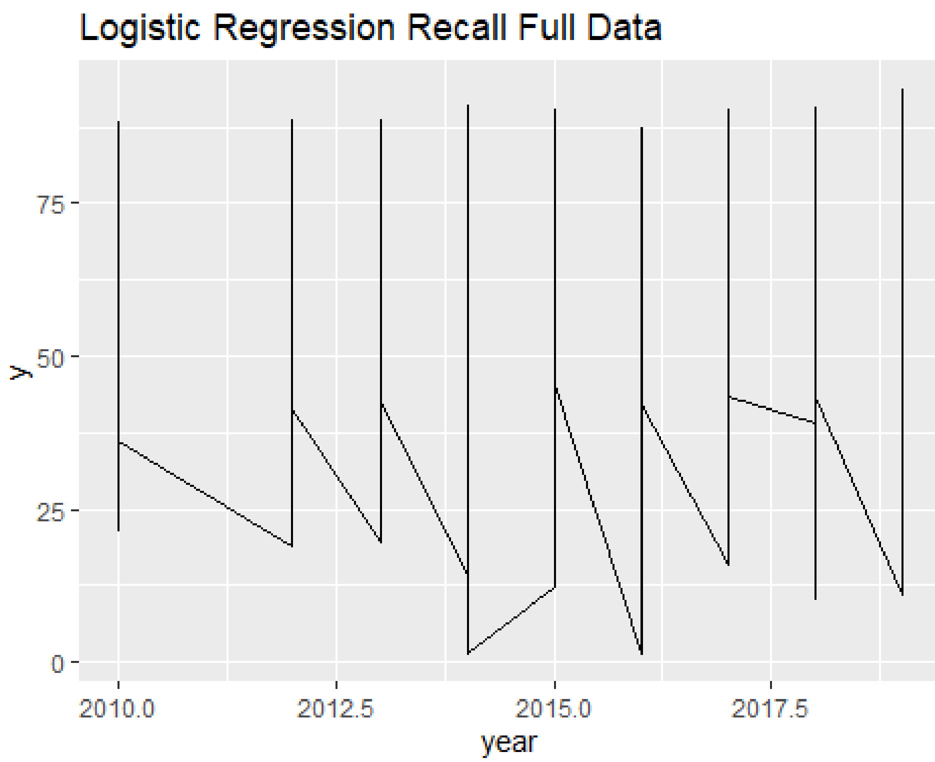

```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
nb <- filter(foo, classifier == "Naive Bayes")
nb %>%
  ggplot(aes(year, y)) +
  geom_line() +
  ggtitle("Naive Bayes Recall Full Data ")
```



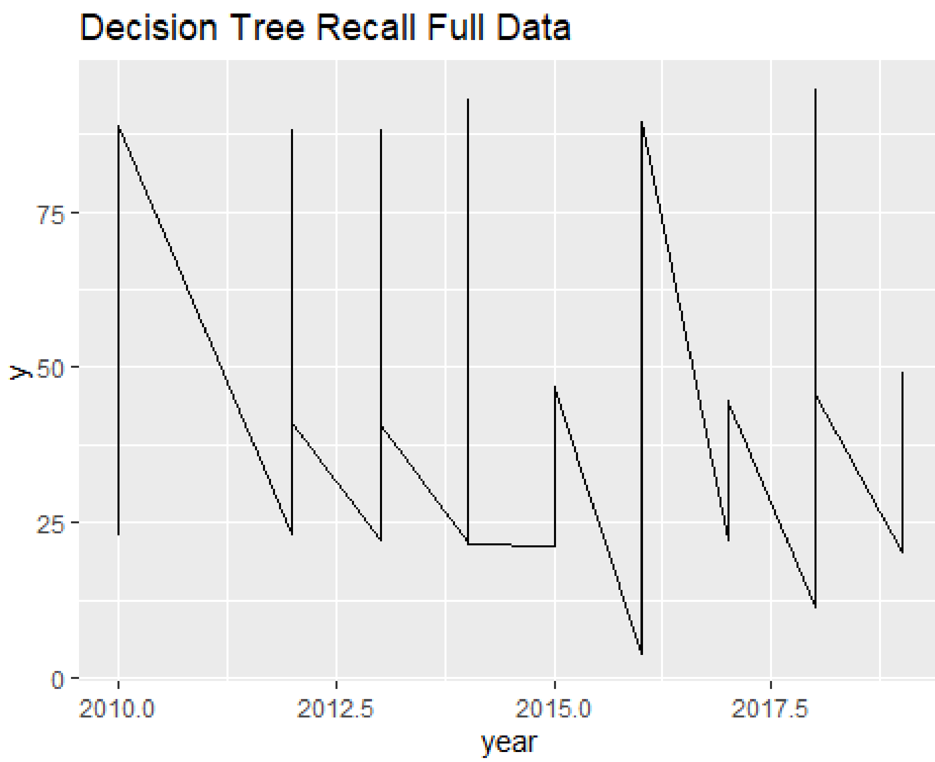

```r
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
rf <- filter(foo, classifier == "Random Forest")
rf %>%
  ggplot(aes(year, y)) +
  geom_line() +
  ggtitle("Random Forest Recall Full Data ")
```



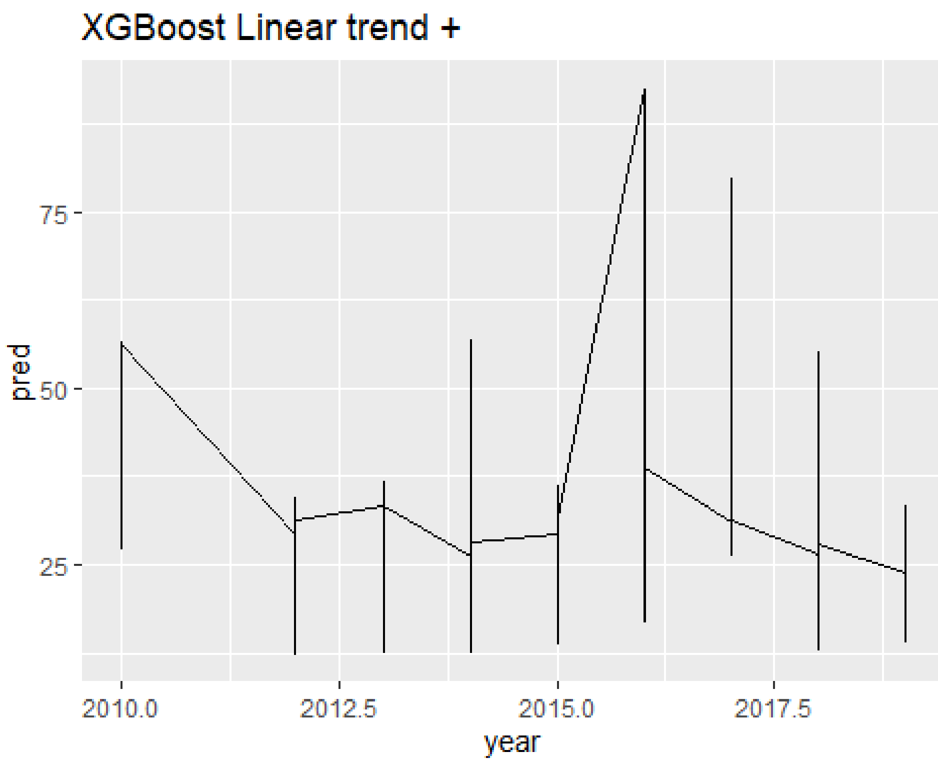

```r
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
lr <- filter(foo, classifier == "Logistic Reg")
lr %>%
  ggplot(aes(year, y)) +
  geom_line() +
  ggtitle("Logistic Regression Recall Full Data ")
```



