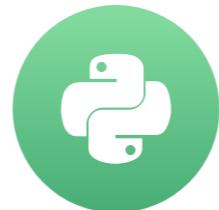


Introducing DataFrames

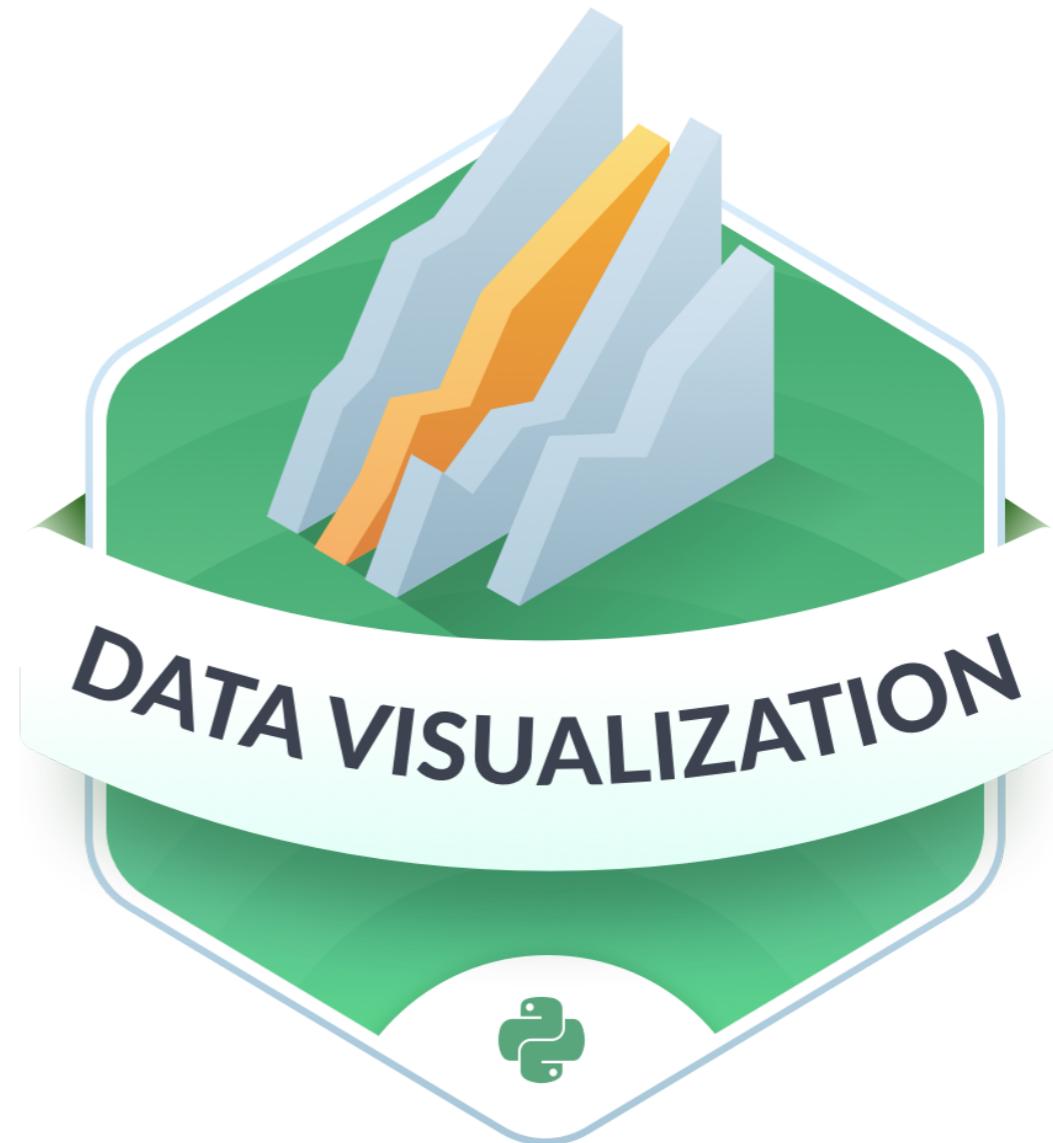
DATA MANIPULATION WITH PANDAS



Richie Cotton

Curriculum Architect at DataCamp

What's the point of pandas?



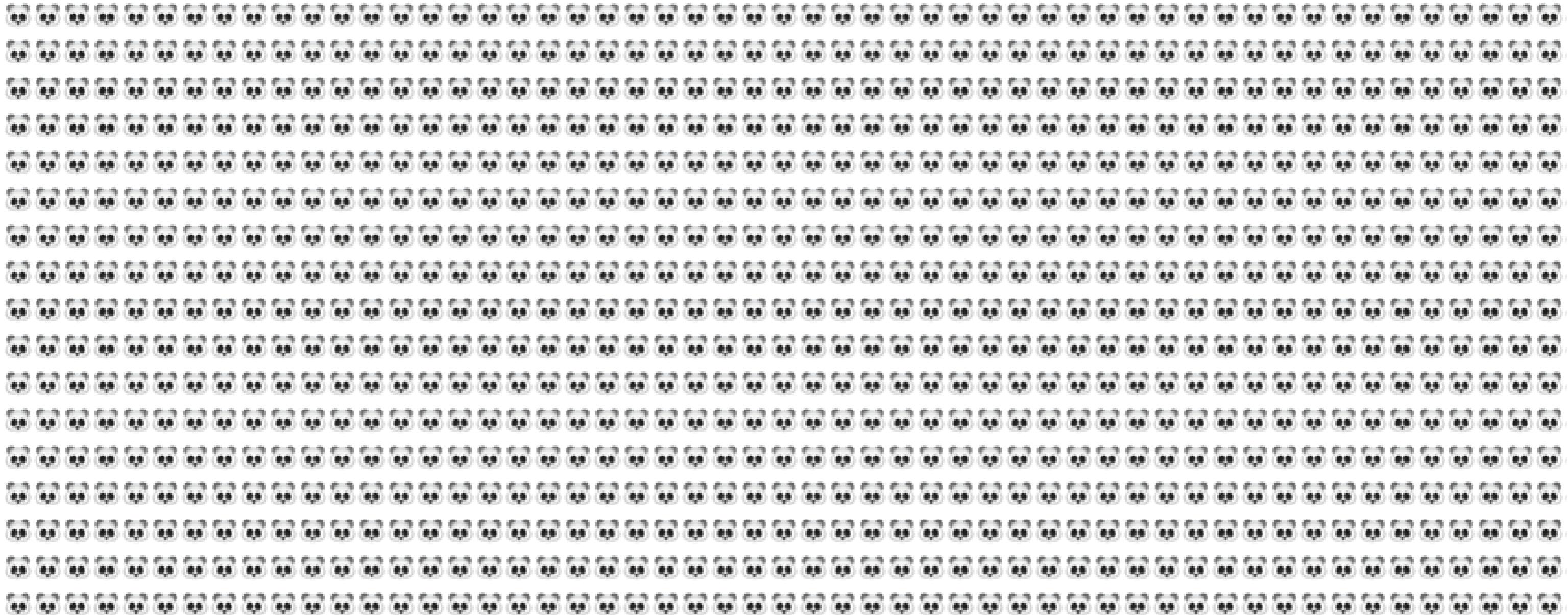
Course outline

- **Chapter 1: DataFrames**
 - Sorting and subsetting
 - Creating new columns
- **Chapter 2: Aggregating Data**
 - Summary statistics
 - Counting
 - Grouped summary statistics
- **Chapter 3: Slicing and Indexing Data**
 - Subsetting using slicing
 - Indexes and subsetting using indexes
- **Chapter 4: Creating and Visualizing Data**
 - Plotting
 - Handling missing data
 - Reading data into a DataFrame

pandas is built on NumPy and Matplotlib



pandas is popular



¹ <https://pypistats.org/packages/pandas>

Rectangular data

Name	Breed	Color	Height (cm)	Weight (kg)	Date of Birth
Bella	Labrador	Brown	56	25	2013-07-01
Charlie	Poodle	Black	43	23	2016-09-16
Lucy	Chow Chow	Brown	46	22	2014-08-25
Cooper	Schnauzer	Gray	49	17	2011-12-11
Max	Labrador	Black	59	29	2017-01-20
Stella	Chihuahua	Tan	18	2	2015-04-20
Bernie	St. Bernard	White	77	74	2018-02-27

pandas DataFrames

```
print(dogs)
```

	name	breed	color	height_cm	weight_kg	date_of_birth
0	Bella	Labrador	Brown	56	24	2013-07-01
1	Charlie	Poodle	Black	43	24	2016-09-16
2	Lucy	Chow Chow	Brown	46	24	2014-08-25
3	Cooper	Schnauzer	Gray	49	17	2011-12-11
4	Max	Labrador	Black	59	29	2017-01-20
5	Stella	Chihuahua	Tan	18	2	2015-04-20
6	Bernie	St. Bernard	White	77	74	2018-02-27

Exploring a DataFrame: .head()

```
dogs.head()
```

	name	breed	color	height_cm	weight_kg	date_of_birth
0	Bella	Labrador	Brown	56	24	2013-07-01
1	Charlie	Poodle	Black	43	24	2016-09-16
2	Lucy	Chow Chow	Brown	46	24	2014-08-25
3	Cooper	Schnauzer	Gray	49	17	2011-12-11
4	Max	Labrador	Black	59	29	2017-01-20

Exploring a DataFrame: .info()

```
dogs.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 7 entries, 0 to 6  
Data columns (total 6 columns):  
 name            7 non-null object  
 breed           7 non-null object  
 color           7 non-null object  
 height_cm       7 non-null int64  
 weight_kg       7 non-null int64  
 date_of_birth   7 non-null object  
 dtypes: int64(2), object(4)
```

Exploring a DataFrame: .shape

```
dogs.shape
```

```
(7, 6)
```

Exploring a DataFrame: .describe()

```
dogs.describe()
```

```
height_cm    weight_kg
count      7.000000    7.000000
mean      49.714286   27.428571
std       17.960274   22.292429
min      18.000000    2.000000
25%      44.500000   19.500000
50%      49.000000   23.000000
75%      57.500000   27.000000
max      77.000000   74.000000
```

Components of a DataFrame: .values

```
dogs.values
```

```
array([['Bella', 'Labrador', 'Brown', 56, 24, '2013-07-01'],
       ['Charlie', 'Poodle', 'Black', 43, 24, '2016-09-16'],
       ['Lucy', 'Chow Chow', 'Brown', 46, 24, '2014-08-25'],
       ['Cooper', 'Schnauzer', 'Gray', 49, 17, '2011-12-11'],
       ['Max', 'Labrador', 'Black', 59, 29, '2017-01-20'],
       ['Stella', 'Chihuahua', 'Tan', 18, 2, '2015-04-20'],
       ['Bernie', 'St. Bernard', 'White', 77, 74, '2018-02-27']],
      dtype=object)
```

Components of a DataFrame: .columns and .index

```
dogs.columns
```

```
Index(['name', 'breed', 'color', 'height_cm', 'weight_kg', 'date_of_birth'],  
      dtype='object')
```

```
dogs.index
```

```
RangeIndex(start=0, stop=7, step=1)
```

pandas Philosophy

There should be one -- and preferably only one -- obvious way to do it.

- *The Zen of Python* by Tim Peters, Item 13



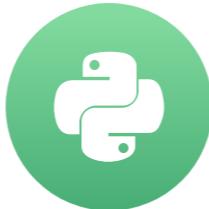
¹ <https://www.python.org/dev/peps/pep-0020/>

Let's practice!

