Programming Principles in Python (CSCI 503)

Data

Dr. David Koop
pandas

• Contains high-level data structures and manipulation tools designed to make data analysis fast and easy in Python

• Built on top of NumPy

• Built with the following requirements:
  - Data structures with labeled axes (aligning data)
  - Support time series data
  - Do arithmetic operations that include metadata (labels)
  - Handle missing data
  - Add merge and relational operations
Series

- A one-dimensional array (with a type) with an index
- Index defaults to numbers but can also be text (like a dictionary)
- Allows easier reference to specific items
- \( \text{obj} = \text{pd.Series}([7, 14, -2, 1]) \)
- Basically two arrays: \( \text{obj.values} \) and \( \text{obj.index} \)
- Can specify the index explicitly and use strings
- \( \text{obj2} = \text{pd.Series}([4, 7, -5, 3], \)
  \( \quad \text{index}=['d', 'b', 'a', 'c']) \)
- Kind of like fixed-length, ordered dictionary + can create from a dictionary
- \( \text{obj3} = \text{pd.Series}({'Ohio': 35000, 'Texas': 71000, \)
  \( \quad 'Oregon': 16000, 'Utah': 5000}) \)
Data Frame

- A dictionary of Series (labels for each series)
- A spreadsheet with row keys (the index) and column headers
- Has an index shared with each series
- Allows easy reference to any cell
- \[\text{df} = \text{DataFrame} \left\{ \begin{array}{l}
    'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada'], \\
    'pop': [1.5, 1.7, 3.6, 2.4] \\
\end{array} \right. \]

- Index is automatically assigned just as with a series but can be passed in as well via index kwarg
- Can reassign column names by passing columns kwarg
DataFrame Access and Manipulation

• df.values → 2D NumPy array

• Accessing a column:
  - df["<column>"]
  - df.<column>
  - Both return Series
  - Dot syntax only works when the column is a valid identifier

• Assigning to a column:
  - df["<column>"] = <scalar> # all cells set to same value
  - df["<column>"] = <array>    # values set in order
  - df["<column>"] = <series>  # values set according to match
    # between df and series indexes
```python
df = pd.read_csv('penguins_lter.csv')
```

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>36.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>339</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>340</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>341</td>
<td>122</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>342</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
<tr>
<td>343</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>49.9</td>
</tr>
</tbody>
</table>

344 rows x 17 columns
## Data Frame

```python
df = pd.read_csv('penguins_lter.csv')
```

### Column Names

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>36.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>122</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
<tr>
<td>PAL0910</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>49.9</td>
</tr>
</tbody>
</table>

344 rows x 17 columns
<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adelae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adelae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adelae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adelae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adelae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>36.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>122</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>49.9</td>
</tr>
</tbody>
</table>

344 rows x 17 columns
Data Frame

```python
import pandas as pd
df = pd.read_csv('penguins_lter.csv')
```

### Index

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td></td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td></td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td></td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td></td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td></td>
<td>36.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td></td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td></td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>122</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td></td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td></td>
<td>45.2</td>
</tr>
<tr>
<td>PAL0910</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td></td>
<td>49.9</td>
</tr>
</tbody>
</table>

344 rows x 17 columns

Column: `df['Island']`
## Data Frame

```python
df = pd.read_csv('penguins_lter.csv')
```

### Column Names

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>36.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>122</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
<tr>
<td>PAL0910</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>49.9</td>
</tr>
</tbody>
</table>

344 rows x 17 columns

### Row: `df.loc[2]`

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
</tbody>
</table>

### Index

Column: `df['Island']`
Data Frame

```
df = pd.read_csv('penguins_lter.csv')
```

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Cullmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>36.7</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
</tbody>
</table>

Row: `df.loc[2]`  
Cell: `df.loc[341,'Species']`  
Column: `df['Island']`

344 rows x 17 columns
Data Frame

```
import pandas as pd
df = pd.read_csv('penguins_lter.csv')
```

