Jupyter Organization Tips

Introduction

All of these slides are **recommendations**.

Goals:

- Reproducible Analysis
- Readable Code
- Understandable by others (and your future self)

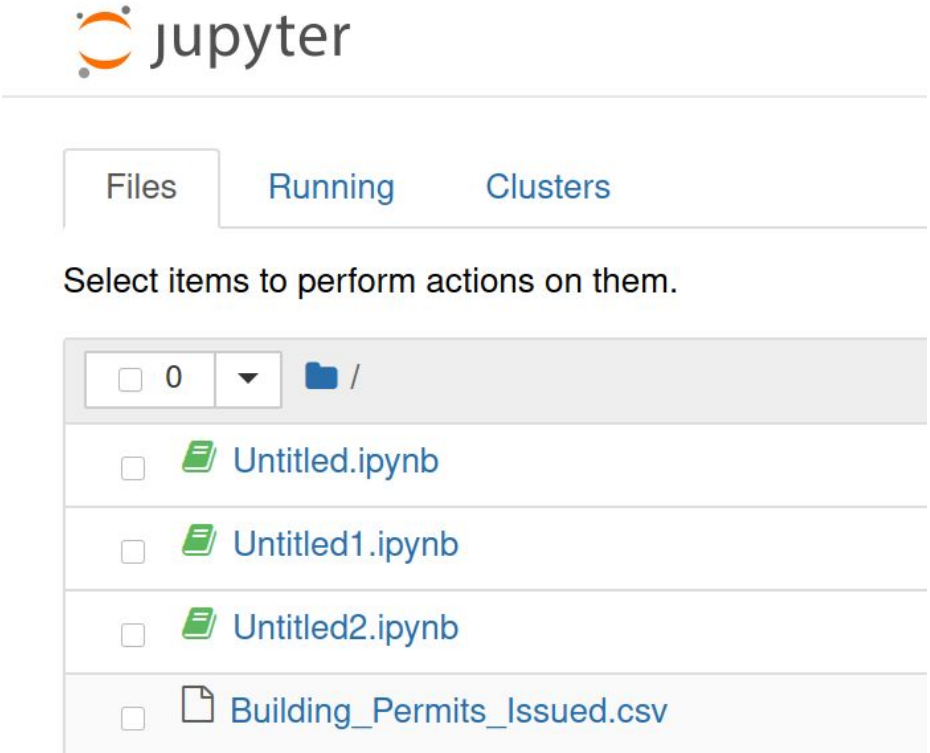
General Guidelines for Python Code

PEP 8 - Style Guide for Python Code

Guidelines to improve readability and consistency of code.

Recommendations

Name your notebooks!



Recommendations

Use Markdown to annotate your workflow.

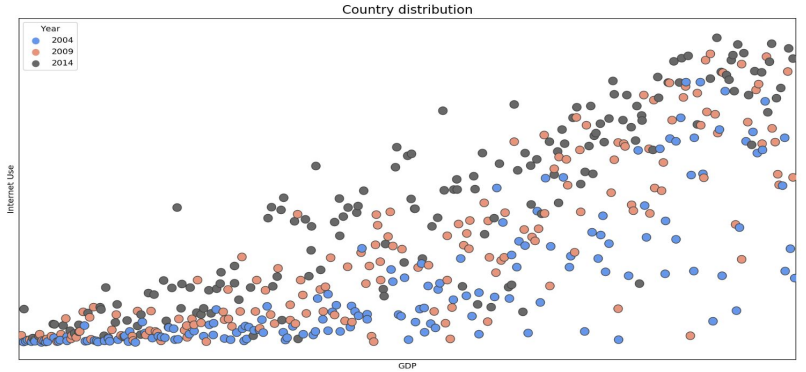
Comment code when appropriate.

Country GDP and internet usage distributions

Plotting of `Year` with x-axis as `GDP_Per_Capita` and y-axis as `Internet_Users_Pct`.

