Visualizing your data

DATA MANIPULATION WITH PANDAS

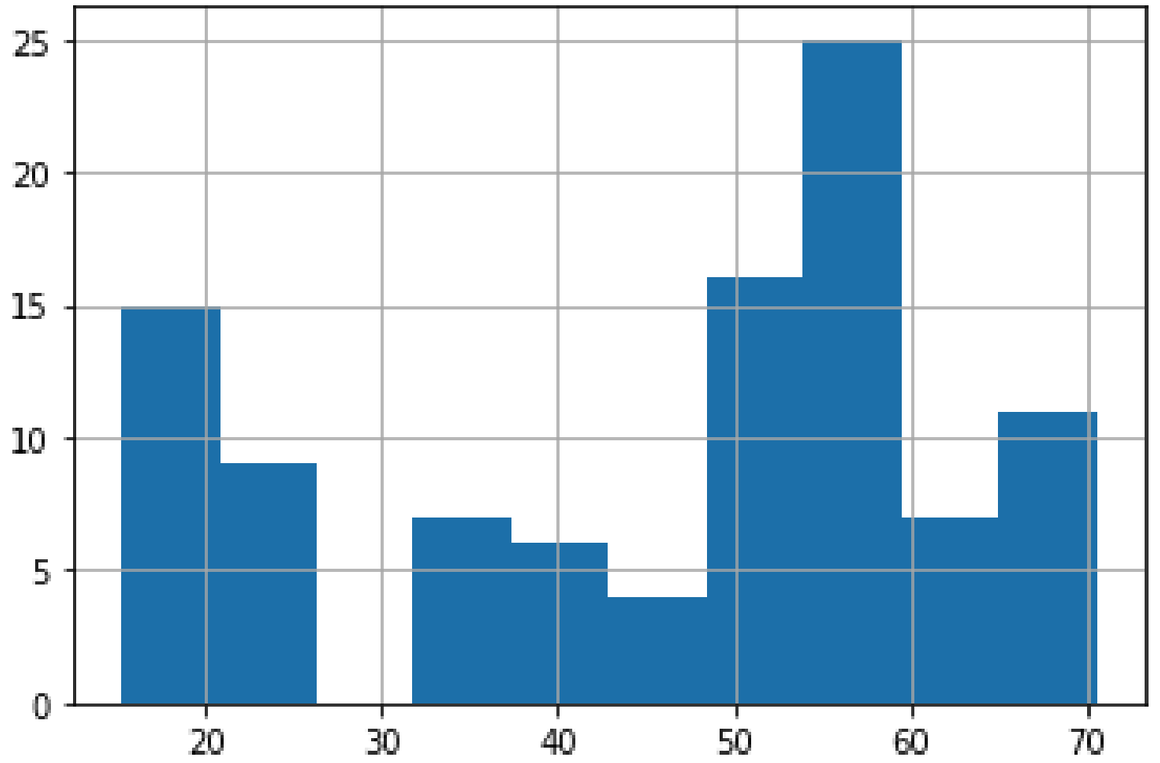
Maggie Matsui
Senior Content Developer at DataCamp

