Data cleaning

```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

```
import os
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
from sklearn import linear_model, preprocessing
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn import feature_selection, metrics
from pandas.plotting import scatter_matrix
from seaborn import pairplot
from sklearn import feature_selection, metrics
from sklearn.preprocessing import StandardScaler
```

```
os.chdir('/content/drive/MyDrive/big_data_pred/cw/')

dbdata = pd.read_csv('diabetic_data.csv')
```

```
#shape of the daibetes dataframe
dbdata.shape
#this implies that the data has 101,766 rows (observations) and 50 columns (attributes)
```

```
(101766, 50)
```

```
dbdata.head()
```

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time
0	2278392	8222157	Caucasian	Female	[0-10)	?	6	25	1	
1	149190	55629189	Caucasian	Female	[10-20)	?	1	1	7	
2	64410	86047875	AfricanAmerican	Female	[20-30)	?	1	1	7	
3	500364	82442376	Caucasian	Male	[30-40)	?	1	1	7	
4	16680	42519267	Caucasian	Male	[40-50)	?	1	1	7	

```
dbdata.info()
#some of the columns which are meant to be categorical are numerical in the data
#hence we need to convert them to categorical
```

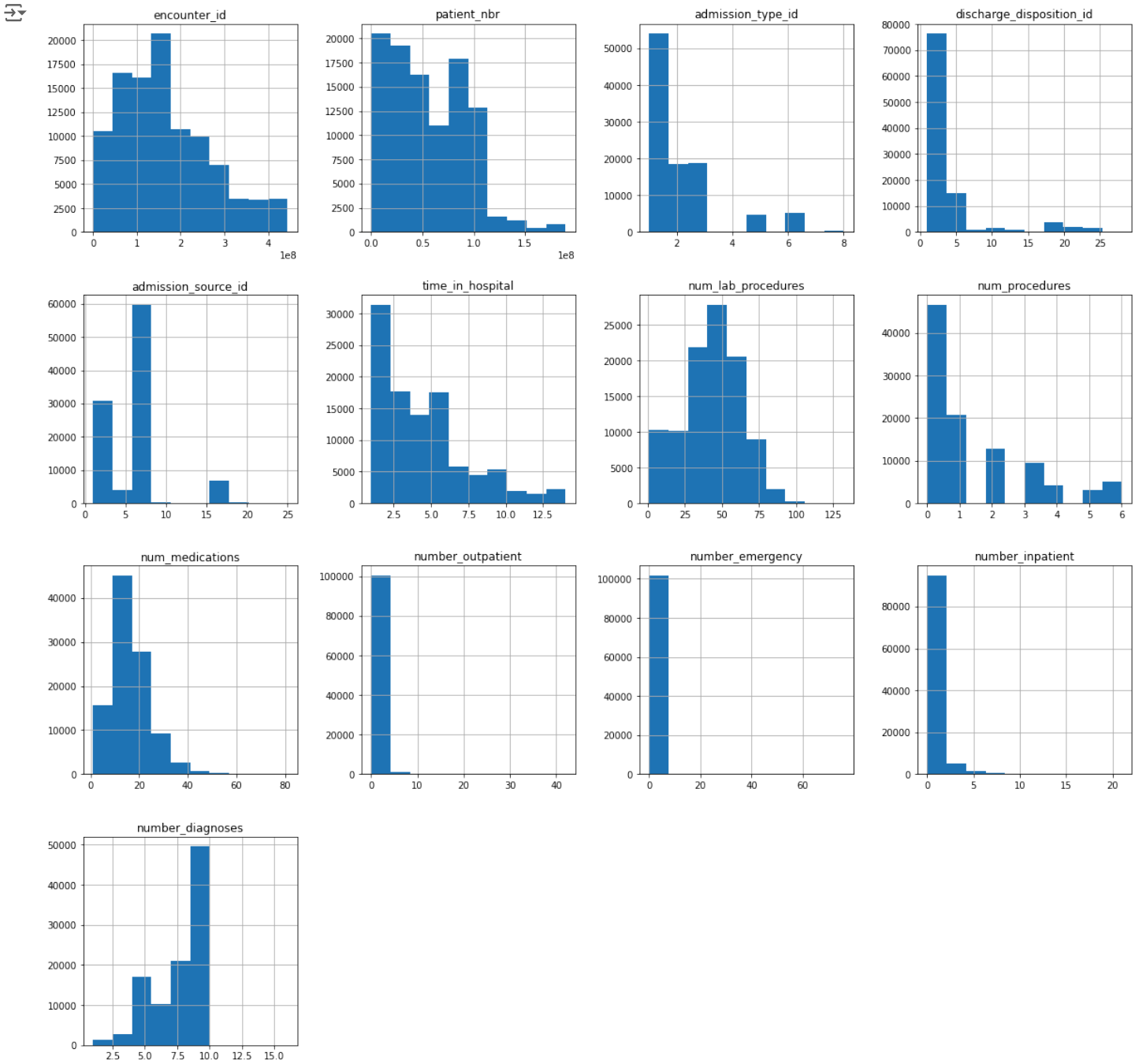
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 50 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   encounter_id              101766 non-null  int64
 1   patient_nbr               101766 non-null  int64
 2   race                      101766 non-null  object
 3   gender                    101766 non-null  object
 4   age                       101766 non-null  object
 5   weight                    101766 non-null  object
 6   admission_type_id         101766 non-null  int64
 7   discharge_disposition_id  101766 non-null  int64
 8   admission_source_id       101766 non-null  int64
 9   time_in_hospital          101766 non-null  int64
 10  payer_code                101766 non-null  object
 11  medical_specialty         101766 non-null  object
```

```
 12  num_lab_procedures        101766 non-null  int64