### Column Names

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Row: `df.loc[2]`

```
Row:
df.loc[2]
```

### Index

```
Index
```

### Cell: `df.loc[341,'Species']`

```
Cell: df.loc[341,'Species']
```

### Column: `df['Island']`

344 rows x 17 columns
Indexing

- Same as with NumPy arrays but can use Series's index labels
- Slicing with labels: NumPy is *exclusive*, Pandas is *inclusive*!
  - \( s = \text{Series}(\text{np.arange}(4)) \)
    \[ s[0:2] \] # gives two values like numpy
  - \( s = \text{Series}(\text{np.arange}(4), \text{index}=[\text{'a'}, \text{'b'}, \text{'c'}, \text{'d'}]) \)
    \[ s[\text{'a'}:${\text{'c'}}] \] # gives three values, not two!

- Obtaining data subsets
  - [ ]: get columns by label
  - loc: get rows/cols by label
  - iloc: get rows/cols by position (integer index)
  - For single cells (scalars), also have at and iat
Filtering

• Same as with numpy arrays but allows use of column-based criteria
  - `data[data < 5] = 0`
  - `data[data['three'] > 5]`

• `data < 5` → boolean data frame, can be used to select specific elements

• Multiple criteria, use `&`, `|`, and `~`; remember parentheses!
  - `data[(data['three'] > 5) & (data['two'] < 10)]`
### Filtering

```python
df[df['Culmen Length (mm)'] > 40]
```

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>36.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N3B1A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A2A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>122</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A2A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
<tr>
<td>PAL0910</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>49.9</td>
</tr>
</tbody>
</table>

344 rows × 17 columns
Filtering

\[ \text{df[df['Culmen Length (mm)'] > 40]} \]

<table>
<thead>
<tr>
<th>studyName</th>
<th>Sample Number</th>
<th>Species</th>
<th>Region</th>
<th>Island</th>
<th>Stage</th>
<th>Individual ID</th>
<th>Clutch Completion</th>
<th>Date Egg</th>
<th>Culmen Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL0708</td>
<td>1</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A1</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.1</td>
</tr>
<tr>
<td>PAL0708</td>
<td>2</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N1A2</td>
<td>Yes</td>
<td>11/11/07</td>
<td>39.5</td>
</tr>
<tr>
<td>PAL0708</td>
<td>3</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>40.3</td>
</tr>
<tr>
<td>PAL0708</td>
<td>4</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N2A2</td>
<td>Yes</td>
<td>11/16/07</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0708</td>
<td>5</td>
<td>Adelie Penguin (Pygoscelis adeliae)</td>
<td>Anvers</td>
<td>Torgersen</td>
<td>Adult, 1 Egg Stage</td>
<td>N3A1</td>
<td>Yes</td>
<td>11/16/07</td>
<td>36.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>PAL0910</td>
<td>120</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N38A2</td>
<td>No</td>
<td>12/1/09</td>
<td>NaN</td>
</tr>
<tr>
<td>PAL0910</td>
<td>121</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>46.8</td>
</tr>
<tr>
<td>PAL0910</td>
<td>122</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N39A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>50.4</td>
</tr>
<tr>
<td>PAL0910</td>
<td>123</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A1</td>
<td>Yes</td>
<td>11/22/09</td>
<td>45.2</td>
</tr>
<tr>
<td>PAL0910</td>
<td>124</td>
<td>Gentoo penguin (Pygoscelis papua)</td>
<td>Anvers</td>
<td>Biscoe</td>
<td>Adult, 1 Egg Stage</td>
<td>N43A2</td>
<td>Yes</td>
<td>11/22/09</td>
<td>49.9</td>
</tr>
</tbody>
</table>

