KEY_Lesson16_Basic_Stats_II_Percents

May 28, 2020

1 Basic Statistics I: Percents

A **percentage** is a number or ratio expressed as a fraction of 100. We'll do some examples together to learn how to calculate percentages.

Example 1: For a basket of 18 fruits, there are 5 apples, 3 bananas, 6 peaches, and 4 oranges.

What percentage of fruits are apples?

[0]:
```
# Calculate percentage for apples
5/18*100
```

What percentage of fruits are oranges **and** peaches?

[0]:
```
# Calculate percentage for oranges and peaches
(4+6)/18*100
```

Example 2: Let's learn to calculate percentages by using real world data. We will work with a dataset of Boston housing prices.

[0]:
```
# Import the load_boston method
from sklearn.datasets import load_boston
```

[0]:
```
# Import pandas, so that we can work with the data frame version of the Boston␣
↪housing data
import pandas as pd
```

[0]:
```
# Load the dataset of housing prices in Boston, and convert to
# a data frame format so it's easier to view and process
boston = load_boston()
boston_df = pd.DataFrame(boston['data'], columns = boston['feature_names'])
boston_df['PRICE'] = boston.target
boston_df
```

CHAS is the indicator variable we used last week, where 1 indicates that the property (tract) is on the Charles River and 0 means otherwise.

What percentage of the tracts bound the Charles River? We'll see how to do this using the query method AND using boolean indexing.

[0]:
```
# Determine number of tracts that bound the Charles River two ways:
# (1) with the query function
num_bound_river = len(boston_df.query("CHAS == 1"))
num_bound_river
```

[0]:
```
# (2) using boolean indexing
num_bound_river = sum(boston_df["CHAS"] == 1)
num_bound_river
```

How do these two methods give the same answer?

[0]:
```
# Determine the total number of tracts in the dataset
total_num = len(boston_df)

# Now calculate the percentage of tracts that bounds the Charles River.
num_bound_river/total_num*100
```

[0]:
```
import numpy as np
```

What percentage of tracts have a median price less than $10,000?

[0]:
```
# Determine number of tracts that cost less than $10,000
num_cost_less_10k = sum(boston_df["PRICE"] < 10)

# Calculate the percentage of tracts that cost less than $10k.
num_cost_less_10k/total_num*100
```

What percentage of tracts have a median price **between** $10,000 and $30,000?

[0]:
```
# Make an array of booleans with cost greater than $10,000 AND less than $30,000
between_10k_and_30k = (boston_df["PRICE"] > 10) & (boston_df["PRICE"] < 30)

# Determine number of tracts that cost between $10,000 and $30,000
num_between_10k_and_30k = sum(between_10k_and_30k)

# Calculate the percentage of tracts between $10,000 and $30,000
num_between_10k_and_30k/total_num*100
```

Good work! You just learned about how to calculate percentages in Python!