# Statistical Programming with Python

Pepe García

2020-04-20

# Statistical Programming with Python

CSV & JSON files

# Plan for this session

- Learn about CSV files
- Learn about JSON files

# Plan for this session

Clone session-9 repository, we will be using it today.

**https://github.com/mbd-statistical-programming-section-1/session-9**

# CSV files

Python comes with a **CSV** library that we can use out of the box. We use it by **importing** it. **Imports** are commonly added at the top of the file.

```
import csv
```

# CSV files

The **csv** library is based on the idea of readers and writers. One can read all lines in a file like so:

```
with open("file.csv") as f:
    reader = csv.reader(f)
    for line in reader:
        print(line)  #line will be a list here
```

first we open the file normally

Then we create a reader using **csv.reader()**

Finally, we operate with the reader

# CSV files

writing is not very different from reading:

```
lines = [
  ["asdf", "qwer"],
  ["hello", "world"]
]

with open("file.csv", "a") as f:
    writer = csv.writer(f)
    for line in lines:
        writer.writerow(line)
```

First we need some data to put in the csv file

Then we open the file with the append mode

Later, we create a **csv.writer**

# CSV files. Dictionaries

We can use specific writers for dictionaries!

```
beatles = [
    {"name": "John", "instrument": "voice"},
    {"name": "Paul", "instrument": "guitar"},
    {"name": "George", "instrument": "bass"},
    {"name": "Ringo", "instrument": "drums"}
]

with open("beatles.csv", "w") as my_file:
    writer = csv.DictWriter(my_file, ["name", "instrument"])
    writer.writeheader()
    for beatle in beatles:
        writer.writerow(beatle)
```

First we need some data to put in the csv file

# CSV files. Dictionaries

We can use specific readers too

```
with open("beatles.csv") as my_file:
    reader = csv.DictReader(my_file)
    for beatle in reader:
        print(beatle["name"] + " -> " + beatle["instrument"])
```

Then we open the file with the read mode (default)

Later, we create a **csv.DictReader**

Each element in the reader will be a dictionary already

# JSON files

Python also comes with a **JSON** library that we can use out of the box. We use it by **importing** it. **Imports** are commonly added at the top of the file.

```
import json
```

# JSON files

The **json** library can dump or load dictionaries into/from files

```
with open("input.json") as input_file:
    content = json.load(input_file)
    # Process content

with open("output.json", "w") as output_file:
    json.dump(content, output_file)
```