Frequentist Inference Case Study - Part B

Learning objectives

Welcome to Part B of the Frequentist inference case study! The purpose of this case study is to help you apply the concepts associated with Frequentist inference in Python. In particular, you'll practice writing Python code to apply the following statistical concepts:

- the *z*-statistic
- the *t*-statistic
- the difference and relationship between the two
- the Central Limit Theorem, including its assumptions and consequences
- how to estimate the population mean and standard deviation from a sample
- the concept of a sampling distribution of a test statistic, particularly for the mean
- how to combine these concepts to calculate a confidence interval

In the previous notebook, we used only data from a known normal distribution. **You'll now tackle real data, rather than simulated data, and answer some relevant real-world business problems using the data.**

Hospital medical charges

Imagine that a hospital has hired you as their data scientist. An administrator is working on the hospital's business operations plan and needs you to help them answer some business questions.

In this assignment notebook, you're going to use frequentist statistical inference on a data sample to answer the questions:

- has the hospital's revenue stream fallen below a key threshold?
- are patients with insurance really charged different amounts than those without?

Answering that last question with a frequentist approach makes some assumptions, and requires some knowledge, about the two groups.

We are going to use some data on medical charges obtained from Kaggle (https://www.kaggle.com/easonlai/sample-insurance-claim-prediction-dataset).

For the purposes of this exercise, assume the observations are the result of random sampling from our single hospital. Recall that in the previous assignment, we introduced the Central Limit Theorem (CLT), and its consequence that the distributions of sample statistics approach a normal distribution as n increases. The amazing thing about this is that it applies to the sampling distributions of statistics that have been calculated from even highly non-normal distributions of data! Recall, also, that hypothesis testing is very much based on making inferences about such sample statistics. You're going to rely heavily on the CLT to apply frequentist (parametric) tests to answer the questions in this notebook.

In [39]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import t
from numpy.random import seed
medical = pd.read_csv('/Users/jasonzhou/Downloads/insurance2.csv')
```

In [3]:
```
medical.shape
```

Out[3]:
```
(1338, 8)
```

In [4]:
```
medical.head()
```

Out[4]:

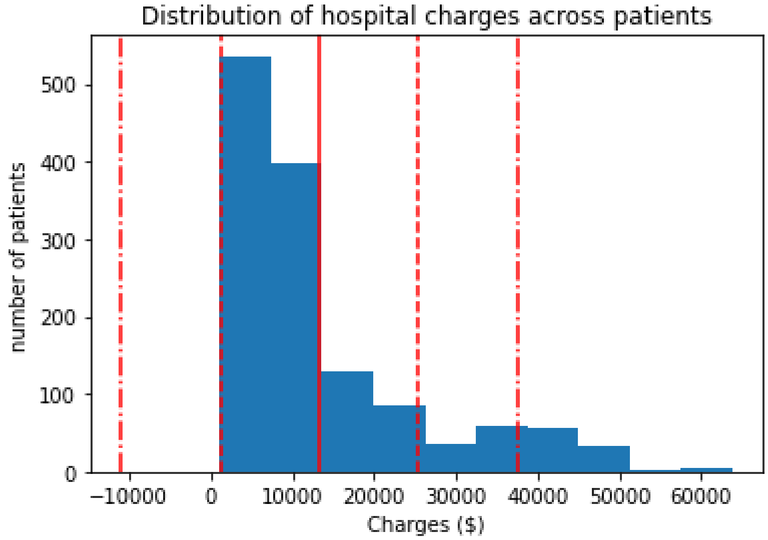
	age	sex	bmi	children	smoker	region	charges	insuranceclaim
0	19	0	27.900	0	1	3	16884.92400	1
1	18	1	33.770	1	0	2	1725.55230	1
2	28	1	33.000	3	0	2	4449.46200	0
3	33	1	22.705	0	0	1	21984.47061	0
4	32	1	28.880	0	0	1	3866.85520	1

Q1: Plot the histogram of charges and calculate the mean and standard deviation. Comment on the appropriateness of these statistics for the data.

A: The overall shape of the graph could make sense, most medical needs should fall on the cheaper side, with more serious injuries/sicknesses being less common

In [12]:

```
_ = plt.hist(medical['charges'], bins=10)
_ = plt.xlabel('Charges ($)')
_ = plt.ylabel('number of patients')
_ = plt.title('Distribution of hospital charges across patients')
_ = plt.axvline(chargesmean, color='r')
_ = plt.axvline(chargesmean + chargesstd, color='r', linestyle='--')
_ = plt.axvline(chargesmean - chargesstd, color='r', linestyle='--')
_ = plt.axvline(chargesmean + 2 * chargesstd, color='r', linestyle='-.')
_ = plt.axvline(chargesmean - 2 * chargesstd, color='r', linestyle='-.')
```



In [10]:

```
chargesmean = np.mean(medical['charges'])
print('Mean: ' + str(chargesmean))
chargesstd = np.std(medical['charges'])
print('Std: ' + str(chargesstd))
```

