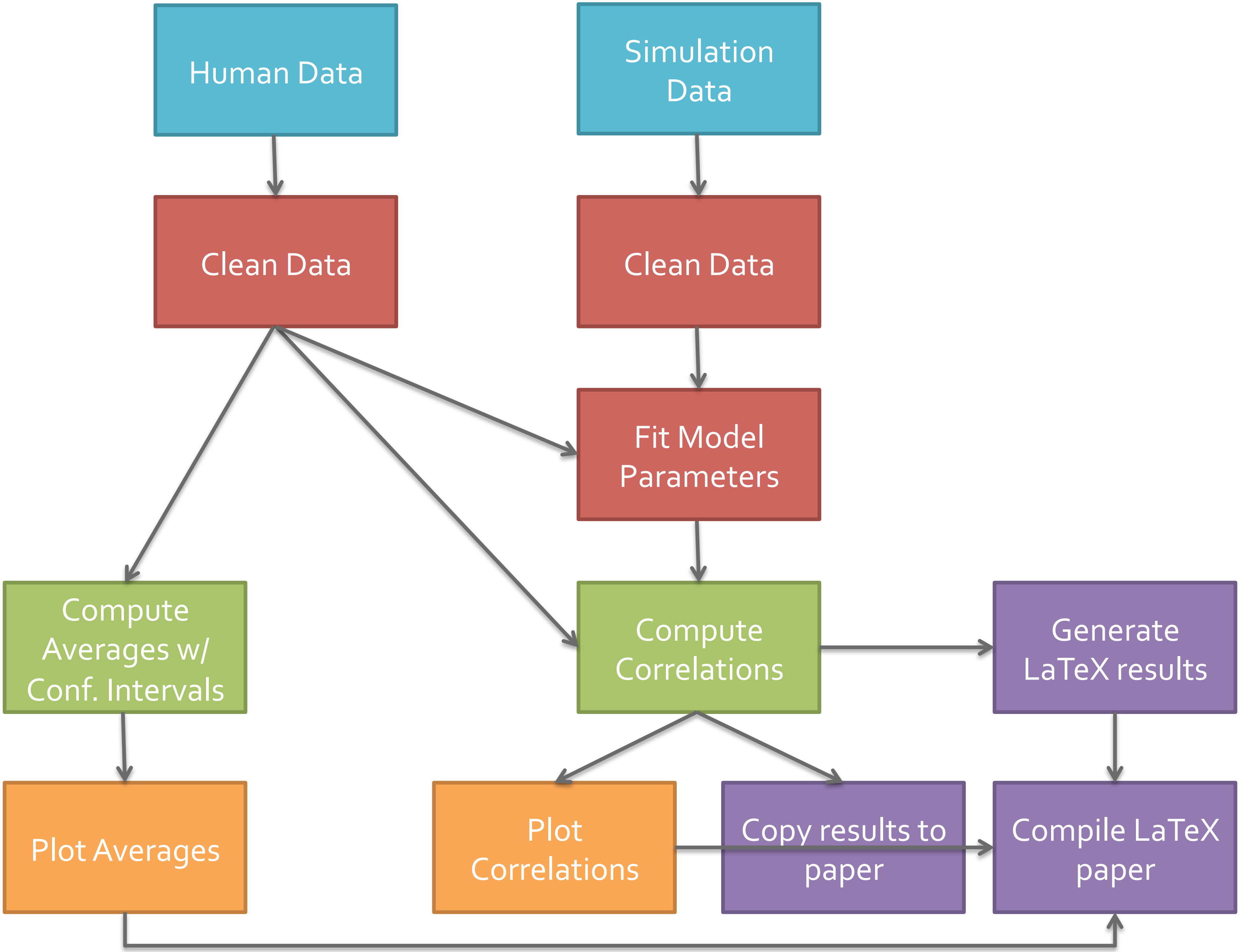
Reproducible, One-Button Workflows with the Jupyter Notebook and SCons

