Preliminary Investigation

Ethan Allavarpu (UID: 405287603)

10/27/2020

```
set.seed(110920)
# sample <- read.csv("sample.csv", stringsAsFactors = TRUE)
# sample
```

```
training <- read.csv("training.csv", stringsAsFactors = TRUE)
sort(abs(cor(training)["class", ]), decreasing = TRUE)[2:19]
```

```
Broad_H4K20me1_percentage   Broad_H3K9ac_percentage            H4K20me1_width
                0.5309561                0.4810581                 0.4789003
       BioGRID_log_degree           H3K79me2_height            H3K79me2_width
                0.4764364                0.4719210                 0.4709266
 Broad_H3K4me2_percentage  Broad_H3K27ac_percentage           H4K20me1_height
                0.4693101                0.4641367                 0.4595187
 Broad_H3K4me1_percentage Broad_H3K79me2_percentage  Broad_H3K4me3_percentage
                0.4580446                0.4571273                 0.4315878
Broad_H3K36me3_percentage             H3K4me1_width            H3K36me3_width
                0.4306293                0.4287817                 0.4275987
              pLOF_Zscore                     N_LOF             H3K4me2_width
                0.4227907                0.4212450                 0.4080466
```

```
training$class <- factor(training$class)
levels(training$class) <- c("NG", "OG", "TSG")
dim(training)
```

```
[1] 3177   99
```

```
names(training)[c(1, 99)]
```

```
[1] "id"    "class"
```

```
barplot(table(training$class))
table(training$class) / nrow(training)
```

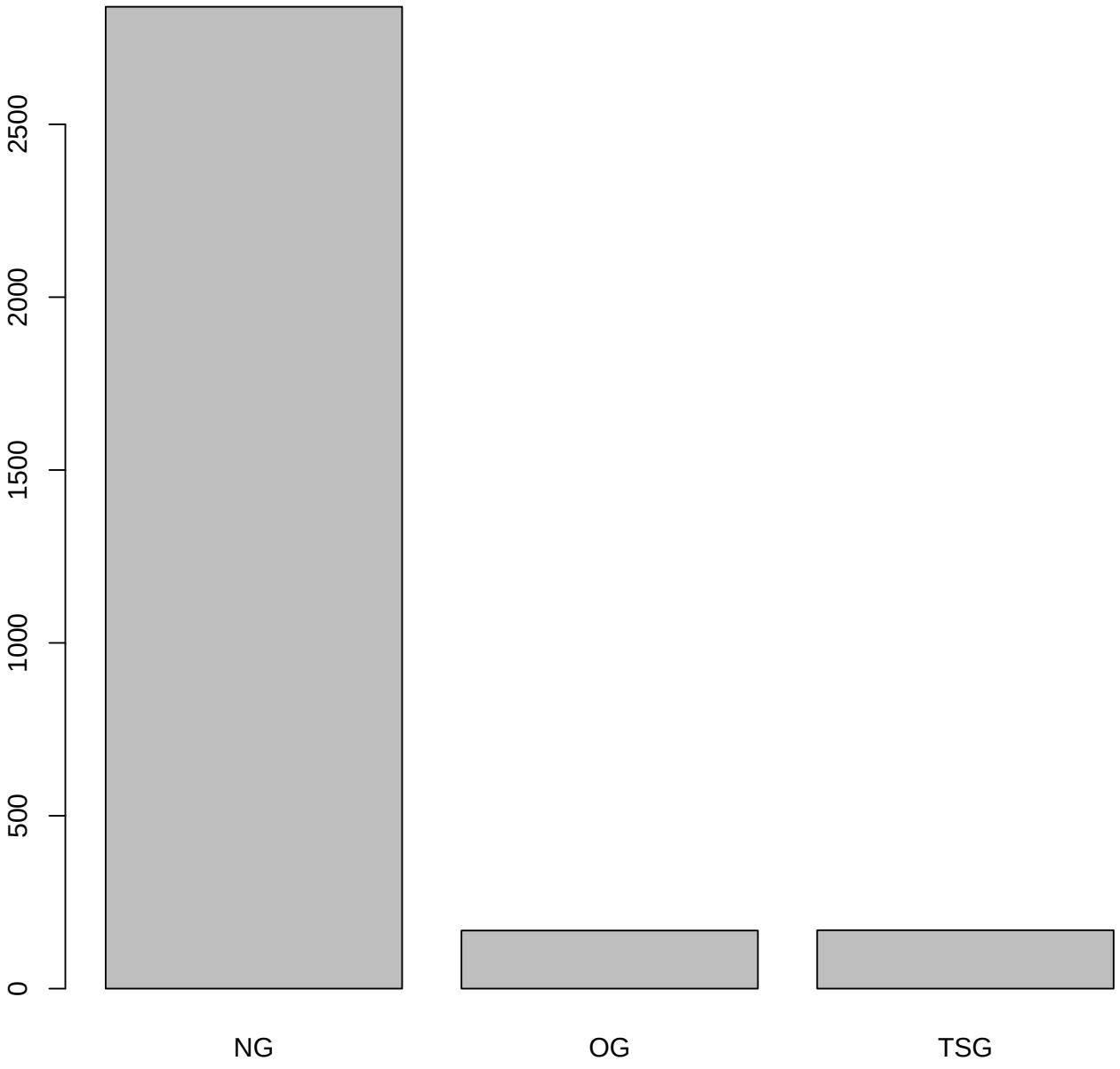


```
        NG         OG        TSG
0.89392509 0.05288008 0.05319484
```

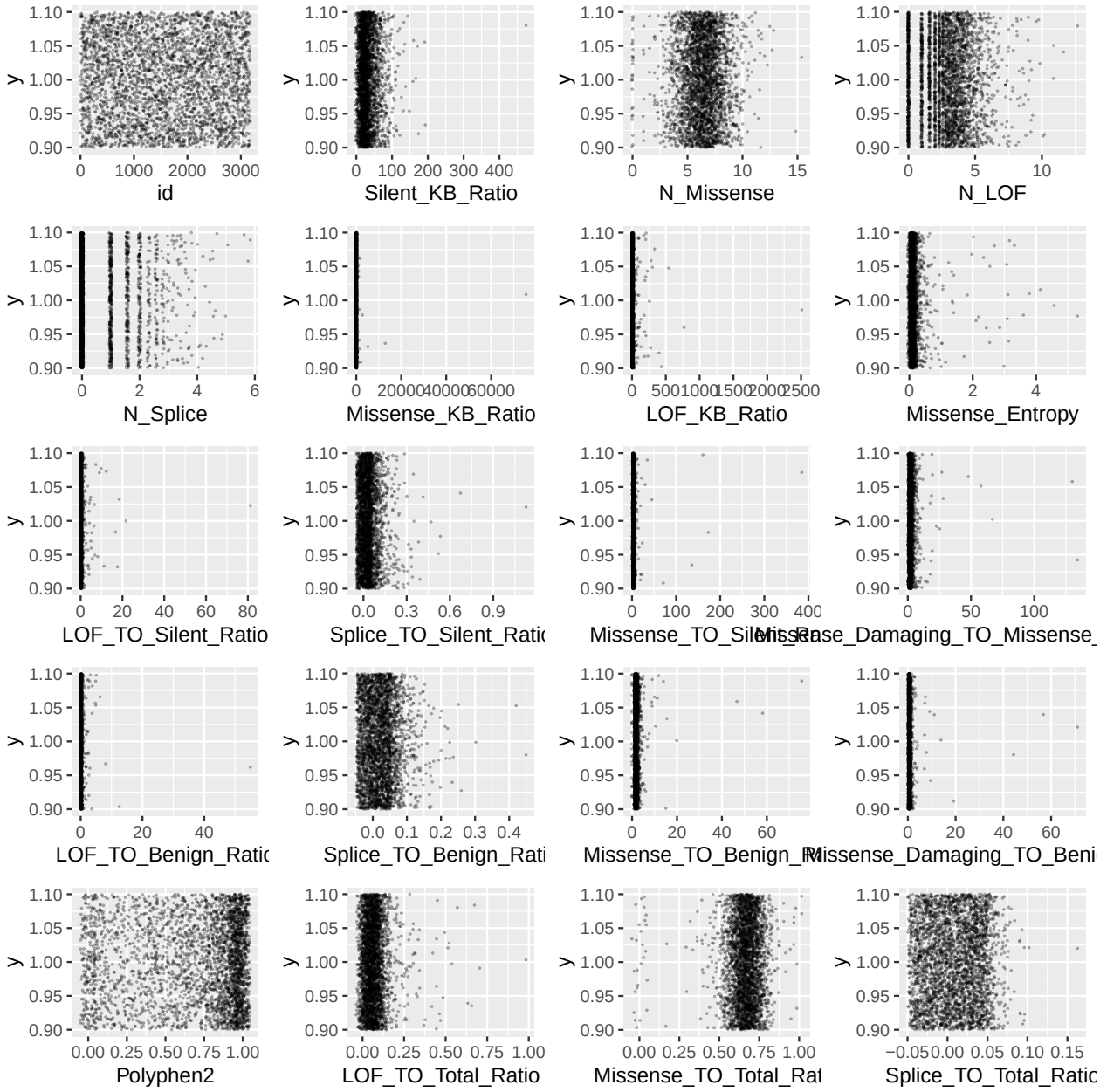
```
any(is.na(training))
```

```
[1] FALSE
```

