432 Class 6 Slides

github.com/THOMASELOVE/432-2018

2018-02-01

Setup

```
library(skimr)
library(broom)
library(modelr)
library(leaps)
library(tidyverse)

oh_count <- read.csv("data/counties2017a.csv") %>% tbl_df
lbw <- read.csv("data/lbw.csv") %>% tbl_df
```

Today's Materials

- Ohio County Health Rankings Data
- Variable Selection via Best Subsets
- Cross-Validating to Compare Model-Building Approaches
- Assessing Residual Diagnostic Plots
- Dealing with Non-Linearity: Spending Degrees of Freedom

Last time, we looked at Ohio County Health Rankings Data http://www.countyhealthrankings.org/rankings/data/oh

Codebook (2017 County Health Rankings), I

Variable	Description
`fips`	FIPS code for county (an ID)
`state`	Ohio in all cases
`county`	County Name (88 counties in Ohio)
`years_lost`	Years of potential life lost before age 75 per 100,000 population (age-adjusted, 2012-14)
`population`	County population, Census Population Estimates, 2015
`female`	% female (Census Population Estimates, 2015)
`rural`	3 categories from % rural (0-20: Urban, 20.1-50: Suburban, 50.1+: Rural; Census 2015)
`non_white`	4 categories from 100 - % white non-hispanic: (> 20: High, 10.1-20: Medium, 5.1-10: Low, <=5: Very Low, Census 2015)

Codebook (2017 County Health Rankings), II

Variable	Description
`sroh_fairpoor`	% of adults reporting fair or poor health (age-adjusted via 2015 BRFSS)
`smoker_pct`	% of adults who currently smoke (2015 BRFSS)
`food_envir`	Food environment index (0 = worst, 10 = best) (via USDA Map the Meal 2014)
`exer_access`	% of population with adequate access to locations for physical activity (several sources)
`income_ratio`	Ratio of household income at the 80th percentile to income at the 20th percentile (ACS 2011-15)
`air_pollution`	Mean daily density of fine particulate matter in micrograms per cubic meter (PM2.5)
`health_costs`	Health Care Costs (from Dartmouth Atlas, 2014)

Using "Best Subsets" to Select Variables

Using "Best Subsets" to Select Variables

We'll consider models using some combination of the 11 available meaningful predictors.

```
bs_preds <- with(oh_count, cbind(population, female, rural,
                                 non_white, sroh_fairpoor,
                                 smoker_pct, food_envir,
                                 exer_access, income_ratio,
                                 air_pollution, health_costs))
```

We'll look for models using up to 8 of those predictors.

```
bs_subs <- regsubsets(bs_preds,
                      y = oh_count$years_lost,
                      nvmax = 8)
bs_mods <- summary(bs_subs)

bs_mods$aic.c <- 88*log(bs_mods$rss / 88) + 2*(2:9) +
    (2 * (2:9) * ((2:9)+1) / (88 - (2:9) - 1))
```