344 rows × 17 columns
# Reading & Writing Data in Pandas

<table>
<thead>
<tr>
<th>Format</th>
<th>Data Description</th>
<th>Reader</th>
<th>Writer</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>CSV</td>
<td>read_csv</td>
<td>to_csv</td>
</tr>
<tr>
<td>text</td>
<td>Fixed-Width Text File</td>
<td>read_fwf</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>JSON</td>
<td>read_json</td>
<td>to_json</td>
</tr>
<tr>
<td>text</td>
<td>HTML</td>
<td>read_html</td>
<td>to_html</td>
</tr>
<tr>
<td>text</td>
<td>Local clipboard</td>
<td>read_clipboard</td>
<td>to_clipboard</td>
</tr>
<tr>
<td></td>
<td>MS Excel</td>
<td>read_excel</td>
<td>to_excel</td>
</tr>
<tr>
<td>binary</td>
<td>OpenDocument</td>
<td></td>
<td></td>
</tr>
<tr>
<td>binary</td>
<td>HDF5 Format</td>
<td>read_hdf</td>
<td>to_hdf</td>
</tr>
<tr>
<td>binary</td>
<td>Feather Format</td>
<td>read_feather</td>
<td>to_feather</td>
</tr>
<tr>
<td>binary</td>
<td>Parquet Format</td>
<td>read_parquet</td>
<td>to_parquet</td>
</tr>
<tr>
<td>binary</td>
<td>ORC Format</td>
<td>read_orc</td>
<td></td>
</tr>
<tr>
<td>binary</td>
<td>Msgpack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>binary</td>
<td>Stata</td>
<td>read_stata</td>
<td>to_stata</td>
</tr>
<tr>
<td>binary</td>
<td>SAS</td>
<td>read_sas</td>
<td></td>
</tr>
<tr>
<td>binary</td>
<td>SPSS</td>
<td>read_spss</td>
<td></td>
</tr>
<tr>
<td>binary</td>
<td>Python Pickle Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL</td>
<td>SQL</td>
<td>read_sql</td>
<td>to_sql</td>
</tr>
<tr>
<td>SQL</td>
<td>Google BigQuery</td>
<td>read_gbq</td>
<td>to_gbq</td>
</tr>
</tbody>
</table>

[https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html]
read_csv

- Convenient method to read csv files
- Lots of different options to help get data into the desired format
- Basic: \texttt{df = pd.read_csv(fname)}
- Parameters:
  - \texttt{path}: where to read the data from
  - \texttt{sep} (or \texttt{delimiter}): the delimiter (',', ' ', '	', 's+')
  - \texttt{header}: if \texttt{None}, no header
  - \texttt{index_col}: which column to use as the row index
  - \texttt{names}: list of header names (e.g. if the file has no header)
  - \texttt{skiprows}: number of list of lines to skip
Writing CSV data with pandas

- Basic: `df.to_csv(<fname>)`
- Change delimiter with sep kwarg:
  - `df.to_csv('example.dsv', sep='|')`
- Change missing value representation:
  - `df.to_csv('example.dsv', na_rep='NULL')`
- Don't write row or column labels:
  - `df.to_csv('example.csv', index=False, header=False)`
- Series may also be written to csv
Documentation

- pandas documentation is pretty good
- Lots of recipes on stackoverflow for particular data manipulations/queries
Assignment 7

- Downloading and unarchiving files
- File system manipulation
- Threading
- Basic Data Manipulation
- Due Friday
Derived Data

• Create new columns from existing columns
  - \( r["PctFail"] = r["Fail"] / r["Total"] \)

• Note that operations are computed in a vectorized manner

• Similarities to functional paradigm (map/filter):
  - specify the operation once
  - no loops
  - interpreted as an operation on the entire column
Aggregation

• Descriptive statistics
  - df['Culmen Length (mm)'].mean()
  - .median()
  - .describe()
  - .count()
  - .min(), .max()

• Also general methods
  - .sum()
  - .product()
Aggregation of time series data, a special use case of `groupby`, is referred to as `resampling` in this book and will receive separate treatment in Chapter 10.

