

432 Class 06 Slides

thomaseLove.github.io/432

2021-02-18

Today's Agenda

- Predicting a Binary outcome
 - using a linear probability model
 - using logistic regression and `glm`

Setup

```
library(here); library(magrittr)
library(janitor); library(knitr)
library(patchwork); library(broom)
library(simputation); library(naniar)
library(rsample); library(yardstick)
library(tidyverse)

theme_set(theme_bw())
```

Regression on a Binary Outcome

Linear Probability Model (a linear model)

```
lm(event ~ predictor1 + predictor2 + ..., data = tibblename)
```

- $\Pr(\text{event})$ is linear in the predictors

Logistic Regression Model (generalized linear model)

```
glm(event ~ pred1 + pred2 + ..., data = tibblename,  
     family = binomial(link = "logit"))
```

- Logistic Regression forces a prediction in $(0, 1)$
- $\log(\text{odds}(\text{event}))$ is linear in the predictors

The logistic regression model

$$\text{logit}(\text{event}) = \log \left(\frac{\text{Pr}(\text{event})}{1 - \text{Pr}(\text{event})} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$

$$\text{odds}(\text{event}) = \frac{\text{Pr}(\text{event})}{1 - \text{Pr}(\text{event})}$$

$$\text{Pr}(\text{event}) = \frac{\text{odds}(\text{event})}{\text{odds}(\text{event}) + 1}$$

$$\text{Pr}(\text{event}) = \frac{\exp(\text{logit}(\text{event}))}{1 + \exp(\text{logit}(\text{event}))}$$

BRFSS and SMART (Creating smart3)

```
smart3 <- read_csv(here("data/smart_ohio.csv")) %>%  
  mutate(SEQNO = as.character(SEQNO)) %>%  
  select(SEQNO, mmsa, mmsa_wt, landline,  
         age_imp, healthplan, dm_status,  
         fruit_day, drinks_wk, activity,  
         smoker, physhealth, bmi, genhealth)
```

smart3 Variables, by Type

Variable	Type	Description
landline	Binary (1/0)	survey conducted by landline? (vs. cell)
healthplan	Binary (1/0)	subject has health insurance?
age_imp	Quantitative	age (imputed from groups - see Notes)
fruit_day	Quantitative	mean servings of fruit / day
drinks_wk	Quantitative	mean alcoholic drinks / week
bmi	Quantitative	body-mass index (in kg/m^2)
physhealth	Count (0-30)	of last 30 days, # in poor physical health
dm_status	Categorical	diabetes status (4 levels, <i>we'll collapse to 2</i>)
activity	Categorical	physical activity level (4 levels, <i>we'll re-level</i>)
smoker	Categorical	smoking status (4 levels, <i>we'll collapse to 3</i>)
genhealth	Categorical	self-reported overall health (5 levels)

Collapsing Two Factors, Re-leveling another

```
smart3 <- smart3 %>% type.convert() %>%  
  mutate(SEQNO = as.character(SEQNO)) %>%  
  mutate(dm_status =  
    fct_collapse(factor(dm_status),  
                  Yes = "Diabetes",  
                  No = c("No-Diabetes",  
                        "Pre-Diabetes",  
                        "Pregnancy-Induced")))) %>%  
  mutate(smoker =  
    fct_collapse(factor(smoker),  
                  Current = c("Current_not_daily",  
                              "Current_daily")))) %>%  
  mutate(activity =  
    fct_relevel(factor(activity),  
                "Highly_Active", "Active",  
                "Insufficiently_Active",  
                "Inactive"))
```


“Simple” Imputation Strategy, with Shadow

```
smart3_sh <- smart3 %>% bind_shadow()

set.seed(2021432)

smart3_sh <- smart3_sh %>%
  data.frame() %>%
    impute_rhd(dm_status + smoker ~ 1) %>%
    impute_rhd(healthplan + activity ~ 1) %>%
    impute_rlm(age_imp + fruit_day + drinks_wk + bmi ~
      mmsa + landline + healthplan) %>%
    impute_knn(physhealth ~ bmi) %>%
    impute_cart(genhealth ~ activity + physhealth +
      mmsa + healthplan) %>%
  tibble()
```

