Programming Principles in Python (CSCI 503/490)

Data

Dr. David Koop

Arrays

- Usually a fixed size—lists are meant to change size

- Are mutable—tuples are not

- Store only one type of data—lists and tuples can store anything

- Are faster to access and manipulate than lists or tuples

- Can be multidimensional:

  - Can have list of lists or tuple of tuples but no guarantee on shape

  - Multidimensional arrays are rectangles, cubes, etc.

NumPy Arrays

- import numpy as np
- Creating:
  - `data1 = [6, 7, 8, 0, 1]`
  - `arr1 = np.array(data1)`
  - `arr1_float = np.array(data1, dtype='float64')`
  - `np.ones((4,2)) # 2d array of ones`
  - `arr1_ones = np.ones_like(arr1) # [1, 1, 1, 1, 1]`
- Type and Shape Information:
  - `arr1.dtype # int64 # type of values stored in array`
  - `arr1.ndim # 1 # number of dimensions`
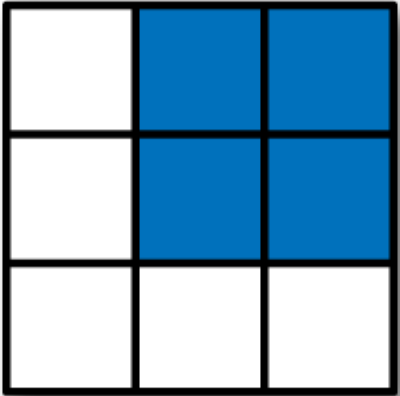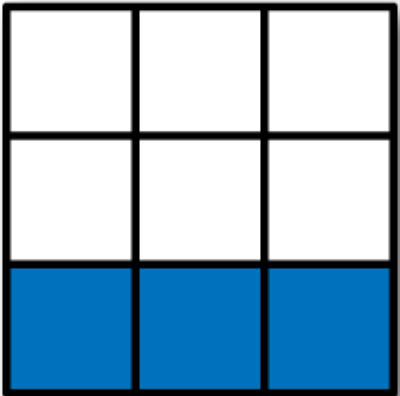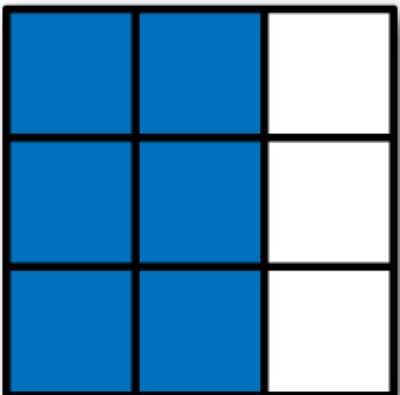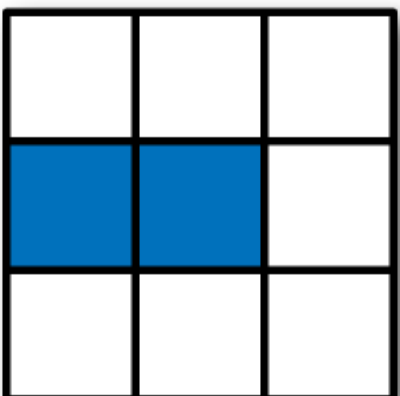 - `arr1.shape # (5,) # shape of the array`

Array Operations

- ```
  a = np.array([1,2,3])
  b = np.array([6,4,3])
  ```

- (Array, Array) Operations (**Element-wise**)

  - Addition, Subtraction, Multiplication

  - ```
    a + b # array([7, 6, 6])
    ```

- (Scalar, Array) Operations (**Broadcasting**):

  - Addition, Subtraction, Multiplication, Division, Exponentiation

