

Advanced Data Management (CSCI 640/490)

Data & Pandas

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.

Why NumPy?

- Fast **vectorized** array operations for data munging and cleaning, subsetting and filtering, transformation, and any other kinds of computations
- Common array algorithms like sorting, unique, and set operations
- Efficient descriptive statistics and aggregating/summarizing data
- Data alignment and relational data manipulations for merging and joining together heterogeneous data sets
- Expressing conditional logic as array expressions instead of loops with `if-elif-else` branches
- Group-wise data manipulations (aggregation, transformation, function application).

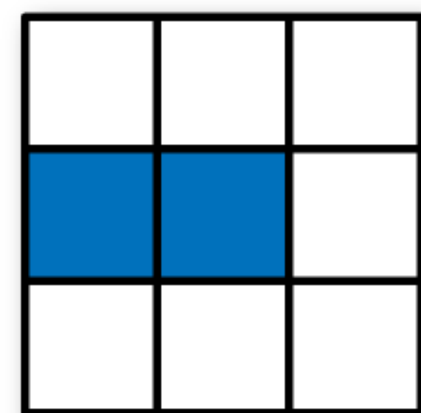
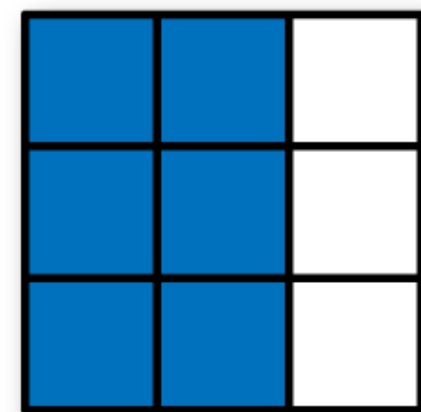
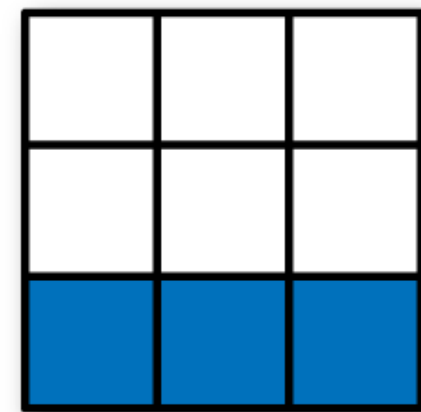
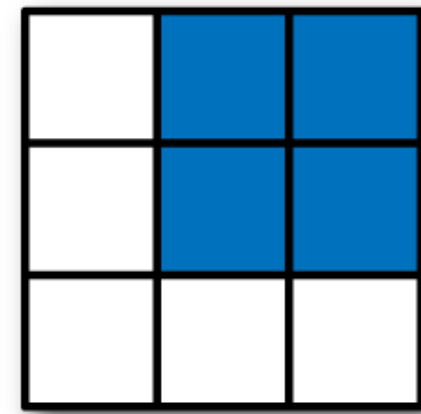
[W. McKinney, Python for Data Analysis]

NumPy Arrays

- `data1 = [6, 7.5, 8, 0, 1]`
`arr1 = np.array(data1)`
- Zeros: `np.zeros(10)`, Ones: `np.ones((4, 5))`,
Empty: `np.empty((2, 2))`
- # of dimensions: `arr2.ndim`, Shape: `arr2.shape`, Type: `arr2.dtype`
- Types: Each array has a fixed type unlike other variables in python

2D Array Slicing

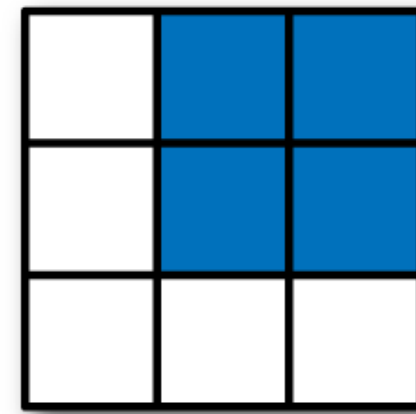
How to obtain the blue slice
from array `arr`?



[W. McKinney, Python for Data Analysis]

2D Array Slicing

How to obtain the blue slice
from array `arr`?

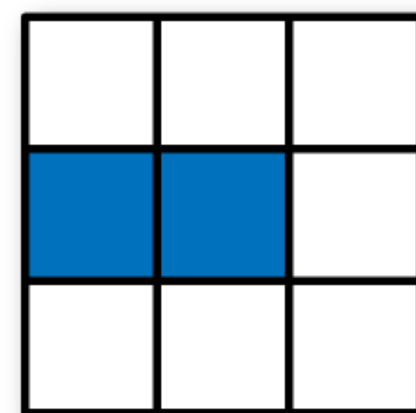
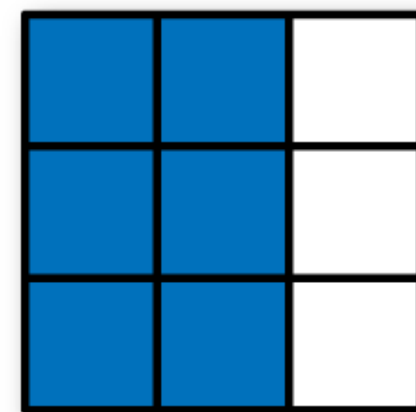
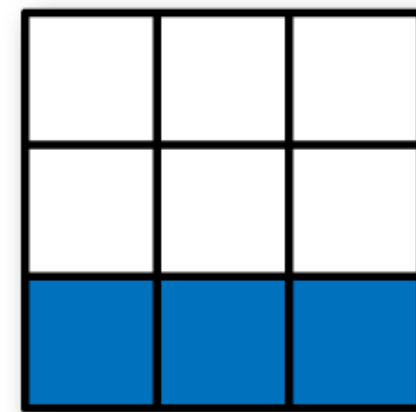


Expression

`arr[:2, 1:]`

Shape

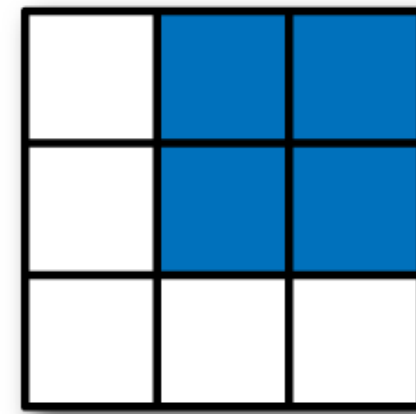
`(2, 2)`



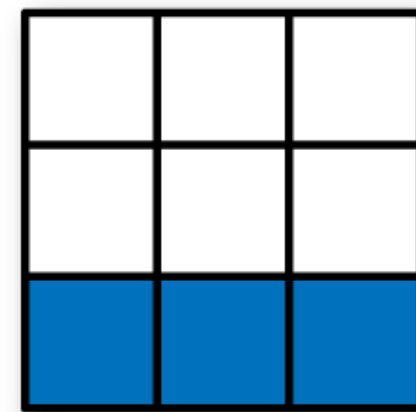
[W. McKinney, Python for Data Analysis]

2D Array Slicing

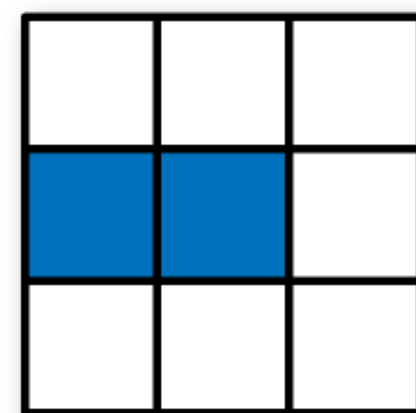
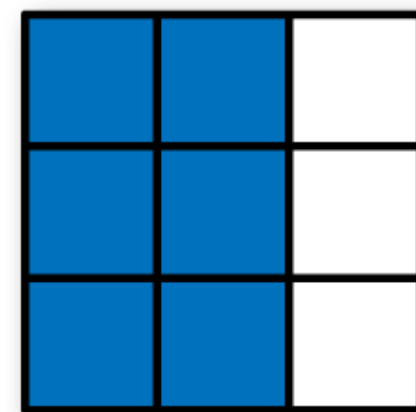
How to obtain the blue slice
from array `arr`?



