

432 Class 9 Slides

github.com/THOMASELOVE/432-2018

2018-02-13

Setup

```
library(skimr)
library(broom)
library(Hmisc)
library(rms)
library(pROC)
library(ROCR)
library(tidyverse)
```

Today's Materials

- Logistic Regression and the Low Birth Weight data
- How well does the 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
- Fitting a Logistic Model with `lrm`
 - Nagelkerke R^2 , Somers' d etc.
 - Validating Summary Statistics
 - Summaries of Effects
 - Plotting In-Sample Predictions
 - Influence
 - Calibration
 - Nomograms

The Low Birth Weight data, again

```
lbw1 <- read.csv("data/lbw.csv") %>% tbl_df

lbw1 <- lbw1 %>%
  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")) %>%
  select(subject, low, lwt, 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)
lwt	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.

Model 1

Our current model

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

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

Coefficients:

(Intercept)	lwt
0.99831	-0.01406

Degrees of Freedom: 188 Total (i.e. Null); 187 Residual

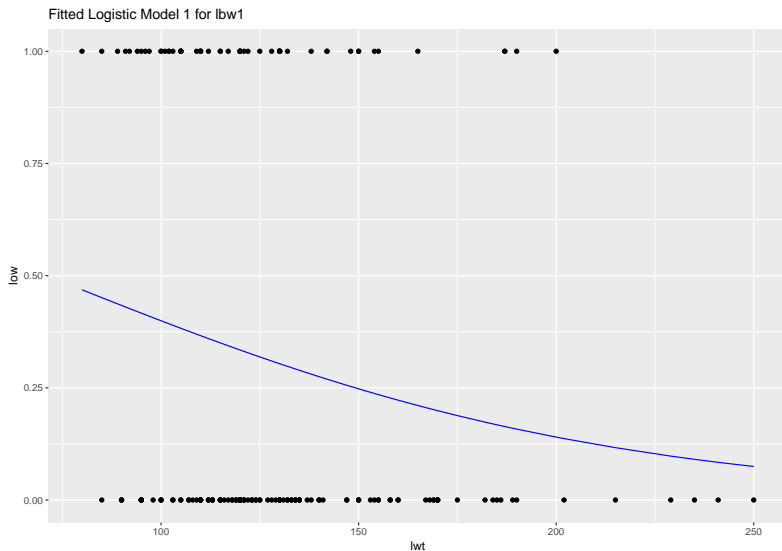
Null Deviance: 234.7

Residual Deviance: 228.7 AIC: 232.7

Plotting the Logistic Regression Model (as last time)

```
mod1.aug <- augment(model.1, lbw1,  
                     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 lbw1")
```


Plotting the Logistic Regression Model (as last time)

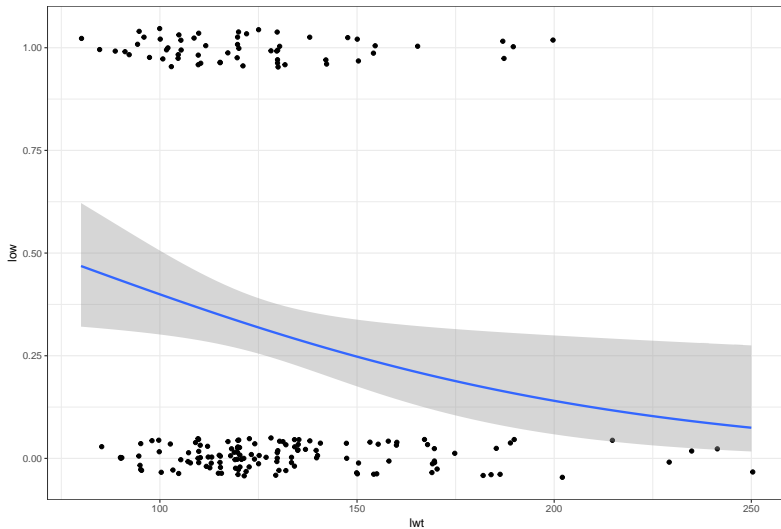


Plotting a Simple Logistic Model using binomial_smooth

```
binomial_smooth <- function(...) {  
  geom_smooth(method = "glm",  
              method.args = list(family = "binomial"), ...)  
}  
  
ggplot(lbw1, aes(x = lwt, 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

Logistic Regression Model 1



glance on model.1

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

	null.deviance	df.null	logLik	AIC	BIC
1	234.672	188	-114.3453	232.6907	239.1742
	deviance	df.residual			
1	228.6907	187			

- Deviance = $-2 \times \log(\text{likelihood})$
- 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 ~ lwt, 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
lwt          -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
1wt	-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.
- 1wt has a Wald Z of -2.279, yielding $p = 0.0227$
 - H_0 : 1wt does not have an effect on the log odds of low
 - H_A : 1wt does have such an effect
- If the coefficient (on the logit scale) for 1wt was truly 0, this would mean that:
 - the log odds of low birth weight did not change based on 1wt,
 - the odds of low birth weight were unchanged based on 1wt ($OR = 1$), and
 - the probability of low birth weight was unchanged based on the 1wt.

Confidence Intervals for Coefficients

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

```
(Intercept)          lwt  
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  
lwt          -0.02696198 -0.002650036
```

- The coefficient of `lwt` 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)	lwt
2.7137035	0.9860401

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

	2.5 %	97.5 %
(Intercept)	0.6180617	13.6228447
lwt	0.9733982	0.9973535

- Odds Ratio for low based on a one pound increase in lwt is 0.986 (95% CI: 0.973, 0.997).
 - Estimated odds of low birth weight will be smaller (odds < 1) for those with larger lwt 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?

