ML10000PatientDataSet-1

February 22, 2019

In [2]:
```
##########################################################################################
### A 10,000-patient database that contains in total 10,000 patients, 36,143 admission
##########################################################################################




###################################
### PatientCorePopulatedTable ###
###################################
#
#[PatientID] - a unique ID representing a patient.
#[PatientGender] - Male/Female.
#[PatientDateOfBirth] - Date Of Birth.
#[PatientRace] - African American, Asian, White.
#[PatientMaritalStatus] - Single, Married, Divorced, Separated, Widowed.
#[PatientLanguage] - English, Icelandic, Spanish.
#[PatientPopulationPercentageBelowPoverty] - given in %.
#
#####################################
### AdmissionsCorePopulatedTable ###
#####################################
#
#[PatientID] - a unique ID representing a patient.
#[AdmissionID] - an admission ID for the patient.
#[AdmissionStartDate] - start date.
#[AdmissionEndDate] - end date.
#
#############################################
### AdmissionsDiagnosesCorePopulatedTable ###
#############################################
#
#[PatientID] - a unique ID representing a patient.
#[AdmissionID] - an admission ID for the patient.
#[PrimaryDiagnosisCode] - ICD10 code for admission's primary diagnosis.
#[PrimaryDiagnosisDescription] - admission's primary diagnosis description.
#
##############################
```

```
        ### LabsCorePopulatedTable ###
        ##############################
        #
        #[PatientID] - a unique ID representing a patient.
        #[AdmissionID] - an admission ID for the patient.
        #[LabName] - lab's name, including:
        #[LabValue] - lab's value
        #[LabUnits] - lab's units.
        #[LabDateTime] - date.
```

In [3]:
```
        from __future__ import print_function

        print(__doc__)

        import pandas as pd
        import re
        from pathlib import Path
        from sklearn.model_selection import train_test_split
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.tree import DecisionTreeClassifier
        from sklearn import metrics
        import numpy as np
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D
        from sklearn import decomposition
        from sklearn import datasets
        from sklearn import preprocessing
        from sklearn.impute import SimpleImputer
        from scipy.ndimage import convolve
        from sklearn import linear_model, datasets, metrics
        from sklearn.model_selection import train_test_split
        from sklearn.neural_network import BernoulliRBM
        from sklearn.pipeline import Pipeline
        from sklearn.base import clone
        from sklearn.datasets import make_multilabel_classification
        from sklearn.multiclass import OneVsRestClassifier
        from sklearn.svm import SVC
        from sklearn.decomposition import PCA
        from sklearn.cross_decomposition import CCA
        from matplotlib.mlab import PCA
        from sklearn.preprocessing import StandardScaler
        from sklearn.preprocessing import StandardScaler, OneHotEncoder
        from sklearn.compose import ColumnTransformer, make_column_transformer
        from sklearn.model_selection import train_test_split
        from sklearn.pipeline import make_pipeline
        from sklearn.linear_model import LogisticRegression
        from sklearn.impute import SimpleImputer
        from scipy.stats import ttest_ind
```

Automatically created module for IPython interactive environment

In [4]:
```
#data_folder = Path("/local/") #Path("C:/Users/david_000/Desktop/healthdata/csv/")
```

In [5]:
```
###################################
### PatientCorePopulatedTable ###
###################################


file_to_open = "AdmissionsCorePopulatedTable.csv"
columnsadmit = ['PatientID', 'AdmissionID', 'AdmissionStartDate', 'AdmissionEndDate']
f = open(file_to_open)
admitdf = pd.read_csv(f, index_col=False, names=columnsadmit)
admitdf.head()
```

In [6]:
```
#####################################
### AdmissionsCorePopulatedTable ###
#####################################


file_to_open1 =  "PatientCorePopulatedTable.csv"
columnspatient = ['PatientID', 'PatientGender', 'PatientDateOfBirth', 'PatientRace', '
f1 = open(file_to_open1)
patientdf = pd.read_csv(f1, index_col=False, names=columnspatient)
patientdf.head()
```

In [7]:
```
###############################################
### AdmissionsDiagnosesCorePopulatedTable ###
###############################################

file_to_open2 =  "AdmissionsDiagnosesCorePopulatedTable.csv"
columnsdiagnoses = ['PatientID', 'PatientGender', 'PrimaryDiagnosisCode', 'PrimaryDiagn
f2 = open(file_to_open2, encoding="latin-1")
diagnosesdf = pd.read_csv(f2, index_col=False, names=columnsdiagnoses)
del diagnosesdf['PatientMaritalStatus']
diagnosesdf.head()
```

In [8]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open3 = "1.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[LabDa
df3 = open(file_to_open3)
labsdf1 = pd.read_csv(df3, index_col=False , names=columnslabs,sep="\t", engine='pytho



labsdf1.head()
```

In [9]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open32 =  "2.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[LabDa
df32 = open(file_to_open32)
labsdf2 = pd.read_csv(df32, index_col=False,names=columnslabs ,sep="\t", engine='pythor

#pd.read_csv(name,sep=";|,")


labsdf2.head()
```

In [10]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open33 = "3.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
df33 = open(file_to_open33)
labsdf3 = pd.read_csv(df33, index_col=False ,names=columnslabs, sep="\t", engine='pytl

#pd.read_csv(name,sep=";|,")


labsdf3.head()
```

In [11]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open34 = "4.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
df34 = open(file_to_open34)
labsdf4 = pd.read_csv(df34, index_col=False , names=columnslabs,sep="\t", engine='pytl

#pd.read_csv(name,sep=";|,")


labsdf4.head()
```

In [12]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open35 =  "5.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
df35 = open(file_to_open35)
labsdf5 = pd.read_csv(df35, index_col=False , names=columnslabs,sep="\t", engine='pytl

#pd.read_csv(name,sep=";|,")
```

```
labsdf5.head()
```

In [13]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open36 =  "6.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
df36 = open(file_to_open36)
labsdf6 = pd.read_csv(df36, index_col=False , names=columnslabs,sep="\t", engine='pytl

#pd.read_csv(name,sep=";|,")


labsdf6.head()
```