Saving the smart3 and smart3_sh tibbles to .Rds

```
saveRDS(smart3, "data/smart3.Rds")
saveRDS(smart3_sh, "data/smart3_sh.Rds")
```

Create binary outcome variable

```
smart3_sh <- readRDS("data/smart3_sh.Rds") %>%  
  mutate(bmilt30 = as.numeric(bmi < 30),  
         dm_status = fct_relevel(dm_status, "No"))  
  
smart3_sh %>%  
  group_by(bmilt30) %>%  
  summarize(n = n(), mean(bmi), min(bmi), max(bmi)) %>%  
  kable(digits = 2)
```

bmilt30	n	mean(bmi)	min(bmi)	max(bmi)
0	2343	35.84	30.0	75.52
1	5069	25.26	13.3	29.99

Predicting BMI < 30 using diabetes status (a factor)

```
mod_DM <- smart3_sh %$%  
  glm(bmilt30 ~ dm_status,  
      family = binomial(link = logit))  
  
tidy(mod_DM) %>% select(term, estimate) %>%  
  kable(digits = 3)
```

term	estimate
(Intercept)	0.947
dm_statusYes	-1.053

Equation: $\text{logit}(\text{BMI} < 30) = 0.947 - 1.053 (\text{dm_statusYes})$

How can we interpret this result?

Interpreting the mod_DM Equation

$$\text{logit}(\text{BMI} < 30) = 0.947 - 1.053 (\text{dm_status} = \text{Yes})$$

- Harry has diabetes.
 - His predicted $\text{logit}(\text{BMI} < 30)$ is $0.947 - 1.053 (1) = -0.106$
- Sally does not have diabetes.
 - Her predicted $\text{logit}(\text{BMI} < 30)$ is $0.947 - 1.053 (0) = 0.947$

Now, $\text{logit}(\text{BMI} < 30) = \log(\text{odds}(\text{BMI} < 30))$, so exponentiate to get the odds...

- Harry has predicted $\text{odds}(\text{BMI} < 30) = \exp(-0.106) = 0.899$
- Sally has predicted $\text{odds}(\text{BMI} < 30) = \exp(0.947) = 2.578$

Can we convert these odds into something more intuitive?

Converting Odds to Probabilities

- Harry has predicted odds($BMI < 30$) = $\exp(-0.106) = 0.899$
- Sally has predicted odds($BMI < 30$) = $\exp(0.947) = 2.578$

$$odds(BMI < 30) = \frac{Pr(BMI < 30)}{1 - Pr(BMI < 30)}$$

and

$$Pr(BMI < 30) = \frac{odds(BMI < 30)}{odds(BMI < 30) + 1}$$

- So Harry's predicted $Pr(BMI < 30) = 0.899 / 1.899 = 0.47$
- Sally's predicted $Pr(BMI < 30) = 2.578 / 3.578 = 0.72$
- odds range from 0 to ∞ , and $\log(odds)$ range from $-\infty$ to ∞ .
- odds > 1 if probability > 0.5 . If odds = 1, then probability = 0.5.

What about the odds ratio?

$\text{logit}(\text{BMI} < 30) = 0.947 - 1.053 (\text{dm_status} = \text{Yes})$

- Harry, with diabetes, has $\text{odds}(\text{BMI} < 30) = 0.899$
- Sally, without diabetes, has $\text{odds}(\text{BMI} < 30) = 2.578$

Odds Ratio for $\text{BMI} < 30$ associated with having diabetes (vs. not) =

$$\frac{0.899}{2.578} = 0.349$$

- Our model estimates that a subject with diabetes has 34.9% of the odds of a subject without diabetes of having $\text{BMI} < 30$.

Can we calculate the odds ratio from the equation's coefficients?

- Yes, $\exp(-1.053) = 0.349$.

Tidy with exponentiation

```
tidy(mod_DM, exponentiate = TRUE,  
      conf.int = TRUE, conf.level = 0.9) %>%  
  select(term, estimate, conf.low, conf.high) %>%  
  kable(digits = 3)
```

term	estimate	conf.low	conf.high
(Intercept)	2.578	2.462	2.700
dm_statusYes	0.349	0.313	0.389

- The odds ratio for BMI < 30 among subjects with diabetes as compared to those without diabetes is 0.349
- The odds of BMI < 30 are 34.9% as large for subjects with diabetes as they are for subjects without diabetes, according to this model.
- A 90% uncertainty interval for the odds ratio estimate includes (0.313, 0.389).

