Getting Started with Jupyter Notebooks

Reproducible Research Workshop

dataservices.library.jhu.edu

JHU Data Services: Marley Kalt, Reina Murray, Pete Lawson, Chen Chiu

Date: December 8, 2020

If you have not already installed Anaconda, please do so now!

Get it here: https://www.anaconda.com/products/individual
Click "Download" or scroll down to "Anaconda Installers"

Outline:

- Learning Objectives
- Software
    - Interactive Notebooks
    - Jupyter
    - IPython
    - Anaconda
- Tutorial
- Reproducible Notebooks
- Resources

JHU DATA SERVICES

WE HELP FACULTY, RESEARCHERS AND STUDENTS

FIND **USE** **MANAGE** **VISUALIZE** **SHARE**

FIND OUT MORE

GO TO dataservices.library.jhu.edu

EMAIL dataservices@jhu.edu

SHARE AT archive.data.jhu.edu

JOHNS HOPKINS LIBRARIES | Data Services

Today, you will learn:

- What interactive notebooks are, including Jupyter Notebooks
- The basics of running code and writing markdown-formatted text in a Jupyter Notebook
- How interactive notebooks are useful for reproducible research

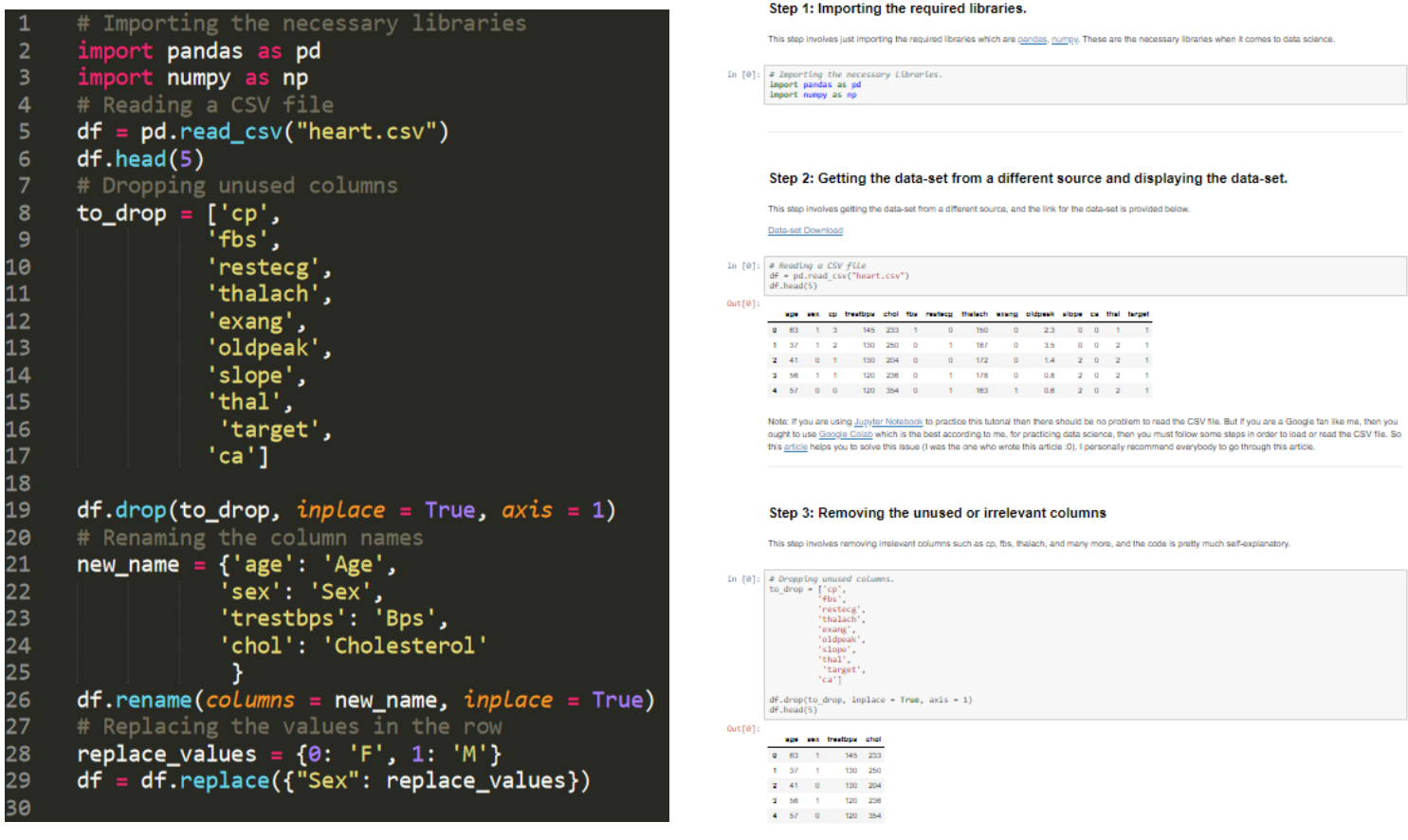
You will not learn:

- How to write Python code or use specific Python libraries

Imagine you want to know how your colleague has cleaned their data.

