Multivariate Regressions

Michele Coscia

First Year Project #1

February 17th, 2022

Lecture Plan

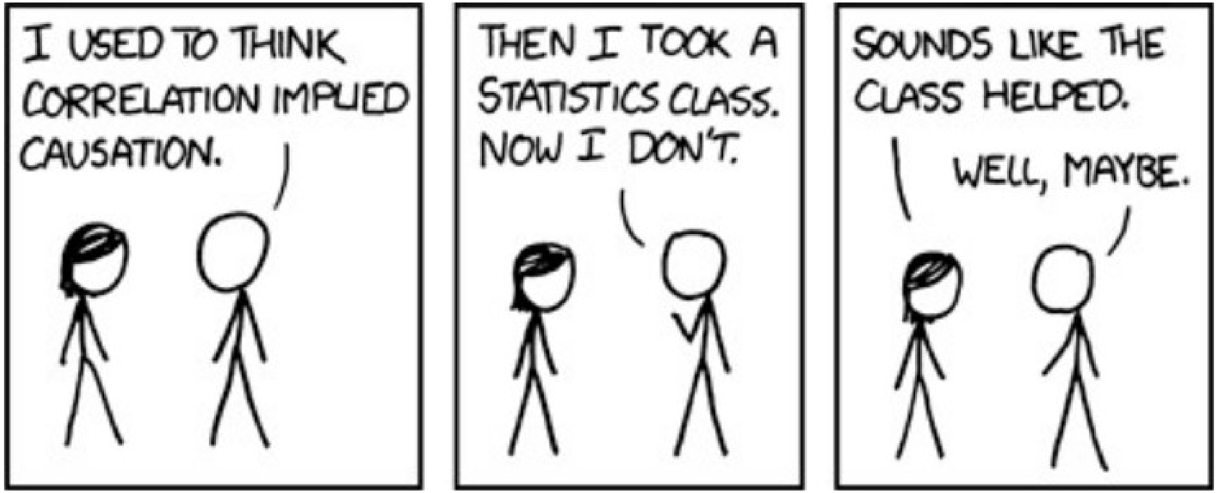
1) (February 8th) Intro
2) (February 10th) Geospatial Basics
3) (February 15th) Estimating Associations
4) **(Today) Multivariate Regression**
5) (February 22nd) Interventions
6) (February 24th) Project Run Through
7) (March 1st) Q&A – Open Supervision
8) (March 3rd) Q&A – Open Supervision

Outline

- Multivariate Regression Basics
- Fixed Effects
- Clustered Standard Errors

Multivariate Regression Basics

Always Remember


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Correlation != Causation

- Because X & Y could be both dependent on Z
- Pearson cannot handle this
- Multivariate regression can!
- You can control for Z
- → Investigate the effect of X keeping Z constant

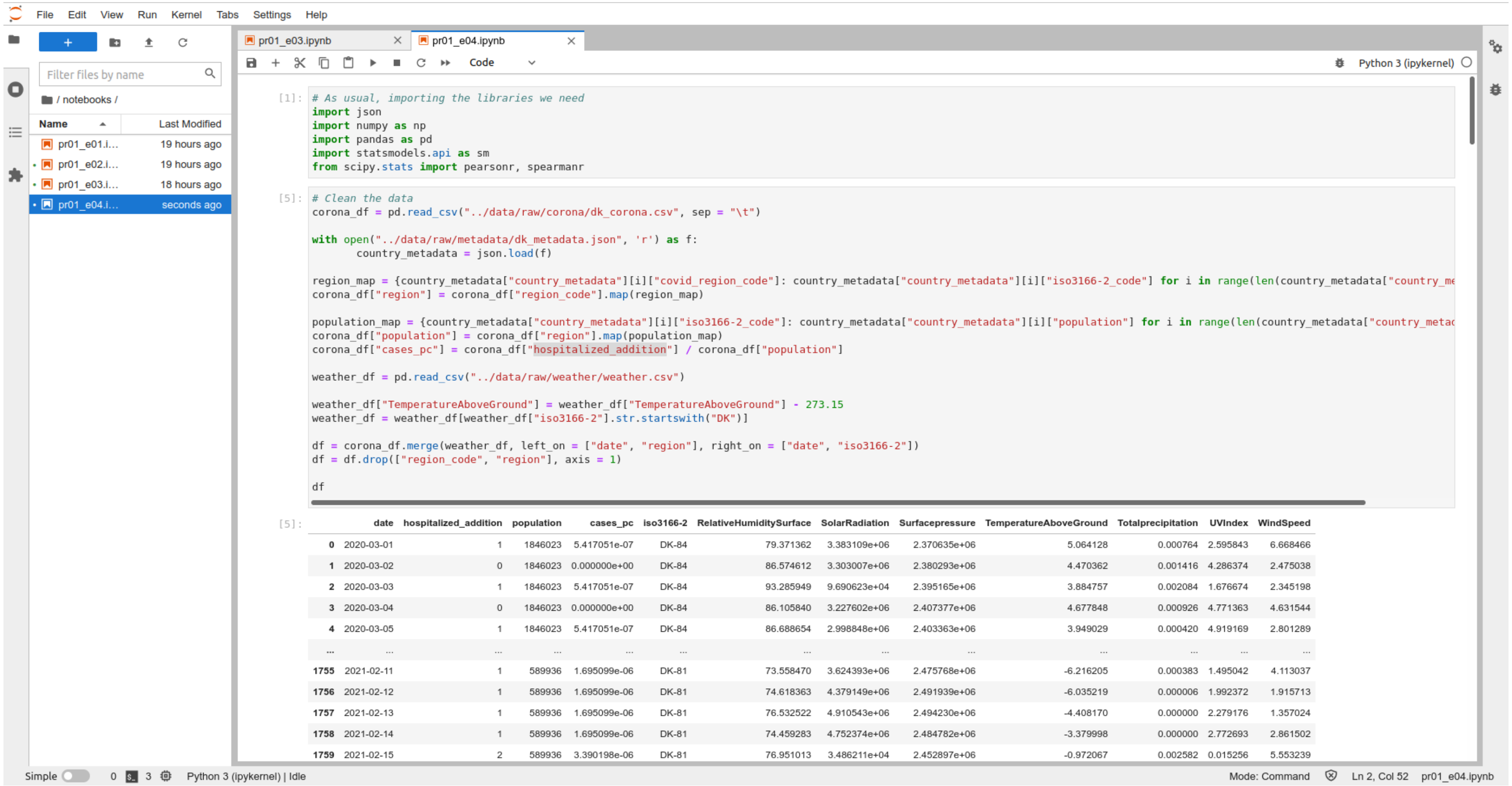
Multivariate regression is still not enough!