Expression	Shape
<code>arr[:2, 1:]</code>	<code>(2, 2)</code>



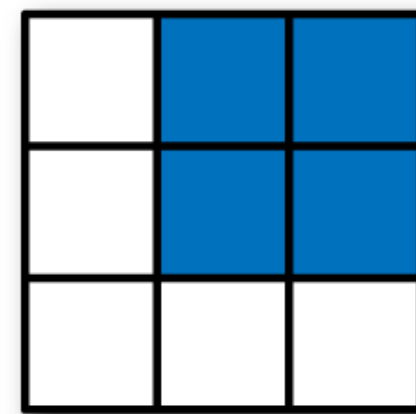
<code>arr[2]</code>	<code>(3,)</code>
<code>arr[2, :]</code>	<code>(3,)</code>
<code>arr[2:, :]</code>	<code>(1, 3)</code>



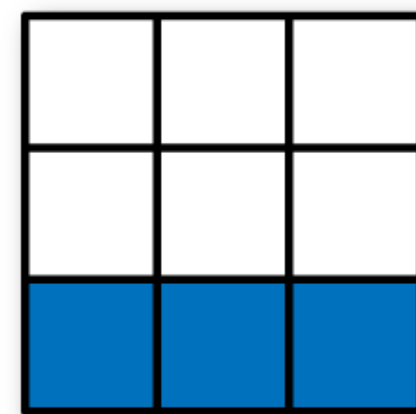
[W. McKinney, Python for Data Analysis]

2D Array Slicing

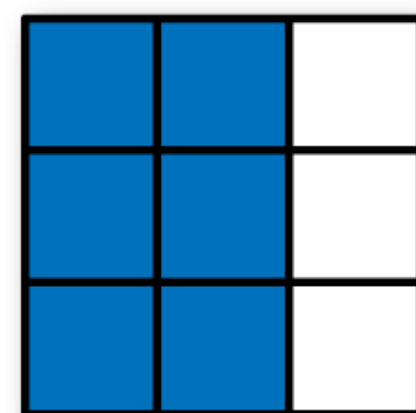
How to obtain the blue slice from array `arr`?



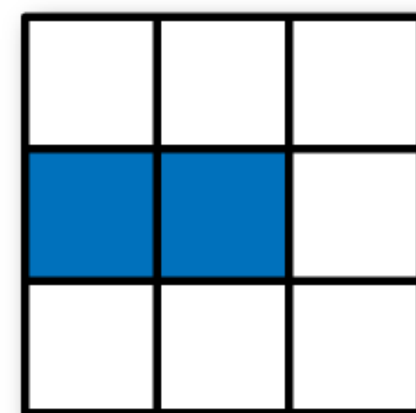
Expression	Shape
<code>arr[:2, 1:]</code>	<code>(2, 2)</code>



<code>arr[2]</code>	<code>(3,)</code>
<code>arr[2, :]</code>	<code>(3,)</code>
<code>arr[2:, :]</code>	<code>(1, 3)</code>



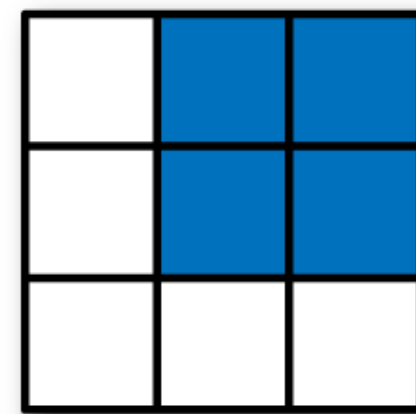
<code>arr[:, :2]</code>	<code>(3, 2)</code>
-------------------------	---------------------



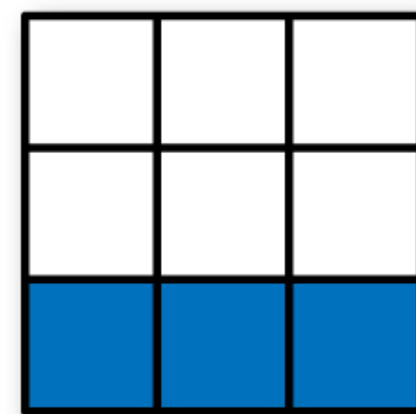
[W. McKinney, Python for Data Analysis]

2D Array Slicing

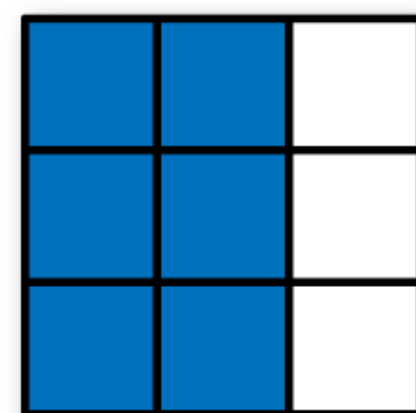
How to obtain the blue slice from array `arr`?



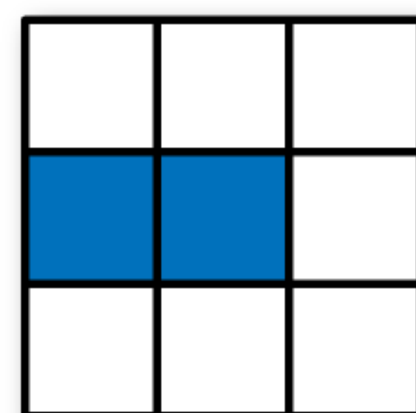
Expression	Shape
<code>arr[:2, 1:]</code>	<code>(2, 2)</code>



<code>arr[2]</code>	<code>(3,)</code>
<code>arr[2, :]</code>	<code>(3,)</code>
<code>arr[2:, :]</code>	<code>(1, 3)</code>



<code>arr[:, :2]</code>	<code>(3, 2)</code>
-------------------------	---------------------



<code>arr[1, :2]</code>	<code>(2,)</code>
<code>arr[1:2, :2]</code>	<code>(1, 2)</code>

[W. McKinney, Python for Data Analysis]

Assignment 2

- Assignment 1 Questions with pandas, DuckDB, and Ibis
- CS 640 students do all, CS 490 do pandas & DuckDB (Ibis is EC)
- Can work by framework or by query
- Most questions can be answered with a single statement... but that statement can take a while to write
 - Read documentation
 - Check hints

More Reshaping

- reshape:
 - `arr2.reshape(4,2)` # returns new view
- resize:
 - `arr2.resize(4,2)` # no return, modifies `arr2` in place
- flatten:
 - `arr2.flatten()` # `array([1.5, 2., 3., 4., 5., 6., 7., 8.])`
- ravel:
 - `arr2.ravel()` # `array([1.5, 2., 3., 4., 5., 6., 7., 8.])`
- flatten and ravel look the same, but ravel is a **view**

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`

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])` # ???

numpy Functions

- Unary: `abs`, `sqrt`, `log`, `ceil`, `sin`, `cos`, `tan`, `arccos`, `arcsin`, ...
- Binary: `add`, `subtract`, `multiple`, `divide`, ... `<`, `>`, `>=`, `<=`, `==`, `!=`
- Statistics: `sum`, `mean`, `std`, `min`, `max`, `argmin`, `argmax`
- Boolean: `any`, `all`
- Others: `sort`, `unique`
- Linear Algebra (`numpy.linalg`)
- Pseudorandom Number Generation (`numpy.random`)

Data

Data

- What is data?
 - Types
 - Semantics
- How is data structured?
 - Tables (Data Frames)
 - Databases
 - Data Cubes
- What formats is data stored in?
- Raw versus derived data

Data

- What is this data?