Jessica Hamrick

UC Berkeley & Project Jupyter

@jhamrick

An idealized analysis pipeline…

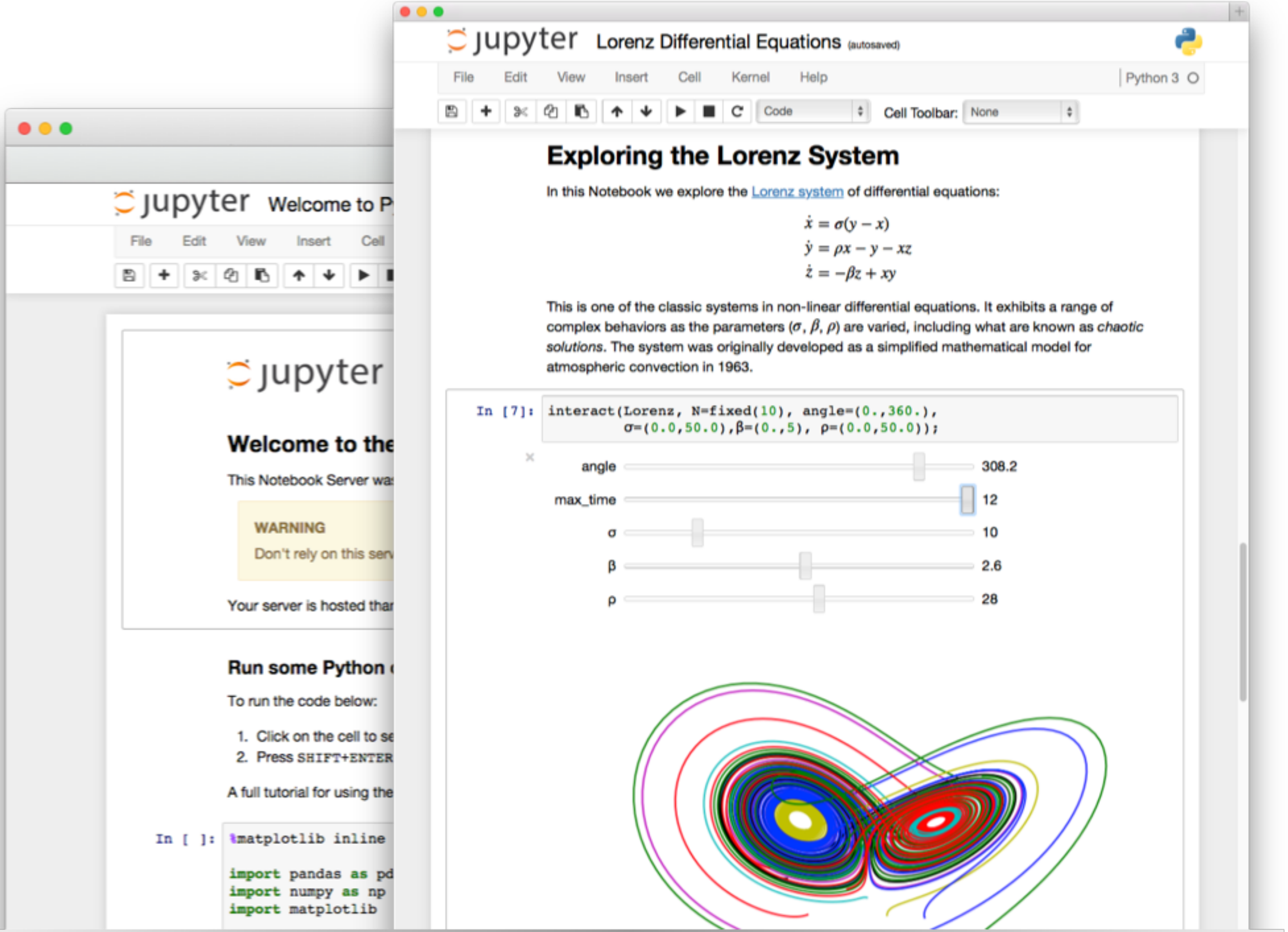

Human Data
Simulation Data
Clean Data
Clean Data
Fit Model Parameters
Compute Averages w/ Conf. Intervals
Compute Correlations
Generate LaTeX results
Plot Averages
Plot Correlations
Copy results to paper
Compile LaTeX paper

How do we achieve this idealized workflow?

Approach 1: one big script

Drawback: no interactivity ☹

Notebooks are interactive!



Welcome to the

This Notebook Server wa

WARNING

Don't rely on this serv

Your server is hosted thar

Run some Python

To run the code below:

1. Click on the cell to se
2. Press `SHIFT+ENTER`

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
        import matplotlib
```

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0));
```

