**Array-Oriented Programming with NumPy**

* Introduction

7.1 Q1: Which of the following statements is false?

a. The NumPy (Numerical Python) library is the preferred Python array implementation—it offers a high-performance, richly functional n-dimensional array type called ndarray, which you can refer to by its synonym, array.

b. Operations on arrays are up to two orders of magnitude faster than those on lists.

c. Many popular data science libraries such as Pandas, SciPy (Scientific Python) and Keras (for deep learning) are built on or depend on NumPy.

d. A strength of NumPy is “array-oriented programming,” which uses functional-style programming with external iteration to make array manipulations concise and straightforward, eliminating the kinds of bugs that can occur with the internal iteration of explicitly programmed loops.

Answer: d. A strength of NumPy is “array-oriented programming,” which uses functional-style programming with external iteration to make array manipulations concise and straightforward, eliminating the kinds of bugs that can occur with the internal iteration of explicitly programmed loops. Actually, a strength of NumPy is “array-oriented programming,” which uses functional-style programming with internal iteration to make array manipulations concise and straightforward, eliminating the kinds of bugs that can occur with the external iteration of explicitly programmed loops.

* Creating **array**s from Existing Data

7.2 Q1: The NumPy array function receives as an argument an array or other collection of elements and returns a new array containing the argument’s elements. Based on the statement:

numbers = np.array([2, 3, 5, 7, 11])

what type will be output by the following statement?

type(numbers)

a. array

b. ndarray

c. numpy.ndarray

d. numpy

Answer: c. numpy.ndarray. Note that the *type* is numpy.ndarray, but the string representation of an array uses “array.”

7.2 Q1: Which of the following statements is false?

a. The array function copies its argument’s dimensions.

b. The following creates an array from a two-row-by-three-column list:

np.array([[1, 2, 3], [4, 5, 6]])

c. NumPy auto-formats arrays, based on their number of dimensions, aligning the columns within each row.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* **array** Attributes

For the questions in this section, assume the following array definitions:

import numpy as np  
  
integers = np.array([[1, 2, 3], [4, 5, 6]])  
  
floats = np.array([0.0, 0.1, 0.2, 0.3, 0.4])

7.3 Q1: Which of the following statements is false?

a. The array function determines an array’s element type from its argument’s elements.

b. You can check the element type with an array’s dtype attribute.

c. For performance reasons, NumPy is written in the Java programming language and uses Java’s data types.

d. All of the above statements are true.

Answer: c. For performance reasons, NumPy is written in the Java programming language and uses Java’s data types. Actually, for performance reasons, NumPy is written in the C programming language and uses C’s data types.

7.3 Q2: The attribute          contains an array’s number of dimensions and the attribute          contains a          specifying an array’s dimensions:

a. dim, size, list

b. ndim, shape, tuple

c. dim, size, tuple

d. ndim, shape, list

Answer: b. ndim, shape, tuple

7.3 Q3: Which of the following statements is false?

a. You can view an array’s total number of elements with the attribute size and the number of bytes required to store each element with itemsize.

b. The array integers’ size is the product of the shape tuple’s values—two rows of three elements each for a total of six elements.

c. In each case, itemsize is 8 because integers contains int64 values and floats contains float64 values, which each occupy 8 bytes.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.3 Q4: Which of the following statements is false?

a. You’ll generally manipulate arrays using concise functional-style programming techniques with internal iteration.

b. Because arrays are iterable, you cannot use external iteration with them.

c. You can iterate through a multidimensional array as if it were one-dimensional by using its flat attribute.

d. All of the above statements are true.

Answer: b. Because arrays are iterable, you cannot use external iteration with them. Actually, because arrays are iterable, you *can* use external iteration with them.

* Filling **array**s with Specific Values

7.4 Q1: Which of the following statements is false?

a. NumPy provides functions zeros, ones and full for creating arrays containing 0s, 1s or a specified value, respectively.

b. The first argument to the functions in Part (a) must be an integer or a tuple of integers specifying the desired dimensions. For an integer, each function returns a one-dimensional array with the specified number of elements. For a tuple of integers, these functions return a multidimensional array with the specified dimensions.

c. The array returned by NumPy function full contains elements with the second argument’s value and type.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* Creating **array**s from Ranges

7.5 Q1: Which of the following statements about NumPy’s linspace function is false?

a. You can produce evenly spaced floating-point ranges with linspace.

b. The function’s first two arguments specify the starting and ending values in the range, and the ending value is included in the array.

c. The optional keyword argument num specifies the number of evenly spaced values to produce—this argument’s default value is 50.

d. All of the above statements are true.

