1

The Life-Changing Magic of Tidying Up

Ruby on Rails Application

```
.
├── Gemfile
├── Gemfile.lock
├── Guardfile
├── LICENSE
├── README.md
├── README.nitrous.md
├── Rakefile
├── app
│   ├── assets
│   ├── controllers
│   ├── helpers
│   ├── mailers
│   ├── models
│   └── views
├── bin
│   ├── bundle
│   ├── rails
│   └── rake
├── config
│   ├── application.rb
│   ├── boot.rb
│   ├── cucumber.yml
│   ├── database.yml.example
│   ├── environment.rb
│   ├── environments
│   ├── initializers
│   ├── locales
│   └── routes.rb
├── config.ru
├── db
│   ├── migrate
│   ├── schema.rb
│   └── seeds.rb
├── features
│   ├── signing_in.feature
│   ├── step_definitions
│   └── support
├── lib
│   ├── assets
│   └── tasks
├── log
├── public
│   ├── 404.html
│   ├── 422.html
│   ├── 500.html
│   ├── assets
│   ├── favicon.ico
│   └── robots.txt
├── script
│   └── cucumber
├── spec
│   ├── controllers
│   ├── factories.rb
│   ├── helpers
│   ├── models
│   ├── requests
│   ├── spec_helper.rb
│   └── support
└── vendor
    └── assets
```

Django Application

```
.
├── README.md
├── media
│   └── init.txt
├── projectname
│   ├── __init__.py
│   ├── home
│   │   ├── __init__.py
│   │   ├── models.py
│   │   ├── tests.py
│   │   └── views.py
│   ├── manage.py
│   ├── settings
│   │   ├── __init__.py
│   │   ├── default.py
│   │   └── local.template.py
│   ├── urls.py
│   └── wsgi.py
├── requirements.txt
├── static-assets
│   ├── apple-touch-icon.png
│   ├── css
│   │   └── main.css
│   ├── favicon.ico
│   ├── humans.txt
│   ├── images
│   │   └── init.txt
│   ├── js
│   │   ├── main.coffee
│   │   ├── main.js
│   │   └── main.map
│   ├── libs
│   │   ├── bootstrap-3.3.5
│   │   ├── font-awesome-4.3.0
│   │   ├── html5shiv.js
│   │   ├── jquery
│   │   └── modernizr
│   ├── media -> ../media/
│   └── robots.txt
└── templates
    ├── 404.html
    ├── 500.html
    ├── base.html
    └── home.html
```

```
bob@ubuntu:/$ ls -l /
total 81
drwxr-xr-x   2 root  root  4096 Mar 22 10:20 bin
drwxr-xr-x   3 root  root  4096 Mar 22 10:23 boot
drwxr-xr-x  15 root  root  4380 Apr 14 07:17 dev
drwxr-xr-x 131 root  root 12288 Apr 22 06:16 etc
drwxr-xr-x   4 root  root  4096 Apr 22 06:14 home
lrwxrwxrwx   1 root  root    34 Mar 22 09:47 initrd.img -> /boot/initrd.img-3.11.
0-15-generic
drwxr-xr-x  20 root  root  4096 Mar 22 10:21 lib
drwx------   2 root  root 16384 Mar 22 09:45 lost+found
drwxr-xr-x   4 root  root  4096 Mar 22 09:46 media
drwxrwxrwx   4 root  root  1024 Apr 14 07:25 mnt
drwxr-xr-x   2 root  root  4096 Mar 22 09:52 opt
dr-xr-xr-x 165 root  root     0 Apr 14 06:54 proc
drwx------   8 antun root  4096 Apr 10 02:23 root
drwxr-xr-x  21 root  root   760 Apr 23 01:01 run
drwxr-xr-x   2 root  root  4096 Apr 14 06:46 sbin
drwxr-xr-x   2 root  root  4096 Mar  5  2012 selinux
drwxr-xr-x   2 root  root  4096 Mar 22 09:45 srv
dr-xr-xr-x  13 root  root     0 Apr 14 06:54 sys
drwxrwxrwt  12 root  root  4096 Apr 23 06:25 tmp
drwxr-xr-x  10 root  root  4096 Mar 22 09:45 usr
drwxr-xr-x  13 root  root  4096 Apr 14 06:54 var
lrwxrwxrwx   1 root  root    30 Mar 22 09:47 vmlinuz -> boot/vmlinuz-3.11.0-15-ge
neric
bob@ubuntu:/$
```

Linux File System Hierarchy v2.0

/ "ROOT"

/BIN
"ESSENTIAL BINARIES"

- CAT
- CHGRP
- CHMOD
- CHOWN
- CP
- DATA
- DD
- DF
- DMESG
- ECHO
- FALSE
- HOSTNAME
- KILL
- LN
- LOGIN
- LS
- MKDIR
- MKNOD
- MORE
- MOUNT
- MV
- PS
- PWD
- RM
- RMDIF
- SED
- SH
- STTY
- SU
- SYNCH
- TRUE
- UMOUNT
- UNAME

/BOOT
"STATIC FILES OF BOOT LOADER"

- KERNEL
- SYSTEM.MAP
- VMLINUZ
- INITRD
- GRUB
- MODULEINFO
- BOOT

/ETC
"HOST SPECFIC SYSTEM CONFIG"

- CSH.LOGIN
- EXPORTS
- FSTAB
- FTPUSERS
- GATEWAYS
- GETTYDEFS
- GROUP
- HOST.CONF
- HOSTS
- HOSTS.ALLOW
- HOSTS.DENY
- HOSTS.EQUIV
- HOSTS.LPD
- INETD.CONF
- INITTAB
- ISSUE
- LS.SO.CONF
- MOTD
- MTAB
- MTOOLS
- NETWORKS
- PASSWD
- PRINTCAP
- PROFILE
- PROTOCOLS
- RESOLV.CONF
- RPC
- SECURETTY
- SERVICES
- SHELLS
- SYSLOG.CONF

/OPT
"CONFIG FILES FOR ADD ON APPLICATIONS"

/X11
"X WINDOW CONFIG FILES"

/USR
"MULTI-USER SHARE AND READ-ONLY DATA"

/LOCAL
"LOCAL SOFTWARES"

- /BIN
- /GAMES
- /INCLUDE
- /LIB
- /MAN
- /SBIN
- /SHARE
- /SRC

/SHARE
"ARCHITECTURE INDEPENDENT SHARED DATA"

/MAN
"MANUAL PAGES"

- /MAN1 "user programs"
- /MAN2 "system calls"
- /MAN3 "lib functions"
- /MAN4 "special file"
- /MAN5 "file formats"
- /MAN6 "games"
- /MAN7 "misc"
- /MAN8 "system admin"

/BIN
"MOST USER COMMANDS"

/INCLUDE
"STANDARD INCLUDE FILES FOR 'C' PROG."

