Classification

K Arnold

Objectives

- Apply the `tidymodels` pipeline to a classification task
- Identify when validation is necessary to believe an outcome
- Apply the concepts of sensitivity and specificity

(punted to next time:)

- Identify corresponding elements between R and Python data wrangling and modeling workflows

Outline

- Problem introduction
  - Data wrangling in R
- Classification workflow
  - Models: decision tree, logistic regression
  - Model outputs: scores and decisions
  - Model metrics: accuracy, sensitivity, specificity
  - Validation

Logistics notes

- Discussion on fairness definitions posted, due next Tuesday.
- Your **project** should be
  - **Interesting**: not every project will go into depth in every aspect (e.g., some won't have much data wrangling), but all projects should be interesting in *some* aspect.
  - **Your own**: examples abound on the Internet. Following a tutorial is a very boring project. Adapting its approach to a new dataset or question? Interesting.

Setup

```r
library(tidyverse)
library(tidymodels)
library(ggridges)
```

Can a blood test diagnose autism?

We'll use an example from https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005385

```r
data_filename <- "data/autism.csv"
if (!file.exists(data_filename)) {
  dir.create("data")
  download.file("https://doi.org/10.1371/journal.pcbi.1005385.s001", data_filename)
}
```

```r
col_names <- names(read_csv(data_filename, n_max = 1, col_types = cols(.default = col_character())))
autism <- read_csv(data_filename, skip = 2, col_names = col_names, col_types = cols(
  .default = col_double(),
  Group = col_character()
)) %>% mutate(
  Group = as_factor(Group)
)
```

We have 3 kinds of data about 206 children:

1. The outcome (`Group`): ASD (diagnosed with ASD), SIB (sibling not diagnosed with ASD), and NEU (age-matched neurotypical children, for control)

```
autism %>% group_by(Group) %>% summarize(n = n()) %>% kable()                    (r)
```

Group	n
ASD	83
NEU	76
SIB	47

1. The outcome (`Group`): ASD, SIB, NEU
2. Concentrations of various metabolites in a blood sample:

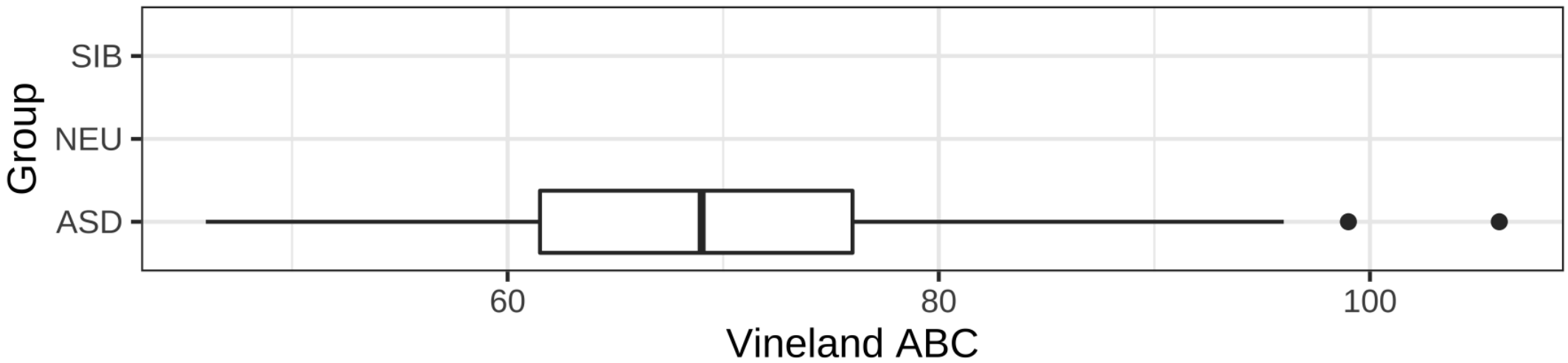
```
autism %>% select(-1, -last_col())                                                (r)
```

```
## # A tibble: 206 x 24
##   Methion.   SAM   SAH `SAM/SAH` `% DNA methylat… `8-OHG` Adenosine Homocysteine Cysteine
##      <dbl> <dbl> <dbl>     <dbl>             <dbl>   <dbl>     <dbl>        <dbl>    <dbl>
## 1     17.3  56.2 15.2       3.69              3.35   0.055     0.103         4.48     215.
## 2     14.9  37.2  7.58      4.91              3.04   0.045     0.055         4.31     163.
## 3     15.9  37.9  9.87      3.84              2.81   0.058     0.071         6.21     159.
## 4     18.7  79.2 24.5       3.23              4.24   0.085     0.079         7.15     226.
## 5     21.5  77.6 19.2       4.04              3.49   0.041     0.058         3.82     201.
## 6     18.1  67.6 12.8       5.30              3.01   0.156     0.091         4.03     205.
## # … with 200 more rows, and 15 more variables: `Glu.-Cys.` <dbl>, `Cys.-Gly.` <dbl>, tGSH <dbl>,
## #   fGSH <dbl>, GSSG <dbl>, `fGSH/GSSG` <dbl>, `tGSH/GSSG` <dbl>, Chlorotyrosine <dbl>,
## #   Nitrotyrosine <dbl>, Tyrosine <dbl>, Tryptophane <dbl>, fCystine <dbl>, fCysteine <dbl>,
## #   `fCystine/fCysteine` <dbl>, `% oxidized` <dbl>
```

