STAT 213
Logistic Regression: Fit, Assess, Test

Colin Reimer Dawson

Oberlin College

April 9-11, 2018

Outline

Outline

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

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

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

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

$$P(\text{Data}|\text{Model}) = \prod_{i=1}^{n} \hat{\pi}_i^{Y_i} (1 - \hat{\pi}_i)^{1-Y_i}$$

$$= \prod_{i=1}^{n} \left(\frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X_i}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_i}} \right)^{Y_i} \left(\frac{1}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_i}} \right)^{1-Y_i}$$

Maximum Likelihood

- $P(\text{Data}|\text{Model})$ is called the **likelihood** of the model.
- In fact, when we assume heteroskedastic Normal residuals, the SSE 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.

Linear vs. Logistic Regression

Goal	Linear	Logistic
Estimate coefs	Minimize SSE	Maximize Likelihood

Outline

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

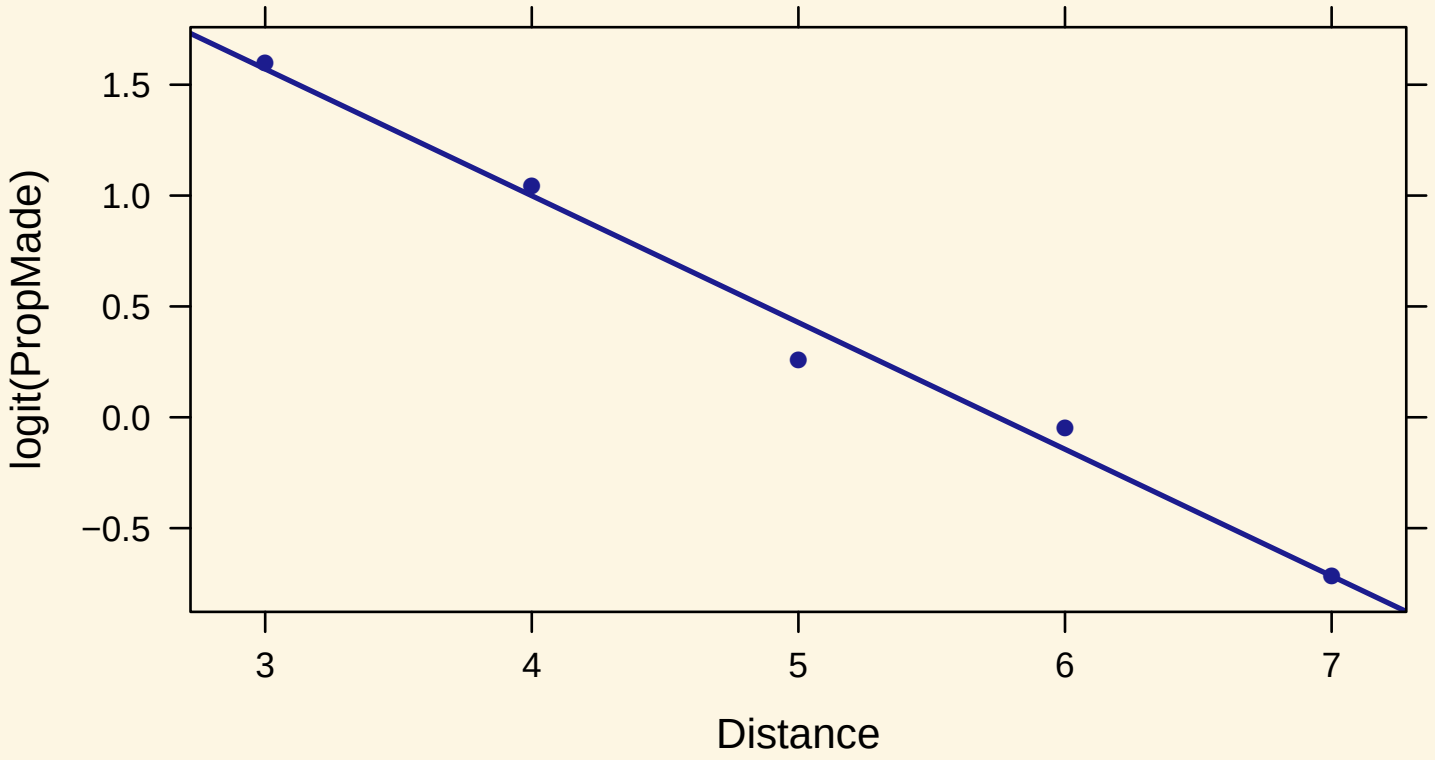
Example: Golf Putts

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
```

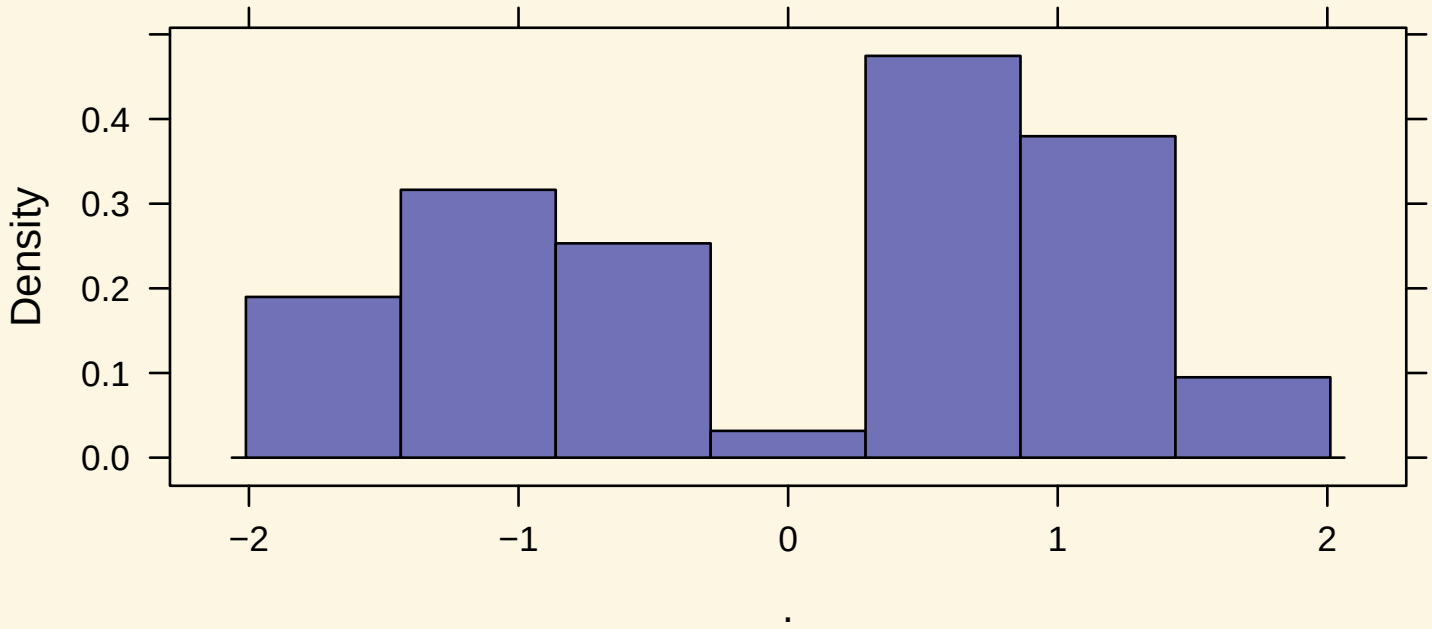
```
Call:  glm(formula = cbind(Made, Missed) ~ Distance, family = "binomial",
    data = Putts)

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 $:= -2\ell$ measures "total discrepancy" between data and model
- Individual **deviance residual** d_i measures discrepancy for a single point, "reverse engineered" so that Deviance $= \sum_i d_i^2$

