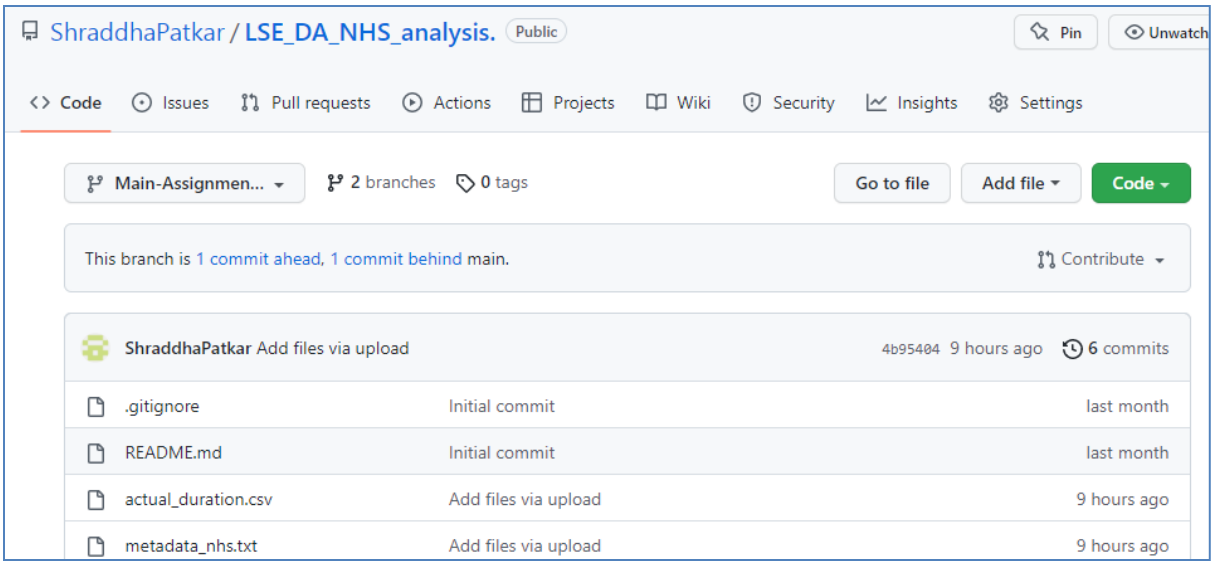
1.Prepare your GitHub repository

Link:

https://github.com/ShraddhaPatkar/LSE_DA_NHS_analysis./tree/Main-Assignment-Supporting-files



This branch is 1 commit ahead, 1 commit behind main.

ShraddhaPatkar Add files via upload 4b95404 9 hours ago 6 commits

.gitignore	Initial commit	last month
README.md	Initial commit	last month
actual_duration.csv	Add files via upload	9 hours ago
metadata_nhs.txt	Add files via upload	9 hours ago

2. Background/context of the business:

National Health Services (NHS), is a publicly funded healthcare system in England. The NHS incurs significant, potentially avoidable, costs when patients miss general practitioner (GP) appointments. The reasons for missed appointments need to be better understood.

2.1 Business Problem

The reasons for missed appointments need to be better understood also the government needs a data-informed approach for deciding how best to handle this problem.

2.2 Focus of this Data analytics project

- Reducing or eliminating missed appointments for social and financial benefits
- Has there been adequate staff and capacity in the networks?
- What was the actual utilisation of resources?

2.3 Additional Questions or analytics worth exploring

- 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 Twitter related to healthcare in the UK?