Answer: d. All of the above statements are true.

7.5 Q2: Which of the following statements is false?

a. You can create an array from a range of elements, then use array method reshape to transform the one-dimensional array into a multidimensional array.

b. The following code creates an array containing the values from 1 through 20, then reshapes it into four rows by five columns:

np.arange(1, 21).reshape(4, 5)

c. A 24-element one-dimensional array can be reshaped into a 2-by-12, 8-by-3 or 4-by-8 array, and vice versa.

d. All of the above are *true*.

Answer: c. A 24-element one-dimensional array can be reshaped into a 2-by-12, 8-by-3 or 4-by-8 array, and vice versa. Actually, a 24-element one-dimensional array cannot be reshaped into a 4-by-8 array and vice versa. The target array must have the same number of elements as the array being reshaped.

7.5 Q3: Which of the following statements is true with respect to displaying an array of 1000 items or more?

a. NumPy always drops the middle rows and middle columns from the output.

b. NumPy always drops the middle rows from the output.

c. NumPy always drops the middle columns from the output.

d. NumPy drops the middle rows, middle columns or both from the output.

Answer: d. NumPy drops the middle rows, middle columns or both from the output.

* List vs. **array** Performance: Introducing **%timeit**

7.6 Q1: Which of the following statements is false?

a. Most array operations execute significantly faster than corresponding list operations.

b. With the IPython %timeit magic command, you can time the average duration of operations.

c. The times displayed on one system may vary from those shown on another.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.6 Q2: Which of the following statements is false?

a. The following code uses the random module’s randrange function with a list comprehension to create a list of six million die rolls and time the operation using %timeit:

import random  
  
%timeit rolls\_list = \  
 [random.randrange(1, 7) for i in range(0, 6\_000\_000)]

b. By default, %timeit executes a statement in a loop, and it runs the loop seven times.

c. After executing the statement, %timeit displays the statement’s average execution time, as well as the standard deviation of all the executions.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* **array** Operators

7.7 Q1: Which of the following statements is false?

a. Element-wise operations are applied to every array element, so given an integer array named numbers, the expression

numbers \* 2

multiplies every element by 2, and the expression

numbers \*\* 3

cubes every element.

b. The expressions in Part (a) do not modify the array numbers.

c. Operators + and \* are commutative, so numbers \* 2 could also be written as 2 \* numbers.

d. Augmented assignments modify every element in the right operand.

Answer: d. Augmented assignments *modify* every element in the right operand. Actually, augmented assignments modify every element in the *left* operand.

7.7 Q2: Which of the following statements is false?

a. Normally, the arithmetic operations require as operands two arrays of the same size and shape.

b. When one operand is a single value, called a scalar, NumPy performs the element-wise calculations as if the scalar were an array of the same shape as the other operand, but with that scalar value in all its elements.

c. If numbers is a five-element integer array, numbers \* 2 is equivalent to:

numbers \* [2, 2, 2, 2, 2]

d. Broadcasting can only be applied between arrays of the same size and shape, enabling some concise and powerful manipulations.

Answer: d. Broadcasting can only be applied between arrays of the same size and shape. Actually, broadcasting can be applied between arrays of different, but compatible, sizes and shapes, enabling some concise and powerful manipulations.

7.7 Q3: Which of the following statements is false?

a. You may perform arithmetic operations and augmented assignments between arrays of the same shape.

b. Arithmetic between arrays of integers and floating-point numbers results in an array of integers.

c. You can compare arrays with individual values and with other arrays. Comparisons are performed element-wise. Such comparisons produce arrays of Boolean values in which each element’s True or False value indicates the comparison result.

d. The expression numbers >= 13 uses broadcasting to determine whether each element of numbers is greater than or equal to 13.

Answer: b. Arithmetic between arrays of integers and floating-point numbers results in an array of integers. Actually, arithmetic between arrays of integers and floating-point numbers results in an array of *floating-point numbers*.