Checking for Outliers

```
### Model of med school acceptance probability by MCAT score
library("Stat2Data"); data("MedGPA")
medschool.model <-
    glm(Acceptance ~ MCAT, data = MedGPA, family = "binomial")
## Check for outliers by plotting residual distribution
## (Note: will almost always be bimodal; *not* expecting normality)
residuals(medschool.model, type = "deviance") %>% histogram()
```

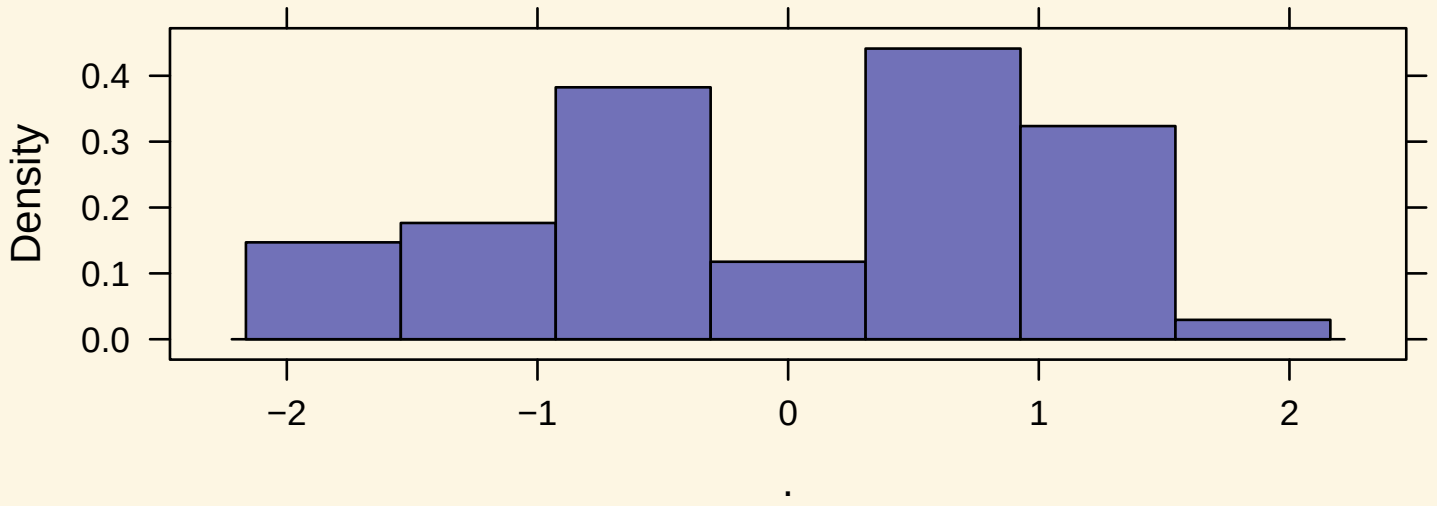


Pearson Residuals

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

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

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

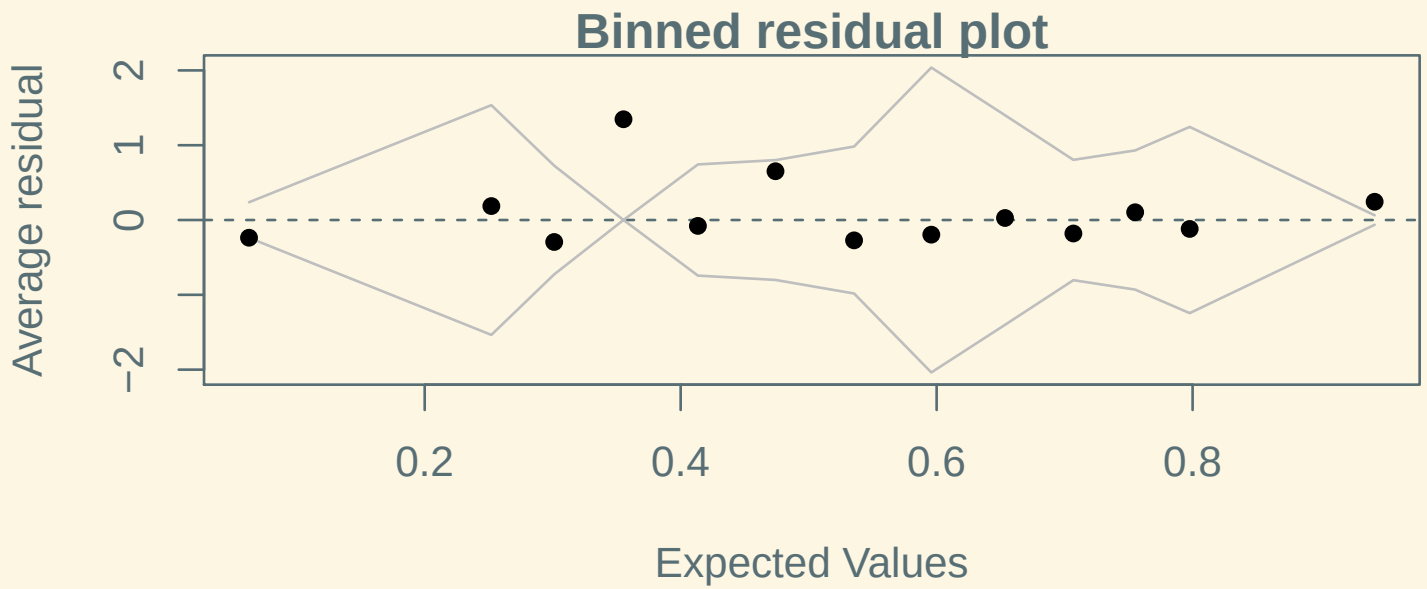


.

Pearson Residuals vs. Fitted Values Plot

Can check logit-linearity for unbinned data by binning residuals and constructing fitted values vs. (average) residuals plot

```
library("arm") ## for binnedplot(); may need to install.packages() first
binnedplot(fitted(medschool.model),
           residuals(medschool.model, type = "pearson"),
           nclass = 25 # number of bins to use
           )
```



Linear vs. Logistic Regression

Goal	**Linear**	**Logistic**
Estimate coefs	Minimize SSE	Maximize Likelihood
Check conditions	*Linearity/Const. var.*: Residual vs. Fitted *Normality*: QQ Plots	*Logit linearity*: Binned residuals vs. fitted

Outline

Hypothesis Test for β_1

In linear regression, we computed the test statistic:

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

(number of standard errors $\hat{\beta}_1$ is from 0).
P-value: prob. of getting a test stat this big by chance if H_0 true (i.e., $\beta_1 = 0$)

In logistic regression we can do the same thing, but with Normal instead of t distribution.

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

and get P-value: prob of a test stat this big if H_0 true

In R

```
data("Election08")
summary(medschool.model) %>% coef() %>% round(3)

                Estimate Std. Error z value Pr(>|z|)
    (Intercept)   -8.712      3.236  -2.692    0.007
    MCAT           0.246      0.089   2.752    0.006
```

Only 0.6% chance we'd get $\left|\hat{\beta}_1\right| \geq 0.246$ if the association is an illusion (due solely to noise in the data)

Linear vs. Logistic Regression

Goal	**Linear**	**Logistic**
Estimate coefs	Minimize SSE	Maximize Likelihood
Check conditions	*Linearity/Const. var.*: Residual vs. Fitted *Normality*: QQ Plots	*Logit linearity*: Binned residuals vs. fitted
Test coefs	Measure SEs from 0, P-value using t	Measure SEs from 0 P-value using Normal

Confidence Interval for β_1

Same principle applies for confidence interval...

