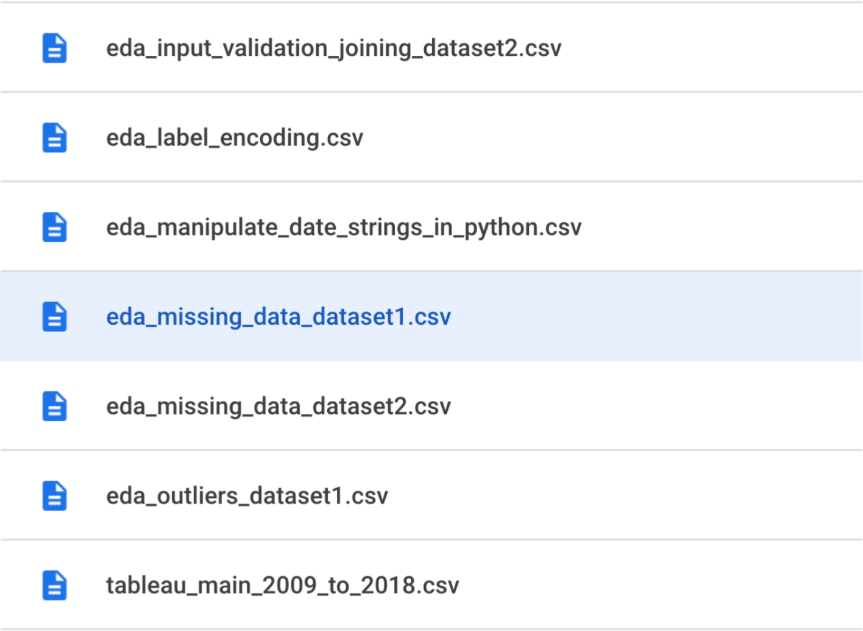
Reference guide: Import datasets with Python

In your career as a data professional, you will come across various datasets that have different file types or are stored in various databases. As you've learned previously, it is critical for you to know what these data types are and how to import data using Python. Below you will find examples of importing both databases through connections and data files into Python.

Although you will use the Coursera platform for Python coding, you will need to know how to work with and import CSV files if you'd like to download and open them outside of Coursera.

How to import a dataset from a CSV file

For this example, find a CSV file referenced in one of the videos and download it on your computer:



eda_input_validation_joining_dataset2.csv

eda_label_encoding.csv

eda_manipulate_date_strings_in_python.csv

eda_missing_data_dataset1.csv

eda_missing_data_dataset2.csv

eda_outliers_dataset1.csv

tableau_main_2009_to_2018.csv

There are several different ways to import a CSV file into Python, but we will only review some of the more common ways. Start by importing the Python library "CSV." You can do that by opening a notebook and typing the following:

```
import csv
```

Next you can use the ‘with()’ function to define the **file name (or file path)** and then the **mode:**

```
with(“filename / file path”, ‘mode’)
```

So the syntax so far is:

```
import csv
with(“filename / file path”, ‘mode’)
```

The mode is telling the Python library what to do with the file. When defining the **mode**, you use one of the following options:

- ‘r’ – read
- ‘w’ – write
- ‘a’ – append
- ‘+’ – create new file

Typically, you’ll be defining the mode inside the “`with open()`” argument field as ‘r,’ because you want Python to open and read the CSV file.

Next, we’ll add “`as file`” to the end, which you’ll recall is giving the file a data frame name, in this case, we’ll name it `csv_df`.

```
import csv
with open("Downloads:\\eda_missing_data_dataset1.csv",'r') as file:
      csv_df = file.read()
```

Importing a CSV file using pandas

Instead of using the CSV package, you can also use pandas to import a CSV data file. First, of course, you’ll want to import pandas in your Python notebook.

```
import pandas as pd
```

Next, you'll use the "read_csv" function to load the data into a dataframe. Most data professionals use abbreviations, like "DF." The file name then goes in the argument field.

```
df = pd.read_csv ('E:\\V-136_df2.csv')
```

Note: *You can also use this same syntax for importing a CSV file that is stored on the internet. In the place of the filename, you would simply copy and paste the url.*

How to import data by connecting to a database

There are a number of databases that you can connect to with Python,such as BigQuery, MySQL, SQLite, and Oracle. Databases are a convenient way for companies and organizations to store huge amounts of data.

Data from databases can be accessed and analyzed via a query within your code, rather than downloading a file. Data professionals are regularly tasked with analyzing data from databases. Below you will find a step-by-step guide for connecting to datasets in BigQuery, which is a large online database.

Step 1: Access BigQuery

First off when you access BigQuery, you'll start a new Google Cloud project. Make sure the billing is enabled. You'll also need to set up your authentication in BigQuery which is described in Google Cloud's resource, Getting started with authentication.

Step 2: Open your Python notebook

Next you should open a Python notebook. You can use IPython magic commands to connect to BigQuery, which makes the import quite simple. Begin by opening a Jupyter notebook and loading the magic commands for big query:

```
%load_ext google.cloud.bigquery
```

From there, you can input the following magic command “%%bigquery” along with the name of any set of data in the database.

```
%%bigquery - country_names_area df
```

Step 3: Select your data

You can then use SQL commands like `SELECT` and `FROM` to select the data you want to use.

```
SELECT * FROM `country_names_area`
```

Key takeaways

There are lots of different kinds of data, which means there are numerous ways to import data. As a data professional, learning several methods to import data, whether it be from a data file or a database, will build a solid foundation for your career.

Resources for more information

To learn more about importing data into Python, you can refer to the following links:

- An overview of importing data in Python
- How to connect to BigQuery from a Colab