- Linear, just like Pearson
- Finding all Zs is hard
- Some controls are bad!
- It won’t tell you if X → Y or Y → X
- Causality is hard...

Linearity

- Running a regression comes with assumptions
- Non-linear data normally break these assumptions
- Hard way: check the assumption list and satisfy them
- For now: log-transform

Let’s Get Started!



```
[1]: # As usual, importing the libraries we need
     import json
     import numpy as np
     import pandas as pd
     import statsmodels.api as sm
     from scipy.stats import pearsonr, spearmanr
```

```
[5]: # Clean the data
     corona_df = pd.read_csv("../data/raw/corona/dk_corona.csv", sep = "\t")

     with open("../data/raw/metadata/dk_metadata.json", 'r') as f:
            country_metadata = json.load(f)

     region_map = {country_metadata["country_metadata"][i]["covid_region_code"]: country_metadata["country_metadata"][i]["iso3166-2_code"] for i in range(len(country_metadata["country_me
     corona_df["region"] = corona_df["region_code"].map(region_map)

     population_map = {country_metadata["country_metadata"][i]["iso3166-2_code"]: country_metadata["country_metadata"][i]["population"] for i in range(len(country_metadata["country_metac
     corona_df["population"] = corona_df["region"].map(population_map)
     corona_df["cases_pc"] = corona_df["hospitalized_addition"] / corona_df["population"]

     weather_df = pd.read_csv("../data/raw/weather/weather.csv")

     weather_df["TemperatureAboveGround"] = weather_df["TemperatureAboveGround"] - 273.15
     weather_df = weather_df[weather_df["iso3166-2"].str.startswith("DK")]

     df = corona_df.merge(weather_df, left_on = ["date", "region"], right_on = ["date", "iso3166-2"])
     df = df.drop(["region_code", "region"], axis = 1)

     df
```

[5]:

	date	hospitalized_addition	population	cases_pc	iso3166-2	RelativeHumiditySurface	SolarRadiation	Surfacepressure	TemperatureAboveGround	Totalprecipitation	UVIndex	WindSpeed
0	2020-03-01	1	1846023	5.417051e-07	DK-84	79.371362	3.383109e+06	2.370635e+06	5.064128	0.000764	2.595843	6.668466
1	2020-03-02	0	1846023	0.000000e+00	DK-84	86.574612	3.303007e+06	2.380293e+06	4.470362	0.001416	4.286374	2.475038
2	2020-03-03	1	1846023	5.417051e-07	DK-84	93.285949	9.690623e+04	2.395165e+06	3.884757	0.002084	1.676674	2.345198
3	2020-03-04	0	1846023	0.000000e+00	DK-84	86.105840	3.227602e+06	2.407377e+06	4.677848	0.000926	4.771363	4.631544
4	2020-03-05	1	1846023	5.417051e-07	DK-84	86.688654	2.998848e+06	2.403363e+06	3.949029	0.000420	4.919169	2.801289
...	...	...	...	...	...	...	...	...	...	...	...	...
1755	2021-02-11	1	589936	1.695099e-06	DK-81	73.558470	3.624393e+06	2.475768e+06	-6.216205	0.000383	1.495042	4.113037
1756	2021-02-12	1	589936	1.695099e-06	DK-81	74.618363	4.379149e+06	2.491939e+06	-6.035219	0.000006	1.992372	1.915713
1757	2021-02-13	1	589936	1.695099e-06	DK-81	76.532522	4.910543e+06	2.494230e+06	-4.408170	0.000000	2.279176	1.357024
1758	2021-02-14	1	589936	1.695099e-06	DK-81	74.459283	4.752374e+06	2.484782e+06	-3.379998	0.000000	2.772693	2.861502
1759	2021-02-15	2	589936	3.390198e-06	DK-81	76.951013	3.486211e+04	2.452897e+06	-0.972067	0.002582	0.015256	5.553239