angle 308.2

max_time 12

σ 10

β 2.6

ρ 28

Approach 2: one big notebook

Drawback: can be unwieldy ☹

Approach 3: multiple scripts and / or notebooks

Drawback: dependency hell ☹

Approach 4: multiple scripts and / or notebooks with a traditional build system

Traditional Build Systems

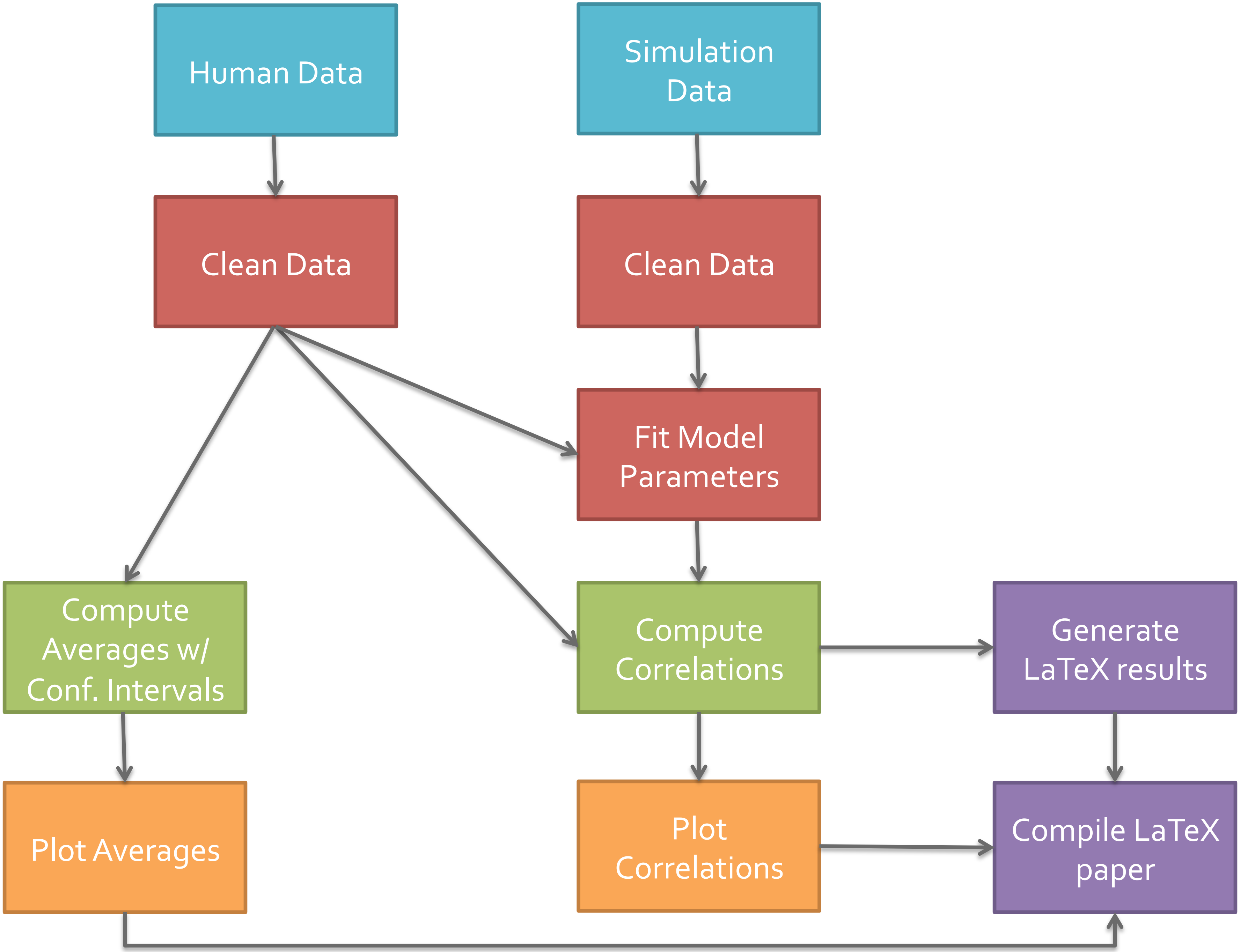
Make

CMake

SCons

…

Idea: Run a series of commands in order, ensuring dependencies are satisfied



Human Data

Simulation Data

Clean Data

Clean Data

Fit Model Parameters

Compute Averages w/ Conf. Intervals

Compute Correlations

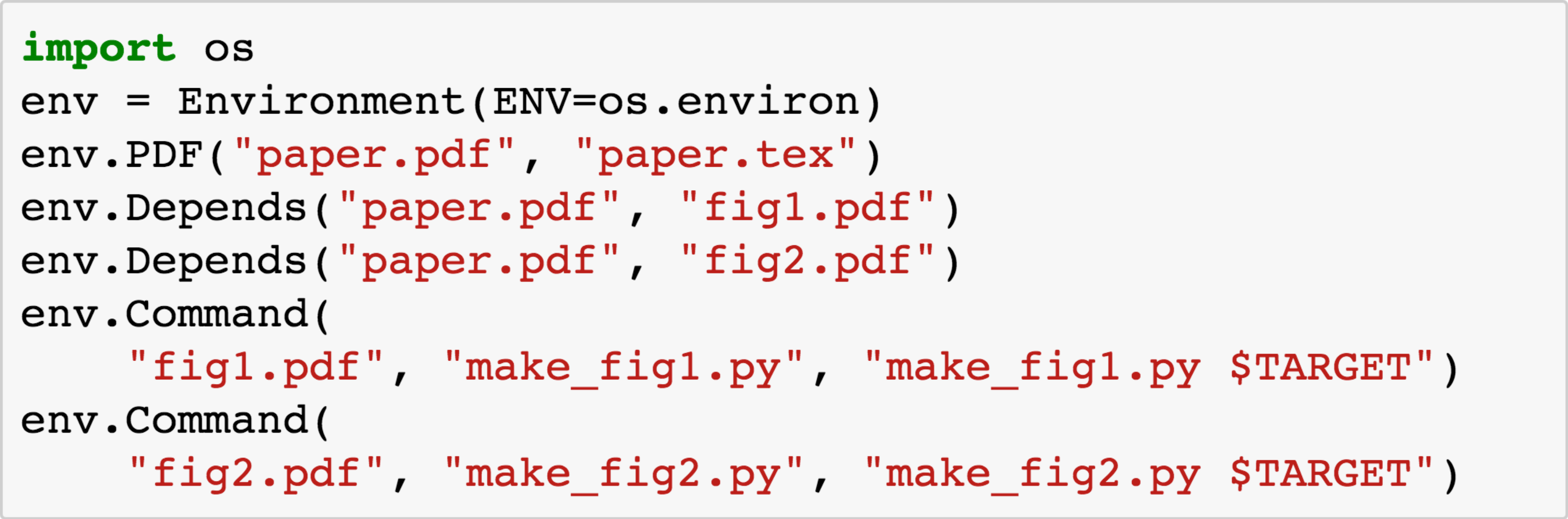
