Longitudinal Data Analysis Using R

Stephen Vaisey

2021-10-29

Start here!

```r
# make sure you have run the following code at least once
install.packages(c(
  "remotes",              # to be able to install from github
  "tidyverse",            # dplyr and ggplot
  "panelr",               # panel data made easier; wbm() function
  "lmtest",               # robust SEs
  "broom.mixed",          # extracting info from mixed models
  "ggeffects",            # calculating effects
  "huxtable",             # making regression tables
  "patchwork",            # organizing multiple plots
  "here",                 # for setting sensible file paths in projects
  "skimr",                # for panelr summary statistics
  "plm",                  # for two-way fixed effects
  "clubSandwich",         # additional robust SEs
  "jtools",               # for theme_nice()
  "modelsummary",         # summarizing models (mostly plotting coefficients)
  "car",                  # for linearHypothesis (testing joint significance)
  "optimx",               # additional (g)lmer optimizers
  "dfoptim"               # additional (g)lmer optimizers
))
remotes::install_github("jacob-long/dpm")                       # dynamic panel models
remotes::install_github("vincentarelbundock/marginaleffects")   # marginal effects
```

The big picture

What is longitudinal data?

- Repeated measures of the same units over time.
- Individuals measured several times is the most common scenario. But it could be any entity, like states or firms.
- In general, these methods are designed for situations where we have many more *units* than we have *time periods* (i.e., $N \gg T$), though there is no bright line between this and other possible scenarios.

The bottom line

There are two basic things you can do with longitudinal or panel data.

1. You can model the trajectory of an outcome in groups or individuals over time.
2. You can use repeated measures information to make better causal inferences.

In most cases, however, you will *not* be able to do both (at least not fully).

Data and research questions

The structure of the data determines what you *can* do

Your research question determines what you *should* do!

What do we want to know?

I assume in this course that you're interested in estimating a **treatment effect** of some kind. That is, you want to know how some X-variable *causally* affects the outcome. The language of "treatment" comes from experiments but it is used much more broadly these days.

The simplest treatment effect to understand is the **average treatment effect** or *ATE*. This is an estimate of the average *causal* difference the treatment would make to an outcome for a member of the target population.

For example, we might ask how much more college graduates make than non-college graduates *because* of their degree.

See Lundberg et al. 2021. "What Is Your Estimand? Defining the Target Quantity Connects Statistical Evidence to Theory". *American Sociological Review*.

Defining treatment effects (ATE)

Assume a binary treatment like *union membership* and an outcome like wages. We define the **average treatment effect** as:

$$\text{ATE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i^1 - Y_i^0)$$

n is size of the population;

Y is *wages* ;

Y_i^1 is either (a) person i's wages if they belong to a union or (b) what person i's wages *would be* if they *did* belong to a union;

Y_i^0 is either (a) person i's wages if they *don't* belong to a union or (b) what person i's wages *would be* if they *didn't* belong to a union

When ATE doesn't make sense

Sometimes it won't make sense to think of an average difference in a whole population. For example, it doesn't make sense for *all* adults in a society (including, say, teachers and engineers) to participate in a government job training program intended for low-income workers.

Defining treatment effects (ATT)

In cases like that (and others), we often are interested in the **average treatment effect on the treated**. This is the difference the treatment made to *those who received it*.

$$\text{ATT} = \frac{1}{n}\sum_{i=1}^{n}\{Y_i(\text{treated}) - Y_i^0(\text{treated})\}$$

Note here that Y_i is the *observed* value of the outcome because treated cases were, in fact, treated. So we need some way of calculating the *counterfactual untreated value* (Y^0) for the treated cases.

This topic can be complicated but we won't go too much deeper than this. But this should provide a background for what we are usually doing: trying to estimate the *effect* some cause has on some outcome for some population using panel data.

Panel data and inference

How does panel data help with causal inference. It depends!

Y can change?	X can change?	Options
Yes	No	trajectories
Yes	Yes	trajectories *or* better causal inference
No	Either	not really panel data!

It's not *quite* this simple but this is a useful heuristic. If we have experimental data, for example, we can get trajectories *and* causal inference.

Remember

The structure of the data determines what you ***can*** do

Your research question determines what you ***should*** do!

Thinking about your **research question** will make your life so much easier!

Panel data in practice

Panel data comes in one of two forms:

1. Long form, where each row is a unit-observation and most units will be represented by multiple rows.
2. Wide form, where each unit is represented in one row and repeated measures are encoded in several variables.

Example: Wage data

Let's say we follow a sample of adult workers and interview them once a year for three years. Each time, we collect three pieces of information:

1. Log yearly earnings
2. Number of weeks worked in the previous year
3. Years of education completed (asked only the first time; *Why?*)

Let's look at two of these respondents' data in two different forms.

Wage data: long form

id	lwage	t	wks	ed
1	5.56068	1	32	9
1	5.72031	2	43	9
1	5.99645	3	40	9
2	6.16331	1	34	11
2	6.21461	2	27	11
2	6.26340	3	33	11

Wage data: wide form

id	ed	lwage_1	lwage_2	lwage_3	wks_1	wks_2	wks_3
1	9	5.56068	5.72031	5.99645	32	43	40
2	11	6.16331	6.21461	6.26340	34	27	33

It's easy to see here that `lwage` and `wks` are **time-varying** within units, whereas `ed` is **time-constant** within units. This is an important distinction!

The same information

Both formats contain the *same information*. But we will almost always use long form because it can store data more compactly in the following situations:

- unbalanced data (different units have different #s of obs)
- missing data (some whole observations are missing)
- unequally spaced data (e.g., date of patient visit)

Wide and long in R

We can reshape the data frame in two main ways:

- `dpylr`'s `pivot_wider()` and `pivot_longer()`
- `panelr`, by Jacob Long, has `widen_panel()` and `long_panel()`, which are even easier

```r
data("teen_poverty", package = "panelr")  # get dataset from R package
glimpse(teen_poverty)                     # get an overview
```

```
## Rows: 1,151
## Columns: 28
## $ id        <dbl> 22, 75, 92, 96, 141, 161, 220, 229, 236, 240, 245, 249, 255,~
## $ pov1      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, ~
## $ mother1   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, ~
## $ spouse1   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ inschool1 <dbl> 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, ~
## $ hours1    <dbl> 21, 8, 30, 19, 0, 0, 6, 0, 0, 18, 0, 0, 0, 12, 0, 19, 25, 20~
## $ pov2      <dbl> 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, ~
## $ mother2   <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, ~
## $ spouse2   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ~
## $ inschool2 <dbl> 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, ~
## $ hours2    <dbl> 15, 0, 27, 54, 6, 15, 8, 32, 20, 0, 0, 0, 23, 0, 0, 20, 30, ~
## $ pov3      <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, ~
## $ mother3   <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, ~
## $ spouse3   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ~
## $ inschool3 <dbl> 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, ~
## $ hours3    <dbl> 3, 0, 24, 0, 0, 37, 6, 0, 0, 0, 0, 30, 23, 0, 0, 0, 20, 55, ~
## $ pov4      <dbl> 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, ~
## $ mother4   <dbl> 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, ~
## $ spouse4   <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, ~
## $ inschool4 <dbl> 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, ~
## $ hours4    <dbl> 0, 4, 31, 26, 0, 0, 12, 15, 40, 85, 0, 0, 58, 0, 0, 27, 38, ~
## $ age       <dbl> 16, 17, 16, 17, 16, 17, 17, 16, 17, 16, 16, 16, 16, 17, 16, ~
## $ black     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, ~
## $ pov5      <dbl> 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ~
## $ mother5   <dbl> 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, ~
## $ spouse5   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, ~
## $ inschool5 <dbl> 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, ~
## $ hours5    <dbl> 0, 0, 0, 36, 0, 0, 0, 20, 40, 90, 27, 0, 25, 12, 0, 46, 0, 2~
```

```r
teen_pov <- teen_poverty |>
  long_panel(
    id = "id",        # the name of the existing ID variable
    wave = "t",       # the name you want for the new time variable
    begin = 1,        # the indicator of the first period
    end = 5           # the indicator of the last period
  )
glimpse(teen_pov)
```

```
## Rows: 5,755
## Columns: 9
## Groups: id [1,151]
## $ id       <fct> 22, 22, 22, 22, 22, 75, 75, 75, 75, 75, 92, 92, 92, 92, 92, 9~
## $ t        <dbl> 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1~
## $ age      <dbl> 16, 16, 16, 16, 16, 17, 17, 17, 17, 17, 16, 16, 16, 16, 16, 1~
## $ black    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ pov      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0~
## $ mother   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0~
## $ spouse   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ inschool <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1~
## $ hours    <dbl> 21, 15, 3, 0, 0, 8, 0, 0, 4, 0, 30, 27, 24, 31, 0, 19, 54, 0,~
```

Pros and cons of `panelr`

Pros

- easy reshaping from wide to long
- convenience functions for complex models (e.g., between-within, CRE, asymmetric FE, dynamic models)
- easy descriptive visualization with `line_plot()`
- better summary statistics using `skimr`

Cons

- `panel_data`-class objects don't always play well with outside functions*
- convenience functions can allow you to estimate models you don't really understand

* If you get an odd error using a panel data object, you can always coerce it to a vanilla data frame by wrapping it in `as.data.frame()`.

Panel possibilities

There are five basic situations we will consider. Each provides different possibilities and challenges.

- When X doesn't change
- When X changes once (for some or for everyone)
- When X changes at different times, in the same direction
- When X changes at different times, in any direction
- As above, plus X is determined dynamically

Outline of the course

- Two types of variance
- When X doesn't change
  - mixed models and growth curves
- When X changes once
  - pre/post
  - difference in differences
- When X changes at different times, in the same direction
  - two-way fixed effects
  - difference in differences for staggered treatments*
- When X changes at different times, in any direction
  - two-way fixed effects (again)
  - mixed models (again)
  - within-between and correlated random effects
- Dynamic treatments
  - dynamic panel SEM*

* indicates very limited coverage of a complex topic that would need its own course

Two types of variance

Earnings data

```
glimpse(WageData)
```

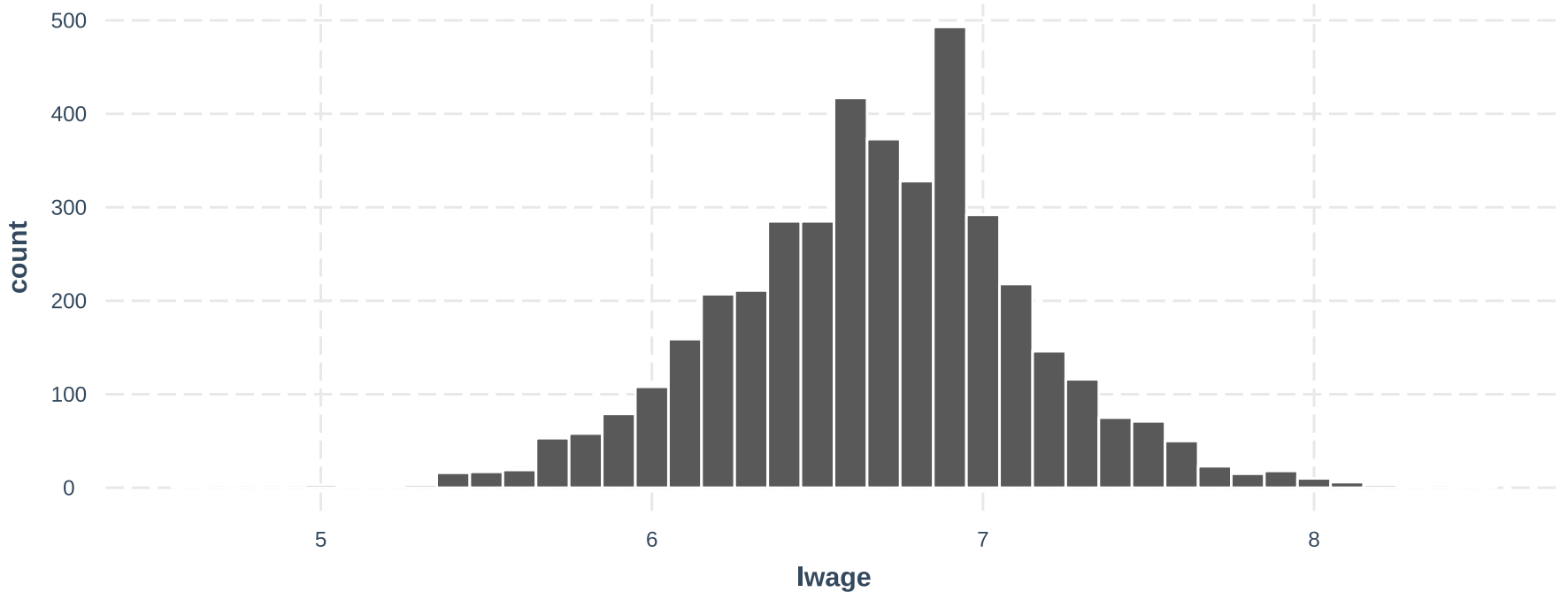
```
## Rows: 4,165
## Columns: 14
## $ exp    <dbl> 3, 4, 5, 6, 7, 8, 9, 30, 31, 32, 33, 34, 35, 36, 6, 7, 8, 9, 10,~
## $ wks    <dbl> 32, 43, 40, 39, 42, 35, 32, 34, 27, 33, 30, 30, 37, 30, 50, 51, ~
## $ occ    <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ ind    <dbl> 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0~
## $ south  <dbl> 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ smsa   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1~
## $ ms     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0~
## $ fem    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1~
## $ union  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0~
## $ ed     <dbl> 9, 9, 9, 9, 9, 9, 9, 11, 11, 11, 11, 11, 11, 11, 12, 12, 12, 12,~
## $ blk    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1~
## $ lwage  <dbl> 5.56068, 5.72031, 5.99645, 5.99645, 6.06146, 6.17379, 6.24417, 6~
## $ t      <dbl> 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 1~
## $ id     <dbl> 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4~
```

Two types of variance

- The outcome is *log wage*
- This outcome variable is measured annually for 7 years
- Because people have different average earnings and those earnings change over time, log wage has both **between-person variance** and **within-person variance**
- Sometimes this is called **between-subject** and **within-subject** variance
- All outcome variables in panel data are like this

Always look at the distribution

It always pays to visualize the data!



Intraclass correlation (ICC)

How much of the variance is within and how much is between? We can get a simple answer to this question by examining the **intraclass correlation**, defined as:

$$\text{ICC} = \frac{\tau^2}{\tau^2 + \sigma^2}$$

where τ^2 is the between-person variance and σ^2 is the within-person variance.

Calculating ICC

```
# BETWEEN VARIANCE
b_var <- WageData |>
  group_by(id) |>                                  # do calculations separately for ids
  mutate(mean_lwage = mean(lwage)) |>              # get each id's mean wage
  slice(1) |>                                      # keep one row of each id
  ungroup() |>                                     # do calculations on whole data frame
  summarize(b_var = var(mean_lwage)) |>            # get variance of means
  as.numeric()                                     # output as a number
b_var
```

```
## [1] 0.1554242
```

```
# WITHIN VARIANCE
w_var <- WageData |>
  group_by(id) |>                                  # do calculations separately for ids
  mutate(dev_lwage = lwage - mean(lwage)) |>       # create time devs from each id's mean
  ungroup() |>                                     # calcs on whole data frame
  summarize(w_var = var(dev_lwage)) |>             # get variance of deviations
  as.numeric()                                     # output as number
w_var
```

```
## [1] 0.05779328
```

```
b_var / (b_var + w_var)                            # the intraclass correlation
```

```
## [1] 0.7289468
```

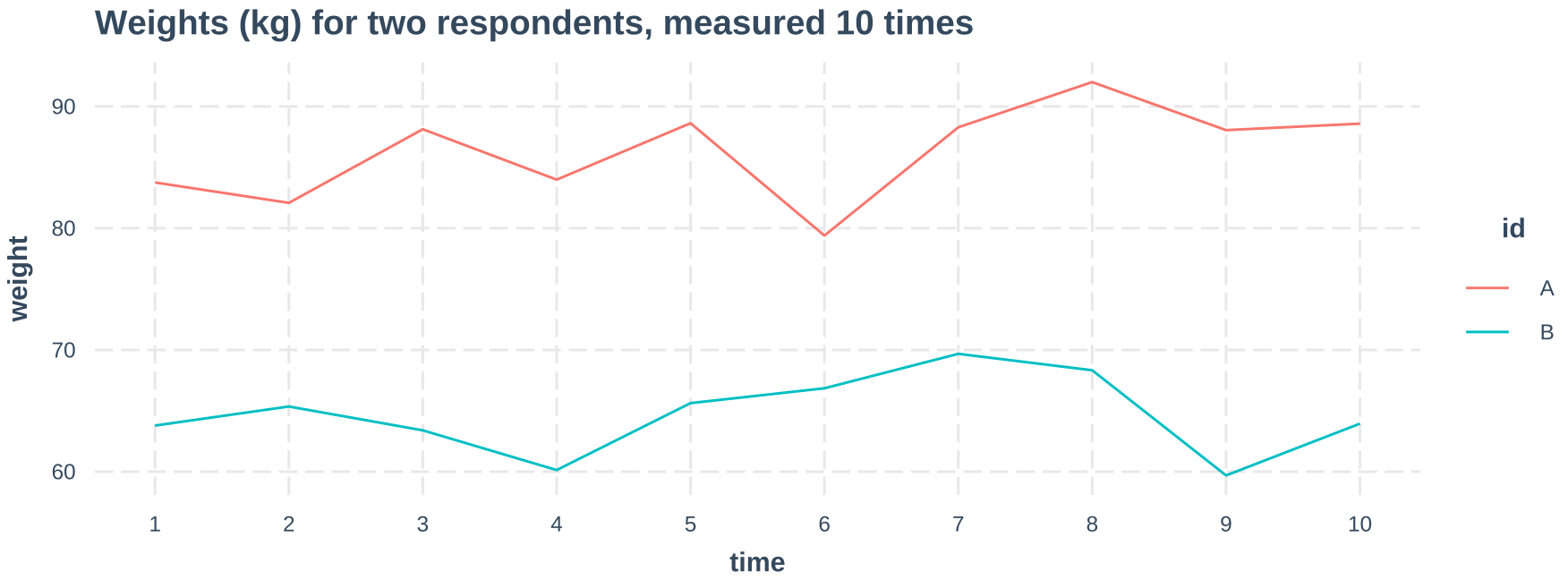
Interpreting ICC

We can say that 73 percent of the variance is **between persons** and the rest is **within persons**. So these workers' earnings fluctuate over time; sometimes a person earns more or less than they usually do. But most of the differences are between *persons*, some who generally earn more (or less) than others.

