



ANALYSIS REPORT

2024

Prepared By:
Beata Faitli

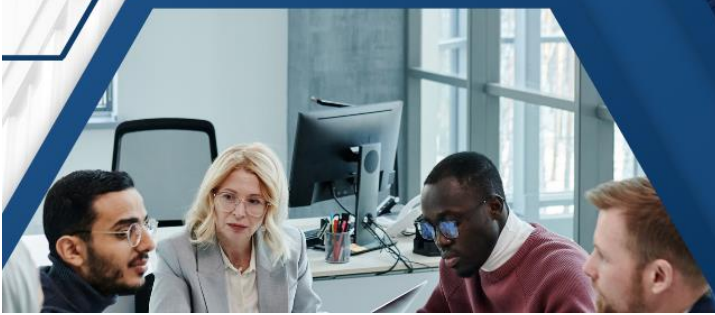




Table of Contents

Executive Summary	2
NHS (National Health Service) Data Analysis Report	3
APPENDIX 1 – Business questions	6
APPENDIX 2 – Datasets provided	10
APPENDIX 3 – NHS context and background information	14
APPENDIX 4 – Pre-analysis problem mapping	16
Stakeholder Analysis.....	16
5 Whys Analysis of the problem.....	19
SWOT Analysis of the scenario	21
APPENDIX 5 – Data Analysis steps	23
APPENDIX 6 – Visualisation for Business Stakeholders Rationale	41
APPENDIX 7 – Recommendations	43
APPENDIX 8 – REFERENCES.....	46



Executive Summary

This analysis aimed to evaluate the NHS's staffing adequacy, capacity, and resource utilization across different segments from January 2020 to July 2022, a period significantly impacted by the COVID-19 pandemic. Key findings reveal that the NHS operated above its capacity during peak periods, particularly on weekdays, with notable regional disparities in staffing. London, for instance, had lower staffing levels and higher missed appointment rates compared to other regions, highlighting the strain on resources in densely populated areas.

The analysis also showed a significant shift towards telephone consultations during the pandemic, although the adoption of video consultations remained low, indicating potential areas for improvement in telehealth services. Recommendations include increasing staffing during peak times, dynamic seasonal capacity and resource allocation, enhancing telehealth offerings, and implementing proactive patient engagement strategies to reduce missed appointments. Further research into the reasons for missed appointments, pandemic/seasonal illnesses and weather or other factors impacting on the service.



NHS (National Health Service) Data Analysis Report

Background/context of the business scenario:

- The primary objectives of this analysis were to evaluate staff adequacy and capacity, and to assess actual resource utilization across the NHS. Detailed business and analytical questions are provided in Appendix 1.
- Data provided: See Appendix 2. for details on the datasets provided. The appointment data covered a 30 month reporting period spanning from 01/2020 to 07/2022.
- Context: The time period was heavily influenced by COVID-19. It is important to note that the NHS functioned as Clinical Commissioning Groups during this period and the Integrated Care Boards were introduced at the end of this period in 07/2022. See additional details in Appendix 3.
- Problem analysis: The 5 Whys and SWOT problem-solving frameworks were used to gain deeper insights into the business problem, along with stakeholder mapping. See Appendix 4 for more details.

Analytical approach:

- The Jupyter notebook was organized into sections with a table of contents for easy navigation, improving workflow management.
- Key sections include setup, initial data checks, data cleaning, exploratory data analysis (EDA) and visualization, followed by answering individual questions and additional geographic analysis, predictive analytics and exploring additional relationships for COVID-19 and weather data. A PDF file was created to save EDA visualizations, visualisations for the business questions were saved as individual png files.
- Logging was used to execute multiple commands in a single block, reducing the notebook's length.
- While some code blocks are repeated, especially datetime functions, the focus was on integrating data cleaning, EDA, and visualization into a repeatable workflow.
- Appendix 5 – includes a detailed insights into the codes and rationales used during this analysis.
- Additional datasets, such as Twitter data, staffing levels, patient population, geographic data, weather, and COVID-19 data, were incorporated for broader insights. Although reliable sources were used, caution is advised in interpreting these data points. See Appendix 5 for more details.
- A limitation was the inability to merge all three main datasets due to memory constraints, leading to the analysis of datasets individually or in smaller merges. Given the different timeframes and daily/monthly data, merging would only provide insights for a smaller timeframe that all three files cover.
- The code of the jupyter notebook aimed to comply to the PEP 8 styling guide for python coding.
Comments were used throughout the notebook and markdown cells to display the results or separating sections. Line length compliance remains a limitation of the code.
- In some cases a styled output was coded to display the data in html or styled table rather than the pythonic output to improve user experience.



Visualisations and Insights:

- The Jupyter notebook has several visualisations. Not all visualisations were used to present to the stakeholders. Some visualizations were used primarily for analytical purposes, while others were tailored for stakeholder presentations.
- The visualizations for the stakeholder presentation have been carefully selected to address key business questions posed by the NHS, focusing on staff adequacy, resource utilization, and appointment patterns. The goal is to provide clear, actionable insights into these areas by visually representing the data in a way that is both intuitive and informative. Each visualization is chosen to highlight specific trends, patterns, and correlations that are critical to understanding the operational efficiency and challenges within the NHS's network of care providers. As the additional weather and COVID-19 data did not yield considerable insights, these were not included in the stakeholder presentation.
- The visualisation process often required several trials before the created visualisation looked presentable. One main issue was the overlapping text on the x axis. Often the axis had to be switched or the text rotated by 45-90 degrees to avoid overlapping. Appendix 6. Has further information on the rationale of the selection of the visualisations for business stakeholders. Several types of visualisations were used in the notebook, such as scatter plots, boxplots, bar and line charts and maps.
- Please refer to the jupyter notebook, the saved files and the presentation for the available visualisations. See appendix 6 for the rationale of the selected visualisations for the business stakeholders.

Service utilization insights:

- **Appointment patters:** Tuesdays were the busiest days, whilst there is a low appointment count at the weekends, suggesting that only few practices operate during the weekends. September to November and March are the busiest months, whilst April and August the quietest. (likely due to staffing levels during holiday periods). Spring and Winter were the most demanding seasons.
- **Capacity threshold:** The suggested 1.2M capacity threshold was not breached when all days of the month was included, however when weekend days were accounted for NHS functioned beyond this threshold the majority of times.
- **Appointment duration:** 42.2% of appointments were over 20 minutes long indicating time demands of staff
- **Unmapped appointments** for context types, national categories, service settings followed a similar declining pattern. Potential improvement in the recording of these appointments or using a different mapping of the appointments.
- **Regional analysis** highlighted differences between regions and ICB analysis highlighted differences between rural and urban ICBs. In terms of staffing South West had above England average staffing levels, whilst London had the lowest averages for all GP staffing groups. London also contributed to the highest number of missed appointments.
- **COVID-19 Impact:** The initial lockdown led to a shift from face-to-face to telephone appointments. Video/online modes remained low throughout the reporting period.



Patterns and predictions:

- **Trend analysis:** When trends were explored over the reporting period appointment numbers shown an increasing trend, however when it was limited only for the last year the trends were showing a stagnant overall appointment trend. Indicating the potential impact of COVID-19 restrictions and routine appointment handling.
- Average wait times were around 12 days, and both appointment length and wait times followed similar trends.
- **Predictive analytics:** SARIMA predictions were successful over the full period but not for the final year, highlighting limitations in shorter datasets. Only ARIMA method was included in the submitted files for the last year.

Recommendations:

- **Service Improvements:** Detailed recommendations for service enhancements and further data analysis are included in Appendix 7, emphasizing the need for improved data integration and predictive capabilities.



APPENDIX 1 – Business questions

The two main questions posed by the NHS are:

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

Sub-questions:

- What is the number of locations, service settings, context types, national categories, and appointment statuses in the data sets?
- What is the date range of the provided data sets, and which service settings reported the most appointments for a specific period?
- What is the number of appointments and records per month?
- What monthly and seasonal trends are evident, based on the number of appointments for service settings, context types, and national categories?
- What are the top trending hashtags (#) on the supplied Twitter data set and how can this be used in the decision-making process?
- Was there adequate staff and capacity in the networks?
- What was the actual utilisation of resources?
- What insights can be gained by looking at missed appointments?
- What are the most important patterns visible in the data relating to the use case?
- What insights can be gained from the data, and what recommendations can be made to the NHS based on these insights?

Refining Business Questions into Actionable Analytic Questions

1. Has there been adequate staff and capacity in the networks?

Analytic Questions:

- How many staff members are employed in each service setting?
- What is the staff-to-patient ratio in each location and service setting?
- How does staff availability correlate with appointment availability and missed appointments?

2. What was the actual utilization of resources?

Analytic Questions:

- What is the total number of appointments per service setting, location, and context type?
- What percentage of scheduled appointments were attended versus missed?
- How do utilization rates vary across different national categories and service settings?

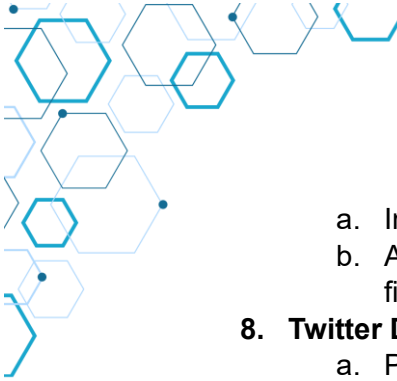


Additional Analytic Questions for Comprehensive Analysis

3. **Data Characteristics and Trends:**
 - What is the distribution of locations, service settings, context types, national categories, and appointment statuses in the data sets?
 - What is the date range of the provided data sets?
 - Which service settings reported the most appointments in the specified period?
4. **Appointment Patterns and Trends:**
 - What is the monthly and seasonal trend of appointments across different service settings and context types?
 - Are there any noticeable trends in appointment statuses over time?
5. **Insights from Missed Appointments:**
 - What are the most common reasons for missed appointments?
 - How do missed appointments vary by service setting, location, and time period?
 - What is the financial impact of missed appointments on the NHS?
6. **Twitter Data Insights:**
 - What are the top trending hashtags related to NHS services?
 - How can sentiment analysis of tweets inform decision-making and public communication strategies?

Exploratory Data Analysis (EDA) Plan

1. **Data Exploration:**
 - a. Load the provided data sets into Python.
 - b. Understand the structure, types, and completeness of the data.
 - c. Generate summary statistics for key variables.
2. **Data Wrangling:**
 - a. Clean the data to handle missing values, duplicates, and inconsistent formats.
 - b. Normalize and preprocess data for analysis, including date formatting and categorical encoding.
3. **Data Visualization:**
 - a. Create visualizations to understand the distribution of locations, service settings, context types, national categories, and appointment statuses.
 - b. Plot the date range and number of appointments over time to identify trends.
4. **Trend Analysis:**
 - a. Analyse the number of appointments and records per month to identify monthly and seasonal trends.
 - b. Visualize trends across different service settings, context types, and national categories.
5. **Utilization Analysis:**
 - a. Calculate utilization rates for resources in various settings.
 - b. Compare attended versus missed appointments across different variables
6. **Staff and Capacity Analysis:**
 - a. Analyze staff availability and its correlation with appointment availability and missed appointments.
 - b. Visualize staff-to-patient ratios and their impact on service delivery.
7. **Missed Appointments Analysis:**

- 
- a. Investigate the reasons for missed appointments.
 - b. Analyze the impact of missed appointments on resource utilization and financial costs.

8. Twitter Data Analysis:

- a. Perform text analysis to identify trending hashtags and topics.
- b. Conduct sentiment analysis to gauge public opinion and concerns.

Diagnostic Analysis Plan

1. Advanced Statistical Analysis:

- a. **Correlation Analysis:** Identify and quantify the relationships between different variables, such as how staff levels correlate with missed appointment rates.
- b. **Regression Analysis:** Explore how predictors such as time of year, service settings, or patient demographics impact appointment outcomes. This can help to model and predict behaviors based on input variables.
- c. **Factor Analysis:** Reduce the number of observed variables into a few interpretable underlying factors (e.g., grouping different types of appointments into broader categories based on underlying patterns) – This was not performed.

2. Root Cause Analysis: (Due to unavailability of datapoints this was not performed)

- a. **Pareto Analysis:** Apply the Pareto principle to identify the 'vital few' root causes that lead to the majority of missed appointments or other issues.
- b. **Cause and Effect Diagrams:** Also known as fishbone diagrams, these can help systematically explore potential causes of a specific problem, like missed appointments – This would be for further analysis.

