# CPSC 368 Project Midway Checkpoint

Group Star - Briana Pavey, Hannah Martin, Sadie Lee

March 2025

## 1 Project Summary

By comparing efficiency in terms of wait times from request to diagnostic imaging test for different modalities, we can evaluate systematic trends in the different commission regions of NHS England. Furthermore, by predicting wait times of each region for the following year, this can be used as a basis for outlining regions in which NHS England can invest to reduce wait times for diagnostic imaging services and improve patient outcomes.

## 2 Research Questions

We will compare the efficiency of diagnostic imaging services in the public healthcare system of England for each commissioning region by modality during the 2021 year, i.e. mid-pandemic. Efficiency will be explored by calculating the median adult patient wait times in days from request to test for each region and each modality.

Secondarily, we will perform regression analysis to predict wait times for the following year in each region given the wait times, in addition to features including the number of healthcare workers in the region and gross disposable household income (GDHI).

Our old research statement had the secondary outcomes of seeing if:

1. There is a correlation between gross disposable household income (GDHI) and patient wait times for each region.
2. There is a correlation between the number of healthcare workers and patient wait times for each region.

We changed these correlations to the regression analysis given our TA's feedback that simply looking at correlations is not a sufficient methodology.

## 3 Data Cleaning

For our three datasets, modality wait times, GDHI, and healthcare workforce, the following data cleaning steps were done as follows before combining and can be found/reproduced in the Jupyter notebook file attached to the assignment.

1. Modality and wait times data
   (a) Load the Excel file with the relevant sheet (2nd sheet).
   (b) Find the start row of the data to keep and filter out the irrelevant top rows that contain notes like title, written summary, etc.
   (c) Drop the last 9 rows that contain more notes that we do not want in the dataset - these notes were important for us to read but we do not want to use them as data.
   (d) Drop the first column that was loaded as an additional index column and contained no values.
   (e) Save the cleaned data as a csv file for easier use and access later on.

2. GDHI data

    (a) Load the Excel file.

    (b) Drop the first 6 rows that contain notes like copyright and title.

    (c) Drop the last 4 rows that contain data quality notes - these notes were important for us to read but we do not want to use them as data.

    (d) Make the first row of the data frame the column headers.

    (e) Save the cleaned data as a csv file for easier use and access later on.

3. Healthcare workforce data

    (a) Load the Excel file with the relevant sheet (4th sheet).

    (b) Drop the first 6 rows that contain notes like title.

    (c) Drop the last 15 rows that contain notes like copyright and data quality - these notes were important for us to read but we do not want to use them as data.

    (d) Drop the rows that act as 'spacers' in the Excel file, i.e. the row has NaN values for every column.

    (e) Make the first row of the data frame the column headers.

    (f) Rename the first 4 columns to be more interpretable - Region_Num, Region_Name, Org_Name, Org_Code respectively.

    (g) Filter only for 2020-2021 data - the data starts in 2009.

    (h) Propagate the last valid value forwards using `ffill()` for Region_Num and Region_Name as they were organized to not all be filled in, although each NaN value in these columns is mapped to the last valid Region_Num or Region_Name value. This will be easier to use for visualization and analysis.

    (i) Save the cleaned data as a csv file for easier use and access later on.

# 4 Exploratory Data Analysis

In addition to visualizations, we explored each of our datasets by looking at the summary statistics of the data with the `describe()` method, looking at the datatypes of each column with the `info()` method, and inspected any NA or null values by summing `isna()` and `isnull()`.

For the main visualization we are submitting to this assignment, we aimed to look at the distribution of wait times for each modality in each region, which is what our research question also aims to look at and created a faceted box plot. A box plot is suitable to look at the distributions of at least one group of numeric data [2], and is particularly suitable to compare between multiple groups. While the underlying distribution would be the same as a histogram, we wanted to be able to easily see high-level information, i.e. outliers and symmetry, and so chose to use a box plot. Moreover, we chose to use faceted box plots, where each plot represents a different imaging modality, so that we could clearly see the differences between each region, for each modality, as including all of this in one plot would be more difficult to read.

The visualization is shown on the last page, as well as attached to the assignment. The code and data files are also attached to the assignment.

In regard to interpretation of the visualization, it looks like the modality with the greatest variability in interquartile range (IQR) is Single Photon Emission Computerized Tomography (SPECT). This may be due to it being a technique of nuclear imaging, which may not be as common as other modalities such as X-rays, and so waiting times vary greatly by region. Plain radiography and medical photography have little to no variability in IQR for every region, although there are significant outliers in plain radiography.

## 5 EDA Results

The visualization seems to show that for every modality, there are some data that are not mapped to a given region. This may be from the first 9 rows that are not mapped to a region, as no other NA or null values were found in the region column. These rows will likely need to be filtered out in further analysis. Moreover, while cleaning and exploring data, we also saw that there are many missing values in the data that are not NA or null, and have strings such as '.' or '*'. These will need to be handled properly, likely by imputing, but we will have to look into how to do this properly since these values are strings and not numeric data types.

## 6 SQL Script

See the script attached in the assignment. So that we do not have to use individual INSERT statements for every tuple, we created external tables to load our data. Information about external tables was found from [1].

## 7 Data Schema

The schema for the 3 tables are as follows. Primary keys are underlined, foreign keys are bolded.

