Outputs inspection half CIFAR10

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
library("ggpubr")
```

```
## Warning: package 'ggpubr' was built under R version 4.0.5
```

```
library(LDATS)
```

```
## Warning: package 'LDATS' was built under R version 4.0.5
```

```
library(stringr)
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.0.3

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ggVennDiagram)
```

```
## Warning: package 'ggVennDiagram' was built under R version 4.0.5
```

```
library(reticulate)
```

```
## Warning: package 'reticulate' was built under R version 4.0.5
```

```
library(abind)
```

```
## Warning: package 'abind' was built under R version 4.0.3
```

```
np <- import("numpy")

source("utils.R")
```

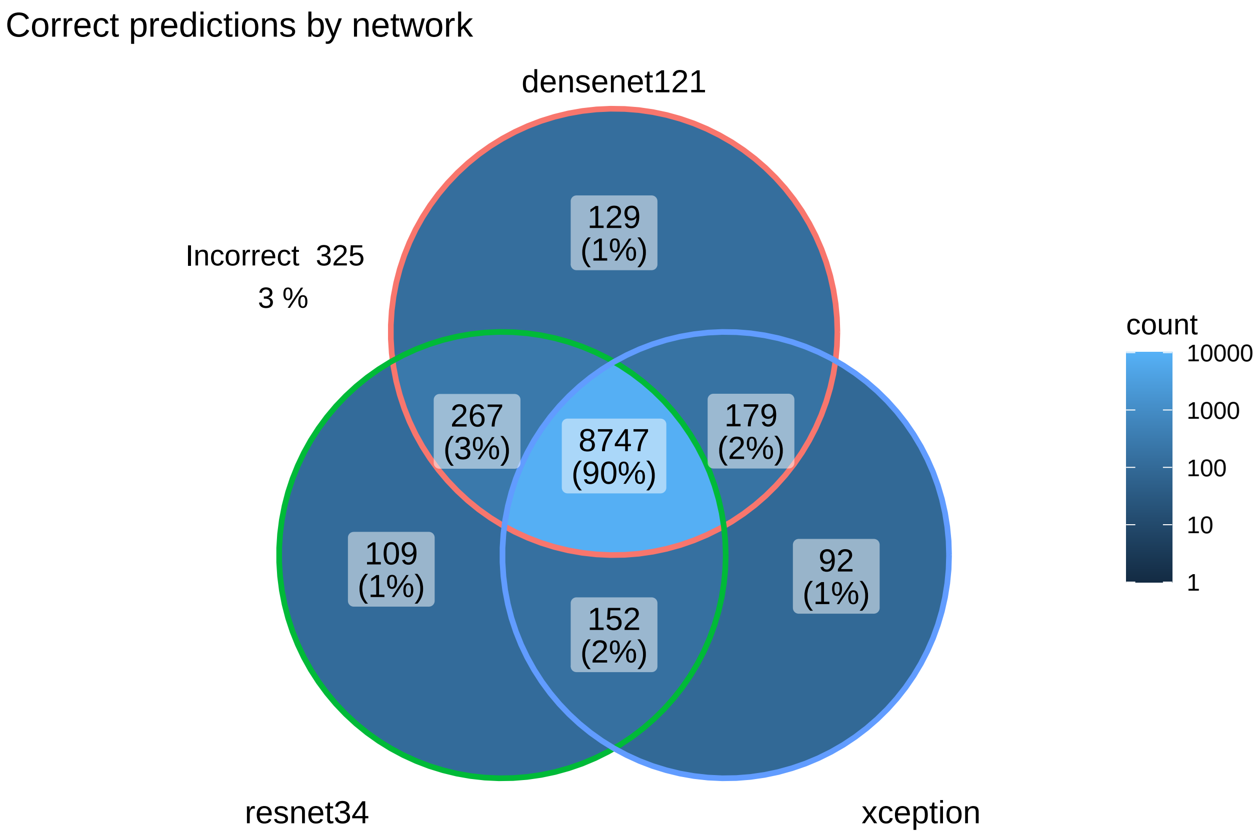
```
## Warning: package 'hash' was built under R version 4.0.5

## hash-2.2.6.1 provided by Decision Patterns

## Warning: package 'berryFunctions' was built under R version 4.0.5

##
## Attaching package: 'berryFunctions'

## The following object is masked from 'package:ggVennDiagram':
##
##     circle

## The following object is masked from 'package:dplyr':
##
##     between

## Warning: package 'purrr' was built under R version 4.0.3
```

Visualization on CIFAR10. We are using data of three neural networks trained on reduced CIFAR10 training set. Half of the CIFAR10 training set was extracted as a validation set. We then divided both the reduced training set and validation set into 50 disjoint subsets and trained an ensemble on each of them. In this visualization, we are trying to inspect the outputs deeper, mainly to make sense of strange behavior of nll metric for ensemble outputs.

```
base_dir <- "../data/data_train_val_half_c10"
repls <- 0:0
folds <- 0:49
classes <- 10

nets_outputs <- load_network_outputs(base_dir, repls)
ens_outputs <- load_ensemble_outputs(base_dir, repls, folds)
net_results <- read.csv(file.path(base_dir, "net_accuracies.csv"))
ens_results <- read.csv(file.path(base_dir, "ensemble_accuracies.csv"))
```

```
sort_ind <- function(lst)
{
  return(sort(lst, index.return=TRUE, decreasing=TRUE)$ix)
}
nets_test_top_indices <- apply(X=nets_outputs$test_outputs, MARGIN=c(1, 2, 3), FUN=sort_ind)[1, , , ]
r_n <- length(repls)
samples_n <- dim(nets_outputs$test_labels)[2]
nets_n <- length(nets_outputs$networks)
test_labs <- nets_outputs$test_labels + 1
dim(test_labs) <- c(r_n, 1, samples_n)
test_labs <- aperm(abind(array(rep(aperm(test_labs, perm=c(2, 1, 3)), nets_n), c(r_n, samples_n, nets_n)
if (r_n == 1)
{ dim(test_labs) <- dim(test_labs)[-1] }
nets_test_cor_preds <- test_labs == nets_test_top_indices
```

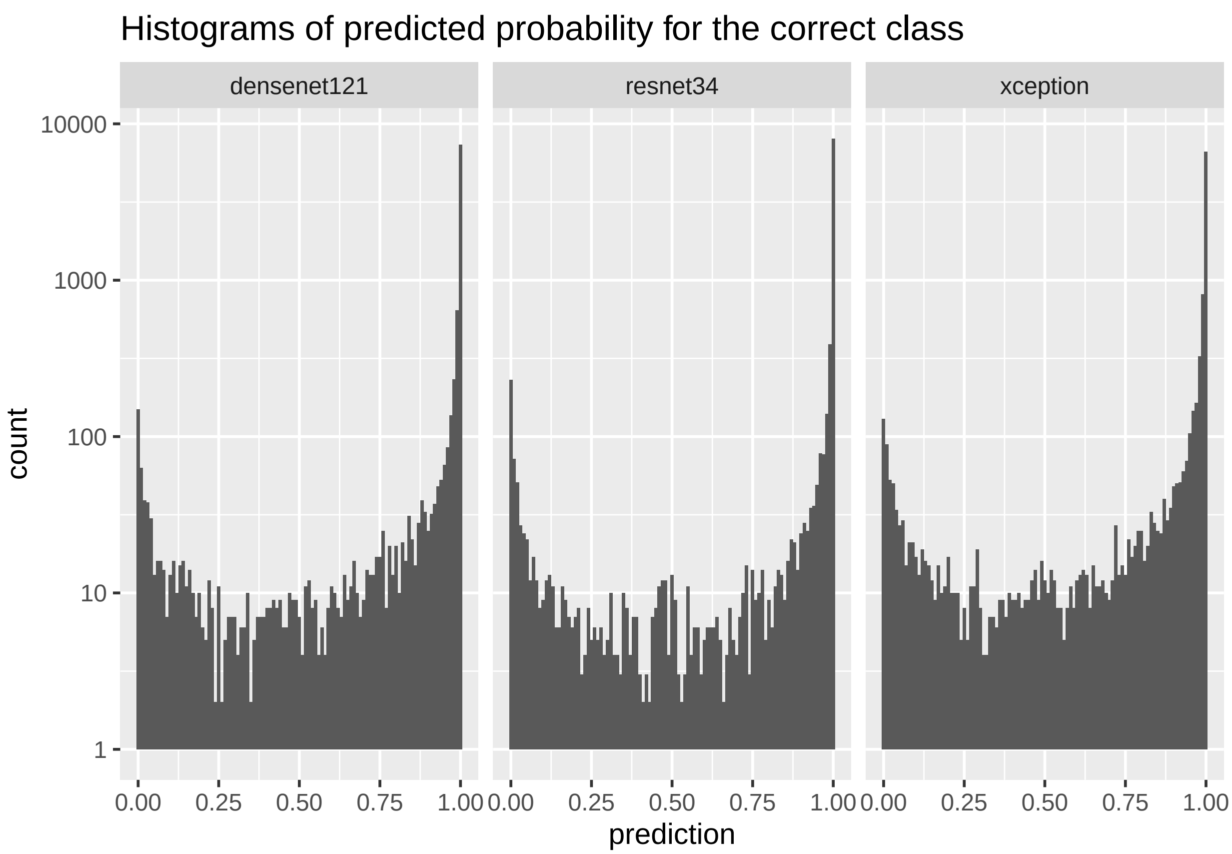
```
nets_cor_list <- list()
incor <- 1:samples_n
for (ni in 1:nets_n)
{
  cor_list <- which(nets_test_cor_preds[ni, ])
  nets_cor_list[[nets_outputs$networks[ni]]] = cor_list
  incor <- setdiff(incor, cor_list)
}
incor_n <- length(incor)
venn_diag <- ggVennDiagram(nets_cor_list) + scale_fill_gradient(trans="log10", name="count", limits=c(1
  annotate(geom="text", x=-4, y=5, label=paste("Incorrect ", incor_n, "\n", round(incor_n / samples_n *
    ggtitle("Correct predictions by network") +
    scale_x_continuous(limits=c(-8, 10))
print(venn_diag)
```



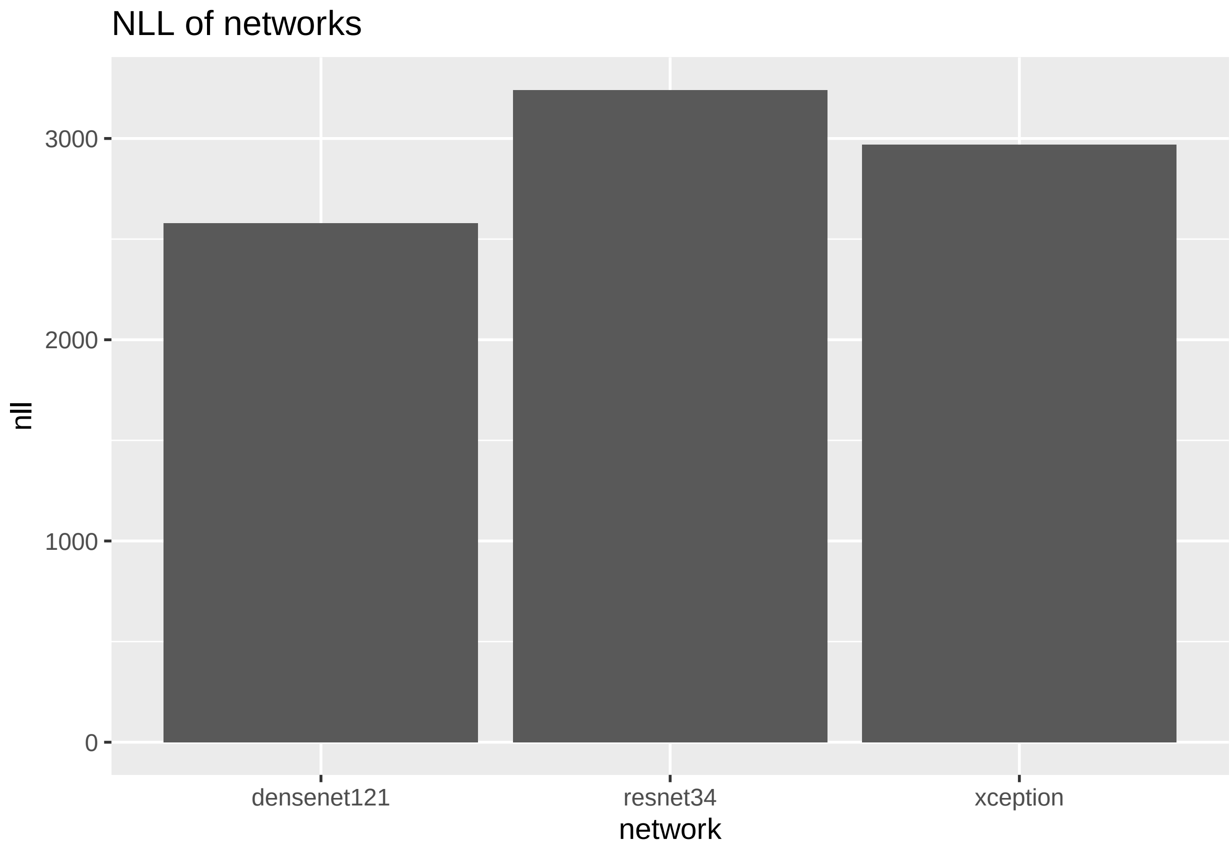
Compared to training the networks on almost complete CIFAR 10 training set, in this case all networks are correct in fewer samples. Otherwise the observations hold.

```
preds <- nets_outputs$test_outputs
for (ri in repls + 1)
{
  for (net_i in seq_along(nets_outputs[["networks"]]))
  {
    preds[ri, net_i, ,] <- softmax(preds[ri, net_i, , ])
  }
}
nets_test_cor_probs <- gather(preds, 1 + nets_outputs$test_labels[1, ], 3, 4)
nets_test_cor_probs <- melt(nets_test_cor_probs)
nets_test_cor_probs <- nets_test_cor_probs[, c(-3, -4)]
names(nets_test_cor_probs) <- c("replication", "network", "prediction")
nets_test_cor_probs$network <- as.factor(nets_test_cor_probs$network)
levels(nets_test_cor_probs$network) <- nets_outputs$networks
```

```
nets_cor_preds_histo <- ggplot(data=nets_test_cor_probs) + geom_histogram(mapping=aes(x=prediction), bi
  ggtitle("Histograms of predicted probability for the correct class") + facet_wrap(~network) + scale_y
nets_cor_preds_histo
```

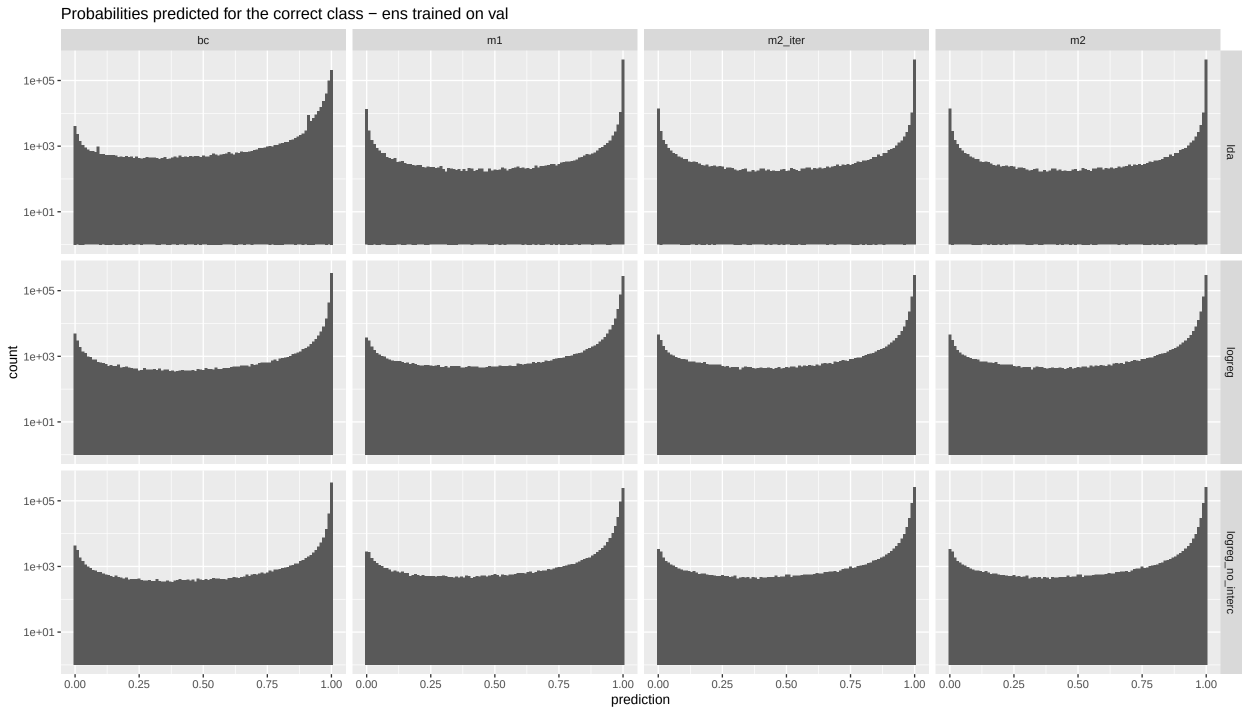



```
networks_nll <- ggplot(data=net_results) + geom_bar(mapping=aes(x=network, y=nll), stat="identity") + gg
networks_nll
```




```
val_ens_cor_probs <- gather(ens_outputs$val_training, 1 + nets_outputs$test_labels[1, ], 5, 6)
val_ens_cor_probs <- melt(val_ens_cor_probs)
val_ens_cor_probs <- val_ens_cor_probs[, c(-5, -6)]
names(val_ens_cor_probs) <- c("replication", "combining_method", "coupling_method", "fold", "prediction
val_ens_cor_probs[, c("combining_method", "coupling_method")] <- lapply(val_ens_cor_probs[, c("combining
levels(val_ens_cor_probs$combining_method) <- ens_outputs$combining_methods
levels(val_ens_cor_probs$coupling_method) <- ens_outputs$coupling_methods
```

```
val_ens_cor_preds_histo <- ggplot(data=val_ens_cor_probs) + geom_histogram(mapping=aes(x=prediction), bi
val_ens_cor_preds_histo
```

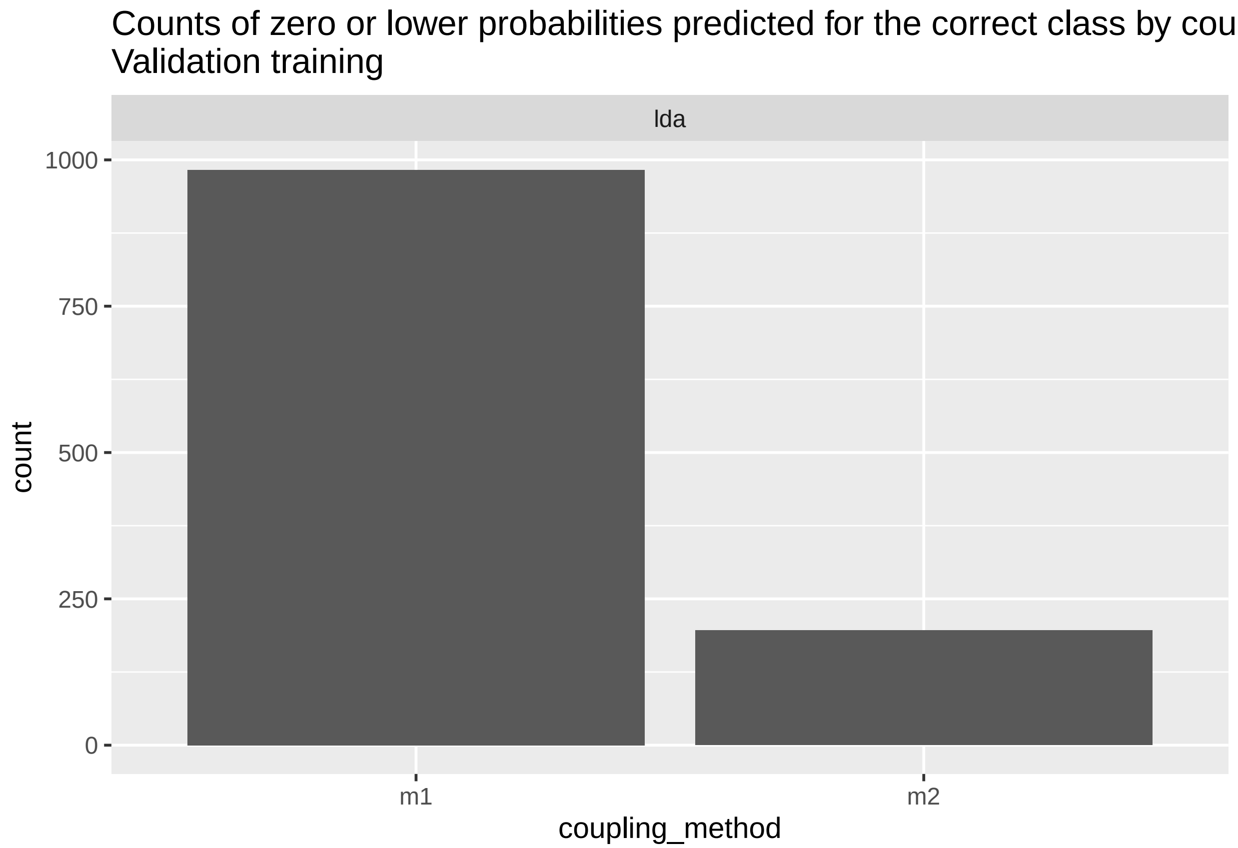


In the lda case, coupling method bc produces fewer probabilities falling into the lowest bin for the correct class than m1 and m2. In logreg cases all methods perform similarly.

```
val_ens_zero_counts <- ggplot(data=val_ens_cor_probs[val_ens_cor_probs$prediction <= 0, ]) + geom_histo
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

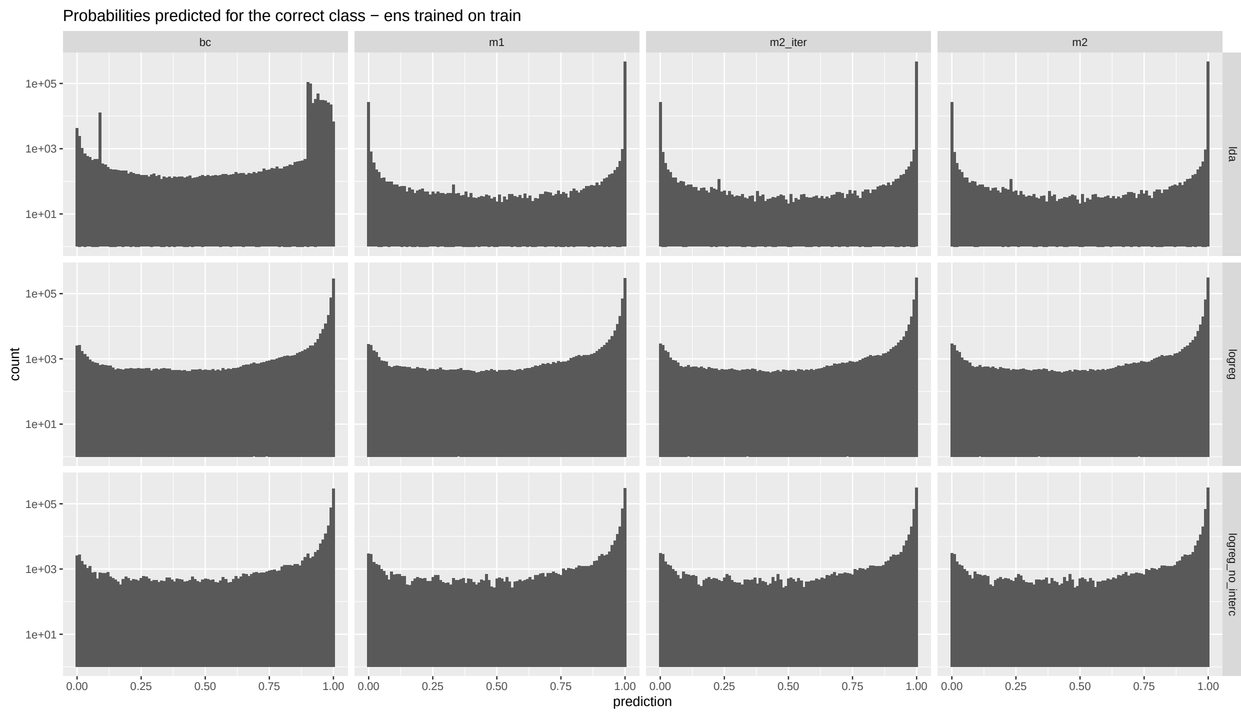
```
val_ens_zero_counts
```



m2_iter and bc didn't produce any zero probability outputs. Neither were there any zero probability outputs for combining method logistic regression.

```
train_ens_cor_probs <- gather(ens_outputs$train_training, 1 + nets_outputs$test_labels[1, ], 5, 6)
train_ens_cor_probs <- melt(train_ens_cor_probs)
train_ens_cor_probs <- train_ens_cor_probs[, c(-5, -6)]
names(train_ens_cor_probs) <- c("replication", "combining_method", "coupling_method", "fold", "predicti
train_ens_cor_probs[, c("combining_method", "coupling_method")] <- lapply(train_ens_cor_probs[, c("comb
levels(train_ens_cor_probs$combining_method) <- ens_outputs$combining_methods
levels(train_ens_cor_probs$coupling_method) <- ens_outputs$coupling_methods
```

```
train_ens_cor_preds_histo <- ggplot(data=train_ens_cor_probs) + geom_histogram(mapping=aes(x=prediction
train_ens_cor_preds_histo
```

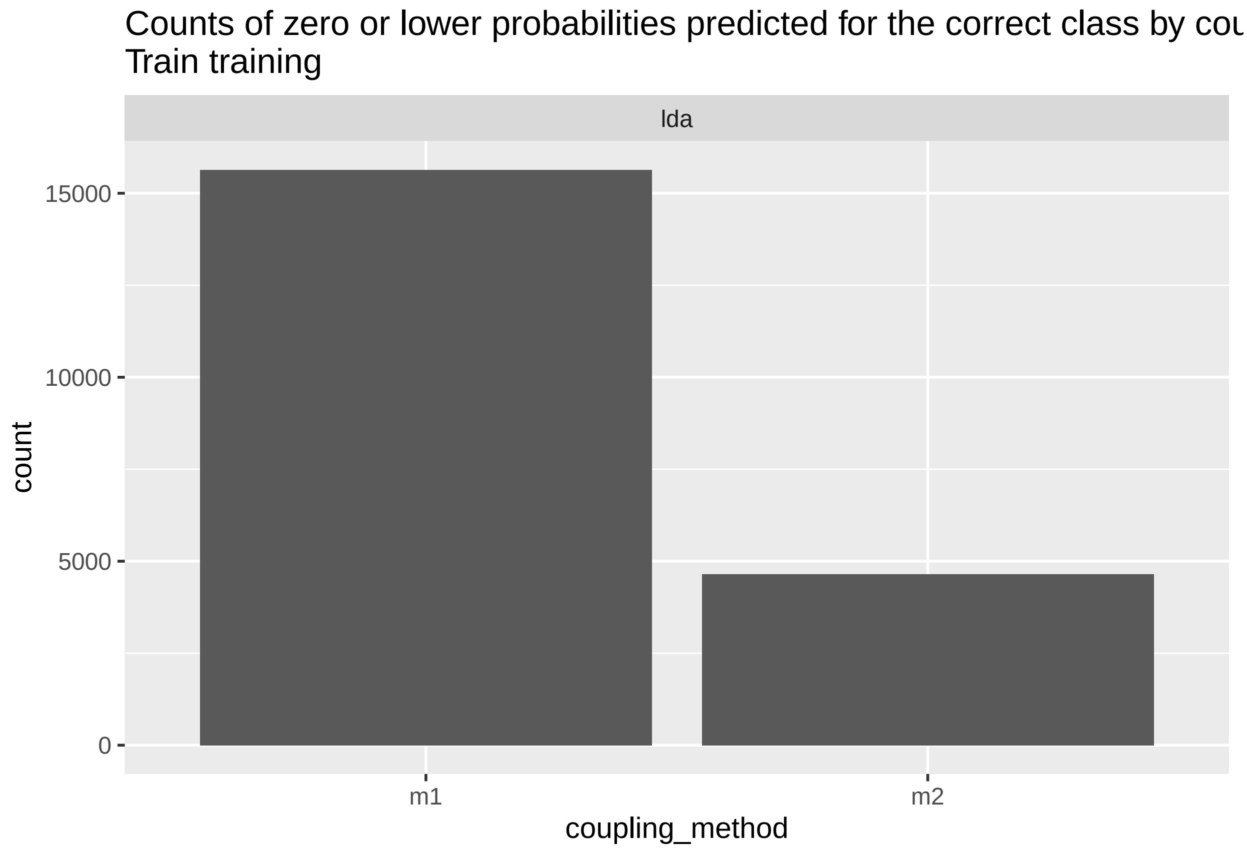


Again, in case of combining method lda, coupling method bc produces fewer probabilities falling into the lowest bin for the correct class than m1 and m2. In case of logreg without intercept, outputs are more jagged than in case with intercept.

```
train_ens_zero_counts <- ggplot(data=train_ens_cor_probs[train_ens_cor_probs$prediction <= 0, ]) + geom_
```

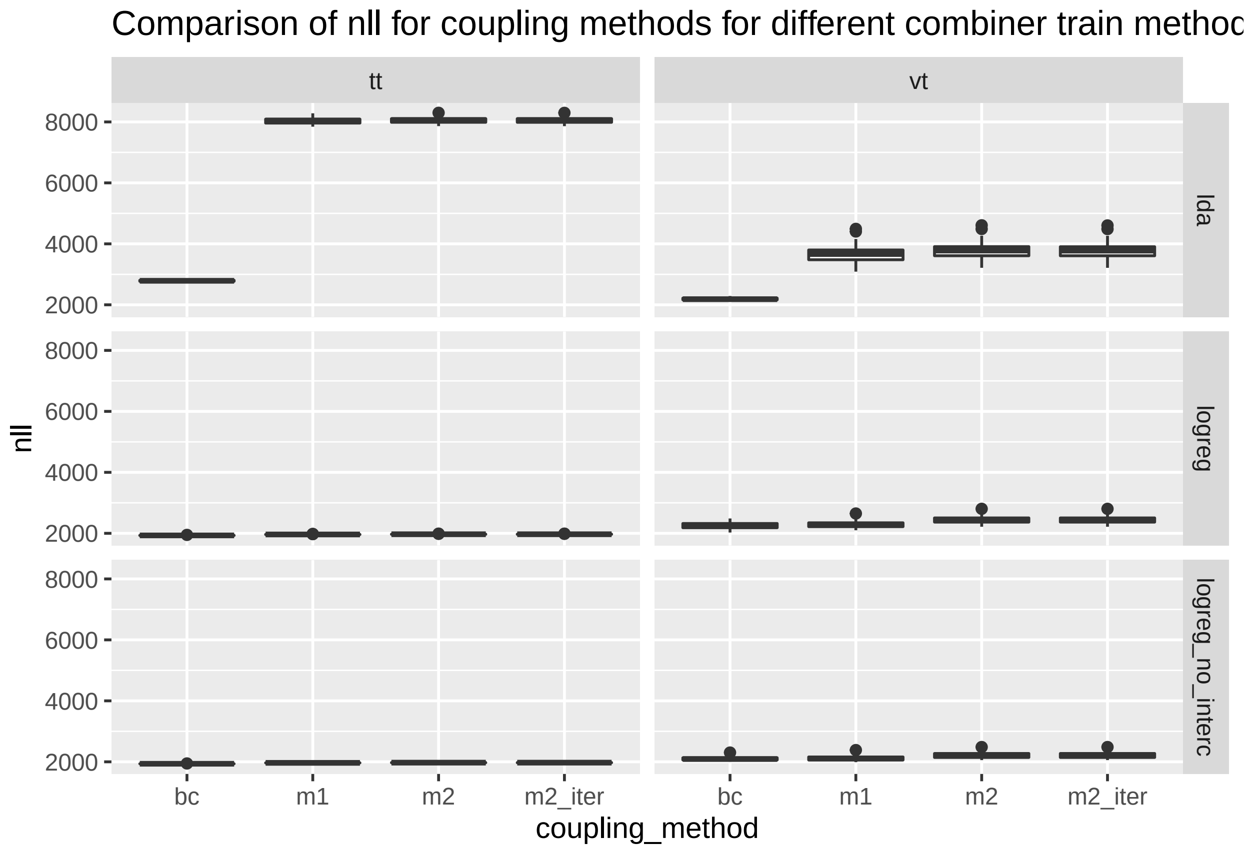
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
train_ens_zero_counts
```



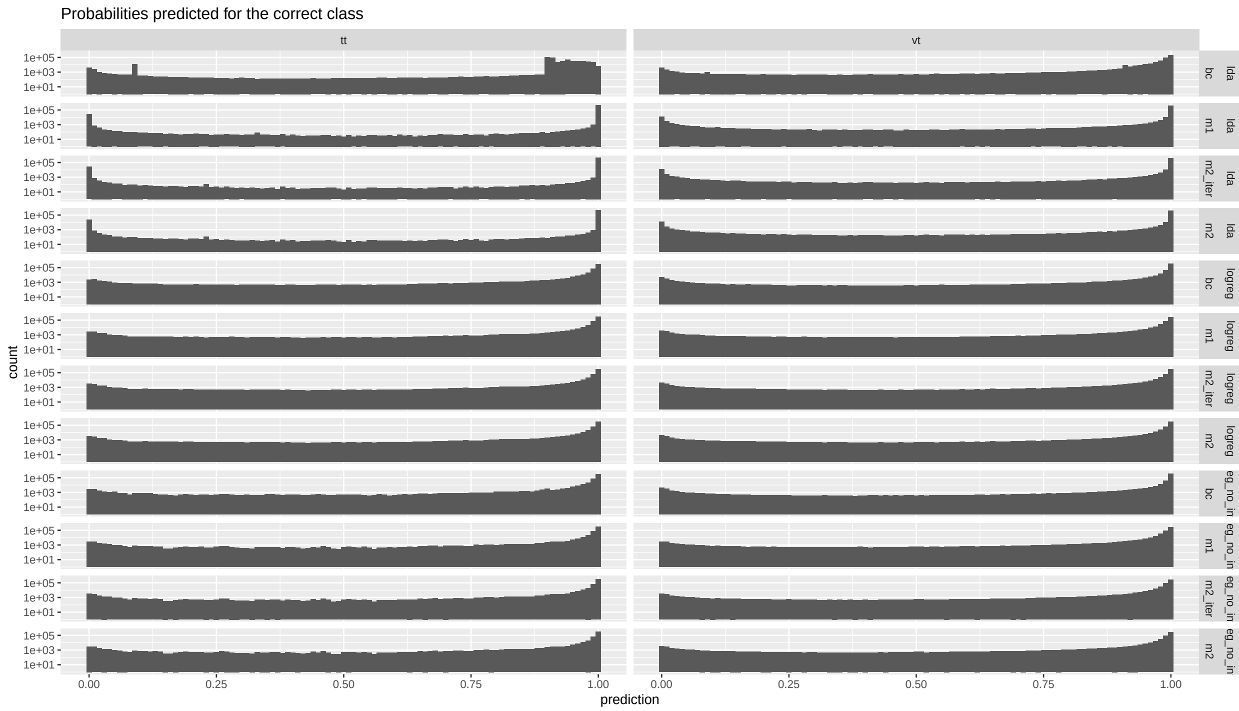
m2_iter and bc didn't produce any zero probability outputs. Neither were there any zero probability outputs for combining method logistic regression.

```
val_ens_nll <- ggplot(data=ens_results) + geom_boxplot(mapping=aes(x=coupling_method, y=nll)) + facet_g
  ggtitle("Comparison of nll for coupling methods for different combiner train methodologies")
val_ens_nll
```




```
val_ens_cor_probs$train_type <- "vt"
train_ens_cor_probs$train_type <- "tt"
ens_cor_probs <- rbind(val_ens_cor_probs, train_ens_cor_probs)
```

```
ens_cor_preds_histo <- ggplot(data=ens_cor_probs) + geom_histogram(mapping=aes(x=prediction), binwidth=(
ens_cor_preds_histo
```



Bayes covariant coupling method produces more smoothly distributed predictions than methods m1 and m2. Also, there is a big difference in each method between ensemble trained on validation and ensemble trained on train set. Ensembles trained on validation set produce generally more smoothly distributed predictions. This is mainly visible for combining methods lda and logreg without intercept. However, ensembles trained on training set attain statistically significantly higher accuracy. Similar results to those in visualizations_ensemble_outputs_CIF10.

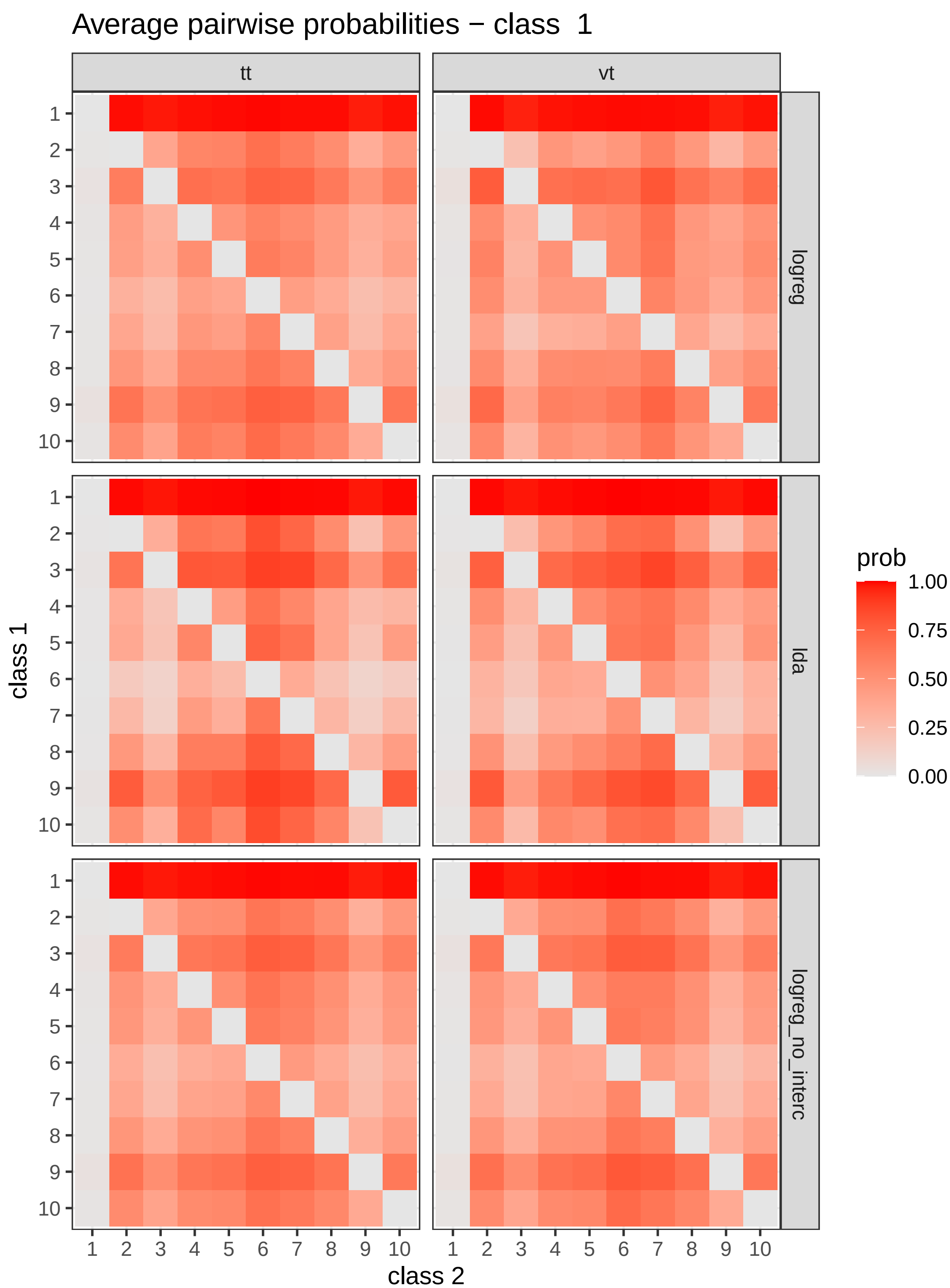
```
df_val_Rs <- melt(np$load(file.path(base_dir, "val_training_class_aggr_R.npy")))
df_train_Rs <- melt(np$load(file.path(base_dir, "train_training_class_aggr_R.npy")))
co_m_R <- read.csv(file.path(base_dir, "R_mat_co_m_names.csv"), header=FALSE)
```

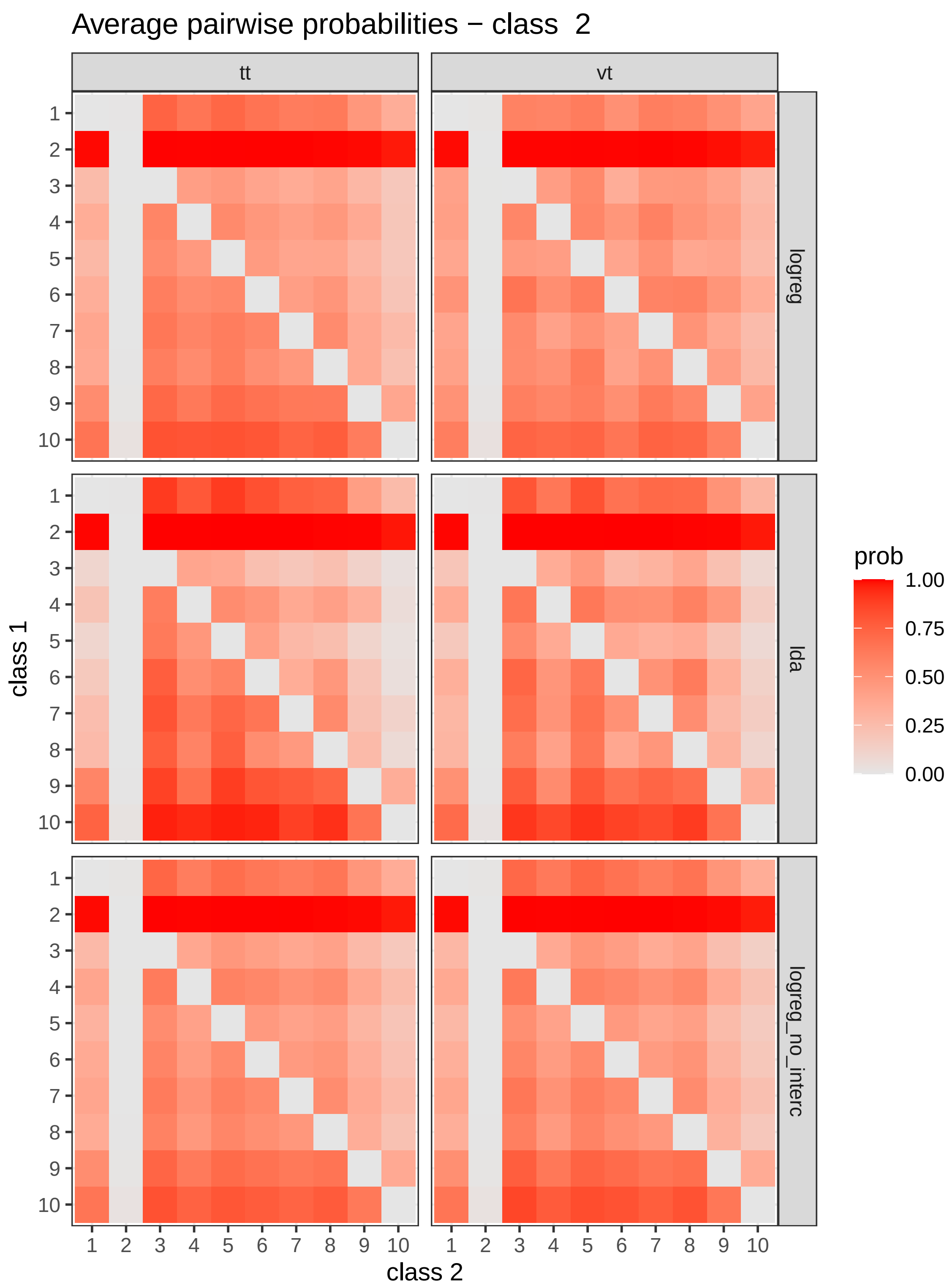
```
names(df_val_Rs) <- c("combining_method", "precision", "class", "class1", "class2", "prob")
names(df_train_Rs) <- c("combining_method", "precision", "class", "class1", "class2", "prob")

df_val_Rs[,c("class", "class1", "class2", "combining_method")] <- lapply(df_val_Rs[,c("class", "class1"
df_train_Rs[,c("class", "class1", "class2", "combining_method")] <- lapply(df_train_Rs[,c("class", "cla

levels(df_val_Rs$combining_method) <- co_m_R$V1
levels(df_train_Rs$combining_method) <- co_m_R$V1
```

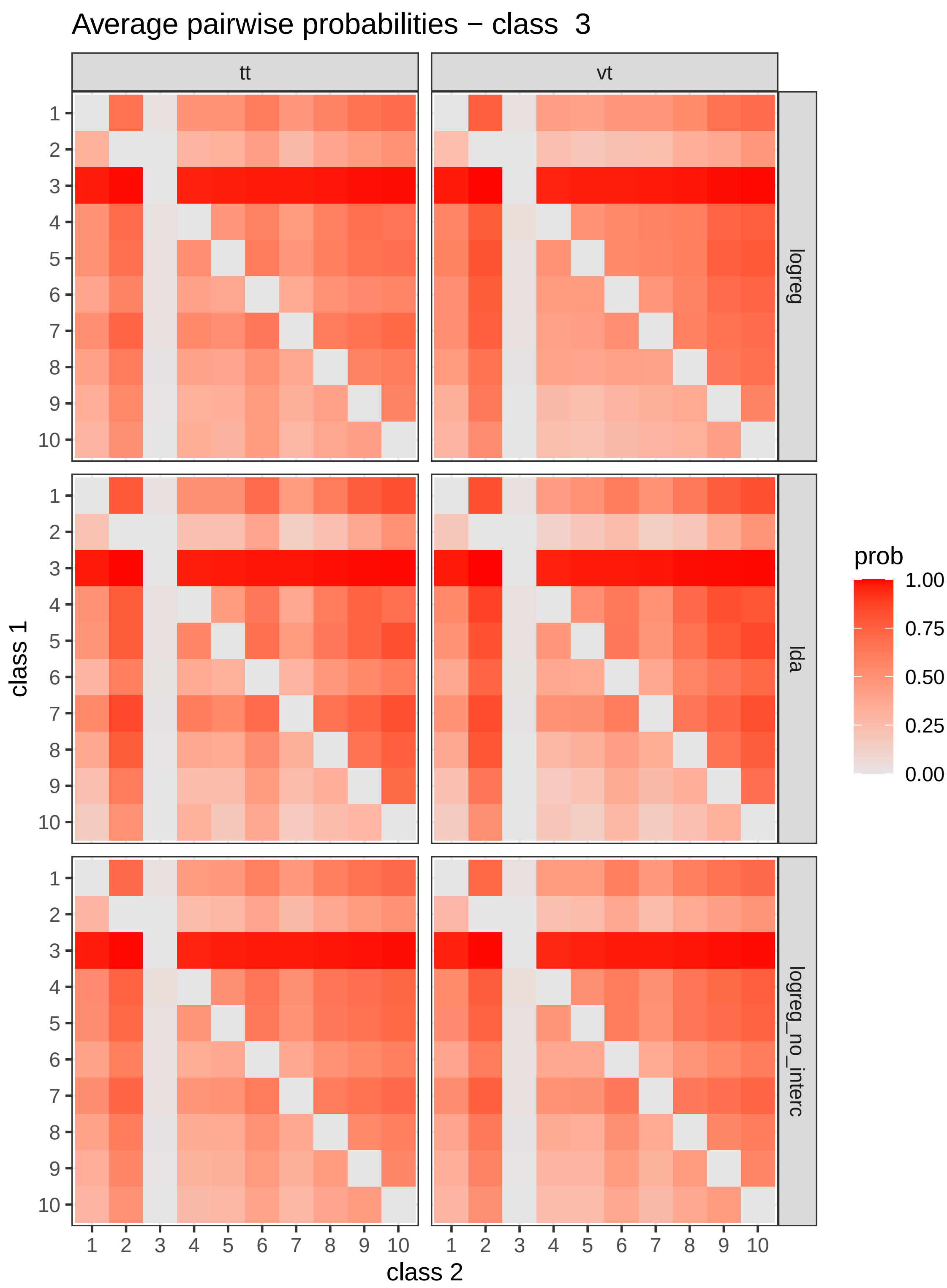
```
df_val_Rs$train_type <- "vt"
df_train_Rs$train_type <- "tt"
class_mean_Rs <- rbind(df_val_Rs, df_train_Rs)