DATA MANIPULATION WITH PANDAS

Sorting and subsetting

DATA MANIPULATION WITH PANDAS



Richie Cotton

Curriculum Architect at DataCamp

Sorting

```
dogs.sort_values("weight_kg")
```

		name	breed	color	height_cm	weight_kg	date_of_birth
5	Stella	Chihuahua	Tan		18	2	2015-04-20
3	Cooper	Schnauzer	Gray		49	17	2011-12-11
0	Bella	Labrador	Brown		56	24	2013-07-01
1	Charlie	Poodle	Black		43	24	2016-09-16
2	Lucy	Chow Chow	Brown		46	24	2014-08-25
4	Max	Labrador	Black		59	29	2017-01-20
6	Bernie	St. Bernard	White		77	74	2018-02-27

Sorting in descending order

```
dogs.sort_values("weight_kg", ascending=False)
```

		name	breed	color	height_cm	weight_kg	date_of_birth
6	Bernie	St. Bernard	White		77	74	2018-02-27
4	Max	Labrador	Black		59	29	2017-01-20
0	Bella	Labrador	Brown		56	24	2013-07-01
1	Charlie	Poodle	Black		43	24	2016-09-16
2	Lucy	Chow Chow	Brown		46	24	2014-08-25
3	Cooper	Schnauzer	Gray		49	17	2011-12-11
5	Stella	Chihuahua	Tan		18	2	2015-04-20

Sorting by multiple variables

```
dogs.sort_values(["weight_kg", "height_cm"])
```

		name	breed	color	height_cm	weight_kg	date_of_birth
5	Stella	Chihuahua	Tan		18	2	2015-04-20
3	Cooper	Schnauzer	Gray		49	17	2011-12-11
1	Charlie	Poodle	Black		43	24	2016-09-16
2	Lucy	Chow Chow	Brown		46	24	2014-08-25
0	Bella	Labrador	Brown		56	24	2013-07-01
4	Max	Labrador	Black		59	29	2017-01-20
6	Bernie	St. Bernard	White		77	74	2018-02-27

Sorting by multiple variables

```
dogs.sort_values(["weight_kg", "height_cm"], ascending=[True, False])
```

		name	breed	color	height_cm	weight_kg	date_of_birth
5	Stella	Chihuahua	Tan		18	2	2015-04-20
3	Cooper	Schnauzer	Gray		49	17	2011-12-11
0	Bella	Labrador	Brown		56	24	2013-07-01
2	Lucy	Chow Chow	Brown		46	24	2014-08-25
1	Charlie	Poodle	Black		43	24	2016-09-16
4	Max	Labrador	Black		59	29	2017-01-20
6	Bernie	St. Bernard	White		77	74	2018-02-27

Subsetting columns

```
dogs[ "name" ]
```

```
0      Bella
1    Charlie
2      Lucy
3   Cooper
4      Max
5    Stella
6    Bernie
Name: name, dtype: object
```

Subsetting multiple columns

```
dogs[["breed", "height_cm"]]
```

```
breed  height_cm  
0      Labrador      56  
1      Poodle        43  
2      Chow Chow     46  
3      Schnauzer     49  
4      Labrador      59  
5      Chihuahua     18  
6      St. Bernard    77
```

```
cols_to_subset = ["breed", "height_cm"]  
dogs[cols_to_subset]
```

```
breed  height_cm  
0      Labrador      56  
1      Poodle        43  
2      Chow Chow     46  
3      Schnauzer     49  
4      Labrador      59  
5      Chihuahua     18  
6      St. Bernard    77
```

Subsetting rows

```
dogs["height_cm"] > 50
```

```
0    True
1   False
2   False
3   False
4    True
5   False
6    True
Name: height_cm, dtype: bool
```

Subsetting rows

```
dogs[dogs["height_cm"] > 50]
```

	name	breed	color	height_cm	weight_kg	date_of_birth
0	Bella	Labrador	Brown	56	24	2013-07-01
4	Max	Labrador	Black	59	29	2017-01-20
6	Bernie	St. Bernard	White	77	74	2018-02-27

Subsetting based on text data

```
dogs[dogs["breed"] == "Labrador"]
```

	name	breed	color	height_cm	weight_kg	date_of_birth
0	Bella	Labrador	Brown	56	24	2013-07-01
4	Max	Labrador	Black	59	29	2017-01-20

Subsetting based on dates

```
dogs[dogs["date_of_birth"] > "2015-01-01"]
```

	name	breed	color	height_cm	weight_kg	date_of_birth
1	Charlie	Poodle	Black	43	24	2016-09-16
4	Max	Labrador	Black	59	29	2017-01-20
5	Stella	Chihuahua	Tan	18	2	2015-04-20
6	Bernie	St. Bernard	White	77	74	2018-02-27

Subsetting based on multiple conditions

```
is_lab = dogs["breed"] == "Labrador"  
is_brown = dogs["color"] == "Brown"  
dogs[is_lab & is_brown]
```

```
   name      breed  color  height_cm  weight_kg  date_of_birth  
0  Bella    Labrador  Brown        56         24  2013-07-01
```

```
dogs[ (dogs["breed"] == "Labrador") & (dogs["color"] == "Brown") ]
```

Subsetting using .isin()

```
is_black_or_brown = dogs[ "color" ].isin([ "Black" , "Brown" ])  
dogs[is_black_or_brown]
```

	name	breed	color	height_cm	weight_kg	date_of_birth
0	Bella	Labrador	Brown	56	24	2013-07-01
1	Charlie	Poodle	Black	43	24	2016-09-16
2	Lucy	Chow Chow	Brown	46	24	2014-08-25
4	Max	Labrador	Black	59	29	2017-01-20

Let's practice!

DATA MANIPULATION WITH PANDAS

New columns

DATA MANIPULATION WITH PANDAS



Richie Cotton

Curriculum Architect at DataCamp

Adding a new column

```
dogs["height_m"] = dogs["height_cm"] / 100  
print(dogs)
```

	name	breed	color	height_cm	weight_kg	date_of_birth	height_m
0	Bella	Labrador	Brown	56	24	2013-07-01	0.56
1	Charlie	Poodle	Black	43	24	2016-09-16	0.43
2	Lucy	Chow Chow	Brown	46	24	2014-08-25	0.46
3	Cooper	Schnauzer	Gray	49	17	2011-12-11	0.49
4	Max	Labrador	Black	59	29	2017-01-20	0.59
5	Stella	Chihuahua	Tan	18	2	2015-04-20	0.18
6	Bernie	St. Bernard	White	77	74	2018-02-27	0.77

Doggy mass index

$$\text{BMI} = \text{weight in kg}/(\text{height in m})^2$$

```
dogs["bmi"] = dogs["weight_kg"] / dogs["height_m"] ** 2  
print(dogs.head())
```

	name	breed	color	height_cm	weight_kg	date_of_birth	height_m	bmi
0	Bella	Labrador	Brown	56	24	2013-07-01	0.56	76.530612
1	Charlie	Poodle	Black	43	24	2016-09-16	0.43	129.799892
2	Lucy	Chow Chow	Brown	46	24	2014-08-25	0.46	113.421550
3	Cooper	Schnauzer	Gray	49	17	2011-12-11	0.49	70.803832
4	Max	Labrador	Black	59	29	2017-01-20	0.59	83.309394

Multiple manipulations

```
bmi_lt_100 = dogs[dogs["bmi"] < 100]  
  
bmi_lt_100_height = bmi_lt_100.sort_values("height_cm", ascending=False)  
  
bmi_lt_100_height[["name", "height_cm", "bmi"]]
```

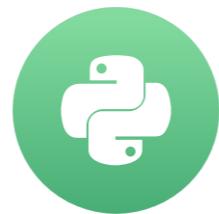
```
   name  height_cm      bmi  
4    Max       59  83.309394  
0    Bella      56  76.530612  
3   Cooper      49  70.803832  
5   Stella      18  61.728395
```

Let's practice!

DATA MANIPULATION WITH PANDAS

Summary statistics

DATA MANIPULATION WITH PANDAS



Maggie Matsui

Senior Content Developer at DataCamp

Summarizing numerical data

```
dogs["height_cm"].mean()
```

```
49.714285714285715
```

- `.median()` , `.mode()`
- `.min()` , `.max()`
- `.var()` , `.std()`
- `.sum()`
- `.quantile()`

Summarizing dates

Oldest dog:

```
dogs["date_of_birth"].min()
```

```
'2011-12-11'
```

Youngest dog:

```
dogs["date_of_birth"].max()
```

```
'2018-02-27'
```

The .agg() method

```
def pct30(column):  
    return column.quantile(0.3)
```

```
dogs["weight_kg"].agg(pct30)
```

```
22.599999999999998
```

Summaries on multiple columns

```
dogs[["weight_kg", "height_cm"]].agg(pct30)
```

```
weight_kg    22.6
height_cm    45.4
dtype: float64
```

Multiple summaries

```
def pct40(column):  
    return column.quantile(0.4)
```

```
dogs["weight_kg"].agg([pct30, pct40])
```

```
pct30    22.6  
pct40    24.0  
Name: weight_kg, dtype: float64
```

Cumulative sum

```
dogs["weight_kg"]
```

```
0    24  
1    24  
2    24  
3    17  
4    29  
5     2  
6   74  
  
Name: weight_kg, dtype: int64
```

```
dogs["weight_kg"].cumsum()
```

```
0    24  
1    48  
2    72  
3    89  
4   118  
5   120  
6   194  
  
Name: weight_kg, dtype: int64
```

Cumulative statistics

- `.cummax()`
- `.cummin()`
- `.cumprod()`

Walmart

```
sales.head()
```

	store	type	dept	date	weekly_sales	is_holiday	temp_c	fuel_price	unemp
0	1	A	1	2010-02-05	24924.50	False	5.73	0.679	8.106
1	1	A	2	2010-02-05	50605.27	False	5.73	0.679	8.106
2	1	A	3	2010-02-05	13740.12	False	5.73	0.679	8.106
3	1	A	4	2010-02-05	39954.04	False	5.73	0.679	8.106
4	1	A	5	2010-02-05	32229.38	False	5.73	0.679	8.106

Let's practice!

DATA MANIPULATION WITH PANDAS

Counting

DATA MANIPULATION WITH PANDAS



Maggie Matsui

Senior Content Developer at DataCamp

Avoiding double counting



Vet visits

```
print(vet_visits)
```

```
      date      name     breed  weight_kg
0  2018-09-02    Bella  Labrador     24.87
1  2019-06-07     Max  Labrador     28.35
2  2018-01-17   Stella Chihuahua     1.51
3  2019-10-19    Lucy  Chow Chow     24.07
...
71  2018-01-20   Stella Chihuahua     2.83
72  2019-06-07     Max  Chow Chow     24.01
73  2018-08-20    Lucy  Chow Chow     24.40
74  2019-04-22     Max  Labrador     28.54
```

Dropping duplicate names

```
vet_visits.drop_duplicates(subset="name")
```

	date	name	breed	weight_kg
0	2018-09-02	Bella	Labrador	24.87
1	2019-06-07	Max	Chow Chow	24.01
2	2019-03-19	Charlie	Poodle	24.95
3	2018-01-17	Stella	Chihuahua	1.51
4	2019-10-19	Lucy	Chow Chow	24.07
7	2019-03-30	Cooper	Schnauzer	16.91
10	2019-01-04	Bernie	St. Bernard	74.98
(6	2019-06-07	Max	Labrador	28.35)

Dropping duplicate pairs

```
unique_dogs = vet_visits.drop_duplicates(subset=[ "name" , "breed" ])  
print(unique_dogs)
```

	date	name	breed	weight_kg
0	2018-09-02	Bella	Labrador	24.87
1	2019-03-13	Max	Chow Chow	24.13
2	2019-03-19	Charlie	Poodle	24.95
3	2018-01-17	Stella	Chihuahua	1.51
4	2019-10-19	Lucy	Chow Chow	24.07
6	2019-06-07	Max	Labrador	28.35
7	2019-03-30	Cooper	Schnauzer	16.91
10	2019-01-04	Bernie	St. Bernard	74.98

Easy as 1, 2, 3

```
unique_dogs["breed"].value_counts()
```

```
Labrador      2  
Schnauzer     1  
St. Bernard    1  
Chow Chow      2  
Poodle         1  
Chihuahua      1  
Name: breed, dtype: int64
```

```
unique_dogs["breed"].value_counts(sort=True)
```

```
Labrador      2  
Chow Chow      2  
Schnauzer     1  
St. Bernard    1  
Poodle         1  
Chihuahua      1  
Name: breed, dtype: int64
```

Proportions

```
unique_dogs[ "breed" ].value_counts(normalize=True)
```

```
Labrador      0.250
Chow Chow     0.250
Schnauzer     0.125
St. Bernard   0.125
Poodle        0.125
Chihuahua    0.125
Name: breed, dtype: float64
```

Let's practice!