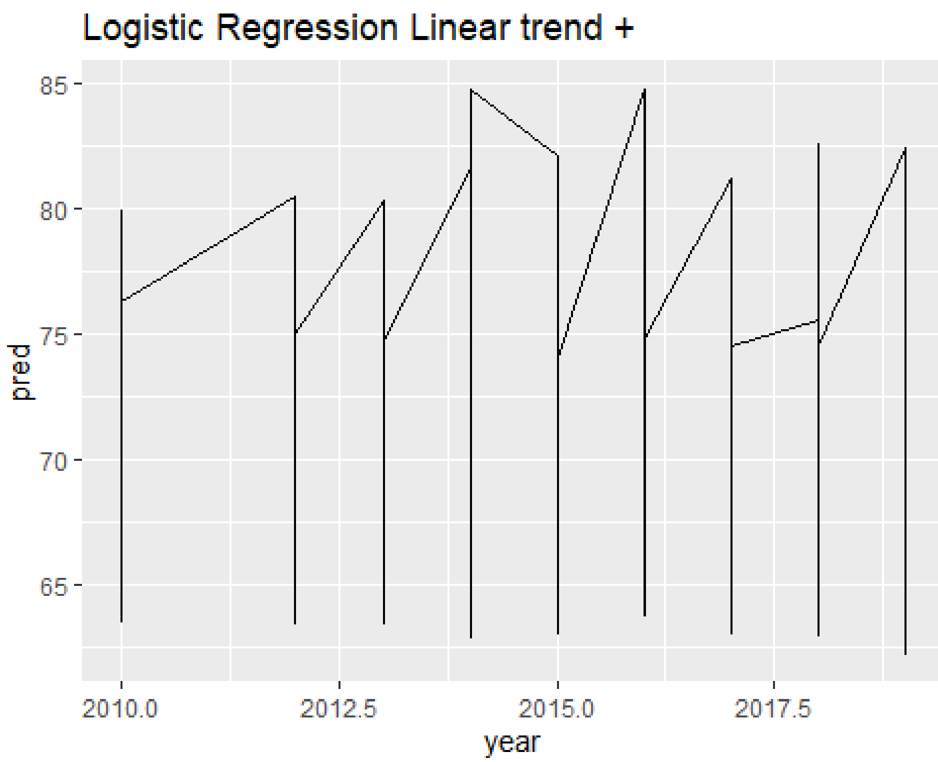

```
# foo = tibble(x=precision, y=recall, classifier=classifier,
sampling=sampling, technique=technique, year=year)
dt <- filter(foo, classifier == "DecisionTree")
dt %>%
  ggplot(aes(year, y)) +
  geom_line() +
  ggtitle("Decision Tree Recall Full Data ")
```



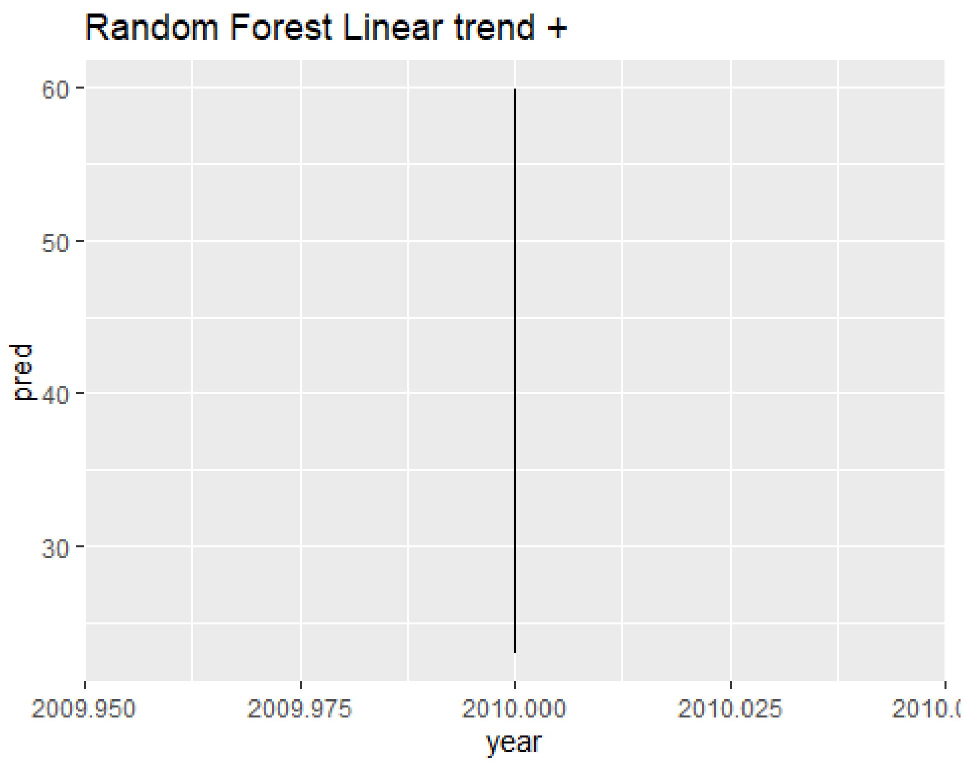

```
xg_mod <- lm(x ~ y, classifier == "XGBoost", data = foo)
xg %>%
  add_predictions(xg_mod) %>%
  ggplot(aes(year, pred)) +
  geom_line() +
  ggtitle("XGBoost Linear trend + ")
```



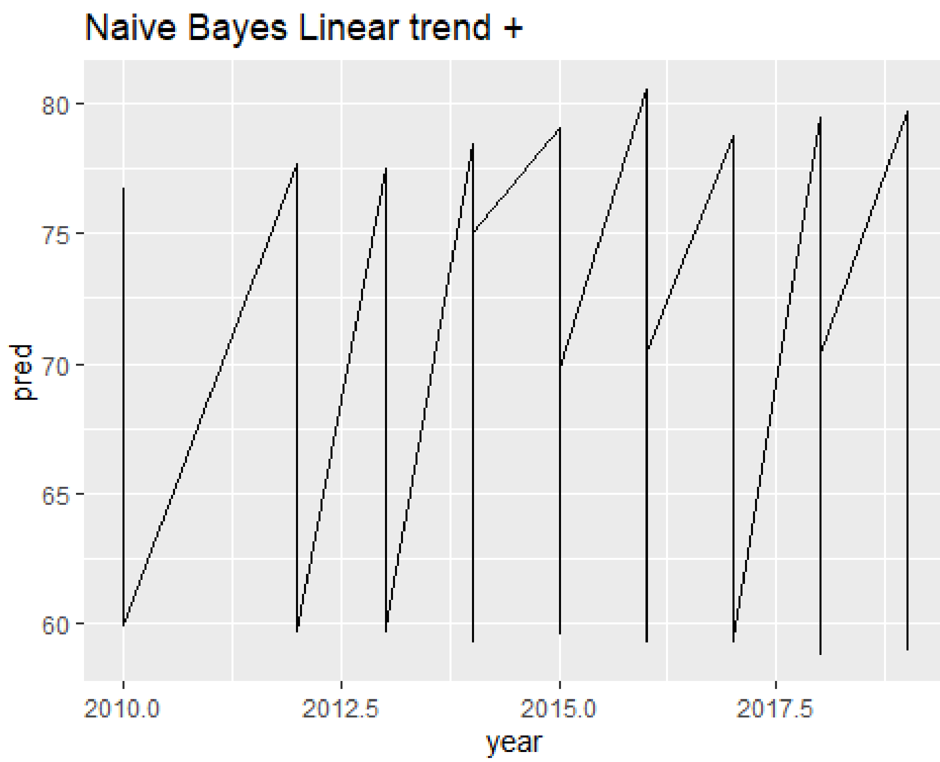

```
lr_mod <- lm(x ~ y,  classifier == "Logistic Reg", data = foo)
lr %>%
  add_predictions(lr_mod) %>%
  ggplot(aes(year, pred)) +
  geom_line() +
  ggtitle("Logistic Regression Linear trend + ")
```



