432 Class 6 Slides

github.com/THOMASELOVE/2019-432

2019-02-12

Today's Example: Low Birth Weight

Linear Regression on a Quantitative Outcome

- Using "best subsets" variable selection while forcing in a predictor
- Comparing Candidate Models (training and cross-validation)
- What about interaction terms?
- Limitations of Best Subsets

Logistic Regression on a Binary outcome

- Problems with the Linear Probability Model
- The Logit Link and Logistic Function
- Using `glm` to fit the model and make predictions
- Interpreting the Model: Odds
- Summarizing and Evaluating the Model

Setup

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

lbw <- read_csv("data/lbw.csv") %>% clean_names()
```

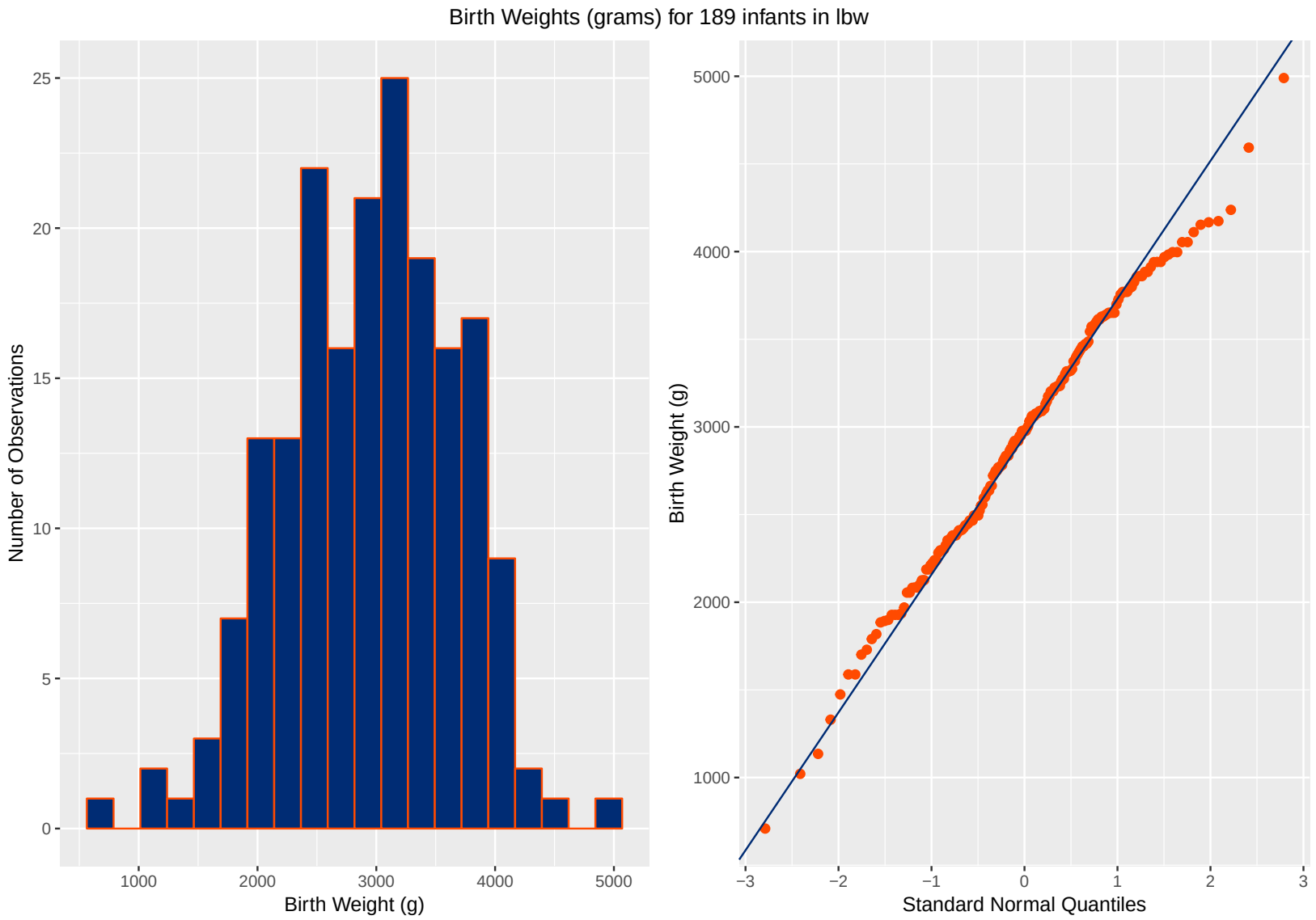
The Low Birth Weight Data (`lbw.csv`) from Hosmer and Lemeshow and Sturdivant, 3rd edition

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 $= 0$

```
lbw1 <- 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

```
lbw1 %>% 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 ▅▇▅▂▁▁▁▁
```

Best Subsets and Predicting `bwt`

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 = lbw1, 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)
```

