## Assignment: Diagnostic Analysis using Python

### 1. Background of the business

We are part of a team of data analysts contracted by National Health Services (NHS), a publicly funded healthcare system in England. The NHS incurs significant, potentially avoidable, costs when patients miss general practitioner (GP) appointments.

Therefore, reducing or eliminating missed appointments would be beneficial financially as well as socially. The government needs a data-informed approach to deciding how best to handle this problem. At this stage of the project the two main questions posed by the NHS are:

- Has there been adequate staff and capacity in the networks?
- What was the actual utilization of resources?

**We will explore the data to investigate the following:**

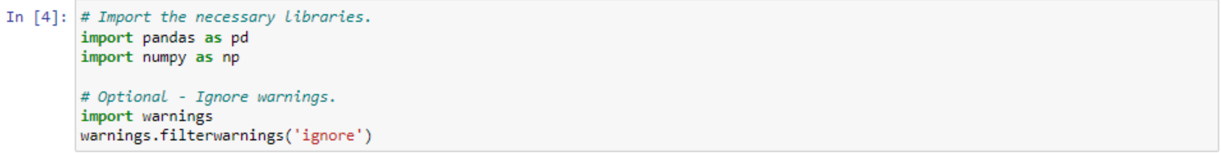
- What is the number of locations, service settings, context types, national categories, and appointment statuses in the data sets?
- What is the date range of the provided data sets, and which service settings reported the most appointments for a specific period?
- What is the number of appointments and records per month?
- What monthly and seasonal trends are evident, based on the number of appointments for service settings, context types, and national categories?
- What are the top trending hashtags (#) on Twitter related to healthcare in the UK?
- Were there adequate staff and capacity in the networks?
- What was the actual utilization of resources?
- What possible recommendations does the data provide for the NHS?

We aim to provide data and recommendations to address the growing concern of missed appointments, and to identify any further trends that may impact the Healthcare system in England.

## 2. Analytical Approach

For this project the team decided to utilize Python, as it allows for reading of large data sets at a considerable speed vs Excel.

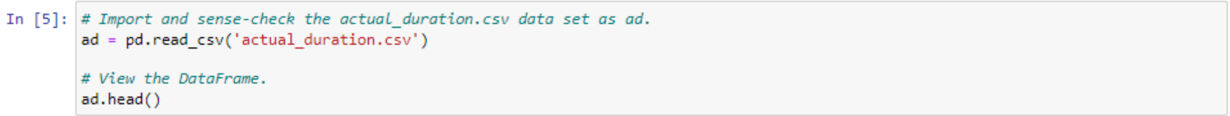
The methodology we used with Python we start off by importing the relevant libraries such as Panda, Seaborn and Numpy that assist in the coding process, Panda is known for providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. Numpy is used for working with arrays, and has functions for working in domains of linear algebra, matrices and fourier transform. Seaborn is a Python data visualization library based on another Python library, it provides a high-level interface for drawing attractive and informative statistical graphics.



```
In [4]: # Import the necessary libraries.
        import pandas as pd
        import numpy as np

        # Optional - Ignore warnings.
        import warnings
        warnings.filterwarnings('ignore')
```

After the relevant libraries are imported, we import the CSV or excel files via Python leveraging Panda.

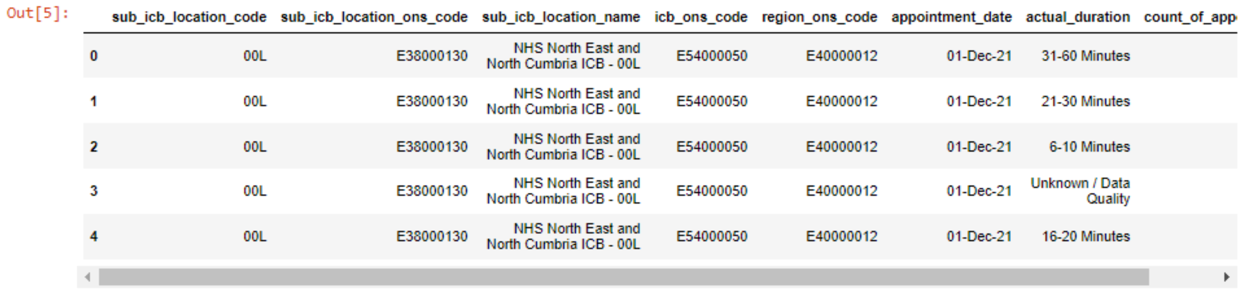


```
In [5]: # Import and sense-check the actual_duration.csv data set as ad.
        ad = pd.read_csv('actual_duration.csv')

        # View the DataFrame.
        ad.head()
```

Good practice is to examine the shape of the imported file, albeit at times there are difficulties reading the CSV or excel files. If errors occur, try opening the file and saving it again that resolves the problem.

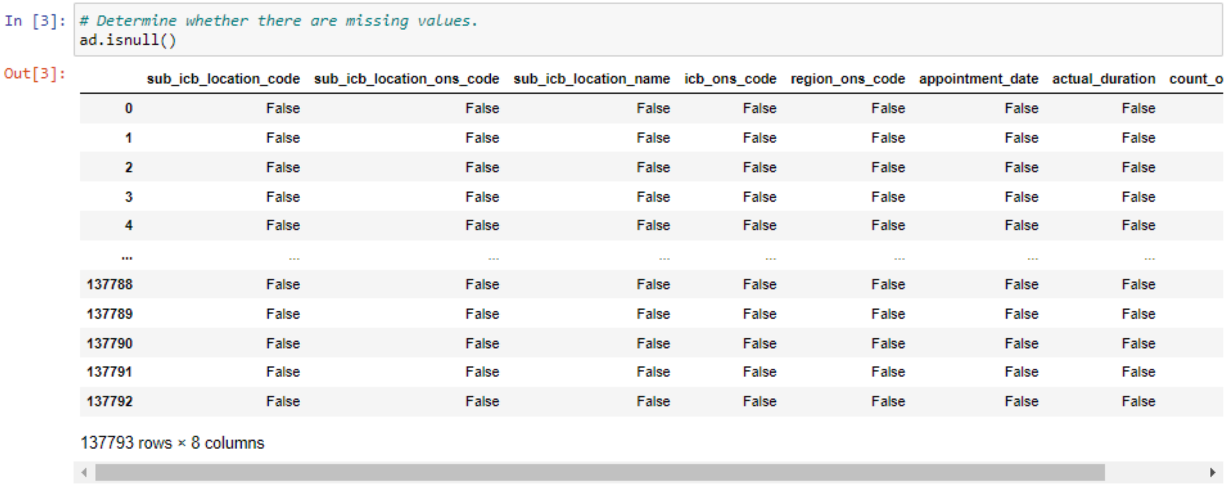
This is an example of examining the shape of the imported file, using the (name of variable).head() command



