Final_Project-FINALSLIDES-FINALCOPY

January 11, 2023

1 Disadvantages and Rates of Diseases in the Greater Toronto Area

Arushi, Jiangyuan, Hyunbeen, Bibelta

TUT0301

Group 4

2 Intro/Data

2.1 Question: Do Disadvantages Affect the Rates of Diseases in the Greater Toronto Area?

[6]:
```
%pip install xlrd
```

```
Requirement already satisfied: xlrd in /opt/conda/lib/python3.8/site-packages
(2.0.1)
Note: you may need to restart the kernel to use updated packages.
```

2.2 Clean DataFrame:

[7]:
```
import pandas as pd

ahd_hbp = pd.read_excel('1_ahd_neighb_db_ast_hbp_mhv_copd_2007-Copy1.xls',
                                sheet_name = '1_ahd_neighb_hbp_2007',
                                header = 11)

ahd_diabetes = pd.read_excel('1_ahd_neighb_db_ast_hbp_mhv_copd_2007-Copy1.xls',
                                sheet_name = '1_ahd_neighb_diabetes_2007',
                                header = 12)

ahd_copd = pd.read_excel('1_ahd_neighb_db_ast_hbp_mhv_copd_2007-Copy1.xls',
                                sheet_name = '1_ahd_neighb_copd_2007',
                                header = 11)

socdem_raw = pd.read_excel('1_socdem_neighb_2006-2-Copy1.xls',
                                sheet_name = 'socdem_2006',
                                header = 10)
```

[8]:
```
hbp = ahd_hbp[ahd_hbp.columns[[1, 11]]]

diabetes = ahd_diabetes[ahd_diabetes.columns[[1, 11]]]

copd = ahd_copd[ahd_copd.columns[[1, 11]]]

socdem = socdem_raw[socdem_raw.columns[[1, 7, 10, 24]]]
```

[9]:
```
colnames = {'Unnamed: 1': 'Neighb_Name',
            '% With high blood pressure.2' : 'percent_HBP'}

hbp = hbp.copy()
hbp.rename(columns = colnames, inplace = True)

colnames1 = {'Unnamed: 1': 'Neighb_Name',
             '% With diabetes.2' : 'percent_Diabetes'}

diabetes = diabetes.copy()
diabetes.rename(columns = colnames1, inplace = True)

colnames2 = {'Unnamed: 1': 'Neighb_Name',
             '% With COPD.2' : 'percent_COPD'}

copd = copd.copy()
copd.rename(columns = colnames2, inplace = True)

colnames3 = {'Neighbourhood Name': 'Neighb_Name',
             '% Lone parent families §' : 'percent_Lone_Parent_Families',
```

```
            'Median household income after-tax $ ‡' :␣
 ↪'Median_Household_Income',
            '% Visible minority' : 'percent_Visible_Minority'}

socdem = socdem.copy()
socdem.rename(columns = colnames3, inplace = True)
```

[10]:
```
two_diseases = hbp.merge(diabetes, left_on='Neighb_Name',␣
 ↪right_on='Neighb_Name')
three_diseases = two_diseases.merge(copd, left_on='Neighb_Name',␣
 ↪right_on='Neighb_Name')
clean_data = three_diseases.merge(socdem, left_on='Neighb_Name',␣
 ↪right_on='Neighb_Name')
clean_data.style.set_caption('Disease Types, Social-demograpic variables and␣
 ↪Neighbourhood information')
```

[10]:
```
<pandas.io.formats.style.Styler at 0x7f4e37059970>
```

3 Methods

3.1 Data Wrangling and Extraction Methods

We extracted our datasets from the excel files and then extracted columns of importance. We merged our four datasets and named the new dataset: `clean_data`

3.2 Quantiles Methods

For each social demographic variable, we created 5 separate dataframes based upon the corresponding variable quantiles. We set the quantile conditions, and repeated the same process for the other two sociodemographic variables.

3.3 Bar Charts Methods

We graphed 5 quantiles in three bar charts. Each bar chart included the three sociodemographic variables and one respective disease. The value of each quantile corresponded to the mean disease rate.

