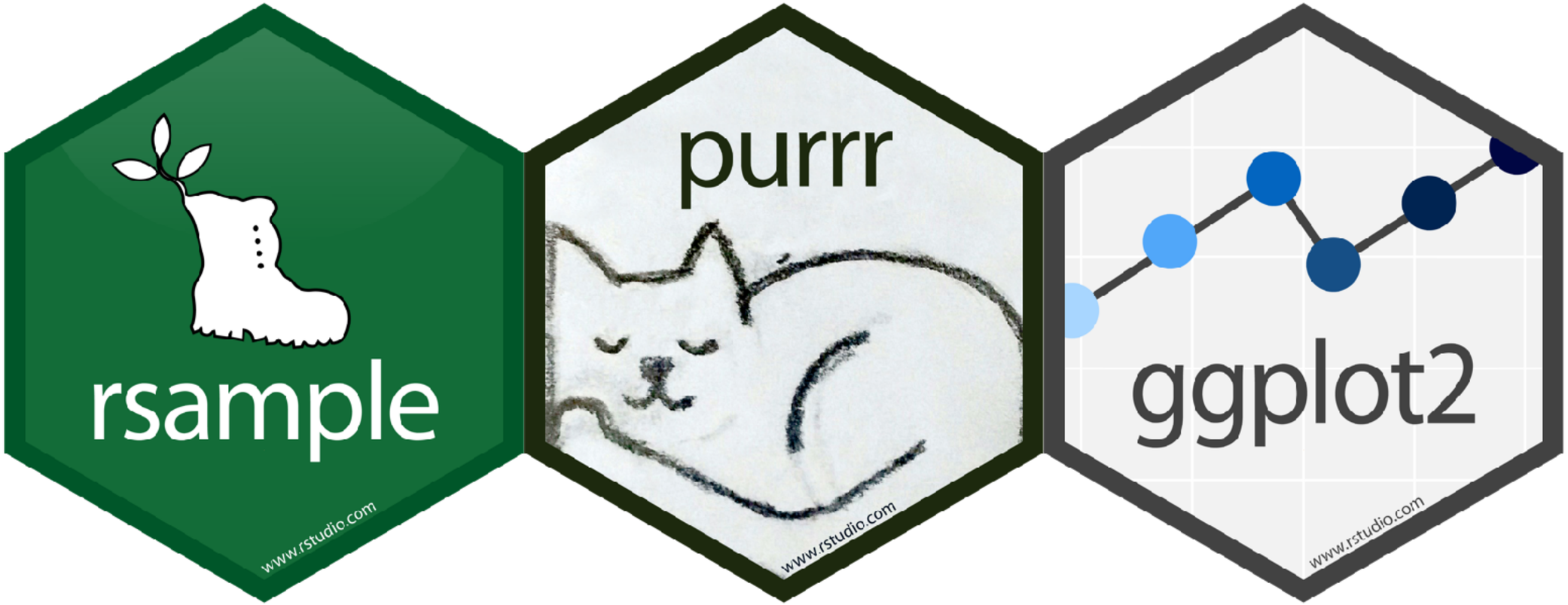
Case Study

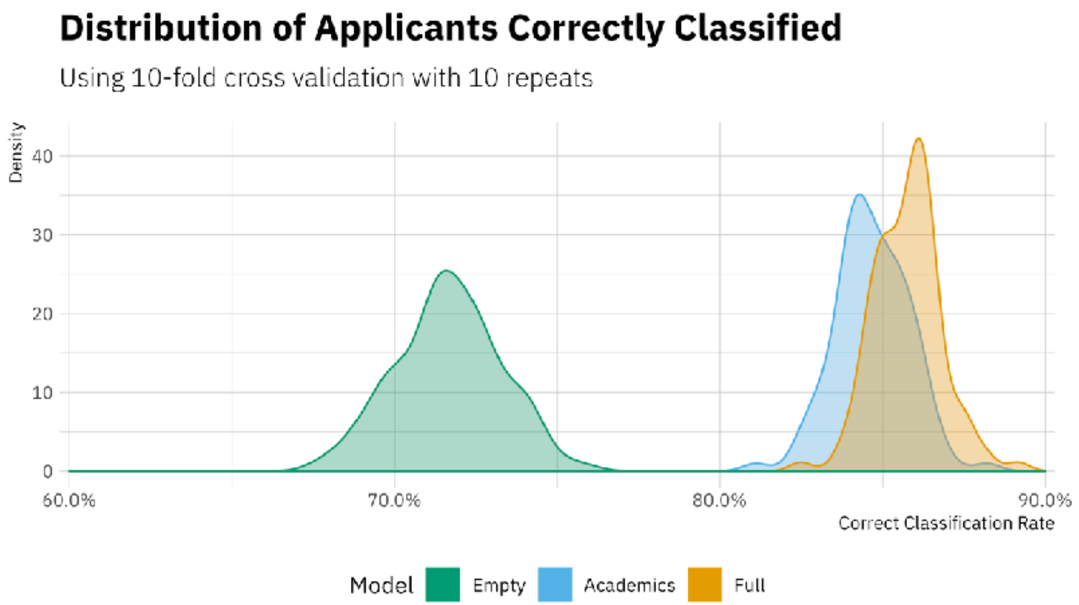
Jake Thompson
wjakethompson.com
@wjakethompson