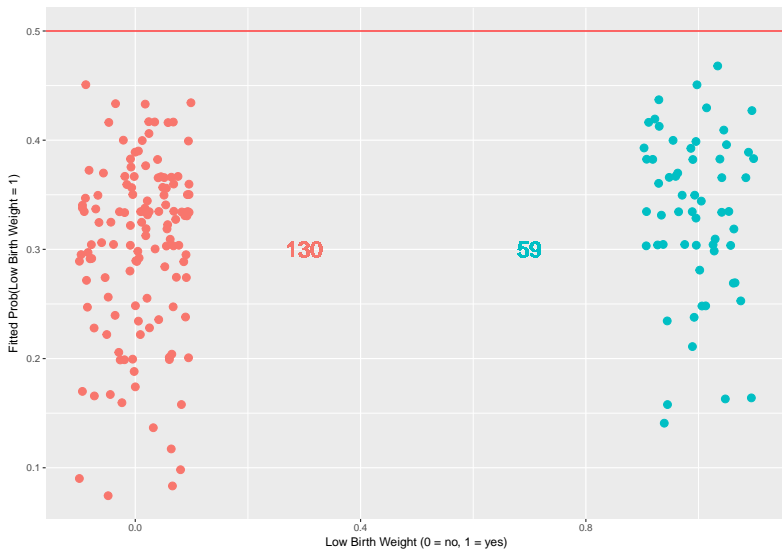
One possible rule: if predicted $\Pr(\text{low} = 1) \geq 0.5$, then we predict “low birth weight”

```
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



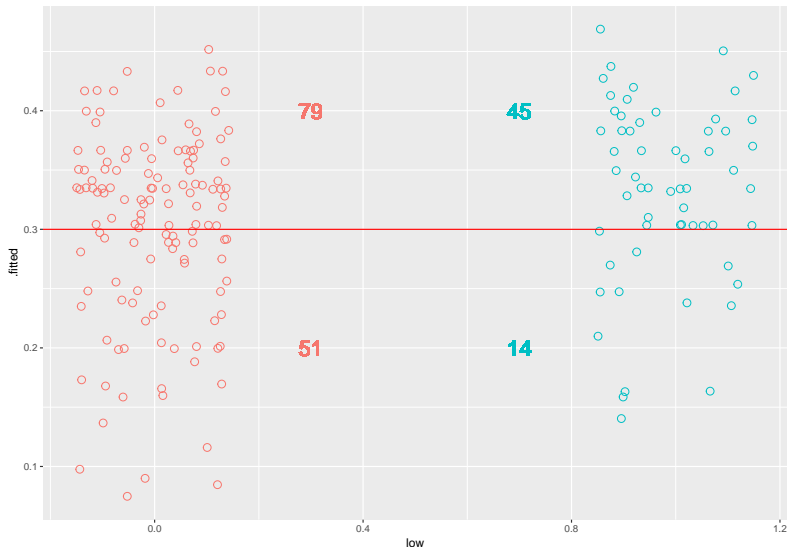
How Well Does Our model.1 Classify Subjects?

A new rule: if predicted $\Pr(\text{low} = 1) \geq 0.3$, 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 Weight", "OK Birth Weight"))  
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 $\Pr(\text{low birth weight})$.
- 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 $\Pr(\text{low} = 1)$ to be small, but the baby is born at a low birth weight.
- A false negative occurs when we predict $\Pr(\text{low} = 1)$ 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 $\Pr(\text{Mom weight} < R \mid \text{baby actually has low} = 1)$.
- The false positive fraction (FPF) for a specific weight cutoff R is $\Pr(\text{Mom weight} < R \mid \text{baby has low} = 0)$.

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))

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))
```

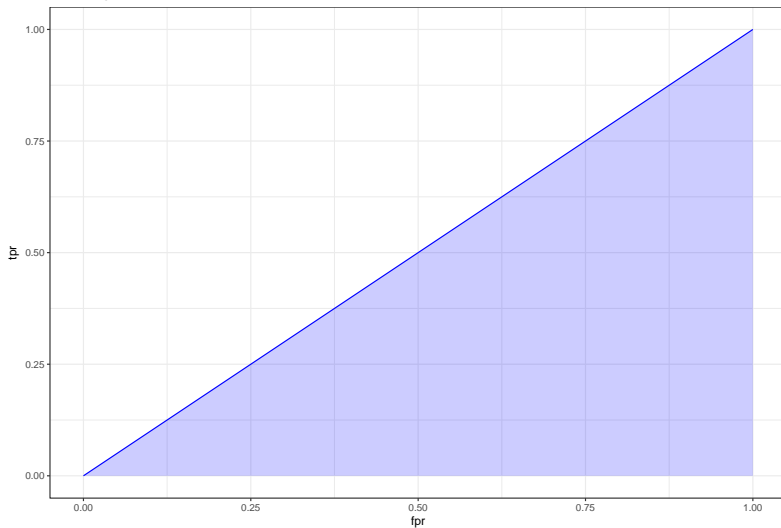
What if we are guessing?

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?

Guessing: ROC Curve w/ AUC=0.5



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?

