Lecture 8: Regression Analysis

James Sears*
AFRE 891 SS 24
Michigan State University

*Parts of these slides are adapted from **"Data Science for Economists"** by Grant McDermott and **"Advanced Data Analytics"** by Nick Hagerty.

Table of Contents

Prologue

Empirical Analysis

We've spent the first half of our course building a **foundational knowledge** of data manipulation in R.

While cleaning, wrangling, and visualizing data can be fun, it's generally not the end goal - that's usually some form of **statistical/econometric analysis**.

Over the next weeks we're going to focus on the analysis side, using both **traditional** and **non-traditional** methods.

Empirical Analysis

This Lecture: Regression

- Standard regression: `lm()`
- Fixed effects regression: **fixest** and `feols`
- Formulas
- Choice of standard errors
- Regression Tables
- Visualizing regression output
- IV Regression
- Diff-in-Diff
  - Staggered Adoption
- Event Study and Sun-Abraham estimator

Next Lecture: Synthetic Control Methods

- Canonical Synthetic Control
- Synthetic Diff-in-Diff
  - Uniform Adoption
  - Staggered Adoption
- Partially Pooled Synthetic Control

Prologue

Packages we'll use today:

```
remotes::install_github("lrberge/fixest")
```

```
if (!require("pacman")) install.packages("pacman")
pacman::p_load(broom, dslabs, fixest, tictoc, tidyverse)
```

as well, let's load data necessary to replicate results from **Taylor and Druckenmiller AER (2022)**[1]

```
wl_df ← read_csv("data/zip_PANEL_data.csv") %>%
  drop_na(county_fips, state_fips, claims)
```

[1] There is a *lot* of great stuff in this paper and it's all super replicable and well-documented. I highly recommend checking it out if you're curious (only caution is that there are some *big* files in there).

Taylor and Druckenmiller AER (2022)

"Wetlands, Flooding, and the Clean Water Act" explores the benefits of wetlands in mitigating flood damage

FIGURE 2. WETLANDS AND FLOODING SCHEMATIC

- 2020 EPA decision narrowed wetland protections under Clean Water Act
- Wetlands provide a range of critical ecosystem services that benefit society
- Private benefits to landowners is relatively small

Taylor and Druckenmiller AER (2022)

"Wetlands, Flooding, and the Clean Water Act" explores the benefits of wetlands in mitigating flood damage

FIGURE 2. WETLANDS AND FLOODING SCHEMATIC

This paper estimates the **value of wetlands for flood mitigation**

Let's explore this using regression.

Basic Regression Analysis

Basic Regression Analysis

Suppose we want to estimate the following model:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + u_i$$

We can use base R's `lm()` command (**l**inear **m**odels) to run basic regression analysis:

```
lm(y ~ x1 + x2 + x3, data = df)
```

Two arguments:

1. `formula`: the equation to estimate (`y ~ x1 + x2 + x3`)
2. `data source`: the dataframe containing the variables (`df`)

Basic Regression Analysis

Let's run a basic simple linear regression of national flood insurance claims on total wetland area in zipcode i:

```
floodreg_1 ← lm(claims ~ wetland_ha, data = wl_df)
```

This adds an `lm` object to memory containing a bunch of useful info

```
View(floodreg_1)
```

Basic Regression Analysis

Use the generic `summary()` function to get fairly standard regression output:

```
summary(floodreg_1)
```

```
##
## Call:
## lm(formula = claims ~ wetland_ha, data = wl_df)
##
## Residuals:
##       Min         1Q    Median         3Q       Max
##  -1657005   -105263   -101902   -101588 418101306
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.016e+05  9.049e+03   11.23  < 2e-16 ***
## wetland_ha  5.948e+00  1.918e+00    3.10  0.00193 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3155000 on 131586 degrees of freedom
```

Basic Regression Analysis

To retrieve just the **coefficients table:**

- Point estimates and standard errors
- t stat and p-value (for $\beta = 0$)
- Stored as a `matrix`

```
reg1_coef ← summary(floodreg_1)$coefficients
reg1_coef
```

```
##                 Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 1.015883e+05 9048.768214 11.226756 3.107159e-29
## wetland_ha  5.947641e+00    1.918354  3.100388 1.933083e-03
```

Basic Regression Analysis

Often we'll want to convert the regression output to a **tidy dataframe**.

