Assignment: NHS Diagnostic Analysis using Python

Background & Scenario

NHS capacity is debated among stakeholders. Investigate reasons for missed appointments as they lead to significant costs for the NHS. Key questions:

1. **What is the actual utilisation of resources?**
2. **Has there been adequate staff and capacity in the networks?**
3. **What are the main reasons for missed appointments?**
4. **What is the public opinion on NHS on capacity? Twitter-based analysis**

Datasets Exploration

Available datasets

Available Datasets

1. actual_duration.csv – Details of appointments made by patients. For example, the regional information, date, duration, and number of appointments pertaining to a certain class.
2. appointments_regional.csv – Details on the type of appointments made by patients. For example, regional information, the month of appointment, appointment status, healthcare professional, appointment mode, the time between booking and the appointment, as well as the number of appointments pertaining to a certain class.
3. national_categories.xlsx – Details of the national categories of appointments made by patients. For example, the regional information, date of appointment, service setting, type of context, national category, and the number of appointments pertaining to a certain class.
4. tweets.csv – Data related to healthcare in the UK scraped from Twitter.
5. metadata_nhs.txt – Details of the data set, data quality, and reference.

Reading and Mapping Variables across the datasets

As there are several files, the 1st step was importing all libraries and files

Steps

1. Import all necessary libraries.
2. Import the files using the read functions for each data type (excel, csv, etc)
3. Visualise what is in the datasets using. column; .head(), .shape ; .dtypes.
    a. Map unique values in the series below, the functions used were .nunique () and, then, groupby.
        i. Number of unique locations: **106**

Question 1: How many locations are there in the data set?

A1. Number of unique locations: 106

```
num_locations = ad['sub_icb_location_name'].nunique()
print("Number of unique locations:", num_locations)
```

```
Number of unique locations: 106
```

```
nc.groupby('sub_icb_location_name')[['count_of_appointments']].sum().\
sort_values('count_of_appointments', ascending=False).round(1)
```

sub_icb_location_name	count_of_appointments
NHS North West London ICB - W2U3Z	12142390
NHS North East London ICB - A3A8R	9588891
NHS Kent and Medway ICB - 91Q	9286167
NHS Hampshire and Isle Of Wight ICB - D9Y0V	8288102
NHS South East London ICB - 72Q	7850170
...	...
NHS Cheshire and Merseyside ICB - 01V	641149
NHS Nottingham and Nottinghamshire ICB - 02Q	639660
NHS Greater Manchester ICB - 00V	639211
NHS Cheshire and Merseyside ICB - 01T	606606
NHS Lancashire and South Cumbria ICB - 02G	554694

b. Which are the five locations with the highest number of **records?**

groupby function using the **count** estimator.

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

groupby function using the **sum** estimator.

Q. 2A) What are the five locations with the highest number of records?

These are the 5 locations with the highest number of records

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

In [28]:
```
# Determine the top five locations based on COUNT.
nc.groupby('sub_icb_location_name')[['icb_ons_code']].count().sort_values('icb_ons_code', ascending=False).round(1)
```

Out[28]:

sub_icb_location_name	icb_ons_code
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
...	...
NHS North East and North Cumbria ICB - 00N	4210
NHS Lancashire and South Cumbria ICB - 02G	4169
NHS Cheshire and Merseyside ICB - 01V	3496
NHS Cheshire and Merseyside ICB - 01T	3242

Q. 2B) What are the five locations with the highest number of APPOINTMENTS?

These are the 5 locations with the highest number of appointments NHS North West London ICB - W2U3Z 12142390
NHS North East London ICB - A3A8R 9588891
NHS Kent and Medway ICB - 91Q 9286167
NHS Hampshire and Isle Of Wight ICB - D9Y0V 8288102
NHS South East London ICB - 72Q 7850170\

```python
# Determine the top five locations based on SUM of appointments

nc.groupby('sub_icb_location_name')[['count_of_appointments']].sum().sort_values('count_of_appointments', ascending=False)\
.round(1)
```

sub_icb_location_name	count_of_appointments
NHS North West London ICB - W2U3Z	12142390
NHS North East London ICB - A3A8R	9588891
NHS Kent and Medway ICB - 91Q	9286167
NHS Hampshire and Isle Of Wight ICB - D9Y0V	8288102
NHS South East London ICB - 72Q	7850170
...	...
NHS Cheshire and Merseyside ICB - 01V	641149
NHS Nottingham and Nottinghamshire ICB - 02Q	639660
NHS Greater Manchester ICB - 00V	639211
NHS Cheshire and Merseyside ICB - 01T	606606
NHS Lancashire and South Cumbria ICB - 02G	554694

106 rows × 1 columns

Number of context_type: 3

```python
nc.groupby('context_type')[['count_of_appointments']].count().sort_values('count_of_appointments', ascending=False).round(1)
```

context_type	count_of_appointments
Care Related Encounter	700481
Inconsistent Mapping	89494
Unmapped	27419

Number of appointment_status: 3

```python
ar.groupby('appointment_status')[['count_of_appointments']].sum()\
.sort_values('count_of_appointments', ascending=False).round(1)
```

appointment_status	count_of_appointments
Attended	677755876
Unknown	34137416
DNA	30911233

Number of national_category: **18**

```
nc.groupby('national_category')[['count_of_appointments']].sum()\
.sort_values('count_of_appointments', ascending=False).round(1)
```

national_category	count_of_appointments
General Consultation Routine	97271522
General Consultation Acute	53691150
Clinical Triage	41546964
Planned Clinics	28019748
Inconsistent Mapping	27890802
Planned Clinical Procedure	25702694
Unmapped	11080810
Unplanned Clinical Activity	3055794
Home Visit	2144452
Structured Medication Review	1858379

Number of service_settings: **5**

```
nc.groupby('service_setting')[['count_of_appointments']].sum().\
sort_values('count_of_appointments', ascending=False).round(1)
```

service_setting	count_of_appointments
General Practice	270811691
Unmapped	11080810
Primary Care Network	6557386
Other	5420076
Extended Access Provision	2176807

4. Check for missing values using the. isna() function >>>> number of missing values using .shape

```
# Check for missing values
# Create a new DataFrame and use the isna() function to find missing values.
ad_na = ad[ad.isna().any(axis=1)]

# View the shape of the DataFrame.
ad_na.shape

#out=(0, 8)

#the output says there is ZERO rows with NA values across 8 columns (0,8)
```

General Observations

1) The **only** numeric variable in the datasets is **count_of_appointment**.
2) **Absence of** variables on staff numbers or the number of patients to calculate **capacity directly.**
3) **Absence of** variables related to **NHS costs**

Business Questions

After exploring and visualising the datasets, I broke down the core questions in sub-questions.

What was the actual utilisation of resources?

