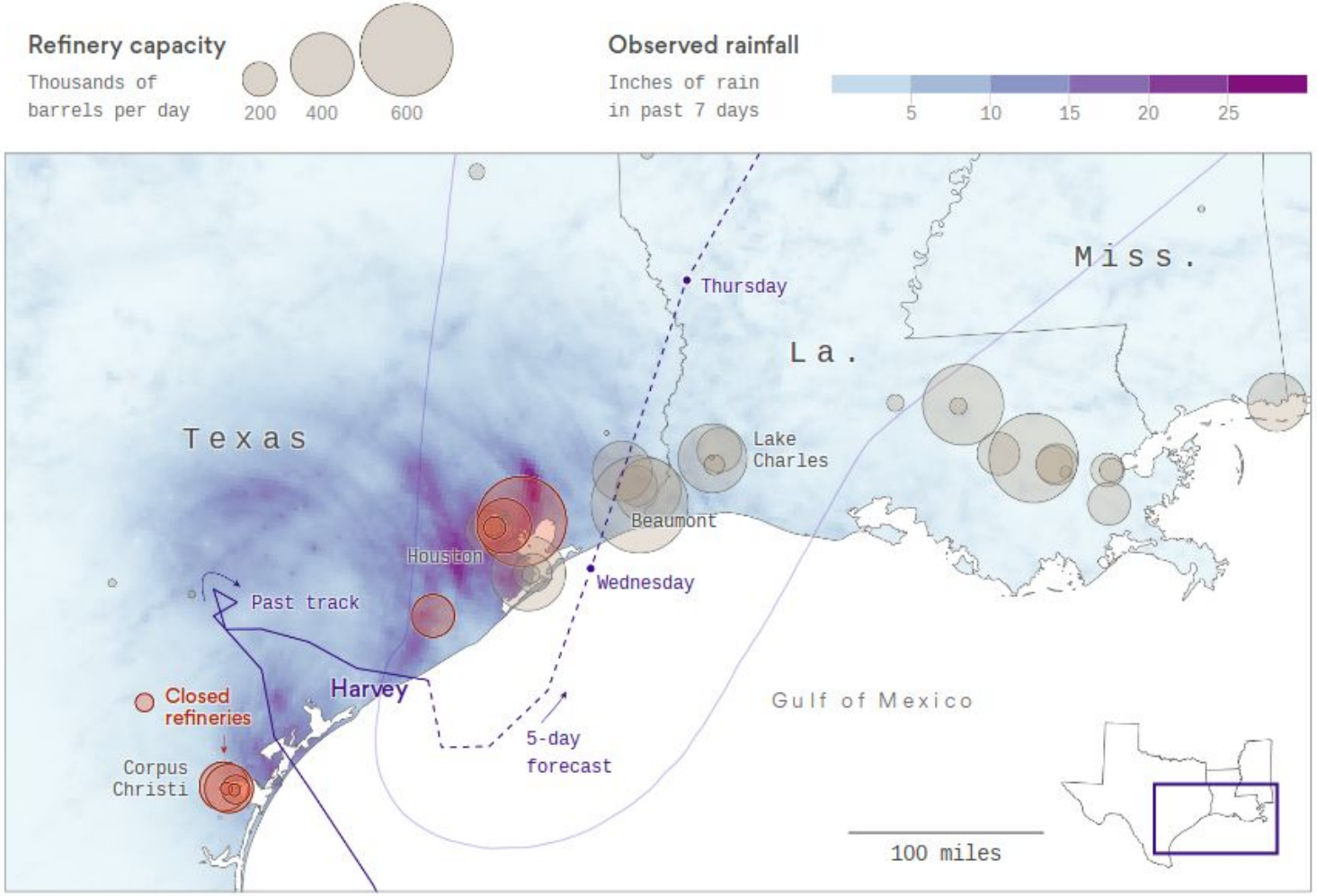
Lecture 2

Spring 2017
Matthew Turk

Warm-Up Activity

1. What is the visualization trying to show?
2. What are its methods?
3. What are the strengths / weaknesses?

Warm-Up Activity



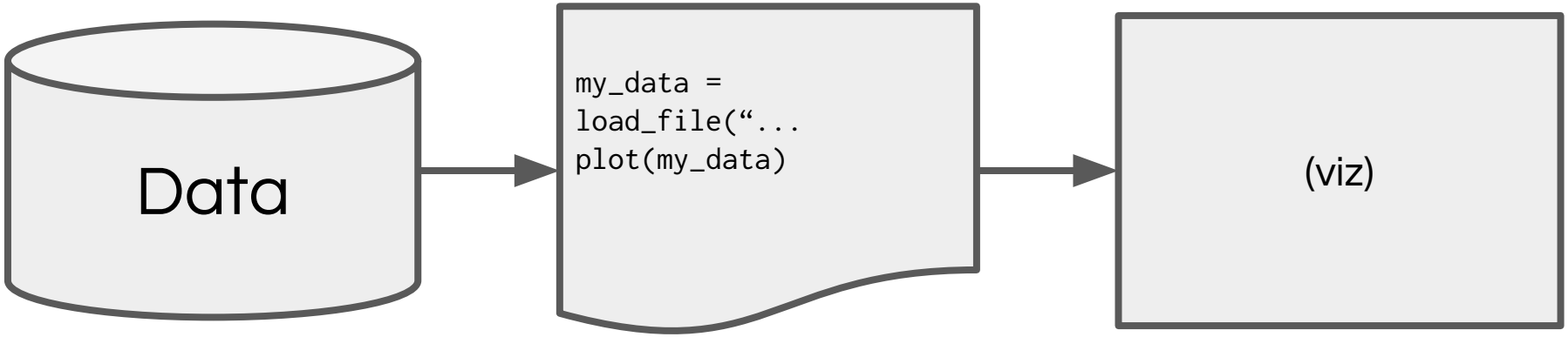
Data: National Hurricane Center, National Weather Service, U.S. Energy Information Administration; Map: Lazaro Gamio / Axios

https://www.axios.com/the-latest-on-harveys-energy-toll-2479055875.html

Topics

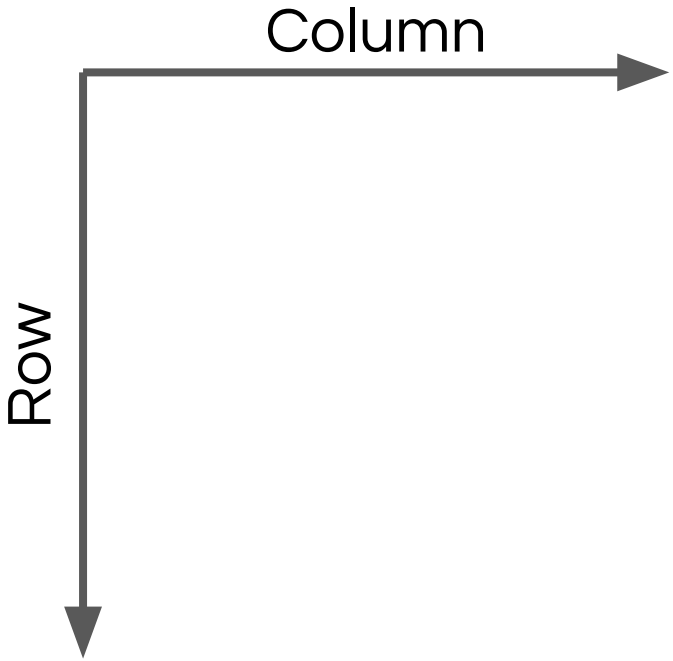
- JupyterHub
- Data Formats
- Operational Palette
- Notebook walkthrough
  - Data structures: lists, dicts, sets
  - Iteration
  - Plotting
  - Numpy and indexing

https://lis590.ncsa.illinois.edu/


Data
my_data =
load_file(“...
plot(my_data)
(viz)

Files, Data and Organization

- Text
  - ASCII (raw)
  - CSV / TSV
  - JSON
- Binary
  - HDF5
  - PNG/BMP/GIF/JPG/etc
  - Excel
  - Arrow
- Query-based
  - SQL
  - JSON



	Column 1	Column 2	Column 3	Column 4
Row 1	11	21	31	41
Row 2	12	22	32	42
Row 3	13	23	33	43