We can do this using the `tidy()` function from the **broom** package:

```
tidy(floodreg_1, conf.int = TRUE)
```

```
## # A tibble: 2 × 7
##   term         estimate std.error statistic  p.value conf.low conf.high
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept) 101588.     9049.        11.2  3.11e-29 83853.   119324.
## 2 wetland_ha       5.95      1.92       3.10 1.93e- 3     2.19       9.71
```

broom

Note that **broom** has several other useful functions too.

For instance: `glance()` summarizes model metadata (R^2, AIC, etc.) in a data frame:

```
glance(floodreg_1)
```

```
## # A tibble: 1 × 12
##   r.squared adj.r.squared    sigma statistic p.value    df  logLik    AIC   
##       <dbl>         <dbl>    <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <
## 1 0.0000730     0.0000654 3154992.      9.61 0.00193     1 -2.16e6 4.31e6 4.
## # ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Fast Fixed Effects Regression

Fast Fixed Effects Regression

Our first regression suggests that a one hectare increase in total wetland area is associated with a $2-10 increase in zipcode-level NFIP claims.

Q: what's the problem with this regression?

A: okay, there's definitely more than one. Here are several that immediately stand out:

- Not accounting for the panel nature of the data
- Specification of standard errors to account for potential error correlation
- Other zipcode characteristics likely associated with NFIP claims and wetland area

Fast Fixed Effects Regression

Most regressions you'll run will involve some kind of **fixed effects**

- unit, time, or both (or multiple in each dimension, or interactive...)

Fortunately for us, the **fixest** ("fast fixed effects") package exists.

- `feols()` makes estimating models with large number of fixed effects quick and painless
  - Like orders of magnitude faster than alternatives (`lm()`, `reghdfe` in Stata)
- Handles simple regressions
- Direct computation of alternate standard errors
- Methods for Diff-in-Diff, IV, event studies, and other estimators
  - GLM, (nonlinear Max) Likelihood, Sun-Abraham,
- Wide range of tools to make producing tables and visuals easy

Fast Fixed Effects Regression

First, let's replicate the simple model we ran before using `feols()`:

```
floodreg_1b ← feols(claims ~ wetland_ha, data = wl_df)
```

Conveniently, the basic syntax is exactly the same.

Fast Fixed Effects Regression

We now have a `fixest` object, which contains more than twice the elements of `lm()`.

We can view the output table similarly with `summary()`:

```
summary(floodreg_1b)
```

```
## OLS estimation, Dep. Var.: claims
## Observations: 131,588
## Standard-errors: IID
##                 Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 101588.31268 9048.76821 11.22676 < 2.2e-16 ***
## wetland_ha       5.94764    1.91835  3.10039 0.0019331 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 3,154,968.2   Adj. R2: 6.545e-5
```

Which now yields a dataframe, but we can still tidy things up to get simpler variable names (and drop the statistical significance factor)

Fast Fixed Effects Regression

Access the coefficients table with `coeftable()`

```
coeftable(floodreg_1b)
```

```
##                 Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 1.015883e+05 9048.768214 11.226756 3.107159e-29
## wetland_ha  5.947641e+00    1.918354  3.100388 1.933083e-03
## attr(,"type")
## [1] "IID"
```

Fast Fixed Effects Regression

View the standard errors individually with `se()`

```
se(floodreg_1b)
```

```
## (Intercept)  wetland_ha
## 9048.768214    1.918354
```

and confidence intervals with `confint()`

