NHS's Capacity and Staff Adequacy Analysis

Study based on data from 01.01.2020 – 30.06.2022

By: Alexandra Akaoui

Presented on: 23.05.2025

Table of Contents

Introduction and Business Context

Analysis Briefing

The NHS is concerned about the adequacy of its current staff and the capacity of its network in the face of a growing and aging population.

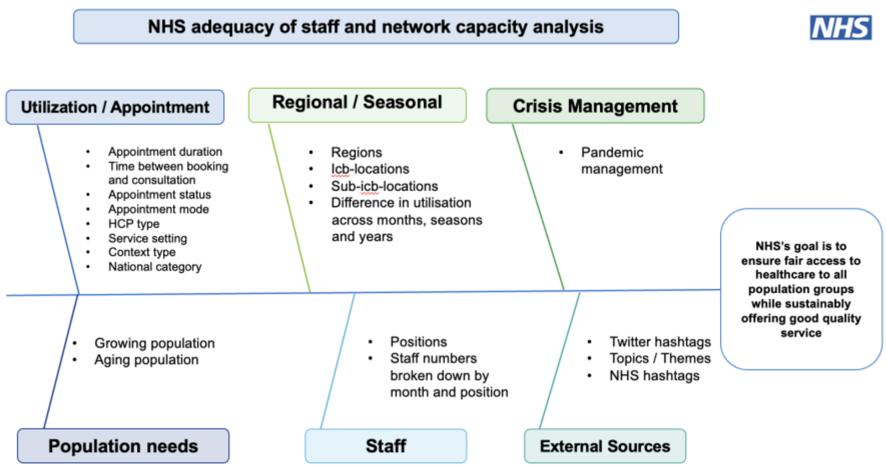
It is also faced by a significant financial burden resulting from no-show (DNA/ Did not attend) appointments.

Business Problem Statement

To ensure equitable and sustainable access to healthcare services, the NHS aims to reach a data-backed answer to the following:

- An understanding of the size of the network and trends broken down by locations, service settings, context types, national categories, appointment modes, durations, and status
- Seasonal/Regional trends
- The actual utilization of the network
- The response of the network in the face of a crisis (Covid)
- The population needs and staff adequacy in numbers
- The usefulness of relying on an external source like Twitter

Analytical Framework



Analytical Exploratory Approach

Preliminary Steps

- The following steps were taken:
- Import of the external files to complement the data
- Check and adjustments of column data types and null values.
- Extraction and average calculation of data from appointment duration and time between booking and consultation columns in the actual duration (ad) and appointments regional (ar) DataFrames
- Pivoting and grouping of the DataFrames based on certain columns, to ensure that the data shown on the charts are for appointment counts and not records.
- Check for commonalities in the DataFrames.

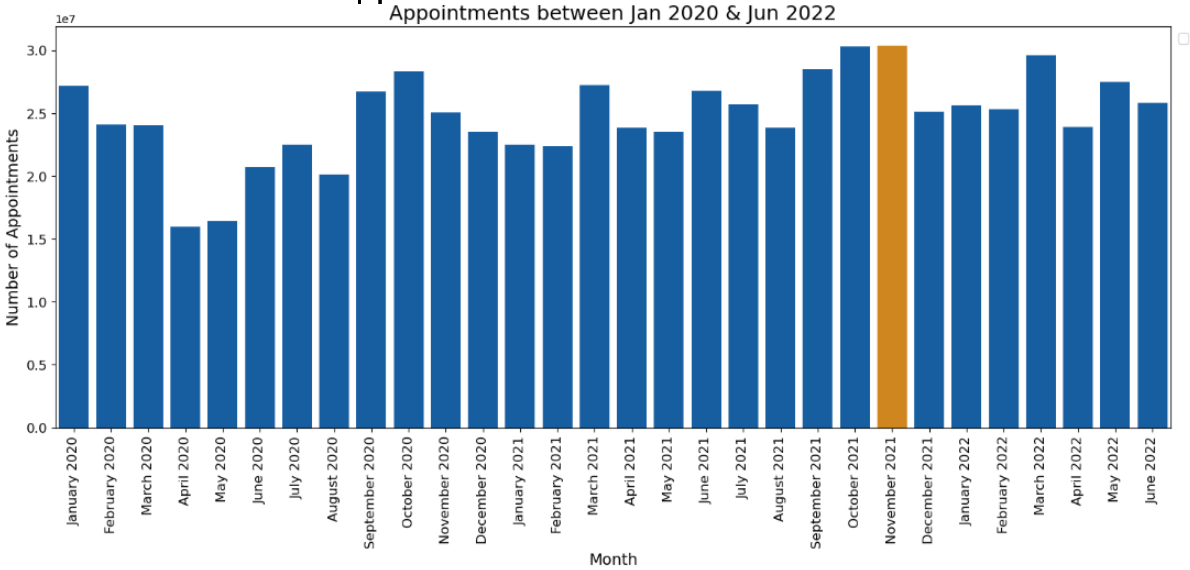
Analysis Areas

The following areas were explored:

Appointment

Appointment Number

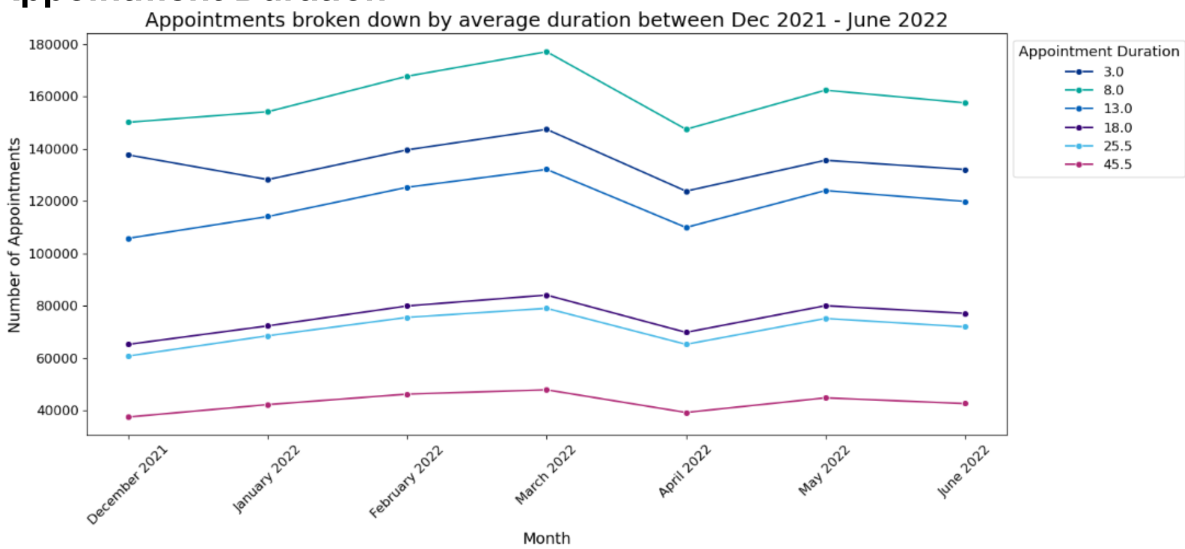
Using the ar DataFrame and a bar plot to show the difference in the number of appointments between months.



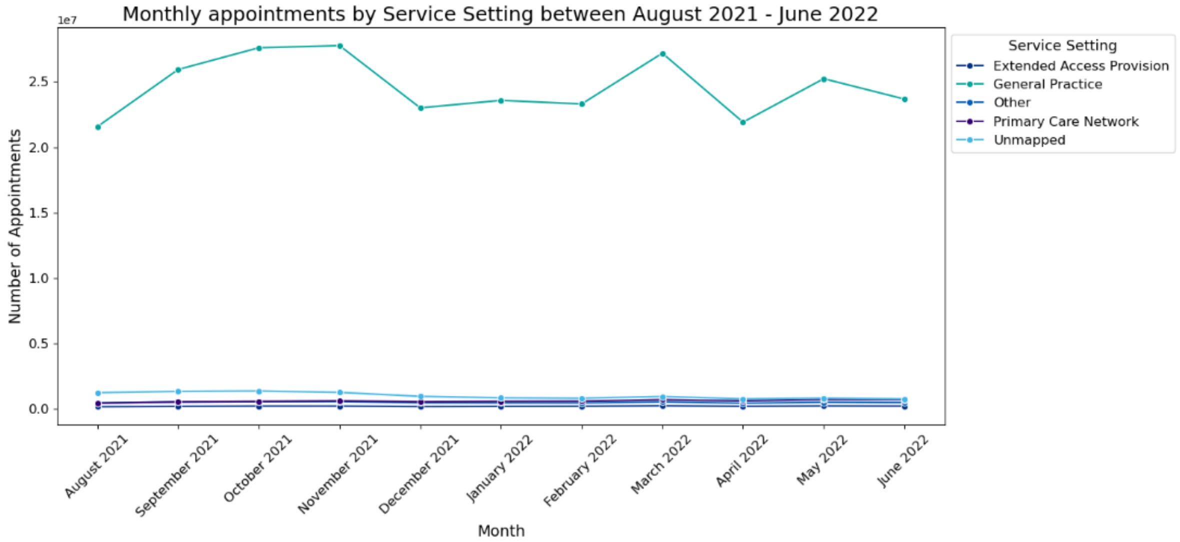
Appointment Details

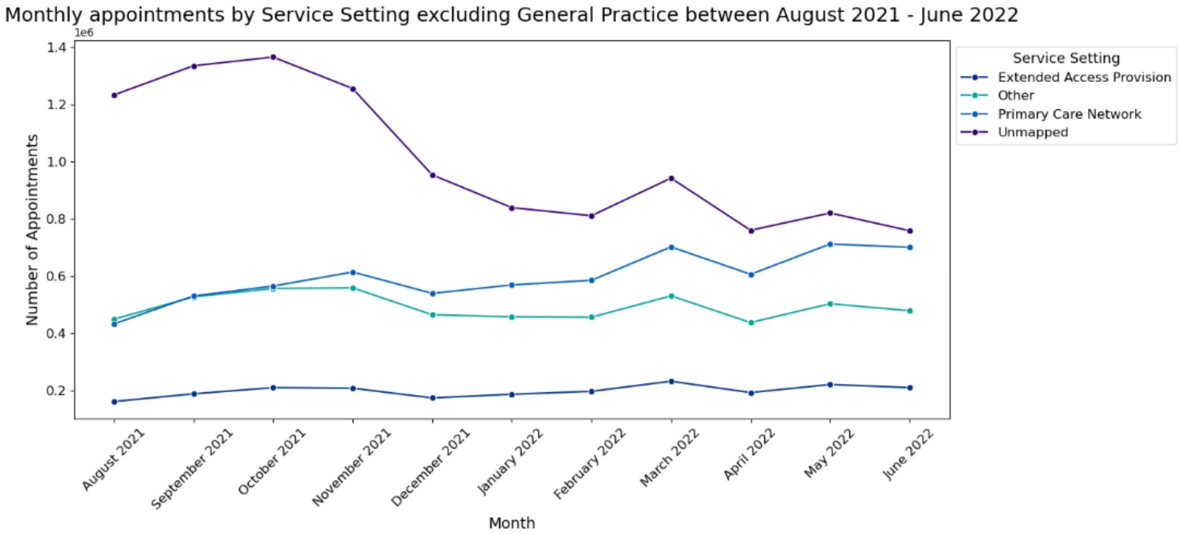
A time series plot was used for all appointment details' categories. This is to get a sense of the general trend in the number of appointments by appointment detail over the given months.
For comparability's sake, time series were plotted for Aug 2021 – J June 2022 using the filtered ar and nc DataFrame
Only exception to this is the average duration time series plotted f for Dec 2021 – Jun 2022 using the ad DataFrame.

Appointment Duration

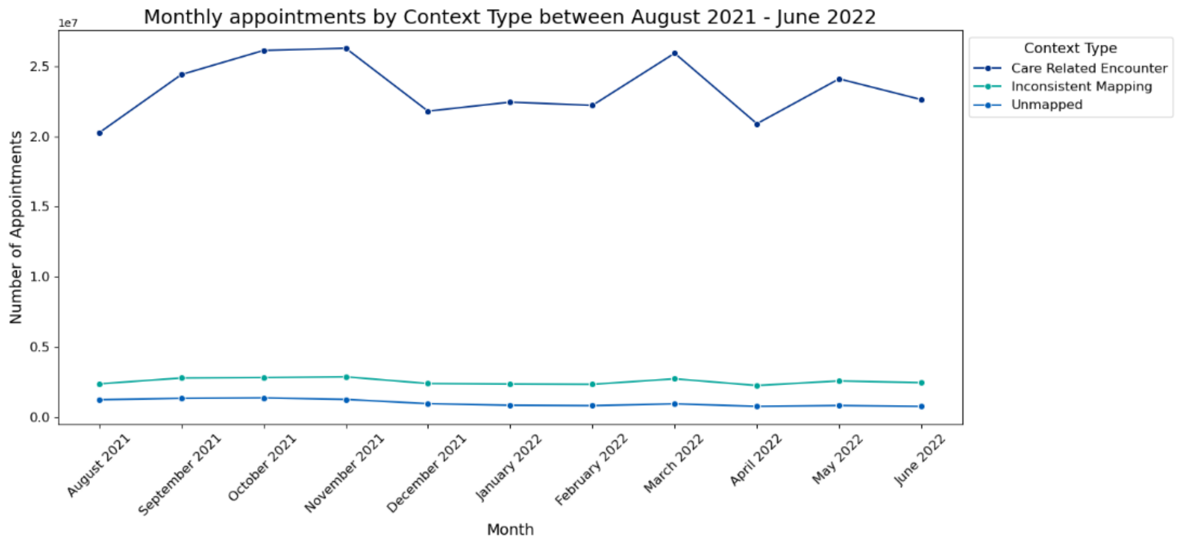


Service Setting:

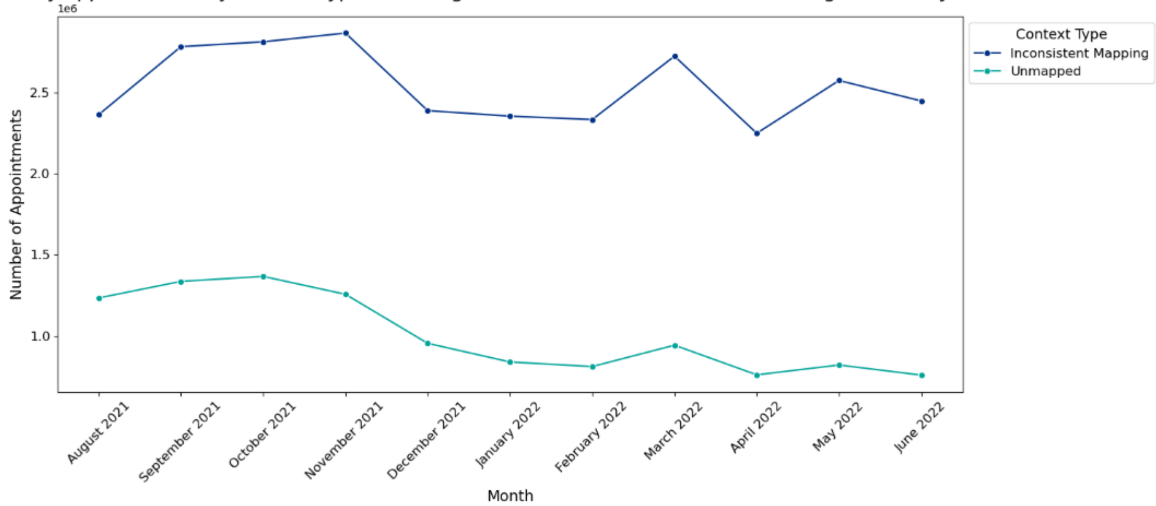




Context Type:

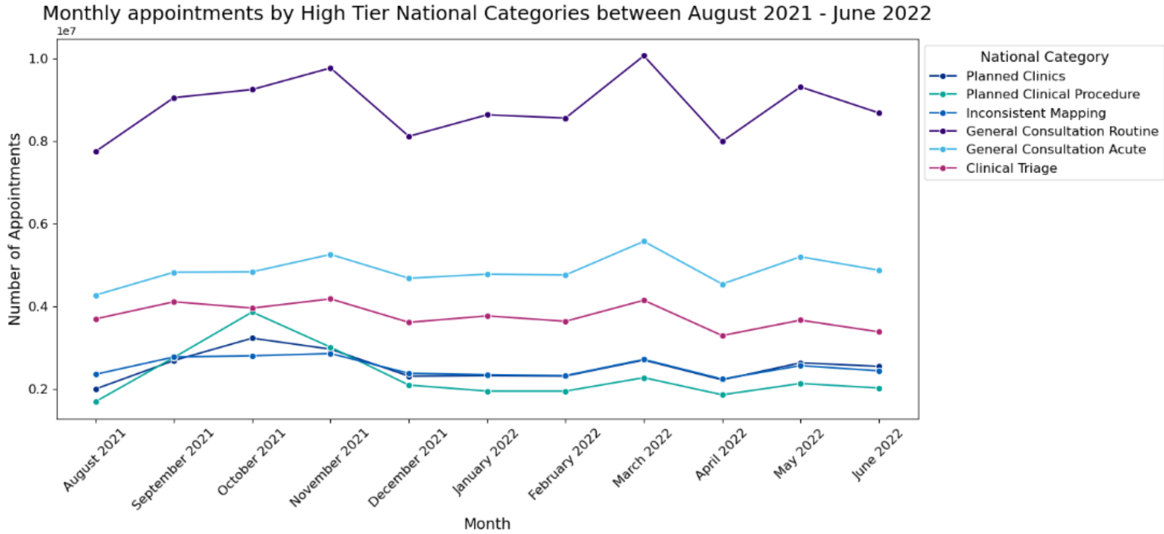


Monthly appointments by Context Type excluding Care Related Encounter between August 2021 - June 2022

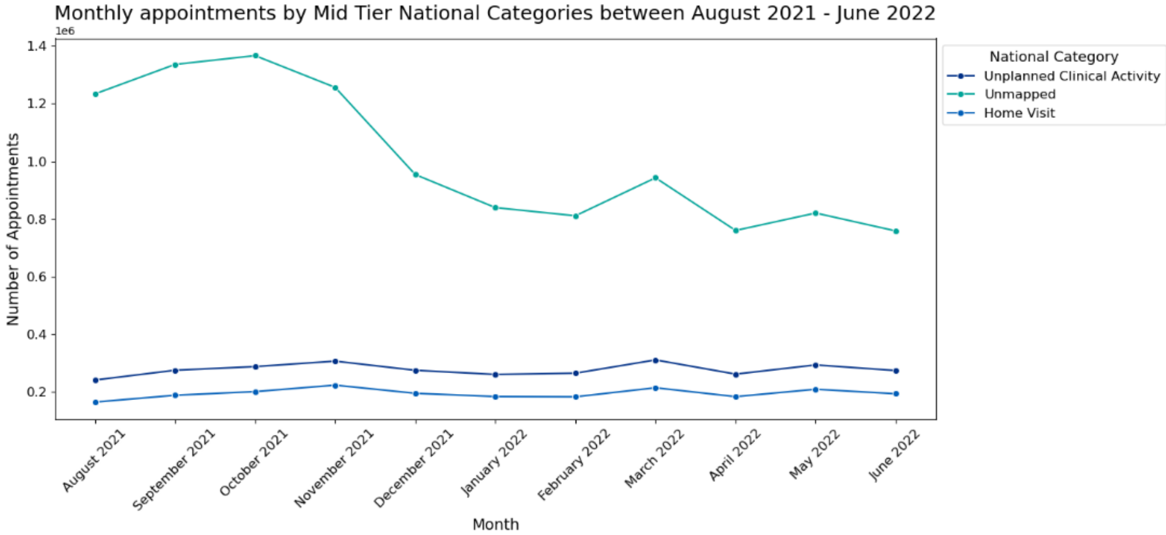


National Category:

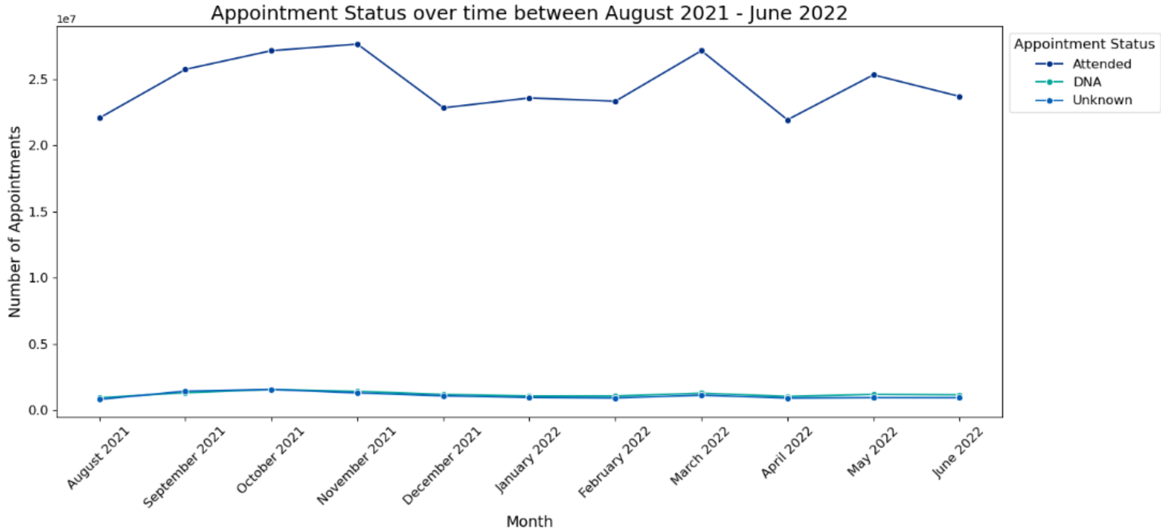
Monthly appointments by High Tier National Categories between August 2021 - June 2022

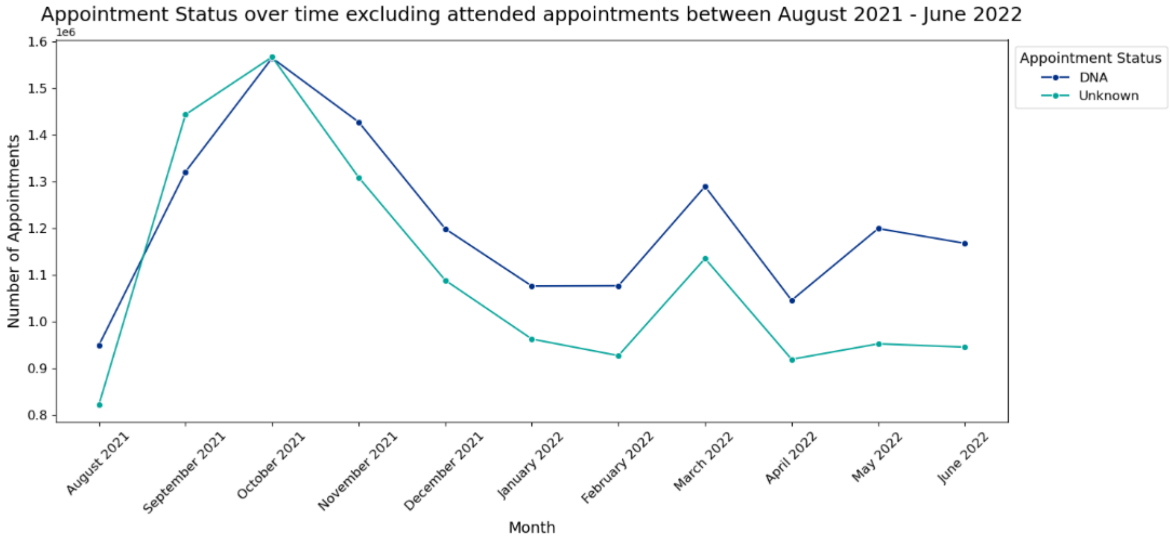


Monthly appointments by Mid Tier National Categories between August 2021 - June 2022



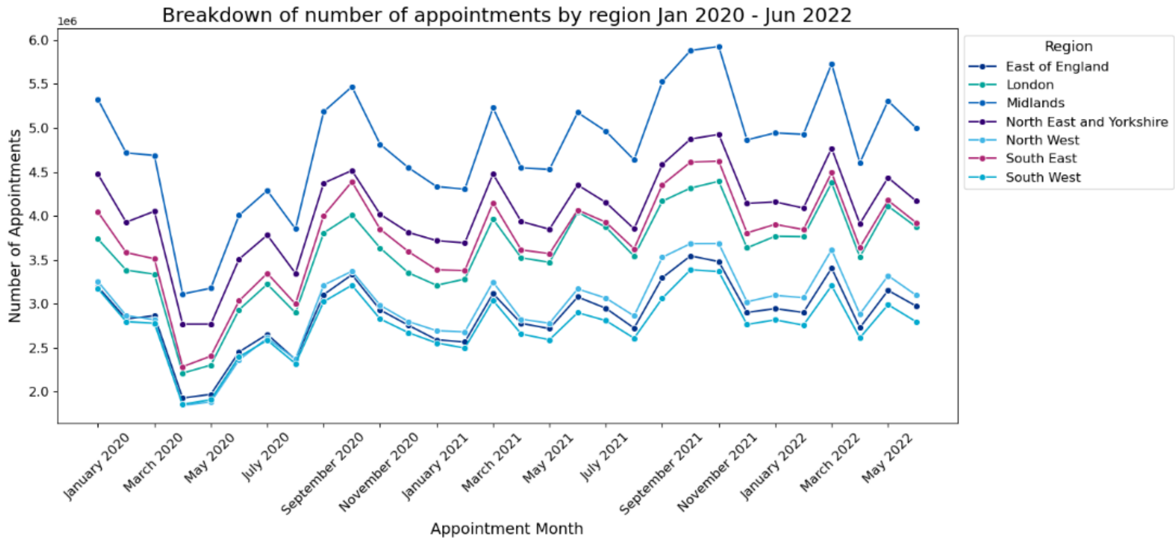
Appointment Status:



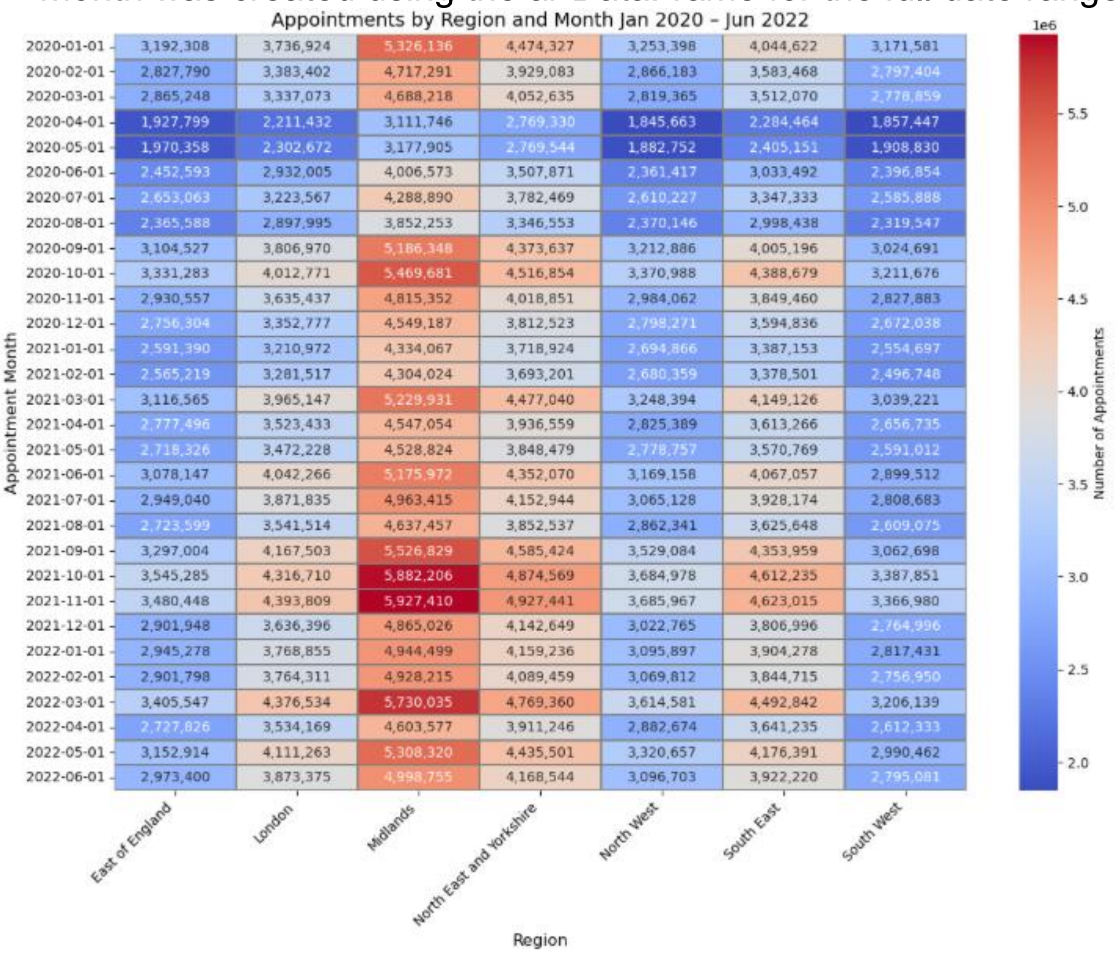


Regional/Seasonal

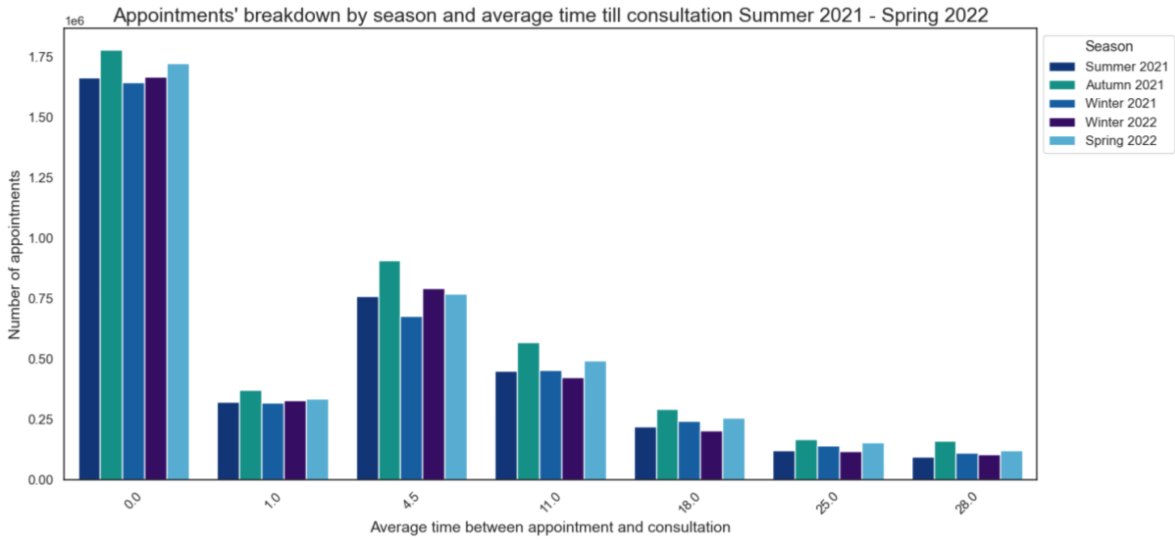
For regional trends, a time series plot was used.This is using the ar DataFrame for the full date range.



