

432 Class 05 Slides

thomaseLove.github.io/432

2022-01-25

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

Setup

```
library(conflicted)                                # a new idea
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
conflict_prefer("filter", "dplyr") # choose over stats
options(dplyr.summarise.inform = FALSE)
```

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

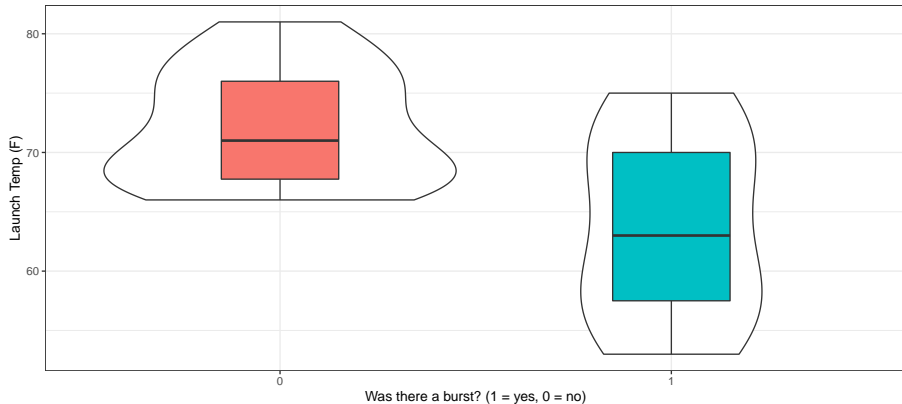
Code to plot burst and temp in our usual way...

```
ggplot(orings1, aes(x = factor(burst), y = temp)) +  
  geom_violin() +  
  geom_boxplot(aes(fill = factor(burst)), width = 0.3) +  
  guides(fill = "none") +  
  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

Are bursts more common at low temperatures?

23 prior space shuttle launches



What if we want to predict $\text{Prob}(\text{burst})$ using temp?

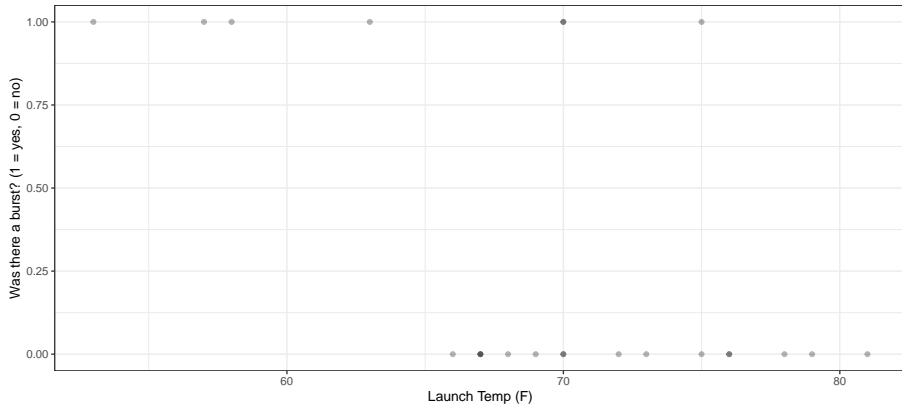
We want to treat the binary variable burst as the outcome, and temp as the predictor...

```
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

Are bursts more common at low temperatures

23 prior space shuttle launches



Fit a linear model to predict Prob(burst)?

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

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

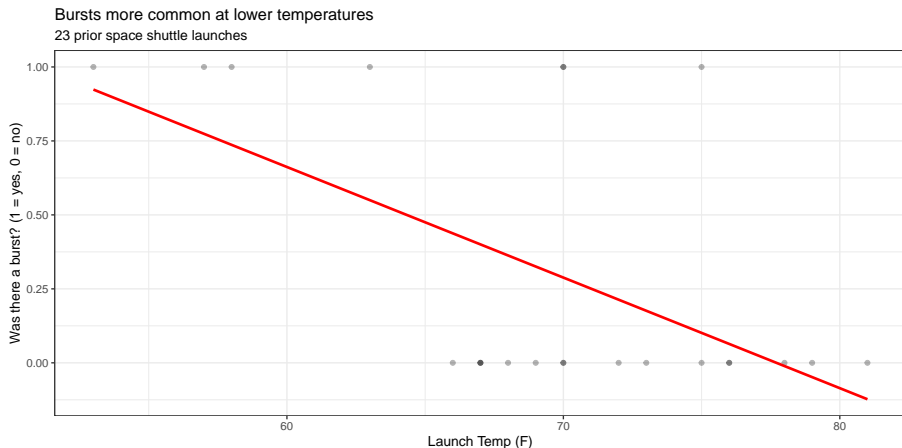
term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	2.905	0.842	3.450	0.002	1.154	4.656
temp	-0.037	0.012	-3.103	0.005	-0.062	-0.012