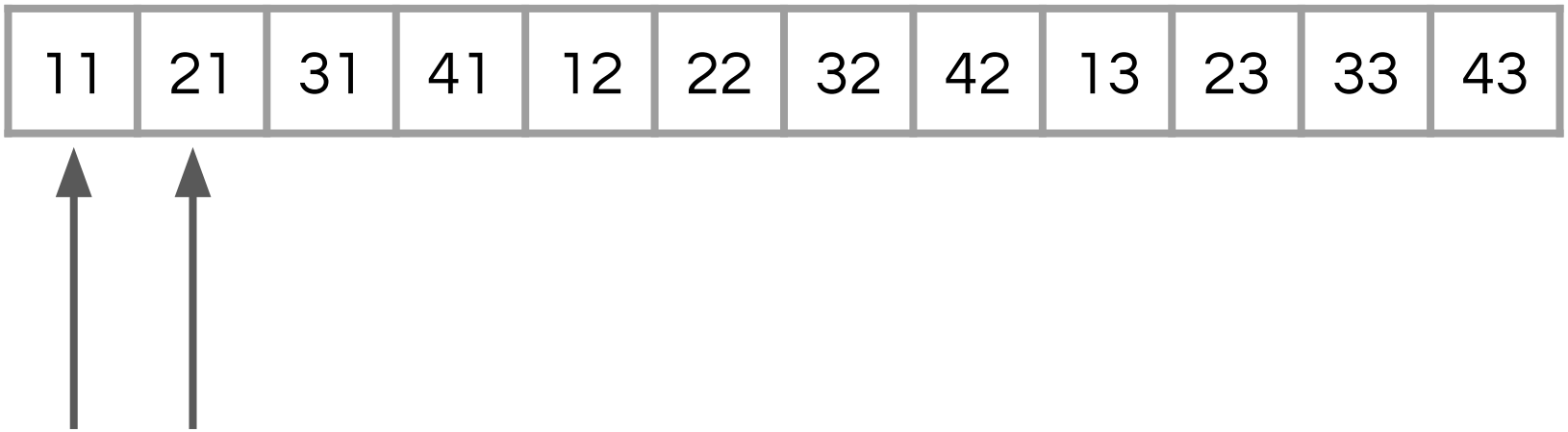
Row-Based Organization:

11	21	31	41	12	22	32	42	13	23	33	43

Column-Based Organization:

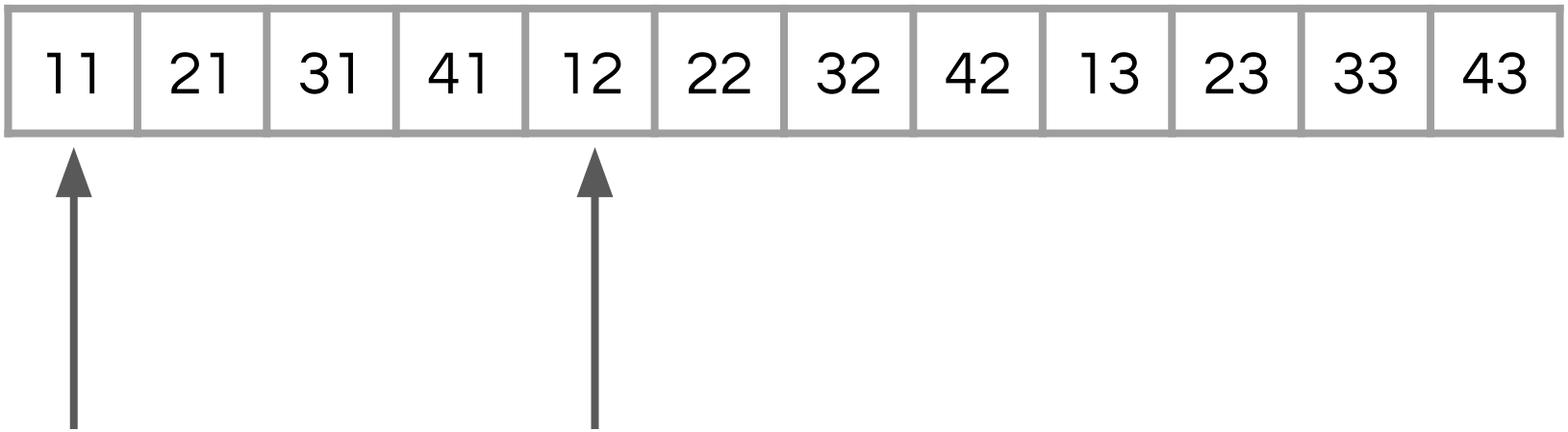
11	12	13	21	22	23	31	32	33	41	42	43

Row-Based Organization:



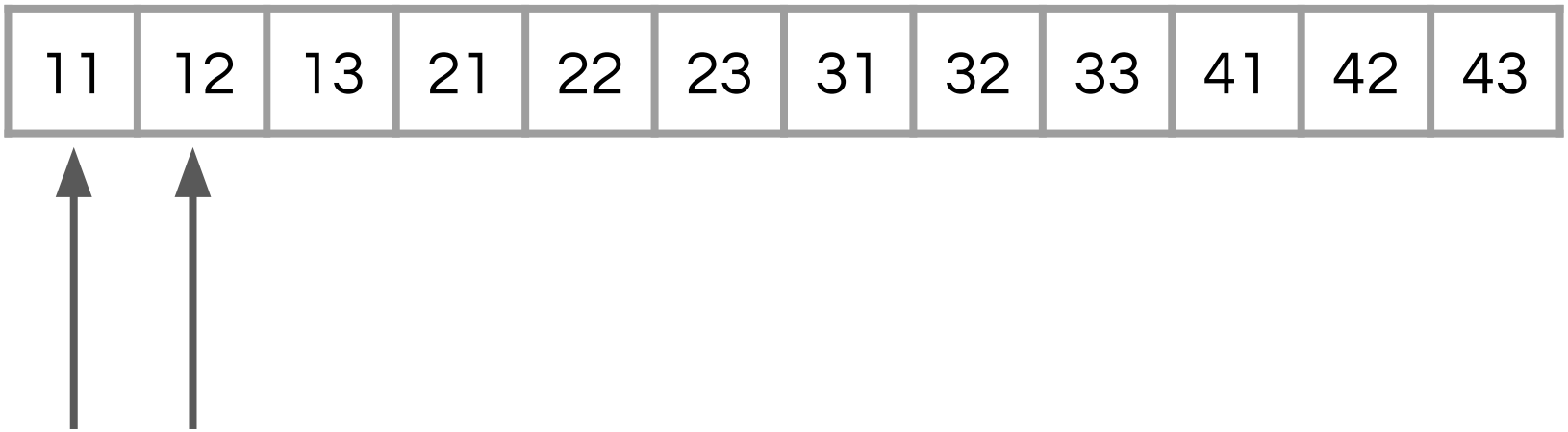
Successive fields in a record are adjacent

Row-Based Organization:



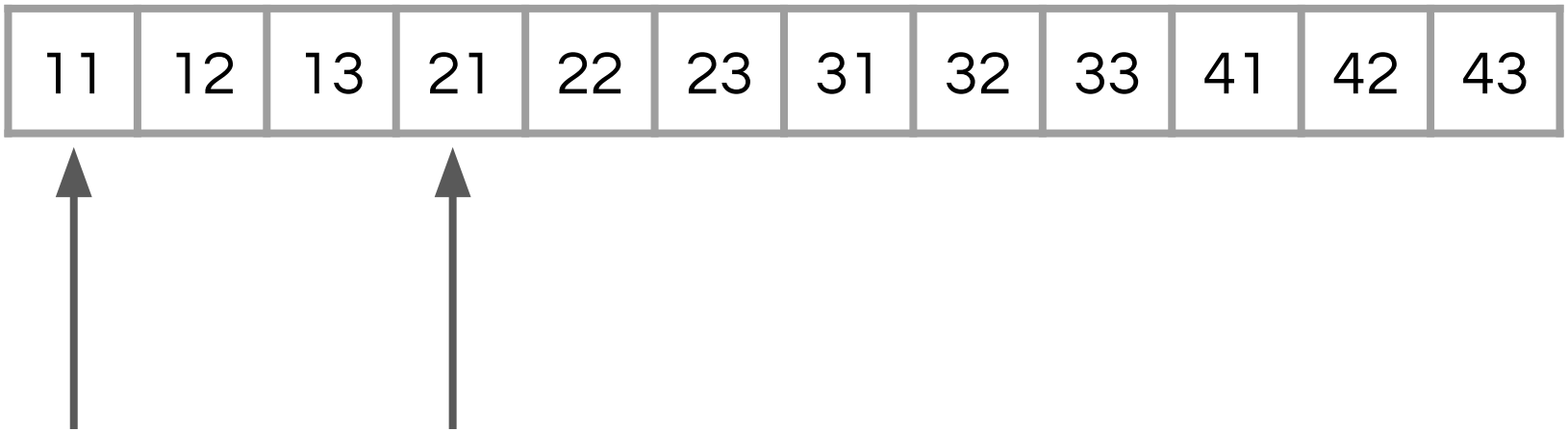
Successive column entries are separated

Column-Based Organization:



Successive records in a column are adjacent

Column-Based Organization:



Successive fields in a record are separated

Files, Data and Organization

- Text
  - ASCII (raw)
  - **CSV / TSV**
  - **JSON**
- Binary
  - **HDF5**
  - PNG/BMP/GIF/JPG/etc
  - Excel
  - Arrow
- Query-based
  - SQL
  - JSON

CSV

Column 1	Column 2	Column 3	Column 4	Column 5
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

- Lowest-common denominator format
- Flexible delimiters
- Ad hoc comments and headers
- Row-oriented
- Row-size can vary: no implicit indexing

```
...
390,1.83970e-003,-4.53930e-004,1.21520e-002
395,4.61530e-003,-1.04640e-003,3.11100e-002
400,9.62640e-003,-2.16890e-003,6.23710e-002
405,1.89790e-002,-4.43040e-003,1.31610e-001
410,3.08030e-002,-7.20480e-003,2.27500e-001
415,4.24590e-002,-1.25790e-002,3.58970e-001
420,5.16620e-002,-1.66510e-002,5.23960e-001
425,5.28370e-002,-2.12400e-002,6.85860e-001
...
```

```
...
390,1.83970e-003,-4.53930e-004,1.21520e-002
395,4.61530e-003,-1.04640e-003,3.11100e-002
400,9.62640e-003,-2.16890e-003,6.23710e-002
405,1.89790e-002,-4.43040e-003,1.31610e-001
410,3.08030e-002,-7.20480e-003,2.27500e-001
415,4.24590e-002,-1.25790e-002,3.58970e-001
420,5.16620e-002,-1.66510e-002,5.23960e-001
425,5.28370e-002,-2.12400e-002,6.85860e-001
...
```

