Readiness Quiz Review

Quiz Review

I hope you found the readiness quiz challenging, but not too difficult. Let's take a few minutes to review the quiz to ensure you have the necessary resources to help you with any questions you missed. We are not going to just give the answers in this review, but rather talking points to help you understand what you should have done. Remember, your favorite search engine is your friend!

Question 1: Complete the following Python function. For a given input list:

1. Cube (element to power 3) each element
2. Return the results as a list

Answer: Iterate through the list of input numbers and return x^3 (written in Python code of course).

For example: 2 cubed = 8, 3 cubed = 27 ....

Resources: https://docs.python.org/3/library/stdtypes.html

Question 2: Complete the following Python function. For a given input list:

1. Inspect each number in the input list and determine if it is even
2. Next square the even values
3. Finally return the results as a list

Answer: Iterate through the list of input numbers and return x^2 only for the even numbers in the list.

For example: given a list [2,3,4], look at the 2 and square it (because it's even)= 4, ignore the 3 (it's odd), 4 (is even) squared is 16.

Resources: https://docs.python.org/3/library/stdtypes.html

Question 3: Which of the following is not an example of a native or built-in data type in Python

(a) boolean (b) integer (c) float (d) heap (e) string (f) varchar

Answer: Look at the resources link to learn about the built-in data types in Python

Resources: https://docs.python.org/3/library/stdtypes.html

Question 4: Complete the following Python function. Create a dictionary from two input lists

Answer: Returns an iterator of tuples, where the i-th tuple contains the i-th element from each of the argument sequences or iterables. The iterator stops when the shortest input iterable is exhausted. With a single iterable argument, it returns an iterator of 1-tuples. With no arguments, it returns an empty iterator.

For example: This function involves a step where elements from separate data structures are paired up. A convenient way to accomplish this pairing is to use Python's zip function. For example, iterating through the generator returned by using zip with ['a','b','c'] and [1,2,3] yields the tuples ('a', 1), ('b', 2), ('c', 3).

Resources: https://docs.python.org/3/library/functions.html#zip

---

Question 5: Mutable data types/collections in Python can be changed in place. Immutable ones can not change in place. Which of the following are mutable?

(a) bool (b) int (c) float (d) set (e) list (f) string (g) tuple (h) complex

Answer: Look at the resources link to learn about the built-in data types in Python

Resources: https://docs.python.org/3/library/stdtypes.html

Question 6: Python is a general-purpose language, but which (1 or more) of the following ideas is not realistic with Python?

Python makes it easy to:

(a) interactively work to test and understand algorithms, (b) to prototype and explore data using Ipython and Jupyter notebooks, (c) carry out unit testing, (d) include comments/pseudocode to better organize code , (e) save files with an editor then subsequently execute them from the command line, (f) naturally parallelizes across cores and machines with little to no overhead, (g) save multiple functions in a file then import those functions from a different file

Answer: All of these are core elements of Python's functionality, except for parallelizing across machines. While it is possible to use Python in such an environment, most solutions using it in conjunction with other technologies, such as using the PySpark framework with Apache Spark.

Resources: https://docs.python.org/3/

Question 7: Given a predefined character list. Complete the following Python function. Return a dictionary of character counts * Count the of the number of times each character appears in the string * Characters with a count of 0 should not be included in the output dictionary

Answer: While it is relatively straightforward to write a function that loops through a list and keeps track of how many times an element has been seen, there are tools in Python's standard library that do this work for you. The Counter object, which is part of the collections module is designed for this purpose. It and other container datatypes in that module can be very helpful when scanning/manipulating data.

For example: If I gave you 'aaaabbccccc' you would iterate through the list and count there is (4) of the letter 'a', (2) of the letter 'b', and (5) of the letter 'c'.

Resources: https://docs.python.org/3/library/collections.html

---

Question 8: For the vector v = [2.0, -3.5, 5.1]:

1. Find the L1 norm of v
2. Return the result as a float

Answer: Utilizing the NumPy package, I would evaluate the given array by iterating through each element's absolute value and calculating the sum.

For example: I would evaluate the first position v[0] and see there is a 2.0, then add it to the absolute value of position v[1] which is 3.5. 2.0+3.5 = 5.5. Finally, looking at position v[2] I would add the last value of 5.1 to get my final answer.

Resources: https://docs.scipy.org/doc/numpy/reference/generated/numpy.array.html , https://docs.scipy.org/doc/numpy/reference/generated/numpy.absolute.html

---

Question 9: NumPy array practice

1. Create a vector that starts at 1 and increases until 150
2. Turn the vector into a matrix with 10, rows and 15 columns
3. Print the sum for the 10 rows. HINT: there should be ten values for the printed sum

Answer: Numpy's *arange* method can be used to generate the data vector. To turn the vector into a matrix, use *reshape* and then apply the *sum* method (which takes parameters to modify the default behavior of just returning a single value).