```
confint(floodreg_1b, level = 0.95) # get 95% Conf. Int.
```

```
##                     2.5 %       97.5 %
## (Intercept) 83852.889735 1.193237e+05
## wetland_ha      2.187702 9.707581e+00
```

Fast Fixed Effects Regression

Easily access fitted values and residuals with `$fitted.values` and `$residuals`, and the input data itself with `fixest_data()`

```
data.frame(
    val = fixest_data(floodreg_1b)$claims[1:10],
    val_hat = floodreg_1b$fitted.values[1:10],
    resid = floodreg_1b$residuals[1:10]
)
```

Adding Fixed Effects

`feols()` makes it easy to add any sort of **fixed effects** you want to your regression:

```
feols(y ~ x | fe_1 + ... + fe_n, data = df)
```

- Add a `|` after your formula and add any fixed effects on the RHS.

For example, if we want to add state fixed effects to our wetlands model:

```
floodreg_2 ← feols(claims ~ wetland_ha | state_fips, data = wl_df)
coeftable(floodreg_2)
```

```
##              Estimate Std. Error    t value  Pr(>|t|)
## wetland_ha -6.34305   8.057274 -0.7872452 0.4347817
## attr(,"type")
## [1] "Clustered (state_fips)"
```

Viewing Fixed Effects

By default, we don't get the fixed effects estimates presented as coefficients - rather they're netted off in a computationally-efficient fashion prior to estimation.

If you care about the fixed effects coefficients themselves, you can recover them with `fixef()`:

```
fixef(floodreg_2)
```

```
## $state_fips
##             1             2             4             5             6
##   57684.8787     745.5667    4892.8401   26938.5931    9804.6145    6686.187
##             9            10            11            12            13
##    2560.6355   45500.1135    4333.5653  221050.0204   34224.6055   26147.360
##            16            17            18            19            20
##    6871.2363    8368.7442    6729.8810    7249.2228    5313.1758   14021.859
##            22            23            24            25            26
## 2843327.9761   13568.0202   14224.4683    3026.6441   23616.1233   38000.096
##            28            29            30            31            32
```

Adding Fixed Effects

The resulting `fixest` object contains a lot of information on our fixed effects, including

- What variables were used for each fixed effect (`fixef_vars`)
- Coefficient size for each fixed effect dimension (`fixed_size`)

If instead we wanted state and year fixed effects:

```
feols(claims ~ wetland_ha | state_fips + year, data = wl_df) %>%
  coeftable()
```

```
##              Estimate Std. Error    t value  Pr(>|t|)
## wetland_ha -6.353344    8.06899 -0.7873778 0.4347048
## attr(,"type")
## [1] "Clustered (state_fips)"
```

Adding Fixed Effects

Or instead interacted county-year fixed effects:[1]

```
floodreg_int_fe ← feols(claims ~ wetland_ha | county_fips^year, data = w
coeftable(floodreg_int_fe)
```

```
##              Estimate Std. Error    t value  Pr(>|t|)
## wetland_ha -0.8240054   2.026503 -0.4066144 0.6842981
## attr(,"type")
## [1] "Clustered (county_fips^year)"
```

```
floodreg_int_fe$fixef_size
```

```
## county_fips^year
##            12524
```

[1] Note that the standard errors have defaulted to clustering at the level of the fixed effects - we'll chat more about this shortly.

Multiple Estimation

`feols()` also makes it easy to run **multiple estimation**.[2]

Using both claims $ and housing values as dependent variables:

```
feols(c(claims, housing_value) ~ wetland_ha | county_fips^year, data = wl
  summary()
```