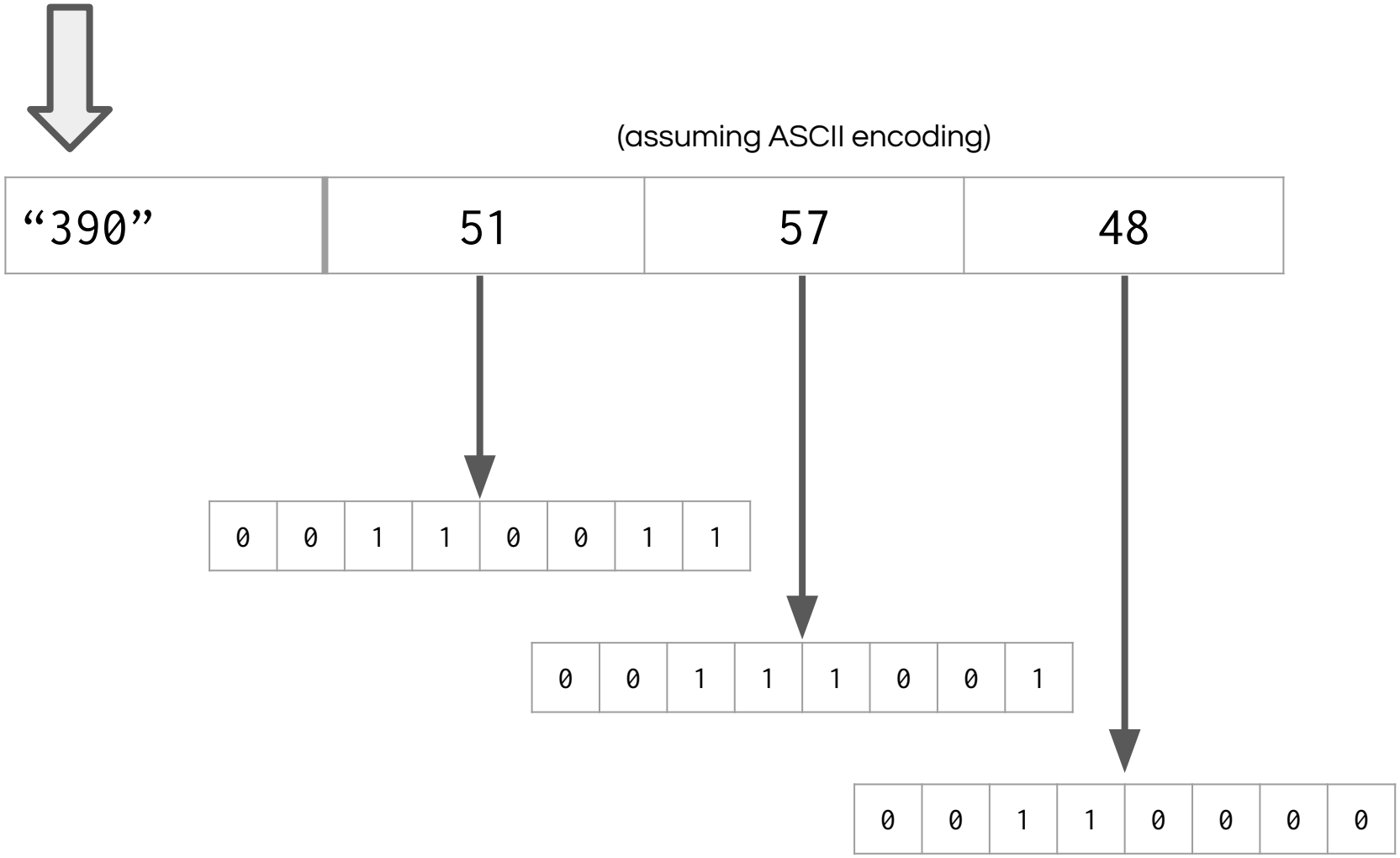
390,1.83970e-003,-4.53930e-004,1.21520e-002

390,1.83970e-003,-4.53930e-004,1.21520e-002

(assuming ASCII encoding)

“390”	51	57	48

390,1.83970e-003,-4.53930e-004,1.21520e-002



390,1.83970e-003,-4.53930e-004,1.21520e-002

(assuming ASCII encoding)

“390”	51	57	48

390.0		0	0	0	0	0	96	120	64

390,1.83970e-003,-4.53930e-004,1.21520e-002

(assuming ASCII encoding)

“1.83970e-003”	49	46	56	51	57	55	48	101	45	48	48	51

390,1.83970e-003,-4.53930e-004,1.21520e-002

(assuming ASCII encoding)

“1.83970e-003”	49	46	56	51	57	55	48	101	45	48	48	51

1.83970e-003		2	166	103	213	66	36	94	63

JSON

Record 1

Record 2

Record 3

- Row-oriented
- Potentially-unknown subcomponent sizes (lists of lists)
- Common response to REST APIs

- String
- Number
- Object (JSON)
- Array (list)
- Boolean
- null

[...
{"Agency Name":"University of Illinois",
 "Address":"501 E Daniel",
 "City":"Champaign",
 "Zip code":61820,
 "Year Acquired":1992,
 "Year Constructed":1935,
 "Square Footage":21845,
 "Total Floors":5}, ...
]

```
[...
{"Agency Name":"University of Illinois",
 "Address":"501 E Daniel",
 "City":"Champaign",
 "Zip code":61820,
 "Year Acquired":1992,
 "Year Constructed":1935,
 "Square Footage":21845,
 "Total Floors":5}, ...
]
```

Array

```
[...,
{"Agency Name":"University of Illinois",
 "Address":"501 E Daniel",
 "City":"Champaign",
 "Zip code":61820,
 "Year Acquired":1992,
 "Year Constructed":1935,
 "Square Footage":21845,
 "Total Floors":5}, ...
]
```

JSON

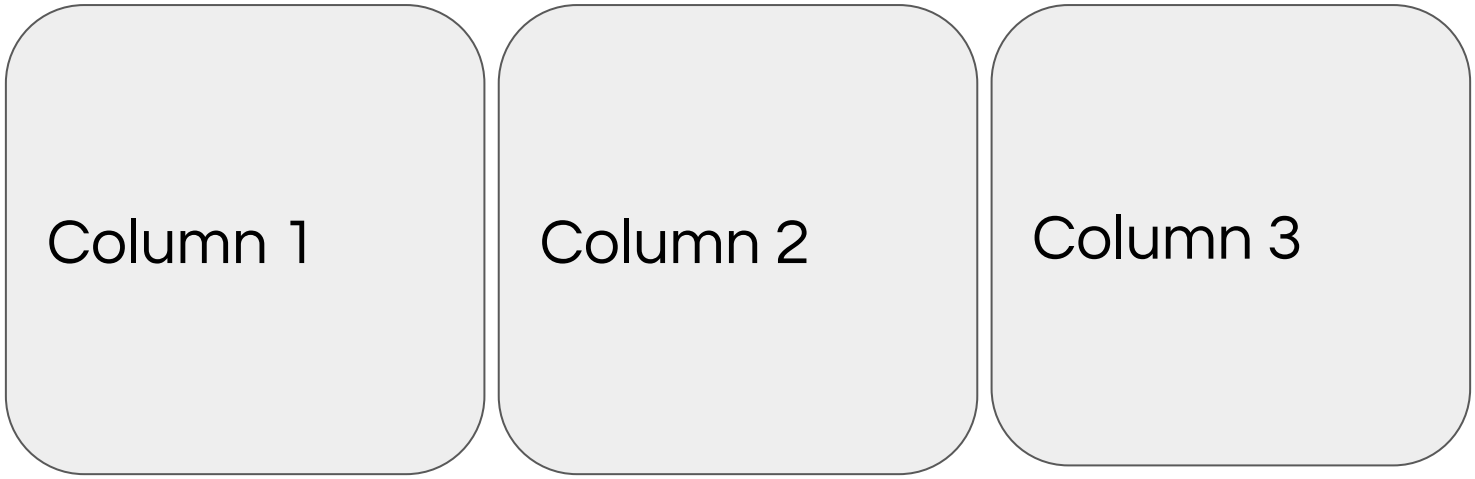
```
[...
{"Agency Name":"University of Illinois",
 "Address":"501 E Daniel",
 "City":"Champaign",
 "Zip code":61820,
 "Year Acquired":1992,
 "Year Constructed":1935,
 "Square Footage":21845,
 "Total Floors":5}, ...
]
```

String

```
[
{"Agency Name":"University of Illinois",
 "Address":"501 E Daniel",
 "City":"Champaign",
 "Zip code":61820,
 "Year Acquired":1992,
 "Year Constructed":1935,
 "Square Footage":21845,
 "Total Floors":5}, ...
]
```