Resources: https://docs.scipy.org/doc/numpy/reference/generated/numpy.array.html , https://docs.scipy.org/doc/numpy/reference/generated/numpy.arange.html , https://docs.scipy.org/doc/numpy/reference/generated/numpy.reshape.html , https://docs.scipy.org/doc/numpy-1.13.0/user/basics.creation.html

---

Question 10: Which of the following pairs of events are mutually exclusive. There can be more than one answer.

(A) Odd numbers and the number 3, (B) Even numbers and numbers greater than 10, (C) Negative numbers and positive numbers less than 25, (D) Numbers between 100-200 and numbers between 201-300, (E) None of the above

Answer: Mutual exclusivity pertains to situations where only one condition can be true at a time. For example, one condition might be: A person born in the year 2000, and a second condition: A person born in the month of June. These conditions are NOT mutually exclusive because it is possible to have been born in June of 2000.

Resources: https://en.wikipedia.org/wiki/Mutual_exclusivity

Question 11: Geometric distribution

The geometric distribution is a useful tool for modeling time to event data A successful street vendor says that on average 1 out of every 10 people who walk by on the street stop to buy a taco.

1. Represent these data with a geometric distribution
2. What is the probability that the vendor has to wait until 20 people walk buy before someone buys a taco?

Answer: A Geometric distribution has one parameter: the probability of success in a single trial, and gives the probability that the first occurrence of success requires k independent trials. You can plug in the probability value into the Scipy Stats geom object, and then apply the appropriate probability function for a first success after k trials.

Resources: https://en.wikipedia.org/wiki/Geometric_distribution , https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.geom.html , https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html

---

Question 12: Poisson distribution

The Poisson distribution is a useful tool for modeling count data given discrete intervals. Based on historical data the expected number of accidents at a busy intersection is 4 per month.

1. Represent these data with a Poisson distribution
2. What is the probability of more than 7 accidents at that intersection next month?

Answer: Similar to the question on the Geometric distribution, this question on the Poisson distribution uses Scipy Stats to calculate a probability. The Poisson distribution's parameter is a rate, sometimes described as "events per unit time". You need to specify this parameter and then compare that distribution with some hypothetical rate to calculate a probability. Note that when questions of probability involve ranges of values on one side of a cutoff (i.e. "more than 7 accidents") then it can be helpful to use a Cumulative Distribution Function in your calculations.

Resources: https://en.wikipedia.org/wiki/Poisson_distribution , https://en.wikipedia.org/wiki/Cumulative_distribution_function , https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.poisson.html , https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html

---

Question 13: Gaussian distribution

The Gaussian or Normal distribution is use heavily thoughout statistics and data science. Lets assume scores for this assessment have a mean of 50% and a standard deviation of 10.

1. Represent these data with a Normal distribution
2. What is the probability of observing a score >= 80?

Answer: This question builds on the tools used in the previous ones on the Geometric and Poisson distributions, this time with the Normal distribution. It will again be helpful to make calculations using the Cumulative Distribution Function.

Resources:https://en.wikipedia.org/wiki/Normal_distribution , https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html , https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html

---

Question 14: Which statement(s) about Pearson correlation and cosine similarity are true?

(a) The dot product of two sample vectors divided by the product of their norms yields the correlation coefficient, (b) The cosine similarity of two centered vectors yields the correlation coefficient, (c) The product of two sample vector norms that have been centered yields the correlation coefficient, (d) The dot products of centered vectors divided by the product of their norms yields the correlation coefficient, (e) Two normalized sample vectors divided by centered dot products yield the correlation coefficient, (f) The cosine similarity of the dot product of two sample vectors is the correlation coefficient

Answer: Compare the geometric interpretation of Pearson correlation, with the definition of Cosine similarity.

Resources:https://en.wikipedia.org/wiki/Pearson_correlation_coefficient#Geometric_interpretation, https://en.wikipedia.org/wiki/Cosine_similarity#Definition

---

Question 15: Perform matrix multiplication on a square matrix HINT: A 2X2 matrix times a 2x2 matrix should yield a 2x2 matrix

Answer: Recall in matrix multiplication, rows from the left matrix are multiplied by columns of the right matrix and these products are summed. If matrices are represented as lists of lists in pure Python (not using Numpy) it is straightforward to access the elements of each row of the left matrix, but trickier to access each element that makes

up a column of the right matrix. One option involves looping through each inner list and extracting the elements at a particular index. This effectively builds a new list for each column in the right matrix, which can then be combined with the row values from the left matrix.

Resources: https://docs.python.org/3/tutorial/datastructures.html

How did you do?

If you struggled with some of these answers, please use the resources provided to build your knowledge. Remember, Google (or whatever search engine you choose) is your friend. Otherwise, let's get started!