Part I

Core pandas

Chapter 1

Introducing pandas

Chapter 2

The Series object

2.1 Overview of a Series

2.1.1 Creating a Series with missing values

Pandas automatically converts numeric values from integers to floating-points when it spots a nan value; this internal technical requirement allows the library to store numeric values and missing values in the same homogeneous Series.

2.2 Retrieving the first and last rows

A Python object has both attributes and methods. An attribute is a piece of data belonging to an object—a characteristic or detail that the data structure can reveal about itself. We accessed Series attributes such as size, shape, values, and index.

By comparison, a method is a function that belongs to an object—an action or command that we ask the object to perform. Methods typically involve some analysis, calculation, or manipulation of the object's attributes. Attributes define an object's state, and methods define an object's behavior.

2.3 Passing the Series to Python's built-in functions

In Python, we use the `in` keyword to check for inclusion. In pandas, we can use the in keyword to check whether a given value exists in the Series' index.

To check for inclusion among the Series' values, we can pair the in keyword with the `values` attribute. Remember that values exposes the ndarray object that holds the data itself.

We can use the inverse not in operator to check for exclusion. The operator returns True if pandas cannot find the value in the Series

Chapter 3

Series methods

3.1 Invoking a function on every Series value with the apply method

A function is a first-class object in Python, which means that the language treats it like any other data type. A function may feel like a more abstract entity, but it's as valid a data structure as any other.

Here's the simplest way to think about first-class objects. Anything that you can do with a number, you can do with a function.