```
## Standard-errors: Clustered (county_fips^year)
## Dep. var.: claims
##              Estimate Std. Error   t value Pr(>|t|)
## wetland_ha -0.824005     2.0265 -0.406614   0.6843
## ---
## Dep. var.: housing_value
##             Estimate Std. Error  t value  Pr(>|t|)
## wetland_ha -0.60365   0.069329 -8.70706 < 2.2e-16 ***
```

[2] See the handy **multiple estimation vignette** for more useful multiple estimation tools (e.g. stepwise inclusion of covariates).

Programmatic Formulas

Programmatic Formulas

There are time we might want to avoid the usual syntax and use a **programmatic formula** for our regression.

Suppose we wanted to see the separate effect of woody vs. herbaceous wetland acreage, and include both state and year fixed effects. Instead of typing the formula, we can put it together programmatically:

```
# first, specify the formula components as strings/vectors
depvar ← "claims"

# Use the collapse argument of paste() to combine elements together with
x ← c("wetland_herb_ha", "wetland_woody_ha") %>% paste(collapse = " + ")
fe ← c("state_fips", "year")  %>% paste(collapse = " + ")

# Combine it all together into a formula
form ← paste(depvar, "~", x, "|", fe) %>%
  as.formula()

form
```

Programmatic Formulas

Once we have a formula object, we can use it as the formula object of `feols()` (or `lm()`):

```
# Use the formula as the first argument of feols
feols(form, data = wl_df) %>% summary()
```

```
## OLS estimation, Dep. Var.: claims
## Observations: 131,588
## Fixed-effects: state_fips: 52,  year: 4
## Standard-errors: Clustered (state_fips)
##                  Estimate Std. Error  t value Pr(>|t|)
## wetland_herb_ha   18.3309    5.72681  3.20090 0.002359 **
## wetland_woody_ha -16.1681   12.80388 -1.26275 0.212421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 3,132,626.1     Adj. R2: 0.013765
##                    Within R2: 2.742e-4
```

Standard Errors

Standard Errors

Let's take a few minutes to chat about the statistical elephant in the room: **standard errors**.

By default, `lm()` uses classical/homoskedastic (IID) standard errors, which you basically should **never** use with observational data.

A distinct advantage of `feols()` is its built-in functionality for **alternate standard errors**.

- Heteroskedasticity-robust ("White")
- Clustered
  - Up to four-way
- Newey-West (heteroskedastic and serially correlated errors)
- Driscoll Kray (cross-sectionally and serially correlated)
- Conley (spatially correlated)

Standard Errors

Many empirical economic studies report **heteroskedasticity-robust** standard errors

- Also known as White or Eicker-Huber-White standard errors

These "Robust" standard errors are **agnostic to the form of error dependency**.

A common alternative is **clustered standard errors**, which are often rationalized through **residual dependency across units in the same cluster**.

- i.e.luster by the geographic unit

Standard Errors

Often these standard errors can differ. By a *lot*.

Let's revisit our first OLS estimate. Recall the **homoskedastic** standard errors:

```
floodreg_iid ← feols(claims ~ wetland_ha, data = wl_df, vcov = "iid")
coeftable(floodreg_iid )
```

```
##                 Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 1.015883e+05 9048.768214 11.226756 3.107159e-29
## wetland_ha  5.947641e+00    1.918354  3.100388 1.933083e-03
## attr(,"type")
## [1] "IID"
```

```
se_iid ← coeftable(floodreg_iid)[2,2] %>% round(4)
```

Standard Errors

If we instead use robust standard errors:

```
floodreg_rob ← feols(claims ~ wetland_ha, data = wl_df,
      vcov = "hetero")
coeftable(floodreg_rob)
```

```
##                 Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 1.015883e+05 8975.872203 11.317932 1.104036e-29
## wetland_ha  5.947641e+00    1.563318  3.804499 1.421557e-04
## attr(,"type")
## [1] "Heteroskedasticity-robust"
```

```
se_rob ← coeftable(floodreg_rob)[2,2] %>% round(4)
```

Standard Errors

Alernatively, we may believe that there is likely correlation in error terms for zip codes within the same **county**

```
floodreg_cl_cty ← feols(claims ~ wetland_ha, data = wl_df,
      cluster = ~ county_fips)
coeftable(floodreg_cl_cty)
```