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

```
confint(medschool.model) %>% round(2)

                2.5 % 97.5 %
(Intercept) -15.77  -3.04
MCAT          0.09   0.44
```

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

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

In R...

```
confint(medschool.model) %>% round(2)
```

```
                2.5 % 97.5 %
(Intercept) -15.77  -3.04
MCAT          0.09   0.44
```

```
confint(medschool.model) %>% exp() %>% round(2)
```

```
                2.5 % 97.5 %
(Intercept)  0.00   0.05
MCAT         1.09   1.55
```

"We are 95% confident that the *odds* (not probability) of admittance increases by a *factor* of (is *multiplied* by) between 1.09 and 1.55 for each additional point of MCAT score"

Linear vs. Logistic Regression

Goal	**Linear**	**Logistic**
Estimate coefs	Minimize SSE	Maximize Likelihood
Check conditions	*Linearity/Const. var.*: Residual vs. Fitted *Normality*: QQ Plots	*Logit linearity*: Binned residuals vs. fitted
Test coefs	Measure SEs from 0, P-value using t	Measure SEs from 0 P-value using Normal
Intervals for Params	Slope: β_1	Odds Ratio: e^{β_1}

CIs at specific values

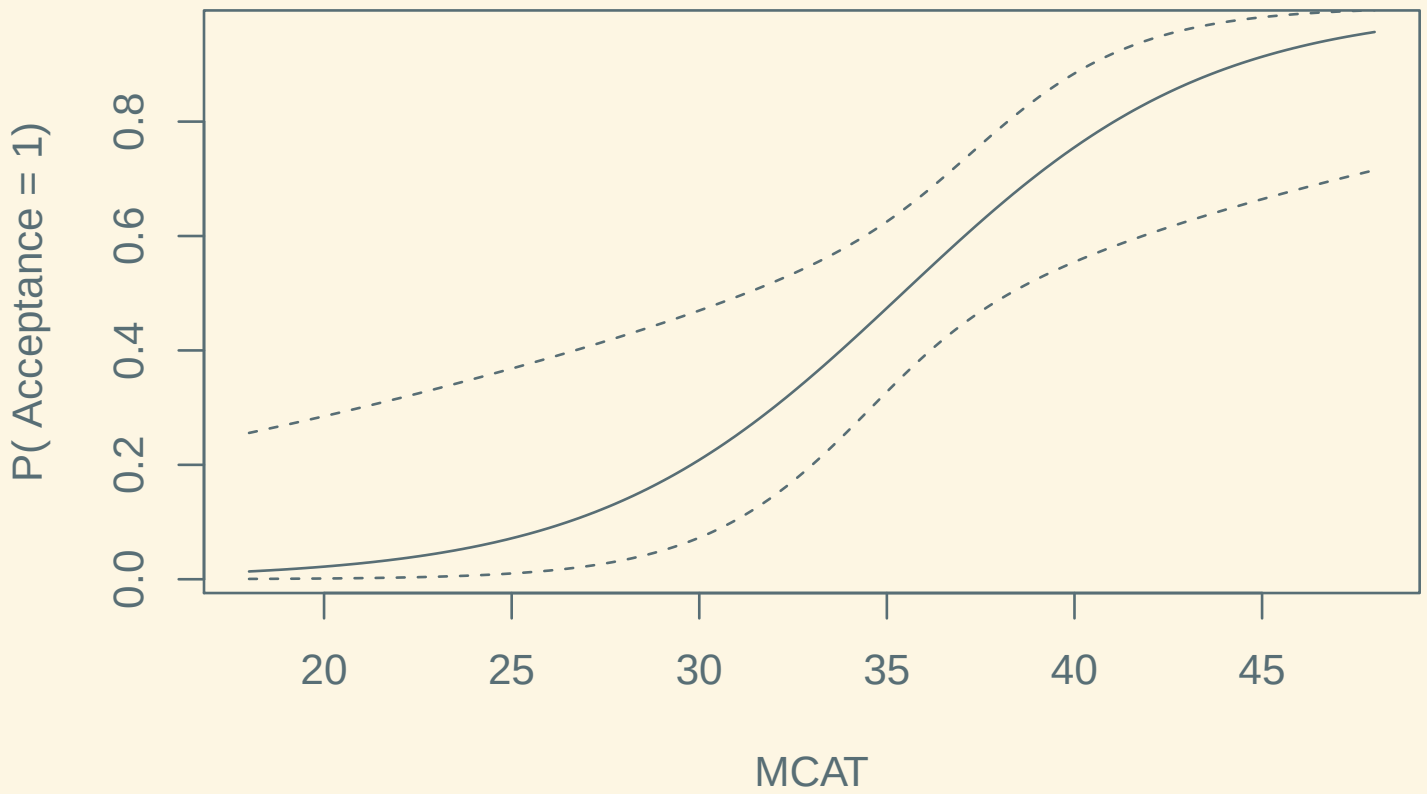
Arguably this is still not easy to interpret, so perhaps better to report CIs at a few specific values.

```
source("http://colindawson.net/stat213/code/helper_functions.R")
## functions made with regular makeFun() give point values but not
## intervals with logistic models, so I wrote a custom function
f.hat <- makeFun.logistic(medschool.model)
quartiles <- quantile(~MCAT, data = MedGPA)
f.hat(MCAT = quartiles, interval = "confidence", level = 0.95) %>% round(2)
```

```
      MCAT pi.hat  lwr  upr
0%      18   0.01 0.00 0.26
25%     34   0.41 0.26 0.58
50%     36   0.54 0.39 0.67
75%     39   0.71 0.52 0.84
100%    48   0.96 0.72 0.99
```

