# Class 27 R Logistic Regression: Odds Ratio And Inferences

library(Stat2Data)  
  
data("Putts1")  
head(Putts1)

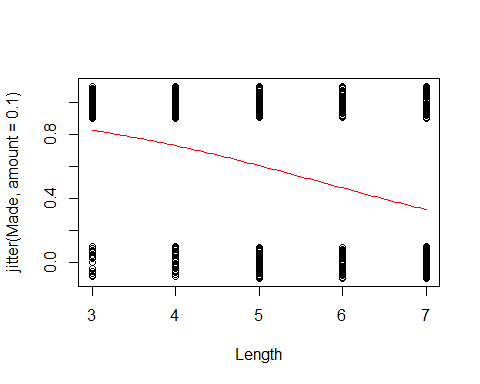
## Length Made  
## 1 3 1  
## 2 3 1  
## 3 3 1  
## 4 3 1  
## 5 3 1  
## 6 3 1

**Logistic Regression for Putting**

# Use glm for different types of graphs  
modPutt=glm(Made~Length,family=binomial,data=Putts1)  
summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

B0 = summary(modPutt)$coef[1]  
B1 = summary(modPutt)$coef[2]  
  
plot(jitter(Made,amount=0.1)~Length,data=Putts1)  
curve(exp(B0+B1\*x)/(1+exp(B0+B1\*x)),add=TRUE, col="red")



**Golf Putts Probabilities** 𝜋̂=𝑒(3.257−0.5661𝐿𝑒𝑛𝑔𝑡ℎ)/(1+𝑒(3.257−0.5661𝐿𝑒𝑛𝑔𝑡ℎ) ) 𝑝̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑡𝑟𝑖𝑎𝑙𝑠) - THis is also a part of Class 26

# This makes a table so we can then make the proportion of success for the golf putts probabilities   
  
Putts.table = table(Putts1$Made, Putts1$Length)  
Putts.table

##   
## 3 4 5 6 7  
## 0 17 31 47 64 90  
## 1 84 88 61 61 44

# Proportion made for each distance   
# takes the probabilities from the putts table and see the proportion for each distance so it's the total for distance 3  
# P(success)/Ntrials  
p.hat = as.vector(Putts.table[2,]/colSums(Putts.table))  
p.hat

## [1] 0.8316832 0.7394958 0.5648148 0.4880000 0.3283582

logit = function(B0, B1, x)  
 {  
 exp(B0+B1\*x)/(1+exp(B0+B1\*x))  
 }

pi.hat=0  
  
for(i in 3:7)  
 {  
 pi.hat[i-2] = logit(B0, B1, i)  
 }  
  
pi.hat

## [1] 0.8261256 0.7295364 0.6049492 0.4650541 0.3304493

Putts = data.frame(  
 "Length" = c(3:7),   
 "p.hat" = p.hat,   
 "pi.hat" = pi.hat)  
  
head(Putts)

## Length p.hat pi.hat  
## 1 3 0.8316832 0.8261256  
## 2 4 0.7394958 0.7295364  
## 3 5 0.5648148 0.6049492  
## 4 6 0.4880000 0.4650541  
## 5 7 0.3283582 0.3304493

he above is all review from last class that we didn’t get to p-hat = the 3:7, length is 3:7; this sis from teh putts data,

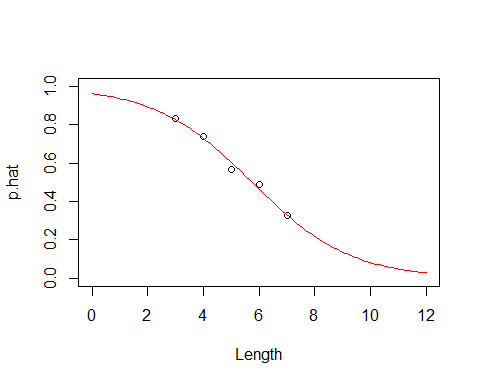
the below plots the curve ontop of it

logit, logs the thing, i think; see th formula above code

if we change x limits, it shows the smaller vs bigger graph

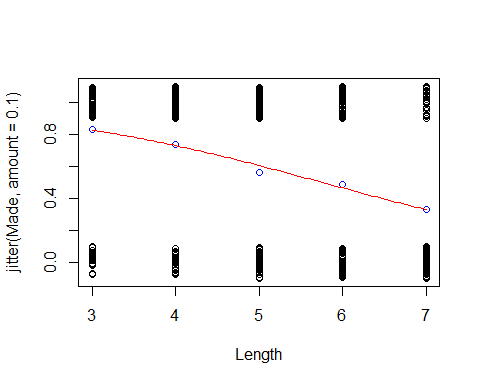
**Probability Form of Putting Model**

plot(p.hat~Length,ylim=c(0,1), xlim=c(0,12), data=Putts)  
# Shows the proportion we are predicting to the prediction plot that we have   
curve(logit(B0, B1, x),add=TRUE, col="red")