1. The outcome (`Group`): ASD, SIB, NEU
2. Concentrations of various metabolites in a blood sample
3. For the ASD children only, a measure of life skills ("Vineland ABC")

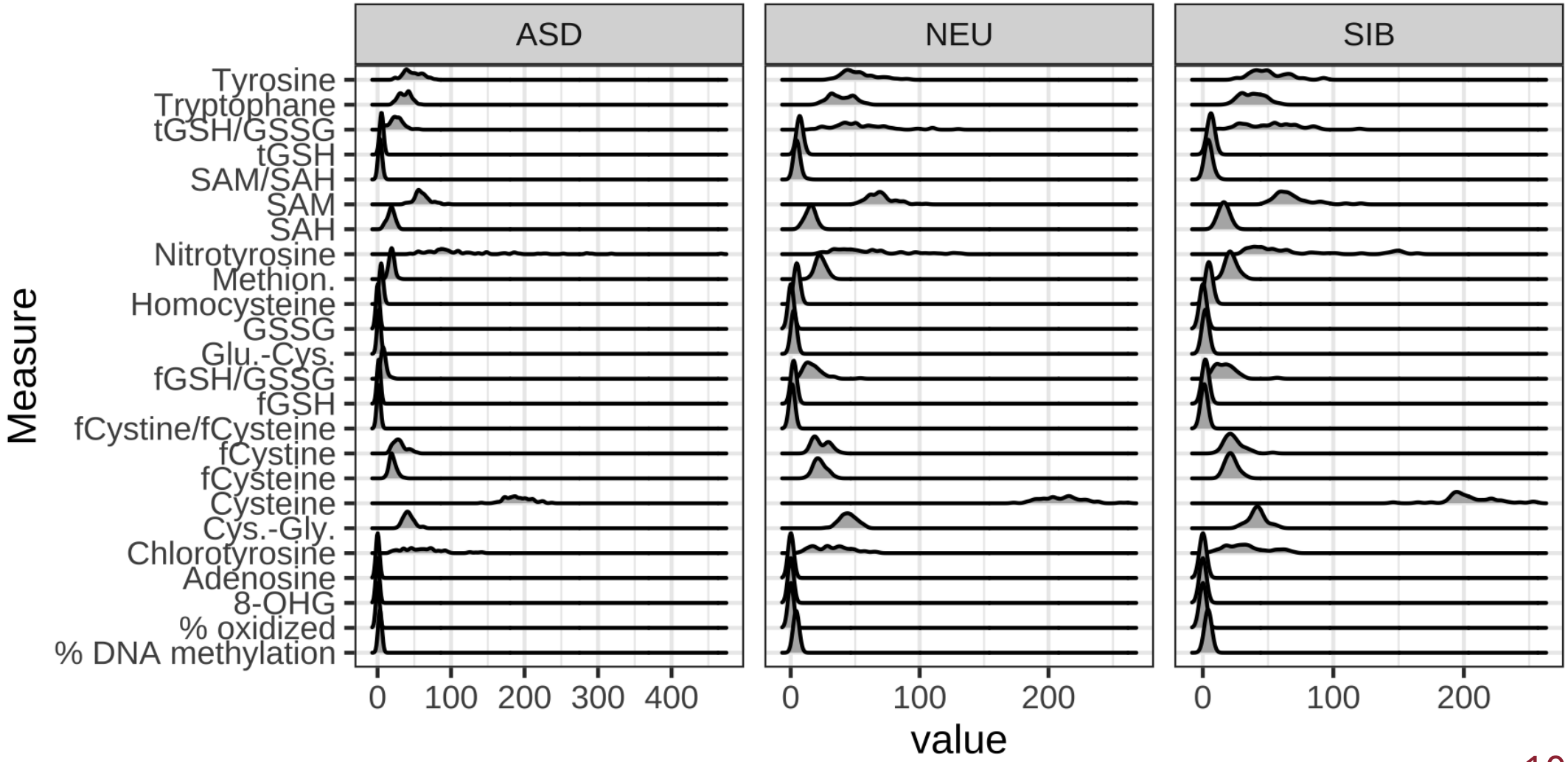
```r
autism %>%
  ggplot(aes(x = `Vineland ABC`, y = Group)) + geom_boxplot()
```

```
## Warning: Removed 159 rows containing non-finite values (stat_boxplot).
```