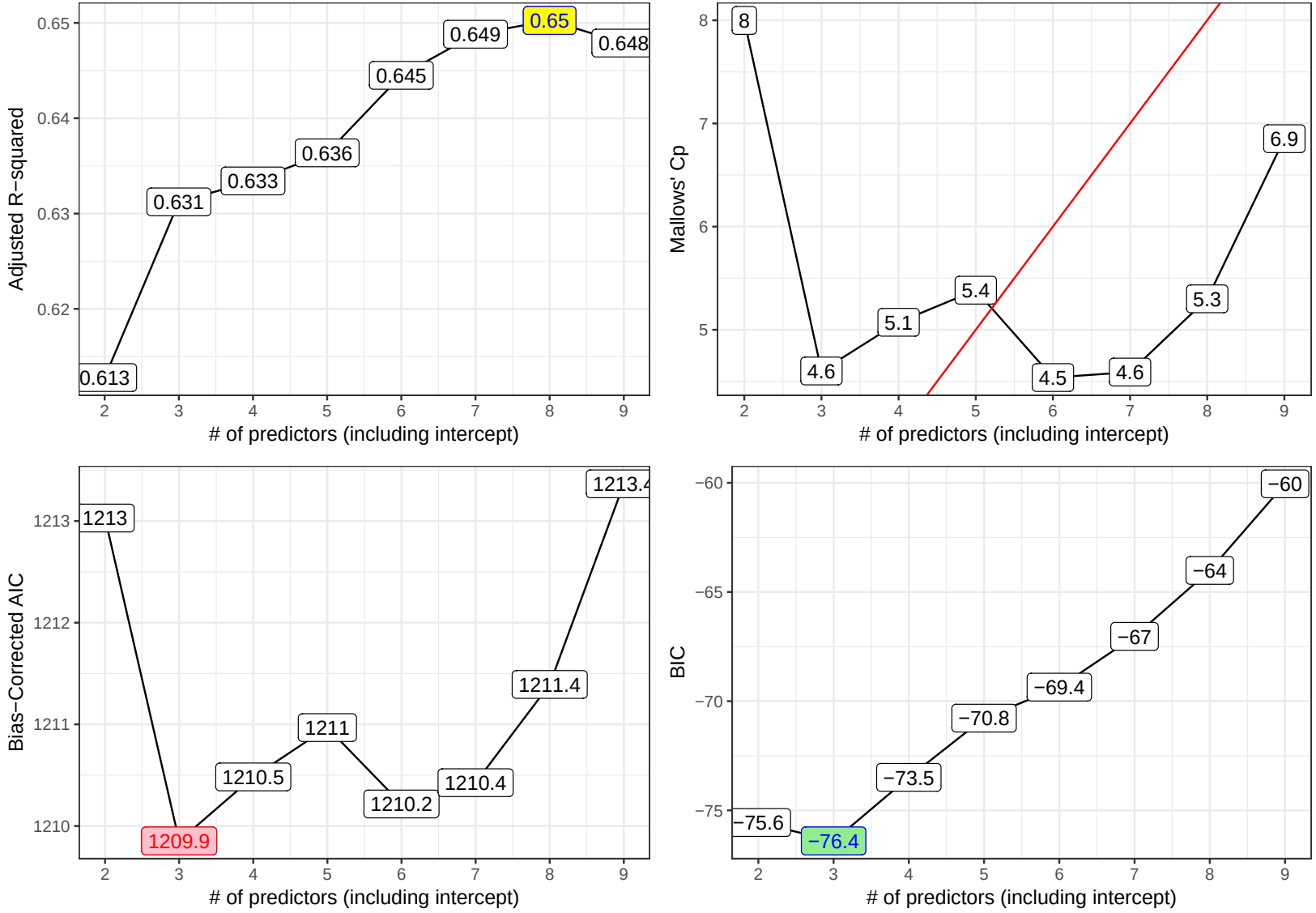
Place winning results in bs_winners

```
bs_winners <- tbl_df(bs_mods$which)
bs_winners$k <- 2:9 ## in general, this is 2:(nvmax + 1)
bs_winners$r2 <- bs_mods$rsq
bs_winners$adjr2 <- bs_mods$adjr2
bs_winners$cp <- bs_mods$cp
bs_winners$aic.c <- bs_mods$aic.c
bs_winners$bic <- bs_mods$bic
```

Building the "Best Subsets" Plots

Code not shown here, but it's in the Markdown file.

The Four Plots



Candidate Models include

Inputs	Raw r^2	Adj. r^2	C_p	BIC	AIC_c
3	.640	.631	4.6	**-76.4**	**1209.9**
5	.653	.636	**5.4**	-70.8	1211.0
8	.678	**.650**	5.3	-64.0	1211.4

- 3: `smoker_pct` + `health_costs`
- 5: Model 3 + `food_envir` + `income_ratio`
- 8: Model 5 + `female` + `exer_access` + `sroh_fairpoor`

Comparing our Candidate Models in our Training Sample

In-Sample Comparisons of our Candidate Models

```
m3 <- lm(years_lost ~ smoker_pct + health_costs,
         data = oh_count)
m5 <- lm(years_lost ~ smoker_pct + health_costs +
             food_envir + income_ratio, data=oh_count)
m8 <- lm(years_lost ~ smoker_pct + health_costs +
              food_envir + income_ratio + female +
             exer_access + sroh_fairpoor, data=oh_count)
```

Models are **nested** so comparisons within samples are straightforward.

Comparisons in-sample with anova

```r
anova(m3, m5, m8)
```

```
Analysis of Variance Table

Model 1: years_lost ~ smoker_pct + health_costs
Model 2: years_lost ~ smoker_pct + health_costs + food_envir +
Model 3: years_lost ~ smoker_pct + health_costs + food_envir +
    female + exer_access + sroh_fairpoor
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1     85 76610187
2     83 73764357  2   2845831 1.6647 0.1957
3     80 68382551  3   5381806 2.0987 0.1069
```

Comparisons in-sample with AIC

```
a <- AIC(m3, m5, m8)
b <- BIC(m3, m5, m8); b$model <- row.names(b)
left_join(a, b)
```

```
Joining, by = "df"
```

```
  df      AIC      BIC model
1  4 1461.301 1471.210    m3
2  6 1461.970 1476.834    m5
3  9 1461.303 1483.599    m8
```

What if the models you're comparing aren't nested?

What if you're comparing:

- Model A: `lm(y = x1 + x2 + x3, data = dataset)`
- Model B: `lm(y = x1 + x4 + x5, data = dataset)`

Then . . .