In [14]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open37 = "7.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
df37 = open(file_to_open37)
labsdf7 = pd.read_csv(df37, index_col=False , names=columnslabs,sep="\t", engine='pytl

#pd.read_csv(name,sep=";|,")


labsdf7.head()
```

In [15]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open38 =  "8.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
df8 = open(file_to_open38)
labsdf8 = pd.read_csv(df8, index_col=False , names=columnslabs,sep="\t", engine='pytho

#pd.read_csv(name,sep=";|,")


labsdf8.head()
```

In [16]:
```
##############################
### LabsCorePopulatedTable ###
##############################
file_to_open39 =  "9.txt"
columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
df9 = open(file_to_open39)
```

```
        labsdf9 = pd.read_csv(df9, index_col=False , names=columnslabs,sep="\t", engine='pytho

        #pd.read_csv(name,sep=";|,")


        labsdf9.head()

In [17]: ##############################
        ### LabsCorePopulatedTable ###
        ##############################
        file_to_open310 =  "10.txt"
        columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
        df310 = open(file_to_open310)
        labsdf10 = pd.read_csv(df310, index_col=False , names=columnslabs,sep="\t", engine='py

        #pd.read_csv(name,sep=";|,")


        labsdf10.head()

In [18]: ##############################
        ### LabsCorePopulatedTable ###
        ##############################
        file_to_open311 = "11.txt"
        columnslabs = ['PatientID', 'AdmissionID', 'LabName', '[LabValue]','[LabUnits]','[Labl
        df311 = open(file_to_open311)
        labsdf11 = pd.read_csv(df311, index_col=False , names=columnslabs,sep="\t", engine='py

        #pd.read_csv(name,sep=";|,")


        labsdf11.head()

In [19]: frames = [labsdf1, labsdf2, labsdf3,labsdf4,labsdf5,labsdf6,labsdf7,labsdf8,labsdf9,la
        result = pd.concat(frames)
        result=result.drop([0])

In [20]: result.head(5)

In [21]: print(diagnosesdf.groupby('PrimaryDiagnosisDescription').PrimaryDiagnosisDescription.c

PrimaryDiagnosisDescription
Abnormal findings on diagnostic imaging of heart and coronary circulation                18
Abnormal results of cardiovascular function studies                                       8
Abnormal results of pulmonary function studies                                           10
Abuse of non-psychoactive substances                                                     14
Acoustic neuritis in infectious and parasitic diseases classified elsewhere              11
Acoustic neuritis in infectious and parasitic diseases classified elsewhere, bilateral   13
Acoustic neuritis in infectious and parasitic diseases classified elsewhere, left ear    10
```

```
Acoustic neuritis in infectious and parasitic diseases classified elsewhere, right ear    10
Acute Chagas' disease with heart involvement                                              12
Acute Chagas' disease without heart involvement                                           12
Acute bronchitis due to Hemophilus influenzae                                             15
Acute bronchitis due to parainfluenza virus                                               13
Acute cerebrovascular insufficiency                                                       17
Acute coronary thrombosis not resulting in myocardial infarction                          13
Acute drug-induced interstitial lung disorders                                            14
Acute erythroid leukemia                                                                  18
Acute erythroid leukemia, in relapse                                                      19
Acute erythroid leukemia, in remission                                                    16
Acute erythroid leukemia, not having achieved remission                                    8
Acute graft-versus-host disease                                                           10
Acute idiopathic pulmonary hemorrhage in infants                                          13
Acute inflammatory disease of uterus                                                      14
Acute lymphoblastic leukemia [ALL]                                                        10
Acute lymphoblastic leukemia not having achieved remission                                15
Acute lymphoblastic leukemia, in relapse                                                  17
Acute lymphoblastic leukemia, in remission                                                13
Acute megakaryoblastic leukemia                                                           18
Acute megakaryoblastic leukemia not having achieved remission                             12
Acute megakaryoblastic leukemia, in relapse                                                7
Acute megakaryoblastic leukemia, in remission                                             16
                                                                                          ..
Vascular anomalies of eyelid                                                              18
Vascular anomalies of left lower eyelid                                                   17
Vascular anomalies of left upper eyelid                                                   11
Vascular anomalies of right lower eyelid                                                  18
Vascular anomalies of right upper eyelid                                                  20
Vascular complications following infusion, transfusion and therapeutic injection         13
Vascular dementia                                                                         28
Vascular dementia with behavioral disturbance                                             18
Vascular dementia without behavioral disturbance                                          14
Vascular disorders of intestine                                                           16
Vascular headache, not elsewhere classified                                               17
Vascular myelopathies                                                                     17
Vascular parkinsonism                                                                     13
Vascular syndromes of brain in cerebrovascular diseases                                   12
Vertiginous syndromes in diseases classified elsewhere                                    11
Vertiginous syndromes in diseases classified elsewhere, bilateral                          7
Vertiginous syndromes in diseases classified elsewhere, left ear                          15
Vertiginous syndromes in diseases classified elsewhere, right ear                         13
Vibrio vulnificus as the cause of diseases classified elsewhere                           11
Viral agents as the cause of diseases classified elsewhere                                19
Voice and resonance disorders                                                             15
Von Willebrand's disease                                                                   8
Wandering in diseases classified elsewhere                                                10
Wegener's granulomatosis with renal involvement                                           18
```

```
Wegener's granulomatosis without renal involvement                    13
Whipple's disease                                                     13
Wilson's disease                                                      12
Yaba pox virus disease                                                11
Zoster ocular disease                                                  9
von Gierke disease                                                    13
Name: PrimaryDiagnosisDescription, Length: 2619, dtype: int64
```