  - ```
    a ** 2 # array([1, 4, 9])
    ```

  - ```
    b + 3 # array([9, 7, 6])
    ```

Indexing

- Same as with lists plus shorthand for 2D+

  - `arr1 = np.array([6, 7, 8, 0, 1])`

  - `arr1[1]`

  - `arr1[-1]`

- What about two dimensions?

  - `arr2 = np.array([[1.5,2,3,4],[5,6,7,8]])`

  - `arr[1][1]`

  - `arr[1,1] # shorthand`

numpy Array Slicing

- Indexing is similar to lists
  - Even in 2D
  - `arr[2][2]` same as `arr[2,2]`

- Slicing is a bit different:
  - Slices are **views**
  - Dimensionality unchanged with pure slicing
  - `arr[1:3][:2] != arr[1:3,:2]`

Expression	Shape
`arr[:2, 1:]`	(2, 2)
`arr[2]`	(3,)
`arr[2, :]`	(3,)
`arr[2:, :]`	(1, 3)
`arr[:, :2]`	(3, 2)
`arr[1, :2]`	(2,)
`arr[1:2, :2]`	(1, 2)

[W. McKinney, Python for Data Analysis]

Assignment 7

- Illinois Unemployment Data

- Downloading and uncompressing files

- Finding files using OS libraries

- Use a match statement to process data

- Store per-county dataframes, each in a csv file

Quiz Wednesday

Array Transformations

- Transpose
  - `arr2.T # flip rows and columns`
- Stacking: take iterable of arrays and stack them horizontally/vertically
  - `arrh1 = np.arange(3)`
  - `arrh2 = np.arange(3,6)`
  - `np.vstack([arrh1, arrh2])`
  - `np.hstack([arr1.T, arr2.T]) # ???`

Boolean Indexing

- `names == 'Bob'` gives back booleans that represent the element-wise comparison with the array `names`

- Boolean arrays can be used to index into another array:
  - `data[names == 'Bob']`

- Can even mix and match with integer slicing

- Can do boolean operations (`&`, `|`) between arrays (just like addition, subtraction)
  - `data[(names == 'Bob') | (names == 'Will')]`

- Note: `or` and `and` do not work with arrays

- We can set values too! `data[data < 0] = 0`

pandas

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

- Originally 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

polars

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

  - Built using Apache Arrow

  - Written from scratch using Rust but with a Python API

  - Parallelized (uses multiple cores)

  - Intuitive API

Code Conventions

- Universal:
  - `import pandas as pd`
  - `import polars as pl`

- Also used:
  - `from pandas import Series, DataFrame`
  - `from polars import Series, DataFrame`

polars Series

- A one-dimensional data structure (with a type)
  - `s = pl.Series([1,2,3])`
- May also have a name
  - `s = pl.Series('name',['a','b','c'])`
- Just like numpy arrays, a series has a dtype
  - `s = pl.Series('name',[1,2,3],dtype=pl.Float)`
- Indexing:
  - `s[0]  # 1.0`

pandas Series

- A one-dimensional array (with a type)
  - `t = pd.Series([1,2,3])`

- May also have a name:
  - `t = pd.Series([1,2,3], name='num')`

- Just like numpy arrays, a series has a dtype
  - `t = pd.Series([1,2,3], name='num', dtype='float')`

- Indexing: `t[0]`

- ...but a panads Series also has an **index** (polars does not)

pandas Series and the Index

- pandas Series is a one-dimensional array (with a type) **plus an index**
- Basically two arrays: `t.values` and `t.index`
  - `obj.index # [0, 1, 2]`
- Can specify the index explicitly (could be strings)
  - `t = pd.Series([1,2,3],['a','b','c'])`
- Kind of like fixed-length, ordered dictionary + can create from a dictionary
  - `t = pd.Series({'a': 1, 'b': 2, 'c': 3})`
- Indexing:
  - `t['a']`
  - What about `t[0]`?

polars Series Operations

- Can do binary operations with two Series
- Just like numpy, between two Series, these are **elementwise**
  - `pl.Series([1,2,3]) + pl.Series([1,2,3]) # pl.Series([2,4,6])`
- Between a Series and a scalar, this is **broadcast**
  - `pl.Series([1,2,3]) + 4 # pl.Series([5,6,7])`
- Have to have the same number of elements
  - `pl.Series([1,2,3]) + pl.Series([1,2,3,4]) # Error`
- Also works with non-numeric operations:
  - `pl.Series(['a','b']) + pl.Series(['c','d'])`

pandas Series Operations

- Same as polars
  - `pd.Series([1,2,3]) + pd.Series([1,2,3]) # pd.Series([2,4,6])`
  - `pd.Series([1,2,3]) + 4 # pd.Series([5,6,7])`

- …but with custom indexes, the operations **align**:
  - `pd.Series([1,2,3],index=list('abc') +`
    `pd.Series([1,2,3],index=list('cba')`
    `# => pd.Series([4,4,4], index=['a','b','c'])`

