

# Microeconometrics (Causal Inference)

## Week 3 - Discrete Choice

Joshua D. Merfeld  
KDI School of Public Policy and Management

2023-09-11

## What are we doing today?

- ▶ This week we will review methods for limited dependent variables
  - ▶ We will also discuss how to interpret the results
- ▶ Some of this will be review, but there will be some new stuff, too
  - ▶ For example, we will discuss poisson regression, which you might not have seen before

## What are we doing today?

- ▶ Much of the mathematical notation and theory comes from *Econometrics* by Bruce Hansen
  - ▶ I already discussed the older version of the textbook that is available for free online
- ▶ A small note:
  - ▶ You can use OLS for binary outcomes! This is actually pretty common in economics.
  - ▶ I'll discuss this more in a bit.
  - ▶ Other disciplines (e.g. public health) really like some of the new methods we will discuss today, so they are worth knowing.

- ▶ Let's start with the simplest possibility: binary choice
- ▶ You can think of this as a No/Yes or False/True question, but we will generally refer to it as 0/1 choice
  - ▶ In programming, always remember that *FALSE* = 0 and *TRUE* = 1
- ▶ Focusing on the binary choice case will allow us to build intuition for the more general case of discrete choice
  - ▶ We will also be able to use the same data as we move to the more general case

- ▶ We will be thinking about this  $Y \in \{0, 1\}$ 
  - ▶ In other words,  $Y$  is a dichotomous (dummy) variable that can only be equal to 0 or 1
- ▶ We are going to discuss methods to output *conditional probabilities*:

$$\mathbb{P}[Y = 1|X] \tag{1}$$

- ▶ Consider the following model:

$$Y = \beta X + \epsilon, \quad (2)$$

where  $X$  can have any number of columns (variables),  $k$ .

- ▶ We already know that  $\epsilon$  does not need to be normally distributed
- ▶ In fact, if  $Y \in \{0, 1\}$ , then  $\epsilon$  *will never be normally distributed*
  - ▶ Why?

- ▶  $\epsilon$  has a two-point conditional distribution:

$$\epsilon = \begin{cases} 1 - P(X), & \text{with probability } P(X) \\ P(X), & \text{with probability } 1 - P(X) \end{cases} \quad (3)$$

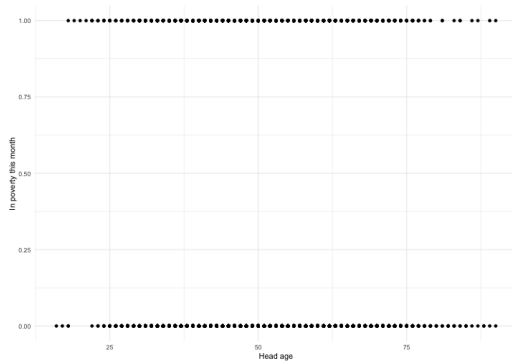
- ▶  $\epsilon$  is *heteroskedastic*:

$$\text{Var}(\epsilon|X) = P(X)(1 - P(X)) \quad (4)$$

- ▶ In fact, the variance of any dummy variable is  $P(1 - P)$ , where  $P$  is the probability of the dummy variable being equal to 1

# Scatterplots are pretty worthless!

```
# for reading in Stata data.  
# Install this using your console  
library(haven)  
# read in data for the week:  
df <- read_dta("data.dta")  
  
# scatter of in_poverty on h_age:  
ggplot(data = df) +  
  geom_point(aes(x = h_age, y = in_poverty)) +  
  labs(x = "Head age", y = "In poverty this month") +  
  theme_minimal()
```

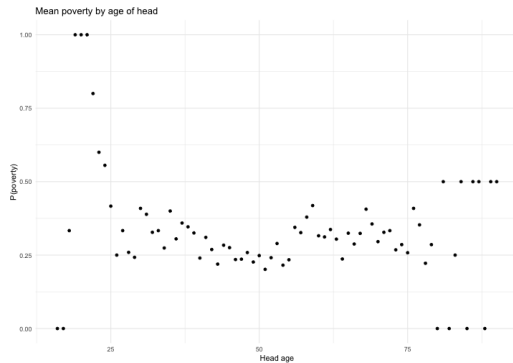




## Let's plot out the conditional probabilities (means)

```
# means by age
povmeans <- df %>%
  group_by(h_age) %>%
  summarize(mean = mean(in_poverty)) %>%
  ungroup

# scatter of in_poverty on h_age:
ggplot(data = povmeans) +
  geom_point(aes(x = h_age, y = mean)) +
  labs(x = "Head age", y = "P(poverty)") +
  theme_minimal()
```



- We can estimate this using OLS:

$$Y = \beta X + \epsilon \quad (5)$$

```
summary(feols(in_poverty ~ h_male + h_age, data = df, vcov = "HC1"))
```

```
## OLS estimation, Dep. Var.: in_poverty
## Observations: 4,609
## Standard-errors: Heteroskedasticity-robust
##           Estimate Std. Error   t value  Pr(>|t|)
## (Intercept)  0.352347   0.037235  9.462735 < 2.2e-16 ***
## h_male       -0.060625   0.025993 -2.332375  0.019724 *
## h_age        -0.000050   0.000553 -0.089808  0.928443
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.455294   Adj. R2: 8.54e-4
```

## Linear probability model (LPM)

```
summary(feols(in_poverty ~ h_male + h_age, data = df, vcov = "HC1"))
```

```
## OLS estimation, Dep. Var.: in_poverty
## Observations: 4,609
## Standard-errors: Heteroskedasticity-robust
##               Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)   0.352347   0.037235   9.462735 < 2.2e-16 ***
## h_male        -0.060625   0.025993  -2.332375  0.019724 *
## h_age         -0.000050   0.000553  -0.089808  0.928443
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.455294   Adj. R2: 8.54e-4
```

- ▶ The interpretation of these coefficients is pretty straightforward:
  - ▶ It is the change in the probability of being in poverty for a one-unit change in the variable of interest
  - ▶ Male-headed households are 6.1 *percentage points* less likely to be poor than female-headed households, controlling for age.
  - ▶ Each addition year of age *increases* the probability of being in poverty by 0.005 *percentage points*, controlling for gender of the head.

- ▶ Sometimes people motivate other estimation methods based on heteroskedasticity
  - ▶ But we can easily correct for this using robust standard errors (HC1 in feols)
- ▶ There are two other problems with LPM, though:
  - ▶ The predicted values can be outside of the 0-1 range
    - ▶ Is this a problem? Maybe. Maybe not. It depends on what you're doing.
  - ▶ Constant effects throughout the probability distribution
    - ▶ Is this realistic? If we think someone has a 95 percent probability of being poor, do we think the percentage point change would be the same for changing a variable relative to someone with a 5 percent probability of being poor?