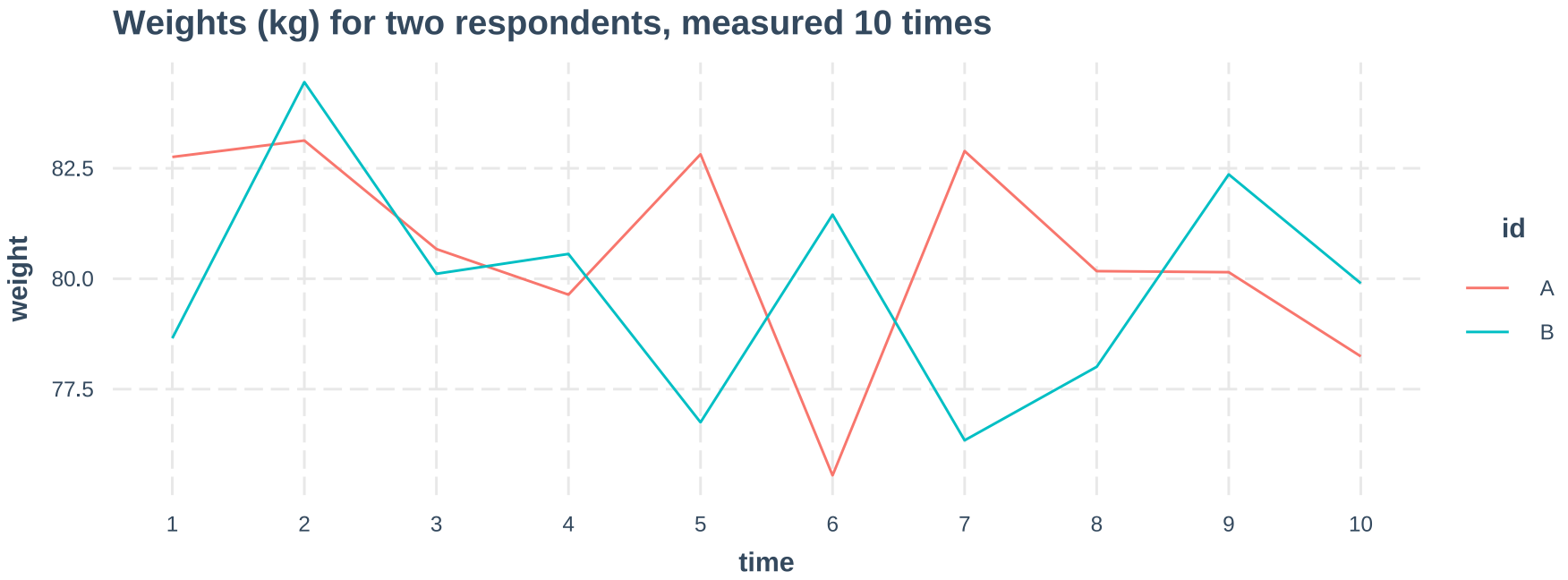
For intuition, think of how weight fluctuates. Some days/months/years I weigh more, some days/months/years I weight less. That's variance. But most of the variance in weight is between persons, who weigh different amounts on average.

The ICC is simply another **descriptive statistic** that helps you understand your data. It's always useful to know where the variance is.

Visualizing high ICC

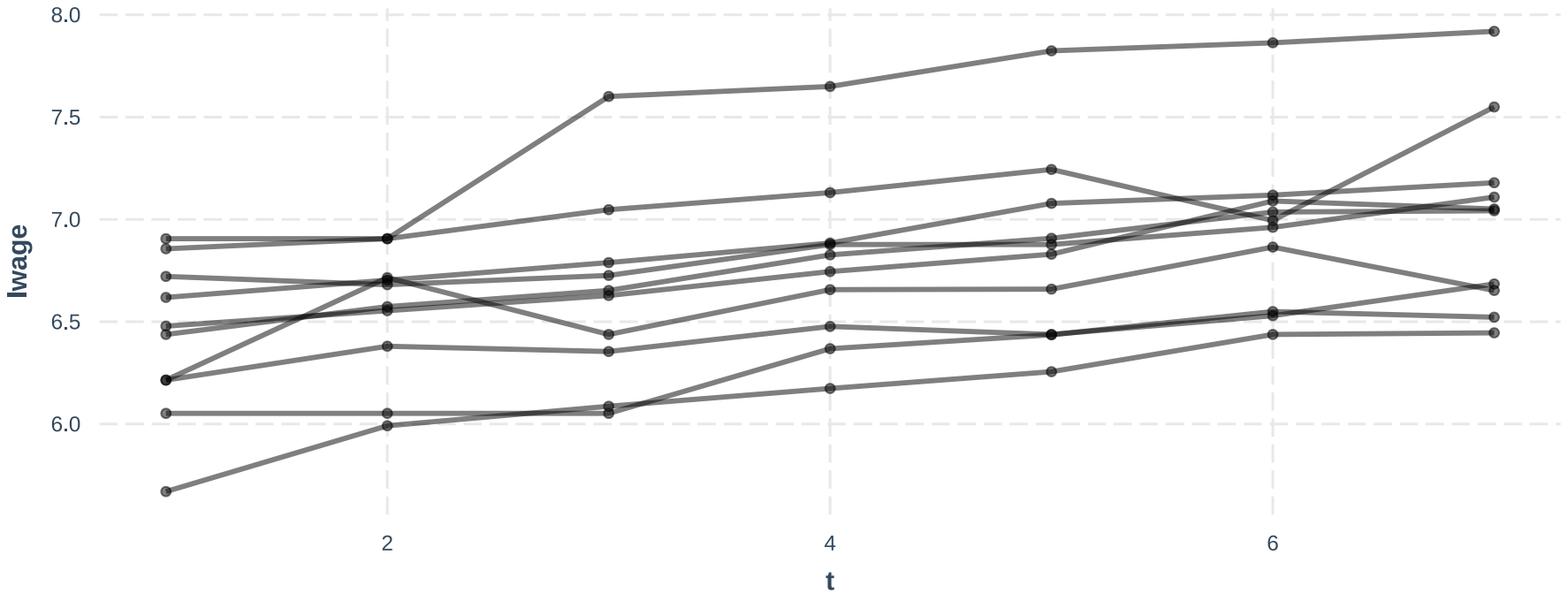


Visualizing low ICC



An easy visualization

```
set.seed(12345)
line_plot(WageData, lwage, id = "id", wave = "t", subset.ids = TRUE, n.random.subset = 10)
```



Two types of variables

Type	Description	Examples
Time-constant	Variables that only vary between units	Birth country, "race"
Time-varying	Variables measured over time but that almost always have both within and between variance	earnings, satisfaction

Checking the wage data

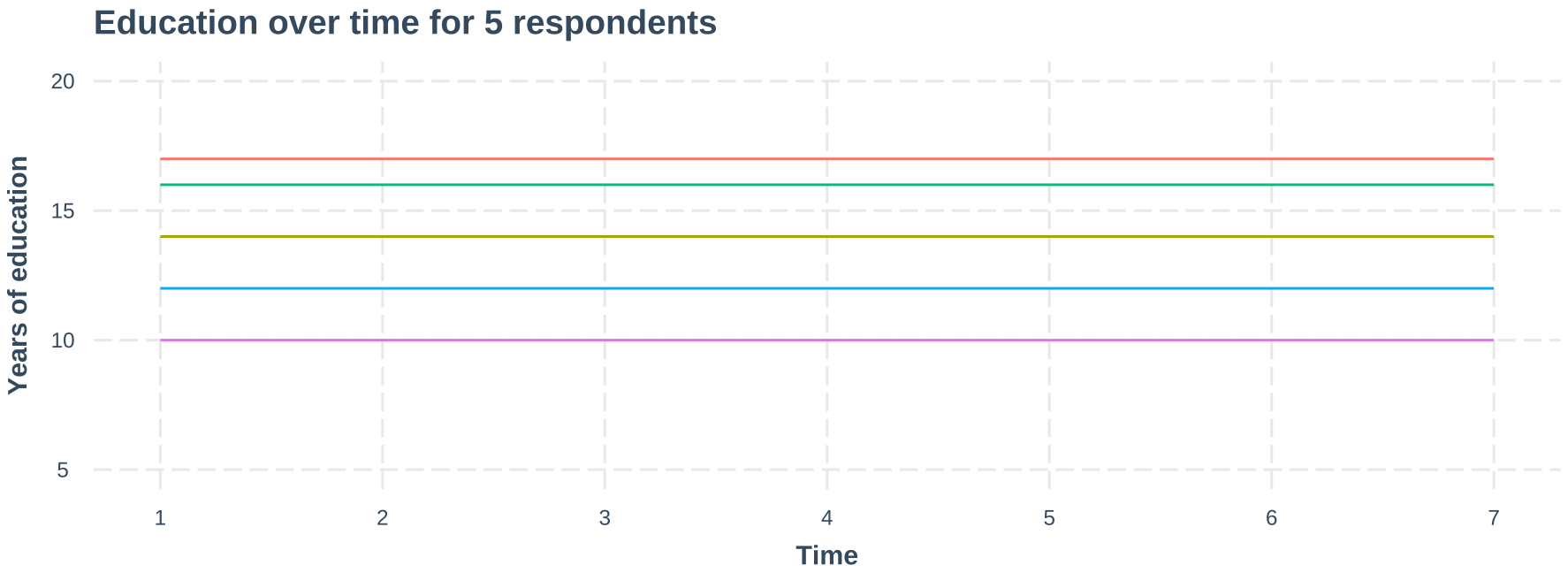
We could check manually using `dplyr` or we can use convenience functions in the `panelr` package. The `panel_data()` function adds panel metadata to the dataframe; the `are_varying()` function checks each variable to see whether it has both sorts of variance.

```
d <- panel_data(WageData,     # define d as panel version of wage data
                id = id,      # id variable
                wave = t)     # time variable
are_varying(d)                # check if time-varying
```

```
##   exp   wks   occ   ind south  smsa    ms   fem union    ed   blk lwage
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE
```

We see that `fem`, `ed`, and `blk` have only between-person variance. The others have both types.

Visualizing ed over time



Here the ICC is 100% because there is *no* within-person variation!

When X doesn't change

Starting simple: linear regression

Let's say we wanted to know the effect of a college degree on wages. We can use the ed variable to define those who do (not) have a college degree. Then we can estimate a regression model.

```
d <- d |>
  mutate(college = if_else(ed >= 16, 1L, 0L))
linreg <- lm(lwage ~ college, data = d)
huxreg("Linear regression" = linreg,
       stars = NULL,
       error_pos = "right",
       statistics = c("Resid. SD" = "sigma"),
       coefs = c("Intercept" = "(Intercept)",
                 "College" = "college"))
```

	Linear regression	
Intercept	6.580	(0.008)
College	0.353	(0.015)
Resid. SD	0.434	

This says that the effect of a college degree on wages is to increase them by $e^{.353} = 42\%$.

What's wrong with this model?

The limits of linear regression

We can write the model we just estimated like this:

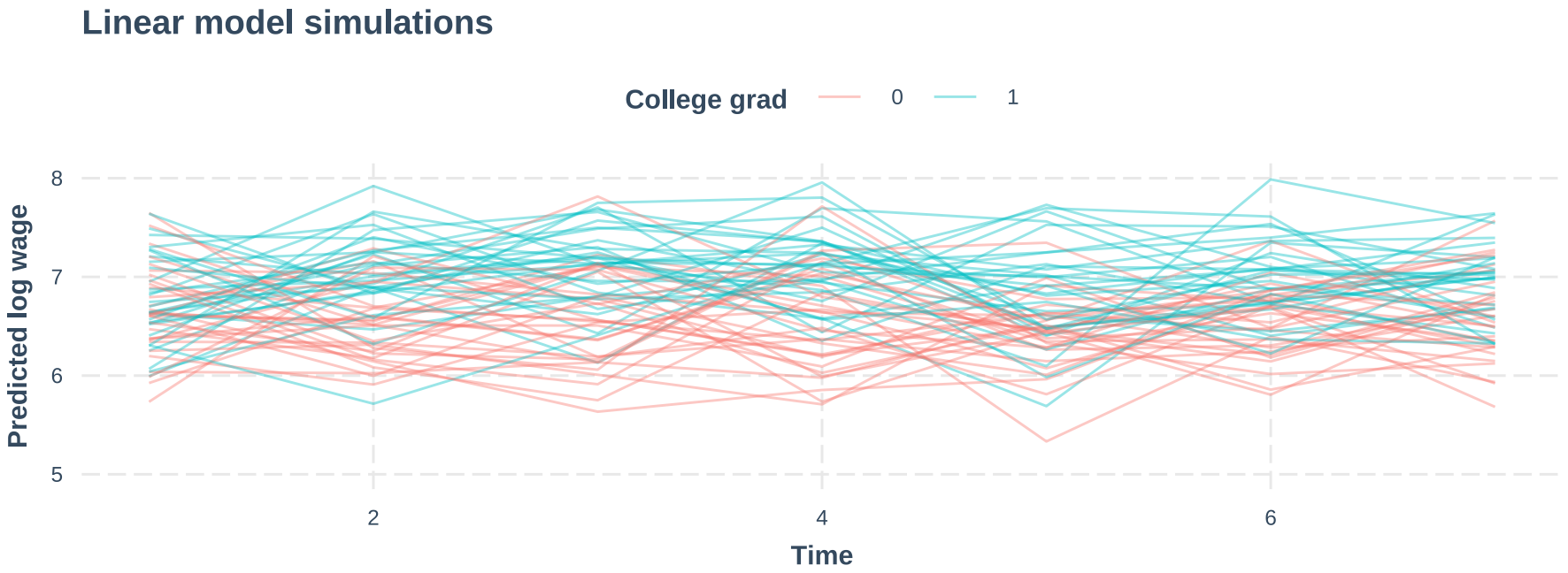
$$\text{lwage}_{it} = \beta_0 + \beta_1 \text{college}_i + \epsilon_{it}$$

This model has three main limitations when used on panel data:

1. It doesn't allow there to be any individual-specific differences in `lwage` except `college`. There are no other terms subscripted with just i.
2. It assumes all fluctuations around the average ($\beta_0 + \beta_1 \text{college}_i$) are simply random (ϵ_{it}).
3. Like any regression model, it assumes independent observations. The model doesn't "know" that 7 observations come from each person. So the standard errors will be too small.

Also, like all such regression models, it only identifies the *treatment effect* of college if there are *no omitted variables*. That's clearly not realistic here!

Visualizing the model



What's wrong with this picture? (No time trends, too much variance...)

What's the solution?

Mixed model

The basic mixed model is just a linear regression with **two error terms**: one for the "level 2 units" (here, persons) and one for the "level 1 units" (here, observations).

$$\text{lwage}_{it} = \beta_0 + \beta_1 \text{college}_i + \alpha_i + \epsilon_{it}$$

We assume α_i is normally distributed with variance equal to τ^2 and is independent of all other terms on the right-hand-side.

With α_i in the model, the ϵ_{it} are assumed to be **conditionally independent** of each other. That is, once we know *which person* the data came from (and adjust for model covariates), the observation-level errors are assumed to be independent.

The easiest way to estimate mixed models in R is using the `lme4` package, which contains `lmer()` (for linear models) and `glmer()` (for non-linear models). The syntax is very similar to `stats::lm()`.

Estimating the model

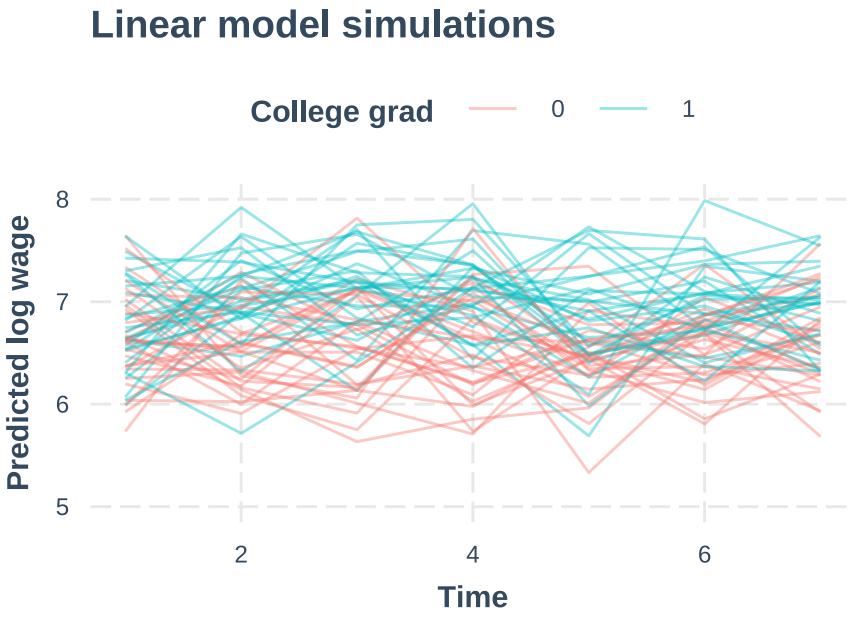
```
mixreg <- lmer(lwage ~ college + (1 | id), # formula
               data = d,                   # data frame
               REML = FALSE)               # ask for ML (not restricted ML)
```

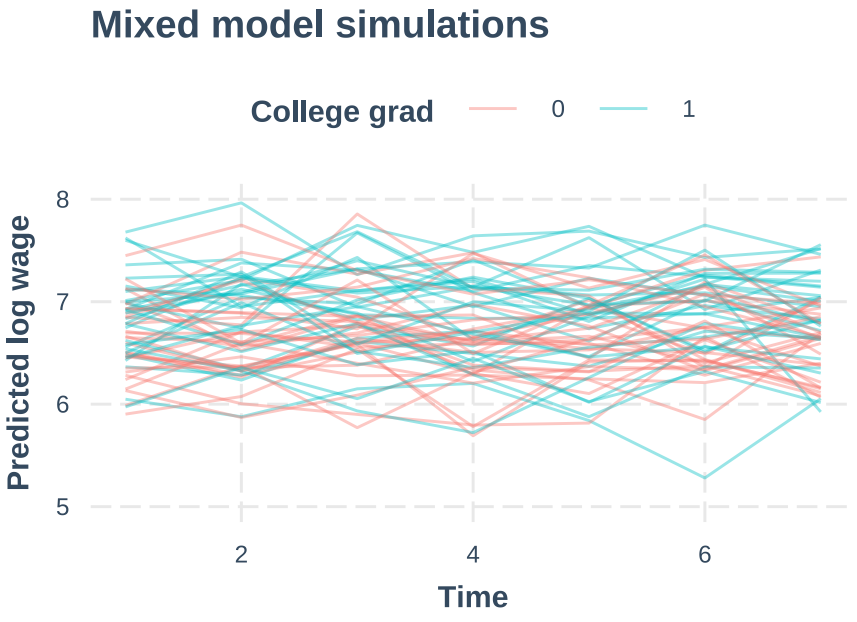
The `+ (1 | id)` adds the error term at the person level. `REML=FALSE` asks R to use maximum likelihood to estimate the model. Let's compare results:

	Linear		Mixed	
Intercept	6.580	(0.008)	6.580	(0.017)
College	0.353	(0.015)	0.353	(0.033)
Resid. SD	0.434		0.260	

The estimates are the same but the standard errors are over twice as large. The residual SD (σ) is also smaller. Why?

