Analyze Open Diabetes Data

Monitoring lab values related to diabetes is an important area for people who are at risk for diabetes. Over 37 million Americans have diabetes (about 1 in 10), and approximately 90-95% of them have type 2 diabetes (CDC website, accessed Jan 10, 2023). Type 2 diabetes most often develops in people over age 45, but more and more children, teens, and young adults are also developing it (CDC website, accessed Jan 10, 2023).

A1C result is an important lab for diabetes monitoring, which measures the amount of hemoglobin bound to glucose and reflects average blood glucose levels over the past 3 months. The A1C test result is reported as a percentage. The higher the A1C percentage is, the higher the blood glucose levels are. A normal A1C level is below 5.7 percent. For diabetics, however, the American Diabetes Association recommends ensuring A1C result values are less than 7%. (CDC website, accessed Jan 10, 2023).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.api.types import CategoricalDtype
import seaborn as sns
!pip install plotly
!pip install cufflinks
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: plotly in /usr/local/lib/python3.8/dist-packages (5.5.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.8/dist-packages (from plotly) (8.1.0)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from plotly) (1.15.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: cufflinks in /usr/local/lib/python3.8/dist-packages (0.17.3)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (1.15.0)
Requirement already satisfied: colorlover>=0.2.1 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (0.3.0)
Requirement already satisfied: numpy>=1.9.2 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (1.21.6)
Requirement already satisfied: setuptools>=34.4.1 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (57.4.0
Requirement already satisfied: plotly>=4.1.1 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (5.5.0)
Requirement already satisfied: ipywidgets>=7.0.0 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (7.7.1)
Requirement already satisfied: ipython>=5.3.0 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (7.9.0)
Requirement already satisfied: pandas>=0.19.2 in /usr/local/lib/python3.8/dist-packages (from cufflinks) (1.3.5)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.8/dist-packages (from ipython>=5.3.0->cuffli
Requirement already satisfied: pickleshare in /usr/local/lib/python3.8/dist-packages (from ipython>=5.3.0->cufflinks
Requirement already satisfied: backcall in /usr/local/lib/python3.8/dist-packages (from ipython>=5.3.0->cufflinks) (
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from ipython>
Requirement already satisfied: pexpect in /usr/local/lib/python3.8/dist-packages (from ipython>=5.3.0->cufflinks) (4
Requirement already satisfied: pygments in /usr/local/lib/python3.8/dist-packages (from ipython>=5.3.0->cufflinks) (
Requirement already satisfied: decorator in /usr/local/lib/python3.8/dist-packages (from ipython>=5.3.0->cufflinks)
Requirement already satisfied: jedi>=0.10 in /usr/local/lib/python3.8/dist-packages (from ipython>=5.3.0->cufflinks)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.8/dist-packages (from ipywidgets>
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/python3.8/dist-packages (from ipywidgets>
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.8/dist-packages (from ipywidgets>=7.0.0->c
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.8/dist-packages (from ipywidgets>=7
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=0.19.2
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=0.19.2->cufflink
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.8/dist-packages (from plotly>=4.1.1->cuffli
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.8/dist-packages (from ipykernel>=4.5.1->ipywid
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.8/dist-packages (from ipykernel>=4.5.1->ipyw
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.8/dist-packages (from jedi>=0.10->ipyth
Requirement already satisfied: wcwidth in /usr/local/lib/python3.8/dist-packages (from prompt-toolkit<2.1.0,>=2.0.0-
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.8/dist-packages (from widgetsnbextension~=3
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.8/dist-packages (from pexpect->ipython>=5.3
Requirement already satisfied: nbformat in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->widgetsnbex
Requirement already satisfied: nbconvert<6.0 in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->widget
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->wi
Requirement already satisfied: jinja2<=3.0.0 in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->widget
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->widgetsnbe
Requirement already satisfied: jupyter-core>=4.4.0 in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->widgetsnb
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.8/dist-packages (from notebook>=4.4.1->wid
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.8/dist-packages (from jinja2<=3.0.0->noteb
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.8/dist-packages (from jupyter-core>=4.4.0
Requirement already satisfied: testpath in /usr/local/lib/python3.8/dist-packages (from nbconvert<6.0->notebook>=4.4
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.8/dist-packages (from nbconvert<6.0->note
Requirement already satisfied: bleach in /usr/local/lib/python3.8/dist-packages (from nbconvert<6.0->notebook>=4.4.1
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.8/dist-packages (from nbconvert<6.0->not
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.8/dist-packages (from nbconvert<6.0->n
Requirement already satisfied: defusedxml in /usr/local/lib/python3.8/dist-packages (from nbconvert<6.0->notebook>=4
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.8/dist-packages (from nbformat->notebook>=4.
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.8/dist-packages (from nbformat->notebook>=4
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr/local/lib/python3.8/dist-packag
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.8/dist-packages (from jsonschema>=2.6->nbform
Requirement already satisfied: importlib-resources>=1.4.0 in /usr/local/lib/python3.8/dist-packages (from jsonschema
Requirement already satisfied: webencodings in /usr/local/lib/python3.8/dist-packages (from bleach->nbconvert<6.0->n
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.8/dist-packages (from importlib-resources>=1.4.
```

```python
sns.set()
```

```python
df = pd.read_csv('https://raw.githubusercontent.com/niteen11/DataAnalyticsAcademy/master/Python/dataset_diabetes/diabet
```

```python
df.shape
```

```
(101766, 49)
```

```python
df.head(10)
```

	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admi
encounter_id								
2278392	8222157	Caucasian	Female	[0-10)	?	6	25	
149190	55629189	Caucasian	Female	[10-20)	?	1	1	
64410	86047875	AfricanAmerican	Female	[20-30)	?	1	1	
500364	82442376	Caucasian	Male	[30-40)	?	1	1	
16680	42519267	Caucasian	Male	[40-50)	?	1	1	
35754	82637451	Caucasian	Male	[50-60)	?	2	1	

35754	82637451	Caucasian	Male	[50-60)	?	2	1
55842	84259809	Caucasian	Male	[60-70)	?	3	1
63768	114882984	Caucasian	Male	[70-80)	?	1	1
12522	48330783	Caucasian	Female	[80-90)	?	2	1
15738	63555939	Caucasian	Female	[90-100)	?	3	3