* NumPy Calculation Methods

7.8 Q1: Which of the following statements is false?

a. Calculating the mean of an array totals all of its elements regardless of its shape, then divides by the total number of elements.

b. You can perform array calculations on each array dimension as well. For example, in a two-dimensional array, you can calculate each row’s mean or each column’s mean.

c. The array methods sum, min, max, mean, std (standard deviation) and var (variance) are each functional-style programming reductions.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*

7.8 Q2: Which of the following statements is false?

a. Many calculation methods can be performed on specific array dimensions, known as the array’s axes. These methods receive an axis keyword argument that specifies which dimension to use in the calculation, giving you a quick way to perform calculations by row or column in a two-dimensional array.

b. Assume that you want to calculate the average grade on each exam, represented by the columns of grades. Specifying axis=0 performs the calculation on all the row values within each column. Similarly, specifying axis=1 performs the calculation on all the column values within each individual row.

c. NumPy does not display trailing 0s to the right of the decimal point. Also, it does not display all element values in the same field width.

d. For a two-dimensional array grades in which each row represents one student’s grades on several exams, we can calculate each student’s average grade with:

grades.mean(axis=1)

Answer: c. NumPy does not display trailing 0s to the right of the decimal point. Also, it does not display all element values in the same field width. Actually, NumPy does display all element values in the same field width.

* Universal Functions

7.9 Q1: Which of the following statements is false?

a. NumPy offers dozens of standalone universal functions (or ufuncs) that perform various element-wise operations.

b. Each ufunc must perform its task using one or two arrays. Some of these functions are called when you use operators like + and \* on arrays. Each returns a new array containing the results.

c. The following snippet calculates the square root of the array numbers’ values, using the sqrt universal function:

np.sqrt(numbers)

d. The following snippet adds two arrays with the same shape, using the add universal function:

np.add(numbers, numbers2)

d. The following snippet adds two arrays with the same shape, using the add universal function:

numbers + numbers2

Answer: b. Each ufunc must perform its task using one or two arrays. Some of these functions are called when you use operators like + and \* on arrays. Each returns a new array containing the results. Actually, each ufunc performs its task using one or two array *or array-like arguments (such as lists)*.

7.9 Q2: Assume the array numbers contains the values 10, 20, 30, 40, 50, 60. Which of the following statements is false?

a. The following code uses the multiply universal function to multiply every element of numbers by the scalar value 5:

np.multiply(numbers, 5)

a. The following code uses the multiply universal function to multiply every element of numbers by the scalar value 5:

numbers \* 5

c. The following code reshapes numbers into a 2-by-3 array, then multiplies its values by a one-dimensional array of three elements:

numbers3 = numbers2.reshape(2, 3)  
numbers4 = np.array([2, 4, 6])  
np.multiply(numbers3, numbers4)

The preceding code works because numbers4 has the length as each column of numbers3.

d. If a universal function receives two arrays of different shapes that do not support broadcasting, a ValueError occurs.

Answer: c. The following code reshapes numbers into a 2-by-3 array, then multiplies its values by a one-dimensional array of three elements:

numbers3 = numbers2.reshape(2, 3)  
numbers4 = np.array([2, 4, 6])  
np.multiply(numbers3, numbers4)

The preceding code works because numbers4 has the same length as each column of numbers3. Actually, the preceding code works because numbers4 has the **same length as each row** of numbers3.

* Indexing and Slicing

7.10 Q1: Assuming the following array grades:

grades = np.array([[87, 96, 70], [100, 87, 90],  
 [94, 77, 90], [100, 81, 82]])

which of the following statements is false?

a. The following code selects an element in grades by specifying a tuple containing the element’s row and column indices in square brackets:

grades[0, 1]

b. To select a single row, specify only one index in square brackets, as in

grades[1]

c. To select multiple sequential rows, use slice notation, as in

grades[0:2]

and to select multiple non-sequential rows, use a list of row indices, as in

grades[[1, 3]]

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.10 Q2: Which of the following statements about two-dimensional arrays is false?

a. You can select subsets of the columns by providing a tuple specifying the row(s) and column(s) to select.

b. The following code selects only the elements in the first column:

grades[:, 0]

c. You can select consecutive columns using a slice, as in

grades[:, 1:3]

or specific columns using a list of column indices, as in

grades[:, [0, 2]]

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* Views: Shallow Copies

7.11 Q1: Objects that “see” the data in other objects, rather than having their own copies of the data are called          objects.

a. subordinate

b. scene

c. view

d. aspect

Answer: c. view

7.11 Q2: Views are also known as          copies.

a. deep

b. secondary

c. reliant

d. shallow

Answer: d. shallow

7.11 Q3: Which of the following statements is false?

a. The following code creates an array and a view of that array:

import numpy as np  
  
numbers = np.arange(1, 6)  
numbers2 = numbers.view()

b. After the statements in Part (a) execute, numbers and numbers2 are the same object.

c. If we modify an element in numbers, numbers2 "sees" the updated data.

d. If we modify an element in numbers2, numbers "sees" the updated data.