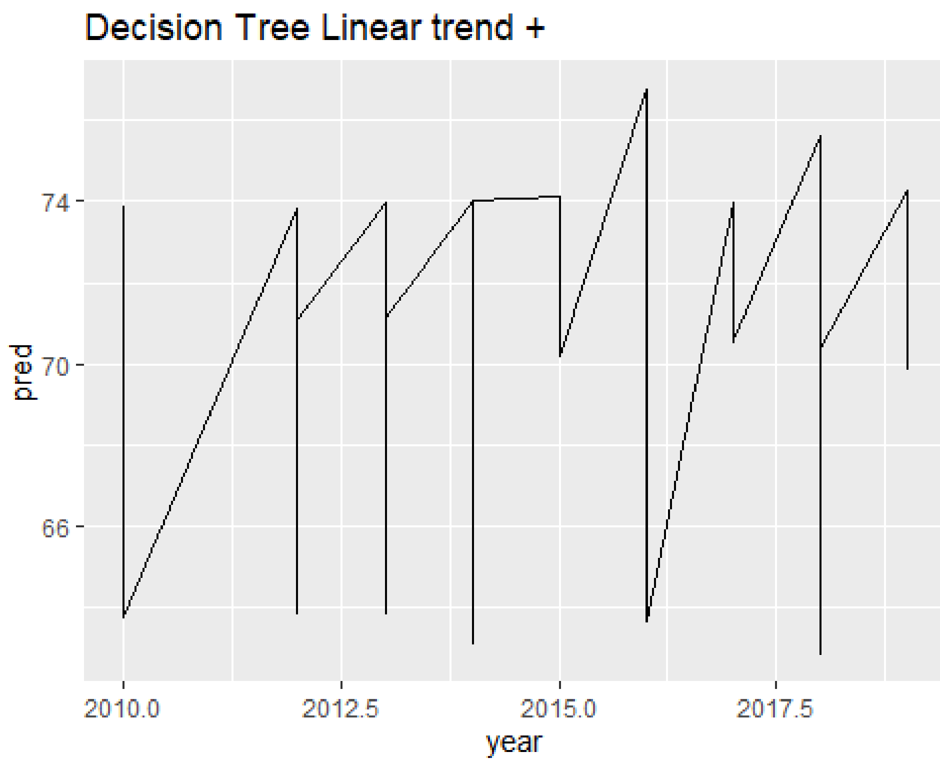

```r
rf_mod <- lm(x ~ y, classifier == "Random Forest", data = foo)
rf %>%
  add_predictions(rf_mod) %>%
  ggplot(aes(year, pred)) +
  geom_line() +
  ggtitle("Random Forest Linear trend + ")
```



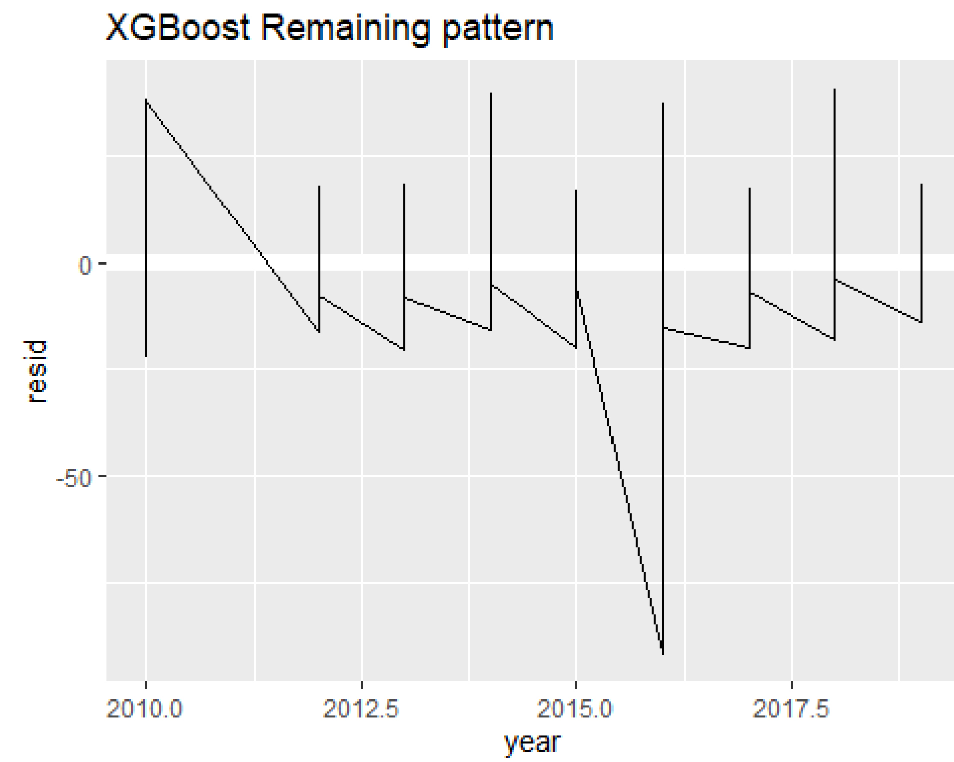

```r
nb_mod <- lm(x ~ y,  classifier == "Naive Bayes", data = foo)
nb %>%
  add_predictions(nb_mod) %>%
  ggplot(aes(year, pred)) +
  geom_line() +
  ggtitle("Naive Bayes Linear trend + ")
```



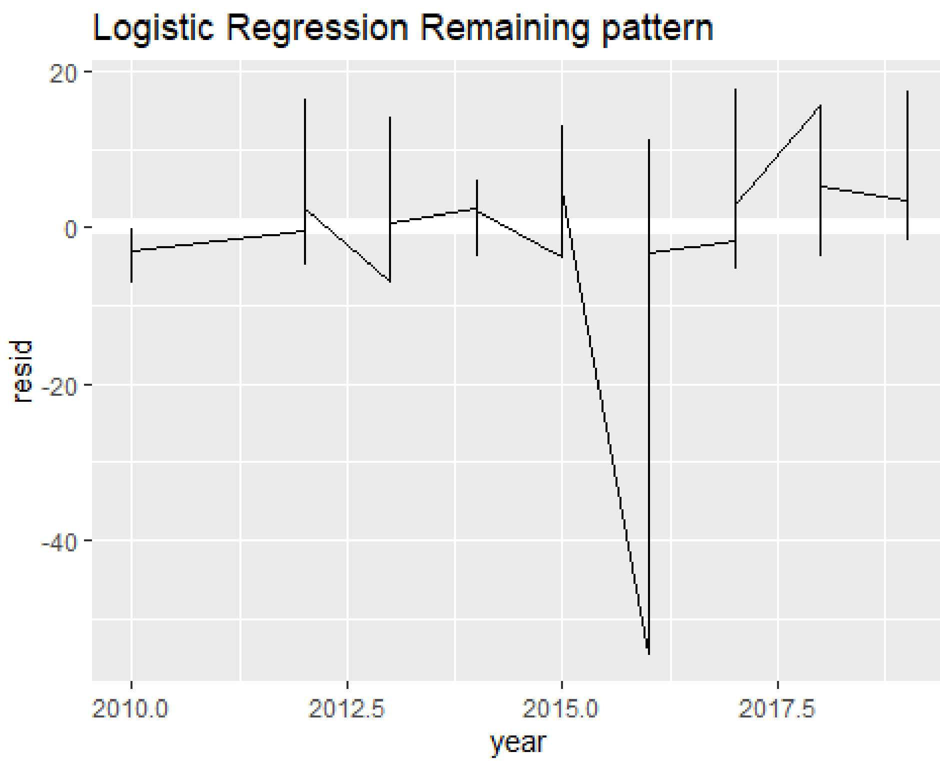

```
dt_mod <- lm(x ~ y, classifier == "DecisionTree", data = foo)
dt %>%
  add_predictions(dt_mod) %>%
  ggplot(aes(year, pred)) +
  geom_line() +
  ggtitle("Decision Tree Linear trend + ")
```