 Putts1 = a different dataset; we want to putthe same points on this differently

the blue points are the mean values for each value of x

plot(jitter(Made,amount=0.1)~Length,data=Putts1)  
  
points(p.hat~Length, data=Putts, col='blue')  
  
curve(logit(B0, B1, x),add=TRUE, col="red")

 THink about the Odds of something happening; - the odds vs probability - the odds against are 4:1; they expect those one horse to lose 4/5 vs 1/5 of the time

if pi = proportion of yes (success 1, etc)

the odds of yes = P(pi)/(P(1-pi)) odds of yes = P(yes)/P(no)

logit form = log(odds of yes) = B0 + B1X Below adds 2 new columns to teh dataframe, this messes with teh data

one is probability of it happened over teh probaility that it doesnt haveppen

pi = predicted

**Odds** The odds against a certain horse winning a race are 4 to 1.  
- What does that mean? – 4 losses for every 1 win – P(Win) = 1/5 – P(Loss) = 4/5

𝑂𝑑𝑑𝑠= (𝑃(𝑊𝑖𝑛))/(𝑃(𝐿𝑜𝑠𝑠))=(1/5)/(4/5)=1/4

**Odds** If pi = proportion of “yes” (success, 1, ….) the odds of yes are(is)

(𝑃(𝑦𝑒𝑠))/(𝑃(𝑛𝑜))=𝜋/(1−𝜋)

With a little bit of algebra… 𝑜𝑑𝑑𝑠=𝜋/(1−𝜋)⇔𝜋=𝑜𝑑𝑑𝑠/(1+𝑜𝑑𝑑𝑠)

**Odds and Logistic Regression** Logit form: log⁡(𝜋/(1−𝜋))=𝛽\_𝑜+𝛽\_1 𝑋 -The logistic model assumes a linear relationship between the predictor and log(odds). - log⁡( 𝑜𝑑𝑑𝑠)=𝛽\_𝑜+𝛽\_1 𝑋

**Logit Form of Putting Model**

**Back to Putting Data** Since we have lots of putts, we can estimate 𝑝̂ (proportion of putts made) at each length 𝑝̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑡𝑟𝑖𝑎𝑙𝑠) and the odds (𝑜𝑑𝑑𝑠)̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑚𝑖𝑠𝑠𝑒𝑑)=𝑝̂/(1−𝑝̂ ) and find log⁡((𝑜𝑑𝑑𝑠̂) at each length.

**Golf Putts Odds** (𝑜𝑑𝑑𝑠)̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑚𝑖𝑠𝑠𝑒𝑑)=𝑝̂/(1−𝑝̂ ) (from sample) (𝑜𝑑𝑑𝑠)̂=𝜋̂/(1−𝜋̂ ) (from logistic regression) Length: 3,4,5,6,7 oddshat(from sample): 4.94,2.84,1.30,0.95,0.49 oddshat(from regression): 4.75,2.70,1.53,0.87,0.49

Putts$p.Odds = Putts$p.hat/(1-Putts$p.hat)  
Putts$pi.Odds = Putts$pi.hat/(1-Putts$pi.hat)  
  
head(Putts)

## Length p.hat pi.hat p.Odds pi.Odds  
## 1 3 0.8316832 0.8261256 4.9411765 4.751277  
## 2 4 0.7394958 0.7295364 2.8387097 2.697355  
## 3 5 0.5648148 0.6049492 1.2978723 1.531320  
## 4 6 0.4880000 0.4650541 0.9531250 0.869348  
## 5 7 0.3283582 0.3304493 0.4888889 0.493539

**Plot for Putts Data** Plot log⁡((𝑜𝑑𝑑𝑠̂) versus Length (3, 4, 5, 6, 7) Add a line with intercept and slope from the logistic model.