Histograms

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

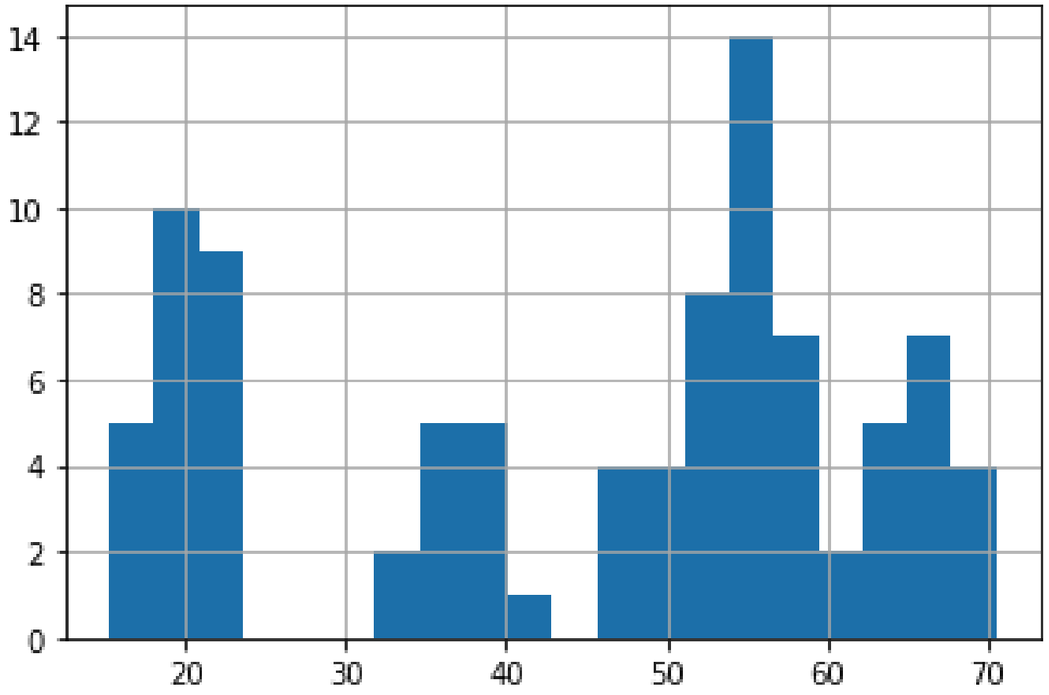
```
dog_pack["height_cm"].hist()
```

```
plt.show()
```

