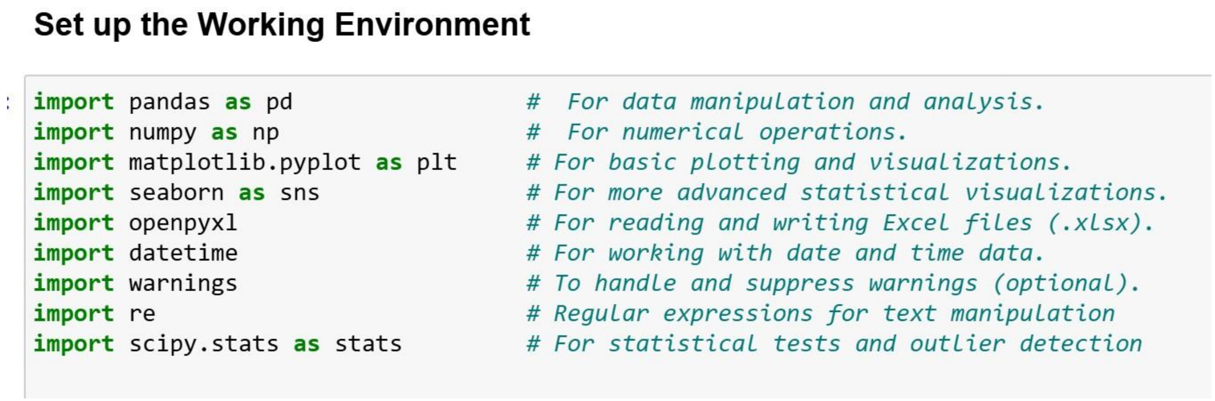
# Appendix

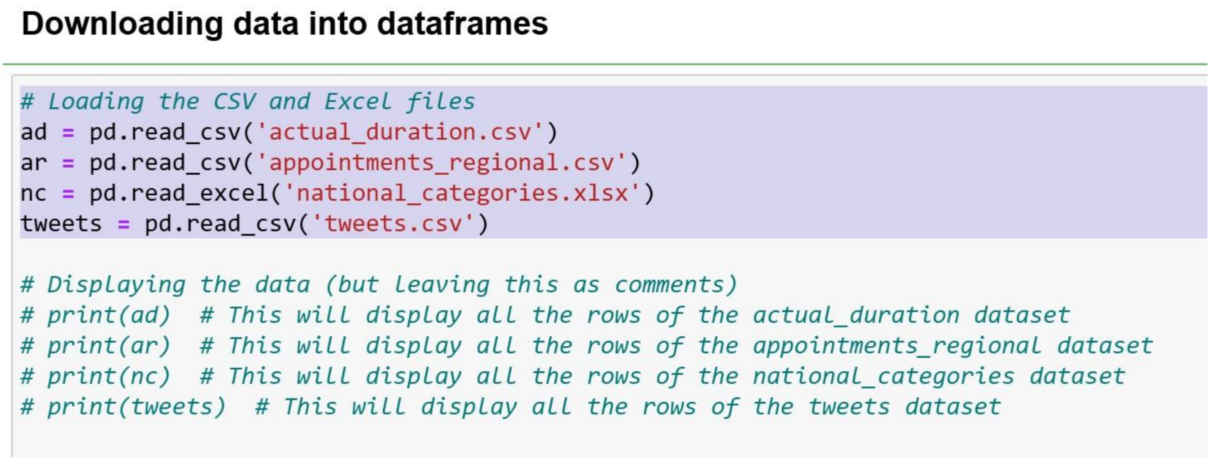
**Best practices in Python coding:**

Defining variables correctly and maintaining consistent naming conventions are key to errorfree data manipulation. Adhering to PEP 8 guidelines ensures clarity and readability. This is essential for accurate analysis and visualisations.



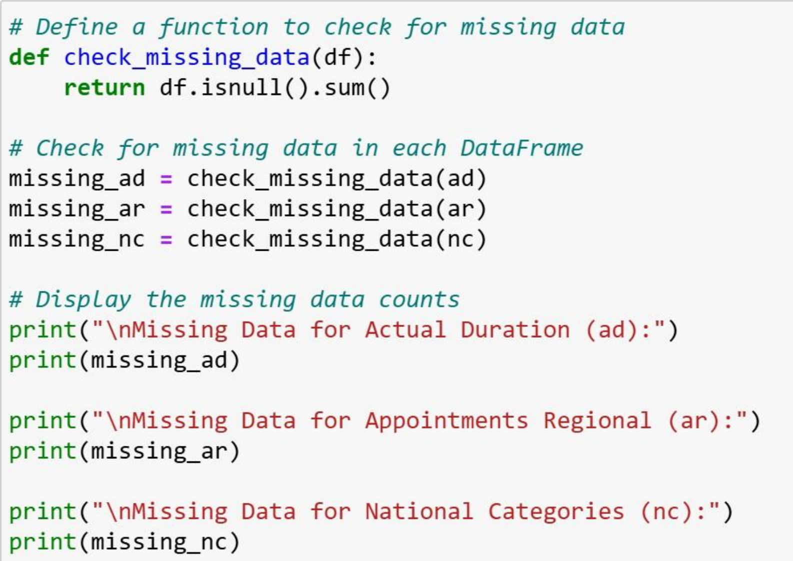
## Set up the Working Environment

```
import pandas as pd                  #  For data manipulation and analysis.
import numpy as np                   #  For numerical operations.
import matplotlib.pyplot as plt      # For basic plotting and visualizations.
import seaborn as sns                # For more advanced statistical visualizations.
import openpyxl                      # For reading and writing Excel files (.xlsx).
import datetime                      # For working with date and time data.
import warnings                      # To handle and suppress warnings (optional).
import re                            # Regular expressions for text manipulation
import scipy.stats as stats          # For statistical tests and outlier detection
```

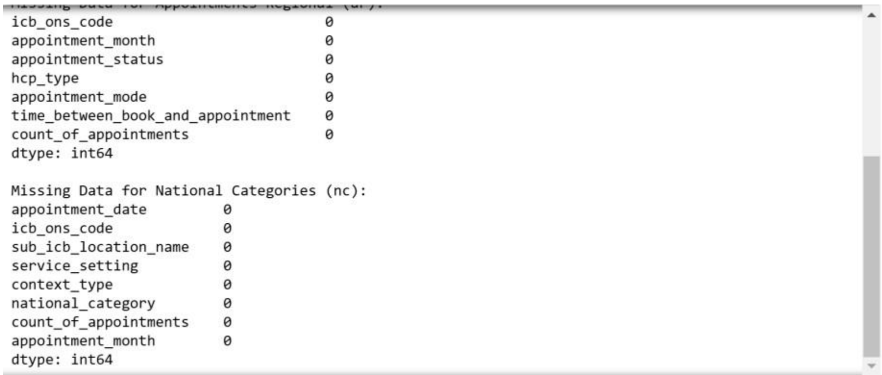


## Downloading data into dataframes

```
# Loading the CSV and Excel files
ad = pd.read_csv('actual_duration.csv')
ar = pd.read_csv('appointments_regional.csv')
nc = pd.read_excel('national_categories.xlsx')
tweets = pd.read_csv('tweets.csv')

# Displaying the data (but leaving this as comments)
# print(ad)  # This will display all the rows of the actual_duration dataset
# print(ar)  # This will display all the rows of the appointments_regional dataset
# print(nc)  # This will display all the rows of the national_categories dataset
# print(tweets)  # This will display all the rows of the tweets dataset
```

