

Preliminary Investigation

Ethan Allavarpu (UID: 405287603)

10/27/2020

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

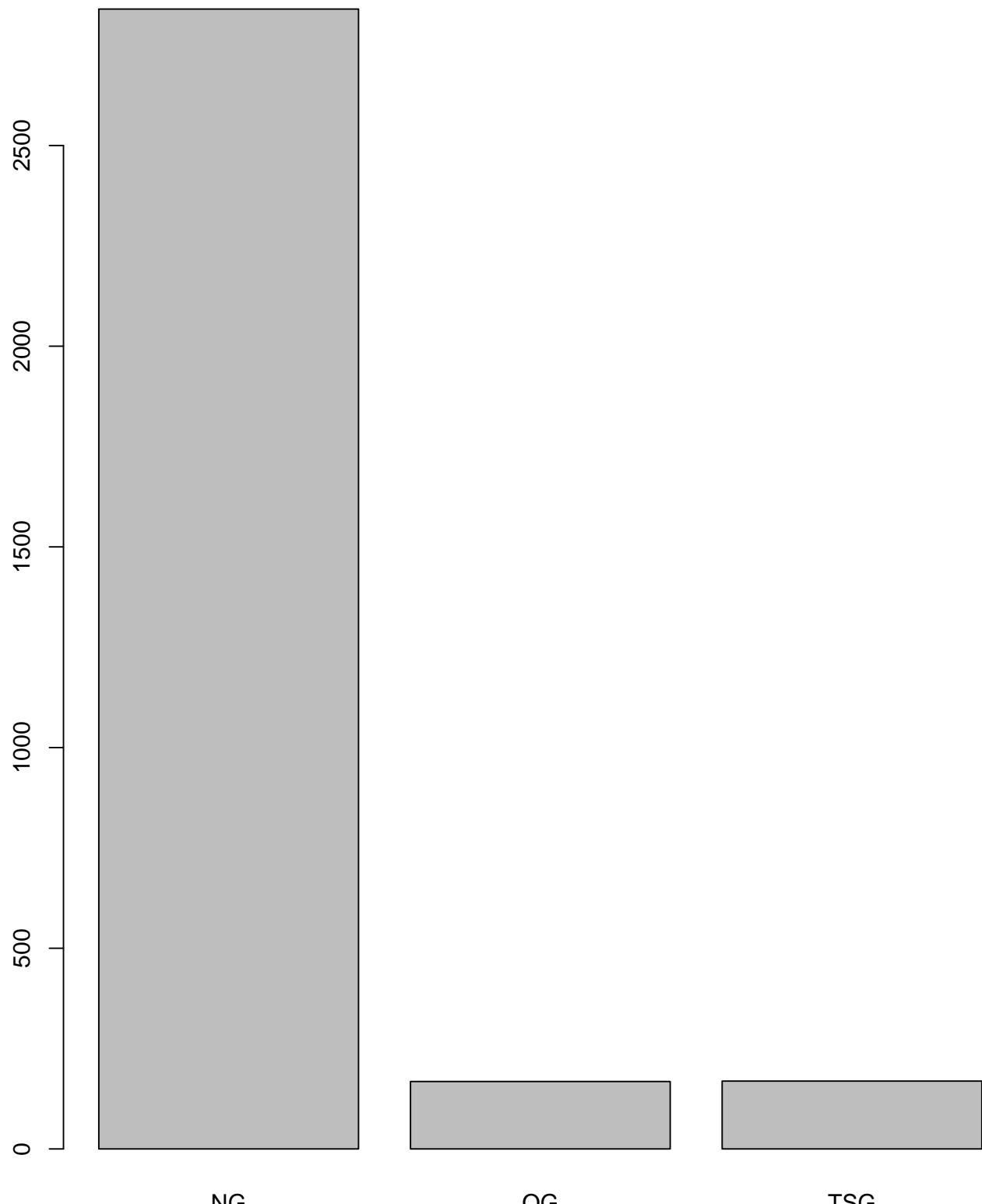
training <- read.csv("training.csv", stringsAsFactors = TRUE)
training$class <- factor(training$class)
levels(training$class) <- c("NG", "OG", "TSG")
dim(training)
```

