Please note: This Jupyter notebook is supplementary to the primary write up. See the document "Start Here."

Part A - Describe a real-world organizational situation or issue in the Data Dictionary

1. Question: Which categorical variables are related to patient hospital readmission?
2. Benefit from analysis: An analysis of categorical variables and their relationship to hospital readmissions allows the organization to identify factors that lead to readmission. If a relationship is identified, the organization would be able to put policies and procedures in place during the initial stay to reduce the risk for readmission. Simply put, an analysis of this data could lead to the organization reducing readmissions and penalties that may occur because of readmissions.
3. Data relevant to the question: Gender, ReAdmis, Soft_drink, Initial_admin, HighBlood, Stroke, Complication_risk, Overweight, Arthritis, Diabetes, Hyperlipidemia, BackPain, Anxiety, Allergic_rhinitis, Reflux_esophagitis, Asthma, Services

See accompanying document for more detail on section A3.

Load Python Libraries

In [1]:
```
import pandas as pd
from scipy.stats import chi2
from scipy.stats import chi2_contingency
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

Read CSV & Load Data in to Pandas Dataframe

In [2]:
```
data = pd.read_csv(r'C:\Users\JoshFunderburk\Desktop\D207 Project\medical_clean_raw_data.csv', index_col=0)
```

In [3]:
```
data.head()
```

Out[3]:

	Customer_id	Interaction	UID	City	State	
CaseOrder						
1	C412403	8cd49b13-f45a-4b47-a2bd-173ffa932c2f	3a83ddb66e2ae73798bdf1d705dc0932	Eva	AL	
2	Z919181	d2450b70-0337-4406-bdbb-bc1037f1734c	176354c5eef714957d486009feabf195	Marianna	FL	
3	F995323	a2057123-abf5-4a2c-abad-8ffe33512562	e19a0fa00aeda885b8a436757e889bc9	Sioux Falls	SD	Min
4	A879973	1dec528d-eb34-4079-adce-0d7a40e82205	cd17d7b6d152cb6f23957346d11c3f07	New Richland	MN	
5	C544523	5885f56b-d6da-43a3-8760-83583af94266	d2f0425877b10ed6bb381f3e2579424a	West Point	VA	

5 rows × 49 columns

Data Cleaning

In [4]:
```
#Rename selected columns to better match the data definition
data.rename(columns={'Item1': 'Survey_Timely_Admission',
                     'Item2': 'Survey_Timely_Treatment',
                     'Item3': 'Survey_Timely_Visit',
                     'Item4': 'Survey_Reliability',
                     'Item5': 'Survey_Options',
                     'Item6': 'Survey_Hours_of_Treatment',
                     'Item7': 'Survey_Courteous_Staff',
                     'Item8': 'Survey_Doctor_Active_Listening',
                    },
            inplace=True)

#Confirm column names have been updated
print(data.columns[-8:])
```

```
Index(['Survey_Timely_Admission', 'Survey_Timely_Treatment',
       'Survey_Timely_Visit', 'Survey_Reliability', 'Survey_Options',
       'Survey_Hours_of_Treatment', 'Survey_Courteous_Staff',
       'Survey_Doctor_Active_Listening'],
      dtype='object')
```

Part B1 - Data Set Analysis (Chi Square Test)

Reference for Chi-Square Python code: Sewell, William (n.d.). Retrieved May 22, 2021, from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=52d9e72f-3309-4780-ac2b-accf014a436f (https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=52d9e72f-3309-4780-ac2b-accf014a436f)

Reference for Chi-Square Python code: Naik, Krish. (2020). Tutorial 33- Chi Square Test Implementation with Python- Hypothesis Testing- Part 2 [Video]. Retrieved 20 May 2021, from https://www.youtube.com/watch?v=w5iKu1IrTJQ (https://www.youtube.com/watch?v=w5iKu1IrTJQ).

B1.1 Use Chi Square to test for a relationship between readmissions and gender

In [5]:

```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Gender'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('Interpret Test Statistic:')
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('Interpret P-Value:')
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Gender   Female  Male  Nonbinary
ReAdmis
No          3205  2995        131
Yes         1813  1773         83

Degrees of Freedom = 2

Expected Values =
 [[3176.8958 3018.6208  135.4834]
 [1841.1042 1749.3792   78.5166]]
Interpret Test Statistic:

Probability =  0.95
Critical Value = 5.991
Test Statistic = 1.586

Outcome: Independent (fail to reject H0)
Interpret P-Value:

Significance = 0.050
P-Value = 0.453

Outcome: Independent (fail to reject H0)
```

B1.2 Use Chi Square to test for a relationship between readmissions and soft drinks

In [6]:

```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Soft_drink'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Soft_drink    No   Yes
ReAdmis
No          4717  1614
Yes         2708   961

Degrees of Freedom = 1

Expected Values =
 [[4700.7675 1630.2325]
 [2724.2325  944.7675]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.557

