Module V

CSV, JSON and SQL

INFO 590: Applied Data Science
Summer 2017

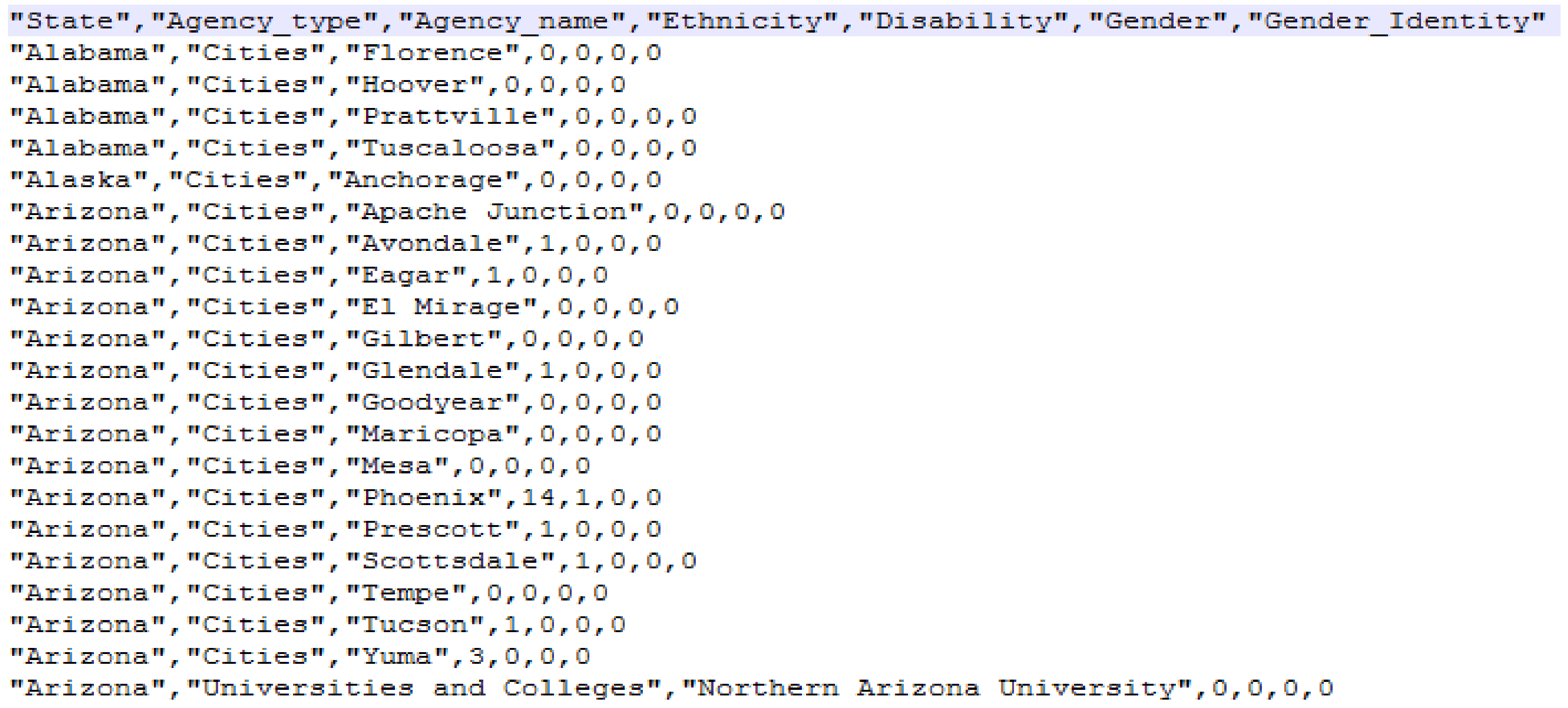
What is a file format?

A file format is a definite structure in which the data is encode for storage in a file. Different file formats can be used to store different types of information like images, text, etc. The format of a file is identified by its extension which is the sequence of letters following the 'dot' at the end of the file name. Eg: .pdf, .csv, .json etc.

What is it important to understand file formats?

For a data scientist, it is important to understand the underlying structure of the data before he/she can start analysing the data. Storing the data in the appropriate format helps in simplifying data pre-processing. Since the data obtained from different sources have different formats, the data scientist has to have an understanding of the different formats in order to effeciently deal with the data.

