GAA EVAL.R

Changhua Yu

4/2/2019

A demo for GAA Submission evaluation package

```
source("../GAA-EVAL.R")
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
## Warning: package 'pROC' was built under R version 3.4.4
```

```
## Warning: package 'reshape2' was built under R version 3.4.3
```

```
## Warning: package 'ggrepel' was built under R version 3.4.4
```

```
## Warning: package 'gmodels' was built under R version 3.4.4
```

```
## Warning: package 'stringr' was built under R version 3.4.4
```

Read in the Experimental data provided by CAGI

```
exp.data <- read.RealData(file = "exp_data.csv", sep = ",",
                          col.id = 2, col.value = 5, col.sd = 6)
# inspect the experimental value
head(exp.data$value)
# inspect the experimental sd
head(exp.data$sd)
```

Read in the submission folders

```
sub.data <- read.Submission.Folder(folder.name = "prediction/",col.id = 1,
                                   col.value = 2, col.sd = 3, real.data = exp.data)
head(sub.data$value)
# inspect the dimension
dim(sub.data$value)
```

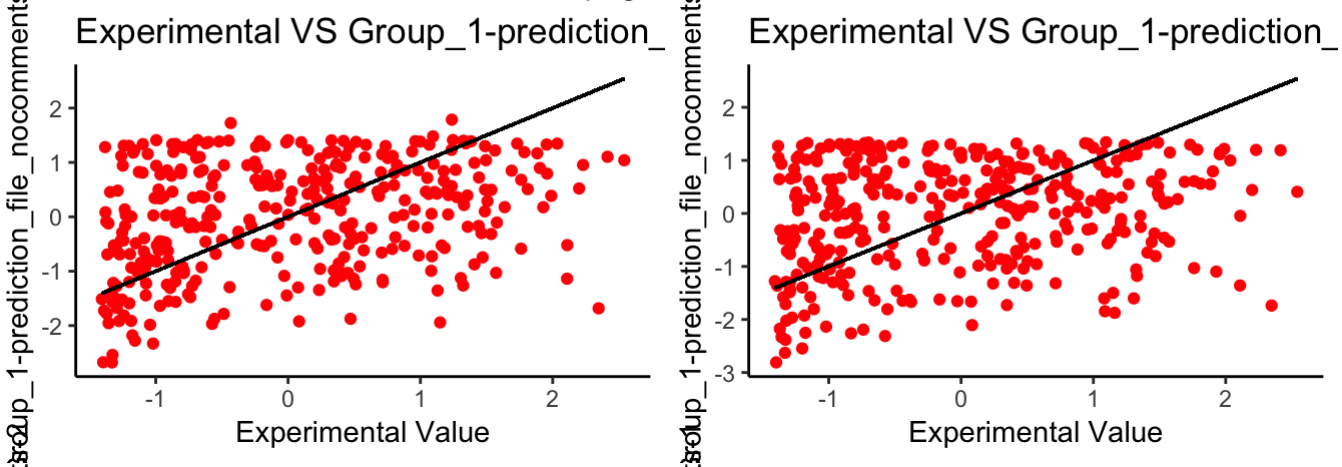
ScatterPlot inspection

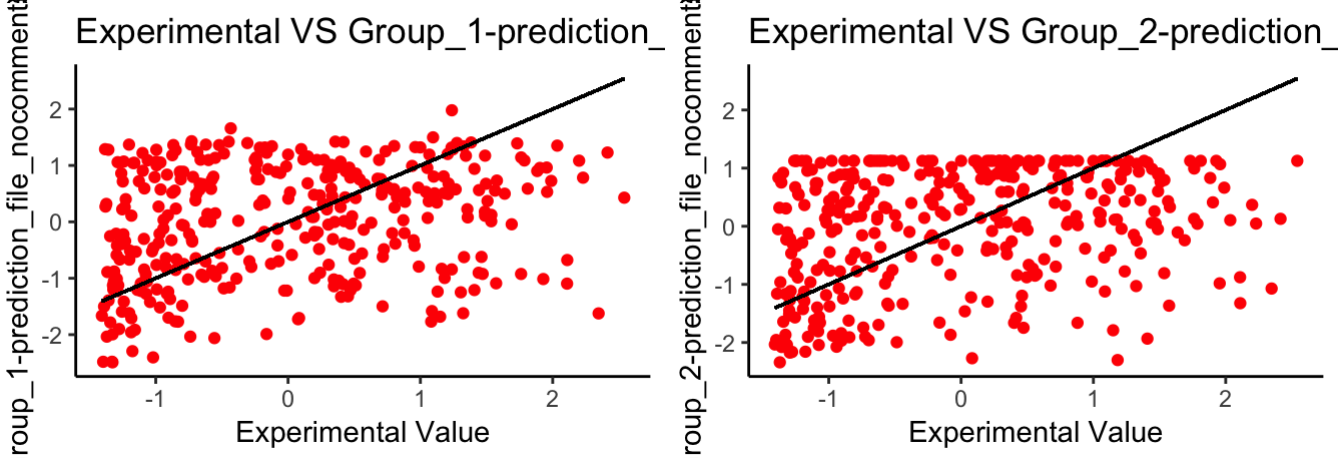
> Apply the z-score transformation for scatterplot to unify the scale of predicted value

```
plot_all_scatter(real.data = exp.data, pred.data = sub.data, z.transform = TRUE)
```

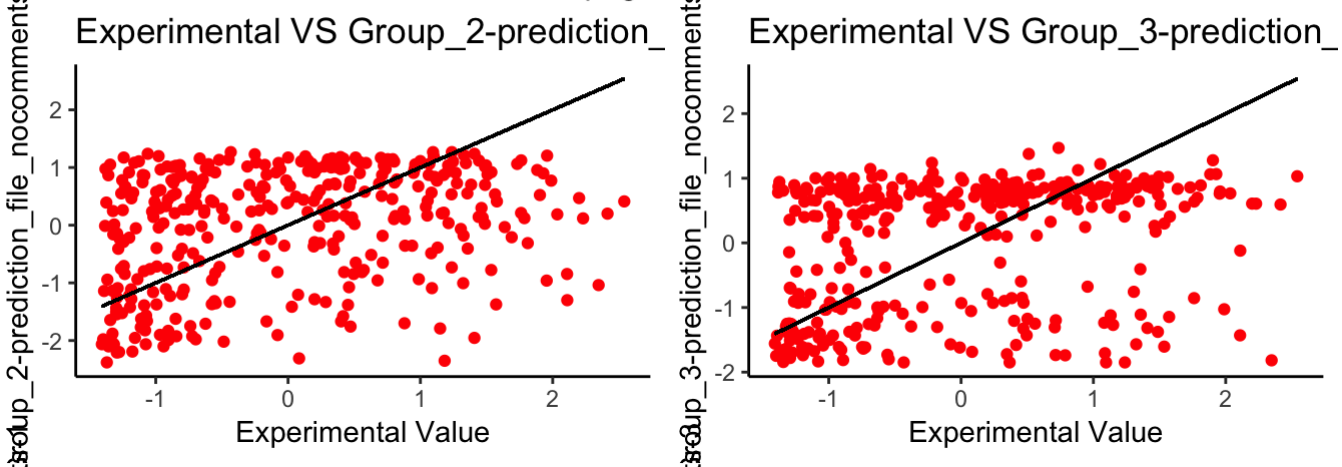
```
## Warning: Removed 4 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).
```

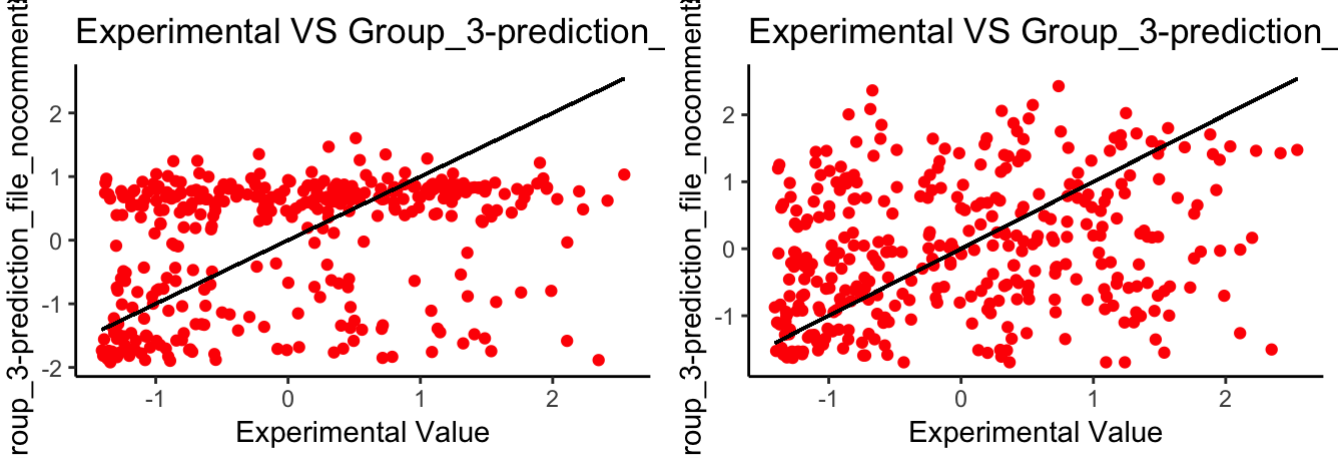
page 1 of 7



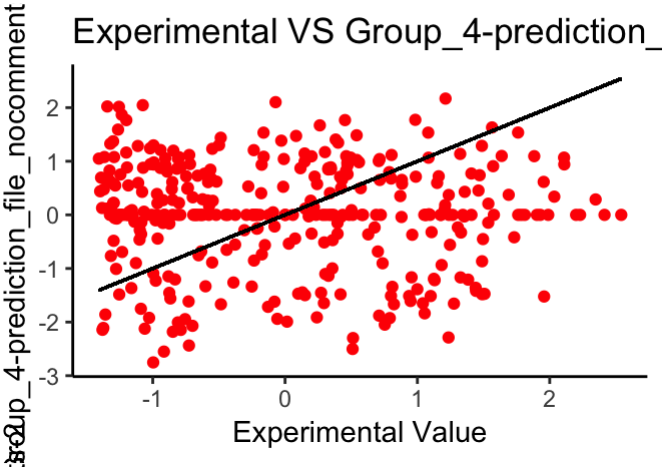


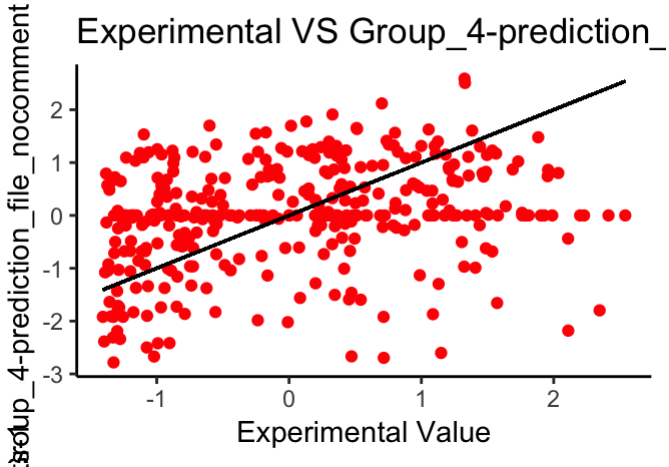
page 2 of 7

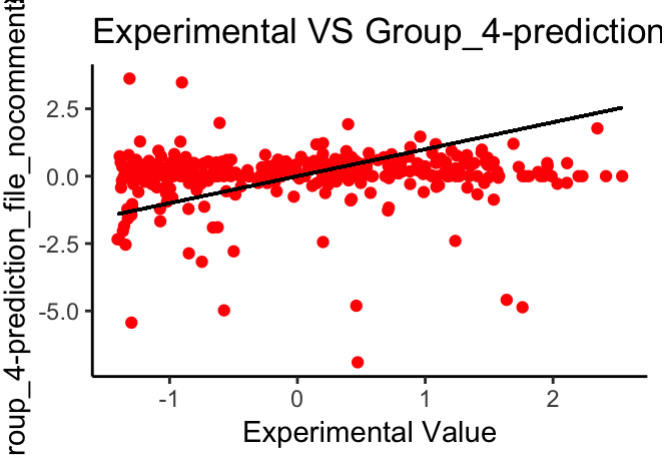


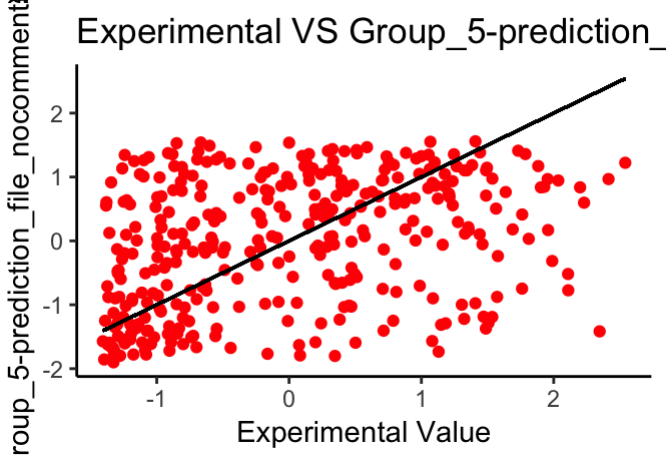


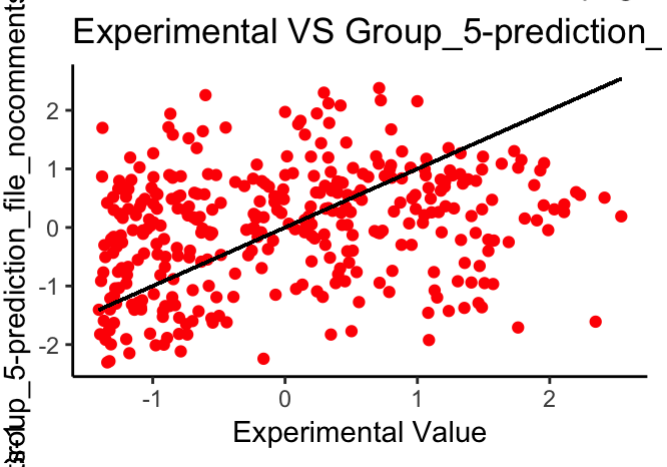
page 3 of 7

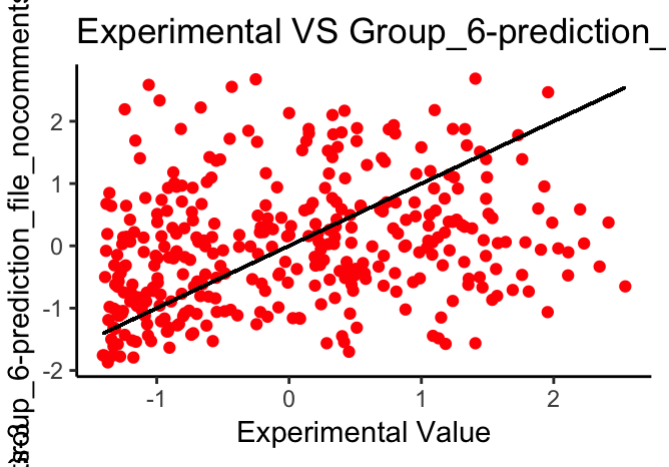

Experimental VS Group_4-prediction_
Experimental Value
Group_4-prediction_file_nocomment