DATA MANIPULATION WITH PANDAS

Grouped summary statistics

DATA MANIPULATION WITH PANDAS



Maggie Matsui

Senior Content Developer at DataCamp

Summaries by group

```
dogs[dogs["color"] == "Black"]["weight_kg"].mean()  
dogs[dogs["color"] == "Brown"]["weight_kg"].mean()  
dogs[dogs["color"] == "White"]["weight_kg"].mean()  
dogs[dogs["color"] == "Gray"]["weight_kg"].mean()  
dogs[dogs["color"] == "Tan"]["weight_kg"].mean()
```

```
26.0  
24.0  
74.0  
17.0  
2.0
```

Grouped summaries

```
dogs.groupby("color")["weight_kg"].mean()
```

```
color
Black    26.5
Brown    24.0
Gray     17.0
Tan      2.0
White    74.0
Name: weight_kg, dtype: float64
```

Multiple grouped summaries

```
dogs.groupby("color")["weight_kg"].agg([min, max, sum])
```

	min	max	sum
color			
Black	24	29	53
Brown	24	24	48
Gray	17	17	17
Tan	2	2	2
White	74	74	74

Grouping by multiple variables

```
dogs.groupby(["color", "breed"])["weight_kg"].mean()
```

```
color   breed
Black   Chow Chow      25
          Labrador     29
          Poodle        24
Brown   Chow Chow      24
          Labrador     24
Gray    Schnauzer     17
Tan     Chihuahua     2
White   St. Bernard    74
Name: weight_kg, dtype: int64
```

Many groups, many summaries

```
dogs.groupby(["color", "breed"])[["weight_kg", "height_cm"]].mean()
```

color	breed	weight_kg	height_cm
Black	Labrador	29	59
	Poodle	24	43
Brown	Chow Chow	24	46
	Labrador	24	56
Gray	Schnauzer	17	49
Tan	Chihuahua	2	18
White	St. Bernard	74	77

Let's practice!

DATA MANIPULATION WITH PANDAS

Pivot tables

DATA MANIPULATION WITH PANDAS



Maggie Matsui

Senior Content Developer at DataCamp

Group by to pivot table

```
dogs.groupby("color")["weight_kg"].mean()
```

```
color
Black    26
Brown    24
Gray     17
Tan      2
White    74
Name: weight_kg, dtype: int64
```

```
dogs.pivot_table(values="weight_kg",
                  index="color")
```

```
weight_kg
color
Black      26.5
Brown      24.0
Gray       17.0
Tan        2.0
White      74.0
```

Different statistics

```
import numpy as np  
dogs.pivot_table(values="weight_kg", index="color", aggfunc=np.median)
```

	weight_kg
color	
Black	26.5
Brown	24.0
Gray	17.0
Tan	2.0
White	74.0

Multiple statistics

```
dogs.pivot_table(values="weight_kg", index="color", aggfunc=[np.mean, np.median])
```

color	mean	median
	weight_kg	weight_kg
Black	26.5	26.5
Brown	24.0	24.0
Gray	17.0	17.0
Tan	2.0	2.0
White	74.0	74.0

Pivot on two variables

```
dogs.groupby(["color", "breed"])["weight_kg"].mean()
```

```
dogs.pivot_table(values="weight_kg", index="color", columns="breed")
```

breed	Chihuahua	Chow Chow	Labrador	Poodle	Schnauzer	St. Bernard
color						
Black	NaN	NaN	29.0	24.0	NaN	NaN
Brown	NaN	24.0	24.0	NaN	NaN	NaN
Gray	NaN	NaN	NaN	NaN	17.0	NaN
Tan	2.0	NaN	NaN	NaN	NaN	NaN
White	NaN	NaN	NaN	NaN	NaN	74.0

Filling missing values in pivot tables

```
dogs.pivot_table(values="weight_kg", index="color", columns="breed", fill_value=0)
```

breed	Chihuahua	Chow Chow	Labrador	Poodle	Schnauzer	St. Bernard
color						
Black	0	0	29	24	0	0
Brown	0	24	24	0	0	0
Gray	0	0	0	0	17	0
Tan	2	0	0	0	0	0
White	0	0	0	0	0	74

Summing with pivot tables

```
dogs.pivot_table(values="weight_kg", index="color", columns="breed",
                  fill_value=0, margins=True)
```

breed	Chihuahua	Chow Chow	Labrador	Poodle	Schnauzer	St. Bernard	All
color							
Black	0	0	29	24	0	0	26.500000
Brown	0	24	24	0	0	0	24.000000
Gray	0	0	0	0	17	0	17.000000
Tan	2	0	0	0	0	0	2.000000
White	0	0	0	0	0	74	74.000000
All	2	24	26	24	17	74	27.714286

Let's practice!

DATA MANIPULATION WITH PANDAS

Explicit indexes

DATA MANIPULATION WITH PANDAS



Richie Cotton

Curriculum Architect at DataCamp

The dog dataset, revisited

```
print(dogs)
```

	name	breed	color	height_cm	weight_kg
0	Bella	Labrador	Brown	56	25
1	Charlie	Poodle	Black	43	23
2	Lucy	Chow Chow	Brown	46	22
3	Cooper	Schnauzer	Gray	49	17
4	Max	Labrador	Black	59	29
5	Stella	Chihuahua	Tan	18	2
6	Bernie	St. Bernard	White	77	74

.columns and .index

```
dogs.columns
```

```
Index(['name', 'breed', 'color', 'height_cm', 'weight_kg'], dtype='object')
```

```
dogs.index
```

```
RangeIndex(start=0, stop=7, step=1)
```

Setting a column as the index

```
dogs_ind = dogs.set_index("name")  
print(dogs_ind)
```

	breed	color	height_cm	weight_kg
name				
Bella	Labrador	Brown	56	25
Charlie	Poodle	Black	43	23
Lucy	Chow Chow	Brown	46	22
Cooper	Schnauzer	Grey	49	17
Max	Labrador	Black	59	29
Stella	Chihuahua	Tan	18	2
Bernie	St. Bernard	White	77	74

Removing an index

```
dogs_ind.reset_index()
```

	name	breed	color	height_cm	weight_kg
0	Bella	Labrador	Brown	56	25
1	Charlie	Poodle	Black	43	23
2	Lucy	Chow Chow	Brown	46	22
3	Cooper	Schnauzer	Grey	49	17
4	Max	Labrador	Black	59	29
5	Stella	Chihuahua	Tan	18	2
6	Bernie	St. Bernard	White	77	74

Dropping an index

```
dogs_ind.reset_index(drop=True)
```

	breed	color	height_cm	weight_kg
0	Labrador	Brown	56	25
1	Poodle	Black	43	23
2	Chow Chow	Brown	46	22
3	Schnauzer	Grey	49	17
4	Labrador	Black	59	29
5	Chihuahua	Tan	18	2
6	St. Bernard	White	77	74

Indexes make subsetting simpler

```
dogs[dogs["name"].isin(["Bella", "Stella"])]
```

```
    name      breed  color  height_cm  weight_kg
0  Bella    Labrador  Brown        56        25
5  Stella   Chihuahua  Tan         18         2
```

```
dogs_ind.loc[['Bella', 'Stella']]
```

```
      breed  color  height_cm  weight_kg
name
Bella    Labrador  Brown        56        25
Stella   Chihuahua  Tan         18         2
```

Index values don't need to be unique

```
dogs_ind2 = dogs.set_index("breed")
print(dogs_ind2)
```

		name	color	height_cm	weight_kg
breed					
Labrador	Bella	Brown		56	25
Poodle	Charlie	Black		43	23
Chow Chow	Lucy	Brown		46	22
Schnauzer	Cooper	Grey		49	17
Labrador	Max	Black		59	29
Chihuahua	Stella	Tan		18	2
St. Bernard	Bernie	White		77	74

Subsetting on duplicated index values

```
dogs_ind2.loc["Labrador"]
```

```
      name  color  height_cm  weight_kg  
breed  
Labrador  Bella  Brown        56        25  
Labrador    Max  Black        59        29
```

Multi-level indexes a.k.a. hierarchical indexes

```
dogs_ind3 = dogs.set_index(["breed", "color"])
print(dogs_ind3)
```

			name	height_cm	weight_kg
breed	color				
Labrador	Brown	Bella		56	25
Poodle	Black	Charlie		43	23
Chow Chow	Brown	Lucy		46	22
Schnauzer	Grey	Cooper		49	17
Labrador	Black	Max		59	29
Chihuahua	Tan	Stella		18	2
St. Bernard	White	Bernie		77	74

Subset the outer level with a list

```
dogs_ind3.loc[["Labrador", "Chihuahua"]]
```

			name	height_cm	weight_kg
breed	color				
Labrador	Brown	Bella		56	25
	Black	Max		59	29
Chihuahua	Tan	Stella		18	2

Subset inner levels with a list of tuples

```
dogs_ind3.loc[[("Labrador", "Brown"), ("Chihuahua", "Tan")]]
```

			name	height_cm	weight_kg
breed	color				
Labrador	Brown	Bella		56	25
Chihuahua	Tan	Stella		18	2

Sorting by index values

```
dogs_ind3.sort_index()
```

			name	height_cm	weight_kg
breed	color				
Chihuahua	Tan	Stella		18	2
Chow Chow	Brown	Lucy		46	22
Labrador	Black	Max		59	29
	Brown	Bella		56	25
Poodle	Black	Charlie		43	23
Schnauzer	Grey	Cooper		49	17
St. Bernard	White	Bernie		77	74

Controlling sort_index

```
dogs_ind3.sort_index(level=["color", "breed"], ascending=[True, False])
```

			name	height_cm	weight_kg
breed	color				
Poodle	Black	Charlie		43	23
Labrador	Black	Max		59	29
	Brown	Bella		56	25
Chow Chow	Brown	Lucy		46	22
Schanuzer	Grey	Cooper		49	17
Chihuahua	Tan	Stella		18	2
St. Bernard	White	Bernie		77	74

Now you have two problems

- Index values are just data
- Indexes violate "tidy data" principles
- You need to learn two syntaxes

Temperature dataset

	date	city	country	avg_temp_c
0	2000-01-01	Abidjan	Côte D'Ivoire	27.293
1	2000-02-01	Abidjan	Côte D'Ivoire	27.685
2	2000-03-01	Abidjan	Côte D'Ivoire	29.061
3	2000-04-01	Abidjan	Côte D'Ivoire	28.162
4	2000-05-01	Abidjan	Côte D'Ivoire	27.547

Let's practice!