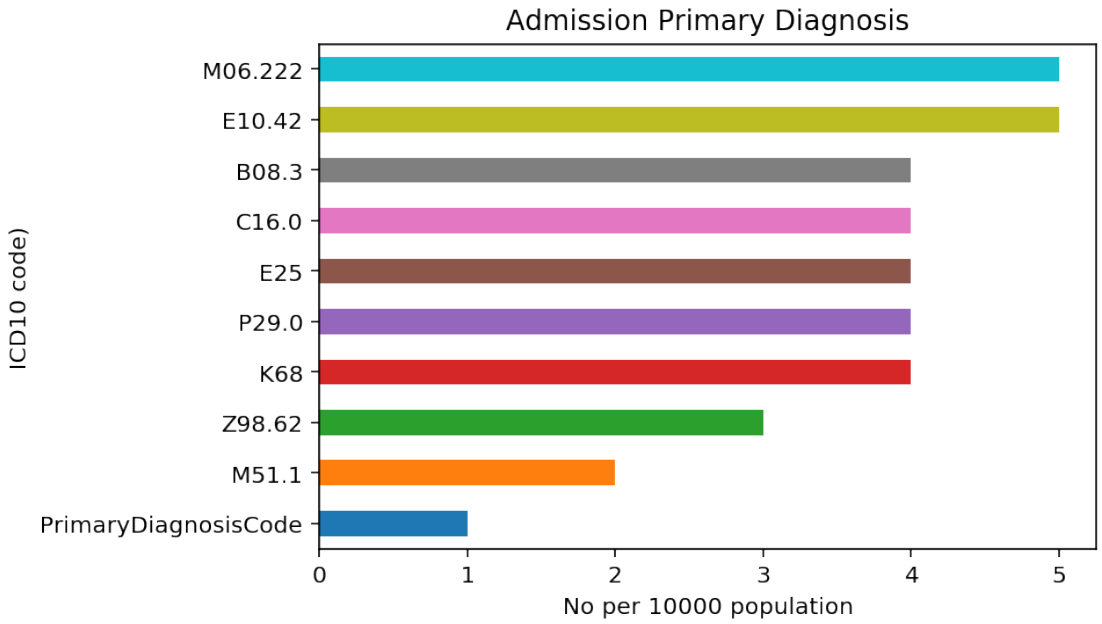
In [0]:

In [22]:
```
#(diagnosesdf['PrimaryDiagnosesCode'].unique)

pulmonarydisdf=diagnosesdf[diagnosesdf.PrimaryDiagnosisCode.str.startswith('I2')]
pulmonarydisdf
```

In [23]:
```
#The prinmary Admissiondiagnosis ICD10 code
ICD = ((diagnosesdf.groupby('PrimaryDiagnosisCode').PrimaryDiagnosisCode.count()))
ICD=ICD.sort_values(ascending=True)
ICDplot = ICD.head(10).plot(kind='barh',legend=None,title="Admission Primary Diagnosis
ICDplot.set_xlabel("No per 10000 population")
ICDplot.set_ylabel("ICD10 code)")
```

Out[23]:
```
Text(0,0.5,'ICD10 code)')
```

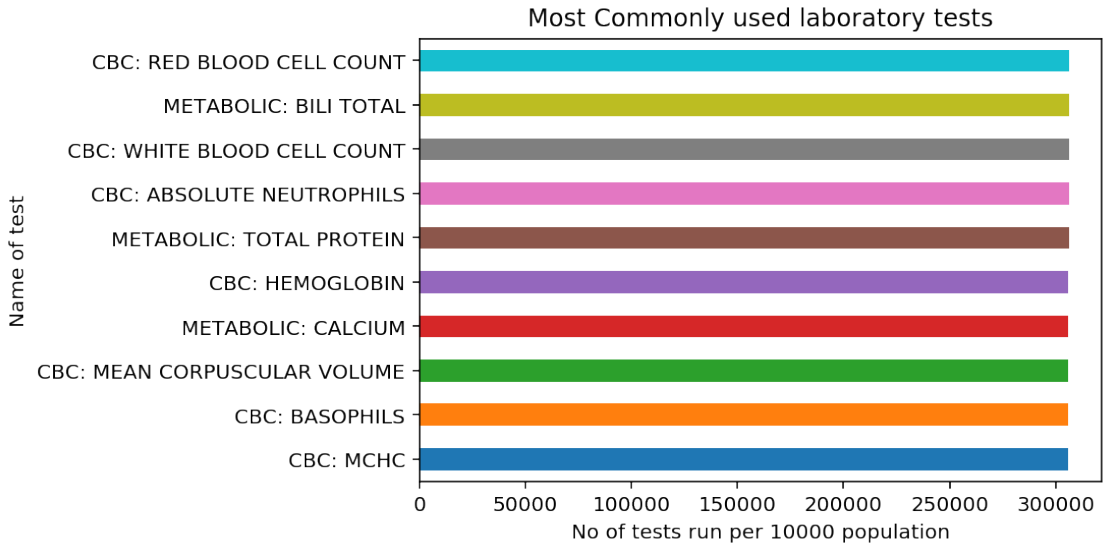
Out[23]:



In [24]:
```
#Laborary Tests by Name wich are on patients (LN)
LN = ((result.groupby('LabName').LabName.count()))
LN=LN.sort_values(ascending=True)
LNplot = LN.head(10).plot(kind='barh',legend=None,title="Most Commonly used laborator
LNplot.set_xlabel("No of tests run per 10000 population")
LNplot.set_ylabel("Name of test")
```

Out[24]:
```
Text(0,0.5,'Name of test')
```

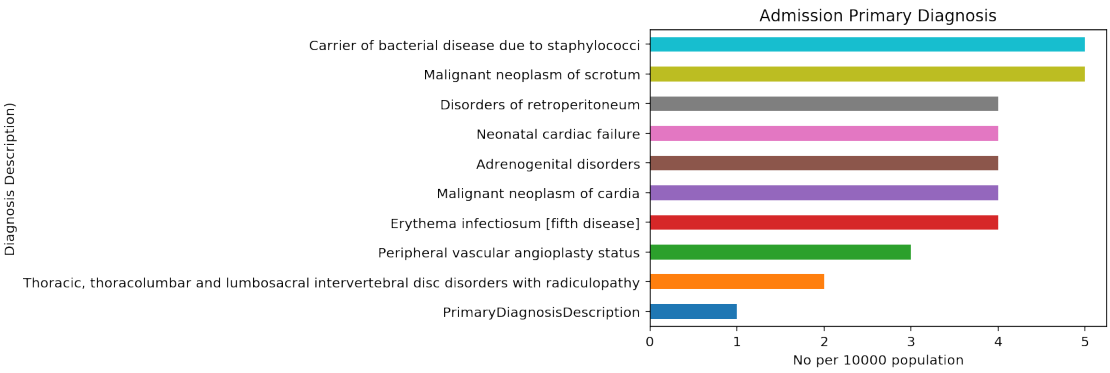
Out[24]:



In [25]:
```
#The primary Admission Diagnosis Description (DD)
DD = ((diagnosesdf.groupby('PrimaryDiagnosisDescription').PrimaryDiagnosisDescription
DD=DD.sort_values(ascending=True)
DDplot = DD.head(10).plot(kind='barh',legend=None,title="Admission Primary Diagnosis"
DDplot.set_xlabel("No per 10000 population")
DDplot.set_ylabel("Diagnosis Description)")
```

Out[25]:
```
Text(0,0.5,'Diagnosis Description)')
```

Out[25]:



```
In [26]: LV=result[['LabName','[LabValue]']]
         LV.head()

In [27]: LV.groupby('LabName', as_index=False)['[LabValue]'].head(5)

Out[27]: 1       40.7
         2        8.4
         3        4.7
         4       15.9
         5      146.6
         6        3.3
         7       17.1
         8        8.4
         9        8.7
         10     110.5
         11       5.5
         12       0.3
         13       5.3
         14       2.2
         15      25.5
         16       0.6
         17       1.0
         18      97.5
         19     109.1
         20      39.9
         21      14.7
         22      44.8
         23       0.2
         24      67.5
         25      38.9
         26       0.2
         27     103.3
         28       0.4
         29      13.7
         30     111.8
                ...
         147      3.3
         149      2.6
         150      0.2
         154     67.1
         156     26.1
         157     24.4
         158     84.0
         159      0.8
         160     20.1
```

```
161     40.6
162    108.8
163      5.9
165     29.6
167     11.4
168     14.5
169     11.1
170      0.3
173     93.1
176      5.3
177     18.2
182    150.8
183      6.4
184     85.9
190      0.6
193      0.2
194      0.1
200      7.4
226      5.5
231      5.7
235      2.3
Name: [LabValue], Length: 175, dtype: float64
```

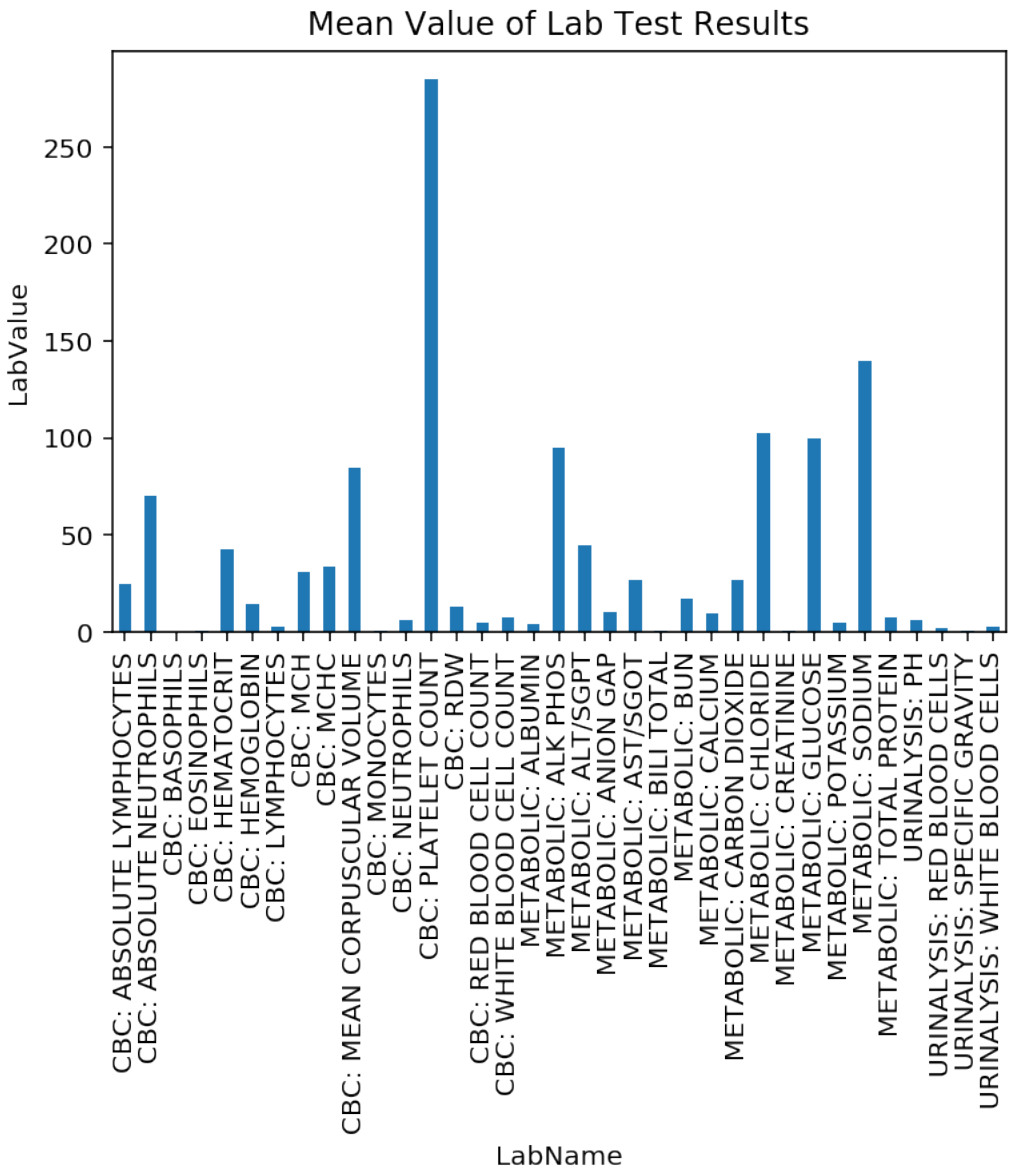
In [28]:
```
LV.groupby('LabName', as_index=False)['[LabValue]'].mean()
```