3. Hypothesis Testing: (Touched on this but was not within the main scope of the analysis)

- a. Develop hypotheses based on observed data trends and test these using appropriate statistical tests (e.g., t-tests, ANOVA) to validate or reject these hypotheses. This could involve testing if differences in no-show rates between different regions are statistically significant.

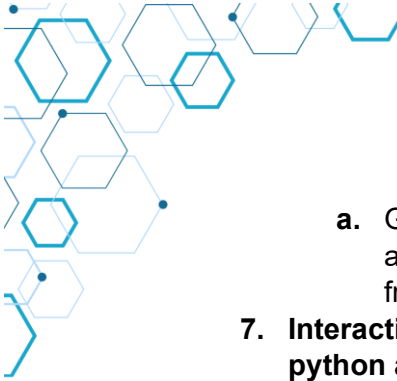
4. Segmentation Analysis: (This would be suggested for further research and analysis)

- a. **Cluster Analysis:** Use unsupervised learning techniques to segment patients or appointments into groups with similar characteristics. This can uncover hidden patterns and subgroup behaviours in appointment attendance.
- b. **Decision Trees:** These can help in understanding the decision paths that might lead to different outcomes such as missed appointments versus attended ones.

5. Time Series Analysis: (Performed only on regional data – potential further research into a more granular level of analysis – again not within the scope of this analysis)

- a. **Seasonal Decomposition:** Explore how different components (trend, seasonality, and noise) contribute to appointment patterns over time.
- b. **Forecasting Models:** Implement ARIMA, SARIMA, or other forecasting methods to predict future trends in appointment data.

6. Enhanced text and Sentiment Analysis:



- a. Go beyond identifying trending hashtags to perform deeper sentiment analysis and natural language processing to extract themes and sentiments from patient feedback or social media that can explain public perceptions.
- 7. Interactive Dashboards and Reporting: (This current analysis focused on python analysis. A dashboard can be created based on stakeholders request)**
 - a. Develop or enhance dashboards to include interactive elements such as drill-downs and sliders that allow users to explore data dynamically.
 - b. Integrate diagnostic tools directly into dashboards, such as statistical summaries by hover or click, and real-time filtering capabilities to explore different hypotheses or scenarios.
- 8. Machine Learning for Predictive Analytics: (Suggestion for further analysis)**
 - a. Use machine learning models to predict outcomes based on historical data. For instance, logistic regression or random forests could predict the likelihood of appointment no-shows based on various factors.
- 9. Feedback Loop Integration: (Suggestion for further analysis)**
 - a. Incorporate mechanisms to collect ongoing feedback on the analytics provided, using this to continually refine models, hypotheses, and understandings

APPENDIX 2 – Datasets provided

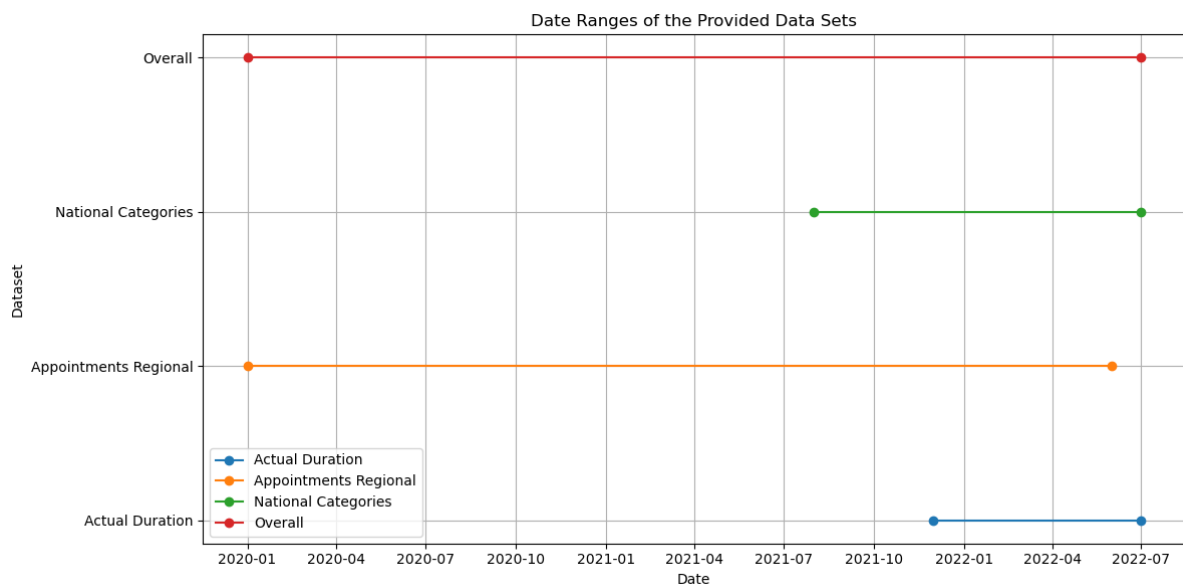
5 files provided for the analysis:

- metadata_nhs.txt
- actual_duration.csv – daily data
- appointments_regional.csv – monthly aggregated data
- national_categories.xlsx – daily data
- tweets.csv

Key information from the metadata:

- See Jupyter Notebook for brief overview of the metadata

Date ranges the datasets cover:

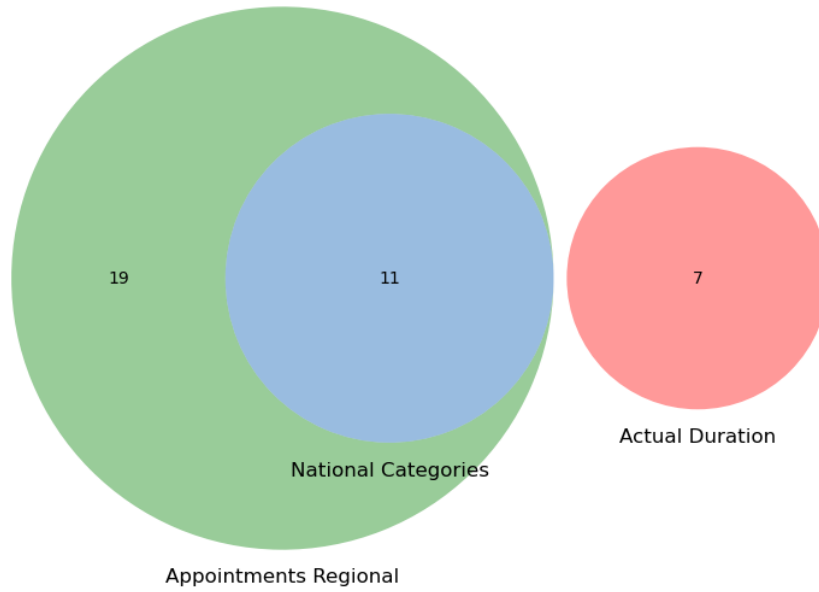


The national categories and the appointments regional files have the same appointment counts at a monthly level, whilst the actual duration file has 8-10% lower counts. This could be due to potential inconsistencies in data reporting/recording.

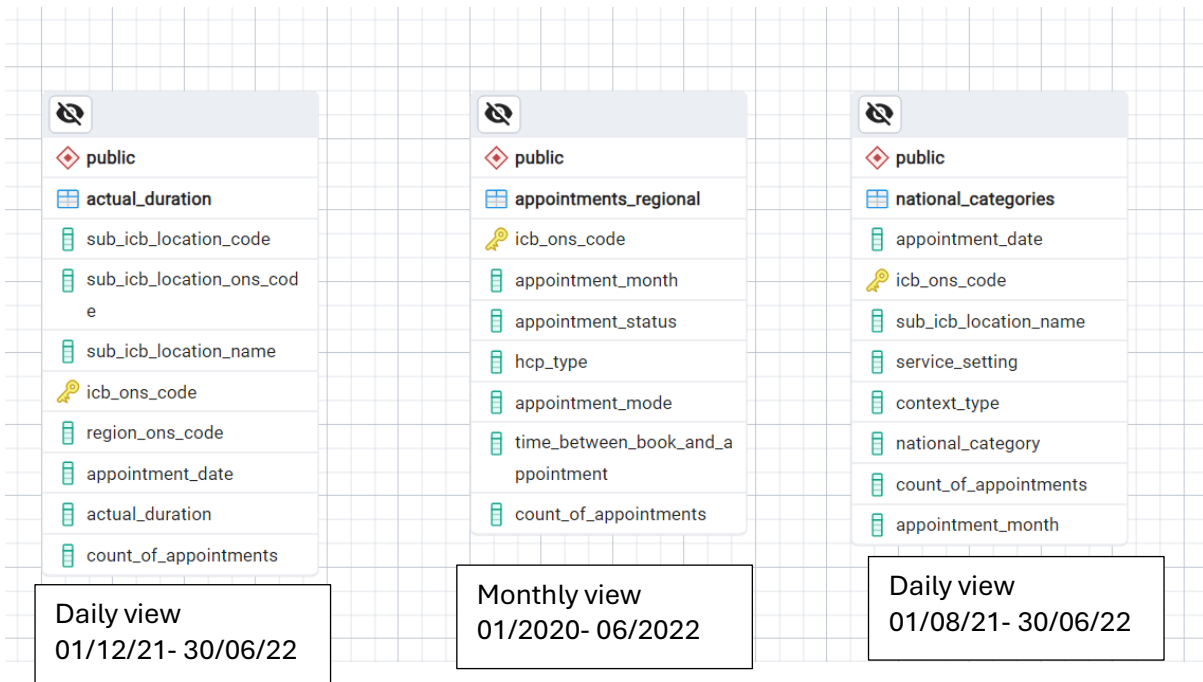
The venn diagram below shows this



Venn Diagram of the Three Datasets




An ERD (Entity Relationship Diagram) was created to visualise the three datasets and the columns that were provided:



Initial insights from studying the metadata, files and the scenario

Challenges

- 
1. **Data Quality and Consistency:** Variations in data quality due to differences in how practices manage appointments. Inconsistent recording of appointment modes and healthcare professional types. Issues with capturing DNA (Did Not Attend) appointments correctly, especially in certain periods and systems.
 2. **Missed Appointments:** High rates of missed appointments lead to inefficient resource utilization and increased costs. Understanding the reasons behind missed appointments is crucial for mitigating this issue.
 3. **Resource Allocation:** Determining the adequacy of staff and capacity across different regions and service settings. Balancing between expanding capacity and optimizing current resources.
 4. **Technological Limitations:** Outdated appointment management systems lacking advanced features like predictive analytics and automated reminders. Incomplete integration with external data sources like social media for more comprehensive insights.

Limitations

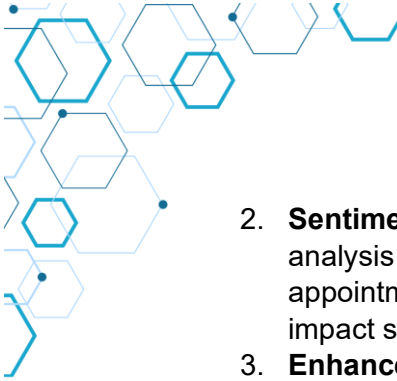
1. **Data Granularity:** The data is aggregated at the Sub-ICB location level, which might obscure finer details needed for specific analyses.
2. **Incomplete Data Coverage:** Not all practices in England are included, leading to estimates that might not fully capture the total picture. Missing or null values for certain data fields, such as actual duration and HCP types, limit the accuracy of analyses.
3. **Mode of Appointment:** Misclassification or inconsistent recording of the mode of appointments (e.g., face-to-face vs. video consultations) can affect the accuracy of utilization analyses.
4. **Context and Category Mapping:** Inconsistent or unmapped context types and national categories can lead to misinterpretation of data.

Key Areas to Focus On

1. **Appointment Trends:** Analyse monthly and seasonal trends in appointments across different service settings, context types, and national categories. Identify peaks and troughs in appointment numbers to better allocate resources.
2. **Missed Appointments:** Deep dive into the reasons for missed appointments and identify patterns that can inform strategies to reduce no-shows. Potentially analyse the financial impact of missed appointments and potential savings from reducing them.
3. **Resource Utilization:** Evaluate the actual utilization of resources (e.g., staff, appointment slots) to identify underutilized areas. Compare attended versus missed appointments to assess the efficiency of current resource allocation.
4. **Staff and Capacity Analysis:** If possible, assess staff-to-patient ratios and correlate with appointment availability and utilization. Analyse the adequacy of staff across different regions and service settings.

Potential Expansions for Additional Insights

1. **Predictive Analytics:** Develop predictive models to forecast appointment demand and identify factors contributing to missed appointments. Use machine learning techniques to predict high-risk no-show appointments and implement targeted interventions.