df_aggr_Rs_diff <- class_mean_Rs %>% pivot_wider(names_from = train_type, values_from = prob) %>% mutate
```

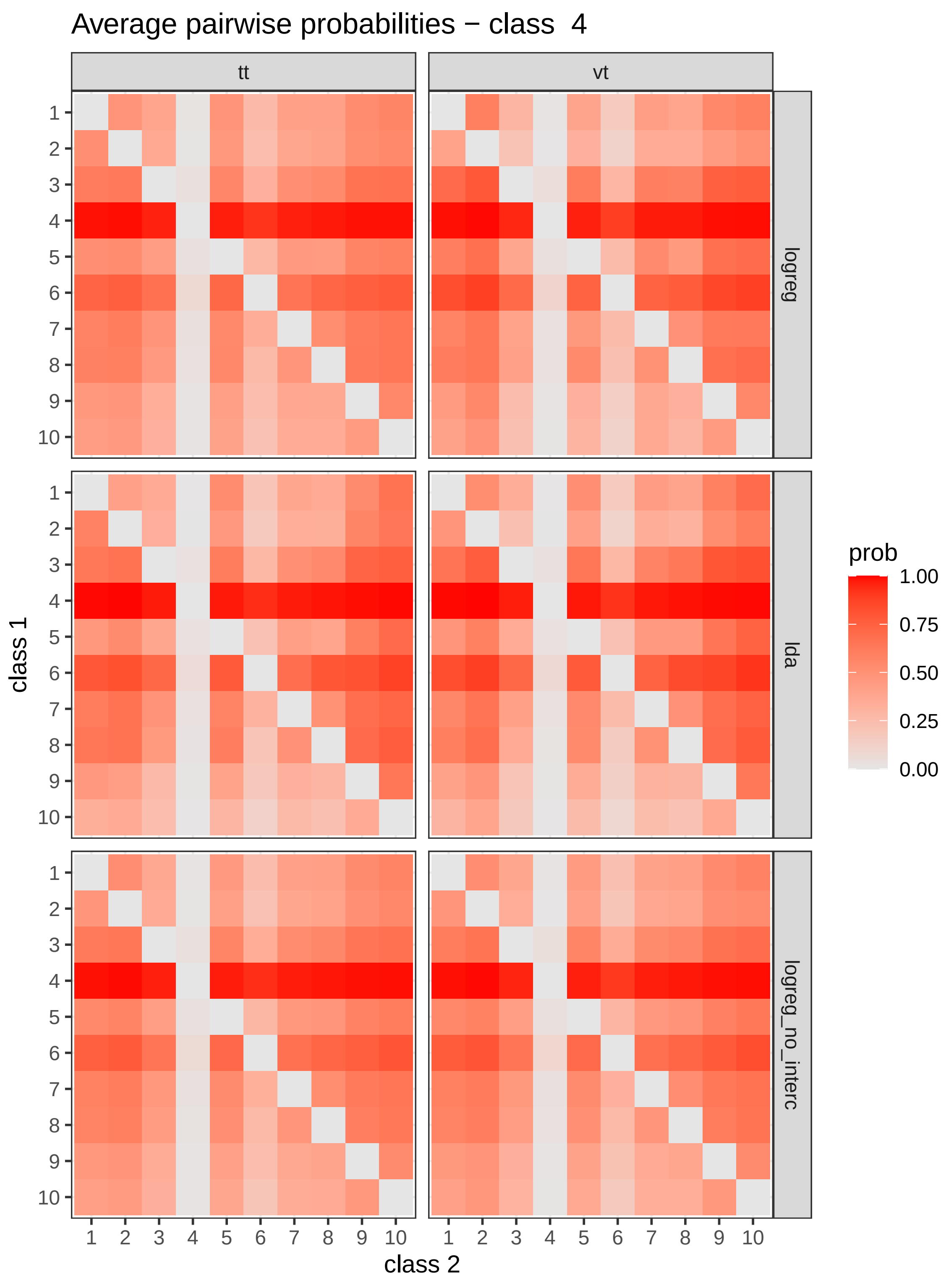
```
for (cls in 1:classes)
{
  cur_class_Rs <- class_mean_Rs %>% filter(class == cls)
  plot_cls <-  ggplot(cur_class_Rs, aes(x = class2, y = class1)) +
```

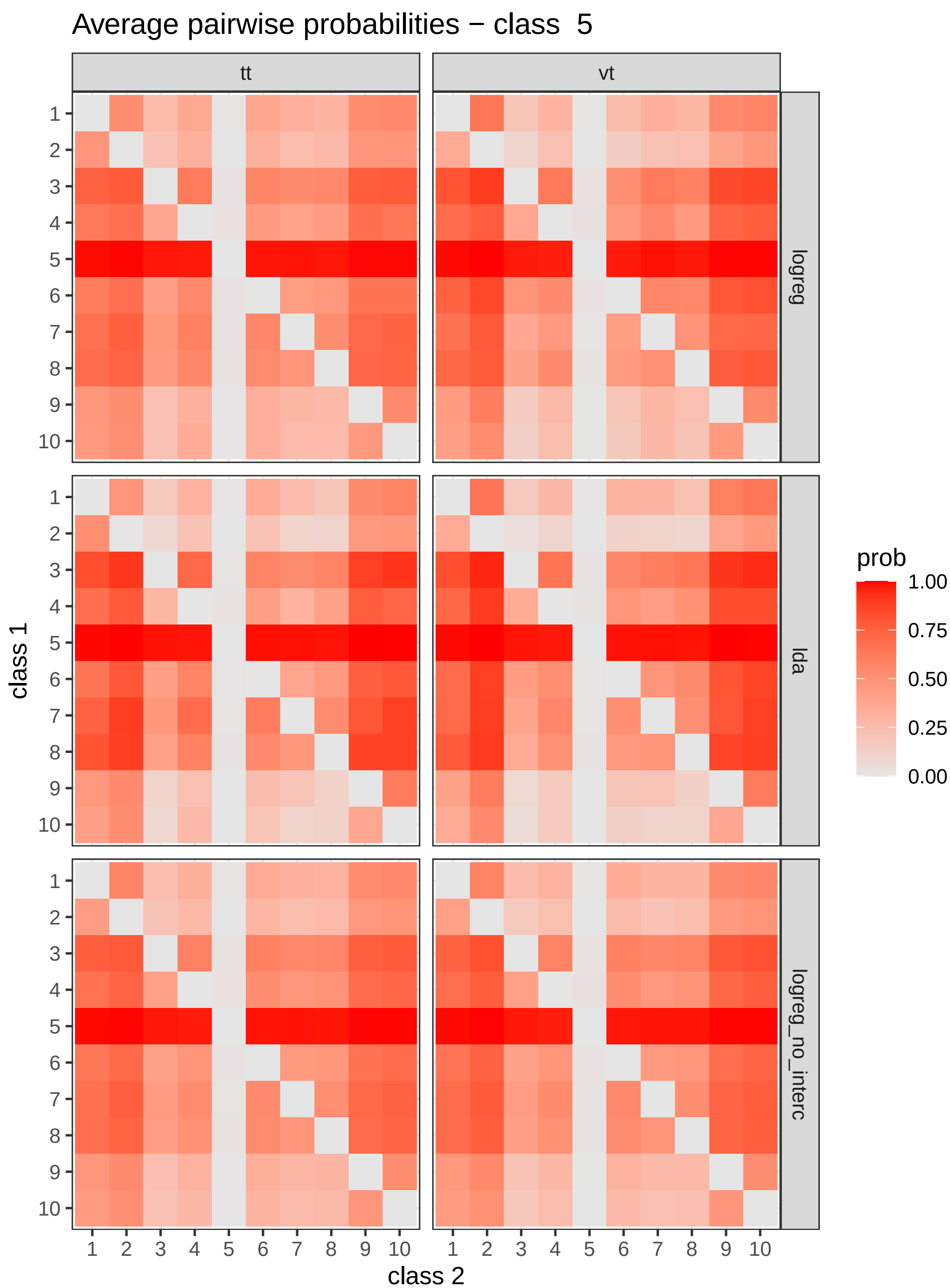
```
    geom_raster(aes(fill=prob)) +
    facet_grid(rows=vars(combining_method), cols=vars(train_type)) +
    scale_fill_gradient(low="grey90", high="red", limits=c(0, 1)) +
    scale_y_discrete(limits=rev) +
    labs(x="class 2", y="class 1", title=paste("Average pairwise probabilities - class ", cls)) +
    theme_bw()

  print(plot_cls)
}
```

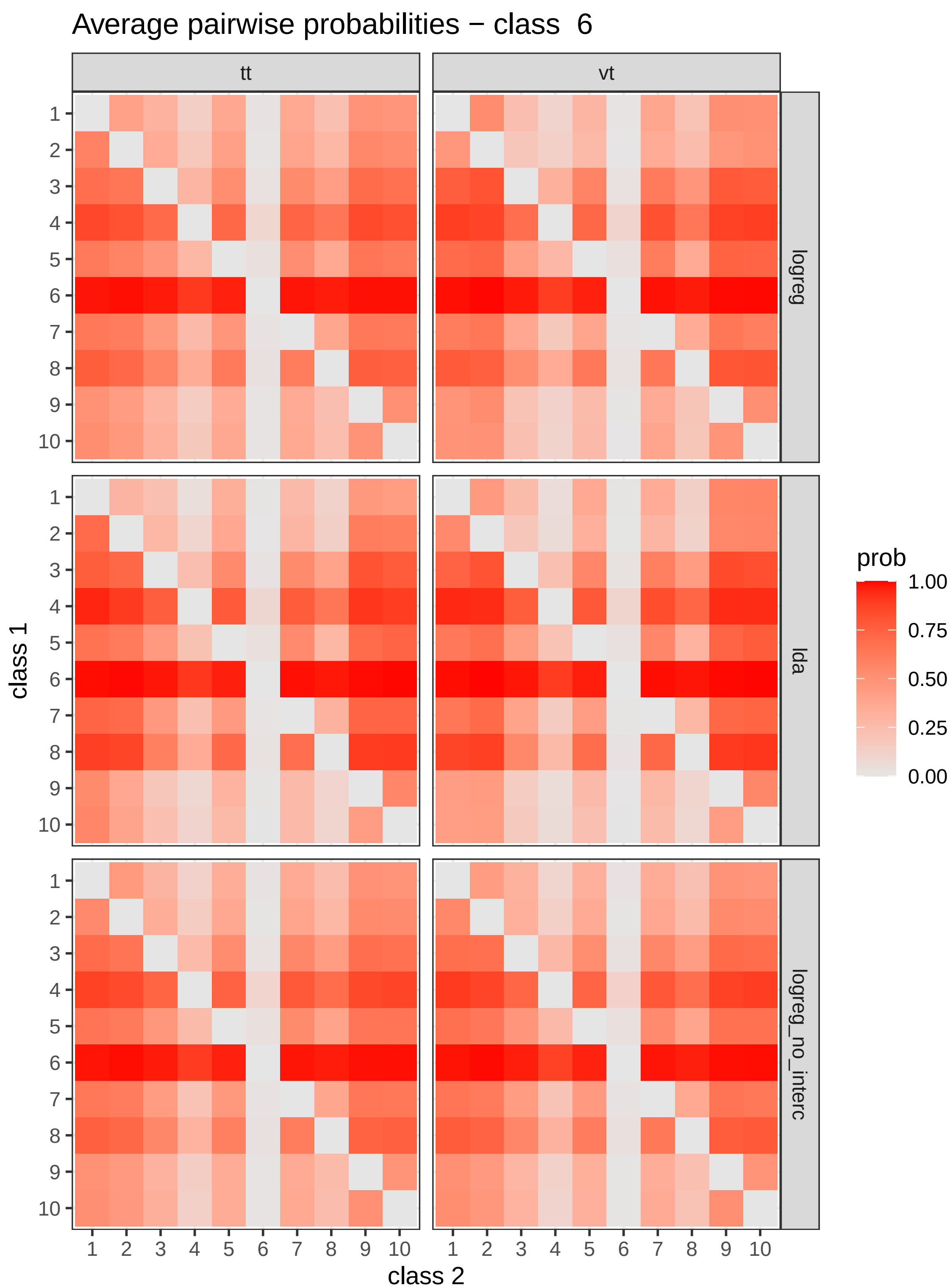

Average pairwise probabilities − class 1
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
1
2
3
4
5
6
7
8
9
10
prob
1.00
0.75
0.50
0.25
0.00

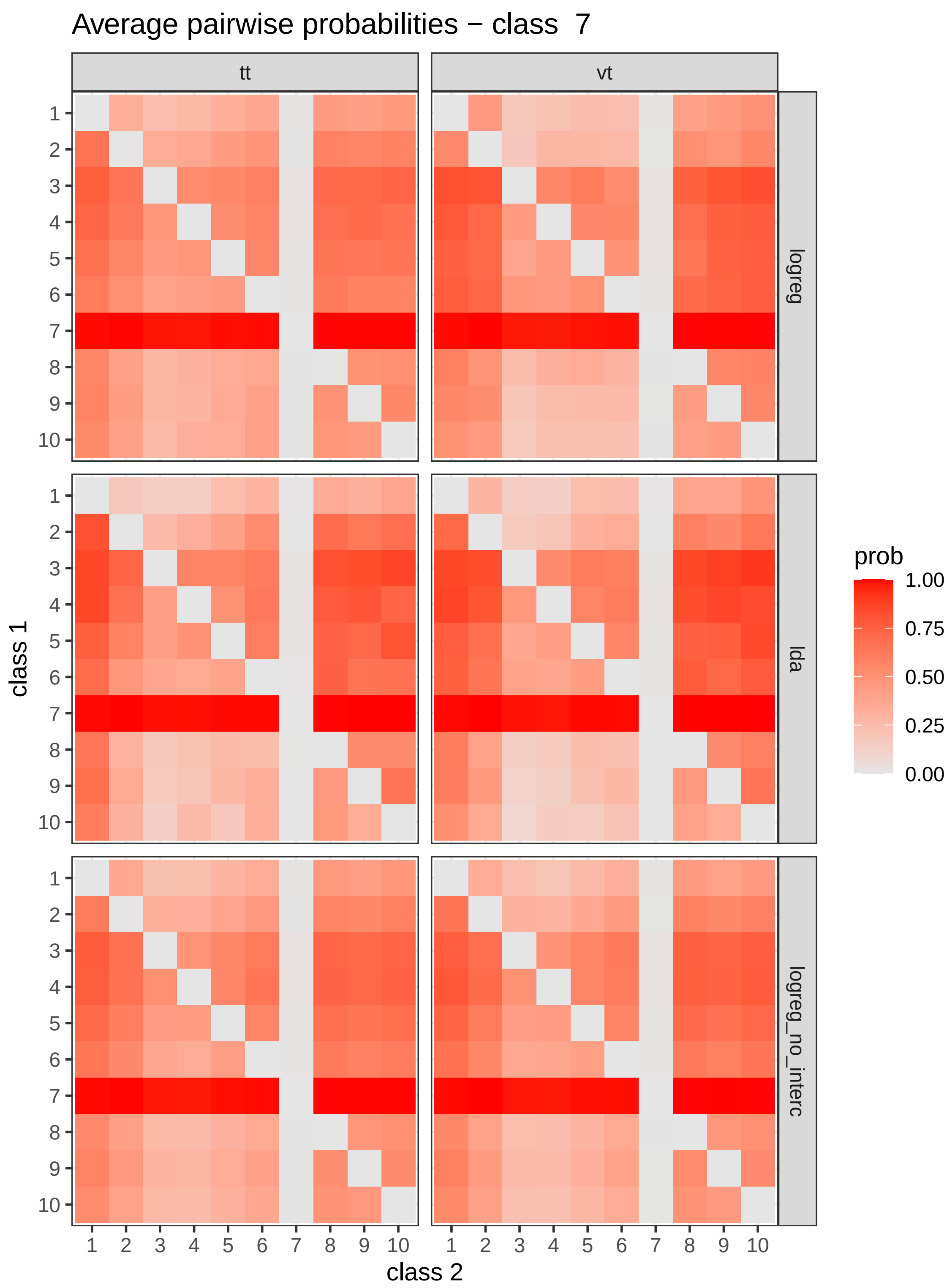

Average pairwise probabilities − class 2
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
prob
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10

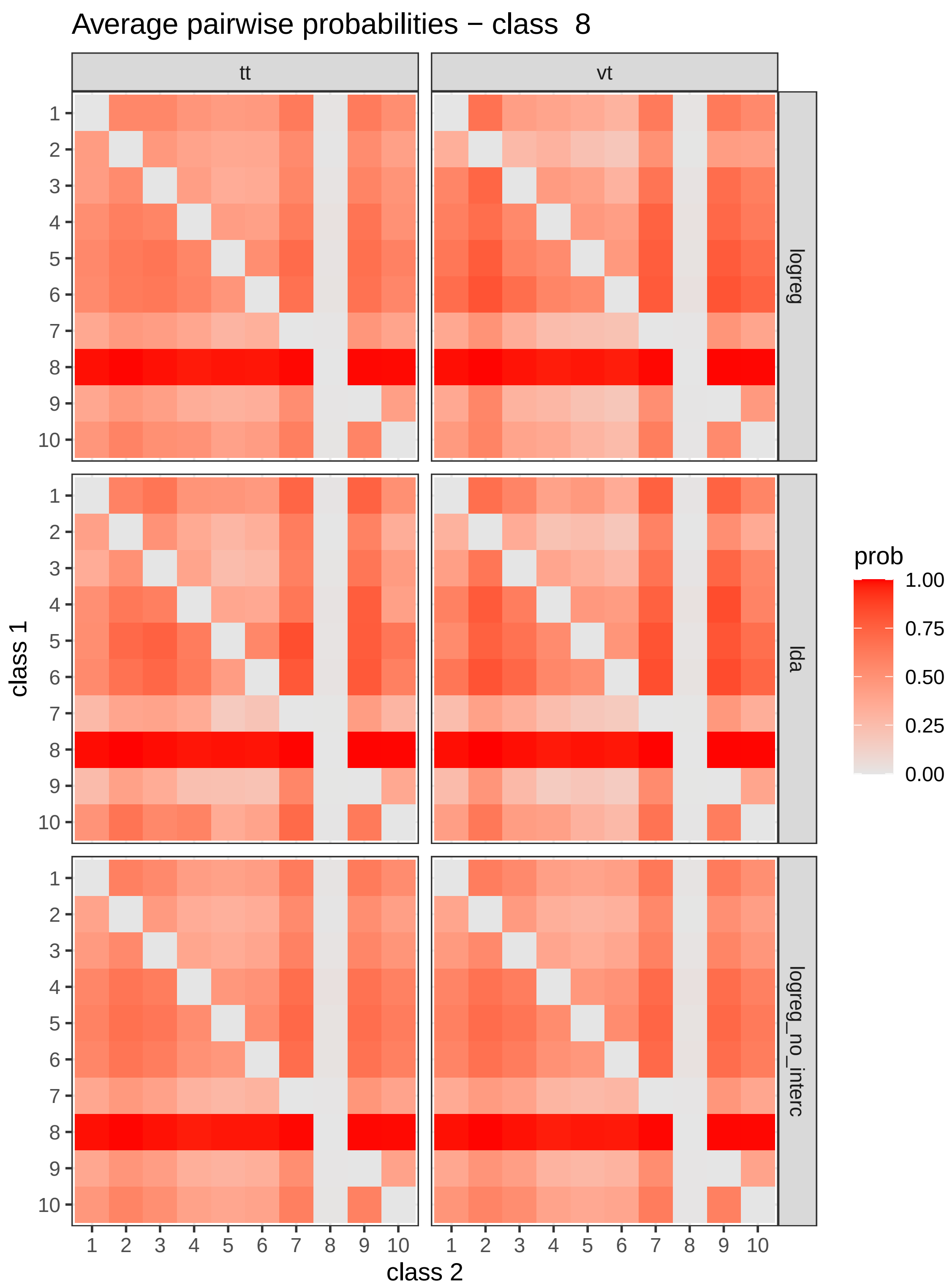

Average pairwise probabilities − class 3
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
prob
1.00
0.75
0.50
0.25
0.00

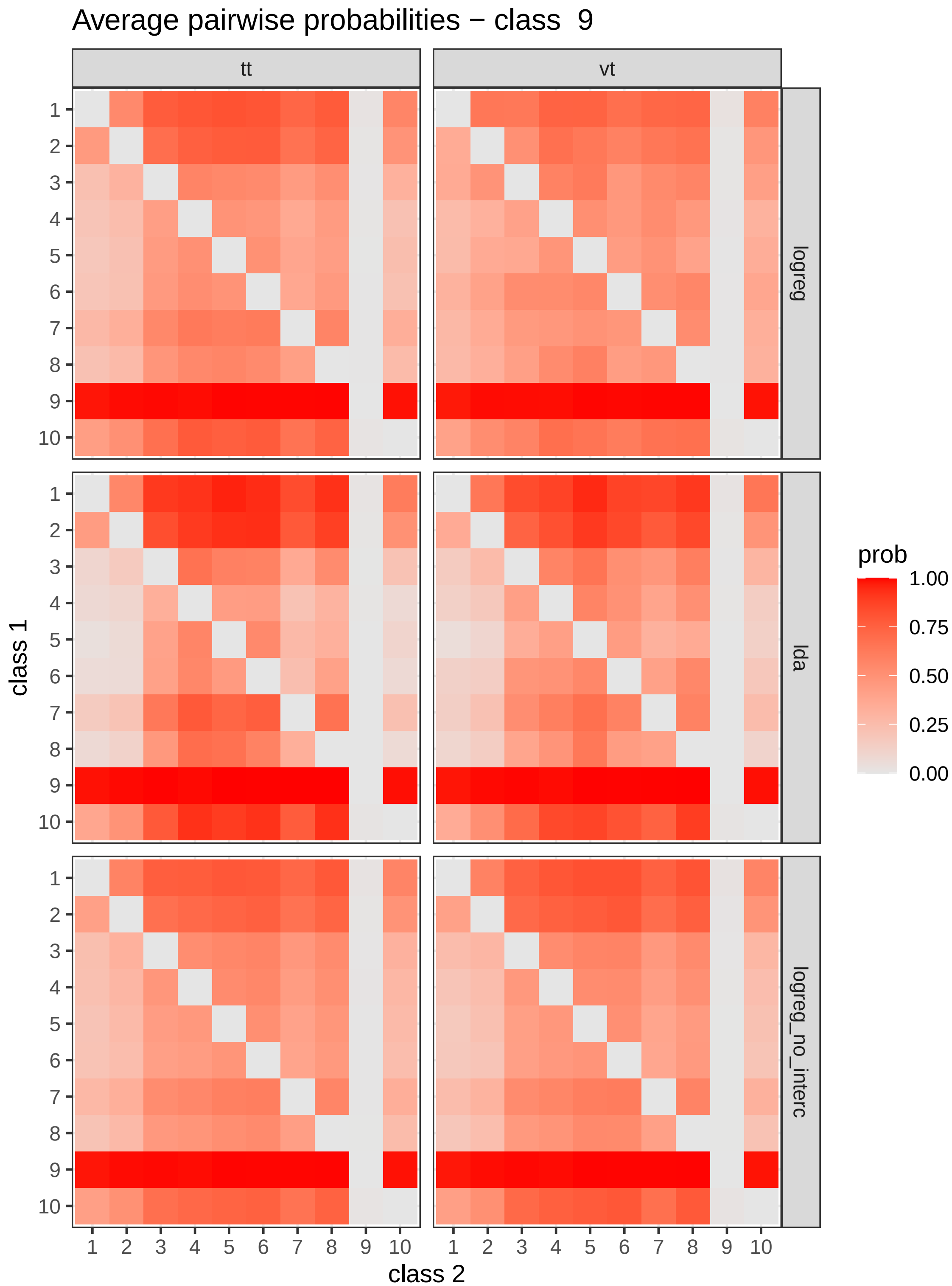

Average pairwise probabilities − class 4
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
prob
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10

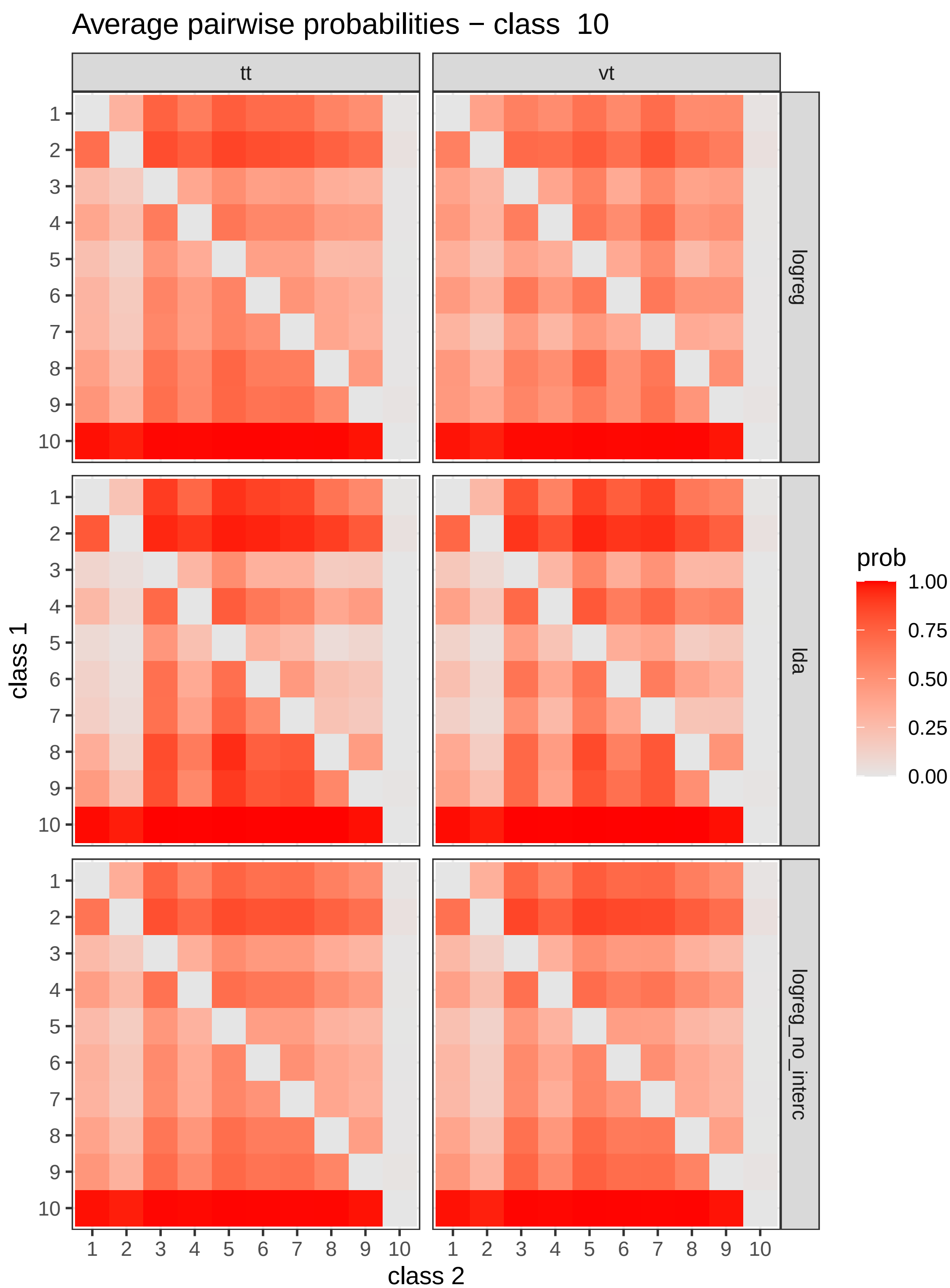

Average pairwise probabilities − class 5
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
prob
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10


Average pairwise probabilities − class 6
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
prob
1.00
0.75
0.50
0.25
0.00


Average pairwise probabilities − class 7
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
prob
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10


Average pairwise probabilities − class 8
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
prob
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10

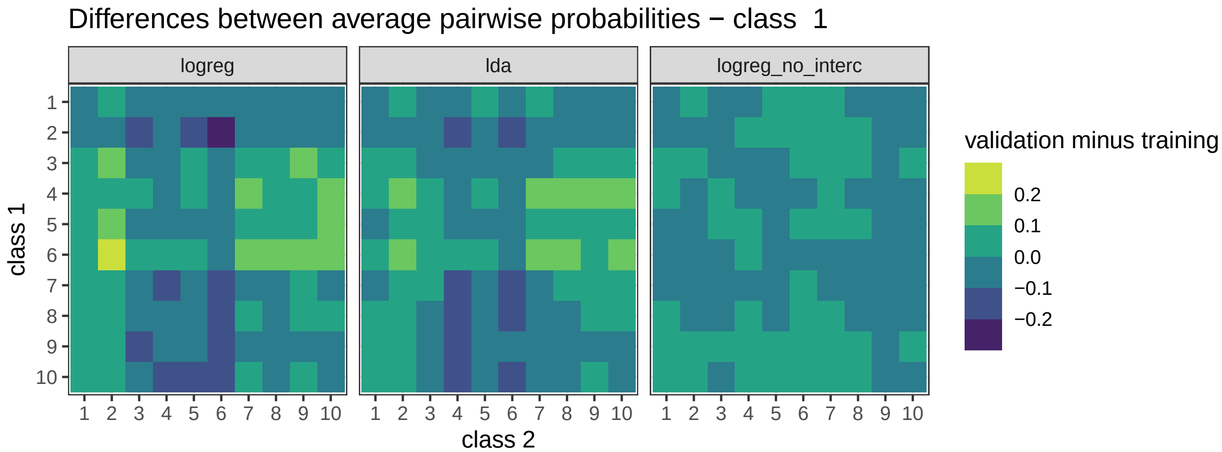

Average pairwise probabilities − class 9
tt
vt
logreg
lda
logreg_no_interc
class 1
class 2
1
2
3
4
5
6
7
8
9
10
prob
1.00
0.75
0.50
0.25
0.00

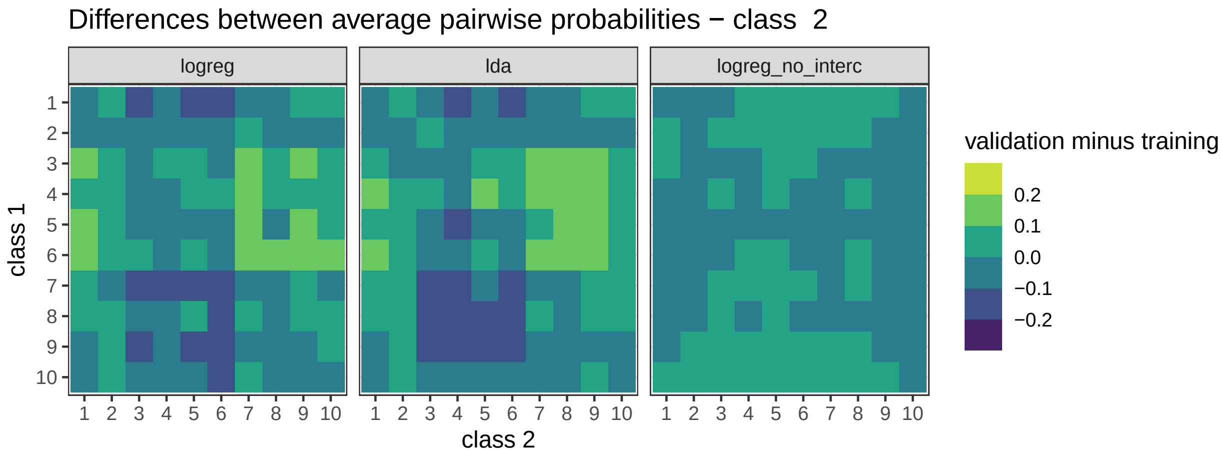


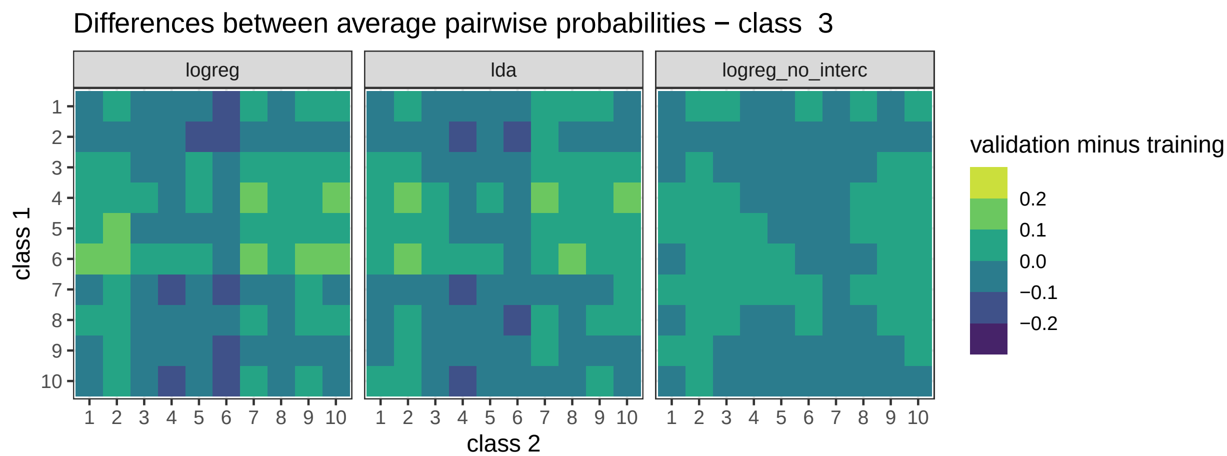
lda has more noise than logreg. However, it is less obvious than in case of training on almost full cifar10 train set.

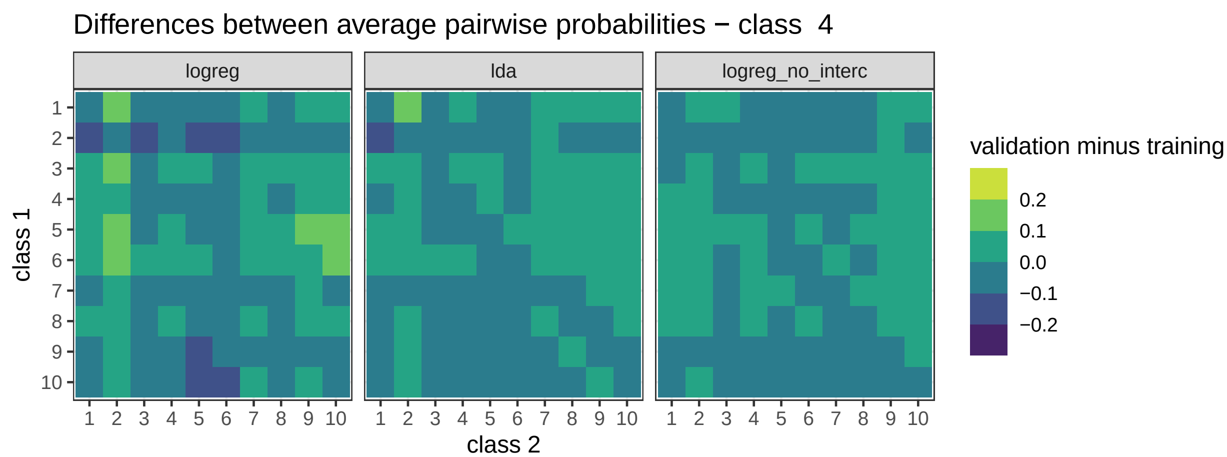
```
for (cls in 1:classes)
{
  cur_class_Rs <- df_aggr_Rs_diff %>% filter(class == cls)
```

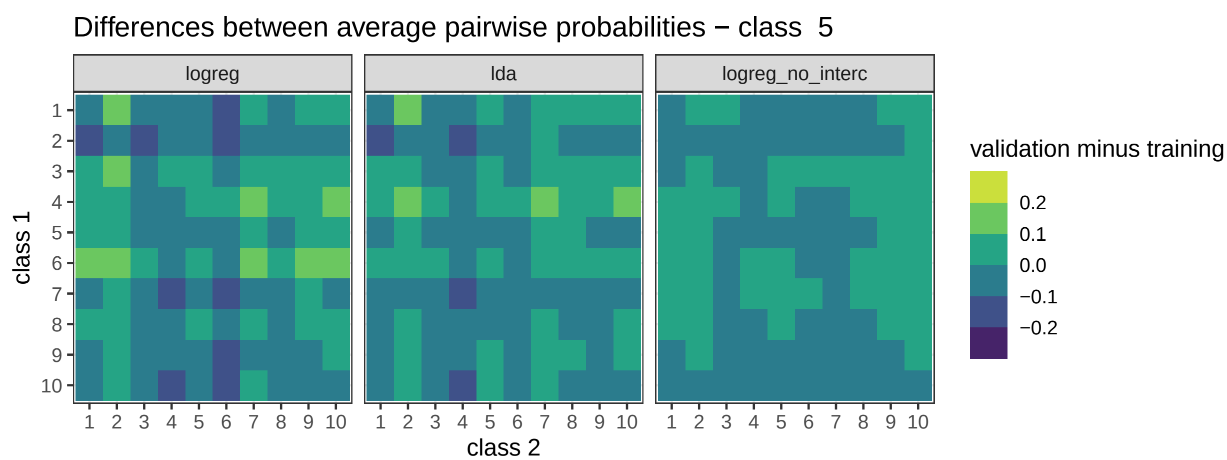
```
  plot_cls <-  ggplot(cur_class_Rs, aes(x = class2, y = class1)) +
    geom_raster(aes(fill=val_min_train)) +
    facet_wrap(~combining_method) +
    scale_fill_binned(type="viridis", limits=c(-0.3, 0.3), name="validation minus training") +
    scale_y_discrete(limits=rev) +
    labs(x="class 2", y="class 1", title=paste("Differences between average pairwise probabilities - cla
    theme_bw()

