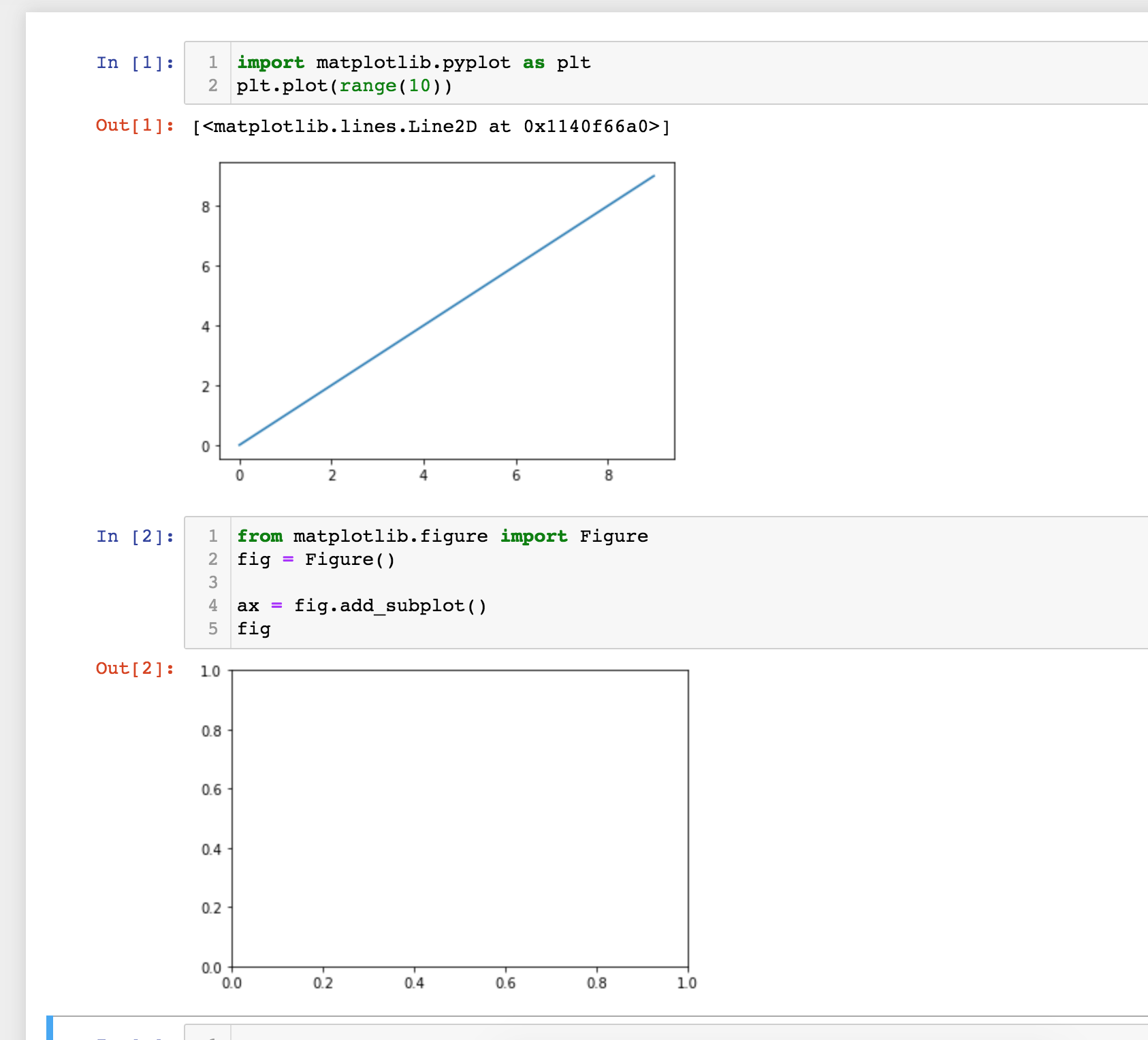
**Homework Week 1**

**Question 1 [20 points]**

**In no more than 500 words, explain the differences between a Jupyter Notebook (.ipynb extension) and a Python file (.py extension). Focus on the advantages and disadvantages of one or the other, particularly for data analysts/scientists. You are encouraged to use examples and screenshots of the different scripts to support your arguments.**