**GroupBy Mechanics**

Hadley Wickham, an author of many popular packages for the R programming language, coined the term `split-apply-combine` for talking about group operations, and I think that's a good description of the process. In the first stage of the process, data contained in a pandas object, whether a Series, DataFrame, or otherwise, is split into groups based on one or more keys that you provide. The splitting is performed on a particular axis of an object. For example, a DataFrame can be grouped on its rows (`axis=0`) or its columns (`axis=1`). Once this is done, a function is applied to each group, producing a new value. Finally, the results of all those function applications are combined into a result object. The form of the resulting object will usually depend on what's being done to the data. See Figure 9-1 for a mockup of a simple group aggregation.

**Figure 9-1. Illustration of a group aggregation**

Each grouping key can take many forms, and the keys do not have to be all of the same type:

- A list or array of values that is the same length as the axis being grouped
- A value indicating a column name in a DataFrame

---

[W. McKinney, Python for Data Analysis]
Split-Apply-Combine

• Similar to Map (split+apply) Reduce (combine) paradigm
• The Pattern:
  1. **Split** the data by some grouping variable
  2. **Apply** some function to each group independently
  3. **Combine** the data into some output dataset
• The apply step is usually one of:
  - Aggregate
  - Transform
  - Filter
In Pandas

- **groupby** method creates a GroupBy object
- **groupby** doesn't actually compute anything until there is an apply/aggregate step or we wish to examine the groups
- Choose keys (columns) to group by
- `size()` is the count of each group
- Other aggregates also work
Split-Apply-Combine

- `df.groupby('Island')[['Culmen Length (mm)', 'Culmen Depth (mm)']].mean()`
- `df.groupby('Island').agg({'Culmen Length (mm)': 'mean', 'Culmen Depth (mm)': 'mean'})`
- `df.groupby('Island').agg({
    'Culmen Length (mm)': ('mean',),
    'Culmen Depth (mm)': ('mean',)
})`

<table>
<thead>
<tr>
<th>Island</th>
<th>cul_length</th>
<th>cul_depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biscoe</td>
<td>45.257485</td>
<td>15.874850</td>
</tr>
<tr>
<td>Dream</td>
<td>44.167742</td>
<td>18.344355</td>
</tr>
<tr>
<td>Torgersen</td>
<td>38.950980</td>
<td>18.429412</td>
</tr>
</tbody>
</table>
Different Data Layouts

<table>
<thead>
<tr>
<th></th>
<th>treatmenta</th>
<th>treatmentb</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>—</td>
<td>2</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>Mary Johnson</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Initial Data

<table>
<thead>
<tr>
<th></th>
<th>John Smith</th>
<th>Jane Doe</th>
<th>Mary Johnson</th>
</tr>
</thead>
<tbody>
<tr>
<td>treatmenta</td>
<td>—</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>treatmentb</td>
<td>2</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

Transpose

<table>
<thead>
<tr>
<th>name</th>
<th>trt</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>a</td>
<td>—</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>a</td>
<td>16</td>
</tr>
<tr>
<td>Mary Johnson</td>
<td>a</td>
<td>3</td>
</tr>
<tr>
<td>John Smith</td>
<td>b</td>
<td>2</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>b</td>
<td>11</td>
</tr>
<tr>
<td>Mary Johnson</td>
<td>b</td>
<td>1</td>
</tr>
</tbody>
</table>

Tidy Data

[H. Wickham, 2014]
Problem: Variables stored in both rows & columns

Mexico Weather, Global Historical Climatology Network

<table>
<thead>
<tr>
<th>id</th>
<th>year</th>
<th>month</th>
<th>element</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>1</td>
<td>tmax</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>1</td>
<td>tmin</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>2</td>
<td>tmax</td>
<td>—</td>
<td>27.3</td>
<td>24.1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>2</td>
<td>tmin</td>
<td>—</td>
<td>14.4</td>
<td>14.4</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>3</td>
<td>tmax</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>32.1</td>
<td></td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>3</td>
<td>tmin</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>14.2</td>
<td></td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>4</td>
<td>tmax</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>4</td>
<td>tmin</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>5</td>
<td>tmax</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>5</td>
<td>tmin</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

