APPROACH USING MANUAL CODING (PYTHON)

INDEX

CODE

```
import pandas as pd
import numpy as np
import os
import pandas as pd
import matplotlib.pyplot as plt
from IPython.display import display, HTML
TRANS_DATA = 'Transactional Data'
REF_DATA = 'Reference Data'


# Question 1
## PRIMARY DIAGNOSIS
accounts = pd.read_csv(TRANS_DATA+'/accounts.csv')
acc=pd.read_csv(TRANS_DATA+'/accounts.csv')
charges = pd.read_csv(TRANS_DATA+'/charges.csv')
diag = pd.read_csv(TRANS_DATA+'/diagnosis.csv')
accounts.head()
accounts.describe()
accounts.isna().sum()
# accounts['Discharge Year'] = pd.to_datetime(accounts['Discharge Year'], format='%Y-%m-%d').dt.year
diag['Diagnosis Sequence'] = diag['Diagnosis Sequence'].astype(str)
diag_grouped = diag.groupby('Account ID').agg({'Diagnosis Sequence':','.join,'Diagnosis Code':','.join}).reset_index()
diag_grouped
accounts_diag_grouped_join = pd.merge(accounts, diag_grouped, on="Account ID")
accounts_diag_grouped_join
# def secondary_diseases_by_year(row):
#     seq = row['Diagnosis Sequence']
#     seq_list = seq.split(',')
#     if '1' in seq_list and '2' in seq_list:
# #         print(seq_list)
#        return row

# accounts_diag_grouped_join =
accounts_diag_grouped_join[accounts_diag_grouped_join['Discharge
Year']==2019].apply(secondary_diseases_by_year, axis =1)
def create_seq_code_col(row):
#     print(row)
    seq = row['Diagnosis Sequence'].split(',')
    code = row['Diagnosis Code'].split(',')
    zip_seq_code = dict(zip(seq,code))
    selected_items = {int(key): value for key, value in zip_seq_code.items()}
#     print(selected_items)
    return dict(sorted(selected_items.items()))
# accounts_diag_grouped_join['seq_code'] =
{accounts_diag_grouped_join['seq_code']:accounts_diag_grouped_join['seq_code']}
accounts_diag_grouped_join = accounts_diag_grouped_join.dropna()
accounts_diag_grouped_join['code_seq_map']=accounts_diag_grouped_join.apply(create_seq_
code_col, axis=1)
accounts_diag_grouped_join
def prim_secondary_codes(code_seq_map):
    prim = None
    if 1 in code_seq_map.keys():
        prim = code_seq_map[1]
    sec = ','.join([value for key, value in code_seq_map.items() if key > 1])

#     return {'prim':prim, 'sec':sec}
    return prim, sec
accounts_diag_grouped_join['code_seq_map']
accounts_diag_grouped_join[['primary', 'secondary']] =
accounts_diag_grouped_join['code_seq_map'].apply(lambda code_seq_map:
pd.Series(prim_secondary_codes(code_seq_map)))
accounts_diag_grouped_join_2019 =
accounts_diag_grouped_join[accounts_diag_grouped_join['Discharge Year']==2019]
accounts_diag_grouped_join_2019
(accounts_diag_grouped_join_2019['primary'].value_counts()>5).sum()
import matplotlib.pyplot as plt

df_2019 = accounts_diag_grouped_join_2019[accounts_diag_grouped_join_2019['Discharge
Year'] == 2019]
primary_counts = df_2019['primary'].value_counts().head(10)
fig, ax = plt.subplots(figsize=(10, 6))
primary_counts.plot(kind='barh', color='darkblue', ax=ax)

ax.set_title('Top 10 Primary Diagnoses for 2019')
ax.set_xlabel('Number of Accounts')
ax.set_ylabel('Diagnosis Code')

ax.invert_yaxis()
for i in ax.patches:
    ax.text(i.get_width() + 0.1, i.get_y() + 0.1,
            str(round((i.get_width()), 2)),
            fontsize=10, color='dimgrey')


plt.show()

accounts_diag_grouped_join_2019['primary'].mode().values[0]
# Most common Primary Diagnosis
### Most common Primary Diagnosis - Z38.00 with 74 number of accounts
.
### SECONDARY DIAGNOSIS
def expand_list(row):
    row_list = row.split(',')
    return row_list
accounts_diag_grouped_join_2019['secondary']=accounts_diag_grouped_join_2019['secondary'].apply(expand_list)
secondary_list = []
for item in accounts_diag_grouped_join_2019['secondary'].values:
    secondary_list = secondary_list+item
secondary_list
second_count = pd.Series(secondary_list)
second_count.mode()
import matplotlib.pyplot as plt


secondary_counts = df_exploded['secondary'].value_counts().head(10)
fig, ax = plt.subplots(figsize=(10, 6))
secondary_counts.plot(kind='barh', color='green', ax=ax)
ax.set_title('Top 10 Secondary Diagnoses for 2019')
ax.set_xlabel('Number of Accounts')
ax.set_ylabel('Diagnosis Code')
ax.invert_yaxis()
for i in ax.patches:
    ax.text(i.get_width() + 0.1, i.get_y() + 0.1,
            str(i.get_width()),
            fontsize=10, color='dimgrey')

plt.tight_layout()
plt.show()

### Most common Secondary Diagnosis I10 with a count of 775 accounts
.
.
# Question 2
lookup_drg= pd.read_csv(REF_DATA+'/lookup_table_MS DRG.csv')
account_charges = pd.merge(accounts, charges, on="Account ID")
accdrg=pd.merge(lookup_drg,account_charges, on="MS DRG Code")
accdrg
total_charges_avg = accdrg.groupby(['MS DRG','MS DRG Code']).agg({
     'Account ID': 'nunique',
    'Total Charges': 'sum',
}).reset_index()
total_charges_avg.sort_values(by="Total Charges",ascending=False)
total_charges_avg['Charge_avg']= total_charges_avg['Total
Charges']/total_charges_avg['Account ID']
total_charges_avg.sort_values(by="Charge_avg",ascending=False)
# highest_avg_charges = sorted_drgs.groupby('MS DRG')['Total Charges'].idxmax()
# highest_avg_table = result.loc[highest_avg_charges]
# highest_avg_table.rename(columns = {"Total Charges":"Highest Average Charge"},
inplace=True)
# print(sorted_drgs)

import matplotlib.pyplot as plt

N = 10
top_drgs = total_charges_avg.sort_values('Charge_avg', ascending=False).head(N)

plt.figure(figsize=(12, 8))
ax = top_drgs.sort_values('Charge_avg', ascending=True).plot(kind='barh', x='MS DRG Code',
y='Charge_avg', legend=False, color='darkgreen')
plt.title('Top 10 MS DRGs by Highest Average Charge')
plt.xlabel('Average Charge')
plt.ylabel('MS DRG')