WHERE: Which five locations have the highest number of appointments?

sub_icb_location_name	count_of_appointments	
NHS North West London ICB - W2U3Z	12142390	HIGHEST # OF APP
NHS North East London ICB - A3A8R	9588891	
NHS Kent and Medway ICB - 91Q	9286167	
NHS Hampshire and Isle Of Wight ICB - D9Y0V	8288102	
NHS South East London ICB - 72Q	7850170	
...	...	
NHS Cheshire and Merseyside ICB - 01V	641149	LOWEST # OF APP
NHS Nottingham and Nottinghamshire ICB - 02Q	639660	
NHS Greater Manchester ICB - 00V	639211	
NHS Cheshire and Merseyside ICB - 01T	606606	
NHS Lancashire and South Cumbria ICB - 02G	55469	

```
## Q. 2B) What are the five locations with the highest number of APPOINTMENTS?
**These are the 5 locations with the highest number of appointments**
NHS North West London ICB - W2U3Z 12142390 \
NHS North East London ICB - A3A8R 9588891 \
NHS Kent and Medway ICB - 91Q 9286167 \
NHS Hampshire and Isle Of Wight ICB - D9Y0V 8288102 \
NHS South East London ICB - 72Q 7850170\
```

```python
# Determine the top five locations based on SUM of appointments
nc.groupby('sub_icb_location_name')[['count_of_appointments']].sum().sort_values('count_of_appointments', ascending=False)\
.round(1)
```

sub_icb_location_name	count_of_appointments
NHS North West London ICB - W2U3Z	12142390
NHS North East London ICB - A3A8R	9588891
NHS Kent and Medway ICB - 91Q	9286167
NHS Hampshire and Isle Of Wight ICB - D9Y0V	8288102
NHS South East London ICB - 72Q	7850170
...	...
NHS Cheshire and Merseyside ICB - 01V	641149
NHS Nottingham and Nottinghamshire ICB - 02Q	639660
NHS Greater Manchester ICB - 00V	639211
NHS Cheshire and Merseyside ICB - 01T	606606
NHS Lancashire and South Cumbria ICB - 02G	554694

106 rows × 1 columns

WHEN: Which months had the highest number of appointments?

I used the groupby function, grouping the sum of ‘count of appointments’ by ‘month_year’

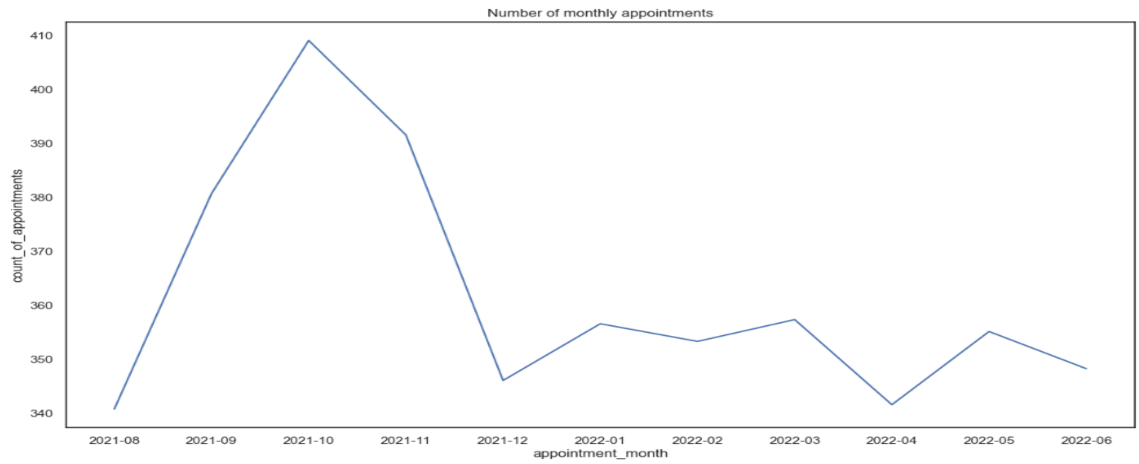
In [116]:
```
nc.groupby('month_year')[['count_of_appointments']].sum().sort_values('count_of_appointments', ascending=False).round(1)
```
Out[116]:

month_year	count_of_appointments
11/21	30405070
10/21	30303834
03/22	29595038
09/21	28522501
05/22	27495508
06/22	25828078
01/22	25635474
02/22	25355260
12/21	25140776
04/22	23913060
08/21	23852171

Oct,Nov_2021 and Mar_2022 registered the peak of appointments. As a context, it coincides with the Omicron Surge (Late_21- Early_22) but also with pre-holidays periods

a. Visualise the number of appointments by month

i. December and April show a fall in appointments, coinciding with the period of Christmas and Easter. Hypothesis = people go less to appointments in holiday periods.



b. 3 Visualisations with monthly appointments by service settings, context types, and national categories

i. For that, I created line plots following the logic below:

1. x=' appointment_month'; y='count_of_appointments', hue = **each of the variables**

ii. **Service Settings**: GP are the predominant service type provided by NHS, followed by unmapped

iii. **Context Type:** since 2022, 'Care Related Encounter' surpassed the number of 'Unmapped'

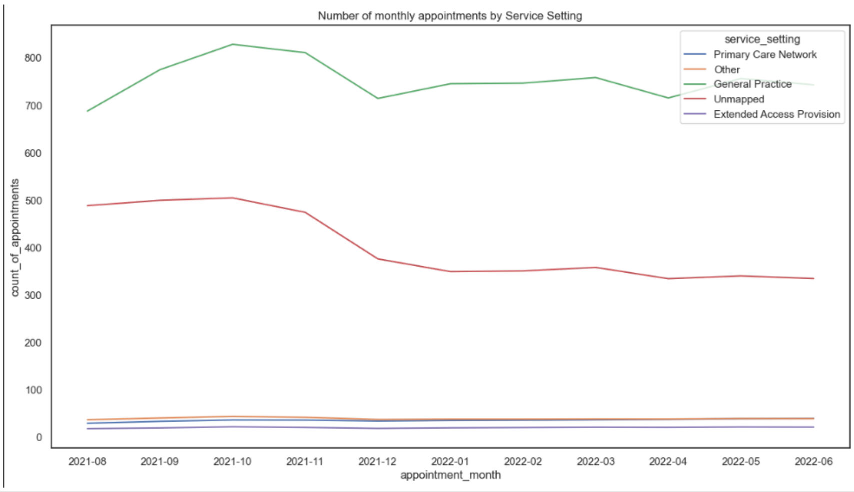
iv. National categories:

1. Due to the wide variety of national categories, I created a new nc dataset , only with the top 5 national category types (ranked by count_of_appointments)

Service settings:

[52]:
```python
# Plot the appointments over the available date range, and review the service settings for months.
# Create a lineplot.
plt.title("Number of monthly appointments by Service Setting")
sns.lineplot(x='appointment_month', y='count_of_appointments', hue='service_setting', data=nc, ci=None)
```

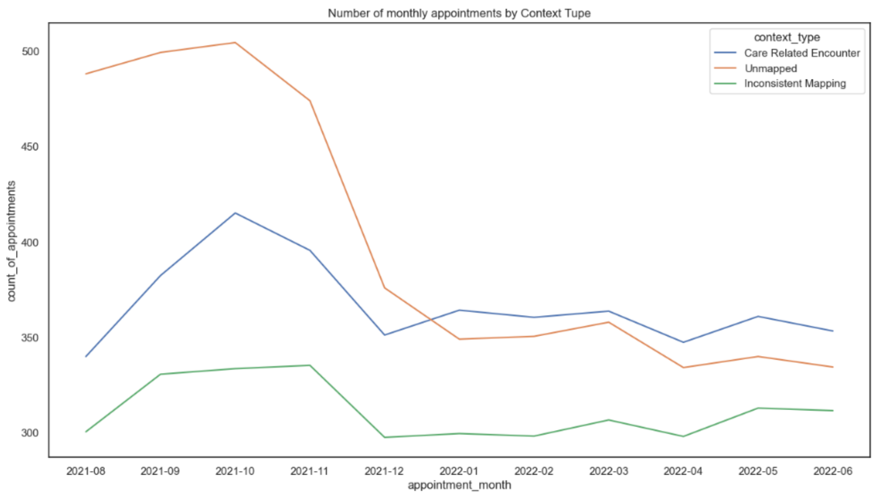
[52]: `<Axes: title={'center': 'Number of monthly appointments by Service Setting'}, xlabel='appointment_month', ylabel='count_of_appointments'>`



