**Machine Learning & Data Mining**

***General-Social-Survey***

***and***

***Sadness***

***Data Analysis***

**Sudarshan.M.S**

**[CB.EN.U4CSE18258]**

***Q1) GSS Analysis***

**Code -**

import csv

import pandas as pd

import numpy as np

import plotly.plotly as py

import plotly.graph\_objs as go

import matplotlib.pyplot as plt

gss = pd.read\_csv('GSS7214.csv', delimiter=",")

gss[0:5].reset\_index(drop=True, inplace=True)

gss.loc[gss['year'] == 1972]

gss.loc[gss['year'] == 1973]

gss.loc[gss['year'] == 1974]

gss.loc[gss['year'] == 1975]

gss.loc[gss['year'] == 1976]

gss.loc[gss['year'] == 1977]

gss.loc[gss['year'] == 1978]

gss.loc[gss['year'] == 1979]

gss.loc[gss['year'] == 1980]

gss.loc[gss['year'] == 1981]

gss.loc[gss['year'] == 1982]

gss.loc[gss['year'] == 1983]

gss.loc[gss['year'] == 1984]

gss.loc[gss['year'] == 1985]

gss.loc[gss['year'] == 1986]

gss.loc[gss['year'] == 1987]

gss.loc[gss['year'] == 1988]

gss.loc[gss['year'] == 1989]

gss.loc[gss['year'] == 1990]

gss.loc[gss['year'] == 1991]

gss.loc[gss['year'] == 1992]

gss.loc[gss['year'] == 1993]

gss.loc[gss['year'] == 1994]

gss.loc[gss['year'] == 1995]

gss.loc[gss['year'] == 1996]

gss.loc[gss['year'] == 1997]

gss.loc[gss['year'] == 1998]

gss.loc[gss['year'] == 1999]

gss.loc[gss['year'] == 2000]

gss.loc[gss['year'] == 2001]

gss.loc[gss['year'] == 2002]

gss.loc[gss['year'] == 2003]

gss.loc[gss['year'] == 2004]

gss.loc[gss['year'] == 2005]

gss.loc[gss['year'] == 2006]

gss.loc[gss['year'] == 2007]

gss.loc[gss['year'] == 2008]

gss.loc[gss['year'] == 2009]

gss.loc[gss['year'] == 2010]

gss.loc[gss['year'] == 2011]

gss.loc[gss['year'] == 2012]

gss.loc[gss['year'] == 2013]

gss.loc[gss['year'] == 2014]

gss = gss[['year','race']]

gss.loc[gss['year'] == 2014][['year','race']].reset\_index(drop=True)

gss[['year',gss.index]].groupby('year').count().rename(columns={gss.index: 'Total'})

df= gss[['year','id']].groupby('year').count().rename(columns={'id': 'Total'})

plt.figure(figsize=(14,7)).suptitle('Graph of Number of Yearly participants from 1977 to 2014', fontsize=20)

plt.plot(df)

plt.xlabel('Years', fontsize=18)

plt.ylabel('Number of Participants', fontsize=18)

plt.savefig('Graphs/line\_graph\_of\_yearly\_participation.jpg')

plt.show()

df.index

plt.figure(figsize=(14,7)).suptitle('Graph of Number of Yearly participants from 1977 to 2014', fontsize=20)

plt.scatter(df.index, df[['Total']], cmap=cmap, edgecolor='None')

plt.xlabel('Years', fontsize=18)

plt.ylabel('Number of Participants', fontsize=18)

plt.savefig('Graphs/scatter\_graph\_of\_yearly\_participation.jpg')

plt.show()

df.index

import pandas as pd

import numpy as np

data = [

go.Bar(

x = df.index,

y = df['Total'],

orientation='v',

)

]

layout = go.Layout(

title='Graph of Number of Yearly participants from 1977 to 2014',

xaxis=dict(

title='Years',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

),

yaxis=dict(

title='Number of Participants',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

)

)

fig = go.Figure(data=data, layout=layout)

plt.show()

trace = go.Scatter(

x = df.index,

y = df['Total'],

)

layout = go.Layout(

title='Graph of Number of Yearly participants from 1977 to 2014',

xaxis=dict(

title='Years',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

),

yaxis=dict(

title='Number of Participants',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