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

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

$$\widehat{\text{burst}} = 2.905 - 0.037(\text{temp}) \quad (1)$$

Add linear probability model to our plot?



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

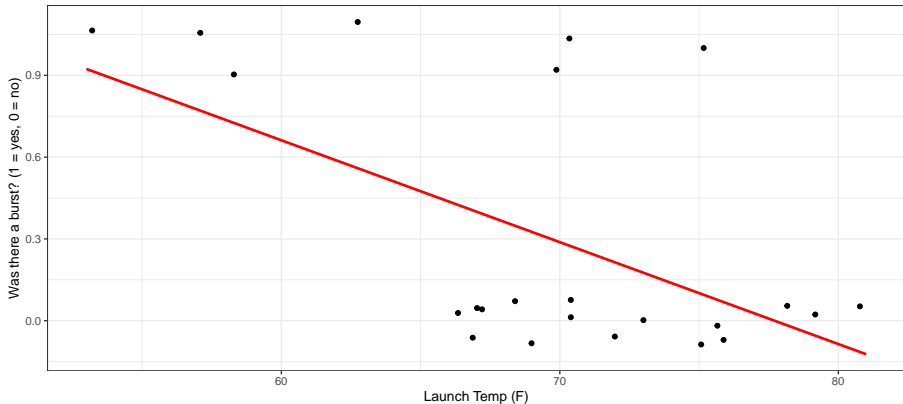
Add vertical jitter and our mod1 model?

```
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

Bursts more common at lower temperatures

23 prior space shuttle launches



- 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))
```

