432 Class 05 Slides

thomaselove.github.io/432

2021-02-16

Moving Forward

- Predicting a Binary outcome
  - using a linear probability model
  - using logistic regression and `glm`
- Creating the `smart3` and `smart3_sh` data
  - A "shadow" to track what is imputed
- Evaluating a Binary Regression Model

Setup

```
library(conflicted)
library(here); library(magrittr)
library(janitor); library(knitr)
library(patchwork); library(broom)
library(equatiomatic)
library(simputation); library(naniar)
library(faraway)                                # for orings data
library(rms)
library(tidyverse)

theme_set(theme_bw())
conflict_prefer("summarize", "dplyr") # choose over Hmisc
```

A First Example: Space Shuttle O-Rings

Challenger Space Shuttle Data

The US space shuttle Challenger exploded on 1986-01-28. An investigation ensued into the reliability of the shuttle's propulsion system. The explosion was eventually traced to the failure of one of the three field joints on one of the two solid booster rockets. Each of these six field joints includes two O-rings which can fail.

The discussion among engineers and managers raised concern that the probability of failure of the O-rings depended on the temperature at launch, which was forecast to be 31 degrees F. There are strong engineering reasons based on the composition of O-rings to support the judgment that failure probability may rise monotonically as temperature drops.

We have data on 23 space shuttle flights that preceded *Challenger* on primary o-ring erosion and/or blowby and on the temperature in degrees Fahrenheit. No previous liftoff temperature was under 53 degrees F.

The "O-rings" data

```
orings1 <- faraway::orings %>%
    tibble() %>%
    mutate(burst = case_when( damage > 0 ~ 1,
                              TRUE ~ 0))

orings1 %>% summary()
```

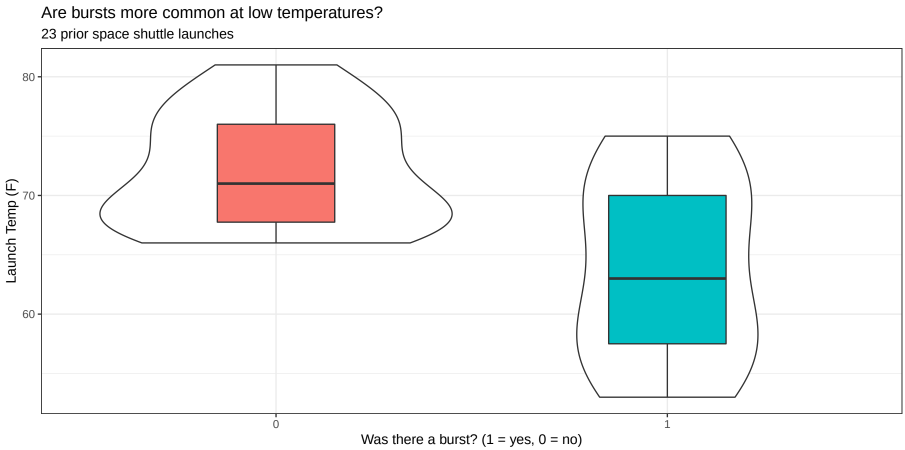
```
      temp           damage           burst
 Min.   :53.00   Min.   :0.0000   Min.   :0.0000
 1st Qu.:67.00   1st Qu.:0.0000   1st Qu.:0.0000
 Median :70.00   Median :0.0000   Median :0.0000
 Mean   :69.57   Mean   :0.4783   Mean   :0.3043
 3rd Qu.:75.00   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :81.00   Max.   :5.0000   Max.   :1.0000
```

- damage = number of damage incidents out of 6 possible
- we set burst = 1 if damage > 0

Code to plot burst and temp in our usual way...

```
ggplot(orings1, aes(x = factor(burst), y = temp)) +
    geom_violin() +
    geom_boxplot(aes(fill = factor(burst)), width = 0.3) +
    guides(fill = FALSE) +
    labs(title = "Are bursts more common at low temperatures?'
         subtitle = "23 prior space shuttle launches",
         x = "Was there a burst? (1 = yes, 0 = no)",
         y = "Launch Temp (F)")
```

Plotted Association of burst and temp

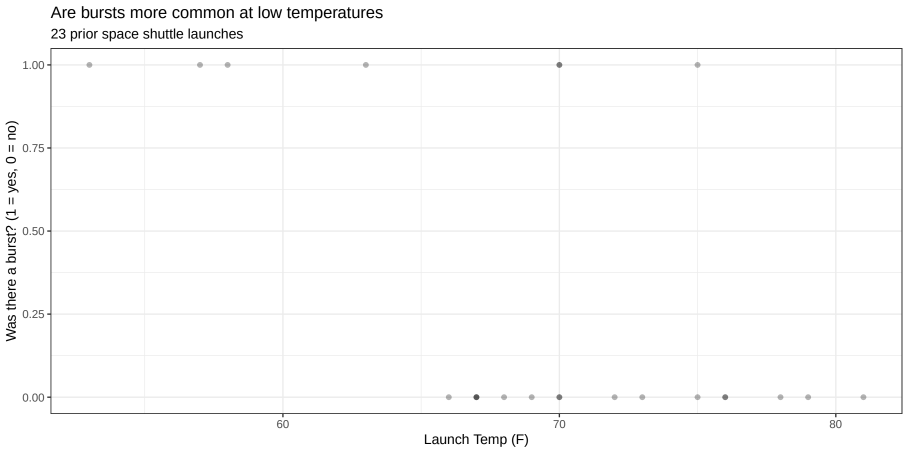


What if we want to predict Prob(burst) using temp?