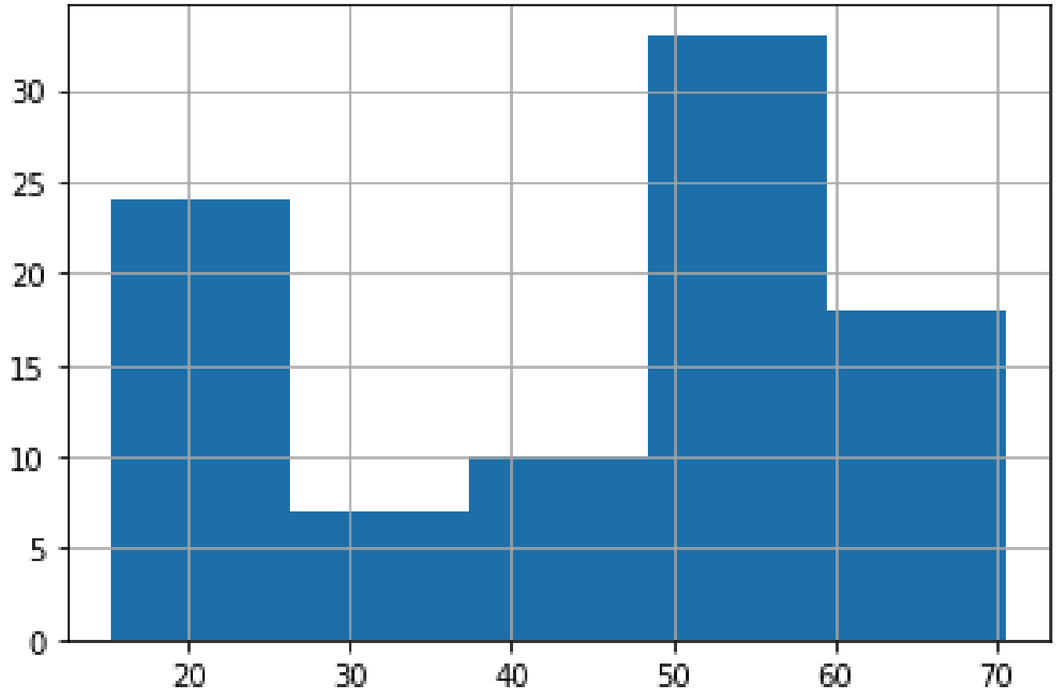


Histograms

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




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



Bar plots

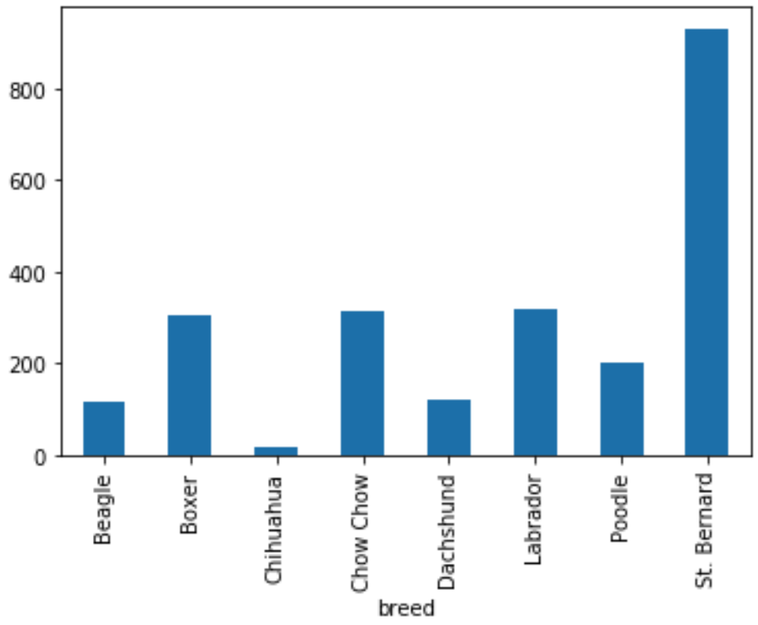
```
avg_weight_by_breed = dog_pack.groupby("breed")["weight_kg"].mean()
print(avg_weight_by_breed)
```

```
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
```

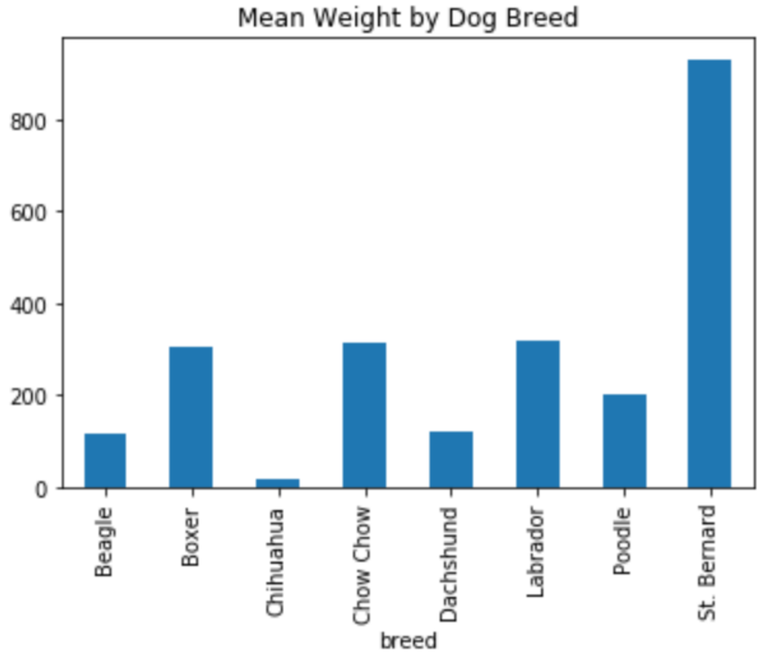
4. Bar plots
Bar plots can reveal relationships between a categorical variable and a numeric variable, like breed and weight. To compute the average weight of each breed, we group by breed, select the weight column, and take the mean, giving us the average weight of each breed.

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()
```

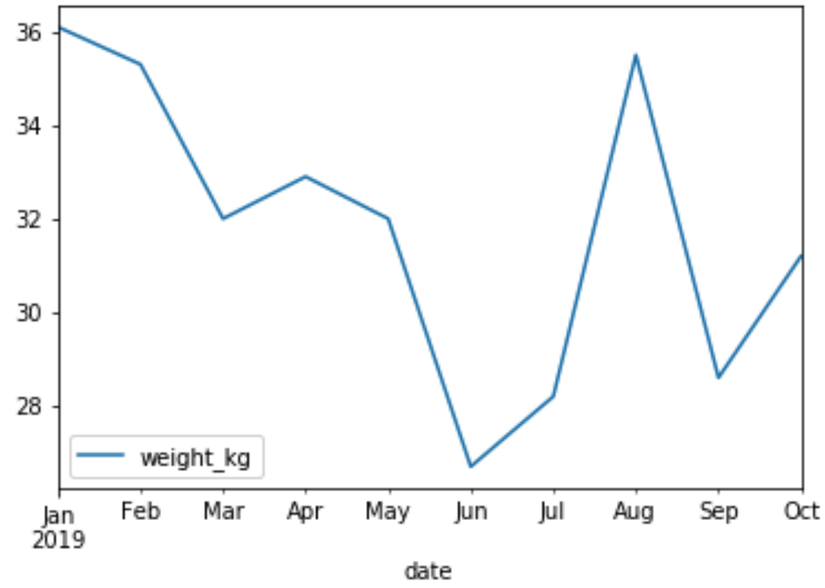


Line plots