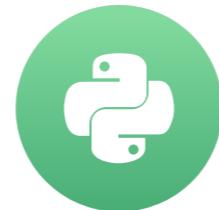
DATA MANIPULATION WITH PANDAS

Slicing and subsetting with .loc and .iloc

DATA MANIPULATION WITH PANDAS

Richie Cotton

Curriculum Architect at DataCamp



Slicing lists

```
breeds = ["Labrador", "Poodle",  
          "Chow Chow", "Schnauzer",  
          "Labrador", "Chihuahua",  
          "St. Bernard"]
```

```
['Labrador',  
 'Poodle',  
 'Chow Chow',  
 'Schnauzer',  
 'Labrador',  
 'Chihuahua',  
 'St. Bernard']
```

```
breeds[2:5]
```

```
['Chow Chow', 'Schnauzer', 'Labrador']
```

```
breeds[:3]
```

```
['Labrador', 'Poodle', 'Chow Chow']
```

```
breeds[:]
```

```
['Labrador', 'Poodle', 'Chow Chow', 'Schnauzer',  
 'Labrador', 'Chihuahua', 'St. Bernard']
```

Sort the index before you slice

```
dogs_srt = dogs.set_index(["breed", "color"]).sort_index()  
print(dogs_srt)
```

			name	height_cm	weight_kg
breed	color				
Chihuahua	Tan	Stella		18	2
Chow Chow	Brown	Lucy		46	22
Labrador	Black	Max		59	29
	Brown	Bella		56	25
Poodle	Black	Charlie		43	23
Schnauzer	Grey	Cooper		49	17
St. Bernard	White	Bernie		77	74

Slicing the outer index level

```
dogs_srt.loc["Chow Chow":"Poodle"]
```

breed	color		name	height_cm	weight_kg
Chow	Chow	Brown	Lucy	46	22
Labrador	Black		Max	59	29
		Brown	Bella	56	25
Poodle	Black	Charlie		43	23

The final value "Poodle" is included

Full dataset

breed	color		name	height_cm	weight_kg
Chihuahua	Tan		Stella	18	2
Chow	Chow	Brown	Lucy	46	22
Labrador	Black		Max	59	29
		Brown	Bella	56	25
Poodle	Black	Charlie		43	23
Schnauzer	Grey		Cooper	49	17
St. Bernard	White	Bernie		77	74

Slicing the inner index levels badly

```
dogs_srt.loc["Tan":"Grey"]
```

Empty DataFrame

Columns: [name, height_cm, weight_kg]

Index: []

Full dataset

breed	color	name	height_cm	weight_kg
Chihuahua	Tan	Stella	18	2
Chow Chow	Brown	Lucy	46	22
Labrador	Black	Max	59	29
	Brown	Bella	56	25
Poodle	Black	Charlie	43	23
Schnauzer	Grey	Cooper	49	17
St. Bernard	White	Bernie	77	74

Slicing the inner index levels correctly

```
dogs_srt.loc[  
    ("Labrador", "Brown") : ("Schnauzer", "Grey")]
```

breed	color		name	height_cm	weight_kg
Labrador	Brown	Bella		56	25
Poodle	Black	Charlie		43	23
Schnauzer	Grey	Cooper		49	17

Full dataset

breed	color		name	height_cm	weight_kg
Chihuahua	Tan		Stella	18	2
Chow Chow	Brown		Lucy	46	22
Labrador	Black		Max	59	29
	Brown		Bella	56	25
Poodle	Black	Charlie		43	23
Schnauzer	Grey	Cooper		49	17
St. Bernard	White	Bernie		77	74

Slicing columns

```
dogs_srt.loc[:, "name":"height_cm"]
```

breed	color		name	height_cm
Chihuahua	Tan	Stella		18
Chow Chow	Brown	Lucy		46
Labrador	Black	Max		59
	Brown	Bella		56
Poodle	Black	Charlie		43
Schnauzer	Grey	Cooper		49
St. Bernard	White	Bernie		77

Full dataset

breed	color		name	height_cm	weight_kg
Chihuahua	Tan	Stella		18	2
Chow Chow	Brown	Lucy		46	22
Labrador	Black	Max		59	29
	Brown	Bella		56	25
Poodle	Black	Charlie		43	23
Schnauzer	Grey	Cooper		49	17
St. Bernard	White	Bernie		77	74

Slice twice

```
dogs_srt.loc[  
    ("Labrador", "Brown") : ("Schnauzer", "Grey"),  
    "name": "height_cm"]
```

			name	height_cm
breed	color			
Labrador	Brown	Bella		56
Poodle	Black	Charlie		43
Schanuzer	Grey	Cooper		49

Full dataset

breed	color	name	height_cm	weight_kg
Chihuahua	Tan	Stella	18	2
Chow Chow	Brown	Lucy	46	22
Labrador	Black	Max	59	29
	Brown	Bella	56	25
Poodle	Black	Charlie	43	23
Schnauzer	Grey	Cooper	49	17
St. Bernard	White	Bernie	77	74

Dog days

```
dogs = dogs.set_index("date_of_birth").sort_index()  
print(dogs)
```

	name	breed	color	height_cm	weight_kg
date_of_birth					
2011-12-11	Cooper	Schanuzer	Grey	49	17
2013-07-01	Bella	Labrador	Brown	56	25
2014-08-25	Lucy	Chow Chow	Brown	46	22
2015-04-20	Stella	Chihuahua	Tan	18	2
2016-09-16	Charlie	Poodle	Black	43	23
2017-01-20	Max	Labrador	Black	59	29
2018-02-27	Bernie	St. Bernard	White	77	74

Slicing by dates

```
# Get dogs with date_of_birth between 2014-08-25 and 2016-09-16  
dogs.loc["2014-08-25":"2016-09-16"]
```

	name	breed	color	height_cm	weight_kg
date_of_birth					
2014-08-25	Lucy	Chow Chow	Brown	46	22
2015-04-20	Stella	Chihuahua	Tan	18	2
2016-09-16	Charlie	Poodle	Black	43	23

Slicing by partial dates

```
# Get dogs with date_of_birth between 2014-01-01 and 2016-12-31  
dogs.loc["2014":"2016"]
```

	name	breed	color	height_cm	weight_kg
date_of_birth					
2014-08-25	Lucy	Chow Chow	Brown	46	22
2015-04-20	Stella	Chihuahua	Tan	18	2
2016-09-16	Charlie	Poodle	Black	43	23

Subsetting by row/column number

```
print(dogs.iloc[2:5, 1:4])
```

```
breed  color  height_cm  
2  Chow  Chow  Brown      46  
3  Schnauzer  Grey      49  
4  Labrador  Black      59
```

Full dataset

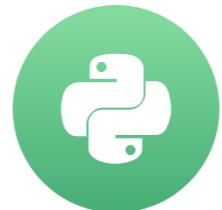
```
name  breed  color  height_cm  weight_kg  
0  Bella  Labrador  Brown      56      25  
1  Charlie  Poodle  Black      43      23  
2  Lucy  Chow Chow  Brown      46      22  
3  Cooper  Schnauzer  Grey      49      17  
4  Max  Labrador  Black      59      29  
5  Stella  Chihuahua  Tan      18      2  
6  Bernie  St. Bernard  White    77      74
```

Let's practice!

DATA MANIPULATION WITH PANDAS

Working with pivot tables

DATA MANIPULATION WITH PANDAS



Richie Cotton

Curriculum Architect at DataCamp

A bigger dog dataset

```
print(dog_pack)
```

```
breed  color  height_cm  weight_kg
0      Boxer  Brown       62.64     30.4
1      Poodle  Black       46.41     20.4
2      Beagle  Brown       36.39     12.4
3  Chihuahua  Tan        19.70      1.6
4    Labrador  Tan        54.44     36.1
...
87     Boxer  Gray        58.13     29.9
88  St. Bernard  White      70.13     69.4
89     Poodle  Gray        51.30     20.4
90     Beagle  White       38.81      8.8
91     Beagle  Black       33.40     13.5
```

Pivoting the dog pack

```
dogs_height_by_breed_vs_color = dog_pack.pivot_table(  
    "height_cm", index="breed", columns="color")  
print(dogs_height_by_breed_vs_color)
```

color	Black	Brown	Gray	Tan	White
breed					
Beagle	34.500000	36.4500	36.313333	35.740000	38.810000
Boxer	57.203333	62.6400	58.280000	62.310000	56.360000
Chihuahua	18.555000	NaN	21.660000	20.096667	17.933333
Chow Chow	51.262500	50.4800	NaN	53.497500	54.413333
Dachshund	21.186667	19.7250	NaN	19.375000	20.660000
Labrador	57.125000	NaN	NaN	55.190000	55.310000
Poodle	48.036000	57.1300	56.645000	NaN	44.740000
St. Bernard	63.920000	65.8825	67.640000	68.334000	67.495000

.loc[] + slicing is a power combo

```
dogs_height_by_breed_vs_color.loc["Chow Chow":"Poodle"]
```

color	Black	Brown	Gray	Tan	White
breed					
Chow Chow	51.262500	50.480	NaN	53.4975	54.413333
Dachshund	21.186667	19.725	NaN	19.3750	20.660000
Labrador	57.125000	NaN	NaN	55.1900	55.310000
Poodle	48.036000	57.130	56.645	NaN	44.740000

The axis argument

```
dogs_height_by_breed_vs_color.mean(axis="index")
```

```
color
Black      43.973563
Brown      48.717917
Gray       48.107667
Tan        44.934738
White      44.465208
dtype: float64
```

Calculating summary stats across columns

```
dogs_height_by_breed_vs_color.mean(axis="columns")
```

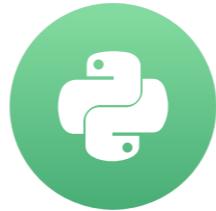
```
breed
Beagle      36.362667
Boxer       59.358667
Chihuahua   19.561250
Chow Chow    52.413333
Dachshund   20.236667
Labrador     55.875000
Poodle       51.637750
St. Bernard  66.654300
dtype: float64
```

Let's practice!