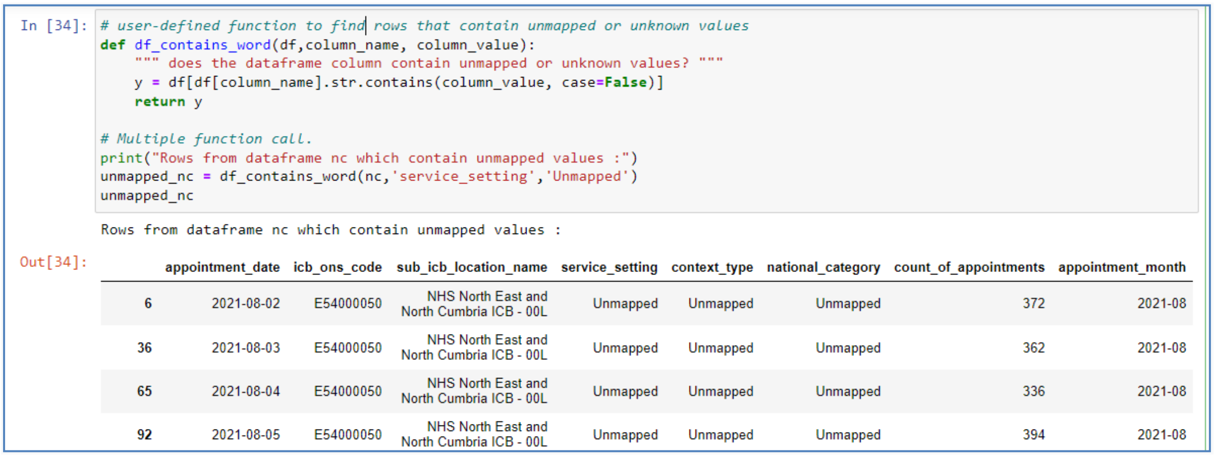
2.4 Questions for client

- There are some unmapped and unknown values, can that data be made available, it should be ignored or can be replaced by other values
- Impact on various factors based on whether or not visits are attended?

3 Analytical approach (Data Exploration)

3.1 Import data and explore missing values, erroneous data, wrong data types

- Imported the necessary libraries (e.g. Pandas and Numpy) for functions and seaborn, matplotlip for visualisation etc.
- Imported csv and excel files into data frames with proper naming conventions
- I preferred .info () method as it gives almost full summary of dataframe like colums, rows, null values, datatypes etc.
- I also used isnull () method as another approach to confirm on null values. And isna to check missing values.
- What I observed is that there are no straight forward missing values but different data frame has unknown unmapped values in different columns, to handle this created user defined function which can find/segregate any unmapped/unknown values from any data frame column. Also portrayed lambda function as another technical alternative

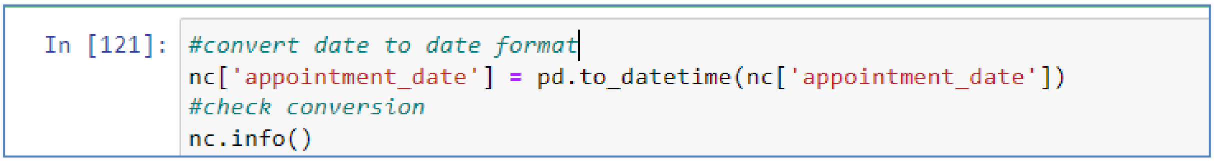


```
In [34]: # user-defined function to find rows that contain unmapped or unknown values
         def df_contains_word(df,column_name, column_value):
             """ does the dataframe column contain unmapped or unknown values? """
             y = df[df[column_name].str.contains(column_value, case=False)]
             return y

         # Multiple function call.
         print("Rows from dataframe nc which contain unmapped values :")
         unmapped_nc = df_contains_word(nc,'service_setting','Unmapped')
         unmapped_nc

         Rows from dataframe nc which contain unmapped values :
Out[34]:
```

