Logistic Regression (cont.)

Lecture 06

Dr. Colin Rundel

Full Model

Model

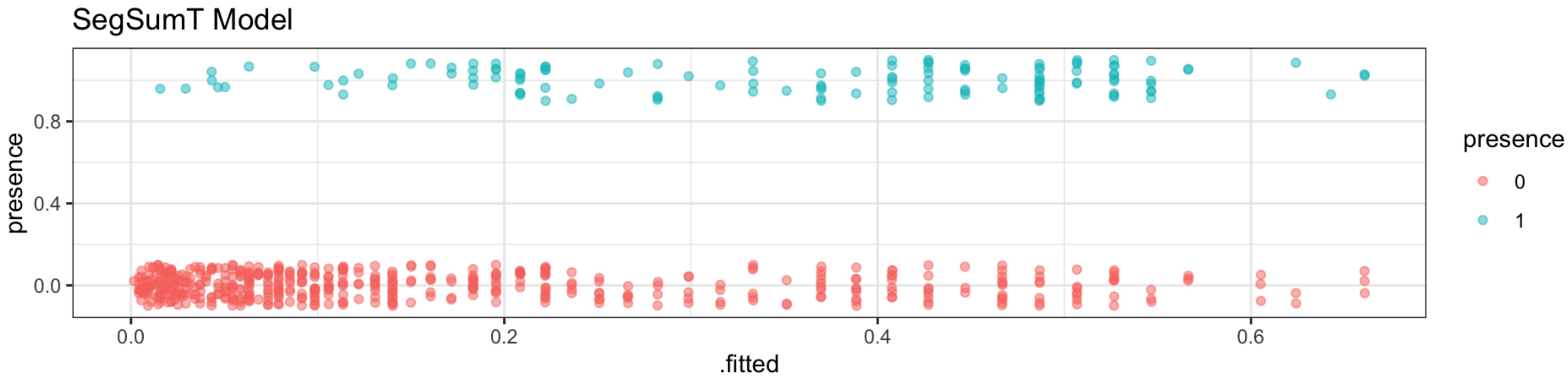
```r
f = glm(presence~., family=binomial, data=anguilla_train)
summary(f)
```

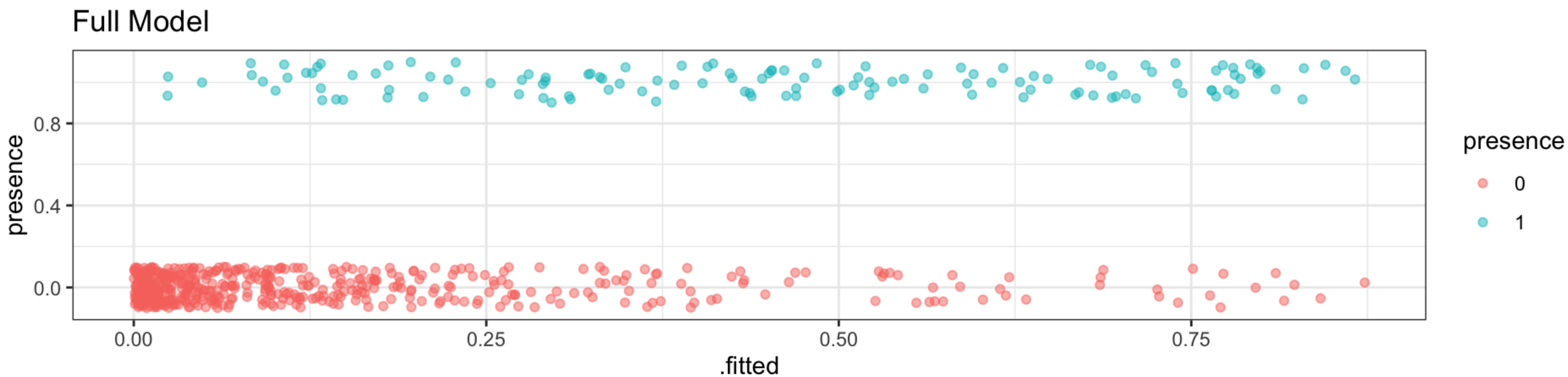
```
Call:
glm(formula = presence ~ ., family = binomial, data = anguilla_train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.03162  -0.55711  -0.27105  -0.08103   2.73104

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -9.352885   1.761202  -5.311 1.09e-07 ***
SegSumT        0.654186   0.096921   6.750 1.48e-11 ***
DSDist        -0.004837   0.002302  -2.102  0.03559 *
DSMaxSlope    -0.030776   0.061995  -0.496  0.61959
USRainDays    -0.710920   0.225814  -3.148  0.00164 **
USSlope       -0.069814   0.025443  -2.744  0.00607 **
```

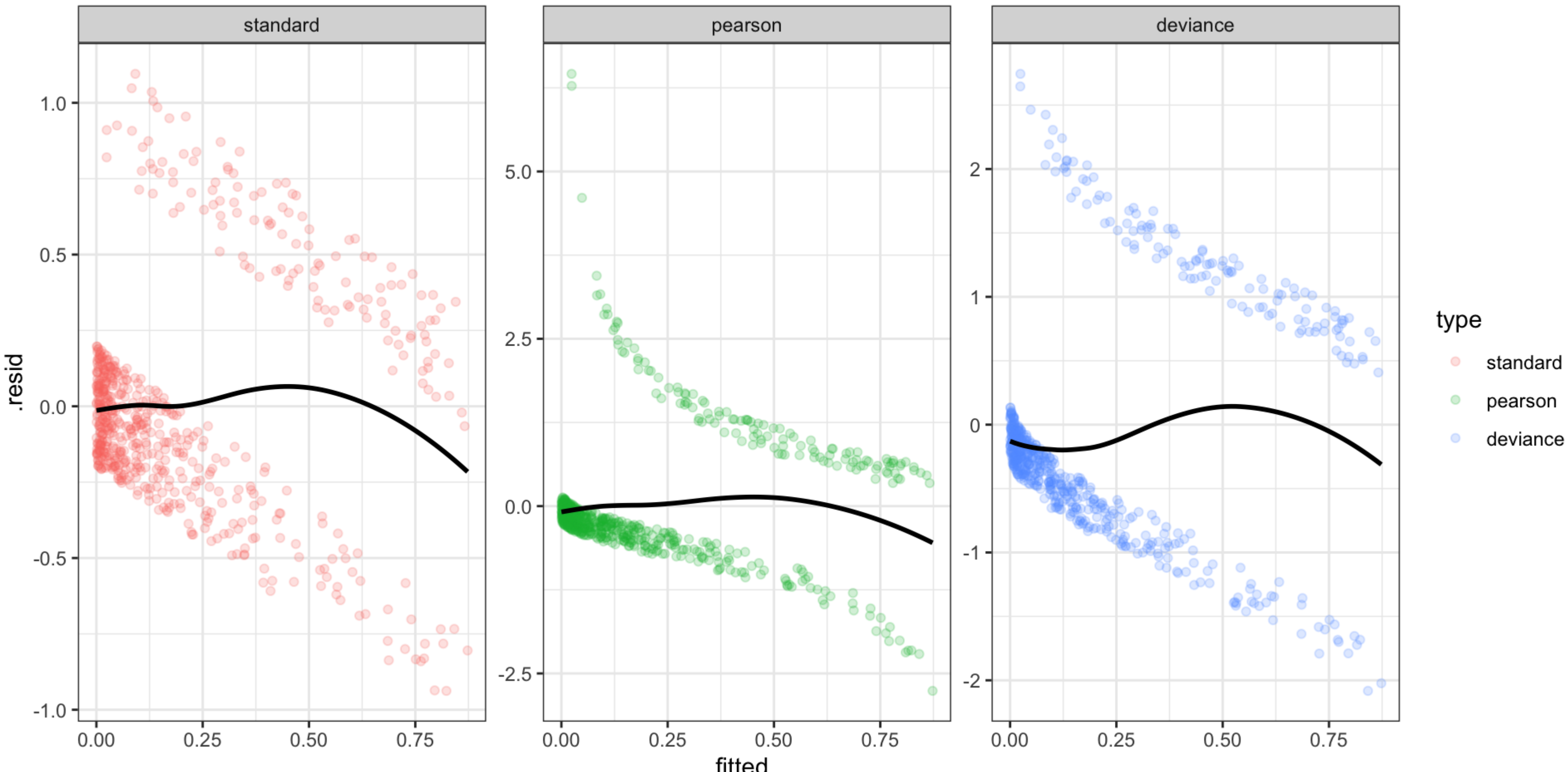
Separation


SegSumT Model
presence
0.8
0.4
0.0
0.0
0.2
0.4
0.6
.fitted
presence
0
1


Full Model
presence
0.8
0.4
0.0
0.00
0.25
0.50
0.75
.fitted
presence
0
1

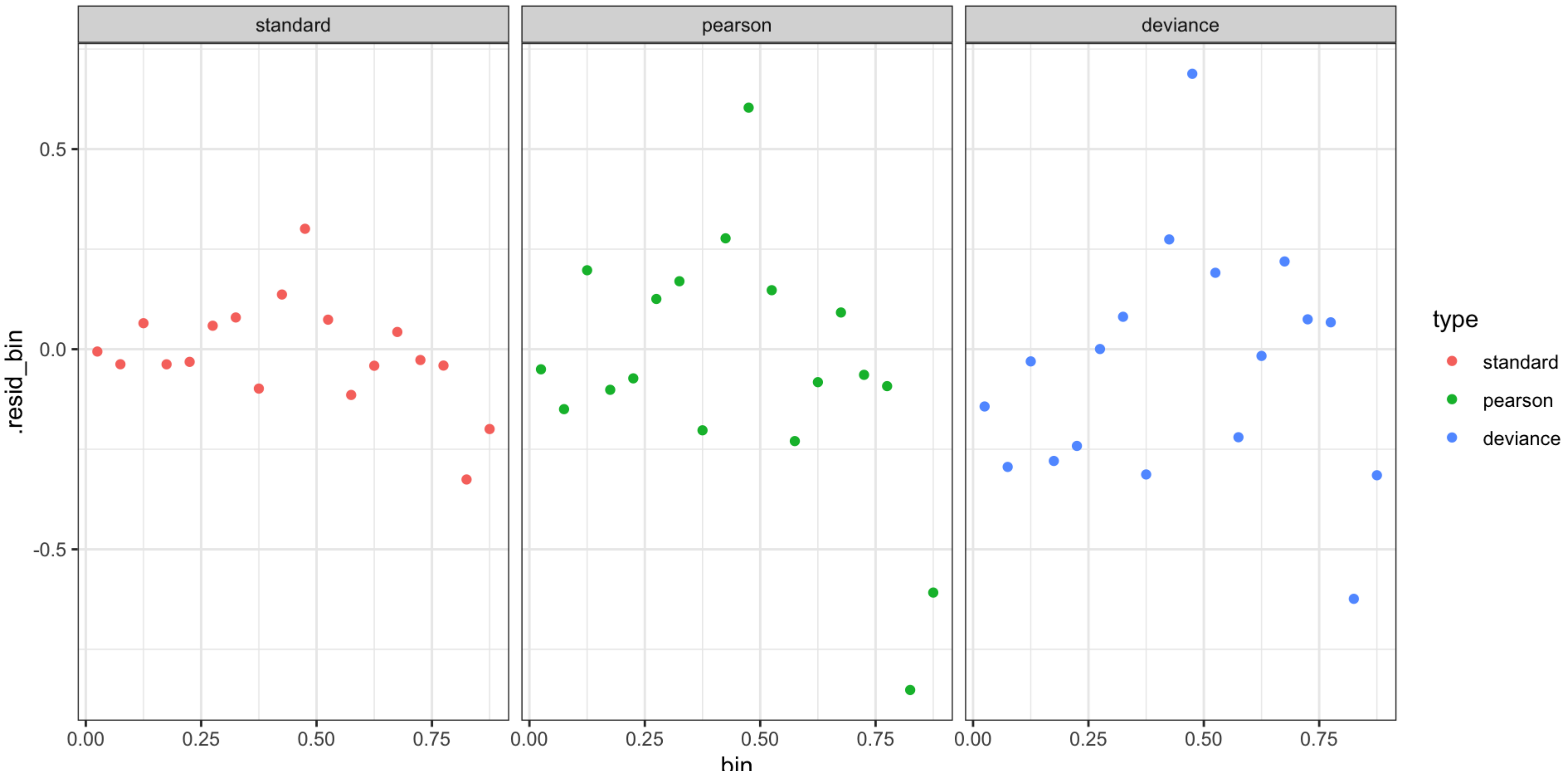
Residuals vs fitted

```r
f_resid |>
  ggplot(aes(x=.fitted, y=.resid, color=type)) +
  geom_jitter(height=0.2, alpha=0.2) +
  facet_wrap(~type, ncol=3, scale="free_y") +
  geom_smooth(se = FALSE, color="black")
```