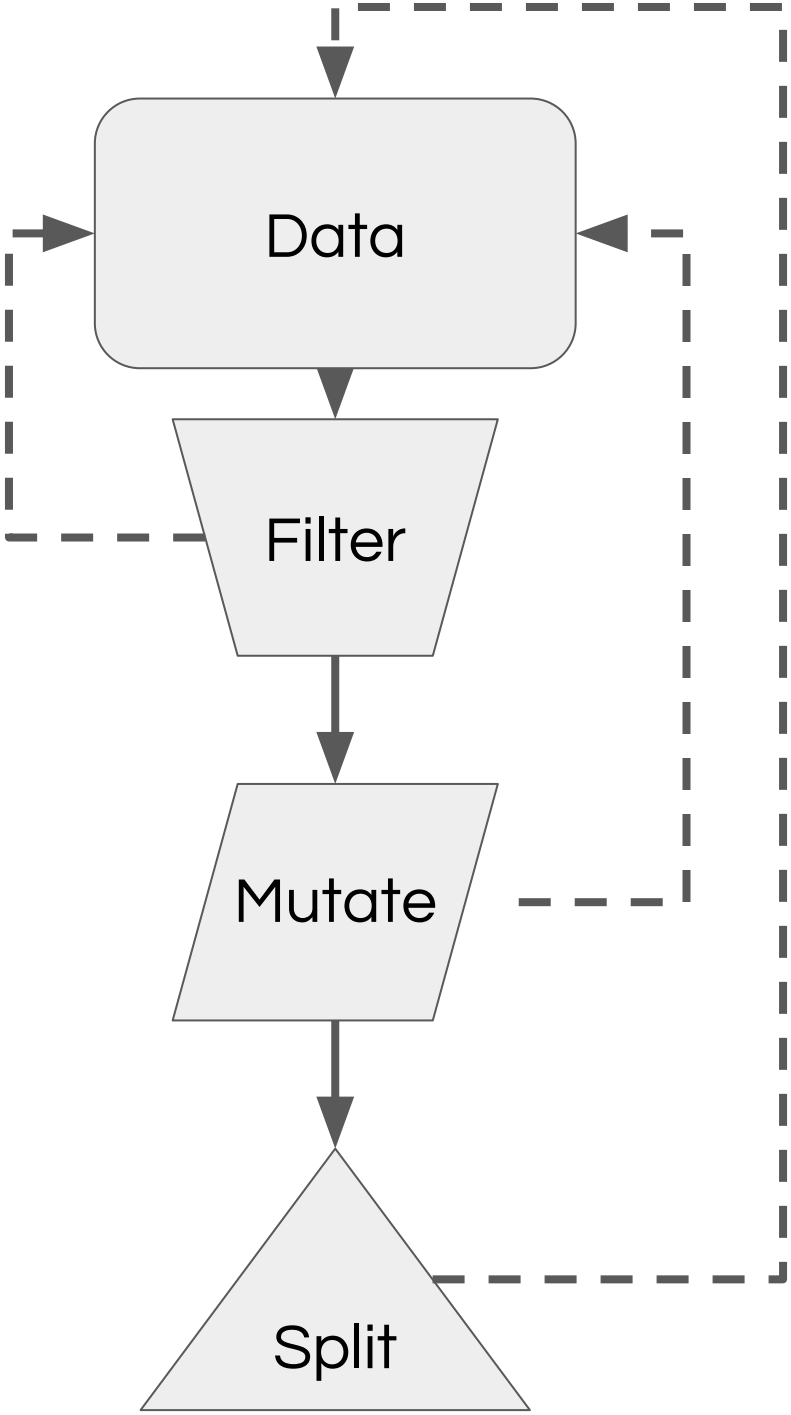

Number

HDF5



- Columnar store
- Chunking
- Can be extended
- Flexible data types in-memory and on-disk
- Hyperslab and boolean indexing

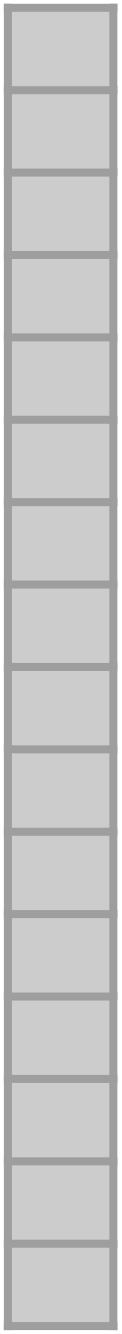
- Numeric
- Fixed-length strings
- Variable strings
- Groups & hierarchies
- Fine-grained key/val metadata



You have a palette of operations to apply.

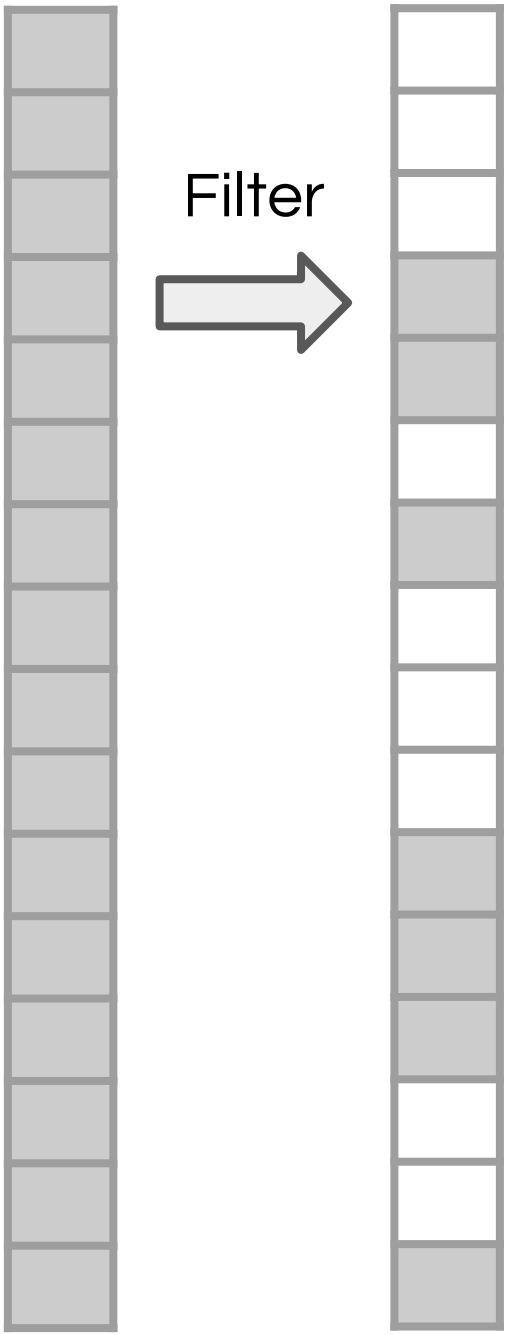
Filtering operations

- Relationships:
  - Equality, inequality
  - Quantitative value (less than, greater than)
  - Intersection, disjoint
- Subsampling
  - Regular sampling
  - Randomized sampling
  - Nyquist frequency
- Related data queries
  - Queries on other columns at fixed row location
  - External membership queries



Filtering operations

- Relationships:
  - Equality, inequality
  - Quantitative value (less than, greater than)
  - Intersection, disjoint
- Subsampling
  - Regular sampling
  - Randomized sampling
  - Nyquist frequency
- Related data queries
  - Queries on other columns at fixed row location
  - External membership queries



Relationships Examples

- Equality
  - Identity
  - Quantitative values
- Ordering or quantitative
  - Less than (or equal)
  - Greater than (or equal)
  - “Comes before” and “Comes after”
- Set-based operations
  - “Is a member”
  - “Is not a member”
  - “Shares members”
  - “Shares no members”