Exploratory Data Analysis (EDA)

What do these metabolites look like?

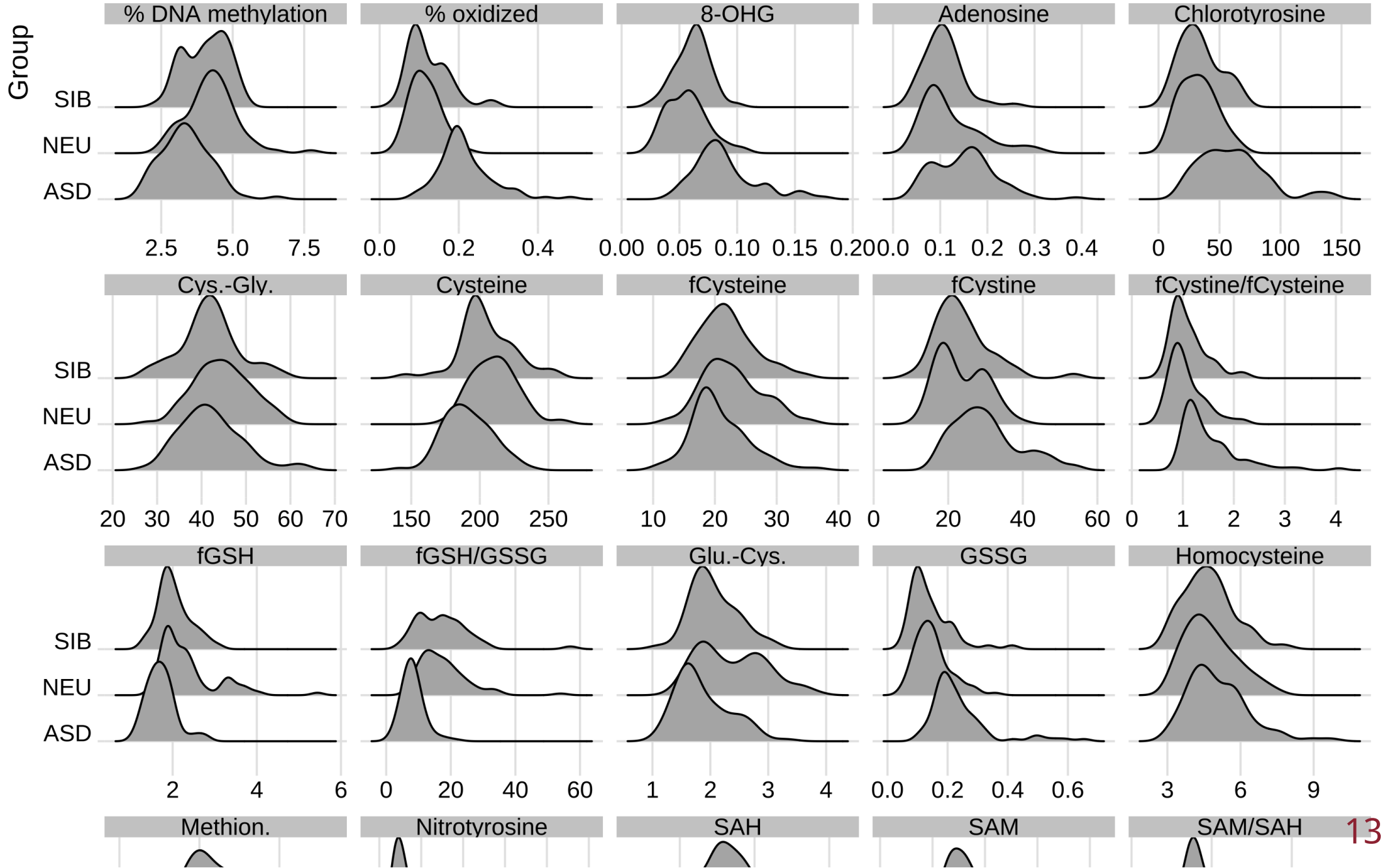


code for the previous plot:

```r
autism %>%
  select(-`Vineland ABC`) %>%
  pivot_longer(-Group, names_to = "Measure") %>%
  ggplot(aes(x = value, y = Measure)) +
  geom_density_ridges() +
  facet_wrap(vars(Group), scales = "free_x")
```