Interpreting these summaries

Connecting the Odds Ratio and Log Odds Ratio to probability statements. . .

- If the probabilities were the same (for diabetes and non-diabetes subjects) of having $\text{BMI} < 30$, then the odds would also be the same, and so the odds ratio would be 1.
- If the probabilities of $\text{BMI} < 30$ were the same and thus the odds were the same, then the log odds ratio would be $\log(1) = 0$.

$\text{logit}(\text{BMI} < 30) = 0.947 - 1.053 (\text{dm_status} = \text{Yes})$

- 1 If the log odds of a coefficient (like $\text{diabetes} = \text{Yes}$) are negative, then what does that imply?
- 2 What if we flipped the order of the levels for diabetes so our model was about $\text{diabetes} = \text{No}$?

Flipping the model changes slope and intercept!

```
mod_DM_no <- smart3_sh %$%  
  glm(bmilt30 ~ (dm_status == "No"),  
      family = binomial(link = logit))  
  
tidy(mod_DM_no) %>% select(term, estimate) %>%  
  kable(digits = 3)
```

term	estimate
(Intercept)	-0.106
dm_status == "No"TRUE	1.053

Old: $\text{logit}(\text{BMI} < 30) = 0.947 - 1.053 (\text{dm_status} = \text{Yes})$ New:
 $\text{logit}(\text{BMI} < 30) = -0.106 + 1.053 (\text{dm_status} = \text{No})$

Predictions from the two models?

DMYes: $\text{logit}(\text{BMI} < 30) = 0.947 - 1.053$ ($\text{dm_status} = \text{Yes}$)

DMNo: $\text{logit}(\text{BMI} < 30) = -0.106 + 1.053$ ($\text{dm_status} = \text{No}$)

Harry lives with diabetes. Sally does not.

Using the DMYes model:

- $\text{logit}(\text{Harry's BMI} < 30) = 0.947 - 1.053 = -0.106$
- $\text{logit}(\text{Sally's BMI} < 30) = 0.947$

Using the DMNo model:

- $\text{logit}(\text{Harry's BMI} < 30) = -0.106$
- $\text{logit}(\text{Sally's BMI} < 30) = -0.106 + 1.053 = 0.947$

Two Predictor Model

We'll fit the model with and without an interaction term...

```
mod_FDmain <- glm(bmilt30 ~ fruit_day + dm_status,  
                  data = smart3_sh, family = binomial)  
mod_FDint <- glm(bmilt30 ~ fruit_day * dm_status,  
                 data = smart3_sh, family = binomial)
```

- 1 How will we interpret the model coefficients?
- 2 Can we compare the models based on in-sample performance?
- 3 (later) How can we assess predictive quality (with a holdout sample)?

Coefficients in the Model without Interaction

```
tidy(mod_FDmain, conf.int = TRUE, conf.level = 0.95) %>%  
  select(term, estimate, std.error, conf.low, conf.high) %>%  
  kable(dig = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	0.762	0.042	0.679	0.845
fruit_day	0.142	0.025	0.094	0.192
dm_statusYes	-1.051	0.067	-1.182	-0.920

ID	fruit	dm	logit(BMI < 30)	odds(BMI < 30)	Pr(BMI < 30)
Art	1	Yes	-0.147	0.863	0.463
Bob	0	Yes	-0.289	0.749	0.428
Cal	1	No	0.904	2.469	0.712
Don	0	No	0.762	2.143	0.682

Exponentiating fruit_day coefficient to get OR

```
tidy(mod_FDmain, exponentiate = TRUE) %>%  
  select(term, estimate) %>% kable(dig = 2)
```

term	estimate
(Intercept)	2.14
fruit_day	1.15
dm_statusYes	0.35

ID	fruit	dm	logit(BMI < 30)	odds(BMI < 30)	Pr(BMI < 30)
Art	1	Yes	-0.147	0.863	0.463
Bob	0	Yes	-0.289	0.749	0.428

- Art's odds = 0.863. Bob's odds = 0.749
- Art/Bob odds ratio = $0.863/0.749 = 1.15$

Exponentiating dm_status coefficient to get OR

```
tidy(mod_FDmain, exponentiate = TRUE) %>%  
  select(term, estimate) %>% kable(dig = 2)
```

term	estimate
(Intercept)	2.14
fruit_day	1.15
dm_statusYes	0.35

ID	fruit	dm	logit(BMI < 30)	odds(BMI < 30)	Pr(BMI < 30)
Bob	0	Yes	-0.289	0.749	0.428
Don	0	No	0.762	2.143	0.682

- Bob's odds = 0.749, Don's odds (2.143)
- Bob/Don odds ratio = $0.749/2.143 = 0.35$

Same result when fruit_day = 1?