In [8]:
```
plt.figure(figsize=(16,8), clear=True);

ax1 = sns.stripplot(x="GDP_Per_Capita",
                    y="Internet_Users_Pct",
                    data=df_04_09_14,
                    jitter=2,
                    hue="Year",
                    size=10,
                    linewidth=.8,
                    dodge=True);
ax1.set_xlabel('GDP');
ax1.set_ylabel('Internet Use');
ax1.set_yticks([]);
ax1.set_xticks([]);
ax1.axes.set_title('Country distribution', fontsize=15);
```



Observing the plot `ax1` above, we notice that in general, there looks to be a positive correlation between GDP and internet usage. This correlation seems strongest in years 2009 and 2014.

In [ ]:
```
# Data contains two lines of description text, skip to avoid errors.
df = pd.read_csv('data.csv', skiprows=2)
```

Recommendations

Keep imports in the first cell

Remove old/unnecessary code

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt

        %matplotlib inline
```

```
In [2]: crashes = pd.read_csv('crashes_2018.csv')
```

```
In [3]: crashes.head(2)
```

Out[3]:

	Accident Number	Date and Time	Number of Motor Vehicles	Number of Injuries	Number of Fatalities	Property Damage	Hit and Run	Reporting Officer	Collision Type Code	Collision Type Description	...	Harmful Code Description	Street Address	
0	20181075326	12/31/2018 11:10:00 PM	1.0	0	0	NaN	N	240964.0	0.0	NOT COLLISION W/MOTOR VEHICLE-TRANSPORT	...	CURB;SHRUBBERY	ROCKWOOD DR & SADDLESTONE DR	HERMI
1	20181075323	12/31/2018 11:09:00 PM	3.0	1	0	NaN	N	110062.0	4.0	ANGLE	...	MOTOR VEHICLE IN TRANSPORT;PARKED MOTOR VEHICLE	SPRING ST & N 1ST ST	NASH

2 rows × 25 columns

```
In [5]: crashes = crashes.drop(columns = 'Weather Code')
```

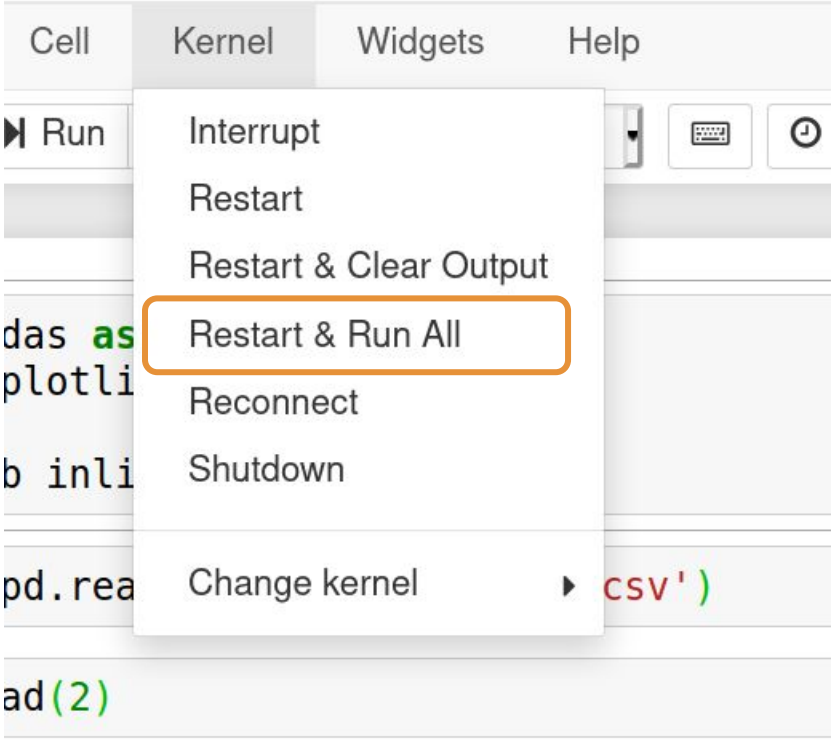
```
In [7]: crashes.head(2)
```

Out[7]:

	Accident Number	Date and Time	Number of Motor Vehicles	Number of Injuries	Number of Fatalities	Property Damage	Hit and Run	Reporting Officer	Collision Type Code	Collision Type Description	...	Harmful Code Description	Street Address	

Recommendations

Periodically check that you can Restart & Run All error-free.



Recommendations

Use logical and human-readable variable names