3.4 Multiple Linear Regression Methods

For our Multiple Linear Regression model, our independent variables were Lone Parent Families , Median Household Income, and Percent Visibl Minority. Our dependent variables were Percent HBP, Percent COPD, and Percent Diabetes. We performed MLR three times upon the entire dataset, once for each dependent variable.

3.5 P-value DataFrames Methods

We computed the difference between the means of the lowest income quartile and highest income quartile, and performed a hypothesis test using p-values, to conclude whether the difference between

means was due to random chance or not.

4 Results

4.1 Bar Charts

In terms of income, there is a negative relationship between quantile number and disease rate.

[11]:
```
Q1 = clean_data.loc[clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.2),]

Quantile_20_40 = (clean_data['percent_Lone_Parent_Families'].quantile(0.2) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.4))
Q2 = clean_data.loc[Quantile_20_40,]

Quantile_40_60 = (clean_data['percent_Lone_Parent_Families'].quantile(0.4) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.6))
Q3 = clean_data.loc[Quantile_40_60,]

Quantile_60_80 = (clean_data['percent_Lone_Parent_Families'].quantile(0.6) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.8))
Q4 = clean_data.loc[Quantile_60_80,]

Quantile_80_100 = (clean_data['percent_Lone_Parent_Families'].quantile(0.8) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(1))
Q5 = clean_data.loc[Quantile_80_100,]
```

[12]:
```
q1 = clean_data.loc[clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.2),]

quantile_20_40 = (clean_data['percent_Visible_Minority'].quantile(0.2) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.4))
q2 = clean_data.loc[quantile_20_40,]
```

```
quantile_40_60 = (clean_data['percent_Visible_Minority'].quantile(0.4) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.6))
q3 = clean_data.loc[quantile_40_60,]

quantile_60_80 = (clean_data['percent_Visible_Minority'].quantile(0.6) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.8))
q4 = clean_data.loc[quantile_60_80,]

quantile_80_100 = (clean_data['percent_Visible_Minority'].quantile(0.8) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(1))
q5 = clean_data.loc[quantile_80_100,]
```

[13]:
```
qq1 = clean_data.loc[clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.2),]

qquantile_20_40 = (clean_data['Median_Household_Income'].quantile(0.2) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.4))
qq2 = clean_data.loc[qquantile_20_40,]

qquantile_40_60 = (clean_data['Median_Household_Income'].quantile(0.4) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.6))
qq3 = clean_data.loc[qquantile_40_60,]

qquantile_60_80 = (clean_data['Median_Household_Income'].quantile(0.6) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.8))
qq4 = clean_data.loc[qquantile_60_80,]

qquantile_80_100 = (clean_data['Median_Household_Income'].quantile(0.8) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(1))
qq5 = clean_data.loc[qquantile_80_100,]
```