```
[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)
```

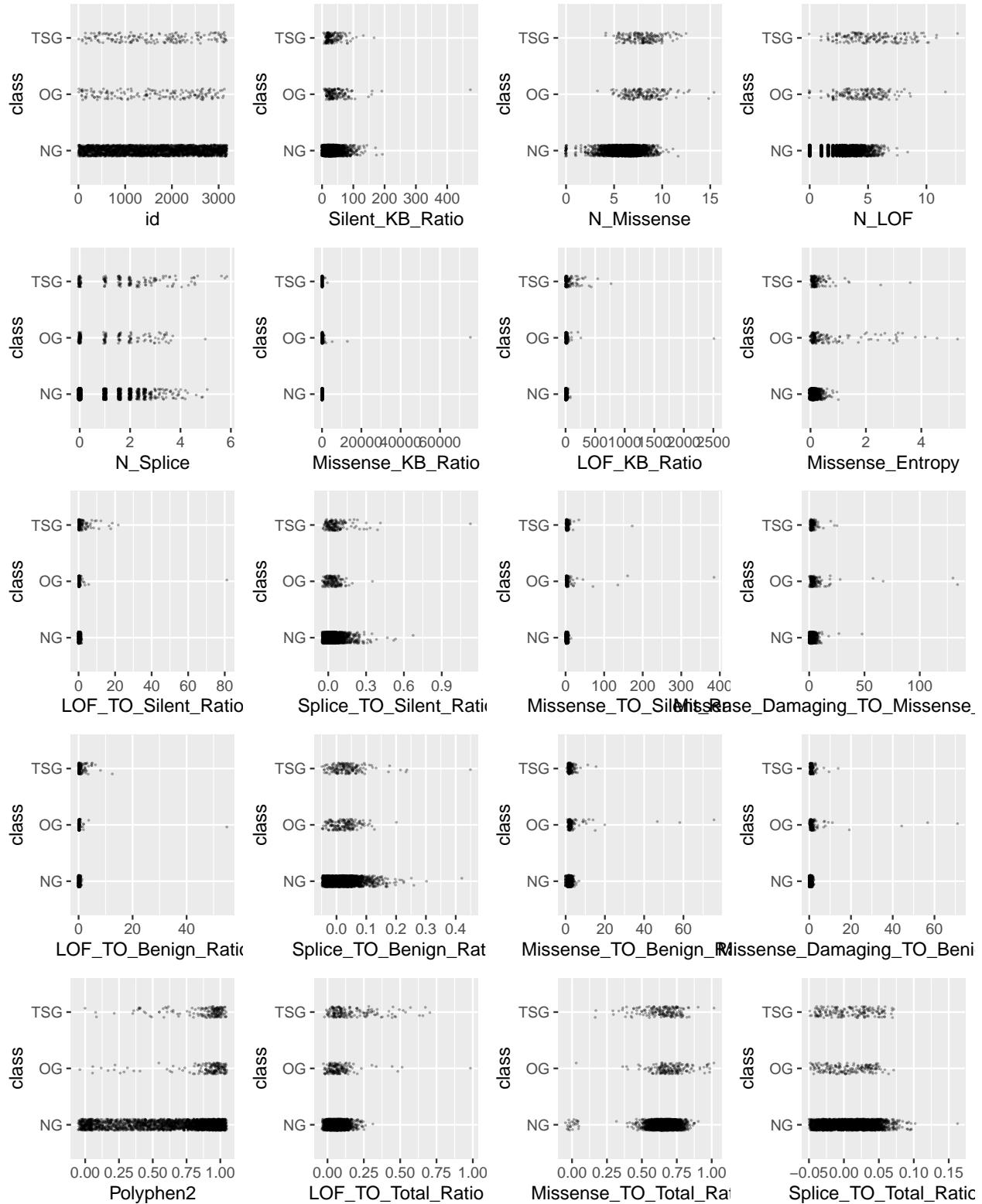