(a) Molten data

<table>
<thead>
<tr>
<th>id</th>
<th>date</th>
<th>tmax</th>
<th>tmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX17004</td>
<td>2010-01-30</td>
<td>27.8</td>
<td>14.5</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-02</td>
<td>27.3</td>
<td>14.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-03</td>
<td>24.1</td>
<td>14.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-11</td>
<td>29.7</td>
<td>13.4</td>
</tr>
</tbody>
</table>

(b) Tidy data

Table 11: Original weather dataset. There is a column for each possible day in the month. Columns d9 to d31 have been omitted to conserve space.

Table 12: (a) Molten weather dataset. This is almost tidy, but instead of values, the element column contains names of variables. Missing values are dropped to conserve space. (b) Tidy weather dataset. Each row represents the meteorological measurements for a single day. There are two measured variables, minimum (tmin) and maximum (tmax); all other variables are fixed.

[H. Wickham, 2014]
Problem: Variables stored in both rows & columns

<table>
<thead>
<tr>
<th>id</th>
<th>year</th>
<th>month</th>
<th>element</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>1</td>
<td>tmax</td>
<td></td>
<td></td>
<td>27.3</td>
<td>24.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>1</td>
<td>tmin</td>
<td></td>
<td></td>
<td>14.4</td>
<td>14.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>2</td>
<td>tmax</td>
<td></td>
<td></td>
<td>14.4</td>
<td>14.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>3</td>
<td>tmax</td>
<td></td>
<td></td>
<td>32.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>3</td>
<td>tmin</td>
<td></td>
<td></td>
<td>14.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>4</td>
<td>tmax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>4</td>
<td>tmin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>5</td>
<td>tmax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX17004</td>
<td>2010</td>
<td>5</td>
<td>tmin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Variable in columns: day; Variable in rows: tmax/tmin

Mexico Weather, Global Historical Climatology Network

[H. Wickham, 2014]
Solution: Melting + Pivot

<table>
<thead>
<tr>
<th>id</th>
<th>date</th>
<th>element</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX17004</td>
<td>2010-01-30</td>
<td>tmax</td>
<td>27.8</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-01-30</td>
<td>tmin</td>
<td>14.5</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-02</td>
<td>tmax</td>
<td>27.3</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-02</td>
<td>tmin</td>
<td>14.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-03</td>
<td>tmax</td>
<td>24.1</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-03</td>
<td>tmin</td>
<td>14.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-11</td>
<td>tmax</td>
<td>29.7</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-11</td>
<td>tmin</td>
<td>13.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-23</td>
<td>tmax</td>
<td>29.9</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-23</td>
<td>tmin</td>
<td>10.7</td>
</tr>
</tbody>
</table>

(a) Molten data

<table>
<thead>
<tr>
<th>id</th>
<th>date</th>
<th>tmax</th>
<th>tmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX17004</td>
<td>2010-01-30</td>
<td>27.8</td>
<td>14.5</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-02</td>
<td>27.3</td>
<td>14.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-03</td>
<td>24.1</td>
<td>14.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-11</td>
<td>29.7</td>
<td>13.4</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-02-23</td>
<td>29.9</td>
<td>10.7</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-03-05</td>
<td>32.1</td>
<td>14.2</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-03-10</td>
<td>34.5</td>
<td>16.8</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-03-16</td>
<td>31.1</td>
<td>17.6</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-04-27</td>
<td>36.3</td>
<td>16.7</td>
</tr>
<tr>
<td>MX17004</td>
<td>2010-05-27</td>
<td>33.2</td>
<td>18.2</td>
</tr>
</tbody>
</table>

(b) Tidy data

[H. Wickham, 2014]
Melt

- Want to keep each observation separate (tidy), aka pivot_longer

```r
df.melt(id_vars=["location", "Temperature"],
    var_name="Date", value_name="Value")
```
• Sometimes, we have data that is given in "long" format and we would like "wide" format (aka pivot_wider)

• Long format: column names are data values...