We want to treat the binary variable `burst` as the outcome, and `temp` as the predictor...

```
ggplot(orings1, aes(x = temp, y = burst)) +
    geom_point(alpha = 0.3) +
    labs(title = "Are bursts more common at low temperatures",
         subtitle = "23 prior space shuttle launches",
         y = "Was there a burst? (1 = yes, 0 = no)",
         x = "Launch Temp (F)")
```

Plot of Prob(burst) by temperature at launch



Fit a linear model to predict Prob(burst)?

```
mod1 <- lm(burst ~ temp, data = orings1)

tidy(mod1, conf.int = T) %>% kable(digits = 3)
```

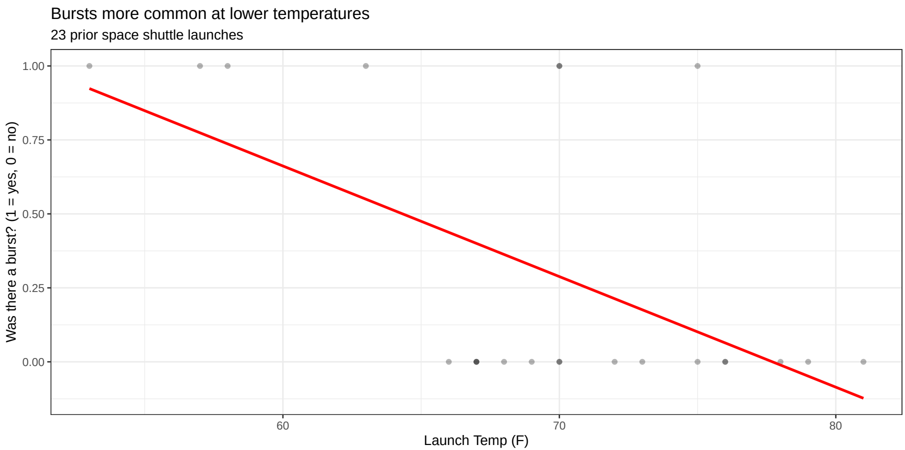
term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	2.905	0.842	3.450	0.002	1.154	4.656
temp	-0.037	0.012	-3.103	0.005	-0.062	-0.012

- This is a **linear probability model**.

```
extract_eq(mod1, use_coefs = TRUE, coef_digits = 3)
```

$$\text{burst} = 2.905 - 0.037(\text{temp}) + \epsilon$$

Add linear probability model to our plot?

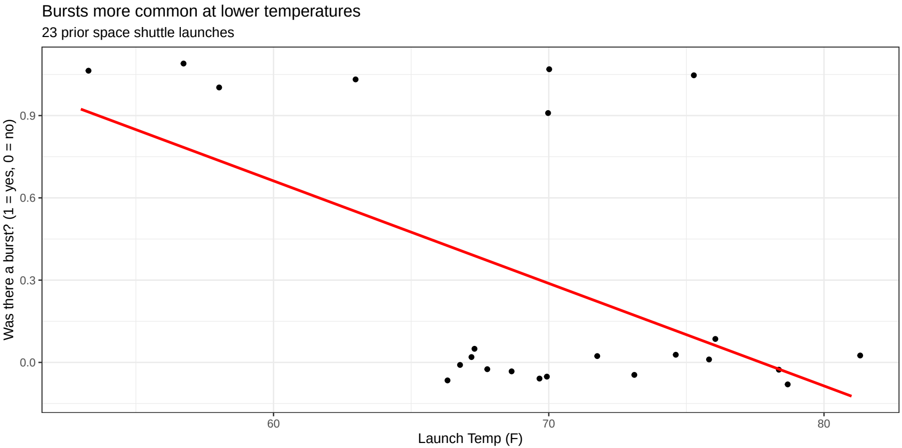


- It would help if we could see the individual launches. . .

Add vertical jitter and our mod1 model?

```
ggplot(orings1, aes(x = temp, y = burst)) +
    geom_jitter(height = 0.1) +
    geom_smooth(method = "lm", se = F, col = "red",
                formula = y ~ x) +
    labs(title = "Bursts more common at lower temperatures",
         subtitle = "23 prior space shuttle launches",
         y = "Was there a burst? (1 = yes, 0 = no)",
         x = "Launch Temp (F)")
```

Resulting plot with points jittered and linear model



- What's wrong with this picture?

Making Predictions with `mod1`

```
tidy(mod1, conf.int = T) %>%
    kable(digits = c(0,5,3,3,3,3,3))
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	2.90476	0.842	3.450	0.002	1.154	4.656
temp	-0.03738	0.012	-3.103	0.005	-0.062	-0.012

- What does `mod1` predict for the probability of a burst if the temperature at launch is 70 degrees F?

$$Prob(burst) = 2.90476 - 0.03738(70) = 0.288$$

- What if the temperature was actually 60 degrees F?

Making Several Predictions with `mod1`

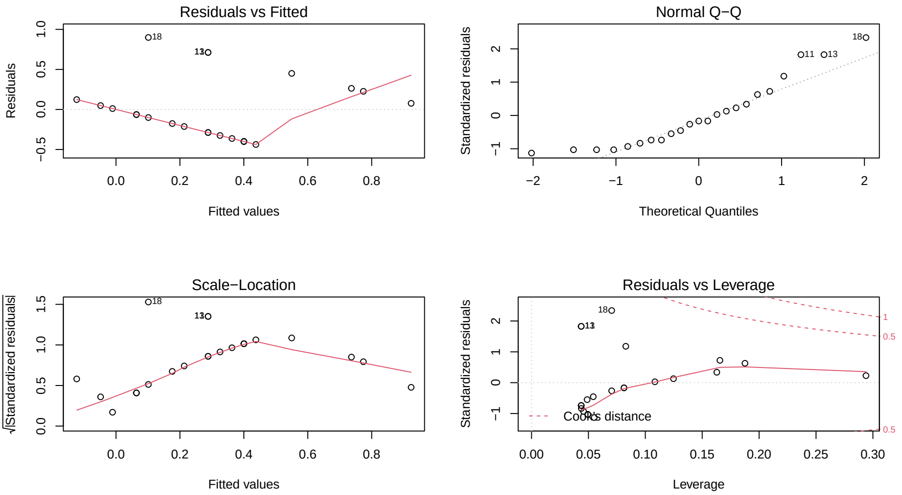
Let's use our linear probability model `mod1` to predict the probability of a burst at some other temperatures...