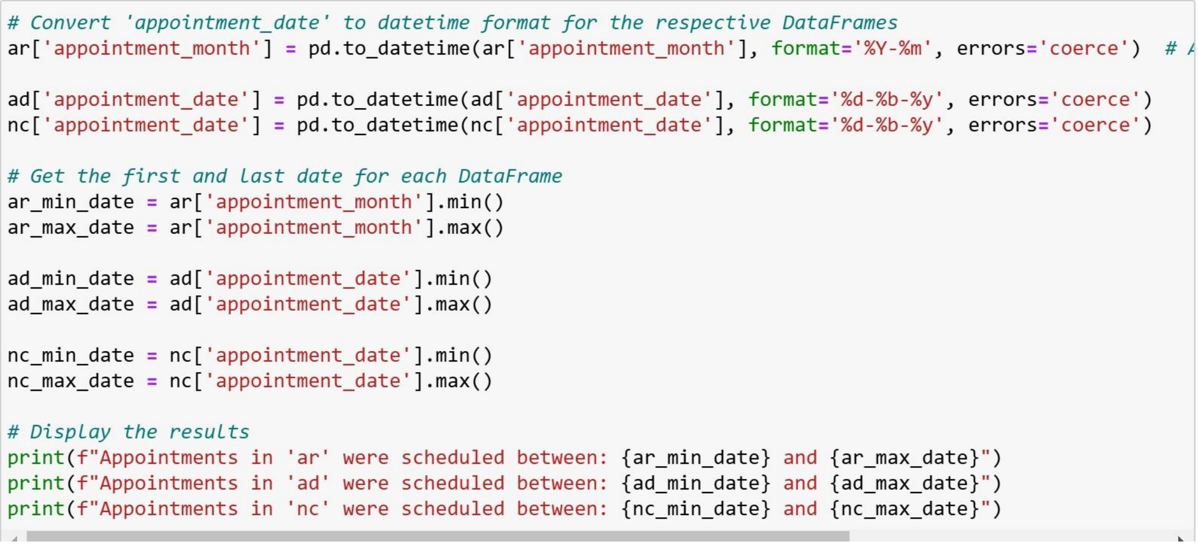


```
# Define a function to check for missing data
def check_missing_data(df):
    return df.isnull().sum()

# Check for missing data in each DataFrame
missing_ad = check_missing_data(ad)
missing_ar = check_missing_data(ar)
missing_nc = check_missing_data(nc)

# Display the missing data counts
print("\nMissing Data for Actual Duration (ad):")
print(missing_ad)

print("\nMissing Data for Appointments Regional (ar):")
print(missing_ar)

print("\nMissing Data for National Categories (nc):")
print(missing_nc)
```

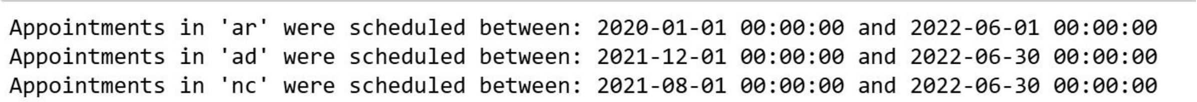


```
Missing Data for Appointments Regional (ar):
icb_ons_code                         0
appointment_month                    0
appointment_status                   0
hcp_type                             0
appointment_mode                     0
time_between_book_and_appointment    0
count_of_appointments                0
dtype: int64

Missing Data for National Categories (nc):
appointment_date         0
icb_ons_code             0
sub_icb_location_name    0
service_setting          0
context_type             0
national_category        0
count_of_appointments    0
appointment_month        0
dtype: int64
```

**This output shows that there are no missing data or null/NaN values in any of the specified columns across the three datasets (ad, ar, nc). This indicates that the dat quality is good.**

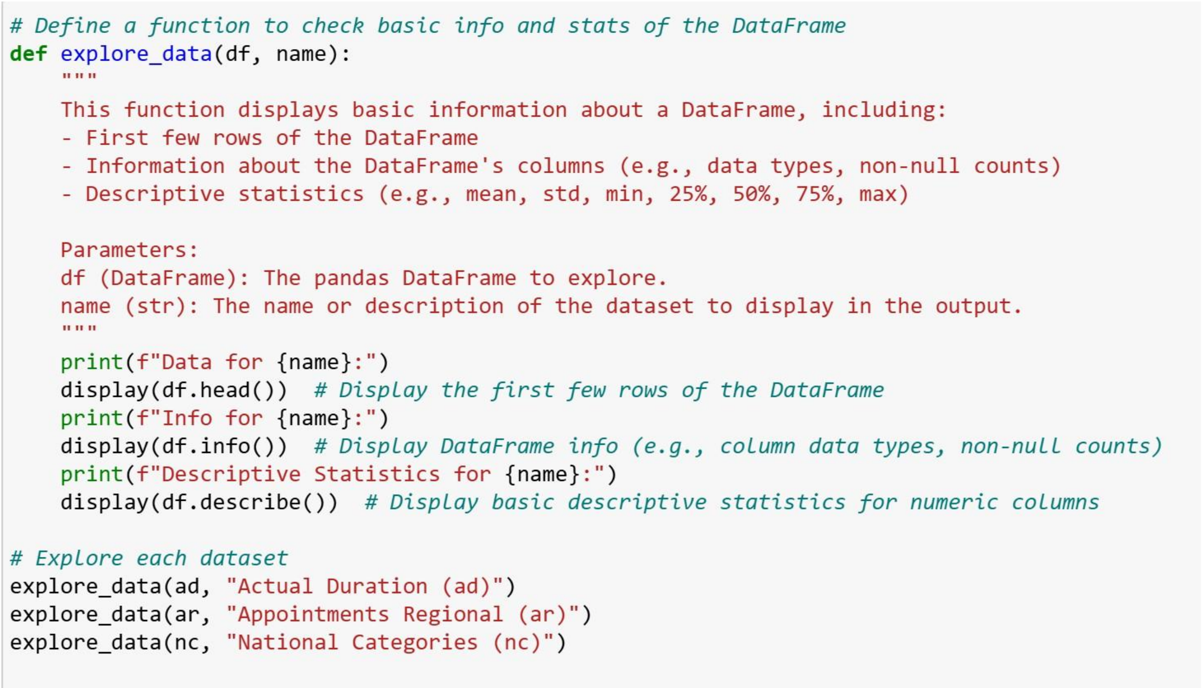


```
# Convert 'appointment_date' to datetime format for the respective DataFrames
ar['appointment_month'] = pd.to_datetime(ar['appointment_month'], format='%Y-%m', errors='coerce')  # A

ad['appointment_date'] = pd.to_datetime(ad['appointment_date'], format='%d-%b-%y', errors='coerce')
nc['appointment_date'] = pd.to_datetime(nc['appointment_date'], format='%d-%b-%y', errors='coerce')

# Get the first and last date for each DataFrame
ar_min_date = ar['appointment_month'].min()
ar_max_date = ar['appointment_month'].max()

ad_min_date = ad['appointment_date'].min()
ad_max_date = ad['appointment_date'].max()

nc_min_date = nc['appointment_date'].min()
nc_max_date = nc['appointment_date'].max()

# Display the results
print(f"Appointments in 'ar' were scheduled between: {ar_min_date} and {ar_max_date}")
print(f"Appointments in 'ad' were scheduled between: {ad_min_date} and {ad_max_date}")
print(f"Appointments in 'nc' were scheduled between: {nc_min_date} and {nc_max_date}")
```



```
Appointments in 'ar' were scheduled between: 2020-01-01 00:00:00 and 2022-06-01 00:00:00
Appointments in 'ad' were scheduled between: 2021-12-01 00:00:00 and 2022-06-30 00:00:00
Appointments in 'nc' were scheduled between: 2021-08-01 00:00:00 and 2022-06-30 00:00:00
```