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.455

Outcome: Independent (fail to reject H0)
```

B1.3 Use Chi Square to test for a relationship between readmissions and initial admission reason

In [7]:
```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Initial_admin'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Initial_admin  Elective Admission  Emergency Admission  Observation Admissio
n
ReAdmis
No                            1608                 3156                  1567
Yes                            896                 1904                   869

Degrees of Freedom = 2

Expected Values =
 [[1585.2824 3203.486  1542.2316]
 [ 918.7176 1856.514   893.7684]]

Probability =  0.95
Critical Value = 5.991
Test Statistic = 3.890

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.143

Outcome: Independent (fail to reject H0)
```

B1.4 Use Chi Square to test for a relationship between readmissions and high blood pressure

In [8]:

```python
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['HighBlood'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 HighBlood    No   Yes
ReAdmis
No         3747  2584
Yes        2163  1506

Degrees of Freedom = 1

Expected Values =
 [[3741.621 2589.379]
 [2168.379 1500.621]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.042

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.837

Outcome: Independent (fail to reject H0)
```

B1.5 Use Chi Square to test for a relationship between readmissions and stroke

In [9]:
```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Stroke'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Stroke     No   Yes
ReAdmis
No       5071  1260
Yes      2936   733

Degrees of Freedom = 1

Expected Values =
 [[5069.2317 1261.7683]
 [2937.7683  731.2317]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.004

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.947

Outcome: Independent (fail to reject H0)
```

B1.6 Use Chi Square to test for a relationship between readmissions and complication risk level

In [10]:

```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Complication_risk'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Complication_risk  High   Low  Medium
ReAdmis
No                 2135  1343    2853
Yes                1223   782    1664

Degrees of Freedom = 2

Expected Values =
 [[2125.9498 1345.3375 2859.7127]
 [1232.0502  779.6625 1657.2873]]

Probability =  0.95
Critical Value = 5.991
Test Statistic = 0.159

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.924

Outcome: Independent (fail to reject H0)
```

B1.7 Use Chi Square to test for a relationship between readmissions and overweight

In [11]:
```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Overweight'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Overweight    No   Yes
ReAdmis
No          1821  4510
Yes         1085  2584

Degrees of Freedom = 1

Expected Values =
 [[1839.7886 4491.2114]
 [1066.2114 2602.7886]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.698

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.403

Outcome: Independent (fail to reject H0)
```

B1.8 Use Chi Square to test for a relationship between readmissions and arthritis

In [12]:

```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Arthritis'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Arthritis    No   Yes
ReAdmis
No         4086  2245
Yes        2340  1329

Degrees of Freedom = 1

Expected Values =
 [[4068.3006 2262.6994]
 [2357.6994 1311.3006]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.555

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.456

Outcome: Independent (fail to reject H0)
```

B1.9 Use Chi Square to test for a relationship between readmissions and diabetes

In [13]:
```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Diabetes'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Diabetes    No   Yes
ReAdmis
No        4591  1740
Yes       2671   998

Degrees of Freedom = 1

Expected Values =
 [[4597.5722 1733.4278]
 [2664.4278 1004.5722]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.080

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.778

Outcome: Independent (fail to reject H0)
```

B1.10 Use Chi Square to test for a relationship between readmissions and hyperlipidemia

In [14]:
```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Hyperlipidemia'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Hyperlipidemia    No   Yes
ReAdmis
No              4206  2125
Yes             2422  1247

Degrees of Freedom = 1

Expected Values =
 [[4196.1868 2134.8132]
 [2431.8132 1237.1868]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.167

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.683

Outcome: Independent (fail to reject H0)
```

B1.11 Use Chi Square to test for a relationship between readmissions and back pain

In [15]:

```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['BackPain'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 BackPain    No   Yes
ReAdmis
No        3758  2573
Yes       2128  1541

Degrees of Freedom = 1

Expected Values =
 [[3726.4266 2604.5734]
 [2159.5734 1509.4266]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 1.717

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.190

Outcome: Independent (fail to reject H0)
```

B1.12 Use Chi Square to test for a relationship between readmissions and anxiety

In [16]:

```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Anxiety'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Anxiety    No   Yes
ReAdmis
No       4301  2030
Yes      2484  1185

Degrees of Freedom = 1

Expected Values =
 [[4295.5835 2035.4165]
 [2489.4165 1179.5835]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.048

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.827

Outcome: Independent (fail to reject H0)
```

B1.13 Use Chi Square to test for a relationship between readmissions and allergic rhinitis

In [17]:
```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Allergic_rhinitis'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Allergic_rhinitis    No   Yes
ReAdmis
No                  3825  2506
Yes                 2234  1435

Degrees of Freedom = 1

Expected Values =
 [[3835.9529 2495.0471]
 [2223.0471 1445.9529]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.197

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.657

Outcome: Independent (fail to reject H0)
```

B1.14 Use Chi Square to test for a relationship between readmissions and reflux esophagitis

In [18]:

```python
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Reflux_esophagitis'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Reflux_esophagitis    No   Yes
ReAdmis
No                   3726  2605
Yes                  2139  1530