```
df.tail()
#head and tail show ? as missing data
```

	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admis
encounter_id								
443847548	100162476	AfricanAmerican	Male	[70-80)	?	1	3	
443847782	74694222	AfricanAmerican	Female	[80-90)	?	1	4	
443854148	41088789	Caucasian	Male	[70-80)	?	1	1	
443857166	31693671	Caucasian	Female	[80-90)	?	2	3	
443867222	175429310	Caucasian	Male	[70-80)	?	1	1	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 101766 entries, 2278392 to 443867222
Data columns (total 49 columns):
 #   Column                    Non-Null Count   Dtype 
---  ------                    --------------   ----- 
 0   patient_nbr               101766 non-null  int64 
 1   race                      101766 non-null  object
 2   gender                    101766 non-null  object
 3   age                       101766 non-null  object
 3   age                       101766 non-null  object
 4   weight                    101766 non-null  object
 5   admission_type_id         101766 non-null  int64
 6   discharge_disposition_id  101766 non-null  int64
 7   admission_source_id       101766 non-null  int64
 8   time_in_hospital          101766 non-null  int64
 9   payer_code                101766 non-null  object
 10  medical_specialty         101766 non-null  object
 11  num_lab_procedures        101766 non-null  int64
 12  num_procedures            101766 non-null  int64
 13  num_medications           101766 non-null  int64
 14  number_outpatient         101766 non-null  int64
 15  number_emergency          101766 non-null  int64
 16  number_inpatient          101766 non-null  int64
 17  diag_1                    101766 non-null  object
 18  diag_2                    101766 non-null  object
 19  diag_3                    101766 non-null  object
 20  number_diagnoses          101766 non-null  int64
 21  max_glu_serum             101766 non-null  object
 22  A1Cresult                 101766 non-null  object
 23  metformin                 101766 non-null  object
 24  repaglinide               101766 non-null  object
 25  nateglinide               101766 non-null  object
 26  chlorpropamide            101766 non-null  object
 27  glimepiride               101766 non-null  object
 28  acetohexamide             101766 non-null  object
 29  glipizide                 101766 non-null  object
 30  glyburide                 101766 non-null  object
 31  tolbutamide               101766 non-null  object
 32  pioglitazone              101766 non-null  object
 33  rosiglitazone             101766 non-null  object
 34  acarbose                  101766 non-null  object
 35  miglitol                  101766 non-null  object
 36  troglitazone              101766 non-null  object
 37  tolazamide                101766 non-null  object
 38  examide                   101766 non-null  object
 39  citoglipton               101766 non-null  object
 40  insulin                   101766 non-null  object
 41  glyburide-metformin       101766 non-null  object
 42  glipizide-metformin       101766 non-null  object
 43  glimepiride-pioglitazone  101766 non-null  object
 44  metformin-rosiglitazone   101766 non-null  object
 45  metformin-pioglitazone    101766 non-null  object
 46  change                    101766 non-null  object
 47  diabetesMed               101766 non-null  object
 48  readmitted                101766 non-null  object
dtypes: int64(12), object(37)
memory usage: 38.8+ MB
```

```
#One possible assumption in this dataset on diabetes is that many patients have diabetes.
#To truly confirm if diabetes, consider adding a brand new column for future database from medical history, confirming
# In protocols, the disease needs to be diagnosed by a qualified physician to be eligible for a diabetes related treatm
# If a patient is pre-diabetic or has a high risk for diabetes, then the diagnsois needs to state this, as the patient
# Also, if a patient is a healthy normal patient, then the diagnosis section should state that instead.
```

```
df.duplicated(keep=False)
#remove duplicates
```

```
encounter_id
2278392      False
149190       False
64410        False
500364       False
16680        False
35754        False
55842        False
63768        False
12522        False
15738        False
28236        False
36900        False
40926        False
42570        False
62256        False
73578        False
77076        False
84222        False
89682        False
148530       False
150006       False
150048       False
```

```
182796      False
183930      False
216156      False
221634      False
236316      False
248916      False
250872      False
252822      False
253380      False
253722      False
260166      False
293058      False
293118      False
325848      False
325866      False
326028      False
358776      False
377268      False
383430      False
419304      False
421194      False
449142      False
450210      False
464994      False
486156      False
498030      False
537834      False
544194      False
550098      False
584136      False
590346      False
591996      False
604188      False
623880      False
630342      False
```

```
# Count duplicate on a column
df.patient_nbr.duplicated().sum()
```

```
30248
```

```
df.describe()
#the data are very wide ranging based on the mean, min, max values
```

	patient_nbr	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	num_lab_proc
count	1.017660e+05	101766.000000	101766.000000	101766.000000	101766.000000	101766
mean	5.433040e+07	2.024006	3.715642	5.754437	4.395987	43
std	3.869636e+07	1.445403	5.280166	4.064081	2.985108	19
min	1.350000e+02	1.000000	1.000000	1.000000	1.000000	1
25%	2.341322e+07	1.000000	1.000000	1.000000	2.000000	31
50%	4.550514e+07	1.000000	1.000000	7.000000	4.000000	44
75%	8.754595e+07	3.000000	4.000000	7.000000	6.000000	57
max	1.895026e+08	8.000000	28.000000	25.000000	14.000000	132

```
df.dtypes
```

```
patient_nbr                   int64
race                         object
gender                       object
age                          object
weight                       object
admission_type_id             int64
discharge_disposition_id      int64
admission_source_id           int64
time_in_hospital              int64
payer_code                   object
medical_specialty            object
num_lab_procedures            int64
num_procedures                int64
num_medications               int64
number_outpatient             int64
number_emergency              int64
```

```
number_inpatient             int64
diag_1                      object
diag_2                      object
diag_3                      object
number_diagnoses             int64
max_glu_serum               object
A1Cresult                   object
metformin                   object
repaglinide                 object
nateglinide                 object
chlorpropamide              object
glimepiride                 object
acetohexamide               object
glipizide                   object
glyburide                   object
tolbutamide                 object
pioglitazone                object
rosiglitazone               object
acarbose                    object
miglitol                    object
troglitazone                object
tolazamide                  object
examide                     object
citoglipton                 object
insulin                     object
glyburide-metformin         object
glipizide-metformin         object
glimepiride-pioglitazone    object
metformin-rosiglitazone     object
metformin-pioglitazone      object
change                      object
diabetesMed                 object
readmitted                  object
dtype: object
```

```
A1Ccounts = df.groupby(['A1Cresult'])['A1Cresult'].count()
A1Ccounts
#None values (84,748) are greatest. Most healthy people don't get A1C labs taken, indicating we likely have a very larg
#As our focus is on diabetes, the >7 and >8 A1C results are important indicators for diabetes.
#A1C results can also be calculated using maximum serum glucose.
```

```
#To rule out this possibility that many None values may be due to max glu serum values, we need to see: Can the A1C res
```

```
A1Cresult
>7       3812
>8       8216
None    84748
Norm     4990
Name: A1Cresult, dtype: int64
```

```
max_glu_serum_counts = df.groupby(['max_glu_serum'])['max_glu_serum'].count()
max_glu_serum_counts
#No, A1C results cannot be calculated using maximum glucose serum in this dataset. None values are greater than A1C cou
```

```
max_glu_serum
>200     1485
>300     1264
None    96420
Norm     2597
Name: max_glu_serum, dtype: int64
```

```
#The following are research questions based on the above initial inquiry:
```

In the US, patients with diabetes or high risk tend to get treated, and manage their symptoms. This data analysis is only a check that patients with suspected diabetes are treated, for oversight purposes and ensure patient safety.