The below code does something

the line tells you how the probaility chanes as teh rate of other things change.

so we need to think fo teh odds ratio, a common way to compare two groiups is to look at a ratio of their odds

odds ratio (OR) = Odd.R = Odd1/Odd2

odds using data from 4 ft = 2.84 odds using data from 3 feet = 4.94

odds ratio ( 4 to 3) = 2.84/4.94 = 0.57

the odds of making a putt from 4 feet are 57% of the odds of making from 3 feet

*Interpreting Slope Using Odds Ratio*

log(Odds) = B0+B1X -> odds = e^(B0+B1\*X)

*CI for Slope and ODds Ratio* - Using teh SE for the slope, find a CI for B1 with:

B-hat1 +/- z-star \* SE

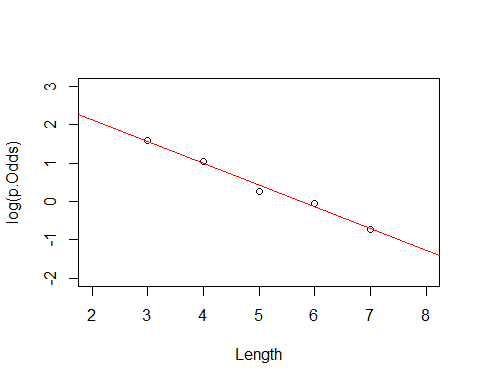
the above will get you theCI for teh odds ratio (E^B1) exponentiate the CI for B1

ex:

CI for slope = (0.5, 0.6) CI for OR = e^0.5, e^0.6 = (0.497, 0.648)

**Logit Form of Putting Model**

plot(log(p.Odds)~Length, data=Putts, xlim=c(2,8), ylim=c(-2,3))  
abline(B0, B1, col="red")



**Odds Ratio** A common way to compare two groups is to look at the ratio of their odds “Odds Ratio”=” OR “=(”Odd” “s” \_1)/(“Odd” “s” \_2 )

**Putting Data** Odds using data from 4 feet = 2.84 Odds using data from 3 feet = 4.94 Odds ratio (4 ft to 3 ft) = 2.84/4.94=0.57 The odds of making a putt from 4 feet are 57% of the odds of making from 3 feet.

**Odds Ratios for Putts** From fitted logistic: Length: 3,4,5,6,7, pihat: 0.826,0.730,0.605,0.465,0.331 odds hat: 4.75,2.70,1.53,0.84,0.49

To Odds Ratio Length: 4-3 ft, 5-4 ft, 6-5 ft, 7-6 ft Odds Ratio: 0.57, 0.57, 0.57, 0.57

In a logistic model, the odds ratio when changing the predictor by one is constant.

**Odds Ratios for Putts** From samples at each distance: Length: 3,4,5,6,7 phat: 0.832, 0.739, 0.565, 0.488, 0.328 oddshat: 4.94, 2.84, 1.30, 0.95, 0.49

To Odds Ratio: Length: 4-3ft, 5-4 ft, 6-5ft, 7-6ft Odds Ratio: 0.57, 0.46, 0.73, 0.51

**Interpreting “Slope” using Odds Ratio** log⁡(𝑜𝑑𝑑𝑠)=𝛽\_0+𝛽\_1 𝑥goes to 𝑜𝑑𝑑𝑠=𝑒^(𝛽\_0+𝛽\_1 𝑥) What happens when we increase x by one? 𝑒^(𝛽\_0+𝛽\_1 (𝑥+1))=𝑒^(𝛽\_0+𝛽\_1 𝑥)∙𝑒^(𝛽\_1 ) When we increase x by one, the odds increase/decrease by a factor of 𝑒^(𝛽\_1 ) (odds ratio). For putts: The odds of making a putt decrease by a factor of 0.57 (𝑒^(−0.566)) for every extra foot of length.

**CI for Slope and Odds Ratio** Using the SE for the slope, find a CI for β1 with 𝛽̂\_1±𝑧^∗∙𝑆𝐸 <- this is just the formula for confidence intervals To get CI for the odds ratio (𝑒^(𝛽\_1 )) exponentiate the CI for β1

CI for slope: −0.566±1.96(0.06747) =(−0.698,−0.434)

CI for OR: 〖(𝑒〗(−0.698),𝑒(−0.438)) =(0.497, 0.648)

SE\_B1 = summary(modPutt)$coef[2,2]  
exp(B1 - SE\_B1\*qnorm(0.975))

## [1] 0.4973894

exp(B1 + SE\_B1\*qnorm(0.975))

## [1] 0.6479761

in practice we are not going to use teh confint.default function; because the default forces the thing to use z scores; and teh not by default uses some lo glikelihoods to get this thing

