Binning

Often, we want to break input data into categories called “buckets” or “bins”, then do stats (e.g., median) on each bucket

all data

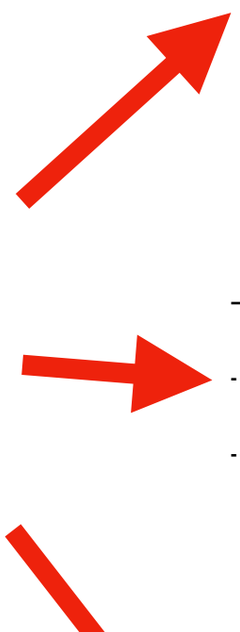
Year	ID	Speed
2014	A	123
2015	B	120
2015	C	140
2016	D	100
2015	E	130
2016	F	200

Binning

Often, we want to break input data into categories called "buckets" or "bins", then do stats (e.g., median) on each bucket

all data

Year	ID	Speed
2014	A	123
2015	B	120
2015	C	140
2016	D	100
2015	E	130
2016	F	200



bin for 2014

Year	ID	Speed
2014	A	123

bin for 2015

Year	ID	Speed
2015	B	120
2015	C	140
2015	E	130

bin for 2016

Year	ID	Speed
2016	D	100
2016	F	200

Binning

Often, we want to break input data into categories called "buckets" or "bins", then do stats (e.g., median) on each bucket

all data

Year	ID	Speed
2014	A	123
2015	B	120
2015	C	140
2016	D	100
2015	E	130
2016	F	200

bin for 2014

Year	ID	Speed
2014	A	123

→ ?

bin for 2015

Year	ID	Speed
2015	B	120
2015	C	140
2015	E	130

→ 130 (median)

bin for 2016

Year	ID	Speed
2016	D	100
2016	F	200

→ ?

Binning

Often, we want to break input data into categories called "buckets" or "bins", then do stats (e.g., median) on each bucket

all data

Year	ID	Speed
2014	A	123
2015	B	120
2015	C	140
2016	D	100
2015	E	130
2016	F	200

bin for 2014

Year	ID	Speed
2014	A	123

123 (median)

bin for 2015

Year	ID	Speed
2015	B	120
2015	C	140
2015	E	130

130 (median)

bin for 2016

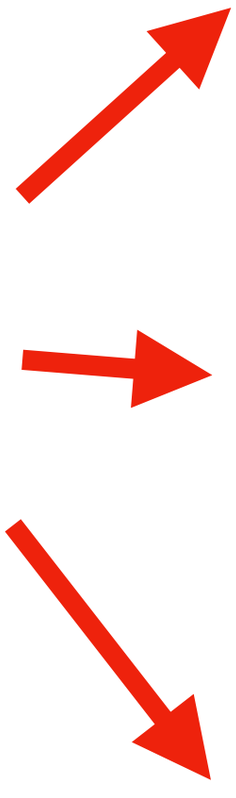
Year	ID	Speed
2016	D	100
2016	F	200

150 (median)

Bins with lists and dicts

all data

```
rows = [
    [2014, “A”, 123],
    [2015, “B”, 120],
    [2015, “C”, 140],
    [2016, “D”, 100],
    [2015, “E”, 130],
    [2016, “F”, 200],
]
```



bin for 2014

```
bin2014 = [
    [2014, “A”, 123],
]
```

bin for 2015

```
bin2015 = [
    [2015, “B”, 120],
    [2015, “C”, 140],
    [2015, “E”, 130],
]
```

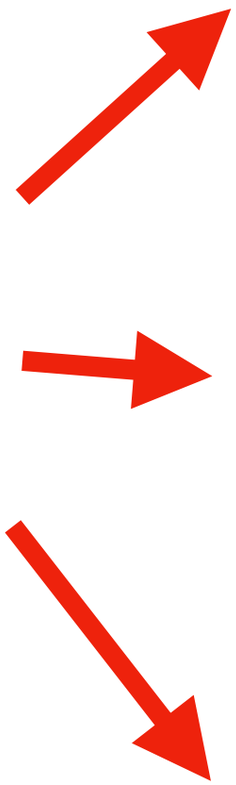
bin for 2016

```
bin2016 = [
    [2016, “D”, 100],
    [2016, “F”, 200],
]
```

Bins with lists and dicts

all data

```
rows = [
    [2014, “A”, 123],
    [2015, “B”, 120],
    [2015, “C”, 140],
    [2016, “D”, 100],
    [2015, “E”, 130],
    [2016, “F”, 200],
]
```



bin for 2014

```
bin2014 = [
    [2014, “A”, 123],
]
```

bin for 2015

```
bin2015 = [
    [2015, “B”, 120],
    [2015, “C”, 140],
    [2015, “E”, 130],
]
```

bin for 2016

```
bin2016 = [
    [2016, “D”, 100],
    [2016, “F”, 200],
]
```

how to keep track of all the lists?

