Social Data Science: Machine Learning & Econometrics

Exercise class 6

March 27, 2020

Todays quick warmup

Q: Write a *decorator* `retry(n)` that when applied to a function (e.g. a scraper) retries a failed function call up to n times.

Apply your decorator to the following function

```
from random import uniform

def this_might_fail():
    c = uniform(0,1) > 0.8
    if c: return 'Success'
    raise ValueError('c to small')
```

Todays quick warmup - solution

Functions can return functions! We need 3 layers because we want to pass the argument *n* to the nested function.

```
def retry(n):
    def _outer(f):
        def _wrapped(*args, **kwargs):
            for _ in range(n):
                try:
                    return f(*args, **kwargs)
                except: pass
            raise ValueError(f'No luck in {n} tries')
        return _wrapped
    return _outer
```

- Add a `time.sleep(0.2)` and use this for your scrapers!
- Decorators add functionality *around* the function they wrap.

Last lecture in a nutshell

- Install geopandas and play around, it is brilliant software.
- Geographical boundaries can be a source of "random" variation. Especially if boundaries are placed arbitrarily.