2. Statistical Tests: How Precise is Our Knowledge?

- So far, we discussed how to obtain *estimates* of a parameter
  - But note: We estimate $\hat{\beta}$ in a sample (a subset of the population) and it is to some extent random which observations are part of the sample
  - Therefore $\hat{\beta}$ will in general differ from ("true") population coefficient β
- An important part of empirical work is to *test* whether the true parameter differs from a hypothesized quantity
- To do a *statistical test* we need to know something about the distribution of $\hat{\beta}$
- Useful thought experiment: think of the variance in the obtained estimates $\hat{\beta}$ when drawing different samples from a population
  - When the variance of the obtained estimated $\hat{\beta}$ is small
    → you can be rather certain that $\hat{\beta}$ is close to the true β
  - Otherwise you may not learn so much from $\hat{\beta}$ about the true association and have to be more careful in your interpretation

Your Task

Simulated data set

Create a new notebook in which you generate a large simulated data set:

- Specify the number of observations in the large data set:

```
n=100000
```

- Create DataFrame with n rows and columns x and y:

```
df=pd.DataFrame(index=range(n), columns=['x','y'])
```

- Set x to a vector of n normally distributed random variables:

```
df['x']=np.random.normal(100,15,n)
```

- Set y according to the following formula and add some noise:

```
df['y']=200+2*df['x'] + np.random.normal(0,100,n)
```

Now consider the following:

- What is the “true” population coefficient of a regression of y on x ?
- Add a regression of y on x and compare your estimate to the true coefficient

Your Task — Simulated data set

- Suppose now you are a researcher interested in studying the relationship between y and x
- But instead of the large data set before you only have access to a smaller sample of 100 observations drawn from the large sample
- To simulate this, we can use the pandas sample method

  df.sample(n=100) draws a random sample of n=100 rows from Dataframe df
- Add a new cell to the notebook using this function to run the regression on a sample with 100 observations
  - mydata = df.sample(n=100)
  - pf.etable(pf.feols('y~x', data= mydata))
- Run this last cell several times and write down (& compare) the estimated coefficients of x
- Save the notebook as SimulateSample

3.1 Testing Hypotheses about a Parameter

Consider again the bivariate case where

$$\hat{\beta}_1 = \frac{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})\,(Y_i - \bar{Y})}{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

This can be rewritten (using that $Y_i = \beta_0 + \beta_1 X_i + e_i$) to become

$$\hat{\beta}_1 = \beta_1 + \frac{\sum_{i=1}^{N}(X_i - \bar{X})\,e_i}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

→ Estimate is sum of population value and a function of the disturbances e_i

When disturbances e_i are independent & their variance is σ^2 the variance of $\hat{\beta}$ is

$$V\left[\hat{\beta} \middle| X_1, X_2, \dots, X_N\right] = \frac{\sigma^2}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

→ The estimated $\hat{\beta}$ fluctuate around the true β with standard deviation

$$\frac{\sigma}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2}}$$

The Standard Error of a Regression Coefficient

- If we knew this standard deviation of $\hat{\beta}$ we would know how accurate our knowledge about the true relationship is
- The good news: We can estimate this standard deviation from our sample even when we have only one!
- The standard error $se(\hat{\beta})$ is

  … an estimate of the standard deviation of the regression coefficients

  … we would obtain from different samples drawn from the population

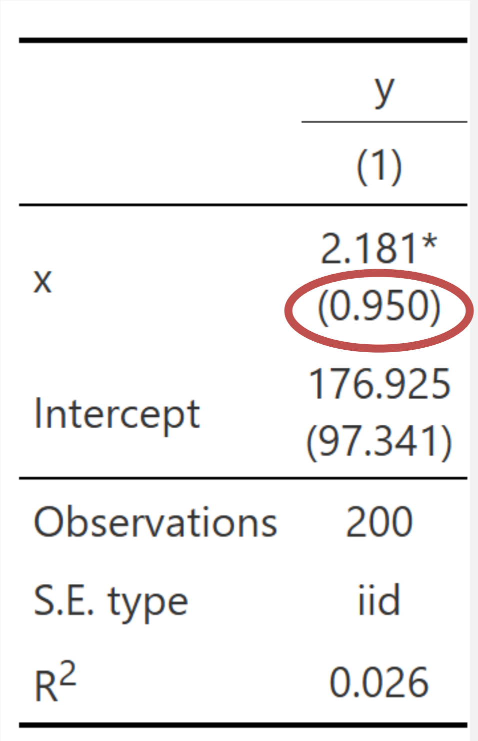
We can estimate the standard error either

- Using an analytical expression based on properties of the OLS estimator (estimate standard deviation of disturbances from regression residuals)
  - For the derivation see Wooldridge, Chapter 2
  - This is what econometrics software such as pyfixest reports
- Or only from the data through so-called bootstrapping
  - Useful to get an intuitive understanding of the standard error

Python

Estimating the Standard Error

- If you regress pf.etable(pf.feols('y~x', data=df_sample))
- Pyfixest will display the standard errors of the coefficients in brackets below the coefficients



	y
	(1)
x	2.181* (0.950)
Intercept	176.925 (97.341)
Observations	200
S.E. type	iid
R^2	0.026

Use Bootstaping to estimate the standard error

- The idea is simple:
  Mimic randomly sampling from a large population
- You have one sample with *n* observations
- Repeat *k* times: that is for *i = 1* to *k*
  - Randomly draw *n* observations from the sample *with replacement* (!)
    That is, an observation from the original sample can be several times in the bootstrap sample, while others may not be in the sample
  - The bootstrap samples will reflect the same underlying population CEF but produce different estimates due to random sampling
  - Estimate β with a regression run on the bootstrap sample *i*
  - Store the estimate $\hat{\beta}_i$
- Then compute the standard deviation of these *k* estimates $\hat{\beta}_i$
  → you will have an estimate of the standard error