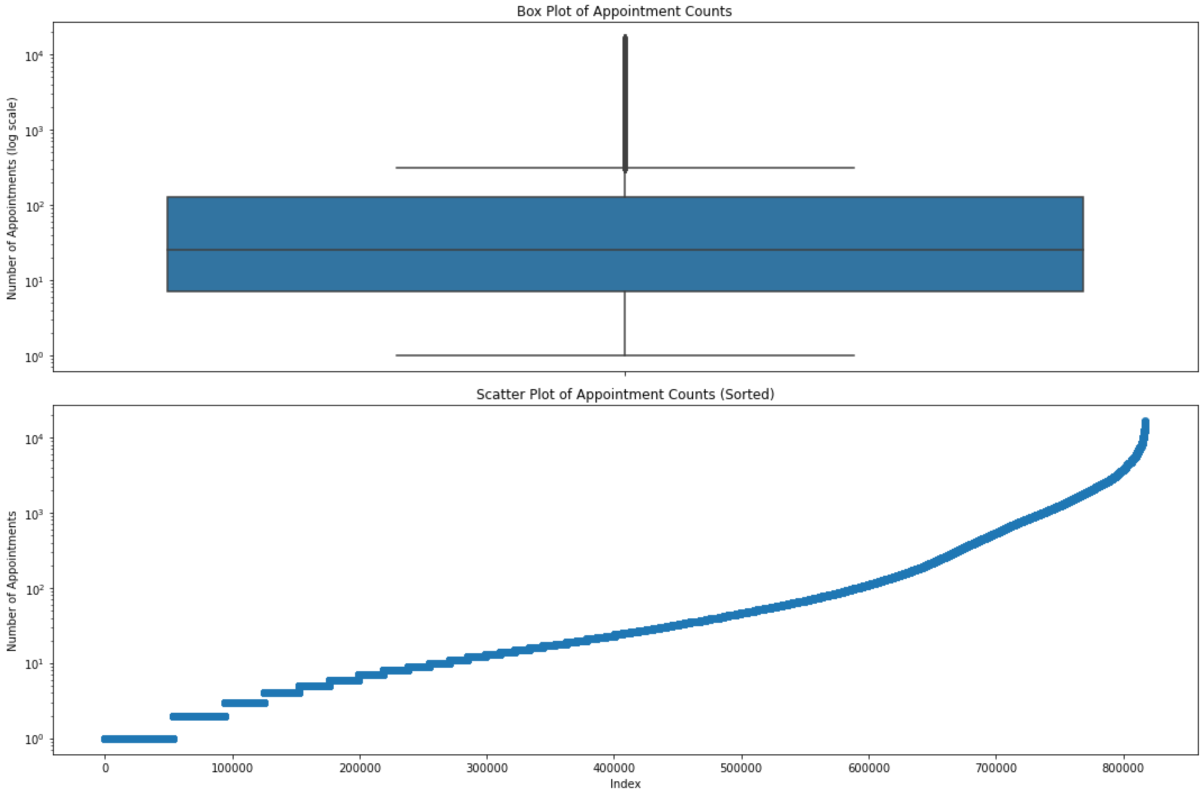
## Descriptive stats



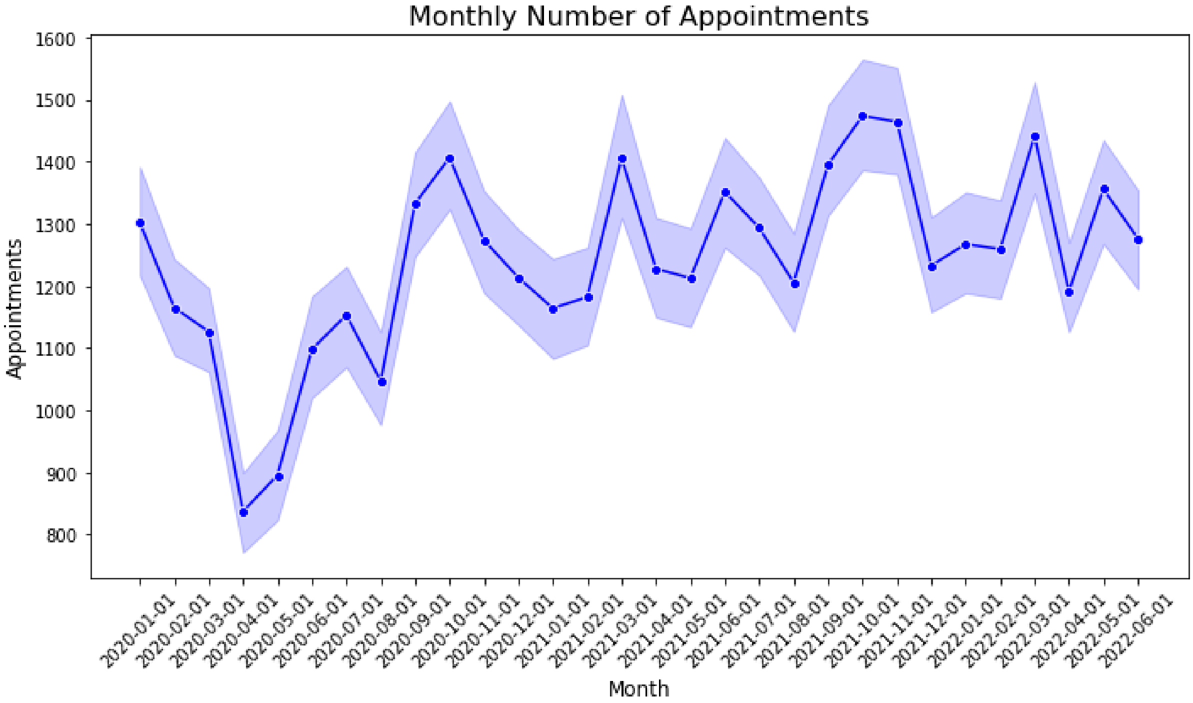
```
# Define a function to check basic info and stats of the DataFrame
def explore_data(df, name):
    """
    This function displays basic information about a DataFrame, including:
    - First few rows of the DataFrame
    - Information about the DataFrame's columns (e.g., data types, non-null counts)
    - Descriptive statistics (e.g., mean, std, min, 25%, 50%, 75%, max)

    Parameters:
    df (DataFrame): The pandas DataFrame to explore.
    name (str): The name or description of the dataset to display in the output.
    """
    print(f"Data for {name}:")
    display(df.head())  # Display the first few rows of the DataFrame
    print(f"Info for {name}:")
    display(df.info())  # Display DataFrame info (e.g., column data types, non-null counts)
    print(f"Descriptive Statistics for {name}:")
    display(df.describe())  # Display basic descriptive statistics for numeric columns

# Explore each dataset
explore_data(ad, "Actual Duration (ad)")
explore_data(ar, "Appointments Regional (ar)")
explore_data(nc, "National Categories (nc)")
```

Below are some charts for visualisation.

