

STAT 215

Logistic Regression II

Colin Reimer Dawson

Oberlin College

November 14, 2017

Outline

Fitting the Model

Assessment and Testing

- Checking Linearity

- Residuals in Logistic Regression

- Tests and Intervals

- Overall Fit Measures

Binary Logistic Regression

Response variable (Y) is categorical with two categories (i.e., binary).

- Code Y as an indicator variable: 0 or 1
- Assume (for now) a single quantitative predictor, X

Two Equivalent Forms of Logistic Regression

Probability Form

$$\pi = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Logit Form

$$\log\left(\frac{\pi}{1 - \pi}\right) = \beta_0 + \beta_1 X$$

π : **Probability** that $Y = 1$

$\frac{\pi}{1 - \pi}$: **Odds** that $Y = 1$

$\log\left(\frac{\pi}{1 - \pi}\right)$: Log odds, or **logit** that $Y = 1$

Reconstructing Odds Ratio

- The logistic regression output from R gives us $\hat{\beta}_0$ and $\hat{\beta}_1$. But unlike in linear regression, these are not very interpretable on their own.
- We have seen that β_1 corresponds to “rate of change in log odds”. (Slightly) better to convert to “odds ratio” per unit change in X .
- We get this by exponentiating β_1

$$e^{\beta_1} = \text{Multiplicative change in odds that } Y = 1 \\ \text{for a one unit change in } X$$

Choosing $\hat{\beta}_0$ and $\hat{\beta}_1$

Recall that in linear regression, we choose $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize

$$RSS = \sum_i (Y_i - f(X_i))^2 = \sum_i (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X)^2$$

For a logistic model, choose $\hat{\beta}_0$ and $\hat{\beta}_1$ to *maximize the probability of the data according to the model.*

$$\begin{aligned} Pr(\text{Data}|\text{Model}) &= \prod_{i=1}^n \hat{\pi}_i^{Y_i} (1 - \hat{\pi}_i)^{1-Y_i} \\ &= \prod_{i=1}^n \left(\frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X_i}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_i}} \right)^{Y_i} \left(\frac{1}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_i}} \right)^{1-Y_i} \end{aligned}$$

Maximum Likelihood

- $Pr(\text{Data}|\text{Model})$ is called the **likelihood** of the model.
- In fact, when we assume heteroskedastic Normal residuals, the RSS is the negative log likelihood.
- So we've secretly been doing max likelihood this whole time.
- But whereas MLE for Normal-linear model was a calculus problem, MLE for logistic requires an iterative algorithm.

Conditions for Logistic Regression

1. Logit-Linearity (*log odds* depends linearly on X)
2. Independence (no clustering or time/space dependence)
3. Random (data comes from a random sample, or random assignment)
4. ~~Normality~~ no longer applies! (Response is binary, so it can't)
5. ~~Constant Variance~~ no longer required! (In fact, more variance when $\hat{\pi}$ near 0.5)

Checking Linearity

- Can't just transform response via logit to check linearity...
- Unless data is binned... then can take logit of proportion per bin

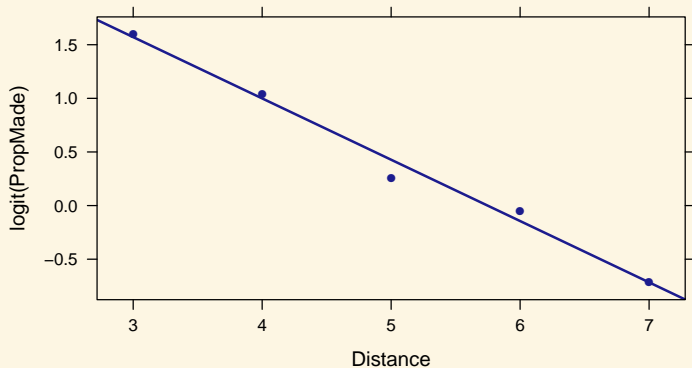
Example: Golf Putts

Distance (ft)	3	4	5	6	7
# Made	84	88	61	61	44
# Missed	17	31	47	64	90
Odds	4.94	2.84	1.30	0.95	0.49
Log Odds	1.60	1.04	0.26	-0.05	-0.71

```
library("mosaic")
Putts <- data.frame(Distance = 3:7, Made = c(84,88,61,61,44),
                    Total = c(101,119,108,125,134))
Putts <- mutate(Putts, PropMade = Made / Total)
```

