432 Class 7 Slides

github.com/THOMASELOVE/2019-432

2019-02-14

Setup

```
library(skimr); library(broom); library(janitor)
library(pROC); library(ROCR)  # these are new to us
library(tidyverse)

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

The Low Birth Weight data, again

```
lbw1 <- lbw %>%
    mutate(race_f = fct_recode(factor(race), white = "1",
                               black = "2", other = "3"),
           race_f = fct_relevel(race_f, "white", "black")) %>%
    mutate(preterm = fct_recode(factor(ptl > 0),
                                yes = "TRUE",
                                no = "FALSE")) %>%
    rename(momwt = lwt) %>%
    select(subject, low, momwt, age, ftv, ht, race_f,
           preterm, smoke, ui)
```

The `lbw1` data (n = 189 infants)

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

Source: Hosmer, Lemeshow and Sturdivant, *Applied Logistic Regression* 3rd edition. Data from Baystate Medical Center, Springfield MA in 1986.

Our current model

```
model.1 <- glm(low ~ momwt, data = lbw1, family = binomial)
model.1
```

```
Call:  glm(formula = low ~ momwt, family = binomial, data = lb

Coefficients:
(Intercept)        momwt
    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:

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

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

So we have:

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

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

```
model.1 <- glm(low ~ momwt, data = lbw1, family = binomial)
```

To predict on the probability scale, we can use

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

```
        1
0.3038016
```

Plotting the Logistic Regression Model

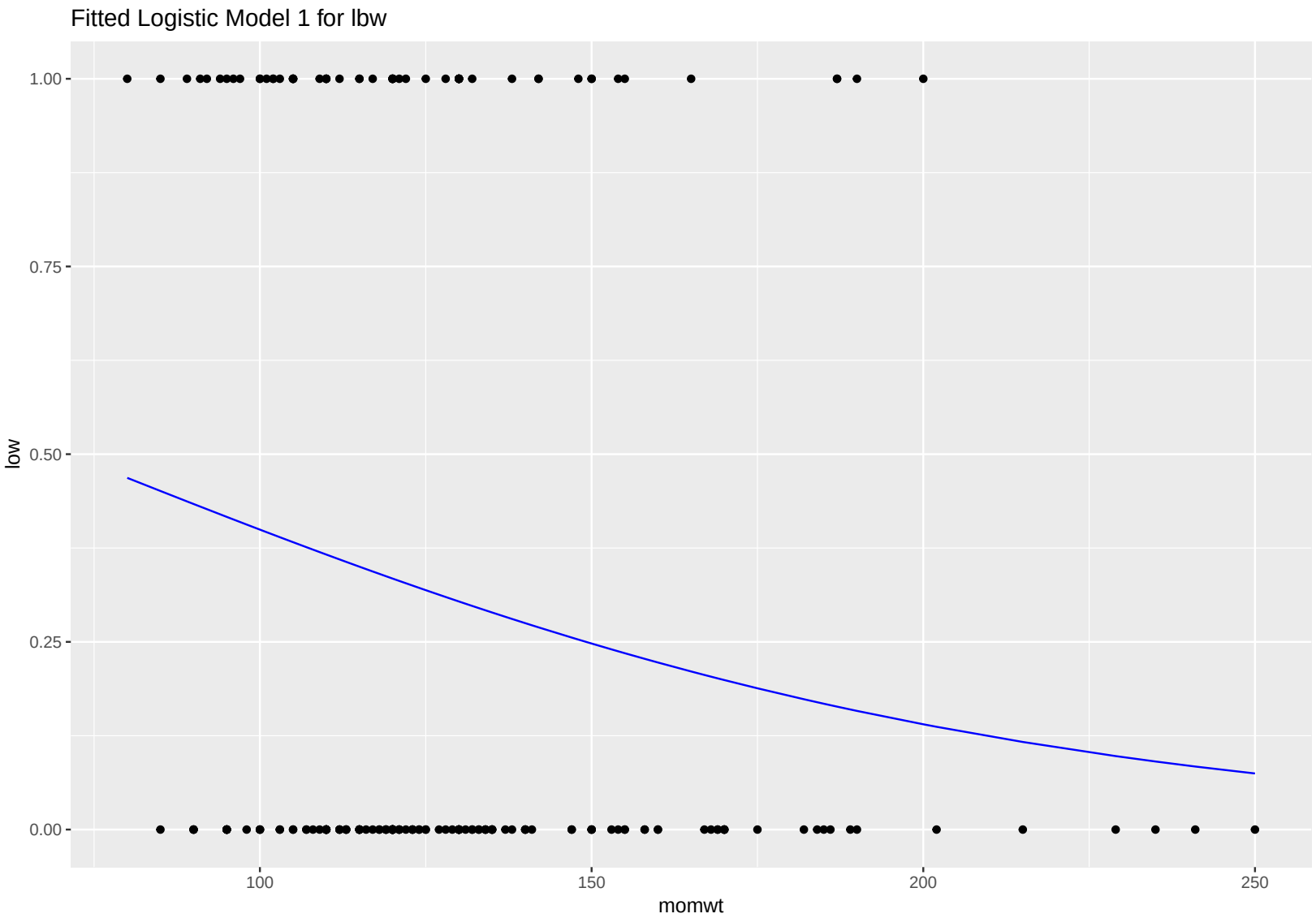
We can use the `augment` function from the `broom` package to get our fitted probabilities included in the data.

```
mod1.aug <- augment(model.1, lbw1,
                    type.predict = "response")

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

- Results on next slide

Plotting the Logistic Regression Model



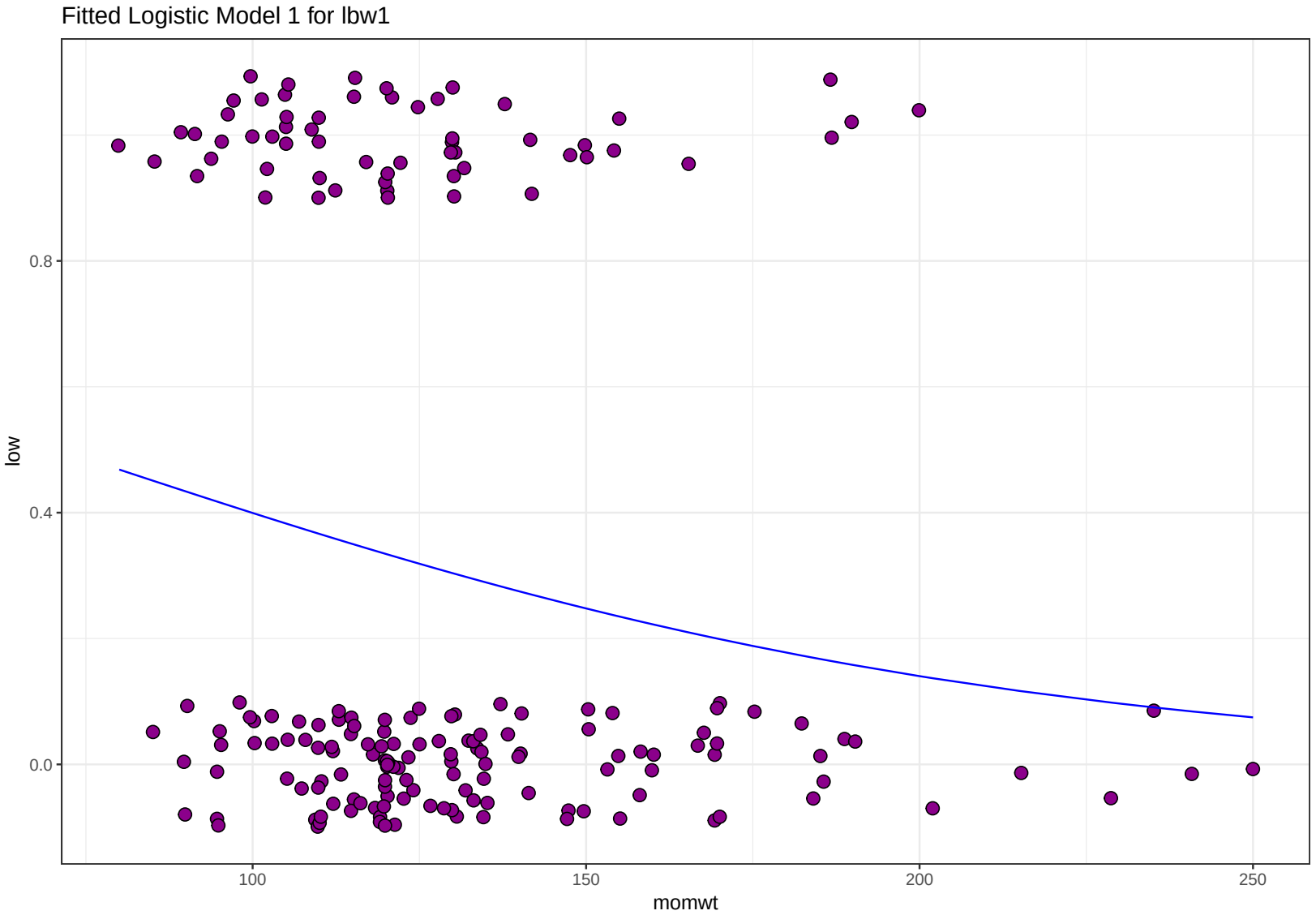
Cleaning up the plot

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 = momwt, y = low)) +
    geom_jitter(height = 0.1, size = 3, pch = 21,
                fill = "darkmagenta") +
    geom_line(aes(x = momwt, y = .fitted), col = "blue") +
    labs(title = "Fitted Logistic Model 1 for lbw1") +
    theme_bw()
```