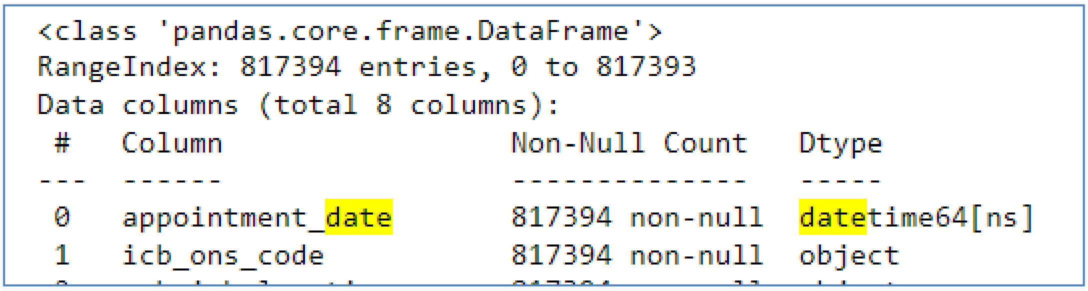
	appointment_date	icb_ons_code	sub_icb_location_name	service_setting	context_type	national_category	count_of_appointments	appointment_month
6	2021-08-02	E54000050	NHS North East and North Cumbria ICB - 00L	Unmapped	Unmapped	Unmapped	372	2021-08
36	2021-08-03	E54000050	NHS North East and North Cumbria ICB - 00L	Unmapped	Unmapped	Unmapped	362	2021-08
65	2021-08-04	E54000050	NHS North East and North Cumbria ICB - 00L	Unmapped	Unmapped	Unmapped	336	2021-08
92	2021-08-05	E54000050	NHS North East and North Cumbria ICB - 00L	Unmapped	Unmapped	Unmapped	394	2021-08

- Changed data type format as required

➢ Converted date to date format

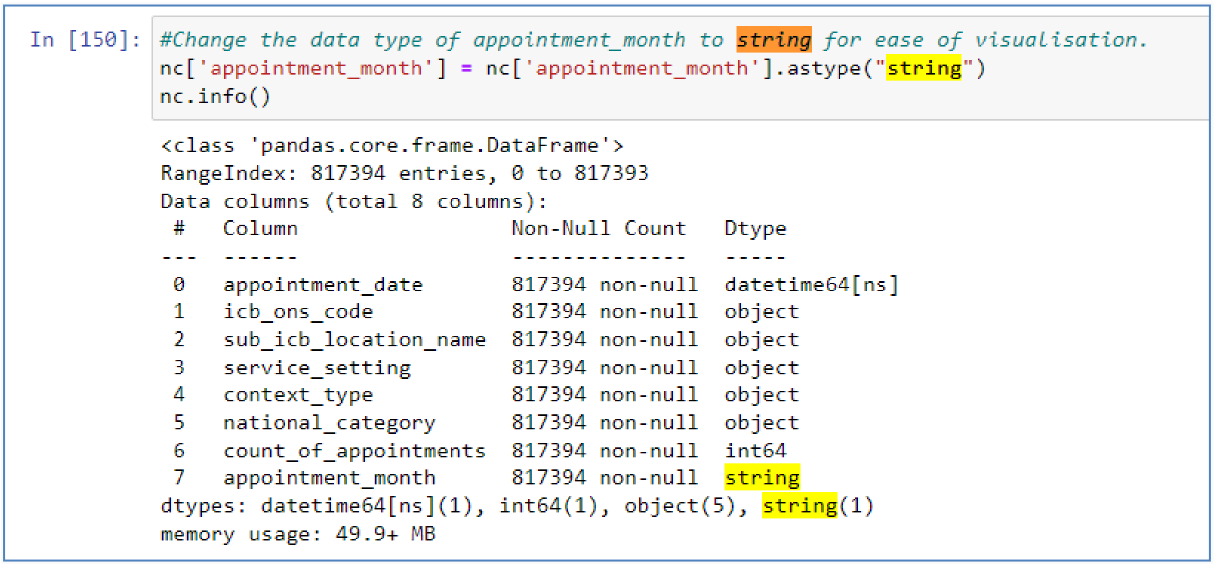


```
In [121]: #convert date to date format
          nc['appointment_date'] = pd.to_datetime(nc['appointment_date'])
          #check conversion
          nc.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 817394 entries, 0 to 817393
Data columns (total 8 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   appointment_date       817394 non-null  datetime64[ns]
 1   icb_ons_code           817394 non-null  object
```

