Programming Principles in Python (CSCI 503/490)

Data

Dr. David Koop

Quiz

Quiz

1. Evaluate `pd.Series([1,2,3]) + pd.Series([3,2,1],[2,1,0])`.
   (a) `pd.Series([2,4,6],[0,1,2])`
   (b) `pd.Series([4,4,4],[0,1,2])`
   (c) `pd.Series([1,2,3],[0,1,2])`
   (d) There is an error.

Quiz

2. Given the array `arr = np.array([[1,2,3],[4,5,6]])`, what is `arr[:,1].shape`?
   (a) `(2,)`
   (b) `(1,3)`
   (c) `(2,1)`
   (d) `(1,2)`

Quiz

3. Which of the following is not a difference between numpy arrays and python lists?
   (a) Arrays are mutable; lists are not
   (b) Arrays require that all elements have the same type; lists do not
   (c) Array slices are views over the original array; list slices are not views
   (d) Arrays are faster to access than lists

Quiz

4. Which is not a valid case in a match statement?
   (a) case ("abc" & "def")
   (b) case ("abc" | "def")
   (c) case {"abc": v}
   (d) case [_, "ab", *fnames]

Quiz

5. Which of the following is not a Python library used for manipulating data?
   (a) numpy
   (b) pandas
   (c) polars
   (d) grizzlies

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

Series

- A one-dimensional data structure (with a type)
  - `s = pl.Series([1,2,3])`
  - `t = pd.Series([1,2,3])`
- May also have a name and dtype
  - `s = pl.Series('name',['a','b','c'],dtype=pl.Float)`
  - `t = pd.Series([1,2,3], name='num',dtype='float')`
- In pandas, a series has an index
  - `ti = pd.Series([1,2,3],['a','b','c']) # index ['a','b','c']`
  - `ti = pd.Series({'a': 1, 'b': 2, 'c': 3}) # same index`
- Indexing: `s[0], t[0], ti['a'], ti.iloc[0], ti.loc['a']`

Series Operations

- Like numpy: elementwise / broadcasting
  - `Series([1,2,3]) + Series([1,2,3]) # Series([2,4,6])`
  - `Series([1,2,3]) + 4 # Series([5,6,7])`
- …but for pandas, with custom indexes, the operations **align** on the index:
  - ```
    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'])
    ```
  - also have `.add`, `.subtract`, … with `fill_value` argument

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

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

```
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

```
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

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

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

Column: `df['Island']`

pandas DataFrame

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

Column Names

Row: `df.loc[2]`

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

Row: `df.loc[2]`

Index

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

	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

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']`

	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
			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']`

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]`

Column: `df['Island']`

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

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
		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

Row: `df[2]`

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

Column: `df['Island']`

polars DataFrame

Column Names & Types

Row: `df[2]`

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

Missing Data

Column: `df['Island']`

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"	
…	…	…	…	…	…	…	…	…	…
"PAL0910"	120	"Gentoo penguin (Pygoscelis pap...	"Anvers"	"Biscoe"	"Adult, 1 Egg Stage"	"N38A2"	"No"	"12/1/09"	null
		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

Filtering

- polars: `df.filter(pl.col('Culmen Length (mm)') > 40)`
- pandas: `dfa[dfa['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

Assignment 7

- Downloading and uncompressing files
- Finding files using OS libraries
- Use a match statement to process data
- Can use polars or pandas
- Store per-year dataframes, each in a csv file

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})`
      ```

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

Missing Data

- polars: shows `null`
- pandas: shows `NaN` (or `NA` or `None` depending on dtype)
- Checking if missing:
  - polars: `pl.col('pop').is_null(), .is_not_null()`
  - pandas: `dfa['pop'].isnull(), .notnull()`
- Drop missing data:
  - polars: `pl.col('pop').drop_nulls()`, pandas: `dfa['pop'].dropna()`
- Filling in missing data:
  - polars: `pl.col('pop').fill_null()`, (`forward, backward, max,`…)
  - pandas: `dfa['pop'].fillna()`, now `ffill(), bfill()`

Derived Data

- Create new columns from existing columns
- pandas
  - `dfa["CulmenRatio"] = dfa['CLength'] / dfa['CDepth'] # Mut!`
  - `dfa = dfa.assign(CulmenRatio= dfa['CLength'] / dfa['CDepth'])`
- polars
  - ```
    df.with_columns(
              (df['CLength'] / df['CDepth']).alias('CulmenRatio'))
    ```
- Note that operations are computed in a vectorized manner
- Similarities to functional paradigm (map/filter):
  - specify the operation once, on entire column/frame
  - no loops
    ```