  print(plot_cls)
}
```

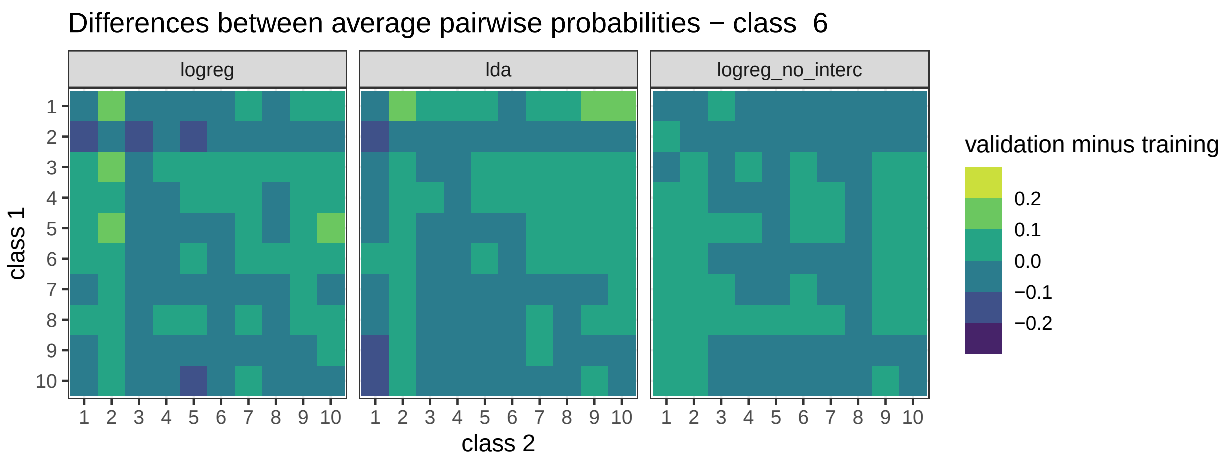


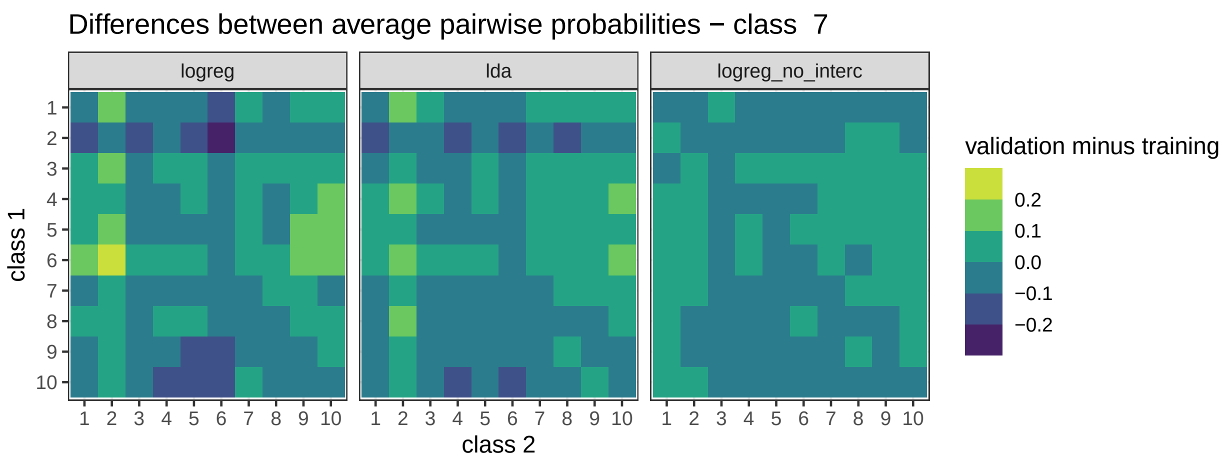


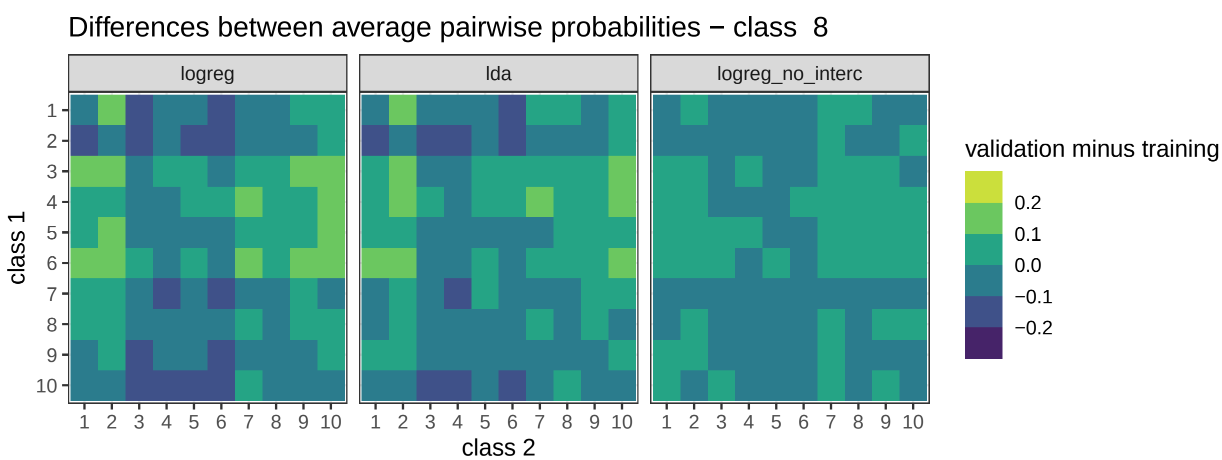

Differences between average pairwise probabilities − class 3
logreg
lda
logreg_no_interc
class 1
class 2
validation minus training
0.2
0.1
0.0
−0.1
−0.2

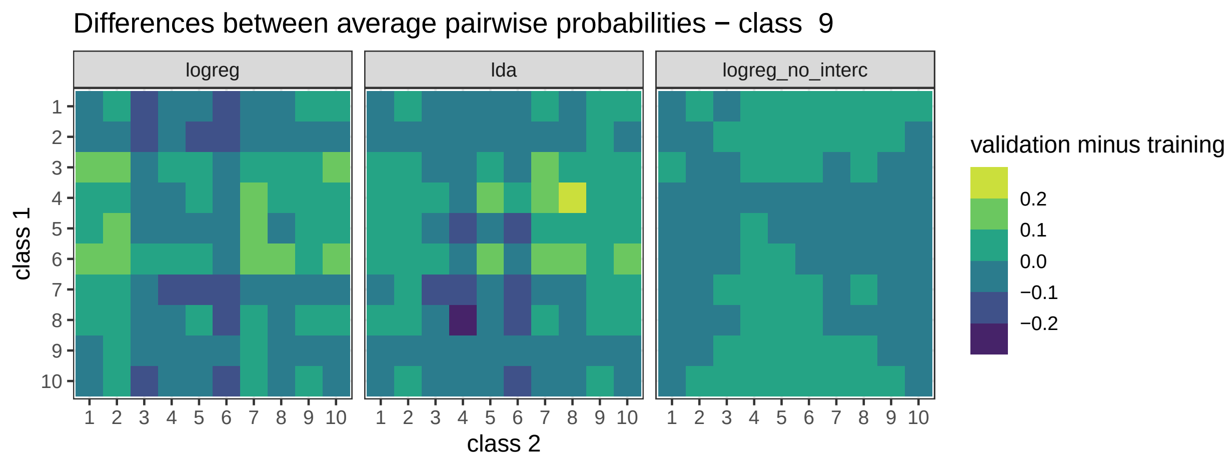

Differences between average pairwise probabilities − class 4
logreg
lda
logreg_no_interc
class 1
class 2
validation minus training
0.2
0.1
0.0
−0.1
−0.2

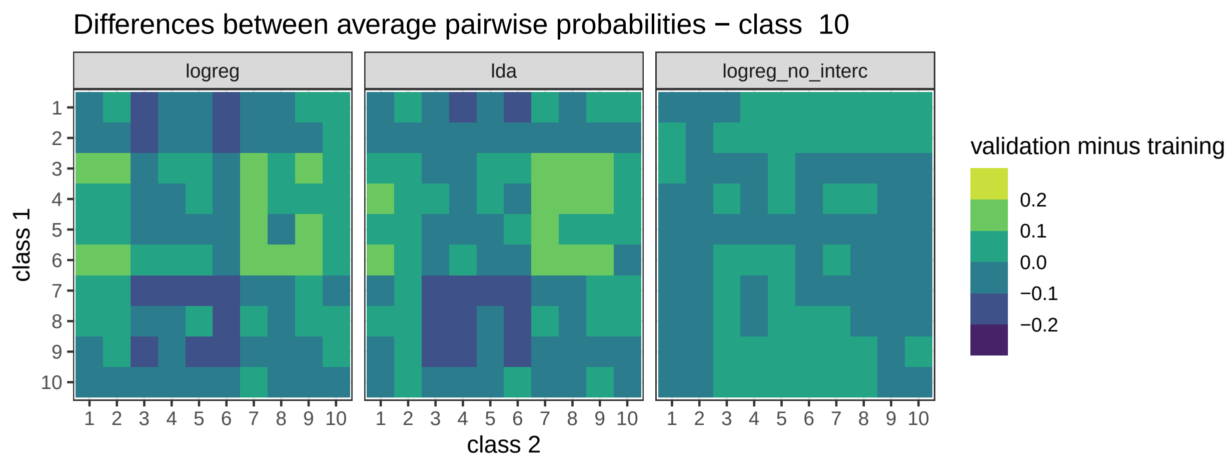

Differences between average pairwise probabilities − class 5
logreg
lda
logreg_no_interc
class 1
class 2
validation minus training
0.2
0.1
0.0
−0.1
−0.2


Differences between average pairwise probabilities − class 6
logreg
lda
logreg_no_interc
class 1
class 2
1 2 3 4 5 6 7 8 9 10
validation minus training
0.2
0.1
0.0
−0.1
−0.2


Differences between average pairwise probabilities − class 7
logreg
lda
logreg_no_interc
class 1
class 2
1 2 3 4 5 6 7 8 9 10
validation minus training
0.2
0.1
0.0
−0.1
−0.2


Differences between average pairwise probabilities − class 8
logreg
lda
logreg_no_interc
class 1
class 2
1 2 3 4 5 6 7 8 9 10
validation minus training
0.2
0.1
0.0
−0.1
−0.2

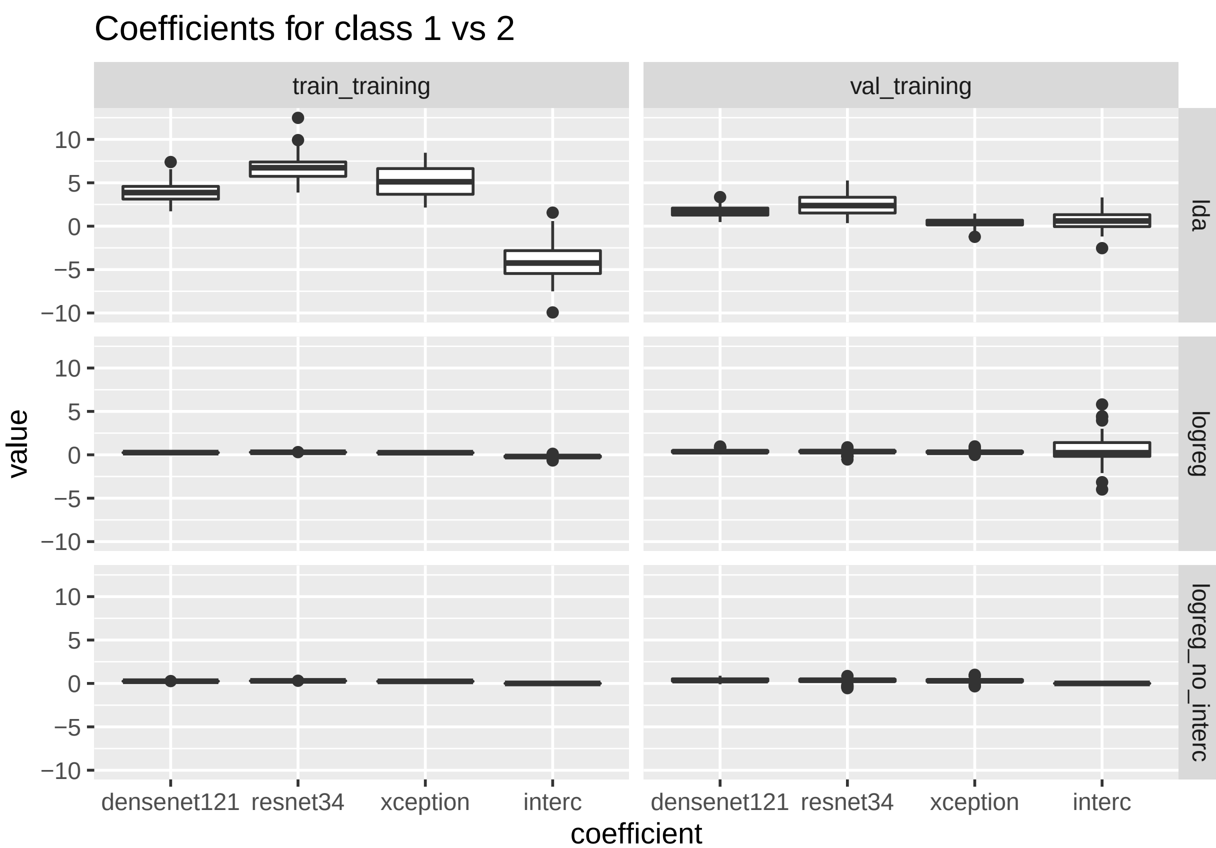


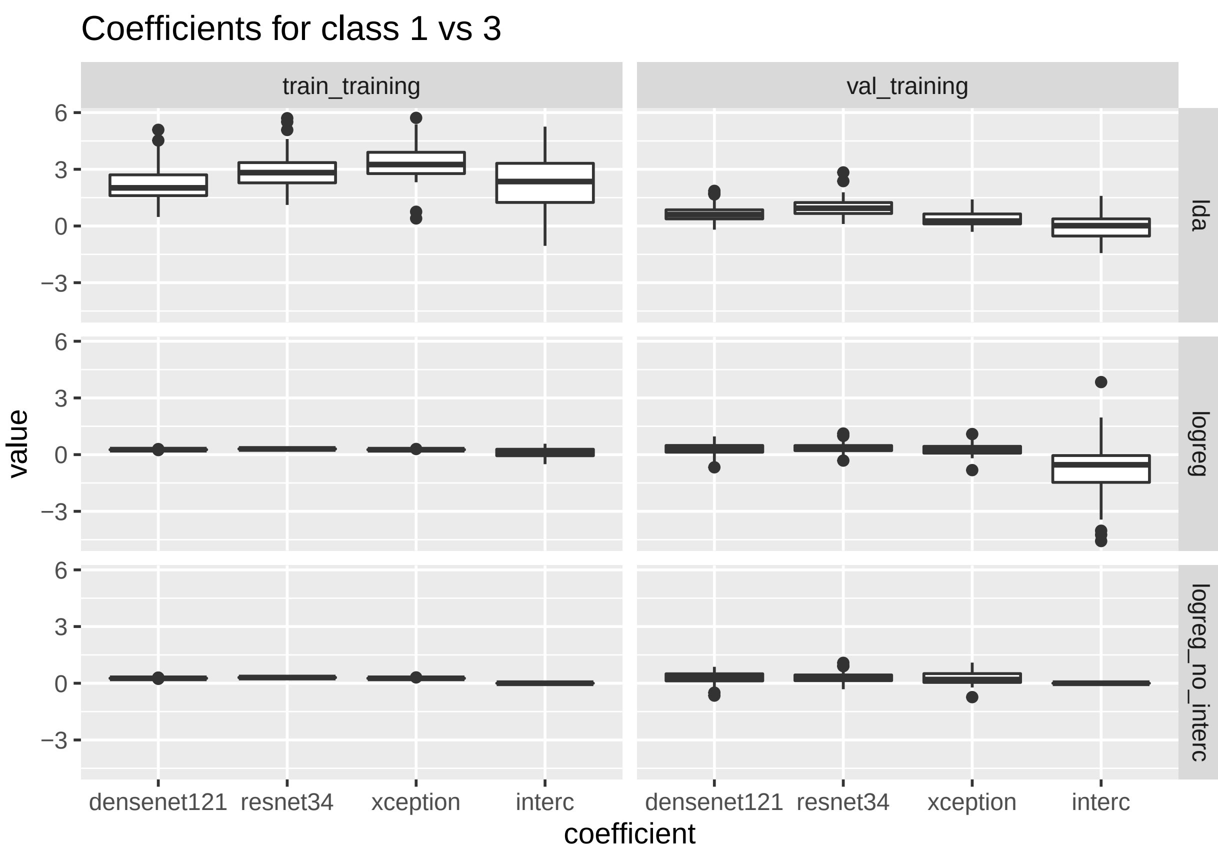


Logistic regression without intercept has lower differences between tt and vt R matrices than other combining methods.

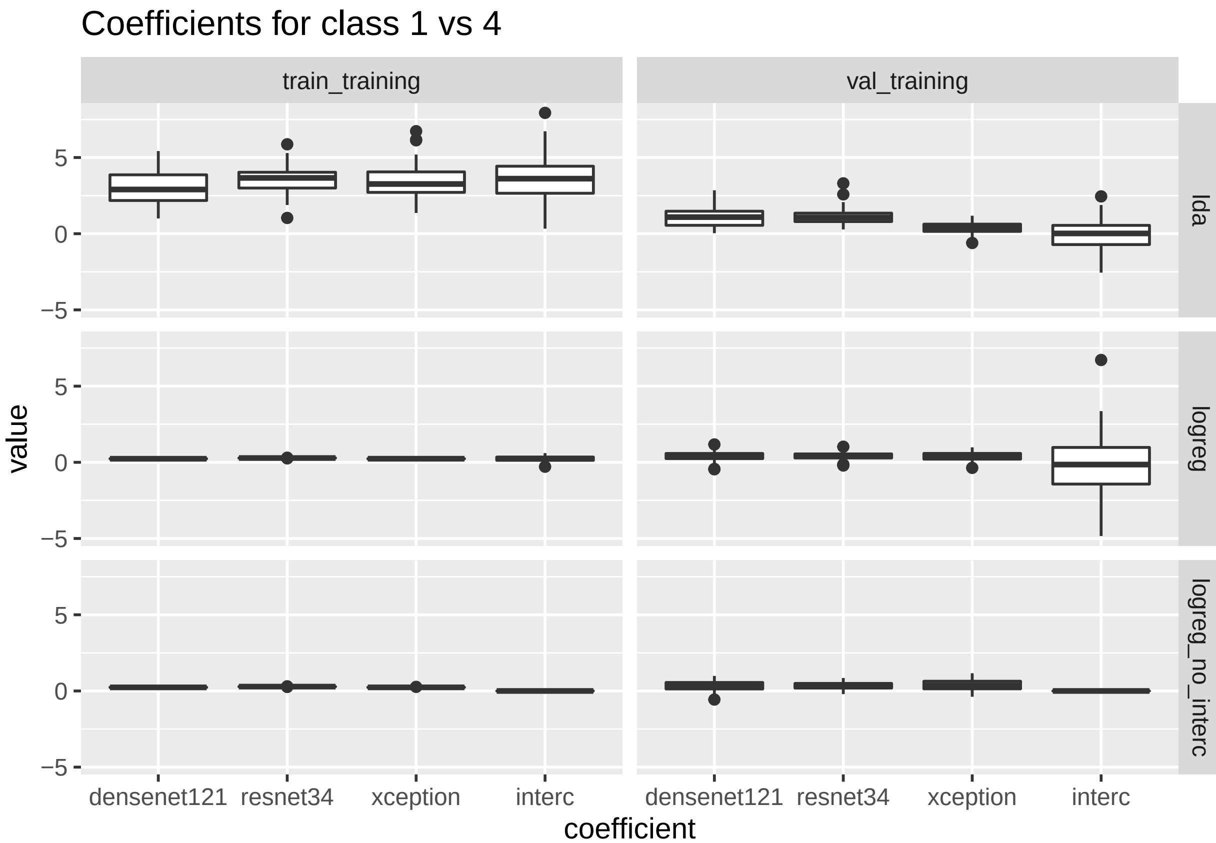
```
combiner_coefs <- load_combiner_coefs(base_dir, repls, folds)
```

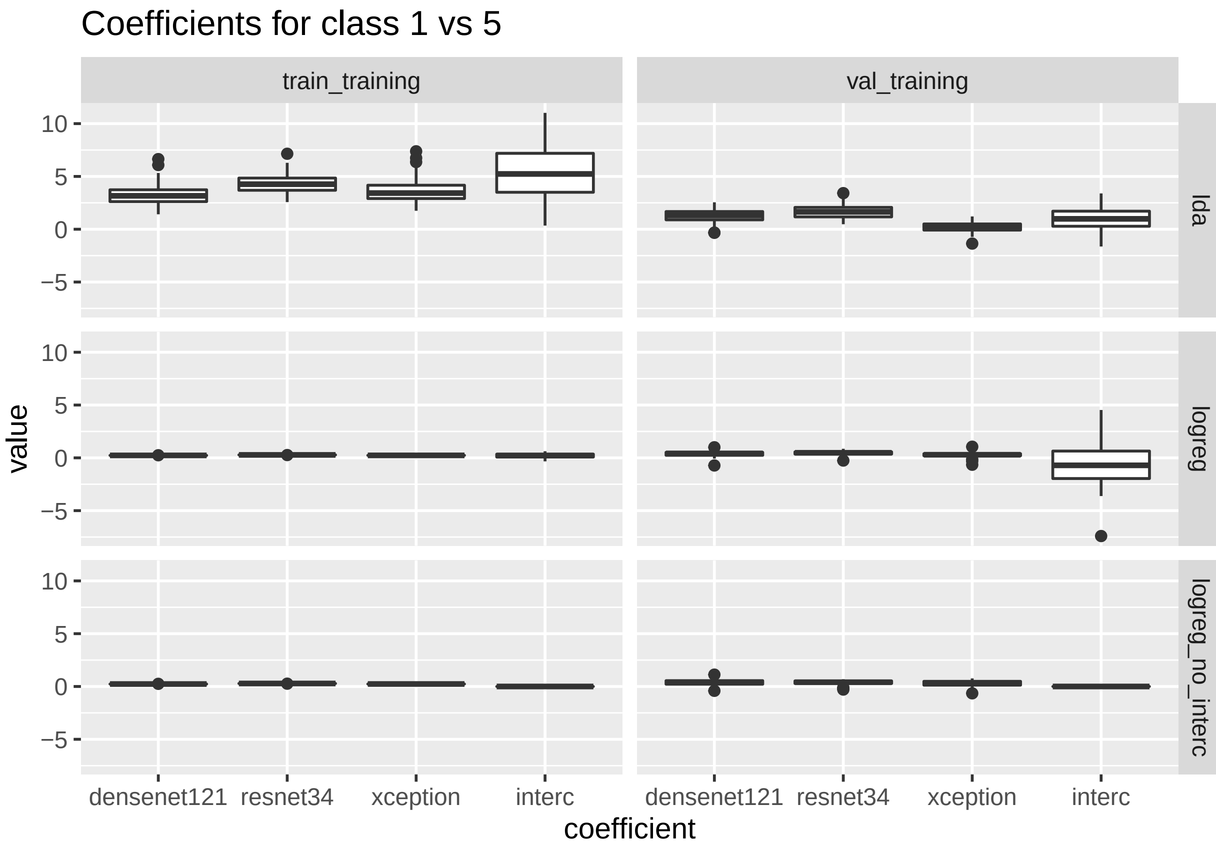
```
for (cl1 in 1:(classes - 1))
{
  for (cl2 in (cl1 + 1):classes)
  {
    cur_plt <- combiner_coefs %>% filter(class1 == cl1 & class2 == cl2) %>% ggplot() + geom_boxplot(aes
      facet_grid(cols=vars(train_type), rows=vars(combining_method)) + ggtitle(paste("Coefficients for c
    print(cur_plt)
  }
}
```

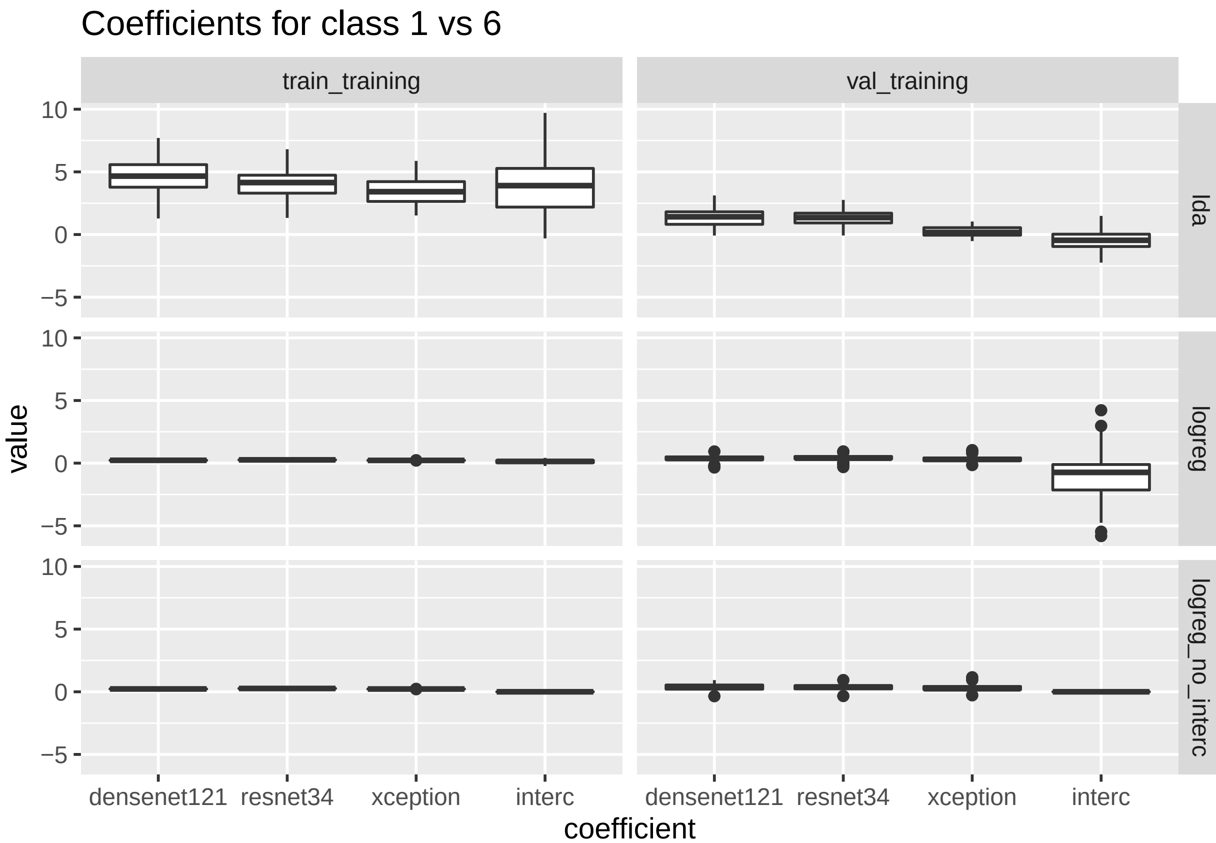

Coefficients for class 1 vs 2
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
−10
densenet121
resnet34
xception
interc
coefficient



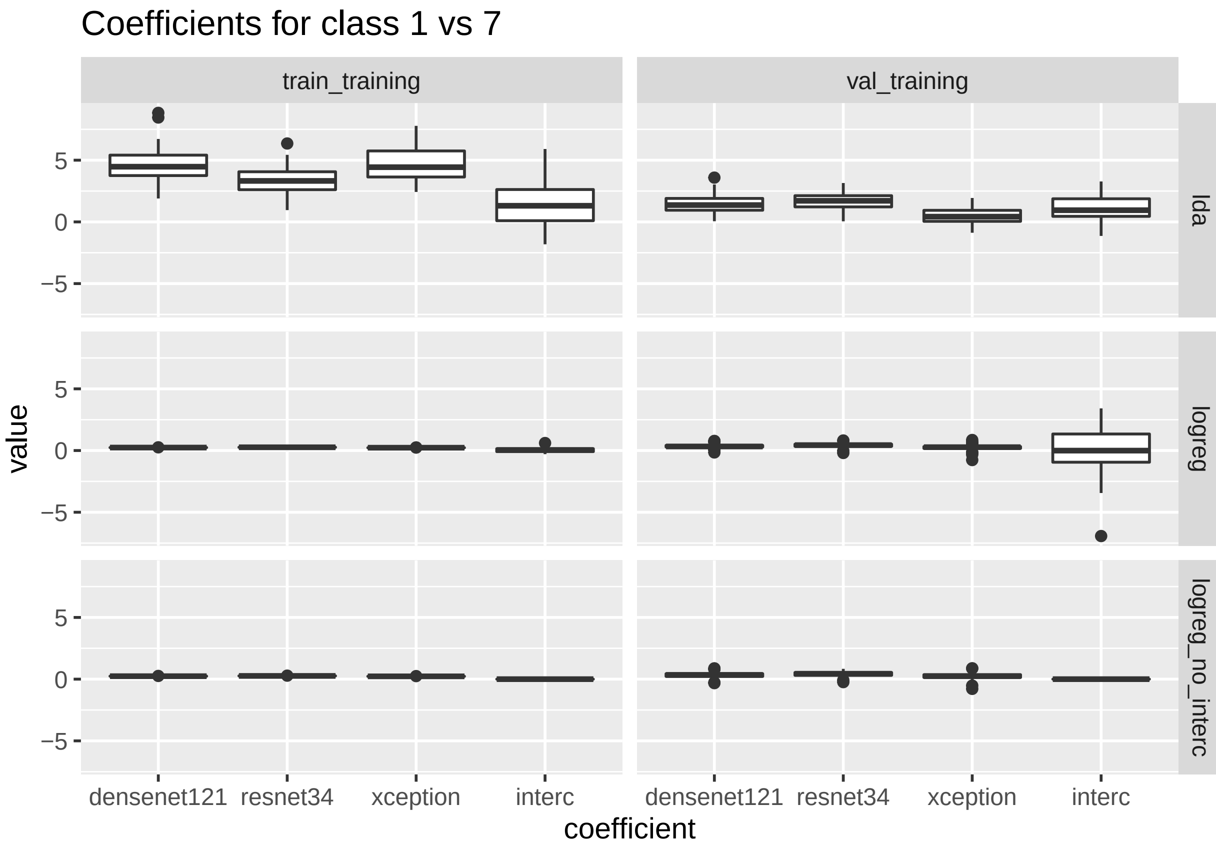
Coefficients for class 1 vs 3
train_training
val_training
lda
logreg
logreg_no_interc
value
6
3
0
−3
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 1 vs 4
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient

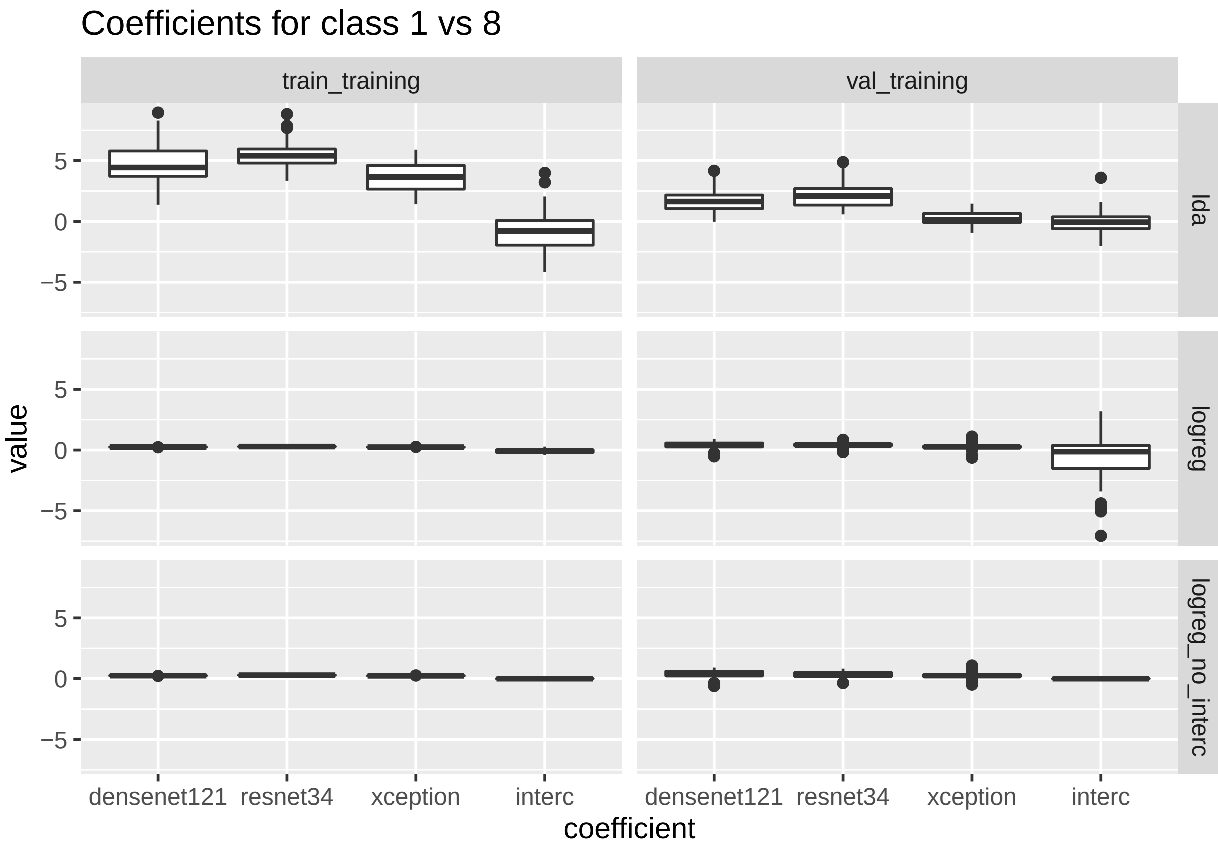

Coefficients for class 1 vs 5
train_training
val_training
lda
logreg
logreg_no_interc
10
5
0
−5
value
densenet121
resnet34
xception
interc
coefficient



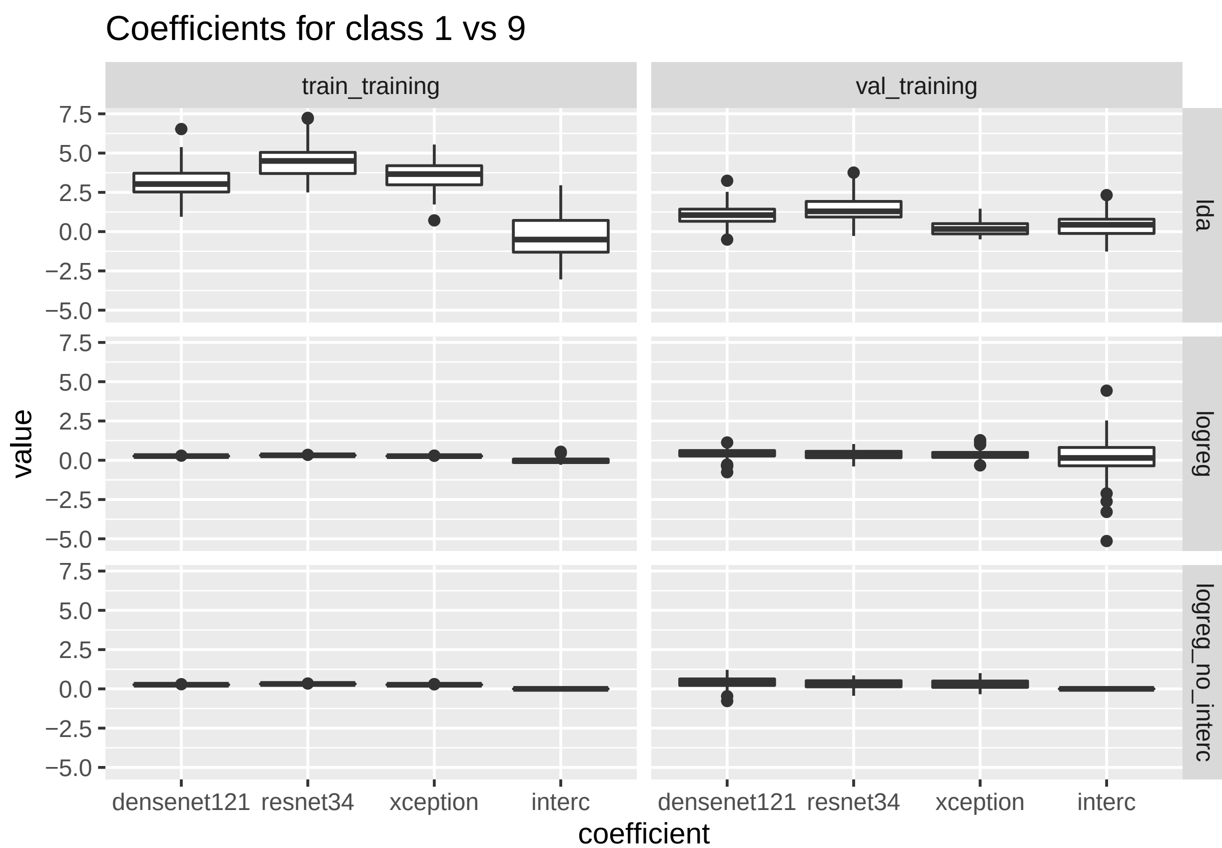
Coefficients for class 1 vs 6
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient



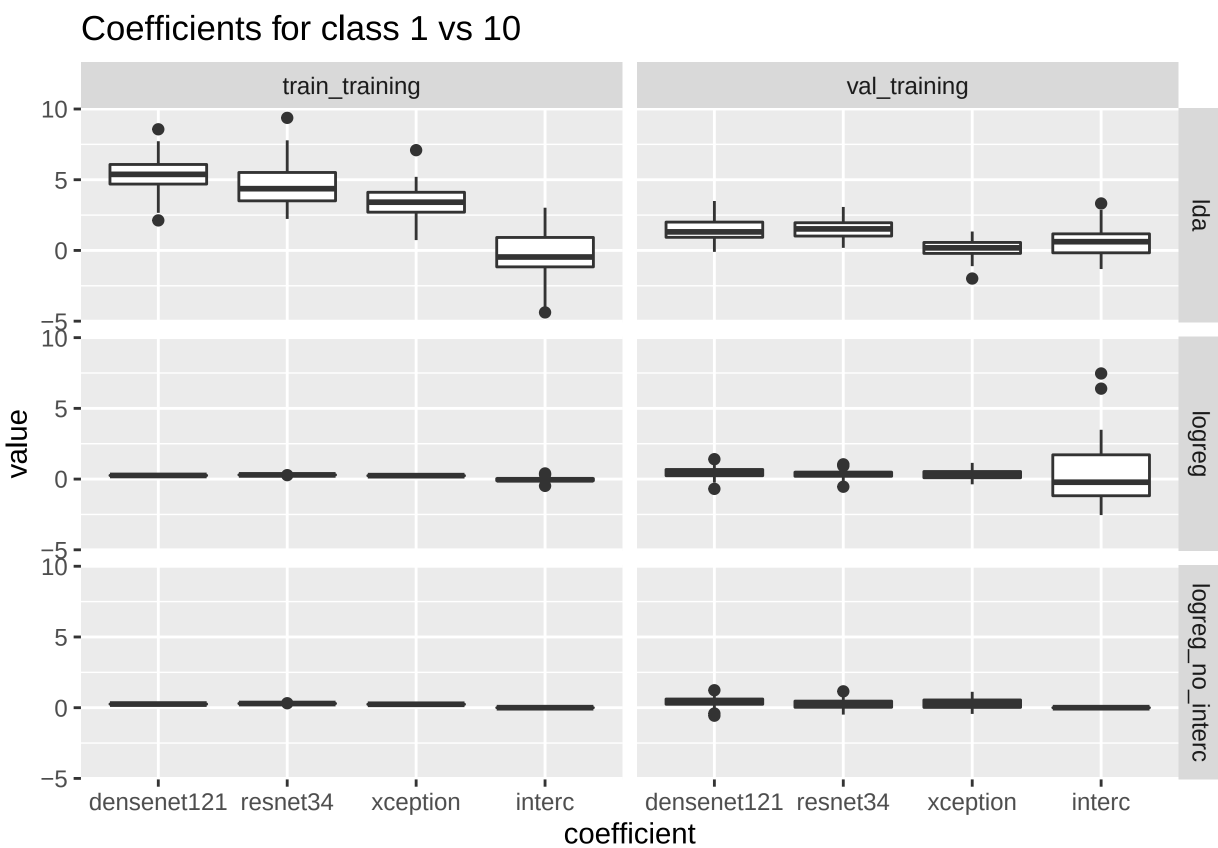
Coefficients for class 1 vs 7
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient



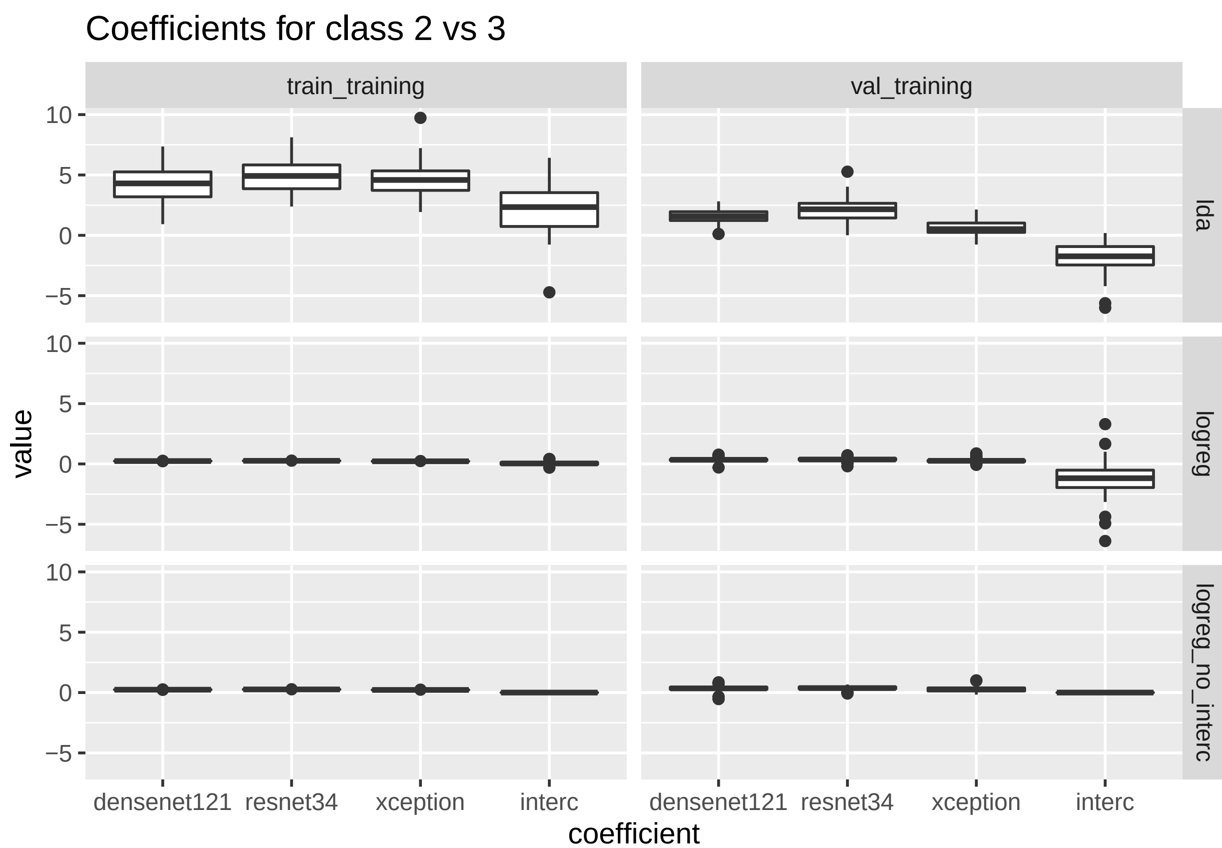
Coefficients for class 1 vs 8
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient

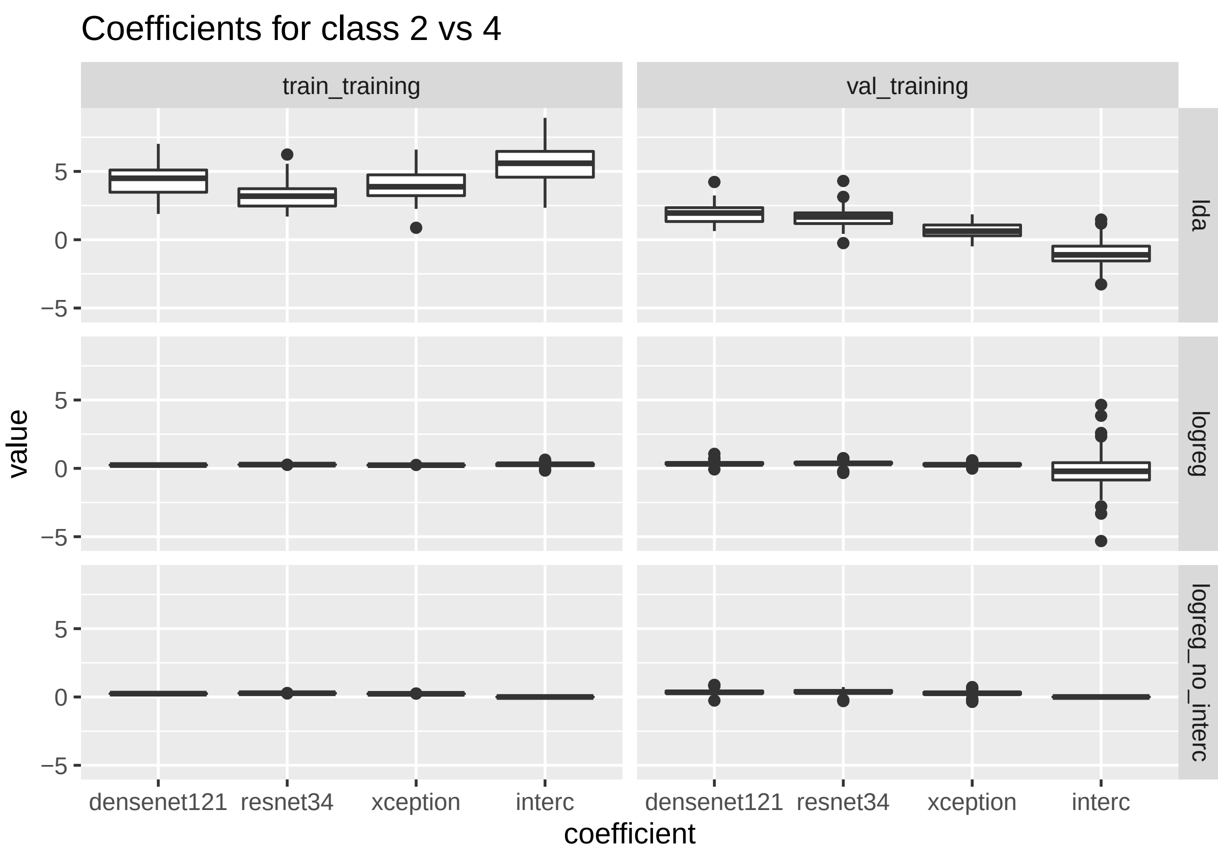


