LECTURE 43

PANDAS

MCS 275 Spring 2021
Emily Dumas

LECTURE 43: PANDAS

Course bulletins:

- **Complete your course evaluations**
- Project 4 is due 6pm CDT Friday April 30.
- The project 4 autograder is now open.
- Install pandas with

  ```
  python3 -m pip install pandas
  ```

- Today's pandas intro notebook

PANDAS

Pandas is a module for working with **tabular data**, i.e. data in a 2D array where each column has a name and fixed data type (e.g. name "year", type integer).

id	name	atomic mass
14	Silicon	28.086
78	Platinum	195.084

In pandas, every row must have a unique identifier called its **index**. It can be just a 0-based index (the default) but other types are allowed (e.g. date/time).

TLDR

Pandas provides a data structure to *properly* represent the contents of a CSV or spreadsheet in Python.

or

```
pandas : csv :: bs4 : html.parser
```

PANDAS FEATURES

Excellent file format support: CSV, TSV, JSON, XLS, XLSX, SQL, SAS, HTML tables, ...

Searching, filtering, and transformation, with interface similar to numpy.

Excellent interoperability with numpy and matplotlib.

Wide user base, active development. (4 releases since MCS 275 started!)

DO I NEED THIS?

There are lots of *storage* formats for tabular data you might consider. Most have Python I/O modules.

Of these, only SQL provides the kind of advanced searching, filtering, etc., that pandas offers. However, SQL syntax is comparatively heavy and not tightly integrated with Python language. (But SQL is great!)

Another option is to just use a spreadsheet program. Scripts/notebooks offer better formalization, documentation, and reproducibility of analysis, though.

IN DEFENSE OF SQL

Pandas is for data analysis, usually by one person, working with data that can fit in memory of one machine. It does not specify a storage format or provide for concurrent access.

For persistent data that an application program will access in a predictable way, you should probably use SQL.

For exploration, visualization, cleaning, and transformation of a small dataset (a few GB max), pandas is an excellent choice.

TEMPLATE

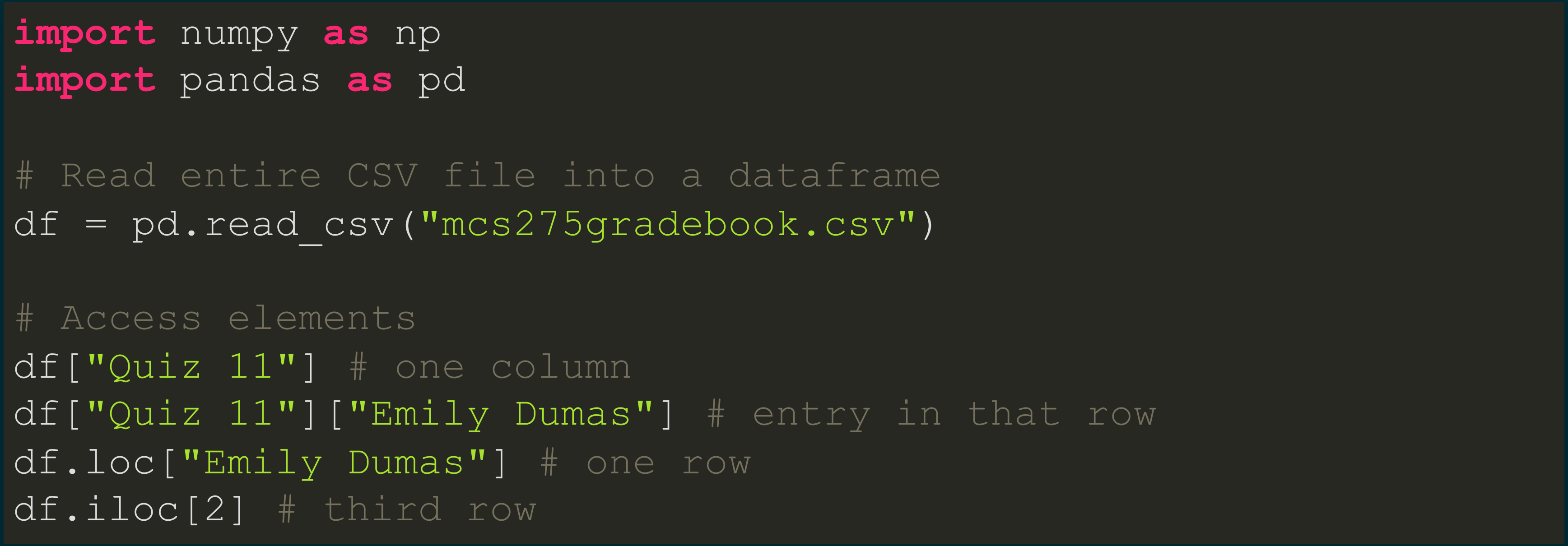
```python
import numpy as np
import pandas as pd

# and optionally
import matplotlib.pyplot as plt
```

CORE PANDAS CONCEPTS

- **index** - The unique identifier of a row.
- **`pd.Series`** - A single column of tabular data; behaves like a blend of numpy array (typed, fast) and a dictionary (index of arbitrary type).
- **`pd.DataFrame`** - A table with named, typed columns and ordered, indexed rows. Equivalent to a collection of series that all share the same index.

SPECIAL DATA TYPES

```
pd.Timestamp   # like datetime.datetime but can autoparse
pd.Timedelta   # like datetime.timedelta
pd.Categorical # enumerated type (fixed set of possible values)
```



```python
import numpy as np
import pandas as pd

# Read entire CSV file into a dataframe
df = pd.read_csv("mcs275gradebook.csv")

# Access elements
df["Quiz 11"] # one column
df["Quiz 11"]["Emily Dumas"] # entry in that row
df.loc["Emily Dumas"] # one row
df.iloc[2] # third row
```

REFERENCES

- Chapter 3 of *Python Data Science Handbook* by Jake VanderPlas
- pandas documentation

REVISION HISTORY

- 2021-04-28 Notebook link
- 2021-04-28 Initial publication