Activity_Explore sampling

January 7, 2024

1 Activity: Explore sampling

1.1 Introduction

In this activity, you will engage in effective sampling of a dataset in order to make it easier to analyze. As a data professional you will often work with extremely large datasets, and utilizing proper sampling techniques helps you improve your efficiency in this work.

For this activity, you are a member of an analytics team for the Environmental Protection Agency. You are assigned to analyze data on air quality with respect to carbon monoxide—a major air pollutant—and report your findings. The data utilized in this activity includes information from over 200 sites, identified by their state name, county name, city name, and local site name. You will use effective sampling within this dataset.

1.2 Step 1: Imports

1.2.1 Import packages

Import `pandas`, `numpy`, `matplotlib`, `statsmodels`, and `scipy`.

[1]:
```
# Import libraries and packages

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from scipy import stats
```

1.2.2 Load the dataset

As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

[2]:
```
# RUN THIS CELL TO IMPORT YOUR DATA.
```

```
### YOUR CODE HERE ###
epa_data = pd.read_csv("c4_epa_air_quality.csv", index_col = 0)
```

Hint 1

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

Hint 2

Use the `read_csv` function from the pandas `library`. Set the `index_col` parameter 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

1.3.1 Examine the data

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

[3]:
```
# First 10 rows of the data

epa_data.head(10)
```

[3]:

	date_local	state_name	county_name	city_name	local_site_name	parameter_name
0	2018-01-01	Arizona	Maricopa	Buckeye	BUCKEYE	Carbon monoxide
1	2018-01-01	Ohio	Belmont	Shadyside	Shadyside	Carbon monoxide
2	2018-01-01	Wyoming	Teton	Not in a city	Yellowstone National Park - Old Faithful Snow …	Carbon monoxide
3	2018-01-01	Pennsylvania	Philadelphia	Philadelphia	North East Waste (NEW)	Carbon monoxide
4	2018-01-01	Iowa	Polk	Des Moines	CARPENTER	Carbon monoxide
5	2018-01-01	Hawaii	Honolulu	Not in a city	Kapolei	Carbon monoxide
6	2018-01-01	Hawaii	Honolulu	Not in a city	Kapolei	Carbon monoxide
7	2018-01-01	Pennsylvania	Erie	Erie	NaN	Carbon monoxide
8	2018-01-01	Hawaii	Honolulu	Honolulu	Honolulu	Carbon monoxide
9	2018-01-01	Colorado	Larimer	Fort Collins	Fort Collins - CSU - S. Mason	Carbon monoxide

	units_of_measure	arithmetic_mean	aqi
0	Parts per million	0.473684	7
1	Parts per million	0.263158	5
2	Parts per million	0.111111	2
3	Parts per million	0.300000	3
4	Parts per million	0.215789	3
5	Parts per million	0.994737	14
6	Parts per million	0.200000	2
7	Parts per million	0.200000	2
8	Parts per million	0.400000	5
9	Parts per million	0.300000	6

Hint 1

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

Hint 2

Use the `head` function from the `pandas` library. Set the `n` parameter to `10` to print out the first 10 rows.

Question: What does the `aqi` column represent?

The aqi is showing the air quality rating.

1.3.2 Generate a table of descriptive statistics

Generate a table of some descriptive statistics about the data. Specify that all columns of the input be included in the output.

