Modeling: Introduction

DATA 202 21FA

Logistics

- Midterm 1 Online Portion is on Moodle
  - Don't forgot to copy-paste responses and completion code to Moodle
  - Closes Wed evening
- Project Milestone 2 due Friday
- No quiz or Discussion this week

Q&A!

> What kinds of things can functions return?

Anything you can assign to a variable. (numbers, strings, data frames, plots, ...)

> Are `as.numeric` and `as.integer` the same?

```
as.numeric("1.7")
```

```
[1] 1.7
```

```
as.integer("1.7")
```

```
[1] 1
```

```
as.integer("One Point Seven")
```

```
Warning: NAs introduced by coer
```

```
[1] NA
```

2016 Honda Odyssey
EX-L 4dr Minivan (3.5L 6cyl 6A)

Mileage	37,183
Condition	Outstanding
Exterior	Modern Steel Metallic

Your appraisal As of 10/26/2019

Email report

	Trade-In	Private Party	Dealer Retail
	$21,645		**$23,731**
			$25,666
National Base Price	$19,676	$21,694	$23,484
Color Adjustment	-$43	-$48	-$52
Regional Adjustment	$103	$114	$123
Mileage Adjustment	$680	$680	$680
Condition Adjustment	$1,229	$1,291	$1,431
Value	**$21,645**	**$23,731**	**$25,666**

What data frame does Edmunds have? Sketch an example. (What are the columns? What does each row represent?)

Edmunds.com

Missing Data

- Tools so far very good when we have all the data we want.
- Usually we *don't* -- or even we *can't*.

Different kinds of missingness:

1. How much will this car / house sell for?
2. What will Bitcoin trade for at end-of-day today?
3. What will this addition do to the price of my house?
4. I feel rotten. Is it Covid?
5. What word will the user type next on their keyboard?
6. Will this home-buyer default on their loan?

Discussion:

1. For each of these, discuss **what data is missing** (*why*?).
2. Which of these situations are similar to each other?

Some terminology

- **Supervised learning**: for each item independently, fill in an unobserved variable
  - **Regression**: fill in a *number*
  - **Classification**: fill in a *choice*
- **Unsupervised learning**: identify *relationships* between items
- **Forecasting**: predict how a sequence will continue (future observations)
- **Statistical inference**: fill in summary statistics (we wanted a population but only got a sample)
- **Causal inference**: fill in counterfactuals (what if?)

Predictive Analytics

Mostly *supervised learning* (*regression* and *classification*), some *forecasting*.

- A powerful tool to turn data into action.
- It works because God made the universe predictable (and successful prediction rewarding)
- **Need for wisdom**: It can be used for great good and great harm

Power of Predictive Modeling

- **Medicine**: wearable monitor for seizures or falls, detect malaria from blood smears, find effective drug regimens from medical records
- **Drug Discovery**: predict the efficacy of a synthesis plan for a drug
- **Precision Agriculture**: predict effect of micro-climate on plant growth
- **Urban Planning**: forecast resource needs, extreme weather risks, ...
- **Government**: classify feedback from constituents
- **Retail**: predict items in a grocery order
- **Recommendation systems**: Amazon, Netflix, YouTube, ...
- **User interfaces**: gesture typing, autocomplete / autocorrect

and so much more...

The universe is surprisingly predictable

- God created the world with actionable structure
  - We gradually learn how to perceive that structure and act within it.
  - The better our perceptions align with how the universe is structured, the better our actions
  - We can discover that structure by learning to be less surprised by what we see (= predicting our perceptions)
- Perceptions are thus both accurate and fallable.

Predictive modeling technology: Need for wisdom

- Potential for great good
- But also great harm:
  - Lack of **fairness** in facial recognition, sentencing, lending, job applicant scoring, ...
  - Lack of **transparency** in how "Big Data" systems make conclusions
  - Lack of **privacy** as data is increasingly collected and aggregated
  - Amplification of extreme positions in social media, YouTube, etc.
  - Oversimplification of human experience
  - Hidden human labor
  - Illusion of objectivity
  - ...!

Wednesday

Q&A

> How will we code predictive models?