```
sully.head()
```

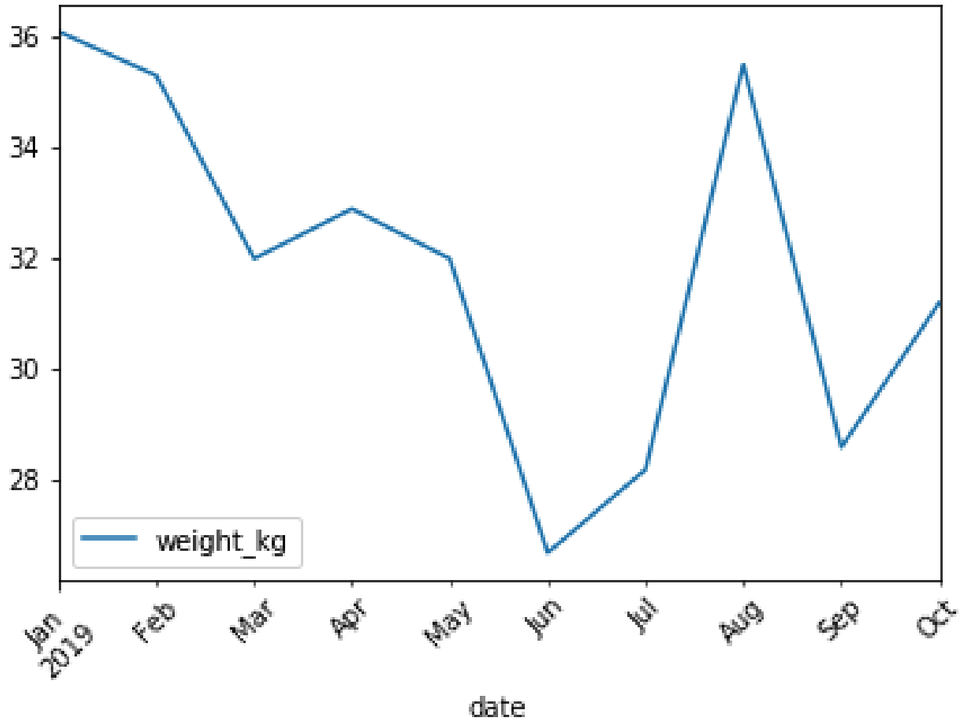
```
         date    weight_kg
0  2019-01-31         36.1
1  2019-02-28         35.3
2  2019-03-31         32.0
3  2019-04-30         32.9
4  2019-05-31         32.0
```

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



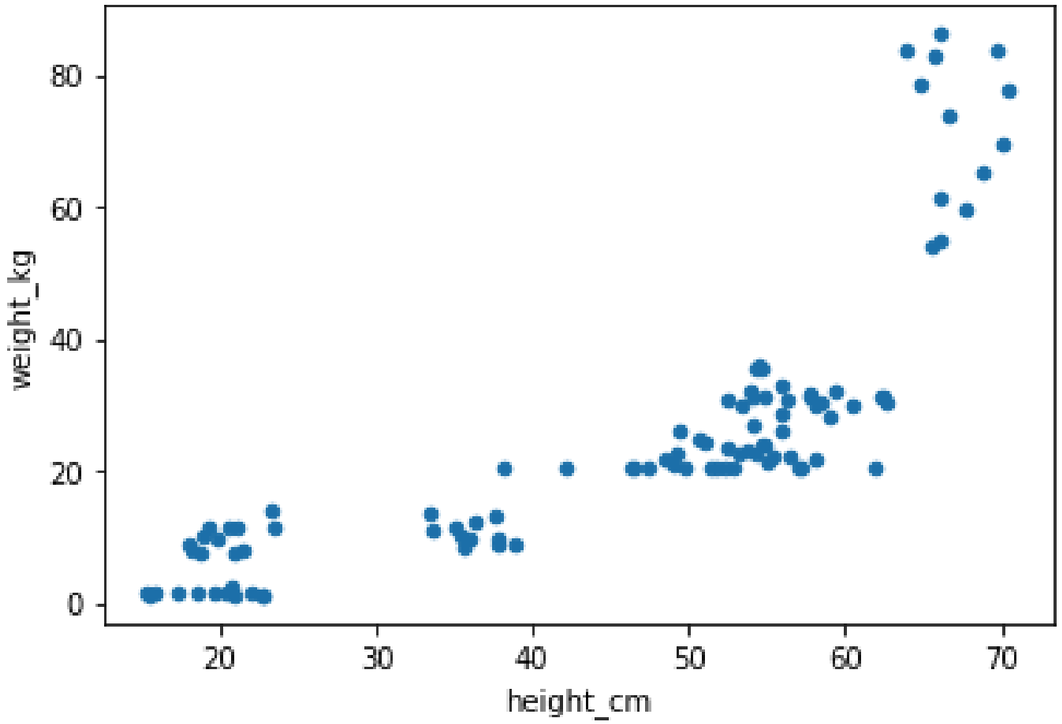
Rotating axis labels