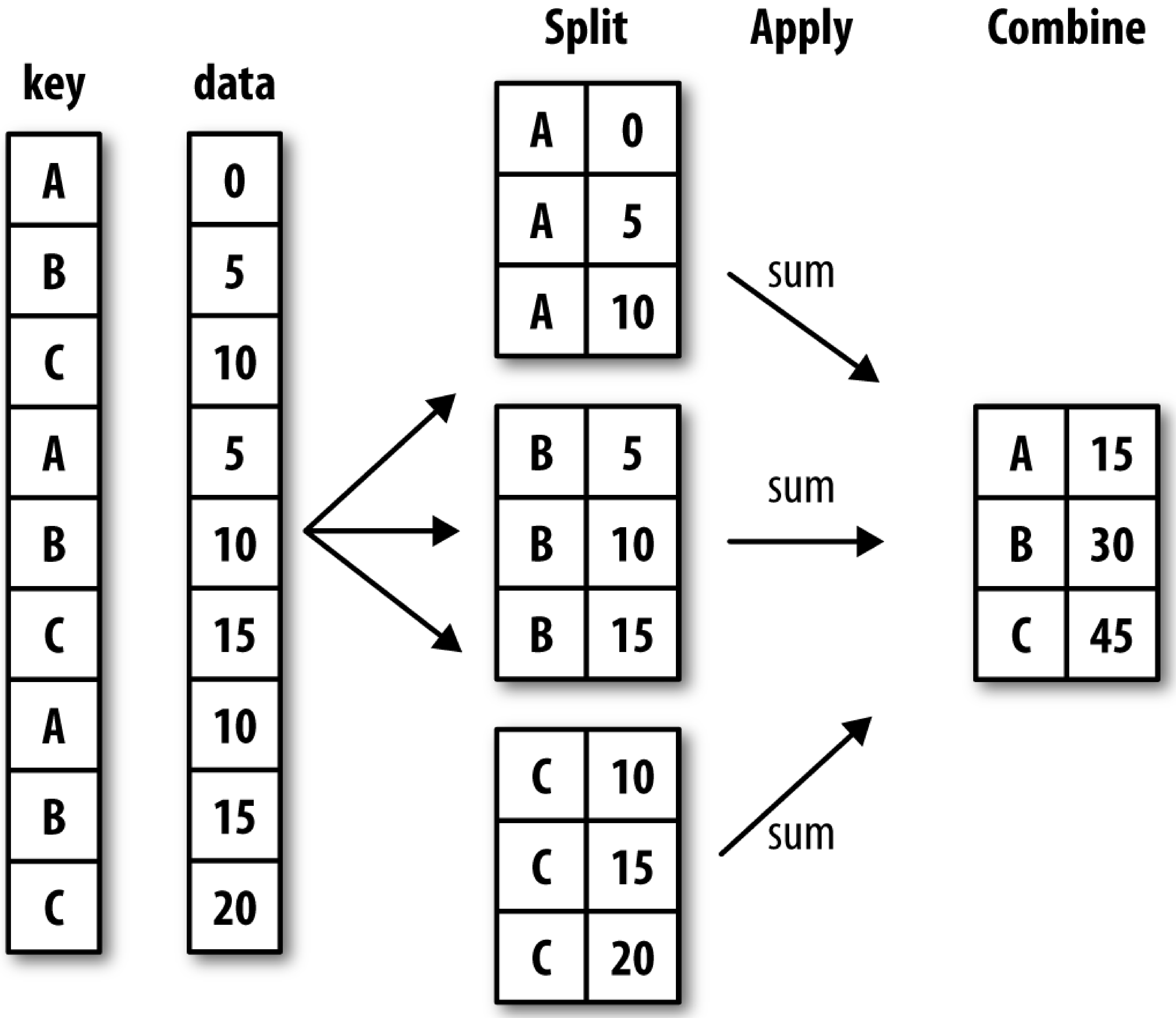
pandas inplace

- Generally, when we modify a data frame, we reassign:
    - `rdf = dfa.reset_index()`
    - This is usually very **efficient**
    - Allows for method chaining
- There are versions where you can do this "inplace" (**try to avoid this**)
    - `dfa.reset_index(inplace=True)`
    - This means **no reassignment**, but it isn't usually any faster nor better
    - Sometimes still creates a copy
    - Will likely be <u>deprecated</u>

Aggregation

- Descriptive statistics
  - `df['Culmen Length (mm)'].mean()`
  - `.median()`
  - `.describe()`
  - `.count()`
  - `.min(), .max()`
- Also general methods
  - `.sum()`
  - `.product()`

Split-Apply-Combine



[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

[T. Brandt]

Group By

- Polars: `group_by`, Pandas: `groupby`
- `group_by` method creates a `GroupBy` object
- `group_by` **does not compute** anything until there is an aggregate step
- Sizes of groups:
  - `df.group_by('Island').agg(pl.len()) # DataFrame`
  - `dfa.groupby('Island').size() # Series`
- Can iterate through the groups (names and dataframes):
  - ```
    for name, gdf in df.group_by('Island'):
        display(name, gdf)
    ```
    ```

Aggregation

- Single Column:
  - `df.group_by('Island').agg(pl.col('Length (mm)').mean())`
  - `dfa.groupby('Island')['Length (mm)'].mean()`
- pandas returns a Series, polars returns a DataFrame
- List of Values:
  - `df.group_by('Island').agg(pl.col('Length (mm)'))`
  - `dfa.groupby('Island')['Length (mm)'].apply(list)`

Aggregation (Multiple Columns)

- Multiple columns in an aggregation