Advantages of Jupyter Notebook:

1. Interactive: Jupyter Notebook provides an interactive environment for data analysis and scientific computing, allowing users to write and run code, visualise data, and document their work all in one place. This makes it easier for users to explore data and test ideas quickly.
2. Cell output: Jupyter Notebook displays output inline with the code via clear cells, allowing users to see the results of their code immediately after running it. This is particularly useful for data analysis, where users need to see the output of their code to make decisions about how to proceed.



1. Rich media support: Jupyter Notebook supports a wide range of media types, including images, videos, and HTML. This makes it easy for users to create rich, interactive documents that can be shared with others.
2. Easy collaboration: Jupyter Notebook allows users to share their work easily with others, either by sharing the notebook file or by publishing it as a web page. This makes it easier for teams to collaborate on data analysis and scientific computing projects.

Disadvantages of Jupyter Notebook:

1. Version control: Jupyter Notebook files are not always easy to version control using Git or other version control systems. This is because the files contain both code and text, making it difficult to track changes to the code over time.
2. Debugging: Debugging Jupyter Notebook files can be more difficult than debugging Python files, especially if the code is spread across multiple cells. This is because the code is executed in a non-linear way, which can make it harder to trace the flow of data through the code.

And vice versa, the advantages of Python files:

1. Version control: Python files are easier to version control using Git or other version control systems.
2. Debugging: Debugging Python files is usually easier than debugging Jupyter Notebook files, especially if the code is well-structured and follows best practices.

Disadvantages of Python files:

1. Lack of interactivity: Python files are not as interactive as Jupyter Notebook files, which can make it more difficult for users to explore data and test ideas quickly.
2. Lack of inline output: Python files do not display output inline with the code, which can make it harder for users to see the results of their code immediately after running it.

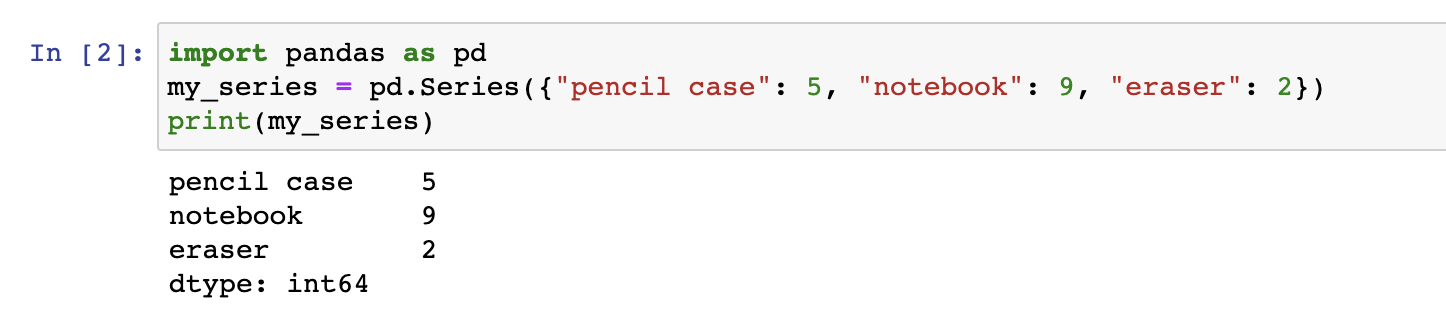
Overall, Jupyter Notebook is more suitable for exploratory data analysis and quick prototyping, while Python files are better for building more structured and scalable projects. Data analysts and scientists often use Jupyter Notebook for exploratory data analysis and Python files for building production-ready models and applications. For example, a data analyst might use Jupyter Notebook to explore a dataset and test different machine learning algorithms, then use Python files to build a production-ready model based on the results of their analysis.