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



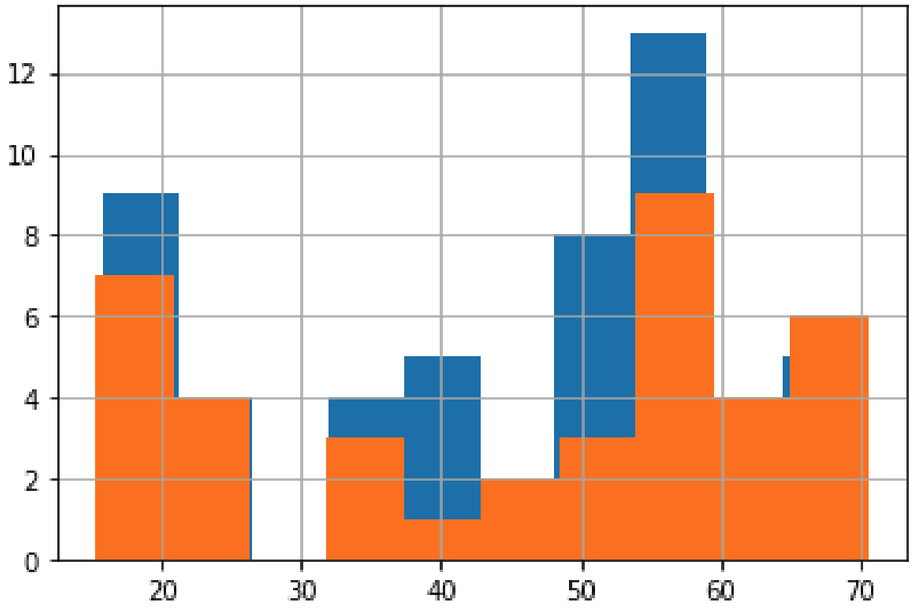
Scatter plots

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



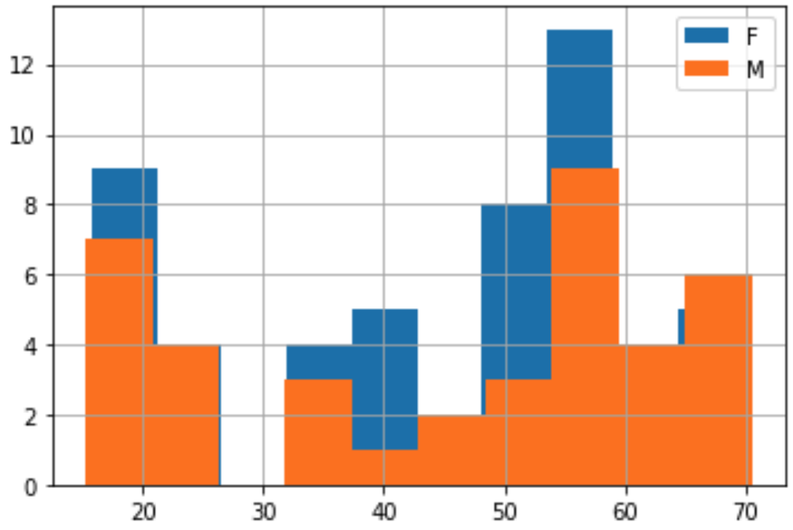
Layering plots

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



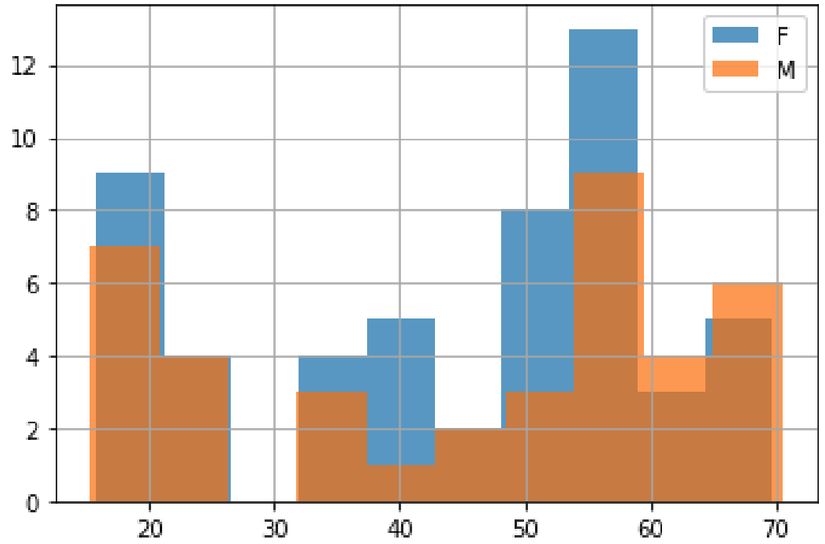
Add a legend