- Results on next slide

Cleaned up Plot of Model 1

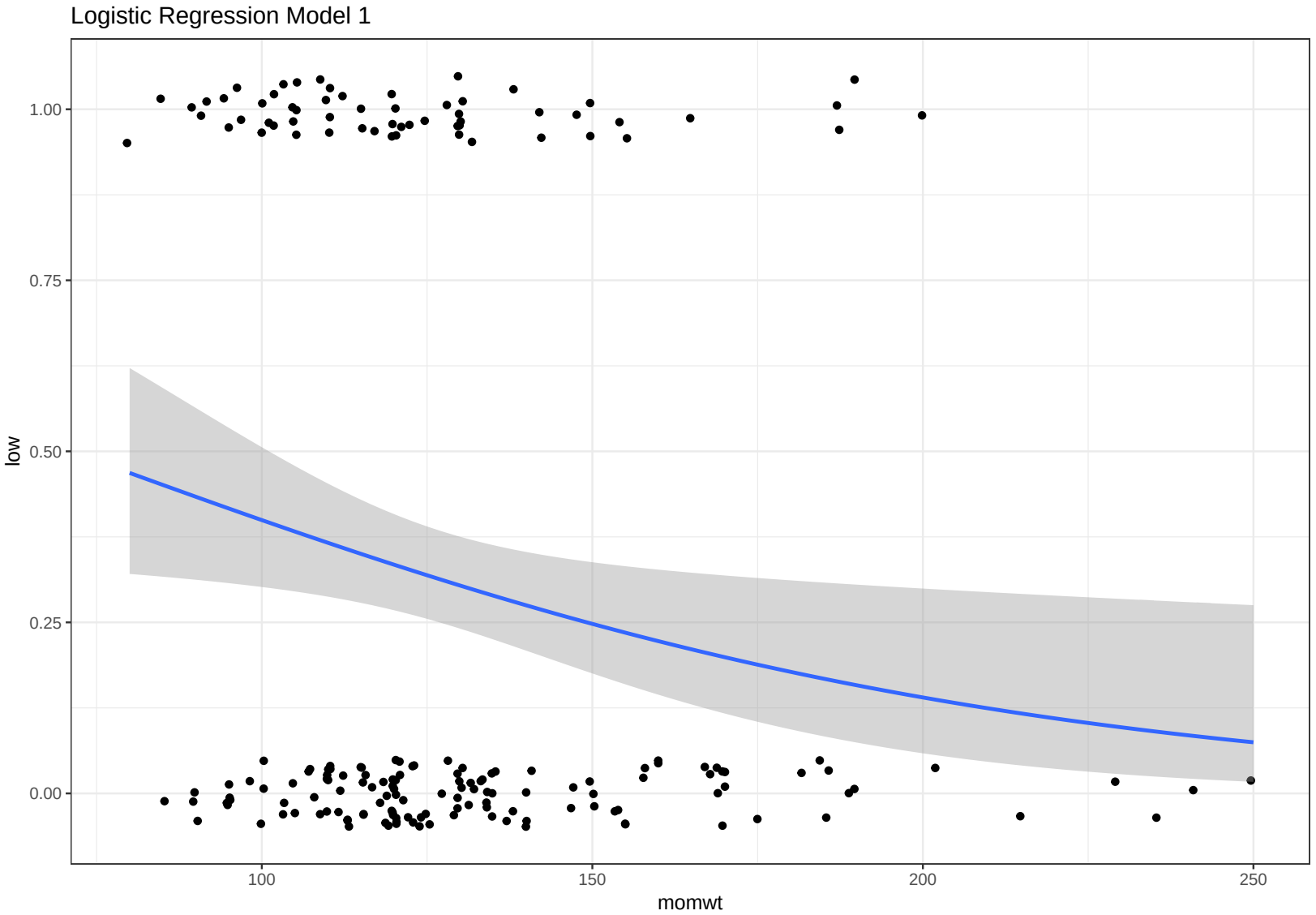


Plotting a Simple Logistic Model using `binomial_smooth`

```
binomial_smooth <- function(...) {
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"), ...)
}

ggplot(lbw1, aes(x = momwt, y = low)) +
  geom_jitter(height = 0.05) +
  binomial_smooth() +
    ## ...smooth(se=FALSE) to leave out interval
  labs(title = "Logistic Regression Model 1") +
  theme_bw()
```

The Resulting Plot



Studying the Model, Again

```
model.1
```

```
Call:  glm(formula = low ~ momwt, family = binomial, data = lb

Coefficients:
(Intercept)        momwt
    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 momwt
  - so . . . as momwt 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?

Harry (momwt = 130) vs. Sally (momwt = 150)

```
predict(model.1, newdata = data.frame(momwt = 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

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

```
(Intercept)       momwt
 0.99831432 -0.01405826
```

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

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

```
(Intercept)       momwt
  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 `momwt` 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 `momwt` 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
momwt  1   5.9813       187     228.69
```

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

```
[1] 0.01445834
```

glance on model.1

```
glance(model.1)
```

```
# A tibble: 1 x 7
  null.deviance df.null logLik   AIC   BIC deviance
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>
1          235.     188  -114.  233.  239.     229.
# ... with 1 more variable: df.residual <int>
```

- Deviance $= -2 \times$ log (likelihood)
- AIC and BIC are based on the deviance, but with differing penalties for complicating the model
- AIC and BIC remain useful for comparing multiple models for the same outcome

summary of model.1

```
> summary(model.1)

Call:
glm(formula = low ~ momwt, family = binomial, data = lbw1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0951  -0.9022  -0.8018   1.3609   1.9821

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.99831    0.78529   1.271   0.2036
momwt       -0.01406    0.00617  -2.279   0.0227 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 228.69  on 187  degrees of freedom
AIC: 232.69

Number of Fisher Scoring iterations: 4
```

Coefficients output

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.99831    0.78529   1.271   0.2036
momwt        -0.01406    0.00617  -2.279   0.0227 *
```

- We have a table of coefficients with standard errors, and hypothesis tests, although these are Wald z-tests, rather than the t tests we saw in linear modeling.
- `momwt` has a Wald Z of -2.279, yielding $p = 0.0227$
  - H_0: `momwt` does not have an effect on the log odds of `low`
  - H_A: `momwt` does have such an effect
- If the coefficient (on the logit scale) for `momwt` was truly 0, this would mean that:
  - the log odds of low birth weight did not change based on `momwt`,
  - the odds of low birth weight were unchanged based on `momwt` (OR = 1), and
  - the probability of low birth weight was unchanged based on the `momwt`.

Confidence Intervals for Coefficients

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

```
(Intercept)       momwt
 0.99831432 -0.01405826
```

```
confint(model.1, level = 0.95)
```

```
Waiting for profiling to be done...

                  2.5 %       97.5 %
(Intercept) -0.48116701  2.611748138
momwt       -0.02696198 -0.002650036
```

- The coefficient of momwt has a point estimate of -0.014 and a 95% confidence interval of (-0.027, -0.003).
- On the logit scale, this isn't that interpretable, but we will often exponentiate to describe odds ratios.

Odds Ratio Interpretation of exp(Coefficient)

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

```
(Intercept)           momwt
  2.7137035   0.9860401
```

```
exp(confint(model.1, level = 0.95))
```

```
                2.5 %     97.5 %
(Intercept) 0.6180617 13.6228447
momwt       0.9733982  0.9973535
```

- Odds Ratio for `low` based on a one pound increase in `momwt` is 0.986 (95% CI: 0.973, 0,997).
  - Estimated odds of low birth weight will be smaller (odds < 1) for those with larger `momwt` values.
  - Smaller odds(low birth weight) = smaller Prob(low birth weight).

Deviance Residuals

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0951  -0.9022  -0.8018   1.3609   1.9821
```

- The deviance residuals for each individual subject sum up to the deviance statistic for the model, and describe the contribution of each point to the model likelihood function. The formula is in the Course Notes.
- Logistic Regression is a non-linear model, and it doesn't come with either an assumption that the residuals will follow a Normal distribution, or an assumption that the residuals will have constant variance, so when we build diagnostics for the logistic regression model, we'll use different plots and strategies than we used in linear models.

Other New Things

```
(Dispersion parameter for binomial family taken to be 1)

Number of Fisher Scoring iterations: 4
```

- Dispersion parameters matter for some generalized linear models. For binomial family models like the logistic, it's always 1.
- The solution of a logistic regression model involves maximizing a likelihood function. Fisher's scoring algorithm needed just four iterations to perform this fit. The model converged, quickly.

How Well Does Our `model.1` Classify Subjects?

One possible rule: if predicted Pr(low = 1) $\geq$ 0.5, then we predict "low birth weight"

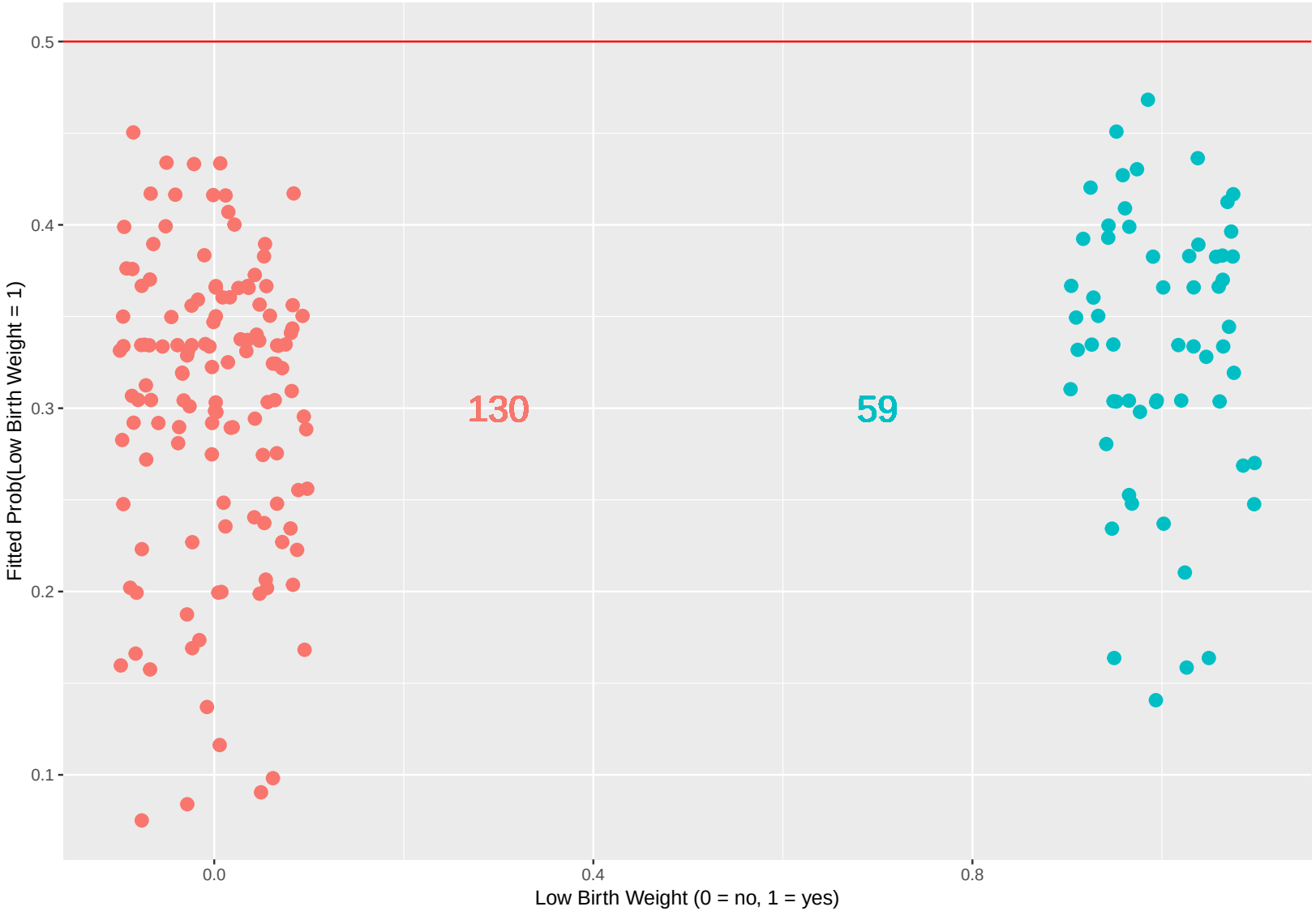
```
mod1.aug$rule.5 <- ifelse(mod1.aug$.fitted >= 0.5,
                          "Predict Low", "Predict Not Low")

table(mod1.aug$rule.5, mod1.aug$low)
```

```
                    0   1
  Predict Not Low 130  59
```