 13  num_procedures            101766 non-null  int64
 14  num_medications           101766 non-null  int64
 15  number_outpatient         101766 non-null  int64
 16  number_emergency          101766 non-null  int64
 17  number_inpatient          101766 non-null  int64
 18  diag_1                    101766 non-null  object
 19  diag_2                    101766 non-null  object
 20  diag_3                    101766 non-null  object
 21  number_diagnoses          101766 non-null  int64
 22  max_glu_serum             101766 non-null  object
 23  A1Cresult                 101766 non-null  object
 24  metformin                 101766 non-null  object
 25  repaglinide               101766 non-null  object
 26  nateglinide               101766 non-null  object
 27  chlorpropamide            101766 non-null  object
 28  glimepiride               101766 non-null  object
 29  acetohexamide             101766 non-null  object
 30  glipizide                 101766 non-null  object
 31  glyburide                 101766 non-null  object
 32  tolbutamide               101766 non-null  object
 33  pioglitazone              101766 non-null  object
 34  rosiglitazone             101766 non-null  object
 35  acarbose                  101766 non-null  object
 36  miglitol                  101766 non-null  object
 37  troglitazone              101766 non-null  object
 38  tolazamide                101766 non-null  object
 39  examide                   101766 non-null  object
 40  citoglipton               101766 non-null  object
 41  insulin                   101766 non-null  object
 42  glyburide-metformin       101766 non-null  object
 43  glipizide-metformin       101766 non-null  object
 44  glimepiride-pioglitazone  101766 non-null  object
 45  metformin-rosiglitazone   101766 non-null  object
 46  metformin-pioglitazone    101766 non-null  object
 47  change                    101766 non-null  object
 48  diabetesMed               101766 non-null  object
 49  readmitted                101766 non-null  object
dtypes: int64(13), object(37)
memory usage: 38.8+ MB
```

```
dbdata.describe()
```