```
newtemps <- tibble(temp = c(80, 70, 60, 50, 31))

augment(mod1, newdata = newtemps)
```

```
# A tibble: 5 x 2
   temp .fitted
  <dbl>   <dbl>
1    80 -0.0857
2    70  0.288
3    60  0.662
4    50  1.04
5    31  1.75
```

- Uh, oh.

Residual Plots for `mod1`?



- Uh, oh.

Models to predict a Binary Outcome

Our outcome takes on two values (zero or one) and we then model the probability of a "one" response given a linear function of predictors.

Idea 1: Use a *linear probability model*

- Main problem: predicted probabilities that are less than 0 and/or greater than 1
- Also, how can we assume Normally distributed residuals when outcomes are 1 or 0?

Idea 2: Build a *non-linear* regression approach

- Most common approach: logistic regression, part of the class of *generalized* linear models

The Logit Link and Logistic Function

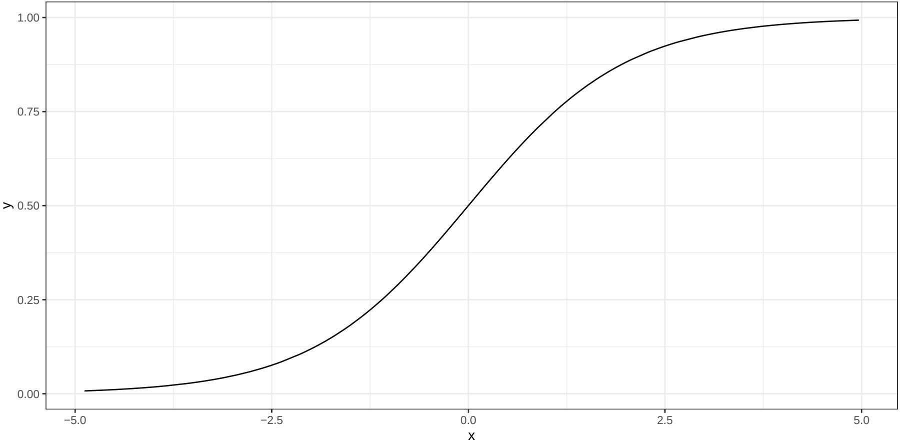
The particular link function we use in logistic regression is called the **logit link**.

$$logit(\pi) = log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$$

The inverse of the logit function is called the **logistic function**. If logit(π) $= \eta$, then $\pi = \frac{exp(\eta)}{1+exp(\eta)}$.

- The logistic function $\frac{e^x}{1+e^x}$ takes any value x in the real numbers and returns a value between 0 and 1.

The Logistic Function $y = \frac{e^x}{1+e^x}$



The logit or log odds

We usually focus on the **logit** in statistical work, which is the inverse of the logistic function.

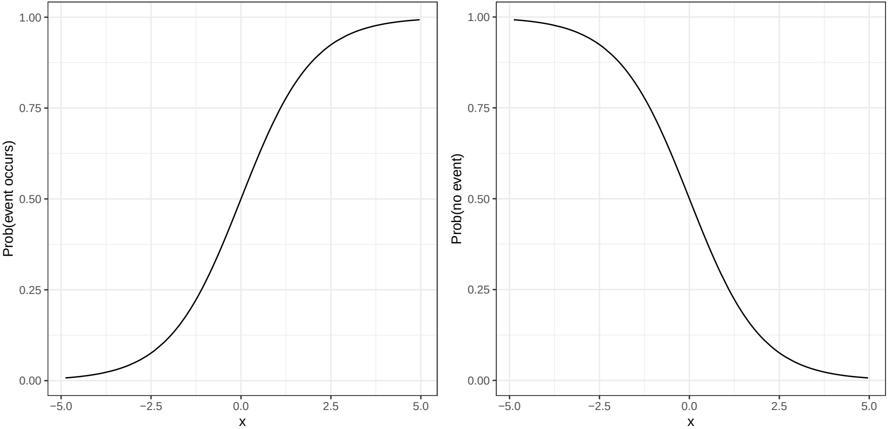
- If we have a probability $\pi < 0.5$, then $logit(\pi) < 0$.
- If our probability $\pi > 0.5$, then $logit(\pi) > 0$.
- Finally, if $\pi = 0.5$, then $logit(\pi) = 0$.

Why is this helpful?

- log(odds(Y = 1)) or logit(Y = 1) covers all real numbers.
- Prob(Y = 1) is restricted to [0, 1].

Predicting Pr(event) or Pr(no event)

- Can we flip the story?



Returning to the prediction of Prob(burst)

We'll use the `glm` function in R, specifying a logistic regression model.

- Instead of predicting $Pr(burst)$, we're predicting $log(odds(burst))$ or $logit(burst)$.

```
mod2 <- glm(burst ~ temp, data = orings1,
            family = binomial(link = "logit"))

tidy(mod2, conf.int = TRUE) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = c(0,4,3,3,3))
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	15.0429	7.379	3.331	34.342
temp	-0.2322	0.108	-0.515	-0.061

Our model mod2

```r
extract_eq(mod2, use_coefs = TRUE, coef_digits = 4)
```

$$\log\left[\frac { P(\operatorname{burst} = 1) }{ 1 - P(\operatorname{burst} = 1) } \right] = 15.0429 - 0.2322(\operatorname{temp}) + \epsilon$$

$$logit(burst) = log(odds(burst)) = 15.0429 - 0.2322 temp$$

- For a temperature of 70 F at launch, what is the prediction?

Let's look at the results

- For a temperature of 70 F at launch, what is the prediction?

log(odds(burst)) = 15.0429 - 0.2322 (70) = -1.211

- Exponentiate to get the odds, on our way to estimating the probability.

odds(burst) = exp(-1.211) = 0.2979

- so, we can estimate the probability by

$$Pr(burst) = \frac{0.2979}{(0.2979 + 1)} = 0.230.$$

Prediction from `mod2` for temp = 60

What is the predicted probability of a burst if the temperature is 60 degrees?