➢ Changed data type of appointment month to string for ease of visualisation



```
In [150]: #Change the data type of appointment_month to string for ease of visualisation.
          nc['appointment_month'] = nc['appointment_month'].astype("string")
          nc.info()

          <class 'pandas.core.frame.DataFrame'>
          RangeIndex: 817394 entries, 0 to 817393
          Data columns (total 8 columns):
           #   Column                 Non-Null Count   Dtype
          ---  ------                 --------------   -----
           0   appointment_date       817394 non-null  datetime64[ns]
           1   icb_ons_code           817394 non-null  object
           2   sub_icb_location_name  817394 non-null  object
           3   service_setting        817394 non-null  object
           4   context_type           817394 non-null  object
           5   national_category      817394 non-null  object
           6   count_of_appointments  817394 non-null  int64
           7   appointment_month      817394 non-null  string
          dtypes: datetime64[ns](1), int64(1), object(5), string(1)
          memory usage: 49.9+ MB
```

3.2 Explored Outliers

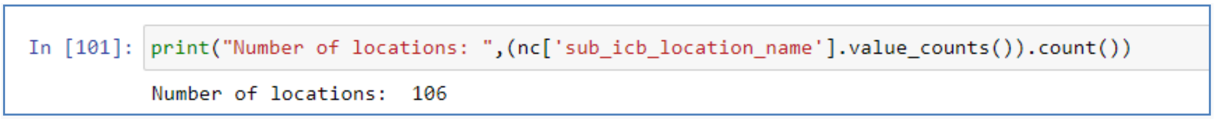
- Used data subsetting on national_categories related dataframe to explore which combination of columns can be checked for outliers.
- Using boxplot visualised count of appointments for General practice service setting, across different context types.
- Technically I filtered service settings column of data frame to concentrate on GP records and in box plot use hue to divide GP records across different context types.
- Also used set style to improve readability of box plot. (outlier functional observations detailed in presentation)

3.3 Exploratory analysis of data

Q1. What is the number of locations, service settings, context types, national categories, and appointment statuses in the data sets?

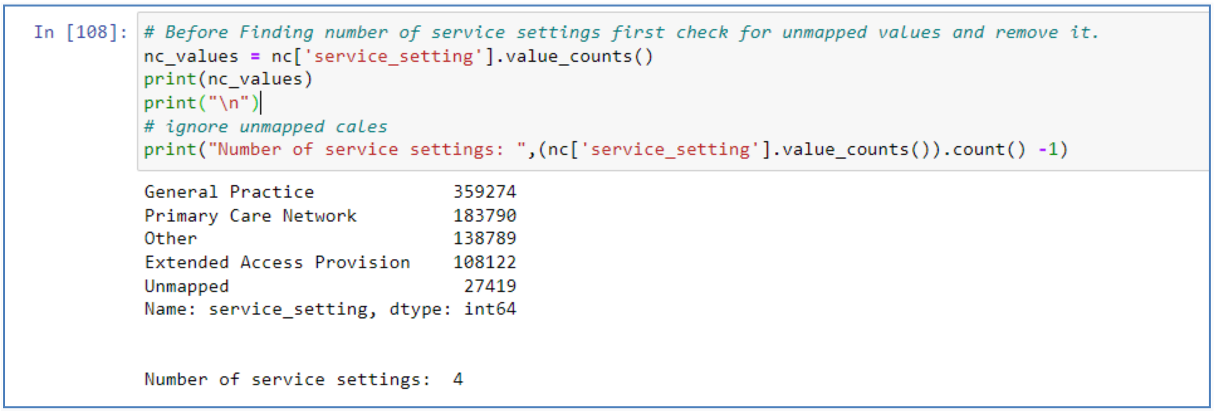
This is exploratory stage of analysis, and this question helps to explore what different combination of parameter values and counts that we need to analyse and explore, before solving final problem. This is just to explore data.