CSV - Comma Separated Values



```
"State","Agency_type","Agency_name","Ethnicity","Disability","Gender","Gender_Identity"
"Alabama","Cities","Florence",0,0,0,0
"Alabama","Cities","Hoover",0,0,0,0
"Alabama","Cities","Prattville",0,0,0,0
"Alabama","Cities","Tuscaloosa",0,0,0,0
"Alaska","Cities","Anchorage",0,0,0,0
"Arizona","Cities","Apache Junction",0,0,0,0
"Arizona","Cities","Avondale",1,0,0,0
"Arizona","Cities","Eagar",1,0,0,0
"Arizona","Cities","El Mirage",0,0,0,0
"Arizona","Cities","Gilbert",0,0,0,0
"Arizona","Cities","Glendale",1,0,0,0
"Arizona","Cities","Goodyear",0,0,0,0
"Arizona","Cities","Maricopa",0,0,0,0
"Arizona","Cities","Mesa",0,0,0,0
"Arizona","Cities","Phoenix",14,1,0,0
"Arizona","Cities","Prescott",1,0,0,0
"Arizona","Cities","Scottsdale",1,0,0,0
"Arizona","Cities","Tempe",0,0,0,0
"Arizona","Cities","Tucson",1,0,0,0
"Arizona","Cities","Yuma",3,0,0,0
"Arizona","Universities and Colleges","Northern Arizona University",0,0,0,0
```

- Each line of the CSV file indicates a record
- The first line in the file is usually the header which contains the column names
- Columns can be of any data type including string, integer, float, data, time, etc.
- Two consecutive commas in an entry indicate an empty value for the corresponding column

CSV - Comma Separated Values

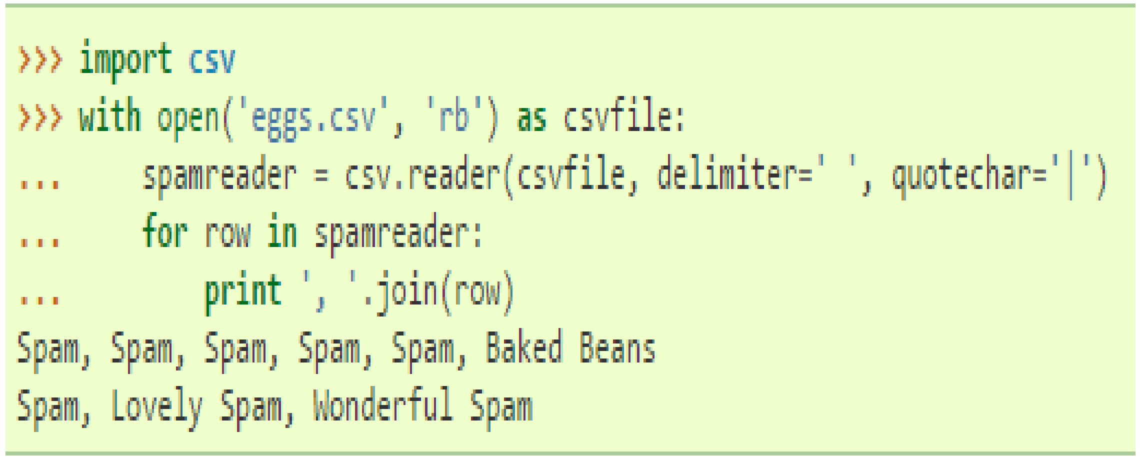
CSV library in Python

Reading CSV files -

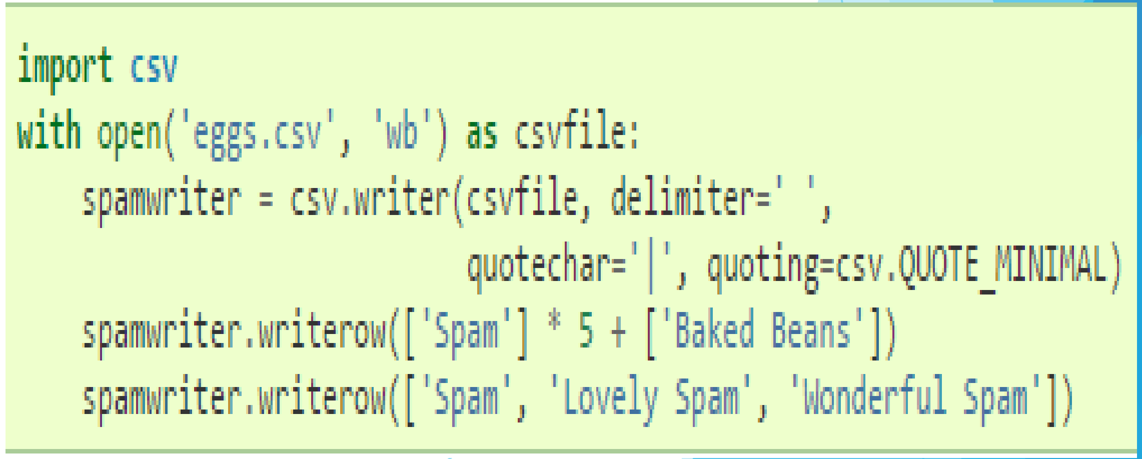
- csv.reader(file) : reads the csv file which can then be stored as list, dictionary, array etc.
- csv.DictReader(file) : maps the information read into a dict whose keys are given by the optional fieldnames parameter

