The Difference of Income Levels and its affect on the Percentage of Obese Adults (1)

April 29, 2024

```
[53]: #Import and Install all packages needed for analysis
      import pandas as pd
```

```
[54]: import os
```

```
[55]: pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\mandy\anaconda3\lib\site-
packages (3.7.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: packaging>=20.0 in
c:\users\mandy\appdata\roaming\python\python39\site-packages (from matplotlib)
(23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: importlib-resources>=3.2.0 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (5.12.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\mandy\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: numpy>=1.20 in
c:\users\mandy\appdata\roaming\python\python39\site-packages (from matplotlib)
(1.23.5)
Requirement already satisfied: zipp>=3.1.0 in
c:\users\mandy\appdata\roaming\python\python39\site-packages (from importlib-
resources>=3.2.0->matplotlib) (3.15.0)
Requirement already satisfied: six>=1.5 in
c:\users\mandy\appdata\roaming\python\python39\site-packages (from python-
dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

WARNING: Ignoring invalid distribution -ensorflow-intel
(c:\users\mandy\appdata\roaming\python\python39\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel
(c:\users\mandy\appdata\roaming\python\python39\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel
(c:\users\mandy\appdata\roaming\python\python39\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel
(c:\users\mandy\appdata\roaming\python\python39\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel
(c:\users\mandy\appdata\roaming\python\python39\site-packages)
WARNING: Ignoring invalid distribution -ensorflow-intel
(c:\users\mandy\appdata\roaming\python\python39\site-packages)

[notice] A new release of pip is available: 23.1.2 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[56]:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
```

[57]:

```
import statistics as STAT
```

[58]:

```
# library
import seaborn as sns
import matplotlib.pyplot as plt
```

[59]:

```
from scipy.stats import rankdata
```

[60]:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

[105]:

```
os.chdir("C:\\Users\\mandy\\OneDrive - ccac.edu\\CCAC\\CCAC\\Classes\\HIT 216")
#Set working directory so that Python knows where to find the file
```

[106]:

```
#calling the file in Python "OB" and using Pandas to read the cv
OB = pd.read_csv(r'\Users\mandy\OneDrive - ccac.edu\CCAC\CCAC\Classes\HIT␣
 ↪216\Nutrition__Physical_Activity__and_Obesity_-_Behavioral_Risk_Factor_Surveillance_System_
 ↪csv')
```

[107]:

```
#Assigned OB to a data frame so its easier to work with
df = pd.DataFrame(OB)
```

[108]:

```
#Drop the comlumns we don't need
OB1 = df.drop(columns=['Race/Ethnicity','Education','Gender',␣
 ↪'High_Confidence_Limit ', 'ClassID', 'TopicID', 'Low_Confidence_Limit',␣
 ↪'Sample_Size', 'Age(years)', 'YearStart', 'Datasource', 'Class', 'Topic',␣
 ↪'Data_Value_Unit', 'Data_Value_Type','Data_Value_Alt',␣
 ↪'Data_Value_Footnote_Symbol', 'Data_Value_Footnote', 'Total',␣
 ↪'GeoLocation','DataValueTypeID','LocationID'])
```

[109]:

```
#Filtering dataframe into National Obese Adults and Income levels
# Filtering the DataFrame where the location is National and the Percent of␣
 ↪Adults with Obesity, Q036
OBI = OB1[(OB1['StratificationCategory1'] == 'Income')]
```

[110]:

```
# Filtering the DataFrame where the location is National and the Percent of␣
 ↪Adults with Obesity, Q036
OBIN = OBI[(OBI['LocationAbbr'] == 'US') &
        (OBI['QuestionID'] == 'Q036') &
        (OBI['StratificationCategory1'] == 'Income')]
```

[111]:

```
OBIN
```

[111]:

```
       YearEnd LocationAbbr LocationDesc  \
2         2013           US     National
30        2014           US     National
68        2011           US     National
128       2014           US     National
167       2016           US     National
…          …            …          …
88755     2021           US     National
92561     2022           US     National
92569     2022           US     National
92577     2022           US     National
92578     2022           US     National

                                                Question  Data_Value  \
2      Percent of adults aged 18 years and older who …        28.8
30     Percent of adults aged 18 years and older who …        32.2
68     Percent of adults aged 18 years and older who …        32.3
128    Percent of adults aged 18 years and older who …        35.2
167    Percent of adults aged 18 years and older who …        32.0
…                                                    …         …
88755  Percent of adults aged 18 years and older who …        37.2
92561  Percent of adults aged 18 years and older who …        35.7
92569  Percent of adults aged 18 years and older who …        34.1
92577  Percent of adults aged 18 years and older who …        35.6
92578  Percent of adults aged 18 years and older who …        36.5
```

```
                   Income QuestionID StratificationCategory1  \
2        $50,000 - $74,999       Q036                  Income
30       $15,000 - $24,999       Q036                  Income
68       Less than $15,000       Q036                  Income
128      Less than $15,000       Q036                  Income
167      $35,000 - $49,999       Q036                  Income
…                   …        …                     …
88755    $25,000 - $34,999       Q036                  Income
92561    $50,000 - $74,999       Q036                  Income
92569   $75,000 or greater       Q036                  Income
92577    $35,000 - $49,999       Q036                  Income
92578    $25,000 - $34,999       Q036                  Income

          Stratification1 StratificationCategoryId1 StratificationID1
2        $50,000 - $74,999                       INC           INC5075
30       $15,000 - $24,999                       INC           INC1525
68       Less than $15,000                       INC         INCLESS15
128      Less than $15,000                       INC         INCLESS15
167      $35,000 - $49,999                       INC           INC3550
…                   …                       …               …
88755    $25,000 - $34,999                       INC           INC2535
92561    $50,000 - $74,999                       INC           INC5075
92569   $75,000 or greater                       INC         INC75PLUS
92577    $35,000 - $49,999                       INC           INC3550
92578    $25,000 - $34,999                       INC           INC2535

[84 rows x 11 columns]
```