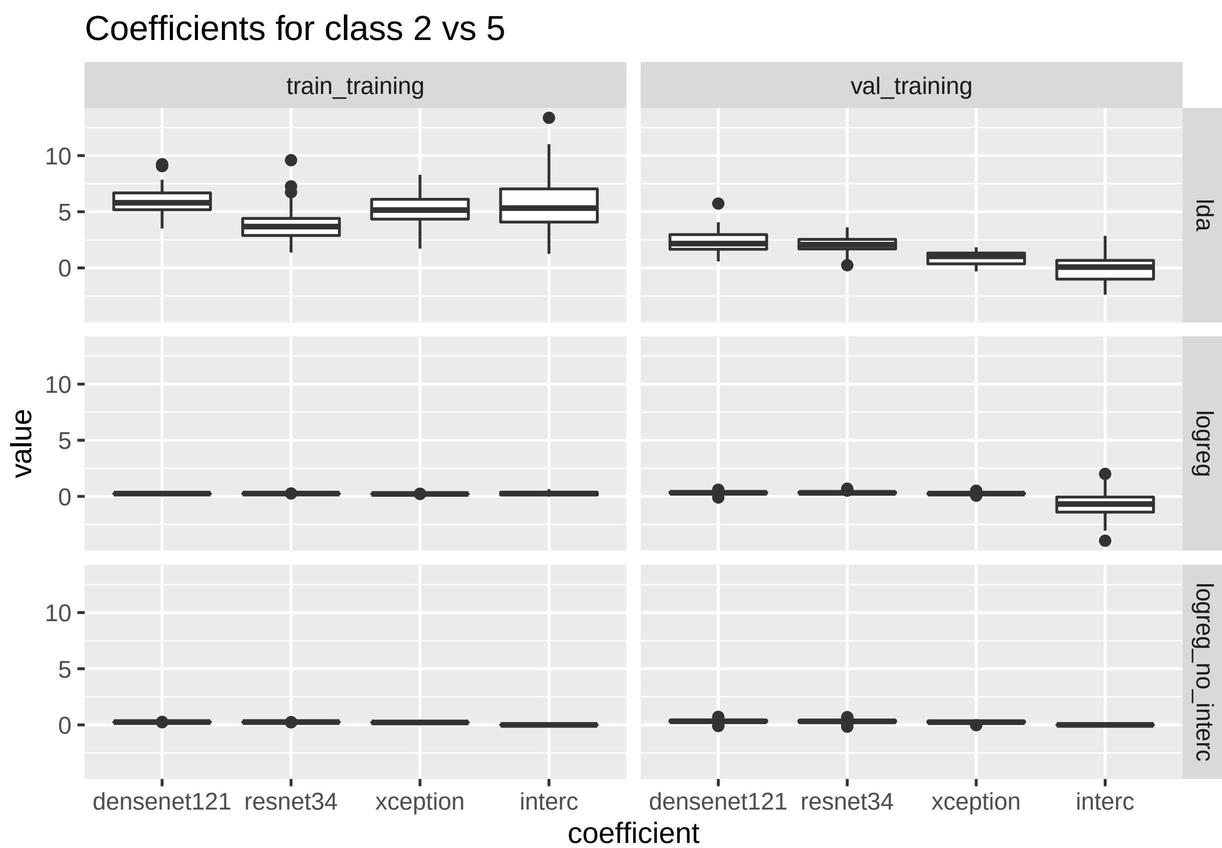
Coefficients for class 1 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
7.5
5.0
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient

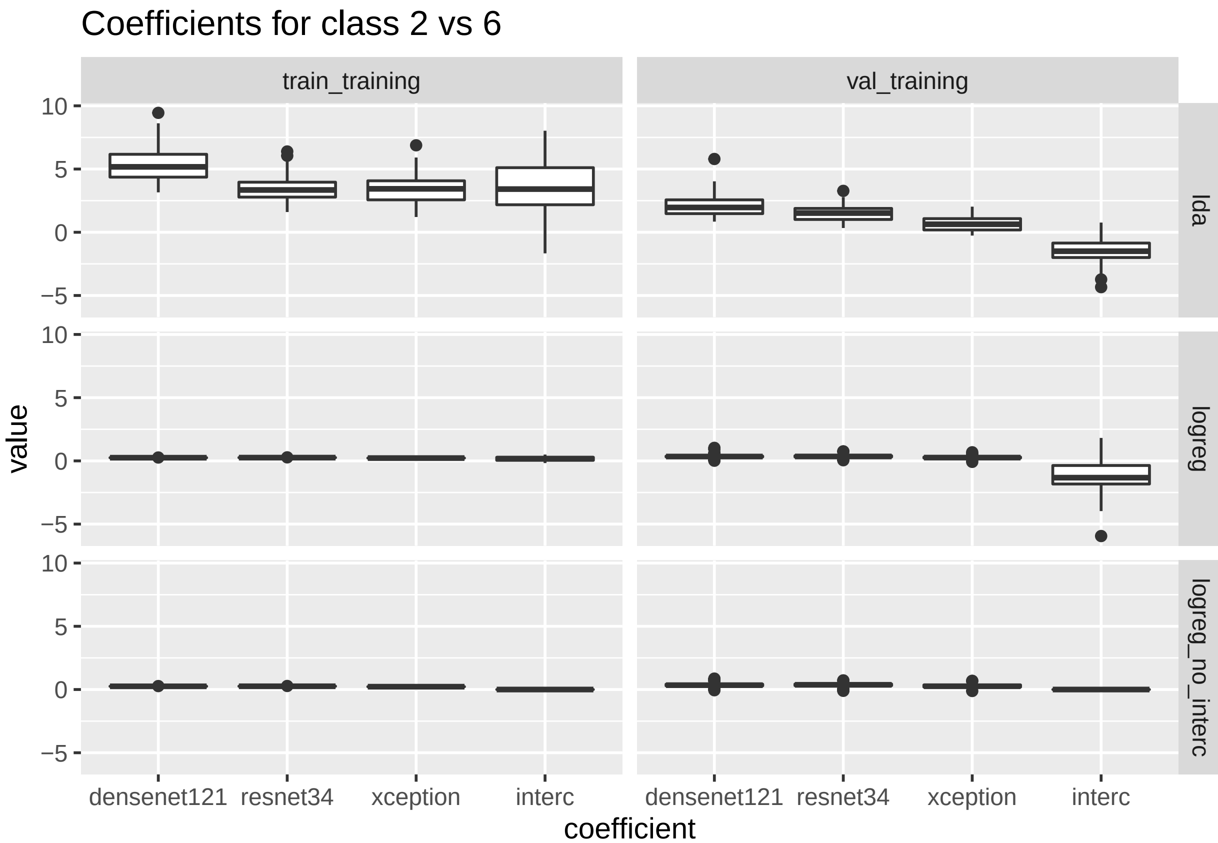


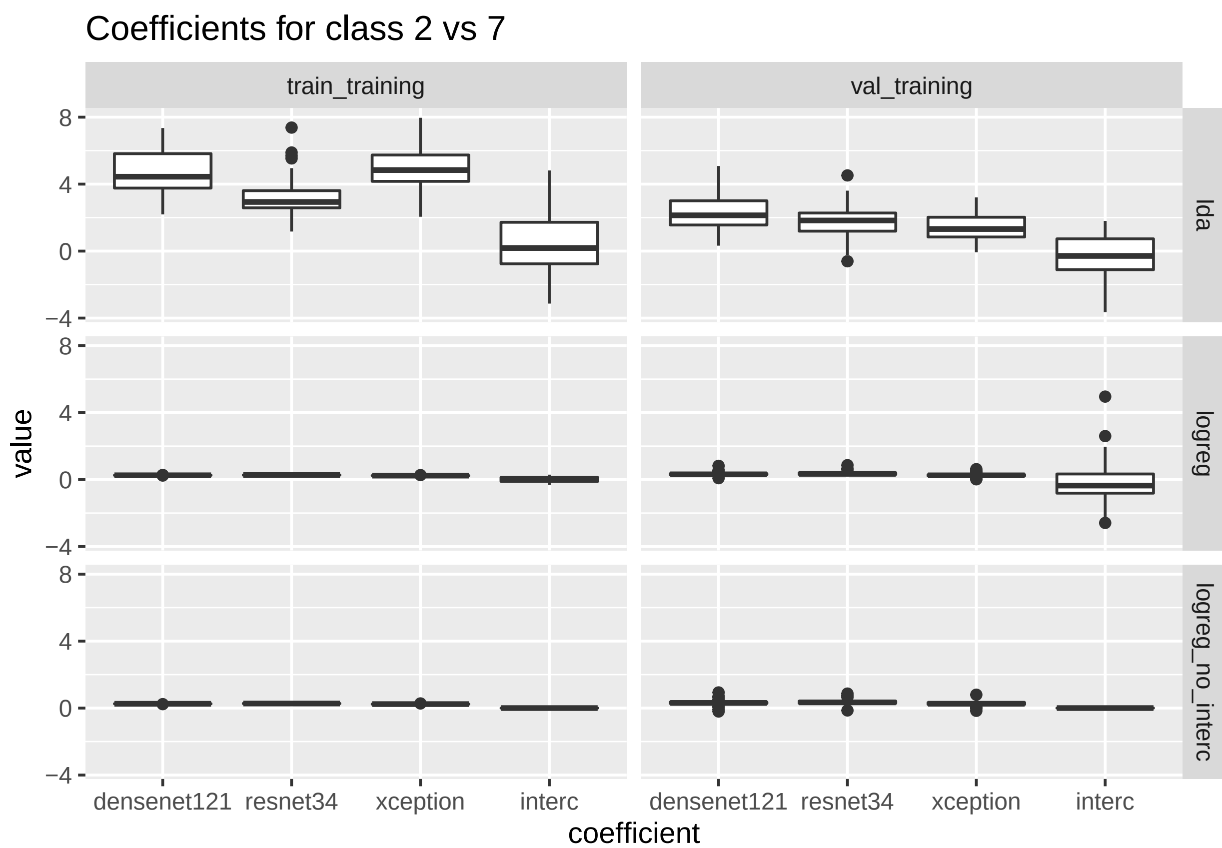
Coefficients for class 1 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 2 vs 3
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient

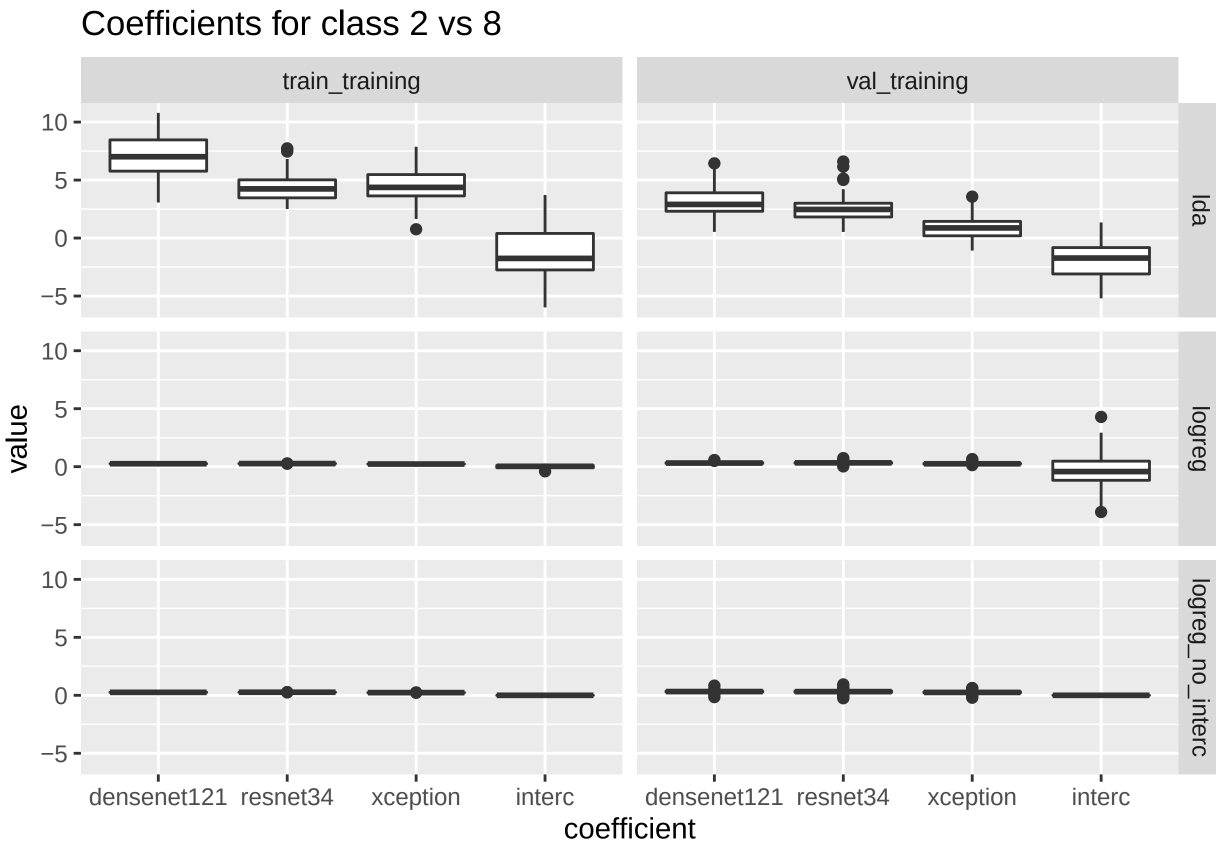

Coefficients for class 2 vs 4
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient

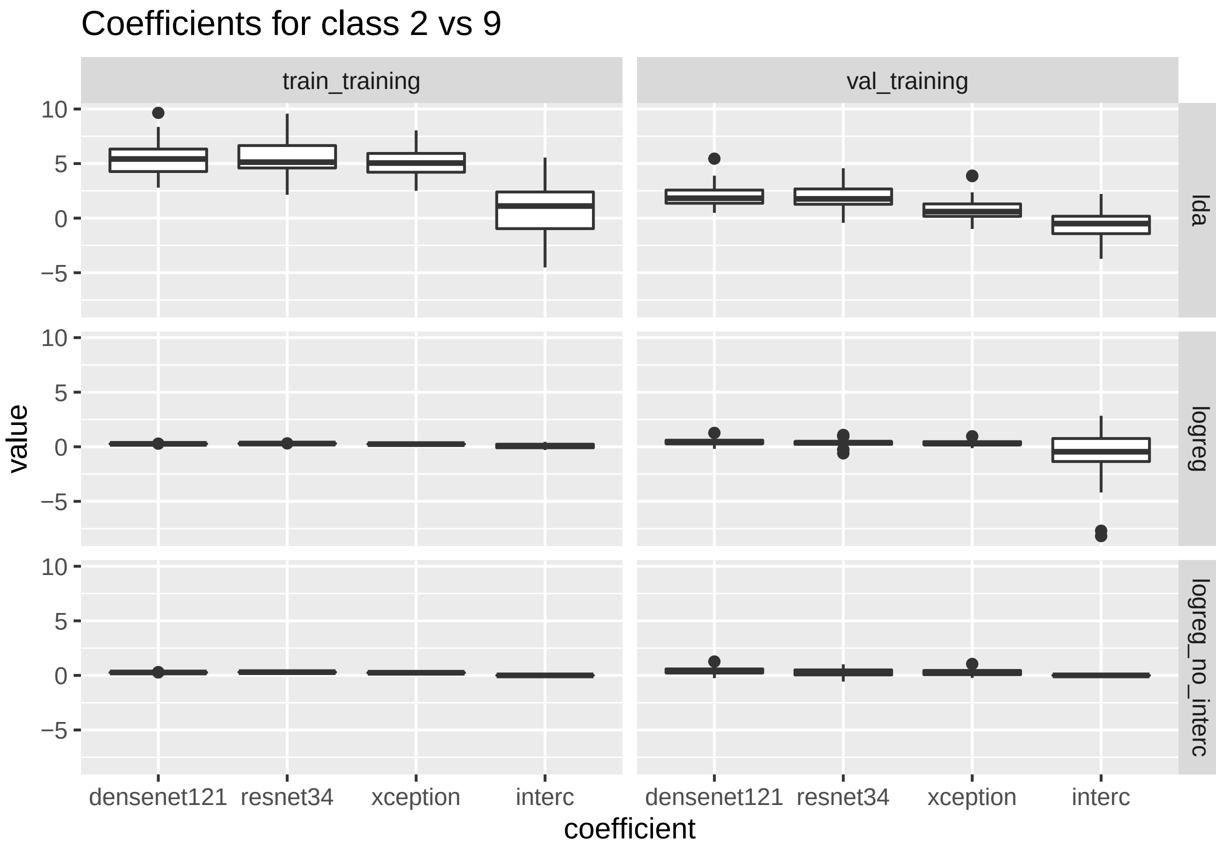

Coefficients for class 2 vs 5
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
densenet121
resnet34
xception
interc
coefficient

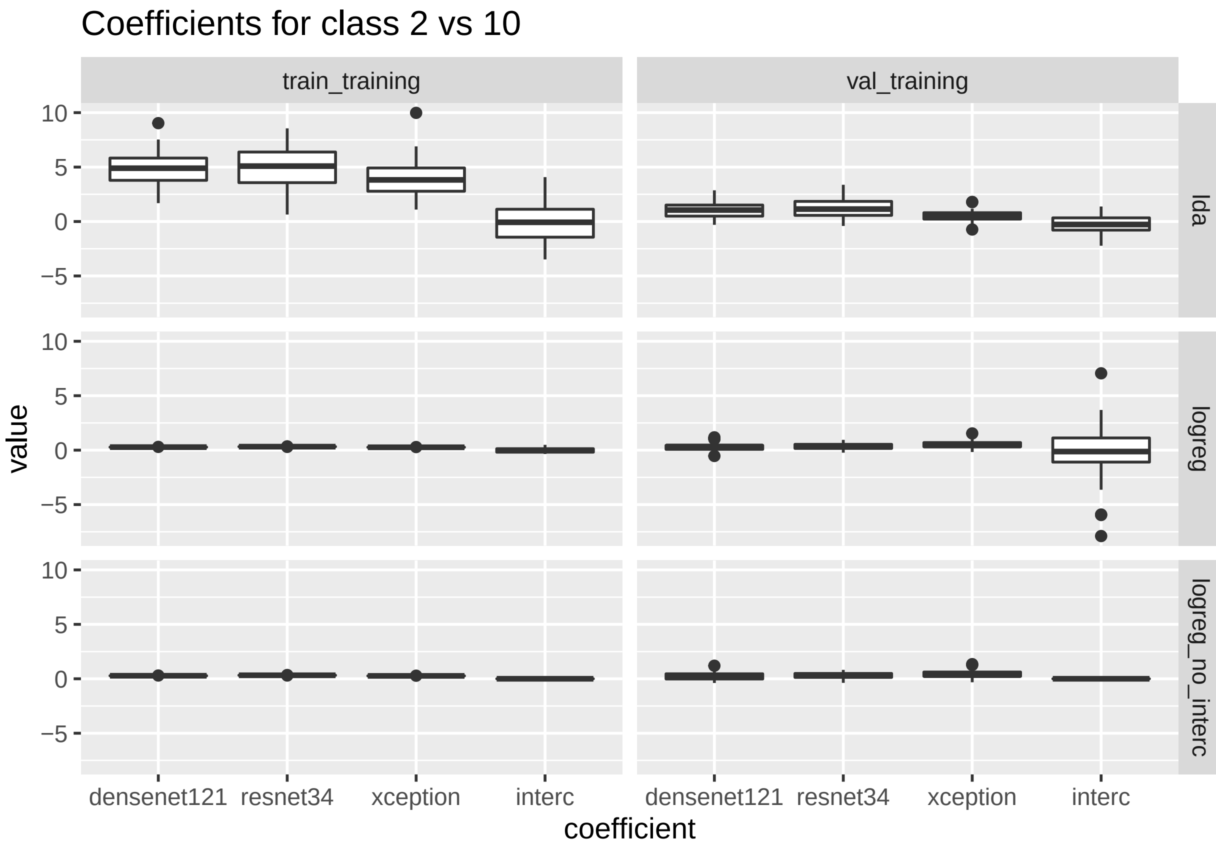

Coefficients for class 2 vs 6
train_training
val_training
lda
logreg
logreg_no_interc
value
coefficient
densenet121
resnet34
xception
interc
10
5
0
−5



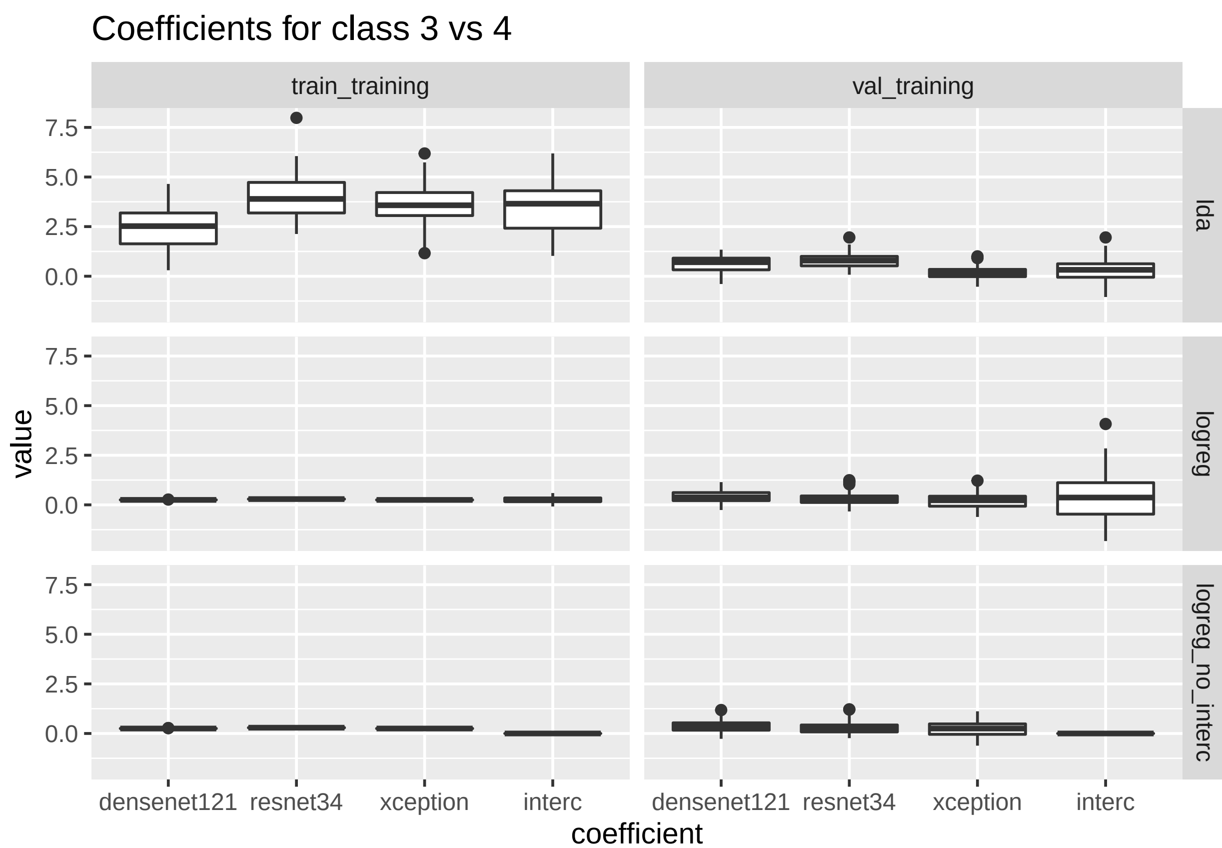
Coefficients for class 2 vs 7
train_training
val_training
lda
logreg
logreg_no_interc
value
8
4
0
−4
densenet121
resnet34
xception
interc
coefficient

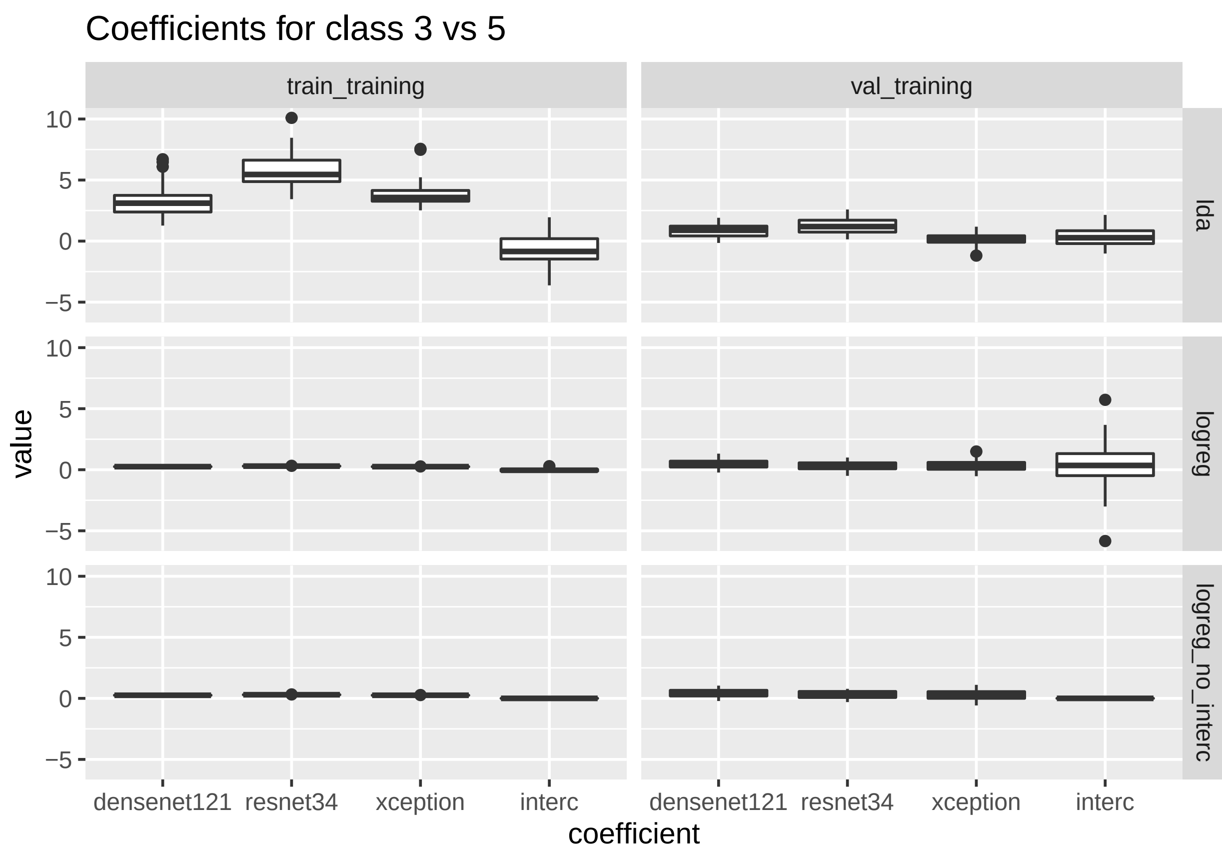

Coefficients for class 2 vs 8
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 2 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient

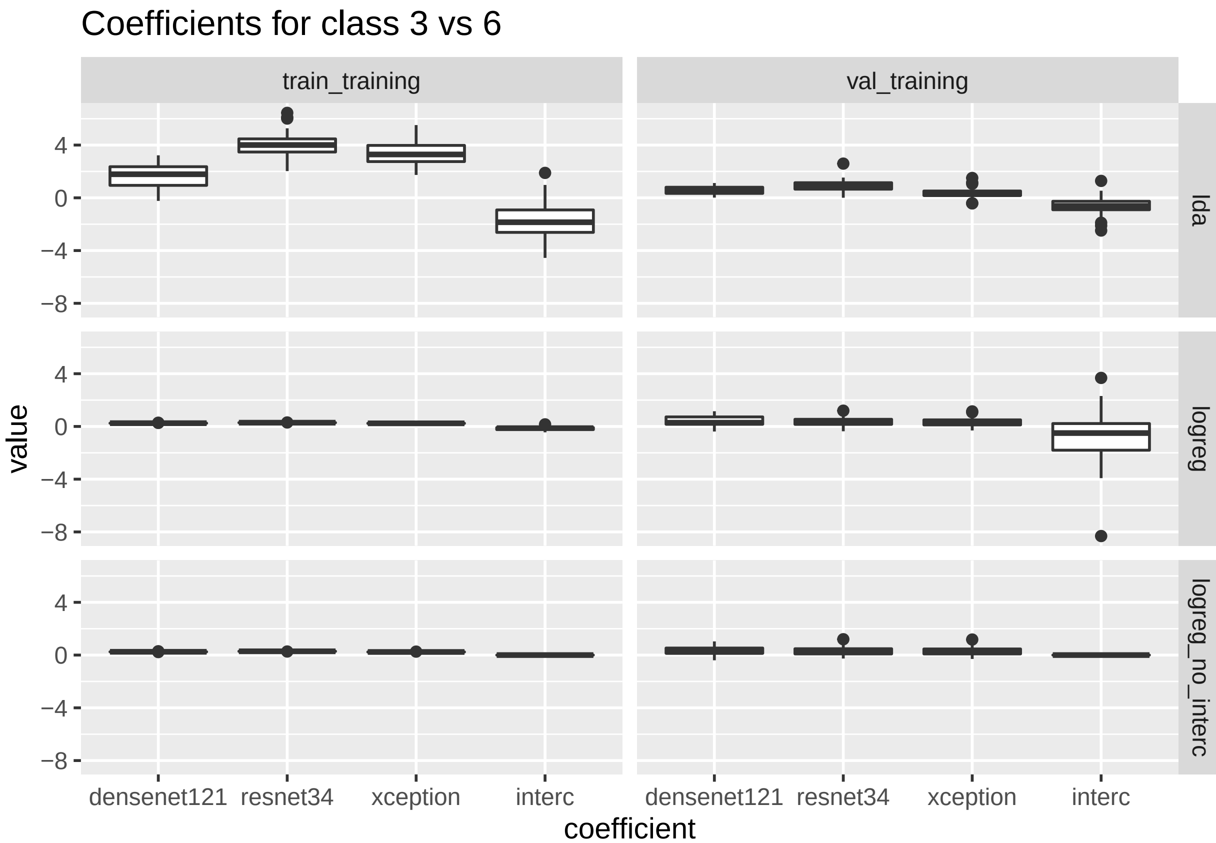


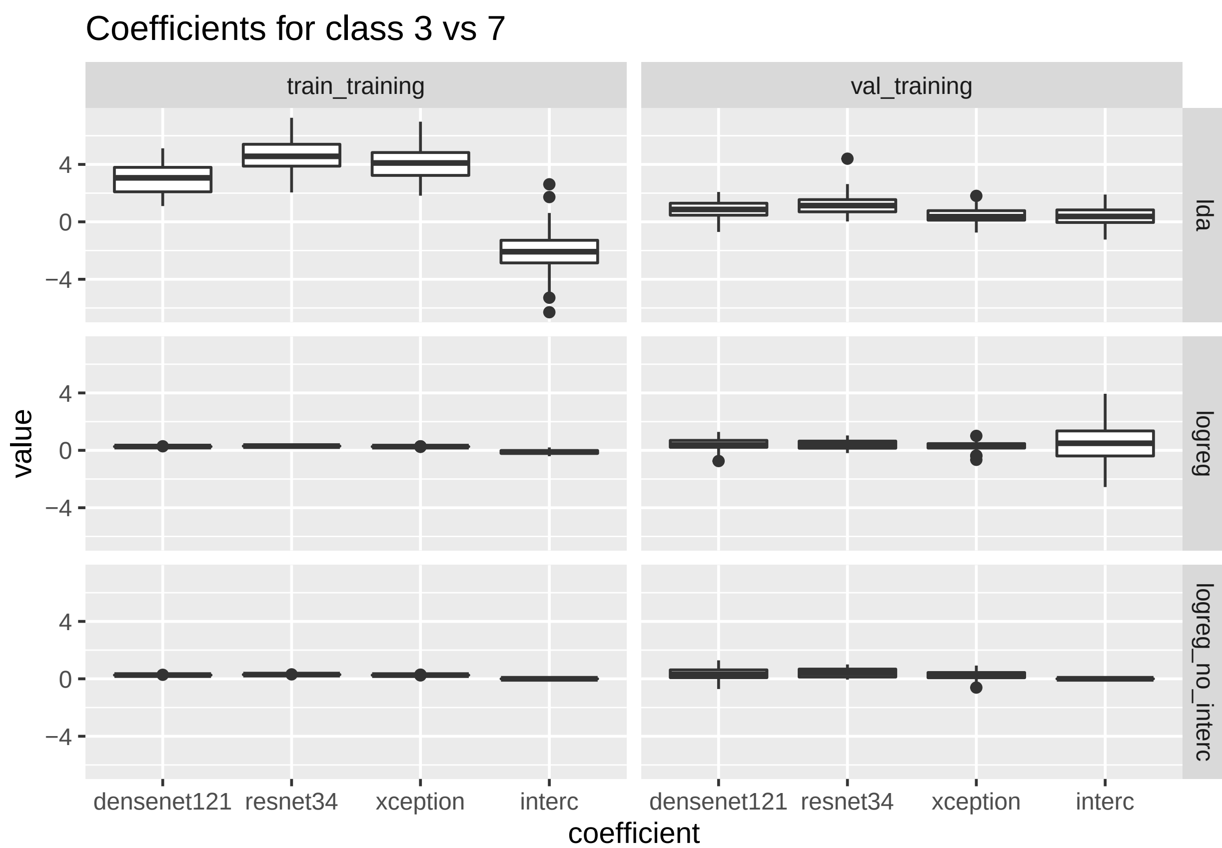
Coefficients for class 2 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 3 vs 4
train_training
val_training
lda
logreg
logreg_no_interc
value
7.5
5.0
2.5
0.0
densenet121
resnet34
xception
interc
coefficient

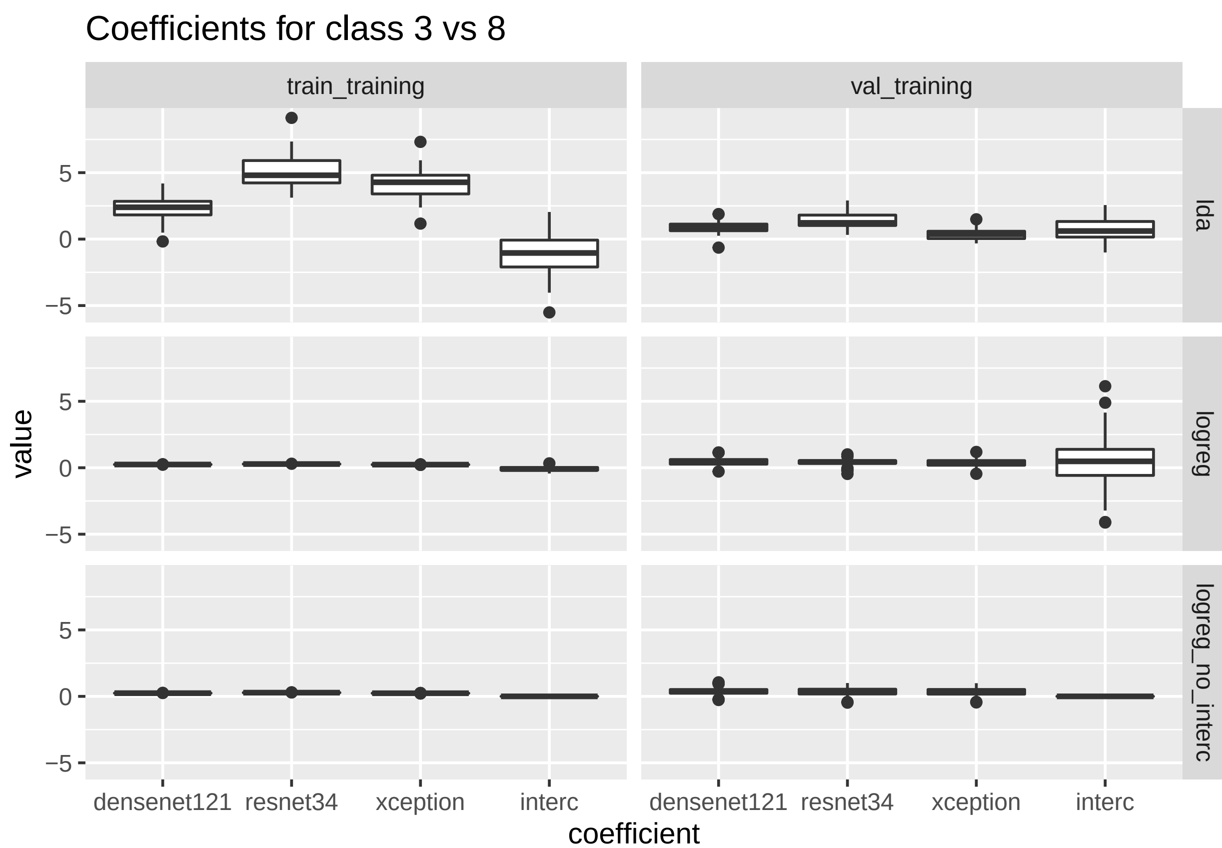


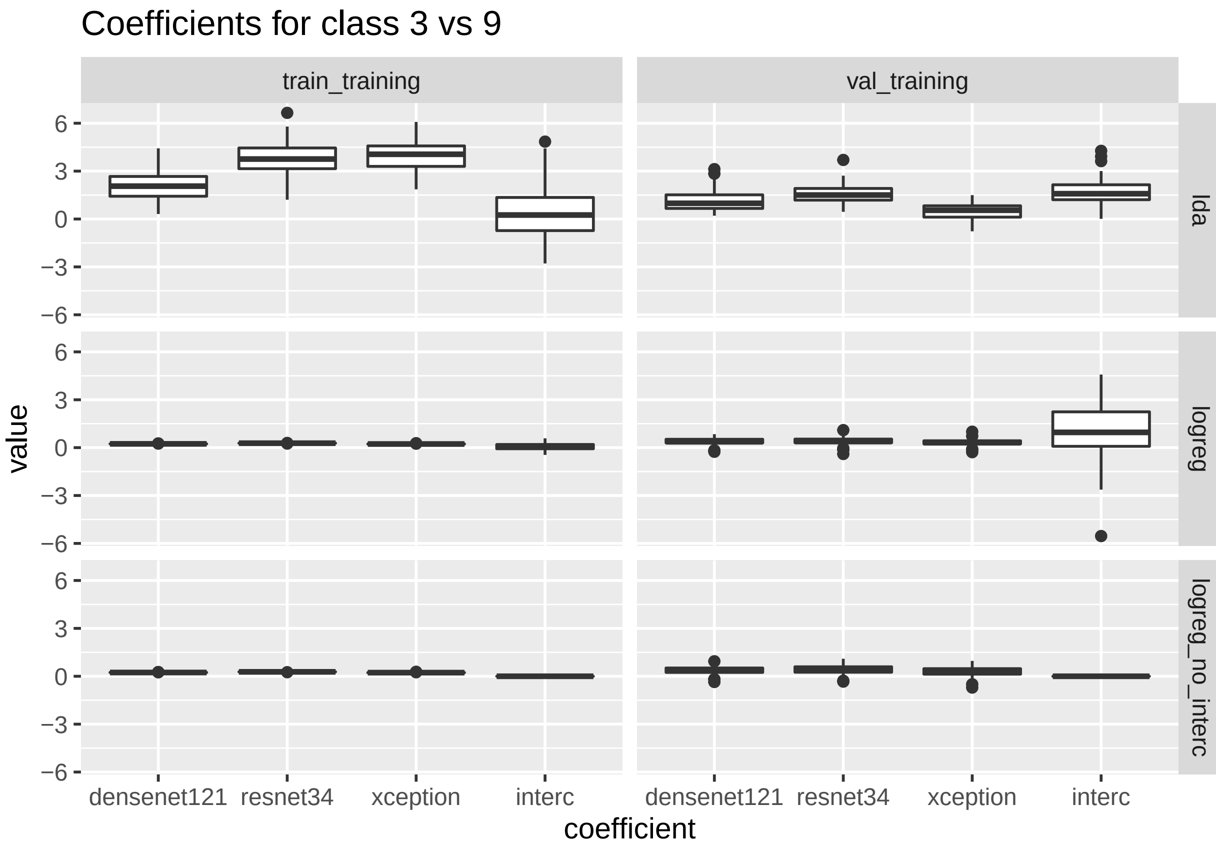
Coefficients for class 3 vs 5
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 3 vs 6
train_training
val_training
lda
logreg
logreg_no_interc
value
4
0
−4
−8
densenet121
resnet34
xception
interc
coefficient

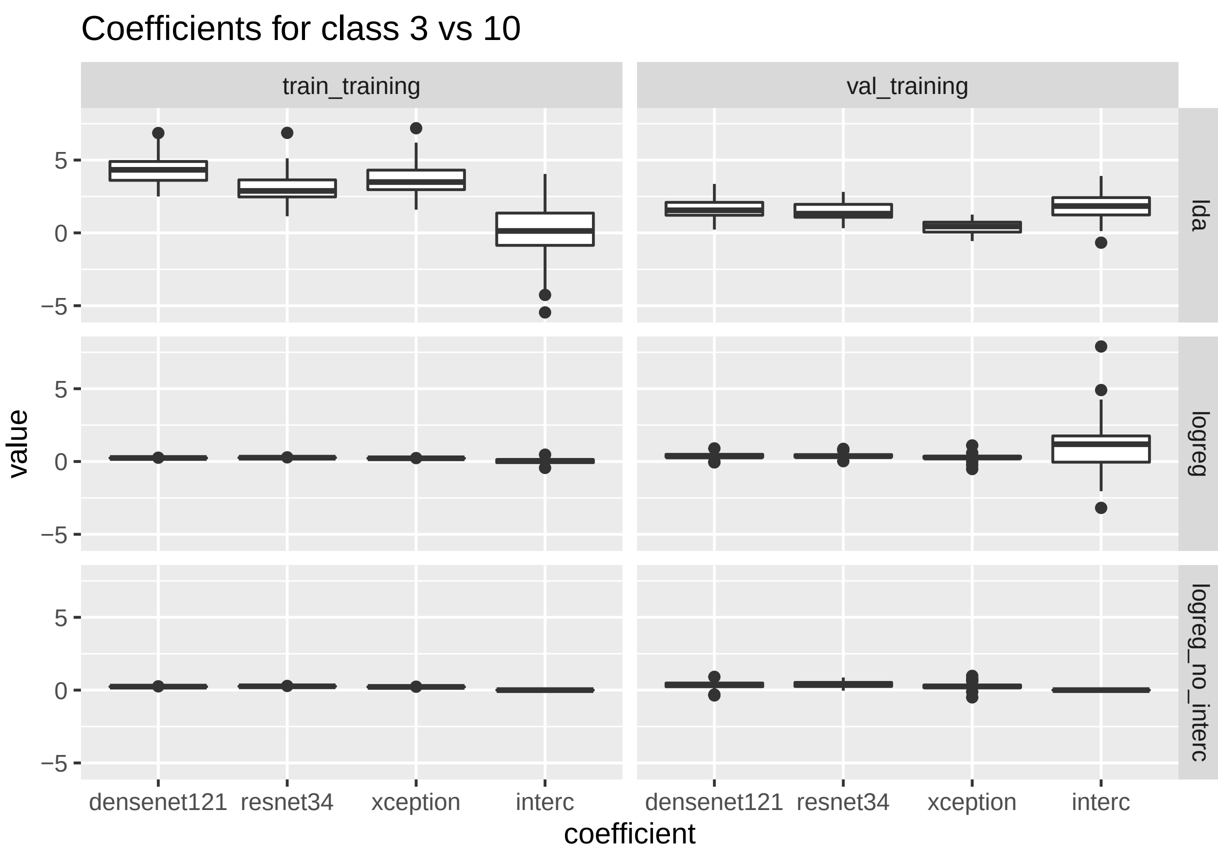


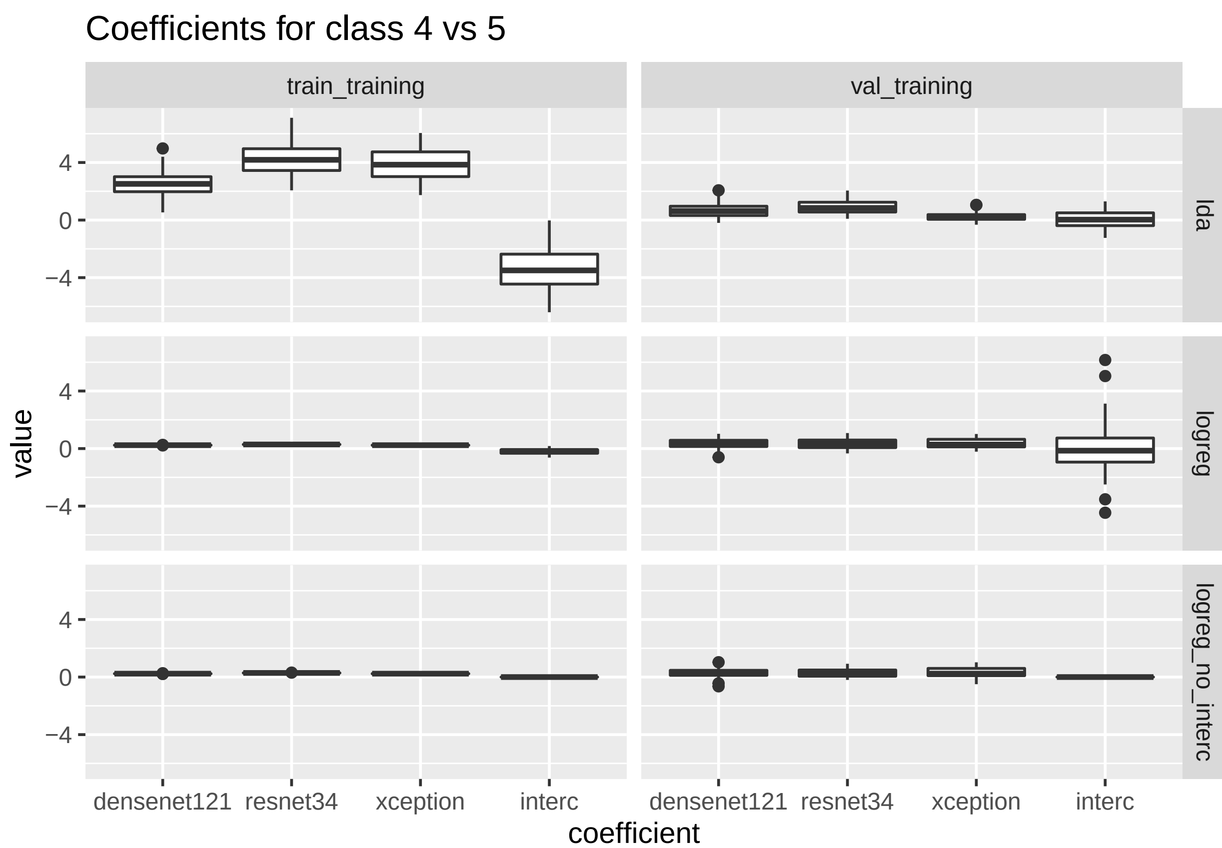
Coefficients for class 3 vs 7
train_training
val_training
lda
logreg
logreg_no_interc
value
4
0
−4
densenet121
resnet34
xception
interc
coefficient

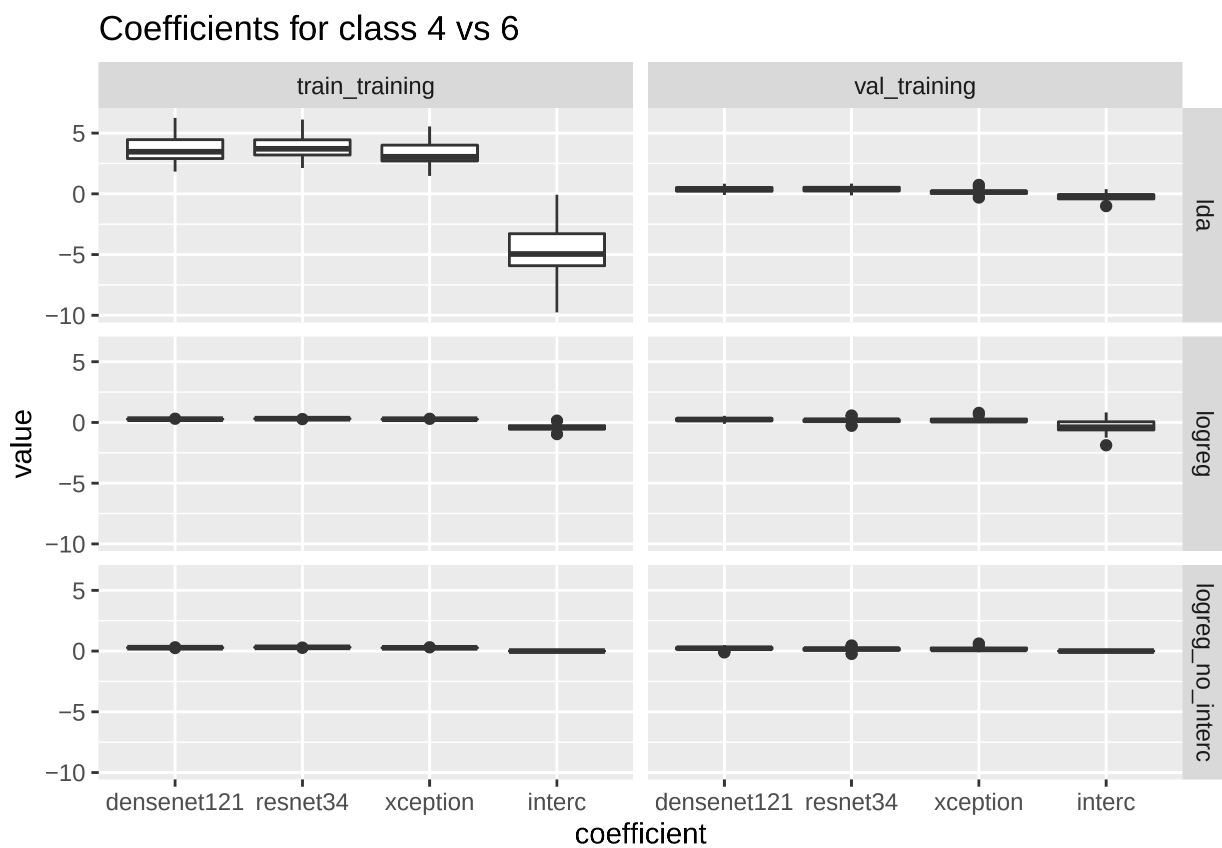

Coefficients for class 3 vs 8
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient



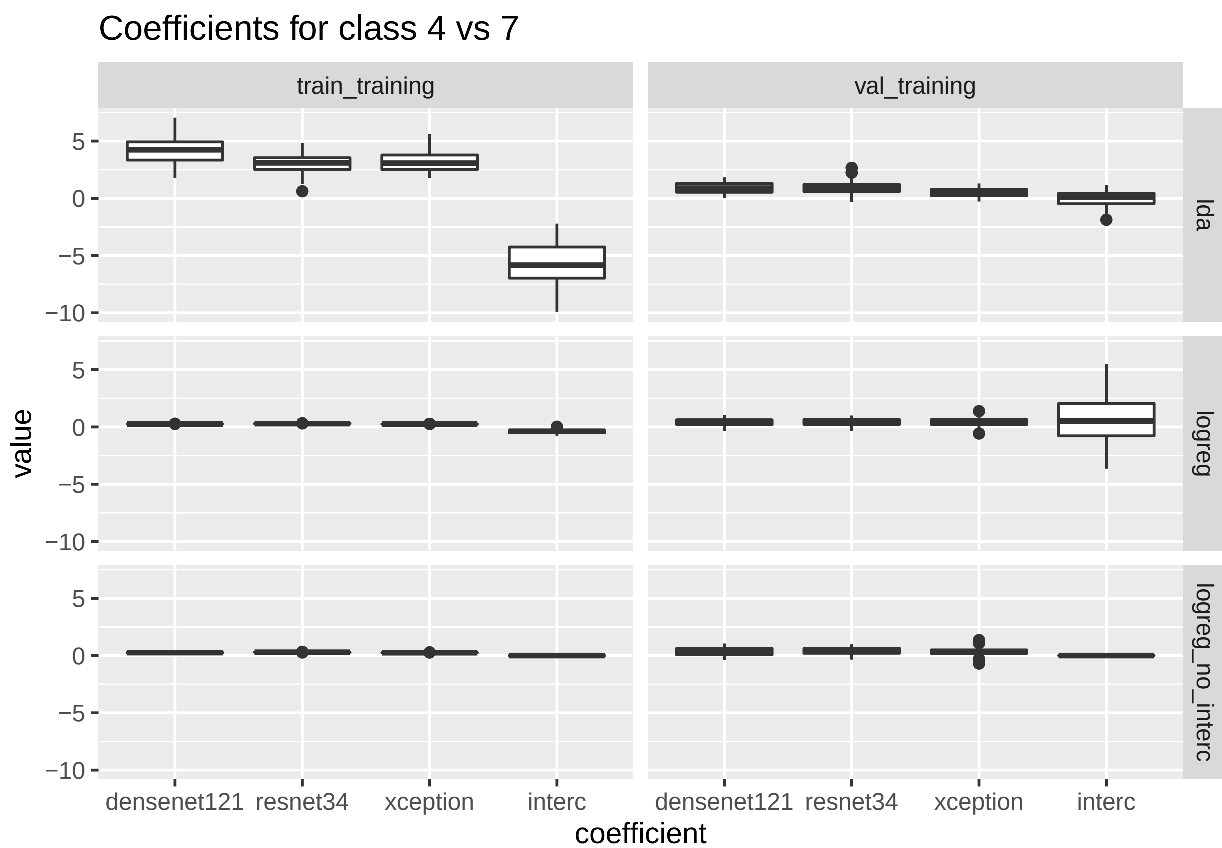
Coefficients for class 3 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
6
3
0
−3
−6
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 3 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient

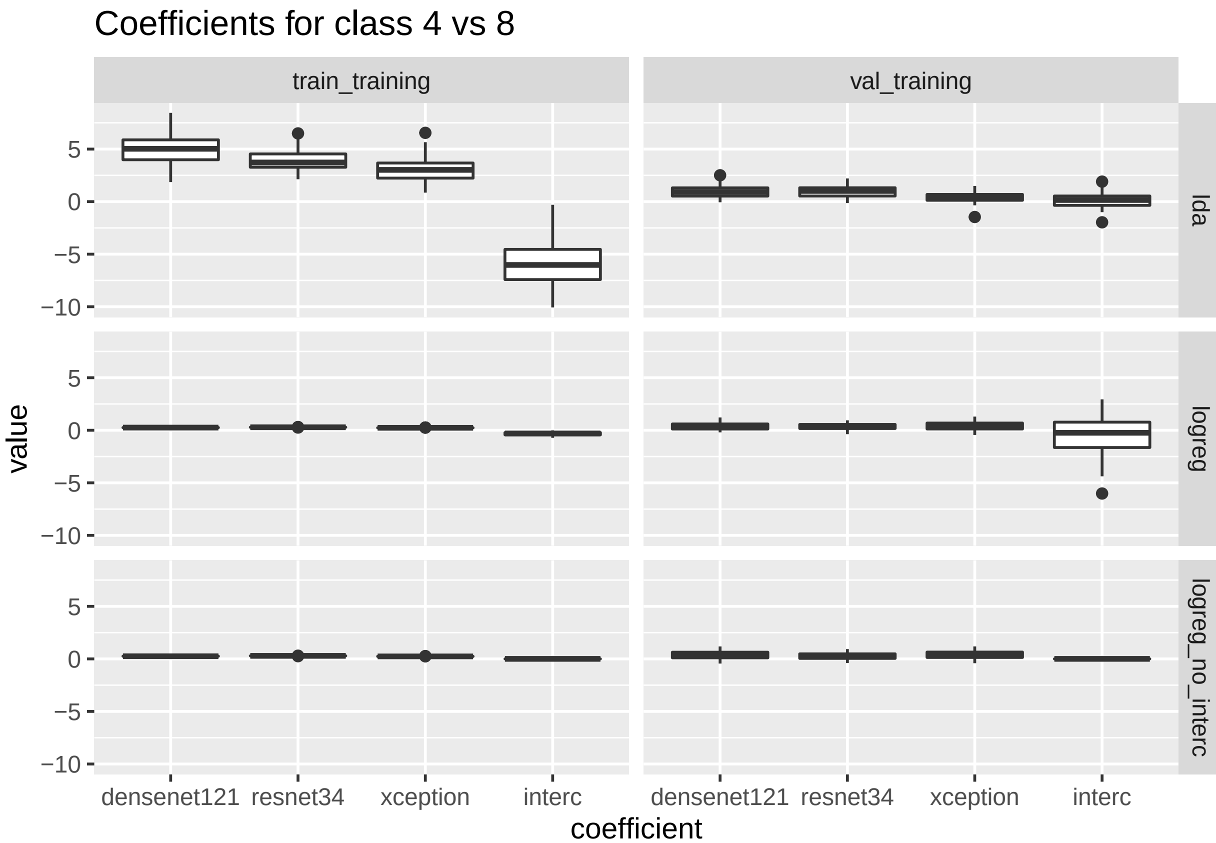

Coefficients for class 4 vs 5
train_training
val_training
lda
logreg
logreg_no_interc
value
4
0
−4
densenet121
resnet34
xception
interc
coefficient

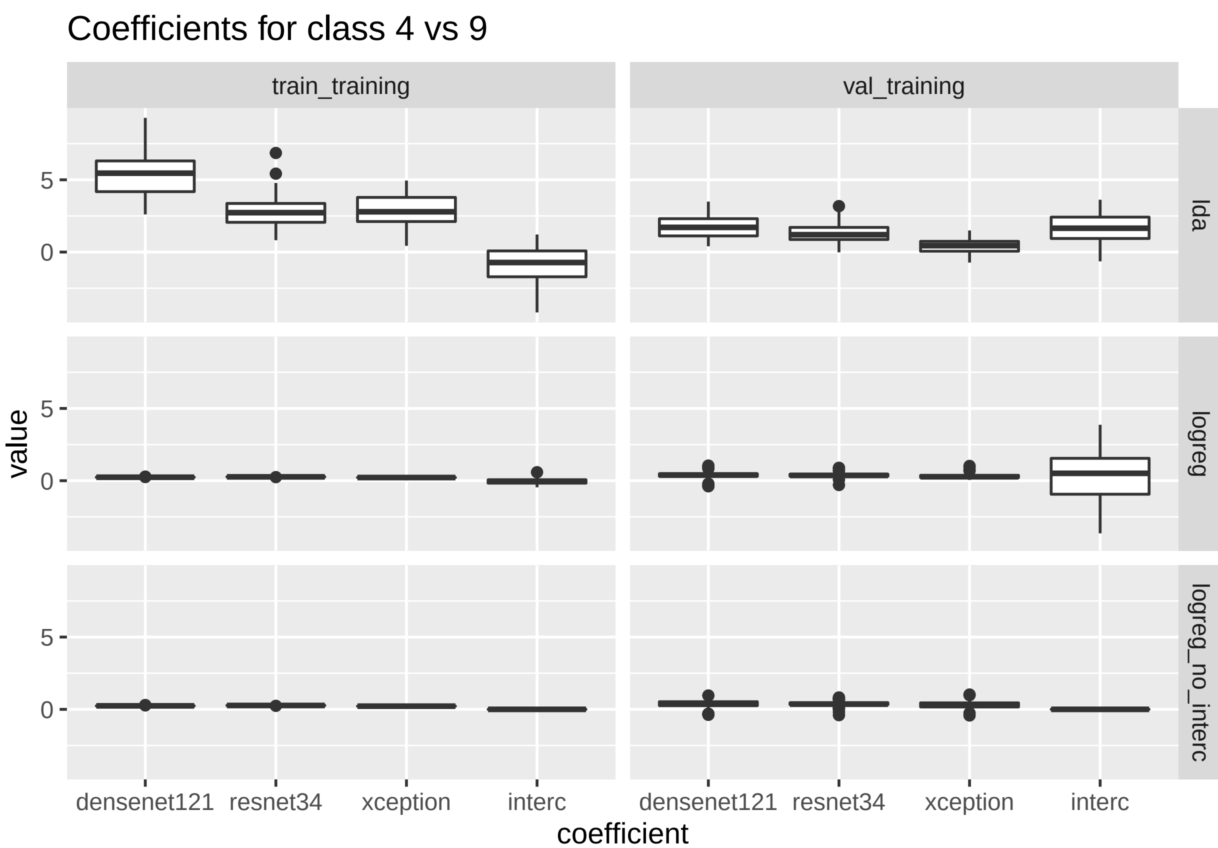


Coefficients for class 4 vs 6
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
−10
densenet121
resnet34
xception
interc
coefficient

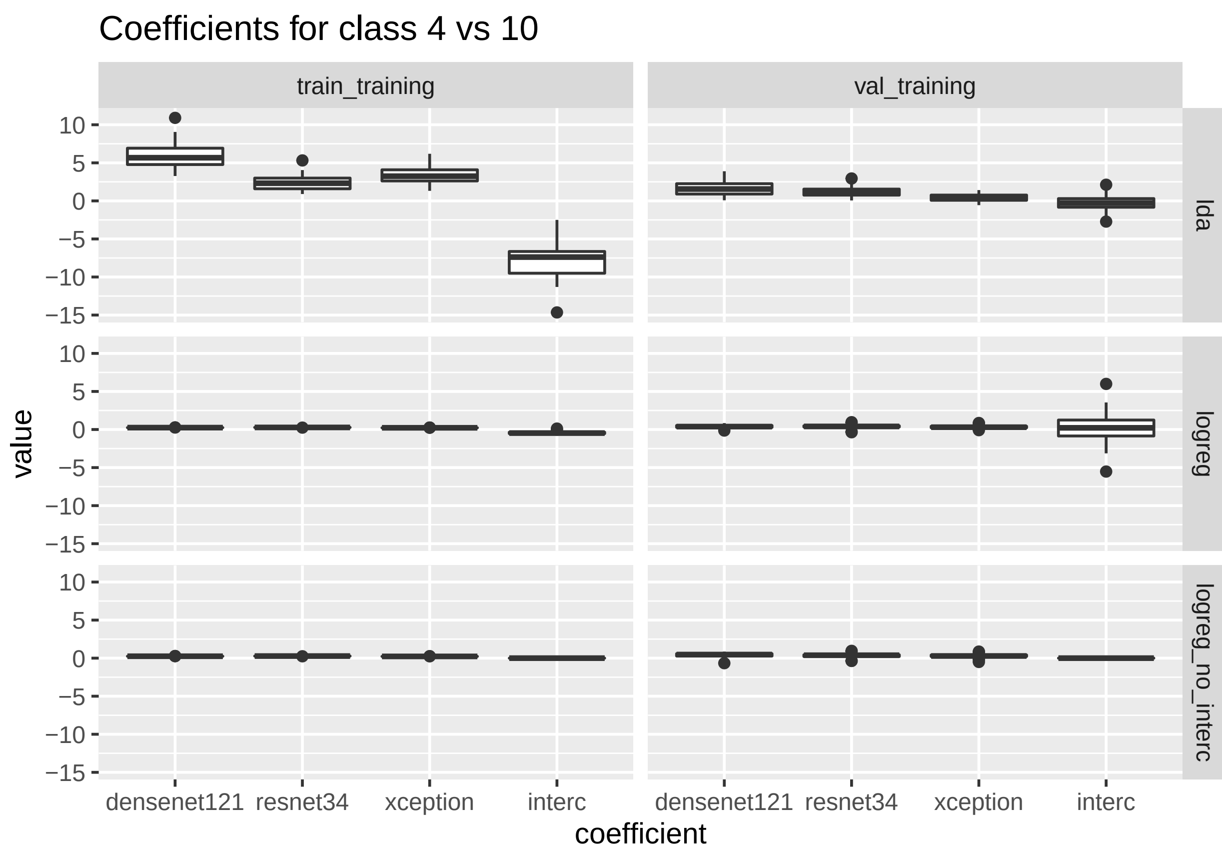


Coefficients for class 4 vs 7
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