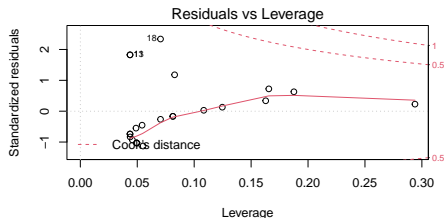
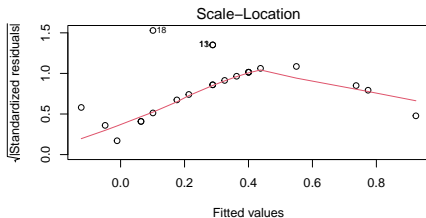
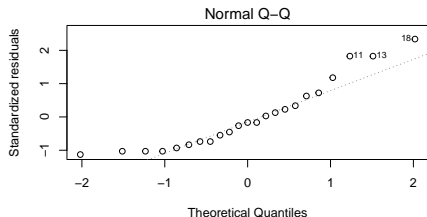
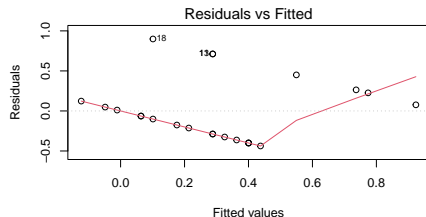
```
augment(mod1, newdata = newtemps)
```

```
# 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

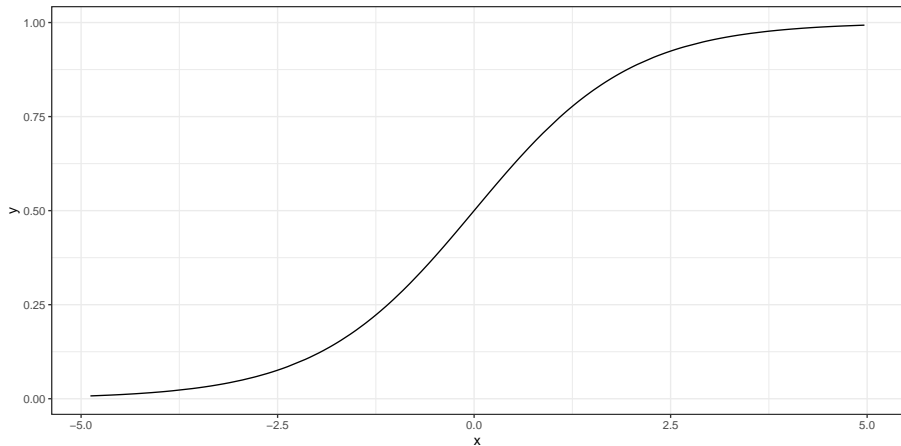
The function we use in logistic regression is called the **logit link**.

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

The inverse of the logit function is called the **logistic function**. If $\text{logit}(\pi) = \eta$, 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.

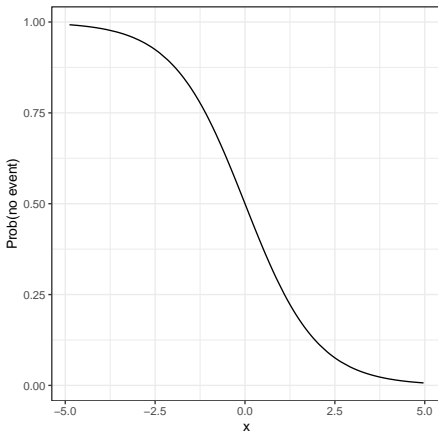
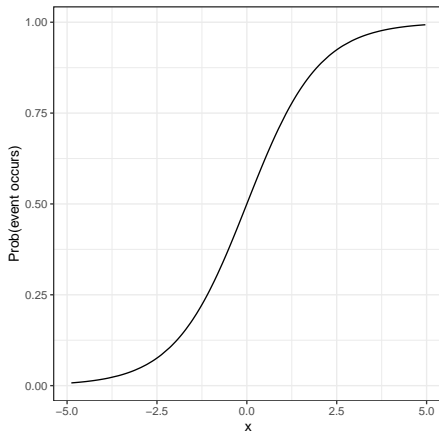
- If we have a probability $\pi < 0.5$, then $\text{logit}(\pi) < 0$.
- If our probability $\pi > 0.5$, then $\text{logit}(\pi) > 0$.
- Finally, if $\pi = 0.5$, then $\text{logit}(\pi) = 0$.

Why is this helpful?

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

Predicting $\Pr(\text{event})$ or $\Pr(\text{no event})$

- 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 $\text{logit}(burst)$.

```
mod2 <- glm(burst ~ temp, data = orings1,  
            family = binomial(link = "logit"))  
  
tidy(mod2, conf.int = TRUE) %>%  
  select(term, estimate, std.error, conf.low, conf.high) %>%  
  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

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

$$\log \left[\frac{P(\widehat{\text{burst}} = 1)}{1 - P(\widehat{\text{burst}} = 1)} \right] = 15.0429 - 0.2322(\text{temp}) \quad (2)$$

$$\text{logit}(\text{burst}) = \log(\text{odds}(\text{burst})) = 15.0429 - 0.2322\text{temp}$$

- 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?

$$\log(\text{odds}(\text{burst})) = 15.0429 - 0.2322 (70) = -1.211$$

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

$$\text{odds}(\text{burst}) = \exp(-1.211) = 0.2979$$

- so, we can estimate the probability by

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

Prediction from mod2 for temp = 60

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

- $\log(\text{odds}(\text{burst})) = 15.0429 - 0.2322 (60) = 1.1109$
- $\text{odds}(\text{burst}) = \exp(1.1109) = 3.0371$
- $\text{Pr}(\text{burst}) = 3.0371 / (3.0371 + 1) = 0.752$

Will augment do this, as well?

```
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