```
scatter <- function(var) {  
  ggplot(training, aes_string(var, "class")) +  
    geom_jitter(width = 0.05, height = 0.1, size = 0.1,  
               colour = rgb(0, 0, 0, alpha = 1 / 3))  
}
```

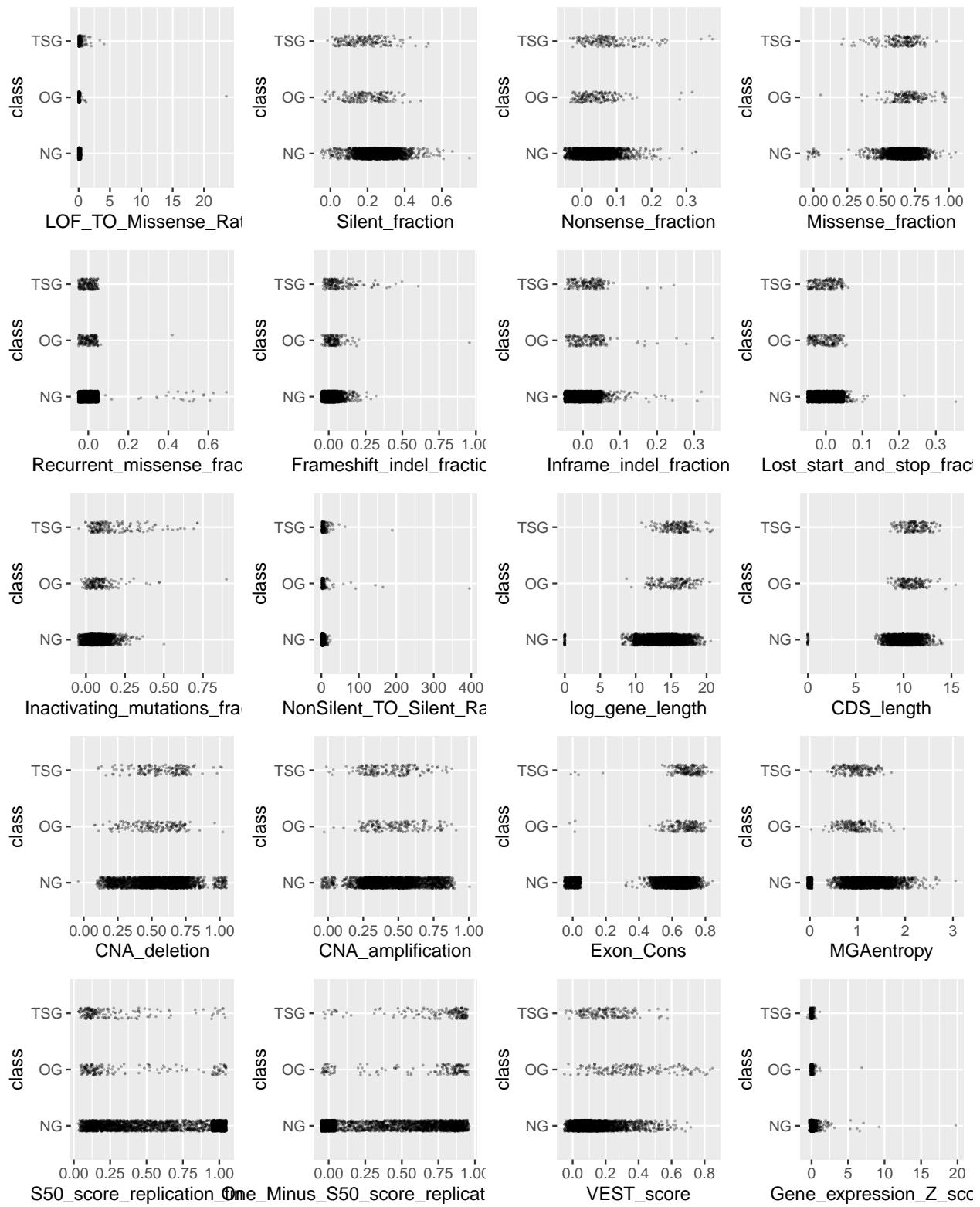
```

scat_plot <- lapply(names(training)[-99], scatter)
library(gridExtra)
grid.arrange(grobs = scat_plot[1:20], ncol = 4)

