NHS Technical Report

Leonard Okhani

Background

The NHS (National Health Services) is a publicly funded healthcare system located in England. They are concerned about the services' utilisation, missed appointments and potential in using external data sources such as Twitter (currently known as X). Additionally, there is a debate of whether the NHS should increase their resources to match their growing population or to just make use of what they already have. The goal of this project is to answer the 2 biggest questions posed by the NHS:

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

Analytic Approach

With the implementation of Python, the four following datasets were analysed:

- actual_duration.csv – Details of appointments made (date, duration, number of appointments per class, region).
- appointments_regional.csv – Details of the type of appointments made (appointment mode, appointment month, appointment status, duration between booking and appointment, healthcare professional, number of appointments per class, region).
- national_categories.xlsx – Details of the national categories of appointments (appointment date, context type, national category, number of appointments per class, region, service setting).
- tweets.csv – Data regarding the UK healthcare, which was taken from Twitter.

Importing and exploring data

The datasets were imported into a Jupyter workbook. The first 3 datasets were converted into DataFrames – ad, ar & nc. They were sense-checked with:

- df.isnull().sum() to determine sum of missing values.
- df.info() for names of columns & data types.
- df.describe() for descriptive statistics.
- df.duplicated().sum() to determine sum of duplicate values.
- df.shape[] to determine number of rows and columns.

Proceeded to use the value_counts() and print() methods to work out these questions:

1. How many locations are there in the data set?

```
# Determine the number of locations.
nc_locations = nc['sub_icb_location_name'].value_counts()
print('Count of locations:', nc_locations.size)
```

```
Count of locations: 106
```

2. What are the five locations with the highest number of appointments?

```
# Determine the top five locations based on record count.
nc_toplocations = nc['sub_icb_location_name'].value_counts()
print('The five locations with the highest number of appointments:', nc_toplocations.head(5))
```

```
The five locations with the highest number of appointments: sub_icb_location_name
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
```

3. How many service settings, context types, national categories and appointment statuses are there?

```
# Determine the number of service settings.
nc_ss = nc['service_setting'].value_counts()
print('Count of service settings:', nc_ss.size)
```

```
Count of service settings: 5
```

```
# Determine the number of context types.
nc_ct = nc['context_type'].value_counts()
print('Count of context types:', nc_ct.size)
```

```
Count of context types: 3
```

```
# Determine the number of national categories.
nc_nc = nc['national_category'].value_counts()
print('Count of national categories:', nc_nc.size)
```

```
Count of national categories: 18
```

```
# Determine the number of appointment statuses.
ar_as = ar['appointment_status'].value_counts()
print('Count of appointment statuses:', ar_as.size)
```

```
Count of appointment statuses: 3
```

Analysing the data

First part of the analysis was checking the DataFrames for insights regarding the appointments' dates. These are additional questions NHS want answered:

1. Between what dates were appointments scheduled?

```
# Determine the minimum and maximum dates in the ad DataFrame.
# Use appropriate docstrings.
ad['appointment_date'].agg(['min', 'max'])
print(min(ad['appointment_date']), ': the minumum date')
print(max(ad['appointment_date']), ': the maximum date')
```

```
2021-12-01 00:00:00 : the minumum date
2022-06-30 00:00:00 : the maximum date
```

```
# Determine the minimum and maximum months in the ar DataFrame.
# Use appropriate docstrings.
ar_clean['appointment_month'].agg(['min', 'max'])
print(min(ar_clean['appointment_month']), ': the minimum month')
print(max(ar_clean['appointment_month']), ': the maximum month')
```

```
2020-01 : the minimum month
2022-06 : the maximum month
```

```
# Determine the minimum and maximum dates in the nc DataFrame.
# Use appropriate docstrings.
nc['appointment_date'].agg(['min', 'max'])
print(min(nc['appointment_date']), ': the minumum date')
print(max(nc['appointment_date']), ': the maximum date')
```

```
2021-08-01 00:00:00 : the minumum date
2022-06-30 00:00:00 : the maximum date
```