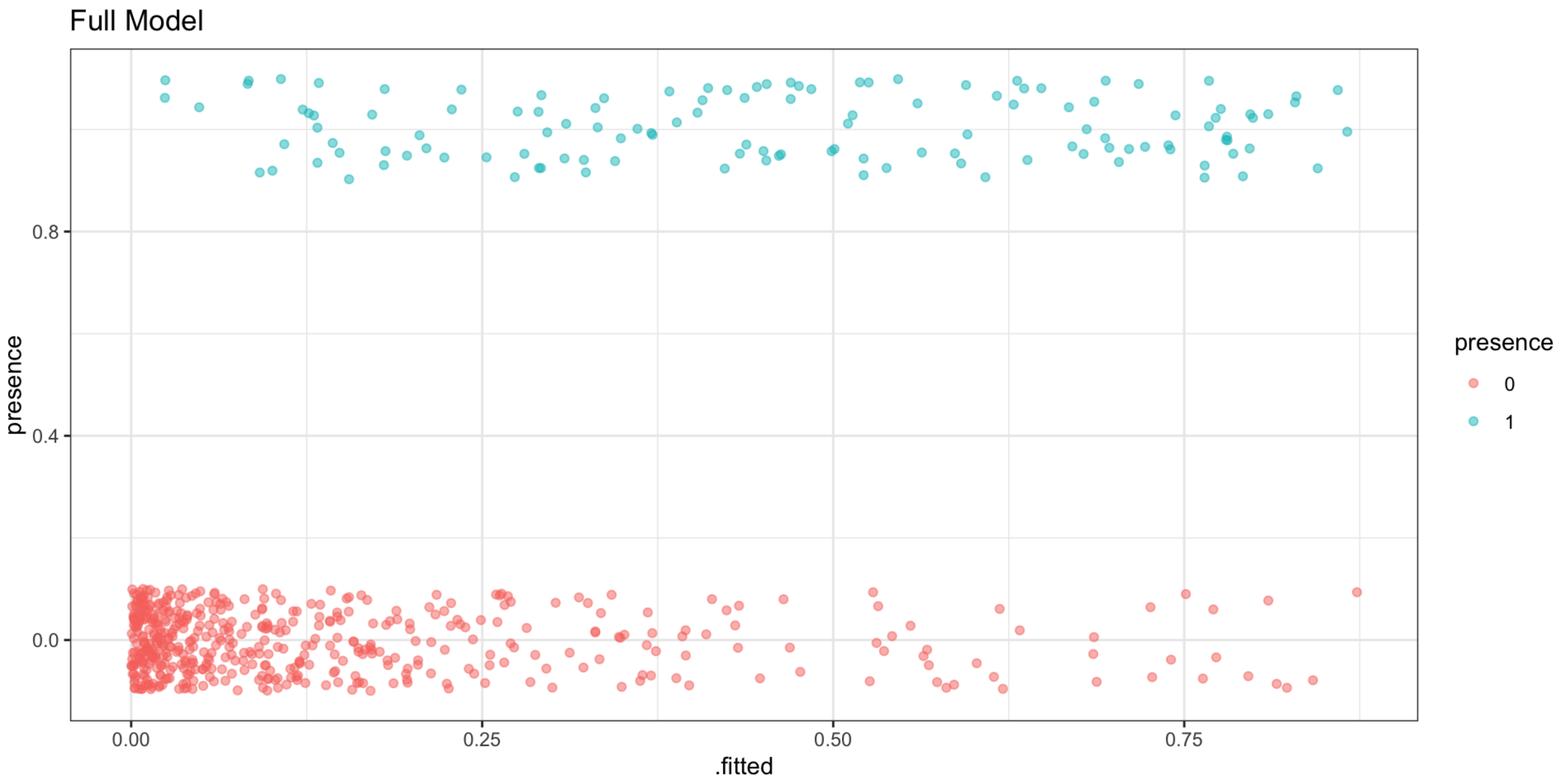
Residuals (binned) vs fitted

```
f_resid_bin |>
  mutate(type = as_factor(type)) |>
  ggplot(aes(x=bin, y=.resid_bin, color=type)) +
  geom_point() +
  facet_wrap(~type, ncol=3)
```