- log(odds(burst)) = 15.0429 - 0.2322 (60) = 1.1109
- odds(burst) = exp(1.1109) = 3.0371
- Pr(burst) = 3.0371 / (3.0371 + 1) = 0.752

Will augment do this, as well?

```r
temps <- tibble(temp = c(60,70))

augment(mod2, newdata = temps, type.predict = "link")
```

```
# A tibble: 2 x 2
   temp .fitted
  <dbl>   <dbl>
1    60    1.11
2    70   -1.21
```

```r
augment(mod2, newdata = temps, type.predict = "response")
```

```
# A tibble: 2 x 2
   temp .fitted
  <dbl>   <dbl>
1    60   0.753
2    70   0.230
```

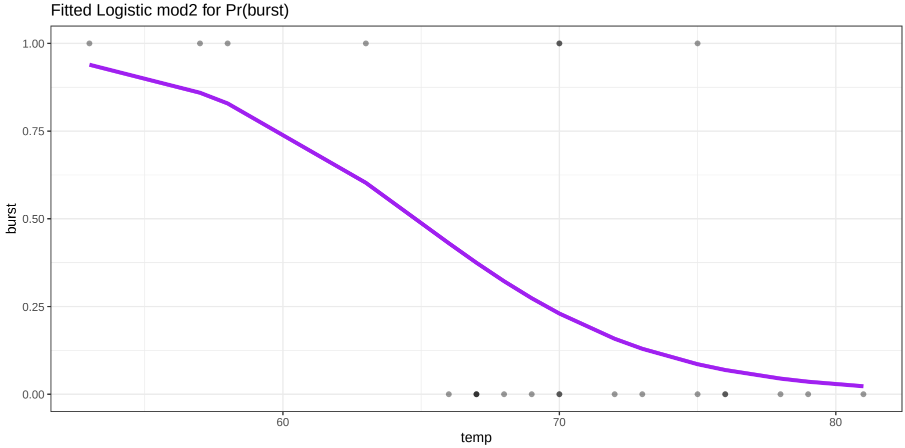
Plotting the Logistic Regression Model

Use the augment function to get the fitted probabilities into the original data, then plot.

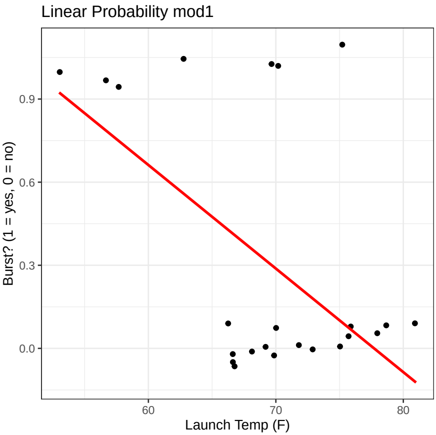
```
mod2_aug <- augment(mod2, type.predict = "response")

ggplot(mod2_aug, aes(x = temp, y = burst)) +
  geom_point(alpha = 0.4) +
  geom_line(aes(x = temp, y = .fitted),
            col = "purple", size = 1.5) +
  labs(title = "Fitted Logistic mod2 for Pr(burst)")
```

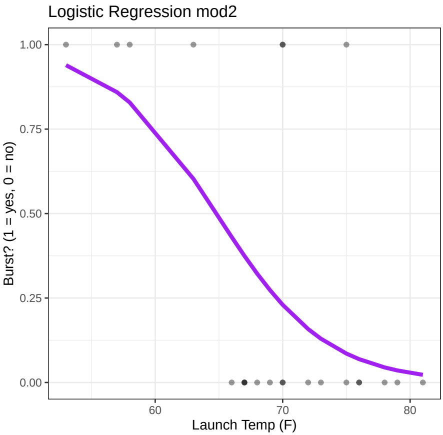
- Results on next slide

Plotting Model `m2`



Comparing the fits of `mod1` and `mod2`...





Could we try exponentiating the `mod2` coefficients?

How can we interpret the coefficients of the model?

$$logit(burst) = log(odds(burst)) = 15.043 - 0.232 temp$$

Exponentiating the coefficients is helpful...

```
exp(-0.232)
```

```
[1] 0.7929461
```

Suppose Launch A's temperature was one degree higher than Launch B's.

- The **odds** of Launch A having a burst are 0.793 times as large as they are for Launch B.
- Odds Ratio estimate comparing two launches whose `temp` differs by 1 degree is 0.793

Exponentiated and tidied `mod2` coefficients

```
tidy(mod2, exponentiate = TRUE, conf.int = TRUE) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	3412315.488	7.379	27.955	8.214986e+14
temp	0.793	0.108	0.597	9.410000e-01

- What would it mean if the Odds Ratio for `temp` was 1?
- How about an odds ratio that was greater than 1?

Building the `smart3` tibble

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

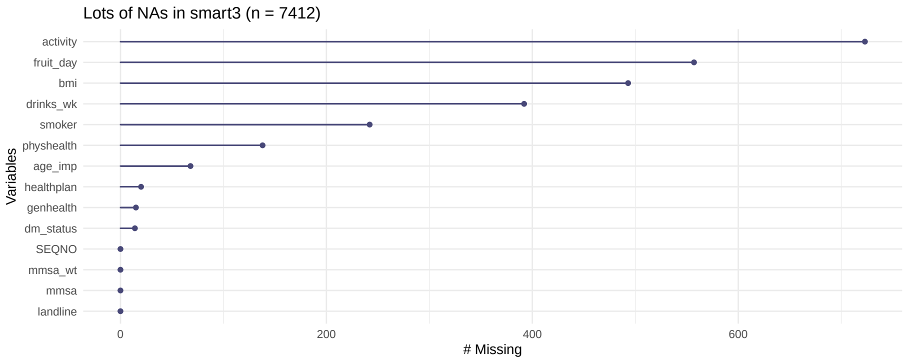
Variable	Type	Description
`landline`	Binary (1/0)	survey conducted by landline? (vs. cell)
`healthplan`	Binary (1/0)	subject has health insurance?
`age_imp`	Quantitative	age (imputed from groups - see Notes)
`fruit_day`	Quantitative	mean servings of fruit / day
`drinks_wk`	Quantitative	mean alcoholic 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, *we'll collapse to 2*)
`activity`	Categorical	physical activity level (4 levels, *we'll re-level*)
`smoker`	Categorical	smoking status (4 levels, *we'll collapse to 3*)
`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"))
```

Visualizing Missingness in Variables

```r
gg_miss_var(smart3) +
  labs(title = "Lots of NAs in smart3 (n = 7412)")
```



