st-4035-s15355-assignment1

May 29, 2024

#

ST4035 - s15355 - Assignment1

1 a)

1.1 Python Librarys

[1]:
```
import pandas as pd
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.preprocessing import StandardScaler, OrdinalEncoder
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

1.2 Data preprocessing for train dataset

1.3 Loading Data Set

[2]:
```
df = pd.read_csv('train.csv',engine='python')
df.head()
```

[2]:

	ID	Year	Month	Hospital	Sample	ICU	OPD	Sex	Age	Ethnicity	…	\
0	1	2018	11	7	1	2	2	2	53	1	…	
1	2	2018	1	7	1	2	2	1	17	1	…	
2	3	2018	5	7	1	2	2	1	47	1	…	
3	4	2018	1	7	1	2	2	1	21	1	…	
4	5	2016	8	7	1	2	1	1	99	1	…	

	FU_L.interrogansserovarMankarsostr.Mankarso	\
0	NaN	
1	NaN	

```
2                                          NaN
3                                          NaN
4                                          NaN

   FU_L.santarosaiserovarGeorgiastr.LT117  \
0                                     NaN
1                                     NaN
2                                     NaN
3                                     NaN
4                                     NaN

   FU_L.santarosaiserovarPyrogenesstr.Salinem  \
0                                         NaN
1                                         NaN
2                                         NaN
3                                         NaN
4                                         NaN

   FU_L.interrogansserovarBataviaestr.VanTienan  \
0                                           NaN
1                                           NaN
2                                           NaN
3                                           NaN
4                                           NaN

   FU_L.interrogansserovarAlexistr.616  \
0                                  NaN
1                                  NaN
2                                  NaN
3                                  NaN
4                                  NaN

   FU_L.interrogansserovarAustralisstr.Ballico  \
0                                         NaN
1                                         NaN
2                                         NaN
3                                         NaN
4                                         NaN

   FU_L.interrogansserovarwolfiistr.3705  FU_L.interrogansserovarWeerasinghe  \
0                                    NaN                                NaN
1                                    NaN                                NaN
2                                    NaN                                NaN
3                                    NaN                                NaN
4                                    NaN                                NaN

   FU_Patoc  Final
```

```
0       NaN      2
1       NaN      1
2       NaN      2
3       NaN      2
4       NaN      2

[5 rows x 806 columns]
```

1.4 Handling missing values in train dataset

[3]:
```
df = df.drop('ID', axis=1)
```

[4]:
```
df.replace(['99', 99], np.nan,  inplace=True)
```

[5]:
```
df.isna().sum()
```

[5]:
```
Year                                          0
Month                                         0
Hospital                                      0
Sample                                        0
ICU                                          83
                                            ...
FU_L.interrogansserovarAustralisstr.Ballico    1265
FU_L.interrogansserovarwolfiistr.3705          1265
FU_L.interrogansserovarWeerasinghe             1265
FU_Patoc                                       1265
Final                                             0
Length: 805, dtype: int64
```

[6]:
```
missing_percentage = df.isnull().mean() * 100
print(missing_percentage)
```

```
Year                                          0.000000
Month                                         0.000000
Hospital                                      0.000000
Sample                                        0.000000
ICU                                           5.984138
                                               ...
FU_L.interrogansserovarAustralisstr.Ballico    91.204037
FU_L.interrogansserovarwolfiistr.3705          91.204037
FU_L.interrogansserovarWeerasinghe             91.204037
FU_Patoc                                       91.204037
Final                                          0.000000
Length: 805, dtype: float64
```

[7]:
```
print(missing_percentage[missing_percentage > 30])
```

```
Income                                          38.644557
Usualdrinkingwatersource                        69.646720
Usualbathingwatersource                         69.502523
Sourceofwaterforhousehold                       69.646720
Garbagedisposalprocedure                        69.718818
                                                   …
FU_L.interrogansserovarAlexistr.616             91.204037
FU_L.interrogansserovarAustralisstr.Ballico     91.204037
FU_L.interrogansserovarwolfiistr.3705           91.204037
FU_L.interrogansserovarWeerasinghe              91.204037
FU_Patoc                                        91.204037
Length: 762, dtype: float64
```

[8]:
```
threshold = 30
```

[9]:
```
cols_to_drop = missing_percentage[missing_percentage > threshold].index
```

[10]:
```
df = df.drop(columns=cols_to_drop)
```

[11]:
```
# print(f"Dropped columns: {cols_to_drop.tolist()}")
```

[12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1387 entries, 0 to 1386
Data columns (total 43 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Year                    1387 non-null   int64
 1   Month                   1387 non-null   int64
 2   Hospital                1387 non-null   int64
 3   Sample                  1387 non-null   int64
 4   ICU                     1304 non-null   float64
 5   OPD                     1304 non-null   float64
 6   Sex                     1242 non-null   float64
 7   Age                     1217 non-null   float64
 8   Ethnicity               1242 non-null   float64
 9   Education               1027 non-null   float64
 10  TertiaryEducation       1027 non-null   float64
 11  Prophylactics           1084 non-null   float64
 12  Pasttreatments          1088 non-null   float64
 13  Pastantibiotics         1086 non-null   float64
 14  Chronicillness          1068 non-null   float64
 15  Possibleexposure        1078 non-null   float64
 16  Feveronset              1030 non-null   float64
 17  Headacheonset           1021 non-null   float64
 18  Musclepainonset         1030 non-null   float64
 19  Cnsuffusiononset        1030 non-null   float64
```

```
 20  Jaundiceonset           1030 non-null   float64
 21  Skinrashonset           1030 non-null   float64
 22  Oliguriaonset           1030 non-null   float64
 23  Anuriaonset             1030 non-null   float64
 24  SOBonset                1030 non-null   float64
 25  Coughonset              1030 non-null   float64
 26  Haemoptasisonset        1030 non-null   float64
 27  Chestpainonset          1029 non-null   float64
 28  Nauseaonset             1030 non-null   float64
 29  Vomitingonset           1030 non-null   float64
 30  Diarrhoeaonset          1030 non-null   float64
 31  Bleedingonset           1028 non-null   float64
 32  Mucosalrashonset        1030 non-null   float64
 33  Prostrationonset        1030 non-null   float64
 34  Rigorsonset             1030 non-null   float64
 35  Photophobiaonset        1030 non-null   float64
 36  Chillsonset             1030 non-null   float64
 37  Muscletendernessonset   1030 non-null   float64
 38  Psychoticsymptomsonset  1030 non-null   float64
 39  Confusiononset          1030 non-null   float64
 40  WPqPCRDiagnosis         1155 non-null   float64
 41  Isolate                 1387 non-null   int64
 42  Final                   1387 non-null   int64
dtypes: float64(37), int64(6)
memory usage: 466.1 KB
```

[13]:
```python
for column in df.columns:
    unique_values = df[column].unique()
    print(f"Column: {column}")
    print(f"Unique values: {unique_values}")
    print(f"Number of unique values: {len(unique_values)}")
    print("\n")
```

```
Column: Year
Unique values: [2018 2016 2017 2019]
Number of unique values: 4


Column: Month
Unique values: [11  1  5  8 12  6  7  9 10  2  3  4]
Number of unique values: 12


Column: Hospital
Unique values: [7 5 1 8 4 3 2 6]
Number of unique values: 8
```

```
Column: Sample
Unique values: [1 2]
Number of unique values: 2


Column: ICU
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: OPD
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Sex
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Age
Unique values: [53. 17. 47. 21. nan 64. 50. 59. 55. 65. 52. 26. 24. 40. 45. 35.
41. 30.
 33. 34. 20. 49. 63. 73. 60. 43. 38. 61. 39. 51. 14. 23. 54. 19. 57. 28.
 31. 22. 42. 68. 32. 37. 56. 25. 36. 46. 58. 27. 29. 48. 44. 75. 70. 69.
 71. 13. 62. 66. 15. 16. 18. 67. 76.  2. 72. 74.  5. 79.  6. 87.  8. 12.
 11. 80.  9. 85.]
Number of unique values: 76


Column: Ethnicity
Unique values: [ 1. nan  6.  3.  2.]
Number of unique values: 5


Column: Education
Unique values: [ 9. 11.  8. 10.  3.  2.  4. 12. nan  7. 13.  0.  5.  6.  1.]
Number of unique values: 15


Column: TertiaryEducation
Unique values: [ 3. nan  2.  1.]
Number of unique values: 4


Column: Prophylactics
Unique values: [ 3.  2. nan  1.]
Number of unique values: 4
```

```
Column: Pasttreatments
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Pastantibiotics
Unique values: [ 1.  2.  3. nan]
Number of unique values: 4


Column: Chronicillness
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Possibleexposure
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Feveronset
Unique values: [ 1. nan  2.]
Number of unique values: 3


Column: Headacheonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Musclepainonset
Unique values: [ 1. nan  2.]
Number of unique values: 3


Column: Cnsuffusiononset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Jaundiceonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Skinrashonset
```

```
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Oliguriaonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Anuriaonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: SOBonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Coughonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Haemoptasisonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Chestpainonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Nauseaonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Vomitingonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Diarrhoeaonset
Unique values: [ 1.  2. nan]
Number of unique values: 3
```

```
Column: Bleedingonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Mucosalrashonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Prostrationonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Rigorsonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Photophobiaonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Chillsonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Muscletendernessonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Psychoticsymptomsonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Confusiononset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: WPqPCRDiagnosis
Unique values: [ 3.  1.  2. nan]
```

```
Number of unique values: 4


Column: Isolate
Unique values: [ 2 98  1]
Number of unique values: 3


Column: Final
Unique values: [2 1]
Number of unique values: 2
```

[14]:
```
numerical_columns = ['Age']
categorical_columns = [col for col in df.columns if col not in␣
 ↪numerical_columns]
categorical_columns
```

[14]:
```
['Year',
 'Month',
 'Hospital',
 'Sample',
 'ICU',
 'OPD',
 'Sex',
 'Ethnicity',
 'Education',
 'TertiaryEducation',
 'Prophylactics',
 'Pasttreatments',
 'Pastantibiotics',
 'Chronicillness',
 'Possibleexposure',
 'Feveronset',
 'Headacheonset',
 'Musclepainonset',
 'Cnsuffusiononset',
 'Jaundiceonset',
 'Skinrashonset',
 'Oliguriaonset',
 'Anuriaonset',
 'SOBonset',
 'Coughonset',
 'Haemoptasisonset',
 'Chestpainonset',
 'Nauseaonset',
```

```
 'Vomitingonset',
 'Diarrhoeaonset',
 'Bleedingonset',
 'Mucosalrashonset',
 'Prostrationonset',
 'Rigorsonset',
 'Photophobiaonset',
 'Chillsonset',
 'Muscletendernessonset',
 'Psychoticsymptomsonset',
 'Confusiononset',
 'WPqPCRDiagnosis',
 'Isolate',
 'Final']
```

[15]:
```
for col in categorical_columns:
    df[col] = df[col].astype('category')
```

[16]:
```
for col in categorical_columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

[17]:
```
for col in numerical_columns:
    df[col].fillna(df[col].mean(), inplace=True)
```

[18]:
```
df.info(all)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1387 entries, 0 to 1386
Data columns (total 43 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Year                    1387 non-null   category
 1   Month                   1387 non-null   category
 2   Hospital                1387 non-null   category
 3   Sample                  1387 non-null   category
 4   ICU                     1387 non-null   category
 5   OPD                     1387 non-null   category
 6   Sex                     1387 non-null   category
 7   Age                     1387 non-null   float64
 8   Ethnicity               1387 non-null   category
 9   Education               1387 non-null   category
 10  TertiaryEducation       1387 non-null   category
 11  Prophylactics           1387 non-null   category
 12  Pasttreatments          1387 non-null   category
 13  Pastantibiotics         1387 non-null   category
 14  Chronicillness          1387 non-null   category
 15  Possibleexposure        1387 non-null   category
```

```
 16  Feveronset               1387 non-null   category
 17  Headacheonset            1387 non-null   category
 18  Musclepainonset          1387 non-null   category
 19  Cnsuffusiononset         1387 non-null   category
 20  Jaundiceonset            1387 non-null   category
 21  Skinrashonset            1387 non-null   category
 22  Oliguriaonset            1387 non-null   category
 23  Anuriaonset              1387 non-null   category
 24  SOBonset                 1387 non-null   category
 25  Coughonset               1387 non-null   category
 26  Haemoptasisonset         1387 non-null   category
 27  Chestpainonset           1387 non-null   category
 28  Nauseaonset              1387 non-null   category
 29  Vomitingonset            1387 non-null   category
 30  Diarrhoeaonset           1387 non-null   category
 31  Bleedingonset            1387 non-null   category
 32  Mucosalrashonset         1387 non-null   category
 33  Prostrationonset         1387 non-null   category
 34  Rigorsonset              1387 non-null   category
 35  Photophobiaonset         1387 non-null   category
 36  Chillsonset              1387 non-null   category
 37  Muscletendernessonset    1387 non-null   category
 38  Psychoticsymptomsonset   1387 non-null   category
 39  Confusiononset           1387 non-null   category
 40  WPqPCRDiagnosis          1387 non-null   category
 41  Isolate                  1387 non-null   category
 42  Final                    1387 non-null   category
dtypes: category(42), float64(1)
memory usage: 74.2 KB
```

1.5 Checking the duplicates

[19]:
```
df.duplicated()
```

[19]:
```
0       False
1       False
2       False
3       False
4       False
        …
1382    False
1383    False
1384    False
1385    False
1386    False
Length: 1387, dtype: bool
```

[20]:
```
df.duplicated().sum()
```

[20]: 117

[21]:
```
df=df.drop_duplicates()
```

[22]:
```
df.duplicated().sum()
```

[22]: 0

1.6 Descriptive analysis

[23]:
```
# Summary Statistics
df.info(all)
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1270 entries, 0 to 1386
Data columns (total 43 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Year                    1270 non-null   category
 1   Month                   1270 non-null   category
 2   Hospital                1270 non-null   category
 3   Sample                  1270 non-null   category
 4   ICU                     1270 non-null   category
 5   OPD                     1270 non-null   category
 6   Sex                     1270 non-null   category
 7   Age                     1270 non-null   float64
 8   Ethnicity               1270 non-null   category
 9   Education               1270 non-null   category
 10  TertiaryEducation       1270 non-null   category
 11  Prophylactics           1270 non-null   category
 12  Pasttreatments          1270 non-null   category
 13  Pastantibiotics         1270 non-null   category
 14  Chronicillness          1270 non-null   category
 15  Possibleexposure        1270 non-null   category
 16  Feveronset              1270 non-null   category
 17  Headacheonset           1270 non-null   category
 18  Musclepainonset         1270 non-null   category
 19  Cnsuffusiononset        1270 non-null   category
 20  Jaundiceonset           1270 non-null   category
 21  Skinrashonset           1270 non-null   category
 22  Oliguriaonset           1270 non-null   category
 23  Anuriaonset             1270 non-null   category
 24  SOBonset                1270 non-null   category
 25  Coughonset              1270 non-null   category
 26  Haemoptasisonset        1270 non-null   category
```

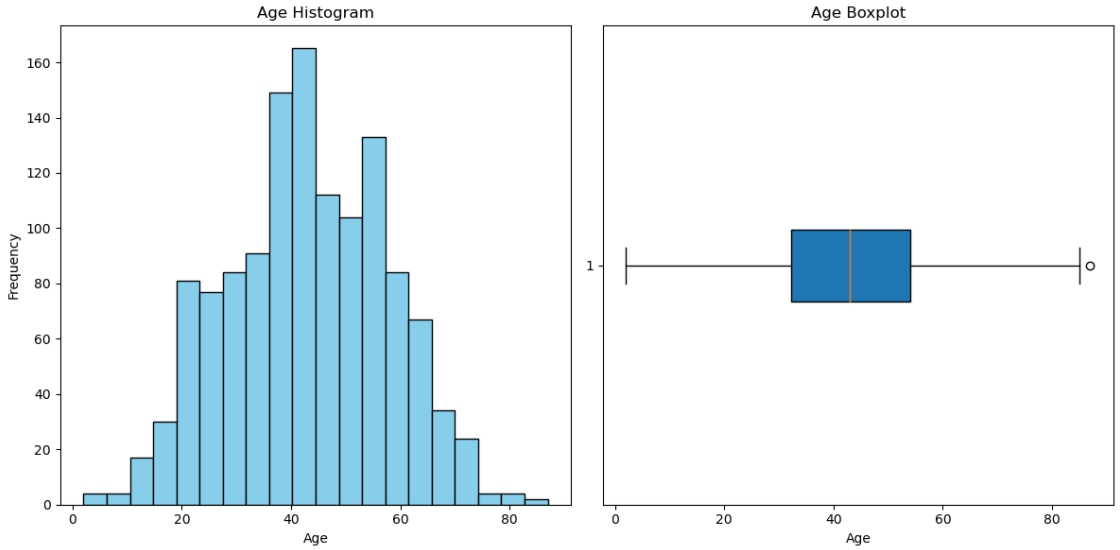
```
 27  Chestpainonset          1270 non-null   category
 28  Nauseaonset             1270 non-null   category
 29  Vomitingonset           1270 non-null   category
 30  Diarrhoeaonset          1270 non-null   category
 31  Bleedingonset           1270 non-null   category
 32  Mucosalrashonset        1270 non-null   category
 33  Prostrationonset        1270 non-null   category
 34  Rigorsonset             1270 non-null   category
 35  Photophobiaonset        1270 non-null   category
 36  Chillsonset             1270 non-null   category
 37  Muscletendernessonset   1270 non-null   category
 38  Psychoticsymptomsonset  1270 non-null   category
 39  Confusiononset          1270 non-null   category
 40  WPqPCRDiagnosis         1270 non-null   category
 41  Isolate                 1270 non-null   category
 42  Final                   1270 non-null   category
dtypes: category(42), float64(1)
memory usage: 78.2 KB
```

[24]:
```
# Summary for the Continuous Variables
df[numerical_columns].describe()
```

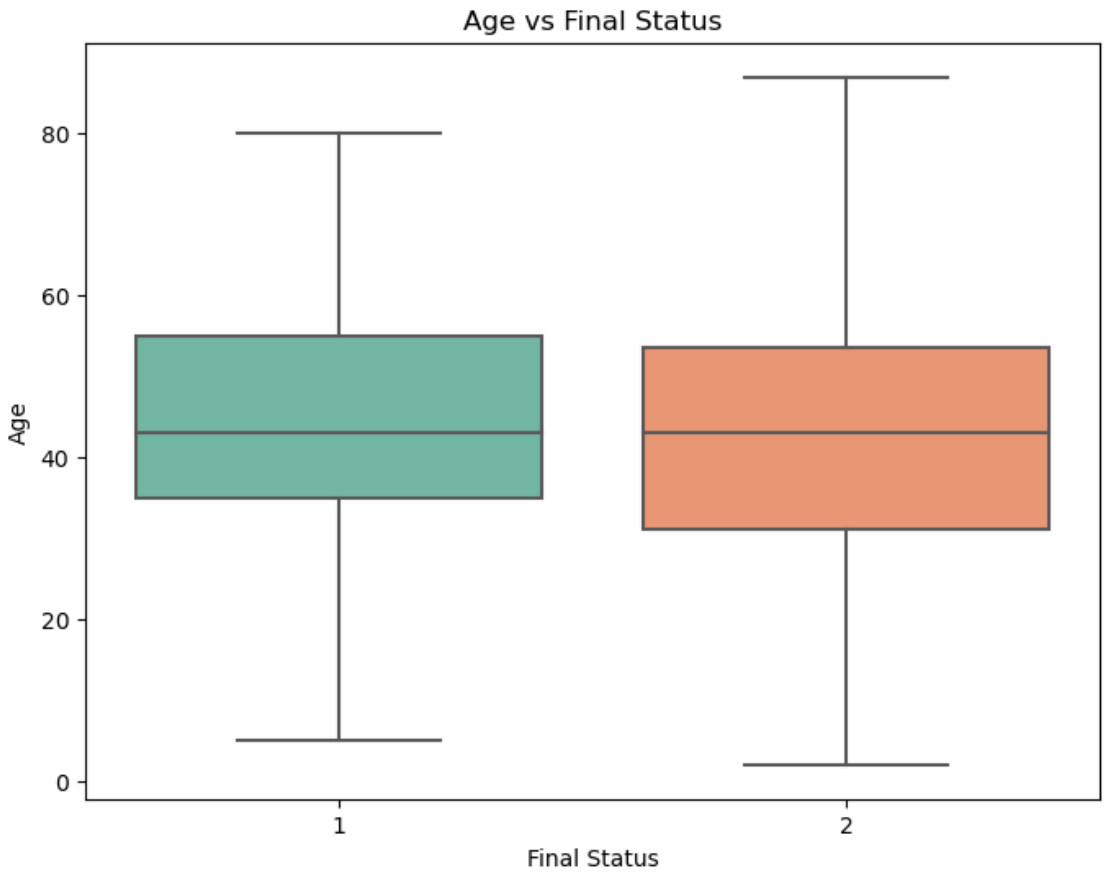
[24]:
```
                Age
count  1270.000000
mean     42.980581
std      14.521136
min       2.000000
25%      32.250000
50%      42.955629
75%      54.000000
max      87.000000
```

[25]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))

# Plot histogram on the first subplot (left side)
ax1.hist(df['Age'], bins=20, color='skyblue', edgecolor='black')
ax1.set_title('Age Histogram')
ax1.set_xlabel('Age')
ax1.set_ylabel('Frequency')

# Plot boxplot on the second subplot (right side)
ax2.boxplot(df['Age'], vert=False, patch_artist=True)
ax2.set_title('Age Boxplot')
ax2.set_xlabel('Age')

# Adjust layout to prevent overlap
plt.tight_layout()
```