- default p values from the ANOVA table comparing Model A to Model B aren't reasonable
- AIC and BIC are OK, can also used adjusted R^2 to help make a decision within the model building sample
- Still useful to think about out-of-sample prediction and cross-validation

Comparing out-of-sample predictive ability of our Candidate Models with cross-validation

10-fold Cross-Validation for Model 3

```
set.seed(432012)

cv_3 <- oh_count %>%
  crossv_kfold(k = 10) %>%
  mutate(model = map(train, ~ lm(years_lost ~
                      smoker_pct + health_costs, data = .)))

cv3_pred <- cv_3 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv3_res <- cv3_pred %>%
  summarize(Model = "3",
            RMSE = sqrt(mean((years_lost - .fitted) ^2)),
            MAE = mean(abs(years_lost - .fitted)))
```

10-fold Cross-Validation for Model 5

```
set.seed(432013)

cv_5 <- oh_count %>%
  crossv_kfold(k = 10) %>%
  mutate(model = map(train, ~ lm(years_lost ~
                      smoker_pct + health_costs +
                      food_envir + income_ratio, data = .)))

cv5_pred <- cv_5 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv5_res <- cv5_pred %>%
  summarize(Model = "5",
            RMSE = sqrt(mean((years_lost - .fitted) ^2)),
            MAE = mean(abs(years_lost - .fitted)))
```

10-fold Cross-Validation for Model 8

```
set.seed(432014)

cv_8 <- oh_count %>%
  crossv_kfold(k = 10) %>%
  mutate(model = map(train, ~ lm(years_lost ~
                      smoker_pct + health_costs +
                      food_envir + income_ratio +
                      female + exer_access +
                      sroh_fairpoor, data = .)))

cv8_pred <- cv_8 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv8_res <- cv8_pred %>%
  summarize(Model = "8",
            RMSE = sqrt(mean((years_lost - .fitted) ^2)),
            MAE = mean(abs(years_lost - .fitted)))
```

Cross-Validation Results

```r
bind_rows(cv3_res, cv5_res, cv8_res)
```

```
# A tibble: 3 x 3
  Model  RMSE   MAE
  <chr> <dbl> <dbl>
1 3       975   785
2 5       976   797
3 8      1004   809
```

Fitting the Chosen Model

Fitting the Chosen Model

```
m3 <- lm(years_lost ~ smoker_pct + health_costs,
         data = oh_count)

arm::display(m3)
```

```
lm(formula = years_lost ~ smoker_pct + health_costs, data = oh
             coef.est coef.se
(Intercept)  -5749.51  1248.81
smoker_pct     517.62    61.10
health_costs     0.34     0.15
---
n = 88, k = 3
residual sd = 949.37, R-Squared = 0.64
```