Have all patients with clinically significant A1C values indicating diabetes received any management of their symptoms, based on the data? If no management, is physician consultation recommended, for possible confirm of any diabetes and treatment? Which patients are recommended for physician referral? (to Submit as a Dear Patient letter and or phone call.) We need to prioritize this review for patient safety reasons.

```
missing_value = ['?']
#remove the missing value
```

```
df = pd.read_csv('https://raw.githubusercontent.com/niteen11/DataAnalyticsAcademy/master/Python/dataset_diabetes/diabet
```

```
/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3326: DtypeWarning: Columns (10) have mixed
  exec(code_obj, self.user_global_ns, self.user_ns)
```

```
df.head()
```

	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admis
encounter_id								
2278392	8222157	Caucasian	Female	[0-10)	NaN	6	25	
149190	55629189	Caucasian	Female	[10-20)	NaN	1	1	
64410	86047875	AfricanAmerican	Female	[20-30)	NaN	1	1	
500364	82442376	Caucasian	Male	[30-40)	NaN	1	1	
16680	42519267	Caucasian	Male	[40-50)	NaN	1	1	

```
df = df.dropna()
#drop all rows with NaN
```

```
df.shape
```

```
(1043, 49)
```

```
df.columns
```

```
Index(['patient_nbr', 'race', 'gender', 'age', 'weight', 'admission_type_id',
       'discharge_disposition_id', 'admission_source_id', 'time_in_hospital',
       'payer_code', 'medical_specialty', 'num_lab_procedures',
       'num_procedures', 'num_medications', 'number_outpatient',
       'number_emergency', 'number_inpatient', 'diag_1', 'diag_2', 'diag_3',
       'number_diagnoses', 'max_glu_serum', 'A1Cresult', 'metformin',
       'repaglinide', 'nateglinide', 'chlorpropamide', 'glimepiride',
       'acetohexamide', 'glipizide', 'glyburide', 'tolbutamide',
       'pioglitazone', 'rosiglitazone', 'acarbose', 'miglitol', 'troglitazone',
       'tolazamide', 'examide', 'citoglipton', 'insulin',
       'glyburide-metformin', 'glipizide-metformin',
       'glimepiride-pioglitazone', 'metformin-rosiglitazone',
       'metformin-pioglitazone', 'change', 'diabetesMed', 'readmitted'],
      dtype='object')
```

```
#drop columns
df.drop('admission_type_id', axis = 1, inplace = True)
df.drop('discharge_disposition_id', axis = 1, inplace = True)
df.drop('admission_source_id', axis = 1, inplace = True)
df.drop('payer_code', axis = 1, inplace = True)
df.drop('medical_specialty', axis = 1, inplace = True)
df.drop('num_lab_procedures', axis = 1, inplace = True)
df.drop('num_procedures', axis = 1, inplace = True)
df.drop('num_medications', axis = 1, inplace = True)
df.drop('number_outpatient', axis = 1, inplace = True)
df.drop('number_emergency', axis = 1, inplace = True)
df.drop('number_inpatient', axis = 1, inplace = True)
df.drop('diag_1', axis = 1, inplace = True)
df.drop('diag_2', axis = 1, inplace = True)
df.drop('diag_3', axis = 1, inplace = True)

df
```

	patient_nbr	race	gender	age	weight	time_in_hospital	number_diagnoses	max_glu_seru
encounter_id								
88792836	100654011	Caucasian	Female	[70-80)	[75-100)	10	9	Nor

88792836	100654011	Caucasian	Female	[70-80)	[75-100)	10	9	Nor
88986678	58682736	Caucasian	Male	[80-90)	[50-75)	6	9	Nor
89032962	69250302	Caucasian	Male	[60-70)	[100-125)	2	9	Nor
89191392	62022042	Caucasian	Male	[40-50)	[75-100)	3	9	Nor
89277516	30950811	Caucasian	Male	[50-60)	[100-125)	2	7	Nor
89307582	58763808	Caucasian	Female	[70-80)	[75-100)	10	9	Nor
89343738	63813420	Caucasian	Male	[60-70)	[50-75)	3	9	Nor
89583948	84387969	Caucasian	Male	[50-60)	[100-125)	4	9	Nor
89583978	110949741	Caucasian	Male	[70-80)	[100-125)	2	7	Nor
89727588	49167621	Caucasian	Female	[70-80)	[75-100)	3	7	Nor
89776728	56356434	Caucasian	Female	[50-60)	[50-75)	1	9	Nor
89986632	109527102	Caucasian	Male	[40-50)	[125-150)	1	9	Nor
90093678	78098634	Caucasian	Female	[80-90)	[25-50)	2	9	Nor
90136908	21850101	Caucasian	Male	[50-60)	[75-100)	4	9	Nor
90234618	99090900	Caucasian	Male	[70-80)	[75-100)	5	6	Nor
90409224	79327116	Caucasian	Male	[40-50)	[75-100)	11	9	Nor
90443064	83177253	Caucasian	Male	[70-80)	[100-125)	2	9	Nor
90710628	106419474	Caucasian	Female	[70-80)	[75-100)	3	9	Nor
90739116	83232054	Caucasian	Female	[70-80)	[75-100)	1	6	Nor
90832170	108730161	Caucasian	Male	[80-90)	[50-75)	5	9	Nor
90850632	49469211	Caucasian	Female	[60-70)	[75-100)	8	9	Nor
90863208	77730093	Caucasian	Female	[80-90)	[75-100)	7	9	Nor
90884442	114086232	Caucasian	Female	[80-90)	[75-100)	14	9	Nor