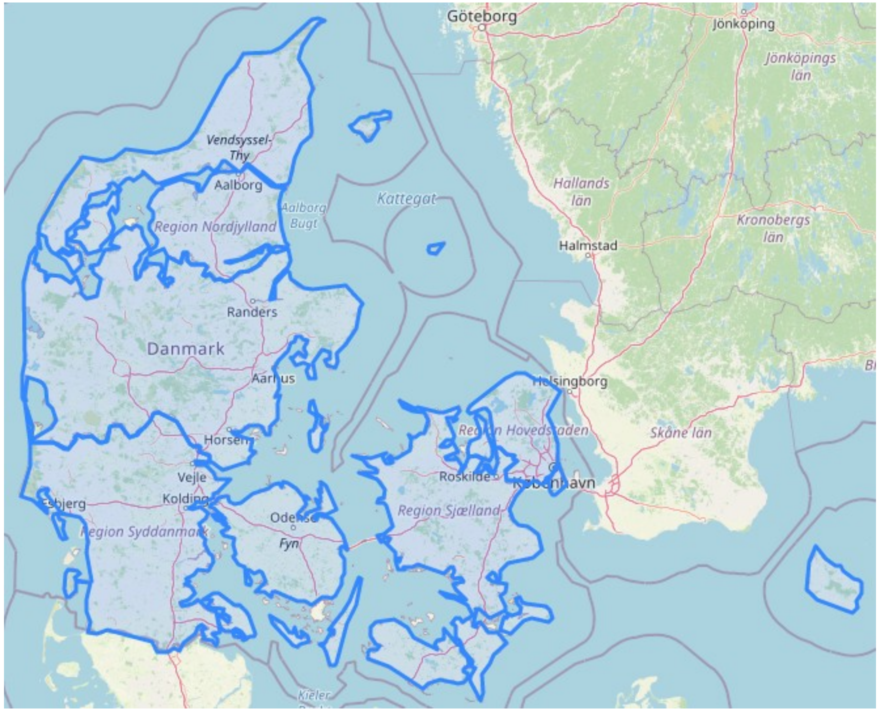
Fixed Effects

Problem

- Sometimes you **know** something affected your outcome
- You just don’t have any measure for it
- In our case: different local governments work differently

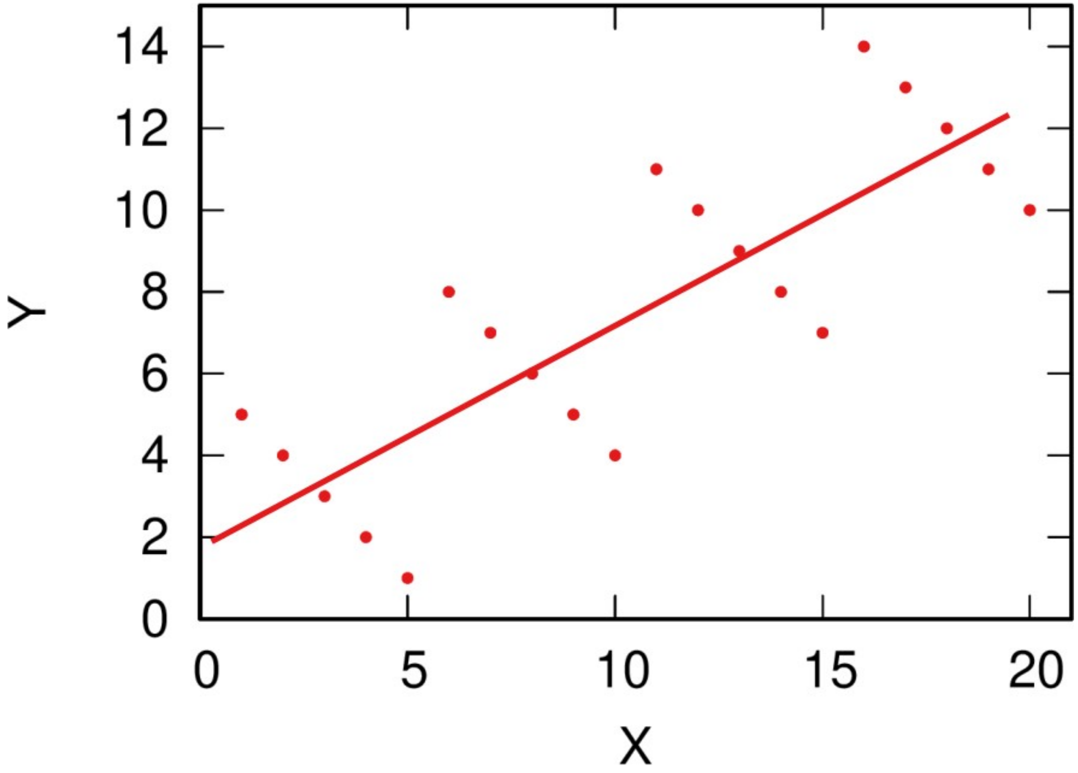
Fixed Effects

- You know your observations belong to specific groups
  - In our case, Danish regions
- The avg of each group is fixed
- Everything that group does differently from the other groups is captured here



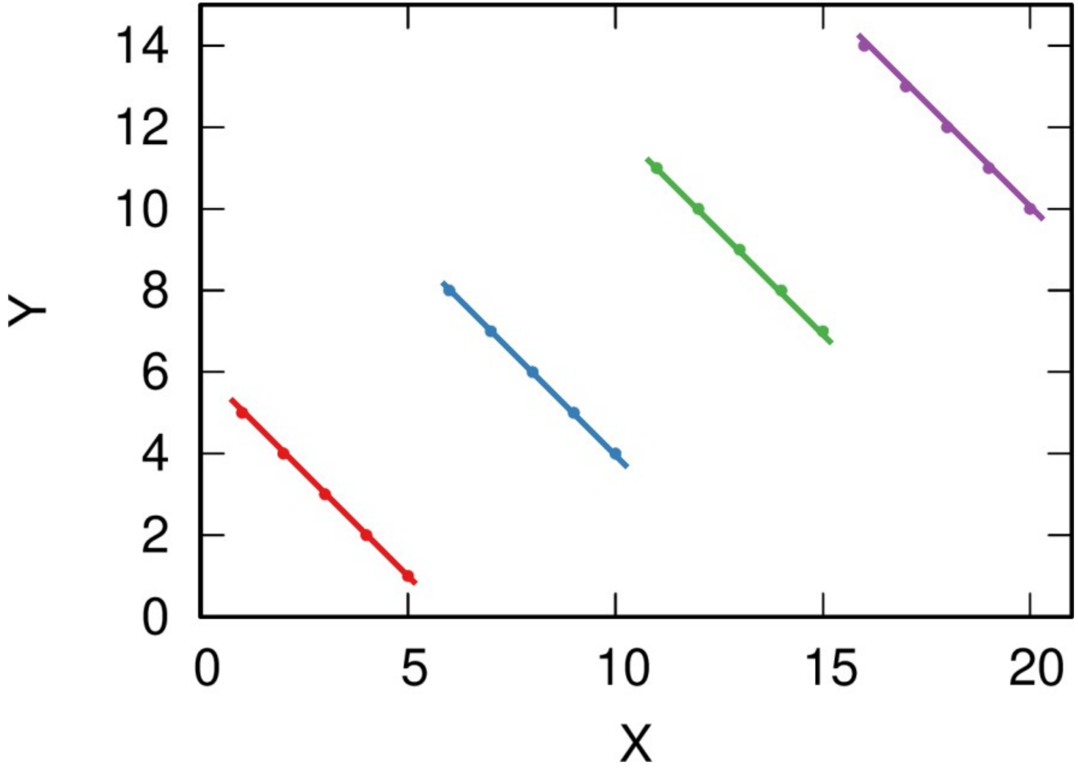
Why would we do this?