−10
densenet121
resnet34
xception
interc
coefficient

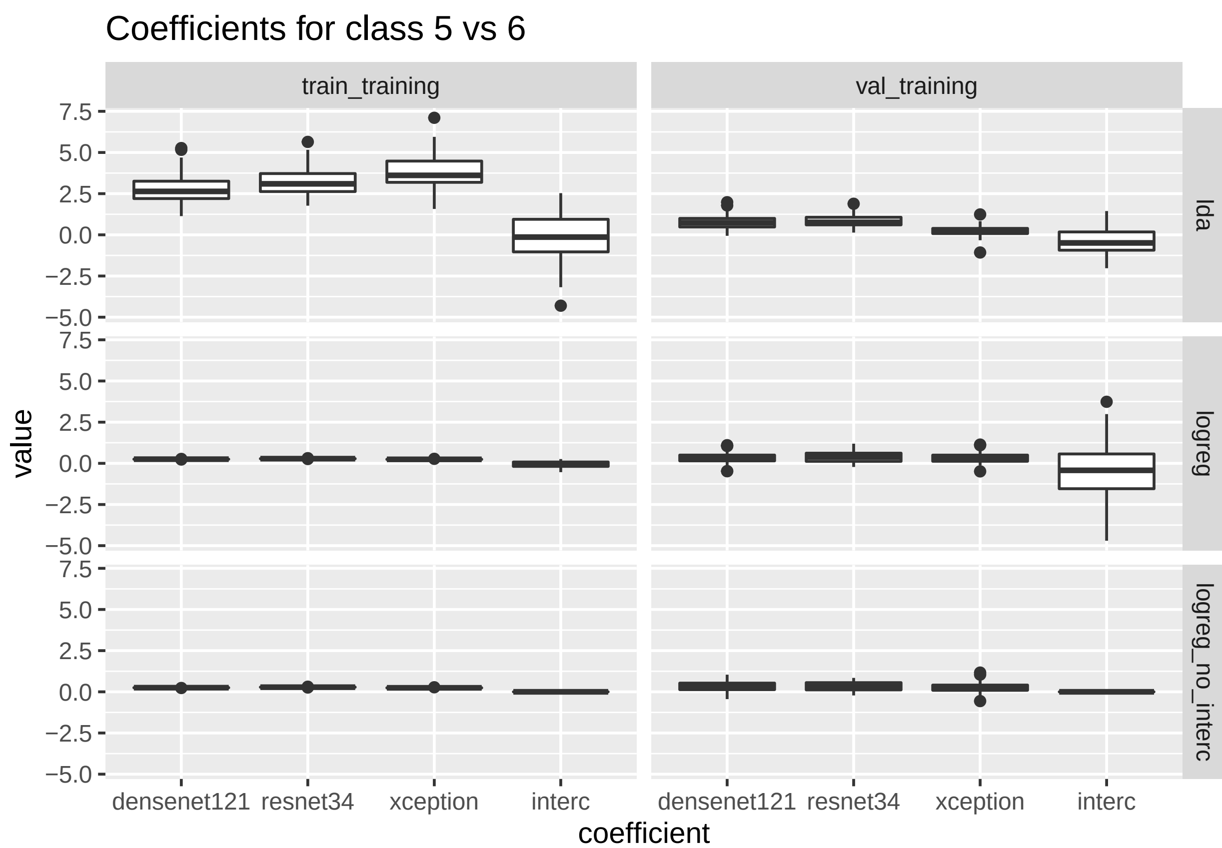

Coefficients for class 4 vs 8
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
−10
densenet121
resnet34
xception
interc
coefficient

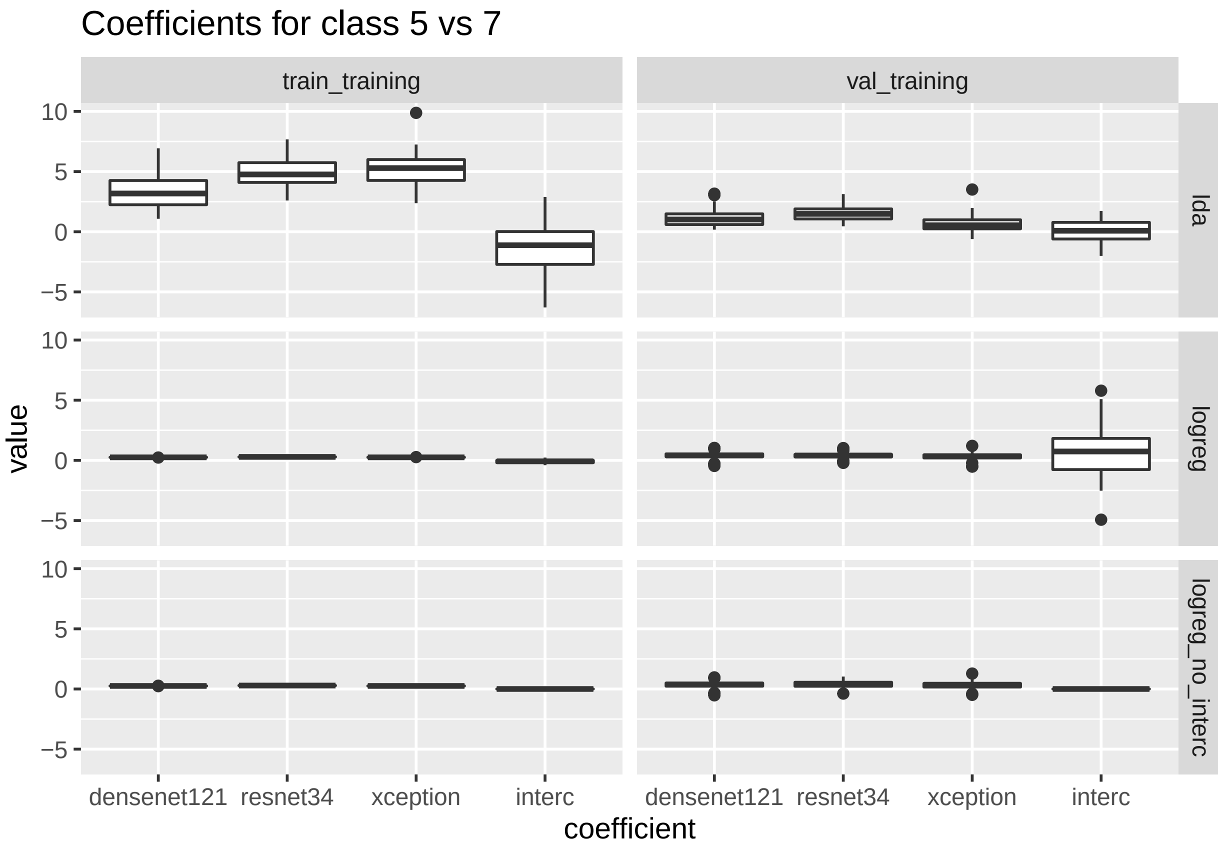


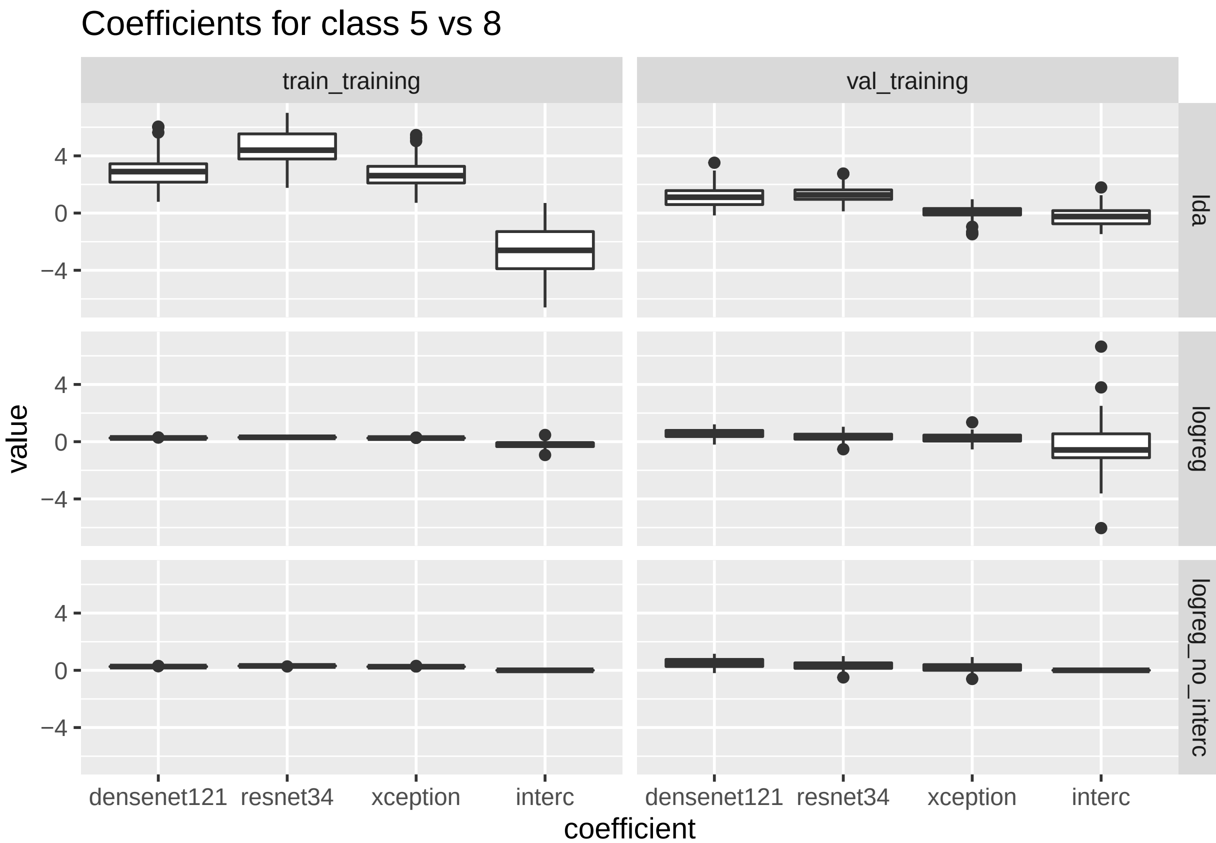
Coefficients for class 4 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
densenet121
resnet34
xception
interc
coefficient

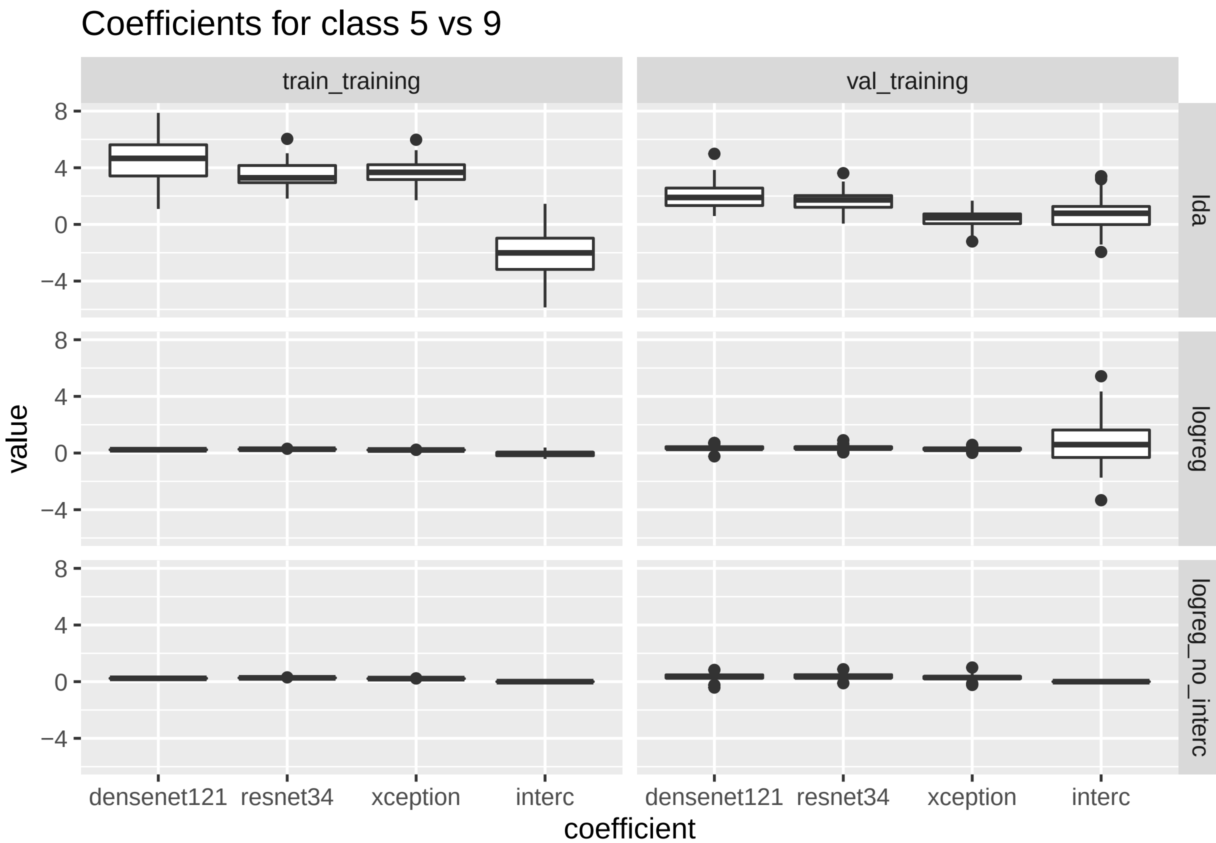


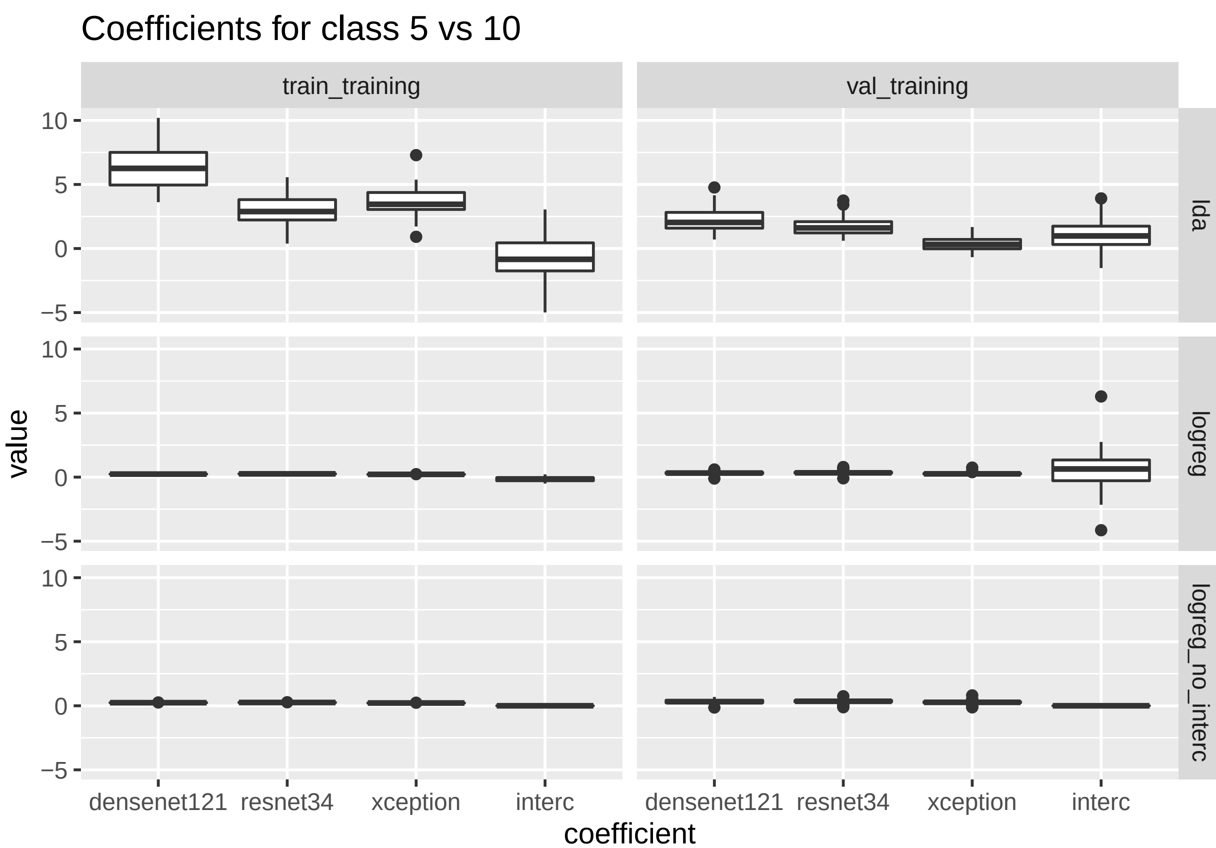
Coefficients for class 4 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
−10
−15
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 5 vs 6
train_training
val_training
lda
logreg
logreg_no_interc
value
7.5
5.0
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient

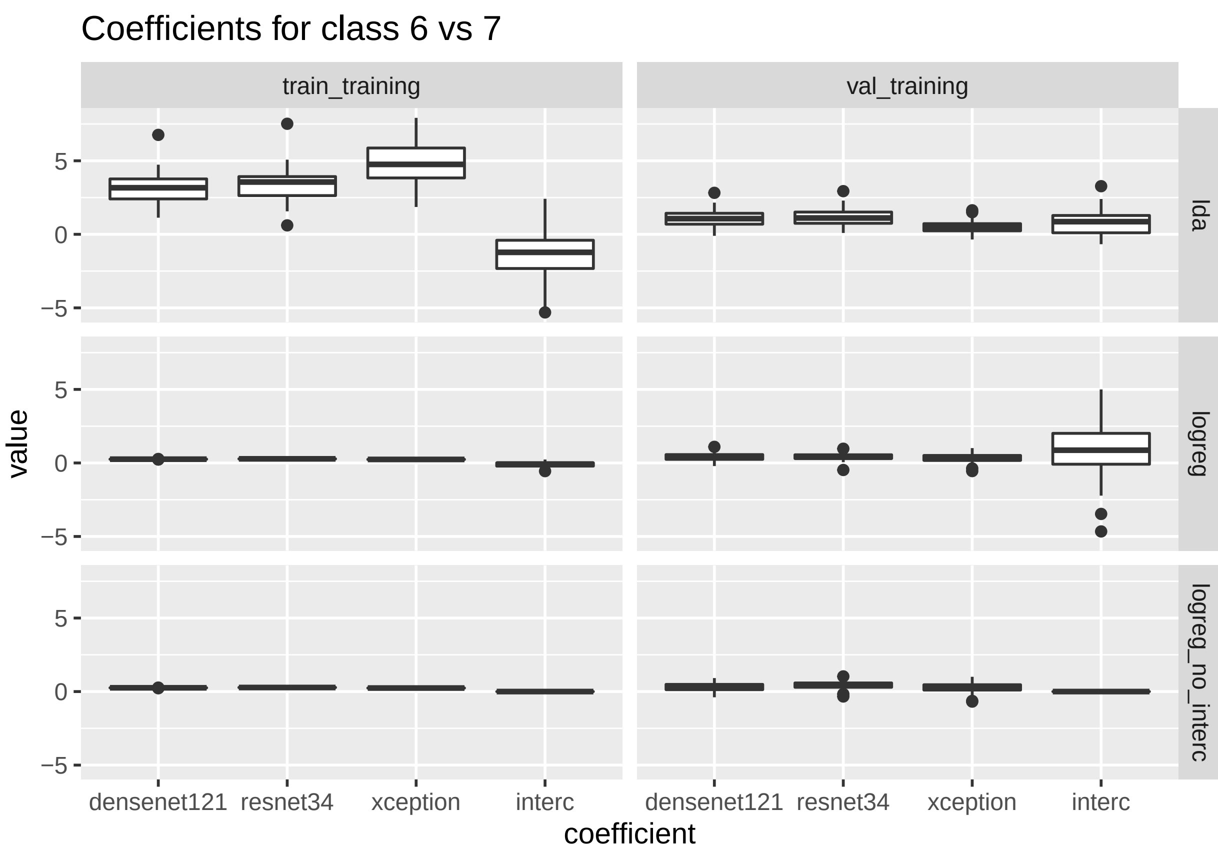

Coefficients for class 5 vs 7
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 5 vs 8
train_training
val_training
lda
logreg
logreg_no_interc
value
4
0
−4
densenet121
resnet34
xception
interc
coefficient

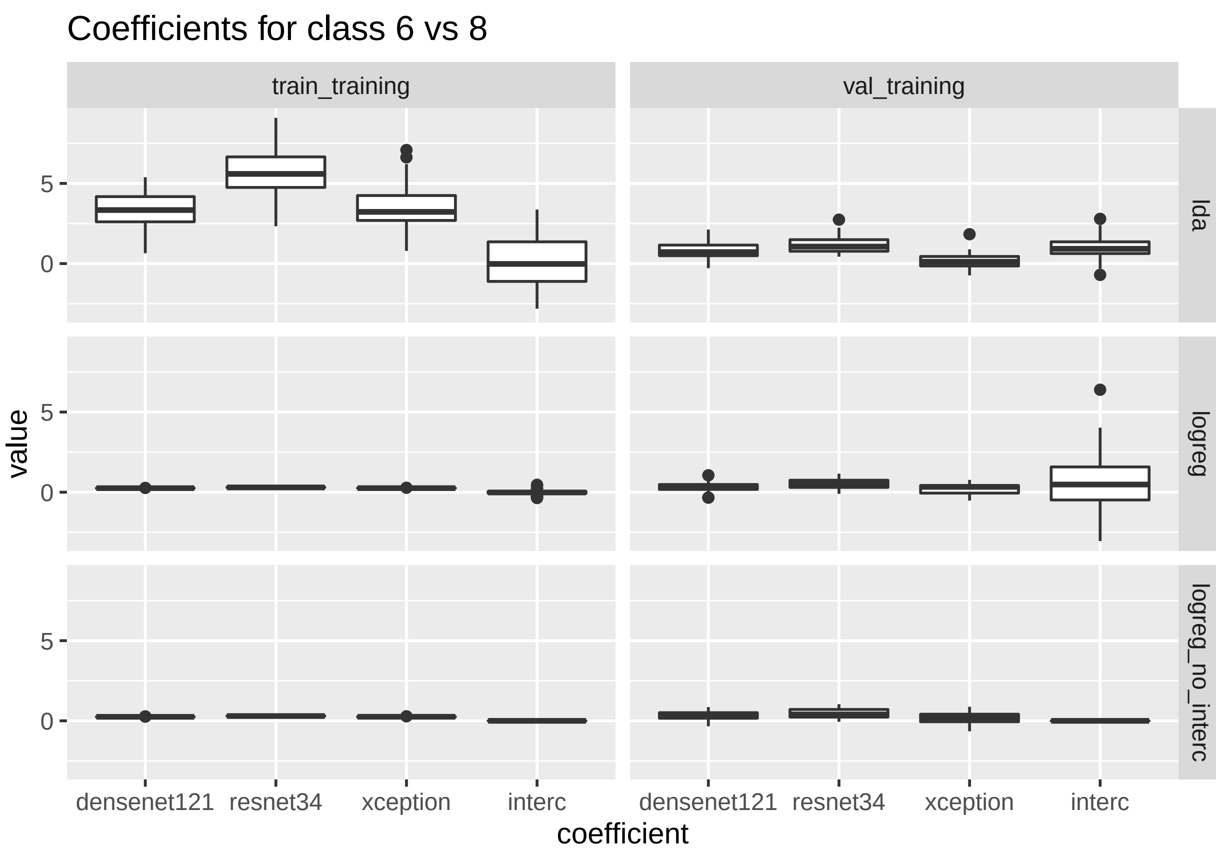

Coefficients for class 5 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
8
4
0
−4
densenet121
resnet34
xception
interc
coefficient



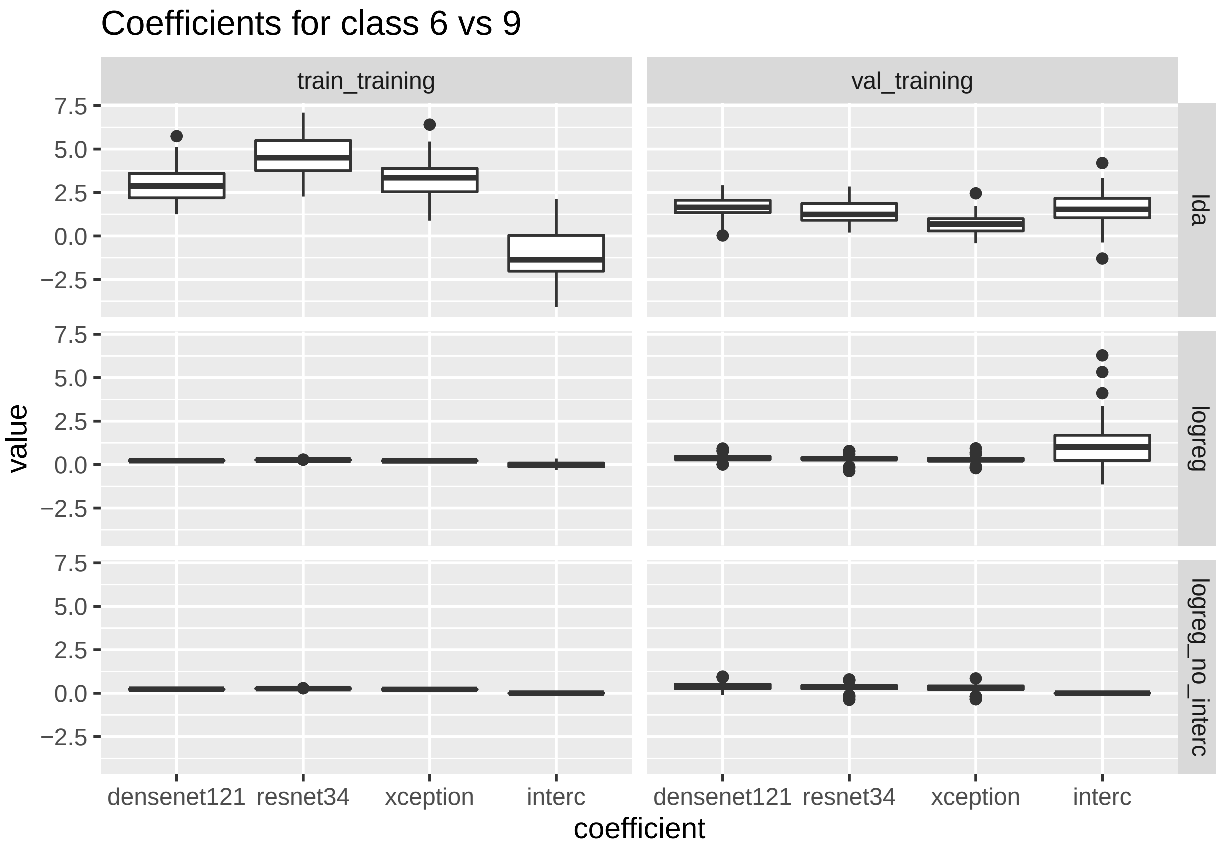
Coefficients for class 5 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
coefficient
10
5
0
−5
densenet121
resnet34
xception
interc



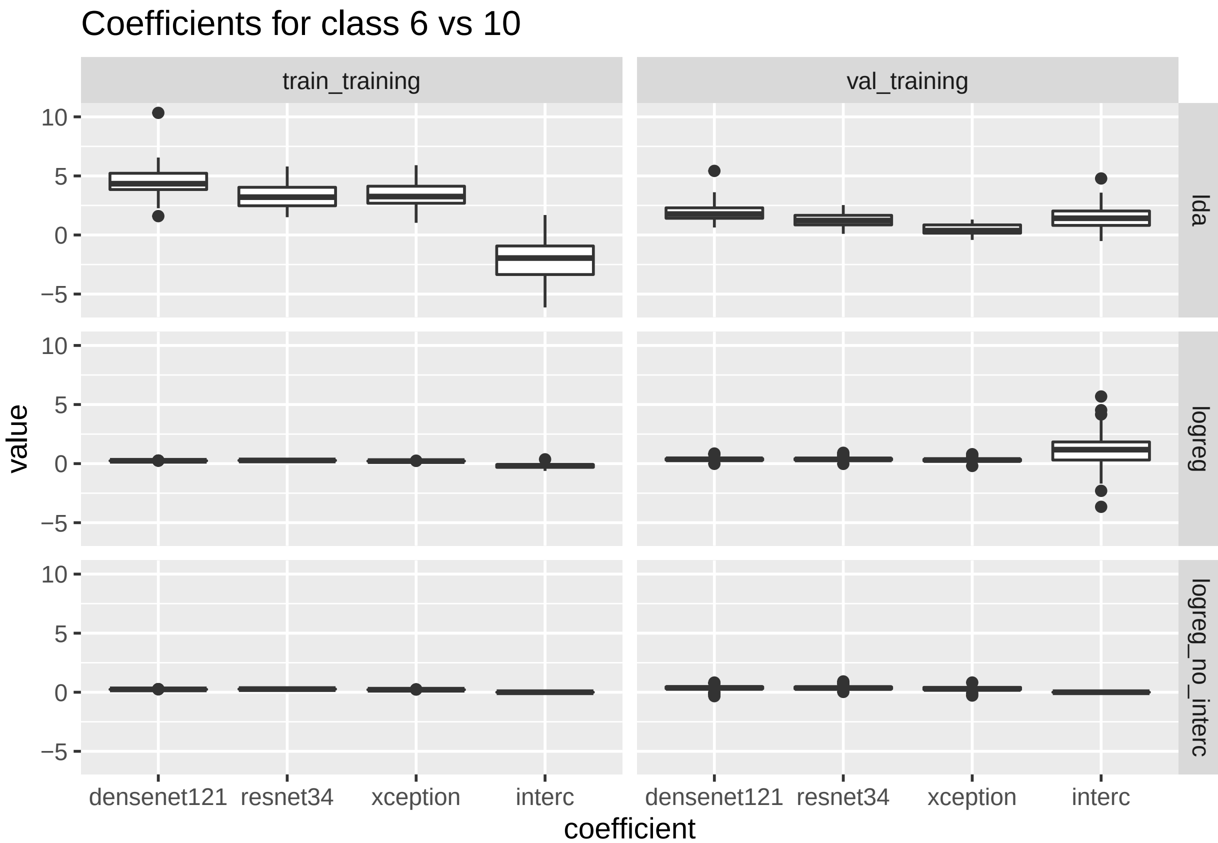
Coefficients for class 6 vs 7
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient

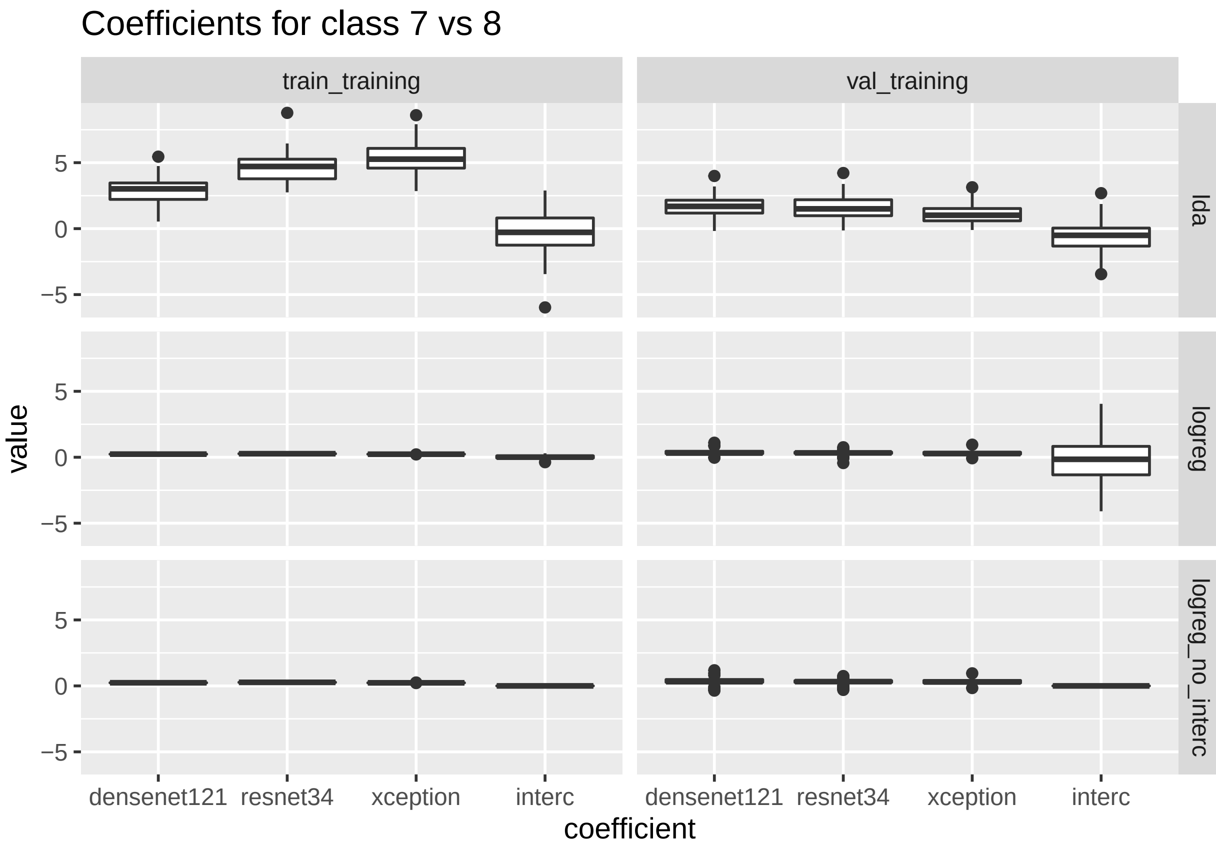


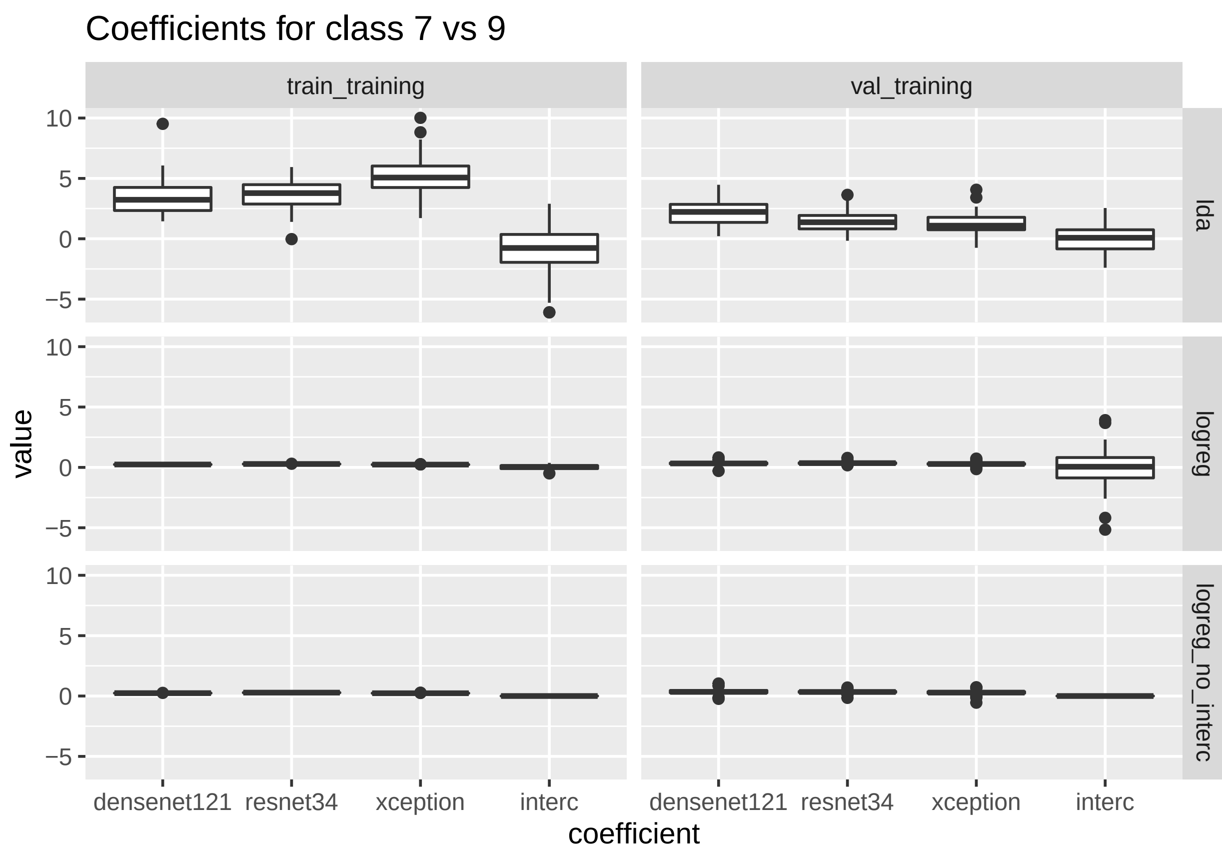
Coefficients for class 6 vs 8
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
densenet121
resnet34
xception
interc
coefficient

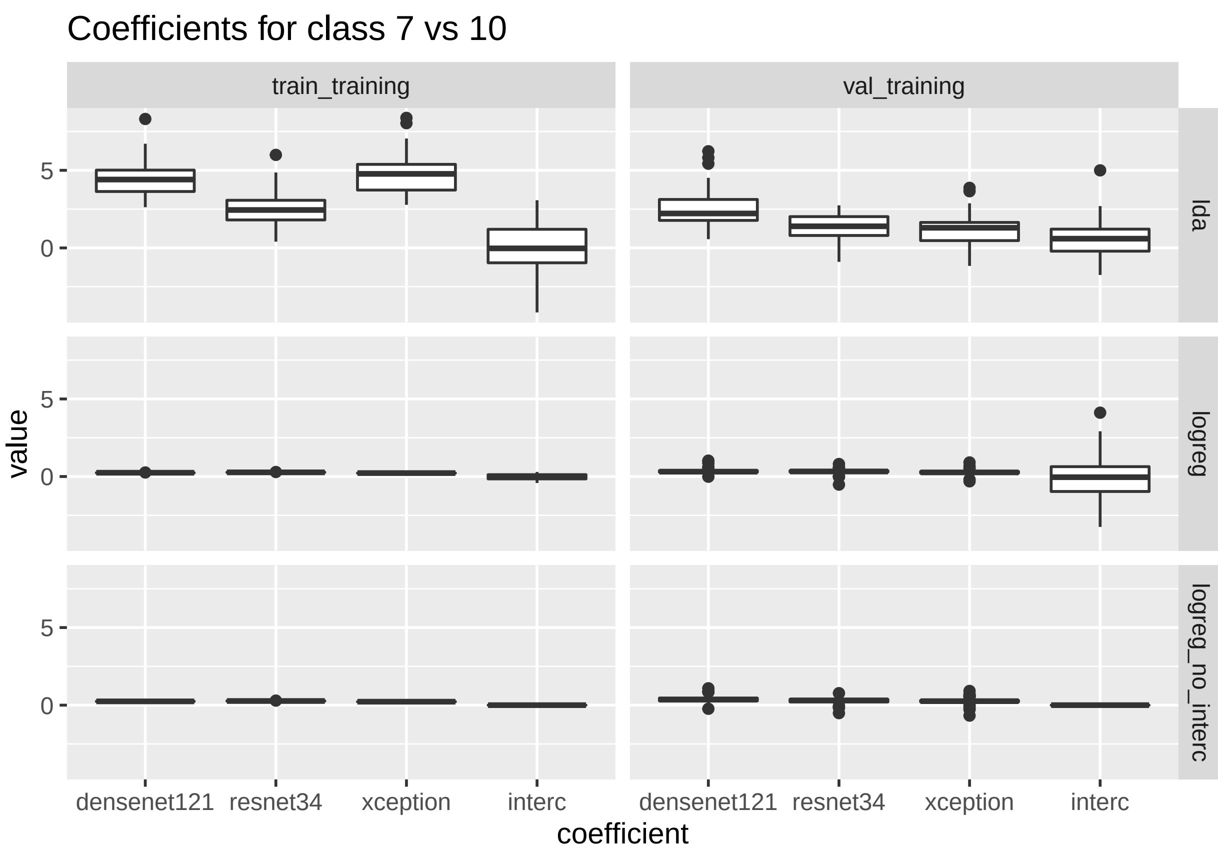


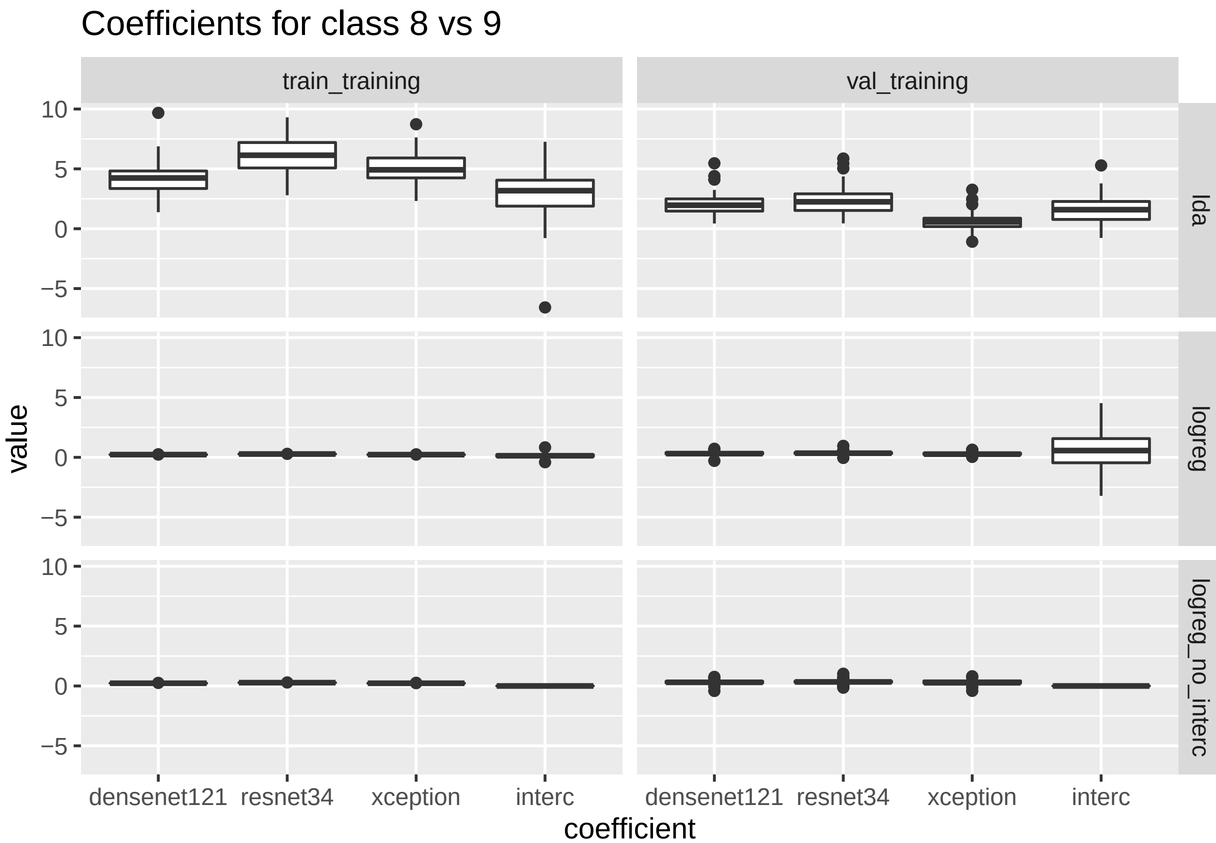
Coefficients for class 6 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
7.5
5.0
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

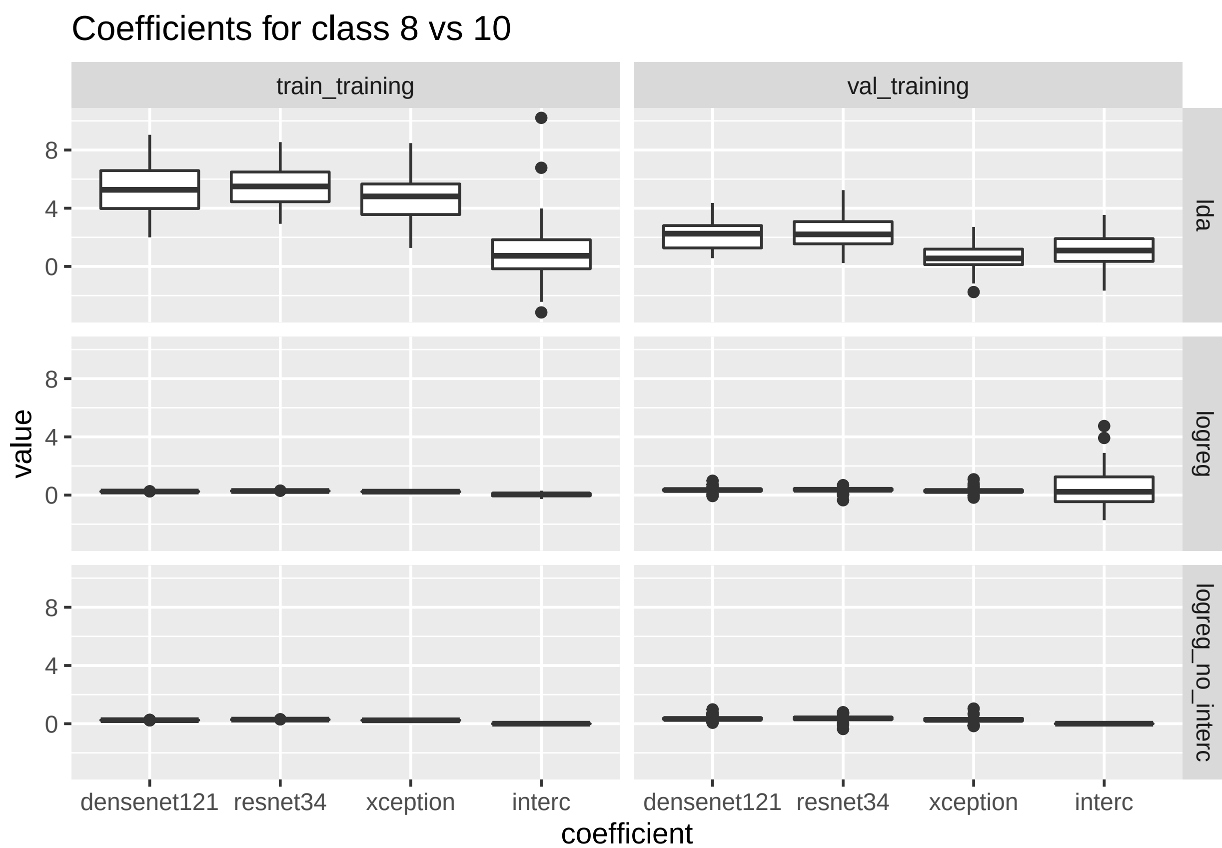

Coefficients for class 6 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient

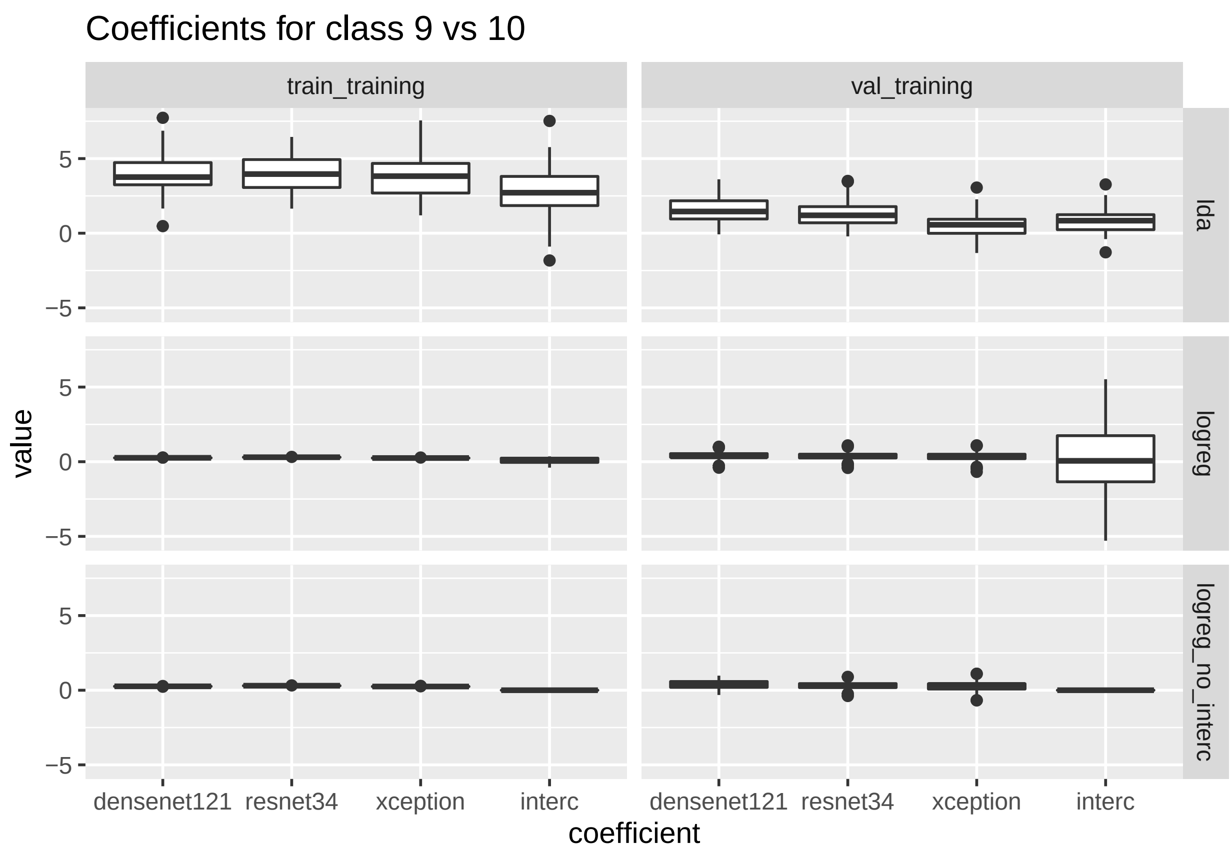

Coefficients for class 7 vs 8
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
−5
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 7 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient

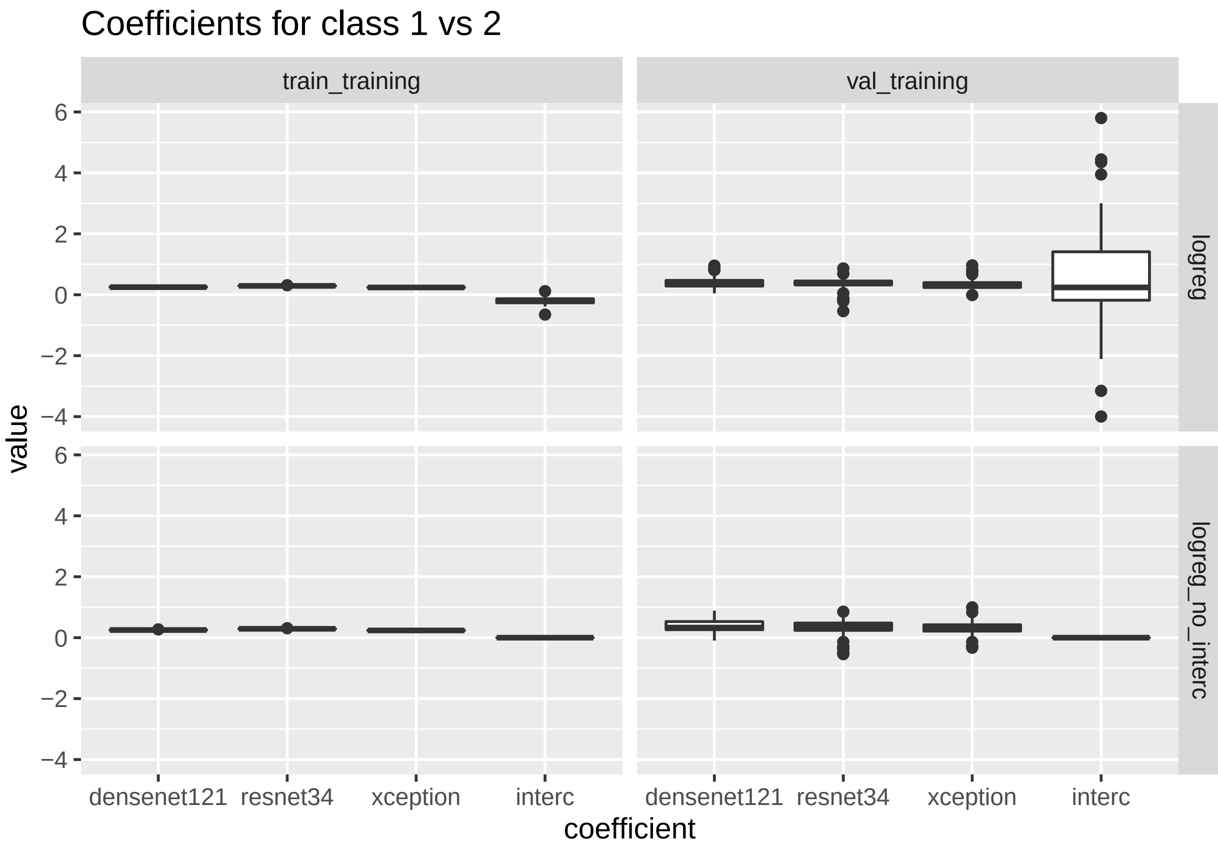

Coefficients for class 7 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
5
0
densenet121
resnet34
xception
interc
coefficient

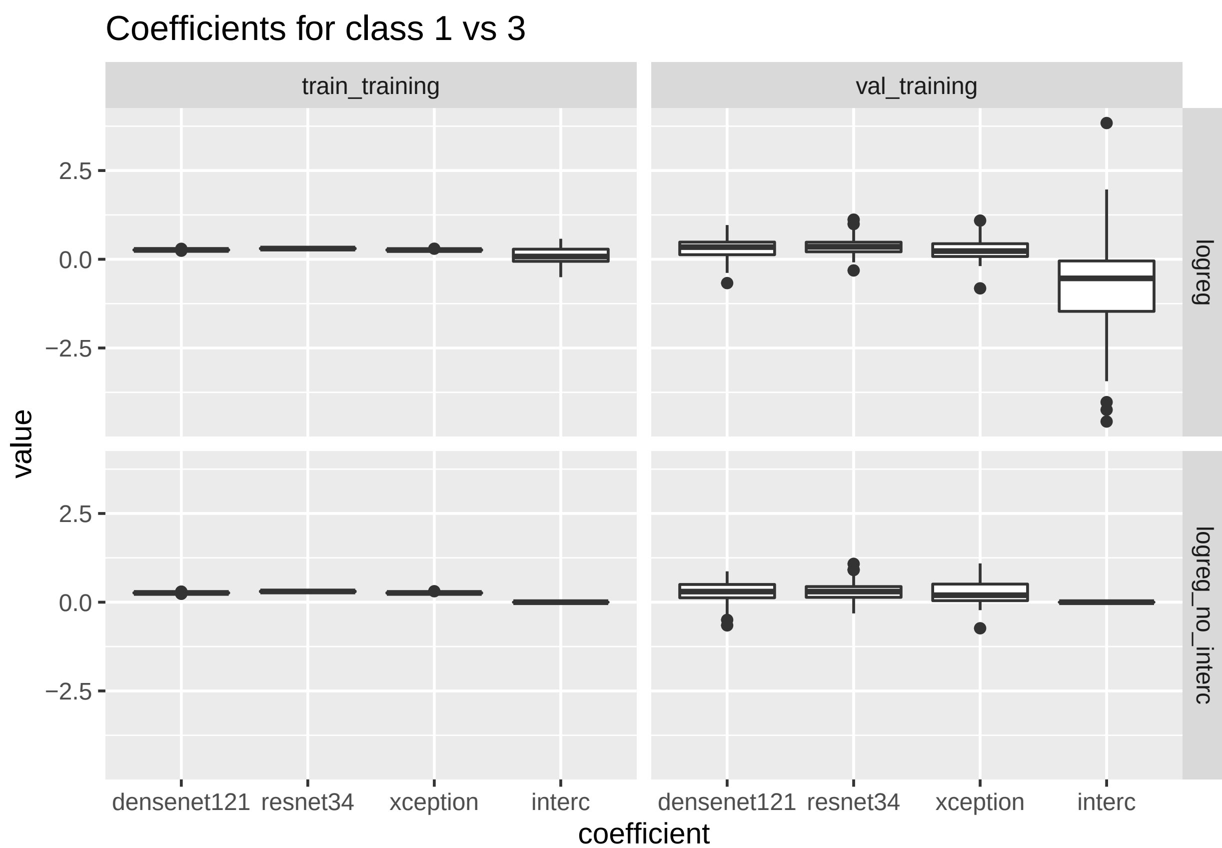

Coefficients for class 8 vs 9
train_training
val_training
lda
logreg
logreg_no_interc
value
10
5
0
−5
densenet121
resnet34
xception
interc
coefficient

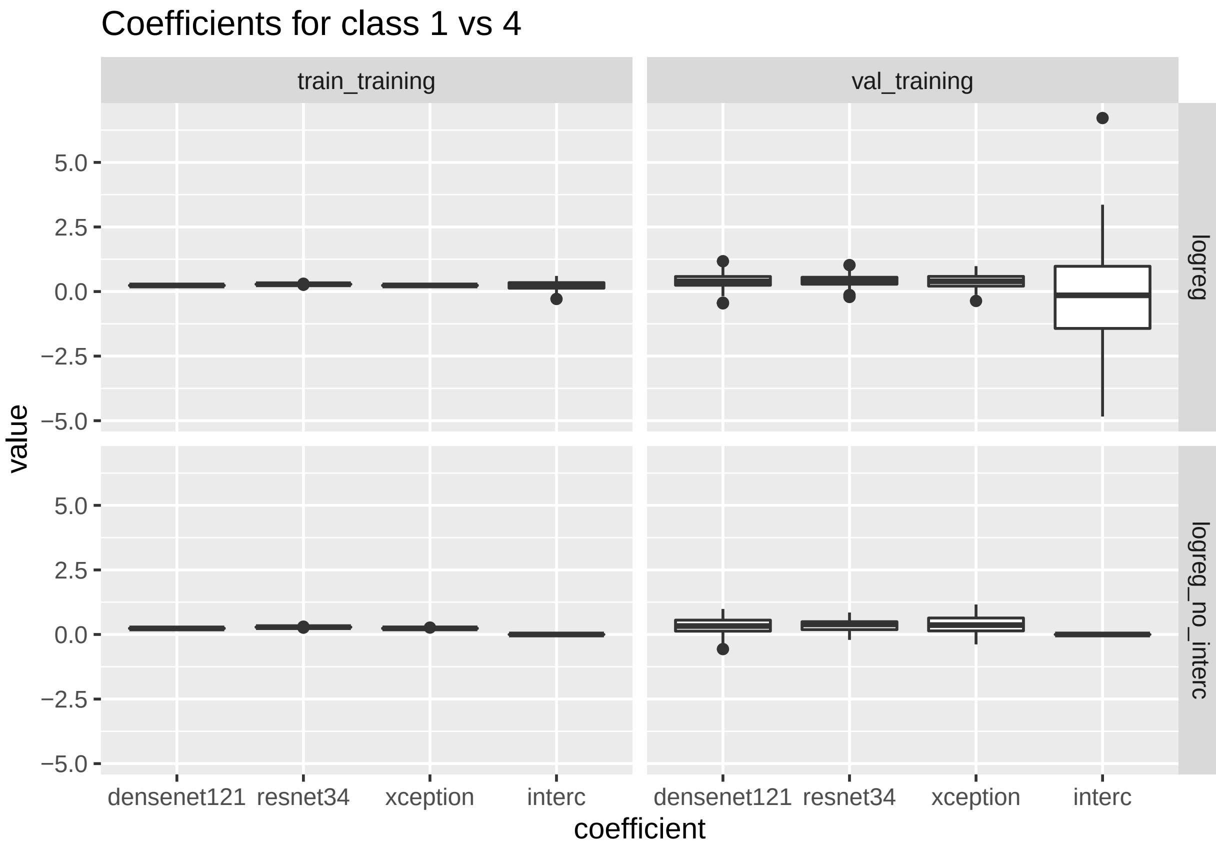

Coefficients for class 8 vs 10
train_training
val_training
lda
logreg
logreg_no_interc
value
8
4
0
densenet121
resnet34
xception
interc
coefficient



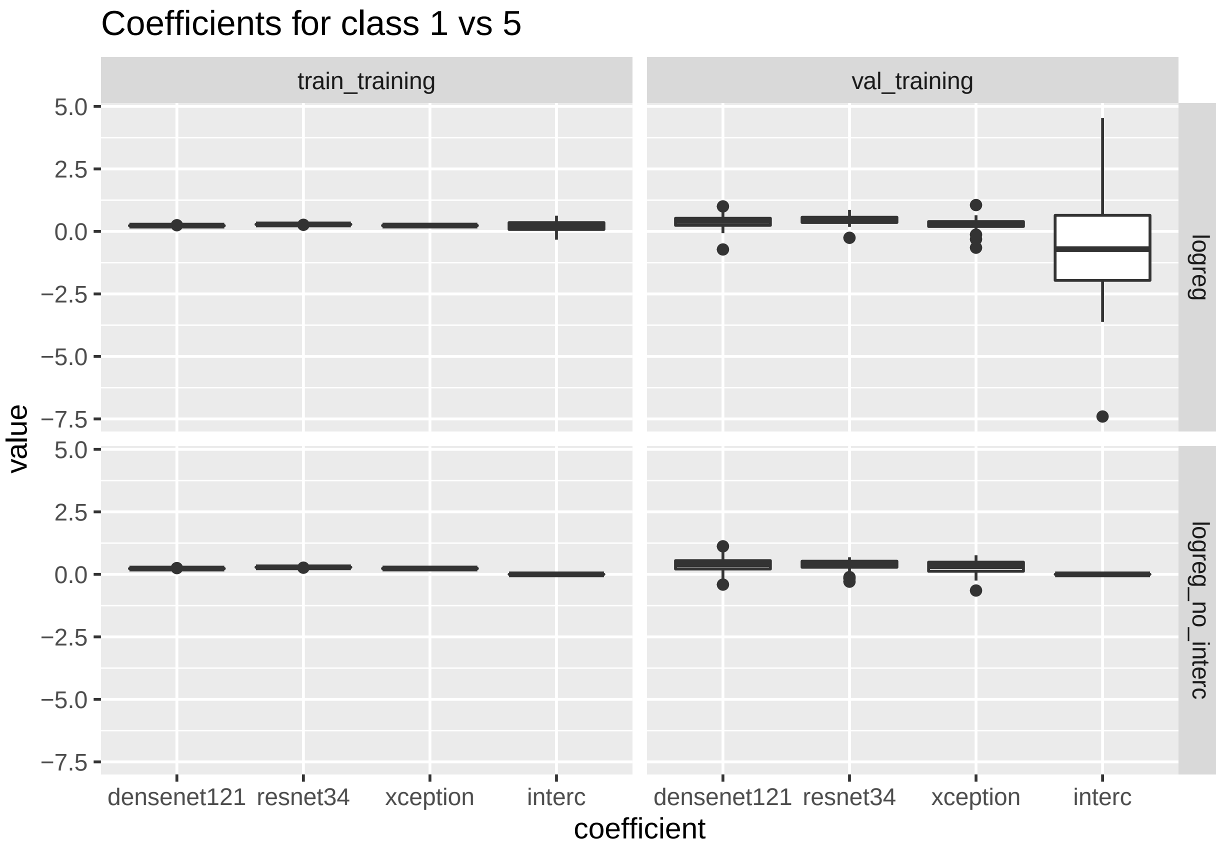