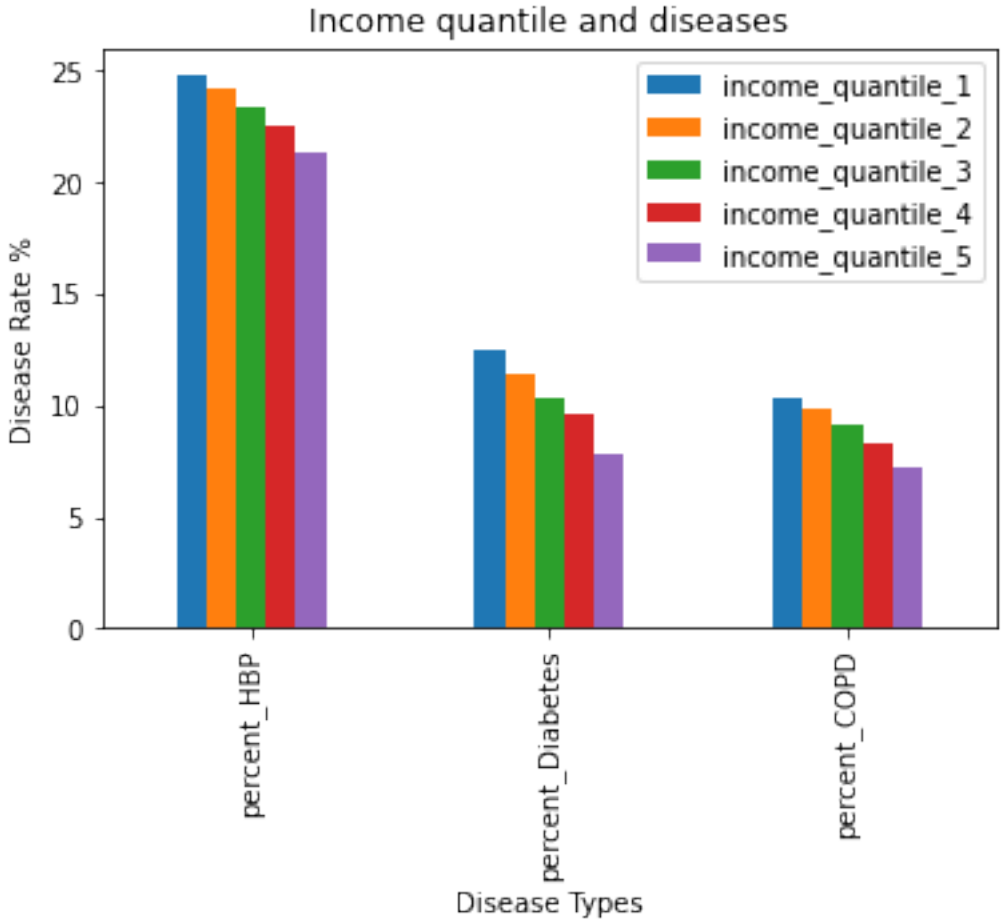
[14]:
```
q1 = clean_data.loc[clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.2),]

quantile_20_40 = (clean_data['percent_Visible_Minority'].quantile(0.2) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.4))
q2 = clean_data.loc[quantile_20_40,]

quantile_40_60 = (clean_data['percent_Visible_Minority'].quantile(0.4) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.6))
q3 = clean_data.loc[quantile_40_60,]

quantile_60_80 = (clean_data['percent_Visible_Minority'].quantile(0.6) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.8))
q4 = clean_data.loc[quantile_60_80,]

quantile_80_100 = (clean_data['percent_Visible_Minority'].quantile(0.8) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(1))
q5 = clean_data.loc[quantile_80_100,]

Q1 = clean_data.loc[clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.2),]

Quantile_20_40 = (clean_data['percent_Lone_Parent_Families'].quantile(0.2) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.4))
Q2 = clean_data.loc[Quantile_20_40,]

Quantile_40_60 = (clean_data['percent_Lone_Parent_Families'].quantile(0.4) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.6))
Q3 = clean_data.loc[Quantile_40_60,]

Quantile_60_80 = (clean_data['percent_Lone_Parent_Families'].quantile(0.6) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.8))
```

```
Q4 = clean_data.loc[Quantile_60_80,]

Quantile_80_100 = (clean_data['percent_Lone_Parent_Families'].quantile(0.8) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(1))
Q5 = clean_data.loc[Quantile_80_100,]

qq1 = clean_data.loc[clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.2),]

qquantile_20_40 = (clean_data['Median_Household_Income'].quantile(0.2) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.4))
qq2 = clean_data.loc[qquantile_20_40,]

qquantile_40_60 = (clean_data['Median_Household_Income'].quantile(0.4) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.6))
qq3 = clean_data.loc[qquantile_40_60,]

qquantile_60_80 = (clean_data['Median_Household_Income'].quantile(0.6) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.8))
qq4 = clean_data.loc[qquantile_60_80,]

qquantile_80_100 = (clean_data['Median_Household_Income'].quantile(0.8) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(1))
qq5 = clean_data.loc[qquantile_80_100,]


diseases = ['percent_HBP','percent_Diabetes','percent_COPD']

income_quantile = {'income_quantile_1': qq1[diseases].mean(),
                   'income_quantile_2': qq2[diseases].mean(),
                   'income_quantile_3': qq3[diseases].mean(),
                   'income_quantile_4': qq4[diseases].mean(),
                   'income_quantile_5': qq5[diseases].mean()}
```