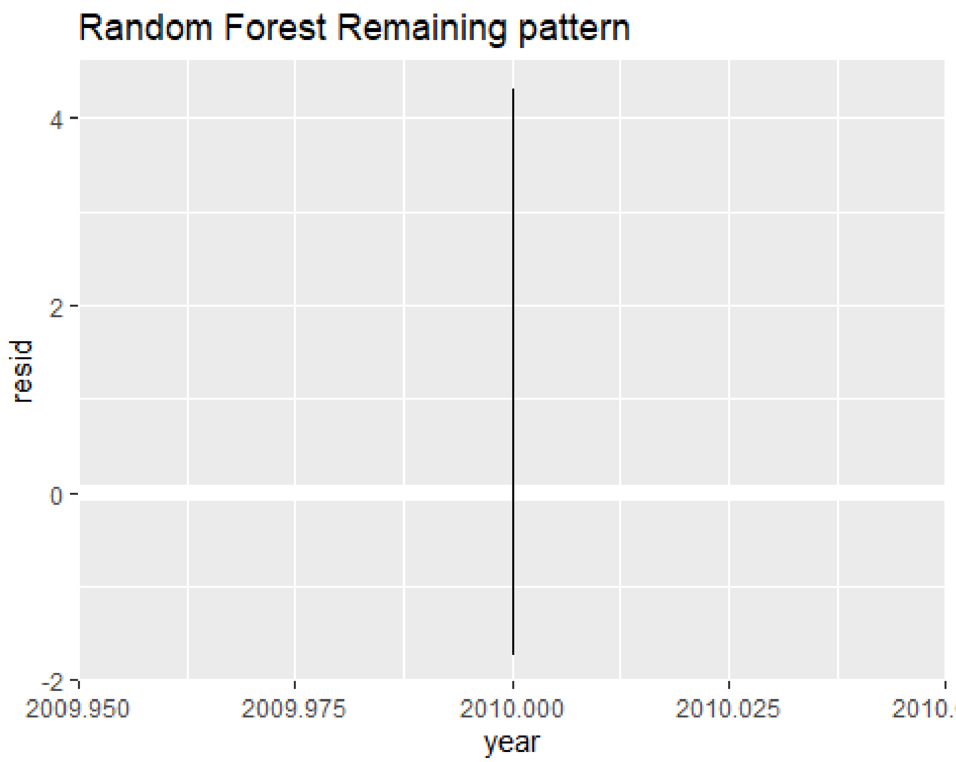

```r
xg %>%
  add_residuals(xg_mod) %>%
  ggplot(aes(year, resid)) +
  geom_hline(yintercept = 0, colour = "white", size = 3) +
  geom_line() +
  ggtitle("XGBoost Remaining pattern")
```



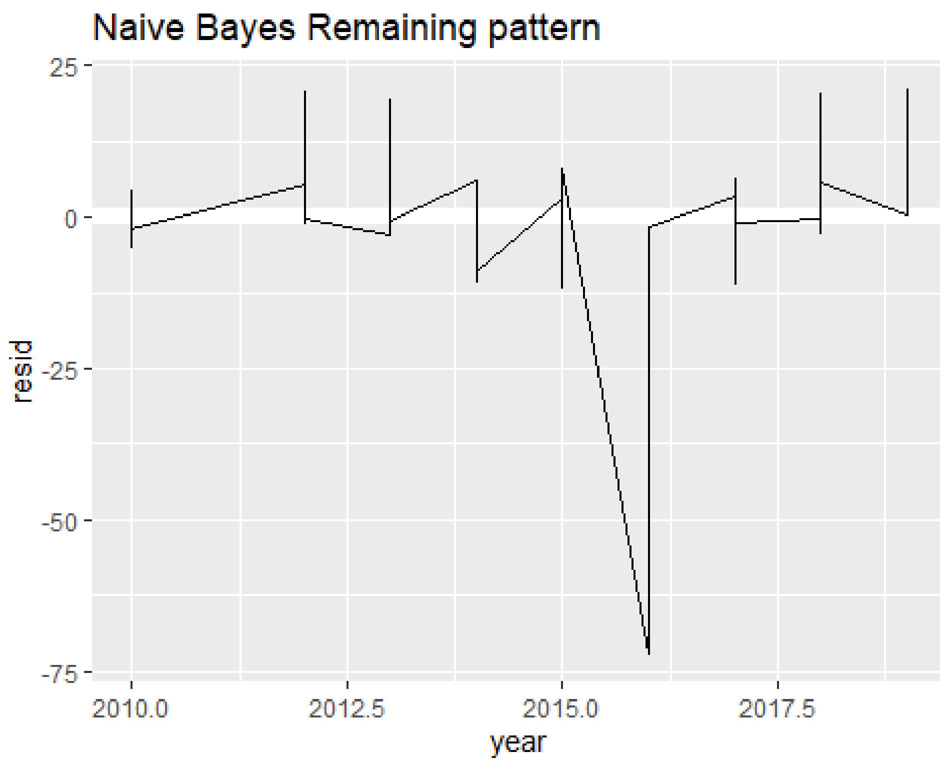

```r
lr %>%
  add_residuals(lr_mod) %>%
  ggplot(aes(year, resid)) +
  geom_hline(yintercept = 0, colour = "white", size = 3) +
  geom_line() +
  ggtitle("Logistic Regression Remaining pattern")
```



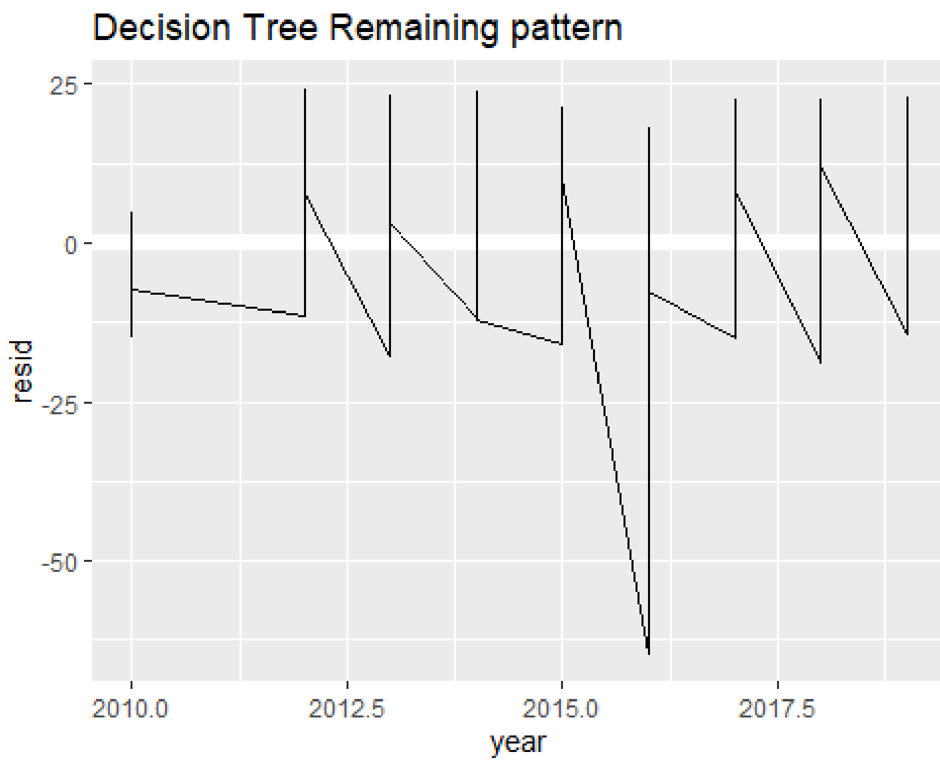

```
rf %>%
  add_residuals(rf_mod) %>%
  ggplot(aes(year, resid)) +
  geom_hline(yintercept = 0, colour = "white", size = 3) +
  geom_line() +
  ggtitle("Random Forest Remaining pattern")
```




```
nb %>%
  add_residuals(nb_mod) %>%
  ggplot(aes(year, resid)) +
  geom_hline(yintercept = 0, colour = "white", size = 3) +
  geom_line() +
  ggtitle("Naive Bayes Remaining pattern")
```




```
dt %>%
  add_residuals(dt_mod) %>%
  ggplot(aes(year, resid)) +
  geom_hline(yintercept = 0, colour = "white", size = 3) +
  geom_line() +
  ggtitle("Decision Tree Remaining pattern")
```




```
by_side <- foo %>%
  group_by(classifier, sampling, technique) %>%
  nest()
by_side
```

```
## # A tibble: 33 × 4
## # Groups:   classifier, sampling, technique [33]
##    classifier    sampling      technique data
##    <chr>         <chr>         <chr>     <list>
##  1 Naive Bayes   Imbalanced    N/A       <tibble [9 × 5]>
##  2 Logistic Reg  Imbalanced    N/A       <tibble [9 × 5]>
##  3 XGBoost       Imbalanced    N/A       <tibble [9 × 5]>
##  4 DecisionTree  Imbalanced    N/A       <tibble [9 × 5]>
##  5 Random Forest Imbalanced    N/A       <tibble [1 × 5]>
##  6 Naive Bayes   Undersampling NearMiss  <tibble [9 × 5]>
##  7 Logistic Reg  Undersampling NearMiss  <tibble [9 × 5]>
##  8 XGBoost       Undersampling NearMiss  <tibble [9 × 5]>
##  9 DecisionTree  Undersampling NearMiss  <tibble [9 × 5]>
## 10 Random Forest Undersampling NearMiss  <tibble [1 × 5]>
## # … with 23 more rows
```

```
by_side$data[[1]]
```

```
## # A tibble: 9 × 5
##         x     y   year lclassifier  resid
```

```
##    <dbl> <dbl> <int>       <dbl>  <dbl>
## 1 74.5  16.7   2010       -2.18    6.99
## 2 83.0  12.7   2012       -0.396  15.4
## 3 74.7  13.3   2013       -5.46    7.20
## 4 84.5   9.13  2014       -2.88   17.0
## 5 82.1   6.44  2015       -7.21   14.6
## 6  8.44  0.25  2016      -63.9   -59.1
## 7 82.3   7.92  2017       -5.62   14.8
## 8 79.1   4.85  2018      -10.9    11.6
## 9 80.1   3.79  2019      -11.2    12.6
```