EDA

Better question: **Can these metabolites help us distinguish autism?**


Group
SIB
NEU
ASD
% DNA methylation
% oxidized
8-OHG
Adenosine
Chlorotyrosine
Cys.-Gly.
Cysteine
fCysteine
fCystine
fCystine/fCysteine
fGSH
fGSH/GSSG
Glu.-Cys.
GSSG
Homocysteine
Methion.
Nitrotyrosine
SAH
SAM
SAM/SAH

code for previous plot:

```r
autism %>%
  select(-`Vineland ABC`) %>%
  pivot_longer(-Group, names_to = "Measure") %>%
  ggplot(aes(x = value, y = Group)) +
  geom_density_ridges() +
  facet_wrap(vars(Measure), scales = "free_x") +
  theme_ridges()
```

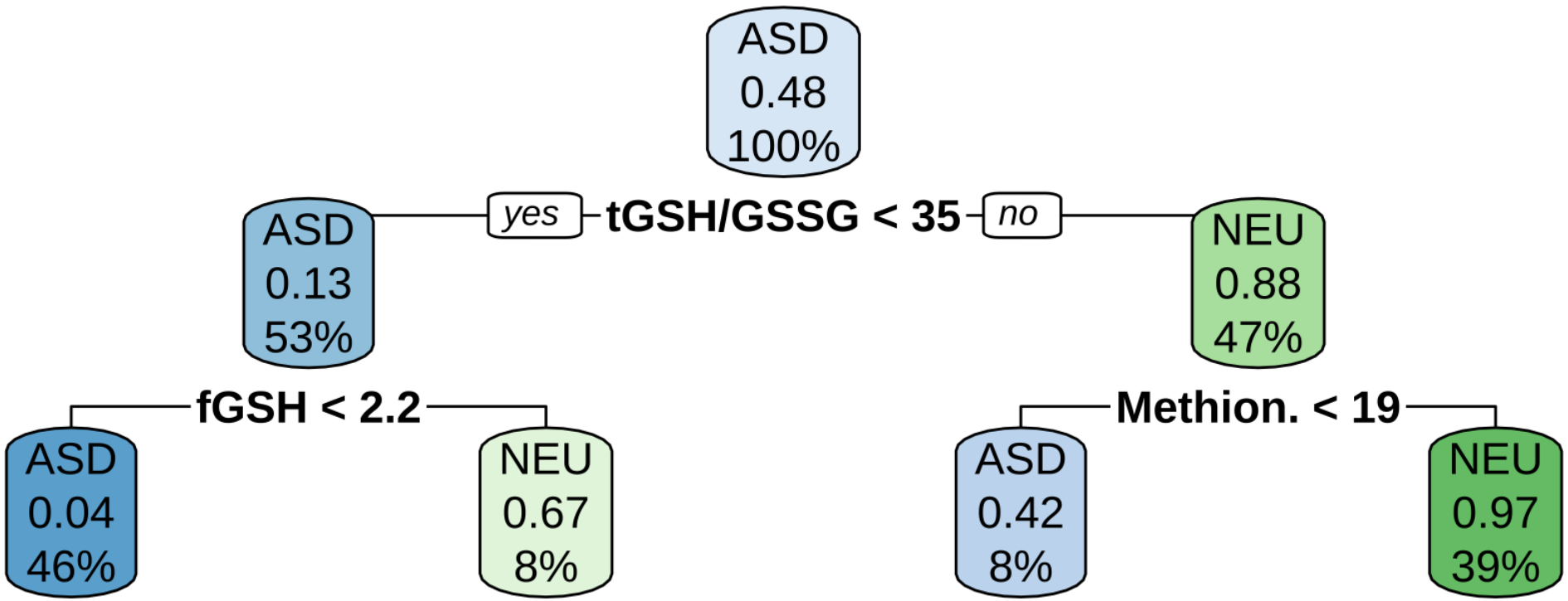
Can we predict ASD vs non-ASD from metabolites?