A heatmap of the pivoted number of appointments by region and month was created using the ar DataFrame for the full date range.

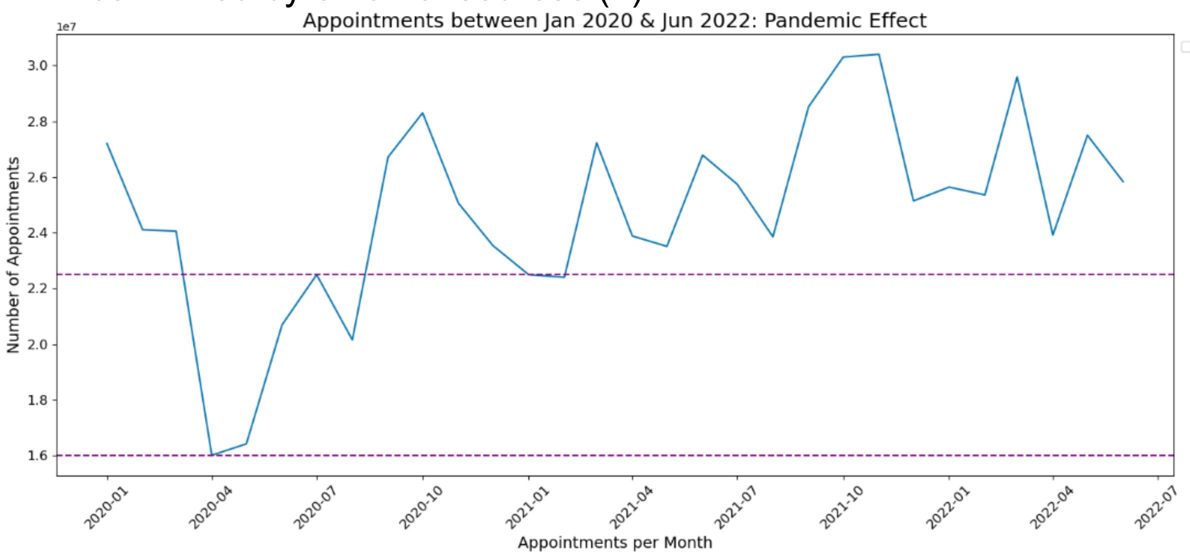


For seasonal trends affecting the time between booking and consultation a stacked bar chart was used to analyze Summer 2021 to Spring 2021 and that's to include full seasons.



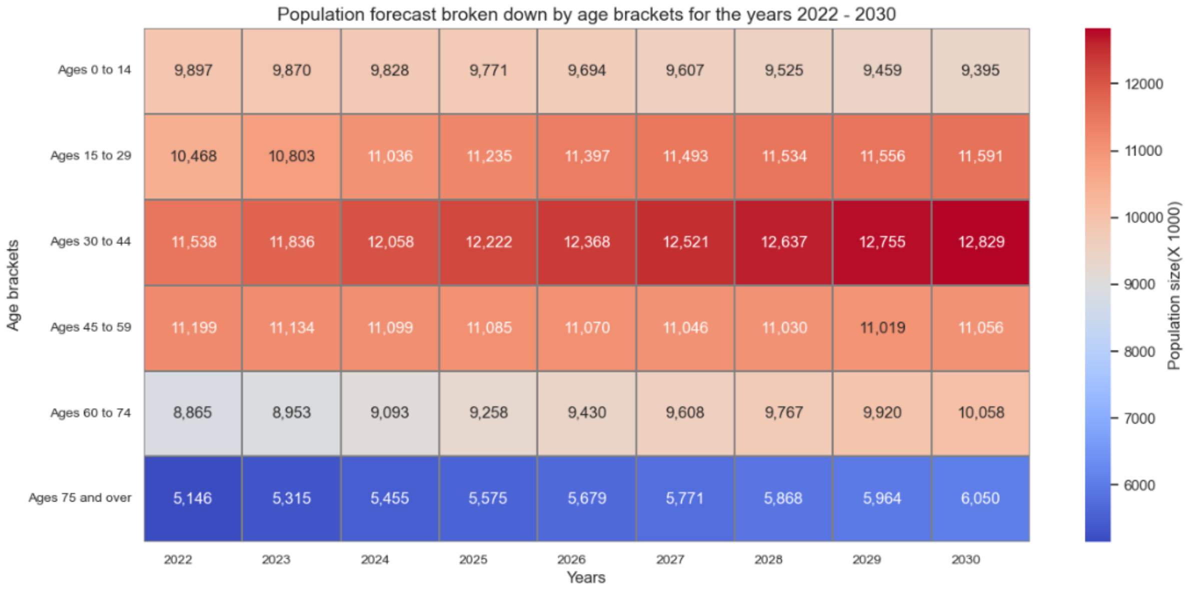
Crisis Management

The covid effect is highlighted on the time series below for the ar DataFrame. The two purple lines mark the drop. This pattern was confirmed by external sources (1)



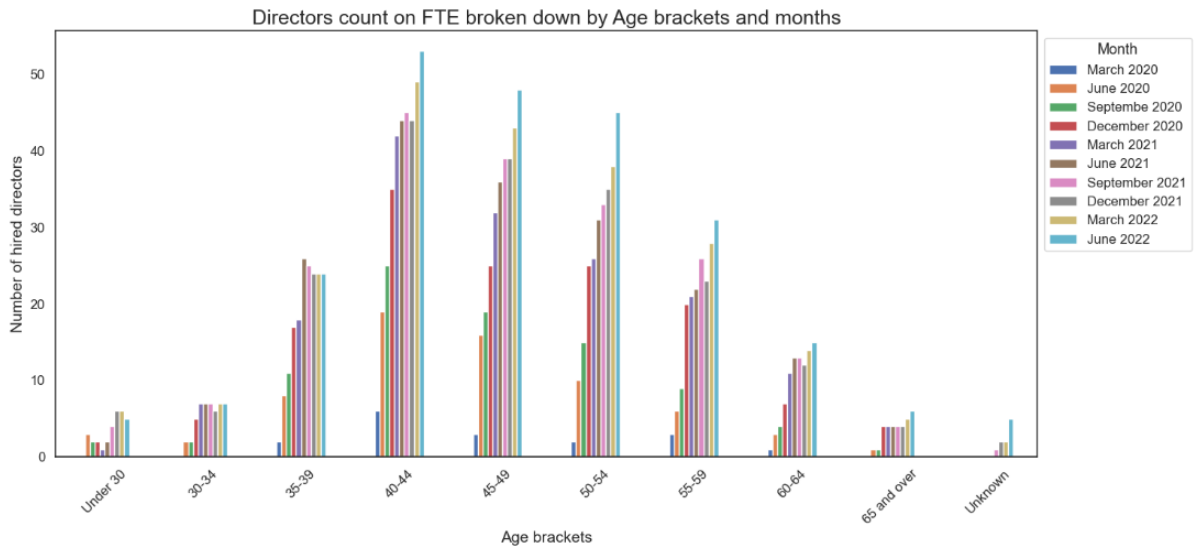
Population Needs

A forecast of the population size was added from an external source (2)
A heatmap was used to highlight the age bracket with the lowest and highest ranges.



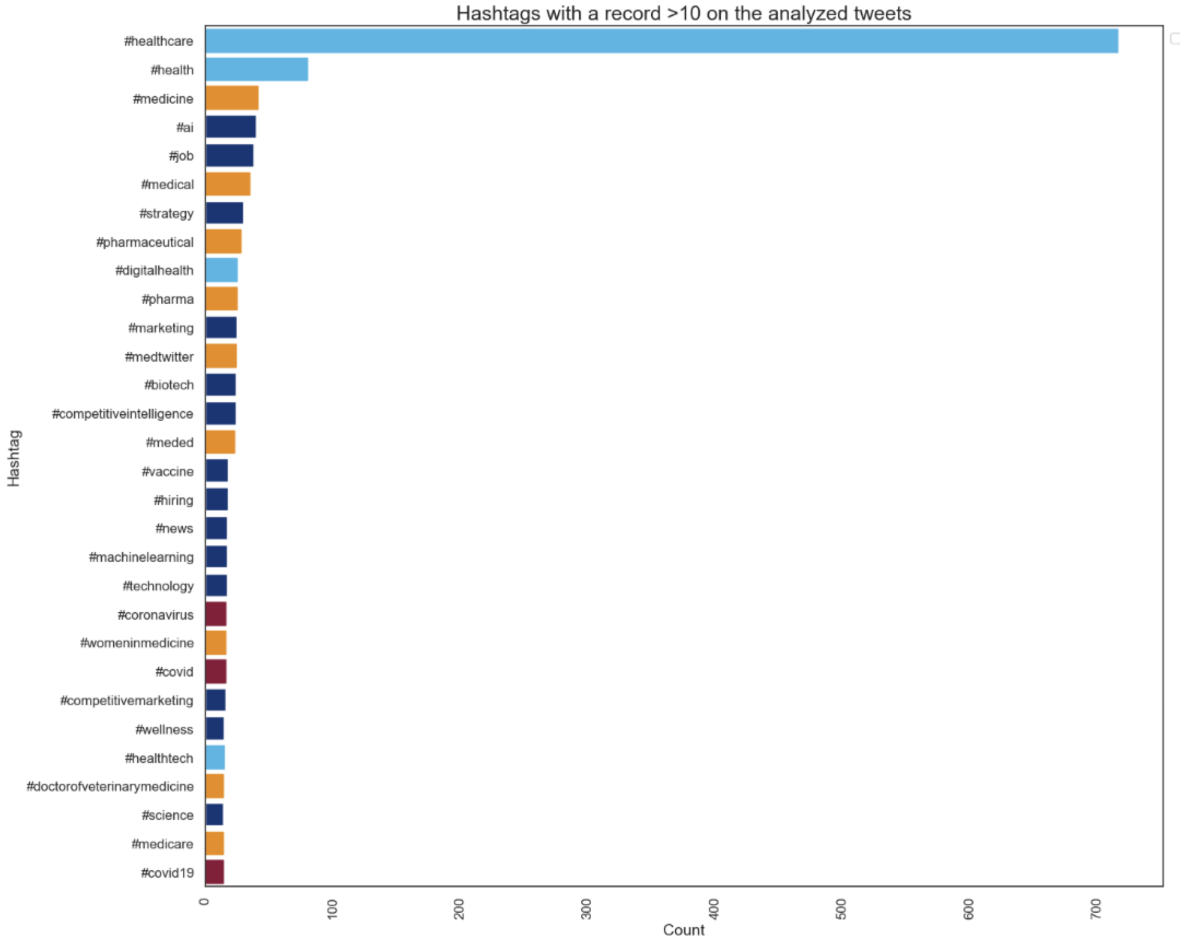
Staff

The number of staff members on an FTE basis was added to the analysis (3). This is to provide insight into the hiring pattern of the NHS.



Twitter

Hashtags were extracted from the tweet and the top 30 hashtags were plotted on a bar chart.



Limitations

The provided data sets didn’t include the following:

- Number of current patients served by the network
- Number of staff on FTE (Full Time Equivalent) basis
- Breakdown of the patients to staff ratio by region

External Sources

The analysis relied on external sources to fill the gap in the provided data, this includes the following:

- Region names (3)
- Data on FTE Staff numbers and positions (4)
- Population Forecast between 2022 and 2030 (5)

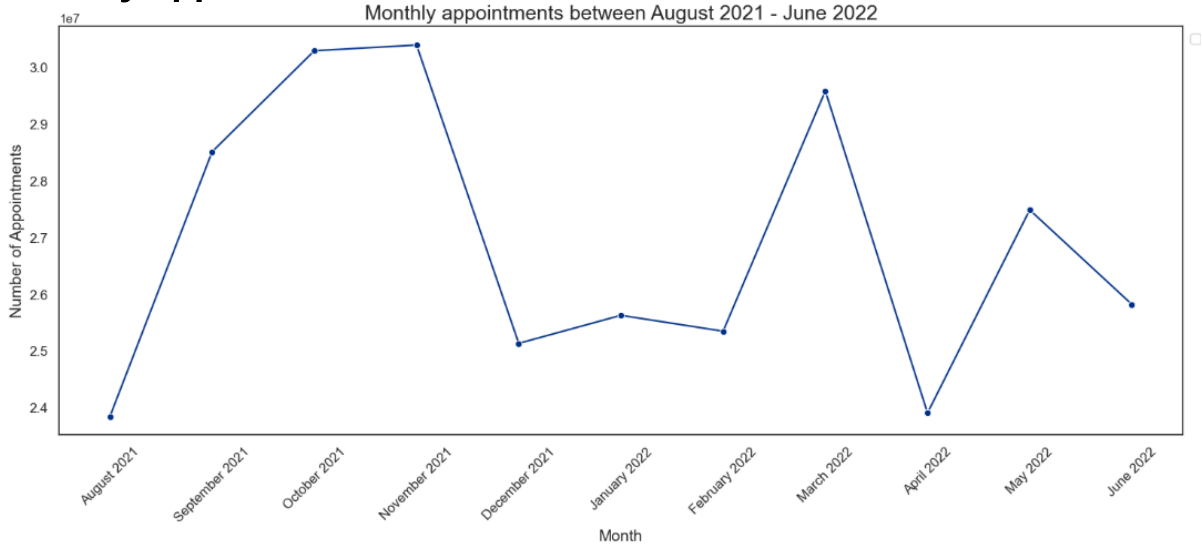
Visualizations & Insights

Utilization

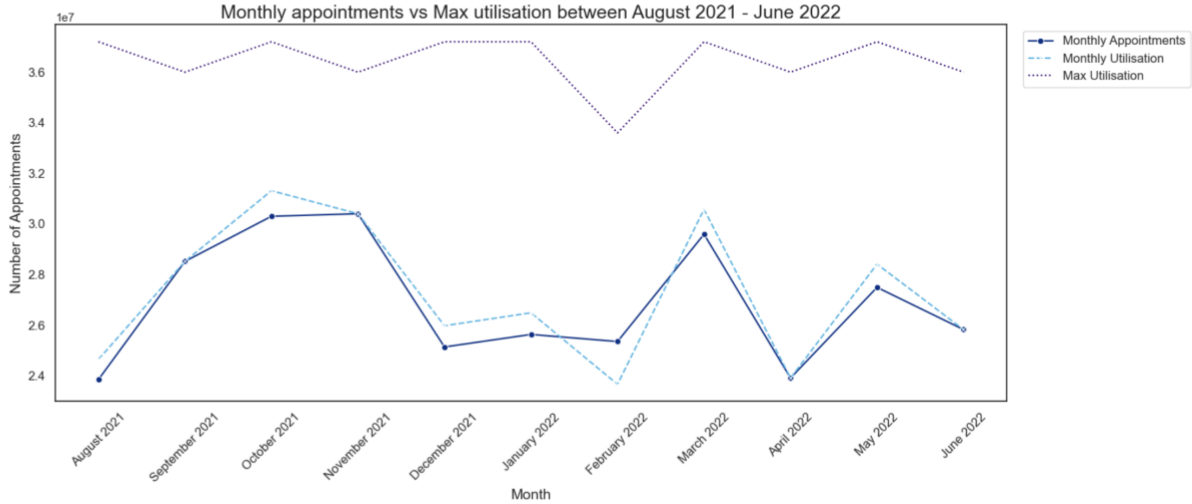
The utilization of the network followed the following logic:
- Daily utilization = Monthly number of appointments/30
- Monthly utilization = Daily utilization * actual number of days in a given month
- Max utilization = Given max daily average (1200000)*30

All three were plotted on a line chart to compare them and see if the appointments exceeded the given max utilization at any point.

Monthly appointments



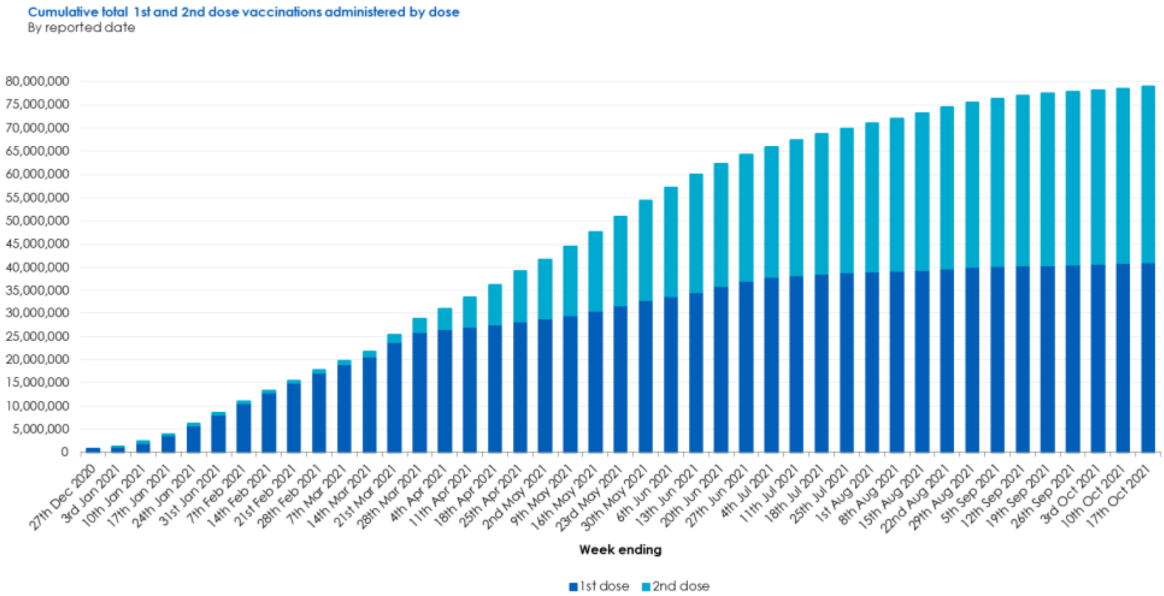
Monthly appointment vs Monthly utilization & Max utilization:
The following chart shows that the max utilization was not exceeded and that the system is working within the permissible limits.



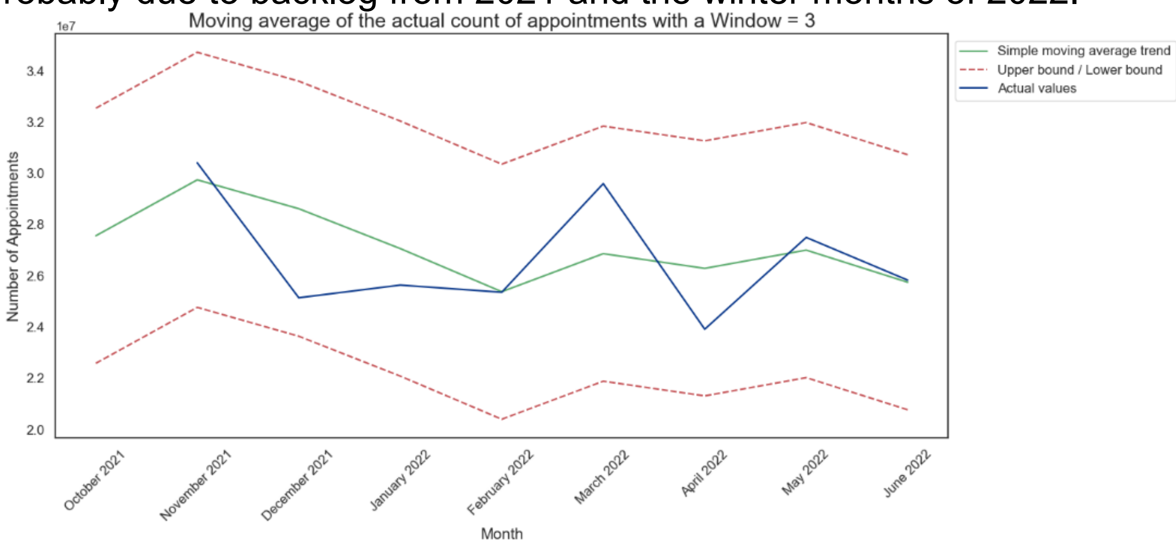
Moving Average Charts and Insights

Moving average (MA) charts were created for all sudden fluctuations noticed in the exploratory analysis phase.
This is because the MA smoothens them out and is a better indicator of future trends.

March 2022:
Apart from the peak in the number of appointments in November 2021 due to Covid vaccinations (5),

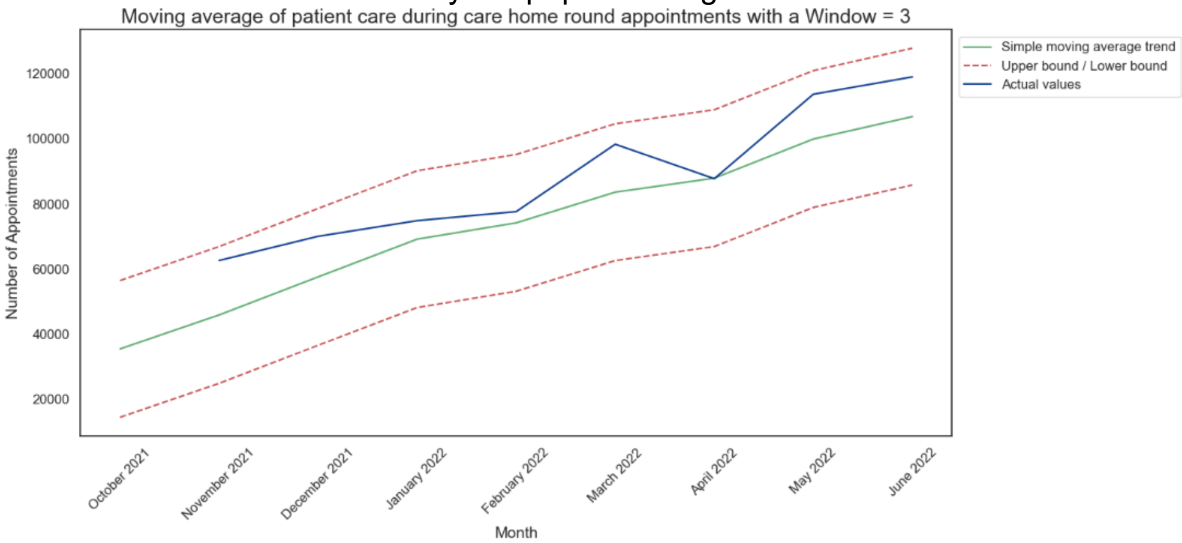


the noticed peak in March 2022 is smoothed out on the MA chart suggesting that it's not likely to repeat in the coming months. It was probably due to backlog from 2021 and the winter months of 2022.



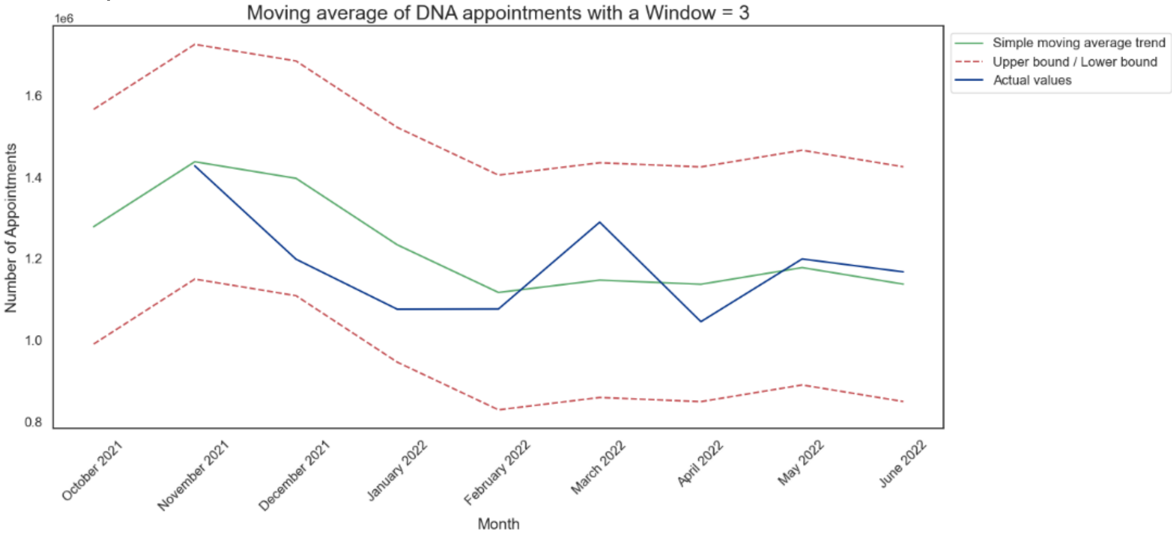
Patient Care During Care Home Rounds:
Compared to the other national categories, patient care during home rounds showed an out-of-trend chart compared to the other categories.

The chart confirms the positive slope suggesting an increase in the coming months. This is also confirmed by the population age data.



DNA Appointments

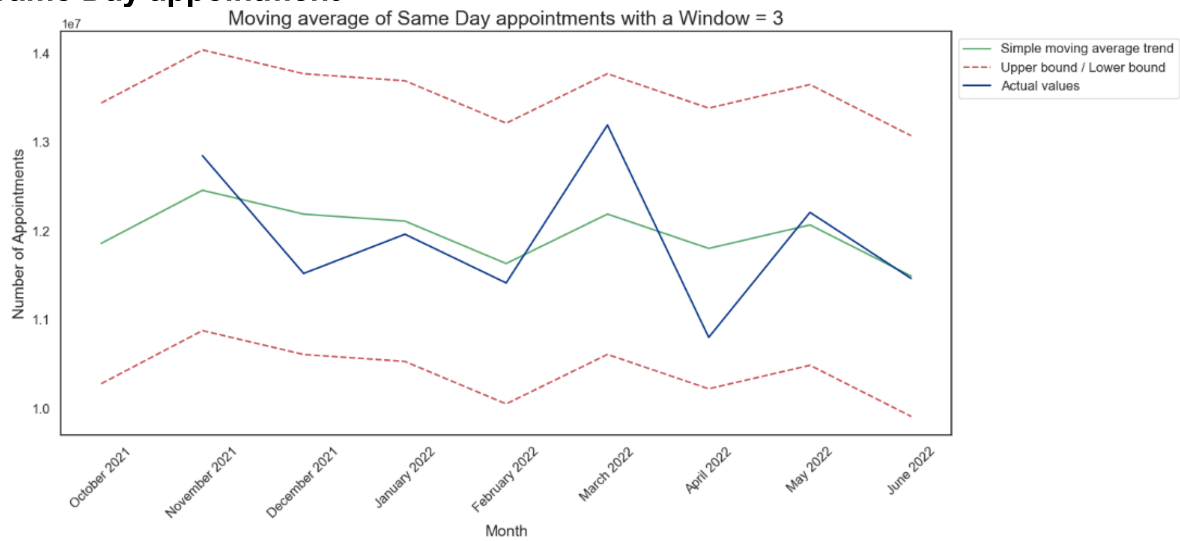
The efforts of the NHS in developing appointment reminder mechanisms have paid off as the MA chart is showing a stabilization in the DNA appointments curve after the peak in November 2021.



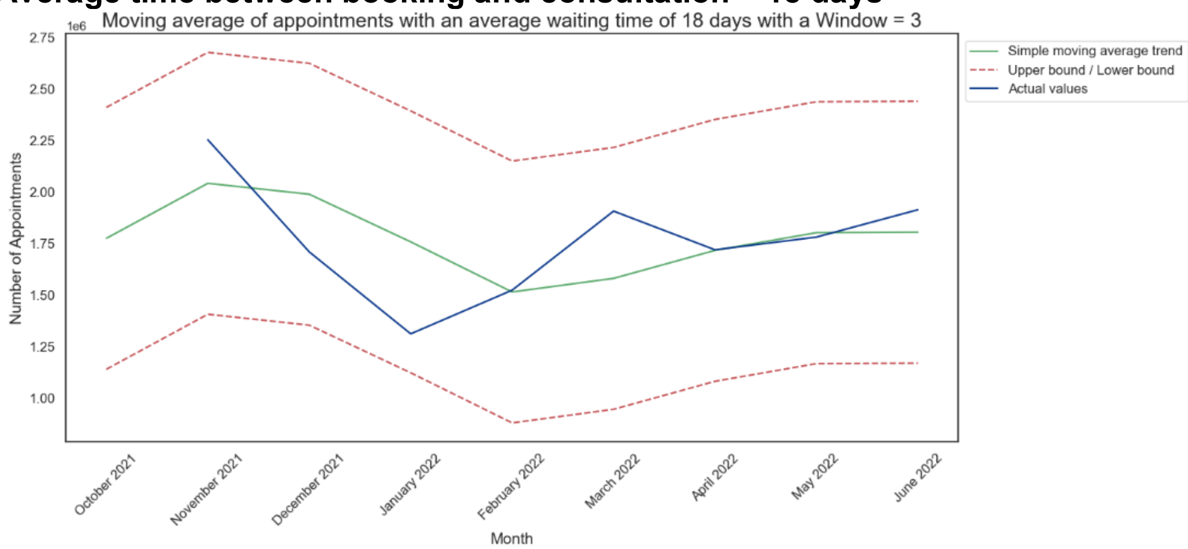
Time between booking and Consultation:

The MA charts for time between booking and consultation suggest an increase in the number of appointments with longer wait times in the coming months.
This was also confirmed by checking the charts for short wait time appointments.

