Quant II

Lab 2: Regression

Giacomo Lemoli

February 2, 2023

Today's plan

- Regression: bridging different approaches

Today's plan

- Regression: bridging different approaches
- Robust inference, part I

Today's plan

- Regression: bridging different approaches
- Robust inference, part I
- Regressions and inference in practice

Different approaches to linear regression

- Linear models are the “default” modeling choice in empirical social science

Different approaches to linear regression

- Linear models are the “default” modeling choice in empirical social science
- The choice of a linear regression model can be motivated by different sets of assumptions and statistical perspectives (Aronow and Miller, 2019)

Different approaches to linear regression

- Linear models are the “default” modeling choice in empirical social science
- The choice of a linear regression model can be motivated by different sets of assumptions and statistical perspectives (Aronow and Miller, 2019)
- Standard approaches: regression derives from structural assumptions about the DGP

Different approaches to linear regression

- Linear models are the “default” modeling choice in empirical social science
- The choice of a linear regression model can be motivated by different sets of assumptions and statistical perspectives (Aronow and Miller, 2019)
- Standard approaches: regression derives from structural assumptions about the DGP
- Modern dominating approach: “agnostic” $\Longrightarrow$ accept DGP can’t be known and make inference about it with as few assumptions as possible

Parametric approach

- Fully specify a statistical model assumed to be true

$$Y_i = X_i'\beta + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

Parametric approach

- Fully specify a statistical model assumed to be true

$$Y_i = X_i'\beta + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

- Derive the distribution for Y_i

$$Y_i \sim N(X_i'\beta, \sigma^2)$$

Parametric approach

- Fully specify a statistical model assumed to be true

$$Y_i = X_i'\beta + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

- Derive the distribution for Y_i

$$Y_i \sim N(X_i'\beta, \sigma^2)$$

- Find the *likelihood function* of the parameters

$$\mathcal{L}((\beta, \sigma)|Y_i, X_i) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(Y_i - X_i'\beta)^2}{2\sigma^2}}$$

Parametric approach

- Fully specify a statistical model assumed to be true

$$Y_i = X_i'\beta + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

- Derive the distribution for Y_i

$$Y_i \sim N(X_i'\beta, \sigma^2)$$

- Find the *likelihood function* of the parameters

$$\mathcal{L}((\beta, \sigma)|Y_i, X_i) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(Y_i - X_i'\beta)^2}{2\sigma^2}}$$

- If Y_i are not i.i.d. (e.g. unequal variances, clustered) the model is mispecified

Parametric approach

- Compute the log of the likelihood function, and solve the maximization problem for (β, σ)

Parametric approach

- Compute the log of the likelihood function, and solve the maximization problem for (β, σ)
- The value of β that maximizes the likelihood function is

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y$$

Semi-parametric approach

- Weaker restrictions on the distribution of ε: $\mathbb{E}[\varepsilon|\mathbf{X}] = 0$ and $Var[\varepsilon|\mathbf{X}] = \sigma^2\mathbb{I}$

Semi-parametric approach

- Weaker restrictions on the distribution of ε: $\mathbb{E}[\varepsilon|\mathbf{X}] = 0$ and $Var[\varepsilon|\mathbf{X}] = \sigma^2\mathbb{I}$
- The OLS estimator

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y$$

  is the **B**est **L**inear **U**nbiased **E**stimator

Agnostic approach

- Object of interest: Conditional Expectation Function, which summarizes the predictive power of one variable to another

$$G_Y(x) = \mathbb{E}[Y|X = x]$$

Agnostic approach

- Object of interest: Conditional Expectation Function, which summarizes the predictive power of one variable to another

$$G_Y(x) = \mathbb{E}[Y|X = x]$$

- We can write Y as

$$Y \equiv \mathbb{E}[Y|X = x] + \varepsilon$$
$$\varepsilon \equiv Y - \mathbb{E}[Y|X = x]$$

Agnostic approach

- Object of interest: Conditional Expectation Function, which summarizes the predictive power of one variable to another

$$G_Y(x) = \mathbb{E}[Y|X = x]$$

- We can write Y as

$$Y \equiv \mathbb{E}[Y|X = x] + \varepsilon$$
$$\varepsilon \equiv Y - \mathbb{E}[Y|X = x]$$