- Let's start by (1) ignoring the behavior scores (that's an *outcome*) and comparing just ASD and NEU.
- We need to drop SIB... and tell the model that we don't actually care about it.

```r
data <-
  autism %>%
  select(-`Vineland ABC`) %>%
  filter(Group != "SIB") %>%
  mutate(Group = factor(Group))
```

Decision Tree *Classification*

```
spec <- workflow() %>% add_recipe(                                    (r)
  recipe(Group ~ ., data = data)) %>%
  add_model(decision_tree(mode = "classification") %>% set_engine("rpart"))
model <- spec %>% fit(data)
```



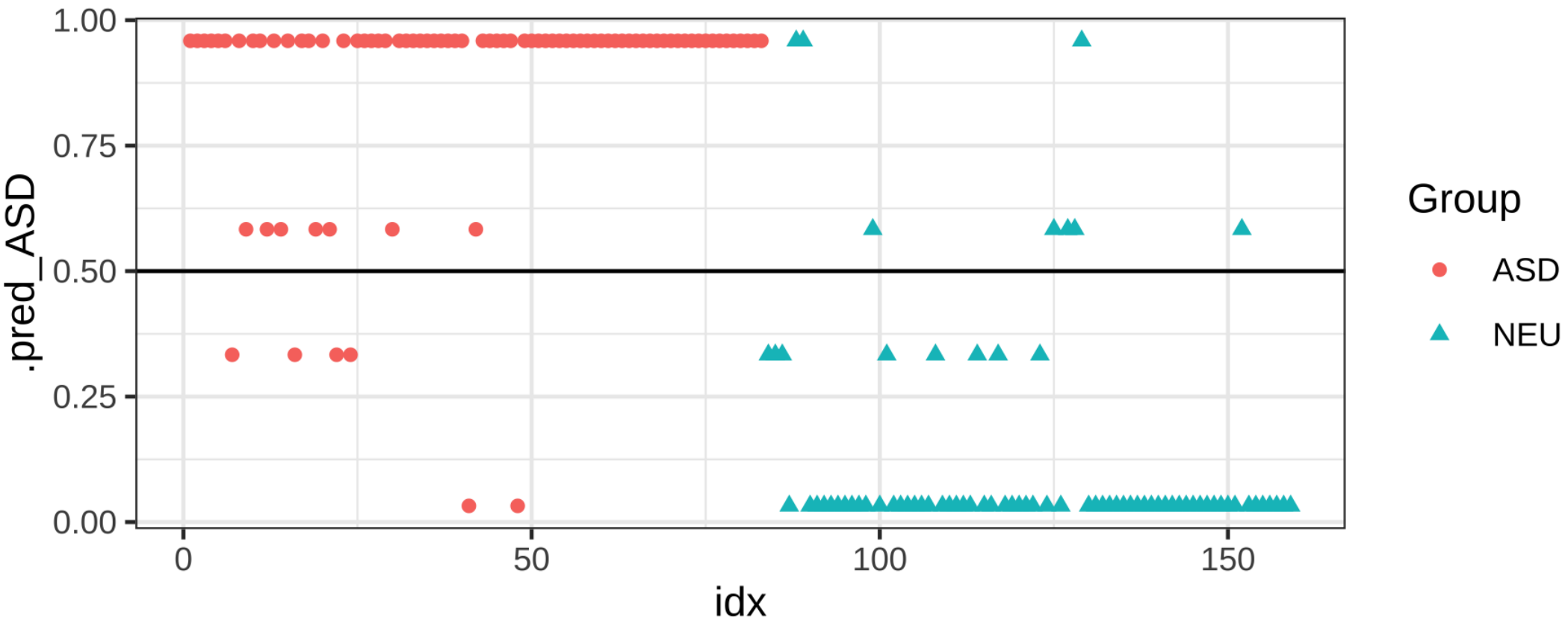
What do the *predictions* look like?

```r
model %>% predict(data, type = "prob")
```

```
## # A tibble: 159 x 2
##   .pred_ASD .pred_NEU
##       <dbl>     <dbl>
## 1     0.959    0.0411
## 2     0.959    0.0411
## 3     0.959    0.0411
## 4     0.959    0.0411
## 5     0.959    0.0411
## 6     0.959    0.0411
## # … with 153 more rows
```

Were those predictions *good*?

```r
model %>%
  predict(data, type = "prob") %>%
  bind_cols(data) %>%
  mutate(idx = row_number()) %>%
  ggplot(aes(x = idx, y = .pred_ASD, color = Group, shape = Group)) +
  geom_hline(yintercept = .5) +
  geom_point()
```



Quantifying that:

```
metrics <- yardstick::metric_set(accuracy, sensitivity, specificity)                    (r)
model %>%
  predict(data, type = "class") %>%
  bind_cols(data) %>%
  metrics(truth = Group, estimate = .pred_class)
```

```
## # A tibble: 3 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.912
## 2 sens     binary         0.928
## 3 spec     binary         0.895
```

Recall from Week 6...