90962598	84746214	Caucasian	Female	[70-80)	[50-75)	6	9	Nor
91048026	20875761	Caucasian	Female	[80-90)	[50-75)	3	9	Nor
91108776	38644884	Caucasian	Male	[50-60)	[75-100)	4	7	Nor
91153740	3749778	Caucasian	Female	[70-80)	[150-175)	8	9	Nor
91188102	63002484	Caucasian	Male	[60-70)	[100-125)	4	9	Nor
91234476	107551395	AfricanAmerican	Male	[50-60)	[75-100)	4	7	Nor

```
df.duplicated(keep=False)
# remove duplicates
```

```
encounter_id
88792836     False
88986678     False
89032962     False
89191392     False
89277516     False
89307582     False
89343738     False
89583948     False
89583978     False
89727588     False
89776728     False
89986632     False
90093678     False
90136908     False
90234618     False
90409224     False
90443064     False
90710628     False
90739116     False
90832170     False
90850632     False
90863208     False
90884442     False
90962598     False
91048026     False
91108776     False
```

```
91153740    False
91188102    False
91234476    False
91237050    False
91244268    False
91255104    False
91345014    False
91421286    False
91985298    False
92065956    False
92068782    False
92140044    False
92221062    False
92264376    False
92321928    False
92442756    False
92489178    False
92605908    False
92610858    False
93029880    False
93134874    False
93155916    False
93211218    False
93272010    False
93314040    False
93426900    False
93518082    False
93534636    False
93763092    False
94009398    False
94037142    False
```

```
df.patient_nbr.duplicated().sum()
#Count duplicate on a column
```

```
47
```

```
94919640    22855590    Caucasian    Male    [80-90)    [75-100)    11    4    Nor
```

```
df
```

	patient_nbr	race	gender	age	weight	time_in_hospital	number_diagnoses	max_glu_seru
encounter_id								
88792836	100654011	Caucasian	Female	[70-80)	[75-100)	10	9	Nor
88986678	58682736	Caucasian	Male	[80-90)	[50-75)	6	9	Nor
89032962	69250302	Caucasian	Male	[60-70)	[100-125)	2	9	Nor
89191392	62022042	Caucasian	Male	[40-50)	[75-100)	3	9	Nor
89277516	30950811	Caucasian	Male	[50-60)	[100-125)	2	7	Nor
89307582	58763808	Caucasian	Female	[70-80)	[75-100)	10	9	Nor
89343738	63813420	Caucasian	Male	[60-70)	[50-75)	3	9	Nor
89583948	84387969	Caucasian	Male	[50-60)	[100-125)	4	9	Nor
89583978	110949741	Caucasian	Male	[70-80)	[100-125)	2	7	Nor
89727588	49167621	Caucasian	Female	[70-80)	[75-100)	3	7	Nor
89776728	56356434	Caucasian	Female	[50-60)	[50-75)	1	9	Nor
89986632	109527102	Caucasian	Male	[40-50)	[125-150)	1	9	Nor
90093678	78098634	Caucasian	Female	[80-90)	[25-50)	2	9	Nor
90136908	21850101	Caucasian	Male	[50-60)	[75-100)	4	9	Nor
90234618	99090900	Caucasian	Male	[70-80)	[75-100)	5	6	Nor
90409224	79327116	Caucasian	Male	[40-50)	[75-100)	11	9	Nor
90443064	83177253	Caucasian	Male	[70-80)	[100-125)	2	9	Nor
90710628	106419474	Caucasian	Female	[70-80)	[75-100)	3	9	Nor
90739116	83232054	Caucasian	Female	[70-80)	[75-100)	1	6	Nor
90832170	108730161	Caucasian	Male	[80-90)	[50-75)	5	9	Nor
90850632	49469211	Caucasian	Female	[60-70)	[75-100)	8	9	Nor

90863208	77730093	Caucasian	Female	[80-90)	[75-100)	7	9	Nor
90884442	114086232	Caucasian	Female	[80-90)	[75-100)	14	9	Nor
90962598	84746214	Caucasian	Female	[70-80)	[50-75)	6	9	Nor
91048026	20875761	Caucasian	Female	[80-90)	[50-75)	3	9	Nor
91108776	38644884	Caucasian	Male	[50-60)	[75-100)	4	7	Nor
91153740	3749778	Caucasian	Female	[70-80)	[150-175)	8	9	Nor
91188102	63002484	Caucasian	Male	[60-70)	[100-125)	4	9	Nor
91234476	107551395	AfricanAmerican	Male	[50-60)	[75-100)	4	7	Nor
91237950	101722140	Caucasian	Male	[50-60)	[150-175)	6	9	Nor

```
df.index = pd.to_numeric(df.index, errors='coerce')
#add a regular index
```

91255104	99154062	Caucasian	Female	[80-90)	[75-100)	3	8	Nor

```
df.sort_index()
```