Same Day appointment

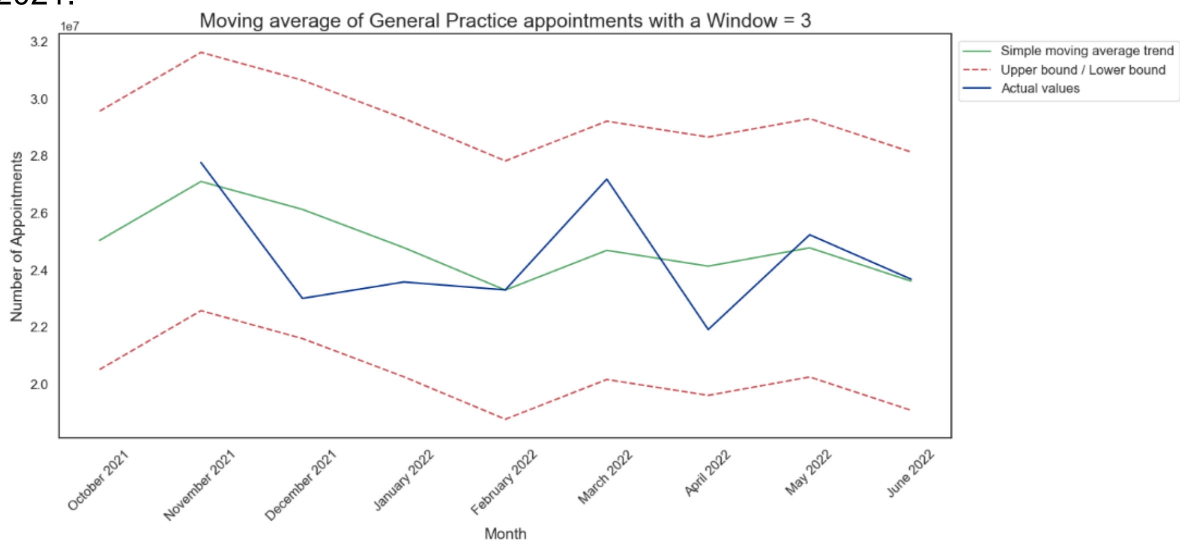


Average time between booking and consultation – 18 days



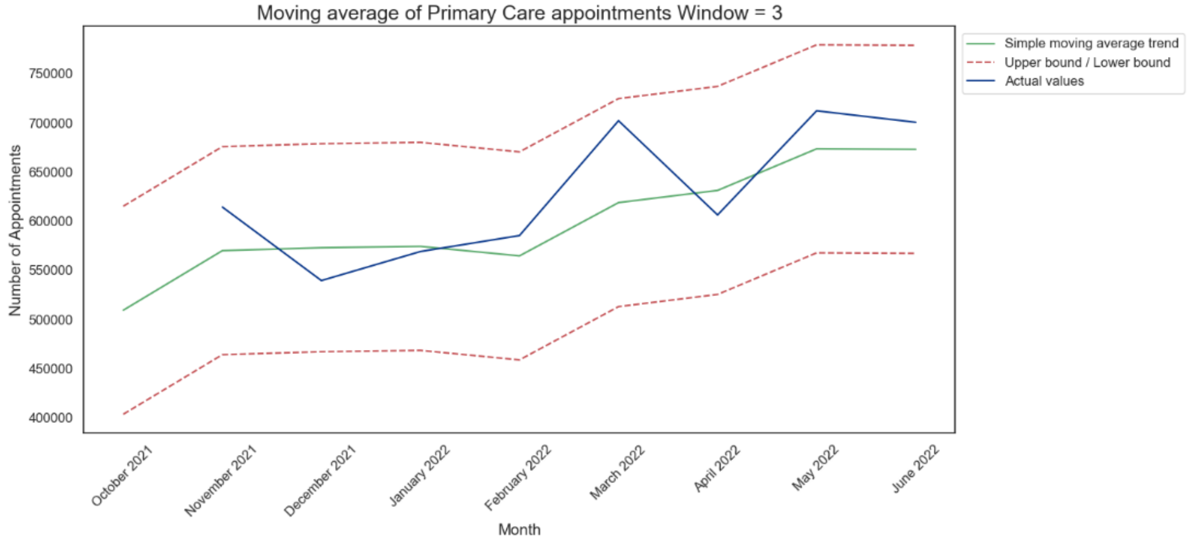
Service Setting

Appointments for General Practice have stabilized after the peak in November 2021.

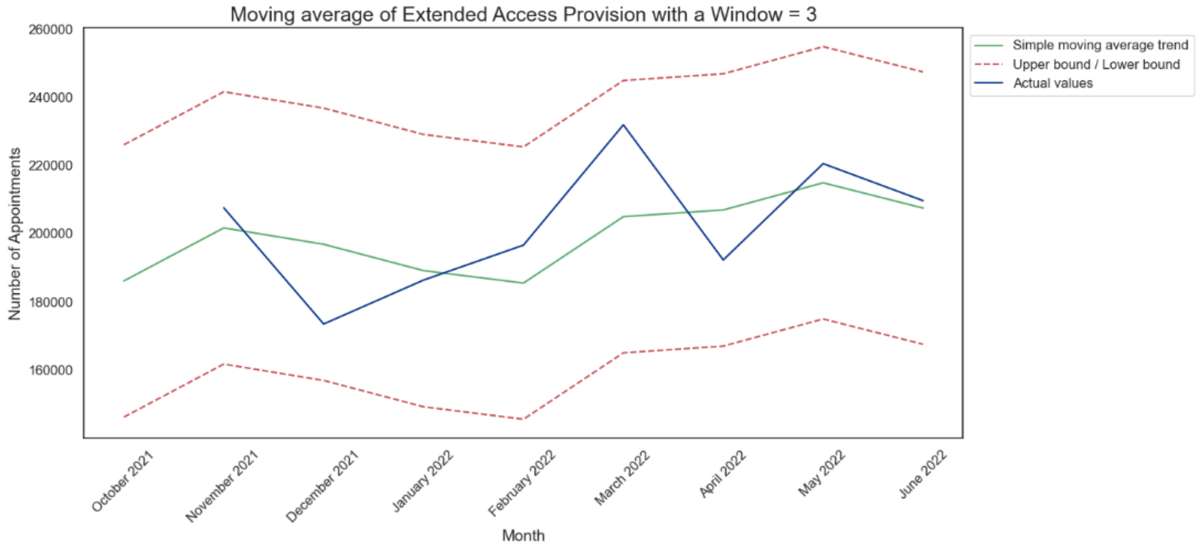


While Extended access provision and Primary care appointments are showing a positive slope. This is also confirmed by the following (6)

Primary Care Network



Extended Access Provision



Recommendations & Future Investigations

Recommendations

- NHS should keep up with its appointment reminder strategies as they have had a very positive impact on their reduction.
- The max utilization hasn't been reached even during the Covid crisis, but the expected increase in wait times could reflect negatively on the people's perception of the network.
- NHS should focus its new hires on primary care network and extended access services to flexibly meet the local needs and the expected increase in the number of appointments. This can be tested in the top regions by number of appointments:
  - Midlands
  - North East and Yorkshire
  - South East
  - London
- It should also focus on staff hiring for the elderly services and care homes as by 2030 more than 25% of the population would be 60 and above.
- Instead of working with 75% FTE staff (7), the NHS should aim to increase this percentage by 2030 to meet the demanding nature of an aging population.
- A follow-up analysis should be done in 2 years to assess the progress of the identified metrics.
- Unless coupled with a sentiment analysis tool, Twitter data would not add much value.

Future Investigations

- A detailed analysis of staff capacity, training periods, and turn-over rates would be necessary to analyse the reasons behind the expected longer wait times between appointments.

Appendix

I. Five-Whys Framework – Framework:

The NHS has concerns regarding the adequacy of staff and capacity in the networks.

Why does the NHS have concerns regarding the adequacy of staff and capacity in the networks?

This is because, according the Office of National Statistics, the population in the UK is both aging and growing. The NHS has the moral obligation to future-proof its infrastructure and staff capacity. (1)

Why does the NHS have the moral obligation to future-proof its network?

To ensure that patients served by the network receive timely and adequate health care and that the network takes all regions and health care needs into consideration.

Why does the NHS want to offer timely and adequate health care across regions and health needs?

To ensure consistent quality of care and to optimize appointment scheduling, both in terms of duration and wait times between booking and consultation. This would as a result reflect positively on the quality of care, overall patient satisfaction and cost optimization.

Why does the NHS want to ensure the quality of their care services while optimizing their costs?

So that the NHS continues to sustainably serve the population across regions and care contexts.

Why does the NHS put such emphasis on delivering adequate and sustainable care?

To abide by its ethical responsibility to ensure equitable healthcare access that prevents the increase in health inequalities across different population groups.

II. DataFrames Names - Appendix

- ad: actual_duration
- ar: appointment_regional
- nc: national_categories
- tw: tweets

III. Descriptive Statistics – Appendix:

- Descriptive statistics were calculated for all DataFrames after proper data types adjustments

ad Descriptive Statistics:

- The average duration data is skewed to the right. Since the mean > the median

	average_duration	count_of_appointments
count	117632.000000	137793.000000
mean	18.635138	1219.080011
std	13.815502	1546.902956
min	3.000000	1.000000
25%	8.000000	194.000000
50%	13.000000	696.000000
75%	25.500000	1621.000000
max	45.500000	15400.000000

ar Descriptive Statistics:

- The avg-time-between-book-appointment is symmetric with the mean being very close to the median.

	avg_time_between_book_appointment	count_of_appointments
count	567134.000000	596821.000000
mean	11.079312	1244.601857
std	10.308394	5856.887042
min	0.000000	1.000000
25%	1.000000	7.000000
50%	11.000000	47.000000
75%	18.000000	308.000000
max	28.000000	211265.000000

nc Descriptive Statistics:

- The count of appointments is very positively skewed with a very big gap between the mean and the median

	count_of_appointments
count	817394.000000
mean	362.183684
std	1084.576600
min	1.000000
25%	7.000000
50%	25.000000
75%	128.000000
max	16590.000000

tw Descriptive Statistics:

- The stats for both the retweet count and the favorite count suggest a low overall user engagement
- This is more the case for the favorite count, where the like function seemed to have been deactivated except for a few posts

	tweet_retweet_count	tweet_favorite_count
count	1174.000000	1174.00000
mean	8.629472	0.37138
std	29.784675	2.04470
min	0.000000	0.00000
25%	0.000000	0.00000
50%	1.000000	0.00000
75%	3.000000	0.00000
max	303.000000	42.00000

IV. Date Range for the provided Data Sets – Appendix

After checking the date range for all three DataFrames:

- Common data range for the three DataFrames is between 01.12.2021 & 30.06.2022
- Common date range for ar & nc is between 01.08.2021 & 30.06.2022

ad Date Range

- Start: 01.12.2021
- End: 30.06.2022

ar Date Range

- Start: 01.2020
- End: 06.2022

Nc Date Range

- Start: 01.08.2021
- End: 30.06.2022

V. User Defined Functions – Appendix

- ### Function to filter the DataFrames by date

```
# Define function to filter DataFrame by date taking start, end date and DataFrame as parameters

def filter_date(
    df,
    month_column,
    start,end
):
    '''
    This function's aim is to filter any DataFrame by date.
        Parameters:
            df: DataFrame to filter
            month_column: Column to filter
            start: Start date
            end: End date
        Return:
            filtered_df: The filtered DataFrame
    '''

    # Filter the DataFrame by including all the rows within the start and end date and sort by the month column
    filtered_df = df[
        (df[month_column]>=start)& (df[month_column]<=end)
    ].sort_values(by =month_column, ascending = True)

    return filtered_df
```

- ### Function to clean up all plotted charts and standardize fonts

```
def chart_clean(
    title,
    xlabel,
    ylabel,
    legend
):
    '''
    The purpose of this function is to standardize the style and fontsize of all created charts
    Parameters:
        title: Title of the chart
        xlabel: x axis label of the chart
        ylabel: y axis label of the chart
        legend: Legend of the chart
    '''
    # Format the title of the chart
    plt.title(title,fontsize = 18)

    # Format the xlabel of the chart
    plt.xlabel(xlabel,fontsize=14)

    # Format the ylabel of the chart
    plt.ylabel(ylabel,fontsize=14)

    # Format the xticks by rotating them and adjusting the size
    plt.xticks(rotation =45,fontsize= 12)

     # Format the yticks by rotating them and adjusting the size
    plt.yticks(fontsize = 12)

    # Format the legend and place it outside of the chart area
    plt.legend(title=legend,fontsize=12,title_fontsize=13,bbox_to_anchor=(1,1),loc='upper left')

    plt.tight_layout()
```

- **Function to extract the integers of both the actual duration column and the time between book and appointment**

```
# Define a function that would be used to extract the duration in Minutes from the actual_duration string
# and the number of days between appointment booking and actual appointment date

def extract_integers(value):
    ''' The role of this function is to extract integers from a string.
        It also takes into account unknown values and replaces them with NaN.
        It also takes same-day appointments into account and replaces them with 0
        Parameters:
            value: cell values
        Returns:
            numbers: returns the average of the numbers series or the single value saved in it as integer
    '''
    # Change the string to lower case
    value = str(value).lower()

    # Check if the string is Unkown and returns NaN
    if 'unkown' in value:
        return np.nan

    # Check if the string is Same and returns 0
    if 'same' in value: # Check if the string is Same and returns 0
        return 0

    # Save the extracted integers in a series numbers
    numbers = pd.Series(value).str.findall(r'\d+').iloc[0]

    # Check the length of the numbers series and take the average of both integers if the length is 2
    if len(numbers) == 2:
        return (int(numbers[1])+int(numbers[0]))/2

    # Check the length of the number series and return the number as an integer
    elif len(numbers) == 1:
        return int(numbers[0])
```

- **Function to add a season column to all DataFrames**

```
# Function to assign a season after checking the month
def season_func(month):
    '''
        The goal of this function is to test the month and assign the correct season.
        Creating a function would help apply that to multiple DataFrames and avoid repetitive tasks.
        Parameters:
            month: The month to assign the correct season
        Returns:
            Summer: for months 6, 7, 8
            Autumn: for months 9, 10,11
            Winter: for months 12, 1, 2
            Spring: for months 3, 4, 5
    '''
    if month in [6,7,8]:
        return 'Summer'
    elif month in [9,10,11]:
        return 'Autumn'
    elif month in [12,1,2]:
        return 'Winter'
    else:
        return 'Spring'
```

- **Function to add a region column to all DataFrames**

```python
# Define a function to add the region names as a column to the DataFrames
def region(df):
    '''
    This function mainly merges the region column to the DataFrame that is passed to it based on the "icb_ons_code"
    Parameters:
        df: DataFrame
    Return:
        df: DataFrame with added region column
    '''
    df=pd.merge(df,
                regions_unique,how='left',
                left_on='icb_ons_code',
                right_on='ICB23CD')
    df=df.drop(columns='ICB23CD')
    df=df.rename(columns={'NHSER23NM':'region'})
    return df
```

- **Function to add a percentage column to a DataFrame**

```python
# Create function to add a percentage column to a DataFrame
def percent(df):
    '''
    The purpose of this function is to add a percentage column to a DataFrame.
    The percentage is calculated based on the count of appointments' column
    Parameters:
        df: DataFrame to add the percentage column
    return:
        df: Adjusted DataFrame after adding the percentage column
    '''

    df['percentage'] = (df['count_of_appointments'] / df['count_of_appointments'].sum()) * 100
    df['percentage'] = df['percentage'].round(2)  # optional: round to 2 decimals
    return(df)
```

- **Function to pivot a DataFrame by a certain column and then plot a boxplot**

```python
def pivot_boxplot(df,column):
    '''
    The aim of this function is to pivot the DataFrame by appointment month and region.
    The pivoting will only be for one column to be able by the count of appointments.
        Parameters:
            df: DataFrame
            column: Column to pivoted and plotted on a boxplot
        Return:
            df_pivot: Return the pivoted DataFrame
    '''
    df_pivot = df.pivot_table(index = ['appointment_month_str','region'],
                              columns =[column],
                              values = ['count_of_appointments'],
                              aggfunc ='sum').reset_index()

    # Flatten MultiIndex columns
    df_pivot.columns = [
        col if isinstance(col, str) else
        col[1] if col[0] == 'count_of_appointments' else
        '_'.join([str(c) for c in col if c])
        for col in df_pivot.columns.values
    ]

    # Set the figure size
    plt.figure(figsize=(15,7))

    # Define the color palette of the boxplot
    nhs_colors = [
    '#003087',  # NHS Dark Blue
    '#005EB8',  # NHS Blue
    '#41B6E6',  # NHS Light Blue
    '#00A9CE',  # NHS Aqua Blue
    '#00A499',  # NHS Aqua Green
    '#330072',  # NHS Purple
    '#7C2855',  # NHS Dark Pink
    '#AE2573',  # NHS Pink
    '#8A1538',  # NHS Dark Red
    '#DA291C',  # NHS Red
    '#ED8B00'   # NHS Orange
    ]

    # Plot the boxplot
    sns.boxplot(df_pivot,palette=nhs_colors)
    return df_pivot
```

- **Function to group the nc DataFrame by a certain column**

```python
# Function to groupby the nc DataFrame
def group_nc(column):

    ''' This function is used to group nc by a specific column
            Parameters: column
            Return: the grouped nc DataFrame
    '''
    df_group = nc.groupby(['appointment_month',
                           'appointment_month_str',
                           column])[['count_of_appointments']]\
    .sum().sort_values(by='appointment_month',ascending=True).reset_index()
    return df_group
```

- **Function to plot the moving average**

```python
# This is a function to calculate and plot the simple moving average:
def plot_moving_average(series, window, plot_intervals=False, scale=1.96):

    rolling_mean = series.rolling(window=window).mean()
    # Set the color palette
    sns.set_palette(nhs_colors)

    plt.figure(figsize=(15,7))
    plt.title("Moving average\n window size = {}".format(window))
    plt.plot(rolling_mean, 'g', label="Simple moving average trend")

    # Plot confidence intervals for smoothed values.
    if plot_intervals:
        mae = mean_absolute_error(series[window:], rolling_mean[window:])
        deviation = np.std(series[window:] - rolling_mean[window:])
        lower_bound = rolling_mean - (mae + scale * deviation)
        upper_bound = rolling_mean + (mae + scale * deviation)
        plt.plot(upper_bound, 'r--', label="Upper bound / Lower bound")
        plt.plot(lower_bound, 'r--')

    plt.plot(series[window:], label="Actual values")
    plt.legend(loc='best')
    plt.grid(False)
```

VI. Null Values – Appendix

1. **Null Values ad DataFrame – average_duration column:**
   a. Percentage of null values in the average_duration column in the ad DataFrame is: 14.63%
   b. Region with the highest null values in the average_duration column is North West followed ny North East and Yorkshire

```
region
North West                 4849
North East and Yorkshire   4252
Midlands                   3675
East of England            2748
South East                 2216
South West                 1384
London                     1037
```

   c. Icb_ons_code with the highest null values in the average_duration column is **E54000057, E54000008, E54000050**

```
icb_ons_code
E54000057    1837
E54000008    1584
E54000050    1458
E54000048    1428
E54000051    1159
```

   d. Season with the highest null values in the average_duration column is Spring 2022:

```
season_year
Spring 2022    8821
Winter 2022    5647
Winter 2021    2885
Summer 2022    2808
```

   e. The month with the highest null values in the average_duration column is March 2022:

```
appointment_month_str
March 2022       3080
May 2022         2934
January 2022     2898
December 2021    2885
June 2022        2808
April 2022       2807
February 2022    2749
```

2. **Null values ar DataFrame in the avg_time_between_book_appointment column**
    a. Percentage of null values in the avg_time_between_book_appointment in the ar DataFrame is: 4.97%
    b. Region with the highest null values in the avg_time_between_book_appointment is **North West** and **North East and Yorkshire**

```
region
North West                     6404
North East and Yorkshire       5288
Midlands                       5273
South East                     3897
East of England                3241
London                         2807
South West                     2777
```

    c. icb_ons_code with the highest null values in the avg_time_between_book_appointment is **E54000057, E54000008, E54000050**

```
icb_ons_code
E54000057    2465
E54000008    2409
E54000050    1953
E54000048    1530
E54000051    1243
```

    d. The season with the highest null values in the avg_time_between_book_appointment is Spring 2022

```
season_year
Autumn 2021    3237
Spring 2022    3202
Summer 2021    3058
Autumn 2020    2982
Winter 2021    2933
```

    e. The month with the highest null values in the avg_time_between_book and appointment is March 2022

```
appointment_month_str
March 2022        1110
October 2021      1105
May 2022          1103
November 2021     1091
June 2022         1066
```

3. **Conclusion**

    a. Highest regions by number of null values in both DataFrames are **North West and North East and Yorkshire**
    b. Highest icb ons code by number of null values in both DataFrames are **E54000057, E54000008, E54000050**
    c. Highest month by number of null values in both DataFrames is **March 2022**

VII. Number of records – Appendix

The records counts for all DataFrames was calculated using the value.counts() method

ad DataFrame Records

	appointment_month	count_of_appointments
0	March 2022	21236
1	May 2022	20128
2	January 2022	19643
3	December 2021	19507
4	June 2022	19227

ar DataFrame Records

	appointment_month	count_of_appointments
0	March 2020	21350
1	January 2020	20889
2	November 2021	20766
3	February 2020	20689
4	October 2021	20562

nc DataFrame Records

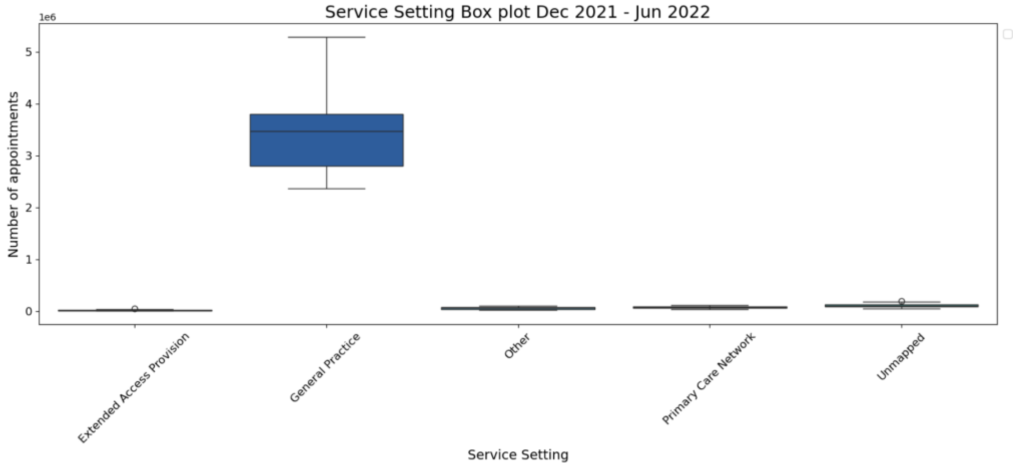
	appointment_month	count_of_appointments
0	March 2022	82822
1	November 2021	77652
2	May 2022	77425
3	September 2021	74922
4	June 2022	74168
5	October 2021	74078
6	December 2021	72651
7	January 2022	71896
8	February 2022	71769
9	April 2022	70012
10	August 2021	69999

VIII. Service Setting – Appendix:

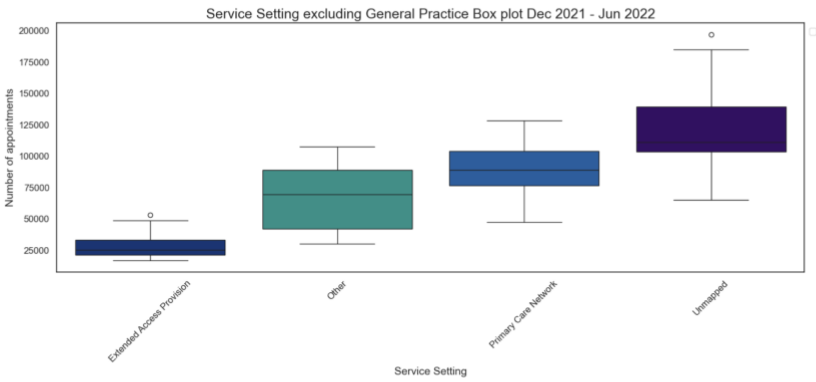
- There are 5 unique service settings
- nc DataFrame was filtered to include appointments between Dec 2021 – Jun 2022
- **General Practice** is the service setting with the highest number of appointments
- Followed by **Unmapped** appointments showing a large gap in the data collection and quality
- **Extended Access Provision** is the lowest service setting in terms of number of appointments

service_setting	count_of_appointments	percentage
General Practice	167920958	91.78
Unmapped	5887700	3.22
Primary Care Network	4415148	2.41
Other	3328530	1.82
Extended Access Provision	1410858	0.77

Service Setting Boxplot



Service Setting Boxplot excluding General Practice

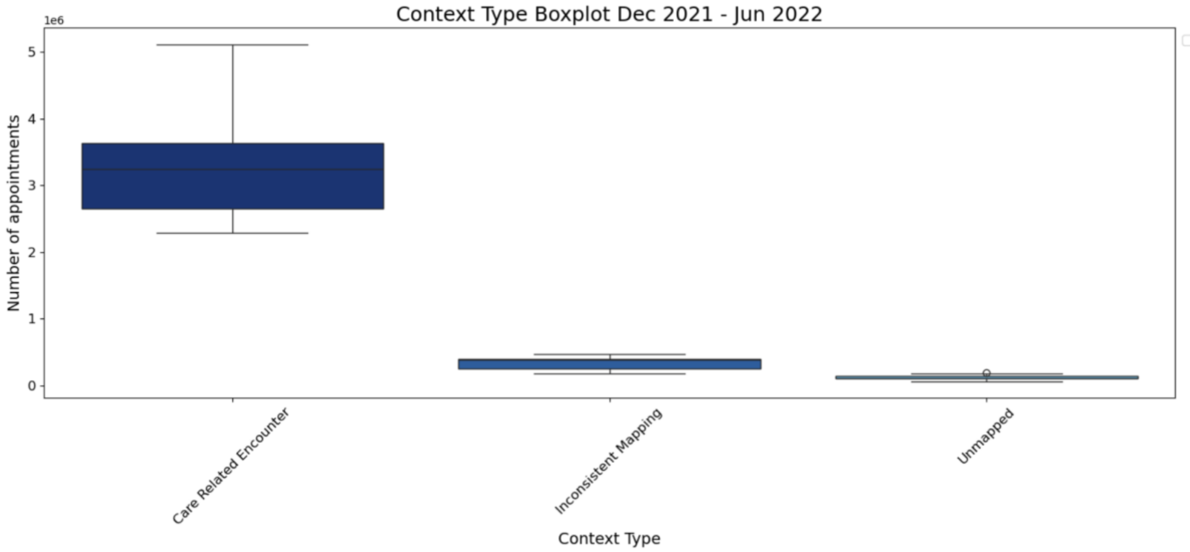


IX. Context Type – Appendix:

- There are 3 unique context types
- nc DataFrame was filtered to include appointments between Dec 2021 – Jun 2022
- **Care-related encounter** is the context type with the highest number of appointments
- Followed by appointments categorized under **inconsistent mapping**

	count_of_appointments	percentage
context_type		
Care Related Encounter	160007693	87.45
Inconsistent Mapping	17067801	9.33
Unmapped	5887700	3.22