Python

Estimating the Standard Error: Bootstrapping

- You can also “manually” estimate the standard error by bootstrapping (you typically do not need to, but it may help you to understand the idea of a standard error)

- Draw a sample with replacement from your data with
  df.sample(n=100, replace=True)

- Access regression coefficient of variable x in pyfixest by calling reg.coef().x

- Run a loop each time drawing samples of the same size as the original data set and running a regression on this sample (here we draw 100 samples)

```
coefs = []
nobs = df.shape[0] ## This is the number of observations in df
for i in range(100):
        bootstrap_sample = df.sample(n= nobs, replace=True)
        reg=pf.feols('y~x', data= bootstrap_sample)
        coefs.append(reg.coef().x)
```

- And then compute the sample standard deviation of the coefficients:
  np.std(coeff, ddof=1)

A key question we often ask is: Is there really an association between

- the outcome variable Y and
- a predictor X?

The problem is:

- Due to the specific sample at hand
  - we may just by chance observe a sizeable regression coefficient …
  - even when in fact there is no association

But:

- We can use the standard error to say something about how likely this is!

Your Task **Simulated data set**

- Change the code in the SimulateSample notebook to generate *pure noise*

```
df['y']=200 + np.random.normal(0,100,n)
```

- Run the cell and note down your coefficient

Use standard error to obtain what is called a *test statistic* for a coefficient:

- Our *Null hypothesis* is $H_0\colon \beta = 0$
- The alternative hypothesis is $H_1\colon \beta \neq 0$

When, for instance, $\hat{\beta} \sim N\left(\beta, V\left(\hat{\beta}\right)\right)$ we have that

$$\frac{\hat{\beta} - \beta}{sd\left(\hat{\beta}\right)} \sim N(0,1)$$

- But: As we do not know $sd\left(\hat{\beta}\right)$ precisely, we have to use its estimate:

  → The *standard error* $se\left(\hat{\beta}\right)$

- As this estimate is (somewhat) noisy in itself

  … the distribution of $\frac{\widehat{\beta}-\beta}{sd\left(\widehat{\beta}\right)}$

  … is not identical to the distribution of $\frac{\widehat{\beta}-\beta}{se\left(\widehat{\beta}\right)}$

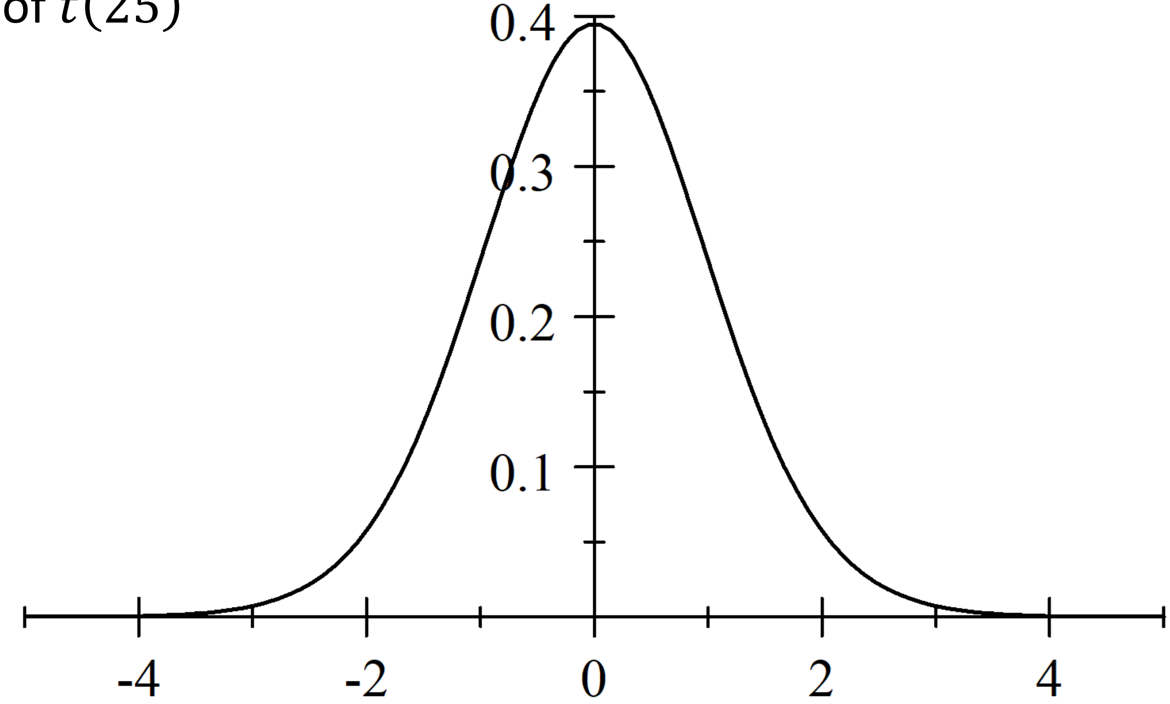
- But it is pretty close!