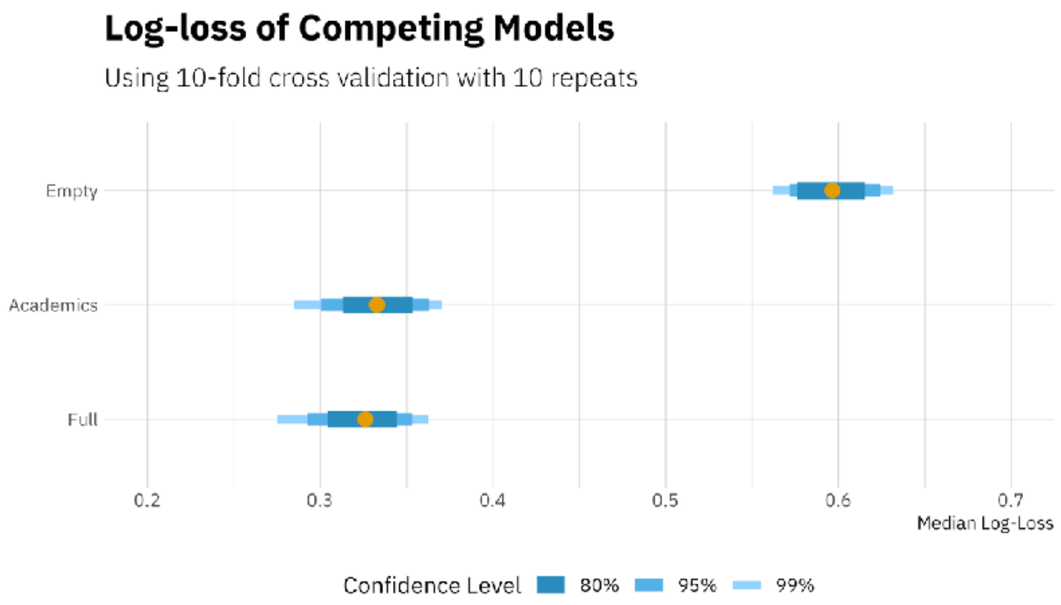
ATLAS
Accessible Teaching, Learning, and Assessment Systems


rsample
www.rstudio.com
purrr
www.rstudio.com
ggplot2
www.rstudio.com

Your Turn 0

- Open **10-Case-Study-2.Rmd**
- Run the setup chunk

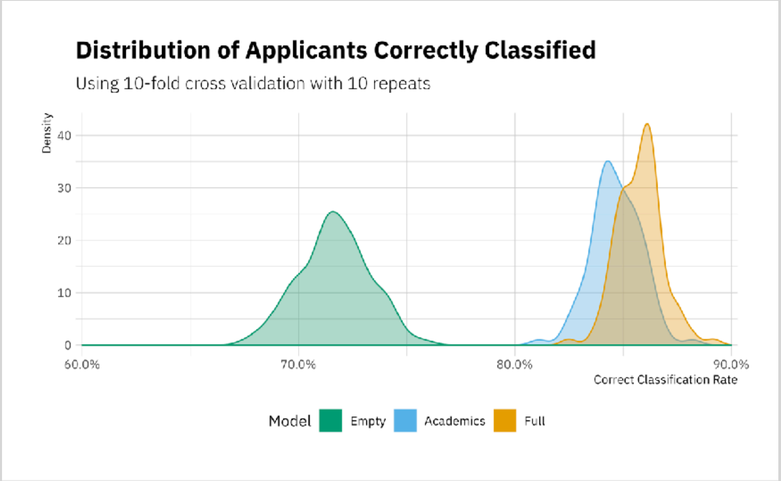


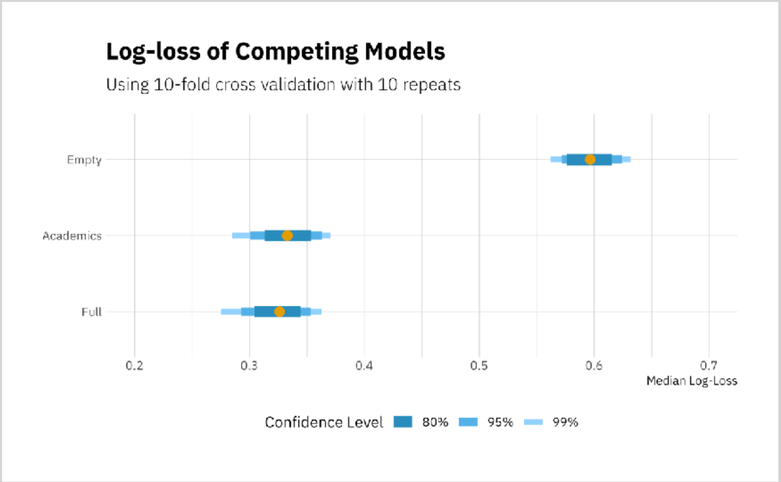


Can we replicate these plots?