This rule might be a problem for us. What % are correct?

A plot of classifications with the 0.5 rule



How Well Does Our `model.1` Classify Subjects?

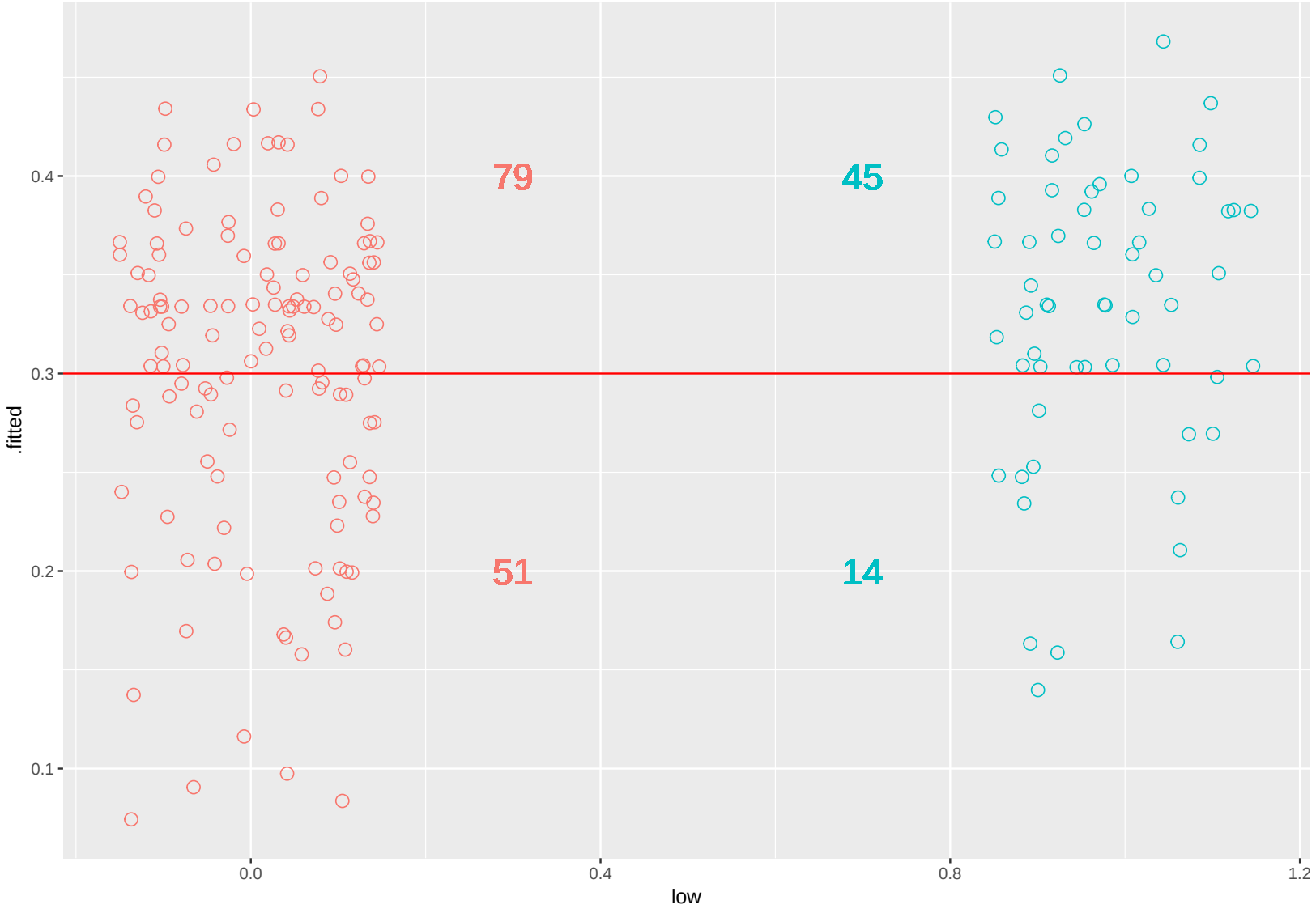
A new rule: if predicted Pr(low = 1) $\geq$ 0.3, then we predict "high risk of low birth weight" and otherwise, we predict "low risk of low birth weight"

```
mod1.aug$rule.3 <- ifelse(mod1.aug$.fitted >= 0.3,
                          "High Risk of LBW", "Low Risk of LBW")
mod1.aug <- mod1.aug %>%
    mutate(outcome_f = fct_recode(factor(low),
                                  "Low Birth Weight" = "1",
                                  "OK Birth Weight" = "0"),
           outcome_f = fct_relevel(outcome_f, "Low Birth Weigh

table(mod1.aug$rule.3, mod1.aug$outcome_f)
```

	Low Birth Weight	OK Birth Weight
High Risk of LBW	45	79
Low Risk of LBW	14	51

A plot of classifications with the 0.3 rule



The C Statistic (Area under the ROC Curve)

Our Model as Diagnostic Test

We want to assess predictive accuracy of our model.

- One approach: Receiver Operating Characteristic (ROC) curve analysis.
- A common choice for assessing diagnostic tests in medicine.

Consider two types of errors made by our model, in combination with a classification rule.

- Our model uses Mom's weight at last period to predict Pr(low birth weight).
- Lighter moms had higher model probabilities, so our rule would be: Predict low birth weight if Mom's last weight is no more than R pounds.

But the choice of R is available to us. Any value we select can lead to good outcomes (of our prediction) or to errors.

Test Results

- One good outcome of our "model/test" would be if the Mom's weight is less than R and her baby is born at a low birth weight.
- The other good outcome is if Mom's weight is greater than R and her baby is born at a non-low weight.

But we can make errors, too.

- A false positive occurs when we predict Pr(low = 1) to be small, but the baby is born at a low birth weight.
- A false negative occurs when we predict Pr(low = 1) to be large, but the baby is born at a non-low weight.

We identify two key summaries:

- The true positive fraction (TPF) for a specific weight cutoff R is Pr(Mom weight $<$ R | baby actually has low = 1).
- The false positive fraction (FPF) for a specific weight cutoff R is Pr(Mom weight $<$ R | baby has low = 0).

The ROC Curve

Since the cutoff R is not fixed in advanced, we can plot the value of TPF (on the y axis) against FPF (on the x axis) for all possible values of R, and this is what the ROC curve is.

- We calculate AUC = the area under the ROC curve (a value between 0 and 1) and use it to help summarize the effectiveness of the predictions made by the model on the following scale:
  - AUC above 0.9 = excellent discrimination of low = 1 from low = 0
  - AUC between 0.8 and 0.9 = good discrimination
  - AUC between 0.6 and 0.8 = mediocre/fair discrimination
  - AUC of 0.5 = random guessing
  - AUC below 0.5 = worse than guessing

Others refer to the Sensitivity on the Y axis, and 1-Specificity on the X axis, and this is the same idea. The TPF is called the sensitivity. 1 - FPF is the true negative rate, called the specificity.

