This is a technical report to explain how the data was analysed to draw derive answers to the questions posed.

Importing and exploring the data.

- Import "actual_duration" (csv), "appointments_regional" (csv) and "national_categories" (xlsx).
- Import "Pandas" and "Numpy" libraires.
- Create 3 data frames based on imported files and assign aliases:
    - actual_duration = "ad"
    - appointments_regional = "ar"
    - national_categories = "nc"
- Check for missing values using "isnull" function.
- Verify metadata using "info" function.
- Check descriptive data using "describe" function.
- To find number of locations, look up number of unique "icb_ons_code" in "nc".
- To find number of sub-locations, look up number of unique "sub_icb_location_name" in "nc".
- To find the five locations with the highest number of records, use "value_count" function to count the number of times each location in the "sub_icb_location_name" was referenced and highlighted the top 5.
- To find the number of service settings look up number of unique service_settings in the "nc". Repeat this step for context_types, national_categories and appointment_status.
- **NOTE**: *"nc" data frame was used to answer all questions pertaining to locations, service settings, context types, national categories and appointment statuses because, it had the largest amount of data, all data required to answer the questions was in its data frame and most of the columns in the other data frames were also present in the "nc" data frame.*

Analyse the data

Question 1: Between what dates were appointments scheduled?

- Check data types of appointment_date in "ad" and "nc" by checking metadata and confirm "datetime" format. Convert to "datetime" format if necessary.
- To determine appointment date range, use the "min" and "max" functions to find the earliest and latest appointment dates in "ad" and "nc".

Question 2: Which service setting was the most popular for NHS North West London from 1 January to 1 June 2022?

- Create a subset of the "nc" based on "sub_icb_location_name" value "NHS North West London ICB - W2U3Z".
- Filter subset by the date range in question.
- Count number of records of unique "service_setting" values using the "value_count" function and list results in descending order.

Question 3: Which month had the highest number of appointments?

- Sum "count_of_appointments" using "sum" function and group results by "appointment_month" using "groupby" function then list results in descending order.

Question 4: What was the total number of records per month?

- Count number of records per "appointment_month" using "size" function and group results by "appointment_month" using "groupby" function.

<u>Visualise and identify initial trends</u>

Objective 1: Create three visualisations indicating the number of appointments per month for service settings, context types, and national categories.

- Import "Seaborn" and "Matplotlib" libraries.
- Preset figure sizes and plot sizes to "15:12" and "white" respectively using using "sns" function.
- Create subset of "nc" data with columns "service_setting", "context_type", "national_category", "count_of_appointments" and "appointment_month".
- Change data type of "appointment_month" to a string to allow for easier plotting and confirm change by querying metadata.
- Sum "count_of_appointments" using "sum" function and group results by "service_setting" and "appointment_month" using "groupby" function.
- Create a lineplot off all unique "service_setting" values based on "count_of_appointments" per "appointment_month". Repeat this step for "context_type" and "national_category".

Objective 2: Create visualisations indicating the number of appointments for service setting per season. The seasons are summer (August 2021), autumn (October 2021), winter (January 2022), and spring (April 2022).

- Create subset of "nc" with columns service_setting, count_of_appointments and appointment_month.
- Filter out months that did not meet the criteria of the objective by creating an index and applying the "loc" function to filter out unwanted data.
- Sum "count_of_appointments" using "sum" function and group results by "service_setting" and "appointment_month" using "groupby" function.
- Create a lineplot off all unique "service_setting" values based on "count_of_appointments" per "appointment_month".

<u>Analyse the Twitter data</u>

- Import "Seaborn" and "Pandas" libraries.
- Import tweets (csv) and create data frame with alias "tw"

- Preset figure sizes and plot sizes to "15:12" and "white" respectively using using "sns" function.
- Preset column width to "200" using "options display" function.
- Check for missing values using "isnull" function.
- Verify metadata using "info" function.
- Check descriptive data using "describe" function.
- Create subset of "tw" data with column "tweet_full_text".
- Create a variable and assign an empty list to it.
- Loop through "tw_text" and create a series of values containing # symbol. These values will be housed in "tags".
- View the first 30 records of "tags" series by calling up default python indices (0:30).
- Convert series to data frame using "pd.dataframe" function and name columns.
- Count number of times each hashtag has been tweeted using "count" function and group result by "tweets" using "groupby" function.
- Check "tweets_data" metadata to ensure count is integer.
- Filter for hashtags tweeted more than 10 times in "tweets_data" by creating an index and calling it up.
- Create bar plot of tweets versus counts for "tweets_10" data frame by referencing "tweets" and "counts".
- Create subset of "tweets_10" data frame by creating a copy of "tweets_10" using the "copy" function then filtering out overrepresented hashtags using an index.
- Create cleaner bar plot of popular hashtags using from "overrep_tweets" data frame by referencing "tweets" and "counts".

Make recommendations

Question 1: Should the NHS start looking at increasing staff levels

- Create a subset of "ar" with columns "appointment_month", "appointment_status", "hcp_type", "appointment_mode", "time_between_book_and_appointment" and "count_of_appointments".
- Filter out any dates less than "2021-08" using the "loc" function.
- Sum "count_of_appointments" by "aggregate" function and group results by "appointment_month" using "groupby" function.
- Calculate average appointments per day by dividing monthly "count_of_appointments" by 30 using "lambda" function and house result in new column called "average_utilisation".
- Convert "appointment_month" column to string for ease of visualisation.
- Create lineplot on "ar_agg" of "appointment_month" versus "average_utilisation".

Question 2: How do the healthcare professional types differ over time?

- Create a lineplot on "ar_agg" of "appointment_month" versus "count_of_appointments" for all "hcp_type" values. All results can be represented on one plot using "ISIN" function.

Question 3: Are there significant changes in whether or not visits are attended?

- Create a on lineplot "ar_agg data" "appointment_month" versus "count_of_appointments" for "Attended" values of "appointment_mode" column only.

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

- Identify the 3 busiest months by using "sort_values" function on "count_of_appointments" and sort in descending order.
- Create lineplot on "ar_agg" of "count_of_appointments" per "appointment_month" for "appointment_mode".

Question 5: Are there any trends in time between booking an appointment?

- Create lineplot on "ar_agg" of "count_of_appointments" per appointment_month for "time_between_book_and_appointment".

Question 6: How do the various service settings compare?

- Create data frame by summing "count_of_appointments" and grouping by "service_setting" and "appointment_month".
- Create a boxplot to visualise "count_of_appointments" against "service_setting".
- Create copy of data frame above and filter out "General Practice" value in "service_setting" column by using an index than can be referenced.
- Create a new boxplot to visualise "count_of_appointments" against "service_setting" excluding "General Practice" value.