Out[5]:

| | sub_icb_location_code | sub_icb_location_ons_code | sub_icb_location_name | icb_ons_code | region_ons_code | appointment_date | actual_duration | count_of_app |
|---|---|---|---|---|---|---|---|---|
| 0 | 00L | E38000130 | NHS North East and North Cumbria ICB - 00L | E54000050 | E40000012 | 01-Dec-21 | 31-60 Minutes | |
| 1 | 00L | E38000130 | NHS North East and North Cumbria ICB - 00L | E54000050 | E40000012 | 01-Dec-21 | 21-30 Minutes | |
| 2 | 00L | E38000130 | NHS North East and North Cumbria ICB - 00L | E54000050 | E40000012 | 01-Dec-21 | 6-10 Minutes | |
| 3 | 00L | E38000130 | NHS North East and North Cumbria ICB - 00L | E54000050 | E40000012 | 01-Dec-21 | Unknown / Data Quality | |
| 4 | 00L | E38000130 | NHS North East and North Cumbria ICB - 00L | E54000050 | E40000012 | 01-Dec-21 | 16-20 Minutes | |

To ensure the data is not missing any values, we run the isnull() command that returns the amount of NaN values for us. After reviewing the three files delivered from NHS, they did not contain any missing values.



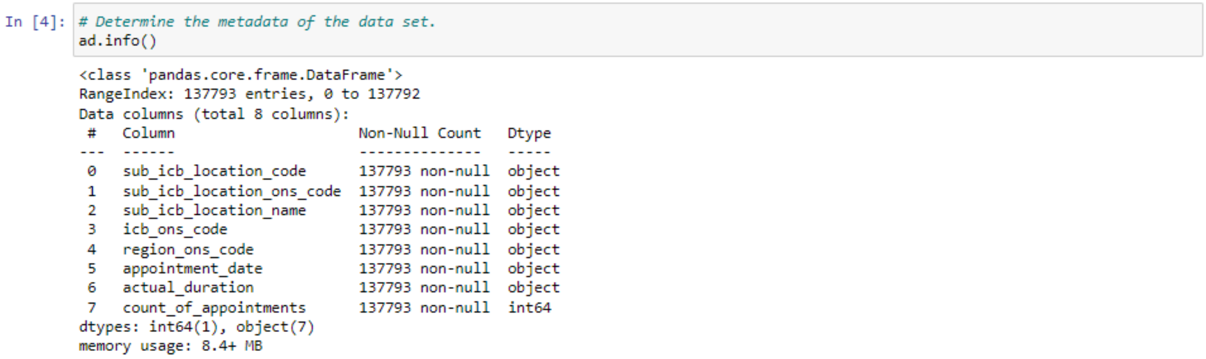
In [3]:
```
# Determine whether there are missing values.
ad.isnull()
```

Out[3]:

| | sub_icb_location_code | sub_icb_location_ons_code | sub_icb_location_name | icb_ons_code | region_ons_code | appointment_date | actual_duration | count_o |
|---|---|---|---|---|---|---|---|---|
| 0 | False | False | False | False | False | False | False | |
| 1 | False | False | False | False | False | False | False | |
| 2 | False | False | False | False | False | False | False | |
| 3 | False | False | False | False | False | False | False | |
| 4 | False | False | False | False | False | False | False | |
| ... | ... | ... | ... | ... | ... | ... | ... | |
| 137788 | False | False | False | False | False | False | False | |
| 137789 | False | False | False | False | False | False | False | |
| 137790 | False | False | False | False | False | False | False | |
| 137791 | False | False | False | False | False | False | False | |
| 137792 | False | False | False | False | False | False | False | |

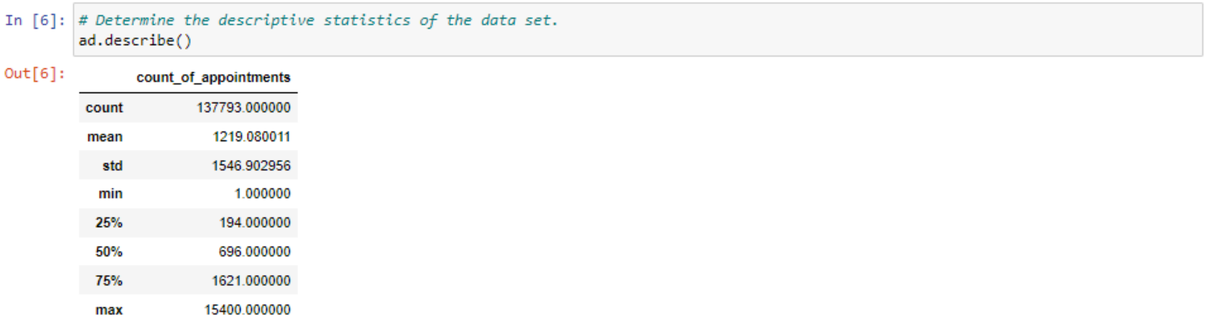
137793 rows × 8 columns

We used the x.describe() and x.info() commands to determine the meta data and the statistics of the data, this is useful for determining the shape of our data frame and where we want to explore deeper.



In [4]:
```
# Determine the metadata of the data set.
ad.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 137793 entries, 0 to 137792
Data columns (total 8 columns):
 #   Column                     Non-Null Count   Dtype 
---  ------                     --------------   ----- 
 0   sub_icb_location_code      137793 non-null  object
 1   sub_icb_location_ons_code  137793 non-null  object
 2   sub_icb_location_name      137793 non-null  object
 3   icb_ons_code               137793 non-null  object
 4   region_ons_code            137793 non-null  object
 5   appointment_date           137793 non-null  object
 6   actual_duration            137793 non-null  object
 7   count_of_appointments      137793 non-null  int64 
dtypes: int64(1), object(7)
memory usage: 8.4+ MB
```