Creating a “Shadow” to track what is imputed

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

smart3_sh creates new variables, ending in _NA

```
names(smart3_sh)
```

```
 [1] "SEQNO"          "mmsa"           "mmsa_wt"
 [4] "landline"       "age_imp"        "healthplan"
 [7] "dm_status"      "fruit_day"      "drinks_wk"
[10] "activity"       "smoker"         "physhealth"
[13] "bmi"            "genhealth"      "SEQNO_NA"
[16] "mmsa_NA"        "mmsa_wt_NA"     "landline_NA"
[19] "age_imp_NA"     "healthplan_NA" "dm_status_NA"
[22] "fruit_day_NA"   "drinks_wk_NA"   "activity_NA"
[25] "smoker_NA"      "physhealth_NA" "bmi_NA"
[28] "genhealth_NA"
```

What are the new variables tracking?

```
smart3_sh %>% count(smoker, smoker_NA)
```

```
# A tibble: 4 x 3
  smoker  smoker_NA     n
  <fct>   <fct>     <int>
1 Current !NA        1290
2 Former  !NA        1999
3 Never   !NA        3881
4 <NA>    NA          242
```

The `fct_explicit_na` warning: A pain point

My general preference is to not use `fct_explicit_na`, and if I see a warning about that, I typically suppress it from printing.

"Simple" Imputation Strategy

```
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()
```

Check to see that imputation worked...

Before imputation, what fraction of our cases are complete?

```
pct_complete_case(smart3)
```

```
[1] 81.08473
```

After imputation, do any of our cases have missing values?

```
pct_miss_case(smart3_sh)
```

```
[1] 0
```

Saving the `smart3` and `smart3_sh` tibbles to `.Rds`

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

Today's Questions

Can we predict Prob(BMI < 30) for a subject in the `smart3_sh` data:

- using the mean number of servings of fruit per day that they consume?
- using their diabetes status?

Using fruit servings consumed per day to predict Prob(BMI $<$ 30)

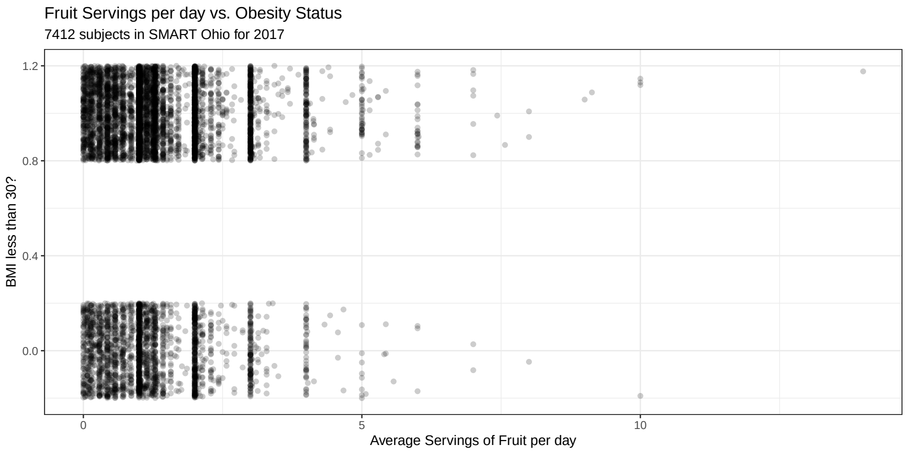
Predicting Prob(BMI < 30)

```
smart3_sh <- smart3_sh %>%
  mutate(bmilt30 = as.numeric(bmi < 30),
         dm_status = fct_relevel(dm_status, "No"))

smart3_sh %>% tabyl(bmilt30) %>% adorn_pct_formatting()
```

```
 bmilt30    n percent
       0 2343   31.6%
       1 5069   68.4%
```

Association of BMI < 30 and Fruit Consumption

Plot includes some vertical jitter and shading to the plot



Model `m1` for Prob(BMI < 30)

Linear Probability Model for Prob(BMI < 30)?

```
m1 <- smart3_sh %$% lm(bmilt30 ~ fruit_day)

tidy(m1, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	0.645	0.009	0.628	0.662
fruit_day	0.029	0.005	0.019	0.039

Linear Probability Model to predict BMI < 30?

```
tidy(m1, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 3)
```

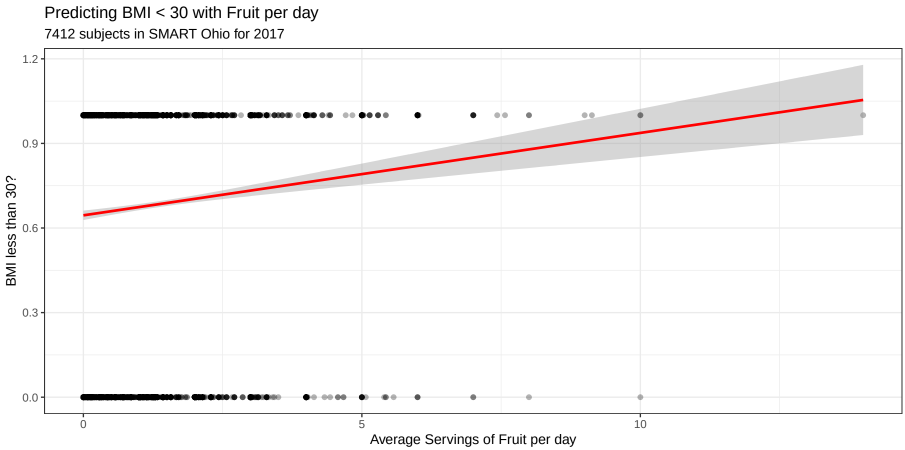
term	estimate	std.error	conf.low	conf.high
(Intercept)	0.645	0.009	0.628	0.662
fruit_day	0.029	0.005	0.019	0.039

- What's the predicted probability of BMI < 30 if a subject eats 5 servings of fruit per day?

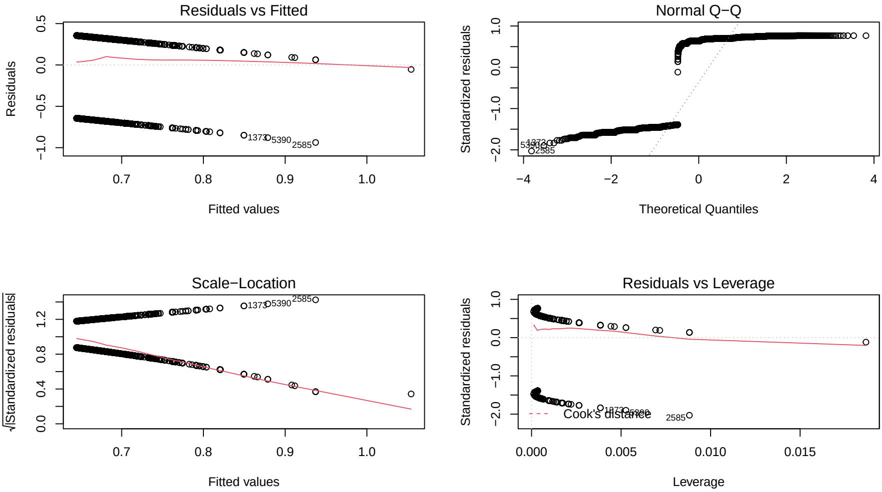
$$Pr(BMI < 30) = 0.645 + 0.029(5) = 0.645 + 0.145 = 0.790$$

- What's the predicted probability of BMI < 30 if a subject eats no fruit?

Linear Probability Model `m1` predicting BMI < 30



Residual Plots for the Linear Probability Model (`m1`)



Model `m2` for Prob(BMI < 30)

Logistic Regression for Prob(BMI < 30)

We'll use the `glm` function in R, specifying a logistic regression model.

- We're now predicting $log(odds(BMI < 30))$ or $logit(BMI < 30)$.