Answer: b. After the statements in Part (a) execute, numbers and numbers2 are the same object. Actually, numbers and numbers2 are *different* objects, but both arrays’ elements refer to the same data.

7.11 Q4: Which of the following statements is false?

a. The following code makes numbers2 a slice that views only the first three elements of numbers:

numbers2 = numbers[0:3]

b. After the statement in Part (a) executes, numbers and numbers2 are different objects.

c. After the statement in Part (a) executes, modifying one an element in numbers2 also modifies the corresponding element in numbers.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

* Deep Copies

7.12 Q1: Which of the following statements is false?

a. Though views are separate array objects, they save memory by sharing element data from other arrays. However, when sharing mutable values, sometimes it’s necessary to create a deep copy with an independent copy of the original data. This is especially important in multi-core programming, where separate parts of your program could attempt to modify your data at the same time, possibly corrupting it. The array method copy returns a new array object with a deep copy of the original array object’s data.

b. Assuming numbers is an array of integers, the following statement creates a deep copy of that array:

numbers2 = numbers.copy()

c. After the statement in Part (b) executes, numbers2 has a separate copy of the data in numbers.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.12 Q2: If you need          copies of other types of Python objects, pass them to the          module’s          function.

a. shallow, shallowcopy, copy

b. deep, deepcopy, copy

c. shallow, copy, copy

d. deep, copy, deepcopy

Answer: d. deep, copy, deepcopy

* Reshaping and Transposing

7.13 Q1: We use array method \_\_\_\_\_\_\_\_ to produce two-dimensional arrays from one-dimensional ranges.

a. shape

b. rectangularize

c. reshape

d. None of the above.

Answer: c. reshape

7.13 Q2: Which of the following statements is false?

a. The array methods reshape and resize both enable you to change an array’s dimensions.

b. Method reshape returns a deep copy of the original array with the new dimensions. It does not modify the original array.

c. Method resize modifies the original array’s shape.

d. All of the above statements are true.

Answer: Method reshape returns a deep copy of the original array with the new dimensions. It does not modify the original array. Actually, Method reshape returns a *view* (shallow copy) of the original array with the new dimensions.

7.13 Q3: Which of the following statements is false?

a. You can take a multidimensional array and flatten it into a single dimension with the methods flatten and ravel.

b. Method flatten deep copies the original array’s data.

c. Modifying a flattened array does not modify the original array’s data.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.13 Q4: Which of the following statements is false?

a. You can quickly transpose an array’s rows and columns—that is “flip” the array, so the rows become the columns and the columns become the rows. The T attribute returns a transposed view (deep copy) of the array.

b. If a grades array represents two students’ grades (the rows) on three exams (the columns), the following code transposes the rows and columns to view the data as the grades on three exams (the rows) for two students (the columns):

grades.T

c. Transposing does not modify the original array.

d. All of the above statements are true.

Answer: a. You can quickly transpose an array’s rows and columns—that is “flip” the array, so the rows become the columns and the columns become the rows. The T attribute returns a transposed view (deep copy) of the array. Actually, the T attribute returns a transposed view, which is a *shallow copy* of the original array.

7.13 Q5: Which of the following statements is false?

a. You can combine arrays by adding more columns or more rows—known as vertical stacking and horizontal stacking, respectively.

b. NumPy’s hstack (horizontal stack) function receives a tuple containing the arrays to combine, as in

np.hstack((grades, grades2))

The extra parentheses are required because hstack expects one argument.

c. Functions hstack and vstack modify the first element of their argument tuples.

d. All of the above statements are true.

Answer: a. You can combine arrays by adding more columns or more rows—known as vertical stacking and horizontal stacking, respectively. Actually, these are known as horizontal stacking and vertical stacking, respectively.

* Intro to Data Science: pandas **Series** and **DataFrame**s

7.14 Q1: Which of the following statements is false?

a. NumPy’s array is optimized for heterogeneous numeric data that’s accessed via integer indices.

b. Data science presents unique demands for which more customized data structures are required.

c. Big data applications must support mixed data types, customized indexing, missing data, data that’s not structured consistently and data that needs to be manipulated into forms appropriate for the databases and data analysis packages you use. Pandas is the most popular library for dealing with such data.

d. Pandas provides two key collections—Series for one-dimensional collections and DataFrames for two-dimensional collections.