```
for (cl1 in 1:(classes - 1))
{
  for (cl2 in (cl1 + 1):classes)
  {
    cur_plt <- combiner_coefs %>% filter(class1 == cl1 & class2 == cl2 & combining_method!="lda") %>% gg
      facet_grid(cols=vars(train_type), rows=vars(combining_method)) + ggtitle(paste("Coefficients for c
    print(cur_plt)
  }
}
```

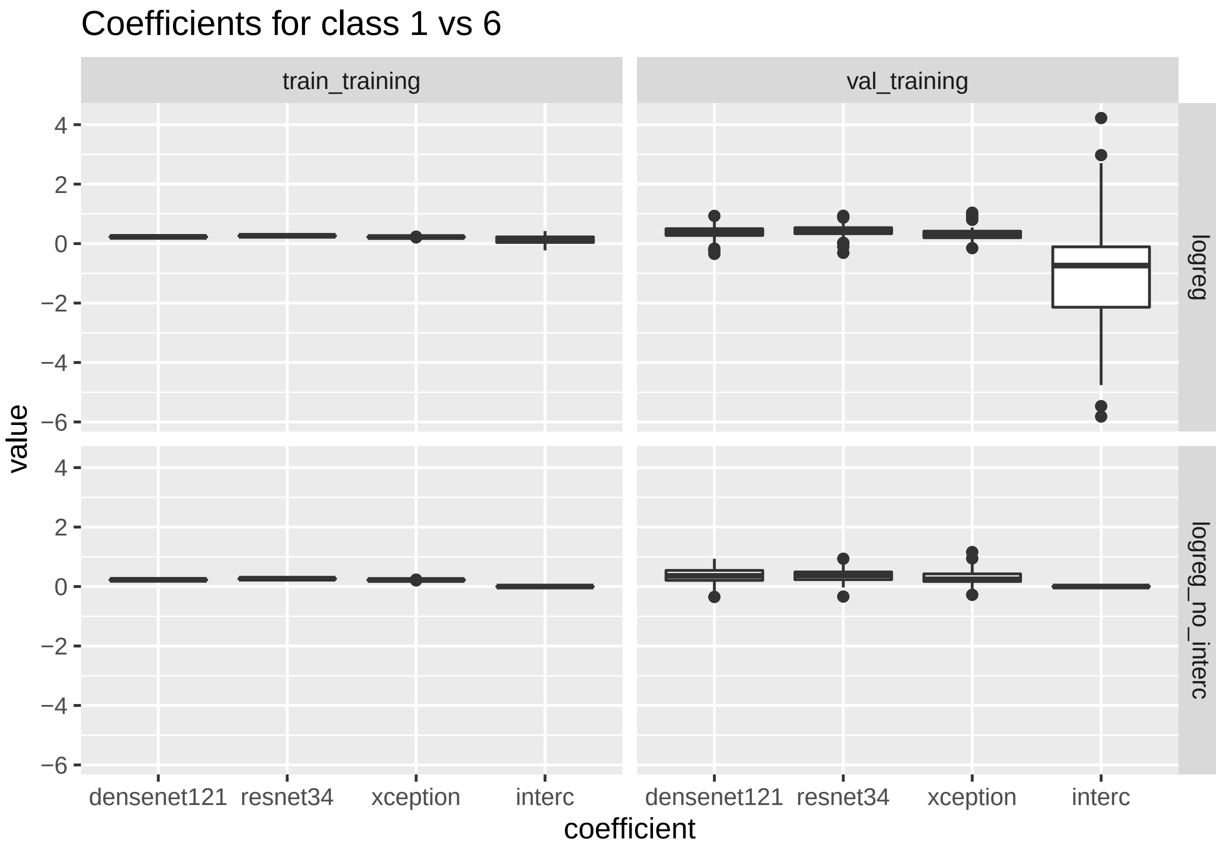

Coefficients for class 1 vs 2
train_training
val_training
logreg
logreg_no_interc
value
6
4
2
0
−2
−4
densenet121
resnet34
xception
interc
coefficient

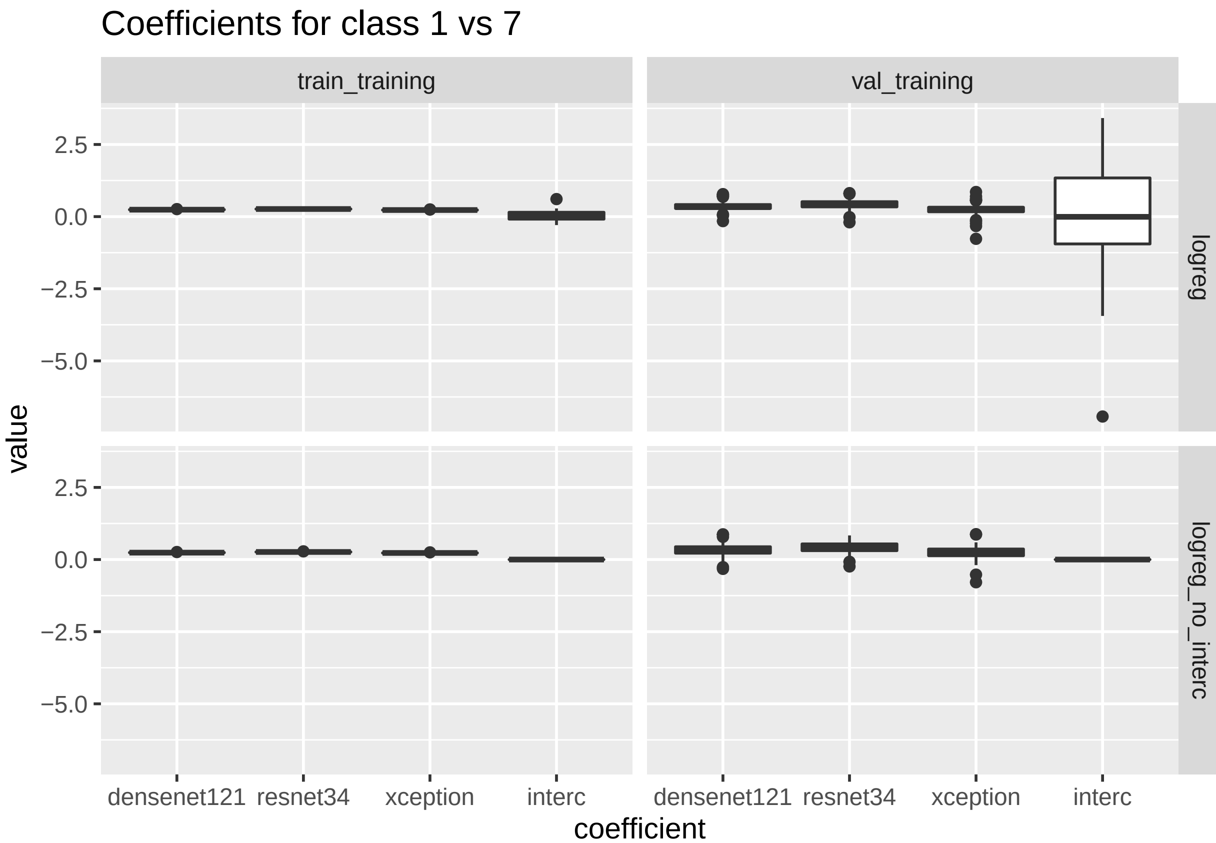

Coefficients for class 1 vs 3
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
densenet121
resnet34
xception
interc
coefficient

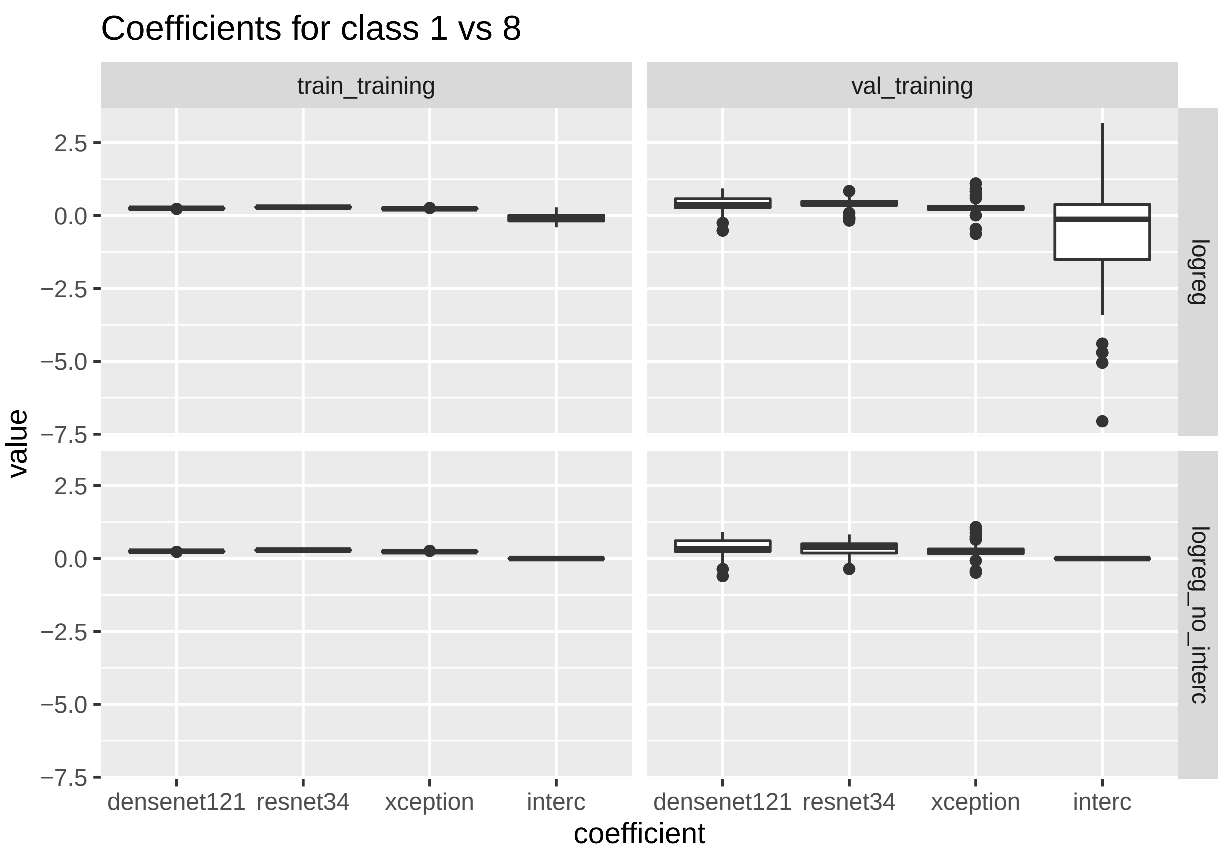


Coefficients for class 1 vs 4
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient

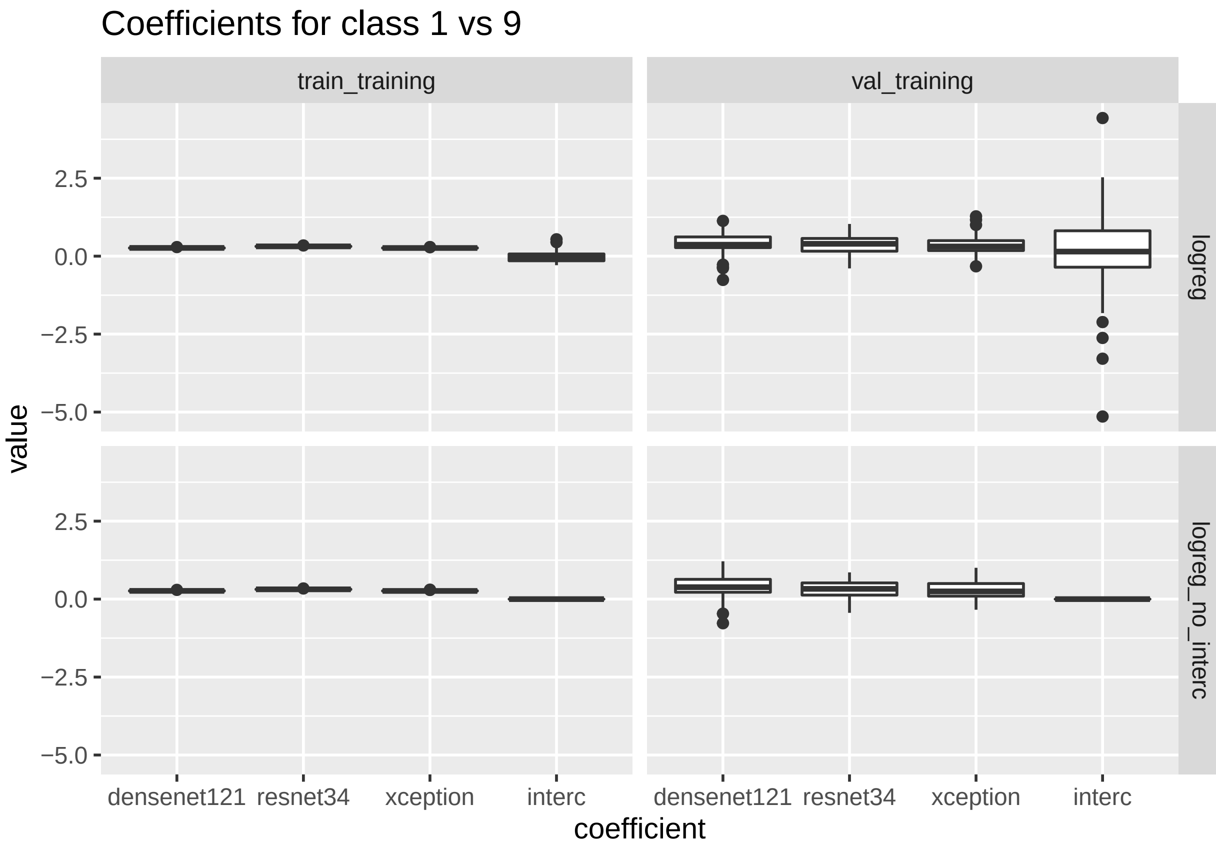

Coefficients for class 1 vs 5
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
−5.0
−7.5
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 1 vs 6
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
−4
−6
densenet121
resnet34
xception
interc
coefficient

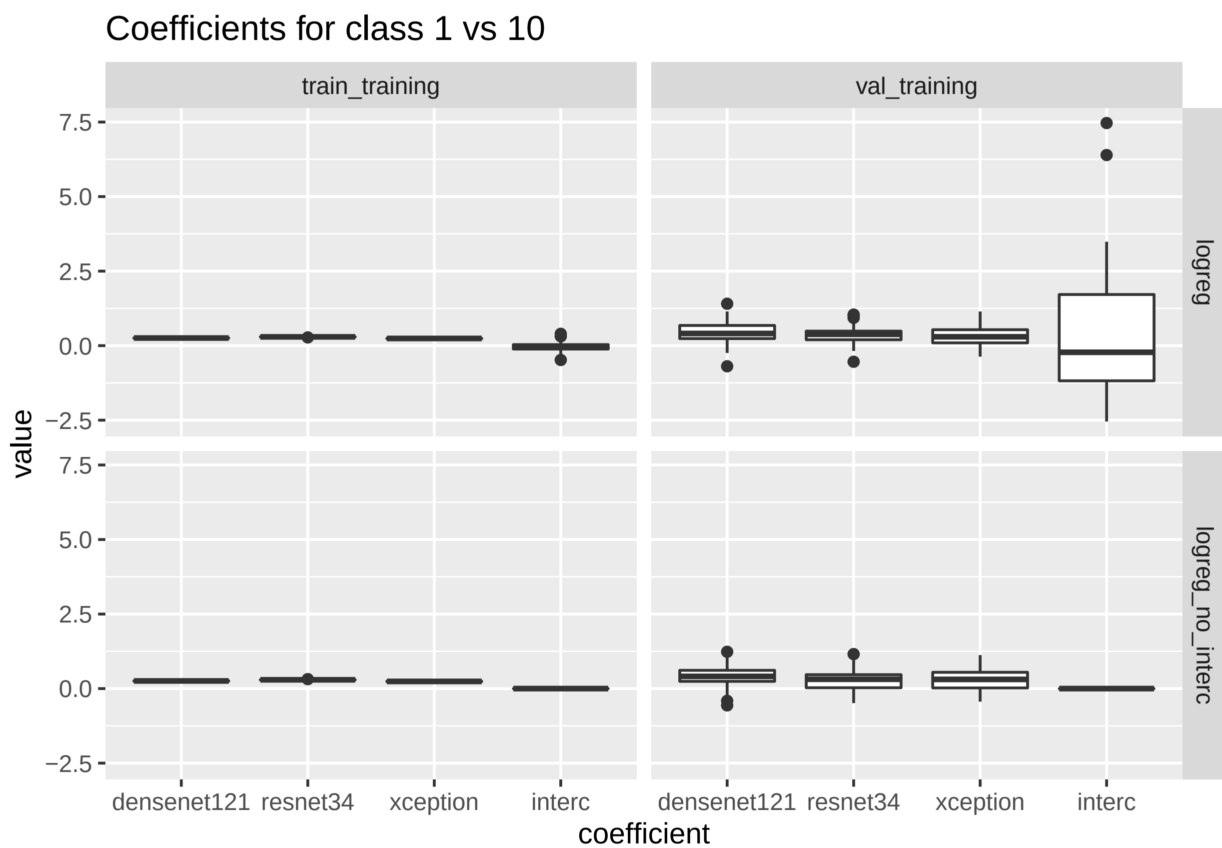

Coefficients for class 1 vs 7
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient

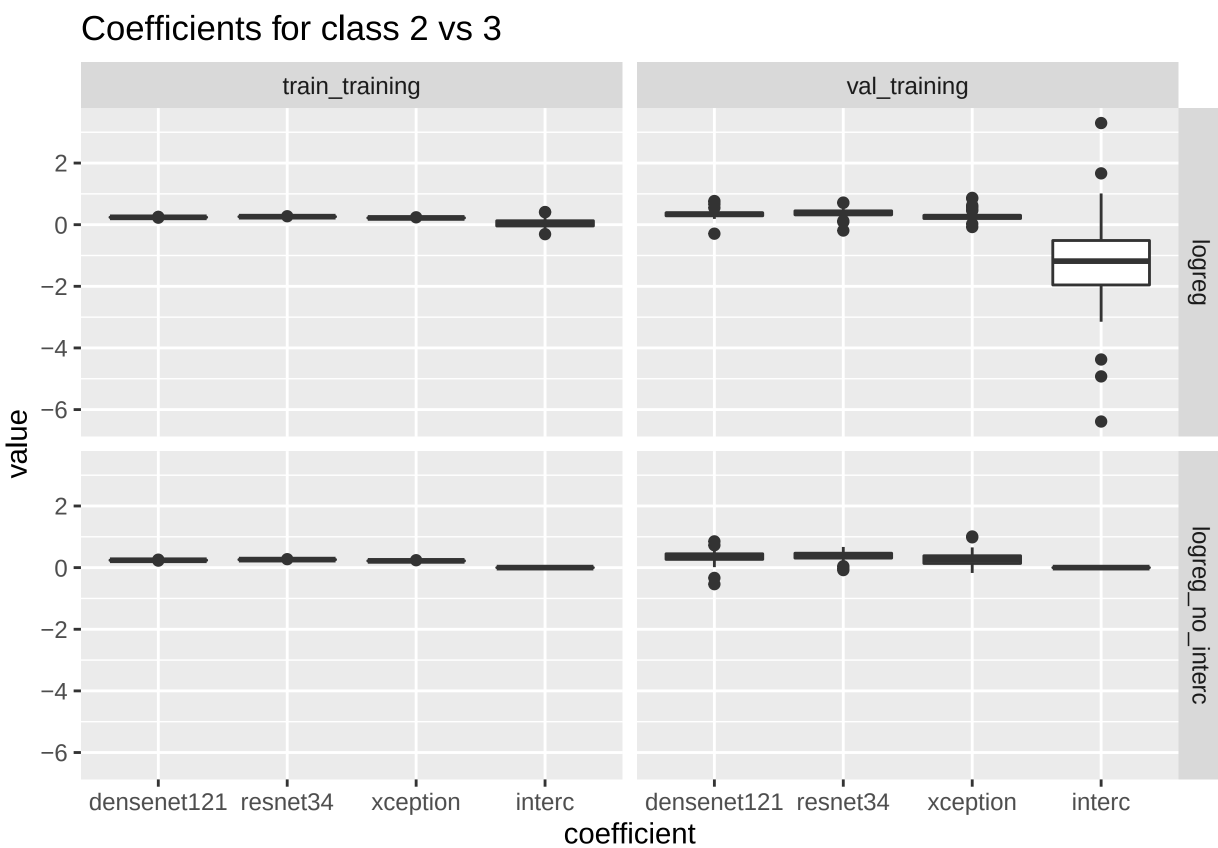


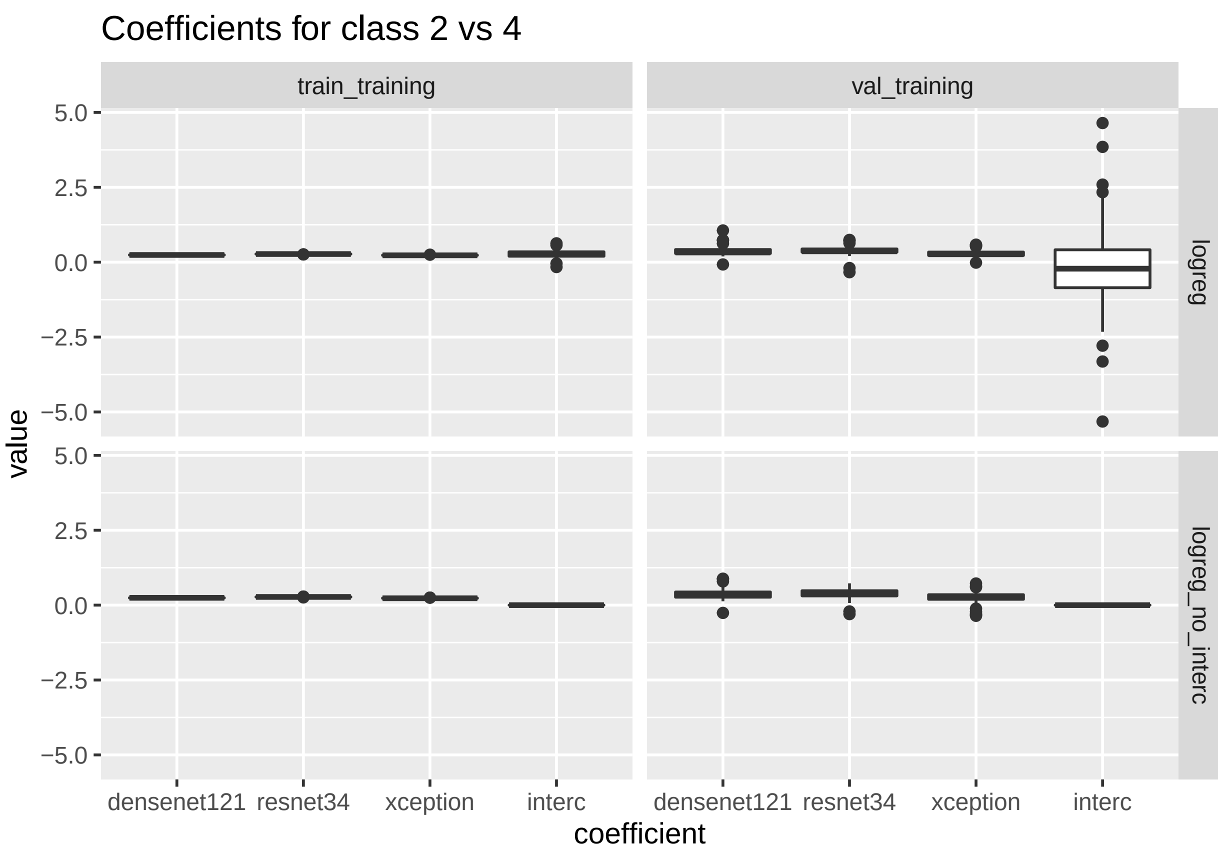
Coefficients for class 1 vs 8
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
−5.0
−7.5
densenet121
resnet34
xception
interc
coefficient

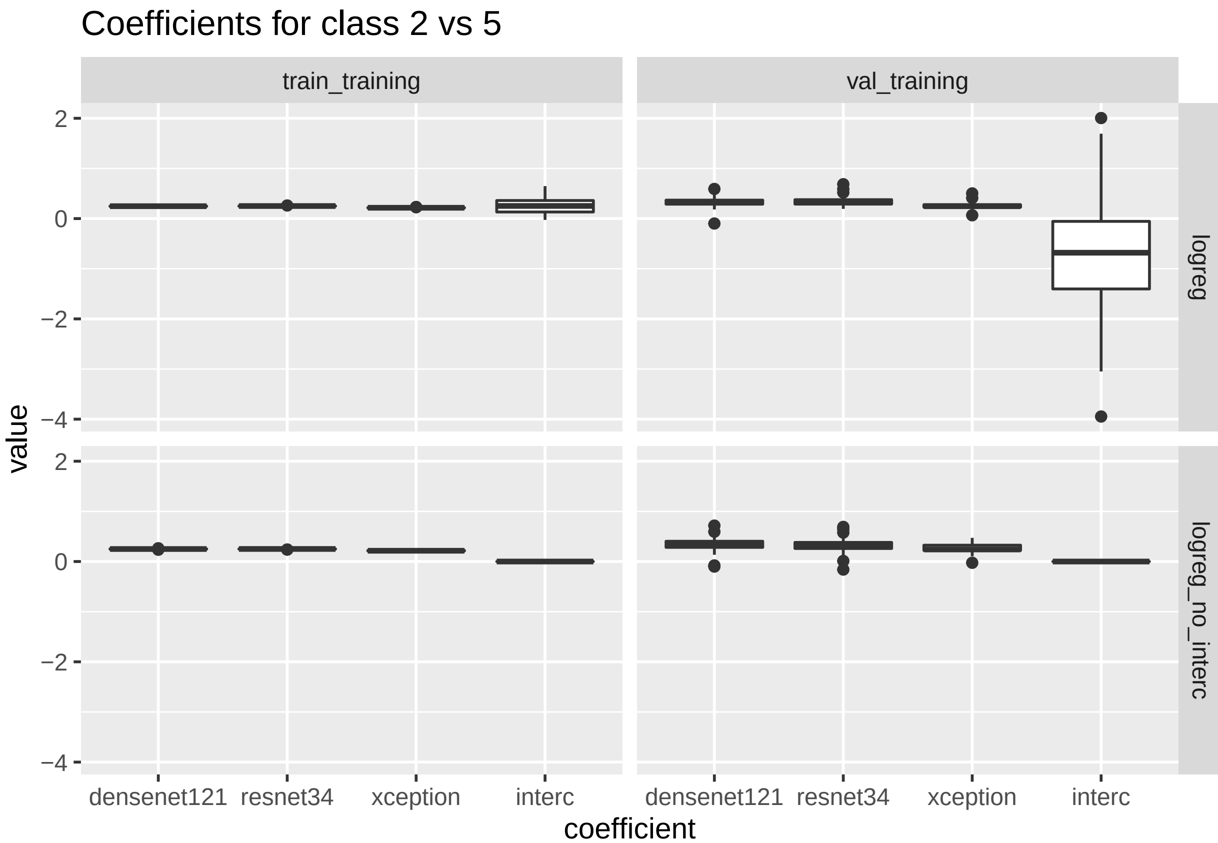


Coefficients for class 1 vs 9
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient

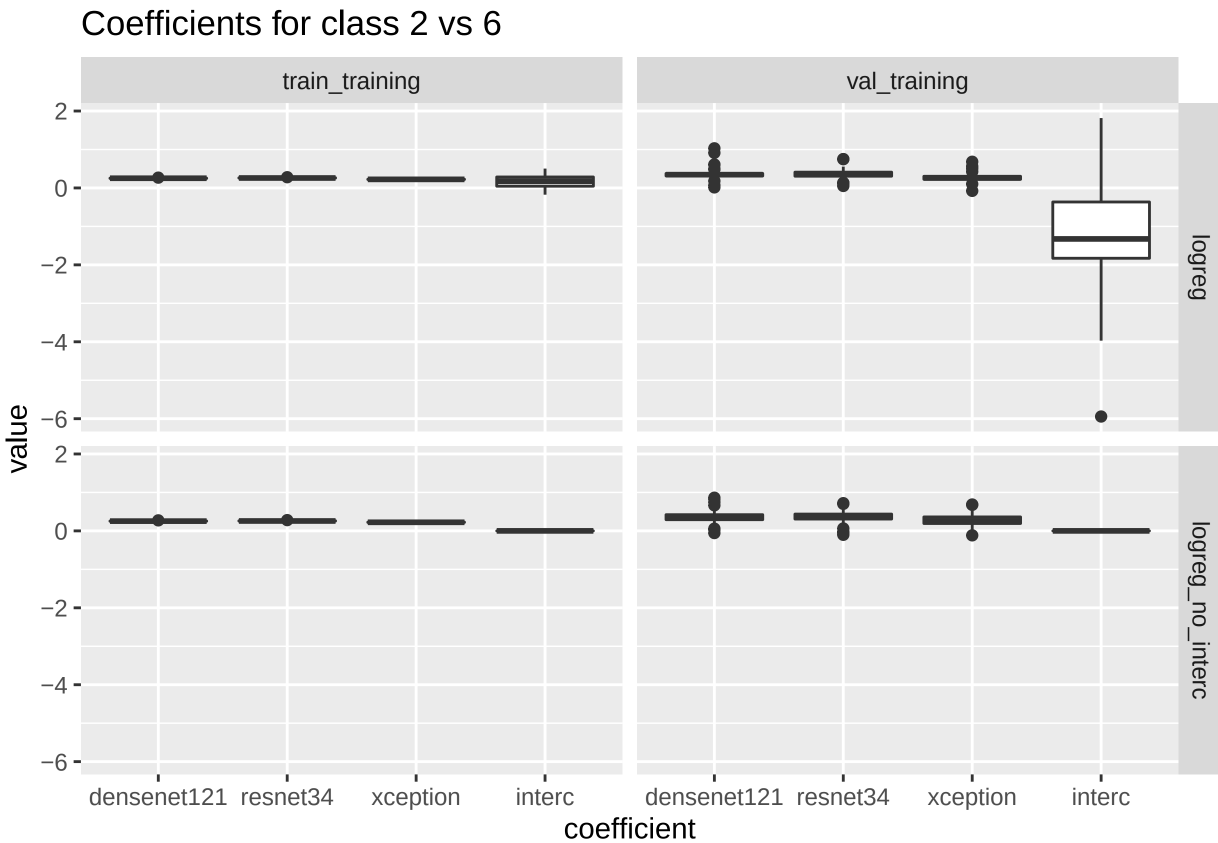

Coefficients for class 1 vs 10
train_training
val_training
logreg
logreg_no_interc
value
7.5
5.0
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

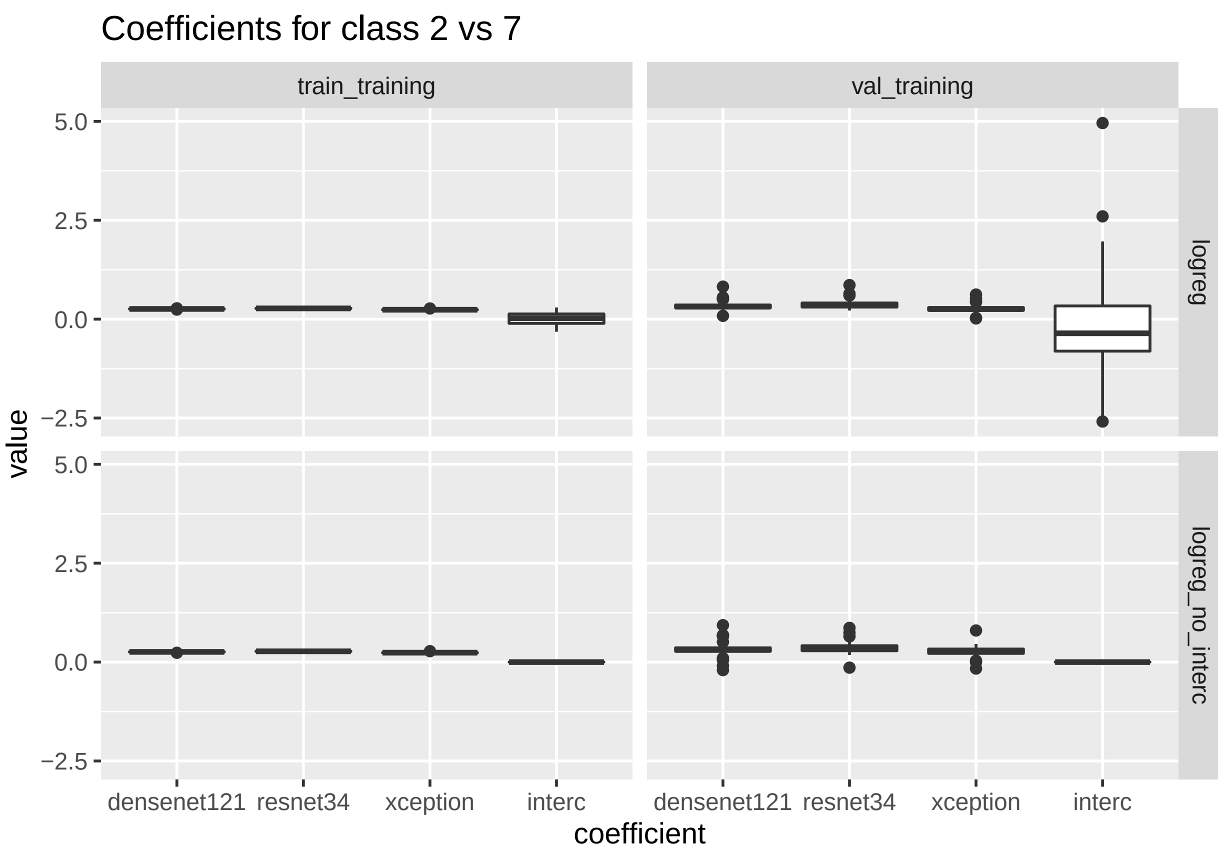

Coefficients for class 2 vs 3
train_training
val_training
logreg
logreg_no_interc
value
2
0
−2
−4
−6
densenet121
resnet34
xception
interc
coefficient

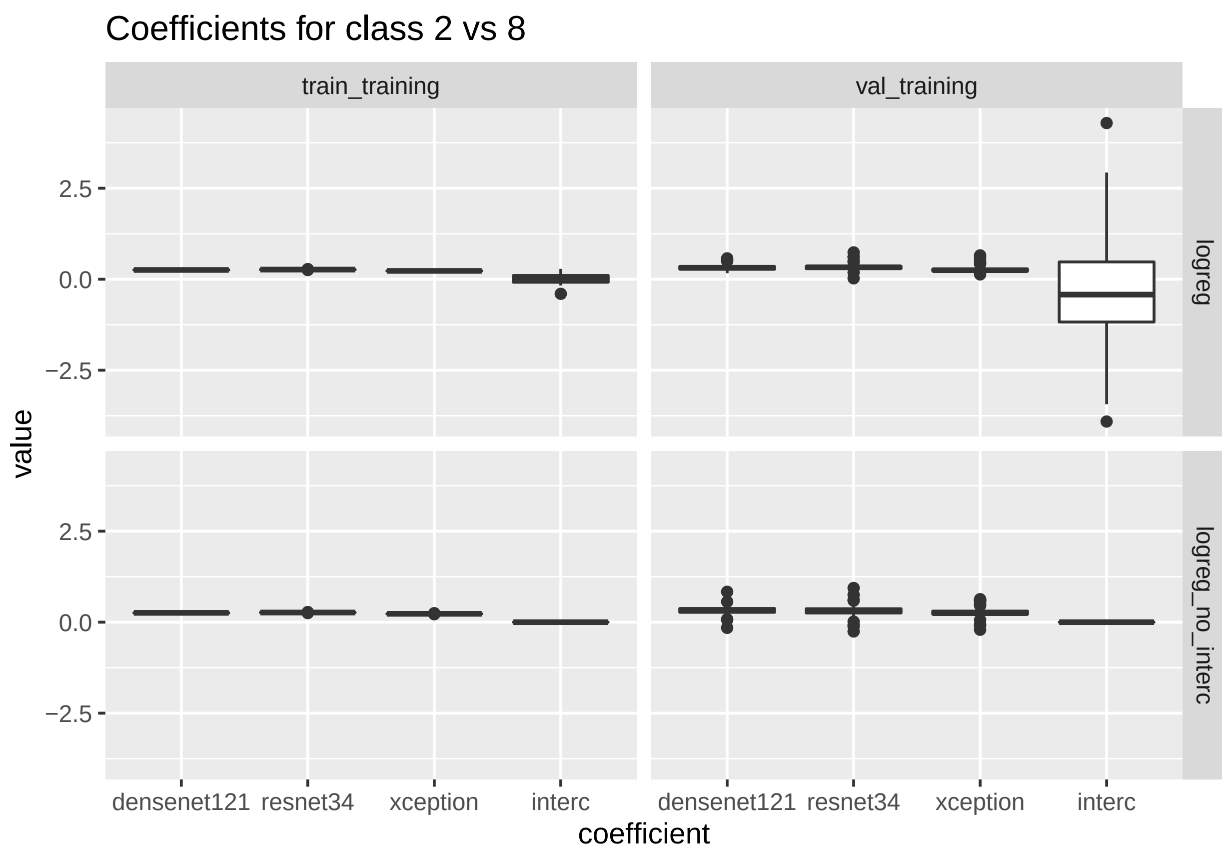

Coefficients for class 2 vs 4
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient



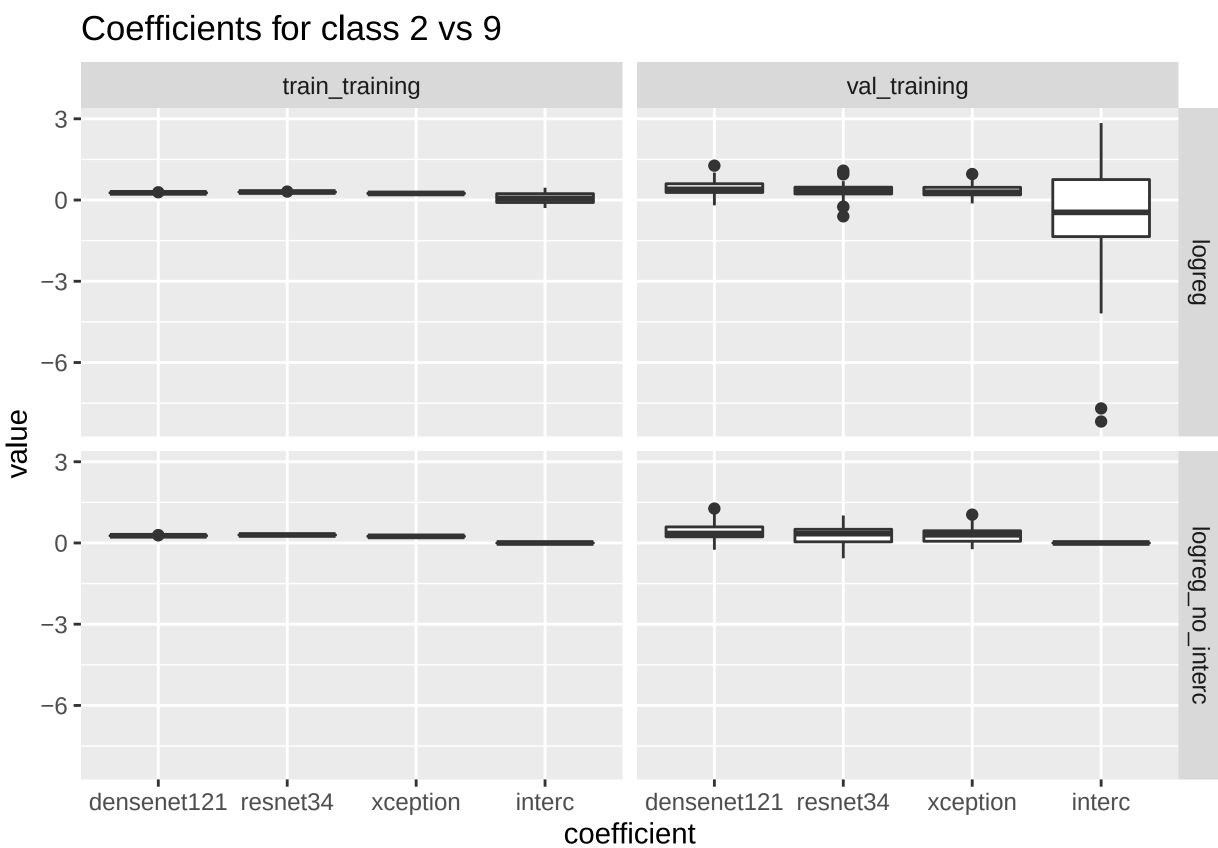
Coefficients for class 2 vs 5
train_training
val_training
logreg
logreg_no_interc
value
2
0
−2
−4
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 2 vs 6
train_training
val_training
logreg
logreg_no_interc
value
2
0
−2
−4
−6
densenet121
resnet34
xception
interc
coefficient

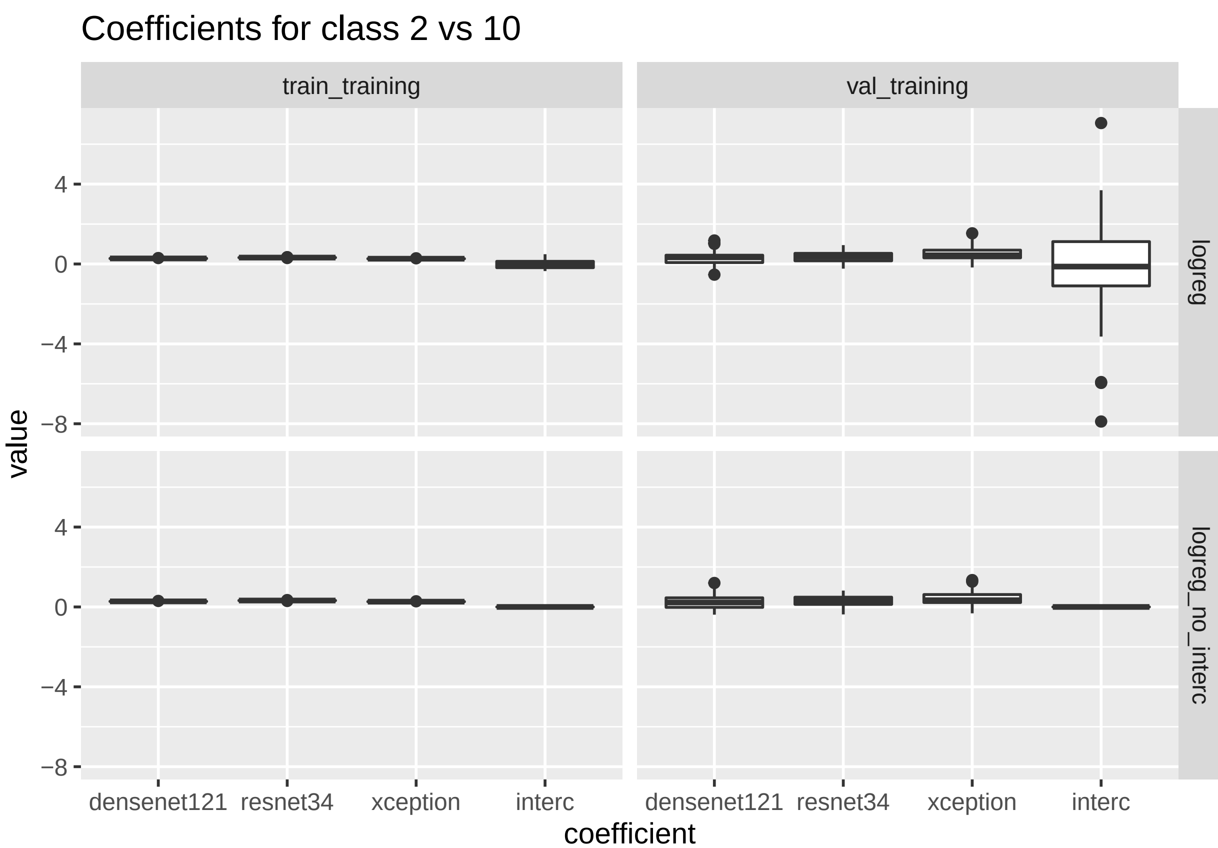

Coefficients for class 2 vs 7
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

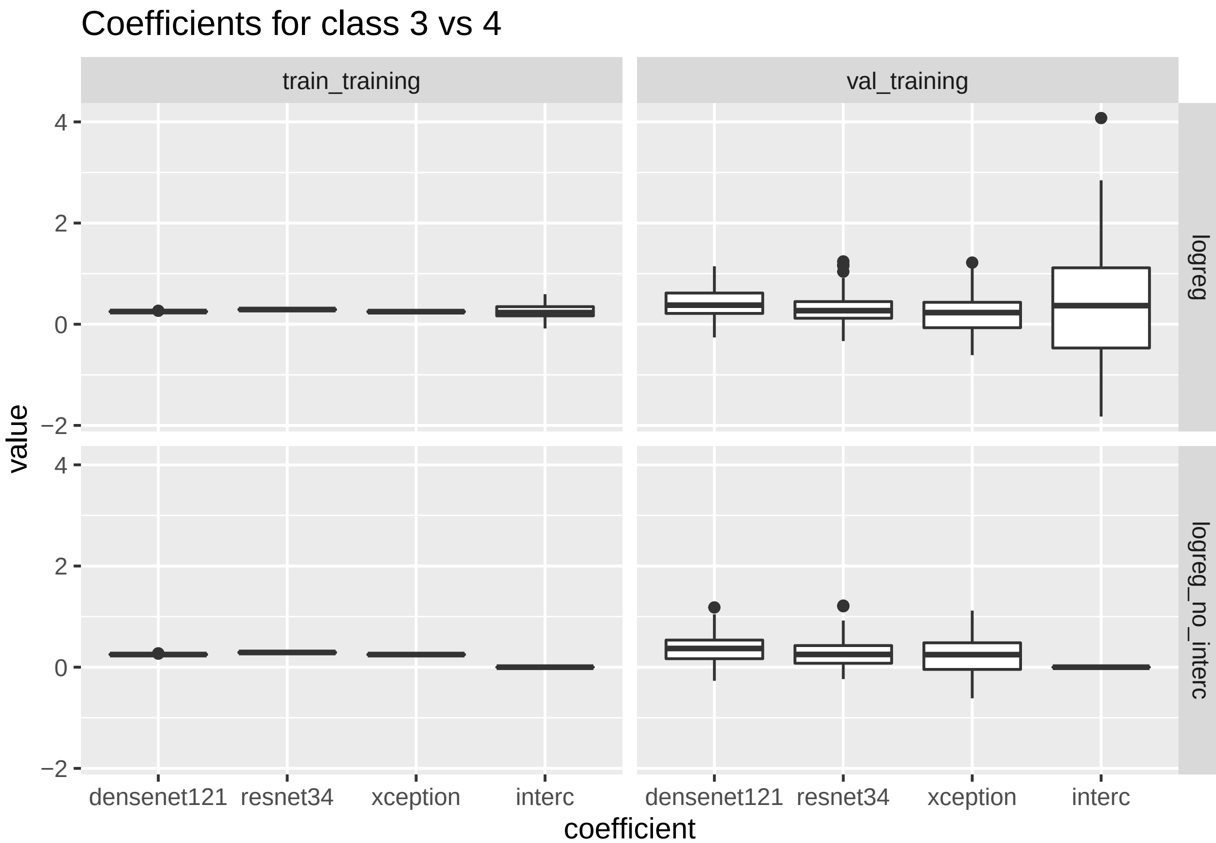


Coefficients for class 2 vs 8
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

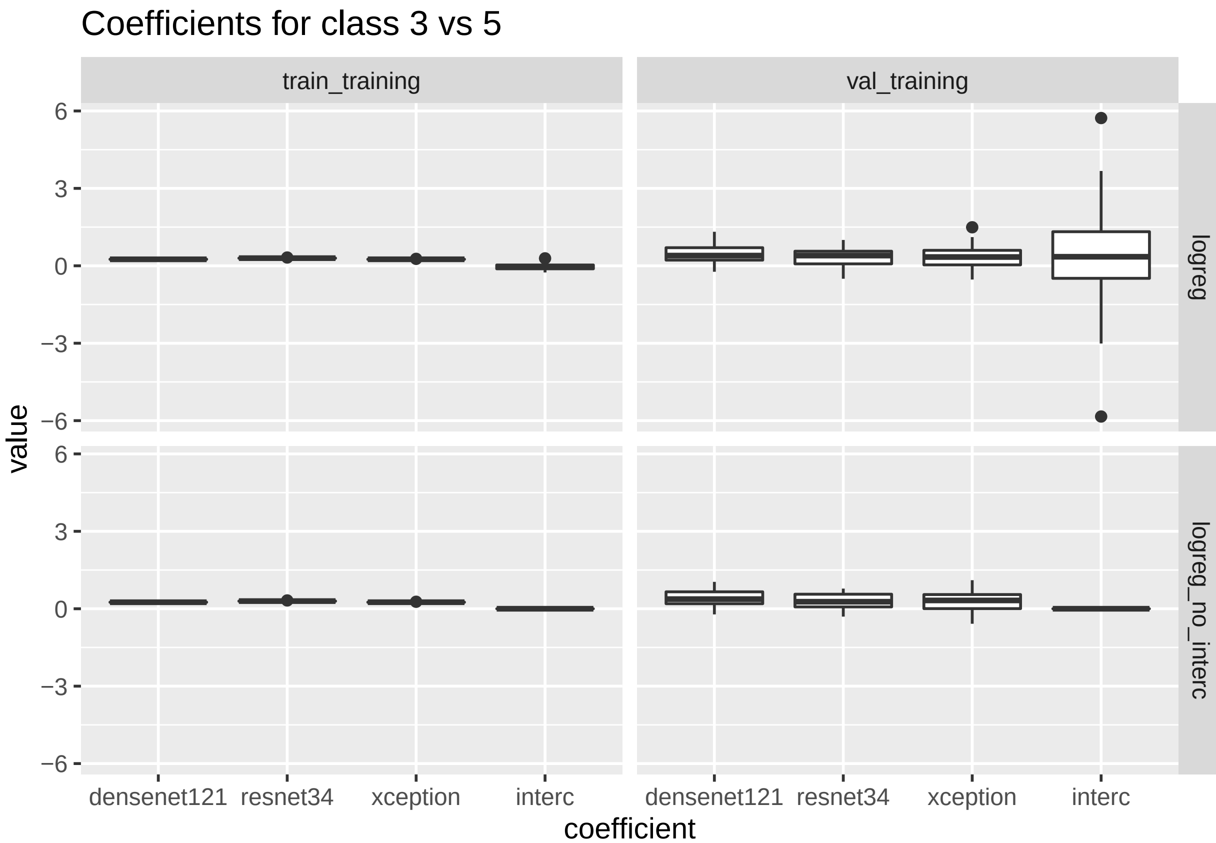


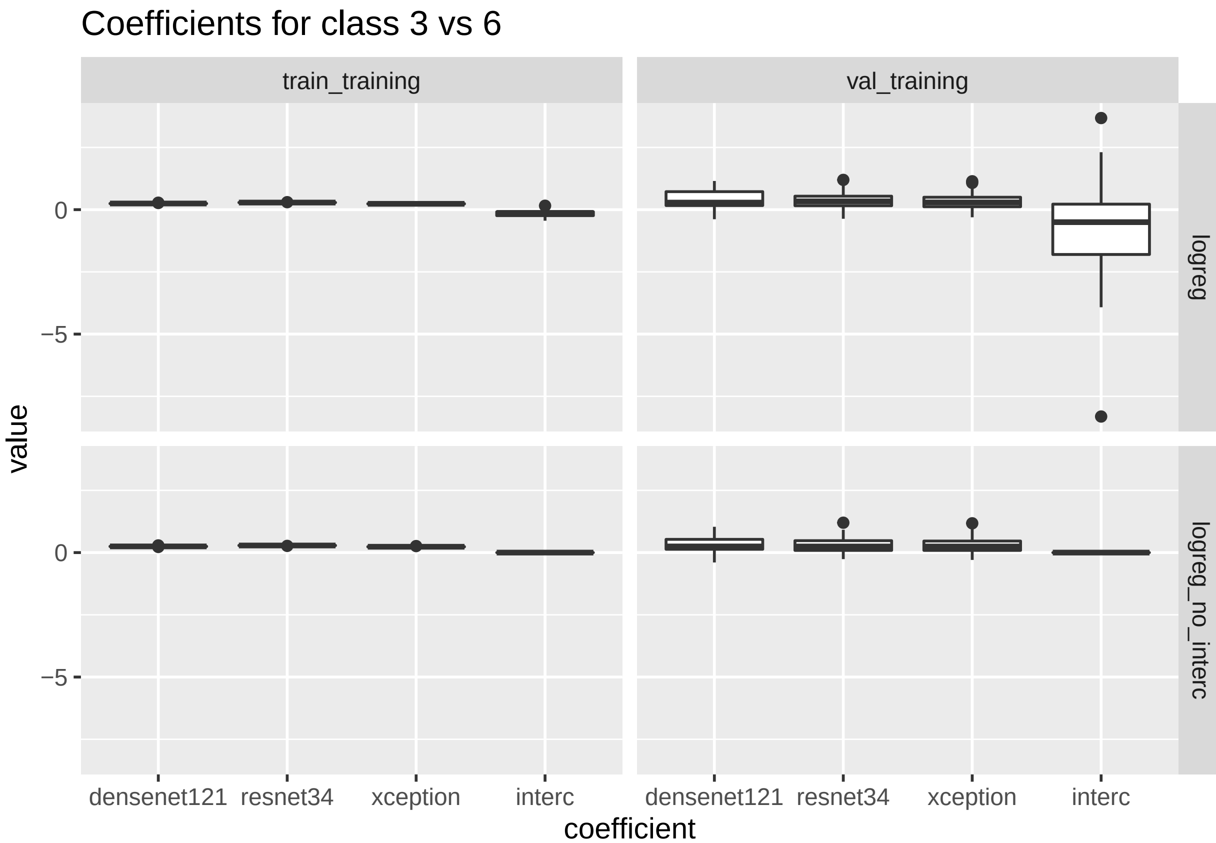
Coefficients for class 2 vs 9
train_training
val_training
logreg
logreg_no_interc
value
3
0
−3
−6
densenet121
resnet34
xception
interc
coefficient

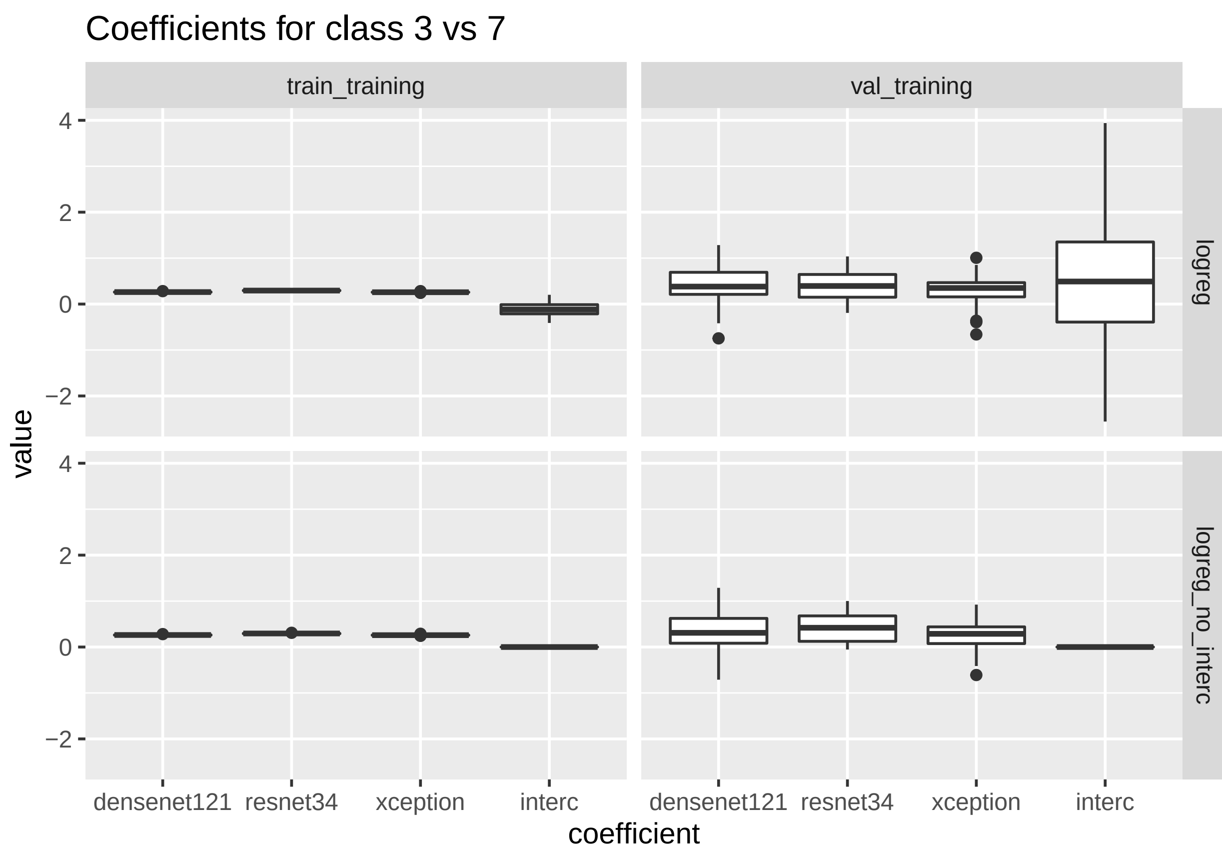

Coefficients for class 2 vs 10
train_training
val_training
logreg
logreg_no_interc
value
4
0
−4
−8
densenet121
resnet34
xception
interc
coefficient

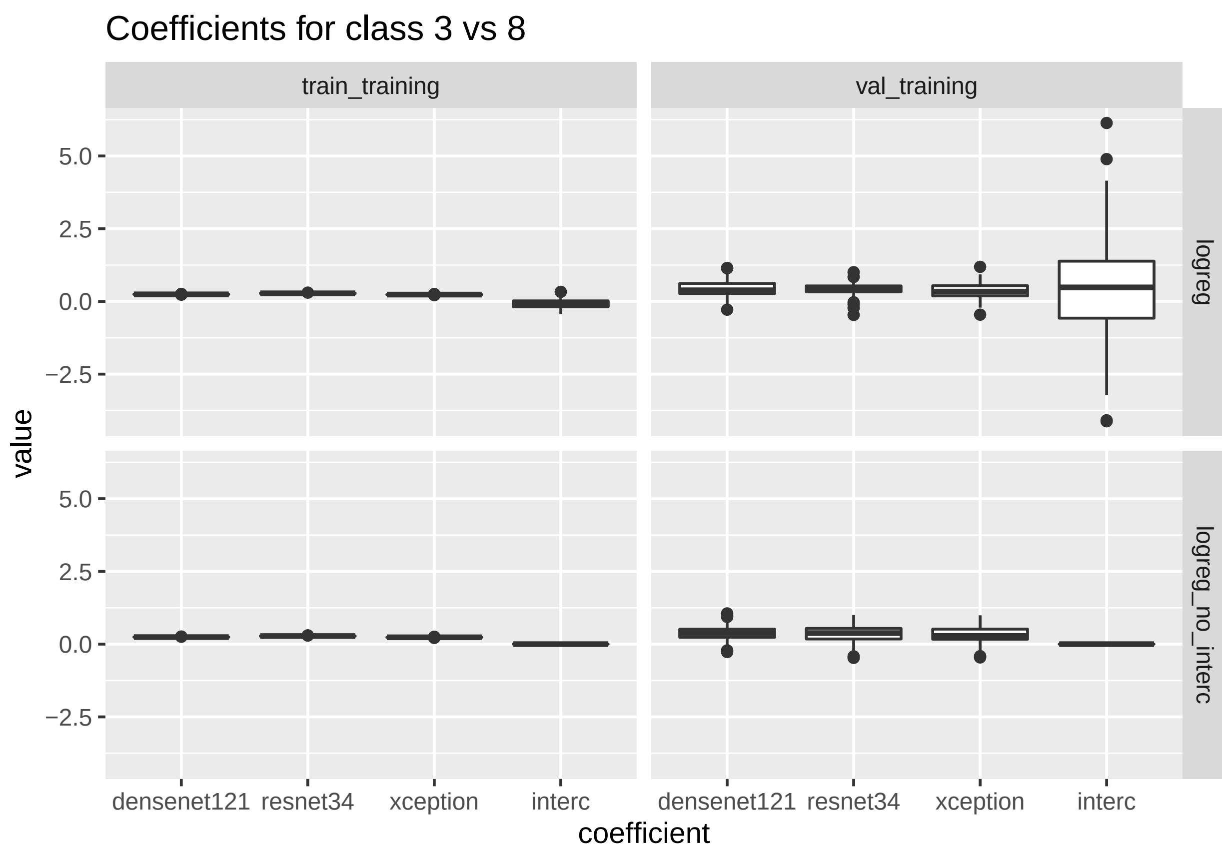


Coefficients for class 3 vs 4
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
densenet121
resnet34
xception
interc
coefficient

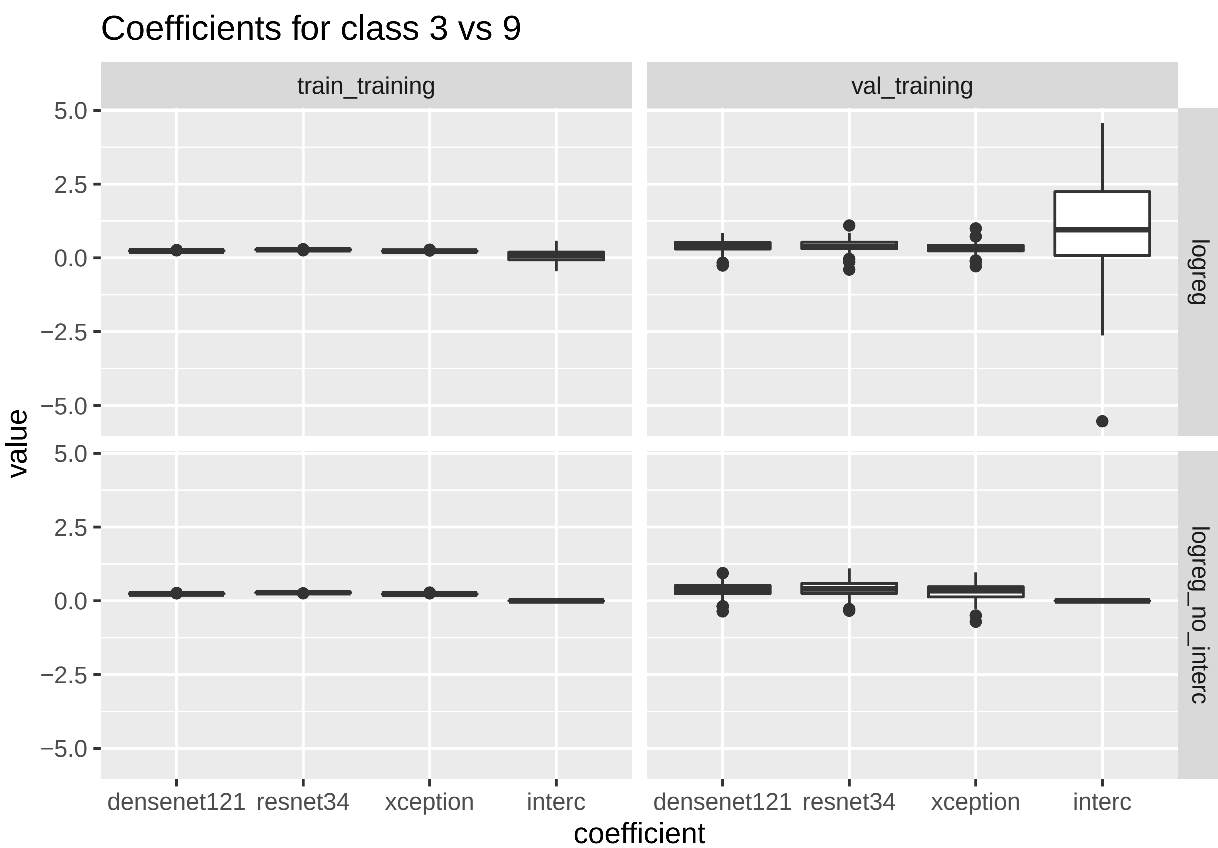

Coefficients for class 3 vs 5
train_training
val_training
logreg
logreg_no_interc
value
6
3
0
−3
−6
densenet121
resnet34
xception
interc
coefficient

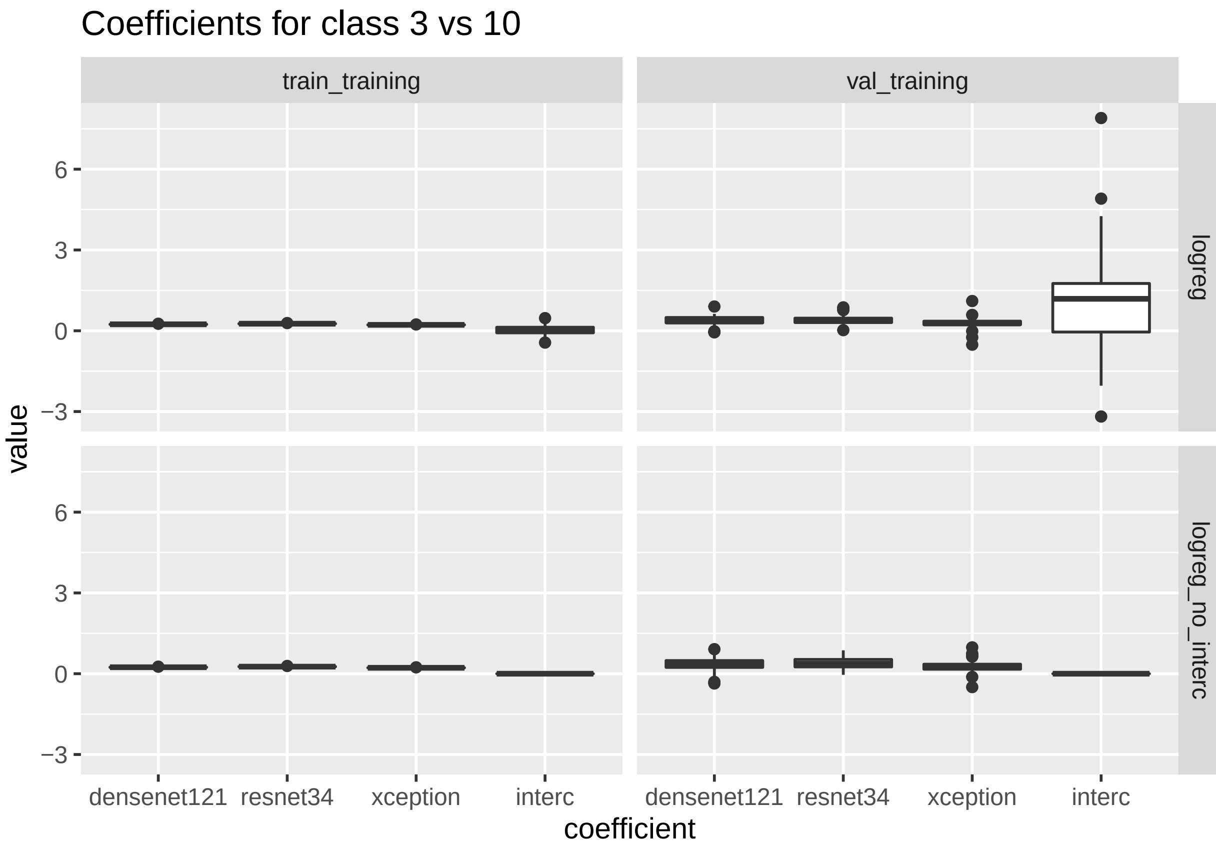

Coefficients for class 3 vs 6
train_training
val_training
logreg
logreg_no_interc
value
0
−5
0
−5
densenet121
resnet34
xception
interc
densenet121
resnet34
xception
interc
coefficient

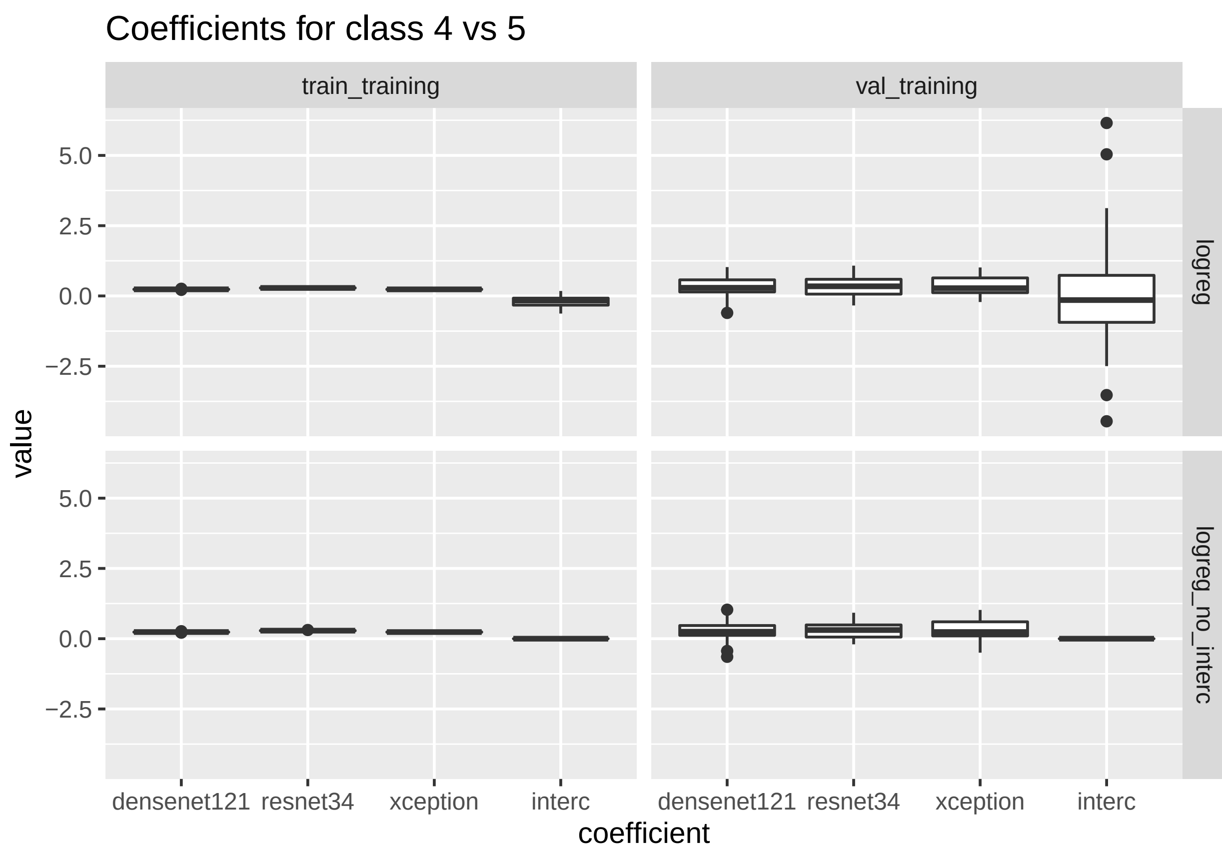

Coefficients for class 3 vs 7
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
densenet121
resnet34
xception
interc
coefficient

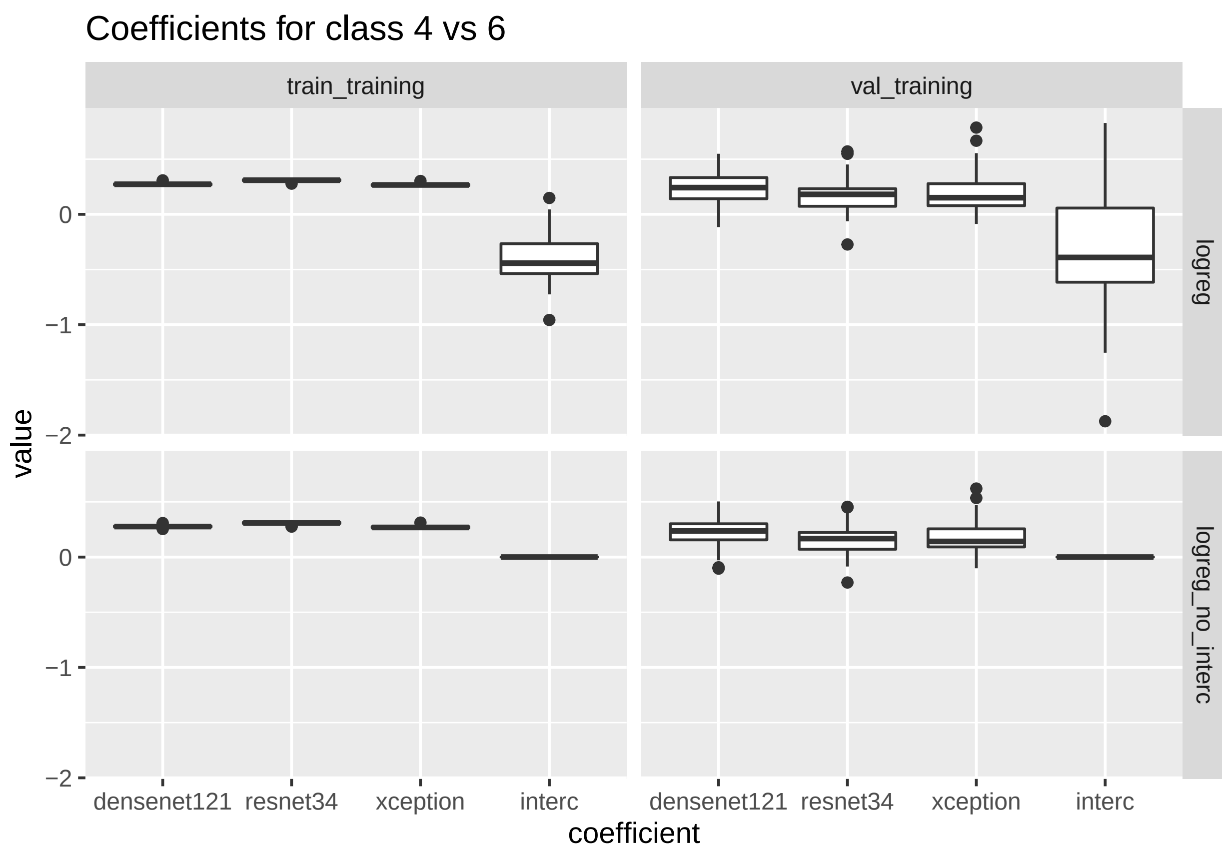


Coefficients for class 3 vs 8
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

