Effective Programming Practices for Economists

Data management with pandas

DataFrames and Series

Janoś Gabler and Hans-Martin von Gaudecker

What is a DataFrame

```
>>> df = pd.read_csv(path, engine="pyarrow")
>>> df
```

	country	continent	year	life_exp
0	Cuba	Americas	2002	77.16
1	Cuba	Americas	2007	78.27
2	Spain	Europe	2002	79.78
3	Spain	Europe	2007	80.94

```
>>> df.columns

Index(['country', 'continent', 'year', 'life_exp'],
dtype='string')

>>> df.index

RangeIndex(start=0, stop=4, step=1)
```

- Tabular dataset, typically loaded from a file
- Two mental models:
  1. Matrix/Array with labels
  2. Dictionary of columns
- Can inspect index and column names

What is a Series?

```
>>> sr = df["country"]
>>> type(sr)
pandas.core.series.Series

>>> sr

0     Cuba
1     Cuba
2    Spain
3    Spain
Name: country, dtype: string
```

- Each column of a DataFrame is a Series
- Mental model: Vector with an index
- All entries in a Series have the same dtype

Creating DataFrames and Series

```
>>> df = pd.DataFrame(
...     data=[[1, "bla"], [3, "blubb"]],
...     columns=["a", "b"],
...     index=["c", "d"]
... )
>>> df
```

	a	b
c	1	bla
d	3	blubb

```
>>> pd.Series(
...     [3.0, 4.5], index=["x", "y"],
... )

x    3.0
y    4.5
dtype: float64
```

- Data for a DataFrame can be nested lists or similar things
- Columns and index can be ints, strings and tuples
- **Powerful strategy**: Whenever you learn pandas or debug problems, create tiny DataFrames and Series to gain better understanding

Assignment is index aligned!

We continue using df from before

```
>>> sr = pd.Series(
...     [2.71, 3.14],
...     index=["d", "c"],
... )
>>> df["new_col"] = sr
```

	a	b	new_col
c	1	bla	3.14
d	3	blubb	2.71

- New columns can be assigned with square brackets
- Index is automatically aligned!
  - Makes many things safer!
  - Can make pandas slow