Loading Libraries

```
In [1]: # Importing libraries
        import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
```

Loading Dataset

```
In [2]: # Load the dataset
        data = pd.read_csv("Hospital.csv")
```

Display First 5 Rows

```
In [ ]: # Display first 5 rows
        data.head()
```

Out[ ]:

	OrganisationID	OrganisationCode	OrganisationType	SubType	Sector	Organisati
0	17970	NDA07	Hospital	Hospital	Independent Sector	
1	17981	NDA18	Hospital	Hospital	Independent Sector	
2	18102	NLT02	Hospital	Hospital	NHS Sector	
3	18138	NMP01	Hospital	Hospital	Independent Sector	
4	18142	NMV01	Hospital	Hospital	Independent Sector	

5 rows × 22 columns

Display Last 5 Rows

```
In [ ]: # Display last 5 rows
        data.tail()
```

Out[ ]:

	OrganisationID	OrganisationCode	OrganisationType	SubType	Sector	Orga
1206	10956142	U7P1U	Hospital	UNKNOWN	Independent Sector	
1207	10956143	B6Q2K	Hospital	UNKNOWN	Independent Sector	
1208	10956150	K5E9C	Hospital	UNKNOWN	Independent Sector	
1209	10956151	L4S0G	Hospital	UNKNOWN	Independent Sector	
1210	10956153	P3P8S	Hospital	UNKNOWN	Independent Sector	

5 rows × 22 columns

Display Data Types Of Columns

```
In [ ]: # Display data types of columns
        data.dtypes
```

```
Out[ ]:                          0
           OrganisationID      int64
         OrganisationCode     object
         OrganisationType     object
                  SubType     object
                   Sector     object
       OrganisationStatus     object
           IsPimsManaged       bool
         OrganisationName     object
                 Address1     object
                 Address2     object
                 Address3     object
                     City     object
                   County     object
                 Postcode     object
                 Latitude    float64
                Longitude    float64
           ParentODSCode      object
               ParentName     object
                    Phone     object
                    Email     object
                  Website     object
                   Fax,,,     object

         dtype: object
```

Display Five Point Summary

```
In [ ]: # Five point summary
        data.describe(include='all')
```

Out[ ]:

	OrganisationID	OrganisationCode	OrganisationType	SubType	Sector	Organisati
count	1.211000e+03	1211	1211	1211	1211	
unique	NaN	1211	1	3	2	
top	NaN	NDA07	Hospital	Hospital	NHS Sector	
freq	NaN	1	1211	961	743	
mean	1.375611e+06	NaN	NaN	NaN	NaN	
std	3.024986e+06	NaN	NaN	NaN	NaN	
min	1.797000e+04	NaN	NaN	NaN	NaN	
25%	4.064900e+04	NaN	NaN	NaN	NaN	
50%	4.311000e+04	NaN	NaN	NaN	NaN	
75%	7.610700e+04	NaN	NaN	NaN	NaN	
max	1.095615e+07	NaN	NaN	NaN	NaN	

11 rows × 22 columns

Removing Outlier

```
In [3]: from scipy.stats import zscore
        z_scores = data.select_dtypes(include=['number']).apply(zscore)
        data_no_outliers = data[(z_scores < 3).all(axis=1)]
```

Display Total Number Of Rows & Columns

```
In [ ]: # Display total number of rows and columns
        data.shape
```

```
Out[ ]: (1211, 22)
```

Finding Duplicates Values

```
In [ ]: # Finding duplicate values count
        data.duplicated().sum()
```

```
Out[ ]: 0
```

Finding Missing Values

```
In [ ]: # Finding missing values summary
        data.isnull().sum()
```

```
Out[ ]:                          0
              OrganisationID     0
            OrganisationCode     0
            OrganisationType     0
                     SubType     0
                      Sector     0
          OrganisationStatus     0
              IsPimsManaged      0
            OrganisationName     0
                    Address1   328
                    Address2   484
                    Address3  1064
                        City    15
                      County   238
                    Postcode     1
                    Latitude     2
                   Longitude     2
               ParentODSCode     0
                  ParentName     0
                       Phone   250
                       Email   789
                     Website   358
                      Fax,,,     2

        dtype: int64
```

Dropping Missing Values

```
In [ ]: # Dropping Missing Values
        data = data.dropna(axis=1)
```

Finding Uniques Values For Each Columns

```
In [ ]: # Finding unique values for each column
        data.nunique()
```

```
Out[ ]:                       0
              OrganisationID  1211
            OrganisationCode  1211
            OrganisationType     1
                     SubType     3
                      Sector     2
          OrganisationStatus     1
              IsPimsManaged      2
            OrganisationName  1189
               ParentODSCode   307
                  ParentName   307

        dtype: int64
```

Finding object columns

```
In [ ]: # Finding object columns
        object_cols = data.select_dtypes(include=['object']).columns.tolist()
        object_cols
```

```
Out[ ]:  ['OrganisationCode',
          'OrganisationType',
          'SubType',
          'Sector',
          'OrganisationStatus',
          'OrganisationName',
          'ParentODSCode',
          'ParentName']
```

Finding numerical columns

```
In [ ]: # Finding numerical columns
        numerical_cols = data.select_dtypes(include=['number']).columns.tolist()
        numerical_cols
```

```
Out[ ]:  ['OrganisationID']
```

Calculate & Display the Correlation Matrix for t Numeric Columns

In [ ]:

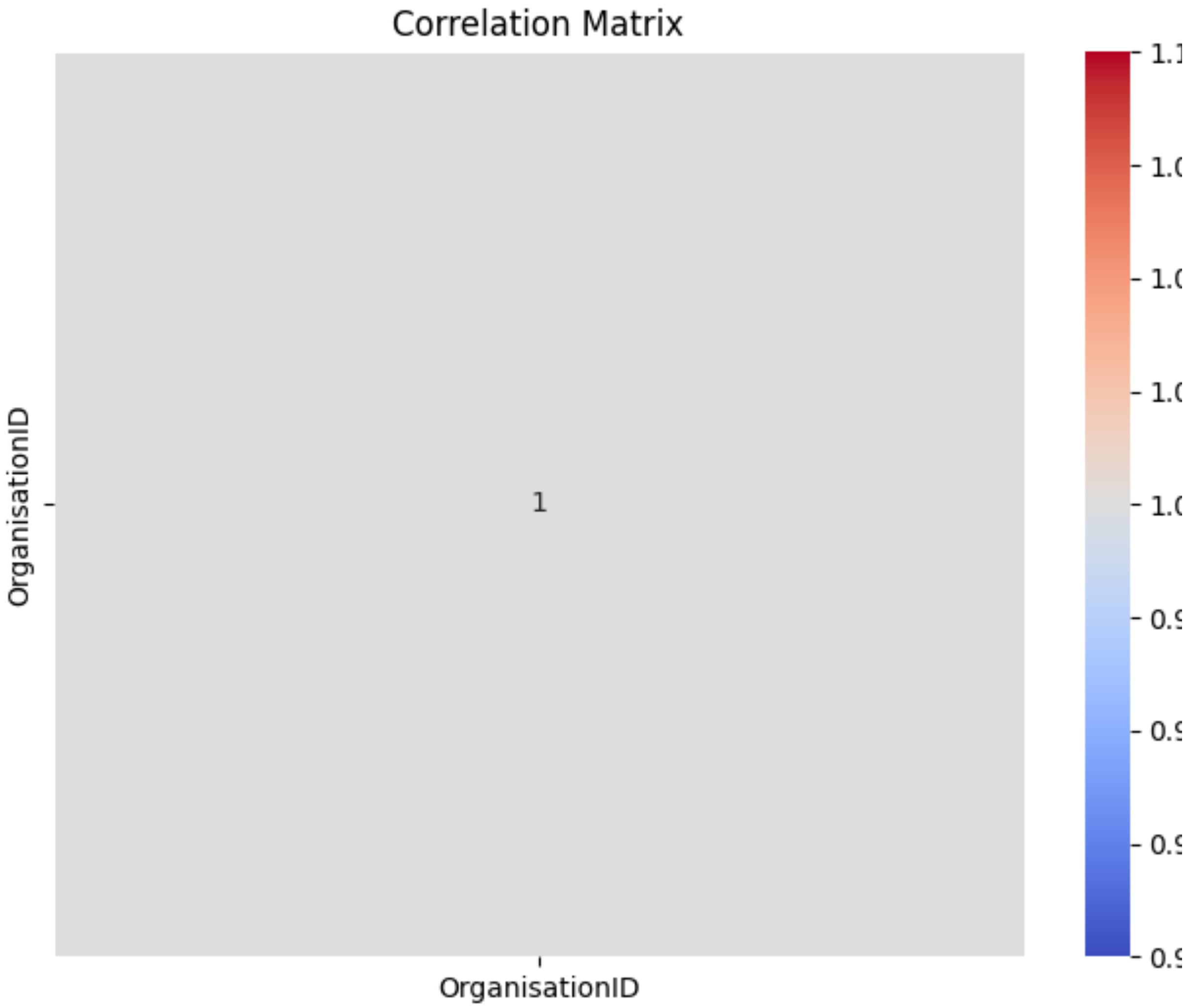
```
# Step 1: Select only numeric columns
numeric_cols = data.select_dtypes(include=['number']).columns

# Step 2: Compute the correlation matrix for numeric columns
correlation_matrix = data[numeric_cols].corr()

# Step 3: Display the correlation matrix
print(correlation_matrix)
```

```
                OrganisationID
OrganisationID             1.0
```

Display Heatmap for Numeric Columns

In [ ]:

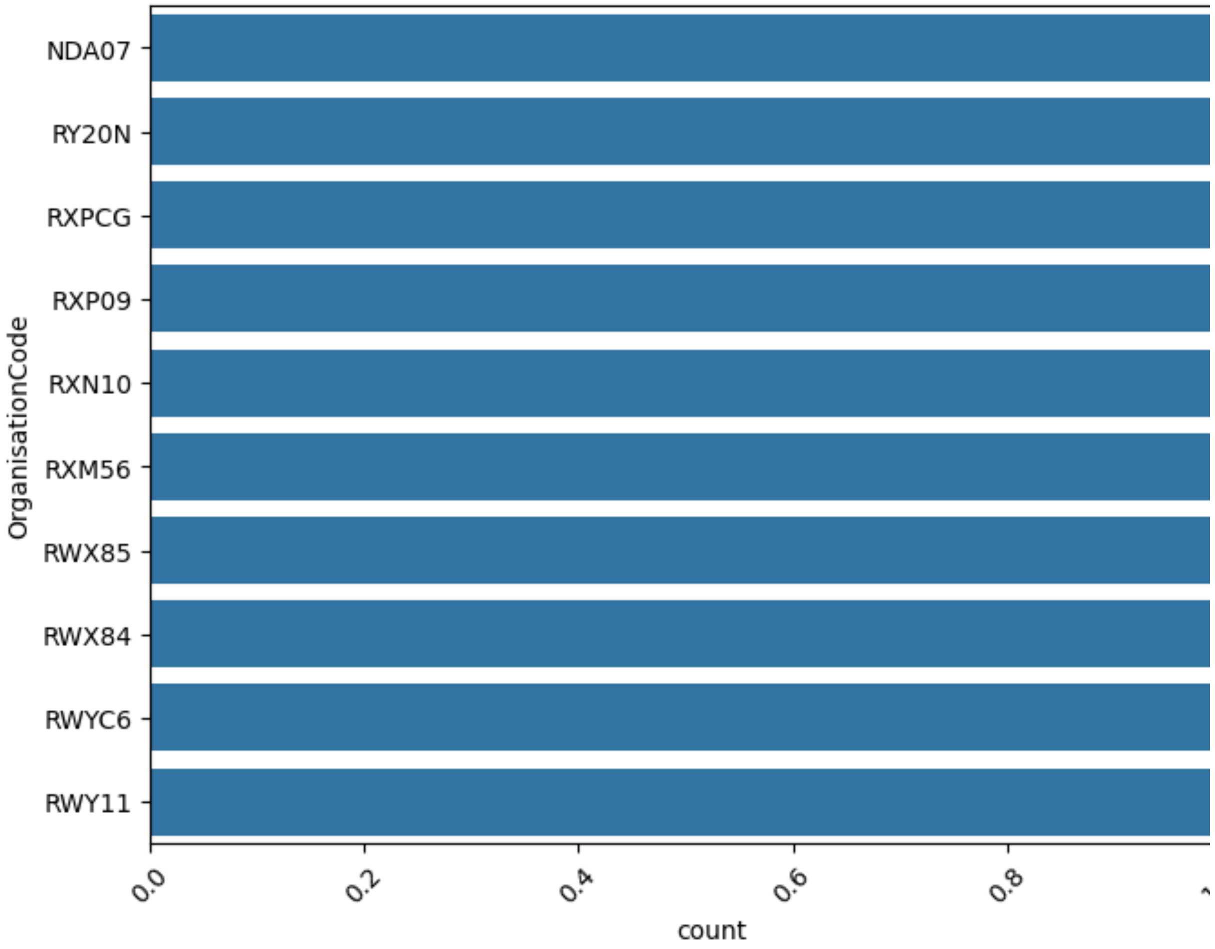
```
# Select only numeric columns before calculating correlations
numeric_data = data.select_dtypes(include=['number'])

# Calculate the correlation matrix
corr_matrix = numeric_data.corr()

# Plot the correlation matrix
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Matrix")
plt.show()
```

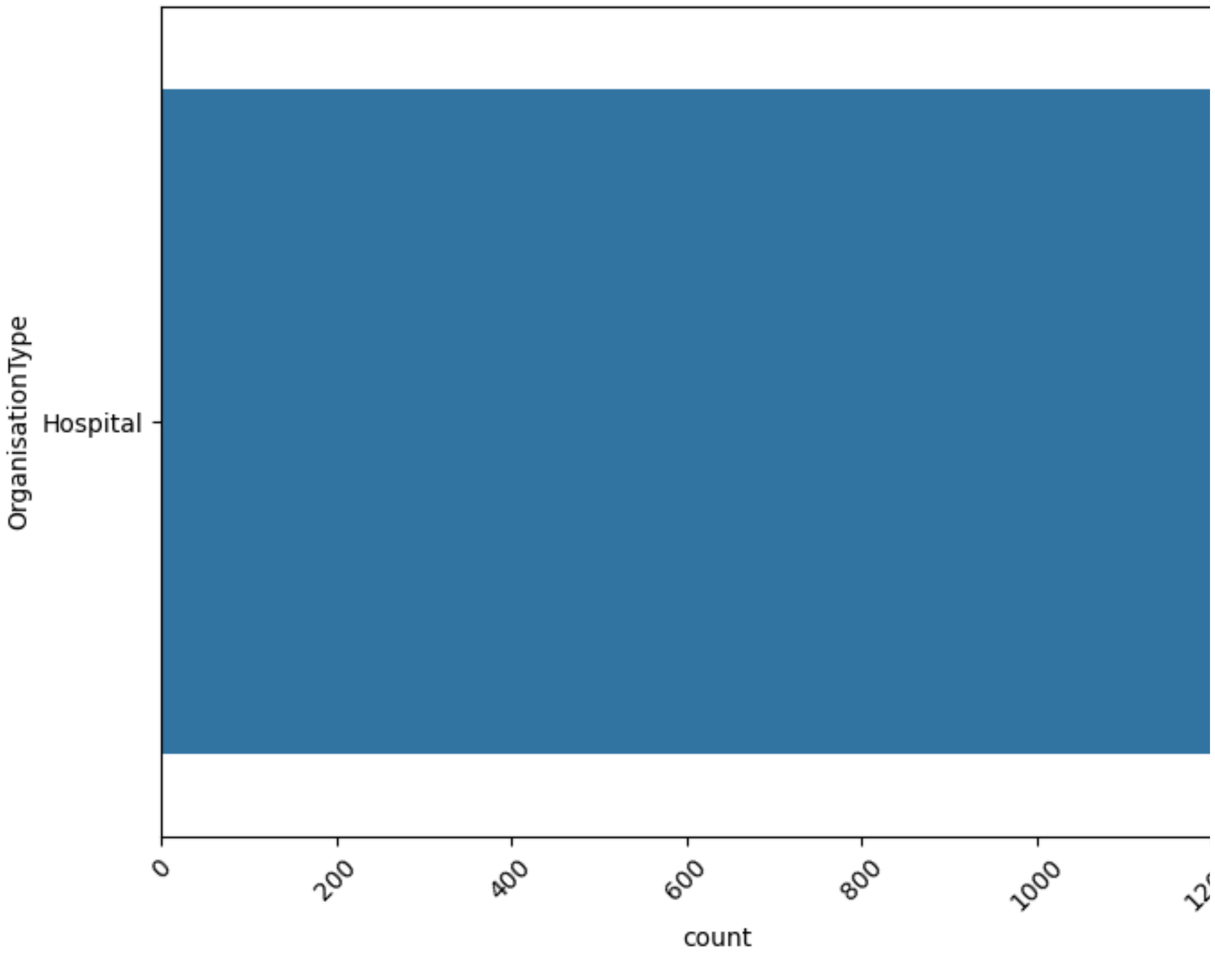


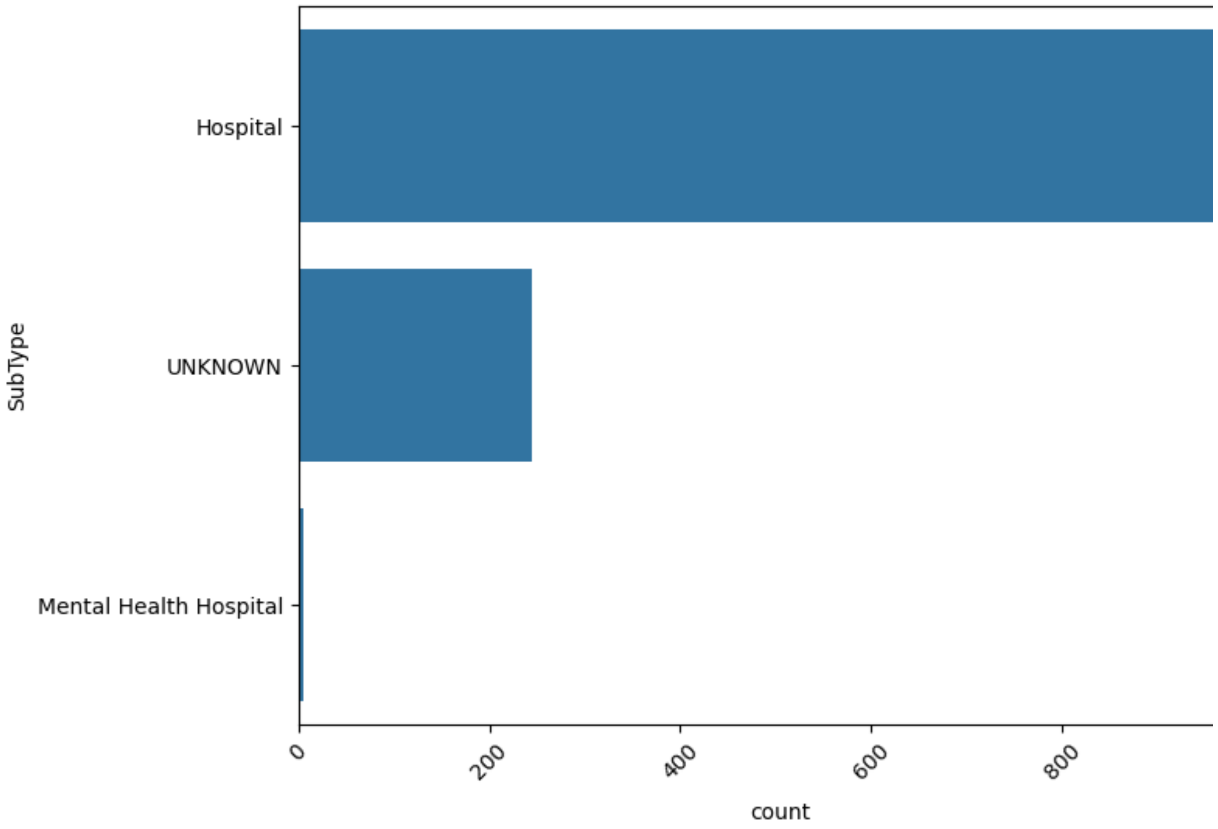
Count Plots For Each Object Type Columns

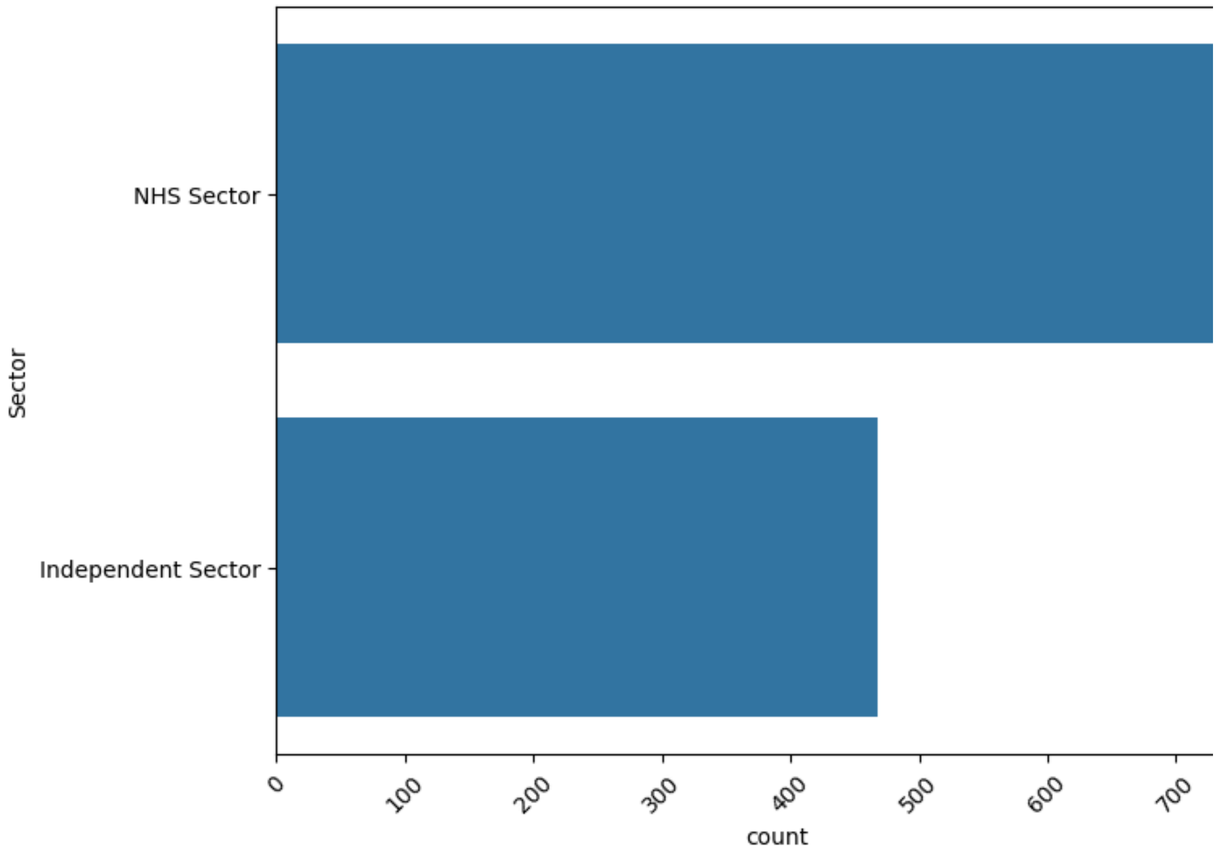
In [ ]:

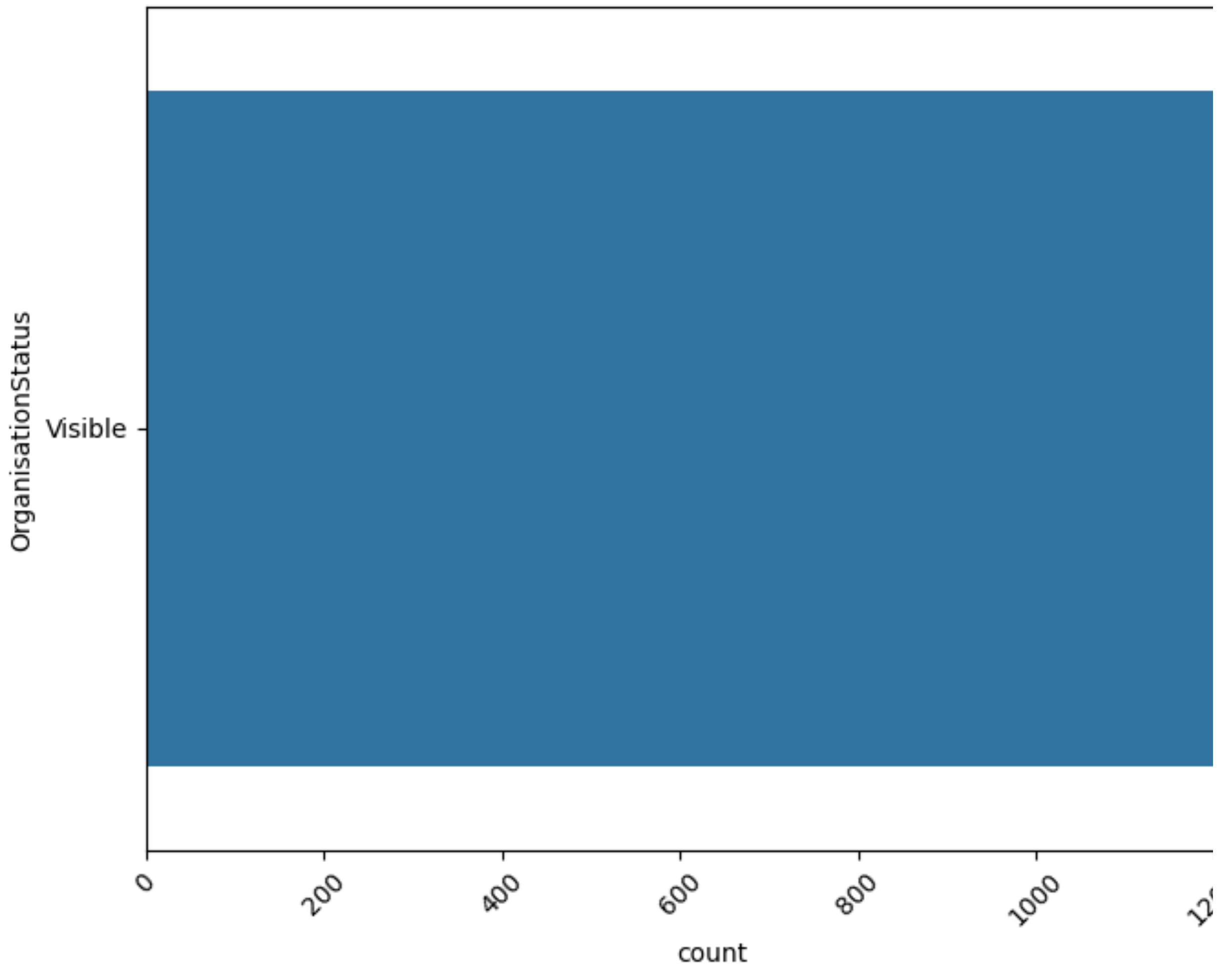
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'data' is your DataFrame and 'object_columns' is a list of cate
for col in object_columns:
    plt.figure(figsize=(8, 6))  # Increase plot size
    top_categories = data[col].value_counts().nlargest(10).index  # Limit
    sns.countplot(y=col, data=data, order=top_categories)  # Use order fo
    plt.xticks(rotation=45)  # Rotate x-axis labels for better readabilit
    plt.show()
```

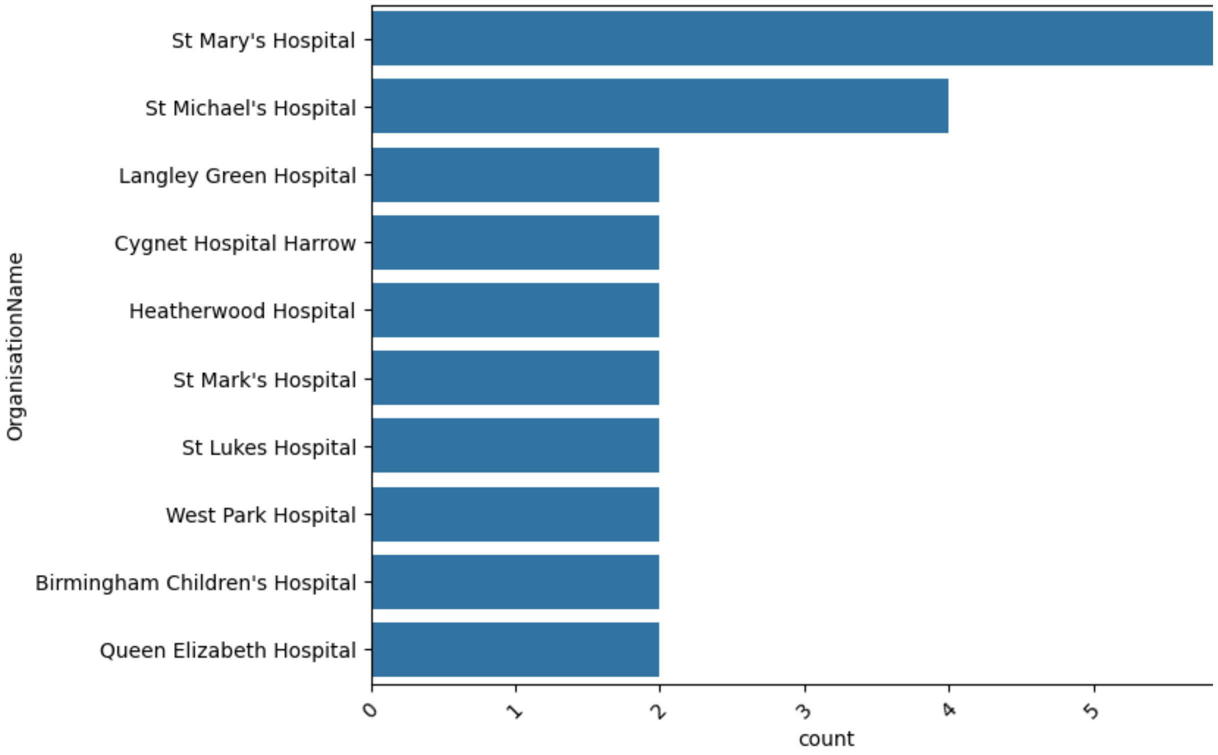

OrganisationCode
NDA07
RY20N
RXPCG
RXP09
RXN10
RXM56
RWX85
RWX84
RWYC6
RWY11
0.0
0.2
0.4
0.6
0.8
count

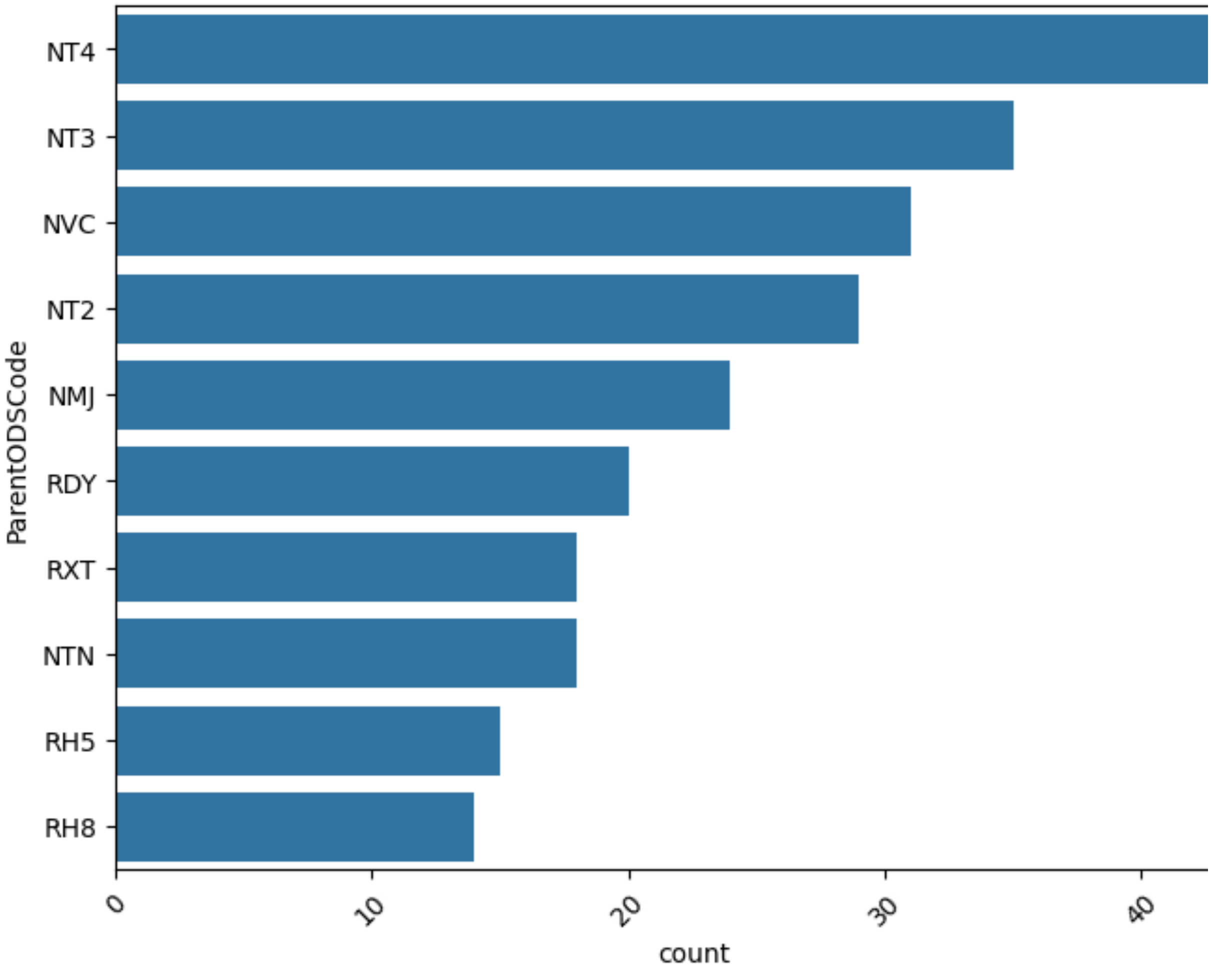

OrganisationType
Hospital
0
200
400
600
800
1000
count

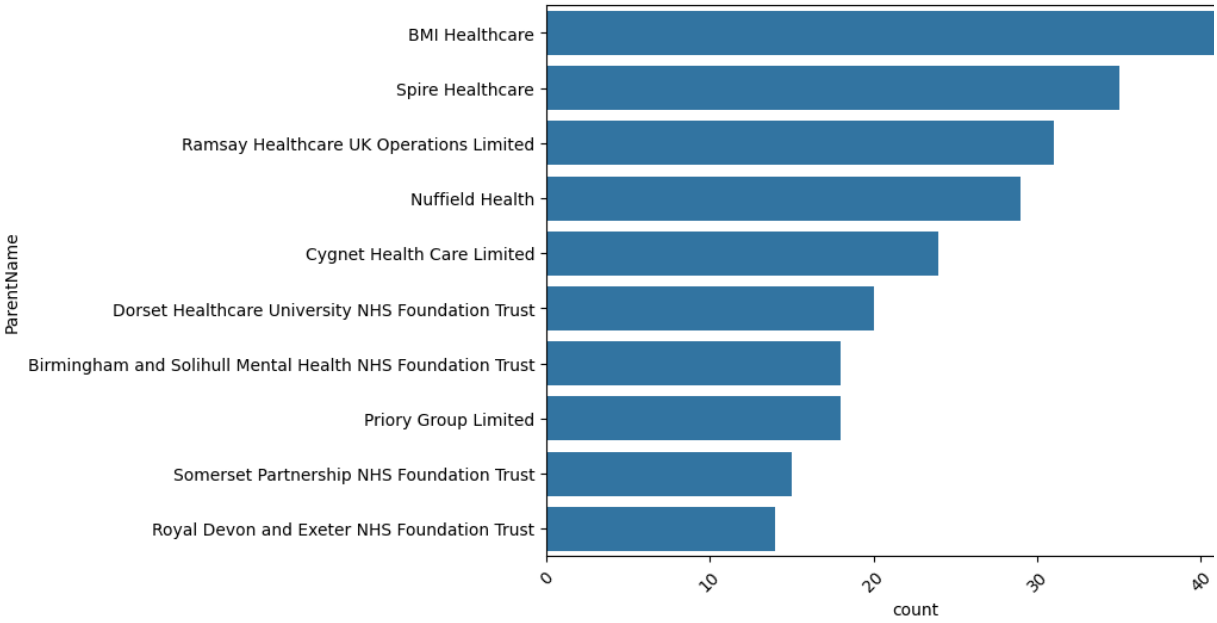

SubType
Hospital
UNKNOWN
Mental Health Hospital
0
200
400
600
800
count


Sector
NHS Sector
Independent Sector
0
100
200
300
400
500
600
700
count


OrganisationStatus
Visible
0
200
400
600
800
1000
12
count

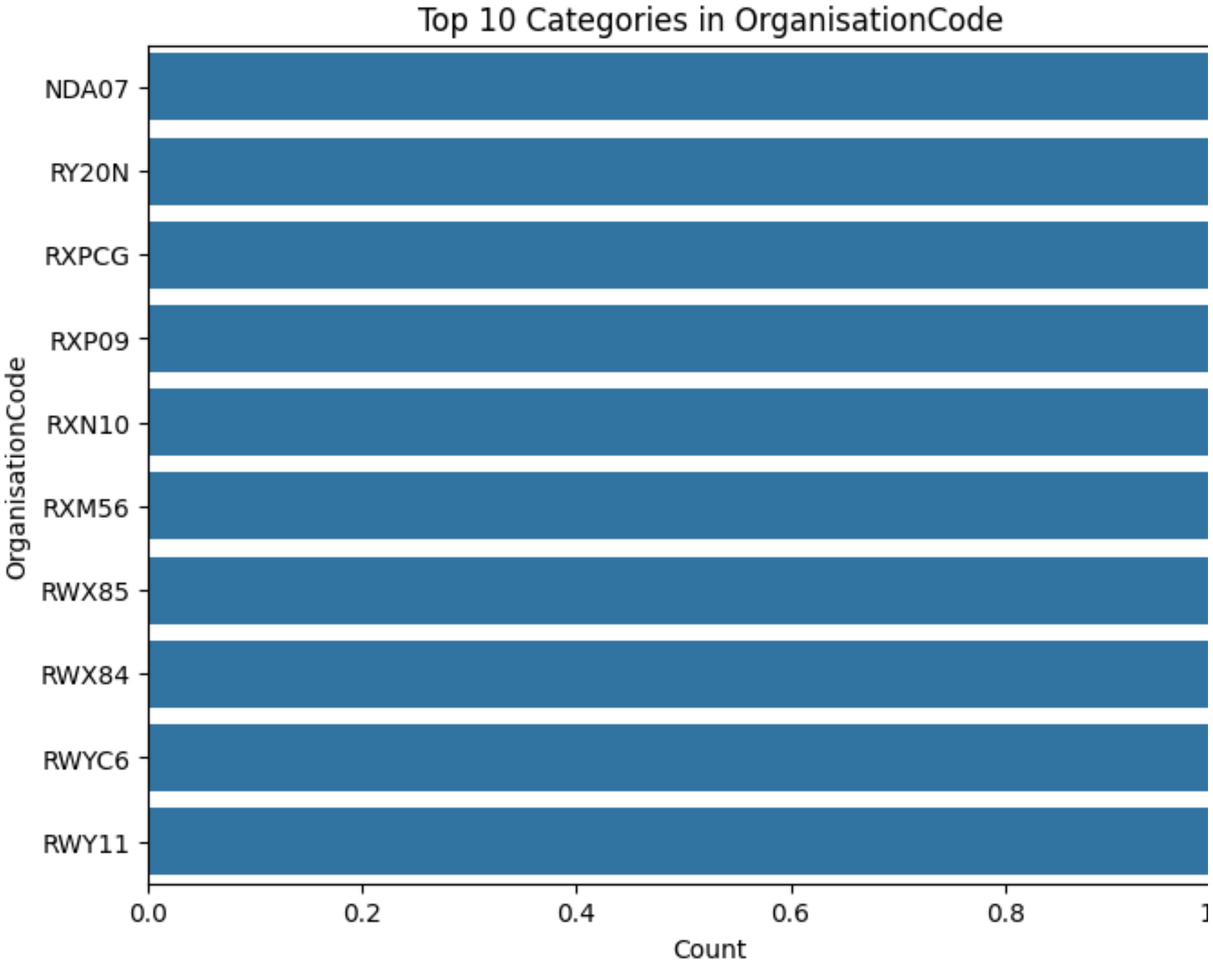

OrganisationName
St Mary's Hospital
St Michael's Hospital
Langley Green Hospital
Cygnet Hospital Harrow
Heatherwood Hospital
St Mark's Hospital
St Lukes Hospital
West Park Hospital
Birmingham Children's Hospital
Queen Elizabeth Hospital
0
1
2
3
4
5
count

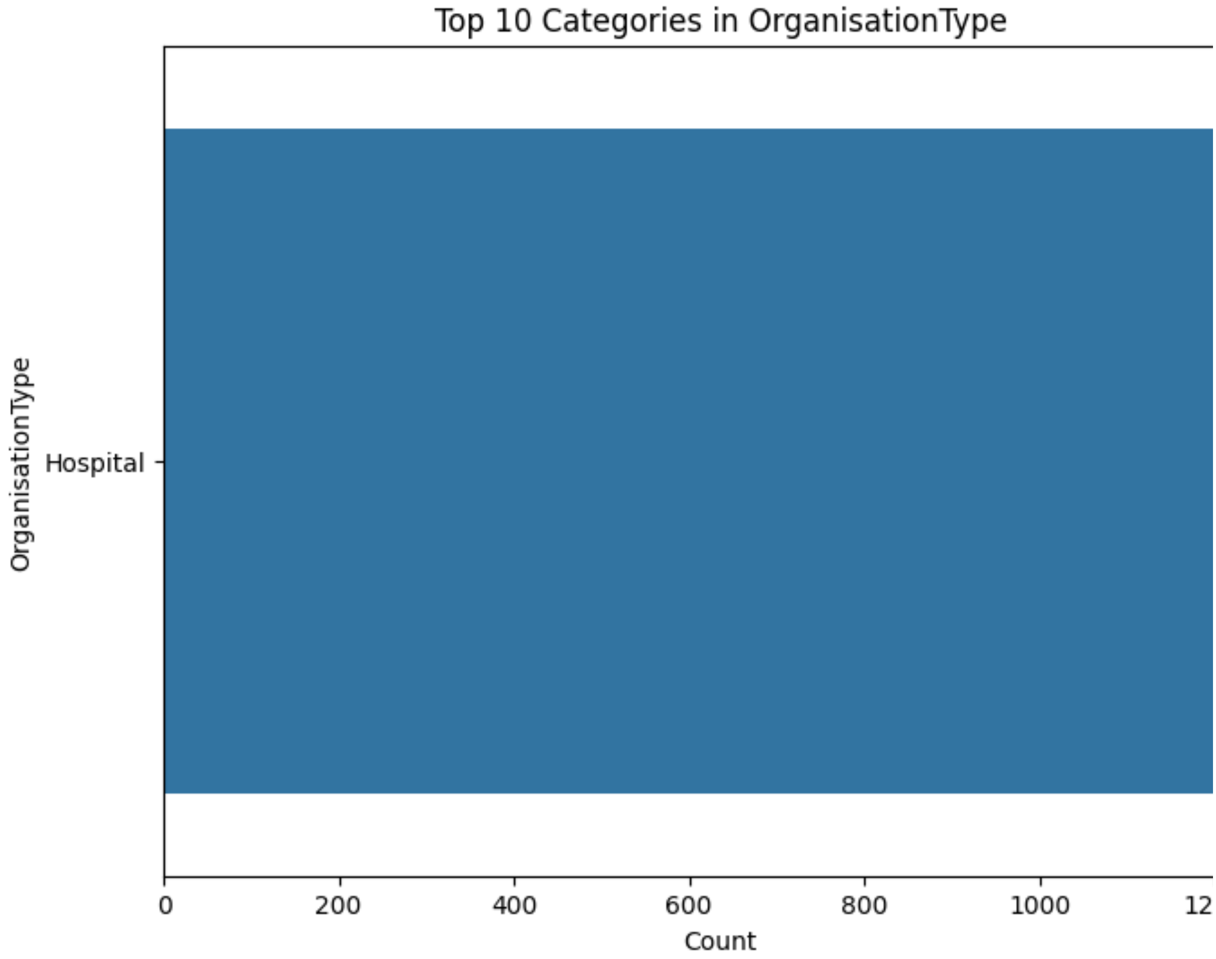


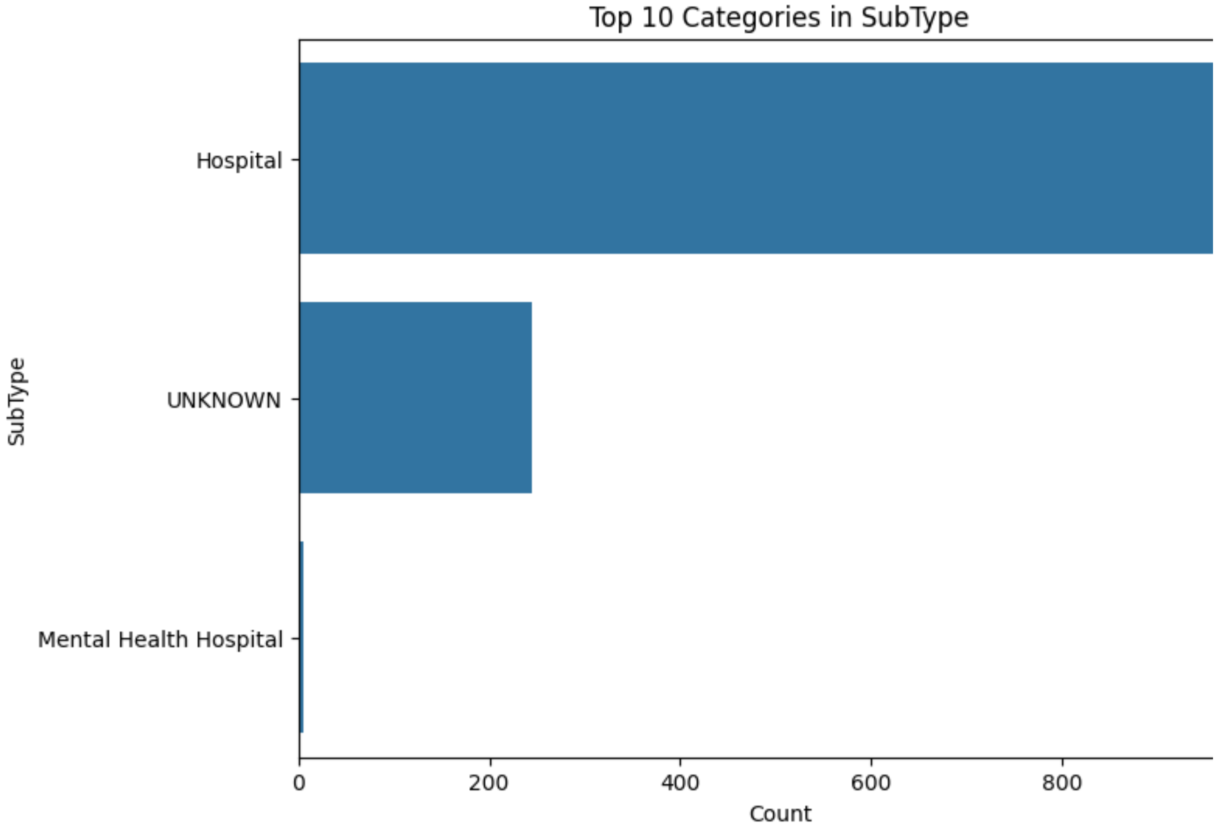


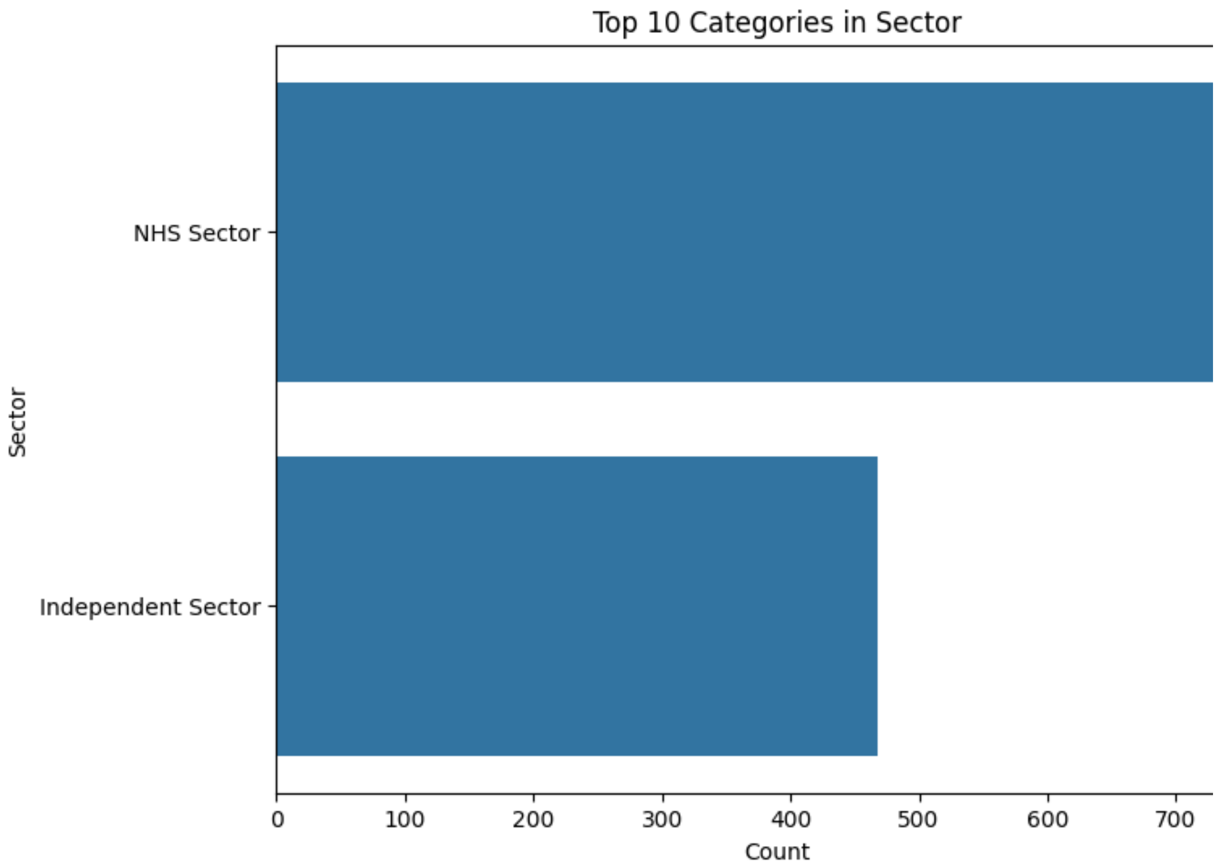
Object-Type BarPlot

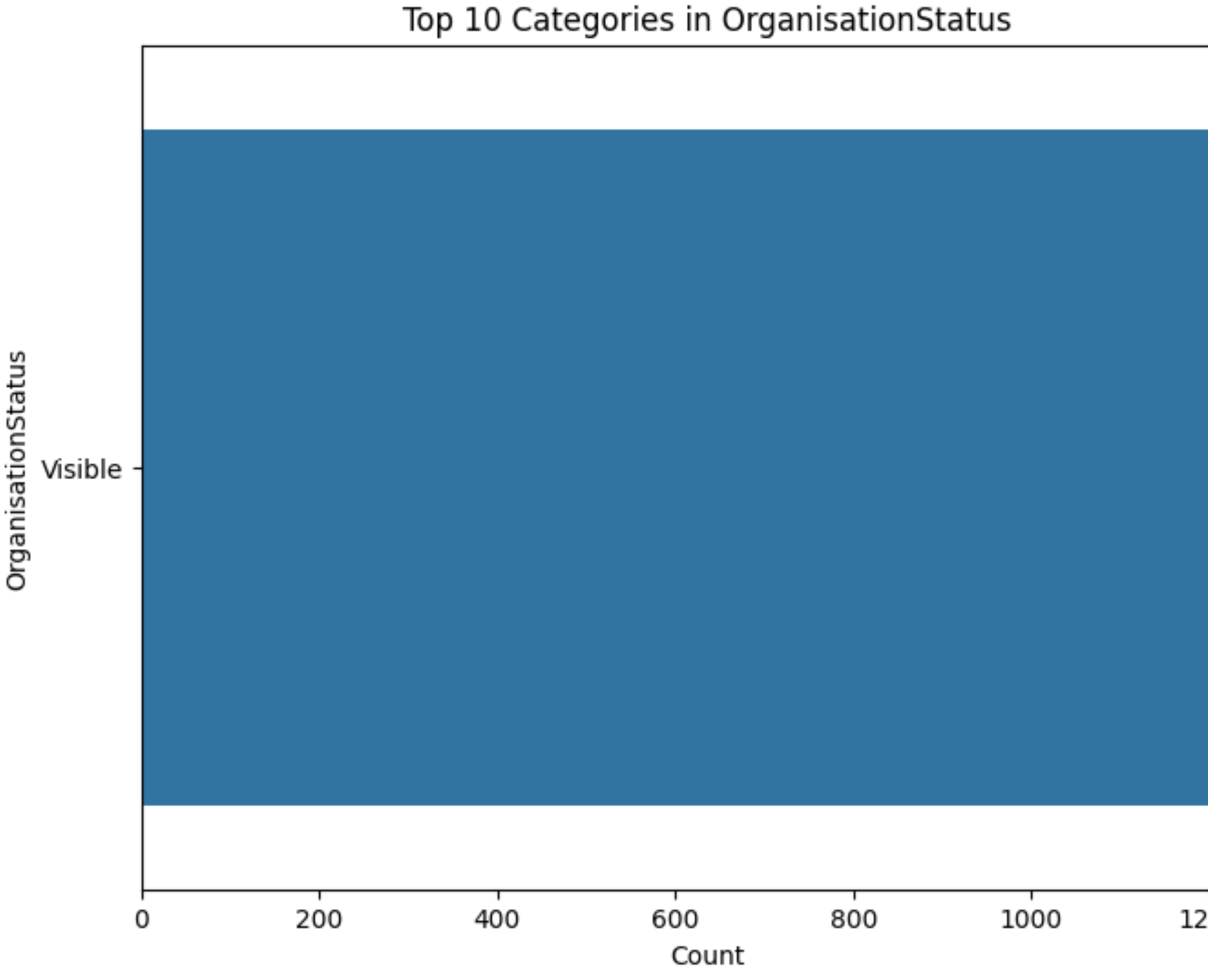
```
In [ ]: for col in object_columns:
            plt.figure(figsize=(8, 6))
            top_values = data[col].value_counts().head(10)  # Get the top 10 most
            sns.barplot(y=top_values.index, x=top_values.values)
            plt.title(f'Top 10 Categories in {col}')
            plt.xlabel('Count')
            plt.ylabel(col)
            plt.show()
```

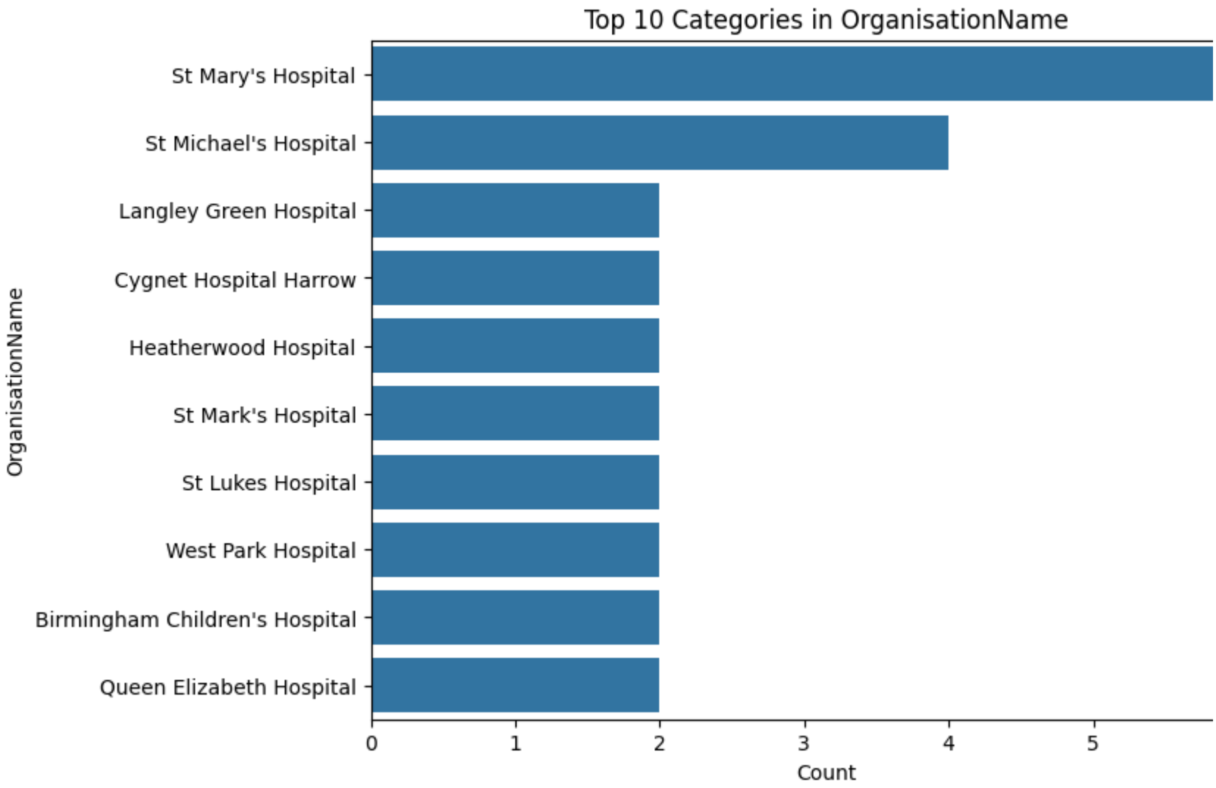


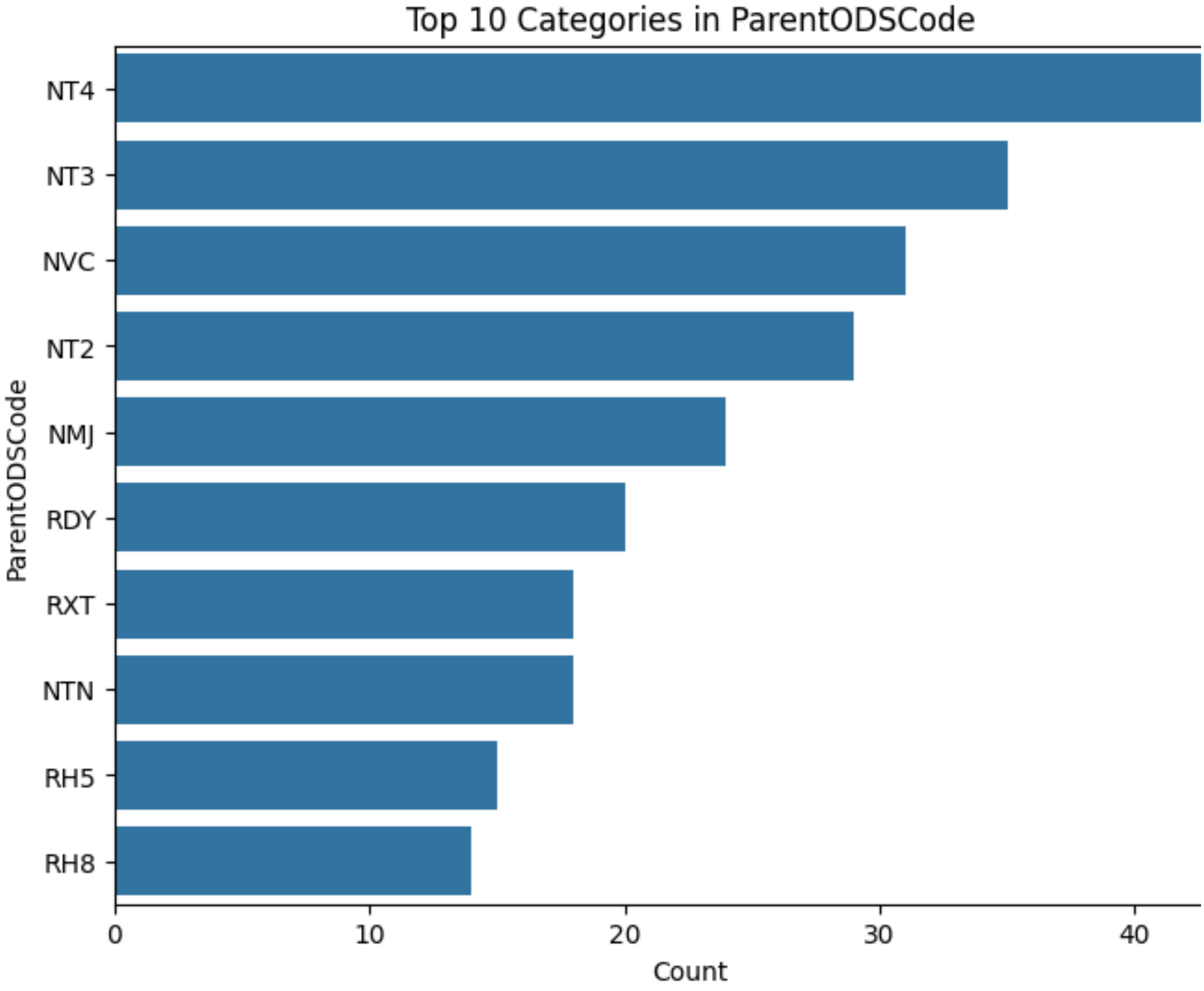

Top 10 Categories in OrganisationType
OrganisationType
Hospital
0
200
400
600
800
1000
12
Count

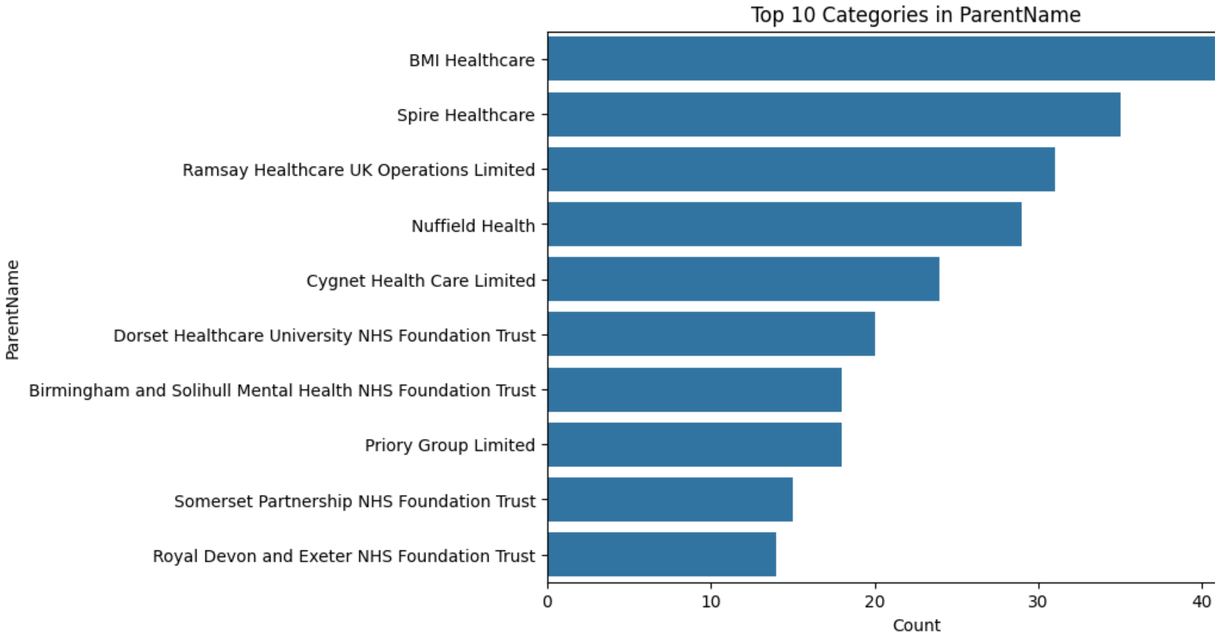

Top 10 Categories in SubType
SubType
Hospital
UNKNOWN
Mental Health Hospital
0
200
400
600
800
Count


Top 10 Categories in Sector
Sector
NHS Sector
Independent Sector
0
100
200
300
400
500
600
700
Count


Top 10 Categories in OrganisationStatus
OrganisationStatus
Visible
0
200
400
600
800
1000
12
Count

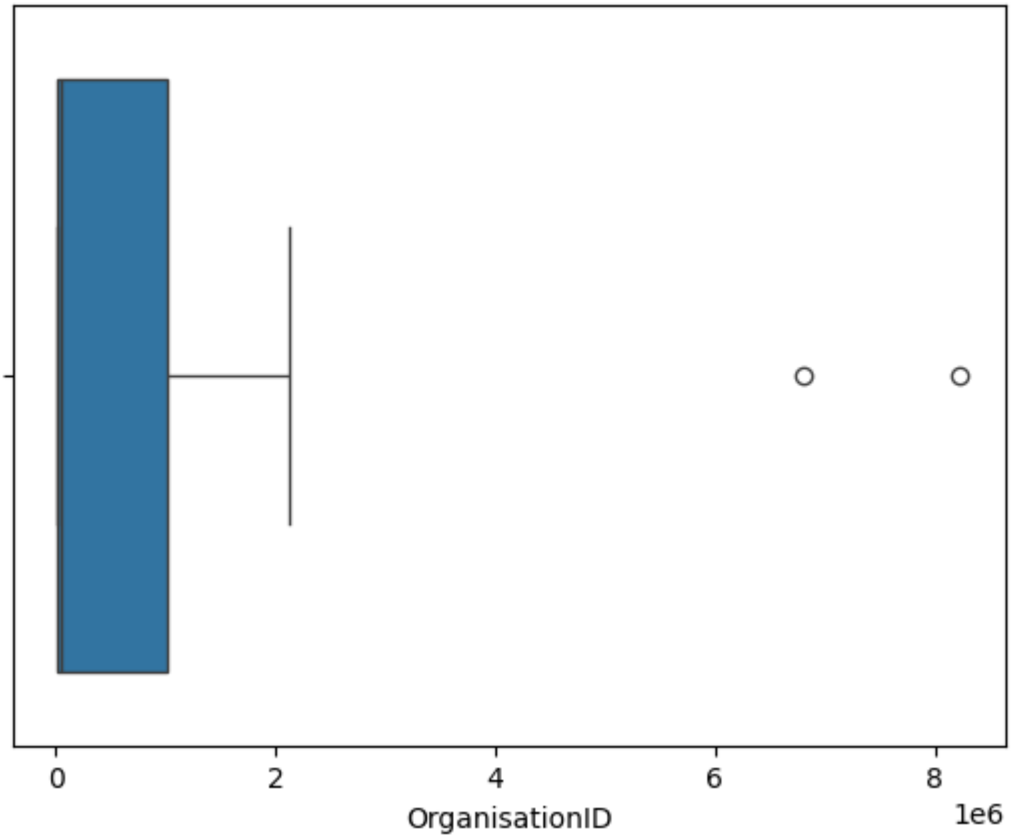

Top 10 Categories in OrganisationName
OrganisationName
St Mary's Hospital
St Michael's Hospital
Langley Green Hospital
Cygnet Hospital Harrow
Heatherwood Hospital
St Mark's Hospital
St Lukes Hospital
West Park Hospital
Birmingham Children's Hospital
Queen Elizabeth Hospital
0
1
2
3
4
5
Count

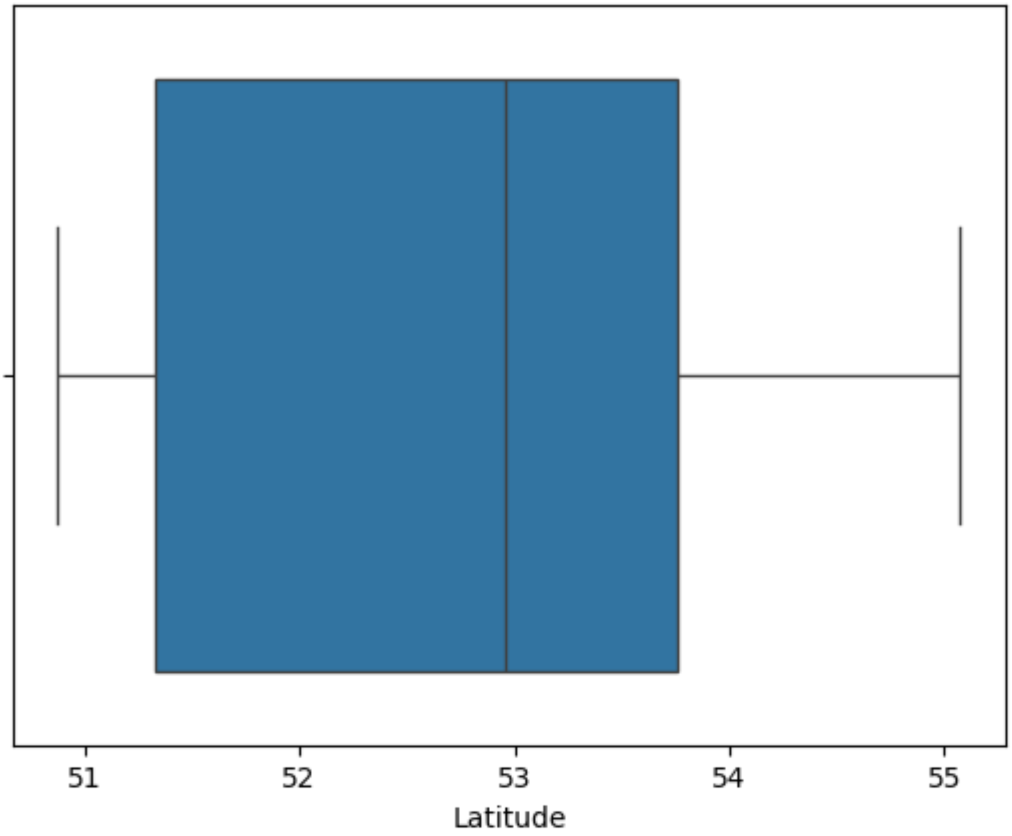

Top 10 Categories in ParentODSCode
ParentODSCode
NT4
NT3
NVC
NT2
NMJ
RDY
RXT
NTN
RH5
RH8
0
10
20
30
40
Count

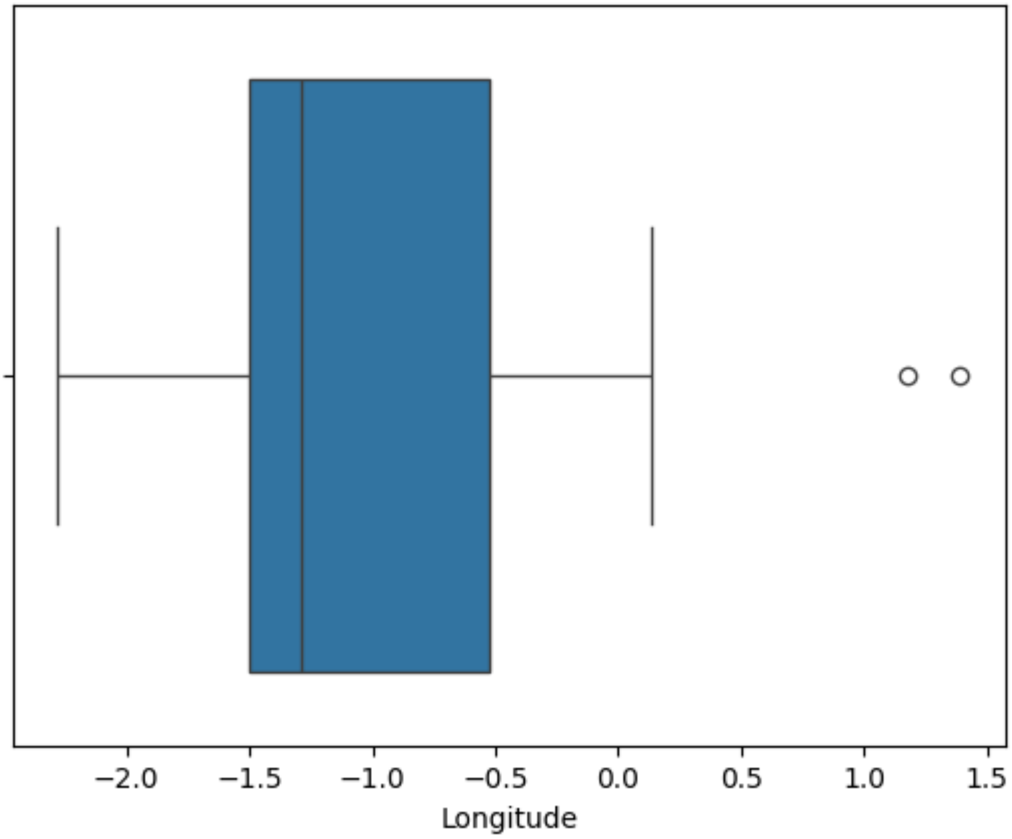


Numeric Columns Boxplot

```
In [ ]: for col in numerical_columns:
            sns.boxplot(x=data[col])
            plt.show()
```

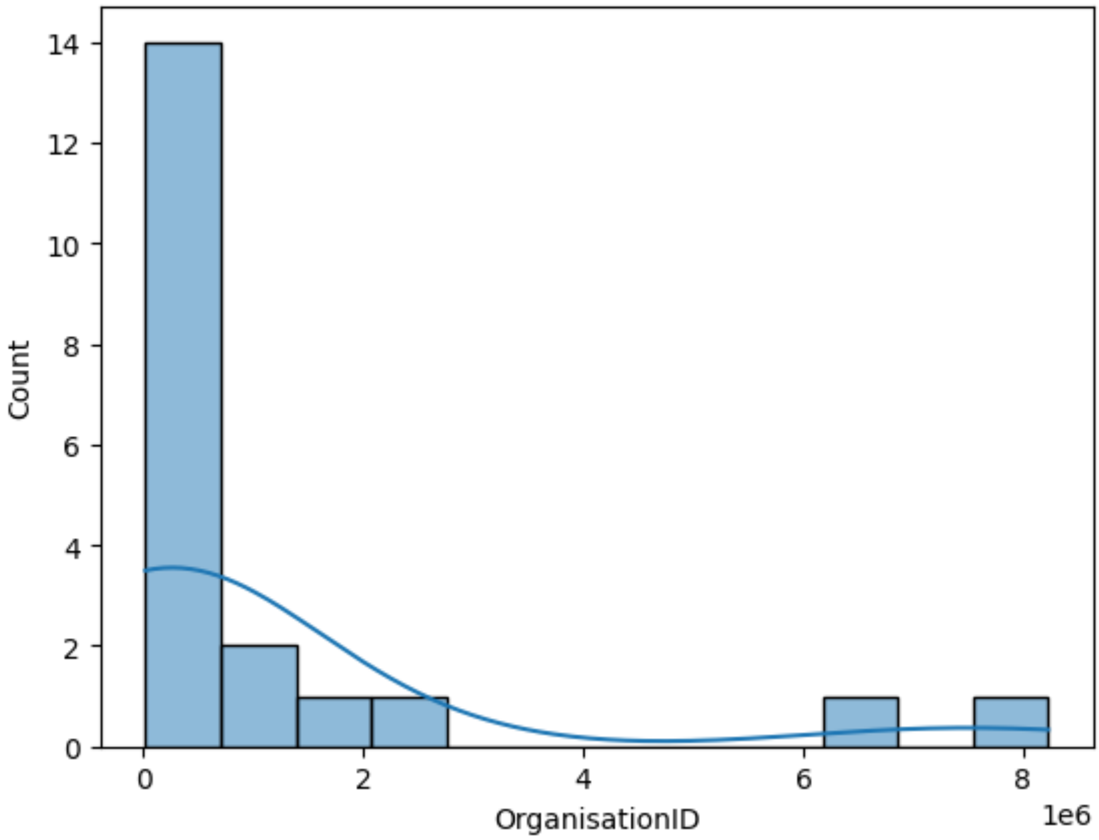


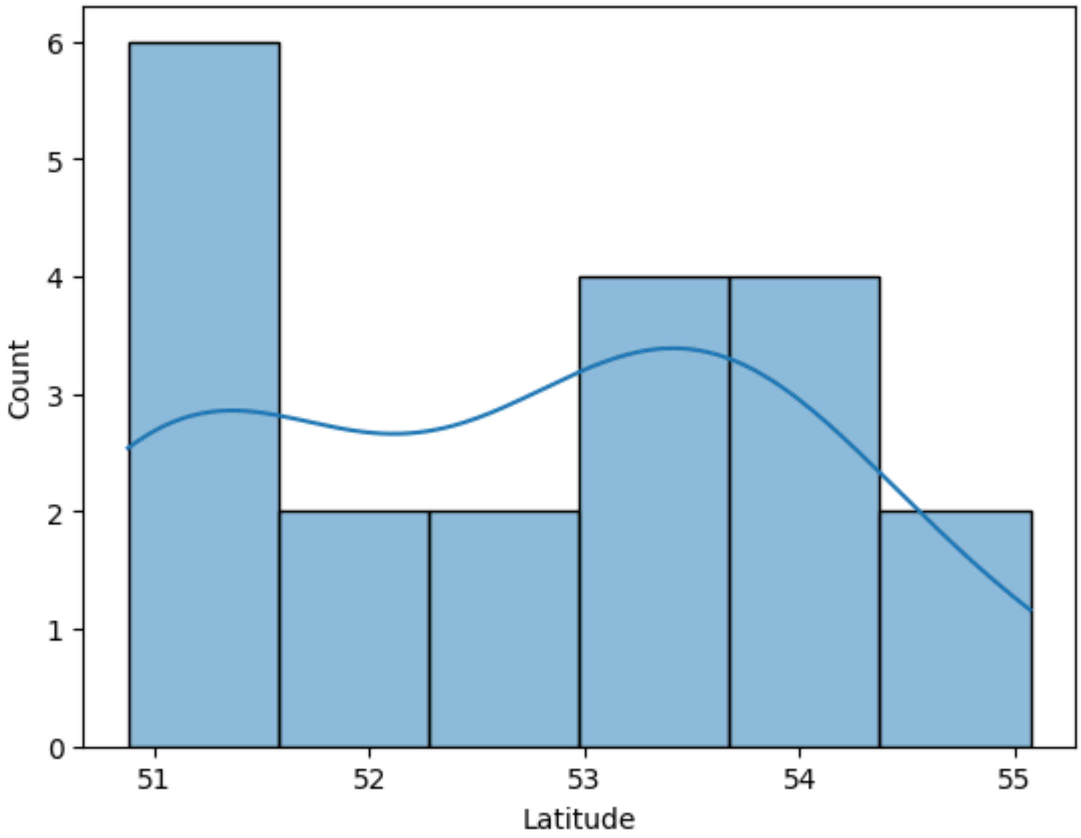


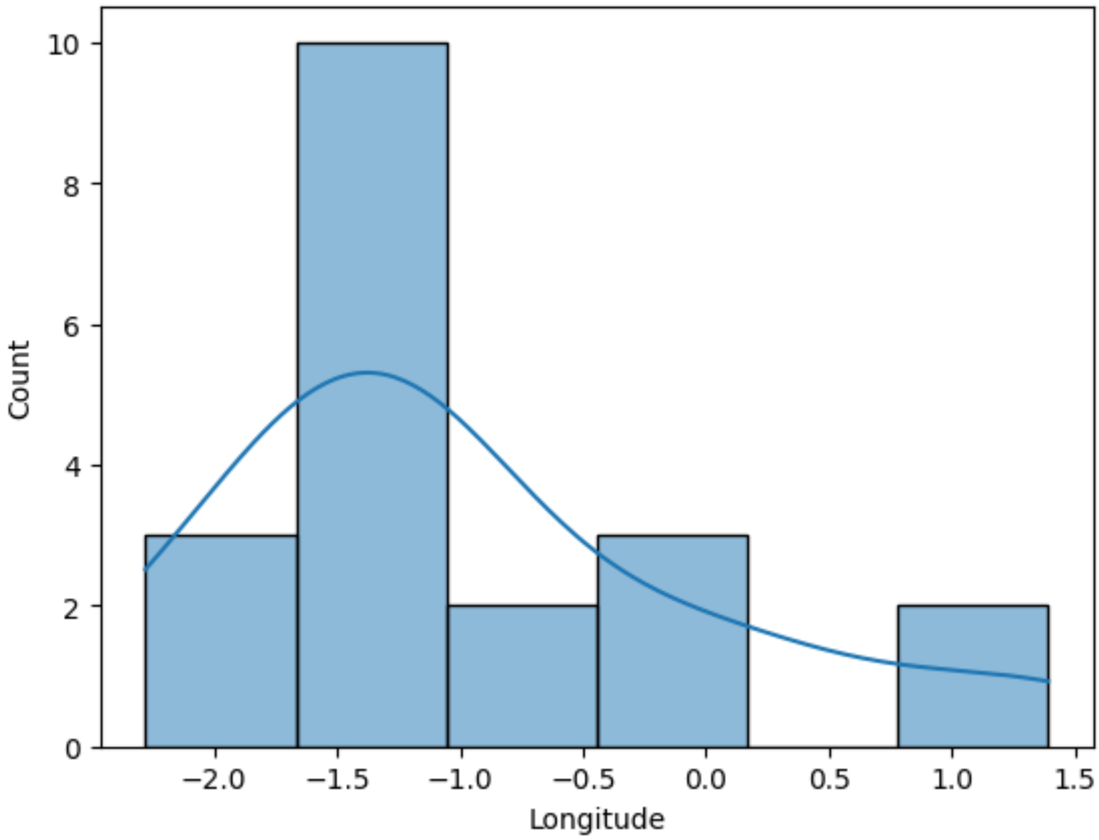


Histograms with a Kernel Density Estimate (KD

```
In [ ]: for col in numerical_columns:
            sns.histplot(data[col], kde=True)
            plt.show()
```

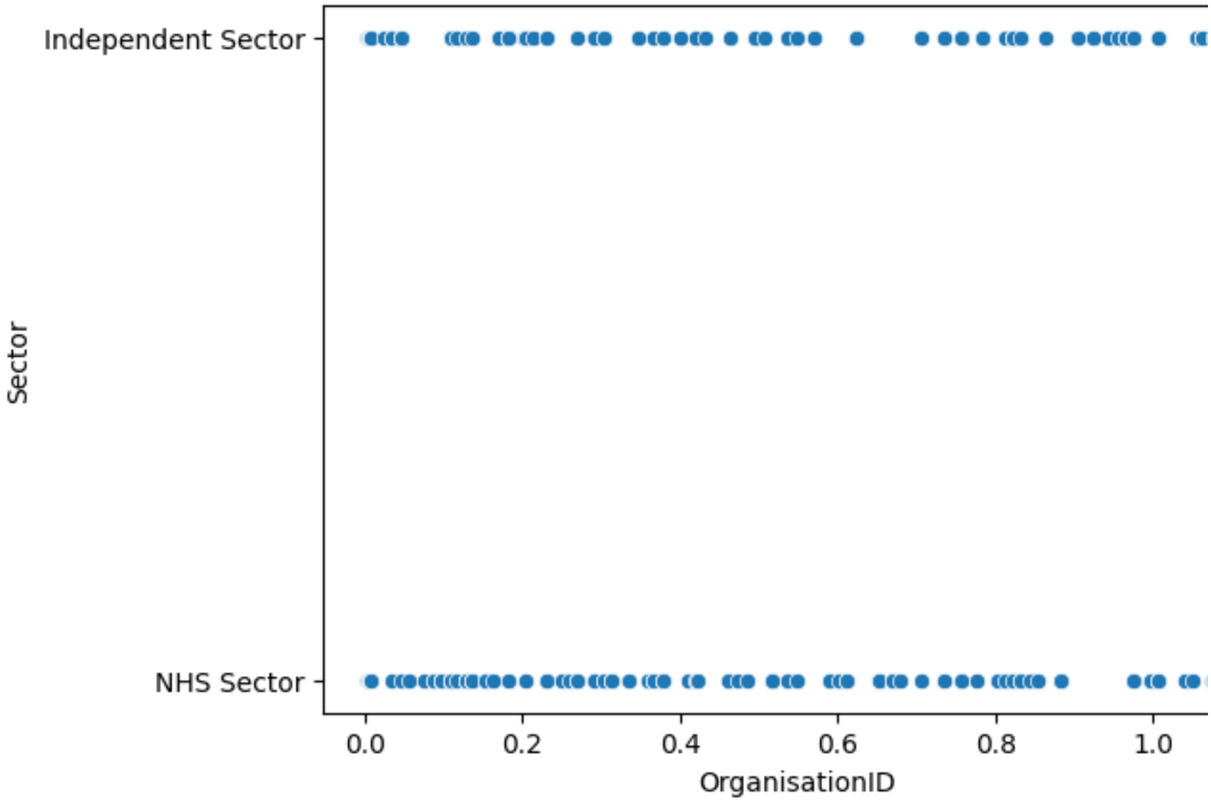






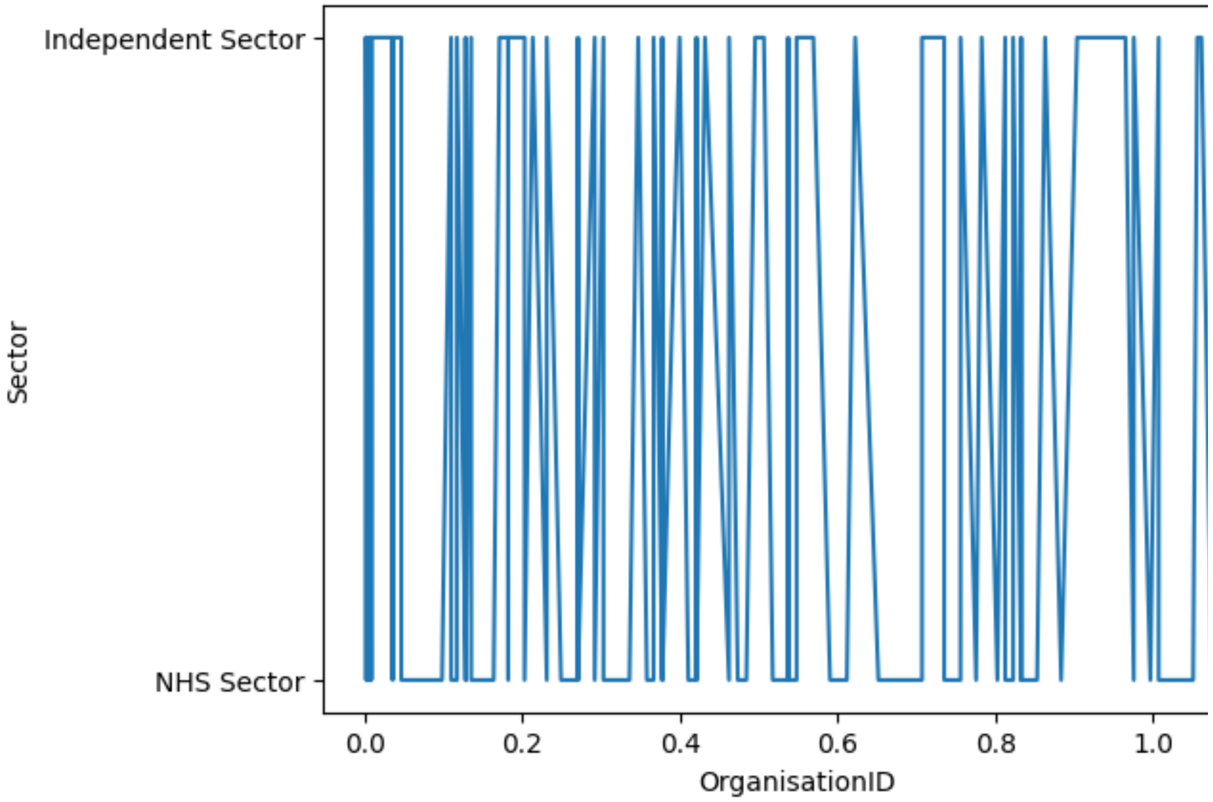
```
In [ ]: sns.scatterplot(data=data, x='OrganisationID', y='Sector')
```