)

)

fig = go.Figure(data=data, layout=layout)

plt.show()

trace = go.Scatter(

x = df.index,

y = df['Total'],

mode = 'markers',

name = 'markers'

)

layout = go.Layout(

title='Graph of Number of Yearly participants from 1977 to 2014',

xaxis=dict(

title='Years',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

),

yaxis=dict(

title='Number of Participants',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

)

)

fig = go.Figure(data=data, layout=layout)

plt.show()

trace = go.Scattergl(

x = df.index,

y = df['Total'],

mode = 'markers',

marker = dict(

line = dict(

width = 1,

color = '#404040')

)

)

layout = go.Layout(

title='Graph of Number of Yearly participants from 1977 to 2014',

xaxis=dict(

title='Years',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

),

yaxis=dict(

title='Number of Participants',

titlefont=dict(

family='Courier New, monospace',

size=18,

color='#7f7f7f'

)

)

)

fig = go.Figure(data=data, layout=layout)

plt.show()

trace = go.Table(

header=dict(values=list(df.columns),

fill = dict(color='#C2D4FF'),

align = ['left'] \* 5),

cells=dict(values=[df.index, df['Total']],

fill = dict(color='#F5F8FF'),

align = ['left'] \* 5))

data = [trace]

plt.show()

gss2014 = gss.loc[gss['year'] == 2014][['year','race']].reset\_index(drop=True)

gss2014 = gss2014[['year','race']].groupby('race').count()

gss2014

plt.figure(figsize=(14,7)).suptitle('SELF IDENTIFICATION OF RACE', fontsize=20)

plt.pie(gss2014['year'],

labels=['White', 'Black', 'Other'], autopct='%1.1f%%',explode = (0.1, 0.1, 0.1),

shadow=True, startangle=90

)

plt.axis('equal')

plt.tight\_layout()

plt.savefig('Graphs/pie\_chart\_of\_yearly\_participation.jpg')

plt.show()

trace = go.Pie(title='SELF IDENTIFICATION OF RACE', titlefont=dict(

family='Courier New, monospace',

size=36,

color='#7f7f7f'

),labels=['White', 'Black', 'Other'], values=gss2014['year'])

fig = go.Figure(data=data, layout=layout)

plt.show()

gss = pd.read\_csv('GSS7214.csv', delimiter=",")

gss2014 = gss.loc[gss['year'] == 2014][['year','racecen1']]

gss2014

gss2014 = gss2014[['year','racecen1']].groupby('racecen1').count()

gss2014

trace = go.Pie(title='SELF IDENTIFICATION OF RACE WITH MORE OPTIONS', titlefont=dict(

family='Courier New, monospace',

size=36,

color='#7f7f7f'

),labels=['WHITE', 'BLACK', 'AMERICAN INDIAN','ASIAN INDIAN', 'CHINESE', 'FILIPINO', 'JAPANESE', 'KOREAN', 'VIETNAMESE', 'OTHER ASIAN'

'NATIVE HAWAIIAN', 'GUAMANIAN', 'SAMOAN', 'OTHER PACIFIC ISLANDER', 'HISPANIC', 'NAP', 'DK', 'NA'], values=gss2014['year'])

fig = go.Figure(data=data, layout=layout)

plt.show()

gss2014 = gss.loc[gss['year'] == 2014][['year','sex']]

gss2014 = gss2014[['year','sex']].groupby('sex').count()

gss2014

trace = go.Pie(title='MALE OR FEMALE', titlefont=dict(

family='Courier New, monospace',

size=36,

color='#7f7f7f'

),labels=['MALE', 'FEMALE'], values=gss2014['year'])

fig = go.Figure(data=data, layout=layout)

plt.show()

gss2014 = gss.loc[gss['year'] == 2014][['year', 'age']]

gss2014 = gss2014[['year','age']].groupby('age').count()

gss2014

gss2014.index

data = [

go.Bar(

x = gss2014.index,

y = gss2014['year'],

orientation='v',

)