**Question 2 [10 points]**

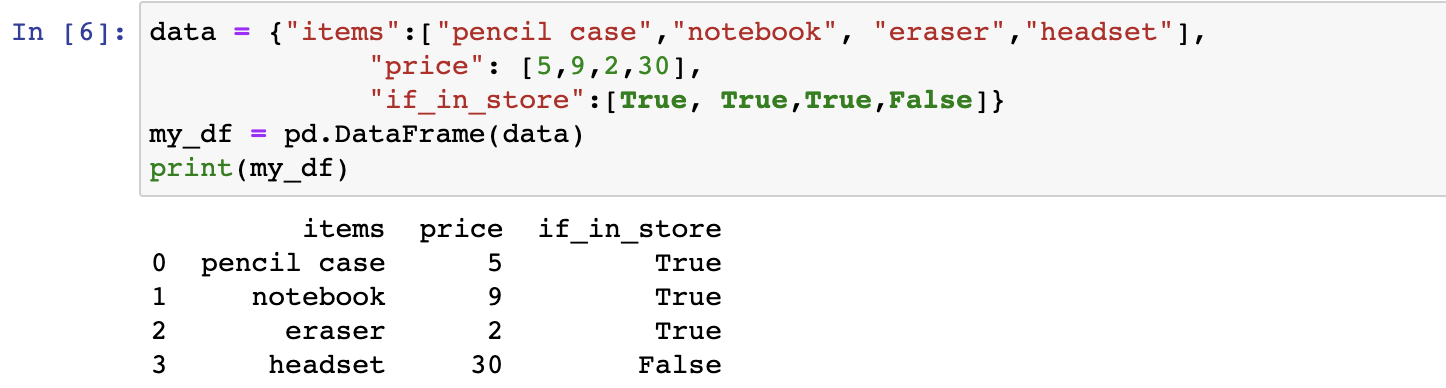
**What is the difference between a Pandas DataFrame and a Pandas series. Show an example of how you create each of them.**

In Pandas, a Series is a one-dimensional labelled array, while a DataFrame is a two-dimensional labelled data structure with columns of potentially different types. In other words, a DataFrame is a collection of Series.

A Pandas Series can be created using the pd.Series() function, and it can contain any data type, such as integers, strings, floats, or even other objects like lists or dictionaries. Here's an example:



A DataFrame, on the other hand, can be created using the pd.DataFrame() function, and it can contain multiple columns, each of which can have a different data type. Here's an example:



**Question 3 [10 points]**

**Starting from the argument that a Pandas DataFrame represents rectangular data, use the internet and other resources to describe in no more than a few sentences the difference between rectangular and non-rectangular data.**

Rectangular data is organised in a tabular format with rows and columns, where each row represents a unique observation and each column represents a variable. Each cell in a rectangular data structure contains a single value or data point. In contrast, non-rectangular data structures do not have a tabular format and may contain complex or hierarchical data structures. Examples of non-rectangular data structures include JSON files, XML files, and hierarchical data formats such as HDF5. Non-rectangular data structures may require specialised tools and techniques for manipulation and analysis, whereas rectangular data can be easily analysed using tools such as Pandas DataFrames.

**Question 4 [10 points]**

**Starting from the data visualisation usage from Session 2, give examples of when figures could be:**

1. **Of use to the data scientist to identify patterns in the data/highlight the important parts of a data set;**

As a data scientist, figures can be of great use to identify patterns in the data or highlight important parts of a data set. For example, a scatter plot can be used to identify relationships between two variables, a histogram can be used to identify the distribution of a single variable, and a boxplot can be used to identify the spread of the data and potential outliers. By creating figures, data scientists can quickly identify trends, outliers, and patterns in the data, which can help inform decision-making and guide further analysis.