**Similar tests/measures for logistic regression?** Recall: “Ordinary” Regression lm(formula = Active ~ Rest)

Coefficients: Estimate Std. Error t value Pr(>|t|)  
(Intercept) 8.75340 5.60773 1.561 0.12  
Rest 1.18387 0.08214 14.413 <2e-16 \*\*Rest, above, tests for individual coefficients\*

*For the first three italics, there are for comparing the models* Residual standard error: *14.39* on 310 degrees of freedom Multiple R-squared: *0.4012*, Adjusted R-squared: *0.3993* F-statistic: *207.7* on 1 and 310 DF, p-value: *< 2.2e-16* <- test overall fit

**Test for Individual Coefficients** Ho: Bi = 0 Ha: Bi != 0

t.s. = Bhat/SEofBhat (R will give you all of these variables) Interpret as with individual t-tests in ordinary regression P-value = 2P( Z > |t.s.| )

Estimate Std. Error z value Pr(>|z|)

(Intercept) 3.25684 0.36893 8.828 <2e-16  ***Length -0.56614 0.06747 -8.391 <2e-16***

**Estimating Parameters in Ordinary Regression** Coefficients are chosen to minimize the sum of the squared errors in the observed sample. (Least Squares Estimation) 𝑆𝑆𝐸=〖Σ(𝑦−𝑦̂)〗^2 **WE WANT A SMALL SSE**

**Test for Overall Fit** Ho: B1 = 0 -> log(odds) = Bo Ha: B1 != 0 -> log(Odds) = Bo + B1X

How much “better” does the linear model do than one with a constant? Is it “significantly” better?

**Maximizing the Likelihood of the Sample** - Suppose that there are three decks of cards: 1. Standard 52 card deck 2. Euchre deck (9, 10, J, Q, K, A) 3. Deck with all red cards If two cards were drawn from a deck (without replacement), a Jack of Hearts, then a King of Hearts, from which deck do you think that there were chosen?

* Suppose that there are three decks of cards:

1. Standard 52 card deck; (1/52)(1/51)≈“0.000377”
2. Euchre deck (9, 10, J, Q, K, A); (1/24)(1/23)≈“0.001812”
3. Deck with all red cards; (1/26)(1/25)≈“0.001538”

**Estimating Parameters in Logistic Regression** Parameters are chosen to maximize the likelihood of the observed sample. (Maximum Likelihood Estimation) If the ith data point is YES (yi=1), calculate 𝜋̂\_𝑖 If the ith data point is NO (yi=0), calculate 1−𝜋̂\_𝑖

Likelihood:𝐿=∏〖𝜋̂\_𝑖〗^(𝑦\_𝑖 ) (1−𝜋̂\_𝑖 )^(1−𝑦\_𝑖 ) **WE WANT A HIGH LIKELIHOOD**

**Test for Overall Fit** Length: 3,4,5,6,7, Made: 84,88,61,61,44 Missed: 17,31,47,64,90 Ratio: 0.826, 0.730, 0.605, 0.465, 0.330

𝐿=∏〖𝜋̂\_𝑖〗^(𝑦\_𝑖 ) (1−𝜋̂\_𝑖 )^(1−𝑦\_𝑖 ) Ho: B1 = 0 -> log(odds) = Bo L = .576^338\*(1-.576)^249

Ha: B1 != 0 -> log(Odds) = Bo + B1X L = (0.826^84 \* 0.174^17) \* (0.730^88 \* 0.270^31) \* (0.605^61 \* 0.395^47) \* (0.465^61 \* 0.535^64) \* (0.330^44 \* 0.670^90)

exp(confint.default(modPutt))

## 2.5 % 97.5 %  
## (Intercept) 12.6006177 53.5133410  
## Length 0.4973894 0.6479761

exp(confint(modPutt))

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 12.7974573 54.4505172  
## Length 0.4960611 0.6464444

just keep in mind the units for the CI

Similar tests/measures for logistic regression

recall: “Ordinary” regression

*Test for idividual coeff*

Ho: Bi = 0  
Ha: Bi -/= 0

t.s = B-hati/SE(B-hati)

R WIll give you all of these numbers

interpret as with individual t tests in ordinary regression

p-value = 2P(Z>abs(t.s))

*Estimating Parameters in ORd Regression*

Coeff are chosen to min the sum of the squared errors in teh observed sample (LEast Squares Estimation

SEE = sum(y-y-hat)^2

We want a small SSE

*Test for Overall Fit* H0: B1 = 0 Ha: B1 =/= 0 log(odds) = B0 log(odds) = B0 + B1X; these are competing models

how much better does the lienar mdoel do than one with a constatst? IS sit sig better?