2. Which service setting was the most popular for NHS North West London from 1st January to 1st June 2022? **General Practice.**

```
# For each of these service settings, determine the number of records available for the period and the location.
nc_subset = nc[nc['sub_icb_location_name'] == 'NHS North West London ICB - W2U3Z']
nc_subset_daterange = nc_subset[(nc_subset['appointment_date'] >= '2022-01-01') &
                                (nc_subset['appointment_date'] <= '2022-06-01')]

# View the output.
nc_subset_daterange['service_setting'].value_counts()
```

```
service_setting
General Practice             2104
Other                        1318
Primary Care Network         1272
Extended Access Provision    1090
Unmapped                      152
```

3. Which month had the highest number of appointments? **November.**

```
# Number of appointments per month == sum of count_of_appointments by month.
# Use the groupby() and sort_values() functions.
months_appointments_sum = nc.groupby('appointment_month')['count_of_appointments'].sum()
months_appointments_sum_sort = months_appointments_sum.sort_values(ascending=False)
print(months_appointments_sum_sort, "Number of appointments per month:")
```

```
appointment_month
2021-11    30405070
2021-10    30303834
2022-03    29595038
2021-09    28522501
2022-05    27495508
2022-06    25828078
2022-01    25635474
2022-02    25355260
2021-12    25140776
2022-04    23913060
2021-08    23852171
```

4. What was the total number of records per month?

```
# Total number of records per month.
months_appointments_count = nc.groupby('appointment_month')['count_of_appointments'].count()
months_appointments_count_sort = months_appointments_count.sort_values(ascending=False)
print("Total number of records per month:", months_appointments_count_sort)
```

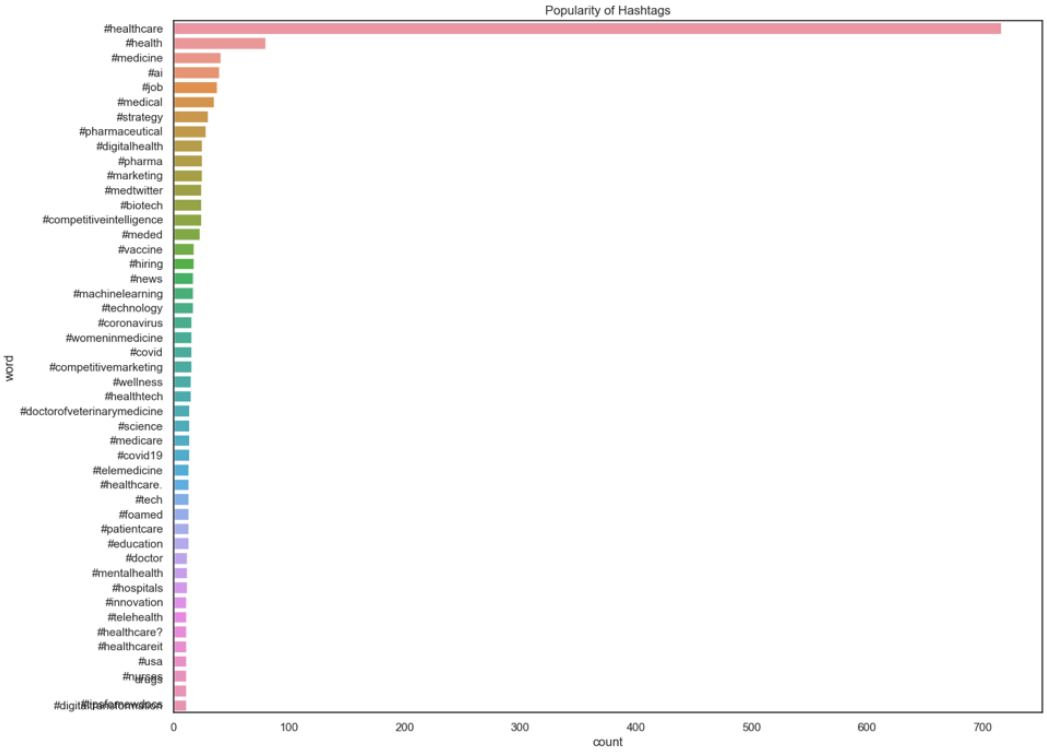
```
Total number of records per month: appointment_month
2022-03    82822
2021-11    77652
2022-05    77425
2021-09    74922
2022-06    74168
2021-10    74078
2021-12    72651
2022-01    71896
2022-02    71769
2022-04    70012
2021-08    69999
```

Second part of the analysis was inspecting tweets.csv. It was important to see the effects Twitter had on the NHS as shown in recent years, social media has been used to share stories and build a community within the public. Started by exploring the tweet_retweet_count and tweet_favorite_count columns with the value_counts() function to see the most retweeted and favourited tweets respectively. This was useful as it will support the NHS to see the types of tweets which speak to their community.