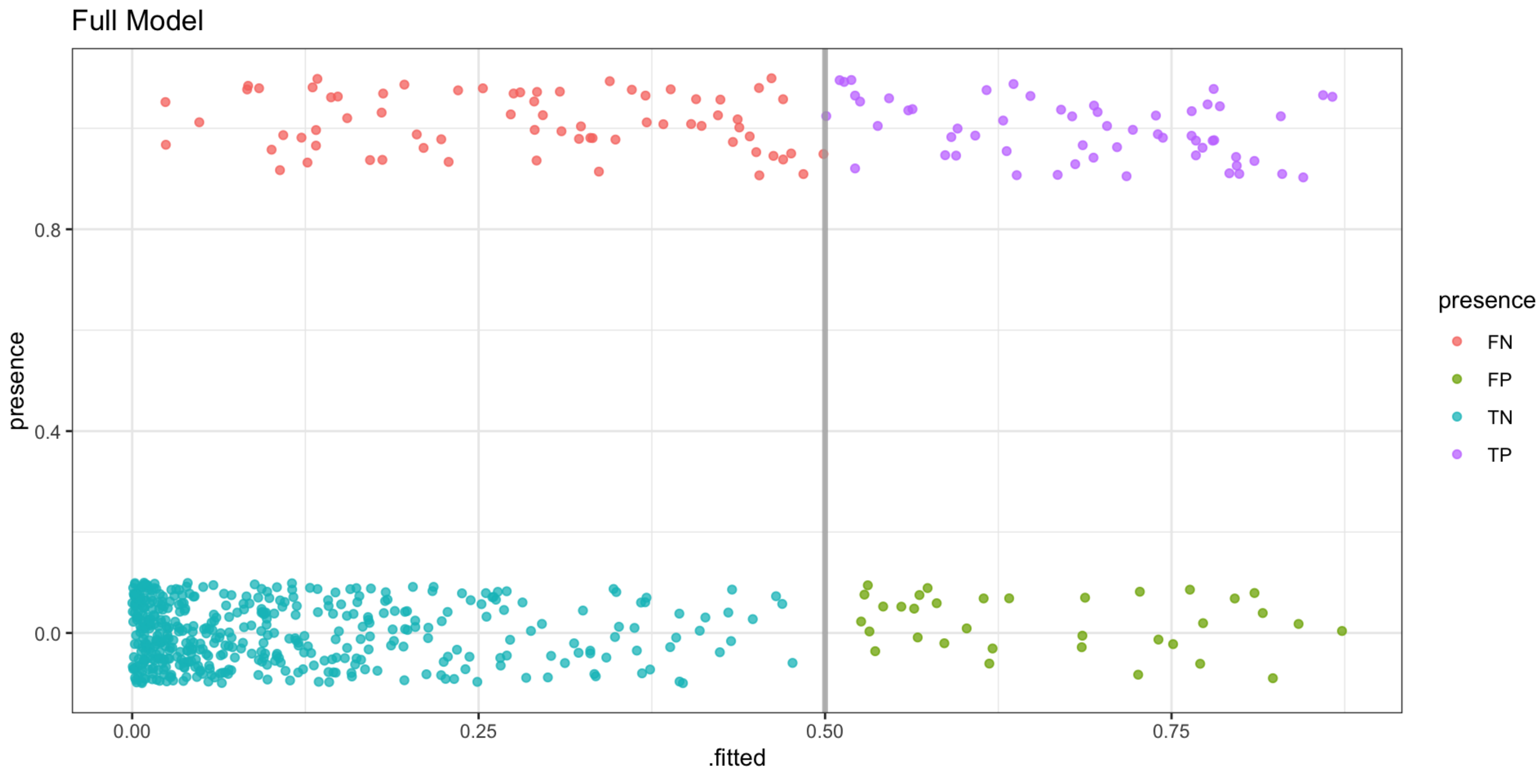


Model Performance

Confusion Matrix

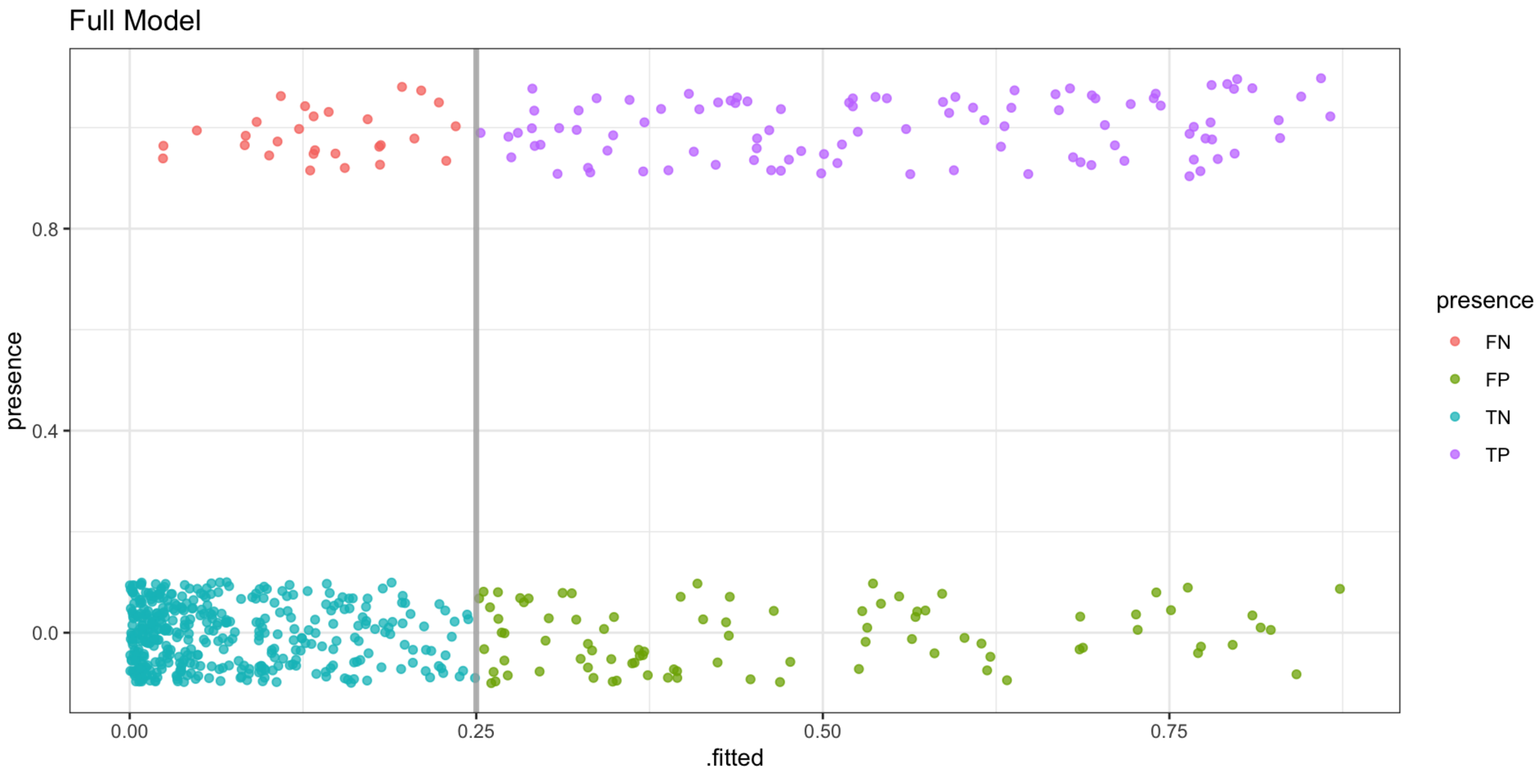


Confusion Matrix - 50% threshold



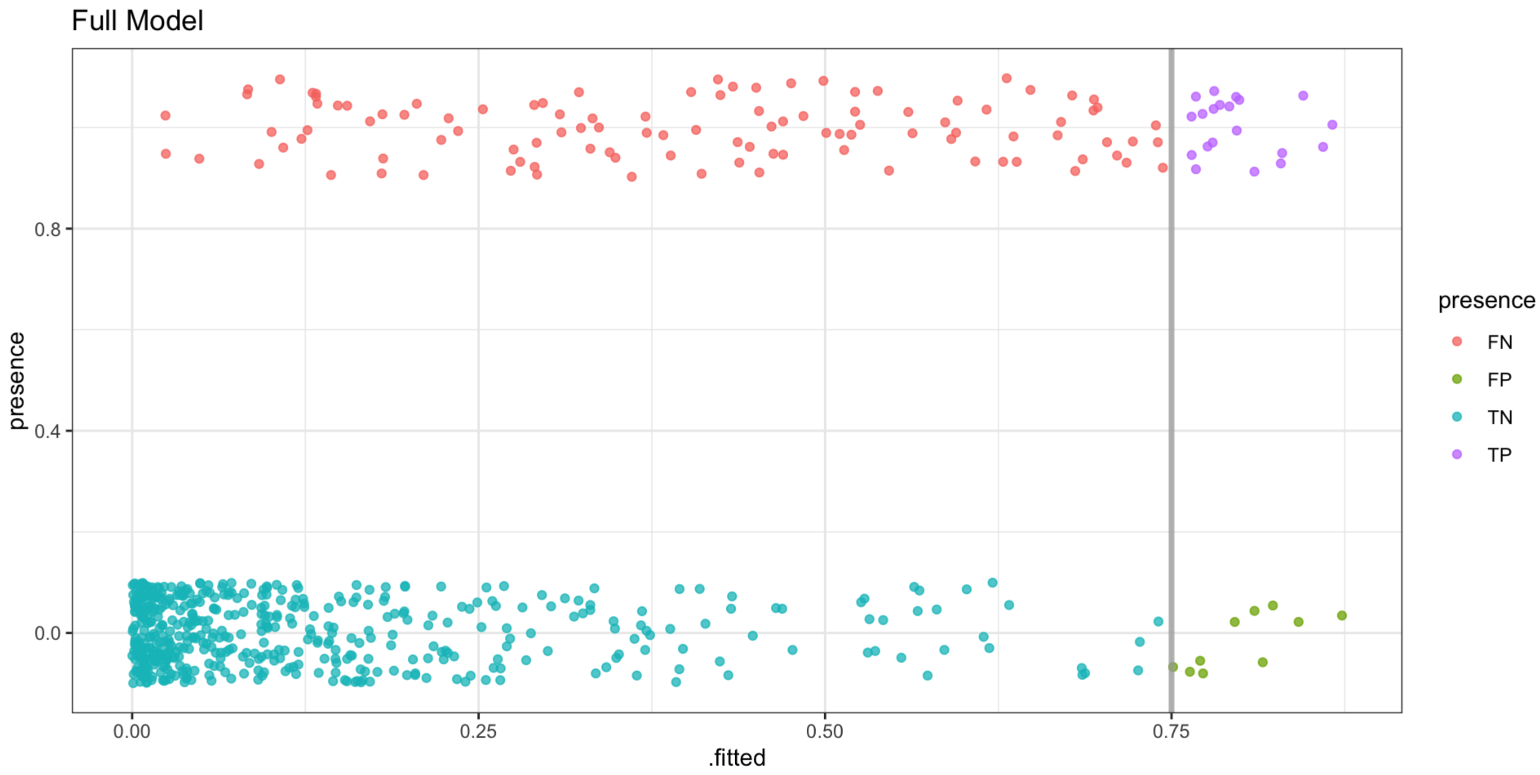

```
# A tibble: 4 × 2
  result     n
  <chr>  <int>
1 FN        70
2 FP        34
3 TN       457
4 TP        57
```

Confusion Matrix - 25% threshold




```
# A tibble: 4 × 2
  result     n
  <chr>  <int>
1 FN        28
2 FP        88
3 TN       403
4 TP        99
```

Confusion Matrix - 75% threshold




```
# A tibble: 4 × 2
  result     n
  <chr>  <int>
1 FN       107
2 FP        10
3 TN       481
4 TP        20
```

Confusion Matrix statistics

$$\text{Sensitivity} = \text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 1 - \text{FNR}$$

$$\text{Specificity} = \text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 1 - \text{FPR}$$

$$\text{Precision} = \text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F}_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Combining model predictions

```
( model_comb  = bind_rows(
    g_std |> mutate(model = "SegSumT"),
    f_std |> mutate(model = "Full")
  ) |>
    group_by(model)
)
```

```
# A tibble: 1,236 × 17
# Groups:   model [2]
   presence SegSumT .fitted  .resid .std.r…¹    .hat .sigma .cooksd model DSDist
      <int>   <dbl>   <dbl>   <dbl>    <dbl>   <dbl>  <dbl>   <dbl> <chr>  <dbl>
 1        0    16.4  0.131  -0.131    -0.530 0.00260  0.903 1.97e-4 SegS…     NA
 2        1    17.1  0.209   0.791     1.77  0.00232  0.901 4.43e-3 SegS…     NA
 3        0    14    0.0216 -0.0216   -0.209 0.00231  0.903 2.56e-5 SegS…     NA
 4        0    18.2  0.389  -0.389    -0.994 0.00364  0.903 1.17e-3 SegS…     NA
 5        0    15.6  0.0735 -0.0735   -0.391 0.00286  0.903 1.14e-4 SegS…     NA
 6        0    18.3  0.408  -0.408    -1.03  0.00395  0.902 1.37e-3 SegS…     NA
 7        0    18.5  0.447  -0.447    -1.09  0.00466  0.902 1.90e-3 SegS…     NA
 8        0    16.2  0.114  -0.114    -0.492 0.00270  0.903 1.74e-4 SegS…     NA
 9        0    18    0.351  -0.351    -0.932 0.00313  0.903 8.53e-4 SegS…     NA
10        1    17.3  0.236   0.764     1.70  0.00233  0.901 3.79e-3 SegS…     NA
# … with 1,226 more rows, 7 more variables: DSMaxSlope <dbl>, USRainDays <dbl>,
#   USSlope <dbl>, USNative <dbl>, DSDam <int>, Method <fct>, LocSed <dbl>, and
```

Receiver operating characteristic (ROC)

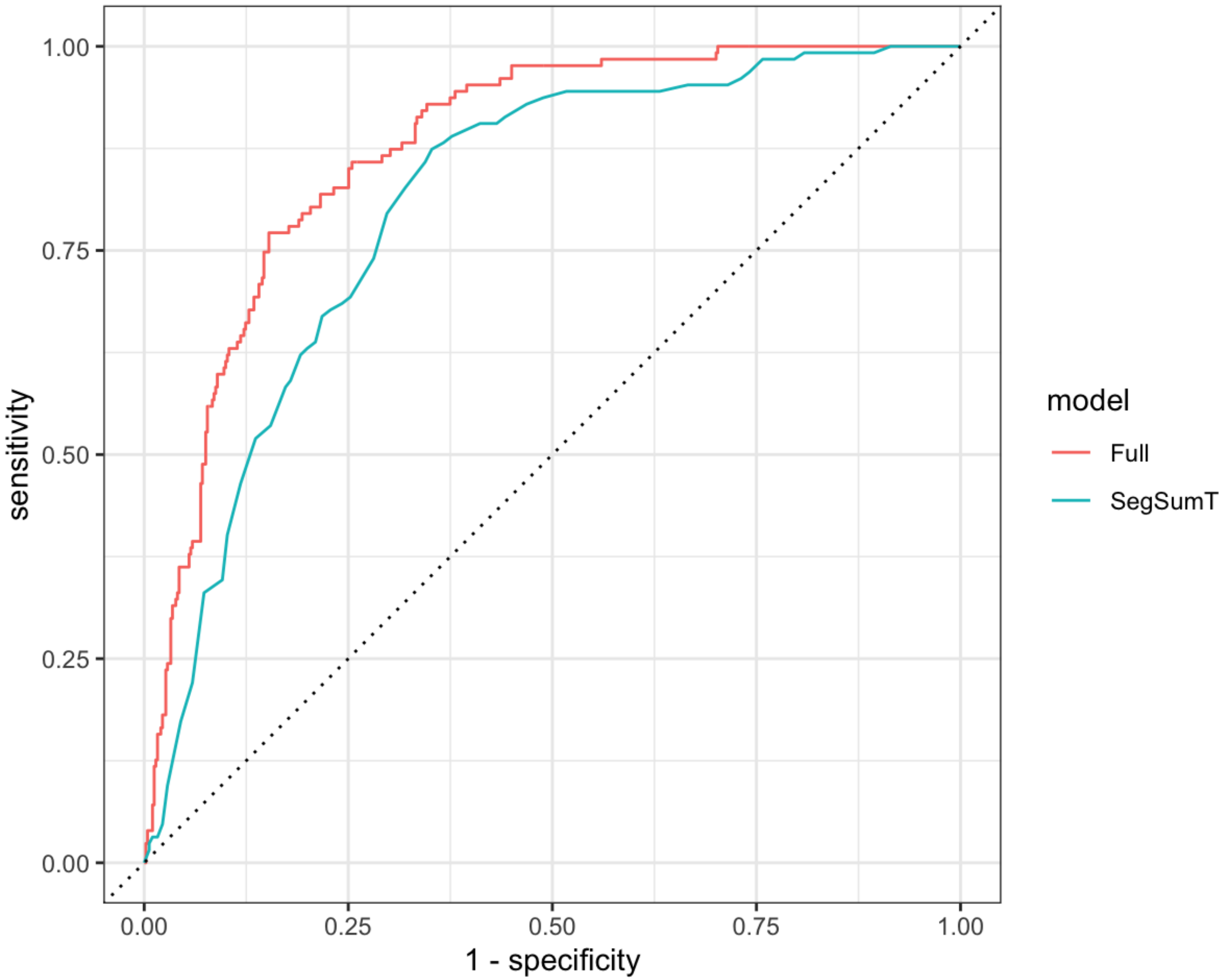
$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

```r
( model_roc = model_comb |>
    yardstick::roc_curve(factor(presence, levels = c(1,0)), .fitted)
)
```

```
# A tibble: 696 × 4
# Groups:   model [2]
   model  .threshold specificity sensitivity
   <chr>       <dbl>       <dbl>       <dbl>
 1 Full   -Inf             0               1
 2 Full      0.000132      0               1
 3 Full      0.000425      0.00204         1
 4 Full      0.000453      0.00407         1
 5 Full      0.000755      0.00611         1
 6 Full      0.000761      0.00815         1
 7 Full      0.000792      0.0102          1
 8 Full      0.00108       0.0122          1
 9 Full      0.00126       0.0143          1
10 Full      0.00146       0.0163          1
# … with 686 more rows
```

ROC Curve

```
model_roc |>
  autoplot()
```



AUC (area under the curve)

```r
model_comb |>
  yardstick::roc_auc(factor(presence, levels = c(1,0)), .fitted)
```

```
# A tibble: 2 × 4
  model   .metric .estimator .estimate
  <chr>   <chr>   <chr>          <dbl>
1 Full    roc_auc binary         0.875
2 SegSumT roc_auc binary         0.806
```

A model that randomly assigns classes to the data is expected to achieve an AUC of 0.5 (dotted line on the previous plot) while a perfect model would achieve an AUC of 1.