Degrees of Freedom = 1

Expected Values =
 [[3713.1315 2617.8685]
 [2151.8685 1517.1315]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 0.272

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.602

Outcome: Independent (fail to reject H0)
```

B1.15 Use Chi Square to test for a relationship between readmissions and asthma

In [19]:

```python
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Asthma'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Asthma     No   Yes
ReAdmis
No       4462  1869
Yes      2645  1024

Degrees of Freedom = 1

Expected Values =
 [[4499.4417 1831.5583]
 [2607.5583 1061.4417]]

Probability =  0.95
Critical Value = 3.841
Test Statistic = 2.857

Outcome: Independent (fail to reject H0)

Significance = 0.050
P-Value = 0.091

Outcome: Independent (fail to reject H0)
```

B1.16 Use Chi Square to test for a relationship between readmissions and the services received

In [20]:

```
#Contingency table
contingency_table = pd.crosstab(data['ReAdmis'], data['Services'])
print('Contingency Table = \n', contingency_table)

#Store contingency table values
observed_values = contingency_table.values

#Identify the test statistic, p-value, degrees of freedom, and expected values
stat, p, dof, expected = chi2_contingency(observed_values)

print('\nDegrees of Freedom =', dof)
print ('\nExpected Values =\n', expected)

#Interpret test statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('\nProbability = ', prob)
print('Critical Value = %.3f' % critical)
print('Test Statistic = %.3f' % stat)

if abs(stat) >= critical:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')

#Interpret p-value
alpha = 1.0 - prob
print('\nSignificance = %.3f' % alpha)
print('P-Value = %.3f' % p)

if p <= alpha:
    print('\nOutcome: Dependent (reject H0)')
else:
    print('\nOutcome: Independent (fail to reject H0)')
```

```
Contingency Table =
 Services  Blood Work  CT Scan  Intravenous  MRI
ReAdmis
No              3335      737         2027  232
Yes             1930      488         1103  148

Degrees of Freedom = 3

Expected Values =
 [[3333.2715  775.5475 1981.603   240.578 ]
 [1931.7285  449.4525 1148.397   139.422 ]]

Probability =  0.95
Critical Value = 7.815
Test Statistic = 8.893

Outcome: Dependent (reject H0)

Significance = 0.050
P-Value = 0.031

Outcome: Dependent (reject H0)
```

Part B3 - Justification of analysis technique

I chose to use Chi Square in my analysis because it allowed me to build one block of code and repeat it to test multiple categorical values against readmissions. Focusing in on categorical values with Chi Square gave me insights into the probability of relationship between readmissions and sixteen other variables. Although ANOVA or a T-Test give great insight into non-categorical values, those statistical methods must be catered to each field and would not have allowed me to do such a broad analysis across the data set. However, Performing ANOVA and T-Tests for the remaining non-categorical fields would be wise to ensure all relationships to hospital readmissions are uncovered. For the purpose of this analysis and assessment requirements, Chi Square was the best place to start to establish a foundation for relationships. Additionally, Chi Square was the best technique to answer the question posed in Section A1 due to the technique's purpose of analyzing categorical values.

Part C - Univariate Statistics

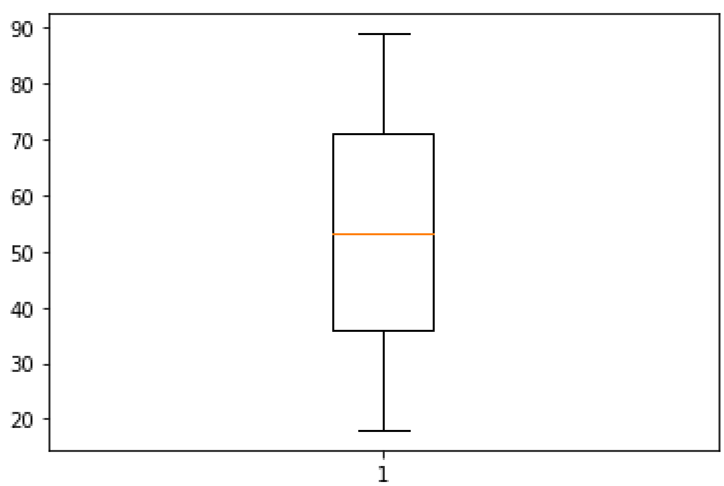
Age Statistics (Continuous Variable)

In [21]:
```
data['Age'].describe()
```

Out[21]:
```
count    10000.000000
mean        53.511700
std         20.638538
min         18.000000
25%         36.000000
50%         53.000000
75%         71.000000
max         89.000000
Name: Age, dtype: float64
```

In [22]:
```
plt.boxplot(data.Age)
```

Out[22]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x13b63925dc8>,
  <matplotlib.lines.Line2D at 0x13b639358c8>],
 'caps': [<matplotlib.lines.Line2D at 0x13b63935dc8>,
  <matplotlib.lines.Line2D at 0x13b63935f48>],
 'boxes': [<matplotlib.lines.Line2D at 0x13b63925c48>],
 'medians': [<matplotlib.lines.Line2D at 0x13b6393c908>],
 'fliers': [<matplotlib.lines.Line2D at 0x13b639251c8>],
 'means': []}
```



