

## Outputs inspection half CIFAR100

```
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)
```

```
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 'reticulate' was built under R version 4.0.5
```

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

```
##
```

```
## Attaching package: 'berryFunctions'
```

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

```
##
```

```
## between
```

Visualization on CIFAR100. We are using data of three neural networks trained on reduced CIFAR100 training set. Half of the CIFAR100 training set was extracted as a validation set. We then divided both the reduced training set and validation set into 5 disjoint subsets and trained an ensemble on each of them. This was done in 10 replications, each time with random split of the training set into validation and new training set. In this visualization, we are trying to inspect the outputs deeper, mainly to make sense of strange behavior of nll metric for ensemble outputs.

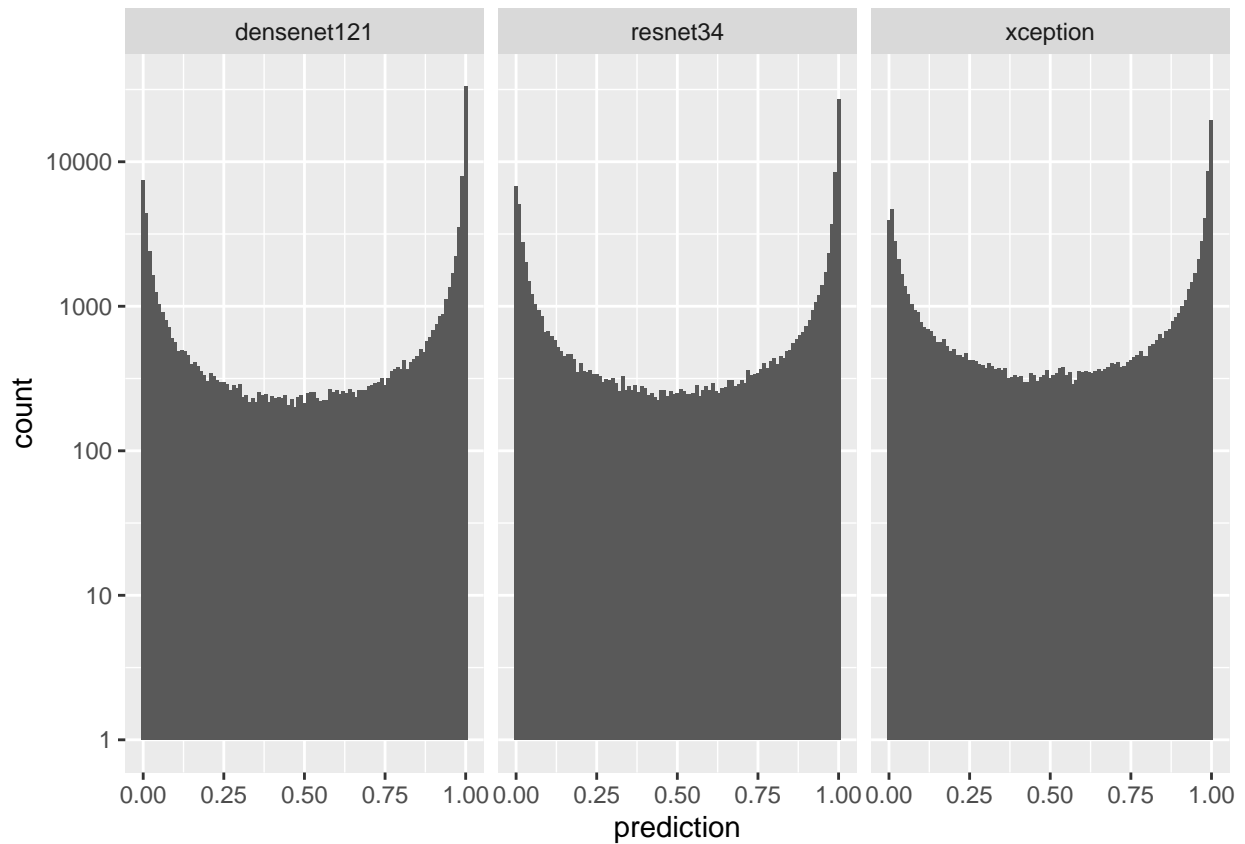
```