[4]:
```
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

Use function in the `pandas` library that allows you to generate a table of basic descriptive statistics in a DataFrame.

Hint 2

Use the `describe` function from the `pandas` library. Set the `include` parameter passed in to this function to 'all' to specify that all columns of the input be included in the output.

Question: Based on the preceding table of descriptive statistics, what is the mean value of the `aqi` column?

The mean aqi is 6.76

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

It is equal to the arithmetic mean column

1.3.3 Use the `mean()` function on the `aqi` column

Now, use the `mean()` function on the `aqi` column and assign the value to a variable `population_mean`. The value should be the same as the one generated by the `describe()` method in the above table.

[7]:
```
population_mean = epa_data['aqi'].mean()
population_mean
```

[7]:
```
6.757692307692308
```

Hint 1

Use the function in the `pandas` library that allows you to generate a mean value for a column in a DataFrame.

Hint 2

Use the `mean()` method.

1.4 Step 3: Statistical tests

1.4.1 Sample with replacement

First, name a new variable `sampled_data`. Then, use the `sample()` dataframe method to draw 50 samples from `epa_data`. Set `replace` equal to `'True'` to specify sampling with replacement. For `random_state`, choose an arbitrary number for random seed. Make that arbitrary number `42`.

[8]:
```
sampled_data = epa_data.sample(n = 50, replace= True, random_state=42)
```

1.4.2 Output the first 10 rows

Output the first 10 rows of the DataFrame.

[9]:
```
sampled_data.head(10)
```

[9]:

	date_local	state_name	county_name	city_name	local_site_name	parameter_name	units_of_measure	arithmetic_mean	aqi
102	2018-01-01	Texas	Harris	Houston	Clinton	Carbon monoxide	Parts per million	0.157895	2
106	2018-01-01	California	Imperial	Calexico	Calexico-Ethel Street	Carbon monoxide	Parts per million	1.183333	26
71	2018-01-01	Alabama	Jefferson	Birmingham	Arkadelphia/Near Road	Carbon monoxide	Parts per million	0.200000	2
188	2018-01-01	Arizona	Maricopa	Tempe	Diablo	Carbon monoxide	Parts per million	0.542105	10
20	2018-01-01	Virginia	Roanoke	Vinton	East Vinton Elementary School	Carbon monoxide	Parts per million	0.100000	1
102	2018-01-01	Texas	Harris	Houston	Clinton	Carbon monoxide	Parts per million	0.157895	2
121	2018-01-01	North Carolina	Mecklenburg	Charlotte	Garinger High School	Carbon monoxide	Parts per million	0.200000	2
214	2018-01-01	Florida	Broward	Davie	Daniela Banu NCORE	Carbon monoxide	Parts per million	0.273684	5
87	2018-01-01	California	Humboldt	Eureka	Jacobs	Carbon monoxide	Parts per million	0.393750	5
99	2018-01-01	California	Santa Barbara	Goleta	Goleta	Carbon monoxide	Parts per million	0.222222	3

Hint 1

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

Hint 2

Use the `head` function from the `pandas` library. Set the `n` parameter to `10` to print out the first 10 rows.

Question: In the DataFrame output, why is the row index 102 repeated twice?

We set the replace to True so we allowed the random selections be chosen twice.

Question: What does `random_state` do?

Random state allows us to choose a random seed to then go through and select variables at random

1.4.3 Compute the mean value from the `aqi` column

Compute the mean value from the `aqi` column in `sampled_data` and assign the value to the variable `sample_mean`.

[11]:
```
sampled_mean = sampled_data['aqi'].mean()
sampled_mean
```

[11]: 5.54

Question: Why is `sample_mean` different from `population_mean`?

This will not be the same due to the random seed being smaller than the entire population.

1.4.4 Apply the central limit theorem

Imagine repeating the the earlier sample with replacement 10,000 times and obtaining 10,000 point estimates of the mean. In other words, imagine taking 10,000 random samples of 50 AQI values and computing the mean for each sample. According to the **central limit theorem**, the mean of a sampling distribution should be roughly equal to the population mean. Complete the following steps to compute the mean of the sampling distribution with 10,000 samples.

- Create an empty list and assign it to a variable called `estimate_list`.
- Iterate through a `for` loop 10,000 times. To do this, make sure to utilize the `range()` function to generate a sequence of numbers from 0 to 9,999.
- In each iteration of the loop, use the `sample()` function to take a random sample (with replacement) of 50 AQI values from the population. Do not set `random_state` to a value.
- Use the list `append()` function to add the value of the sample `mean` to each item in the list.

[12]:
```
estimate_list = []
for i in range(10000):
    estimate_list.append(epa_data['aqi'].sample(n=50,replace=True).mean())
```

Hint 1

Review the content about sampling in Python.

1.4.5 Create a new DataFrame

Next, create a new DataFrame from the list of 10,000 estimates. Name the new variable `estimate_df`.

[13]:
```
estimate_df = pd.DataFrame(data={'estimate': estimate_list})
estimate_df
```

[13]:
```
      estimate
0         7.72
1         4.64
2         5.82
3         6.40
4         5.24
…          …
9995      7.54
9996      6.58
9997      6.72
9998      9.12
9999      7.40