- Give a linear representation of the CEF (assumed or just an approximation): $\mathbb{E}[Y|X] = X'\beta$

Agnostic approach

- Then, find a Best Linear Predictor, i.e. $\hat{\beta}$ which minimizes the Mean Squared Error

$$
\begin{aligned}
&\min_{\beta} \mathbb{E}[\varepsilon_i^2] \\
&= \min_{\beta} \mathbb{E}[(Y_i - \mathbb{E}[Y_i|X_i])^2] \\
&= \min_{\beta} \mathbb{E}[(Y_i - X_i'\beta)^2]
\end{aligned}
$$

Agnostic approach

- Then, find a Best Linear Predictor, i.e. $\hat{\beta}$ which minimizes the Mean Squared Error

$$\begin{aligned} &\min_{\beta} \mathbb{E}[\varepsilon_i^2] \\ &= \min_{\beta} \mathbb{E}[(Y_i - \mathbb{E}[Y_i|X_i])^2] \\ &= \min_{\beta} \mathbb{E}[(Y_i - X_i'\beta)^2] \end{aligned}$$

- The BLP is

$$\beta^* = \mathbb{E}[X_i X_i']^{-1}\mathbb{E}[X_i Y_i]$$

Agnostic approach

- Then, find a Best Linear Predictor, i.e. $\hat{\beta}$ which minimizes the Mean Squared Error

$$\begin{aligned} &\min_\beta \mathbb{E}[\varepsilon_i^2] \\ &= \min_\beta \mathbb{E}[(Y_i - \mathbb{E}[Y_i|X_i])^2] \\ &= \min_\beta \mathbb{E}[(Y_i - X_i'\beta)^2] \end{aligned}$$

- The BLP is

$$\beta^* = \mathbb{E}[X_i X_i']^{-1}\mathbb{E}[X_i Y_i]$$

- The sample analogue to estimate it is

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y$$

Inference under the agnostic approach

- In lecture we have derived the asymptotic distribution of the OLS estimator. The sample variability is given by the asymptotic variance

$$V = \mathbb{E}[X_i' X_i]^{-1} \mathbb{E}[\varepsilon_i^2 X_i' X_i] \mathbb{E}[X_i' X_i]^{-1}$$

Inference under the agnostic approach

- In lecture we have derived the asymptotic distribution of the OLS estimator. The sample variability is given by the asymptotic variance

$$V = \mathbb{E}[X_i' X_i]^{-1} \mathbb{E}[\varepsilon_i^2 X_i' X_i] \mathbb{E}[X_i' X_i]^{-1}$$

- The estimator for the asymptotic variance has the general form

$$\hat{V}(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' \hat{\Psi} \mathbf{X} (\mathbf{X}'\mathbf{X})^{-1}$$

where $\hat{\Psi}$ is an estimator of $plim[\varepsilon\varepsilon']$. Different estimators differ by the elements $\hat{\psi}_i$ in the diagonal matrix $\hat{\Psi}$.

Inference under the agnostic approach

- In lecture we have derived the asymptotic distribution of the OLS estimator. The sample variability is given by the asymptotic variance

$$V = \mathbb{E}[X_i'X_i]^{-1}\mathbb{E}[\varepsilon_i^2 X_i'X_i]\mathbb{E}[X_i'X_i]^{-1}$$

- The estimator for the asymptotic variance has the general form

$$\hat{V}(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\hat{\Psi}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

  where $\hat{\Psi}$ is an estimator of $plim[\varepsilon\varepsilon']$. Different estimators differ by the elements $\hat{\psi}_i$ in the diagonal matrix $\hat{\Psi}$.

- These estimators are "robust" because they don't require specific distributional assumptions

Robust variance estimators in regression

From MHE, Ch.8.

- Non-robust/conventional: $\hat{\psi}_i = \hat{\psi} = \frac{\sum_{i=1}^{n} e_i^2}{n-k} = \hat{\sigma}$

Robust variance estimators in regression

From MHE, Ch.8.