## 1.Visualisation of outliers:

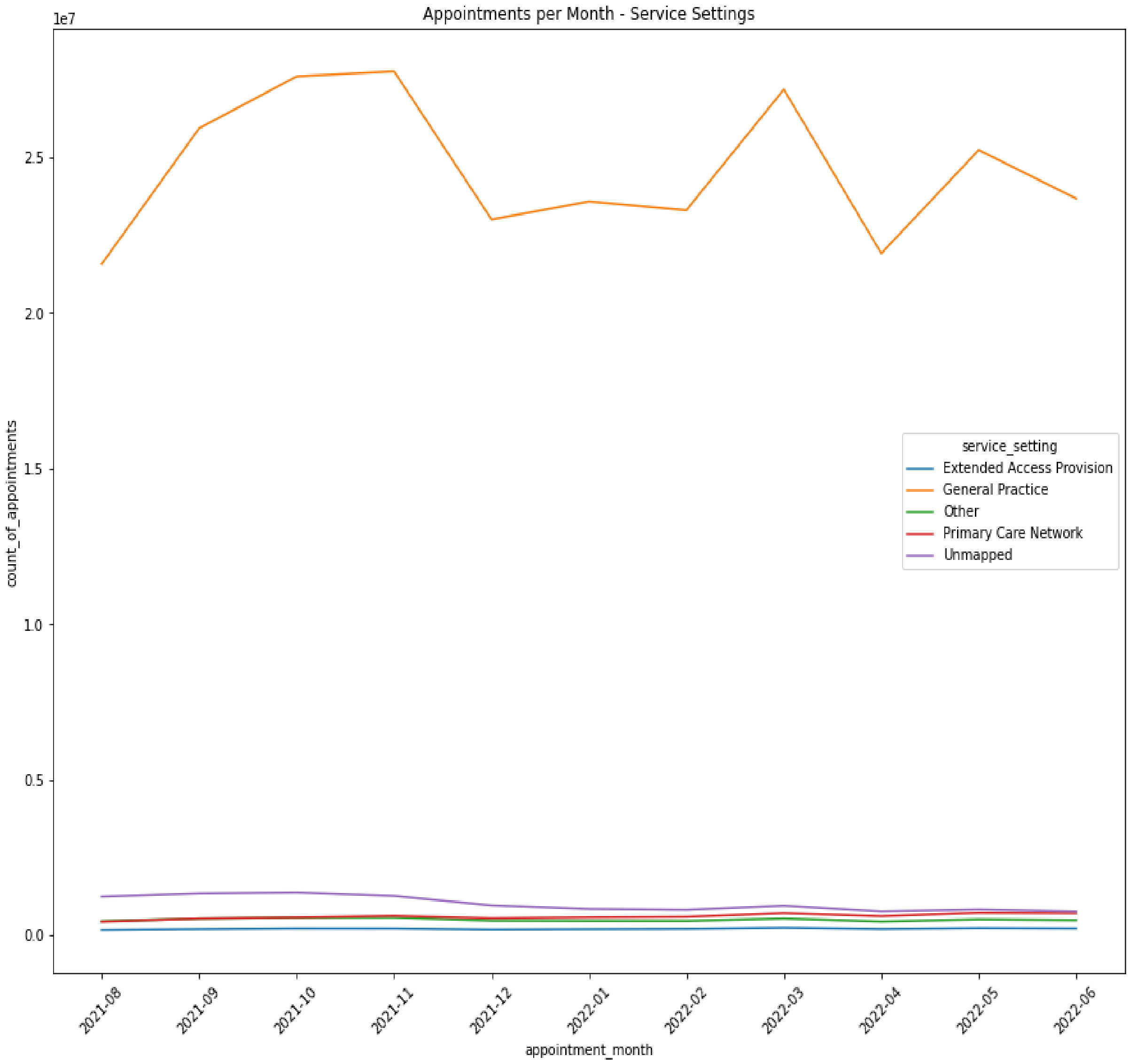


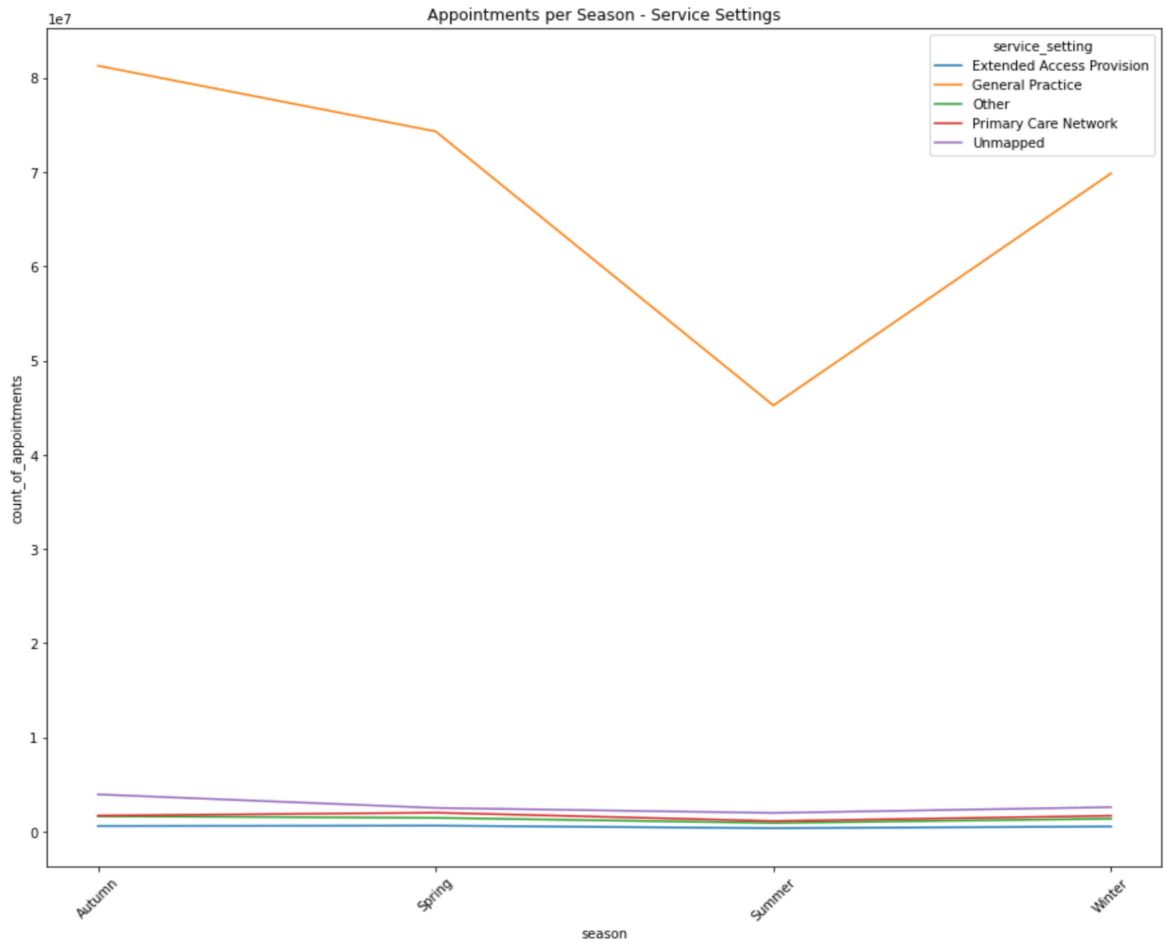
2.