```
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
```

6. Detecting any missing values
If we chain dot-isna with dot-any, we get one value for each variable that tells us if there are any missing values in that column. Here, we see that there's at least one missing value in the weight column, but not in any of the others.

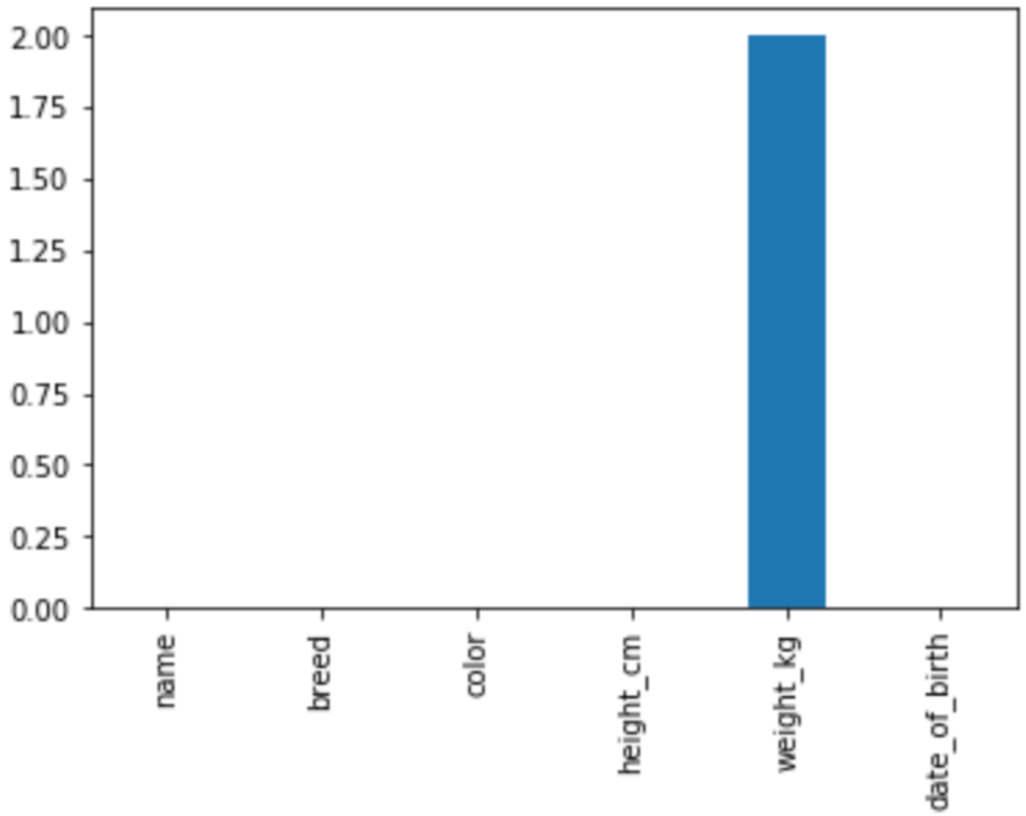
Counting missing values

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

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

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

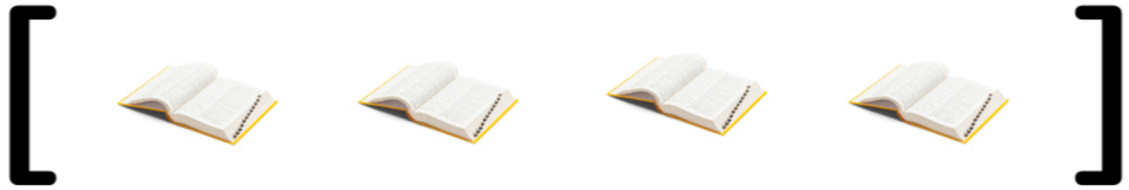
```
value1
```