[15]:

```
income_disease = pd.DataFrame(income_quantile)
income_disease.plot.bar(xlabel='Disease Types', ylabel='Disease Rate %', title␣
↪= 'Income quantile and diseases')
```

[15]:
```
<AxesSubplot:title={'center':'Income quantile and diseases'}, xlabel='Disease
Types', ylabel='Disease Rate %'>
```



[ ]:
```

```

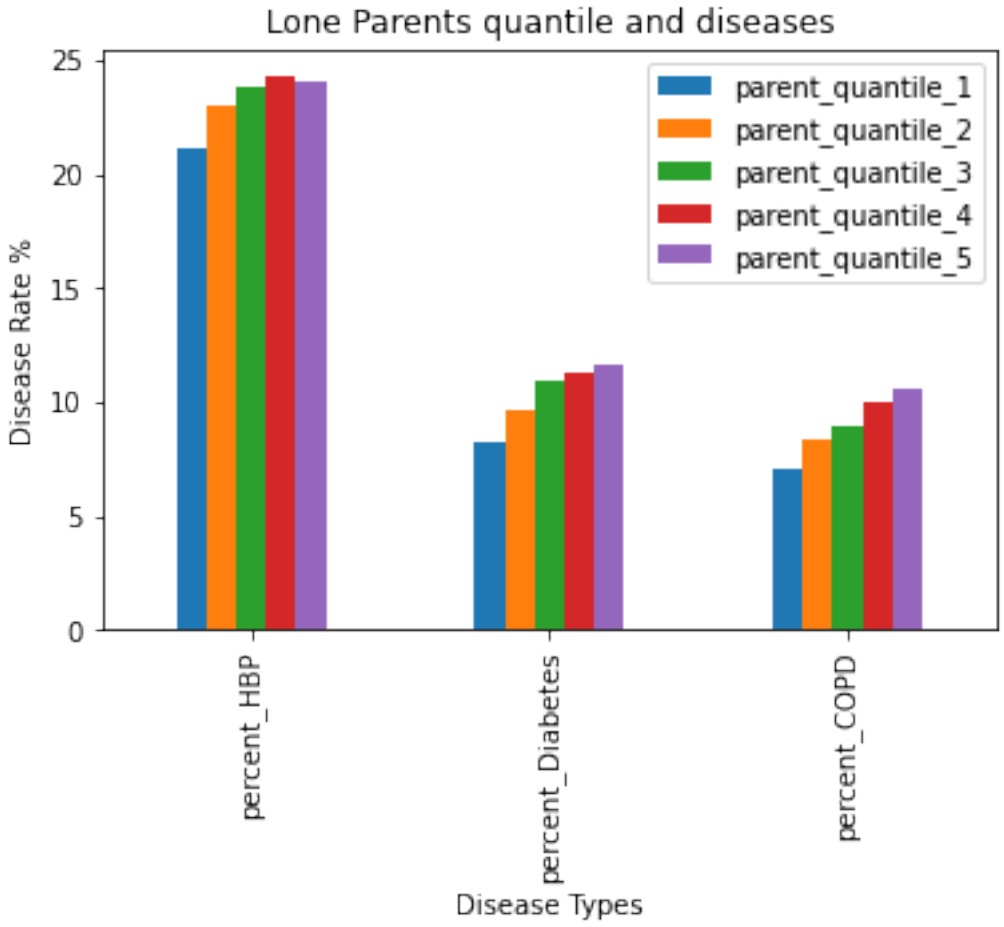
In terms of Lone Parent Percentage, there is a positive relationship between quantile number and disease rate.