A Simulation

```
set.seed(43223)
sim.temp <- data_frame(x = rnorm(n = 200),
                       prob = exp(x)/(1 + exp(x)),
                       y = as.numeric(1 * runif(200) < prob))
```

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

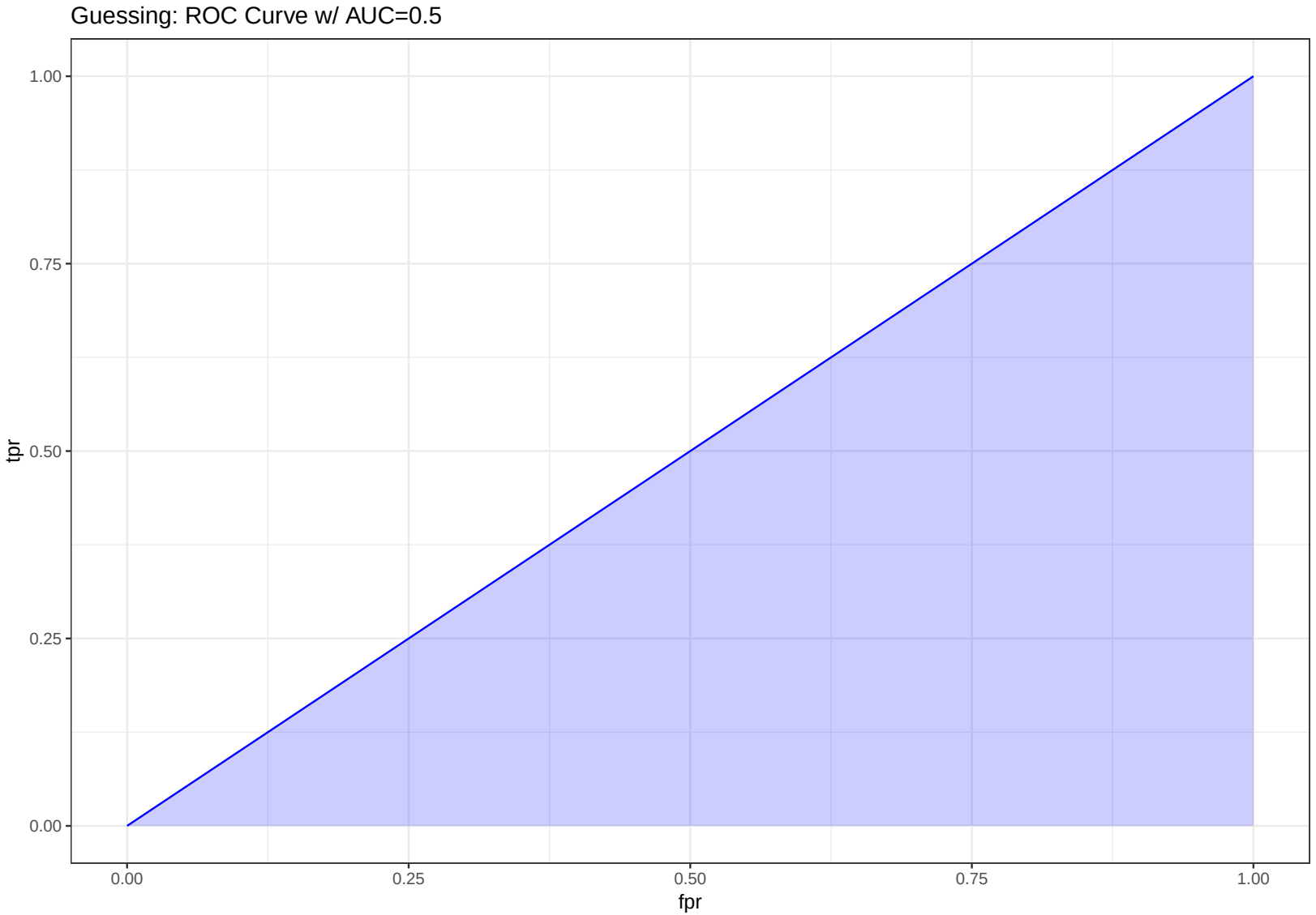
```
sim.temp <- sim.temp %>%
    mutate(p_guess = 1,
           p_perfect = y,
           p_bad = exp(-2*x) / (1 + exp(-2*x)),
           p_ok = prob + (1-y)*runif(1, 0, 0.05),
           p_good = prob + y*runif(1, 0, 0.27))
```

What if we are guessing?

If we're guessing completely at random, then the model should correctly classify a subject (as died or not died) about 50% of the time, so the TPR and FPR will be equal. This yields a diagonal line in the ROC curve, and an area under the curve (C statistic) of 0.5.

Plot is on the next slide. . .

What if we are guessing?



Building that ROC curve, Code part 1

This approach requires the loading of the ROCR package. . .

```
pred_guess <- prediction(sim.temp$p_guess, sim.temp$y)
perf_guess <- performance(pred_guess, measure = "tpr",
                          x.measure = "fpr")
auc_guess <- performance(pred_guess, measure="auc")

auc_guess <- round(auc_guess@y.values[[1]],3)
roc_guess <- data.frame(fpr=unlist(perf_guess@x.values),
                        tpr=unlist(perf_guess@y.values),
                        model="GLM")
```

Building that ROC curve, Code part 2

```
ggplot(roc_guess, aes(x=fpr, ymin=0, ymax=tpr)) +
    geom_ribbon(alpha=0.2, fill = "blue") +
    geom_line(aes(y=tpr), col = "blue") +
    labs(title = paste0("Guessing: ROC Curve w/ AUC=",
                        auc_guess)) +
    theme_bw()
```

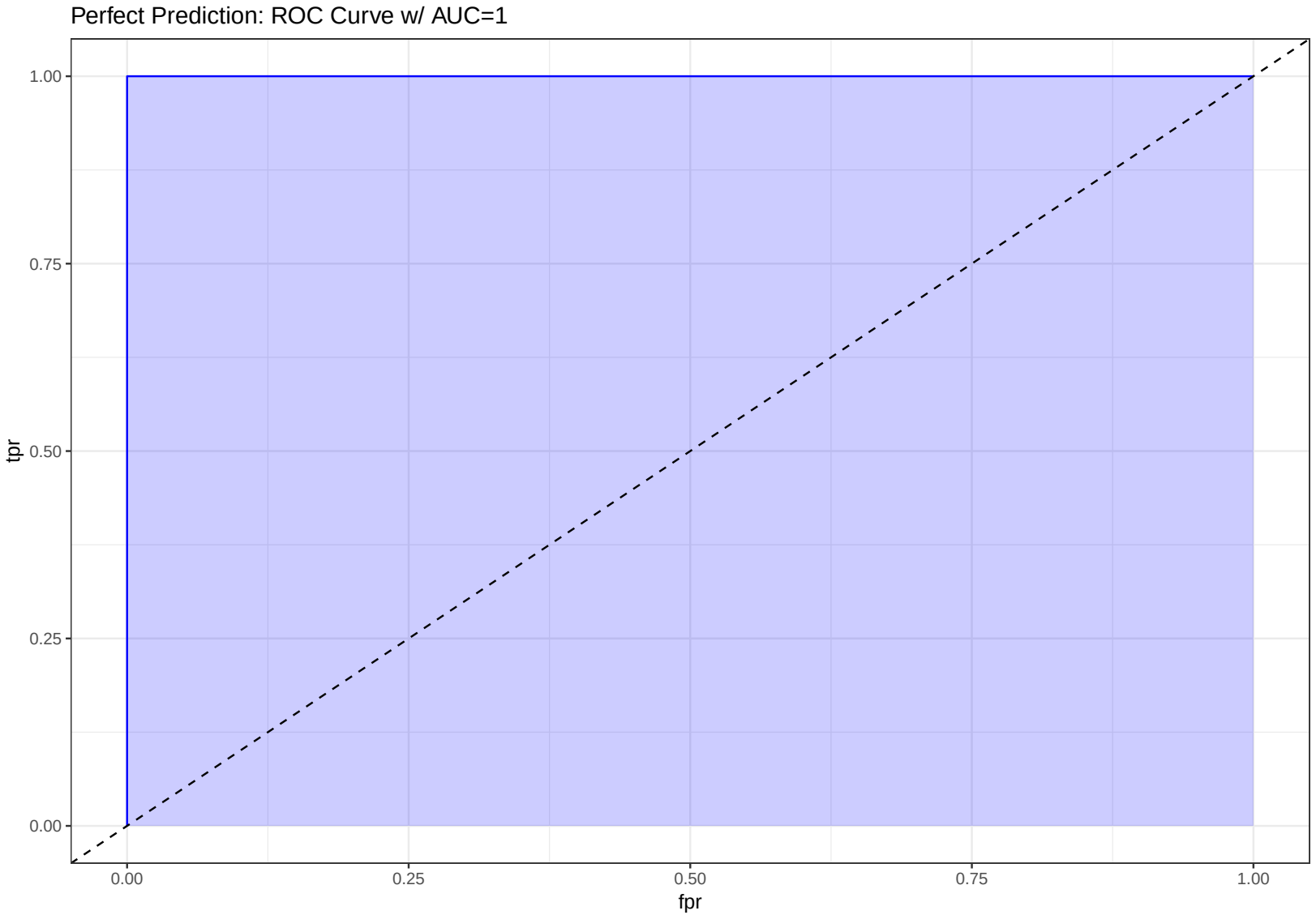
What if our model classifies things perfectly?

If we're classifying subjects perfectly, then we have a TPR of 1 and an FPR of 0.