for i in ax.patches:
    plt.text(i.get_width() + 0.1, i.get_y() + 0.1, str(round(i.get_width(), 2)), fontsize=10,
color='black')
plt.tight_layout()
plt.show()


top_drgs
```

```
sorted_table = total_charges_avg.sort_values('Charge_avg', ascending=False)
top_diagnosis = sorted_table.iloc[0]['MS DRG']
top_average_charge = sorted_table.iloc[0]['Charge_avg']

print(f"The diagnosis related group with the highest average charge is: {top_diagnosis} with an
average charge of {top_average_charge:.2f}")

### The diagnosis related group with the highest average charge is: 004 - TRACHEOSTOMY WITH
MV >96 HOURS OR PRINCIPAL DIAGNOSIS EXCEPT FACE, MOUTH AND NECK WITHOUT MAJOR
O.R. PROCEDURES with an average charge of 630043.40 USD
.
.
# Question 3
lookup_provider = pd.read_csv(REF_DATA+'/lookup_table_provider.csv')
lookup_prov_account = pd.merge(accounts, lookup_provider, left_on="Attending Provider ID",
right_on="Provider ID")
lookup_prov_account.drop(columns=['Attending Provider ID'],inplace=True)
lookup_prov_account
lookup_prov_account['Admit Date'] = pd.to_datetime(lookup_prov_account['Admit Date'])
dec_2018_dates = lookup_prov_account[(lookup_prov_account['Admit Date'].dt.year == 2018)
& (lookup_prov_account['Admit Date'].dt.month == 12)]
dec_2018_dates
result=dec_2018_dates.groupby(['Provider ID'])['Account
ID'].count().sort_values(ascending=False).reset_index()
result



provider_max = result['Provider ID'].iloc[0]
max_acc = result['Account ID'].iloc[0]

print(f"Attending Provider ID which had the highest volume of accounts in the month of
December in 2018: {provider_max}")
print(f"Volume of Accounts: {max_acc}")

### Attending Provider ID which had the highest volume of accounts in the month of December
in 2018: PHY_003777
### Volume of Accounts: 10
import seaborn as sns
import matplotlib.pyplot as plt

N = 10
df = result.head(N)
```

```
# Plotting
plt.figure(figsize=(12, 8))
df['Provider'] = df['Provider ID']
df.sort_values(by='Account ID', ascending=True).plot(kind='barh', x='Provider', y='Account ID', legend=False, color='green')
plt.title('Top 10 Providers by Account ID Count')
plt.xlabel('Count of Account ID')
plt.ylabel('Provider ID with Service')
plt.tight_layout()
plt.show()

result_service=dec_2018_dates.groupby(['Provider Service'])['Account ID'].count().sort_values(ascending=False).reset_index()
result_service
provider_service_max = result_service['Provider Service'].iloc[0]
max_serv_acc = result_service['Account ID'].iloc[0]

print(f"Attending Provider Service which had the highest volume of accounts in the month of December in 2018: {provider_service_max}")
print(f"Volume of Accounts: {max_serv_acc}")
### Attending Provider Service which had the highest volume of accounts in the month of December in 2018: Internal Medicine
### Volume of Accounts: 77
import seaborn as sns
import matplotlib.pyplot as plt

N = 10
df = result_service.head(N)

# Plotting
plt.figure(figsize=(12, 8))
df['Provider'] =  df['Provider Service']
df.sort_values(by='Account ID', ascending=True).plot(kind='barh', x='Provider', y='Account ID', legend=False, color='green')
plt.title('Top 10 Providers by Account ID Count')
plt.xlabel('Count of Account ID')
plt.ylabel('Provider ID with Service')
plt.tight_layout()
plt.show()

.
.
# Question 4
acc['Admit Date'] = pd.to_datetime(acc['Admit Date'])
```

```
acc['Discharge Date'] = pd.to_datetime(acc['Discharge Date'])
acc['length_of_stay'] = acc['LOS']
acc['Discharge Year']
acc['Discharge Year'] = acc['Discharge Year'].astype(int)
acc = acc[acc['length_of_stay'] >= 0]
acc
lookup_facility = pd.read_csv(REF_DATA+'/lookup_table_facility.csv')
facilitymerge=pd.merge(lookup_facility,acc,on="Facility Code")
facilitymerge.groupby(['Facility Code','Facility Name','Discharge
Year'])['length_of_stay'].mean().reset_index().sort_values(['Discharge Year'],ascending=False)
import seaborn as sns
import matplotlib.pyplot as plt

df = acc.groupby(['Facility Code','Discharge
Year'])['length_of_stay'].mean().reset_index().sort_values(['Facility Code', 'Discharge
Year'],ascending=True)


plt.figure(figsize=(14, 7))
sns.lineplot(x='Facility Code', y='length_of_stay', hue='Discharge Year', marker="o", data=df,
palette="Set2")
plt.title('Average Length of Stay by Facility and Discharge Year')
plt.ylabel('Average Length of Stay')
plt.legend(title='Discharge Year')
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.show()


### .
### .
# Question 5
lookup_msdrg = pd.read_csv(REF_DATA+'/lookup_table_MS DRG.csv')
lookup_msdrg_service_line = pd.read_csv(REF_DATA+'/lookup_table_MS DRG Service Line.csv')
msdrg_service_merge = pd.merge(lookup_msdrg,lookup_msdrg_service_line, on="MS DRG
Code")
msdrg_service_accounts_merge = pd.merge(accounts, msdrg_service_merge, on = 'MS DRG
Code')
msdrg_service_accounts_merge.sort_values(by="Arithmetic Mean LOS",ascending=False)


stay_len = msdrg_service_accounts_merge.groupby(['Service Line Description']).agg({
    'LOS': 'mean',
    'Account ID': 'nunique',
    'Arithmetic Mean LOS': 'mean'
}).reset_index()

stay_len
stay_len.sort_values('Arithmetic Mean LOS', ascending=False)
stay_len['difference']=stay_len['LOS']-stay_len['Arithmetic Mean LOS']
# length_merge
stay_len
stay_len.sort_values('difference', ascending=False)
max_diff_index = stay_len['difference'].idxmax()

service = stay_len['Service Line Description'].iloc[max_diff_index]
max_diff_value = stay_len['difference'].iloc[max_diff_index]

print(f"The Service Line Description with the highest difference in length of stay is: {service}")
print(f"The maximum difference in length of stay value is: {max_diff_value:.2f}")
```