Confidence Bands

```
## Requires sourcing helper_functions.R
## Can supply level=, xlim=, xlab= and ylab= to customize graph
plot.logistic.bands(medschool.model)
```



Linear vs. Logistic Regression

Goal	**Linear**	**Logistic**
Estimate coefs	Minimize SSE	Maximize Likelihood
Check conditions	*Linearity/Const. var.*: Residual vs. Fitted *Normality*: QQ Plots	*Logit linearity*: Binned residuals vs. fitted
Test coefs	Measure SEs from 0, P-value using t	Measure SEs from 0 P-value using Normal
Intervals for Params	Slope: β_1	Odds Ratio: e^{β_1}
Intervals for Fitted Vals.	Confidence and prediction intervals	Confidence intervals only

Outline

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.

```
Call:
glm(formula = Acceptance ~ MCAT, family = "binomial", data = MedGPA)

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

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
Residual deviance: 64.697  on 53  degrees of freedom
AIC: 68.697

Number of Fisher Scoring iterations: 4
```

Linear vs. Logistic Regression

Goal	**Linear**	**Logistic**
Estimate coefs	Minimize SSE	Maximize Likelihood
Check conditions	*Linearity/Const. var.*: Residual vs. Fitted *Normality*: QQ Plots	*Logit linearity*: Binned residuals vs. fitted
Test coefs	Measure SEs from 0, P-value using t	Measure SEs from 0 P-value using Normal
Intervals for Params	Slope: β_1	Odds Ratio: e^{β_1}
Intervals for Fitted Vals.	Confidence and prediction intervals	Confidence intervals only
Measure Fit	R^2 ($\uparrow$ better)	Deviance ($\downarrow$ better)

"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: Acceptance

     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
```

Linear vs. Logistic Regression

Goal	**Linear**	**Logistic**
Estimate coefs	Minimize SSE	Maximize Likelihood
Check conditions	*Linearity/Const. var.*: Residual vs. Fitted *Normality*: QQ Plots	*Logit linearity*: Binned residuals vs. fitted
Test coefs	Measure SEs from 0, P-value using t	Measure SEs from 0 P-value using Normal
Intervals for Params	Slope: β_1	Odds Ratio: e^{β_1}
Intervals for Fitted Vals.	Confidence and prediction intervals	Confidence intervals only
Measure Fit	R^2 ($\uparrow$ better)	Deviance ($\downarrow$ better)
Nested test	F-test	Likelihood Ratio Test