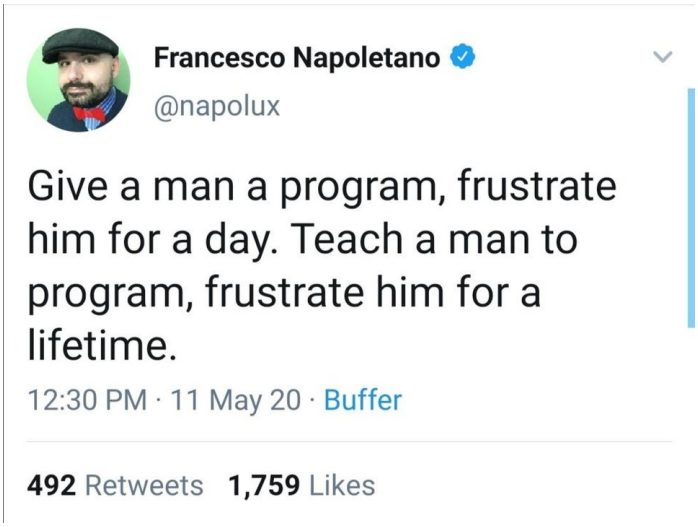
# DBA Advanced Data and Regression Techniques

Summer 2020, Joost Impink (joost@ufl.edu)

## About the course

### Main idea

Your dissertation requires an empirical component. Either ‘survey/experimental’ (Walter Leite’s course) or ‘archival’ (this course). This course will help you get your data analytics skills (creating datasets, regressions) at the right level.




### Typical programs/languages used:

SAS - strength: creating datasets

Stata - strength: easy to do statistical analyses, and organize output

SPSS - strengths: -

Excel - strengths: organize hand-collected data, making ‘pretty’ tables, graphs

R, Python - strengths: machine learning, textual analysis, automating tasks

# Residency 1 topics:

Getting familiar with a typical ‘pipeline’, such as Excel -> SAS -> Stata

Each day we will create a dataset in SAS, and bring it to Stata, so you will learn:

- Commonly used datasets on WRDS (Compustat, CRSP)
- How to to merge datasets (and common issues) and creating variables
- Then bring the dataset to Stata, where making tables (descriptives, correlation, etc) and doing regressions is easy and also more manageable
    - For example, a logistic regression in SAS is 4-5 lines of code, and SAS creates seven different output datasets, which is a pain if you want to create a table with (say) four different regression specifications. In Stata each regression is one line (including ‘storing’ it), and exporting the stored output as an Excel sheet is also a one-liner.

The preparation materials for this course covered the most commonly used SAS procedures.

- Note: I don’t expect you to have memorized/mastered these at this point

We will keep using these main procedures and we will also go into writing SAS macros. Macros allows us to create reusable ‘components’, giving concise code, lower odds of having errors, and a faster workflow when the code needs to be reused (in a sensitivity test or another project).

- ‘New’ researchers often have a single *very* long SAS file (think 2,000-5,000 lines) with all the code
- ‘Effective’ researchers have many small SAS files, with the ‘main’ SAS file (think 200-300 lines) using the smaller ones (that is, the main file consists of macro calls, and the other files are macros)

# Residency 2 topics:

Continuation of residency 1.

More SAS macros and methods related topics:

- Firm fixed effects
- Clustering of standard errors
- Dealing with outliers
- Multicollinearity
- Scaling
- How to organize sensitivity tests

# Overall

Keep this in mind:

- No need to remember details, but pay attention to what the different SAS procedures can do (a handful of procedures cover 95% of the work you would typically do)
- Copy/pasting working code and edit it (as opposed to typing into an empty screen)
- Google searches are often very effective
- The SAS log gives helpful errors (mostly), keep an eye on the #observations for ‘working’ code
- Inspect datasets that you created (if no errors doesn’t mean you got what you had in mind)
- Often it makes sense to first write code on a small test dataset before running on the full dataset (less waiting)
- If you already know other software quite well (Excel, SPSS), it is natural to be attached to using these. However, Excel/SPSS are rather limited in capabilities.
- Add comments to your code, so that you remember a few weeks later what you were thinking
- For your dissertation: don’t cut corners, take your time

# Course grading

See syllabus: 50% assignments, 50% participation. After residency 1 and 2 I will assign a few problems that I will grade. .