Learn to code for data analysis

Week 2 glossary

Programming and data analysis concepts

The **bitwise operators** `&` (and) and `|` (or) are used in pandas to build more complicated expressions from two comparison expressions (typically involving column comparisons).

A **Boolean** has one of two possible values: `True` or `False`.

A **Comma Separated Values (CSV)** file is a plain text file that is used to hold tabular data.

A **list** is a sequence of values, separated by commas, and written within square brackets.

There are six **comparison operators** that can be used to compare number, string and date values. Expressions composed of these operators evaluate to `True` or `False`. These operators can also be used to compare every value in a column, row by row, against some number, string or date value. When used in this manner the operators return a series of Boolean values.

The **'dot' notation** is used to access a dataframe's methods and attributes.

The `Series` data type is a collection of values with an integer index that starts from zero. Each column in a dataframe is an example of the `Series`data type. The `Series` data type has many of the same methods as the`DataFrame` data type.

The `object` data type is how pandas represents strings.

The `datetime64` data type is how pandas represents dates.

The `int64` data type is how pandas represents integers (whole numbers).

The `float64` data type is how pandas represents floating point numbers (decimals).

Functions and methods

`asType(aType)` when applied to a dataframe column, the method changes the data type of each value in that column to the type given by the string `aType`.

`datetime(yyyy, mm, dd)` the function takes three arguments, `yyyy` a four digit integer representing a year, `mm` a two digit integer representing a month and `dd` a two digit integer representing a day. From these arguments the function creates and returns a value of `datetime64`.

`dropna()` when applied to a dataframe returns a new dataframe without the rows that have at least one missing value.

`head()` gets and displays the first five rows of a dataframe. Optionally the method can take an integer argument to specify how many rows (from and including row 0) to get and display.

`irow(index)` gets and displays the row in the dataframe indicated by argument `index`.

`isnull()` is a series method that checks which rows in that series have a missing value.

`fillna(value)` is a series method that returns a new series in which all missing values have been filled with the given value.

`plot()` when applied to a dataframe column of numeric values, the method displays a graph of those values. The x-axis shows the dataframe's index and the y-axis the range of the column's values. Before the method is called you first need to execute `%matplotlib inline`.

`read_csv(csvFile)` creates a dataframe from the dataset in the CSV file.

`rename(columns={oldName : newName})` renames the column `oldName`to `newName`.

`str.rstrip(suffix)` when applied to a dataframe column of string values, the method removes the argument `suffix` from the end of each string value in the column.

`tail()` gets and displays the last five rows of a dataframe. Optionally the method can take an integer argument to specify how many rows (until and including the last row) to get and display.

`to_datetime(aSeries)` when applied to a series, typically a column from a dataframe, this function returns a new series in which each value in`aSeries` has been changed to type `datetime64`.