• Wide format: more like spreadsheet format

• Example:

<table>
<thead>
<tr>
<th>date</th>
<th>item</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1959-03-31</td>
<td>realgdp</td>
<td>2710.349</td>
</tr>
<tr>
<td>1 1959-03-31</td>
<td>infl</td>
<td>0.000</td>
</tr>
<tr>
<td>2 1959-03-31</td>
<td>unemp</td>
<td>5.800</td>
</tr>
<tr>
<td>3 1959-06-30</td>
<td>realgdp</td>
<td>2778.801</td>
</tr>
<tr>
<td>4 1959-06-30</td>
<td>infl</td>
<td>2.340</td>
</tr>
<tr>
<td>5 1959-06-30</td>
<td>unemp</td>
<td>5.100</td>
</tr>
<tr>
<td>6 1959-09-30</td>
<td>realgdp</td>
<td>2775.488</td>
</tr>
<tr>
<td>7 1959-09-30</td>
<td>infl</td>
<td>2.740</td>
</tr>
<tr>
<td>8 1959-09-30</td>
<td>unemp</td>
<td>5.300</td>
</tr>
<tr>
<td>9 1959-12-31</td>
<td>realgdp</td>
<td>2785.204</td>
</tr>
</tbody>
</table>

```python
date item   value
0 1959-03-31 realgdp 2710.349
1 1959-03-31 infl 0.000
2 1959-03-31 unemp 5.800
3 1959-06-30 realgdp 2778.801
4 1959-06-30 infl 2.340
5 1959-06-30 unemp 5.100
6 1959-09-30 realgdp 2775.488
7 1959-09-30 infl 2.740
8 1959-09-30 unemp 5.300
9 1959-12-31 realgdp 2785.204
```

.Pivot('date', 'item', 'value')

<table>
<thead>
<tr>
<th>item</th>
<th>infl</th>
<th>realgdp</th>
<th>unemp</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1959-03-31</td>
<td>0.00</td>
<td>2710.349</td>
<td>5.8</td>
</tr>
<tr>
<td>1959-06-30</td>
<td>2.34</td>
<td>2778.801</td>
<td>5.1</td>
</tr>
<tr>
<td>1959-09-30</td>
<td>2.74</td>
<td>2775.488</td>
<td>5.3</td>
</tr>
<tr>
<td>1959-12-31</td>
<td>0.27</td>
<td>2785.204</td>
<td>5.6</td>
</tr>
<tr>
<td>1960-03-31</td>
<td>2.31</td>
<td>2847.699</td>
<td>5.2</td>
</tr>
</tbody>
</table>

[W. McKinney, Python for Data Analysis]
Reshaping Data

- Reshape/pivoting are fundamental operations
- Can have a nested index in pandas
- Example: Congressional Districts (Ohio's 1st, 2nd, 3rd, Colorado's 1st, 2nd, 3rd) and associated representative rankings
- Could write this in different ways:

<table>
<thead>
<tr>
<th>number</th>
<th>one</th>
<th>two</th>
<th>three</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ohio</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Colorado</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state</th>
<th>number</th>
<th>Ohio</th>
<th>Colorado</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>one</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>two</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>three</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state</th>
<th>number</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ohio</td>
<td></td>
<td>one</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>two</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>three</td>
<td>2</td>
</tr>
<tr>
<td>Colorado</td>
<td></td>
<td>one</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>two</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>three</td>
<td>5</td>
</tr>
</tbody>
</table>
Reshaping Data