Answer: a. NumPy’s array is optimized for heterogeneous numeric data that’s accessed via integer indices. Actually, NumPy’s array is optimized for *homogeneous* numeric data that’s accessed via integer indices.

7.14 Q2: Which of the following statements is false?

a. You can use pandas’ MultiIndex to manipulate multi-dimensional data in the context of Series and DataFrames.

b. The name pandas is derived from the term “panel data,” which is data for spatial measurements, such as stock prices or historical temperature readings.

c. NumPy and pandas are intimately related. Series and DataFrames use arrays “under the hood.” Series and DataFrames are valid arguments to many NumPy operations. Similarly, arrays are valid arguments to many Series and DataFrame operations.

d. Series and DataFrames make it easy for you to perform common tasks like selecting elements a variety of ways, filter/map/reduce operations (central to functional-style programming and big data), mathematical operations, visualization, data preparation and more.

Answer: b*.* The name pandas is derived from the term “panel data,” which is data for spatial measurements, such as stock prices or historical temperature readings. Actually, the name pandas is derived from the term “panel data,” which is data for measurements *over time*, such as stock prices or historical temperature readings.

* pandas Series

7.14 Q3: Which of the following statements is false?

a. NumPy arrays use only zero-based integer indexes.

b. Like arrays, Series use only zero-based integer indexes.

c. Series may have missing data, and many Series operations ignore missing data by default.

d. All of the above statements are true.

Answer: b. Like arrays, Series use only zero-based integer indexes. Actually, Series support custom indexing, including even non-integer indexes like strings.

7.14 Q4: Which of the following statements is false?

a. By default, a Series has integer indexes numbered sequentially from 0.

b. The following code creates a Series of student grades from a list of integers:

import pandas as pd  
grades = pd.Series([87, 100, 94])

c. The Series argument in Part (b)’s code also may be a tuple, a dictionary, an array, another Series or a single value.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.14 Q5: Which of the following statements is false?

a. Pandas displays a Series in two-column format with the indexes left aligned in the left column and the values right aligned in the right column.

b. After listing the Series elements, pandas shows the data type (dtype) of the underlying array’s elements.

c. It is easier to display a list than a Series in this nice two-column format.

d. All of the above statements are true.

Answer: c. It is easier to display list than a Series in this nice two-column format. Actually, it is easier to display a Series than a list in this nice, two-column format, because pandas handles the formatting for you. With a list, you’d have to create this format yourself.

7.14 Q6: Which of the following statements is false?

a. Series provides many methods for common tasks including producing various descriptive statistics, like count, mean, min, max, etc.

b. Each of the descriptive statistics in Part (a) is a functional-style reduction.

c. Calling Series method describe produces all the Part (a) descriptive stats and more.

d. The 25%, 50% and 75% statistics produced by Series method describe are quartiles: 50% represents the median of the sorted values, 25% represents the median of every second one of the sorted values and 75% represents the median of every fourth one of the sorted values.

Ans: d. The 25%, 50% and 75% statistics produced by Series method describe are quartiles: 50% represents the median of the sorted values, 25% represents the median of every second one of the sorted values and 75% represents the median of every fourth one of the sorted values. Actually, 50% represents the median of the sorted values, 25% represents the median *of the first half of the sorted values* and 75% represents the *median of the second half of the sorted values*.

7.14 Q7: The interquartile range is the 75% quartile minus the 25% quartile, which is another measure of         , like standard deviation and variance.

a. central tendency

b. flexibility

c. dispersion

d. consistency

Answer: c. dispersion

7.14 Q8: Which of the following statements is false?

a. You can specify custom indices for a Series with the index keyword argument when creating the Series, as in

grades = pd.Series([87, 100, 94], index=['Wally', 'Eva', 'Sam'])

b. In Part(a), we used string indices, but you can use other immutable types, including integers not beginning at 0 and nonconsecutive integers.

c. If you initialize a Series with a dictionary, its values become the Series’ indices, and its keys become the Series’ element values.

d. All of the above statements are true.

Answer: If you initialize a Series with a dictionary, its values become the Series’ indices, and its keys become the Series’ element values. Actually, if you initialize a Series with a dictionary, its keys become the Series’ indices, and its values become the Series’ element values.