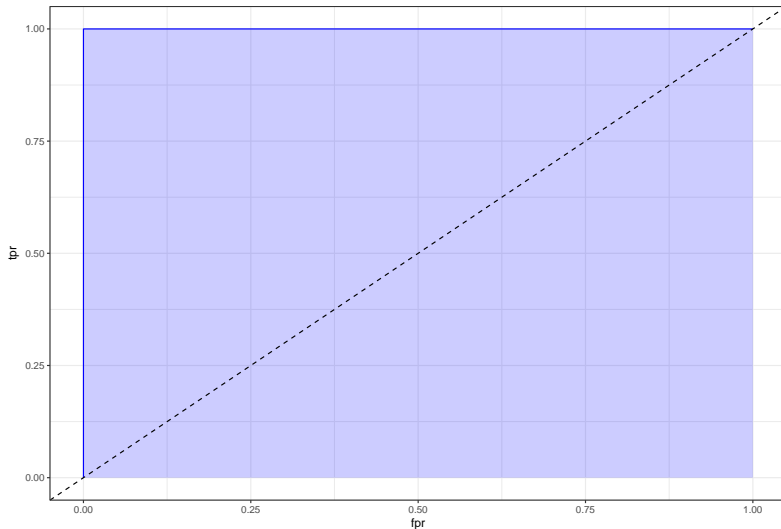
If we're classifying subjects perfectly, then we have a TPR of 1 and an FPR of 0.

- 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.

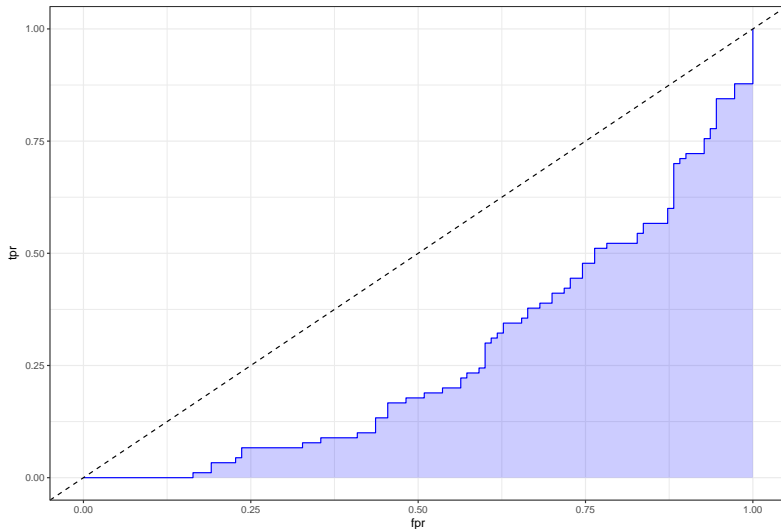
What if our model classifies things perfectly?

Perfect Prediction: ROC Curve w/ AUC=1



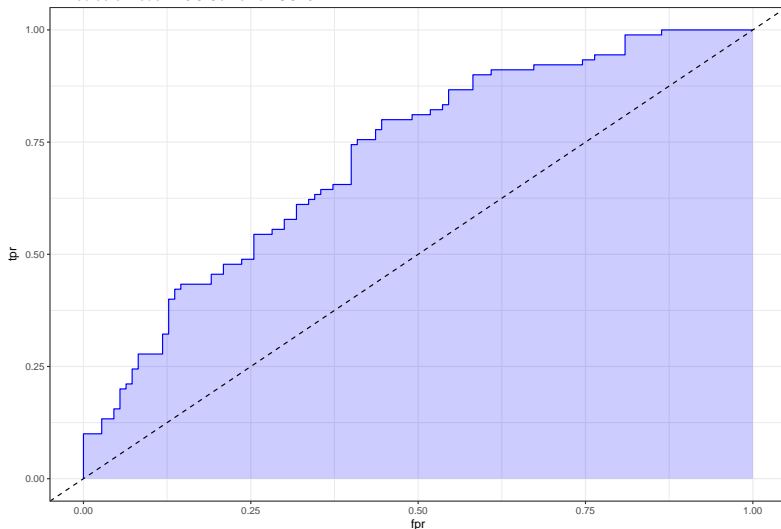
What does “worse than guessing” look like?

A Bad Model: ROC Curve w/ AUC=0.269



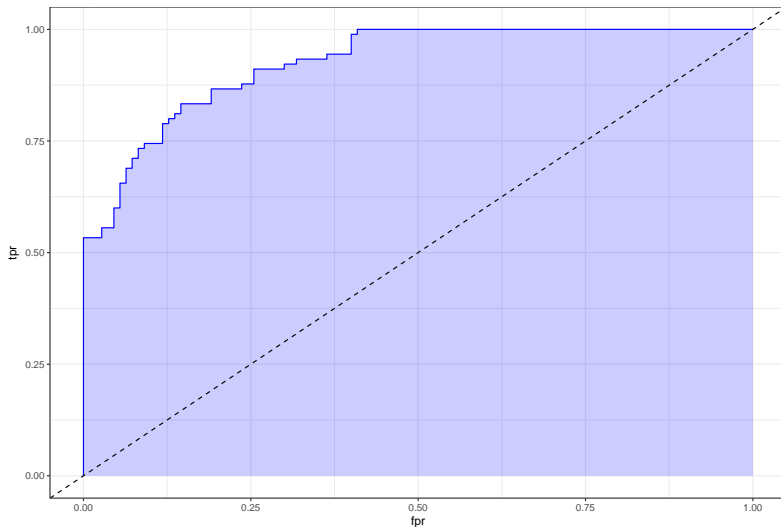
What does “better than guessing” look like?