	encounter_id	patient_nbr	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	num_lab_procedures
count	1.017660e+05	1.017660e+05	101766.000000	101766.000000	101766.000000	101766.000000	101766.000000
mean	1.652016e+08	5.433040e+07	2.024006	3.715642	5.754437	4.395987	43.095641
std	1.026403e+08	3.869636e+07	1.445403	5.280166	4.064081	2.985108	19.674362
min	1.252200e+04	1.350000e+02	1.000000	1.000000	1.000000	1.000000	1.000000
25%	8.496119e+07	2.341322e+07	1.000000	1.000000	1.000000	2.000000	31.000000
50%	1.523890e+08	4.550514e+07	1.000000	1.000000	7.000000	4.000000	44.000000
75%	2.302709e+08	8.754595e+07	3.000000	4.000000	7.000000	6.000000	57.000000
max	4.438672e+08	1.895026e+08	8.000000	28.000000	25.000000	14.000000	132.000000

```
dbdata.hist(figsize = (20,20))
plt.show()
```




```
dbdata.isnull().sum()
#while there are several missing data in the dataframe, isnull() function shows that there is non.
#this is becuase the null data are not represented by na or nan, but rather by space and ?
#to replace the null values with na so that we can track them i'll use the regex functions
```

```
encounter_id                0
patient_nbr                 0
race                        0
gender                      0
age                         0
weight                      0
admission_type_id           0
discharge_disposition_id    0
admission_source_id         0
time_in_hospital            0
payer_code                  0
```

```
medical_specialty           0
num_lab_procedures          0
num_procedures              0
num_medications             0
number_outpatient           0
number_emergency            0
number_inpatient            0
diag_1                      0
diag_2                      0
diag_3                      0
number_diagnoses            0
max_glu_serum               0
A1Cresult                   0
metformin                   0
repaglinide                 0
nateglinide                 0
chlorpropamide              0
glimepiride                 0
acetohexamide               0
glipizide                   0
glyburide                   0
tolbutamide                 0
pioglitazone                0
rosiglitazone               0
acarbose                    0
miglitol                    0
troglitazone                0
tolazamide                  0
examide                     0
citoglipton                 0
insulin                     0
glyburide-metformin         0
glipizide-metformin         0
glimepiride-pioglitazone    0
metformin-rosiglitazone     0
metformin-pioglitazone      0
change                      0
diabetesMed                 0
readmitted                  0
dtype: int64
```

```
dbdata = dbdata.replace('?', np.nan) # replace ? with nan
dbdata = dbdata.replace('^\s+', np.nan, regex=True) # replace empty spaces with nan
print(dbdata.isna().sum())
print(dbdata.shape[0])
```

```
encounter_id                    0
patient_nbr                     0
race                         2273
gender                          0
age                             0
weight                      98569
admission_type_id               0
discharge_disposition_id        0
admission_source_id             0
time_in_hospital                0
payer_code                  40256
medical_specialty           49949
num_lab_procedures              0
num_procedures                  0
num_medications                 0
number_outpatient               0
number_emergency                0
number_inpatient                0
diag_1                         21
diag_2                        358
diag_3                       1423
number_diagnoses                0
max_glu_serum                   0
A1Cresult                       0
metformin                       0
repaglinide                     0
nateglinide                     0
chlorpropamide                  0
glimepiride                     0
acetohexamide                   0
glipizide                       0
glyburide                       0
tolbutamide                     0
pioglitazone                    0
rosiglitazone                   0
acarbose                        0
miglitol                        0
```

```
troglitazone                   0
tolazamide                     0
examide                        0
citoglipton                    0
insulin                        0
glyburide-metformin            0
glipizide-metformin            0
glimepiride-pioglitazone       0
metformin-rosiglitazone        0
metformin-pioglitazone         0
change                         0
diabetesMed                    0
readmitted                     0
dtype: int64
101766
```

```
#Drop column with more than 50% missing values
dbdata.dropna(thresh=len(dbdata.index)/2, axis=1, inplace=True)
dbdata.shape
```

```
(101766, 49)
```