Context Type – Boxplot

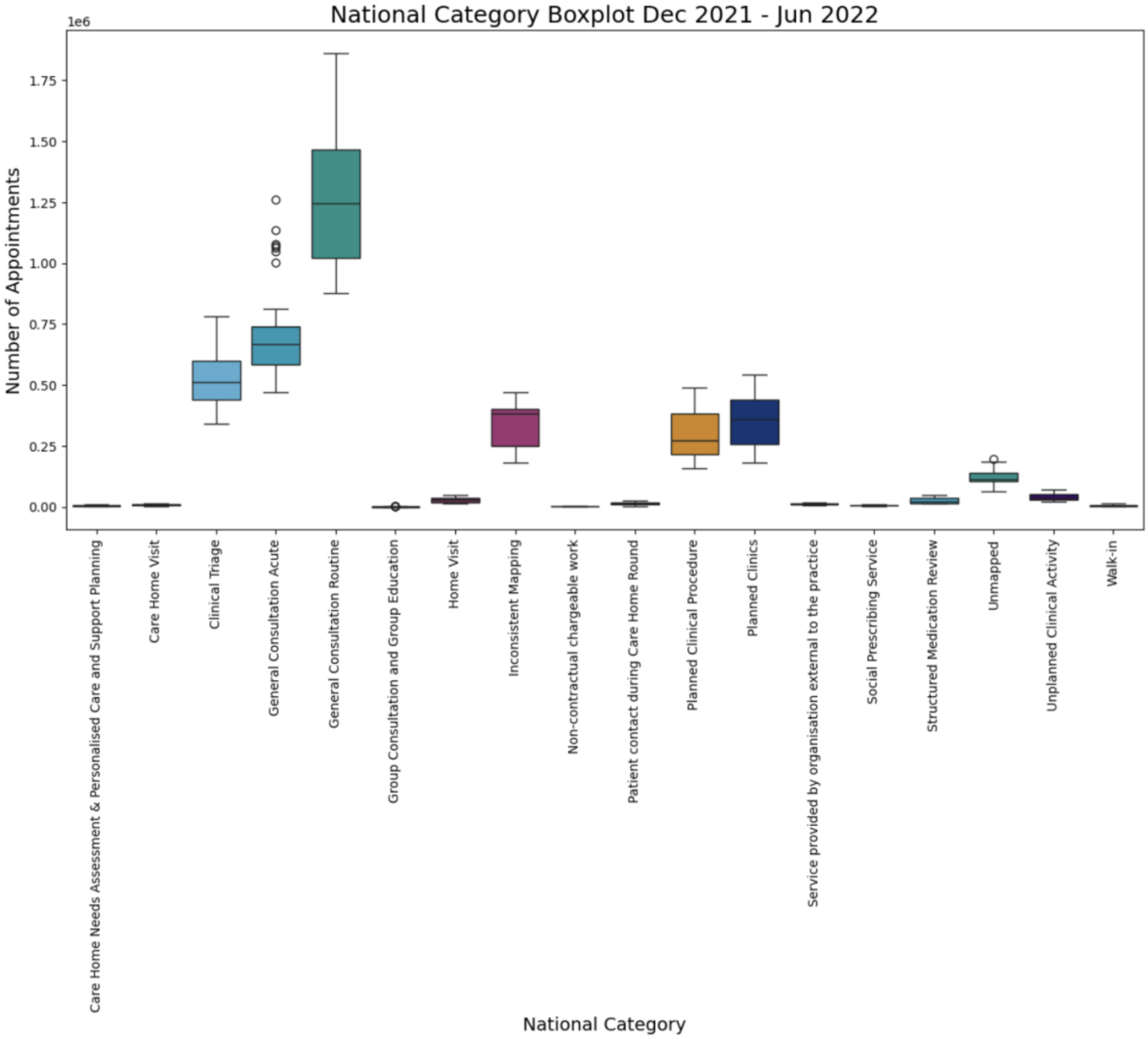


X. National Category – Appendix:

- There are 18 unique context types
- nc DataFrame was filtered to include appointments between Dec 2021 – Jun 2022
- General Consultation routine is the national category with the highest number of appointments
- Followed by appointments categorized under general consultation acute and clinical triage

	count_of_appointments	percentage
national_category		
General Consultation Routine	61419764	33.57
General Consultation Acute	34466994	18.84
Clinical Triage	25568097	13.97
Planned Clinics	17105825	9.35
Inconsistent Mapping	17067801	9.33
Planned Clinical Procedure	14339997	7.84
Unmapped	5887700	3.22
Unplanned Clinical Activity	1943489	1.06
Home Visit	1365406	0.75
Structured Medication Review	1227033	0.67
Patient contact during Care Home Round	641365	0.35
Service provided by organisation external to the practice	547716	0.30
Care Home Visit	411648	0.22
Social Prescribing Service	329585	0.18
Care Home Needs Assessment & Personalised Care and Support Planning	271457	0.15
Walk-in	243049	0.13
Non-contractual chargeable work	88124	0.05
Group Consultation and Group Education	38144	0.02

National Category Boxplot

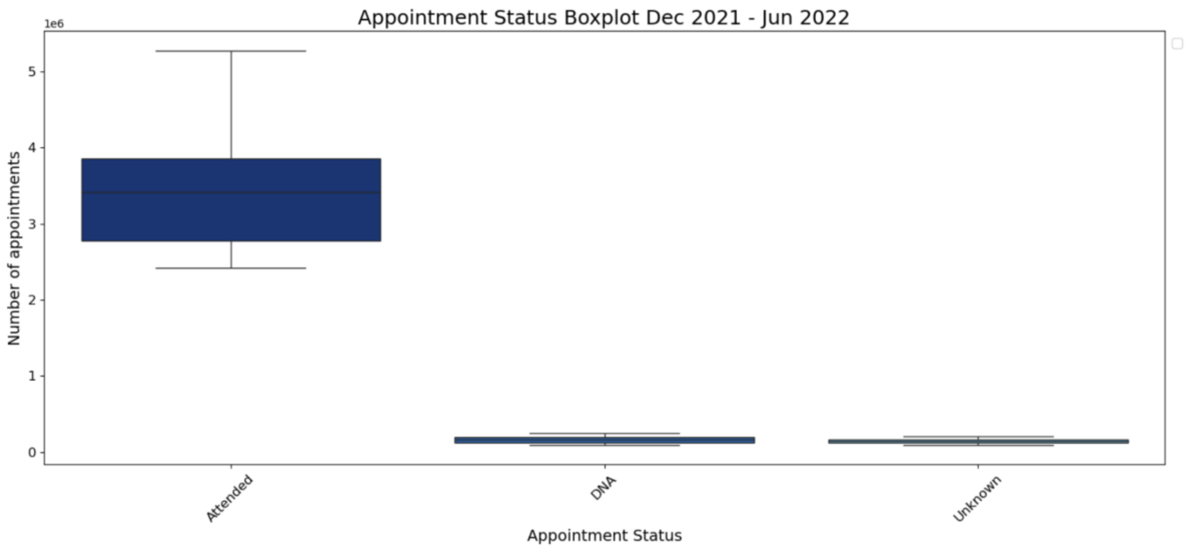


XI. Appointment Status – Appendix:

- There are 3 unique appointment statuses
- The ar DataFrame was filtered to only include appointments between Dec 2021 – Jun 2022
- Attended appointment have the largest number of appointments under appointment status
- Followed by the DNA appointments
- The missed appointments consist 4.4% of the total number of appointments between Dec 2021 - Jun 2022

	appointment_status	count_of_appointments	percentage
0	Attended	167980692	91.81
1	DNA	8054188	4.40
2	Unknown	6928314	3.79

Appointment Status Boxplot:

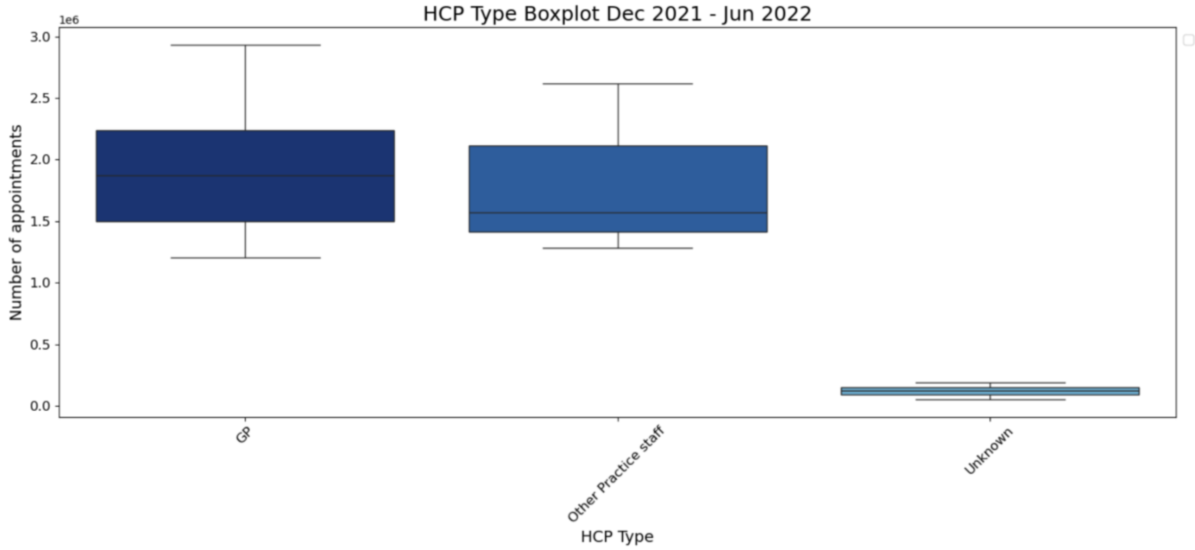


XII. HCP Type – Appendix:

- There are 3 unique HCP types
- The ar DataFrame was filtered to only include appointments between Dec 2021 – Jun 2022
- GP is the hcp type with the highest number of appointments
- Followed by other practice staff

	hcp_type	count_of_appointments	percentage
0	GP	92386135	50.49
1	Other Practice staff	84682917	46.28
2	Unknown	5894142	3.22

HCP Type Boxplot

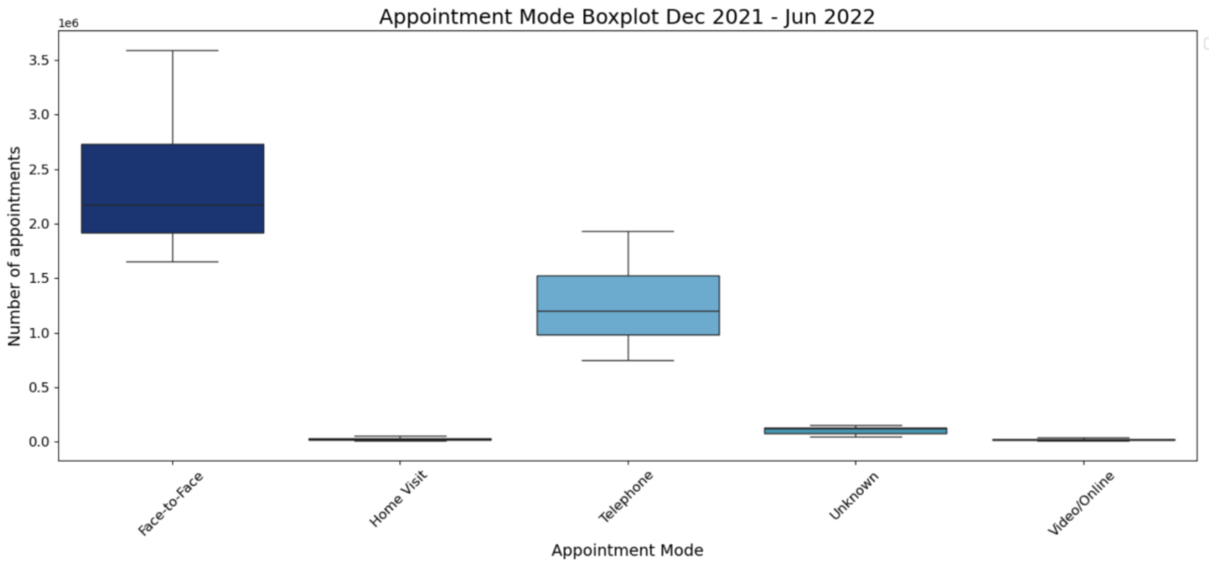


XIII. Appointment Mode – Appendix

- There are 5 unique appointment modes
- The ar DataFrame was filtered to only include appointments between Dec 2021 – Jun 2022
- Face-to-Face is the appointment mode with the highest number of appointments
- Followed by the telephone appointment mode
- Face-to-Face are within range of what the NHS has published (2)

	appointment_mode	count_of_appointments	percentage
0	Face-to-Face	114016772	62.32
1	Telephone	61613653	33.68
2	Unknown	5103949	2.79
3	Home Visit	1285751	0.70
4	Video/Online	943069	0.52

Appointment mode Boxplot



XIV. Average Time between Booking and Consultation date – Appendix:

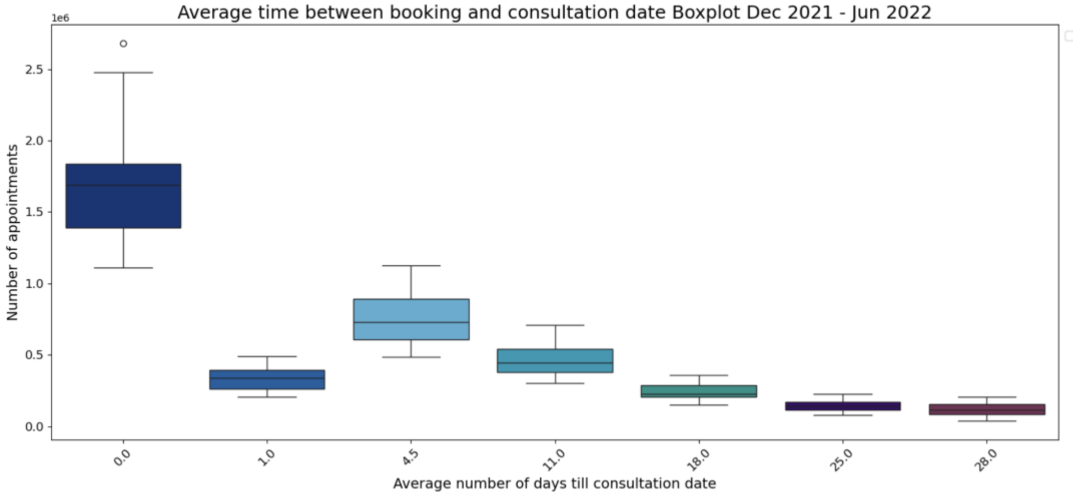
- There are 8 unique time brackets for the average time between booking and consultation

0	Same Day Appointments
1	Next Day Appointments
4.5	4.5 days on avg between booking and consultation
11	11 days on avg between booking and consultation
18	18 days on avg between booking and consultation
25	25 days on avg between booking and consultation
28	28 days on avg between booking and consultation

- This chart is relying on taking the average of the extracted integers from the time between book and appointment column.
- Plotting the averages make the visualizations easier to read
- The ar DataFrame was filtered to only include appointments between Dec 2021 – Jun 2022
- The top highest number of appointments are scheduled within a week
- Same day appointments are the highest in this case

	avg_time_between_book_appointment	count_of_appointments	percentage
0	0.0	82554607	45.15
1	4.5	36707478	20.08
2	11.0	22684786	12.41
3	1.0	16083535	8.80
4	18.0	11871816	6.49
5	25.0	7067239	3.87
6	28.0	5868098	3.21

Average Time between Book and Consultation Boxplot



Average Appointment Duration – Appendix

- There are 7 unique appointment duration

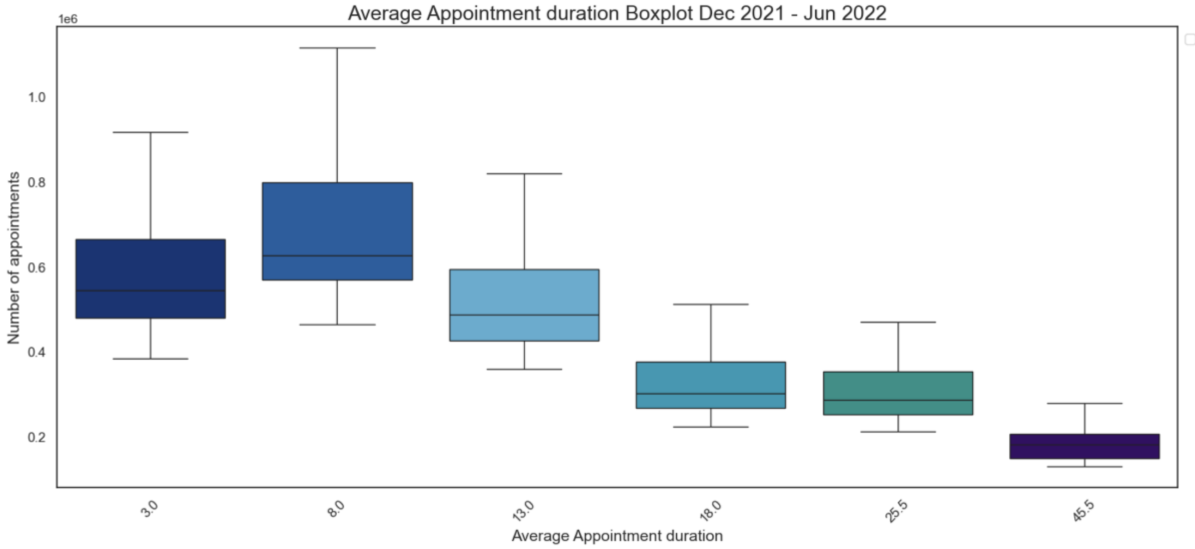
3	Appointment duration 3 mins on average
8	Appointment duration 8 mins on average
13	Appointment duration 3 mins on average
18	Appointment duration 3 mins on average
25.5	Appointment duration 3 mins on average
45.5	Appointment duration 3 mins on average

- Appointments with an average duration of 8 mins are the highest in number of appointments

- Followed by appointments with an average duration of 3 mins

	average_duration	count_of_appointments	percentage
0	8.0	33800815	26.47
1	3.0	28600865	22.40
2	13.0	25160882	19.70
3	18.0	16004247	12.53
4	25.5	15026365	11.77
5	45.5	9103432	7.13

Average Appointment Duration – Boxplot



XV. Regions – Appendix:

- There are 7 unique regions.

The region with the highest number of appointments

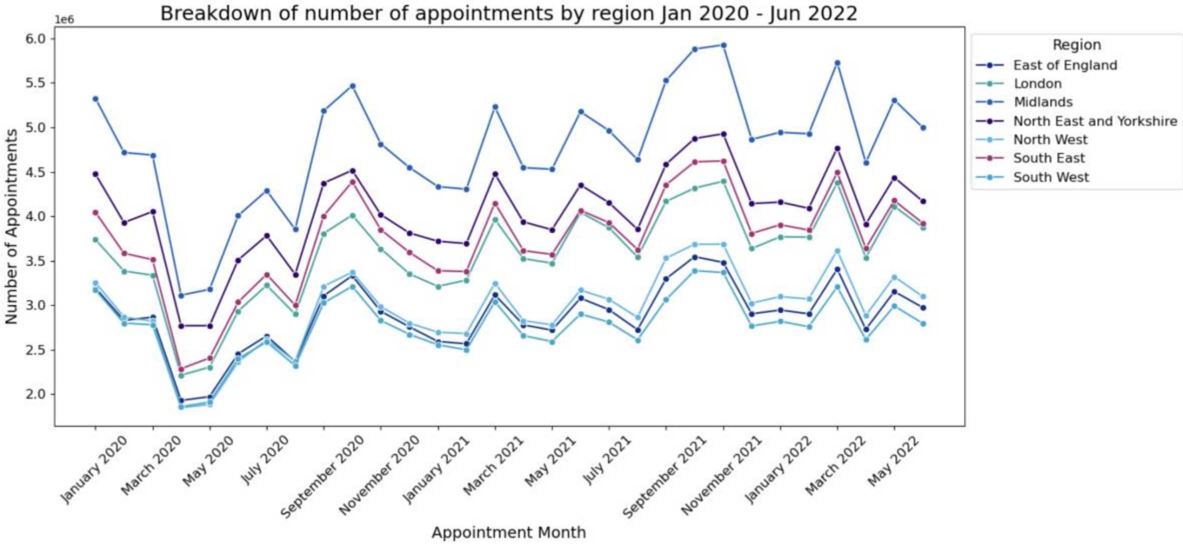
The aggregation is using the ar DataFrame as it's the DataFrame with the longest date range.

- **Midlands** has the highest number of appointments
- **London** comes 4th by number of appointments

region	count_of_appointments
Midlands	143625196
North East and Yorkshire	121448860
South East	112144789
London	107684862
North West	88702868
East of England	86228648
South West	82969302

Plotting the data for the previous question as a time series, we can draw the following insights:

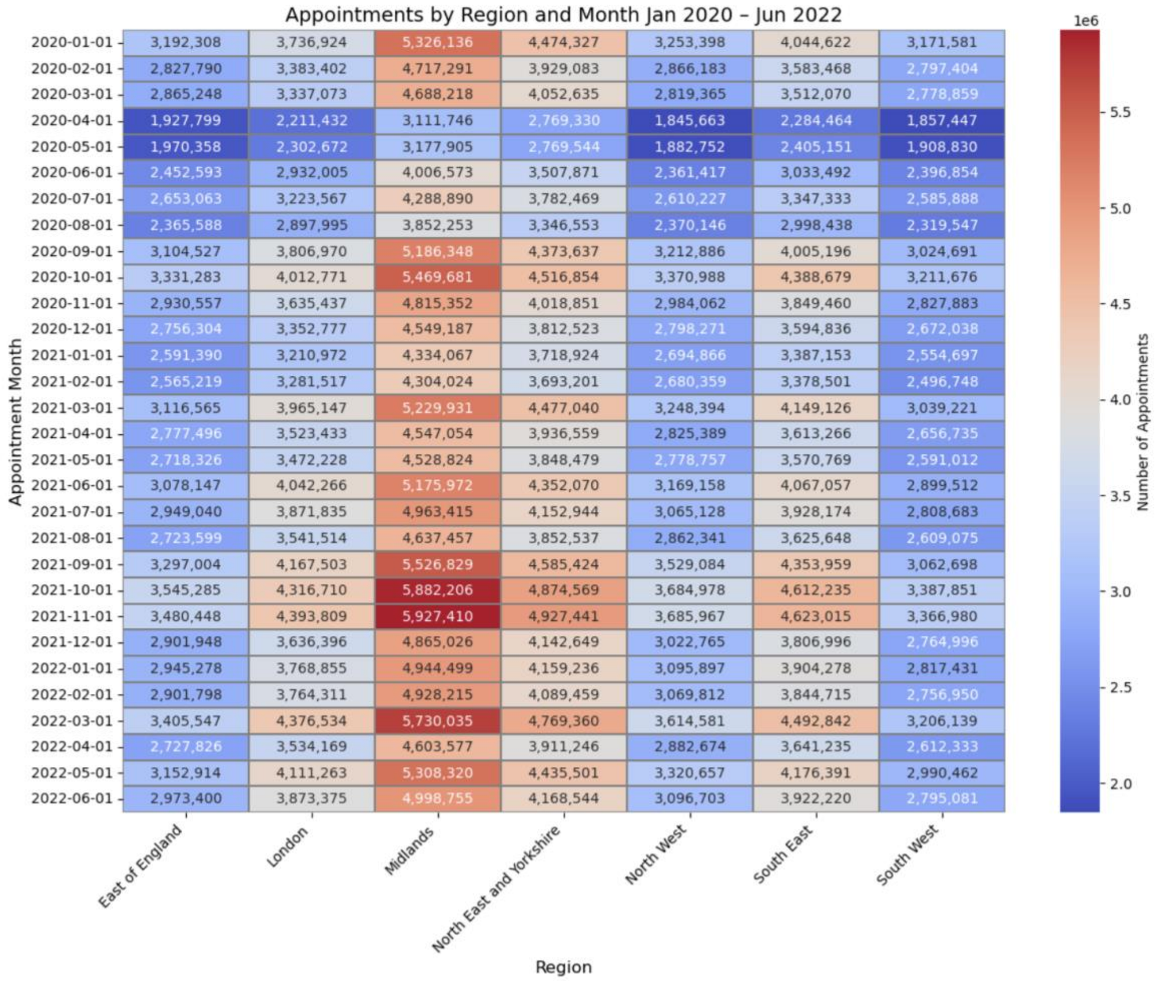
- All regions have a consistent pattern across tracked months
- **Midlands** is the highest in terms of number of appointments and there is a big gap between it and all other regions
- **The set is then divided in two main categories:**
- **London, North West and North East and Yorkshire** are close in number of appointments especially in 2022 and they fall between the Midlands and the lowest set of regions.
- **North West, South West and East England** are similar in number of appointments and they're the regions with the lowest number of appointments.
- **North West, South West and East England** almost overlapped in 2020 during the onset of the pandemic and till **September 2020**



Checking the DataFrame with the longest date range would allow us to see trends across seasons, years and special cases like the pandemic period.

This is why the answer to this question relies on the **ar DataFrame**.

Plotting the DataFrame on a heatmap shows the regions and months with the highest number of appointments clearly as it also shows the actual number of appointments.

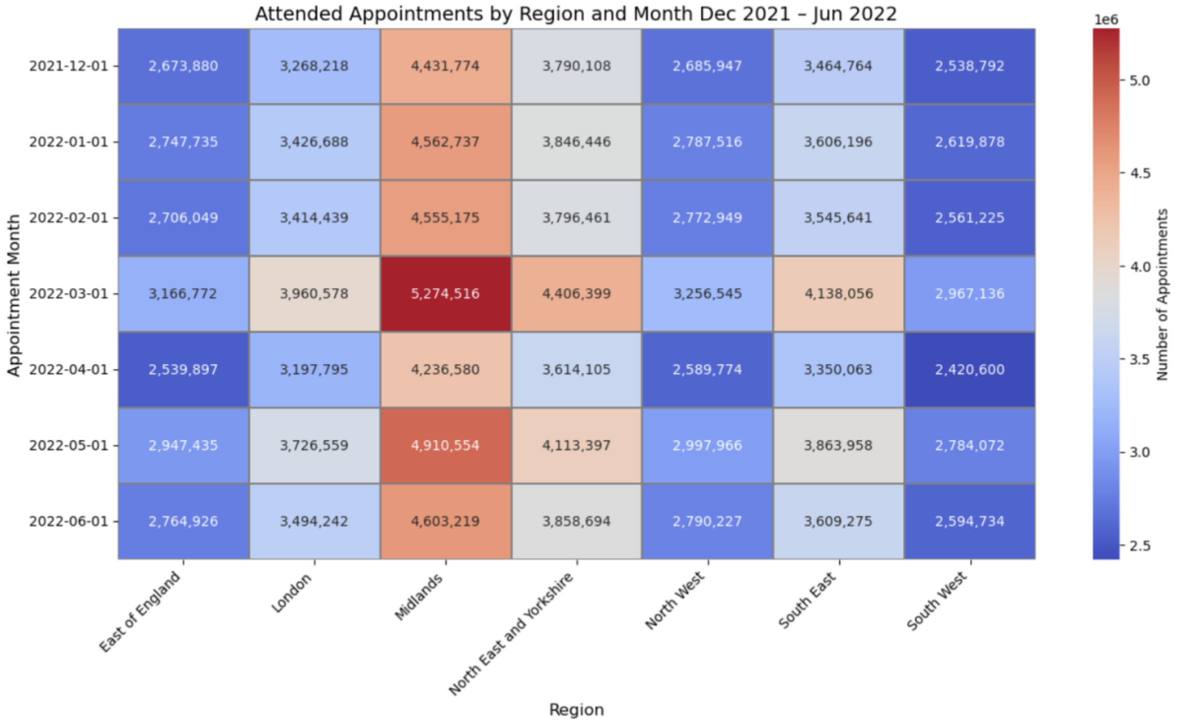


Attended Appointments between Dec 2021 – Jun 2022

The following heatmap only focuses on the attended appointments across regions and over the course of Dec 2021 - Jun 2022

The sudden surge in the number of appointments in March 2022 is confirmed by the following source: (3)

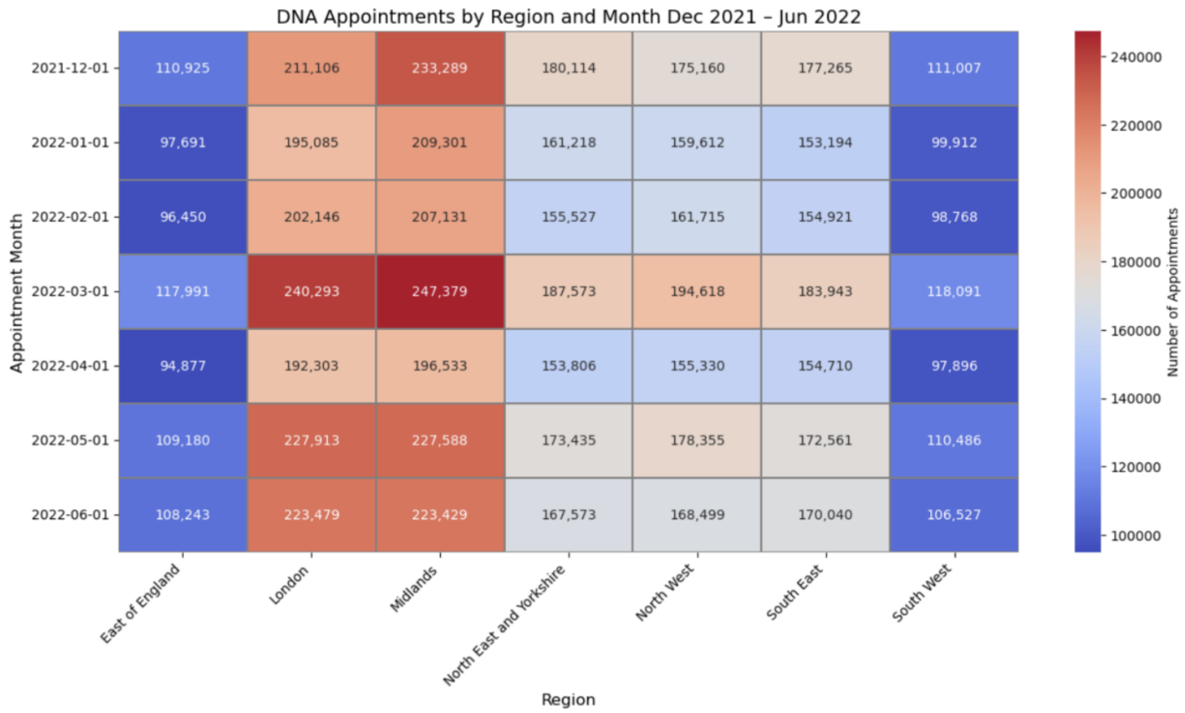
Another source confirms that March 2022 witnessed a record in staff decrease: (4)



