Exemplar_Explore descriptive statistics

February 29, 2024

1 Exemplar: Explore descriptive statistics

1.1 Introduction

Data professionals often use descriptive statistics to understand the data they are working with and provide collaborators with a summary of the relative location of values in the data, as well an information about its spread.

For this activity, you are a member of an analytics team for the United States Environmental Protection Agency (EPA). You are assigned to analyze data on air quality with respect to carbon monoxide, a major air pollutant. The data includes information from more than 200 sites, identified by state, county, city, and local site names. You will use Python functions to gather statistics about air quality, then share insights with stakeholders.

1.2 Step 1: Imports

Import the relevant Python libraries `pandas` and `numpy`.

[1]:
```
# Import relevant Python libraries.

### YOUR CODE HERE ###

import pandas as pd
import numpy as np
```

Load the dataset into a DataFrame. The dataset provided is in the form of a .csv file named `c4_epa_air_quality.csv`. It contains a subset of data from the U.S. EPA.

[2]:
```
# Load data from the .csv file into a DataFrame and save in a variable.

### YOUR CODE HERE

epa_data = pd.read_csv("c4_epa_air_quality.csv", index_col = 0)
```

Hint 1

Refer to the video about loading data in Python.

Hint 2

There is a function in the `pandas` library that allows you to read in data from a .csv file and load it into a DataFrame.

Hint 3

Use the `read_csv` function from the pandas `library`. The `index_col` parameter can be set to `0` to read in the first column as an index (and to avoid `"Unnamed: 0"` appearing as a column in the resulting DataFrame).

1.3 Step 2: Data exploration

To understand how the dataset is structured, display the first 10 rows of the data.

[3]:
```
# Display first 10 rows of the data.

### YOUR CODE HERE

epa_data.head(10)
```

[3]:

	date_local	state_name	county_name	city_name	local_site_name	parameter_name	units_of_measure	arithmetic_mean	aqi
250	2018-01-01	California	Los Angeles	West Los Angeles	West Los Angeles	Carbon monoxide	Parts per million	0.655556	11
251	2018-01-01	Colorado	Denver	Denver	La Casa	Carbon monoxide	Parts per million	0.342105	5
252	2018-01-01	Ohio	Hamilton	Cincinnati	Cincinnati Near Road	Carbon monoxide	Parts per million	0.226316	3
253	2018-01-01	Oregon	Washington	Tualatin	Tualatin Bradbury Court (TBC) - Near Road Site	Carbon monoxide	Parts per million	0.100000	1
254	2018-01-01	Arizona	Pima	Tucson	CHERRY & GLENN	Carbon monoxide			
255	2018-01-01	District Of Columbia	District of Columbia	Washington	Near Road	Carbon monoxide			
256	2018-01-01	Wisconsin	Dodge	Kekoskee	HORICON WILDLIFE AREA	Carbon monoxide			
257	2018-01-01	Kentucky	Jefferson	Louisville	CANNONS LANE	Carbon monoxide			
258	2018-01-01	Nebraska	Douglas	Omaha	NaN	Carbon monoxide			
259	2018-01-01	North Carolina	Wake	Not in a city	Triple Oak	Carbon monoxide			

```
254  Parts per million         0.563158   14
255  Parts per million         0.244444    3
256  Parts per million         0.200000    2
257  Parts per million         0.163158    2
258  Parts per million         0.421053    9
259  Parts per million         0.188889    2
```

Hint 1

Refer to the video about exploratory data analysis in Python.

Hint 2

There is a function in the `pandas` library that allows you to get a specific number of rows from the top of a DataFrame.

Hint 3

Use the `head()` function from the `pandas` library.

Question: What does the `aqi` column represent?

The `aqi` column represents the EPA's Air Quality Index (AQI).

Now, get a table that contains some descriptive statistics about the data.

[4]:
```
# Get descriptive stats.

### YOUR CODE HERE

epa_data.describe()
```

[4]:
```
       arithmetic_mean         aqi
count       260.000000  260.000000
mean          0.403169    6.757692
std           0.317902    7.061707
min           0.000000    0.000000
25%           0.200000    2.000000
50%           0.276315    5.000000
75%           0.516009    9.000000
max           1.921053   50.000000
```

Hint 1

Refer to the video about descriptive statistics in Python.

Hint 2

There is a function in the `pandas` library that allows you to generate a table of basic descriptive statistics about the numeric columns in a DataFrame.

Hint 3

Use the `describe()` function from the `pandas` library.

Question: Based on the table of descriptive statistics, what do you notice about the count value for the `aqi` column?

The count value for the `aqi` column is 260. This means there are 260 aqi measurements represented in this dataset.

Question: What do you notice about the 25th percentile for the `aqi` column? This is an important measure for understanding where the aqi values lie.

The 25th percentile for the `aqi` column is 2. This means that 25% of the aqi values in the data are below 2.

Question: What do you notice about the 75th percentile for the `aqi` column? This is another important measure for understanding where the aqi values lie.

The 75th percentile for the aqi column is 9. This means that 75% of the aqi values in the data are below 9.

1.4 Step 3: Statistical tests ## Step 3. Statistical Tests

Next, get some descriptive statistics about the states in the data.

[5]:
```
# Get descriptive stats about the states in the data.

### YOUR CODE HERE

epa_data["state_name"].describe()
```

[5]:
```
count            260
unique            52
top       California
freq              66
Name: state_name, dtype: object
```

Hint 1

Refer to the video about descriptive statistics in Python.

Hint 2

There is a function in the `pandas` library that allows you to generate basic descriptive statistics about a DataFrame or a column you are interested in.