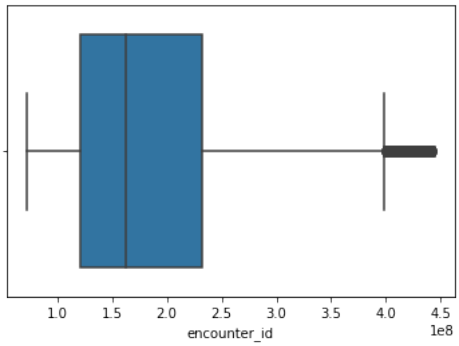
```
# def remove(df):
#   for x in df.columns:
#     f = df[x].value_counts()/df.shape[0]
#     if f.to_frame().iloc[:, 0].max() > 0.95:
#       df.drop(x, axis = 1, inplace = True)


# remove(dbdata)
# dbdata.shape


for x in dbdata.columns:
    f = dbdata[x].value_counts()/dbdata.shape[0]
    if f.to_frame().iloc[:, 0].max() >= 0.95:
      dbdata.drop(x, axis = 1, inplace = True)
print(dbdata.shape)
```

```
(101766, 33)
```

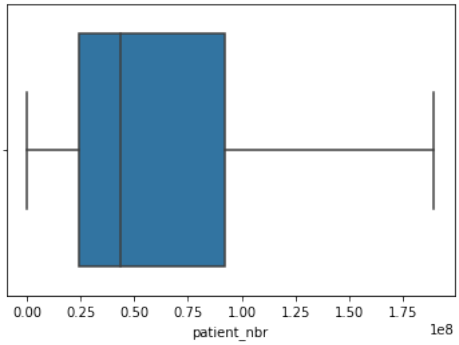
```
dbdata['age'].value_counts()


def age(df):
  for i in range(df.shape[0]):
    if(df.loc[i,'age']=='[70-80)'):
        df.loc[i,'age']=75
    elif(df.loc[i,'age']=='[60-70)'):
        df.loc[i,'age']=65
    elif(df.loc[i,'age']=='[50-60)'):
        df.loc[i,'age']=55
    elif(df.loc[i,'age']=='[80-90)'):
        df.loc[i,'age']=85
    elif(df.loc[i,'age']=='[40-50)'):
        df.loc[i,'age']=45
    elif(df.loc[i,'age']=='[30-40)'):
        df.loc[i,'age']=35
    elif(df.loc[i,'age']=='[90-100)'):
        df.loc[i,'age']=95
    elif(df.loc[i,'age']=='[20-30)'):
        df.loc[i,'age']=25
    elif(df.loc[i,'age']=='[10-20)'):
        df.loc[i,'age']=15
    elif(df.loc[i,'age']=='[0-10)'):
        df.loc[i,'age']=5

age(dbdata)
# Source: https://stackoverflow.com/questions/55159244/age-range-to-age-numerical-valuepython


dbdata['age'] = pd.to_numeric(dbdata['age'], errors='coerce')
```

```
#replacing missing values in the follwing column with
diagcols = ['diag_1','diag_2', 'diag_3']

dbdata[diagcols] = dbdata[diagcols].fillna(0)



dbdata.dropna(inplace = True)


numeric_cols = dbdata.select_dtypes(include=['int64']).copy()
categorica_cols = dbdata.select_dtypes(include=['object']).copy()



for x in numeric_cols:
  sns.boxplot(dbdata.loc[:,x])
  plt.show()
```

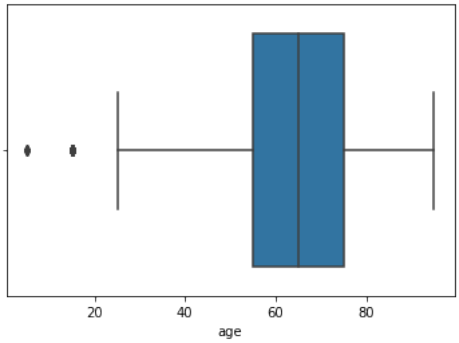
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



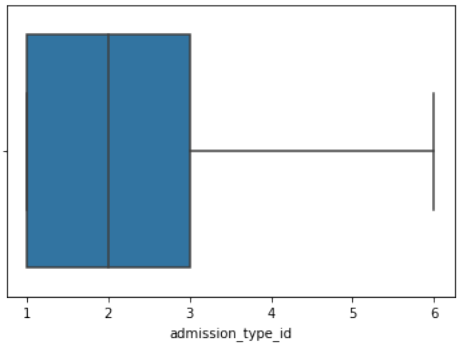

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



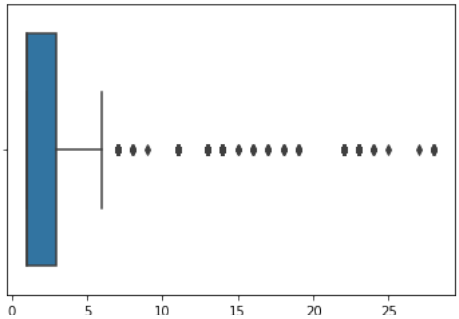

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

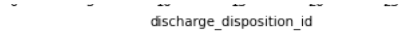



```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

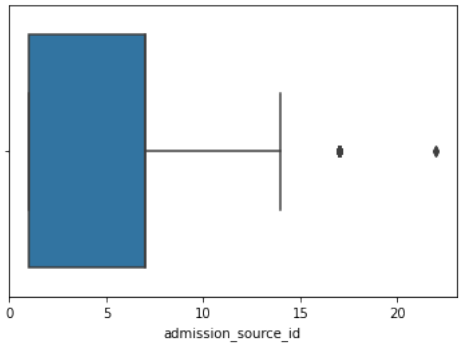