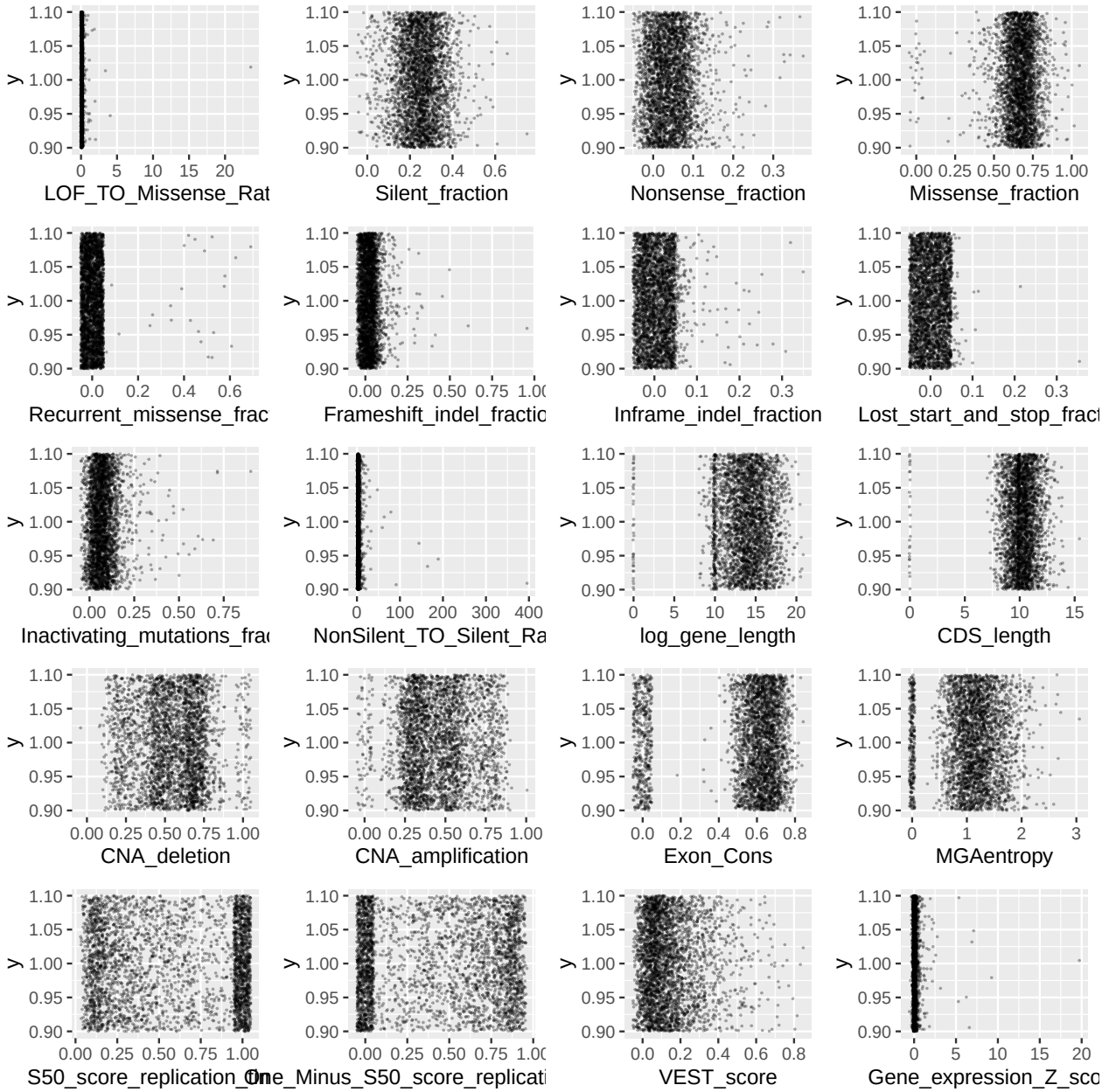
```
library(ggplot2)
```



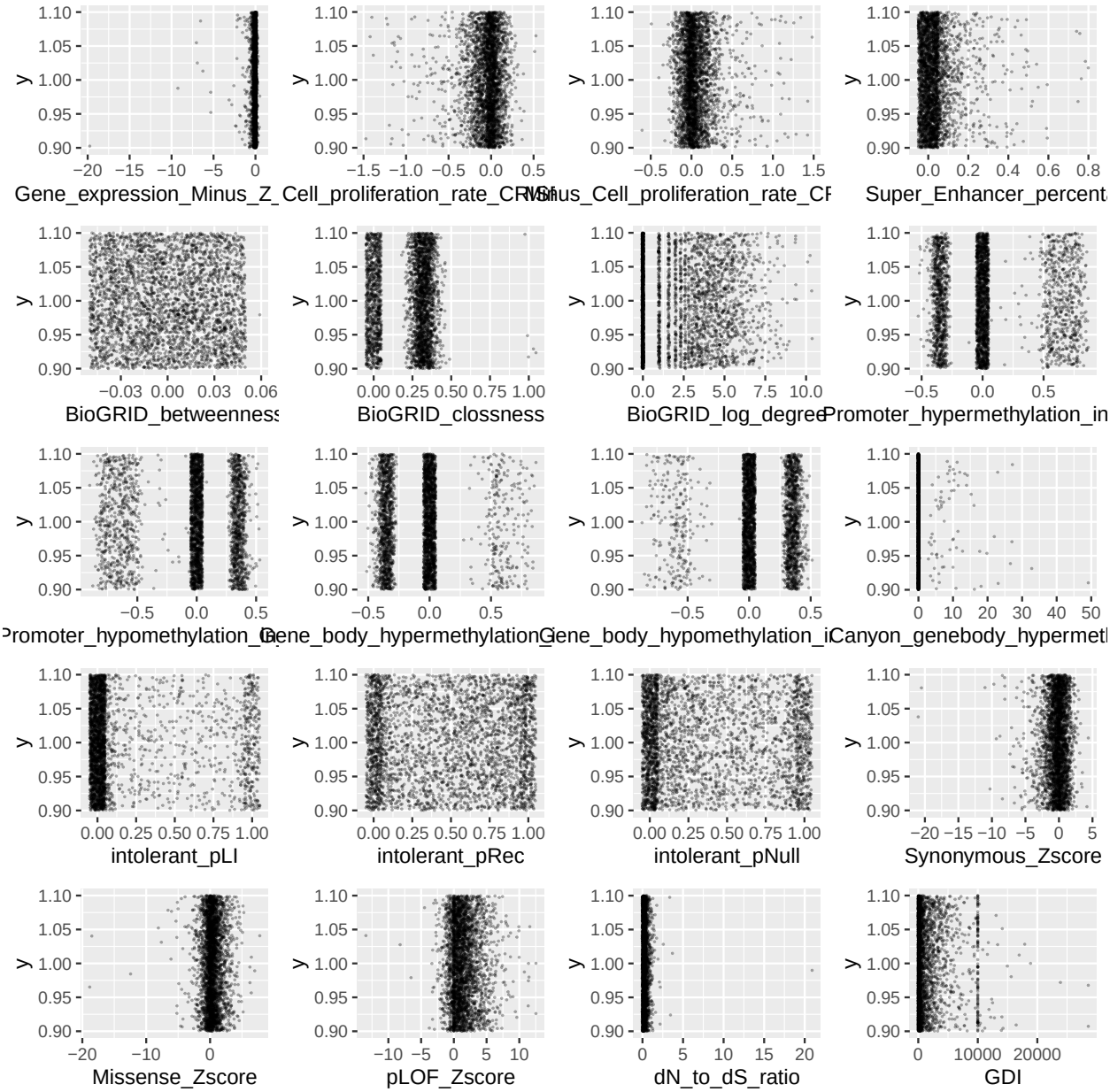

```
scatter <- function(var) {
  ggplot(training, aes_string(var, 1)) +
    geom_jitter(width = 0.05, height = 0.1, size = 0.1,
                colour = rgb(0, 0, 0, alpha = 1 / 3))
}
scat_plot <- lapply(names(training)[-99], scatter)
library(gridExtra)
grid.arrange(grobs = scat_plot[1:20], ncol = 4)
```