base_dir <- "../data/data_train_val_half_c100"
repls <- 0:9
folds <- 0:4

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"))
```

```
classes <- dim(nets_outputs[["test_outputs"]])[4]
nets_num <- length(nets_outputs[["networks"]])
preds <- nets_outputs[["test_outputs"]]
labs <- nets_outputs[["test_labels"]]
network <- c()
prediction <- c()
replication <- c()
for (ri in repls + 1)
{
  for (net_i in seq_along(nets_outputs[["networks"]]))
  {
    net_outputs_prob <- softmax(preds[ri, net_i, , ])
    cur_pred <- c()
    for (class in 1:classes)
    {
      class_probs <- net_outputs_prob[labs[ri, ] == (class - 1), class]
      cur_pred <- c(cur_pred, class_probs)
    }
    network <- c(network, rep(nets_outputs[["networks"]][net_i], length(cur_pred)))
    prediction <- c(prediction, cur_pred)
  }
  replication <- c(replication, rep(ri, length(prediction) - length(replication)))
}

networks_predictions <- data.frame(network, replication, prediction)
```

```
nets_cor_preds_histo <- ggplot(data=networks_predictions) + geom_histogram(mapping=aes(x=prediction), b
nets_cor_preds_histo
```



```
classes <- dim(nets_outputs[["test_outputs"]])[4]
labs <- nets_outputs[["test_labels"]]
method <- c()
prediction <- c()
replication <- c()
fold <- c()
for (ri in repls + 1)
{
  for (m_i in seq_along(ens_outputs$methods))
  {
    for (f_i in folds + 1)
    {
      cur_pred <- c()
      for (class in 1:classes)
      {
        class_probs <- ens_outputs$val_training[ri, m_i, f_i, labs[ri, ] == (class - 1), class]
        cur_pred <- c(cur_pred, class_probs)
      }
      method <- c(method, rep(ens_outputs$methods[m_i], length(cur_pred)))
      prediction <- c(prediction, cur_pred)
      fold <- c(fold, rep(f_i - 1, length(cur_pred)))
    }
  }
}
```

```

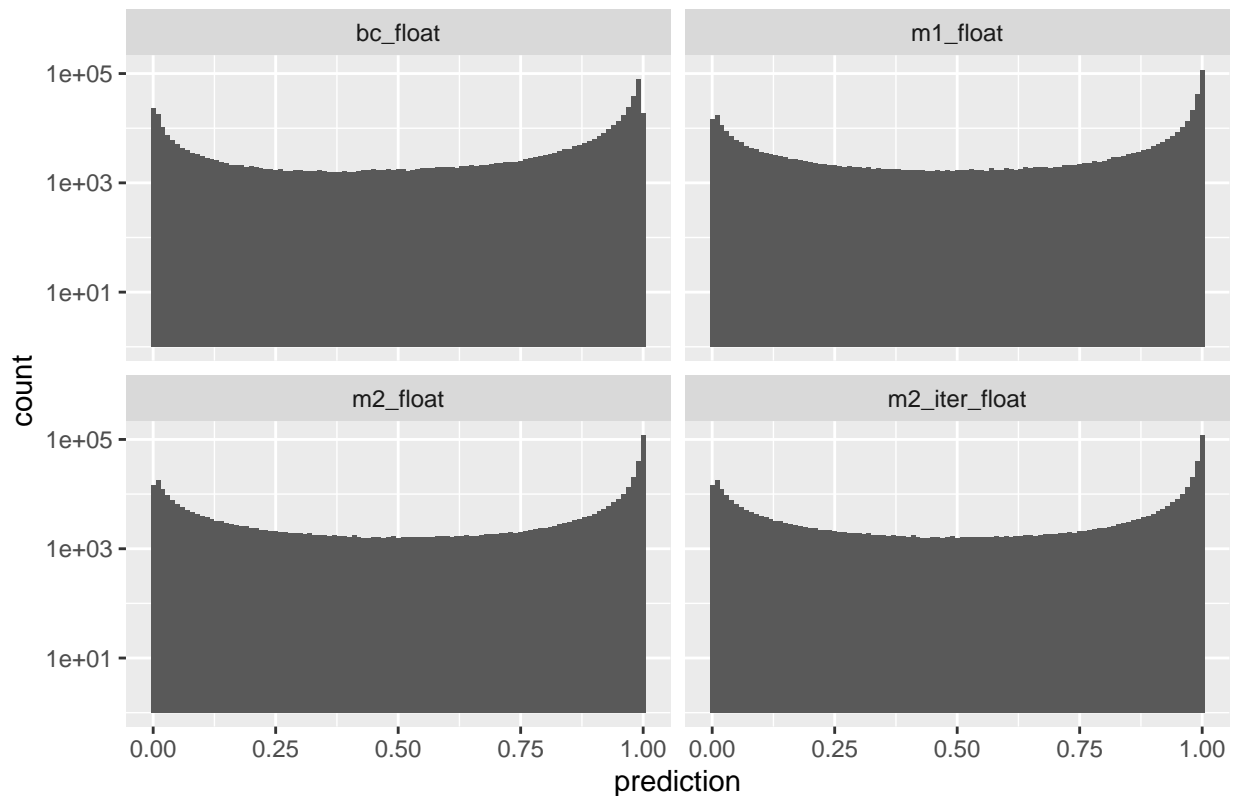
}
replication <- c(replication, rep(ri - 1, length(cur_pred) - length(replication)))
}

val_ens_predictions <- data.frame(method, replication, fold, prediction)

val_ens_cor_preds_histo <- ggplot(data=val_ens_predictions) + geom_histogram(mapping=aes(x=prediction),
val_ens_cor_preds_histo