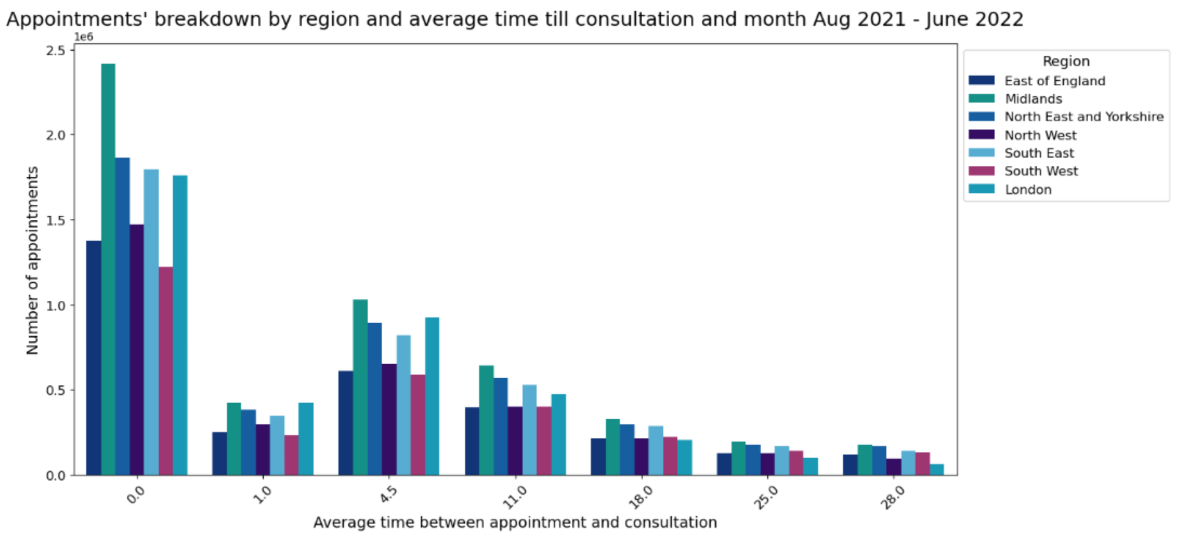
Attended Appointments by Region and Month Dec 2021 – Jun 2022
Appointment Month
Region
Number of Appointments
1e6
5.0
4.5
4.0
3.5
3.0
2.5
East of England
London
Midlands
North East and Yorkshire
North West
South East
South West
2021-12-01
2,673,880
3,268,218
4,431,774
3,790,108
2,685,947
3,464,764
2,538,792
2022-01-01
2,747,735
3,426,688
4,562,737
3,846,446
2,787,516
3,606,196
2,619,878
2022-02-01
2,706,049
3,414,439
4,555,175
3,796,461
2,772,949
3,545,641
2,561,225
2022-03-01
3,166,772
3,960,578
5,274,516
4,406,399
3,256,545
4,138,056
2,967,136
2022-04-01
2,539,897
3,197,795
4,236,580
3,614,105
2,589,774
3,350,063
2,420,600
2022-05-01
2,947,435
3,726,559
4,910,554
4,113,397
2,997,966
3,863,958
2,784,072
2022-06-01
2,764,926
3,494,242
4,603,219
3,858,694
2,790,227
3,609,275
2,594,734

DNA Appointments The following heatmap only focuses on the DNA appointments across regions and over the course of Dec 2021 - Jun 2022 The heatmap shows the following:

- **Midlands and London** are the highest regions with unattended appointments
- This is especially the case in **March 2022**



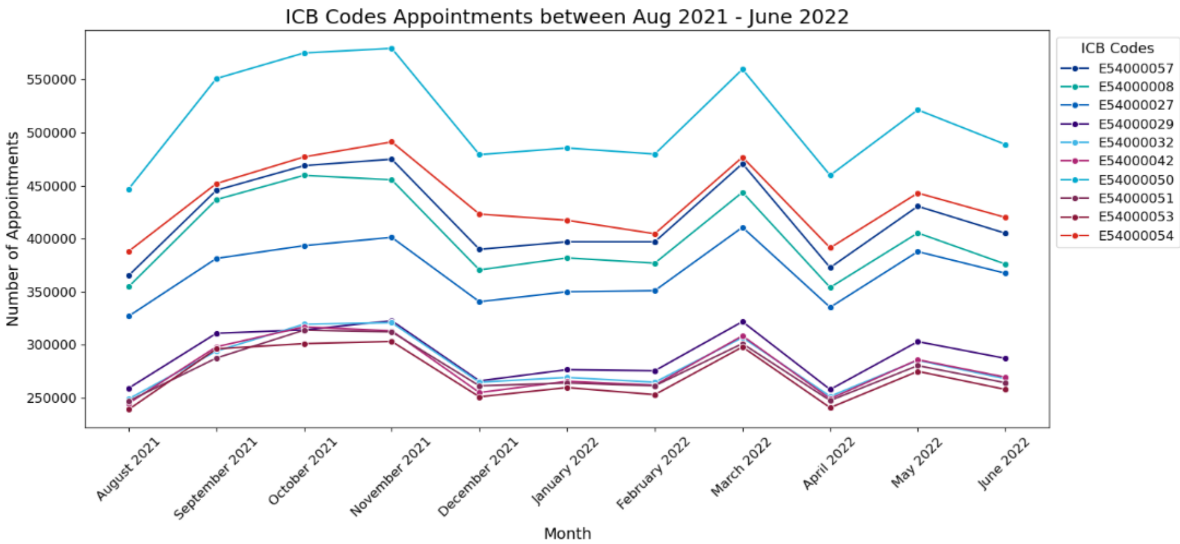
Appointments' breakdown by region and average time till consultation



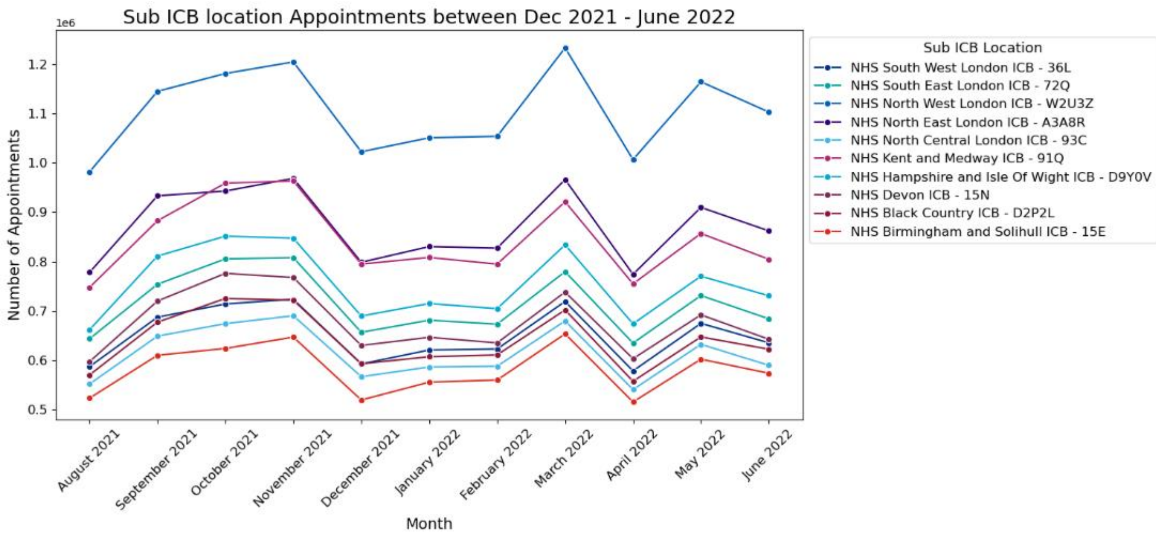
Location

Lineplot of the top 10 ICB Codes by number of appointments between August 2021 - June 2022

- The chart shows a peak in November 2021 and in March 2022.



Five sub-locations with the highest number of appointments between 01.08.2021 - 30.06.2022?



XVI. Commonalities between DataFrames – Appendix:

ad & nc DataFrames:

- By filtering the ar DataFrame to only. include appointments between 01.08.2021 – 30.06.2022, it was found that it matches the number of appointments in the nc DataFrame
- Both DataFrames were grouped by month, season and icb ons code and the compared.

nc DataFrame after grouping:

	appointment_month_str	season_year	icb_ons_code	count_of_appointments
0	November 2021	Autumn 2021	E54000050	1738485
1	October 2021	Autumn 2021	E54000050	1725403
2	March 2022	Spring 2022	E54000050	1679342
3	September 2021	Autumn 2021	E54000050	1653075
4	May 2022	Spring 2022	E54000050	1564541
...	...	...	...	...
457	February 2022	Winter 2022	E54000011	220679
458	January 2022	Winter 2022	E54000011	218581
459	December 2021	Winter 2021	E54000011	217784
460	August 2021	Summer 2021	E54000011	209824
461	April 2022	Spring 2022	E54000011	207690

ar DataFrame after grouping:

	appointment_month_str	season_year	icb_ons_code	count_of_appointments
0	November 2021	Autumn 2021	E54000050	1738485
1	October 2021	Autumn 2021	E54000050	1725403
2	March 2022	Spring 2022	E54000050	1679342
3	September 2021	Autumn 2021	E54000050	1653075
4	May 2022	Spring 2022	E54000050	1564541
...	...	...	...	...
457	February 2022	Winter 2022	E54000011	220679
458	January 2022	Winter 2022	E54000011	218581
459	December 2021	Winter 2021	E54000011	217784
460	August 2021	Summer 2021	E54000011	209824
461	April 2022	Spring 2022	E54000011	207690

To confirm the match between both DataFrames each row containing the count of appointments was subtracted from the other DataFrame.

```python
# Comparison of the grouped DataFrames by subtraction of each element (row by row) of the count of appointment column
comparison = nc_group['count_of_appointments'] - filtered_ar_nc_group ['count_of_appointments']

# Print the result of the comparison
print('The difference between the grouped DataFrames: ',
      comparison.sum(),
      'which means they match on regions and number of appointment per date')
```

```
The difference between the grouped DataFrames:  0 which means they match on regions and number of appointment per date
```

The return value for this test resulted in 0 which confirms that both DataFrames match in the number of appointments for each region and month.

ar & ad DataFrames

- The ad DataFrame only carries data for the attended appointments
- By filtering the ar DataFrame to only include attended appointments between 01.12.2021 and 30.06.2022, it was found that both DataFrames match in the number of appointments
- Both DataFrames were grouped for month, season and icb ons code, to check their match

ar DataFrame:

	appointment_month_str	season_year	icb_ons_code	count_of_appointments
0	March 2022	Spring 2022	E54000050	1545887
1	May 2022	Spring 2022	E54000050	1443421
2	June 2022	Summer 2022	E54000050	1349404
3	January 2022	Winter 2022	E54000050	1340200
4	February 2022	Winter 2022	E54000050	1330642
...	...	...	...	...
289	February 2022	Winter 2022	E54000011	205046
290	June 2022	Summer 2022	E54000011	203745
291	January 2022	Winter 2022	E54000011	203005
292	December 2021	Winter 2021	E54000011	199845
293	April 2022	Spring 2022	E54000011	192134

Failed Merge attempt for the filtered ar DataFrame & nc for the date range 01.08.2021 - 30.06.2022

The logic behind the aggregation is to fix the icb_ons_code and the date, pivot the other columns, and aggregate them by summing their number of appointments. This was done to both the nc and filtered_ar_nc DataFrames to prepare them for the merge.

- Pivoting the tables results in a multi-index that's why flattening the columns was necessary.
- The Merge attempt failed and the pivoted results were not used for further calculations to not jeopardize the integrity of the data.

Pivoting the nc DataFrame

```
# Aggregate nc
# Pivot table in preparation for a join
nc_pivot = nc.pivot_table(index =
                          ['appointment_month',
                           'icb_ons_code']
                          ,columns = ['service_setting',
                                      'context_type',
                                      'national_category']
                          ,values = ['count_of_appointments'],
                          aggfunc = 'sum'
                         )

# Reset the index of the pivoted DataFrame
nc_pivot=nc_pivot.reset_index()

# Flatten MultiIndex columns
nc_pivot.columns = [
    '_'.join([str(i) for i in col if i and i != 'count_of_appointments']).strip('_')
    if isinstance(col, tuple) else str(col)
    for col in nc_pivot.columns
]
```

Meting the nc DataFrame:

This is to get a vertical DataFrame instead of the wide DataFrame created by the Pivoting

```python
# Melt the nc_pivot in prep for merge
nc_melt = pd.melt(nc_pivot,
                  id_vars=['appointment_month',
                           'icb_ons_code'],
                  value_name= 'count_of_appointments',
                  var_name='appointments'
                 )

# Test the nc_melt DataFrame by checking the sum of count of appointments
nc_melt['count_of_appointments'].sum()
nc_melt_adj = nc_melt

nc_melt_adj
```

	appointment_month	icb_ons_code	appointments	count_of_appointments
0	2021-08-01	E54000008	Extended Access Provision_Care Related Encount...	NaN
1	2021-08-01	E54000010	Extended Access Provision_Care Related Encount...	NaN
2	2021-08-01	E54000011	Extended Access Provision_Care Related Encount...	NaN
3	2021-08-01	E54000013	Extended Access Provision_Care Related Encount...	NaN
4	2021-08-01	E54000015	Extended Access Provision_Care Related Encount...	2.0
...	...	...	...	...
31873	2022-06-01	E54000058	Unmapped_Unmapped_Unmapped	19944.0
31874	2022-06-01	E54000059	Unmapped_Unmapped_Unmapped	15376.0
31875	2022-06-01	E54000060	Unmapped_Unmapped_Unmapped	26117.0
31876	2022-06-01	E54000061	Unmapped_Unmapped_Unmapped	18514.0
31877	2022-06-01	E54000062	Unmapped_Unmapped_Unmapped	9438.0

31878 rows × 4 columns

Create a user-defined function to split the columns

```python
# Create a user-defined function to split the appointments column into three different columns and give them their correct original name
def split_func(df,
               name1,
               name2,
               name3
              ):
    # Splits the appointments column by the '_' i
    df[[name1, name2, name3]] = df['appointments'].str.split('_', expand=True)

    # Drop the original appointments column after the split
    df.drop(columns='appointments', inplace=True)
    return df
```

```python
# Apply the split function on nc_melt to separate the appointments column to three individual columns
# This is in preparation for the join with the filtered ar DataFrame

nc_split = split_func(nc_melt_adj,
                      'service_setting',
                      'context_type',
                      'national_category'
                     )
nc_split
```

	appointment_month	icb_ons_code	count_of_appointments	service_setting	context_type	national_category
0	2021-08-01	E54000008	NaN	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care...
1	2021-08-01	E54000010	NaN	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care...
2	2021-08-01	E54000011	NaN	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care...
3	2021-08-01	E54000013	NaN	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care...
4	2021-08-01	E54000015	2.0	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care...
...	...	...	...	...	...	...
31873	2022-06-01	E54000058	19944.0	Unmapped	Unmapped	Unmapped
31874	2022-06-01	E54000059	15376.0	Unmapped	Unmapped	Unmapped
31875	2022-06-01	E54000060	26117.0	Unmapped	Unmapped	Unmapped
31876	2022-06-01	E54000061	18514.0	Unmapped	Unmapped	Unmapped
31877	2022-06-01	E54000062	9438.0	Unmapped	Unmapped	Unmapped

31878 rows × 6 columns

Check the sum of appointments

```
# Recheck the sum appointments to make sure the DataFrame wasn't incorrectly manipulated
nc_join['count_of_appointments'].sum()
```

```
296046770.0
```

Repeat the same steps for the filtered ar DataFrame:

```
# Pivot table in preparation for a join
filtered_ar_pivot = filtered_ar_nc.pivot_table(index =['appointment_month',
                                                       'icb_ons_code'],
                                               columns = ['appointment_status',
                                                          'hcp_type',
                                                          'appointment_mode'],
                                               values = ['count_of_appointments'],
                                               aggfunc = 'sum'
                                              )
filtered_ar_pivot = filtered_ar_pivot.reset_index()

# Flatten the MultiIndex Columns
filtered_ar_pivot.columns = ['_'.join([str(i) for i in col if i and i != 'count_of_appointments']).strip('_')
                             if isinstance(col, tuple)
                             else col
                             for col in filtered_ar_pivot.columns]
filtered_ar_pivot
```

	appointment_month	icb_ons_code	Attended_GP_Face-to-Face	Attended_GP_Home Visit	Attended_GP_Telephone	Attended_GP_Unknown	Attended_GP_Video/Online
0	2021-08-01	E54000008	241116.0	4873.0	298572.0	7281.0	1468.
1	2021-08-01	E54000010	82390.0	564.0	116085.0	NaN	666.
2	2021-08-01	E54000011	50872.0	895.0	47929.0	3206.0	30.
3	2021-08-01	E54000013	59033.0	133.0	59624.0	NaN	439.
4	2021-08-01	E54000015	125535.0	129.0	113815.0	1.0	1512.
...	...	...	...	...	...	...	..
457	2022-06-01	E54000058	146089.0	1257.0	84144.0	1.0	1360.
458	2022-06-01	E54000059	77363.0	159.0	71944.0	NaN	348.
459	2022-06-01	E54000060	141676.0	795.0	105153.0	NaN	2808.
460	2022-06-01	E54000061	167211.0	1263.0	108251.0	NaN	3018.
461	2022-06-01	E54000062	171405.0	1515.0	137198.0	8.0	981.

462 rows × 47 columns

```python
# Apply the split function on nc_melt_adj to separate the appointments column to three individual columns
# This is in preparation for the join with the filtered ar DataFrame

ar_split = split_func(ar_melt,
                      'appointment_status',
                      'hcp_type',
                      'appointment_mode')
ar_split
```

	appointment_month	icb_ons_code	count_of_appointments	appointment_status	hcp_type	appointment_mode
0	2021-08-01	E54000008	241116.0	Attended	GP	Face-to-Face
1	2021-08-01	E54000010	82390.0	Attended	GP	Face-to-Face
2	2021-08-01	E54000011	50872.0	Attended	GP	Face-to-Face
3	2021-08-01	E54000013	59033.0	Attended	GP	Face-to-Face
4	2021-08-01	E54000015	125535.0	Attended	GP	Face-to-Face
...	...	...	...	...	...	...
20785	2022-06-01	E54000058	NaN	Unknown	Unknown	Video/Online
20786	2022-06-01	E54000059	NaN	Unknown	Unknown	Video/Online
20787	2022-06-01	E54000060	NaN	Unknown	Unknown	Video/Online
20788	2022-06-01	E54000061	NaN	Unknown	Unknown	Video/Online
20789	2022-06-01	E54000062	NaN	Unknown	Unknown	Video/Online

Merging the DataFrames

```python
merge_nc_ar_filtered = nc_join.merge(ar_join,how='left')
merge_nc_ar_filtered
```

	appointment_month	icb_ons_code	service_setting	context_type	national_category	count_of_appointments	appointment_status	hcp_type	appo
0	2021-08-01	E54000008	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care and Support Planning	NaN	Attended	GP	
1	2021-08-01	E54000008	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care and Support Planning	NaN	Attended	GP	
2	2021-08-01	E54000008	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care and Support Planning	NaN	Attended	GP	
3	2021-08-01	E54000008	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care and Support Planning	NaN	Attended	GP	
4	2021-08-01	E54000008	Extended Access Provision	Care Related Encounter	Care Home Needs Assessment & Personalised Care and Support Planning	NaN	Attended	GP	
...	...	...	...	...	...	...	...	...	
2889965	2022-06-01	E54000058	Unmapped	Unmapped	Unmapped	19944.0	NaN	NaN	
2889966	2022-06-01	E54000059	Unmapped	Unmapped	Unmapped	15376.0	NaN	NaN	
2889967	2022-06-01	E54000060	Unmapped	Unmapped	Unmapped	26117.0	NaN	NaN	
2889968	2022-06-01	E54000061	Unmapped	Unmapped	Unmapped	18514.0	NaN	NaN	
2889969	2022-06-01	E54000062	Unmapped	Unmapped	Unmapped	9438.0	NaN	NaN	

2889970 rows × 9 columns

Check after merge

Checking the sum of appointments after the merge resulted in a higher number of appointments, which is incorrect.

The merge was aborted to not jeopardize the data and the analysis.

```
# Rechecking the sum of count_of_appointment flagged an error as it doesn't match the original sum
merge_nc_ar_filtered['count_of_appointments'].sum()
```

```
296890869.0
```

ad DataFrame:

	appointment_month_str	season_year	icb_ons_code	count_of_appointments
0	March 2022	Spring 2022	E54000050	1545887
1	May 2022	Spring 2022	E54000050	1443421
2	June 2022	Summer 2022	E54000050	1349404
3	January 2022	Winter 2022	E54000050	1340200
4	February 2022	Winter 2022	E54000050	1330642
...	...	...	...	...
289	February 2022	Winter 2022	E54000011	205046
290	June 2022	Summer 2022	E54000011	203745
291	January 2022	Winter 2022	E54000011	203005
292	December 2021	Winter 2021	E54000011	199845
293	April 2022	Spring 2022	E54000011	192134

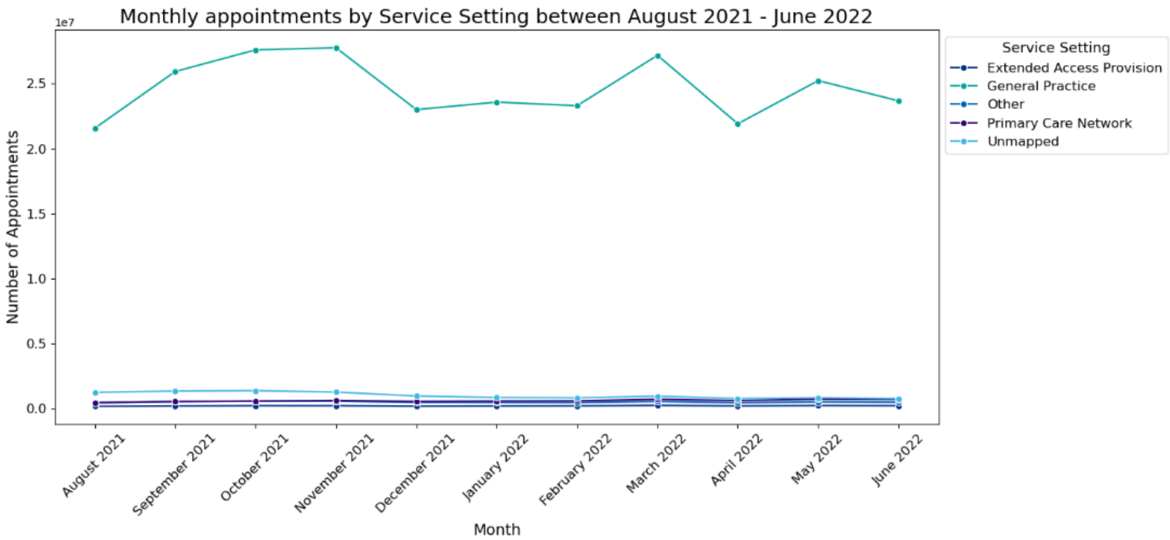
Both DataFrames were then compared to confirm the match by subtracting the count of appointments in each row. The difference between both turned out to be 0, which confirms the match based on the number of appointments per month and icb ons code.

```
# Comparison of the grouped DataFrames by subtraction
comparisonadar = filtered_ar_sum_appointments['count_of_appointments'] - ad_sum_appointments ['count_of_appointments']

print('The difference between the grouped DataFrames: ',comparison.sum(),
      'which means they match on regions and number of appointment per date')
```

```
The difference between the grouped DataFrames:  0 which means they match on regions and number of appointment per date
```

XVII. Utilization/Appointment Detail & Info Charts

Service Settings:

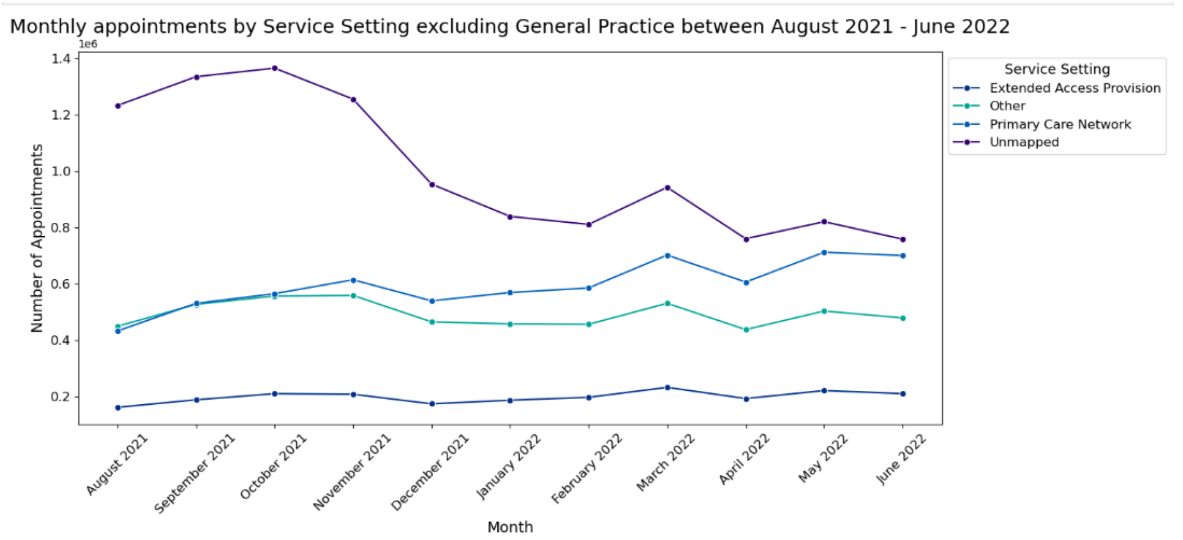


Service Settings Excluding General Practice:

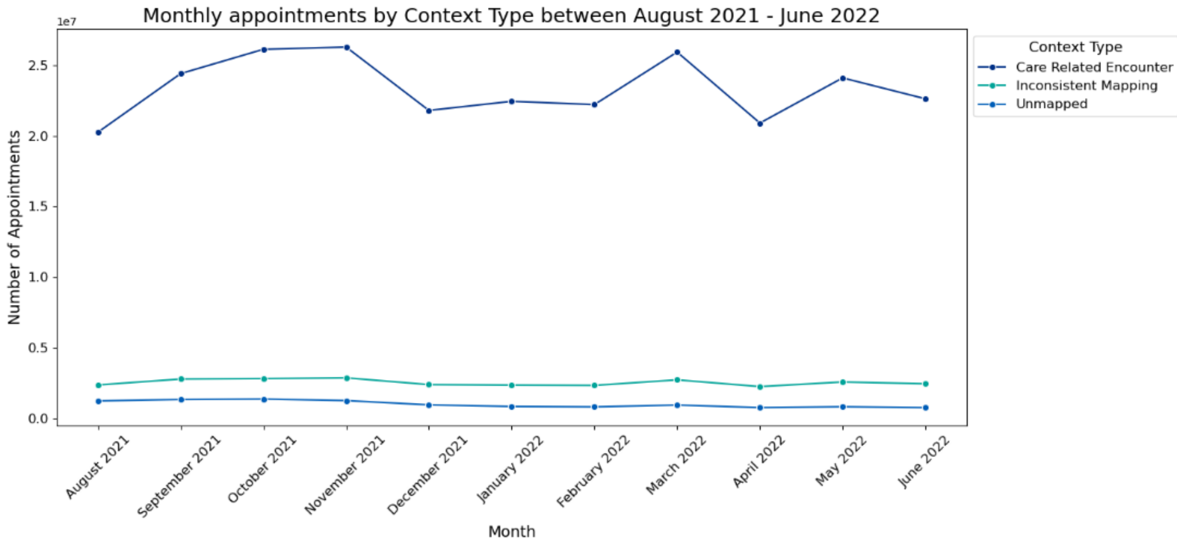
As the gap between General practice and the other service settings is big, filtering the general practice out would allow us to get a detailed look at the other service types.

- **Unmapped** appointments have a negative slope suggesting an improvement in the tracking of appointment's service settings
- **Other and Primary Care Network** appointments overlapped between August 2021 to October 20021.
- **Primary Care Network appointments** surpassed other appoointment between October 2021 to June 2022
- **Extended Care Provision** are the lowest service setting

There is a significant improvement in data quality shown by the negative slope of the Unmapped curve. This is also confirmed by the NHS's commitment to improve data security and to retrain partners in the network to make sure that the data is collected according to its set standards. (8)



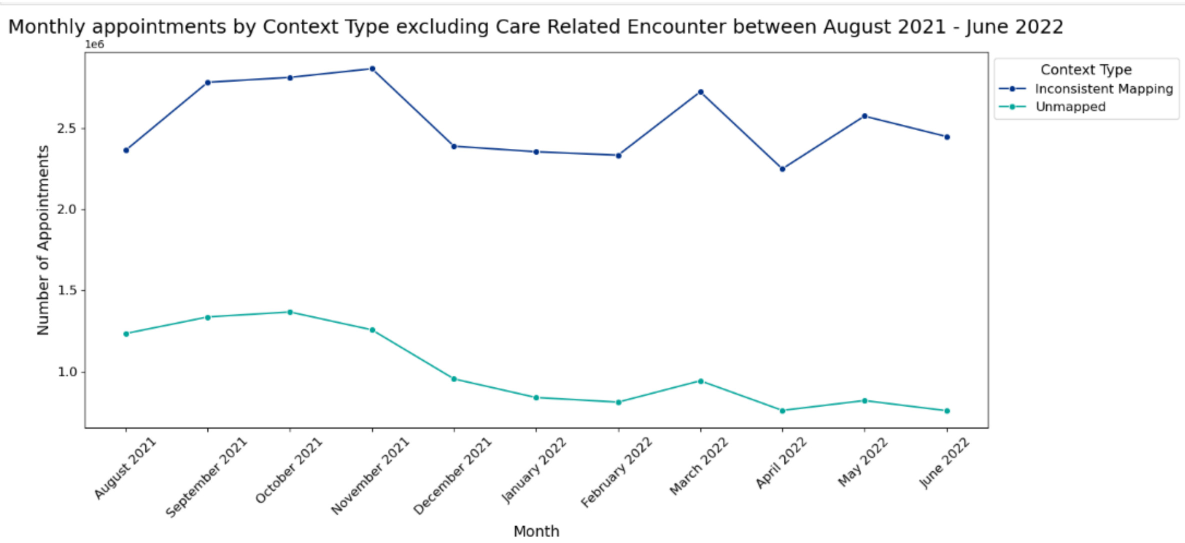
Context Type



Context Type Excluding Care Related Encounter:

Consistently with the pattern we see on the previous chart, the unmapped appointments seem to have a negative slope for the context types too.

This is suggesting a strong improvement in data collection for both context type and service setting.



National Categories

National categories: Since the national categories include 18 categories, the chart showing them all on one chart is a little cumbersome.