```
m2 <- smart3_sh %$%
  glm(bmilt30 ~ fruit_day, family = binomial)

tidy(m2, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	0.583	0.040	0.505	0.662
fruit_day	0.145	0.025	0.097	0.194

Our model `m2`

$\text{logit}(\text{BMI} < 30) = \log(\text{odds}(\text{BMI} < 30)) = 0.583 + 0.145 \text{ fruit\_day}$

- If Rebecca consumes 5 servings per day, what is the prediction?

$\log(\text{odds}(\text{BMI} < 30)) = 0.583 + 0.145\ (5) = 0.583 + 0.725 = 1.308$

- Exponentiate to get the odds, on our way to estimating the probability.

$\text{odds}(\text{BMI} < 30) = \exp(1.308) = 3.699$

- so, we can estimate Rebecca's Probability of BMI < 30 as...

$$Pr(BMI < 30) = \frac{3.699}{(3.699 + 1)} = 0.787.$$

Another Prediction

What is the predicted probability of BMI < 30 if a subject (Keeley) eats no fruit?

- $\log(\text{odds}(\text{BMI} < 30)) = 0.583 + 0.145\ (0) = 0.583$
- $\text{odds}(\text{BMI} < 30) = \exp(0.583) = 1.791$
- $\Pr(\text{BMI} < 30) = 1.791\ /\ (1.791 + 1) = 0.642$

Can we use `augment` for this?

Will augment do this, as well?

```
new2 <- tibble( fruit_day = c(0, 5) )

augment(m2, newdata = new2, type.predict = "link")
```

```
# A tibble: 2 x 2
  fruit_day .fitted
      <dbl>   <dbl>
1         0   0.583
2         5   1.31
```

```
augment(m2, newdata = new2, type.predict = "response")
```

```
# A tibble: 2 x 2
  fruit_day .fitted
      <dbl>   <dbl>
1         0   0.642
2         5   0.787
```

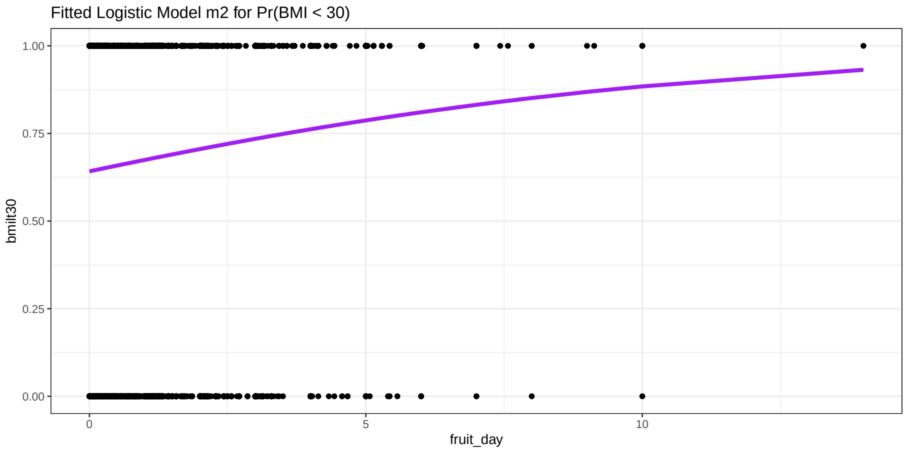
Plotting the Logistic Regression Model

Use the augment function to get the fitted probabilities into the original data, then plot.

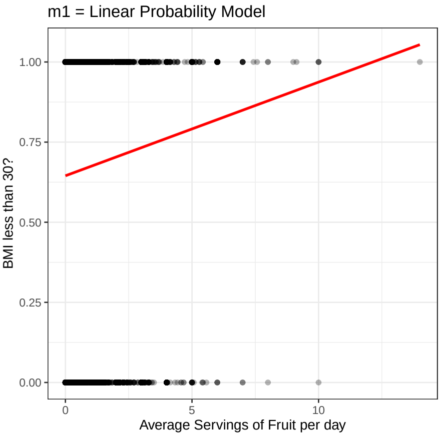
```
m2_aug <- augment(m2, type.predict = "response")

ggplot(m2_aug, aes(x = fruit_day, y = bmilt30)) +
  geom_point() +
  geom_line(aes(x = fruit_day, y = .fitted),
            col = "purple", size = 1.5) +
  labs(title = "Fitted Logistic Model m2 for Pr(BMI < 30)")
```

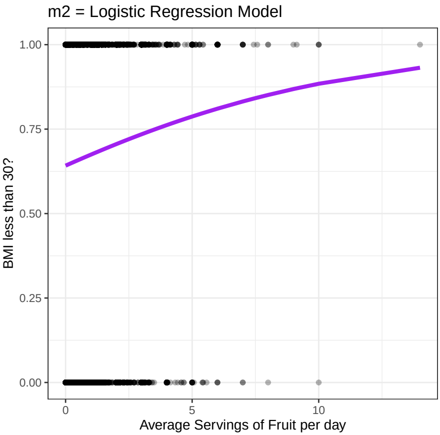
- Results on next slide

Plotting Model `m2`



Comparing the fits of `m1` and `m2`...