```

Probabilities predicted for the correct class – ens trained on val



```

val_ens_zero_counts <- ggplot(data=val_ens_predictions[val_ens_predictions$prediction == 0, ]) + geom_h

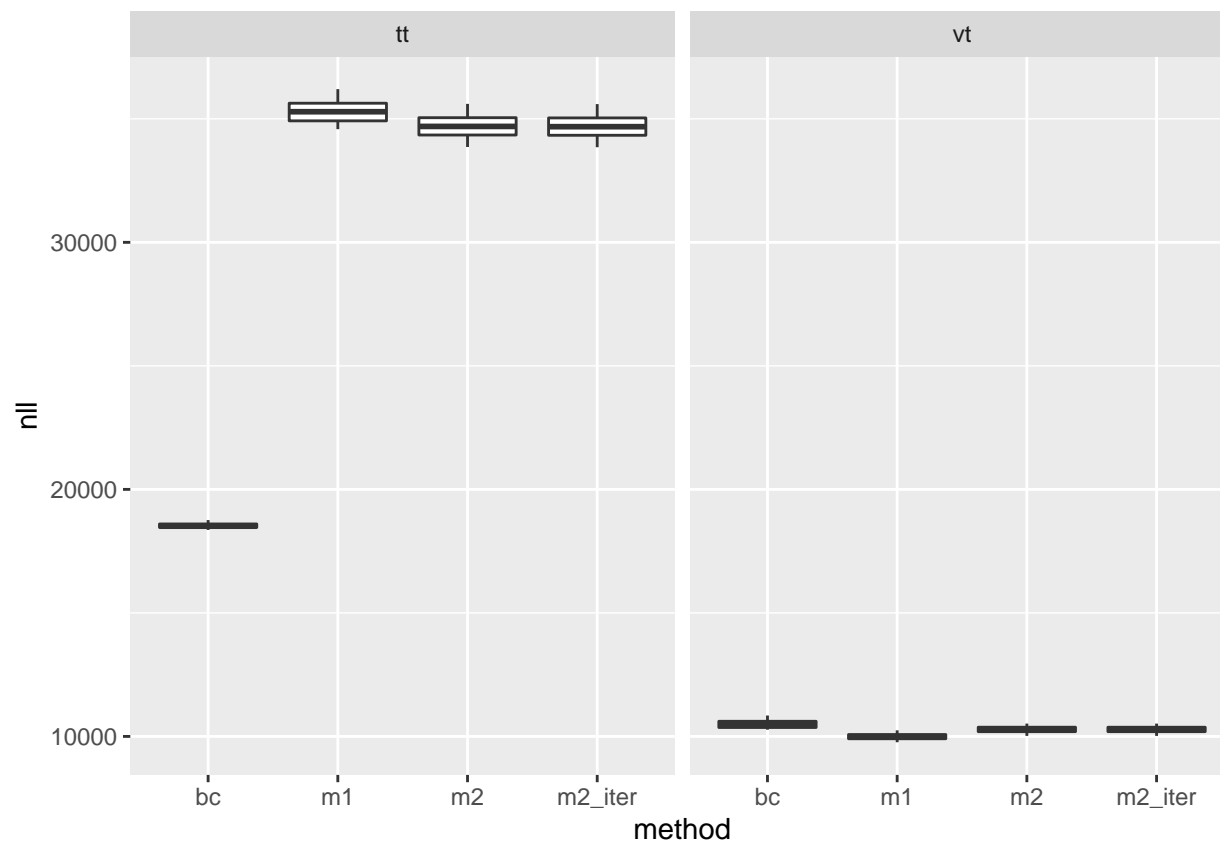
## Warning: Ignoring unknown parameters: binwidth, bins, pad

val_ens_zero_counts

