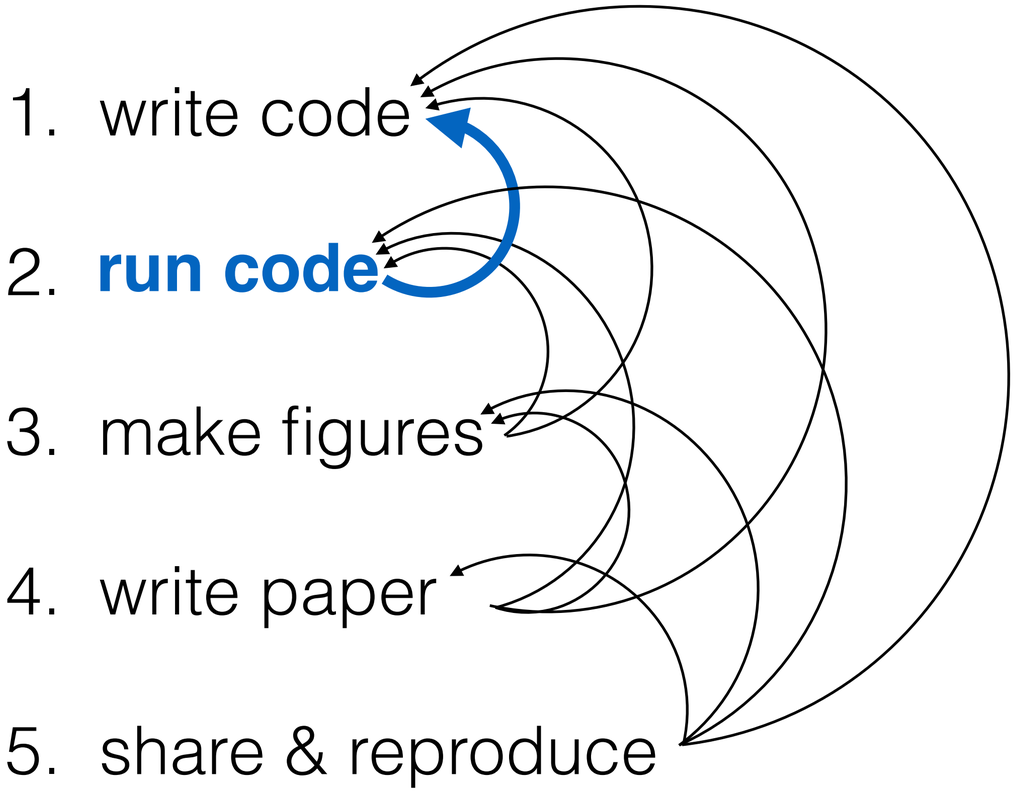
Interactive CSE with IPython and Jupyter

Min Ragan-Kelley, Simula Research Lab
Project Jupyter Contributors

How do we do computational research?



IPython

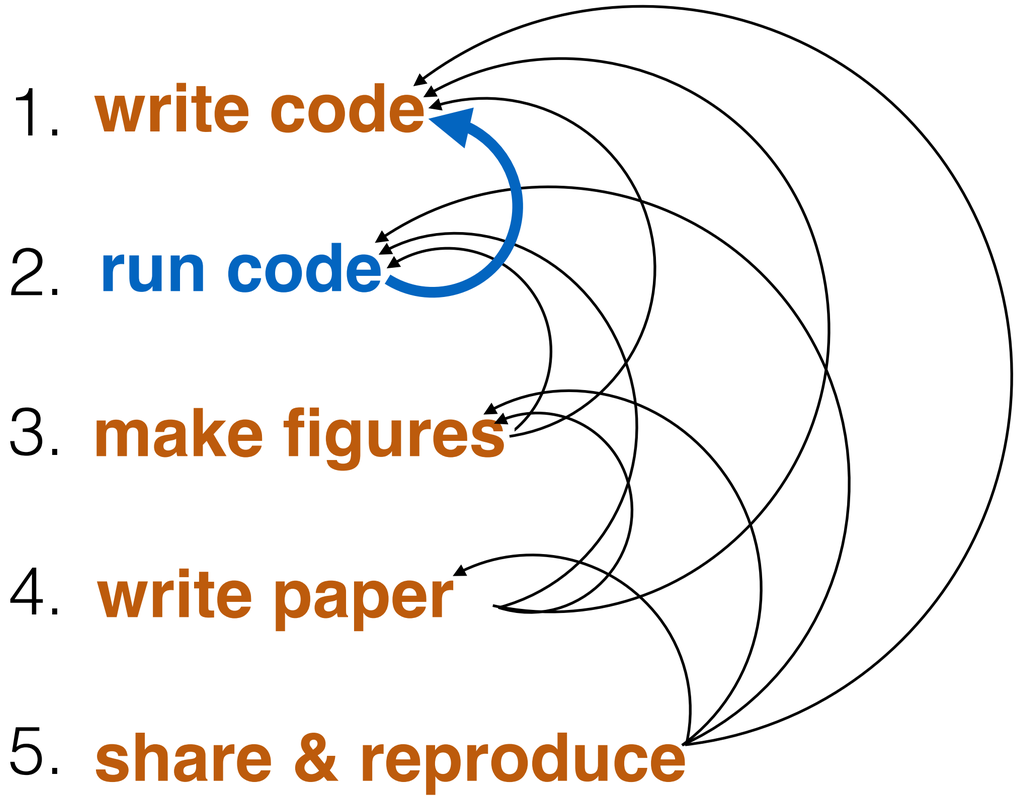
IPython
Interactive Python

helps run code

- tab completion
- introspection
- %magics

What about Jupyter?