R011	42ND STREET & 8TH AVENUE	00228985	00008471	00000441	00001455	00000134	00033341	00071255
R170	14TH STREET-UNION SQUARE	00224603	00011051	00000827	00003026	00000660	00089367	00199841
R046	42ND STREET & GRAND CENTRAL	00207758	00007908	00000323	00001183	00003001	00040759	00096613

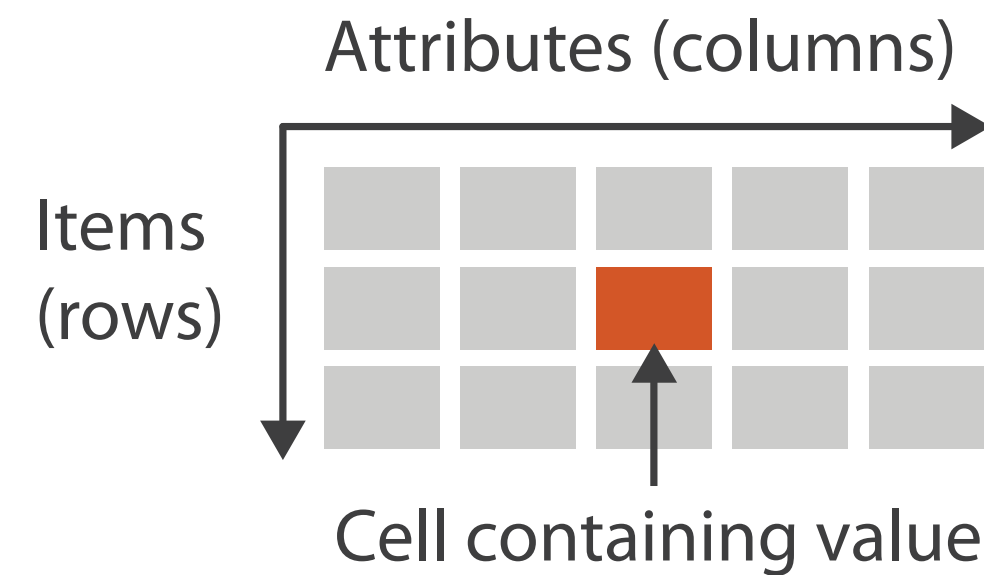
- Semantics: real-world meaning of the data
- Type: structural or mathematical interpretation
- Both often require metadata
 - Sometimes we can infer some of this information
 - Line between data and metadata isn't always clear

Data

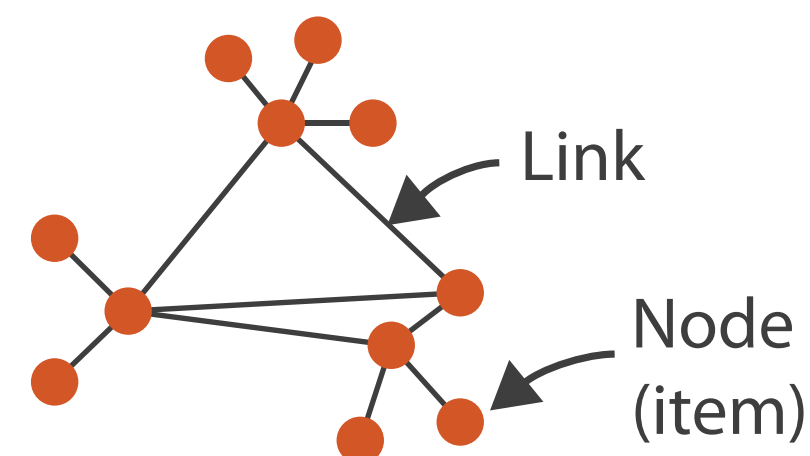
	REMOTE	STATION	FF ▼	SEN/DIS	7-D AFAS UNL	D AFAS/RMF I	JOINT RR TKT	7-D UNL	30-D UNL
1	R011	42ND STREET & 8TH AVENUE	00228985	00008471	00000441	00001455	00000134	00033341	00071255
2	R170	14TH STREET-UNION SQUARE	00224603	00011051	00000827	00003026	00000660	00089367	00199841
3	R046	42ND STREET & GRAND CENTRAL	00207758	00007908	00000323	00001183	00003001	00040759	00096613
4	R012	34TH STREET & 8TH AVENUE	00188311	00006490	00000498	00001279	00003622	00035527	00067483
5	R293	34TH STREET - PENN STATION	00168768	00006155	00000523	00001065	00005031	00030645	00054376
6	R033	42ND STREET/TIMES SQUARE	00159382	00005945	00000378	00001205	00000690	00058931	00078644
7	R022	34TH STREET & 6TH AVENUE	00156008	00006276	00000487	00001543	00000712	00058910	00110466
8	R084	59TH STREET/COLUMBUS CIRCLE	00155262	00009484	00000589	00002071	00000542	00053397	00113966
9	R020	47-50 STREETS/ROCKEFELLER	00143500	00006402	00000384	00001159	00000723	00037978	00090745
10	R179	86TH STREET-LEXINGTON AVE	00142169	00010367	00000470	00001839	00000271	00050328	00125250
11	R023	34TH STREET & 6TH AVENUE	00134052	00005005	00000348	00001112	00000649	00031531	00075040
12	R029	PARK PLACE	00121614	00004311	00000287	00000931	00000792	00025404	00065362
13	R047	42ND STREET & GRAND CENTRAL	00100742	00004273	00000185	00000704	00001241	00022808	00068216
14	R031	34TH STREET & 7TH AVENUE	00095076	00003990	00000232	00000727	00001459	00024284	00038671
15	R017	LEXINGTON AVENUE	00094655	00004688	00000190	00000833	00000754	00020018	00055066
16	R175	8TH AVENUE-14TH STREET	00094313	00003907	00000286	00001144	00000256	00038272	00074661
17	R057	BARCLAYS CENTER	00093804	00004204	00000454	00001386	00001491	00039113	00068119
18	R138	WEST 4TH ST-WASHINGTON SO	00093562	00004677	00000251	00000965	00000127	00031628	00074458

Dataset Types

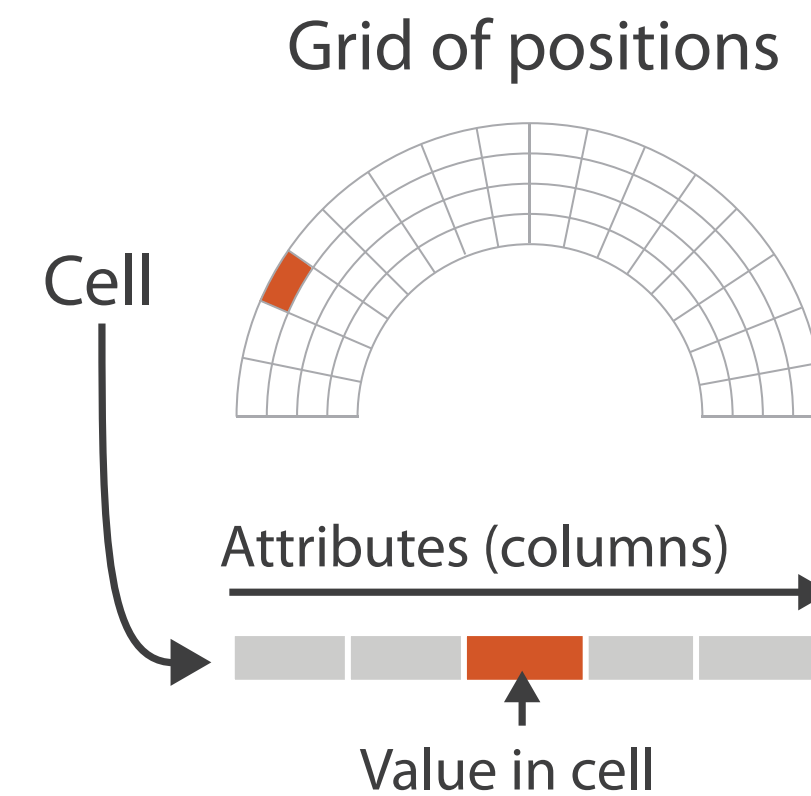
→ Tables



→ Networks



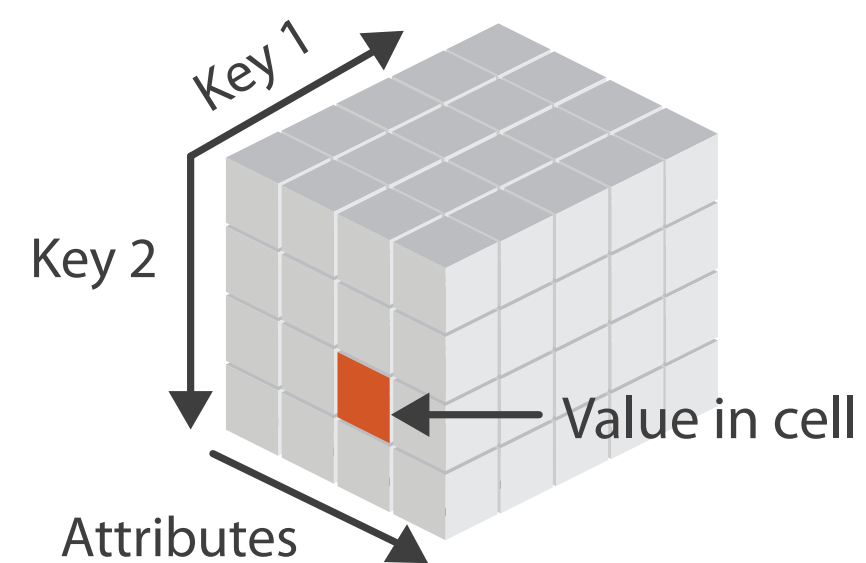
→ Fields (Continuous)



