Lesson 8

Clark

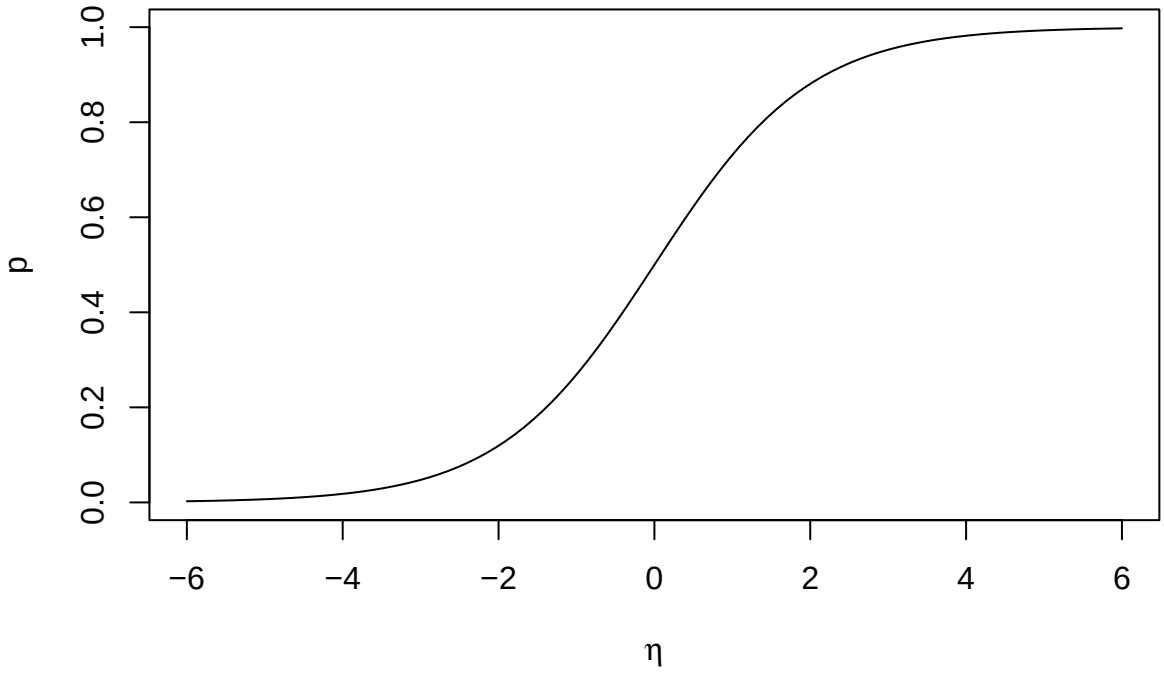
Over the last several lessons we have been talking in general terms about Generalized Linear Models. Today, we're going to start talking about specific cases. We will start with one that you've likely seen before. When our data are binary (or are counts of the number of successes out of n trials) we typically assume that our data have the following distribution:

Although we typically refer to this as logistic regression, we will reserve that for when we are using a logit link (which is somethings called the logistic equation).

In general, if our data are binary, we can deconstruct our binomial distribution and note that our link function is mapping η_i to p_i.

The canonical link function, as we saw above, is the logit link, which is sometimes called the log-odds.

```
library(tidyverse)
library(faraway)
curve(ilogit(x),-6,6, xlab=expression(eta),ylab="p")
```



Assuming we are using this link, when we fit a GLM we are constructing a linear predictor mapping our covariates to:

So, this model yields that the $\frac{\partial \eta_i}{\partial \beta_j} = x_{ij}$. Meaning, the change in x_{ij} creates a change in the *log odds* of p_i, or the change in one unit of $x_{i,j}$ creates a change in the *odds* by a multiple of e^{β_j}. So, if our odds were 3 and $\beta_1 = 0.1$, when we change $x_{i,1}$ by one unit (and hold everything else constant), we change our odds to:

```
3*exp(.1)
```

```
## [1] 3.315513
```

If $\beta_1 = -0.1$ we change our odds to:

```
3*exp(-.1)
```

```
## [1] 2.714512
```

Sometimes people are tempted to use a linear link funciton for a binary response. That is, they assume:

What might be the issues with doing this?

So, it's not that you *can't* do it, it's just that we don't have to and it doesn't make sense in a lot of cases to say that we are absolutely positive that $P(Z_i) = 1$

Let's go through an example that Faraway uses. Here we are looking at possible factors that impact whether or not men aged 39-59 in San Francisco get heart disease.

```
data(wcgs)
wcgs %>% group_by(chd)%>%
  summarize(counts = n(),avg_age=mean(age),
            avg_weight=mean(weight),
            avg_cits=mean(cigs))
```

```
## # A tibble: 2 x 5
##   chd   counts avg_age avg_weight avg_cits
##   <fct>  <int>   <dbl>      <dbl>    <dbl>
## 1 no      2897    46.1       170.     11.2
## 2 yes      257    48.5       174.     16.7
```

From our exploratory analysis what do we see?

```
wcgs %>%ggplot(aes(x=height,y=cigs))+
  geom_point(alpha=0.2,position=position_jitter())+
  facet_grid(~chd)
```

What are we looking at here?

Consider a model that includes weight and cigarette count. How would we write this model using mathematical notation?

```
one_glm <- glm(chd~cigs+weight,family=binomial(link="logit"),
               data=wcgs)
```

If I want to determine if the model is a good fit I could compute the deviance. For a logistic regression with a logit link our formula for deviance is:

We can compare our model to the null model doing:

```r
null_glm <- glm(chd~1,family=binomial(link="logit"),
                data=wcgs)

deviance(null_glm) - deviance(one_glm)
```

```
## [1] 46.63251
```

We know this has a χ^2 distribution with $n-2$ degrees of freedom.

```r
 anova(null_glm,one_glm,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: chd ~ 1
## Model 2: chd ~ cigs + weight
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      3153     1781.2
## 2      3151     1734.6  2   46.633 7.48e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can compare our model to the saturated model by looking at the likelihood ratio test which IS the deviance

```r
1-pchisq(deviance(one_glm),nrow(wcgs)-
           length(one_glm$coefficients))
```

```
## [1] 1
```

However, there's an issue here. As noted on pg 181 of Agresti, Unless we have grouped data, meaning we are looking at the number of successes out of a fixed number of trials, we CANNOT use deviance to test for goodness of fit. In this case, the chi-squared limit does not occur and the statistic can actually be uniformative about lack of fit.

To combat this, we can artificially create groups and compute what's called the Hosmer Lemeshow statistic.

```r
#install.packages("glmtoolbox")
library(glmtoolbox)
hltest(one_glm)
```

```
##
##    The Hosmer-Lemeshow goodness-of-fit test
##
##  Group Size Observed  Expected
##      1  291       10 12.802831
```

```
##      2  315       11 16.236220
##      3  329       17 18.696036
##      4  303       23 18.706485
##      5  309       22 20.758820
##      6  311       20 23.014986
##      7  315       30 26.089637
##      8  318       36 30.164565
##      9  320       39 35.949975
##     10  315       45 47.464690
##     11   28        4  7.115755
##
##          Statistic =  8.29936
## degrees of freedom =  9
##            p-value =  0.50428
```

Here since the p-value is large we have no evidence of lack of fit.

To compute confidence intervals for β we can compute the Wald intervals $\hat{\beta} \pm z^{\alpha/2} se(\hat{\beta}$, however the Wald intervals are typically not the preferred method for confidence intervals (although annoyingly this is what the `summary` output for `glm` in R give us).

Instead, we should compute what are called the profile likelihood-based confidence intervals which are based on the likelihood ratio test

```
confint(one_glm)
```

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

```
##                    2.5 %      97.5 %
## (Intercept) -5.80034440 -3.74407727
## cigs         0.01631961  0.03213002
## weight       0.00591983  0.01745093
```

Note that these still are confidence intervals for β which do NOT give us intervals for p_i. If instead you want a confidence interval for p_i you would need to transform the bounds of the interval. Note that this is NOT the best (or even the most correct) way to do it. A better way to do this is using the Delta Method that you should have covered in MA476 that gives the distribution for a function of the MLEs, but I'm *ok* if in this course you just back transform the end points of the confidence interval.

That is, if we have a simple logistic regression model we can create a 95% CI for p_i by computing:

What would happen if we tried to fit an identity link?

```
lin_glm <- glm(chd~cigs+weight,family=binomial(link="identity"),
               data=wcgs)
```

What do you think is going on?

What would be wrong with:

```
lin2_lm <- lm((as.numeric(chd)-1)~cigs+weight,
              data=wcgs)
```

https://statmodeling.stat.columbia.edu/2020/01/10/linear-or-logistic-regression-with-binary-outcomes/

Some other modeling choices we could make:

- Additional Terms in model
- Modify covariates
- Include interaction terms
- Different link functions
- Others?

So, we're able to conduct inference, but does our model fit our data? Before we get too crazy perhaps we should analyze this. To do this, we need to talk about residuals. Next class we'll revisit deviance residuals, we'll talk about predictions (including confusion matrices and perhaps ROC curves), then we'll cover different choices for link functions.