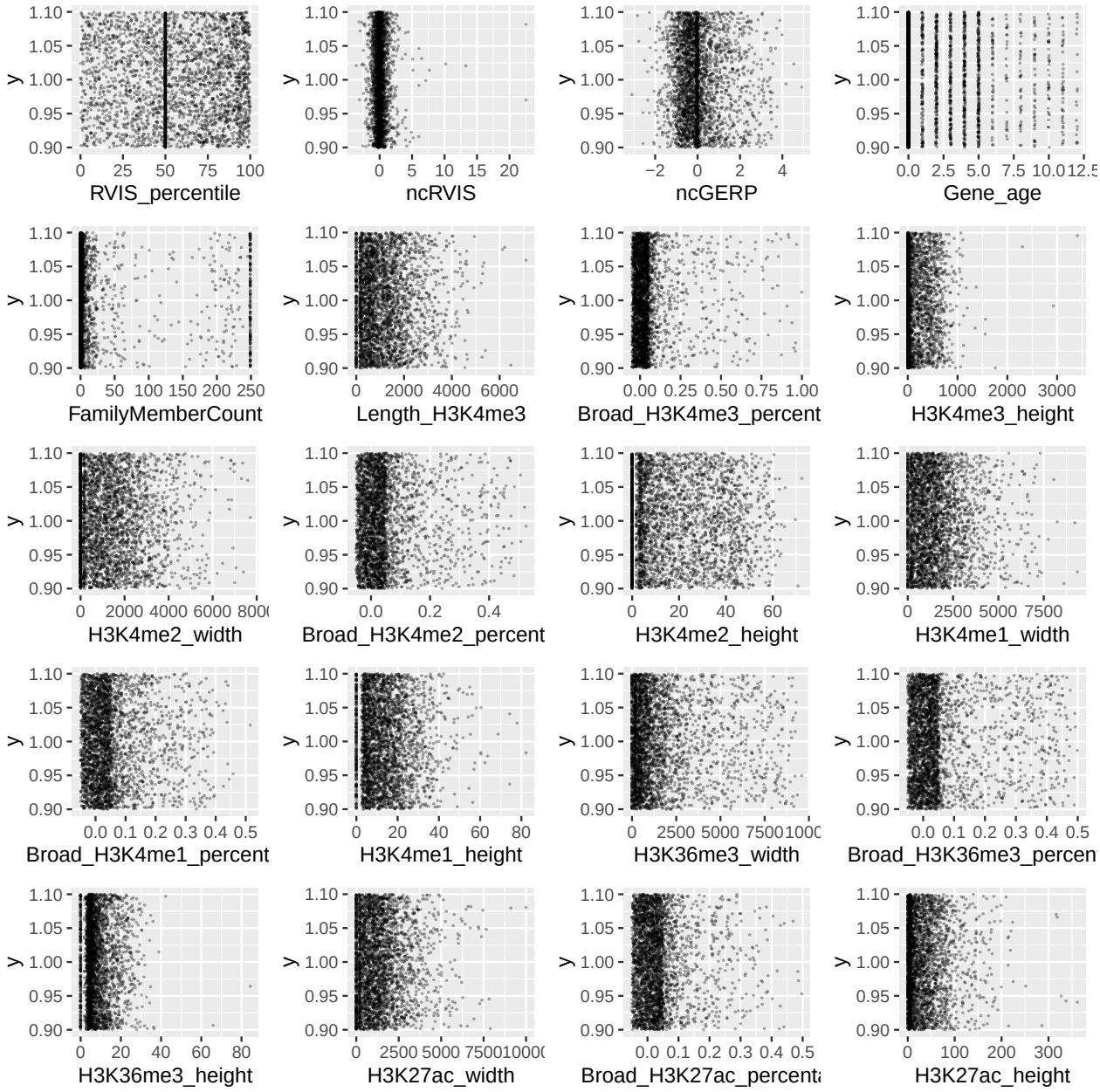

```
grid.arrange(grobs = scat_plot[21:40], ncol = 4)
```