In [30]:
```
meandf=LV.groupby('LabName', as_index=False)['[LabValue]'].mean()
meandf.set_index("LabName",drop=True,inplace=True)
meanplot = plt.figure(figsize = (60,80))
meanplot = meandf.plot(kind='bar',legend=None,title="Mean Value of Lab Test Results" )
meanplot.set_xlabel("LabName")
meanplot.set_ylabel("LabValue")
```

Out[30]:
```
Text(0,0.5,'LabValue')
```

Out[30]:
```
<matplotlib.figure.Figure at 0x7f313412f6a0>
```

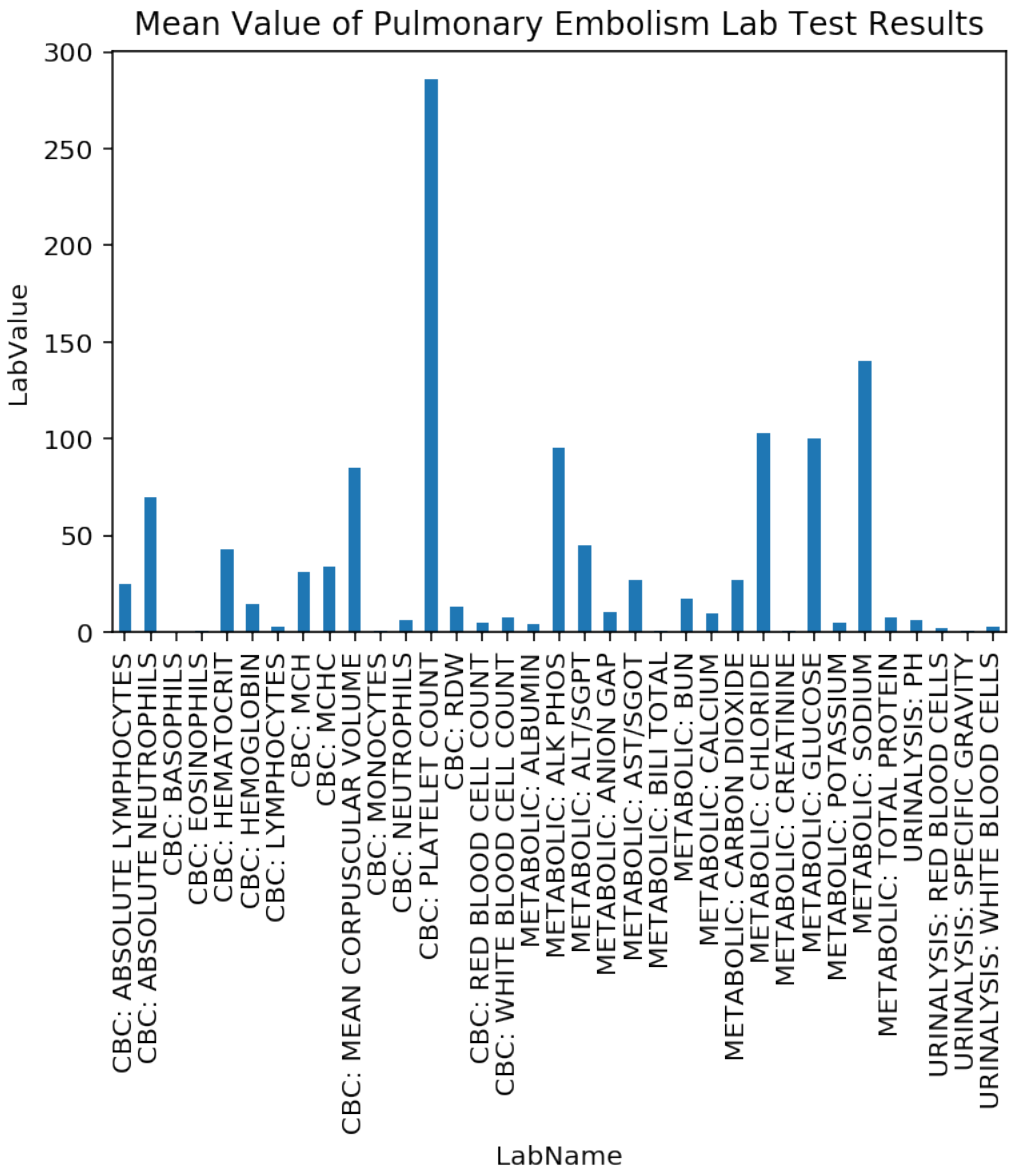
Out[30]:




```
In [31]: results = pd.merge(result, diagnosesdf[['PatientID', 'PatientGender', 'PrimaryDiagnosi
         results.head()

In [32]: (diagnosesdf[diagnosesdf.PatientID == '74CBA06C-1029-4B38-920E-638B3ACF0009'])

In [33]: print(diagnosesdf[diagnosesdf.PatientID == '74CBA06C-1029-4B38-920E-638B3ACF0009'])

                                    PatientID PatientGender  \
33901  74CBA06C-1029-4B38-920E-638B3ACF0009              1
33902  74CBA06C-1029-4B38-920E-638B3ACF0009              2
33903  74CBA06C-1029-4B38-920E-638B3ACF0009              3
```

```
      PrimaryDiagnosisCode                        PrimaryDiagnosisDescription
33901                R93.1  Abnormal findings on diagnostic imaging of hea...
33902                M01.X  Direct infection of joint in infectious and pa...
33903                D36.1  Benign neoplasm of peripheral nerves and auton...
```

```
In [34]: pulmonaryLabsdf=results[results.PrimaryDiagnosisCode.str.startswith('I2')]
         pulmonaryLabsdf
         pulmonaryLabsdf.to_csv("pulmonaryLabsdf.csv", index=False, encoding='utf8')

In [35]: Pl=pulmonaryLabsdf[['LabName','[LabValue]']]
         Pl.groupby('LabName', as_index=False)['[LabValue]'].mean()

In [0]:

In [37]: #Pl=pulmonaryLabsdf[['LabName','[LabValue]']]
         meanPuldf=Pl.groupby('LabName', as_index=False)['[LabValue]'].mean()
         meanPuldf.set_index("LabName",drop=True,inplace=True)
         meanPulplot = plt.figure(figsize = (60,80))
         meanPulplot = meanPuldf.plot(kind='bar',legend=None,title="Mean Value of Pulmonary Em
         meanPulplot.set_xlabel("LabName")
         meanPulplot.set_ylabel("LabValue")

Out[37]: Text(0,0.5,'LabValue')

Out[37]: <matplotlib.figure.Figure at 0x7f314292edd8>

Out[37]:
```