The chart reveals seasonal trends, demand fluctuations, and potential outliers in appointment numbers. It helps identify peak periods, ensuring better resource planning and operational adjustments.

## 3. Aggregate appointments per season/Service setting

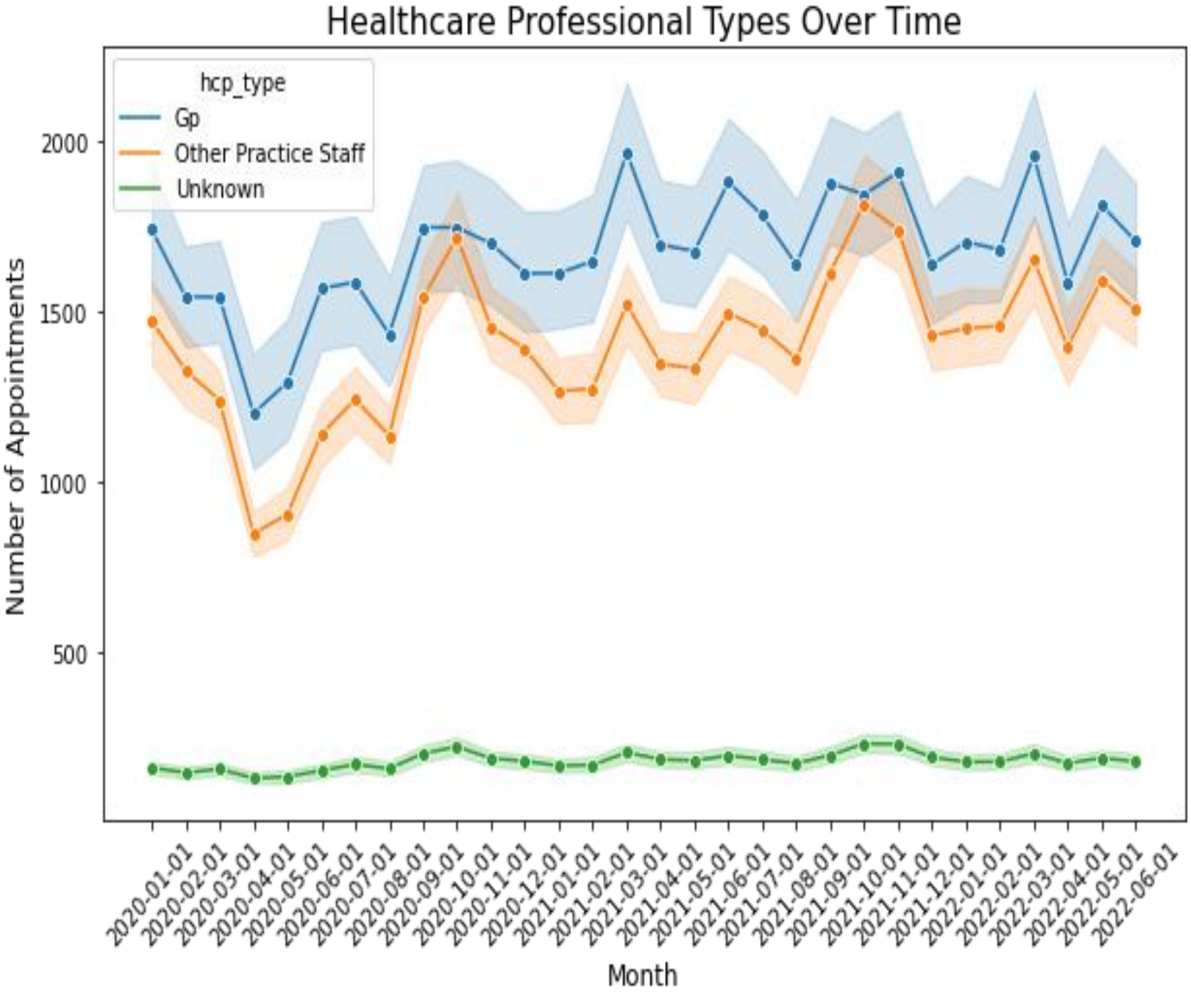




Spring and autumn have high demand, while summer dips and winter rises. NHS should adjust resources accordingly, reduce staff in summer, and focus on proactive campaigns. Use telemedicine for routine visits, run winter flu campaigns, and streamline urgent care.

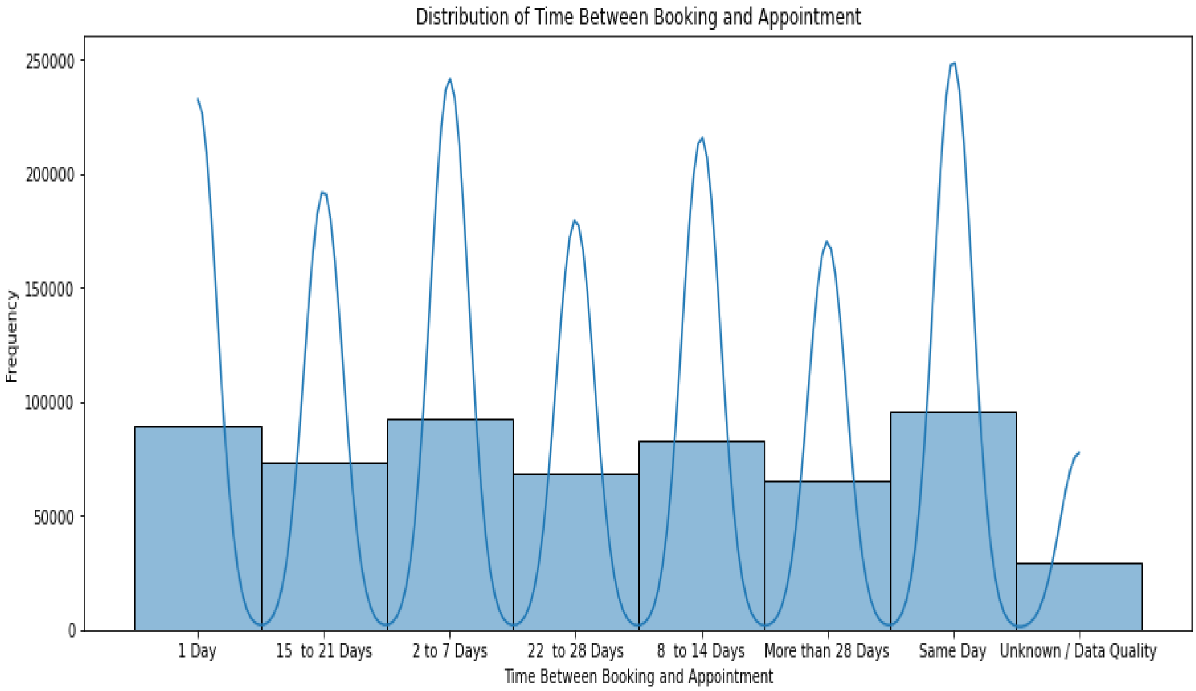
Implement predictive scheduling and allow winter pre-bookings to manage peak demand**.**

## 4. Number of appointments for each healthcare professional



The chart tracks the number of appointments for each healthcare professional type (hcp_type) over time, showing trends for each category by month. It allows for the identification of seasonality, fluctuations, and demand spikes for specific HCP types. By comparing trends, it can highlight which professionals are in higher demand. This information is useful for forecasting staffing needs and optimising resource allocation across healthcare services.