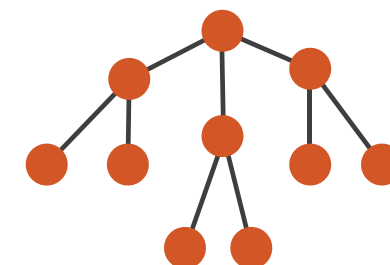
→ Geometry (Spatial)



→ Multidimensional Table



→ Trees



[Munzner (ill. Maguire), 2014]

Data Terminology

- Items
 - An **item** is an individual discrete entity
 - e.g., a row in a table
- Attributes
 - An **attribute** is some specific property that can be measured, observed, or logged
 - a.k.a. variable, (data) dimension
 - e.g., a column in a table

Tables

A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box		7/17/07
32	7/16/07	2-High	Medium Box		7/18/07
32	7/16/07	2-High	Medium Box	0.65	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	1/20/05	5-Low	Wrap Bag	0.56	1/20/05
69	5	4-Not Specified	Small Pack	0.44	6/6/05
69	5	4-Not Specified	Wrap Bag	0.6	6/6/05
70	12/18/06	5-Low	Small Box	0.59	12/23/06
70	12/18/06	5-Low	Wrap Bag	0.82	12/23/06
96	4/17/05	2-High	Small Box	0.55	4/19/05
97	1/29/06	3-Medium	Small Box	0.38	1/30/06
129	11/19/08	5-Low	Small Box	0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

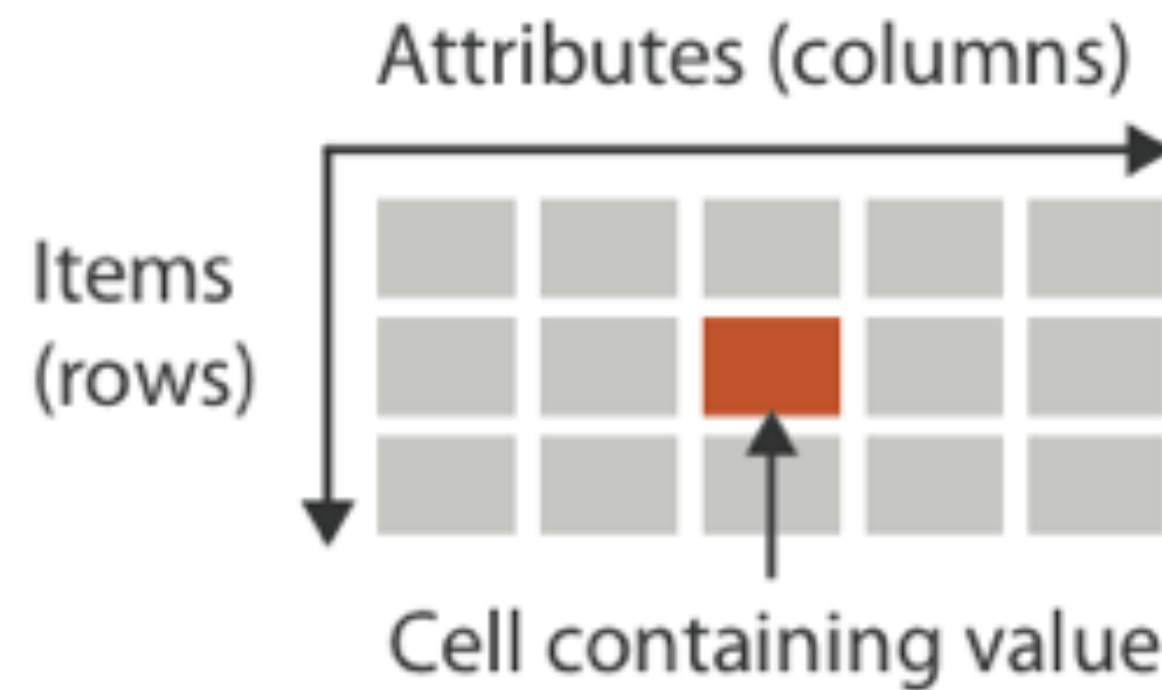
attribute

cell

item

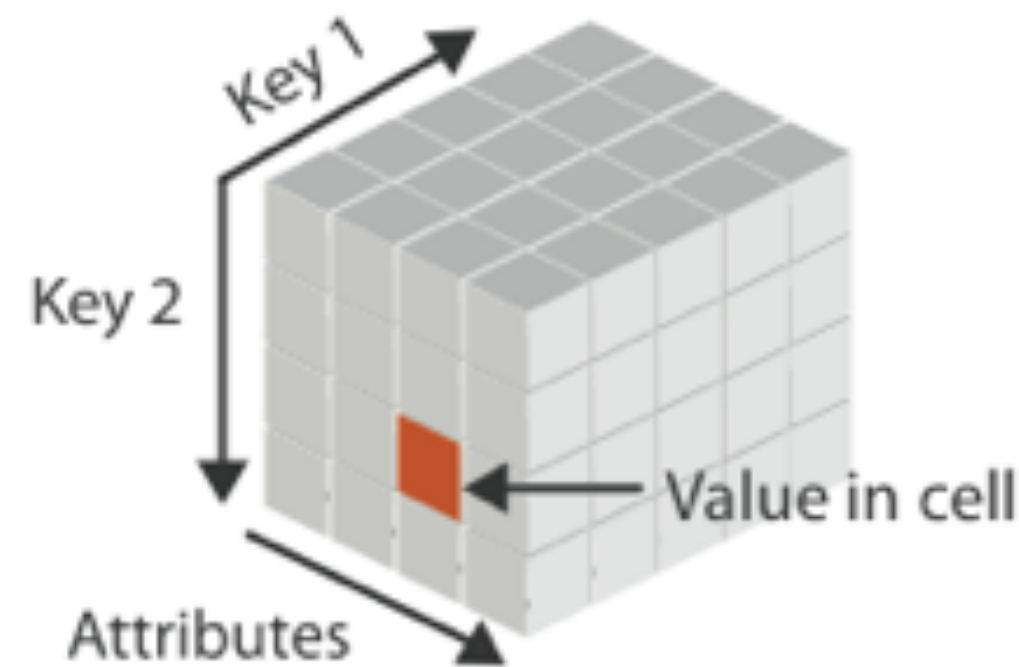
Tables

Flat



- Data organized by rows & columns
 - row ~ item (usually)
 - column ~ attribute
 - label ~ attribute name
- Key: identifies each item (row), usually unique
 - Allows **join** of data from 2+ tables
 - Compound key: key split among multiple columns, e.g. (state, year) for population
- Multidimensional:
 - Split compound key
 - e.g. a data cube with (state, year)

Multidimensional



[Munzner (ill. Maguire), 2014]