DATA MANIPULATION WITH PANDAS

Visualizing your data

DATA MANIPULATION WITH PANDAS



Maggie Matsui

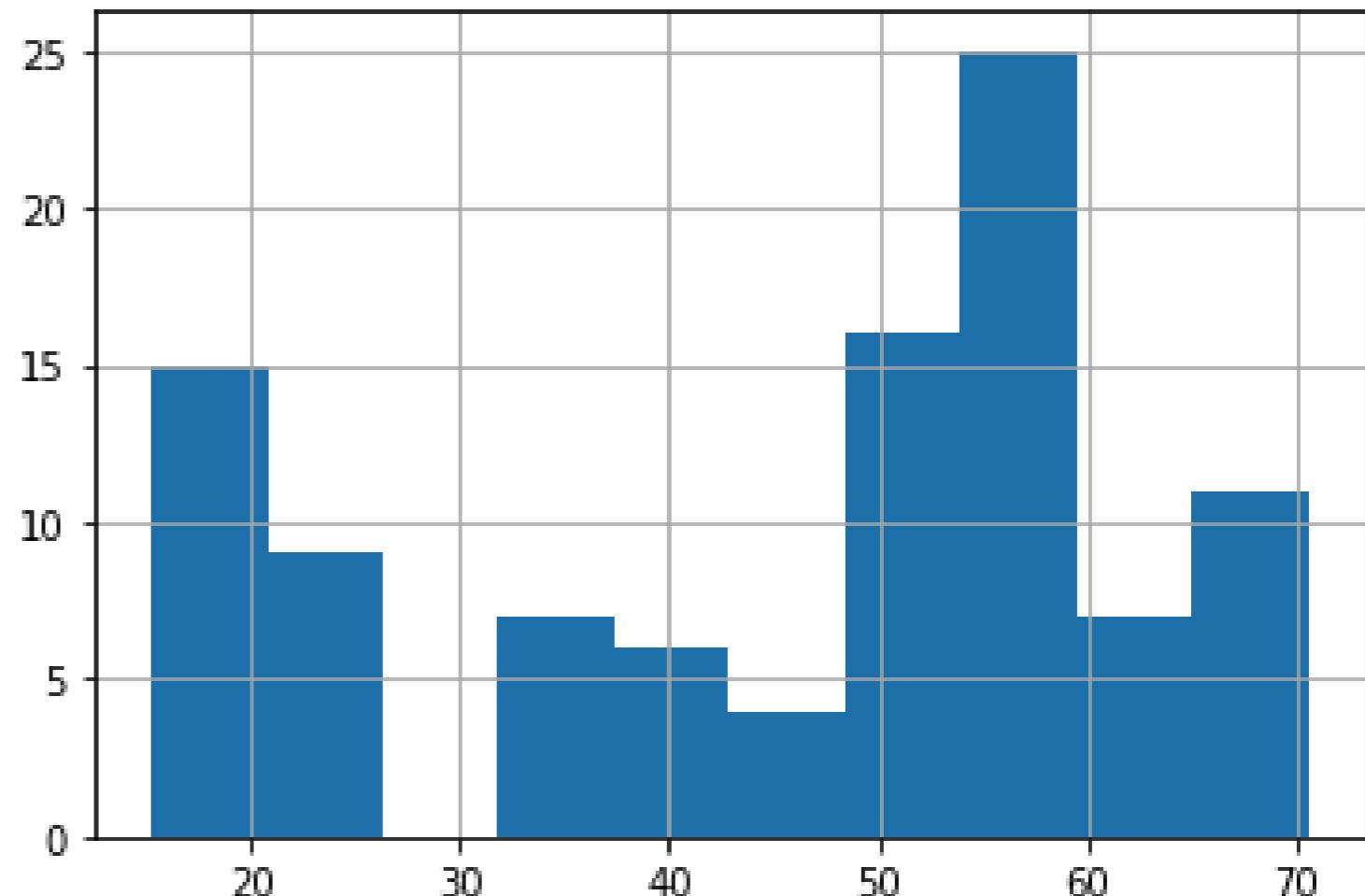
Senior Content Developer at DataCamp

Histograms

```
import matplotlib.pyplot as plt
```

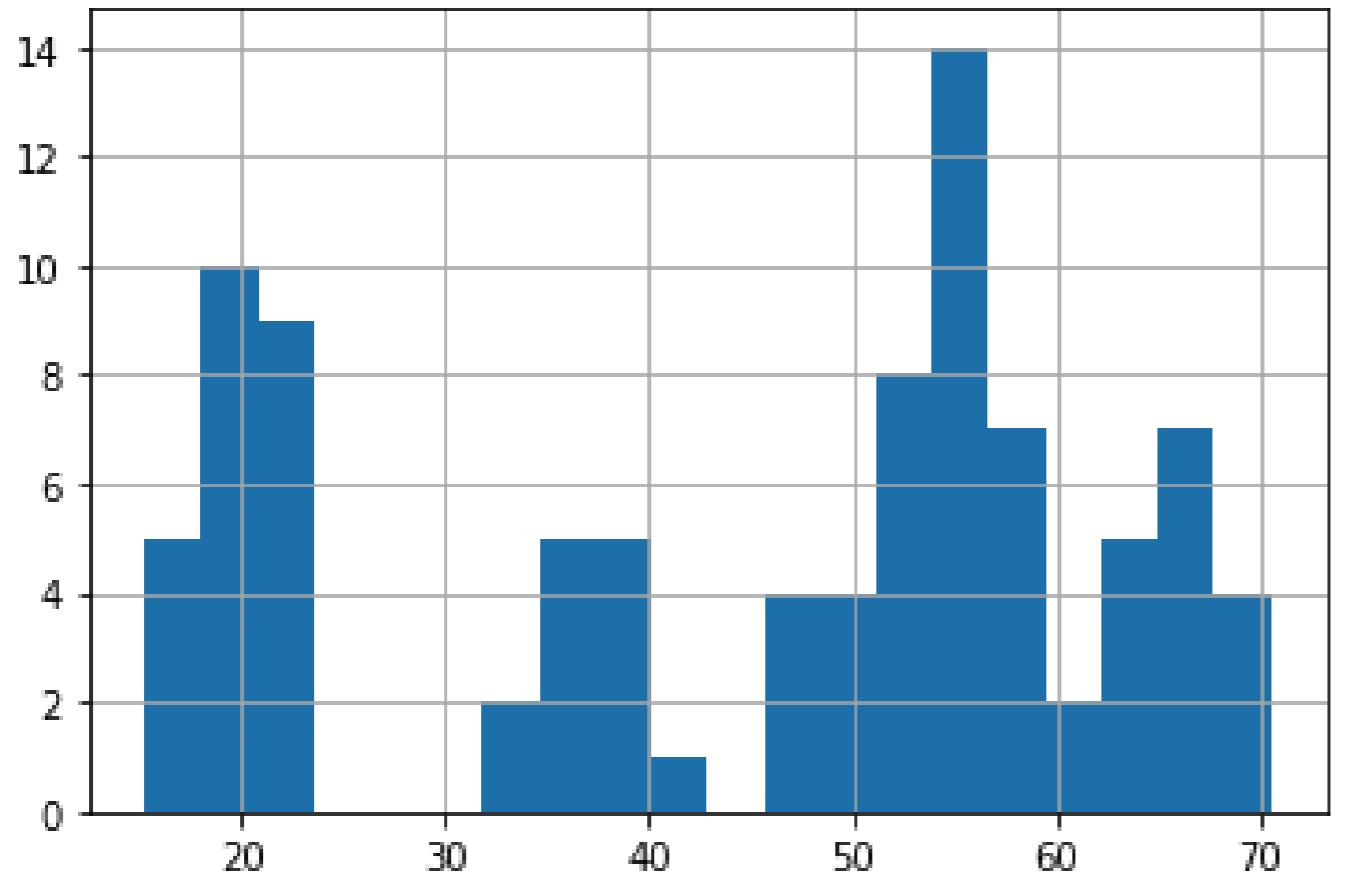
```
dog_pack[ "height_cm" ].hist()
```

```
plt.show()
```

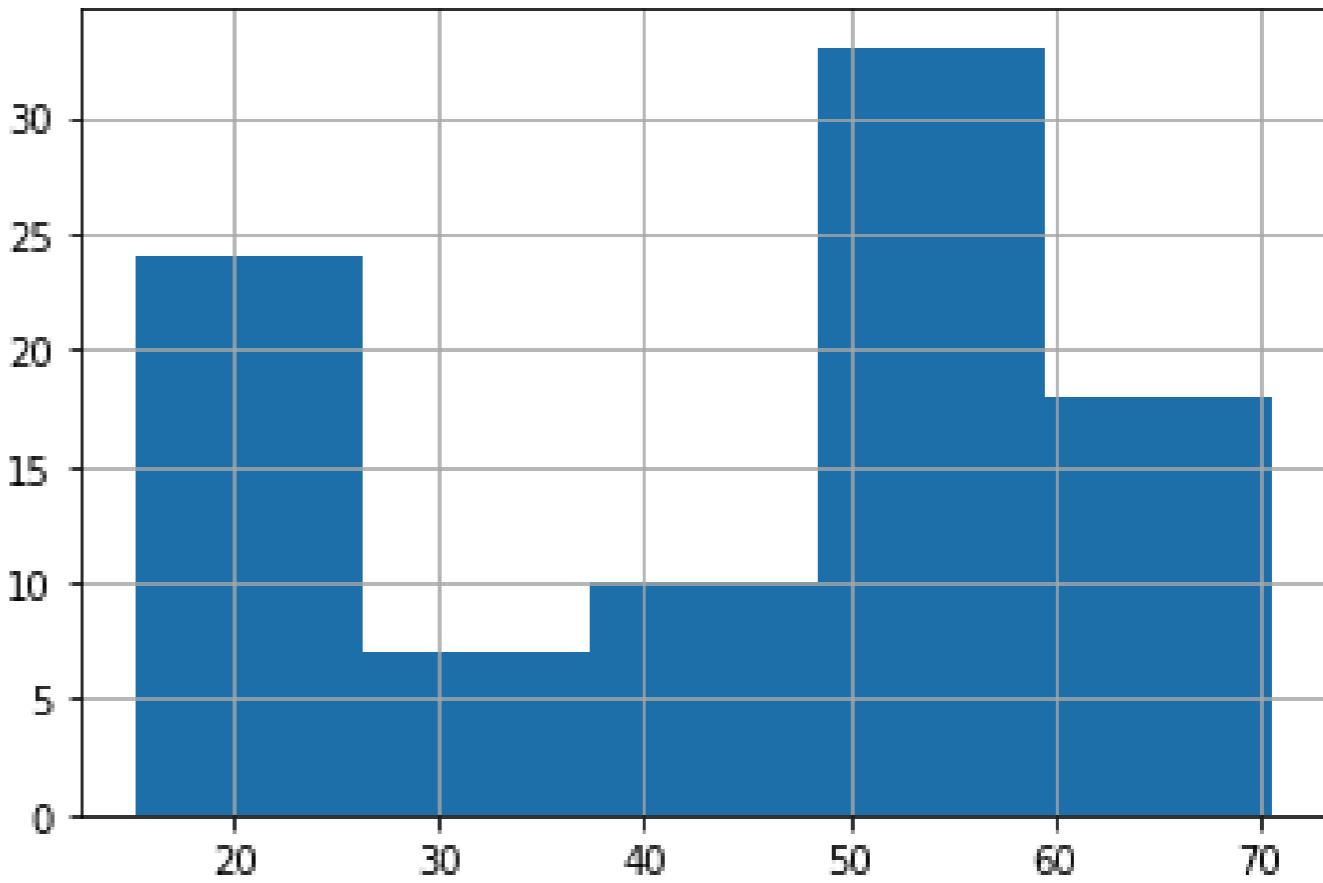


Histograms

```
dog_pack[ "height_cm" ].hist(bins=20)  
plt.show()
```



```
dog_pack[ "height_cm" ].hist(bins=5)  
plt.show()
```



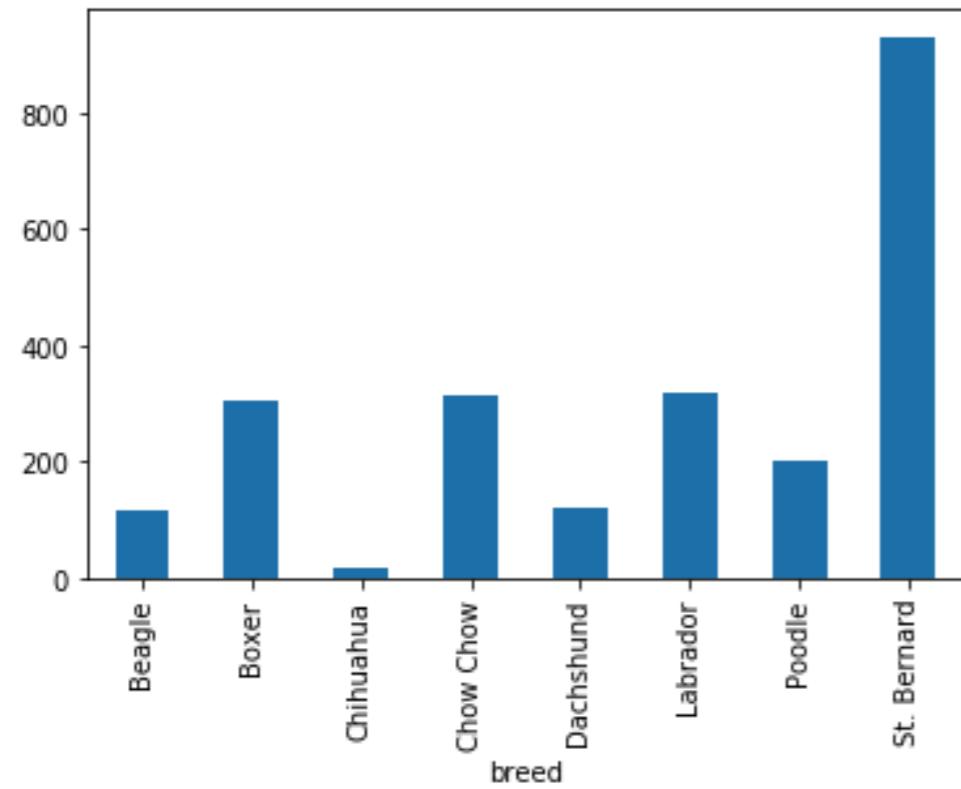
Bar plots

```
avg_weight_by_breed = dog_pack.groupby("breed")["weight_kg"].mean()  
print(avg_weight_by_breed)
```

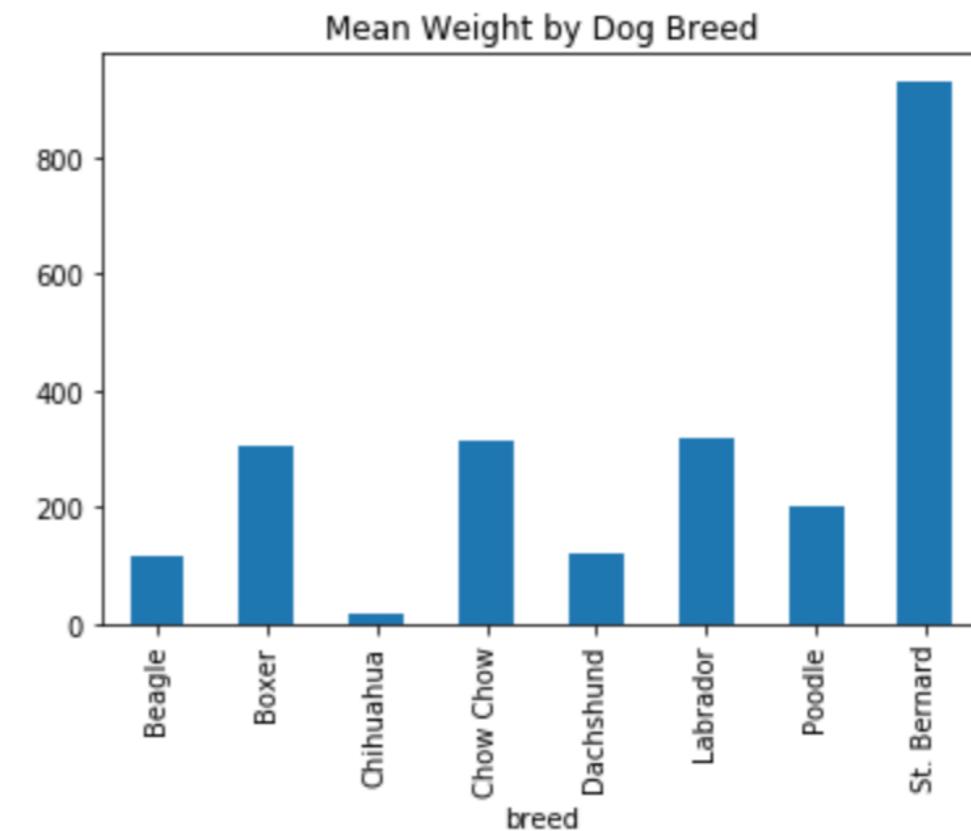
```
breed  
Beagle      10.636364  
Boxer       30.620000  
Chihuahua   1.491667  
Chow Chow    22.535714  
Dachshund   9.975000  
Labrador     31.850000  
Poodle       20.400000  
St. Bernard  71.576923  
Name: weight_kg, dtype: float64
```