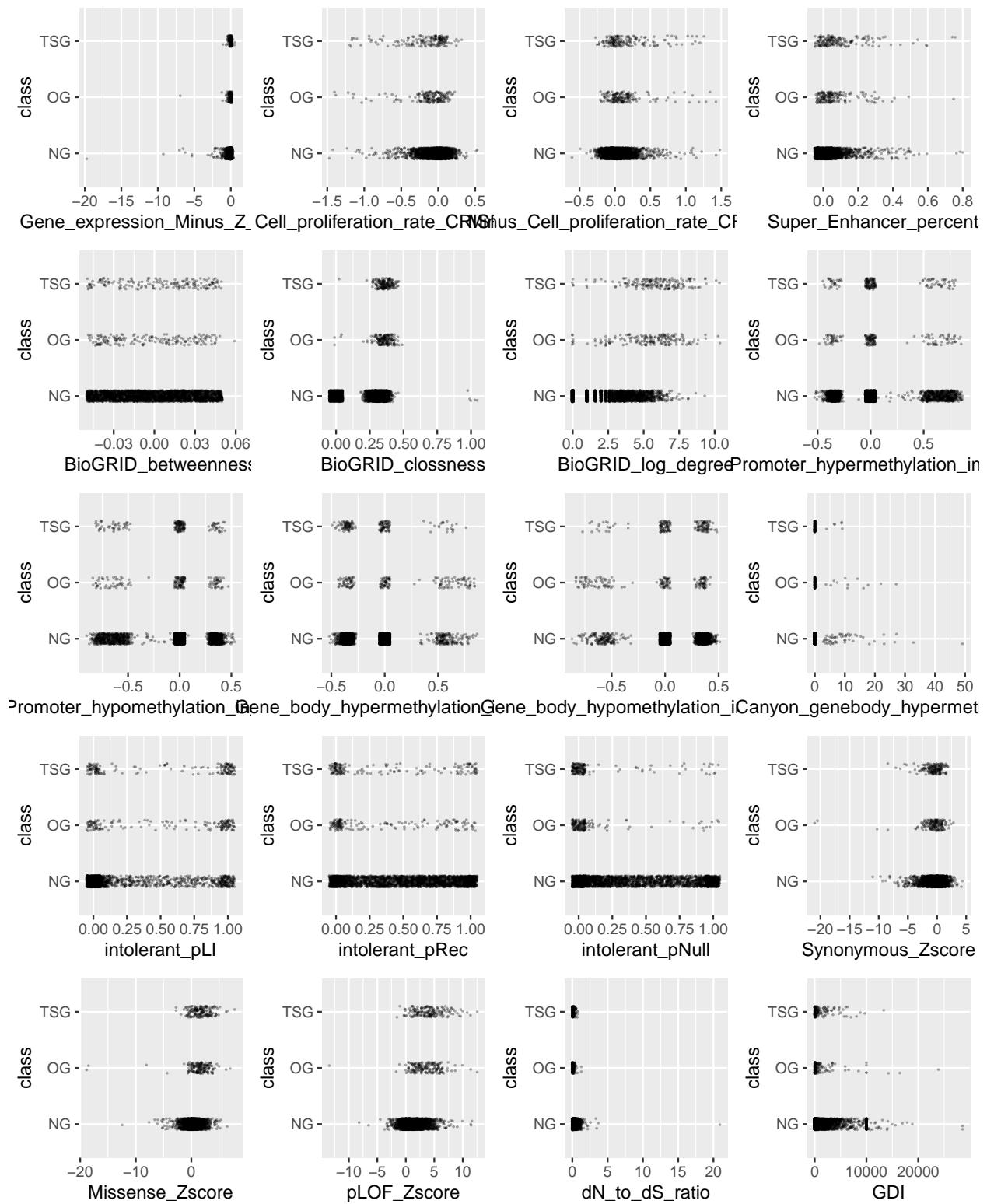
```



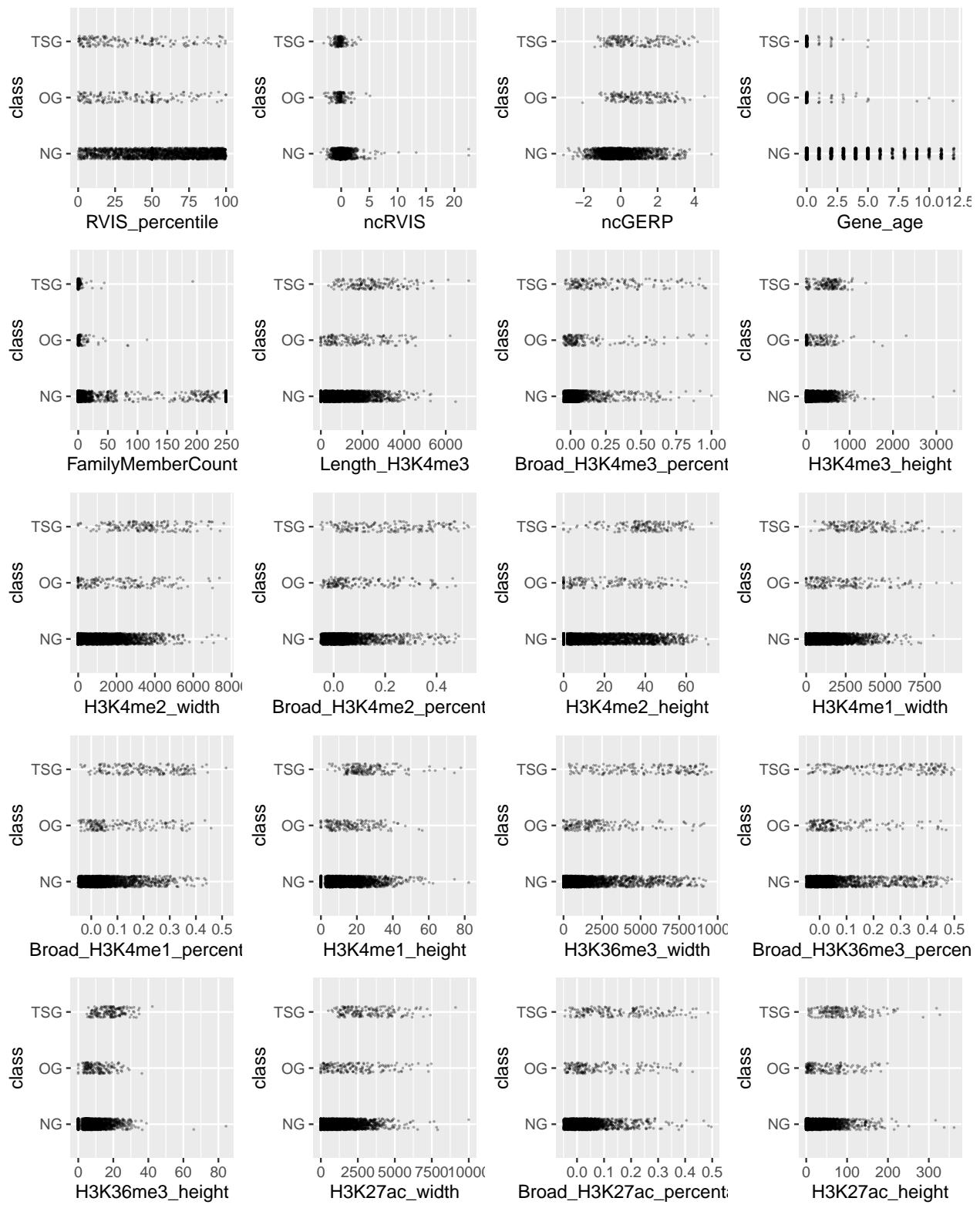