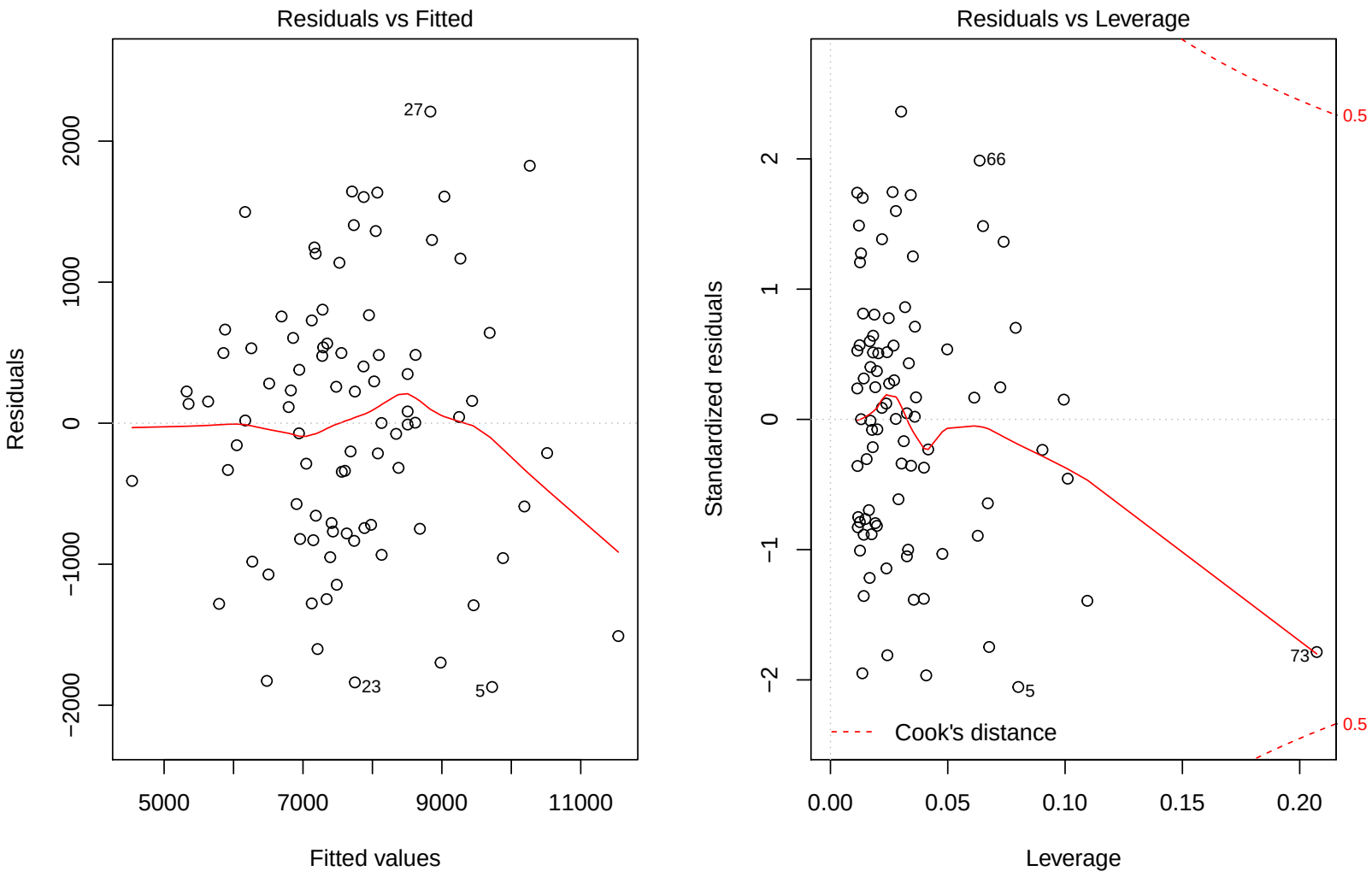
Fitting the Chosen Model

```
glance(m3)
```

```
  r.squared adj.r.squared    sigma statistic      p.value
1  0.639703     0.6312255 949.3663  75.45825 1.439049e-19
  df    logLik      AIC     BIC deviance df.residual
1  3 -726.6504 1461.301 1471.21 76610187          85
```

Residual Plots for the Chosen Model

```
par(mfrow = c(1,2)); plot(m3, which = c(1, 5))
```



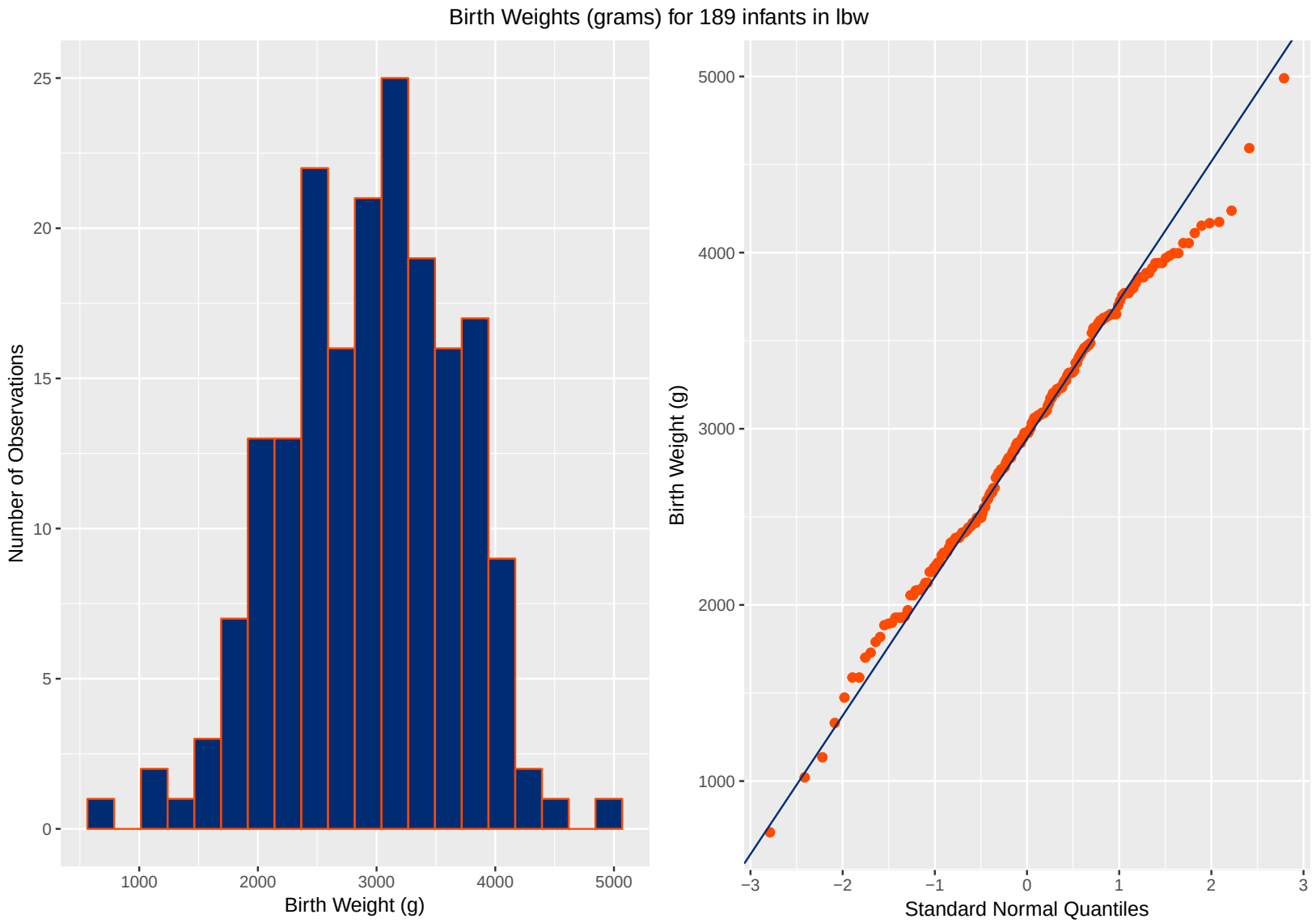
The Low Birth Weight Data (`lbw.csv`) from Hosmer and Lemeshow and Sturdivant, 3rd edition