## 5. Distribution of time



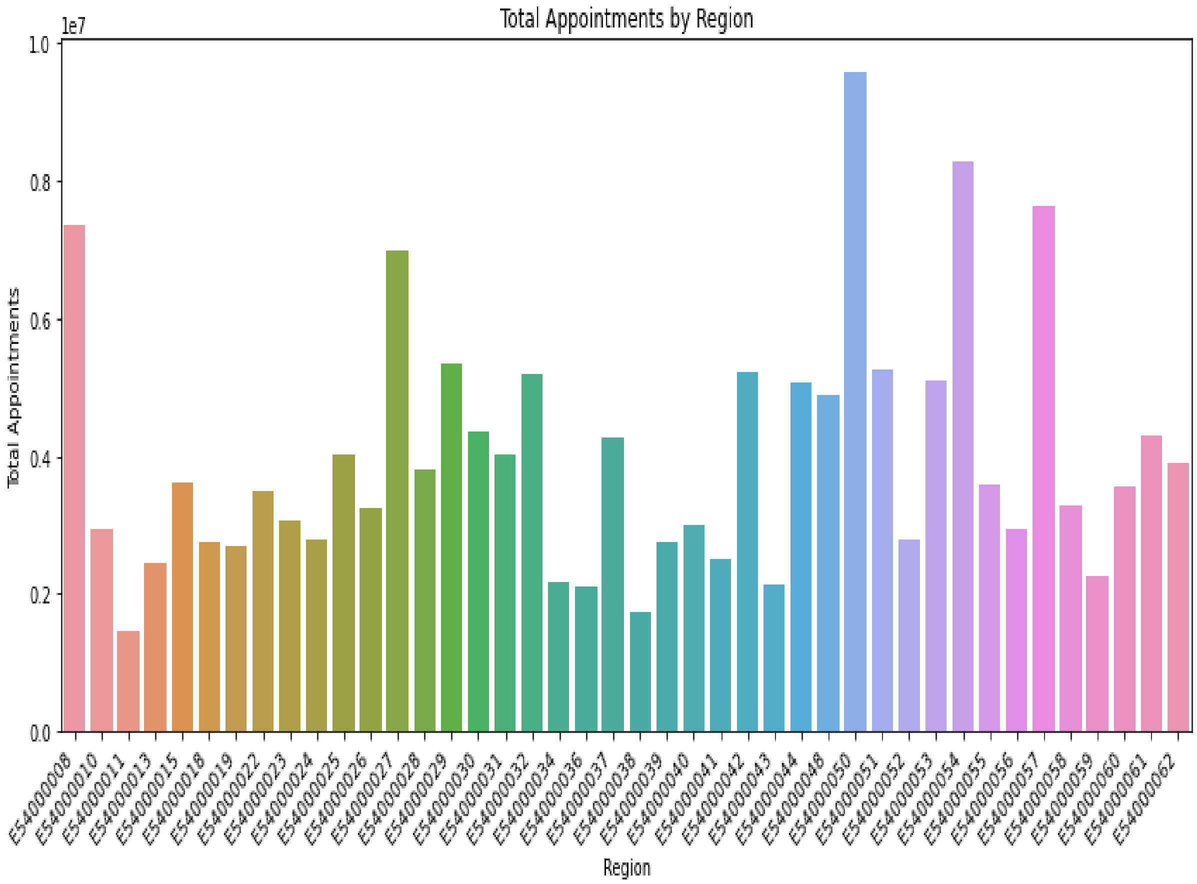
*The same-day appointment booking being the highest suggests that a significant number of patients are scheduling appointments on short notice.*

*The 2-7 days category being the second-highest indicates that many patients are still booking with relatively short lead times.*

*The 8-14 days and 1-day categories having similar numbers suggest that patients are scheduling with varying degrees of foresight but still not far in advance.*

*The least number of appointments in the unknown/date quality category suggests that most appointment dates are being recorded correctly.*