	patient_nbr	race	gender	age	weight	time_in_hospital	number_diagnoses	max_glu_seru
encounter_id								
88792836	100654011	Caucasian	Female	[70-80)	[75-100)	10	9	Nor
88986678	58682736	Caucasian	Male	[80-90)	[50-75)	6	9	Nor
89032962	69250302	Caucasian	Male	[60-70)	[100-125)	2	9	Nor
89191392	62022042	Caucasian	Male	[40-50)	[75-100)	3	9	Nor
89277516	30950811	Caucasian	Male	[50-60)	[100-125)	2	7	Nor
89307582	58763808	Caucasian	Female	[70-80)	[75-100)	10	9	Nor
89343738	63813420	Caucasian	Male	[60-70)	[50-75)	3	9	Nor
89583948	84387969	Caucasian	Male	[50-60)	[100-125)	4	9	Nor

```
89583978   110949741        Caucasian    Male  [70-80)  [100-125)     2     7  Nor
89727588    49167621        Caucasian  Female  [70-80)   [75-100)     3     7  Nor
89776728    56356434        Caucasian  Female  [50-60)    [50-75)     1     9  Nor
89986632   109527102        Caucasian    Male  [40-50)  [125-150)     1     9  Nor
90093678    78098634        Caucasian  Female  [80-90)    [25-50)     2     9  Nor
90136908    21850101        Caucasian    Male  [50-60)   [75-100)     4     9  Nor
90234618    99090900        Caucasian    Male  [70-80)   [75-100)     5     6  Nor
90409224    79327116        Caucasian    Male  [40-50)   [75-100)    11     9  Nor
90443064    83177253        Caucasian    Male  [70-80)  [100-125)     2     9  Nor
90710628   106419474        Caucasian  Female  [70-80)   [75-100)     3     9  Nor
90739116    83232054        Caucasian  Female  [70-80)   [75-100)     1     6  Nor
90832170   108730161        Caucasian    Male  [80-90)    [50-75)     5     9  Nor
90850632    49469211        Caucasian  Female  [60-70)   [75-100)     8     9  Nor
90863208    77730093        Caucasian  Female  [80-90)   [75-100)     7     9  Nor
90884442   114086232        Caucasian  Female  [80-90)   [75-100)    14     9  Nor
90962598    84746214        Caucasian  Female  [70-80)    [50-75)     6     9  Nor
91048026    20875761        Caucasian  Female  [80-90)    [50-75)     3     9  Nor
91108776    38644884        Caucasian    Male  [50-60)   [75-100)     4     7  Nor
91153740     3749778        Caucasian  Female  [70-80)  [150-175)     8     9  Nor
91188102    63002484        Caucasian    Male  [60-70)  [100-125)     4     9  Nor
91234476   107551395  AfricanAmerican    Male  [50-60)   [75-100)     4     7  Nor
91237050   101722149        Caucasian    Male  [50-60)  [150-175)     6     9  Nor
```

```
df.dtypes
```

```
patient_nbr                  int64
race                        object
gender                      object
age                         object
weight                      object
time_in_hospital             int64
number_diagnoses             int64
max_glu_serum               object
A1Cresult                   object
metformin                   object
repaglinide                 object
nateglinide                 object
chlorpropamide              object
glimepiride                 object
acetohexamide               object
glipizide                   object
glyburide                   object
tolbutamide                 object
pioglitazone                object
rosiglitazone               object
acarbose                    object
miglitol                    object
troglitazone                object
tolazamide                  object
examide                     object
citoglipton                 object
insulin                     object
glyburide-metformin         object
glipizide-metformin         object
glimepiride-pioglitazone    object
metformin-rosiglitazone     object
metformin-pioglitazone      object
change                      object
diabetesMed                 object
readmitted                  object
dtype: object
```

93426900	104373990	Caucasian	Female	[70-80)	[75-100)	7	9	Nor

```
df.groupby(['A1Cresult'])['patient_nbr'].count().sort_values(ascending=False)
```

```
A1Cresult
```

```
None    935
>8       57
>7       26
Norm     25
Name: patient_nbr, dtype: int64
```

```
df.index
```

```
Int64Index([ 88792836,  88986678,  89032962,  89191392,  89277516,  89307582,
             89343738,  89583948,  89583978,  89727588,
            ...
            433101896, 433130288, 433719680, 434868986, 436191524, 436721264,
            437928656, 437959022, 439577312, 439606454],
           dtype='int64', name='encounter_id', length=1043)
```

94686084 46422936 Caucasian Male [80-90) [75-100) 1 5 Nor

```
df.reset_index(inplace=True)
df = df.rename(columns = {'index':'Indexed_encounter_id'})
```

94749948 80348409 Caucasian Female [60-70) [100-125) 1 9 Nor

```
df.dtypes
#now encounter_id index is in the dataframe
```

```
encounter_id                 int64
patient_nbr                  int64
race                        object
gender                      object
age                         object
weight                      object
time_in_hospital             int64
number_diagnoses             int64
max_glu_serum               object
A1Cresult                   object
metformin                   object
repaglinide                 object
nateglinide                 object
chlorpropamide              object
glimepiride                 object
acetohexamide               object
glipizide                   object
glyburide                   object
```

```
    glyburide                  object
    tolbutamide                object
    pioglitazone               object
    rosiglitazone              object
    acarbose                   object
    miglitol                   object
    troglitazone               object
    tolazamide                 object
    examide                    object
    citoglipton                object
    insulin                    object
    glyburide-metformin        object
    glipizide-metformin        object
    glimepiride-pioglitazone   object
    metformin-rosiglitazone    object
    metformin-pioglitazone     object
    change                     object
    diabetesMed                object
    readmitted                 object
    dtype: object
```

```
df.groupby(['A1Cresult'])['encounter_id'].count().sort_values(ascending=False)
```

```
    A1Cresult
    None    935
    >8       57
    >7       26
    Norm     25
    Name: encounter_id, dtype: int64
```

99207780 66399966 Caucasian Female [70-80) [100-125) 8 9 Nor

```
df2 = df[(df['A1Cresult'] != 'None')]
#only actual results included. The None values will not appear.
```

99402426 98244774 Caucasian Female [70-80) [75-100) 5 9 Nor

```
A1C_resultsyes = df2.groupby(['A1Cresult'])['A1Cresult'].count()
A1C_resultsyes
```

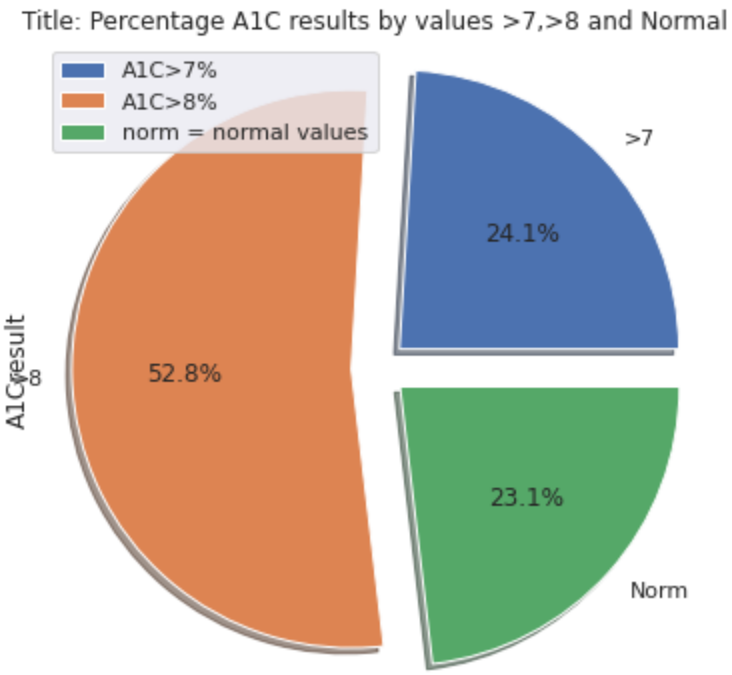
```
    A1Cresult
    >7      26
    >8      57
```

```
Norm    25
Name: A1Cresult, dtype: int64
```

100009428 28219392 Caucasian Female [60-70) [100-125) 3 9 Nor

```
#A1C results are graphed, indicating over half of actual A1C results are A1C >8 % (57, 52.8%), followed by A1C>7% (26,
#combining >8 and >7,  over 75% of actual A1C results are clinically significant.
A1C_resultsyes.plot.pie(figsize = (6, 6), shadow = True, explode = (0.1, 0.1, 0.1), autopct = '%1.1f%%')
labels = ['A1C>7%', 'A1C>8%', 'norm = normal values']
plt.title('Title: Percentage A1C results by values >7,>8 and Normal')
plt.legend(labels, loc = "best")
plt.axis('equal')

plt.show()
```



101705652 21936186 Caucasian Female [70-80) [50-75) 3 9 Nor

over 75% of actual A1C results that are clinically significant.