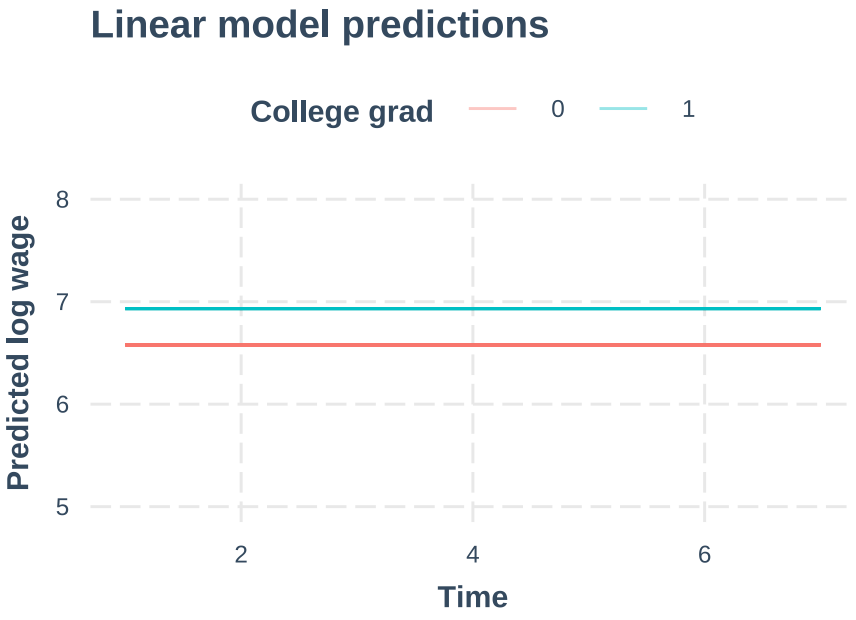
Visualizing model simulations

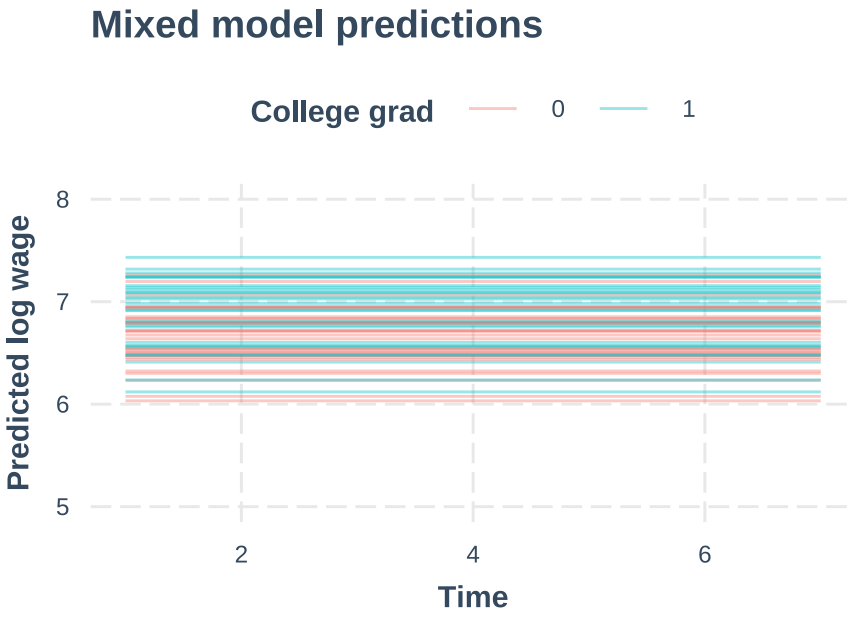




It's hard to see but the two groups are more mixed because individuals are allowed to have different averages for reasons not having to do with college. The individual lines are also less "bouncy" because the residual variance is smaller.

Visualizing model predictions





The mixed model allows each respondent to have their own intercept in addition to the explicit covariates in the model (here, `college`)

About time

We have so far ignored *time*. Since each of our 7 observations per person reflects the passage of one year, we probably want to allow for changes in average earnings over time. Most people earn more as they age (up to a point).

The mixed model we estimated above assumes the conditional independence of the ϵ_{it}. This means that, once we know about *the person* the observation comes from ($\beta_1 \text{college}_i + \alpha_i$), all that person's additional fluctuations over time are assumed to be random. But if wages are *increasing* (or decreasing), this will not be true; observations closer in time will be more similar.

The simplest way to deal with this is to model the passage of time. And the simplest way to do *that* is to allow a linear time trend.

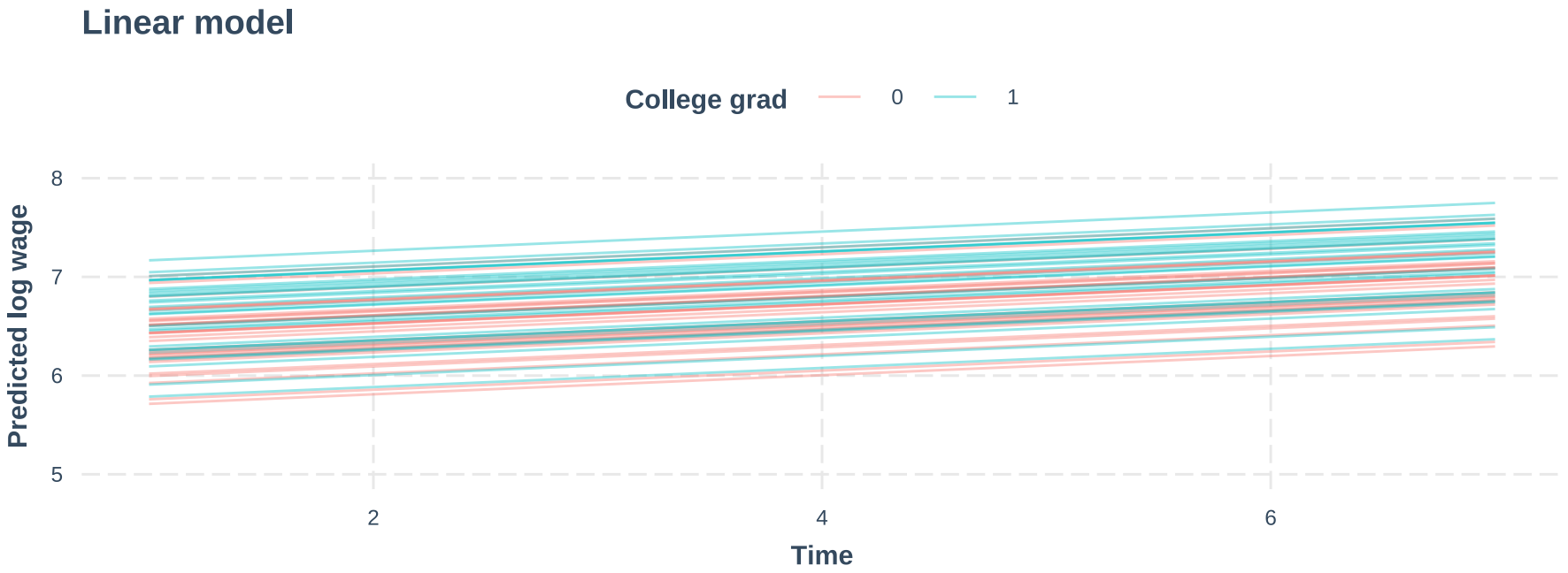
Adding linear time

```
d <- d |> mutate(t0 = t - 1) # start time at 0
mod1 <- lmer(lwage ~ college + t0 + (1 | id),
             data = d,
             REML = FALSE)
huxreg(
  "Linear time" = mod1,
  stars = NULL,
  error_pos = "right",
  statistics = "",
  coefs = c(
    "Intercept" = "(Intercept)",
    "College" = "college",
    "Time" = "t0"
  )
)
```

	Linear time	
Intercept	6.289	(0.018)
College	0.353	(0.033)
Time	0.097	(0.001)

The college/non-college difference is still $e^{.353}$ = 42%. And each year, we expect wages to increase to increase by $e^{.097}$ = 10%.

Visualizing model predictions



All these lines are parallel because that's what we asked the model to do!

Four ways to model time

1. The passage of time affects everyone the same (what we just did)
2. The passage of time affects everyone the same in the same treatment group (e.g., college vs. non-college)
3. Each individual gets their own time trend (this is a latent growth curve model)
4. A combination of (2) and (3)

Selecting a model

```
# NOTE: update() allows you to change something about a model, leaving the rest the same
mod2 <- update(mod1, formula = lwage ~ college * t0 + (1      | id ))
mod3 <- update(mod1, formula = lwage ~ college + t0 + (1 + t0 | id ))
mod4 <- update(mod1, formula = lwage ~ college * t0 + (1 + t0 | id ))
BIC(mod1, mod2, mod3, mod4) |> format(scientific = FALSE)
```

df	BIC
5	-1574.123
6	-1614.956
7	-1930.822
8	-1944.909

The lowest BIC indicates `mod4` is the best fit, suggesting that college and non-college workers have different average trajectories *and* that there is substantial individual heterogeneity beyond that.

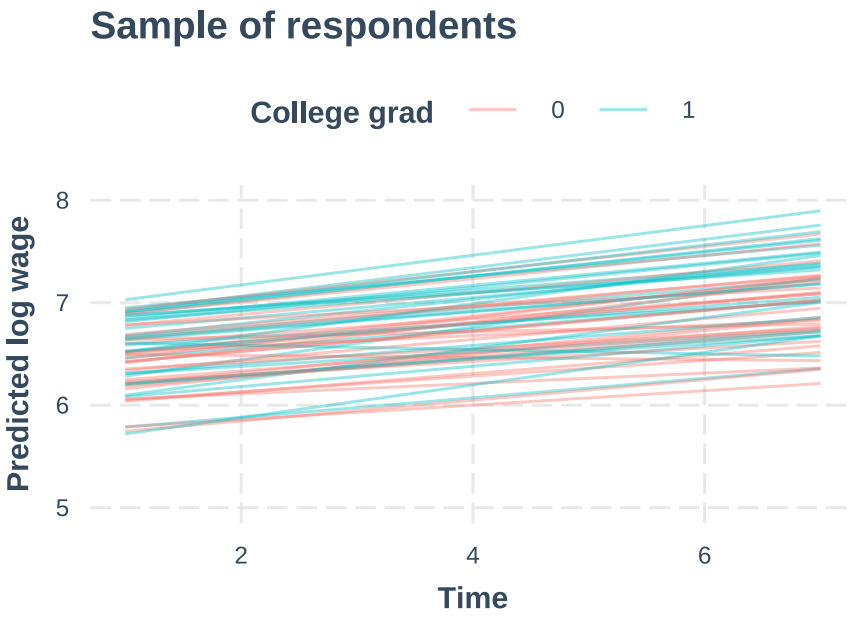
A closer look at `mod4`

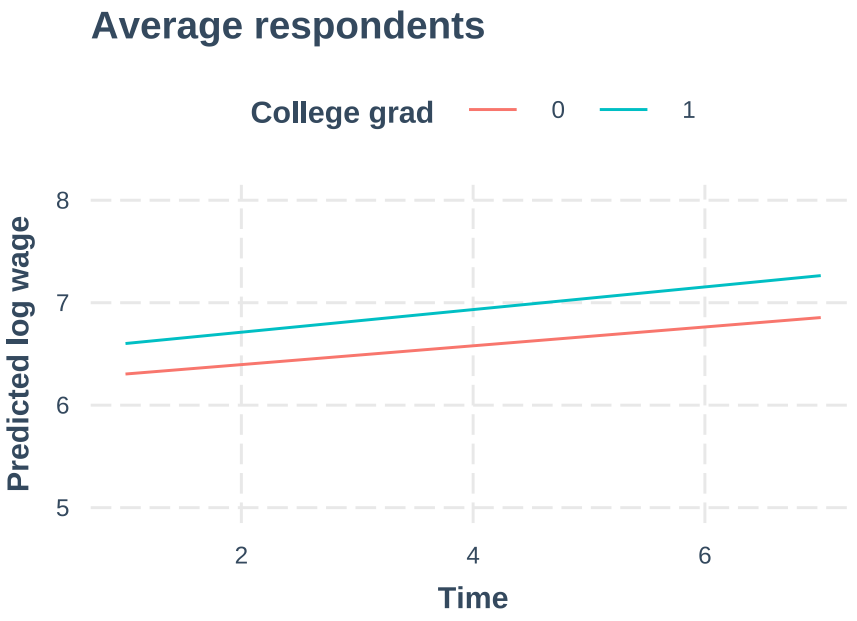
	Model 4	
Intercept	6.304	(0.017)
College	0.298	(0.033)
Time	0.092	(0.002)
College*Time	0.019	(0.004)
N	4165	

We can say a couple of different things here:

- The *initial* college/non-college difference in log wage at time 0 was .298 (35%). Each year, that gap grew by .019. Thus, at the final time period, the college non-college gap was .412 (51%).
- We could also say that log wages for non-college workers increased by .092 (9.6%) per year, whereas log wages for college-educated workers increased by .092 + .019 = .111 (12%) per year.

Visualizing model predictions





Time can be non-linear

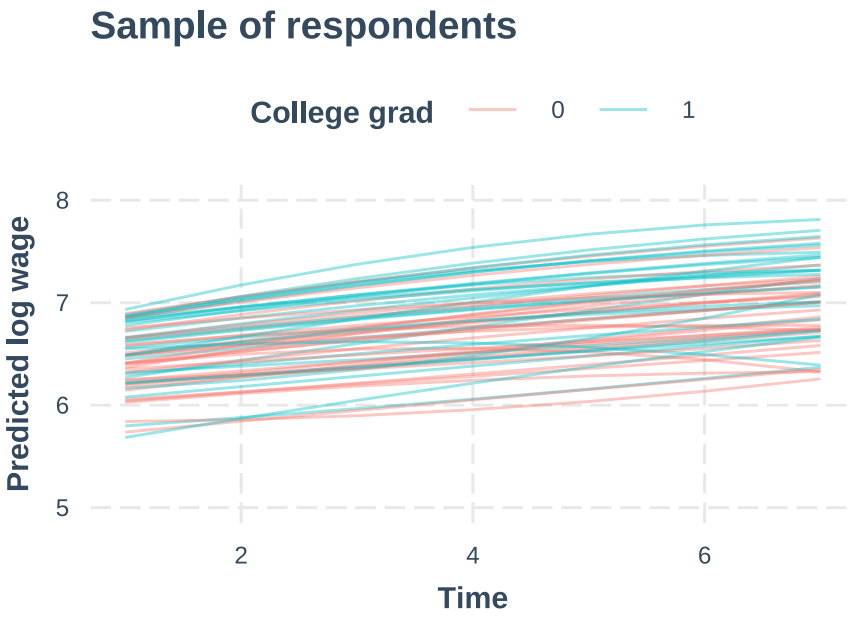
We have assumed the relationship between the outcome and time is linear (with respect to log wages) but we don't need to do that. We could instead use a quadratic term to represent time.

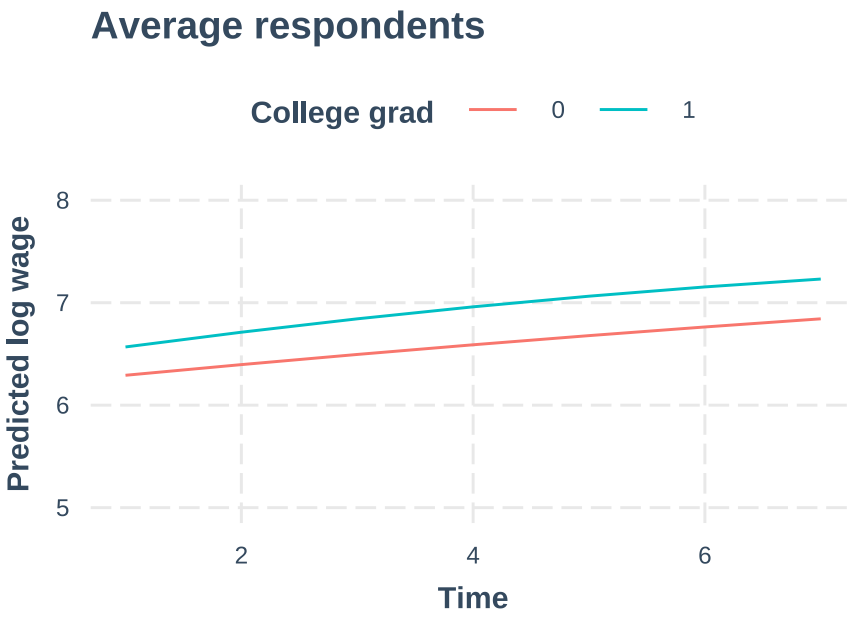
```
mod5 <- update(mod1, lwage ~ college * ( t0 + I(t0^2) ) + (1 + t0 + I(t0^2) | id ))
BIC(mod4, mod5) |> format(scientific = FALSE)
```

df	BIC
8	-1944.909
13	-2008.506

The lower BIC means a better fit so we'd prefer the more complicated model here. And since we're talking about a quadratic relationship and a logged outcome, it's indeed complicated!