- We'll use `tidymodels`, a toolkit like `tidyverse`. Preview today, practice Friday.
- We'll need all our `tidyverse` skills to understand our data before modeling it and to visualize our models.

> What are the limits? Can we predict everything?

- Silicon Valley: "Yes!"
- Wisdom: "No!"
  - What failure to predict life outcomes can teach us
  - Predictions can be inaccurate and biased, with disparate impacts on vulnerable people.

Different kinds of missing data

- Missing *items*: entire rows not in your data.
  - Some people didn't fill out your survey.
  - You only know about people who *did* visit your website, not those that didn't. etc.
- Missing *observations*: row present, but some observations missing
  - someone added a product to their cart but didn't buy it, so you don't know their address.
  - only some people got an expensive diagnostic test. etc.

Why not just ignore what's missing?

- Ignoring missing data leads to selection bias and related biases (see catalogofbias.org).
- Missing *observations* can be exactly the info you need for making a decision, e.g., what price to list your product for.
- Implications:
  - **Never** `drop_na()` without explanation.
  - "The map is not the territory" -- data ≠ objective reality

Examples of Supervised Learning

Regression Example: Home Sales

From Ames, Iowa home sales, 2006-2010. (De Cock, 2011)

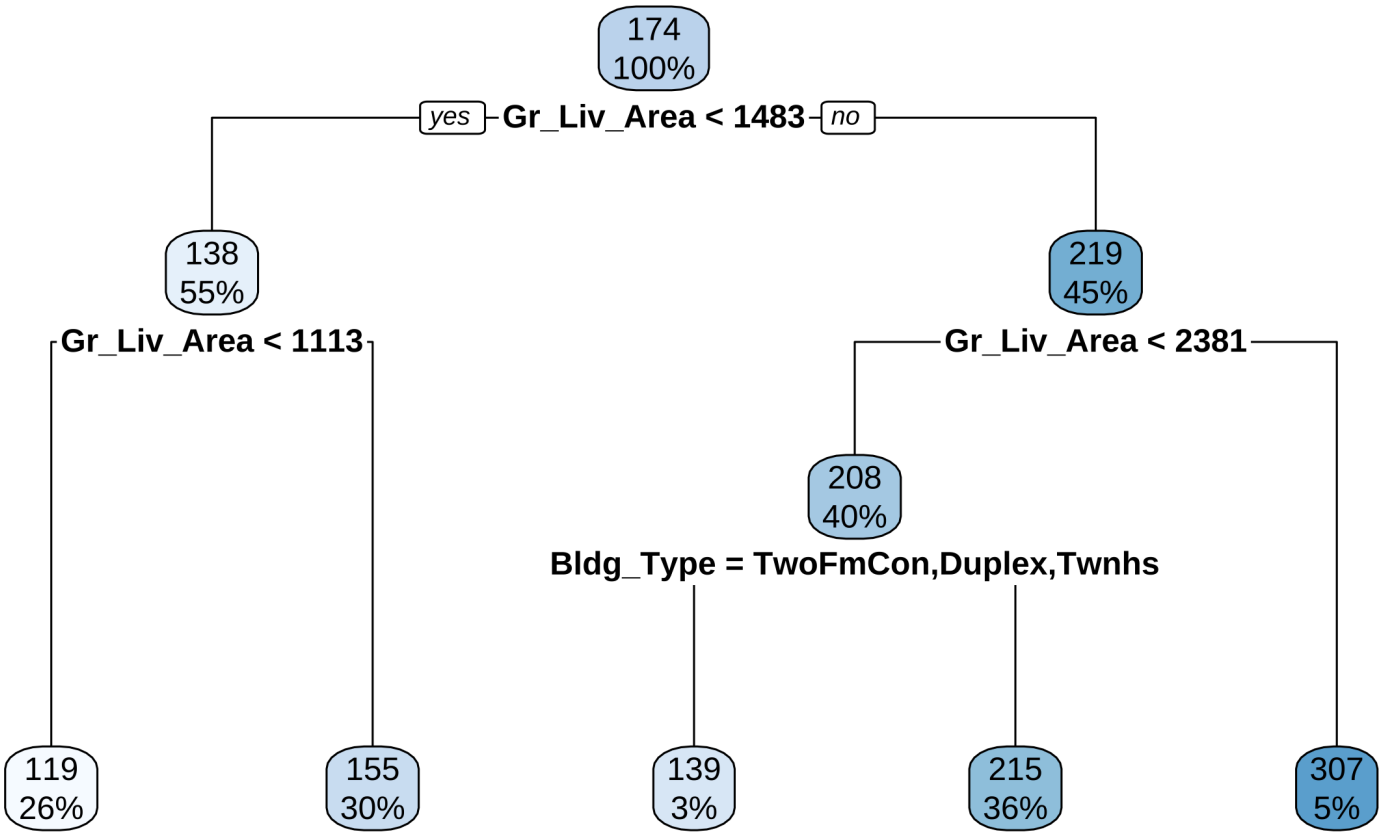
Lot_Area	Bldg_Type	Gr_Liv_Area	Garage_Cars	Sale_Price
12546	OneFam	1440	2	182.9
2645	Twnhs	1586	2	170.0
15312	OneFam	1138	2	148.0
8544	Duplex	1040	2	81.4
12677	TwnhsE	1518	2	274.0