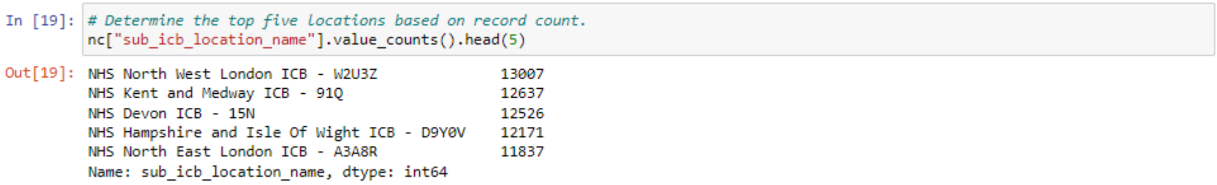
In [6]:
```
# Determine the descriptive statistics of the data set.
ad.describe()
```

Out[6]:

| | count_of_appointments |
|---|---|
| count | 137793.000000 |
| mean | 1219.080011 |
| std | 1546.902956 |
| min | 1.000000 |
| 25% | 194.000000 |
| 50% | 696.000000 |
| 75% | 1621.000000 |
| max | 15400.000000 |

One example is identifying the top five locations based on record count, in the previous step we determined that columns. When we imported the excel file it automatically converted into a data frame (note the above screenshots use the AD file and data frame, moving forward we will use NC (National Categories data frame)).

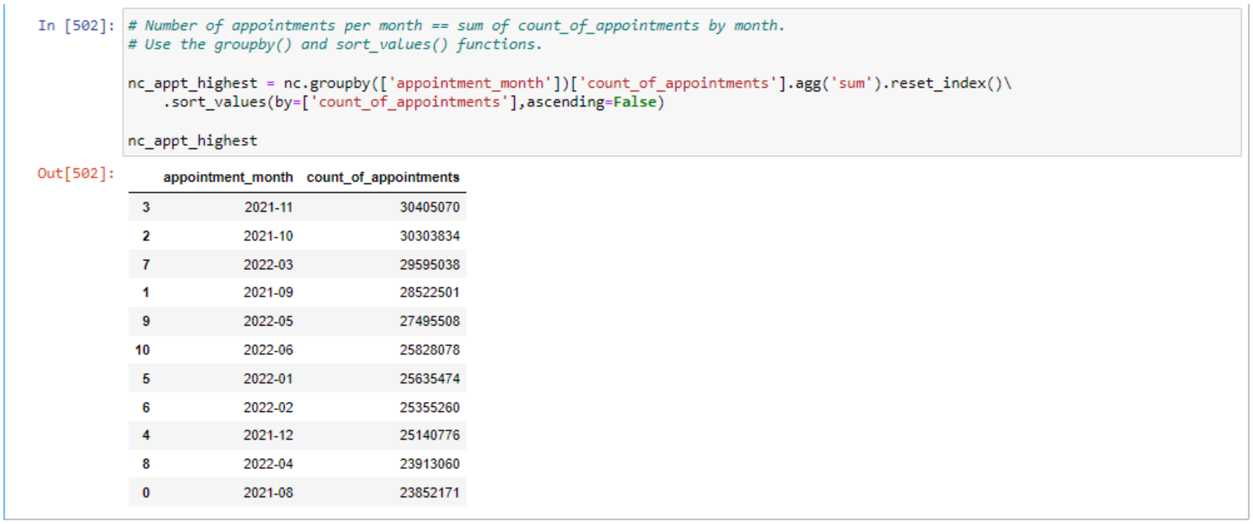
We can identify the top five locations leveraging a command called .value_counts() and head which returns the top five.



```
In [19]: # Determine the top five locations based on record count.
         nc["sub_icb_location_name"].value_counts().head(5)

Out[19]: NHS North West London ICB - W2U3Z             13007
         NHS Kent and Medway ICB - 91Q                 12637
         NHS Devon ICB - 15N                           12526
         NHS Hampshire and Isle Of Wight ICB - D9Y0V   12171
         NHS North East London ICB - A3A8R             11837
         Name: sub_icb_location_name, dtype: int64
```

The majority of the project is focused on deep diving into certain columns leveraging Python, we either use the main data frame or we create sub data frames to exclude certain criteria e.g between specific dates, or a particular service of interest. After the creation of the sub data frames, we either use the groupby() and aggregate sum commands to determine the numbers on specific questions; such as the count of appointments per month.



```
In [502]: # Number of appointments per month == sum of count_of_appointments by month.
          # Use the groupby() and sort_values() functions.

          nc_appt_highest = nc.groupby(['appointment_month'])['count_of_appointments'].agg('sum').reset_index()\
              .sort_values(by=['count_of_appointments'],ascending=False)

          nc_appt_highest
```

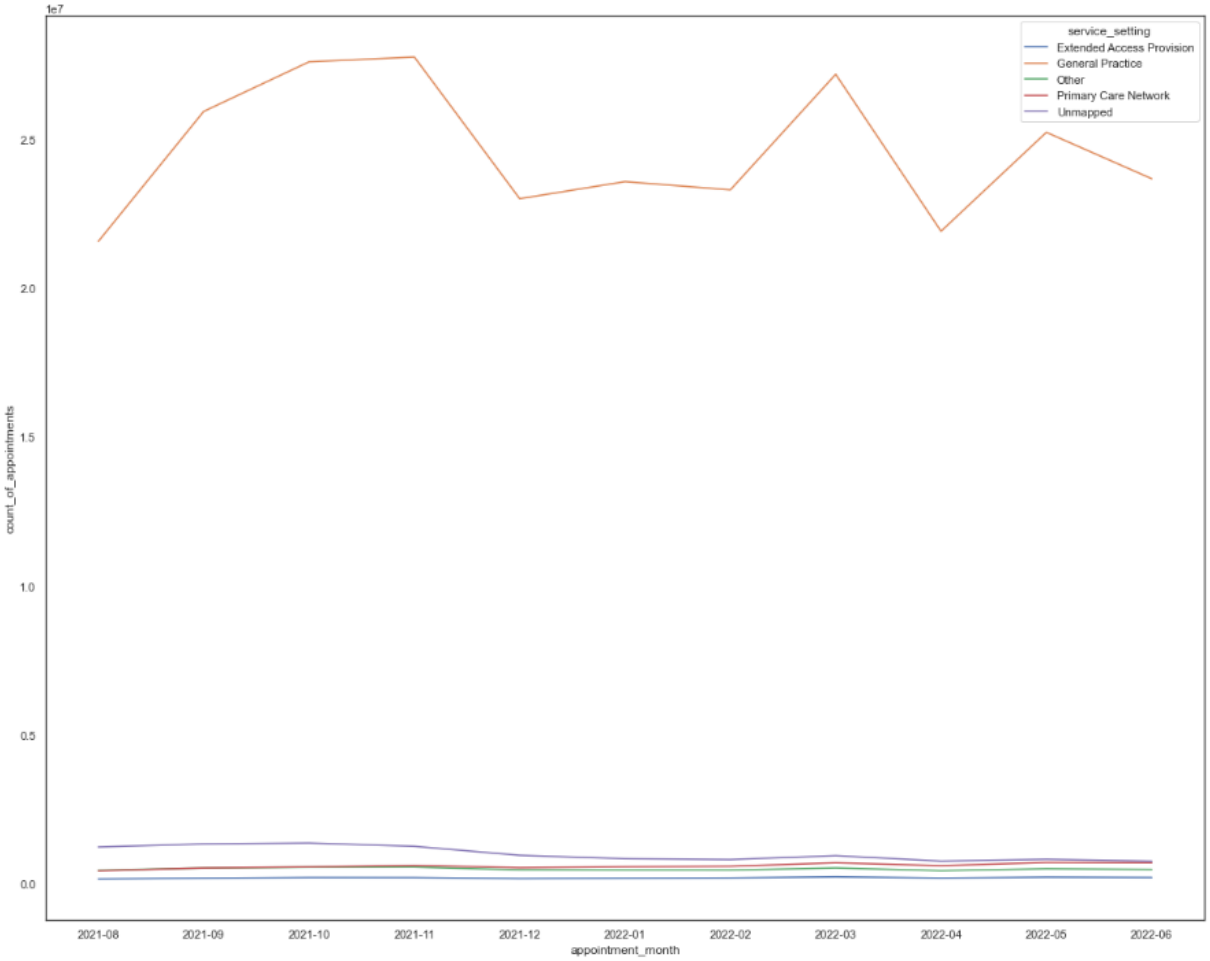
Out[502]:

| | appointment_month | count_of_appointments |
|---|---|---|
| 3 | 2021-11 | 30405070 |
| 2 | 2021-10 | 30303834 |
| 7 | 2022-03 | 29595038 |
| 1 | 2021-09 | 28522501 |
| 9 | 2022-05 | 27495508 |
| 10 | 2022-06 | 25828078 |
| 5 | 2022-01 | 25635474 |
| 6 | 2022-02 | 25355260 |
| 4 | 2021-12 | 25140776 |
| 8 | 2022-04 | 23913060 |
| 0 | 2021-08 | 23852171 |

## 3. Visualization and Insights

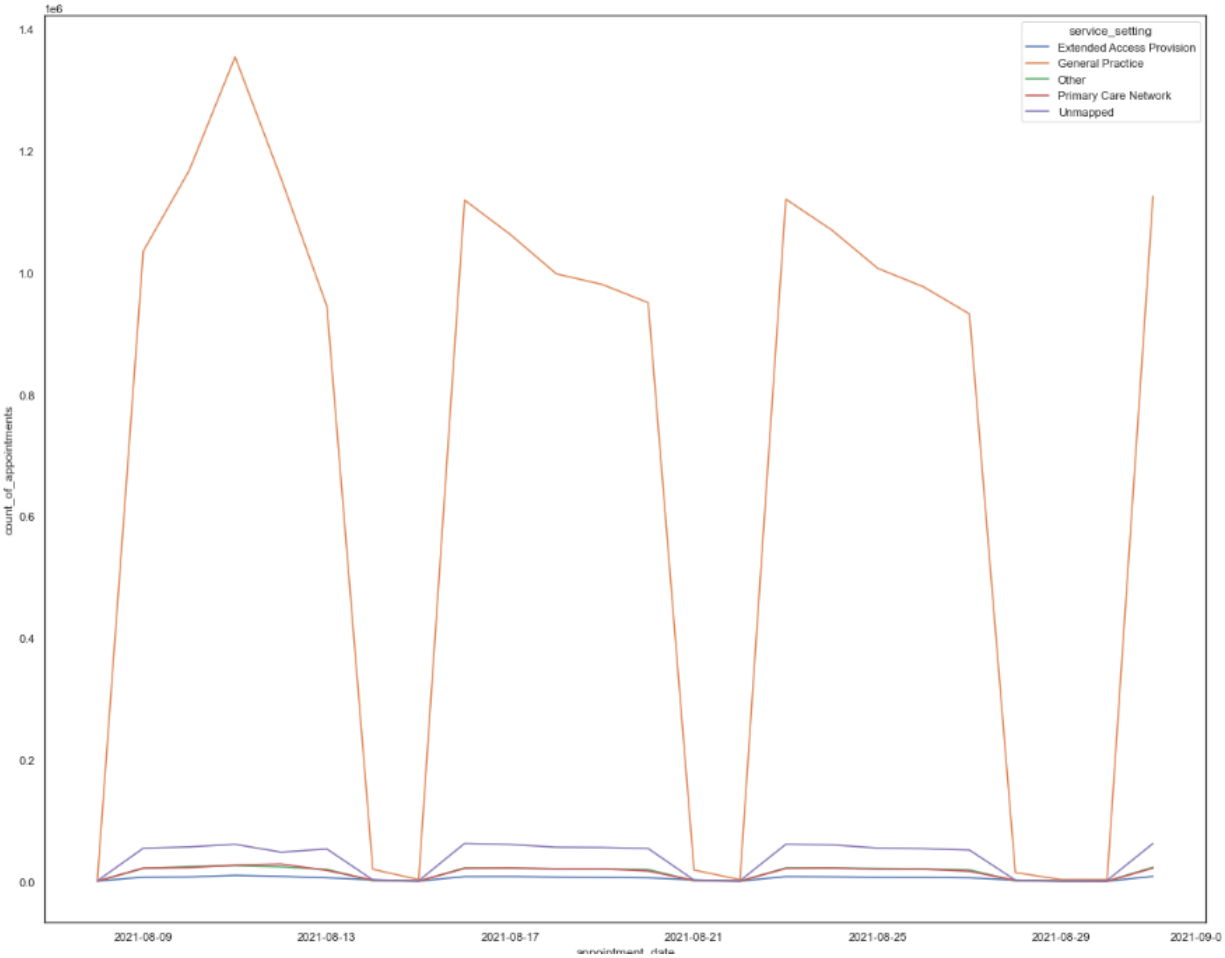
The primary visualizations are line graphs, bar graphs and box plot. The key rationale for these is due to the nature of the data and what its presenting, as line graphs make it easier to see a direct trend over a period of time. Bar graphs assist with direct comparison to recognize patterns and trends far more easily than looking at a table of numerical data. Box plots divide the data into sections it allows us to quickly identify mean values, signs of skewness and dispersion of the data set.