One can show: (proceed analogously in the multivariate case)

- $\frac{\widehat{\beta}-\beta}{se(\widehat{\beta})}$ follows a Student‘s t-distribution with n-2 degrees of freedom:

$$\frac{\hat{\beta}-\beta}{se(\hat{\beta})} \sim t(N-2)$$

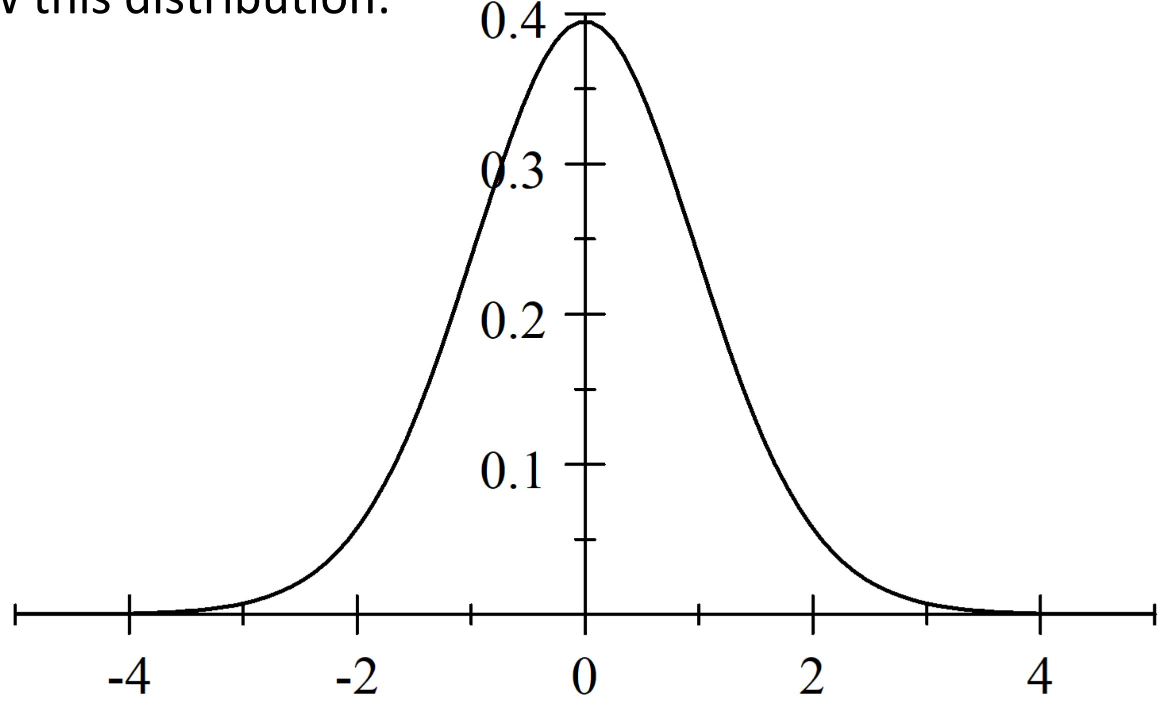
- Note: The t-distribution is very close to the standard normal distribution when the sample is not too small
- Example: Density of $t(25)$



Hence, when $\beta_j = 0$ then the so-called t-statistic

$$t_{\widehat{\beta}} = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$$

will follow this distribution:



That is: Very high and very low values of $t_{\widehat{\beta}}$ are very unlikely when $\beta_j = 0$!

Hence, to test the Null hypothesis that $\beta_j = 0$ we can look at the t−value

$$t_{\hat{\beta}} = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$$

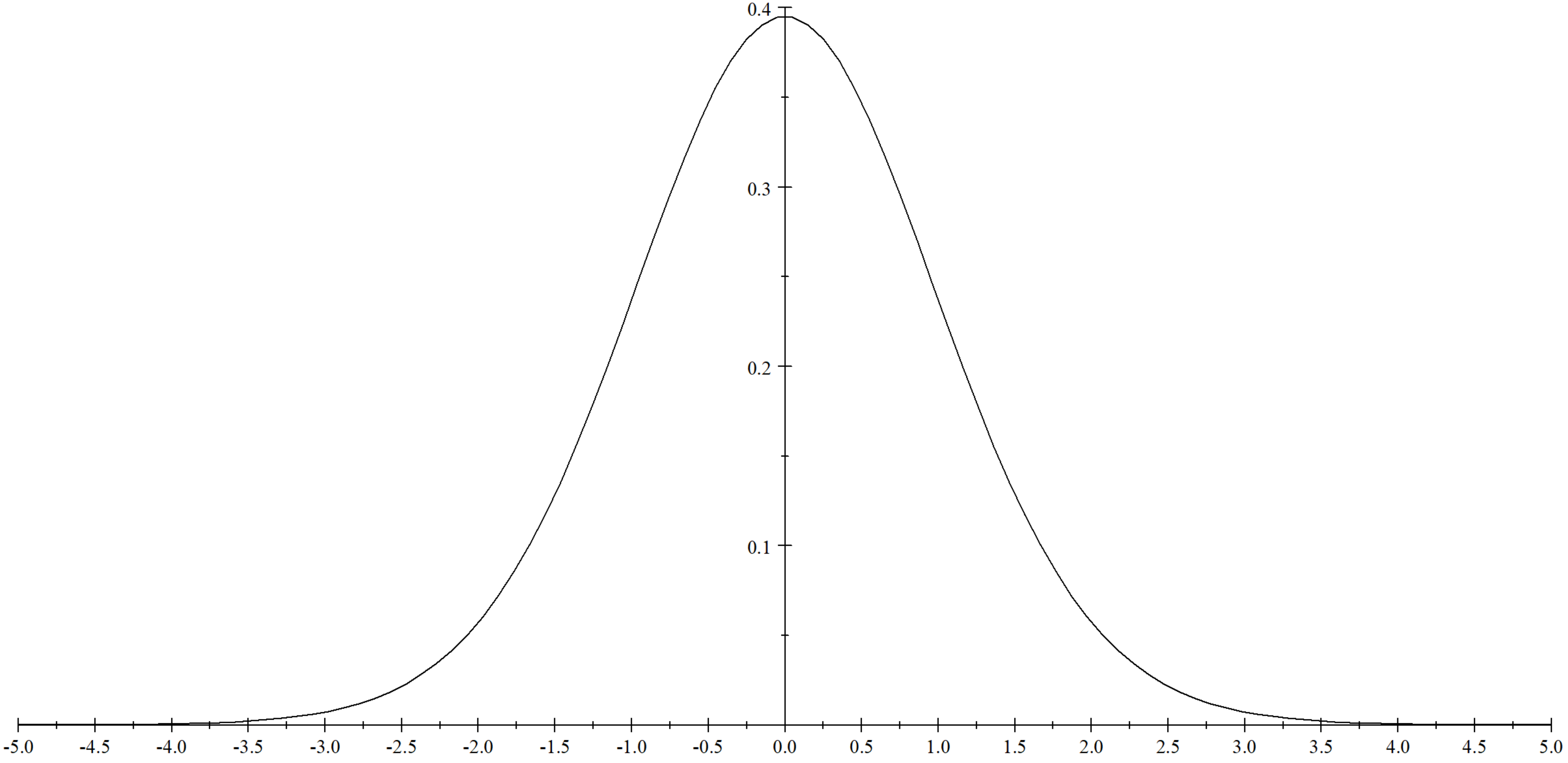
- When we consider a two-sided test,
  - we will reject H_0 whenever $t_{\hat{\beta}}$ is too large or too small
  - then it is unlikely that we would obtain an estimate $\hat{\beta}$ if the true β were 0
- Significance level α: likelihood that H_0 is rejected when it is in fact true
- Hence, we will reject H_0 at a significance level α if