```
Warning: `data_frame()` is deprecated, use `tibble()`.
This warning is displayed once per session.
```

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

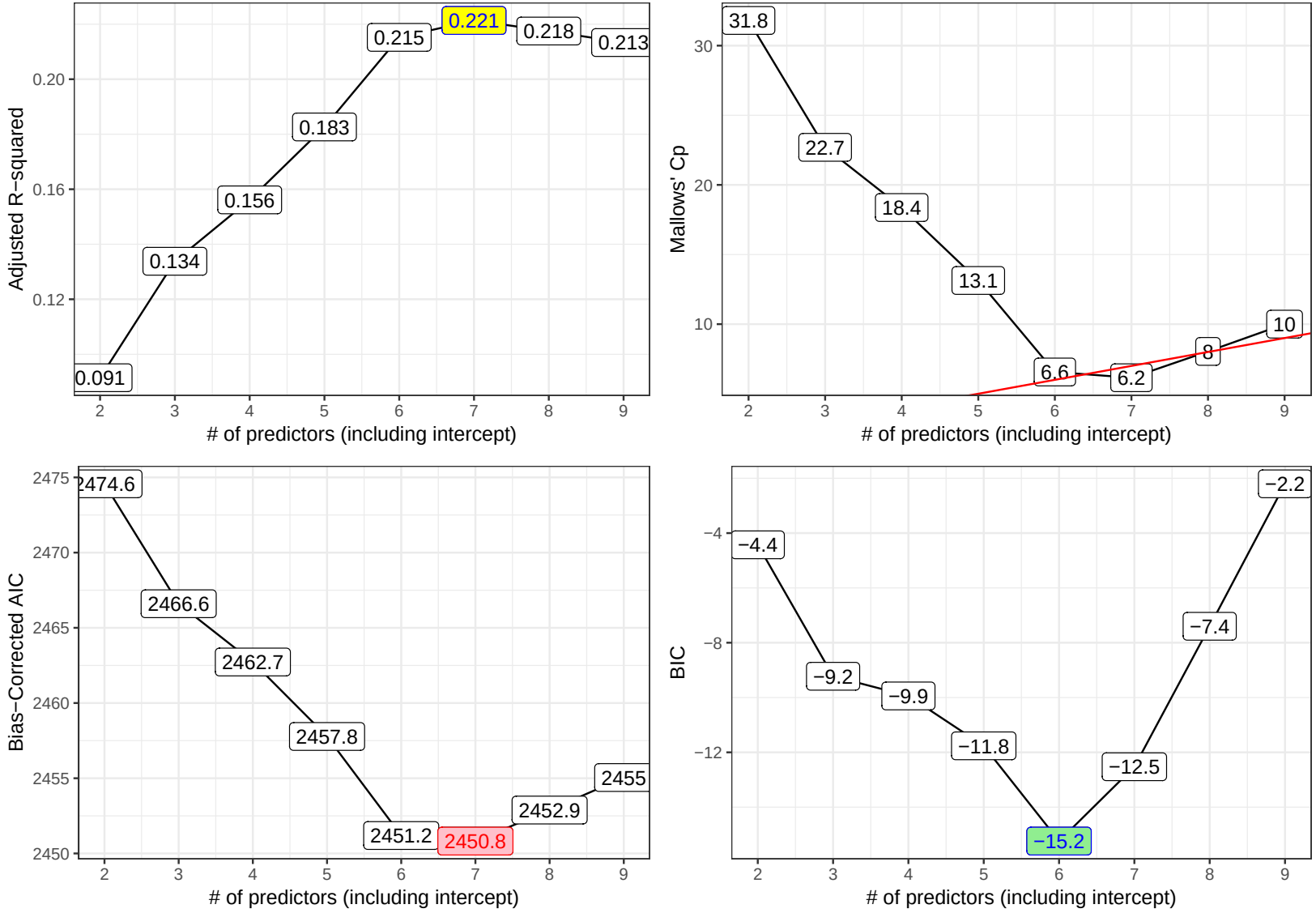
View lbw_win