```
##                 Estimate   Std. Error  t value     Pr(>|t|)
## (Intercept) 1.015883e+05 28861.848410 3.519813 0.0004380073
## wetland_ha  5.947641e+00     2.461991 2.415785 0.0157582439
## attr(,"type")
## [1] "Clustered (county_fips)"
```

```
se_cl_cty ← coeftable(floodreg_cl_cty)[2,2] %>% round(4)
```

Standard Errors

Or we might be worried that state policies are determining the correlation:

```
floodreg_cl_st ← feols(claims ~ wetland_ha, data = wl_df,
      cluster = ~ state_fips)
coeftable(floodreg_cl_st)
```

```
##                 Estimate   Std. Error t value  Pr(>|t|)
## (Intercept) 1.015883e+05 48232.080206 2.10624 0.0401248
## wetland_ha  5.947641e+00     3.851649 1.54418 0.1287276
## attr(,"type")
## [1] "Clustered (state_fips)"
```

```
se_cl_st ← coeftable(floodreg_cl_st)[2,2] %>% round(4)
```

Standard Errors

Or that the year also matters:

```
floodreg_cl_styr ← feols(claims ~ wetland_ha, data = wl_df,
      cluster = ~ state_fips+year)
coeftable(floodreg_cl_styr)
```

```
##                 Estimate   Std. Error  t value   Pr(>|t|)
## (Intercept) 1.015883e+05 41784.653908 2.431235 0.09322706
## wetland_ha  5.947641e+00     4.517639 1.316538 0.27953875
## attr(,"type")
## [1] "Clustered (state_fips & year)"
```

```
se_cl_styr ← coeftable(floodreg_cl_styr)[2,2] %>% round(4)
```

Standard Errors

The choice of standard error **dramatically affects statistical significance.**

Standard Error	Estimate	P-Value
Conventional	1.9184	0.0019
Robust	1.5633	0.0014
Clustered (County)	2.462	0.0158
Clustered (State)	3.8516	0.1287
Clustered (State + Year)	4.5176	0.2795

State-clustered errors are 2.46 times larger than robust ones. So, which do we choose?

Clustered Standard Errors

When it comes to clustering, it turns out that there are three main misconceptions.[3]

1: The need for clustering hinges on the presence of non-zero correlation between residuals for units in the same cluster.

- Presence of non-zero correlation isn't sufficient to justify clustering
- Absence of this correlation doesn't rule out clustering as necessary

[3] I highly recommend reading **"When You Should Adjust Standard Errors for Clustering?" by Abadie et al.** in QJE in 2023 for complete discussion.

Clustered Standard Errors

When it comes to clustering, it turns out that there are three main misconceptions.[3]

2: If clustering standard errors makes a difference, you should do it

- Leads to overly-conservative standard errors in contexts where clustering isn't justifiable
  - i.e. Random sample across clusters but outcomes positively correlated within clusters

[3] I highly recommend reading **"When You Should Adjust Standard Errors for Clustering?" by Abadie et al.** in QJE in 2023 for complete discussion.

Clustered Standard Errors

When it comes to clustering, it turns out that there are three main misconceptions.[3]

3: There are only two choices: fully adjust for clusters, or use robust SEs

- Abadie et al. (2023) show that you can make substantial improvements by combining both

[3] I highly recommend reading **"When You Should Adjust Standard Errors for Clustering?" by Abadie et al.** in QJE in 2023 for complete discussion.

Clustered Standard Errors

In short, it's the **sampling process** and the **treatment assignment mechanism** that solely determine the correct level of clustering.

No clustered sampling:

- If you have a random sample with random treatment assignment, clustering isn't appropriate
- If the sample is a large fraction of the population and treatment effects are heterogeneous across units, robust standard errors are already conservative

Clustered Standard Errors

In short, it's the **sampling process** and the **treatment assignment mechanism** that solely determine the correct level of clustering.

Clustered assignment:

- If assignment is perfectly clustered (all units in same cluster have same treatment assignment), clustering is justified
- If assignment is partially clustered and cluster sizes are large, clustering is justified but standard methods can be extremely conservative

Clustered Standard Errors

In short, it's the **sampling process** and the **treatment assignment mechanism** that solely determine the correct level of clustering.

Cluster Sampling:

- If clusters themselves are sampled, then adjusting for clustering is justified
- If the fraction of sampled clusters is small relative to the total in the population, clustered variance estimator is asymptotically correct (regardless of clustered assignment)
- Same holds if the fraction of sampled clusters is high but only a small fraction of within-cluster observations are sampled
- In other cases, cluster estimates will be overly conservative

Alternative Standard Errors

`feols()` also makes it easy in case we're concerned about alternate correlation structures

- Newey-West (heteroskedastic and serially correlated errors)
  - Panel or Time Series
- Driscoll Kray (cross-sectionally and serially correlated)
  - Panel or Time Series
- Conley (spatially correlated)
- Externally-computed covariance matrices[4]
  - i.e. Abadie et al. 2023 Causal cluster variance (CCV), two-stage cluster bootstrap (TSCB)

[4] The **sandwich** package allows for a wide range of clustered and bootstrapped standard errors.

Conley Standard Errors

If we believe in error dependency based on **spatial proximity**, we can correct for that using **Conley standard errors**.

For example, if we think this dependency likely extends up to 100km, we can specify Conley standard errors as