(2412 total rows)

- *y*: *response* variable (aka *outcome*, *dependent variable*): `Sale_Price`
- *X*: *features* (aka *predictors*, *covariates*, etc.): everything else

Note: `X` is much easier to measure than `y`

Model | Predictions | Model code



Were those predictions good?

```
metrics <- yardstick::metric_set(mae, mape, rsq_trad, rmse)
decision_tree_fit %>%
  predict(ames_test) %>%
  bind_cols(ames_test) %>%
  metrics(truth = Sale_Price, estimate = .pred) %>%
  select(-.estimator) %>% knitr::kable()
```

.metric	.estimate
mae	35.3756288
mape	21.1170598
rsq_trad	0.5254953
rmse	50.5840333

- Traditional R^2 (fraction of variance explained)
- MAE: Mean Absolute Error ("predictions are usually off by $xxx")
- MAPE: Mean Absolute Percent Error ("predictions are usually off by yy%")

Classification example: Can a blood test diagnose autism?

We'll use an example from a 2017 PLOS Computational Biology paper

```
autism
```

```
# A tibble: 206 × 26
  Group Methion.   SAM   SAH `SAM/SAH` `% DNA methylatio… `8-OHG`
  <chr>    <dbl> <dbl> <dbl>     <dbl>               <dbl>   <dbl>
1 ASD       17.3  56.2 15.2       3.69                3.35   0.055
2 ASD       14.9  37.2  7.58      4.91                3.04   0.045
3 ASD       15.9  37.9  9.87      3.84                2.81   0.058
4 ASD       18.7  79.2 24.5       3.23                4.24   0.085
5 ASD       21.5  77.6 19.2       4.04                3.49   0.041
6 ASD       18.1  67.6 12.8       5.30                3.01   0.156
# … with 200 more rows, and 19 more variables: Adenosine <dbl>,
#   Homocysteine <dbl>, Cysteine <dbl>, 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>,
```

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()) %>% knitr::kab
```

Group	n
ASD	83
NEU	76
SIB	47

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

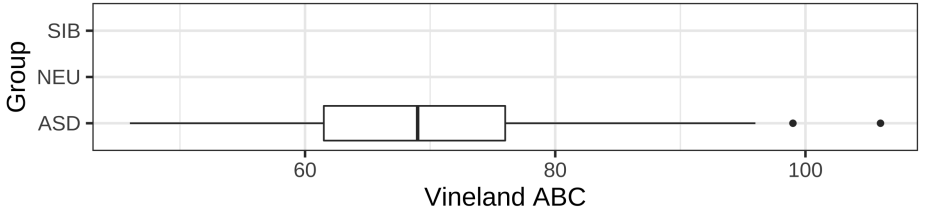
```
autism %>% select(-1, -last_col())
```