- We can think of this as a *latent variable* model:

$$Y^* = \beta X + \epsilon \quad (6)$$

$$\epsilon \sim G(\epsilon) \quad (7)$$

$$Y = \mathbb{1}(Y^* > 0) = \begin{cases} 1, & \text{if } Y^* > 0 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where  $Y^*$  is the latent variable,  $G(\cdot)$  is the distribution of  $\epsilon$ , and  $\mathbb{1}(\cdot)$  is the indicator function.

- One way to think about this is that  $y^*$  is utility, but we only observe whether the choice increases utility ( $y = 1$ ) or doesn't ( $y = 0$ ).

## Let's give this a bit more structure

- ▶ Note that  $Y = 1$  iff  $Y^* > 0$ , which is the same as saying  $\beta X + \epsilon > 0$
- ▶ The response probability is then given by the CDF of  $\epsilon$  evaluated at  $-\beta X$ :

$$\mathbb{P}[Y = 1|X] = \mathbb{P}[\epsilon > -\beta X] = 1 - G(-\beta X) = G(\beta X) \quad (9)$$

- ▶ Note that CDFs (cumulative distribution functions) give us probabilities of being *less than or equal to* a given value
  - ▶ The last equality holds because  $G(\cdot)$  will always be *symmetric around 0* here
  - ▶ That value here is  $\beta X$

- ▶ The function  $G(\cdot)$  is called the *link function* and plays an important role here
- ▶ Two common link functions are the *logit* and *probit* link functions:
  - ▶ They are defined as follows:
  - ▶ Logit:  $G(\epsilon) = \frac{e^\epsilon}{1+e^\epsilon} = \frac{1}{1+e^{-\epsilon}}$
  - ▶ Probit:  $G(\epsilon) = \Phi(\epsilon)$ , where  $\Phi(\cdot)$  is the CDF of the standard normal distribution
- ▶ We will discuss these in more detail in a bit

- ▶ Likelihood: the joint probability of the data evaluated with the sample, as a function of the parameters
  - ▶ What?
- ▶ Let's start with probit, which uses a normal distribution. Here is the conditional density of  $Y$  given  $X$  under this assumption:

$$f(Y|X) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(Y - \beta X)^2\right) \quad (10)$$



$$f(Y|X) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \beta X)^2\right) \quad (11)$$

- In this case, what is the probability we *observe our sample given the values of  $\beta$  and  $\sigma$* ?

$$f(y_1, \dots, y_n | x_1, \dots, x_n) = \prod_{i=1}^n f(y_i | x_i) \quad (12)$$

$$= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta x_i)^2\right) \quad (13)$$

$$= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \prod_{i=1}^n (y_i - \beta x_i)^2\right) \quad (14)$$

$$= L_n(\beta, \sigma^2) \quad (15)$$

$$f(y_1, \dots, y_n | x_1, \dots, x_n) = L_n(\beta, \sigma^2) \quad (16)$$

- ▶ This is the *likelihood function*
  - ▶ Note that it is a function of the parameters,  $\beta$  and  $\sigma^2$
- ▶ The properties of logs make this easier to work with:

$$\ell_n(\beta, \sigma^2) = \log(L_n(\beta, \sigma^2)) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \beta x_i)^2 \quad (17)$$

- ▶ This is the *log likelihood function*
  - ▶ It is of course also a function of the parameters,  $\beta$  and  $\sigma^2$

$$\ell_n(\beta, \sigma^2) = \log(L_n(\beta, \sigma^2)) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \beta x_i)^2 \quad (18)$$

- ▶ We have our log likelihood function, which is a function of the parameters,  $\beta$  and  $\sigma^2$
- ▶ What we want to do is find the values of  $\beta$  and  $\sigma^2$  that *maximize* this function
  - ▶ In other words, the values that make our sample the most likely to have been observed, or the biggest probability of observing our sample
- ▶ This is called *maximum likelihood estimation*

$$(\hat{\beta}, \hat{\sigma}^2) = \underset{\beta \in \mathbb{R}^k, \sigma^2 > 0}{\operatorname{argmax}} \ell_n(\beta, \sigma^2) \quad (19)$$

where  $k$  is the number of variables (coefficients), including the intercept.

- ▶ A simple example is a coin flip
- ▶ Let's say we flip a coin. If it's a fair coin, what is the probability of obtaining heads?

- ▶ A simple example is a coin flip
- ▶ Let's say we flip a coin. If it's a fair coin, what is the probability of obtaining heads?
  - ▶ 50% or 0.5 (we generally work with the proportion 0.5, and not the percent)

- ▶ A simple example is a coin flip
- ▶ Let's say we flip a coin. If it's a fair coin, what is the probability of obtaining heads?
  - ▶ 50% or 0.5 (we generally work with the proportion 0.5, and not the percent)
- ▶ What is the probability of obtaining heads twice in a row?

- ▶ A simple example is a coin flip
- ▶ Let's say we flip a coin. If it's a fair coin, what is the probability of obtaining heads?
  - ▶ 50% or 0.5 (we generally work with the proportion 0.5, and not the percent)
- ▶ What is the probability of obtaining heads twice in a row?
  - ▶  $0.5 * 0.5 = 0.25$

- ▶ A simple example is a coin flip
- ▶ Let's say we flip a coin. If it's a fair coin, what is the probability of obtaining heads?
  - ▶ 50% or 0.5 (we generally work with the proportion 0.5, and not the percent)
- ▶ What is the probability of obtaining heads twice in a row?
  - ▶  $0.5 * 0.5 = 0.25$
- ▶ Three times in a row?



- ▶ A simple example is a coin flip
- ▶ Let's say we flip a coin. If it's a fair coin, what is the probability of obtaining heads?
  - ▶ 50% or 0.5 (we generally work with the proportion 0.5, and not the percent)
- ▶ What is the probability of obtaining heads twice in a row?
  - ▶  $0.5 * 0.5 = 0.25$
- ▶ Three times in a row?
  - ▶  $0.5 * 0.5 * 0.5 = 0.125$

- ▶ Say we flip the coin a bunch of times
  - ▶ For argument's sake, let's say we flip it 100 times and obtain 60 heads
- ▶ If we know nothing about whether the coin is actually fair, what is the *most likely distribution* that would give us 60 heads and 40 tails?

