Cmd 1

Cmd 2

Hypothesis Testing (25 mins)

Objective: *Perform a two-sample, two-sided t-test to compare sample means.*

In this lab, you will complete a series of exercises to perform a t-test.

We want to determine whether the population mean of daily steps taken for the athlete health tracker users is equal to the population mean of daily steps taken for the cardio enthusiast health tracker users.

The null and alternative hypotheses for this test are:

- H_0 = the mean of daily steps taken for athlete users is equal to the mean of daily steps taken for cardio enthusiast users.
- H_1 = the mean of daily steps taken for athlete users is not equal to the mean of daily steps taken for cardio enthusiast users.

Cmd 3

```
%run "../../Includes/Classroom-Setup"
```

Cmd 4

Exercise 1

Compute the sample mean of daily steps taken for sedentary users and cardio enthusiast users.

Cmd 5

```
%sql
-- ANSWER
SELECT lifestyle, avg(steps) AS mean
FROM dsfda.ht_daily_metrics
WHERE lifestyle = "Athlete" OR lifestyle = "Cardio Enthusiast"
GROUP BY lifestyle
```

Cmd 6

Exercise 2

Compute the sample variance of daily steps taken for sedentary users and cardio enthusiast users.

Cmd 7

```
%sql
-- ANSWER
SELECT lifestyle, var_samp(steps) AS variance
FROM dsfda.ht_daily_metrics
WHERE lifestyle = "Athlete" OR lifestyle = "Cardio Enthusiast"
GROUP BY lifestyle
```

Cmd 8

Exercise 3

Compute the sample size for sedentary users and cardio enthusiast users.

Cmd 9

```
%sql
-- ANSWER
SELECT lifestyle, count(*) AS sample_size
FROM dsfda.ht_daily_metrics
WHERE lifestyle = "Athlete" OR lifestyle = "Cardio Enthusiast"
GROUP BY lifestyle
```

Cmd 10

Exercise 4

Compute the T-statistic using the sample statistics.

Cmd 11

```
# ANSWER
from math import sqrt

athlete_mean = 11001.597413366928
athlete_variance = 9623550.2344004
athlete_size = 313535

cardio_mean = 13235.376990421259
cardio_variance = 13171492.338618632
cardio_size = 388360

```

```
test_statistic = (athlete_mean - cardio_mean) / sqrt((athlete_variance / athlete_size) + (cardio_variance / cardio_size))
print(f"T-statistic = {test_statistic}")
```

Cmd 12

Exercise 5

Compute the degrees of freedom using the sample statistics.

Cmd 13

```
# ANSWER
df_numerator = ((athlete_variance / athlete_size) + (cardio_variance / cardio_size))**2
df_denominator = (athlete_variance / athlete_size)**2 / (athlete_size - 1) + ((cardio_variance / cardio_size)**2 / (cardio_size - 1))
df = df_numerator / df_denominator
print(f"Degrees-of-freedom = {df}")
```

Cmd 14

Exercise 6

Compute the p-value for this T-test by passing in the `test_statistic` and the `df` to `t.cdf()`.

Cmd 15

```
# ANSWER
from scipy.stats import t
p_value = t.cdf(test_statistic, df)
print(f"p-value = {p_value}")
```

Cmd 16

Exercise 7

Determine whether we should reject the null hypothesis.

 Use a significance level of 0.05.

Cmd 17

```
# ANSWER
print(f"The p-value {p_value} is less than 0.05. Thus, we reject the null hypothesis.")
```

Cmd 18

Exercise 8

Phew! That was a lot of work to answer a simple question.

Luckily, Python's `scipy` module makes this process a bit easier than it already has.

Check out the demonstration below showing how to perform this same test in a single step only using Python.

Cmd 19

```
from scipy.stats import ttest_ind

athlete_daily_steps = spark.sql("SELECT steps FROM dsfda.ht_daily_metrics WHERE lifestyle = 'Athlete'").toPandas()["steps"]
cardio_daily_steps = spark.sql("SELECT steps FROM dsfda.ht_daily_metrics WHERE lifestyle = 'Cardio Enthusiast'").toPandas()["steps"]

ttest_ind(athlete_daily_steps, cardio_daily_steps, equal_var = False)
```

Cmd 20

Notice that the *same test statistic* and *same p-value* were calculated with far less code!

While it's good to understand how these `scipy` tools work, it's good practice and efficient to use them as much as possible.

Cmd 21

Shift+Enter to run