This notebook is an exercise in the Data Cleaning course. You can reference the tutorial at this link.

In this exercise, you'll apply what you learned in the **Handling missing values** tutorial.

Setup

The questions below will give you feedback on your work. Run the following cell to set up the feedback system.

In [ ]:
```
from learntools.core import binder
binder.bind(globals())
from learntools.data_cleaning.ex1 import *
print("Setup Complete")
```

1) Take a first look at the data

Run the next code cell to load in the libraries and dataset you'll use to complete the exercise.

In [ ]:
```
# modules we'll use
import pandas as pd
import numpy as np

# read in all our data
sf_permits = pd.read_csv("../input/building-permit-applications-data/Building_Permits.csv")

# set seed for reproducibility
np.random.seed(0)
```

Use the code cell below to print the first five rows of the `sf_permits` DataFrame.

In [ ]:
```
# TODO: Your code here!
sf_permits.head()
```

Does the dataset have any missing values? Once you have an answer, run the code cell below to get credit for your work.

In [ ]:
```
# Check your answer (Run this code cell to receive credit!)
q1.check()
```

In [ ]:
```
# Line below will give you a hint
#q1.hint()
```

2) How many missing data points do we have?

What percentage of the values in the dataset are missing? Your answer should be a number between 0 and 100. (If 1/4 of the values in the dataset are missing, the answer is 25.)

In [ ]:
```
# TODO: Your code here!
total_cell = np.product(sf_permits.shape)
missing_value = sf_permits.isnull().sum()
total_missing = missing_value.sum()
percent_missing = (total_missing/total_cell)*100
# Check your answer
q2.check()
```

In [ ]:
```
# Lines below will give you a hint or solution code
#q2.hint()
#q2.solution()
```

3) Figure out why the data is missing

Look at the columns **"Street Number Suffix"** and **"Zipcode"** from the San Francisco Building Permits dataset. Both of these contain missing values.

- Which, if either, are missing because they don't exist?
- Which, if either, are missing because they weren't recorded?

Once you have an answer, run the code cell below.

In [ ]:
```
# Check your answer (Run this code cell to receive credit!)
q3.check()
```

In [ ]:
```
# Line below will give you a hint
sf_permits[0:10]
#q3.hint()
```

4) Drop missing values: rows

If you removed all of the rows of `sf_permits` with missing values, how many rows are left?

Note: Do not change the value of `sf_permits` when checking this.

In [ ]:
```
# TODO: Your code here!
sf_permits.dropna()
```

Once you have an answer, run the code cell below.

In [ ]:
```
# Check your answer (Run this code cell to receive credit!)
q4.check()
```

In [ ]:
```
# Line below will give you a hint
#q4.hint()
```

5) Drop missing values: columns

Now try removing all the columns with empty values.

- Create a new DataFrame called `sf_permits_with_na_dropped` that has all of the columns with empty values removed.
- How many columns were removed from the original `sf_permits` DataFrame? Use this number to set the value of the `dropped_columns` variable below.

In [ ]:
```
# TODO: Your code here
# remove all columns with at least one missing value
sf_permits_with_na_dropped = sf_permits.dropna(axis=1)

# calculate number of dropped columns
cols_in_original_dataset = sf_permits.shape[1]
cols_in_na_dropped = sf_permits_with_na_dropped.shape[1]
dropped_columns = cols_in_original_dataset - cols_in_na_dropped
# Check your answer
q5.check()
```

In [ ]:
```
# Lines below will give you a hint or solution code
#q5.hint()
#q5.solution()
```

In [ ]:
```
sf_permits_with_na_dropped.head()
```

6) Fill in missing values automatically

Try replacing all the NaN's in the `sf_permits` data with the one that comes directly after it and then replacing any remaining NaN's with 0. Set the result to a new DataFrame `sf_permits_with_na_imputed`.

In [ ]:
```
sf_permits_with_na_imputed = sf_permits.fillna(method='bfill', axis=0).fillna(0)

# Check your answer
q6.check()
```

In [ ]:
```
# Lines below will give you a hint or solution code
#q6.hint()
#q6.solution()
```

More practice

If you're looking for more practice handling missing values:

- Check out this noteboook on handling missing values using scikit-learn's imputer.
- Look back at the "Zipcode" column in the `sf_permits` dataset, which has some missing values. How would you go about figuring out what the actual zipcode of each address should be? (You might try using another dataset. You can search for datasets about San Fransisco on the Datasets listing.)

Keep going

In the next lesson, learn how to **apply scaling and normalization** to transform your data.

Have questions or comments? Visit the Learn Discussion forum to chat with other Learners.