101870766 110854314 Caucasian Female [80-90) [75-100) 2 9 Nor

df code list

df code list

df - A1C result is none, normal, >7, or >8

df2 - A1C result is normal, >7, or >8

df3 - A1C result is >7 or >8

```
df3 = df2[(df2['A1Cresult'] != 'Norm')]
df3
#focus on >7% and >8% values only, which are the clinically significant A1C values according to the CDC and ADA researc
#there are 83 patients
```

	encounter_id	patient_nbr	race	gender	age	weight	time_in_hospital	number_diagnoses	max_gl
565	161410230	45004320	Caucasian	Female	[50-60)	[50-75)	3	9	
571	162997158	110587005	Other	Male	[0-10)	[0-25)	4	3	
584	165270666	76346199	Caucasian	Male	[60-70)	[75-100)	6	7	
587	166247610	110907864	Caucasian	Male	[30-40)	[50-75)	10	9	
595	166784748	65011347	Caucasian	Male	[60-70)	[125-150)	8	9	
596	166898568	107235018	Caucasian	Male	[60-70)	[50-75)	3	9	
598	167360376	102491856	Caucasian	Female	[80-90)	[50-75)	2	9	
600	167955030	49850208	AfricanAmerican	Male	[30-40)	[125-150)	2	6	
602	168258654	111481452	Caucasian	Female	[70-80)	[50-75)	4	8	
606	168409050	109214118	Caucasian	Female	[60-70)	[75-100)	8	9	
609	168894804	95580027	Caucasian	Male	[80-90)	[75-100)	2	9	
616	170057718	11282175	Caucasian	Female	[40-50)	[125-150)	13	9	
617	170477718	6428826	Caucasian	Male	[60-70)	[25-50)	2	8	

624	171201534	104752071	Caucasian	Female	[30-40)	[75-100)	5	9
641	175554114	92295351	Caucasian	Male	[60-70)	[75-100)	6	7
643	176209518	11620728	Caucasian	Female	[60-70)	[50-75)	4	8
653	178883262	82967229	Caucasian	Male	[30-40)	[75-100)	5	9
657	179182146	81229968	Other	Female	[50-60)	[125-150)	5	9
662	181247406	19609686	Caucasian	Male	[40-50)	[75-100)	3	3
666	182852886	82184103	Caucasian	Female	[90-100)	[50-75)	4	9
671	184058778	65349441	Caucasian	Female	[70-80)	[50-75)	3	6
690	190585200	73878543	Caucasian	Female	[40-50)	[50-75)	3	5
698	191691198	72389961	Caucasian	Female	[50-60)	[75-100)	7	9
701	192208788	113243436	Caucasian	Male	[70-80)	[75-100)	4	9
704	192835434	62058582	Caucasian	Male	[50-60)	[100-125)	3	9
715	204459354	69172002	AfricanAmerican	Female	[40-50)	[100-125)	3	9
724	208829172	84178620	Caucasian	Male	[70-80)	[50-75)	4	6
726	209406216	101132973	Caucasian	Female	[50-60)	[125-150)	4	8
733	210628770	73122228	Caucasian	Male	[60-70)	[100-125)	10	9
736	211386834	92434806	Caucasian	Female	[20-30)	[50-75)	4	9
752	216655782	92960091	Caucasian	Male	[30-40)	[100-125)	4	9

```
A1C_thresholdvals = df3.groupby(['A1Cresult', 'gender'])['gender'].count()
A1C_thresholdvals
#The number of males and females is about the same, for patients with A1C >7 and >8 A1C results.
```

```
A1Cresult  gender
>7         Female    12
           Male      14
>8         Female    30
           Male      27
```

```
          Male      27
Name: gender, dtype: int64
```

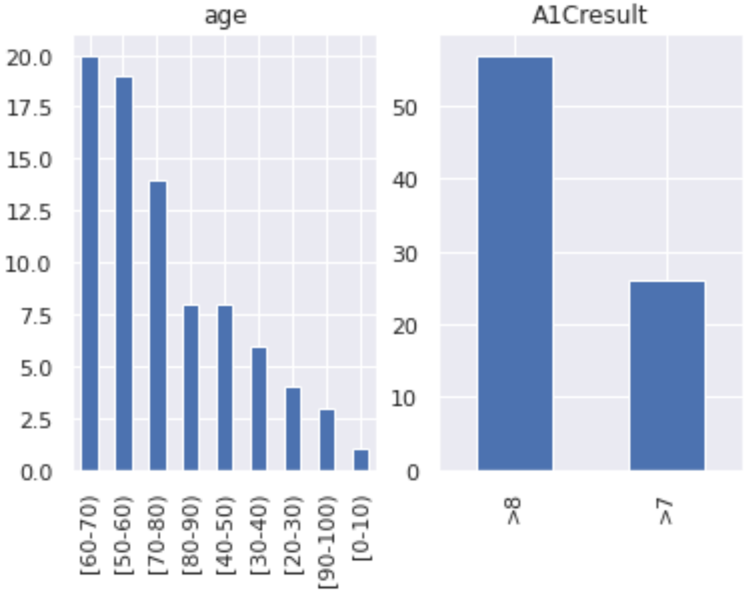
```
A1C_thresholdvals = df3.groupby(['A1Cresult', 'age'])['age'].count()
A1C_thresholdvals
#the age distribution trending for A1C results is expected based on known diabetes and aging research. With increasing
```

```
A1Cresult  age     
>7         [30-40)      2
           [40-50)      3
           [50-60)      4
           [60-70)     10
           [70-80)      7
>8         [0-10)       1
           [20-30)      4
           [30-40)      4
           [40-50)      5
           [50-60)     15
           [60-70)     10
           [70-80)      7
           [80-90)      8
           [90-100)     3
Name: age, dtype: int64
```

```
df3['age'].value_counts()
#totals are combined for patients with A1Cresult >7 and >8
#there are no clinically significant A1C patients in the 10 to 20 range.  What is concerning is 1 child in the 0-10 yea
```

```
[60-70)     20
[50-60)     19
[70-80)     14
[80-90)      8
[40-50)      8
[30-40)      6
[20-30)      4
[90-100)     3
[0-10)       1
Name: age, dtype: int64
```

```
categorical_features = ["age", "A1Cresult"]
fig, ax = plt.subplots(1, len(categorical_features))
for i, categorical_feature in enumerate(df3[categorical_features]):
    df3[categorical_feature].value_counts().plot(kind = "bar", ax=ax[i]).set_title(categorical_feature)
fig.show()
#The age groups with most clinically significant A1C values (A1C >7 or >8).  20 are [60-70) years age, 10 are [50-60) y
```



```
df3.describe()
# The maximum days for this subgroup in hospital is 14 days, and maximum number diagnoses is 9, suggesting need for mar
```