Visualizing model predictions



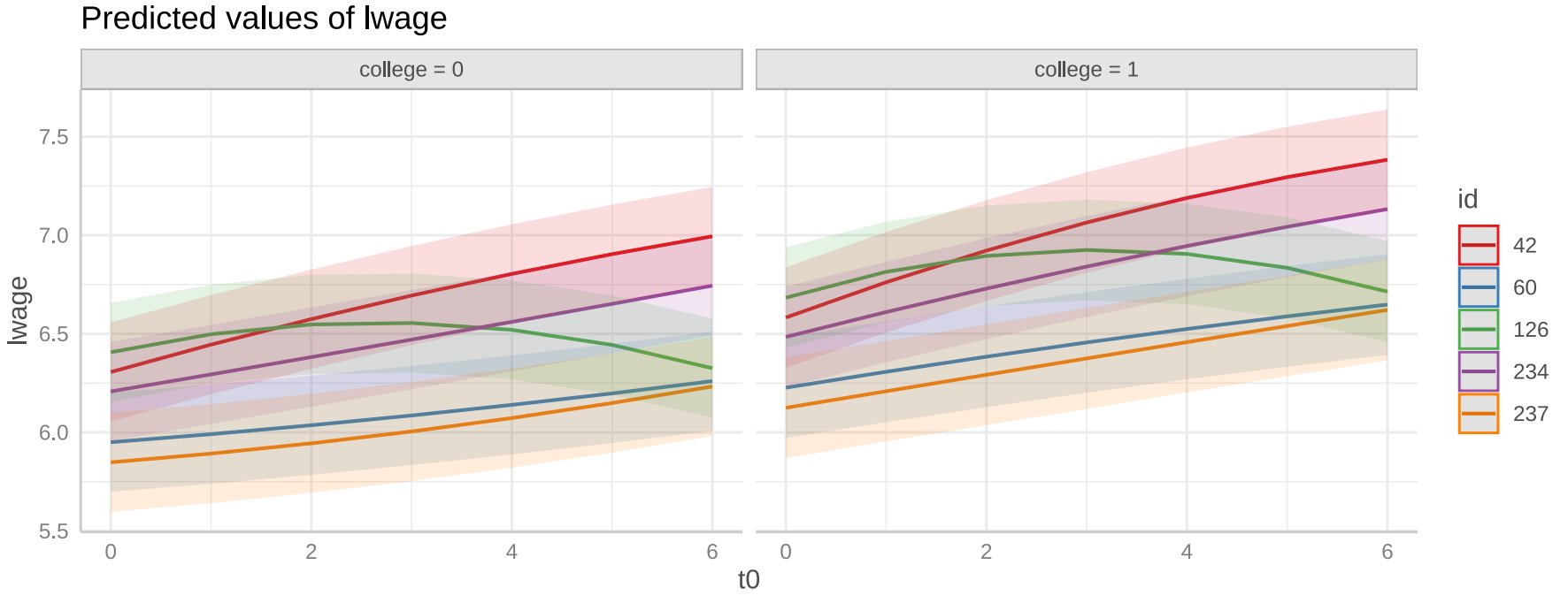


You can see that some of these individual trajectories go up then down again.

Visualization tools

I have been coding all graphs by hand for maximum control. But there are easier tools to try, like `ggpredict()` from the `ggeffects` package.

```
set.seed(123456)
ggpredict(mod5, terms = c("t0 [all]", "id [sample = 5]", "college"), type = "re") |> plot()
```



Adding adjustment variables

We can add covariates to the model in the usual way. For example:

```
mod6 <- update(mod5, lwage ~ college * (t0 + I(t0^2)) + occ + ind + south +
                 smsa + fem + union + blk + (1 + t0 + I(t0^2) | id),
               control = lmerControl(optimizer ="Nelder_Mead"))
```

Side note: the default optimizer for `lmer()` is BOBYQA*. You can change to Nelder-Mead if you're having trouble with optimization. In general, scaling your variables so that they are on the same order of magnitude helps as well.

*BOBYQA stands for *bound optimization by quadratic approximation*, an algorithm by Michael J. D. Powell.

term	estimate	std.error
(Intercept)	6.35	0.0234
college	0.233	0.0301
t0	0.106	0.00534
I(t0^2)	-0.00231	0.000807
occ	-0.0522	0.0127
ind	0.0199	0.0135
south	-0.0935	0.0224
smsa	0.0628	0.0166
fem	-0.415	0.0398
union	0.0384	0.0132
blk	-0.142	0.0487
college:t0	0.0445	0.0102
college:I(t0^2)	-0.00424	0.00154

These beta coefficients could all be interpreted in the usual way: the expected difference in the outcome when X is one unit higher.

However, we are only including them in an attempt to identify the *treatment effect* of college on log wages. That's our research question. So we have no need to interpret these additional coefficients.*

If we have adjusted properly for all variables that confound the causal relationship between college and wages (which is unlikely!) then (and only then) we have identified the ATE.

* See Hünermund and Louw. 2020. "On the Nuisance of Control Variables in Regression Analysis."

Limited dependent variables

Not all outcomes can be treated as continuous. We can augment binary, count, etc. models in exactly the same way, by adding the α_i term to capture individual differences. For a binary outcome, Y, an example equation would be:

$$\log\left(\frac{P(y_{it}=1)}{1-P(y_{it}=1)}\right) = \beta_0 + \beta_1 \text{college}_i + \gamma t + \alpha_i$$

As with the linear model, we assume α_i is normally distributed with variance equal to τ^2 and is independent of all other terms on the right-hand-side.

College degree is (once again) a time-constant predictor (note the absence of a t subscript). And we have allowed the "effect" of time to be linear (γt), though we could change that if we wanted. This is just one example model.

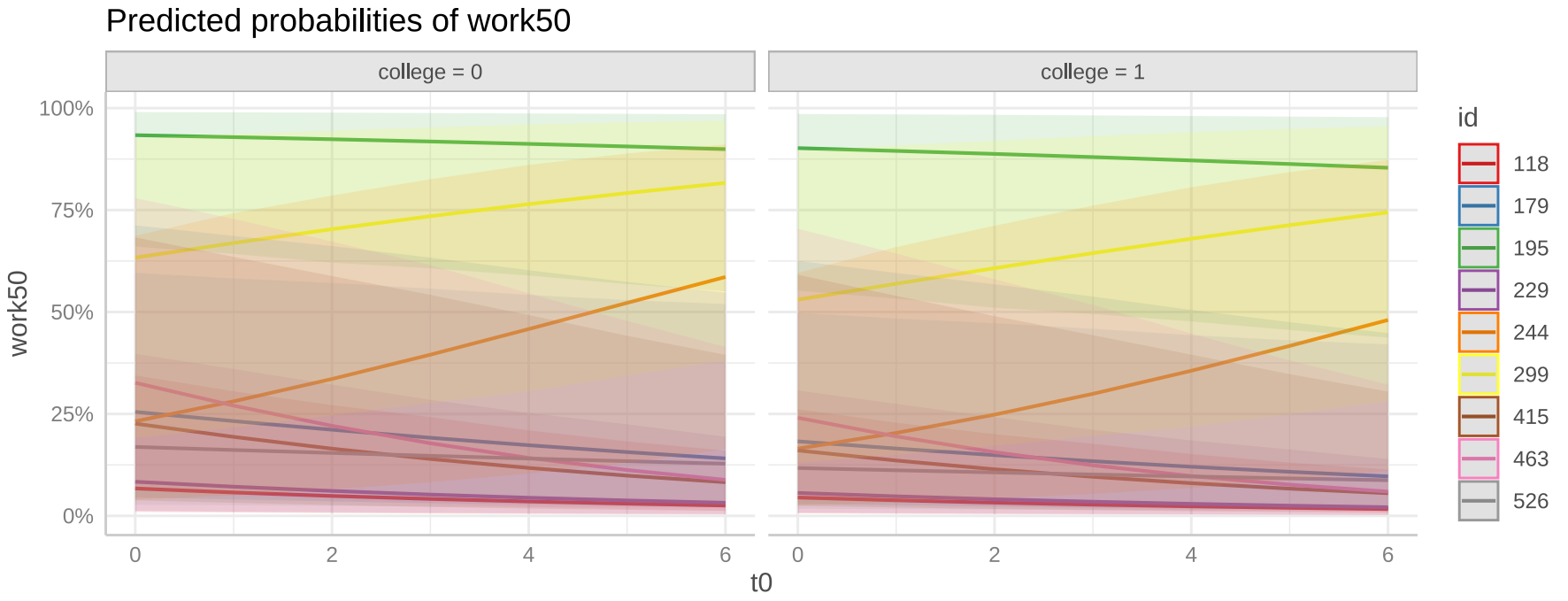
Estimating binary models

Let's say we're trying to predict working 50 or more weeks per year (0/1) as a function of college education and time. We would use the `glmer()` function and specify `family = binomial`.

```
d <- d |> mutate(work50 = if_else(wks >=50, 1L, 0L))      # create new outcome variable
binmod1 <- glmer(work50 ~ college + t0 + (1 + t0 | id),   # formula
                 data = d,                                # data
                 family = binomial)                       # distribution for outcome
```

For latent growth-type models (i.e., models with varying time slopes by individual), `glmer()` uses the Laplace approximation to maximize the likelihood. This usually works fine but is not ideal.

Visualizing model predictions



glmmTMB vs. lme4

```
library(glmmTMB)
bgmod1 <- glmmTMB(work50 ~ college + t0 + (1 + t0 | id), # formula (same as lme4)
                  data = d,                              # data
                  family = binomial)                     # distribution for outcome
```

The glmmTMB package is newer so might play less well with certain things. But it's faster (parallelized) and has more options.

Comparing the fits

	glmer		glmmTMB	
Intercept	-1.191	(0.165)	-1.191	(0.165)
College	-0.425	(0.243)	-0.426	(0.243)
Time	-0.164	(0.039)	-0.164	(0.040)
Log-likelihood	-2014.576		-2014.573	

`glmmTMB()` achieved a slightly better likelihood. The results are very similar but not identical. Both models indicate that college is associated with odds of working 50+ weeks per year that are $e^{-.43} =$ only 65% as great as those without a college degree.

Another example: missing data

```
nida <- haven::read_dta(here("data", "nidalong.dta")) |>  # import Stata data
  select(subjid, treat, opiates, week)                   # keep needed variables
summary(nida)                                            # get descriptive statistics
```

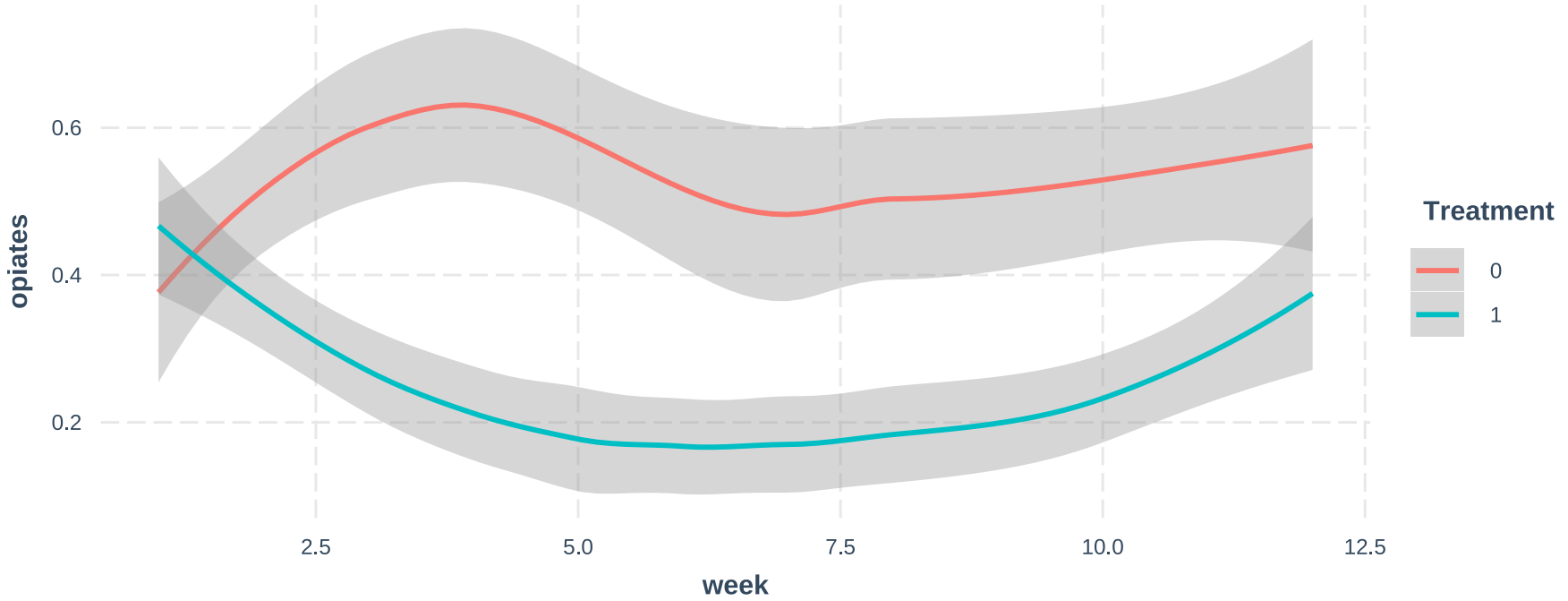
```
##      subjid             treat           opiates            week      
##  Min.   :10000153   Min.   :0.0000   Min.   :0.0000   Min.   : 1.00  
##  1st Qu.:10028995   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 3.75  
##  Median :10051902   Median :0.0000   Median :0.0000   Median : 6.50  
##  Mean   :10050440   Mean   :0.4805   Mean   :0.3528   Mean   : 6.50  
##  3rd Qu.:10072886   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 9.25  
##  Max.   :10099069   Max.   :1.0000   Max.   :1.0000   Max.   :12.00  
##                                      NA's   :924
```

This is data from 134 opioid-addicted youths. Half were randomly assigned to a new drug treatment over a 12-week period and the others got standard detox therapy. Dependent variable = 1 if subject tested positive for opiates in a given week, otherwise 0. Percentage missing data at each week ranged from 10% to 63%.

Estimating a mixed model using maximum likelihood will deal with this kind of missingness under the MAR (missing at random) assumption. That is, missingness can depend on treatment status, or any past or future value of Y, just not on Y_{it} itself.

Descriptive visualization

```
ggplot(nida, aes(y = opiates, x = week, group = factor(treat), color = factor(treat))) +
  geom_smooth(method = "loess") +
  theme_nice() +
  labs(color = "Treatment")
```



Selecting a model

We don't want to overfit our current sample. So we want to select a specification of time that is likely to make good out-of-sample predictions.

df	BIC
4	981.0463
5	986.1380
5	979.9660
7	962.9725

```
# start time at zero
nida$t0 <- nida$week-1

# linear and parallel
omod1 <- glmmTMB(opiates ~ treat + t0 + (1 | subjid),
                 data = nida,
                 family = "binomial")

# linear and not parallel
omod2 <- update(omod1, opiates ~ treat *  t0 + (1 | subjid))

# quadratic and parallel
omod3 <- update(omod1, opiates ~ treat +  t0 + I(t0^2)  + (1 | subjid))

# quadratic and not parallel
omod4 <- update(omod1, opiates ~ treat * (t0 + I(t0^2)) + (1 | subjid))
```

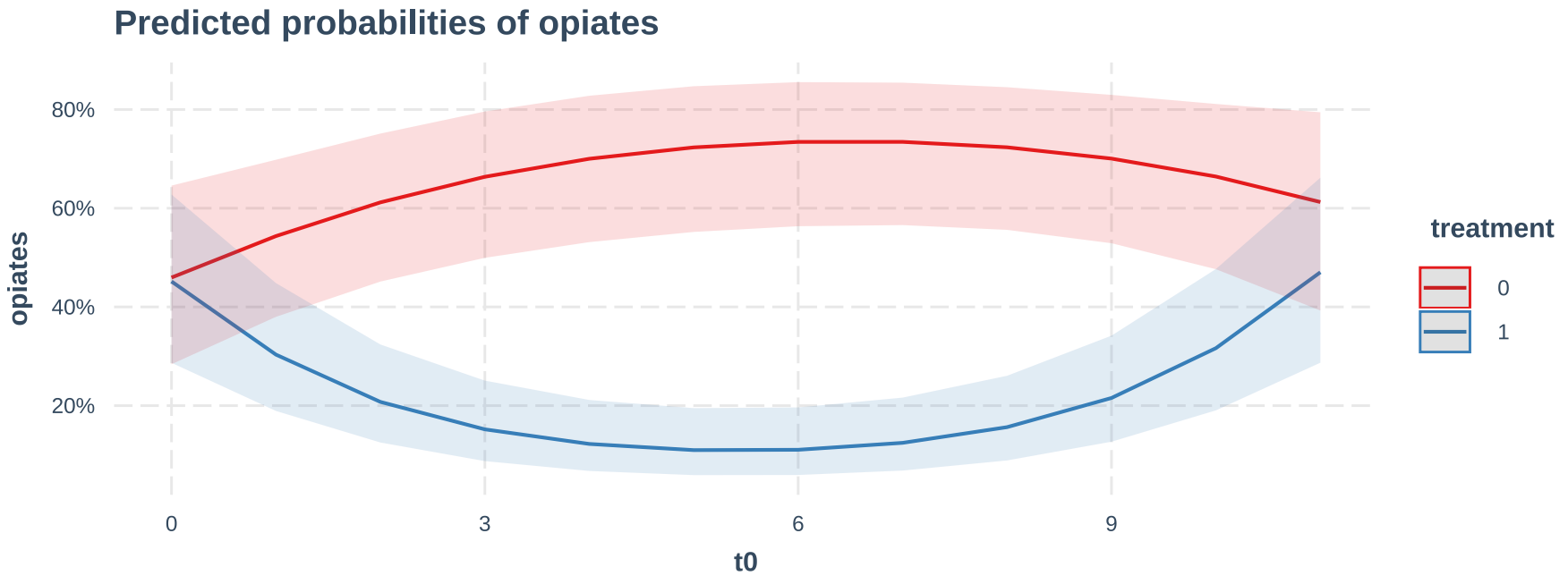
Model results

```
summary(omod4)
```

```
##  Family: binomial  ( logit )
## Formula:
## opiates ~ treat + t0 + I(t0^2) + (1 | subjid) + treat:t0 + treat:I(t0^2)
## Data: nida
##
##      AIC      BIC   logLik deviance df.resid
##    929.2    963.0   -457.6    915.2      917
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  subjid (Intercept) 4.037    2.009
## Number of obs: 924, groups:  subjid, 134
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.16218    0.38932  -0.417   0.6770
## treat         -0.03196    0.53456  -0.060   0.9523
## t0             0.36475    0.14639   2.492   0.0127 *
## I(t0^2)       -0.02803    0.01297  -2.161   0.0307 *
## treat:t0      -1.06639    0.19864  -5.368 7.94e-08 ***
## treat:I(t0^2)  0.09244    0.01771   5.220 1.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Visualizing model predictions