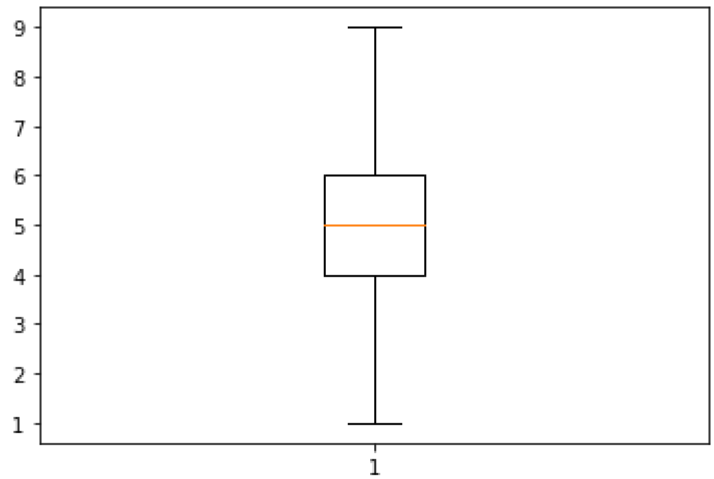
Doctor Visit Statistics (Continuous Variable)

In [23]:
```
data['Doc_visits'].describe()
```

Out[23]:
```
count    10000.000000
mean         5.012200
std          1.045734
min          1.000000
25%          4.000000
50%          5.000000
75%          6.000000
max          9.000000
Name: Doc_visits, dtype: float64
```

In [24]:
```
plt.boxplot(data.Doc_visits)
```

Out[24]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x13b639e3888>,
  <matplotlib.lines.Line2D at 0x13b639e3d08>],
 'caps': [<matplotlib.lines.Line2D at 0x13b639e3e48>,
  <matplotlib.lines.Line2D at 0x13b639e86c8>],
 'boxes': [<matplotlib.lines.Line2D at 0x13b639e32c8>],
 'medians': [<matplotlib.lines.Line2D at 0x13b639e8bc8>],
 'fliers': [<matplotlib.lines.Line2D at 0x13b639e8d08>],
 'means': []}
```



Services Statistics (Categorical Variable)

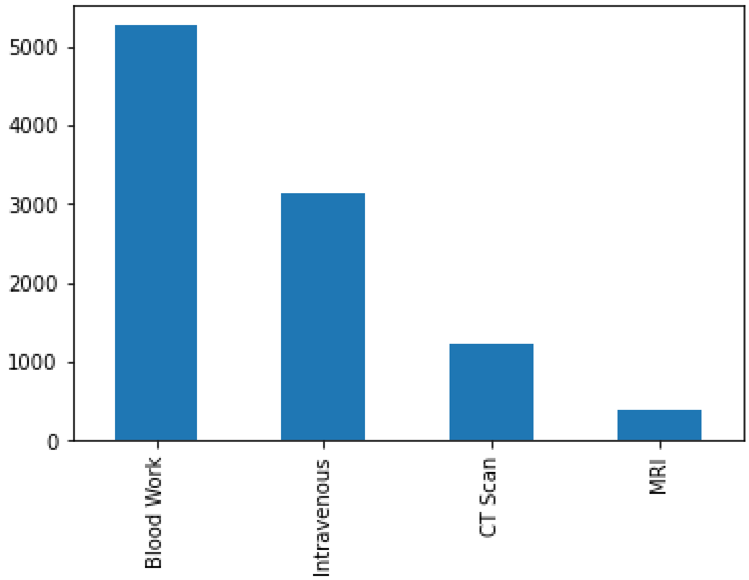
Source: Piush, Vaish. (2021, May 15). Visualise Categorical Variables in Python. Retrieved from https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/ (https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/)

In [25]:
```
data['Services'].describe()
```

Out[25]:
```
count          10000
unique             4
top       Blood Work
freq            5265
Name: Services, dtype: object
```