The chart will be split into three charts:

High Tier:

- General Consultation Routine
- General Consultation Acute
- Clinical Triage
- Planned Clinics
- Inconsistent Mapping
- Planned Clinical Procedure

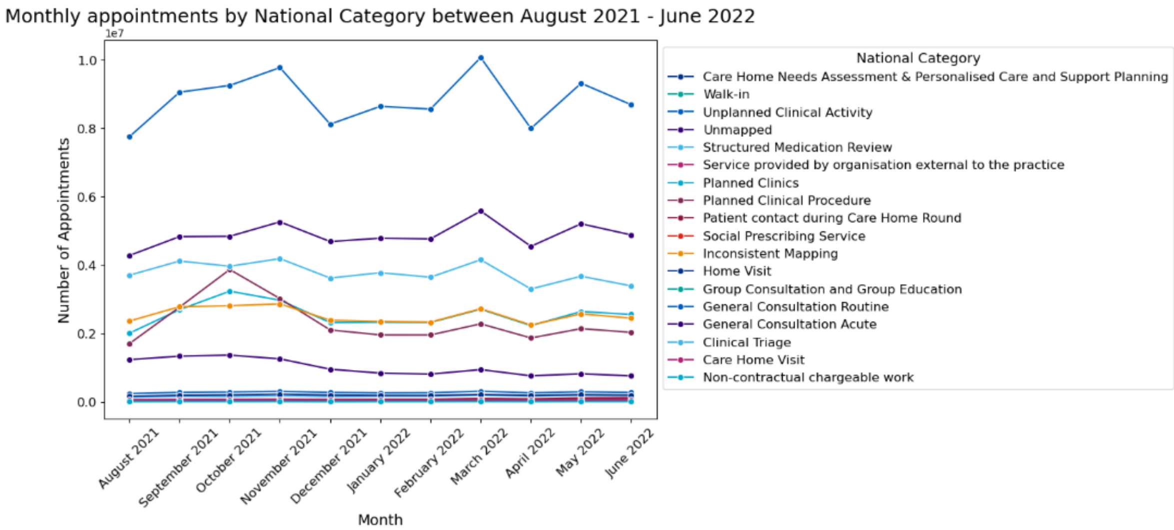
Mid Tier:

- Unmapped
- Unplanned Clinical Activity
- Home Visit

Low Tier:

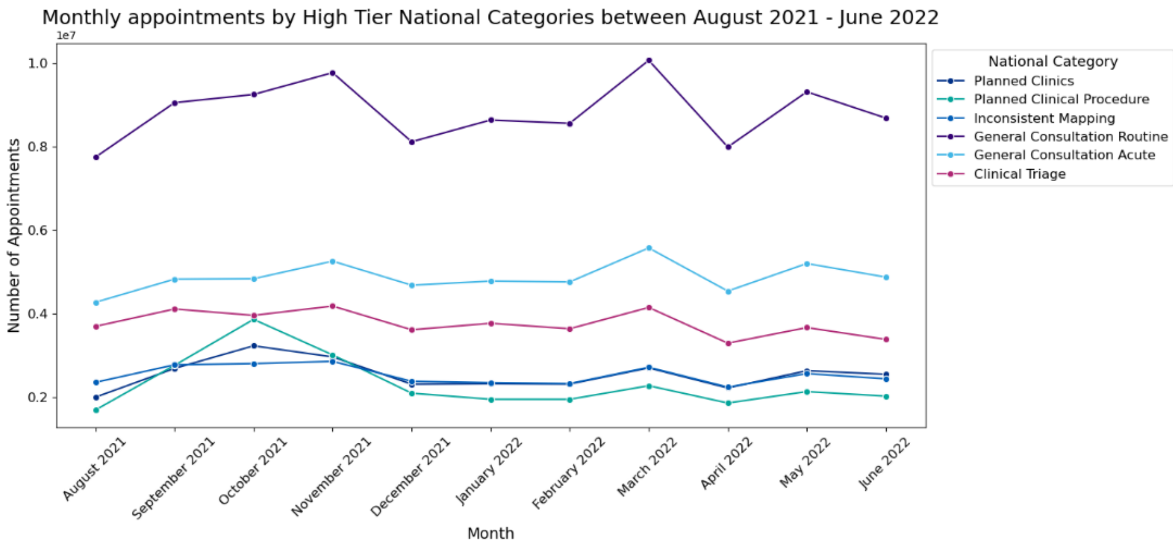
- Service provided by organisation external to the practice
- Patient contact during Care Home Round
- Care Home Visit
- Social Prescribing Service
- Walk-in

- Care Home Needs Assessment & Personalised Care & Personalised Care and Support Planning
- Non-contractual chargeable work
- Group Consultation and Group Education



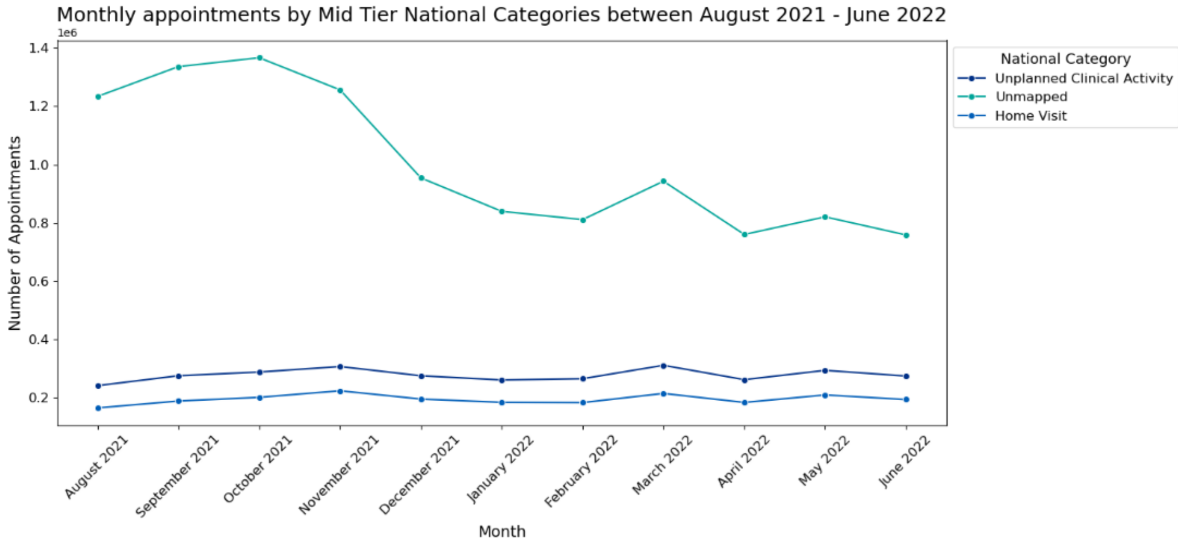
High Tier

- **Planned Clinical Procedure and Clinical Triage** have the same number of appointments in Oct 2021
- **General Consultation Acute and Clinical Triage** follow the same pattern
- **Planned Clinics and Inconsistent Mapping** overlapp between Dec 2021 and June 2022



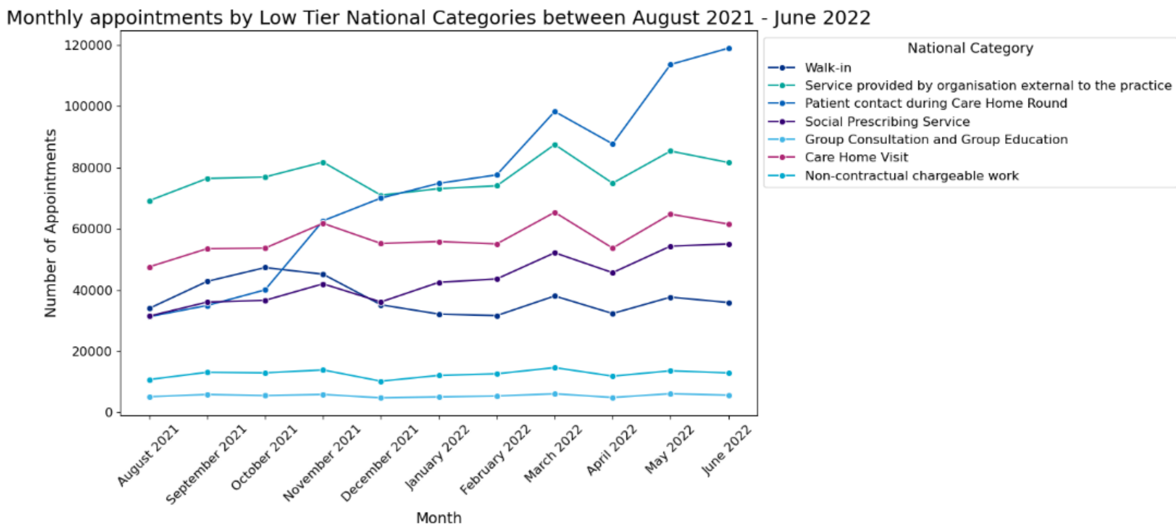
Mid Tier

- **Unmapped** is the highest in this tier
- **Home Visit and Unplanned Clinical Activity** follow the same trend for the whole date range



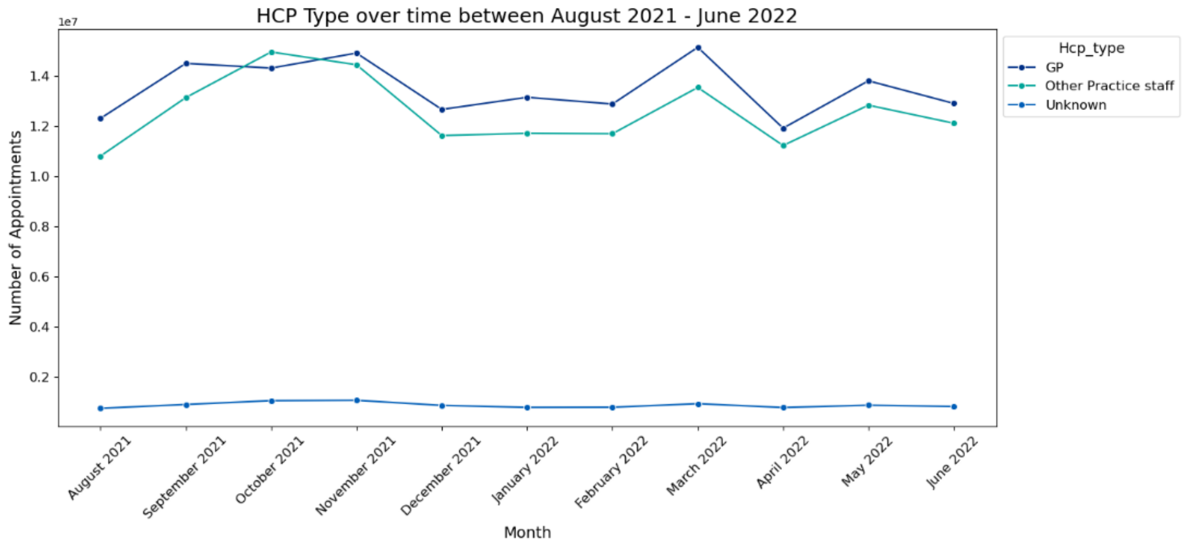
Low Tier

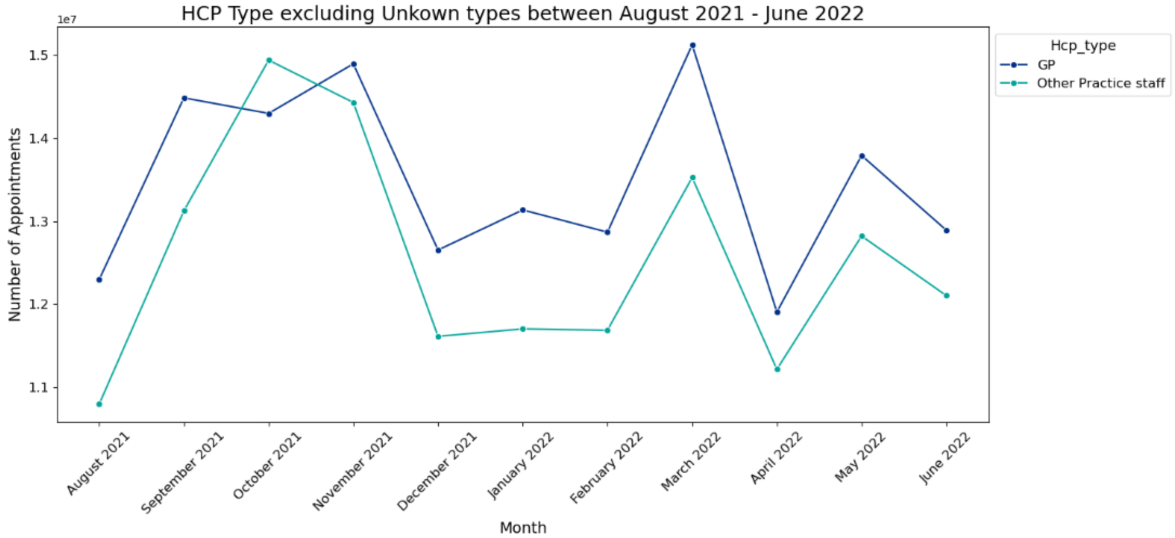
- **Patient Contact during Care Home Round** has a steep positive slope suggesting a strong change starting Oct 2021
- **Group Consultation and Group Education** are the lowest overall in terms of number of appointments



HCP Type

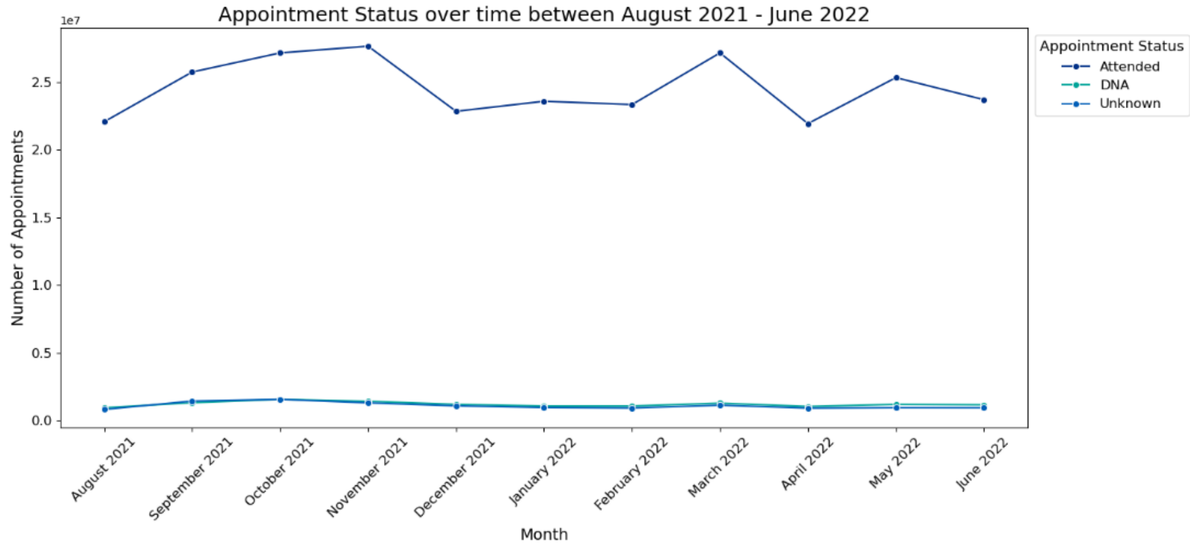
- **GP** Appointments are higher over the course of the analysed date range, except for **Oct 2021** where Other practice staff appointments exceeded them
- Appointments categorized under **Unkown** would be filtered out to better see the trend of GP and other practice staff appointments





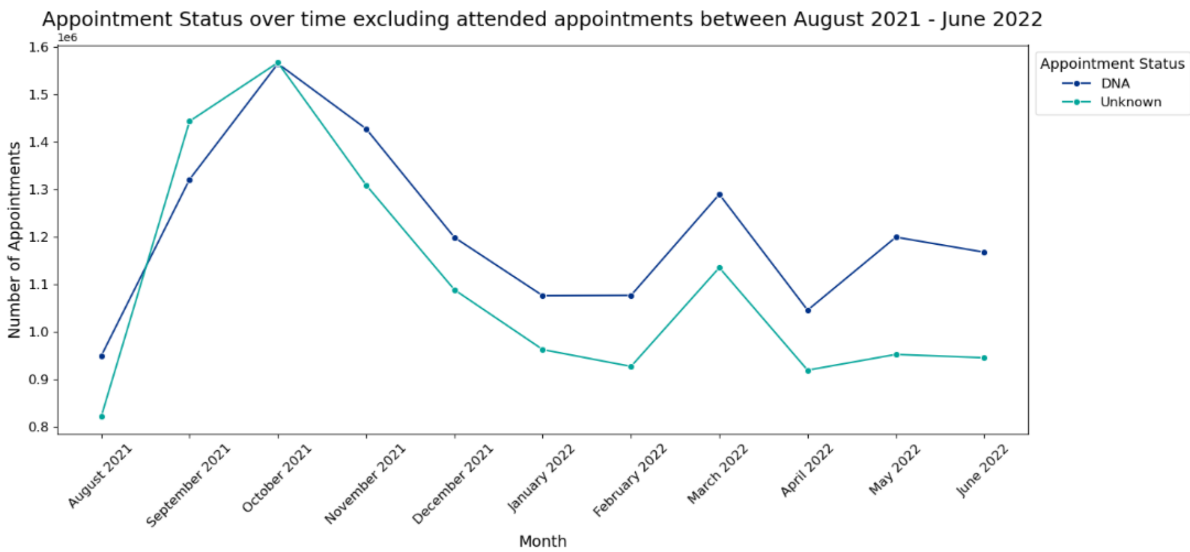
Appointment Status

- **Attended** appointments are the highest in the date range between August 2021 and June 2022
- To be able to better see the trends of DNA and Unknown appointments, the attended appointments will be filtered out on the second chart.



DNA Appointments

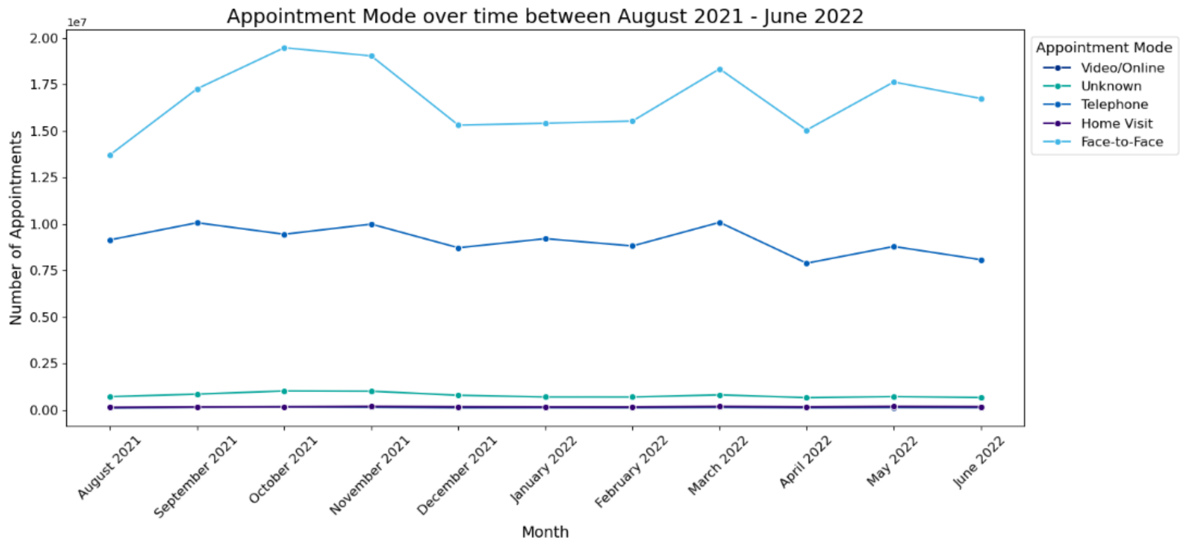
- DNA appointments were at their highest in October 2021 where they actually matched the Unknown appointments.
- Both appointment statuses started having a negative slope from their peak in Oct 2021.

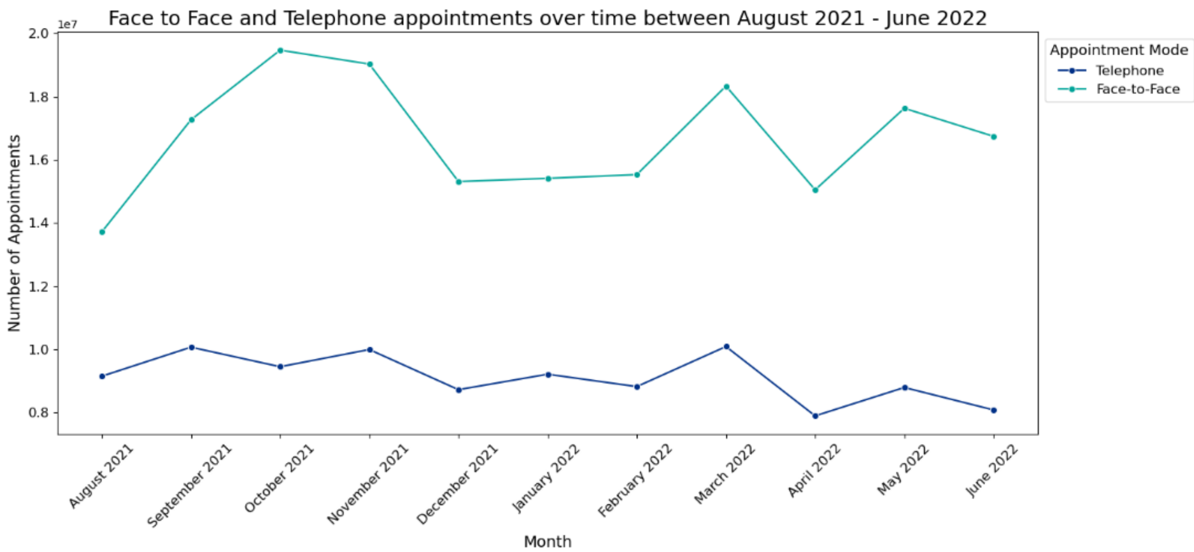


Appointment Mode

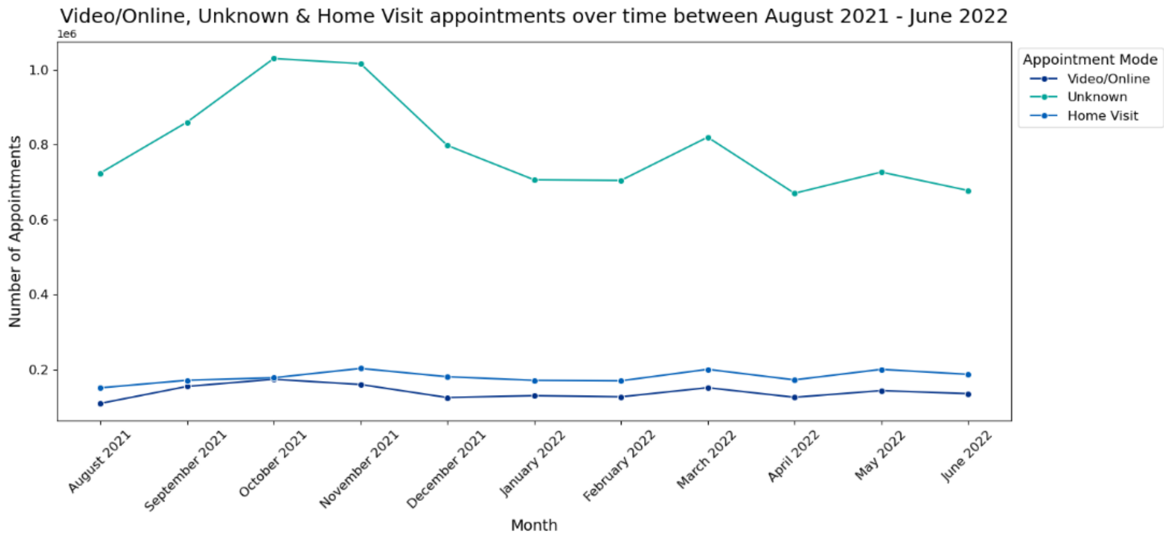
- **Face-to-Face** appointments are the highest over the course of the date range
- They are followed by **Telephone** appointments

To better see the trends of the different appointment modes, the set will be divided in: -High-Tier appointments with Face-to-Face and Telephone appointments -Low-Tier appointments with Vidoe/Online, Home visits and Unkown

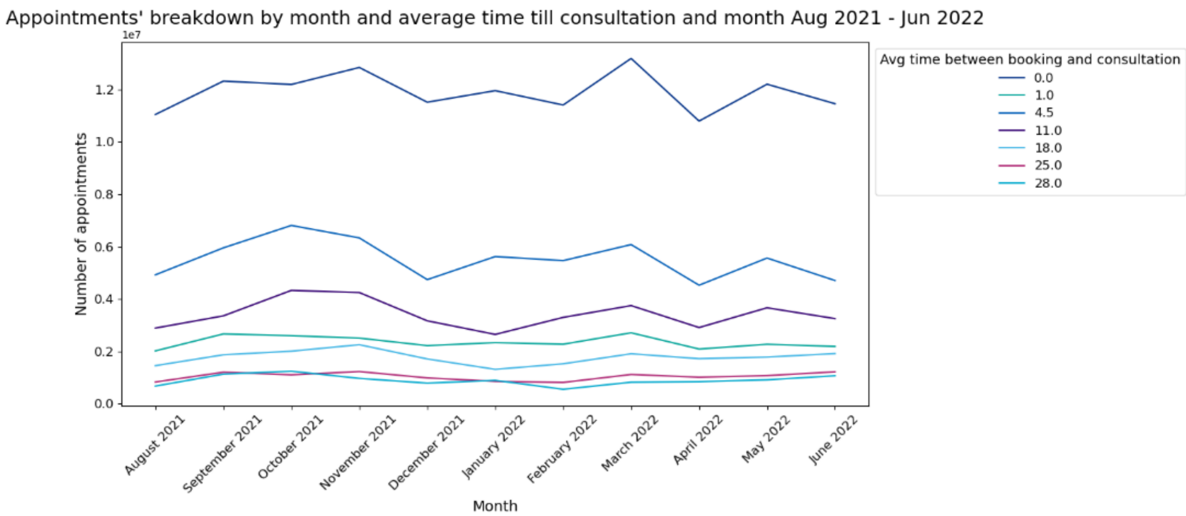




- **Unkown** appointments are higher than Video and home visits, suggesting an issue in the data collection.



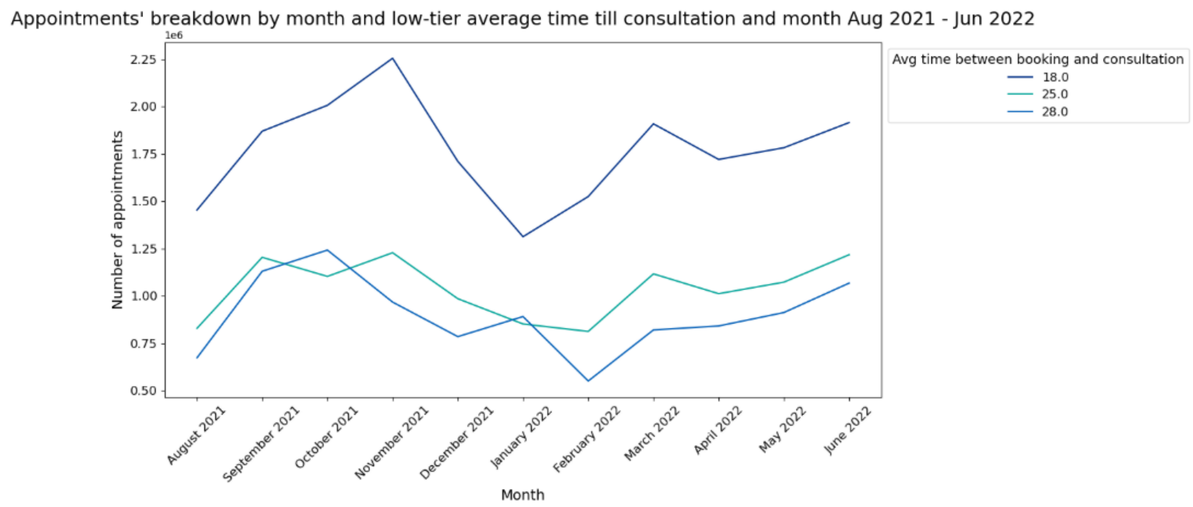
Average Time between book and appointment



Low Tier Average Time between booking and consultation

To better see the trend of appointments with a longer wait time till consultation, a low-tier subset was created to look at appointments with 18, 25 and 28 days on average between booking and consultation.

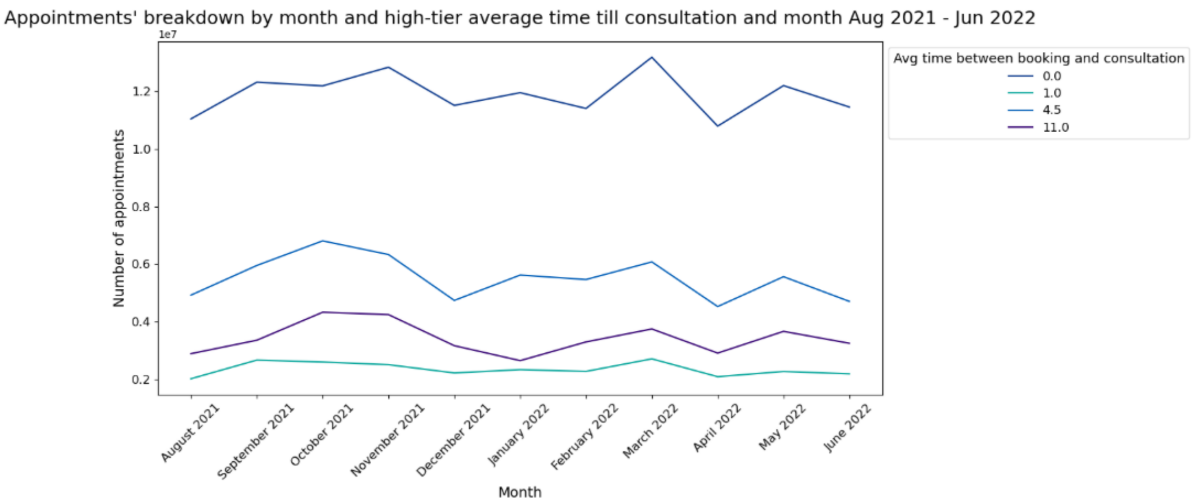
- All three curves show a positive slope towards May and June 2022 which is suggesting an expected increase in the wait times.