7.14 Q9: Which of the following statements is false? Consider a Series of hardware-related strings:

hardware = pd.Series(['Hammer', 'Saw', 'Wrench'])

a. Pandas left-aligns string element values and the dtype for strings is object.

b. The following code calls string method contains on each element to determine whether the value of each element contains a lowercase 'a':

hardware.str.contains('a')

and returns a Series containing bool values indicating the contains method’s result for each element.

c. The following code uses the Series str attribute to invoke string method upper on every Series element, producing a new Series containing the uppercase versions of each element in hardware:

hardware.str.upper()

d. All of the above statements are true.

Answer: a. Pandas left-aligns string element values and the dtype for strings is object. Actually, pandas *right-aligns* string element values and the dtype for strings is object.

* DataFrames

7.14 Q10: A pandas          is an enhanced two-dimensional array.

a. Series

b. DataFrame

c. dictionary

d. array

Answer: b. DataFrame

7.14 Q11: Which of the following statements is false?

a. Like Series, DataFrames can have custom row and column indices, and offer additional operations and capabilities that make them more convenient for many data-science oriented tasks.

b. DataFrames support missing data.

c. The Series representing each column may contain different element types.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.14 Q12: Which of the following statements is false?

a. With DataFrames you can specify custom indexes with the index keyword argument when we create a DataFrame, as in:

pd.DataFrame(grades\_dict, index=['Test1', 'Test2', 'Test3'])

b. The following code uses the index attribute to change the DataFrame’s indexes from sequential integers to labels:

grades.index = ['Test1', 'Test2', 'Test3']

c. When specifying the indexes, you must provide a one-dimensional collection that has the same number of elements as there are rows in the DataFrame; otherwise, a ValueError occurs.

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.14 Q13: Assuming the following DataFrame grades:

Wally Eva Sam Katie Bob  
Test1 87 100 94 100 83  
Test2 96 87 77 81 65  
Test3 70 90 90 82 85

Which of the following statements is false?

a. One benefit of pandas is that you can quickly and conveniently look at your data in many different ways, including selecting portions of the data.

b. The following expression selects 'Eva' column and returns it as a Series:

grades['Eva']

c. If a DataFrame’s column-name strings are valid Python identifiers, you can use them as attributes. The following code selects the 'Sam' column using the Sam attribute:

grades.Sam

d. All of the above statements are true.

Answer: d. All of the above statements are *true*.

7.14 Q14: Which of the following statements about DataFrames is false?

a. The index can be a slice. In the following slice containing, the range specified includes the high index ('Test3'):

grades.loc['Test1':'Test3']

b. When using slices containing integer indices with iloc, the range you specify excludes the high index (2):

grades.iloc[0:2]

c. To select specific rows, use a tuple rather than slice notation with loc or iloc.

d. All of the above statements are true.

Answer: c. To select specific rows, use a tuple rather than slice notation with loc or iloc. Actually, to select specific rows, use a *list* rather than slice notation with loc or iloc.

7.15 Q1: Which of the following statements is false?

a. DataFrames have a describe method that calculates basic descriptive statistics for the data and returns them as two-dimensional array.

b. In a DataFrame, the statistics are calculated by column.

c. Method describe nicely demonstrates the power of array-oriented programming with a functional-style call—Pandas handles internally all the details of calculating these statistics for each column.

d. You can control the precision of floating-point values and other default settings with pandas’ set\_option function.

Answer: a. DataFrames have a describe method that calculates basic descriptive statistics for the data and returns them as a two-dimensional array. Actually, the DataFrame describe method calculates basic descriptive statistics for the data and returns them as a *DataFrame*.

7.16 Q1: Which of the following statements is false?

a. You can quickly transpose a DataFrame’s rows and columns—so the rows become the columns, and the columns become the rows—by using the T attribute.

b. T returns a transposed copy of the DataFrame.

c. Assuming the following DataFrame grades:

Wally Eva Sam Katie Bob  
Test1 87 100 94 100 83  
Test2 96 87 77 81 65  
Test3 70 90 90 82 85

rather than getting the summary statistics by student, you can get them by test. Simply call describe on grades.T, as in:

grades.T.describe()

d. To see the average of all the students’ grades on each test, call mean on the T attribute:

grades.T.mean()

Answer: b. T returns a transposed copy of the DataFrame. Actually, T returns a transposed *view* (not a copy) of the DataFrame.