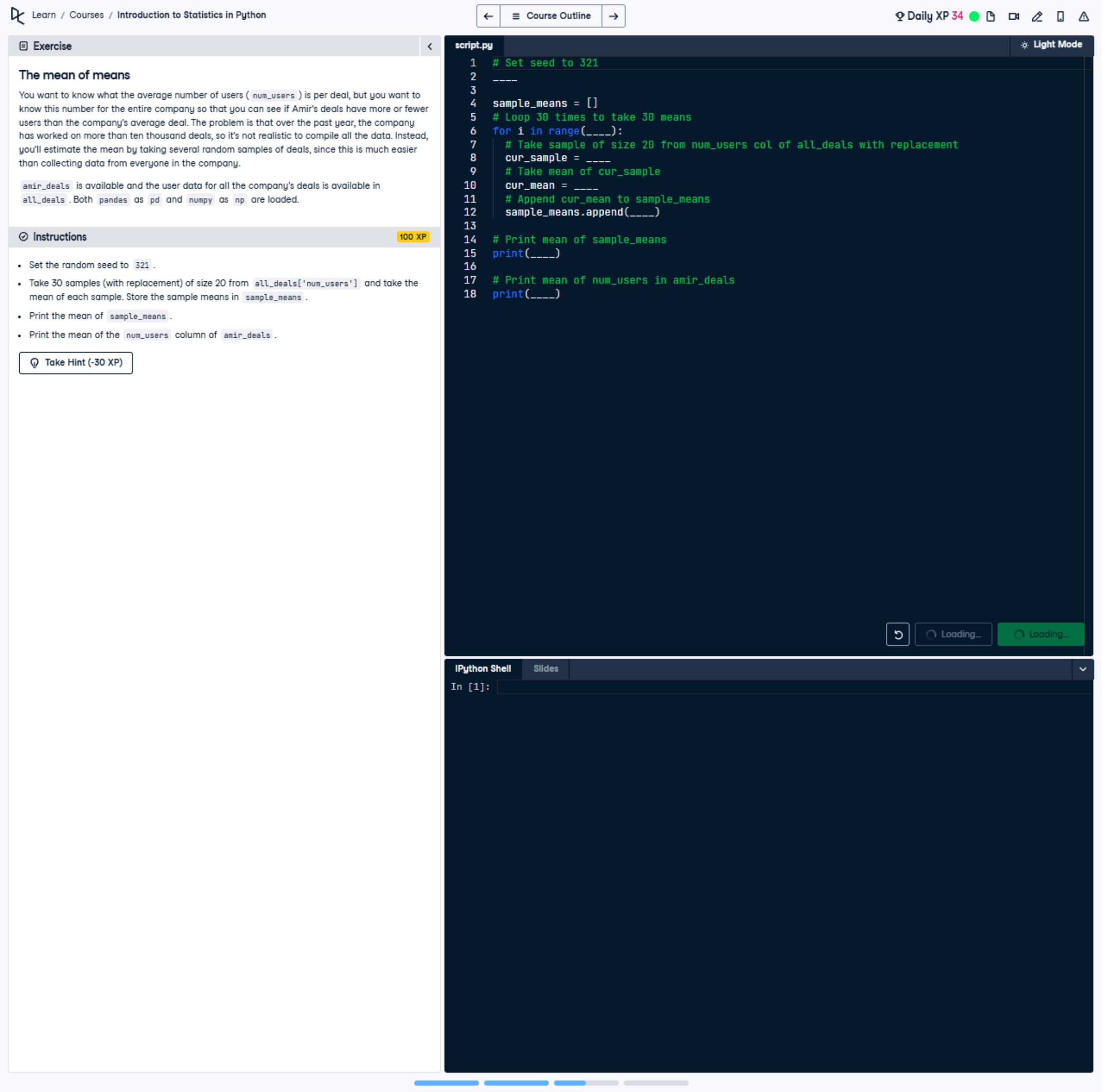
The Mean of Means



The mean of means

You want to know what the average number of users (`num_users`) is per deal, but you want to know this number for the entire company so that you can see if Amir's deals have more or fewer users than the company's average deal. The problem is that over the past year, the company has worked on more than ten thousand deals, so it's not realistic to compile all the data. Instead, you'll estimate the mean by taking several random samples of deals, since this is much easier than collecting data from everyone in the company.

`amir_deals` is available and the user data for all the company's deals is available in `all_deals` . Both `pandas` as `pd` and `numpy` as `np` are loaded.

- Set the random seed to `321` .
- Take 30 samples (with replacement) of size 20 from `all_deals['num_users']` and take the mean of each sample. Store the sample means in `sample_means` .
- Print the mean of `sample_means` .
- Print the mean of the `num_users` column of `amir_deals` .

Take Hint (-30 XP)

```python
# Set seed to 321
____

sample_means = []
# Loop 30 times to take 30 means
for i in range(____):
  # Take sample of size 20 from num_users col of all_deals with replacement
  cur_sample = ____
  # Take mean of cur_sample
  cur_mean = ____
  # Append cur_mean to sample_means
  sample_means.append(____)

# Print mean of sample_means
print(____)

# Print mean of num_users in amir_deals
print(____)
```

```
In [1]:
```

Question:

You want to calculate the average number of users ('num_users') per deal for Amir’s entire company dataset. Instead of calculating the mean for all deals, you repeatedly sample and calculate the mean, then compare with the overall mean.

Explanation of the Question:

This exercise focuses on approximating the mean of a dataset by taking repeated random samples of size 20. The sample means are calculated and printed alongside the overall mean of the dataset to show the accuracy of sampling.

Answer:

```
# Import necessary libraries
import numpy as np
import pandas as pd

# Set seed to 321 for reproducibility
np.random.seed(321)

# Initialize an empty list to store sample means
sample_means = []

# Loop 30 times to take 30 means
for i in range(30):
    # Take a sample of size 20 from num_users column of all_deals with replacement
    cur_sample = all_deals['num_users'].sample(20, replace=True)
    # Take mean of the current sample
    cur_mean = np.mean(cur_sample)
    # Append the mean to sample_means
    sample_means.append(cur_mean)

# Print the mean of sample_means
print(np.mean(sample_means))

# Print the mean of the num_users column in amir_deals
print(np.mean(amir_deals['num_users']))
```

Explanation of the Answer:

This corrected solution involves setting a random seed for reproducibility, then taking 30 samples, each of size 20, from the 'num_users' column of the 'all_deals' dataset. The mean of each sample is calculated and stored in a list. Finally, the mean of the sample means is compared to the mean of the 'num_users' column in 'amir_deals' to validate the accuracy of the sampling approach.