- Corr ~ 0.8!!
- Best fit
- Something Fishy...



Why would we do this?

- Groups!
- Related with X
- The true relationship is actually negative!



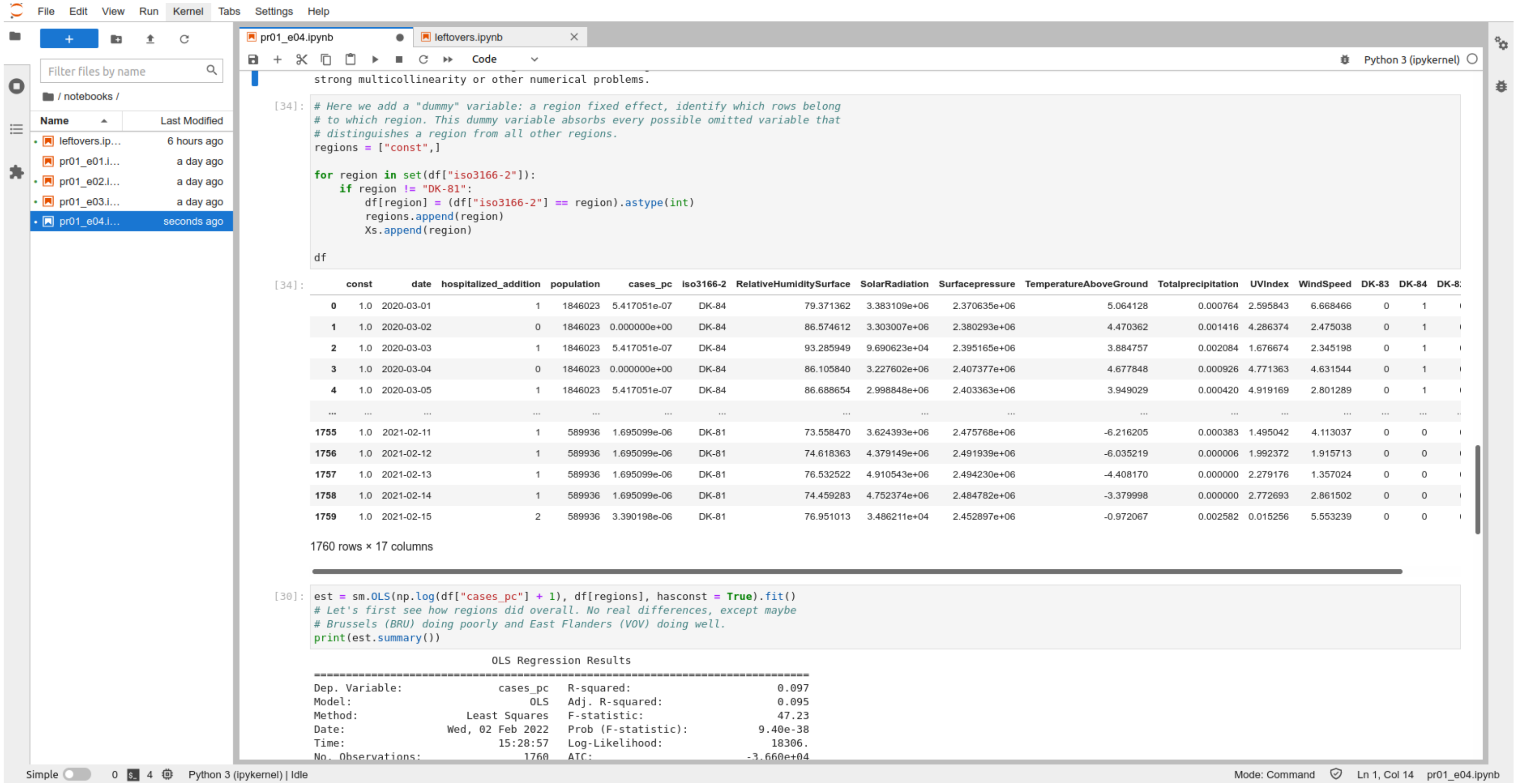
How to do it practically

- In R, it’s automatic
  - Just pass a categorical variable to your regression function
- In general, you can add a “dummy variable”
  - One variable per group
  - 1 if observation belongs to the group, 0 otherwise
  - You need to omit one group (the reference)

Interpretation

- Coefficient tells you the effect of being part of the group
  - Specifically: the difference between your group and the reference one
- If group membership is important for your question, you can interpret it
  - But careful, because you’re absorbing everything!
- Most often, it’s just a control → Ignore

Let’s Fix Some Effects!