Bar plots

```
avg_weight_by_breed.plot(kind="bar")  
plt.show()
```



```
avg_weight_by_breed.plot(kind="bar",  
                         title="Mean Weight by Dog Breed")  
plt.show()
```

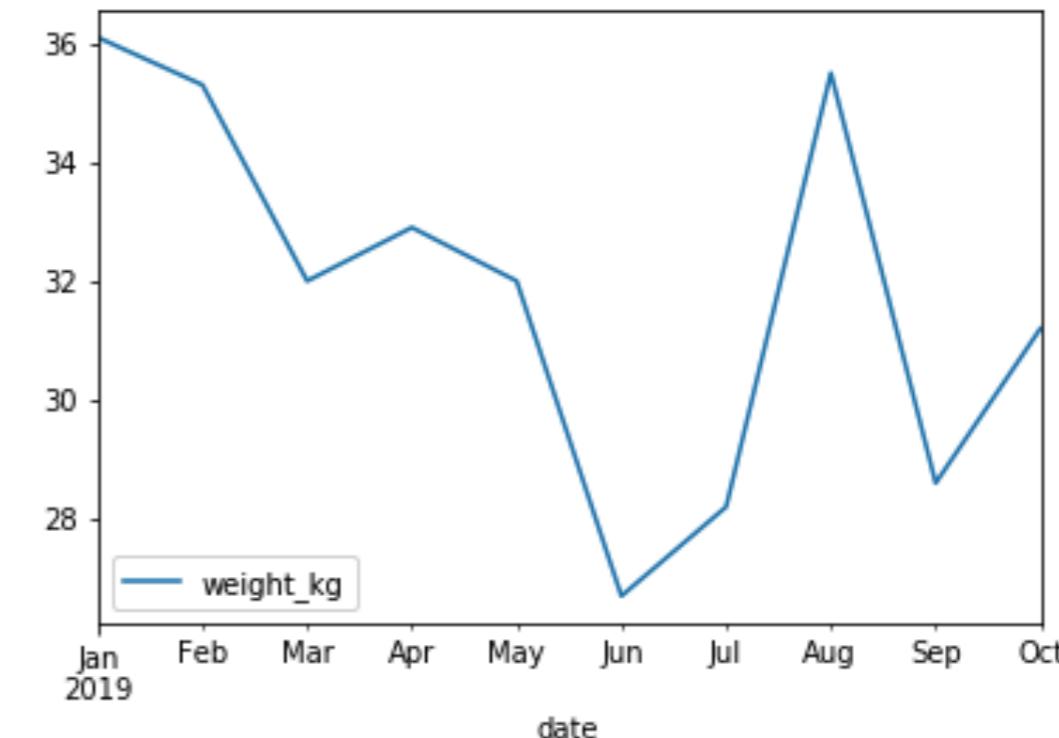


Line plots

```
sully.head()
```

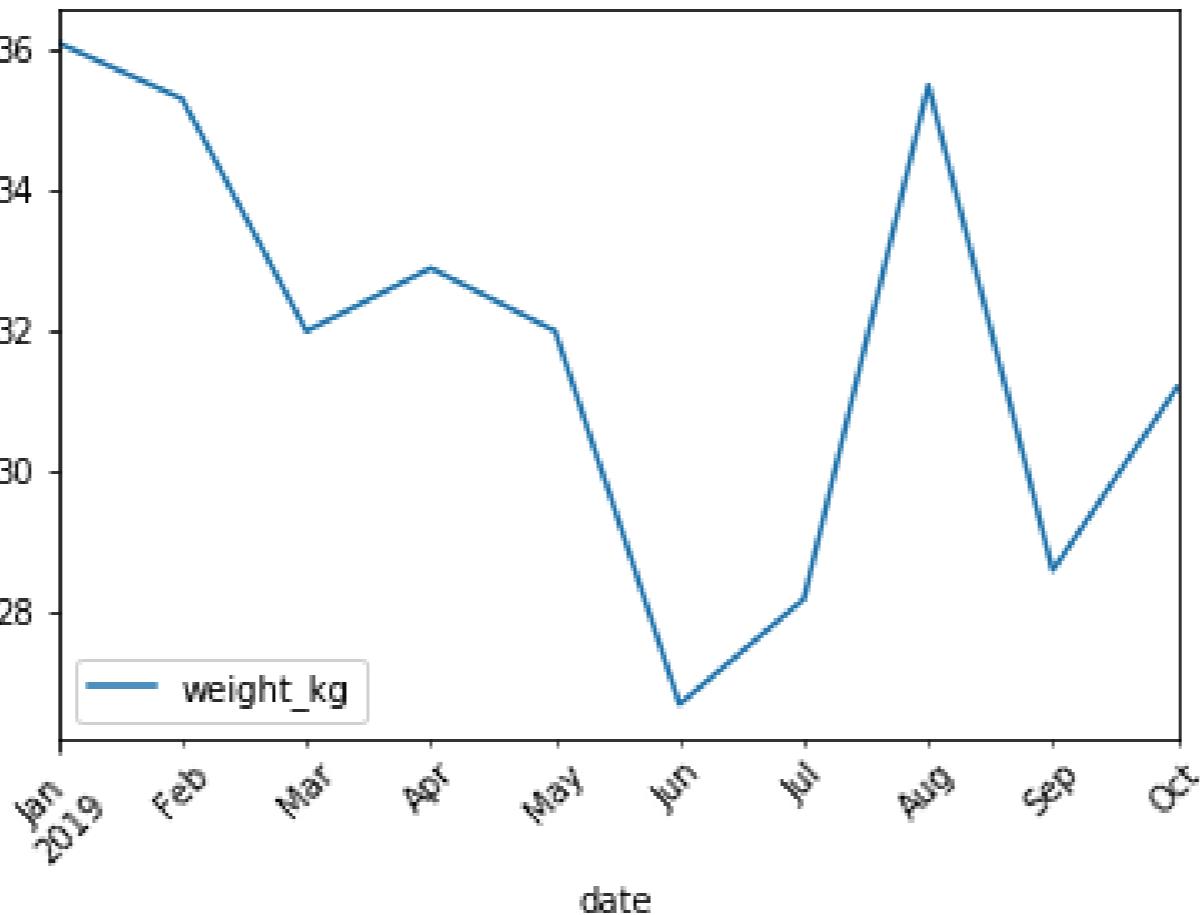
```
      date    weight_kg
0 2019-01-31      36.1
1 2019-02-28      35.3
2 2019-03-31      32.0
3 2019-04-30      32.9
4 2019-05-31      32.0
```

```
sully.plot(x="date",
            y="weight_kg",
            kind="line")
plt.show()
```



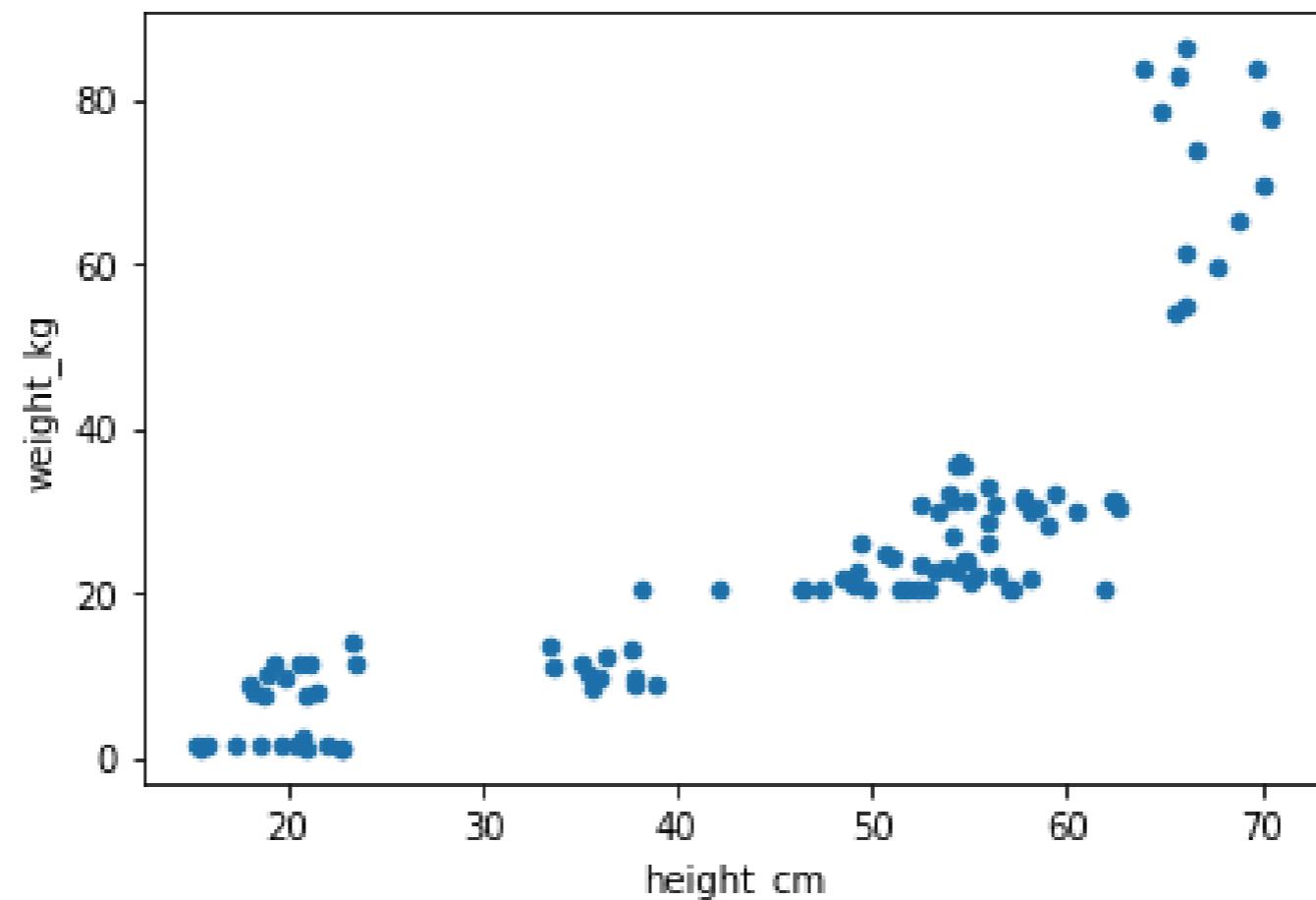
Rotating axis labels

```
sully.plot(x="date", y="weight_kg", kind="line", rot=45)  
plt.show()
```