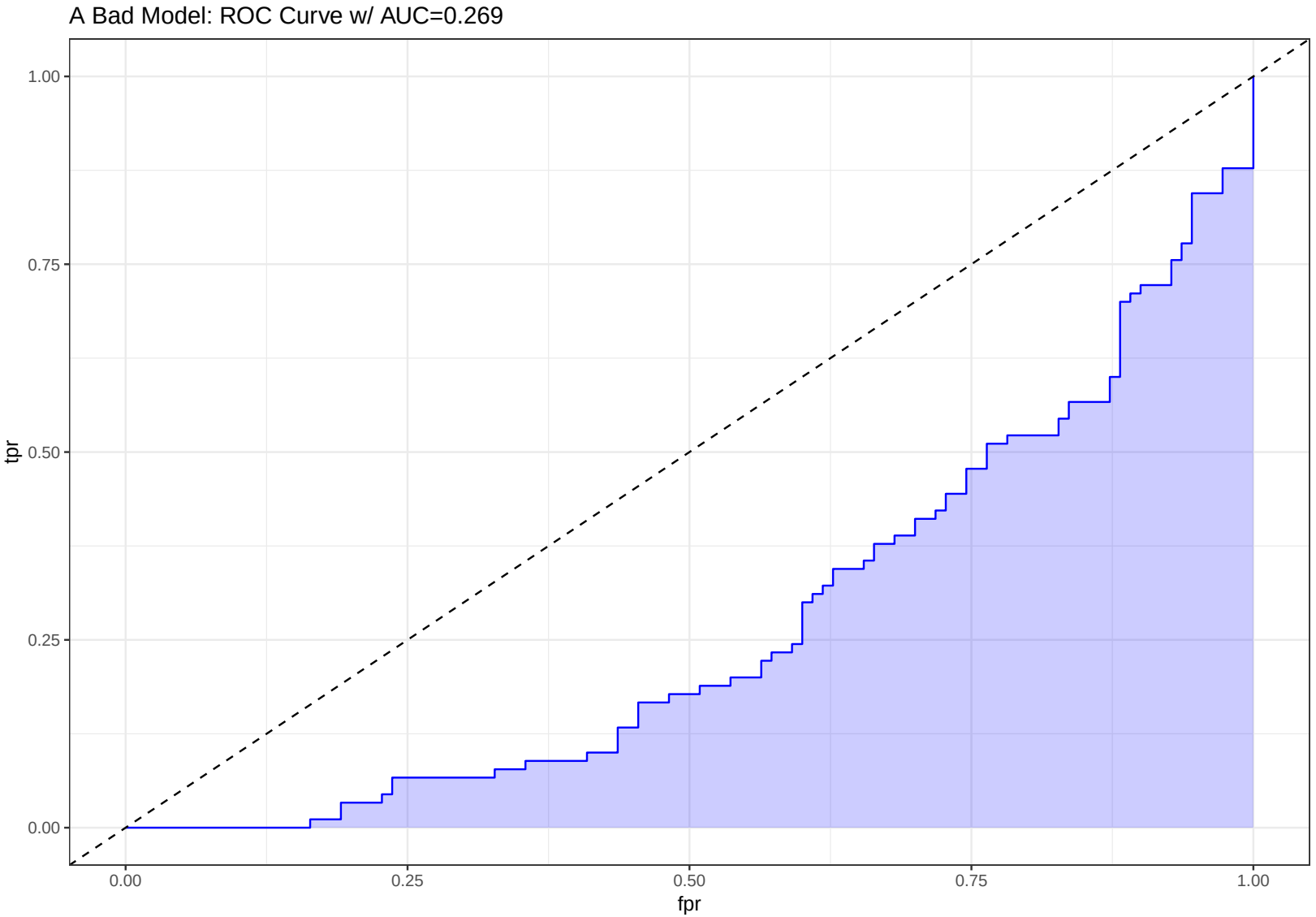
- That yields an ROC curve that looks like the upper and left edges of a box.
- If our model correctly classifies a subject (as died or not died) 100% of the time, the area under the curve (c statistic) will be 1.0.

I added in a diagonal dashed black line to show how this model compares to random guessing.

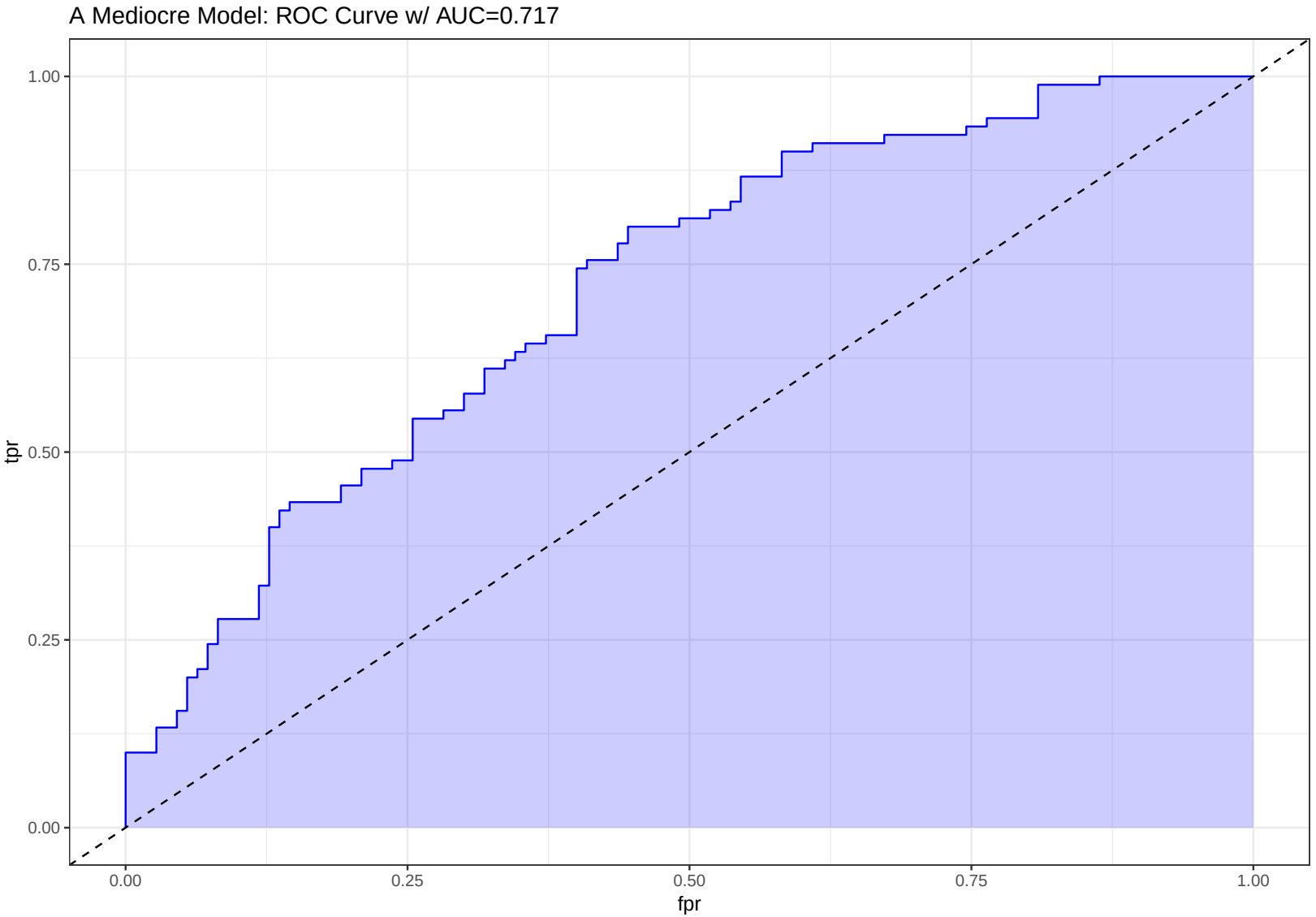
What if our model classifies things perfectly?



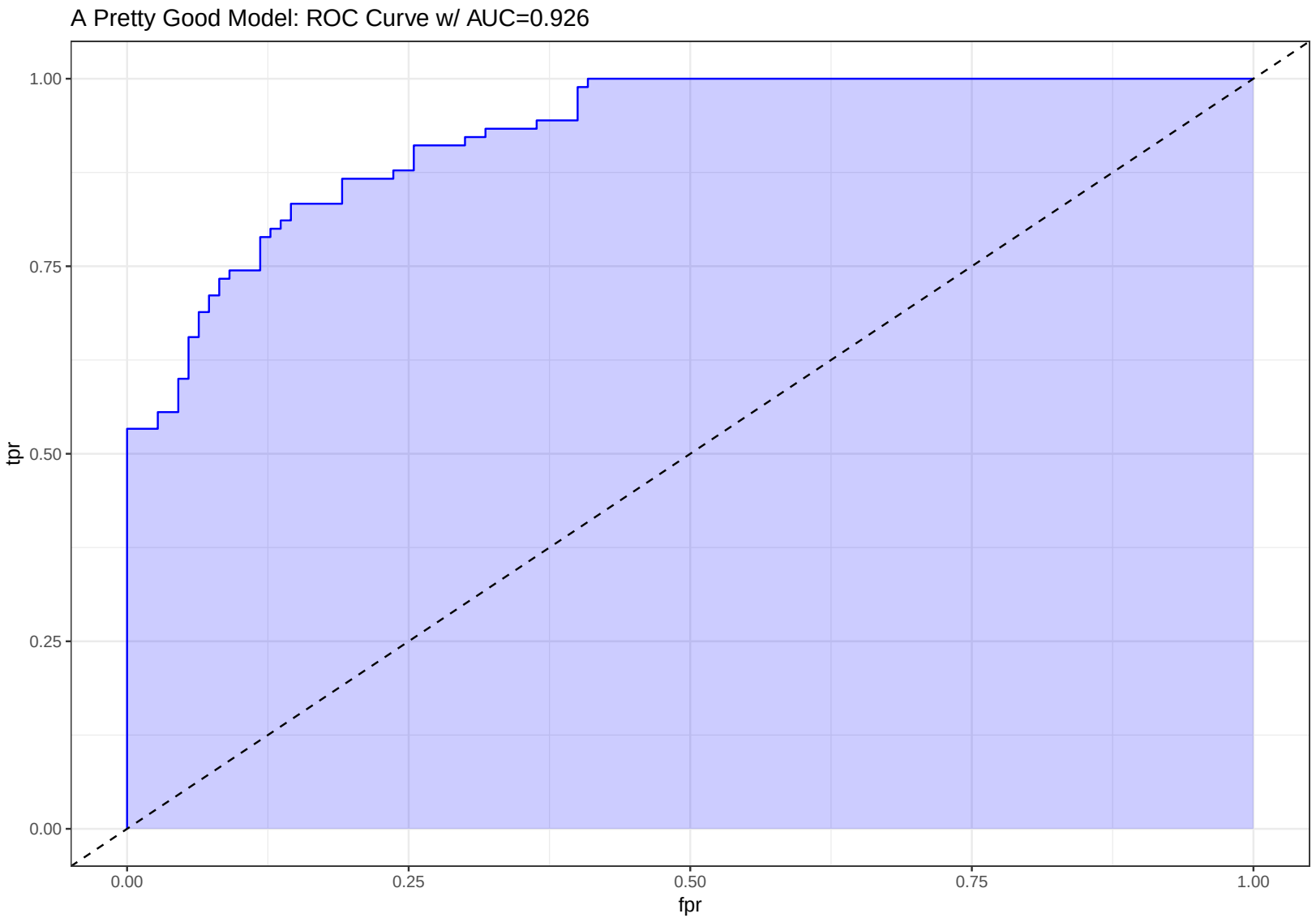
What does "worse than guessing" look like?



What does "better than guessing" look like?



What does "pretty good" look like?