```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

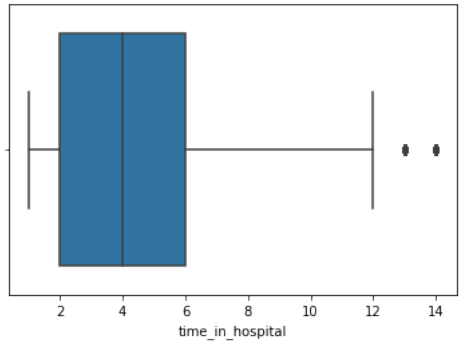




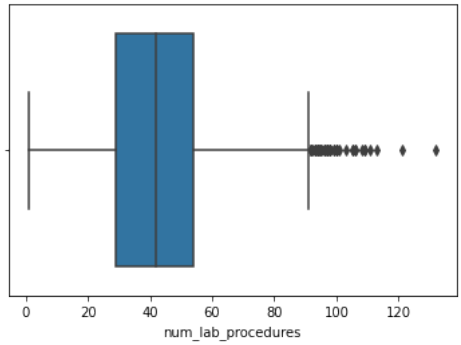

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



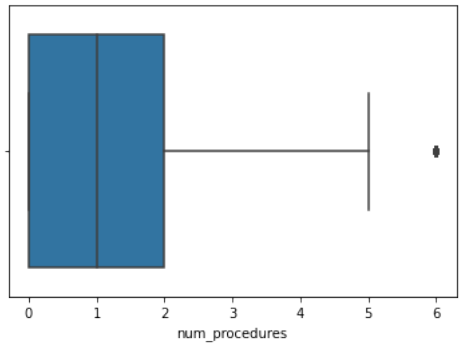

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



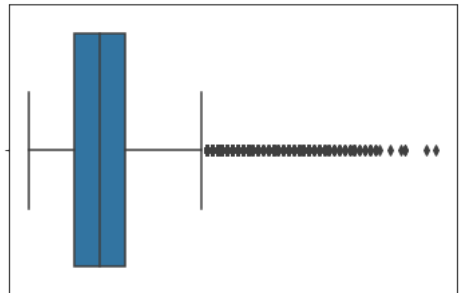

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

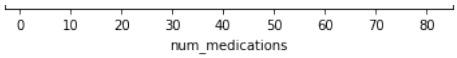



```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

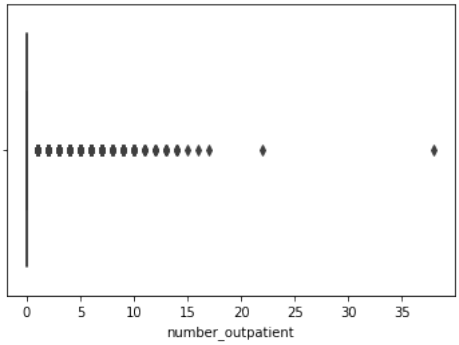



```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

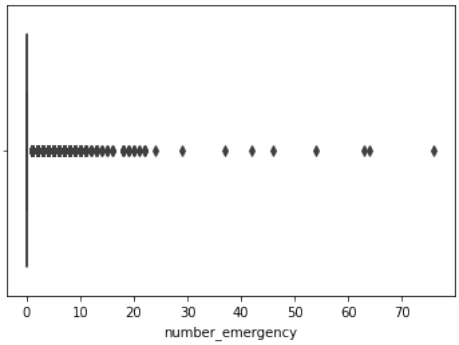




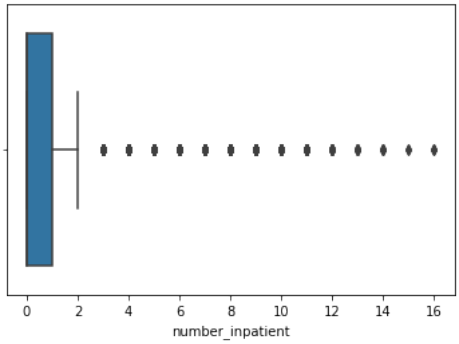

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



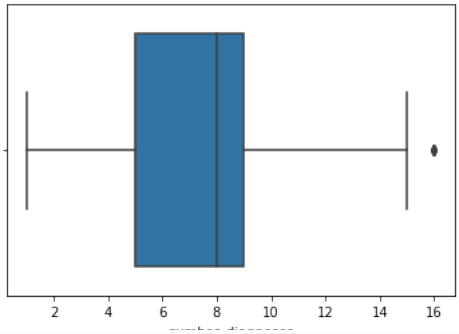

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```




```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```