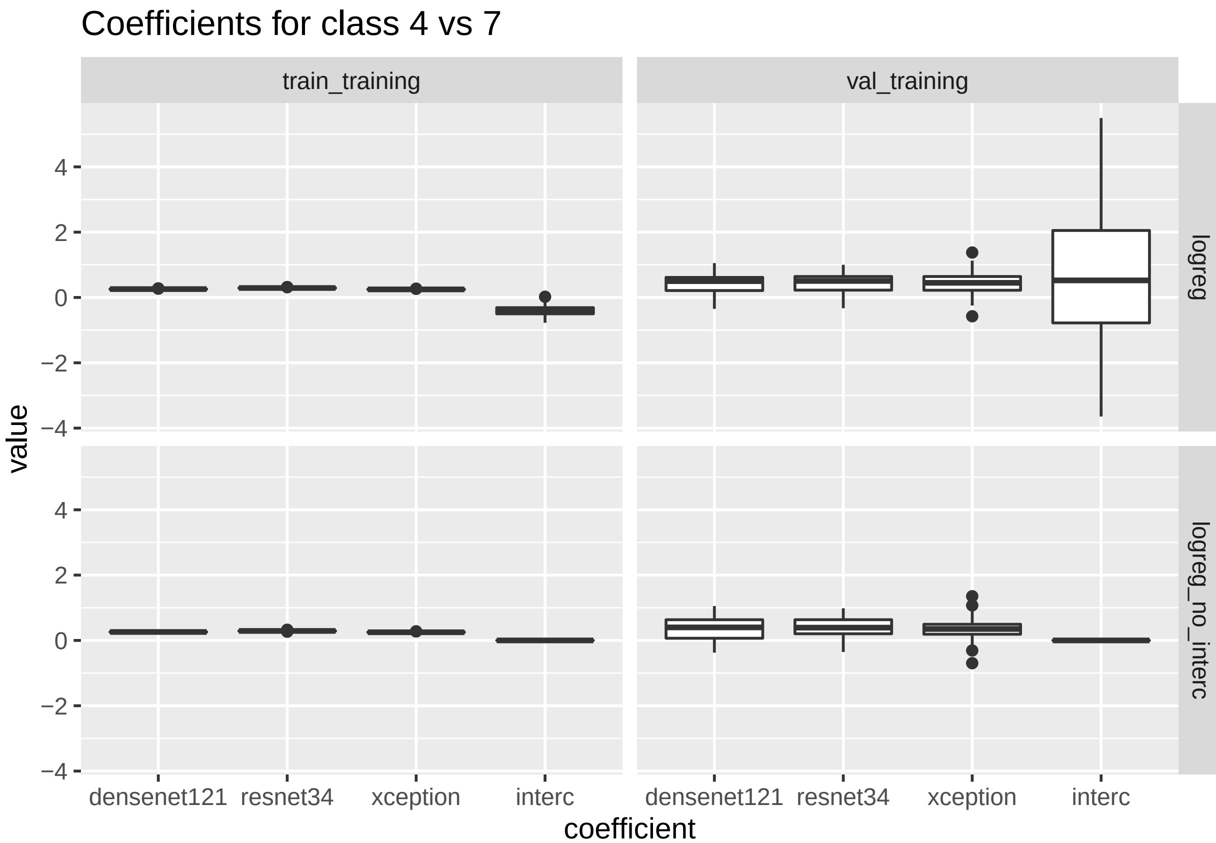

Coefficients for class 3 vs 9
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient

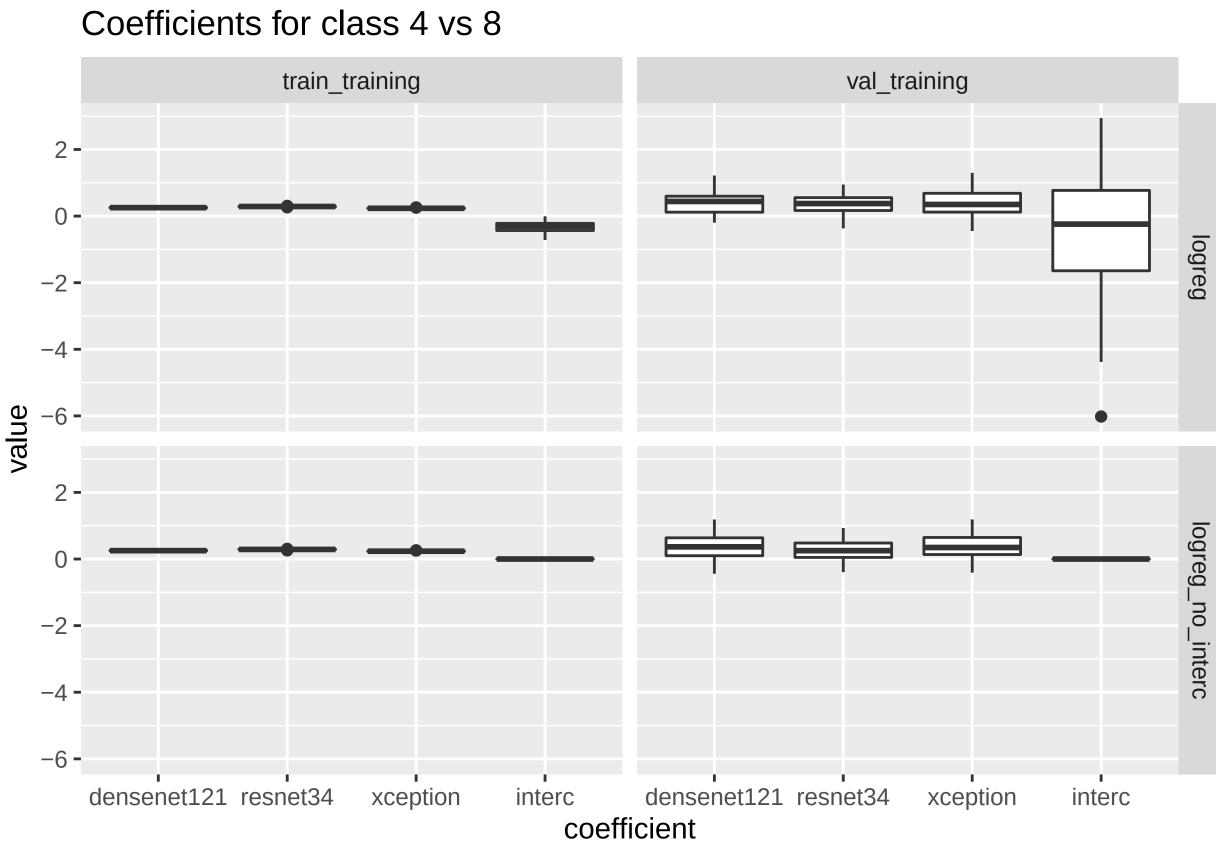

Coefficients for class 3 vs 10
train_training
val_training
logreg
logreg_no_interc
value
6
3
0
−3
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 4 vs 5
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

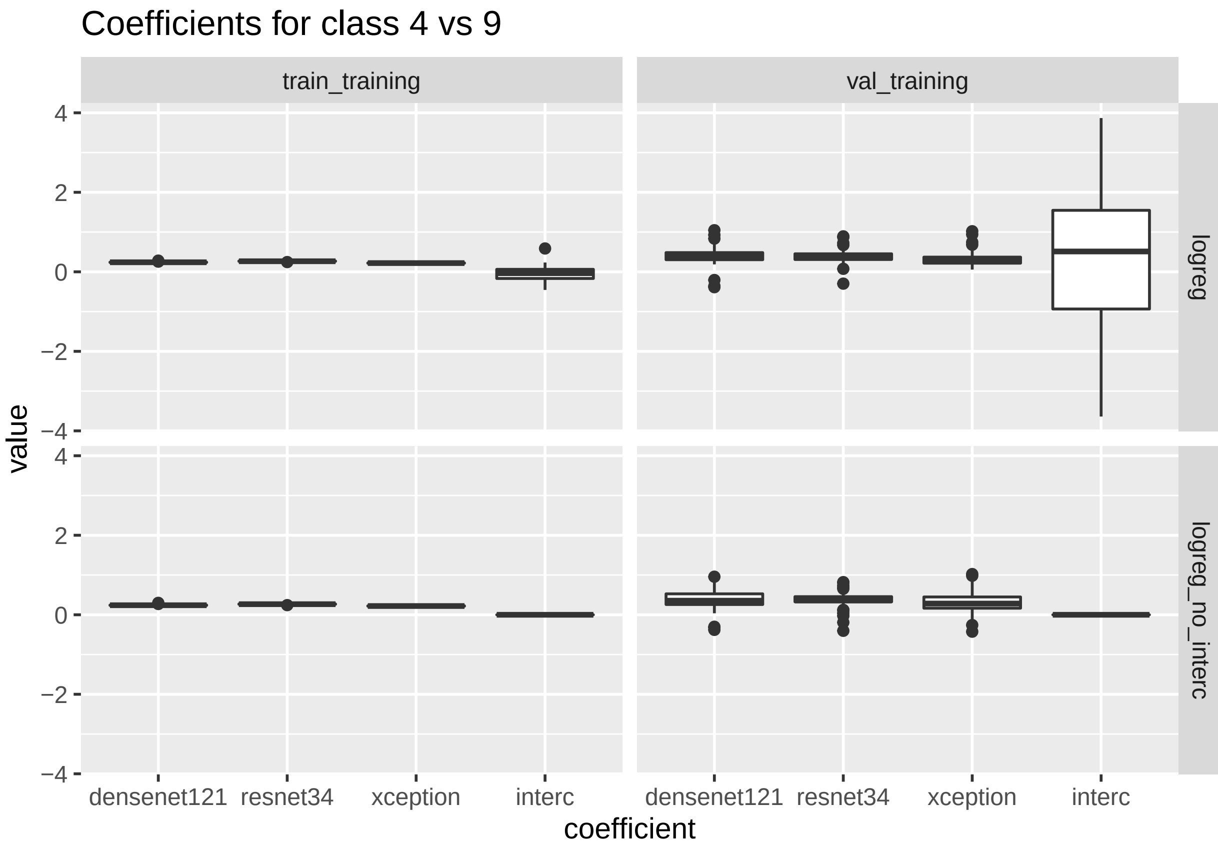


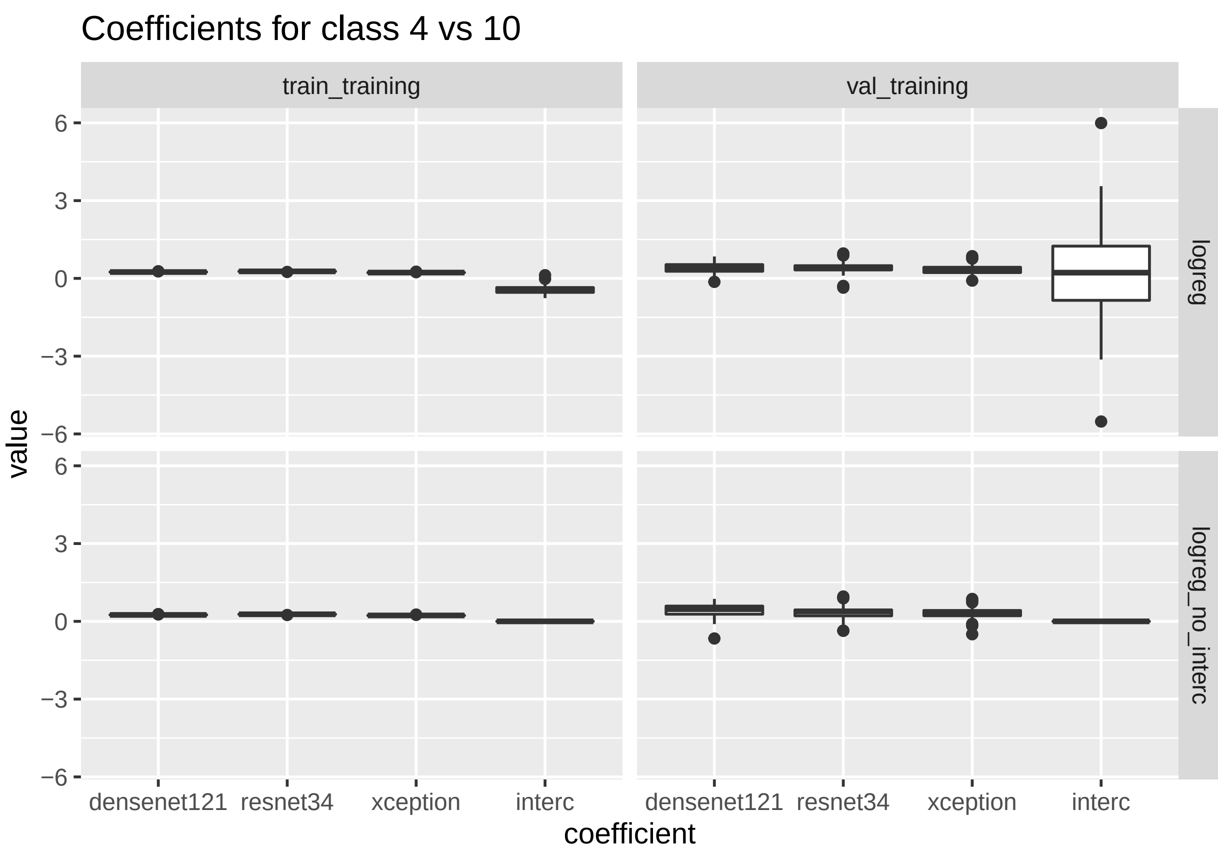
Coefficients for class 4 vs 6
train_training
val_training
logreg
logreg_no_interc
value
0
−1
−2
0
−1
−2
densenet121
resnet34
xception
interc
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 4 vs 7
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
−4
densenet121
resnet34
xception
interc
coefficient

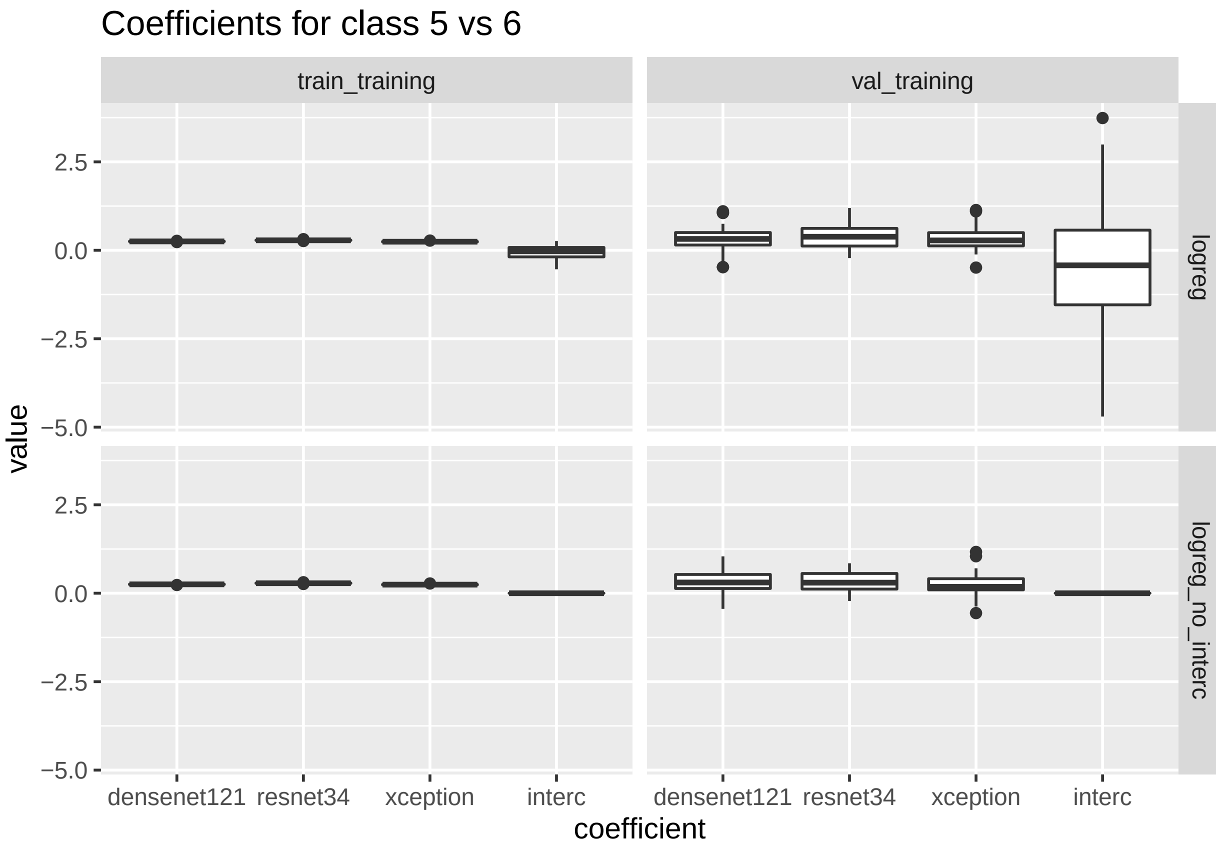


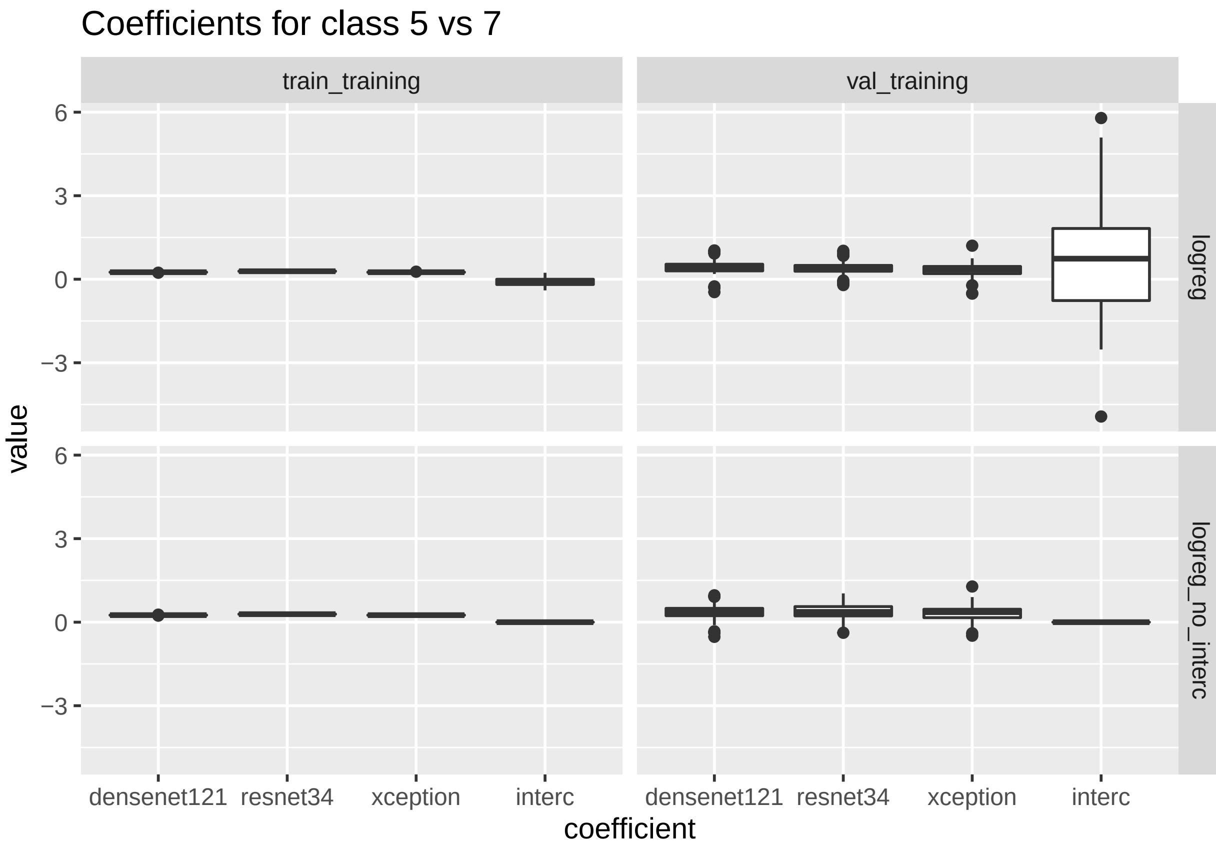
Coefficients for class 4 vs 8
train_training
val_training
logreg
logreg_no_interc
value
2
0
−2
−4
−6
densenet121
resnet34
xception
interc
coefficient

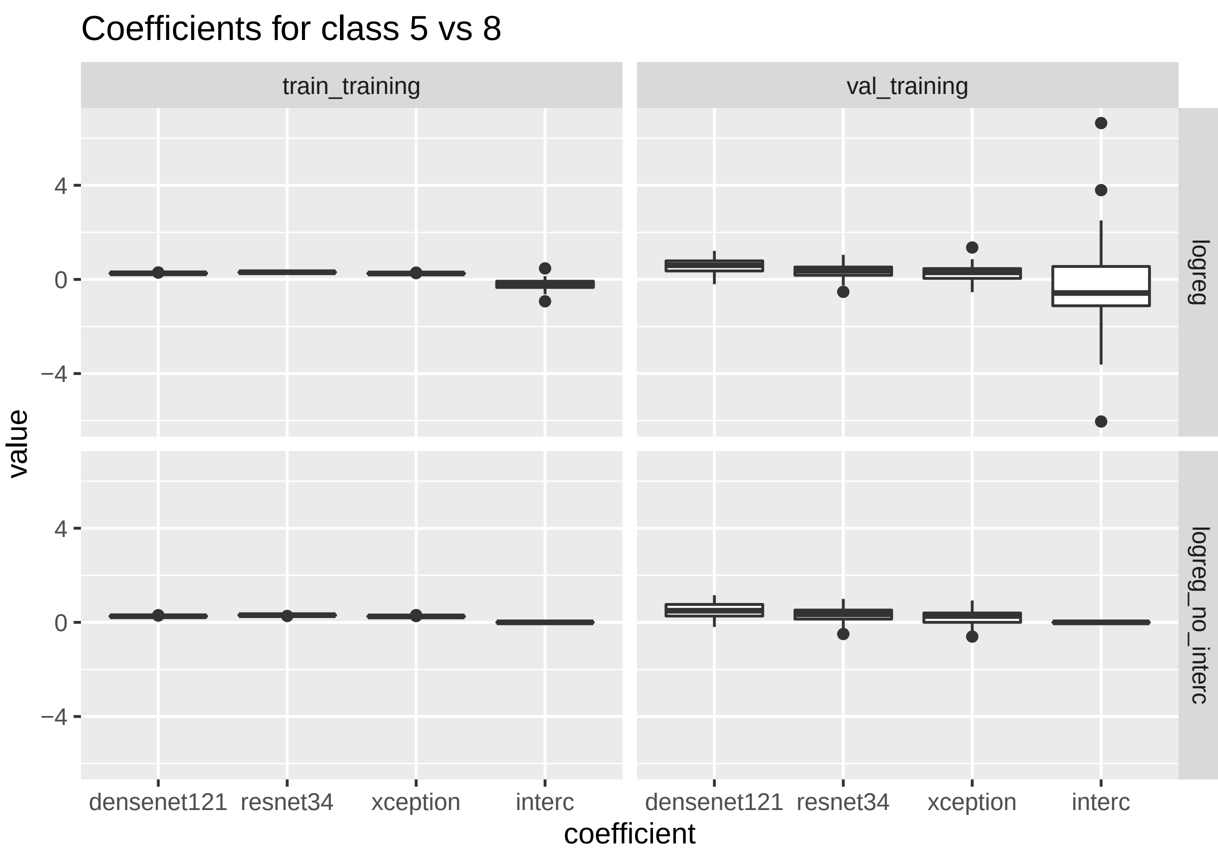

Coefficients for class 4 vs 9
train_training
val_training
logreg
logreg_no_interc
value
coefficient
densenet121
resnet34
xception
interc
4
2
0
−2
−4



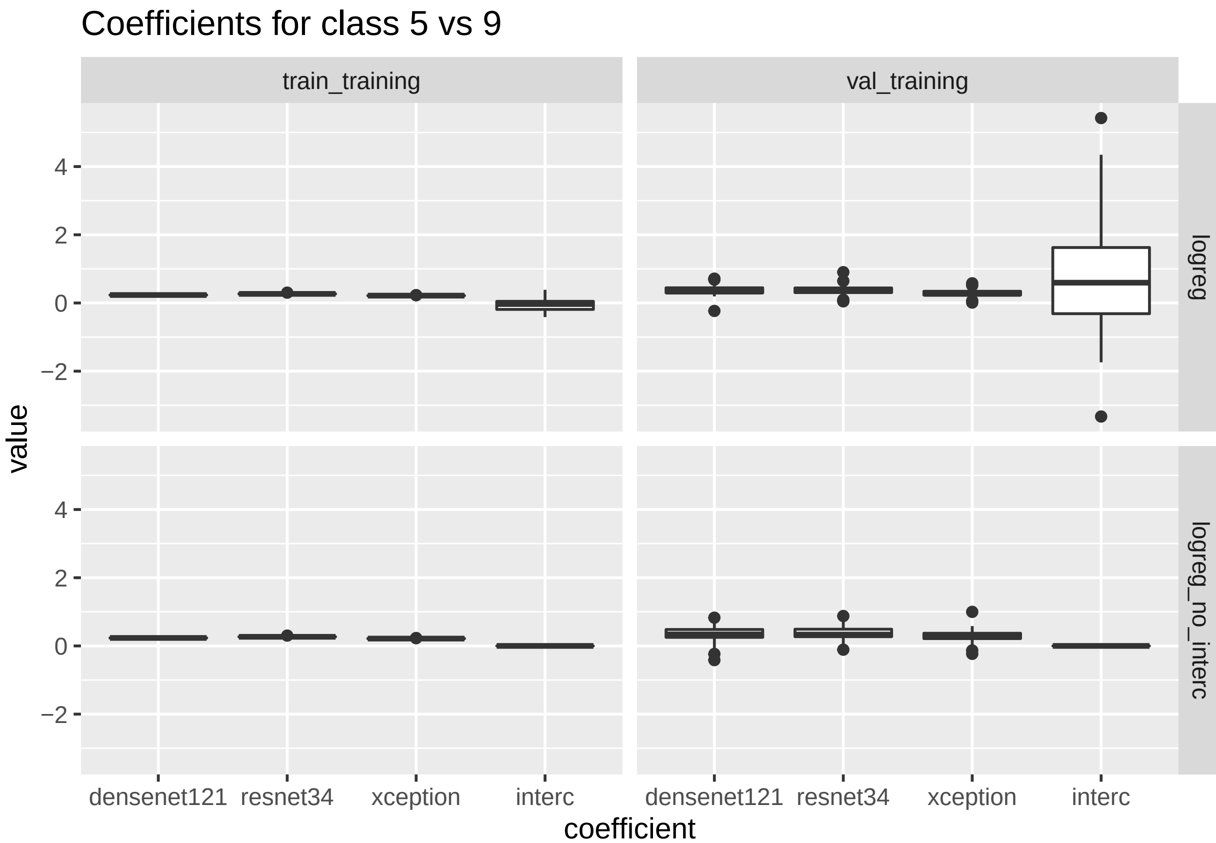
Coefficients for class 4 vs 10
train_training
val_training
logreg
logreg_no_interc
value
6
3
0
−3
−6
densenet121
resnet34
xception
interc
coefficient

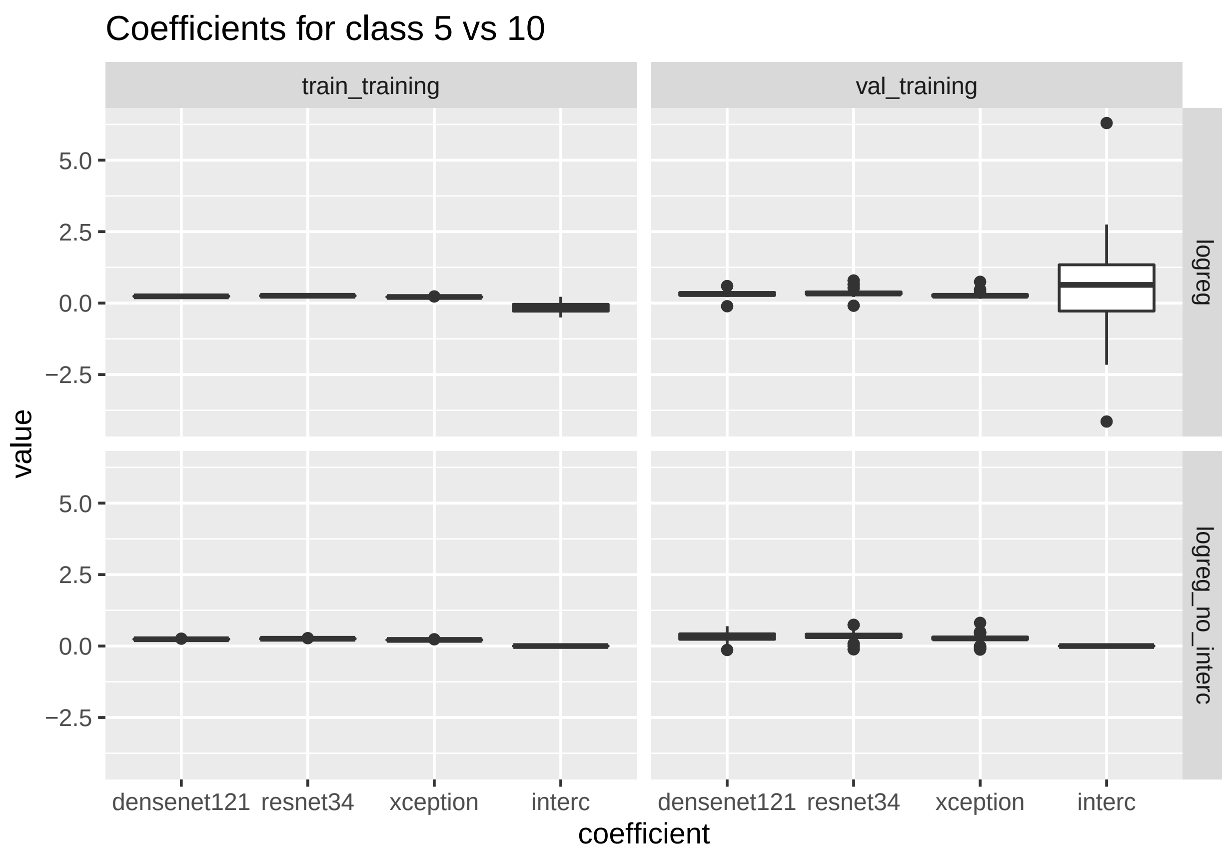

Coefficients for class 5 vs 6
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 5 vs 7
train_training
val_training
logreg
logreg_no_interc
value
6
3
0
−3
densenet121
resnet34
xception
interc
coefficient

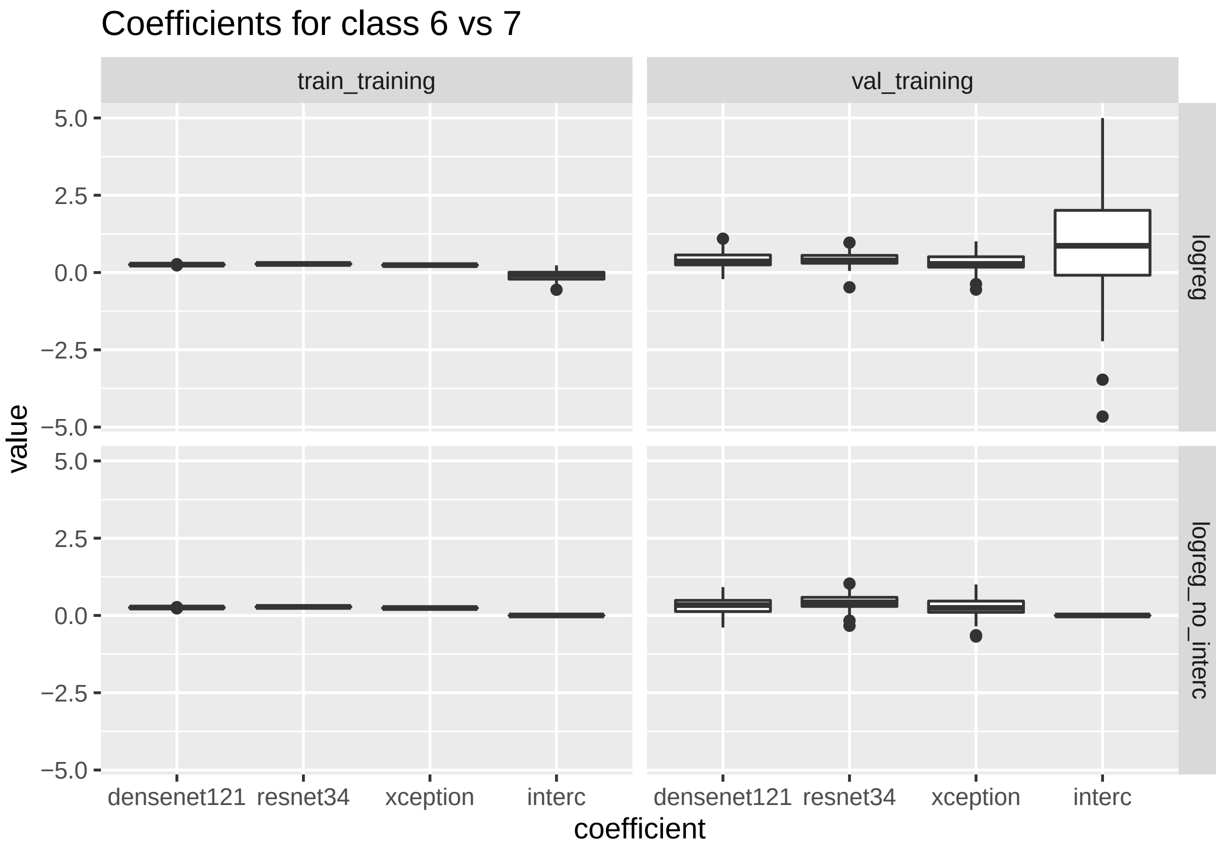


Coefficients for class 5 vs 8
train_training
val_training
logreg
logreg_no_interc
value
4
0
−4
4
0
−4
densenet121
resnet34
xception
interc
densenet121
resnet34
xception
interc
coefficient

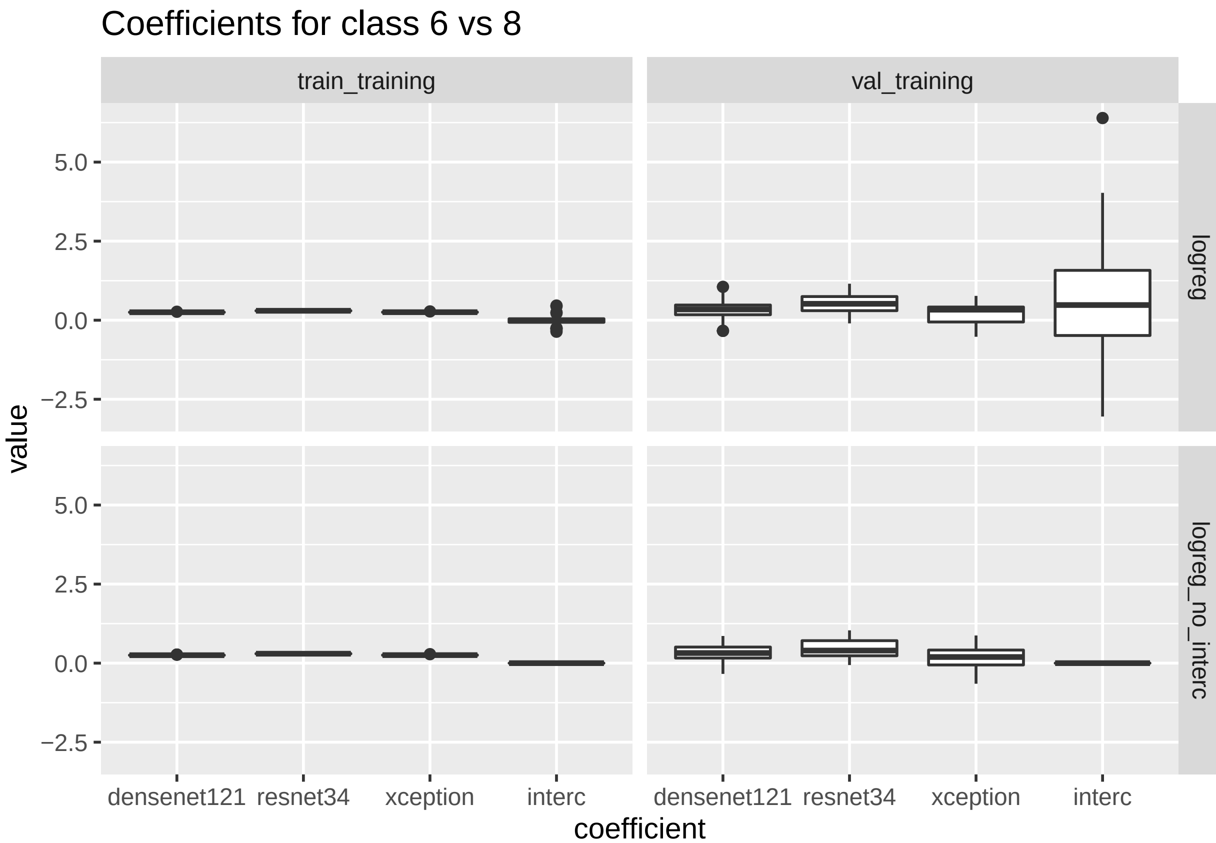

Coefficients for class 5 vs 9
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
densenet121
resnet34
xception
interc
coefficient

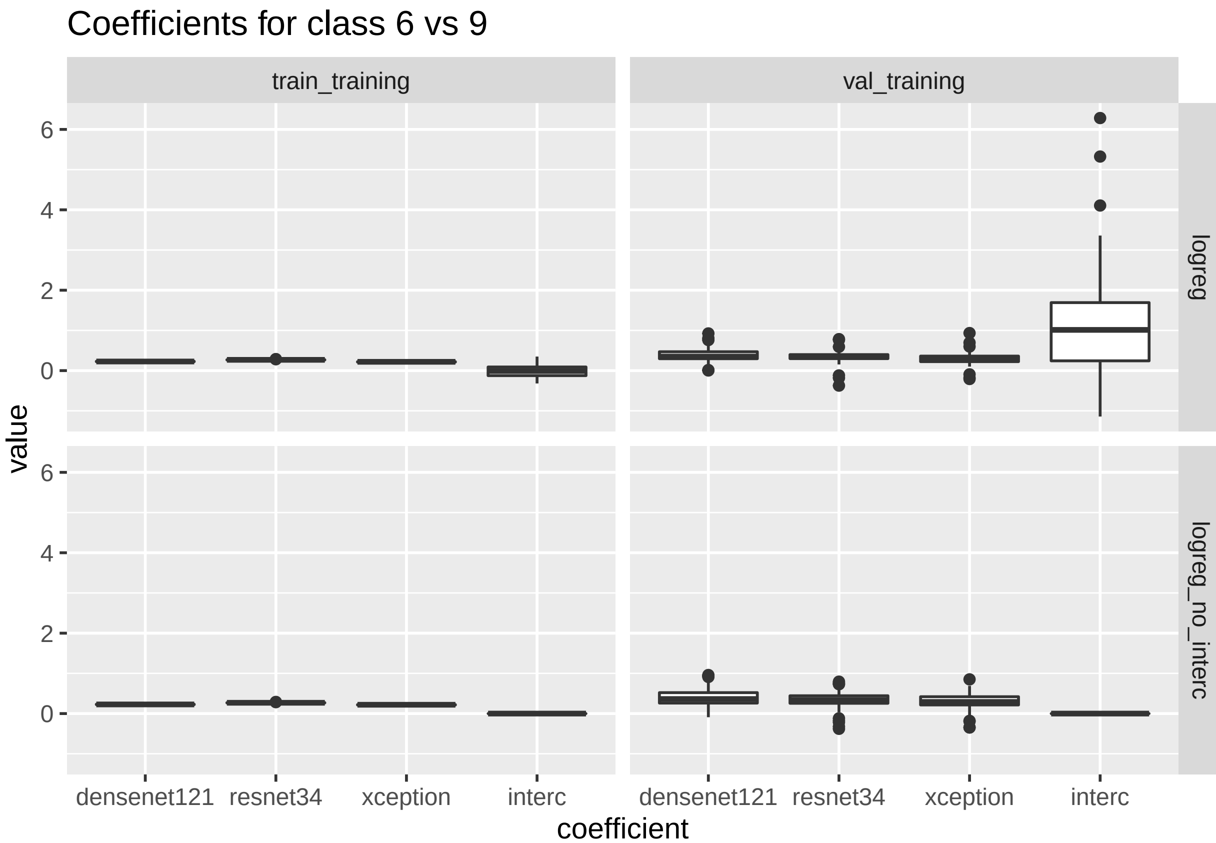


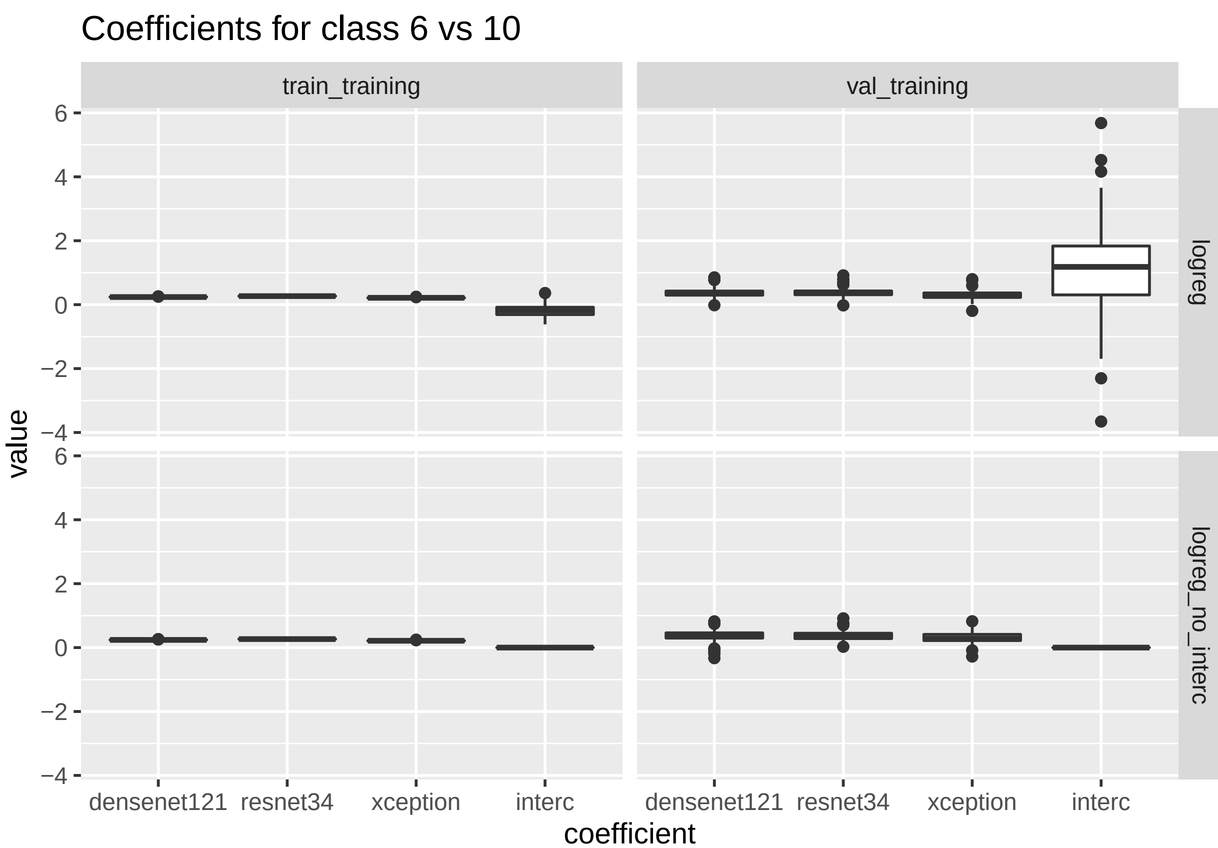
Coefficients for class 5 vs 10
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

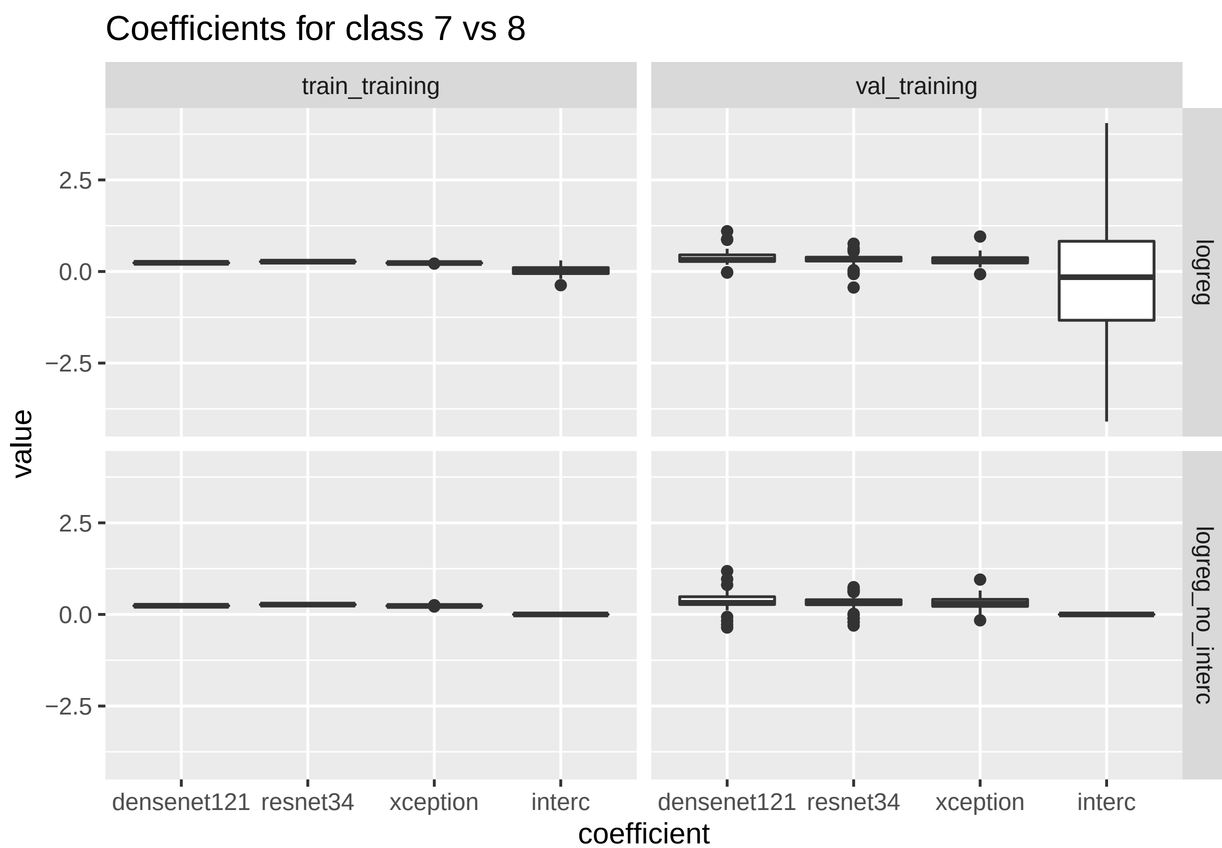


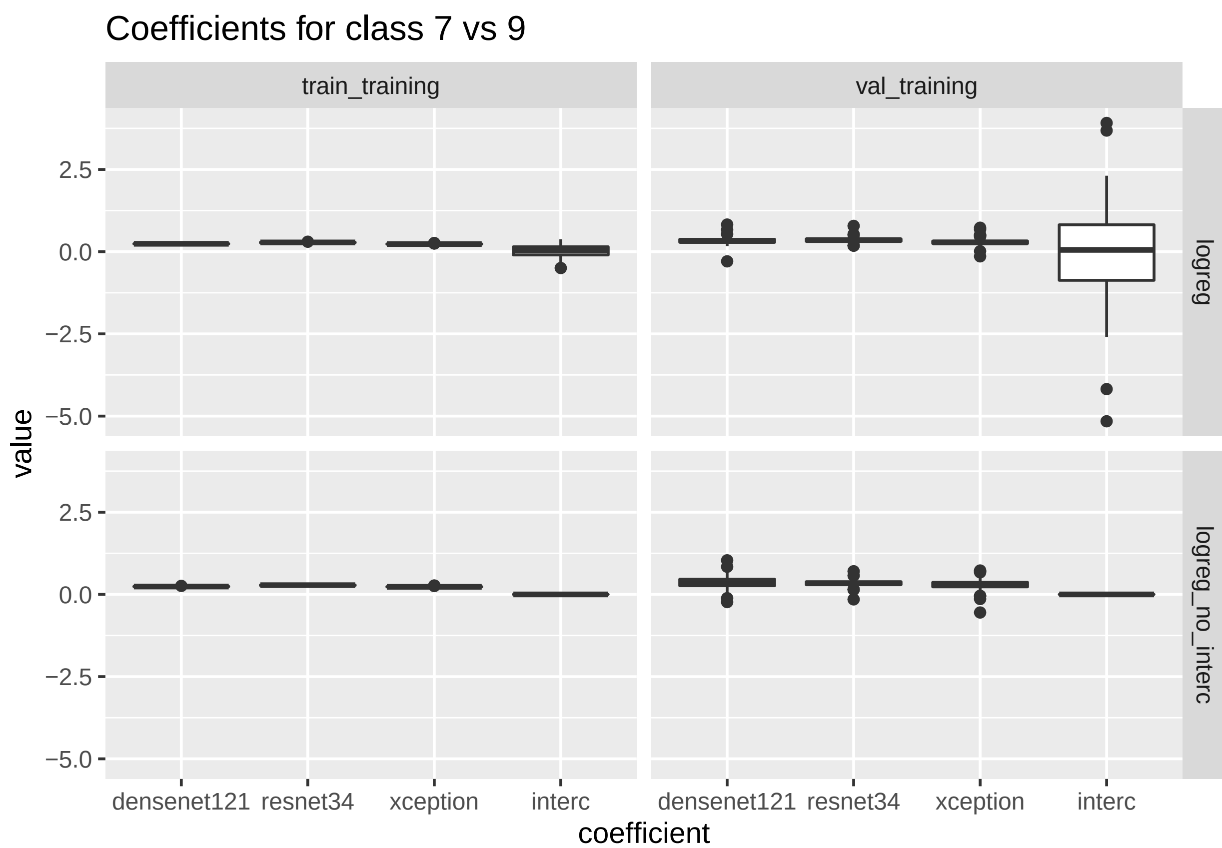
Coefficients for class 6 vs 7
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 6 vs 8
train_training
val_training
logreg
logreg_no_interc
value
5.0
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
coefficient

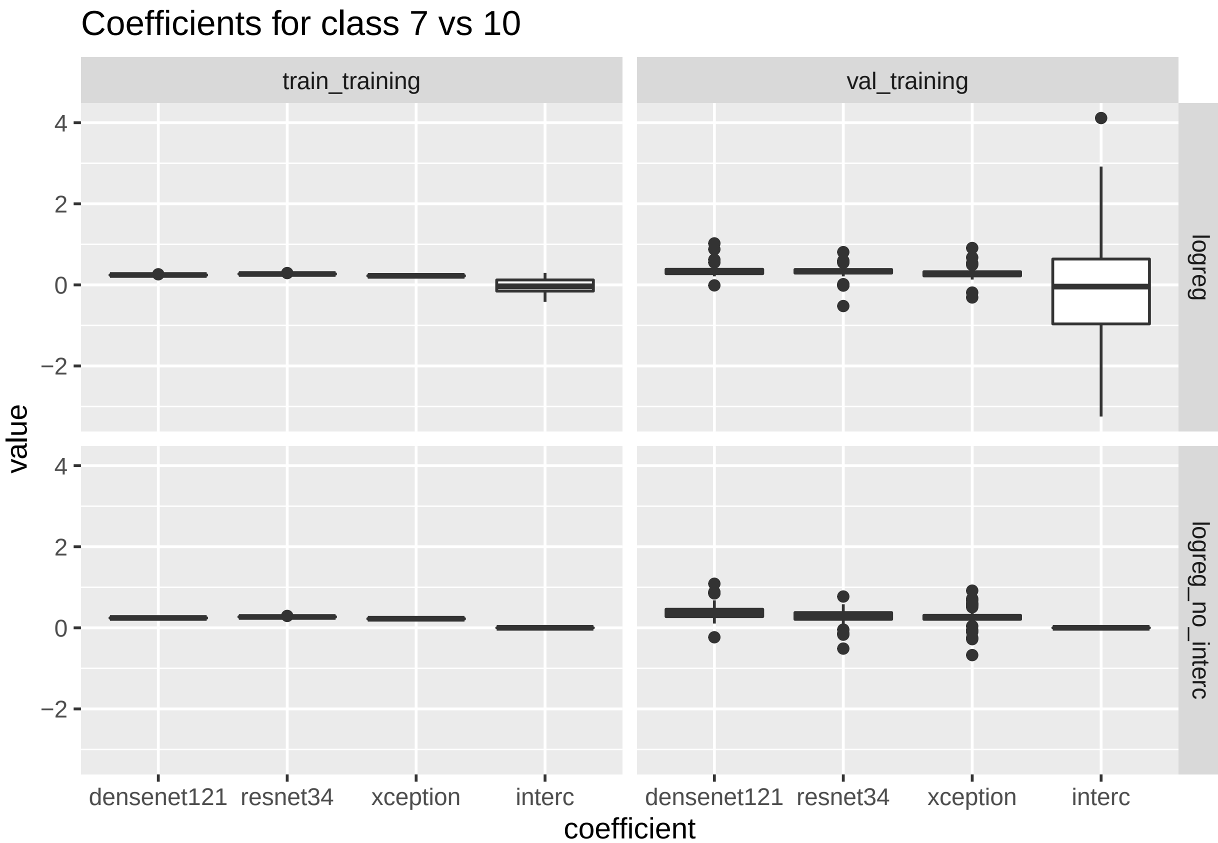

Coefficients for class 6 vs 9
train_training
val_training
logreg
logreg_no_interc
value
6
4
2
0
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 6 vs 10
train_training
val_training
logreg
logreg_no_interc
value
6
4
2
0
−2
−4
densenet121
resnet34
xception
interc
coefficient

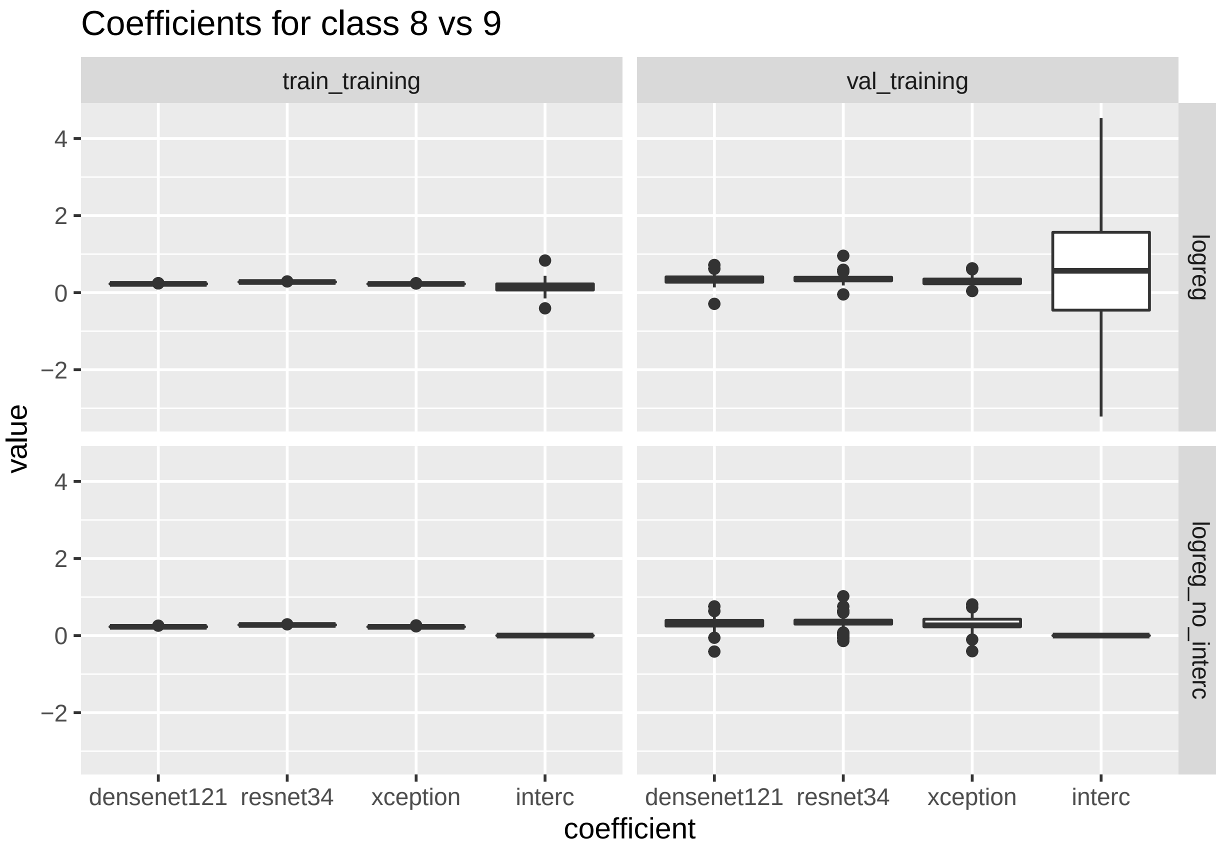

Coefficients for class 7 vs 8
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
2.5
0.0
−2.5
densenet121
resnet34
xception
interc
densenet121
resnet34
xception
interc
coefficient

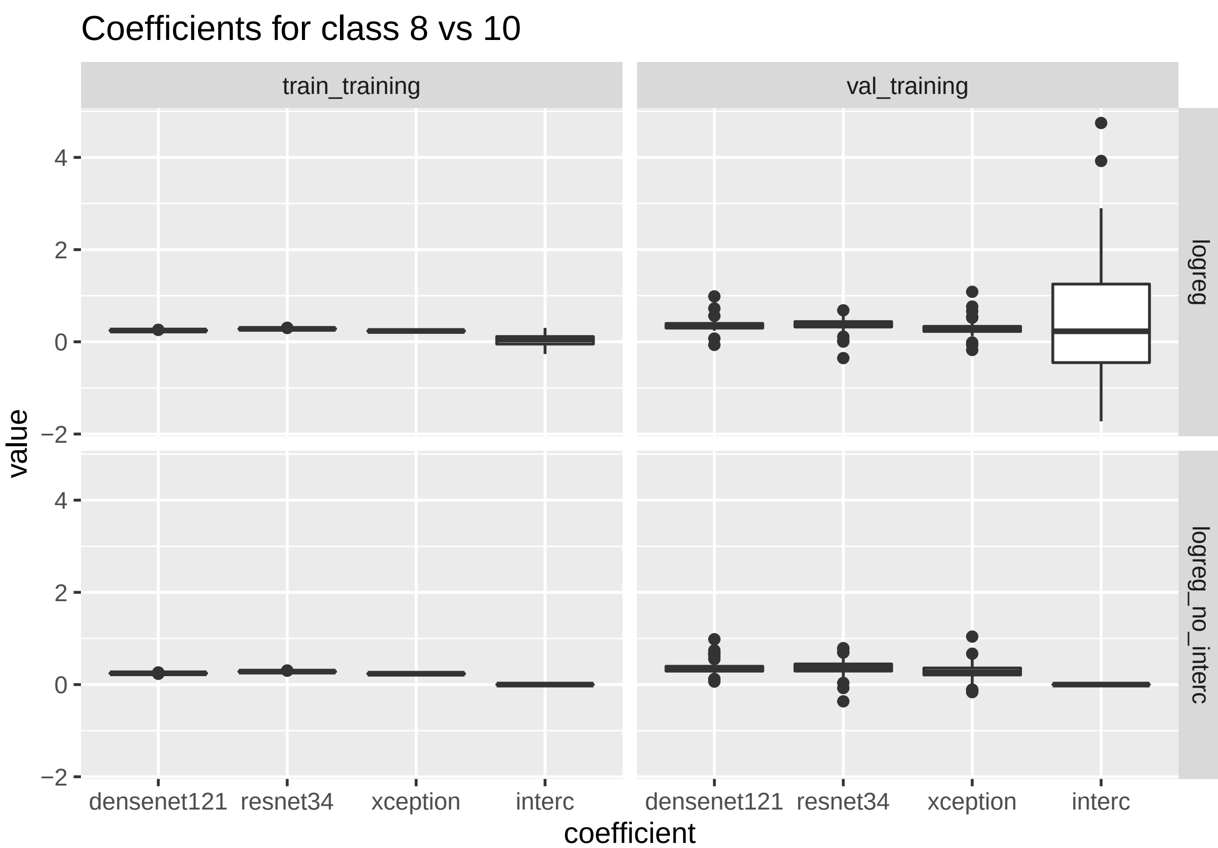


Coefficients for class 7 vs 9
train_training
val_training
logreg
logreg_no_interc
value
2.5
0.0
−2.5
−5.0
densenet121
resnet34
xception
interc
coefficient

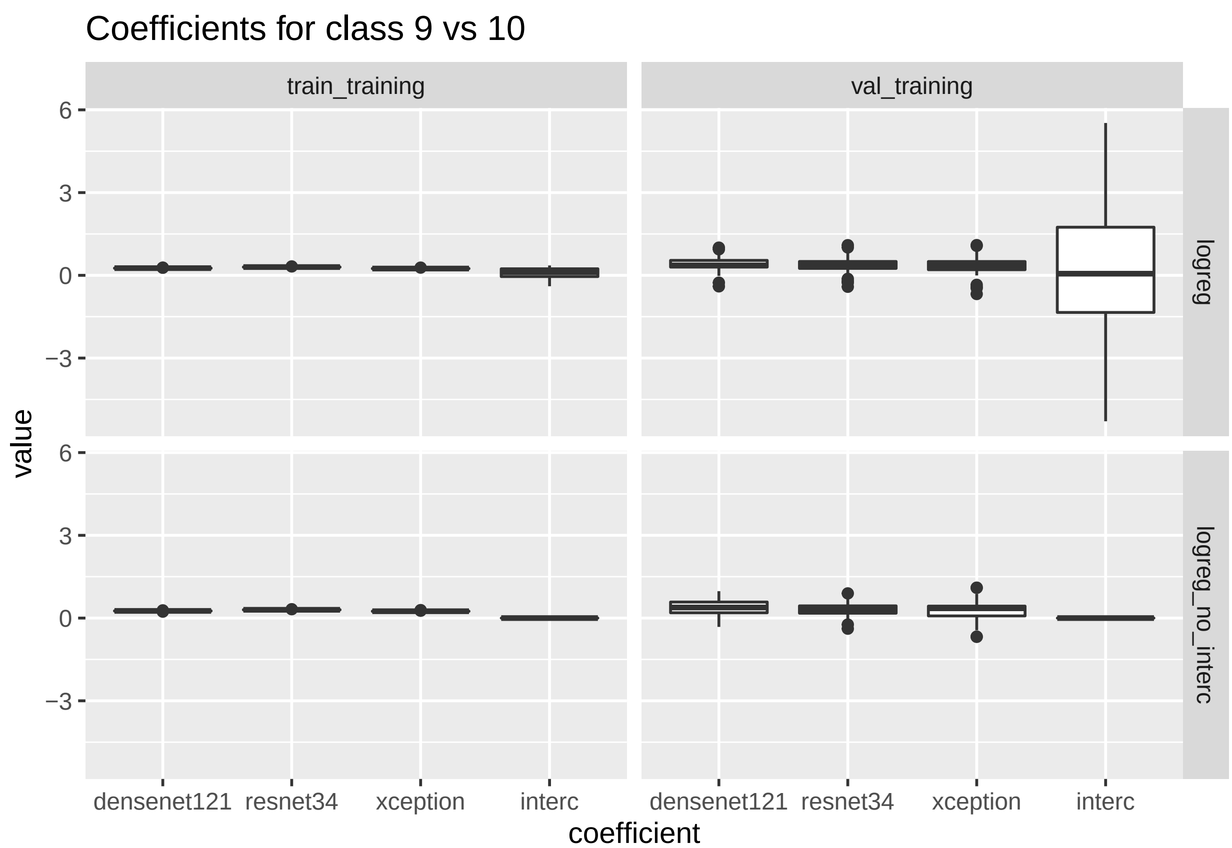


Coefficients for class 7 vs 10
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
densenet121
resnet34
xception
interc
coefficient


Coefficients for class 8 vs 9
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
densenet121
resnet34
xception
interc
coefficient



Coefficients for class 8 vs 10
train_training
val_training
logreg
logreg_no_interc
value
4
2
0
−2
densenet121
resnet34
xception
interc
coefficient



In case of lda train_training coefficients tend to have higher variance, in case of logreg it seem to be the oposite, val_training coefficients have higher variance.

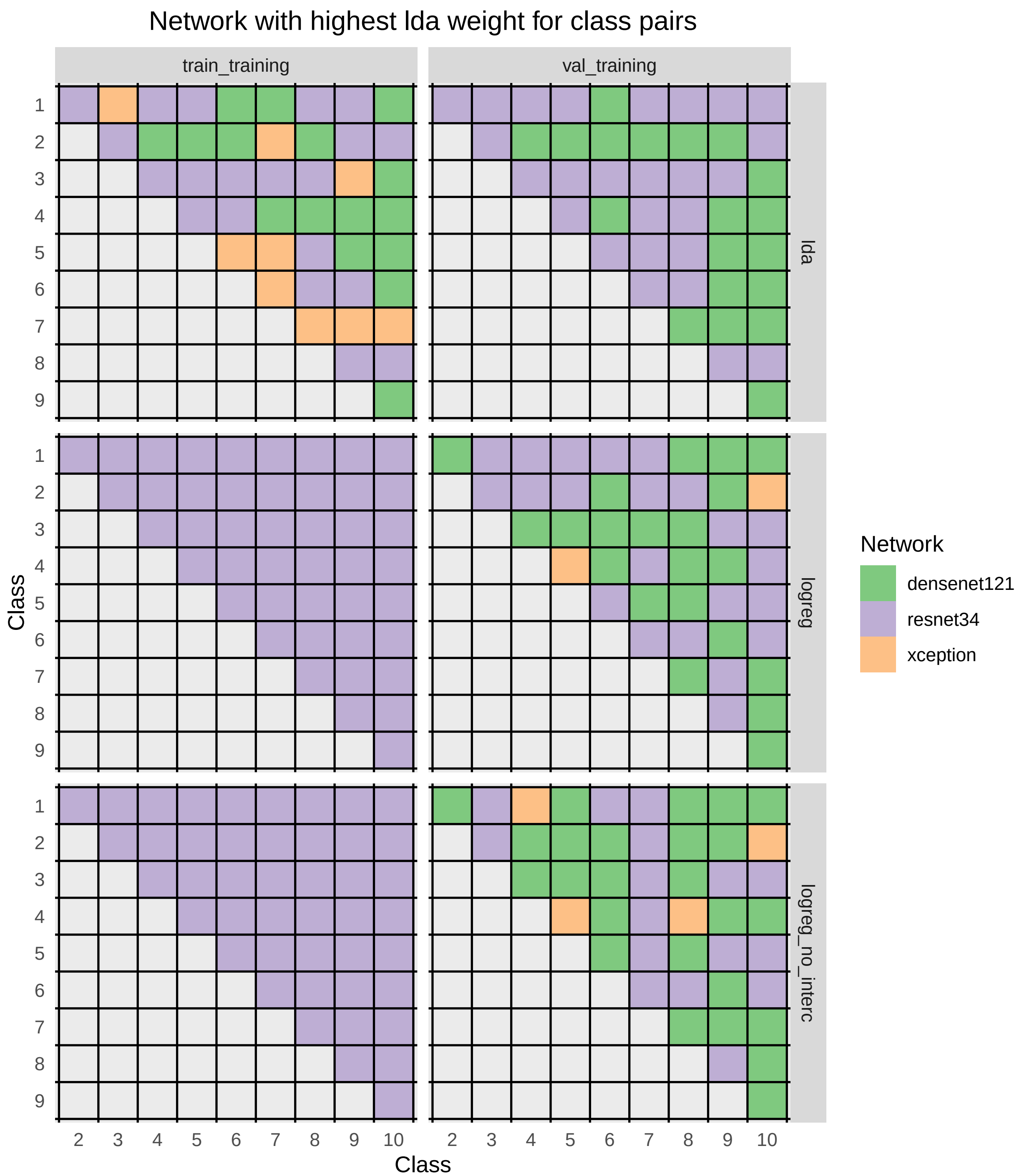
```
avg_combiner_coefs <- combiner_coefs %>% filter(coefficient != "interc") %>% group_by(class1, class2, pr
```