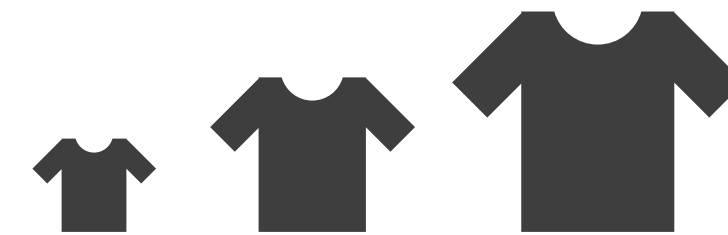
Attribute Types

→ Categorical

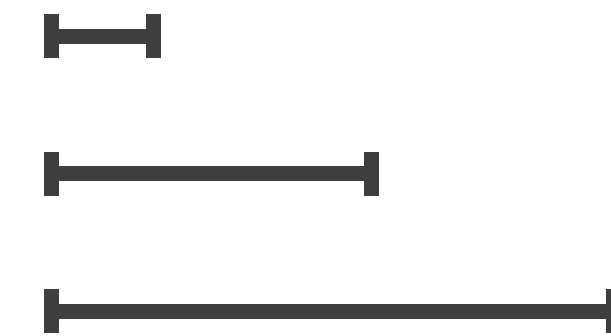


→ Ordered

→ *Ordinal*



→ *Quantitative*



Categorical, Ordinal, and Quantitative

A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	0.72	7/17/07
32	7/16/07	2-High	Medium Box	0.6	7/18/07
32	7/16/07	2-High	Medium Box	0.65	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	1/20/05	5-Low	Wrap Bag	0.56	1/20/05
69	6/4/05	4-Not Specified	Small Pack	0.44	6/6/05
69	6/4/05	4-Not Specified		0.6	6/6/05
70	12/18/06	5-Low		0.59	12/23/06
70	12/18/06	5-Low		0.82	12/23/06
96	4/17/05	2-High		0.55	4/19/05
97	1/29/06	3-Medium		0.38	1/30/06
129	11/19/08	5-Low		0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

quantitative
ordinal
categorical

Categorical, Ordinal, and Quantitative

A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	0.72	7/17/07
32	7/16/07	2-High	Medium Box	0.6	7/18/07
32	7/16/07	2-High	Medium Box	0.65	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	1/20/05	5-Low	Wrap Bag	0.56	1/20/05
69	6/4/05	4-Not Specified	Small Pack	0.44	6/6/05
69	6/4/05	4-Not Specified		0.6	6/6/05
70	12/18/06	5-Low		0.59	12/23/06
70	12/18/06	5-Low		0.82	12/23/06
96	4/17/05	2-High		0.55	4/19/05
97	1/29/06	3-Medium		0.38	1/30/06
129	11/19/08	5-Low		0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

quantitative
ordinal
categorical

Attribute Types

- May be further specified for computational storage/processing
 - **Categorical**: string, boolean, blood type
 - **Ordered**: enumeration, t-shirt size
 - **Quantitative**: integer, float, fixed decimal, datetime
- Sometimes, types can be **inferred** from the data
 - e.g. numbers and none have decimal points → integer
 - could be incorrect (data doesn't have floats, but could be)

Ordering Direction

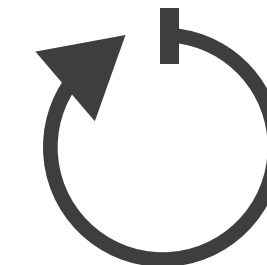
→ Sequential



→ Diverging

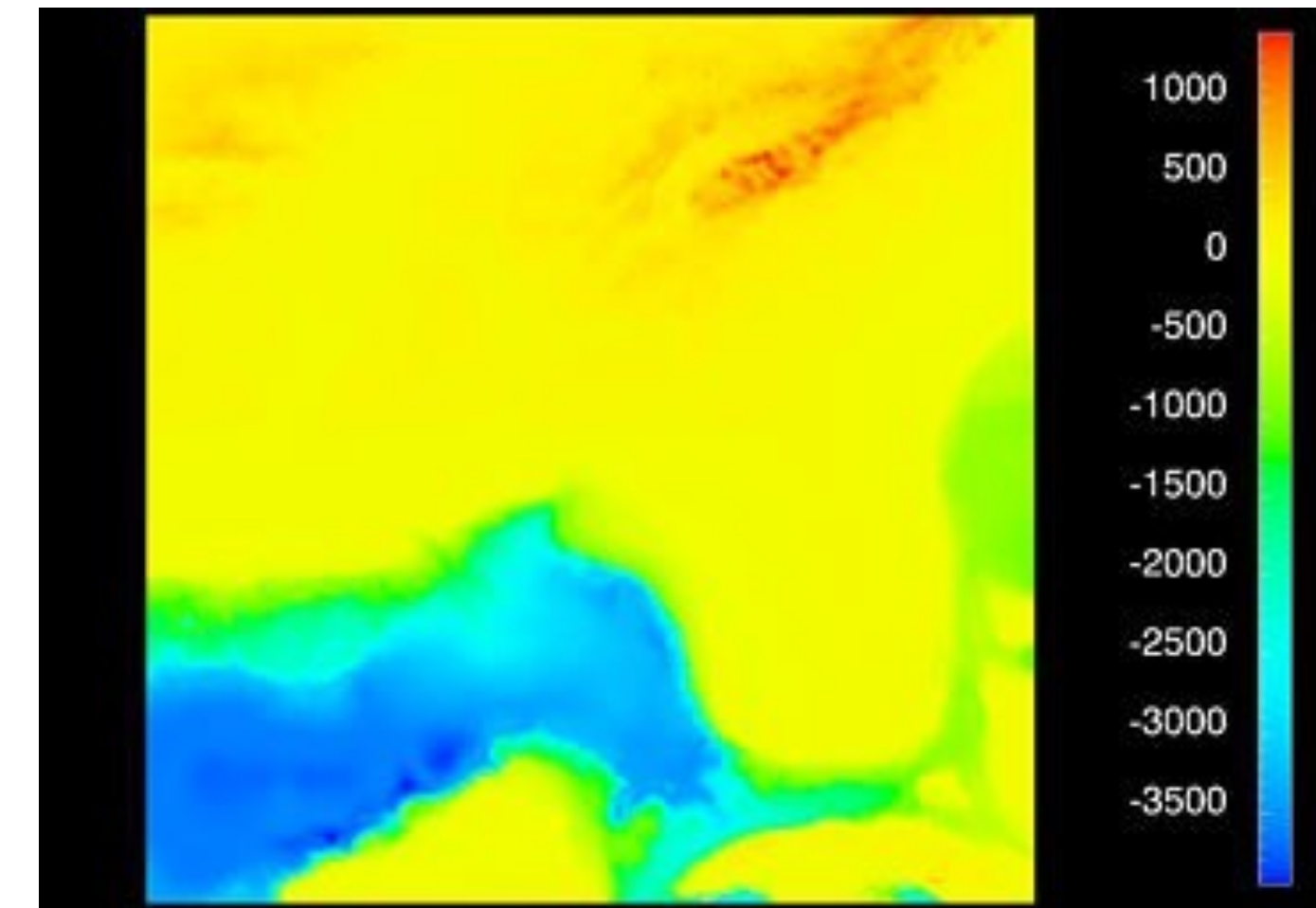


→ Cyclic



Sequential and Diverging Data

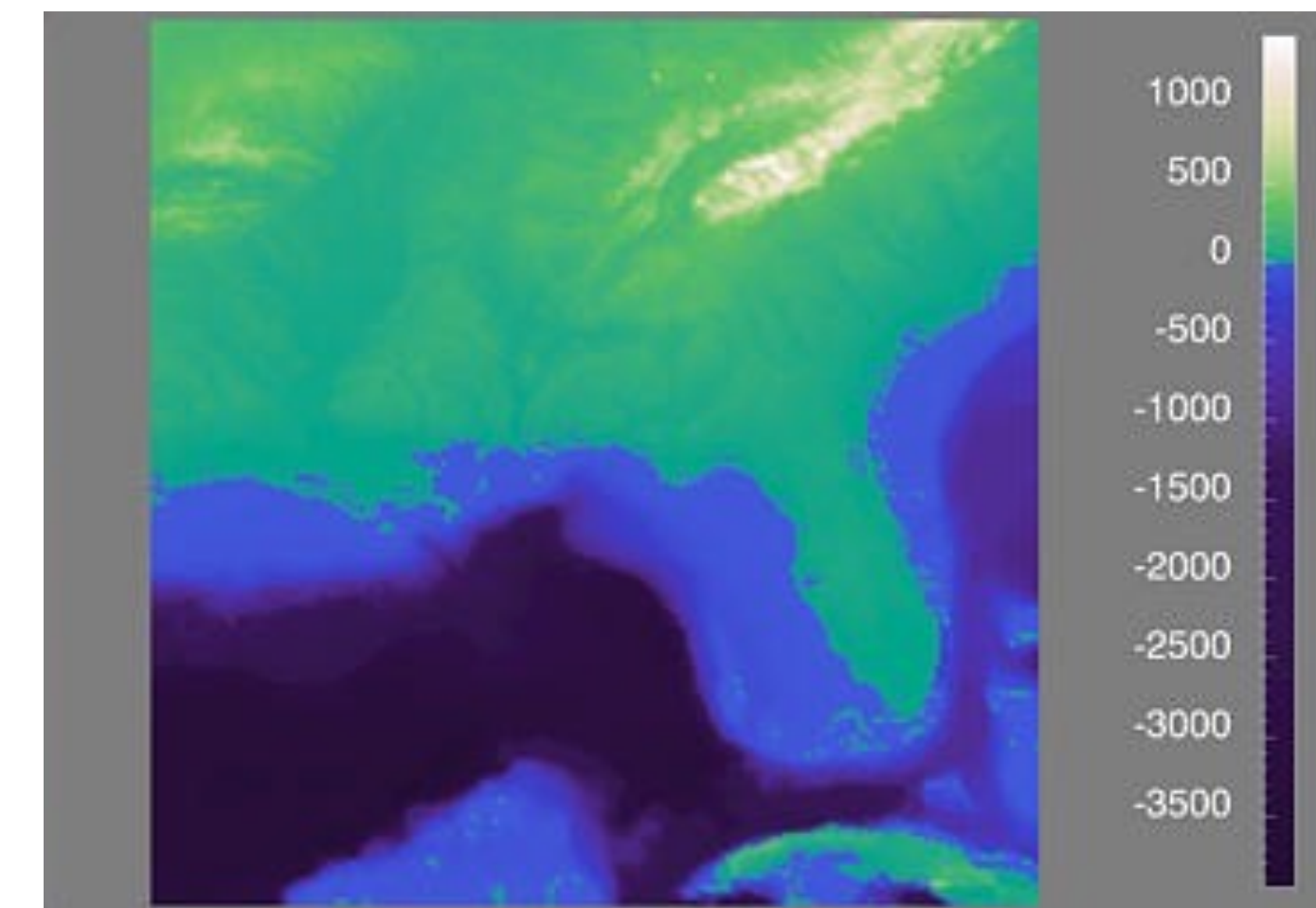
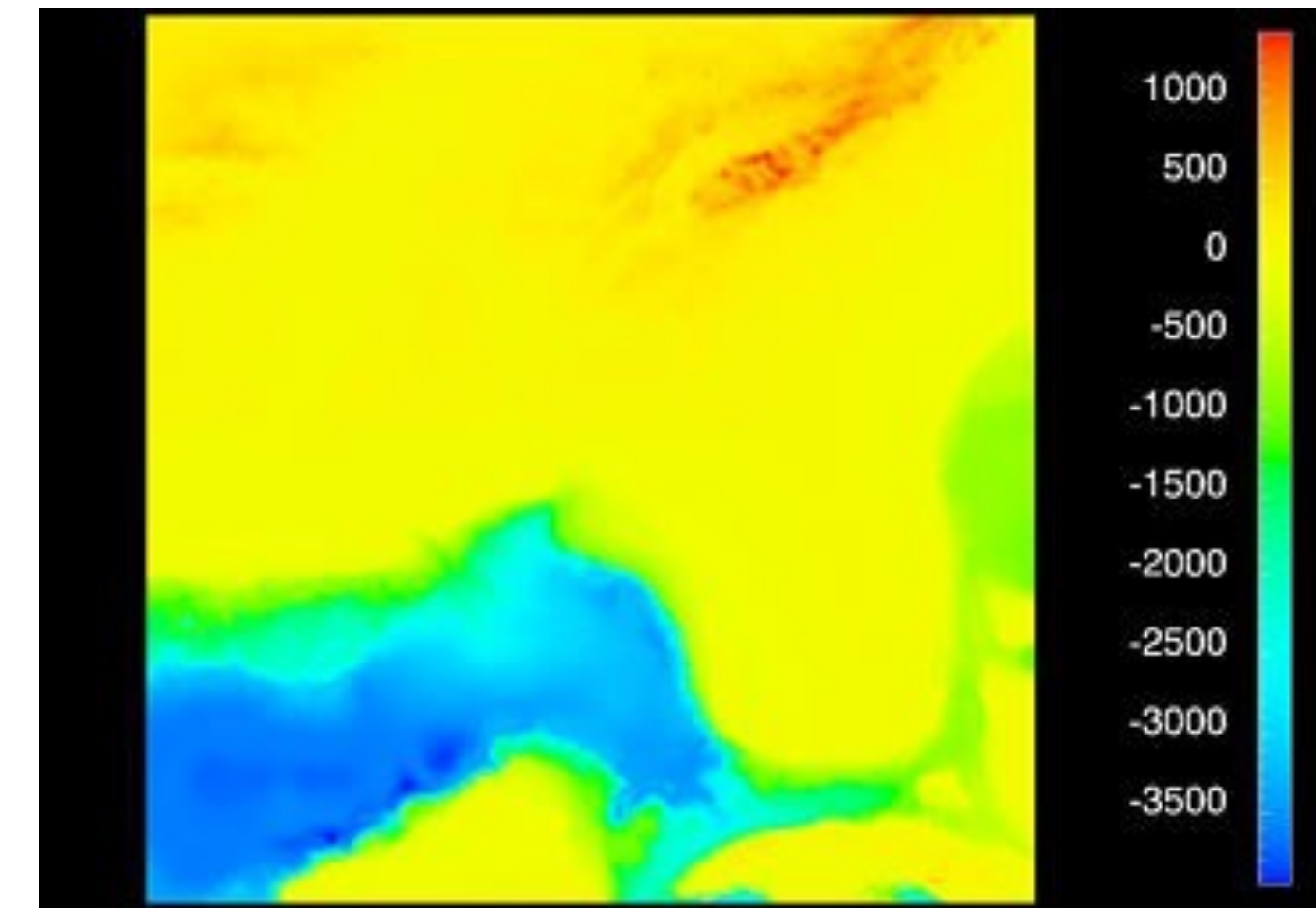
- Sequential: homogenous range from a minimum to a maximum
 - Examples: Land elevations, ocean depths
- Diverging: can be deconstructed into two sequences pointing in opposite directions
 - Has a **zero point** (not necessary 0)
 - Example: Map of both land elevation and ocean depth



[Rogowitz & Treinish, 1998]

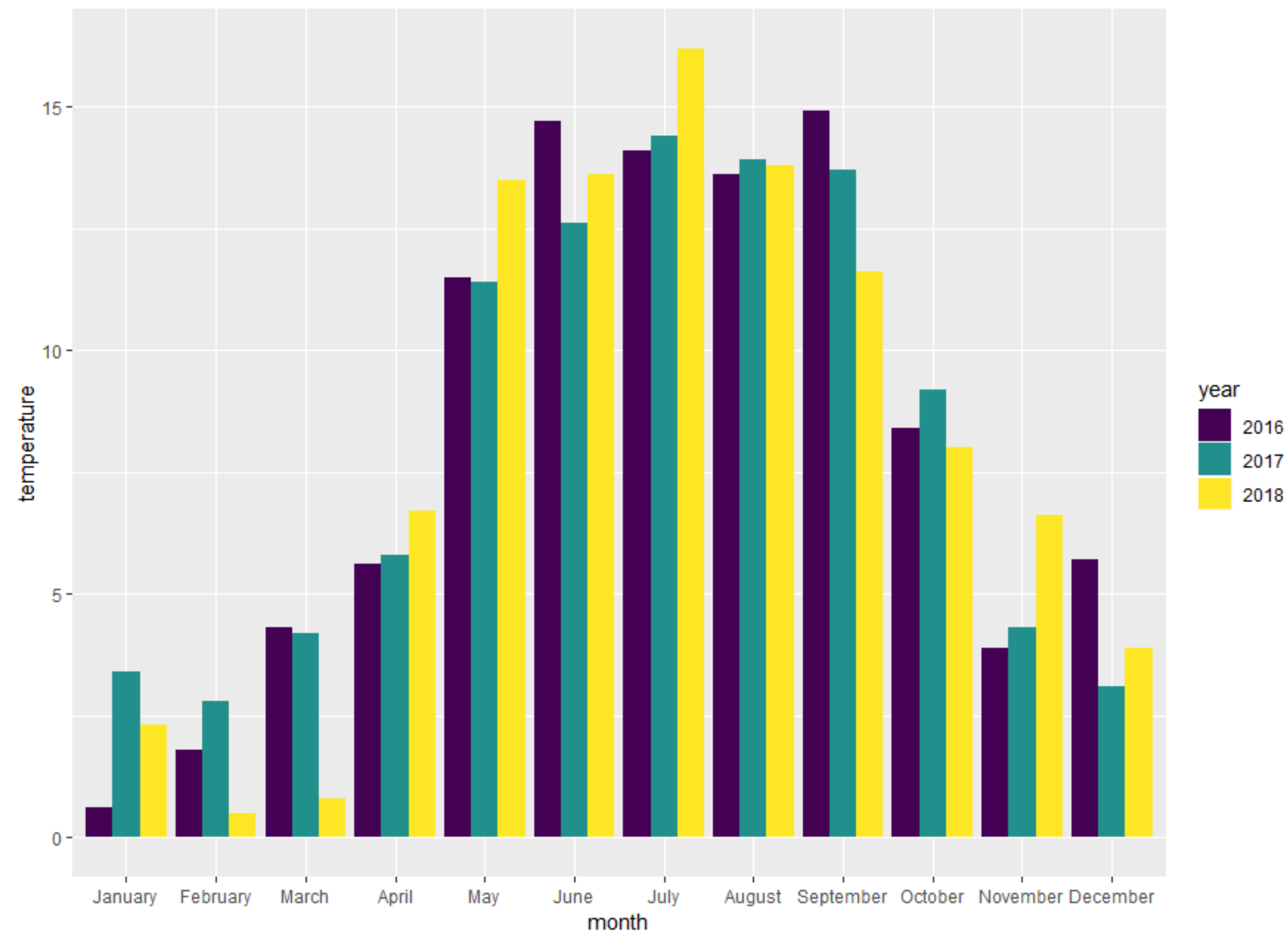
Sequential and Diverging Data

- Sequential: homogenous range from a minimum to a maximum
 - Examples: Land elevations, ocean depths
- Diverging: can be deconstructed into two sequences pointing in opposite directions
 - Has a **zero point** (not necessary 0)
 - Example: Map of both land elevation and ocean depth

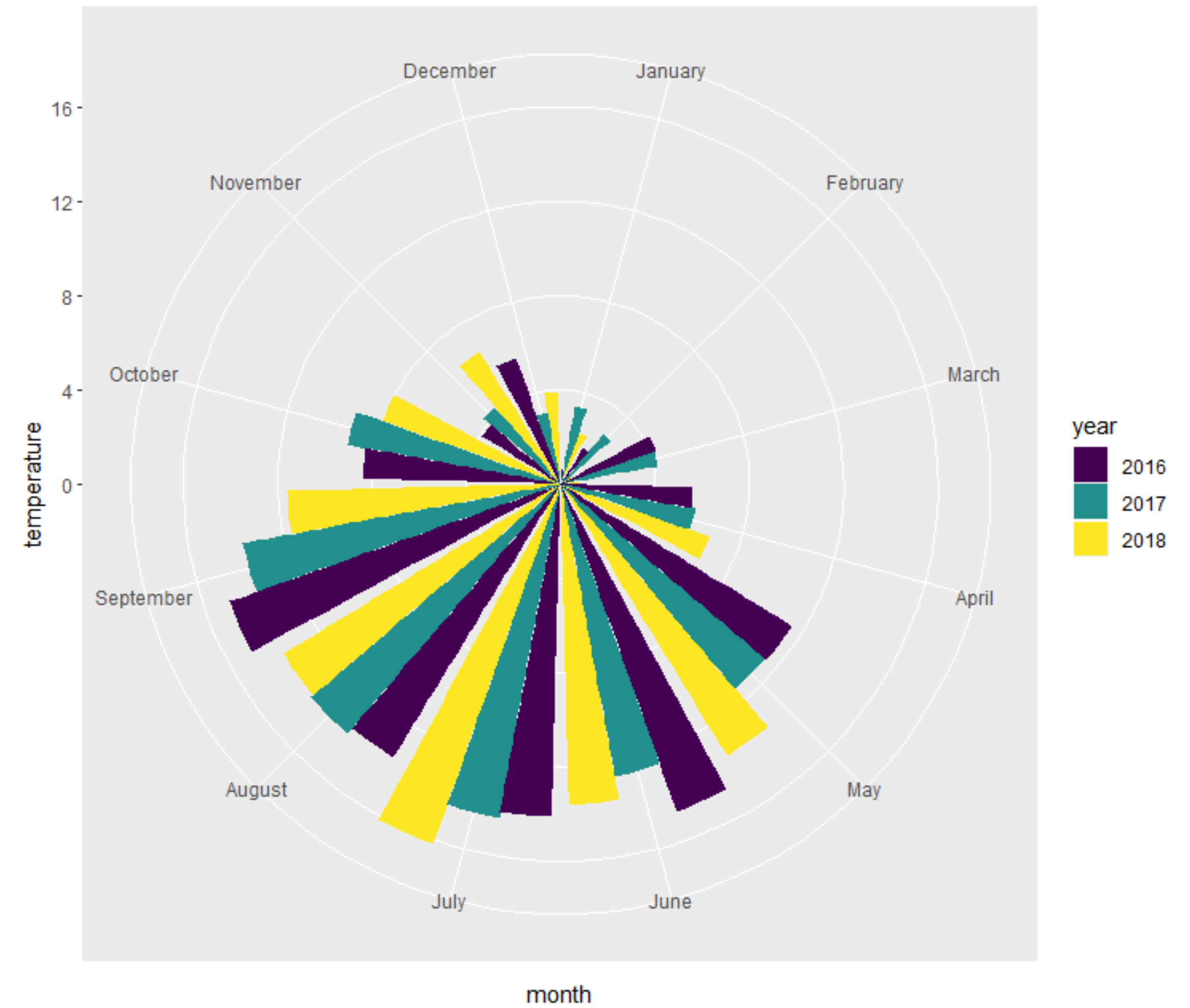
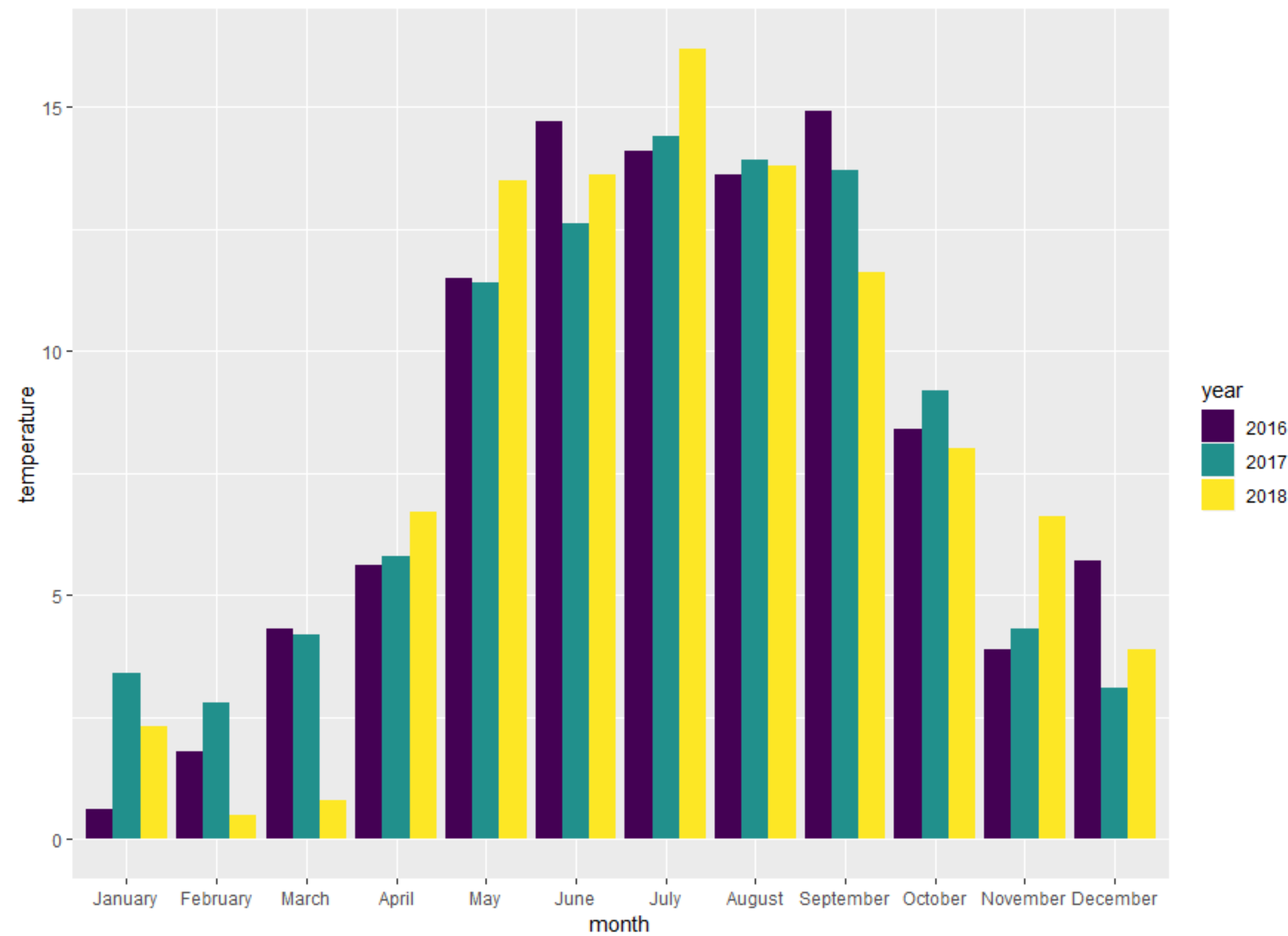