```
In [28]: obj3          In [29]: obj4              In [30]: obj3 + obj4
Out[28]:               Out[29]:                   Out[30]:
Ohio      35000        California      NaN        California      NaN
Oregon    16000        Ohio          35000        Ohio          70000
Texas     71000        Oregon        16000        Oregon        32000
Utah       5000        Texas         71000        Texas        142000
dtype: int64           dtype: float64             Utah            NaN
                                                  dtype: float64
```

[W. McKinney, Python for Data Analysis]

pandas Series Operations

- Missing labels lead to `NaN` (not a number) values

```
In [28]: obj3          In [29]: obj4              In [30]: obj3 + obj4
Out[28]:               Out[29]:                   Out[30]:
Ohio        35000      California      NaN        California        NaN
Oregon      16000      Ohio          35000        Ohio            70000
Texas       71000      Oregon        16000        Oregon          32000
Utah         5000      Texas         71000        Texas          142000
dtype: int64           dtype: float64             Utah              NaN
                                                  dtype: float64
```

- also have `.add`, `.subtract`, … that allow `fill_value` argument
- `obj3.add(obj4, fill_value=0)`

DataFrame

- A collection of Series (uniquely named)
  - Similar to a table in a database
  - Similar to a sheet in a spreadsheet
- ```
  df = DataFrame({'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada'],
                  'year': [2000, 2001, 2002, 2001],
                  'pop': [1.5, 1.7, 3.6, 2.4]})
  ```
- In pandas:
  - Has an index shared with each series
  - Index is automatically assigned just as with a series but can be passed in as well via `index` kwarg

pandas DataFrame Constructor Inputs

Type	Notes
2D ndarray	A matrix of data, passing optional row and column labels
dict of arrays, lists, or tuples	Each sequence becomes a column in the DataFrame. All sequences must be the same length.
NumPy structured/record array	Treated as the "dict of arrays" case
dict of Series	Each value becomes a column. Indexes from each Series are unioned together to form the result's row index if no explicit index is passed.
dict of dicts	Each inner dict becomes a column. Keys are unioned to form the row index as in the "dict of Series" case.
list of dicts or Series	Each item becomes a row in the DataFrame. Union of dict keys or Series indexes become the DataFrame's column labels
List of lists or tuples	Treated as the "2D ndarray" case
Another DataFrame	The DataFrame's indexes are used unless different ones are passed
NumPy MaskedArray	Like the "2D ndarray" case except masked values become NA/missing in the DataFrame result

[W. McKinney, Python for Data Analysis]

DataFrame Columns

- Access:
  - polars: `df['state']`
  - pandas: `dfa['state']` or `dfa.state` (doesn't always work!)
- Modification:
  - polars: `df.with_columns(pl.Series('state',`
    `['Ohio','Ohio','Texas','Nevada'))`
  - pandas: `df.assign(state=['Ohio','Ohio','Texas','Nevada'])`
  - Both create **new** data frames
  - pandas: `df['state'] = ['Ohio','Ohio','Texas','Nevada']`
  - This **mutates** the dataframe but causes problems so avoid it!

DataFrame Multiple Columns

- polars:
  - `df.select('state','year')`
- pandas:
  - `df[['state','year']]`
  - Not a new operator! It is a subscript where the argument is a list

DataFrame Indexing and Slicing

- polars:
  - `df[0], df[0:1]` # equivalent, data frame with single row
- pandas:
  - `dfa[0]` # error
  - `dfa.loc[0]` # a Series!
  - `dfa[0:2]` # a data frame with two rows
- pandas with an index (`dfi = dfa.set_index('state')`)
  - `dfi['Texas'], dfi['Ohio']` # a Series, a DataFrame!
  - `dfi.loc['Ohio':'Texas']` # inclusive slice!
  - `dfi.iloc[0:2]` # not inclusive!

pandas DataFrame Indexing and Slicing

- Same as with NumPy arrays but can use index labels

- Slicing with labels: NumPy is **exclusive**, Pandas is **inclusive**!

