Assignment Activity 4: Visualising and identifying initial trends



This assignment activity will help you to successfully 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 are encouraged to think critically and apply logic to identify patterns and trends that can be used by the organisation and how to present this to technical and business stakeholders.

**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 were tasked to create visualisations and identify possible monthly and seasonal trends for service settings, context types, and national categories.

**Objective**

Visualisation can be used for exploratory analysis and communicating insights to the organisation. In this activity, you'll begin to explore the data using visual techniques and then proceed to present your insights to the organisation. Create the following visualisations to determine monthly and seasonal trends:

1. Create three visualisations indicating the number of appointments per month for service settings, context types, and national categories.
2. Create four visualisations indicating the number of appointments for service setting per season. The seasons are summer (June to August 2021), autumn (September to November 2021), winter (December to February 2022), and spring (March to May 2022).

**Approach**

1. **Continue** to work on the Jupyter Notebook that you created for [**Assignment Activity 3: Analyse the data**](https://fourthrev.instructure.com/courses/895/pages/assignment-activity-3-analysing-the-data). Remember to ensure that your Notebook is in the correct state by re-executing all the code from the start of the Notebook.
2. **Prepare** your workstation by importing the necessary libraries for visualisation (e.g. matplotlib and seaborn). Set the figure size (e.g. 15, 12) and the plot style as white.
3. **Follow** the guidelines proposed for objectives 1 and 2. You will begin by exploring the data visually. 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).

Objective 1

Create three visualisations indicating the number of appointments per month for service settings, context types, and national categories.

1. Change the data type of appointment\_month to string for ease of visualisation.
2. Aggregate the appointments per month, and determine the sum of the appointments per month.
   * Create a new DataFrame (e.g. nc\_ss).
   * Use the groupby() function to group the monthly appointments and service settings for the nc DataFrame.
   * Calculate the total number of appointments (sum). (**Hint:** Remember to reset the index.)
   * View the new DataFrame.
3. Create a lineplot with Seaborn.
   * Specify the x and y variables.
   * Specify the hue and data.
   * Set ci=None. (**Note:** When creating a lineplot using Seaborn version 0.12.0 onwards, the ci parameter has been deprecated. errorbar=None should be used instead.)
4. Repeat steps b and c for the context types and national categories.
   * Set the DataFrames of context types to nc\_ct and national categories to nc\_nc.

Objective 2

Create four visualisations (one per season) to indicate the number of appointments for service setting per season. The seasons are summer (June to August 2021), autumn (September to November 2021), winter (December to February 2022), and spring (March to May 2022).

1. Aggregate the appointments per month, and determine the sum of the records per season (specified month).
   * Create a new DataFrame (e.g. nc\_ss\_day).
   * Use the groupby() function to group the date of appointment, monthly appointments, and service settings for the nc DataFrame.
   * Calculate the total number of appointments (sum). (**Hint:** Remember to reset the index.)
   * View the new DataFrame.
2. Create a lineplot with seaborn:
   * Specify the x and y variables.
   * Specify the hue and data. (**Hint:** Remember to filter the data and specify the month (e.g. =='2021-08').)
   * Set ci=None. (**Note:** When creating a lineplot using Seaborn version 0.12.0 onwards, the ci parameter has been deprecated. errorbar=None should be used instead.)
3. Repeat Step b to visualise all the seasons.
4. **Revisit** the visualisations created in step 3 and make the necessary changes to allow you to use them to communicate your insights to business users. (**Hint**: Formatting, perspective.)
5. **Save** all the visualisations as PNG files to use in your report and/or presentation.
6. **Summarise** and explain any trends you've identified for both aggregated monthly and seasonal visualisations.
7. **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).