Equality Examples

```
value == “hello”
value == 10
```

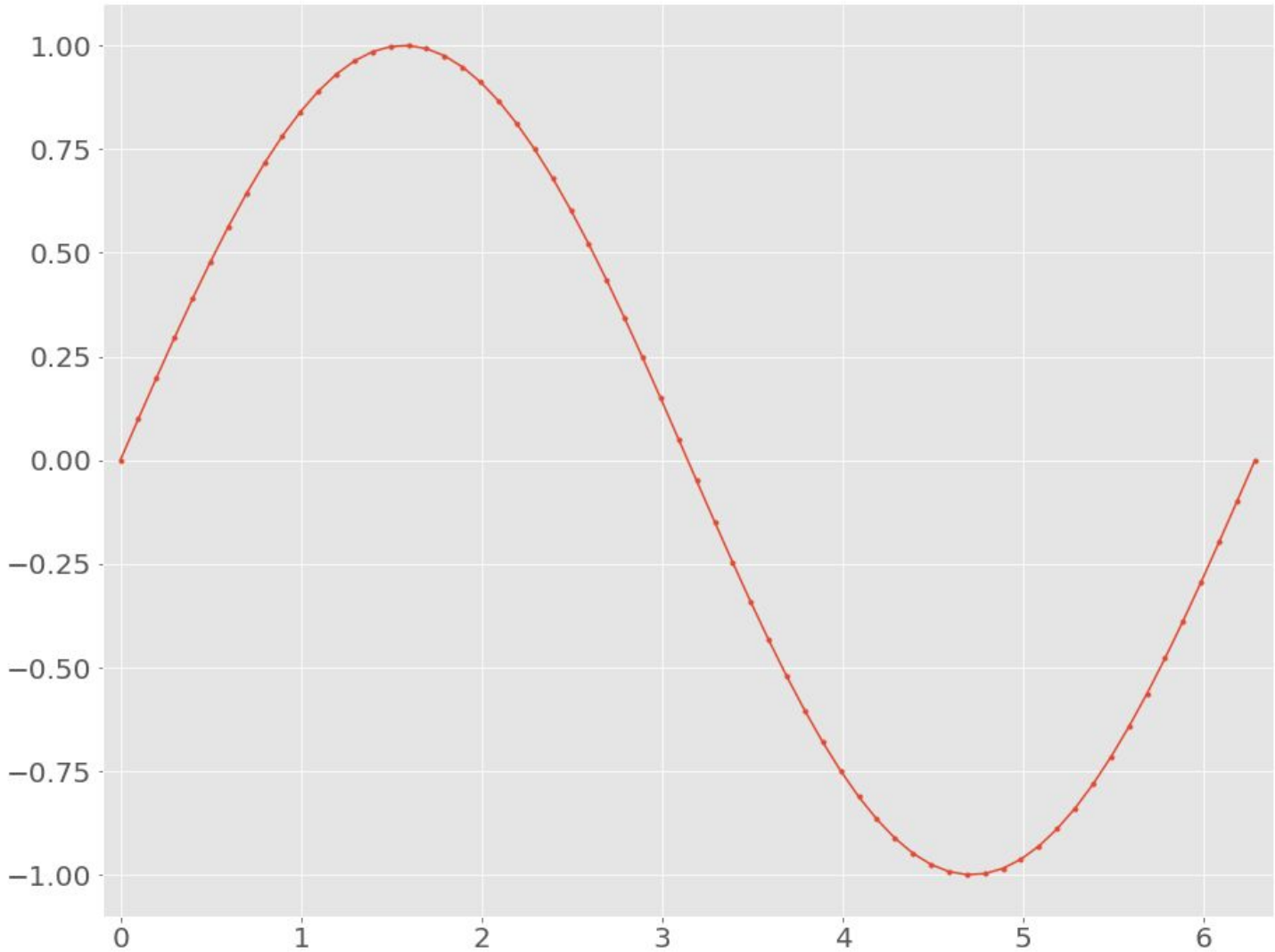
Ordering and Quantitative Examples

```
value < 30
value > July 1, 2010
```

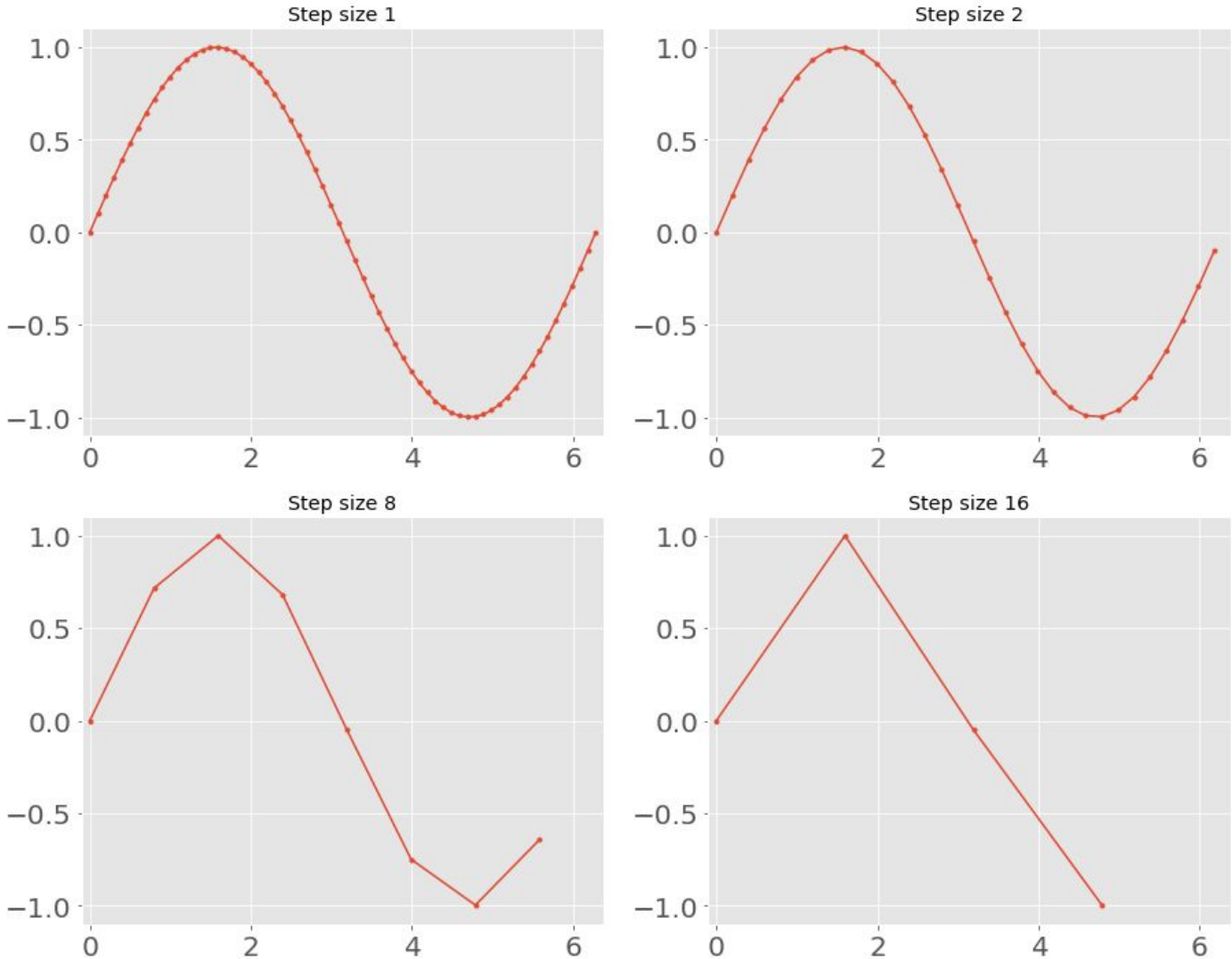
Set-Based Examples

```
value in (“red”, “blue”)
value not in (3.141, 2.7)
```