Name	Last Modified
leftovers.ip…	6 hours ago
pr01_e01.i…	a day ago
pr01_e02.i…	a day ago
pr01_e03.i…	a day ago
pr01_e04.i…	seconds ago

```
strong multicollinearity or other numerical problems.
```

```
[34]: # Here we add a "dummy" variable: a region fixed effect, identify which rows belong
      # to which region. This dummy variable absorbs every possible omitted variable that
      # distinguishes a region from all other regions.
      regions = ["const",]

      for region in set(df["iso3166-2"]):
          if region != "DK-81":
              df[region] = (df["iso3166-2"] == region).astype(int)
              regions.append(region)
              Xs.append(region)

      df
```

[34]:

	const	date	hospitalized_addition	population	cases_pc	iso3166-2	RelativeHumiditySurface	SolarRadiation	Surfacepressure	TemperatureAboveGround	Totalprecipitation	UVIndex	WindSpeed	DK-83	DK-84	DK-8
0	1.0	2020-03-01	1	1846023	5.417051e-07	DK-84	79.371362	3.383109e+06	2.370635e+06	5.064128	0.000764	2.595843	6.668466	0	1	
1	1.0	2020-03-02	0	1846023	0.000000e+00	DK-84	86.574612	3.303007e+06	2.380293e+06	4.470362	0.001416	4.286374	2.475038	0	1	
2	1.0	2020-03-03	1	1846023	5.417051e-07	DK-84	93.285949	9.690623e+04	2.395165e+06	3.884757	0.002084	1.676674	2.345198	0	1	
3	1.0	2020-03-04	0	1846023	0.000000e+00	DK-84	86.105840	3.227602e+06	2.407377e+06	4.677848	0.000926	4.771363	4.631544	0	1	
4	1.0	2020-03-05	1	1846023	5.417051e-07	DK-84	86.688654	2.998848e+06	2.403363e+06	3.949029	0.000420	4.919169	2.801289	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1755	1.0	2021-02-11	1	589936	1.695099e-06	DK-81	73.558470	3.624393e+06	2.475768e+06	-6.216205	0.000383	1.495042	4.113037	0	0	
1756	1.0	2021-02-12	1	589936	1.695099e-06	DK-81	74.618363	4.379149e+06	2.491939e+06	-6.035219	0.000006	1.992372	1.915713	0	0	
1757	1.0	2021-02-13	1	589936	1.695099e-06	DK-81	76.532522	4.910543e+06	2.494230e+06	-4.408170	0.000000	2.279176	1.357024	0	0	
1758	1.0	2021-02-14	1	589936	1.695099e-06	DK-81	74.459283	4.752374e+06	2.484782e+06	-3.379998	0.000000	2.772693	2.861502	0	0	
1759	1.0	2021-02-15	2	589936	3.390198e-06	DK-81	76.951013	3.486211e+04	2.452897e+06	-0.972067	0.002582	0.015256	5.553239	0	0	

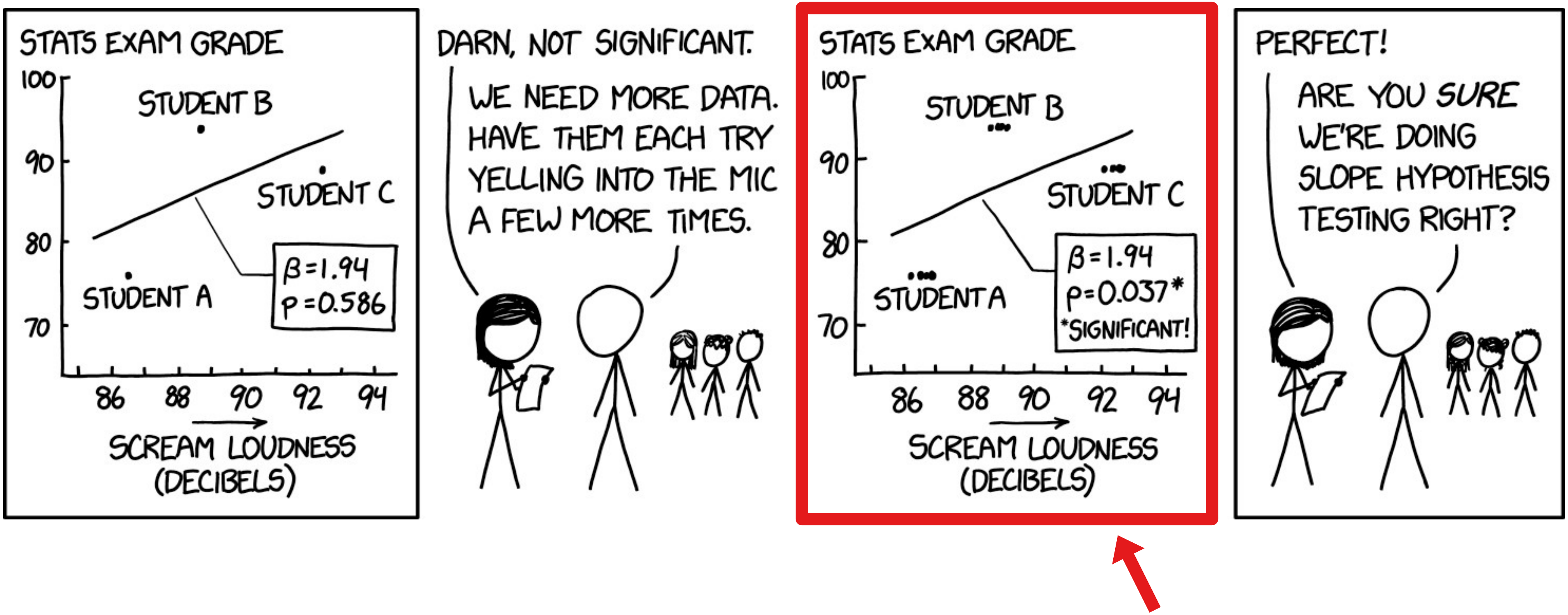
1760 rows × 17 columns

```
[30]: est = sm.OLS(np.log(df["cases_pc"] + 1), df[regions], hasconst = True).fit()
      # Let's first see how regions did overall. No real differences, except maybe
      # Brussels (BRU) doing poorly and East Flanders (VOV) doing well.
      print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               cases_pc   R-squared:                       0.097
Model:                            OLS   Adj. R-squared:                  0.095
Method:                 Least Squares   F-statistic:                     47.23
Date:                Wed, 02 Feb 2022   Prob (F-statistic):           9.40e-38
Time:                        15:28:57   Log-Likelihood:                 18306.
No. Observations:                1760   AIC:                        -3.660e+04
```