## 6. *Appointments by region*

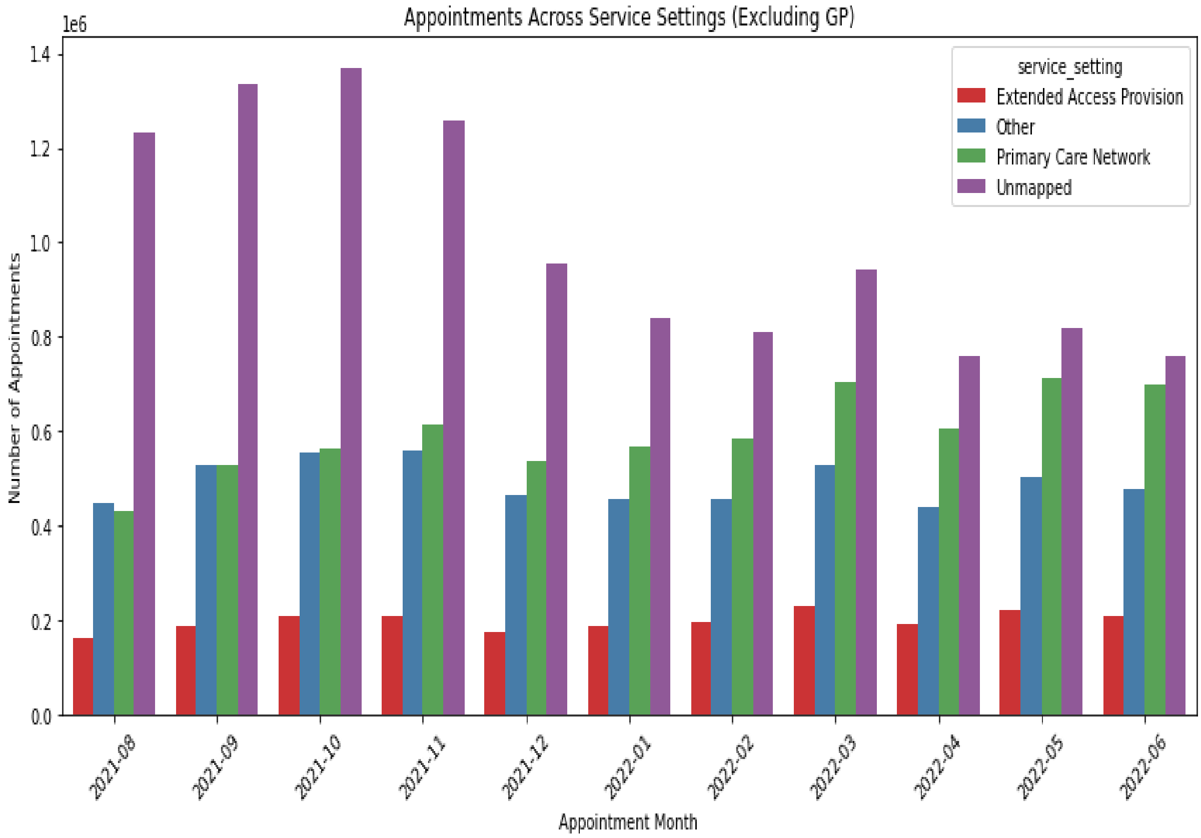


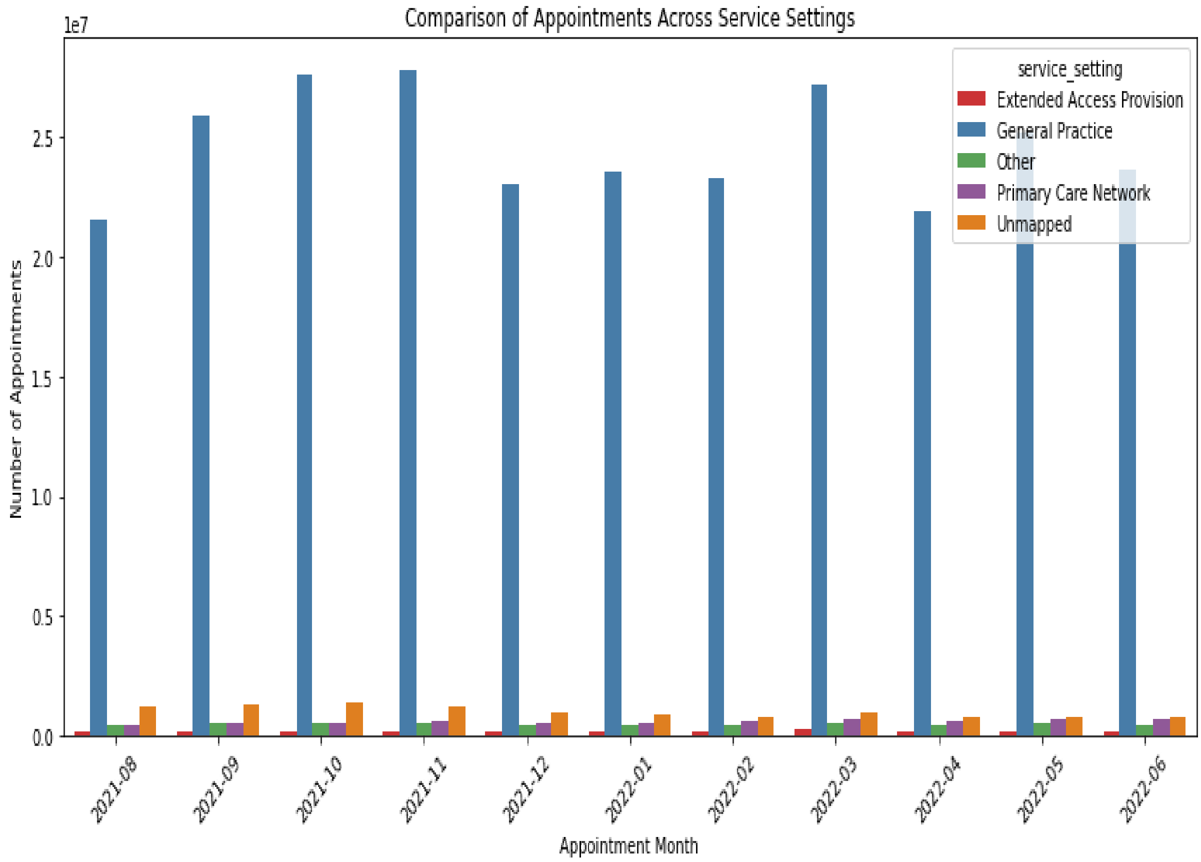
*This chart shows that E54000050, which is the NHS North East and North Cumbria, has the highest number of appointments.*

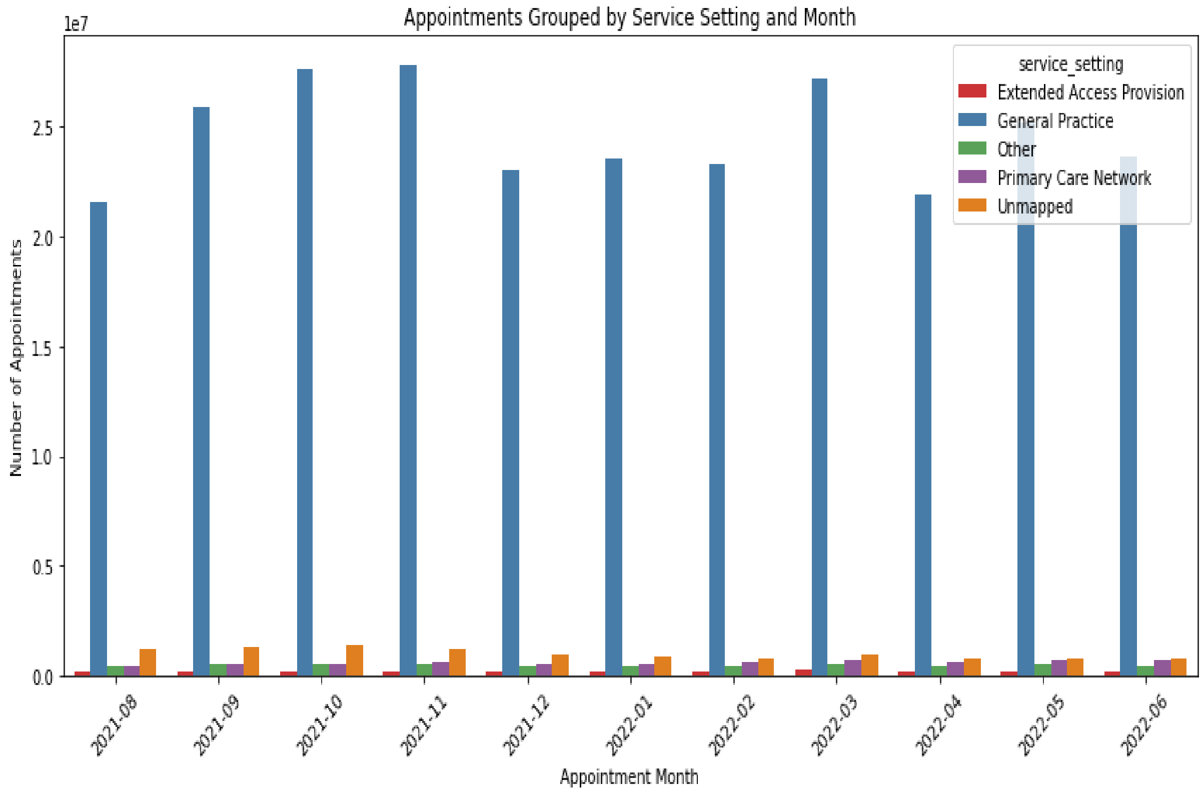
*The least is E54000011, which is NHS Shropshire Telford and Wrekin.*