```
- s = Series(np.arange(4))
  s[0:2] # gives two values like numpy
```

```
- s = Series(np.arange(4), index=['a', 'b', 'c', 'd'])
  s['a':'c'] # gives three values, not two!
```

- Obtaining data subsets

  - `loc`: get rows/cols by label

  - `iloc`: get rows/cols by position (integer index)

DataFrame Filtering

- polars:
  - `df['pop'] > 2 # boolean Series`
  - `df.filter(pl.col('pop') > 2) # subset of dataframe`
- pandas:
  - `dfa['pop'] > 2 # boolean Series`
  - `dfa[dfa['pop'] > 2] # subset of dataframe`
  - `dfa.query('pop > 2') # subset of dataframe`
- Multiple criteria, use `&`, `|`, and `~`; remember parentheses!
  - `df.filter((pl.col('year') < 2002) & (pl.col('pop') > 2))`
  - `dfa[(dfa['year'] < 2002) & (dfa['pop'] > 2)]`

Sorting

- polars: `df.sort('pop')`

- pandas: `dfa.sort_values('pop')`

- Can sort by multiple columns, too

- pandas also has a `sort_index` method to sort by the index

  - `dfa.sort_index()`

Statistics

- Many common statistical methods can be used (min, max, median, etc.)
- `describe`: shortcut for easy stats!

```
In [204]: df.describe()
Out[204]:
            one        two
count  3.000000   2.000000
mean   3.083333  -2.900000
std    3.493685   2.262742
min    0.750000  -4.500000
25%    1.075000  -3.700000
50%    1.400000  -2.900000
75%    4.250000  -2.100000
max    7.100000  -1.300000
```

```
In [205]: obj = Series(['a', 'a', 'b', 'c'] * 4)