```
Charlotte's Web
```

Creating DataFrames

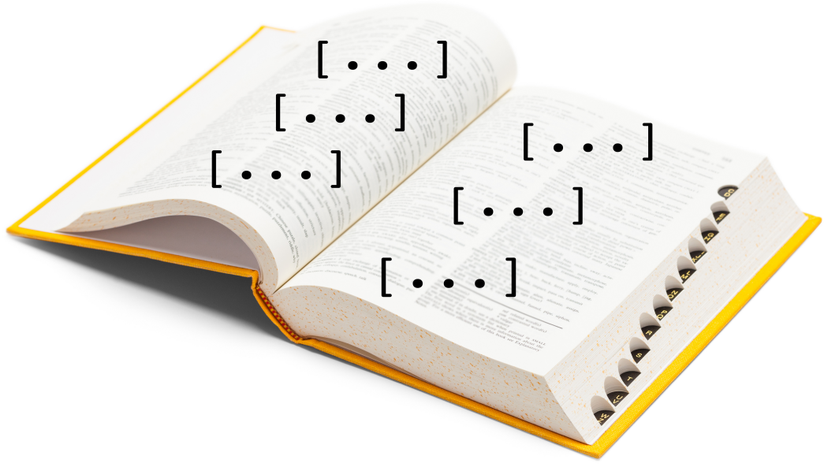
From a list of dictionaries

- 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	weight	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

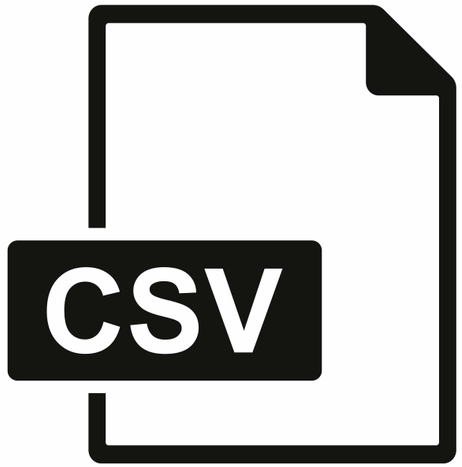
Reading and writing CSVs

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
```

6. DataFrame to CSV
Now that we've changed the data let's create an updated CSV file to share with the dogs' owners. To convert a DataFrame to a CSV, we can use new_dogs dot to-underscore-csv, and pass in a new file path. If we take a look at the new file, it contains the BMI column.

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

- **Joining Data with pandas**
- **Streamlined Data Ingestion with pandas**
- **Analyzing Police Activity with pandas**
- **Analyzing Marketing Campaigns with pandas**

3. More to learn
I hope you are convinced that pandas is a powerful tool to analyze tabular data. In fact, pandas is so powerful that there are many features that we didn't get around to discussing in this course. To begin with, everything in this course involved a single DataFrame, but sometimes you need to join or "merge" several DataFrames together. Reading from CSV files barely scratches the surface of the options for importing data into pandas. You can also perform very sophisticated exploratory data analysis using pandas.

Congratulations!

DATA MANIPULATION WITH PANDAS