```
> 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 = lbw1)
lbw_m7 <- lm(bwt ~ lwt + race_f + smoke + ht + ui + preterm,
             data = lbw1)
```

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 <- lbw1 %>%
  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 <- lbw1 %>%
  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 = lbw1, 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.

Logistic Regression

Goals for Today and Next Time

1. Fit and evaluate the fit of a logistic regression model to predict the probability of a low birth weight (`low` = 1) using the mom's weight at her last menstrual period (`lwt`).
2. Fit and evaluate a larger logistic regression model to predict `low` on the basis of a larger group of predictors drawn from the available options, which include: `lwt`, `age`, `ftv`, `ht`, `race_f`, `preterm`, `smoke` and `ui`.
3. Learn about the use of both `glm` and `lrm` (from the `rms` package) to fit and evaluate logistic regression models.

EDA for Task 1

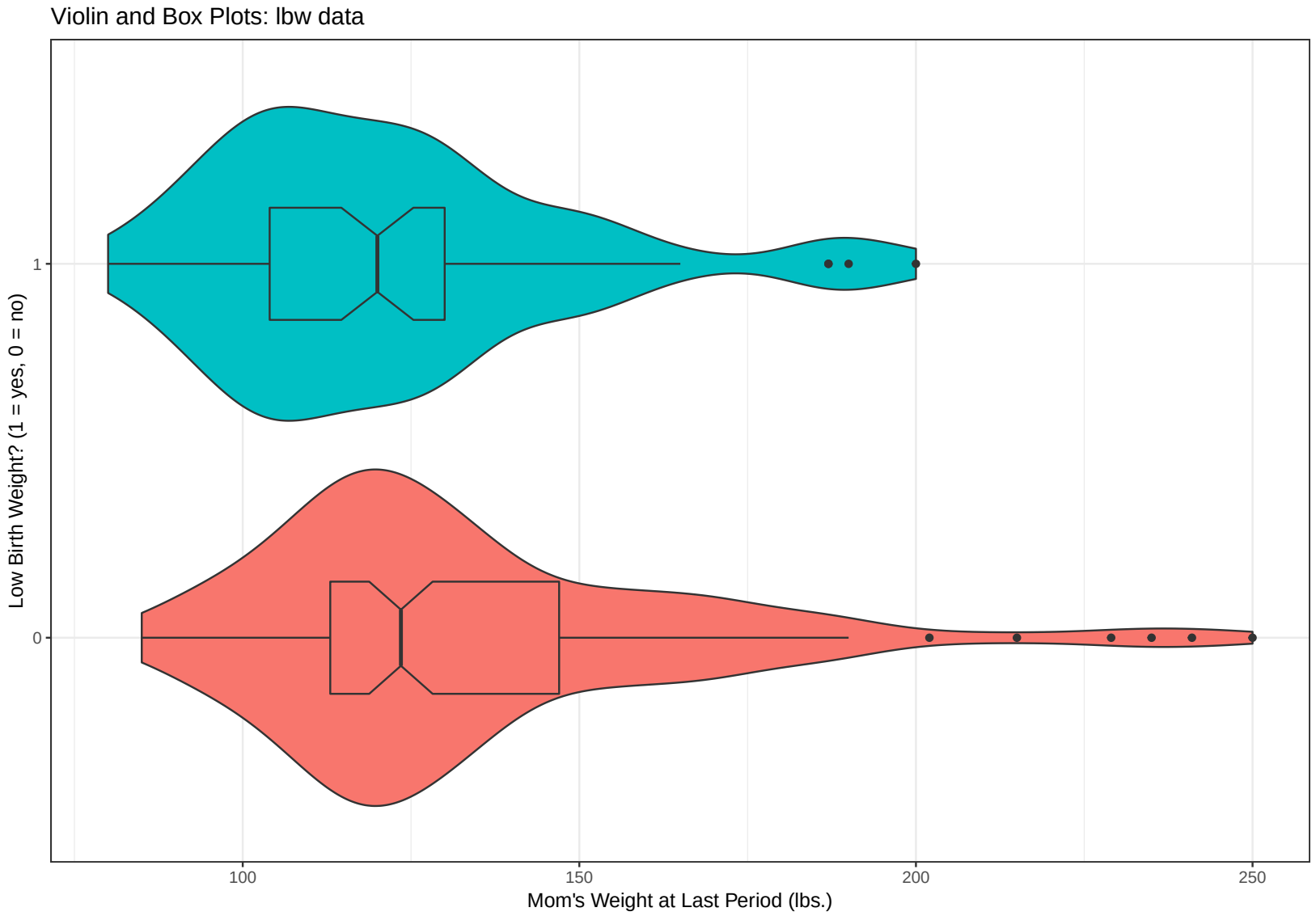
We want to look at the probability of a low birth weight (`low` = 1) on the basis of the mom's weight at her last menstrual period (`lwt`).

```
lbw1 %>% group_by(low) %>% skim(lwt)
```

```
> lbw1 %>% group_by(low) %>% skim(lwt)
Skim summary statistics
 n obs: 189
 n variables: 10
 group variables: low