GDHI(region, gdhi)
Modality(region, org_code, provider_name, modality, apr, may, jun, jul, aug, sep, oct, nov, dec, jan, feb, mar, year)
Workforce(**region_num**, **org_code**, region_name, org_name, apr_2020, may_2020, jun_2020, jul_2020, aug_2020, sep_2020, oct_2020, nov_2020, dec_2020, jan_2021, feb_2021, mar_2021)

| **Table** | **Primary Key** | **Foreign Key** | **Candidate Keys** | **NOT NULL constraints** |
|---|---|---|---|---|
| GDHI | region | N/A | region | gdhi |
| Modality | region, org_code, provider_name, modality | N/A | region, org_code, modality<br>region, provider_name, modality | N/A |
| Workforce | region_num, org_code | cell9 | cell4 | cell5 |

## 8 AI Tool Use Declaration

For the visualization shown below, we used ChatGPT to ask: ”The yticks in my box plot using matplotlib are very close together and there are too many. How can I only show certain intervals instead of every single point?”

Output
Option 1: Set Custom Intervals

```
import numpy as np

# Define the interval (e.g., show yticks every 5 days)
y_min = df["y"].min()
y_max = df["y"].max()
tick_interval = 5  # Change this based on your data

plt.yticks(np.arange(y_min, y_max + tick_interval, tick_interval))
```

Option 2: Reduce the Number of Ticks Automatically

```
import matplotlib.ticker as mticker

plt.gca().yaxis.set_major_locator(mticker.MaxNLocator(integer=True, nbins=8))  # Adjust 'nbins' as needed
```

## References

[1] Oracle utilities external tables concepts, 2025. [Online; last accessed 10-Mar-2025].

[2] Mike Yi. A complete guide to box plots, 2024. [Online; last accessed 10-Mar-2025].

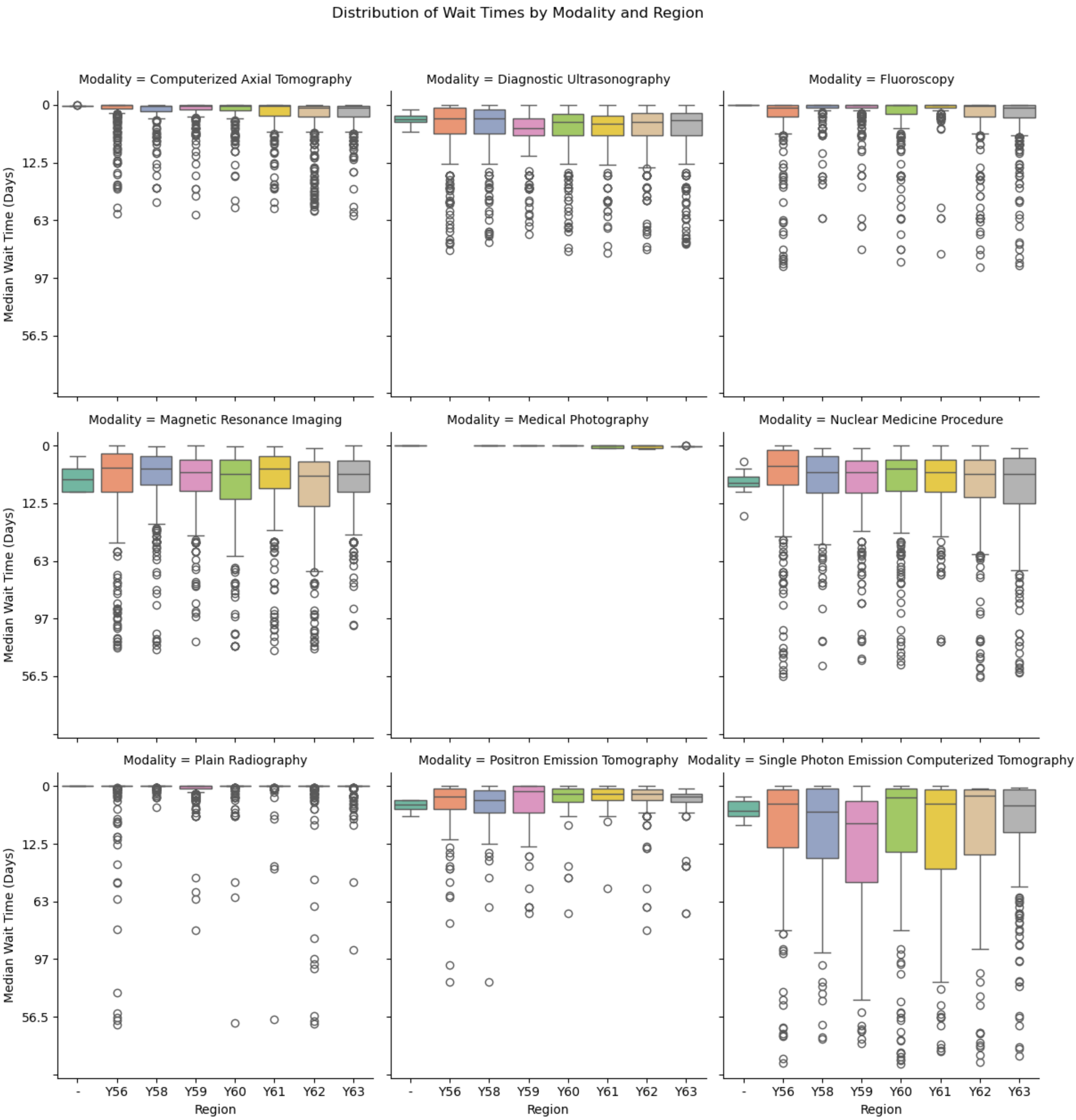

Figure 1: Distribution of wait times by modality and region.