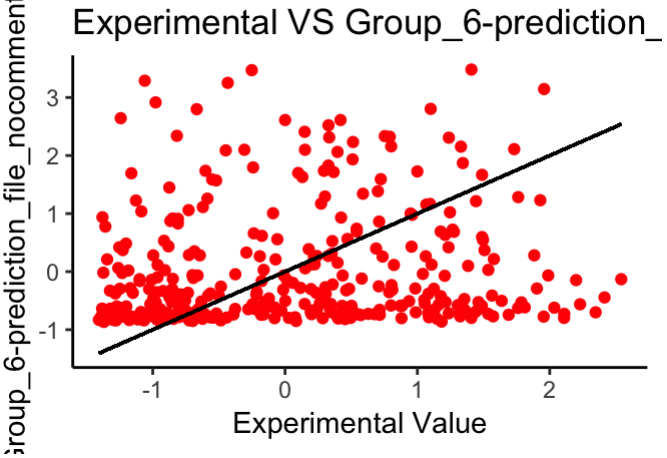

Experimental VS Group_4-prediction_
Experimental Value
Group_4-prediction_file_nocomment

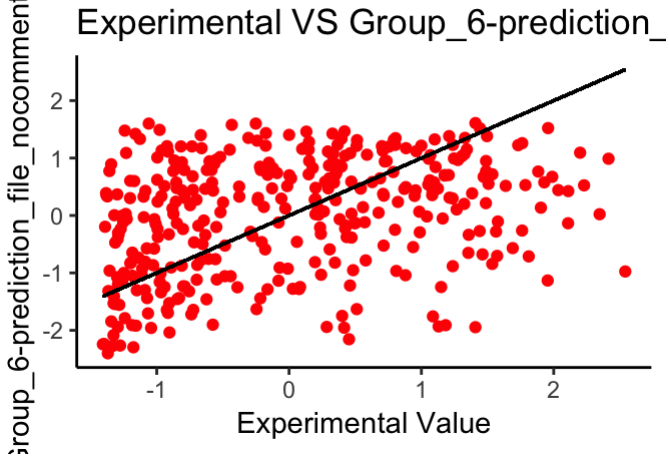

Experimental VS Group_4-prediction
Experimental Value
Group_4-prediction_file_nocomments-2


Experimental VS Group_5-prediction_
Experimental Value
Group_5-prediction_file_nocomments-2

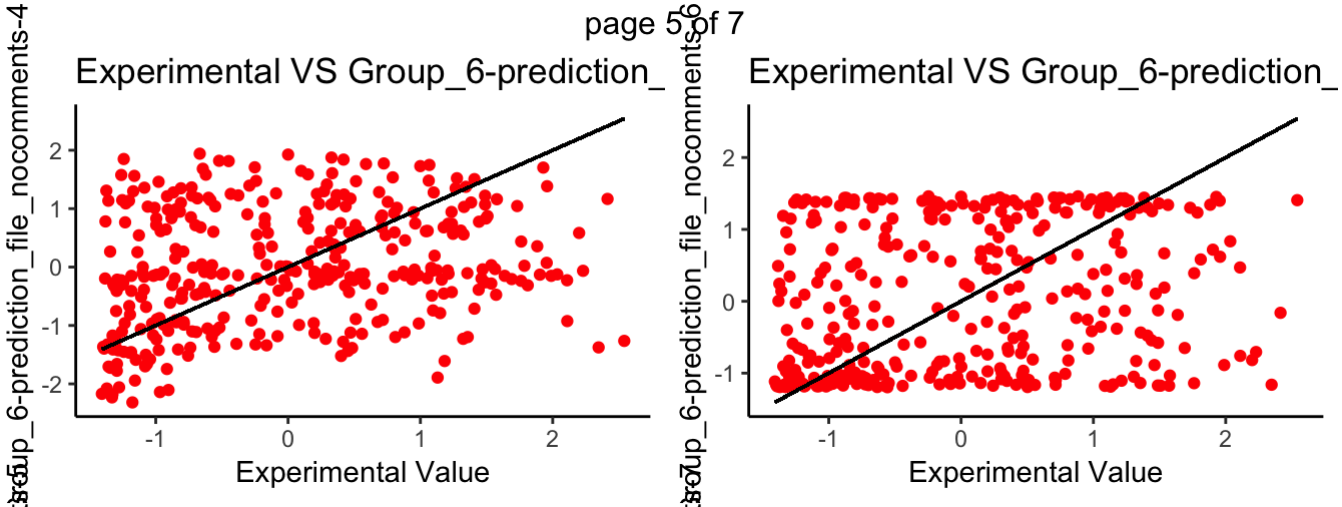

Experimental VS Group_5-prediction_
Experimental Value
Group_5-prediction_file_nocomments-1

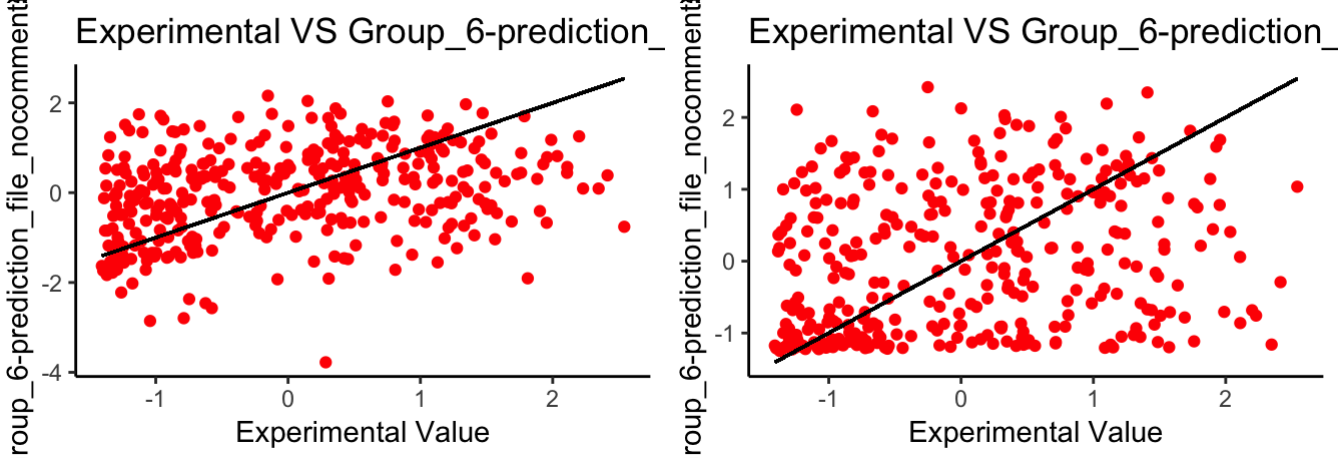

Experimental VS Group_6-prediction_
Experimental Value
Group_6-prediction_file_nocomments-3


Experimental VS Group_6-prediction_
Experimental Value
Group_6-prediction_file_nocomments-1

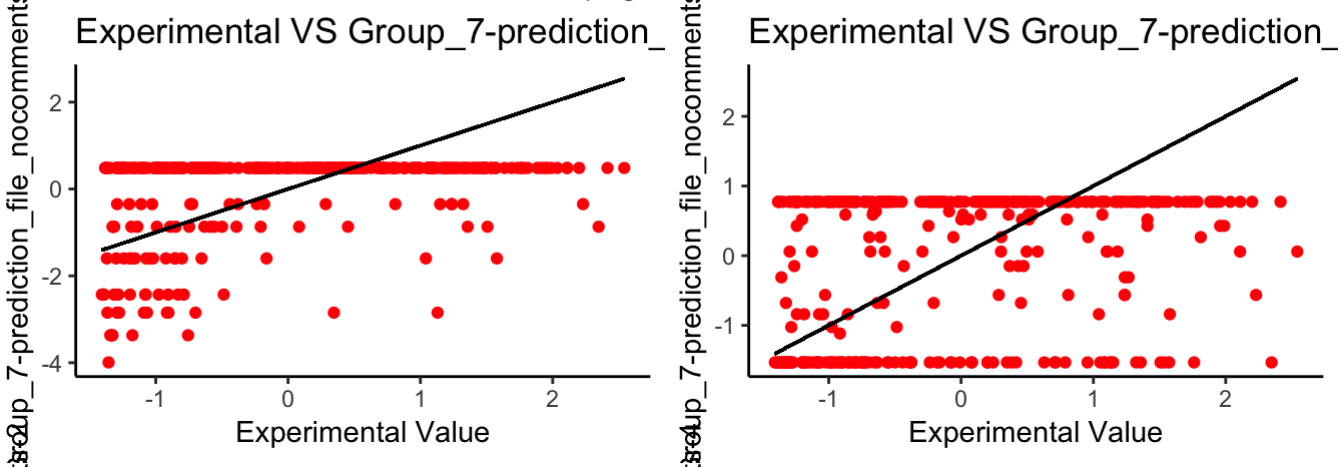

Experimental VS Group_6-prediction_
Experimental Value
Group_6-prediction_file_nocomments-2