Your Turn 1

- Take a look at **admission**
- With your neighbor, brainstorm the steps needed to get the data in a form ready to make the plots.

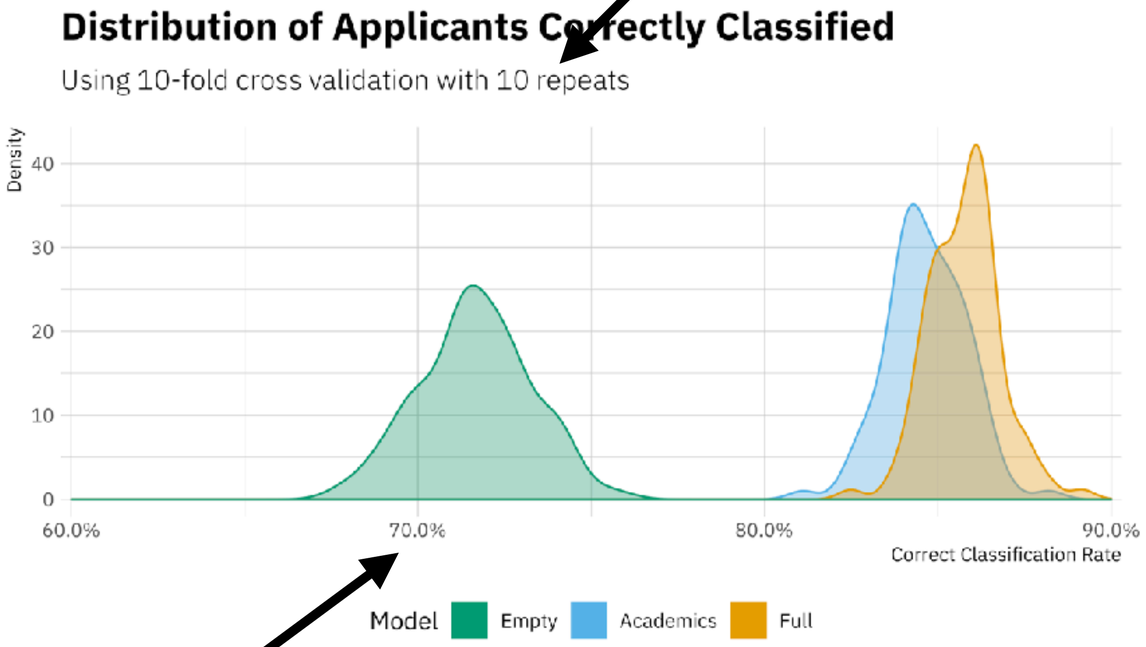




05 : 00

ATLAS

`vfold_cv()`



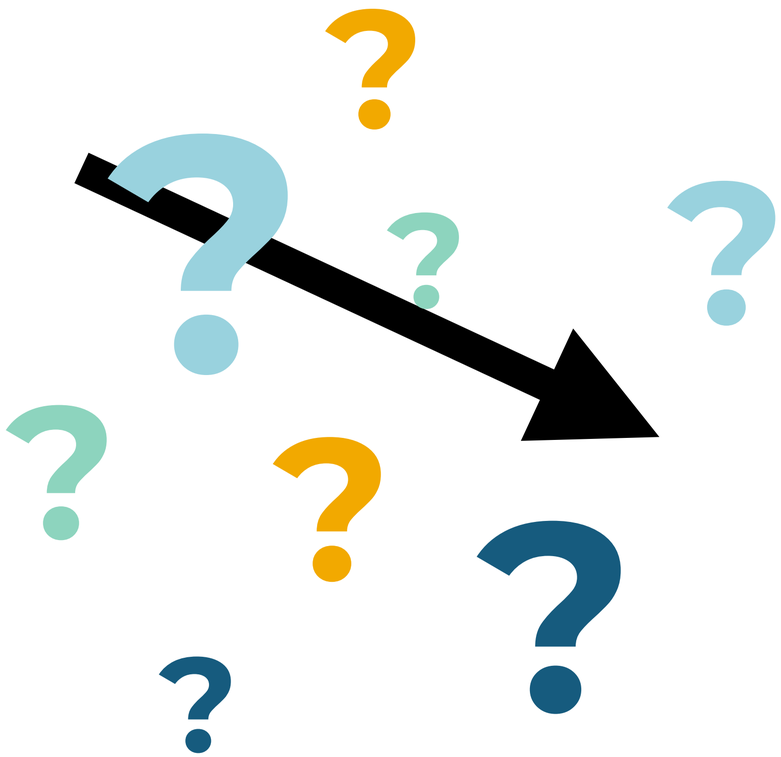
percent correctly classified

Data required to make the plot

id	id2	model	pct_cor
Repeat01	Fold01	empty	0.72
Repeat01	Fold01	academics	0.85
Repeat01	Fold01	full	0.89
Repeat01	Fold02	empty	0.68
Repeat01	Fold02	academics	0.83
...	...	...	...