```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```




```
#removing outliers above 3 std from the mean of the numeric columns
upperlimit = numeric_cols.mean() + 3*numeric_cols.std()
rmoutlier = numeric_cols[numeric_cols < upperlimit]

for x in numeric_cols.columns:
  shell = dbdata[x] <= (dbdata[x].mean() + 3*dbdata[x].std())
  dbdata = dbdata[shell]

dbdata.shape
```

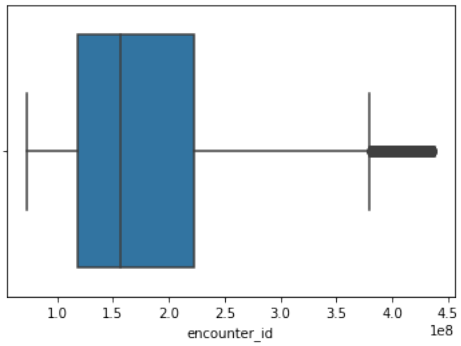
```
(23423, 33)
```

```
dbdata.drop_duplicates(subset=['patient_nbr'], inplace=True)
dbdata.shape
```

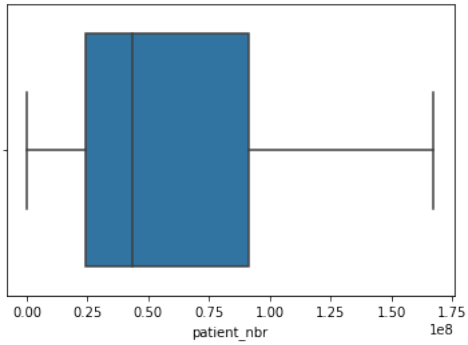
```
(17601, 33)
```

```
for x in numeric_cols:
  sns.boxplot(dbdata.loc[:,x])
  plt.show()
```

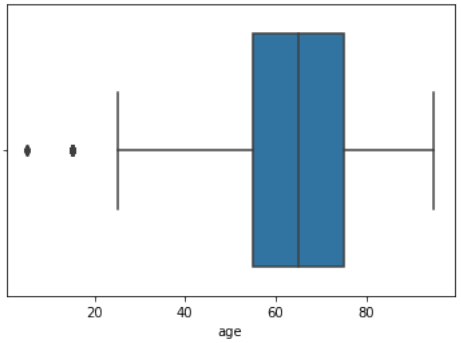
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



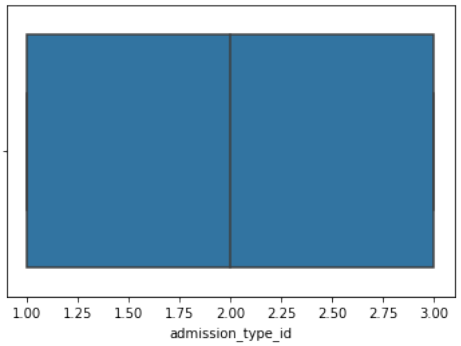

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



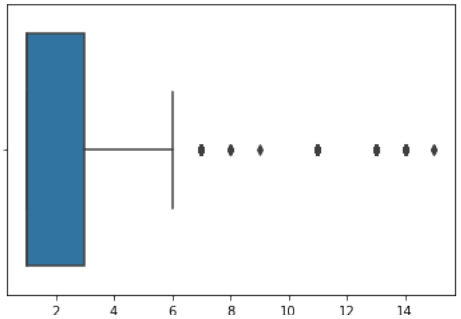

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```




```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

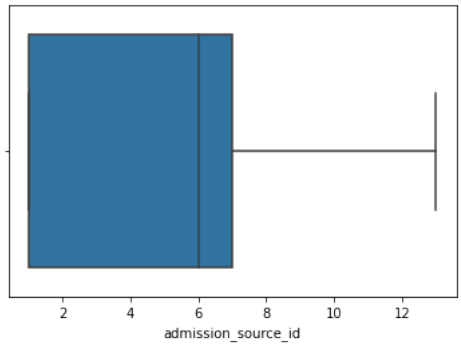