Code Book (n = 189 infants)

Variable	Description
`subject`	id code
`low`	indicator of low birth weight ($<$ 2500 g)
`age`	age of mother in years
`lwt`	mom's weight at last menstrual period (lbs.)
`race`	1 = white, 2 = black, 3 = other
`smoke`	1 = smoked during pregnancy, 0 = did not
`ptl`	count of prior premature labors (we see 0, 1, 2, 3)
`ht`	history of hypertension: 1 = yes, 0 = no
`ui`	presence of uterine irritability: 1 = yes, 0 = no
`ftv`	count of physician visits in first trimester (0 to 6)
`bwt`	recorded birth weight (in g)

Data from Baystate Medical Center, Springfield MA in 1986.

A closer look at our outcome, `bwt`



Code for Plot on Previous Slide

```
slo <- diff( quantile(lbw$bwt, c(0.25, 0.75)) ) /
    diff( qnorm(c(0.25, 0.75)) )
int <- quantile(lbw$bwt, c(0.25, 0.75))[1L] -
    slo * qnorm(c(0.25, 0.75))[1L]

p1 <- ggplot(lbw, aes(x = bwt)) +
    geom_histogram(bins = 20,
                   fill = "#002C74", col = "#FF4A00") +
    labs(x = "Birth Weight (g)",
         y = "Number of Observations")
```

(continues on next slide)

```r
p2 <- ggplot(lbw, aes(sample = bwt)) +
    geom_qq(col = "#FF4A00", size = 2) +
    geom_abline(intercept = int, slope = slo,
                col = "#002C74") +
    labs(y = "Birth Weight (g)",
         x = "Standard Normal Quantiles")

gridExtra::grid.arrange(p1, p2, nrow = 1,
   top = "Birth Weights (grams) for 189 infants in lbw")
```

Specifying some factors

1. Specify `race` as a factor (`race_f`), and order its levels "White", "Black", "Other".
2. Specify that the 1/0 variables `ht`, `smoke` and `ui` are 1/0 factors.
3. Specify `preterm` as a yes/no factor with yes meaning ptl > 0, so no means ptl $= 0$

```
lbw <- lbw %>%
    mutate(race_f = fct_recode(factor(race), white = "1",
                               black = "2", other = "3"),
         race_f = fct_relevel(race_f, "white", "black")) %>%
    mutate_at(c("ht", "smoke", "ui"), funs(factor(.))) %>%
    mutate(preterm = fct_recode(factor(ptl > 0),
                                yes = "TRUE",
                                no = "FALSE"))
```

Describing the Data

```
lbw %>% select(-subject, -low, -race, -ptl) %>% skim()
```

```
Skim summary statistics
 n obs: 189
 n variables: 9

Variable type: factor
 variable missing complete   n n_unique                        top_counts ordered
       ht       0      189 189        2             0: 177, 1: 12, NA: 0   FALSE
  preterm       0      189 189        2           no: 159, yes: 30, NA: 0   FALSE
   race_f       0      189 189        3 whi: 96, oth: 67, bla: 26, NA: 0   FALSE
    smoke       0      189 189        2             0: 115, 1: 74, NA: 0   FALSE
       ui       0      189 189        2             0: 161, 1: 28, NA: 0   FALSE

Variable type: integer
 variable missing complete   n    mean     sd  p0  p25 median  p75 p100     hist
      age       0      189 189   23.24   5.3  14   19     23   26   45 ▅▇▅▂▁▁▁▁
      bwt       0      189 189 2944.66 729.02 709 2414   2977 3475 4990 ▂▅▇▇▅▂▁▁
      ftv       0      189 189    0.79   1.06   0    0      0    1    6 ▇▂▂▁▁▁▁▁
      lwt       0      189 189  129.81  30.58  80  110    121  140  250 ▅▇▅▂▁▁▁▁
```