```



```
val_ens_nll <- ggplot(data=ens_results) + geom_boxplot(mapping=aes(x=method, y=nll)) + facet_wrap(~train  
val_ens_nll
```



```

classes <- dim(nets_outputs[["test_outputs"]])[4]
labs <- nets_outputs[["test_labels"]]
method <- c()
prediction <- c()
replication <- c()
fold <- c()
for (ri in repls + 1)
{
  for (m_i in seq_along(ens_outputs$methods))
  {
    for (f_i in folds + 1)
    {
      cur_pred <- c()
      for (class in 1:classes)
      {
        class_probs <- ens_outputs$train_training[ri, m_i, f_i, labs[ri, ] == (class - 1), class]
        cur_pred <- c(cur_pred, class_probs)
      }
      method <- c(method, rep(ens_outputs$methods[m_i], length(cur_pred)))
      prediction <- c(prediction, cur_pred)
      fold <- c(fold, rep(f_i - 1, length(cur_pred)))
    }
  }
  replication <- c(replication, rep(ri - 1, length(cur_pred) - length(replication)))
}

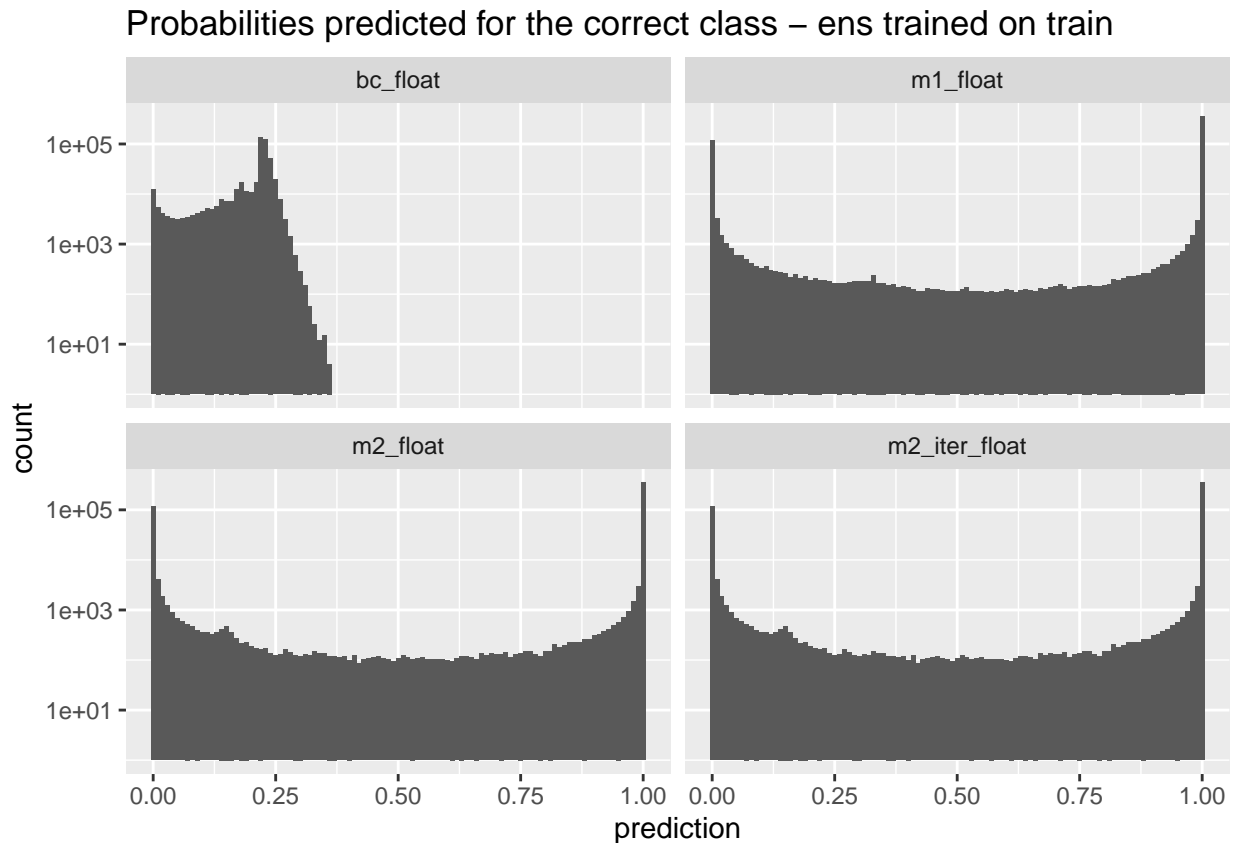
```

```
train_ens_predictions <- data.frame(method, replication, fold, prediction)
```