Generate LaTeX results

Plot Averages

Plot Correlations

Compile LaTeX paper

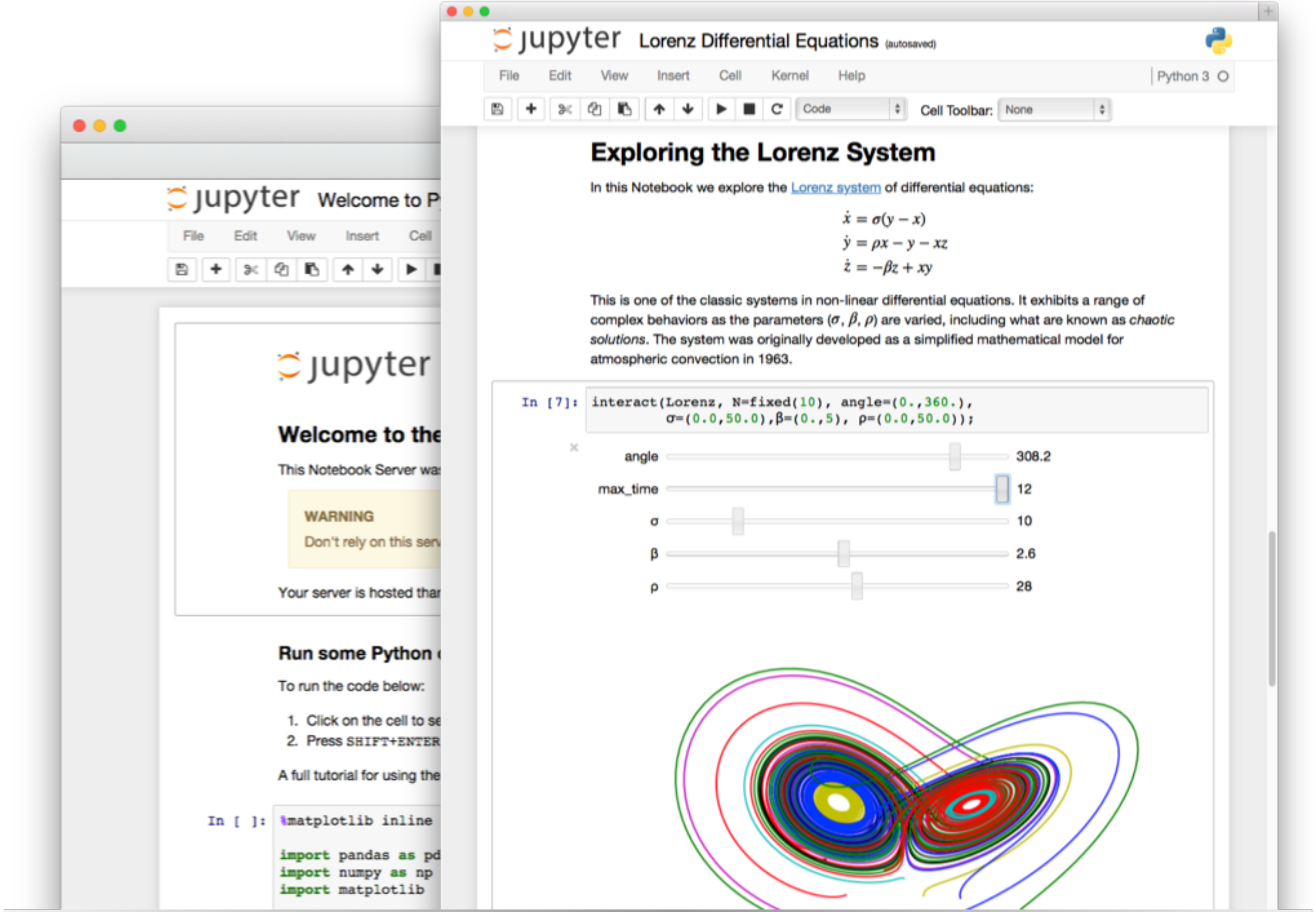
An example SConstruct file



```
import os
env = Environment(ENV=os.environ)
env.PDF("paper.pdf", "paper.tex")
env.Depends("paper.pdf", "fig1.pdf")
env.Depends("paper.pdf", "fig2.pdf")
env.Command(
    "fig1.pdf", "make_fig1.py", "make_fig1.py $TARGET")
env.Command(
    "fig2.pdf", "make_fig2.py", "make_fig2.py $TARGET")
```

```
$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
... a whole bunch of output ...
scons: done building targets.
```