Which will be easier to understand?

```python
# Importing the necessary libraries
import pandas as pd
import numpy as np
# Reading a CSV file
df = pd.read_csv("heart.csv")
df.head(5)
# Dropping unused columns
to_drop = ['cp',
           'fbs',
           'restecg',
           'thalach',
           'exang',
           'oldpeak',
           'slope',
           'thal',
           'target',
           'ca']

df.drop(to_drop, inplace = True, axis = 1)
# Renaming the column names
new_name = {'age': 'Age',
            'sex': 'Sex',
            'trestbps': 'Bps',
            'chol': 'Cholesterol'
             }
df.rename(columns = new_name, inplace = True)
# Replacing the values in the row
replace_values = {0: 'F', 1: 'M'}
df = df.replace({"Sex": replace_values})

```



Step 1: Importing the required libraries.

This step involves just importing the required libraries which are pandas, numpy. These are the necessary libraries when it comes to data science.

```
In [0]: # Importing the necessary libraries.
        import pandas as pd
        import numpy as np
```

Step 2: Getting the data-set from a different source and displaying the data-set.

This step involves getting the data-set from a different source, and the link for the data-set is provided below.

Data-set Download

```
In [0]: # Reading a CSV file
        df = pd.read_csv("heart.csv")
        df.head(5)
```

Out[0]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Note: If you are using Jupyter Notebook to practice this tutorial then there should be no problem to read the CSV file. But if you are a Google fan like me, then you ought to use Google Colab which is the best according to me, for practicing data science, then you must follow some steps in order to load or read the CSV file. So this article helps you to solve this issue (I was the one who wrote this article :0). I personally recommend everybody to go through this article.

Step 3: Removing the unused or irrelevant columns

This step involves removing irrelevant columns such as cp, fbs, thalach, and many more, and the code is pretty much self-explanatory.

```
In [0]: # Dropping unused columns.
        to_drop = ['cp',
                   'fbs',
                   'restecg',
                   'thalach',
                   'exang',
                   'oldpeak',
                   'slope',
                   'thal',
                   'target',
                   'ca']

        df.drop(to_drop, inplace = True, axis = 1)
        df.head(5)
```

Out[0]:

	age	sex	trestbps	chol
0	63	1	145	233
1	37	1	130	250
2	41	0	130	204
3	56	1	120	236
4	57	0	120	354

Software

Interactive Notebooks

- A graphical user interface (GUI) for writing code, text, visualizations, and more
- Designed to make it easier to read and write code
- Typically made of cells, or chunks of code/text, that can be run individually to explore step-by-step results
- Can be general purpose, or specific to a programming language or discipline
- Overview of different notebooks from Morphocode

Why use interactive notebooks?

- Developing and debugging code
    - Enables users to test small chunks of code and examine output in real time
- Sharing code
    - Keeps code and output in a single document
    - Easier for a non-technical audience to read, run, and understand results
    - Can export code, visualizations, and analysis in multiple formats, including PDF and HTML
- Explaining code
    - Notebooks give space for a written explanation of your code and analysis
    - Jupyter Notebooks use markdown for text
        - Markdown: syntax for formatting plain text
        - Learn more at https://www.markdownguide.org/

Jupyter

Website: jupyter.org

Project Jupyter

- Jupyter is an open source project for creating interactive and reproducible code
- Creator of Jupyter Notebooks, one of the most popular interactive notebooks for data analysis and visualization

Jupyter Notebooks

- Interactive notebooks for data analysis and visualization
- Editable user interface, runs in a web browser
- Kernels, software that executes the code
    - Supports several programming languages (Python, R, Julia, and more)

Screenshot of a Jupyter Notebook:

Step 1: Importing the required libraries.

This step involves just importing the required libraries which are pandas, numpy. These are the necessary libraries when it comes to data science.