Binned Data

```
xyplot(logit(PropMade) ~ Distance, data = Putts, type = c("p", "r"))
```



Logits are fairly linear

Equivalent Model Code for Binned Data

```
Putts <- mutate(Putts, Missed = Total - Made)
m2 <- glm(cbind(Made, Missed) ~ Distance, data = Putts, family = "binomial")
m2
```

```
Call: glm(formula = cbind(Made, Missed) ~ Distance, family = "binomial",
          data = Putts)
```

Coefficients:

(Intercept)	Distance
3.2568	-0.5661

Degrees of Freedom: 4 Total (i.e. Null); 3 Residual

Null Deviance: 81.39

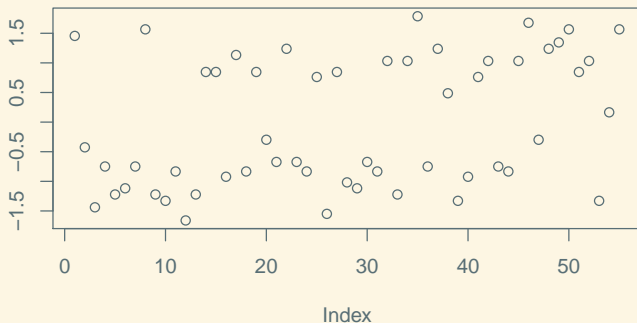
Residual Deviance: 1.069 AIC: 30.18

Deviance Residuals

- Total log likelihood $\ell = \log P(\text{Data} \mid \text{Model})$
- Deviance = -2ℓ measures “total discrepancy” between data and model
- Individual **deviance residual** d_i measures discrepancy for a single point, so that Deviance = $\sum_i d_i^2$

Predicting Med School Acceptance

```
### Model of med school acceptance probability by MCAT score
library("Stat2Data"); data("MedGPA")
medschool.model <-
  glm(Accept ~ MCAT, data = MedGPA, family = "binomial")
residuals(medschool.model, type = "deviance") %>% plot()
```



Deviance Residuals vs. Fitted Values Plot

```
library("arm") ## need to install.packages()
binnedplot(fitted(medschool.model),
           residuals(medschool.model, type = "deviance"),
           nclass = 25)
```

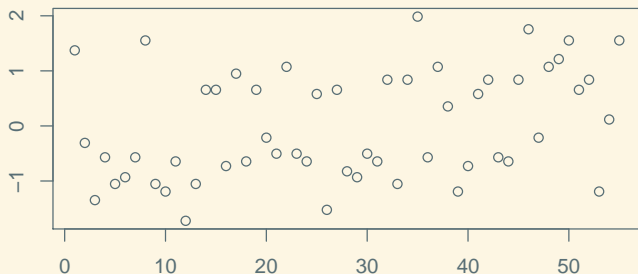


Pearson Residuals

Another way to conceive of residuals is by “standardized distance” from the predicted value

$$\text{Pearson's residual} = \frac{Y_i - \pi_i}{\sqrt{\pi_i(1 - \pi_i)}}$$

```
residuals(medschool.model, type = "pearson") %>% plot()
```



Pearson Residuals vs. Fitted Values Plot

```
library("arm") ## need to install.packages()
binnedplot(fitted(medschool.model),
           residuals(medschool.model, type = "pearson"),
           nclass = 25)
```



