Quartiles, Quintiles, and Deciles - Updated

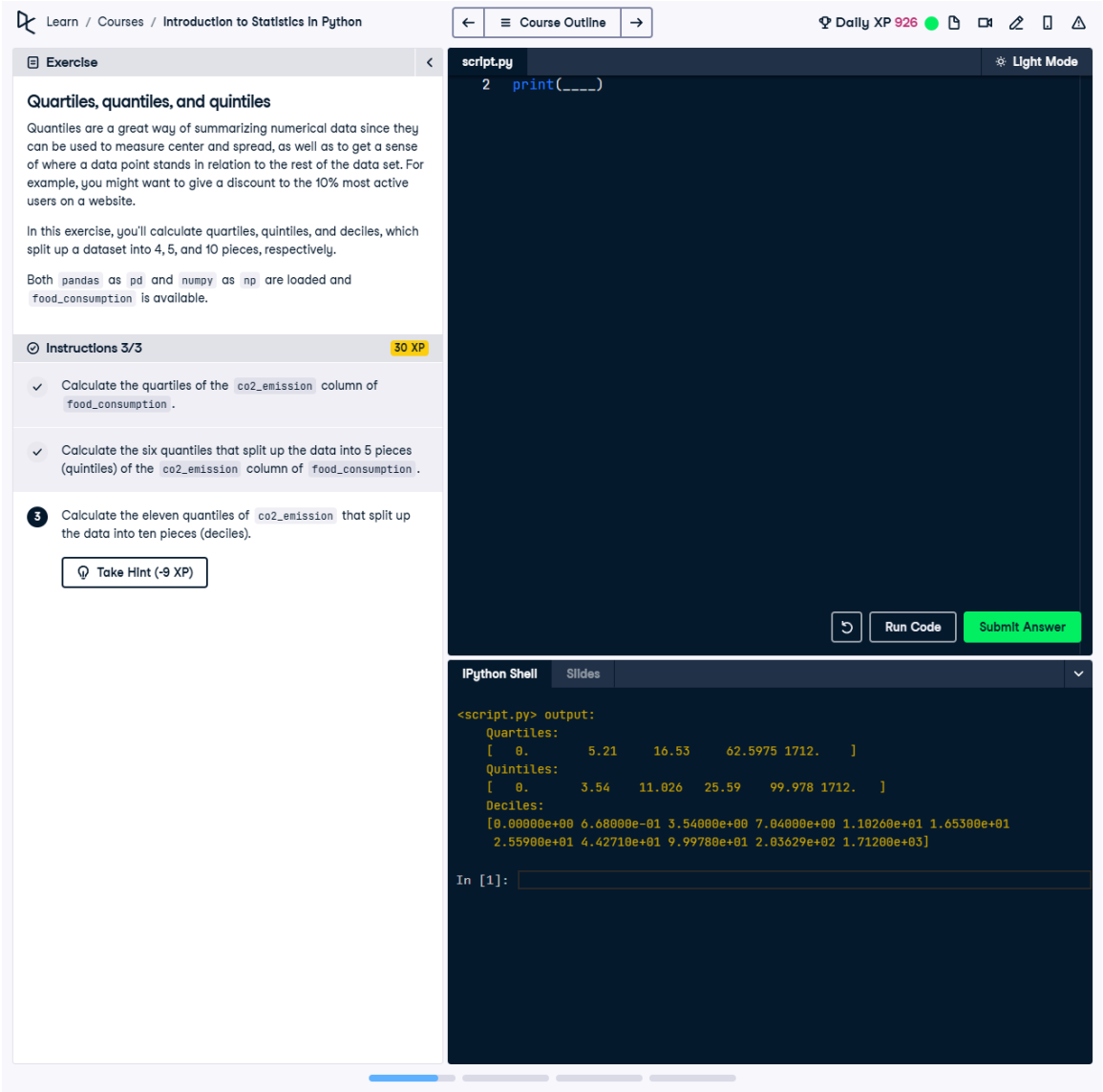

Figure 1: Screenshot showing the task to calculate quartiles, quintiles, and deciles for CO2 emissions with the correct output.

Question

Quantiles are a great way of summarizing numerical data since they can be used to measure center and spread, as well as to get a sense of where a data point stands in relation to the rest of the dataset. For example, you might want to give a discount to the 10% most active users on a website.

In this exercise, you'll calculate quartiles, quintiles, and deciles, which split

up a dataset into 4, 5, and 10 pieces, respectively.

Instructions:
1. Calculate the quartiles of the `co2_emission` column of `food_consumption`.
2. Calculate the six quantiles that split up the data into 5 pieces (quintiles) of the `co2_emission` column of `food_consumption`.
3. Calculate the eleven quantiles of `co2_emission` that split up the data into ten pieces (deciles).

Corrected Code Solution

```
import numpy as np

# Calculate quartiles (4 quantiles)
quartiles = np.quantile(food_consumption['co2_emission'], [0, 0.25, 0.5, 0.75, 1])
print("Quartiles:")
print(quartiles)

# Calculate quintiles (5 quantiles)
quintiles = np.quantile(food_consumption['co2_emission'], [0, 0.2, 0.4, 0.6, 0.8, 1])
print("Quintiles:")
print(quintiles)

# Calculate deciles (10 quantiles)
deciles = np.quantile(food_consumption['co2_emission'], np.linspace(0, 1, 11))
print("Deciles:")
print(deciles)
```

Answer Explanation

1. **Quartiles:** The quartiles divide the data into 4 equal parts:
   - `0%`: The minimum value.
   - `25%`: The first quartile (Q1).
   - `50%`: The median (Q2).
   - `75%`: The third quartile (Q3).
   - `100%`: The maximum value.

2. **Quintiles:** The quintiles divide the data into 5 equal parts, adding

more granularity compared to quartiles. These are calculated at 0%, 20%, 40%, 60%, 80%, and 100%.

3. **Deciles:** The deciles divide the data into 10 equal parts, offering a finer breakdown of the distribution. These are calculated at equal intervals between 0% and 100%.

4. **Using np.quantile():** The `np.quantile()` function calculates quantiles for the specified probabilities. This method is flexible for generating any number of quantiles and provides a comprehensive view of the data distribution.