/LIB
".OBJ .BIN .LIB FILES FOR PROG AND PACKAGES"

/SBIN
"NON ESSENTIAL BINARIES"

/VAR
"VARIABLE DATA FILES"

/CACHE
"APPLICATION CACHE DATA"

/LIB
"VARIABLE STATE INFORMATION REMAINS AFTER REBOOT"

/YP
"DATA FOR NIS SERVICES"

/LOCK
"LOCK FILES FOR SHARED RESOURCES"

/OPT
"VARIABLE DATA OF PACKAGES INSTALLED"

/RUN
"INFO OF SYSTEM SINCE IT WAS BOOTED"

/TMP
"AVAILABLE FOR PROG"

/SPOOL
"DATA AWAITING PROCESSING"

- /LPD
- /MQUEUE
- /NEWS
- /RWHO
- /UUCP
- /MAIL

/LOG
"LOG FILES"

- LASTLOG
- MESSAGES
- WTMP

/SBIN
"SYSTEM BINARIES"

- FASTBOOT
- FASTHALT
- FDISK
- FSCK
- GETTY
- HALT
- IFCONFIG
- INIT
- MKFS
- MKSWAP
- REBOOT
- ROUTE
- SWAPON
- SWAPOFF
- UPDATE

/TMP
"TEMPORARY FILES DELETED ON BOOTUP"

/DEV
"ESSENTIAL DEVICES (CONTAINS MAKEDEV)"

/HOME
"USER HOME DIRECTORIES"

/LIB
"LIBRARY AND KERNEL MODULES"

/MNT
"TEMPERORY FILESYSTEM MOUNTPOINT"

/OPT
"ADD-ON APPLICATION SOFTWARE"

/ROOT
"HOME DIR FOR ROOT USER"

/SRV
"SERVER DATA"

/MEDIA
"CD/DVD MOUNT POINT"

/PROC
"PROCESS FILESYSTEM"

Version 2.0 - 17-06-2015
- Improved: Added title and version history.
- Improved: Added /srv, /media and /proc.
- Improved: Updated descriptions to reflect modern Linux File Systems.
- Fixed: Multiple typo's.
- Fixed: Appearance and colour.

Version 1.0 - 14-02-2015
- Created: Initial diagram.
- Note: Discarded lowercase version.

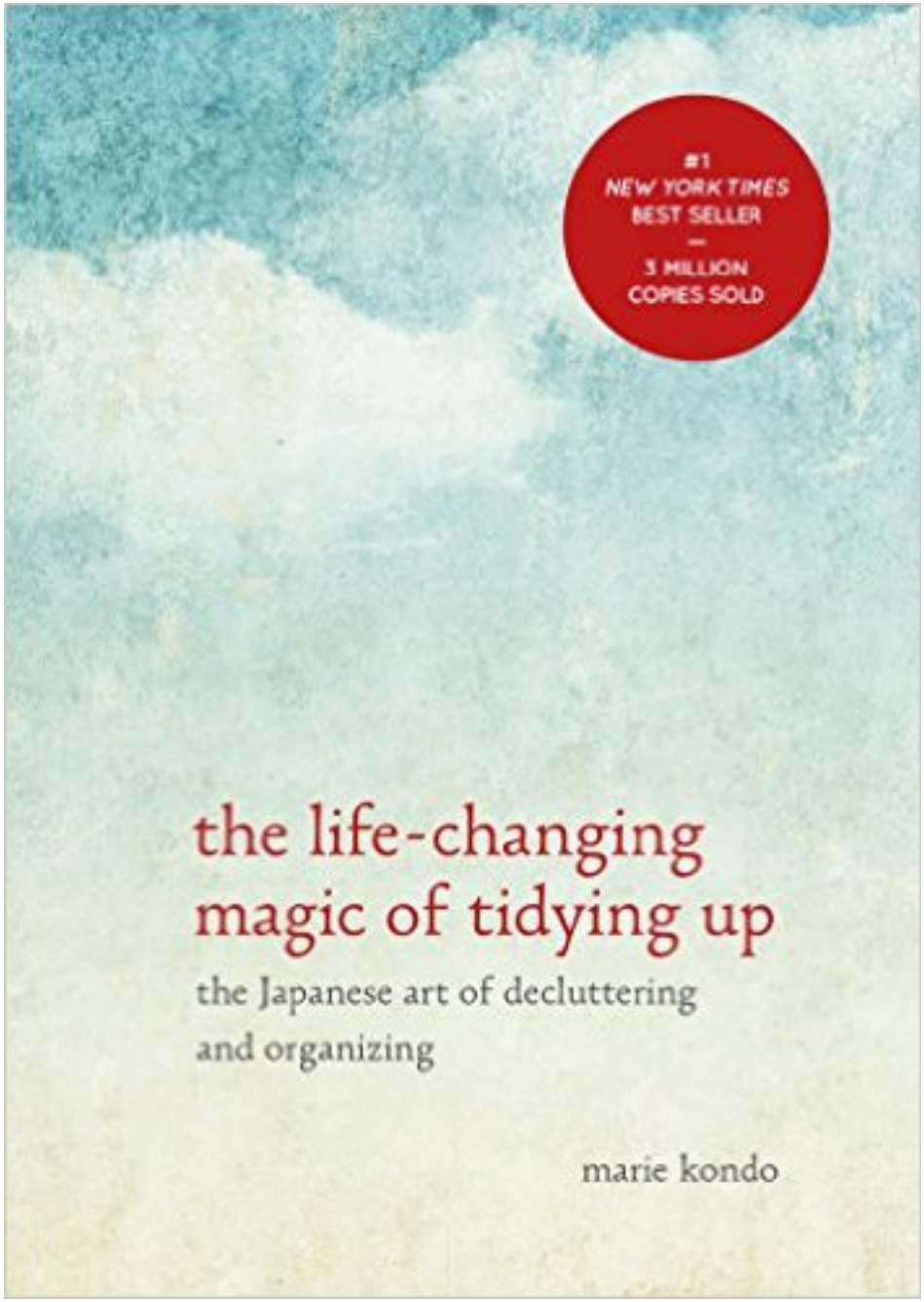
Created by: http://www.blackmoreops.com/