- Non-robust/conventional: $\hat{\psi}_i = \hat{\psi} = \frac{\sum_{i=1}^{n} e_i^2}{n-k} = \hat{\sigma}$
- HC0: $\hat{\psi}_i = \hat{e_i}^2$

Robust variance estimators in regression

From MHE, Ch.8.

- Non-robust/conventional: $\hat{\psi}_i = \hat{\psi} = \frac{\sum_{i=1}^{n} e_i^2}{n-k} = \hat{\sigma}$
- HC0: $\hat{\psi}_i = \hat{e}_i{}^2$
- HC1: $\hat{\psi}_i = \frac{n}{n-k} \hat{e}_i{}^2$

Robust variance estimators in regression

From MHE, Ch.8.

- Non-robust/conventional: $\hat{\psi}_i = \hat{\psi} = \frac{\sum_{i=1}^{n} e_i^2}{n-k} = \hat{\sigma}$
- HC0: $\hat{\psi}_i = \hat{e}_i{}^2$
- HC1: $\hat{\psi}_i = \frac{n}{n-k} \hat{e}_i{}^2$
- HC2: $\hat{\psi} = \frac{1}{1-h_{ii}} \hat{e}_i{}^2$, where h_{ii} is the *leverage* of unit i (the influence of i on the regression line)

Robust variance estimators in regression

From MHE, Ch.8.

- Non-robust/conventional: $\hat{\psi}_i = \hat{\psi} = \frac{\sum_{i=1}^{n} e_i^2}{n-k} = \hat{\sigma}$
- HC0: $\hat{\psi}_i = \hat{e_i}^2$
- HC1: $\hat{\psi}_i = \frac{n}{n-k}\hat{e_i}^2$
- HC2: $\hat{\psi} = \frac{1}{1-h_{ii}}\hat{e_i}^2$, where h_{ii} is the *leverage* of unit i (the influence of i on the regression line)
- HC3: $\hat{\psi} = \frac{1}{(1-h_{ii})^2}\hat{e_i}^2$

Robust variance estimators

- Different approaches: in the parametric approach, robust variance estimators are introduced to correct a “bug”. Under an “agnostic” approach, it is a “natural” estimator to use given the lack of structural assumptions

Robust variance estimators

- Different approaches: in the parametric approach, robust variance estimators are introduced to correct a "bug". Under an "agnostic" approach, it is a "natural" estimator to use given the lack of structural assumptions
- Next week(s): move from inference about descriptive quantities to inference about causal quantities

Robust variance estimators

- Different approaches: in the parametric approach, robust variance estimators are introduced to correct a "bug". Under an "agnostic" approach, it is a "natural" estimator to use given the lack of structural assumptions
- Next week(s): move from inference about descriptive quantities to inference about causal quantities
- Direct correspondence between robust variance estimators for regression coefficients and variance estimators for estimators of causal effects in randomized experiments

Suggested readings

- Aronow and Miller (2019), Foundations of Agnostic Statistics, CUP
  - Equivalence between BLP and OLS estimator under linearity
  - Comparison with parametric regression approaches

Linear regression in R

- In base R, `lm()` implements the OLS estimation

Linear regression in R

- In base R, `lm()` implements the OLS estimation
- Now several different packages are available

Linear regression in R

- In base R, `lm()` implements the OLS estimation
- Now several different packages are available
- `estimatr::lm_robust()`. Developed for analysis and design of randomized experiments.

Linear regression in R

- In base R, `lm()` implements the OLS estimation
- Now several different packages are available
- `estimatr::lm_robust()`. Developed for analysis and design of randomized experiments.
    - Options for robust and cluster SE

Linear regression in R

- In base R, `lm()` implements the OLS estimation
- Now several different packages are available
- `estimatr::lm_robust()`. Developed for analysis and design of randomized experiments.
  - Options for robust and cluster SE
  - Not suitable for panel data models with many FE

Linear regression in R

- In base R, `lm()` implements the OLS estimation
- Now several different packages are available
- `estimatr::lm_robust()`. Developed for analysis and design of randomized experiments.
  - Options for robust and cluster SE
  - Not suitable for panel data models with many FE
- `lfe::felm()` and `fixest::feols()` are developed for datasets with many FE