```
df = pd.read_csv('Building_Permits_Issued.csv',
                 low_memory = True)
```

```
df2 = df[df['Permit Subtype Description'].
         str.contains('Residential')]
```

```
building_permits = pd.read_csv('Building_Permits_Issued.csv',
                               low_memory = True)
```

```
residential_permits = building_permits[building_permits['Permit Subtype Description'].
                                       str.contains('Residential')]
```

Recommendations

- Combine together multiple cells if they don’t produce output
- Take advantage of method chaining

```
In [4]: crashes = crashes.drop(columns = ['Weather Code', 'Illumination Code', 'Harmful Code', 'Mapped Location'])

In [5]: crashes = crashes.rename(columns = lambda col: col.lower().replace(' ', '_'))

In [6]: crashes = crashes.dropna(subset = ['latitude', 'longitude'])
```

```
In [4]: crashes = crashes.drop(columns = ['Weather Code', 'Illumination Code', 'Harmful Code', 'Mapped Location'])
        crashes = crashes.rename(columns = lambda col: col.lower().replace(' ', '_'))
        crashes = crashes.dropna(subset = ['latitude', 'longitude'])
```

Recommendations

- Combine together multiple cells if they don’t produce output
- Take advantage of method chaining

```
In [4]: crashes = crashes.drop(columns = ['Weather Code', 'Illumination Code', 'Harmful Code', 'Mapped Location'])

In [5]: crashes = crashes.rename(columns = lambda col: col.lower().replace(' ', '_'))

In [6]: crashes = crashes.dropna(subset = ['latitude', 'longitude'])
```

```
In [4]: crashes = (crashes
                   .drop(columns = ['Weather Code', 'Illumination Code', 'Harmful Code', 'Mapped Location'])
                   .rename(columns = lambda col: col.lower().replace(' ', '_'))
                   .dropna(subset = ['latitude', 'longitude'])
                  )
```

Recommendations

- Think carefully about **exploration** vs **explanation**.
  - Exploration: Preparing/getting to know the data
  - Explanation: What you want to show to other people
- Consider multiple notebooks for different tasks:
  - Initial cleaning and preparation
  - Exploratory Analysis
  - Model Building
  - Presentation

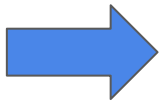
Recommendations

DRY: Don't Repeat Yourself:

- Covert repetitive code blocks into reusable functions.
- Avoid Copy/Paste
- Utilize for loops and list comprehensions
- Use functions to abstract away complexity.

```
# ugly example
pd.qcut(df['Fare'], q=4, retbins=True)[1]
# returns array([0., 7.8958, 14.4542, 31.275, 512.3292])

df.loc[ df['Fare'] <= 7.90, 'FareBand'] = 0
df.loc[(df['Fare'] > 7.90) & (df['Fare'] <= 14.454), 'FareBand'] = 1
df.loc[(df['Fare'] > 14.454) & (df['Fare'] <= 31.275), 'FareBand'] = 2
df.loc[ df['Fare'] > 31.275, 'FareBand'] = 3
```



```
# after refactoring into a function
df['FareBand'] = categorize_column(df['Fare'],
                                   num_bins=4)
```

Other Recommendations

This notebook on creating reproducible research in Jupyter:
https://www.kaggle.com/rtatman/reproducible-research-best-practices-jupytercon

Cookiecutter Data Science:
https://drivendata.github.io/cookiecutter-data-science/

- A template for data science/analytics work
- Probably overkill for us, but there are some useful principles that I recommend adopting (eg. a folder for data and a folder for notebooks, using sensible naming conventions for your notebooks).