Section 1: EMR Bots 30-Day Readmission Activity¶

In [19]:

```
# import required modules

import csv
import pandas as pd
import numpy as np
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
```

In [15]:

```
#Section Q1) Create your dataframe

encounter_info = pd.read_csv(r'C:\Users\vitak\Downloads\Assignment 1 Datasets\readmissions\encounter_info.csv')
encounter_labs = pd.read_csv(r'C:\Users\vitak\Downloads\Assignment 1 Datasets\readmissions\encounter_labs.csv')
readmissions = pd.read_csv(r'C:\Users\vitak\Downloads\Assignment 1 Datasets\readmissions\readmission_outcome.csv')

alldata = pd.merge(pd.merge(encounter_info,encounter_labs,on='Encounter_ID'),readmissions,on='Encounter_ID')

alldata.head()
```

Out[15]:

	Patient_ID	Encounter_ID	AdmissionStartDate	AdmissionEndDate	PatientGender	PatientRace	PatientEncounterAge	Lab_DTTM	CBC..ABSOLUTE.LYMPHOCYTES	CBC..ABSOLUTE.NEUTROPHILS	...	CBC..RED.BLOOD.CELL.COUNT	CBC..WHITE.BLOOD.CELL.COUNT	METABOLIC..ALBUMIN	METABOLIC..BILI.TOTAL	METABOLIC..BUN	METABOLIC..CALCIUM	METABOLIC..CREATININE	METABOLIC..POTASSIUM	METABOLIC..SODIUM	outcome
0	100000	100000_2	1996-01-20 14:32:48.440	1996-02-06 17:11:05.247	Female	White	36.456455	1996-01-20 17:06:30	25.0	70.0	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
1	100000	100000_2	1996-01-20 14:32:48.440	1996-02-06 17:11:05.247	Female	White	36.456455	1996-01-20 17:19:57	25.0	64.6	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
2	100000	100000_2	1996-01-20 14:32:48.440	1996-02-06 17:11:05.247	Female	White	36.456455	1996-01-21 00:22:16	25.0	64.6	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
3	100000	100000_2	1996-01-20 14:32:48.440	1996-02-06 17:11:05.247	Female	White	36.456455	1996-01-21 00:30:20	25.0	64.6	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.8	140.0	0
4	100000	100000_2	1996-01-20 14:32:48.440	1996-02-06 17:11:05.247	Female	White	36.456455	1996-01-21 02:15:42	25.0	64.6	...	5.2	7.5	4.2	0.6	17.5	11.5	0.9	4.8	140.0	0

5 rows Ã 25 columns

Section 1 Question 1¶

In [28]:

```
#Age: Means and Standard Deviations

age_re_yes = alldata.loc[alldata['outcome']==1, 'PatientEncounterAge'].mean()

age_re_yes_sd = alldata.loc[alldata['outcome']==1, 'PatientEncounterAge'].std()

age_re_no = alldata.loc[alldata['outcome']==0, 'PatientEncounterAge'].mean()

age_re_no_sd = alldata.loc[alldata['outcome']==0, 'PatientEncounterAge'].std()

print(age_re_yes)
print(age_re_no)
print(age_re_yes_sd)
print(age_re_no_sd)
```

```
44.122799185314314
41.751522158101004
17.37523833562371
18.05639322612448
```

In [36]:

```
# There was some ambiguity about the requirements for this section.
# So I conducted the analysis on the entire dataset (labs, info, outcomes), and the outcomes data alone.


#chi square to compare categorical features

alldata_2 = pd.merge(encounter_info,readmissions,on='Encounter_ID')

outcome1 = alldata_2.loc[:,"outcome"]
age1 = alldata_2.loc[:,"PatientEncounterAge"]
gender1 = alldata_2.loc[:,"PatientGender"]
race1 = alldata_2.loc[:,"PatientRace"]

#Age CHI2 Analysis
age_chi2= pd.crosstab(age1, outcome1)
print(age_chi2)
print(stats.chi2_contingency(age_chi2))



#Gender CHI2 Analysis
gender_chi2= pd.crosstab(gender1, outcome1)
print(gender_chi2)
print(stats.chi2_contingency(gender_chi2))


#Race CHI2 Analysis
race_chi2= pd.crosstab(race1, outcome1)
print(race_chi2)
print(stats.chi2_contingency(race_chi2))
```

```
outcome              0  1
PatientEncounterAge
18.011880            1  0
18.015880            1  0
18.016663            1  0
18.019975            1  0
18.020753            1  0
...                 .. ..
92.323557            1  0
92.339812            1  0
92.364228            1  0
92.901825            1  0
92.958042            1  0

[36143 rows x 2 columns]
(36142.999999999985, 0.49752695298415395, 36142, array([[0.99645851, 0.00354149],
       [0.99645851, 0.00354149],
       [0.99645851, 0.00354149],
       ...,
       [0.99645851, 0.00354149],
       [0.99645851, 0.00354149],
       [0.99645851, 0.00354149]]))
outcome           0   1
PatientGender
Female        18812  64
Male          17203  64
(0.1734064984959499, 0.6771019211053179, 1, array([[18809.15087292,    66.84912708],
       [17205.84912708,    61.15087292]]))
outcome               0   1
PatientRace
African American   5382  21
Asian              8251  33
Unknown            4682  19
White             17700  55
(1.9654048475773294, 0.5796180166618972, 3, array([[5.38386534e+03, 1.91346595e+01],
       [8.25466231e+03, 2.93376864e+01],
       [4.68435147e+03, 1.66485350e+01],
       [1.76921209e+04, 6.28791191e+01]]))
```