Variable type: integer
 low variable missing complete   n   mean    sd p0 p25 median p75 p100     hist
   0      lwt       0      130 130 133.3  31.72 85 113  123.5 147  250 ▅▇▅▂▁▁▁▁
   1      lwt       0       59  59 122.14 26.56 80 104  120   130  200 ▅▇▅▂▂▁▁▂
```

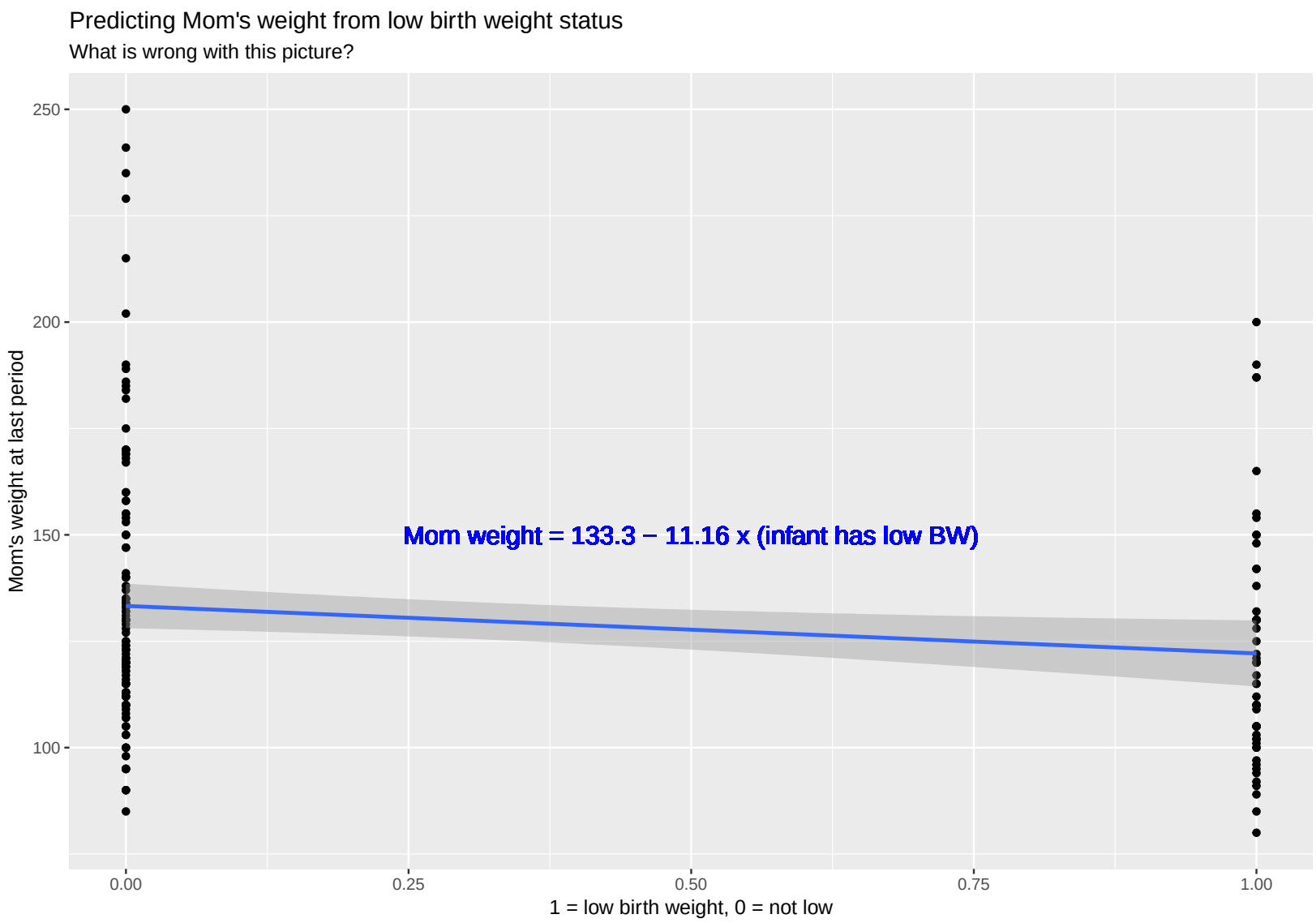
Can we predict Pr(low) effectively with `lwt`?



Code for Previous Slide

```
ggplot(lbw, aes(x = factor(low), y = lwt,
                fill = factor(low))) +
    geom_violin() +
    geom_boxplot(width = .3, notch = TRUE) +
    guides(fill = FALSE) +
    labs(x = "Low Birth Weight? (1 = yes, 0 = no)",
         y = "Mom's Weight at Last Period (lbs.)",
         title = "Violin and Box Plots: lbw data") +
    theme_bw() +
    coord_flip()
```

Working in Reverse: Can we predict `lwt` with `low`?



Working in Reverse: Predicting `lwt` with `low`

Easy to go in the other direction...

```
lm(lwt ~ low, data = lbw)
```

```
Call:
lm(formula = lwt ~ low, data = lbw)

Coefficients:
(Intercept)          low
     133.30       -11.16
```

Weight at Last Period = 133.3 - 11.16 * (baby is low bw)

- But that's reversing the outcome and predictor...

Can we fit a linear probability model? Sure, but . . .

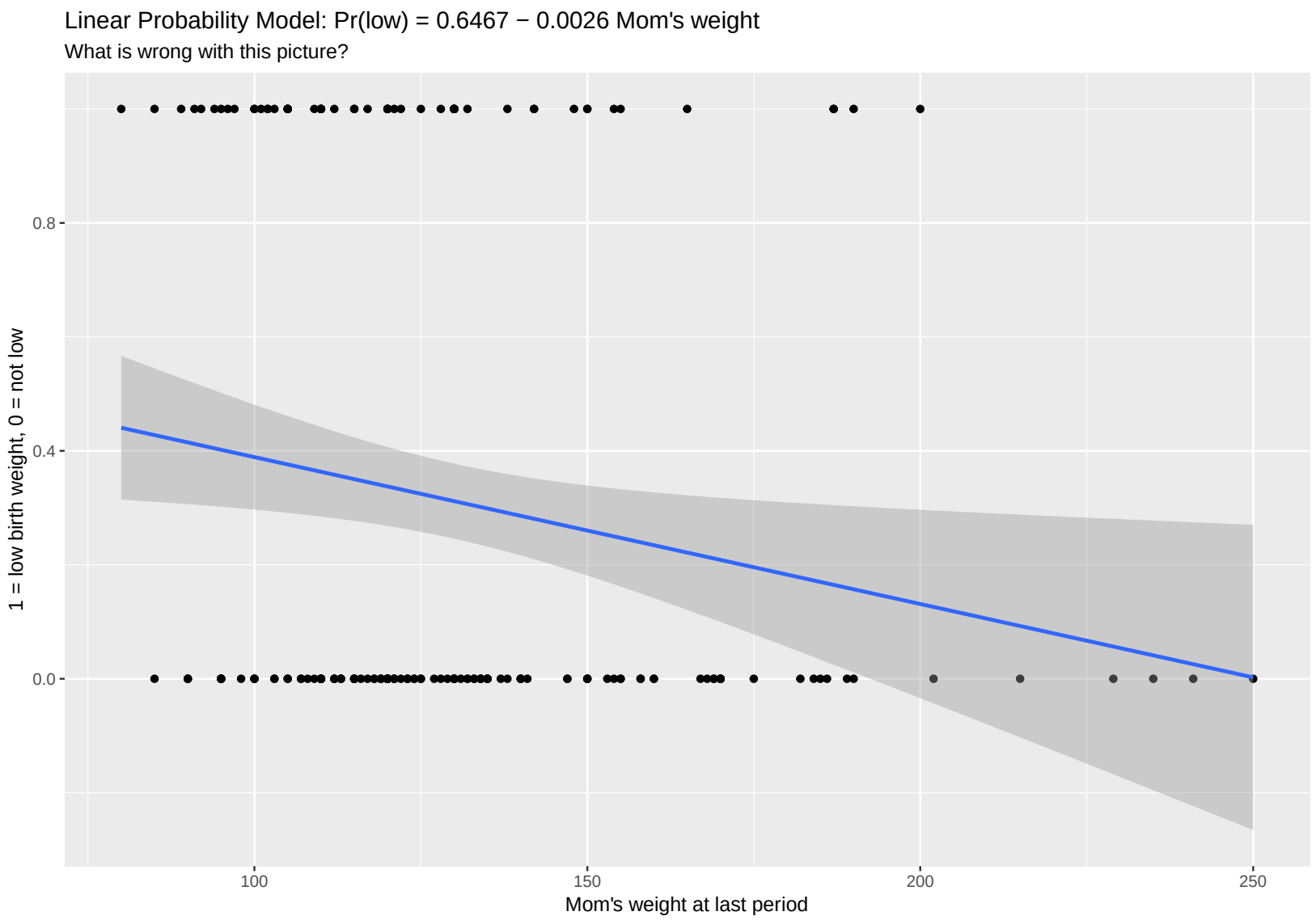
```r
lm(low ~ lwt, data = lbw)
```