[10000 rows x 1 columns]
```

Hint 1

Review the content about sampling in Python.

Hint 2

Use the `mean()` function.

1.4.6 Compute the mean() of the sampling distribution

Next, compute the `mean()` of the sampling distribution of 10,000 random samples and store the result in a new variable `mean_sample_means`.

[18]:
```
mean_sample_means = estimate_df['estimate'].mean()
mean_sample_means
```

[18]:
```
6.749734000000004
```

Hint 1

Use the function in the `pandas` library that allows you to generate a mean value for a column in a DataFrame.

Hint 2

Use the `mean()` function.

Question: What is the mean for the sampling distribution of 10,000 random samples?

6.75 but will vary as the random_state wasn't set to a value

Hint 3

This value is contained in `mean_sample_means`.

Hint 4

According to the central limit theorem, the mean of the preceding sampling distribution should be roughly equal to the population mean.

Question: How are the central limit theorem and random sampling (with replacement) related?

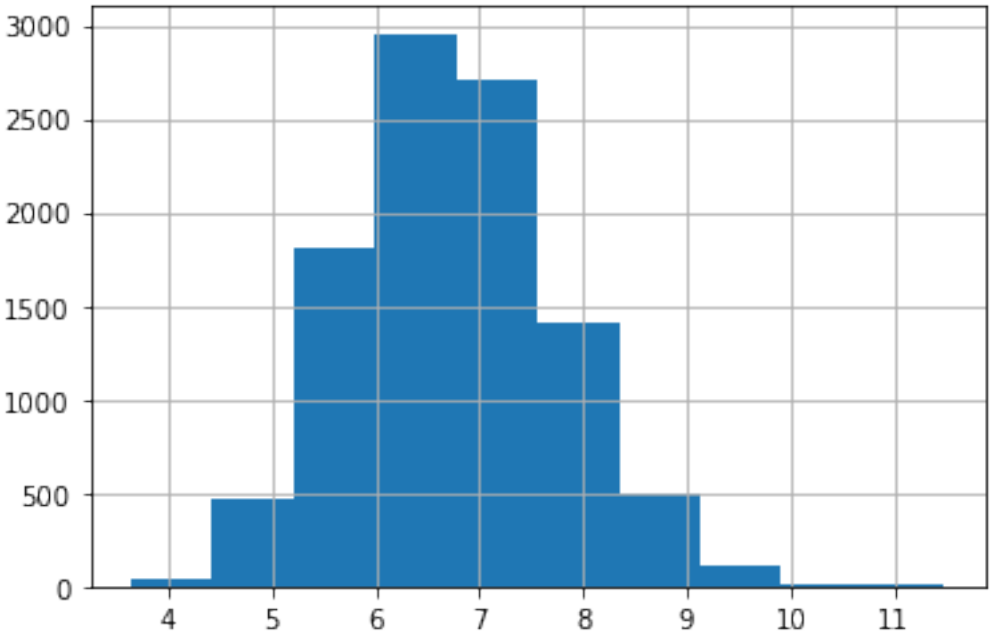
Both methods help identify the standard error and the normal distribution

1.4.7 Output the distribution using a histogram

Output the distribution of these estimates using a histogram. This provides an idea of the sampling distribution.

[20]:
```
estimate_df['estimate'].hist()
```

[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8c526dda10>
```



Hint 1

Use the `hist()` function.

1.4.8 Calculate the standard error

Calculate the standard error of the mean AQI using the initial sample of 50. The **standard error** of a statistic measures the sample-to-sample variability of the sample statistic. It provides a numerical measure of sampling variability and answers the question: How far is a statistic based on one particular sample from the actual value of the statistic?