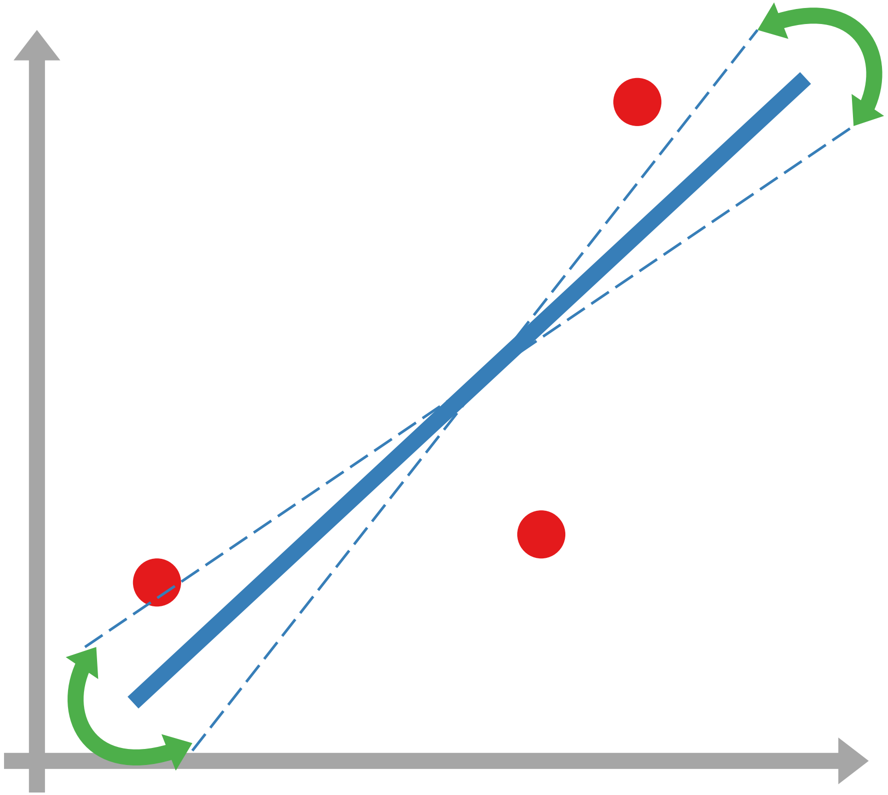
Clustered Standard Errors

What’s Wrong Here?



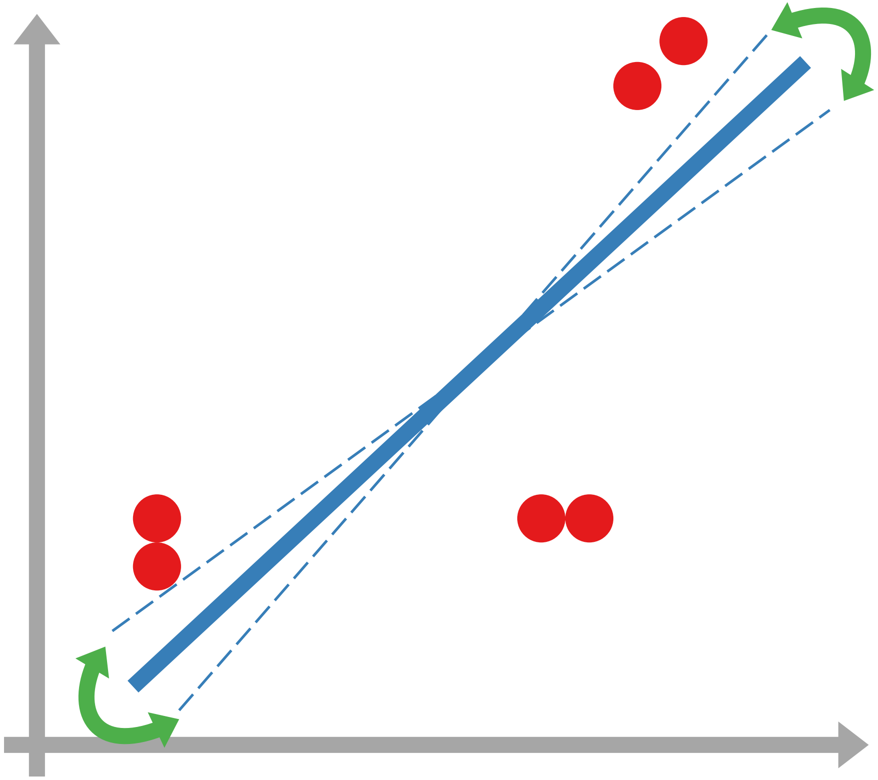
You’re lying to your model!
You’re saying you have 12 **independent** observations
Not true! You have 3

What’s a standard error?