```
In [26]: data['Services'].value_counts().plot.bar()
```

```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x13b63a69ec8>
```



Initial Admission Reason Statistics (Categorical Variable)

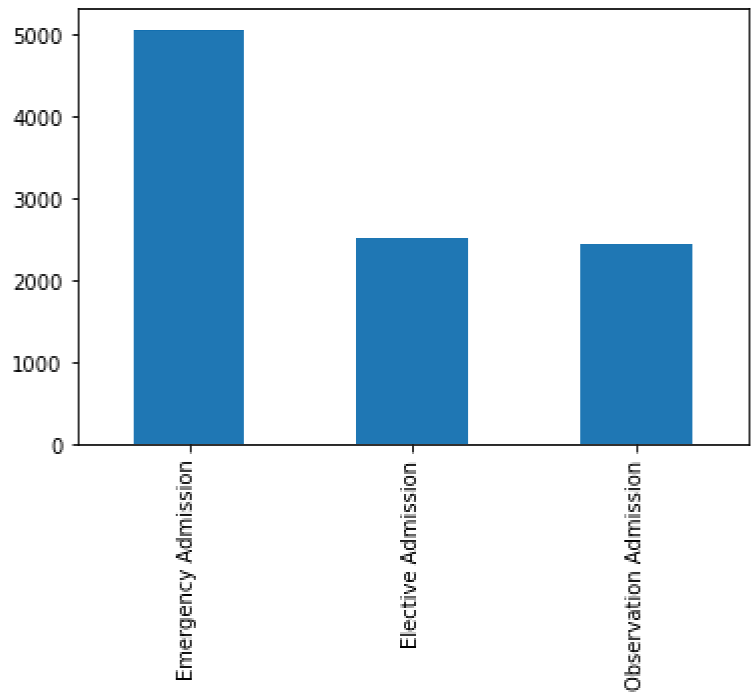
Source: Piush, Vaish. (2021, May 15). Visualise Categorical Variables in Python. Retrieved from https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/ (https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/)