```
feols(claims ~ wetland_ha, data = wl_df,
      vcov = conley(cutoff = 100))
```

```
## OLS estimation, Dep. Var.: claims
## Observations: 131,588
## Standard-errors: Conley (100km)
##                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept) 101588.31268 47509.01354 2.13830 0.032495 *
## wetland_ha       5.94764     2.93936 2.02345 0.043029 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 3,154,968.2   Adj. R2: 6.545e-5
```

Driscoll Kray Standard Errors

Driscoll Kray standard errors provide an estimate robust to both spatial and temporal dependence.

To estimate Driscoll Kray with a 1 year lag:

```
feols(claims ~ wetland_ha, data = wl_df,
      vcov = DK ~ year)
```

```
## OLS estimation, Dep. Var.: claims
## Observations: 131,588
## Standard-errors: Driscoll-Kraay (L=1)
##                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept) 101588.31268 21118.20440 4.81046 0.017106 *
## wetland_ha       5.94764     1.75838 3.38245 0.043014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 3,154,968.2   Adj. R2: 6.545e-5
```

Driscoll Kray Standard Errors

Driscoll Kray standard errors provide an estimate robust to both spatial and temporal dependence.

Comparing to a two-year lag:

```
feols(claims ~ wetland_ha, data = wl_df,
      vcov = DK(2) ~ year)
```

```
## OLS estimation, Dep. Var.: claims
## Observations: 131,588
## Standard-errors: Driscoll-Kraay (L=2)
##                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept) 101588.31268 22330.49234 4.54931 0.019898 *
## wetland_ha       5.94764     2.00352 2.96860 0.059135 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 3,154,968.2   Adj. R2: 6.545e-5
```

Newey West Standard Errors

Newey West (1987) provide a heteroskedasticity and autocorrelation consistent covariance estimator.

To estimate Newey West standard errors, we need to specify the panel structure and the desired lag. For a two-year lag:

```
feols(claims ~ wetland_ha, data = wl_df,
      vcov = NW(2) ~ zip + year)
```

```
## OLS estimation, Dep. Var.: claims
## Observations: 131,588
## Standard-errors: Newey-West (L=2)
##                 Estimate  Std. Error t value  Pr(>|t|)
## (Intercept) 101588.31268 10445.26830 9.72577 0.0023089 **
## wetland_ha       5.94764     1.89476 3.13899 0.0517038 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 3,154,968.2   Adj. R2: 6.545e-5
```

Table of Contents