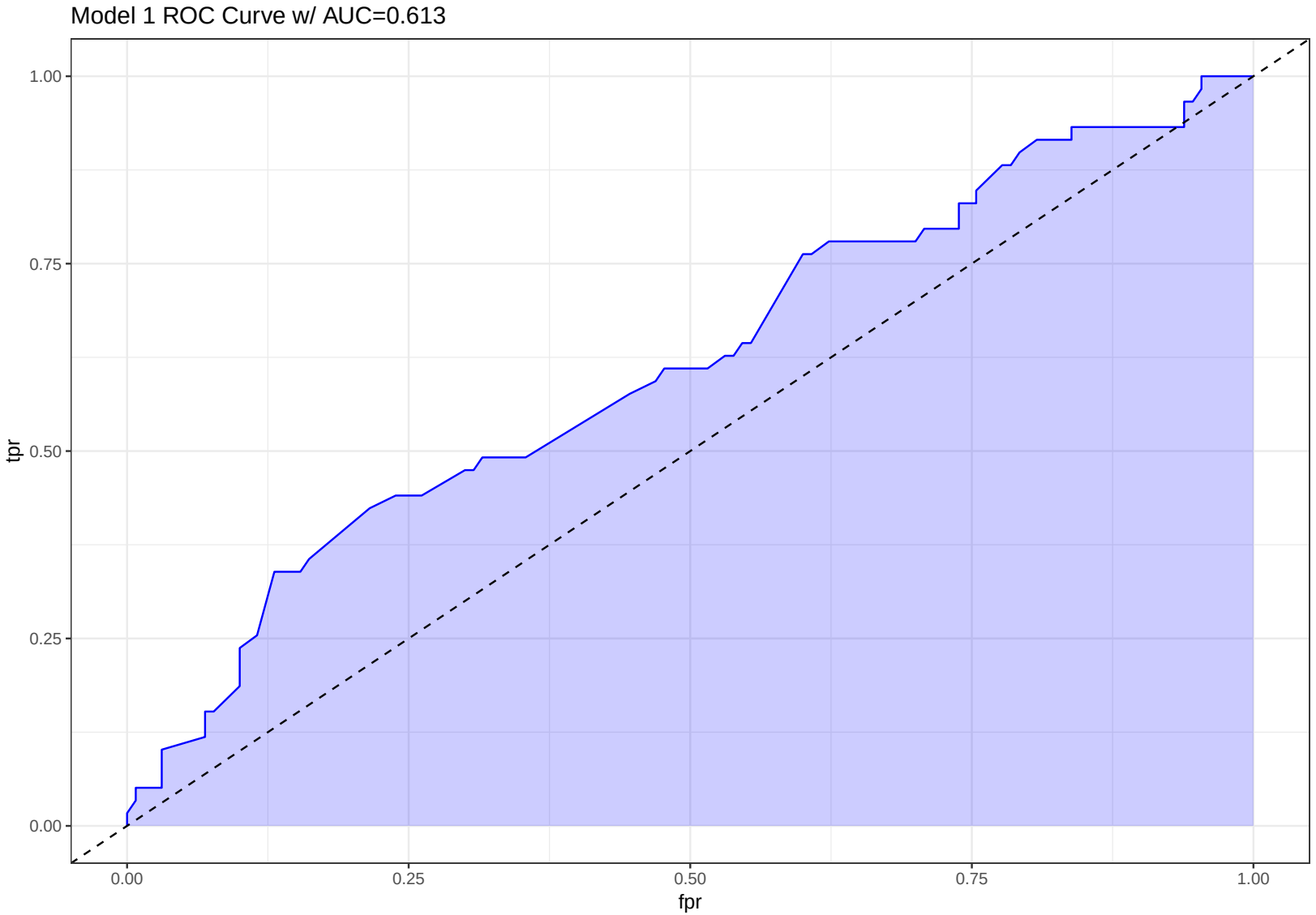
The ROC plot for our Model 1 (code)

```
## requires ROCR package
prob <- predict(model.1, lbw1, type="response")
pred <- prediction(prob, lbw1$low)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
auc <- performance(pred, measure="auc")

auc <- round(auc@y.values[[1]],3)
roc.data <- data.frame(fpr=unlist(perf@x.values),
                       tpr=unlist(perf@y.values),
                       model="GLM")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
    geom_ribbon(alpha=0.2, fill = "blue") +
    geom_line(aes(y=tpr), col = "blue") +
    geom_abline(intercept = 0, slope = 1, lty = "dashed") +
    labs(title = paste0("Model 1 ROC Curve w/ AUC=", auc)) +
    theme_bw()
```

The ROC plot for our Model 1 (Result)



Interpreting the C statistic (0.613) for Model 1

C statistic	Interpretation
0.90 to 1.00	model does an excellent job at discriminating "yes" from "no" (A)
0.80 to 0.90	model does a good job (B)
0.70 to 0.80	model does a fair job (C)
0.60 to 0.70	model does a poor job (D)
0.50 to 0.60	model fails (F)
below 0.50	model is worse than random guessing

Another way to plot the ROC Curve

If we've loaded the pROC package, we can also use the following (admittedly simpler) approach to plot the ROC curve, without ggplot2, and to obtain the C statistic, and a 95% confidence interval around that C statistic.

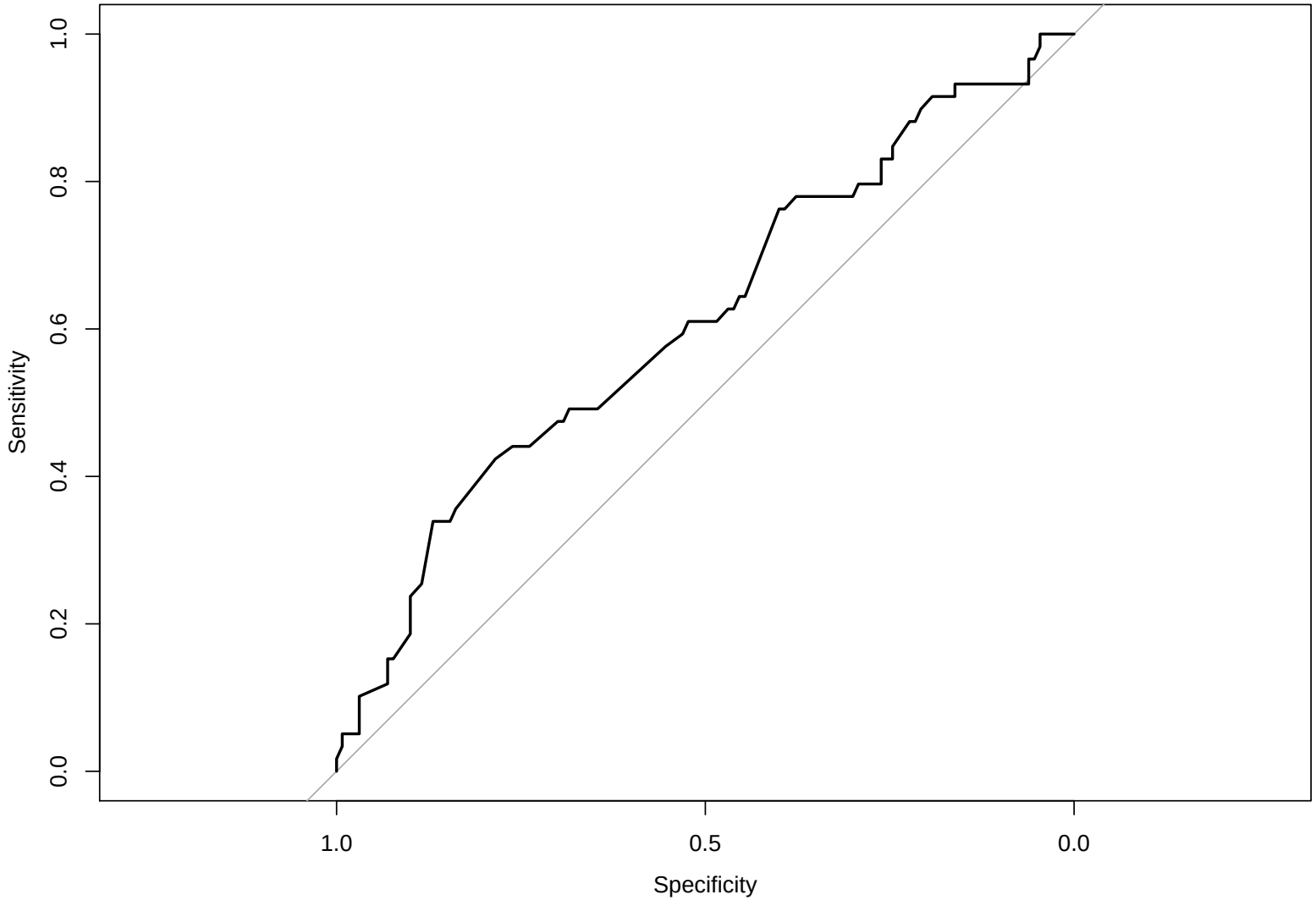
```
## requires pROC package
roc.mod1 <-
    roc(lbw1$low ~ predict(model.1, type="response"),
        ci = TRUE)
```

```
> roc.mod1

Call:
roc.formula(formula = lbw1$low ~ predict(model.1, type = "response"),     ci = TRUE)

Data: predict(model.1, type = "response") in 130 controls (lbw1$low 0) < 59 cases (lbw1$low 1).
Area under the curve: 0.6131
95% CI: 0.5245-0.7017 (DeLong)
```

Result of `plot(roc.mod1)`



Plotting Residuals of a Logistic Regression

Residual Plots for `model.1`?