]

fig = go.Figure(data=data, layout=layout)

plt.show()

gss2014['year']

gss2014 = gss.loc[gss['year'] == 2014][['year', 'age']]

gss2014

data = [

go.Histogram(

x = gss2014['year']

)

]

fig = go.Figure(data=data, layout=layout)

url = py.offline.plot(fig, filename='Graphs/histogram\_of\_participants.html')

gss2014 = gss.loc[gss['year'] == 2014][['year', 'marhomo']]

gss2014 = gss2014[['year','marhomo']].groupby('marhomo').count()

gss2014

trace = go.Pie(title='SAME SEX MARRIAGE OPINION', titlefont=dict(

family='Courier New, monospace',

size=36,

color='#7f7f7f'

),labels=['NAP','STRONGLY AGREE','AGREE', 'NEITHER AGREE NOR DISAGREE','DISAGREE', 'STRONGLY DISAGREE',

'CANT CHOOSE', 'N/A'], values=gss2014['year'])

fig = go.Figure(data=data, layout=layout)

plt.show()

gss2014 = gss.loc[gss['year'] == 2014][['year', 'jobfind']]

gss2014 = gss2014[['year','jobfind']].groupby('jobfind').count()

gss2014

trace = go.Pie(title='DIFFICULTY OF FINDING JOB', titlefont=dict(

family='Courier New, monospace',

size=36,

color='#7f7f7f'

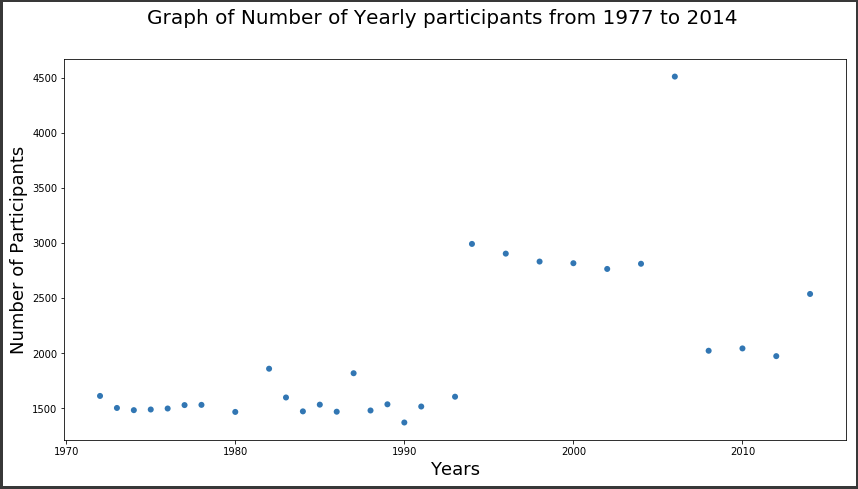
),labels=['IAP','VERY EASY','SOMEWHAT EASY', 'NOT EASY','DK', 'N/A',

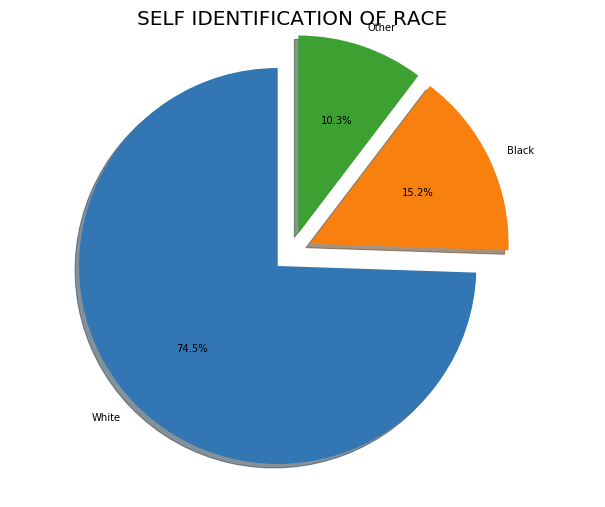
'CANT CHOOSE', 'N/A'], values=gss2014['year'])

fig = go.Figure(data=data, layout=layout)