### The Service Line Description with the highest difference in length of stay is: SURGICAL TRACHEOSTOM

### The maximum difference in length of stay value is: 19.32

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


top_10 = stay_len.sort_values(by='difference', ascending=False).head(10)
sns.set_theme(style="whitegrid")
plt.figure(figsize=(12, 8))
sns.barplot(x='difference', y=top_10['Service Line Description'].astype(str), data=top_10, palette="viridis")
plt.xlabel('Difference in Length of Stay')
plt.ylabel('MS DRG Code')
plt.title('Top 10 Service with Highest diff_length_of_stay')
plt.show()
```

.

.

## EXTRA ANALYSIS

```
extra=accounts_diag_grouped_join
extra = pd.merge(extra, lookup_drg, on="MS DRG Code")
extra
```

## Which ZipCode has the highest number of patients recorded?

```
import pandas as pd

zip_counts = extra.groupby('Patient ZipCode')['Account ID'].nunique().sort_values(ascending=False)
```

```
highest_zip = zip_counts.idxmax()
highest_count = zip_counts.max()

print(f"ZipCode with highest number of patients: {highest_zip} with {highest_count} patients")



### ZipCode with highest number of patients: 45202 with 217 patients
import matplotlib.pyplot as plt
import seaborn as sns

N = 10
fig, ax = plt.subplots(figsize=(12, 6))
zip_counts.head(N).plot(kind='bar', color='#006d77', ax=ax)
plt.title(f'Top {N} ZipCodes with Highest Number of Patients')
plt.ylabel('Number of Patients')
plt.xlabel('ZipCode')
plt.xticks(rotation=45)

for p in ax.patches:
   ax.annotate(str(p.get_height()), (p.get_x() * 1.005, p.get_height() * 1.005))

plt.show()

## Which facility service had the highest revenue ?
lookup_facility = pd.read_csv(REF_DATA+'/lookup_table_facility.csv')
facility_service_merge = pd.merge(lookup_facility,accdrg, on="Facility Code")
facility_service_merge
revenue_per_facility = facility_service_merge.groupby(['Facility Code','Facility Name'])['Total Charges'].sum()
sorted_revenue = revenue_per_facility.sort_values(ascending=False)
sorted_revenue.reset_index()

# import seaborn as sns
# import matplotlib.pyplot as plt
# sorted_data = sorted_revenue.reset_index()
# heatmap_data = sorted_data.set_index('Facility Name')['Total Charges']
# heatmap_data = heatmap_data.to_frame()
# plt.figure(figsize=(12,8))
# sns.heatmap(heatmap_data, annot=True, fmt='.2f', cmap='viridis_r', cbar_kws={'label': 'Total Charges'})
# plt.title('Revenue per Facility')
# plt.show()
```

```
sns.set_theme(style="whitegrid")
plt.figure(figsize=(8, 6))
sns.barplot(x=sorted_data['Total Charges'], y=sorted_data['Facility Name'].astype(str), data=top_10, palette="viridis")
plt.xlabel('Total Charges')
plt.ylabel('Facility Name')
plt.title('Revenue per Facility')
plt.show()
highest_revenue_facility = revenue_per_facility.idxmax()
highest_revenue = revenue_per_facility.max()

print(f"The facility {highest_revenue_facility} had the highest revenue of ${highest_revenue}.")
```

### TRINITY HOSPITAL (F_0120) had the highest revenue of $39228315.52

## Who provided the most treatments ?

```
facility_merge = pd.merge(lookup_facility,accdrg, on="Facility Code")
lookup_charges = pd.read_csv(REF_DATA+'/lookup_table_charge.csv')
charges_merge=pd.merge(lookup_charges,facility_merge, on="Charge Code")
fac_prov_merge = pd.merge(charges_merge,lookup_provider, left_on="Attending Provider ID", right_on="Provider ID")
fac_prov_merge.drop(columns=['Attending Provider ID'],inplace=True)
fac_prov_merge
provider_treatments = fac_prov_merge.groupby(['Provider name','Facility Name'])['Charge Description'].nunique().sort_values(ascending=False)
top_provider, top_facility = provider_treatments.idxmax()
top_count = provider_treatments.max()
print(f"The provider {top_provider} at facility {top_facility} provided the most treatments with a total of {top_count} unique charge descriptions.")
provider_treatments.reset_index()
```

### The provider Coffey, S at facility LONGWOOD provided the most treatments with a total of 685 unique charge descriptions.

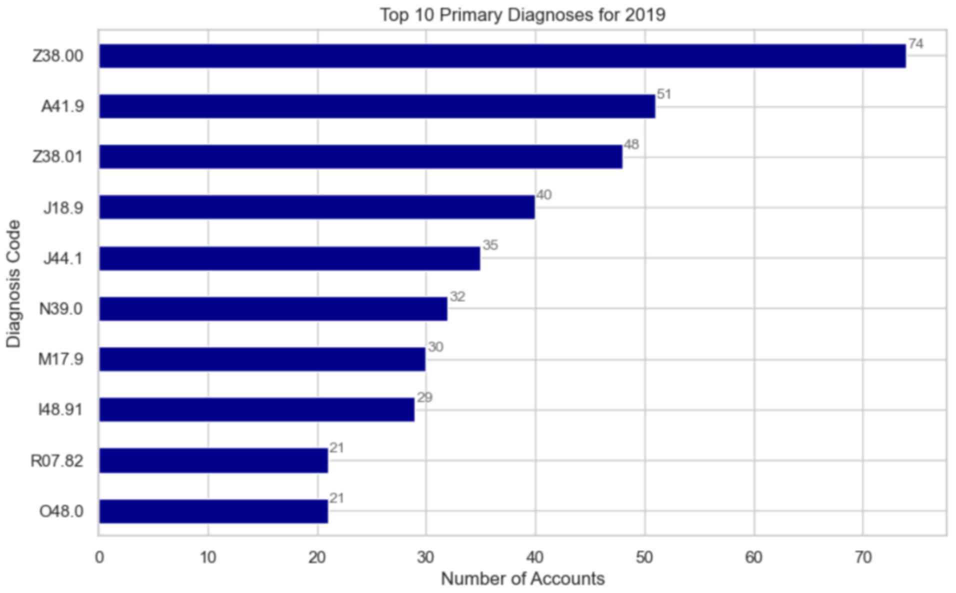
```
import matplotlib.pyplot as plt