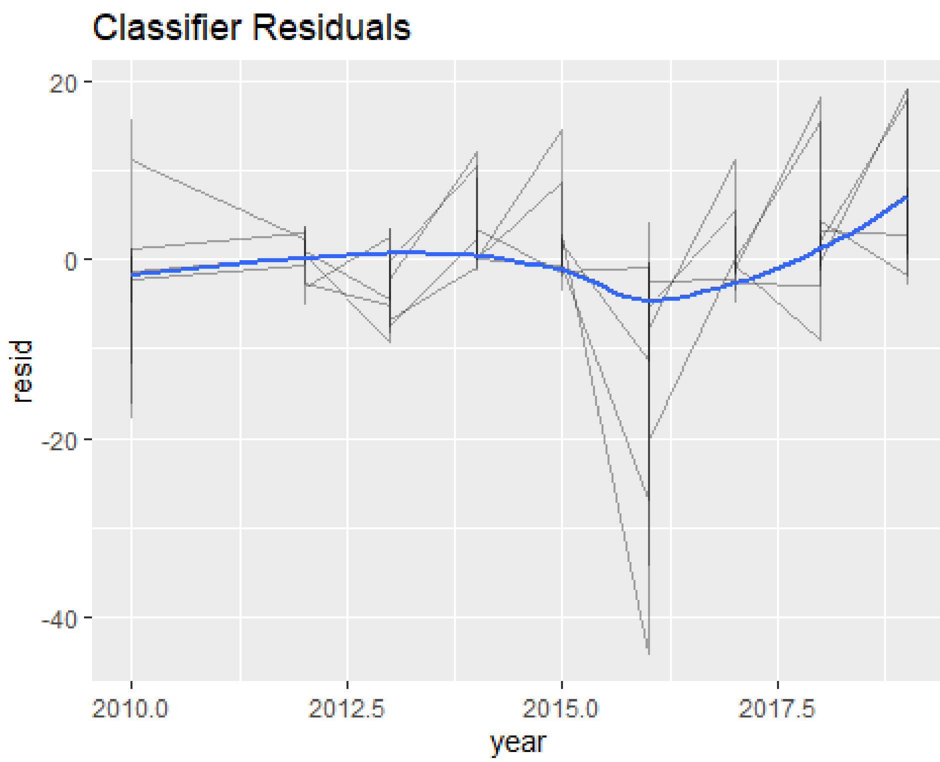
```
foo_model <- function(df){
  lm(x ~ y, data = df)
}
models <- map(by_side$data, foo_model)
by_side <- by_side %>%
  mutate(model = map(data, foo_model))

# by_side %>%
#   filter(classifier == "XGBoost")

by_side %>%
  arrange(classifier, sampling, technique)

## # A tibble: 33 × 5
## # Groups:   classifier, sampling, technique [33]
##    classifier   sampling      technique       data             model
##    <chr>        <chr>         <chr>           <list>           <list>
##  1 DecisionTree FS            Standard Scalar <tibble [1 × 5]> <lm>
##  2 DecisionTree Imbalanced    N/A             <tibble [9 × 5]> <lm>
##  3 DecisionTree Oversampling  ROS             <tibble [9 × 5]> <lm>
##  4 DecisionTree Oversampling  SMOTE           <tibble [9 × 5]> <lm>
##  5 DecisionTree Undersampling NearMiss        <tibble [9 × 5]> <lm>
##  6 DecisionTree Undersampling RUS             <tibble [9 × 5]> <lm>
##  7 DecisionTree Undersampling Tomelinks       <tibble [9 × 5]> <lm>
##  8 Logistic Reg FS            SS & SKB        <tibble [1 × 5]> <lm>
##  9 Logistic Reg Imbalanced    N/A             <tibble [9 × 5]> <lm>
## 10 Logistic Reg Oversampling  ROS             <tibble [9 × 5]> <lm>
## # … with 23 more rows

by_side <- by_side %>%
  mutate(
    resids = map2(data, model, add_residuals)
  )
by_side

## # A tibble: 33 × 6
## # Groups:   classifier, sampling, technique [33]
##    classifier    sampling      technique data             model   resids
```

```
##    <chr>          <chr>         <chr>     <list>            <list> <list>
##  1 Naive Bayes   Imbalanced    N/A       <tibble [9 × 5]> <lm>   <tibble>
##  2 Logistic Reg  Imbalanced    N/A       <tibble [9 × 5]> <lm>   <tibble>
##  3 XGBoost       Imbalanced    N/A       <tibble [9 × 5]> <lm>   <tibble>
##  4 DecisionTree  Imbalanced    N/A       <tibble [9 × 5]> <lm>   <tibble>
##  5 Random Forest Imbalanced    N/A       <tibble [1 × 5]> <lm>   <tibble>
##  6 Naive Bayes   Undersampling NearMiss  <tibble [9 × 5]> <lm>   <tibble>
##  7 Logistic Reg  Undersampling NearMiss  <tibble [9 × 5]> <lm>   <tibble>
##  8 XGBoost       Undersampling NearMiss  <tibble [9 × 5]> <lm>   <tibble>
##  9 DecisionTree  Undersampling NearMiss  <tibble [9 × 5]> <lm>   <tibble>
## 10 Random Forest Undersampling NearMiss  <tibble [1 × 5]> <lm>   <tibble>
## # … with 23 more rows
```

```
resids <- unnest(by_side, resids)
resids
```