```
ggeffect(omod4,
         terms = c("t0", "treat")) |>
  plot() + labs(color = "treatment") + theme_nice()
```



Summary: MMs for panel data

Facts

Mixed models...

- correct for the dependence of multiple observations per individual
- can deal with missing outcome data under the MAR assumption
- efficiently combine within and between data to obtain optimal group-level estimates

Misconceptions

Mixed models...

- do *not* "control for" unobserved unit-level heterogeneity
- do *not* remove *all* problems related to missing data
- do *not* improve causal identification; MMs only identify a treatment effect if all confounders have been properly modeled

Why MM's don't improve causal inference

- Despite what some people seem to think*, the α_i term in a mixed model is **just another error term**. It doesn't "control for" other factors any more than ϵ_i "controls for" other causes of Y in a linear regression. The model *assumes* that these terms are independent of the predictors.
- The only ways to identify the effect of a time-constant treatment with a mixed model are (1) to have experimental data (like the opiates example) or (2) to adjust for all relevant confounders. This is exactly the same as "regular" regression. The panel structure of the data is doing nothing here to establish causality.
- If you want stronger causal inferences with observational panel data, you must specifically use **within-person variance** or **time-varying treatments**. That is our next topic.

* See Halaby, "Panel Models in Sociological Research" and Allison, *Fixed Effects Regression Models*.

When X changes once

Between, within, and counterfactuals

- Causal inference is hard because we're usually comparing a "treated" person to an "untreated" person. That is, we're using a comparison between two *different* individuals to estimate the effect of some X. But the two people might be different for reasons other than X. This risk of *confounding* is the main limitation in using **between-subject variation** to estimate causal effects.
- When possible, we can make causal inferences more plausible by comparing a "treated" person to *themselves* when they were untreated. That is, a person serves as their own **counterfactual**. Comparisons between different "versions" of the same person are less likely be confounded because many characteristics of persons do not change over time. Thus, **within-subject variation** is more useful for causal inference in observational data.
- We begin with the simplest case: time-varying treatments that change once.

Pre-post

Within variation and "fixed effects"

Using within-person variation to estimate a treatment effect is often called using **fixed effects**.

Let's make this concrete: Let's say I weigh myself in January (time 1) and again in July (time 2). Assume I wasn't dieting before January but I do diet between January and July. In July, I find I weigh 10kg less.

I want to know how much the diet *caused* my weight to change. That is, I'd like to compare my actual weight loss ($\text{weight}^1_{July} - \text{weight}^1_{Jan}$) to a counterfactual weight loss in a world where I didn't diet ($\text{weight}^0_{July} - \text{weight}^0_{Jan}$).

Weight loss example (1)

We can write the following equations to represent my weight at the two time points:

$$\text{weight}_1 = \beta\ \text{diet}_1 + \mu + \epsilon_1$$
$$\text{weight}_2 = \beta\ \text{diet}_2 + \mu + \epsilon_2$$

There are a few non-obvious terms here:

- β is the *treatment effect* of the diet
- μ represents everything about me that *doesn't change* and that affects my weight; this is my own personal **fixed effect**.
- ϵ_1 and ϵ_2 represent other time-varying effects on my weight

Because I wasn't on a diet at time 1, I can rewrite $\beta\ \text{diet}_1$ as 0. Because I was on a diet at time 2, I can rewrite $\beta\ \text{diet}_1$ as β. Then we can subtract the equations from each other:

$$(\text{weight}_2 - \text{weight}_1) = \beta + (\mu - \mu) + (\epsilon_2 - \epsilon_1)$$

Weight loss example (2)

How can we solve for β ? So far we have:

$$(\text{weight}_2 - \text{weight}_1) = \beta + (\mu - \mu) + (\epsilon_2 - \epsilon_1)$$

We can get rid of $(\mu - \mu)$ by simple subtraction. This is the power of **fixed-effects estimation**; by comparing myself to myself (**within-subject** variance), we have gotten rid of *everything* about me that doesn't change, even if I don't know what it is!

If we're willing to assume that the time-varying shocks (ϵ_1 and ϵ_2) don't differ in expectation over time, then we can eliminate the last term as well. (More on this in a minute.)

This means that the best estimate of β is just $(\text{weight}_2 - \text{weight}_1)$, or how much weight I've lost since January. It's that simple!

Strengths and limitations

- Even a simple pre-post comparison like this can be really powerful because it relies on **within-subject** comparisons, or **fixed effects**.
- This matters because we use counterfactual comparisons between a person and themselves to estimate a treatment effect. This is generally superior to comparing two different people who may have unobserved differences between them.
- However, simple pre-post comparisons rely on the assumption that the "world" isn't changing in the background. That is, we're assuming the later shocks (ϵ_2) and the earlier shocks (ϵ_1) are expected to be the same.
- It's easy to imagine counterexamples. For example, maybe I would have lost weight anyway even without dieting because January is during the holidays when it is easier to gain weight (i.e., $\epsilon_1 \gg \epsilon_2$) .
- To deal with this, we might want to compare people who diet between January and July with those who do *not* diet over the same period to capture the effect of the passage of time. This is called **difference-in-differences** estimation.

Difference-in-differences

The dieters lost 10kg, the non-dieters lost 5kg, so the difference-in-differences estimate is: -10 - (-5) = -5kg. This is a better estimate of the weight loss *caused* by the diet because the lower line approximates the *counterfactual* change in weight for the dieters: $E(\text{weight}^0_{July} - \text{weight}^0_{Jan})$.

Example: National Supported Work Demonstration

```
## Rows: 1,444
## Columns: 12
## Groups: id [722]
## $ id        <fct> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10~
## $ t         <dbl> 75, 78, 75, 78, 75, 78, 75, 78, 75, 78, 75, 78, 75, 78, 75, ~
## $ data_id   <chr> "Lalonde Sample", "Lalonde Sample", "Lalonde Sample", "Lalon~
## $ treat     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ age       <dbl> 37, 37, 22, 22, 30, 30, 27, 27, 33, 33, 22, 22, 23, 23, 32, ~
## $ education <dbl> 11, 11, 9, 9, 12, 12, 11, 11, 8, 8, 9, 9, 12, 12, 11, 11, 16~
## $ black     <dbl> 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, ~
## $ hispanic  <dbl> 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ married   <dbl> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, ~
## $ nodegree  <dbl> 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, ~
## $ re        <dbl> 0.0000, 9930.0459, 0.0000, 3595.8940, 0.0000, 24909.4492, 0.~
## $ post      <int> 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, ~
```

The treatment here (`treat`) is participation in a job training program. The outcome of interest is real earnings (`re`). We want to estimate the average treatment effect for those who participated in the program.

Graphical representation

DiD as a regression model

As a regression model, two-period DiD looks like this:

$$\mathbb{E}(y_{it}) = \beta_0 + \beta_1 \text{treat} + \beta_2 \text{post} + \beta_3 \text{treat} \times \text{post}$$

Where `treat` is a 0/1 variable indicating whether the case will ever be treated and `post` is a 0/1 variable indicating whether the observation is in the second period of observation (here, 1978).

β_3 is our estimate of the **average treatment effect on the treated**. This is because we are using the untreated people to estimate what the treated people *would have looked like* in the absence of the training program. That is, they help us estimate the *counterfactual*.

Implementation via `lmer()`

The two-period difference in differences is easy to implement.

```
didreg <- lmer(re ~ treat * post + (1 | id), # formula including alpha_i for repeat measures
               data = nsw_long,              # data
               REML = FALSE)                 # maximum likelihood
```

term	estimate	std.error
(Intercept)	3027	275
treat	39	429
post	2063	359
treat:post	847	559

Our best estimate of the effect of the program is to increase earnings by 847 USD, although the relatively large standard error (559 USD) makes that pretty uncertain.

Adding adjustment variables to DiD

term	estimate	std.error
(Intercept)	2243	1782
treat	-64	424
post	2063	359
age	18	25
education	193	122
black	-1822	543
hispanic	-749	712
married	1646	436
nodegree	-413	506
treat:post	847	559

Adding covariates here doesn't change the `treat:post` coefficient or its standard error. This is because none of these covariates change over time. And using within variation automatically eliminates any *time-constant* confounding between the treatment and the outcome.

But it may be interesting for other reasons (or appease reviewers!).

Assumptions and caveats

DiD (like all approaches) makes assumptions. The most important are:

1. Parallel trends: in the absence of the treatment, the two groups' trajectories would have changed the same amount.
2. No anticipation: the treated group doesn't change their behavior in *anticipation* of the treatment.

These assumptions are impossible to test in the two-period case. With two pre-treatment periods, they can be evaluated.

This is from a different version (Dehejia-Wahba sample) of the NSW with two pre-treatment periods. It doesn't look *entirely* parallel but the difference is small.

Testing the assumption

```
pt_data <- filter(nsw_dw_long, post==0)  # keep only pretreatment
pt_test <- lmer(re ~ treat*factor(t) + (1|id), data = pt_data, REML = FALSE)
```

term	estimate	std.error
(Intercept)	2107	272
treat	-11	423
factor(t)75	-840	252
treat:factor(t)75	277	391

The interaction term here is the test of the pre-treatment parallel trends assumption. We could not reject the null hypothesis here that the two trends are parallel.

Parallel trends and measurement

If pre-treatment trends are parallel in *dollars*, then they cannot (strictly) be parallel in *log dollars*. Of course, the test might be "non-significant" either way but you want to think about this.

DiD augmentations

Even the two-period DiD can be enhanced. One strategy is to use a matching or weighting algorithm to make the control cases look as similar as possible to the treated cases on all *pre-treatment* covariates. Tackling this would introduce way too many new concepts for this short course!

For more on this, see Sant'Anna and Zhao (2020), "Doubly Robust Difference-in-Differences Estimators." *Journal of Econometrics*.

When X changes at different times, in the same direction

A fork in the road

When treatment timing varies (i.e., doesn't happen at the same time) and is not subsequently reversed this is usually called differential timing or staggered treatments. We will always have three or more waves in this situation, by definition. We have two major options here:

1. Use two-way fixed effects regression (TWFE), which includes fixed effects for *unit* and *time period*.
2. Use a technique built for differential timing DiD, such as the one outlined in Callaway and Sant'Anna (2020)

Prior to 2019, almost everyone would have used (1). But *many* recent papers have shown that TWFE only identifies the average treatment effect on the treated (ATT) when it is the *same regardless of timing.* The techniques in (2) allow for different effects for differently timed groups. We do not have the time in this course to cover this fast-evolving literature. But we do offer a full course on DiD.

See Borusyak and Jaravel (2018), de Chaisemartin and D'Haultfoeuille (2020), Goodman-Bacon (2020), and Sun and Abraham (2020) for more on this.

Two-way fixed effects

We will focus here on two-way fixed effects. This estimation strategy attempts to remove *two* major sources of confounding to identify the treatment effect:

1. The individual unit fixed effect
2. The effect of being observed at a particular time

This leaves the presence or absence of the treatment as varying within persons. For those who are sometimes treated and sometimes not, we can compare their treated and untreated times to estimate the effect.

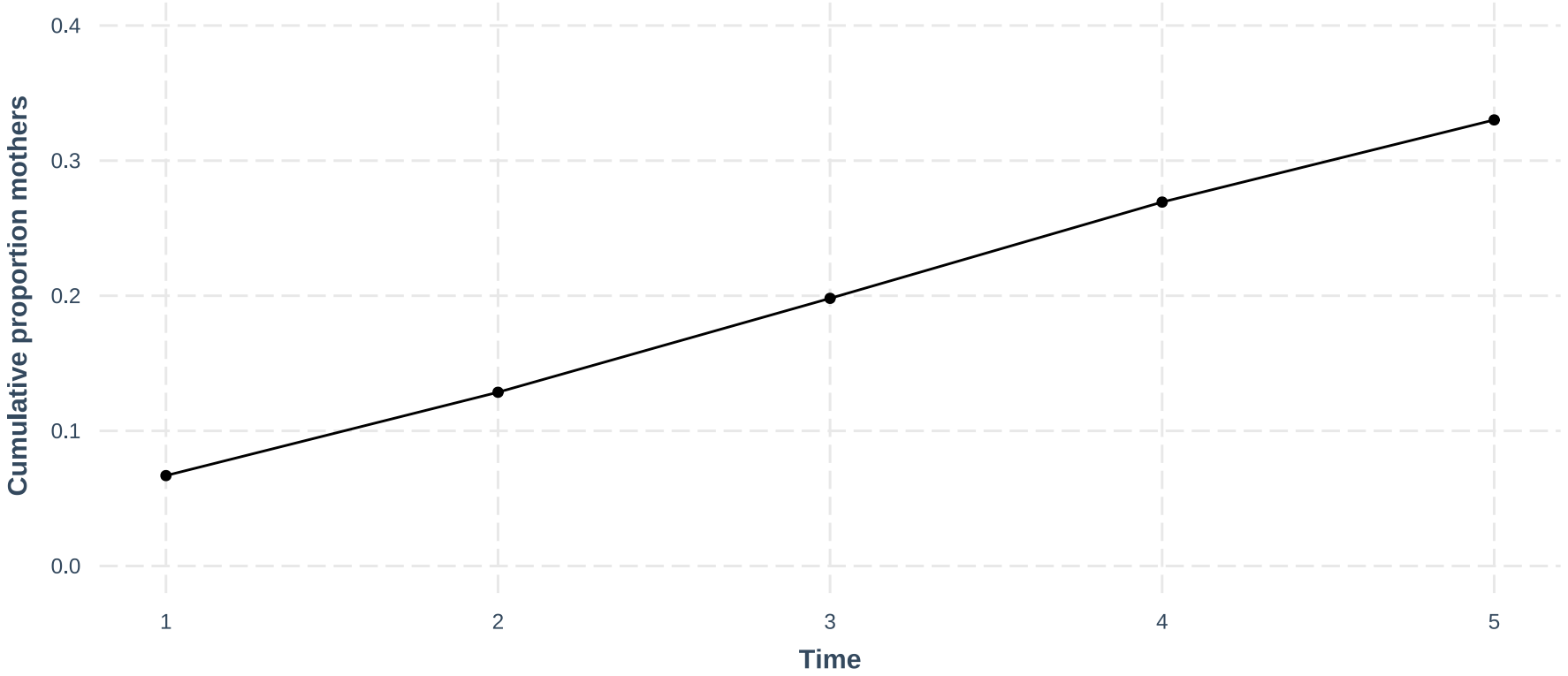
Those who are *always* or *never* treated do not contribute any information to fixed effects estimation. We don't typically have to drop them from the dataset; their information is simply not used.

Example: poverty and motherhood

We'll return to the teen poverty data. Our question is whether having a baby increases poverty risk above and beyond one's *fixed baseline tendency* to be in poverty and the risks associated with a *particular time period*. And, of course, we should remember that the fact that teen motherhood is a risk for poverty in the US is a policy choice!*

*See Brady et al. 2017. "Rethinking the risks of poverty: A framework for analyzing prevalences and penalties." *American Journal of Sociology*.

Timing of motherhood "treatment"



This graph makes it clear that about 7% of the respondents are always mothers and about 66% are never mothers during the study. So our estimates of the motherhood effect will be based on the remaining 27% of respondents.

The TWFE model

Even though poverty is a binary outcome, we will first use a linear model (the linear probability model or LPM) to estimate this effect.

$$\text{poor}_{it} = \beta\, \text{mother}_{it} + \mu_i + \theta_t + \epsilon_{it}$$

β is the estimated treatment effect of motherhood on poverty (i.e., what we want to know);

μ_i represents everything fixed about person i that affects her probability of being poor. It is equivalent to having 1,151 dummy variables (one for each girl) in the model.

θ_t represents everything about each of the 5 time periods that makes poverty more or less likely during that period. This is equivalent to having 4 time dummy variables in the model (the first time period is the reference category).

Estimating the TWFE model

For general TWFE, the `plm::plm()` function is useful. If we want SEs clustered by unit *and* time,* we can use `lmtest::coeftest()` on the model object.

```
twfe <- plm(pov ~ mother,                                  # formula
            data = teen_pov,                               # data
            index = c("id", "t"),                          # variables identifying unit and time
            model = "within",                              # 'within' variance only (fixed effects)
            effect = "twoways")                            # fix.eff. for unit AND time
twfe_robust <- lmtest::coeftest(twfe, vcov = vcovDC)       # double clustered SEs
tidy(twfe_robust)
```

term	estimate	std.error	statistic	p.value
mother	0.0849	0.0267	3.18	0.00146

We estimate that motherhood increases the probability of being poor by 8.5 points.

*See Cameron et al. (2011). "Robust inference with multiway clustering." *Journal of Business & Economic Statistics* and Thompson (2011). "Simple formulas for standard errors that cluster by both firm and time." *Journal of Financial Economics*.

Conditional logit

The fixed effects logistic version of this model is very similar:

$$\log\left(\frac{P(\text{poor}_{it}=1)}{1-P(\text{poor}_{it}=1)}\right) = \beta\,\text{mother}_{it} + \mu_i + \theta_t$$

With μ_i and θ_t defined as on the previous slide.

This is *conditional* logit because the μ_i term allows us to condition out the *number* of times a respondent is in poverty and to use the time-varying variables (mother and time) to predict *which* time periods those poverty spells occur.

Because of this, anyone who is in either always or never in poverty cannot contribute to the estimation.

Estimating the model

In the logistic case, we need to *condition out* the μ_i terms during the estimation. We will use `survival::clogit()` which, unlike `plm()`, requires us to specify the time dummies manually.

```
library(survival)
cond_logit <- clogit(pov ~ mother + strata(id) + factor(t),  # strata is unit id variable
                     data = teen_pov,                       # data
                     method = "exact")                      # to get conditional logit
tidy(cond_logit) |> filter(term == "mother")
```

term	estimate	std.error	statistic	p.value
mother	0.464	0.147	3.14	0.00166

Net of fixed effects, we estimate that motherhood increases the odds of being poor by $e^{.464} = 1.59$ times.

What TWFE can and can't do

TWFE can...

- adjust for *all* time-constant confounders
- adjust for the effects of specific time periods
- identify homogeneous treatment effects if assumptions are met

TWFE can't...

- rule out time-varying confounders (without modeling them)
- rule out reverse causality
- deal with heterogeneous treatment effects

Adding adjustment variables to TWFE

Because TWFE alone cannot rule out *time-varying* confounding, we can adjust the estimates for other time-varying predictors. Here's an example for the linear FE model.

```
twfe2 <- update(twfe, pov ~ mother + spouse + inschool + hours)
twfe2_robust <- lmtest::coeftest(twfe2, vcov = vcovDC)
tidy(twfe2_robust)
```

term	estimate	std.error	statistic	p.value
mother	0.103	0.0296	3.48	0.000501
spouse	-0.132	0.0333	-3.97	7.41e-05
inschool	0.0462	0.0289	1.6	0.11
hours	-0.00343	0.000743	-4.61	4.11e-06

Net of unit and time fixed effects *and* holding these *time-varying* covariates constant, we estimate that motherhood increases poverty risk by 10.3 percentage points. But if, say, work hours is post-treatment (affected by motherhood), then this estimate will be biased.

TWFE conclusion

Two-way fixed effects models can be a powerful tool for estimating treatment effects when their assumptions are met. If you have few waves of data spaced relatively close together (e.g., 3 or 4 annual waves) the assumption that the treatment effect does not vary by time-of-adoption may be pretty realistic.

A very, very brief note on staggered DiD

You can estimate staggered DiD without assuming homogeneity using the `did` package.* I put the code in the slides .Rmd.

For the effect of motherhood on poverty risk, you get an ATT estimate of 0.085 with a SE of 0.032. This is very similar to the original TWFE estimate in this case. That will not always be true, however, especially as the the post-treatment period gets longer for some units.

*Callaway and Sant'Anna (2020). https://bcallaway11.github.io/did/

When X changes at different times, in any direction

Overview

So far, all the treatments we have seen "turn on" at some point but then never "turn off" again. But many real-life treatments are not that way. Consider dieting, union membership, watching cable news, attending church, or adhering to a medication regime. These are all examples of **reversible treatments**.

In more general terms, this is a situation where both Y *and* X might vary from time period to time period. This extra variation introduces a few new possibilities.

Example

Let's return to the wage data and consider `union` as a "treatment." We want to know how much (if at all) union membership *causally affects* wages in this population.

First let's get a sense of the variation in this treatment variable.