Exponentiating the m2 coefficients?

How can we interpret the coefficients of the model?

$$logit(BMI < 30) = log(odds(BMI < 30)) = 0.583 + 0.145 fruit$$

Exponentiating the coefficients is helpful...

```
exp(coef(m2))
```

```
(Intercept)   fruit_day
   1.792206    1.156012
```

Suppose Ted ate one more piece of fruit per day than Roy.

- The **odds** of Ted having BMI < 30 are 1.156 times as large as they are for Roy.
- Odds Ratio estimate comparing two subjects whose `fruit_day` differ by 1 serving is 1.156.

Exponentiated and tidied m2 coefficients

```
tidy(m2, exponentiate = TRUE, conf.int = TRUE) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	1.792	0.040	1.656	1.939
fruit_day	1.156	0.025	1.101	1.214

- What would it mean if the Odds Ratio for `fruit_day` was 1?
- If Ted eats more servings of fruit than Roy, what would an odds ratio for `fruit_day` that was greater than 1 mean?
- How about an odds ratio that was less than 1?
- What is the smallest possible Odds Ratio?

m2: some additional output

```
m2
```

```
Call:  glm(formula = bmilt30 ~ fruit_day, family = binomial)

Coefficients:
(Intercept)    fruit_day
     0.5834       0.1450

Degrees of Freedom: 7411 Total (i.e. Null);  7410 Residual
Null Deviance:      9249
Residual Deviance: 9213     AIC: 9217
```

- Think of the Deviance as a measure of "lack of fit".
- Deviance accounted for by m2 is
  - 9249 - 9213 = 36 points on 7411 - 7410 = 1 df
- Can do a likelihood ratio test via anova.

anova(m2) for our logistic regression model

```
anova(m2, test = "LRT")
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: bmilt30

Terms added sequentially (first to last)

          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                       7411     9248.7
fruit_day  1   35.744      7410     9213.0 2.251e-09 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

m2: output from glance

```
glance(m2) %>% select(1,2,6,7,3)
```

```
# A tibble: 1 x 5
  null.deviance df.null deviance df.residual logLik
          <dbl>   <int>    <dbl>       <int>  <dbl>
1         9249.    7411    9213.        7410 -4606.
```

```
glance(m2) %>% select(4,5,8)
```

```
# A tibble: 1 x 3
    AIC   BIC  nobs
  <dbl> <dbl> <int>
1 9217. 9231.  7412
```

- AIC and BIC still useful for comparing models using the same outcome.
- The deviance is -2(log likelihood).
- Elements of the difference-in-deviance statistic are here.

Comparing models `m1` and `m2` via AIC/BIC

We have `m1` and `m2` so far. Each predicts `BMI < 30` using `fruit_day`, but `m1` uses the linear probability model, and `m2` the logistic regression model.

```
bind_rows(glance(m1) %>% select(AIC, BIC),
          glance(m2) %>% select(AIC, BIC)) %>%
  mutate(mod = c("m1 (Lin. Prob.)", "m2 (Logistic)")) %>%
  kable(digits = 1)
```

AIC	BIC	mod
9653.7	9674.4	m1 (Lin. Prob.)
9217.0	9230.8	m2 (Logistic)

By AIC and BIC, which model looks better?

Get predictions for all subjects in our data

```
m1_aug <- augment(m1)
m2_aug <- augment(m2, type.predict = "response")
```

The predicted probabilities are in the .fitted column.

```
m1_aug %>% select(bmilt30, .fitted) %>% slice(1)
```

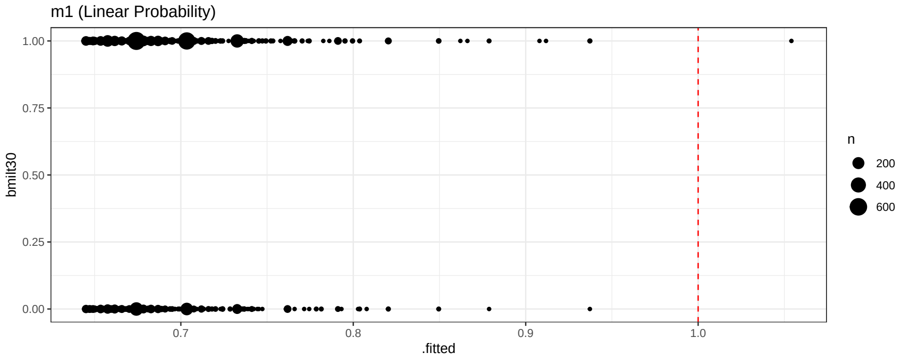
```
# A tibble: 1 x 2
  bmilt30 .fitted
    <dbl>   <dbl>
1       1   0.687
```

```
m2_aug %>% select(bmilt30, .fitted) %>% slice(1)
```

```
# A tibble: 1 x 2
  bmilt30 .fitted
    <dbl>   <dbl>
1       1   0.688
```

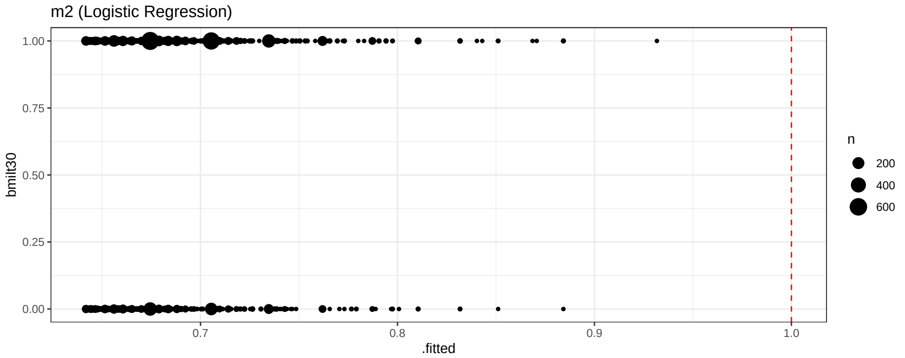
Plot observed vs. predicted values for m1

```
ggplot(m1_aug, aes(x = .fitted, y = bmilt30)) +
  geom_count() +
  geom_vline(xintercept = 1, col = "red", lty = "dashed") +
  labs(title = "m1 (Linear Probability)")
```