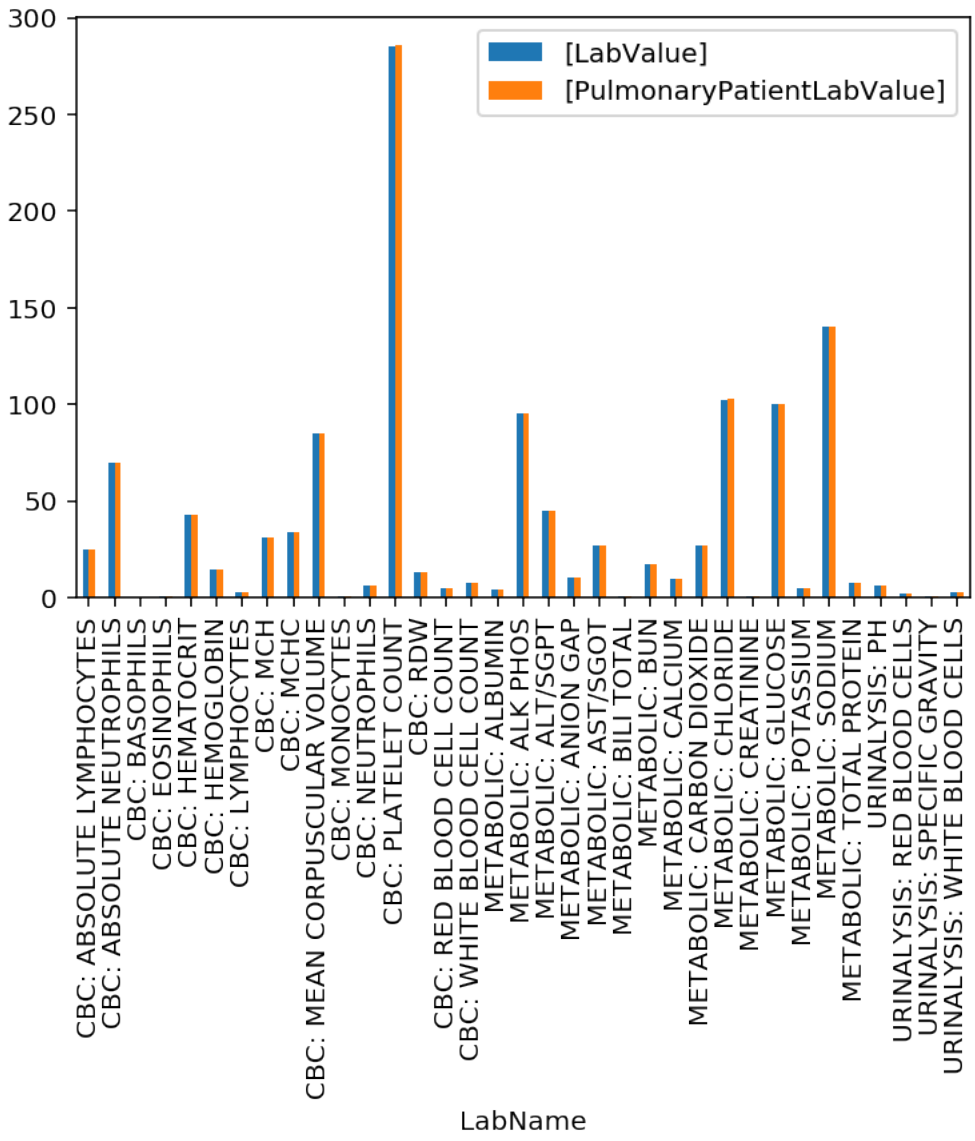



```
In [38]: meanPuldf.columns= ['[PulmonaryPatientLabValue]']
         meanPuldf

In [39]: meandf.columns.values

Out[39]: array(['[LabValue]'], dtype=object)

In [40]: meanPuldf

In [41]: #Labresults
         labresults=meandf.join(meanPuldf, how='outer')
         labresults
```

In [43]: labresults[['[LabValue]','[PulmonaryPatientLabValue]']].plot(kind='bar')

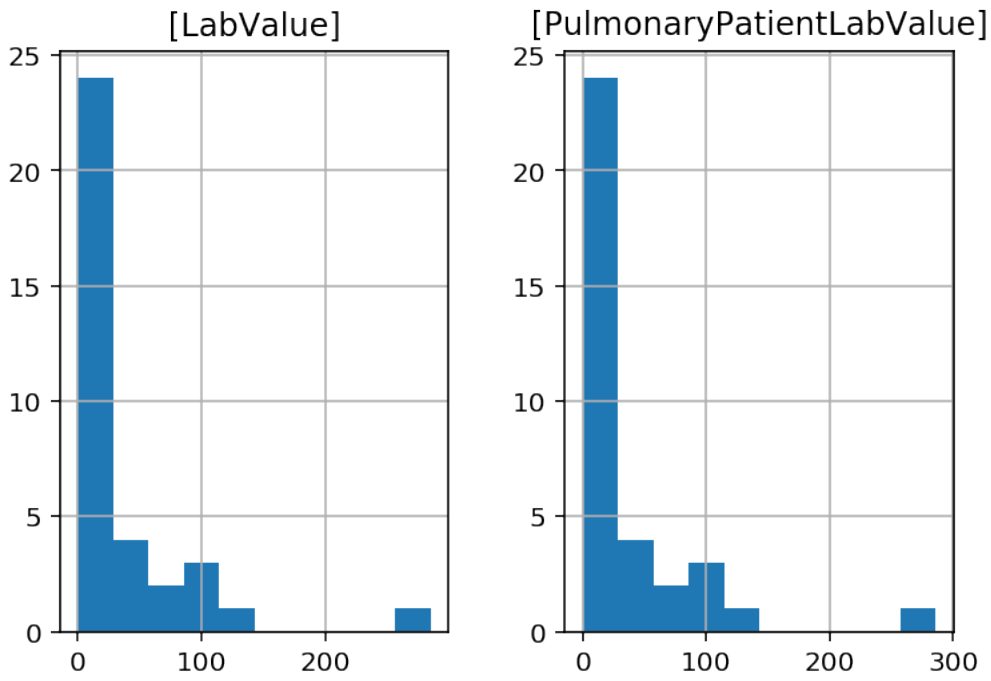
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7f312ec05cc0>