Hypothesis Test for β_1

In linear regression, we computed

$$t_{obs} = \frac{\hat{\beta}_1 - 0}{\hat{se}(\hat{\beta}_1)}$$

and found $P\text{-value} = Pr(|T_{n-2}| \geq |t_{obs}|)$

In logistic regression we can use a Normal approximation:

$$z_{obs} = \frac{\hat{\beta}_1 - 0}{\hat{se}(\hat{\beta}_1)}$$

and get $P\text{-value} = Pr(|Z| \geq |z_{obs}|)$

In R

```
data("Election08")
summary(medschool.model)
```

Call:

```
glm(formula = Accept ~ MCAT, family = "binomial", data = MedGPA)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6601	-0.9225	-0.4256	1.0330	1.7878

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	8.71245	3.23645	2.692	0.00710 **
MCAT	-0.24596	0.08938	-2.752	0.00592 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 75.791 on 54 degrees of freedom

Confidence Interval for β_1

Same principle applies for confidence interval...

$$CI(\Delta\text{logit}) : \hat{\beta}_1 \pm z^* \cdot \hat{se}(\hat{\beta}_1)$$

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	8.7124520	3.23645295	2.691975	0.007103017
MCAT	-0.2459643	0.08937837	-2.751944	0.005924264

But... β_1 is the rate of change of the logit, which is hard to understand. More common to report a CI for the odds ratio.

$$CI(OR) : (e^{\beta_1^{(lwr)}}, e^{\beta_1^{(upr)}})$$

Or, in R...

```
confint(medschool.model)
```

		2.5 %	97.5 %
(Intercept)	3.0445836	15.76542012	
MCAT	-0.4412673	-0.08990626	

```
confint(medschool.model) %>% exp()
```

		2.5 %	97.5 %
(Intercept)	21.0012835	7.028052e+06	
MCAT	0.6432208	9.140169e-01	

CI's at specific values

Arguably this is still not very interpretable, so perhaps better to report CI's at a few specific values.

- Script

Logistic Analogs of F -test, R^2 , etc.

- Rather than R^2 , we can use the **residual deviance** to measure *lack* of fit (so, smaller is better)

$$\text{Deviance}(\text{Model}) = -2 \log(\text{likelihood}(\text{Model}))$$

$$\text{Residual Deviance} = \text{Deviance}(\text{Fitted Model})$$

$$\text{Null Deviance} = \text{Deviance}(\text{Null Model})$$

Logistic Analogs of F -test, R^2 , etc.

```
Error in object[[i]]: object of type 'closure' is not subsettable
```


“Analysis of Deviance” Likelihood Ratio Test

Instead of an F -statistic, we can compare two models using the (log) **likelihood ratio**

- Like with R^2 , in-sample likelihood always goes up (deviance goes down) if we add a predictor.
- But if it goes up more than expected by chance, that is evidence the predictor matters.
- $2 \times \log$ of the likelihood ratio = Difference in deviance
- Instead of an F -distribution, this statistic has (for large samples) a χ^2 distribution.

“Analysis of Deviance”: Likelihood Ratio Test

```
anova(medschool.model, test = "LRT")
```

```
Analysis of Deviance Table
```

```
Model: binomial, link: logit
```

```
Response: Accept
```

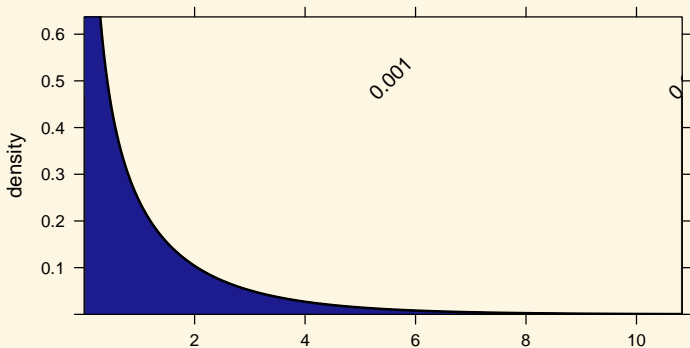
```
Terms added sequentially (first to last)
```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL			54	75.791	
MCAT	1	11.094	53	64.697	0.0008663 ***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
test.stat <- with(medschool.model, null.deviance - deviance)
test.df <- with(medschool.model, df.null - df.residual)
xpchisq(test.stat, df = 1, lower.tail = FALSE)
```



```
[1] 0.0008662948
```