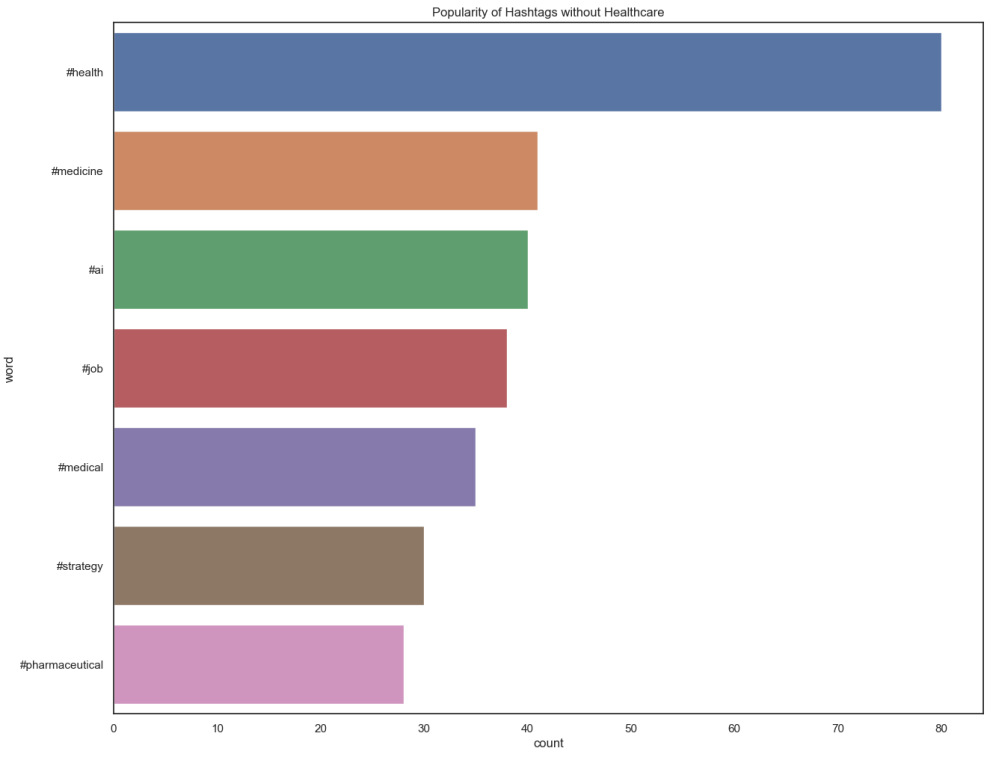
A new DataFrame (tweets_text) and variable (tags) were created. The following code was used to inspect the messages and create a list of values, which contain the # symbol.

```
for y in [x.split(' ') for x in tweets['tweet_full_text'].values]:
    for z in y:
        if '#' in z:
            # Change to lowercase.
            tags.append(z.lower())
```

For visualisation, tags was converted into a DataFrame to produce a Seaborn barplot - showcasing all the hashtags which were used over 10 times.



As '#healthcare' was the most popular hashtag, another barplot was created to see which other hashtags (count between 25 and 100) were trending.



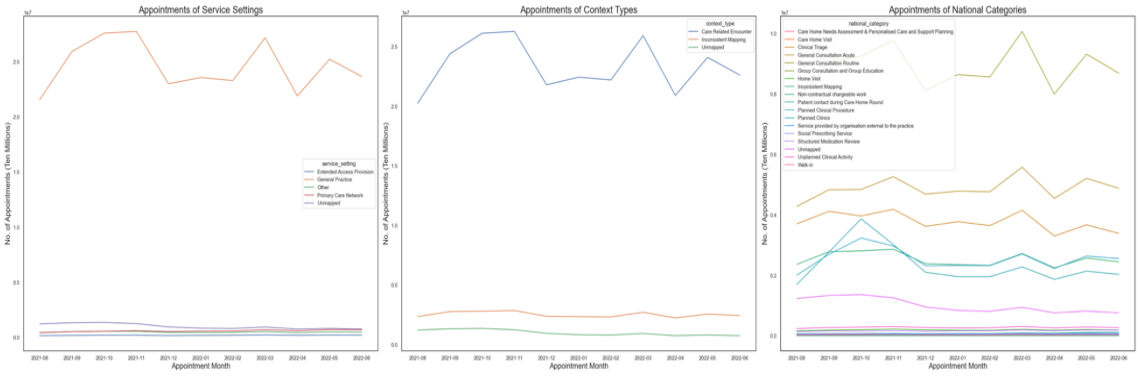
Visualisations and Insights

Many visualisations were created to uncover useful insights for the NHS.