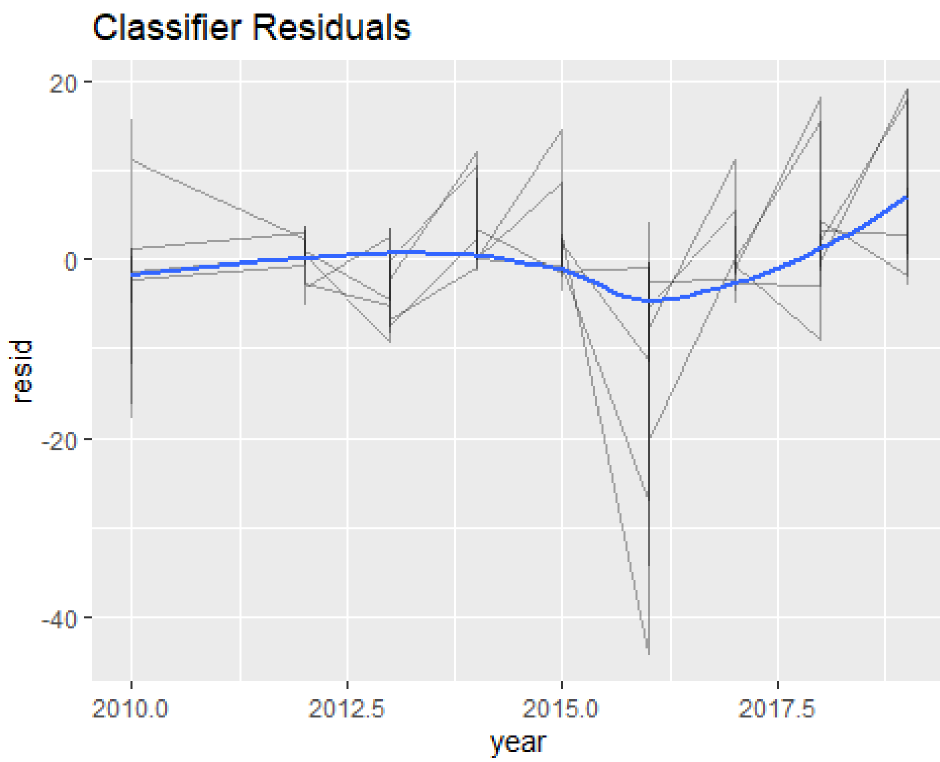
```
## # A tibble: 225 × 10
## # Groups:   classifier, sampling, technique [33]
##    classifier   sampling  technique data     model     x     y  year
lclassifier
##    <chr>        <chr>     <chr>     <list>   <lis> <dbl> <dbl> <int>
<dbl>
##  1 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   74.5  16.7   2010
-2.18
##  2 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   83.0  12.7   2012
-0.396
##  3 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   74.7  13.3   2013
-5.46
##  4 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   84.5   9.13  2014
-2.88
##  5 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   82.1   6.44  2015
-7.21
##  6 Naive Bayes  Imbalanc… N/A       <tibble> <lm>    8.44  0.25  2016      -
63.9
##  7 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   82.3   7.92  2017
-5.62
##  8 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   79.1   4.85  2018      -
10.9
##  9 Naive Bayes  Imbalanc… N/A       <tibble> <lm>   80.1   3.79  2019      -
11.2
## 10 Logistic Reg Imbalanc… N/A       <tibble> <lm>   72.8  21.2   2010
7.68
## # … with 215 more rows, and 1 more variable: resid <dbl>
```

```
resids %>%
  ggplot(aes(year, resid)) +
    geom_line(aes(group = classifier), alpha = 1 / 3) +
    geom_smooth(se = FALSE) +
    ggtitle("Classifier Residuals")
```



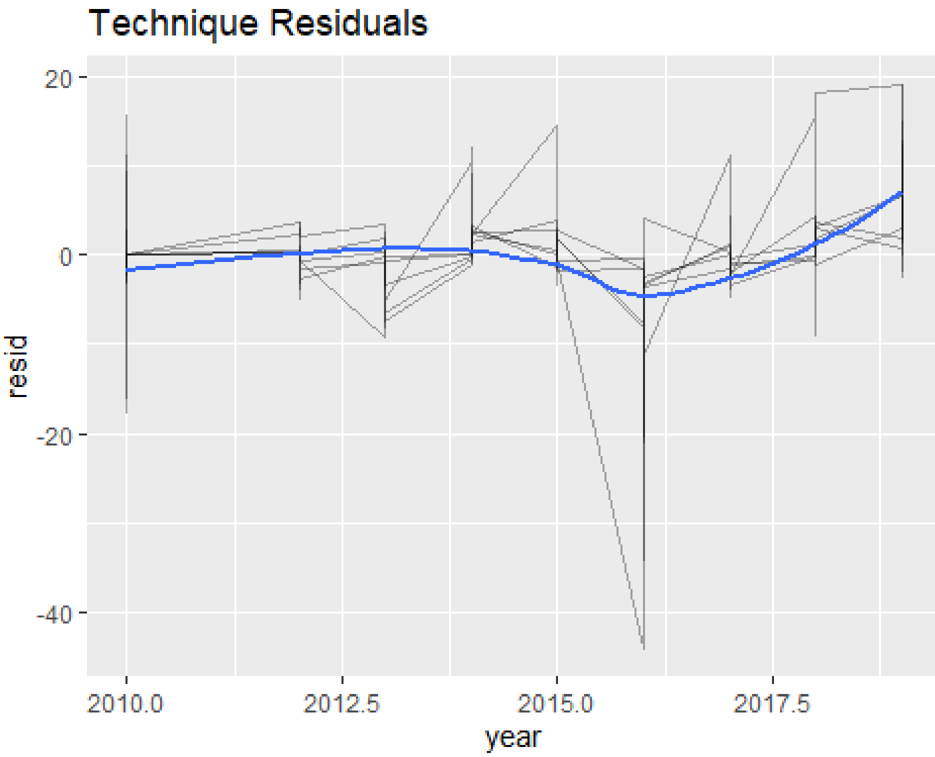

```
#> `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
resids %>%
  ggplot(aes(year, resid)) +
    geom_line(aes(group = classifier), alpha = 1 / 3) +
    geom_smooth(se = FALSE) +
    ggtitle("Classifier Residuals")
```



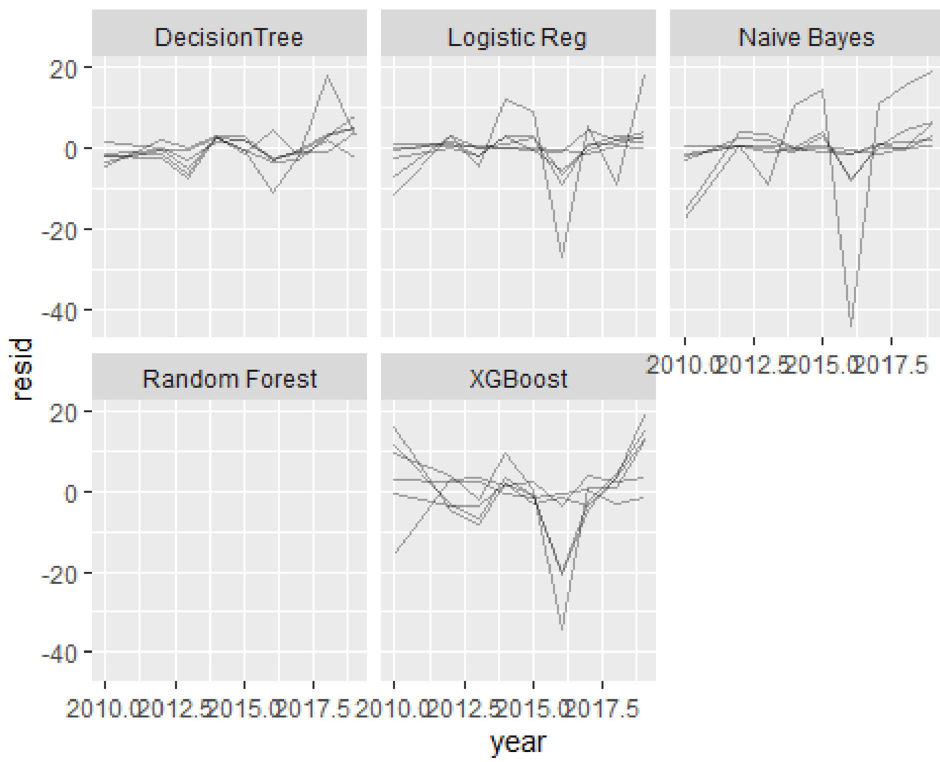

```
#> `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
resids %>%
  ggplot(aes(year, resid)) +
    geom_line(aes(group = technique), alpha = 1 / 3) +
    geom_smooth(se = FALSE) +
    ggtitle("Technique Residuals")
```