```
d <- d |> # already grouped
  mutate(everunion = max(union),
         alwaysunion = min(union))
mean(d$everunion)*595         # number of Rs ever in a union (595 is the overall N)
```

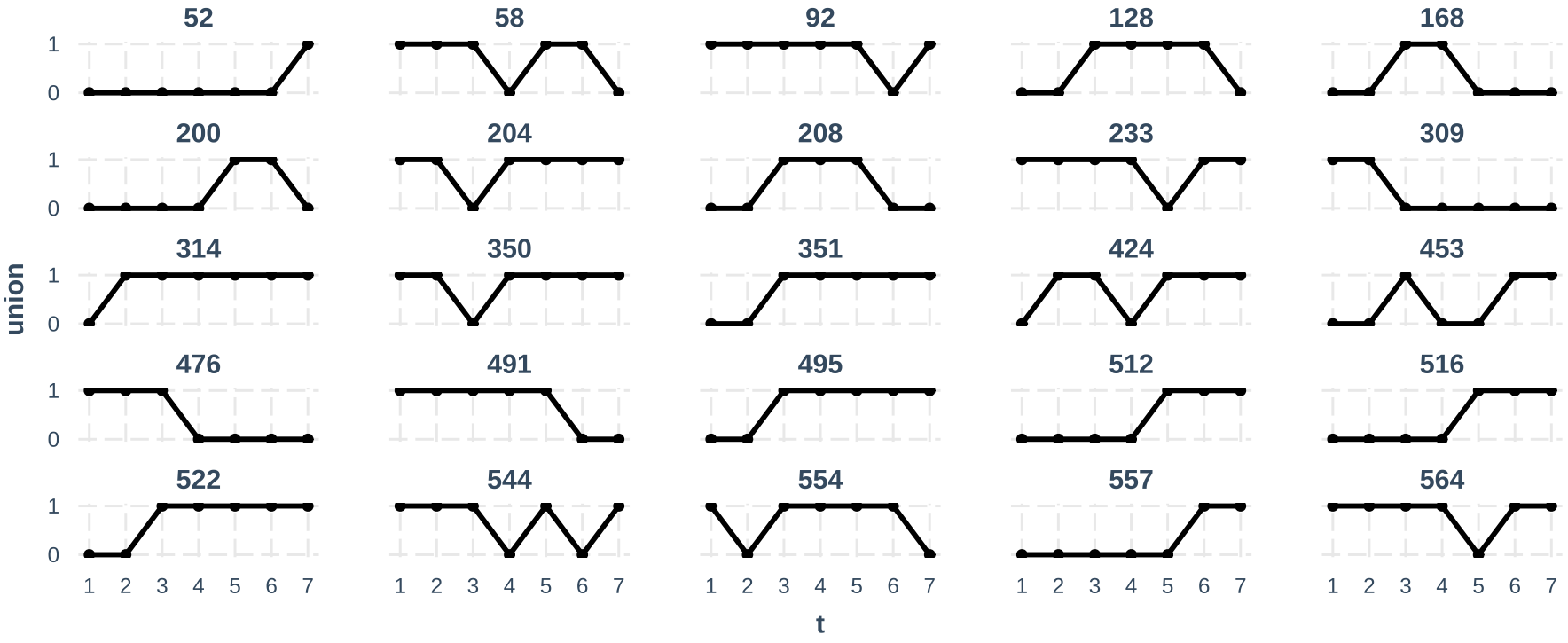
```
## [1] 258
```

```
mean(d$alwaysunion)*595       # number of Rs always in a union
```

```
## [1] 172
```

This means we have 86 respondents who are sometimes in a union and sometimes not. We can use their variation in `union` and `lwage` to estimate a treatment effect for union membership. If we believe the expected effect of union membership is the same for everyone on average, we can use changing union status to better estimate counterfactual wages within persons.

Visualizing union



Sample of union-status changers

Two-way fixed effects (again)

TWFE is still useful

We can still use two-way fixed effects to estimate the effect of union membership. We will include all time-varying covariates in the model as well.

```
union_twfe <- plm(lwage ~ union + wks + occ + ind + south + smsa + ms,
                  index = c("id","t"),
                  model = "within",
                  effect = "twoways",
                  data = d)
union_twfe_robust <- lmtest::coeftest(union_twfe, vcov = vcovDC)
tidy(union_twfe_robust) |> filter(term == "union") |> select(term:std.error)
```

term	estimate	std.error
union	0.0307	0.0258

There may be a small positive effect of union membership on wages (about +3%), but the standard error is quite large compared to the estimate. By conventional standards, we would not want to conclude that union membership had a causal effect on wages for this population.

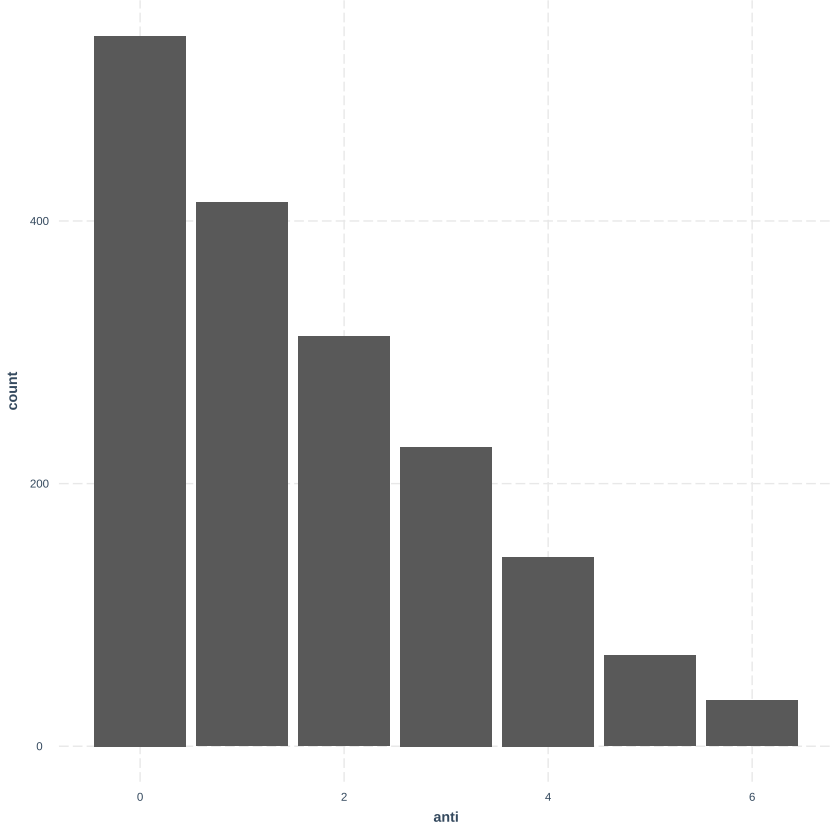
TWFE for continuous treatments

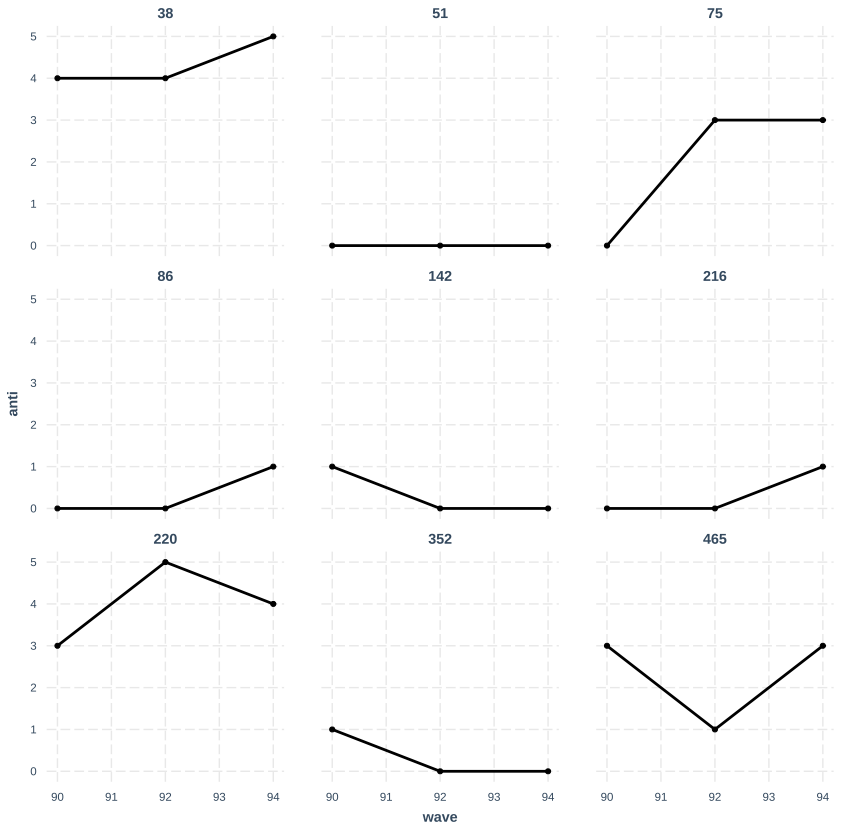
We can use TWFE with continuous treatments. Here we're looking for the average treatment effect of increasing the "dose" of X by one unit. Let's consider an example. We'll use data from the 1990-1994 National Longitudinal Study of Youth. Our main outcome will be antisocial behavior (`anti`; measured from 0-6) and our "treatment" of interest will be self-esteem (`self`; ranging from 6 to 24).

```
data(nlsy, package = "panelr")                      # get the wide data
nlong <- nlsy |>                                    # create new "long" object
  long_panel(periods = c(90, 92, 94), id = "id")    # year suffixes and ID
glimpse(nlong)
```

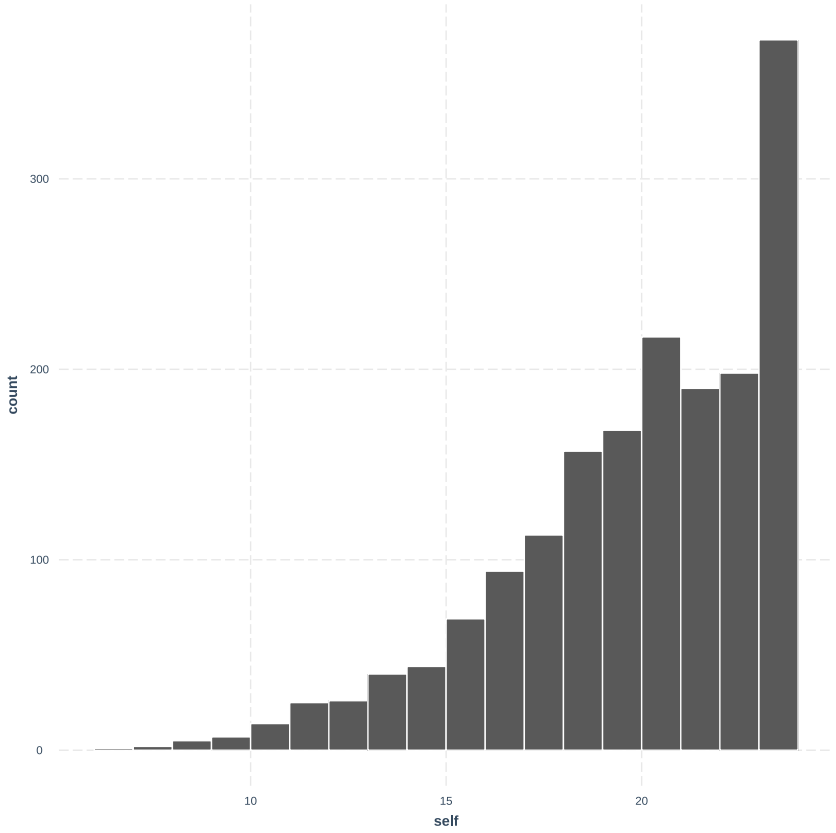
```
## Rows: 1,743
## Columns: 12
## Groups: id [581]
## $ id       <fct> 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7~
## $ wave     <dbl> 90, 92, 94, 90, 92, 94, 90, 92, 94, 90, 92, 94, 90, 92, 94, 9~
## $ momage   <dbl> 21, 21, 21, 22, 22, 22, 18, 18, 18, 24, 24, 24, 22, 22, 22, 2~
## $ gender   <dbl> 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1~
## $ childage <dbl> 8.000000, 8.000000, 8.000000, 8.416667, 8.416667, 8.416667, 8~
## $ hispanic <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ black    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ momwork  <dbl> 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1~
## $ married  <dbl> 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1~
## $ anti     <dbl> 1, 1, 1, 0, 0, 0, 5, 5, 5, 2, 3, 1, 1, 0, 0, 1, 1, 1, 3, 3, 4~
## $ self     <dbl> 21, 24, 23, 20, 24, 24, 21, 24, 24, 23, 21, 21, 22, 23, 24, 1~
## $ pov      <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

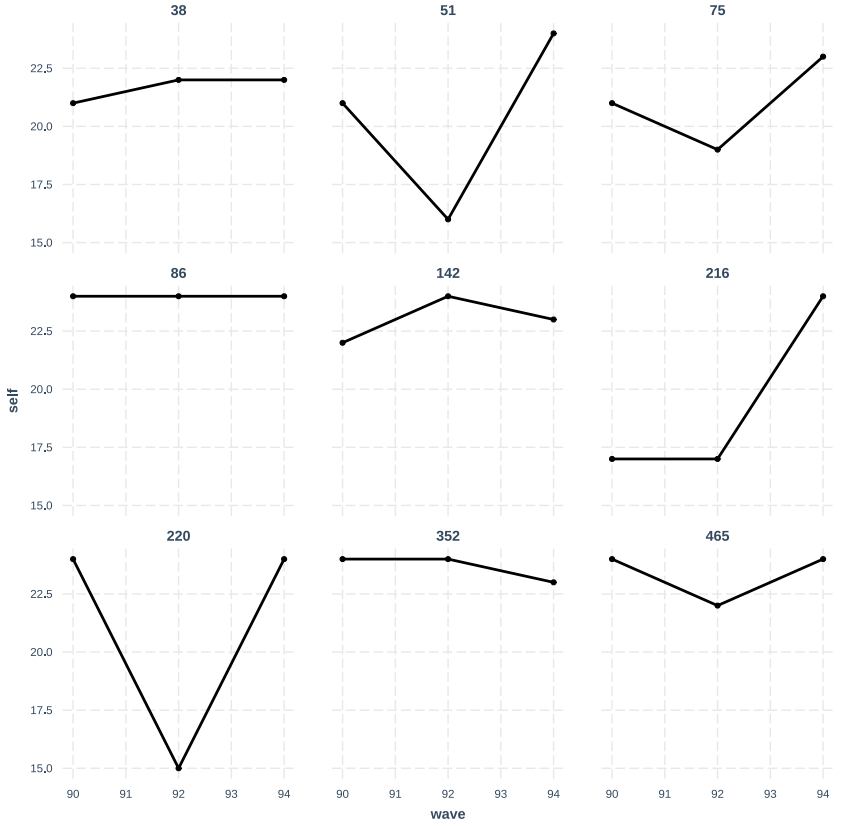
Visualizing the outcome: anti





Visualizing the treatment: `self`





The continuous TWFE model

The model is the same as what we've seen already. We are interested in β_1 and we probably want to control for the only other time-varying covariate, poverty status (`pov`, coded 0/1).

$$\text{anti}_{it} = \beta_1 \text{self}_{it} + \beta_2 \text{pov}_{it} + \mu_i + \theta_t + \epsilon_{it}$$

We can use `plm()` to estimate the TWFE exactly as before, which yields the following result:

term	estimate	std.error
self	-0.0552	0.0102
pov	0.112	0.093

Thus, our best estimate is that each point of self-esteem reduces anti-social behavior by .055 points on average. This is net of all *time-constant* characteristics and poverty status. Of course, other time-varying confounding is possible.

Summary: FE for reversible treatments

- FE models are simple and effective at ruling out time-constant confounding
- We can estimate these models linearly (e.g., with `plm()`) or using a logistic model for binary outcomes (e.g., using `survival::clogit()`)
- FE models can be augmented by time-varying adjustment variables to attempt to rule out time-varying confounding as well
- **However**, FE models only use data from units that spend at least some time in and out of the treatment condition
- If only a few units switch treatment statuses, your analyses may not have enough power to estimate an effect accurately

Mixed models for time-varying treatments

Another look at MMs

- At the beginning of the course, we focused on mixed models as an approach for modeling time-constant treatments (like the opiate treatment example)
- We can use MMs to model time-varying treatments as well, but we have to be careful about our model specfications and model assumptions.
- MMs are not as simple as fixed-effects models for causal inference. But used appropriately, they are a powerful tool for modeling.

The two datasets

Every panel dataset really contains two datasets: the **between** data and the **within** data. We can separate them out to demonstrate this.

Let's start with the between data.

```
# between data
n_between <- nlong |> # already grouped
  summarize(across(.cols = everything(),
                   .fns = ~ mean(.x))) |>
  unpanel() |>
  select(-wave)
datasummary_skim(n_between)
```

The between data

	Unique (#)	Missing (%)	Mean	SD	Min	Median	Max	
momage	10	0	20.7	2.2	16.0	21.0	25.0	
gender	2	0	0.5	0.5	0.0	1.0	1.0	
childage	25	0	8.9	0.6	8.0	8.9	10.0	
hispanic	2	0	0.2	0.4	0.0	0.0	1.0	
black	2	0	0.4	0.5	0.0	0.0	1.0	
momwork	2	0	0.3	0.5	0.0	0.0	1.0	
married	2	0	0.2	0.4	0.0	0.0	1.0	
anti	18	0	1.6	1.3	0.0	1.3	5.7	
self	37	0	20.4	2.4	12.3	21.0	24.0	
pov	4	0	0.3	0.4	0.0	0.0	1.0	

Making the within data

```
# create demeaning function
demean <- function(x) x - mean(x)

# unpanel (for max control) and make wave a factor
nlong2 <- nlong |>
  unpanel() |>
  mutate(wave = factor(wave))

# create within data
n_within <- nlong2 |>
  group_by(id) |>
  mutate(across(.cols = where(is.numeric),
                .fns = demean))

# summary statistics from modelsummary package
datasummary_skim(n_within)
```

Within data (1)

	Unique (#)	Missing (%)	Mean	SD	Min	Median	Max	
momage	1	0	0.0	0.0	0.0	0.0	0.0	
gender	1	0	0.0	0.0	0.0	0.0	0.0	
childage	1	0	0.0	0.0	0.0	0.0	0.0	
hispanic	1	0	0.0	0.0	0.0	0.0	0.0	
black	1	0	0.0	0.0	0.0	0.0	0.0	
momwork	1	0	0.0	0.0	0.0	0.0	0.0	
married	1	0	0.0	0.0	0.0	0.0	0.0	
anti	42	0	0.0	0.8	-3.7	0.0	3.3	
self	75	0	-0.0	2.3	-8.7	0.0	9.3	
pov	5	0	0.0	0.3	-0.7	0.0	0.7	

Within data (2)

	Unique (#)	Missing (%)	Mean	SD	Min	Median	Max	
anti	42	0	0.0	0.8	-3.7	0.0	3.3	
self	75	0	-0.0	2.3	-8.7	0.0	9.3	
pov	5	0	0.0	0.3	-0.7	0.0	0.7	

After dropping all the within variables that have no variation.