1. write code
2. run code
3. make figures
4. write paper
5. share & reproduce

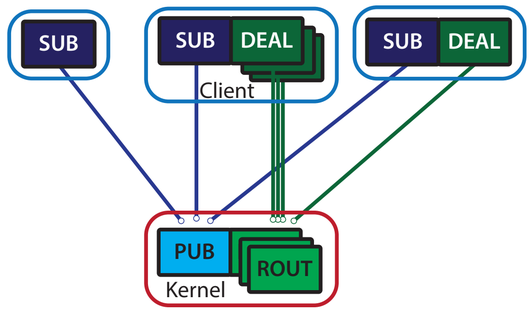


IPython

jupyter

What is Jupyter?

Rich REPL Protocol



ØMQ + JSON

Document Format

We have already computed $P(X|A)$ above. On the other hand, $P(X|\sim A)$ is subjective: our code can pass tests but still have a bug in it, though the probability there is a bug present is reduced. Note this is dependent on the number of tests performed, the degree of complication in the tests, etc. Let's be conservative and assign $P(X|\sim A) = 0.5$. Then

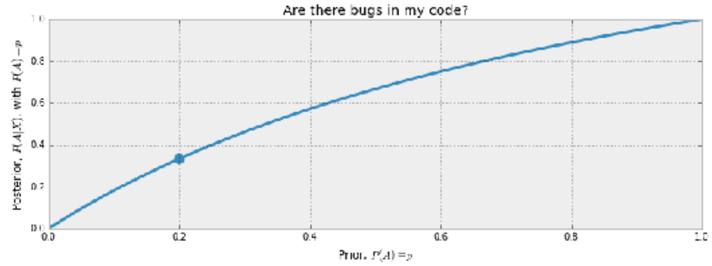
$$P(A|X) = \frac{1 \cdot p}{1 \cdot p + 0.5(1-p)}$$

$$= \frac{2p}{1+p}$$

This is the posterior probability. What does it look like as a function of our prior, $p \in [0,1]$?

```
figsize(12.5, 4)
p = np.linspace(0, 1, 50)
plt.plot(p, 2 * p / (1 + p), color="#348ABD", lw=3)
# plt.fill_between(p, 2*p/(1+p), alpha=.5, facecolor=["#A60628"])
plt.scatter(0.2, 2 * (0.2) / 1.2, s=140, c="#348ABD")
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel("Prior, $P(A) = p$")
plt.ylabel("Posterior, $P(A|X)$, with $P(A) = p$")
plt.title("Are there bugs in my code?")
```

```
<matplotlib.text.Text at 0x10b1de550>
```



https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers

Jupyter Protocol

REP*L over JSON + ØMQ

- Read
- Eval
- Print*
- Loop

```
msg_type = 'execute_request'
content = {
    'code' : """
        import pandas as pd
        df = pd.read_csv('mydata.csv')
        """,
```

```
msg_type = 'execute_reply'
content = {
```