```
grid.arrange(grobs = scat_plot[21:40], ncol = 4)
```



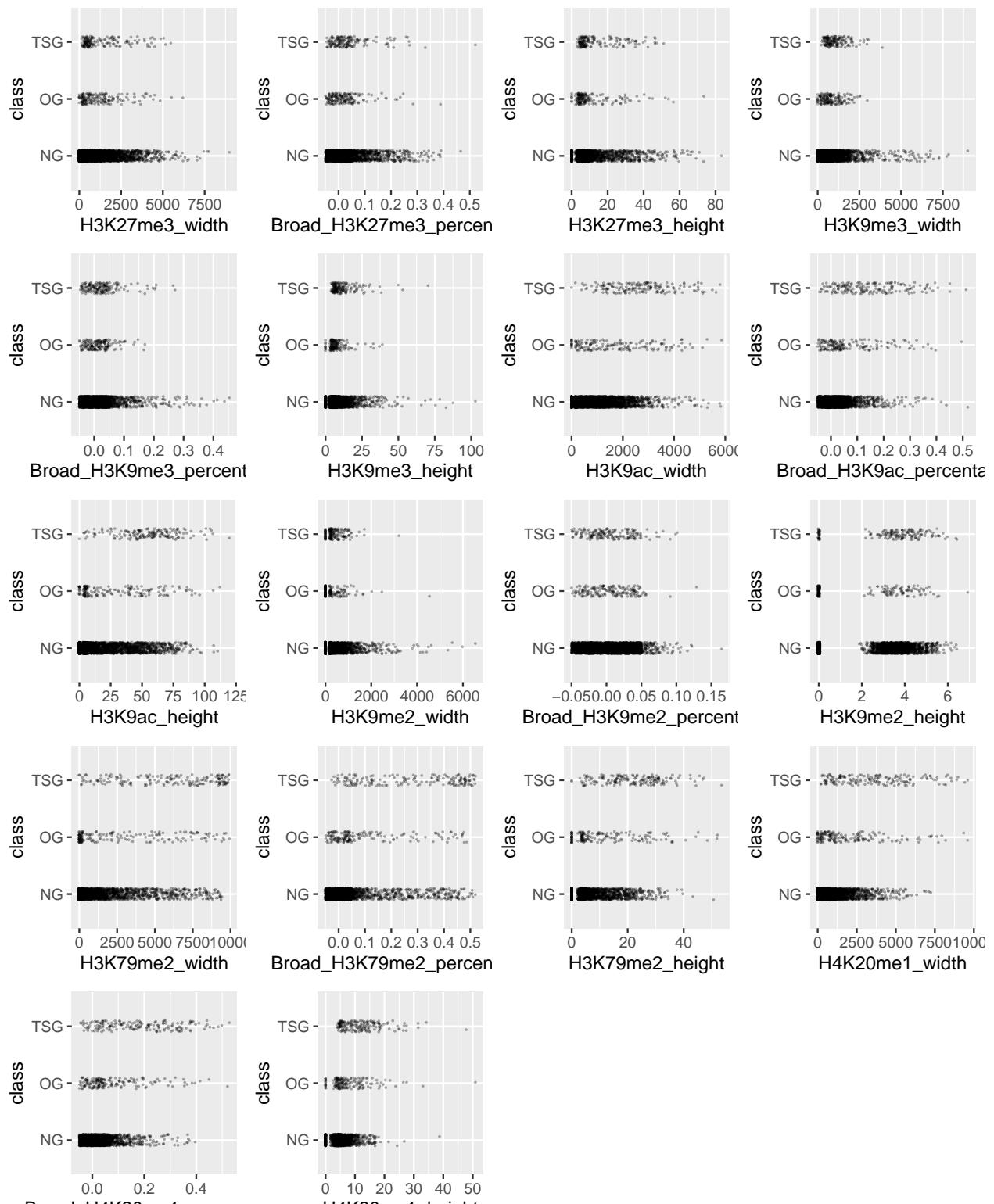
```
grid.arrange(grobs = scat_plot[41:60], ncol = 4)
```