	encounter_id	patient_nbr	time_in_hospital	number_diagnoses
count	8.300000e+01	8.300000e+01	83.000000	83.000000
mean	2.593809e+08	7.654217e+07	5.120482	8.168675
std	8.363129e+07	3.447691e+07	3.033945	1.429816
min	1.614102e+08	6.428826e+06	1.000000	3.000000
25%	1.873220e+08	5.831086e+07	3.000000	8.000000

50%	2.299112e+08	8.296723e+07	4.000000	9.000000
75%	3.048641e+08	1.027450e+08	7.000000	9.000000
max	4.331019e+08	1.824088e+08	14.000000	9.000000

```
pd.set_option('display.max_columns', None)
#display all columns
```

```
pd.set_option("max_rows", None)
#display all rows
```

```
     113178870       98886357       Caucasian      Male    [60-70)    [50-75)                    3                  9          Nor
```

```
df3
```

	encounter_id	patient_nbr	race	gender	age	weight	time_in_hospital	number_diagnoses	max_gl
565	161410230	45004320	Caucasian	Female	[50-60)	[50-75)	3	9	
571	162997158	110587005	Other	Male	[0-10)	[0-25)	4	3	
584	165270666	76346199	Caucasian	Male	[60-70)	[75-100)	6	7	
587	166247610	110907864	Caucasian	Male	[30-40)	[50-75)	10	9	
595	166784748	65011347	Caucasian	Male	[60-70)	[125-150)	8	9	
596	166898568	107235018	Caucasian	Male	[60-70)	[50-75)	3	9	
598	167360376	102491856	Caucasian	Female	[80-90)	[50-75)	2	9	
600	167955030	49850208	AfricanAmerican	Male	[30-40)	[125-150)	2	6	
602	168258654	111481452	Caucasian	Female	[70-80)	[50-75)	4	8	
606	168409050	109214118	Caucasian	Female	[60-70)	[75-100)	8	9	
609	168894804	95580027	Caucasian	Male	[80-90)	[75-100)	2	9	
616	170057718	11282175	Caucasian	Female	[40-50)	[125-150)	13	9	

617	170477718	6428826	Caucasian	Male	[60-70)	[25-50)	2	8
624	171201534	104752071	Caucasian	Female	[30-40)	[75-100)	5	9
641	175554114	92295351	Caucasian	Male	[60-70)	[75-100)	6	7
643	176209518	11620728	Caucasian	Female	[60-70)	[50-75)	4	8
653	178883262	82967229	Caucasian	Male	[30-40)	[75-100)	5	9
657	179182146	81229968	Other	Female	[50-60)	[125-150)	5	9
662	181247406	19609686	Caucasian	Male	[40-50)	[75-100)	3	3
666	182852886	82184103	Caucasian	Female	[90-100)	[50-75)	4	9
671	184058778	65349441	Caucasian	Female	[70-80)	[50-75)	3	6
690	190585200	73878543	Caucasian	Female	[40-50)	[50-75)	3	5
698	191691198	72389961	Caucasian	Female	[50-60)	[75-100)	7	9
701	192208788	113243436	Caucasian	Male	[70-80)	[75-100)	4	9
704	192835434	62058582	Caucasian	Male	[50-60)	[100-125)	3	9
715	204459354	69172002	AfricanAmerican	Female	[40-50)	[100-125)	3	9
724	208829172	84178620	Caucasian	Male	[70-80)	[50-75)	4	6
726	209406216	101132973	Caucasian	Female	[50-60)	[125-150)	4	8
733	210628770	73122228	Caucasian	Male	[60-70)	[100-125)	10	9
736	211386834	92434806	Caucasian	Female	[20-30)	[50-75)	4	9
752	216655782	92960991	Caucasian	Male	[30-40)	[100-125)	4	9

```
diabetesMed_counts = df3.groupby(['diabetesMed'])['diabetesMed'].count()
diabetesMed_counts
#that is good news.  Of the 83 patients with clinically significant A1C values, only 10 patients appear to have taken r
#Second, let's also confirm that of those "Yes" = diabetesMed, we actually have recorded medication of some kind. If nc
```

```
diabetesMed
No      10
```

```
Yes    73
Name: diabetesMed, dtype: int64
```

```
 776      227017800      80895078        Caucasian     Male    [70-80)    [75-100)                  5                  9
```

```
df4 = df3.loc[(df3['diabetesMed'] == 'No')]
df4
```

	encounter_id	patient_nbr	race	gender	age	weight	time_in_hospital	number_diagnoses	max_glu
571	162997158	110587005	Other	Male	[0-10)	[0-25)	4	3	
598	167360376	102491856	Caucasian	Female	[80-90)	[50-75)	2	9	
671	184058778	65349441	Caucasian	Female	[70-80)	[50-75)	3	6	
704	192835434	62058582	Caucasian	Male	[50-60)	[100-125)	3	9	
715	204459354	69172002	AfricanAmerican	Female	[40-50)	[100-125)	3	9	
733	210628770	73122228	Caucasian	Male	[60-70)	[100-125)	10	9	
780	227815044	60401475	Caucasian	Male	[60-70)	[125-150)	4	8	
817	241498266	21148866	Caucasian	Male	[70-80)	[100-125)	2	8	
967	353773754	104956029	Caucasian	Male	[40-50)	[75-100)	4	7	
1021	419597300	82133919	Caucasian	Female	[60-70)	[100-125)	4	8	

```
pd.set_option('display.max_columns', None)
#display all columns
```

```
df4
```

```
        encounter_id  patient_nbr            race  gender      age    weight  time_in_hospital  number_diagnoses  max_glu
```

