1 newpage

2 Data Science: Bridging Principles and Practice

2.1 Part 2: Introduction to Python [SOLUTIONS]

2.2 The Data: Rocket Fuel Ad Campaign

QUESTION: Take another look at the data in the `ads` DataFrame. Which columns have numerical data? Which columns have string data? Which column has Boolean data?

ANSWER:

- "user id": numerical
- "test group": string
- "converted": Boolean
- "total ads": numerical
- "most ads day": string
- "most ads hour": numerical

3 2. Python

3.1 2a. Expressions

EXERCISE: Scroll back up to the cell that generated an error. Fix the error, and re-run the cell to check that the error has been fixed.

```
In [1]: # add a closing parethesis at the end of the line
        print("This line is missing something.")
```

```
This line is missing something.
```

3.2 2b. Names

EXERCISE: Before Rocket Fuel can evaluate the effectiveness of the ad campaign, they need to know how much it cost.

The *total number of advertisements* was14597182. The *CPM* (cost per thousand ads) was$9. Use these numbers to assign the correct values to `total_ads`, `cpm`, and `cost_per_ad`.

Note: for the third variable, we want the cost *for each ad*. What do we need to do to the CPM to get the per-ad cost?

```
In [3]: # replace the ... with the total number of ads
        total_ads = 14597182

        # replace the ... with the cost per thousand ads
        cpm = 9

        # replace the ... with an expression to calculate the cost per ad
        cost_per_ad = cpm / 1000
        cost_per_ad
```

```
Out[3]: 0.009
```

EXERCISE: Then, calculate the overall cost by multiplying the number of ads by how much each ad cost. Assign this value to the name `cost`.

Hint: you can do the calculation by using only using `total_ads`, `cost_per_ad`, and the `*` multiplication operator- no numbers needed. Your answer should be a six-digit number (before the decimal).

```
In [4]: # replace the ... with an expression to calculate the cost of the ad campaign
        cost = total_ads * cost_per_ad
        cost

Out[4]: 131374.63799999998
```

3.3 2c. Functions

PRACTICE:

Try calling `abs` and `max` in the cell below. What does each function do?

Also try calling each function *incorrectly*, such as with the wrong number of arguments. What kinds of error messages do you see?

```
In [5]: # abs gets the absolute value of a number
        abs(-5300)

Out[5]: 5300

In [6]: # max returns the maximum of a sequence of numbers
        max(4, 700, -1.88, 298, 699)

Out[6]: 700
```

3.3.1 Dot Notation

PRACTICE: `math` also has a function called `sqrt` that takes one argument and returns the square root. Call `math.sqrt` on 16 in the next cell.

```
In [7]: # use math.sqrt to get the square root of 16
        root_16 = math.sqrt(16)

        # show the result of the expression
        root_16

Out[7]: 4.0
```

3.4 2d. Sequences

PRACTICE: Try indexing the second-to-last item from the `prices` array. Save it to the name `next_to_last_price`.

```
In [8]: # index the second-to-last item in the list
        next_to_last_price = prices[3]

        # show the second-to-last item in the array
        next_to_last_price

Out[8]: 130.0
```

EXERCISE: Use the `prices` and `tax_rates` arrays to try some operations.

```
In [9]: #  use an element-wise operation to add 10 to all the prices
        price_plus_10 = prices + 10
        price_plus_10

Out[9]: array([115.99, 109.99, 129.95, 140.  , 134.99])
```

```
In [10]: # use an element-wise operation to divide all the taxes values by 2
         taxes_reduced_by_half = tax_rates / 2
         taxes_reduced_by_half

Out[10]: array([0.0475, 0.055 , 0.0435, 0.05  , 0.042 ])

In [11]: #  use the np.max reduction function to get the maximum value from the prices arra
         max_price = np.max(prices)
         max_price

Out[11]: 130.0
```