[Rogowitz & Treinish, 1998]

Cyclic Data



Cyclic Data



Semantics

- The meaning of the data
- Example: 94023, 90210, 02747, 60115

Semantics

- The meaning of the data
- Example: 94023, 90210, 02747, 60115
 - Attendance at college football games?

Semantics

- The meaning of the data
- Example: 94023, 90210, 02747, 60115
 - Attendance at college football games?
 - Salaries?

Semantics

- The meaning of the data
- Example: 94023, 90210, 02747, 60115
 - Attendance at college football games?
 - Salaries?
 - Zip codes?
- Cannot always infer based on what the data looks like
- Often require semantics to better understand data, column names help
- May also include rules about data: a zip code is part of an address that uniquely identifies a residence
- Useful for asking good questions about the data

Data Model vs. Conceptual Model

- Data Model: raw data that has a specific data type (e.g. floats):
 - Temperature Example: `[32.5, 54.0, -17.3]` (floats)
- Conceptual Model: how we think about the data
 - Includes semantics, reasoning
 - Temperature Example:
 - Quantitative: `[32.50, 54.00, -17.30]`

[via A. Lex, 2015]

Data Model vs. Conceptual Model

- Data Model: raw data that has a specific data type (e.g. floats):
 - Temperature Example: [32.5, 54.0, -17.3] (floats)
- Conceptual Model: how we think about the data
 - Includes semantics, reasoning
 - Temperature Example:
 - Quantitative: [32.50, 54.00, -17.30]
 - Ordered: [warm, hot, cold]

[via A. Lex, 2015]

Data Model vs. Conceptual Model

- Data Model: raw data that has a specific data type (e.g. floats):
 - Temperature Example: `[32.5, 54.0, -17.3]` (floats)
- Conceptual Model: how we think about the data
 - Includes semantics, reasoning
 - Temperature Example:
 - Quantitative: `[32.50, 54.00, -17.30]`
 - Ordered: `[warm, hot, cold]`
 - Categorical: `[not burned, burned, not burned]`

[via A. Lex, 2015]

Derived Data

Derived Data

- Often, data in its original form isn't as useful as we would like
- Examples: Data about a basketball team's games

Derived Data

- Often, data in its original form isn't as useful as we would like
- Examples: Data about a basketball team's games
- Example 1: `1stHalfPoints`, `2ndHalfPoints`
 - More useful to know total number of points
 - `Points = 1stHalfPoints + 2ndHalfPoints`

Derived Data

- Often, data in its original form isn't as useful as we would like
- Examples: Data about a basketball team's games
- Example 1: `1stHalfPoints`, `2ndHalfPoints`
 - More useful to know total number of points
 - `Points = 1stHalfPoints + 2ndHalfPoints`
- Example 2: `Points`, `OpponentPoints`
 - Want to have a column indicating win/loss
 - `Win = True if (Points > OpponentPoints) else False`

Derived Data

- Often, data in its original form isn't as useful as we would like
- Examples: Data about a basketball team's games
- Example 1: `1stHalfPoints`, `2ndHalfPoints`
 - More useful to know total number of points
 - `Points = 1stHalfPoints + 2ndHalfPoints`
- Example 2: `Points`, `OpponentPoints`
 - Want to have a column indicating win/loss
 - `Win = True if (Points > OpponentPoints) else False`
- Example 3: `Points`
 - Want to have a column indicating how that point total ranks
 - `Rank = index in sorted list of all Point values`

pandas

- Contains high-level data structures and manipulation tools designed to make data analysis fast and easy in Python
- Built on top of NumPy
- Requirements:
 - Data structures with labeled axes (aligning data)
 - Time series data
 - Arithmetic operations that include metadata (labels)
 - Handle missing data
 - Merge and relational operations

Pandas Code Conventions

- Universal:
 - `import pandas as pd`
- Also used:
 - `from pandas import Series, DataFrame`

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
- `obj = pd.Series([7, 14, -2, 1])`
- Basically two arrays: `obj.values` and `obj.index`
- Can specify the index explicitly and use strings
- `obj2 = pd.Series([4, 7, -5, 3],
index=['d', 'b', 'a', 'c'])`
- Kind of like fixed-length, ordered dictionary + can create from a dictionary
- `obj3 = pd.Series({'Ohio': 35000, 'Texas': 71000,
'Oregon': 16000, 'Utah': 5000})`

Series

- Indexing: `s[1]` or `s['Oregon']`
- Can check for missing data: `pd.isnull(s)` or `pd.notnull(s)`
- Both index and values can have an associated name:
 - `s.name = 'population'; s.index.name = 'state'`
- Addition and NumPy ops work as expected and preserve the index-value link
- These operations **align**:

```
In [28]: obj3
Out[28]:
Ohio      35000
Oregon     16000
Texas      71000
Utah        5000
dtype: int64
```

```
In [29]: obj4
Out[29]:
California    NaN
Ohio          35000
Oregon        16000
Texas         71000
dtype: float64
```

```
In [30]: obj3 + obj4
Out[30]:
California    NaN
Ohio          70000
Oregon        32000
Texas        142000
Utah           NaN
dtype: float64
```

[W. McKinney, Python for Data Analysis]

Data Frame

- A dictionary of Series (labels for each series)
- A spreadsheet with column headers
- Has an index shared with each series
- Allows easy reference to any cell
- ```
df = DataFrame({'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada'],
 'year': [2000, 2001, 2002, 2001],
 'pop': [1.5, 1.7, 3.6, 2.4]})
```
- 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

# Data Frame

```
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 x 17 columns



# Data Frame

```
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 x 17 columns





# Data Frame

```
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) |
|-----------|---------------|---------|--------|--------|-------|---------------|-------------------|----------|--------------------|
|-----------|---------------|---------|--------|--------|-------|---------------|-------------------|----------|--------------------|

Index

|     |         |     |                                     |        |           |                    |       |     |          |      |
|-----|---------|-----|-------------------------------------|--------|-----------|--------------------|-------|-----|----------|------|
| 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 x 17 columns



# Data Frame

```
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) |
|-----------|---------------|---------|--------|--------|-------|---------------|-------------------|----------|--------------------|
|-----------|---------------|---------|--------|--------|-------|---------------|-------------------|----------|--------------------|

Index

|     |         |     |                                     |        |           |                    |       |     |          |      |
|-----|---------|-----|-------------------------------------|--------|-----------|--------------------|-------|-----|----------|------|
| 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 x 17 columns

Column: df[ 'Island' ]



# Data Frame

```
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 x 17 columns

Row: df.loc[2]

Index

Column: df['Island']

# Data Frame

```
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 x 17 columns

Column: df['Island']

# Data Frame

```
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 |                    |
| ... | ...       | ...           | ...                                 | ...    | ...       | ...                | ...           | ...               | ...      | ...                |
| 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

Missing Data

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

Column: df['Island']

344 rows x 17 columns