```
Call:
lm(formula = low ~ lwt, data = lbw)

Coefficients:
(Intercept)          lwt
   0.646733    -0.002577
```

Pr(low birth weight) = 0.6467 - 0.0026 (Mom's weight at last period)

Plotting the Linear Probability Model



Fitting a Model to predict a Binary Outcome

Logistic regression is the most common model used when the outcome is binary. Our response variable is assumed to take on two values - zero or one, and we then describe the probability of a "one" response, given a linear function of explanatory predictors.

- Linear regression approaches to the problem of predicting probabilities are problematic for several reasons: not least of which being that they predict probabilities greater than one and less than zero.

Logistic regression is a non-linear regression approach, since the equation for the mean of the 0/1 Y values conditioned on the values of our predictors $X_1, X_2, ..., X_k$ turns out to be non-linear in the β coefficients.

The Logit Link and Logistic Function

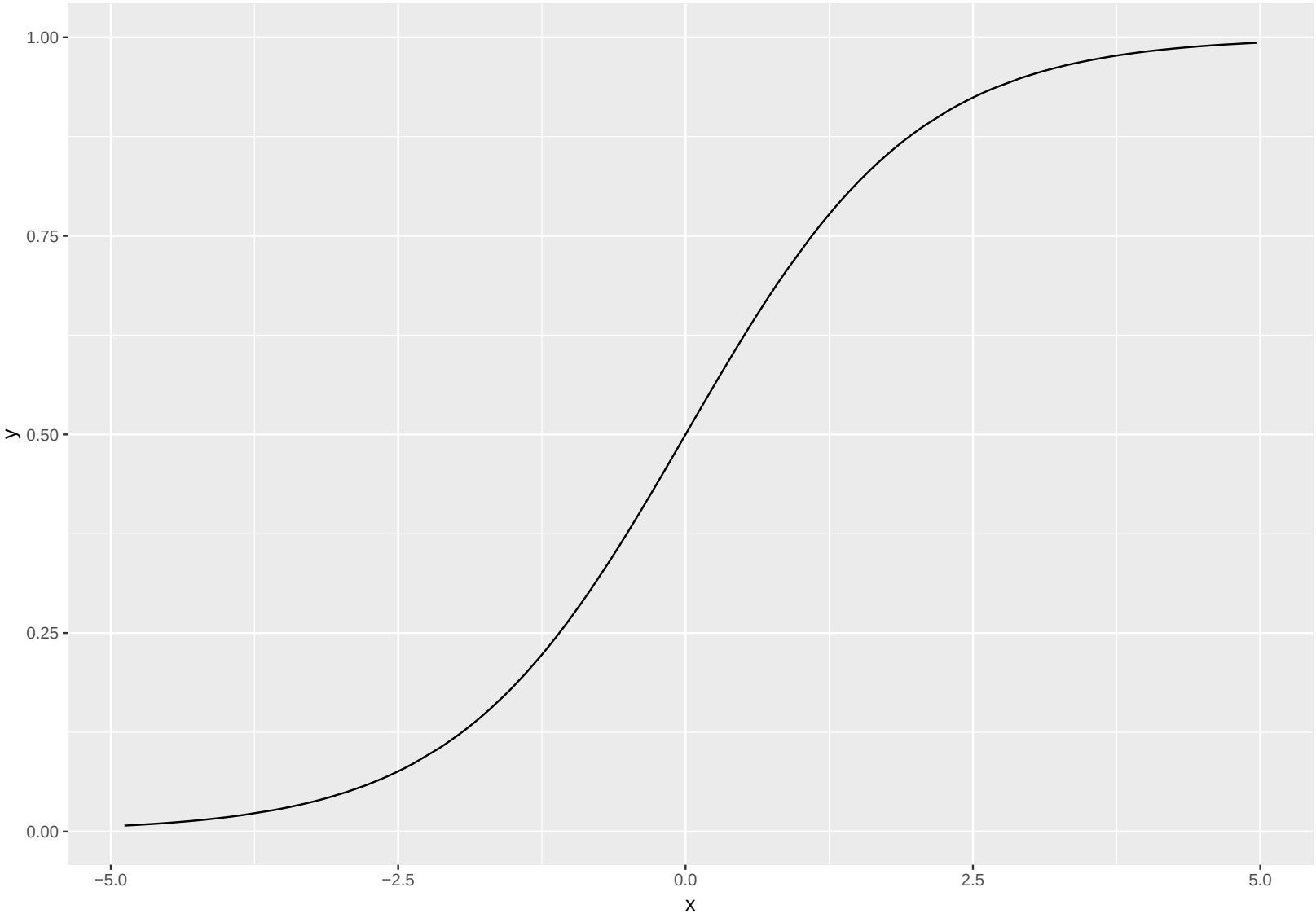
The particular link function we use in logistic regression is called the **logit link**.

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

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 $logit(\pi) = 0$.

Model 1

We'll use `glm` to get started.