- 
2. **Sentiment Analysis from Social Media:** Leverage Twitter data to perform sentiment analysis and gain insights into public perceptions and potential barriers to attending appointments. Identify trending health-related issues and public concerns that could impact service utilization.
 3. **Enhanced Patient Communication:** Explore the effectiveness of different communication channels (e.g., SMS, email, phone calls) in reducing missed appointments. Implement and evaluate automated reminder systems and their impact on appointment attendance.
 4. **Integration with External Data Sources:** Integrate weather data, local events, and transportation information to understand external factors affecting appointment attendance. Use socio-economic data to identify vulnerable populations and tailor interventions to their specific needs.
 5. **Detailed Geographic Analysis:** Conduct a more granular geographic analysis to identify specific areas with high missed appointment rates and resource constraints. Implement localized strategies to address the unique challenges of different regions.



APPENDIX 3 – NHS context and background information

COVID-19 Pandemic


- **January 2020 - July 2022:** The entire period was heavily influenced by the COVID-19 pandemic. The NHS faced unprecedented demand, with resources diverted to manage the crisis, including hospitalizations, vaccinations, and routine testing. (<https://www.gov.uk/coronavirus>, <https://www.nhs.uk/conditions/coronavirus-covid-19/>)
- **March 2020:** The UK went into its first lockdown, significantly impacting routine NHS services as resources were reallocated to cope with the influx of COVID-19 patients (<https://www.bbc.com/news/uk-51506729>)
- **December 2020:** The UK began its vaccination campaign against COVID-19, starting with the most vulnerable groups (<https://www.bbc.com/news/health-55274833>)
- **Omicron Variant Surge:** The emergence of the Omicron variant in late 2021 and its rapid spread in December 2021 and January 2022 led to a surge in cases, impacting the NHS heavily with increased hospitalizations and staff shortages. (<http://bbc.com/news/health-omicron>, <http://gov.uk/omicron>)
- **November 2021:** Winter pressures, combined with the ongoing impact of COVID-19 and other seasonal illnesses, led to increased demand for NHS services. This period saw heightened concerns about the capacity to handle the dual pressures of the pandemic and regular winter illnesses. (<https://www.theguardian.com/society/2021/nov/25/nhs-braced-for-toughest-winter-ever-as-staff-warn-of-burnout>, <https://www.independent.co.uk/news/health/nhs-winter-crisis-staff-burnout-b1944537.html>)
- **March 2022:** The UK government began to phase out COVID-19 restrictions, marking a transition towards living with the virus. This period saw a shift in NHS focus towards catching up on delayed elective procedures and routine care that had been postponed due to the pandemic (<https://www.theguardian.com/world/2022/feb/24/end-england-covid-restrictions-measures-to-stay>, <https://www.gov.uk/government/news/prime-minister-sets-out-plan-for-living-with-covid>, <https://www.bbc.com/news/health-60662037>)

NHS Structure Change

- **July 2022 – Integrated Care Board Establishment Order –** The NHS to switch to the new ICB structure. Important to note that the data provided has the ICB structure and reporting. But the actual functioning of the NHS was still based on the previous system of using clinical commissioning groups. (<https://www.england.nhs.uk/wp-content/uploads/2022/05/B1770-integrated-care-boards-establishment-order-2022.pdf>)

Other Background information

- **NHS Staff Shortages -** Throughout this period, the NHS also faced significant staff shortages, exacerbated by the pandemic. This affected the ability to meet patient demand and maintain routine services.
- **Financial and Policy Changes -** There were also notable changes in policy and funding, with the UK government injecting additional funds into the NHS to cope with the increased demand and to support recovery efforts post-COVID.

- 
- **Post-Brexit issues:** The UK's exit from the European Union had various impacts on the NHS, including potential staff shortages due to changes in immigration policies and supply chain issues for medical supplies. (<http://gov.uk/brexit> , <http://nhsconfed.org/brexit>)
 - **Seasonal Flu and Other Illnesses:** Each winter, the NHS faces increased demand due to seasonal flu and other respiratory illnesses, adding to the healthcare system's strain. (<http://gov.uk/seasonal-flu>, <http://bbc.com/news/flu>)

ICB Information:

The following links provide additional information on the new NHS structure:

There are 106 Sub-ICB locations, 42 ICB locations and 7 NHS Regions in this new structure. This replaced the existing Clinical Commissioning Groups.

<https://digital.nhs.uk/data-and-information/publications/statistical/patients-registered-at-a-gp-practice/august-2022>

<https://cms.nhsbsa.nhs.uk/sicbls-icbs-and-other-providers/organisation-and-prescriber-changes/icbs>



APPENDIX 4 – Pre-analysis problem mapping

Stakeholder Analysis

The key stakeholders of an Integrated Care Board (ICB) are diverse and encompass a broad range of organizations and individuals who play crucial roles in the planning, commissioning, and delivery of health and care services.

1. NHS Providers:

Hospitals and NHS Trusts: Deliver secondary and tertiary care services, including specialized treatments.

Community Health Services: Provide essential care outside of hospital settings, such as district nursing and rehabilitation services.

Mental Health Services: Offer support and treatment for mental health conditions, including inpatient and community-based services.

2. Primary Care Providers:

General Practitioners (GPs): Primary point of contact for most patients, providing comprehensive healthcare services.

Pharmacies: Supply medications and offer health advice and services like vaccinations.

Dentists and Optometrists: Provide specialized primary care services in oral health and vision care.

3. Local Authorities:

Public Health Departments: Focus on preventive health measures, health promotion, and addressing the wider determinants of health.

Social Services: Provide social care services, support for vulnerable populations, and safeguarding services.

4. Voluntary and Community Sector:

Charities and Non-Profit Organizations: Offer a wide range of support services, advocacy, and specialized care, often targeting specific health conditions or vulnerable groups.


Community Groups: Engage local communities in health initiatives, provide grassroots support, and facilitate public involvement in health planning.

5. Patients and the Public:

Patient Representatives: Include individuals and groups that represent the views and interests of patients, ensuring their voices are heard in decision-making processes.

Public Engagement: Mechanisms for involving the wider public in consultations and feedback on health services.

6. Integrated Care Partnerships (ICPs):



Collaborate with ICBs to align health and care services across a region, focusing on integrated and coordinated care delivery.

7. Clinical Leaders:

Medical Directors and Clinical Leads: Provide clinical expertise and leadership in the planning and commissioning of services.

Nursing and Allied Health Professionals: Contribute to service design and implementation, ensuring high standards of care.

8. Health and Wellbeing Boards:

Facilitate collaboration between health and social care leaders, local authorities, and other stakeholders to improve health and wellbeing outcomes in the local area.

9. Regulatory and Oversight Bodies:

Care Quality Commission (CQC): Regulates and inspects health and social care services to ensure they meet quality and safety standards.

NHS England: Provides oversight, support, and strategic direction for ICBs.

10. Academic and Research Institutions:

Universities and Research Centers: Contribute to evidence-based practice, innovation, and the evaluation of health interventions.

11. Private Sector Partners:

Private Healthcare Providers: May be involved in delivering certain health services under contract with the NHS.

Technology and Pharmaceutical Companies: Provide essential products and services, including medical devices, health IT systems, and medications.

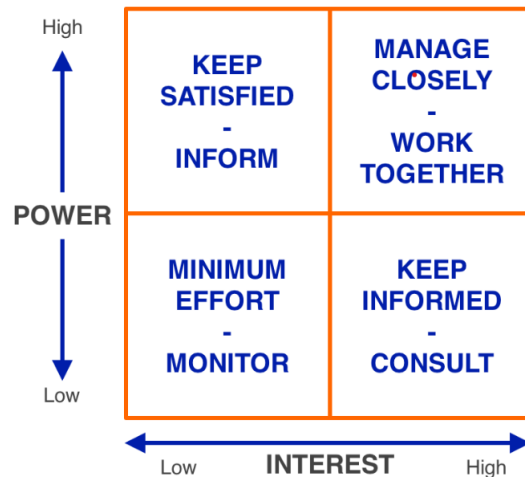
12. Health and Social Care Workforce:

Includes all professionals and staff involved in the delivery of health and care services, whose input and engagement are critical to the successful functioning of the ICB.

13. Commissioning Support Units (CSUs):

Provide business intelligence, procurement, and other support services to ICBs to aid in effective commissioning.

These stakeholders play pivotal roles in the operation of ICBs, contributing to the overarching goal of providing integrated, high-quality, and patient-centered care across the healthcare system. Their collaboration and engagement are essential for the success of ICBs in meeting the health and care needs of their populations.



Picture –
mapping

The 'Standard' Stakeholder Map

Reproduced from The Influence Agenda
by Dr Mike Clayton (Palgrave Macmillan)

Stakeholder

<https://i0.wp.com/onlinepmcourses.com/wp-content/uploads/2017/07/Standard-Stakeholder-Map.png?ssl=1>

Based on the stakeholder mapping as per the picture above the key stakeholders were grouped in the following categories:

Stakeholder Grouping:

1. Manage Closely - Work Together:

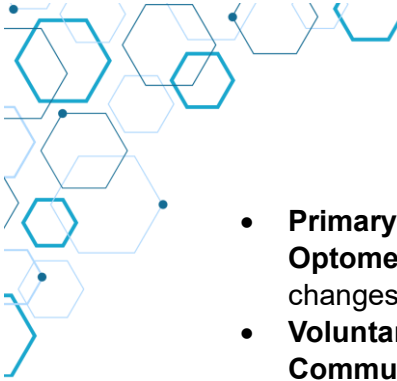
- **ICB Board Members:** High power, high interest. Engage them actively in strategic planning, decision-making processes, and regular updates to ensure their insights and authority guide the projects effectively.
- **NHS Providers (Hospitals and NHS Trusts, Community Health Services, Mental Health Services):** High power, high interest. Collaborate closely with them as they are crucial for service delivery and strategic planning.

2. Keep Satisfied - Inform:

- **Local Authorities (Public Health Departments, Social Services):** High power, lower interest. Keep them informed and involved in key decisions to ensure alignment and cooperation.
- **Health and Wellbeing Boards:** High power, lower interest. Keep them satisfied by informing them about significant developments and ensuring their strategic priorities are considered.

3. Keep Informed - Consult:

- **Technical Staff (Program Developers and Data Analysts):** Lower power, high interest. Maintain open communication and consult them regularly for technical input.

- 
- **Primary Care Providers (General Practitioners, Pharmacies, Dentists, Optometrists):** Lower power, high interest. Keep them informed about relevant changes and consult them to ensure primary care integration.
 - **Voluntary and Community Sector (Charities, Non-Profit Organizations, Community Groups):** Lower power, high interest. Engage them through consultations and keep them informed to leverage their community connections and support.

4. Minimum Effort - Monitor:

- **Patients and the Public:** Lower power, lower interest. Monitor their needs and feedback to ensure patient-centered care but with less frequent engagement.
- **Academic and Research Institutions (Universities and Research Centers):** Lower power, lower interest. Monitor their research outputs and collaborate as needed for evidence-based practices.
- **Private Sector Partners (Private Healthcare Providers, Technology and Pharmaceutical Companies):** Lower power, lower interest. Monitor their contributions and involve them as needed for specialized services.
- **Health and Social Care Workforce:** Lower power, lower interest. Monitor workforce satisfaction and engagement to ensure smooth service delivery.

5 Whys Analysis of the problem

Based on Sebastian Traeger - Root Cause Analysis with 5 Whys Technique

Main Problem Statement

The National Health Service (NHS) faces significant challenges in managing its capacity and resources due to increasing population demands and a high number of missed appointments. To ensure efficient service delivery and financial sustainability, the NHS must optimize resource utilization, understand appointment trends, and improve patient attendance using a data-driven approach, incorporating both internal and external data sources.

5 Why's Analysis

Problem: High number of missed appointments leading to inefficient resource utilization and increased costs for the NHS.

Why 1: Why are there a high number of missed appointments?

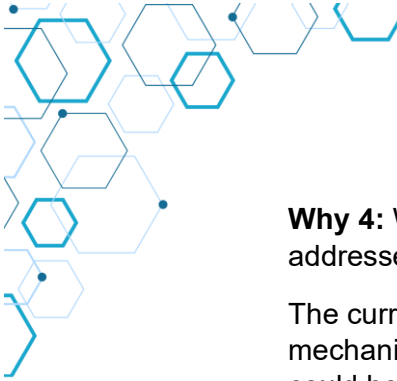
Many patients do not attend their scheduled appointments.

Why 2: Why do patients not attend their scheduled appointments?

There could be various reasons, including forgetfulness, transportation issues, long waiting times, or lack of awareness about the appointment.

Why 3: Why do patients forget or face issues attending their appointments?