In [0]:
```
# Importing the necessary libraries.
import pandas as pd
import numpy as np
```

Step 2: Getting the data-set from a different source and displaying the data-set.

This step involves getting the data-set from a different source, and the link for the data-set is provided below.

Data-set Download

In [0]:
```
# Reading a CSV file
df = pd.read_csv("heart.csv")
df.head(5)
```

Out[0]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Note: If you are using Jupyter Notebook to practice this tutorial then there should be no problem to read the CSV file. But if you are a Google fan like me, then you ought to use Google Colab which is the best according to me, for practicing data science, then you must follow some steps in order to load or read the CSV file. So this article helps you to solve this issue (I was the one who wrote this article :D). I personally recommend everybody to go through this article.

Step 3: Removing the unused or irrelevant columns

This step involves removing irrelevant columns such as cp, fbs, thalach, and many more, and the code is pretty much self-explanatory.

In [0]:
```
# Dropping unused columns.
to_drop = ['cp',
           'fbs',
           'restecg',
           'thalach',
           'exang',
           'oldpeak',
           'slope',
           'thal',
           'target',
           'ca']

df.drop(to_drop, inplace = True, axis = 1)
df.head(5)
```

Out[0]:

	age	sex	trestbps	chol
0	63	1	145	233
1	37	1	130	250
2	41	0	130	204
3	56	1	120	236
4	57	0	120	354

IPython

Website: ipython.org

- A command line interface to use Python interactively
- A Jupyter kernel to write Python code within a Jupyter notebook
- .ipynb file extension
    - File extension for Jupyter Notebooks (started with the IPython kernel)
    - Stores notebooks in JSON format

Screenshot of a Jupyter Notebook, in a text editor:

Source: Data Cleaning using Python with Pandas Library by Tanu Nanda Prabhu
Retrieved from: A Gallery of Interesting Jupyter Notebooks

Anaconda Individual Edition

Website: anaconda.com

- An open source distribution for scientific computing
- Includes Python, Jupyter Notebooks, hundreds of additional libraries, a package and environment manager, and a graphical user interface (GUI)
- Popular in the data science community

Jupyter Notebook Tutorial

In this tutorial, we will:

1. Learn how to create a new notebook and open an existing notebook
2. Add and remove code cells
3. Switch between code and markdown
4. Run code and markdown cells

For screenshots, written instructions, and additional functions of Jupyter Notebooks, see the provided **JupyterNotebookTutorial.ipynb** file

Note: Make sure to have the provided Images folder saved in the same directory as the tutorial

Jupyter Notebooks + Reproducibility

Jupyter Notebooks + Reproducibility

- Jupyter Notebooks are a tool to help create reproducible research
- Code written in a notebook is not automatically reproducible

Reproducibility includes:

- Availability of data
- Availability of software and libraries/packages
- Documentation of procedures, environments, and versions
- Persistent location

Discussion: Is this notebook reproducible?

Analysis and visualization of a public OKCupid profile dataset using python and pandas

Author: Alessandro Giusti (web, email), Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI.

In [1]:
```
# notebook setup
%matplotlib inline
%config InlineBackend.figure_format='svg'
from IPython.display import display,HTML
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from prettypandas import PrettyPandas
sns.set_style("ticks")
sns.set_context(context="notebook",font_scale=1)
```

In [ ]:
```
d=pd.read_csv("profiles.csv/profiles.csv")
print("The dataset contains {} records".format(len(d)))

m=d[d["sex"]=="m"] # male users
f=d[d["sex"]=="f"] # female users
print("{} males ({:.1%}), {} females ({:.1%})".format(
    len(m),len(m)/len(d),
    len(f),len(f)/len(d)))

PrettyPandas(d                                    # Prettyprints pandas dataframes
    .head(10)                                     # Sample the first 10 rows
    [[c for c in d.columns if "essay" not in c]] # Ignore columns with "essay" in the name (they are long)
)
```

In [ ]:
```
""" Height distribution, compare with data from CDC """

fig,(ax,ax2) = plt.subplots(nrows=2,sharex=True,figsize=(6,6),gridspec_kw={'height_ratios':[2,1]})

# Plot histograms of height
bins=range(55,80)
sns.distplot(m["height"].dropna(), ax=ax,
             bins=bins,
             kde=False,
             color="b",
             label="males")
sns.distplot(f["height"].dropna(), ax=ax,
             bins=bins,
             kde=False,
             color="r",
             label="females")
ax.legend(loc="upper left")
ax.set_xlabel("")
ax.set_ylabel("Number of users with given height")
ax.set_title("Height distribution of male and female users");
```