	Seizure happened	No seizure happened
Seizure predicted	True positive	False positive (Type 1 error)
No seizure predicted	False negative (Type 2 error)	True negative

Recall from Week 6...

	Seizure happened	No seizure happened
Seizure predicted	True positive	False positive (Type 1 error)
No seizure predicted	False negative (Type 2 error)	True negative

- **Accuracy** (% correct) = (TP + TN) / (# episodes)
- **False negative** ("miss") **rate** = FN / (# actual seizures)
- **False positive** ("false alarm") **rate** = FP / (# true non-seizures)

Recall from Week 6...

	Seizure happened	No seizure happened
Seizure predicted	True positive	False positive (Type 1 error)
No seizure predicted	False negative (Type 2 error)	True negative

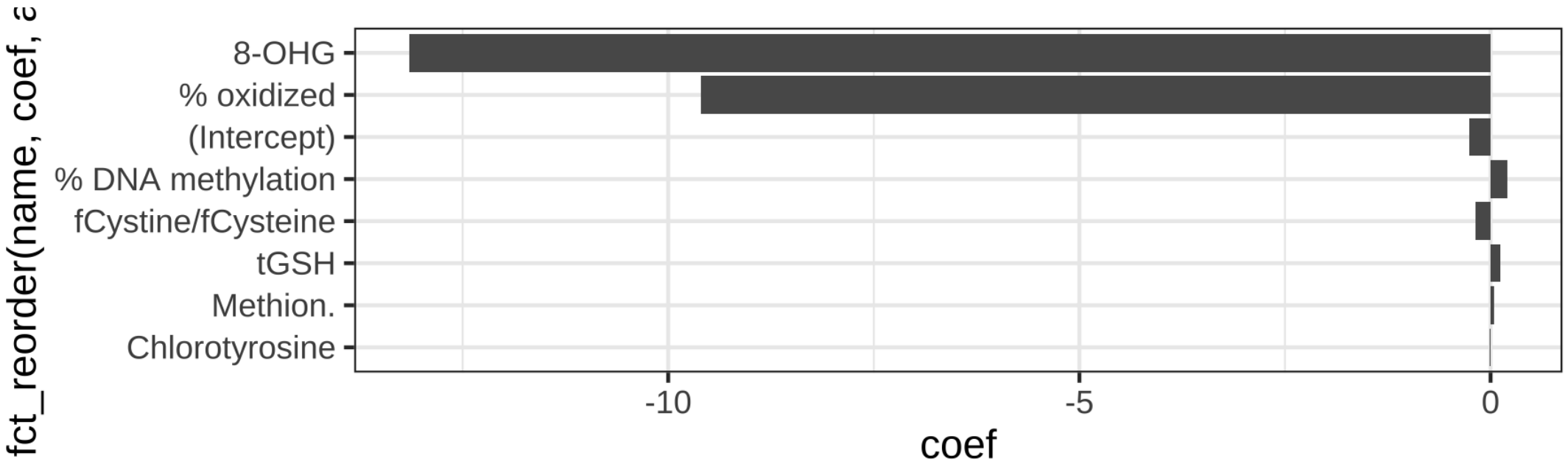
- **Accuracy** (% correct) = (TP + TN) / (# episodes)
- **False negative** ("miss") **rate** = FN / (# actual seizures)
- **False positive** ("false alarm") **rate** = FP / (# true non-seizures)
- **Sensitivity** ("true positive rate") = TP / (# actual seizures)
  - Sensitivity = 1 – False negative rate
- **Specificity** ("true negative rate") = TN / (# actual seizures)
  - Specificity = 1 – False positive rate
- Wikipedia article

Logistic Regression

Logistic Regression

```
spec <- workflow() %>% add_recipe(                                                    (r)
  recipe(Group ~ ., data = data)) %>%
  add_model(logistic_reg(penalty = .001) %>% set_engine("glmnet"))
model <- spec %>% fit(data)
```

```
model %>% pull_workflow_fit() %>% pluck('fit') %>% coef(s = .1) %>% as.matrix() %>% as_tibble(rownames = "name") %>%   (r)
  rename(coef = 2) %>% filter(abs(coef) > .01) %>%
  ggplot(aes(x = coef, y = fct_reorder(name, coef, abs))) + geom_col()
```