Patients may not receive timely reminders or face barriers such as poor communication, inconvenient appointment times, or lack of accessible transportation.



Why 4: Why are timely reminders and convenient appointments not adequately addressed?

The current appointment management system may lack robust reminder mechanisms, flexibility in scheduling, and integration with external data sources that could help predict and mitigate missed appointments.

Why 5: Why does the appointment management system lack these features?

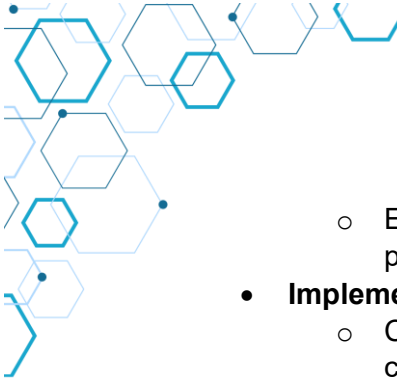
There might be insufficient investment in technology, data analytics, and patient engagement strategies to enhance the appointment management system and reduce missed appointments.

Summary of Root Cause

The root cause of the high number of missed appointments is the inadequate appointment management system, which lacks robust reminder mechanisms, flexibility in scheduling, and integration with external data sources. This is compounded by insufficient investment in technology, data analytics, and patient engagement strategies.

Recommendations to Address the Root Cause

- **Enhance Appointment Management System:**
 - Invest in a modernized appointment management system with features such as automated reminders via SMS, email, and phone calls.
 - Implement flexible scheduling options to accommodate patient preferences and availability.
- **Utilize Predictive Analytics:**
 - Develop predictive models using historical appointment data to identify high-risk no-show appointments.
 - Implement targeted interventions for high-risk patients, such as additional reminders and follow-up calls.
- **Improve Patient Communication:**
 - Establish a multi-channel communication strategy to ensure patients receive timely and clear information about their appointments.
 - Provide clear instructions on how to cancel or reschedule appointments easily.
- **Leverage External Data Sources:**
 - Integrate external data sources, such as social media and transportation information, to gain insights into factors affecting appointment attendance.
 - Use sentiment analysis from social media to understand patient concerns and address potential barriers to attending appointments.
- **Increase Patient Engagement:**
 - Educate patients about the importance of attending appointments through community outreach programs and educational materials.
 - Provide incentives for patients to attend their appointments, such as loyalty programs or small rewards.
- **Optimize Resource Allocation:**
 - Analyze appointment trends to identify peak times and allocate resources accordingly to reduce waiting times and improve patient satisfaction.

- 
- Ensure adequate staffing levels during high-demand periods to manage patient flow efficiently.
 - **Implement Feedback Mechanisms:**
 - Collect patient feedback on appointment scheduling and reminders to continuously improve the system.
 - Use feedback to identify areas for improvement and implement changes promptly.
 - By addressing these recommendations, the NHS can reduce the number of missed appointments, optimize resource utilization, and enhance overall patient satisfaction and service delivery.

SWOT Analysis of the scenario

Based on MindTools SWOT Analysis the following was learnt

Strengths

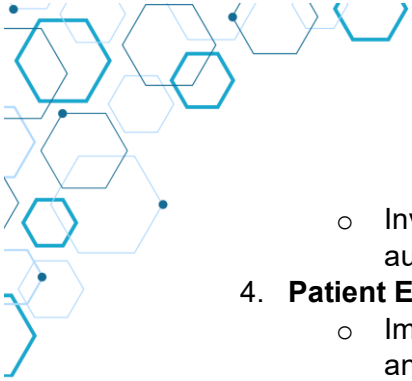
1. **Established Infrastructure:**
 - The NHS has a well-established healthcare infrastructure and a broad network of service providers.
2. **Experienced Workforce:**
 - Skilled and dedicated healthcare professionals and staff.
3. **Public Trust:**
 - High level of public trust and recognition as a leading healthcare provider.
4. **Comprehensive Data:**
 - Access to extensive internal healthcare data that can be leveraged for in-depth analysis and decision-making.

Weaknesses

1. **Missed Appointments:**
 - High rates of missed appointments leading to inefficient use of resources and increased costs.
2. **Resource Constraints:**
 - Limited resources and capacity to handle increasing population demands effectively.
3. **Technological Limitations:**
 - Outdated appointment management systems lacking advanced features like predictive analytics and robust reminder mechanisms.
4. **Communication Gaps:**
 - Inefficient communication with patients regarding appointments, leading to higher no-show rates.

Opportunities

1. **Data-Driven Decision Making:**
 - Utilizing advanced data analytics to better understand and predict healthcare service utilization and patient behaviours
2. **External Data Integration:**
 - Leveraging external data sources like social media to gain insights into patient sentiments and public health trends.
3. **Technology Enhancement:**



- Investing in modernizing appointment management systems with features like automated reminders, flexible scheduling, and telehealth options.
- 4. **Patient Engagement:**
 - Improving patient engagement and education to reduce missed appointments and enhance overall health outcomes.

Threats

1. **Increasing Demand:**
 - Growing population and increased demand for healthcare services may strain existing resources further.
2. **Financial Pressures:**
 - Budget constraints and financial pressures could limit the ability to invest in necessary technological upgrades and innovations.
3. **Competition:**
 - Competition from private healthcare providers offering more flexible and technologically advanced services.
4. **Public Health Crises:**
 - Unpredictable public health crises (e.g., pandemics) could disrupt service delivery and strain resources.

Strategic Recommendations

Based on the SWOT analysis, here are some strategic recommendations:

- **Enhance Appointment Management:** Invest in advanced appointment management systems with predictive analytics, automated reminders, and flexible scheduling options to reduce missed appointments.
- **Leverage Data Analytics:** Utilize internal and external data sources to gain insights into service utilization trends and patient behaviours, aiding in better resource allocation and decision-making.
- **Improve Patient Communication:** Develop comprehensive patient communication strategies, including multi-channel reminders and educational campaigns to improve appointment adherence.
- **Expand Capacity Strategically:** Focus on strategic expansion of capacity in high-demand areas while optimizing existing resources to meet growing population needs effectively.
- **Foster Public-Private Partnerships:** Explore partnerships with private healthcare providers and technology firms to enhance service delivery and technological capabilities.

APPENDIX 5 – Data Analysis steps

1. Library Imports:

As the analysis progressed several additional imports were needed, these were added to the beginning of the notebook. Installation of some of the libraries may be required before the codes can be executed, using the pip install method within the anaconda prompt environment.

Imports

```
|: # Standard Library imports:
import os
import sys
import logging
import subprocess
import warnings
import re

# Third-party Library imports:
# Note you may need to install these prior to the import in the Anaconda prompt:
# pip install pandas numpy matplotlib seaborn ipython geopandas folium /
# contextily requests pillow matplotlib-venn textblob
# pip install spacy geopy
# python -m spacy download en_core_web_sm
from datetime import datetime
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib_venn import venn3
import seaborn as sns
from matplotlib.backends.backend_pdf import PdfPages
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from statsmodels.tsa.seasonal import STL
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from scipy.stats import ttest_ind, f_oneway
from IPython.display import display, Markdown, HTML
import geopandas as gpd
import folium
import contextily as ctx
import requests
from PIL import Image
from io import BytesIO
from textblob import TextBlob
import spacy
from geopy.geocoders import Nominatim

# Local application/library-specific imports:
```

- Setting up a configuration section** - In order to access the notebook on a different computer data paths will need to be adjusted. Warning messages were suppressed for a better user experience, however certain code block may be deprecated in future versions of python and will need to be amended if used in the future.

Configuration settings and constants

```
: # To print your directory
current_directory = os.getcwd()
print("Current Directory:", current_directory)

Current Directory: C:\Users\beata\OneDrive\Beata\DA Career Accelerator\Assignment 2 -python\assignment_final

: # Configuration settings and constants

# Suppress FutureWarning
warnings.filterwarnings("ignore")

# Suppress informational messages
logging.getLogger('matplotlib').setLevel(logging.WARNING)

# Insert your own directory where data files are stored. Use \\ sign to separate in the path.
DATA_PATH = 'OneDrive\Beata\DA Career Accelerator\Assignment 2 -python\assignment_final'
# Directory path where output files will be saved
OUTPUT_PATH = DATA_PATH
# Ensure the directories exist
LOG_FILE = os.path.join(OUTPUT_PATH, 'analysis.log') # Log file path

os.makedirs(DATA_PATH, exist_ok=True)
os.makedirs(OUTPUT_PATH, exist_ok=True)

# Set up Logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s', handlers=[
    logging.FileHandler(LOG_FILE),
    logging.StreamHandler()
])

# Version information
__version__ = "1.0.0"

# Print version
print(f"Current version: {__version__}")

print("Configuration settings:")
print(f"DATA_PATH: {DATA_PATH}")
print(f"OUTPUT_PATH: {OUTPUT_PATH}")
print(f"Output path exists: {os.path.exists(OUTPUT_PATH)}")
print(f"Output path is writable: {os.access(OUTPUT_PATH, os.W_OK)}")
print(f"LOG_FILE: {LOG_FILE}")
```

3. Functions to load and save data - Individual visualisations were saved using `plt.savefig('...png')` method, however were multiple visualisations were created with a block of code the `save_plot` function was used. The EDA visualisations were saved into a pdf file. Excel files

Functions to load and save data

```
# Define a function to Load data
def load_data(file_path, file_type='csv'):
    try:
        if file_type == 'csv':
            df = pd.read_csv(file_path)
        elif file_type == 'excel':
            df = pd.read_excel(file_path)
        logging.info(f"Data loaded successfully from {file_path}")
        return df
    except Exception as e:
        logging.error(f"Error loading data from {file_path}: {str(e)}")
        return None

# Function to save results
def save_results(df, file_path):
    if df is None:
        logging.error(f"Cannot save results to {file_path}: DataFrame is None")
        return
    try:
        logging.info(f'Saving results to {file_path}')
        df.to_csv(file_path, index=False)
        logging.info(f'Successfully saved results to {file_path}')
    except Exception as e:
        logging.error(f"Error saving file to {file_path}: {str(e)}")
        raise

# Optional, to save it to pickle for faster file handling
# def save_to_pickle(df, file_path):
#     logging.info(f'Saving DataFrame to {file_path}')
#     df.to_pickle(file_path)

# def load_from_pickle(file_path):
#     logging.info(f'Loading DataFrame from {file_path}')
#     return pd.read_pickle(file_path)

# Utility function to save plots
def save_plot(filename):
    plt.savefig(filename, bbox_inches='tight')
    print(f"Plot saved to {filename}")
```

Sense Checking Raw Data

```
# Function to sense check data
def sense_check_data(df, file_name):
    try:
        logging.info(f"Sense checking data for {file_name}")
        logging.info(f"\n{file_name} - Data Info:\n")
        df.info()
        logging.info(f"\n{file_name} - Data Shape: {df.shape}")
        # Logging.info(f"\n{file_name} - Data Description:\n{df.describe(include='all')}")
        # If you would like to include the statistical description at this stage too.

        display(df.head())
        display(df.tail())

        # Plot Histograms for numerical columns
        numerical_cols = df.select_dtypes(include=[np.number]).columns.tolist()
        if numerical_cols:
            df[numerical_cols].hist(figsize=(10, 10))
            plt.suptitle(f"Histograms for {file_name} - Numerical Columns")
            plt.show()

        # Plot count plots for non-numerical columns
        categorical_cols = df.select_dtypes(include=['object', 'category']).columns.tolist()
        if categorical_cols:
            for col in categorical_cols:
                # Get top 10 and bottom 10 unique values
                value_counts = df[col].value_counts()
                if len(value_counts) > 20:
                    top_10 = value_counts.nlargest(10)
                    bottom_10 = value_counts.nsmallest(10).sort_values(ascending=False)
                    value_counts = pd.concat([top_10, bottom_10])

                plt.figure(figsize=(10, 6))
                sns.countplot(y=df[col], order=value_counts.index)
                plt.title(f"Count Plot of {col} in {file_name}")
                plt.xlabel('Count')
                plt.ylabel(col)

                # Add subtitle below the title
                plt.text(0.5, -0.15, "Note: For larger sets, the top and bottom 10 is displayed",
                        ha='center', va='center', transform=plt.gca().transAxes, fontsize=10, style='italic')

                plt.tight_layout()
                plt.show()

            # Plot all columns
            df.plot(figsize=(10, 6))
            plt.title(f"Plot of all columns in {file_name}")
            plt.tight_layout()
            plt.show()

        except Exception as e:
            logging.error(f"Error during sense check for {file_name}: {str(e)}")

# Main workflow
def main():
    file_names = ['actual_duration.csv', 'appointments_regional.csv', 'national_categories.xlsx']

    for file_name in file_names:
        file_type = 'excel' if file_name.endswith('.xlsx') else 'csv'
        df = load_data(file_name, file_type=file_type)
        if df is not None:
            sense_check_data(df, file_name)

if __name__ == "__main__":
    main()
```

The above section allows sense checking the provided files. Basic visualisations are provided to see the data in a visual way, as there are lots of categorical values in certain fields, the visualisations were limited to the top and bottom 10.