```
model.1 <- glm(low ~ lwt, data = lbw, family = binomial)
model.1
```

```
Call:  glm(formula = low ~ lwt, family = binomial, data = lbw)

Coefficients:
(Intercept)          lwt
    0.99831     -0.01406

Degrees of Freedom: 188 Total (i.e. Null);  187 Residual
Null Deviance:      234.7
Residual Deviance: 228.7    AIC: 232.7
```

Our logistic regression model

The logistic regression equation is:

$$logit(Pr(low = 1)) = log\left(\frac{Pr(low = 1)}{1 - Pr(low = 1)}\right) = 0.99831 - 0.01406 \times lwt$$

Suppose, for instance, that we are interested in making a prediction when Mom's weight at her last period, `lwt` = 130 lbs.

So we have:

$$logit(Pr(low = 1)) = 0.99831 - 0.01406x130 = -0.82949$$

Getting a Prediction from R for the Model

```
model.1 <- glm(low ~ lwt, data = lbw, family = binomial)
```

To predict on the log odds scale, we use

```
predict(model.1, newdata = data.frame(lwt = 130))
```

```
         1
-0.8292596
```

The Model in terms of Odds

We can exponentiate to state the odds, rather than the log odds. For a Mom at 130 lbs, we have:

$$log\left(\frac{Pr(low=1)}{1-Pr(low=1)}\right) = 0.99831 - 0.01406 \times 130 = -0.82949$$

and so we have

$$Odds(low=1|lwt=130) = exp(-0.82949) = 0.4362717$$

Making a Prediction about Probability

$$Odds(low = 1|lwt = 130) = \frac{Pr(low = 1)}{1 - Pr(low = 1)} = 0.4362717$$

so

$$Pr(low = 1|lwt = 130) = \frac{Odds(low = 1|lwt = 130)}{1 + Odds(low = 1|lwt = 130)} = \frac{0.4362717}{1 + 0.4362717}$$

which is 0.304.

Obtaining a Prediction from R for Prob(low = 1)

```
model.1 <- glm(low ~ lwt, data = lbw, family = binomial)
```

To predict on the probability scale, we can use

```
predict(model.1, newdata = data.frame(lwt = 130),
        type = "response")
```