Updated on: June 17, 2015

```
.
├── Inspection_count_min.jpeg
├── README.Rmd
├── README.html
├── dd_dictionary.csv
├── mallet.rar
├── scripts\ and\ data
|   ├── AllViolations.csv
|   ├── PhaseIISubmissionFormat.csv
|   ├── build_rev_tm.R
|   ├── docsAsTopicsProbs_noStopwords.txt
|   ├── feature_eng.R
|   ├── features_test_phase2.csv
|   ├── features_train_phase2.csv
|   ├── learning_final.R
|   ├── negative-words.txt
|   ├── positive-words.txt
|   ├── rand_neg.txt
|   ├── restaurant_ids_to_yelp_ids.csv
|   ├── rev_tm.txt
|   ├── review_sentiscored.csv
|   ├── run.R
|   ├── sentiment_script.R
|   ├── sub_2_PhaseII_h20.csv
|   ├── yelp.stops
|   └── yelp_academic_dataset_business.json
├── varimp_gbm1.jpeg
├── varimp_gbm2.jpeg
└── varimp_sev.jpeg
```

```
.
├── AllViolations.csv
├── BusinessClass.py
├── GenLearningData.py
├── GenTestingData.py
├── InspectionClass.py
├── LearnTest.py
├── PhaseIISubmissionFormat.csv
├── PhaseIISubmissionFormat_final.csv
├── PhaseIISubmissionFormat_test.csv
├── README.txt
├── ReviewClass.py
├── restaurant_ids_to_yelp_ids.csv
├── yelp_boston_academic_dataset
└── yelp_duplicate_ids.csv
```

```
.
├── Step\ 1\ -\ install\ necessary\ software\ and\ packages.txt
├── Step\ 2\ -\ one-off\ step\ to\ create\ postgresql\ server\ instance\ and\ a\ database.txt
├── Step\ 3\ -\ one-off\ step\ to\ create\ tables\ and\ views\ in\ postgresql.py
└── Step\ 4\ -\ The\ only\ file\ to\ run\ when\ you\ want\ to\ run\ models\ and\ generate\ new\ scores.py
```


#1
NEW YORK TIMES
BEST SELLER
—
3 MILLION
COPIES SOLD
the life-changing
magic of tidying up
the Japanese art of decluttering
and organizing
marie kondo

Cookiecutter Data Science
Why use this project structure?
Other people will thank you
You will thank you
Nothing here is binding
Getting started
Requirements
Starting a new project
Example
Directory structure
Opinions
Data is immutable
Notebooks are for exploration and communication
Analysis is a DAG
Build from the environment up
Keep secrets and configuration out of version control
Be conservative in changing the default folder structure
Contributing
Links to related projects and references

Cookiecutter Data Science

A logical, reasonably standardized, but flexible project structure for doing and sharing data science work.

Why use this project structure?

> We're not talking about bikeshedding the indentation aesthetics or pedantic formatting standards — ultimately, data science code quality is about correctness and reproducibility.