```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

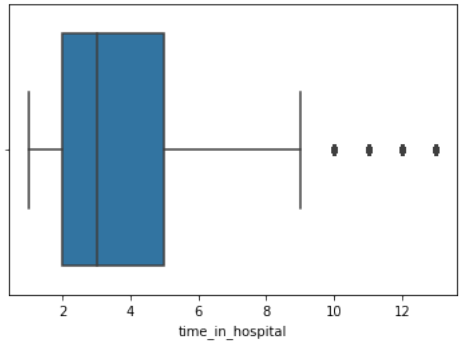


discharge_disposition_id

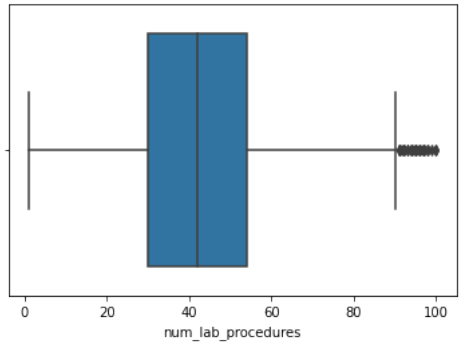
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



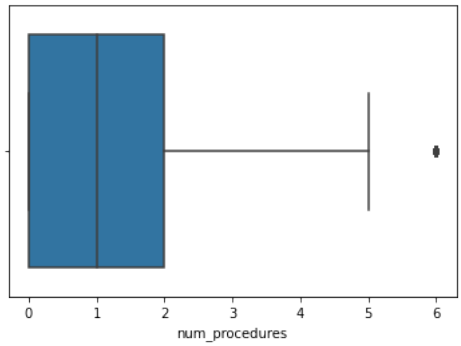

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



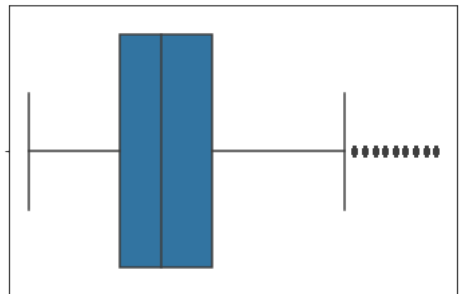

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

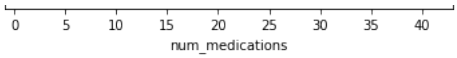



```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

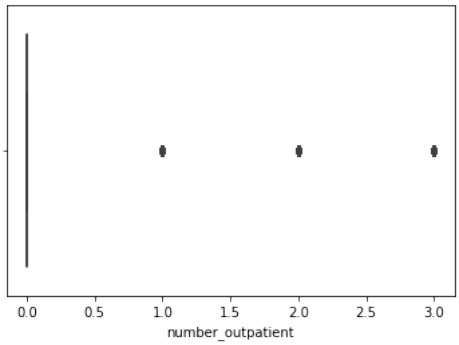



```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```

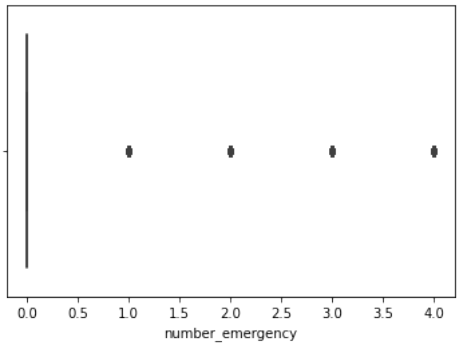




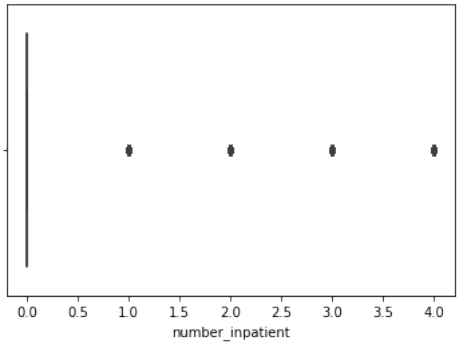

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```



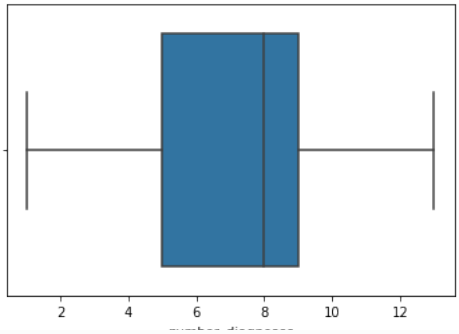

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```




```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```




```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. Fro
  FutureWarning
```