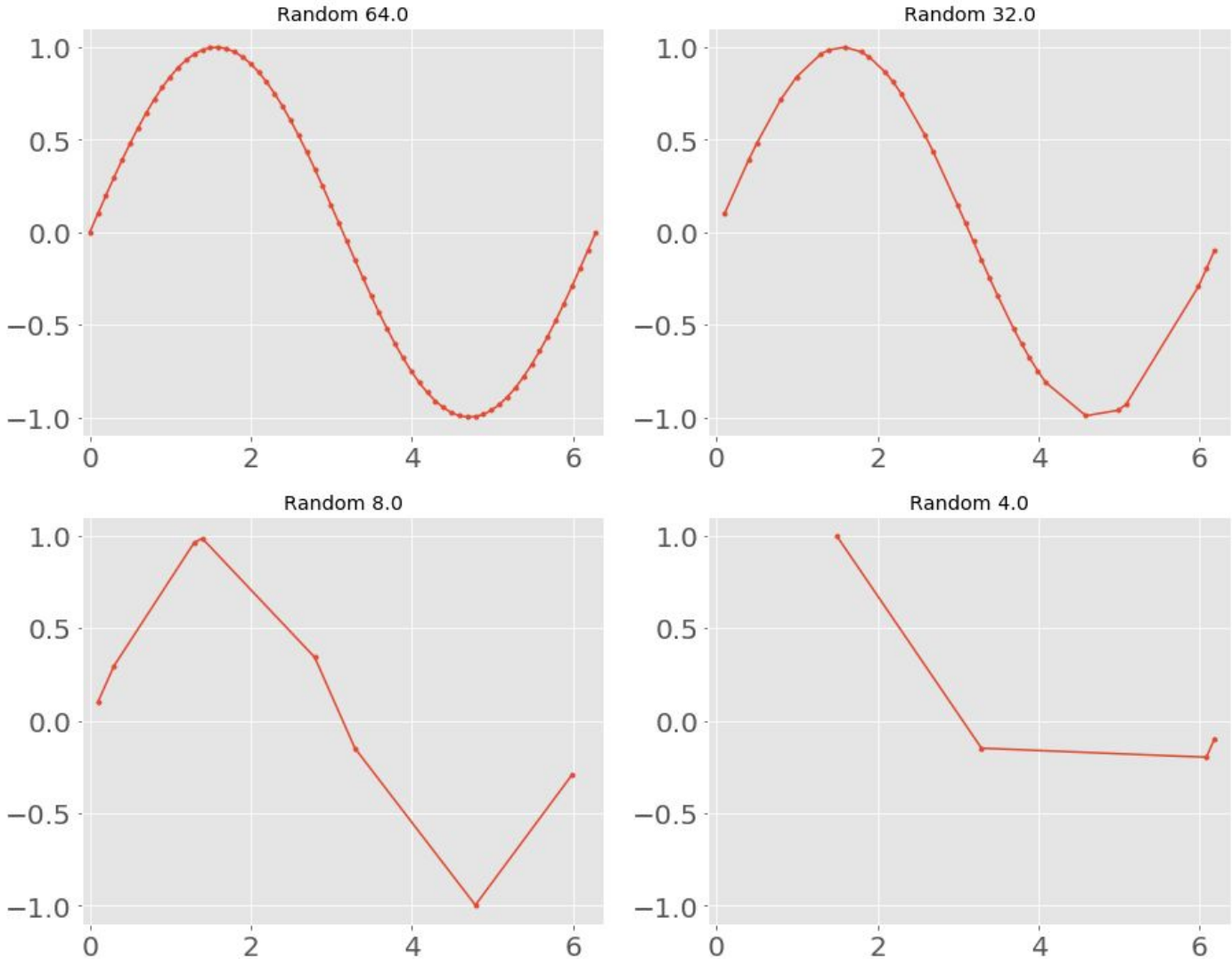
Sampling Examples



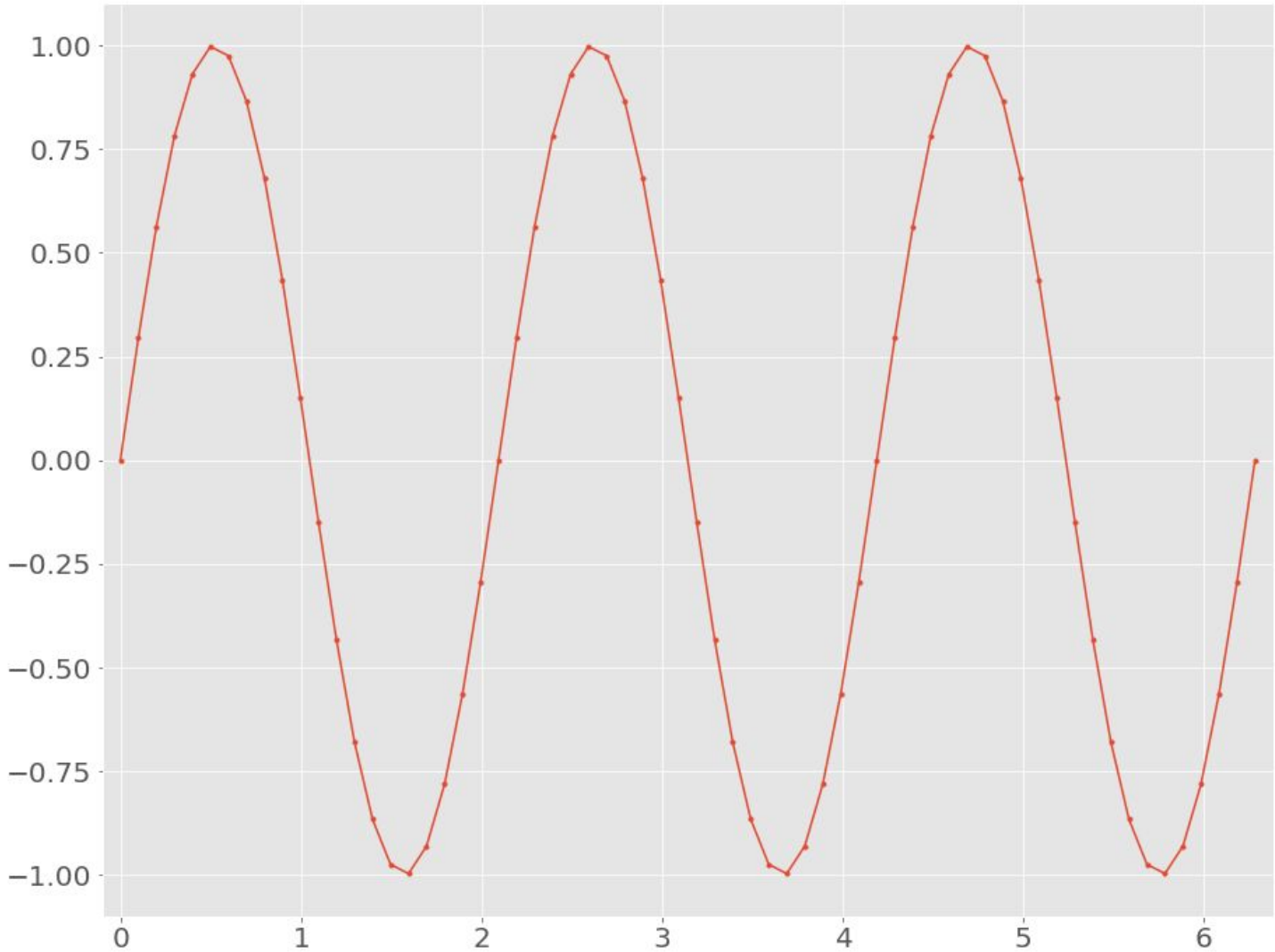
Sampling Examples



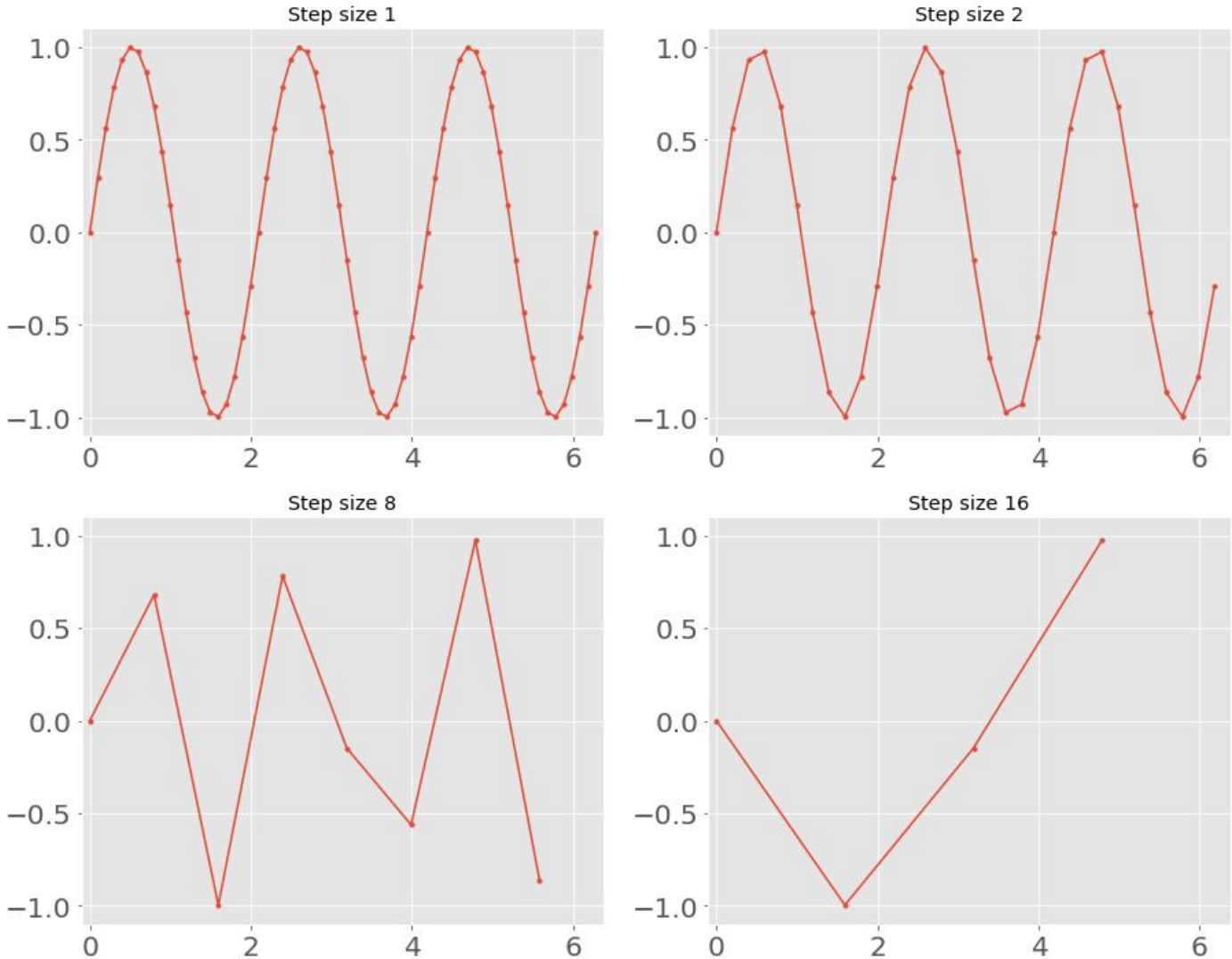
Sampling Examples



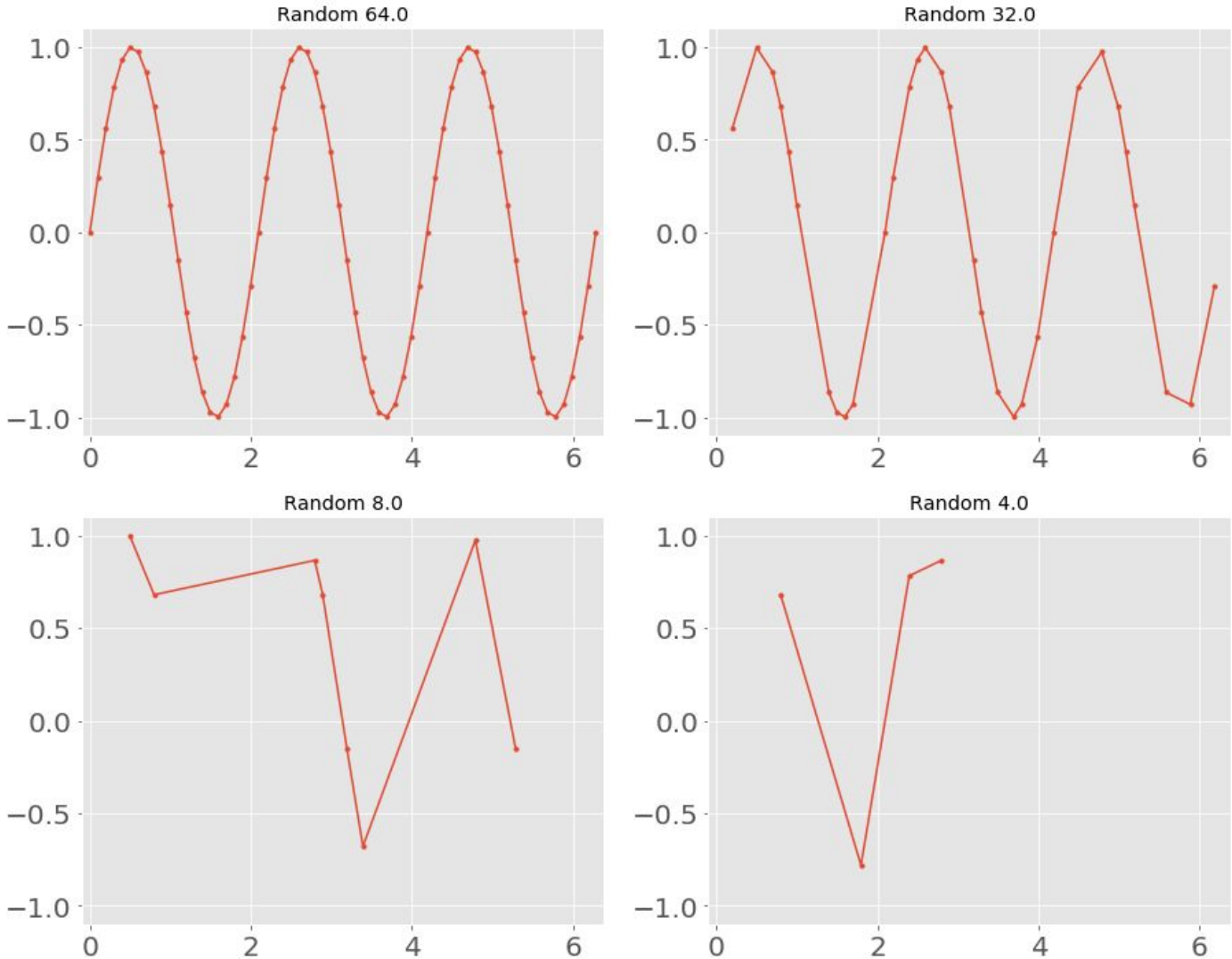
Sampling Examples



Sampling Examples



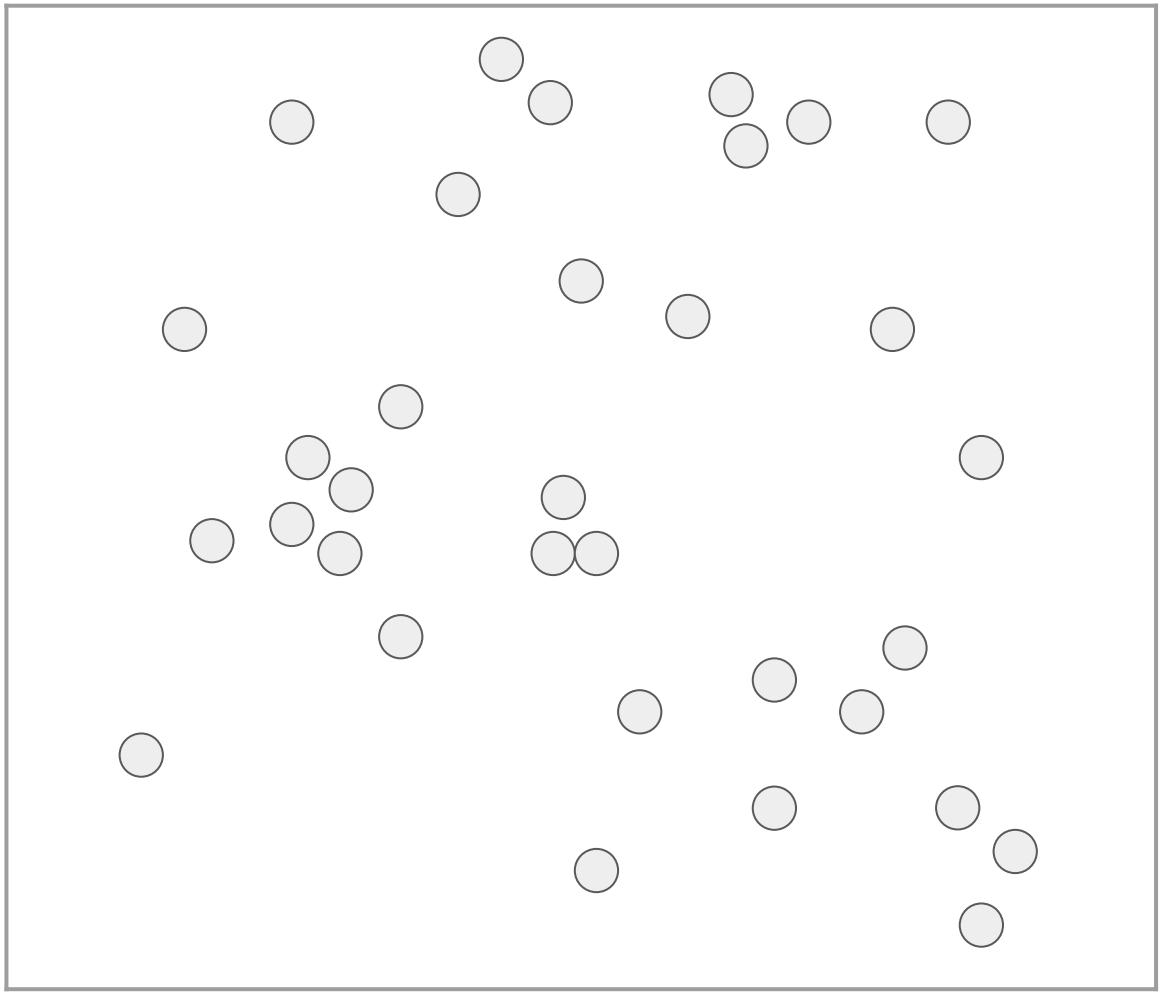
Sampling Examples



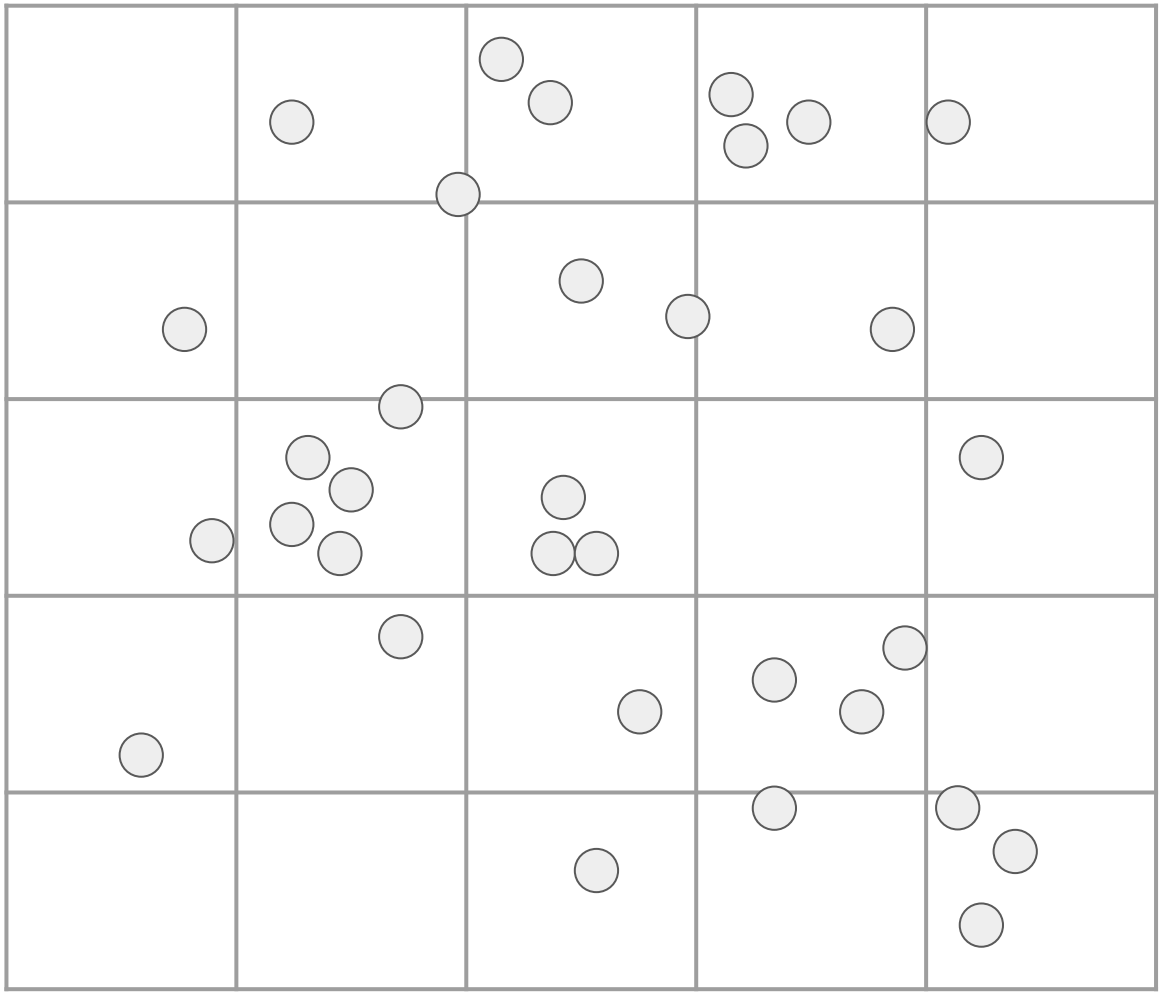
Mutation Operations

- Mathematical operations, such as injective operations.
  - Logarithmic versus linear representations
  - Arithmetic or multiplicative relationships
  - Manifold remapping
- Smoothing (reduction; not injective)