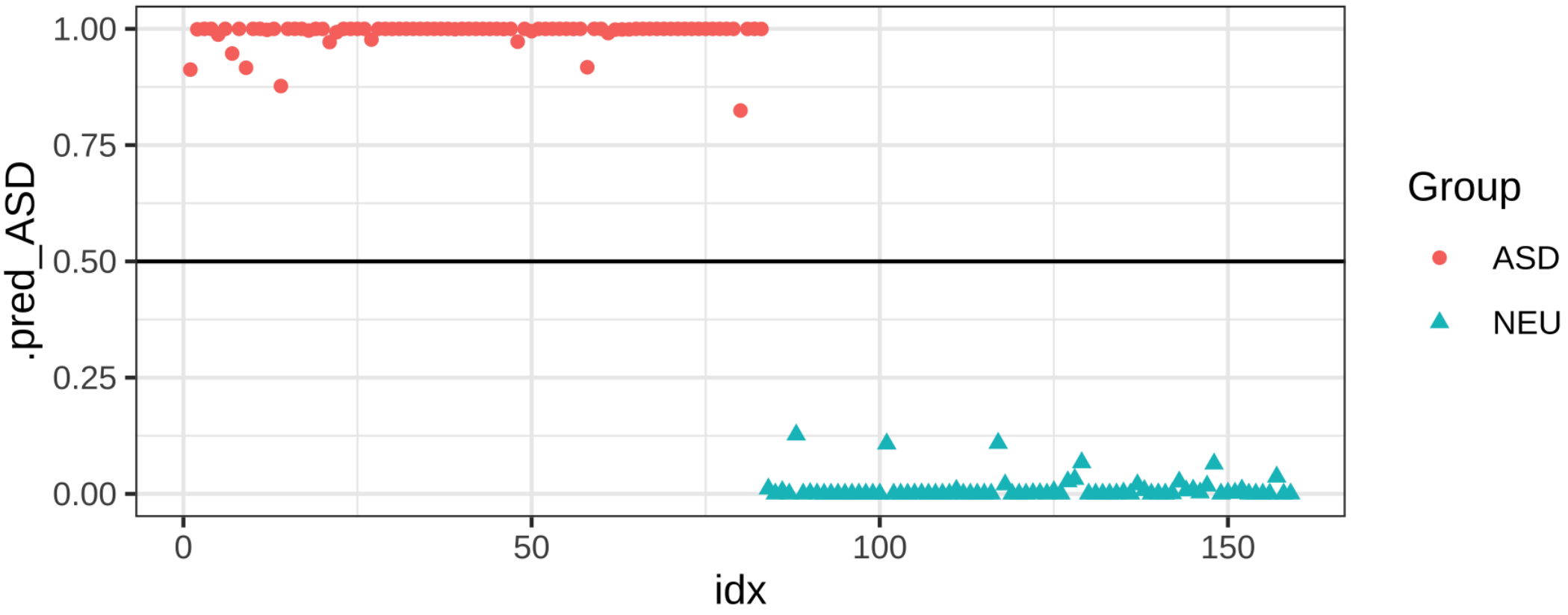
```r
model %>% predict(data, type = "prob")
```

```
## # A tibble: 159 x 2
##   .pred_ASD    .pred_NEU
##       <dbl>        <dbl>
## 1     0.912 0.0876
## 2     0.999 0.000964
## 3     1.00  0.0000000524
## 4     1.00  0.0000000369
## 5     0.988 0.0124
## 6     1.00  0.0000222
## # … with 153 more rows
```

Are those predictions good?

```r
model %>%
  predict(data, type = "prob") %>%
  bind_cols(data) %>%
  mutate(idx = row_number()) %>%
  ggplot(aes(x = idx, y = .pred_ASD, color = Group, shape = Group)) +
  geom_hline(yintercept = .5) +
  geom_point()
```

(r)



Quantifying that...

```
model %>%                                                                    (r)
  predict(data, type = "class") %>%
  bind_cols(data) %>%
  metrics(truth = Group, estimate = .pred_class)
```

```
## # A tibble: 3 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary             1
## 2 sens     binary             1
## 3 spec     binary             1
```