The two regressions

```
between_reg <- lm(anti ~ momage + gender + childage + hispanic + black +
                    momwork + married + self + pov,
                  data = n_between)
```

```
within_reg <- lm(anti ~ self + pov + wave,
                 data = n_within)
```

	Between	Within
momage	-0.011	
gender	-0.508	
childage	0.086	
hispanic	-0.280	
black	0.111	
momwork	0.164	
married	-0.128	
self	-0.090	-0.055
pov	0.616	0.112
wave92		0.044
wave94		0.211
N	581	1743

What do these coefficients mean?

The between coefficients are how *averages* of X predict *averages* of Y.

The within coefficients are how *deviations* from the average X predict *deviations* from the average Y.

Note that the `self` and `pov` within betas are the same as what we got using TWFE! This is because both methods only use within-unit variance.

Return of the mixed model

```
mixed_reg <-
  lmer(anti ~ momage + gender + childage + hispanic + black +
         momwork + married + self + pov + wave + (1 | id),
       REML = FALSE,
       data = nlong2)
```

	Between	Within	Mixed
momage	-0.011		-0.022
gender	-0.508		-0.483
childage	0.086		0.088
hispanic	-0.280		-0.218
black	0.111		0.227
momwork	0.164		0.261
married	-0.128		-0.050
self	-0.090	-0.055	-0.062
pov	0.616	0.112	0.247
wave92		0.044	0.047
wave94		0.211	0.216
N	581	1743	1743

You can see that the variables that have *both* within and between variance (here, just `self` and `pov`) are assigned coefficients that are somewhere *between* the pure between and pure within estimates.

But why? What's going on here?

What is the model doing?

$$\overline{\text{anti}_{i.}} = \gamma_0 + \gamma_1 \text{momage}_i + \gamma_2 \text{gender}_i + \gamma_3 \text{childage}_i + \gamma_4 \text{hispanic} + \gamma_5 \text{black}_i + \gamma_6 \text{momwork}_i + \gamma_7 \text{married}_i + \beta_1 \overline{\text{self}_{i.}} + \beta_2 \overline{\text{pov}_{i.}} + \alpha_i$$

$$(\text{anti}_{it} - \overline{\text{anti}_{i.}}) = \beta_1 (\text{self}_{it} - \overline{\text{self}_{i.}}) + \beta_2 (\text{pov}_{it} - \overline{\text{pov}_{i.}}) + \theta_t + \epsilon_{it}$$

The first equation is the **between equation** and the second equation is the **within equation**.

In the mixed model, it's as if the two equations are being estimated simultaneously. The constraint is that β_1 and β_2 have to have the *same values across both equations*.

The mixed model explained

If we just work with one time-constant variable (Z) and one time-varying variable (X), this is a little easier to see.

$$\begin{aligned}
\bar{y}_{i.} &= \gamma_0 + \gamma_1 z_i + \beta \bar{x}_{i.} + \alpha_i \\
(y_{it} - \bar{y}_{i.}) &= \beta(x_{it} - \bar{x}_{i.}) + \theta_t + \epsilon_{it} \\
\alpha_i &\sim N(0, \tau^2) \\
\epsilon_{it} &\sim N(0, \sigma^2) \\
\alpha_i &\perp \epsilon_{it}
\end{aligned}$$

I added the other assumptions just for completeness.

β appears in both equations and must take on the same value. This means that the jointly estimated β will lead to somewhat worse predictions (i.e, biased estimates) than when the equations are estimated separately. This is because β is trying to do two things at once, serve two masters, etc.

The conventional mixed model (1)

We can write the model combined like this:

$$y_{it} = \gamma_0 + \gamma_1 z_i + \beta x_{it} + \theta_t + \alpha_i + \epsilon_{it}$$

It means *exactly* the same thing as on the last slide but it's a little harder to see the assumptions.

The conventional mixed model (2)

In some fields, it's conventional to write it using two (different) equations:

$$y_{it} = \beta_{0i} + \beta_1 x_{it} + \theta_t + \epsilon_{it}$$
$$\beta_{0i} = \gamma_0 + \gamma_1 z_i + \alpha_i$$

Here we're making the *intercept* of the first equation something that varies across respondents and a function of the time-constant parts of the model.

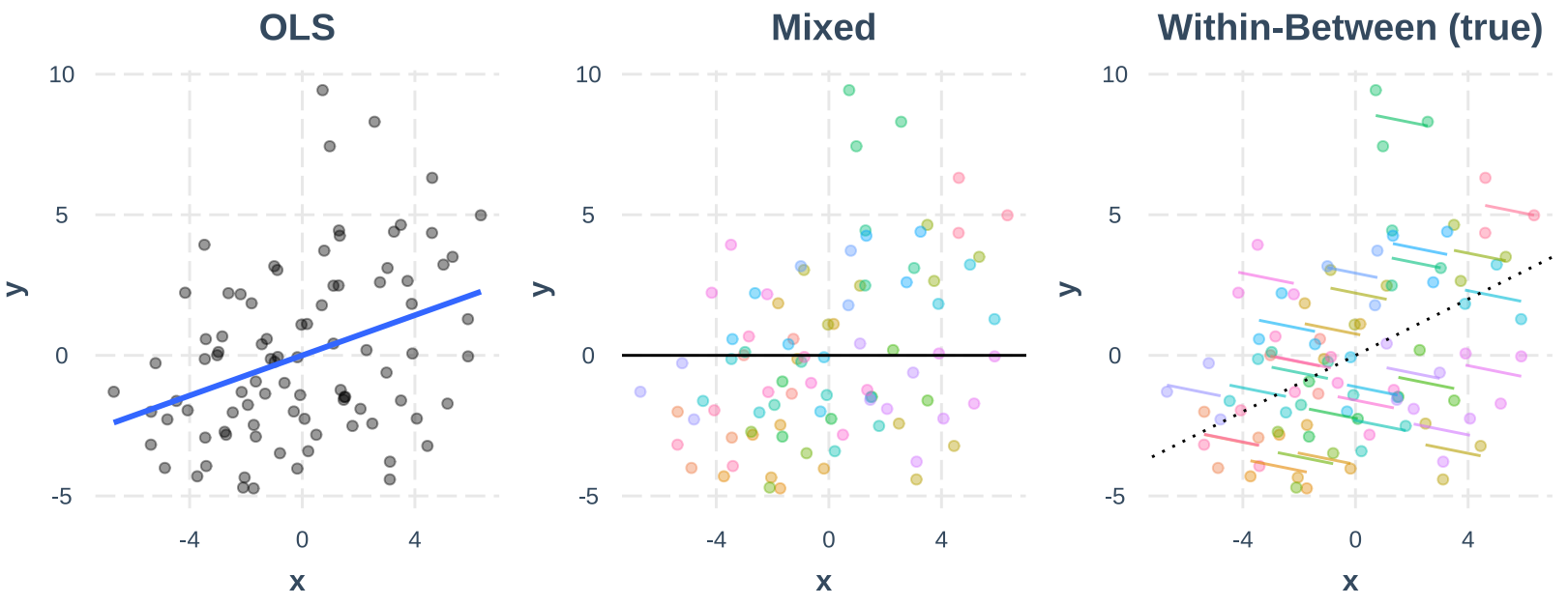
Again, this means *exactly* the same thing but it obscures the fact that the first equation is estimated using both between and within variance on X.

	Between	Within	Mixed
momage	-0.011		-0.022
gender	-0.508		-0.483
childage	0.086		0.088
hispanic	-0.280		-0.218
black	0.111		0.227
momwork	0.164		0.261
married	-0.128		-0.050
self	-0.090	-0.055	-0.062
pov	0.616	0.112	0.247
wave92		0.044	0.047
wave94		0.211	0.216
N	581	1743	1743

Looking at these coefficients again, we can understand why the mixed model betas for the time-varying variables (`self` and `pov`) are between the other two versions of those betas. They represent a compromise.

And because the `self` and `pov` coefficients are different, the other *mixed* coefficients have to be different from those in the *between* model to compensate.

An implication: for time-constant variables (like `gender` or `married`), the difference between the *between* and *mixed* versions of their betas is a function of how associated they are with the means of `self` and `pov`.



This is the result of a simulated process where the between effects and the within effects differ so you can get a sense of how this works.

	OLS	Mixed	W-B
x	0.357	-0.000	
d_x			-0.200
m_x			0.400
BIC	457.961	371.266	368.086

Why mixed models?

If mixed models are less accurate than separate within and between models, why do we use them?

- They allow us to estimate the effect (β) of a time-varying X using *both* within and between variance
- This effectively gives us more data!

Consider poverty (pov). There are 581 respondents in this dataset. 303 are *always* in poverty and 107 are *never* in poverty. So 71% of the respondents have no within-unit variance on this variable. That means that the within coefficients are estimated using data from only 171 respondents.

The information we're getting from those 171 is *high quality*, however, because using within-subject variance allows us to *treat each person as their own counterfactual*, thus controlling for *all* unchanging characteristics.

The between-subject variance cannot rely on this counterfactual approach and thus depends on **explicitly modeling confounders** to identify the treatment effect. This means we're much more at risk of omitted variable bias. This is the danger of mixed models.

Within-between and correlated random effects models

Extending the mixed model

Once we understand how the mixed model works, we can customize it to be more flexible. Specifically, we can estimate models that allow us to decouple the within and between components in the same equation. There are two basic ways of doing this:

Within-between ("hybrid") model

$$y_{it} = \gamma_0 + \gamma_1 z_i + \beta_W (x_{it} - \bar{x}_{i.}) + \beta_B \bar{x}_{i.} + \theta_t + \alpha_i + \epsilon_{it}$$

Correlated random effects ("contextual") model

$$y_{it} = \gamma_0 + \gamma_1 z_i + \beta_W x_{it} + \beta_C \bar{x}_{i.} + \theta_t + \alpha_i + \epsilon_{it}$$

Both fit the data equally well, they just have different interpretations since $\beta_C = \beta_B - \beta_W$. That is, β_C estimates the *difference* between the within and between effects.*

* See Schunck and Perales, "Within- and between-cluster effects in generalized linear mixed models".

Estimation with `panelr::wbm()`

Within-between

```
wb_mod <- wbm(anti ~ self + pov | momage + gender + childage +
                hispanic + black + momwork + married,
              model = "w-b",                                    # type of model
              REML = FALSE,                                     # ML
              use.wave = TRUE,                                  # include time
              wave.factor = TRUE,                               # include time as dummies
              data = nlong)                                     # using a panel_data object
```

Contextual

```
cr_mod <- update(wb_mod, model = "contextual")
```

In the formula syntax, we put the time-varying variables we want to "split" before the vertical bar and the time-constant variables (or time-varying variables we don't want to split) after the bar.

	Within-Between		Contextual	
self	-0.055	(0.011)	-0.055	(0.011)
pov	0.112	(0.093)	0.112	(0.093)
(Intercept)	2.993	(1.150)	2.993	(1.150)
imean(self)	-0.090	(0.022)	-0.035	(0.024)
imean(pov)	0.616	(0.155)	0.504	(0.181)
momage	-0.011	(0.025)	-0.011	(0.025)
gender	-0.508	(0.106)	-0.508	(0.106)
childage	0.086	(0.090)	0.086	(0.090)
hispanic	-0.280	(0.138)	-0.280	(0.138)
black	0.111	(0.131)	0.111	(0.131)
momwork	0.164	(0.118)	0.164	(0.118)
married	-0.128	(0.127)	-0.128	(0.127)
wave.L	0.149	(0.042)	0.149	(0.042)
wave.Q	0.050	(0.041)	0.050	(0.041)
BIC	5963.856		5963.856	

The `imean(self)` and `imean(pov)` coefficients are the only one that differ between the two models.

In the WB model they are β_B-type coefficients; that is, the estimated effects of each subject's average value of that variable.

In the contextual (CRE) model, they are β_C-type coefficients; that is, estimates of the *difference* of the β_B coefficient from the β_W coefficient.

Classical ("Hausman") test

The traditional way to determine whether one is "allowed" to use a mixed model instead of fixed effects is to test whether the within and between effects are the same. This is often called a Hausman test. In the correlated random effects (contextual) model, we can test this using a joint test of the difference of both `imean()` coefficients from zero.

The null hypothesis is that the between and within effects are *equal* conditional on the model (i.e., that $\beta_{\mathbf{C}} = 0$). Thus, rejecting the null hypothesis at a given level of "significance" means that we should use within estimates only rather than a mixed model due to bias concerns.

```
car::linearHypothesis(cr_mod, c("`imean(self)` = 0",
                                "`imean(pov)` = 0"))
```

Df	Chisq	Pr(>Chisq)
2	10	0.00671

In this case, for example, assuming an α level of .01, we would reject the null hypothesis that the effects are equal and go back to the TWFE model if we're interested in causal inference.

See Wooldridge, "Correlated random effects models with unbalanced panels".

Split or no split?

As an alternative to a null hypothesis test, we can estimate different versions of the model to see which fits best.

```
cr_pov_only <- wbm(anti ~ pov | self + momage + gender + childage +  # note placement of |
                     hispanic + black + momwork + married,
                   model = "contextual",                              # type of model
                   REML = FALSE,                                      # ML
                   use.wave = TRUE,                                   # include time
                   wave.factor = TRUE,                                # include time as dummies
                   data = nlong)
```

```
cr_self_only <- wbm(anti ~ self | pov + momage + gender + childage + # note placement of |
                      hispanic + black + momwork + married,
                    model = "contextual",                             # type of model
                    REML = FALSE,                                     # ML
                    use.wave = TRUE,                                  # include time
                    wave.factor = TRUE,                               # include time as dummies
                    data = nlong)
```

Comparing model fit

```
BIC(mixed_reg,
    cr_pov_only,
    cr_self_only,
    cr_mod) |>
  format(scientific = FALSE)
```

df	BIC
14	5958.885
15	5958.460
15	5964.099
16	5963.856

The lowest relative BIC value indicates that the mixed model and the model that splits pov into two variables are essentially both equally plausible. Both models are better than the other two, although the differences aren't massive.

Limited dependent variables

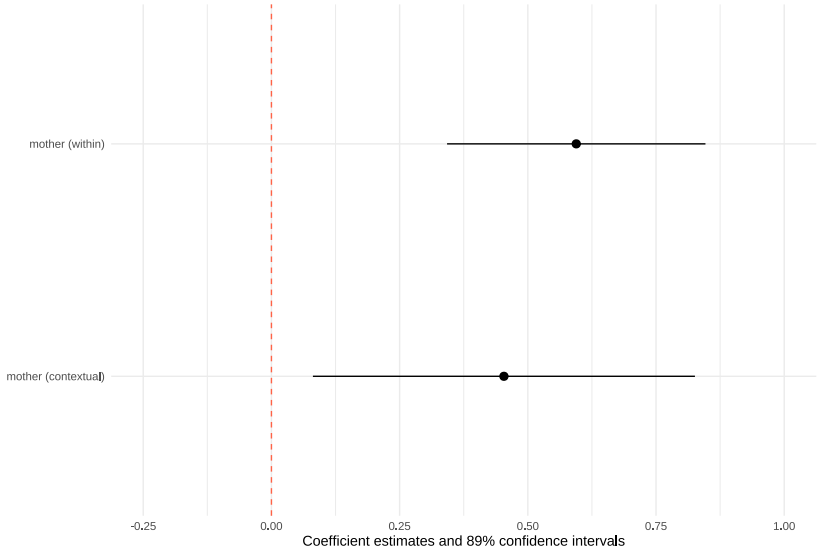
We can also use `panelr::wbm()` to estimate correlated random effects models using generalized linear models. Let's return (with all necessary caveats) to the example of motherhood and poverty from the `teen_pov` data.

```
cr_pov_logit <- wbm(pov ~ mother + spouse + inschool + hours | age + black + factor(t),
                    family = "binomial",
                    model = "contextual",
                    data = teen_pov)
```

We can estimate this model in (almost) the usual way. I found I had to specify the time variable manually because the `use.wave` and `wave.factor` arguments didn't seem to be working.

Visualizing the results

We can use `modelsummary::modelplot()` to visualize the relevant coefficients if we like. Check out `ldar-slides.Rmd` for the code.



The within coefficient (β_W) indicates that becoming a mother (for those that do), increases the odds of poverty by $e^{.594} = 81\%$ net of time-constant characteristics and the time-varying adjustment variables. We interpret this coefficient exactly like any coefficient from logistic regression.

The contextual coefficient (β_C) shows that the within and between effects are different ($\beta_W + \beta_B = \beta_C$) . This means we should probably stick with the within coefficient only for causal inference.

More flexibility with mixed models

We have now seen how we can selectively use "fixed effects" (i.e., within-subject) estimation in the context of a mixed model. This can be useful for a variety of reasons. The most obvious are:

1. Using random slopes for time to allow every individual to have their own trajectory. This probably only makes sense with a relatively high number of time periods.
2. Using random slopes on the treatment to allow the effect of the treatment to vary by individual.

Convergence problems

When we estimate complex models with `lmer()` or (especially) `glmer()`, we may run into convergence problems. Sometimes these warnings are false positives. To facilitate fit, we can try a couple of basic steps:

1. Center and scale all *continuous* predictors
2. Try all available optimizers to ensure agreement

A challenging model

This model has random slopes for time (`t0`) and `mother` and is a logistic correlated random effects model.

```
# prep
teen_pov <- teen_pov |>                 # panel data already grouped
  mutate(t0 = as.integer(t-1),          # scale time to run 0-4
         mean_mother = mean(mother))    # create mean of mother indicator for CRE
```

```
# model
tough_model <-
  glmer(
    pov ~ mother + mean_mother + t0 + age + black +
      spouse + inschool + hours + (t0 + mother | id),
    data = teen_pov,
    family = "binomial"
  )
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: pov ~ mother + mean_mother + t0 + age + black + spouse + inschool +
##     hours + (t0 + mother | id)
##    Data: teen_pov
##
##      AIC      BIC   logLik deviance df.resid
##   6846.9   6946.8  -3408.4   6816.9     5740
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2878 -0.5944 -0.4056  0.7107  4.5057
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  id     (Intercept) 1.69608  1.3023
##         t0          0.02602  0.1613   -0.77
##         mother      0.14413  0.3796    0.26  0.41
## Number of obs: 5755, groups:  id, 1151
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.389397   0.744626   0.523  0.60101
## mother       0.523514   0.163878   3.195  0.00140 **
## mean_mother  0.967698   0.220751   4.384 1.17e-05 ***
## t0           0.083018   0.030180   2.751  0.00595 **
## age         -0.092155   0.046524  -1.981  0.04761 *
## black        0.557612   0.096909   5.754 8.72e-09 ***
## spouse      -1.204969   0.154721  -7.788 6.81e-15 ***
## inschool    -0.090292   0.097986  -0.921  0.35680
## hours       -0.025948   0.002851  -9.100  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.219547 (tol = 0.002, component 1)
```

Note the warning at the very bottom.