But what if we want to run *notebooks?*



jupyter

Welcome to the

This Notebook Server wa

WARNING

Don't rely on this serv

Your server is hosted thar

Run some Python

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
        import matplotlib
```

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

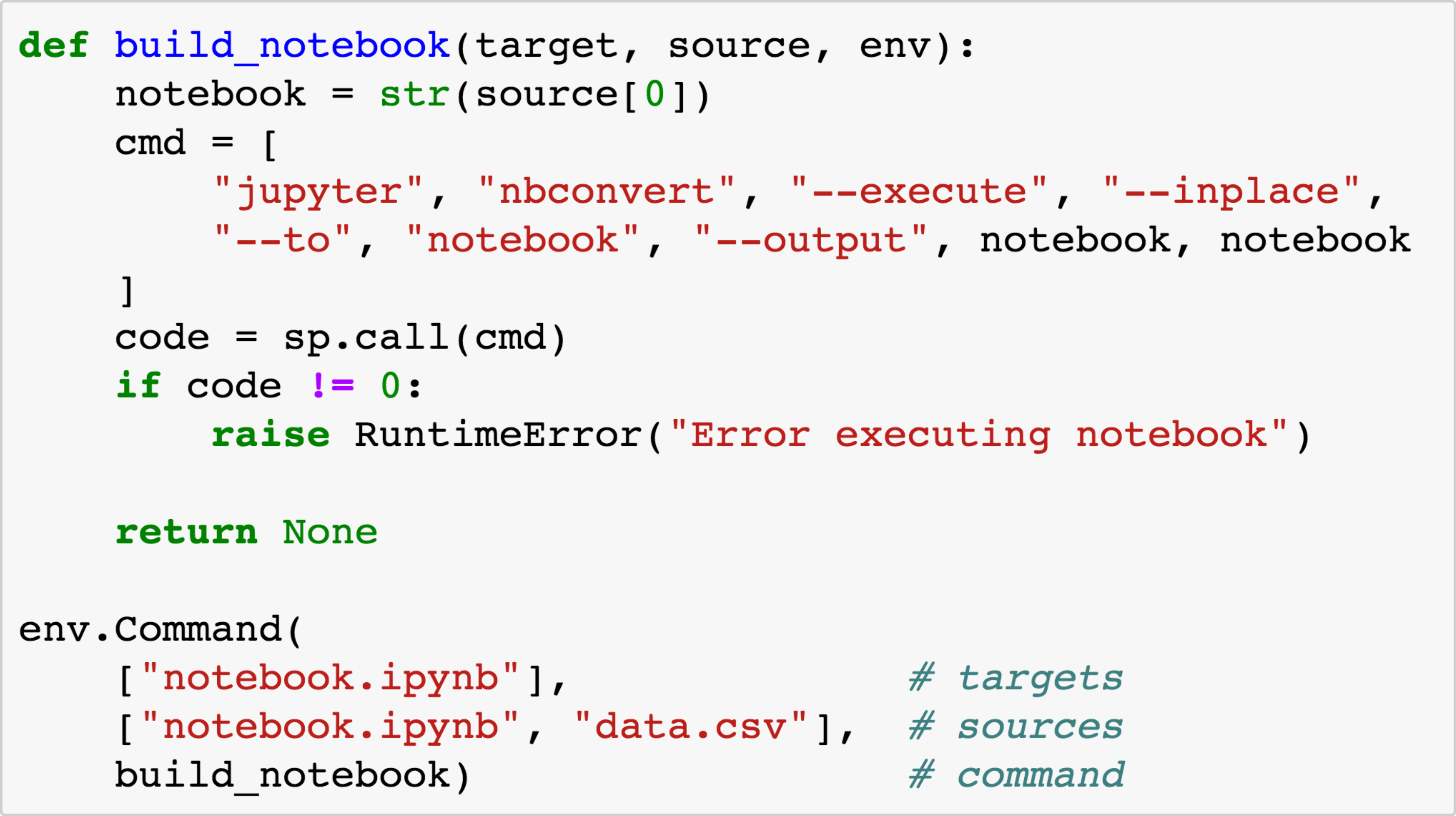
```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0));
```

```
angle     308.2
max_time  12
σ         10
β         2.6
ρ         28
```

Notebooks can be commands!

```
$ jupyter nbconvert \
  --execute \
  --to notebook \
  --output notebook.ipynb \
  notebook.ipynb
```

Notebooks can be **SCons** commands!