Context types

In [117]:
```python
plt.title("Number of monthly appointments by Context Tupe")

sns.lineplot(x='appointment_month', y='count_of_appointments', hue='context_type', data=nc,ci=None)
```

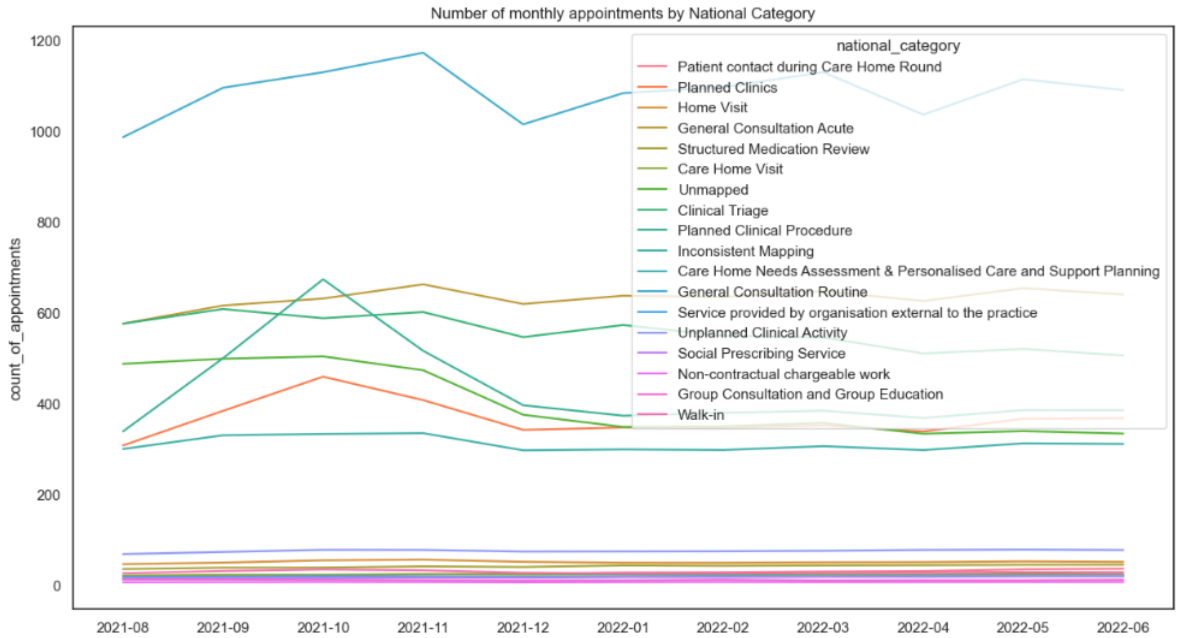
Out[117]: `<Axes: title={'center': 'Number of monthly appointments by Context Tupe'}, xlabel='appointment_month', ylabel='count_of_appointments'>`



National categories:

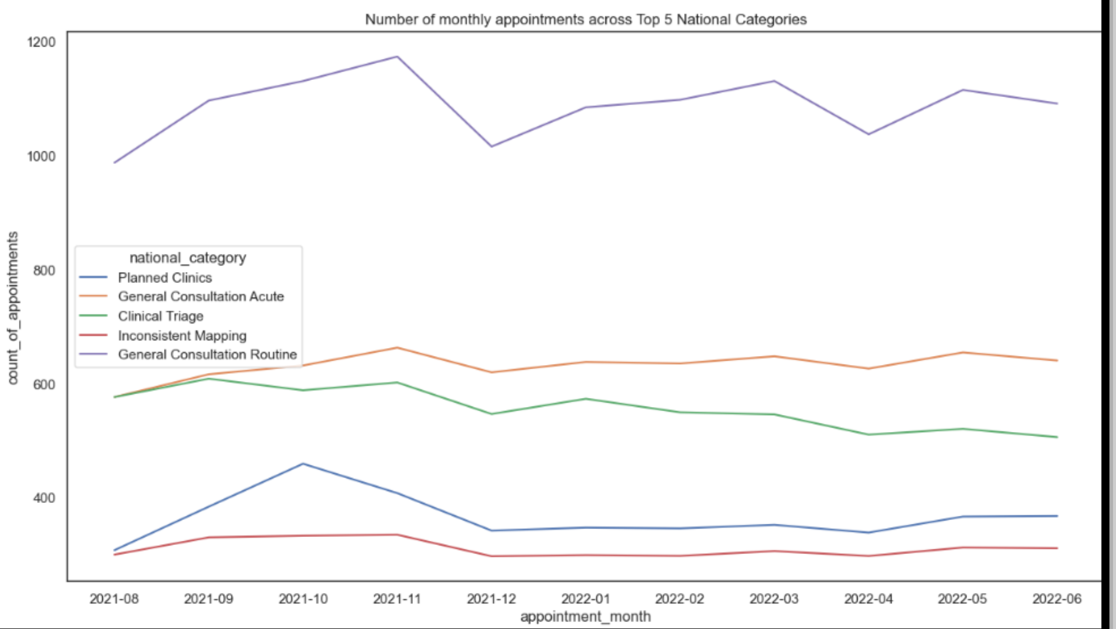
```
plt.figure(figsize=(15, 8))
plt.title("Number of monthly appointments by National Category")
sns.lineplot(x='appointment_month', y='count_of_appointments', hue='national_category', data=nc, ci= None)
```

```
<Axes: title={'center': 'Number of monthly appointments by National Category'}, xlabel='appointment_month', ylabel='count_of_appointments'>
```



In [126]:
```
plt.title("Number of monthly appointments across Top 5 National Categories")
sns.lineplot(x='appointment_month', y='count_of_appointments', \
             hue='national_category', ci=None, data=nc_filtered_top5national_category)
```

Out[126]:
```
<Axes: title={'center': 'Number of monthly appointments across Top 5 National Categories'}, xlabel='appointment_month', ylabel='count_of_appointments'>
```



Q1) Should the NHS start looking at increasing staff levels?

a. Determine the number of appointments by month using the groupby function and, then create a new dataset

```
#determine the number of appointments by month
ar2.groupby('appointment_month')[['count_of_appointments']].sum().sort_values('count_of_appointments', ascending=False).round(1)
```

b.