Start

```
# A tibble: 9,416 x 6
   admit gre_v gre_q gre_w   gpa gender
   <int> <int> <int> <dbl> <dbl> <fct>
 1     0   142   151   2.5  3.07 Female
 2     0   148   140   2.5  3.42 Male
 3     1   156   147   2.5  3.48 Male
 4     0   154   152   4    3.3  Male
     ...
```



```
# A tibble: 300 x 4
   id       id2    model     pct_cor
   <chr>    <chr>  <chr>       <dbl>
 1 Repeat01 Fold01 acadm_mod   0.838
 2 Repeat01 Fold01 compl_mod   0.849
 3 Repeat01 Fold01 empty_mod   0.708
 4 Repeat01 Fold02 acadm_mod   0.843
 5 Repeat01 Fold02 compl_mod   0.865
 6 Repeat01 Fold02 empty_mod   0.711
 7 Repeat01 Fold03 acadm_mod   0.849
 8 Repeat01 Fold03 compl_mod   0.862
 9 Repeat01 Fold03 empty_mod   0.723
10 Repeat01 Fold04 acadm_mod   0.847
# … with 290 more rows
```

One such process

- Create cross validation sets
- Fit all three models to each analysis set
- Get predictions for each assessment set
- Calculate outcome measures for each set of predictions
- Plot the distributions of the outcome measures

Your Turn 2

- Create a cross validation resampling with 10 folds and 10 repeats

```
set.seed(32011)
models <- admission %>%
  vfold_cv(v = 10, repeats = 10)
models
# A tibble: 100 x 3
   splits             id       id2
   <list>             <chr>    <chr>
 1 <split [8.5K/942]> Repeat01 Fold01
 2 <split [8.5K/942]> Repeat01 Fold02
 3 <split [8.5K/942]> Repeat01 Fold03
 4 <split [8.5K/942]> Repeat01 Fold04
 5 <split [8.5K/942]> Repeat01 Fold05
 6 <split [8.5K/942]> Repeat01 Fold06
 7 <split [8.5K/941]> Repeat01 Fold07
 8 <split [8.5K/941]> Repeat01 Fold08
 9 <split [8.5K/941]> Repeat01 Fold09
10 <split [8.5K/941]> Repeat01 Fold10
# … with 90 more rows
```

Your Turn 3

- Write a function that takes a **splits** and a **formula** and returns predictions
- The model should be fit using the **analysis** data
- Predictions should be made on the **assessment** data
- Use **mutate** to add to the predictions
  - Predicted acceptance is 1 if **.fitted** is greater than 0.5, 0 otherwise
  - Prediction is correct if the predicted value (from above) is the same as **admit**

05 : 00

```r
holdout_results <- function(splits, formula) {
  # Fit the model to the analysis set
  mod <- glm(formula, data = analysis(splits), family = binomial)

  # Save the assessment data
  holdout <- assessment(splits)

  # `augment` will save the predictions with the holdout data set
  res <- broom::augment(mod, newdata = holdout, type.predict = "response") %>%
    mutate(prediction = ifelse(.fitted > 0.5, 1L, 0L),
           correct = prediction == admit)

  # Return the assessment data set with the additional columns
  res
}
```

Your Turn 4

- Use **mutate** and **map** to use the **holdout_results** function to fit each model to the cross validation sets
- Formulas for each model have already been saved for your convenience

05 : 00

```
empty <- as.formula(admit ~ 1)
academics <- as.formula(admit ~ gre_v * gre_q + gre_w + gpa)
full <- as.formula(admit ~ gre_v * gre_q + gre_w + gpa + gender)

models %>%
  mutate(empty_mod = map(splits, holdout_results, formula = empty),
         acadm_mod = map(splits, holdout_results, formula = academics),
         compl_mod = map(splits, holdout_results, formula = full))
# A tibble: 100 x 6
   splits             id       id2    empty_mod           acadm_mod           compl_mod
 * <list>             <chr>    <chr>  <list>              <list>              <list>
 1 <split [8.5K/942]> Repeat01 Fold01 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 2 <split [8.5K/942]> Repeat01 Fold02 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 3 <split [8.5K/942]> Repeat01 Fold03 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 4 <split [8.5K/942]> Repeat01 Fold04 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 5 <split [8.5K/942]> Repeat01 Fold05 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 6 <split [8.5K/942]> Repeat01 Fold06 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 7 <split [8.5K/941]> Repeat01 Fold07 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
 8 <split [8.5K/941]> Repeat01 Fold08 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
 9 <split [8.5K/941]> Repeat01 Fold09 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
10 <split [8.5K/941]> Repeat01 Fold10 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
# … with 90 more rows
```

```
# A tibble: 100 x 6
   splits             id       id2    empty_mod           acadm_mod           compl_mod
 * <list>             <chr>    <chr>  <list>              <list>              <list>
 1 <split [8.5K/942]> Repeat01 Fold01 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 2 <split [8.5K/942]> Repeat01 Fold02 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 3 <split [8.5K/942]> Repeat01 Fold03 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 4 <split [8.5K/942]> Repeat01 Fold04 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 5 <split [8.5K/942]> Repeat01 Fold05 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 6 <split [8.5K/942]> Repeat01 Fold06 <tibble [942 × 10]> <tibble [942 × 10]> <tibble [942 × 10]>
 7 <split [8.5K/941]> Repeat01 Fold07 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
 8 <split [8.5K/941]> Repeat01 Fold08 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
 9 <split [8.5K/941]> Repeat01 Fold09 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
10 <split [8.5K/941]> Repeat01 Fold10 <tibble [941 × 10]> <tibble [941 × 10]> <tibble [941 × 10]>
# … with 90 more rows
```

```
# A tibble: 300 x 4
   id       id2    model      pct_cor
   <chr>    <chr>  <chr>        <dbl>
 1 Repeat01 Fold01 acadm_mod    0.838
 2 Repeat01 Fold01 compl_mod    0.849
 3 Repeat01 Fold01 empty_mod    0.708
 4 Repeat01 Fold02 acadm_mod    0.843
 5 Repeat01 Fold02 compl_mod    0.865
 6 Repeat01 Fold02 empty_mod    0.711
 7 Repeat01 Fold03 acadm_mod    0.849
 8 Repeat01 Fold03 compl_mod    0.862
 9 Repeat01 Fold03 empty_mod    0.723
10 Repeat01 Fold04 acadm_mod    0.847
# … with 290 more rows
```