```
        1
0.3038016
```

Plotting the Logistic Regression Model

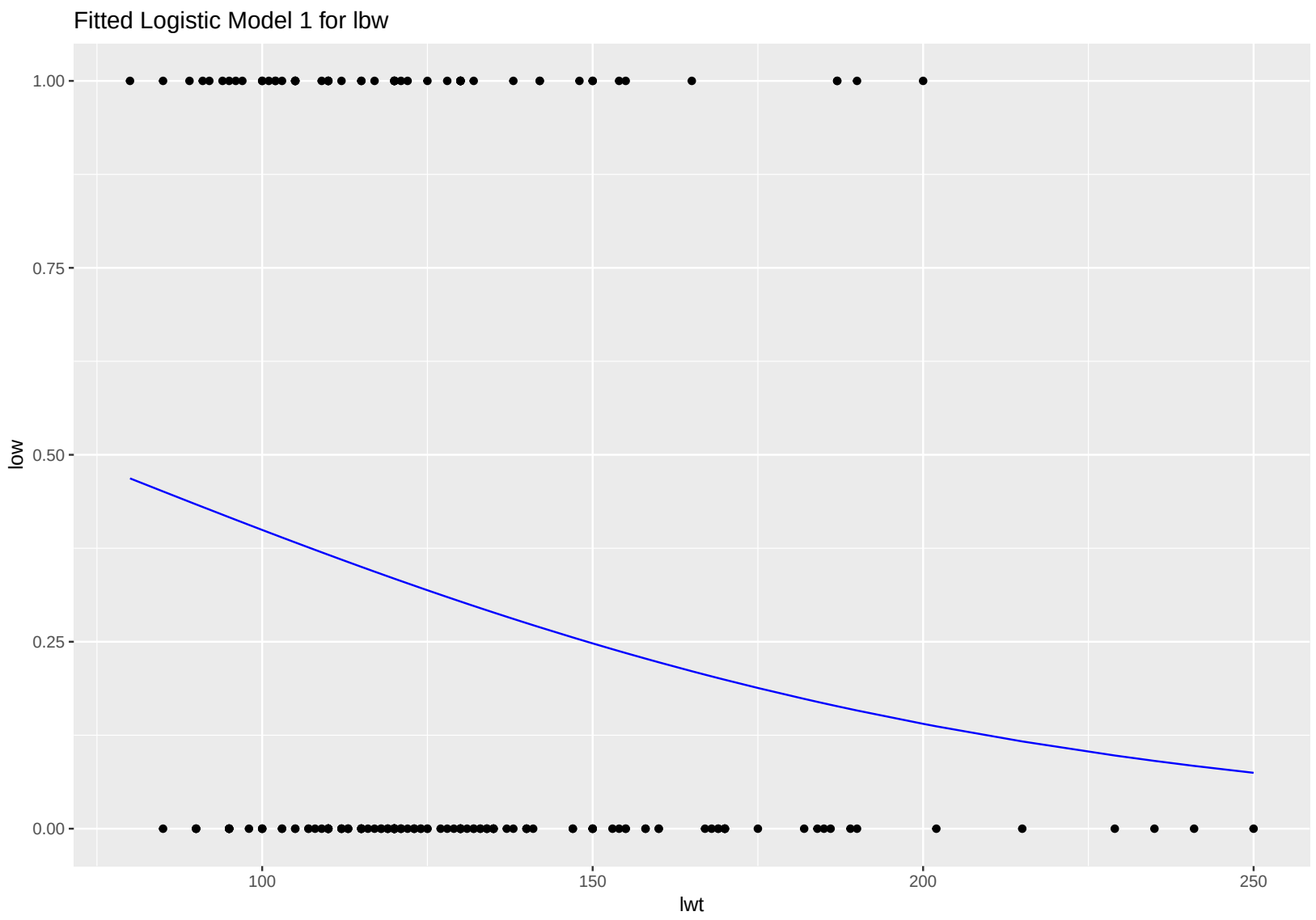
We can use the `augment` function from the `broom` package to get our fitted probabilities included in the data.

```
mod1.aug <- augment(model.1, lbw,
                    type.predict = "response")

ggplot(mod1.aug, aes(x = lwt, y = low)) +
    geom_point() +
    geom_line(aes(x = lwt, y = .fitted), col = "blue") +
    labs(title = "Fitted Logistic Model 1 for lbw")
```

- Results on next slide

Plotting the Logistic Regression Model



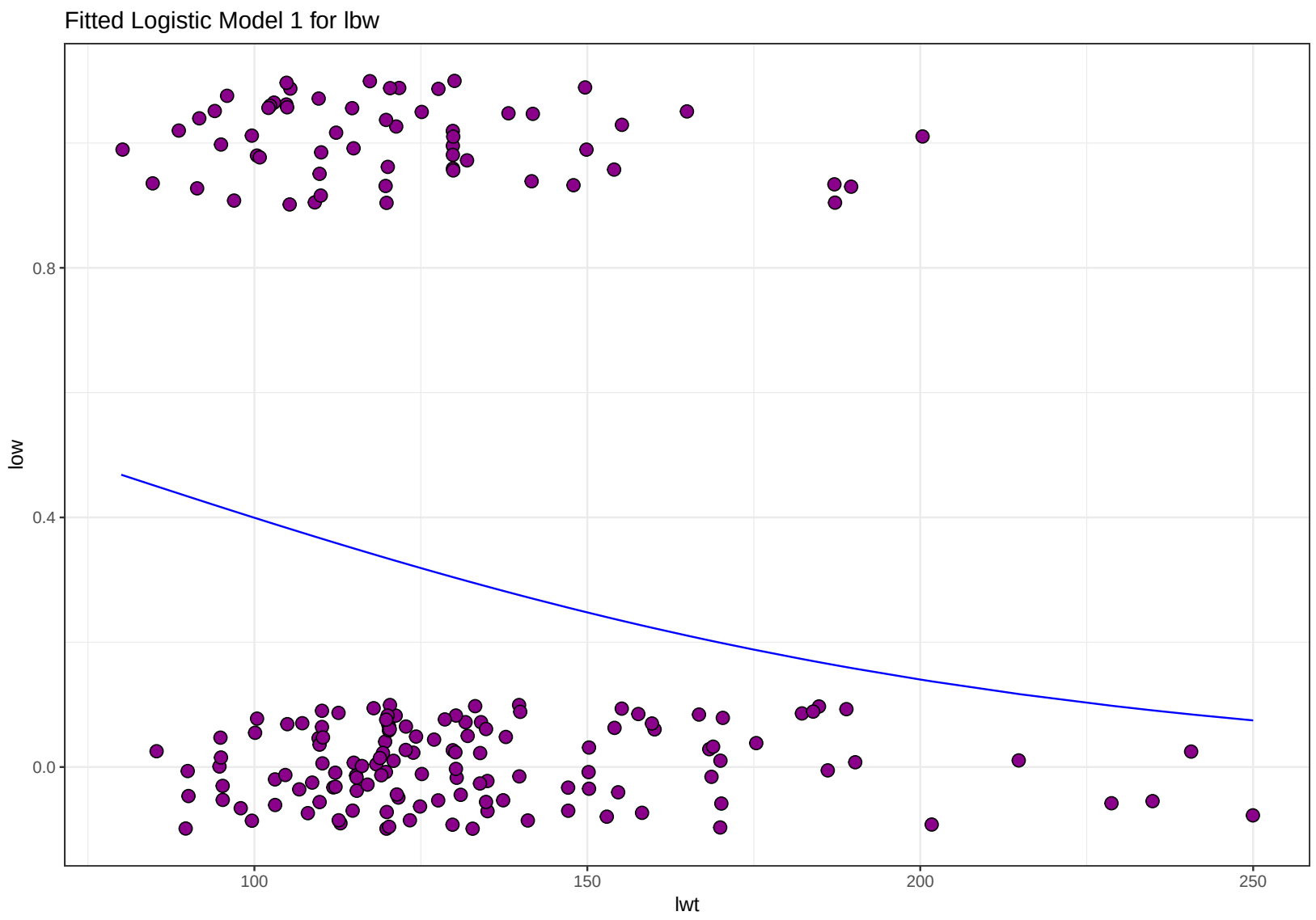
Cleaning up the plot

I'll add a little jitter on the vertical scale to the points, so we can avoid overlap, and also make the points a little bigger.