```
def build_notebook(target, source, env):
    notebook = str(source[0])
    cmd = [
        "jupyter", "nbconvert", "--execute", "--inplace",
        "--to", "notebook", "--output", notebook, notebook
    ]
    code = sp.call(cmd)
    if code != 0:
        raise RuntimeError("Error executing notebook")

    return None

env.Command(
    ["notebook.ipynb"],                       # targets
    ["notebook.ipynb", "data.csv"],           # sources
    build_notebook)                           # command
```

Approach 4: multiple scripts and / or notebooks with SCons

Drawback: keeping the SConstruct file up-to-date is a hassle ☹

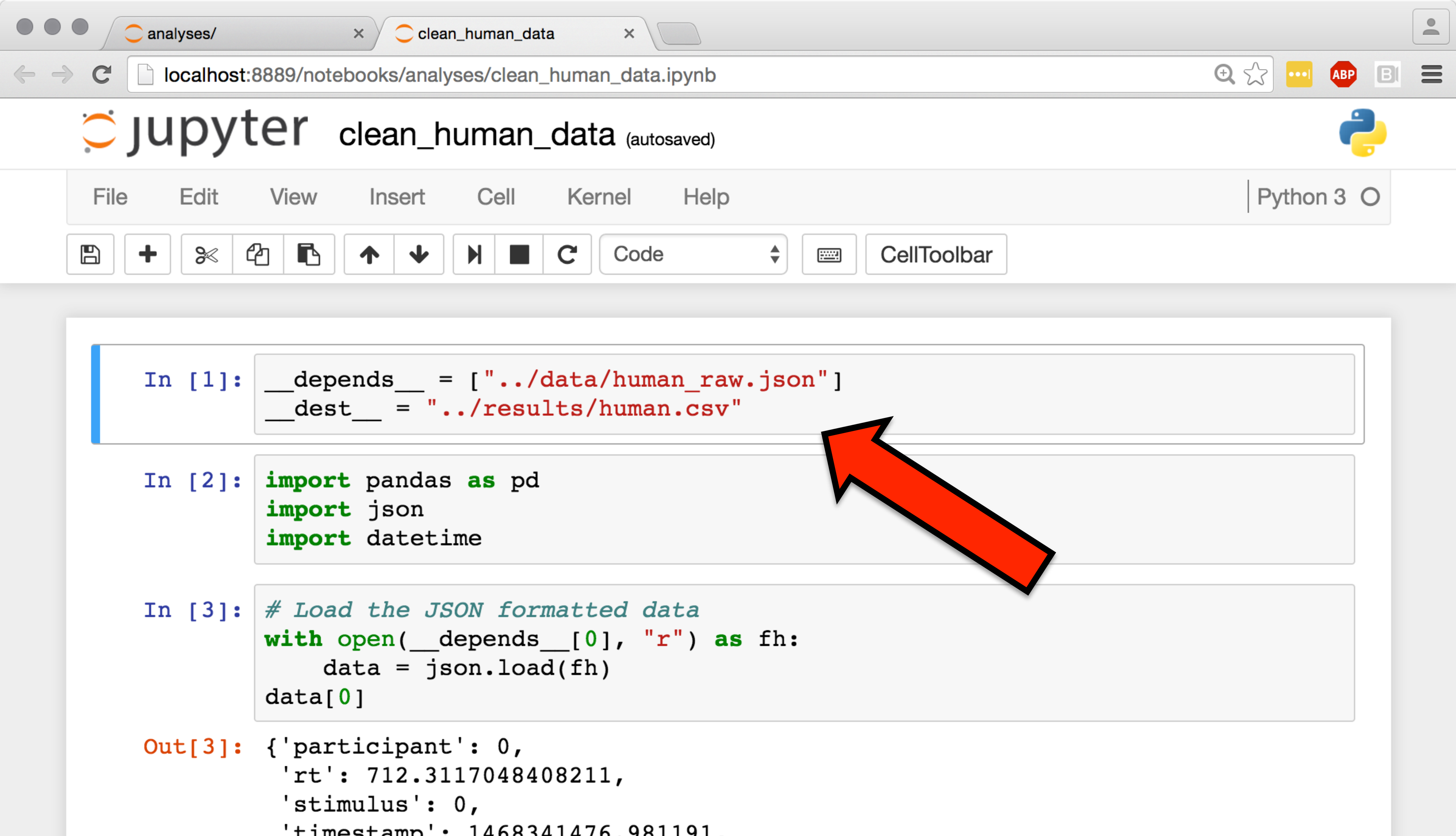
Approach 4: multiple scripts and / or notebooks with nbflow

Drawback: notebooks must be Python, requires Python 2 ☹ (but notebooks can be Python 3)

Installing nbflow

```
$ pip2 install \
  git+git://github.com/jhamrick/nbflow.git
```

Notebooks with nbflow



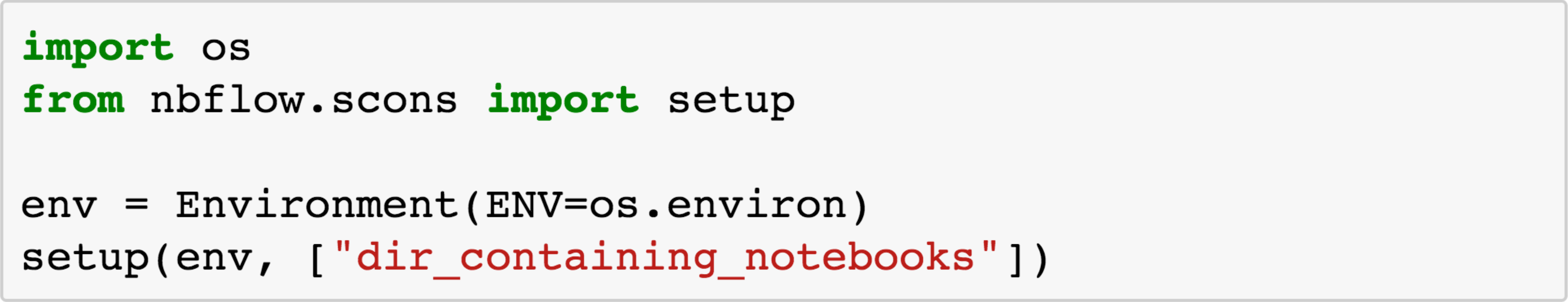
```
In [1]: __depends__ = ["../data/human_raw.json"]
        __dest__ = "../results/human.csv"
```