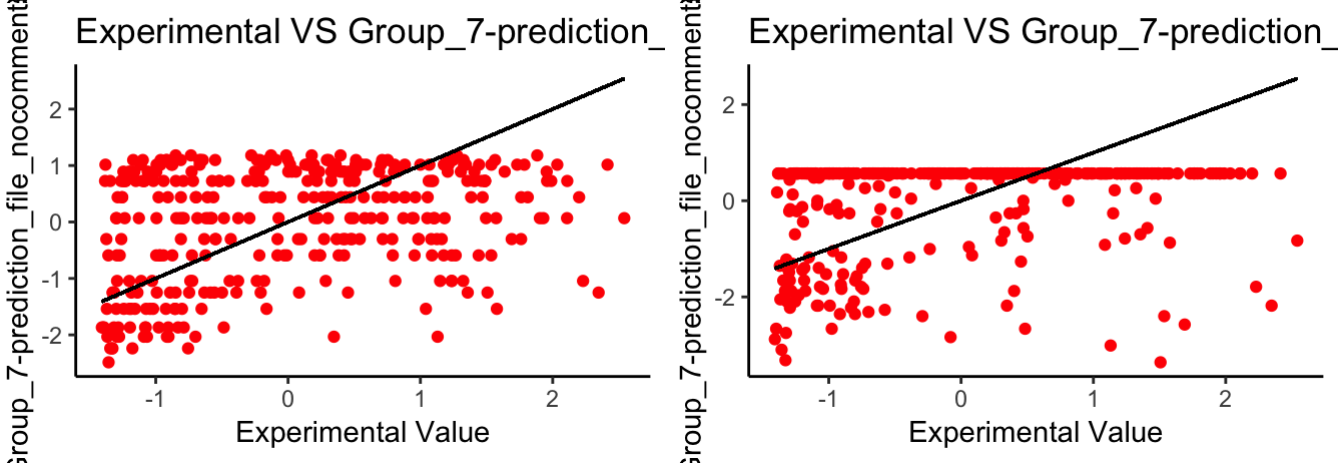
page 5 of 7

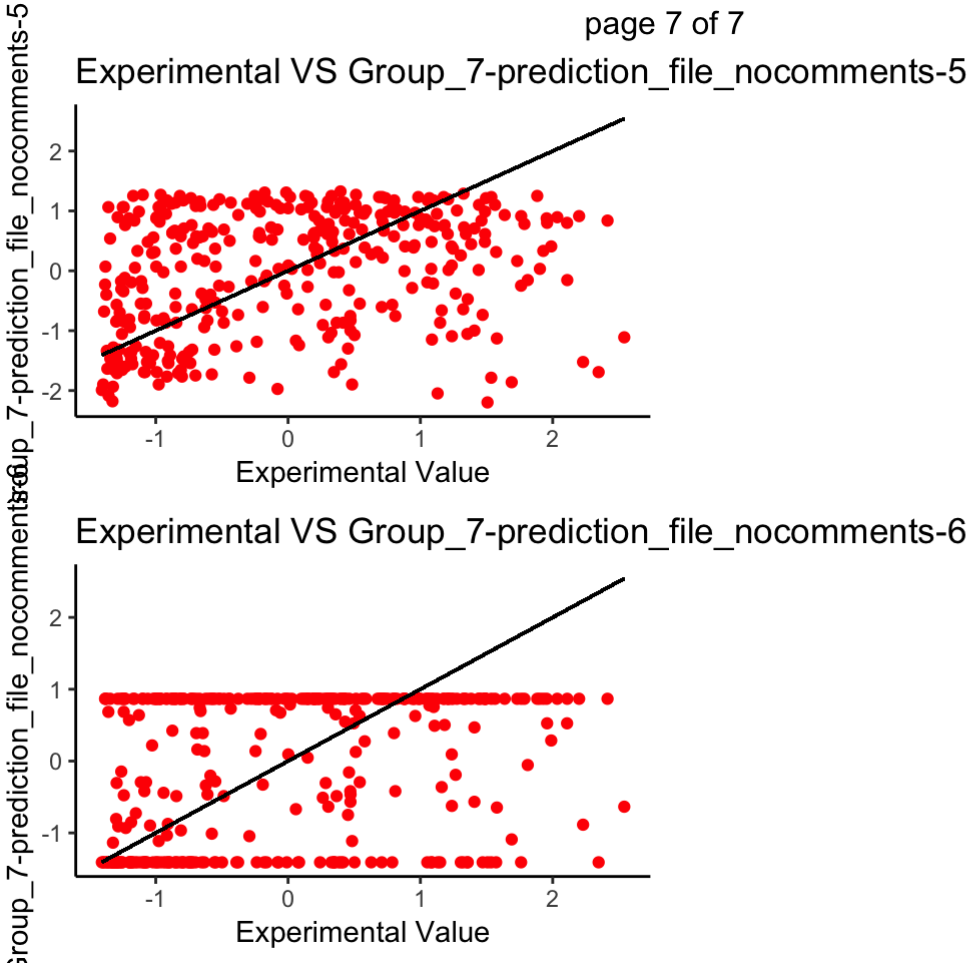




page 6 of 7







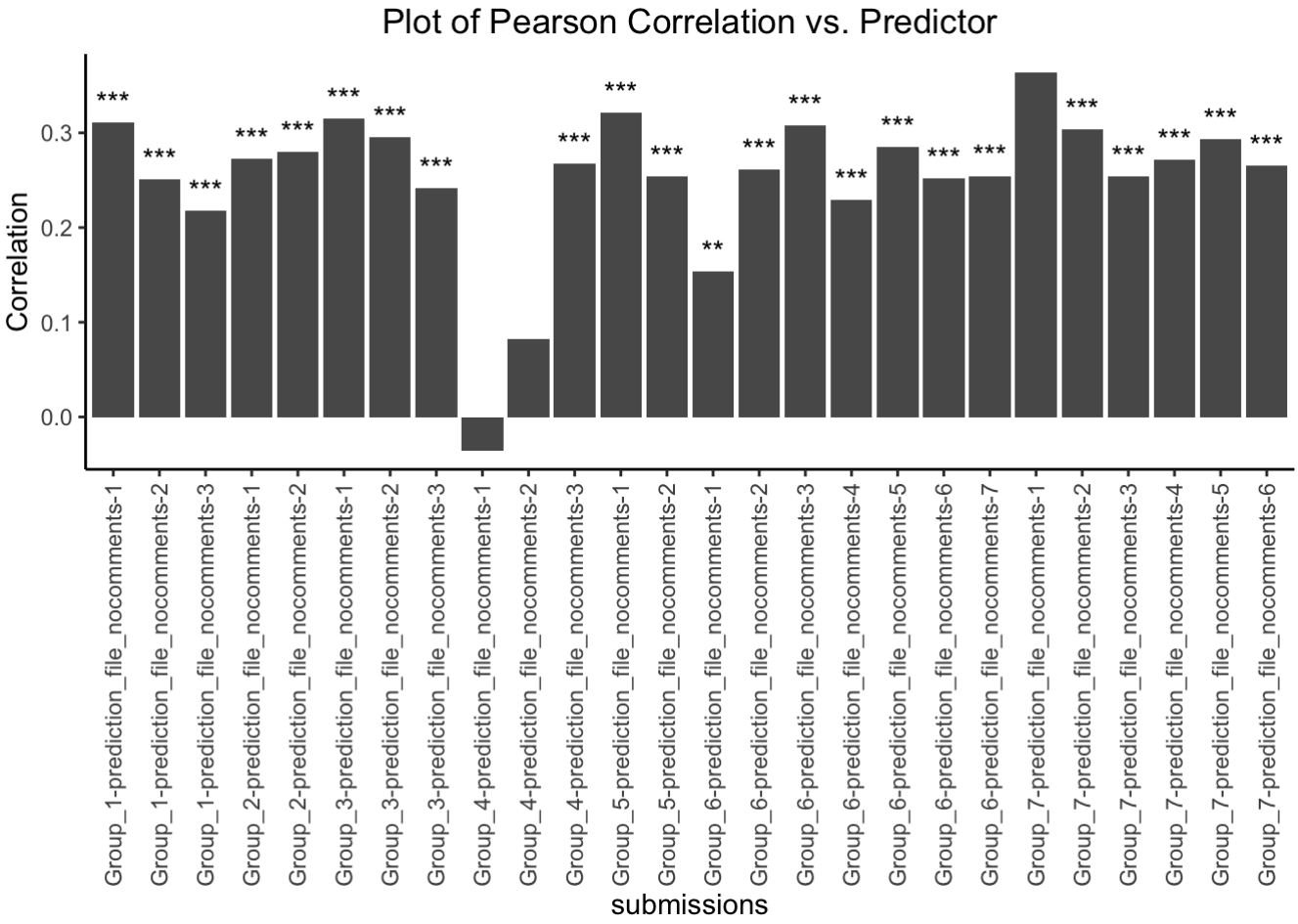
Correlation-based Evaluation without/with bootstrap

> without bootstrap method = “pearson”: In this example, we use pearson correlation
> sd.use = 0.3: experimental value with sd larger than 0.3 is filtered out
> z.transformation does not have effect to pearson correlation

```
# 1. Render coefficient value
result.cor.pearson <- eval.Correlation(real.data = exp.data, pred.data = sub.data,
                                       method = "pearson", sd.use = 0.3,z.transform = TR
UE)
head(result.cor.pearson)
```

```
##                                       pearson.coefficient.n=355.sd<0.3
## Group_1-prediction_file_nocomments-1                            0.3108669
## Group_1-prediction_file_nocomments-2                            0.2509373
## Group_1-prediction_file_nocomments-3                            0.2175485
## Group_2-prediction_file_nocomments-1                            0.2723800
## Group_2-prediction_file_nocomments-2                            0.2800299
## Group_3-prediction_file_nocomments-1                            0.3149842
##                                            p.value
## Group_1-prediction_file_nocomments-1 2.157713e-09
## Group_1-prediction_file_nocomments-2 1.683052e-06
## Group_1-prediction_file_nocomments-3 3.563631e-05
## Group_2-prediction_file_nocomments-1 1.859902e-07
## Group_2-prediction_file_nocomments-2 8.082396e-08
## Group_3-prediction_file_nocomments-1 1.287089e-09
```

```
# 2. Plot Correlation
plot.Correlation(result.cor.pearson, "Pearson")
```

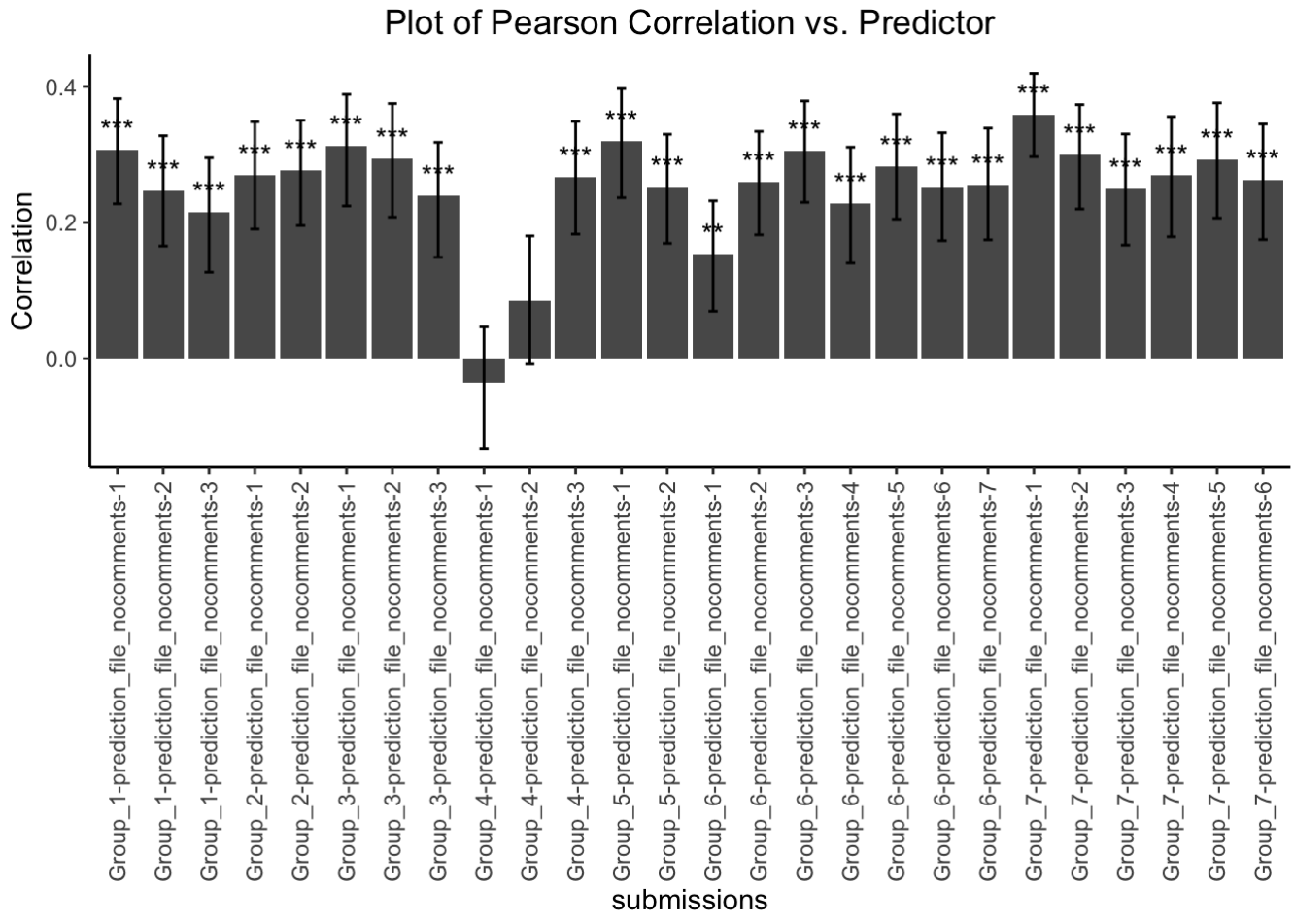


Bootstrap using row replacement (rep.time = 500 as a fixed input for now) boot = T, boot.var = F

```
# 1. Render coefficient value
boot.result.cor.pearson <- eval.Correlation(real.data = exp.data, pred.data = sub.data,
                                            method = "pearson", sd.use = 0.3,z.transform = TR
UE,boot = T)
# For Correlation-based Evaluation, provide mean, CI, and median pval
head(boot.result.cor.pearson)
```

```
##                                                avg    low_ci   high_ci
## Group_1-prediction_file_nocomments-1 0.3066170 0.2275263 0.3821523
## Group_1-prediction_file_nocomments-2 0.2465266 0.1653698 0.3276328
## Group_1-prediction_file_nocomments-3 0.2148069 0.1269072 0.2951286
## Group_2-prediction_file_nocomments-1 0.2701282 0.1902947 0.3481628
## Group_2-prediction_file_nocomments-2 0.2771144 0.1954804 0.3503623
## Group_3-prediction_file_nocomments-1 0.3118958 0.2242984 0.3884650
##                                                 sd      p.value
## Group_1-prediction_file_nocomments-1 0.04901241 3.652800e-09
## Group_1-prediction_file_nocomments-2 0.05023731 2.990593e-06
## Group_1-prediction_file_nocomments-3 0.05269642 3.926904e-05
## Group_2-prediction_file_nocomments-1 0.04706952 2.759994e-07
## Group_2-prediction_file_nocomments-2 0.04686972 1.029971e-07
## Group_3-prediction_file_nocomments-1 0.05113794 1.483798e-09
```

```
# 2. Plot Correlation
plot.Correlation(boot.result.cor.pearson, "Pearson",boot = TRUE)
```

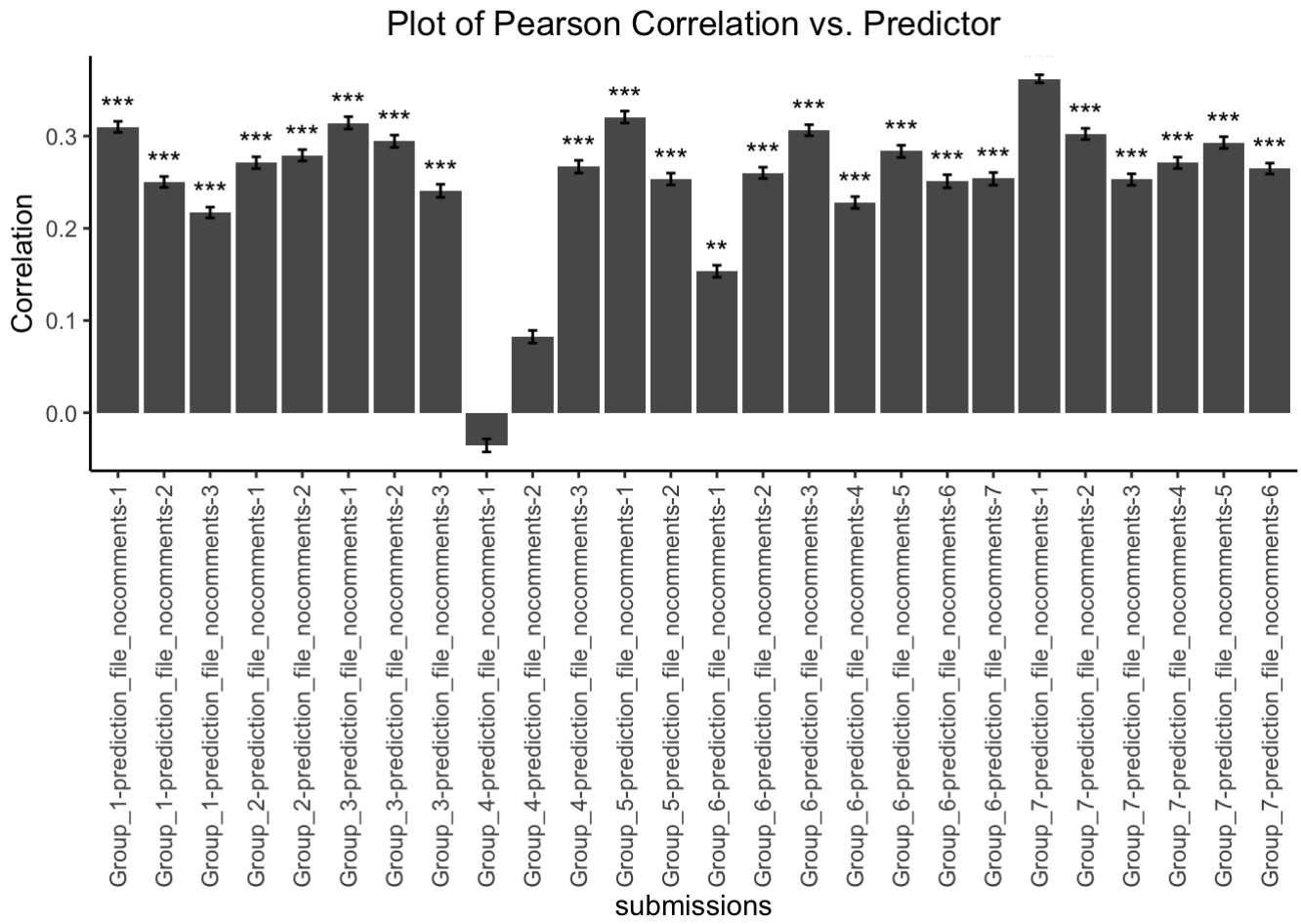


> Bootstrap using distribution modelling (rep.time = 500 as a fixed input for now) boot = T, boot.var = T

```
bootvar.result.cor.pearson <- eval.Correlation(real.data = exp.data, pred.data = sub.dat
a,
                                        method = "pearson", sd.use = 0.3,z.transform = TR
UE,boot = T,boot.var = T )
head(bootvar.result.cor.pearson)
```

```
##                                             avg    low_ci   high_ci
## Group_1-prediction_file_nocomments-1 0.3098081 0.3040298 0.3159479
## Group_1-prediction_file_nocomments-2 0.2500853 0.2443017 0.2562133
## Group_1-prediction_file_nocomments-3 0.2169032 0.2112245 0.2229135
## Group_2-prediction_file_nocomments-1 0.2714877 0.2648111 0.2774905
## Group_2-prediction_file_nocomments-2 0.2790994 0.2730558 0.2852922
## Group_3-prediction_file_nocomments-1 0.3140110 0.3076190 0.3210243
##                                                sd      p.value
## Group_1-prediction_file_nocomments-1 0.003568889 2.426110e-09
## Group_1-prediction_file_nocomments-2 0.003564544 1.828919e-06
## Group_1-prediction_file_nocomments-3 0.003507820 3.785398e-05
## Group_2-prediction_file_nocomments-1 0.003668964 2.067416e-07
## Group_2-prediction_file_nocomments-2 0.003569346 8.890819e-08
## Group_3-prediction_file_nocomments-1 0.004036481 1.443387e-09
```

```
plot.Correlation(bootvar.result.cor.pearson, "Pearson",boot = TRUE)
```