In [206]: obj.describe()
Out[206]:
count      16
unique      3
top         a
freq        8
dtype: object
```

Unique Values and Value Counts

- polars: `unique()` returns a Series/DataFrame with duplicates dropped

- pandas is more complicated

  - Series `unique()` returns an array with only the unique values (no index)

    - `s = Series(['c','a','d','a','a','b','b','c','c'])`
      `s.unique() # array(['c', 'a', 'd', 'b'])`

  - Data Frame `drop_duplicates` returns a DataFrame with duplicates dropped

- Also `nunique()/n_unique()` to count number of unique entries

- `value_counts` returns a Series/DataFrame with index frequencies:

  - `s.value_counts() # Series({'c': 3,'a': 3,'b': 2,'d': 1})`

pandas DataFrame

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

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

344 rows × 17 columns

pandas DataFrame

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

Column Names

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

344 rows × 17 columns

pandas DataFrame

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

Column Names

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

Index

344 rows × 17 columns

pandas DataFrame

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

Column Names

Index

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

344 rows × 17 columns

Column: `df['Island']`

pandas DataFrame

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

Column Names

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

Row: `df.loc[2]`

Index

344 rows × 17 columns

Column: `df['Island']`

pandas DataFrame

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

Column Names

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

Row: `df.loc[2]`

Index

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

344 rows × 17 columns

Column: `df['Island']`

pandas DataFrame


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

Column Names

Row: `df.loc[2]`

Index

Missing Data

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

Column: `df['Island']`

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

344 rows × 17 columns

polars DataFrame

shape: (344, 10)

studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)
str	i64	str	str	str	str	str	str	str	f64
"PAL0708"	1	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A1"	"Yes"	"11/11/07"	39.1
"PAL0708"	2	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A2"	"Yes"	"11/11/07"	39.5
"PAL0708"	3	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A1"	"Yes"	"11/16/07"	40.3
"PAL0708"	4	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A2"	"Yes"	"11/16/07"	null
"PAL0708"	5	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N3A1"	"Yes"	"11/16/07"	36.7
…	…	…	…	…	…	…	…	…	…
"PAL0910"	120	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N38A2"	"No"	"12/1/09"	null
"PAL0910"	121	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A1"	"Yes"	"11/22/09"	46.8
"PAL0910"	122	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A2"	"Yes"	"11/22/09"	50.4
"PAL0910"	123	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A1"	"Yes"	"11/22/09"	45.2
"PAL0910"	124	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A2"	"Yes"	"11/22/09"	49.9

polars DataFrame

Column Names
& Types

shape: (344, 10)

studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)
str	i64	str	str	str	str	str	str	str	f64
"PAL0708"	1	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A1"	"Yes"	"11/11/07"	39.1
"PAL0708"	2	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A2"	"Yes"	"11/11/07"	39.5
"PAL0708"	3	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A1"	"Yes"	"11/16/07"	40.3
"PAL0708"	4	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A2"	"Yes"	"11/16/07"	null
"PAL0708"	5	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N3A1"	"Yes"	"11/16/07"	36.7
...	...	...	...	...	...	...	...	...	...
"PAL0910"	120	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N38A2"	"No"	"12/1/09"	null
"PAL0910"	121	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A1"	"Yes"	"11/22/09"	46.8
"PAL0910"	122	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A2"	"Yes"	"11/22/09"	50.4
"PAL0910"	123	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A1"	"Yes"	"11/22/09"	45.2
"PAL0910"	124	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A2"	"Yes"	"11/22/09"	49.9

polars DataFrame

Column Names & Types

shape: (344, 10)

studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)
str	i64	str	str	str	str	str	str	str	f64
"PAL0708"	1	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A1"	"Yes"	"11/11/07"	39.1
"PAL0708"	2	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A2"	"Yes"	"11/11/07"	39.5
"PAL0708"	3	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A1"	"Yes"	"11/16/07"	40.3
"PAL0708"	4	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A2"	"Yes"	"11/16/07"	null
"PAL0708"	5	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N3A1"	"Yes"	"11/16/07"	36.7
...	...	...	...	...	...	...	...	...	...
"PAL0910"	120	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N38A2"	"No"	"12/1/09"	null
"PAL0910"	121	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A1"	"Yes"	"11/22/09"	46.8
"PAL0910"	122	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A2"	"Yes"	"11/22/09"	50.4
"PAL0910"	123	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A1"	"Yes"	"11/22/09"	45.2
"PAL0910"	124	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"					49.9

Column: `df['Island']`

polars DataFrame

Column Names
& Types

Row: `df[2]`

shape: (344, 10)

studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)
str	i64	str	str	str	str	str	str	str	f64
"PAL0708"	1	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A1"	"Yes"	"11/11/07"	39.1
"PAL0708"	2	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A2"	"Yes"	"11/11/07"	39.5
"PAL0708"	3	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A1"	"Yes"	"11/16/07"	40.3
"PAL0708"	4	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A2"	"Yes"	"11/16/07"	null
"PAL0708"	5	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N3A1"	"Yes"	"11/16/07"	36.7
...	...	...	...	...	...	...	...	...	...
"PAL0910"	120	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N38A2"	"No"	"12/1/09"	null
"PAL0910"	121	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A1"	"Yes"	"11/22/09"	46.8
"PAL0910"	122	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A2"	"Yes"	"11/22/09"	50.4
"PAL0910"	123	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A1"	"Yes"	"11/22/09"	45.2
"PAL0910"	124	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"					49.9

Column: `df['Island']`

polars DataFrame

shape: (344, 10)

Column Names & Types

Row: `df[2]`

Cell: `df['Species'][341]`

Column: `df['Island']`

studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)
str	i64	str	str	str	str	str	str	str	f64
"PAL0708"	1	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A1"	"Yes"	"11/11/07"	39.1
"PAL0708"	2	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A2"	"Yes"	"11/11/07"	39.5
"PAL0708"	3	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A1"	"Yes"	"11/16/07"	40.3
"PAL0708"	4	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A2"	"Yes"	"11/16/07"	null
"PAL0708"	5	"Adelie Penguin (Pygoscelis ade...	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N3A1"	"Yes"	"11/16/07"	36.7
...	...	...	...	...	...	...	...	...	...
"PAL0910"	120	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N38A2"	"No"	"12/1/09"	null
"PAL0910"	121	Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A1"	"Yes"	"11/22/09"	46.8
"PAL0910"	122	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A2"	"Yes"	"11/22/09"	50.4
"PAL0910"	123	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A1"	"Yes"	"11/22/09"	45.2
"PAL0910"	124	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"				49.9

polars DataFrame

shape: (344, 10)

Column Names & Types

Row: `df[2]`

Cell: `df['Species'][341]`

studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)
str	i64	str	str	str	str	str	str	str	f64
"PAL0708"	1	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A1"	"Yes"	"11/11/07"	39.1
"PAL0708"	2	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N1A2"	"Yes"	"11/11/07"	39.5
"PAL0708"	3	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A1"	"Yes"	"11/16/07"	40.3
"PAL0708"	4	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N2A2"	"Yes"	"11/16/07"	null
"PAL0708"	5	"Adelie Penguin (Pygoscelis ade…	"Anvers"	"Torgersen"	"Adult, 1 Egg Stage"	"N3A1"	"Yes"	"11/16/07"	…
…	…	…	…	…	…	…	…	…	…
"PAL0910"	120	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N38A2"	"No"	"12/1/09"	null
"PAL0910"	121	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A1"	"Yes"	"11/22/09"	46.8
"PAL0910"	122	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N39A2"	"Yes"	"11/22/09"	50.4
"PAL0910"	123	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N43A1"	"Yes"	"11/22/09"	45.2
"PAL0910"	124	"Gentoo penguin (Pygoscelis pap…	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"				49.9

Missing Data

Column: `df['Island']`

pandas Filtering

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

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

344 rows × 17 columns

pandas Filtering

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

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

344 rows × 17 columns

polars Filtering