$$\left|t_{\hat{\beta}}\right| > t_{\frac{\alpha}{2}}$$

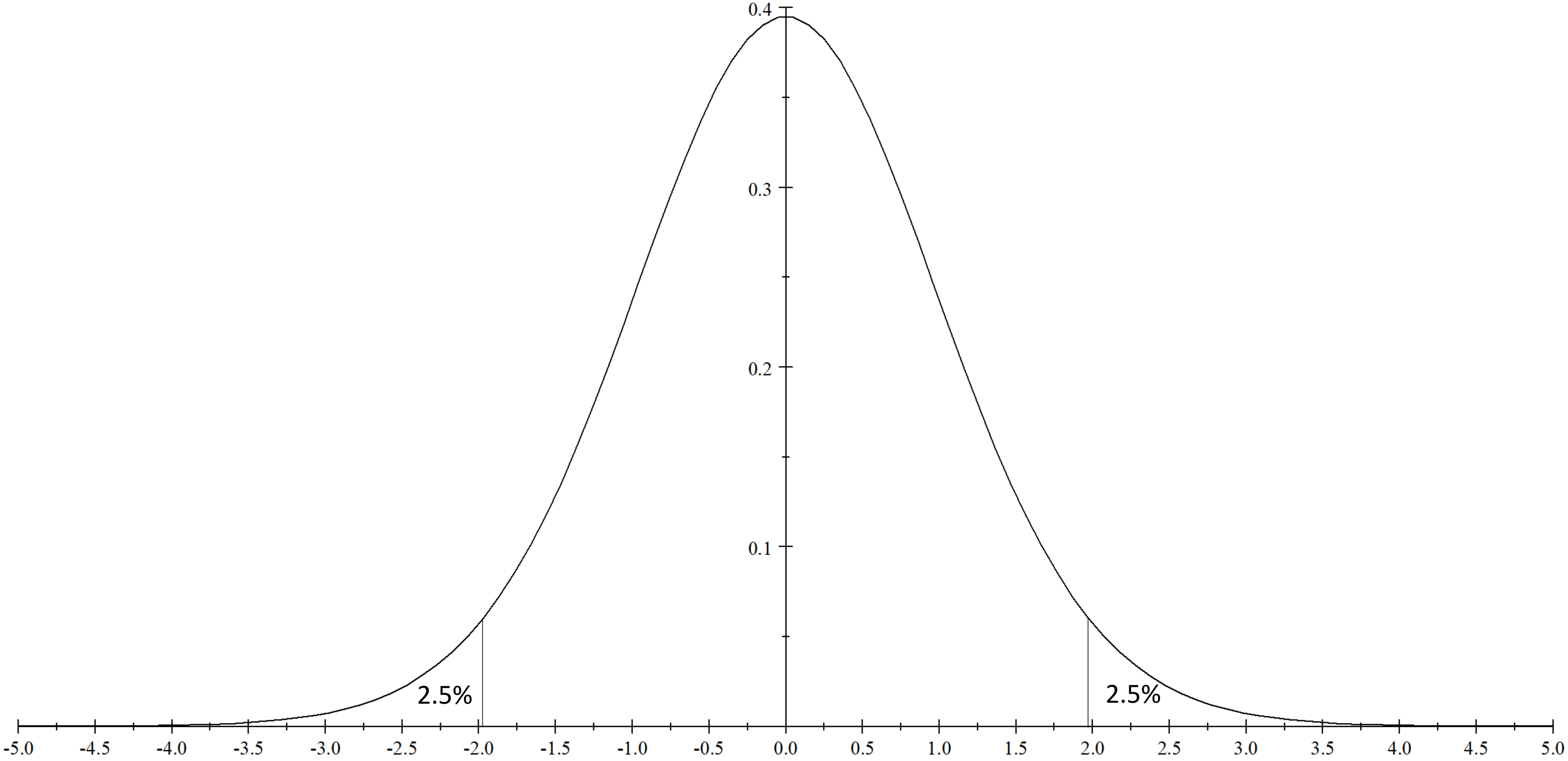
where $t_{\frac{\alpha}{2}}$ is the respective quantile of the t-distribution

Example: When $\alpha = 0.05$ we compare $\left|t_{\hat{\beta}}\right|$ with $t_{\frac{\alpha}{2}} = 1.962$ for $n - k - 1 = 1000$