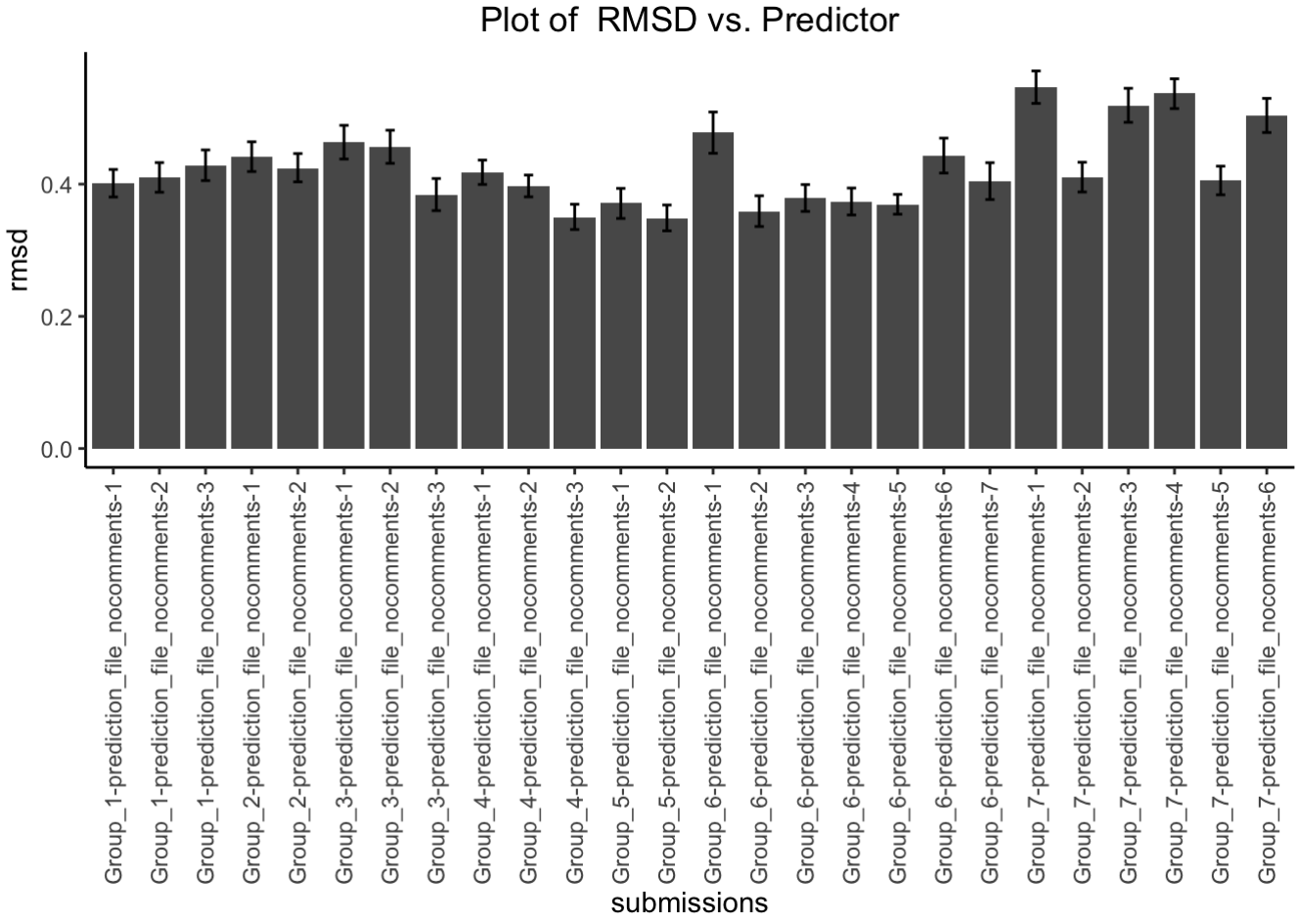
RMSD-based Evaluation

- (for demonstration, only present the bootstrapped result)
- variance.normalization: defined on PPT
- density.distance: RMSD based on distribution model of experimental value

```
# without variance.normalization + without density.distance
boot.result.rmsd4 <- eval.RMSD(real.data = exp.data, pred.data = sub.data,sd.use = NA,
                      density.distance = FALSE,variance.normalization = FALSE,boot = TRU
E)
head(boot.result.rmsd4)
```

```
##                                              RMSD    low_ci   high_ci
## Group_1-prediction_file_nocomments-1 0.4010775 0.3805267 0.4221817
## Group_1-prediction_file_nocomments-2 0.4104334 0.3877751 0.4325619
## Group_1-prediction_file_nocomments-3 0.4287069 0.4054087 0.4517231
## Group_2-prediction_file_nocomments-1 0.4410671 0.4190720 0.4640579
## Group_2-prediction_file_nocomments-2 0.4243437 0.4034699 0.4462245
## Group_3-prediction_file_nocomments-1 0.4631417 0.4380073 0.4889377
##                                                sd
## Group_1-prediction_file_nocomments-1 0.01360405
## Group_1-prediction_file_nocomments-2 0.01414572
## Group_1-prediction_file_nocomments-3 0.01421772
## Group_2-prediction_file_nocomments-1 0.01392664
## Group_2-prediction_file_nocomments-2 0.01319777
## Group_3-prediction_file_nocomments-1 0.01537482
```

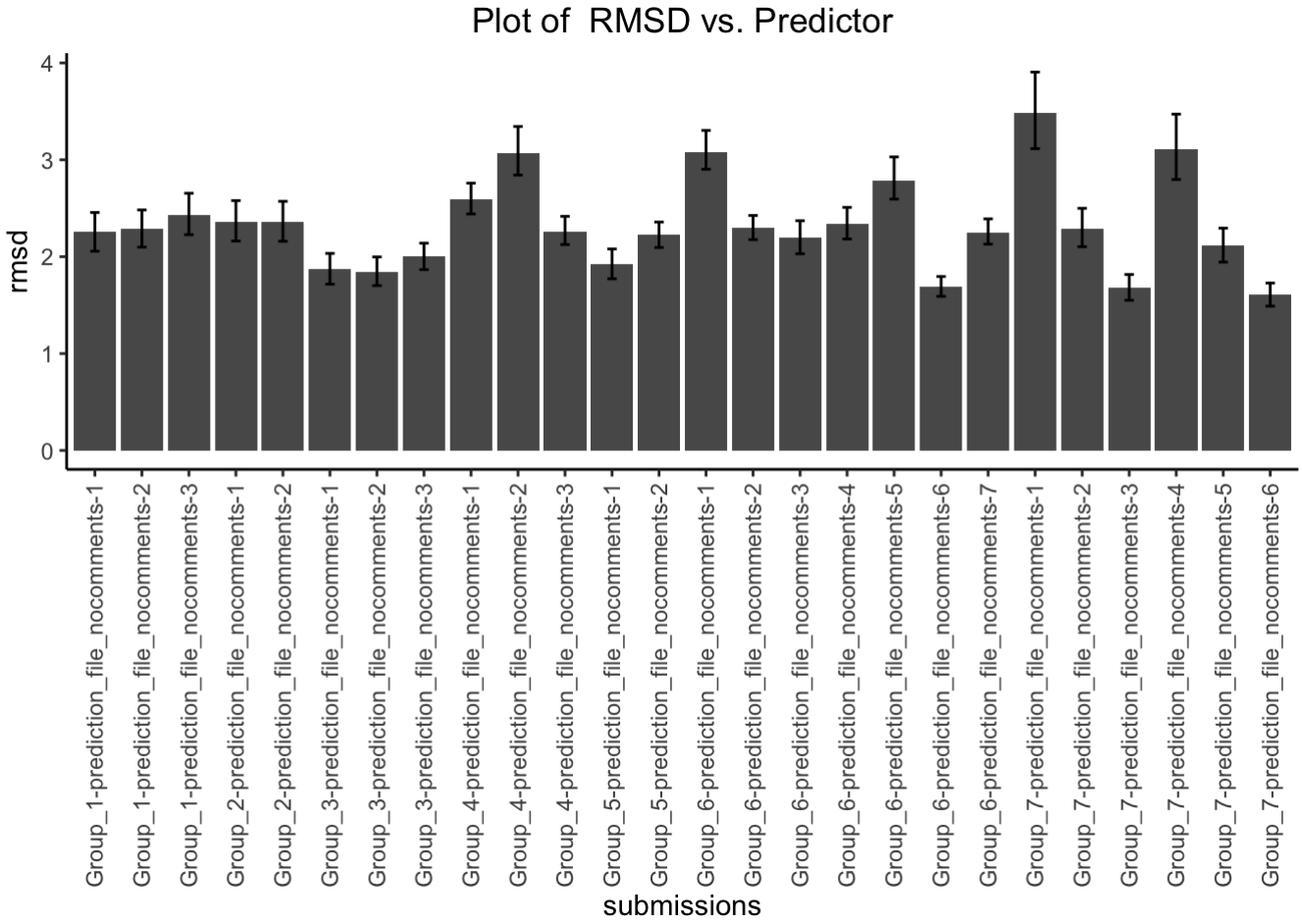
```
plot.RMSD(boot.result.rmsd4, method="",boot = TRUE)
```




```
# with variance.normalization + without density.distance
boot.result.rmsd2 <- eval.RMSD(real.data = exp.data, pred.data = sub.data,sd.use = NA,
                      density.distance = FALSE,variance.normalization = TRUE,boot = TRUE
)
head(boot.result.rmsd2)
```

```
##                                          RMSD   low_ci  high_ci        sd
## Group_1-prediction_file_nocomments-1 2.252916 2.056278 2.456234 0.1230616
## Group_1-prediction_file_nocomments-2 2.290883 2.098477 2.482414 0.1207208
## Group_1-prediction_file_nocomments-3 2.432993 2.226952 2.654717 0.1333272
## Group_2-prediction_file_nocomments-1 2.363973 2.161389 2.578943 0.1275932
## Group_2-prediction_file_nocomments-2 2.356854 2.158908 2.571599 0.1255129
## Group_3-prediction_file_nocomments-1 1.872031 1.716226 2.034078 0.1000889
```

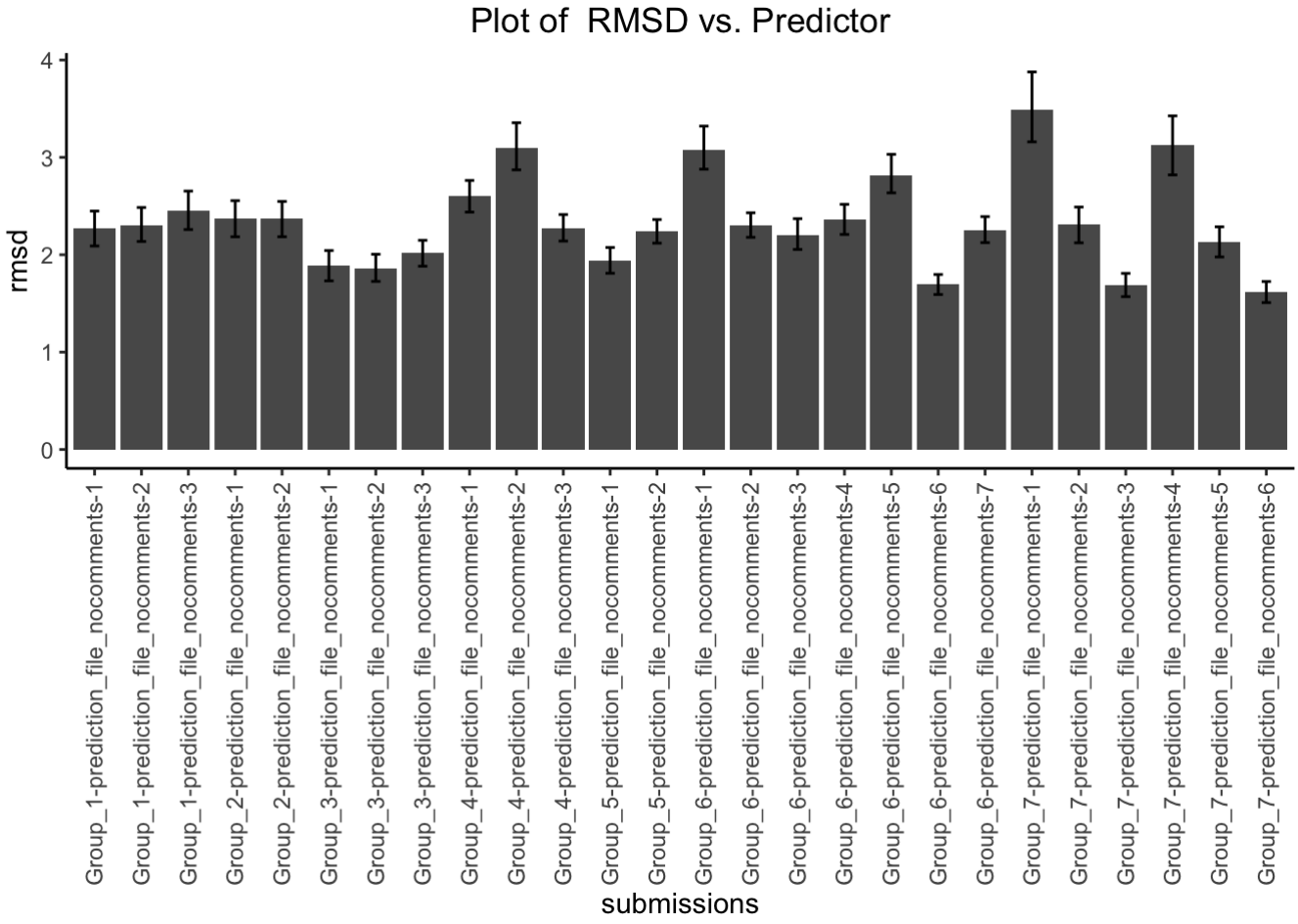
```
plot.RMSD(boot.result.rmsd2, method="",boot = TRUE)
```




```
# with variance.normalization + with density.distance
boot.result.rmsd <- eval.RMSD(real.data = exp.data, pred.data = sub.data,sd.use = NA,
                      density.distance = TRUE, variance.normalization = TRUE, boot = TRU
E)
head(boot.result.rmsd)
```

```
##                                          RMSD   low_ci  high_ci        sd
## Group_1-prediction_file_nocomments-1 2.270940 2.090101 2.448806 0.1085032
## Group_1-prediction_file_nocomments-2 2.306305 2.136658 2.485175 0.1057066
## Group_1-prediction_file_nocomments-3 2.449651 2.259246 2.653981 0.1204197
## Group_2-prediction_file_nocomments-1 2.374673 2.184303 2.555388 0.1142511
## Group_2-prediction_file_nocomments-2 2.369393 2.184681 2.547442 0.1120477
## Group_3-prediction_file_nocomments-1 1.886298 1.731956 2.042980 0.0938061
```

```
plot.RMSD(boot.result.rmsd, method="",boot = TRUE)
```