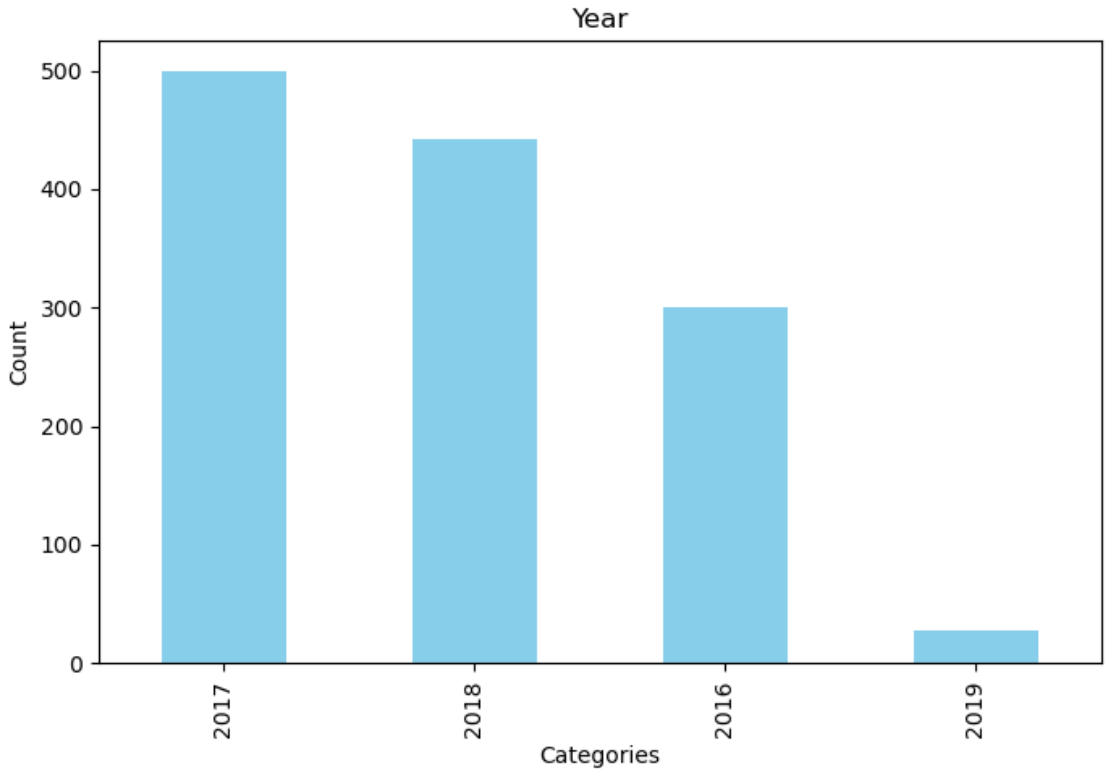
```
# Show the plots
plt.show()
```

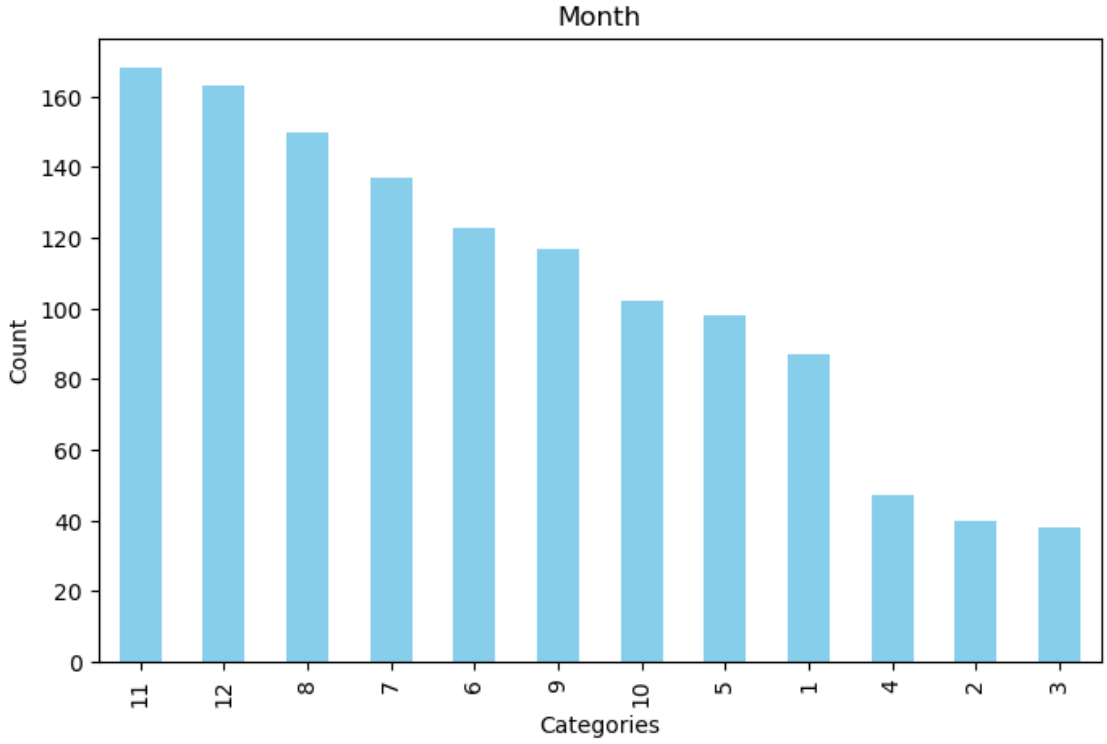


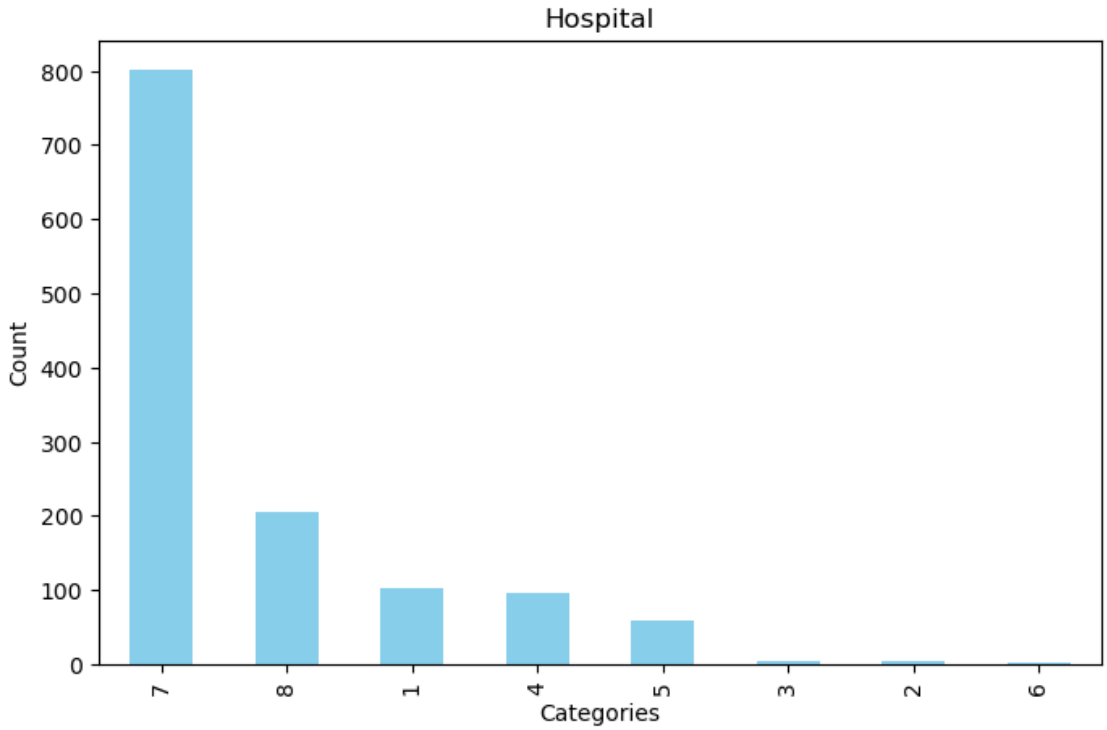
[26]:
```
# Boxplot for numerical variables grouped by Final Status
for col in numerical_columns:
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='Final', y=col, data=df, palette='Set2')
    plt.title(col + " vs Final Status")
    plt.xlabel("Final Status")
    plt.ylabel(col)
    plt.show()
```

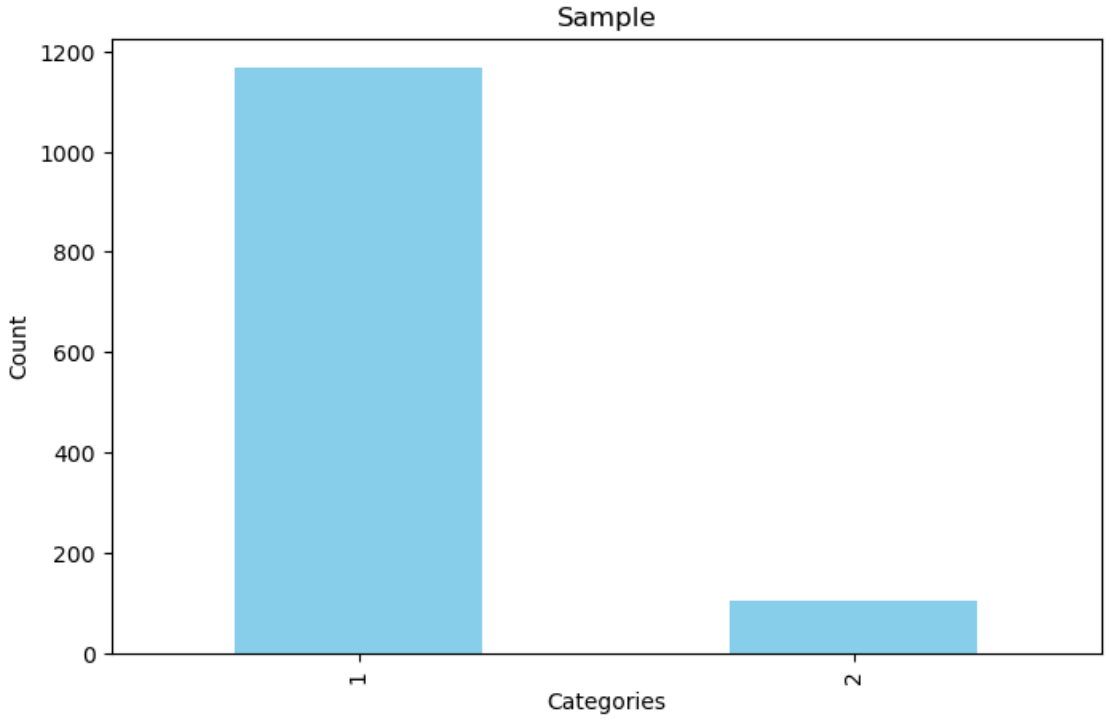


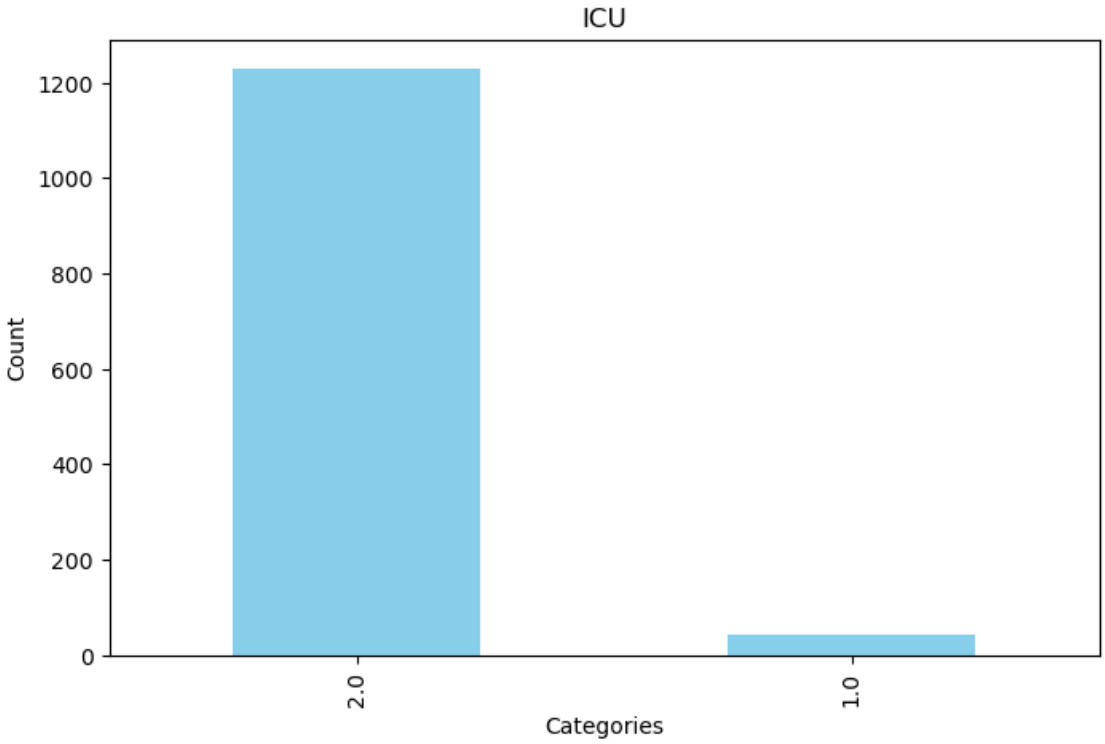
[27]:
```
# Bar plots for categorical variables
for col in categorical_columns:
    plt.figure(figsize=(8, 5))
    df[col].value_counts().plot(kind='bar', color='skyblue')
    plt.title(col)
    plt.xlabel('Categories')
    plt.ylabel('Count')
    plt.show()
```

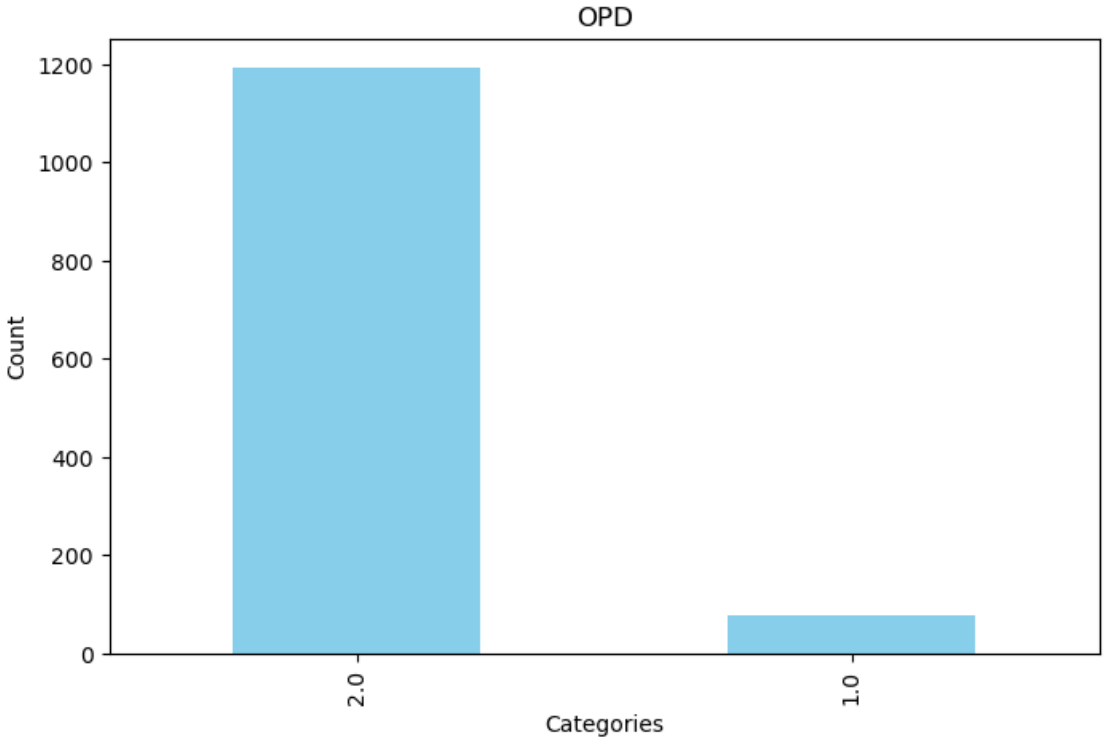

Year
Count
500
400
300
200
100
0
2017
2018
2016
2019
Categories

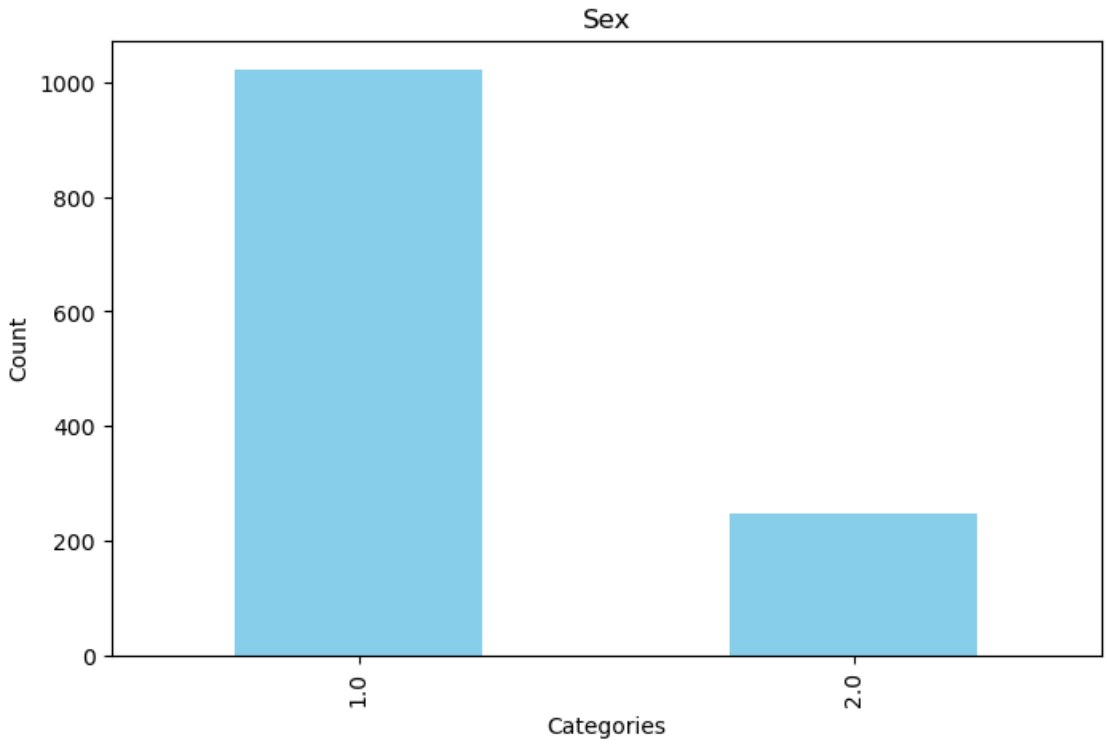

Month
Count
160
140
120
100
80
60
40
20
0
11
12
8
7
6
9
10
5
1
4
2
3
Categories

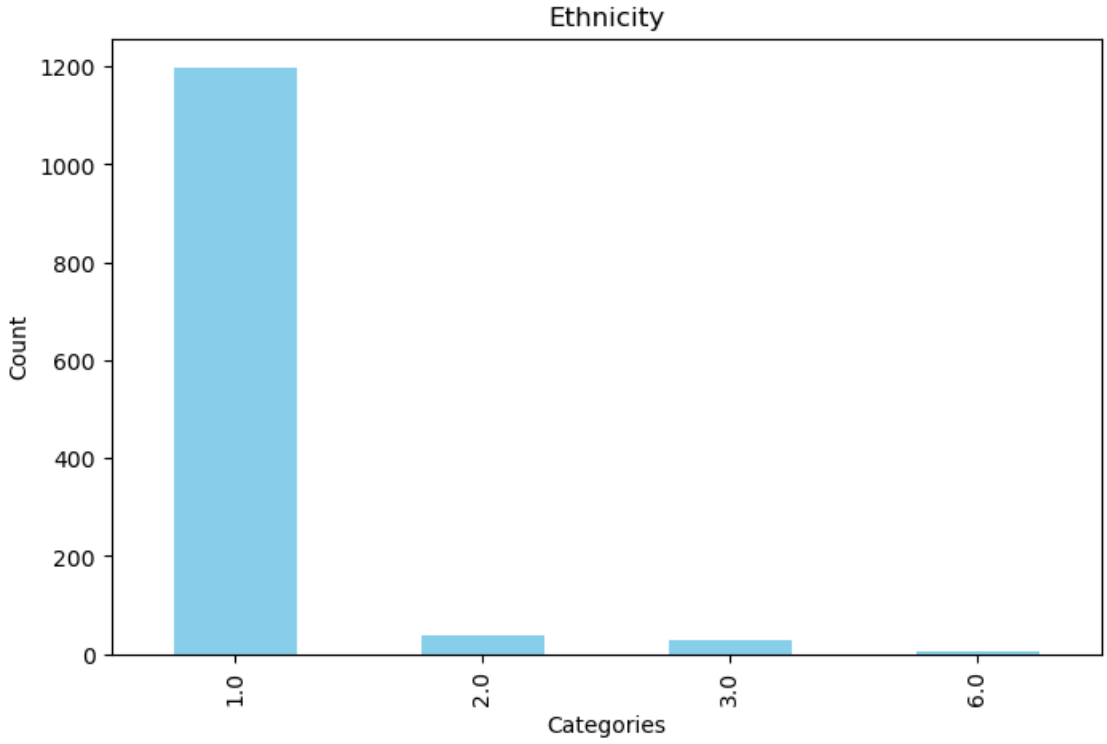

Hospital
Count
800
700
600
500
400
300
200
100
0
7
8
1
4
5
3
2
6
Categories

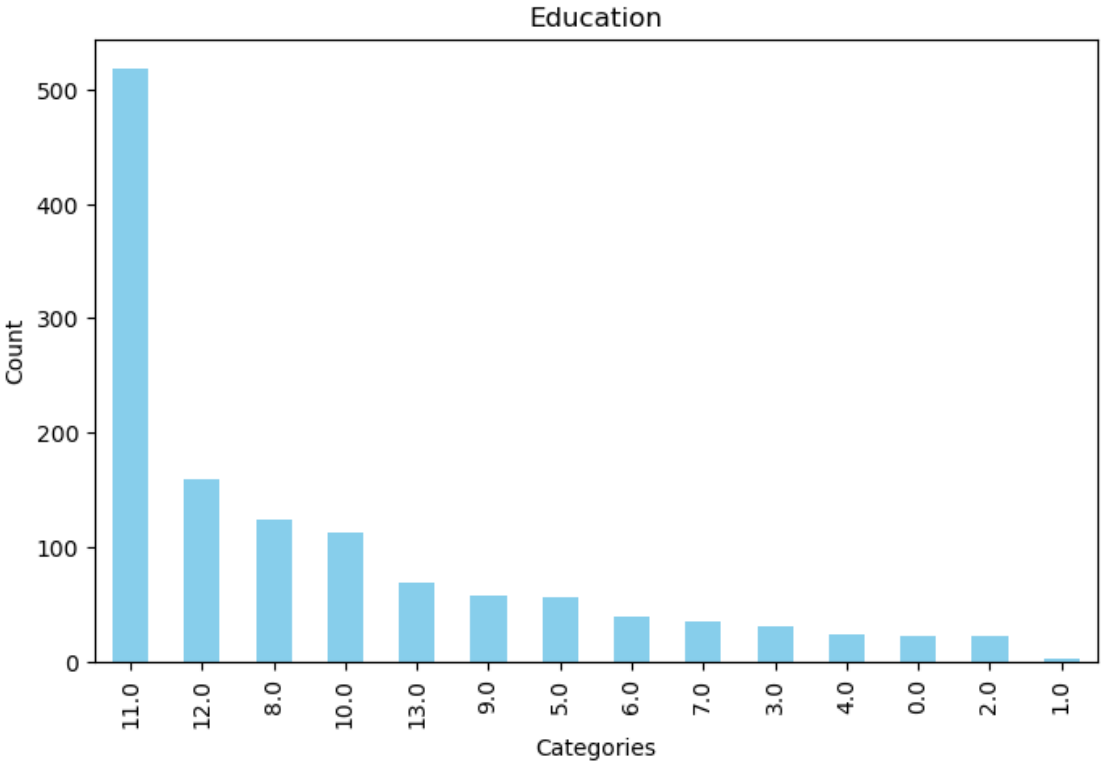

Sample
Count
1200
1000
800
600
400
200
0
1
2
Categories

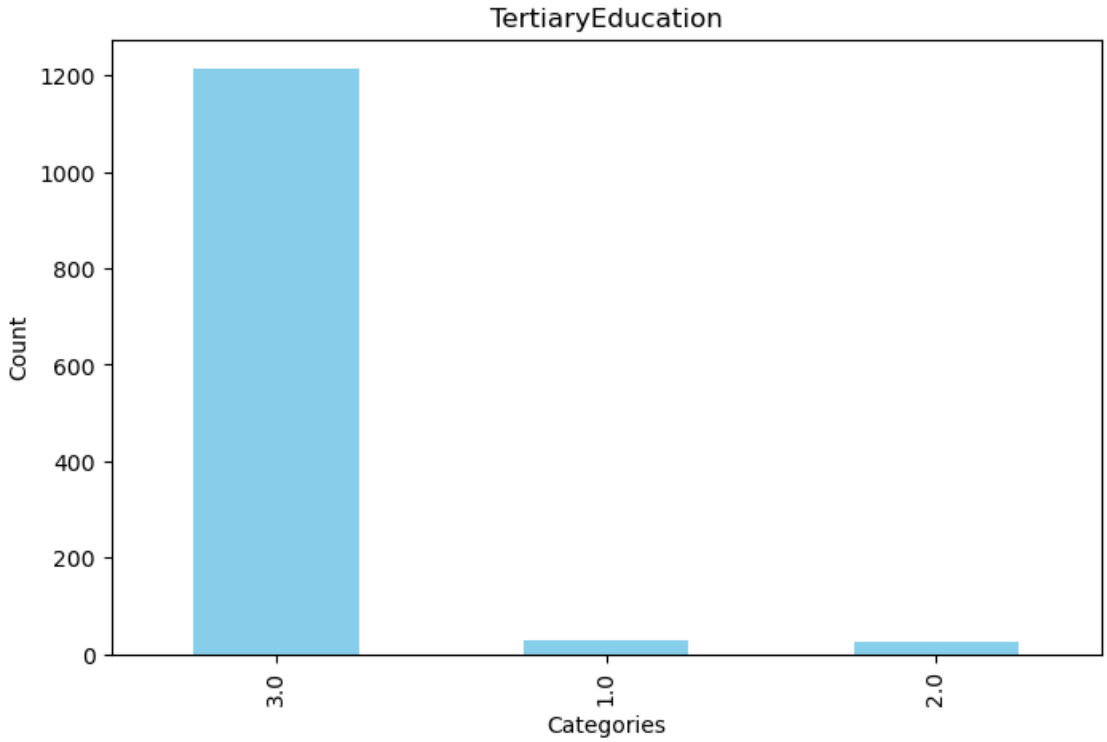

ICU
Count
1200
1000
800
600
400
200
0
2.0
1.0
Categories

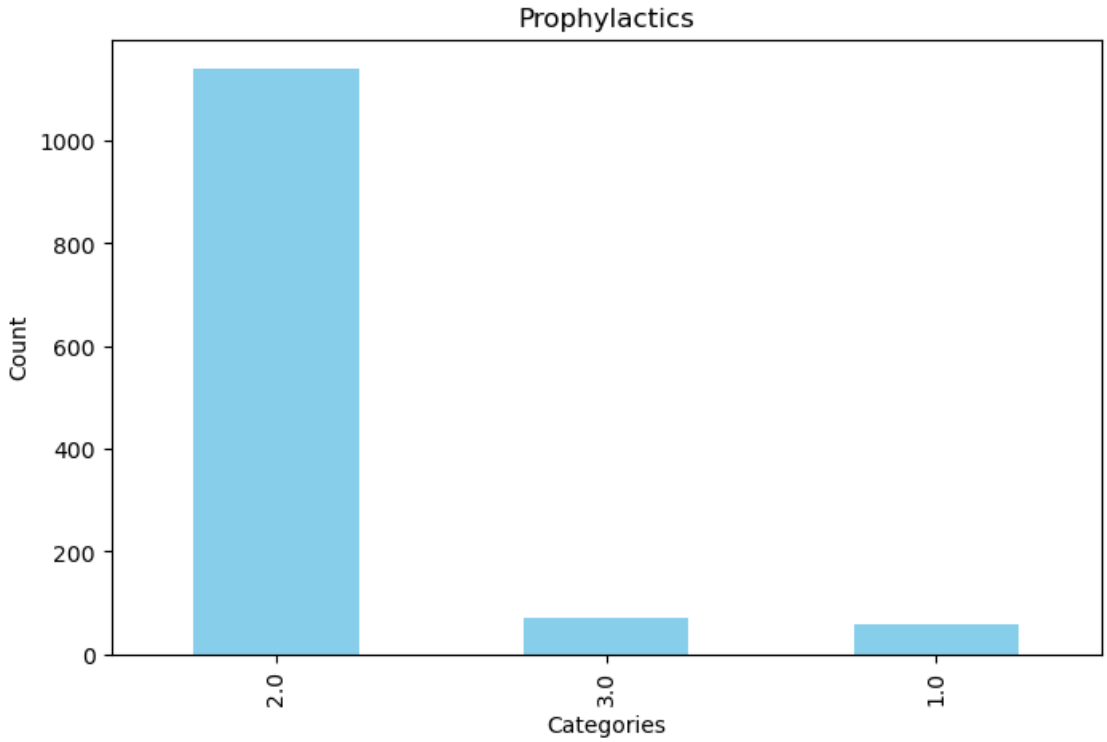

OPD
Count
1200
1000
800
600
400
200
0
2.0
1.0
Categories

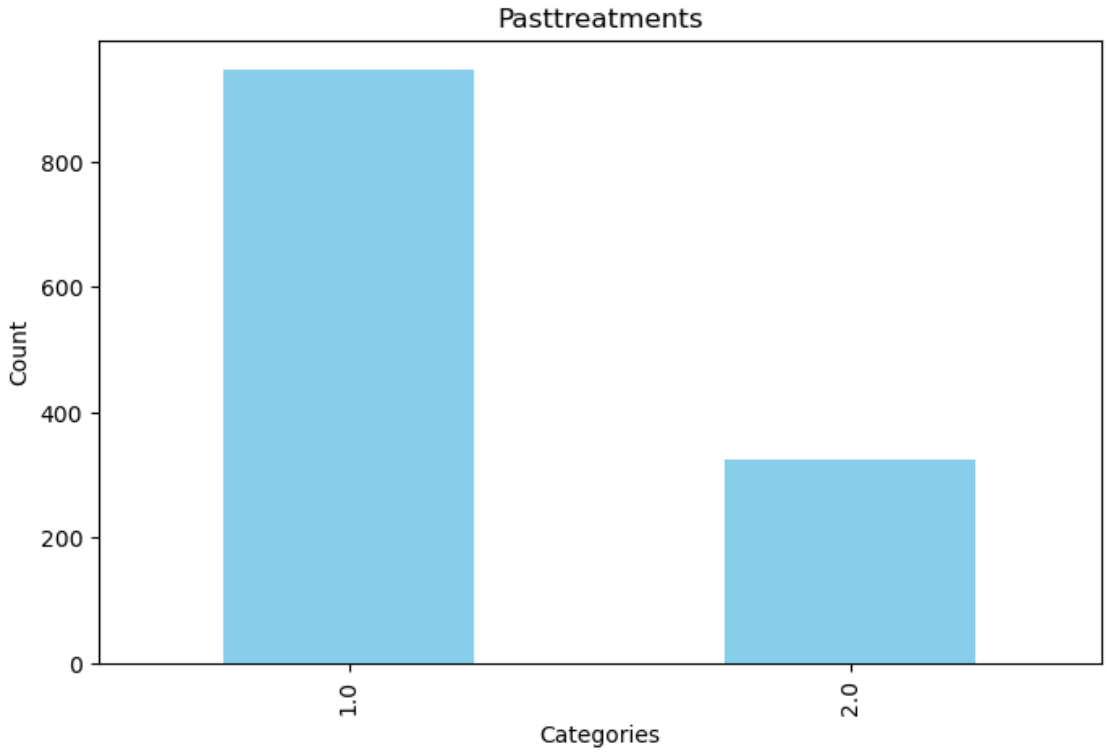

Sex
1000
800
600
400
200
0
Count
1.0
2.0
Categories

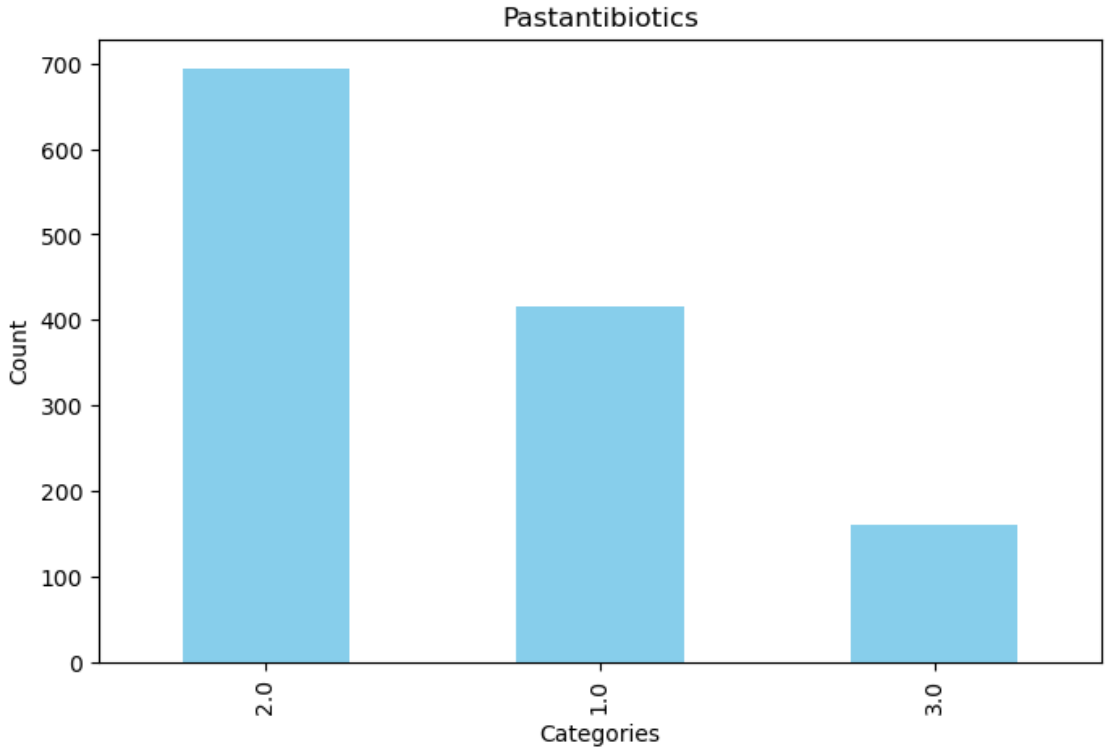

Ethnicity
1200
1000
800
600
400
200
0
Count
1.0
2.0
3.0
6.0
Categories

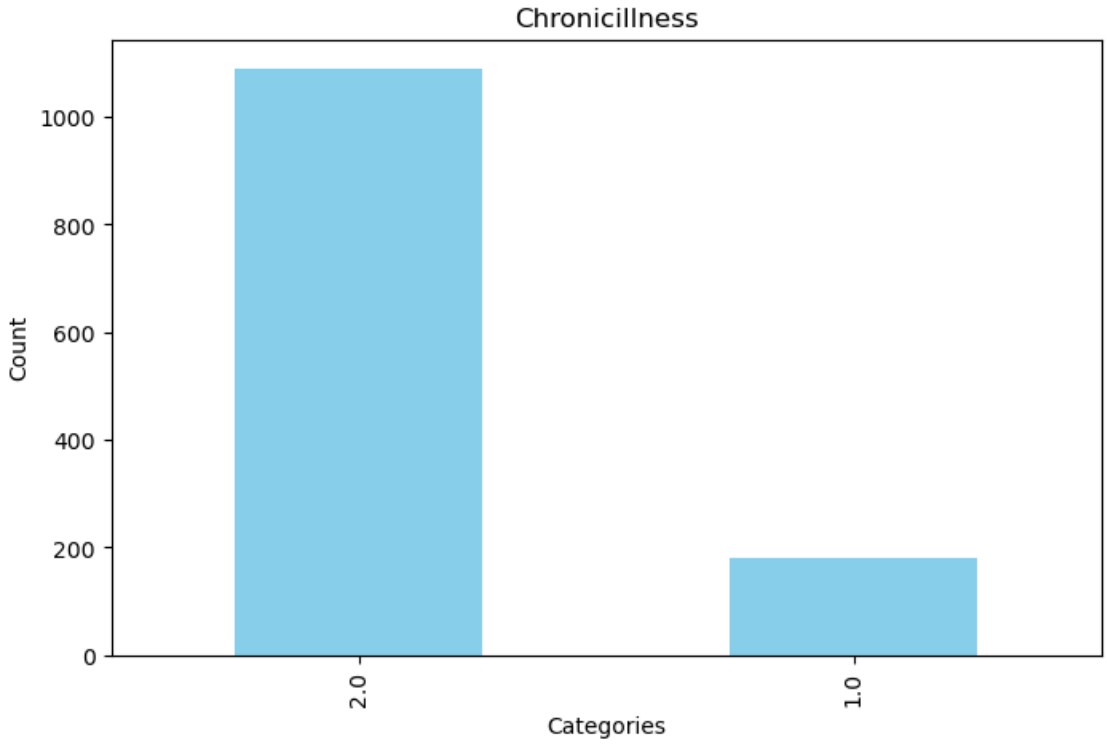

Education
Count
500
400
300
200
100
0
11.0
12.0
8.0
10.0
13.0
9.0
5.0
6.0
7.0
3.0
4.0
0.0
2.0
1.0
Categories

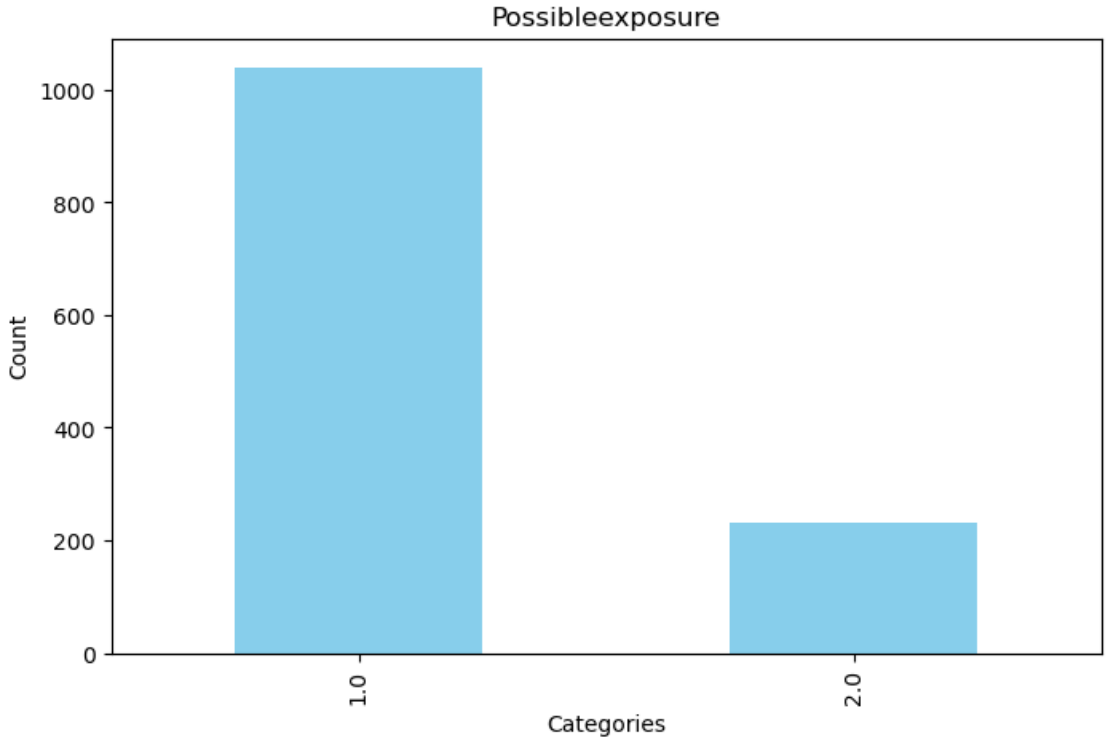

TertiaryEducation
1200
1000
800
600
400
200
0
Count
3.0
1.0
2.0
Categories

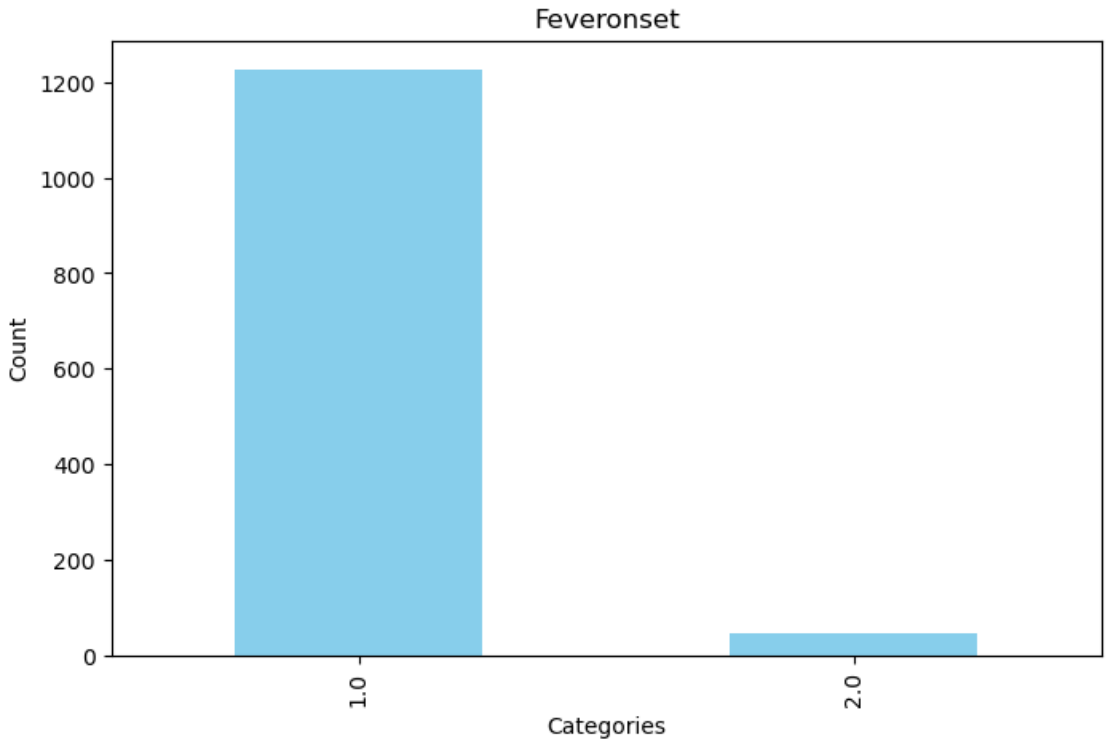

Prophylactics
1000
800
600
400
200
0
Count
2.0
3.0
1.0
Categories

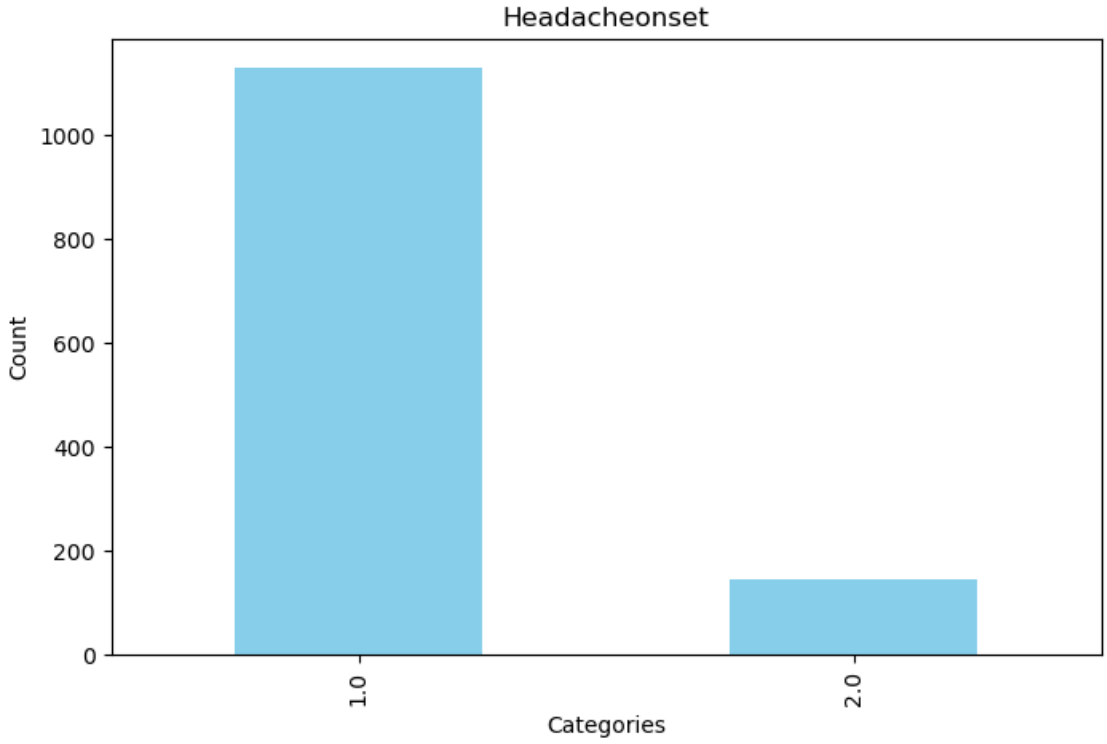

Pasttreatments
800
600
400
200
0
Count
1.0
2.0
Categories

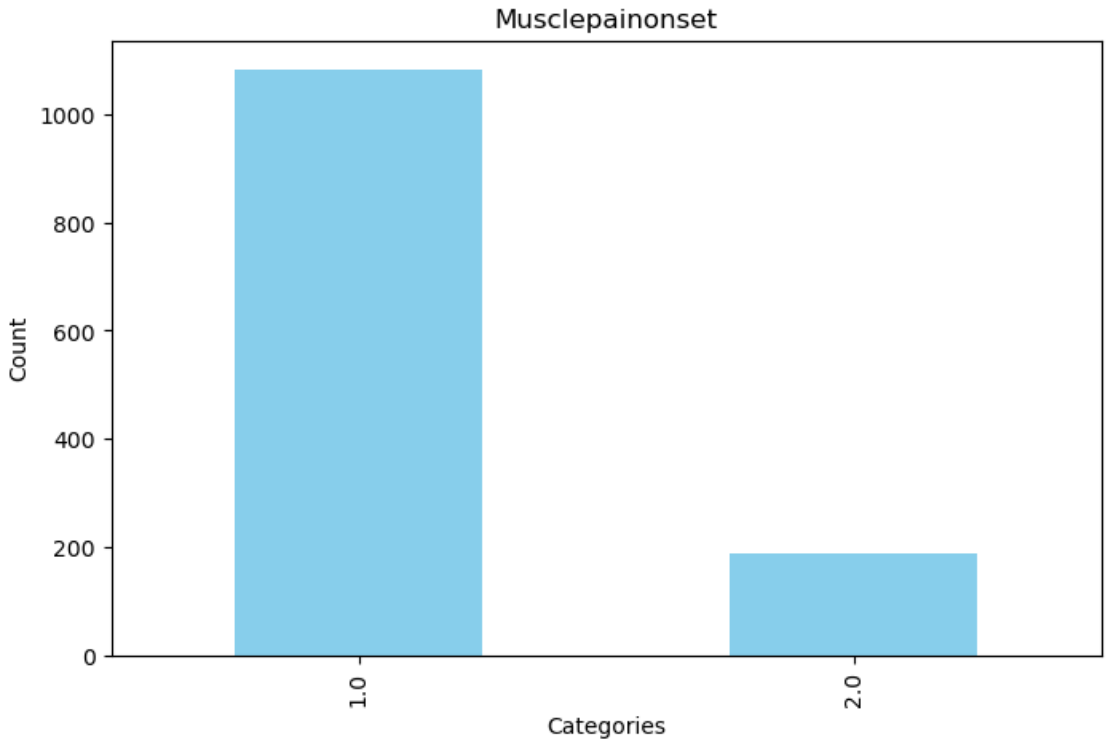

Pastantibiotics
700
600
500
400
300
200
100
0
Count
2.0
1.0
3.0
Categories

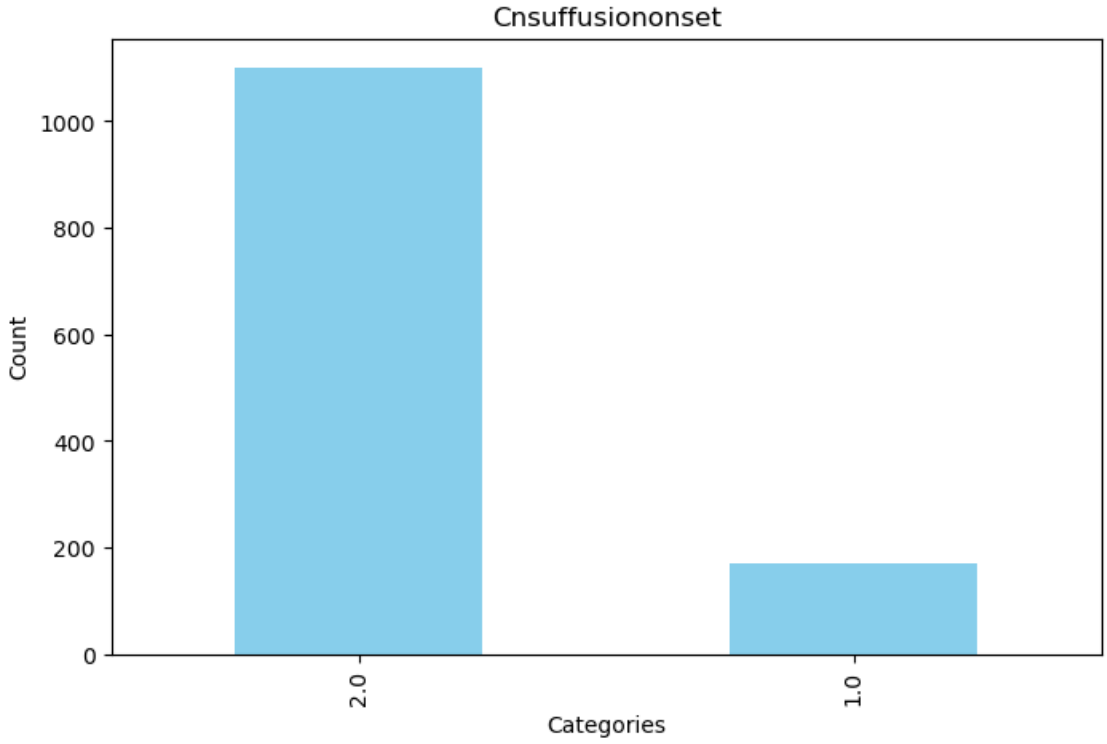

Chronicillness
Count
1000
800
600
400
200
0
2.0
1.0
Categories

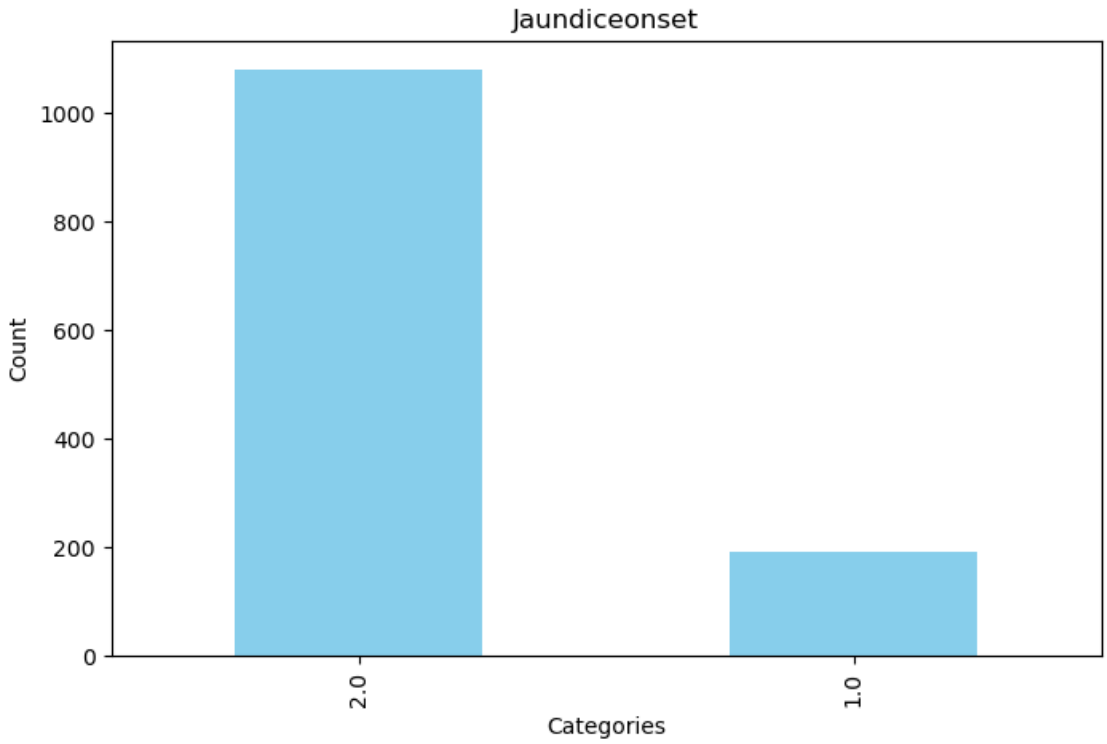

Possibleexposure
Count
1000
800
600
400
200
0
1.0
2.0
Categories

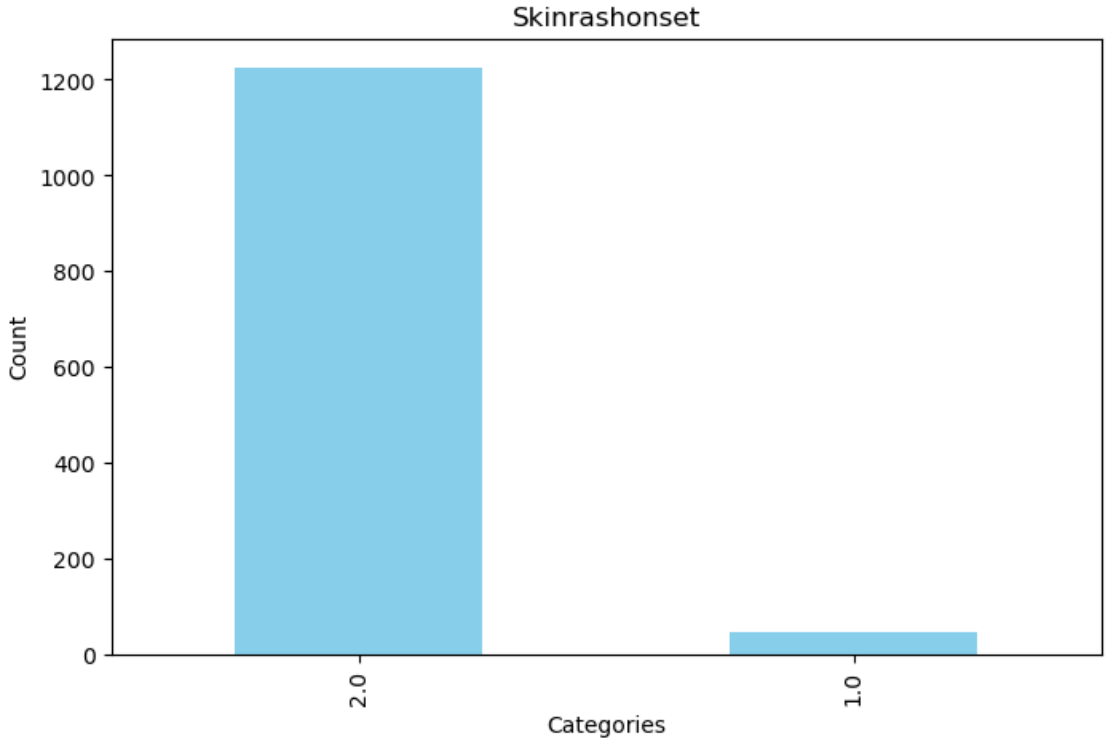

Feveronset
1200
1000
800
600
400
200
0
Count
1.0
2.0
Categories

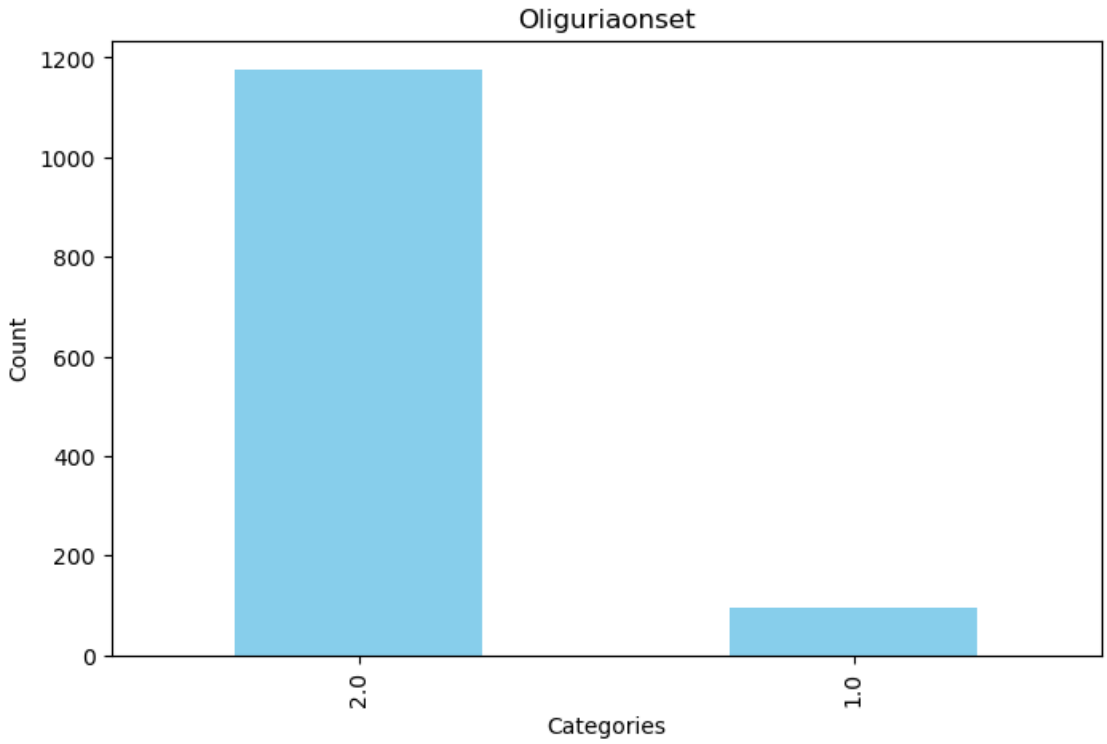

Headacheonset
1000
800
600
400
200
0
Count
1.0
2.0
Categories

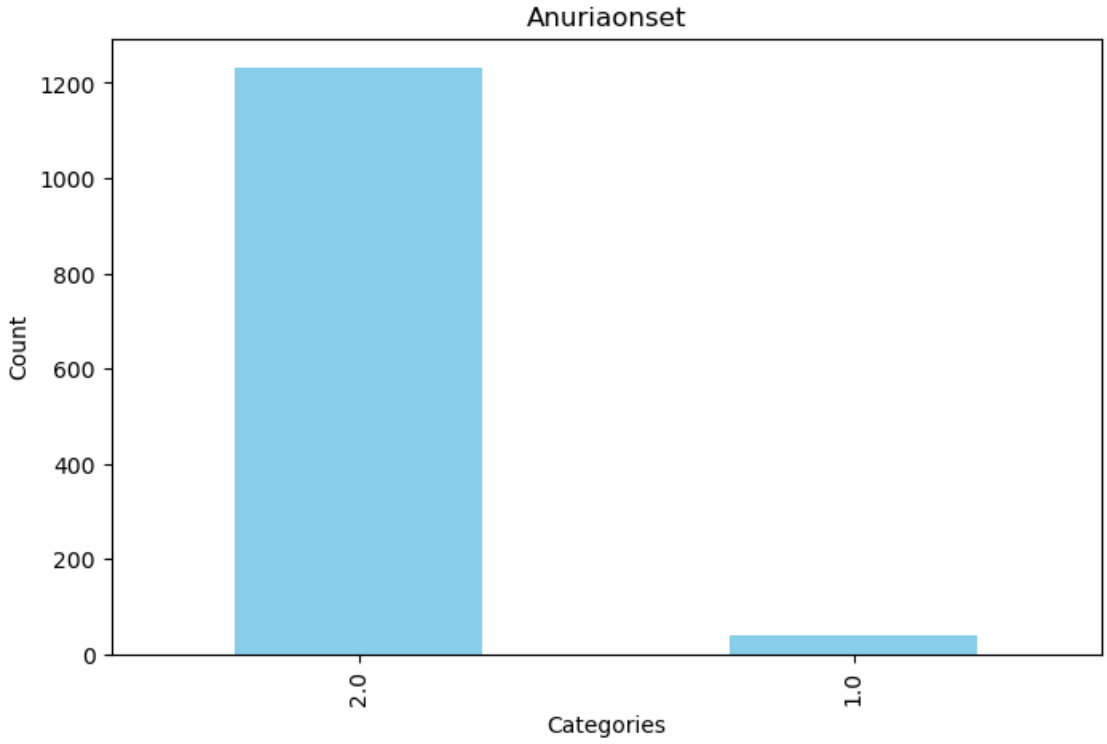

Musclepainonset
Count
1000
800
600
400
200
0
1.0
2.0
Categories

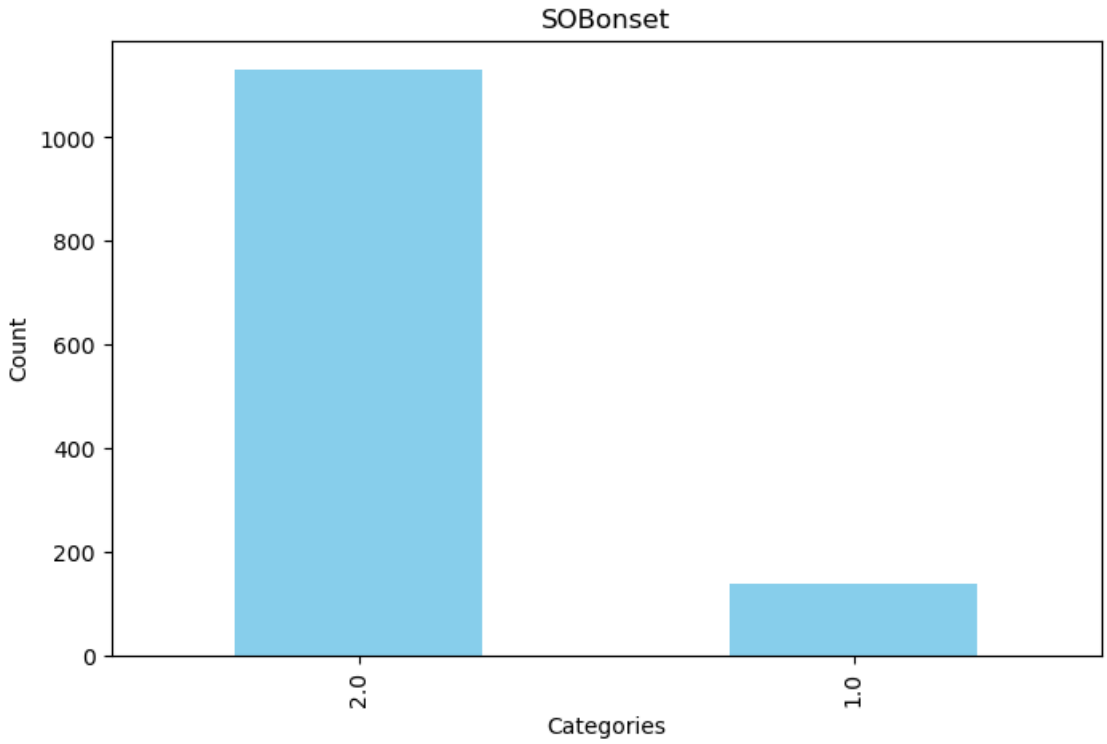

Cnsuffusiononset
Count
1000
800
600
400
200
0
2.0
1.0
Categories

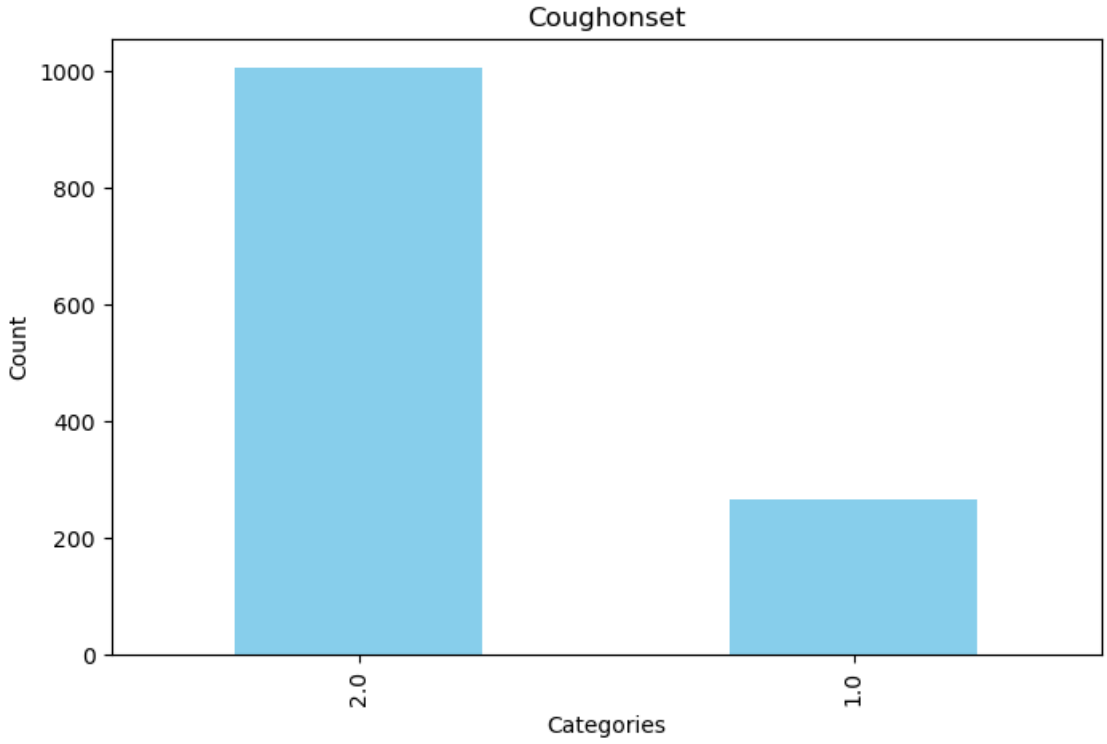

Jaundiceonset
Count
0
200
400
600
800
1000
2.0
1.0
Categories

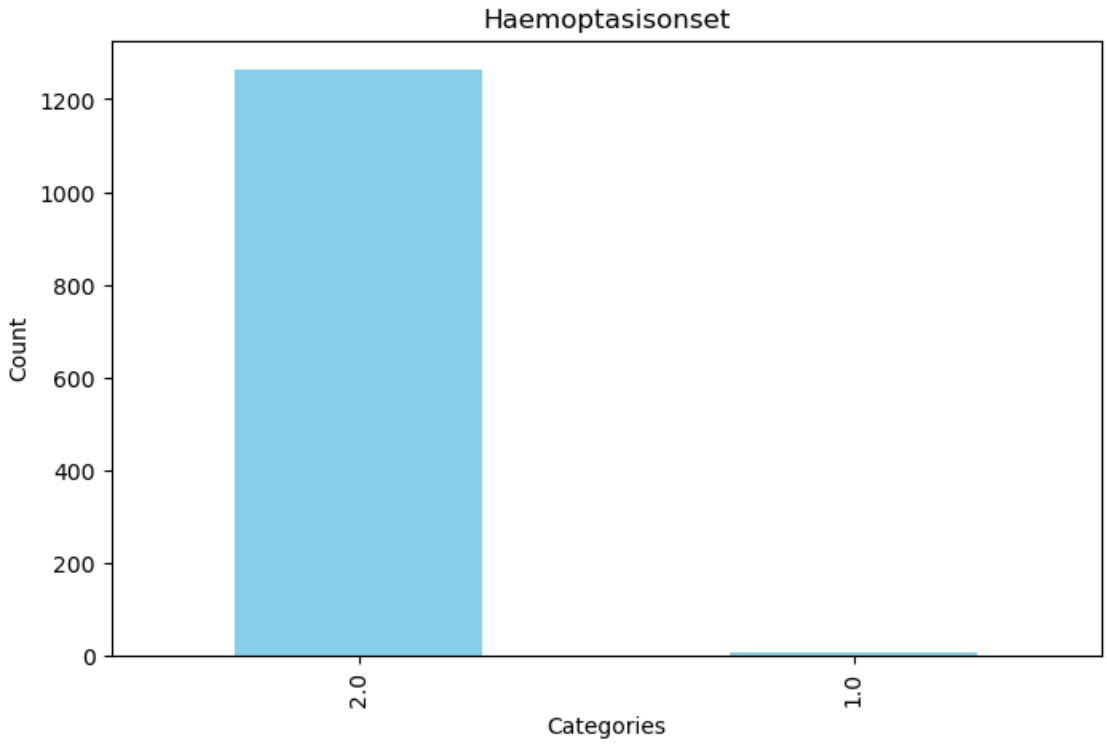

Skinrashonset
Count
0
200
400
600
800
1000
1200
2.0
1.0
Categories

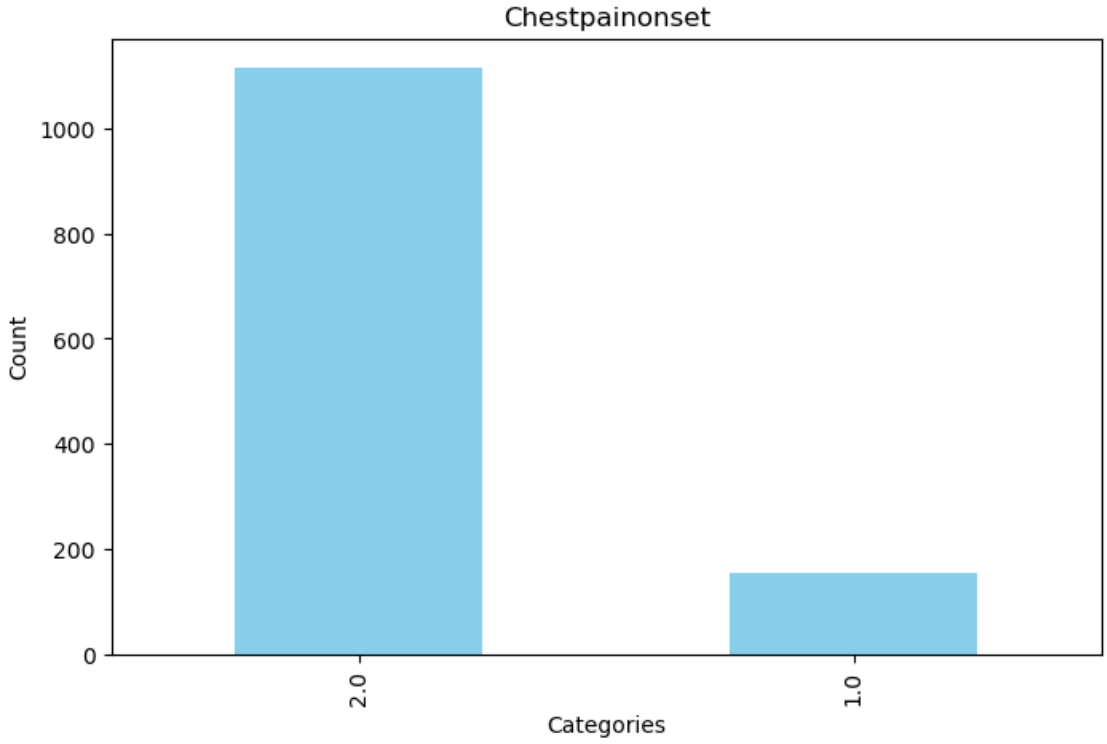

Oliguriaonset
1200
1000
800
600
400
200
0
Count
2.0
1.0
Categories

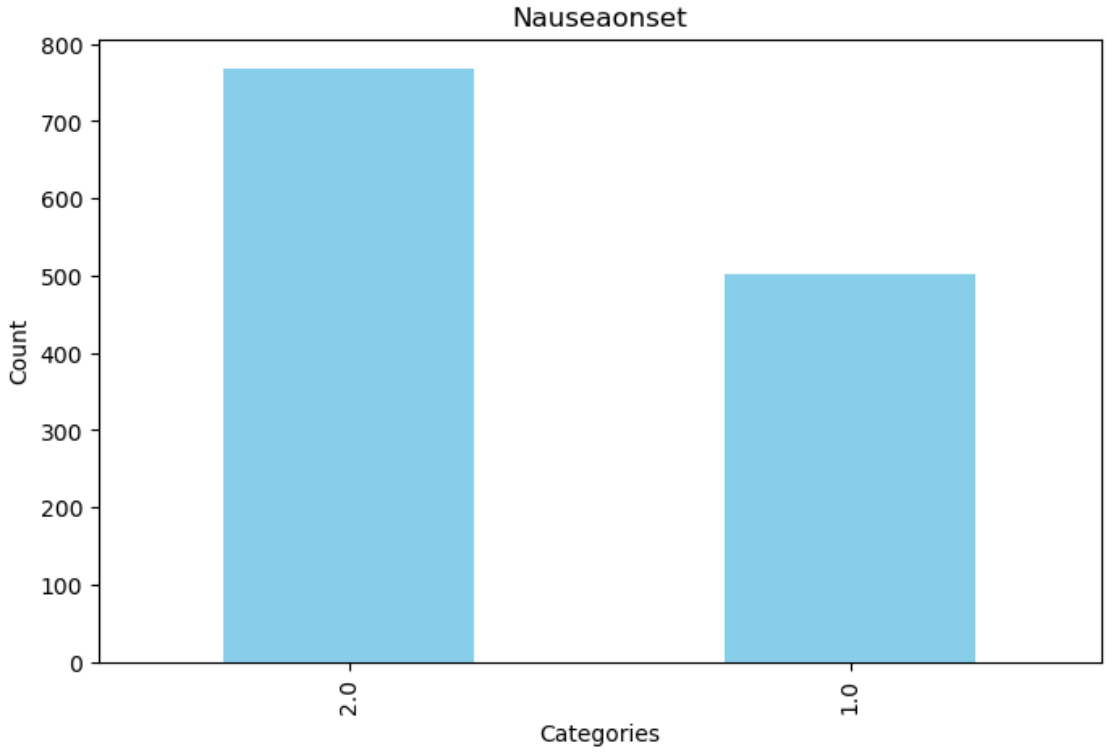

Anuriaonset
1200
1000
800
600
400
200
0
Count
2.0
1.0
Categories

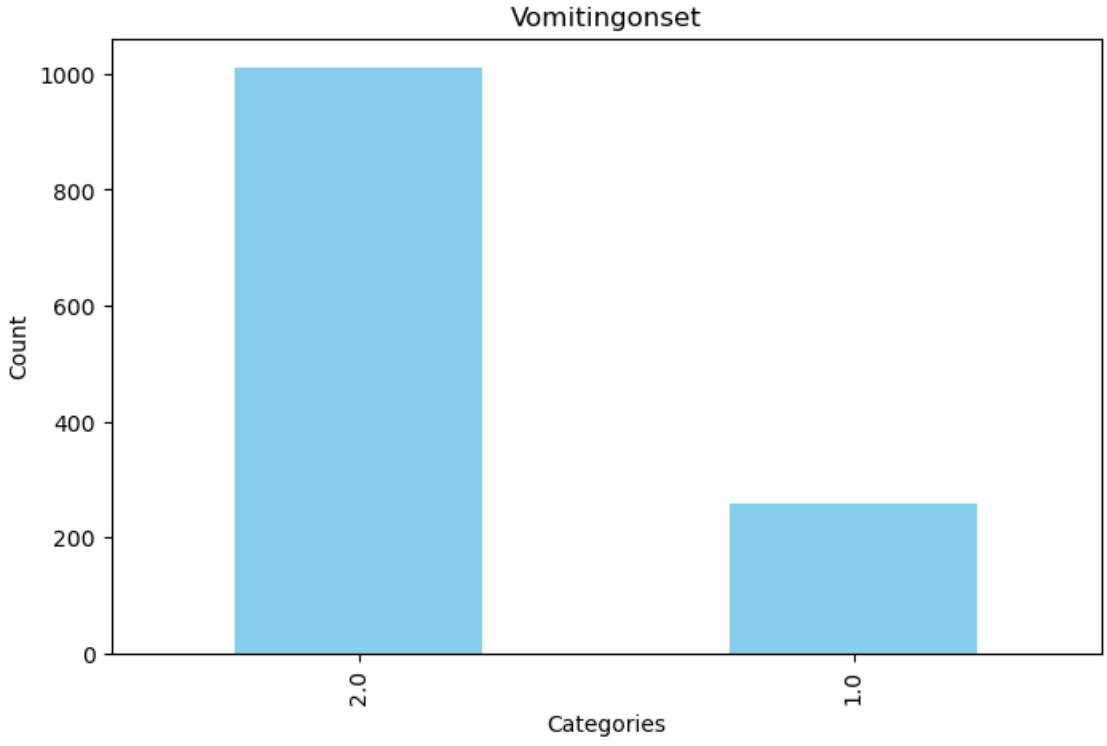

SOBonset
Count
1000
800
600
400
200
0
2.0
1.0
Categories

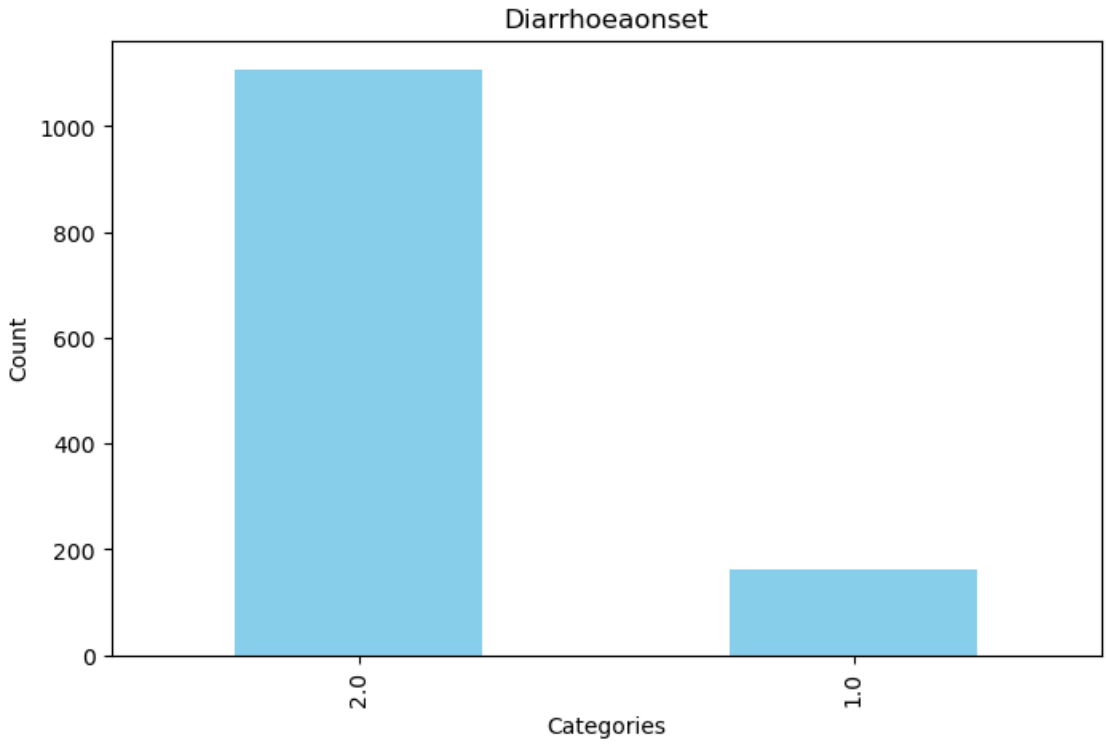

Coughonset
Count
1000
800
600
400
200
0
2.0
1.0
Categories

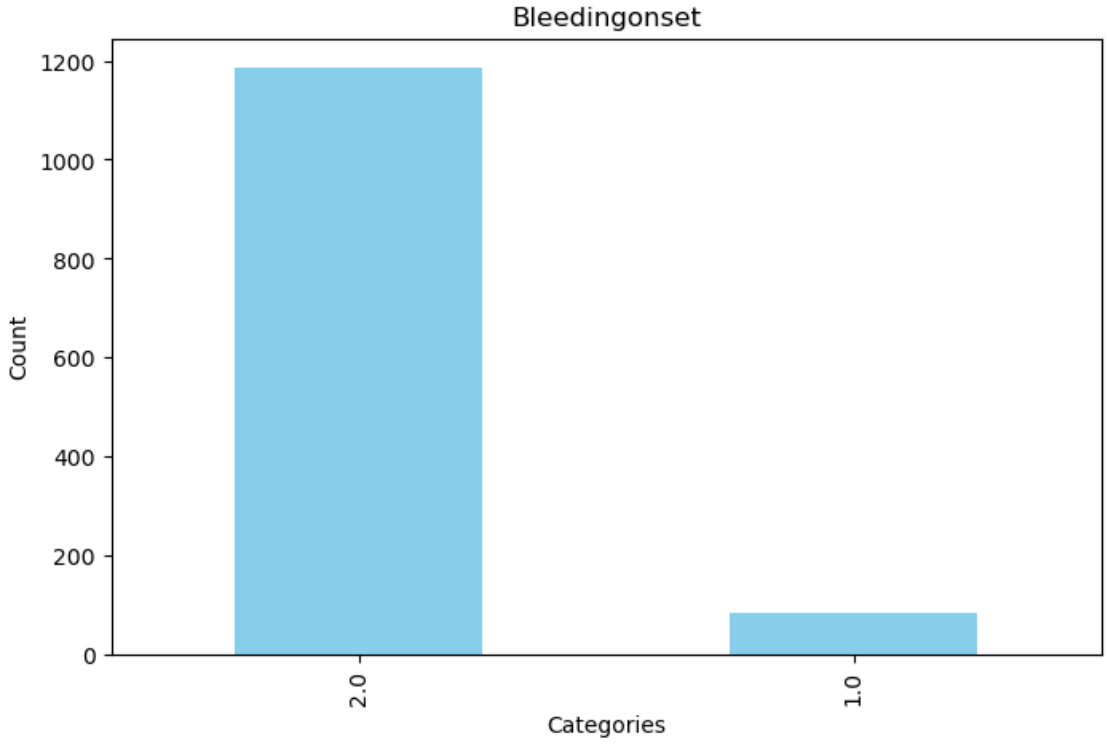

Haemoptasisonset
Count
1200
1000
800
600
400
200
0
2.0
1.0
Categories

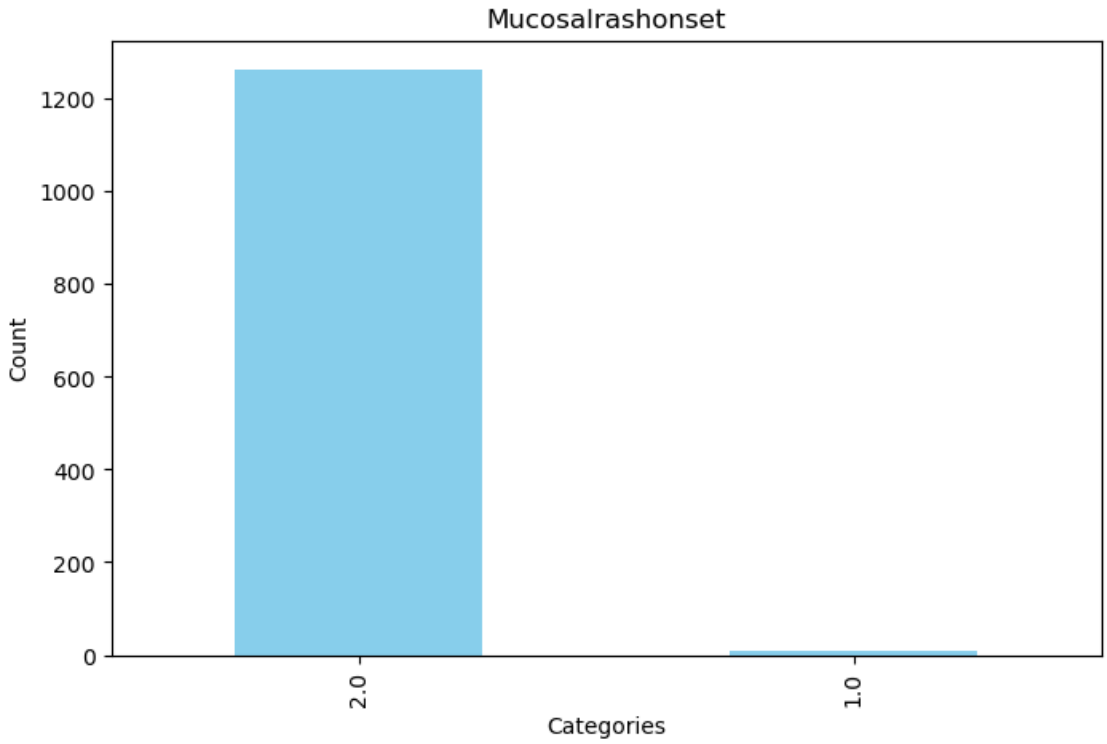

Chestpainonset
Count
1000
800
600
400
200
0
2.0
1.0
Categories

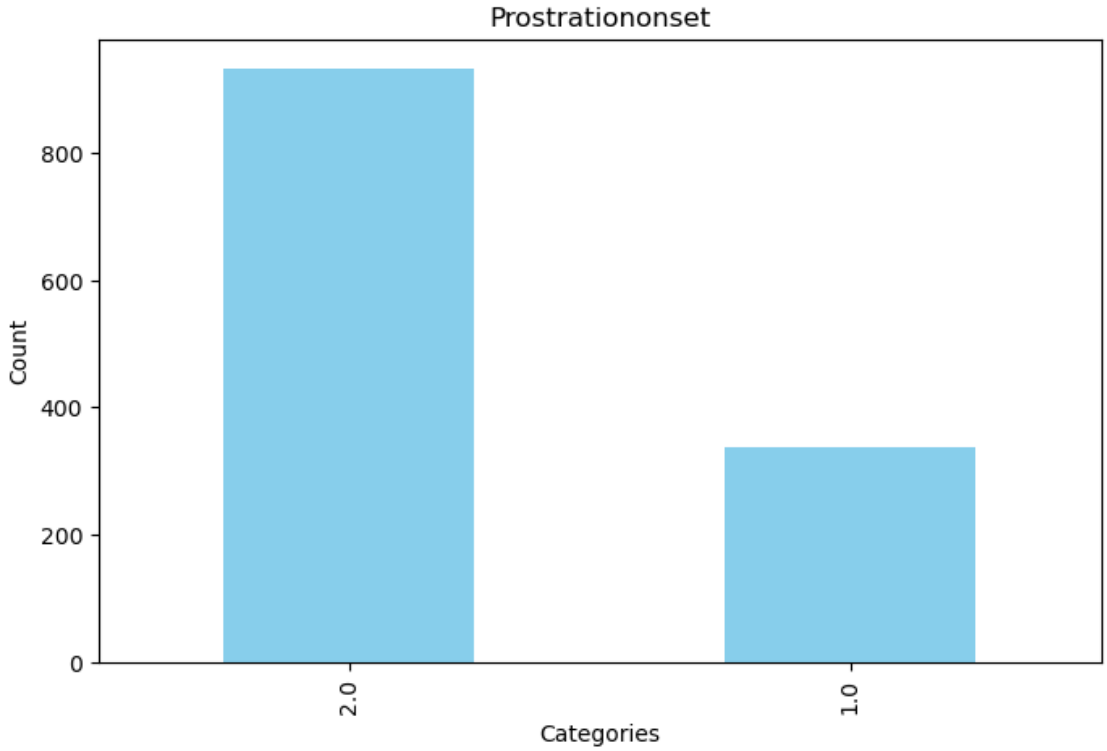

Nauseaonset
800
700
600
500
400
300
200
100
0
Count
2.0
1.0
Categories

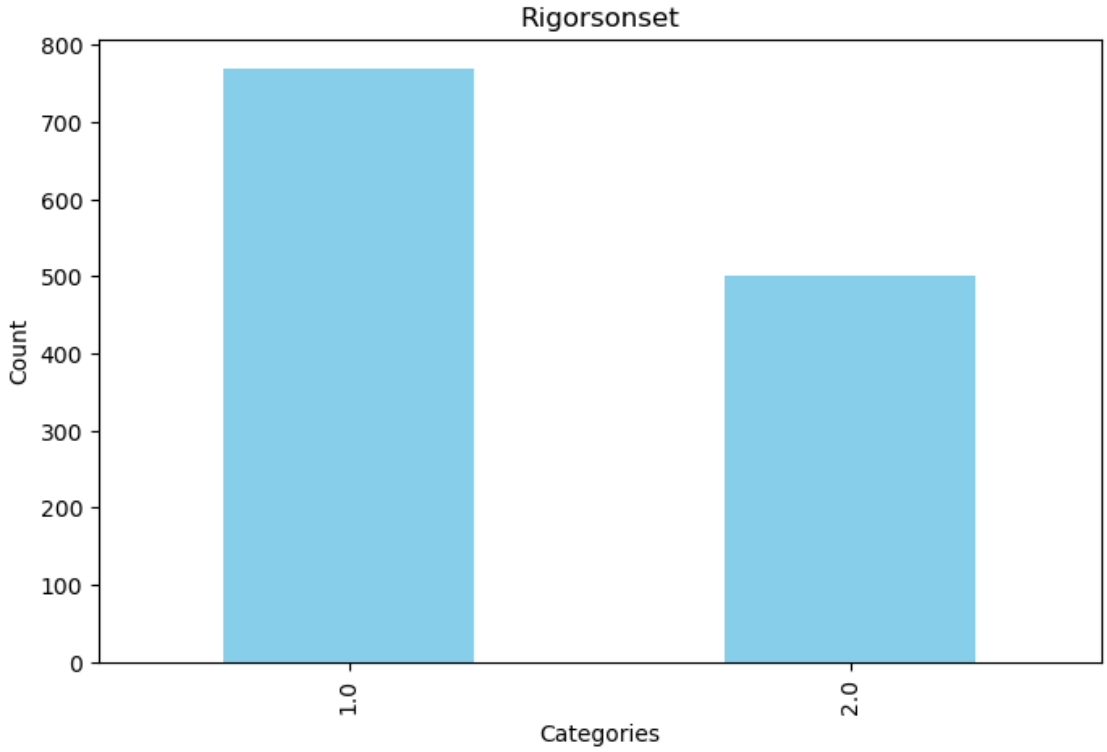

Vomitingonset
1000
800
600
400
200
0
Count
2.0
1.0
Categories

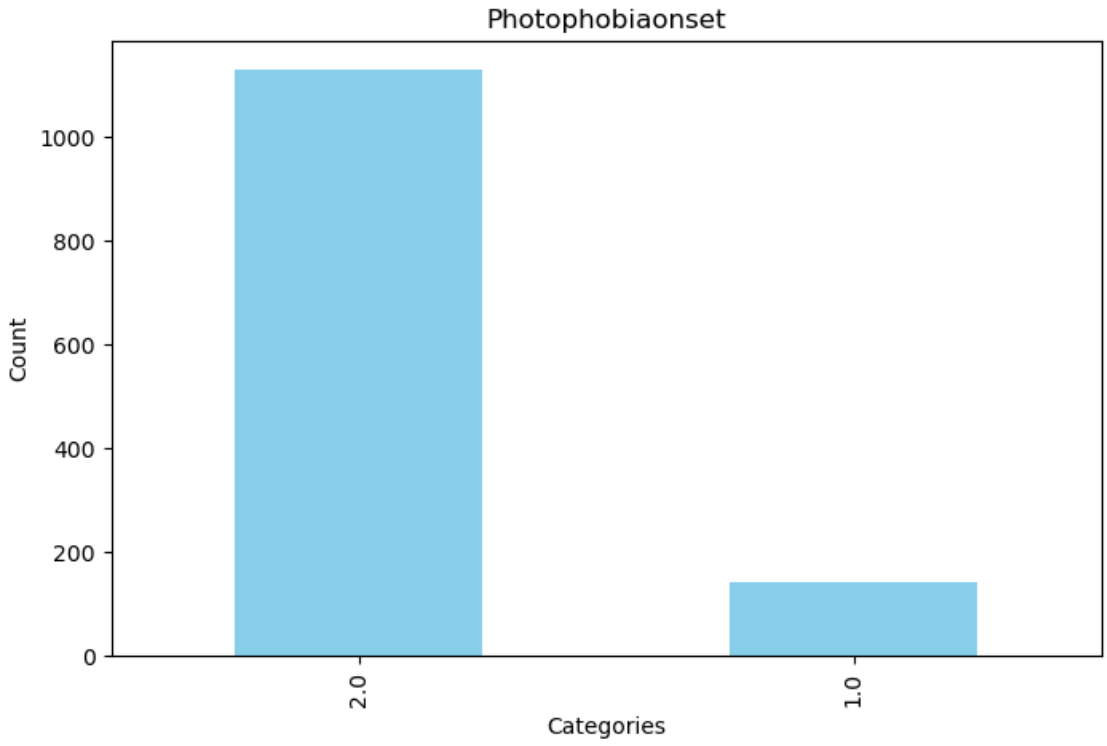

Diarrhoeaonset
Count
1000
800
600
400
200
0
2.0
1.0
Categories

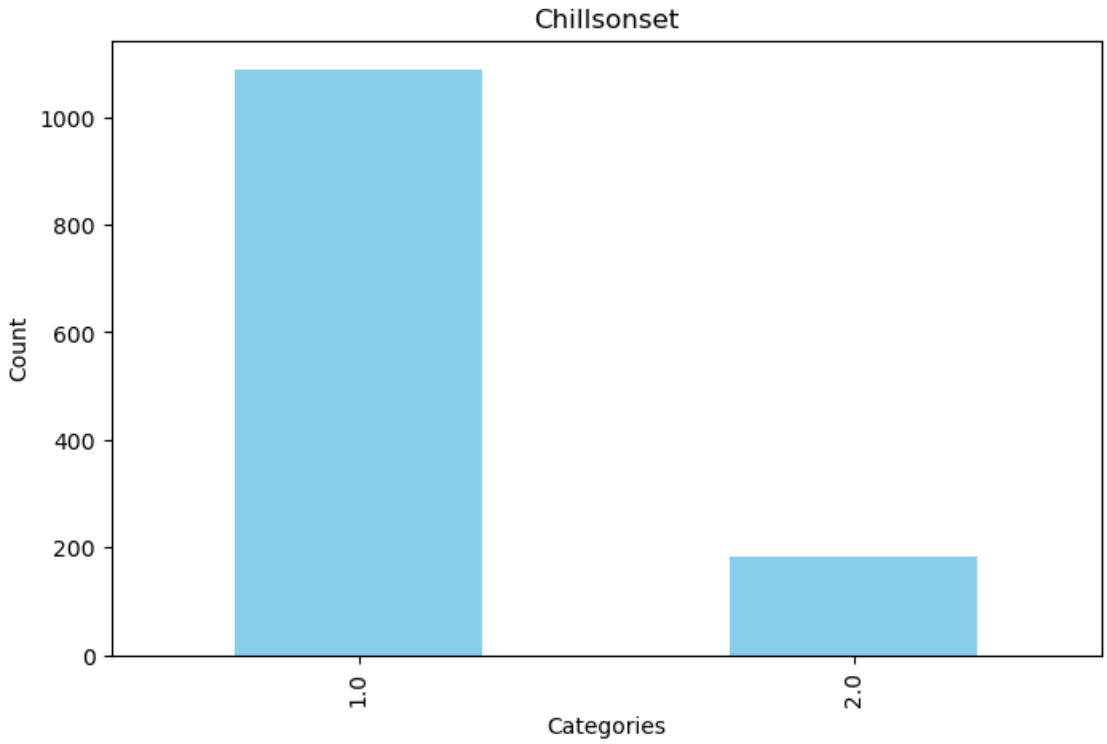

Bleedingonset
Count
1200
1000
800
600
400
200
0
2.0
1.0
Categories

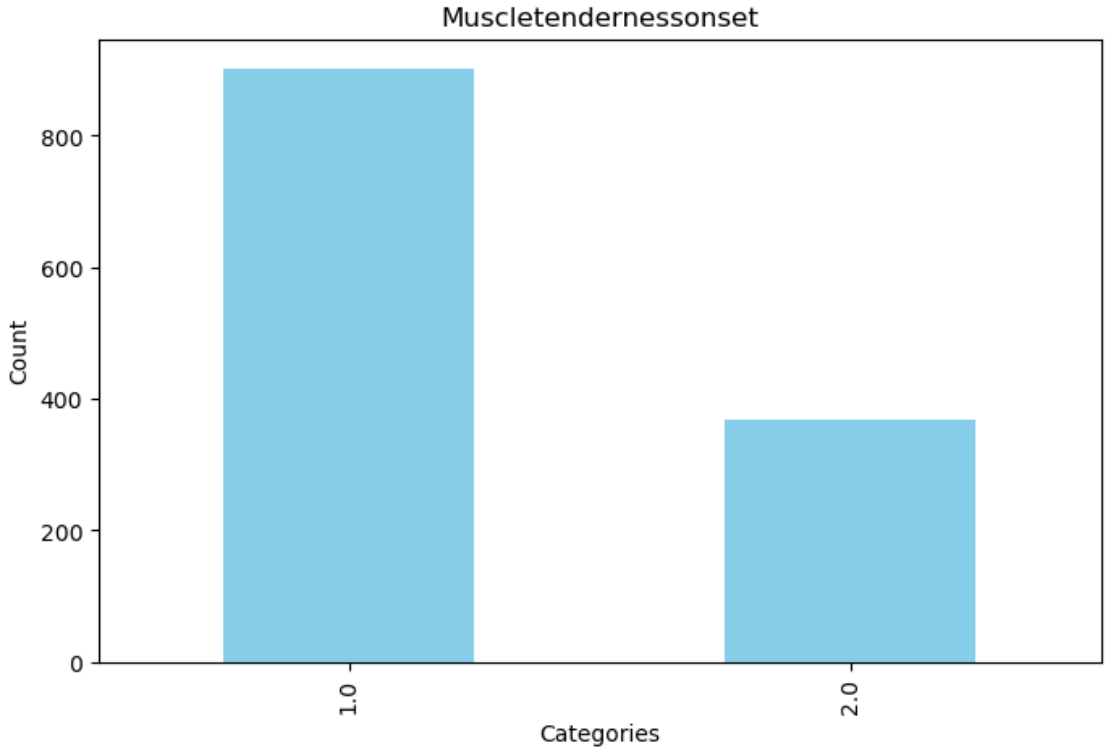

MucosaIrashonset
1200
1000
800
600
400
200
0
Count
2.0
1.0
Categories

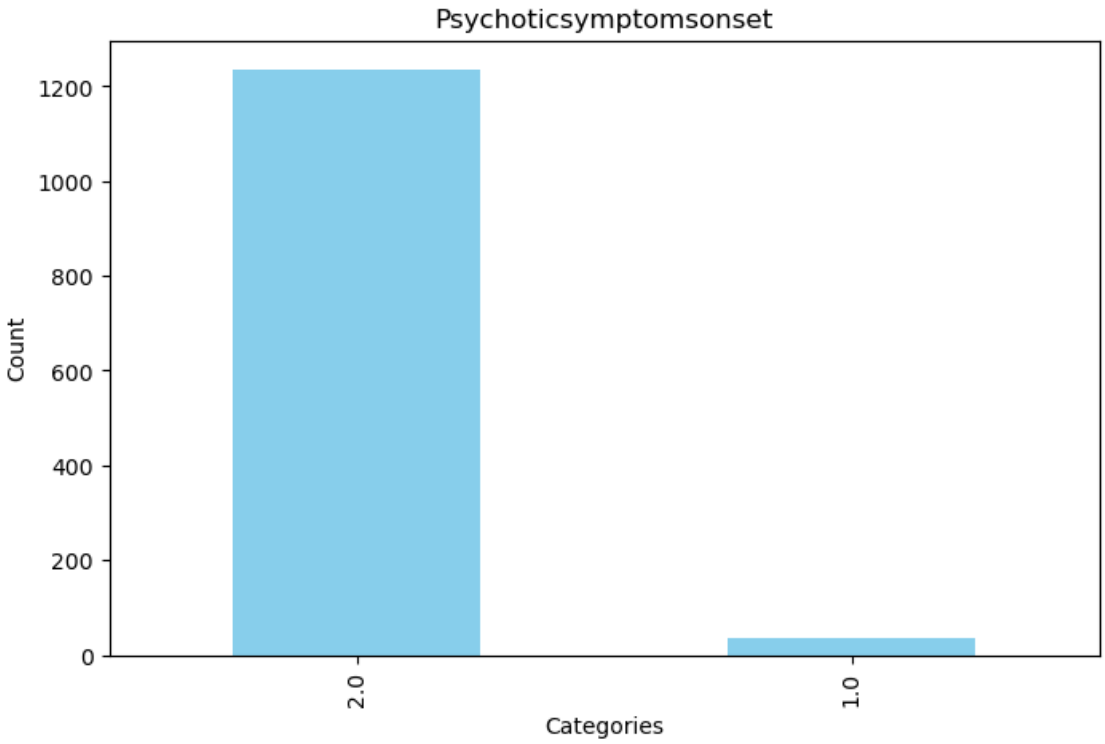

Prostrationonset
800
600
400
200
0
Count
2.0
1.0
Categories

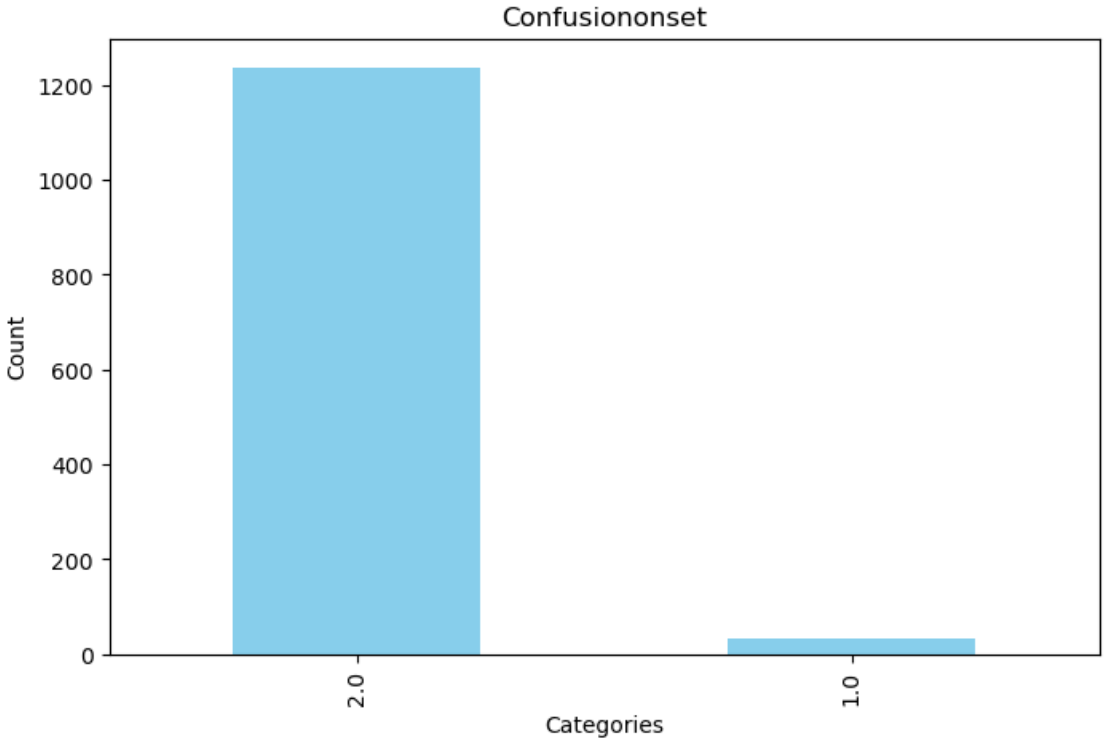

Rigorsonset
800
700
600
500
400
300
200
100
0
Count
1.0
2.0
Categories

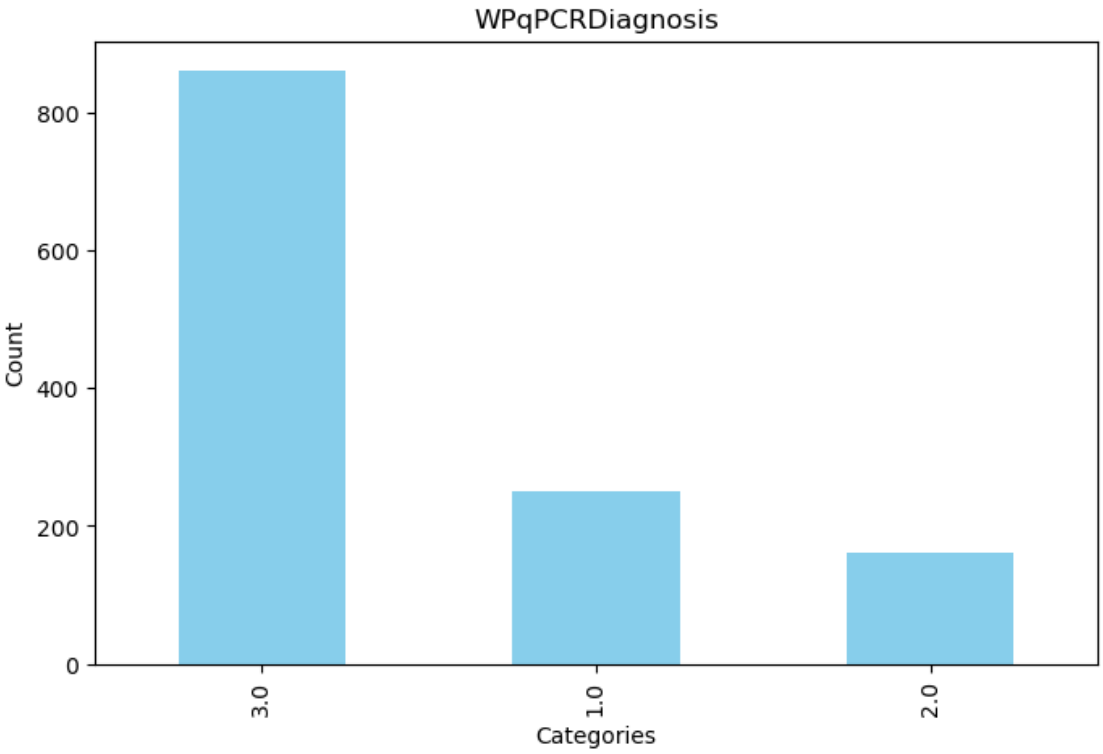

Photophobiaonset
1000
800
600
400
200
0
Count
2.0
1.0
Categories

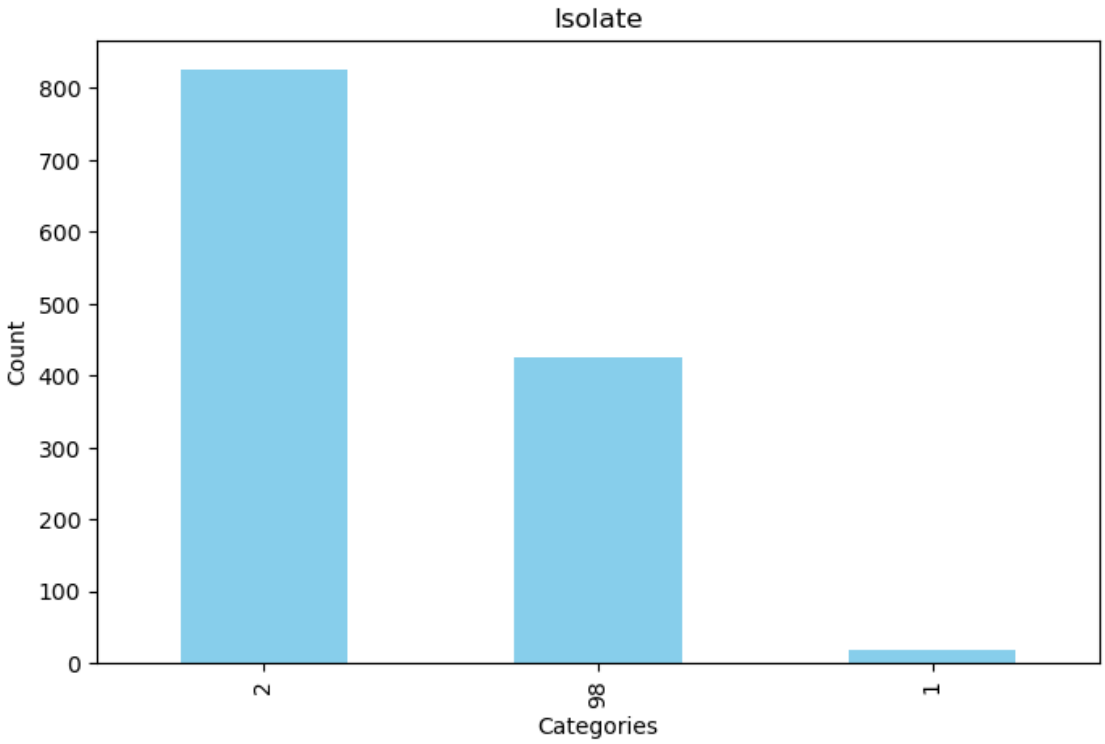

Chillsonset
1000
800
600
400
200
0
Count
1.0
2.0
Categories

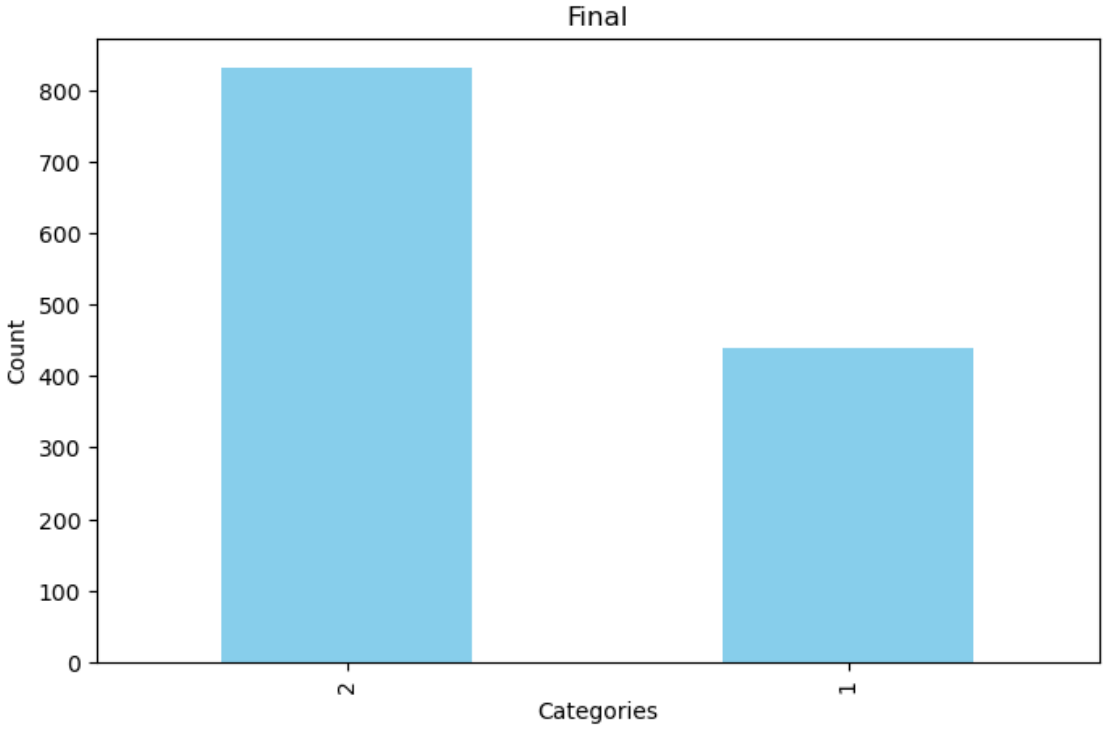

Muscletendernessonset
800
600
400
200
0
Count
1.0
2.0
Categories


Psychoticsymptomsonset
1200
1000
800
600
400
200
0
Count
2.0
1.0
Categories

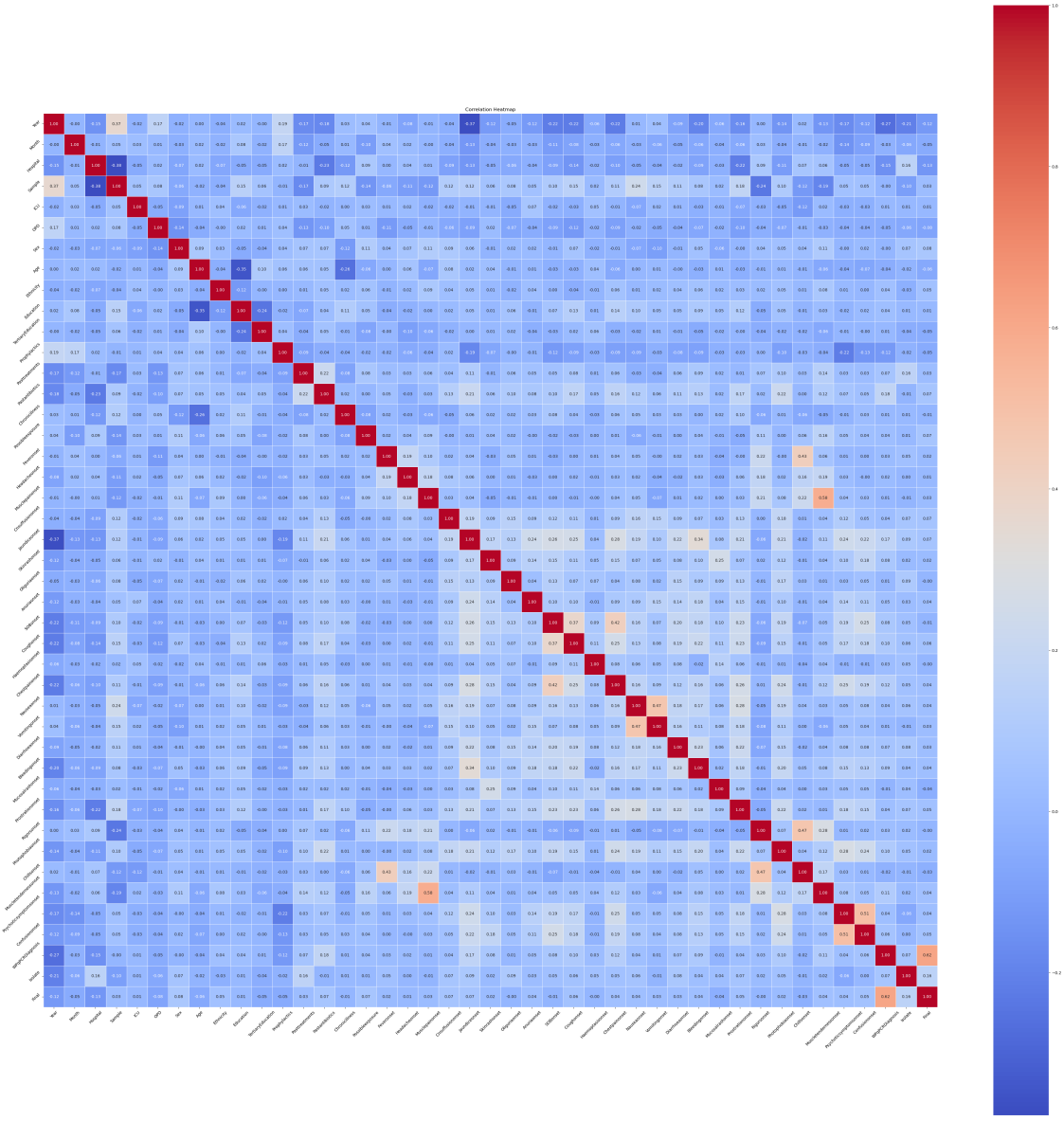

Confusiononset
1200
1000
800
600
400
200
0
Count
2.0
1.0
Categories


WPqPCRDiagnosis
800
600
400
200
0
Count
3.0
1.0
2.0
Categories


Isolate
Count
800
700
600
500
400
300
200
100
0
2
98
1
Categories


Final
Count
800
700
600
500
400
300
200
100
0
2
1
Categories

[28]:
```
# Compute correlation matrix

#********************* Double Click on the Correlation Map to see Better␣
 ↪Description *********************
corr_matrix = df.corr()
plt.figure(figsize=(50, 50))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5,␣
 ↪square=True)

plt.xticks(rotation=45)
plt.yticks(rotation=45)

plt.title('Correlation Heatmap')
plt.show()
```



2 b)

2.1 Fitting a Model

[29]:
```
#splitting into training and testing
X=df.drop('Final',axis=1)
y=df['Final']
```

[30]:
```
cat_col_x = X.select_dtypes(include='category').columns.tolist()
num_col_x=  X.select_dtypes(include=['Int64', 'float64']).columns.tolist()
```

[31]:
```
#transforming
transformer_num=Pipeline(steps=[('scaler',StandardScaler())])
transformer_cat=Pipeline(steps=[('encoder',OrdinalEncoder())])
```

[32]:
```
#applying transformer to features
preprocessor=ColumnTransformer(transformers=[('numeric',transformer_num,num_col_x),('category'
```

[33]:
```
Logistic_Model=Pipeline(steps=[('preprocessor',preprocessor),('classifier',LogisticRegression(
```

[34]:
```
# Fit the model
Logistic_Model.fit(X, y)

# Evaluate the model
accuracy = Logistic_Model.score(X, y)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8307086614173228
```

[35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1270 entries, 0 to 1386
Data columns (total 43 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Year                    1270 non-null   category
 1   Month                   1270 non-null   category
 2   Hospital                1270 non-null   category
 3   Sample                  1270 non-null   category
 4   ICU                     1270 non-null   category
 5   OPD                     1270 non-null   category
 6   Sex                     1270 non-null   category
 7   Age                     1270 non-null   float64
```

```
 8   Ethnicity                1270 non-null   category
 9   Education                1270 non-null   category
 10  TertiaryEducation        1270 non-null   category
 11  Prophylactics            1270 non-null   category
 12  Pasttreatments           1270 non-null   category
 13  Pastantibiotics          1270 non-null   category
 14  Chronicillness           1270 non-null   category
 15  Possibleexposure         1270 non-null   category
 16  Feveronset               1270 non-null   category
 17  Headacheonset            1270 non-null   category
 18  Musclepainonset          1270 non-null   category
 19  Cnsuffusiononset         1270 non-null   category
 20  Jaundiceonset            1270 non-null   category
 21  Skinrashonset            1270 non-null   category
 22  Oliguriaonset            1270 non-null   category
 23  Anuriaonset              1270 non-null   category
 24  SOBonset                 1270 non-null   category
 25  Coughonset               1270 non-null   category
 26  Haemoptasisonset         1270 non-null   category
 27  Chestpainonset           1270 non-null   category
 28  Nauseaonset              1270 non-null   category
 29  Vomitingonset            1270 non-null   category
 30  Diarrhoeaonset           1270 non-null   category
 31  Bleedingonset            1270 non-null   category
 32  Mucosalrashonset         1270 non-null   category
 33  Prostrationonset         1270 non-null   category
 34  Rigorsonset              1270 non-null   category
 35  Photophobiaonset         1270 non-null   category
 36  Chillsonset              1270 non-null   category
 37  Muscletendernessonset    1270 non-null   category
 38  Psychoticsymptomsonset   1270 non-null   category
 39  Confusiononset           1270 non-null   category
 40  WPqPCRDiagnosis          1270 non-null   category
 41  Isolate                  1270 non-null   category
 42  Final                    1270 non-null   category
dtypes: category(42), float64(1)
memory usage: 78.2 KB
```

[36]:
```
print(df.isnull().sum().sum())
```

```
0
```

3 c)

3.1 Data preprocessing for test dataset

3.2 Loading a Test Data Set

[37]:
```
df_test = pd.read_csv('test.csv',engine='python')
df_test.head()
```

[37]:

	ID	Year	Month	Hospital	Sample	ICU	OPD	Sex	Age	Ethnicity	…	\
0	1	2017	6	1	1	2	2	1	49	1	…	
1	2	2017	6	1	1	2	2	1	47	1	…	
2	3	2017	6	1	1	2	2	1	51	1	…	
3	4	2017	6	1	1	2	2	2	37	1	…	
4	5	2017	6	1	1	2	1	1	99	1	…	

	FU_L.interrogansserovarIcterohaemorrhagiaestr.RGA	\
0	NaN	
1	NaN	
2	NaN	
3	NaN	
4	NaN	

	FU_L.interrogansserovarMankarsostr.Mankarso	\
0	NaN	
1	NaN	
2	NaN	
3	NaN	
4	NaN	

	FU_L.santarosaiserovarGeorgiastr.LT117	\
0	NaN	
1	NaN	
2	NaN	
3	NaN	
4	NaN	

	FU_L.santarosaiserovarPyrogenesstr.Salinem	\
0	NaN	
1	NaN	
2	NaN	
3	NaN	
4	NaN	

	FU_L.interrogansserovarBataviaestr.VanTienan	\
0	NaN	
1	NaN	
2	NaN	

```
3                                                 NaN
4                                                 NaN

   FU_L.interrogansserovarAlexistr.616  \
0                                  NaN
1                                  NaN
2                                  NaN
3                                  NaN
4                                  NaN

   FU_L.interrogansserovarAustralisstr.Ballico  \
0                                          NaN
1                                          NaN
2                                          NaN
3                                          NaN
4                                          NaN

   FU_L.interrogansserovarwolfiistr.3705  FU_L.interrogansserovarWeerasinghe  \
0                                    NaN                                NaN
1                                    NaN                                NaN
2                                    NaN                                NaN
3                                    NaN                                NaN
4                                    NaN                                NaN

   FU_Patoc
0       NaN
1       NaN
2       NaN
3       NaN
4       NaN

[5 rows x 805 columns]
```

3.3 Handling missing values in test dataset

[38]:
```
df1 = df_test.drop('ID', axis=1)
```

[39]:
```
df1.replace(['99', 99], np.nan, inplace=True)
```

[40]:
```
df1.isna().sum()
```

[40]:
```
Year                                    0
Month                                   0
Hospital                                0
Sample                                  0
ICU                                    19
                                      ...
```

```
FU_L.interrogansserovarAlexistr.616            317
FU_L.interrogansserovarAustralisstr.Ballico    317
FU_L.interrogansserovarwolfiistr.3705          317
FU_L.interrogansserovarWeerasinghe             317
FU_Patoc                                       317
Length: 804, dtype: int64
```

[41]:
```
missing_percentage1 = df1.isnull().mean() * 100
print(missing_percentage1)
```

```
Year                                          0.000000
Month                                         0.000000
Hospital                                      0.000000
Sample                                        0.000000
ICU                                           5.475504
                                                ...
FU_L.interrogansserovarAlexistr.616          91.354467
FU_L.interrogansserovarAustralisstr.Ballico  91.354467
FU_L.interrogansserovarwolfiistr.3705        91.354467
FU_L.interrogansserovarWeerasinghe           91.354467
FU_Patoc                                     91.354467
Length: 804, dtype: float64
```

[42]:
```
print(missing_percentage1[missing_percentage1 > 30])
```

```
Income                                       33.429395
Usualdrinkingwatersource                     73.775216
Usualbathingwatersource                      73.775216
Sourceofwaterforhousehold                    73.775216
Garbagedisposalprocedure                     73.775216
                                               ...
FU_L.interrogansserovarAlexistr.616          91.354467
FU_L.interrogansserovarAustralisstr.Ballico  91.354467
FU_L.interrogansserovarwolfiistr.3705        91.354467
FU_L.interrogansserovarWeerasinghe           91.354467
FU_Patoc                                     91.354467
Length: 733, dtype: float64
```

[43]:
```
threshold = 30
```

[44]:
```
cols_to_drop1 = missing_percentage1[missing_percentage1 > threshold].index
```

[45]:
```
df1 = df1.drop(columns=cols_to_drop1)
```

[46]:
```
# print(f"Dropped columns: {cols_to_drop1.tolist()}")
```

[47]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 347 entries, 0 to 346
Data columns (total 71 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Year                     347 non-null    int64
 1   Month                    347 non-null    int64
 2   Hospital                 347 non-null    int64
 3   Sample                   347 non-null    int64
 4   ICU                      328 non-null    float64
 5   OPD                      328 non-null    float64
 6   Sex                      312 non-null    float64
 7   Age                      304 non-null    float64
 8   Ethnicity                312 non-null    float64
 9   Education                257 non-null    float64
 10  TertiaryEducation        257 non-null    float64
 11  Prophylactics            269 non-null    float64
 12  Pasttreatments           272 non-null    float64
 13  Pastantibiotics          270 non-null    float64
 14  Chronicillness           268 non-null    float64
 15  Possibleexposure         270 non-null    float64
 16  Feveronset               263 non-null    float64
 17  Headacheonset            257 non-null    float64
 18  Musclepainonset          263 non-null    float64
 19  Cnsuffusiononset         263 non-null    float64
 20  Jaundiceonset            263 non-null    float64
 21  Skinrashonset            263 non-null    float64
 22  Oliguriaonset            263 non-null    float64
 23  Anuriaonset              263 non-null    float64
 24  SOBonset                 263 non-null    float64
 25  Coughonset               263 non-null    float64
 26  Haemoptasisonset         263 non-null    float64
 27  Chestpainonset           263 non-null    float64
 28  Nauseaonset              263 non-null    float64
 29  Vomitingonset            263 non-null    float64
 30  Diarrhoeaonset           263 non-null    float64
 31  Bleedingonset            262 non-null    float64
 32  Mucosalrashonset         263 non-null    float64
 33  Prostrationonset         263 non-null    float64
 34  Rigorsonset              263 non-null    float64
 35  Photophobiaonset         263 non-null    float64
 36  Chillsonset              263 non-null    float64
 37  Muscletendernessonset    263 non-null    float64
 38  Psychoticsymptomsonset   263 non-null    float64
 39  Confusiononset           263 non-null    float64
 40  Feverad                  247 non-null    float64
 41  Headachead               247 non-null    float64
 42  Chillsad                 246 non-null    float64
```

```
 43  Rigorsad                247 non-null    float64
 44  Musclepainad            247 non-null    float64
 45  Muscletendernessad      247 non-null    float64
 46  Nauseaad                247 non-null    float64
 47  Vomitingadmission       247 non-null    float64
 48  Cnsuffusionad           247 non-null    float64
 49  Skinrashad              247 non-null    float64
 50  Mucosalrashad           247 non-null    float64
 51  Prostrationad           247 non-null    float64
 52  Diarrhoeaad             247 non-null    float64
 53  OliguriaAd              248 non-null    float64
 54  Anuriaad                247 non-null    float64
 55  Jaundicead              248 non-null    float64
 56  Hepatictendernessad     247 non-null    float64
 57  Hepatomegalyad          248 non-null    float64
 58  Spleenimegalyad         247 non-null    float64
 59  Lympadenopathyad        247 non-null    float64
 60  Photophobiaad           247 non-null    float64
 61  Neckstiffnessad         247 non-null    float64
 62  Psychoticsymptomsad     247 non-null    float64
 63  Confusionad             247 non-null    float64
 64  Coughad                 247 non-null    float64
 65  Haemoptasisad           247 non-null    float64
 66  SOBadd                  247 non-null    float64
 67  Chestpainad             247 non-null    float64
 68  Bleedingad              244 non-null    float64
 69  WPqPCRDiagnosis         300 non-null    float64
 70  Isolate                 347 non-null    int64
dtypes: float64(66), int64(5)
memory usage: 192.6 KB
```

[48]:
```python
for column in df1.columns:
    unique_values = df1[column].unique()
    print(f"Column: {column}")
    print(f"Unique values: {unique_values}")
    print(f"Number of unique values: {len(unique_values)}")
    print("\n")
```

```
Column: Year
Unique values: [2017 2018 2019 2016]
Number of unique values: 4


Column: Month
Unique values: [ 6  7  8  9 10  3  5 11 12  2  4  1]
Number of unique values: 12
```

```
Column: Hospital
Unique values: [1 2 3 4 5 6 7 8]
Number of unique values: 8


Column: Sample
Unique values: [1 2]
Number of unique values: 2


Column: ICU
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: OPD
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Sex
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Age
Unique values: [49. 47. 51. 37. nan 70. 29. 42. 30. 34. 65. 38. 17. 15. 18. 57.
43. 36.
 52. 69. 67. 62. 50. 48. 28. 45. 41. 24. 33. 56. 40. 44. 76. 55. 68.  9.
 61. 32. 59. 58. 25. 22. 54. 60. 39. 64. 35. 46. 20. 19. 72.  2. 53.  7.
 26. 31. 21. 63. 66. 13. 71. 16. 79. 27. 77.]
Number of unique values: 65


Column: Ethnicity
Unique values: [ 1. nan  3.  2.]
Number of unique values: 4


Column: Education
Unique values: [ 5. 10. 11. nan 12.  8.  9.  6.  0.  4.  2. 13.  1.  7.  3.]
Number of unique values: 15


Column: TertiaryEducation
Unique values: [ 3. nan  1.  2.]
Number of unique values: 4
```

```
Column: Prophylactics
Unique values: [ 2.  1. nan  3.]
Number of unique values: 4


Column: Pasttreatments
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Pastantibiotics
Unique values: [nan  3.  1.  2.]
Number of unique values: 4


Column: Chronicillness
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Possibleexposure
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Feveronset
Unique values: [ 1. nan  2.]
Number of unique values: 3


Column: Headacheonset
Unique values: [ 1. nan  2.]
Number of unique values: 3


Column: Musclepainonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Cnsuffusiononset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Jaundiceonset
Unique values: [ 1.  2. nan]
```

```
Number of unique values: 3


Column: Skinrashonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Oliguriaonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Anuriaonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: SOBonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Coughonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Haemoptasisonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Chestpainonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Nauseaonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Vomitingonset
Unique values: [ 2.  1. nan]
Number of unique values: 3
```

```
Column: Diarrhoeaonset
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Bleedingonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Mucosalrashonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Prostrationonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Rigorsonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Photophobiaonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Chillsonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Muscletendernessonset
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Psychoticsymptomsonset
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Confusiononset
Unique values: [ 2. nan  1.]
Number of unique values: 3
```

```
Column: Feverad
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Headachead
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Chillsad
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Rigorsad
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Musclepainad
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Muscletendernessad
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Nauseaad
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Vomitingadmission
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Cnsuffusionad
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: Skinrashad
```

```
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Mucosalrashad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Prostrationad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Diarrhoeaad
Unique values: [ 2.  1. nan]
Number of unique values: 3


Column: OliguriaAd
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Anuriaad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Jaundicead
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Hepatictendernessad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Hepatomegalyad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Spleenimegalyad
Unique values: [ 2. nan  1.]
Number of unique values: 3
```

```
Column: Lympadenopathyad
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Photophobiaad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Neckstiffnessad
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Psychoticsymptomsad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Confusionad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Coughad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Haemoptasisad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: SOBadd
Unique values: [ 1.  2. nan]
Number of unique values: 3


Column: Chestpainad
Unique values: [ 2. nan  1.]
Number of unique values: 3


Column: Bleedingad
Unique values: [ 2. nan  1.]
```

```
Number of unique values: 3


Column: WPqPCRDiagnosis
Unique values: [ 3.  2. nan  1.]
Number of unique values: 4


Column: Isolate
Unique values: [ 2  1 98]
Number of unique values: 3
```

[49]:
```
numerical_columns1 = ['Age']
categorical_columns1 = [col for col in df1.columns if col not in␣
 ↪numerical_columns1]
categorical_columns1
```

[49]:
```
['Year',
 'Month',
 'Hospital',
 'Sample',
 'ICU',
 'OPD',
 'Sex',
 'Ethnicity',
 'Education',
 'TertiaryEducation',
 'Prophylactics',
 'Pasttreatments',
 'Pastantibiotics',
 'Chronicillness',
 'Possibleexposure',
 'Feveronset',
 'Headacheonset',
 'Musclepainonset',
 'Cnsuffusiononset',
 'Jaundiceonset',
 'Skinrashonset',
 'Oliguriaonset',
 'Anuriaonset',
 'SOBonset',
 'Coughonset',
 'Haemoptasisonset',
 'Chestpainonset',
 'Nauseaonset',
```

```
'Vomitingonset',
'Diarrhoeaonset',
'Bleedingonset',
'Mucosalrashonset',
'Prostrationonset',
'Rigorsonset',
'Photophobiaonset',
'Chillsonset',
'Muscletendernessonset',
'Psychoticsymptomsonset',
'Confusiononset',
'Feverad',
'Headachead',
'Chillsad',
'Rigorsad',
'Musclepainad',
'Muscletendernessad',
'Nauseaad',
'Vomitingadmission',
'Cnsuffusionad',
'Skinrashad',
'Mucosalrashad',
'Prostrationad',
'Diarrhoeaad',
'OliguriaAd',
'Anuriaad',
'Jaundicead',
'Hepatictendernessad',
'Hepatomegalyad',
'Spleenimegalyad',
'Lympadenopathyad',
'Photophobiaad',
'Neckstiffnessad',
'Psychoticsymptomsad',
'Confusionad',
'Coughad',
'Haemoptasisad',
'SOBadd',
'Chestpainad',
'Bleedingad',
'WPqPCRDiagnosis',
'Isolate']
```

[50]:
```
for col in categorical_columns1:
    df1[col] = df1[col].astype('category')
```

[51]:
```
for col in categorical_columns1:
    df1[col].fillna(df1[col].mode()[0], inplace=True)
```

[52]:
```
for col in numerical_columns1:
    df1[col].fillna(df1[col].mean(), inplace=True)
```

[53]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 347 entries, 0 to 346
Data columns (total 71 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Year                    347 non-null    category
 1   Month                   347 non-null    category
 2   Hospital                347 non-null    category
 3   Sample                  347 non-null    category
 4   ICU                     347 non-null    category
 5   OPD                     347 non-null    category
 6   Sex                     347 non-null    category
 7   Age                     347 non-null    float64
 8   Ethnicity               347 non-null    category
 9   Education               347 non-null    category
 10  TertiaryEducation       347 non-null    category
 11  Prophylactics           347 non-null    category
 12  Pasttreatments          347 non-null    category
 13  Pastantibiotics         347 non-null    category
 14  Chronicillness          347 non-null    category
 15  Possibleexposure        347 non-null    category
 16  Feveronset              347 non-null    category
 17  Headacheonset           347 non-null    category
 18  Musclepainonset         347 non-null    category
 19  Cnsuffusiononset        347 non-null    category
 20  Jaundiceonset           347 non-null    category
 21  Skinrashonset           347 non-null    category
 22  Oliguriaonset           347 non-null    category
 23  Anuriaonset             347 non-null    category
 24  SOBonset                347 non-null    category
 25  Coughonset              347 non-null    category
 26  Haemoptasisonset        347 non-null    category
 27  Chestpainonset          347 non-null    category
 28  Nauseaonset             347 non-null    category
 29  Vomitingonset           347 non-null    category
 30  Diarrhoeaonset          347 non-null    category
 31  Bleedingonset           347 non-null    category
 32  Mucosalrashonset        347 non-null    category
 33  Prostrationonset        347 non-null    category
```

```
 34  Rigorsonset              347 non-null    category
 35  Photophobiaonset         347 non-null    category
 36  Chillsonset              347 non-null    category
 37  Muscletendernessonset    347 non-null    category
 38  Psychoticsymptomsonset   347 non-null    category
 39  Confusiononset           347 non-null    category
 40  Feverad                  347 non-null    category
 41  Headachead               347 non-null    category
 42  Chillsad                 347 non-null    category
 43  Rigorsad                 347 non-null    category
 44  Musclepainad             347 non-null    category
 45  Muscletendernessad       347 non-null    category
 46  Nauseaad                 347 non-null    category
 47  Vomitingadmission        347 non-null    category
 48  Cnsuffusionad            347 non-null    category
 49  Skinrashad               347 non-null    category
 50  Mucosalrashad            347 non-null    category
 51  Prostrationad            347 non-null    category
 52  Diarrhoeaad              347 non-null    category
 53  OliguriaAd               347 non-null    category
 54  Anuriaad                 347 non-null    category
 55  Jaundicead               347 non-null    category
 56  Hepatictendernessad      347 non-null    category
 57  Hepatomegalyad           347 non-null    category
 58  Spleenimegalyad          347 non-null    category
 59  Lympadenopathyad         347 non-null    category
 60  Photophobiaad            347 non-null    category
 61  Neckstiffnessad          347 non-null    category
 62  Psychoticsymptomsad      347 non-null    category
 63  Confusionad              347 non-null    category
 64  Coughad                  347 non-null    category
 65  Haemoptasisad            347 non-null    category
 66  SOBadd                   347 non-null    category
 67  Chestpainad              347 non-null    category
 68  Bleedingad               347 non-null    category
 69  WPqPCRDiagnosis          347 non-null    category
 70  Isolate                  347 non-null    category
dtypes: category(70), float64(1)
memory usage: 36.2 KB
```

3.4 Predictions the Model

[54]:
```
X_test=df1
```

[55]:
```
predictions = Logistic_Model.predict(X_test)
print(predictions)
```

```
[2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 1 2 2 1 1 1 2 2 2 2 1 2
2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 1 2 2
2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1
1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2 1 1 1 1 2 1 1 2 1
2 2 2 2 1 1 1 1 2 2 2 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
2 1 2 1 1 2 1 2 1 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2 1 2 1 1 1 2 1 1 1 1
2 1 2 2 1 1 1 2 2 2 2 2 1 2 2 1 2 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1
2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 2 1 2 2 2 2 2
2 2 2 1 2 2 2 2 2 2 2 2 2 2 1]
```

[56]:
```
predictions_df = pd.DataFrame({
    'ID': df_test['ID'],  # Use the IDs corresponding to non-duplicates
    'Final': predictions
})
```

[57]:
```
predictions_df.to_csv("D:\\4th Year 1st Sem\\4rth year  - 1st sem\\ST 4035 -␣
↪Data Science\\Assignment 1\\st40352023\\sample_submission.csv",index=False)
```

[ ]:
```

```