A Mediocre Model: ROC Curve w/ AUC=0.717



What does “pretty good” look like?

A Pretty Good Model: ROC Curve w/ AUC=0.926



This is as far as we got in Class 09

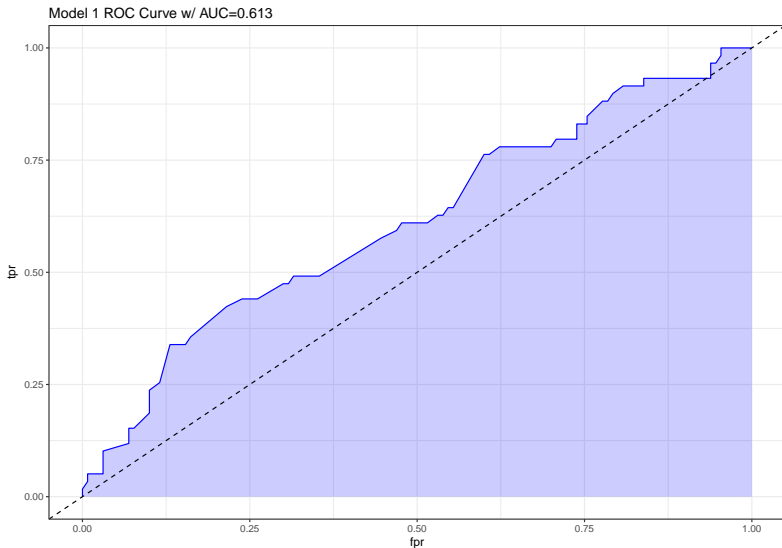
The ROC plot for our Model 1 (code)

```
## 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.

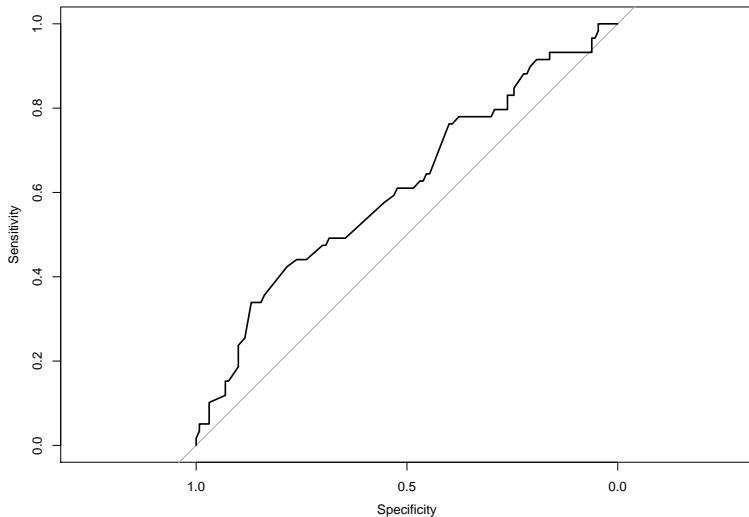
```
## requires pROC package
roc.mod1 <-
  roc(lbw1$low ~ predict(model.1, type="response"),
      ci = TRUE)
```

```
> 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?

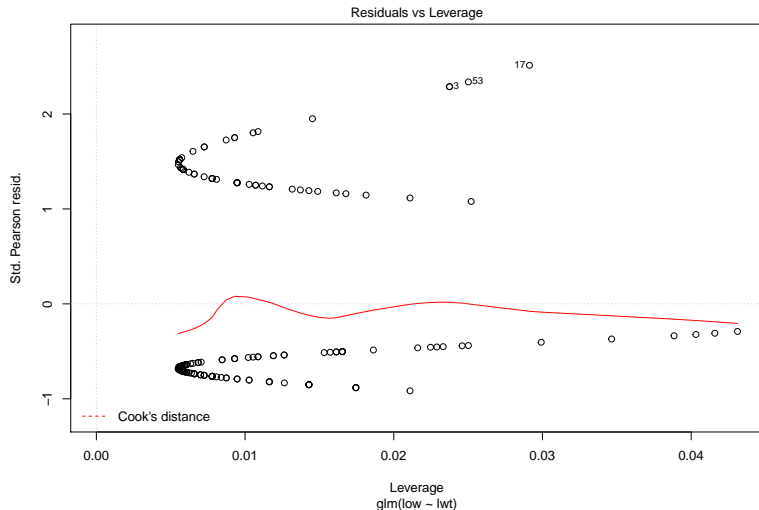
- Yes/No outcomes contain less information than quantitative outcomes
- Residuals cannot be observed - predicted
 - There are several different types of residuals defined
- Assumptions of logistic regression are different
 - Model is deliberately non-linear
 - Error variance is a function of the mean, 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 ~ lwt + age + ftv + ht + race_f +  
               preterm + smoke + ui,  
               data = lbw1, family = binomial)
```

Variable	Description
low	indicator of low birth weight (< 2500 g)
lwt	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 ~ lwt + age + ftv + ht + race_f + preterm +  
smoke + ui, family = binomial, data = lbw1)
```

Coefficients:

