

Evaluate testing data (regression) - xgboost

EVE W.

2019-04-08

Contents

0. Load Data	1
1. Scores	2
correlation	2
2. Important Features	4
3. Hyper-parameters	7
all grid search results	8
over best parameter combo per cv	10
more about the best parameter combination selection	12

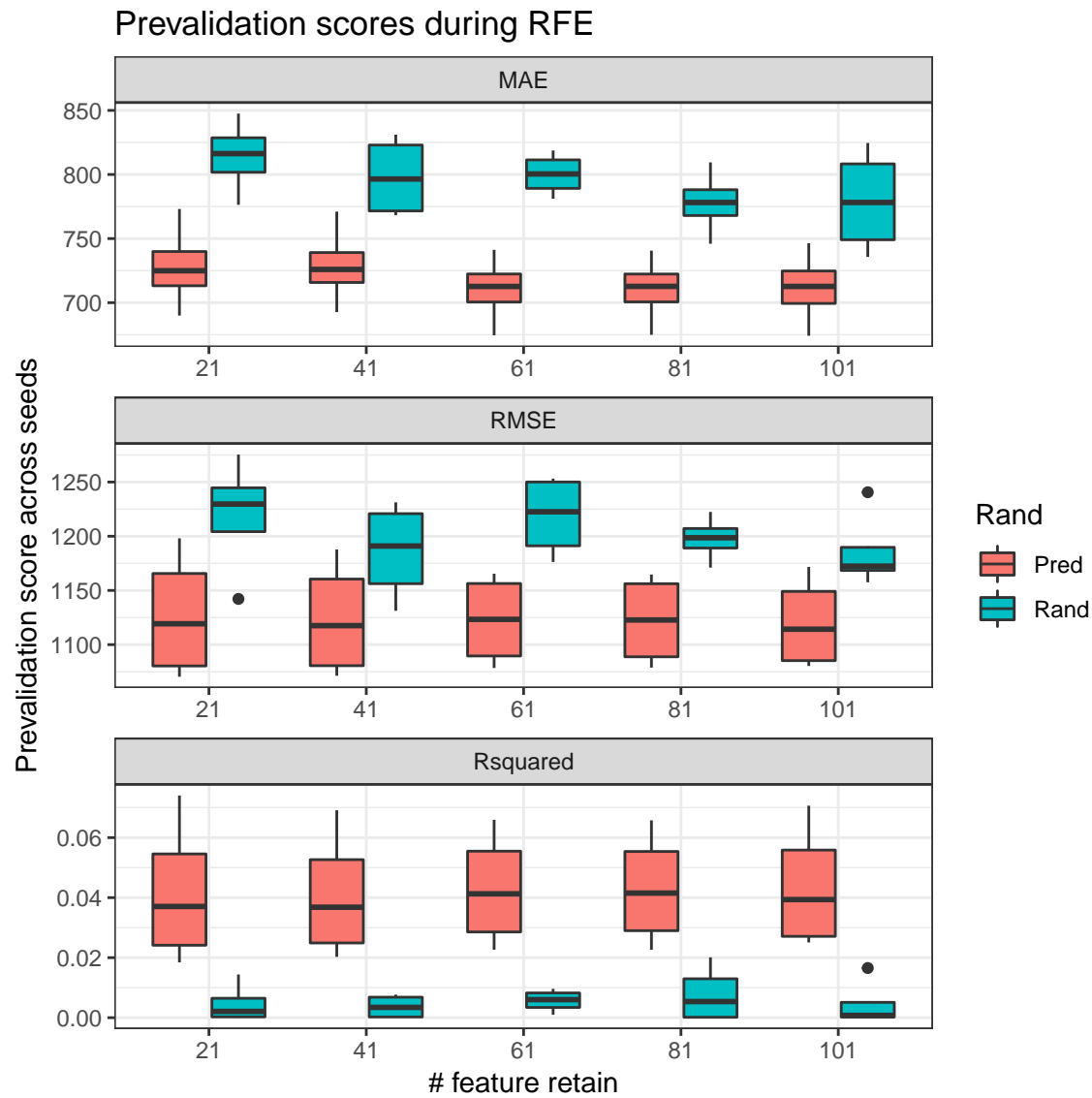
```
## user input
project_home <- "~/EVE/examples"
project_name <- "xgboostR_regression_1"
```

0. Load Data

```
## 300 of samples were used
## 101 of full features
## 4 runs, each run contains 3 CVs.
## os_time :
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0.0   182.8   480.0   889.4  1221.2  7125.0
```

run with XGBoost.r with evaluation metric: default.