1. **Tell a story and create business presentations.**

Data visualisation is also crucial in creating business presentations as it helps tell a story and communicate insights effectively. For instance, a histogram can show the trend of a company's sales over time, a bar chart can show the distribution of revenue across different departments, and a heat map can highlight the areas where the company is performing well or needs improvement. By presenting data in a visual format, business presentations can effectively communicate key insights to stakeholders, help guide decision-making, and drive organisational change.

**Question 5 [50 points]**

**Each sub-question is worth 10 points.**

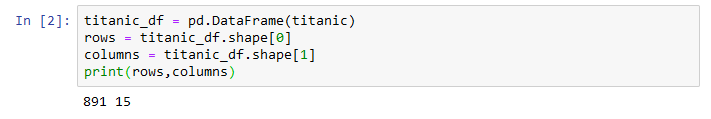
**Using the *titanic dataset* which you can read into your notebook using the following code,**

import pandas as pd

titanic = pd.read\_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/titanic.csv')

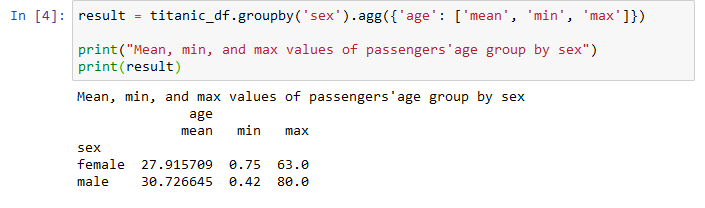
**answer the following questions:**

1. **How many columns and rows does the data have?**

****

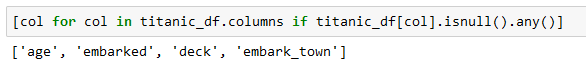
There are 15 columns and 891 rows in this csv file.

1. **Get a sense of your data and find the *min, max,* and *count/mean* depending on the data type.**

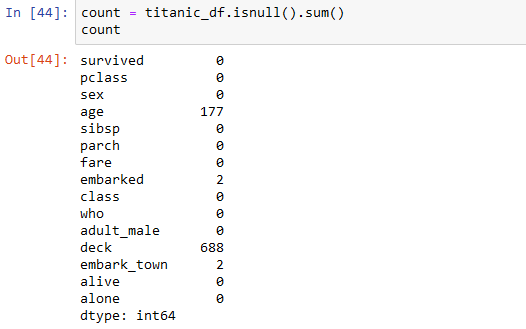


I calculated the mean, min, and max values of passengers’ age group by sex to get an overview of the age range of different gender.

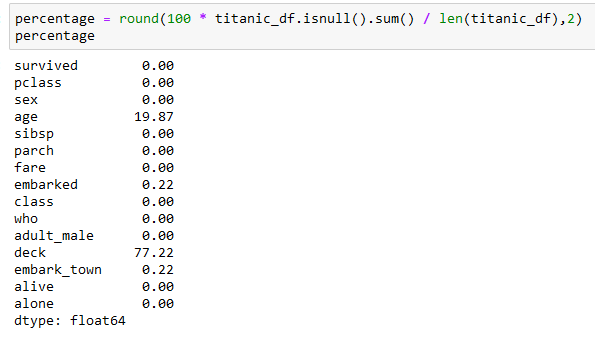
1. **Give an overview (code and an explanation) of all missing values in the data.**



This line of code will show which columns contain NaN values - as we can see, there are a total of 4 columns containing missing values, namely 'age', 'embarked', 'deck', and 'embark\_town.'