**Visualization 1: Service Settings over appointment months (Month example)**



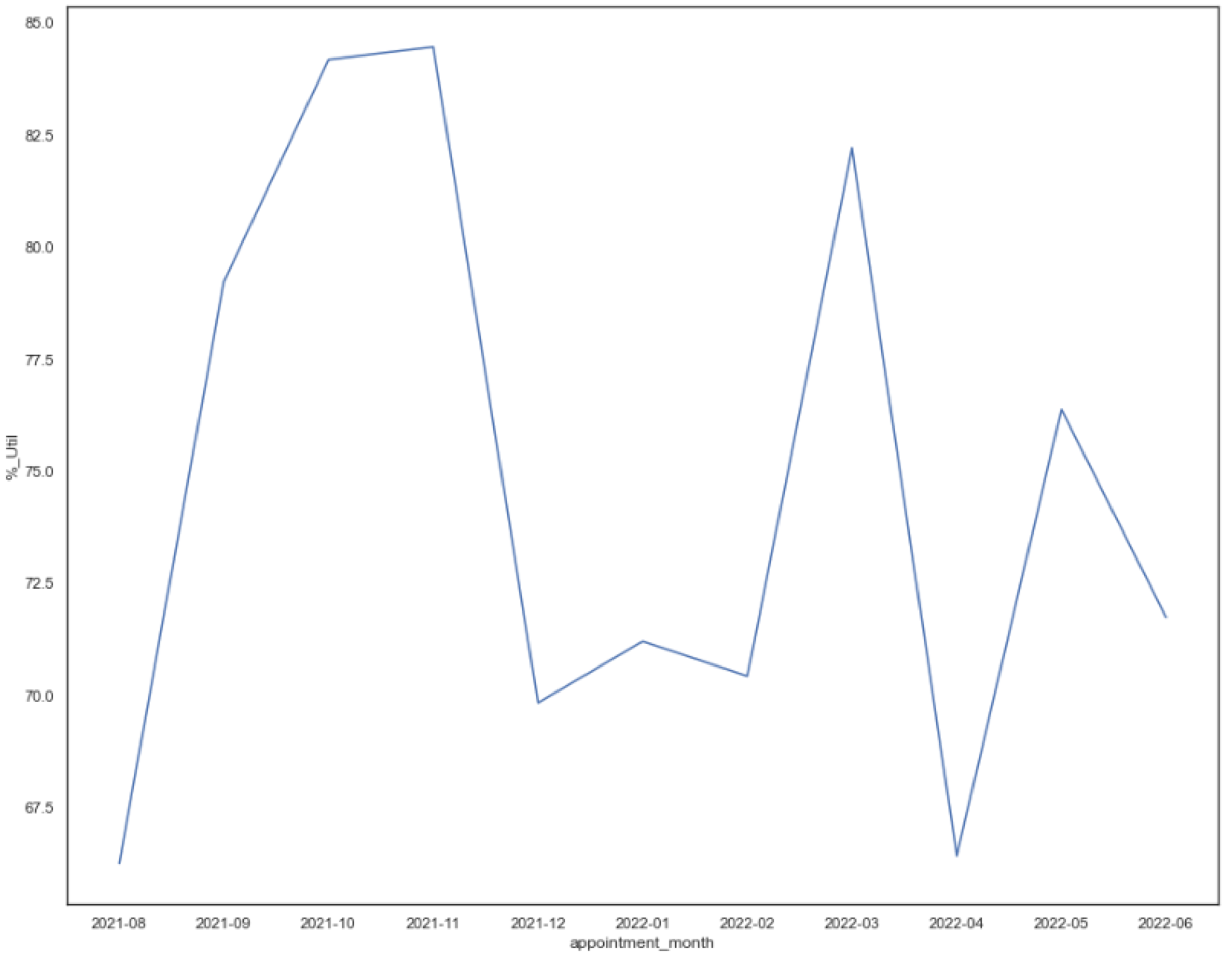
General Practice is the leading service of choice, followed by Extended Access Provision. Other and Primary Care Network are close to each other, while unmapped is the lowest. There was a decline in growth for the other service settings, and significant growth for General Practice (from 2.2b to 2.4b~).

**Visualization 2: Service Settings over appointment months (Month example)**

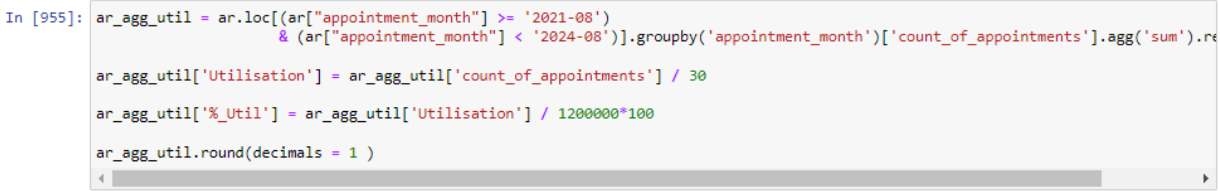


This is an example of a day-by-day chart with Seaborn Python, from this chart we can see that appointments are slowly decreasing as the month continues. Weekends show no sign of activity, assume the clinic closes during the weekend. The clinics are busy at the start of the week and slow down.

**Visualization 3: Utilization chart**



In python, we added an additional column that calculated the utilization based on the daily count. NHS stated their daily max is 1,200,000 appointments, we divided the monthly by 30 and then used that value against 1,200,000 to determine the utilization on a monthly basis. From this chart, the highest is 84% (October 2021) with the lowest being 65% (April 2022). NHS is not understaffed but they can benefit from internal optimization.



