

COLLEGE FOR KIDS & TEENS

Introduction to Data Science

Patrick Williams, 2021 Instructor

- Intro: What is Data Science?
- ETL & Data Sourcing
- Data Exploration & Visualization
- Model Building & Implementation



The UWM College For Kids & Teens
Data Science Certificate Track –
In Partnership With Northwestern
Mutual Data Science Institute

NORTHWESTERN MUTUAL
DATA SCIENCE INSTITUTE



Visit uwm.edu/sce/kids

Configuring MacBooks as an Open-Source Data Science Environment

June 17, 2021

Contents:

Introduction

1. Pin App Icons to Taskbar

2. Create the /Python Projects Directory

3. Download Software to Python Projects Directory

Python® Download

4. Install Python®-related Software in the Python Projects Directory

Python® Install Verify Python® and PIP® Versions

5. Install Non-Python® Software in Default Directories

R®

6. Install and Configure Jupyter® Notebook for R

Install Jupyter® Notebook Add R® Kernel to Jupyter® Notebook

----- For readability, trademarks are used only on the first two pages in their initial prominent uses. -----

Trademark Notices:

Software

Chromebook®

Debian®

Linux®

Python®, PIP®

Git®

R®

Jupyter®

Organization to which registered

Google, LLC

Software in the Public Interest, Inc.

Linus Thorvalds

Python Software Foundation

Software Freedom Conservancy

The R Foundation for Statistical Computing

NumFocus Foundation

Introduction

The purpose of this document is to describe how to configure a MacBook® to engage in data science. This includes the use of the following open-source or free software:

- **Python®** – a powerful and popular language used in data science to access, transform, explore, model and visualize data, with many other applications outside of data science.
- **R®** - a very powerful and popular language and environment for statistical computing and graphics/visualization.
- **Jupyter® Notebook** – a simple open-source browser-based application that comes with Python®. It allows the creation and sharing of Python® code, visualizations, narrative text and comments, etc. Jupyter Notebook works with Python by default, but we will configure it to work also with R.

Data science work often involves writing commands in the terminal. Here is a recommended free resource for looking up command syntax for the Mac Terminal: <https://www.makeuseof.com/tag/mac-terminal-commands-cheat-sheet/>

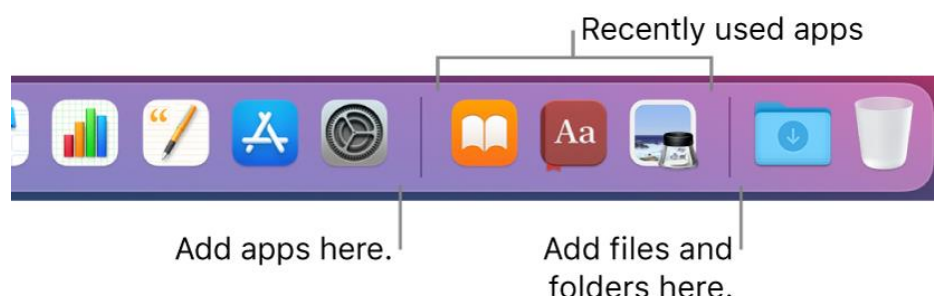
This document focuses on the use of MacBook®. Installing and configuring a data science system as described in this document should give even novice users the ability to jump right into the many educational services available for learning data science, including Udemy®, Coursera®, Data Camp®, etc.

Good luck and good learning!

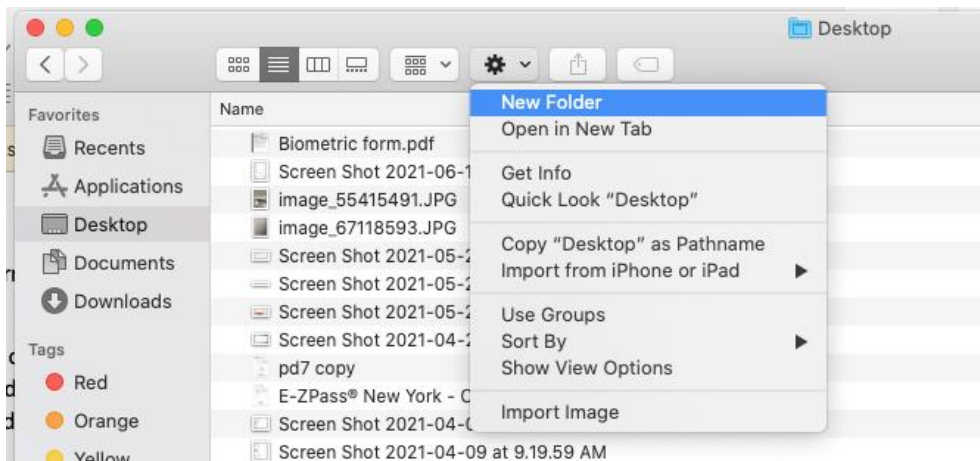
1. Add App Icons to the Dock

Because we may be using the Terminal, TextEdit and the class Folder apps frequently, we will pin them to the Dock for convenience. To do this:

- Drag apps to the left side of (or above) the line that separates the recently used apps. Drag files and folders to the right side of (or below) the other line that separates recently used apps. An *alias* for the item is placed in the Dock.