4. Data cleaning - When cleaning data, several dimensions were considered to ensure high-quality data. These dimensions include data accuracy, completeness, validity, timeliness, uniqueness, and consistency. This methodology derives from best practices in data quality management and governance frameworks.
Dimensions of Data Quality:



- **Data Accuracy:** Ensures that the data correctly represents the real-world entities it is supposed to model. Methods: Validation checks, comparing with trusted sources, correcting inaccuracies.
- **Data Completeness:** Ensures that all required data is present and no critical data is missing. Methods: Checking for missing values, filling in gaps, ensuring mandatory fields are populated.
- **Data Validity:** Ensures that data conforms to the defined formats and standards. Methods: Using regex patterns, data type checks, and range validations.
- **Data Timeliness:** Ensures that data is up-to-date and available when needed. Methods: Timestamp checks, ensuring data updates are within acceptable timeframes.
- **Data Uniqueness:** Ensures that each record is unique and there are no duplicate entries. Methods: Removing duplicates, enforcing unique constraints.
- **Data Consistency:** Ensures that data is consistent across different systems and datasets. Methods: Cross-validation between systems, ensuring referential integrity.
- **References:** ISO 8000-1:2022, DAMADMBOK, Gartner Data Quality Framework

The data cleaning workflow included steps to identify missing values, format the date columns, manage the sub_icb_location_name columns to create a new column with the ICB names only, adding a duplicate flag column to identify potential duplicates within the data. Creating numerical columns for the categorical columns of actual_duration and time_between_book_and_appointment in preparation for further analysis. To load the .xlsx file it takes a long time, however when tried to manually change the file into a csv in excel, some data points were lost. Therefore left the xlsx file were used for the sense checking and data cleaning steps. The cleaned files were changed to a new csv file for further use.

```

# Data Cleaning Functions
# Function to remove blank spaces in the data
def clean_data(df, file_name):
    try:
        logging.info(f'\nCleaning data in {file_name}')
        logging.info(f'Shape before cleaning: {df.shape}')
        df.columns = df.columns.str.strip()
        df = df.apply(lambda x: x.str.strip() if x.dtype == "object" else x)
        logging.info(f'Shape after cleaning: {df.shape}')
        return df
    except Exception as e:
        logging.error(f'Error cleaning data for {file_name}: {str(e)}')
        return df

# Function to check missing values
def check_missing_values(df, file_name):
    logging.info(f'\nChecking for missing values in {file_name}')
    missing_values = df.isnull().sum()
    logging.info(f'\nMissing values in {file_name}: \n{missing_values}\n')
    return missing_values

# Function to correct date and format appointment_date columns
def correct_date_types(df, date_columns, file_name):
    logging.info(f'\nCorrecting date types in {file_name} for columns: {date_columns}')
    for column in date_columns:
        original_non_null = df[column].notnull().sum()
        df[column] = pd.to_datetime(df[column], errors='coerce')
        converted_non_null = df[column].notnull().sum()
        logging.info(f'{column} - Converted {converted_non_null} out of {original_non_null} non-null values')
        logging.info(f'Data types after conversion in {file_name}: \n{df.dtypes}\n')
    # If 'appointment_date' column is in the date_columns, create 'appointment_month' column
    if 'appointment_date' in date_columns:
        df['appointment_month'] = df['appointment_date'].dt.to_period('M').dt.to_timestamp()
        logging.info(f'Created appointment_month column in {file_name}')
    return df

# Function to correct and format appointment_month column
def format_appointment_month(df, column, file_name):
    logging.info(f'\nFormatting {column} in {file_name}')
    original_non_null = df[column].notnull().sum()
    df[column] = pd.to_datetime(df[column], errors='coerce').dt.to_period('M')
    converted_non_null = df[column].notnull().sum()
    logging.info(f'{column} - Converted {converted_non_null} out of {original_non_null} non-null values')
    logging.info(f'Formatted {column} in {file_name}: \n{df[column].head()}\n')
    return df

# Function to create an icb_location_name columns in ad
def create_new_icb_location_name(df, column_name, file_name):
    logging.info(f'\nCreating new icb_location_name column from {column_name} in {file_name}')
    # Split the string based on 'ICB - ' and retain the full ICB name
    df['icb_location_name'] = df[column_name].str.split('ICB - ').str[0] + 'ICB'
    return df
    # Reorder columns to place the new columns next to each other
    new_column_order = ['icb_ons_code', 'sub_icb_location_ons_code', 'sub_icb_location_name',
                        'icb_location_name', 'region_ons_code', 'appointment_date',
                        'appointment_month', 'actual_duration',
                        'count_of_appointments']
    df = df[new_column_order]
    return df

# Function to create a new icb_location_name and sub_icb_location code columns in nc
def split_code_after_hyphen_and_reorder_columns(df, column_name, file_name):
    logging.info(f'\nSplitting code after hyphen in {column_name} in {file_name}')
    # Split the string based on 'ICB - ' and retain the full ICB name
    split_columns = df[column_name].str.split('ICB - ', n=1, expand=True)
    df['sub_icb_location_code'] = split_columns[1].str.strip()
    df['icb_location_name'] = split_columns[0].str.strip() + 'ICB'
    # Reorder columns to place the new columns next to each other
    new_column_order = ['icb_ons_code', 'sub_icb_location_code', 'sub_icb_location_name',
                        'icb_location_name', 'appointment_date',
                        'appointment_month', 'service_setting',
                        'context_type', 'national_category',
                        'count_of_appointments']
    df = df[new_column_order]
    return df

```

```

# Checking for duplicate values, adding a new column, where 1=duplicate, 0=not duplicate
def add_duplicate_flag(df, file_name):
    logging.info(f"Adding duplicate flag. DataFrame shape before: {df.shape}")
    try:
        df['duplicate_flag'] = 0 # Initialize the flag column with 0
        df.loc[df.duplicated(keep=False), 'duplicate_flag'] = 1
        logging.info(f"Duplicate flag added. DataFrame shape after: {df.shape}")
        return df
    except Exception as e:
        logging.error(f"Error adding duplicate flag: {str(e)}")
        return df # Return the original DataFrame if the operation fails

# Function to categorize the actual_duration column
def convert_duration(actual_duration):
    if pd.isna(actual_duration):
        return np.nan
    if '1-5' in actual_duration:
        return '1-5'
    elif '6-10' in actual_duration:
        return '6-10'
    elif '11-15' in actual_duration:
        return '11-15'
    elif '16-20' in actual_duration:
        return '16-20'
    elif '21-30' in actual_duration:
        return '21-30'
    elif '31-60' in actual_duration:
        return '31-60'
    elif 'unknown' in actual_duration or 'Data Quality' in actual_duration:
        return 'Unknown'
    else:
        return 'Unknown'

# Function to categorize the time_between_book_and_appointment column
def convert_duration_day(time_between_book_and_appointment):
    if pd.isna(time_between_book_and_appointment):
        return 'Unknown'
    time_between_book_and_appointment = ' '.join(time_between_book_and_appointment.split()).lower()
    if time_between_book_and_appointment == 'same day':
        return '0'
    elif time_between_book_and_appointment == '1 day':
        return '1'
    elif time_between_book_and_appointment == '2 to 7 days':
        return '2-7'
    elif time_between_book_and_appointment == '8 to 14 days':
        return '8-14'
    elif time_between_book_and_appointment == '15 to 21 days':
        return '15-21'
    elif time_between_book_and_appointment == '22 to 28 days':
        return '22-28'
    elif time_between_book_and_appointment == 'more than 28 days':
        return '28+'
    elif time_between_book_and_appointment == 'unknown / data quality':
        return 'Unknown'
    else:
        return 'Unknown'

# Function to display DataFrame
def display_dataframe(df, name, num_rows=5):
    if df is not None:
        logging.info(f"Displaying the first {num_rows} rows of {name}")
        display(Markdown(f"### First {num_rows} rows of {name}:"))
        display(df.head(num_rows))
        display(Markdown("<br><br>"))
    else:
        logging.error(f"DataFrame {name} is None and cannot be displayed")

```

```

# Main workflow
def main():
    file_names = ['actual_duration.csv', 'appointments_regional.csv', 'national_categories.xlsx']
    dataframes = {}

    for file_name in file_names:
        file_type = 'excel' if file_name.endswith('.xlsx') else 'csv'
        df = load_data(file_name, file_type=file_type)
        if df is not None:
            dataframes[file_name] = df

    ad = dataframes.get('actual_duration.csv')
    ar = dataframes.get('appointments_regional.csv')
    nc = dataframes.get('national_categories.xlsx')

    # Log the shapes of the Loaded data
    logging.info(f"Shape of ad after loading: {ad.shape if ad is not None else 'None'}")
    logging.info(f"Shape of ar after loading: {ar.shape if ar is not None else 'None'}")
    logging.info(f"Shape of nc after loading: {nc.shape if nc is not None else 'None'}")

    # Ensure data is loaded before proceeding
    if ad is None or ar is None or nc is None:
        logging.error("Failed to load one or more data files. Exiting.")
        return

    # Clean data
    ad = clean_data(ad, 'actual_duration.csv')
    ar = clean_data(ar, 'appointments_regional.csv')
    nc = clean_data(nc, 'national_categories.xlsx')

    # Log the shapes of the cleaned data
    logging.info(f"Shape of ad after cleaning: {ad.shape if ad is not None else 'None'}")
    logging.info(f"Shape of ar after cleaning: {ar.shape if ar is not None else 'None'}")
    logging.info(f"Shape of nc after cleaning: {nc.shape if nc is not None else 'None'}")

    # Check if any DataFrame is None after cleaning
    if ad is None or ar is None or nc is None:
        logging.error("One or more Dataframes became None after cleaning. Exiting.")
        return

    # Format appointment_date columns
    ad = correct_date_types(ad, ['appointment_date'], 'actual_duration.csv')
    nc = correct_date_types(nc, ['appointment_date'], 'national_categories.xlsx')

    # Format appointment_month columns
    ar = format_appointment_month(ar, 'appointment_month', 'appointments_regional.csv')
    nc = format_appointment_month(nc, 'appointment_month', 'national_categories.xlsx')

    # Check missing values
    check_missing_values(ad, 'actual_duration.csv')
    check_missing_values(ar, 'appointments_regional.csv')
    check_missing_values(nc, 'national_categories.xlsx')

    # Format sub_icb_location_name columns in ad
    ad = create_new_icb_location_name(ad, 'sub_icb_location_name', 'actual_duration.csv')

    # Format sub_icb_location_name columns in nc
    nc = split_code_after_hyphen_and_reorder_columns(nc, 'sub_icb_location_name', 'national_categories.xlsx')

    # Process data
    ad['actual_duration_minutes'] = ad['actual_duration'].apply(convert_duration)
    ar['days_between_booking_and_appointment'] = ar['time_between_book_and_appointment'].apply(convert_duration_day)

    ad = add_duplicate_flag(ad, 'actual_duration.csv')
    ar = add_duplicate_flag(ar, 'appointments_regional.csv')
    nc = add_duplicate_flag(nc, 'national_categories.xlsx')

    # Display cleaned data
    display_dataframe(ad, 'cleaned_actual_duration')
    display_dataframe(ar, 'cleaned_appointments_regional')
    display_dataframe(nc, 'cleaned_national_categories')

    # Save results
    OUTPUT_PATH = './' # Define your output path here
    save_results(ad, os.path.join(OUTPUT_PATH, 'cleaned_actual_duration.csv'))
    save_results(ar, os.path.join(OUTPUT_PATH, 'cleaned_appointments_regional.csv'))
    save_results(nc, os.path.join(OUTPUT_PATH, 'cleaned_national_categories.csv'))


    logging.info("Data cleaning completed successfully.")

if __name__ == "__main__":
    main()

```

5. Exploratory Data analysis, statistics and data normalisation and standardisation

During this section the summary statistics is applied to the three datasets, numerical and non-numerical columns are separated for this. An outlier column was created to mark values that are outliers within the count_of_appointments column. However this was not routinely removed as it seemed that a very large number of appointments would have been excluded. Instead a separate file was created with removing the outlier values for further use. By



loading the filtered file instead of the main files. I felt this was a better approach as it excludes mainly the highest number of appointments. A new column was created for the various data standardisation and normalisation techniques. However it was not routinely used in the analysis process. Further predictive analysis was not within the scope of this analysis.