Precision / Recall

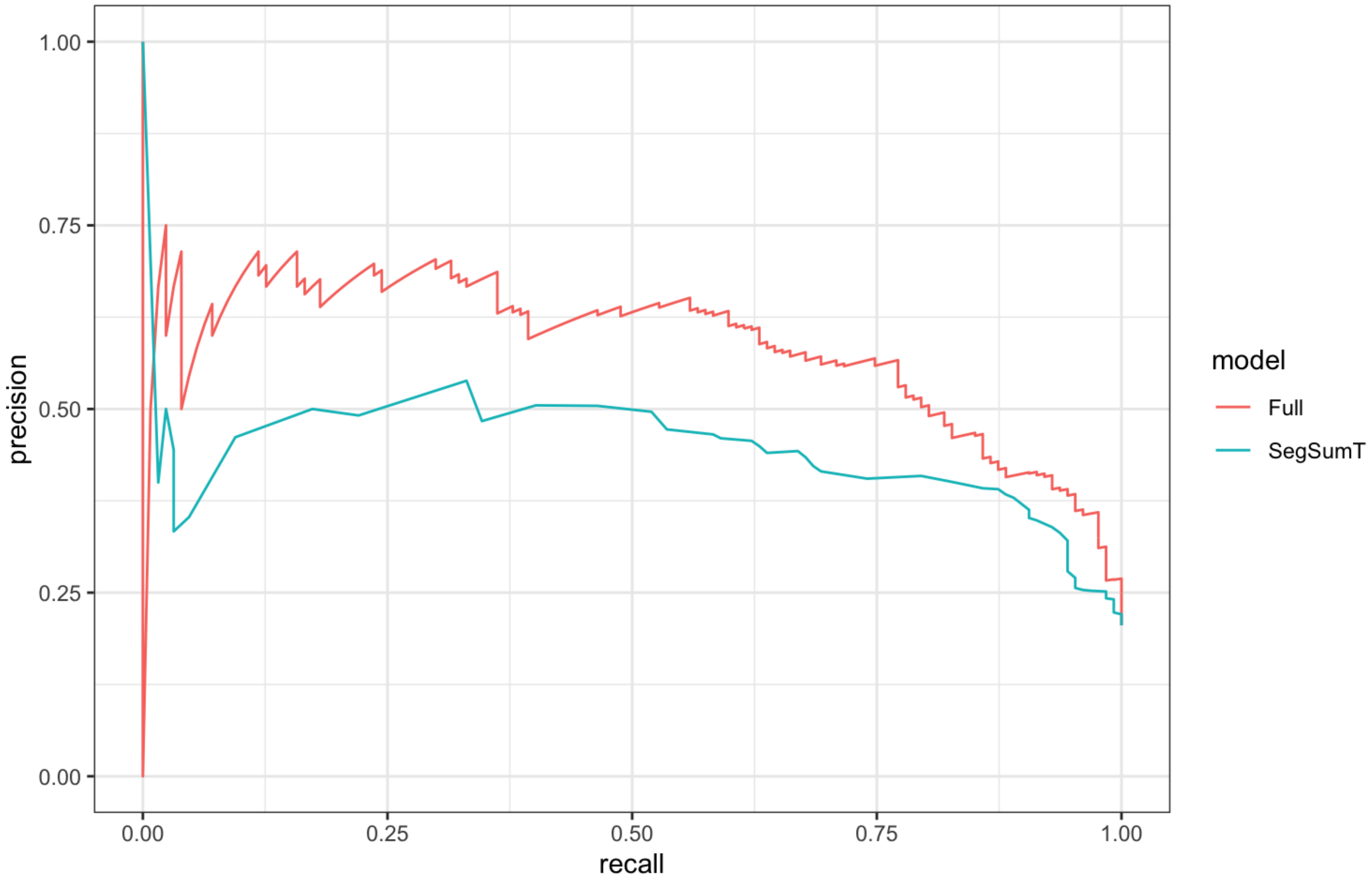
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

```r
( model_pr = model_comb |>
    yardstick::pr_curve(factor(presence, levels = c(1,0)), .fitted)
)
```

```
# A tibble: 694 × 4
# Groups:   model [2]
   model .threshold  recall precision
   <chr>      <dbl>   <dbl>     <dbl>
 1 Full      Inf     0              1
 2 Full        0.873 0              0
 3 Full        0.866 0.00787      0.5
 4 Full        0.859 0.0157       0.667
 5 Full        0.845 0.0236       0.75
 6 Full        0.842 0.0236       0.6
 7 Full        0.830 0.0315       0.667
 8 Full        0.829 0.0394       0.714
 9 Full        0.823 0.0394       0.625
10 Full        0.816 0.0394       0.556
# … with 684 more rows
```

Precision Recall curve

```
model_pr |>
  autoplot()
```



Precision Recall auc

```
model_comb |>
  yardstick::pr_auc(factor(presence, levels = c(1,0)), .fitted)
```

```
# A tibble: 2 × 4
  model   .metric .estimator .estimate
  <chr>   <chr>   <chr>          <dbl>
1 Full    pr_auc  binary         0.583
2 SegSumT pr_auc  binary         0.447
```

A model that randomly assigns classes to the data is expected to achieve an PR-AUC of # successes / n while a perfect model would achieve an AUC of 1 (a point at a coordinate of (1,1)).

What about the test data?

Combining predicitons

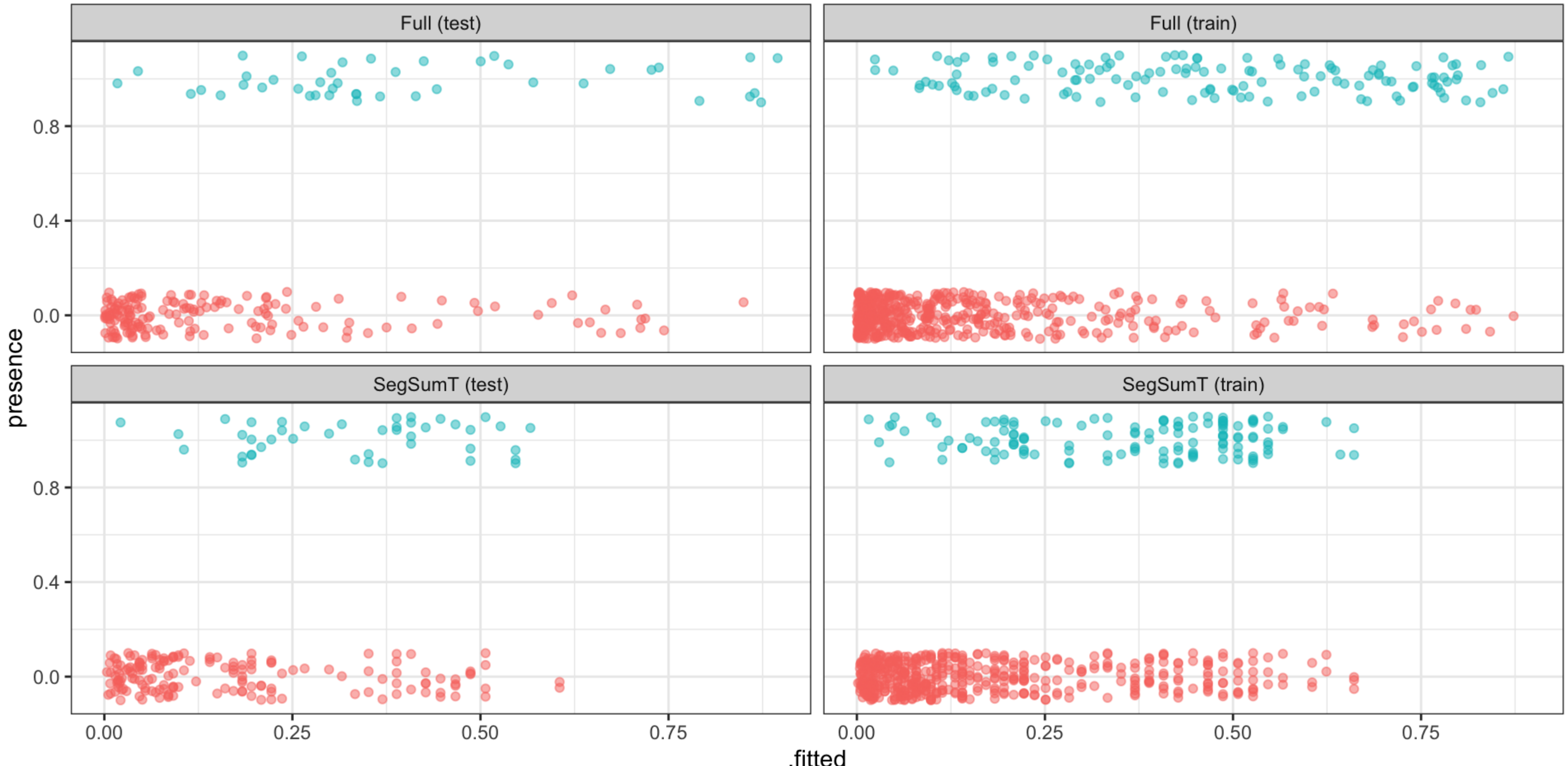
```r
(model_comb = bind_rows(
    broom::augment(g, newdata=anguilla_train,   type.predict="response") |>
      mutate(model = "SegSumT (train)"),
    broom::augment(g, newdata=anguilla_test,    type.predict="response") |>
      mutate(model = "SegSumT (test)"),
    broom::augment(f, newdata=anguilla_train,   type.predict="response") |>
      mutate(model = "Full (train)"),
    broom::augment(f, newdata=anguilla_test,    type.predict="response") |>
      mutate(model = "Full (test)"),
  ) |>
    group_by(model)
)
```

```
# A tibble: 1,648 × 12
# Groups:   model [4]
   presence SegSumT DSDist DSMaxSl…¹ USRai…² USSlope USNat…³ DSDam Method LocSed
      <int>   <dbl>  <dbl>     <dbl>   <dbl>   <dbl>   <dbl> <int> <fct>   <dbl>
 1        0    16.4  97.8       6.28   1.51     24.6    0.81     0 elect…    4.5
 2        1    17.1  13.9       0.57   1.98      3.3    0.13     0 net       1.8
 3        0    14     1.84      0.57   0.29     10.1    0.37     0 elect…    4.7
 4        0    18.2 121.        0.57   0.894     1.1    0.02     0 trap      2
 5        0    15.6  55.1       5.14   3.3      27.6    0.98     0 elect…    5.4
 6        0    18.3 107.        0.57   0.85      1.1    0        0 trap      2.4
 7        0    18.5  81.5       2.29   1.26     22.8    0.94     0 elect…    5.2
 8        0    16.2 272.        3.43   0.56     27.2    0.95     1 elect…    3.4
```

```
 9         0     18     24.4        0.17   0.601    19.5    0.16      0 elect…    1.2
10         1     17.3   11.9        0.57   2.14      3.9    0.04      0 elect…    4.3
# … with 1,638 more rows, 2 more variables: .fitted <dbl>, model <chr>, and
```

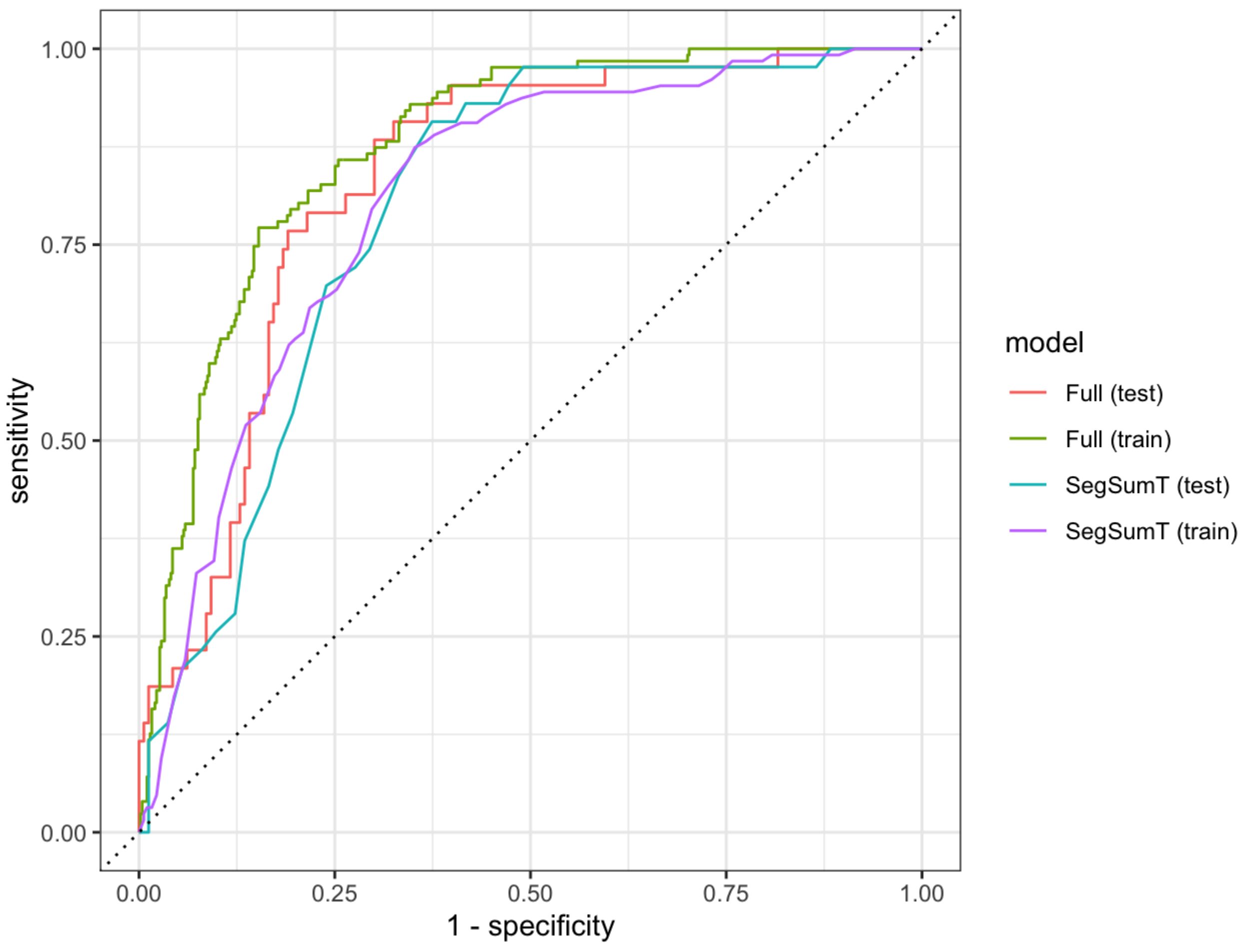
Separation

```r
model_comb |>
  ggplot(aes(x=.fitted, y=presence, color=as.factor(presence))) +
    geom_jitter(height=0.1, alpha=0.5) +
    facet_wrap(~model, ncol=2) +
    guides(color="none")
```