Getting close!

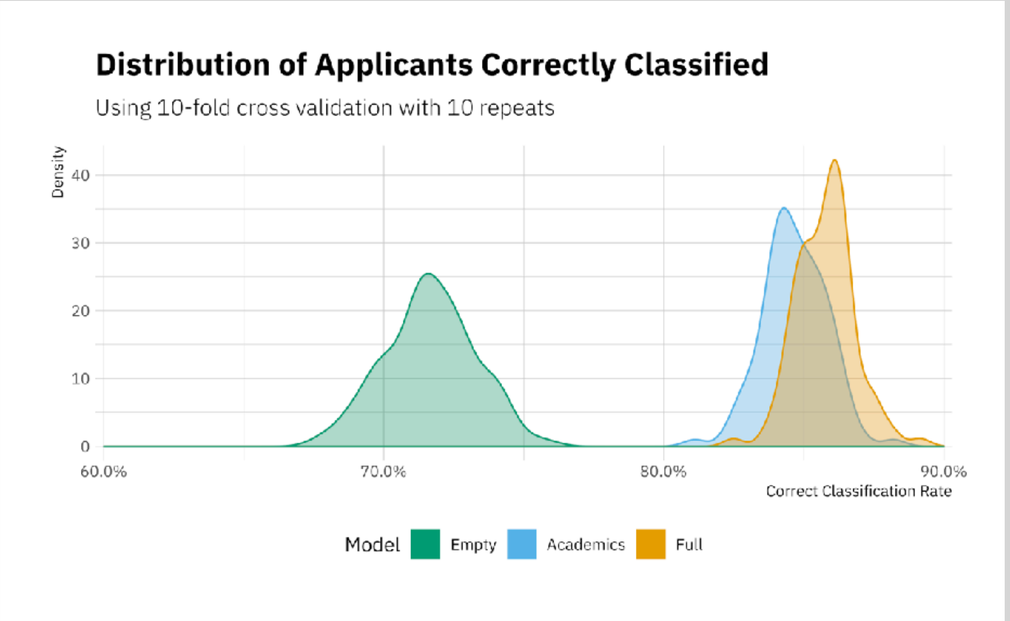
Your Turn 5

- Tidy the data so that the models are all in one column (**results**) with an identifier column (**model**)
- Expand the **results** so we can do calculations on the predictions

```
all_mods %>%
  select(-splits) %>%
  pivot_longer(contains("mod"), names_to = "model", values_to = "results") %>%
  unnest(results)
# A tibble: 282,480 x 13
   id       id2    model     admit gre_v gre_q gre_w   gpa gender .fitted .se.fit prediction correct
   <chr>    <chr>  <chr>     <int> <int> <int> <dbl> <dbl> <fct>    <dbl>   <dbl>      <int> <lgl>
 1 Repeat01 Fold01 empty_mod     1   160   164   4    2.87 Male     0.283 0.00489          0 FALSE
 2 Repeat01 Fold01 empty_mod     0   132   145   2    2.47 Male     0.283 0.00489          0 TRUE
 3 Repeat01 Fold01 empty_mod     0   151   144   4    2.54 Male     0.283 0.00489          0 TRUE
 4 Repeat01 Fold01 empty_mod     0   156   149   4.5  2.48 Female   0.283 0.00489          0 TRUE
 5 Repeat01 Fold01 empty_mod     1   145   143   2.5  3.08 Male     0.283 0.00489          0 FALSE
 6 Repeat01 Fold01 empty_mod     1   170   161   3.5  3.13 Male     0.283 0.00489          0 FALSE
 7 Repeat01 Fold01 empty_mod     0   153   153   3.5  3.2  Female   0.283 0.00489          0 TRUE
 8 Repeat01 Fold01 empty_mod     0   157   152   3.5  2.98 Male     0.283 0.00489          0 TRUE
 9 Repeat01 Fold01 empty_mod     0   142   143   3.5  3    Male     0.283 0.00489          0 TRUE
10 Repeat01 Fold01 empty_mod     0   158   145   5    3.21 Female   0.283 0.00489          0 TRUE
# … with 282,470 more rows
```