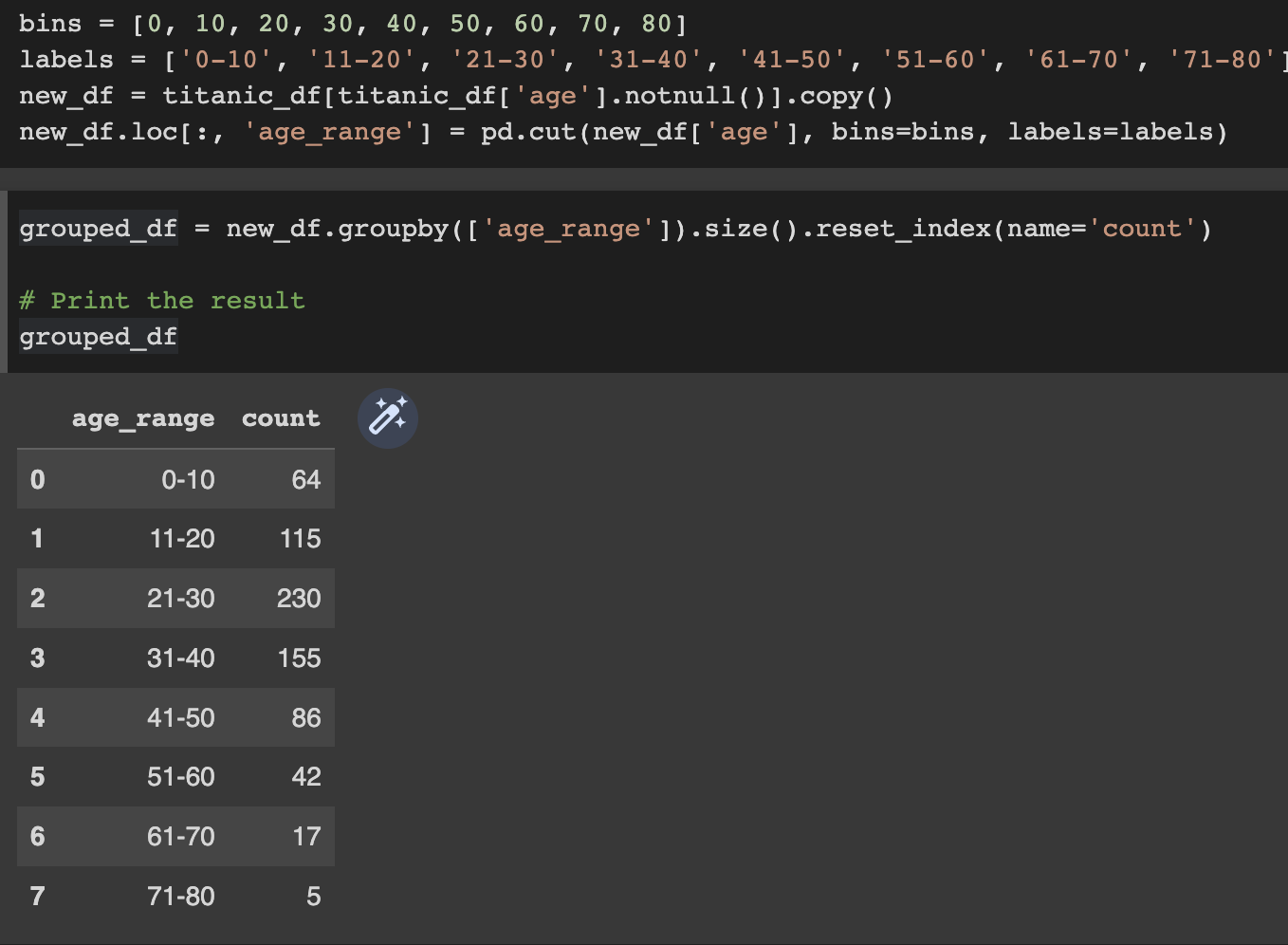


This displays the numbers of missing values in each column.



The output of this cell of codes shows the proportion of missing values among all values in each column.

1. **Delete the rows where you do not have information about the *age* of the person. Then group the passengers in a 10 year age range (for example, you can do something like 0 – 10, 11 – 20, 21 – 30, etc).**



1. **For each age category created in d), find out how many passengers are *female/male,* and how many travelled in each *class*.**