```
In [27]: data['Initial_admin'].describe()
```

```
Out[27]: count                   10000
         unique                      3
         top       Emergency Admission
         freq                     5060
         Name: Initial_admin, dtype: object
```

In [28]:
```
data['Initial_admin'].value_counts().plot.bar()
```

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x13b63afde08>



Part D - Bivariate Statistics

Age vs ReAdmissions (Continuous Variable)

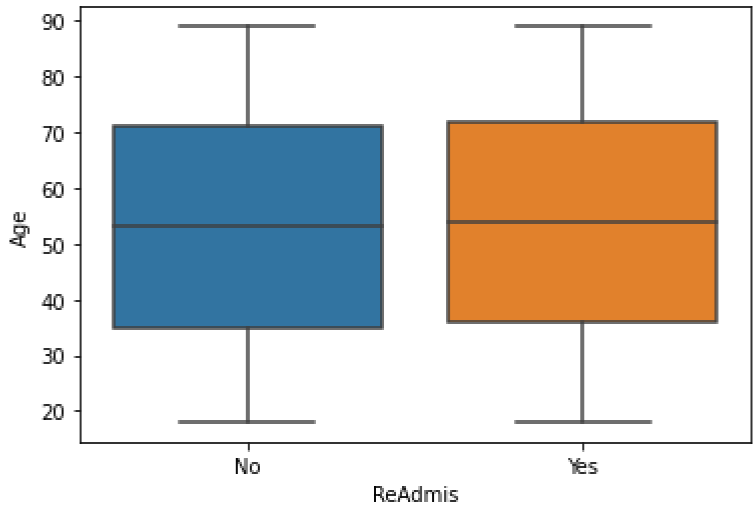
In [29]:
```
ReAdmis_Age_Grouped = pd.crosstab(index=data['ReAdmis'], columns=data['Age'])
ReAdmis_Age_Grouped
```

Out[29]:

Age	18	19	20	21	22	23	24	25	26	27	...	80	81	82	83	84	85	86	87	88	89
ReAdmis																					
No	85	92	86	81	88	98	84	87	95	79	...	71	85	74	86	89	72	91	91	83	86
Yes	48	45	34	44	53	39	60	43	49	56	...	45	46	50	48	38	63	65	45	60	46

2 rows × 72 columns

In [30]:
```
sns.boxplot(x='ReAdmis', y='Age', data=data)
plt.show()
```



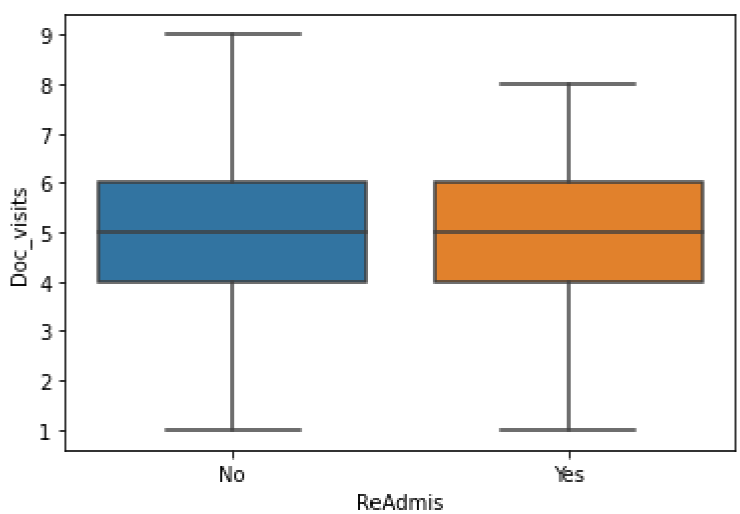
of Doctor Visits vs ReAdmissions (Continuous Variable)

In [31]:
```
ReAdmis_DoctorVisits_Grouped = pd.crosstab(index=data['ReAdmis'], columns=data
['Doc_visits'])
ReAdmis_DoctorVisits_Grouped
```

Out[31]:

Doc_visits	1	2	3	4	5	6	7	8	9
ReAdmis									
No	4	36	375	1511	2416	1558	392	37	2
Yes	2	22	220	874	1407	878	242	24	0

In [32]:
```
sns.boxplot(x='ReAdmis', y='Doc_visits', data=data)
plt.show()
```



Initial Days vs ReAdmissions (Continuous Variable)

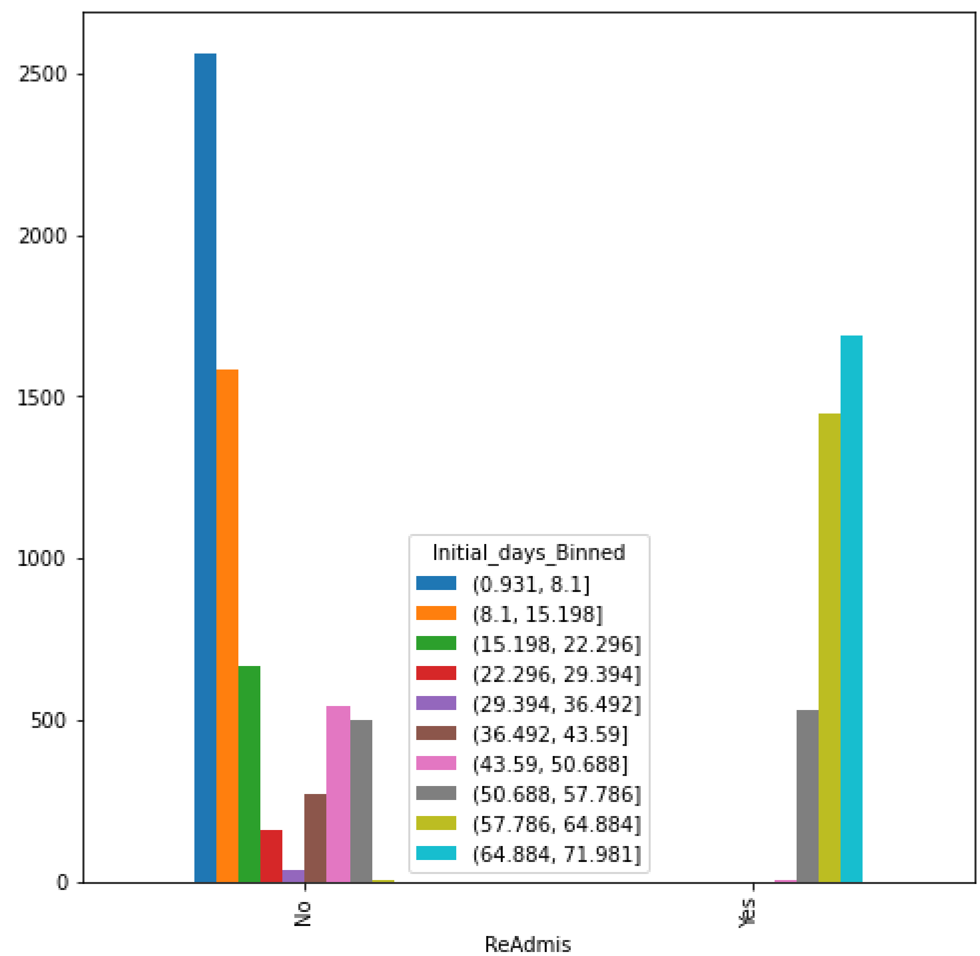
In [33]:
```
data['Initial_days_Binned'] = pd.cut(data['Initial_days'], 10)

ReAdmis_DoctorVisits_Grouped = pd.crosstab(index=data['ReAdmis'], columns=data
['Initial_days_Binned'])
ReAdmis_DoctorVisits_Grouped
```

Out[33]:

Initial_days_Binned	(0.931, 8.1]	(8.1, 15.198]	(15.198, 22.296]	(22.296, 29.394]	(29.394, 36.492]	(36.492, 43.59]	(43.59, 50.688]	(50.688, 57.786]	(57.78 64.88
ReAdmis									
No	2563	1586	669	157	34	271	544	502	
Yes	0	0	0	0	0	0	2	531	14

In [34]:
```
ReAdmis_DoctorVisits_Grouped.plot(kind="bar", figsize=(8,8))
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x13b640aef48>



Services vs ReAdmissions (Categorical Variable)