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

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

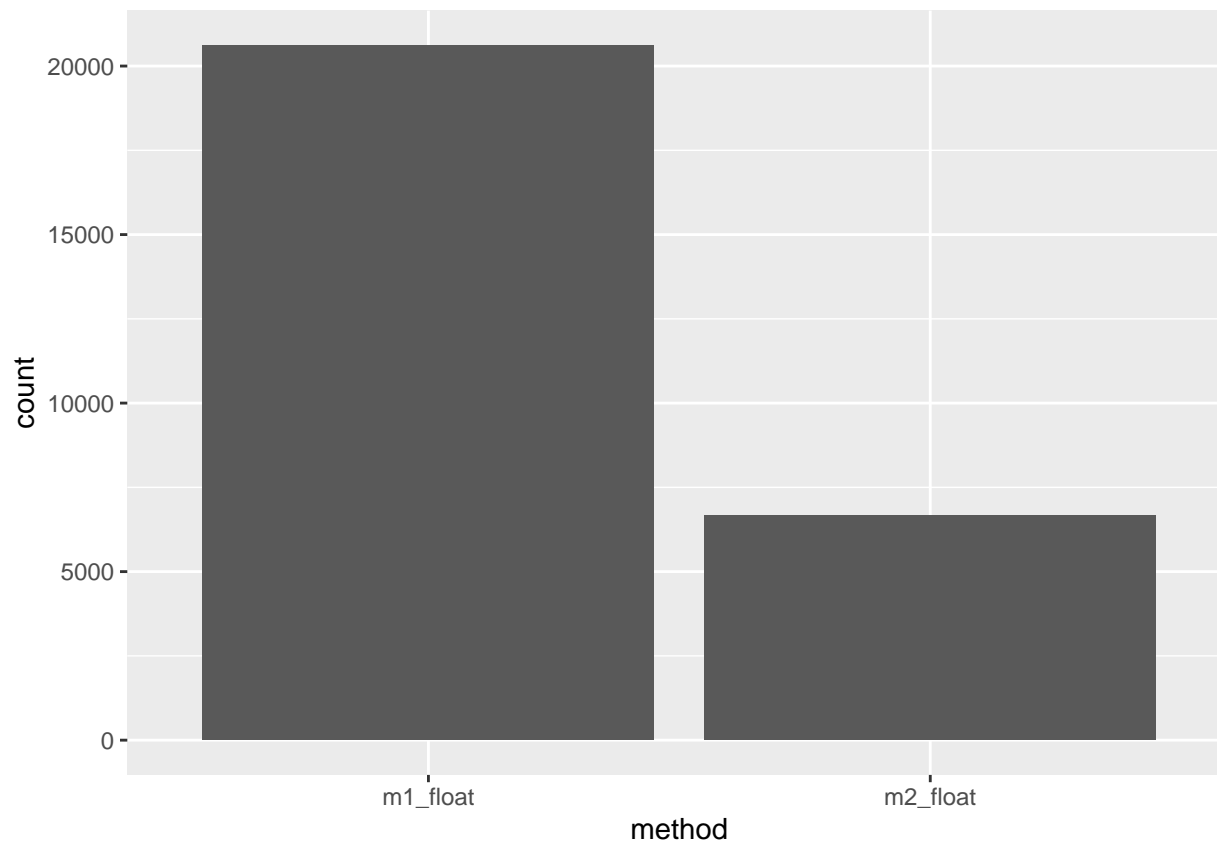
```
## Warning: Removed 64 rows containing missing values (geom_bar).
```



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

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

```
train_ens_zero_counts
```

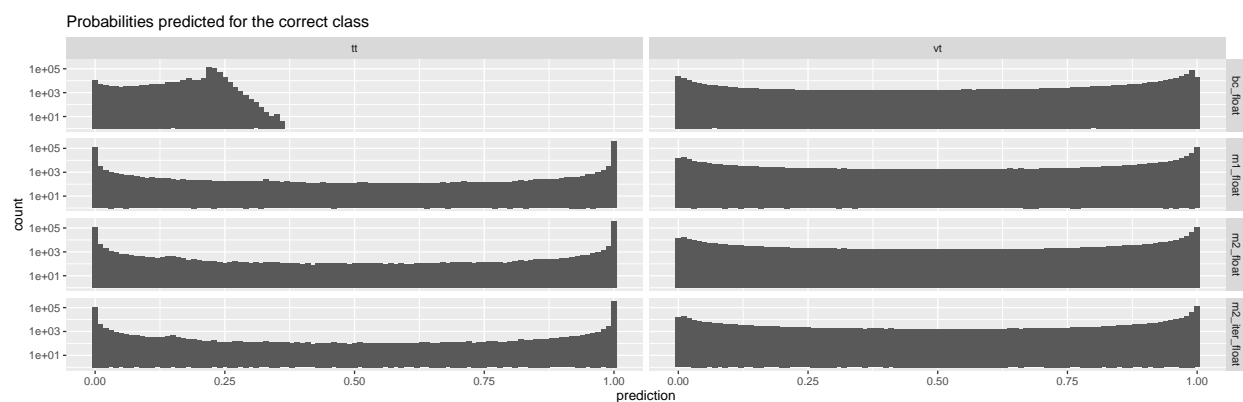


```
val_ens_predictions$train_type <- "vt"
train_ens_predictions$train_type <- "tt"
ens_predictions <- rbind(val_ens_predictions, train_ens_predictions)
```

```
ens_cor_preds_histo <- ggplot(data=ens_predictions) + geom_histogram(mapping=aes(x=prediction), binwidth=0.05)
ens_cor_preds_histo
```

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 64 rows containing missing values (geom\_bar).



Very strange behavior for the bc method trained on training set. Needs further attention.