Cut-off-based Evaluation

threshold = 0.6: The cutoff for positive/negative value

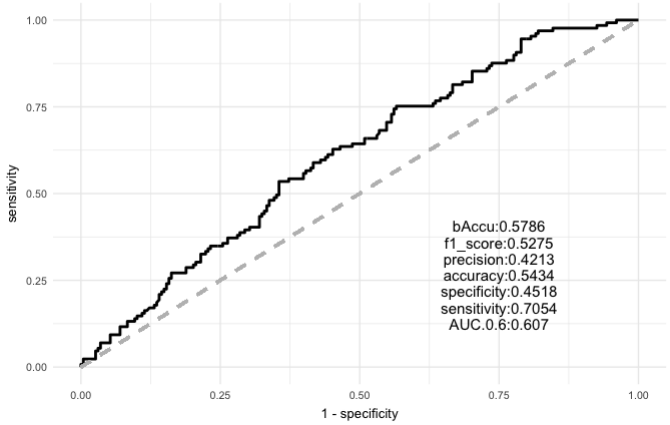
```
result.auc.0.6 <- eval.AUC(real.data =exp.data, pred.data = sub.data,
                           threshold = 0.6)
head(result.auc.0.6$results)
```

```
##                                      AUC.0.6 sensitivity specificity
## Group_1-prediction_file_nocomments-1  0.6070      0.7054      0.4518
## Group_1-prediction_file_nocomments-2  0.5775      0.6977      0.4693
## Group_1-prediction_file_nocomments-3  0.5743      0.7287      0.3991
## Group_2-prediction_file_nocomments-1  0.5811      0.7674      0.3509
## Group_2-prediction_file_nocomments-2  0.5960      0.7674      0.3640
## Group_3-prediction_file_nocomments-1  0.6335      0.7674      0.4079
##                                      accuracy precision f1_score  bAccu
## Group_1-prediction_file_nocomments-1   0.5434    0.4213   0.5275 0.5786
## Group_1-prediction_file_nocomments-2   0.5518    0.4265   0.5294 0.5835
## Group_1-prediction_file_nocomments-3   0.5182    0.4069   0.5222 0.5639
## Group_2-prediction_file_nocomments-1   0.5014    0.4008   0.5266 0.5592
## Group_2-prediction_file_nocomments-2   0.5098    0.4057   0.5308 0.5657
## Group_3-prediction_file_nocomments-1   0.5378    0.4231   0.5455 0.5877
```

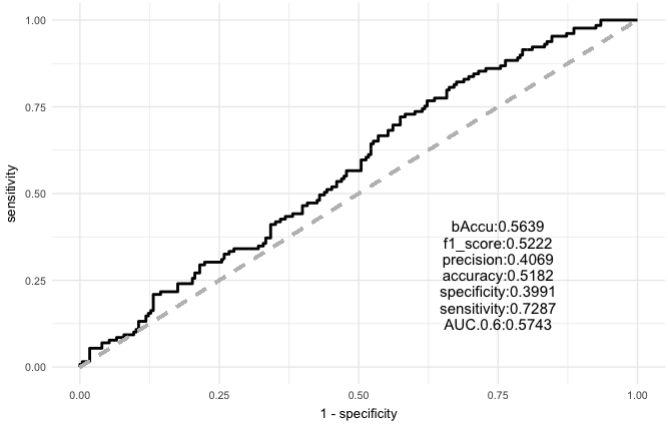
```
plot.AUC(result.auc.0.6)
```

page 1 of 7

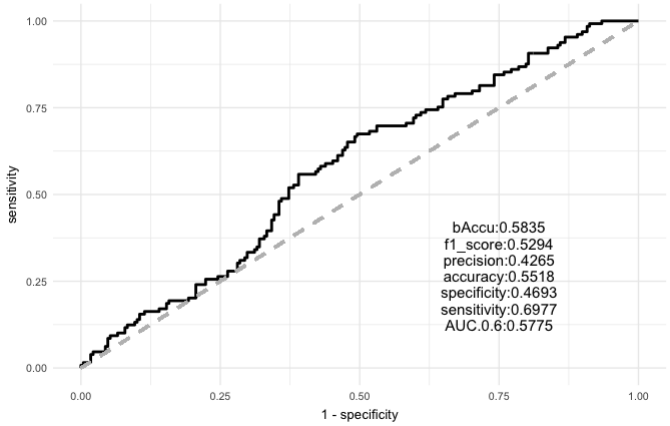
Group_1-prediction_file_nocomments-1thresh=0.6NAROCcurve



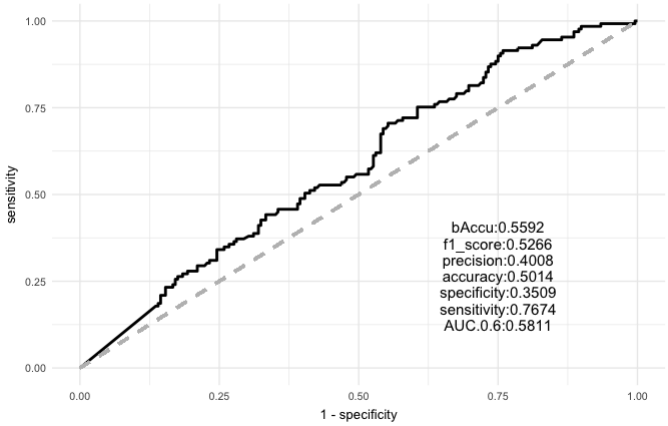
Group_1-prediction_file_nocomments-3thresh=0.6NAROCcurve



Group_1-prediction_file_nocomments-2thresh=0.6NAROCcurve

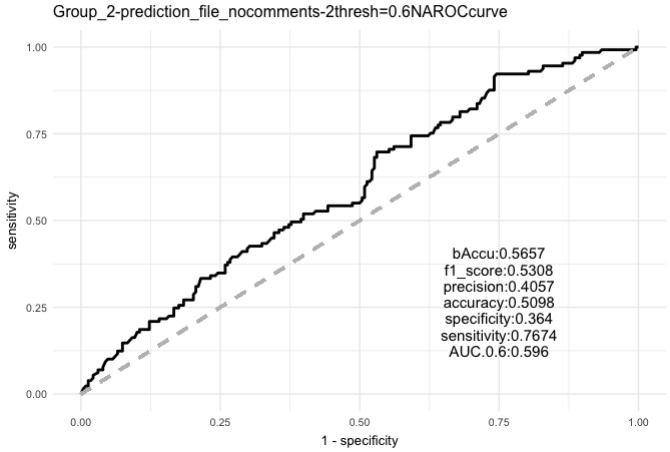


Group_2-prediction_file_nocomments-1thresh=0.6NAROCcurve

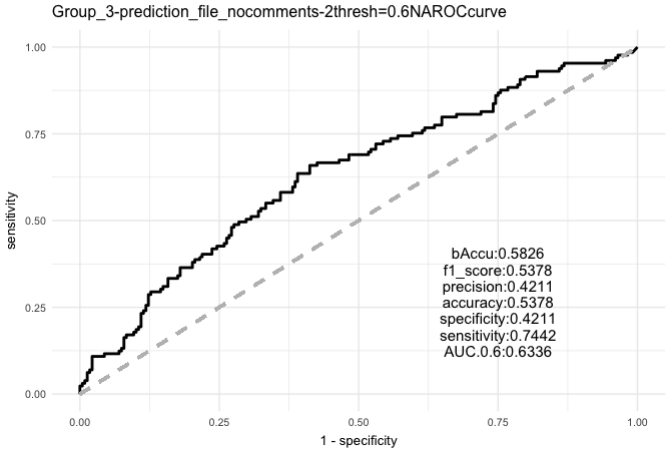


page 2 of 7

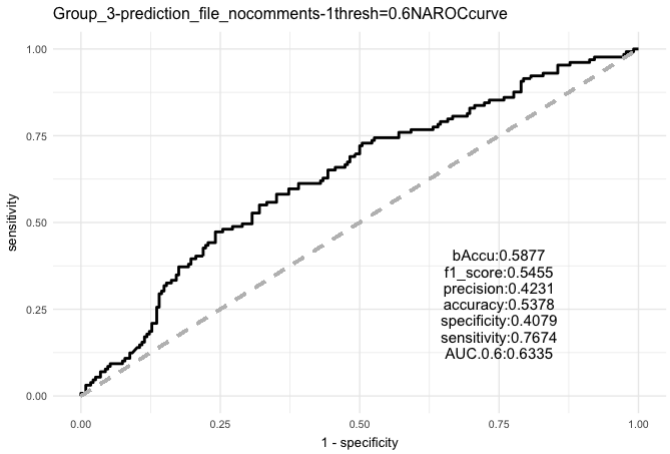
Group_2-prediction_file_nocomments-2thresh=0.6NAROCcurve



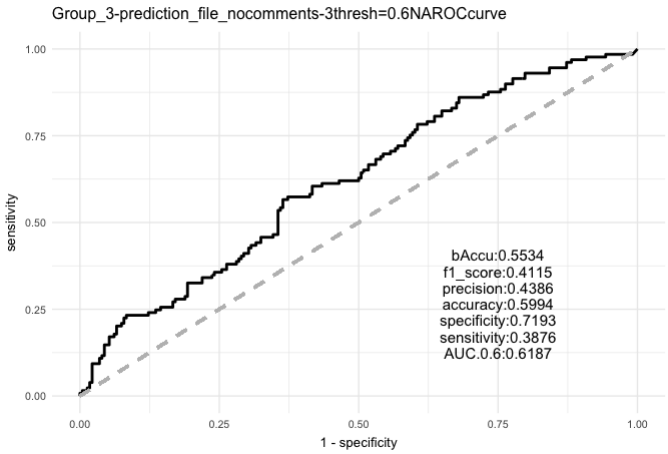
Group_3-prediction_file_nocomments-2thresh=0.6NAROCcurve



Group_3-prediction_file_nocomments-1thresh=0.6NAROCcurve

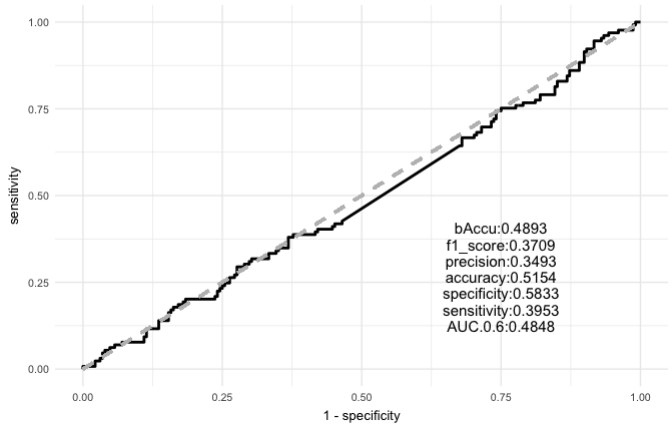


Group_3-prediction_file_nocomments-3thresh=0.6NAROCcurve

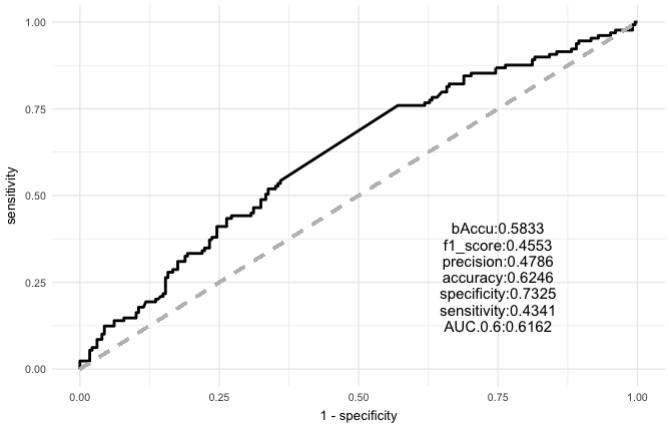


page 3 of 7

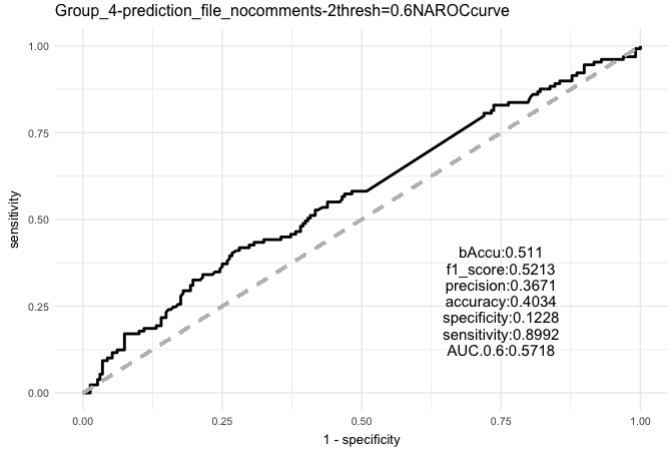
Group_4-prediction_file_nocomments-1thresh=0.6NAROCcurve



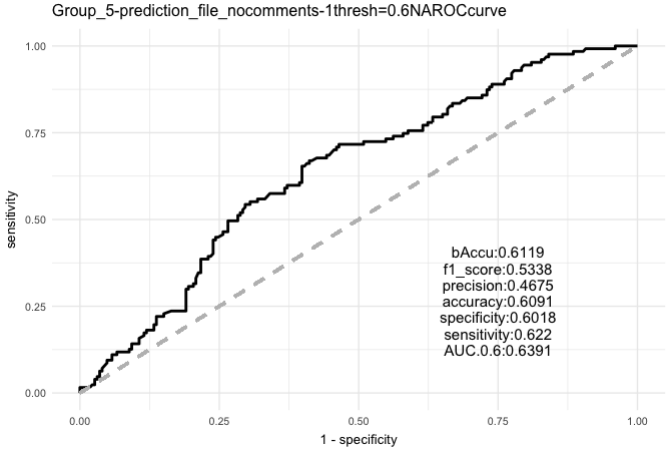
Group_4-prediction_file_nocomments-3thresh=0.6NAROCcurve



