

regs

March 31, 2024

```
[ ]: import polars as pl
import statsmodels.formula.api as smf
import numpy as np
from stargazer.stargazer import Stargazer

[ ]: df = pl.read_parquet("../data/combined_all.parquet")
pd_df = df.to_pandas()

[ ]: pd_df["arwu_ranked_num"] = pd_df["arwu_ranked_num"] / pd_df["population"] * 1_000_000
pd_df["gdp_pc_growth"] = 100 * pd_df["gdp_pc_growth"]
# pd_df["gdp_pc"] = np.log(pd_df["gdp_pc"])
# pd_df["population"] = np.log(pd_df["population"])

[ ]: pisa_no_fix = smf.ols("gdp_pc_growth ~ gdp_pc + imo_total_score + math + in_math99 + arwu_ranked_num*gdp_pc + primary_completion + lower_sec_completion + upper_sec_completion + population + democracy_eiu", pd_df[pd_df["year"].isin([2003, 2006, 2009, 2012, 2015, 2018])]).fit()
pisa = smf.ols("gdp_pc_growth ~ gdp_pc + imo_total_score + math + in_math99 + arwu_ranked_num*gdp_pc + primary_completion + lower_sec_completion + upper_sec_completion + population + democracy_eiu + C(year)", pd_df[pd_df["year"].isin([2003, 2006, 2009, 2012, 2015, 2018])]).fit()
pisa_country = smf.ols("gdp_pc_growth ~ gdp_pc + imo_total_score + math + in_math99 + arwu_ranked_num*gdp_pc + primary_completion + lower_sec_completion + upper_sec_completion + population + democracy_eiu + country", pd_df[pd_df["year"].isin([2003, 2006, 2009, 2012, 2015, 2018])]).fit()
pisa_no_score = smf.ols("gdp_pc_growth ~ gdp_pc + imo_total_score + arwu_ranked_num*gdp_pc + primary_completion + lower_sec_completion + upper_sec_completion + population + democracy_eiu + C(year)", pd_df[pd_df["year"].isin([2003, 2006, 2009, 2012, 2015, 2018]) & (pd_df["math"].notnull())]).fit()
pisa_all_no_interac = smf.ols("gdp_pc_growth ~ gdp_pc + imo_total_score + math + read + science + in_math99 + arwu_ranked_num + primary_completion + lower_sec_completion + upper_sec_completion + population + democracy_eiu + C(year)", pd_df[pd_df["year"].isin([2003, 2006, 2009, 2012, 2015, 2018])]).fit()
```

```

pisa_all = smf.ols("gdp_pc_growth ~ gdp_pc + imo_total_score + math + read +
↳science + in_math99 + arwu_ranked_num*gdp_pc + primary_completion +
↳lower_sec_completion + upper_sec_completion + population + democracy_eiu +
↳C(year) ", pd_df[pd_df["year"].isin([2003, 2006, 2009, 2012, 2015, 2018])]).
↳fit()
non_pisa = smf.ols("gdp_pc_growth ~ gdp_pc + democracy_eiu + imo_total_score +
↳arwu_ranked_num*gdp_pc + primary_completion + lower_sec_completion +
↳upper_sec_completion + population + C(year) ", pd_df).fit()
non_pisa_country = smf.ols("gdp_pc_growth ~ gdp_pc + democracy_eiu +
↳imo_total_score + arwu_ranked_num*gdp_pc + primary_completion +
↳lower_sec_completion + upper_sec_completion + population + C(year) +
↳country", pd_df).fit()
non_pisa_no_interac = smf.ols("gdp_pc_growth ~ gdp_pc + democracy_eiu +
↳imo_total_score + arwu_ranked_num + primary_completion +
↳lower_sec_completion + upper_sec_completion + population + C(year) ", pd_df).
↳fit()

```

```

[ ]: pisa_sg = Stargazer([pisa_no_score, pisa_no_fix, pisa, pisa_country])

```

```

[ ]: pisa_sg.covariate_order(["in_math99", "imo_total_score", "arwu_ranked_num",
↳"arwu_ranked_num:gdp_pc", "math", "gdp_pc", "primary_completion",
↳"lower_sec_completion", "upper_sec_completion"])
pisa_sg.add_line('Time Effects', ['Yes', 'No', 'Yes', 'No'])
pisa_sg.add_line('Fixed Effects', ['No', 'No', 'No', 'Yes'])
pisa_sg.add_line('Entities', [48, 48, 48, 48])
pisa_sg