N = 10
top_provider_treatments = provider_treatments.head(N)
plt.figure(figsize=(12, 8))
top_provider_treatments.sort_values().plot(kind='barh', color='#006d77')
plt.title('Top Providers and Facilities by Number of Unique Treatments')
plt.xlabel('Number of Unique Treatments')
plt.ylabel('Provider and Facility')
plt.tight_layout()
plt.show()
```

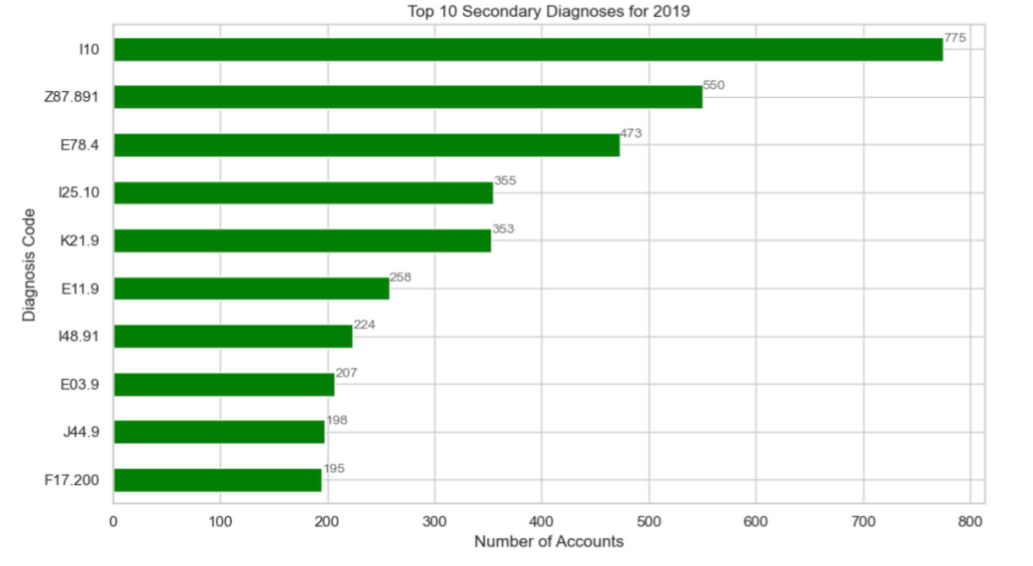
```
## Which were the most common charge descriptions/prescriptions that were given to patients?
most_common = charges_merge.groupby('Charge Description')['Account ID'].count().sort_values(ascending=False)

most_common.reset_index()
import matplotlib.pyplot as plt

top_10_charge_descriptions = most_common.sort_values(ascending=False).head(10)

# Plot
plt.figure(figsize=(12, 8))
top_10_charge_descriptions.plot(kind='barh', color='#006d77')
plt.title('Top 10 Most Common Charge Description for patients')
plt.xlabel('Number of Occurrences')
plt.ylabel('Charge Description')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()

### HC RX – SELF ADMINISTERABLE DRUGS was the most common charge description
```

OBSERVATIONS

ANALYSIS 1

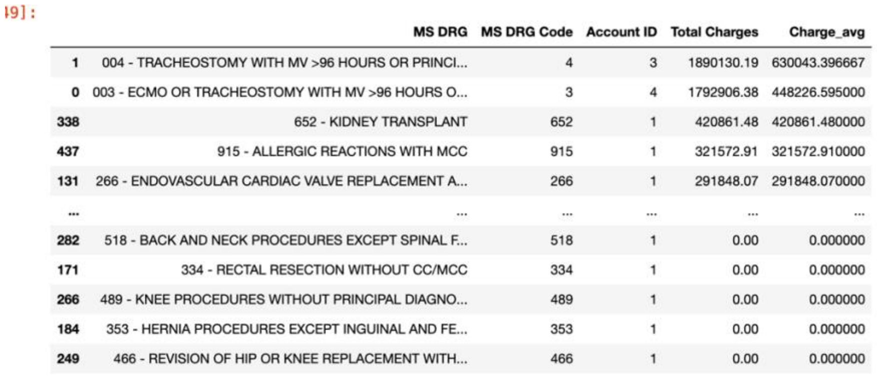
Most common Primary Diagnosis - **Z38.00 with 74 number of accounts**



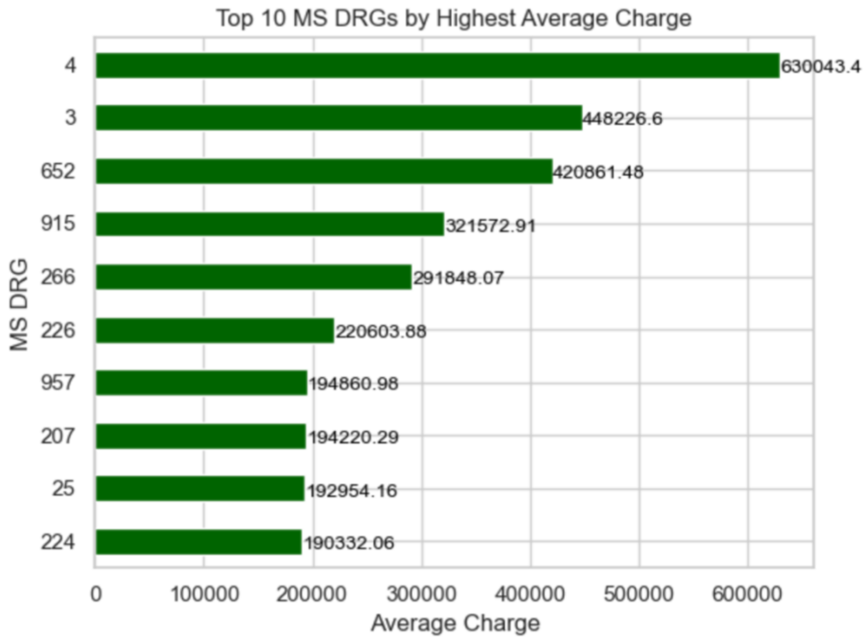
MOST COMMON SECONDARY DIAGNOSIS **IS I10 with 775 number of accounts**	

ANALYSIS 2

The diagnosis related group with the highest average charge is: **004 - TRACHEOSTOMY WITH MV >96 HOURS OR PRINCIPAL DIAGNOSIS EXCEPT FACE, MOUTH AND NECK WITHOUT MAJOR O.R. PROCEDURES with an average charge of 630043.40 USD**

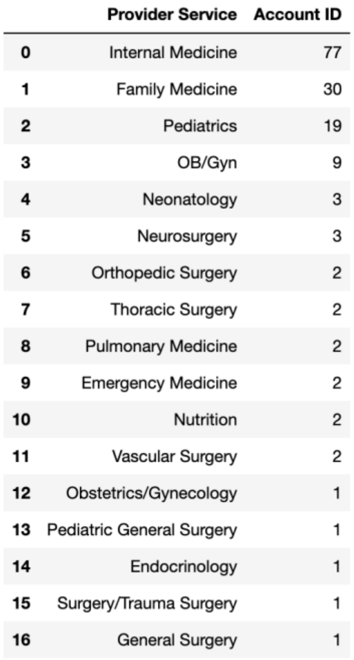