1. Scores

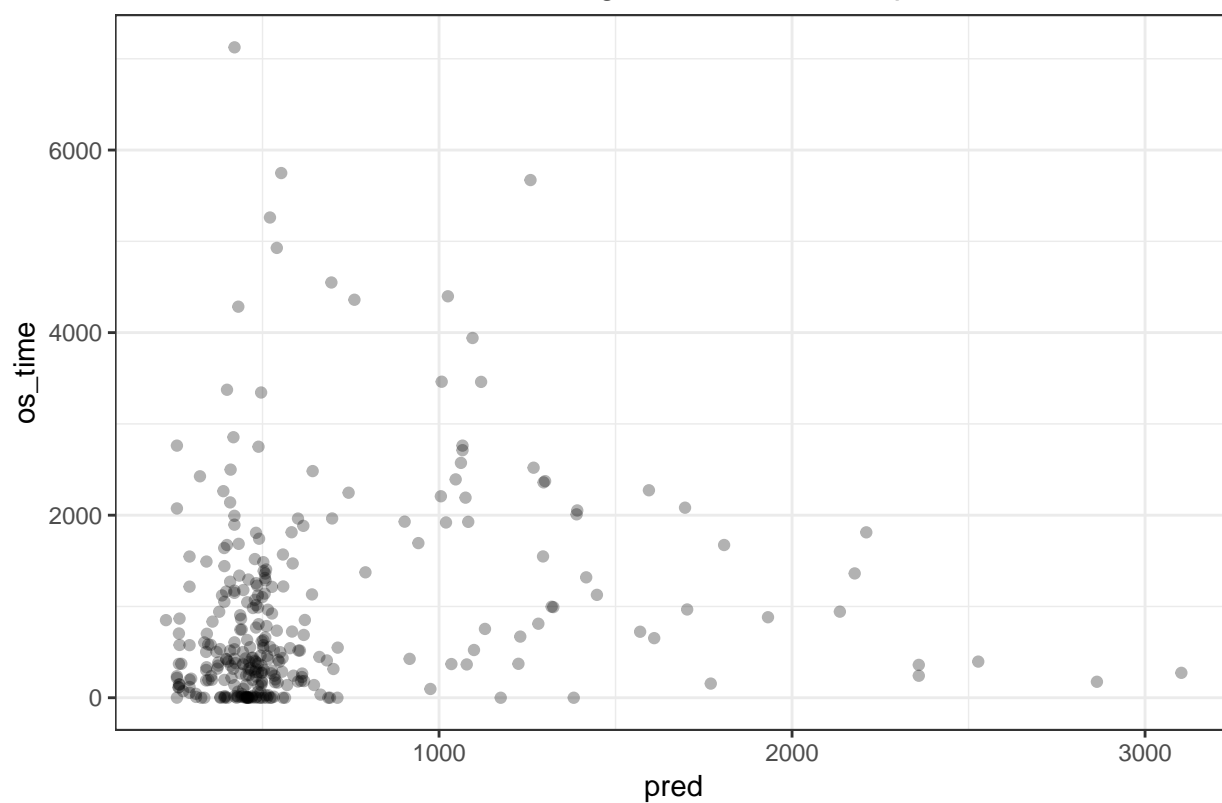


metrics	size.max	median.max	size.min	median.min
MAE	41	725.942	61	712.645
RMSE	61	1123.314	101	1114.231
Rsquared	81	0.042	41	0.037

correlation

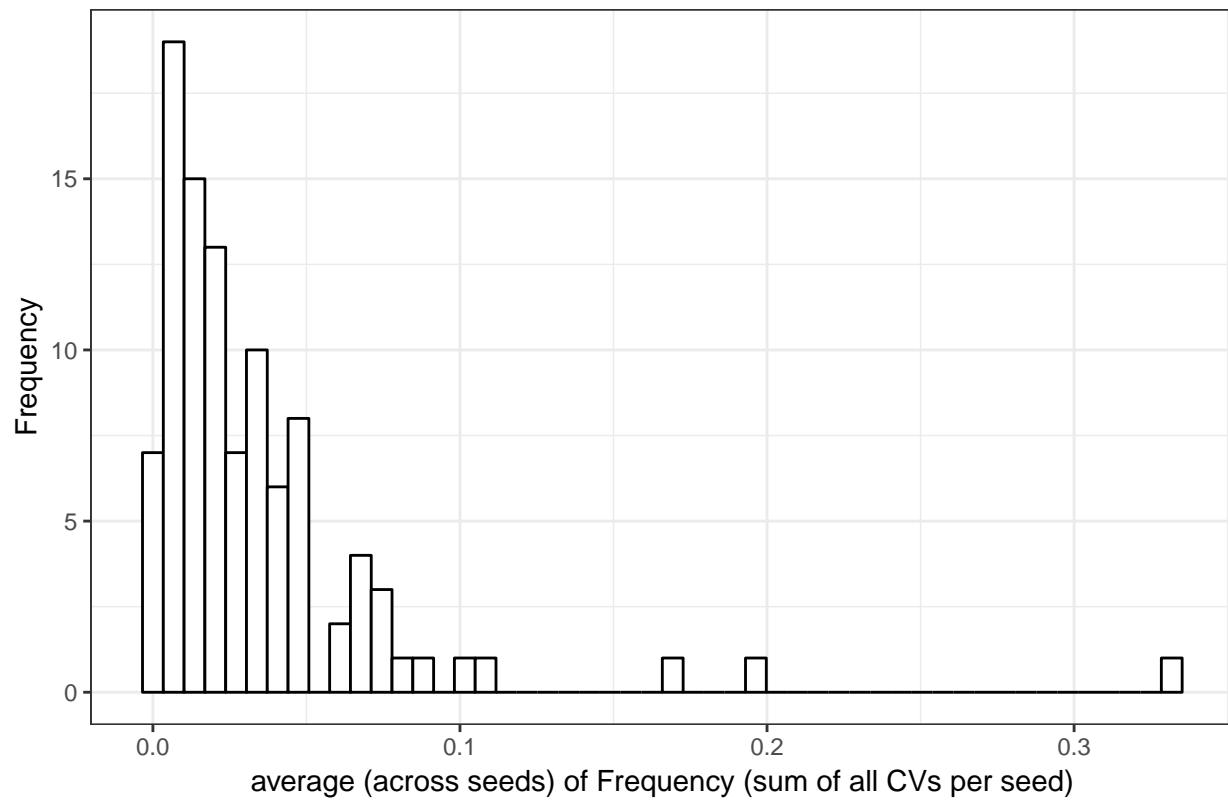
```
##
##
## Table: Averaged pearson correlation across seeds
##
##   cor.avg   cor.sdt
## -----
## 0.198039  0.0505366
```

Correlation at seed = 1001 using 101 feature set input

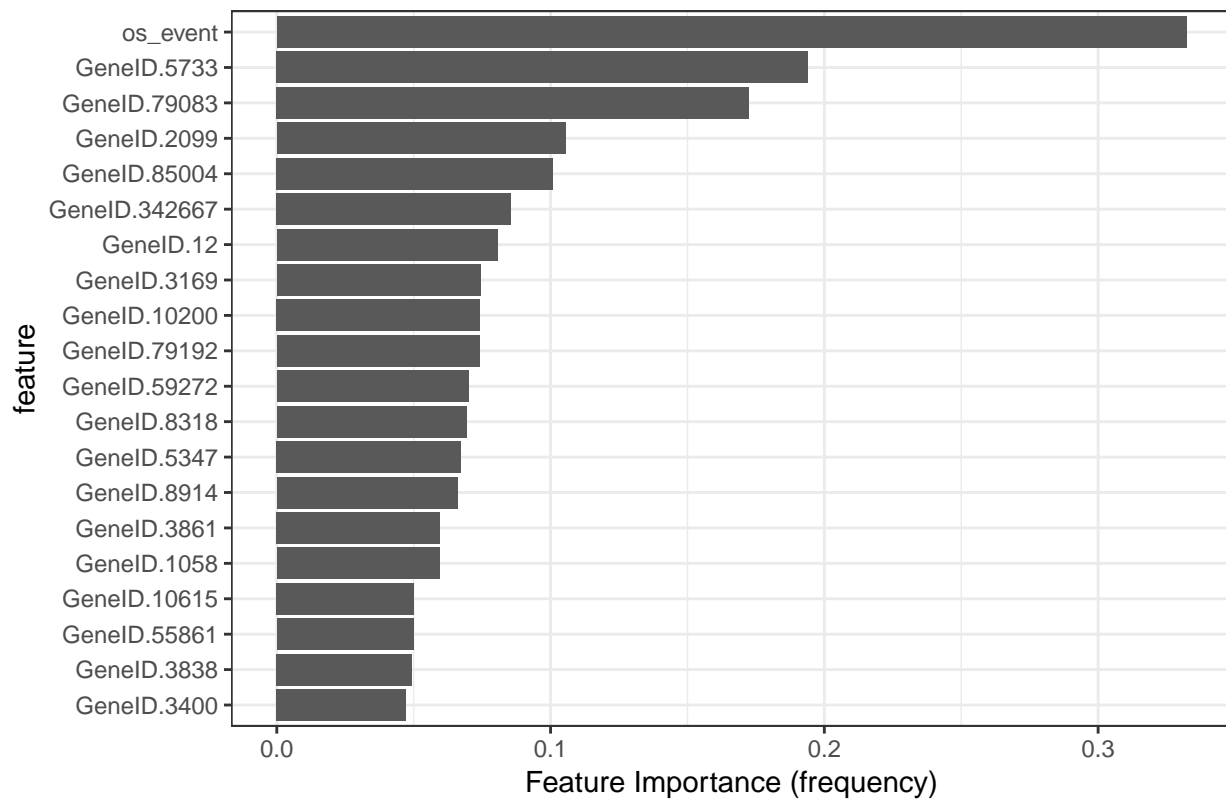


2. Important Features

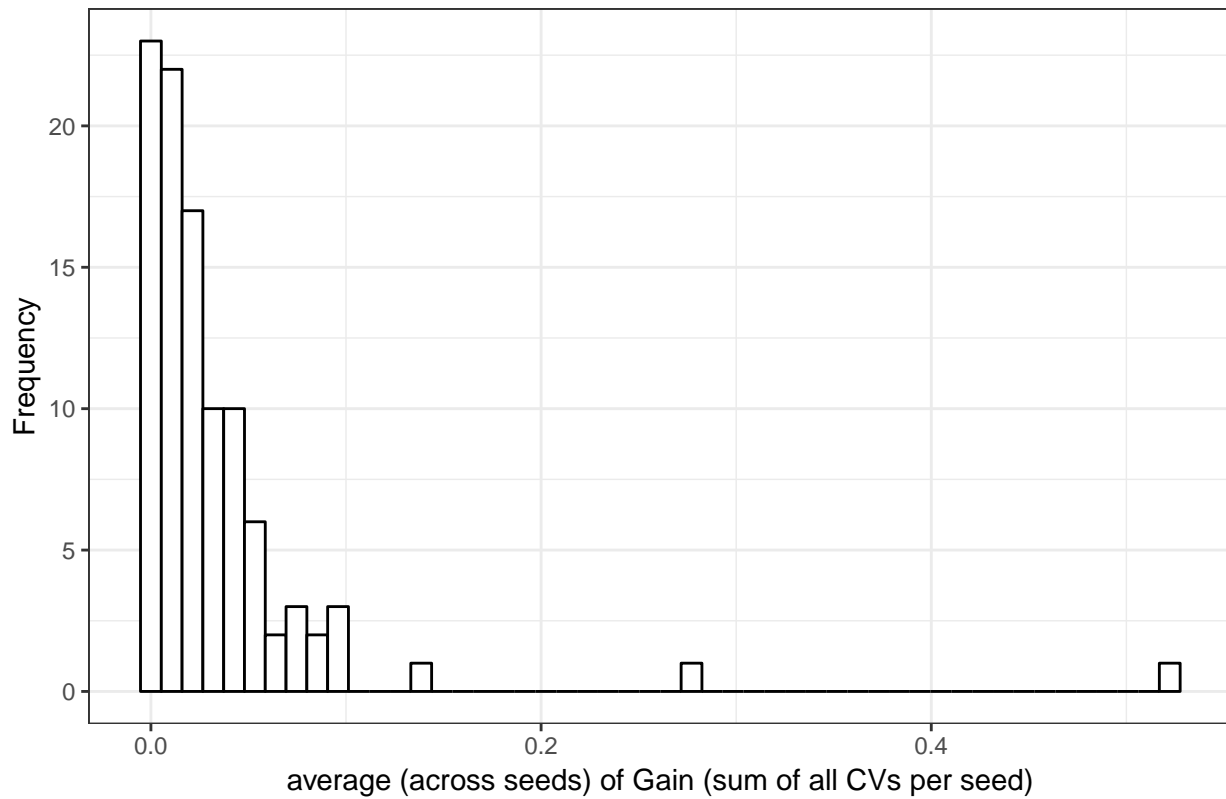
with 101 features based on Frequency



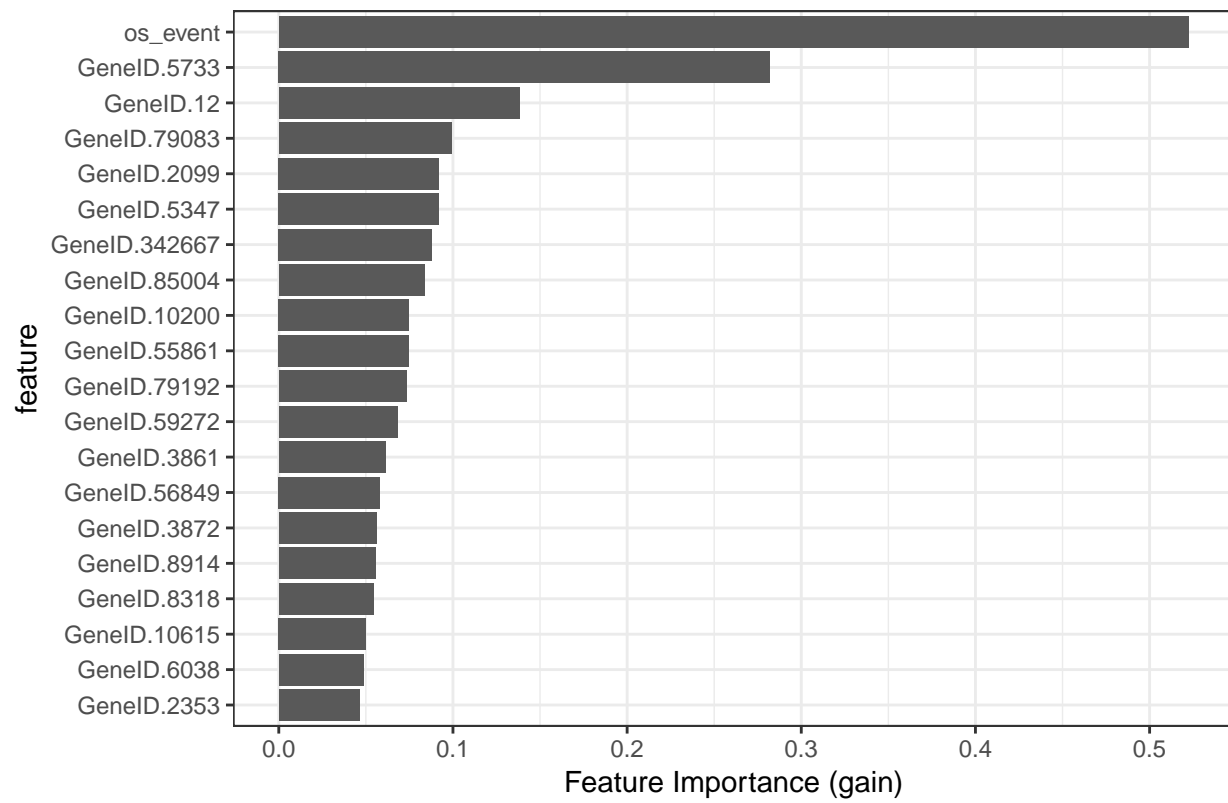
Top 20 features at 101 feature set based on Frequency

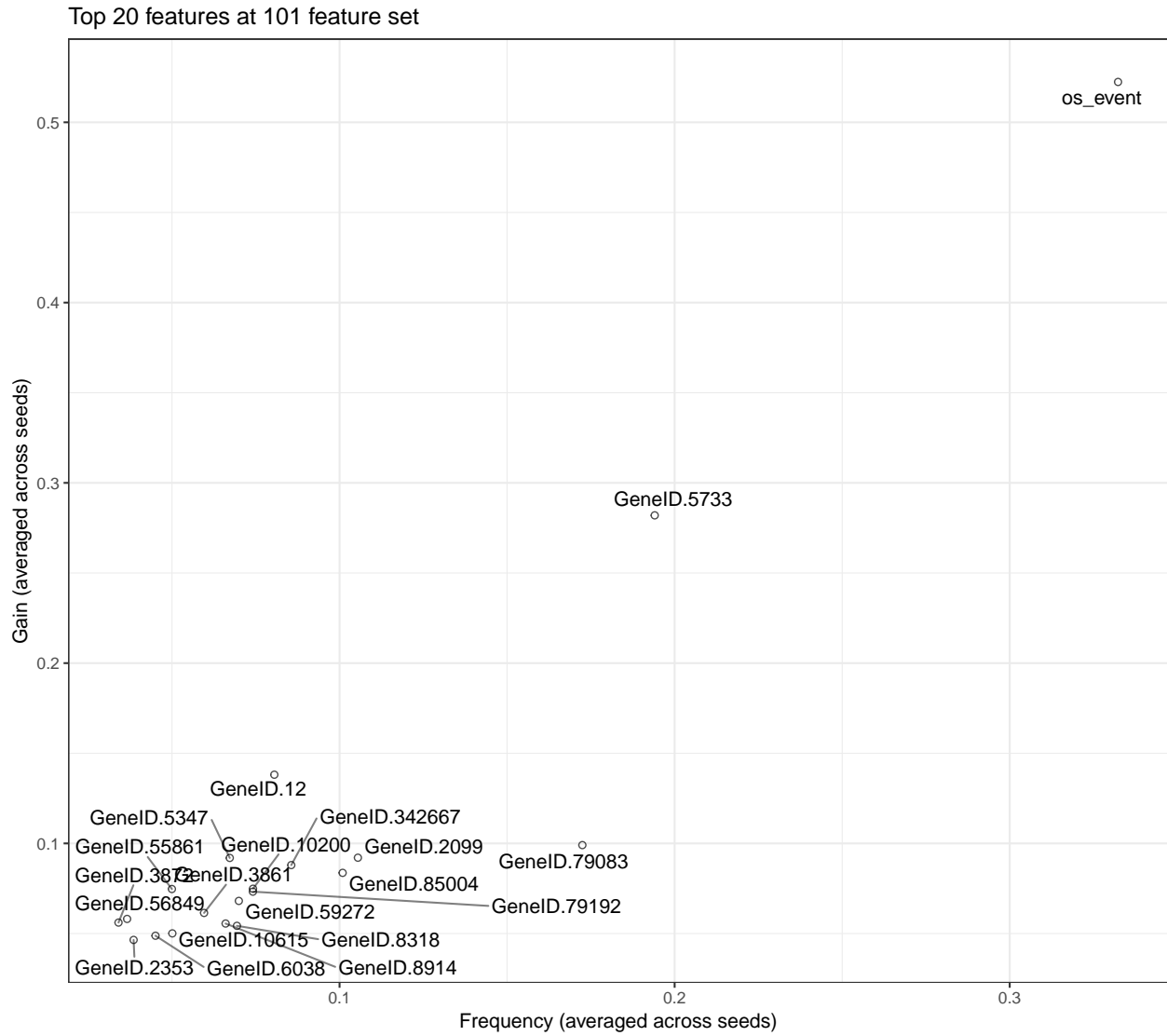


with 101 features based on Gain



Top 20 features at 101 feature set based on Gain

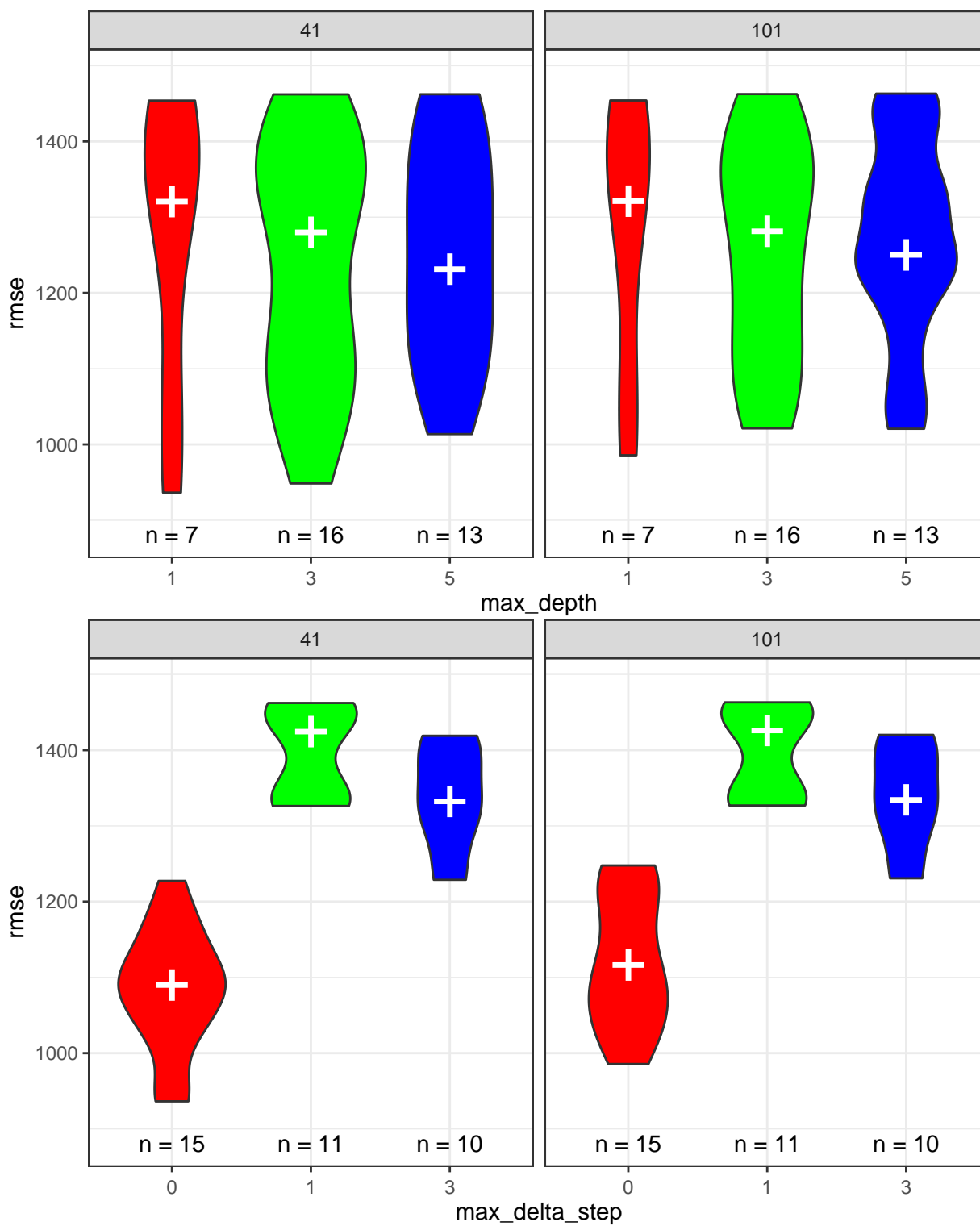


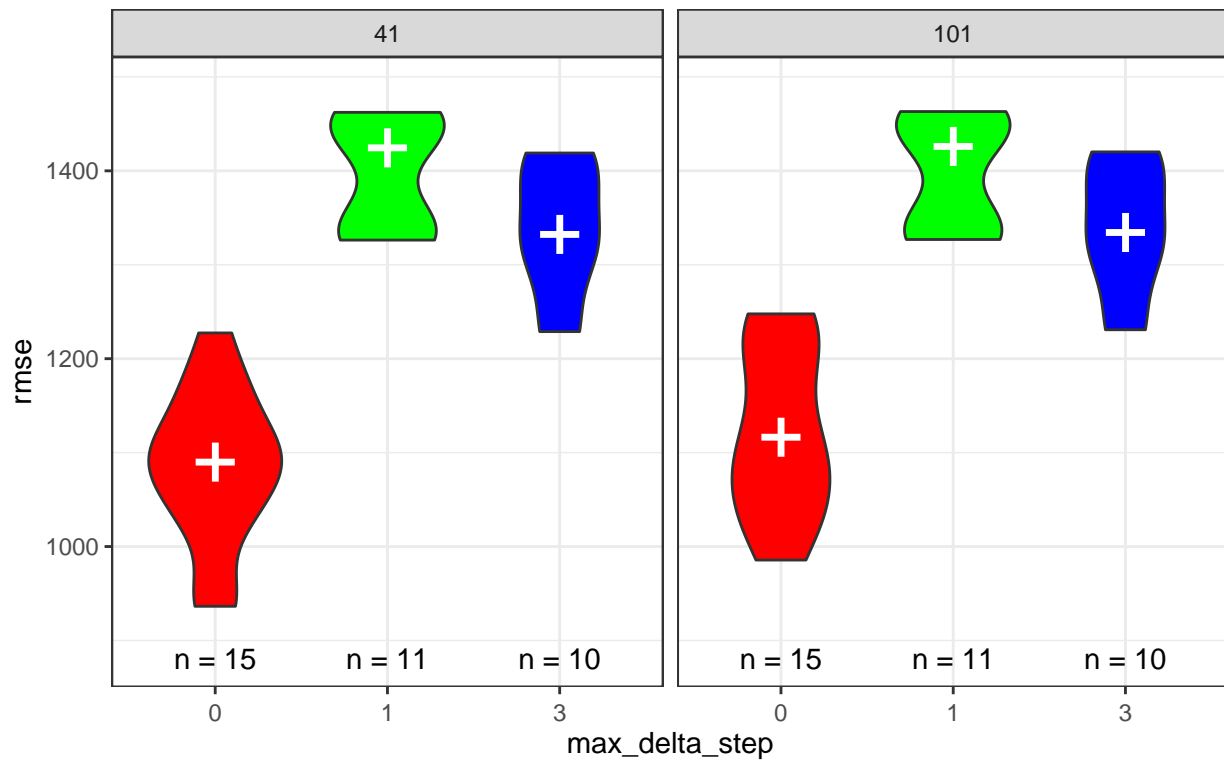


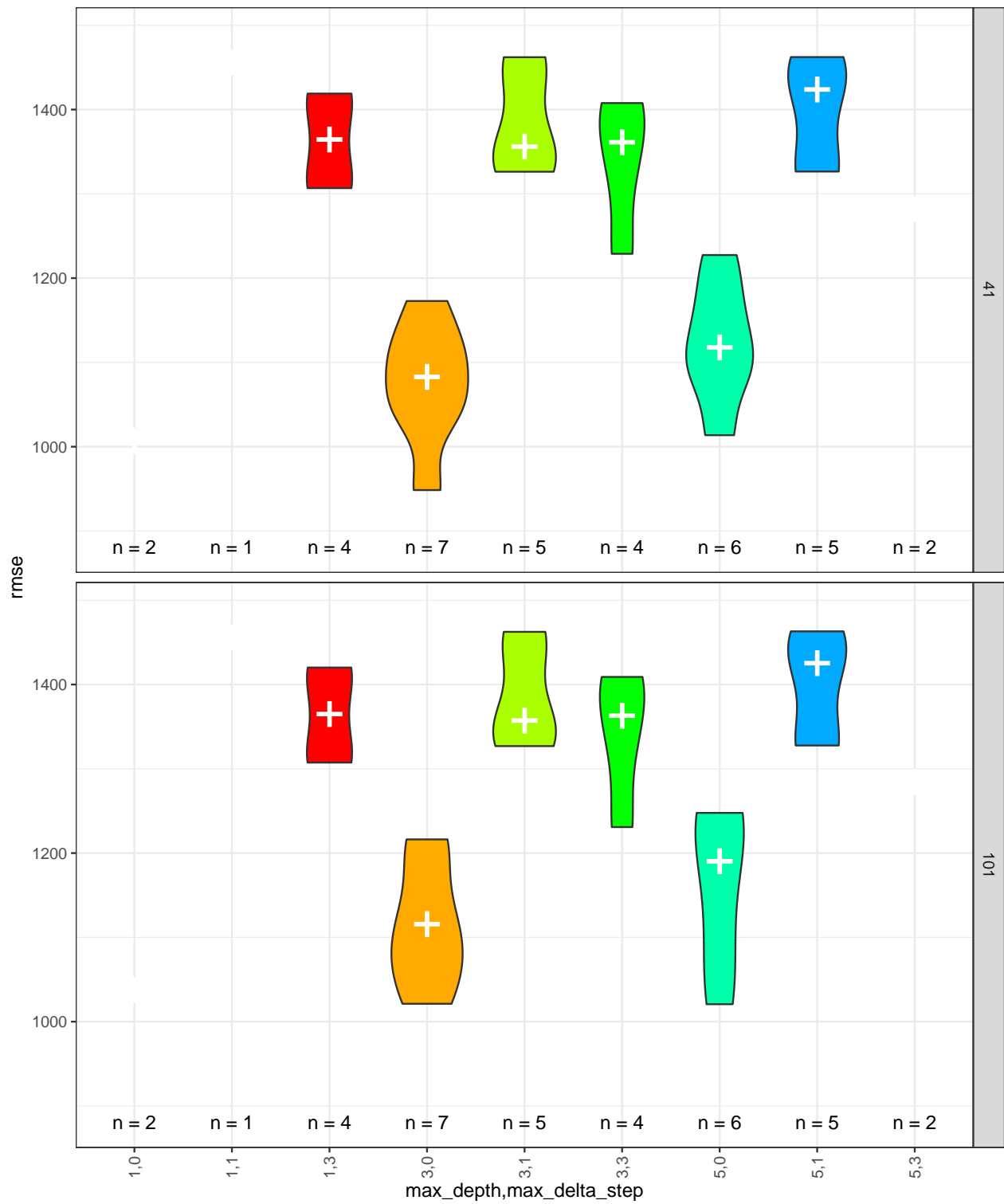
3. Hyper-parameters

parameter optimization file (72 records) includes 4 seeds. Each seed generates 3 cv splits. Within each cv split, there is a 2 step RFE (at 41, 101). So $72 / 4 / 3 / 2 = 3$ parameter combinations tried in each cv split.