```
ggplot(mod1.aug, aes(x = lwt, y = low)) +
    geom_jitter(height = 0.1, size = 3, pch = 21,
                fill = "darkmagenta") +
    geom_line(aes(x = lwt, y = .fitted), col = "blue") +
    labs(title = "Fitted Logistic Model 1 for lbw") +
    theme_bw()
```

- Results on next slide

Cleaned up Plot of Model 1



Studying the Model, Again

```
model.1
```

```
Call:  glm(formula = low ~ lwt, family = binomial, data = lbw)

Coefficients:
(Intercept)          lwt
    0.99831     -0.01406

Degrees of Freedom: 188 Total (i.e. Null);  187 Residual
Null Deviance:      234.7
Residual Deviance: 228.7    AIC: 232.7
```

- logit(Pr(low = 1)) = 0.998 - 0.014 lwt
  - so ... as lwt increases, what happens to Pr(low = 1)?
  - if Harry's mom weighed 130 lbs and Sally's weighed 150 lbs, how can we compare the predicted Pr(low = 1) for Harry and Sally?

Comparing Harry (lwt = 130) to Sally (lwt = 150)

```
predict(model.1, newdata = data.frame(lwt = c(130, 150)),
        type = "response")
```

```
        1         2
0.3038016 0.2477917
```

- Harry's mom weighed 130 lbs, and his predicted probability of low birth weight is 0.304
- Sally's mom weighed 150 lbs, and her predicted Pr(low = 1) = 0.248

Interpreting the Coefficients of the Model

```r
coef(model.1)
```

```
(Intercept)         lwt
 0.99831432 -0.01405826
```

To understand the effect of `lwt` on `low`, try odds ratios.

```r
exp(coef(model.1))
```

```
(Intercept)         lwt
  2.7137035   0.9860401
```

Suppose Charlie's Mom weighed one pound more than Harry's.

- The **odds** of low birth weight are 0.986 times as large for Charlie as Harry.
- In general, odds ratio comparing two subjects whose `lwt` differ by 1 pound is 0.986

Comparing Harry to Charlie

Charlie's mom weighed 1 pound more than Harry's. The estimated odds ratio for low birth weight from the model associated with a one pound increase in `lwt` is 0.986.

- If the odds ratio was 1, that would mean that Charlie and Harry had the same estimated odds of low birth weight, and thus the same estimated probability of low birth weight, despite having Moms with different weights.
- Since the odds ratio is less than 1, it means that **Charlie** has a **lower** estimated odds of low birth weight than Harry, and thus that Charlie has a lower estimated probability of low birth weight than Harry.
- If the odds ratio was greater than 1, it would mean that Charlie had a higher estimated odds of low birth weight than Harry, and thus that Charlie had a higher estimated probability of low birth weight than Harry.

The smallest possible odds ratio is . . . ?

The rest of the model's output

```
Degrees of Freedom: 188 Total (i.e. Null);  187 Residual
Null Deviance:      234.7
Residual Deviance: 228.7    AIC: 232.7
```

	Model	Null	Residual	Δ (`model.1`)
Deviance (lack of fit)	234.7	228.7		6.0
Degrees of Freedom	188	187		1

- Deviance accounted for by `model.1` is 6 points on 1 df
- Can compare to a χ^2 distribution for a p value via `anova`

AIC = 232.7, still useful for comparing models for the same outcome

anova on a glm model

```
anova(model.1)
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: low

Terms added sequentially (first to last)

     Df Deviance Resid. Df Resid. Dev
NULL                    188     234.67
lwt   1   5.9813       187     228.69
```

```
pchisq(5.9813, 1, lower.tail = FALSE)
```

```
[1] 0.01445834
```

Next Time

- How well does this model classify subjects?
- Receiver Operating Characteristic Curve Analysis
  - The C statistic (Area under the curve)
- Assessing Residual Plots for a Logistic Regression
- A "Kitchen Sink" Logistic Regression Model
  - Comparing Models
  - Interpreting Models with Multiple Predictors