```
tidy(mod_FDmain, exponentiate = TRUE) %>%  
  select(term, estimate) %>% kable(dig = 2)
```

term	estimate
(Intercept)	2.14
fruit_day	1.15
dm_statusYes	0.35

ID	fruit	dm	logit(BMI < 30)	odds(BMI < 30)	Pr(BMI < 30)
Art	1	Yes	-0.147	0.863	0.463
Cal	1	No	0.904	2.469	0.712

- Art's odds = 0.863, Cal's odds (2.469)
- Art/Cal odds ratio = $0.863/2.469 = 0.35$ as well.

Coefficients in the Model WITH Interaction

```
tidy(mod_FDint, conf.int = TRUE, conf.level = 0.95) %>%  
  select(term, estimate, std.error, conf.low, conf.high) %>%  
  kable(dig = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	0.762	0.045	0.673	0.851
fruit_day	0.142	0.028	0.087	0.198
dm_statusYes	-1.053	0.104	-1.258	-0.850
fruit_day:dm_statusYes	0.001	0.062	-0.119	0.125

- Again, these estimates are on the logit scale
- Does the interaction look like it has a large effect?

Resulting Predictions from `mod_FDint`

ID	fruit	dm	logit(BMI < 30)	odds(BMI < 30)	Pr(BMI < 30)
Art	1	Yes	-0.148	0.862	0.463
Bob	0	Yes	-0.291	0.748	0.428
Cal	1	No	0.904	2.469	0.712
Don	0	No	0.762	2.143	0.682

Predicted logits (also rounded to 3 decimal places) from `mod_FDmain` were:

- Art: -0.147, Bob: -0.289, Cal: 0.904, Don: 0.762

That's a small enough difference to leave the (rounded) probability estimates unchanged.

Split the Sample and Compare Models?

Let's split our sample

```
set.seed(432)

sm3_split <- initial_split(smart3_sh, prop = 0.7,
                           strata = bmilt30)

sm3_train <- training(sm3_split)
sm3_test  <- testing(sm3_split)
```

- What does `strata = bmilt30` do?

Impact of using strata = bmilt30 in split

```
sm3_train %>% tabyl(bmilt30)
```

bmilt30	n	percent
0	1641	0.316185
1	3549	0.683815

```
sm3_test %>% tabyl(bmilt30)
```

bmilt30	n	percent
0	702	0.3159316
1	1520	0.6840684

Now, we'll build and compare three models...

```
mod_2 <- glm(bmilt30 ~ fruit_day + dm_status,  
             data = sm3_train, family = binomial)  
  
mod_4 <- glm(bmilt30 ~ fruit_day + dm_status +  
             genhealth + age_imp,  
             data = sm3_train, family = binomial)  
  
mod_6 <- glm(bmilt30 ~ fruit_day + dm_status +  
             genhealth + age_imp + smoker + physhealth,  
             data = sm3_train, family = binomial)
```

Tidied Coefficients for Model 2

```
tidy(mod_2, conf.int = TRUE) %>%  
  select(term, est = estimate, se = std.error,  
         conf.low, conf.high) %>%  
  kable(dig = 3)
```

term	est	se	conf.low	conf.high
(Intercept)	0.732	0.051	0.633	0.832
fruit_day	0.164	0.030	0.105	0.224
dm_statusYes	-1.045	0.080	-1.202	-0.888

Tidied Coefficients for Model 6

term	est	se	conf.low	conf.high
(Intercept)	1.020	0.140	0.747	1.297
fruit_day	0.129	0.030	0.070	0.189
dm_statusYes	-0.903	0.086	-1.072	-0.734
genhealth2_VeryGood	-0.620	0.117	-0.853	-0.394
genhealth3_Good	-1.072	0.116	-1.303	-0.848
genhealth4_Fair	-1.217	0.129	-1.473	-0.967
genhealth5_Poor	-1.338	0.158	-1.650	-1.029
age_imp	0.010	0.002	0.007	0.014