Out[43]:



In [44]: hist = labresults.hist(bins=10)

Out[44]:

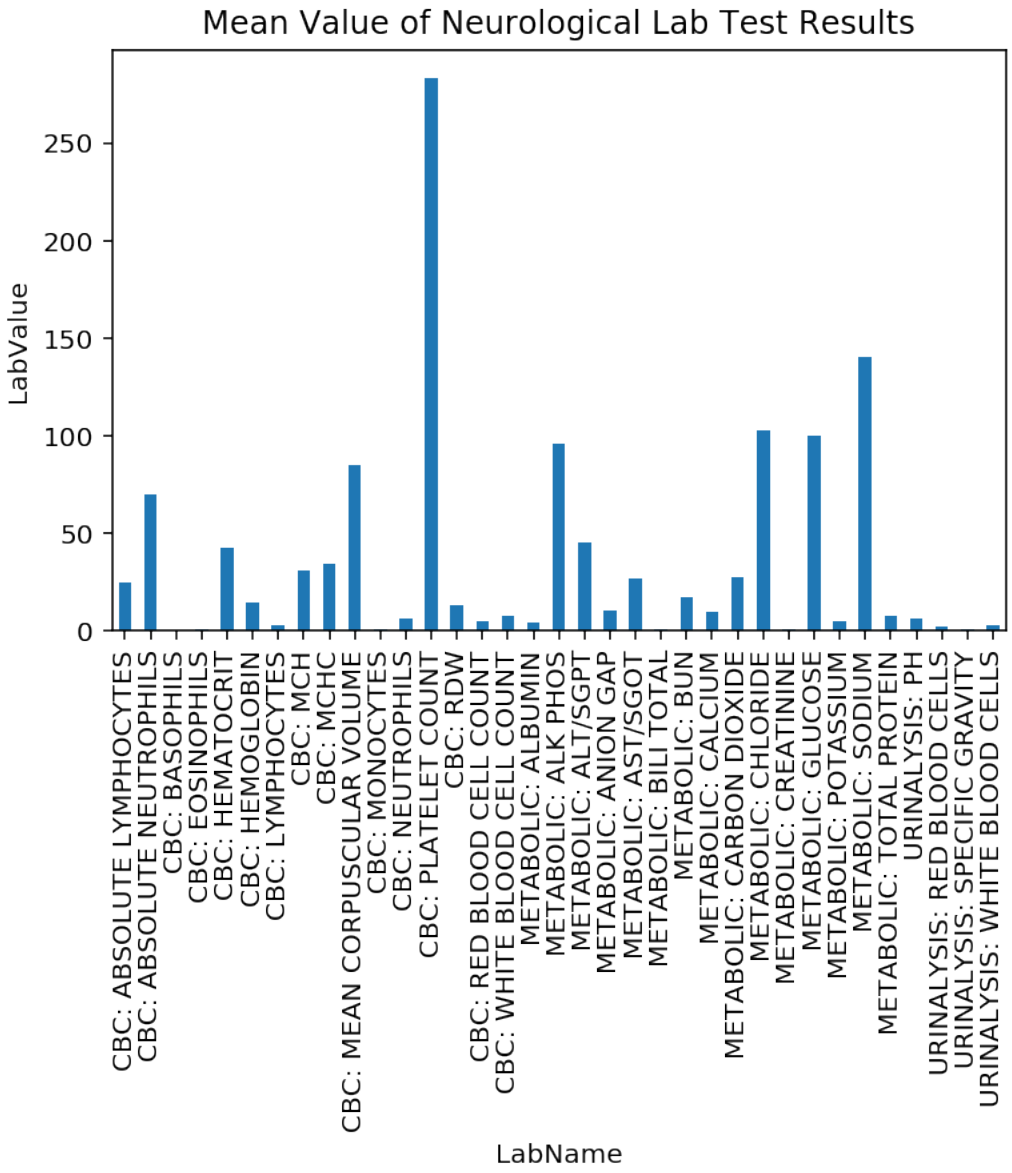



```
In [45]: labresults.columns.values

Out[45]: array(['[LabValue]', '[PulmonaryPatientLabValue]'], dtype=object)

In [46]: #Test for significant statisical difference

         ttest_ind(labresults['[LabValue]'].values, labresults['[PulmonaryPatientLabValue]'].va

Out[46]: Ttest_indResult(statistic=-0.0019413413751742322, pvalue=0.9984567187423385)

In [47]: neurodf=results[results.PrimaryDiagnosisCode.str.startswith('F2')]
         neurodf

In [48]: nl=neurodf[['LabName','[LabValue]']]
         nl.groupby('LabName', as_index=False)['[LabValue]'].mean()

In [0]:

In [0]:

In [50]: meanneurodf=nl.groupby('LabName', as_index=False)['[LabValue]'].mean()
         meanneurodf.set_index("LabName",drop=True,inplace=True)
         meanneuroplot = plt.figure(figsize = (60,80))
         meanneuroplot = meanneurodf.plot(kind='bar',legend=None,title="Mean Value of Neurologi
         meanneuroplot.set_xlabel("LabName")
         meanneuroplot.set_ylabel("LabValue")
```

Out[50]: Text(0,0.5,'LabValue')

Out[50]: <matplotlib.figure.Figure at 0x7f31388d0a90>

Out[50]:



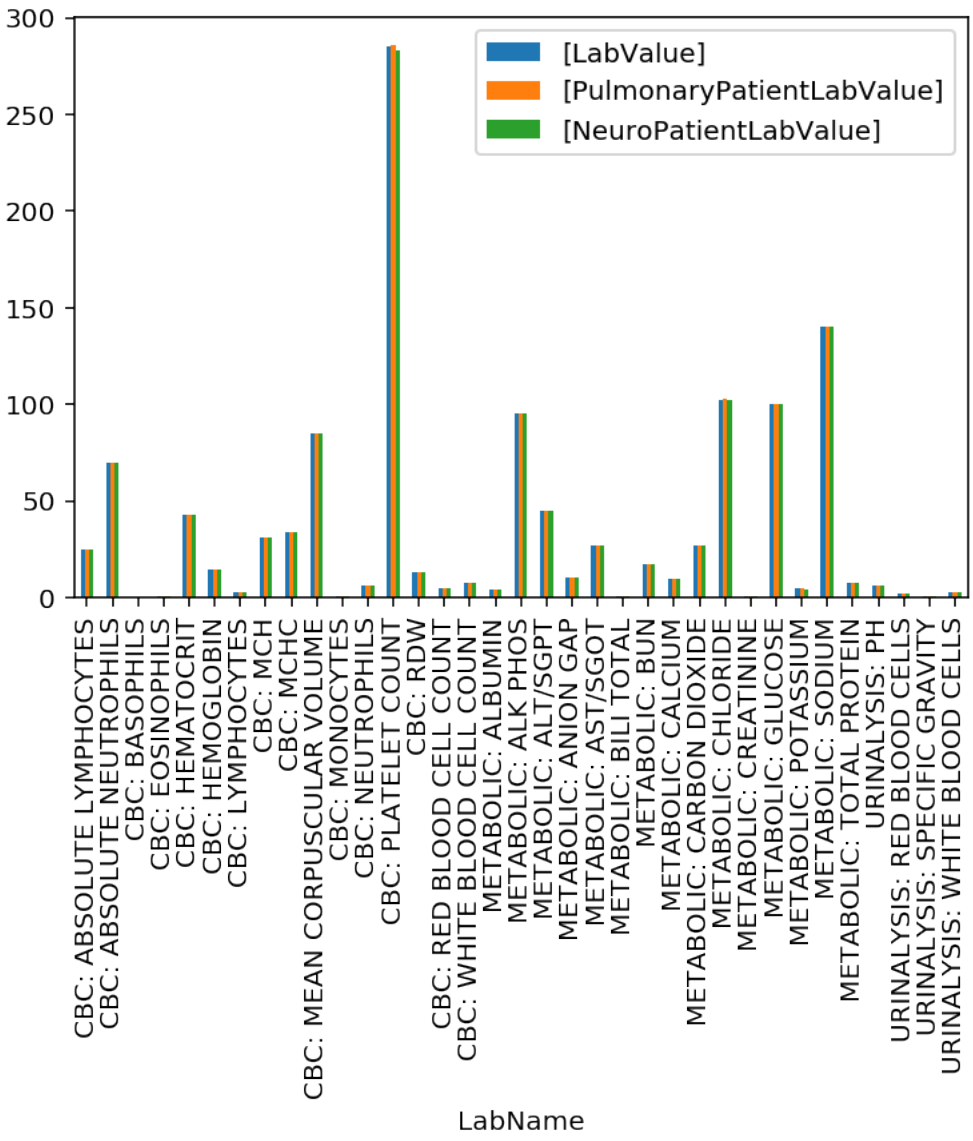
In [51]:
```
meanneurodf.columns= ['[NeuroPatientLabValue]']
meanneurodf
```

In [52]:
```
#Labresults 3
labresultsall=meanneurodf.join(labresults, how='outer')
labresultsall
```

In [54]: labresultsall[['[LabValue]','[PulmonaryPatientLabValue]','[NeuroPatientLabValue]']].

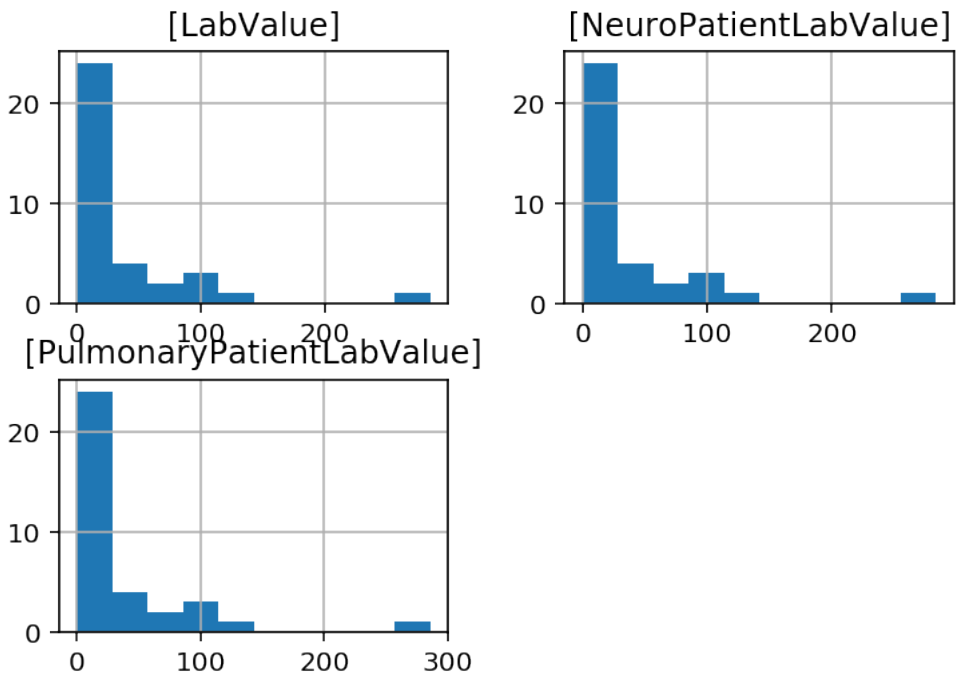
Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3129979828>

Out[54]:



In [56]: hist = labresultsall.hist(bins=10)

Out[56]:




```
In [57]: #Test for significant statisical difference

         ttest_ind(labresultsall['[LabValue]'].values, labresultsall['[NeuroPatientLabValue]']

Out[57]: Ttest_indResult(statistic=0.001775284000266084, pvalue=0.9985887269258076)

In [0]:
```