- Histograms (reduction; not injective)

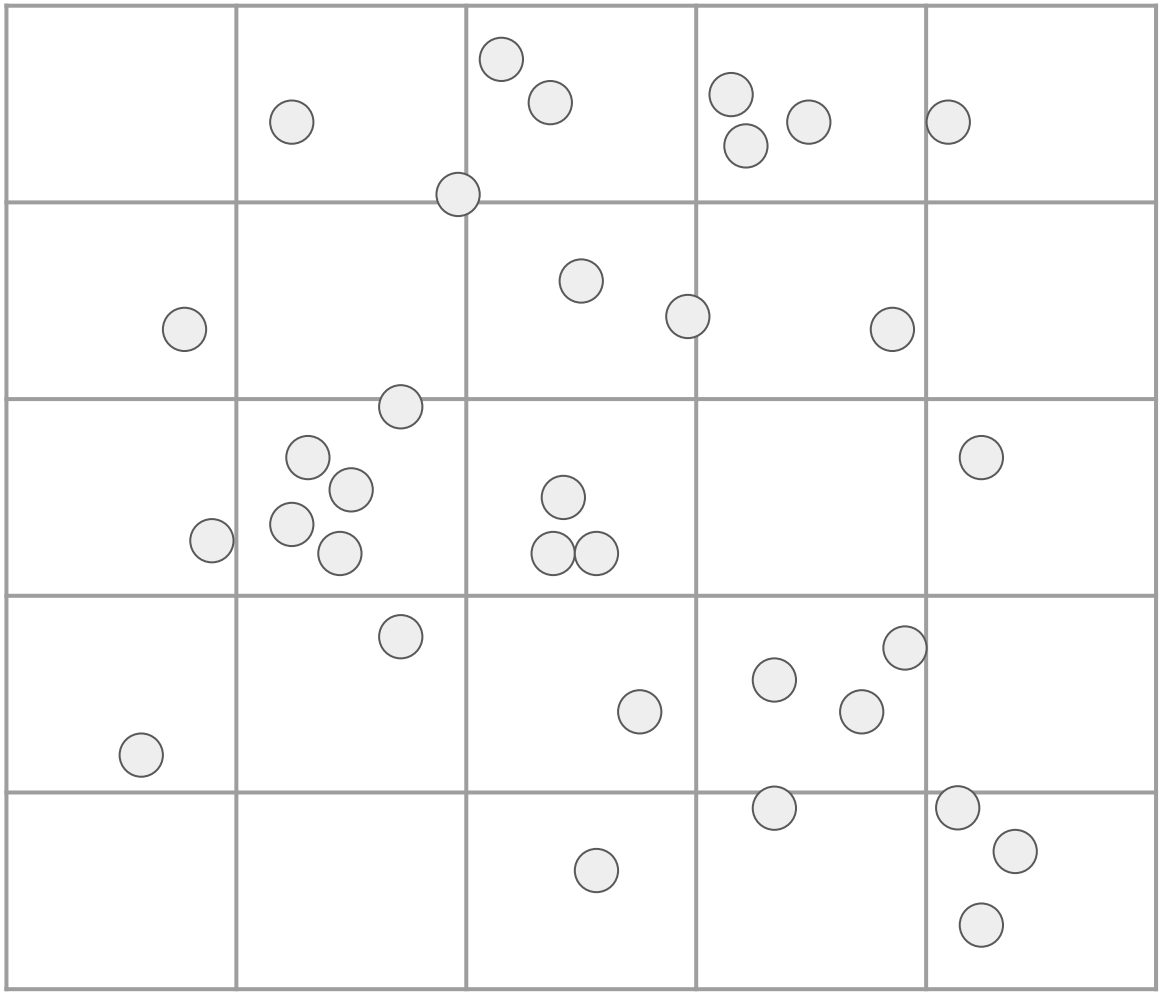
Binning and histograms



Binning and histograms



Binning and histograms



- Counts

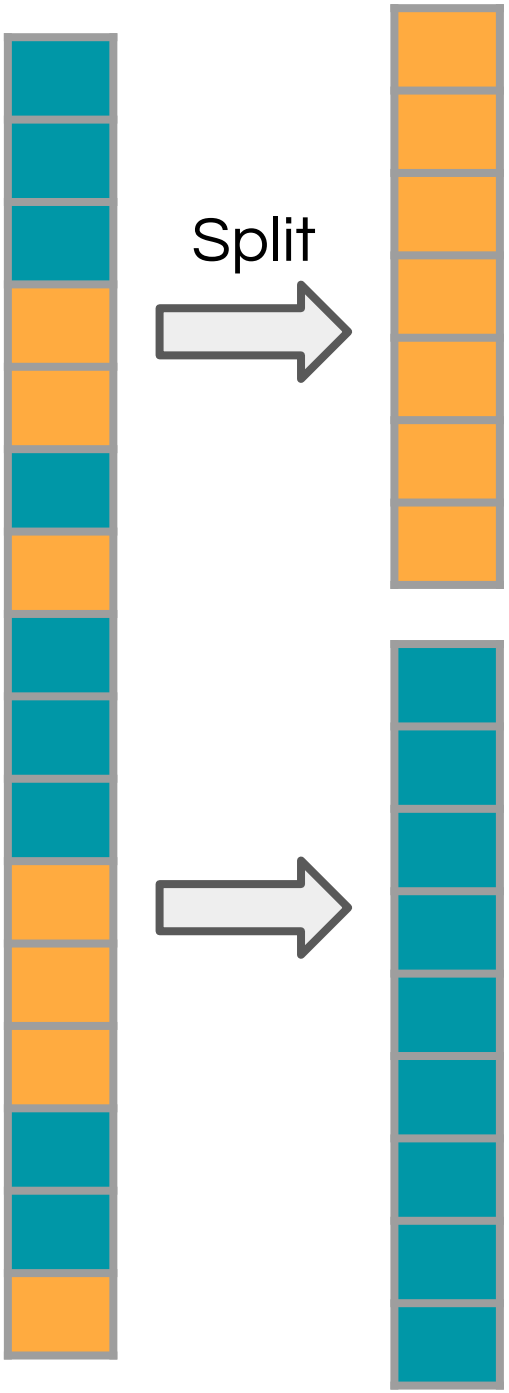
  sum(1)
- Sum

  sum(v_i)
- Average

  sum(v_i)/sum(1)
- Weighted Average

  sum(v_i * w_i) / sum(w_i)

Splitting Operations



Python Basics

- Variables
  - Strings, numbers, mutability
  - Assignment and comparisons
- Data Structures
  - Lists
  - Dicts
- Flow control
  - for / while
  - if / elif / else
  - Functions
- Packages
  - numpy
  - matplotlib

Variables

```
my_name = "Matt"

n_students = 7
n_students += 1

n_students_orig = n_students
n_students += 3
```

Data Structures

```
c = []
c.append(2)
c.append('hi there')

d = {}
d[1] = 'b'
d['hello'] = 10
```

Flow Control

```
for obj in [1, 2, 3]:
    print(obj)

a = []
while len(a) < 5:
    a.append(input("Hello!"))
```

Next Up

- Interactive time! Go to the JupyterHub.
  - “Jupyter”
  - Orientation in Python
  - Data structures
  - Iteration
  - Buildings owned by the state of Illinois
- Next week
  - Basic quantitative plotting
  - Components of a plot
  - Making, adjusting, and designing a visualization