[16]:
```
diseases = ['percent_HBP','percent_Diabetes','percent_COPD']

lone_parent_quantile = {'parent_quantile_1': Q1[diseases].mean(),
                        'parent_quantile_2': Q2[diseases].mean(),
                        'parent_quantile_3': Q3[diseases].mean(),
                        'parent_quantile_4': Q4[diseases].mean(),
```

```
                    'parent_quantile_5': Q5[diseases].mean()}
lone_parent_disease = pd.DataFrame(lone_parent_quantile)
```

[17]:
```
#vismin_disease['quantile'] = [1,2,3,4,5]
lone_parent_disease.plot.bar(xlabel='Disease Types', ylabel='Disease Rate %',␣
 ↪title = 'Lone Parents quantile and diseases')
```

[17]:
```
<AxesSubplot:title={'center':'Lone Parents quantile and diseases'},
xlabel='Disease Types', ylabel='Disease Rate %'>
```



In terms of Visible Minority Percentage, there is a positive relationship between quantile number and disease rate.

[20]:
```
vismin_disease.transpose().plot.bar(xlabel='Disease Types', ylabel='Disease␣
 ↪Rate %', title = 'Visible Minority quantile and diseases')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
```

```
Input In [20], in <cell line: 1>()
----> 1 vismin_disease.transpose().plot.bar(xlabel='Disease Types',␣
 ↪ylabel='Disease Rate %', title = 'Visible Minority quantile and diseases')

NameError: name 'vismin_disease' is not defined
```

4.2 Multiple Linear Regression

Our p values for each variable in each model is less than 0.1

Model 1: Dependent Variable - Percent HBP

[21]:
```
from statsmodels.formula.api import ols

regmod1 = ols('percent_HBP ~ percent_Lone_Parent_Families +␣
 ↪Median_Household_Income + percent_Visible_Minority', data = clean_data)
regmod_fit1 = regmod1.fit()
regmod_fit1.params
import warnings
warnings.filterwarnings('ignore') # turn off warnings
```

[22]:
```
from statsmodels.formula.api import ols

regmod1 = ols('percent_HBP ~ percent_Lone_Parent_Families +␣
 ↪Median_Household_Income + percent_Visible_Minority', data = clean_data)
regmod_fit1 = regmod1.fit()
regmod_fit1.params
import warnings
warnings.filterwarnings('ignore') # turn off warnings
```

[23]:
```
regsum1 = regmod_fit1.summary()
regsum1.tables[1]
```

[23]:
```
<class 'statsmodels.iolib.table.SimpleTable'>
```

Model 2: Dependent Variable - Percent COPD

[24]:
```
regmod2 = ols('percent_COPD ~ percent_Lone_Parent_Families +␣
 ↪Median_Household_Income + percent_Visible_Minority', data = clean_data)
regmod_fit2 = regmod2.fit()
regmod_fit2.params

warnings.filterwarnings('ignore') # turn off warnings
```

[25]:
```
regmod2 = ols('percent_COPD ~ percent_Lone_Parent_Families +␣
 ↪Median_Household_Income + percent_Visible_Minority', data = clean_data)
regmod_fit2 = regmod2.fit()
```

```
regmod_fit2.params

warnings.filterwarnings('ignore') # turn off warnings
```

[26]:
```
regsum2 = regmod_fit2.summary()
regsum2.tables[1]
```

[26]:
```
<class 'statsmodels.iolib.table.SimpleTable'>
```

Model 3: Dependent Variable - Percent Diabetes

[27]:
```
regmod3 = ols('percent_Diabetes ~ percent_Lone_Parent_Families +␣
 ↪Median_Household_Income + percent_Visible_Minority', data = clean_data)
regmod_fit3 = regmod3.fit()
regmod_fit3.params

warnings.filterwarnings('ignore') # turn off warnings
```

[28]:
```
regmod3 = ols('percent_Diabetes ~ percent_Lone_Parent_Families +␣
 ↪Median_Household_Income + percent_Visible_Minority', data = clean_data)
regmod_fit3 = regmod3.fit()
regmod_fit3.params

warnings.filterwarnings('ignore') # turn off warnings
```