During the EDA it became apparent that `national_categories` and `actual_duration` datasets had not been flagged by duplicates. Whilst `appointments_regional` file had a high number of duplicates. It was cross-checked and it became apparent that on a monthly level `national_categories` have the exact same appointments for the period that it covers as `appointment_regional`. So it was concluded that the identified duplicates are a result of the daily appointments being aggregated into a monthly level. The average monthly duplicates are similar before the dates `national_categories` file covers and for the dates that file covers, so it can be inferred that the flagged duplicate data in `appointment_regional` in fact are not duplicates, but are a result of the monthly aggregation. Therefore the created duplicate flag column was removed from the datasets.

You may notice that the datasets are often re-loaded from the files at each section or question. This may make the code to run longer, however I used this approach to prevent accidental overriding of the data-frames within the code. The necessary major steps were saved into a cvs for future use.

Stage 2 - Exploratory Data Analysis - Statistics, data normalisation and standardisation

```
]:  
# Functions for EDA  
# Function to generate summary statistics  
def generate_summary_statistics(df, file_name):  
    logging.info(f'Summary statistics for {file_name}:')  
    numerical_cols = df.select_dtypes(include=[np.number]).columns.tolist()  
    categorical_cols = df.select_dtypes(exclude=[np.number]).columns.tolist()  
    if numerical_cols:  
        logging.info(f'Numerical summary statistics for {file_name}: \n{df[numerical_cols].describe()}\n')  
    if categorical_cols:  
        logging.info(f'Categorical summary statistics for {file_name}: \n{df[categorical_cols].describe(include=["object", "category"])\n')  
  
# Function to detect outlier values  
def detect_outlier_values(df, column, file_name):  
    """  
    Detects outliers in a specified numerical column of a DataFrame using the IQR method and adds a flag column.  
  
    Parameters:  
    - df: DataFrame to analyze.  
    - column: Column name to detect outliers.  
    - file_name: Name of the file or dataset for logging purposes.  
  
    Returns:  
    - Dictionary with IQRLower, IQRUpper, and the number of outliers.  
    """  
    logging.info(f'\nOutlier detection for {file_name} - {column}:')  
  
    Q1 = df[column].quantile(0.25)  
    Q3 = df[column].quantile(0.75)  
    IQR = Q3 - Q1  
    IQRLower = Q1 - 1.5 * IQR  
    IQRUpper = Q3 + 1.5 * IQR  
  
    outliers = df[(df[column] < IQRLower) | (df[column] > IQRUpper)]  
    logging.info(f'Column: {column}')  
    logging.info(f'Q1: {Q1}, Q3: {Q3}, IQR: {IQR}')  
    logging.info(f'IQRLower: {IQRLower}, IQRUpper: {IQRUpper}')  
    logging.info(f'Number of outliers: {outliers.shape[0]}')  
  
    # Add outlier flag column  
    outlier_flag_column = f'outlier_flag'  
    df[outlier_flag_column] = 0  
    df.loc[(df[column] < IQRLower) | (df[column] > IQRUpper), outlier_flag_column] = 1  
  
    return {  
        'IQRLower': IQRLower,  
        'IQRUpper': IQRUpper,  
        'num_outliers': outliers.shape[0]  
    }  
  
# Function to display distinct categories and their counts  
def display_distinct_categories(df, file_name):  
    print(f'\nDistinct categories and their counts in {file_name}:')  
    for column in df.columns:  
        distinct_counts = df[column].value_counts()  
        print(f'\nDistinct Categories in "{column}":')  
        print(distinct_counts)  
  
# Function to perform absolute max scaling  
def absolute_max_scaling(df, column):  
    max_value = df[column].abs().max()  
    abs_max_scaled_column = df[column] / max_value  
    return abs_max_scaled_column  
  
# Function to perform min-max scaling  
def min_max_scaling(df, column):  
    min_value = df[column].min()  
    max_value = df[column].max()  
    min_max_scaled_column = (df[column] - min_value) / (max_value - min_value)  
    return min_max_scaled_column  
  
# Function to perform z-score normalization  
def z_score_normalization(df, column):  
    z_score_scaled_column = (df[column] - df[column].mean()) / df[column].std()  
    return z_score_scaled_column
```



```

# Main workflow for EDA
def main():
    try:
        # Load cleaned data
        ad = load_data("cleaned_actual_duration.csv")
        ar = load_data("cleaned_appointments_regional.csv")
        nc = load_data("cleaned_national_categories.csv")

        # Ensure data is loaded before proceeding
        if ad is None or ar is None or nc is None:
            logging.error("Failed to load one or more data files. Exiting.")
            return

        # Generate summary statistics
        generate_summary_statistics(ad, 'cleaned_actual_duration')
        generate_summary_statistics(ar, 'cleaned_appointments_regional')
        generate_summary_statistics(nc, 'cleaned_national_categories')

        # Outlier detection
        num_outliers_ad = detect_outlier_values(ad, 'count_of_appointments', 'cleaned_actual_duration')
        num_outliers_ar = detect_outlier_values(ar, 'count_of_appointments', 'cleaned_appointments_regional')
        num_outliers_nc = detect_outlier_values(nc, 'count_of_appointments', 'cleaned_national_categories')

        logging.info(f"Number of outliers in cleaned_actual_duration: {num_outliers_ad['num_outliers']}")
        logging.info(f"Number of outliers in cleaned_appointments_regional: {num_outliers_ar['num_outliers']}")
        logging.info(f"Number of outliers in cleaned_national_categories: {num_outliers_nc['num_outliers']}")

        # Display distinct categories for ad
        display_distinct_categories(ad, 'cleaned_actual_duration')
        # Display distinct categories for ar
        display_distinct_categories(ar, 'cleaned_appointments_regional')
        # Display distinct categories for nc
        display_distinct_categories(nc, 'cleaned_national_categories')

        # Perform absolute max scaling
        ad['max_scaled_appointments'] = absolute_max_scaling(ad, 'count_of_appointments')
        ar['max_scaled_appointments'] = absolute_max_scaling(ar, 'count_of_appointments')
        nc['max_scaled_appointments'] = absolute_max_scaling(nc, 'count_of_appointments')

        logging.info("Performed absolute max scaling on count_of_appointments for all datasets.")

        # Perform min-max scaling
        ad['min_max_scaled_appointments'] = min_max_scaling(ad, 'count_of_appointments')
        ar['min_max_scaled_appointments'] = min_max_scaling(ar, 'count_of_appointments')
        nc['min_max_scaled_appointments'] = min_max_scaling(nc, 'count_of_appointments')

        logging.info("Performed min-max scaling on count_of_appointments for all datasets.")

        # Perform z-score normalization
        ad['z_score_scaled_appointments'] = z_score_normalization(ad, 'count_of_appointments')
        ar['z_score_scaled_appointments'] = z_score_normalization(ar, 'count_of_appointments')
        nc['z_score_scaled_appointments'] = z_score_normalization(nc, 'count_of_appointments')

        logging.info("Performed z-score normalization on count_of_appointments for all datasets.")

        # Based on the metadata calculation add the estimated total appointments to ar
        # Define the percentage of patients included in the data collection
        percentage_of_patients_included = 0.964 # 96.4%

        # Calculate the estimated total number of appointments for England
        ar['estimated_total_appointments'] = ar['count_of_appointments'] / percentage_of_patients_included

        # Save results
        save_results(ad, 'scaled_actual_duration.csv')
        save_results(ar, 'scaled_appointments_regional.csv')
        save_results(nc, 'scaled_national_categories.csv')

        logging.info("EDA completed successfully.")

    except Exception as e:
        logging.exception("An error occurred during EDA")

if __name__ == "__main__":
    main()

```

6. **The EDA visualizations section** – the boxplots for the count_of_appointments were log scaled in order to visualize it as without this scaling it is not possible to see the box and whiskers due to the large range of the dataset. The output of the visualisations were saved to a pdf file. It is important to note that the countplots show the number of times a category occurred and it was not related to the number of appointments per that category.

Exploratory Data Visualisation

```
[ ]: # Histogram chart function
def plot_histograms(df, columns, file_name, pdf):
    for column in columns:
        try:
            logging.info(f"Attempting to plot histogram for {column} in {file_name}")
            plt.figure(figsize=(10, 6))
            sns.histplot(df[column], kde=True, bins=30)
            plt.title(f"Histogram of {column} in {file_name}")
            plt.xlabel(column)
            plt.ylabel('Frequency')
            pdf.savefig()
            plt.show()
        except KeyError:
            logging.error(f"{column} not found in DataFrame for {file_name}")
        except Exception as e:
            logging.error(f"Error while plotting histogram for {column} in {file_name}: {str(e)}")

# Function for correlation matrix
def plot_correlation_matrix(df, file_name, pdf):
    try:
        logging.info(f"Plotting correlation matrix for {file_name}")
        plt.figure(figsize=(12, 8))
        numeric_df = df.select_dtypes(include=[np.number])
        correlation_matrix = numeric_df.corr()
        sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
        plt.title(f'Correlation matrix for {file_name}')
        plt.tight_layout()
        pdf.savefig()
        plt.show()
        logging.info(f"Finished plotting correlation matrix for {file_name}")
    except Exception as e:
        logging.exception(f"An error occurred while plotting correlation matrix for {file_name}")

# Function to create countplot visualization
def plot_countplot(df, column, file_name, pdf):
    plt.figure(figsize=(14, 8))
    value_counts = df[column].value_counts()
    if len(value_counts) > 20:
        top_10 = value_counts.nlargest(10)
        bottom_10 = value_counts.nsmallest(10).sort_values(ascending=False)
        value_counts = pd.concat([top_10, bottom_10])

    sns.countplot(y=df[column], order=value_counts.index)
    plt.title(f'Count Plot of {column} in {file_name}', fontsize=14, fontweight='bold')
    plt.subtitle('Note: For larger sets, the top and bottom 10 is displayed', fontsize=10, style='italic', y=0.92)
    plt.xlabel('Count')
    plt.ylabel(column)
    plt.tight_layout(rect=[0, 0, 1, 0.95])
    pdf.savefig()
    plt.show()

# Function to create a visualization showing trends over time
def plot_trends_over_time(df, date_column, value_column, file_name, pdf):
    plt.figure(figsize=(14, 7))
    df[date_column] = pd.to_datetime(df[date_column])
    df.set_index(date_column)[value_column].resample('M').sum().plot()
    plt.title(f'Trend of {value_column} over time in {file_name}')
    plt.xlabel('Date')
    plt.ylabel(value_column)
    plt.tight_layout()
    pdf.savefig()
    plt.show()
```

```

# Function to plot box plots with Log transformation and enhanced annotations
def plot_boxplots(df, column, file_name, pdf):
    plt.figure(figsize=(10, 6))
    try:
        # Apply a logarithmic transformation to data; add a small constant to avoid log(0)
        data = np.log(df[column] + 1)
        sns.boxplot(x=data)

        # Main title and subtitle
        plt.suptitle(f'Log Transformed Box Plot of {column}', fontsize=14, fontweight='bold')
        plt.title('Values are log-scaled to reduce skewness', fontsize=10, style='italic')

        # Customizing x-axis label to reflect Log transformation
        plt.xlabel(f'Log of {column}')

        # Adding grid for better readability
        plt.grid(True)

        # Add annotations for e2, e4, e6, etc.
        tick_values = [2, 4, 6, 8, 10, 12]
        tick_labels = [f'{np.exp(tick):.2f}' for tick in tick_values]
        plt.xticks(ticks=tick_values, labels=tick_labels)

        plt.tight_layout(rect=[0, 0.03, 1, 0.95]) # Adjust layout to make sure everything fits without overlap
        pdf.savefig()
        plt.show()

    except Exception as e:
        logging.error(f'Error while plotting box plot for {column} in {file_name}: {str(e)}")

# Main workflow
def main():
    file_names = ['scaled_actual_duration.csv', 'scaled_appointments_regional.csv', 'scaled_national_categories.csv']
    columns_to_plot = ['count_of_appointments']

    with PdfPages('EDA_visualizations.pdf') as pdf:
        try:
            for file_name in file_names:
                df = load_data(file_name)
                if df is not None:
                    plot_histograms(df, columns_to_plot, file_name, pdf)
                    plot_correlation_matrix(df, file_name, pdf)
                    for column in columns_to_plot:
                        plot_boxplots(df, column, file_name, pdf)
                    for column in df.select_dtypes(include=['object', 'category']).columns:
                        plot_countplot(df, column, file_name, pdf)
                    if 'appointment_date' in df.columns:
                        plot_trends_over_time(df, 'appointment_date', 'count_of_appointments', file_name, pdf)
                    elif 'appointment_month' in df.columns:
                        plot_trends_over_time(df, 'appointment_month', 'count_of_appointments', file_name, pdf)
                    logging.info("Data visualization completed successfully.")
        except Exception as e:
            logging.exception("An error occurred during data visualization")

if __name__ == "__main__":
    main()

```

- Data merging:** The computer the analysis was performed did not have enough working memory to perform a direct merge of the files, even when data was limited to a few columns. pgAdmin was used to complete the merge of the files and then these were loaded to the jupyter notebook. pgAdmin was also checked to confirm the potential duplicates within the datasets.