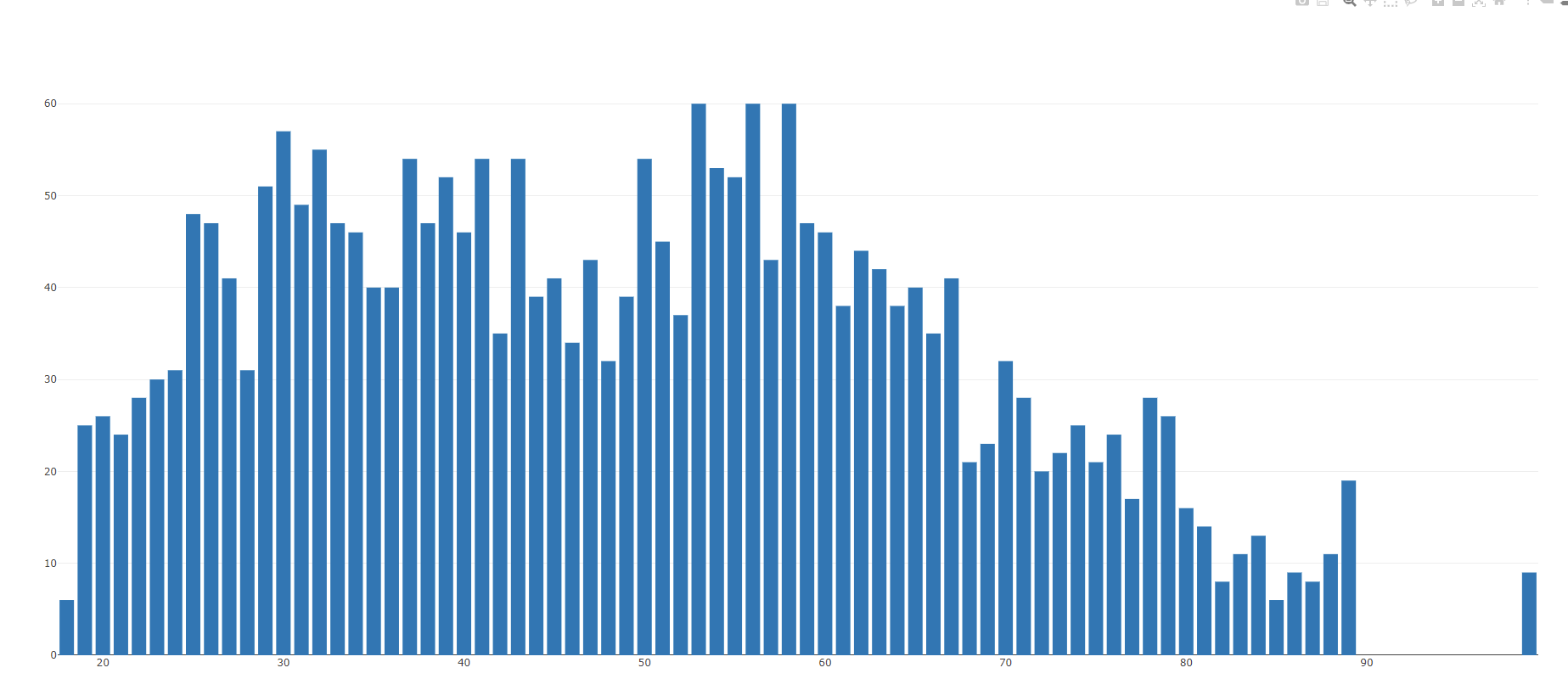
plt.show()

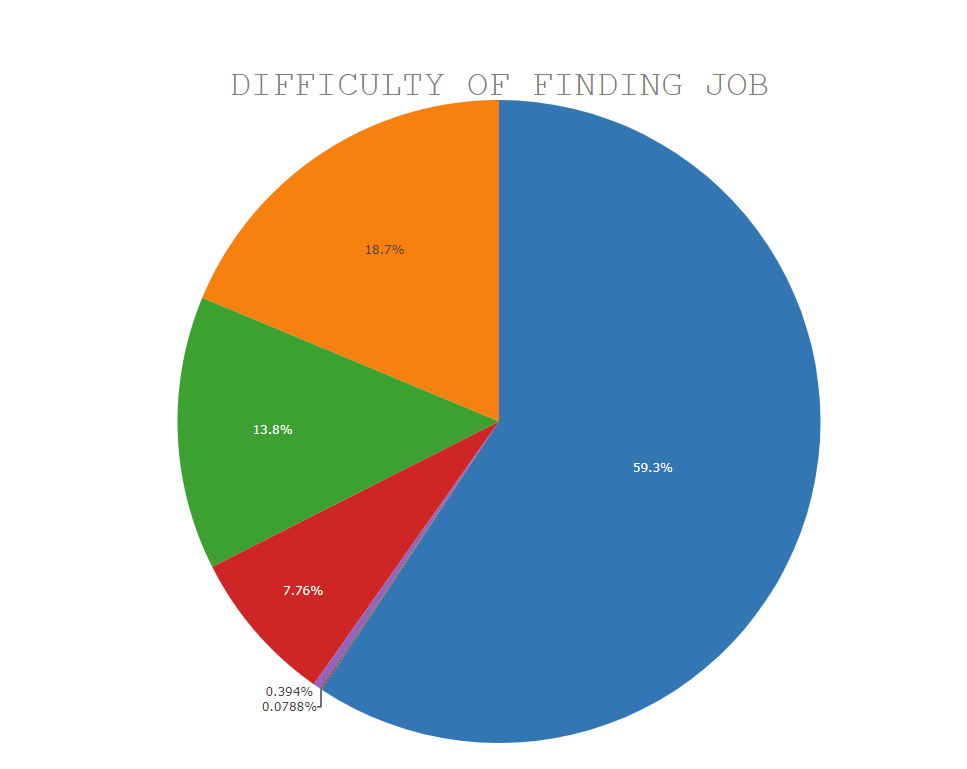
**Output Screenshots -**

****

****

**bar\_graph\_of\_participants**

****

****

***Q2) Sadness Analysis***

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import math

from scipy.stats import ttest\_1samp

from scipy import stats

from sklearn.preprocessing import LabelEncoder

from sklearn.feature\_selection import chi2

np.random.seed(6)

import warnings

warnings.filterwarnings('ignore')

sadness\_df = pd.read\_csv('Sadness.csv')

print("df is \n",sadness\_df)

sadness\_df['Total\_people\_surveyed'] = sadness\_df['All\_or\_most\_of\_the\_time'] + sadness\_df['Some\_of\_the\_time']

print("Changed df is \n",sadness\_df)

print("Number of columns in dataset",len(list(sadness\_df.columns)))

columns\_list = list(sadness\_df.columns)

print("The columns in the Dataset are \n")

for col in columns\_list:

print(col)

sadness\_numerical\_df = sadness\_df.select\_dtypes(include = 'number', exclude = None)

print("Numerical Data in the Dataset is \n",sadness\_numerical\_df)

n\_of\_total = sum(list(sadness\_df['Total\_people\_surveyed']))

print("The total number of people surveyed is ",n\_of\_total)

tset, pval = ttest\_1samp(sadness\_numerical\_df, 30)

print("Inducing significance level on the Probability Values in the data")

print("Type of pval ",type(pval))

print("Probability values p-values",pval)

significance\_level = 0.01

if pval.any() < significance\_level: # alpha value (significance value) is 0.01 or 1%

print(" we are rejecting null hypothesis")

else:

print("we are accepting null hypothesis")

print("\n")

print(" ----------------------------------------------------------------------------------- \n")

print("The Degree of sadness has been calculated for each of the population as X/total(column) \n, where X is the count of current column and total(column) is sum of all the values in the column \n\n")

n\_for\_allormost = sum(list(sadness\_df['All\_or\_most\_of\_the\_time']))

n\_for\_some = sum(list(sadness\_df['Some\_of\_the\_time']))

print("Total number of people surveyed as sad for all or most of the time ",n\_for\_allormost)

print("Total number of people surveyed as sad for some of the time ",n\_for\_some)

print("Preprocessing the counts for people who are sad for all or most of the time ... \n\n")

sadness\_df['Degree\_of\_sadness\_allormost'] = sadness\_df['All\_or\_most\_of\_the\_time']/n\_for\_allormost

sadness\_df['Degree\_of\_sadness\_some'] = sadness\_df['Some\_of\_the\_time']/n\_for\_some

sadness\_df['Total\_Degree\_of\_sadness'] = sadness\_df['Total\_people\_surveyed']/n\_of\_total

print("Changed DataFrame after applying Degree of Sadness \n")

print(sadness\_df)

sadness\_df\_cor = sadness\_df.corr(method = 'kendall')