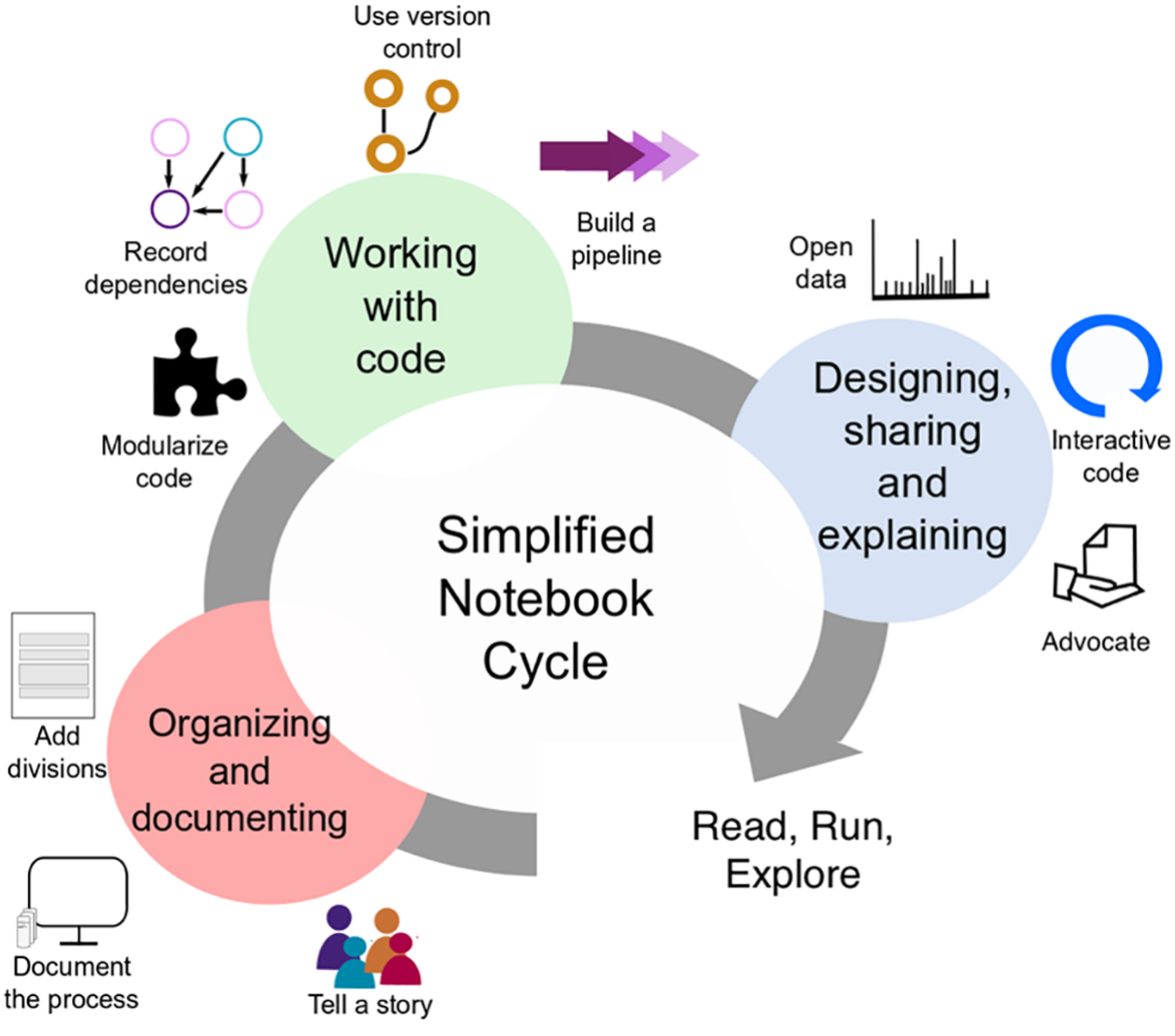
```
ax.set_title("Height distribution of male and female users");

# Make aligned boxplots
sns.boxplot(data=d,y="sex",x="height",orient="h",ax=ax2,palette={"m":"b","f":"r"})
plt.setp(ax2.artists, alpha=.5)
ax2.set_xlim([min(bins),max(bins)])
ax2.set_xlabel("Self-reported height [inches]")

sns.despine(ax=ax)
fig.tight_layout()
```

In [ ]:
```
# Import a CSV file for growth chart data
cdc=pd.read_csv("https://www.cdc.gov/growthcharts/data/zscore/statage.csv")
```

Tips for creating reproducible notebooks:

- Make use of markdown (section headings, narrative text)
- One step = one code cell
- Pay attention to dependencies, notebooks should run top to bottom
- Use descriptive variable names and document them
- Make data accessible, use relative file paths
- Clean up your notebook!