```
49]:
                                                  MS DRG  MS DRG Code  Account ID  Total Charges     Charge_avg
  1    004 - TRACHEOSTOMY WITH MV >96 HOURS OR PRINCI...            4           3     1890130.19  630043.396667
  0   003 - ECMO OR TRACHEOSTOMY WITH MV >96 HOURS O...            3           4     1792906.38  448226.595000
338                              652 - KIDNEY TRANSPLANT          652           1      420861.48  420861.480000
437                     915 - ALLERGIC REACTIONS WITH MCC          915           1      321572.91  321572.910000
131   266 - ENDOVASCULAR CARDIAC VALVE REPLACEMENT A...          266           1      291848.07  291848.070000
...                                                  ...          ...         ...            ...            ...
282     518 - BACK AND NECK PROCEDURES EXCEPT SPINAL F...          518           1           0.00       0.000000
171               334 - RECTAL RESECTION WITHOUT CC/MCC          334           1           0.00       0.000000
266    489 - KNEE PROCEDURES WITHOUT PRINCIPAL DIAGNO...          489           1           0.00       0.000000
184     353 - HERNIA PROCEDURES EXCEPT INGUINAL AND FE...          353           1           0.00       0.000000
249      466 - REVISION OF HIP OR KNEE REPLACEMENT WITH...          466           1           0.00       0.000000
```

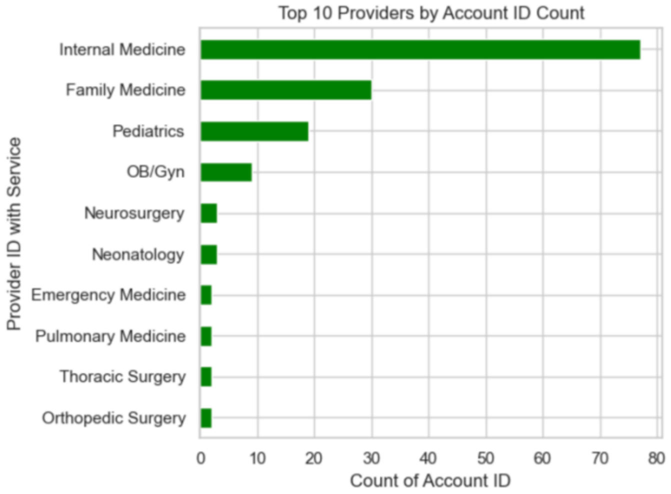


ANALYSIS 3

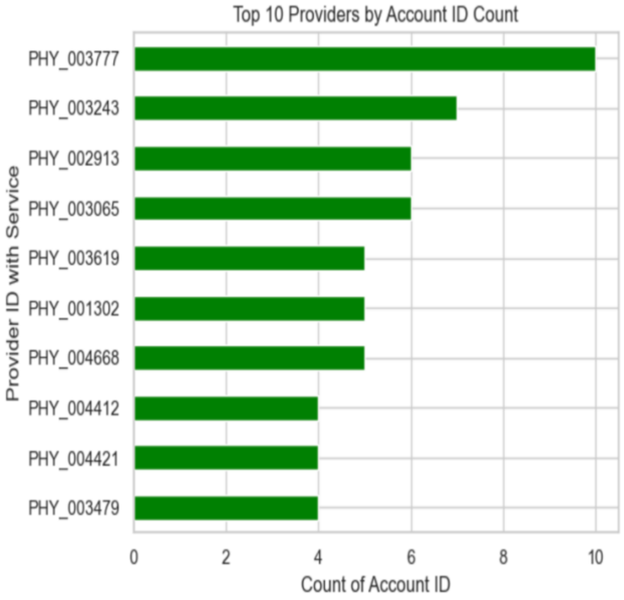
Attending Provider Service which had the highest volume of accounts in the month of December in 2018**: Internal Medicine with a Volume of Accounts as 77**