Hint 3

Use the `describe()` function from the `pandas` library. Note that this function can be used: - “on a DataFrame (to find descriptive statistics about the numeric columns)” - “directly on a column containing categorical data (to find pertinent descriptive statistics)”

Question: What do you notice while reviewing the descriptive statistics about the states in the data?

Note: Sometimes you have to individually calculate statistics. To review to that approach, use the `numpy` library to calculate each of the main statistics in the preceding table for the `aqi` column.

There are 260 state values, and 52 of them are unique. California is the most commonly occurring state in the data, with a frequency of 66. (In other words, 66 entries in the data correspond to aqi measurements taken in California.)

1.5 Step 4. Results and evaluation

Now, compute the mean value from the `aqi` column.

[6]:
```
# Compute the mean value from the aqi column.

### YOUR CODE HERE

np.mean(epa_data["aqi"])
```

[6]:
```
6.757692307692308
```

Hint 1

Refer to the video about descriptive statistics in Python.

Hint 2

There is a function in the `numpy` library that allows you to get the mean value from an array or a Series of values.

Hint 3

Use the `mean()` function from the `numpy` library.

Question: What do you notice about the mean value from the `aqi` column?

This is an important measure, as it tells you what the average air quality is based on the data.

The mean value for the `aqi` column is approximately 6.76 (rounding to 2 decimal places here). This means that the average aqi from the data is approximately 6.76.

Next, compute the median value from the aqi column.

[7]:
```
# Compute the median value from the aqi column.

### YOUR CODE HERE

np.median(epa_data["aqi"])
```

[7]:
```
5.0
```

Hint 1

Refer to the video about descriptive statistics in Python.

Hint 2

There is a function in the `numpy` library that allows you to get the median value from an array or a series of values.

Hint 3

Use the `median()` function from the `numpy` library.

Question: What do you notice about the median value from the `aqi` column? This is an important measure for understanding the central location of the data.

The median value for the aqi column is 5.0. This means that half of the aqi values in the data are below 5.

Next, identify the minimum value from the `aqi` column.

[8]:
```
# Identify the minimum value from the aqi column.

### YOUR CODE HERE

np.min(epa_data["aqi"])
```

[8]:
```
0
```

Hint 1

Refer to the video about descriptive statistics in Python.

Hint 2

There is a function in the `numpy` library that allows you to get the minimum value from an array or a Series of values.

Hint 3

Use the `min()` function from the `numpy` library.

Question: What do you notice about the minimum value from the aqi column? This is an important measure, as it tell you the best air quality observed in the data.

The minimum value for the `aqi` column is 0. This means that the smallest aqi value in the data is 0.

Now, identify the maximum value from the `aqi` column.

[9]:
```
# Identify the maximum value from the aqi column.

### YOUR CODE HERE

np.max(epa_data["aqi"])
```

[9]:
```
50
```

Hint 1

Refer to the video about descriptive statistics in Python.

Hint 2

There is a function in the `numpy` library that allows you to get the maximum value from an array or a Series of values.

Hint 3

Use the `max()` function from the `numpy` library.

Question: What do you notice about the maximum value from the `aqi` column? This is an important measure, as it tells you which value in the data corresponds to the worst air quality observed in the data.

The maximum value for the `aqi` column is 50. This means that the largest aqi value in the data is 50.

Now, compute the standard deviation for the `aqi` column.

By default, the `numpy` library uses 0 as the Delta Degrees of Freedom, while `pandas` library uses 1. To get the same value for standard deviation using either library, specify the `ddof` parameter to 1 when calculating standard deviation.

[10]:
```
# Compute the standard deviation for the aqi column.

### YOUR CODE HERE

np.std(epa_data["aqi"], ddof=1)
```

[10]:
```
7.0617066788207215
```

Hint 1

Refer to the video section about descriptive statistics in Python.

Hint 2

There is a function in the `numpy` library that allows you to get the standard deviation from an array or a series of values.

Hint 3

Use the `std()` function from the `numpy` library. Make sure to specify the `ddof` parameter as 1. To read more about this function, refer to its documentation in the references section of this lab.

Question: What do you notice about the standard deviation for the `aqi` column? This is an important measure of how spread out the aqi values are.

The standard deviation for the aqi column is approximately 7.05 (rounding to 2 decimal places here). This is a measure of how spread out the aqi values are in the data.

1.6 Considerations

What are some key takeaways that you learned during this lab? Functions in the `pandas` and `numpy` libraries can be used to find statistics that describe a dataset. The `describe()` function

from `pandas` generates a table of descriptive statistics about numerical or categorical columns. The `mean()`, `median()`, `min()`, `max()`, and `std()` functions from `numpy` are useful for finding individual statistics about numerical data.

How would you present your findings from this lab to others? Consider the following relevant points noted by AirNow.gov as you respond: - "AQI values at or below 100 are generally thought of as satisfactory. When AQI values are above 100, air quality is considered to be unhealthy—at first for certain sensitive groups of people, then for everyone as AQI values increase." - "An AQI of 100 for carbon monoxide corresponds to a level of 9.4 parts per million."

The average AQI value in the data is approximately 6.76, which is considered safe with respect to carbon monoxide. Further, 75% of the AQI values are below 9.

What summary would you provide to stakeholders? Use the same information provided previously from AirNow.gov as you respond.

- 75% of the AQI values in the data are below 9, which is considered good air quality.
- Funding should be allocated for further investigation of the less healthy regions in order to learn how to improve the conditions.

References

Air Quality Index - A Guide to Air Quality and Your Health. (2014,February)

Numpy.Std — NumPy v1.23 Manual

US EPA, OAR. (2014, 8 July).*Air Data: Air Quality Data Collected at Outdoor Monitors Across the US*.

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.