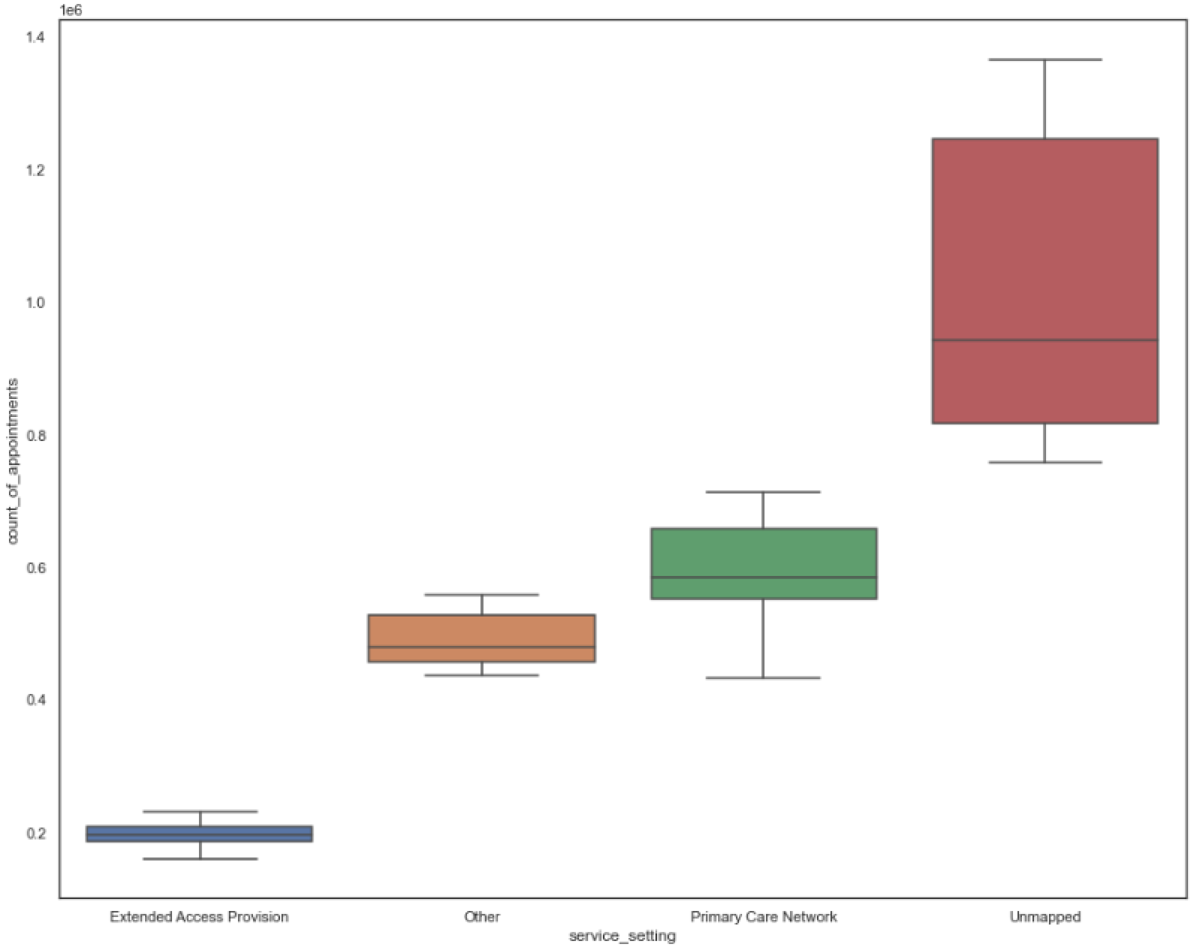
```
In [955]: ar_agg_util = ar.loc[(ar["appointment_month"] >= '2021-08')
                              & (ar["appointment_month"] < '2024-08')].groupby('appointment_month')['count_of_appointments'].agg('sum').re

          ar_agg_util['Utilisation'] = ar_agg_util['count_of_appointments'] / 30

          ar_agg_util['%_Util'] = ar_agg_util['Utilisation'] / 1200000*100

          ar_agg_util.round(decimals = 1 )
```

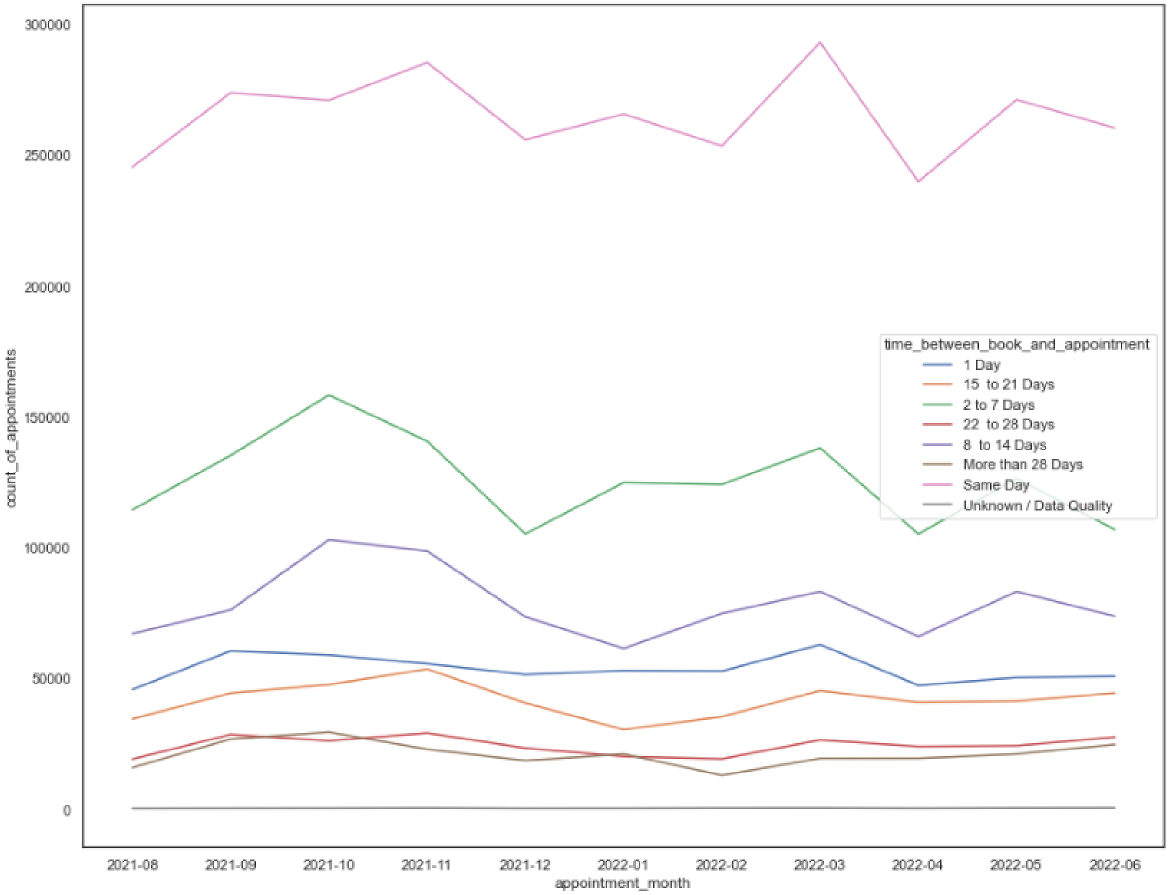
**Visualization 4:** Box Plot of Service Settings vs Count of Appointments