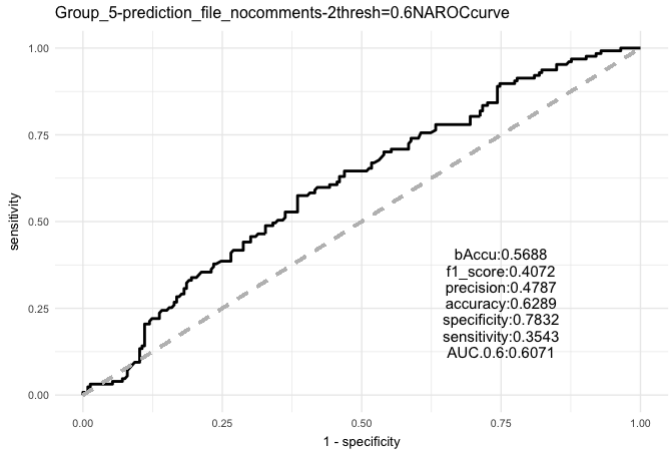
Group_4-prediction_file_nocomments-2thresh=0.6NAROCcurve



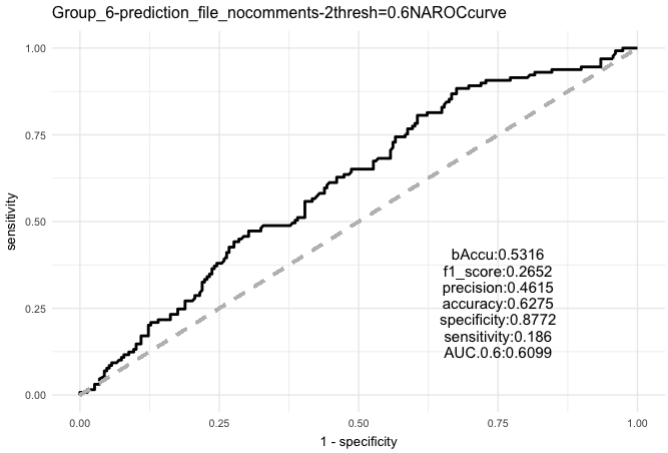
Group_5-prediction_file_nocomments-1thresh=0.6NAROCcurve



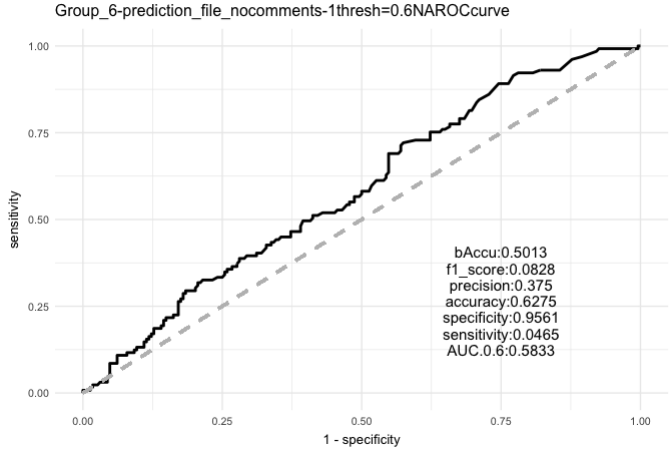
Group_5-prediction_file_nocomments-2thresh=0.6NAROCcurve



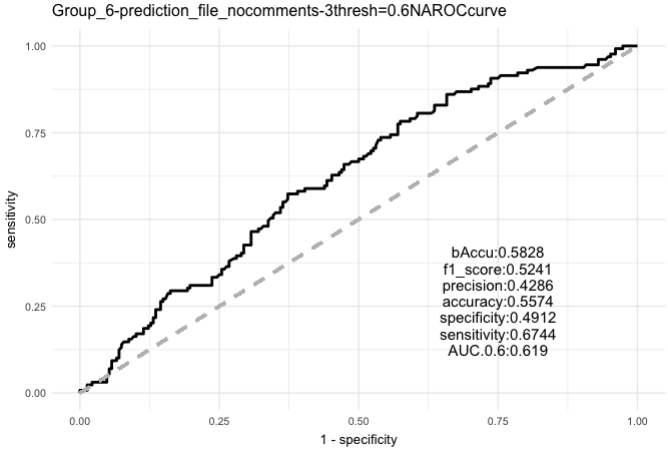
Group_6-prediction_file_nocomments-2thresh=0.6NAROCcurve



Group_6-prediction_file_nocomments-1thresh=0.6NAROCcurve

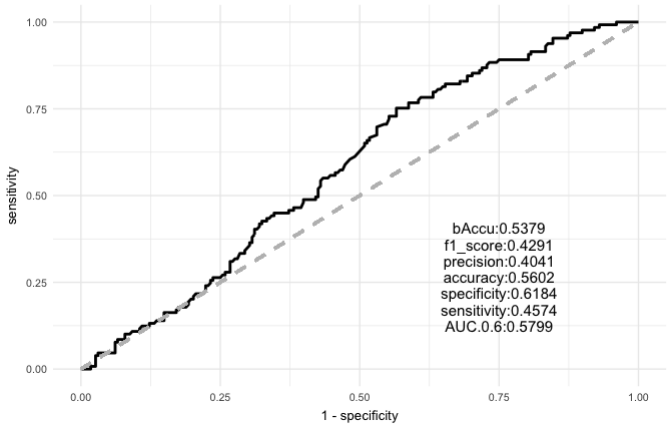


Group_6-prediction_file_nocomments-3thresh=0.6NAROCcurve

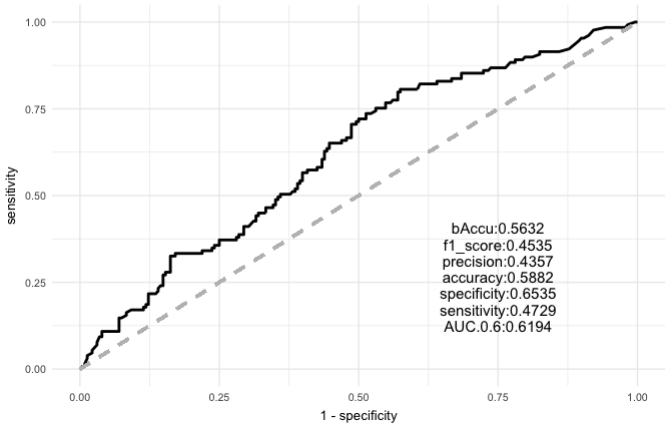


page 5 of 7

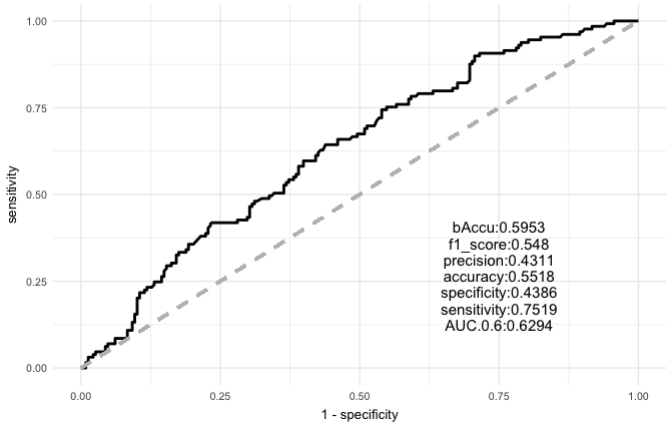
Group_6-prediction_file_nocomments-4thresh=0.6NAROCcurve



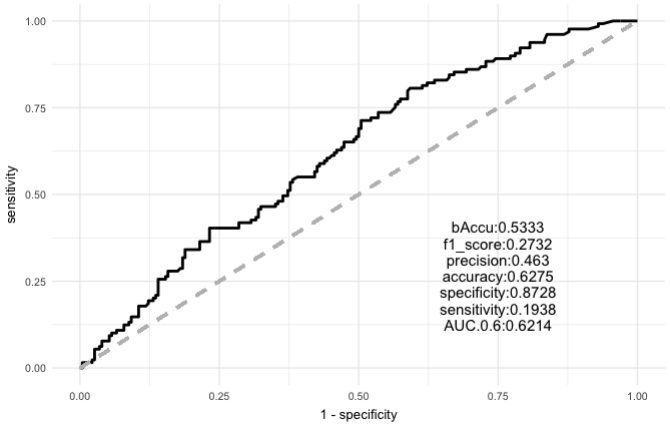
Group_6-prediction_file_nocomments-6thresh=0.6NAROCcurve



Group_6-prediction_file_nocomments-5thresh=0.6NAROCcurve

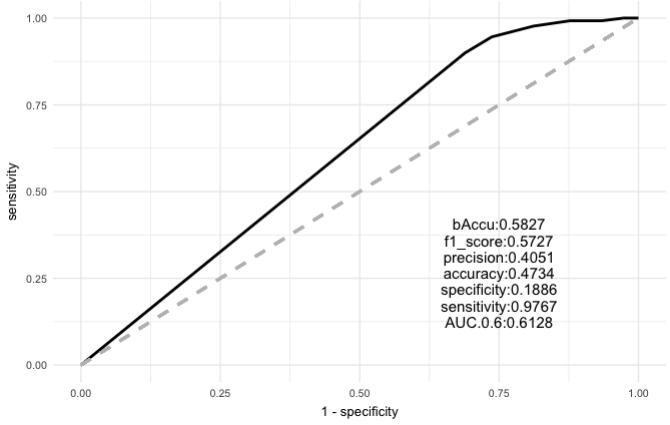


Group_6-prediction_file_nocomments-7thresh=0.6NAROCcurve

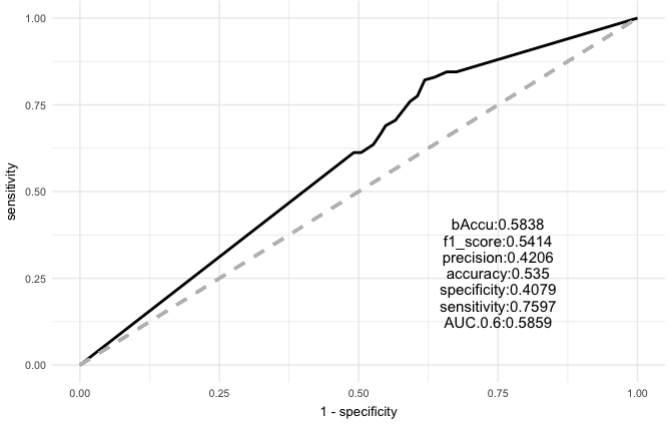


page 6 of 7

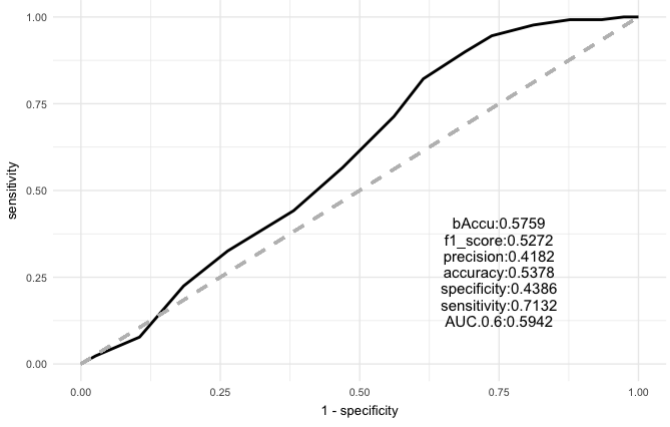
Group_7-prediction_file_nocomments-1thresh=0.6NAROCcurve



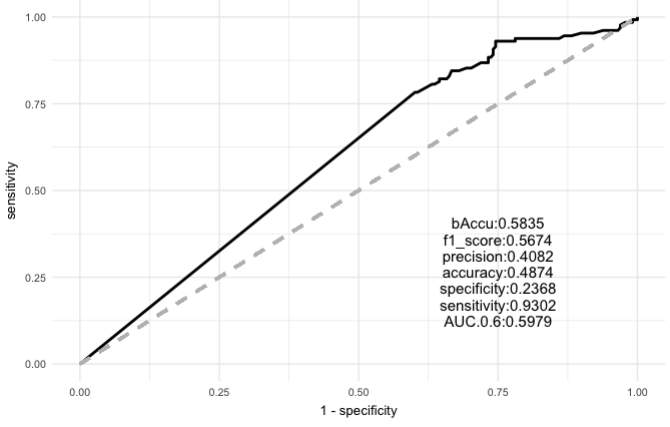
Group_7-prediction_file_nocomments-3thresh=0.6NAROCcurve