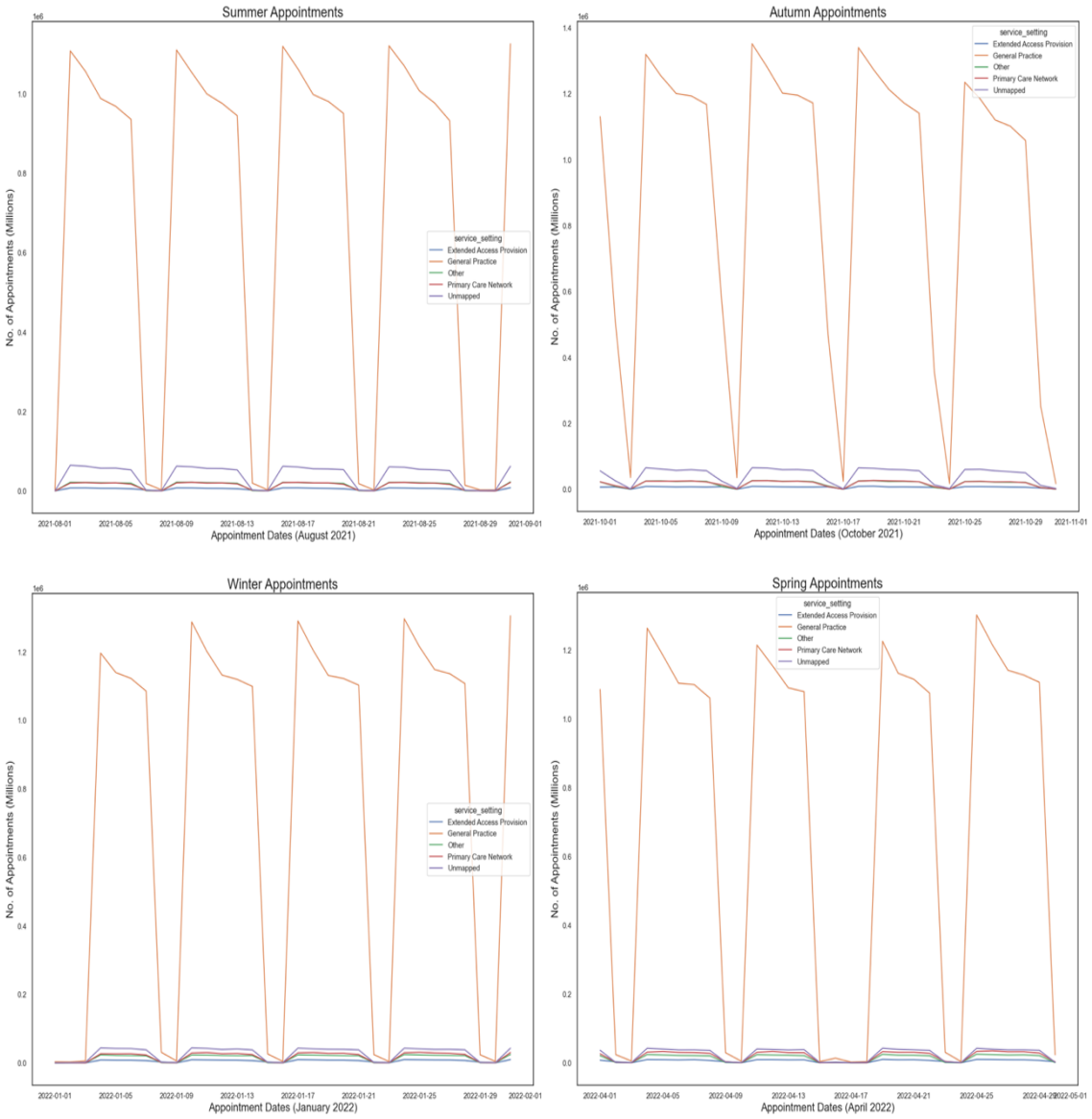
Visualising and identifying initial trends

First objective was to create 3 visualisations to showcase the number of appointments per month for service settings, context types and national categories. Started by changing the data type of appointment_month to string for easier visualisation. For each category, a new DataFrame was created and the groupby() function was applied to group the monthly appointments. Used the sum().reset_index() to sum the number of appointments as well as reset the index. As for the lineplot:

- The x and y variables were specified.
- The category and DataFrame were assigned to the hue and data respectively.
- The confidence interval (ci) was set to 0.