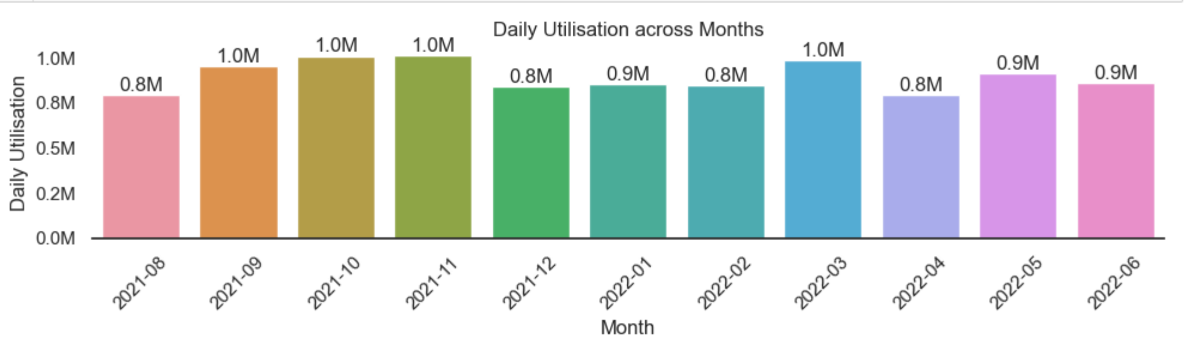
c. Add a new column ['utilisation'], by dividing 'count_of_appointment' per 30

```
# Determine the total number of appointments per month.
ar3 = ar2[['appointment_month', 'count_of_appointments']].groupby(['appointment_month']).sum().sort_values('appointment_month').r

# Add a new column to indicate the average utilisation of services.
# Monthly aggregate / 30 to get to a daily value.
ar3['utilisation'] = round(ar3['count_of_appointments'] / 30, 1)

# View the DataFrame.
ar3
```

d.

e. The NHS has provided an average of 1.2M daily appointments as a guideline for maximum capacity.

f. Visualise the monthly evolution of daily appointments to check if there is any period that exceeds the maximum capacity




```
#table with similar info
ar3.groupby('appointment_month')[['utilisation']].sum().sort_values('utilisation', ascending=False).round(1)
```

The NHS has provided a figure of an average of 1.2M daily appointments for maximum capacity. As you can see in the graph above, every month is within the NHS's maximum capacity: 1.2M daily appointments. It would be prudent to rotate staff so that peak months (Sep, Oct,Nov) have a larger staff, while less busy months (December) have a smaller staff. Concentrating the vacation period of NHS employees in December could be a good strategy, adding some benefits to whom take vacation during this period. However, watching out to maintain ideal capacity in December as well.

g.

Conclusion: All months are within the NHS's capacity: 1.2M . It would be prudent to rotate staff so that peak months (Sep, Oct, Nov) have a more extensive staff, while less busy months (December) have a smaller staff.

Q2) How do the healthcare professional types differ over time?

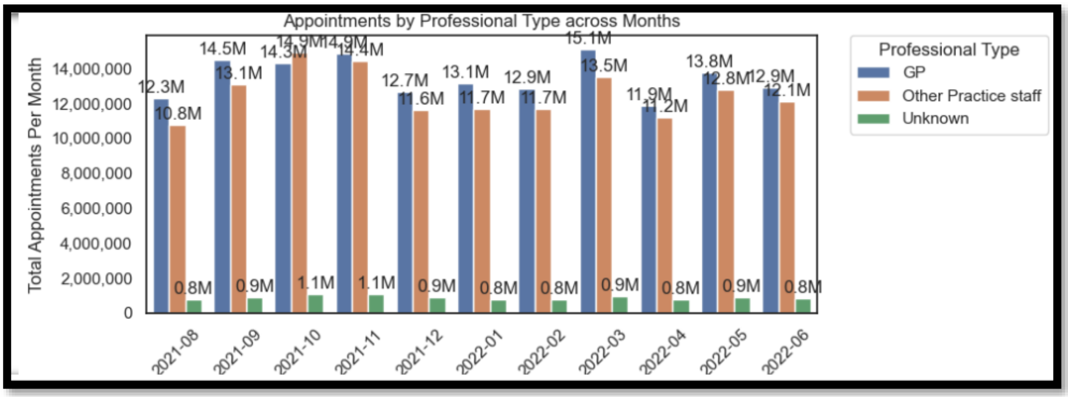
Create a new df *ar_hcp* by sub-setting the relevant columns

```
In [177]: # New df for hcp_type
          ar_hcp = ar2 [['appointment_month', 'hcp_type', 'count_of_appointments']].\
          groupby(['hcp_type', 'appointment_month']).sum().reset_index()
          ar_hcp
```

a. # Plotting with Seaborn - Bar Plot

b. Hue = hcp_type

c. Visualise



d. Use group by ‘hcp_type’ to calculate the aggregated view in the entire period by hcp_type

In [121]:
```
# groupby hcp_type
ar_hcp.groupby('hcp_type')[['count_of_appointments']].sum().sort_values('count_of_appointments', ascending=False).round(1)
```

Out[121]:

hcp_type	count_of_appointments
GP	119271733
Other Practice staff	114507199
Unknown	8059145

Q3) Are there significant changes in whether visits are attended?

These are the exact steps for generating the visualisations and groupby syntax for health care professional types. Replacing appointment_status as a reference.

```
# New df for appointment_status
ar_appointment_status = ar2 [['appointment_mode', 'appointment_status', 'count_of_appointments',
                             'hcp_type','time_between_book_and_appointment',
                             'appointment_month']].groupby(['appointment_mode','time_between_book_and_appointment',
                                                            'hcp_type','appointment_status', 'appointment_month'
                                                           ]).sum().reset_index()
ar_appointment_status
```

Are there significant changes in whether or not visits are attended?

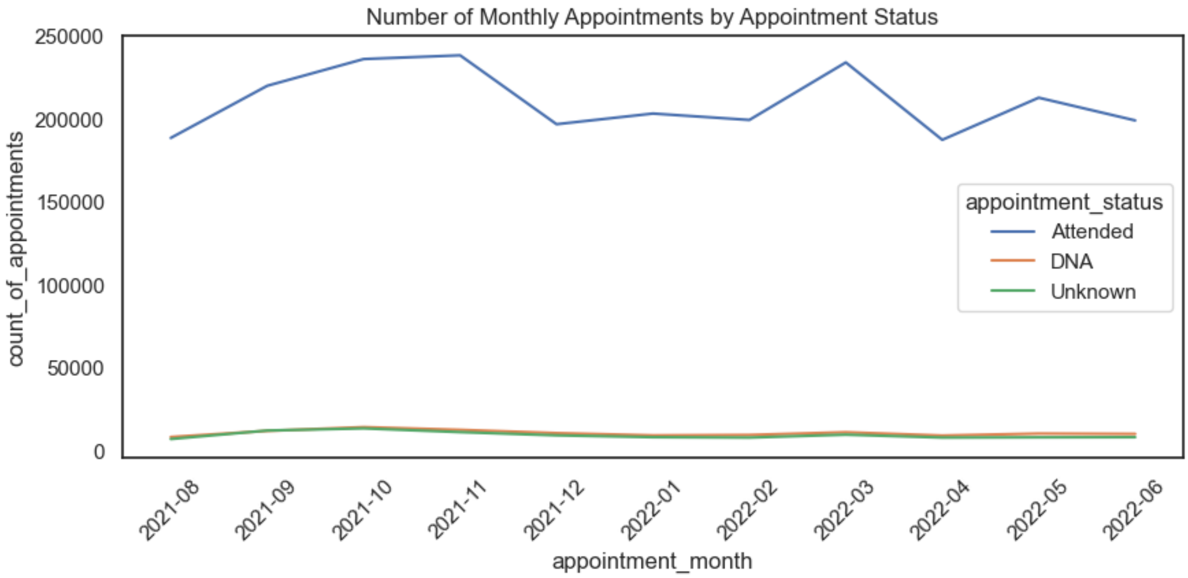
[*]:
```
# New df for appointment_status
ar_appointment_status = ar2 [['appointment_mode', 'appointment_status', 'count_of_appointments','hcp_type','time_between_book_and
ar_appointment_status
```