Bins with lists and dicts

all data

```
rows = [
    [2014, “A”, 123],
    [2015, “B”, 120],
    [2015, “C”, 140],
    [2016, “D”, 100],
    [2015, “E”, 130],
    [2016, “F”, 200],
]
```

```
bins = {
    2014: [
        [2014, “A”, 123],
    ],
    2015: [
        [2015, “B”, 120],
        [2015, “C”, 140],
        [2015, “E”, 130],
    ],
    2016: [
        [2016, “D”, 100],
        [2016, “F”, 200],
    ]
}
```

Bins with lists and dicts

all data

```
rows = [
    [2014, “A”, 123],
    [2015, “B”, 120],
    [2015, “C”, 140],
    [2016, “D”, 100],
    [2015, “E”, 130],
    [2016, “F”, 200],
]
```

```
bins = {
    2014: [
        [2014, “A”, 123],
    ],
    2015: [
        [2015, “B”, 120],
        [2015, “C”, 140],
        [2015, “E”, 130],
    ],
    2016: [
        [2016, “D”, 100],
        [2016, “F”, 200],
    ]
}
```

Demo: Median Tornado Speed per Year

Goal: print median speed of tornados for each year

Input:

- Tornado CSV

Output:

- Median within each year

Example:

prompt> **python tornados.py**
…
2015: 130
2016: 123
2017: 90

Table Representation

name	x	y
Alice	30	20
Bob	5	11
Cindy	-2	50

list of list representation

```
header = [“name”, “x”, “y”]
rows = [
    [“Alice”, 30, 20],
    [“Bob”,   5,  11],
    [“Cindy”, -2, 50],
]
```

Table Representation

name	x	y
Alice	30	20
Bob	5	11
Cindy	-2	50

list of list representation

```
header = [“name”, “x”, “y”]
rows = [
    [“Alice”, 30, 20],
    [“Bob”,   5,  11],
    [“Cindy”, -2, 50],
]
```

list of dict representation

```
[
  {“name”:“Alice”, “x”:30, “y”:20”},
  {“name”:“Bob”,   “x”:5,  “y”:11”},
  {“name”:“Cindy”, “x”:-2, “y”:50”},
]
```

Table Representation

name	x	y
Alice	30	20
Bob	5	11
Cindy	-2	50

list of list representation

```
header = [“name”, “x”, “y”]
rows = [
    [“Alice”, 30, 20],
    [“Bob”,   5,  11],
    [“Cindy”, -2, 50],
]
```

list of dict representation

```
[
  {“name”:“Alice”, “x”:30, “y”:20”},
  {“name”:“Bob”,   “x”:5,  “y”:11”},
  {“name”:“Cindy”, “x”:-2, “y”:50”},
]
```

Table Representation

name	x	y
Alice	30	20
Bob	5	11
Cindy	-2	50

list of list representation

```
header = [“name”, “x”, “y”]
rows = [
    [“Alice”, 30, 20],
    [“Bob”,   5,  11],
    [“Cindy”, -2, 50],
]
```

rows[2][header.index(“y”)]

list of dict representation

```
[
  {“name”:“Alice”, “x”:30, “y”:20”},
  {“name”:“Bob”,   “x”:5,  “y”:11”},
  {“name”:“Cindy”, “x”:-2, “y”:50”},
]
```

Table Representation

name	x	y
Alice	30	20
Bob	5	11
Cindy	-2	50

list of list representation

```
header = [“name”, “x”, “y”]
rows = [
    [“Alice”, 30, 20],
    [“Bob”,   5,  11],
    [“Cindy”, -2, 50],
]
```

rows[2][header.index(“y”)]

list of dict representation

```
[
  {“name”:“Alice”, “x”:30, “y”:20”},
  {“name”:“Bob”,   “x”:5,  “y”:11”},
  {“name”:“Cindy”, “x”:-2, “y”:50”},
]
```

rows[2][“y”]

Demo: Table Transform

Goal: create function that transforms list of lists table
to a list of dicts table

Input:

- List of lists (from a CSV)

Output:

- List of dicts

Example:

```
>>> header = [“x”,”y”]
>>> rows = [[1,2], [3,4]]
>>> transform(header, rows)
[{“x”:1, “y”:2}, {“x”:3, “y”:4}]
```