Second objective was to create 4 visualisations to show the number of appointments per service setting each season. The seasons were Summer (August), Autumn (October), Winter (January) and Spring (April). Followed a similar process to the first objective when creating the DataFrames.



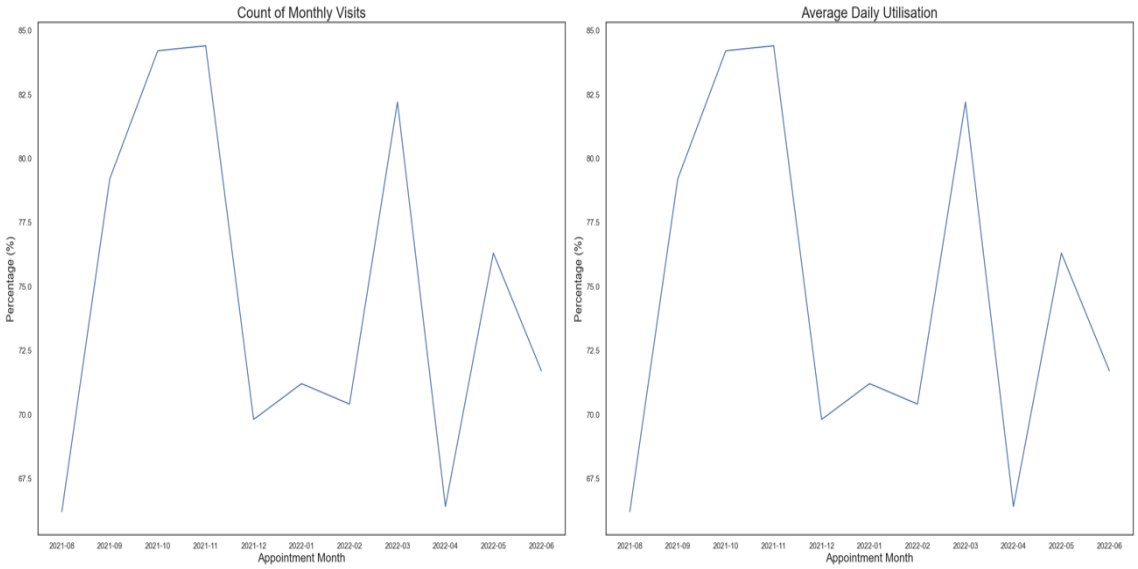
The 4 charts have shown to all have similar patterns – particularly with their peaks being at the beginning of the weeks and going down as the week progresses. In addition, they all share the common fact of 'General Practice' being the most popular service setting. Furthermore, in terms of the most popular season, Autumn is the one with the most appointments while the season with the least number of appointments is Summer.

Patterns and Recommendations

To work out the patterns, the ar dataset was revisited and filtered to only show dates from August 2021 to June 2022. By answering the questions below, recommendations will be made for NHS to make sure they are reducing the number of missed appointments.