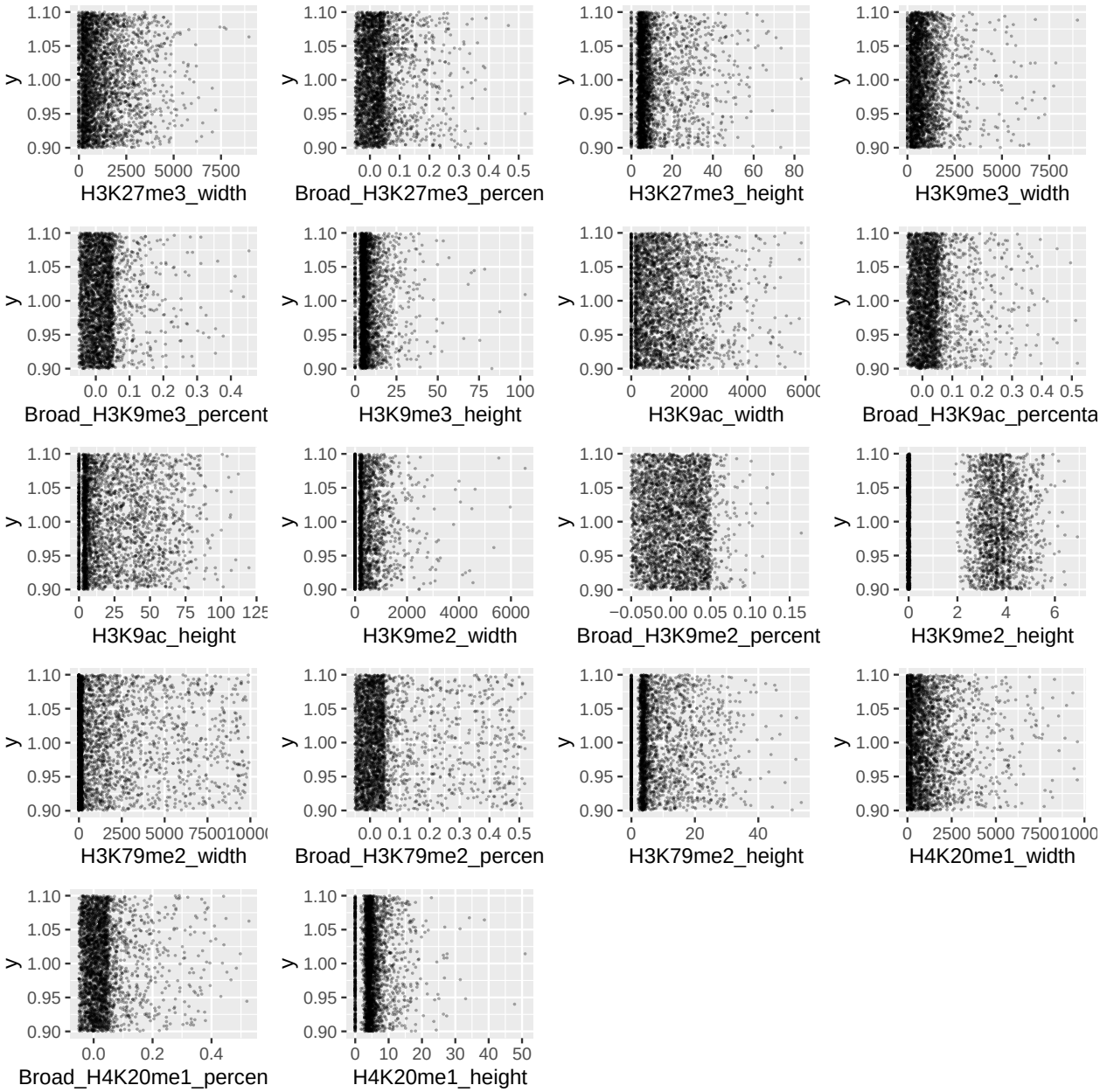

```
grid.arrange(grobs = scat_plot[41:60], ncol = 4)
```




```
grid.arrange(grobs = scat_plot[61:80], ncol = 4)
```




```
grid.arrange(grobs = scat_plot[81:98], ncol = 4)
```



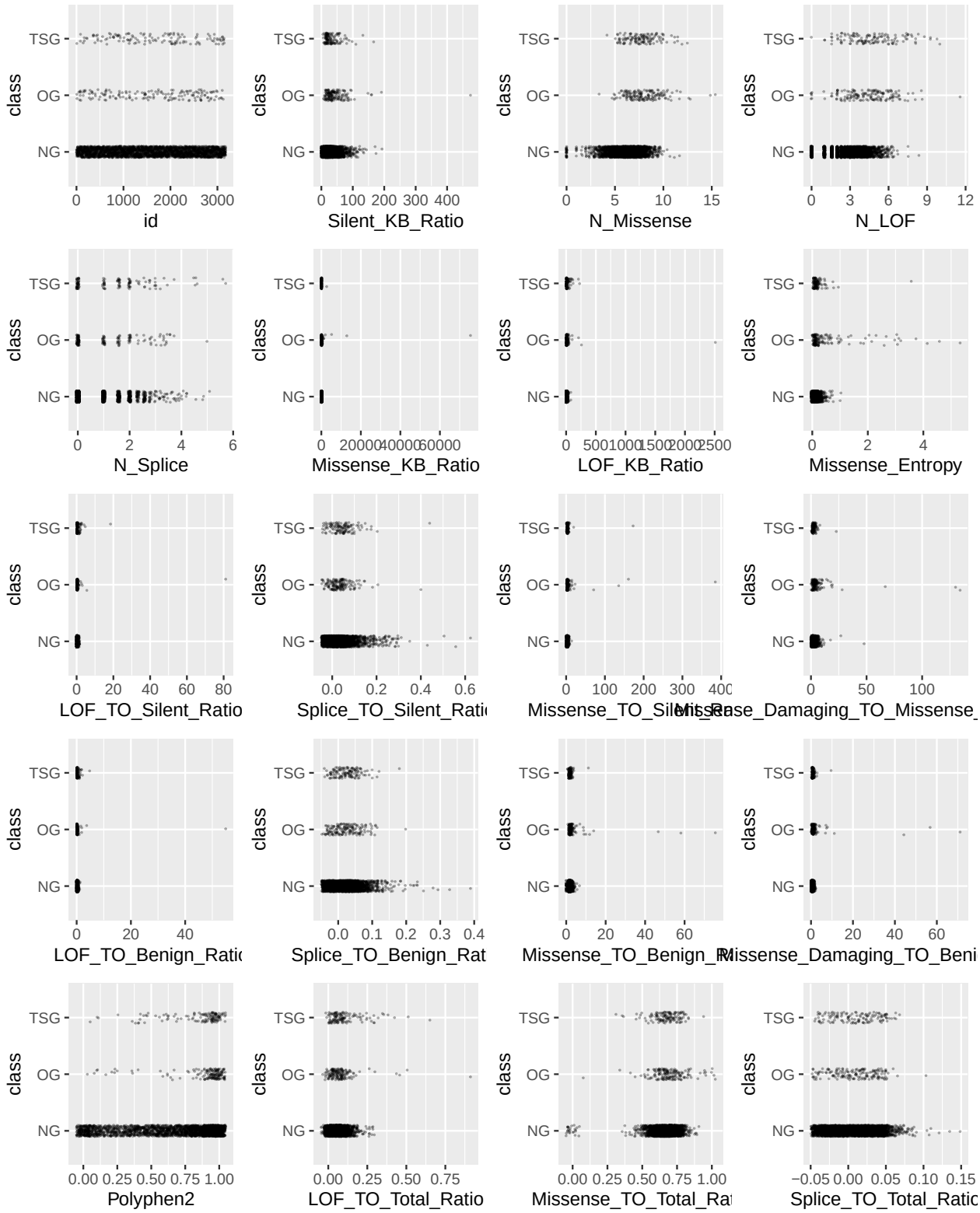

```
outlier <- function(data) {
  q1q3 <- IQR(data)
  q1 <- quantile(data)[[2]]
  q3 <- quantile(data)[[4]]
  low <- q1 - 1.5 * q1q3
  high <- q3 + 1.5 * q1q3
  which(data < low | data > high)
}
outlier_index <- sort(table(unlist(lapply(training[,-99], outlier))), decreasing = TRUE)
outlier_index[1:100]
```

```
 740  915  806  150  517 2918 2641 2621  276 1749 1911 2518 2815  277  341 2215
  49   46   45   43   43   43   42   41   40   40   40   40   40   39   39   39
3049  259  698 1171 1858 1556 1726 2182 2555 2694 3166  422  441 1914 2968 1280
```

