

```
In [ ]: import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
from scipy.stats import ttest_ind
import pandas as pd
import numpy as np
import seaborn as sns

%matplotlib inline
pd.set_option('display.max_columns', None)
# read the file of medical data from CSV file
df = pd.read_csv (r'C:\Users\MichaelRupert\Downloads\e9d8sm5uf8df75k650df\medical_cleaned_data.csv')
#set up the correlation matrix
df.test=df[['Age', 'ReAdmis', 'Doc_visits', 'Soft_drink', 'HighBlood', 'Stroke', 'Complication_risk'
            , 'Overweight', 'Arthritis', 'Diabetes',
            'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis', 'Reflux_esophagitis'
            , 'Asthma', 'Initial_days', 'Timely_Admission', 'Timely_Treatment']]
print(df.test.corr(method = 'pearson'))

#use this variable list to run a loop used to output T Test results
VariableList = ['Age', 'ReAdmis', 'Doc_visits', 'Soft_drink', 'HighBlood', 'Stroke', 'Complication_risk',
                'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain',
                'Anxiety', 'Allergic_rhinitis',
                'Reflux_esophagitis'
                , 'Asthma', 'Initial_days', 'Timely_Admission', 'Timely_Treatment']

#T-Test output through use of a for loop compare each variable to ReAdmis
for x in VariableList:
    no = df.loc[df.ReAdmis == 0, x]
    yes = df.loc[df.ReAdmis == 1, x]

    t_Value,p_value = ttest_ind(yes,no)#store the statistic and p-value
    one_tailed_p_value = p_value/2 #divide by two to get the onesided test value
    #print out the results
    print("Result for " + x + " statistic: " + str(t_Value) + " p-value: " + str(one_tailed_p_value))

    alpha = 0.05 #use 0.05 as a typical level of significance.
    #check the level of sginificance compared to the resultant p value.
    #Fail to reject the Null Hyp or reject.
    if((one_tailed_p_value) <= alpha):
        print("Conclusion: " + "Since p-value(=%f) "%one_tailed_p_value + " < " + "alpha(=%f)." %alpha +
              " We reject the null hypothesis H0." + "\n" + "So we conclude that there is evidence that "
              + x + " could be a factor in patient Readmission." + "\n" +
              " i.e.,  $\mu = 165$  at %f level of significance" %alpha + "\n" +
              " This is an DEPENDENT variable.\n")
    else:
        print('Conclusion', 'n', 'Since p-value(=%f) "%one_tailed_p_value, '>', 'alpha(=%f) "%alpha,
              'We fail reject the null hypothesis H0.' + "\n" +
              "This is a INDEPENDENT variable. \n")

df = pd.read_csv (r'C:\Users\MichaelRupert\Downloads\e9d8sm5uf8df75k650df\medical_raw_data.csv')
#Histograms Univariate Stats
df.hist('Initial_days',bins=75)

df.hist('Age',bins=40)

table = pd.pivot_table(df.groupby(['Overweight']).size().reset_index(),values=0,index='Overweight',
                        columns=['Overweight'],aggfunc=np.sum)
table.plot(kind='bar',stacked=True)

table = pd.pivot_table(df.groupby(['Asthma']).size().reset_index(),values=0,index='Asthma',
                        columns=['Asthma'],aggfunc=np.sum)
table.plot(kind='bar',stacked=True)

#Bivariate statistics
sns.catplot(x="Doc_visits", y="Initial_days", data=df)

sns.catplot(y="Age", x="Doc_visits", data=df)

tbl = pd.pivot_table(df.groupby(['ReAdmis', 'Arthritis']).size().reset_index(),
                      values=0,
                      index='ReAdmis',
                      columns=['Arthritis'],
                      aggfunc=np.sum)
tbl.plot(kind='bar',stacked=True)

tbl = pd.pivot_table(df.groupby(['ReAdmis', 'BackPain']).size().reset_index(),
                      values=0,
                      index='ReAdmis',
                      columns=['BackPain'],
                      aggfunc=np.sum)
tbl.plot(kind='bar',stacked=True)
```