[29]:
```
regsum3 = regmod_fit3.summary()
regsum3.tables[1]
```

[29]:
```
<class 'statsmodels.iolib.table.SimpleTable'>
```

4.3 Hypothesis Testing with T-tests and P-values

All of our p values except for Visible Minority and COPD Rates are less than 0.1

[30]:
```
q1 = clean_data.loc[clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.2),]

quantile_20_40 = (clean_data['percent_Visible_Minority'].quantile(0.2) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.4))
q2 = clean_data.loc[quantile_20_40,]

quantile_40_60 = (clean_data['percent_Visible_Minority'].quantile(0.4) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.6))
```

```
q3 = clean_data.loc[quantile_40_60,]

quantile_60_80 = (clean_data['percent_Visible_Minority'].quantile(0.6) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.8))
q4 = clean_data.loc[quantile_60_80,]

quantile_80_100 = (clean_data['percent_Visible_Minority'].quantile(0.8) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(1))
q5 = clean_data.loc[quantile_80_100,]


Q1 = clean_data.loc[clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.2),]

Quantile_20_40 = (clean_data['percent_Lone_Parent_Families'].quantile(0.2) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.4))
Q2 = clean_data.loc[Quantile_20_40,]

Quantile_40_60 = (clean_data['percent_Lone_Parent_Families'].quantile(0.4) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.6))
Q3 = clean_data.loc[Quantile_40_60,]

Quantile_60_80 = (clean_data['percent_Lone_Parent_Families'].quantile(0.6) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.8))
Q4 = clean_data.loc[Quantile_60_80,]

Quantile_80_100 = (clean_data['percent_Lone_Parent_Families'].quantile(0.8) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(1))
Q5 = clean_data.loc[Quantile_80_100,]


qq1 = clean_data.loc[clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.2),]
```

```
qquantile_20_40 = (clean_data['Median_Household_Income'].quantile(0.2) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.4))
qq2 = clean_data.loc[qquantile_20_40,]

qquantile_40_60 = (clean_data['Median_Household_Income'].quantile(0.4) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.6))
qq3 = clean_data.loc[qquantile_40_60,]

qquantile_60_80 = (clean_data['Median_Household_Income'].quantile(0.6) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.8))
qq4 = clean_data.loc[qquantile_60_80,]

qquantile_80_100 = (clean_data['Median_Household_Income'].quantile(0.8) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(1))
qq5 = clean_data.loc[qquantile_80_100,]
```

[31]:
```
from scipy import stats

stats, diabetes_vismin = stats.ttest_ind(q5['percent_Diabetes'],␣
 ↪q1['percent_Diabetes'])

from scipy import stats

stats, diabetes_income = stats.ttest_ind(qq5['percent_Diabetes'],␣
 ↪qq1['percent_Diabetes'])

from scipy import stats

stats, diabetes_lone_parents = stats.ttest_ind(Q5['percent_Diabetes'],␣
 ↪Q1['percent_Diabetes'])

from scipy import stats
stats, HBP_vismin = stats.ttest_ind(q5['percent_HBP'], q1['percent_HBP'])

from scipy import stats
stats, HBP_income = stats.ttest_ind(qq5['percent_HBP'], qq1['percent_HBP'])

from scipy import stats
```

```
stats, HBP_lone_parents = stats.ttest_ind(Q5['percent_HBP'], Q1['percent_HBP'])

from scipy import stats
stats, COPD_vismin = stats.ttest_ind(q5['percent_COPD'], q1['percent_COPD'])

from scipy import stats
stats, COPD_income = stats.ttest_ind(qq5['percent_COPD'], qq1['percent_COPD'])

from scipy import stats
stats, COPD_lone_parents = stats.ttest_ind(Q5['percent_COPD'],␣
 ↪Q1['percent_COPD'])
```

[32]:
```
pvalue = {
    'HBP':{
        'Visible Minority': HBP_vismin.round(5),
        'Median Income': HBP_income.round(5),
        'Lone Parents': HBP_lone_parents.round(5)
    },
    'COPD':{
        'Visible Minority': COPD_vismin.round(5),
        'Median Income': COPD_income.round(5),
        'Lone Parents': COPD_lone_parents.round(5)
    },
    'Diabetes':{
        'Visible Minority': diabetes_vismin.round(5),
        'Median Income': diabetes_income.round(5),
        'Lone Parents': diabetes_lone_parents.round(5)
    }
}
```