(Intercept)	lwt	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 ~ lwt

Model 2: low ~ lwt + age + ftv + ht + race_f + preterm + smoke

	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)
```

	null.deviance	df.null	logLik	AIC	BIC
1	234.672	188	-98.37504	216.7501	249.1676
	deviance	df.residual			
1	196.7501	179			

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

	null.deviance	df.null	logLik	AIC	BIC
1	234.672	188	-114.3453	232.6907	239.1742
	deviance	df.residual			
1	228.6907	187			

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	
lwt	-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 lwt 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%
lwt	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 lwt 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?

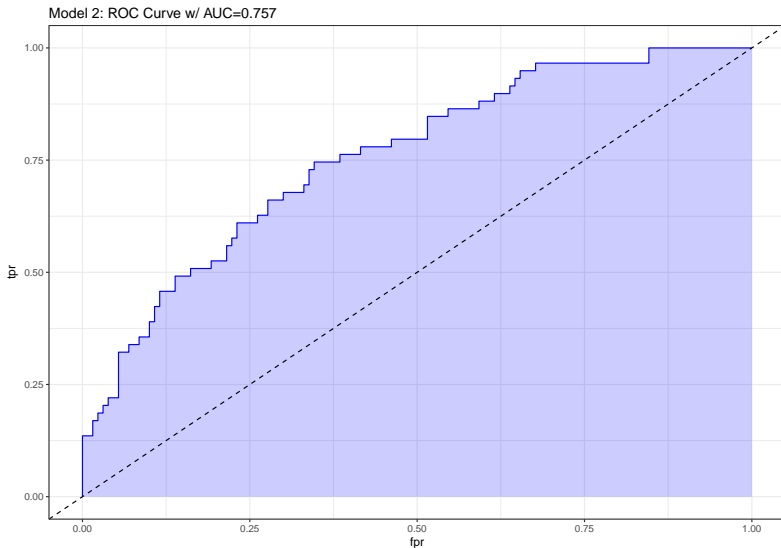
ROC curve for Model 2 (Code)

```
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)

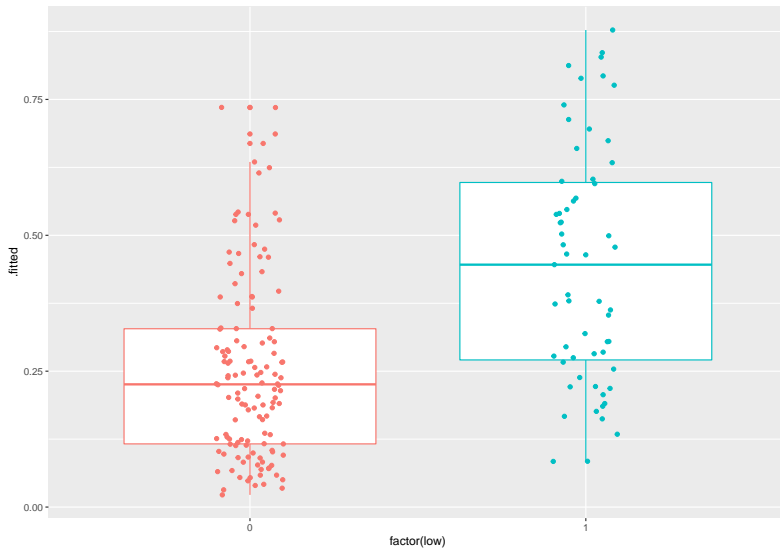


Using augment to capture the fitted probabilities

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

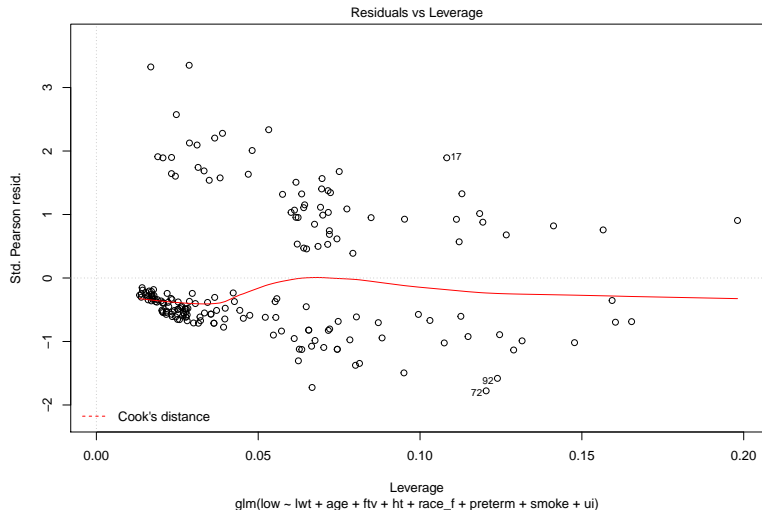
	subject	low	lwt	age	ftv	ht	race_f	preterm	smoke	ui
1	4	1	120	28	0	0	other	yes	1	1
2	10	1	130	29	2	0	white	no	0	1
3	11	1	187	34	0	1	black	no	1	0
	.fitted		.se.fit		.resid		.hat		.sigma	
1	0.7932350	0.10827348	0.6806406	0.07148005	1.050016					
2	0.1622751	0.08507415	1.9070723	0.05324823	1.041036					
3	0.6032496	0.21773929	1.0054096	0.19808668	1.047977					
	.cooksd		.std.resid							
1	0.002161114	0.7063537								
2	0.030667798	1.9599685								
3	0.020259128	1.1227403								