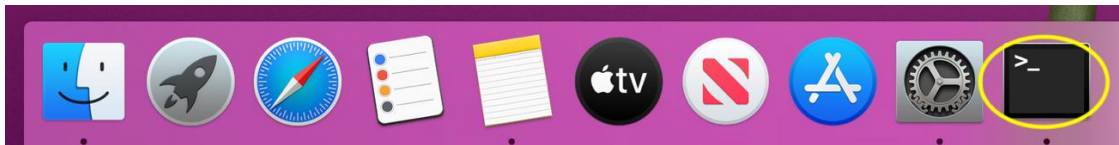
- For the Terminal, press **COMMAND + SPACE** and search for terminal. It will open and you'll see it's icon/alias in the Dock on the Recently used apps right side. Right click the icon for terminal and go to Options → Keep in Dock. This will move it to the left side of the Dock and will stay there.
- For TextEdit, either click on Launchpad to find the TextEdit app and drag it to the Dock OR press **COMMAND + SPACE**, search for TextEdit and do the same as above as you did for Terminal.
- For Files – whichever Folder you are using for this course, let's say it's in Documents. Go to Finder and right click DOCUMENTS on the left side of the screen, and click 'Add to Dock'. You can also use DESKTOP if you'd like instead. Make sure you save whatever files for this class in one consistent place, so please choose Documents or Desktop and create a folder within that.
- To create a folder in Desktop or Documents, click on the button as depicted below and click 'New Folder'. Rename the folder to whatever you are choosing to name the class and to find it easily. Example 'UWM College for Kids Data Science'. (see picture on top of page 3)



2. Create the /Python_Projects Directory

In this step we create a directory for all of your Python-related software and work.

- If you happened to close your Terminal session, click on the **Terminal icon** you just added to the Dock.



- At the Terminal command prompt, type the following commands, using Enter after each (you can copy from commands from this document and paste into the command line):
 - **pwd** (show the current directory, which is /home/baba, or ~)
 - **cd** (move to the top of the directory structure)
 - **ls -a** (list directory contents alphabetically)
 - **mkdir Python_Projects** (make a new directory called Pthyon_Projects)
 - **cd Py*_*** (change directory to Pthyon_Projects using wildcards)
 - **clear** (clear the screen)

```
Python_Projects — -zsh — 82x42
[per5474@S402842 ~ % pwd
/Users/per5474
[per5474@S402842 ~ % cd
[per5474@S402842 ~ % ls -a
.                  .local
..                 .oracle_jre_usage
.CFUserTextEncoding .rstudio-desktop
.DS_Store          .tanium-user-key-encryption.key
.R                 .viminfo
.Rapp.history       .zsh_history
.Renviron           Applications
.Renviron.swp       Desktop
.Rhistory           Documents
.Trash              Downloads
.bash_history       Library
.bash_profile       Movies
.config            Music
.cups               Pictures
.eclipse            Public
.gnupg              jbr_err_pid46660.log
.jenv
[per5474@S402842 ~ % mkdir Python_Projects
[per5474@S402842 ~ % ls -a
.                  .local
..                 .oracle_jre_usage
.CFUserTextEncoding .rstudio-desktop
.DS_Store          .tanium-user-key-encryption.key
.R                 .viminfo
.Rapp.history       .zsh_history
.Renviron           Applications
.Renviron.swp       Desktop
.Rhistory           Documents
.Trash              Downloads
.bash_history       Library
.bash_profile       Movies
.config            Music
.cups               Pictures
.eclipse            Public
.gnupg              Python_Projects
.jenv              jbr_err_pid46660.log
[per5474@S402842 ~ % cd Py*_
[per5474@S402842 Python_Projects % clear
```

- At this point we will be working from the **/Python_Projects** directory, so don't close it.

3. Download Software to Python_Projects Directory

Now we use the Terminal command line to download Python and R. Throughout the rest of this document, anytime a Terminal command returns a confirmation dialog requiring a yes response, enter either **Y**, **Yes** or **y**, depending on what is being requested.