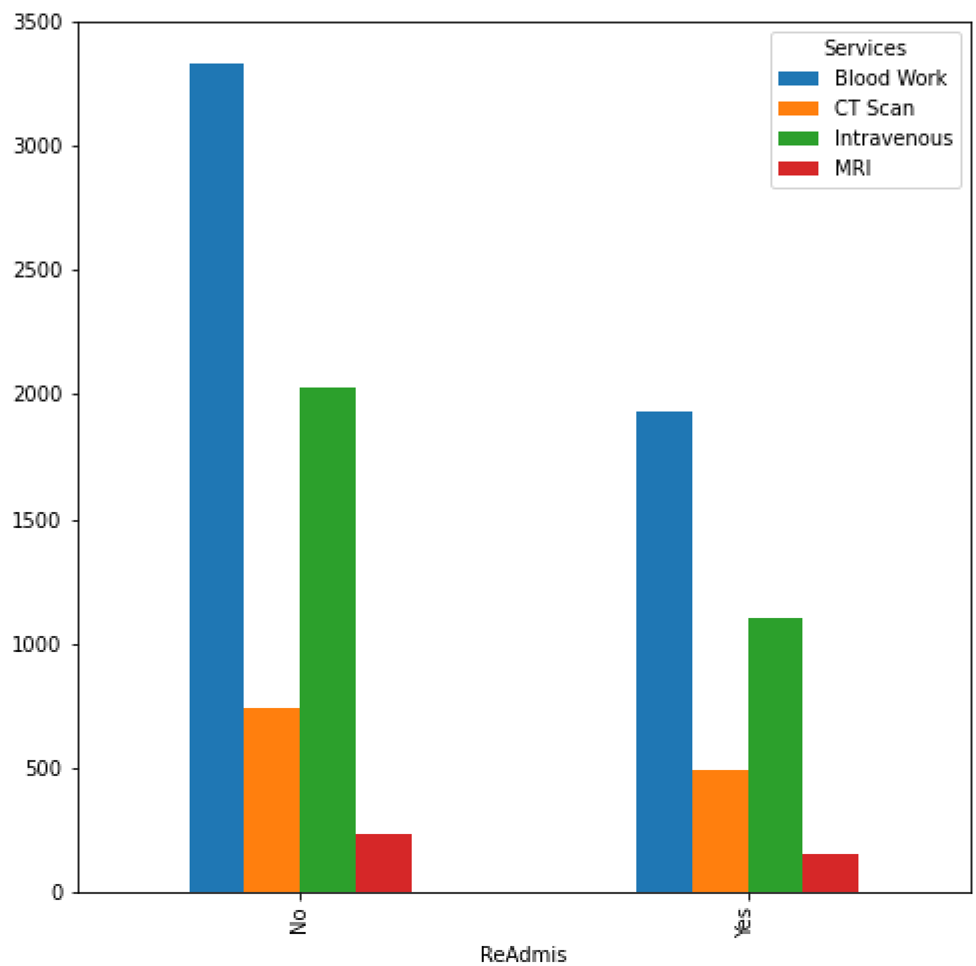
In [35]:
```
ReAdmis_Services_Grouped = pd.crosstab(index=data['ReAdmis'], columns=data['Se
rvices'])
ReAdmis_Services_Grouped
```

Out[35]:

Services	Blood Work	CT Scan	Intravenous	MRI
ReAdmis				
No	3335	737	2027	232
Yes	1930	488	1103	148