	Provider Service	Account ID
0	Internal Medicine	77
1	Family Medicine	30
2	Pediatrics	19
3	OB/Gyn	9
4	Neonatology	3
5	Neurosurgery	3
6	Orthopedic Surgery	2
7	Thoracic Surgery	2
8	Pulmonary Medicine	2
9	Emergency Medicine	2
10	Nutrition	2
11	Vascular Surgery	2
12	Obstetrics/Gynecology	1
13	Pediatric General Surgery	1
14	Endocrinology	1
15	Surgery/Trauma Surgery	1
16	General Surgery	1

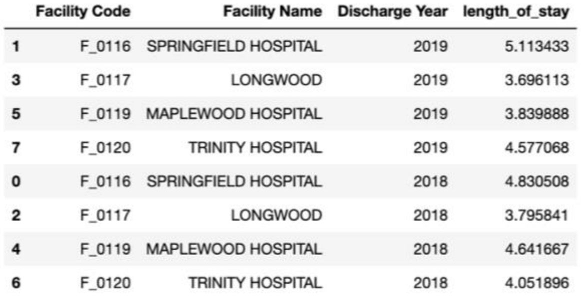


Attending Provider ID which had the highest volume of accounts in the month of **December in 2018: PHY_003777**
Volume of Accounts: 10

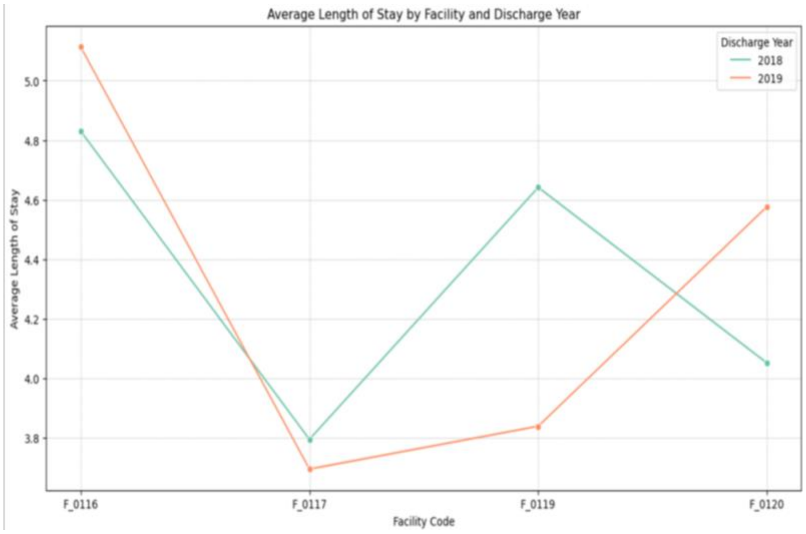


ANALYSIS 4

The average length of stay for each facility in the year 2018 and 2019 is as follows:

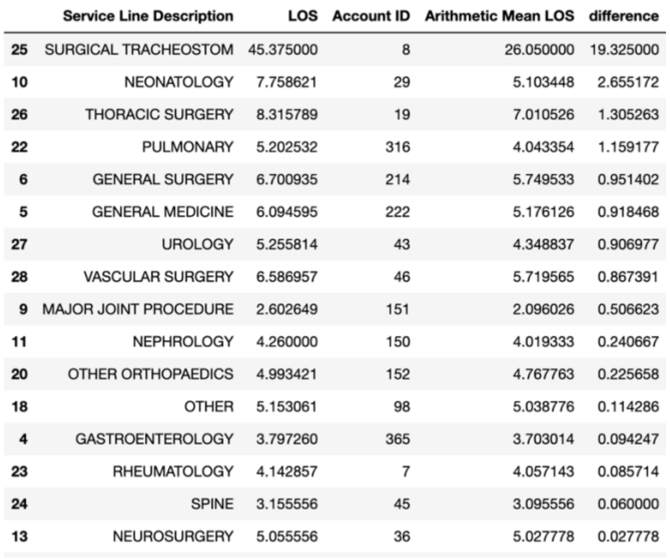


	Facility Code	Facility Name	Discharge Year	length_of_stay
1	F_0116	SPRINGFIELD HOSPITAL	2019	5.113433
3	F_0117	LONGWOOD	2019	3.696113
5	F_0119	MAPLEWOOD HOSPITAL	2019	3.839888
7	F_0120	TRINITY HOSPITAL	2019	4.577068
0	F_0116	SPRINGFIELD HOSPITAL	2018	4.830508
2	F_0117	LONGWOOD	2018	3.795841
4	F_0119	MAPLEWOOD HOSPITAL	2018	4.641667
6	F_0120	TRINITY HOSPITAL	2018	4.051896

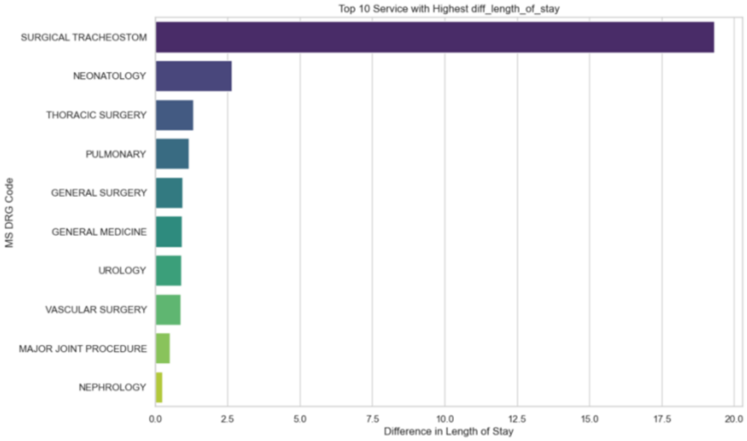


ANALYSIS 5

The **Service Line Description** with the highest difference in length of stay is: **SURGICAL TRACHEOSTOM**
The maximum **difference in length of stay value is: 19.32**



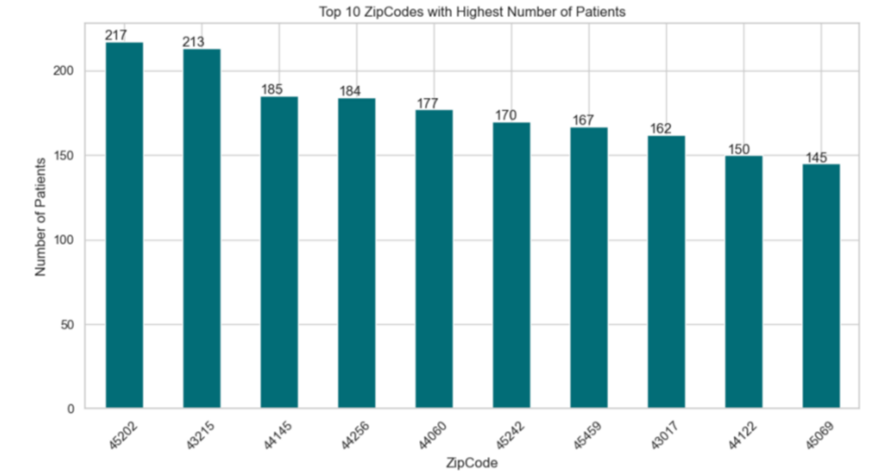
	Service Line Description	LOS	Account ID	Arithmetic Mean LOS	difference