- **Python Download** – Here we install first dependencies needed for Python, and then download the Python software, which will be installed later.

- If you are not already in the directory, enter the command **cd Python_Projects**. Then clear the screen if needed by entering the command **clear**.

```
Python_Projects
[per5474@S402842 ~ % cd Python_Projects
per5474@S402842 Python_Projects % clear
```

- Next, install Homebrew, which is a program that will allow you to download Python onto your MacBook.

- Enter the following into your Terminal command line:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

```
[per5474@S402842 Python_Projects % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password).
[Password:
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
Press RETURN to continue or any other key to abort
```

- Next type this into the terminal command line: **brew install python**

- The terminal will then take a few minutes to download the python software. Wait until it is finished for the next step.

```
per5474@S402842 Python_Projects % brew install python
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
argocd-autopilot  gitwatch          mongocli          revive
at-spi2-atk       gnupg@2.2         mongosh           rmw
at-spi2-core      gpg-tui           mr2               rpg-cli
atuin             gradle@6          neovim-remote    scotch
autoconf@2.69     grepip            nomino            search-that-hash
autoresctic       gtksourceview5    nox               seqkit
```

- If you want to add Python packages, you can do so by writing in the Terminal command line: **pip3 install <package>**. To start Python, just type **Python3** into the terminal command line. To

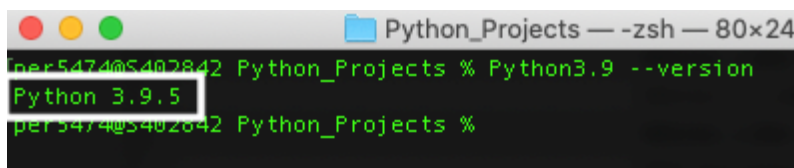
check if Python is working, enter `2+2`. If 4 is returned, then Python is open. To exit Python, write `exit()` and it will bring you back to your base directory in the Terminal.

```
per5474@S402842 ~ % python3
Python 3.8.2 (default, Dec 21 2020, 15:06:04)
[Clang 12.0.0 (clang-1200.0.32.29)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> 2+2
4
[>>> exit()
per5474@S402842 ~ % 2+2
zsh: command not found: 2+2
per5474@S402842 ~ %
```

- Now we can check the installed versions of both Python and PIP. Python is installed with a standard set of libraries (a.k.a., packages or modules) and functionality. However, there are many additional libraries that offer functionality far beyond the standard packages. PIP is installed with Python, allows users to install and manage additional packages.

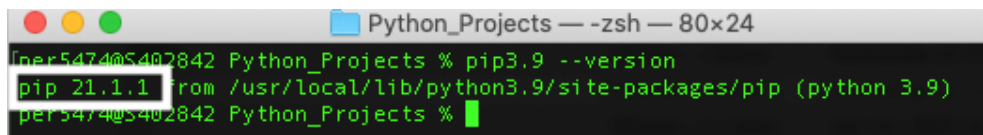
- **Verify Python and PIP Versions**

- **Python Version** – Enter the command `python3.9 --version`.



```
per5474@S402842 Python_Projects % python3.9 --version
Python 3.9.5
per5474@S402842 Python_Projects %
```

- **PIP version** - Enter the command `pip3.9 --version`.



```
per5474@S402842 Python_Projects % pip3.9 --version
pip 21.1.1 from /usr/local/lib/python3.9/site-packages/pip (python 3.9)
per5474@S402842 Python_Projects %
```

- **Note: PIP Upgrade** – When a new version of PIP is available, the user is notified while using it to install software with a message similar to below. When this happens, simply enter the command `python3.9 -m pip install --upgrade pip`.
- Next, install the pandas package. Type `pip install pandas` into the Terminal command line (make sure you exit out of Python3 before).



```
per5474@S402842 Python_Projects % pip install pandas
Collecting pandas
  Downloading pandas-1.2.4-cp39-cp39-macosx_10_9_x86_64.whl (10.7 MB)
    |████████████████████| 10.7 MB 3.5 MB/s
Collecting numpy>=1.16.5
  Downloading numpy-1.20.3-cp39-cp39-macosx_10_9_x86_64.whl (16.1 MB)
    |████████████████████| 16.1 MB 20.2 MB/s
Collecting python-dateutil>=2.7.3
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
    |████████████████████| 227 kB 4.6 MB/s
Collecting pytz>=2017.3
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
    |████████████████████| 510 kB 16.0 MB/s
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, pytz, python-dateutil, numpy, pandas
Successfully installed numpy-1.20.3 pandas-1.2.4 python-dateutil-2.8.1 pytz-2021
.1 six-1.16.0
per5474@S402842 Python_Projects %
```

4. Install Non-Python Software in Default Directories

In this section we install the R software.