1. To get location count I simply used value_counts and count method.



```
In [101]: print("Number of locations: ",(nc['sub_icb_location_name'].value_counts()).count())

          Number of locations:  106
```

2. But since other columns have unused, unmapped values hence I used value counts to get view of unmapped rows and then ignored such records

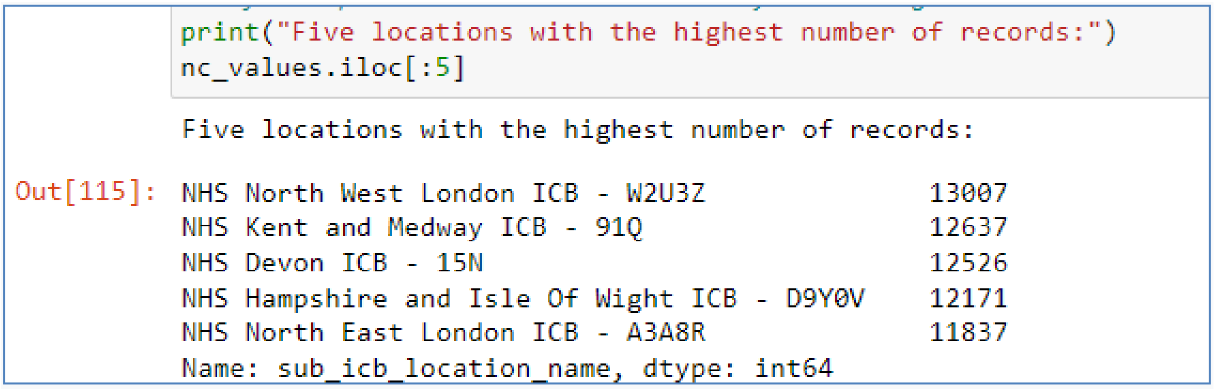


```
In [108]: # Before Finding number of service settings first check for unmapped values and remove it.
          nc_values = nc['service_setting'].value_counts()
          print(nc_values)
          print("\n")
          # ignore unmapped cales
          print("Number of service settings: ",(nc['service_setting'].value_counts()).count() -1)

          General Practice            359274
          Primary Care Network        183790
          Other                       138789
          Extended Access Provision   108122
          Unmapped                     27419
          Name: service_setting, dtype: int64


          Number of service settings:  4
```

Q2. What are the five locations with the highest number of records?

Again this question is for data exploration and not related to finalise trends.

This is answered already in above question; I just used iloc method to display first 5 records,



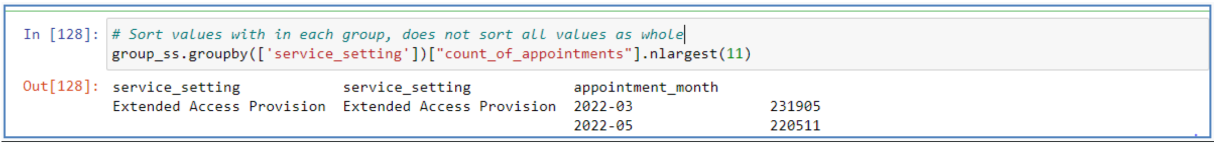
```
           print("Five locations with the highest number of records:")
           nc_values.iloc[:5]

           Five locations with the highest number of records:

Out[115]:  NHS North West London ICB - W2U3Z                 13007
           NHS Kent and Medway ICB - 91Q                     12637
           NHS Devon ICB - 15N                               12526
           NHS Hampshire and Isle Of Wight ICB - D9Y0V       12171
           NHS North East London ICB - A3A8R                 11837
           Name: sub_icb_location_name, dtype: int64
```

Q4. What is the date range of the provided data sets,(Between what dates were appointments scheduled?)

For this technically I used two syntaxes but single approach of min and max aggregation method , please refer In [241]

Q5. Which service settings reported the most appointments for a specific period?