```
Mean: 13270.422265141257
Std: 12105.484975561605
```

Q2: The administrator is concerned that the actual average charge has fallen below 12,000, threatening the hospital's operational model. On the assumption that these data represent a random sample of charges, how would you justify that these data allow you to answer that question? And what would be the most appropriate frequentist test, of the ones discussed so far, to apply?

A: I would say that the sample size we have to work with is enough to say that it is representative of the patient population. We'll want to perform some frequentist inference most likely.

Q3: Given the nature of the administrator's concern, what is the appropriate confidence interval in this case? A ***one-sided*** or ***two-sided*** interval? (Refresh your understanding of this concept on p. 399 of the *AoS*). Calculate the critical value and the relevant 95% confidence interval for the mean, and comment on whether the administrator should be concerned.

A: We'd want to use an upper one-sided interval because we only care about if the sample's mean is greater than the target or not.

```
In [18]: tvalue = t.ppf(0.95, df=len(medical) - 1)
         term = tvalue * chargesstd / np.sqrt(len(medical))
         print(term)

         544.7314053390934
```

```
In [19]: intervalbound = chargesmean - term
         print("Confidence Interval: "+ str(intervalbound) + ", INF)")

         Confidence Interval: 12725.690859802164, INF)
```

```
In [ ]: # Based on this calculated CI, we are fairly confident that the actual aver
        # it, in fact.
```

The administrator then wants to know whether people with insurance really are charged a different amount to those without.

Q4: State the null and alternative hypothesis here. Use the *t*-test for the difference between means, where the pooled standard deviation of the two groups is given by:

$$s_p = \sqrt{\frac{(n_0 - 1)s^2_0 + (n_1 - 1)s^2_1}{n_0 + n_1 - 2}}$$

and the *t*-test statistic is then given by:

$$t = \frac{\bar{x}_0 - \bar{x}_1}{s_p \sqrt{1/n_0 + 1/n_1}}.$$

(If you need some reminding of the general definition of ***t-statistic***, check out the definition on p. 404 of *AoS*).

What assumption about the variances of the two groups are we making here?

A: That their population standard deviations are the same.

Q5: Perform this hypothesis test both manually, using the above formulae, and then using the appropriate function from [scipy.stats (https://docs.scipy.org/doc/scipy/reference/stats.html#statistical-tests)](https://docs.scipy.org/doc/scipy/reference/stats.html#statistical-tests) (hint, you're looking for a function to perform a *t*-test on two independent samples). For the manual approach, calculate the value of the test statistic and then its probability (the p-value). Verify you get the same results from both.

A: h0: People are charged the same amount regardless of whether or not they have insurance h1: People are charged differently based on whether or not they have insurance

```
0: no insurance
1: has insurance
```

In [55]:

```
# Manual Calculations

sample0 = medical['charges'][medical['insuranceclaim'] == 0]
sample1 = medical['charges'][medical['insuranceclaim'] == 1]

s0 = np.std(sample0)
s1 = np.std(sample1)

n0 = len(sample0)
n1 = len(sample1)
print("n0: " + str(n0) + " , n1: " + str(n1))

x0 = sample0.mean()
x1 = sample1.mean()
print("x0: " + str(x0) + " , x1: " + str(x1))

sp = np.sqrt(((n0 - 1) * (s0 ** 2) + (n1 - 1) * (s1 ** 2)) / (n0 + n1 - 2))
print("sp: " + str(sp))

ttest = (x0 - x1) / (sp * np.sqrt((1 / n0) + (1 / n1)))
print("t: " + str(ttest))

tprob = (1 - t.cdf(abs(ttest), n0 + n1 - 2)) * 2
print(tprob)
```

```
n0: 555 , n1: 783
x0: 8821.421892306294 , x1: 16423.928276537663
sp: 11512.282899205744
t: -11.901306943555385
0.0
```

In [57]:

```
from scipy import stats

stats.ttest_ind(sample0, sample1)
```

Out[57]:

```
Ttest_indResult(statistic=-11.893299030876712, pvalue=4.461230231620717e-
31)
```

Both results are basically the same! Our conclusion is that the null hypothesis is to be rejected and that patients do in fact get charged differently based on their insurance status.

In [ ]:

```

```

Congratulations! Hopefully you got the exact same numerical results. This shows that you correctly calculated the numbers by hand. Secondly, you used the correct function and saw that it's much easier to use. All you need to do is pass your data to it.

Q6: Conceptual question: look through the documentation for statistical test functions in scipy.stats. You'll see the above *t*-test for a sample, but can you see an equivalent one for performing a *z*-test from a sample? Comment on your answer.

A: There isn't a function that performs a z-test from a sample. Perhaps this is because the population variance is usually never known, and therefore we'd usually default to a t test anyways.

Learning outcomes

Having completed this project notebook, you now have good hands-on experience:

- using the central limit theorem to help you apply frequentist techniques to answer questions that pertain to very non-normally distributed data from the real world
- performing inference using such data to answer business questions
- forming a hypothesis and framing the null and alternative hypotheses
- testing this using a *t*-test