	encounter_id	patient_nbr	race	gender	age	weight	time_in_hospital	number_diagnoses	max_gl
571	162997158	110587005	Other	Male	[0-10)	[0-25)	4	3	
598	167360376	102491856	Caucasian	Female	[80-90)	[50-75)	2	9	
671	184058778	65349441	Caucasian	Female	[70-80)	[50-75)	3	6	
704	192835434	62058582	Caucasian	Male	[50-60)	[100-125)	3	9	
715	204459354	69172002	AfricanAmerican	Female	[40-50)	[100-125)	3	9	
733	210628770	73122228	Caucasian	Male	[60-70)	[100-125)	10	9	
780	227815044	60401475	Caucasian	Male	[60-70)	[125-150)	4	8	
817	241498266	21148866	Caucasian	Male	[70-80)	[100-125)	2	8	
967	353773754	104956029	Caucasian	Male	[40-50)	[75-100)	4	7	
1021	419597300	82133919	Caucasian	Female	[60-70)	[100-125)	4	8	

```
df4.columns
```

```
Index(['encounter_id', 'patient_nbr', 'race', 'gender', 'age', 'weight',
       'time_in_hospital', 'number_diagnoses', 'max_glu_serum', 'A1Cresult',
       'metformin', 'repaglinide', 'nateglinide', 'chlorpropamide',
       'glimepiride', 'acetohexamide', 'glipizide', 'glyburide', 'tolbutamide',
       'pioglitazone', 'rosiglitazone', 'acarbose', 'miglitol', 'troglitazone',
       'tolazamide', 'examide', 'citoglipton', 'insulin',
       'glyburide-metformin', 'glipizide-metformin',
       'glimepiride-pioglitazone', 'metformin-rosiglitazone',
       'metformin-pioglitazone', 'change', 'diabetesMed', 'readmitted'],
      dtype='object')
```

1017 417898598 100378215 Caucasian Female [60-70) [75-100) 6 9

```
df4[['metformin', 'repaglinide', 'nateglinide', 'chlorpropamide', 'glimepiride', 'acetohexamide', 'glipizide', 'glyburi
    'troglitazone', 'tolazamide', 'examide', 'citoglipton', 'insulin', 'glyburide-metformin', 'glipizide-metformin', '
```

```
     'metformin-pioglitazone']].eq('Yes').all(1).sum()
#All the medication columns have data of Yes or No in them. Are there any columns with Yes?
#The results are 0, meaning All columns are No.  That is, none of the 10 patients are taking medications for their diab
```

```
0
```

```
#Of the 10 patients, which one needs to be contacted immediately and given a phone call, besides the pediatric patient?
#The patient with time_in_hospital = 10 days has the greatest # days in hospital, with 9 diagnoses, and is in the upper
#Patient_nbr 73122228 in row 733.
```

```
121694922     50444955     Caucasian     Male     [60-70)     [75-100)                    4                    9          Nor
```

```
patient_nbr_list = df4['patient_nbr'].tolist()
patient_nbr_list
```

```
[110587005,
 102491856,
 65349441,
 62058582,
 69172002,
 73122228,
 60401475,
 21148866,
 104956029,
 82133919]
```

This request is based on a data analysis, and is only an internal check to confirm our records are current and accurate for oversight purposes, and to help ensure patient safety which is a priority for our organization.

The data check answers the following questions:

-- Have all patients with clinically significant A1C values (A1C >7% or >8%) indicating diabetes received any management of their symptoms, based on the data? Answer: 83 patients have clinically significant A1C values. Of the 83, 73 pts appear to be managing their symptoms based on the data, which is good news. Of the 83, 10 pts (10%) have symptoms but no management documented.

-- If no management was noted in the data, is recommending physician consultation needed, for possible confirm of diabetes diagnosis and treatment? Answer: Yes, some recommendations are for physician referral on possible diabetes and treatment. (Please Submit as a Dear Patient letter and or phone call.) We are prioritizing this list for patient safety reasons.

Recommend physician referral contact for 10 patients. Reconfirm A1C values for possible diabetes, recommend retest A1C to verify, and provide treatment options.

10 Patient numbers are:

1. 110587005, <--pediatric patient, priority to contact.
2. 73122228, <--adult patient, priority to contact.
3. 102491856,
4. 65349441,
5. 62058582,
6. 69172002,
7. 60401475,
8. 21148866,
9. 104956029,
10. 82133919.

Once contact has been made, patient consents to diagnosis, and diabetes and treatment are confirmed, please provide diabetes lab values and medication updates for database. Thank you.

```
      125684196    39866616        Caucasian  Female  [60-70)   [75-100)       5       8    Nor
 1
      125980998    58805109        Caucasian  Female  [80-90)    [50-75)       9       9    Nor
      126071286    21749949        Caucasian    Male  [80-90)    [50-75)       2       9    Nor
      126079608     3021966  AfricanAmerican  Female  [40-50)  [175-200)       8       9    Nor
      126178116    74202138        Caucasian  Female  [50-60)  [100-125)       4       6    Nor
      126379464    27820854        Caucasian  Female  [60-70)    [50-75)       2       9    Nor
      126636348    79519941        Caucasian  Female  [50-60)    [25-50)       2       9    Nor
      126784050   108457488        Caucasian    Male  [40-50)  [150-175)       7       9    Nor
      126801642   114113106        Caucasian    Male  [70-80)   [75-100)       2       9    Nor
      126898770   113164263        Caucasian    Male  [70-80)    [50-75)       8       9    Nor
      126916110    44199522        Caucasian  Female  [80-90)    [50-75)       2       9    Nor
      127019382    62724330        Caucasian  Female  [50-60)  [100-125)       3       7    Nor
      127237116    64593639        Caucasian  Female  [70-80)   [75-100)       4       9    Nor
      127331478    10706778        Caucasian  Female  [70-80)  [100-125)       3       7    Nor
      127534788    50200974        Caucasian    Male  [40-50)  [100-125)       3       9    Nor
      127658238    45170577        Caucasian  Female  [60-70)    [50-75)       3       9    Nor
```