```
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	2.605382e-232
Broad_H3K9ac_percentage	8.993961e-184
H4K20me1_width	3.222855e-181
H3K79me2_height	4.253756e-176
H3K79me2_width	2.449130e-175
Broad_H3K4me2_percentage	4.311054e-174
Broad_H3K79me2_percentage	1.188924e-168
Broad_H3K27ac_percentage	2.345807e-168
H4K20me1_height	1.791736e-164
Broad_H3K4me1_percentage	1.065370e-163
Broad_H3K36me3_percentage	4.471080e-154
H3K36me3_width	2.142472e-147
pLOF_Zscore	5.366468e-145
Broad_H3K4me3_percentage	2.017738e-143
H3K4me1_width	5.760242e-141
VEST_score	8.594619e-138
N_LOF	2.985376e-137
Missense_Entropy	2.153653e-134
H3K4me2_width	3.996102e-127
H3K9ac_width	8.106205e-120
H3K36me3_height	3.589138e-112
H3K27ac_width	1.075439e-109
LOF_TO_Total_Ratio	1.795563e-106

```

Length_H3K4me3
5.345828e-104
H3K27ac_height
2.296798e-103
H3K9ac_height
1.404194e-101
Inactivating_mutations_fraction
3.210725e-84
H3K4me3_height
6.912938e-81
N_Splice
2.765822e-77
Frameshift_indel_fraction
4.847595e-77
H3K4me2_height
4.174575e-71
H3K4me1_height
1.800505e-62
Missense_Damaging_TO_Missense_Benign_Ratio
1.778682e-49
One_Minus_S50_score_replication_timing
4.727754e-44
S50_score_replication_timing
4.727754e-44
Missense_TO_Benign_Ratio
2.805142e-43
Missense_Damaging_TO_Benign_Ratio
1.057841e-41
Super_Enhancer_percentage
1.637695e-38
LOF_KB_Ratio
2.566176e-37
LOF_TO_Silent_Ratio
3.416804e-35
NonSilent_TO_Silent_Ratio
2.030973e-33
CDS_length
3.970782e-32
Missense_TO_Silent_Ratio
2.841077e-24
Inframe_indel_fraction
5.669973e-16
Missense_TO_Total_Ratio
1.973442e-09
Missense_fraction
1.031541e-08

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

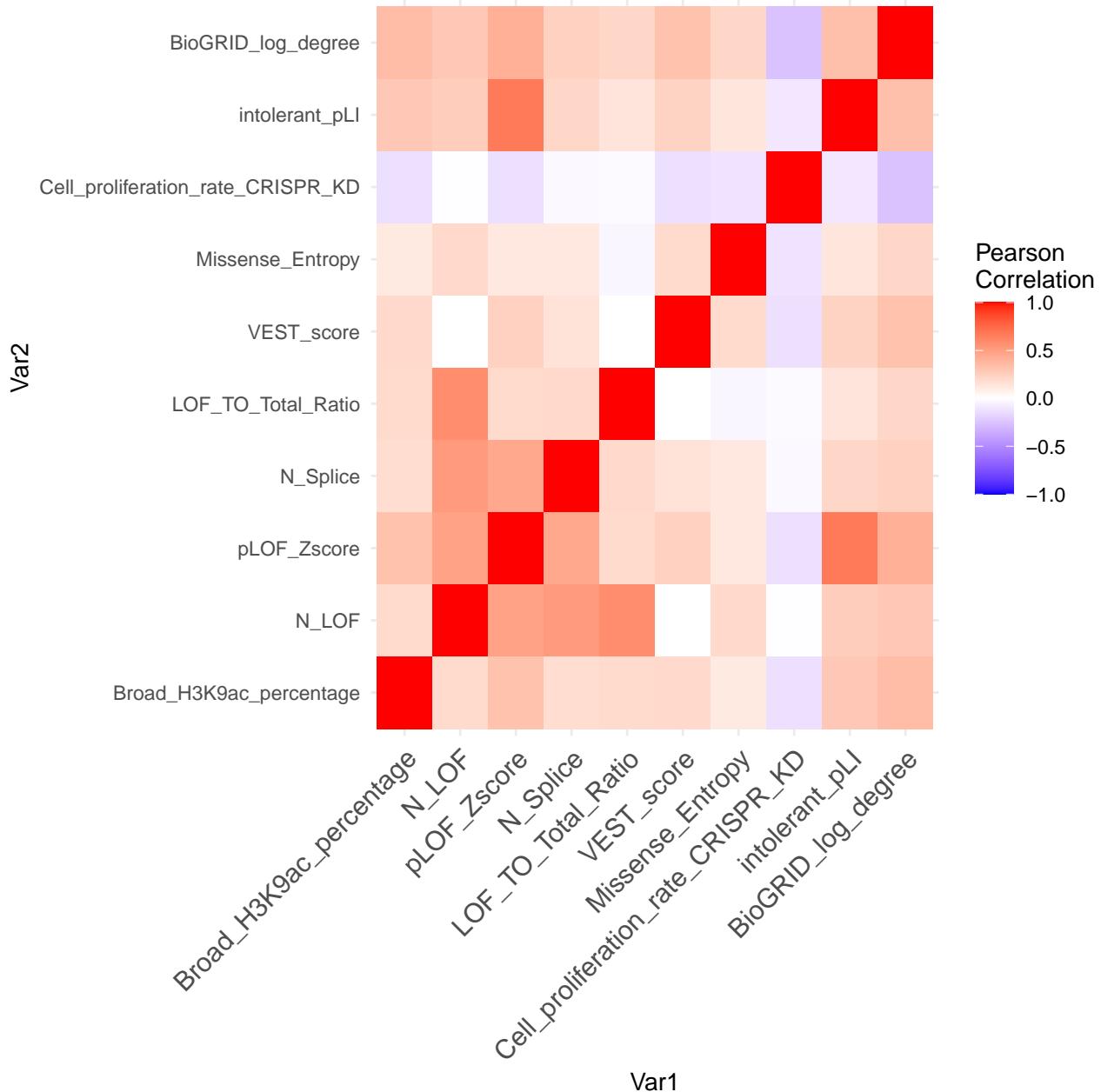
```