```
In [2]: import pandas as pd
        import json
        import datetime
```

```
In [3]: # Load the JSON formatted data
        with open(__depends__[0], "r") as fh:
            data = json.load(fh)
        data[0]
```

```
Out[3]: {'participant': 0,
         'rt': 712.3117048408211,
         'stimulus': 0,
         'timestamp': 1468341476.981191,
```

SConstruct file with nbflow

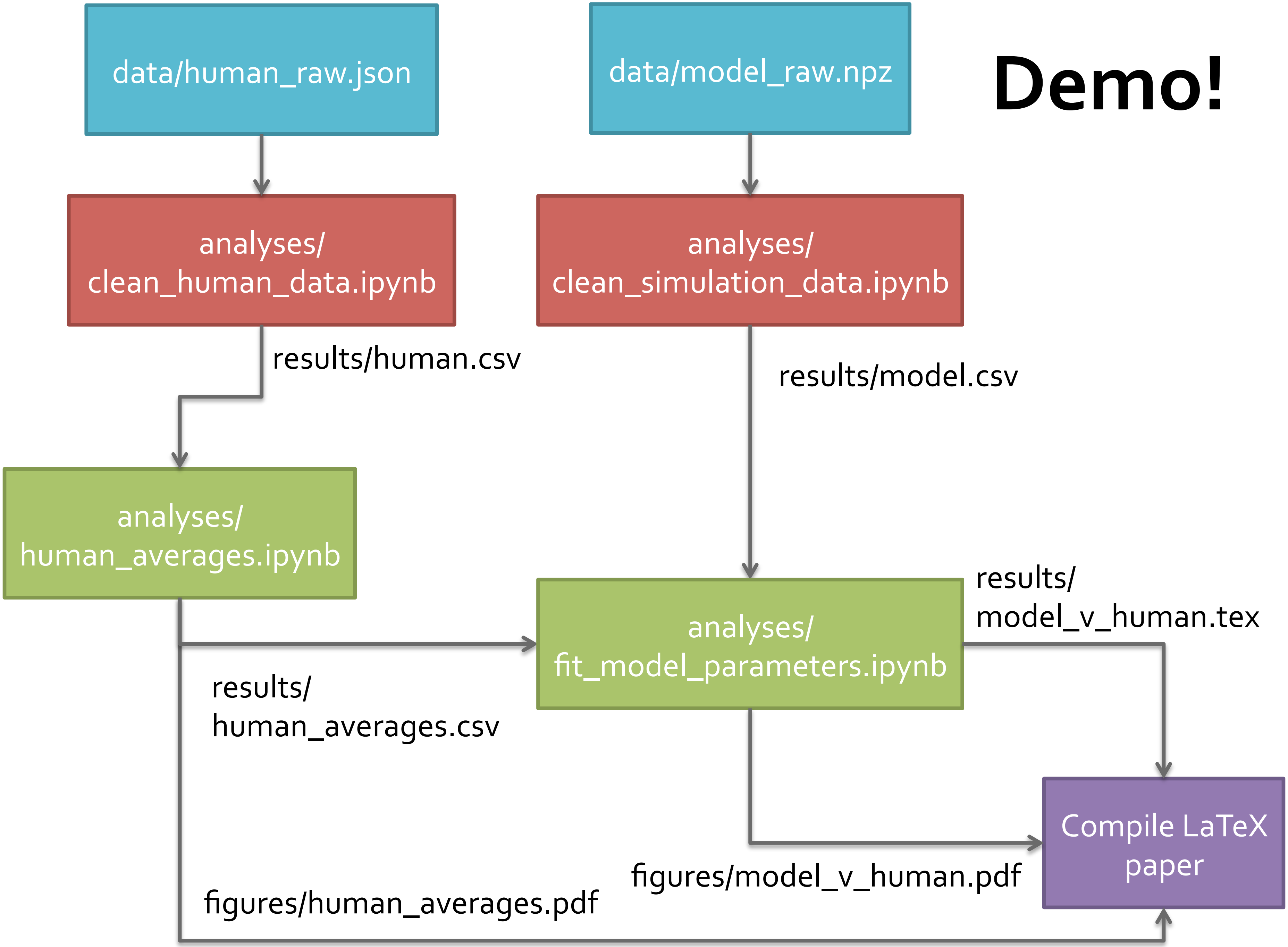