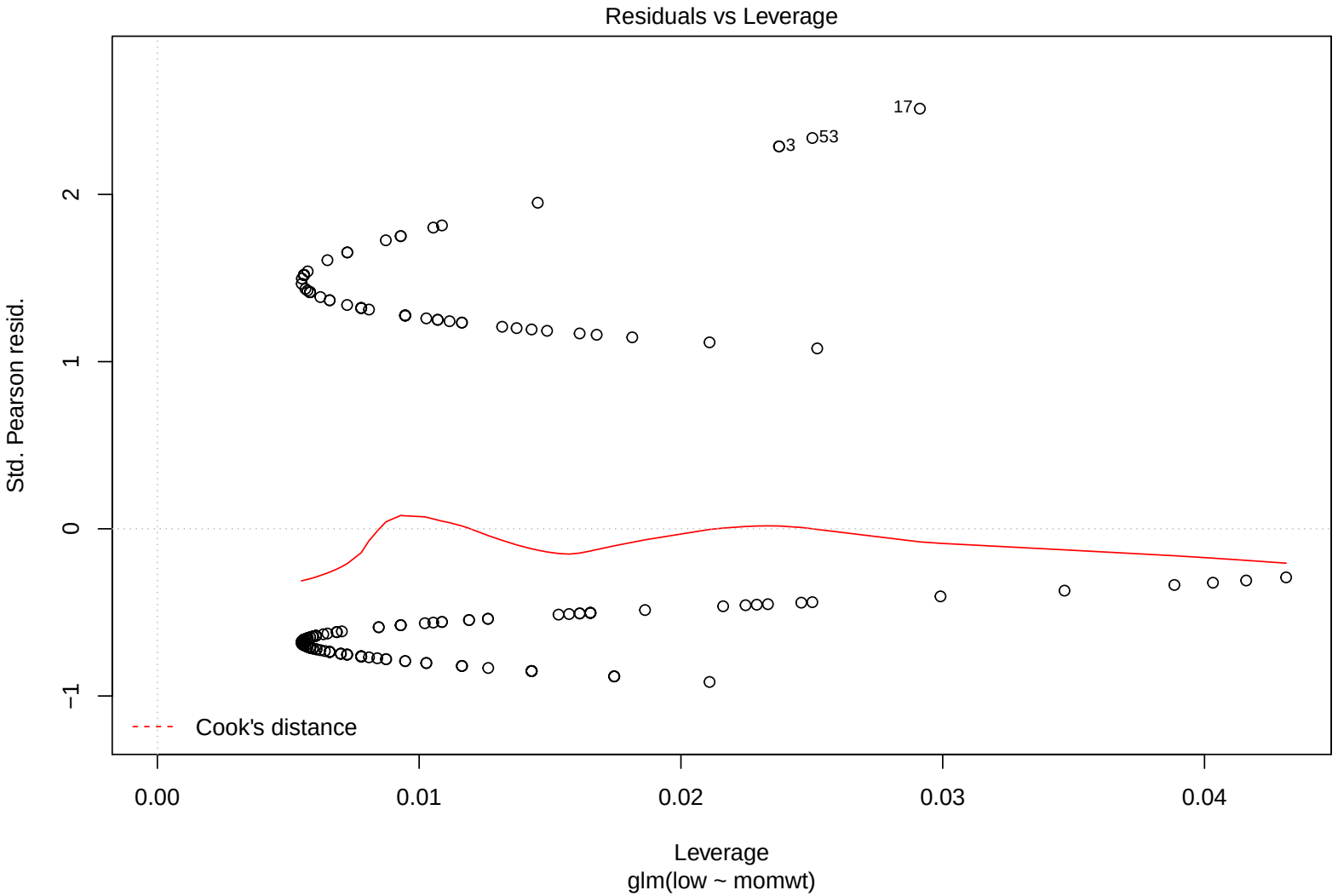
- Yes/No outcomes contain less information than quantitative outcomes
- Residuals cannot be observed - predicted
  - There are several different types of residuals defined
- Assumptions of logistic regression are different
  - Model is deliberately non-linear
  - Error variance is a function of the mean, so it isn't constant
  - Errors aren't assumed to follow a Normal distribution
  - Only thing that's the same: leverage and influence

So, plot 5 (residuals/leverage/influence) can be a little useful, but that's it.

- We'll need better diagnostic tools down the line.

Semi-Useful Residual Plot

```
plot(model.1, which = 5)
```



Building a Bigger Model

Model 2: A "Kitchen Sink" Logistic Regression

```
model.2 <- glm(low ~ momwt + age + ftv + ht + race_f +
                     preterm + smoke + ui,
               data = lbw1, family = binomial)
```

Variable	Description
`low`	indicator of low birth weight (< 2500 g)
`momwt`	mom's weight at last menstrual period (lbs.)
`age`	age of mother in years
`ftv`	physician visits in first trimester (0 to 6)
`ht`	history of hypertension: 1 = yes, 0 = no
`race_f`	race of mom: white, black, other
`preterm`	prior premature labor: 1 = yes, 0 = no
`smoke`	1 = smoked during pregnancy, 0 = did not
`ui`	uterine irritability: 1 = yes, 0 = no

model.2

```
Call:  glm(formula = low ~ momwt + age + ftv + ht + race_f + p
    smoke + ui, family = binomial, data = lbw1)

Coefficients:
(Intercept)        momwt          age          ftv
    0.64448     -0.01508     -0.03955      0.05090
         ht  race_fblack  race_fother   pretermyes
    1.86043      1.21879      0.81944      1.21851
      smoke           ui
    0.85946      0.71930

Degrees of Freedom: 188 Total (i.e. Null);  179 Residual
Null Deviance:      234.7
Residual Deviance: 196.8    AIC: 216.8
```

Comparing model.2 to model.1

```
anova(model.1, model.2)
```

```
Analysis of Deviance Table

Model 1: low ~ momwt
Model 2: low ~ momwt + age + ftv + ht + race_f + preterm + smo
  Resid. Df Resid. Dev Df Deviance
1       187     228.69
2       179     196.75  8   31.941
```

```
pchisq(31.94, 8, lower.tail = FALSE)
```

```
[1] 9.547465e-05
```

Comparing model.2 to model.1

```
glance(model.2)
```

```
# A tibble: 1 x 7
  null.deviance df.null logLik   AIC   BIC deviance
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>
1          235.     188  -98.4  217.  249.     197.
# ... with 1 more variable: df.residual <int>
```

```
glance(model.1)
```

```
# A tibble: 1 x 7
  null.deviance df.null logLik   AIC   BIC deviance
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>
1          235.     188  -114.  233.  239.     229.
# ... with 1 more variable: df.residual <int>
```

Interpreting `model.2`

```
> round(summary(model.2)$coef, 3)
             Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.644      1.224   0.527    0.598
momwt          -0.015      0.007  -2.143    0.032
age            -0.040      0.038  -1.032    0.302
ftv             0.051      0.175   0.290    0.772
ht              1.860      0.708   2.627    0.009
race_fblack     1.219      0.533   2.286    0.022
race_fother     0.819      0.450   1.819    0.069
pretermyes      1.219      0.463   2.632    0.008
smoke           0.859      0.410   2.097    0.036
ui              0.719      0.463   1.552    0.121
```

- Larger Mom `momwt` is associated with a smaller log odds of LBW holding all other predictors constant.

Impact of these predictors via odds ratios

```
exp(coef(model.2)); exp(confint(model.2))
```

Variable	OR est.	2.5%	97.5%
momwt	0.985	0.971	0.998
age	0.961	0.890	1.035
ftv	1.052	0.739	1.478
ht	6.426	1.662	28.187
race_fblack	3.383	1.192	9.808
race_fother	2.269	0.947	5.597
pretermyes	3.382	1.378	8.575
smoke	2.362	1.067	5.375
ui	2.053	0.818	5.101

- Larger Mom `momwt` is associated with a smaller odds of LBW (est OR 0.985, 95% CI 0.971, 0.998) holding all other predictors constant.
- What appears to be associated with larger odds of LBW?

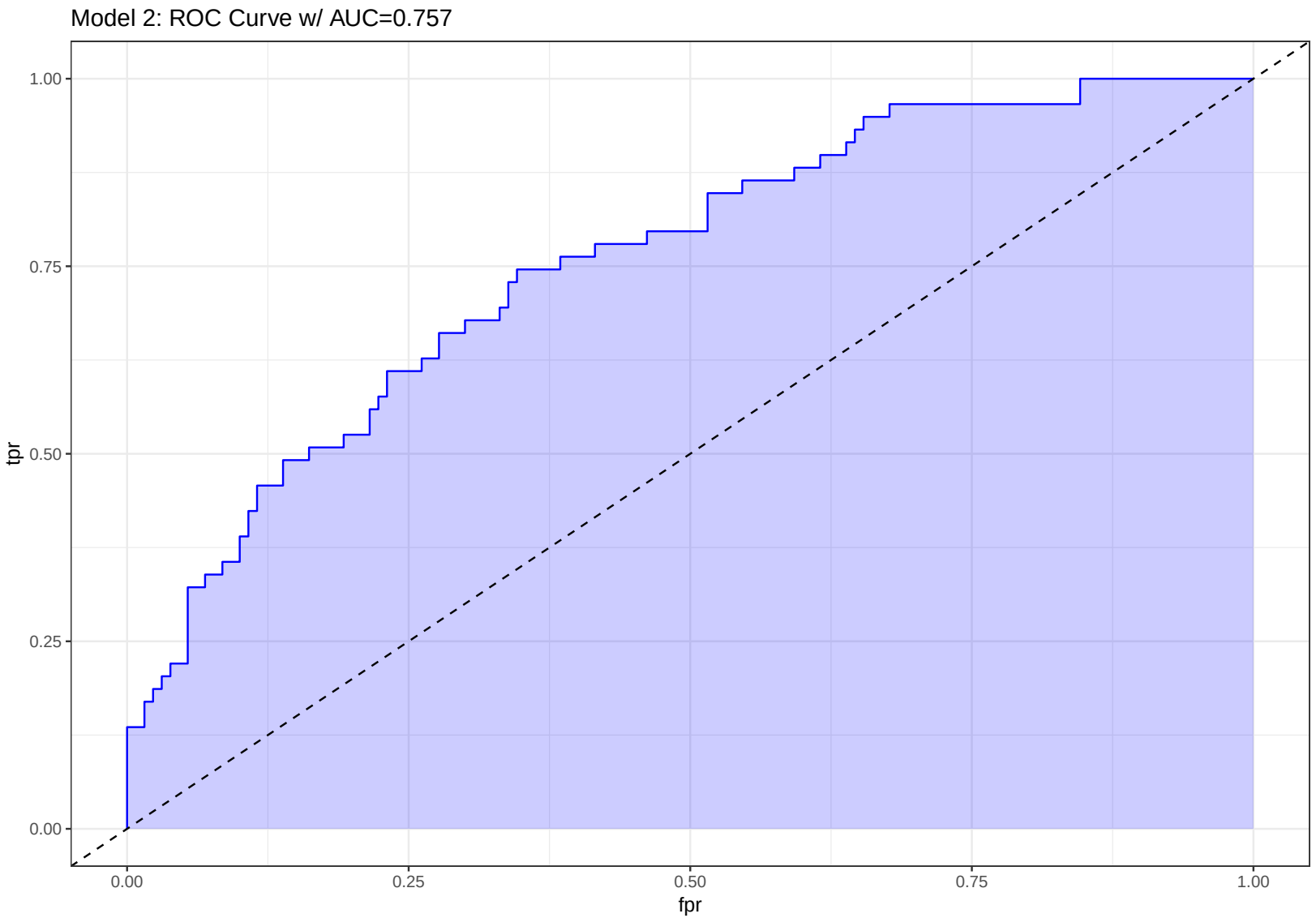
ROC curve for Model 2 (Code)

```
prob <- predict(model.2, lbw1, type="response")
pred <- prediction(prob, lbw1$low)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
auc <- performance(pred, measure="auc")

auc <- round(auc@y.values[[1]],3)
roc.data <- data.frame(fpr=unlist(perf@x.values),
                       tpr=unlist(perf@y.values),
                       model="GLM")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
    geom_ribbon(alpha=0.2, fill = "blue") +
    geom_line(aes(y=tpr), col = "blue") +
    geom_abline(intercept = 0, slope = 1, lty = "dashed") +
    labs(title = paste0("Model 2: ROC Curve w/ AUC=", auc)) +
    theme_bw()
```