ROC

```
model_comb |>
  yardstick::roc_curve(factor(presence, levels = c(1,0)), .fitted) |>
  autoplot()
```



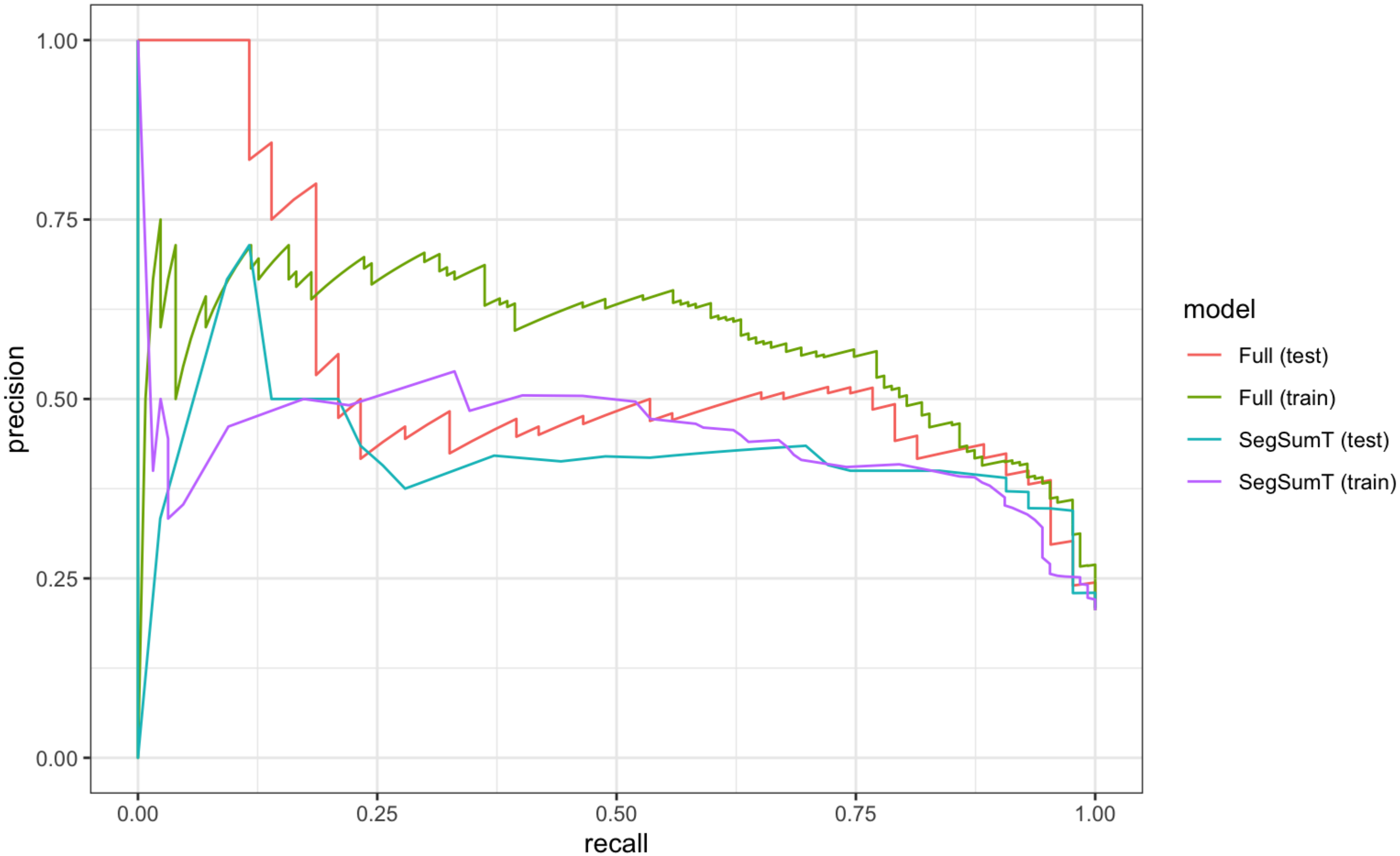
AUC

```r
model_comb |>
  yardstick::roc_auc(factor(presence, levels = c(1,0)), .fitted)
```

```
# A tibble: 4 × 4
  model            .metric .estimator .estimate
  <chr>            <chr>   <chr>          <dbl>
1 Full (test)      roc_auc binary         0.831
2 Full (train)     roc_auc binary         0.875
3 SegSumT (test)   roc_auc binary         0.796
4 SegSumT (train)  roc_auc binary         0.806
```

Precision / Recall

```
model_comb |>
  yardstick::pr_curve(factor(presence, levels = c(1,0)), .fitted) |>
  autoplot()
```



PR-AUC

```r
model_comb |>
  yardstick::pr_auc(factor(presence, levels = c(1,0)), .fitted)
```

```
# A tibble: 4 × 4
  model           .metric .estimator .estimate
  <chr>           <chr>   <chr>          <dbl>
1 Full (test)     pr_auc  binary         0.543
2 Full (train)    pr_auc  binary         0.583
3 SegSumT (test)  pr_auc  binary         0.422
4 SegSumT (train) pr_auc  binary         0.447
```

Aside: Species Distribution Modeling

Model Choice

We have been fitting a model that looks like the following,

$$y_i \sim \text{Bern}(p_i)$$

$$\text{logit}(p_i) = \boldsymbol{X}_{i\cdot}\boldsymbol{\beta}$$

Interpretation of y_i and p_i?

Absence of evidence …

If we observe a species at a particular location what does that tell us?

If we *don't* observe a species at a particular location what does that tell us?

Revised Model

If we allow for crypsis, then

$$y_i \sim \text{Bern}(q_i\, z_i)$$
$$z_i \sim \text{Bern}(p_i)$$

$$\text{logit}(q_i) = \boldsymbol{X}^{\star}_{i\cdot}\boldsymbol{\gamma}$$
$$\text{logit}(p_i) = \boldsymbol{X}_{i\cdot}\boldsymbol{\beta}$$

How should we interpret the parameters / variables: y_i, z_i, p_i, and q_i?

Bayesian Model

brms + logistic regression

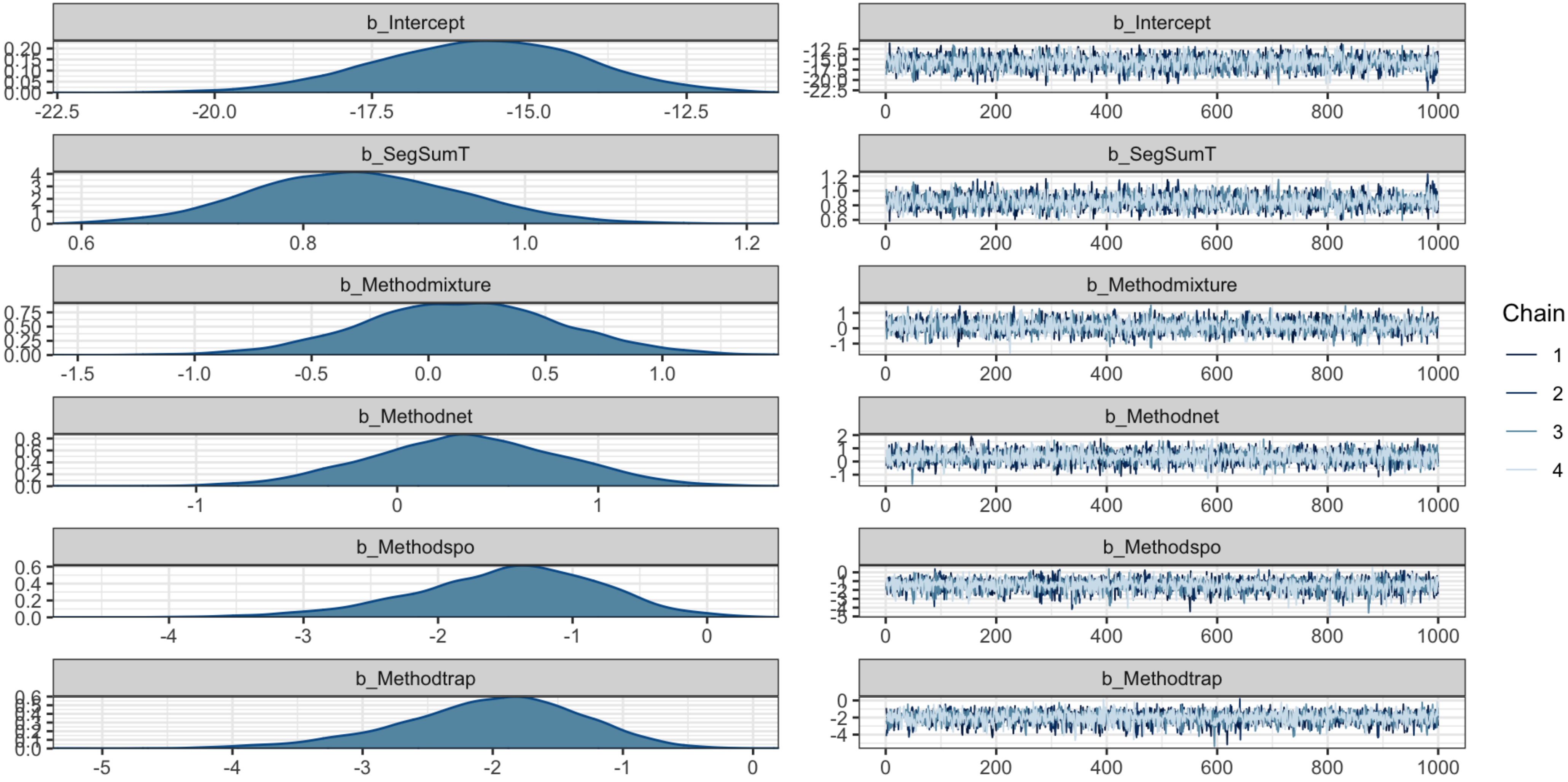
```r
( b = brms::brm(
    presence~SegSumT+Method, family="bernoulli",
    data=anguilla_train
) )
```

```
 Family: bernoulli
  Links: mu = logit
Formula: presence ~ SegSumT + Method
   Data: anguilla_train (Number of observations: 618)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Population-Level Effects:
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept       -15.78      1.65   -19.12   -12.65 1.00     3079     2834
SegSumT           0.85      0.09     0.67     1.04 1.00     3104     2900
Methodmixture     0.14      0.42    -0.69     0.97 1.00     4577     3059
Methodnet         0.35      0.48    -0.59     1.26 1.00     5001     3143
Methodspo        -1.48      0.70    -3.02    -0.24 1.00     4942     2587
Methodtrap       -2.02      0.70    -3.57    -0.79 1.00     4098     2657
```