Building the best predictor subsets to predict bwt

We'll build the best model of size 2:9 again, but this time, forcing in the `lwt` variable.

```
lbw.out <- regsubsets(bwt ~ age + race_f + smoke + ftv +
                          lwt + ht + ui + preterm,
                      data = lbw, nvmax = NULL, nbest = 1,
                      force.in = c("lwt"))

lbw.sum <- summary(lbw.out)
```

Results of `lbw.sum`

```
> lbw.sum
Subset selection object
Call: regsubsets.formula(bwt ~ age + race_f + smoke + ftv + lwt + ht +
    ui + preterm, data = lbw, nvmax = 8, nbest = 1, force.in = c("lwt"))
9 Variables  (and intercept)
            Forced in Forced out
lwt             FALSE      FALSE
age             FALSE      FALSE
race_fblack     FALSE      FALSE
race_fother     FALSE      FALSE
smoke1          FALSE      FALSE
ftv              TRUE      FALSE
ht1             FALSE      FALSE
ui1             FALSE      FALSE
pretermyes      FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         lwt age race_fblack race_fother smoke1 ftv ht1 ui1 pretermyes
2  ( 1 ) "*" " " " "         " "         " "    " " " " "*" " "
3  ( 1 ) "*" " " " "         " "         " "    " " "*" "*" " "
4  ( 1 ) "*" " " "*"         " "         " "    " " "*" "*" " "
5  ( 1 ) "*" " " "*"         "*"         "*"    " " " " "*" " "
6  ( 1 ) "*" " " "*"         "*"         "*"    " " "*" "*" " "
7  ( 1 ) "*" " " "*"         "*"         "*"    " " "*" "*" "*"
8  ( 1 ) "*" " " "*"         "*"         "*"    "*" "*" "*" "*"
```

Building the corrected AIC values

Data includes `nrow(lbw)` = 189 observations, and we run models of size 2:9, when you include the intercept term.

```
lbw.sum$aic.c <- 189*log(lbw.sum$rss / 189) + 2*(2:9) +
    (2 * (2:9) * ((2:9)+1) / (189 - (2:9) - 1))
```

Place winning results in lbw_win

```
lbw_win1 <- data_frame(
    k = 2:9,
    r2 = lbw.sum$rsq,
    adjr2 = lbw.sum$adjr2,
    cp = lbw.sum$cp,
    aic.c = lbw.sum$aic.c,
    bic = lbw.sum$bic)

lbw_win <- bind_cols(lbw_win1, tbl_df(lbw.sum$which))
```