Writing CSV files -

- csv.writer(file) : writes the data in python onto a file as comma separated values
- csv.DictWriter(file) : maps the contents of a dictionary onto a CSV file



```
>>> import csv
>>> with open('eggs.csv', 'rb') as csvfile:
...     spamreader = csv.reader(csvfile, delimiter=' ', quotechar='|')
...     for row in spamreader:
...         print ', '.join(row)
Spam, Spam, Spam, Spam, Spam, Baked Beans
Spam, Lovely Spam, Wonderful Spam
```



```
import csv
with open('eggs.csv', 'wb') as csvfile:
    spamwriter = csv.writer(csvfile, delimiter=' ',
                            quotechar='|', quoting=csv.QUOTE_MINIMAL)
    spamwriter.writerow(['Spam'] * 5 + ['Baked Beans'])
    spamwriter.writerow(['Spam', 'Lovely Spam', 'Wonderful Spam'])
```

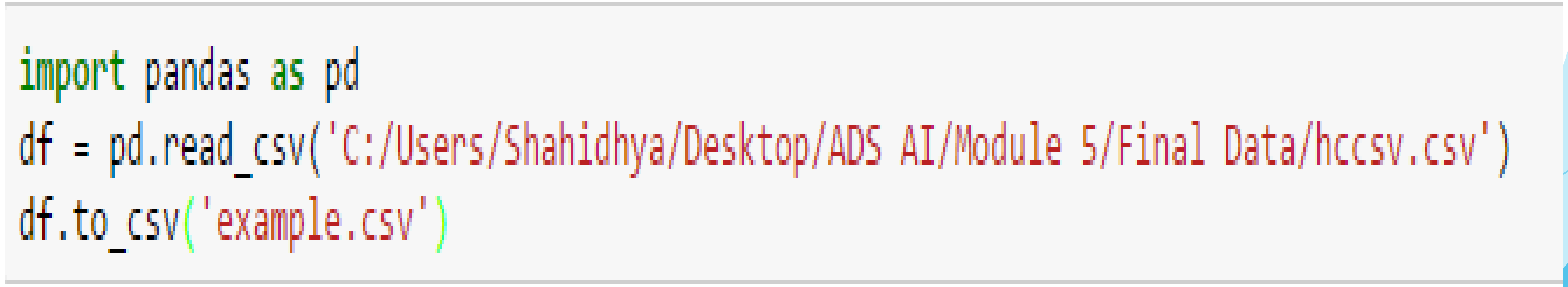
Pandas in Python

Reading CSV files -

- pandas.read_csv(file) : reads the csv file which can then be stored as pandas dataframe
- pandas.DataFrame.from_csv(file) : differs from read_csv in the indexing (takes first column as index by default)

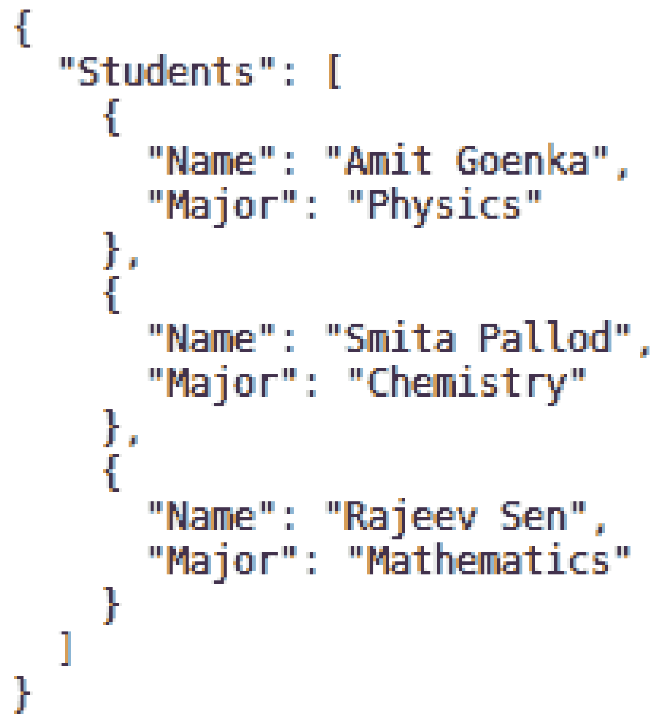
Writing CSV files -

- pandas.DataFrame.to_csv(file) : writes the contents of the data frame into a CSV file



```
import pandas as pd
df = pd.read_csv('C:/Users/Shahidhya/Desktop/ADS AI/Module 5/Final Data/hccsv.csv')
df.to_csv('example.csv')
```