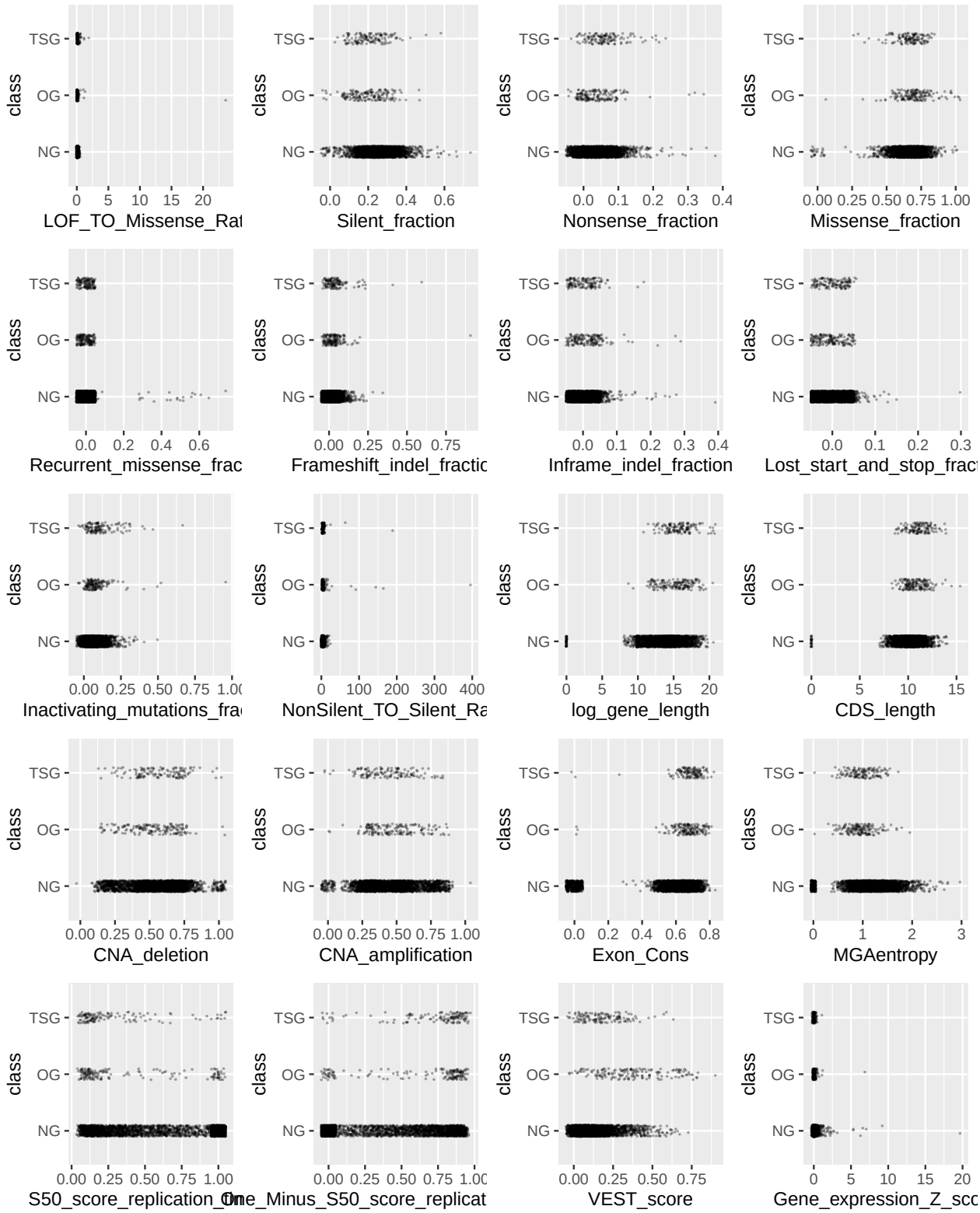
```
  39   38   38   38   38   37   37   37   37   37   37   36   36   36   36   35
1932 2421 3120   73  169  417  460 1955  588  614 1173 1509 2071 2624 1317 1979
  35   35   35   34   34   34   34   34   33   33   33   33   33   33   32   32
2096  657 1135 1138 1188 1258 1809 2005 2022 2093 2278  751 1063 1096 1460 1570
  32   31   31   31   31   31   31   31   31   31   31   30   30   30   30   30
3027 3052  700 1137 2023 2031 2106 2284 2329 2729  343  364 1301 1330 1401 1528
  30   30   29   29   29   29   29   29   29   29   28   28   28   28   28   28
  74  857 1089 2848 2890  368 2492 2561 2654 2930 2983 2998 3080 3134    7    8
  27   27   27   27   27   26   26   26   26   26   26   26   26   26   25   25
 116  361  635  918
  25   25   25   25
```

```
training <- training[-as.numeric(names(outlier_index)[1:50]),]
```

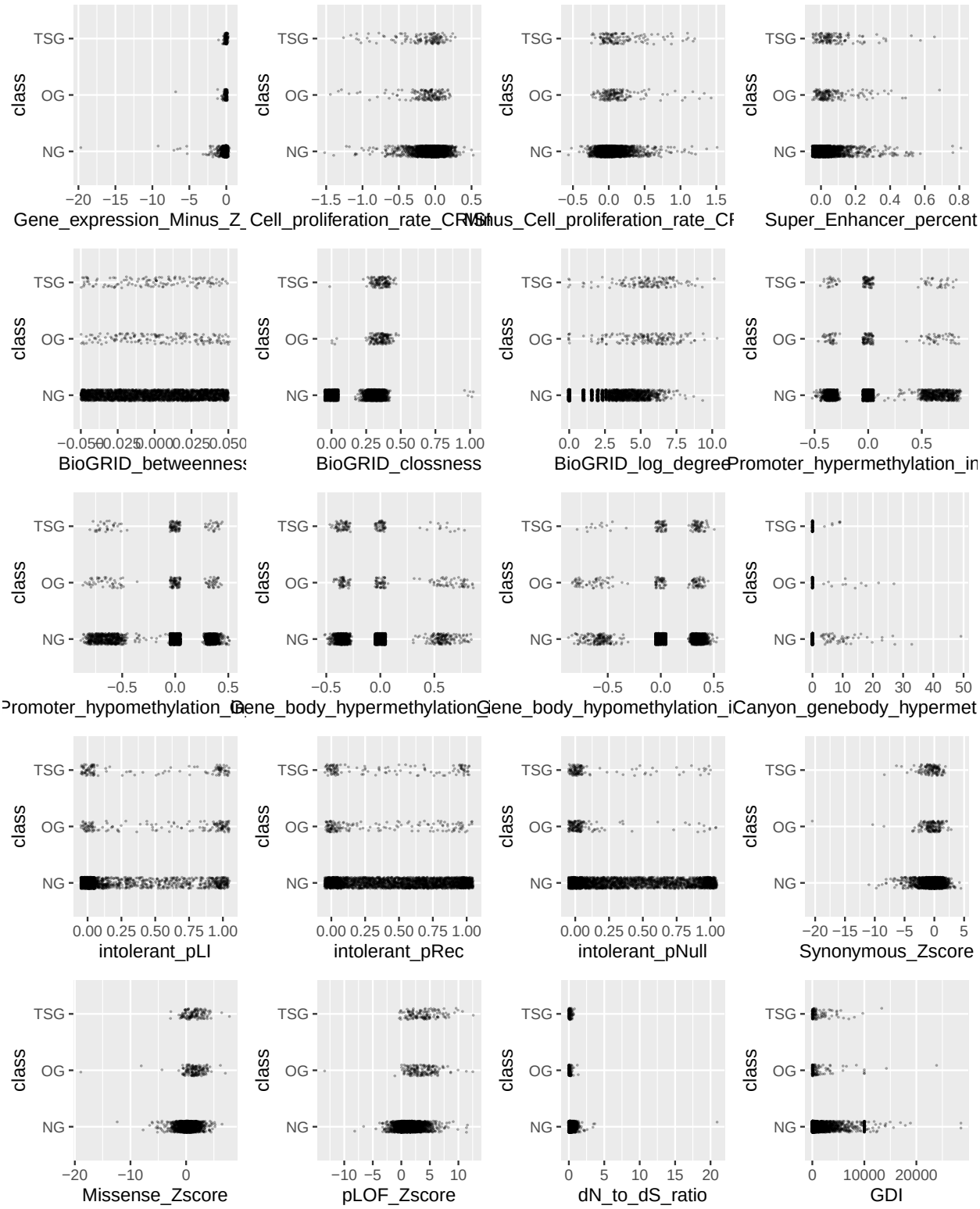
```
library(ggplot2)
scatter <- function(var) {
  ggplot(training, aes_string(var, "class")) +
    geom_jitter(width = 0.05, height = 0.1, size = 0.1,
                colour = rgb(0, 0, 0, alpha = 1 / 3))
}
scat_plot <- lapply(names(training)[-99], scatter)
library(gridExtra)
grid.arrange(grobs = scat_plot[1:20], ncol = 4)
```