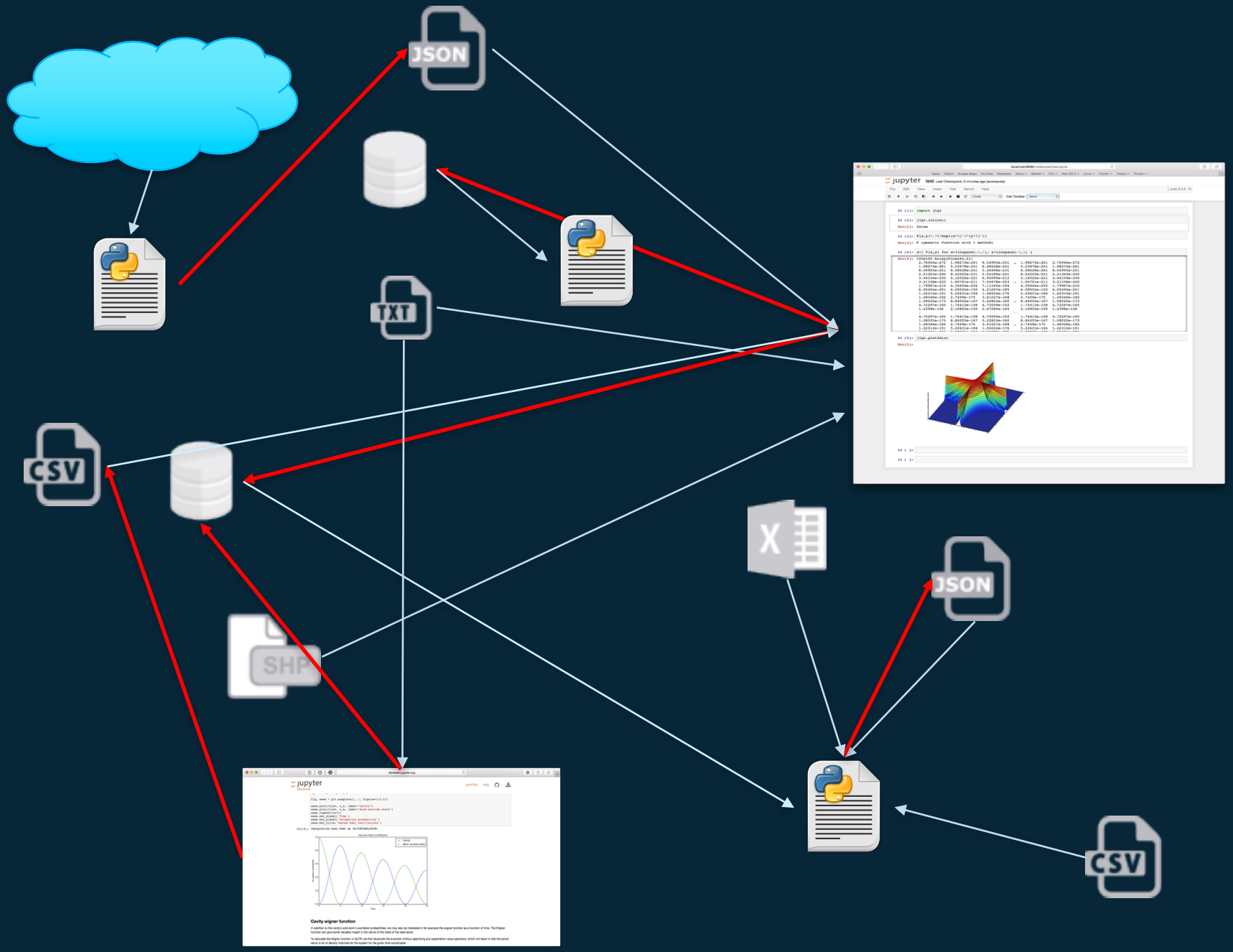
When we think about data analysis, we often think just about the resulting reports, insights, or visualizations. While these end products are generally the main event, it's easy to focus on making the products *look nice* and ignore the *quality of the code that generates them*. Because these end products are created programmatically, **code quality is still important**! And we're not talking about bikeshedding the indentation aesthetics or pedantic formatting standards — ultimately, data science code quality is about correctness and reproducibility.

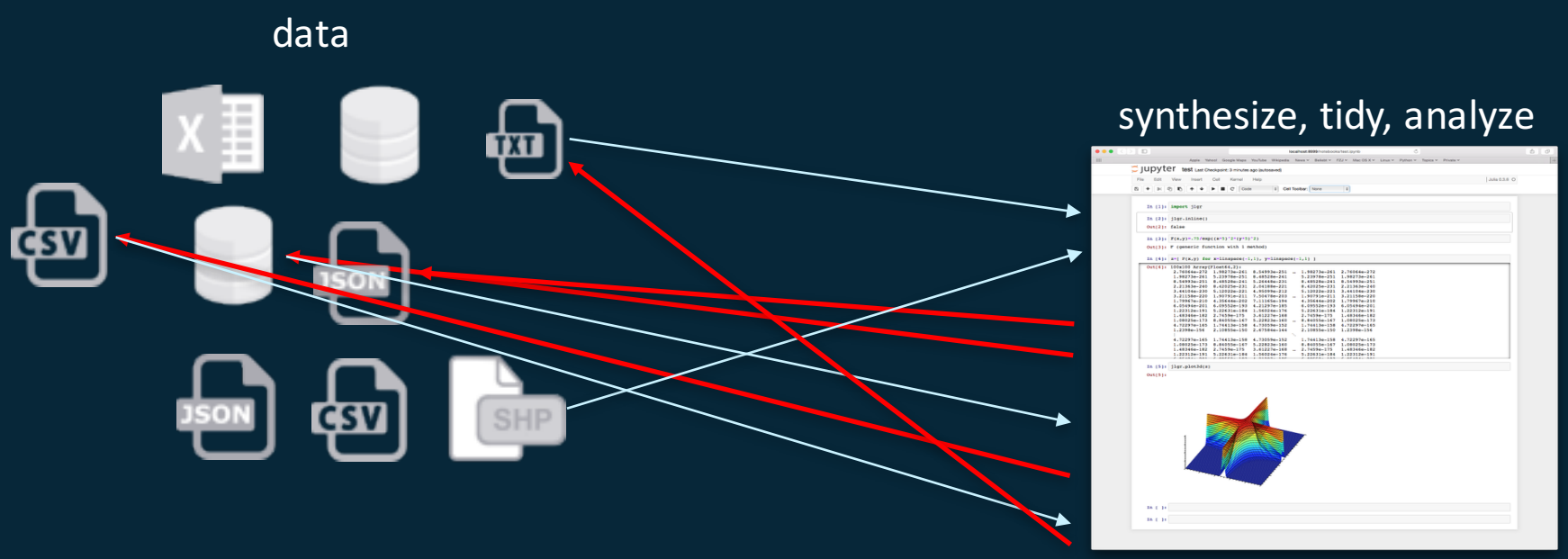
It's no secret that good analyses are often the result of very scattershot and serendipitous explorations. Tentative experiments and rapidly testing approaches that might not work out are all part of the process for getting to the good stuff, and there is no magic bullet to turn data exploration into a simple, linear progression.