- Reshape/pivoting are fundamental operations
- Can have a nested index in pandas
- Example: Congressional Districts (Ohio's 1st, 2nd, 3rd, Colorado's 1st, 2nd, 3rd) and associated representative rankings
- Could write this in different ways:

<table>
<thead>
<tr>
<th>state</th>
<th>number</th>
<th>one</th>
<th>two</th>
<th>three</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ohio</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Colorado</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

MultiIndex

<table>
<thead>
<tr>
<th>state</th>
<th>number</th>
<th>Ohio</th>
<th>Colorado</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ohio</td>
<td>one</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>two</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>three</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Colorado</td>
<td>one</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>two</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>three</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Stack and Unstack

- **stack**: pivots from the columns into rows (may produce a Series!)
- **unstack**: pivots from rows into columns
- unstacking may add missing data
- stacking filters out missing data (unless `dropna=False`)
- can unstack at a different level by passing it (e.g. 0), defaults to innermost level

[W. McKinney, Python for Data Analysis]
String Methods

- Can do many of the same methods used for single strings on entire columns
- Requires `.str` prefix before calling the method
  - violations.value.str.strip().str.split(' - Comments: ')
- Also helps when extracting from a list
  - comments.str[1]
## String Methods

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>Return the number of non-overlapping occurrences of substring in the string.</td>
</tr>
<tr>
<td>endswith</td>
<td>Returns True if string ends with suffix.</td>
</tr>
<tr>
<td>startswith</td>
<td>Returns True if string starts with prefix.</td>
</tr>
<tr>
<td>join</td>
<td>Use string as delimiter for concatenating a sequence of other strings.</td>
</tr>
<tr>
<td>index</td>
<td>Return position of first character in substring if found in the string; raises ValueError if not found.</td>
</tr>
<tr>
<td>find</td>
<td>Return position of first character of first occurrence of substring in the string; like \index, but returns –1 if not found.</td>
</tr>
<tr>
<td>rfind</td>
<td>Return position of first character of last occurrence of substring in the string; returns –1 if not found.</td>
</tr>
<tr>
<td>replace</td>
<td>Replace occurrences of string with another string.</td>
</tr>
<tr>
<td>strip, rstrip, lstrip</td>
<td>Trim whitespace, including newlines; equivalent to x.strip() and rstrip, lstrip, respectively</td>
</tr>
<tr>
<td>split</td>
<td>Break string into list of substrings using passed delimiter.</td>
</tr>
<tr>
<td>lower</td>
<td>Convert alphabet characters to lowercase.</td>
</tr>
<tr>
<td>upper</td>
<td>Convert alphabet characters to uppercase.</td>
</tr>
<tr>
<td>casefold</td>
<td>Convert characters to lowercase, and convert any region-specific variable character combinations to a common comparable form.</td>
</tr>
<tr>
<td>ljust, rjust</td>
<td>Left justify or right justify, respectively; pad opposite side of string with spaces (or some other fill character) to return a string with a minimum width.</td>
</tr>
</tbody>
</table>

Regular Expressions

Regular expressions provide a flexible way to search or match (often more complex) string patterns in text. A single expression, commonly called a regex, is a string formed according to the regular expression language. Python's built-in `re` module is responsible for applying regular expressions to strings; I'll give a number of examples of its use here.

The `re` module functions fall into three categories: pattern matching, substitution, and splitting. Naturally these are all related; a regex describes a pattern to locate in the text, which can then be used for many purposes. Let's look at a simple example:
Support for Datetime

• Python has datetime library to support dates and times
• pandas has a Timestamp data type that functions somewhat similarly
• Pandas can convert timestamps
  - `pd.to_datetime`: versatile, can often guess format
• Like string methods, also a `.dt` accessor for datetime methods/properties
• With a timestamp, filtering based on datetimes becomes easier
  - `df[df['Inspection Date'] > '2021']`
Food Inspections Example