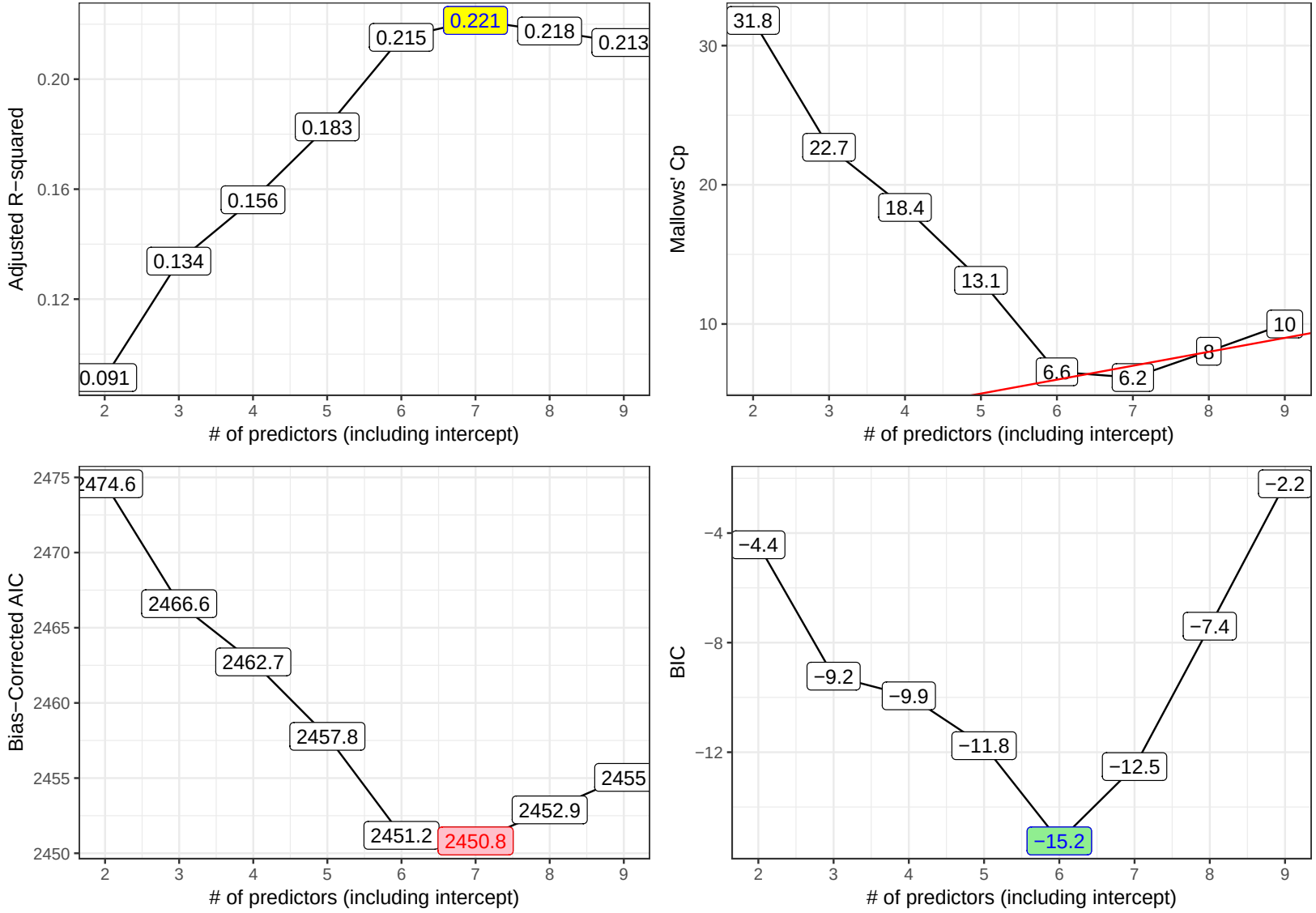
View lbw_win

```
> lbw_win
# A tibble: 8 x 16
      k    r2  adjr2    cp aic.c    bic `(Intercept)` lwt   age   race_fblack race_fother smoke1 ftv   ht1   ui1   pretermyes
  <int> <dbl>  <dbl> <dbl> <dbl>  <dbl> <lgl>         <lgl> <lgl> <lgl>       <lgl>       <lgl> <lgl> <lgl> <lgl> <lgl>
1     2 0.101 0.0915 31.8   2475 - 4.43 T             T     F     F           F           F     F     F     T     F
2     3 0.148 0.134  22.7   2467 - 9.22 T             T     F     F           F           F     F     T     T     F
3     4 0.174 0.156  18.4   2463 - 9.94 T             T     F     T           F           F     F     T     T     F
4     5 0.204 0.183  13.1   2458 -11.8  T             T     F     T           T           T     F     F     T     F
5     6 0.240 0.215   6.56  2451 -15.2  T             T     F     T           T           T     F     T     T     F
6     7 0.250 0.221   6.15  2451 -12.5  T             T     F     T           T           T     F     T     T     T
7     8 0.251 0.218   8.04  2453 - 7.40 T             T     F     T           T           T     T     T     T     T
8     9 0.251 0.213  10.0   2455 - 2.20 T             T     T     T           T           T     T     T     T     T
```

Building The Four Plots for `lbw`

Code in R Markdown file. . .

The Four Plots



Candidate Models are of sizes k = 6 and k = 7

```
lbw_win %>% filter(k %in% c(6, 7))
```

```
> lbw_win %>% filter(k %in% c(6, 7))
# A tibble: 2 x 16
      k    r2 adjr2    cp aic.c   bic `(Intercept)` lwt   age   race_fblack race_fother smoke1 ftv   ht1   ui1   pretermyes
  <int> <dbl> <dbl> <dbl> <dbl> <dbl> <lgl>         <lgl> <lgl> <lgl>       <lgl>       <lgl> <lgl> <lgl> <lgl> <lgl>
1     6 0.240 0.215  6.56  2451 -15.2 T             T     F     T           T           T     F     T     T     F
2     7 0.250 0.221  6.15  2451 -12.5 T             T     F     T           T           T     F     T     T     T
```

The candidate models are:

```
lbw_m6 <- lm(bwt ~ lwt + race_f + smoke + ht + ui,
             data = lbw)
lbw_m7 <- lm(bwt ~ lwt + race_f + smoke + ht + ui + preterm,
             data = lbw)
```