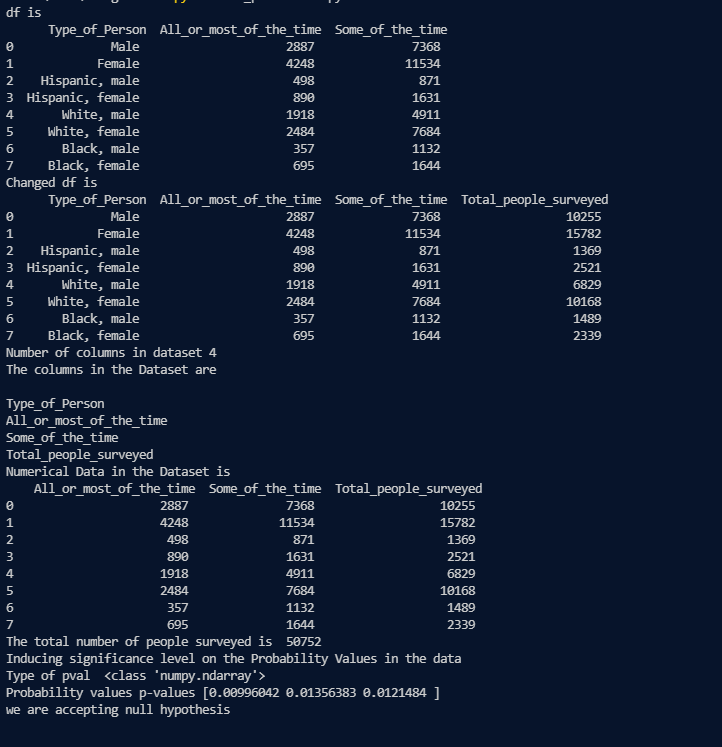
ax = sns.heatmap(sadness\_df\_cor, annot=True)

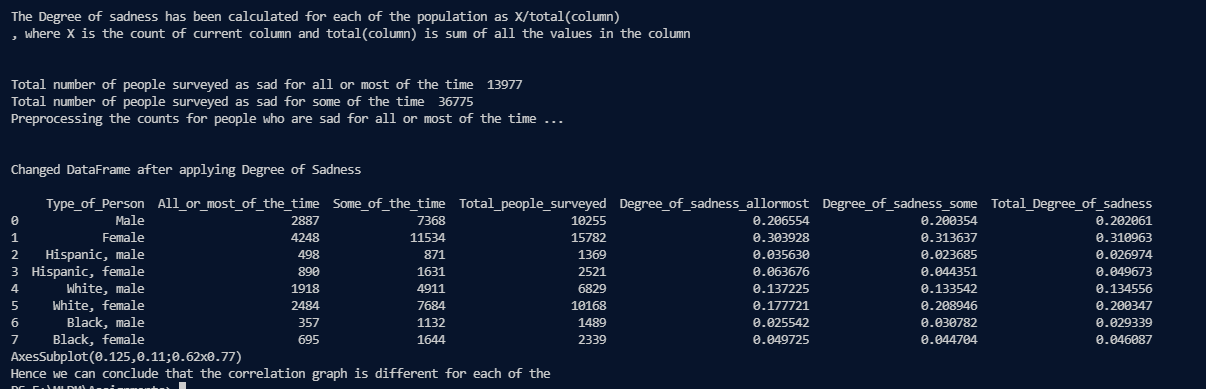
print(ax)