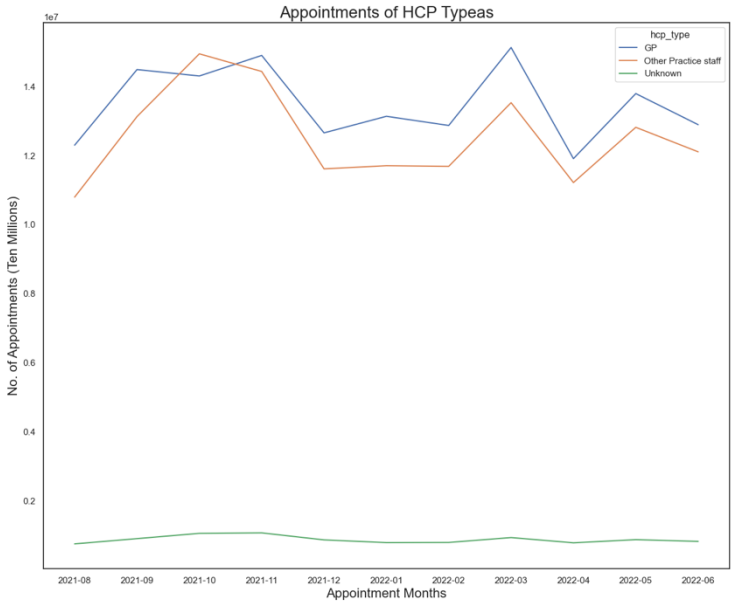
Patterns

1. Should the NHS start looking at increasing levels?



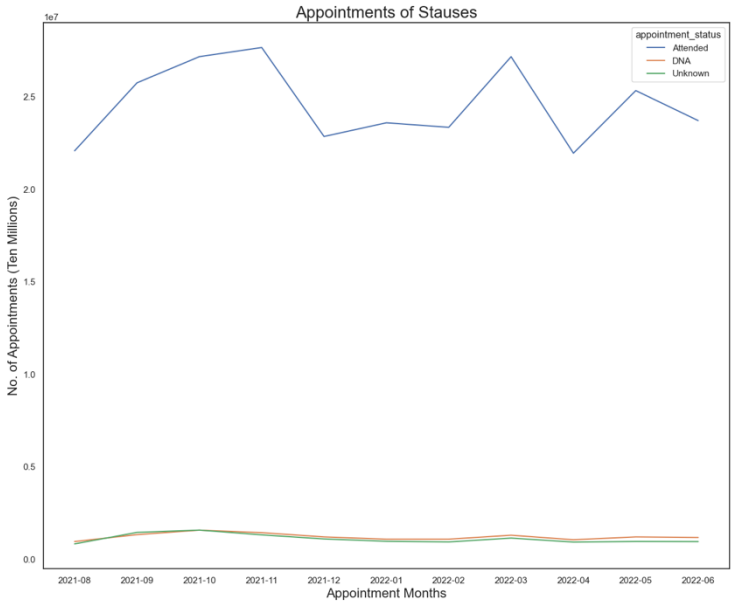
The NHS have a maximum capacity of 1,200,000 appointments per day. November was the month with the most appointments and reached 84% of the capacity – showing that the NHS have a sufficient capacity.

2. How do the healthcare professional types differ over time?



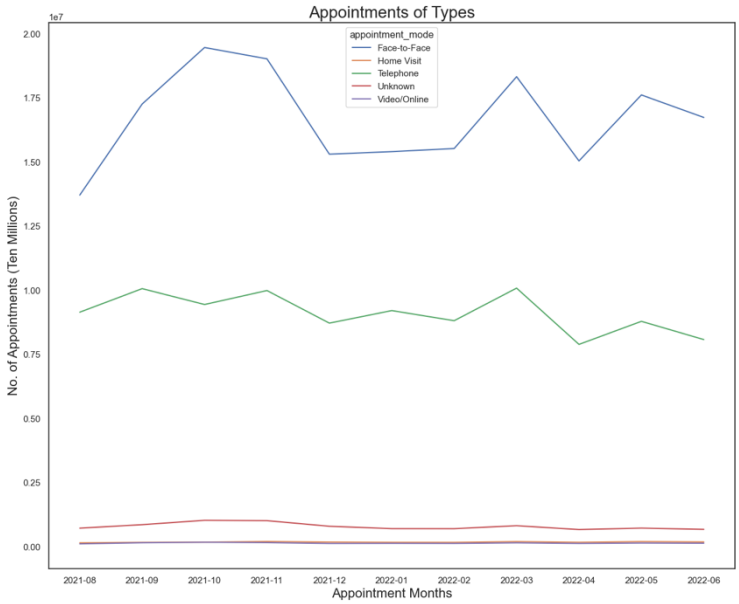
Excluding October, 'GP' always had more appointments that 'Other Practice staff' while 'Unknown' remained under 2 million throughout the period.