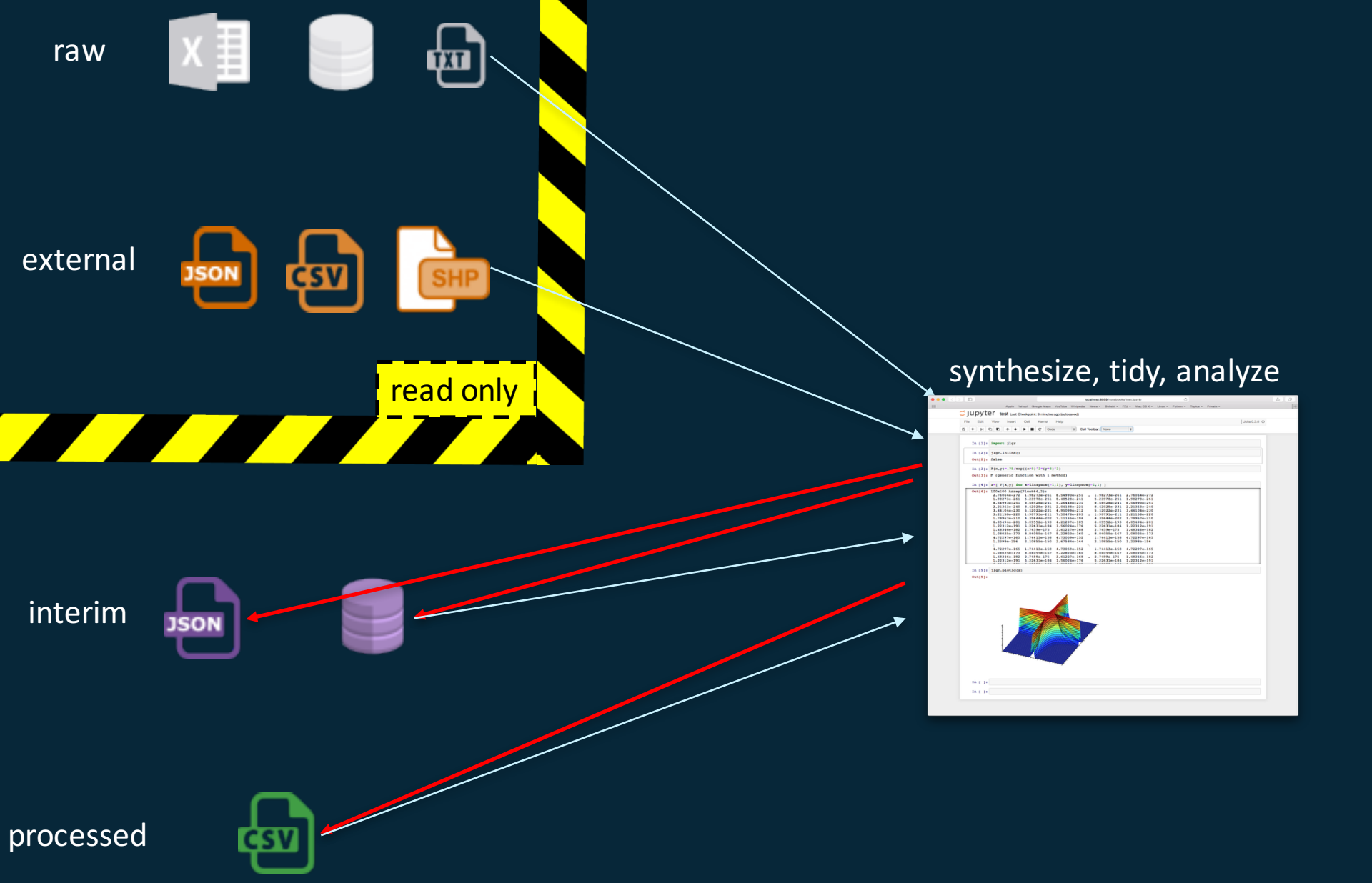
That being said, once started it is not a process that lends itself to thinking carefully about the structure of your code or project layout, so it's best to start with a clean, logical structure and stick to it throughout. We think it's a pretty big win all around to use a fairly standardized setup like this one. Here's why:

drivendata.github.io/cookiecutter-data-science

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.testrun.org
```


JSON
TXT
CSV
SHP
X
JSON
CSV
jupyter
test
Cavity wigner function


data
X
TXT
CSV
JSON
JSON
CSV
SHP
synthesize, tidy, analyze
jupyter


raw
X
TXT
external
JSON
CSV
SHP
read only
synthesize, tidy, analyze
jupyter
interim
JSON
processed
CSV

raw

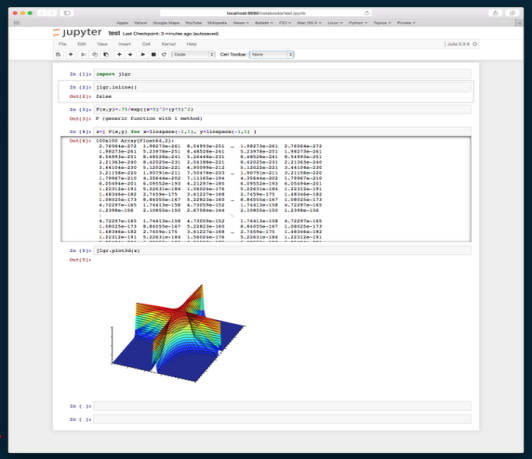
external

read only

interim

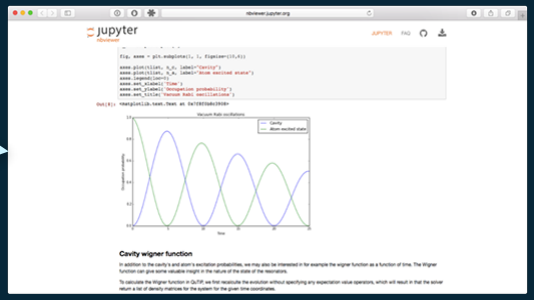
processed