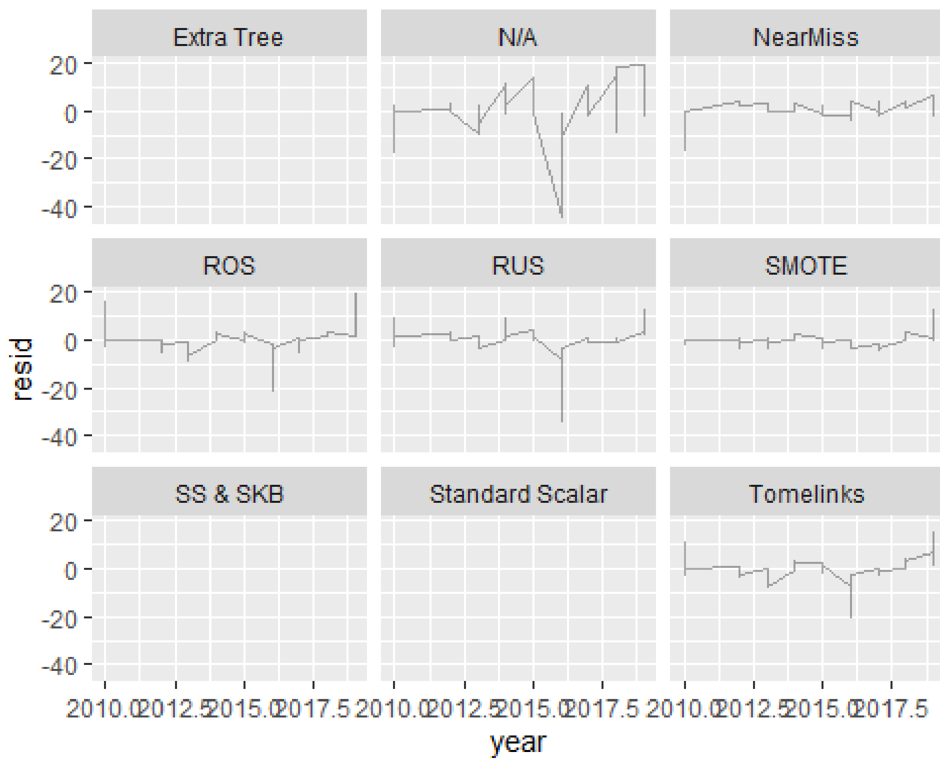



```
#> `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

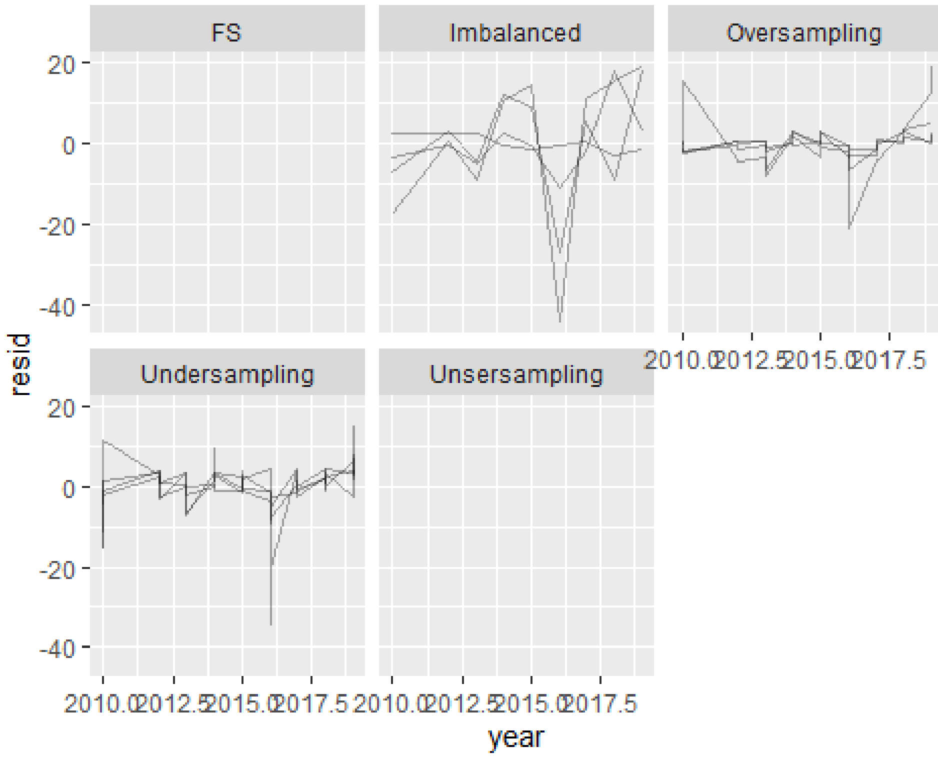
```
resids %>%
  ggplot(aes(year, resid, group = technique)) +
    geom_line(alpha = 1 / 3) +
    facet_wrap(~classifier)
```




```r
resids %>%
  ggplot(aes(year, resid, group = sampling)) +
    geom_line(alpha = 1 / 3) +
    facet_wrap(~technique)
```




```
resids %>%
  ggplot(aes(year, resid, group = classifier)) +
    geom_line(alpha = 1 / 3) +
    facet_wrap(~sampling)
```




```
broom::glance(xg_mod)

## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic    p.value    df logLik   AIC
BIC
##       <dbl>         <dbl> <dbl>     <dbl>      <dbl> <dbl>  <dbl> <dbl>
<dbl>
## 1     0.329         0.316  23.2      25.5 0.00000588     1  -245.  497.
503.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

broom::glance(nb_mod)

## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC
BIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl>
<dbl>
## 1     0.259         0.245  12.4      18.5 0.0000722     1  -216.  437.
443.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

broom::glance(lr_mod)

## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC
BIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl>
```

```
<dbl>
## 1     0.286         0.273  10.2      21.2 0.0000259     1   -205.  416.
422.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
broom::glance(dt_mod)
```

```
## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1    0.0641        0.0464  15.5      3.63  0.0622     1   -228.  461.  467.
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
broom::glance(rf_mod)
```

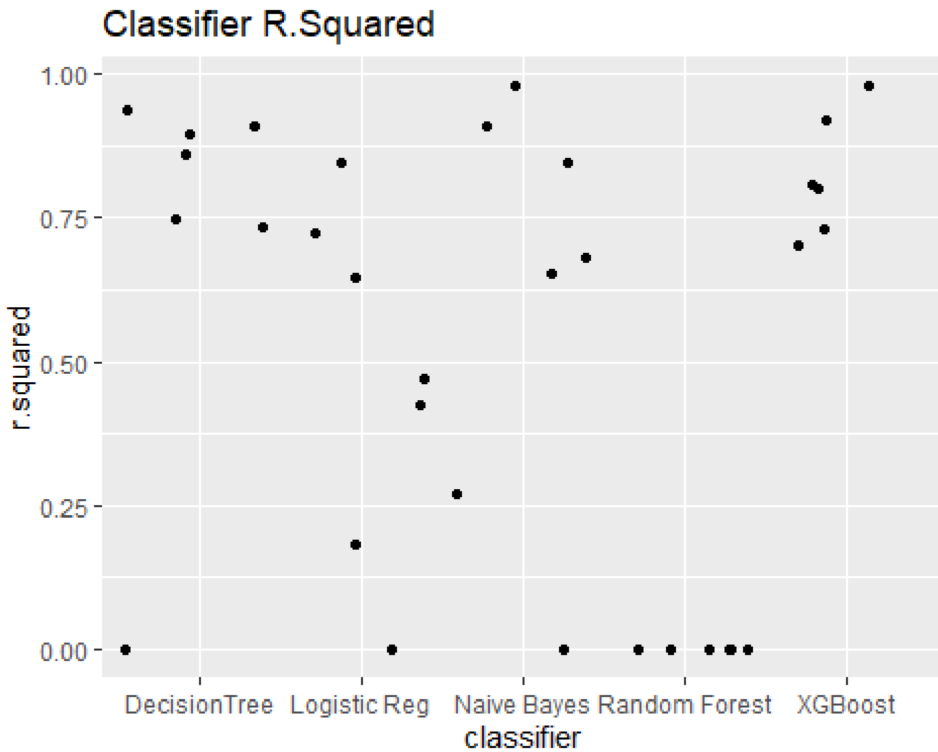
```
## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic   p.value    df logLik   AIC
BIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>  <dbl> <dbl>
<dbl>
## 1     0.986         0.982  2.46      273. 0.0000788     1   -12.7  31.4
30.8
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
by_side %>%
  mutate(glance = map(model, broom::glance)) %>%
  unnest(glance)
```