Tidied Coefficients for Model 6

term	est	se	conf.low	conf.high
(Intercept)	1.284	0.153	0.987	1.587
fruit_day	0.138	0.031	0.079	0.200
dm_statusYes	-0.890	0.087	-1.060	-0.720
genhealth2_VeryGood	-0.628	0.117	-0.861	-0.401
genhealth3_Good	-1.077	0.117	-1.310	-0.850
genhealth4_Fair	-1.171	0.138	-1.444	-0.902
genhealth5_Poor	-1.187	0.183	-1.547	-0.829
age_imp	0.013	0.002	0.009	0.016
smokerFormer	-0.547	0.096	-0.737	-0.358
smokerNever	-0.390	0.089	-0.565	-0.217
physhealth	-0.011	0.004	-0.019	-0.003

Summary Statistics for In-Sample Fit

```
bind_rows(glance(mod_2), glance(mod_4), glance(mod_6)) %>%  
  mutate(preds = c("2", "4", "6")) %>%  
  select(preds, nobs, deviance, df.residual, AIC, BIC) %>%  
  kable(digits = 1)
```

preds	nobs	deviance	df.residual	AIC	BIC
2	5190	6276.3	5187	6282.3	6301.9
4	5190	6109.7	5182	6125.7	6178.1
6	5190	6070.3	5179	6092.3	6164.4

Likelihood Ratio Test comparing Models 2 and 4

```
anova(mod_2, mod_4, test = "LRT")
```

Analysis of Deviance Table

Model 1: bmilt30 ~ fruit_day + dm_status

Model 2: bmilt30 ~ fruit_day + dm_status + genhealth + age_imp

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	5187	6276.3			
2	5182	6109.7	5	166.63	< 2.2e-16 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Could also consider

- Rao's efficient score test (`test = "Rao"`)
- Pearson's chi-square test (`test = "Chisq"`).

Likelihood Ratio Test comparing Models 4 and 6

```
anova(mod_4, mod_6, test = "LRT")
```

Analysis of Deviance Table

Model 1: bmilt30 ~ fruit_day + dm_status + genhealth + age_imp

Model 2: bmilt30 ~ fruit_day + dm_status + genhealth + age_imp
physhealth

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	5182	6109.7			
2	5179	6070.3	3	39.321	1.484e-08 ***

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Mallows' Cp statistic?

```
anova(mod_2, mod_4, mod_6, test = "Cp")
```

Analysis of Deviance Table

Model 1: bmilt30 ~ fruit_day + dm_status

Model 2: bmilt30 ~ fruit_day + dm_status + genhealth + age_imp

Model 3: bmilt30 ~ fruit_day + dm_status + genhealth + age_imp
physhealth

	Resid. Df	Resid. Dev	Df	Deviance	Cp
1	5187	6276.3			6282.3
2	5182	6109.7	5	166.628	6125.7
3	5179	6070.3	3	39.321	6092.3

- Same as what glance provided for AIC in this case.

Build mod_2 Confusion Matrix with 0.5 cutoff

```
mod2_aug <- augment(mod_2, data = sm3_train,  
                    type.predict = "response")  
  
confuse2 <- mod2_aug %>%  
  mutate(bmilt30_act = factor(bmilt30 == "1"),  
         bmilt30_pre = factor(.fitted >= 0.5),  
         bmilt30_act = fct_relevel(bmilt30_act, "TRUE"),  
         bmilt30_pre = fct_relevel(bmilt30_pre, "TRUE")) %$%  
  table(bmilt30_pre, bmilt30_act)
```

- We'll review the confusion matrix on the next slide...