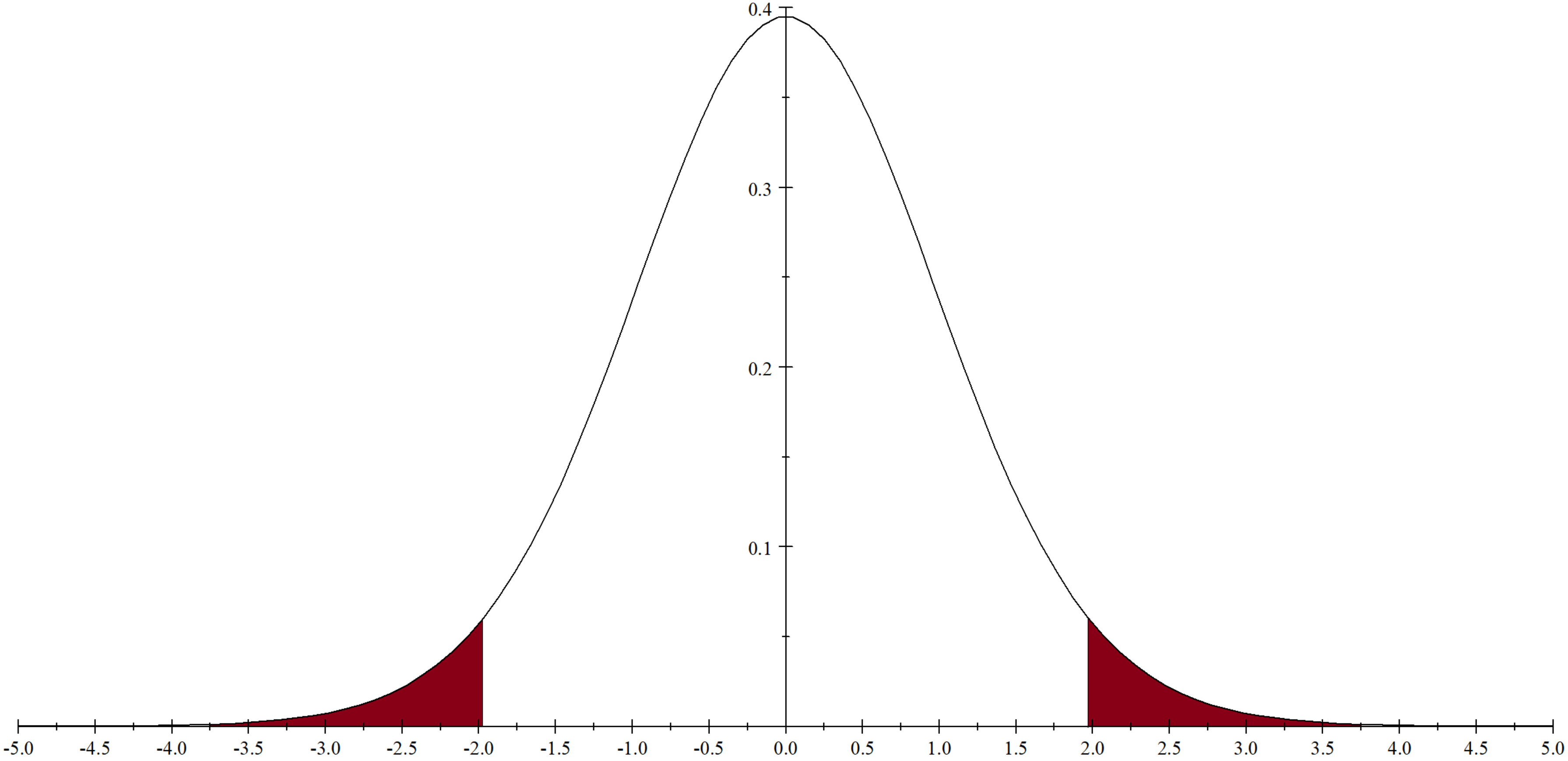
Consider the t-distribution under the Null hypothesis (if the true $\beta = 0$)



The likelihood that we will observe a t-value above 1.96 or below -1.96 is 5%:



We reject the Null hypothesis at a significance level of 5% if we observe a t-value above 1.96 or below -1.96



Your Task

Simulated data set

- Open again the notebook LPPanalysis.ipynb
- Look at the etable display your regression of job satisfaction on the mmagespr dummy
- What is the standard error?
- Please compute the t-value
- Can we reject the Null hypothesis that there is no association between job satisfaction and feedback interviews at the 5% level?
- Look at more detailed statistics with pf.summary()

Standard errors reported in regression tables yield the (estimated) standard deviation of the estimated $\hat{\beta}$

- Intuition: This gives us the following
  - If I would draw different samples (of the given size) I would obtain different estimators $\hat{\beta}$
  - What is the standard deviation of these $\hat{\beta}$?
  - When this is small: we are close to the true β
  - When this is large: there is much noise and therefore it is likely that the estimated $\hat{\beta}$ is further away from the true β

- The standard errors are used
  - to construct the *t-value*
    (reported by pyfixest when using pf.summary())
  - And from that compute the *p-values*

What a p-values tell us:

- What is the probability of obtaining an estimate that is at least as "extreme" (distant from 0) as the value of $\hat{\beta}_j$ I have estimated when the true value of were in fact $\beta = 0$
- When this probability is smaller (computed using the estimated standard errors) we can be more certain that the true β is not zero
- For instance, when $p < 0.05$ for a certain coefficient, we say that the coefficient is *statistically significant at the* 5% *level*

- We mark this in a typical regression table in a paper with
  - * if $p < 0.10$
  - ** if $p < 0.05$ and
  - *** if $p < 0.01$

Your Task

Simulated data set

- Consider again the SimulateSample notebook where you set

  ```
  df['y']=200 + np.random.normal(0,100,n)
  ```

- If you run the cell 100 times (i.e. draw a 100 samples): How often should you get a regression result where the coefficient is significant at the 10% level?