JSON - Java Script Object Notation



```
{
  "Students": [
    {
      "Name": "Amit Goenka",
      "Major": "Physics"
    },
    {
      "Name": "Smita Pallod",
      "Major": "Chemistry"
    },
    {
      "Name": "Rajeev Sen",
      "Major": "Mathematics"
    }
  ]
}
```

- It is a lightweight data-interchange format
- It is language independent because though JSON uses JavaScript syntax, the JSON format is text only. Text can be used by any programming language
- It is a collection of name-value pairs (called objects) and it can be an ordered list of values
- An object is of the format { name : value}. Each object is separated from others by ','

JSON - Java Script Object Notation

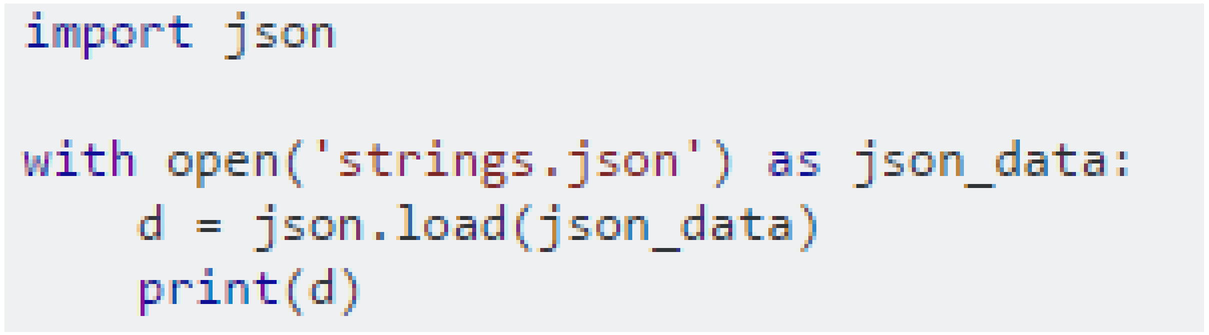
Reading and writing JSON

Reading JSON files -

- json.load(file) : reads the json file into python
- pandas.read_json(file) : reads the json file which can then be stored as pandas dataframe

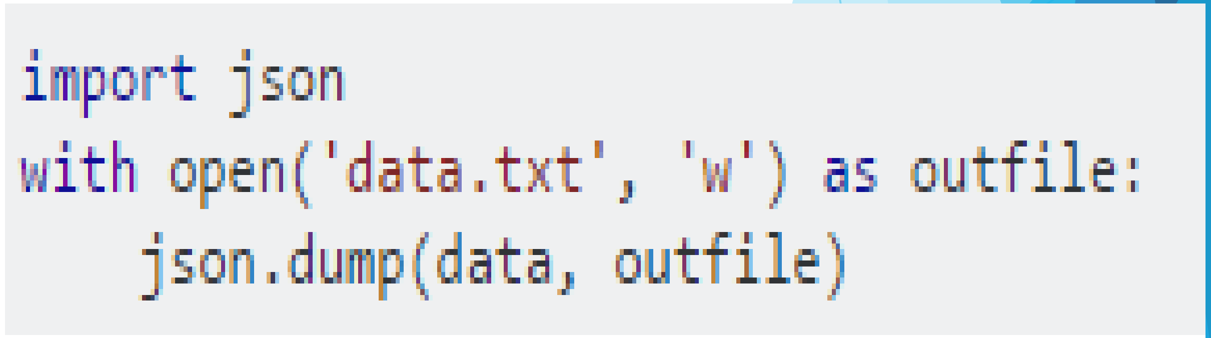
Writing JSON files -

- Json.dump(file) : writes the contents from Python into a JSON file
- pandas.DataFrame.to_json(file) : writes the contents of the data frame into a json file



```
import json

with open('strings.json') as json_data:
    d = json.load(json_data)
    print(d)
```



```
import json
with open('data.txt', 'w') as outfile:
    json.dump(data, outfile)
```