```
df.filter(pl.col('Culmen Length (mm)') > 40)
```

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

344 rows × 17 columns

polars Filtering

```
df.filter(pl.col('Culmen Length (mm)') > 40)
```

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

344 rows × 17 columns

Reading and Writing CSV Files

- polars
  - `df = pl.read_csv(<fname>)`
  - `df.write_csv(<fname>)`
- pandas
  - `dfa = pd.read_csv(<fname>)`
  - `dfa.to_csv(<fname>)`
- Many options available!

Reading & Writing Data in Pandas

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

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

pandas read_csv

- Convenient method to read csv files
- Lots of different options to help get data into the desired format
- Basic: `dfa = pd.read_csv(fname)`
- Parameters:
  - `path`: where to read the data from
  - `sep` (or `delimiter`): the delimiter (`','`, `' '`, `'\t'`, `'\s+'`)
  - `header`: if `None`, no header
  - `index_col`: which column to use as the row index
  - `names`: list of header names (e.g. if the file has no header)
  - `skiprows`: number of list of lines to skip

Writing CSV data with pandas

- Basic: `dfa.to_csv(<fname>)`

- Change delimiter with sep kwarg:

  - `dfa.to_csv('example.dsv', sep='|')`

- Change missing value representation

  - `dfa.to_csv('example.dsv', na_rep='NULL')`

- Don't write row or column labels:

  - `dfa.to_csv('example.csv', index=False, header=False)`

- Series may also be written to csv