Confusion Matrix for mod_2 with 0.5 cutoff

```
confuse2
```

```
          bmilt30_act
bmilt30_pre TRUE FALSE
      TRUE  3291   1333
      FALSE   258    308
```

- Total # of Predictions = $3291 + 1333 + 258 + 308 = 5190$
- $3291 + 308 = 3599$ accurate predictions (69.3% accuracy)
- *Sensitivity* = $3291 / (3291 + 258) = 0.927$
 - if subject actually has BMI < 30 mod_2 with this 0.5 cutoff decision rule correctly predicts BMI < 30 92.7% of the time.
- *Specificity* = $308 / (308 + 1333) = 0.188$
 - if subject actually doesn't have BMI < 30 mod_2 with this 0.5 cutoff decision rule correctly predicts BMI >= 30 18.8% of the time.

Get confusion matrix more easily?

```
mod2_aug <- mod2_aug %>%  
  mutate(obs = factor(bmilt30),  
         pred = factor(ifelse(.fitted >= 0.5, 1, 0)))  
  
conf_mat(data = mod2_aug, truth = obs, estimate = pred)
```

	Truth	
Prediction	0	1
0	308	258
1	1333	3291

Accuracy and Kappa Results for mod_2

```
metrics(data = mod2_aug, truth = obs, estimate = pred)
```

```
# A tibble: 2 x 3
```

	.metric	.estimator	.estimate
	<chr>	<chr>	<dbl>
1	accuracy	binary	0.693
2	kappa	binary	0.140

- Kappa = a correlation statistic from -1 to +1, with complete agreement +1 and complete disagreement -1.
- Kappa measures the inter-rater reliability of our predicted and true classifications.

Confusion Matrix for mod_4 with 0.5 cutoff

```
mod4_aug <- augment(mod_4, data = sm3_train,  
                     type.predict = "response") %>%  
  mutate(obs = factor(bmilt30),  
         pred = factor(ifelse(.fitted >= 0.5, 1, 0)))  
  
conf_mat(data = mod4_aug, truth = obs, estimate = pred)
```

	Truth	
Prediction	0	1
0	309	224
1	1332	3325

- $3325 + 309 = 3634$ accurate predictions (70.0% accuracy)
- $Sensitivity = 3325 / (3325 + 224) = 0.937$
- $Specificity = 309 / (309 + 1332) = 0.188$

Confusion Matrix for mod_6 with 0.5 cutoff

```
conf_mat(data = mod6_aug, truth = obs, estimate = pred)
```

	Truth	
Prediction	0	1
0	312	218
1	1329	3331

- $3331 + 312 = 3643$ accurate predictions (70.2% accuracy)
- $Sensitivity = 3331 / (3331 + 218) = 0.939$
- $Specificity = 312 / (312 + 1329) = 0.190$

metrics for models 2, 4 and 6

```
bind_cols(  
  metrics(data = mod2_aug, truth = obs, estimate = pred) %>%  
    select(.metric, mod2 = .estimate),  
  metrics(data = mod4_aug, truth = obs, estimate = pred) %>%  
    select(mod4 = .estimate),  
  metrics(data = mod6_aug, truth = obs, estimate = pred) %>%  
    select(mod6 = .estimate),  
)
```

```
# A tibble: 2 x 4  
  .metric    mod2    mod4    mod6  
  <chr>     <dbl> <dbl> <dbl>  
1 accuracy 0.693 0.700 0.702  
2 kap      0.140 0.153 0.157
```

Holdout Sample?

```
mod2_aug_test <- augment(mod_2,  
                          newdata = sm3_test,  
                          type.predict = "response") %>%  
mutate(obs = factor(bmilt30),  
       pred = factor(ifelse(.fitted >= 0.5, 1, 0)))
```

- do the same thing for models 4 and 6...

metrics for test sample: models 2, 4 and 6

```
bind_cols(  
  metrics(data = mod2_aug_test,  
           truth = obs, estimate = pred) %>%  
    select(.metric, mod2 = .estimate),  
  metrics(data = mod4_aug_test,  
           truth = obs, estimate = pred) %>%  
    select(mod4 = .estimate),  
  metrics(data = mod6_aug_test,  
           truth = obs, estimate = pred) %>%  
    select(mod6 = .estimate)  
)
```

```
# A tibble: 2 x 4  
  .metric    mod2    mod4    mod6  
  <chr>      <dbl> <dbl> <dbl>  
1 accuracy  0.686  0.694  0.694  
2 kap       0.118  0.136  0.136
```

What's next?

- ROC curve analysis
- Using `lrm` and `ols` from the `rms` package to fit and evaluate logistic and linear regressions, respectively