[*]:
```
ar_appointment_status.dtypes
```

[*]:
```
# Convert the appointment_month to string data type for ease of visualisation.
ar_appointment_status['appointment_month'] = ar_appointment_status['appointment_month'].values.astype('str')
ar_appointment_status.dtypes
```

1st: Creating a Lineplot with by Appointment_Status

[*]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Remove rows with non-numeric values in 'count_of_appointments' column
ar_appointment_status['count_of_appointments'] = pd.to_numeric(ar_appointment_status['count_of_appointments'], errors='coerce')
ar_appointment_status = ar_appointment_status.dropna(subset=['count_of_appointments'])

# Set the figure size
fig, ax = plt.subplots(figsize=(10, 4))

# Title for the plot
plt.title("Number of Monthly Appointments by Appointment Status")

# Create the line plot
sns.lineplot(x='appointment_month', y='count_of_appointments', hue='appointment_status', data=ar_appointment_status, ci=None, ax=

# Rotate x-axis labels for better visibility
plt.xticks(rotation=45)

# Show the plot
plt.show()
```



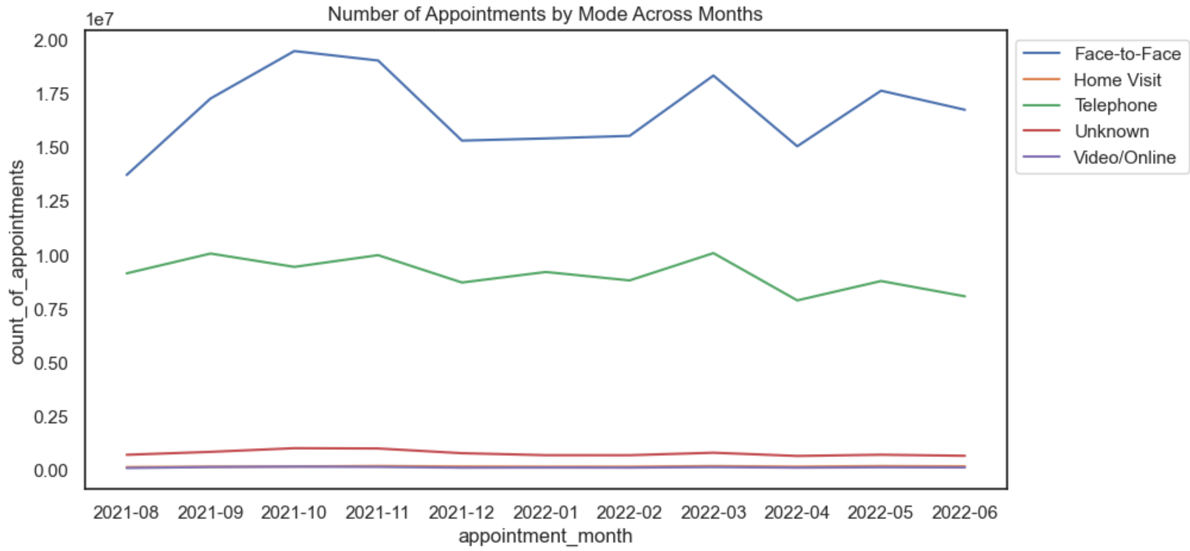
Conclusion: Overall, attended appointments make the majority of NHS dataset, with peaks in the busiest months (Oct,Nov, Mar). There is a fall in attendance in holiday periods: December – Christmas and March – Easter.

Q4) Are there changes in appointment mode in the busiest months?

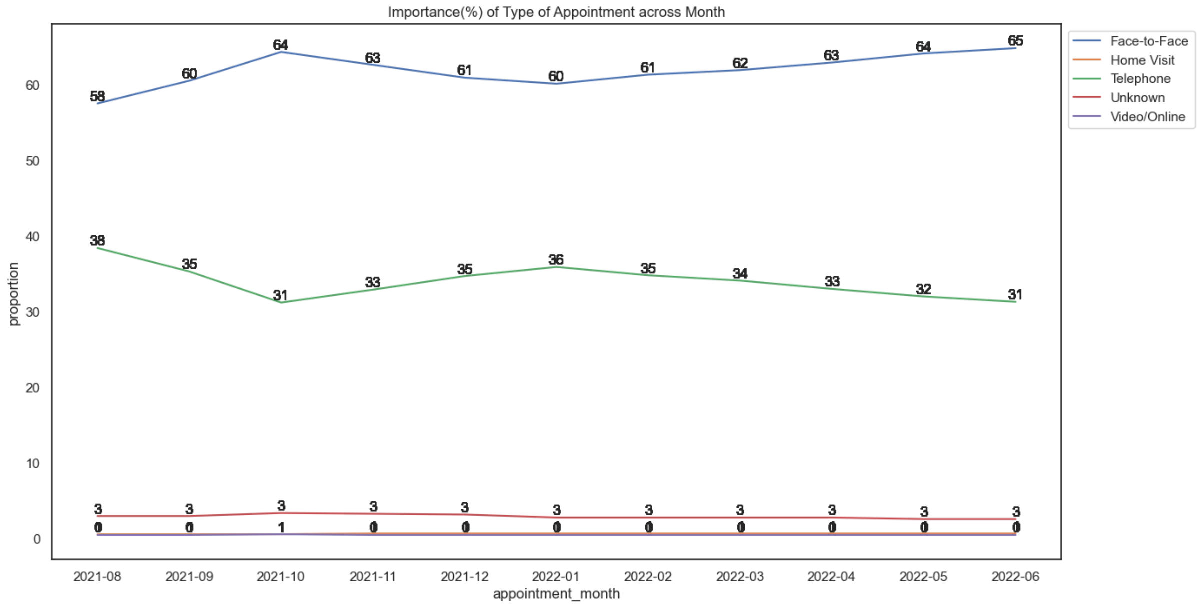
Same steps again. However, this time, I created a line plot instead of a bar plot and used appointment_mode as a reference to generate the new datasets, adapting the respective syntax.

```python
# New df for appointment_tyoe
ar_mode = ar2 [['appointment_mode', 'count_of_appointments',
                'appointment_month']].groupby(['appointment_mode','appointment_month']).sum().reset_index()
ar_mode
```

Number of Monthly Appointments by Mode



Importance (%) of Appointment_Modes across Months



Conclusion: More than 60% of appointments are face-to-face. In the busiest months (Oct,Nov, Mar) there is an uplift of face-to-face in detriment of phone-based, which gain importance during holiday periods (Nov, Dec, Jan).

Q5) Are there any trends in time between booking an appointment?

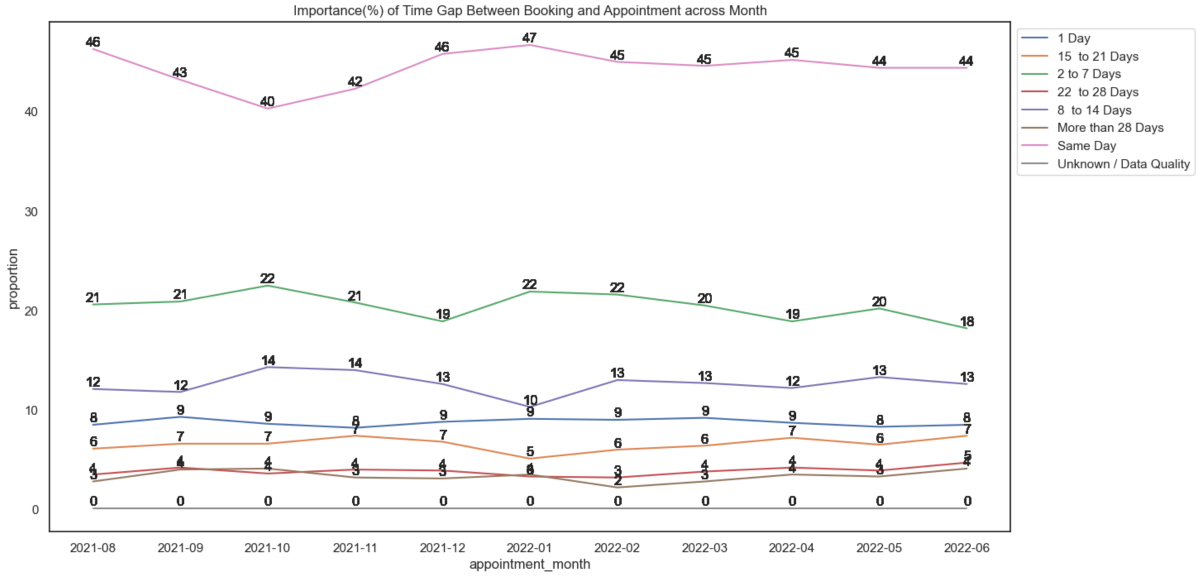
Same steps, but using time_between_book_and_appointment as a reference.