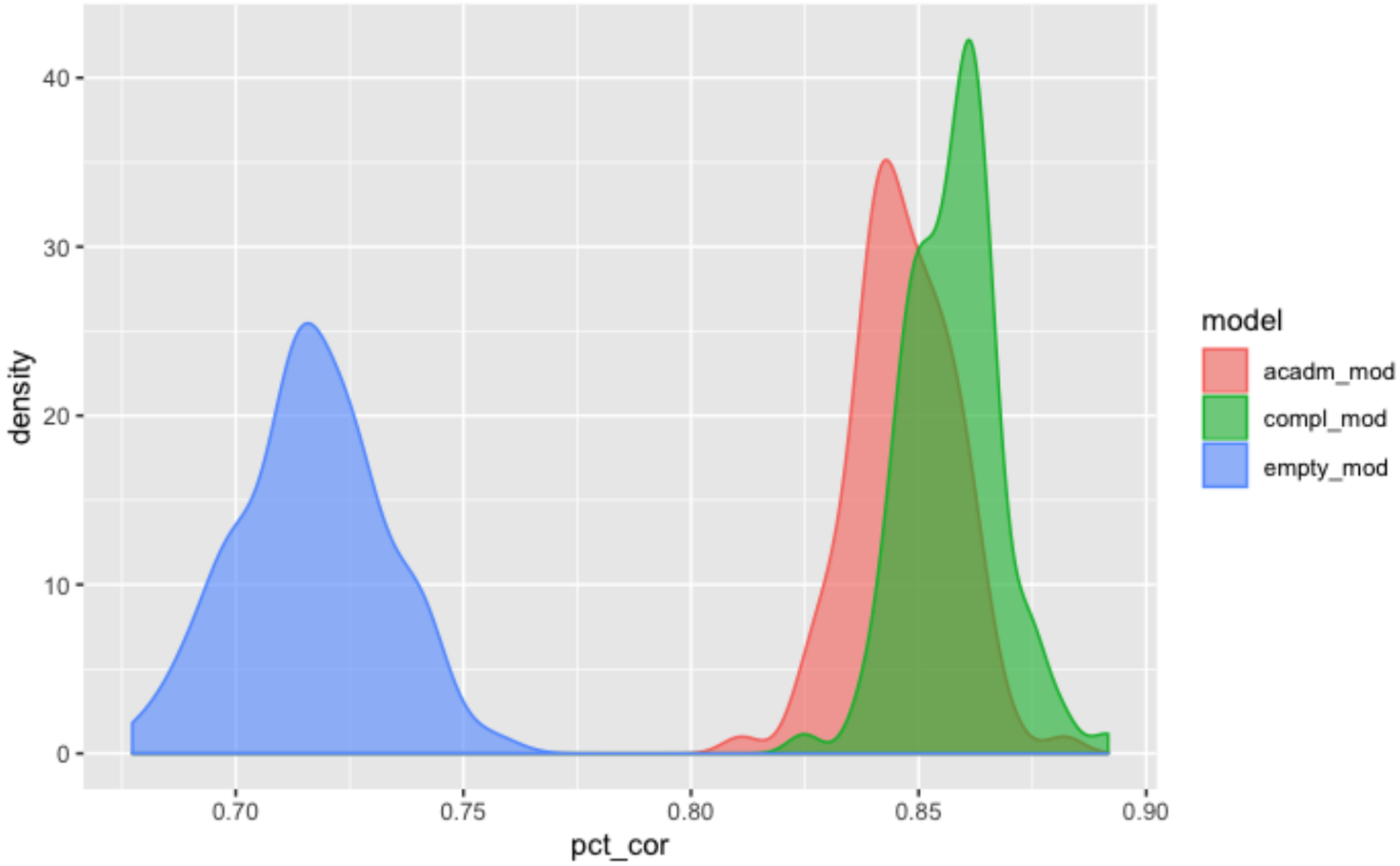
Your Turn 6

- Calculate the percent of applicants correctly classified for each **repeat**, **fold**, and **model**
- Plot the distributions



05 : 00

```
all_preds %>%
  group_by(id, id2, model) %>%
  summarize(pct_cor = mean(correct)) %>%
  ggplot(aes(x = pct_cor)) +
    geom_density(aes(fill = model, color = model), alpha = 0.6)
```



Consider

- What are some limitations of percent correct classification?

Model Evaluation

Applicant	admit	.fitted	prediction	correct
A	1	0.52	1	TRUE
B	1	0.97	1	TRUE

Are these predictions really equivalent?

