Assignment Activity 3: Analysing the data

This assignment activity will help you to complete the final assignment. The provided instructions will guide you to the minimum expected activities to complete during your exploratory data analysis and the presentation of your insights. You're encouraged to think critically and apply logic to identify patterns and trends the organisation can use to answer specific questions. You will also use these insights to inform both your technical report and business presentation.

**Scenario**

Recall the work you're doing for the NHS for your final assignment. Review the requirements introduced in: [**Assignment: Diagnostic Analysis Using Python**](https://fourthrev.instructure.com/courses/895/assignments/2971). You have previously imported and started to explore the data. You'll now continue this exploration to search for answers to more specific questions posed by the NHS. In practice, you will typically receive some guiding questions from the organisation. You should answer the questions, but also supplement this with your own questions and insights.

**Objective**

Analyse the data sets to answer the following questions:

1. Between what dates were appointments scheduled?
2. Which service setting reported the most appointments in North West London from 1 January to 1 June 2022?
3. Which month had the highest number of appointments?
4. What was the total number of appointments per month per data source?

**Approach**

1. **Continue** to work on the Jupyter Notebook that you created for [**Assignment Activity 2: Importing and exploring data**.](https://fourthrev.instructure.com/courses/895/pages/assignment-activity-2-importing-and-exploring-data)
2. **Download** the [**LSE\_DA201\_Assignment\_template.ipynb**](https://fourthrev.instructure.com/courses/895/files/186436/download) file. This Jupyter Notebook provides a suggested workflow to follow as you work through the assignment activities. You're **not required** to use this template to complete the assignment. If you decide to use it for your assignment, make a copy of the Notebook and save it using the assignment naming convention: LastName\_FirstName\_DA201\_Assignment\_Notebook.ipynb.
3. **Follow** the guidelines to answer the four questions. Select each tab to view the suggested approach.
4. **Back up** your work to a safe location. This would allow you to revert to a previous state in the case of making a mistake in the code, or deleting a section by mistake. (A simple way of doing this is to save or mail a compressed version at frequent intervals).

**Question 1**

**Between what dates were appointments scheduled?**

1. **View the appropriate DataFrames and columns to determine the format of the dates.**
2. **Change the date format to a datetime format, and sense-check the DataFrames with dtypes and the head() method.**
3. **Determine the first (e.g. min()) and last (e.g. max()) date of scheduled appointments for each DataFrame.**

**Question 2**

**Which service setting reported the most appointments in North West London from 1 January to 1 June 2022?**

1. **Create a subset of the nc DataFrame (e.g. nc\_subset).**
2. **Use the sub\_icb\_location code of NHS North West London ICB - W2U3Z.**
3. **Specify the date of appointments as 1 January 2022 to 1 June 2022.**
4. **Determine the number of service settings.**

**Question 3**

**Which month had the highest number of appointments?**

1. **Determine the total number of appointments per month with the sum() function.**
2. **Use the groupby() method to sort the appointment\_date column based on year and month. (Hint: Use the dt method with the Series.dt.year and Series.dt.month attributes to calculate the number of appointments per month. The code snippet will be df['col'].groupby([df['col'].dt.year, df['col'].dt.month]).)**
3. **Use the sort\_values() method based on count\_of\_appoints of the nc DataFrame.**
4. **Information**
5. Python has a built-in attribute that can return a Numpy array containing datetime in the underlying data of a given Series object. For example, Series.dt.year will return the year as an array, while Series.dt.month will return the month as an array. This is extremely useful when we want to use the groupby() and aggregate() functions.

**Question 4**

**What was the total number of records per month?**

1. **Display the total number of records per month per data source.**

**Summarise and explain** how you arrived at the answers to the four questions and what, if any, further questions arose in the process.