ROC curve for Model 2 (Result)

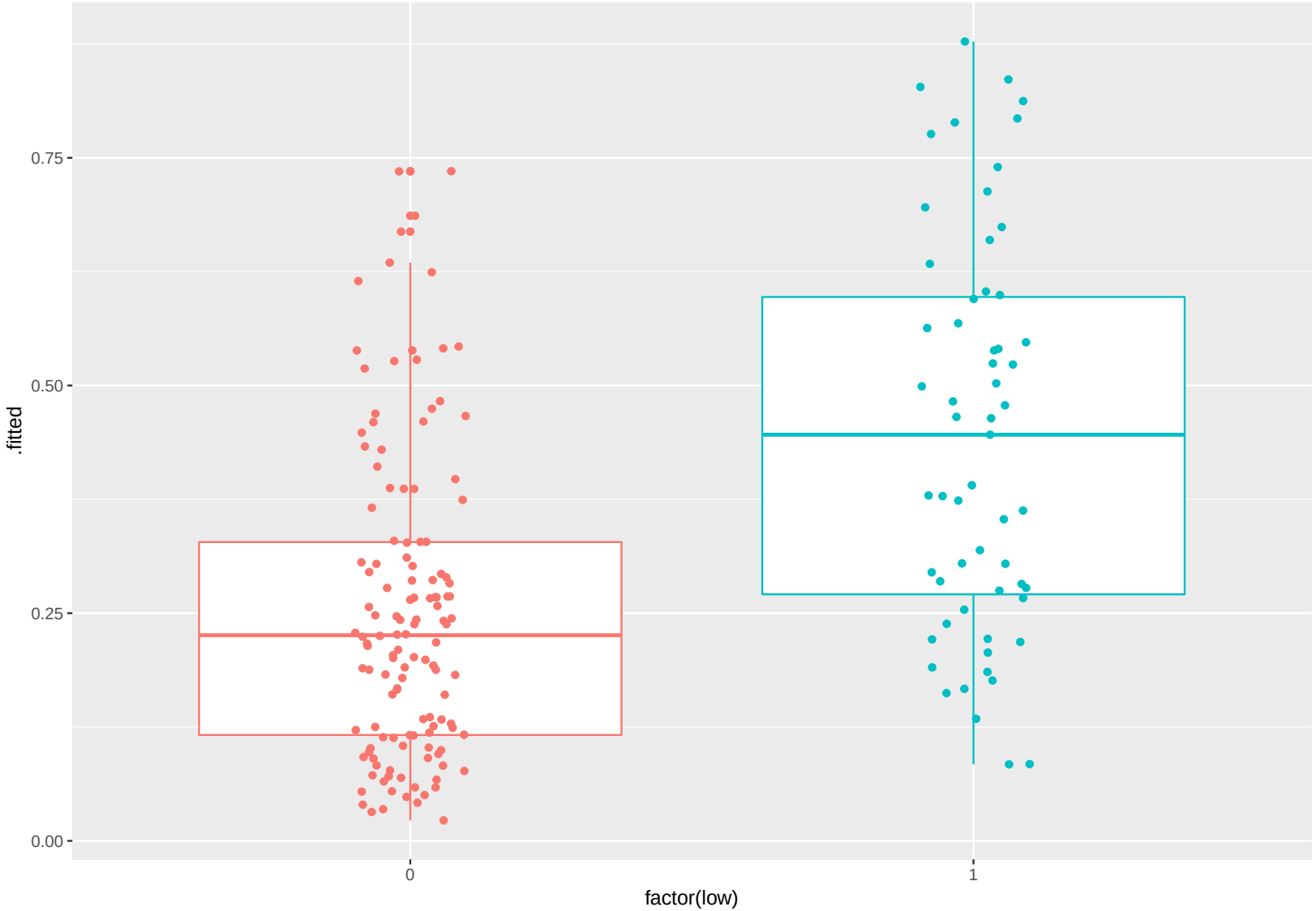


Using augment to capture the fitted probabilities

```
mod2_aug <- augment(model.2, lbw1,
                    type.predict = "response")
head(mod2_aug, 3)
```

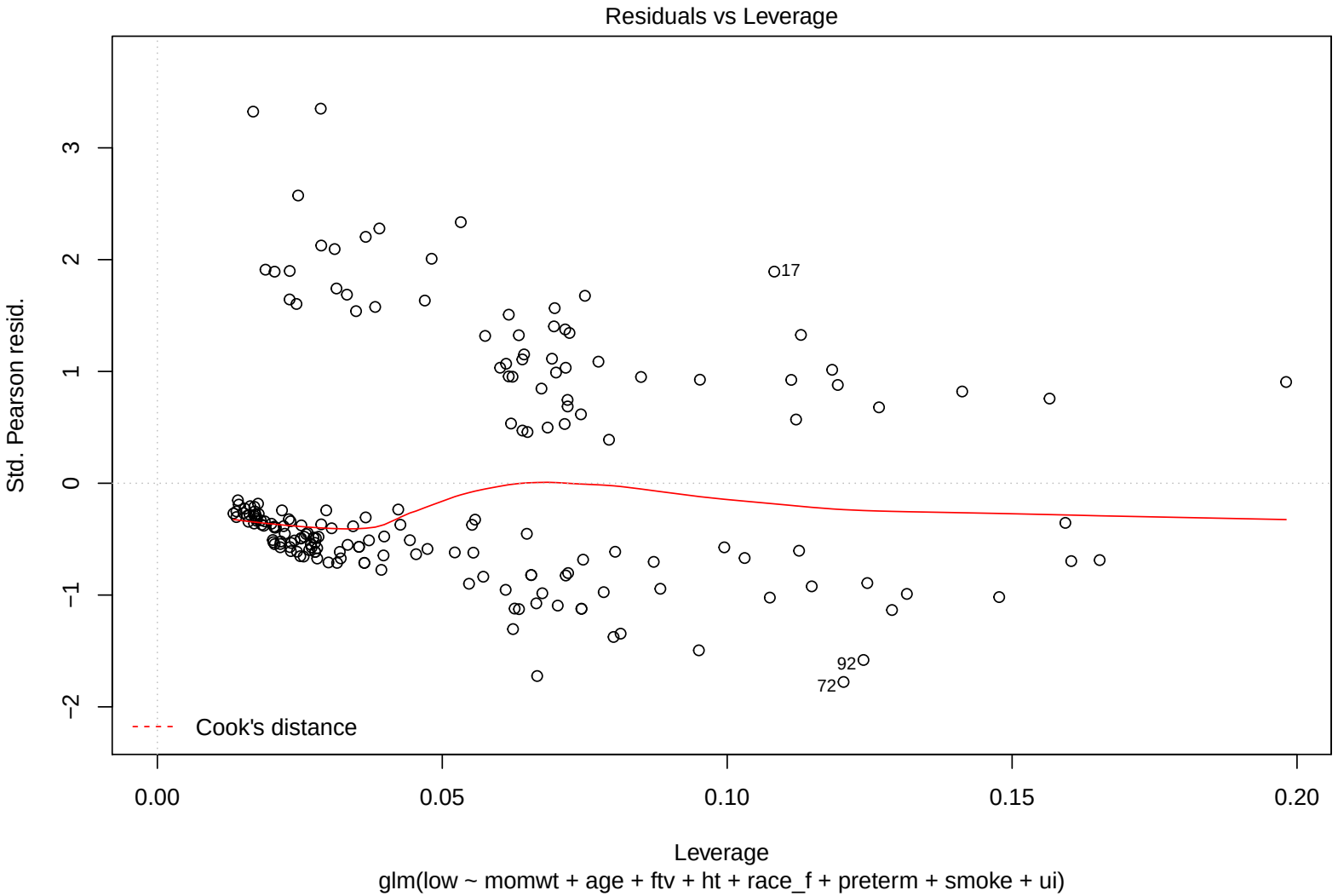
```
# A tibble: 3 x 17
  subject   low momwt   age   ftv    ht race_f preterm smoke
    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <fct>  <fct>   <dbl>
1       4     1   120    28     0     0 other  yes         1
2      10     1   130    29     2     0 white  no          0
3      11     1   187    34     0     1 black  no          1
# ... with 8 more variables: ui <dbl>, .fitted <dbl>,
#   .se.fit <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>,
#   .cooksd <dbl>, .std.resid <dbl>
```

Plotting Model 2 Fits by Observed LBW status



Residuals, Leverage and Influence

```
plot(model.2, which = 5)
```



Next Time

- Fitting Logistic Regression models with `lrm`