In [23]:

```
#In this section, we calculate the encounters with(out) 30-day readmissions.
#Here, "Encounter" is defined as every patient visit, including all of the "LABS" dataset -- where patients have more than one visit.

def mysummary(colname,condition):
    total = len(alldata[colname])
    n_yes = len(alldata[(alldata[colname]==condition)&
                (alldata['outcome']==1)])
    n_no = len(alldata[(alldata[colname]==condition)&
                (alldata['outcome']==0)])
    per_yes = (n_yes/total)*100
    per_no = (n_no/total)*100
    return [condition,n_yes, per_yes, n_no, per_no]

fem = mysummary('PatientGender','Female')
male = mysummary('PatientGender','Male')
black = mysummary('PatientRace','African American')
white = mysummary('PatientRace','White')
asian = mysummary('PatientRace','Asian')
unknown = mysummary('PatientRace','Unknown')

Q1_table = [male,fem,black,white,asian,unknown]

my_table = pd.DataFrame(Q1_table, columns = ["Summary","Encounters with 30-day readmissions(n)","%","Encounters without 30-day readmissions(n)","%"])

display(my_table)
```

```
outcome           0   1
PatientGender
Female        18812  64
Male          17203  64
(0.1734064984959499, 0.6771019211053179, 1, array([[18809.15087292,    66.84912708],
       [17205.84912708,    61.15087292]]))
outcome               0   1
PatientRace
African American   5382  21
```

```
Asian              8251  33
Unknown            4682  19
White             17700  55
(1.9654048475773294, 0.5796180166618972, 3, array([[5.38386534e+03, 1.91346595e+01],
       [8.25466231e+03, 2.93376864e+01],
       [4.68435147e+03, 1.66485350e+01],
       [1.76921209e+04, 6.28791191e+01]]))
```

Section 1 Question 2¶

In [25]:

```
#simplify the "alldata" data set so you analyze only the last-observed lab values for each encounter

# drop irrelevant columns
alldata.drop([
    'Patient_ID',
    'AdmissionStartDate',
    'AdmissionEndDate',
],axis=1,inplace=True)

alldata_pruned = alldata.drop_duplicates(keep='last')

display(alldata_pruned)
alldata_pruned.info()
alldata_pruned.describe()
```

	Encounter_ID	PatientGender	PatientRace	PatientEncounterAge	Lab_DTTM	CBC..ABSOLUTE.LYMPHOCYTES	CBC..ABSOLUTE.NEUTROPHILS	CBC..BASOPHILS	CBC..EOSINOPHILS	CBC..HEMATOCRIT	...	CBC..RED.BLOOD.CELL.COUNT	CBC..WHITE.BLOOD.CELL.COUNT	METABOLIC..ALBUMIN	METABOLIC..BILI.TOTAL	METABOLIC..BUN	METABOLIC..CALCIUM	METABOLIC..CREATININE	METABOLIC..POTASSIUM	METABOLIC..SODIUM	outcome
0	100000_2	Female	White	36.456455	1996-01-20 17:06:30	25.0	70.0	0.1	0.3	42.5	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
1	100000_2	Female	White	36.456455	1996-01-20 17:19:57	25.0	64.6	0.1	0.3	42.5	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
2	100000_2	Female	White	36.456455	1996-01-21 00:22:16	25.0	64.6	0.1	0.3	42.8	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
3	100000_2	Female	White	36.456455	1996-01-21 00:30:20	25.0	64.6	0.1	0.3	42.8	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.8	140.0	0
4	100000_2	Female	White	36.456455	1996-01-21 02:15:42	25.0	64.6	0.1	0.3	42.8	...	5.2	7.5	4.2	0.6	17.5	11.5	0.9	4.8	140.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4902799	110000_5	Male	African American	58.494166	2012-11-07 01:39:35	15.7	67.6	0.1	0.5	49.7	...	6.4	8.3	3.8	0.2	9.0	9.0	0.6	5.0	131.1	0
4902800	110000_5	Male	African American	58.494166	2012-11-07 01:58:02	15.7	67.6	0.1	0.6	49.7	...	6.4	8.3	3.8	0.2	9.0	9.0	0.6	5.0	131.1	0
4902801	110000_5	Male	African American	58.494166	2012-11-07 05:42:19	15.7	67.6	0.1	0.6	49.7	...	6.4	8.3	3.8	0.2	9.0	9.0	0.6	3.5	131.1	0
4902802	110000_5	Male	African American	58.494166	2012-11-07 06:38:10	15.7	67.6	0.1	0.6	49.7	...	6.4	6.4	3.8	0.2	9.0	9.0	0.6	3.5	131.1	0
4902803	110000_5	Male	African American	58.494166	2012-11-07 06:57:41	15.7	67.6	0.1	0.6	49.7	...	6.4	6.4	3.8	0.1	9.0	9.0	0.6	3.5	131.1	0