[33]:
```
pvalue_differences = pd.DataFrame(pvalue)
pvalue_differences.style.set_caption('P-values: differences between highest and␣
 ↪lowest quantile neighbourhood')
```

[33]:
```
<pandas.io.formats.style.Styler at 0x7f4e6b42e730>
```

5 Conclusions

5.1 Restate the question:

Do Disadvantages Affect the Rates of Diseases in the Greater Toronto Area?

Our goal was to depict disadvantages such as income, single parent-family and being a visible minority, and observe their influences on three diseases

5.2 Interpretations

5.2.1 Bar Charts

- The relationship between Lone Parents and the three diseases is positive
  - Neighbourhoods with higher rates of lone parents tend to have higher rates of disease
- The relationship between Visible Minority Rate and diseases is positive
  - Note: For COPD, there is not a specific relationship
- The relationship between Income and the three diseases is negative
  - Neighbourhoods with higher income tend to have lower rates of disease

5.2.2 Multiple Linear Regression

- For all variables in all models p $<$ .1
  - We are 90 % confident that the slope is not 0
    - There is a predictive relationship between each independent variable and rate of respective disease

Model 1: P values: - percent_Visible_Minority - .000 - Median_Household_Income - .079 - percent_Lone_Parent_Families - .058

Model 2: P values: - percent_Visible_Minority - .000 - Median_Household_Income - .001 - percent_Lone_Parent_Families - .000

Model 3: P values: - percent_Visible_Minority - .000 - Median_Household_Income - .016 - percent_Lone_Parent_Families - .005

5.2.3 Hypothesis Testing with T-tests and P-values

Null Hypotheses

1. The difference between the means of high blood pressure rate in lower and higher income neighbourhoods is due to chance
2. The difference between means of COPD rate in lower and higher income neighbourhoods is due to chance
3. The difference between means of Diabetes rate in lower and higher income neighbourhoods is due to chance

- All p-values, except for Visible Minority and COPD rates, are less than 0.1
  - we reject the null hypothesis
- We will not reject the null hypothesis for Visible Minority and COPD rates
  - p value is .815320 $>$ .1

[34]:
```
q1 = clean_data.loc[clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.2),]

quantile_20_40 = (clean_data['percent_Visible_Minority'].quantile(0.2) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.4))
```

```
q2 = clean_data.loc[quantile_20_40,]

quantile_40_60 = (clean_data['percent_Visible_Minority'].quantile(0.4) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.6))
q3 = clean_data.loc[quantile_40_60,]

quantile_60_80 = (clean_data['percent_Visible_Minority'].quantile(0.6) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(0.8))
q4 = clean_data.loc[quantile_60_80,]

quantile_80_100 = (clean_data['percent_Visible_Minority'].quantile(0.8) <␣
 ↪clean_data['percent_Visible_Minority']) &␣
 ↪(clean_data['percent_Visible_Minority'] <=␣
 ↪clean_data['percent_Visible_Minority'].quantile(1))
q5 = clean_data.loc[quantile_80_100,]
```

[35]:
```
Q1 = clean_data.loc[clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.2),]

Quantile_20_40 = (clean_data['percent_Lone_Parent_Families'].quantile(0.2) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.4))
Q2 = clean_data.loc[Quantile_20_40,]

Quantile_40_60 = (clean_data['percent_Lone_Parent_Families'].quantile(0.4) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.6))
Q3 = clean_data.loc[Quantile_40_60,]

Quantile_60_80 = (clean_data['percent_Lone_Parent_Families'].quantile(0.6) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(0.8))
Q4 = clean_data.loc[Quantile_60_80,]

Quantile_80_100 = (clean_data['percent_Lone_Parent_Families'].quantile(0.8) <␣
 ↪clean_data['percent_Lone_Parent_Families']) &␣
 ↪(clean_data['percent_Lone_Parent_Families'] <=␣
 ↪clean_data['percent_Lone_Parent_Families'].quantile(1))
Q5 = clean_data.loc[Quantile_80_100,]
```