Try rescaling

You really want variables to be measured on the same order of magnitude. One way to do this is to rescale continuous variables to range between 0 (their minimum) and 1 (their maximum).

```
# rescale age and hours
teen_pov <- teen_pov |>
  unpanel() |>
  mutate(age0 = (age - min(age)) / (max(age) - min(age)),
         hours0 = (hours - min(hours)) / (max(hours) - min(hours))) |>
  panel_data(wave = t, id = id)
```

```
# reestimate model
tough_model2 <- update(
  tough_model,
  formula = pov ~ mother + mean_mother + t0 + age0 + black +
    spouse + inschool + hours0 + (t0 + mother | id)
)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: pov ~ mother + mean_mother + t0 + age0 + black + spouse + inschool +
##     hours0 + (t0 + mother | id)
##    Data: teen_pov
##
##      AIC      BIC   logLik deviance df.resid
##   6847.6   6947.4  -3408.8   6817.6     5740
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3398 -0.5952 -0.4094  0.7182  4.5809
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  id     (Intercept) 1.58860  1.2604
##         t0          0.01604  0.1267   -0.81
##         mother      0.28635  0.5351    0.20 -0.10
## Number of obs: 5755, groups:  id, 1151
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.86541    0.15512  -5.579 2.42e-08 ***
## mother       0.52956    0.16260   3.257  0.00113 **
## mean_mother  0.95822    0.21450   4.467 7.92e-06 ***
## t0           0.07663    0.02987   2.566  0.01030 *
## age0        -0.28524    0.13899  -2.052  0.04015 *
## black        0.54686    0.09339   5.856 4.74e-09 ***
## spouse      -1.21033    0.15215  -7.955 1.79e-15 ***
## inschool    -0.10295    0.09735  -1.057  0.29029
## hours0      -2.34247    0.25456  -9.202  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

This often helps, but we're still getting an error.

Comparing optimizers

We can compare the results of several different optimizing algorithms using `lme4::allFit()`. If they all agree, then it's probably a false positive warning.

```
tough_model_all <- allFit(tough_model2)
tough_model_summary <- summary(tough_model_all)
```

```
tough_model_summary$which.OK
```

```
##                          bobyqa                    Nelder_Mead
##                            TRUE                           TRUE
##                       nlminbwrap                          nmkbw
##                            TRUE                           TRUE
##                optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
##                            TRUE                           TRUE
##     nloptwrap.NLOPT_LN_BOBYQA
##                            TRUE
```

```
tough_model_summary$fixef
```

```
##                                (Intercept)    mother mean_mother         t0
## bobyqa                          -0.8970290 0.5265632   0.9744952 0.08344553
## Nelder_Mead                     -0.8950012 0.5267259   0.9735789 0.08310606
## nlminbwrap                      -0.8864162 0.5266770   0.9638012 0.08254845
## nmkbw                           -0.8968709 0.5264512   0.9744325 0.08344667
## optimx.L-BFGS-B                 -0.8863252 0.5265664   0.9637173 0.08250274
## nloptwrap.NLOPT_LN_NELDERMEAD   -0.8970520 0.5265607   0.9745042 0.08345033
## nloptwrap.NLOPT_LN_BOBYQA       -0.8969499 0.5265921   0.9744386 0.08343271
##                                      age0     black    spouse    inschool
## bobyqa                          -0.2888162 0.5544826 -1.204885 -0.08761428
## Nelder_Mead                     -0.2891263 0.5545114 -1.205370 -0.08883658
## nlminbwrap                      -0.2878628 0.5476631 -1.196871 -0.08647998
## nmkbw                           -0.2888693 0.5543850 -1.204789 -0.08767111
## optimx.L-BFGS-B                 -0.2879519 0.5478526 -1.196820 -0.08646258
## nloptwrap.NLOPT_LN_NELDERMEAD   -0.2888116 0.5544832 -1.204873 -0.08760035
## nloptwrap.NLOPT_LN_BOBYQA       -0.2888494 0.5545011 -1.204896 -0.08765834
##                                    hours0
## bobyqa                          -2.329323
## Nelder_Mead                     -2.330640
## nlminbwrap                      -2.321359
## nmkbw                           -2.329263
## optimx.L-BFGS-B                 -2.321374
## nloptwrap.NLOPT_LN_NELDERMEAD   -2.329332
## nloptwrap.NLOPT_LN_BOBYQA       -2.329296
```

Getting marginal effects

Model convergence looks good so we can proceed. Probably the most realistic effect to estimate here is the ATT, which means the effect for those who were "treated". We can get marginal effects by restricting the data to those who spend some time in both treatment statuses using the new `marginaleffects` package by Vincent Arel-Bundock.

```
# restricted dataset
just_mothers <- teen_pov |>
  filter(mean_mother > 0 & mean_mother < 1 )

# calculate and summarize AMEs for treated
mfx <- marginaleffects(tough_model2, variables = "mother", newdata = just_mothers)
summary(mfx)
```

type	term	estimate	std.error	statistic	p.value	conf.low	conf.high
response	mother	0.105	0.0312	3.37	0.000741	0.0441	0.166

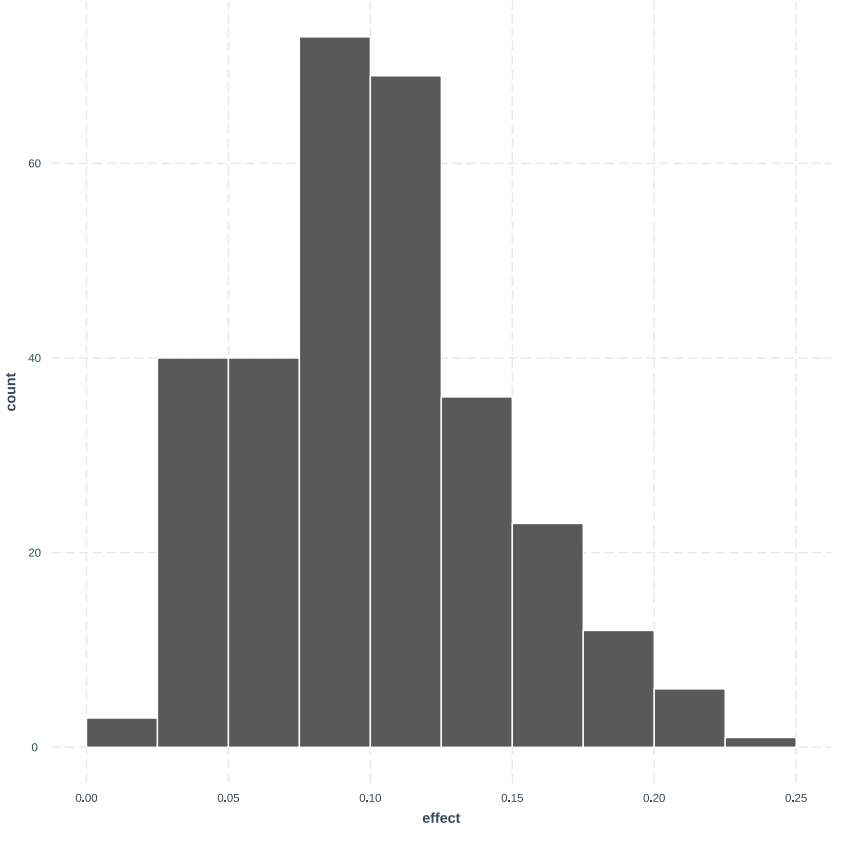
The ATT estimate here is 10.5 percentage point increase in poverty risk.

Treatment heterogeneity

Because the model allowed the effect to be variable, we visualize those differences.

```
mfx |>
  filter(mother == 1) |>              # keep treated spells
  group_by(id) |>                     # weight per person
  summarize(effect = mean(dydx)) |>   # mean effect | treat
  ggplot(aes(x = effect)) +
  geom_histogram(color = "white",
                 boundary = 0,
                 binwidth = .025) +
  theme_nice()
```

This code will show the distribution of estimated effects for individual respondents in the sample.



Summary: WB and CRE

Pros

- We can extend the mixed model to have all the strengths of the traditional "fixed effects" model
- These extensions provide additional features
  - individual growth curves (random coefficient on time)
  - heterogeneous effects (random coefficient on treatment)
  - "Hausman"-like tests for all kinds of models

Cons

- Can require additional data prep
- Harder to estimate
- Harder to interpret

Dynamic panel models

Dynamic models

- So far, we have allowed treatments to be a function of observed and unobserved variables
- We have not, however, considered that treatments might be a function of (1) previous values of the outcome; or (2) previous values of the treatment
- In this final section, we will explore this situation

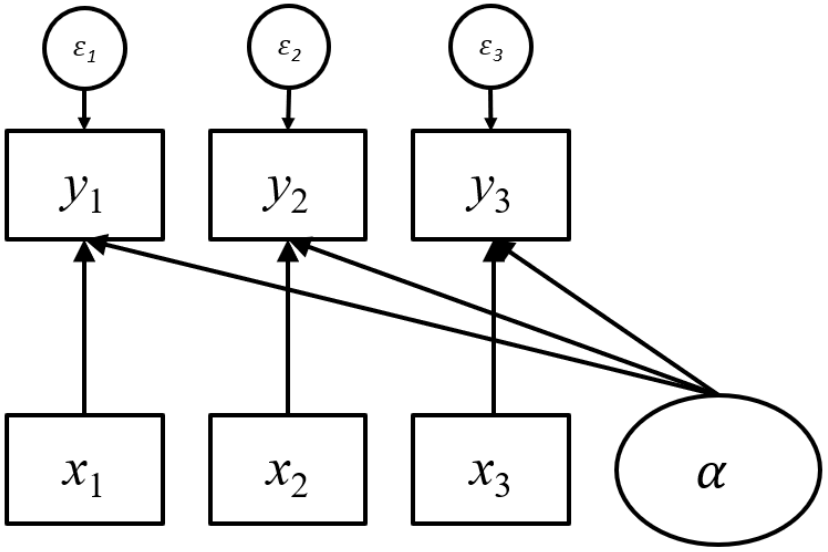
What does dynamic mean?

- A model is *dynamic* when an outcome affects itself (and possibly other variables) at a later time
- One powerful way to estimate dynamic models is using structural equation models

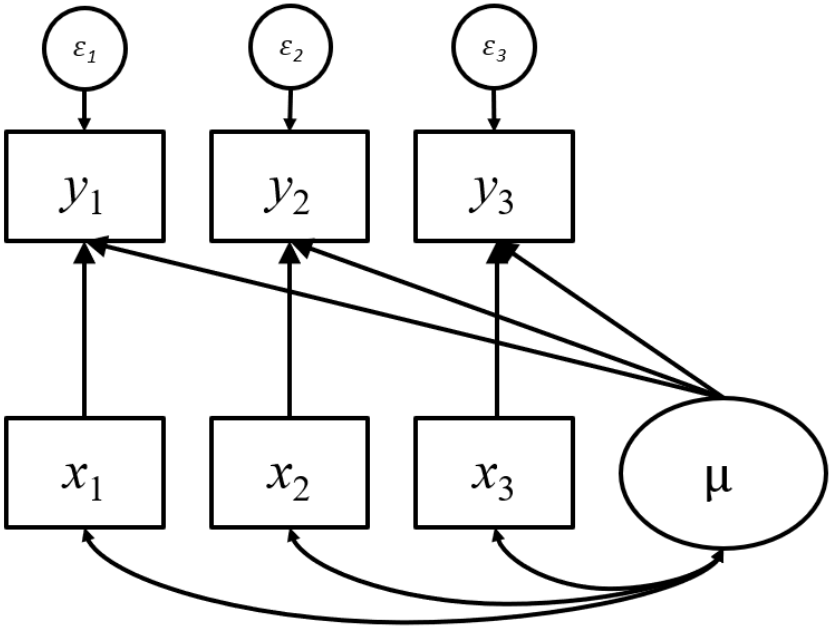
See Allison, P. D., Williams, R., & Moral-Benito, E. "Maximum likelihood for cross-lagged panel models with fixed effects".

Mixed and TWFE models in SEM

Mixed model

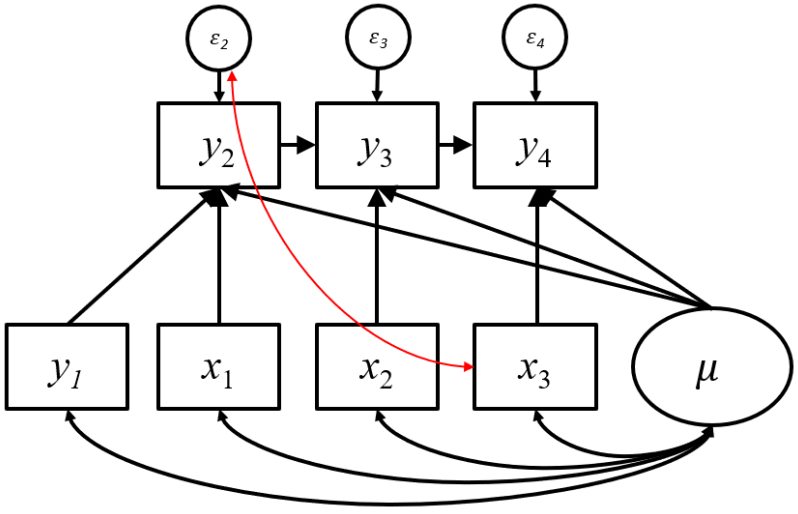


TWFE



Note: all observed *exogenous* variables (x_1, x_2, x_3) are correlated with each other as well.

More complex assumptions



- **Strict exogeneity**: a predictor variable is independent of *past* and *future* values of the outcome (all models so far have assumed this)
- **Sequential exogeneity**: a predictor variable is independent of *future* values of the outcome

The red curved line in the previous slide is the difference between these two assumptions. If it's *absent*, it implies x_3 is *strictly exogenous*. If it's *present*, it implies x_3 is *sequentially exogenous*.

Estimating using `dpm`

Jacob Long has developed a package to estimate these models without dozens of lines of `lavaan` code.

```
library(dpm)
dpm_wks_seq <- dpm(wks ~ pre(lag(union)) + lag(lwage) | ed ,
                   data = d,
                   error.inv = TRUE,     # same error variance each wave
                   estimator = "MLM")    # robust SEs
dpm_wks_str <- dpm(wks ~ lag(union) + lag(lwage) | ed,
                   data = d,
                   error.inv = TRUE,
                   estimator = "MLM")
```

`pre()` means "predetermined" (sequentially exogenous). Variables before the bar are time-varying. Variables after the bar are time-constant. If you put a time-varying variable after the bar, it uses the value in the first year (so `exp` is *starting* experience here).

Comparing models

	Sequential		Strict	
union (t - 1)	-1.206	(0.930)	-0.891	(0.751)
lwage (t - 1)	0.588	(0.589)	0.572	(0.591)
ed	-0.107	(0.073)	-0.091	(0.070)
wks (t - 1)	0.188	(0.037)	0.186	(0.037)

The evidence for a union effect on weeks of work is quite weak. The coefficients aren't much larger than the standard errors.

Getting the lags right

In the last model, we lagged the predictors but there is no guarantee that the lags in the world correspond to the lags in your data.

For example, why should *last year's* union membership predict *this year's* weeks of work? If I pour a bucket of water on you today, will this predict how wet you are 365 days from now?

If we get the lags wrong, we will get biased coefficients; sometimes, we'll even get coefficients with the *wrong sign!* *

*See Vaisey and Miles, "What you can and can't do with three-wave panel data"

Contemporaneous predictor

```
dpm_wks_ctp <- dpm(
  wks ~ pre(union) + lwage | ed ,
  data = d,
  error.inv = TRUE, # same error variance each wave
  estimator = "MLM" # robust SEs
)
```

	Contemporaneous	
union	1.423	(1.419)
lwage	0.284	(0.806)
ed	0.033	(0.092)
wks (t - 1)	0.186	(0.037)

The coefficient is still small relative to the standard error but it has changed signs when we change the lag of the variable. This is real data so we don't know the true effect, if any.

Simulated data

In the code, I create a simulated dataset (N = 100, T = 3) where the true effect of X on Y is "contemporaneous" and equal to 1. If we lagged X to try and isolate causal order, we'd get this:

```
dpm_sim_lag <- dpm(y ~ lag(x),
                   y.lag = 0,          # no autocorrelation
                   error.inv = TRUE,   # equal variance across waves
                   data = sim_df)
dpm_sim_now <- update(dpm_sim_lag,
                      formula = y ~ x)
```

	Lagged		Contemp.	
x (t - 1)	-0.678	(0.166)		
x			0.975	(0.082)

The coefficient is not only wrong, but "statistically significant" and in the opposite direction from the true effect!

Best practices

Even if you believe that a lagged predictor is appropriate, you *must* also include the contemporaneous version of the predictor to ensure the lagged coefficient is correctly estimated.*

	Lagged		Contemp.		Both	
x (t - 1)	-0.678	(0.166)			-0.004	(0.155)
x			0.975	(0.082)	1.079	(0.136)

However, doing this means that you have to say goodbye to using the data to establish causal order!

* See Leszczensky and Wolbring, "How to deal with reverse causality using panel data"

Conclusion

Panel data questions

1. What is my research question?
2. What is my treatment effect of interest?
3. Where is the variance in my outcome?
4. Where is the variance in my "treatment"?
    - How many subjects change?
    - When do they change?
    - Do they only change in one direction?
5. How can I ensure the best counterfactual comparisons?
    - Use within-subjects comparisons where you can
    - Use time to consider counterfactual trajectories
6. What assumptions is my model making?

Additional resources

Allison, *Fixed Effect Regression Models*

- short and clear explanations of FE/RE/W-B differences
- introduction to SEM for panel data
- Stata and Mplus code

Morgan and Winship, *Counterfactuals and Causal Inference*

- good on counterfactuals and defining treatment effects
- chapter 11 combines panel data with counterfactuals

Finch, Bolin, and Kelly, *Multilevel Modeling Using R*

- Good resource for R code (`lme4` and `nlme`)
- One dedicated chapter on longitudinal data

Snijders and Bosker, *Multilevel Analysis*

- Great for more advanced intuition and definitions
- One chapter on longitudinal data
- Not much code