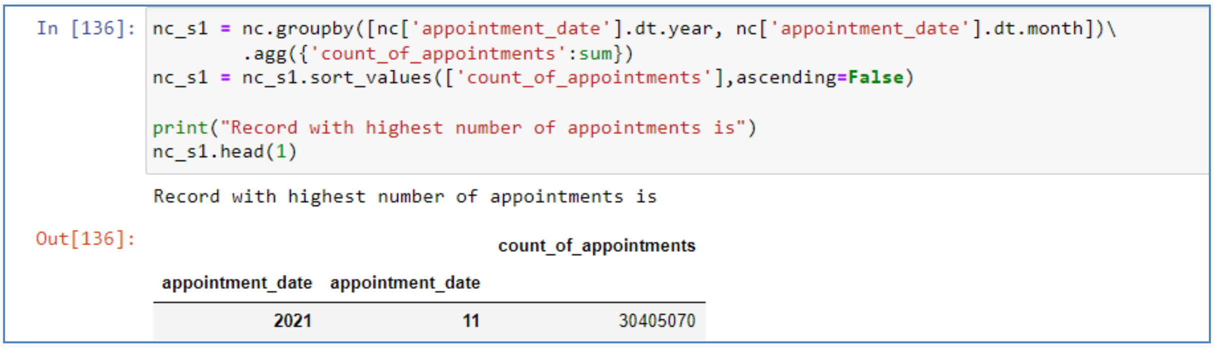
- This is achieved by using group by method to segregate records related to different months, per service settings
- Then used aggregate function sum and sort_values function to get service setting GP has most appointments during October and November 2021
- Again I used 2 technical approaches just sort values as demonstrated in In[127] and nlargest to sort values within groups



```
In [128]: # Sort values with in each group, does not sort all values as whole
          group_ss.groupby(['service_setting'])["count_of_appointments"].nlargest(11)

Out[128]: service_setting            service_setting            appointment_month
          Extended Access Provision  Extended Access Provision  2022-03              231905
                                                                2022-05              220511
```

Q6. Which month had the highest number of appointments?

For this dataframe is sorted just based on appoint date usig .dt.year and dt.month functions



```
In [136]: nc_s1 = nc.groupby([nc['appointment_date'].dt.year, nc['appointment_date'].dt.month])\
                .agg({'count_of_appointments':sum})
          nc_s1 = nc_s1.sort_values(['count_of_appointments'],ascending=False)

          print("Record with highest number of appointments is")
          nc_s1.head(1)

          Record with highest number of appointments is

Out[136]:
                                                     count_of_appointments
           appointment_date  appointment_date
                       2021                11                     30405070
```

Overall functional observations from data exploration are captured in presentation.

4 Visualisation and insights

4.1 Number of appointments per month for service settings, context types, and national categories.

- Changed the data type of appointment_month to string for ease of visualisation
  - Used aggregate and group by to get #sum of the appointments per month from nc dataframe
  - Selected lineplot with marker to compare trends across time
  - Plot details
  - ✓ X-axis parameter ='appointment_month'
  - ✓ Y-axis parameter='count_of_appointments'
  - ✓ hue='service_setting' / 'context types' / national categories
  - ✓ Ci=none (**confidence interval generated)**

Insights:

- ▪ General Practice service setting has maximum appointments per month
- ▪ Maximum appointments per month are related to context- care related encounter
- ▪ Maximum appointments are for General consultation routine

4.2 Create four 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)

- Divided visualisation section in four parts fig2,((top_left,top_right),(bottom_left,bottom_right))=plt.subplots(2,2,figsize=(10,8))
- Created lineplot for each season with marker and without legend used set style for graph readability

- Plot Details:
  x-axis ='appointment_month',
  y-axis='count_of_appointments',hue='service_setting',
  data= nc dataframe sorted by service setting and
  ['appointment_month'] filtered to track different seasons

4.3 Identify and review the top trending hashtags (#) related to healthcare in the UK based on the data set received from the NHS.