4902768 rows Ã 22 columns

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4902768 entries, 0 to 4902803
Data columns (total 22 columns):
 #   Column                       Dtype  
---  ------                       -----  
 0   Encounter_ID                 object 
 1   PatientGender                object 
 2   PatientRace                  object 
 3   PatientEncounterAge          float64
 4   Lab_DTTM                     object 
 5   CBC..ABSOLUTE.LYMPHOCYTES    float64
 6   CBC..ABSOLUTE.NEUTROPHILS    float64
 7   CBC..BASOPHILS               float64
 8   CBC..EOSINOPHILS             float64
 9   CBC..HEMATOCRIT              float64
 10  CBC..HEMOGLOBIN              float64
 11  CBC..PLATELET.COUNT          float64
 12  CBC..RED.BLOOD.CELL.COUNT    float64
 13  CBC..WHITE.BLOOD.CELL.COUNT  float64
 14  METABOLIC..ALBUMIN           float64
 15  METABOLIC..BILI.TOTAL        float64
 16  METABOLIC..BUN               float64
 17  METABOLIC..CALCIUM           float64
 18  METABOLIC..CREATININE        float64
 19  METABOLIC..POTASSIUM         float64
 20  METABOLIC..SODIUM            float64
 21  outcome                      int64  
dtypes: float64(17), int64(1), object(4)
memory usage: 860.3+ MB
```

Out[25]:

	PatientEncounterAge	CBC..ABSOLUTE.LYMPHOCYTES	CBC..ABSOLUTE.NEUTROPHILS	CBC..BASOPHILS	CBC..EOSINOPHILS	CBC..HEMATOCRIT	CBC..HEMOGLOBIN	CBC..PLATELET.COUNT	CBC..RED.BLOOD.CELL.COUNT	CBC..WHITE.BLOOD.CELL.COUNT	METABOLIC..ALBUMIN	METABOLIC..BILI.TOTAL	METABOLIC..BUN	METABOLIC..CALCIUM	METABOLIC..CREATININE	METABOLIC..POTASSIUM	METABOLIC..SODIUM	outcome
count	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06	4.902768e+06
mean	4.175990e+01	2.499706e+01	7.000121e+01	1.086038e-01	3.369535e-01	4.249499e+01	1.450186e+01	2.850215e+02	5.003539e+00	7.513503e+00	4.244626e+00	5.999477e-01	1.750029e+01	9.501873e+00	8.541084e-01	4.497314e+00	1.399799e+02	3.533514e-03
std	1.805458e+01	5.568243e+00	5.570806e+00	7.077017e-02	1.498486e-01	6.954853e+00	2.507913e+00	9.203077e+01	1.114242e+00	2.505532e+00	9.766367e-01	3.372594e-01	6.962744e+00	1.392525e+00	1.994363e-01	8.370857e-01	8.356385e+00	5.933826e-02
min	1.801188e+01	1.500000e+01	6.000000e+01	0.000000e+00	1.000000e-01	3.000000e+01	1.000000e+01	1.200000e+02	3.000000e+00	3.000000e+00	2.500000e+00	0.000000e+00	5.000000e+00	7.000000e+00	5.000000e-01	3.000000e+00	1.250000e+02	0.000000e+00
25%	2.585804e+01	2.030000e+01	6.540000e+01	1.000000e-01	2.000000e-01	3.670000e+01	1.240000e+01	2.084000e+02	4.100000e+00	5.400000e+00	3.400000e+00	3.000000e-01	1.170000e+01	8.300000e+00	7.000000e-01	3.800000e+00	1.330000e+02	0.000000e+00
50%	3.861599e+01	2.500000e+01	7.000000e+01	1.000000e-01	3.000000e-01	4.250000e+01	1.450000e+01	2.851000e+02	5.000000e+00	7.500000e+00	4.200000e+00	6.000000e-01	1.750000e+01	9.500000e+00	9.000000e-01	4.500000e+00	1.400000e+02	0.000000e+00
75%	5.439088e+01	2.960000e+01	7.460000e+01	2.000000e-01	5.000000e-01	4.830000e+01	1.660000e+01	3.616000e+02	5.900000e+00	9.600000e+00	5.100000e+00	9.000000e-01	2.330000e+01	1.070000e+01	1.000000e+00	5.200000e+00	1.469000e+02	0.000000e+00
max	9.295804e+01	3.500000e+01	8.000000e+01	2.000000e-01	6.000000e-01	5.500000e+01	1.900000e+01	4.500000e+02	7.000000e+00	1.200000e+01	6.000000e+00	1.200000e+00	3.000000e+01	1.200000e+01	1.200000e+00	6.000000e+00	1.550000e+02	1.000000e+00

Section 1 Quesiton 3¶

In [29]:

```
#Use the Lab date to create a column for Years.

alldata["Year"] = alldata.Lab_DTTM.str[:4]

alldata["Year"] = alldata["Year"].astype(int)

#display(alldata)
```

In [18]:

```
#Train Test Split

train_data = alldata[alldata["Year"] <= 2003]

display(train_data)
```

	Encounter_ID	PatientGender	PatientRace	PatientEncounterAge	Year	Lab_DTTM	CBC..ABSOLUTE.LYMPHOCYTES	CBC..ABSOLUTE.NEUTROPHILS	CBC..BASOPHILS	CBC..EOSINOPHILS	...	CBC..RED.BLOOD.CELL.COUNT	CBC..WHITE.BLOOD.CELL.COUNT	METABOLIC..ALBUMIN	METABOLIC..BILI.TOTAL	METABOLIC..BUN	METABOLIC..CALCIUM	METABOLIC..CREATININE	METABOLIC..POTASSIUM	METABOLIC..SODIUM	outcome
0	100000_2	Female	White	36.456455	1996	1996-01-20 17:06:30	25.0	70.0	0.1	0.3	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
1	100000_2	Female	White	36.456455	1996	1996-01-20 17:19:57	25.0	64.6	0.1	0.3	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
2	100000_2	Female	White	36.456455	1996	1996-01-21 00:22:16	25.0	64.6	0.1	0.3	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.5	140.0	0
3	100000_2	Female	White	36.456455	1996	1996-01-21 00:30:20	25.0	64.6	0.1	0.3	...	5.0	7.5	4.2	0.6	17.5	11.5	0.9	4.8	140.0	0
4	100000_2	Female	White	36.456455	1996	1996-01-21 02:15:42	25.0	64.6	0.1	0.3	...	5.2	7.5	4.2	0.6	17.5	11.5	0.9	4.8	140.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4902642	110000_2	Male	African American	30.322346	1984	1984-09-04 19:17:23	19.9	60.1	0.1	0.6	...	3.9	9.6	3.7	0.5	25.5	11.3	0.6	3.1	132.3	0
4902643	110000_2	Male	African American	30.322346	1984	1984-09-04 19:49:36	19.9	60.1	0.1	0.6	...	3.9	9.6	3.7	0.5	25.5	11.3	0.6	3.1	140.8	0
4902644	110000_2	Male	African American	30.322346	1984	1984-09-04 19:55:37	19.9	60.1	0.1	0.6	...	3.9	9.6	3.7	1.1	25.5	11.3	0.6	3.1	140.8	0
4902645	110000_2	Male	African American	30.322346	1984	1984-09-04 20:23:38	19.9	60.1	0.1	0.6	...	3.9	9.6	4.5	1.1	25.5	11.3	0.6	3.1	140.8	0
4902646	110000_2	Male	African American	30.322346	1984	1984-09-04 21:06:50	19.9	60.1	0.1	0.6	...	3.9	9.6	4.5	1.1	25.5	11.3	0.6	3.1	140.8	0

2751795 rows Ã 23 columns

In [20]:

```
#Test Test Split

test_data = alldata[alldata["Year"] > 2003]

display(test_data)
```

	Encounter_ID	PatientGender	PatientRace	PatientEncounterAge	Year	Lab_DTTM	CBC..ABSOLUTE.LYMPHOCYTES	CBC..ABSOLUTE.NEUTROPHILS	CBC..BASOPHILS	CBC..EOSINOPHILS	...	CBC..RED.BLOOD.CELL.COUNT	CBC..WHITE.BLOOD.CELL.COUNT	METABOLIC..ALBUMIN	METABOLIC..BILI.TOTAL	METABOLIC..BUN	METABOLIC..CALCIUM	METABOLIC..CREATININE	METABOLIC..POTASSIUM	METABOLIC..SODIUM	outcome
257	100000_5	Female	White	47.688073	2007	2007-04-12 03:57:36	25.0	70.0	0.1	0.3	...	5.0	7.5	4.2	0.6	17.5	9.5	0.9	4.5	140.0	0
258	100000_5	Female	White	47.688073	2007	2007-04-12 04:36:36	20.4	70.0	0.1	0.3	...	5.0	7.5	4.2	0.6	17.5	9.5	0.9	4.5	140.0	0
259	100000_5	Female	White	47.688073	2007	2007-04-12 05:06:34	20.4	70.0	0.1	0.3	...	5.0	7.5	4.2	0.1	17.5	9.5	0.9	4.5	140.0	0
260	100000_5	Female	White	47.688073	2007	2007-04-12 11:21:12	20.4	70.0	0.1	0.3	...	5.0	7.5	4.2	0.1	17.5	9.5	0.9	3.3	140.0	0
261	100000_5	Female	White	47.688073	2007	2007-04-12 11:46:41	20.4	70.0	0.1	0.3	...	5.0	7.5	4.2	0.1	11.3	9.5	0.9	3.3	140.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4902799	110000_5	Male	African American	58.494166	2012	2012-11-07 01:39:35	15.7	67.6	0.1	0.5	...	6.4	8.3	3.8	0.2	9.0	9.0	0.6	5.0	131.1	0
4902800	110000_5	Male	African American	58.494166	2012	2012-11-07 01:58:02	15.7	67.6	0.1	0.6	...	6.4	8.3	3.8	0.2	9.0	9.0	0.6	5.0	131.1	0
4902801	110000_5	Male	African American	58.494166	2012	2012-11-07 05:42:19	15.7	67.6	0.1	0.6	...	6.4	8.3	3.8	0.2	9.0	9.0	0.6	3.5	131.1	0
4902802	110000_5	Male	African American	58.494166	2012	2012-11-07 06:38:10	15.7	67.6	0.1	0.6	...	6.4	6.4	3.8	0.2	9.0	9.0	0.6	3.5	131.1	0
4902803	110000_5	Male	African American	58.494166	2012	2012-11-07 06:57:41	15.7	67.6	0.1	0.6	...	6.4	6.4	3.8	0.1	9.0	9.0	0.6	3.5	131.1	0

2151009 rows Ã 23 columns

In [22]:

```
#Test Data

test_yes = len(test_data[(test_data['outcome']==1)])
perc_test_yes = (test_yes/(len(test_data['outcome'])))*100
test_no = len(test_data[(test_data['outcome']==0)])
perc_test_no = (test_no/(len(test_data['outcome'])))*100

print(test_yes,perc_test_yes,test_no,perc_test_no)
```

```
8776 0.4079945737093615 2142233 99.59200542629064
```

In [23]:

```
#Section 1 Q3
#Train Data

train_yes = len(train_data[(train_data['outcome']==1)])
perc_train_yes = (train_yes/(len(train_data['outcome'])))*100
train_no = len(train_data[(train_data['outcome']==0)])
perc_train_no = (train_no/(len(train_data['outcome'])))*100

print(train_yes,perc_train_yes,train_no,perc_train_no)
```

```
8548 0.3106336046108086 2743247 99.68936639538919
```

Section 2: GUSTO 30-Day Mortality Prediction¶

In [38]:

```
#Section Q1) Create your dataframe

gusto = pd.read_csv(r'C:\Users\vitak\Downloads\gusto_data.csv')

#display(alldata)
gusto.head()
```

Out[38]:

	DAY30	AGE	A65	SEX	KILLIP	SHO	DIA	HYP	HRT	ANT	...	WEI	SMK	HTN	LIP	PAN	FAM	STE	ST4	TTR	GROUP
0	0	70.313	1	0	1	0	0	0	0	1	...	84.0	3	1	1	0	0	1	0	1	west
1	0	59.844	0	0	1	0	1	0	0	1	...	115.0	1	1	0	0	1	6	1	0	west
2	0	59.023	0	0	1	0	0	0	1	0	...	76.0	1	1	0	0	1	3	0	0	west
3	1	80.375	1	1	1	0	0	0	1	0	...	50.0	3	0	0	0	0	3	0	0	west
4	0	64.750	0	0	1	0	0	1	0	0	...	97.4	1	0	0	1	1	2	0	1	west

5 rows Ã 23 columns

Section 2 Question 1¶

In [39]:

```
#GUSTO Age Means and Standard Deviations

age_g_yes = gusto.loc[gusto['DAY30']==1, 'AGE'].mean()

age_g_yes_sd = gusto.loc[gusto['DAY30']==1, 'AGE'].std()

age_g_no = gusto.loc[gusto['DAY30']==0, 'AGE'].mean()

age_g_no_sd = gusto.loc[gusto['DAY30']==0, 'AGE'].std()

print(age_g_yes)
print(age_g_yes_sd)
print(age_g_no)
print(age_g_no_sd)
```

```
71.07616883116889
10.717872813449745
60.16128192419796
11.555210346445092
```

In [40]:

```
#GUSTO Gender and Group Summary

def gsummary(colname,condition):
    total = len(gusto[colname])
    n_yes = len(gusto[(gusto[colname]==condition)&
                (gusto['DAY30']==1)])
    n_no = len(gusto[(gusto[colname]==condition)&
                (gusto['DAY30']==0)])
    per_yes = (n_yes/total)*100
    per_no = (n_no/total)*100
    return [condition,n_yes, per_yes, n_no, per_no]

gmale = gsummary('SEX',0)
gfem = gsummary('SEX',1)
s2 = gsummary('GROUP','sample2')
s4 = gsummary('GROUP','sample4')
s5 = gsummary('GROUP','sample5')
west = gsummary('GROUP','west')

Q2_table = [gmale,gfem,s2,s4,s5,west]

my_gtable = pd.DataFrame(Q2_table, columns = ["Summary","Patients who died within 30 days(n)","%","Patients who were alive at 30 days(n)","%"])

display(my_gtable)
```

	Summary	Patients who died within 30 days(n)	%	Patients who were alive at 30 days(n)	%
0	0	140	3.824092	2581	70.499863
1	1	91	2.485660	849	23.190385
2	sample2	20	0.546299	239	6.528271
3	sample4	52	1.420377	733	20.021852
4	sample5	24	0.655559	405	11.062551
5	west	135	3.687517	2053	56.077574

In [41]:

```
#chi square to compare categorical features - for P Values

outcome2 = gusto.loc[:,"DAY30"]
age2 = gusto.loc[:,"AGE"]
gender2 = gusto.loc[:,"SEX"]
race2 = gusto.loc[:,"GROUP"]

#Gusto Age CHI2 Analysis
age2_chi2 = pd.crosstab(age2, outcome2)
print(age2_chi2)
print(stats.chi2_contingency(age2_chi2))

# Gusto Gender CHI2 Analysis
gender2_chi2= pd.crosstab(gender2, outcome2)
print(gender2_chi2)
print(stats.chi2_contingency(gender2_chi2))


#Gusto Race CHI2 Analysis
race2_chi2= pd.crosstab(race2, outcome2)
print(race2_chi2)
print(stats.chi2_contingency(race2_chi2))
```

```
DAY30   0  1
AGE
23.910  1  0
25.891  2  0
27.754  2  0
28.172  1  0
28.367  1  0
...    .. ..
88.656  0  1
88.828  2  0
88.844  1  0
89.078  0  1
89.484  0  1

[2100 rows x 2 columns]
(2596.5108711016874, 4.208975122544854e-13, 2099, array([[0.93690249, 0.06309751],
       [1.87380497, 0.12619503],
       [1.87380497, 0.12619503],
       ...,
       [0.93690249, 0.06309751],
       [0.93690249, 0.06309751],
       [0.93690249, 0.06309751]]))
DAY30      0    1
SEX
0       2581  140
1        849   91
(23.551637842815246, 1.2160586790628317e-06, 1, array([[2549.31166348,  171.68833652],
       [ 880.68833652,   59.31166348]]))
DAY30        0    1
GROUP
sample2   239   20
sample4   733   52
```

```
sample5   405   24
west     2053  135
(1.4487351294747421, 0.6941517478306458, 3, array([[ 242.65774379,   16.34225621],
       [ 735.46845124,   49.53154876],
       [ 401.93116635,   27.06883365],
       [2049.94263862,  138.05736138]]))
```

Section 2 Question 2¶

In [42]:

```
#Split GUSTO Data into Training and Test Sets.