all grid search results



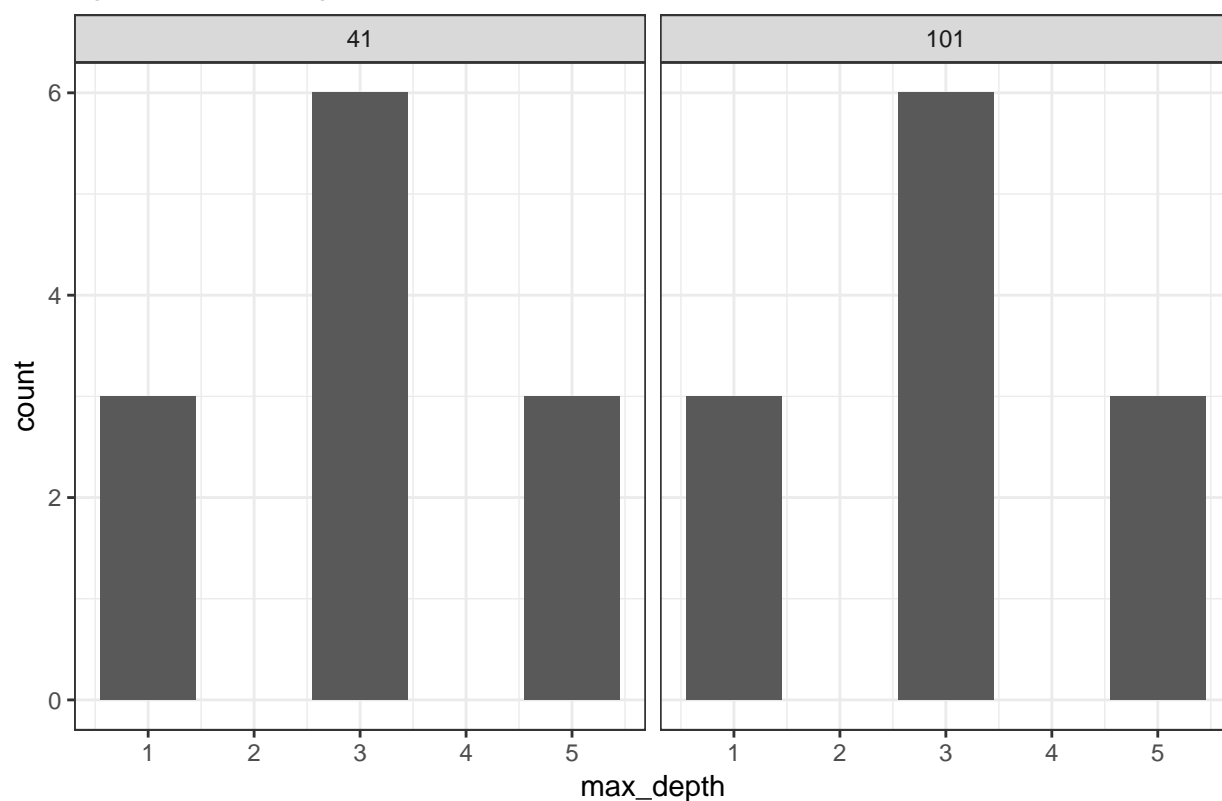




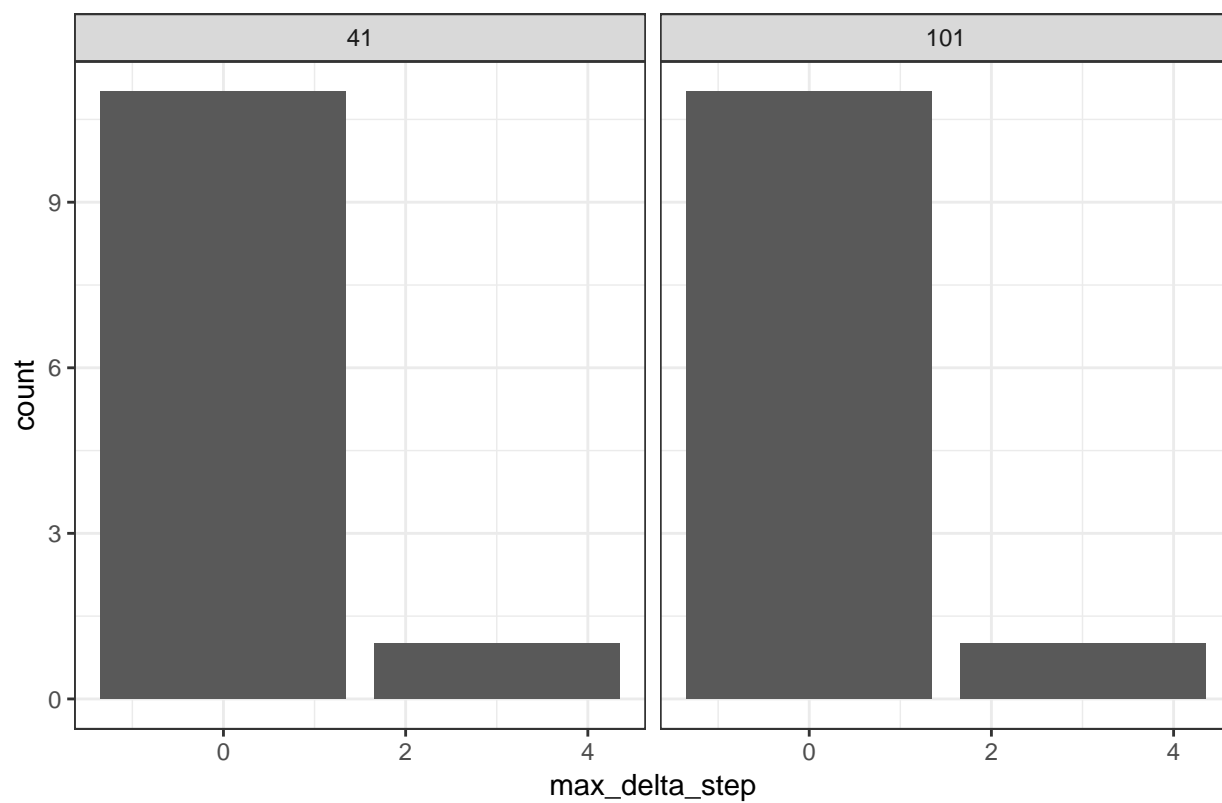
over best parameter combo per cv

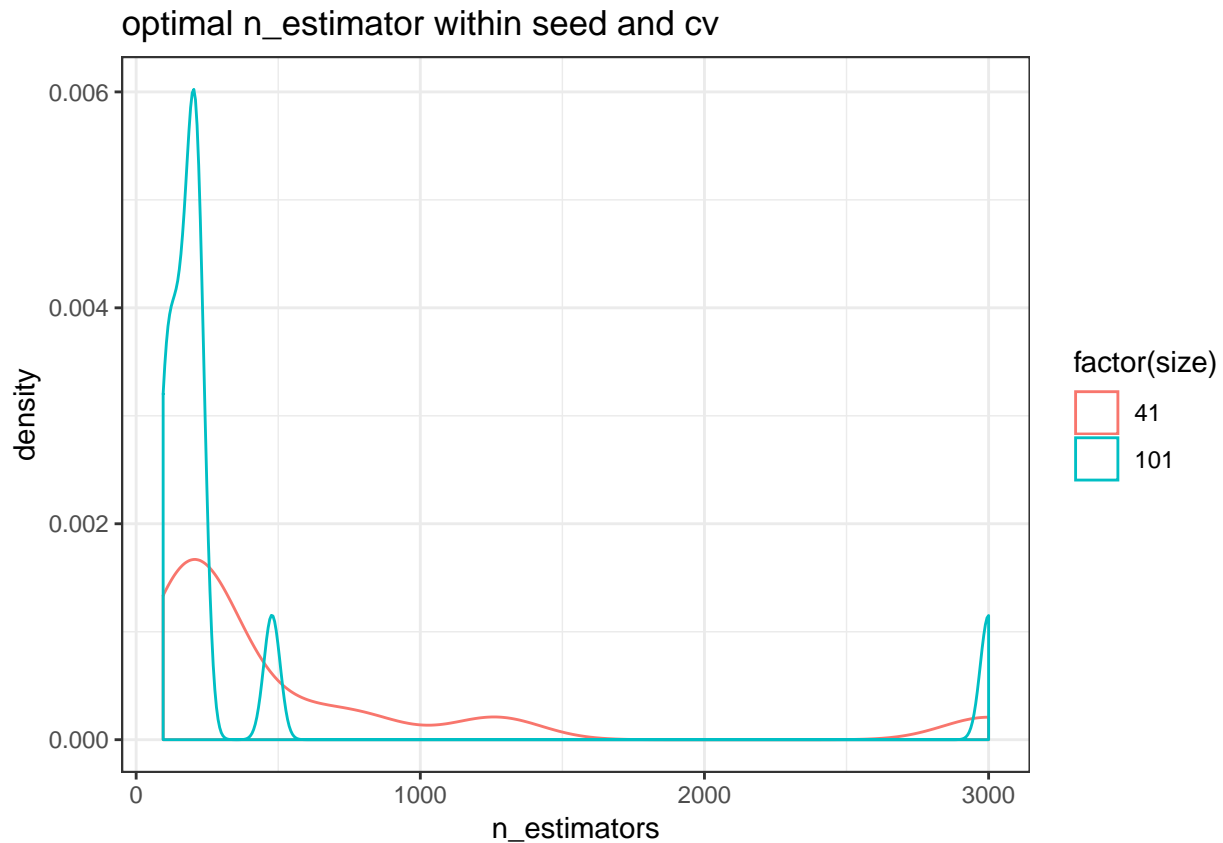
Note the 2nd /3rd best parameter combinations might not be too bad either.

optimal max_depth across seed and cv



optimal max_delta_step across seed and cv





more about the best parameter combination selection

```
select_ft_step <- 101

df1 <- subset(grid_best, size==select_ft_step & max_depth==5 & max_delta_step == 0 )
print( paste('summary of n estimator at',select_ft_step, 'feature step'))

## [1] "summary of n estimator at 101 feature step"
print(summary(df1$n_estimators))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      95.0  109.5   124.0   126.0   141.5   159.0

df2 <- subset(df.grid, size==select_ft_step & max_depth==5 & max_delta_step == 0 )

with(df2, plot(x = n_estimators, y=score, ylab=score_label))
```