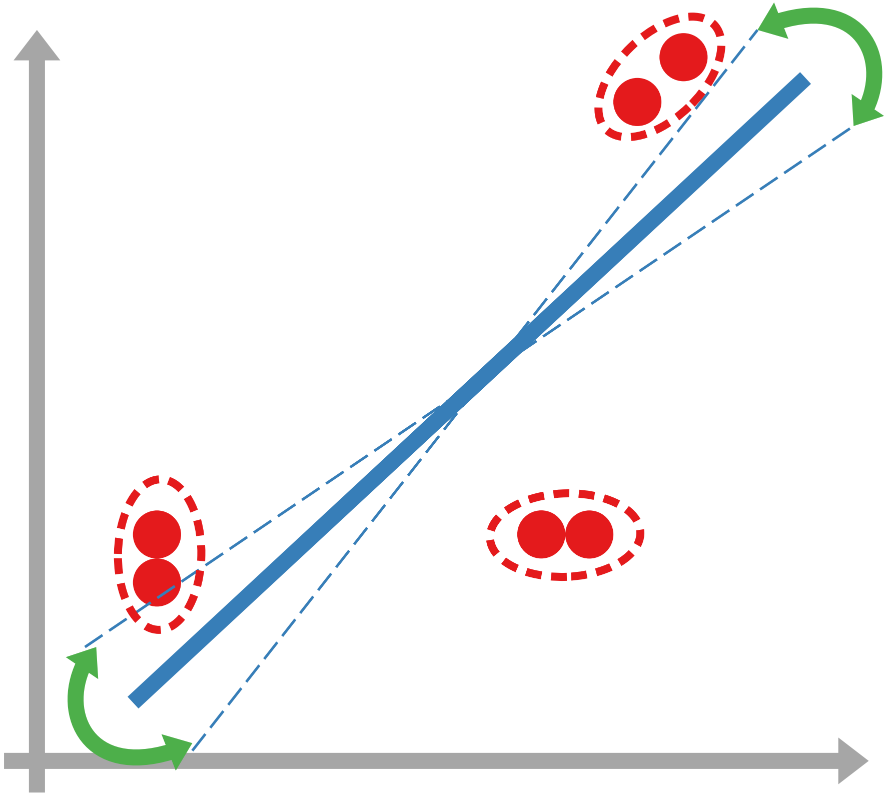
- How sure you are about your estimation

What’s a standard error?



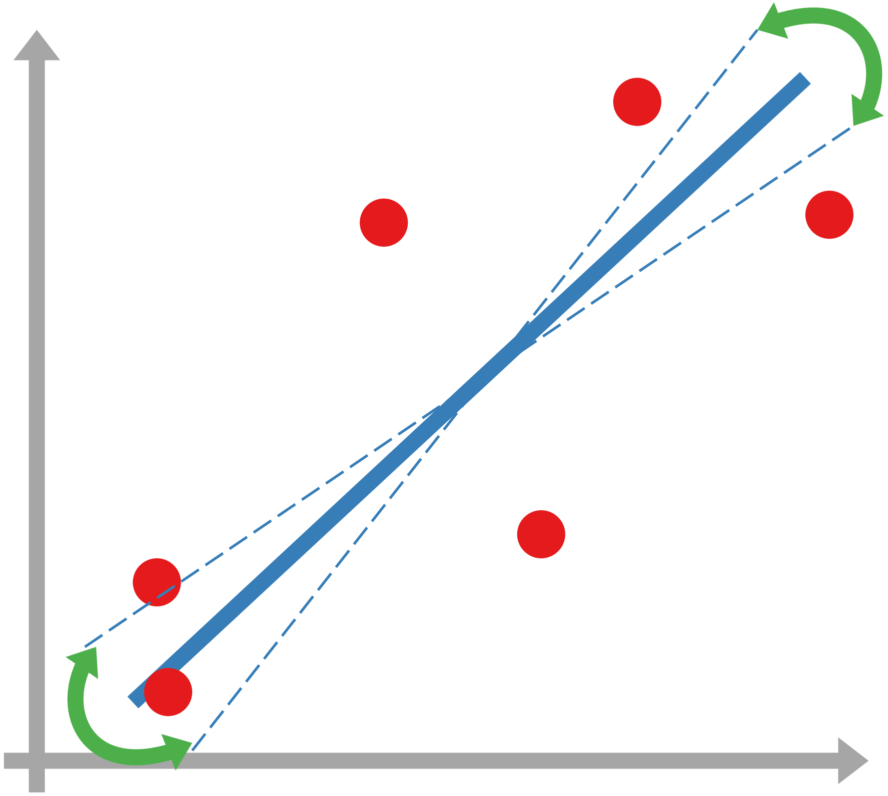
- How sure you are about your estimation
- More obs = more confidence

What’s a standard error?