```
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

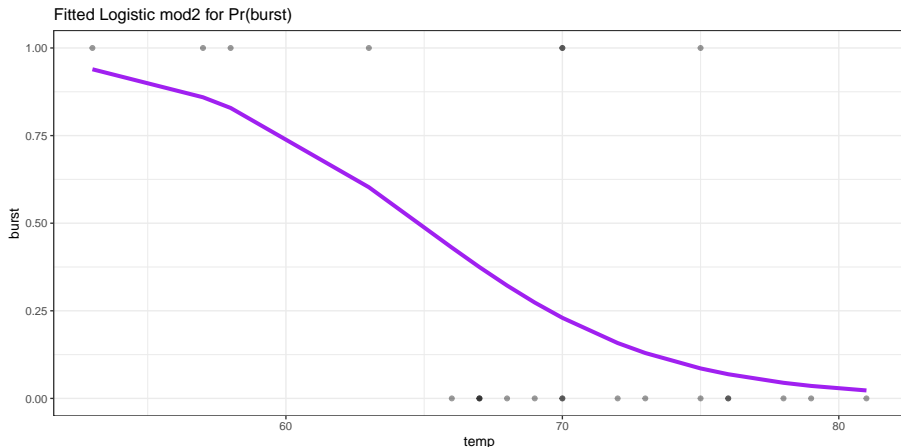
Use the `augment` function to get the fitted probabilities into the original data, then plot.

```
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)")
```

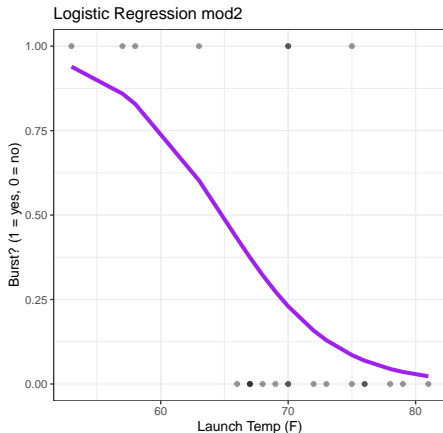
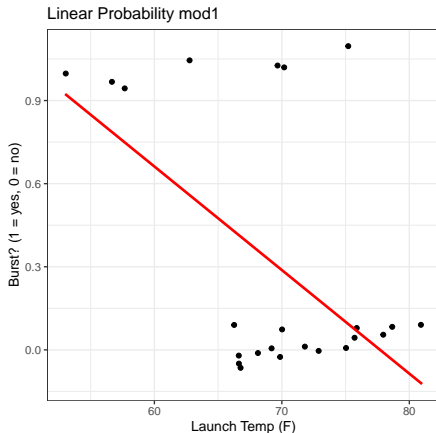
- Results on next slide

Plotting Model `m2`



Note that we're just connecting the predictions made for observed `temp` values with `geom_line`, so the appearance of the function isn't as smooth as the actual logistic regression model.

Comparing the fits of mod1 and mod2...



Could we try exponentiating the mod2 coefficients?

How can we interpret the coefficients of the model?

$$\text{logit}(\text{burst}) = \log(\text{odds}(\text{burst})) = 15.043 - 0.232\text{temp}$$

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 slope of temp (mod2)

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

term	estimate	std.error	conf.low	conf.high
temp	0.793	0.108	0.597	0.941

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

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

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

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

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

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

Building a 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, <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)

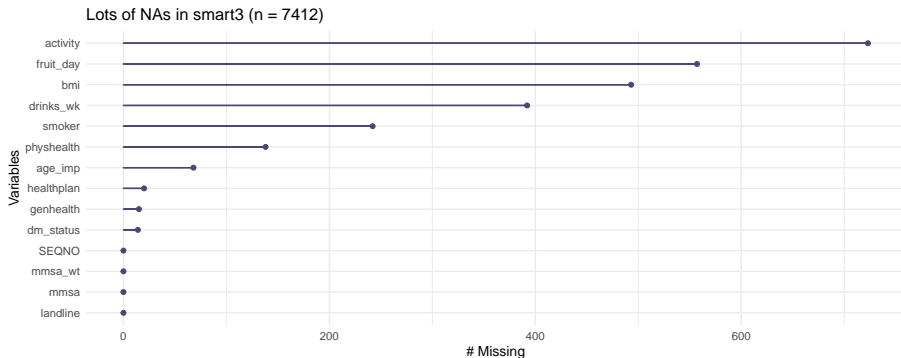
Collapsing Two Factors, Re-leveling another

```
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"))
```

Visualizing Missingness in Variables

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

Warning: It is deprecated to specify ``guide = FALSE`` to remove a guide. Please use ``guide = "none"`` instead.