#Train Data (sample 2-5)

gtrain_data = gusto[gusto["GROUP"] != "west"]

display(gtrain_data)
```

	DAY30	AGE	A65	SEX	KILLIP	SHO	DIA	HYP	HRT	ANT	...	WEI	SMK	HTN	LIP	PAN	FAM	STE	ST4	TTR	GROUP
2188	0	59.844	0	0	1	0	1	0	0	1	...	115.0	1	1	0	0	1	6	1	0	sample2
2189	1	80.375	1	1	1	0	0	0	1	0	...	50.0	3	0	0	0	0	3	0	0	sample2
2190	1	34.430	0	1	1	0	1	0	1	1	...	93.0	1	1	1	1	1	5	1	1	sample2
2191	0	57.977	0	1	2	0	1	0	0	0	...	59.0	3	1	1	0	1	1	0	0	sample2
2192	0	62.305	0	0	1	0	0	0	0	1	...	87.0	2	0	0	1	0	2	0	1	sample2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3656	0	51.836	0	0	1	0	0	0	0	1	...	65.0	1	0	0	0	0	7	1	1	sample5
3657	0	51.836	0	0	1	0	1	0	0	1	...	72.1	3	0	0	1	0	7	1	1	sample5
3658	0	63.977	0	0	1	0	0	0	0	0	...	87.2	2	1	0	0	0	2	0	1	sample5
3659	0	59.758	0	1	1	0	0	0	0	0	...	57.8	3	0	0	0	0	6	1	1	sample5
3660	0	52.195	0	0	1	0	0	0	0	0	...	75.0	2	1	1	0	1	4	0	1	sample5

1473 rows Ã   23 columns

In [43]:

```
#GUSTO Test Data (west)

gtest_data = gusto[gusto["GROUP"] == "west"]

display(gtest_data)
```

	DAY30	AGE	A65	SEX	KILLIP	SHO	DIA	HYP	HRT	ANT	...	WEI	SMK	HTN	LIP	PAN	FAM	STE	ST4	TTR	GROUP
0	0	70.313	1	0	1	0	0	0	0	1	...	84.0	3	1	1	0	0	1	0	1	west
1	0	59.844	0	0	1	0	1	0	0	1	...	115.0	1	1	0	0	1	6	1	0	west
2	0	59.023	0	0	1	0	0	0	1	0	...	76.0	1	1	0	0	1	3	0	0	west
3	1	80.375	1	1	1	0	0	0	1	0	...	50.0	3	0	0	0	0	3	0	0	west
4	0	64.750	0	0	1	0	0	1	0	0	...	97.4	1	0	0	1	1	2	0	1	west
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2183	0	64.609	0	0	1	0	0	0	1	1	...	72.0	3	1	1	0	0	8	1	0	west
2184	0	70.766	1	0	1	0	0	1	0	0	...	97.0	2	0	1	0	1	6	1	1	west
2185	0	53.047	0	1	1	0	0	0	1	1	...	91.0	3	1	0	0	0	7	1	0	west
2186	0	73.141	1	0	1	0	0	0	0	1	...	79.4	2	1	0	0	0	7	1	0	west
2187	0	52.195	0	0	1	0	0	0	0	0	...	75.0	2	1	1	0	1	4	0	1	west

2188 rows Ã   23 columns

In [44]:

```
#GUSTO Test/Train Data Summary

#Training Data

gtrain_yes = len(gtrain_data[(gtrain_data['DAY30']==1)])
gperc_train_yes = (gtrain_yes/(len(gtrain_data['DAY30'])))*100
gtrain_no = len(gtrain_data[(gtrain_data['DAY30']==0)])
gperc_train_no = (gtrain_no/(len(gtrain_data['DAY30'])))*100

print(gtrain_yes,gperc_train_yes,gtrain_no,gperc_train_no)
```

```
96 6.517311608961303 1377 93.48268839103869
```

In [45]:

```
#GUSTO Test/Train Data Summary

#Section 2 Q2
#Test Data

gtest_yes = len(gtest_data[(gtest_data['DAY30']==1)])
gperc_test_yes = (gtest_yes/(len(gtest_data['DAY30'])))*100
gtest_no = len(gtest_data[(gtest_data['DAY30']==0)])
gperc_test_no = (gtest_no/(len(gtest_data['DAY30'])))*100

print(gtest_yes,gperc_test_yes,gtest_no,gperc_test_no)
```

```
135 6.170018281535649 2053 93.82998171846435
```

Section 2 Question 3¶

In [46]:

```
#Summaries of gtest and gtrain data

gtest_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2188 entries, 0 to 2187
Data columns (total 23 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   DAY30   2188 non-null   int64  
 1   AGE     2188 non-null   float64
 2   A65     2188 non-null   int64  
 3   SEX     2188 non-null   int64  
 4   KILLIP  2188 non-null   int64  
 5   SHO     2188 non-null   int64  
 6   DIA     2188 non-null   int64  
 7   HYP     2188 non-null   int64  
 8   HRT     2188 non-null   int64  
 9   ANT     2188 non-null   int64  
 10  PMI     2188 non-null   int64  
 11  HIG     2188 non-null   int64  
 12  HEI     2188 non-null   float64
 13  WEI     2188 non-null   float64
 14  SMK     2188 non-null   int64  
 15  HTN     2188 non-null   int64  
 16  LIP     2188 non-null   int64  
 17  PAN     2188 non-null   int64  
 18  FAM     2188 non-null   int64  
 19  STE     2188 non-null   int64  
 20  ST4     2188 non-null   int64  
 21  TTR     2188 non-null   int64  
 22  GROUP   2188 non-null   object 
dtypes: float64(3), int64(19), object(1)
memory usage: 410.2+ KB
```

In [22]:

```
gtrain_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1473 entries, 2188 to 3660
Data columns (total 23 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   DAY30   1473 non-null   int64  
 1   AGE     1473 non-null   float64
 2   A65     1473 non-null   int64  
 3   SEX     1473 non-null   int64  
 4   KILLIP  1473 non-null   int64  
 5   SHO     1473 non-null   int64  
 6   DIA     1473 non-null   int64  
 7   HYP     1473 non-null   int64  
 8   HRT     1473 non-null   int64  
 9   ANT     1473 non-null   int64  
 10  PMI     1473 non-null   int64  
 11  HIG     1473 non-null   int64  
 12  HEI     1473 non-null   float64
 13  WEI     1473 non-null   float64
 14  SMK     1473 non-null   int64  
 15  HTN     1473 non-null   int64  
 16  LIP     1473 non-null   int64  
 17  PAN     1473 non-null   int64  
 18  FAM     1473 non-null   int64  
 19  STE     1473 non-null   int64  
 20  ST4     1473 non-null   int64  
 21  TTR     1473 non-null   int64  
 22  GROUP   1473 non-null   object 
dtypes: float64(3), int64(19), object(1)
memory usage: 276.2+ KB
```

In [47]:

```
gtest_data.describe()
```

Out[47]:

	DAY30	AGE	A65	SEX	KILLIP	SHO	DIA	HYP	HRT	ANT	...	HEI	WEI	SMK	HTN	LIP	PAN	FAM	STE	ST4	TTR
count	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	...	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000	2188.000000
mean	0.061700	60.469186	0.383455	0.248629	1.132084	0.014625	0.142596	0.096435	0.333638	0.372486	...	172.129936	82.888940	1.866545	0.403565	0.404936	0.340494	0.475777	3.999543	0.356033	0.608775
std	0.240665	12.026568	0.486339	0.432317	0.409550	0.120075	0.349740	0.295254	0.471620	0.483577	...	10.094343	17.692498	0.821252	0.490724	0.490992	0.473984	0.499527	1.878451	0.478935	0.488136
min	0.000000	23.910000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	140.900000	36.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	50.932000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	165.100000	70.900000	1.000000	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000	0.000000
50%	0.000000	60.547000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	173.000000	82.000000	2.000000	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000	1.000000
75%	0.000000	69.922000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	1.000000	...	180.000000	92.050000	3.000000	1.000000	1.000000	1.000000	1.000000	5.000000	1.000000	1.000000
max	1.000000	89.484000	1.000000	1.000000	4.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	205.700000	180.000000	3.000000	1.000000	1.000000	1.000000	1.000000	11.000000	1.000000	1.000000

8 rows Ã 22 columns

In [48]:

```
gtrain_data.describe()
```

Out[48]:

	DAY30	AGE	A65	SEX	KILLIP	SHO	DIA	HYP	HRT	ANT	...	HEI	WEI	SMK	HTN	LIP	PAN	FAM	STE	ST4	TTR
count	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	...	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000	1473.000000
mean	0.065173	61.415623	0.410726	0.268839	1.194840	0.020367	0.114732	0.073320	0.292600	0.361168	...	170.338900	78.200068	1.909029	0.385608	0.386286	0.364562	0.428377	4.150034	0.386965	0.560760
std	0.246915	11.448781	0.492133	0.443507	0.462655	0.141299	0.318806	0.260749	0.455111	0.480502	...	9.779777	16.531963	0.803275	0.486904	0.487063	0.481471	0.495012	1.865345	0.487221	0.496463
min	0.000000	25.891000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	141.000000	37.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	52.578000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	163.800000	68.000000	1.000000	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000	0.000000
50%	0.000000	62.242000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	170.200000	77.000000	2.000000	0.000000	0.000000	0.000000	0.000000	4.000000	0.000000	1.000000
75%	0.000000	70.469000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000	1.000000	...	177.300000	87.000000	3.000000	1.000000	1.000000	1.000000	1.000000	6.000000	1.000000	1.000000
max	1.000000	88.828000	1.000000	1.000000	4.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	199.700000	180.000000	3.000000	1.000000	1.000000	1.000000	1.000000	10.000000	1.000000	1.000000

8 rows Ã 22 columns