Applied Data Analytics

Pandas basics

Boolean indexing

Hans-Martin von Gaudecker and Aapo Stenhammar

Indexing

- Saw `.loc` and `.iloc` for indexing: Need to know the index value or position to access elements
- What if we wanted to access elements based on a condition? Examples
  - Values above 100
  - Values in the third quartile
  - Non-missing values

Example: Two Series

Name	Income
Alice	3000
Bob	750
Charlie	5000

Name	Female
Alice	True
Bob	False
Charlie	False

Select elements based on conditions

```
[1] income.loc[female]
[1] Alice    3000
    Name: Income, dtype: int64


[2] income.loc[[False, True, False]]
[2] Bob      750
    Name: Income, dtype: int64

[3] income.loc[[False, True]]
[3] ...
    IndexError: Boolean index has wrong
    length: 2 instead of 3
```

- Return the values where the Booleans used for indexing are `True`
- The Booleans must have the same length as the index
- If using a pd.Series to index, both indices must match exactly

Must use Booleans

```
[4] income.loc[1 - female]
[4] ---------------------------------------------------------------------------
    KeyError                                  Traceback (most recent call last)
    Cell In[4], line 1
    ----> 1 income.loc[1 - female]

    [...]

    KeyError: "None of [Index([0, 1, 1], dtype='int64')] are in the [index]"
```

Must use binary operators &, |, ~

```
[5] income.loc[not female]
    ---------------------------------------------------------------------------
    ValueError                                Traceback (most recent call last)
    Cell In[5], line 1
    ----> 1 income.loc[not female]

    [...]

    ValueError: The truth value of a Series is ambiguous. Use a.empty, a.bool(),
    a.item(), a.any() or a.all().

[6] income.loc[~female]
[6] Bob         750
    Charlie    5000
    Name: Income, dtype: int64
```