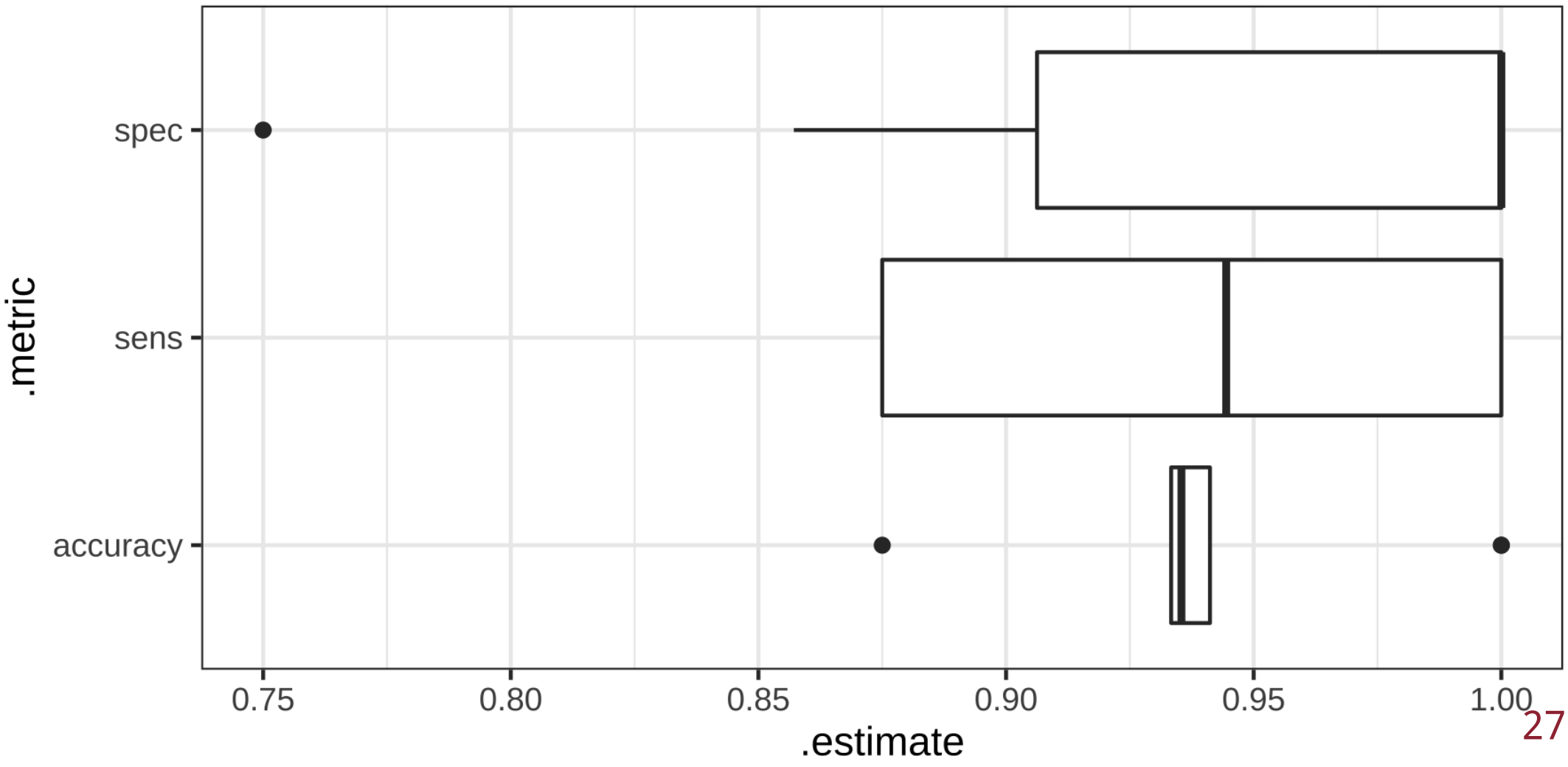
Cross validation!

Stratify by `group` to ensure each fold has good representation.

```
resamples <- data %>% vfold_cv(v = 10, strata = Group)                    (r)
```

```
cv_results <- spec %>%                                                   (r)
  fit_resamples(resamples, metrics = metrics)
```

```r
cv_results %>%
  collect_metrics(summarize = FALSE) %>%
  ggplot(aes(x = .estimate, y = .metric)) + geom_boxplot()
```



Cross-validate the decision tree

```r
spec <- workflow() %>% add_recipe(
  recipe(Group ~ ., data = data)) %>%
  add_model(decision_tree(mode = "classification") %>% set_engine("rpart"))

cv_results <- spec %>%
  fit_resamples(resamples, metrics = metrics)
```

```r
cv_results %>%
  collect_metrics(summarize = FALSE) %>%
  ggplot(aes(x = .estimate, y = .metric)) + geom_boxplot()
```