Plot observed vs. predicted values for m2

```
ggplot(m2_aug, aes(x = .fitted, y = bmilt30)) +
  geom_count() +
  geom_vline(xintercept = 1, col = "red", lty = "dashed") +
  labs(title = "m2 (Logistic Regression)")
```



Making Classification Decisions

- Our outcome is `bmilt30`, where `bmilt30` $= 1$ if BMI < 30, and otherwise `bmilt30` $= 0$.
- We establish a classification rule based on our model's predicted probabilities of BMI < 30.
- 0.5 is a natural cut point but not inevitable. We'll use 0.65!
  - If .fitted is below 0.65, we'll predict that `bmilt30` $= 0$.
  - If .fitted is 0.65 or larger, we'll predict that `bmilt30` $= 1$.

```
m2_aug %$% table(.fitted >= 0.65, bmilt30)
```

```
       bmilt30
           0    1
  FALSE  291  504
  TRUE  2052 4565
```

Standard Epidemiological Format

```
confuse_m2 <- m2_aug %>%
  mutate(bmilt30_act = factor(bmilt30 == "1"),
     bmilt30_pre = factor(.fitted >= 0.65),
     bmilt30_act = fct_relevel(bmilt30_act, "TRUE"),
     bmilt30_pre = fct_relevel(bmilt30_pre, "TRUE")) %$%
  table(bmilt30_pre, bmilt30_act)

confuse_m2
```

```
           bmilt30_act
bmilt30_pre TRUE FALSE
      TRUE  4565  2052
      FALSE  504   291
```

(Mis-)Classification Table / Confusion Matrix

```
confuse_m2
```

```
           bmilt30_act
bmilt30_pre TRUE FALSE
      TRUE  4565  2052
      FALSE  504   291
```

- Total Observations: 4565 + 2052 + 504 + 291 = 7412
- Correct Predictions: 4565 + 291 = 4856, or 65.5% **accuracy**
- Incorrect Predictions: 504 + 2052 = 2556 (34.5%)
- Actual TRUE: 4565 + 504 = 5069, or 68.4% **prevalence**
- Predicted TRUE: 4565 + 2052 = 6617, or 89.3% **detection prevalence**

Other Summaries from a Confusion Matrix

```
confuse_m2
```

```
            bmilt30_act
bmilt30_pre TRUE FALSE
      TRUE  4565  2052
      FALSE  504   291
```

- **Sensitivity** $= 4565 / (4565 + 504) = 90.1\%$ (also called Recall)
  - if the subject actually has BMI < 30 our model predicts that 90.1% of the time.
- **Specificity** $= 291 / (2052 + 291) = 12.4\%$
  - if the subject actually has BMI $>= 30$ our model predicts that 12.4% of the time.
- **Positive Predictive Value** or *Precision* $= 4565 / (4565 + 2052) = 69.0\%$
  - our predictions of BMI < 30 were correct 69.0% of the time.
- **Negative Predictive Value** $= 291 / (291 + 504) = 36.6\%$
  - our predictions that BMI $>= 30$ were correct 36.6% of the time.

Confusion matrix for models `m1` and `m2`

We can obtain a similar confusion matrix for model `m1` using the same (arbitrary) cutoff of `.fitted >= 0.65` to indicate a predicted BMI < 30.

```
confuse_m1
```

```
           bmilt30_act
bmilt30_pre TRUE FALSE
      TRUE  4633  2084
      FALSE  436   259
```

```
confuse_m2
```

```
           bmilt30_act
bmilt30_pre TRUE FALSE
      TRUE  4565  2052
      FALSE  504   291
```

Which of these confusion matrices looks better?

Using diabetes status to predict Prob(BMI $<$ 30): model `m3`

Predicting BMI < 30 using diabetes status (a factor)

```
m3 <- smart3_sh %$%
  glm(bmilt30 ~ dm_status,
      family = binomial(link = logit))

tidy(m3) %>% select(term, estimate) %>%
  knitr::kable(digits = 3)
```

term	estimate
(Intercept)	0.947
dm_statusYes	-1.053

Equation: `logit(BMI < 30) = 0.947 - 1.053 (dm_status = Yes)`

How can we interpret this result?

Interpreting the `m3` Logistic Regression Equation

`logit(BMI < 30) = 0.947 - 1.053 (dm_status = Yes)`

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

Now, `logit(BMI < 30)` = `log(odds(BMI < 30))`, so exponentiate to get the odds...

- Harry has predicted `odds(BMI < 30)` = exp(-0.106) = 0.899
- Sally has predicted `odds(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?

`logit(BMI < 30) = 0.947 - 1.053 (dm_status = Yes)`

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

Odds Ratio for 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 BMI $<$ 30.

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

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

Tidy with exponentiation

```
tidy(m3, exponentiate = TRUE,
     conf.int = TRUE, conf.level = 0.9) %>%
  select(term, estimate, conf.low, conf.high) %>%
  knitr::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 BMI < 30, then the odds would also be the same, and so the odds ratio would be 1.
- If the probabilities of BMI < 30 were the same and thus the odds were the same, then the log odds ratio would be `log(1)` $= 0$.

```
logit(BMI < 30) = 0.947 - 1.053 (dm_status = Yes)
```

1. If the log odds of a coefficient (like `diabetes = 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 `diabetes = No`?

New model: `logit(BMI < 30) = 0.947 + 1.053 (dm_status = No)`

Next Time

- Binary regression models with multiple predictors

Coming Next Week (Class 7)

- Using `ols` to fit a linear model: A preview
  - Spearman ρ^2 plots and data spending
  - ANOVA results
  - Plot effects with `summary` and `Predict`
  - Creating and interpreting a nomogram
  - Validating summary statistics: R^2 and MSE