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(2022432)
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")
```

Using diabetes status (yes/no) to predict
whether $\text{BMI} > 30$

Create binary outcome variable

```
smart3_sh <- readRDS("data/smart3_sh.Rds") %>%  
  mutate(bmigt30 = as.numeric(bmi > 30),  
         dm_status = fct_relevel(dm_status, "No"))  
  
smart3_sh %>%  
  group_by(bmigt30) %>%  
  summarize(n = n(), mean(bmi), min(bmi), max(bmi)) %>%  
  kable(digits = 2)
```

bmigt30	n	mean(bmi)	min(bmi)	max(bmi)
0	5074	25.26	13.30	30.00
1	2338	35.85	30.01	75.52

Predicting BMI > 30 using diabetes status (a factor)

```
mod_DM <- smart3_sh %$%  
  glm(bmigt30 ~ dm_status,  
      family = binomial(link = logit))  
  
tidy(mod_DM) %>% select(term, estimate) %>%  
  kable(digits = 3)
```

term	estimate
(Intercept)	-0.949
dm_statusYes	1.048

Equation: $\text{logit}(\text{BMI} > 30) = -0.949 + 1.048 (\text{dm_statusYes})$

How can we interpret this result?

Interpreting the mod_DM Equation

$$\text{logit}(\text{BMI} > 30) = -0.949 + 1.048 (\text{dm_statusYes})$$

- Harry has diabetes.
 - His predicted $\text{logit}(\text{BMI} > 30)$ is $-0.949 + 1.048 (1) = 0.099$
- Sally does not have diabetes.
 - Her predicted $\text{logit}(\text{BMI} > 30)$ is $-0.949 + 1.048 (0) = -0.949$

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

- Harry has predicted $\text{odds}(\text{BMI} > 30) = \exp(0.099) = 1.104$
- Sally has predicted $\text{odds}(\text{BMI} > 30) = \exp(-0.949) = 0.387$

Can we convert these odds into something more intuitive?

Converting Odds to Probabilities

- Harry has predicted $\text{odds}(\text{BMI} > 30) = \exp(0.099) = 1.104$
- Sally has predicted $\text{odds}(\text{BMI} > 30) = \exp(-0.949) = 0.387$

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

and

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

- So Harry's predicted $\text{Pr}(\text{BMI} > 30) = 1.104 / 2.104 = 0.52$
- Sally's predicted $\text{Pr}(\text{BMI} > 30) = 0.387 / 1.387 = 0.28$
- odds range from 0 to ∞ , and $\log(\text{odds})$ range from $-\infty$ to ∞ .
- odds > 1 if probability > 0.5 . If odds = 1, then probability = 0.5.

What about the odds ratio?

$\text{logit}(\text{BMI} > 30) = -0.949 + 1.048 (\text{dm_statusYes})$

- Harry, with diabetes, has $\text{odds}(\text{BMI} > 30) = 1.104$
- Sally, without diabetes, has $\text{odds}(\text{BMI} > 30) = 0.387$

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

$$\frac{1.104}{0.387} = 2.85$$

- Our model estimates that a subject with diabetes has 2.85 times the odds (285% 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, $\exp(1.048) = 2.85$.

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)	0.387	0.369	0.405
dm_statusYes	2.851	2.556	3.181

- The odds ratio for BMI > 30 among subjects with diabetes as compared to those without diabetes is 2.851
- The odds of BMI > 30 are 285.1% as large (2.851 times 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 (2.556, 3.181).

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 $\log(1) = 0$.

$\text{logit}(\text{BMI} > 30) = -0.949 + 1.048 (\text{dm_statusYes})$

- 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`?

Flipping the model changes slope and intercept!

```
mod_DM_no <- smart3_sh %$%  
  glm(bmigt30 ~ (dm_status == "No"),  
      family = binomial(link = logit))  
  
tidy(mod_DM_no) %>% select(term, estimate) %>%  
  kable(digits = 3)
```

term	estimate
(Intercept)	0.098
dm_status == "No"TRUE	-1.048

Old: $\text{logit}(\text{BMI} > 30) = -0.949 + 1.048 (\text{dm_statusYes})$

New: $\text{logit}(\text{BMI} > 30) = 0.098 - 1.048 (\text{dm_status} = \text{No})$

Predictions from the two models?

DMYes: $\text{logit}(\text{BMI} > 30) = -0.949 + 1.048 \text{ (dm_status = Yes)}$

DMNo: $\text{logit}(\text{BMI} > 30) = 0.098 - 1.048 \text{ (dm_status = No)}$

Harry lives with diabetes. Sally does not.

Using the DMYes model:

- $\text{logit}(\text{Harry's BMI} > 30) = -0.949 + 1.048 = 0.098$
- $\text{logit}(\text{Sally's BMI} > 30) = -0.949$

Using the DMNo model:

- $\text{logit}(\text{Harry's BMI} > 30) = 0.098$
- $\text{logit}(\text{Sally's BMI} > 30) = 0.098 - 1.048 = -0.949$

Comparison to the 2x2 Table Results?

