What is Pandas?

Primary Pandas data types

Pandas is the primary Python module for data science
- It builds on top of the capabilities of NumPy, for efficient storage and manipulation of data arrays
- You can think of Pandas as an extension of NumPy

What does Pandas add to NumPy arrays?
- Separate data types per column
- Labelled columns (not just column number)
- Specific index type for rows (not just row number)

Pandas also provides extra data processing functionality
- E.g. join datasets (like in SQL)
- E.g. pivot tables (like in Excel)
- E.g. string functions and reg exps (like in Python)

Series
- Stores a 1D array of data, like a NumPy 1D ndarray
- Elements are accessed by index

DataFrame
- Stores a 2D array of data, like a NumPy 2D ndarray
- Elements are accessed by column name and index

Index
- Series and DataFrame objects have a specific index
- The index is like a 1D array of row-specifiers
- Can be any type (e.g. integer, date/time, string, etc.)

https://pandas.pydata.org/pandas-docs/stable/reference

Creating a simple `Series` object

Creating a `Series` with an explicit index

Creating a `Series` from a dictionary

You can create a Series from a simple list of items
- Pandas automatically creates an integral index (0, 1, 2, etc...)
- You access items by integral index

Ex1

You can specify an explicit index for a Series object
- This is different to NumPy ndarray, where index is just a number
- You access items by key

Ex2

You can create a Series from a Python dictionary
- Specify keys/value pairs
- You access items by key

Ex3

`Series` with a non-numeric index

`Series` with a numeric index

There are two ways to index into a Series
- Using an explicit index (like a key in a dictionary)
- Using an implicit index (like a row number in an array)

We'll also see how to disambiguate the meaning of [i] if the Series has numeric keys
- Is i an explicit index or a row number...?

If you have a Series with a non-numeric index:
- You can access an item via an explicit index, e.g. ['E101']
- You can also access an item via an implicit index, e.g. [1]

Ex4

If you have a Series with a numeric index, be careful!
- Indexing uses an explicit index, but slicing uses an implicit index!

For clarity:
- Use .loc for an explicit index, or use .iloc for an implicit index

Ex5

Creating a simple `DataFrame` object

Accessing a DataFrame column

Creating a `DataFrame` with an explicit index

Creating a DataFrame from columnar data

In the previous section we showed how to create and use a Pandas Series object
- A Pandas Series is like a NumPy 1D array, with indexing

In this section, we'll show how to create and use a Pandas DataFrame object
- A Pandas DataFrame is like to a NumPy 2D array, with indexing
- Each column has a name and data type
- Each row is accessed by index

There are various ways to create a DataFrame...
- This example creates a DataFrame from a collection of tuples
- Each tuple has 2 values (name, year born)
- The DataFrame has 2 columns, with the specified column names
- The DataFrame has an implicit integral index

Ex6

You can create a DataFrame from a list of dictionaries
- Each dictionary represents one row in the DataFrame

Ex7

You can access a column in a DataFrame by name
- Returns a Series object
- The Series object contains that column's values (plus indices)

Ex8

You can specify an explicit index for a DataFrame object
- Very common, makes it easy to access rows in the DataFrame

Ex9

You can create a DataFrame from columnar data
- Create column as a Series (key is index, value is column data)
- Pandas merges Series together (based on key)
- Pandas fills any gaps with NaN

Ex10

Working with `DataFrame` columns

Working with `DataFrame` rows

Working with `DataFrame` as a 2D array

Let's have a look at how to access columns and rows in a DataFrame
- The syntax is quite subtle, so care is needed!

A "simple index" into a DataFrame gets column(s)

Ex11

A "slice/fancy-index/mask" into a DataFrame gets rows

Ex12

You can use .loc or .iloc to treat a DataFrame as a 2D array, and then use [row-indexer, col-indexer] syntax

Ex13