3.2 Confidence Intervals

- It is often useful to construct so called **confidence intervals** (also called confidence bands) to illustrate the uncertainty we still have when estimating a parameter
- Think of the following task:
  - Suppose you draw many samples
  - What is the range of possible values of $\hat{\beta}$ so that we can be sufficiently sure that the true β is in that range?
- For instance: A 95% confidence interval specifies the smallest range of values such that
  - when we would draw different samples
  - in 95% of the samples the interval will include the true value β

- When constructing a 95% confidence band, we want to find k such that

$$Pr\left(\hat{\beta}_j - k \leq \beta \leq \hat{\beta}_j + k\right) = 0.95$$

or equivalently

$$Pr\left(-k \leq \hat{\beta}_j - \beta \leq k\right) = 0.95$$

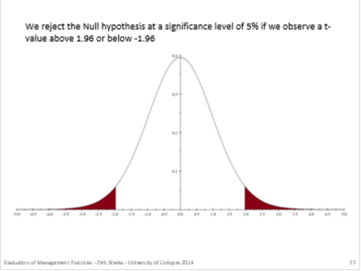
Recall

$$\frac{\hat{\beta}_j - \beta_j}{se\left(\hat{\beta}_j\right)} \sim t(N - k - 1)$$

which is equivalent to

$$\hat{\beta}_j - \beta_j \sim se\left(\hat{\beta}_j\right) \cdot t(N - k - 1)$$

Consider $Pr\left(-k \leq \hat{\beta}_j - \beta \leq k\right) = 0.95$



We reject the Null hypothesis at a significance level of 5% if we observe a t-value above 1.96 or below -1.96

Evaluation of Management Practices - Dirk Sliwka - University of Cologne 2024 17

- Recall the percentiles $t_{0.025} \approx -1.96$ and $t_{0.975} \approx 1.96$
- Hence,

$$Pr\left(-1.96 \cdot se\left(\hat{\beta}_j\right) \leq \hat{\beta}_j - \beta \leq 1.96 \cdot se\left(\hat{\beta}_j\right)\right) = 0.95$$

- The confindence interval is thus $\left[\hat{\beta}_j - 1.96 \cdot se\left(\hat{\beta}_j\right), \hat{\beta}_j + 1.96 \cdot se\left(\hat{\beta}_j\right)\right]$
- The respective 90% interval is $\left[\hat{\beta}_j - 1.65 \cdot se\left(\hat{\beta}_j\right), \hat{\beta}_j + 1.65 \cdot se\left(\hat{\beta}_j\right)\right]$

Importantly: This means

- When we draw different samples, in 95% of the samples the 95% confidence interval will include the true value β
- This does not mean that we can be sure that the true value is in the interval
- But: it will be in the interval in about 19 of 20 samples drawn
- A 95% confidence interval also represents the set of values that are not statistically significantly different from the point estimate at the 5% level

„Absence of Evidence is not Evidence of Absence“

- What is the correct interpretation of insignificant coefficients?
- Sometimes you see people claiming: „As we did not find a statistically significant coefficient we have shown that there is no effect“
- This is wrong (yet you see it surprisingly often)! → Why?

In fact:

- One can never state to have shown that a variable has no effect
- But you can sometimes provide evidence that it is very unlikely that the true effect is large
  - Here confidence bands are very useful
  - If even the upper limit of a confidence band is small, then it is very unlikely that the true association is large

Study

RCT in Retailing: A Bonus for raising the average receipt

Manthei/Sliwka/Vogelsang (Management Science, 2021):

- Two experiments in one region of a retail chain on a bonus for raising the average receipt
- Experiment 1: Bonus for district managers Nov 2015 – Jan 2016
  - 25 district managers (152 stores) randomly assigned to the treatment (Norm. Bonus)
  - 24 district managers (148 stores) in the control group
- Experiment 2: Bonus for *store managers* Nov 2016 – Jan 2017
  - 99 store managers assigned to the treatment Norm. Bonus
  - 95 store managers assigned to the treatment Simple Bonus
  - 95 store managers assigned to the control group
- Average receipt as KPI for bonus assignment in both experiments, experiment on a lower hierarchical level & includes a simplified bonus formula