*Estimating Parameters in Logistic Regression* Parameters are chosen to max the likelihood of the observed sample (MAx likelihood estimation)

If teh ith data poin is YES (yi = 1), calc pi-hati

If teh ith data point is No (yi = 0), calc 1-pi-hati

We want L to be big

THis is where the table(Putts1$MAde) starts

summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

**−2 ln⁡(𝐿) for Constant (H0) Model** For a constant model:  
𝐿\_0=𝜋̂^(#𝑦𝑒𝑠) 〖(1−𝜋̂)〗^(𝑛−#𝑦𝑒𝑠) log⁡(𝐿\_0 )=#𝑦𝑒𝑠∙log⁡𝜋̂ )+#𝑛𝑜∙log-pihat

Combining all putts: 338 made out of 587 𝜋hat =338/587=0.5758 𝐿\_0=〖0.5758〗^338 〖0.4242〗^249

log⁡(𝐿\_0 )=338 log⁡(0.576)+249 log⁡(0.424)=−400.1 〖−2log〗⁡(𝐿\_0 )=800.2

**Putts1: Made~Length** lmodPutt=glm(Made~Length,family=binomial,data=Putts1) summary(lmodPutt)

**Example: Golf Putts** 𝐿=∏〖𝜋̂\_𝑖〗^(𝑦\_𝑖 ) (1−𝜋̂\_𝑖 )^(1−𝑦\_𝑖 ) 𝐿=〖0.826〗^84 〖0.174〗^17 〖0.730〗^88 〖0.270〗31⋯〖0.330〗44 〖0.670〗^90 log⁡(𝐿)=84 log⁡(0.826)+17 log⁡(0.174)+⋯ +44 log⁡(0.330)+90 log⁡(0.670)=−359.9 Coefficients are chosen to get 𝑙𝑜𝑔(𝐿) as big as possible 〖−2log〗⁡(𝐿)=718. 8 <- Minimize residual deviance

* How much “improvement” with the predictor?
* Compare the null deviance with the residual deviance; subtract the two to get your Gstatistic

lmodPutt=glm(Made~Length,data=Putts1,family=binomial,data=Putts1) summary(lmodPutt)

Coefficients: Estimate Std. Error z value Pr(>|z|) (Intercept) 3.25684 0.36893 8.828 <2e-16 Length -0.56614 0.06747 -8.391 <2e-16

Null deviance: 800.21 on 586 degrees of freedom

Residual deviance: 719.89 on 585 degrees of freedom

−2 l𝑜𝑔⁡(𝐿\_0 )−(−2 log⁡(𝐿) )=800.2−719.99=80.3 This difference is called the G statistic.

**Evaluating Overall Fit** Test for overall fit (Similar to regression ANOVA) t.s. = G = improvement in –2log(L) over a model with just a constant term Compare to y2 with k d.f. (chi-square) - k = number of predictiors

The null sys tat it doens’t matter how far we are from teh hole, while teh laternative says that it does matter

Bo = 0 Bo =/= 0

table(Putts1$Made)

##   
## 0 1   
## 249 338

338/(338+249)

## [1] 0.5758092

L.null = (.576)^338\*(1-.576)^249  
L.null

## [1] 1.725431e-174

-2\*log(L.null)

## [1] 800.2087

if the distance matters, then teh difference lengts =will ahev different values

we first calc how you got the sample from 3 ft putts

based on data, we made 0.73 putts at 3ft, then the probabiltiy of making it

the log(L) below is a little bigger than thte above L.null, which means taht we like the second L better

L = 0.826^84\*0.174^17\*0.730^88\*0.270^31\*.605^61\*.395^47\*.465^61\*.535^64\*0.330^44\*0.670^90  
L

## [1] 4.765502e-157

-2\*log(L)

## [1] 719.8889

do things with chi-squared; it lieks chi squared, so we like log?

we cna look at this like a chi squared

how likeily is it to get this on a chi squared distribution

summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

1-pchisq(80.3,1)

## [1] 0

summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

G = summary(modPutt)$null.deviance - summary(modPutt)$deviance  
  
1 - pchisq(G,1)

## [1] 0

the below gives you how likly we would see this by chacne; we if small p value; then we can reject Ho

**Evaluating Overall Fit** Ho: Bi = 0 Ha: Bi != 0

log⁡(𝜋/(1−𝜋))=𝛽\_𝑜+𝛽\_1 𝑋

anova(modPutt, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Made  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 586 800.21   
## Length 1 80.317 585 719.89 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1