- **R Installation** – To download R, go to <https://www.r-project.org/> and click ‘download R’, choose a CRAN mirror (choose one close to you, which is probably Case Western University). Click ‘Download R for MacOS’ and then click on R-4.1.0.pkg. Open the file when it is done downloading. Follow the steps to download the R software and wait for it to download and install.
- You can choose to add R to your Dock if you’d like, just follow the steps in Part 1 of this document. But you can also work with R in the Terminal.
- **Starting and Closing R** – To start R, enter the command **R** on the Terminal command line. The R command line will appear. Enter quit() in the R command line to exit R and return to the Terminal command line.



```
per5474@S402842 Python_Projects % R

R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[> quit()]
Save workspace image? [y/n/c]: n
per5474@S402842 Python_Projects %
```

5. Install and Configure Jupyter Notebook for R

- **Install Jupyter Notebook** – Enter the following four commands:
 - **cd Python_Projects** (if not already there)
 - **pip install jupyter notebook** (installs software)
 - **jupyter notebook** (starts Jupyter Notebook server and opens browser)

```
[per5474@S402842 Python_Projects % jupyter notebook]
[I 21:54:06.276 NotebookApp] Writing notebook server cookie secret to /Users/per5474/Library/Jupyter/runtime/notebook_cookie_secret
[I 21:54:06.550 NotebookApp] Serving notebooks from local directory: /Users/per5474/Python_Projects
[I 21:54:06.551 NotebookApp] Jupyter Notebook 6.4.0 is running at:
[I 21:54:06.551 NotebookApp] http://localhost:8888/?token=7baef4933ffb0b443d209dfea6dc075734fc8c2319d63ccb
[I 21:54:06.551 NotebookApp] or http://127.0.0.1:8888/?token=7baef4933ffb0b443d209dfea6dc075734fc8c2319d63ccb
[I 21:54:06.551 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:54:06.556 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/per5474/Library/Jupyter/runtime/nbserver-26214-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=7baef4933ffb0b443d209dfea6dc075734fc8c2319d63ccb
or http://127.0.0.1:8888/?token=7baef4933ffb0b443d209dfea6dc075734fc8c2319d63ccb
```

A browser opens showing Jupyter Notebook showing access to content from all of the directories below that from which we invoked it, which was /Python_Projects. Hence we know that **we can invoke Notebook from any directory below which our desired content (code) resides**.



Note that when attempting to create a new file (notebook), we are only given the option to create one for Python code. That is by design, since Jupyter Notebook is part of the Python software distribution.

What if we wish to use R code? We do that below.

- **Add R Kernel to Jupyter Notebook** – Do the following:
 - Enter the command **R** on the **Terminal command line**, causing the **R console** to appear.
 - Enter the following four commands on the R command line:
 - **MyRepo='https://cran.case.edu'** (get R software from Case University)
 - **install.packages('IRkernel',MyRepo)** (install the R kernel package)
 - **IRkernel::installspec(user = TRUE)** (apply the R Kernel to the user installing it)
 - **quit()** (answer **n** to saving workspace; returns to the Linux command line, as below)

```
baba@penguin: ~
baba@penguin:~$ R

R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
...
> MyRepo='https://cran.case.edu'
> install.packages('IRkernel',MyRepo)
Warning in install.packages("IRkernel", MyRepo) :
  'lib = "https://cran.case.edu"' is not writable
Would you like to use a personal library instead? (yes/No/cancel) yes
Would you like to create a personal library
'~/R/x86_64-pc-linux-gnu-library/4.0'
to install packages into? (yes/No/cancel) yes
...
> IRkernel::installspec(user = TRUE)
[InstallKernelSpec] Installed kernelspec ir in /home/baba/.local/share/j
>
> quit
function (save = "default", status = 0, runLast = TRUE)
.Internal(quit(save, status, runLast))
<bytecode: 0x5884623ec2e0>
<environment: namespace:base>
> quit()
Save workspace image? [y/n/c]: n
```

- Restart the Jupyter Notebook server by entering **Jupyter notebook** on the **Terminal command line**.

```
baba@penguin: /Python_Projects
baba@penguin: /Python_Projects$ jupyter notebook
```

A Jupyter Notebook opens as before, but with the ability to create a new notebook for R code, as below.



Upon quitting the notebook, the browser closes and the server stops, returning to the Terminal command line.

At this point an excellent collection of Data Science tools has been installed on your machine.
Congratulations!