Plotting Model 2 Fits by Observed LBW status



Residuals, Leverage and Influence

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



Logistic Regression using the `lrm` function

Fitting Model 2 again (as Model 3)

```
dd <- datadist(lbw1)
options(datadist = "dd")

model.3 <- lrm(low ~ lwt + age + ftv + ht + race_f +
               preterm + smoke + ui,
               data = lbw1, x = TRUE, y = TRUE)
```

model.3 output

```
> model.3
```

```
Logistic Regression Model
```

```
lrm(formula = low ~ lwt + age + ftv + ht + race_f + preterm +  
      smoke + ui, data = lbw1, x = TRUE, y = TRUE)
```

		Model Likelihood	Discrimination	Rank Discrim.
		Ratio Test	Indexes	Indexes
Obs	189	LR chi2	R2	C
0	130	d.f.	g	Dxy
1	59	Pr(> chi2)	gr	gamma
max deriv	0.0003	<0.0001	gp	tau-a
			Brier	0.222

	Coef	S.E.	Wald Z	Pr(> Z)
Intercept	0.6445	1.2239	0.53	0.5985
lwt	-0.0151	0.0070	-2.14	0.0321
age	-0.0395	0.0383	-1.03	0.3019
ftv	0.0509	0.1755	0.29	0.7717
ht	1.8604	0.7082	2.63	0.0086
race_f=black	1.2188	0.5332	2.29	0.0223
race_f=other	0.8194	0.4505	1.82	0.0689
preterm=yes	1.2185	0.4630	2.63	0.0085
smoke	0.8595	0.4098	2.10	0.0360
ui	0.7193	0.4634	1.55	0.1206

The Top Section

```
> model.3
Logistic Regression Model

1rm(formula = low ~ lwt + age + ftv + ht + race_f + preterm +
      smoke + ui, data = lbw1, x = TRUE, y = TRUE)

              Model Likelihood          Discrimination          Rank Discrim.
              Ratio Test          Indexes          Indexes
obs              189      LR chi2      37.92      R2              0.256      C              0.757
0                130      d.f.              9      g              1.263      Dxy             0.514
1                 59      Pr(> chi2) <0.0001      gr             3.538      gamma          0.515
max |deriv| 0.0003              gp             0.228      tau-a          0.222
              Brier             0.174
```

- Likelihood ratio test = drop in deviance test
- R^2 = Nagelkerke R^2 = not a percentage of anything
- C = Area under the ROC curve
- Dxy = Somers' d, and note $C = 0.5 + Dxy/2$

The Coefficients Summary

	Coef	S.E.	Wald Z	Pr(> Z)
Intercept	0.6445	1.2239	0.53	0.5985
lwt	-0.0151	0.0070	-2.14	0.0321
age	-0.0395	0.0383	-1.03	0.3019
ftv	0.0509	0.1755	0.29	0.7717
ht	1.8604	0.7082	2.63	0.0086
race_f=black	1.2188	0.5332	2.29	0.0223
race_f=other	0.8194	0.4505	1.82	0.0689
preterm=yes	1.2185	0.4630	2.63	0.0085
smoke	0.8595	0.4098	2.10	0.0360
ui	0.7193	0.4634	1.55	0.1206

ROC Curve Analysis (code)

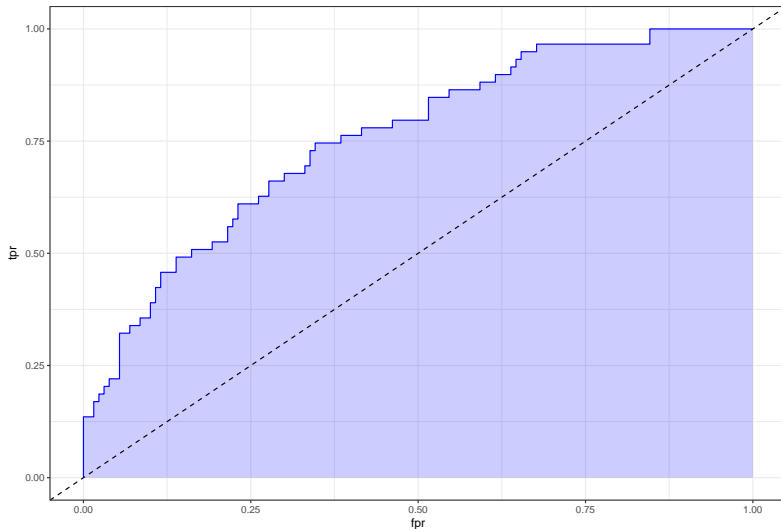
- Note: change prob to describe type = "fitted"
- Note: make sure lbw1 in prob is a data frame

```
prob <- predict(model.3, data.frame(lbw1), type="fitted")
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 3: ROC Curve w/ AUC=", auc)) +
  theme_bw()
```

ROC Curve Analysis (resulting plot)

Model 3: ROC Curve w/ AUC=0.757



Validating Logistic Model Summary Statistics

lrm has a `validate` tool to help perform resampling validation of a model, with or without backwards step-wise variable selection. Here, we'll validate our model's summary statistics using 100 bootstrap replications.

```
set.seed(432001)
validate(model.3, B = 100)
```