```
# A tibble: 206 × 24
  Methion.   SAM   SAH `SAM/SAH` `% DNA methylation` `8-OHG`
     <dbl> <dbl> <dbl>     <dbl>                <dbl>   <dbl>
1     17.3  56.2 15.2       3.69                 3.35   0.055
2     14.9  37.2  7.58      4.91                 3.04   0.045
3     15.9  37.9  9.87      3.84                 2.81   0.058
4     18.7  79.2 24.5       3.23                 4.24   0.085
5     21.5  77.6 19.2       4.04                 3.49   0.041
6     18.1  67.6 12.8       5.30                 3.01   0.156
# … with 200 more rows, and 18 more variables: Adenosine <dbl>,
#   Homocysteine <dbl>, Cysteine <dbl>, 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")

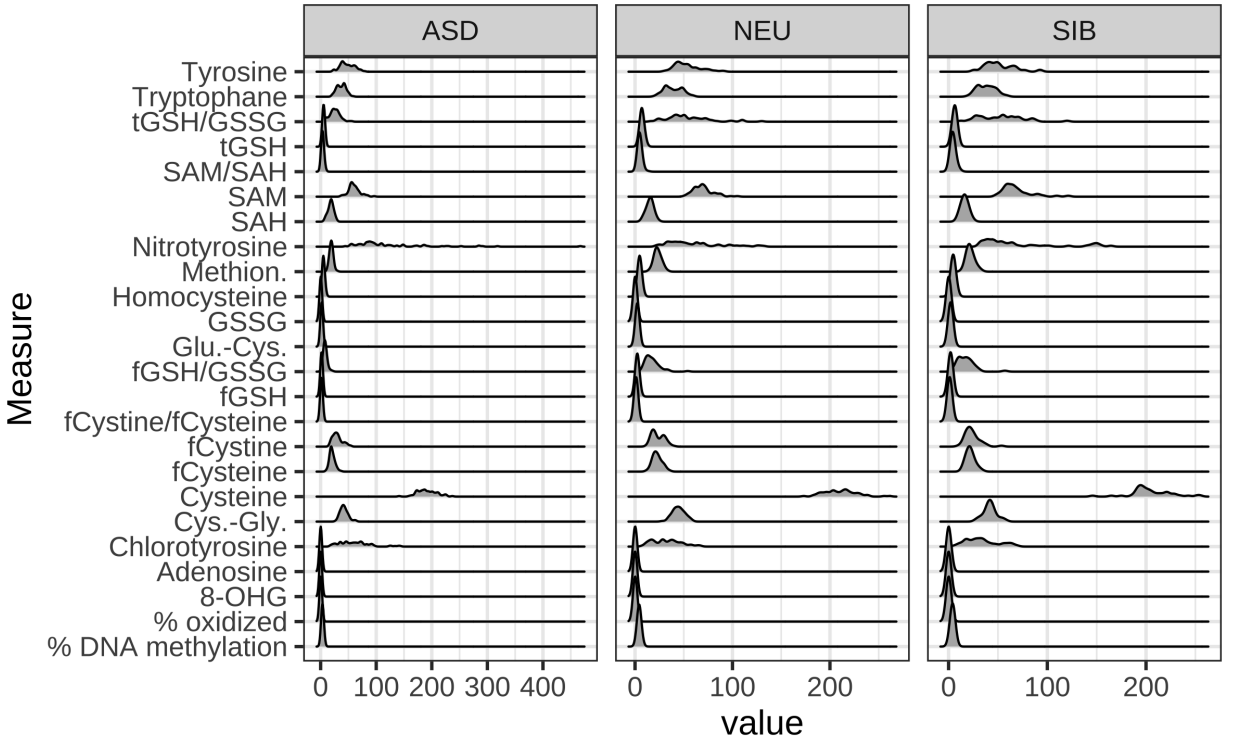
```
autism %>%
  ggplot(aes(x = `Vineland ABC`, y = Group)) + geom_boxplot()
```

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



Exploratory Data Analysis (EDA)

What do these metabolites look like?