- ▶ Say we flip the coin a bunch of times
  - ▶ For argument's sake, let's say we flip it 100 times and obtain 60 heads
- ▶ If we know nothing about whether the coin is actually fair, what is the *most likely distribution* that would give us 60 heads and 40 tails?
  - ▶ It's a distribution in which the probability of heads is 0.6!

- ▶ Say we flip the coin a bunch of times
  - ▶ For argument's sake, let's say we flip it 100 times and obtain 60 heads
- ▶ If we know nothing about whether the coin is actually fair, what is the *most likely distribution* that would give us 60 heads and 40 tails?
  - ▶ It's a distribution in which the probability of heads is 0.6!
- ▶ This is like maximum likelihood estimation. We are trying to find the parameters that makes our sample the most likely to have been observed.
  - ▶ In this case, the parameter would be the true mean of the distribution of the coin (where heads = 1 and tails = 0).
  - ▶ We could then of course test whether this is significantly different from 0.5, which might be our null hypothesis.

$$\left(\hat{\beta}, \hat{\sigma}^2\right) = \underset{\beta \in \mathbb{R}^k, \sigma^2 > 0}{\operatorname{argmax}} \ell_n(\beta, \sigma^2) \quad (20)$$

- ▶ MLE is generally always estimated using numerical optimization
  - ▶ We will not discuss the details of this here
  - ▶ The basic reason is that most likelihood functions are not easy to maximize analytically (i.e. they have no closed-form solution)
- ▶ In the case of the normal regression model, however, there is a closed-form solution
  - ▶ And this is the same closed-form solution as OLS!

- ▶ Let's return to our binary choice model.
  - ▶ Regardless of how you estimate it, the probability mass function for  $Y$  is:

$$\pi(y) = p^y(1 - p)^{1-y}, \quad (21)$$

where  $p$  is the probability of  $Y = 1$ , or the mean. Remember that  $Y \in \{0, 1\}$ ; i.e., it can only equal 0 or 1.

- ▶ Let's bring our link function back into it. The *conditional* probability is:

$$\pi(Y|X) = G(\beta X)^Y(1 - G(\beta X))^{1-Y} = G(\beta X)^Y(1 - G(\beta X))^{1-Y} = G(\beta Z), \quad (22)$$

$$\text{where } Z = \begin{cases} X, & \text{if } Y = 1 \\ -X, & \text{if } Y = 0 \end{cases}$$

$$\pi(Y|X) = G(\beta Z), \quad (23)$$

- Taking logs (because they're easy to work with), we get the log likelihood function:

$$\ell_n(\beta) = \sum_{i=1}^n \log G(\beta Z) \quad (24)$$

- This is the same as the log likelihood function for probit, except that the link function is different.

- ▶ Again, we want to find the values of  $\beta$  (and  $\sigma$ , which will show up in the link function) that maximize this function.

$$(\hat{\beta}, \hat{\sigma}^2) = \underset{\beta \in \mathbb{R}^k, \sigma^2 > 0}{\operatorname{argmax}} \ell_n(\beta, \sigma^2) \quad (25)$$

- ▶ Something interesting is that in practice, we don't numerically optimize...
  - ▶ Instead, we *minimize* the *negative* of the log likelihood function!

$$(\hat{\beta}, \hat{\sigma}^2) = \underset{\beta \in \mathbb{R}^k, \sigma^2 > 0}{\operatorname{argmin}} -\ell_n(\beta, \sigma^2) \quad (26)$$



## An example in R - Household variables and poverty using glm()

```
summary(glm(in_poverty ~ h_male, data = df, family = binomial(link = "logit")))
```

```
##
## Call:
## glm(formula = in_poverty ~ h_male, family = binomial(link = "logit"),
##      data = df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9280  -0.8263  -0.8263   1.5752   1.5752
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6196     0.1101  -5.631  1.8e-08 ***
## h_male       -0.2796     0.1151  -2.428   0.0152 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 5583.2  on 4608  degrees of freedom
## Residual deviance: 5577.5  on 4607  degrees of freedom
## AIC: 5581.5
##
## Number of Fisher Scoring iterations: 4
```

## Interpreting logit output

```
##           Estimate Std. Error   z value    Pr(>|z|)
## (Intercept) -0.6196447  0.1100514 -5.630505 1.796828e-08
## h_male      -0.2795628  0.1151389 -2.428047 1.518036e-02
```

- ▶ How do we interpret these coefficients?
  - ▶ The coefficients are “log odds”
- ▶ For male household heads, the log odds of being in poverty is 0.280 *lower* than that for female household heads
  - ▶ What?

- ▶ What are log odds?
  - ▶ Let's start with odds:

$$\text{odds} = \frac{p}{1-p}, \quad (27)$$

where  $p$  is the probability of  $y = 1$  (being poor in this case).

- ▶ Log odds?

$$\log \text{ odds} = \log \left( \frac{p}{1-p} \right) \quad (28)$$

- Logit regression is basically estimating:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k \quad (29)$$

## The intercept is the log odds of being in poverty for female households

```
##           Estimate Std. Error  z value    Pr(>|z|)
## (Intercept) -0.6196447  0.1100514 -5.630505 1.796828e-08
## h_male      -0.2795628  0.1151389 -2.428047 1.518036e-02
```

$$\log\left(\frac{p}{1-p}\right) = -0.6196447 \quad (30)$$

$$\left(\frac{p}{1-p}\right) = \exp(-0.6196447) \quad (31)$$

$$\left(\frac{p}{1-p}\right) \approx 0.538 \quad (32)$$

$$p = 0.538 - 0.538p \quad (33)$$

$$1.538p = 0.538 \quad (34)$$

$$p \approx 0.350 \quad (35)$$

What is the actual mean for female headed households? 0.35

$$\log \left( \frac{p_m}{1-p_m} \right) - \log \left( \frac{p_f}{1-p_f} \right) = -0.2795628 \quad (36)$$

$$\log \left( \frac{\frac{p_m}{1-p_m}}{\frac{p_f}{1-p_f}} \right) = -0.2795628 \quad (37)$$

$$\left( \frac{\frac{p_m}{1-p_m}}{\frac{p_f}{1-p_f}} \right) = \exp(-0.2795628) \quad (38)$$

$$\left( \frac{\frac{p_m}{1-p_m}}{\frac{p_f}{1-p_f}} \right) = 0.7561142 \quad (39)$$

- In the last line, this is referred to as an *odds ratio*.
  - It's less than one, which means male-headed households are *less likely* to be in poverty.
  - Their *odds* of being in poverty are around 24% lower.