```
smart3_sh %>% tabyl(bmigt30, dm_status)
```

bmigt30	No	Yes
0	4551	523
1	1761	577

That's not quite the 2x2 table we want.

We want to switch the order of both variables

```
temp1 <- smart3_sh %>%  
  select(bmigt30, dm_status, SEQNO)  
  
temp1 <- temp1 %>%  
  mutate(bmigt30 = fct_relevel(factor(bmigt30), "1"),  
         dm_status = fct_relevel(dm_status, "Yes"))  
  
temp1 %>% tabyl(bmigt30, dm_status)
```

	bmigt30 Yes	No
1	577	1761
0	523	4551

Resulting 2x2 Table Result

```
temp1 %$% Epi::twoby2(bmigt30, dm_status)
```

2 by 2 table analysis:

Outcome : Yes

Comparing : 1 vs. 0

	Yes	No	P(Yes)	95% conf. interval
1	577	1761	0.2468	0.2297 0.2647
0	523	4551	0.1031	0.0950 0.1117

		95% conf. interval
Relative Risk:	2.3943	2.1498 2.6666
Sample Odds Ratio:	2.8512	2.5024 3.2486
Conditional MLE Odds Ratio:	2.8506	2.4969 3.2554
Probability difference:	0.1437	0.1246 0.1633

Using smoking status (multi-categorical) to
predict diabetes (yes/no)

Can we use smoker to predict dm_status?

```
smart3_sh %>% tabyl(smoker, dm_status) %>%  
  adorn_totals() %>%  
  adorn_percentages(den = "row") %>%  
  adorn_pct_formatting() %>%  
  adorn_ns(position = "front")
```

smoker	No		Yes	
Current	1170	(87.8%)	163	(12.2%)
Former	1689	(81.9%)	373	(18.1%)
Never	3453	(86.0%)	564	(14.0%)
Total	6312	(85.2%)	1100	(14.8%)

Logistic Regression for dm_status by smoker

```
mod_SM <- glm(dm_status ~ smoker,  
              family = binomial(link = "logit"),  
              data = smart3_sh)  
  
tidy(mod_SM) %>% select(term, estimate) %>% kable(dig = 3)
```

term	estimate
(Intercept)	-1.971
smokerFormer	0.461
smokerNever	0.159

What is being fit, exactly?

```
extract_eq(mod_SM, use_coefs = TRUE, coef_digits = 3,  
           wrap = TRUE, terms_per_line = 2, label = NA)
```

$$\log \left[\frac{P(\widehat{\text{dm_status}} = \text{Yes})}{1 - P(\widehat{\text{dm_status}} = \text{Yes})} \right] = -1.971 + 0.461(\text{smoker}_{\text{Former}}) + 0.159(\text{smoker}_{\text{Never}}) \quad (3)$$

Resulting Predictions from mod_SM

$\text{logit}(\text{dm} = \text{Yes}) = -1.971 + 0.461 \text{ Former} + 0.159 \text{ Never}$

- from logit to odds via exponentiation

Smoking Status	logit(DM = Yes)	odds(DM = Yes)
Current	-1.971	$\exp(-1.971) = 0.139$
Former	$-1.971 + 0.461 = -1.510$	$\exp(-1.510) = 0.221$
Never	$-1.971 + 0.159 = -1.812$	$\exp(-1.812) = 0.163$

- convert from odds to probabilities (do these match our table?)

Smoking Status	odds(DM = Yes)	Pr(DM = Yes)
Current	0.139	$.139/1.139 = 0.122$
Former	0.221	$.221/1.221 = 0.181$
Never	0.163	$.163/1.163 = 0.140$

Next Time

- Binary regression models with multiple predictors
- Assessing the quality of fit for a logistic model