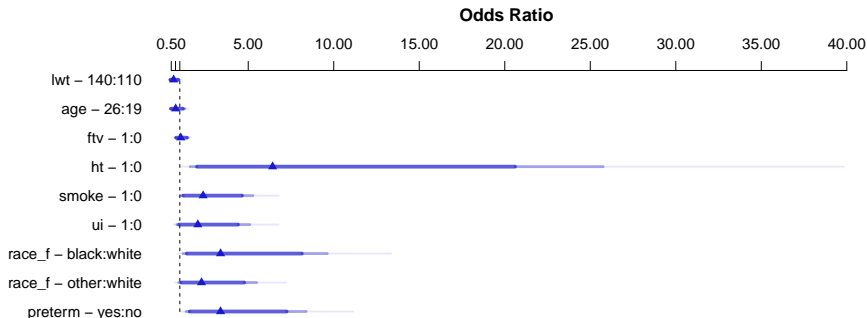
```
> set.seed(432001)
> validate(model.3, B = 100)
```

	index.orig	training	test	optimism	index.corrected	n
Dxy	0.5142	0.5620	0.4640	0.0980	0.4162	100
R2	0.2557	0.3031	0.2041	0.0991	0.1566	100
Intercept	0.0000	0.0000	-0.1649	0.1649	-0.1649	100
Slope	1.0000	1.0000	0.7502	0.2498	0.7502	100

Plotting the Summary of the lrm approach

The summary function applied to an lrm fit shows the effect size comparing the 25th to the 75th percentile of each predictor.

```
plot(summary(model.3))
```



summary(model.3)

```
> summary(model.3)
```

	Effects			Response : low				
Factor	Low	High	Diff.	Effect	S.E.	Lower 0.95	Upper 0.95	
lwt	110	140	30	-0.45233	0.21103	-0.865940	-0.038713	
Odds Ratio	110	140	30	0.63615	NA	0.420660	0.962030	
age	19	26	7	-0.27684	0.26814	-0.802390	0.248710	
Odds Ratio	19	26	7	0.75818	NA	0.448260	1.282400	
ftv	0	1	1	0.05090	0.17546	-0.292990	0.394790	
Odds Ratio	0	1	1	1.05220	NA	0.746030	1.484100	
ht	0	1	1	1.86040	0.70817	0.472430	3.248400	
Odds Ratio	0	1	1	6.42650	NA	1.603900	25.750000	
smoke	0	1	1	0.85946	0.40985	0.056170	1.662700	
Odds Ratio	0	1	1	2.36190	NA	1.057800	5.273800	
ui	0	1	1	0.71930	0.46343	-0.189000	1.627600	
Odds Ratio	0	1	1	2.05300	NA	0.827790	5.091600	
race_f - black:white	1	2	NA	1.21880	0.53318	0.173780	2.263800	
Odds Ratio	1	2	NA	3.38310	NA	1.189800	9.619600	
race_f - other:white	1	3	NA	0.81944	0.45048	-0.063487	1.702400	
Odds Ratio	1	3	NA	2.26920	NA	0.938490	5.486900	
preterm - yes:no	1	2	NA	1.21850	0.46302	0.311010	2.126000	
Odds Ratio	1	2	NA	3.38220	NA	1.364800	8.381400	

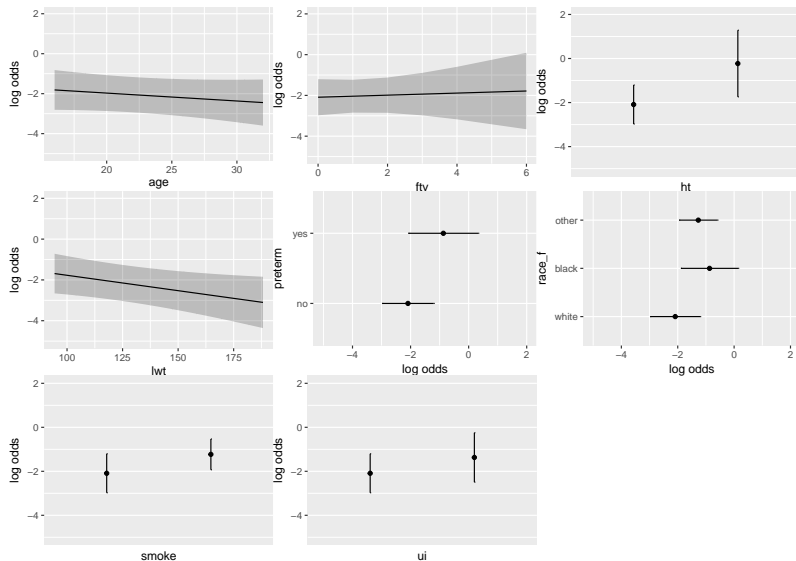
Plot In-Sample Predictions from Model 3

```
ggplot(Predict(model.3))
```

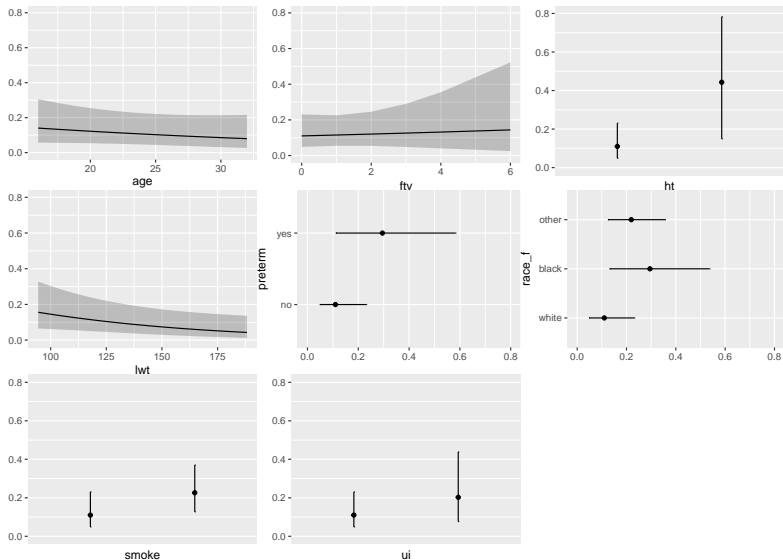
This will plot the effect of each predictor variable (and 95% CI for that effect) across the range of observed values for that predictor, on the log odds of low birth weight. (see next slide)