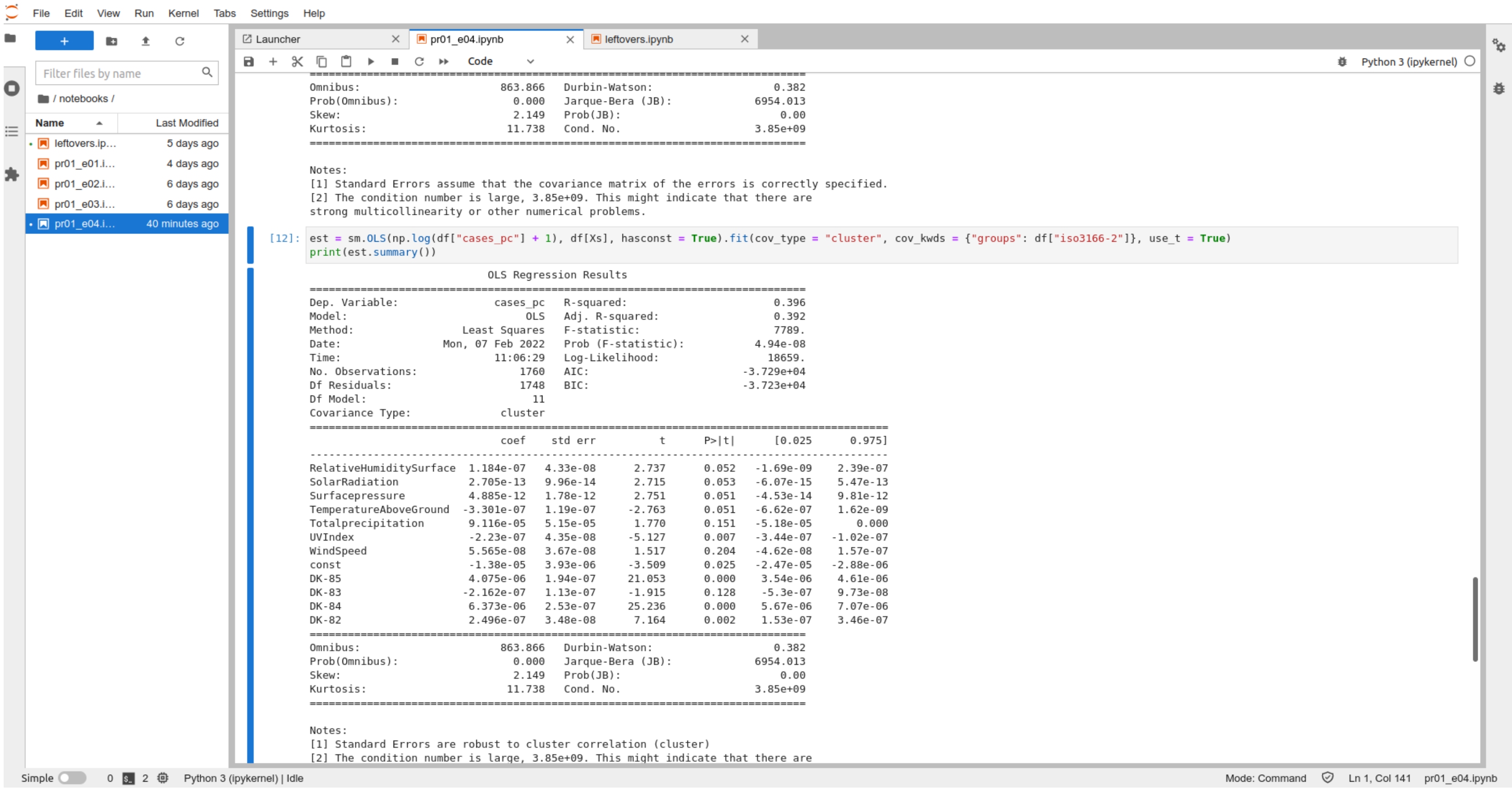
- How sure you are about your estimation
- More obs = more confidence
- But not if they’re part of the same group!

Clustered Standard Errors



- A true new obs would be independent from the previous ones
- Doing CSEs can take this into account

Let’s Cluster Some Errors!



```
==============================================================================
Omnibus:                      863.866   Durbin-Watson:                   0.382
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6954.013
Skew:                           2.149   Prob(JB):                         0.00
Kurtosis:                      11.738   Cond. No.                     3.85e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.85e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
[12]: est = sm.OLS(np.log(df["cases_pc"] + 1), df[Xs], hasconst = True).fit(cov_type = "cluster", cov_kwds = {"groups": df["iso3166-2"]}, use_t = True)
      print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               cases_pc   R-squared:                       0.396
Model:                            OLS   Adj. R-squared:                  0.392
Method:                 Least Squares   F-statistic:                     7789.
Date:                Mon, 07 Feb 2022   Prob (F-statistic):           4.94e-08
Time:                        11:06:29   Log-Likelihood:                 18659.
No. Observations:                1760   AIC:                        -3.729e+04
Df Residuals:                    1748   BIC:                        -3.723e+04
Df Model:                          11
Covariance Type:              cluster
===========================================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------
RelativeHumiditySurface  1.184e-07   4.33e-08      2.737      0.052   -1.69e-09    2.39e-07
SolarRadiation           2.705e-13   9.96e-14      2.715      0.053   -6.07e-15    5.47e-13
Surfacepressure          4.885e-12   1.78e-12      2.751      0.051   -4.53e-14    9.81e-12
TemperatureAboveGround  -3.301e-07   1.19e-07     -2.763      0.051   -6.62e-07    1.62e-09
Totalprecipitation       9.116e-05   5.15e-05      1.770      0.151   -5.18e-05       0.000
UVIndex                  -2.23e-07   4.35e-08     -5.127      0.007   -3.44e-07   -1.02e-07
WindSpeed                5.565e-08   3.67e-08      1.517      0.204   -4.62e-08    1.57e-07
const                    -1.38e-05   3.93e-06     -3.509      0.025   -2.47e-05   -2.88e-06
DK-85                    4.075e-06   1.94e-07     21.053      0.000    3.54e-06    4.61e-06
DK-83                   -2.162e-07   1.13e-07     -1.915      0.128    -5.3e-07    9.73e-08
DK-84                    6.373e-06   2.53e-07     25.236      0.000    5.67e-06    7.07e-06
DK-82                    2.496e-07   3.48e-08      7.164      0.002    1.53e-07    3.46e-07
==============================================================================
Omnibus:                      863.866   Durbin-Watson:                   0.382
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6954.013
Skew:                           2.149   Prob(JB):                         0.00
Kurtosis:                      11.738   Cond. No.                     3.85e+09
==============================================================================

Notes:
[1] Standard Errors are robust to cluster correlation (cluster)
[2] The condition number is large, 3.85e+09. This might indicate that there are
```

Q&A