```
Out[ ]: <Axes: xlabel='OrganisationID', ylabel='Sector'>
```



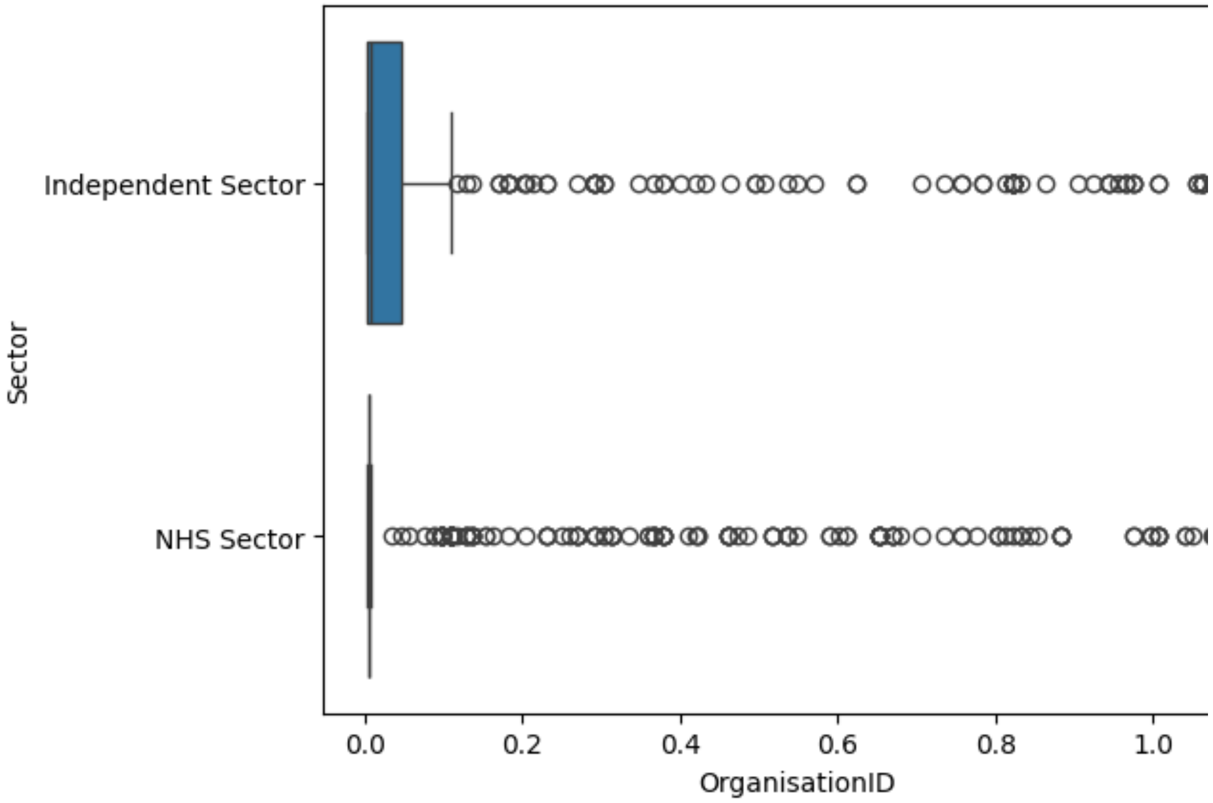
```
In [ ]: sns.lineplot(data=data, x='OrganisationID', y='Sector')
```

```
Out[ ]: <Axes: xlabel='OrganisationID', ylabel='Sector'>
```



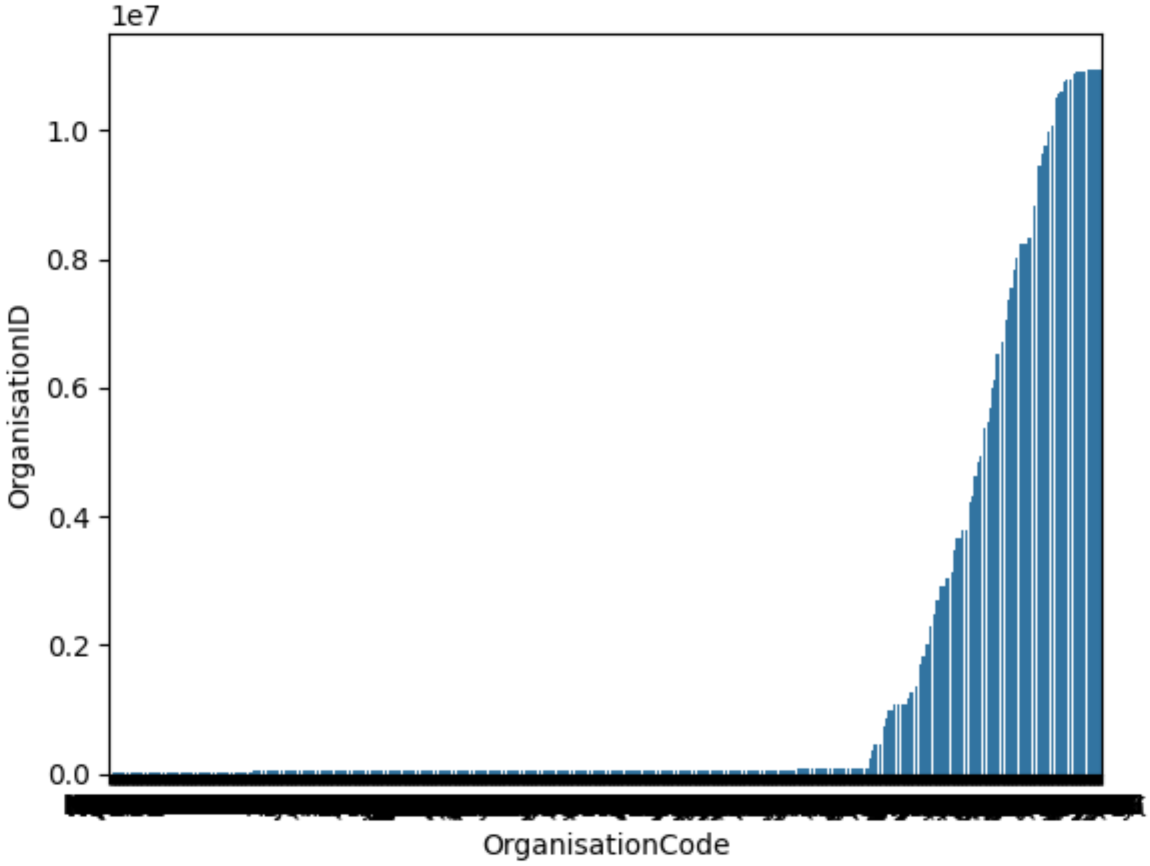
```
In [ ]: sns.boxplot(data=data, x='OrganisationID', y='Sector')
```

```
Out[ ]:  <Axes: xlabel='OrganisationID', ylabel='Sector'>
```



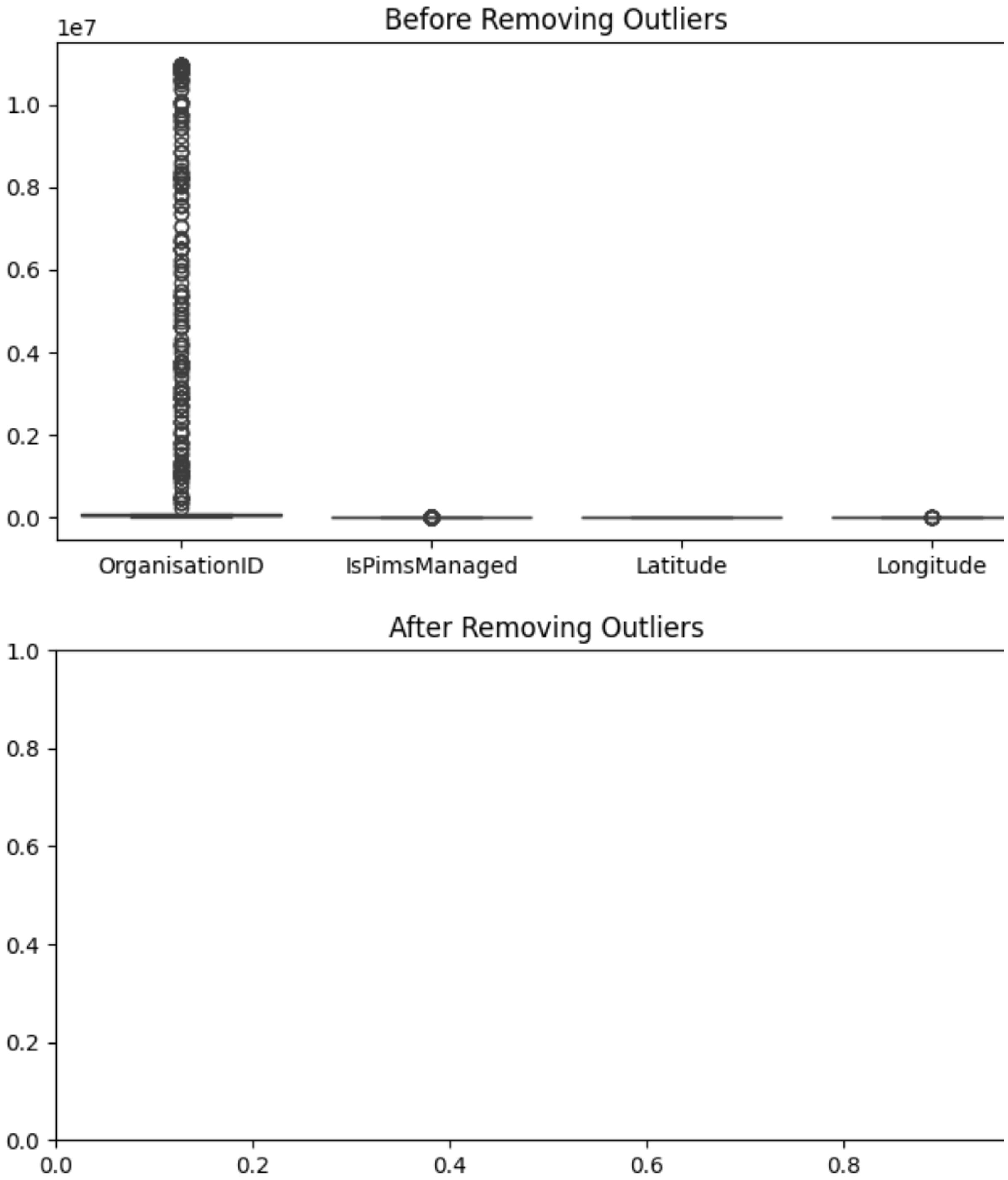
```
In [ ]: sns.barplot(data=data, x='OrganisationCode', y='OrganisationID')
```

```
Out[ ]:  <Axes: xlabel='OrganisationCode', ylabel='OrganisationID'>
```



Removing Outliers

```
In [4]: import matplotlib.pyplot as plt
        import seaborn as sns

        # Boxplot before removing outliers
        plt.figure(figsize=(8, 4))
        sns.boxplot(data=data)
        plt.title("Before Removing Outliers")
        plt.show()

        # Boxplot after removing outliers
        plt.figure(figsize=(8, 4))
        sns.boxplot(data=data_no_outliers)
        plt.title("After Removing Outliers")
        plt.show()
```


Before Removing Outliers
1e7
1.0
0.8
0.6
0.4
0.2
0.0
OrganisationID
IsPimsManaged
Latitude
Longitude
After Removing Outliers
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8