Model Evaluation

Applicant	admit	.fitted	prediction	correct
A	1	0.52	1	TRUE
B	1	0.97	1	TRUE
C	1	0.48	0	FALSE

Is the prediction for Applicant C really that much different from Applicant A?

Is there a better way?

Log Loss

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Total number of cases

Observed outcome

Predicted outcome

Log Loss

When $\mathbf{y_i}$ is 1:

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

$$[1 \log(\hat{y}_i) + (1 - 1)\log(1 - \hat{y}_i)]$$

$$[\log(\hat{y}_i)]$$

Log Loss

When $\mathbf{y_i}$ is 0:

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

$$[0 \log(\hat{y}_i) + (1 - 0)\log(1 - \hat{y}_i)]$$

$$[\log(1 - \hat{y}_i)]$$

Log Loss

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Take the average of the individual penalties, and multiply by -1

For each observation, $\mathbf{y_i}$, calculate the **log** of the probability of that outcome

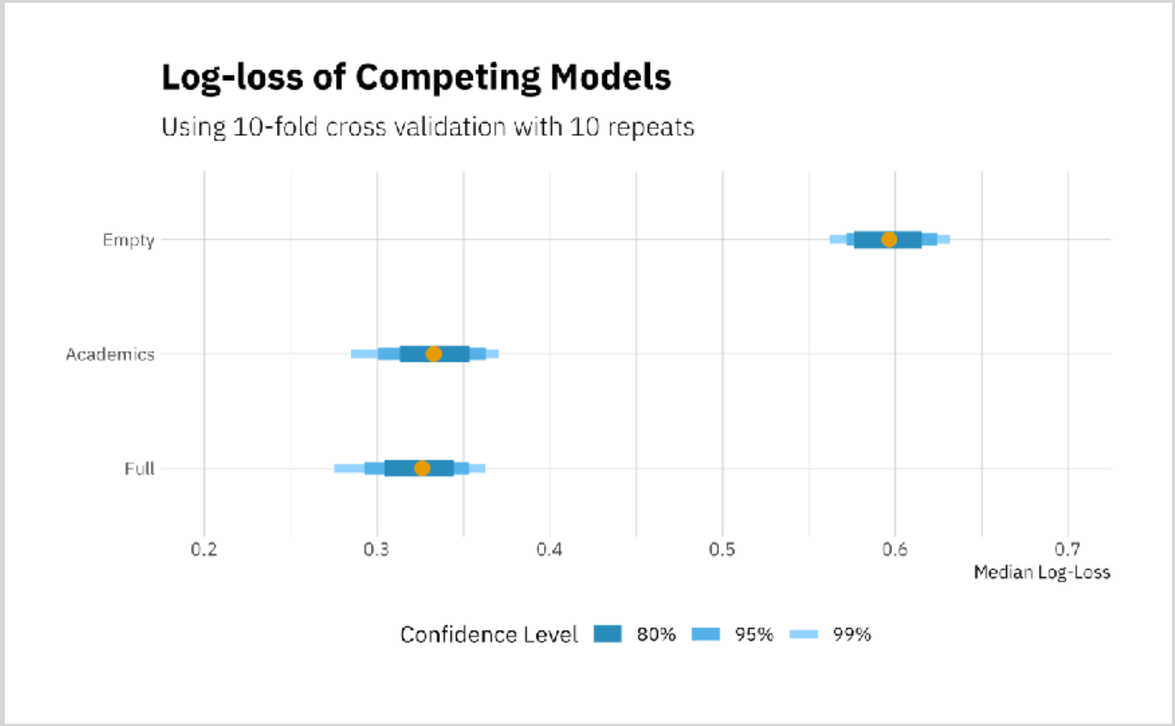
Your Turn 7

- Calculate the Log Loss for each **repeat**, **fold**, and **model**
- Hint: Which variable is $\mathbf{y_i}$ and $\mathbf{\hat{y}_i}$?

$$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

03 : 00

```
all_preds %>%
  group_by(id, id2, model) %>%
  summarize(logloss = -1 * mean((admit * log(.fitted)) + ((1 - admit) * log(1 - .fitted))))
# A tibble: 300 x 4
# Groups:   id, id2 [100]
   id       id2    model     logloss
   <chr>    <chr>  <chr>       <dbl>
 1 Repeat01 Fold01 acadm_mod   0.341
 2 Repeat01 Fold01 compl_mod   0.330
 3 Repeat01 Fold01 empty_mod   0.604
 4 Repeat01 Fold02 acadm_mod   0.322
 5 Repeat01 Fold02 compl_mod   0.317
 6 Repeat01 Fold02 empty_mod   0.601
 7 Repeat01 Fold03 acadm_mod   0.339
 8 Repeat01 Fold03 compl_mod   0.329
 9 Repeat01 Fold03 empty_mod   0.590
10 Repeat01 Fold04 acadm_mod   0.350
# … with 290 more rows
```