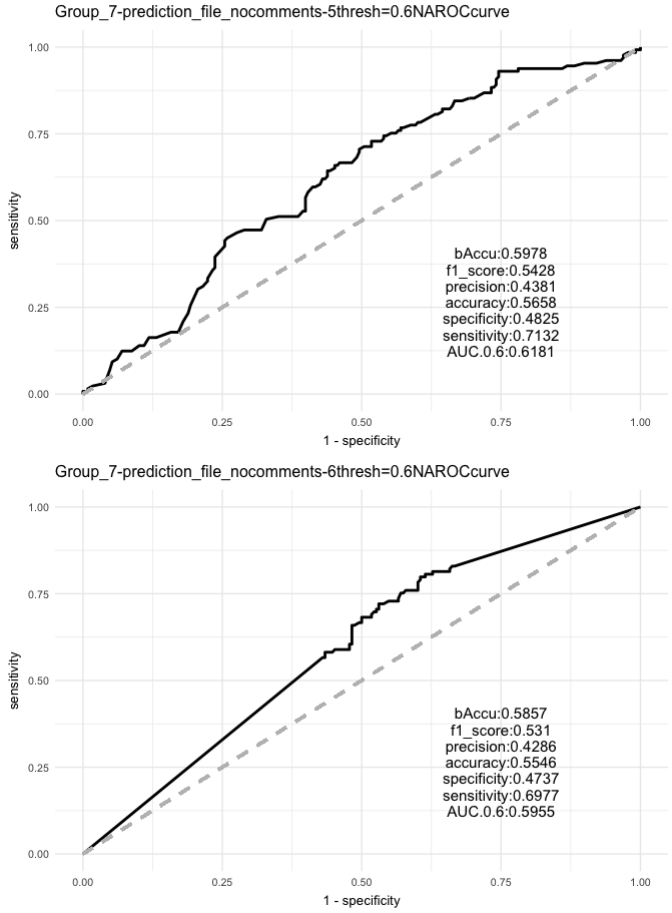


Group_7-prediction_file_nocomments-2thresh=0.6NAROCcurve



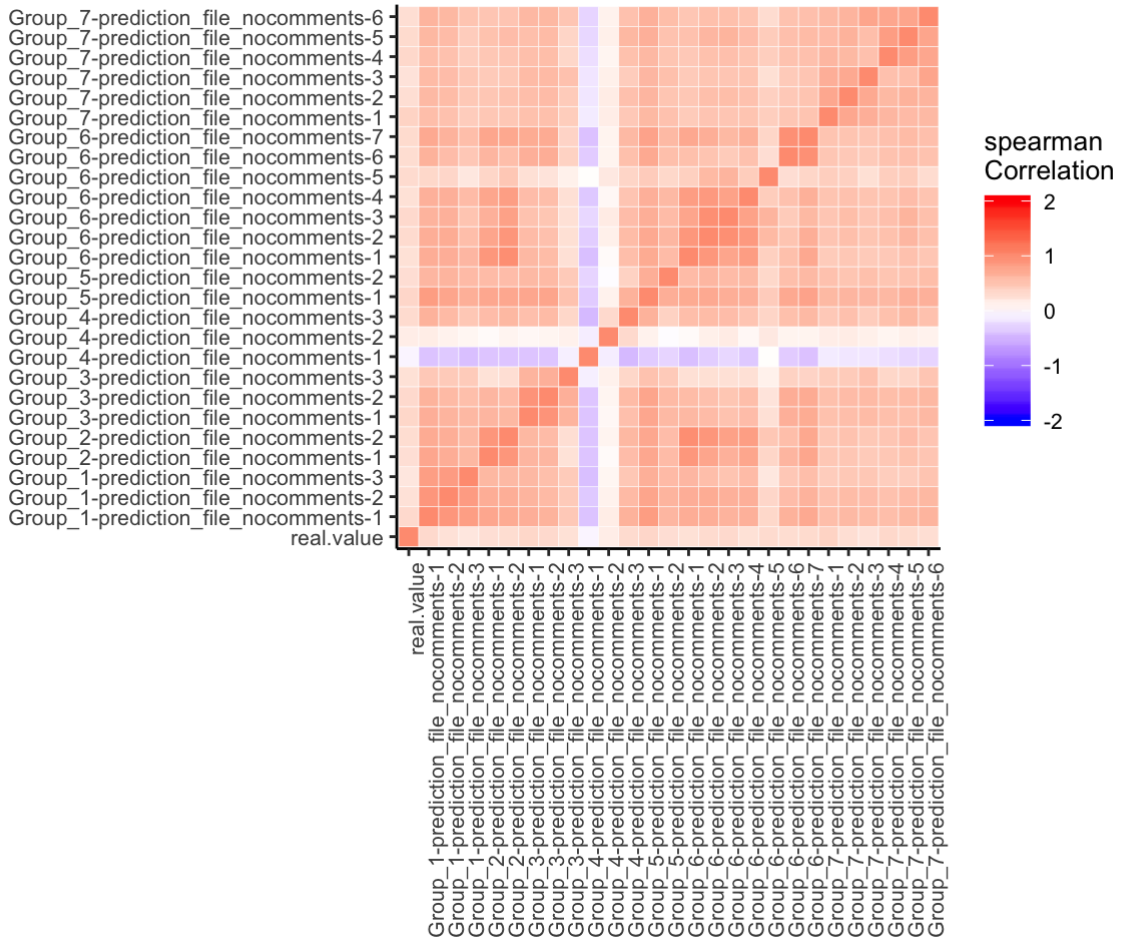
Group_7-prediction_file_nocomments-4thresh=0.6NAROCcurve





Between-method Evaluation

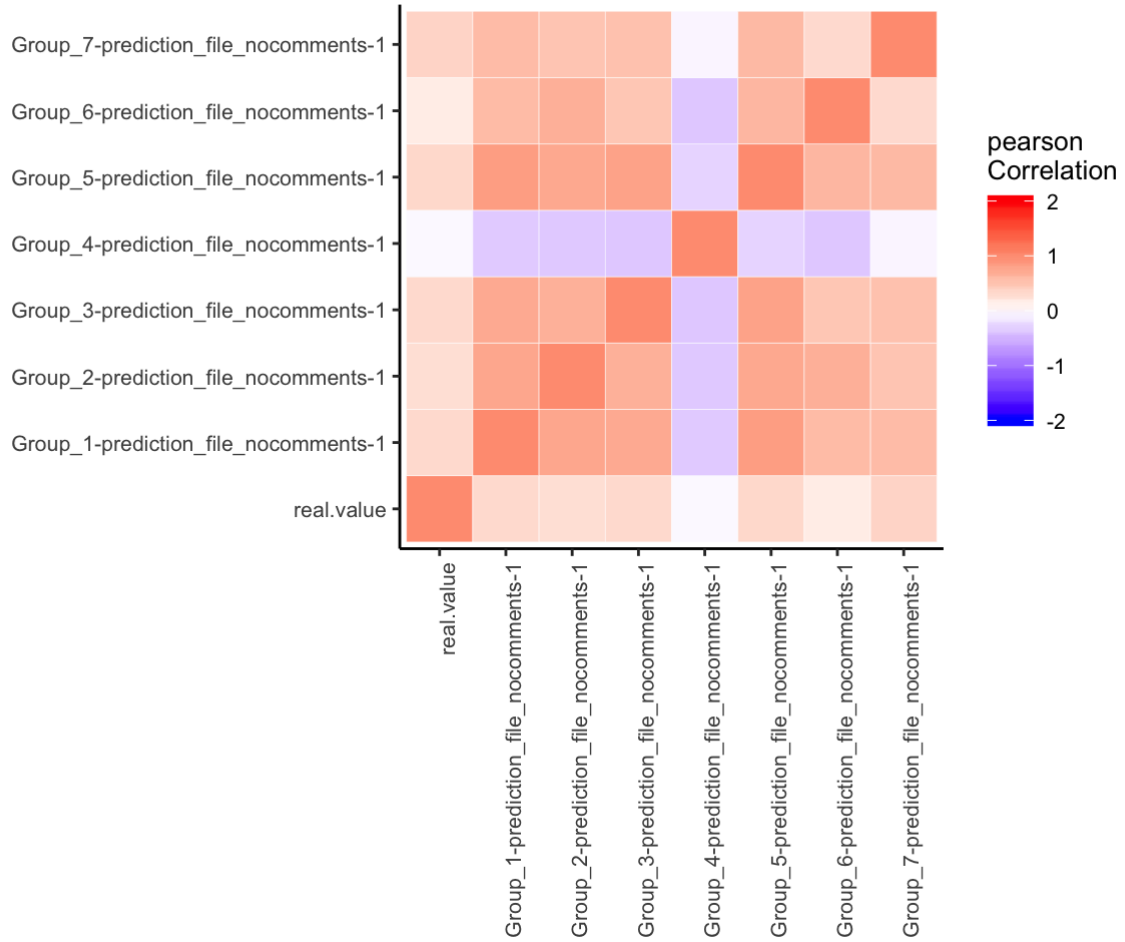
```
# for all the submission files
result.bM.spearman <-eval.Correlation.Between(real.data = exp.data, pred.data = sub.dat
a,
                                                method = "spearman",sd.use = NA,z.transfor
m = TRUE)
plot.Correlation.Between(result.bM.spearman$coefficient, method="spearman")
```




```
# for best submission of each group
result.bM.pearson <-eval.Correlation.Between(real.data = exp.data, pred.data = sub.data,
                                             method = "pearson",sd.use = NA,z.transform
= TRUE,grouped = TRUE)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
plot.Correlation.Between(result.bM.pearson$coefficient, method="pearson")
```



Partial-Correlation Evaluation (controlling the covariates)

```
result.pCor <- eval.Partial.Correlation(real.data = exp.data, pred.data = sub.data, meth
od = "spearman")
```

```
## Loading required package: ppcor
```

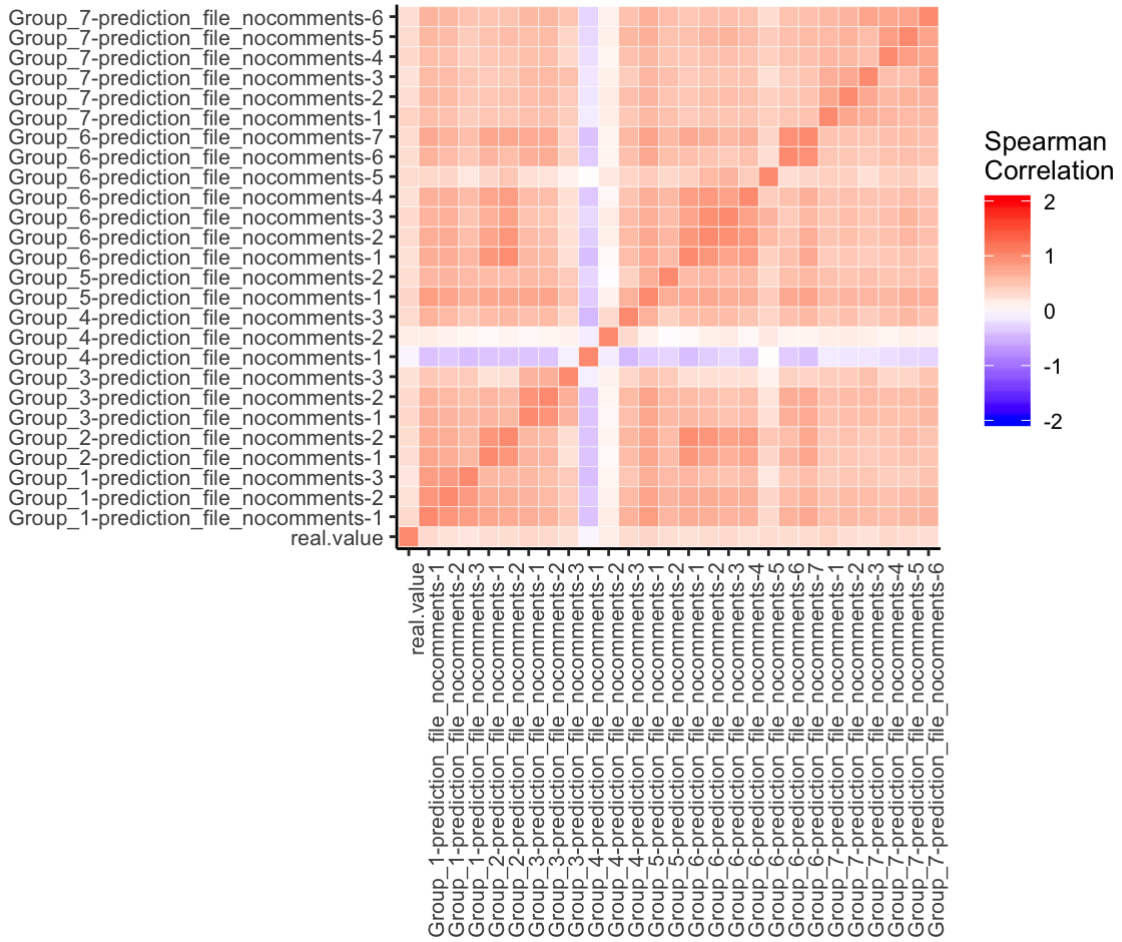
```
## Loading required package: MASS
```

```
## Warning: package 'MASS' was built under R version 3.4.4
```

```
##
## Attaching package: 'MASS'
```

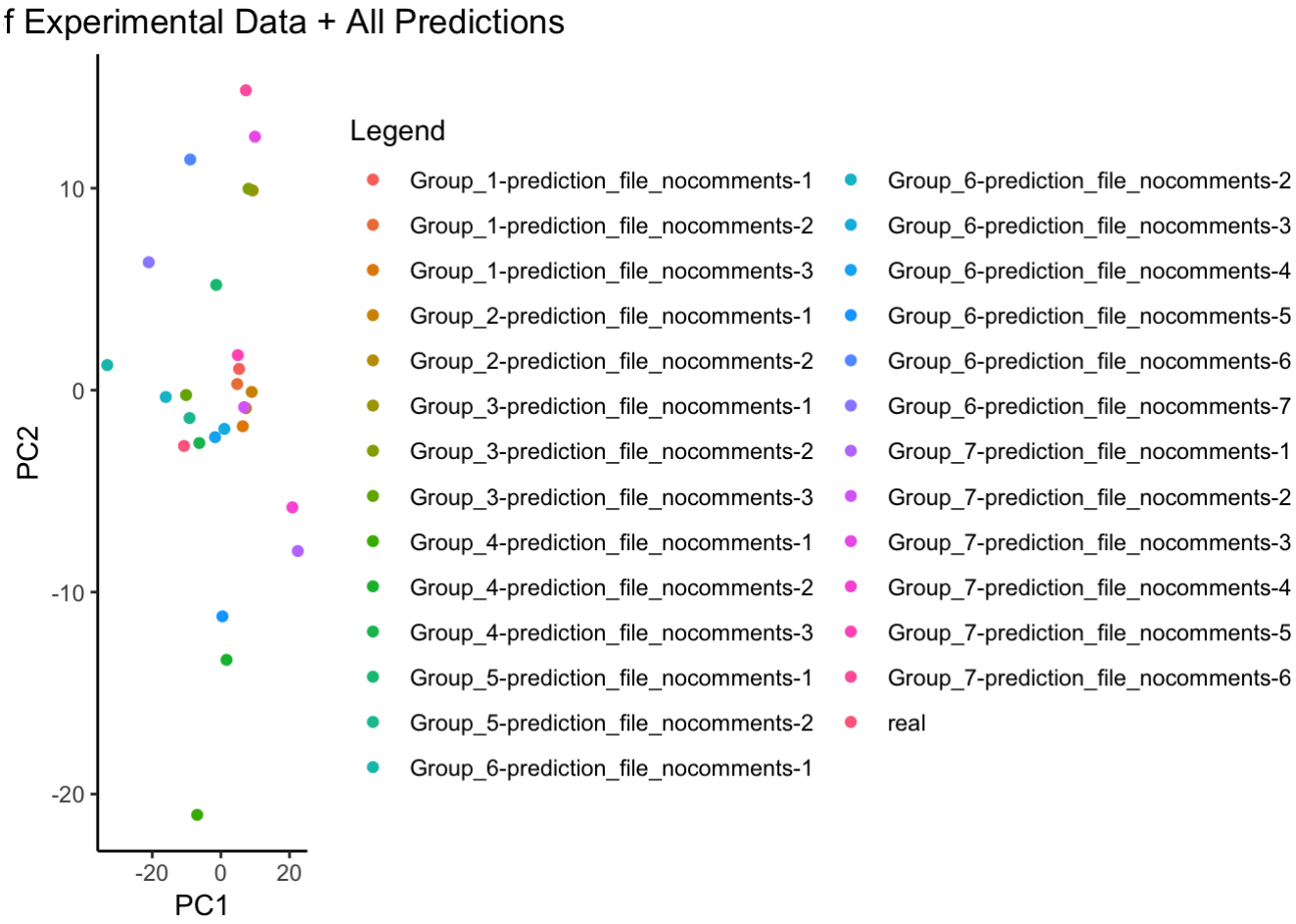
```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
plot.Correlation.Between(result.bM.spearman$coefficient, method="Spearman")
```



PCA Plot

```
total = cbind(real = exp.data$value,sub.data$value)
Plot.PCA(na.omit(total), labels=F, legend=TRUE)
```