- To get these plots on the **probability** scale, we add `fun = plogis` (see two slides from now)

ggplot(Predict(model.3))



```
ggplot(Predict(model.3, fun = plogis))
```



ANOVA from the lrm approach

```
anova(model.3)
```

Wald Statistics

Response: low

Factor	Chi-Square	d.f.	P
lwt	4.59	1	0.0321
age	1.07	1	0.3019
ftv	0.08	1	0.7717
ht	6.90	1	0.0086
race_f	6.23	2	0.0444
preterm	6.93	1	0.0085
smoke	4.40	1	0.0360
ui	2.41	1	0.1206
TOTAL	28.62	9	0.0008

Wald test for the model as a whole shows $p = 0.0008$

Any influential points?

```
inf.3 <- which.influence(model.3, cutoff=0.3)
inf.3
```

```
$Intercept
[1] "40" "53"
```

```
$lwt
[1] "17" "53" "72"
```

```
$age
[1] "11" "92"
```

```
$ftv
[1] "48" "52"
```

```
$ht
[1] "72" "110"
```


Influence within the Data Frame

```
show.influence(object = inf.3, dframe = data.frame(lbw1))
```

	Count	lwt	age	ftv	ht	race_f	ui
2	1	130	29	2	0	white	*1
11	1	105	*32	0	0	white	0
17	2	*200	21	2	0	black	*1
40	2	110	15	0	0	*white	0
48	1	120	17	*3	0	white	0
52	1	105	20	*3	0	other	0
53	2	*190	26	0	0	white	0
72	2	* 95	22	0	*1	other	0
92	2	121	*35	1	0	*black	0
110	1	120	22	1	*1	white	0

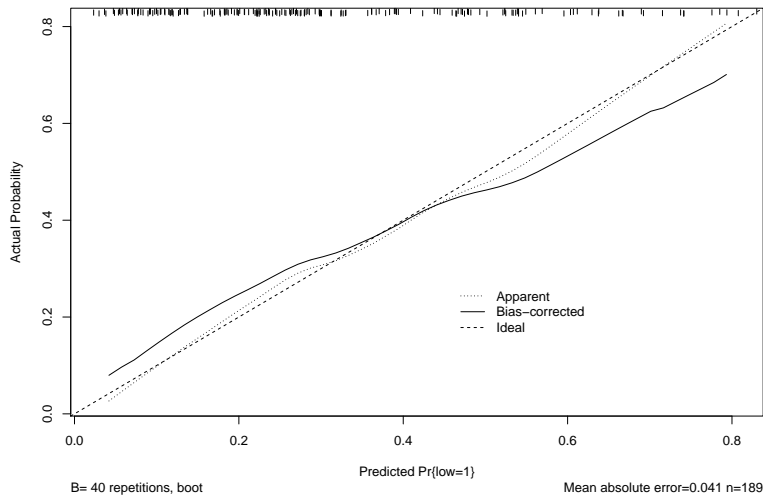
A plot of the model's calibration curve

The `calibrate` function applied to a `lrm` fit provides an assessment of the impact of overfitting on our model.

- The function uses bootstrapping (or cross-validation) to get bias-corrected estimates of predicted vs. observed values based on nonparametric smoothers for logistic regressions.
- In order to obtain this curve, you need to set both `x = TRUE` and `y = TRUE` when fitting the model.
- The errors here refer to the difference between the model predicted values and the corresponding bias-corrected calibrated values.

```
plot(calibrate(model.3))
```

Calibration Curve Plot



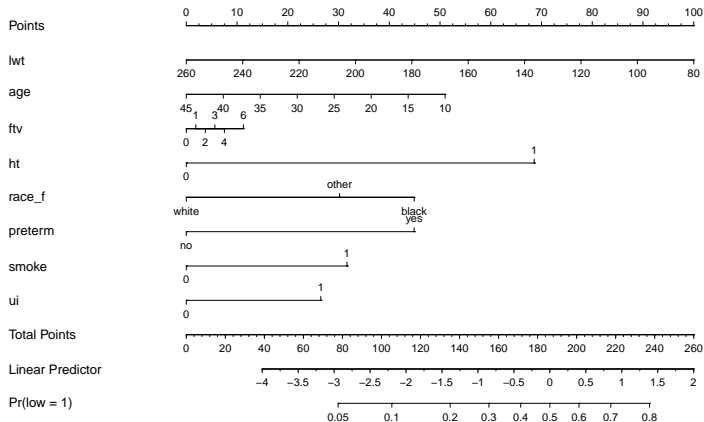
A Nomogram for Model 3

With `lrm`, we can fit a nomogram.

- We use the `plogis` function within a `nomogram` call to get R to produce fitted probabilities (of our outcome, `low`) in this case.

```
plot(nomogram(model.3, fun=plogis,  
             fun.at=c(0.05, seq(0.1, 0.9, by = 0.1), 0.95),  
             funlabel="Pr(low = 1)"))
```

Model 3 Nomogram



Next Up...

Linear Regression using the `ols` function