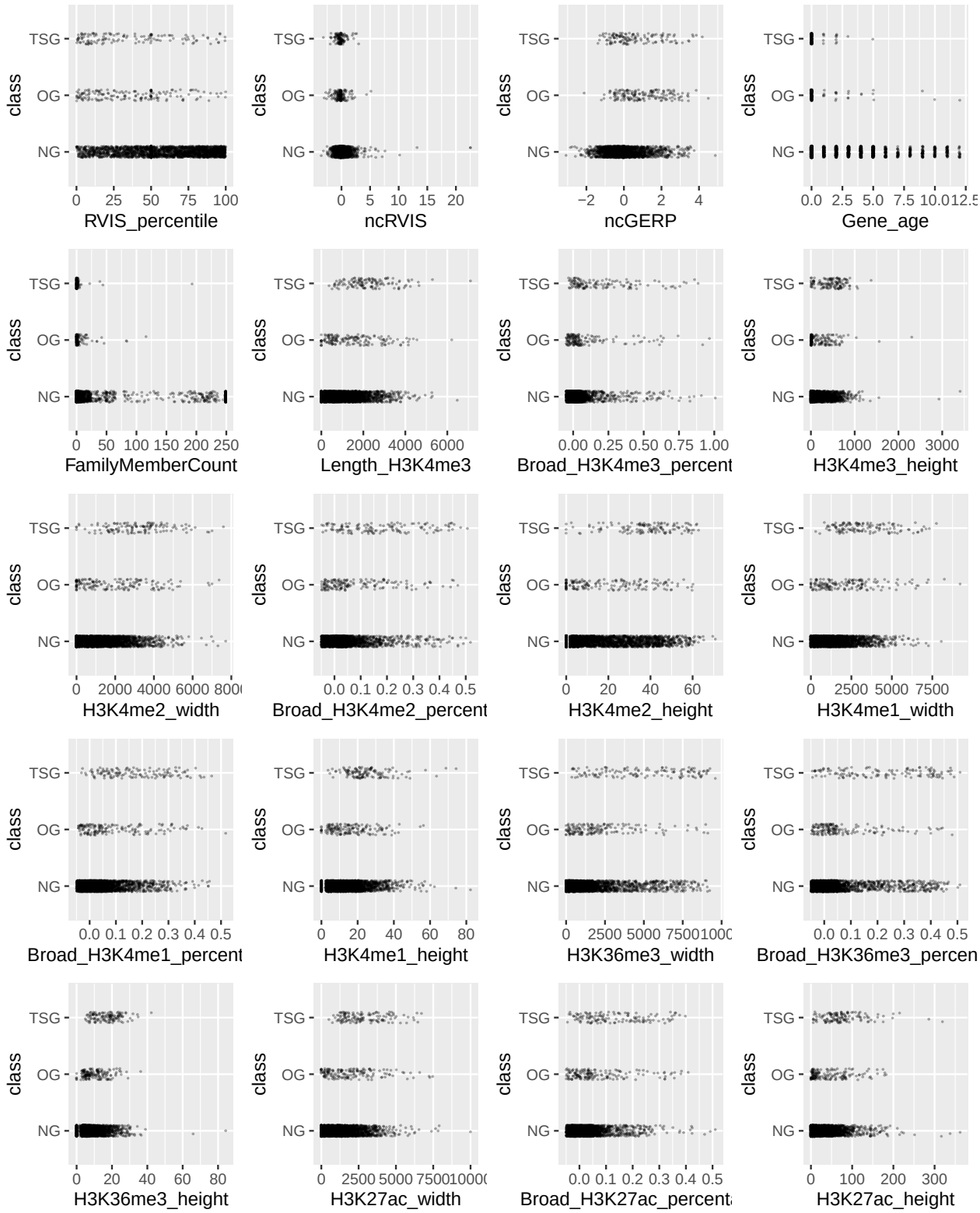

```
grid.arrange(grobs = scat_plot[21:40], ncol = 4)
```