```

```

[ ]: <stargazer.stargazer.Stargazer at 0x32a1605c0>

```

```

[ ]: non_pisa_sg = Stargazer([non_pisa_no_interac, non_pisa, non_pisa_country])
non_pisa_sg.covariate_order(["imo_total_score", "arwu_ranked_num",
↳"arwu_ranked_num:gdp_pc", "gdp_pc", "primary_completion",
↳"lower_sec_completion", "upper_sec_completion"])
non_pisa_sg.add_line('Time Effects', ['Yes', 'Yes', 'Yes'])
non_pisa_sg.add_line('Fixed Effects', ['No', 'No', 'Yes'])
non_pisa_sg.add_line('Entities', [137, 137, 137])
non_pisa_sg

```

```

[ ]: <stargazer.stargazer.Stargazer at 0x32a162240>

```

```

[ ]: group1 = pd_df[(pd_df["arwu_ranked_num"].notnull()) & (pd_df["math"].notnull())
↳& (pd_df["imo_total_score"].notnull()) & (pd_df["primary_completion"].
↳notnull()) & (pd_df["lower_sec_completion"].notnull()) &
↳(pd_df["upper_sec_completion"].notnull()) & (pd_df["democracy_eiu"].
↳notnull())]
group1.shape

```

```
[ ]: (109, 16)
```

```
[ ]: group2 = pd_df[(pd_df["gdp_pc"].notnull()) & (pd_df["arwu_ranked_num"].
↳notnull()) & (pd_df["imo_total_score"].notnull()) &
↳(pd_df["primary_completion"].notnull()) & (pd_df["lower_sec_completion"].
↳notnull()) & (pd_df["upper_sec_completion"].notnull()) &
↳(pd_df["democracy_eiu"].notnull()))
group2.shape
```

```
[ ]: (746, 17)
```

```
[ ]: group1.describe()
```

```
[ ]:
```

	year	gdp_pc	gdp_pc_growth	primary_completion	\
count	109.000000	109.000000	109.000000	109.000000	
mean	2013.266055	9.774552	2.371332	90.126140	
std	4.068038	0.901493	2.401491	10.448936	
min	2006.000000	7.360611	-4.358317	51.353222	
25%	2012.000000	9.118337	0.875477	82.948372	
50%	2015.000000	9.788621	2.105525	93.079300	
75%	2015.000000	10.563727	3.933610	98.932922	
max	2018.000000	11.493145	9.142617	100.000000	

	lower_sec_completion	upper_sec_completion	population	\
count	109.000000	109.000000	109.000000	
mean	74.816173	58.464856	16.677068	
std	19.717275	21.212708	1.516022	
min	31.719400	22.409929	13.005949	
25%	57.490002	39.669998	15.506307	
50%	78.459999	57.732929	16.215255	
75%	93.785004	77.846474	17.921958	
max	99.816658	97.399788	19.604976	

	arwu_ranked_num	arwu_sum_score	math	read	science	\
count	109.000000	109.000000	109.000000	107.000000	109.000000	
mean	12.018349	97.844037	457.618259	458.898614	462.218945	
std	33.010375	390.887652	54.171813	46.089685	49.296882	
min	0.000000	0.000000	327.250491	344.298016	334.946347	
25%	0.000000	0.000000	412.679509	423.652913	422.451115	
50%	2.000000	0.000000	473.142717	468.690188	473.713287	
75%	9.000000	0.000000	497.395753	497.643493	500.048433	
max	217.000000	2190.400000	568.359669	548.583819	548.876010	

	in_math99	imo_total_score	democracy_eiu	arwu_ranked_num_pc
count	109.000000	109.000000	109.000000	109.000000
mean	0.865194	76.944954	7.299083	0.248798
std	1.545727	55.338995	1.335456	0.319323

min	0.000000	0.000000	2.760000	0.000000
25%	0.028512	21.000000	6.550000	0.000000
50%	0.373039	82.000000	7.540000	0.056237
75%	1.041082	117.000000	8.130000	0.477434
max	9.501632	212.000000	9.930000	1.211386