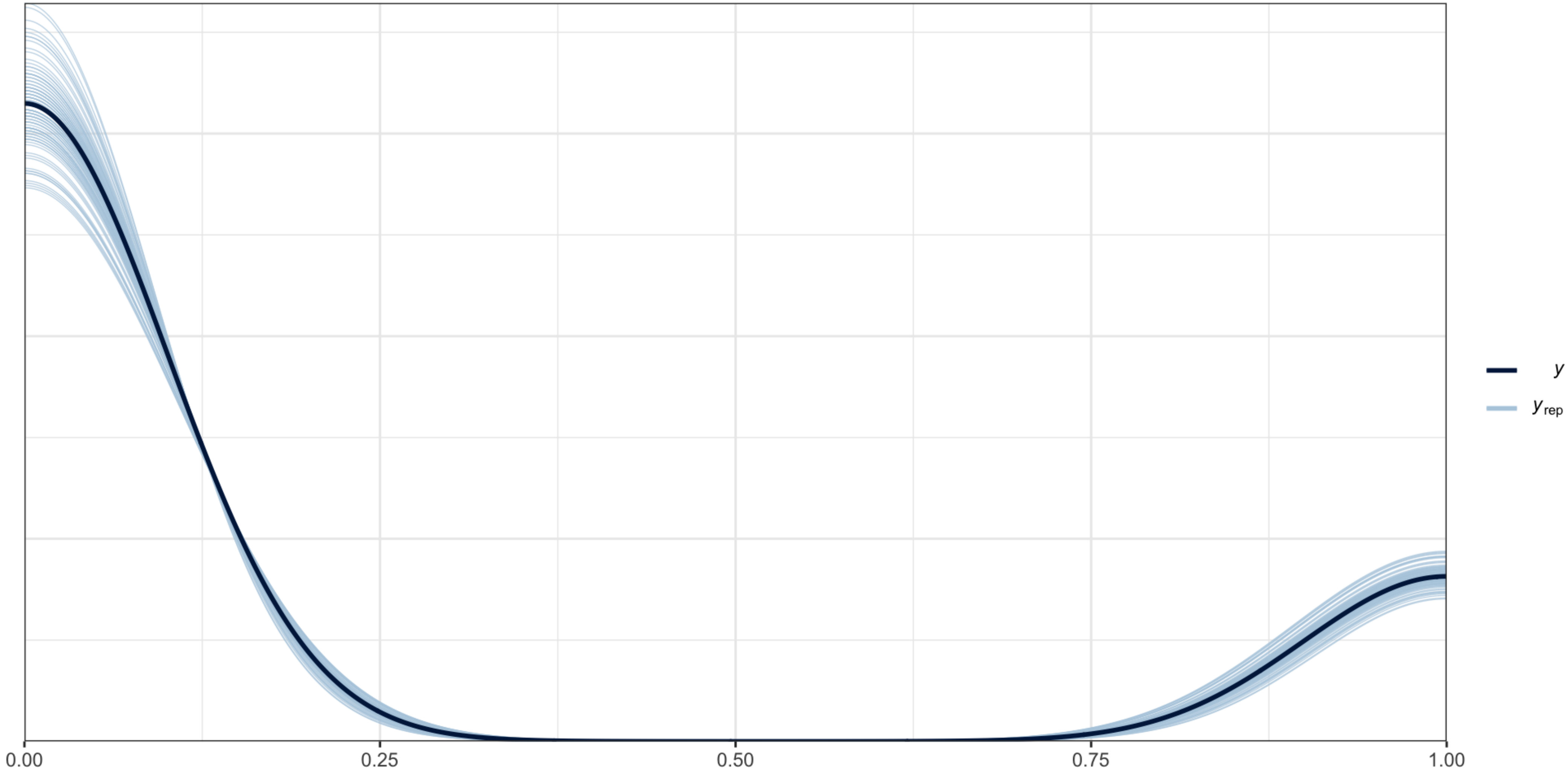
Diagnostics

```r
plot(b, N=6)
```



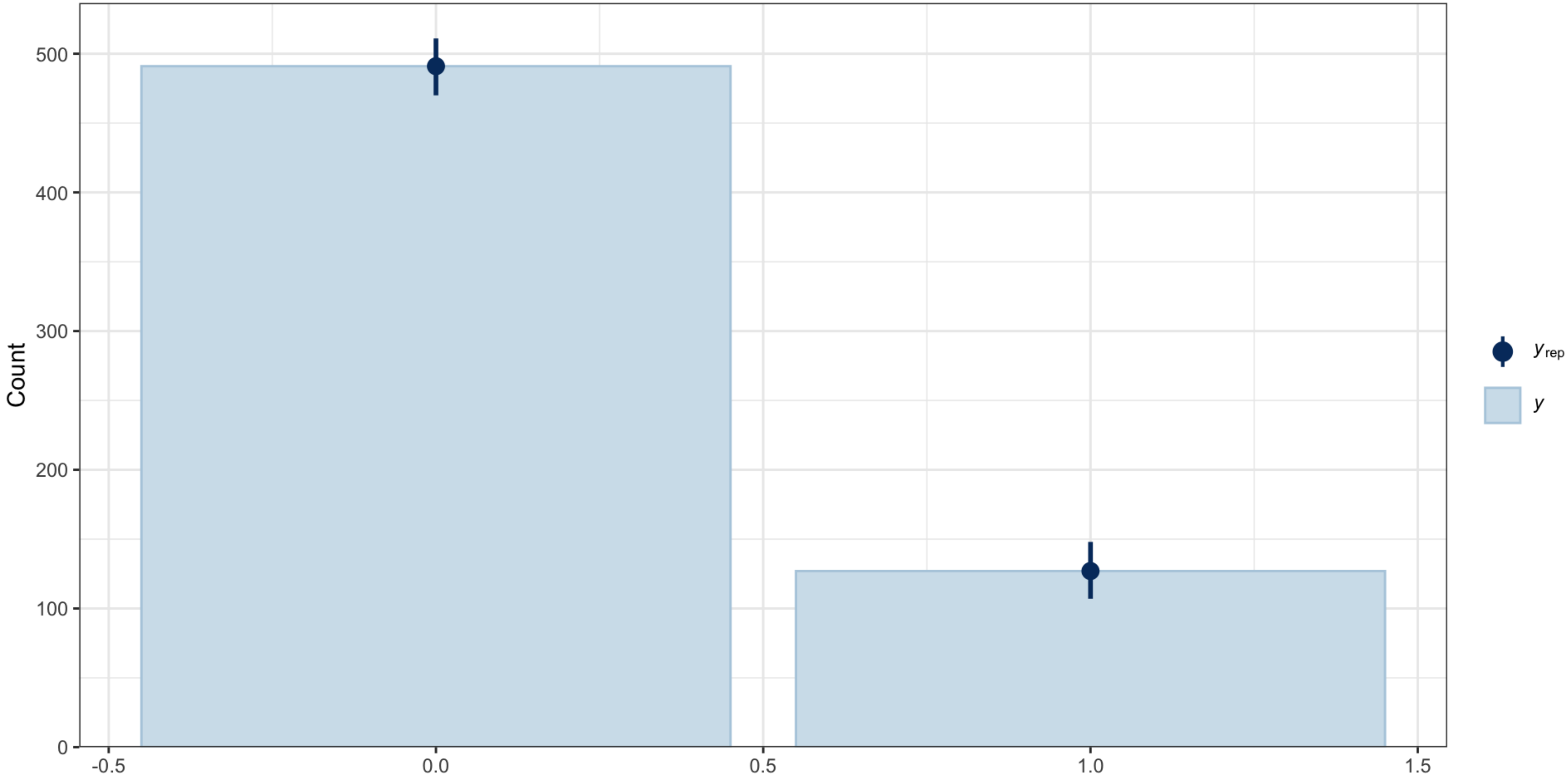
PP checks

```r
brms::pp_check(b, ndraws=100)
```



PP check - bars

```
brms::pp_check(b, type="bars", ndraws=1000)
```



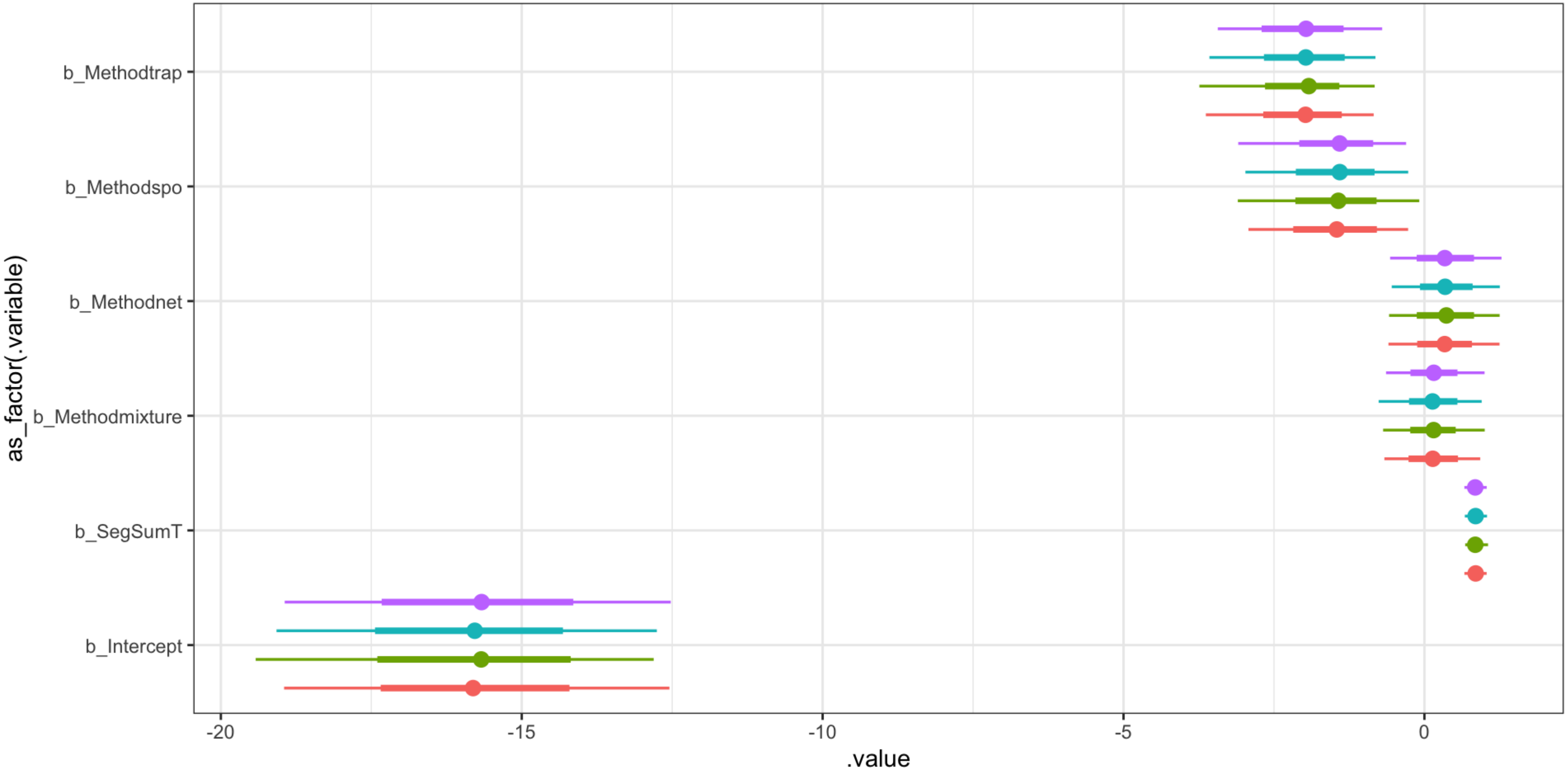
Gathering parameters

```
( b_param = b |>
    tidybayes::gather_draws( `b_.*`, regex = TRUE)
)
```

```
# A tibble: 24,000 × 5
# Groups:   .variable [6]
   .chain .iteration .draw .variable    .value
    <int>      <int> <int> <chr>         <dbl>
 1      1          1     1 b_Intercept  -17.3
 2      1          2     2 b_Intercept  -15.6
 3      1          3     3 b_Intercept  -16.9
 4      1          4     4 b_Intercept  -17.0
 5      1          5     5 b_Intercept  -14.4
 6      1          6     6 b_Intercept  -18.0
 7      1          7     7 b_Intercept  -11.1
 8      1          8     8 b_Intercept  -15.2
```

Caterpillar plot

```
b_param |>
  ggplot(aes(x=.value, y=as_factor(.variable), color=as.factor(.chain))) +
    tidybayes::stat_pointinterval(position = "dodge") +
    guides(color="none")
```