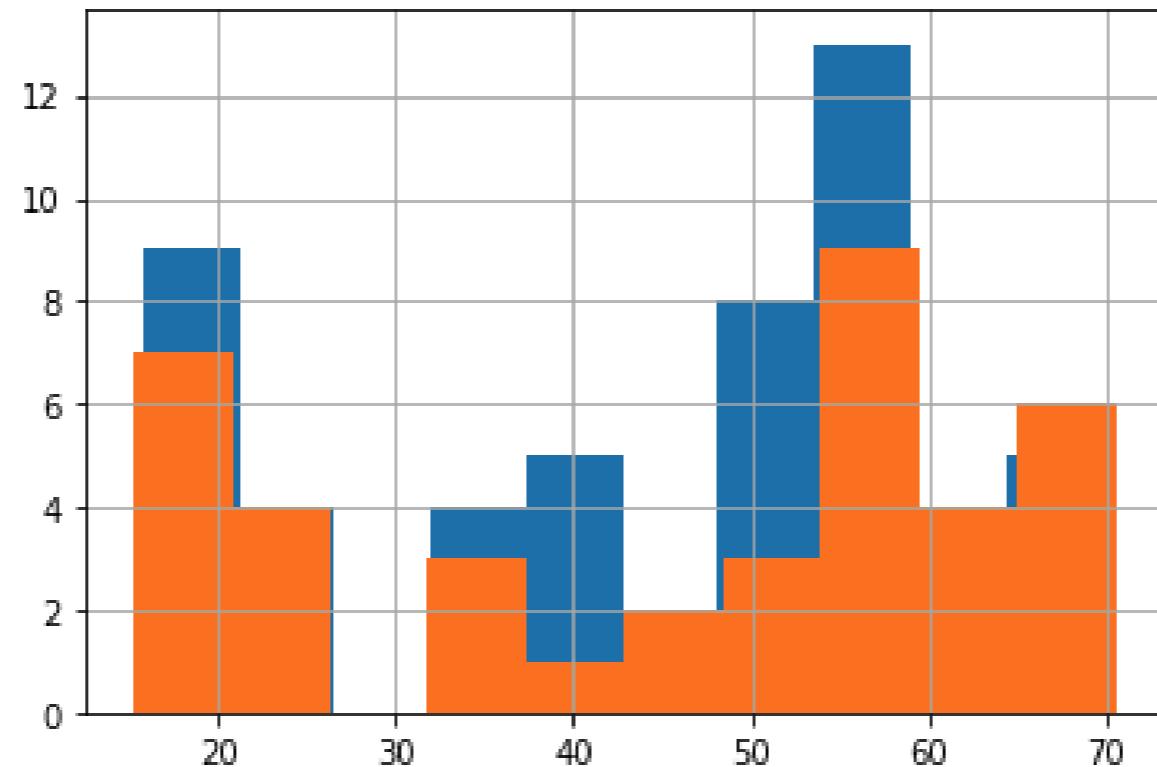
Scatter plots

```
dog_pack.plot(x="height_cm", y="weight_kg", kind="scatter")  
plt.show()
```



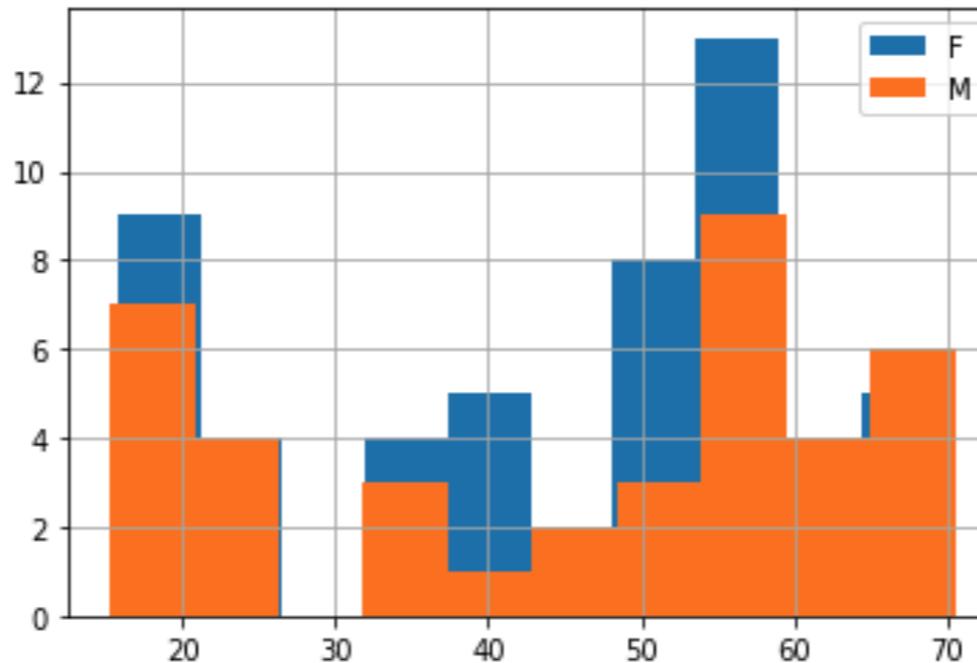
Layering plots

```
dog_pack[dog_pack["sex"]=="F"]["height_cm"].hist()  
dog_pack[dog_pack["sex"]=="M"]["height_cm"].hist()  
plt.show()
```



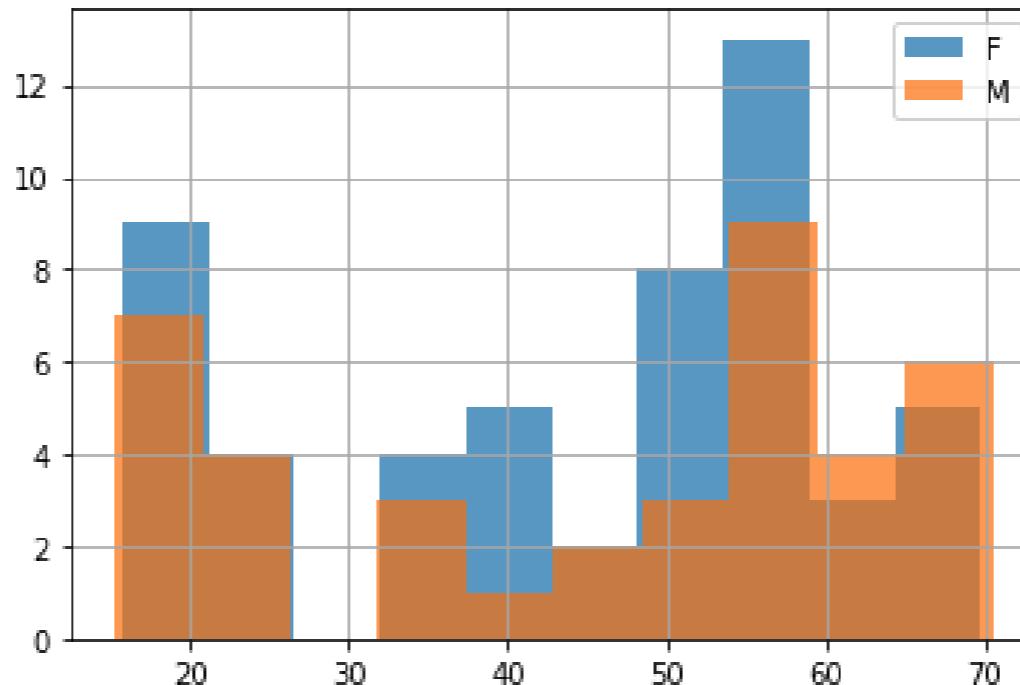
Add a legend

```
dog_pack[dog_pack["sex"]=="F"]["height_cm"].hist()  
dog_pack[dog_pack["sex"]=="M"]["height_cm"].hist()  
plt.legend(["F", "M"])  
plt.show()
```



Transparency

```
dog_pack[dog_pack[ "sex" ]=="F"]["height_cm"].hist(alpha=0.7)  
dog_pack[dog_pack[ "sex" ]=="M"]["height_cm"].hist(alpha=0.7)  
plt.legend( [ "F" , "M" ] )  
plt.show()
```



Avocados

```
print(avocados)
```

	date	type	year	avg_price	size	nb_sold
0	2015-12-27	conventional	2015	0.95	small	9626901.09
1	2015-12-20	conventional	2015	0.98	small	8710021.76
2	2015-12-13	conventional	2015	0.93	small	9855053.66
...
1011	2018-01-21	organic	2018	1.63	extra_large	1490.02
1012	2018-01-14	organic	2018	1.59	extra_large	1580.01
1013	2018-01-07	organic	2018	1.51	extra_large	1289.07

[1014 rows x 6 columns]

Let's practice!

DATA MANIPULATION WITH PANDAS

Missing values

DATA MANIPULATION WITH PANDAS



Maggie Matsui

Senior Content Developer at DataCamp

What's a missing value?

Name	Breed	Color	Height (cm)	Weight (kg)	Date of Birth
Bella	Labrador	Brown	56	25	2013-07-01
Charlie	Poodle	Black	43	23	2016-09-16
Lucy	Chow Chow	Brown	46	22	2014-08-25
Cooper	Schnauzer	Gray	49	17	2011-12-11
Max	Labrador	Black	59	29	2017-01-20
Stella	Chihuahua	Tan	18	2	2015-04-20
Bernie	St. Bernard	White	77	74	2018-02-27

What's a missing value?

Name	Breed	Color	Height (cm)	Weight (kg)	Date of Birth
Bella	Labrador	Brown	56	?	2013-07-01
Charlie	Poodle	Black	43	23	2016-09-16
Lucy	Chow Chow	Brown	46	22	2014-08-25
Cooper	Schnauzer	Gray	49	?	2011-12-11
Max	Labrador	Black	59	29	2017-01-20
Stella	Chihuahua	Tan	18	2	2015-04-20
Bernie	St. Bernard	White	77	74	2018-02-27

Missing values in pandas DataFrames

```
print(dogs)
```

	name	breed	color	height_cm	weight_kg	date_of_birth
0	Bella	Labrador	Brown	56	NaN	2013-07-01
1	Charlie	Poodle	Black	43	24.0	2016-09-16
2	Lucy	Chow Chow	Brown	46	24.0	2014-08-25
3	Cooper	Schnauzer	Gray	49	NaN	2011-12-11
4	Max	Labrador	Black	59	29.0	2017-01-20
5	Stella	Chihuahua	Tan	18	2.0	2015-04-20
6	Bernie	St. Bernard	White	77	74.0	2018-02-27

Detecting missing values

```
dogs.isna()
```

```
    name  breed  color  height_cm  weight_kg  date_of_birth
0  False  False  False      False       True        False
1  False  False  False      False      False        False
2  False  False  False      False      False        False
3  False  False  False      False       True        False
4  False  False  False      False      False        False
5  False  False  False      False      False        False
6  False  False  False      False      False        False
```

Detecting any missing values

```
dogs.isna().any()
```

```
name      False
breed     False
color     False
height_cm False
weight_kg  True
date_of_birth False
dtype: bool
```