Linear regression in R

- In base R, `lm()` implements the OLS estimation
- Now several different packages are available
- `estimatr::lm_robust()`. Developed for analysis and design of randomized experiments.
  - Options for robust and cluster SE
  - Not suitable for panel data models with many FE
- `lfe::felm()` and `fixest::feols()` are developed for datasets with many FE
  - Options for robust and cluster SE, options for IV estimation

Linear regression in Stata

- Proverbial `reg[ress]`

Linear regression in Stata

- Proverbial `reg[ress]`
- In observational studies other packages are used more often, for their handling of FE

Linear regression in Stata

- Proverbial `reg[ress]`
- In observational studies other packages are used more often, for their handling of FE
- `xtreg`, `areg`, `reghdfe`. The latter is probably superior

Robust standard errors

- R: With packages `estimatr`, `lfe`, or `fixest`, specify the SE estimator from inside the function

Robust standard errors

- R: With packages `estimatr`, `lfe`, or `fixest`, specify the SE estimator from inside the function
- With `lm()`, fit the model and then use functions in the package `sandwich` (or customized functions) to compute the standard errors

Robust standard errors

- R: With packages `estimatr`, `lfe`, or `fixest`, specify the SE estimator from inside the function
- With `lm()`, fit the model and then use functions in the package `sandwich` (or customized functions) to compute the standard errors
    - Looks annoying if you come from Stata, but allows flexibility

Robust standard errors

- R: With packages `estimatr`, `lfe`, or `fixest`, specify the SE estimator from inside the function
- With `lm()`, fit the model and then use functions in the package `sandwich` (or customized functions) to compute the standard errors
  - Looks annoying if you come from Stata, but allows flexibility
- **Always** check the function documentation to see what SEs are computed

Robust standard errors

- R: With packages `estimatr`, `lfe`, or `fixest`, specify the SE estimator from inside the function
- With `lm()`, fit the model and then use functions in the package `sandwich` (or customized functions) to compute the standard errors
  - Looks annoying if you come from Stata, but allows flexibility
- **Always** check the function documentation to see what SEs are computed
- Stata: `regress y x, vce()`, the argument of `vce()` can be `robust`, `hc2`, `hc3`, `bootstrap`, `cluster clustervar`

Robust standard errors

- R: With packages `estimatr`, `lfe`, or `fixest`, specify the SE estimator from inside the function
- With `lm()`, fit the model and then use functions in the package `sandwich` (or customized functions) to compute the standard errors
  - Looks annoying if you come from Stata, but allows flexibility
- **Always** check the function documentation to see what SEs are computed
- Stata: `regress y x, vce()`, the argument of `vce()` can be `robust`, `hc2`, `hc3`, `bootstrap`, `cluster clustervar`
- If you replicate analyses in another language, cross-check the documentation, as different adjustments may be applied

Robust standard errors

- R: With packages `estimatr`, `lfe`, or `fixest`, specify the SE estimator from inside the function
- With `lm()`, fit the model and then use functions in the package `sandwich` (or customized functions) to compute the standard errors
    - Looks annoying if you come from Stata, but allows flexibility
- **Always** check the function documentation to see what SEs are computed
- Stata: `regress y x, vce()`, the argument of `vce()` can be `robust`, `hc2`, `hc3`, `bootstrap`, `cluster clustervar`
- If you replicate analyses in another language, cross-check the documentation, as different adjustments may be applied
- Not all packages have all estimators built-in, so you may integrate the SE computation when exporting the results

Robust standard errors in R

```
## Simulate a randomized experiment
set.seed(123)
library(dplyr)

# Simulated population
pop <- data.frame(Y1 = rnorm(1000, 4, 2), Y0 = rnorm(1000, 0.5, 3))

# Random sample
sample <- pop[sample(nrow(pop), 100),]
sample$D <- 0
sample$D[sample(100, 30)] <- 1

# Observed potential outcomes
sample <- sample %>% mutate(Y = D*Y1 + (1-D)*Y0)
```