## The coefficient?

- ▶ Mean for female-headed households: 0.35
  - ▶ odds ( $\frac{p_f}{1-p_f}$ ): 0.538
- ▶ Mean for male-headed households: 0.289
  - ▶ odds ( $\frac{p_m}{1-p_m}$ ): 0.407
- ▶ Odds ratio: 0.756
- ▶ Exponentiating shows us the odds ratio!

- ▶ We can also calculate marginal effects
  - ▶ These are the change in the probability of being in poverty for a one-unit change in the variable of interest
  - ▶ An important note is that this *depends on where you are located in the distribution*
- ▶ We just calculated the means, so with only the binary independent variable, we know that the marginal effect is:
  - ▶ -0.0606
- ▶ We will use the “mfx” package for this, so please install it in the console.



## Marginal effects

```
logitmfx(in_poverty ~ h_male, data = df)

## Call:
## logitmfx(formula = in_poverty ~ h_male, data = df)
##
## Marginal Effects:
##           dF/dx Std. Err.      z    P>|z|
## h_male -0.060649  0.025981 -2.3343 0.01958 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "h_male"
# for binary outcomes, it shows the change from 0 to 1!
```

## logit with more coefficients

```
logitmfx(in_poverty ~ h_male + h_age, data = df)

## Call:
## logitmfx(formula = in_poverty ~ h_male + h_age, data = df)
##
## Marginal Effects:
##           dF/dx   Std. Err.         z   P>|z|
## h_male -6.0623e-02  2.5982e-02 -2.3333 0.01963 *
## h_age  -4.9739e-05  5.3672e-04 -0.0927 0.92616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "h_male"
# for binary outcomes, it shows the change from 0 to 1!
# for continuous variables, it's the derivative (i.e. instantaneous change)!
# By default, it calculates these by holding variables AT THEIR MEANS
```

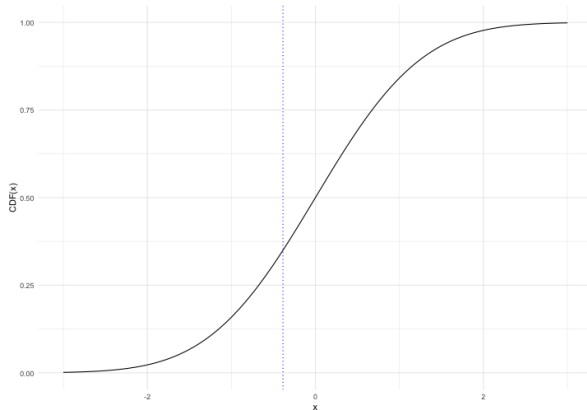
```
summary(glm(in_poverty ~ h_male, data = df, family = binomial(link = "probit")))$coefficients
```

```
##              Estimate Std. Error   z value    Pr(>|z|)
## (Intercept) -0.3856924 0.06759123 -5.706248 1.154935e-08
## h_male      -0.1699918 0.07058912 -2.408188 1.603194e-02
```

- ▶ What about probit coefficients?
  - ▶ These relate to the *CDF of the standard normal distribution*
- ▶ The intercept is the mean for female-headed households

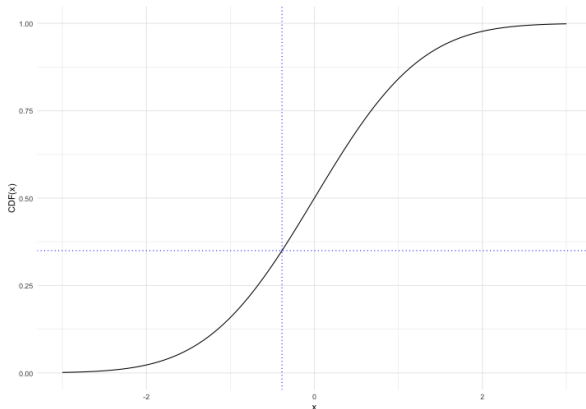
## Standard normal CDF

- ▶ The intercept is  $-0.3856924$ 
  - ▶ The mean poverty for female-headed households is 0.35
- ▶ Here's the CDF for the standard normal distribution with the intercept:



## Standard normal CDF

- ▶ The intercept is -0.3856924
  - ▶ The mean poverty for female-headed households is 0.35
- ▶ Here's the CDF for the standard normal distribution with BOTH:



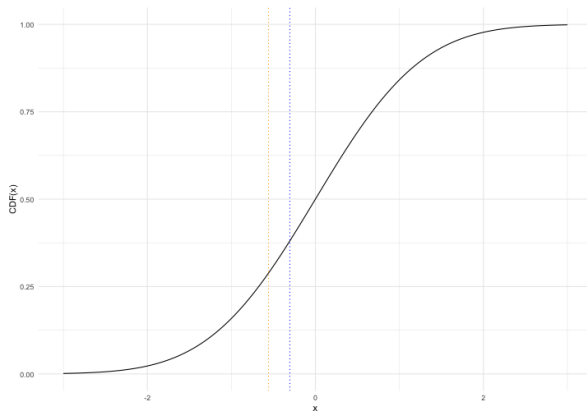
## Now let's look at the coefficient on h\_male

```
summary(glm(in_poverty ~ h_male, data = df, family = binomial(link = "probit")))$coefficients
```

##	Estimate	Std. Error	z value	Pr(> z )
## (Intercept)	-0.3856924	0.06759123	-5.706248	1.154935e-08
## h_male	-0.1699918	0.07058912	-2.408188	1.603194e-02

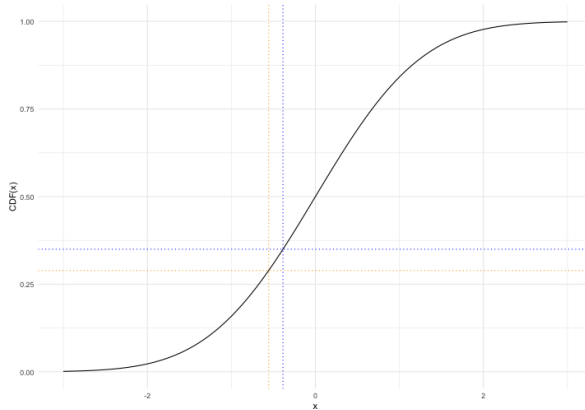
## Standard normal CDF

- ▶ The intercept is -0.3038412 and the coefficient is -0.1699918
- ▶ Here's the CDF for the standard normal distribution with BOTH:



## Standard normal CDF

- ▶ The intercept is -0.3856924 and the coefficient is -0.1699918
- ▶ What's the change in PROBABILITY?  $\text{mean}(\text{male}) - \text{mean}(\text{female})$  or -0.0606





- ▶ The intercept is -0.3856924 and the coefficient is -0.1699918
- ▶ What's the change in PROBABILITY?  $\text{mean}(\text{male}) - \text{mean}(\text{female})$  or -0.0606

```
probitmfx(in_poverty ~ h_male, data = df)
```

```
## Call:
## probitmfx(formula = in_poverty ~ h_male, data = df)
##
## Marginal Effects:
##           dF/dx Std. Err.      z    P>|z|
## h_male -0.060649  0.025981 -2.3343 0.01958 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "h_male"
```

## Probit and marginal effects

```
# change in z-scores
summary(glm(in_poverty ~ h_male + h_age, data = df, family = binomial(link = "probit")))$coefficients
```

```
##              Estimate Std. Error    z value    Pr(>|z|)
## (Intercept) -0.3783277748 0.103202161 -3.66589005 0.0002464798
## h_male      -0.1699296493 0.070592873 -2.40717853 0.0160763088
## h_age       -0.0001469862 0.001557869 -0.09435083 0.9248304734
```

```
# change in probability, holding other variables at their means
probitmfx(in_poverty ~ h_male + h_age, data = df)
```

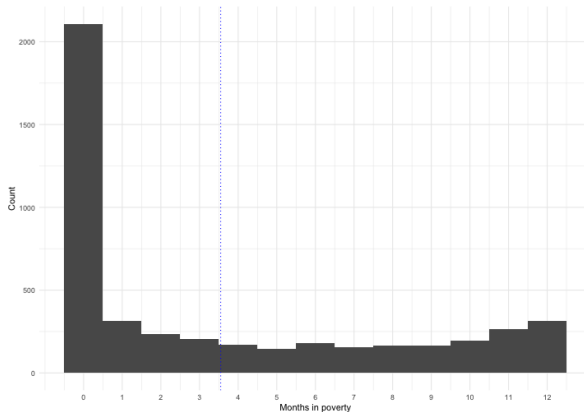
```
## Call:
## probitmfx(formula = in_poverty ~ h_male + h_age, data = df)
##
## Marginal Effects:
##          dF/dx    Std. Err.      z    P>|z|
## h_male -6.0626e-02  2.5982e-02 -2.3334 0.01963 *
## h_age  -5.0621e-05  5.3651e-04 -0.0944 0.92483
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## dF/dx is for discrete change for the following variables:
##
## [1] "h_male"
```

## What if the outcome has more than two categories?

- ▶ Many outcomes are not 0/1.
- ▶ We can think of outcomes with discrete categories, but more than two:
  - ▶ Religion
  - ▶ Political party
  - ▶ Opinion on a likert scale (e.g. strongly agree, agree, neutral, disagree, strongly disagree)
  - ▶ Months in poverty
- ▶ There's a key difference between the first two and the last two:
  - ▶ The first two are *unordered*
  - ▶ The last two are *ordered*

## Months in poverty - distribution

```
ggplot() +  
  geom_histogram(data = df, aes(x = months_in_poverty), binwidth = 1) +  
  theme_minimal() +  
  labs(x = "Months in poverty", y = "Count") +  
  scale_x_continuous(breaks = seq(0, 12, 1)) +  
  geom_vline(xintercept = mean(df$months_in_poverty), linetype = "dotted", color = "blue")
```



## Ordered logistic regression with the polr function

```
# note that the outcome must be a FACTOR variable for polr  
summary(polr(as_factor(months_in_poverty) ~ h_male + h_age, data = df, Hess = TRUE))
```

```
## Call:  
## polr(formula = as_factor(months_in_poverty) ~ h_male + h_age,  
##       data = df, Hess = TRUE)  
##  
## Coefficients:  
##              Value Std. Error t value  
## h_male -0.3826883    0.10035 -3.81337  
## h_age   0.0001876    0.00216  0.08681  
##  
## Intercepts:  
##              Value Std. Error t value  
## 0|1   -0.5168  0.1457   -3.5478  
## 1|2   -0.2434  0.1455   -1.6730  
## 2|3   -0.0376  0.1455   -0.2585  
## 3|4    0.1472  0.1455    1.0119  
## 4|5    0.3078  0.1455    2.1149  
## 5|6    0.4514  0.1456    3.1004  
## 6|7    0.6416  0.1458    4.4014  
## 7|8    0.8171  0.1460    5.5975  
## 8|9    1.0249  0.1463    7.0044  
## 9|10   1.2665  0.1469    8.6198  
## 10|11  1.6049  0.1482   10.8280  
## 11|12  2.2775  0.1529   14.8999  
##  
## Residual Deviance: 18592.59  
## AIC: 18620.59
```

- ▶ When we have ordered discrete variable, we can use an ordered logit or probit model
  - ▶ These are also called *ordinal* logit/probit models
- ▶ The idea is that we have a latent variable,  $Y^*$ , that is continuous
  - ▶ We observe  $Y$  as a discrete variable, but it is *ordered*
  - ▶ We can think of  $Y$  as being *binned* into categories

$$Y = \begin{cases} 1, & \text{if } Y^* \in (-\infty, \theta_1] \\ 2, & \text{if } Y^* \in (\theta_1, \theta_2] \\ \vdots & \vdots \\ J, & \text{if } Y^* \in (\theta_{J-1}, \infty) \end{cases} \quad (40)$$

## The interpretation?

```
## Call:
## polr(formula = as_factor(months_in_poverty) ~ h_male + h_age,
##       data = df, Hess = TRUE)
##
## Coefficients:
##              Value Std. Error  t value
## h_male -0.3826883    0.10035 -3.81337
## h_age   0.0001876    0.00216  0.08681
##
## Intercepts:
##      Value  Std. Error t value
## 0|1  -0.5168  0.1457   -3.5478
## 1|2  -0.2434  0.1455   -1.6730
## 2|3  -0.0376  0.1455   -0.2585
## 3|4   0.1472  0.1455    1.0119
## 4|5   0.3078  0.1455    2.1149
## 5|6   0.4514  0.1456    3.1004
## 6|7   0.6416  0.1458    4.4014
## 7|8   0.8171  0.1460    5.5975
## 8|9   1.0249  0.1463    7.0044
## 9|10  1.2665  0.1469    8.6198
## 10|11 1.6049  0.1482   10.8280
## 11|12 2.2775  0.1529   14.8999
##
## Residual Deviance: 18592.59
## AIC: 18620.59
```

- The interpretation is similar to logit: a change in the log-odds of being in a higher level of months in poverty!

- ▶ There is no r-squared in MLE
  - ▶ It is not a true r-squared *because there is no sense of “mean” with discrete data, especially unordered data*
- ▶ We can use the log likelihood function to compare models
  - ▶ The log likelihood function is a function of the parameters,  $\beta$  and  $\sigma^2$
  - ▶ The higher the log likelihood, the better the fit
- ▶ We can also use the *Akaike Information Criterion* (AIC) and *Bayesian Information Criterion* (BIC)
  - ▶ These are functions of the log likelihood function and the number of parameters
  - ▶ The lower the AIC/BIC, the better the fit
- ▶ These are best thought of as useful for comparing across models
  - ▶ Difficult to interpret them on their own



- ▶ AIC is defined as follows:
  - ▶  $k$  is the number of parameters in the model
  - ▶  $L$  is the log likelihood function
  - ▶ AIC:  $2k - 2 \log(L)$

```
# save our model
results <- glm(in_poverty ~ h_male + h_age, data = df, family = binomial(link = "probit"))
```

```
# log likelihood
logLik(results)
```

```
## 'log Lik.' -2788.727 (df=3)
```

```
# aic from temp
results$aic
```

```
## [1] 5583.453
```

```
# Calculate AIC by hand:
2*3 - 2*(-2788.727)
```

```
## [1] 5583.454
```

- ▶ BIC is defined as follows:
  - ▶  $k$  is the number of parameters in the model
  - ▶  $L$  is the log likelihood function
  - ▶  $n$  is the number of observations
  - ▶ BIC:  $k \log(n) - 2 \log(L)$

```
# save our model
results <- glm(in_poverty ~ h_male + h_age, data = df, family = binomial(link = "probit"))

# log likelihood
logLik(results)
```

```
## 'log Lik.' -2788.727 (df=3)
nrow(df)
```

```
## [1] 4609
# Calculate BIC by hand:
3*log(4609) - 2*(-2788.727)
```

```
## [1] 5602.761
```

- ▶ We can also calculate a pseudo r-squared
  - ▶ This is a measure of the change in the log likelihood function relative to the null model (no coefficients except intercepts)

```
# null model
logLik(glm(in_poverty ~ 1, data = df, family = binomial(link = "probit")))

## 'log Lik.' -2791.606 (df=1)

# our model
logLik(glm(in_poverty ~ h_male + h_age, data = df, family = binomial(link = "probit")))

## 'log Lik.' -2788.727 (df=3)

# pseudo r-squared: 1 - (log likelihood of model / log likelihood of null model)
1 - (-2788.727/-2791.606)

## [1] 0.001031306

# Same thing: (null - model) / null
(-2791.606 - -2788.727)/-2791.606

## [1] 0.001031306
```

- ▶ Ordered logit/probit is used when the outcome is *ordered*
- ▶ But what if it's not, like trying to predict what the species of a flower is?
  - ▶ Let's use a built-in dataset in R called "iris" to look at this:

```
data(iris)
head(iris)
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa

```
data(iris)
table(iris$Species)
```

```
##
##      setosa versicolor  virginica
##         50         50         50
```

```
colnames(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

- Let's use sepal/petal length/width to predict the species

```
data(iris)
library(nnet)
multinomresults <- multinom(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = iris)
```

```
summary(multinomresults)$coefficients
```

```
##           (Intercept) Sepal.Length Sepal.Width Petal.Length Petal.Width
## versicolor    18.69037    -5.458424   -8.707401     14.24477    -3.097684
## virginica     -23.83628    -7.923634  -15.370769     23.65978     15.135301
```

- Note how setosa isn't there... it's the omitted category
  - We can interpret the coefficients as the log odds of being in a particular category relative to setosa (the omitted category)

- ▶ GLM is a very general framework
  - ▶ We can use it for other distributions, too
- ▶ Let's look at a poisson distribution
  - ▶ The poisson distribution is often used for count data
  - ▶ It has assumptions (mean = variance), but violation isn't a big deal!

## The poisson distribution

- ▶ The poisson distribution is defined as follows:

$$f(y; \lambda) = \frac{\lambda^y e^{-\lambda}}{y!} \quad (41)$$

- ▶ Note the  $e$ : this is going to lead to a nice log interpretation
- ▶  $y$  is the number of occurrences of the variable (in our example, it will be number of months in poverty)
- ▶ As mentioned, one implication of this distribution is:

$$E(y) = \text{Var}(y) = \lambda \quad (42)$$

i.e. the mean of  $y$  equals its variance. But we can work around this if it's false (which it probably is).



## Poisson, months in poverty (with feols - feglm)

```
# could save results. Not going to here... just display them
summary(feglm(months_in_poverty ~ h_male + h_age, data = df, family = "poisson"))
```

```
## GLM estimation, family = poisson, Dep. Var.: months_in_poverty
## Observations: 4,609
## Standard-errors: IID
##           Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)  1.456020   0.039857  36.53087 < 2.2e-16 ***
## h_male       -0.277845   0.025915 -10.72158 < 2.2e-16 ***
## h_age         0.001255   0.000625   2.00869  0.04457 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Log-Likelihood: -17,506.6   Adj. Pseudo R2: 0.00303
##           BIC:  35,038.6       Squared Cor.: 0.004978
```

```
# notice the difference in standard errors if we use HC1 (which we want to here because of overdispersion)
summary(feglm(months_in_poverty ~ h_male + h_age, data = df, family = "poisson", vcov = "HC1"))
```

```
## GLM estimation, family = poisson, Dep. Var.: months_in_poverty
## Observations: 4,609
## Standard-errors: Heteroskedasticity-robust
##           Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)  1.456020   0.092210 15.790222 < 2.2e-16 ***
## h_male       -0.277845   0.057268 -4.851679 1.2242e-06 ***
## h_age         0.001255   0.001487  0.844145 3.9859e-01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Log-Likelihood: -17,506.6   Adj. Pseudo R2: 0.00303
##           BIC:  35,038.6       Squared Cor.: 0.004978
```

## Interpreting poisson coefficients

```
summary(feglm(months_in_poverty ~ h_male, data = df, family = "poisson", vcov = "HC1"))$coefficients
```

```
## (Intercept)      h_male  
##  1.5189645  -0.2772304
```

- ▶ How do we interpret poisson coefficients?
  - ▶ They are the *log* of the *rate* of the outcome
  - ▶ i.e. the log of the number of months in poverty
- ▶ The log count for male-headed households is around -0.28 *lower* than for female-headed households

## The intercept is log count for female households

```
summary(feglm(months_in_poverty ~ h_male, data = df, family = "poisson", vcov = "HC1"))$coefficients
```

```
## (Intercept)      h_male
```

```
##  1.5189645  -0.2772304
```

```
# intercept is log count, so exponentiate for levels
```

```
exp(1.5189645)
```

```
## [1] 4.567493
```

```
# what is the mean for female-headed households?
```

```
mean(df$months_in_poverty[df$h_male==0])
```

```
## [1] 4.567493
```

```
# intercept plus coefficient for male-headed households
```

```
exp(1.5189645 - 0.2772304)
```

```
## [1] 3.461611
```

```
# mean for male-headed households?
```

```
mean(df$months_in_poverty[df$h_male==1])
```

```
## [1] 3.461611
```

- ▶ The poisson distribution has a mean = variance assumption
  - ▶ This is often violated
  - ▶ We can use quasi-poisson instead
- ▶ The quasi-poisson is the same as poisson, but the variance is estimated from the data
  - ▶ With feglm, it estimates the variance as a *linear function of the mean*
- ▶ Small note: if you use glm, you can use the “quasipoisson” family
  - ▶ This is more similar to poisson with `vcov = “HC1”`!
    - ▶ If you use “HC1” in both, you’ll get identical results.
  - ▶ This is only about the structure of the error term. Not the coefficients.

## Poisson vs quasi-poisson (note the similar standard errors with HC1 for poisson)

```
# Poisson with HC1
```

```
summary(feglm(months_in_poverty ~ h_male + h_age, data = df, family = "poisson", vcov = "HC1"))
```

```
## GLM estimation, family = poisson, Dep. Var.: months_in_poverty
## Observations: 4,609
## Standard-errors: Heteroskedasticity-robust
##           Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)  1.456020   0.092210 15.790222 < 2.2e-16 ***
## h_male       -0.277845   0.057268 -4.851679 1.2242e-06 ***
## h_age        0.001255   0.001487  0.844145 3.9859e-01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Log-Likelihood: -17,506.6   Adj. Pseudo R2: 0.00303
##           BIC:  35,038.6       Squared Cor.: 0.004978
```

```
# quasipoisson
```

```
summary(feglm(months_in_poverty ~ h_male + h_age, data = df, family = "quasipoisson"))
```

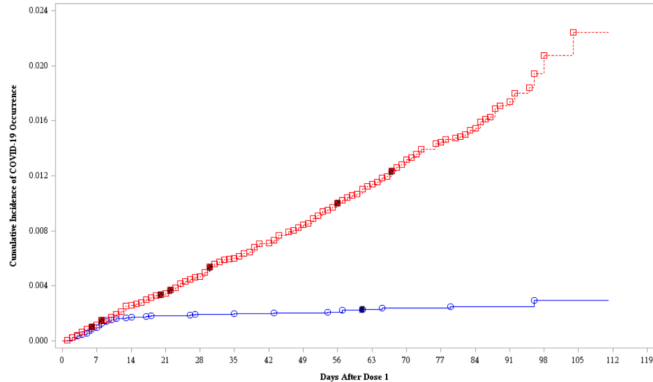
```
## GLM estimation, family = quasipoisson, Dep. Var.: months_in_poverty
## Observations: 4,609
## Standard-errors: IID
##           Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)  1.456020   0.091127 15.977892 < 2.2e-16 ***
## h_male       -0.277845   0.059249 -4.689409 2.8192e-06 ***
## h_age        0.001255   0.001429  0.878563 3.7968e-01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           Squared Cor.: 0.004978
```

## Doesn't have to be integers!

- ▶ You can use poisson for non-integer outcomes, too!
  - ▶ It's just a distribution
  - ▶ It's often used for integer outcomes because it's a count distribution
- ▶ Jeff Wooldridge is a huge proponent of using (quasi) poisson
- ▶ Two really nice things: it is robust and has an easy interpretation

- ▶ These models are used for *duration* data
  - ▶ i.e. time until an event occurs or a state changes
  - ▶ e.g. time until death after being diagnosed with cancer, time until a person leaves poverty, time until a person gets a job, until contracting a disease, etc.
- ▶ You need a very specific kind of data for this. . .
  - ▶ We want to know what kinds of variables predict the occurrence of some event (e.g. death).
- ▶ **Warning:** I am not an expert on survival models. I will give you a brief overview, but you should consult a textbook for more information.

**Figure 13 Cumulative Incidence Curves for the First COVID-19 Occurrence After Dose 1 – Dose 1 All-Available Efficacy Population**



No. with events/No. at risk

A: 0/21314 21/21230 37/21054 39/20481 41/19314 42/18377 42/17702 43/17186 44/15464 47/14038 48/12169 48/9291 48/6403 48/3374 50/1463 50/998 50/0  
 B: 0/21258 25/21170 55/20970 73/20366 97/19209 123/18218 143/17578 166/17025 192/15290 212/13876 235/11994 248/9471 257/6294 267/3301 274/1449 275/998 275/0

—○— A: BNT162b2 (30 µg) —□— B: Placebo



- ▶ Let's use the package survival in R.
  - ▶ It has a dataset set up for us, called diabetic

```
library(survival)
# dataset with "high-risk" diabetic retinopathy patients
head(diabetic)
```

```
##   id laser age  eye trt risk  time status
## 1  5 argon  28 left  0   9 46.23      0
## 2  5 argon  28 right 1   9 46.23      0
## 3 14 xenon  12 left  1   8 42.50      0
## 4 14 xenon  12 right 0   6 31.30      1
## 5 16 xenon   9 left  1  11 42.27      0
## 6 16 xenon   9 right 0  11 42.27      0
```

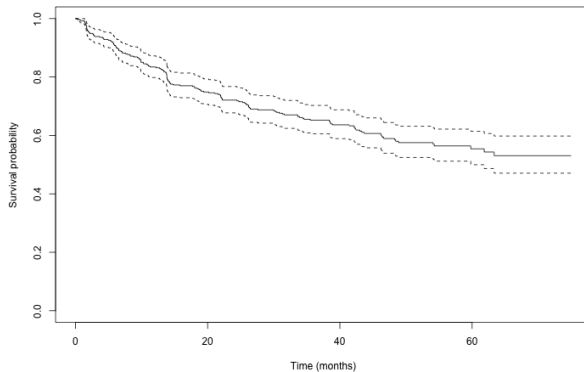
- ▶ Note: this is a *panel* dataset
  - ▶ Each row is a patient (id)
  - ▶ Each patient has multiple observations (time)
  - ▶ Interested in loss of sight (status = 1)

- ▶ Let's first look at the survival function
  - ▶ This is the probability of surviving past a certain time
  - ▶ We're going to use the Kaplan-Meier estimator

```
KM <- survfit(Surv(time = time, event = status) ~ 1, data = diabetic)
# note that Surv() is necessary here. It's a function that creates a survival object.
# The ~ 1 means this is for EVERYONE.
```

# Survival curve

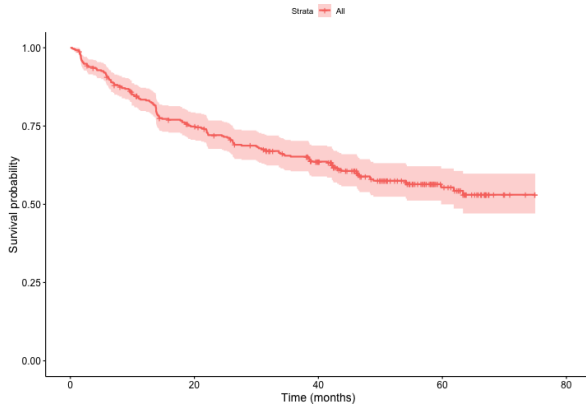
```
KM <- survfit(Surv(time = time, event = status) ~ 1, data = diabetic)
plot(KM, ylab = "Survival probability", xlab = "Time (months)")
```



## Comparing survival curves based on treatment

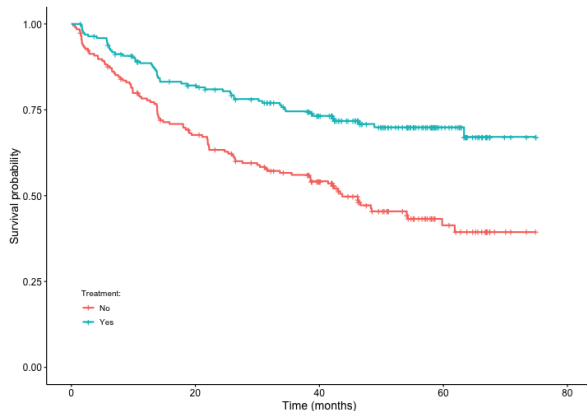
- Base R has ugly plots. We can use survminer to make it work with ggplot.

```
library(survminer)
KM <- survfit(Surv(time = time, event = status) ~ 1, data = diabetic)
ggsurvplot(KM) +
  labs(y = "Survival probability", x = "Time (months)")
```



# Comparing survival curves based on treatment

```
KM <- survfit(Surv(time = time, event = status) ~ trt, data = diabetic)
ggsurvplot(KM)$plot +
  labs(y = "Survival probability", x = "Time (months)") +
  scale_color_discrete("Treatment:", labels = c("No", "Yes")) +
  theme(legend.position = c(0.1, 0.2))
```



## Comparing survival curves based on treatment, empirically

```
survdif(Surv(time = time, event = status) ~ trt, data = diabetic)
```

```
## Call:
```

```
## survdif(formula = Surv(time = time, event = status) ~ trt, data = diabetic)
```

```
##
```

```
##           N Observed Expected (O-E)^2/E (O-E)^2/V
```

```
## trt=0 197      101      71.8      11.9      22.2
```

```
## trt=1 197       54      83.2      10.3      22.2
```

```
##
```

```
## Chisq= 22.2 on 1 degrees of freedom, p= 2e-06
```

```
# changes the weighting (more weight on earlier time points); doesn't matter here! Huge differences.
```

```
survdif(Surv(time = time, event = status) ~ trt, data = diabetic, rho = 1)
```

```
## Call:
```

```
## survdif(formula = Surv(time = time, event = status) ~ trt, data = diabetic,
```

```
##      rho = 1)
```

```
##
```

```
##           N Observed Expected (O-E)^2/E (O-E)^2/V
```

```
## trt=0 197      80.3      57.6      8.95      20.7
```

```
## trt=1 197      43.1      65.8      7.84      20.7
```

```
##
```

```
## Chisq= 20.7 on 1 degrees of freedom, p= 6e-06
```

- ▶ Plotting two survival curves with a simply treatment dummy is straightforward.
  - ▶ But what if we want to add more covariates?
  - ▶ For example, the diabetic dataset has age at diagnosis and which eye the problem is. Does this matter?
- ▶ We can use a Cox proportional hazards model to do this.
  - ▶ This is a semi-parametric model that is very popular in survival analysis.
  - ▶ It is a *proportional hazards* model, which means that the hazard ratio is constant over time.
  - ▶ Its nature means we do not estimate the baseline hazard function.
    - ▶ Instead, we compare across variables

## Adding more covariates

```
coxph(Surv(time = time, event = status) ~ age + as_factor(eye) + trt, data = diabetic)
```

```
## Call:
## coxph(formula = Surv(time = time, event = status) ~ age + as_factor(eye) +
##       trt, data = diabetic)
##
##               coef exp(coef)  se(coef)      z      p
## age           0.003321  1.003326  0.005463  0.608  0.5433
## as_factor(eye)right 0.350484  1.419754  0.162537  2.156  0.0311
## trt           -0.812522  0.443738  0.169412 -4.796 1.62e-06
##
## Likelihood ratio test=27.59  on 3 df, p=4.421e-06
## n= 394, number of events= 155
```

- Note that treatment is randomized, so we *shouldn't* see large changes in the coefficient on treatment when we add covariates.
  - More on this next week!



- ▶ Note that all these methods have assumptions that can sometimes be important
- ▶ Given our time, I am purposefully just showing you the basics
  - ▶ If any of these specific methods interest you, I suggest doing more in-depth readings. I can provide some suggestions.

- ▶ Deserranno, E., Stryjan, M., & Sulaiman, M. (2019). Leader selection and service delivery in community groups: Experimental evidence from uganda. *American Economic Journal: Applied Economics*, 11(4), 240-267.
- ▶ Fischer, T., Frölich, M., & Landmann, A. (2023). Adverse Selection in Low-Income Health Insurance Markets: Evidence from an RCT in Pakistan. *American Economic Journal: Applied Economics*, 15(3), 313-340.