

Linear regression on non-imaging data

Warning: this is a somewhat complicated example and is probably not the final version: I'm hoping to find cleaner ways to do some of this.

The code below regresses the NEO-FFI agreeableness (A) measure against conscientiousness (C), not because I expect the result to be interesting, but because the [Connectome Python tutorial](#) already shows how to get those values.

You'll need a few Python packages installed (numpy, scipy, matplotlib) to make this work. I ran this in an [IPython](#) notebook with pylab for plotting.

```
In [1]: import pyxnat
In [2]: import numpy as np
In [3]: from scipy import stats
In [4]: cdb = pyxnat.Interface('https://db.humanconnectome.org', 'username', 'password')
In [5]: fields = ['{}/{}'.format('nt:scores', f) for f in ['SUBJECT_ID']+['NEO_NEOFAC_{}'.format(e) for e in
'AOCNE']]
In [6]: constraints = [('nt:scores/PROJECTS', 'LIKE', '%HCP_Q2')]
In [7]: scores_json = cdb.select('nt:scores', fields).where(constraints)
In [8]: scores_json.headers()
Out[8]: ['subject_id', 'neo_neofac_a', 'neo_neofac_c', 'neo_neofac_e', 'neo_neofac_n', 'neo_neofac_o']
In [9]: scores = np.array([[float(v) if v else np.nan for v in row.values()[1:]] for row in scores_json.data]).
transpose()
In [10]: missing = np.isnan(scores[0])
In [11]: neo_a = scores[0][~missing]
In [12]: neo_c = scores[1][~missing]
In [13]: slope, intercept, r_value, p_value, std_err = stats.linregress(neo_a, neo_c)
In [14]: line = slope * neo_a + intercept
In [15]: plot(neo_a, line, 'r-', neo_a, neo_c, 'o')
Out[15]: [<matplotlib.lines.Line2D at 0x54bedd0>, <matplotlib.lines.Line2D at 0x54c4050>]
```