25	SURGICAL TRACHEOSTOM	45.375000	8	26.050000	19.325000
10	NEONATOLOGY	7.758621	29	5.103448	2.655172
26	THORACIC SURGERY	8.315789	19	7.010526	1.305263
22	PULMONARY	5.202532	316	4.043354	1.159177
6	GENERAL SURGERY	6.700935	214	5.749533	0.951402
5	GENERAL MEDICINE	6.094595	222	5.176126	0.918468
27	UROLOGY	5.255814	43	4.348837	0.906977
28	VASCULAR SURGERY	6.586957	46	5.719565	0.867391
9	MAJOR JOINT PROCEDURE	2.602649	151	2.096026	0.506623
11	NEPHROLOGY	4.260000	150	4.019333	0.240667
20	OTHER ORTHOPAEDICS	4.993421	152	4.767763	0.225658
18	OTHER	5.153061	98	5.038776	0.114286
4	GASTROENTEROLOGY	3.797260	365	3.703014	0.094247
23	RHEUMATOLOGY	4.142857	7	4.057143	0.085714
24	SPINE	3.155556	45	3.095556	0.060000
13	NEUROSURGERY	5.055556	36	5.027778	0.027778



EXTRA ANALYSIS

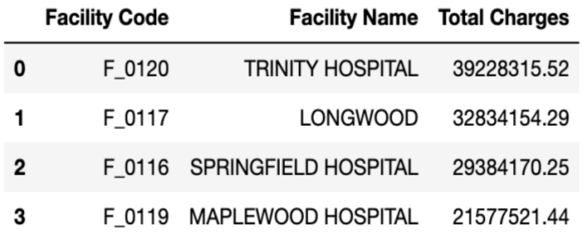
EXTRA ANALYSIS 1 - Which ZipCode has the highest number of patients recorded?

ZipCode with highest number of patients: 45202 with 217 patients

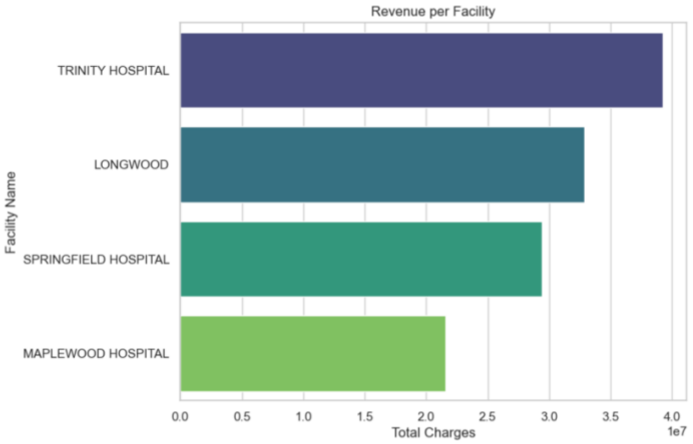


EXTRA ANALYSIS 2 - Which facility service had the highest revenue ?

The facility - **TRINITY HOSPITAL (F_0120)** had the highest revenue of **$ 39228315.52.**

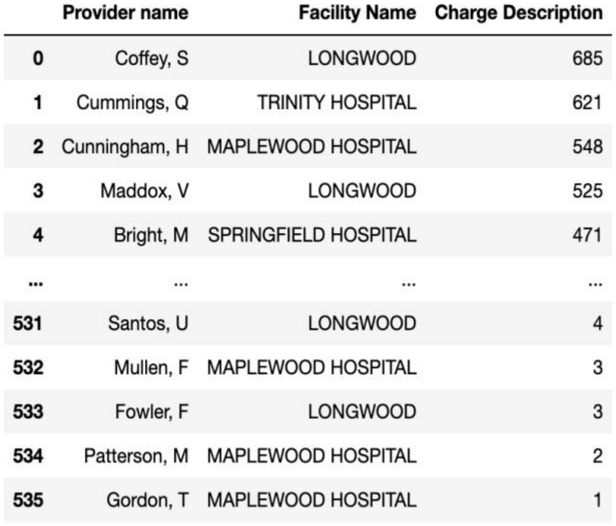


	Facility Code	Facility Name	Total Charges
0	F_0120	TRINITY HOSPITAL	39228315.52
1	F_0117	LONGWOOD	32834154.29
2	F_0116	SPRINGFIELD HOSPITAL	29384170.25
3	F_0119	MAPLEWOOD HOSPITAL	21577521.44

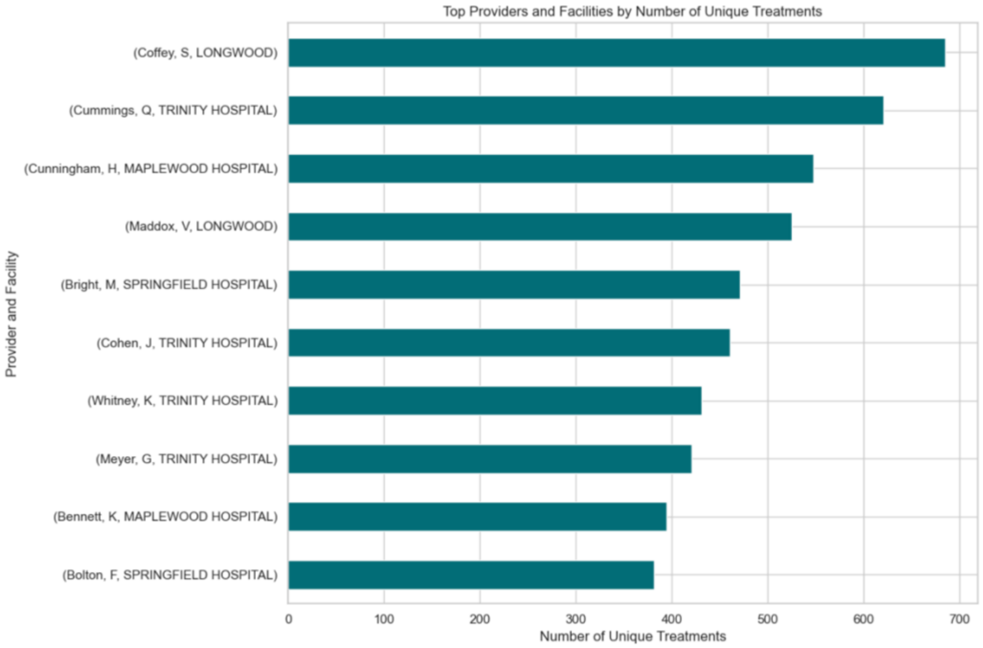


EXTRA ANALYSIS 3 - Who provided the most treatments ?

The provider **Coffey, S at facility LONGWOOD** provided the most treatments with **a total of 685 unique charge descriptions.**