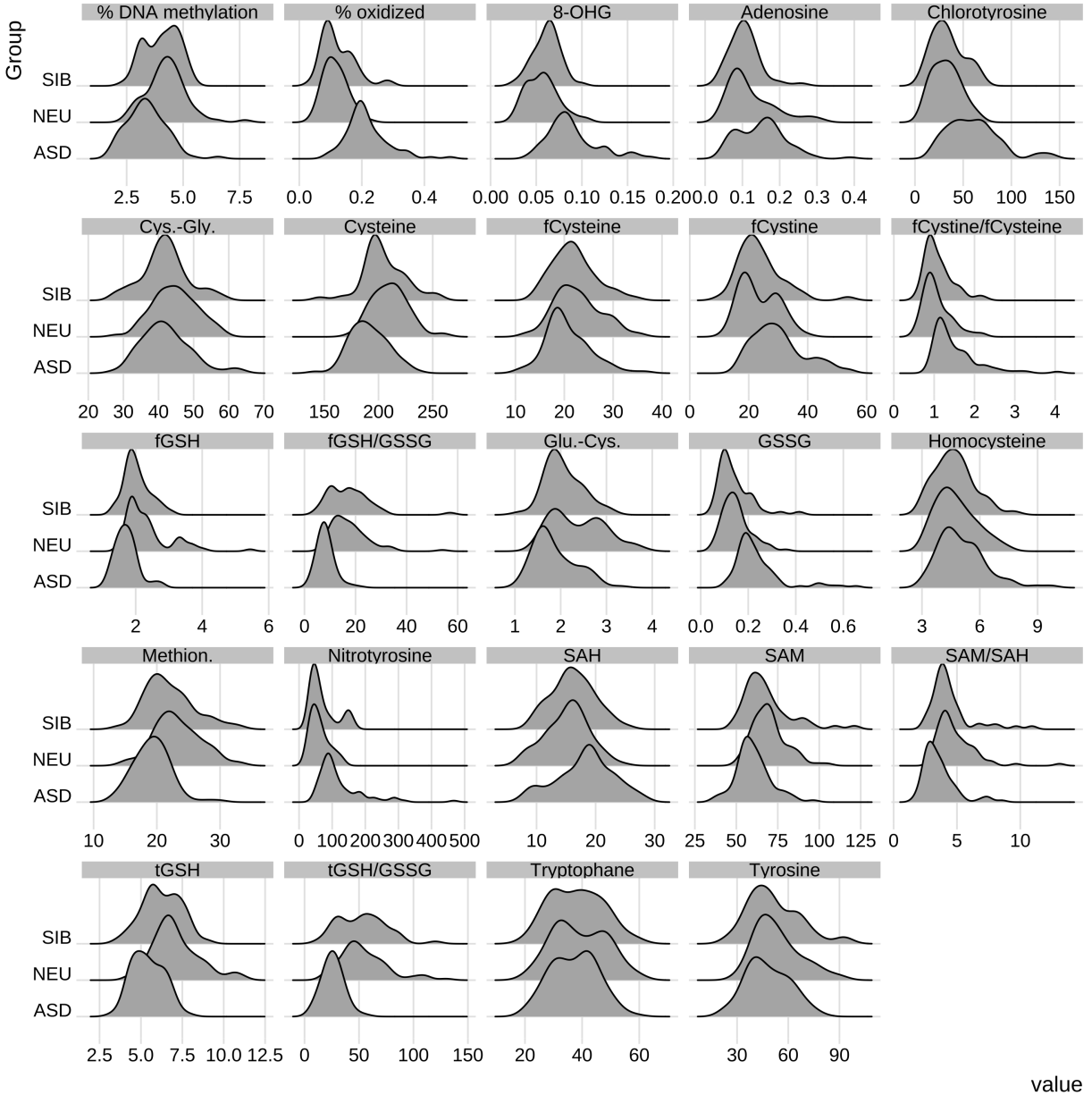


code for the previous plot:

```
library(ggridges)
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
tGSH
tGSH/GSSG
Tryptophane
Tyrosine
value

code for previous plot:

```
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()
```

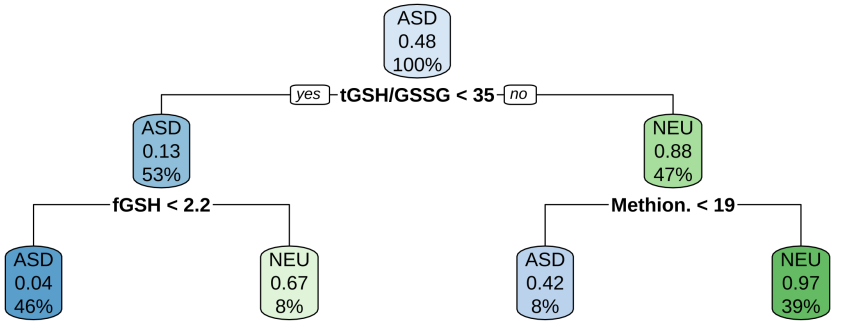
Can we predict ASD vs non-ASD from metabolites?

- 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 encode `Group` as a factor.

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

Decision Tree *Classification*

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



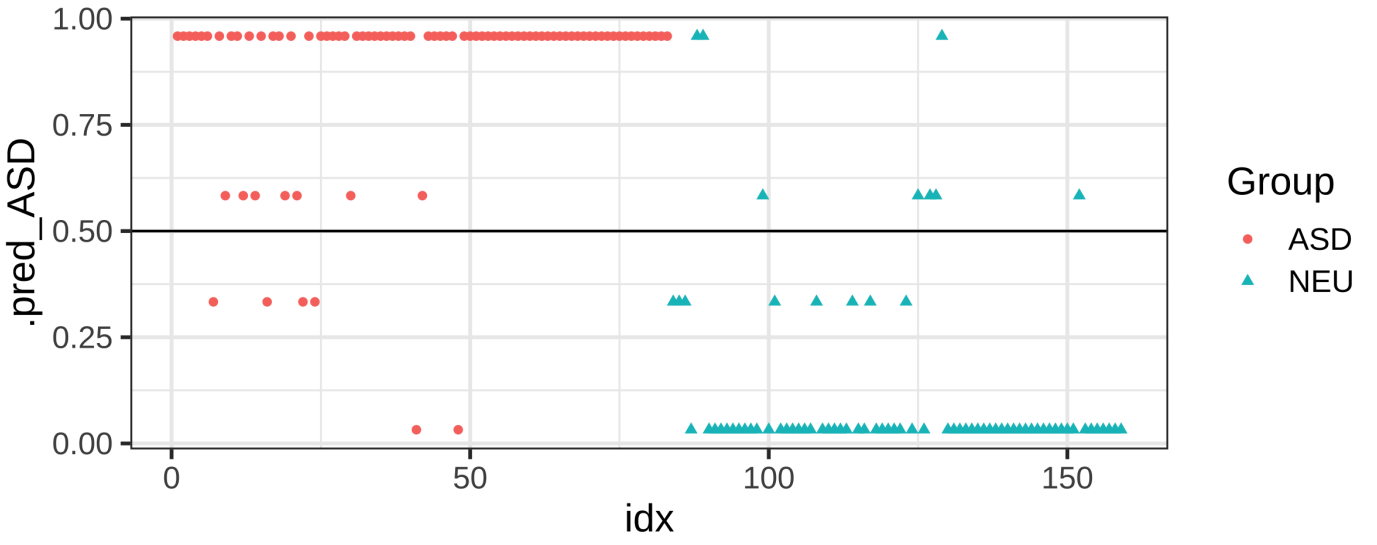
What do the *predictions* look like?

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

```
# A tibble: 159 × 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*?

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

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

Classification Metrics

	Event happened	No event happened
Event predicted	True positive	False positive (Type 1 error)
No event predicted	False negative (Type 2 error)	True negative

Classification Metrics

	Event happened	No event happened
Event predicted	True positive	False positive (Type 1 error)
No event predicted	False negative (Type 2 error)	True negative

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

Classification Metrics

	Event happened	No event happened
Event predicted	True positive	False positive (Type 1 error)
No event predicted	False negative (Type 2 error)	True negative

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