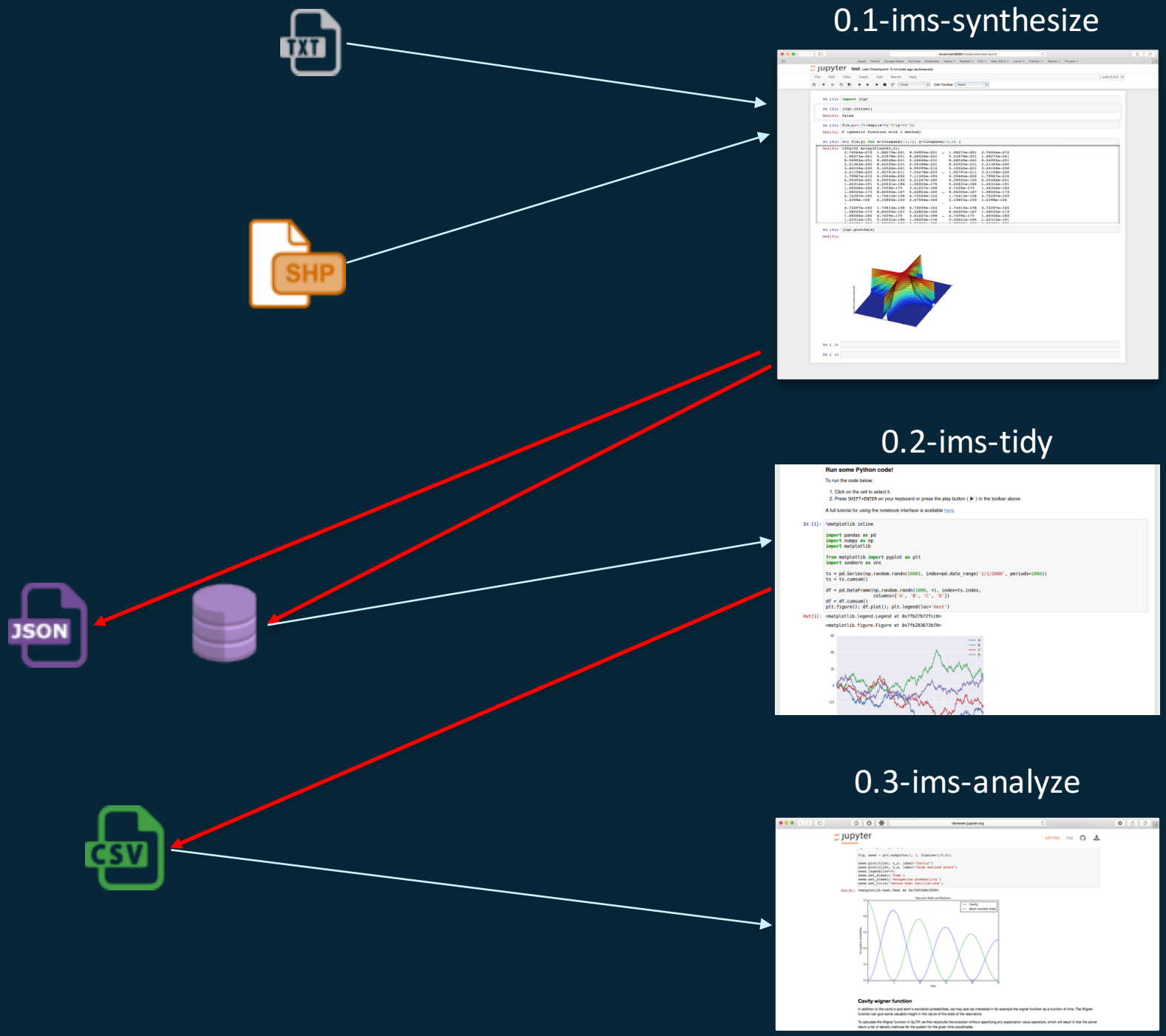
0.1-ims-synthesize

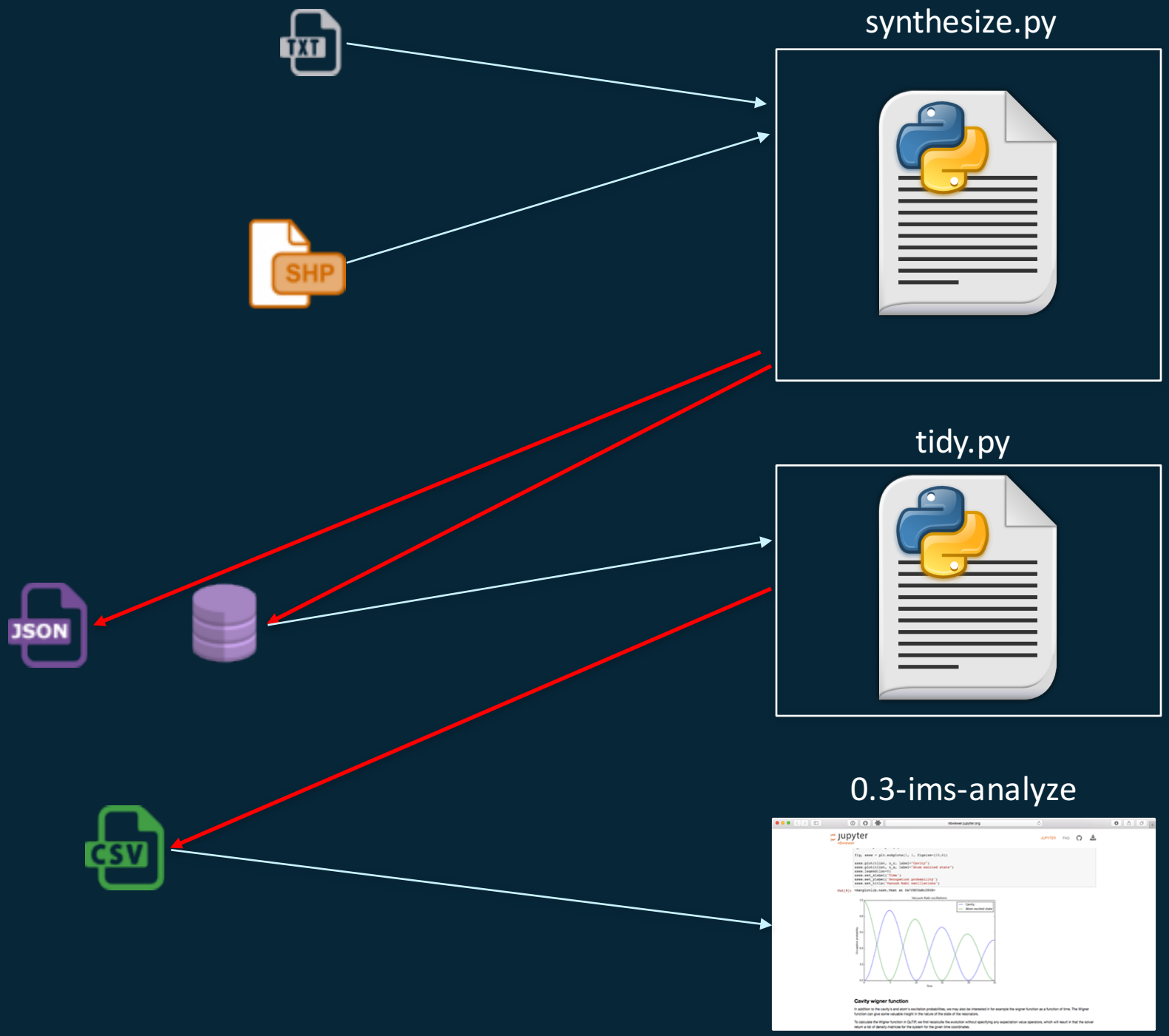


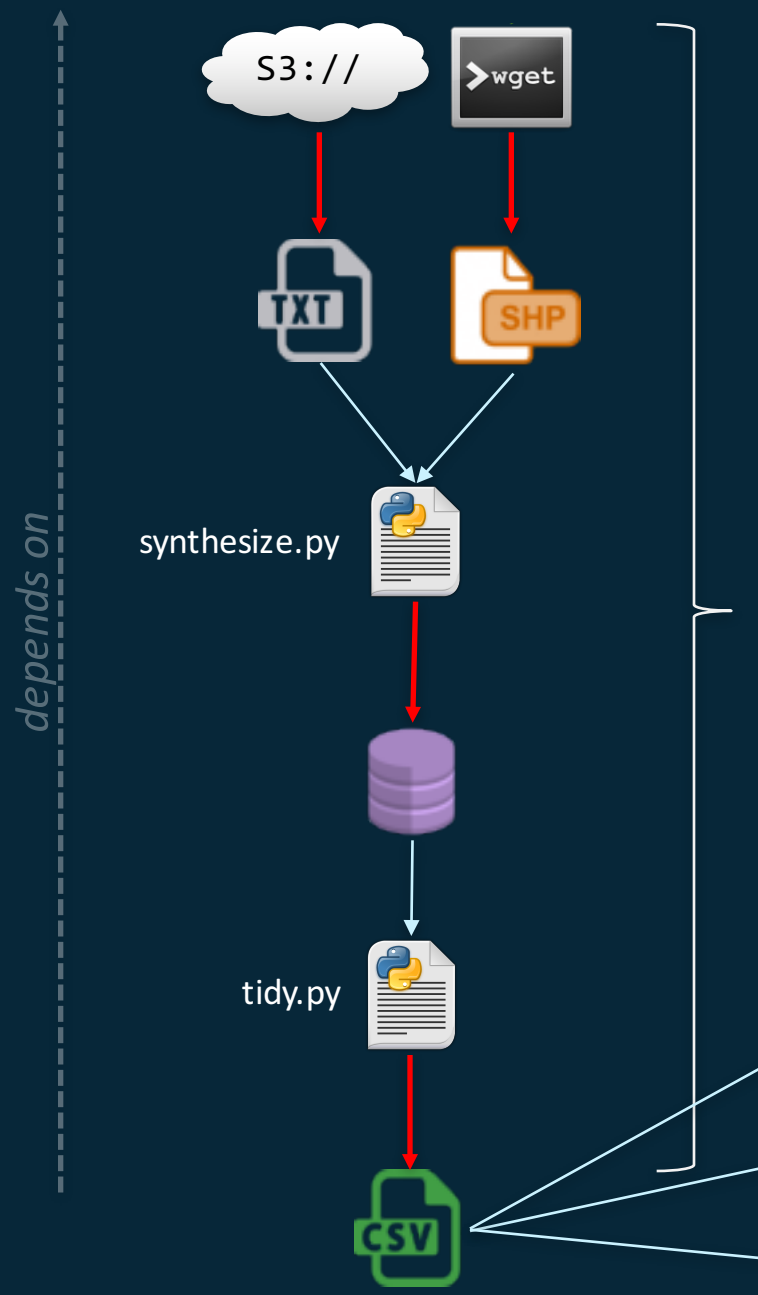
0.2-ims-tidy

0.3-ims-analyze




0.1-ims-synthesize
TXT
SHP
0.2-ims-tidy
Run some Python code!
JSON
0.3-ims-analyze
CSV


synthesize.py
TXT
SHP
tidy.py
JSON
CSV
0.3-ims-analyze
jupyter



Makefile

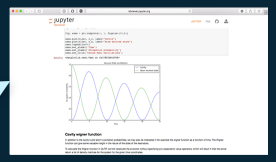
```
data/external/geo.shp:
      wget -O $@ http://.../blah.shp && \
      sync_data_from_s3