Robust standard errors in R

```r
### Example 1: lm_robust
library(estimatr)

# Default: HC2 estimator
lm_robust(Y ~ D, data = sample)
```

```
##              Estimate Std. Error   t value     Pr(>|t|)   CI Lower CI Upper DF
## (Intercept) 0.3137636  0.3844703 0.8160932 4.164261e-01 -0.4492052 1.076732 98
## D           4.1641379  0.5135960 8.1078083 1.493054e-12  3.1449234 5.183352 98
```

```r
# HC1 (Stata's default)
lm_robust(Y ~ D, data = sample, se_type = "stata")
```

```
##              Estimate Std. Error   t value     Pr(>|t|)   CI Lower CI Upper DF
## (Intercept) 0.3137636  0.3855896 0.8137243 4.177757e-01 -0.4514264 1.078954 98
## D           4.1641379  0.5128987 8.1188318 1.414176e-12  3.1463072 5.181969 98
```

```r
### Example 2: lm + lmtest + sandwich
library(sandwich); library(lmtest)
fit <- lm(Y ~ D, data = sample)
coeftest(fit, vcov = vcovHC(fit, type = "HC2"))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.31376    0.38447  0.8161    0.4164
## D            4.16414    0.51360  8.1078 1.493e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Robust standard errors in R

```r
# Default is homoskedastic
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ D, data = sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7960 -2.1130 -0.0758  2.4649  6.0472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3138     0.3446   0.910    0.365
## D             4.1641     0.6292   6.618 1.95e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.884 on 98 degrees of freedom
## Multiple R-squared:  0.3089, Adjusted R-squared:  0.3018
## F-statistic: 43.79 on 1 and 98 DF,  p-value: 1.949e-09
```

```r
N <- nrow(sample)
D <- cbind(rep(1, N), sample$D)
K <- dim(D)[2]

vcov <- solve((t(D) %*% D)) %*% t(D) %*% diag((sum(residuals(fit)^2)/(N-K)), N, N) %*% D %*% solve((t(D) %*% D)

cbind(coef(fit), sqrt(diag(vcov)))
```

```
##                  [,1]      [,2]
## (Intercept) 0.3137636 0.3446480
## D           4.1641379 0.6292383
```

Export regression output in tables:

In R, `stargazer` is perhaps still the most popular

```
library(stargazer)
stargazer(fit, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ----------------------------
##                                  Y
## ------------------------------------------------
## D                             4.164***
##                                (0.629)
##
## Constant                        0.314
##                                (0.345)
##
## ------------------------------------------------
## Observations                     100
## R2                              0.309
## Adjusted R2                     0.302
## Residual Std. Error       2.884 (df = 98)
## F Statistic           43.795*** (df = 1; 98)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

Export regression output in tables:

`modelsummary` is very flexible

```
library(modelsummary)
modelsummary(list(fit, fit),
             vcov=list("stata", "HC2"),
             output = "markdown", gof_omit = "^(?!R2|Num)")
```

	Model 1	Model 2
(Intercept)	0.314	0.314
	(0.386)	(0.384)
D	4.164	4.164
	(0.513)	(0.514)
Num.Obs.	100	100
R2	0.309	0.309
R2 Adj.	0.302	0.302

Export regression output in tables:

`fixest` has its own output-formatting function `etable`

```
library(fixest)
fix <- list(feols(Y ~ D, sample, vcov="iid"),
            feols(Y ~ D, sample, vcov="HC1"))
fix %>% etable()
```

```
##                          model 1           model 2
## Dependent Var.:                Y                 Y
##
## (Intercept)       0.3138 (0.3446)   0.3138 (0.3856)
## D               4.164*** (0.6292) 4.164*** (0.5129)
## _______________ _________________ _________________
## S.E. type                     IID Heteroskeda.-rob.
## Observations                  100               100
## R2                        0.30886           0.30886
## Adj. R2                   0.30181           0.30181
```

Export regression output in tables:

In Stata:

- `outreg2` is the most intuitive

Export regression output in tables:

In Stata:

- `outreg2` is the most intuitive
- `estout` is more flexible for exporting saved estimates