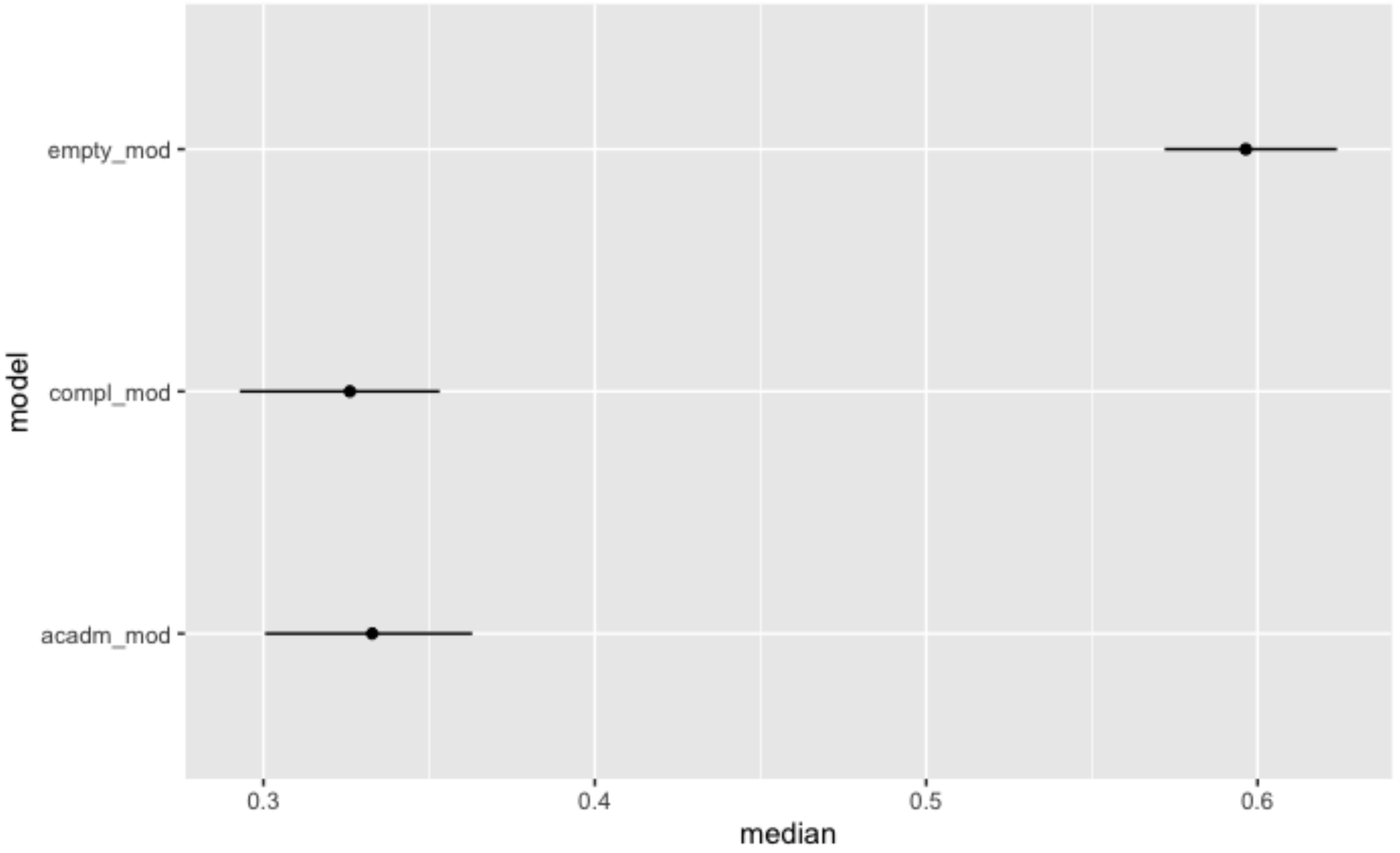


Your Turn 8

- For each model, calculate the median, 2.5 and 97.5 percentiles of the Log Loss
  - 95% confidence interval
- Plot the results using **geom_errorbarh**
- Hint: look at the required aesthetics for by using **?geom_errorbarh**

03:00

```
all_preds %>%
  group_by(id, id2, model) %>%
  summarize(logloss = -1 * mean((admit * log(.fitted)) + ((1 - admit) * log(1 - .fitted)))) %>%
  group_by(model) %>%
  summarize(median = median(logloss),
            lb = quantile(logloss, probs = 0.025),
            ub = quantile(logloss, probs = 0.975)) %>%
  ggplot(aes(y = model)) +
    geom_errorbarh(aes(xmin = lb, xmax = ub), height = 0) +
    geom_point(aes(x = median))
```


empty_mod
compl_mod
acadm_mod
model
0.3
0.4
0.5
0.6
median

Extra Challenges

How would you add multiple error bars for varying confidence intervals?

Are there other methods we could use to measure the predictive accuracy of the models?

Case Study

ATLAS
Accessible Teaching, Learning, and Assessment Systems

wjakethompson.com wjakethompson@ku.edu @wjakethompson @wjakethompson