Study

RCT in Retailing: A Bonus for raising the average receipt

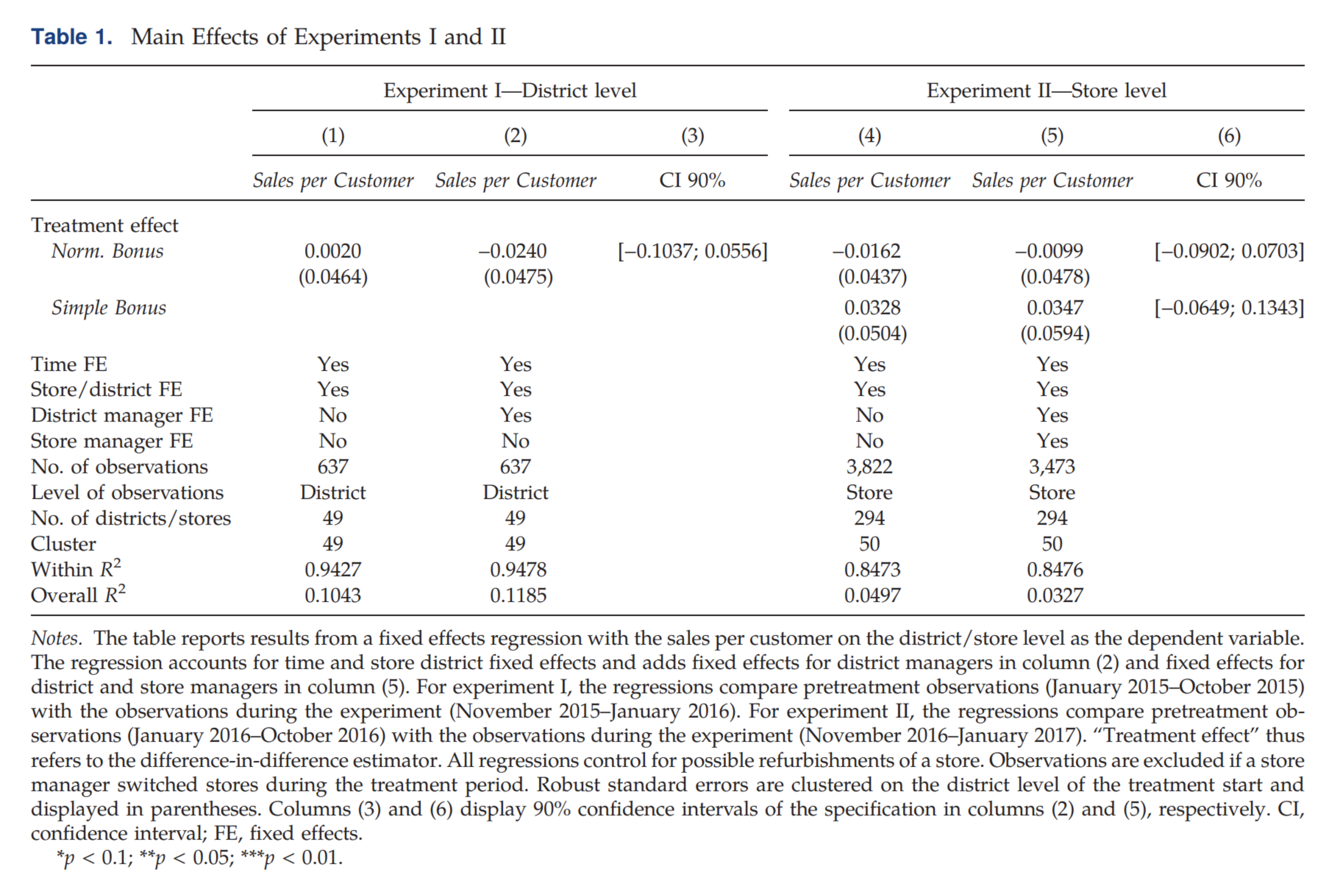


Table 1. Main Effects of Experiments I and II

	Experiment I—District level			Experiment II—Store level		
	(1)	(2)	(3)	(4)	(5)	(6)
	Sales per Customer	*Sales per Customer*	CI 90%	*Sales per Customer*	*Sales per Customer*	CI 90%
Treatment effect						
Norm. Bonus	0.0020	–0.0240	[–0.1037; 0.0556]	–0.0162	–0.0099	[–0.0902; 0.0703]
	(0.0464)	(0.0475)		(0.0437)	(0.0478)	
Simple Bonus				0.0328	0.0347	[–0.0649; 0.1343]
				(0.0504)	(0.0594)	
Time FE	Yes	Yes		Yes	Yes	
Store/district FE	Yes	Yes		Yes	Yes	
District manager FE	No	Yes		No	Yes	
Store manager FE	No	No		No	Yes	
No. of observations	637	637		3,822	3,473	
Level of observations	District	District		Store	Store	
No. of districts/stores	49	49		294	294	
Cluster	49	49		50	50	
Within R^2	0.9427	0.9478		0.8473	0.8476	
Overall R^2	0.1043	0.1185		0.0497	0.0327	

Notes. The table reports results from a fixed effects regression with the sales per customer on the district/store level as the dependent variable. The regression accounts for time and store district fixed effects and adds fixed effects for district managers in column (2) and fixed effects for district and store managers in column (5). For experiment I, the regressions compare pretreatment observations (January 2015–October 2015) with the observations during the experiment (November 2015–January 2016). For experiment II, the regressions compare pretreatment observations (January 2016–October 2016) with the observations during the experiment (November 2016–January 2017). "Treatment effect" thus refers to the difference-in-difference estimator. All regressions control for possible refurbishments of a store. Observations are excluded if a store manager switched stores during the treatment period. Robust standard errors are clustered on the district level of the treatment start and displayed in parentheses. Columns (3) and (6) display 90% confidence intervals of the specification in columns (2) and (5), respectively. CI, confidence interval; FE, fixed effects.

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Your Task — Management Practices

- Open your ManagementPractices notebook
- Go to the regression table where you compared the association between the management score and roce in GB and China with an interaction term
- Add two more regressions, where you directly regress the management score and roce separately only in the UK and only in the Chinese data (Recall: You can access a subset of the DataFrame with df[x=..]
- Interpret your findings:
  - Is the “effect” in GB significant?
  - Is the interaction term significant?
  - Is the “effect” in China significant?
  - How does this fit together?
- Also inspect the confidence bands in the regression where you only consider the Chinese data (pf.summary() displays confidence bands)

3.3 Robust Standard Errors

- We made the assumption that
  - the disturbances are normally distributed
  - they have the same variance for all observations
- In general we cannot be sure that assumption that $e \sim N\left(0, \sigma^2 I_N\right)$ will hold
  - Implies that residuals are normally distributed and
  - that variance of residuals is constant → so-called *Homoscedasticity*
- But it always holds (in the bivariate case) that

$$\hat{\beta}_1 = \beta_1 + \frac{\sum_{i=1}^{N}(X_i - \bar{X})e_i}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

That is, $\hat{\beta}$ is the sum of the true β plus a function of the disturbances