High-Tier average time between booking and consultation

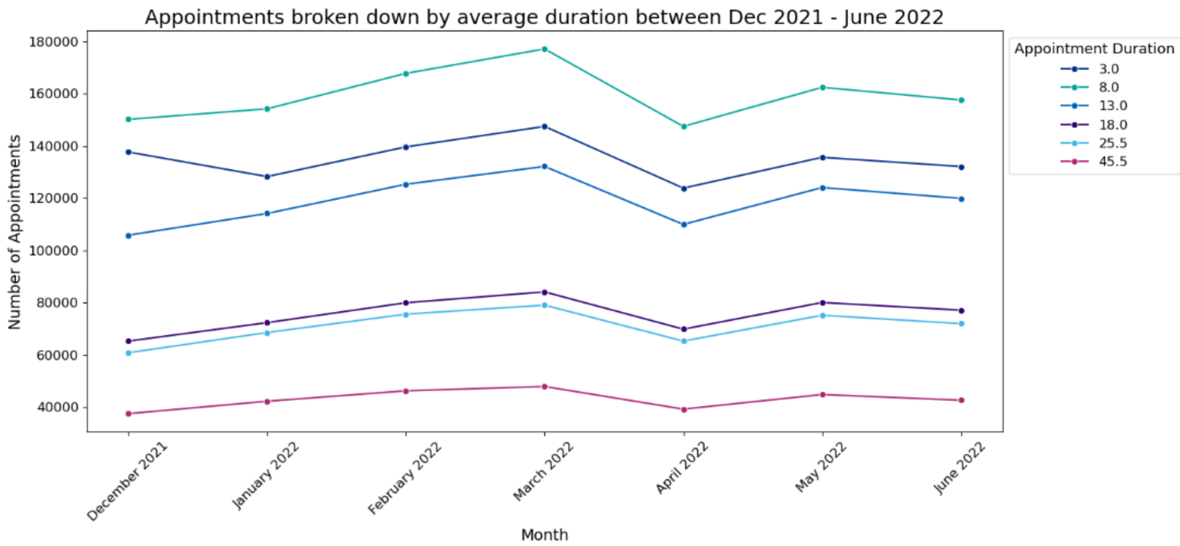
High-tier average time was plotted on another chart to show the trend of quick appointments. The tier includes appointments with a waiting time of 0, 1, 4.5 and 11 days on average till consultation.

- The chart shows a negative slope in all curves towards May and June 2022.
- This is suggesting an expected decrease in the number of quick appointments.



Average Appointment Duration

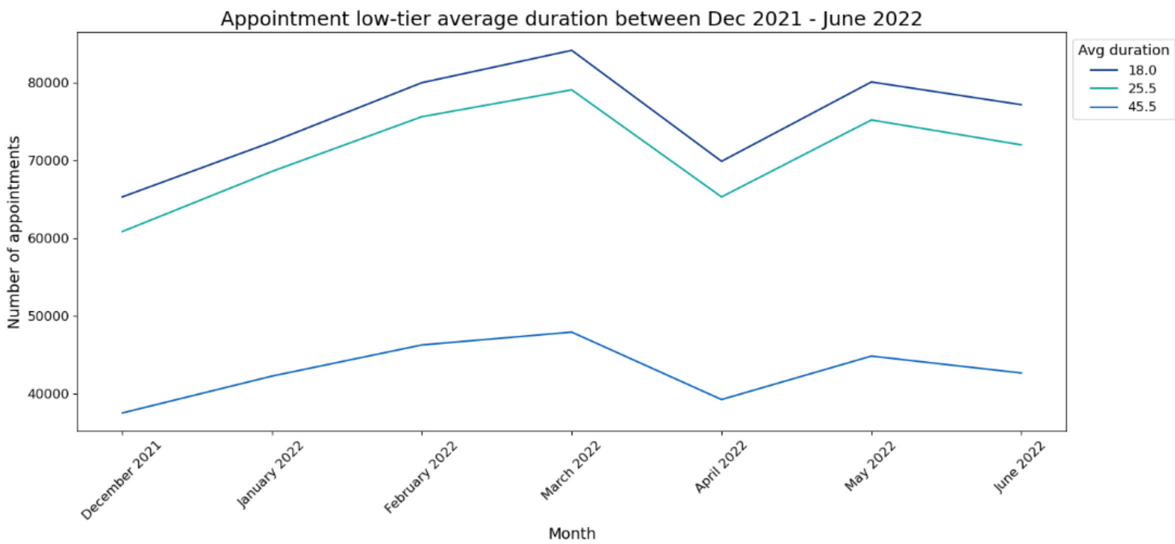
- Appointments with a duration of 8 mins are the highest within the date range Dec 2021 - June 2022.
- They are followed by 3 min and 11 mins appointments.

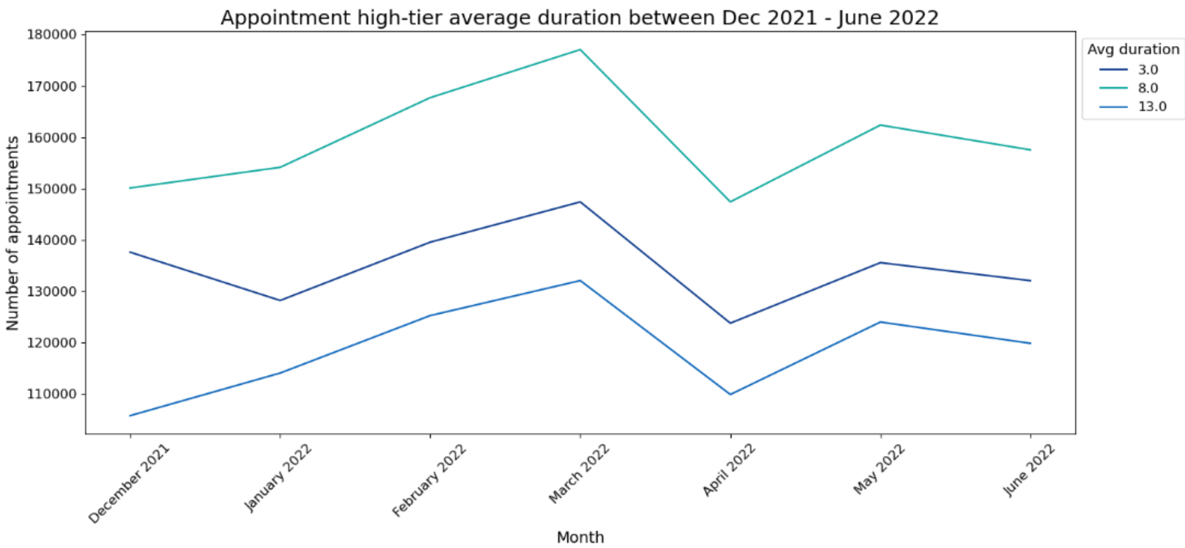


Low-Tier & High-Tier average appointment duration

The separation between low and high tier average duration appointments was made to see the lines and their trends clearly.

- Low-Tier include appointments with an average duration of 18, 25 and 45.5 minutes
- High-Tier include appointments with an average duration of 3, 8 and 13 minutes





Appointment high-tier average duration between Dec 2021 - June 2022
Number of appointments
180000
170000
160000
150000
140000
130000
120000
110000
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month
Avg duration
3.0
8.0
13.0

XVIII. Pandemic Effect on top appointment modes and top hcp types– Appendix

Appointment mode trend during the pandemic months To get a full picture of the behaviour, growth and decline of the two top appointment modes, it was important to extend the date range to include 2020 data and by this using the original ar DataFrame.

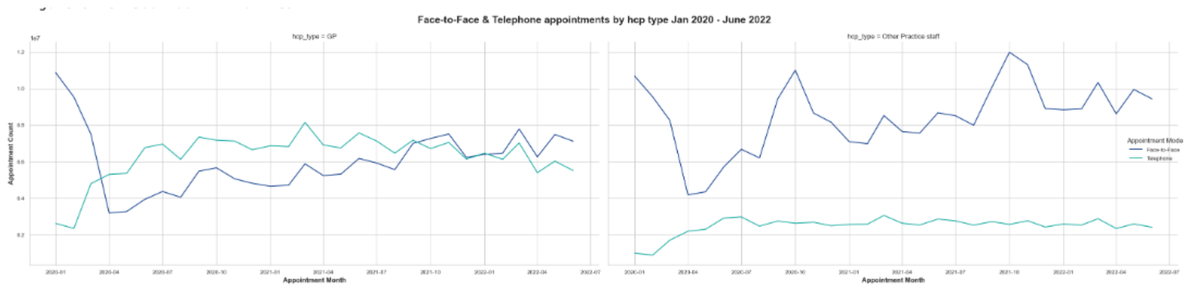
This section would focus on the top two hcp types together with the top two appointment modes. This will help us get additional insights on the relation between the top health care providers and the top appointment modes.

GP Chart insights

- The chart shows a sharp drop in Face-to-Face appointments during the pandemic.
- This was followed by a positive increase in Telephone appointments.
- Both appointment modes overlapped at the start of 2022, but then Face-to-Face took the lead.
- Face-to-Face appointments haven't regained their pre-pandemic levels but they're on a positive slope.

Other Practice Staff insights

- Face-to-Face appointments have also suffered drastically during the pandemic, but unlike GP appointments, they seem to be very close to regaining their pre-pandemic levels.
- Telephone appointments seem to maintain a more or less constant level that was also unaffected by the pandemic.



XIX. Twitter – Appendix

Analyse the top 30 hashtags from Twitter

- **Healthcare** is the top used hashtag

Bars are highlighted based on the theme of the Kws:

Health-related kws

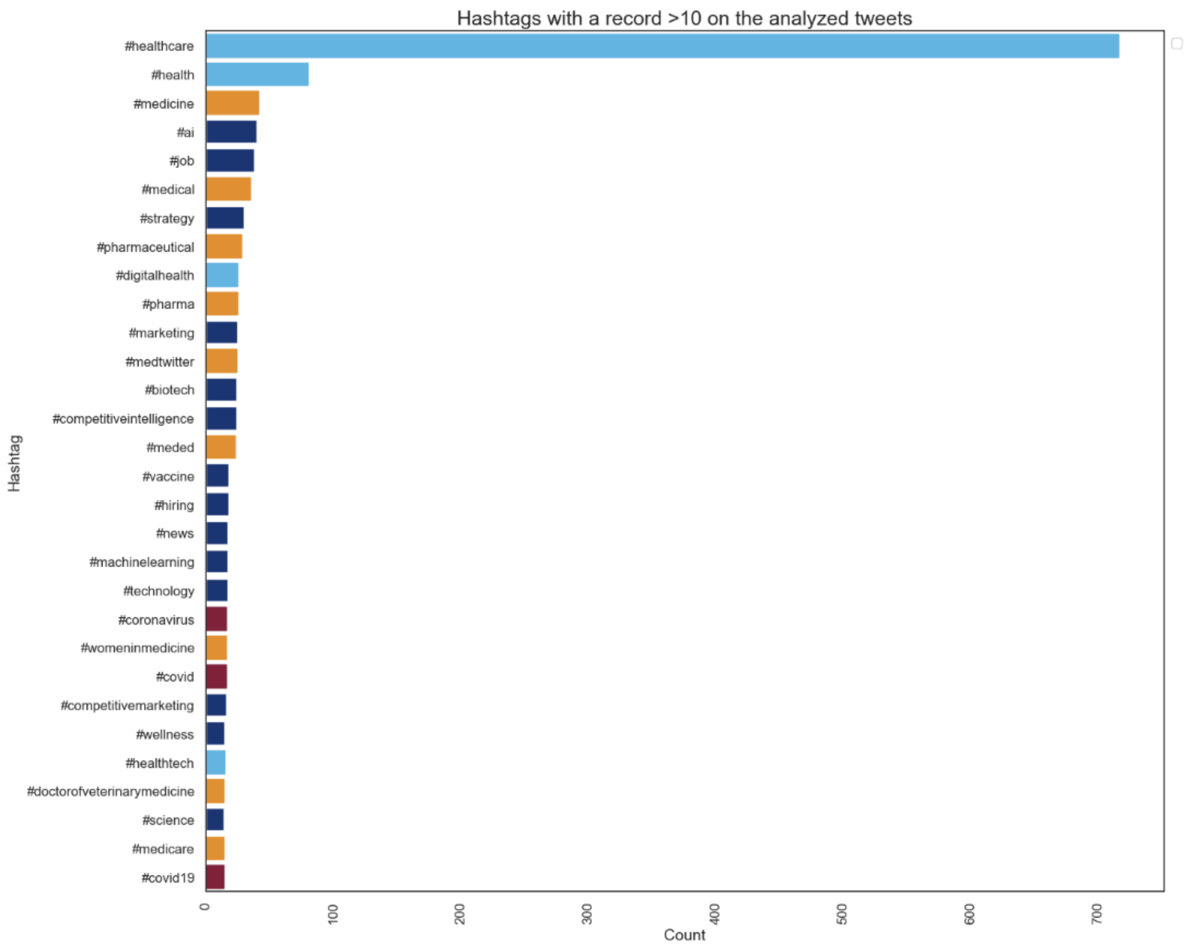
- Highlighting them shows that although they have the highest number they don't necessarily dominate the top 30 kws.

Covid-related kws

- Covid-related kws are amongst the lowest in the top 30. As we don't have the date of the tweets, this could be an indication that the tweets' sample was in late 2021 - 2022.

Medical/Pharma-related kws

- Medical/Pharma-related kws dominate the kws in the top 30 in terms of share.



Scarping the NHS weekly post on the tweets of the week including their hashtags

To get a sense of what Hashtags are normally used by the NHS, the following links for two weeks in 2022 were scraped:

- Week of 14 Jan 2022
- Week of 27 May 2022 These are not representative of all NHS hashtags. They are just sample hashtags for comparison's sake.

1]:
```python
# Scraping top NHS tweets for comparison's sake
# Specify the URL
URL = 'https://blog.horizonsnhs.com/post/102hg1l/tweets-of-the-week-14-january-2022'

# Create a variable
PAGE = requests.get(URL)

# View the HTML
soup = BeautifulSoup(PAGE.content,'html.parser')
#print(soup.prettify())
```

2]:
```python
hashtags_jan = []

for tag in soup.find_all('h2'):
    text = tag.get_text()
    if text.startswith('#'):  # simple check for hashtag
        hashtags_jan.append(text)

print('Hashtags promoted on the week of 14 January 2022: ',hashtags_jan)
```

```
Hashtags promoted on the week of 14 January 2022:  ['#Caring4NHSPeople', '#VirtualCollaborate', '#SolvingTogether']
```

5]:
```python
# Scraping top NHS tweets for comparison's sake
# Specify the URL
url = 'https://blog.horizonsnhs.com/post/102hp6x/tweets-of-the-week-27th-may-2022'

# Create a variable
page = requests.get(url)

# View the HTML
soup = BeautifulSoup(page.content,'html.parser')
#print(soup.prettify())
```

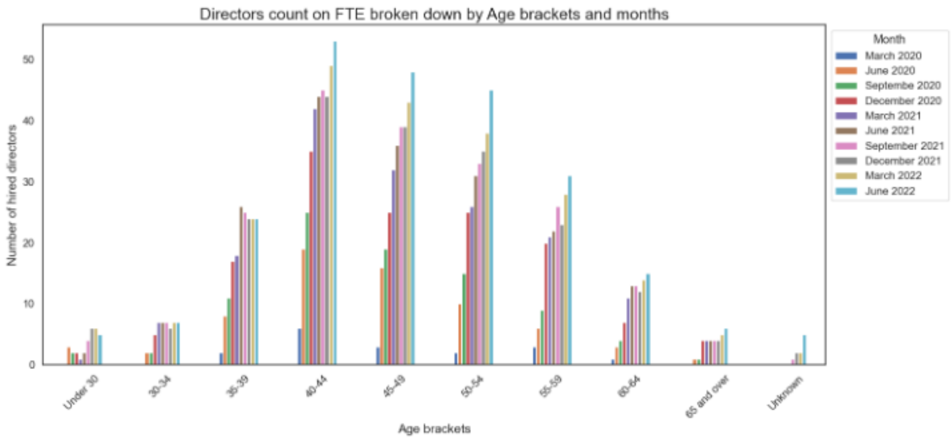
5]:
```python
hashtags_may = []

for tag in soup.find_all('h2'):
    text = tag.get_text()
    if text.startswith('#'):  # simple check for hashtag
        hashtags_may.append(text)

print('Hashtags promoted on the week of 27 May 2022: ',hashtags_may)
```

```
Hashtags promoted on the week of 27 May 2022:  ['#SolvingTogether', '#OurNHSPeople', '#StayandThrive', '#SolvingTogether']
```

XX. Staff

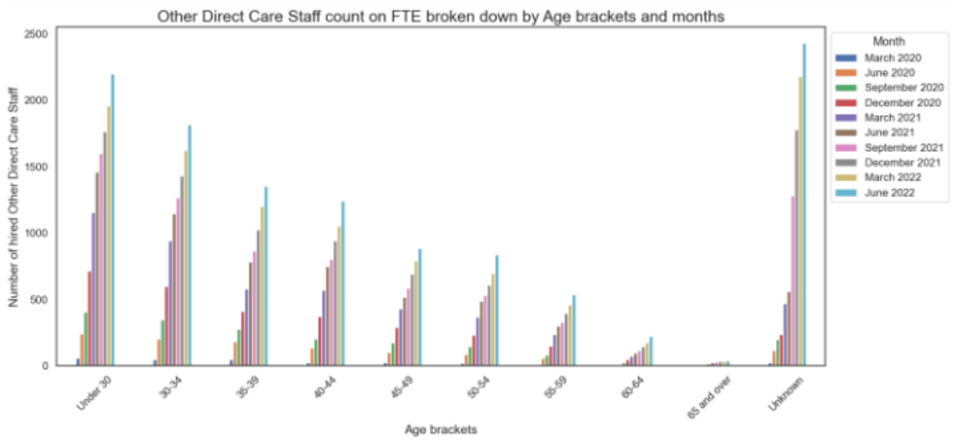
Directors hired in March and June 2022

- In March and June 2022 More directors were hired compared to other months included in the analysis
- They mostly are 40+ in Age



Other direct care staff

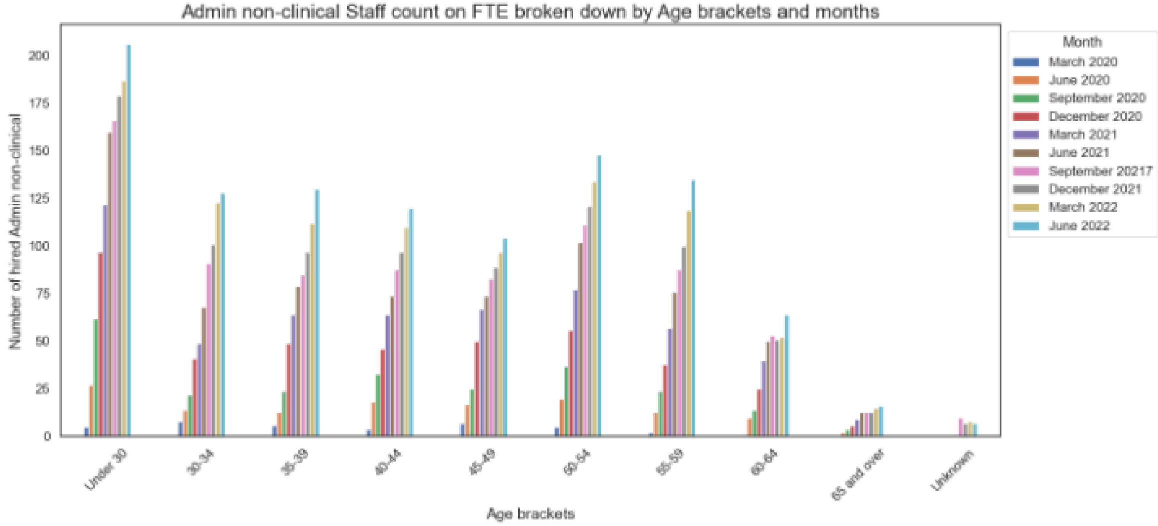
- The chart shows an increase in other direct care staff across all age brackets.
- The hiring for other direct care staff shows a steady growth across months and age groups.
- The hires categorized under Unknown suggest an issue in the data collection quality.



-

Admin Non-Clinical Staff

The chart shows an increase in Admin non-clinical staff across all age brackets. The hiring for Admin non-clinical staff shows a steady growth across months and age groups.



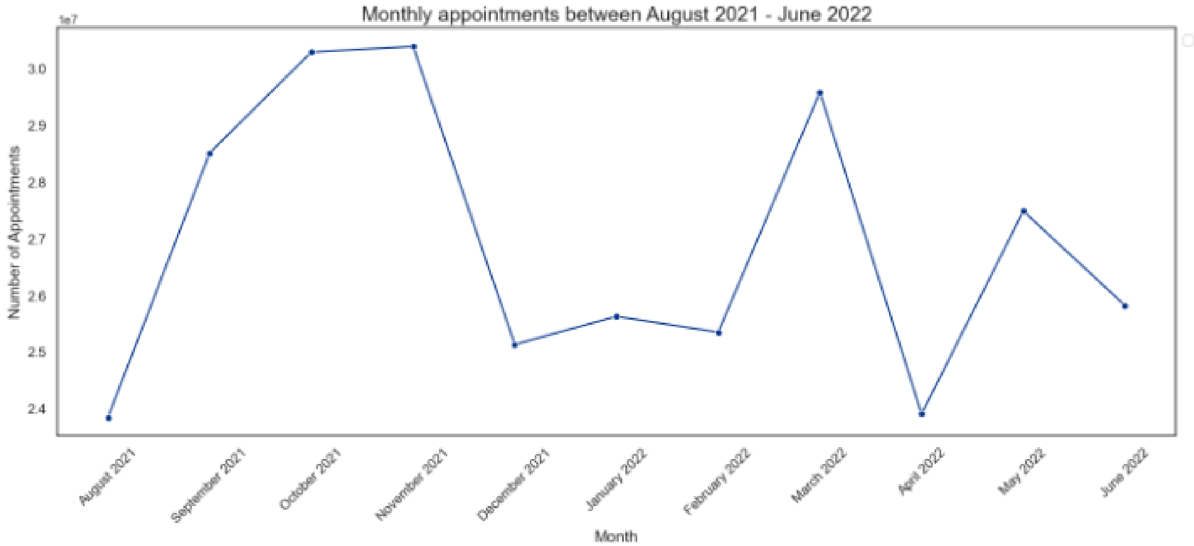
XXI. NHS Recommendations

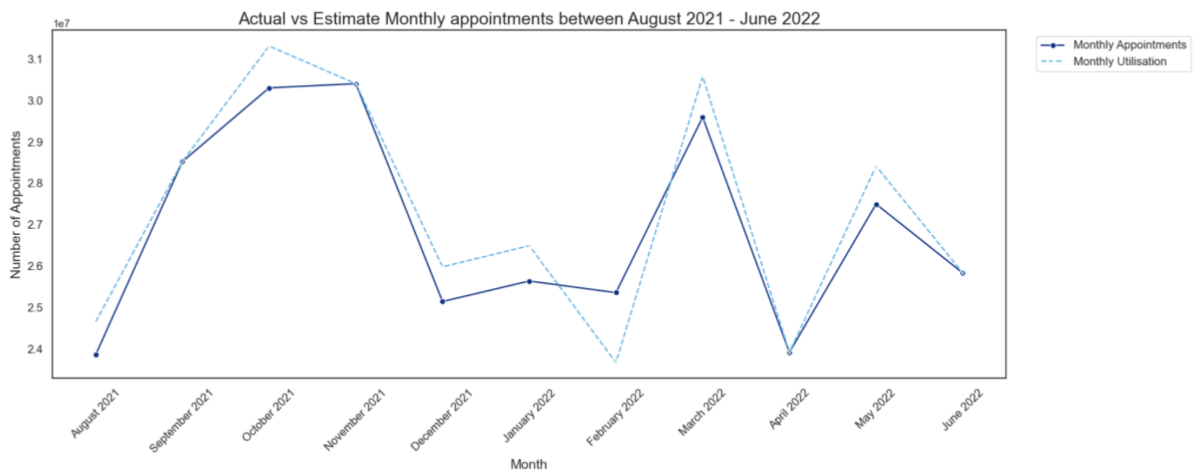
Appointment Utilization

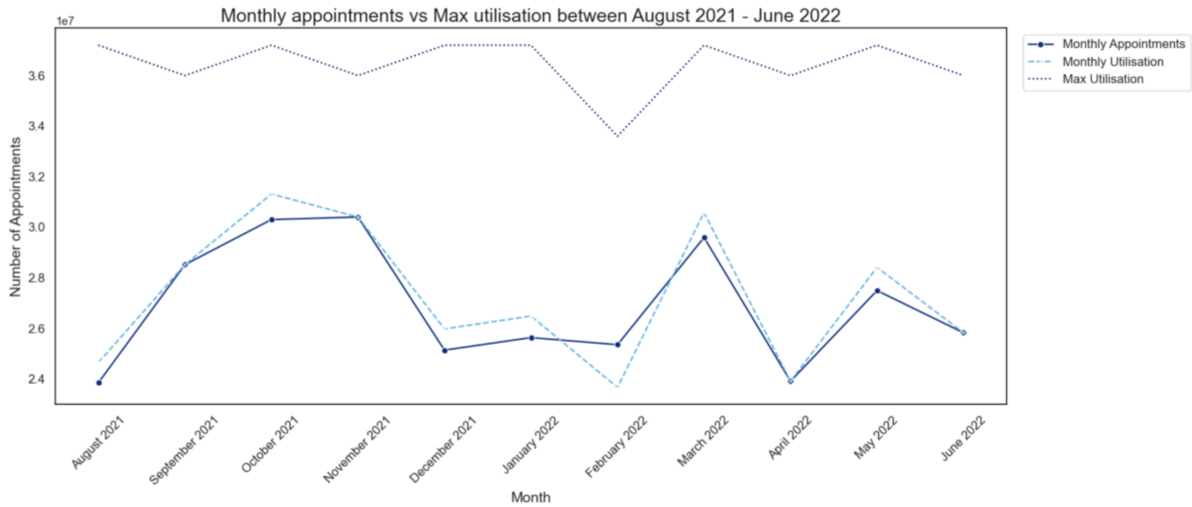
To get an estimate of the current utilization of the network a utilization calculation was included.

- The logic behind it is to divide the number of monthly appointments by 30 to get a daily utilization.
- To get the monthly utilization, the actual number of days in a month was used.
- The daily and monthly utilization can then be compared to the estimate provided of 1200000 daily appointments, to see where the network currently stands.

Insight Currently, the network is within the limit of permissible utilization as shown on the following graphs.







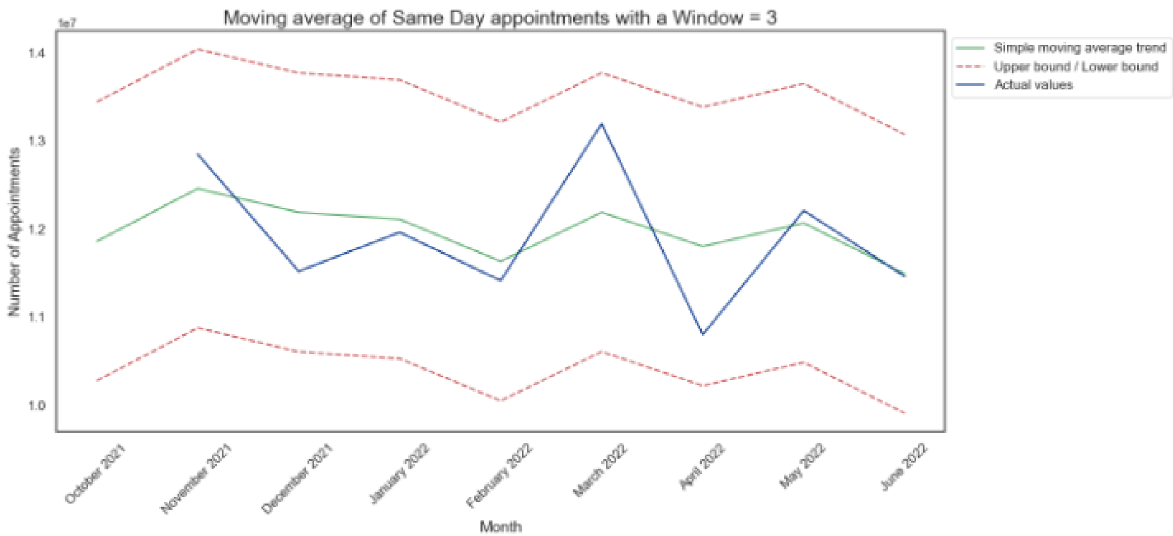
Time between booking and consultation
Boxplots for all DataFrames were checked to make sure there were not outliers.