```
dbdata
```

	encounter_id	patient_nbr	race	gender	age	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hosp
20737	72848634	20377854	Caucasian	Female	65	2	1	1	
20824	73062156	20408121	Caucasian	Female	95	1	1	7	
21083	73731852	20542797	Caucasian	Male	75	1	2	7	
23879	81355914	7239654	Caucasian	Female	75	1	3	6	
23922	81458964	15466212	Caucasian	Male	75	1	3	6	
...	...	...	...	...	...	...	...	...	
101249	436785812	87833862	Caucasian	Male	75	2	3	1	
101268	437270768	124367945	Caucasian	Male	65	1	1	7	
101278	437309498	52653654	Caucasian	Male	65	1	1	7	
101284	437331638	142026269	Caucasian	Male	85	2	1	4	
101286	437331728	46488123	Caucasian	Female	85	2	1	1	

17601 rows x 33 columns

```
readmitted_count = dbdata.groupby(['readmitted']).size().sort_values(ascending=False)
readmitted_count
# Providing the information that 37.25% is the mean average rate of readmittence
```

```
readmitted
NO     11045
>30     5158
<30     1398
dtype: int64
```

Data Exploration

[ ] ↳ *37 cells hidden*

Question 2

Data exploration: Carry out a data exploration using appropriate plots to identify patterns or trends in the data. Bearing in mind our objective, we need to assess the impact of the predictors e.g. age, race, gender, or diagnosis type on the outcome (readmitted). Use graphs to prove or disprove the following hypotheses:

- Age has a higher impact on readmission.
- African Americans are more likely to be re-admitted than other ethnic groups.
- Women patients are more likely to be re-admitted than men.
- Diagnose types have a higher impact on re-admission rates. For this purpose, you need to take into account the icd_codes and plot say diag_1 vs readmitted.

Hint 1: You may want to join both datasets diabetic_data.csv and icd_codes.csv.

Hint 2: Check for distinct values in categorical data and their frequencies. If there are too many distinct values (levels), then you may want to reduce the number of levels by grouping some of the detailed levels. This could be the case for race or diagnosis types.

Hint 3: You may want to transform the readmitted column values to be 0 if the value is NO and 1 otherwise for a better exploration of the data.

[ ] ↳ *11 cells hidden*

Part 3

```
subset_list = ['num_medications', 'number_outpatient', 'number_emergency', 'time_in_hospital',\
'number_inpatient', 'encounter_id', 'age', 'num_lab_procedures', 'number_diagnoses',\
'num_procedures', 'readmitted']

logsubset = dbdata[subset_list]
indp_col = dbdata[subset_list].select_dtypes(include=[np.number])
indp_col
#of all the independent variable only age is not numerical.
```

	num_medications	number_outpatient	number_emergency	time_in_hospital	number_inpatient	encounter_id	age	num_lab_procedures
20737	11	0	0	3	0	72848634	65	59
20824	9	0	0	4	0	73062156	95	56
21083	18	0	0	10	0	73731852	75	68
23879	19	0	0	12	0	81355914	75	77
23922	10	0	0	12	0	81458964	75	60
...	...	...	...	...	...	...	...	...
101249	19	0	0	10	0	436785812	75	59
101268	19	0	0	2	0	437270768	65	53
101278	14	1	0	7	0	437309498	65	54
101284	15	0	0	3	1	437331638	85	1
101286	13	0	0	2	1	437331728	85	41

17601 rows x 10 columns

```
logsubset

readmittion_test = logsubset.copy()
```

```
readmittion_test[readmittion_test['readmitted'] != 'NO']
```

	num_medications	number_outpatient	number_emergency	time_in_hospital	number_inpatient	encounter_id	age	num_lab_procedures
24028	5	0	0	6	0	81762780	55	33
24247	13	0	0	9	0	82331772	75	64
24262	2	0	0	1	0	82348062	65	68
24304	7	0	0	4	0	82491186	55	33
24310	12	3	0	3	0	82496730	75	60
...	...	...	...	...	...	...	...	...
101148	29	0	0	8	3	435565568	75	28
101167	13	0	0	1	0	436065734	75	1
101214	33	0	0	3	0	436644764	45	37
101233	13	0	0	3	0	436704890	85	30
101240	13	0	0	5	0	436726154	95	20

6556 rows x 11 columns

```
readmittion_test.loc[:,'readmitted'][readmittion_test.loc[:,'readmitted'] != 'NO'] = 0
readmittion_test.loc[:,'readmitted'][readmittion_test.loc[:,'readmitted'] == 'NO'] = 1
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
```