1. Methods:
    a. df.mean()
    b. df.median()
    c. df.mode()
    d. df.min()
    e. df.max()
    f. df.sum()
    g. df.quantile()
    h. df.var() - variance
    i. df.std() - standard deviation
    j. df.agg() - aggregate
        i. dogs[“weight”].agg(method_that_returns_numbers)
        ii. dogs[[“weight”, “height”]].agg(methodname)
        iii. dogs[“weight”].agg([method1, method2])
    k. df.cumsum() - cumulative sum, adds values of all the rows of a certain column and prints all the sums as you go down the rows

```
[25] df['calories'].head(10)

0     70
1    120
2     70
3     50
4    110
5    110
6    110
7    130
8     90
9     90
Name: calories, dtype: int64
```

```
[24] df['calories'].head(10).cumsum() #adds all the values of all preceding rows

0     70
1    190
2    260
3    310
4    420
5    530
6    640
7    770
8    860
9    950
Name: calories, dtype: int64
```

        i. df.cummax() - cumulative maximum
        ii. df.cummin() - cumulative minimum
        iii. df.cumprod() - cumulative product

    All of these return an entire column of the dataframe, instead of just a number

    l.

2. In chapter one, you saw that DataFrames are composed of three parts: a NumPy array for the data, and two indexes to store the row and column details.
   df.columns → an Index object of column names
   df.index → an Index object of row numbers

.columns and .index

```
dogs.columns
```

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

```
dogs.index
```

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

3. You can select a feature or a column to be the new index, using:
   df_new = df.set_index(“column_name”)

Setting a column as the index

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

name	breed	color	height_cm	weight_kg
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

4. To undo the new index, do: df_new.reset_index()

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

5. If you want to completely remove the column you previously used as index, use “drop” as an argument: df_new.reset_index(drop=True)

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

Here, setting drop to True entirely removes the dog names.

6. DataFrames have a subsetting method called "loc," which filters on index values.

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"]]
```

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

Now, look at the equivalent when the names are in the index.

7. The values in the index don't need to be unique. Here, there are two Labradors in the index.

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
```

8. Use multiple columns as the index:

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
```

You can include multiple columns in the index by passing a list of column names to set_index.

9. Subset using the outer level index:

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
```

To take a subset of rows at the outer level index, you pass a list of index values to loc.

10. To subset using inner level indexes, you will need TWO tuples (one for each level). The two tuples should be in the form of a list:
df.loc[[(“outer_index_1”, “inner_index_1”), (“outer_index_2”,”inner_index_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
```

To subset on inner levels, you need to pass a list of tuples.

11. To sort a dataframe by index values: df.sort_index()
    By default, it sorts all index values from outer to inner, in ascending order

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

12. To change the default sorting parameters:

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

You can control the sorting by passing lists to the level and ascending arguments.

13.