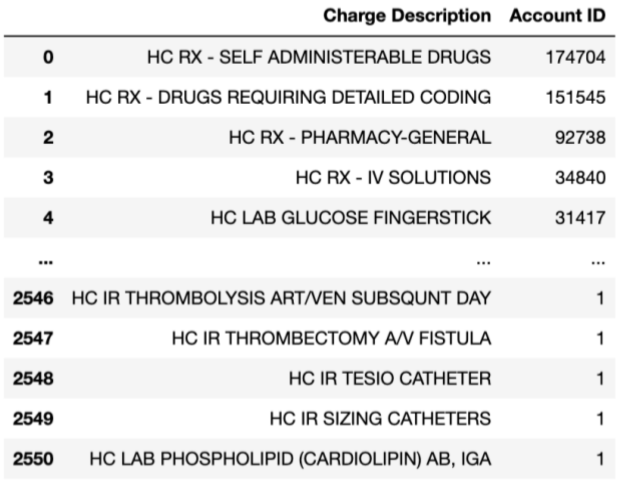


	Provider name	Facility Name	Charge Description
0	Coffey, S	LONGWOOD	685
1	Cummings, Q	TRINITY HOSPITAL	621
2	Cunningham, H	MAPLEWOOD HOSPITAL	548
3	Maddox, V	LONGWOOD	525
4	Bright, M	SPRINGFIELD HOSPITAL	471
...	...	...	...
531	Santos, U	LONGWOOD	4
532	Mullen, F	MAPLEWOOD HOSPITAL	3
533	Fowler, F	LONGWOOD	3
534	Patterson, M	MAPLEWOOD HOSPITAL	2
535	Gordon, T	MAPLEWOOD HOSPITAL	1

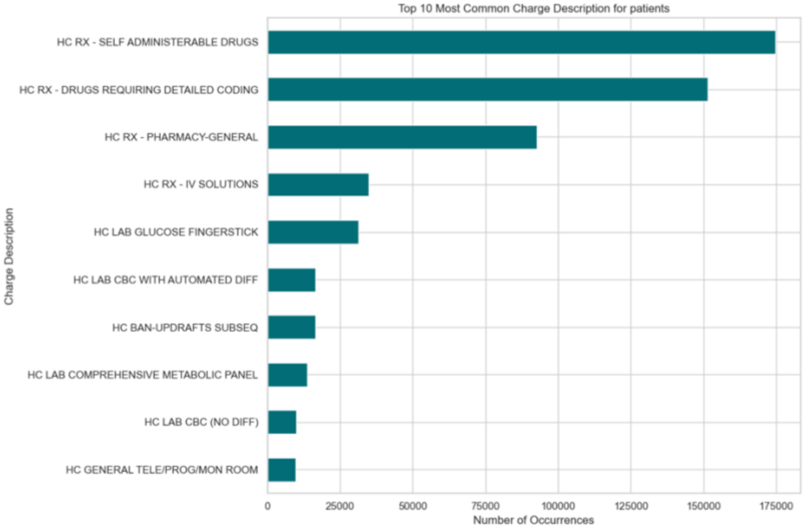


EXTRA ANALYSIS 4 - Which were the most common charge descriptions/prescriptions that were given to patients?

HC RX – SELF ADMINISTERABLE DRUGS was the most common charge description having a count of **174704 accounts**



	Charge Description	Account ID
0	HC RX - SELF ADMINISTERABLE DRUGS	174704
1	HC RX - DRUGS REQUIRING DETAILED CODING	151545
2	HC RX - PHARMACY-GENERAL	92738
3	HC RX - IV SOLUTIONS	34840
4	HC LAB GLUCOSE FINGERSTICK	31417
...	...	...
2546	HC IR THROMBOLYSIS ART/VEN SUBSQUNT DAY	1
2547	HC IR THROMBECTOMY A/V FISTULA	1
2548	HC IR TESIO CATHETER	1
2549	HC IR SIZING CATHETERS	1
2550	HC LAB PHOSPHOLIPID (CARDIOLIPIN) AB, IGA	1


Top 10 Most Common Charge Description for patients
Charge Description
HC RX - SELF ADMINISTERABLE DRUGS
HC RX - DRUGS REQUIRING DETAILED CODING
HC RX - PHARMACY-GENERAL
HC RX - IV SOLUTIONS
HC LAB GLUCOSE FINGERSTICK
HC LAB CBC WITH AUTOMATED DIFF
HC BAN-UPDRAFTS SUBSEQ
HC LAB COMPREHENSIVE METABOLIC PANEL
HC LAB CBC (NO DIFF)
HC GENERAL TELE/PROG/MON ROOM
Number of Occurrences