```
## `summarise()` has grouped output by 'class1', 'class2', 'precision', 'train_type', 'coefficient'. You
```

```
avg_combiner_c_w <- pivot_wider(avg_combiner_coefs, names_from = coefficient, values_from = value)
avg_combiner_c_w[, c("class1", "class2")] <- lapply(avg_combiner_c_w[, c("class1", "class2")], as.factor
avg_combiner_c_w$top_net <- factor(c("densenet121", "resnet34", "xception")[max.col(as.matrix(avg_combin
```

```
coefs_grid <- ggplot(avg_combiner_c_w, aes(x=class2, y=class1, fill=top_net)) +
  geom_raster() +
  scale_fill_brewer(type="qual") +
  facet_grid(cols=vars(train_type), rows=vars(combining_method)) +
  scale_y_discrete(limits=rev) +
  geom_vline(xintercept=seq(-0.5, 9.5, 1.0)) +
  geom_hline(yintercept=seq(-0.5, 9.5, 1.0)) +
  guides(fill=guide_legend(title="Network")) +
  xlab("Class") +
  ylab("Class") +
  ggtitle("Network with highest lda weight for class pairs") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

```
coefs_grid
```



Densenet is far less dominating in this experiment than in visualizations_ensemble_outputs_CIF10. Other networks seem to be more competitive when training is done just on half of CIFAR 10 training set. In case of logreg and train_training, however resnet has highest coefficient for all class pairs.