See below the code that was used in pgAdmin to achieve the merging of the files.

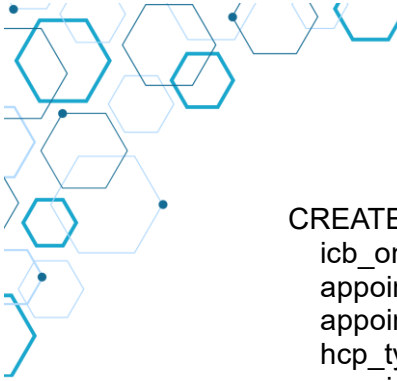
Two files were created merged_ad_nc for daily level view.

Final_merged_data for monthly level view.

```

CREATE TABLE actual_duration (
    sub_icb_location_code VARCHAR (10),
    sub_icb_location_ons_code VARCHAR (12),
    sub_icb_location_name VARCHAR (200),
    icb_ons_code VARCHAR (12),
    region_ons_code VARCHAR (12),
    appointment_date VARCHAR (15),
    actual_duration VARCHAR (50),
    count_of_appointments NUMERIC(10, 0)
);

```



```
CREATE TABLE appointments_regional (  
  icb_ons_code VARCHAR (12),  
  appointment_month VARCHAR (15),  
  appointment_status VARCHAR (50),  
  hcp_type VARCHAR (100),  
  appointment_mode VARCHAR (100),  
  time_between_book_and_appointment VARCHAR (50),  
  count_of_appointments NUMERIC(10, 0)  
);
```

```
CREATE TABLE national_categories (  
  appointment_date DATE,  
  icb_ons_code VARCHAR (12),  
  sub_icb_location_name VARCHAR (200),  
  service_setting VARCHAR (100),  
  context_type VARCHAR (200),  
  national_category VARCHAR (200),  
  count_of_appointments NUMERIC(10, 2),  
  appointment_month VARCHAR (12)  
);
```

```
SELECT * FROM national_categories  
LIMIT 5;
```

```
SELECT * FROM appointments_regional  
LIMIT 5;
```

```
SELECT * FROM actual_duration  
LIMIT 15;
```

```
-- Check for duplicates and count them  
SELECT  
  icb_ons_code,  
  appointment_month,  
  appointment_status,  
  hcp_type,  
  appointment_mode,  
  time_between_book_and_appointment,  
  count_of_appointments,  
  COUNT(*) AS duplicate_count  
FROM  
  appointments_regional  
GROUP BY  
  icb_ons_code,  
  appointment_month,  
  appointment_status,  
  hcp_type,  
  appointment_mode,  
  time_between_book_and_appointment,  
  count_of_appointments
```




```
HAVING
  COUNT(*) > 1;

-- Count the number of sets of exact duplicate rows
SELECT
  COUNT(*) AS number_of_duplicate_sets
FROM
  (
    SELECT
      icb_ons_code,
      appointment_month,
      appointment_status,
      hcp_type,
      appointment_mode,
      time_between_book_and_appointment,
      count_of_appointments
    FROM
      appointments_regional
    GROUP BY
      icb_ons_code,
      appointment_month,
      appointment_status,
      hcp_type,
      appointment_mode,
      time_between_book_and_appointment,
      count_of_appointments
    HAVING
      COUNT(*) > 1
  ) AS duplicate_sets;

CREATE TABLE merged_ad_nc AS
SELECT
  ad.sub_icb_location_name,
  ad.actual_duration,
  nc.service_setting,
  nc.context_type,
  nc.national_category,
  nc.count_of_appointments,
  ad.appointment_date
FROM
  actual_duration ad
JOIN
  national_categories nc
ON
  ad.icb_ons_code = nc.icb_ons_code
  AND ad.appointment_date = nc.appointment_date;

SELECT * FROM merged_ad_nc
LIMIT 10

-- Check row count
SELECT COUNT(*) FROM merged_ad_nc;
```



```

-- Compare with original tables
SELECT COUNT(*) FROM actual_duration;
SELECT COUNT(*) FROM national_categories;

-----this is the correct code to account for the different appointment counts in ad and
nc-----
CREATE TABLE merged_ad_nc_2 AS
SELECT
    ad.icb_ons_code,
    ad.sub_icb_location_name,
    ad.actual_duration,
    ad.count_of_appointments AS ad_count_of_appointments,
    nc.service_setting,
    nc.context_type,
    nc.national_category,
    nc.count_of_appointments AS nc_count_of_appointments,
    ad.appointment_date
FROM
    actual_duration ad
JOIN
    national_categories nc
ON
    ad.icb_ons_code = nc.icb_ons_code
    AND ad.appointment_date = nc.appointment_date;

-----to save the created
COPY merged_ad_nc_2 TO 'C:\\Users\\Public\\psql_play\\merged_ad_nc_2.csv'
WITH CSV HEADER;

ALTER TABLE merged_ad_nc_2
ADD COLUMN appointment_month DATE;


UPDATE merged_ad_nc_2
SET appointment_month = DATE_TRUNC('month', appointment_date);

ALTER TABLE appointments_regional
ADD COLUMN appointment_month_date DATE;

UPDATE appointments_regional
SET appointment_month_date = TO_DATE(appointment_month, 'YYYY-MM');

CREATE TABLE aggregated_merged_ad_nc AS
SELECT
    appointment_month,
    icb_ons_code,
    sub_icb_location_name,
    actual_duration,
    service_setting,
    context_type,
    national_category,
    SUM(nc_count_of_appointments) AS total_nc_count_of_appointments
FROM
    merged_ad_nc_2

```



```

GROUP BY
    icb_ons_code,
        sub_icb_location_name,
        actual_duration,
    service_setting,
    context_type,
    national_category,
    appointment_month;

SELECT * FROM aggregated_merged_ad_nc
LIMIT 5

SELECT * FROM appointments_regional
LIMIT 5

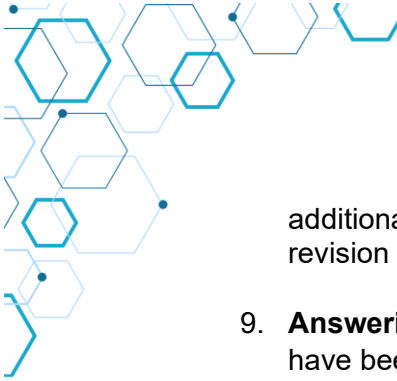
CREATE TABLE final_merged_data AS
SELECT
    a.icb_ons_code,
        a.sub_icb_location_name,
        a.actual_duration,
        a.service_setting,
        a.context_type,
        a.national_category,
    a.appointment_month,
    a.total_nc_count_of_appointments,
    ar.appointment_status,
    ar.hcp_type,
    ar.appointment_mode,
    ar.time_between_book_and_appointment
FROM
    aggregated_merged_ad_nc a
JOIN
    appointments_regional ar
ON
    a.icb_ons_code = ar.icb_ons_code
    AND a.appointment_month = ar.appointment_month_date;

SELECT * FROM final_merged_data
LIMIT 10;

COPY final_merged_data TO 'C:\\Users\\Public\\psql_play\\final_merged.csv' WITH
CSV HEADER;

```

Whilst I was able to create the final merged file with all columns for from the three dataframes I was unable to load this again to python due to limitation of the computing resources. Therefore the planned analysis of finding out which national categories contributed to longer waiting times or if there was any difference in terms of service_setting, as well as examining appointment length by hcp_type, apointemnt_mode and service_setting. A final attempt was made to connect to pgAdmin using python code to access the table from the created database, however again this was putting strain on the computer and was not able to perform this section of the analysis. This remains a limitation of the analysis. The generated file become 47 Gigabytes large csv file. It is possible that the merge has created



additional rows. Unfortunately the timescale of the analysis did not allow for further revision of this.

9. **Answering business questions** – some of the answers for the first questions could have been also gained from the EDA sections.

The visualisations for x or y axis labels are annotated with a $1e6$ – 1 million, $1e7$ – 10 million or $1e8$ – 100 million for the values. If you wish to use the exact number add `plt.ticklabel_format(style='plain', axis='y')` snippet to the visualisations. It was felt that the numbers would be difficult to distinguish, therefore the 1e numbers were used in the visualisations. (For stakeholders presentation this was annotated using the presentation program)

- a. Please refer to the jupyter notebook for the rest of the code for answering the business questions.
- b. When analysing the twitter data it was identified that there are duplicates in the dataset, where all data is the same apart from the twitter ID, the first instances were kept (note, the results were similar even when the whole dataset was examined). As the original twitter data was not focusing on NHS data, another example was brought in from <https://www.kaggle.com/datasets/gpreda/covid19-tweets?resource=download>. Due to changing in twitter/X API use, unable to access more recent data freely.
- c. For regional and geographic analysis the dataframes were mapped with a dictionary to ensure accurate representation of the NHS Regions. Due to the region ONS code covering different areas.

- d. Additional population/patient data was introduced from <https://digital.nhs.uk/data-and-information/publications/statistical/patients-registered-at-a-gp-practice>

The population data was used from 08/2022, which is after the reporting period, however it used the same naming conventions for the ICBs. Previous population/patient data used the CCG names.

- The staffing data was accessed from <https://digital.nhs.uk/data-and-information/publications/statistical/general-and-personal-medical-services/31-july-2022>. From the xlsx file the required tab was saved and cleaned in excel prior to use in python.
- A geographic mapping was included for the 42 ICB locations
- Additional metrics such appointment density (total number of appointments per GP surgery and per capita appointments were examined in the regions and ICBs.
- Predictive analytics was performed on a regional basis, with outliers removed using the interquartile range and limit methods and resampling methods to gain the decomposition and the ARIMA/SARIMA predictions. Note that when the data was limited to the last year the SARIMA predictions did not work due to lack of seasonality data. Further analysis could be done in ICB or Sub-ICB locations in terms of predictive analytics. Offsetting the predictive analytics results from this data with more recent data points to train the models. This could be used for predictions on real time data.
- COVID-19 Data was also sourced. It was found that gaining access to daily England records of COVID cases and deaths was very difficult to obtain.




Two separate datasets were introduced from <https://covid.ourworldindata.org/data/owid-covid-data.csv> and https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv.

No significant correlation was found between the covid cases, deaths and the appointments. Perhaps different approach would have been necessary.

- It was also very difficult to obtain weather data for the specified periods using the weather API. Several sites were accessed. The final set was included from two separate request from <https://www.noaa.gov/> . Data was only available for London and Birmingham in Fahrenheit. Excel was used to calculate the values in Celsius. No significant relationship was found in terms of the examined weather data. Due to the key metrics such as DNA data was on a monthly aggregation, it was difficult to infer impact on a daily level. Additional weather warning and extreme weather points could potentially impact on appointments as well as traffic or other major disruption, future analysis could look into this.

Note that some markdown boxes jumbled the text during the analysis. All effort was made to eliminate this, however some may still appear not properly formatted.