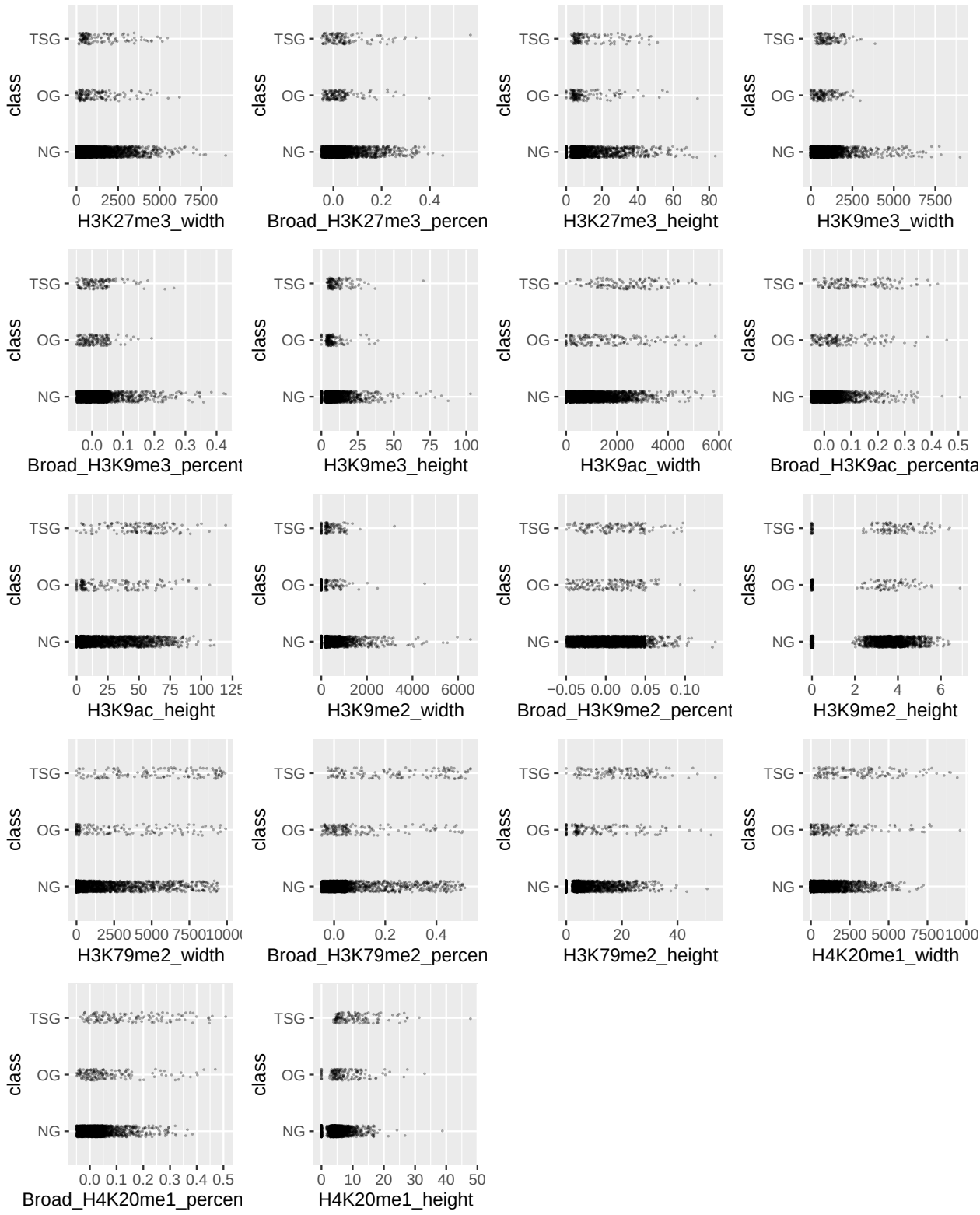

```
grid.arrange(grobs = scat_plot[41:60], ncol = 4)
```




```
grid.arrange(grobs = scat_plot[61:80], ncol = 4)
```




```
grid.arrange(grobs = scat_plot[81:98], ncol = 4)
```




```
sig <- logical(98)
names(sig) <- names(training)[-99]
k <- 1
diffs <- logical(98)
```

```
for (var in names(training)[-99]) {
  model <- aov(training[[var]] ~ factor(training$class))
  sig[k] <- summary(model)[[1]][1, 5]
  diffs[k] <- all(TukeyHSD(model)$`factor(training$class)`[, 4] < 0.05)
  k <- k + 1
}
sort(sig[diffs])
```

```
            Broad_H4K20me1_percentage                                VEST_score
                         1.498169e-147                             6.727430e-129
                     H3K79me2_height                          Missense_Entropy
                         1.937965e-118                             9.205656e-113
                     H3K79me2_width                    Broad_H3K9ac_percentage
                          9.830457e-113                             2.137288e-112
                     H4K20me1_width                           H4K20me1_height
                         1.628744e-111                             1.474127e-109
            Broad_H3K79me2_percentage                           H3K36me3_width
                         8.723288e-106                             2.168247e-104
             Broad_H3K4me2_percentage                 Broad_H3K36me3_percentage
                         5.095574e-104                             8.684169e-104
                      intolerant_pLI                  Broad_H3K27ac_percentage
                         1.014598e-103                             1.516784e-102
             Broad_H3K4me1_percentage                           H3K4me1_width
                         1.114153e-96                              1.171423e-84
                      H3K4me2_width                  Broad_H3K4me3_percentage
                         1.010925e-78                              6.215842e-77
                     H3K36me3_height                           H3K9ac_width
                          8.382413e-77                             1.831327e-75
                      H3K27ac_width                           H3K9ac_height
                          4.331525e-67                             9.013307e-65
                     Length_H3K4me3                                  ncGERP
                          4.243439e-62                             8.644312e-62
                     H3K27ac_height                          H3K4me3_height
                          1.318182e-59                             9.947710e-53
                     H3K4me2_height                                 N_Splice
                          3.384412e-51                             1.301985e-45
                     H3K4me1_height                     BioGRID_betweenness
                          1.309236e-45                             5.506022e-37
                LOF_TO_Total_Ratio One_Minus_S50_score_replication_timing
                          6.074498e-35                             7.975599e-32
      S50_score_replication_timing                 Frameshift_indel_fraction
                          7.975599e-32                             4.803819e-27
    Inactivating_mutations_fraction                 NonSilent_TO_Silent_Ratio
                          2.842485e-24                             1.110805e-23
          Missense_TO_Silent_Ratio                    Inframe_indel_fraction
                          1.824400e-19                             5.282904e-11
```

```
score <- function (conf_mat) {
  print(sum(diag(conf_mat) * c(1, 20, 20)))
  print(sum(diag(conf_mat) * c(1, 20, 20)) / sum(apply(conf_mat, 2, sum) * c(1, 20, 20)))
}

classify <- function(probs) {
  if (any(probs[2:3] > 0.05)) {
```

```
    subset <- probs[2:3]
    output <- which(subset == max(subset))
    if (length(output) > 1) {
        output <- sample(1:2, 1)
    }
  } else {
    output <- 0
  }
  output
}
```

```
library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following object is masked from 'package:gridExtra':

    combine
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

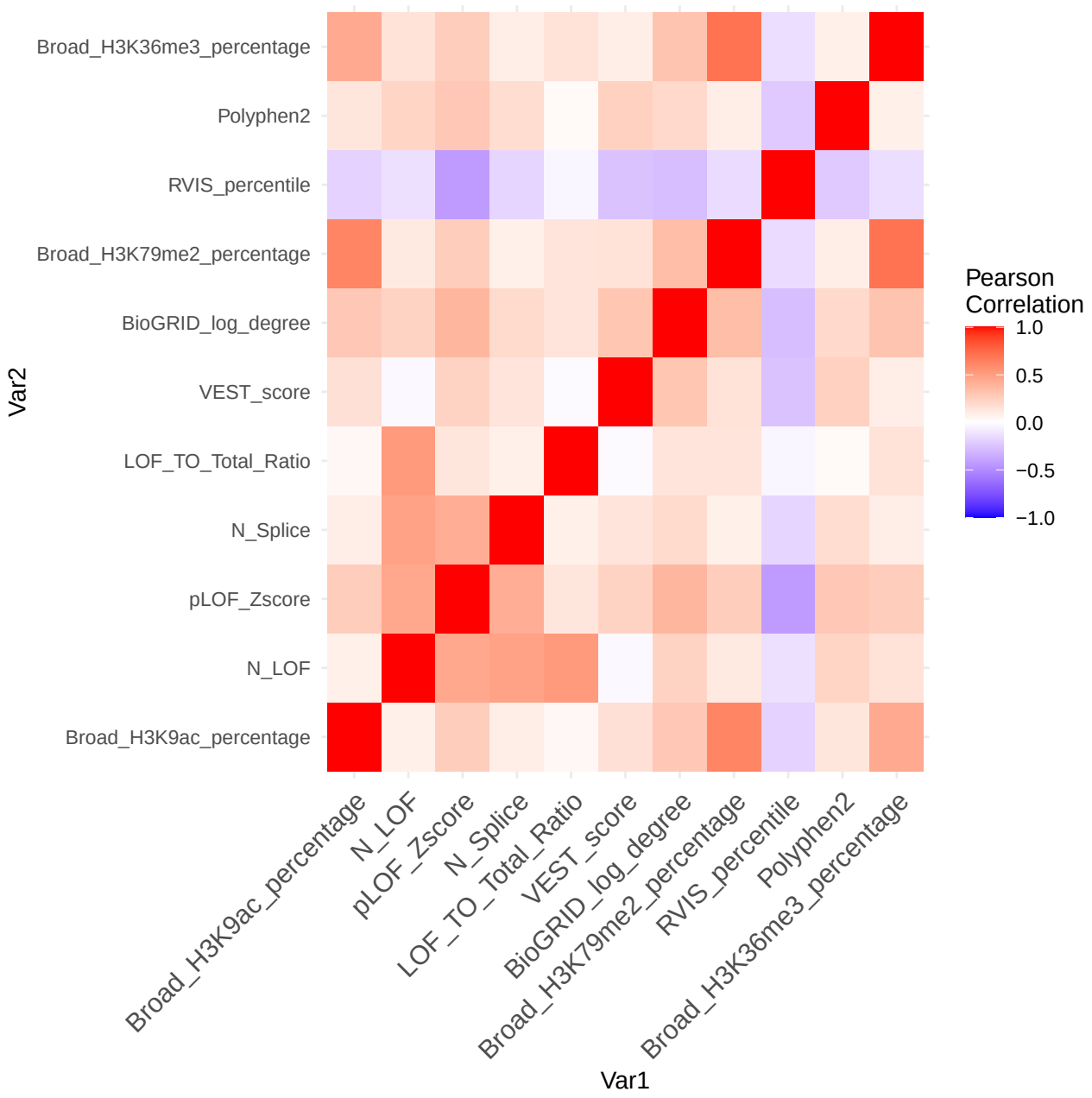
```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
vars <- training %>% select(Broad_H3K9ac_percentage, N_LOF, pLOF_Zscore,
                            N_Splice, LOF_TO_Total_Ratio, VEST_score, BioGRID_log_degree, Broad_H3K79me
                            RVIS_percentile,
                            Polyphen2, Broad_H3K36me3_percentage, class)
vars$class <- factor(vars$class)
levels(vars$class) <- c("NG", "OG", "TSG")
cor_mtx = round(cor(vars[, names(vars) != "class"]), 2)
library(reshape2)
#reshape it
melted_cor_mtx <- melt(cor_mtx)