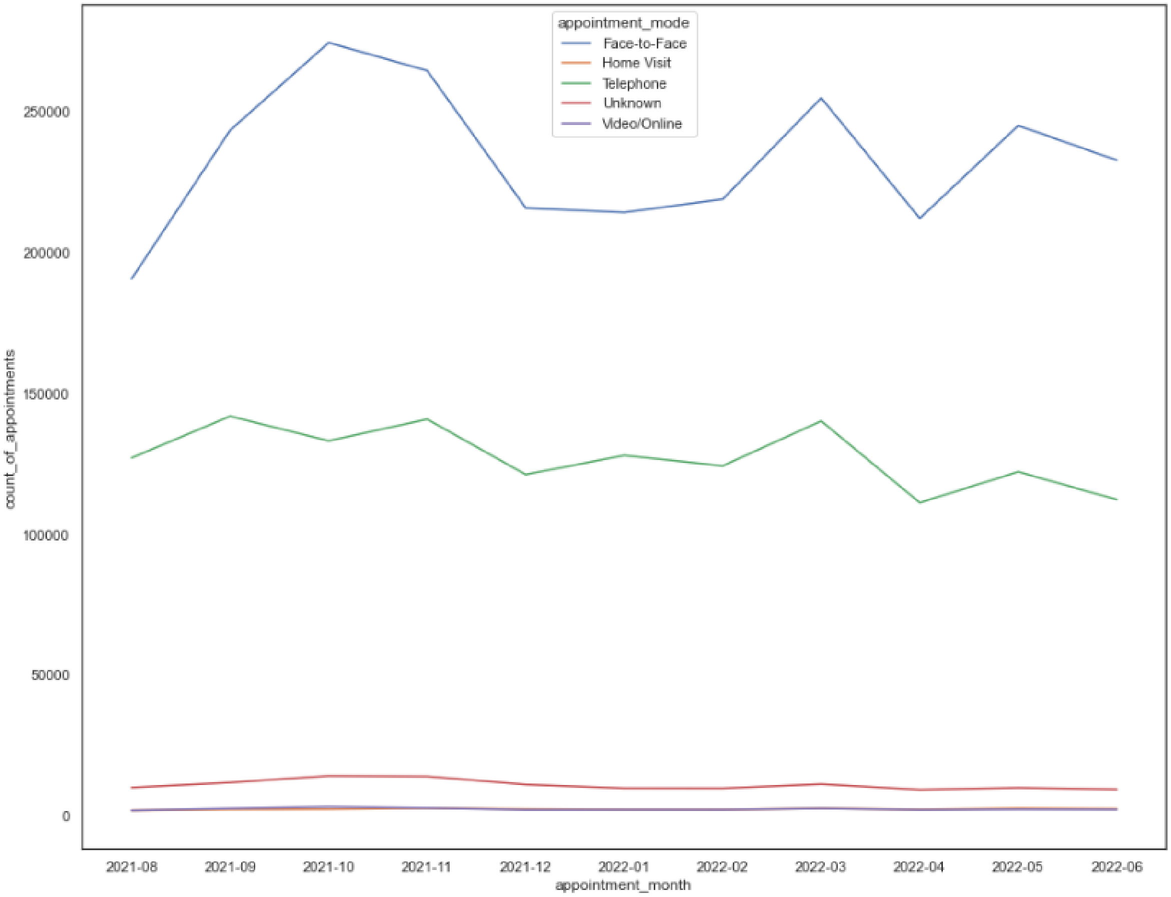
Excluded General Practice due to it distorting the box plot, from this we can determine that Extended Access Provision and Other have less outlier ranges compared to Primary Care Network. Albeit Unmapped at a large range and median is concerning, this implies that the internal processes and policies require an improvement.

## 4. Pattern Predictions

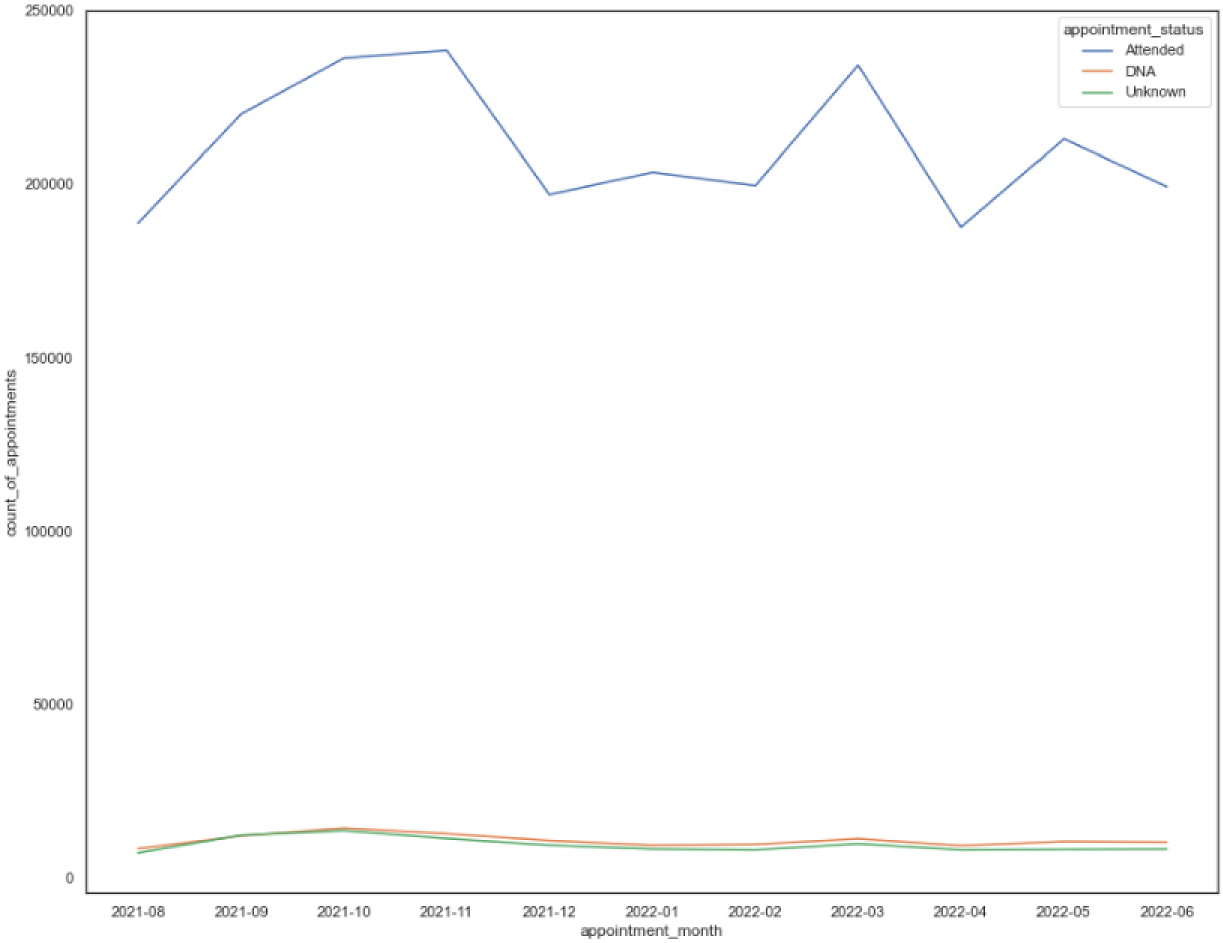
From a high-level observation of the trends, General Practice is the most popular service all the other services make up less than 5% of the total. Patients prefer to book and attend their appointment on the same day, followed by up to 2-7 days, 8 to 14 days and 1 Day with the rest following closely behind