  - `df.group_by('Island').agg(pl.col('Length','Depth').mean())`
  - `dfa.groupby('Island')[['Length','Depth']].mean()`
- Multiple aggregations for a column
  - ```
    df.group_by('Island').agg(pl.col('Length').min().alias('LMin'),
                              pl.col('Length').max().alias('LMax'))
    ```
  - `dfa.groupby('Island').agg({'Length': ['min','max']})`
  - ```
    dfa.groupby('Island').agg(LMin=('Length','min')
                              LMax=('Length','max'))
    ```

Different Data Layouts

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

Initial Data

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

Transpose

name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Tidy Data

[H. Wickham, 2014]

Problem: Variables stored in both rows & columns

Mexico Weather, Global Historical Climatology Network

id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

[H. Wickham, 2014]

Problem: Variables stored in both rows & columns

Mexico Weather, Global Historical Climatology Network

id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

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

[H. Wickham, 2014]

Melting + Pivot

id	date	element	value
MX17004	2010-01-30	tmax	27.8
MX17004	2010-01-30	tmin	14.5
MX17004	2010-02-02	tmax	27.3
MX17004	2010-02-02	tmin	14.4
MX17004	2010-02-03	tmax	24.1
MX17004	2010-02-03	tmin	14.4
MX17004	2010-02-11	tmax	29.7
MX17004	2010-02-11	tmin	13.4
MX17004	2010-02-23	tmax	29.9
MX17004	2010-02-23	tmin	10.7

(a) Molten data

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

(b) Tidy data

[H. Wickham, 2014]