#draw the heatmap
cor_heatmap = ggplot(data = melted_cor_mtx, aes(x=Var1, y=Var2, fill=value)) + geom_tile()
cor_heatmap = cor_heatmap +
scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1), space =
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))

cor_heatmap
```




```
set.seed(nrow(training))
library(caret)
```

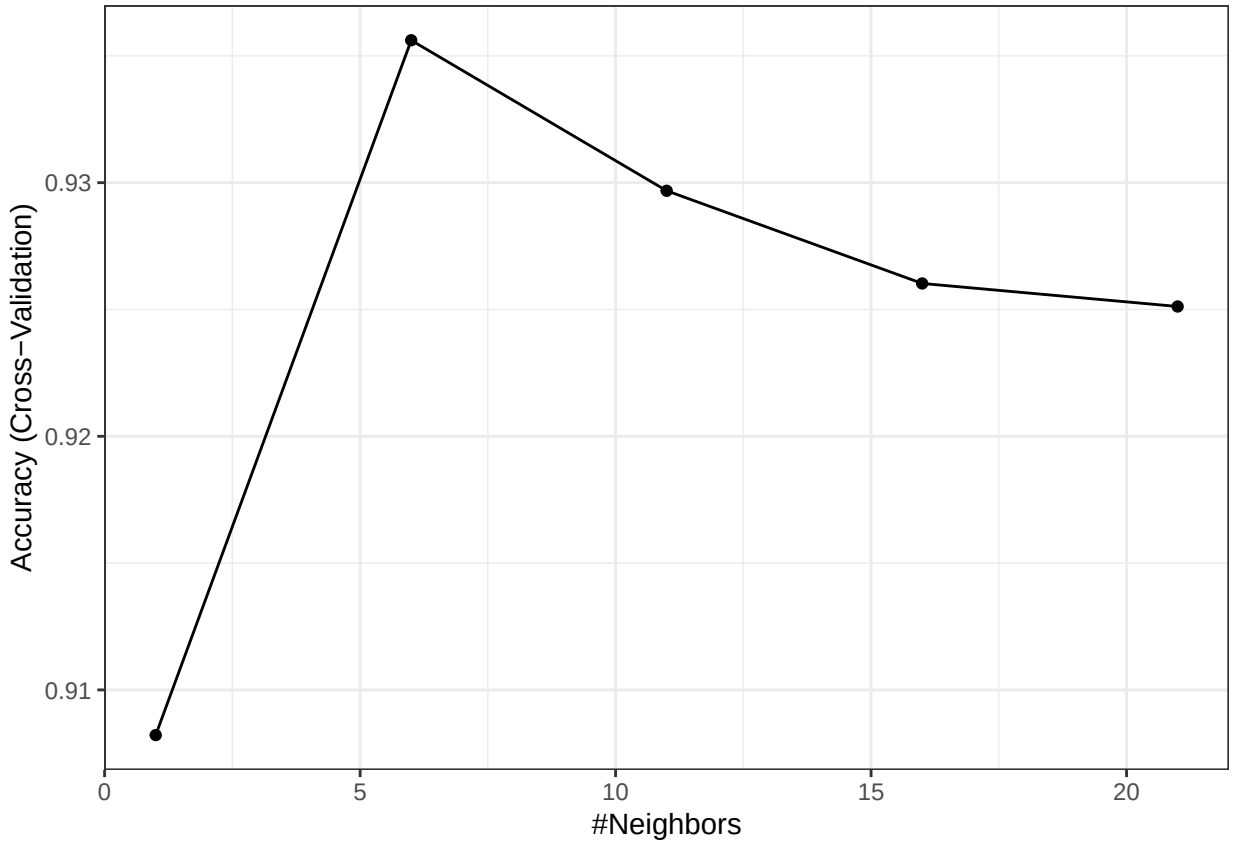
```
Loading required package: lattice
```

```
vars_test <- createDataPartition(vars$class, p = 0.7,
                                 list = FALSE)
vars_train <- vars[vars_test, ]
vars_test <- vars[-vars_test, ]

train_cont <- trainControl(method = "cv", number = 5, classProbs = TRUE, savePredictions = TRUE)


knn_ft <- train(class ~ ., data = vars_train, method = "knn", preProc = c("center", "scale"),
```

```
                trControl = train_cont, tuneGrid = expand.grid(k = seq(from = 1, to = 25, by = 5)))
ggplot(knn_ft) + theme_bw()
```




```
for (k in seq(from = 1, to = 25, by = 5)) {
  preds <- predict(knn_ft, newdata = vars_test, type = "prob")
  knn_mod <- table("pred"=unlist(apply(preds, 1, classify)), "obs" = vars_test$class)
  print(knn_mod)
  score(knn_mod)
}
```

```
    obs
pred  NG  OG TSG
   0 727   8   4
   1  71  29   5
   2  54  10  29
[1] 1887
[1] 0.7394201
    obs
pred  NG  OG TSG
   0 727   8   4
   1  71  29   5
   2  54  10  29
[1] 1887
[1] 0.7394201
    obs
pred  NG  OG TSG
   0 727   8   4
```

```
   1  70  28   8
   2  55  11  26
[1] 1807
[1] 0.7080721
    obs
pred  NG  OG TSG
   0 727   8   4
   1  73  28   7
   2  52  11  27
[1] 1827
[1] 0.7159091
    obs
pred  NG  OG TSG
   0 727   8   4
   1  71  27   7
   2  54  12  27
[1] 1807
[1] 0.7080721
```

```
train_cont <- trainControl(method = "cv", number = 5, classProbs = TRUE, savePredictions = TRUE)
qda_ft <- train(class ~ ., data = vars_train, method = "qda", preProc = c("center", "scale"),
                trControl = train_cont)
preds <- predict(qda_ft, newdata = vars_test, type = "prob")

qda_mod <- table("pred"=apply(preds, 1, classify), "obs" = vars_test$class)
qda_mod
```

```
    obs
pred  NG  OG TSG
   0 743  11   3
   1  54  26   9
   2  55  10  26
```

```
score(qda_mod)
```

```
[1] 1783
[1] 0.6986677
```

```
train_cont <- trainControl(method = "cv", number = 5, classProbs = TRUE, savePredictions = TRUE)
lda_ft <- train(class ~ ., data = vars_train, method = "lda", preProc = c("center", "scale"),
                trControl = train_cont)
preds <- predict(lda_ft, newdata = vars_test, type = "prob")

lda_mod <- table("pred"=apply(preds, 1, classify), "obs" = vars_test$class)
lda_mod
```

```
    obs
pred  NG  OG TSG
   0 739   7   4
   1  69  33   8
   2  44   7  26
```

```
score(lda_mod)
```

```
[1] 1919
[1] 0.7519592
```

```
tests <- read.csv("test.csv")
rel_vars <- training %>% select(Broad_H3K9ac_percentage, N_LOF, pLOF_Zscore,
                            N_Splice, LOF_TO_Total_Ratio, VEST_score, BioGRID_log_degree, Broad_H3K79me
                            RVIS_percentile,
                            Polyphen2, Broad_H3K36me3_percentage, class)
for (i in 1:11) {
  rel_vars[[i]] <- (rel_vars[[i]] - lda_ft$preProcess$mean[i])/ lda_ft$preProcess$std[i]
}
preds <- predict(lda_ft, newdata = tests, type = "prob")
preds <- apply(preds, 1, classify)
names(preds) <- tests$id
csv_file <- data.frame("id" = tests$id,
                       "class" = preds)
write.csv(csv_file, "modelpredictions.csv", row.names = FALSE)
```