```
## # A tibble: 33 × 18
## # Groups:   classifier, sampling, technique [33]
##    classifier sampling technique data     model resids   r.squared
adj.r.squared
##    <chr>      <chr>    <chr>     <list>   <lis> <list>       <dbl>
<dbl>
##  1 Naive Bay… Imbalan… N/A       <tibble> <lm>  <tibble>     0.270
0.166
##  2 Logistic … Imbalan… N/A       <tibble> <lm>  <tibble>     0.424
0.341
##  3 XGBoost    Imbalan… N/A       <tibble> <lm>  <tibble>     0.703
0.660
##  4 DecisionT… Imbalan… N/A       <tibble> <lm>  <tibble>     0.746
0.710
##  5 Random Fo… Imbalan… N/A       <tibble> <lm>  <tibble>     0
0
##  6 Naive Bay… Undersa… NearMiss  <tibble> <lm>  <tibble>     0.909
0.896
##  7 Logistic … Undersa… NearMiss  <tibble> <lm>  <tibble>     0.182
0.0649
##  8 XGBoost    Undersa… NearMiss  <tibble> <lm>  <tibble>     0.920
```

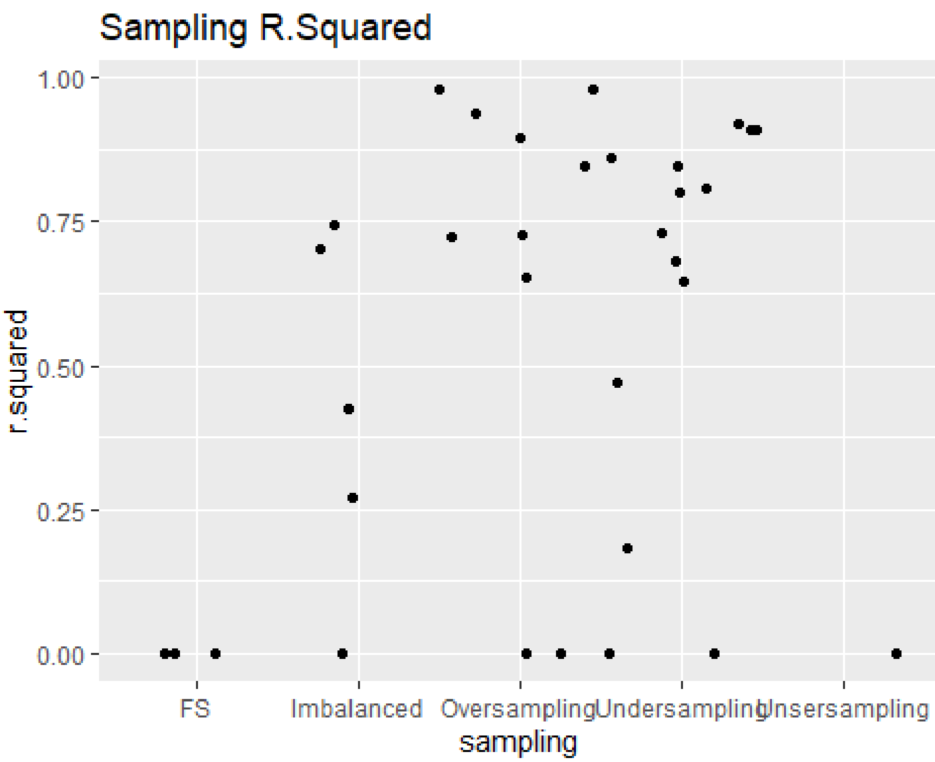
```
0.909
##  9 DecisionT… Undersa… NearMiss  <tibble> <lm>  <tibble>     0.732
0.693
## 10 Random Fo… Undersa… NearMiss  <tibble> <lm>  <tibble>     0
0
## # … with 23 more rows, and 10 more variables: sigma <dbl>, statistic
<dbl>,
## #   p.value <dbl>, df <dbl>, logLik <dbl>, AIC <dbl>, BIC <dbl>,
## #   deviance <dbl>, df.residual <int>, nobs <int>

glance <- by_side %>%
  mutate(glance = map(model, broom::glance)) %>%
  unnest(glance, .drop = TRUE)
# glance
glance %>%
  arrange(r.squared)

## # A tibble: 33 × 18
## # Groups:   classifier, sampling, technique [33]
##    classifier sampling technique data     model resids   r.squared
adj.r.squared
##    <chr>      <chr>    <chr>     <list>   <lis> <list>       <dbl>
<dbl>
##  1 Random Fo… Imbalan… N/A       <tibble> <lm>  <tibble>     0
0
##  2 Random Fo… Undersa… NearMiss  <tibble> <lm>  <tibble>     0
0
##  3 Random Fo… Oversam… SMOTE     <tibble> <lm>  <tibble>     0
0
##  4 Random Fo… Oversam… ROS       <tibble> <lm>  <tibble>     0
0
##  5 Random Fo… Unsersa… RUS       <tibble> <lm>  <tibble>     0
0
##  6 Random Fo… Undersa… Tomelinks <tibble> <lm>  <tibble>     0
0
##  7 DecisionT… FS       Standard… <tibble> <lm>  <tibble>     0
0
##  8 Naive Bay… FS       Extra Tr… <tibble> <lm>  <tibble>     0
0
##  9 Logistic … FS       SS & SKB  <tibble> <lm>  <tibble>     0
0
## 10 Logistic … Undersa… NearMiss  <tibble> <lm>  <tibble>     0.182
0.0649
## # … with 23 more rows, and 10 more variables: sigma <dbl>, statistic
<dbl>,
## #   p.value <dbl>, df <dbl>, logLik <dbl>, AIC <dbl>, BIC <dbl>,
## #   deviance <dbl>, df.residual <int>, nobs <int>
```

```
glance %>%
  ggplot(aes(classifier, r.squared)) +
    geom_jitter(width = 0.5) +
  ggtitle("Classifier R.Squared")
```




```
# glance %>%
#   ggplot(aes(sampling, r.squared)) +
#     geom_jitter(width = 0.5) +
#   ggtitle("Sampling R.Squared")
# Creating error: Validate mapping
# glance %>%
#   ggplot(aes(year, r.squared)) +
#   geom_jitter(width = 0.5) +
#   geom_jitter("Year R.Squared")
```

```
glance %>%
  ggplot(aes(sampling, r.squared)) +
    geom_jitter(width = 0.5) +
  ggtitle("Sampling R.Squared")
```




```
bad_fit <- filter(glance, r.squared < 0.25)
bad_fit

## # A tibble: 10 × 18
## # Groups:   classifier, sampling, technique [10]
##    classifier sampling technique data     model resids   r.squared
adj.r.squared
##    <chr>      <chr>    <chr>     <list>   <lis> <list>       <dbl>
<dbl>
##  1 Random Fo… Imbalan… N/A       <tibble> <lm>  <tibble>     0
0
##  2 Logistic … Undersa… NearMiss  <tibble> <lm>  <tibble>     0.182
0.0649
##  3 Random Fo… Undersa… NearMiss  <tibble> <lm>  <tibble>     0
0
##  4 Random Fo… Oversam… SMOTE     <tibble> <lm>  <tibble>     0
0
##  5 Random Fo… Oversam… ROS       <tibble> <lm>  <tibble>     0
0
##  6 Random Fo… Unsersa… RUS       <tibble> <lm>  <tibble>     0
0
##  7 Random Fo… Undersa… Tomelinks <tibble> <lm>  <tibble>     0
0
##  8 DecisionT… FS       Standard… <tibble> <lm>  <tibble>     0
0
##  9 Naive Bay… FS       Extra Tr… <tibble> <lm>  <tibble>     0
0
```

```
## 10 Logistic … FS       SS & SKB  <tibble> <lm>  <tibble>     0
0
## # … with 10 more variables: sigma <dbl>, statistic <dbl>, p.value <dbl>,
## #   df <dbl>, logLik <dbl>, AIC <dbl>, BIC <dbl>, deviance <dbl>,
## #   df.residual <int>, nobs <int>
```

```
foo %>%
semi_join(bad_fit, by = "classifier") %>%
  ggplot(aes(year, x, colour = classifier)) +
    geom_line() +
    ggplot("Classifier for Precision")
```

```
## Error in `fortify()`:
## ! `data` must be a data frame, or other object coercible by `fortify()`,
not a character vector.
```

```
foo %>%
  semi_join(bad_fit, by = "classifier") %>%
  ggplot(aes(year, y, colour = classifier)) +
    geom_line() +
    ggplot("Classifier for Recall")
```

```
## Error in `fortify()`:
## ! `data` must be a data frame, or other object coercible by `fortify()`,
not a character vector.
```