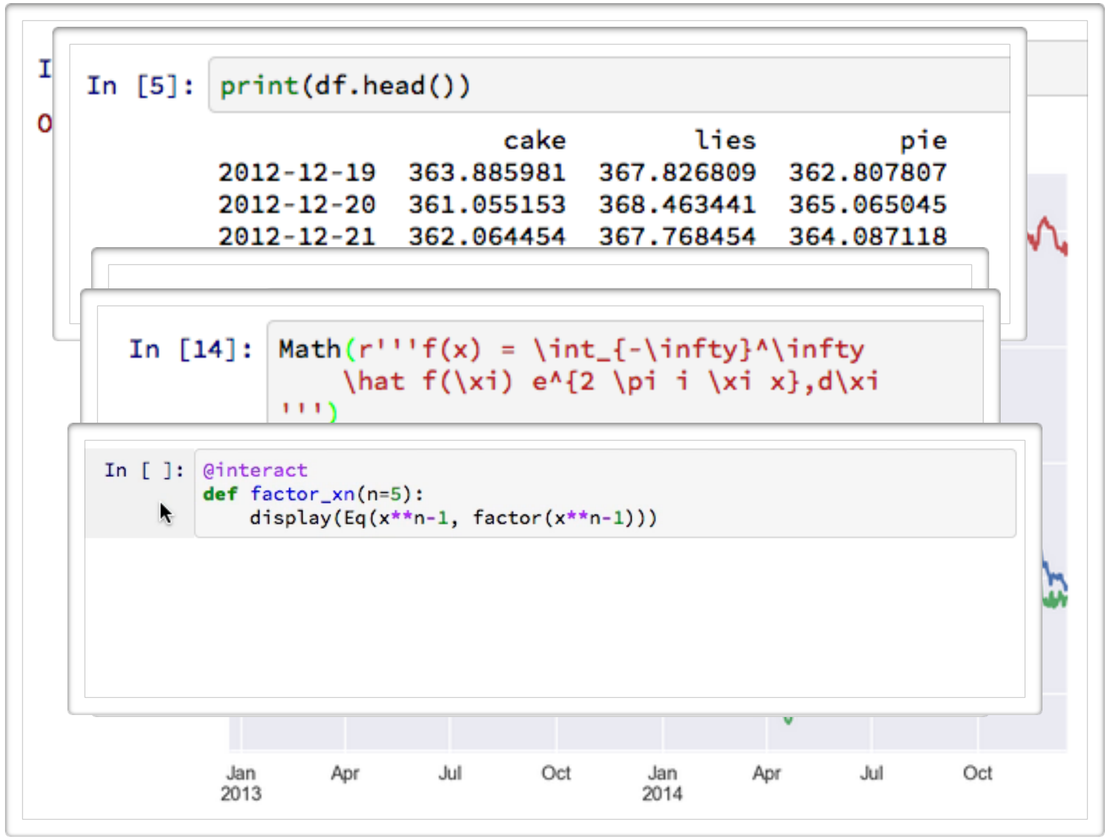
```
msg_type = 'display_data'
content = {
    'data': {
        "text/plain": "<MyDataFrame at 0x...>",
        "text/html": "<table>...</table>",
        ...
    },
    'metadata': {},
    ...
}
```

Jupyter Protocol

supercharge the P in REP*L

any mime-type output

- text
- svg, png, jpeg
- latex, pdf
- html, javascript
- interactive widgets

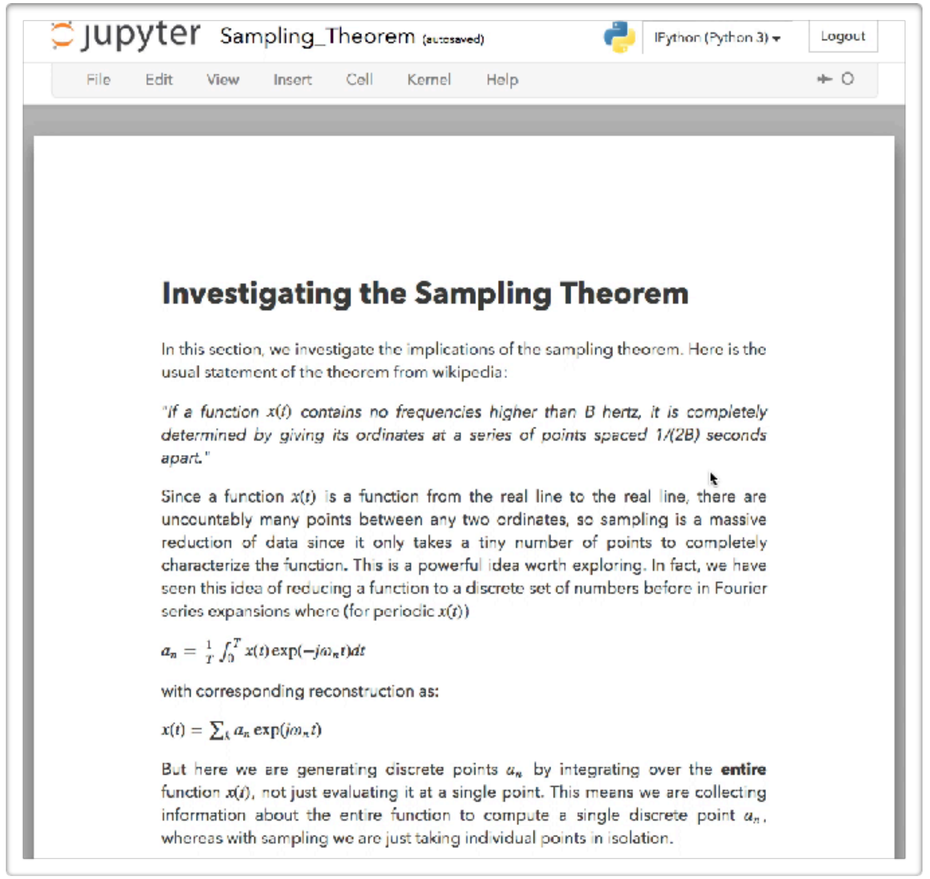


Jupyter Protocol

is language agnostic

Jupyter Notebooks

- notebook = sequence of cells
- text cell = markdown + latex)
- code cell = REP (input + output)
- metadata everywhere



https://github.com/unpingco/Python-for-Signal-Processing