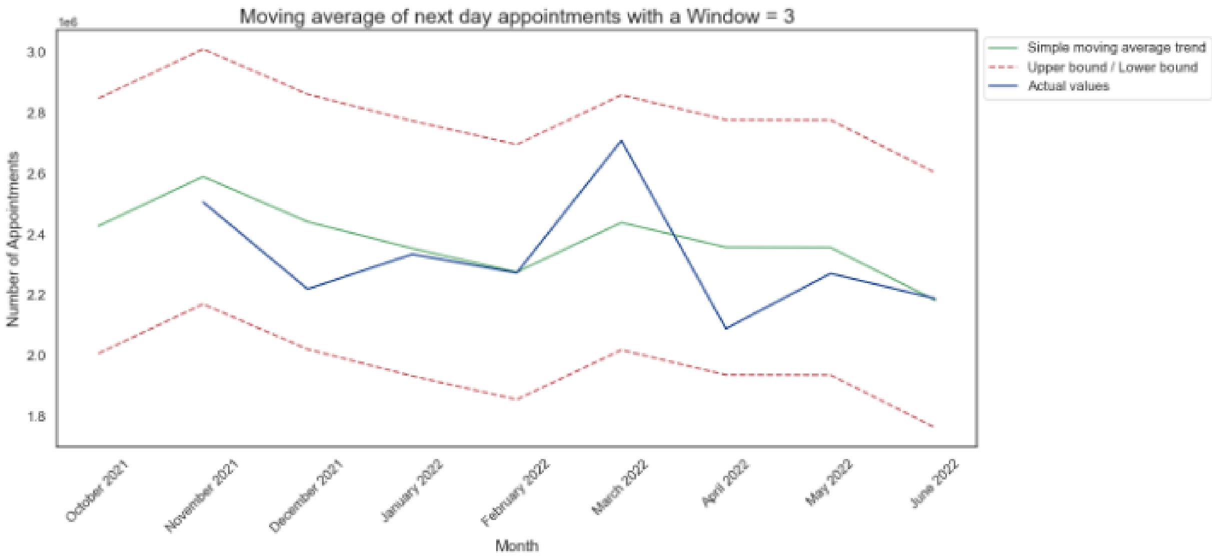
The avgtime DataFrame was subset in all the different waiting times to be able to applyt the moving average method.

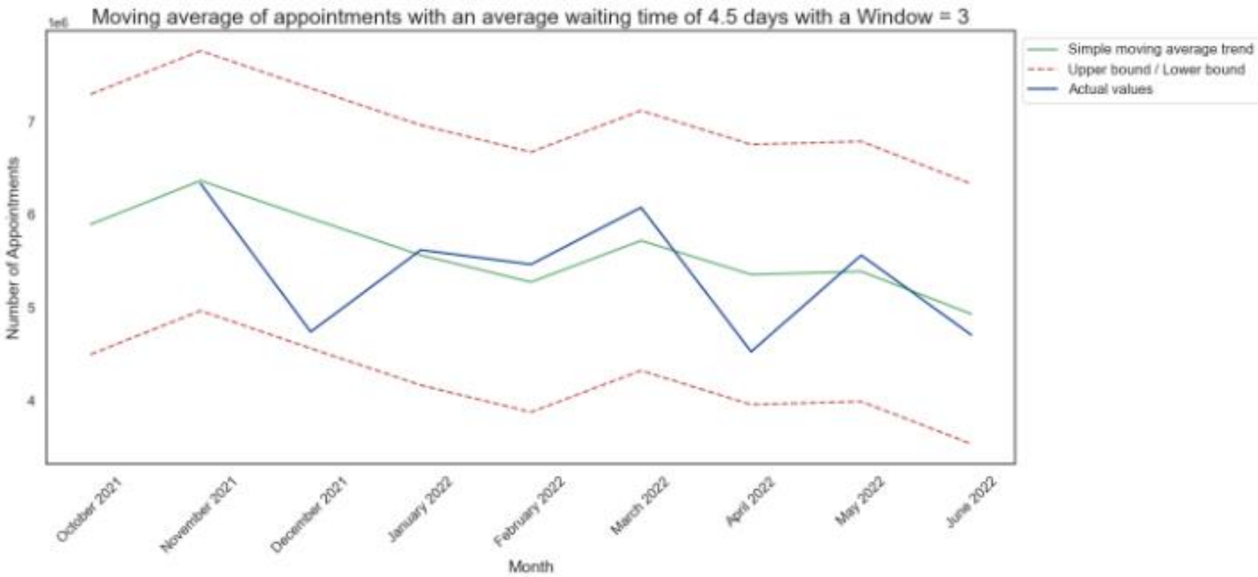
- It was found that appointments with a short waiting time and this includes same day, next day, average wait of 4.5 and 11 days although they currently have the highest share of appointments, have a negative slope. This indicates that these appointments are likely to get fewer appointments in the coming months.

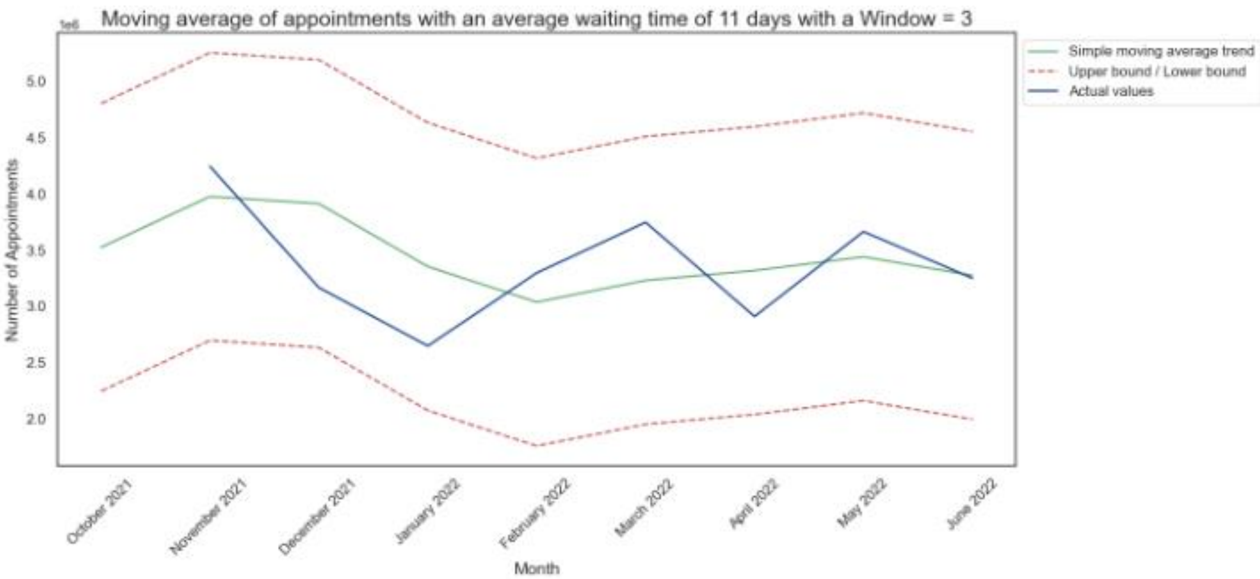
- For appointments with a longer waiting time, the curves show a positive slope which indicates that these appointments are likely to get more appointments in the coming months.

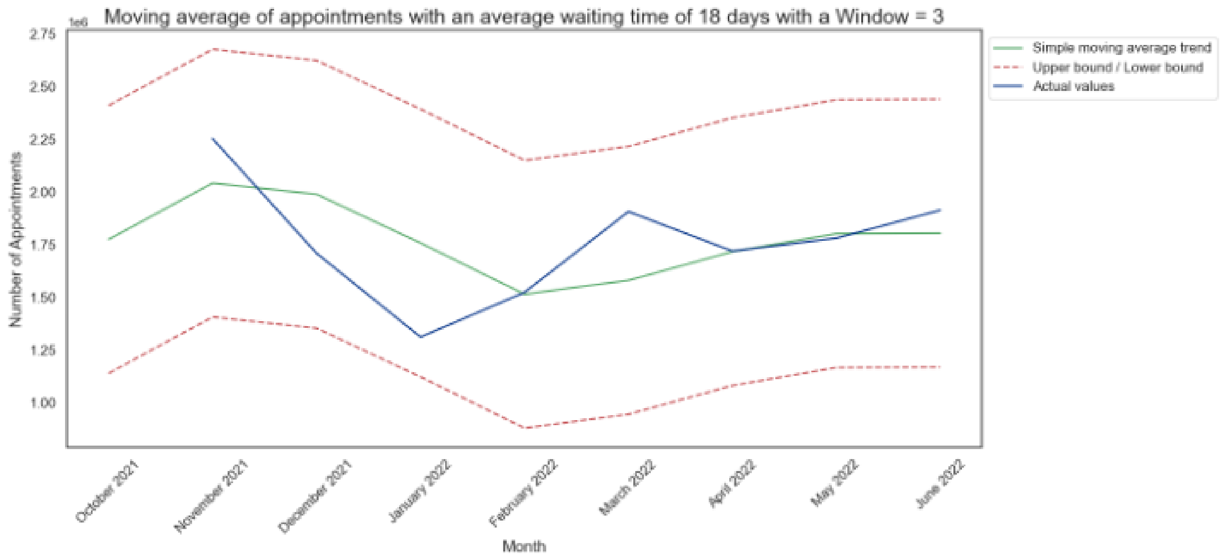
- This finding suggests longer waiting time in the comings months.



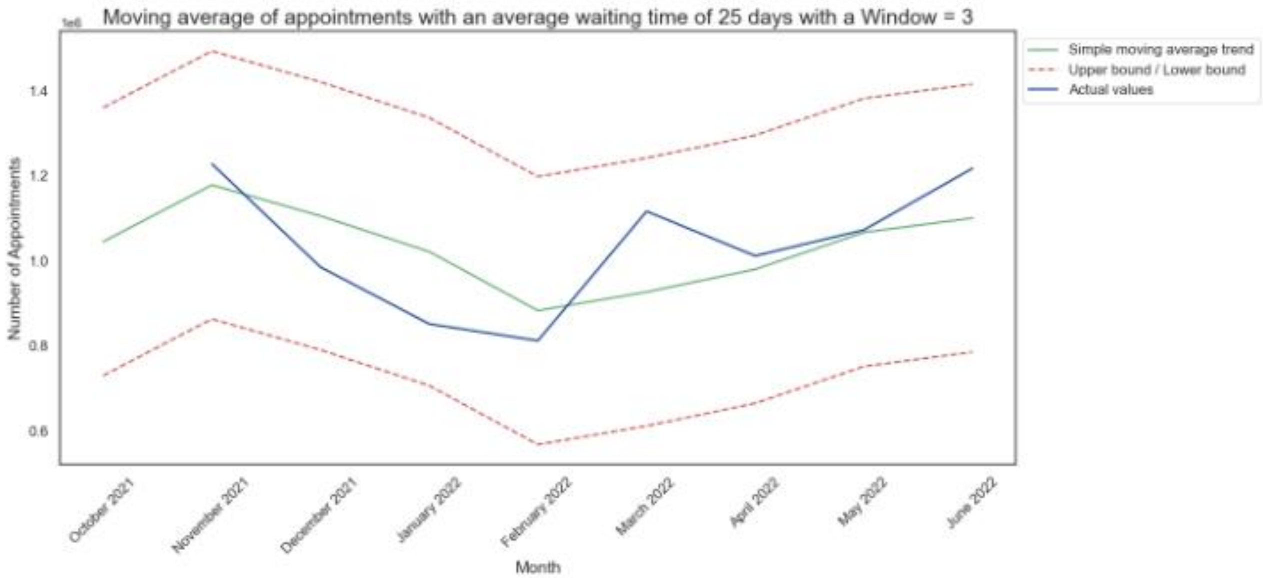

Moving average of next day appointments with a Window = 3
1e6
3.0
2.8
2.6
2.4
2.2
2.0
1.8
Number of Appointments
Simple moving average trend
Upper bound / Lower bound
Actual values
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month


Moving average of appointments with an average waiting time of 4.5 days with a Window = 3
1e6
7
6
5
4
Number of Appointments
Simple moving average trend
Upper bound / Lower bound
Actual values
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month

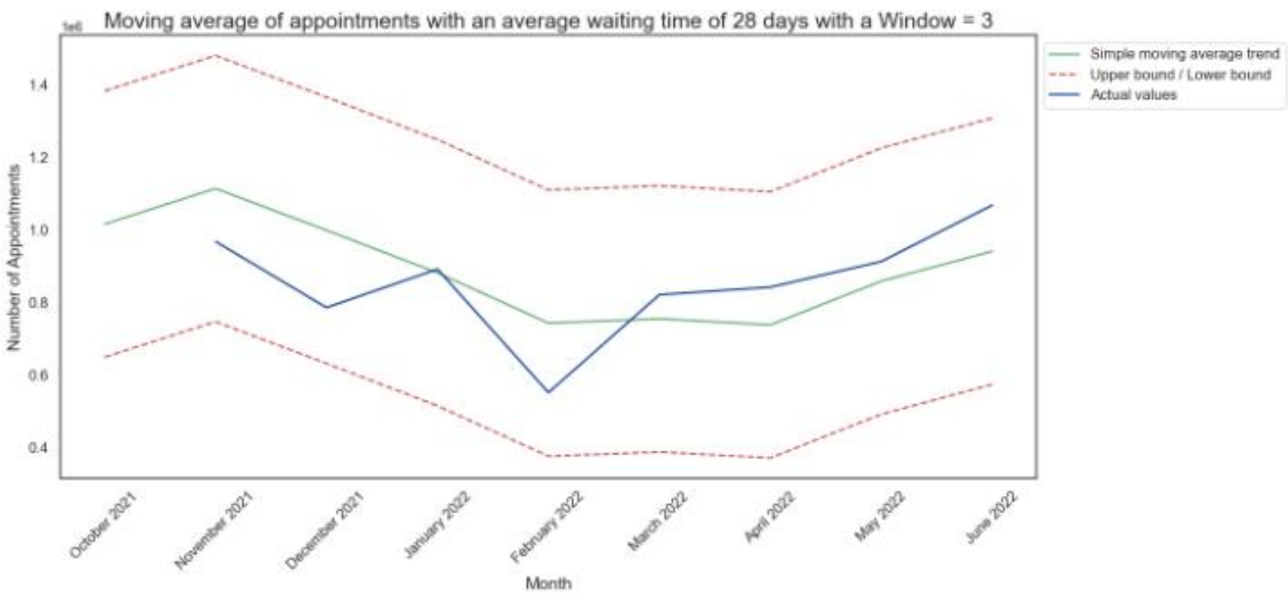

Moving average of appointments with an average waiting time of 11 days with a Window = 3
1e6
5.0
4.5
4.0
3.5
3.0
2.5
2.0
Number of Appointments
Simple moving average trend
Upper bound / Lower bound
Actual values
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month



Moving average of appointments with an average waiting time of 18 days with a Window = 3
1e6
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
Number of Appointments
Simple moving average trend
Upper bound / Lower bound
Actual values
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month



Moving average of appointments with an average waiting time of 25 days with a Window = 3
1e6
1.4
1.2
1.0
0.8
0.6
Number of Appointments
Simple moving average trend
Upper bound / Lower bound
Actual values
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month



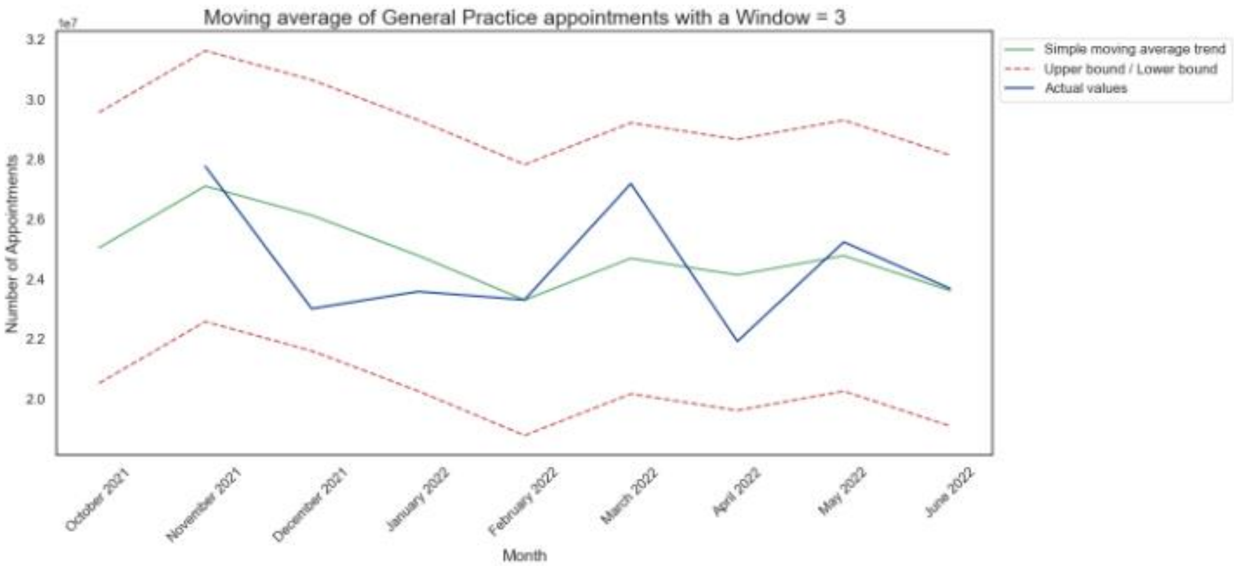
Moving average of appointments with an average waiting time of 28 days with a Window = 3
1e6
1.4
1.2
1.0
0.8
0.6
0.4
Number of Appointments
Simple moving average trend
Upper bound / Lower bound
Actual values
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month

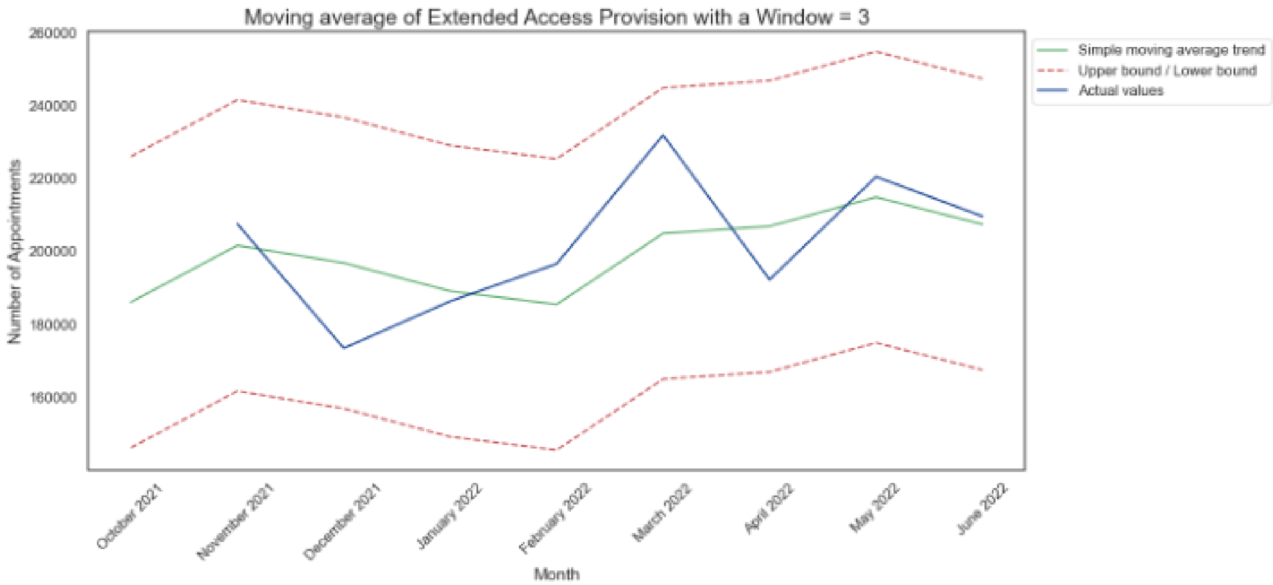
Service Settings

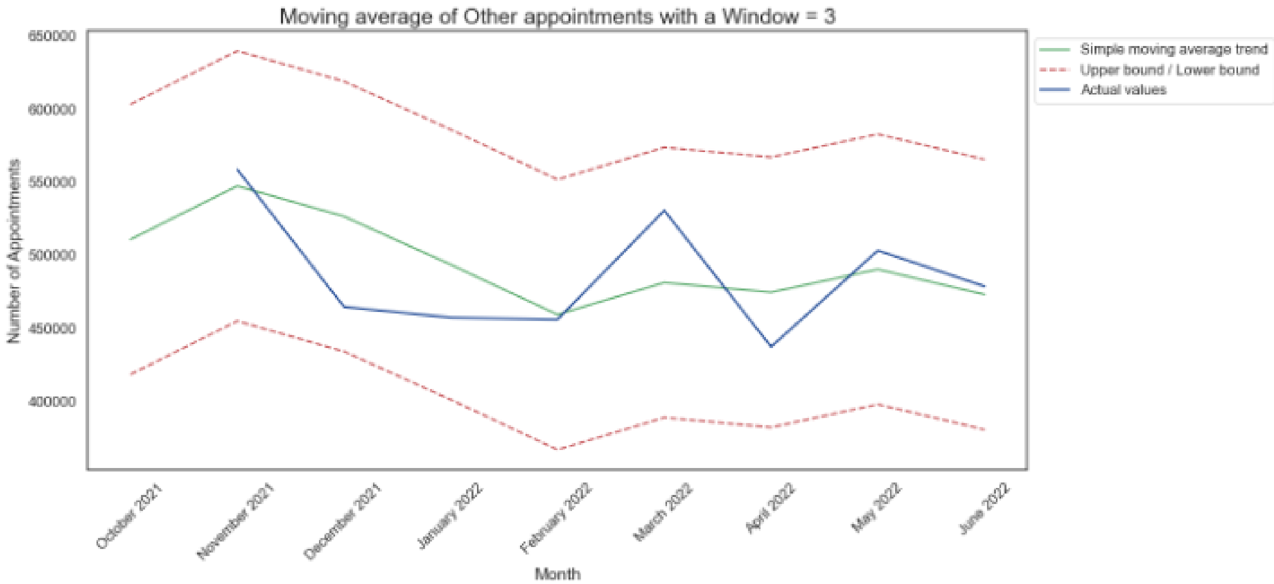
Looking at apointments broken down by their service settings shows the following:

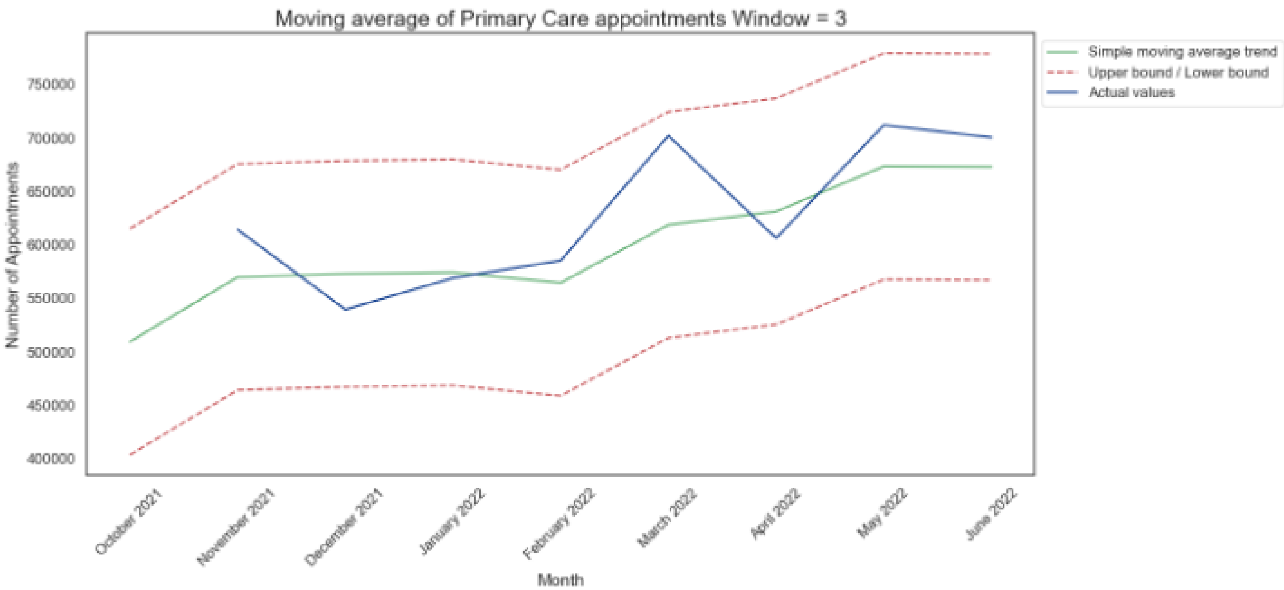
- General practice appointment stabilized after the sudden increase in November 2021.
- Extended Access Provision and Primary care appointments have a positive slope suggesting an increase over the coming months. This is confirmed to be part of the NHS's plan to flexibly cater to the local needs by offering appointments outside of the regular working hours
- Other appointments follow the same pattern as for the General Practice.

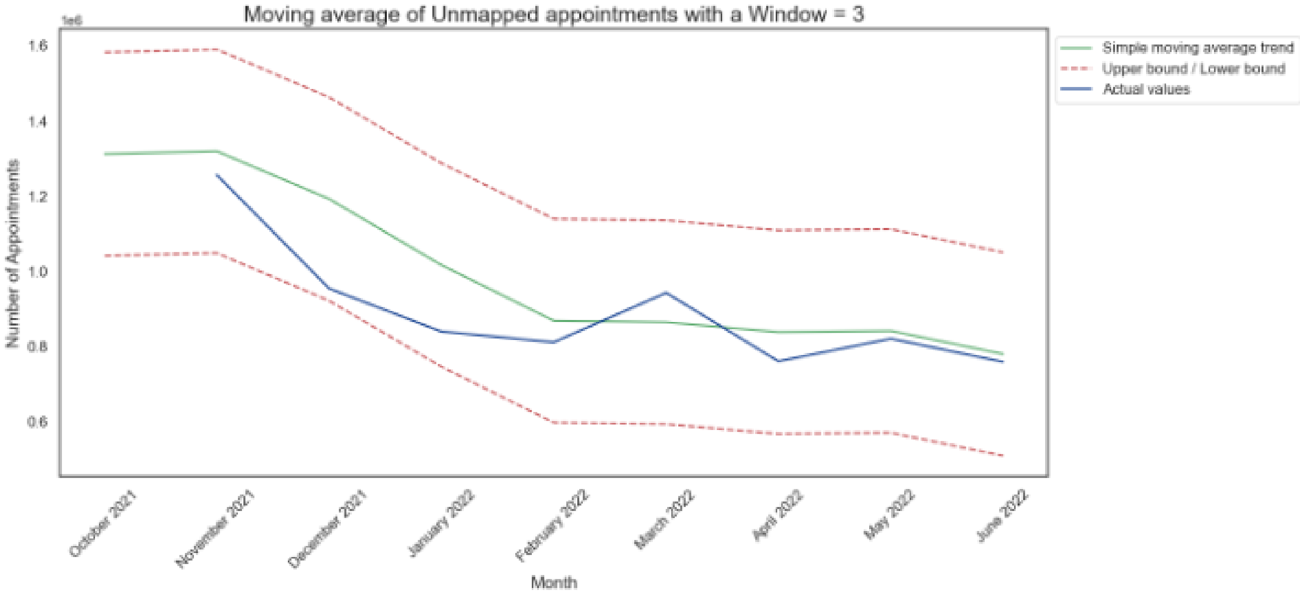
- Primary care appointments have a positive slope suggesting an increase over the coming months.
- Unmapped show a negative slope which is consistent with other finding and it suggests an improve in data quality.




Moving average of Extended Access Provision with a Window = 3
Simple moving average trend
Upper bound / Lower bound
Actual values
Number of Appointments
260000
240000
220000
200000
180000
160000
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month


Moving average of Other appointments with a Window = 3
Simple moving average trend
Upper bound / Lower bound
Actual values
Number of Appointments
650000
600000
550000
500000
450000
400000
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month


Moving average of Primary Care appointments Window = 3
Simple moving average trend
Upper bound / Lower bound
Actual values
Number of Appointments
750000
700000
650000
600000
550000
500000
450000
400000
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month



Moving average of Unmapped appointments with a Window = 3
1e6
1.6
1.4
1.2
1.0
0.8
0.6
Number of Appointments
October 2021
November 2021
December 2021
January 2022
February 2022
March 2022
April 2022
May 2022
June 2022
Month
Simple moving average trend
Upper bound / Lower bound
Actual values

Works Cited

1. The health foundation. [Online] https://www.health.org.uk/features-and-opinion/features/how-has-the-covid-19-pandemic-impacted-primary-care.
2. National population projections table of contents. *Office for national statistics.* [Online] https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationprojections/datasets/2014basednationalpopulationprojectionstableofcontents.
3. Primary Care Network Workforce, 30 June 2022. *NHS.* [Online] https://digital.nhs.uk/data-and-information/publications/statistical/primary-care-network-workforce/30-june-2022.
4. data.gov.uk. *Open Government Data.* [Online] 2023. https://www.data.gov.uk/dataset/b66d098a-71ab-499c-9f62-1ef74f60f8bf/sub-icb-locations-to-integrated-care-boards-to-nhser-april-2023-lookup-in-en.
5. Covid-19 Vaccinations Statistics. *NHS.* [Online] https://www.england.nhs.uk/statistics/wp-content/uploads/sites/2/2021/10/COVID-19-weekly-announced-vaccinations-21-October-2021.pdf.
6. Enhanced access to General Practice services through the network contract DES . *NHS.* [Online] https://www.england.nhs.uk/gp/investment/gp-contract/network-contract-directed-enhanced-service-des/enhanced-access-faqs/.
7. The NHS workforce in numbers. *Nuffield Trust.* [Online] https://www.nuffieldtrust.org.uk/resource/the-nhs-workforce-in-numbers.
8. Data saves lives: reshaping health and social care with data. *Gov.* [Online] https://www.gov.uk/government/publications/data-saves-lives-reshaping-health-and-social-care-with-data/data-saves-lives-reshaping-health-and-social-care-with-data.

Template Source:

9. NHS: NHS England report template OAHC Content [Online]:https://www.england.nhs.uk/wp-content/uploads/2013/11/NHS-England-report-template-OAHC-Acute-including-AE.docx