- One can now show (see Angrist/Pischke, p. 45) that $\hat{\beta}$ is asymptotically normally distributed
  - with probability limit β (i.e. sample size grows → $\hat{\beta}$ comes closer to β)
  - and a covariance matrix that can be estimated from X_i and the residuals

- So called "Robust" standard errors follow from this covariance matrix

- Are called *robust* because
  - they are derived without assuming that the variance of the residuals is independent of X_i (i.e. they allow for heteroscedasticity)
  - and in large enough samples they provide accurate hypothesis testing without further distributional assumptions

Interdependent Observations & Clustered Standard Errors

- Standard errors are estimated under the assumption that
  - data are *independent observations*
  - or in other words each observation is a random draw from the population
- Very often this is not the case, for instance when
  - We observe several employees that come from the same firm
  - Or we observe the same employee at different dates
- The residuals will be correlated when the observations come from the same person or colleagues in the same firm
- Standard errors that are estimated assuming independence of employee observations are then biased
  - They are typically too small
  - p-Values are then smaller than they should be
  - We run into the danger of rejecting the null hypothesis too often

Two possible solutions:

- Use only firm level observations
  - For instance, build a data set that includes one observation per firm
  - (in the above example we would have 1.000 observations)
  - And use the average job satisfaction in the firm as dependent variable

- Use a different method for estimating standard errors
  - Account for the interdependence of observations within groups or clusters
  - So called Eicker–Huber–White standard errors
  - Note: Number of clusters should not be too small (see A/P, chapter 8) otherwise one may use bootstrap methods

Python

Cluster Robust Standard Errors

- In pyfixest you can estimate robust standard errors
- To account for possible heteroscedasticity add

```
fit = pf.feols("Y ~ X1 + X2", data, vcov="hetero")
```

- To account for interdependence within clusters

```
fit2 = pf.feols("Y ~ X1 + X2", data, vcov={"CRV1": "groupvar"})
```

  - where groupvar is the variable that determines the cluster
  - i.e. where we suspect that observations for which groupvar has the same value are not independent

Your Task

Clustered Standard Errors

- Open your ManagementPractices notebook
- Run the notebook
- Inspect the DataFrame
- Note:
  - For each firm (account_id) there are observations from different years
  - These observations will not be independent and thus standard errors will be biased
- Copy your regression commands to have them three times in the cell
- In regression 2 estimate heteroscedasticity robust standard errors
- In regression 3 estimate cluster robust standard errors
- Compare the standard errors in the three regressions

Summary:

- We nearly always estimate the regression parameters from a sample
- We thus have to be aware that the coefficients we have obtained will in general not be equal to the „true“ population coefficients
- But: We can assess how noisy our estimates are

  → We do this by estimating the *standard error* of our coefficient

- Based on standard errors we can construct statistical tests:

  → How likely is it that we see an estimate like this when in fact there is no connection?

- We can also construct confidence intervals

  → For instance: Give me the range of values where the true value is with 95% certainty

- Be have to be careful when sample includes observations that are not independently drawn → Here clustering standard errors is important

Appendix

Consider

$$\hat{\beta}_1 = \frac{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})\,(Y_i - \bar{Y})}{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

We can rewrite this as

$$\frac{\sum_{i=1}^{N}(X_i - \bar{X})\,(\beta_0 + \beta_1 X_i + e_i - (\beta_0 + \beta_1 \bar{X} + \bar{e}))}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

$$= \frac{\sum_{i=1}^{N}(X_i - \bar{X})\,\beta_1 (X_i - \bar{X})}{\sum_{i=1}^{N}(X_i - \bar{X})^2} + \frac{\sum_{i=1}^{N}(X_i - \bar{X})\,e_i}{\sum_{i=1}^{N}(X_i - \bar{X})^2} - \frac{\sum_{i=1}^{N}(X_i - \bar{X})}{\sum_{i=1}^{N}(X_i - \bar{X})^2}\,\bar{e}$$

and as $\sum_{i=1}^{N}(X_i - \bar{X}) = 0$ we obtain

$$\hat{\beta}_1 = \beta_1 + \frac{\sum_{i=1}^{N}(X_i - \bar{X})\,e_i}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

→ The estimate is the sum of the population value and a function of the residuals

Take

$$\hat{\beta}_1 = \beta_1 + \frac{\sum_{i=1}^{N}(X_i - \bar{X})\, e_i}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

When the e are independently distributed and have the same variance σ^2

$$V\,[\hat{\beta}_1 | X_1, X_2, \dots, X_N] = \frac{1}{\left(\sum_{i=1}^{N}(X_i - \bar{X})^2\right)^2} V\left[\sum_{i=1}^{N}(X_i - \bar{X})\, e_i\right]$$

$$= \frac{\sum_{i=1}^{N}(X_i - \bar{X})^2 V[e_i]}{\left(\sum_{i=1}^{N}(X_i - \bar{X})^2\right)^2} = \frac{\sigma^2}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

The Multivariate Case

- One can show analogously when β is a vector that

$$\hat{\beta} = \beta + \left[\sum_{i=1}^{N} X_i\, X_i'\right]^{-1} \sum_{i=1}^{N} X_i\, e_i$$

- In matrix notation

$$\hat{\beta} = \beta + (X'X)^{-1}X'e$$

- When the residuals are normally distributed & have the same variance

$$e \sim N\left(0, \sigma^2 I_N\right)$$

where I_N is the $N \times N$ identity matrix

- One can show that the vector of parameter estimates

$$\hat{\beta} \sim N\left(\beta, \sigma^2 (X'X)^{-1}\right)$$

- And (if there are k parameters to estimate, for each $j = 1, \dots, k$)

$$\frac{\hat{\beta}_j - \beta_j}{se\left(\hat{\beta}_j\right)} \sim t(N - k - 1)$$