Posterior predictive

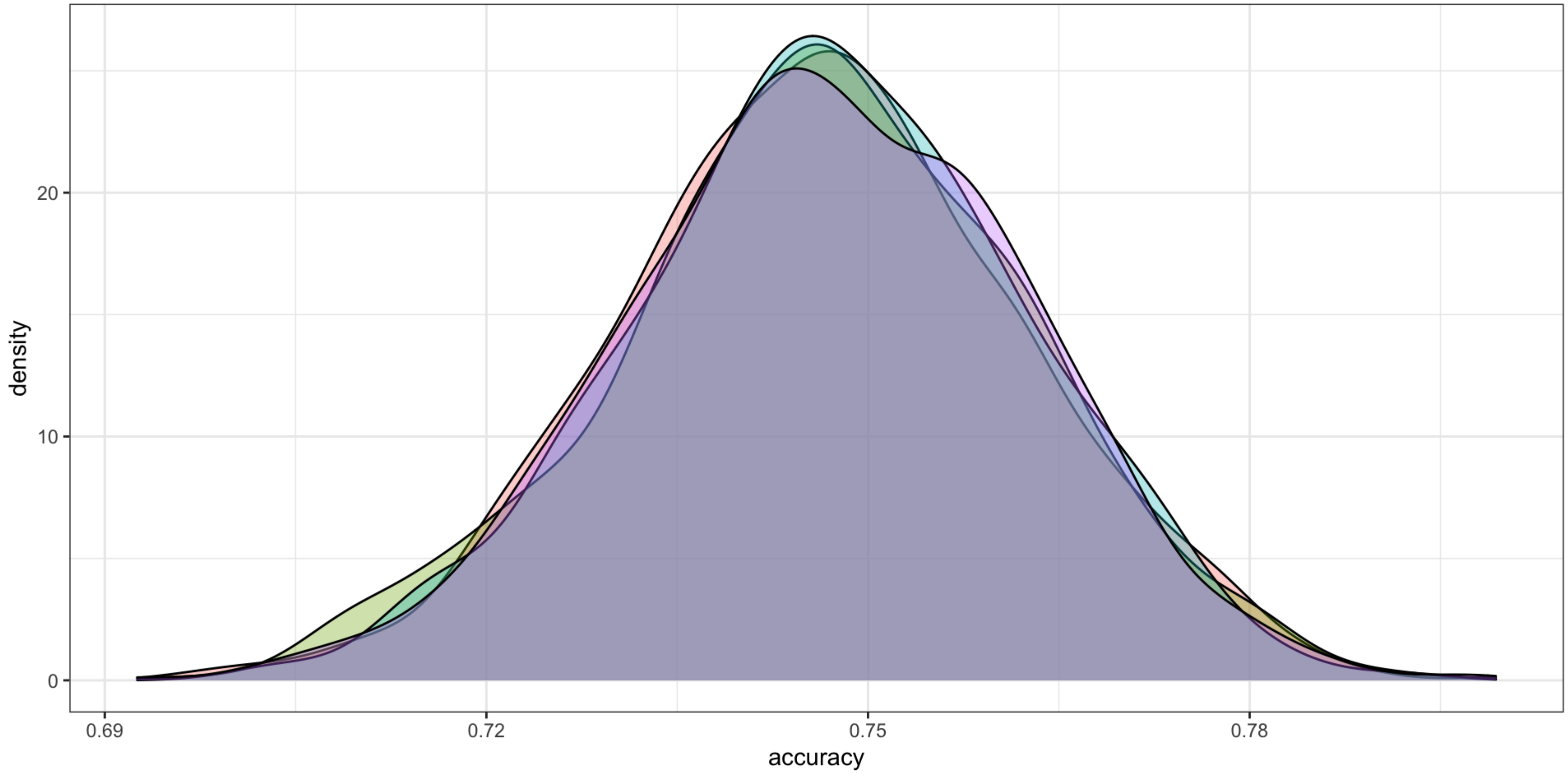
```r
( b_pred = b |>
    predicted_draws_fix(newdata = anguilla_train) |>
    select(presence, .row:.prediction) |>
    mutate( # Fix for yardstick
      presence = factor(presence, levels=c(1,0)),
      .prediction = factor(.prediction, levels=c(1,0))
    )
)
```

```
# A tibble: 2,472,000 × 6
   presence  .row .chain .iteration .draw .prediction
   <fct>    <int>  <int>      <int> <int> <fct>
 1 0            1      1          1     1 0
 2 0            1      1          2     2 0
 3 0            1      1          3     3 1
 4 0            1      1          4     4 0
 5 0            1      1          5     5 0
 6 0            1      1          6     6 0
 7 0            1      1          7     7 0
```

8 0 1 1 8 8 0

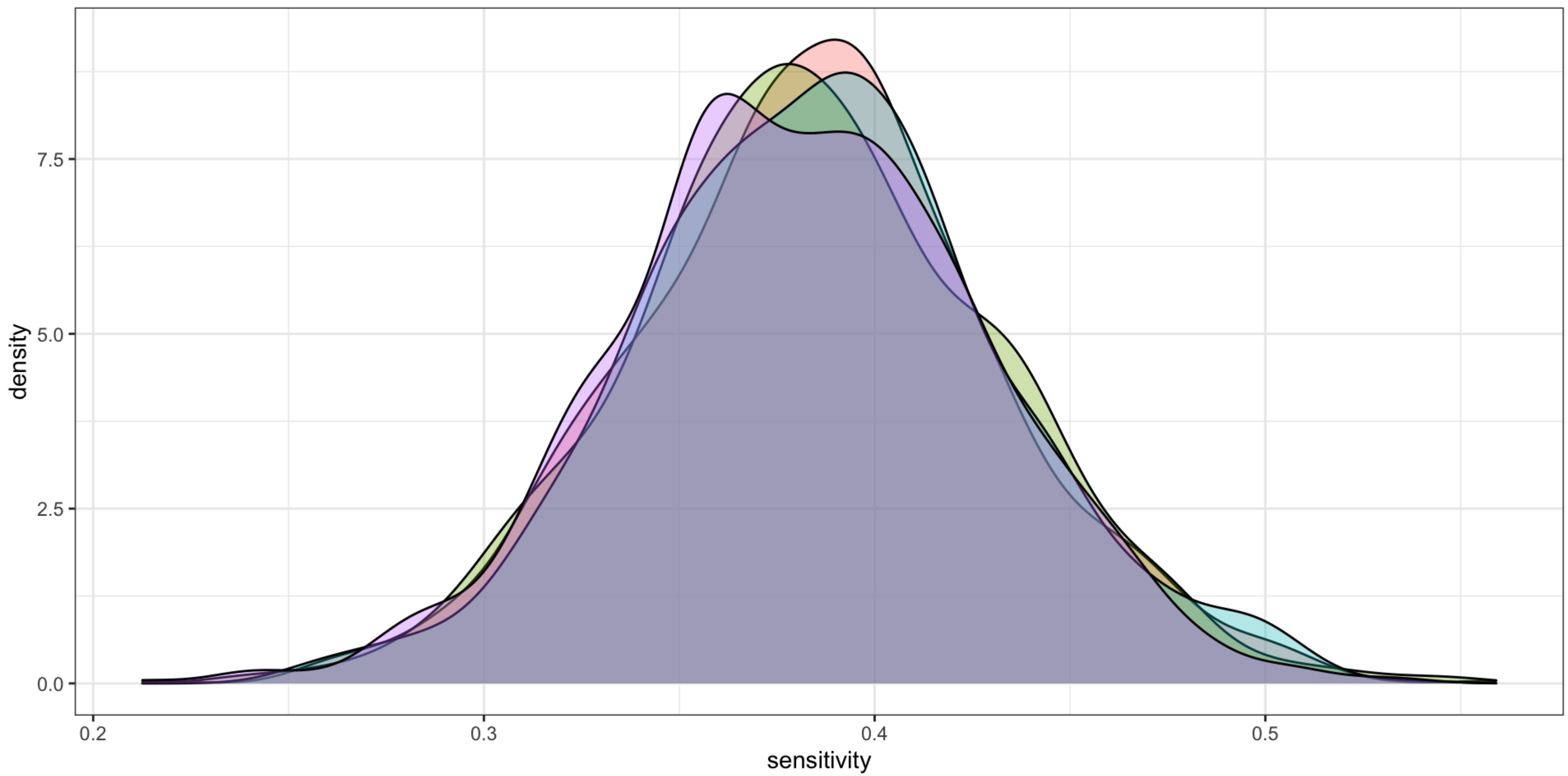
Posterior Accuracy

```
b_pred  |>
  group_by(.chain, .iteration) |>
  summarize(
    accuracy = yardstick::accuracy_vec(presence, .prediction)
  ) |>
  ggplot(aes(x = accuracy, fill = as.factor(.chain))) +
    geom_density(alpha=0.33) +
    guides(fill = "none")
```


density
20
10
0
0.69
0.72
0.75
0.78
accuracy

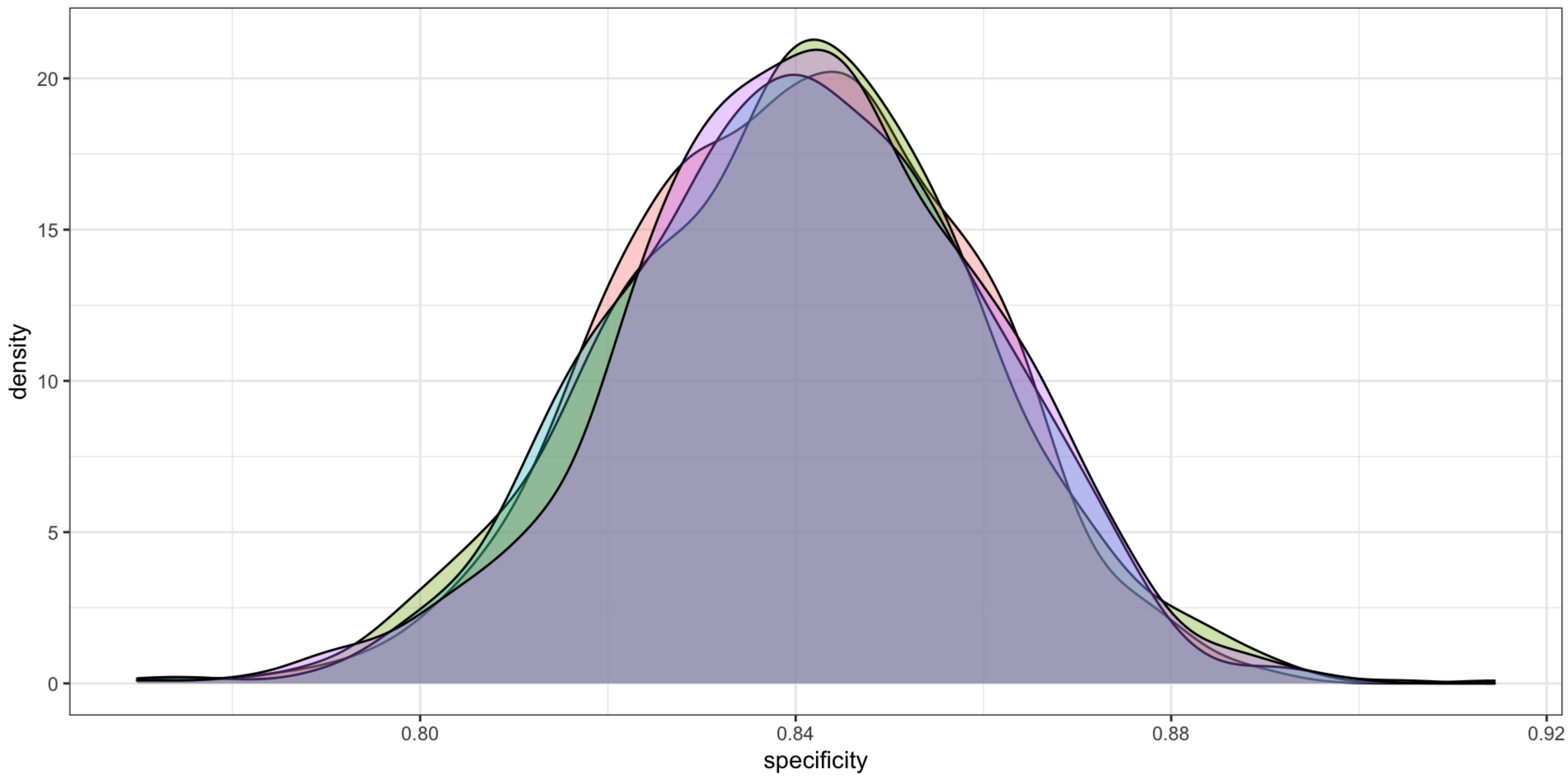
Posterior Sensitivity

```
b_pred  |>
  group_by(.chain, .iteration) |>
  summarize(
    sensitivity = yardstick::sensitivity_vec(presence, .prediction)
  ) |>
  ggplot(aes(x = sensitivity, fill = as.factor(.chain))) +
    geom_density(alpha=0.33) +
    guides(fill = "none")
```


density
7.5
5.0
2.5
0.0
0.2
0.3
0.4
0.5
sensitivity

Posterior Specificity

```r
b_pred  |>
  group_by(.chain, .iteration) |>
  summarize(
    specificity = yardstick::specificity_vec(presence, .prediction)
  ) |>
  ggplot(aes(x = specificity, fill = as.factor(.chain))) +
    geom_density(alpha=0.33) +
    guides(fill = "none")
```