Workflow for reproducible Jupyter Notebooks:



Source: https://doi.org/10.1371/journal.pcbi.1007007

Discussion: Is this notebook reproducible?

Analysis and visualization of a public OKCupid profile dataset using python and pandas

Author: Alessandro Giusti (web, email), Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI.

Discussion

After publication, this notebook received quite a lot of insightful comments on /r/python (link to post) and /r/okcupid (link to post).

Introduction

This document is an analysis of a public dataset of almost 60000 online dating profiles. The dataset has been published in the Journal of Statistics Education, Volume 23, Number 2 (2015) by Albert Y. Kim et al., and its collection and distribution was explicitly allowed by OkCupid president and co-founder Christian Rudder. Using these data is therefore ethically and legally acceptable; this is in contrast to another recent release of a different OkCupid profile dataset, which was collected without permission and without anonymizing the data (more on the ethical issues in this Wired article).

Notebook setup

In [1]:
```
%matplotlib inline
%config InlineBackend.figure_format='svg'
from IPython.display import display,HTML
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from prettypandas import PrettyPandas
sns.set_style("ticks")
sns.set_context(context="notebook",font_scale=1)
```

Dataset details

The data is available at this link. The codebook includes many details about the available fields. The dataset was collected by web scraping the OKCupid.com website on 2012/06/30, and includes almost 60k profiles of people within a 25 mile radius of San Francisco, who were online in the previous year (after 06/30/2011), with at least one profile picture.

The CSV contains a row (observation) for each profile. Let's have a look at the first 10 profiles, excluding the columns whose name contains the string "essay", which contain a lot of text and are not practical at the moment.

Study height distribution and compare with official data from the US Centers of Disease Control and Prevention (CDC)

We first plot the height distribution for males and females in the whole dataset

In [ ]:
```
fig,(ax,ax2) = plt.subplots(nrows=2,sharex=True,figsize=(6,6),gridspec_kw={'height_ratios':[2,1]})

# Plot histograms of height
bins=range(55,80)
sns.distplot(m["height"].dropna(), ax=ax,
```

```
                bins=bins,
                kde=False,
                color="b",
                label="males")
sns.distplot(f["height"].dropna(), ax=ax,
                bins=bins,
                kde=False,
                color="r",
                label="females")
ax.legend(loc="upper left")
ax.set_xlabel("")
ax.set_ylabel("Number of users with given height")
ax.set_title("height distribution of male and female users");

# Make aligned boxplots
sns.boxplot(data=d,y="sex",x="height",orient="h",ax=ax2,palette={"m":"b","f":"r"})
```

Resources

General Resources

Project Jupyter - organization behind Jupyter Notebooks
Anaconda - environment manager and GUI for launching Jupyter Notebooks
RISE slideshow extension for Jupyter Notebooks
Guide to interactive notebooks
Basic Markdown syntax for formatting text elements

Tutorials and Examples

Real Python introduction to Jupyter Notebooks
Jupyter Notebooks documentation and tutorials
Jupyter Notebooks gallery on GitHub
Towards Data Science - online publication, dozens of articles on Jupyter Notebooks and other data science topics for beginner to advanced levels

Reproducibility

Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks
Reproducibility guide for Jupyter Notebooks
Jupyter Notebooks and reproducible data science

Conferences

JupyterCon - past talks available on YouTube

Take our survey to help us improve this workshop:

https://www.surveymonkey.com/r/ReproducibleResearch

If you are attending more webinars from the Reproducible Research series, please take the survey at the end of the day!

Today's schedule:

9:30-10:30am - Introduction to Reproducible Research
10:45-12:15pm - Getting Started with Jupyter Notebooks
1:30-3:00pm - Getting Started with R Markdown
3:10-3:30pm - Drop-in to troubleshoot Git/GitHub installation
3:30-4:30pm - Version Control: Using Git and GitHub

Questions?

Contact us at dataservices@jhu.edu

About this Presentation

This presentation was created using Jupyter Notebooks version 6.0.1 and the RISE notebook extension version 5.6.1.

Terms of Use

> *Johns Hopkins University Data Services. (2020, December 8). Reproducible research: Getting started with Jupyter Notebooks [workshop presentation].*