**7.Aggregate appointments per month by service settings, context types, and national categories.**

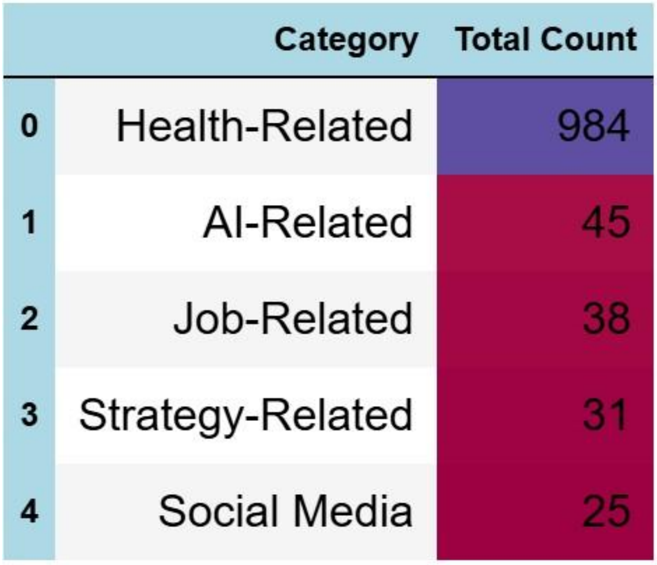
GPs have higher appointment volumes, likely driven by routine care and seasonal trends. Other service settings show lower volumes, indicating more specialised and less frequent appointments. Here is a comparison.


Appointments Across Service Settings (Excluding GP)
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Number of Appointments
service_setting
Extended Access Provision
Other
Primary Care Network
Unmapped
2021-08
2021-09
2021-10
2021-11
2021-12
2022-01
2022-02
2022-03
2022-04
2022-05
2022-06
Appointment Month


Comparison of Appointments Across Service Settings
1e7
0.0
0.5
1.0
1.5
2.0
2.5
Number of Appointments
service_setting
Extended Access Provision
General Practice
Other
Primary Care Network
Unmapped
2021-08
2021-09
2021-10
2021-11
2021-12
2022-01
2022-02
2022-03
2022-04
2022-05
2022-06
Appointment Month

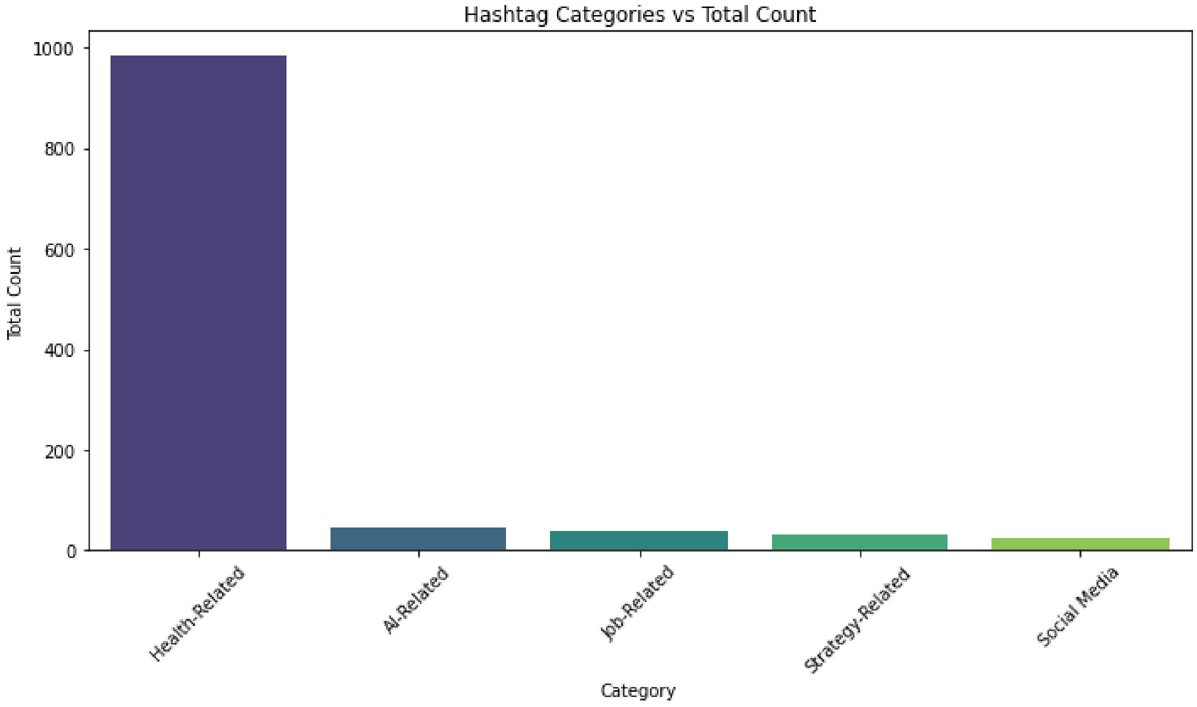

Appointments Grouped by Service Setting and Month
1e7
0.0
0.5
1.0
1.5
2.0
2.5
Number of Appointments
service_setting
Extended Access Provision
General Practice
Other
Primary Care Network
Unmapped
2021-08
2021-09
2021-10
2021-11
2021-12
2022-01
2022-02
2022-03
2022-04
2022-05
2022-06
Appointment Month

### *8.Top trending hashtags*



| | Category | Total Count |
|---|---|---|
| 0 | Health-Related | 984 |
| 1 | AI-Related | 45 |
| 2 | Job-Related | 38 |
| 3 | Strategy-Related | 31 |
| 4 | Social Media | 25 |

### 9.Visualisation of the trending Hashtags



***The majority of the conversations are health-related. There's a clear focus on health topics, with emerging interest in AI and social media's role in healthcare.***

*Recommendations:*

- Plan for seasonal staffing peaks and flexible workforce models to manage high GP appointment demand during specific months.
- Create targeted campaigns to raise awareness of primary care, extended access provision, and unmapped services.
- Create a Seasonal Appointment Management System.
- Implement campaigns, training, and tech improvements to boost telephone and video consultations, reducing pressure on in-person appointments.
- Use Twitter data to monitor public sentiment and emerging health trends for resource optimisation and timely decision-making.
- Introduce appointment reminders and conduct analysis to better understand and reduce patient no-shows.
- Monitor and respond to negative feedback in real-time, track health trends on Twitter.
- Increase GP availability, prioritise urgent appointments and seasonal spikes.
- Leverage other practice staff for non-urgent cases to ease pressure during high demand periods.
- Implement predictive scheduling using historical data to adjust appointment availability and allow pre-booking for high-demand winter periods.
- Monitor trending hashtags and engage with public, respond to emergencies and adapt services.
- Use predictive algorithms to optimise scheduling, same-day appointments and expand telehealth for non-urgent visits.
- Redistribute resources, and improve data quality for better forecasting, optimising efficiency and service delivery.
- Reduce staff in summer, and focus on proactive campaigns.
- Implement reminders, flexible scheduling, and improve data reporting to enhance attendance and decision-making.
- Run winter flu campaigns, streamline urgent care, and use predictive scheduling with pre-bookings to manage peak demand.
- identify and respond to negative feedback, adjusting services or communication strategies to improve public perception.
- Allocate more resources to NHS North East and North Cumbria; investigate access issues or underutilisation in NHS Shropshire Telford and Wrekin.
- Utilise other practice staff (e.g., nurses, physiotherapists) for non-urgent cases to alleviate pressure on GPs during high-demand periods.