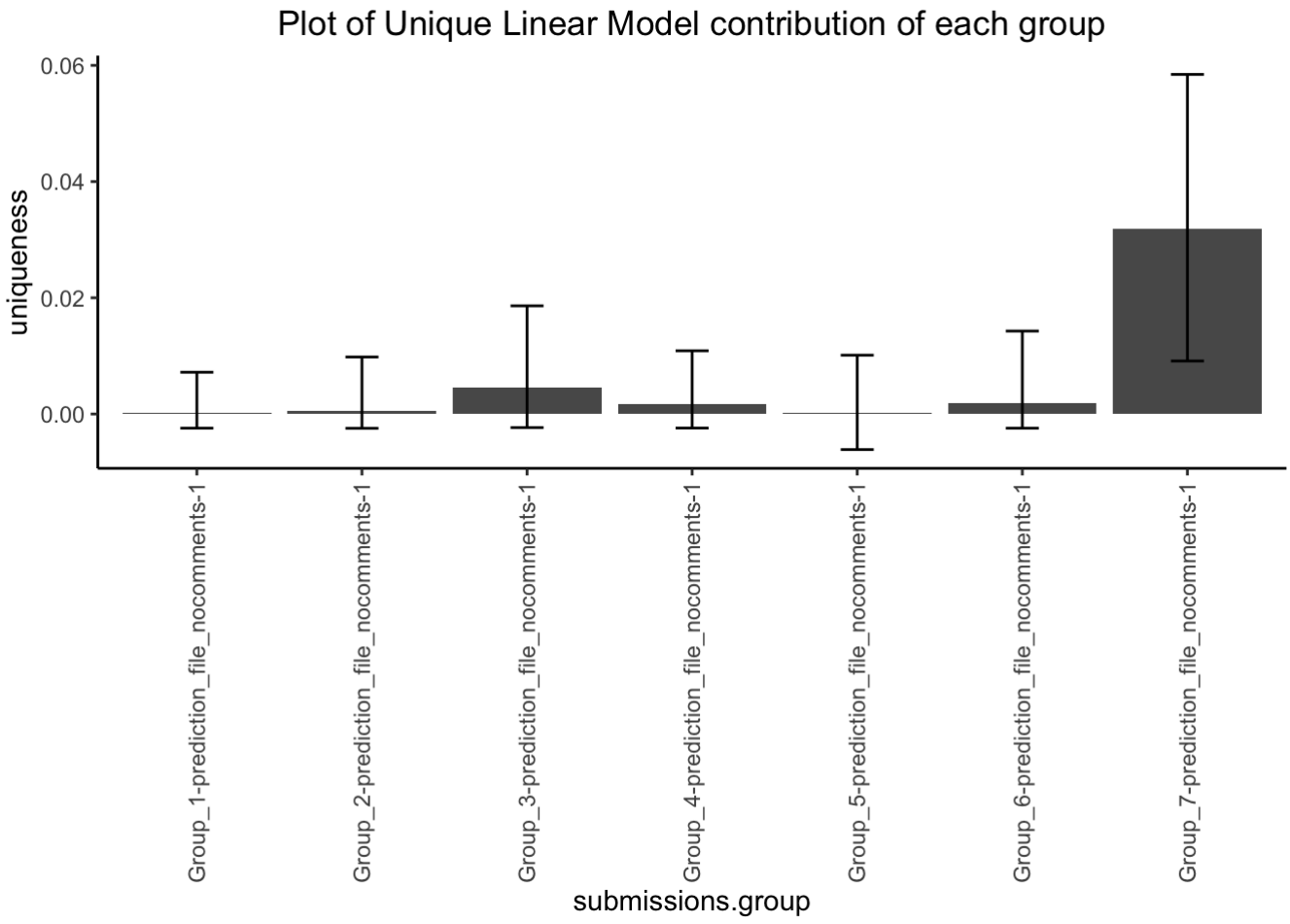
Uniqueness Evaluation

uniqueness as adj.r^2 difference between total linear model and linear models without certain group use.ci = T : if true error bar as confidence interval; if false, error bar as standard deviation

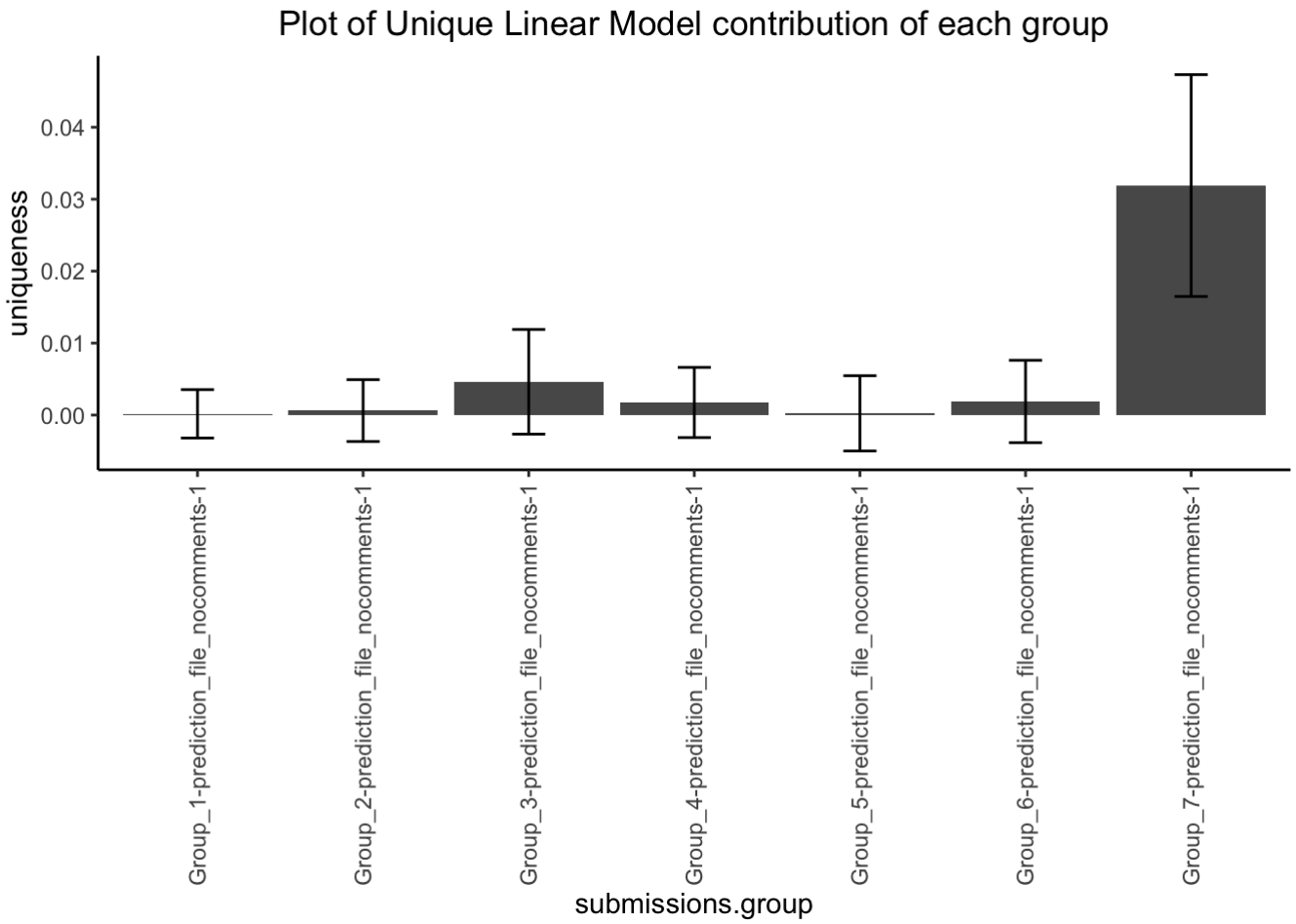
```
# resampling row
result.uniq = eval.uniqueness(real.data = exp.data, pred.data = sub.data,boot = TRUE)
result.uniq
```

```
##                                          uniqueness       low_ci     high_ci
## Group_1-prediction_file_nocomments-1 0.0001689140 -0.002436104 0.007191532
## Group_2-prediction_file_nocomments-1 0.0006190618 -0.002458364 0.009810393
## Group_3-prediction_file_nocomments-1 0.0046221322 -0.002334226 0.018613228
## Group_4-prediction_file_nocomments-1 0.0017466180 -0.002418622 0.010884977
## Group_5-prediction_file_nocomments-1 0.0002317603 -0.006121336 0.010123986
## Group_6-prediction_file_nocomments-1 0.0018855771 -0.002440361 0.014276658
## Group_7-prediction_file_nocomments-1 0.0318935936  0.009125475 0.058443523
##                                                sd
## Group_1-prediction_file_nocomments-1 0.003363947
## Group_2-prediction_file_nocomments-1 0.004297741
## Group_3-prediction_file_nocomments-1 0.007274703
## Group_4-prediction_file_nocomments-1 0.004879735
## Group_5-prediction_file_nocomments-1 0.005230960
## Group_6-prediction_file_nocomments-1 0.005725523
## Group_7-prediction_file_nocomments-1 0.015416475
```

```
plot.uniqueness(result.uniq, method="", boot = TRUE,use.ci = T)
```



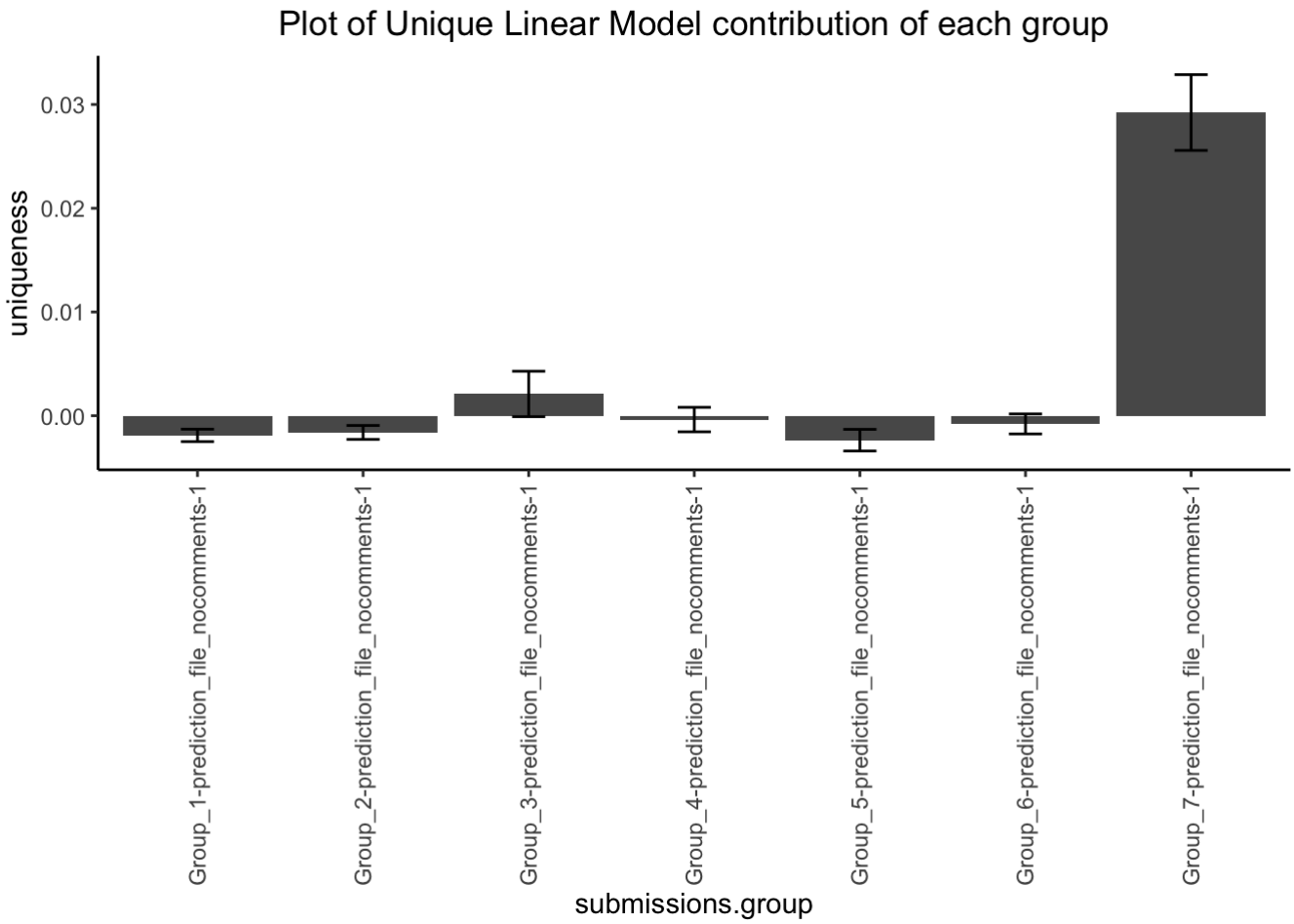

```
plot.uniqueness(result.uniq, method="", boot = TRUE,use.ci = F)
```




```
# generating from experimental distribution
result.bootvar.uniq = eval.uniqueness(real.data = exp.data, pred.data = sub.data,boot =
TRUE,boot.var = T)
result.bootvar.uniq
```

```
##                                       uniqueness        low_ci
## Group_1-prediction_file_nocomments-1 -0.0018969667 -0.0024850231
## Group_2-prediction_file_nocomments-1 -0.0016130751 -0.0024285225
## Group_3-prediction_file_nocomments-1  0.0020983826 -0.0008795615
## Group_4-prediction_file_nocomments-1 -0.0003706045 -0.0019489749
## Group_5-prediction_file_nocomments-1 -0.0023593861 -0.0038966287
## Group_6-prediction_file_nocomments-1 -0.0007913476 -0.0021447765
## Group_7-prediction_file_nocomments-1  0.0292259365  0.0235040612
##                                          high_ci           sd
## Group_1-prediction_file_nocomments-1 -0.0007528102 0.0005996154
## Group_2-prediction_file_nocomments-1 -0.0003943794 0.0006703160
## Group_3-prediction_file_nocomments-1  0.0064608075 0.0021915123
## Group_4-prediction_file_nocomments-1  0.0018032103 0.0011808524
## Group_5-prediction_file_nocomments-1 -0.0003875242 0.0010427794
## Group_6-prediction_file_nocomments-1  0.0008691634 0.0009654445
## Group_7-prediction_file_nocomments-1  0.0355077126 0.0036533292
```

```
plot.uniqueness(result.bootvar.uniq, method="",boot = TRUE,use.ci = F)
```



CAGI-Pymol

```
res = eval.correctness(real.data = exp.data, pred.data = sub.data,threshold = 0.5,sd.use
= NA,lower.positive = F,z.transform = F,"5nn3")
head(res)
```

```
## # A tibble: 6 x 2
##   res   avg.correctness
##   <chr>           <dbl>
## 1 4               0.115
## 2 6               0.885
## 3 8               0.5
## 4 11              0.359
## 5 12              0.635
## 6 15              0.962
```

Then load cagi_plot.py module into pymol and call plot_cagi function

`PlotCorrectness(prot_name,path)` The function output a .pse file for the protein

Red: correctness closer to 100% Green: correctness towards 0%