```
# New df for appointment_tyoe
ar_time = ar2 [['time_between_book_and_appointment', 'count_of_appointments','appointment_month']].\
groupby(['time_between_book_and_appointment','appointment_month']).sum().reset_index()
ar_time
```

I converted the series types to adequate formats for generating line plots with seaborn.

```
In [141]: # Convert the appointment_month to string data type for ease of visualisation.
          ar_time['appointment_month'] = ar_time['appointment_month'].values.astype('str')
          ar_time.dtypes

Out[141]: time_between_book_and_appointment     object
          appointment_month                     object
          count_of_appointments                  int64
          dtype: object

In [142]: ar_time['time_between_book_and_appointment'] = ar_time['time_between_book_and_appointment'].astype('category')
          ar_time['count_of_appointments'] = pd.to_numeric(ar_time['count_of_appointments'])
          ar_time.dtypes

Out[142]: time_between_book_and_appointment     category
          appointment_month                       object
          count_of_appointments                    int64
          dtype: object
```

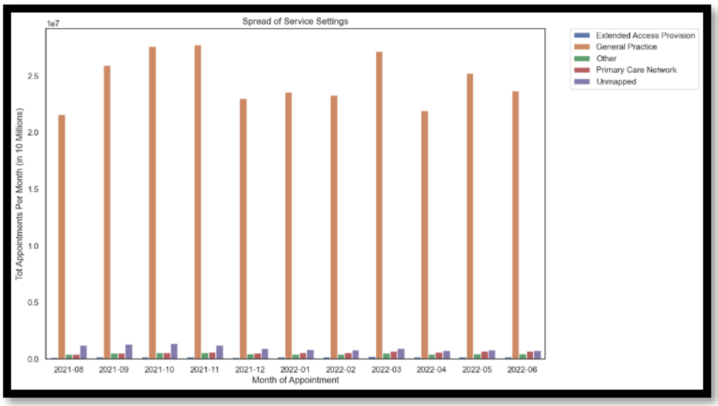
Importance(%) of Time Gap Between Booking and Appointment across Month



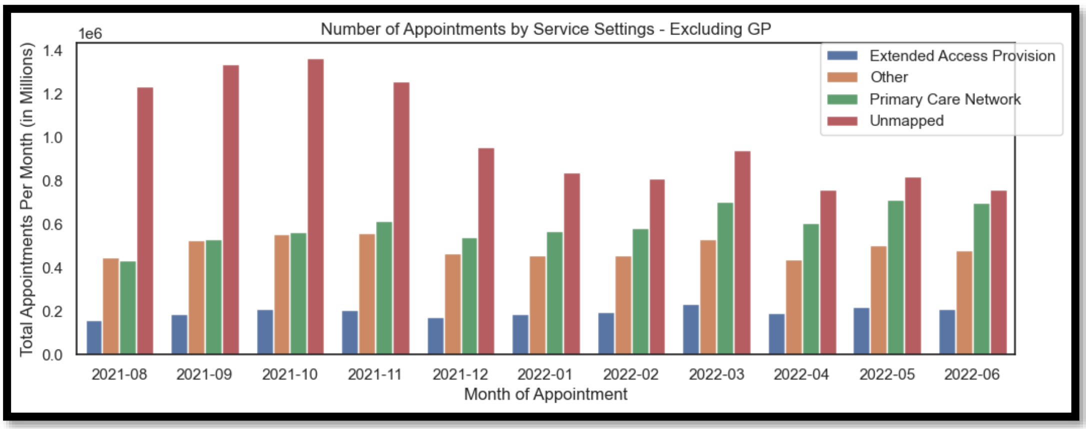
Conclusion: Most appointments are scheduled for the same day, with a gain of imp(%) in Dec/Jan. The same happens for 2-7 days. Hypothesis: there is more urgency during this period as people tend to take holidays, and accidents are more likely to occur during the festive season.

Q6: How does the spread of service settings compare?

1) Visualise national_category df again with the .head() function
2) Create a new df with service_setting, month of appointment and count_of_appointments
3) Create a boxplot to investigate the spread of service_setting



4) Create a boxplot to investigate the service settings, **excluding** GP
data=nc2[nc2.service_setting != 'General Practice']



5)**Conclusion:** GP are the vast majority of NHS services, followed by Unmapped and Primary Care,with Primary Care gaining traction from march-jun.

What are the main reasons for missed appointments?

Create a new df (ar_DNA) only with missed appointments

```
In [142]: ar_DNA = ar_appointment_status[ar_appointment_status['appointment_status'] == 'DNA']
          ar_DNA
```

Out[142]:

	appointment_mode	time_between_book_and_appointment	hcp_type	appointment_status	appointment_month	count_of_appointments
11	Face-to-Face	1 Day	GP	DNA	2021-08	16314
12	Face-to-Face	1 Day	GP	DNA	2021-09	25400
13	Face-to-Face	1 Day	GP	DNA	2021-10	28771
14	Face-to-Face	1 Day	GP	DNA	2021-11	25869
15	Face-to-Face	1 Day	GP	DNA	2021-12	23323
...	...	...	...	...	...	...
3697	Video/Online	Same Day	Unknown	DNA	2022-03	8
3698	Video/Online	Same Day	Unknown	DNA	2022-04	7
3699	Video/Online	Same Day	Unknown	DNA	2022-05	2
3700	Video/Online	Same Day	Unknown	DNA	2022-06	2
3722	Video/Online	Unknown / Data Quality	GP	DNA	2021-12	1

1220 rows × 6 columns

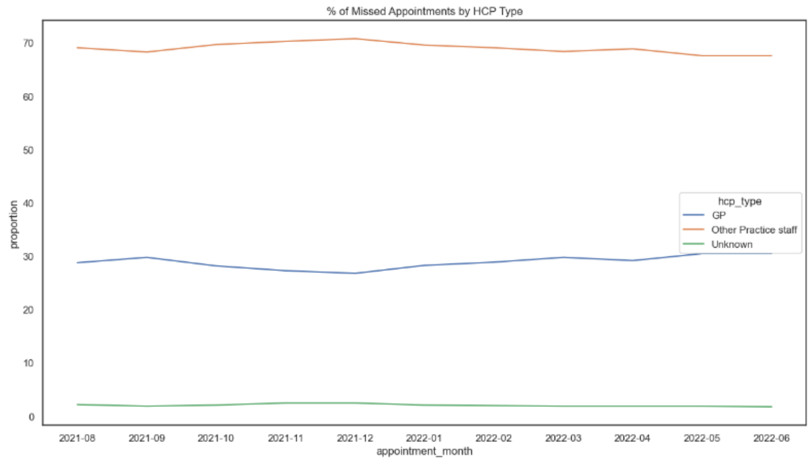
Convert

the series of ar_DNA to the suitable data types to create line plots

```
In [144]: ar_DNA['hcp_type'] = ar_DNA['hcp_type'].astype('category')
          ar_DNA['appointment_status'] = ar_DNA['appointment_status'].astype('category')
          ar_DNA['time_between_book_and_appointment'] = ar_DNA['time_between_book_and_appointment'].astype('category')
          ar_DNA['appointment_mode'] = ar_DNA['appointment_mode'].astype('category')
          ar_DNA['count_of_appointments'] = pd.to_numeric(ar_DNA['count_of_appointments'])
```

From ar_DNA, new datasets to analyse the sum and proportions of **missed appointments** among.

1) Importance (%) of HCP Type among Missed Appointments



Most of the monthly missed appointments (roughly 70%) are non-GP appointments, classified as 'Other Practice'. This proportion intensifies with the closeness to holidays.

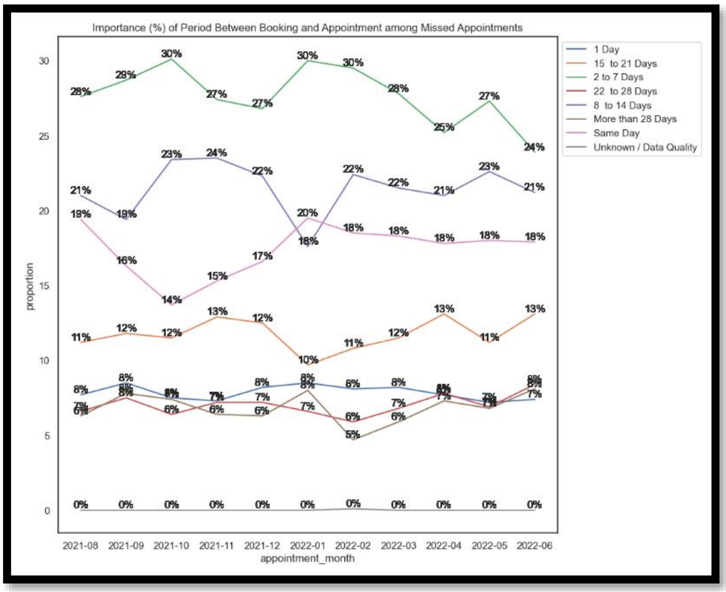
2) Number of Missed Appointments by Time Between Booking and Appointment

Which periods between booking and appointment show higher levels of missed appointments?

```
time_DNA.groupby(['time_between_book_and_appointment']).sum('count_of_appointments')\
.sort_values('count_of_appointments', ascending=False).round(1)
```

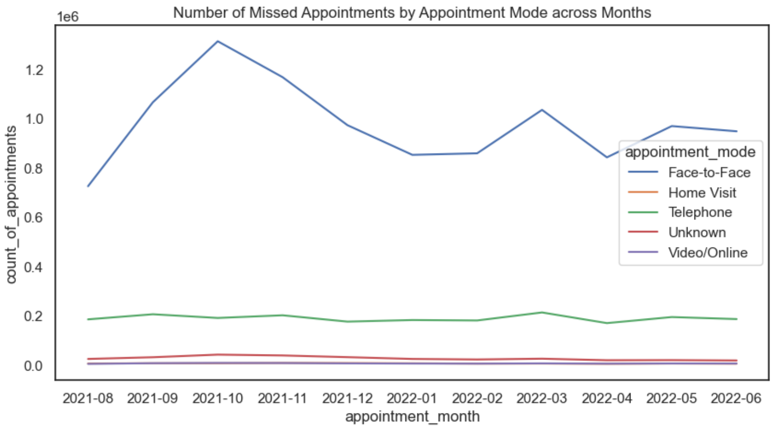
time_between_book_and_appointment	count_of_appointments	proportion
2 to 7 Days	3695097	304.3
8 to 14 Days	2869665	235.9
Same Day	2289152	191.3
15 to 21 Days	1568871	129.3
1 Day	1045140	86.3
22 to 28 Days	935245	77.3
More than 28 Days	910122	75.0
Unknown / Data Quality	5092	0.1

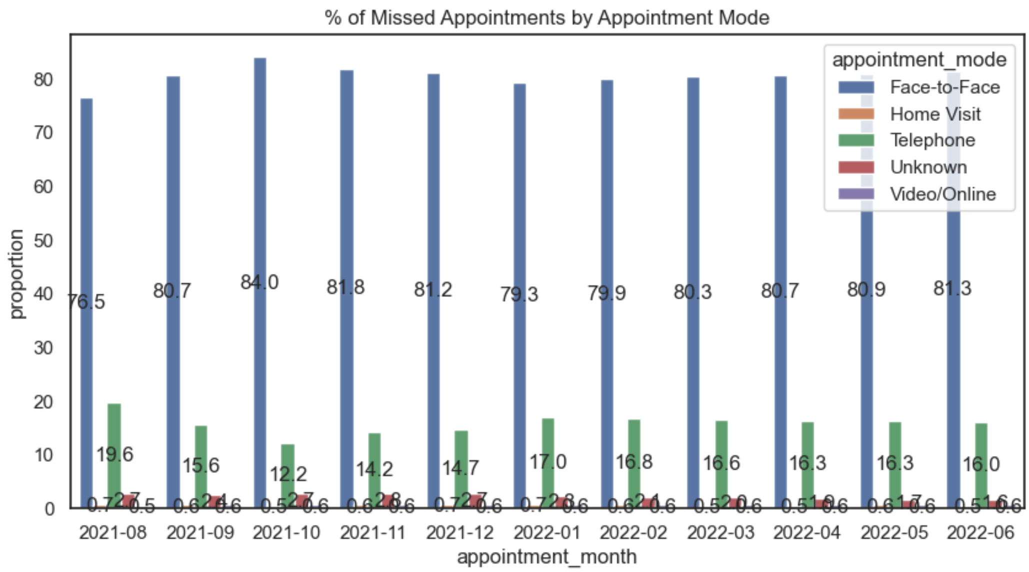
Proportion of Time Gaps between Booking and Appointment across months



Most missed appointments are made 0 to 14 days in advance. Big majority 2-7 days.However, same days appointments show a decline in missed appointments over time, probably because same day appointments tend to be more urgent.

3) Number of Missed Appointments by Appointment Mode across Months





Conclusion: Missed appointments are mostly face-to-face or phone appointments. Telephone missed appointments increase from Jan to Mar after the festive season.

TWITTER ANALYSIS

1) Installed Wordcloud and imported Wordcloud to get a cloud view of key words among top tweets
2) Focus on top favourited/top retweeted tweets/hashtags and tweets

Would it be useful to only look at retweeted and favourite tweet messages?

Explain your answer.

When analysing data scraped from Twitter, it is useful to focus on retweeted and favourited tweets for several reasons:

1. Engagement Metrics: These metrics indicate user engagement and show what content is popular or impactful.
2. Influence: Retweeted and favourited tweets often come from influential users or contain influential content, providing insights into key players or influencers. They can also help evaluate marketing campaigns by indicating which messages are gaining traction.
3. Filter Noise: Retweeted and favourited tweets filter out less relevant content, enabling more targeted analysis.
4. User Sentiment: Retweeted and favourited tweets indicate positive sentiment, which can be useful in helping understand user preferences.

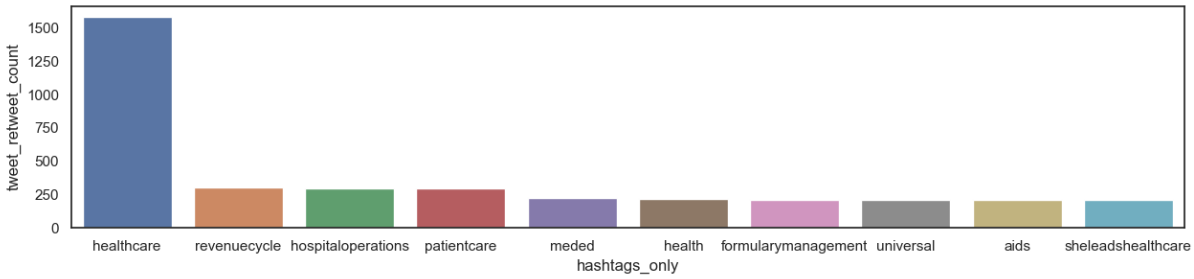
3) Cleaning and structuring the tweets df
    a. remove Nan Values from tweets['tweet_full_text']
    b. # Extracting hashtags from 'tweet_full_text' column
4) Identifying, sub-setting and plotting the Top 10 RT and Top 10 Favourited Tweets and Hashtags
5) Filter tweets with # filter tweets with words capacity, busy or saturate

Only 2 out of 1173 mentioning these words.

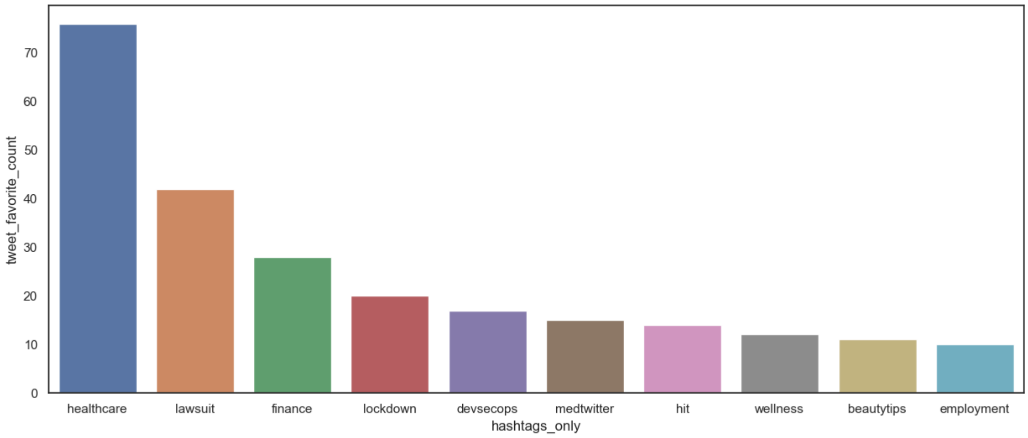
Out[82]:

	tweet_id	tweet_full_text	tweet_entities	tweet_entities_hashtags	tweet_metadata	tweet_retweet_c
436	1567635490899107841	RT @Jaci_Mullins_RP: We are almost at capacity! Register here before it is too late! https://t.co/Pbu79lQW0z\n#transformingradiology #radpar...	{'hashtags': [{'text': 'transformingradiology', 'indices': [109, 131]}], 'symbols': [], 'user_mentions': [{'screen_name': 'Jaci_Mullins_RP', 'name': 'Jaci Mullins', 'id': 1512479196123119621, 'id_...	#transformingradiology	{'iso_language_code': 'en', 'result_type': 'recent'}	
448	1567634057852231680	August 2022 was a very busy month for #cybersecurity - see some of the top stories by reading SWK's monthly recap here - https://t.co/5tYow08eSz\n\n#infosec #breach #twitter #cve #sap #healthcare ...	{'hashtags': [{'text': 'cybersecurity', 'indices': [38, 52]}, {'text': 'infosec', 'indices': [146, 154]}, {'text': 'breach', 'indices': [155, 162]}, {'text': 'twitter', 'indices': [163, 171]}, {'t...	#cybersecurity, #infosec, #breach, #twitter, #cve, #sap, #healthcare, #ransomware, #cisa, #cyberinsurance	{'iso_language_code': 'en', 'result_type': 'recent'}	

6) Bar plot for Top 10 Retweeted hashtags



7) barplot for top 10 favourited hashtags



8) Using Wordcloud To Explore Hashtag And Twitter Texts

- Worldcloud: NHSH ashtags

- Worldcloud : NHS Tweets Texts

Conclusion: Words associated with staff capacity (capacity, busy, saturate) do not appear in the top retweeted/favorited hashtags, top tweets text and in the generated world clouds. Of the

1173 tweets analysed, only two mention capacity, one of which is irrelevant (content about cybersecurity, not about staff). However, there are mentions of words such as employment and job that might indicate a buzz towards the NHS capacity issue.

APPENDIX

---

actual_duration.csv:

<table>
<tr><td>actual_duration</td><td>the length in minutes of the consultation</td></tr>
<tr><td>appointment_date</td><td>The date the patient made the appointment
(01/12/2021 – 30/06/2022)</td></tr>
<tr><td>icb_ons_code</td><td rowspan="5">NHS geographical codes</td></tr>
<tr><td>region_ons_code</td></tr>
<tr><td>sub_icb_location_code</td></tr>
<tr><td>sub_icb_location_name</td></tr>
<tr><td>sub_icb_location_ons_code</td></tr>
<tr><td>count_of_appointments</td><td>Estimated number of total appointments (NUMERIC)</td></tr>
</table>

appointments_regional.csv:

appointment_mode	The mode of the appointment shows the setting of the consultation (telephone, face-to-face etc)
appointment_month	The month in which the appointment is. (Jan -2020 to Jun -2022)
appointment_status	shows whether the appointment is available, booked, has been attended by the patient (or not), or has been cancelled.
hcp_type	Healthcare professional typ
time_between_book_and_appointment	Time from when the booking was made to when the appointment happened

icb_ons_code	ICB: Inpatient Care Base
count_of_appointments	Estimated number of total appointments (NUMERIC)

national_categories.csv:

appointment_date	The date the appointment was made for by the patient (01/08/2021 – 30/06/2022)
appointment_month	The month in which the appointment is.
context_type	whether an appointment is an encounter relating to direct patient care, or an activity undertaken as part of patient care where the patient is not involved
national_category	within each context type, there is an option for practices to choose a 'Does Not Fit, opening space to future new classifications of context types
service_setting	the provision of care in which the appointment was delivered (GP, Primary Care etc)
icb_ons_code	ICB: Inpatient Care Base
sub_icb_location_name	SUB ICB: Sub-Inpatient Care Base
count_of_appointments	Estimated number of total appointments (NUMERIC)

Tweets

tweet_id
tweet_full_text
tweet_entities
tweet_entities_hashtags
tweet_metadata
tweet_retweet_count
tweet_favorite_count
tweet_favorited
tweet_retweeted
tweet_lang