In [36]:
```
ReAdmis_Services_Grouped.plot(kind="bar", figsize=(8,8))
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x13b643134c8>



Complication Risk vs ReAdmissions (Categorical Variable)

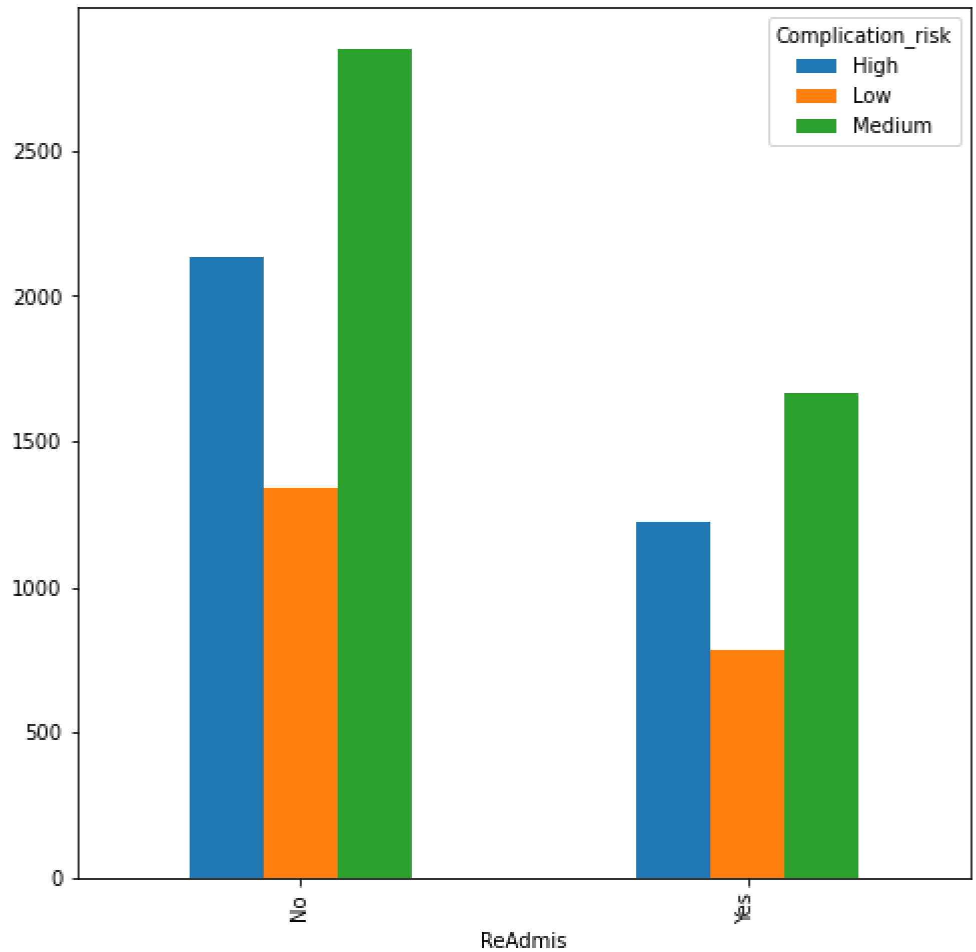
In [37]:
```
ReAdmis_CompRisk_Grouped = pd.crosstab(index=data['ReAdmis'], columns=data['Complication_risk'])
ReAdmis_CompRisk_Grouped
```

Out[37]:

Complication_risk	High	Low	Medium
ReAdmis			
No	2135	1343	2853
Yes	1223	782	1664

In [38]:
```
ReAdmis_CompRisk_Grouped.plot(kind="bar", figsize=(8,8))
```

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x13b6420bfc8>



Part E - Summary of the data analysis implications

E1 Results of the hypothesis test:

Of the sixteen categorical variables analyzed for a relationship to hospital readmissions, the null hypothesis was only rejected for one variable. The Chi Square tests revealed that there is a dependency between readmissions and the services rendered during the initial hospital stay. The following variables were determined to be independent of readmissions: gender, soft drinks, initial admissions reason, high blood pressure, stroke, complication risk level, overweight, arthritis, diabetes, hyperlipidemia, back pain, anxiety, allergic rhinitis, reflux esophagitis, and asthma.

E2 Limitations of the data analysis:

i. A major limitation of this data analysis, and hypothesis testing in general, is that the test does not explain the reason as to why a difference exists ("Limitations of Hypothesis testing in Research"). The results of this analysis simply identify where there are differences – further analysis and consultation with subject matter experts is required to understand why there are differences.

ii. The results of this analysis are based on probabilities ("Limitations of Hypothesis testing in Research"). There cannot be absolute certainty in the results.

E3 Recommended course of action:

i. It is recommended that the relationship between services during the initial stay and readmissions be explored in more detail. A great start would be to look for relationships between the types of services (MRI, Blood Work, etc.) and readmissions to try to identify any type of service that may indicate risk for readmissions. Subject matter experts should also be enlisted to further examine the relationship.

ii. It is imperative to find relationships to readmissions. Thus, re-running the Chi Square tests with a higher alpha, although potentially less accurate, may help find further relationships between categorical values.

iii. Statistical analysis should be expanded beyond categorical values to identify relationships between available data and readmissions.

Works Cited

Works Cited Bruce, P.A. (2020). Practical Statistics for Data Scientists, 50 Essential Concepts Using R and Python. Sebastopol, CA: O’Reilly Media, Incorporated. ISBN: 978-1792072942

Data to Fish. (n.d.). Retrieved from https://datatofish.com/round-values-pandas-dataframe/ (https://datatofish.com/round-values-pandas-dataframe/)

Limitations of Hypothesis testing in Research. (n.d.). Retrieved May 20, 2021, from https://www.wisdomjobs.com/e-university/research-methodology-tutorial-355/limitations-of-the-tests-of-hypotheses-11539.html (https://www.wisdomjobs.com/e-university/research-methodology-tutorial-355/limitations-of-the-tests-of-hypotheses-11539.html)

Naik, Krish. (2020). Tutorial 33- Chi Square Test Implementation with Python- Hypothesis Testing- Part 2 [Video]. Retrieved 20 May 2021, from https://www.youtube.com/watch?v=w5iKu1IrTJQ (https://www.youtube.com/watch?v=w5iKu1IrTJQ).

Pandas.crosstab. (n.d.). Retrieved from https://pandas.pydata.org/docs/reference/api/pandas.crosstab.html (https://pandas.pydata.org/docs/reference/api/pandas.crosstab.html)

Piush, Vaish. (2021, May 15). Visualise Categorical Variables in Python. Retrieved from https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/ (https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/)

Sewell, William (n.d.). Chi-Square for EDA D207 [Video]. Retrieved May 22, 2021, from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=52d9e72f-3309-4780-ac2b-accf014a436f (https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=52d9e72f-3309-4780-ac2b-accf014a436f)

In [ ]: