Lab 1: Environment Setup

Prepared & Presented by Will Claybaugh

CS109A Introduction to Data Science

Pavlos Protopapas and Mark Glickman

Warmup

Windows:

- Open anaconda prompt
- Type `conda -V`
- **If you get an error**, install Anaconda: https://docs.anaconda.com/anaconda/install/windows/
  - #8 is important: **DO NOT** add to your path
- **If no error**, consider upgrading conda:

  `conda update conda`

- Clone https://github.com/Harvard-IACS/2019-CS109B
  (or pull the latest if you've already cloned)

Mac:

- Open a terminal
- Type `conda -V`
- **If you get an error**, install Anaconda: https://docs.anaconda.com/anaconda/install/mac-os/
- **If no error**, consider upgrading conda:

  `conda update conda`

- Clone https://github.com/Harvard-IACS/2019-CS109B
  (or pull the latest if you've already cloned)

Goals (Who this lab is for)

- Set up the tools you'll need for CS109b
  - In a way that won't mess up your other classes
- Teach a workflow that will *keep* your installs tidy
- User-level understanding of why 'environments' are helpful
- *Stretch*: Ability to produce conda environments for future projects

- **TL;DR**: Set up a conda environment with the packages listed in 109b.yml
  - If you already know how to do that, you can skip the lab

Jumpstart

```
(base) C:\Users\Will>conda env create -f C:\Users\Will\Desktop\2019-CS109B-private\content\labs\lab0\109b.yml
Collecting package metadata: done
Solving environment: done
Preparing transaction: done
Verifying transaction: done
Executing transaction: | DEBUG menuinst_win32:__init__(199): Menu: name: 'Anaconda${PY_VER} ${PLATFORM}', prefix: 'C:\Us
ers\Will\Anaconda3\envs\109b', env_name: '109b', mode: 'None', used_mode: 'user'
DEBUG menuinst_win32:create(323): Shortcut cmd is %windir%\System32\cmd.exe, args are ['"/K"', 'C:\\Users\\Will\\Anacond
a3\\Scripts\\activate.bat', 'C:\\Users\\Will\\Anaconda3\\envs\\109b']
\
```

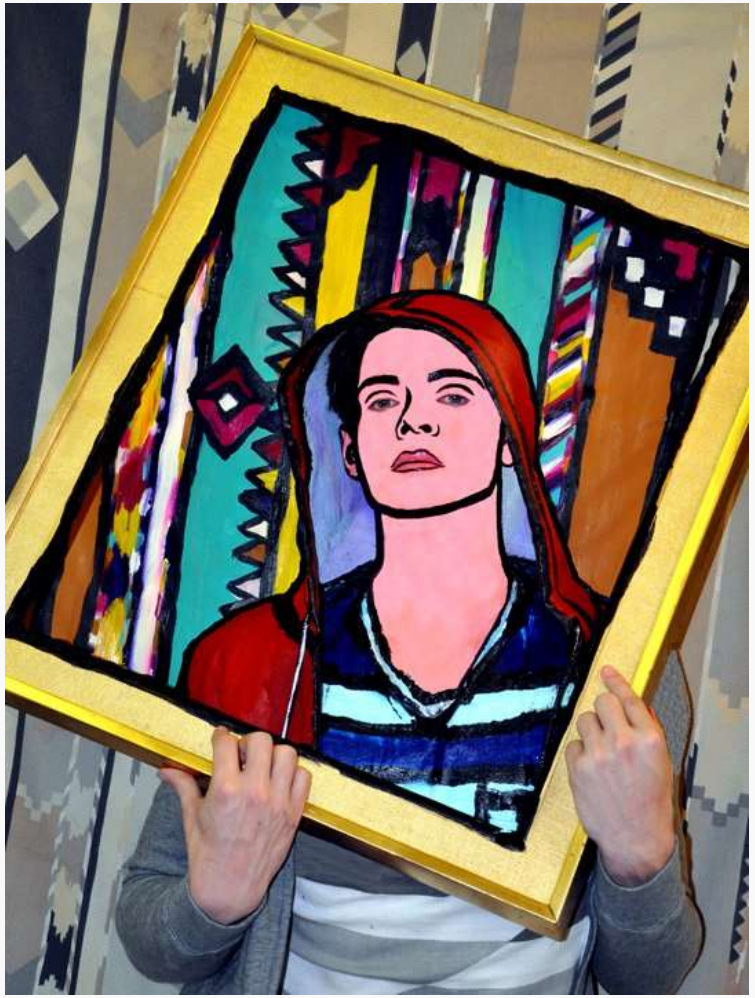
1. Locate the file 2019-cs109b/content/labs/lab1/109b.yml
2. Run `conda env create -f` **`[path]`**`/109b.yml`
   - **Windows:** use \ instead of /, delete the "- pyjags" line from the file
   - pyjags has no plans to support windows : (

- Setup may take a few minutes
- While we wait: Introductions + Norms

Me

- For a scavenger hunt, teamed with college friends to write an end-rhyme rapping Markov Chain
  - M.C. MCMC
  - Later released mix[ing] tape “d/dt: Derivative with respect to rhyme”

- Taught AP Calc; finally understood abstract algebra via tutoring a former student over the phone



Norms

But it's not about me; it's about you

- Most time will be yours to work on exercises
- TFs in the room and on Zoom to answer questions
- You might finish the exercise easily, or you might get stuck
  - Either way, please be patient
  - We'll (quickly?) go over the solutions after each exercise

- Now, what was that code *doing*?

[ANA]CONDA

Python, Anaconda, and Conda, oh my!

- We're creating a separate set of Python language files and packages for cs109
  - Installs/updates for other classes won't break cs109
  - cs109 won't break other classes
  - Can use different versions of Python (we're using 3.6, even though 3.7 is newly released)

- **CONDA** is the tool that manages these *environments*
  - *Anaconda* is the name for a useful set of [data] science packages, including conda itself

The Circle of Life

Environment workflow

Create (once): `conda env create -f [path]`

- Turn on an environment
  - Windows: `conda activate [envname]`
  - Mac: `source activate [envname]`
- Use the environment (write/save code, upgrade/install packages)
- Switch back to the global environment, named (base):
  - Windows: `conda deactivate`
  - Mac: `source deactivate`

Destroy (once): `conda remove --name [envname] --all`

Python, Anaconda, and Conda, oh my!

FAQs

- Can still access all existing files, no matter what environment you activate
- Conda guarantees you get the correct versions of each package
- Can (and should!) have lots of environments; they share what they can safely share and don’t take up much space
- Can install new things to an environment or just burn it down and build a new one

Exercise

Exercise:

1. In the 109b environment, install autodiff_group3 from pip. Verify that you can't `import autodiff` in your base environment
   - Notes on combining pip and conda: [here]
   - TL;DR: conda's update doesn't always know about things installed via pip; try to do all conda things first, then all pip things
2. Also in the 109b environment, open the r_setup.ipynb notebook and run the cells. This will:
   1. Verify the installed packages (especially Keras) will load
   2. Download and some packages in the R language we'll call on later in the course

Solutions

Solutions:

1.
```
(base) C:\Users\Will>conda activate 109b
```

```
(109b) C:\Users\Will>pip install autodiff_group3
Collecting autodiff_group3
```

```
(109b) C:\Users\Will>conda deactivate
```

```
(base) C:\Users\Will>python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import autodiff
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'autodiff'
>>>
```

Solutions

Solutions:

2.
```
(base) C:\Users\Will>conda activate 109b
```
```
(109b) C:\Users\Will>jupyter notebook
```

Use notebook as usual

REVIEW

Review

- Environments keep different package/language versions separate
- Ideally: create an environment for each class or project
- Minimally: do all 109b work in the 109b environment
  - Remember how?

Create (once): `conda env create -f [path]`

Turn on an environment

Windows: `conda activate [envname]`

Mac: `source activate [envname]`

Use the environment (write/save code, upgrade/install packages)

Switch back to the global environment, named (base):

Windows: `conda deactivate`

Mac: `source deactivate`

Destroy (once): `conda remove --name [envname] -all`

- Environments can also be managed via the Anaconda Navigator

JUPYTERHUB

JupyterHub

Poll: How many people used JupyterHub for 109a?

JupyterHub:

- We're paying Amazon to use their CPUs/GPUs/RAM/Disk
- Useful lie: think of it as a (powerful) remote computer
- No GUI operating system installed; some tasks must be done on command line
- Turns off after 1h of idle time
  - WILL NOT shut down while code is running
  - WILL shut down without saving your results! You'll have to re-run the notebook

- <u>Cannot complete your projects without it!!</u>

Exercise

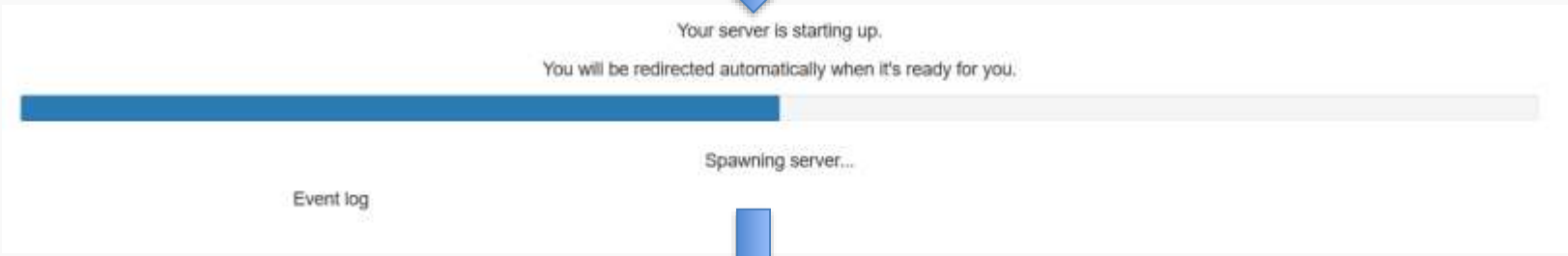
Exercise:

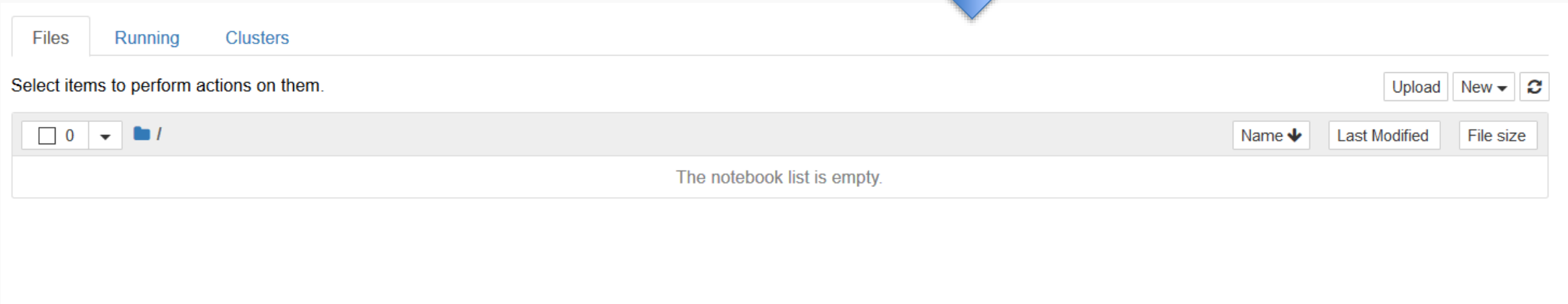
1. Log in to JupyterHub via the 109b Canvas page
   - If you see the familiar Jupyter Home, you succeeded.
2. Upload the r_setup.ipynb notebook
3. Run the notebook to download the courses' R packages
4. Download a copy of the updated notebook via File->Download as

Solutions

Solutions

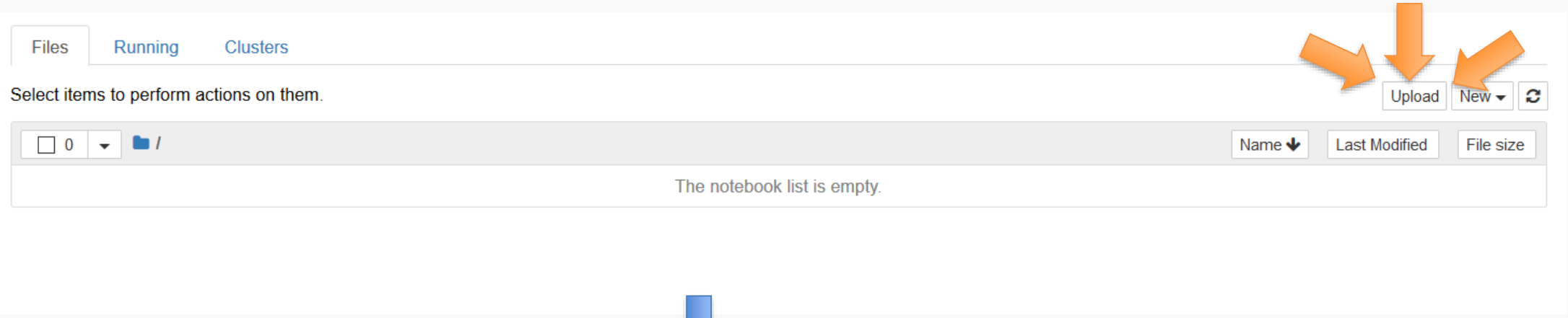
1.

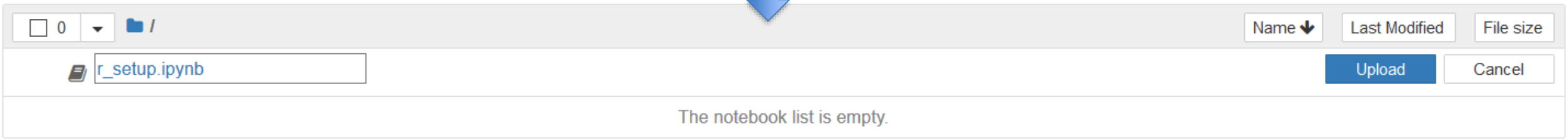




Solutions

2.





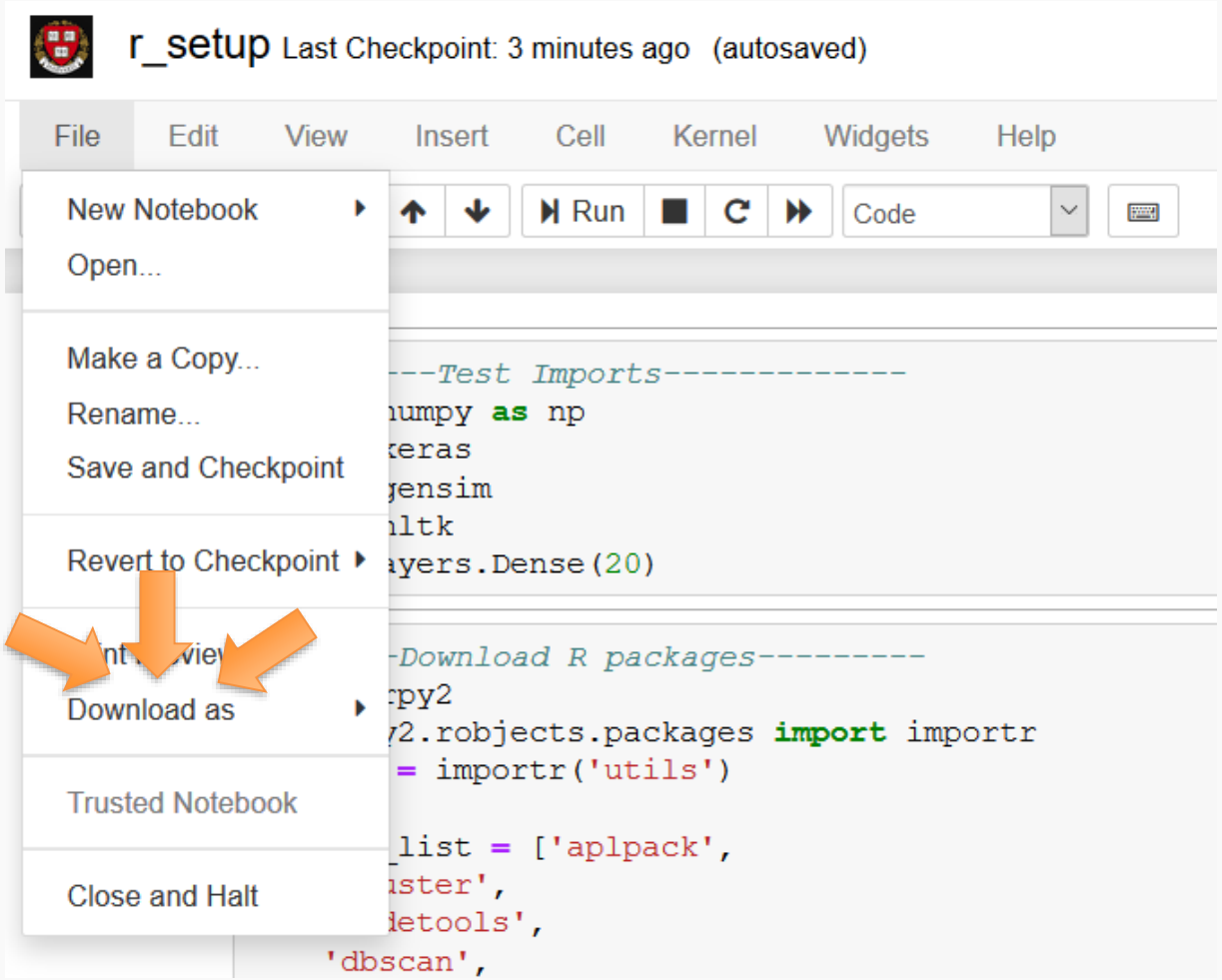
Use notebook as usual

Solutions

3. Trivial- Run the notebook as you normally would

Solutions

4.


r_setup Last Checkpoint: 3 minutes ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
New Notebook
Open...
Make a Copy...
Rename...
Save and Checkpoint
Revert to Checkpoint
Download as
Trusted Notebook
Close and Halt
Run
Code
---Test Imports----------
umpy as np
eras
ensim
ltk
yers.Dense(20)
-Download R packages---------
py2
2.robjects.packages import importr
= importr('utils')
list = ['aplpack',
ster',
etools',
'dbscan',

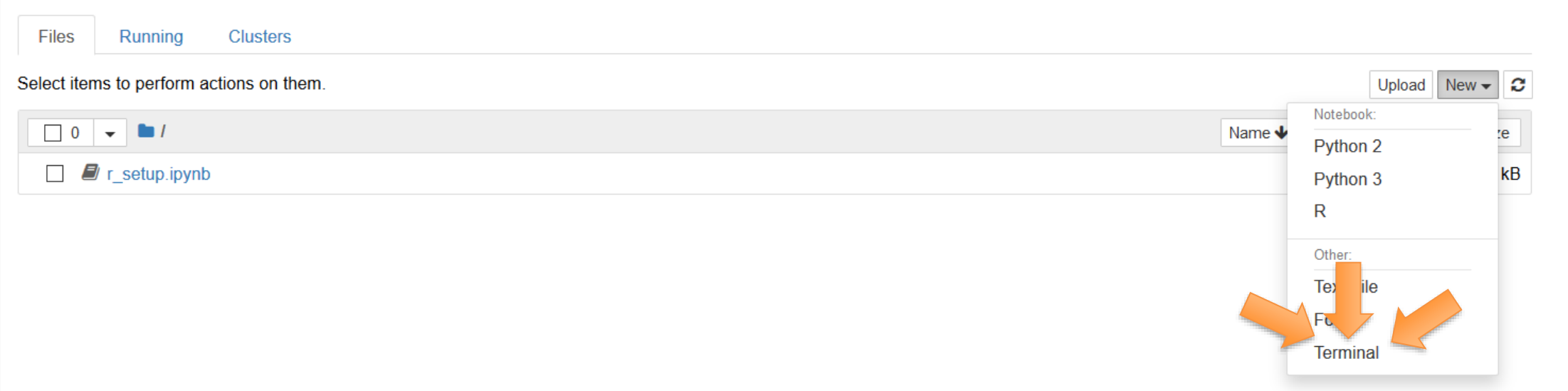
Exercise

Exercise:

1. Open a terminal on the jupyterhub server (On the home screen: New->Terminal)
2. Use `ls` to view all files in the directory
3. Google “linux count lines in file” and determine how many lines are in the r_setup notebook
4. Close the terminal (See the “Running” tab on the home screen)

Solutions

1.


Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
r_setup.ipynb
kB
Notebook:
Python 2
Python 3
R
Other:
Terminal

Solutions

2.

```
(base) root@ip-10-10-229-146:/jupyteruser/40960295# ls
r_setup.ipynb
```

Solutions

3.

```
(base) root@ip-10-10-229-146:/jupyteruser/40960295# wc -l r_setup.ipynb
104 r_setup.ipynb
```

Solutions

4.


Files
Running
Clusters
Currently running Jupyter processes
Terminals
>_ terminals/1
Shutdown
Notebooks
r_setup.ipynb
Python 3
Shutdown
seconds ago

APPENDIX

Contents of 109b.yml:

```
name: 109b
dependencies:
  - python=3.6
  - r-base
  - anaconda
  - seaborn
  - gensim
  - nltk
  - rpy2
  - pip:
    - tensorflow
    - keras
    - pyjags
```

Can you tell how to add more packages, or specify/change version numbers?