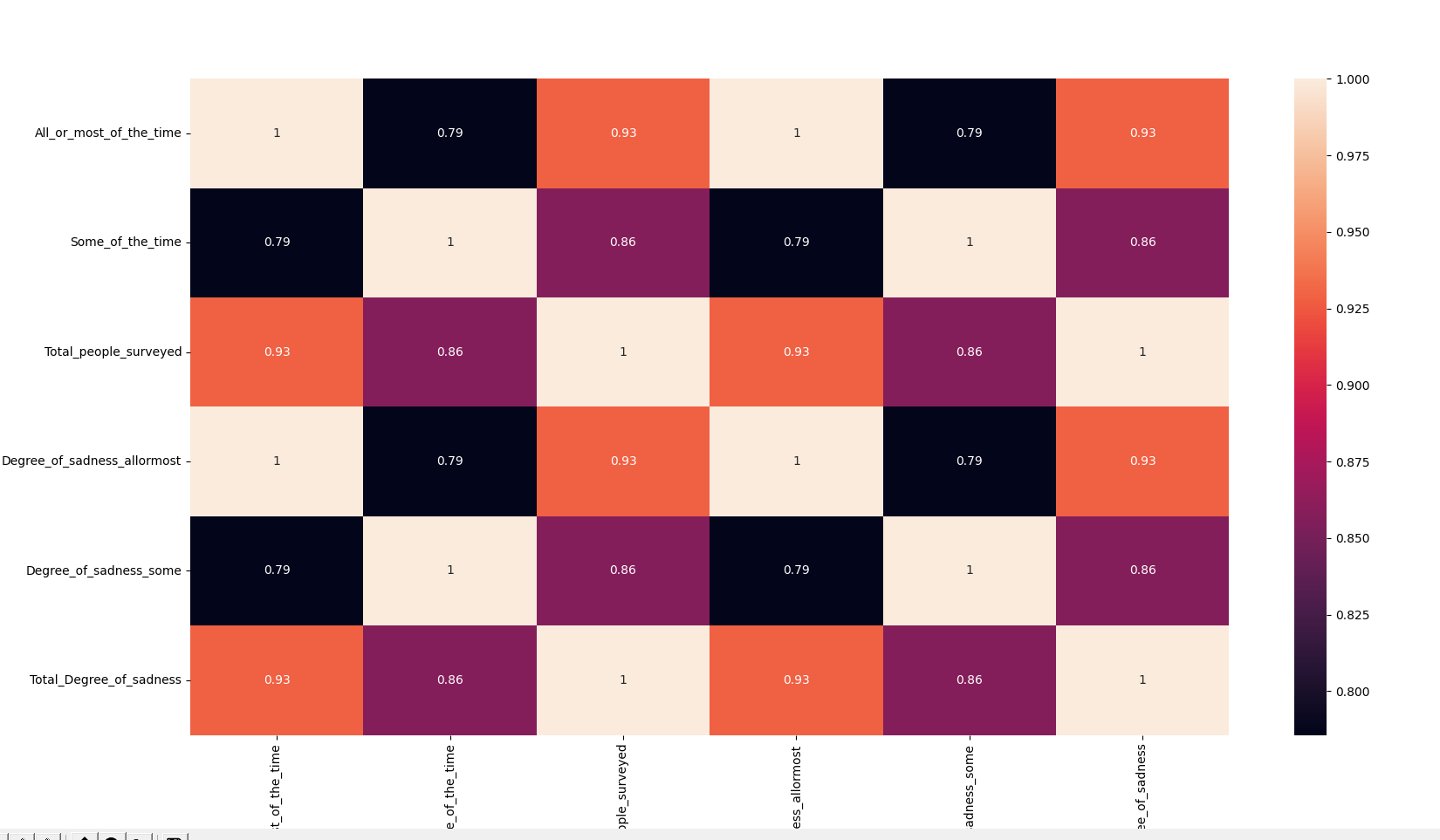
plt.show()

print("Hence we can conclude that the correlation graph is different for each of the ")

**Outputs -**

****

****

****

**Thus the degree of sadness appear to be dependent of gender in some ethnicities but not**

**Others because correlation between attributes using Kendall Statistics differs for them.**