data/interim/joined.db: data/external/geo.shp \
                        data/raw/records.txt
      python synthesize.py

data/processed/data.csv: data/interim/joined.db
      python tidy.py
```

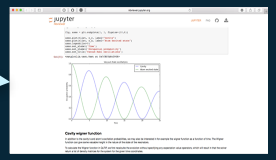
```
$ make data/processed/data.csv
```

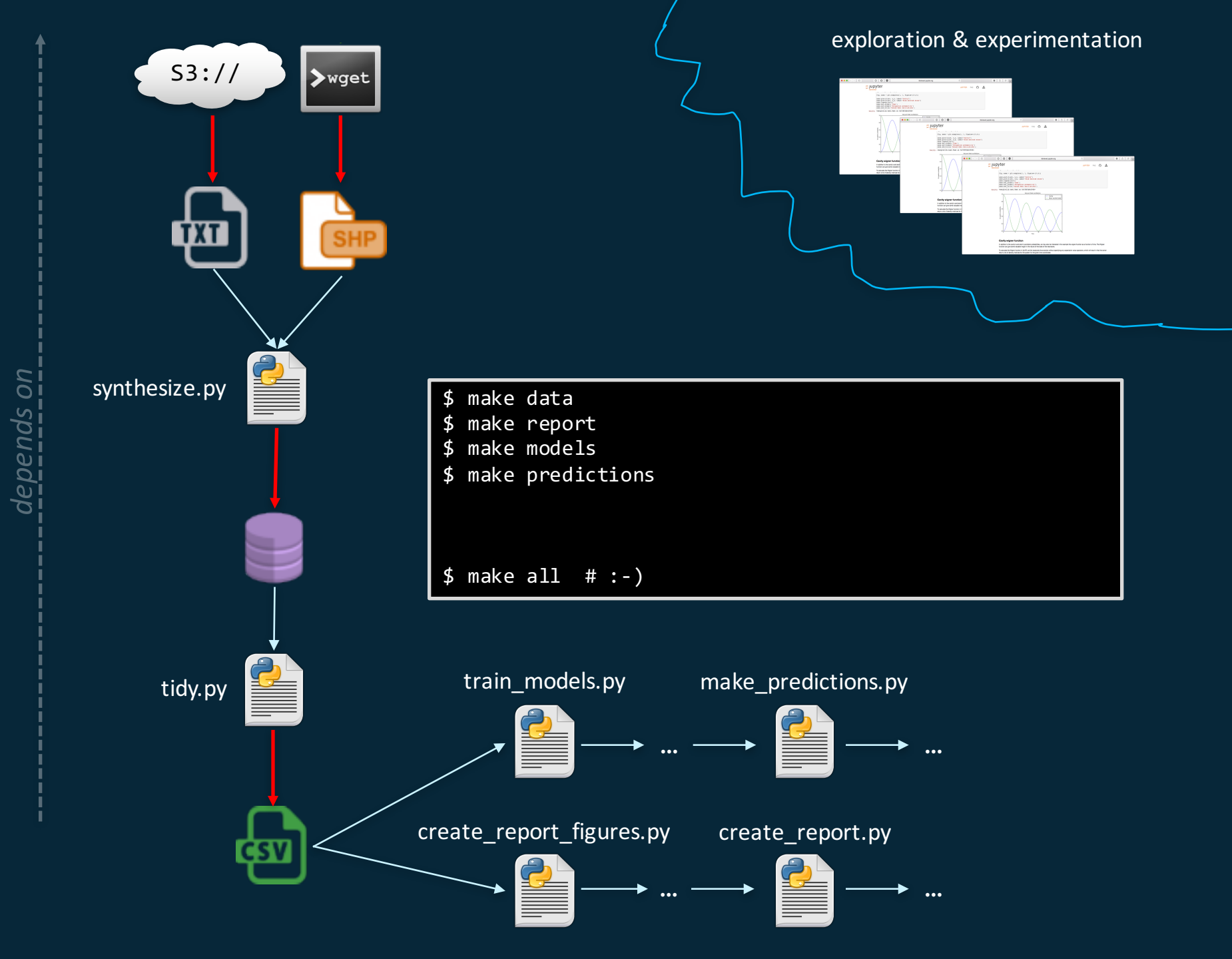
0.3-ims-analyze



0.4-pjb-train-models

0.5-ims-create-report-figures




exploration & experimentation
depends on
S3://
wget
TXT
SHP
synthesize.py
tidy.py
CSV
train_models.py
make_predictions.py
create_report_figures.py
create_report.py
...
$ make data
$ make report
$ make models
$ make predictions
$ make all # :-)

Lab 1

- http://drivendata.github.io/cookiecutter-data-science/
- Install cookiecutter
- Clone the talk repo into `data-science-is-software`
- Use `cookiecutter-data-science` to create a new project folder (we'll call it `water-pumps`)
- Delete the following: `LICENSE`, `README.md`, `references/`, `reports/`, and `docs/`
- Mirror the following from the talk repo (`data-science-is-software`) to your project repo (`data-project`):
  - `requirements.txt`
  - `data/*`
  - `src/*`
  - `notebooks/*`

```
├── data
│   └── raw
│       ├── pumps_train_labels.csv
│       └── pumps_train_values.csv
├── Makefile
├── models
├── notebooks
│   ├── labs
│   │   ├── 2.0-environment-lab.ipynb
│   │   ├── 2.0-environment-solution.ipynb
│   │   ├── 3.0-refactoring-lab.ipynb
│   │   ├── 3.0-refactoring-solution.ipynb
│   │   └── 4.0-testing-lab.ipynb
│   └── lectures
│       ├── 2.0-environment.ipynb
│       ├── 3.0-refactoring.ipynb
│       ├── 4.0-testing.ipynb
│       ├── 5.0-debugging-code-quality.ipynb
│       └── 6.0-collaboration.ipynb
├── requirements.txt
└── src
    ├── features
    │   ├── build_features.py
    │   └── preprocess_solution.py
    ├── __init__.py
    ├── mcmc
    │   └── hamiltonian.py
    ├── model
    │   └── train_model_solution.py
    ├── tests
    │   ├── __init__.py
    │   ├── test_example.py
    │   └── test_lab4_solutions.py
    └── utils.py
```

Lab 1 – solution

```
$ cookiecutter https://github.com/drivendata/cookiecutter-data-science.git
    project_name [project_name]: water-pumps
    …
$ cd water-pumps
$ rm -rf LICENSE README.md references/ docs/
$ cp ../data-science-is-software/requirements.txt ./
$ cp -R ../data-science-is-software/data/* data/
$ cp -R ../data-science-is-software/src/* src/
$ cp -R ../data-science-is-software/notebooks/* notebooks/
```

DRIVENDATA