[21]:
```
standard_error = sampled_data['aqi'].std() / np.sqrt(len(sampled_data))
standard_error
```

[21]:
```
0.7413225908290327
```

Hint 1

Use the `std()` function and the `np.sqrt()` function.

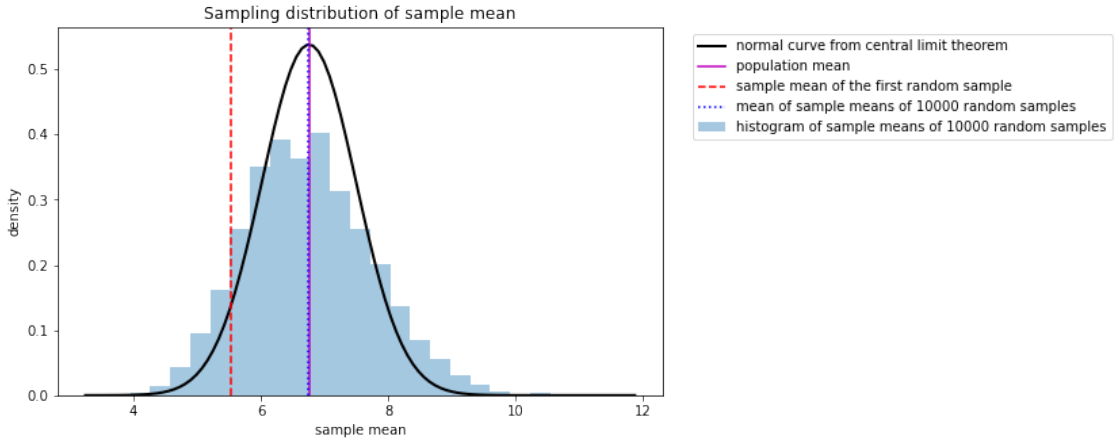
1.5 Step 4: Results and evaluation

1.5.1 Visualize the relationship between the sampling and normal distributions

Visualize the relationship between your sampling distribution of 10,000 estimates and the normal distribution.

1. Plot a histogram of the 10,000 sample means
2. Add a vertical line indicating the mean of the first single sample of 50
3. Add another vertical line indicating the mean of the means of the 10,000 samples
4. Add a third vertical line indicating the mean of the actual population

[23]:
```
plt.figure(figsize=(8,5))
plt.hist(estimate_df['estimate'], bins=25, density=True, alpha=0.4, label =␣
 ↪"histogram of sample means of 10000 random samples")
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100) # generate a grid of 100 values from xmin to␣
 ↪xmax.
p = stats.norm.pdf(x, population_mean, standard_error)
plt.plot(x, p, 'k', linewidth=2, label = 'normal curve from central limit␣
 ↪theorem')
plt.axvline(x=population_mean, color='m', linestyle = 'solid', label =␣
 ↪'population mean')
plt.axvline(x=sampled_mean, color='r', linestyle = '--', label = 'sample mean␣
 ↪of the first random sample')
plt.axvline(x=mean_sample_means, color='b', linestyle = ':', label = 'mean of␣
 ↪sample means of 10000 random samples')
plt.title("Sampling distribution of sample mean")
plt.xlabel('sample mean')
plt.ylabel('density')
plt.legend(bbox_to_anchor=(1.04,1));
```



Question: What insights did you gain from the preceding sampling distribution?

The histogram shows the is very close to the mean using central limit theorem. The red dotted line shows it's a bit off due to having a seperate value of random state.But the population mean and sample mean are right in line with one another meaning they are equal to each other.

2 Considerations

What are some key takeaways that you learned from this lab? Using random sampling can be done in multiple ways, using central limit theorem will really come in handy when working with very large datasets. **What findings would you share with others?** These findings are showing those of "healthy" conditions and the results of those falling out of that range would need to explored still. **What would you convey to external stakeholders?** In most cases the aqi is healthy and has no glaring negative impacts, but there are areas that fall outside of this study that need to be looked into with more focus on why those areas are considered to be unhealthy.

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.