vars <- training %>% select(Broad_H3K9ac_percentage, N_LOF, pLOF_Zscore,
                               N_Splice, LOF_TO_Total_Ratio, VEST_score, Missense_Entropy,
                               Cell_proliferation_rate_CRISPR_KD, intolerant_pLI, BioGRID_log_degree, class)
vars$class <- factor(vars$class)
levels(vars$class) <- c("NG", "OG", "TSG")
cor_mtx = round(cor(vars[-11]), 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

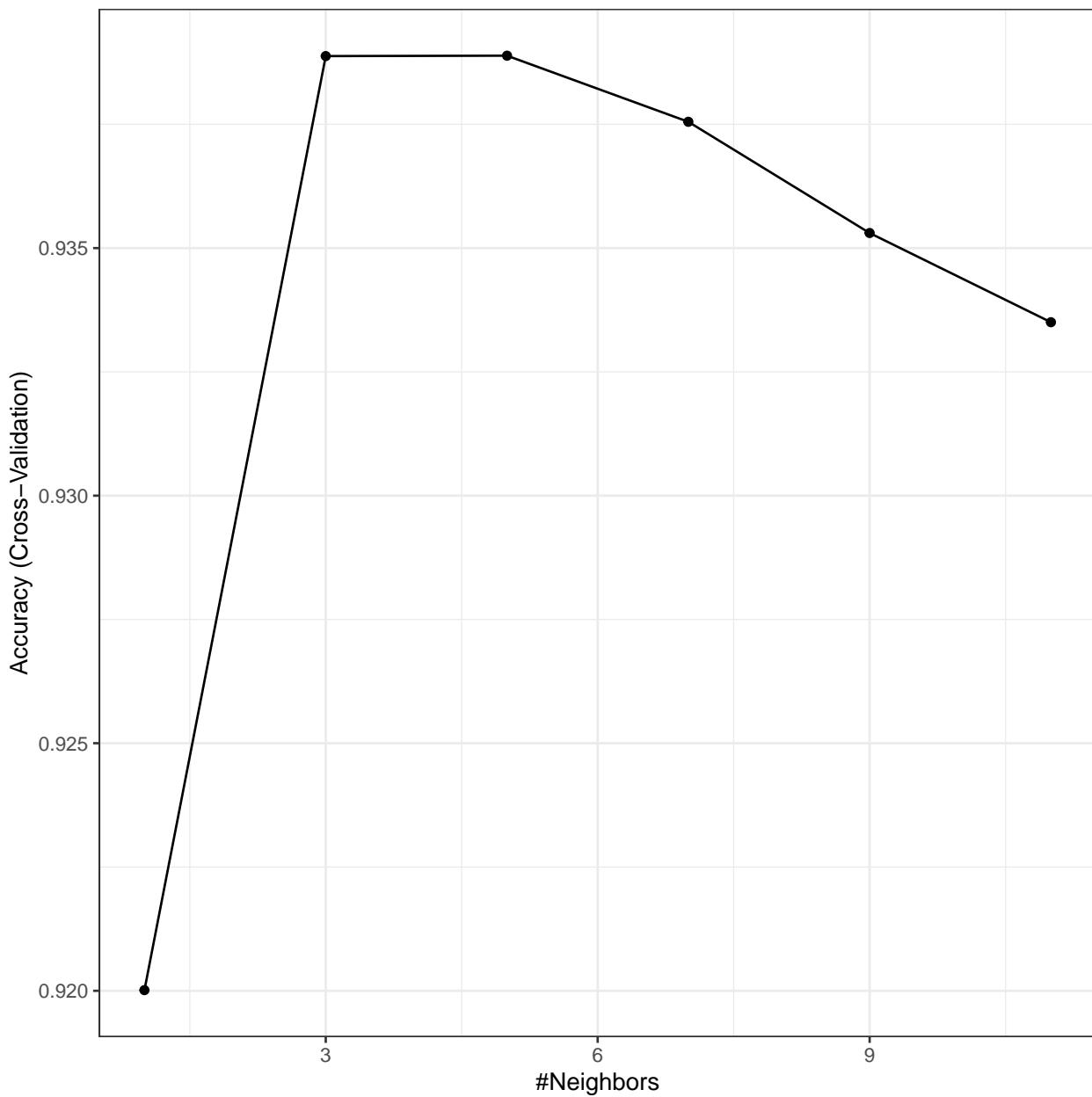
```



```
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 = 10, 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 = 11, by = 2)))
ggplot(knn_ft) + theme_bw()
```



```
for (k in seq(from = 1, to = 11, by = 2)) {
  preds <- predict(knn_ft, newdata = vars_test, type = "prob")
  knn_mod <- table("pred"=unlist(apply(preds, 1, classify)), "obs" = vars_test$class)
  knn_mod
  score(knn_mod)
}
```

```
[1] 2147
[1] 0.752805
[1] 2087
[1] 0.7317672
[1] 2107
[1] 0.7387798
[1] 2087
[1] 0.7317672
```

```

[1] 2107
[1] 0.7387798
[1] 2127
[1] 0.7457924

train_cont <- trainControl(method = "cv", number = 10, classProbs = TRUE, savePredictions = TRUE)
knn_ft <- train(class ~ ., data = vars_train, method = "qda", preProc = c("center", "scale"),
                 trControl = train_cont)
preds <- predict(knn_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
  0 801   6   11
  1 16   29    2
  2 35   15   37

score(qda_mod)

[1] 2121
[1] 0.7436886

train_cont <- trainControl(method = "cv", number = 10, classProbs = TRUE, savePredictions = TRUE)
knn_ft <- train(class ~ ., data = vars_train, method = "lda", preProc = c("center", "scale"),
                 trControl = train_cont)
preds <- predict(knn_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
  0 818   7   13
  1 17   33    2
  2 17   10   35

score(lda_mod)

[1] 2178
[1] 0.7636746

library(leaps)
best_subset <- regsubsets(class ~ ., data = training, nbest = 1, nvmax = 20,
                           intercept = TRUE, method = "forward",
                           really.big = TRUE)

Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
force.in, : 5 linear dependencies found

Reordering variables and trying again:

sumBS <- summary(best_subset)
plot(best_subset)

```

