**DataCamp**

**Project Introduction to DataCamp Projects**

In each DataCamp project, you will carry out an end-to-end analysis of real-world tasks using real-world tools and workflows.

Using Python, R, and SQL to build a project that has a specific solution, with guided tasks and real-time automated code checks.

A **Jupyter** notebook is a document that contains **text cells** and **code cells**. You can change or add code cells, and then run a cell by first selecting it and then clicking the run cell button of **ctrl + enter**.

A Jupyter notebook in DataCamp project will contain an analysis, data exploration, or other narrative combining code, data, and text.

In addition to displaying basic objects like numbers and strings, notebooks can also support complex objects, like DataFrame, used in data science and this makes them great for interactive data analysis.

import pandas as pd

global\_temp = pd.read\_csv(‘datasets/global\_temperatures.csv’)

global\_temp.head()

“Magic commands” affect the way Jupyter notebooks work and start with either % or %%. The magic command needed to nicely display plots inline is %matplotlib inline

%matplotlib inline

import matplotlib.pyplot as plt

plt.plot(global\_temp[‘year’], global\_temp[‘degrees\_celsius’])

plt.xlabel(‘year’)

plt.ylabel(‘degrees celsius’)

Jupyter notebooks can render many more types of outputs such as sound, animation, video, etc. You can also include interactive widgets (**Jupyter Widgets**) directly in a notebook. A finished notebook contains both the results and the code that produced those results. This is useful when you want to share your findings or if you need to update your analysis with new data.

You can check your project by clicking **Check Project**. This will go through all the tasks and check which are complete. For correct tasks, the task circles will turn green and when something needs to be fixed, the task circles will turn orange. You can then get more information about what needs to be done at the bottom of the orange tasks. You can check a project at any time to see how far you’ve gotten.

Jupyter notebooks support SQL commands. The Jupyter extension ipython-sql will enable connecting to a database and then issuing SQL commands directly from the notebook.

%%sql postgresql:///countries

select \* from countries LIMIT 3;

%sql select \* from countries where name = ‘Belgium’

result = %sql select \* from countries;

df = result.DataFrame()

df.info()

**Project Introduction to DataCamp Projects (continued)**

SQLAlchemy is a Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.

from sqlalchemy import create\_engine

engine = create\_engine(“postgresql:///countries”);

result = engine.execute(“select \* from countries”);

result.keys()

Another chart in Jupyter notebooks:

%matplotlib inline

df.continent.value\_counts().plot(kind=’bar’)