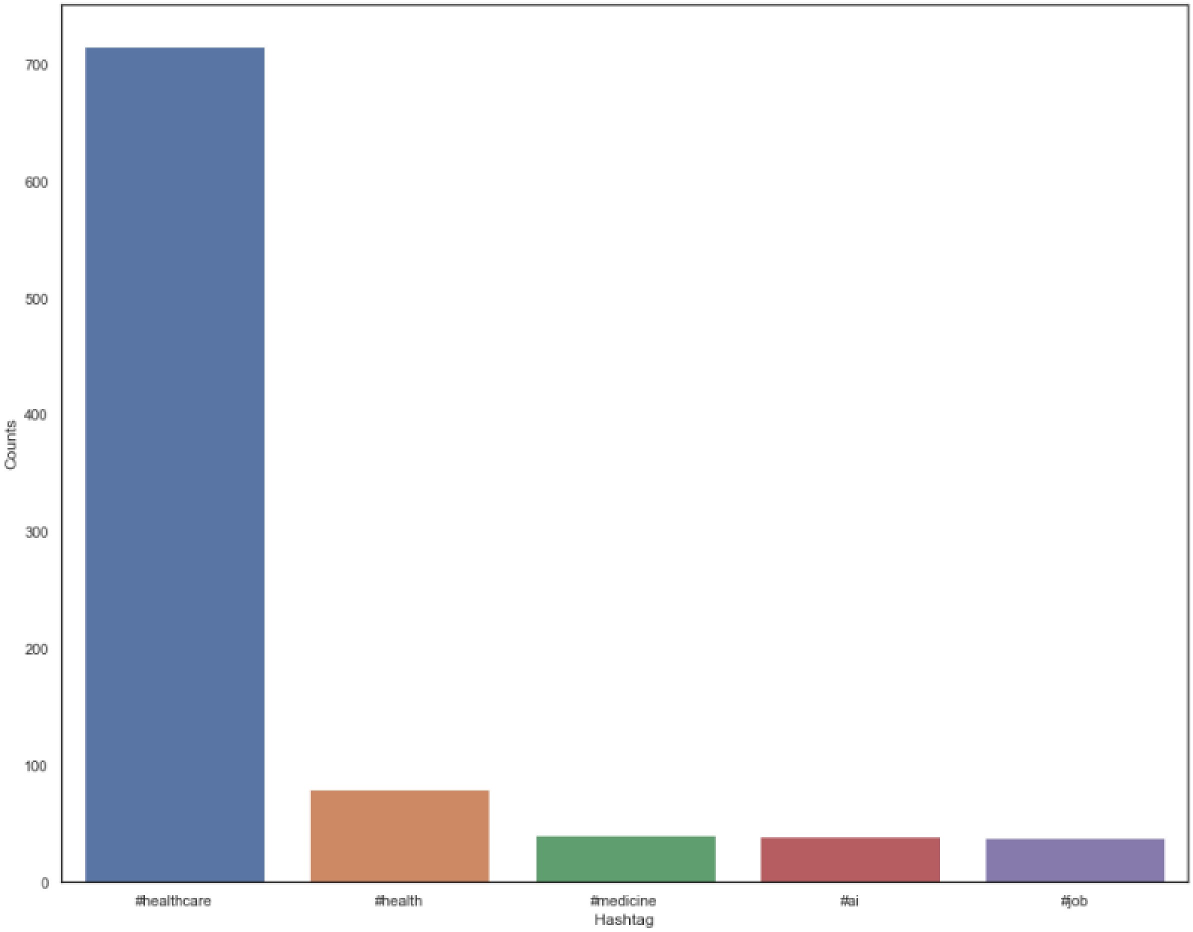


Face-to-Face are the main preferred mode followed by Telephone. Home Visit and Video Online are not as popular.



One of the core concerns were patients not attending, the data shows that there is a slow increase in DNA (Did not attend) and a decent increase in attendance. If we can funnel out the unknown data by implementing better internal controls, we can determine the accurate numbers.





Twitter Analysis indicates that #healthcare, #health, #medicine, #ai, #jobs. The primary discussion was how covid affected the world, and how medicine may assist further. The use of AI to determine the usefulness of particular job functions.

**To answer the two main concerns of NHS:**

- **Has there been adequate staff and capacity in the networks?**
  From the data, there has been adequate staff and the capacity primarily focuses on General Practice
- **What was the actual utilization of resources?**
  The highest is 84% (October 2021) with the lowest being 65% (April 2022). The rest is within that range and never exceeding 100%

**Suggestions:** To focus on the optimization of the internal data control, there are alarming rates of unknown or uncategorized data which may imply a weak data internal system.

Discuss about the other pillars outside of General Practice, are there any areas of expansion worth looking into? We know the patients' preferences are Face to Face contact and same day appointments.