3. Are there significant changes in whether visits are attended?



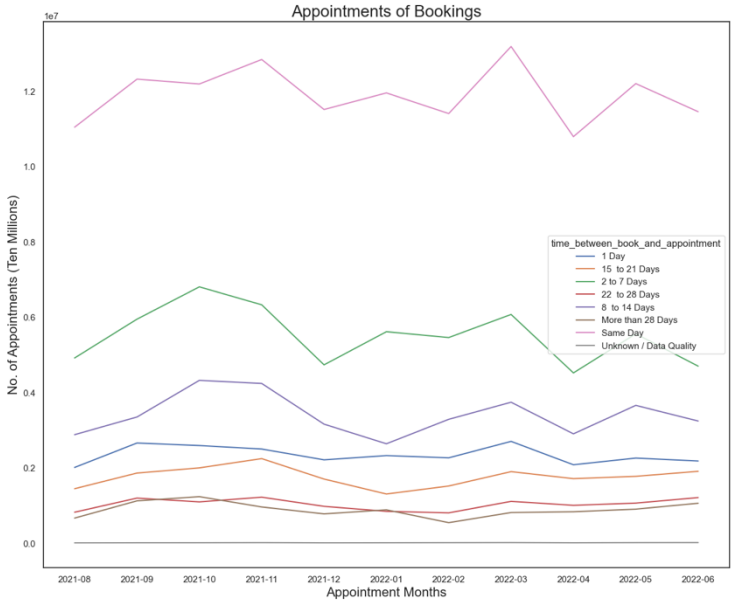
Many visits were attended, and the pattern even resembles the chart for overall monthly appointments.

4. Are there changes in terms of appointment type and the busiest months?



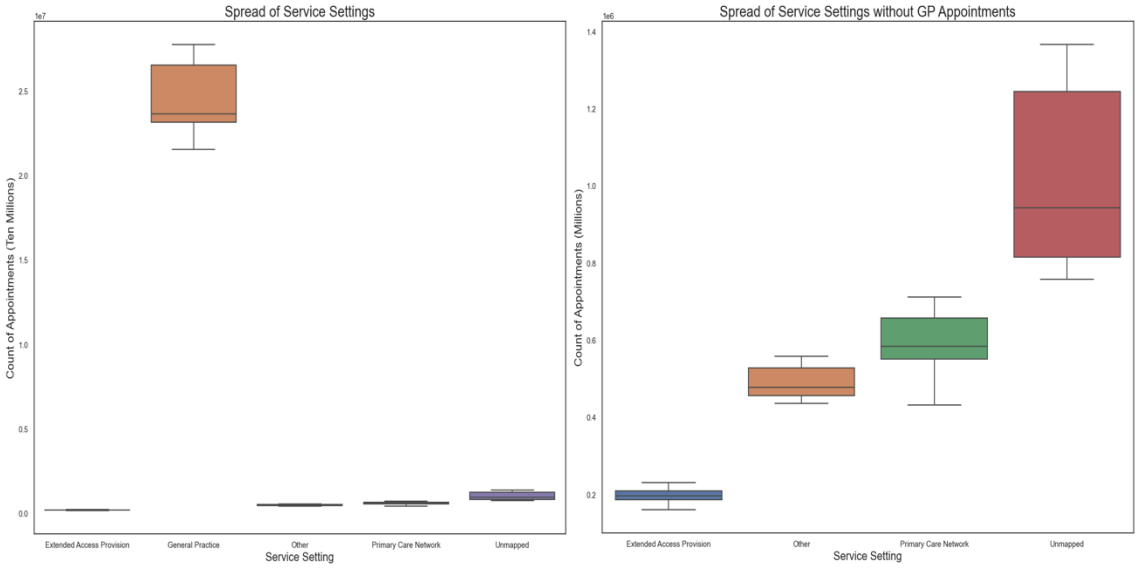
The busiest months were October, November and March, and like with the other months, 'Face-to-Face' was the most popular appointment type and followed the same pattern with the overall monthly appointments.

5. Are there any trends in time between booking and appointment?



Many patients had their appointments on the day they booked while others waited for a couple of days to a week.

6. How do the various service settings compare?



Once again, 'General Practice' was the most popular service setting. Without 'GP', the most popular is 'Unmapped', which means that the NHS need to manage their data better.

Recommendations

After the analysing and uncovering insights, these are the recommendations to help support the NHS:

- Reminders are sent to the patients by either call, text or email at least a day before the appointment, especially if they need to change or cancel the appointment.
- Patients are aware of the consequences of unattended appointments.
- Increase hub hours during the weekends.
- Check that the appointments are assigned category data.
- Confirm the patients know what they're going in for.