density
20
15
10
5
0
0.80
0.84
0.88
0.92
specificity

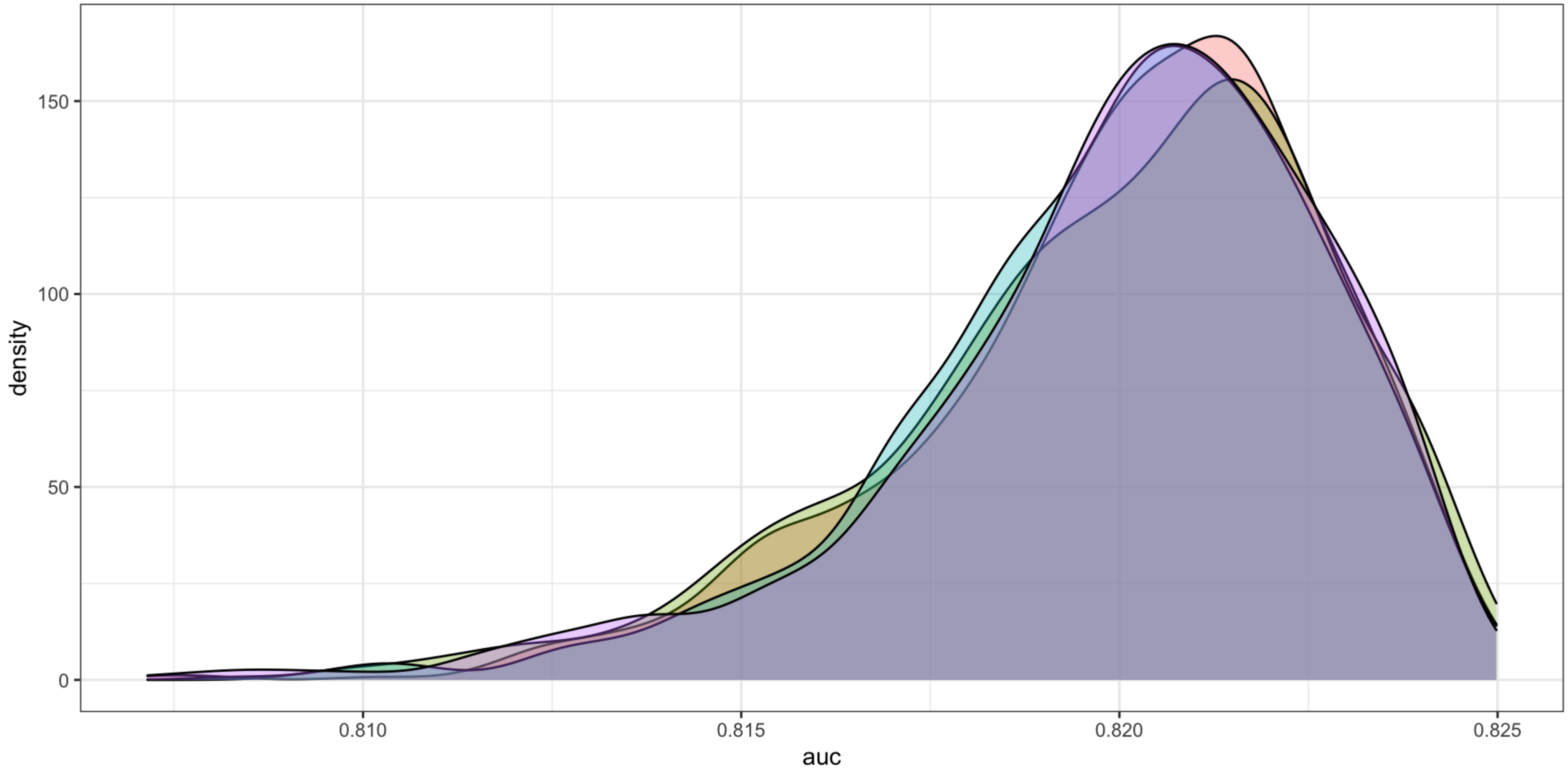
Expected posterior predictive

```
( b_epred = b |>
    epred_draws_fix(newdata = anguilla_train) |>
    select(presence, .row:.epred) |>
    mutate( # Fix for yardstick
      presence = factor(presence, levels=c(1,0))
    )
)
```

```
# A tibble: 2,472,000 × 6
   presence  .row .chain .iteration .draw .epred
   <fct>    <int>  <int>      <int> <int>  <dbl>
 1 0            1      1          1     1  0.119
 2 0            1      1          2     2  0.143
 3 0            1      1          3     3  0.126
 4 0            1      1          4     4  0.127
 5 0            1      1          5     5  0.142
 6 0            1      1          6     6  0.126
 7 0            1      1          7     7  0.175
 8 0            1      1          8     8  0.160
```

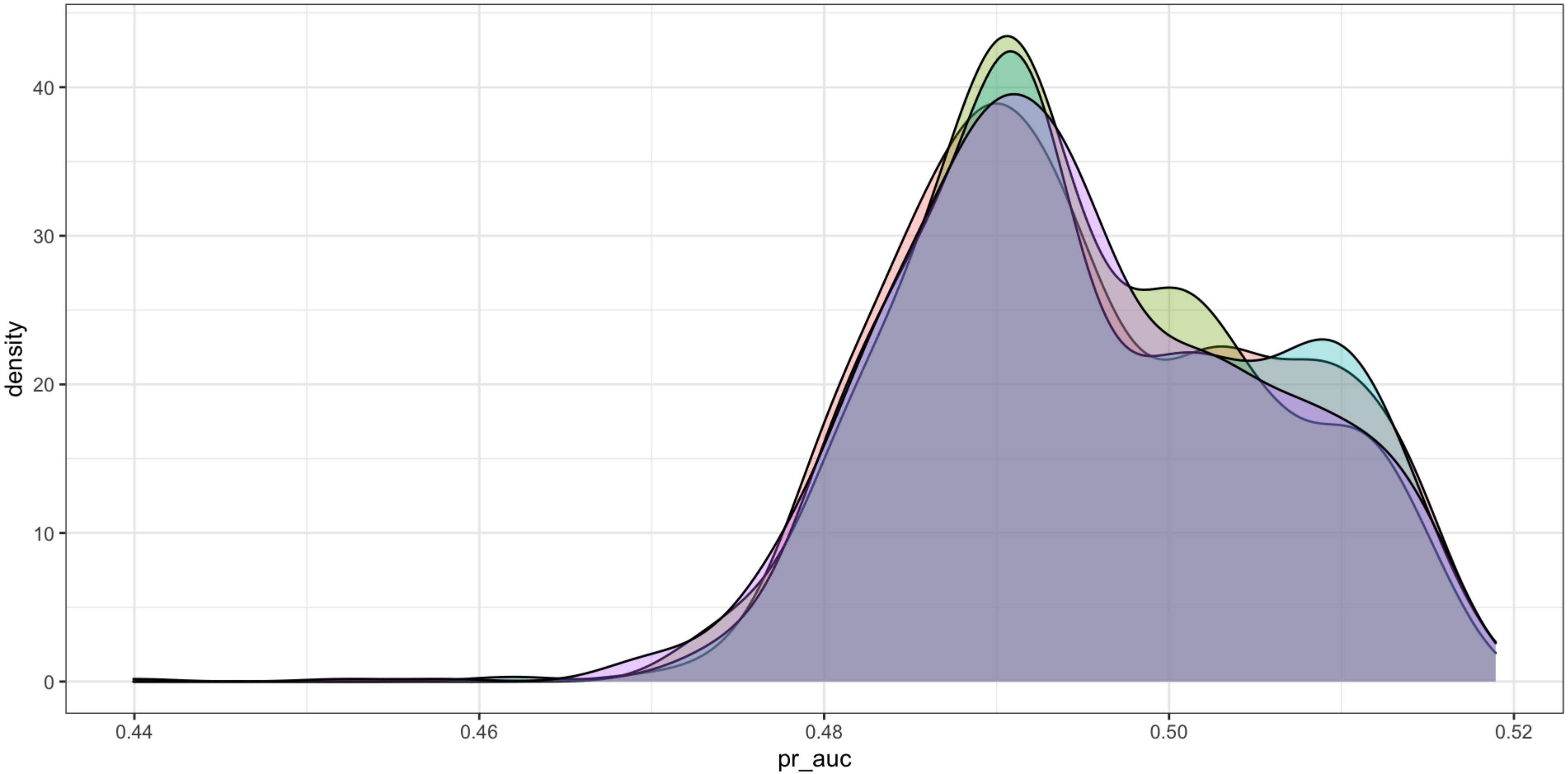
Posterior AUC

```
b_epred  |>
  group_by(.chain, .iteration) |>
  summarize(
    auc = yardstick::roc_auc_vec(presence, .epred)
  ) |>
  ggplot(aes(x = auc, fill = as.factor(.chain))) +
    geom_density(alpha=0.33) +
    guides(fill = "none")
```


150
100
50
0
density
0.810
0.815
0.820
0.825
auc

Posterior PR-AUC

```
b_epred  |>
  group_by(.chain, .iteration) |>
  summarize(
    pr_auc = yardstick::pr_auc_vec(presence, .epred)
  ) |>
  ggplot(aes(x = pr_auc, fill = as.factor(.chain))) +
    geom_density(alpha=0.33) +
    guides(fill = "none")
```


density
0
10
20
30
40
0.44
0.46
0.48
0.50
0.52
pr_auc

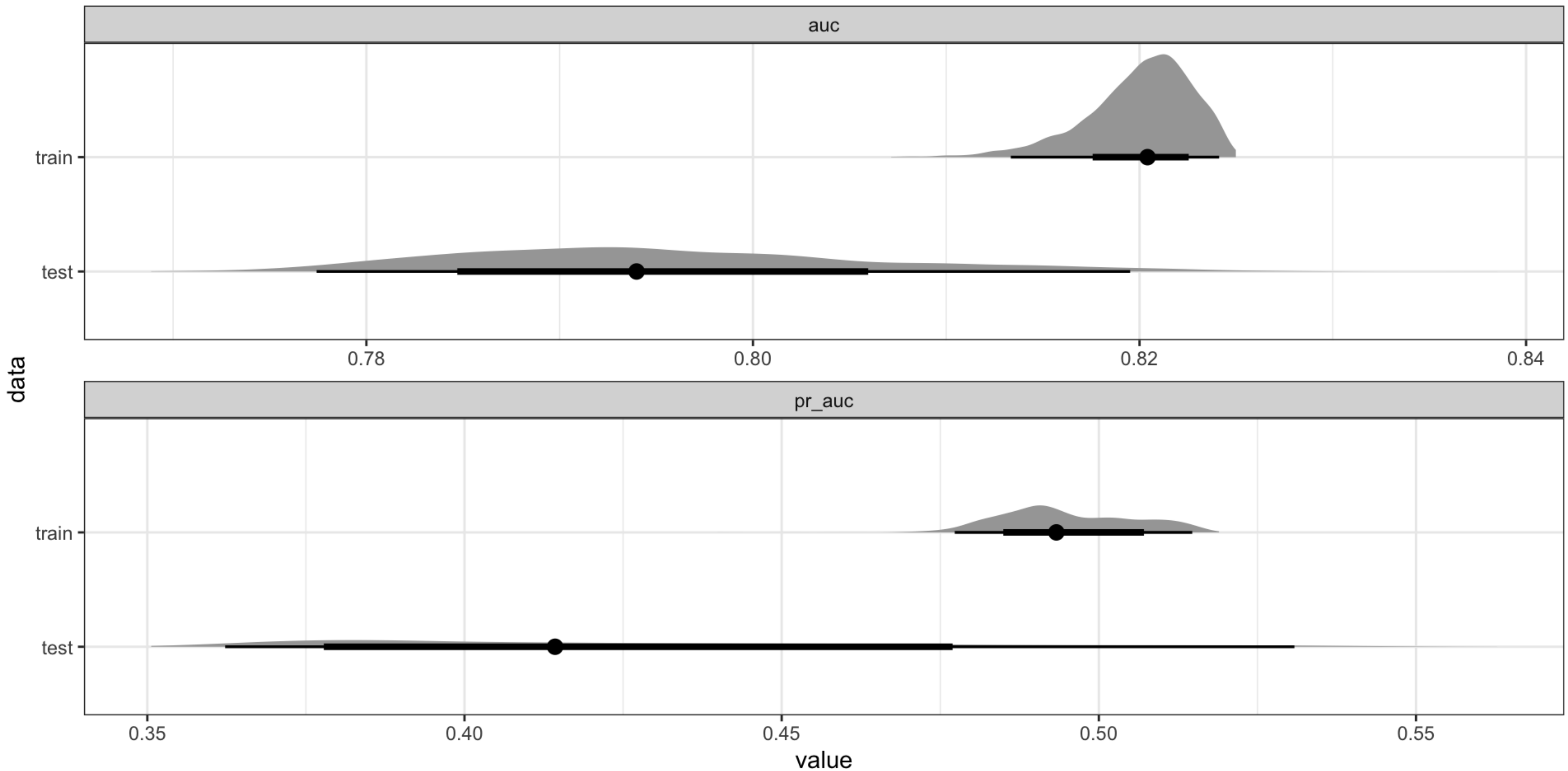
Expected posterior predictive - test

```
b_epred_test = b |>
  epred_draws_fix(newdata = anguilla_test) |>
  select(presence, .row:.epred) |>
  mutate( # Fix for yardstick
    presence = factor(presence, levels=c(1,0))
  )
```

```
b_comb = bind_rows(
  b_epred |> mutate(data = "train"),
  b_epred_test |> mutate(data = "test")
)
```

Comparing AUC / PR-AUC

```r
b_comb  |>
  group_by(.chain, .iteration, data) |>
  summarize(
    auc = yardstick::roc_auc_vec(presence, .epred),
    pr_auc = yardstick::pr_auc_vec(presence, .epred)
  ) |>
  pivot_longer(cols = auc:pr_auc, names_to = "stat", values_to = "value") |>
  ggplot(aes(x = value, y=data)) +
    tidybayes::stat_halfeye() +
    facet_wrap(~stat, ncol=1, scales = "free_x")
```


auc
train
test
0.78
0.80
0.82
0.84
data
pr_auc
train
test
0.35
0.40
0.45
0.50
0.55
value