

# CourseMatch at SciencesPo

---

CompEcon 2019



Monumental Tasks

monumental solutions

# Course Allocation at ScPo

---

- First come first serve
- low satisfaction ratings
- look how Wharton MBA does it?

# Term Project - Desired Output

---

- a working prototype for the course match problem (90%)
- a design document that clearly outlines the code in such a way that it can be passed on. (80%)
- Finish API and testing (0%)

# Set up Teams

---

1. start with multiple teams. Towards end merge into one team
2. will take the best parts of each team's work
3. competition!

# Individual Homework for next time

---

1. read the report and the paper here: <https://github.com/ScPo-CompEcon/CourseMatch.jl>
2. Think about
  1. what is the purpose of the code to be written?
  2. what are the main parts of the problem?
  3. how are those parts related?
3. Try Out the Code - need to port to julia v1.0

# Group Homework for Next Time

---

- Every Group gives a 5 min presentation. no slides.

## Topics

1. **theory**: what is the relevant theory for the ScPo problem. explain efficiency vs fairness.
2. **computation**: what is the computational cost in terms of time? how does it scale with the number of servers? integrated or independent architecture?
3. **algorithm**: explain the algorithms outlined in the paper.

# Design Document

---

After we all agree on a design document, we start implementation.

- how to check whether we get a correct solution?
- start by designing a test suite
- create and assign issues on the repo: don't do anything on the code if there is not an issue for it. Let others know what you are working on!
- attack each issue with a corresponding pull request.
- \* First write a new test if the current test suite does not cover your PR. Then make the PR.



# Groups

---

- Algorithm
- Theory
- computation