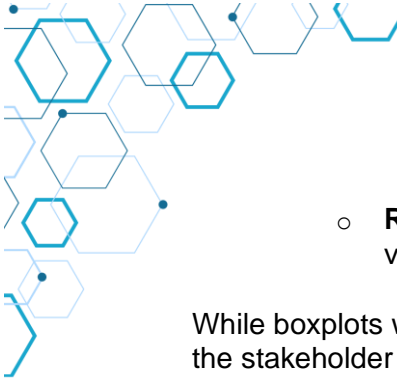


APPENDIX 6 – Visualisation for Business Stakeholders

Rationale

Visualizations Chosen: (Refer to the presentation)

1. **Line Charts for Time-Series Analysis:**
 - **Purpose:** To capture trends over time in appointment data, such as the number of attended, missed appointments, appointment modes or categories over time.
 - **Rationale:** Line charts are particularly effective for visualizing trends over time, allowing stakeholders to easily see fluctuations, seasonal patterns, and long-term trends. This helps in identifying peak times, potential capacity issues, and the impact of external factors on appointment attendance.
2. **Moving Average and Decomposition Plots:**
 - **Purpose:** To smooth out short-term fluctuations and highlight underlying trends and seasonal patterns in appointment data.
 - **Rationale:** Moving averages and decomposition plots are essential for separating the noise from meaningful patterns. This enables the NHS to better understand the cyclical nature of appointments and plan resources accordingly.
3. **Correlation Scatter Plots:**
 - **Purpose:** To explore potential relationships between different variables, such as the correlation between appointment counts and COVID-19 cases or weather data. Scatter plot for regional breakdown of appointment density, per capita values.
 - **Rationale:** Scatter plots are ideal for identifying correlations and potential causal relationships. This helps in determining whether external factors significantly influence appointment attendance, which can inform future decision-making. The scatter plots created for weather and COVID data were not included in the stakeholder presentation, as they did not show correlation between these variables.
4. **Pie Charts:**
 - **Purpose:** To represent the proportion of attended versus missed appointments, and to break down missed appointments by factors such as waiting time, as well as highlighting percentage of appointment duration or waiting times.
 - **Rationale:** Pie charts are effective for displaying parts of a whole, making it easy for stakeholders to understand the distribution of appointment outcomes at a glance. This is crucial for evaluating the efficiency of patient management and identifying areas for improvement.
5. **Bar Charts:**
 - **Purpose:** To compare appointment data across different regions, healthcare settings, or ICBs.
 - **Rationale:** Bar charts allow for straightforward comparison between different categories, such as regions or time periods. This is essential for identifying disparities in resource allocation or performance across the NHS network.
6. **Heat maps:**
 - **Purpose:** ICB insights were visualised on a heatmap of England with the ICB boundaries for stakeholders to easily compare the performance of different ICBs

- 
- **Rationale:** Heat maps over a geographical location aids for better visualisation of geographical data

While boxplots were routinely used during the analysis process, these were not included in the stakeholder visualisations due to more difficulty in interpreting the meanings of these.

Interpretations of Visualization Outputs:

- **Trend Analysis:** The line charts and moving averages revealed clear seasonal patterns in appointment bookings, indicating periods of high demand, especially at late autumn months. This insight is critical for resource planning, enabling the NHS to allocate staff more effectively during peak times. Whilst it was important to bear in mind that the reporting period was highly affected by COVID-19 measures, such as lockdowns.
- **Impact of External Factors:** The scatter plots exploring the correlation between COVID-19 cases and weather changes did not show strong relationships. Further analysis would have been needed into these factors, as there was no daily data available in terms of missed appointment, it was not possible to draw conclusions, weather extreme weather conditions lead to more appointments overall or missed appointments.
- **Proportional Analysis:** The pie charts illustrated that only a very small percentage of appointments are missed, half of these were less booked than a week, whilst the other half over a week. 17.4% of same day appointments and a further 15.4% booked the day before were missed, which could be an important factor to explore the reasons for these.
- **Comparative Analysis:** Bar charts comparing different regions and ICBs identified areas with potential over or under-utilization of healthcare resources, pointing to opportunities for redistributing resources or adjusting capacity in specific regions.

Relevance to Business Objectives: These visualizations directly support the NHS's objectives by providing actionable insights into staff adequacy and resource utilization. By understanding trends, seasonal variations, and the impact of external factors, the NHS can make data-driven decisions to optimize scheduling, improve patient care, and ensure that resources are used efficiently across the network. The insights gained from these visualizations can guide strategic planning, policy development, and operational adjustments to better meet patient needs and enhance service delivery.



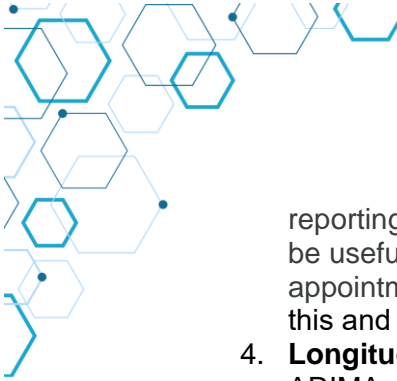
APPENDIX 7 – Recommendations

Service improvement recommendations:

1. **Staff Allocation:** NHS should allocate more staffing on busier days, months, and seasons. This proactive measure will ensure that resources are available when demand is highest.
2. **Systematic Resource Management:** Conduct a detailed review of current resource allocation to identify areas where improvements or additional staffing can mitigate overall service demand. This includes analysing local demands and setting targeted staffing levels accordingly.
3. **Long Appointment Management:** Investigate the reasons for extended appointment durations and explore strategies to manage these more efficiently, potentially reducing staffing demands without compromising care quality.
4. **Data Mapping and Reporting:** Provide additional staff education or implement system improvements to eliminate unmapped or inconsistently mapped appointment data. Accurate data is crucial for making informed decisions.
5. **Regional Staffing Adjustments:** Regions like London, with lower staff-to-patient ratios and high appointment density, require staffing improvements. Additionally, London's high rate of missed appointments suggests the need for targeted interventions.
6. **Resilience Building:** Enhance resilience to major outbreaks by learning from COVID-19's impact on appointment handling, ensuring that future crises do not overwhelm the system.
7. **Telehealth and Online Appointments:** Continue improving telehealth and online appointment services to offer patients more flexible options and potentially reduce in-person demand.
8. **Strain on Services:** Recognize that increased appointment volumes correlate with reduced appointment times and longer wait periods, suggesting that under strain, staff may rush through appointments, and patients face longer wait times.
9. **Proactive Follow-Up Systems:** Implement automated reminders (e.g., SMS, email) for upcoming appointments, especially those scheduled far in advance, to reduce DNA rates. Investigate the high rate of missed same-day or next-day appointments to understand if patients seek alternatives like A&E.
10. **Streamlined Data Collection:** Enhance data collection and reporting processes, providing daily data on missed appointments before monthly aggregation, enabling more granular analysis of missed appointments' causes.
11. **Appointment Length Management:** Integrate data on the planned versus actual appointment length to better understand staff strain.
12. **Waiting time reduction**
13. **Capacity Clarification:** Clarify the 1.2 million appointment threshold's context, determining whether it also represents the planned appointments or just the NHS's maximum capacity.

Further analysis recommendations

1. **Service Overutilization:** Explore the causes of service overutilization beyond the suggested 1.2 million England baseline, setting more localized and specific targets.
2. **Seasonal and External Factors:** Investigate the impact of seasonal illnesses or other external factors on GP appointments, particularly how these influence demand.
3. **Integration of Patient and Staffing Metrics:** Whilst this analysis added patient data from 08/2022 and staffing levels from 07/2022 considering variations over the



reporting period was not within the scope of this analysis. For future analysis it would be useful for the NHS to consider patient metrics and staffing metrics alongside the appointment data. Reliable, original data sources would enhance the robustness of this and future analysis.

4. **Longitudinal Analysis:** Compare actual appointment trends from 2022/23 with ARIMA and SARIMA predictions to evaluate these models' reliability and consider using alternative models like X11 for future forecasts.
5. **Local Performance Analysis:** Conduct focused analyses on specific GPs and ICBs to gain deeper insights into local performance, identifying high-performing areas and leveraging these insights to uplift lower-performing surgeries. This could be further discussed with stakeholders.

Suggestions for further analysis metrics to be considered for the NHS

1. No-Show Rate

This metric tracks the percentage of patients who fail to turn up for their appointments without notifying the clinic or hospital.

$$\text{No-Show Rate} = \left(\frac{\text{Number of No-Shows}}{\text{Total Scheduled Appointments}} \right) \times 100$$

2. Cancellation Rate

Measures the percentage of scheduled appointments that are cancelled by the patient before the appointment date.

$$\text{Cancellation Rate} = \left(\frac{\text{Number of Cancellations}}{\text{Total Scheduled Appointments}} \right) \times 100$$


3. Utilization Rate

Indicates how effectively appointment slots are being used. It's calculated by dividing the number of attended appointments by the total number of available appointment slots.

$$\text{Utilization Rate} = \left(\frac{\text{Attended Appointments}}{\text{Available Slots}} \right) \times 100$$

4. Lead Time for Appointments

Reflects the average time between the day an appointment is booked and the actual appointment date. This metric helps understand how far in advance patients need to book to get an appointment.


$$\text{Lead Time} = \frac{\text{Total Days Until All Appointments}}{\text{Number of Appointments}}$$

5. Patient Scheduling Efficiency

Calculated by assessing how closely the actual appointment lengths match the scheduled times, indicating how accurately the clinic can schedule appointments.

$$\text{Scheduling Efficiency} = \left(\frac{\text{Scheduled Appointment Time}}{\text{Actual Appointment Time}} \right) \times 100$$

6. Rate of Follow-Up Appointments

This metric calculates the percentage of appointments that result in a follow-up appointment, which can indicate the complexity of care or patient retention.

$$\text{Follow-Up Rate} = \left(\frac{\text{Number of Follow-Up Appointments}}{\text{Total Appointments}} \right) \times 100$$

7. Patient Satisfaction Scores

Collected through surveys post-appointment to assess various aspects like the ease of booking, clarity of information provided, and overall satisfaction with the service received.



APPENDIX 8 – REFERENCES

- Sebastian Traeger - Root Cause Analysis with 5 Whys Technique (With Examples)
<https://reliability.com/resources/articles/5-whys-root-cause-analysis/>
- Mind Tools - SWOT Analysis <https://www.mindtools.com/amtbj63/swot-analysis>
- <https://www.gov.uk/coronavirus>, <https://www.nhs.uk/conditions/coronavirus-covid-19/>
<https://www.bbc.com/news/uk-51506729>)
<https://www.bbc.com/news/health-55274833>)
<http://bbc.com/news/health-omicron>
<http://gov.uk/omicron>
<https://www.theguardian.com/society/2021/nov/25/nhs-braced-for-toughest-winter-ever-as-staff-warn-of-burnout>
<https://www.independent.co.uk/news/health/nhs-winter-crisis-staff-burnout-b1944537.html>)
<https://www.theguardian.com/world/2022/feb/24/end-england-covid-restrictions-measures-to-stay>, <https://www.gov.uk/government/news/prime-minister-sets-out-plan-for-living-with-covid>
<https://www.bbc.com/news/health-60662037>
<https://www.england.nhs.uk/wp-content/uploads/2022/05/B1770-integrated-care-boards-establishment-order-2022.pdf>
<http://gov.uk/Brexit>
<http://nhsconfed.org/brexit>
<http://gov.uk/seasonal-flu>
<http://bbc.com/news/flu>
<https://digital.nhs.uk/data-and-information/publications/statistical/patients-registered-at-a-gp-practice/august-2022>
<https://cms.nhsbsa.nhs.uk/sicbls-icbs-and-other-providers/organisation-and-prescriber-changes/icbs>
- ISO 8000-1:2022 - Data quality:
<https://cdn.standards.iteh.ai/samples/81745/8297300701de4336bc72fea9ab655a1e/ISO-8000-1-2022.pdf>
- DAMADMBOK: Data Management Body of Knowledge: Provides comprehensive guidance on data management practices, including data quality management.
<https://dama.org/content/body-knowledge>
- Gartner Data Quality Framework - <https://www.gartner.com/smarterwithgartner/how-to-improve-your-data-quality>
<https://www.kaggle.com/datasets/gpreda/covid19-tweets?resource=download>,
- <https://digital.nhs.uk/data-and-information/publications/statistical/patients-registered-at-a-gp-practice>
- https://geoportal.statistics.gov.uk/datasets/0867d40e053441e582e538aca3a0c59b_0/explor
[e](https://geoportal.statistics.gov.uk/datasets/0867d40e053441e582e538aca3a0c59b_0/explor)
- <https://covid.ourworldindata.org/data/owid-covid-data.csv>
[https://github.com/CSSEGISandData/COVID-19/blob/master/csse covid 19 data/csse covid 19 time series/time series covid19 confir med global.csv](https://github.com/CSSEGISandData/COVID-19/blob/master/csse%20covid%2019%20data/csse%20covid%2019%20time%20series/time%20series%20covid19%20confir%20med%20global.csv)
- <https://peps.python.org/pep-0008/>
-



<https://www.england.nhs.uk/2023/01/nhs-drive-to-reduce-no-shows-to-help-tackle-long-waits-for-care/>
<https://bjgp.org/content/bjgp/71/707/e406.full.pdf>
<https://www.england.nhs.uk/2019/01/misled-gp-appointments-costing-nhs-millions/>

Pictures –

<https://i0.wp.com/onlinepmcourses.com/wp-content/uploads/2017/07/Standard-Stakeholder-Map.png?ssl=1>