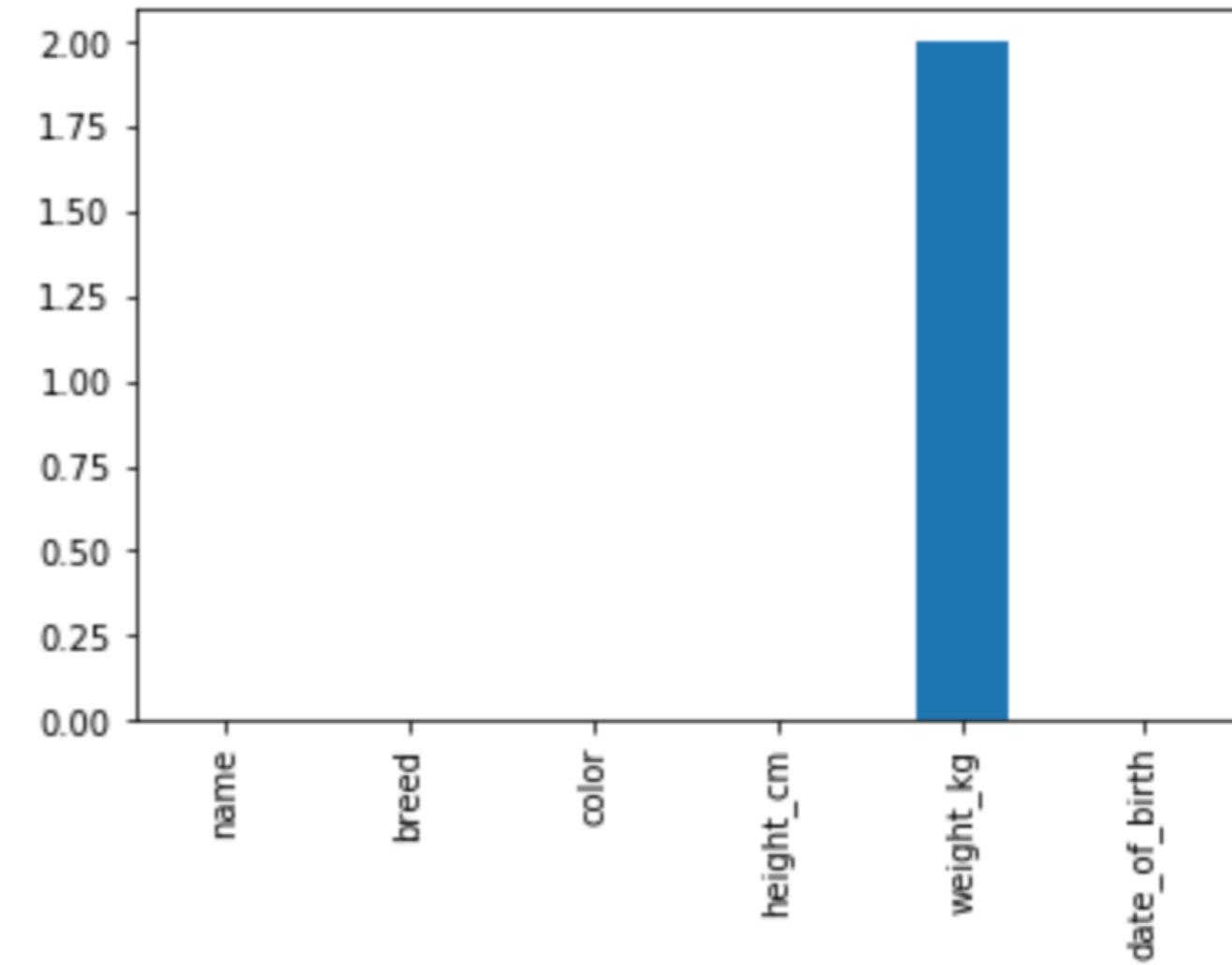
Counting missing values

```
dogs.isna().sum()
```

```
name          0  
breed         0  
color         0  
height_cm     0  
weight_kg     2  
date_of_birth  0  
dtype: int64
```

Plotting missing values

```
import matplotlib.pyplot as plt  
  
dogs.isna().sum().plot(kind="bar")  
plt.show()
```



Removing missing values

```
dogs.dropna()
```

	name	breed	color	height_cm	weight_kg	date_of_birth
1	Charlie	Poodle	Black	43	24.0	2016-09-16
2	Lucy	Chow Chow	Brown	46	24.0	2014-08-25
4	Max	Labrador	Black	59	29.0	2017-01-20
5	Stella	Chihuahua	Tan	18	2.0	2015-04-20
6	Bernie	St. Bernard	White	77	74.0	2018-02-27

Replacing missing values

```
dogs.fillna(0)
```

	name	breed	color	height_cm	weight_kg	date_of_birth
0	Bella	Labrador	Brown	56	0.0	2013-07-01
1	Charlie	Poodle	Black	43	24.0	2016-09-16
2	Lucy	Chow Chow	Brown	46	24.0	2014-08-25
3	Cooper	Schnauzer	Gray	49	0.0	2011-12-11
4	Max	Labrador	Black	59	29.0	2017-01-20
5	Stella	Chihuahua	Tan	18	2.0	2015-04-20
6	Bernie	St. Bernard	White	77	74.0	2018-02-27

Let's practice!

DATA MANIPULATION WITH PANDAS

Creating DataFrames

DATA MANIPULATION WITH PANDAS



Maggie Matsui

Senior Content Developer at DataCamp

Dictionaries

```
my_dict = {  
    "key1": value1,  
    "key2": value2,  
    "key3": value3  
}
```

```
my_dict["key1"]
```

value1

```
my_dict = {  
    "title": "Charlotte's Web",  
    "author": "E.B. White",  
    "published": 1952  
}
```

```
my_dict["title"]
```

Charlotte's Web

Creating DataFrames

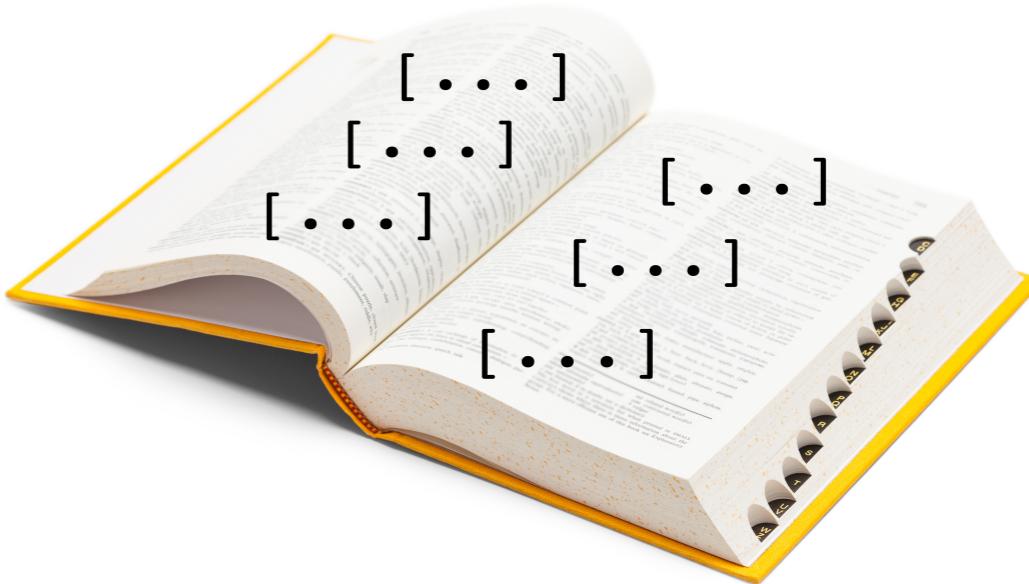
From a list of dictionaries

- Constructed row by row



From a dictionary of lists

- Constructed column by column



List of dictionaries - by row

name	breed	height (cm)	weight (kg)	date of birth
Ginger	Dachshund	22	10	2019-03-14
Scout	Dalmatian	59	25	2019-05-09

```
list_of_dicts = [  
    {"name": "Ginger", "breed": "Dachshund", "height_cm": 22,  
     "weight_kg": 10, "date_of_birth": "2019-03-14"},  
    {"name": "Scout", "breed": "Dalmatian", "height_cm": 59,  
     "weight_kg": 25, "date_of_birth": "2019-05-09"}  
]
```

List of dictionaries - by row

name	breed	height (cm)	weight (kg)	date of birth
Ginger	Dachshund	22	10	2019-03-14
Scout	Dalmatian	59	25	2019-05-09

```
new_dogs = pd.DataFrame(list_of_dicts)  
print(new_dogs)
```

```
      name      breed  height_cm  weight_kg  date_of_birth  
0  Ginger  Dachshund        22         10  2019-03-14  
1    Scout   Dalmatian        59         25  2019-05-09
```

Dictionary of lists - by column

name	breed	height (cm)	weight (kg)	date of birth
Ginger	Dachshund	22	10	2019-03-14
Scout	Dalmatian	59	25	2019-05-09

- **Key** = column name
- **Value** = list of column values

```
dict_of_lists = {  
    "name": [ "Ginger", "Scout" ],  
    "breed": [ "Dachshund", "Dalmatian" ],  
    "height_cm": [ 22, 59 ],  
    "weight_kg": [ 10, 25 ],  
    "date_of_birth": [ "2019-03-14",  
                      "2019-05-09" ]  
}
```

```
new_dogs = pd.DataFrame(dict_of_lists)
```

Dictionary of lists - by column

name	breed	height (cm)	weight (kg)	date of birth
Ginger	Dachshund	22	10	2019-03-14
Scout	Dalmatian	59	25	2019-05-09

```
print(new_dogs)
```

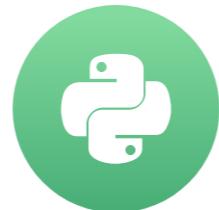
```
      name      breed  height_cm  weight_kg  date_of_birth
0  Ginger  Dachshund        22         10  2019-03-14
1    Scout   Dalmatian        59         25  2019-05-09
```

Let's practice!

DATA MANIPULATION WITH PANDAS

Reading and writing CSVs

DATA MANIPULATION WITH PANDAS



Maggie Matsui

Senior Content Developer at DataCamp

What's a CSV file?

- CSV = comma-separated values
- Designed for DataFrame-like data
- Most database and spreadsheet programs can use them or create them



Example CSV file

name	breed	height (cm)	weight (kg)	date of birth
Ginger	Dachshund	22	10	2019-03-14
Scout	Dalmatian	59	25	2019-05-09

new_dogs.csv

```
name,breed,height_cm,weight_kg,d_o_b  
Ginger,Dachshund,22,10,2019-03-14  
Scout,Dalmatian,59,25,2019-05-09
```

CSV to DataFrame

```
import pandas as pd  
  
new_dogs = pd.read_csv("new_dogs.csv")  
  
print(new_dogs)
```

```
      name      breed  height_cm  weight_kg  date_of_birth  
0  Ginger  Dachshund        22         10  2019-03-14  
1   Scout  Dalmatian        59         25  2019-05-09
```

DataFrame manipulation

```
new_dogs[ "bmi" ] = new_dogs[ "weight_kg" ] / (new_dogs[ "height_cm" ] / 100) ** 2  
print(new_dogs)
```

```
   name      breed  height_cm  weight_kg  date_of_birth        bmi  
0  Ginger  Dachshund       22         10  2019-03-14  206.611570  
1   Scout  Dalmatian       59         25  2019-05-09  71.818443
```

DataFrame to CSV

```
new_dogs.to_csv("new_dogs_with_bmi.csv")
```

`new_dogs_with_bmi.csv`

```
name,breed,height_cm,weight_kg,d_o_b,bmi  
Ginger,Dachshund,22,10,2019-03-14,206.611570  
Scout,Dalmatian,59,25,2019-05-09,71.818443
```

Let's practice!

DATA MANIPULATION WITH PANDAS

Wrap-up

DATA MANIPULATION WITH PANDAS



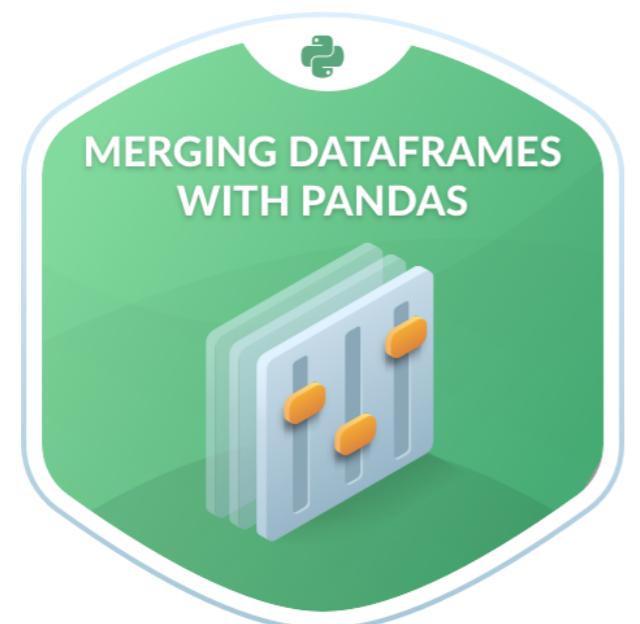
Maggie Matsui

Senior Content Developer at DataCamp

Recap

- Chapter 1
 - Subsetting and sorting
 - Adding new columns
- Chapter 2
 - Aggregating and grouping
 - Summary statistics
- Chapter 3
 - Indexing
 - Slicing
- Chapter 4
 - Visualizations
 - Reading and writing CSVs

More to learn



Congratulations!

DATA MANIPULATION WITH PANDAS