- Explored the twitter data set and confirmed import
- Created new DataFrame with only text values (Data subseting).
- Retrieved hashtags from tweets using for loop
- Identify the top trending hashtags with a visualisation
- ✓ Converted the new Series into a DataFrame
- ✓ Explored column renaming for better visibility
- ✓ Seaborn barplot created for all records with a count>10.
- ✓ Removed any overrepresented hashtags and plot by calculating mean (used value 20 for good visualisation)

5 .Patterns through visualisation

5.1.Should the NHS start looking at increasing staff levels?

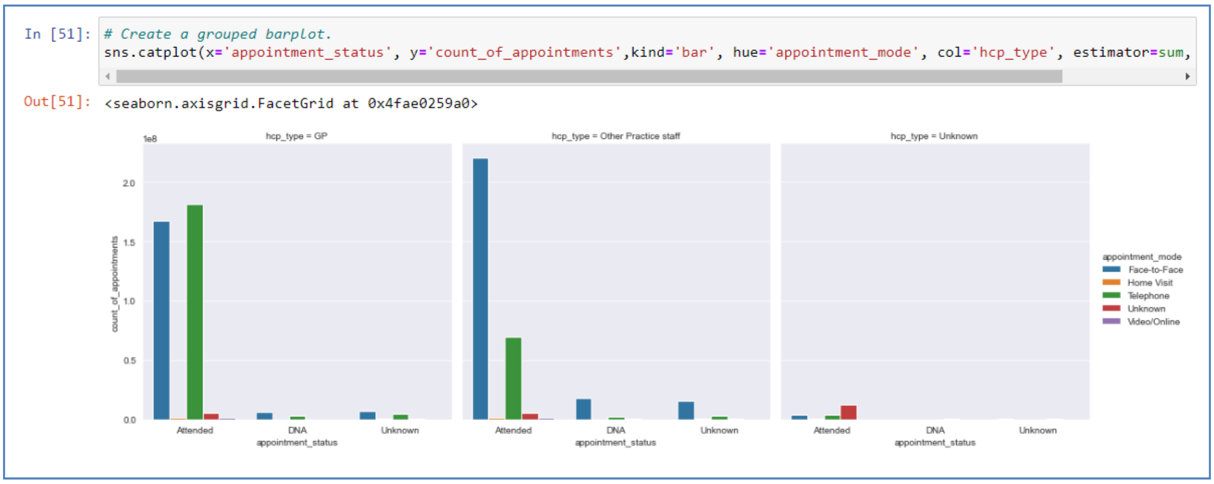
- ✓ calculated the count_of_appointments for different healthcare professionals permonth and rhen created barplot for 'appointment_month' >'2021-08'
- ✓ With (x='appointment_month', y='count_of_appointments', hue='hcp_type)
- ✓ Note: due to time constrain I have not added labels for all graphs, but have depicted learnings wherever possible
- ✓ Then percentage utilisation by booked appointments is compared with attended % (technical demostarations-merge,matplotlib,2 y graphs)---In 46-49

<u>Outcome</u>

- ✓ Mostly monthly appointments for GP are more except during October 2021
- ✓ There is small chunk for which HCP is not known
- ✓ NHS staff is not fully utilised

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

This is explored through cat plot where columns are based on HCP Type



```
In [51]: # Create a grouped barplot.
         sns.catplot(x='appointment_status', y='count_of_appointments',kind='bar', hue='appointment_mode', col='hcp_type', estimator=sum,
```

```
Out[51]: <seaborn.axisgrid.FacetGrid at 0x4fae0259a0>
```

Outcome:

Face to face attended appointments are more for other practice staff and for GP more telephonic appointments attended

Busiest month exploration

Created grouped bar plot for 10 and 11 2021 and outcome is face to face appointments are more during these months

<u>Relation between actual duration and count of appointment</u>

When appointment duration is less there are more count of appointments (explored using scatterplot for observations on single day 25/12/2021)

<u>Relation between time_between_book_and_appointment and count of appointments</u>

When Time from booking to appointment is less count of appointments is more but not true for all cases