[51]:
```
qq1 = clean_data.loc[clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.2),]

qquantile_20_40 = (clean_data['Median_Household_Income'].quantile(0.2) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.4))
qq2 = clean_data.loc[qquantile_20_40,]

qquantile_40_60 = (clean_data['Median_Household_Income'].quantile(0.4) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.6))
qq3 = clean_data.loc[qquantile_40_60,]

qquantile_60_80 = (clean_data['Median_Household_Income'].quantile(0.6) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(0.8))
qq4 = clean_data.loc[qquantile_60_80,]

qquantile_80_100 = (clean_data['Median_Household_Income'].quantile(0.8) <␣
 ↪clean_data['Median_Household_Income']) &␣
 ↪(clean_data['Median_Household_Income'] <=␣
 ↪clean_data['Median_Household_Income'].quantile(1))
qq5 = clean_data.loc[qquantile_80_100,]
```

[52]:
```
from scipy import stats

stats, diabetes_vismin = stats.ttest_ind(q5['percent_Diabetes'],␣
 ↪q1['percent_Diabetes'])
```

6 Limitations

1. No possible explanation for the little difference between COPD rates in the neighbourhood with the highest visible minority and the lowest
2. It would be better if we used maps to compare patterns of each socio-demographic variable and the disease rates
3. Only analyzed the effect of three variables:
   - income
   - being a visible minority
   - being a lone parent

7 Findings

- Income, Lone Parent Family Percentage ,and Visible Minority Percentage are all highly related to disease rates
- Disadvantaged neighbourhoods are more susceptible to disease based upon chosen independent variables

In conclusion, disadvantaged neighbourhoods are more susceptible to disease, to an extent, because not all diseases are a result of external pressures like Income, being a Visible Minority or being a Lone Parent.

[ ]:
```
from scipy import stats

stats, diabetes_income = stats.ttest_ind(qq5['percent_Diabetes'],␣
↪qq1['percent_Diabetes'])
```

[43]:
```
from scipy import stats

stats, diabetes_lone_parents = stats.ttest_ind(Q5['percent_Diabetes'],␣
↪Q1['percent_Diabetes'])
```

[ ]:
```

```

[44]:
```
from scipy import stats
stats, HBP_vismin = stats.ttest_ind(q5['percent_HBP'], q1['percent_HBP'])
```

[45]:
```
from scipy import stats
stats, HBP_income = stats.ttest_ind(qq5['percent_HBP'], qq1['percent_HBP'])
```

[46]:
```
from scipy import stats
stats, HBP_lone_parents = stats.ttest_ind(Q5['percent_HBP'], Q1['percent_HBP'])
```

[ ]:
```

```

[47]:
```
from scipy import stats
stats, COPD_vismin = stats.ttest_ind(q5['percent_COPD'], q1['percent_COPD'])
```

[48]:
```
from scipy import stats
stats, COPD_income = stats.ttest_ind(qq5['percent_COPD'], qq1['percent_COPD'])
```

[49]:
```
from scipy import stats
stats, COPD_lone_parents = stats.ttest_ind(Q5['percent_COPD'],␣
↪Q1['percent_COPD'])
```

[ ]:
```

```

[50]:
```
pvalue = {
    'HBP':{
```

```
        'Visible Minority': HBP_vismin.round(5),
        'Median Income': HBP_income.round(5),
        'Lone Parents': HBP_lone_parents.round(5)
    },
    'COPD':{
        'Visible Minority': COPD_vismin.round(5),
        'Median Income': COPD_income.round(5),
        'Lone Parents': COPD_lone_parents.round(5)
    },
    'Diabetes':{
        'Visible Minority': diabetes_vismin.round(5),
        'Median Income': diabetes_income.round(5),
        'Lone Parents': diabetes_lone_parents.round(5)
    }
}
```