ANOVA comparison of lbw_m6 and lbw_m7

```r
anova(lbw_m6, lbw_m7)
```

```
Analysis of Variance Table

Model 1: bwt ~ lwt + race_f + smoke + ht + ui
Model 2: bwt ~ lwt + race_f + smoke + ht + ui + preterm
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1    182 75911729
2    181 74902970  1   1008759 2.4376 0.1202
```

AIC and BIC within-sample comparisons

```r
AIC(lbw_m6, lbw_m7)
```

```
       df      AIC
lbw_m6  8 2991.089
lbw_m7  9 2990.561
```

```r
BIC(lbw_m6, lbw_m7)
```

```
       df      BIC
lbw_m6  8 3017.023
lbw_m7  9 3019.736
```

5-fold cross-validation of lbw_m6

```
set.seed(43202201)

cv_lbw6 <- lbw %>%
  crossv_kfold(k = 5) %>%
  mutate(model = map(train, ~ lm(bwt ~ lwt + race_f +
                                     smoke + ht + ui,
                                 data = .)))

cv_lbw6_pred <- cv_lbw6 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv_lbw6_results <- cv_lbw6_pred %>%
  summarize(Model = "lbw_m6",
            RMSE = sqrt(mean((bwt - .fitted) ^2)),
            MAE = mean(abs(bwt - .fitted)))
```

5-fold cross-validation of lbw_m7

```
set.seed(43202202)

cv_lbw7 <- lbw %>%
  crossv_kfold(k = 5) %>%
  mutate(model = map(train, ~ lm(bwt ~ lwt + race_f +
                                     smoke + ht + ui +
                                     preterm,
                                 data = .)))

cv_lbw7_pred <- cv_lbw7 %>%
  unnest(map2(model, test, ~ augment(.x, newdata = .y)))

cv_lbw7_results <- cv_lbw7_pred %>%
  summarize(Model = "lbw_m7",
            RMSE = sqrt(mean((bwt - .fitted) ^2)),
            MAE = mean(abs(bwt - .fitted)))
```

Comparison on cross-validated prediction error summaries

```
bind_rows(cv_lbw6_results, cv_lbw7_results)
```

```
# A tibble: 2 x 3
  Model   RMSE   MAE
  <chr>  <dbl> <dbl>
1 lbw_m6   657   536
2 lbw_m7   670   542
```

It looks like `lbw_m6` is a little better in terms of predictive accuracy.

What if we included an interaction term?

What if we include an interaction between `race_f` and `smoke`?

- This time, we won't force anything into the model.
- This doesn't work nicely with interactions including a multi-categorical variable like `race_f`.

```
lbw.out2 <- regsubsets(bwt ~ age + race_f * smoke + ftv +
                           lwt + ht + ui + preterm,
                       data = lbw, nvmax = 6, nbest = 1)

lbw.sum2 <- summary(lbw.out2)
```

Results of `lbw.sum2$which`, transposed

```
> t(lbw.sum2$which)
                      1     2     3     4     5     6
(Intercept)        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
age               FALSE FALSE FALSE FALSE FALSE FALSE
race_fblack       FALSE FALSE FALSE  TRUE  TRUE  TRUE
race_fother       FALSE FALSE FALSE  TRUE  TRUE  TRUE
smoke1            FALSE FALSE FALSE  TRUE  TRUE  TRUE
ftv               FALSE FALSE FALSE FALSE FALSE FALSE
lwt               FALSE FALSE  TRUE FALSE FALSE  TRUE
ht1               FALSE  TRUE  TRUE FALSE FALSE  TRUE
ui1                TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
pretermyes        FALSE FALSE FALSE FALSE FALSE FALSE
race_fblack:smoke1 FALSE FALSE FALSE FALSE FALSE FALSE
race_fother:smoke1 FALSE FALSE FALSE FALSE  TRUE FALSE
```

Models Identified as "Winners" in `lbw.sum2`

k	Predictors
2	`ui`
3	`ui ht`
4	`ui ht lwt`
5	`ui race_fblack race_fother smoke`
6	`ui race_fblack race_fother smoke race_fother:smoke`

And how do we interpret an interaction term that doesn't use all of the levels in `race_f`?

Limitations of "Best Subsets"

- Works only with quantitative outcomes (linear regression)
- Useful only for variable selection of main effects
- Generates a useful pool of candidate models, but doesn't usually center all of its energy on the same model
- Doesn't take into account potential product terms

Possible Solutions for the last issue:

1. Consider interactions beforehand, force them in.
2. Consider interaction terms only after selection of main effects.
3. Do something else entirely.

Next Week

- Spending Degrees of Freedom on Non-Linearity
- The Spearman ρ^2 (rho-squared) plot
- Building Non-Linear Predictors with
  - Polynomial Functions
  - Product Terms
  - Splines, including Restricted Cubic Splines
- Building a Nomogram for a Linear Regression Model
- Getting Started with Logistic Regression