```
import os
from nbflow.scons import setup

env = Environment(ENV=os.environ)
setup(env, ["dir_containing_notebooks"])
```

```
$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
... a whole bunch of output ...
scons: done building targets.
```

Demo!



data/human_raw.json

data/model_raw.npz

analyses/
clean_human_data.ipynb

analyses/
clean_simulation_data.ipynb

results/human.csv

results/model.csv

analyses/
human_averages.ipynb

analyses/
fit_model_parameters.ipynb

results/
model_v_human.tex

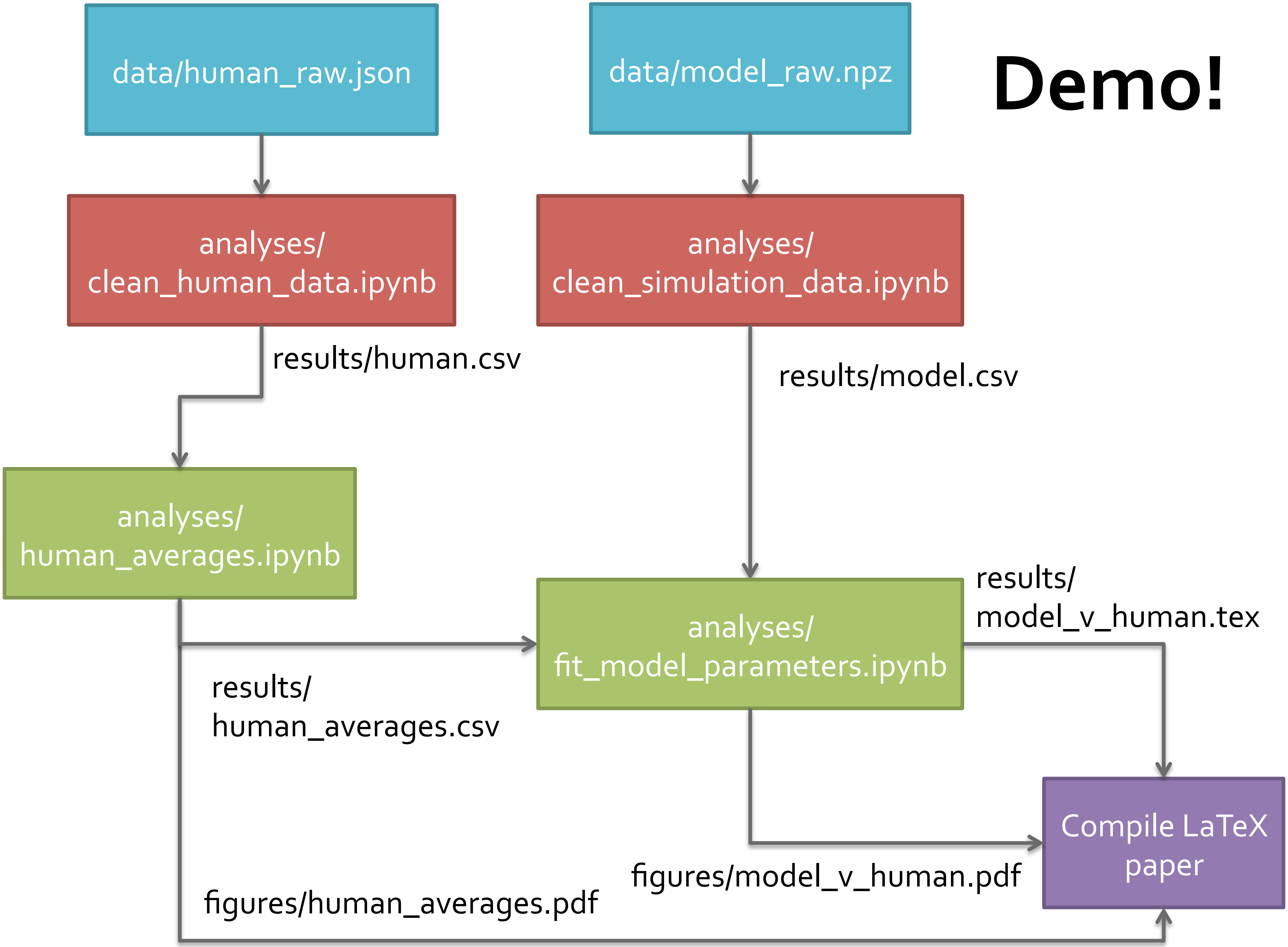
results/
human_averages.csv

Compile LaTeX
paper

figures/model_v_human.pdf

figures/human_averages.pdf

Demo!



data/human_raw.json

data/model_raw.npz

analyses/
clean_human_data.ipynb

analyses/
clean_simulation_data.ipynb

results/human.csv

results/model.csv

analyses/
human_averages.ipynb

analyses/
fit_model_parameters.ipynb

results/
model_v_human.tex

results/
human_averages.csv

Compile LaTeX
paper

figures/model_v_human.pdf

figures/human_averages.pdf

Future Improvements

1. Supporting non-Python kernels
2. Supporting other build systems
3. Python 3

http://tinyurl.com/nbflow-example