[112]:
```
#Have data not reported--will need to drop NaN values
OBIN = OBIN[OBIN['Income'] != 'Data not reported']
```

[113]:
```
#Replacing range of income levels to the maxium income level
#Cleaning data to remove commas, dollar signs, and words
OBIN.loc[OBIN['Income'] == 'Less than $15,000', 'Income'] = '14999'
```

[114]:
```
OBIN.loc[OBIN['Income'] == '$15,000 - $24,999', 'Income'] = '24999'
```

[115]:
```
OBIN.loc[OBIN['Income'] == '$25,000 - $34,999', 'Income'] = '34999'
```

[116]:
```
OBIN.loc[OBIN['Income'] == '$35,000 - $49,999', 'Income'] = '49999'
```

[117]:
```
OBIN.loc[OBIN['Income'] == '$50,000 - $74,999', 'Income'] = '74999'
```

[118]:
```
OBIN.loc[OBIN['Income'] == '$75,000 or greater', 'Income'] = '80000'
```

[119]:
```
#Check OBIN
OBIN
```

```
[119]:        YearEnd LocationAbbr LocationDesc  \
       2         2013           US     National
       30        2014           US     National
       68        2011           US     National
       128       2014           US     National
       167       2016           US     National
       …        …          …          …
       88755     2021           US     National
       92561     2022           US     National
       92569     2022           US     National
       92577     2022           US     National
       92578     2022           US     National

                                                       Question  Data_Value Income  \
       2      Percent of adults aged 18 years and older who …         28.8  74999
       30     Percent of adults aged 18 years and older who …         32.2  24999
       68     Percent of adults aged 18 years and older who …         32.3  14999
       128    Percent of adults aged 18 years and older who …         35.2  14999
       167    Percent of adults aged 18 years and older who …         32.0  49999
       …                                                   …         …    …
       88755  Percent of adults aged 18 years and older who …         37.2  34999
       92561  Percent of adults aged 18 years and older who …         35.7  74999
       92569  Percent of adults aged 18 years and older who …         34.1  80000
       92577  Percent of adults aged 18 years and older who …         35.6  49999
       92578  Percent of adults aged 18 years and older who …         36.5  34999

             QuestionID StratificationCategory1     Stratification1  \
       2           Q036                  Income   $50,000 - $74,999
       30          Q036                  Income   $15,000 - $24,999
       68          Q036                  Income   Less than $15,000
       128         Q036                  Income   Less than $15,000
       167         Q036                  Income   $35,000 - $49,999
       …           …                     …                 …
       88755       Q036                  Income   $25,000 - $34,999
       92561       Q036                  Income   $50,000 - $74,999
       92569       Q036                  Income  $75,000 or greater
       92577       Q036                  Income   $35,000 - $49,999
       92578       Q036                  Income   $25,000 - $34,999

             StratificationCategoryId1 StratificationID1
       2                           INC           INC5075
       30                          INC           INC1525
       68                          INC         INCLESS15
       128                         INC         INCLESS15
       167                         INC           INC3550
       …                          …               …
       88755                       INC           INC2535
```

```
92561                          INC           INC5075
92569                          INC         INC75PLUS
92577                          INC           INC3550
92578                          INC           INC2535

[72 rows x 11 columns]
```

[120]:
```
#Hypothesis- Lower income levels will have a higher percentage of obese adults␣
 ↪than higher incomes
```

[121]:
```
#Perform Statistical testing to understand the data
#Mean percentage of obese adults across all incomes
Obese_Mean = STAT.mean(OBIN['Data_Value'])
```

[122]:
```
Obese_Mean
```

[122]:
```
32.2
```

[123]:
```
#Mode of obese adults across all incomes
Obese_Mode = STAT.mode(OBIN['Data_Value'])
```

[124]:
```
Obese_Mode
```

[124]:
```
32.3
```

[125]:
```
#Median of obese adults across all incomes
Obese_Median = STAT.median(OBIN['Data_Value'])
```

[126]:
```
Obese_Median
```

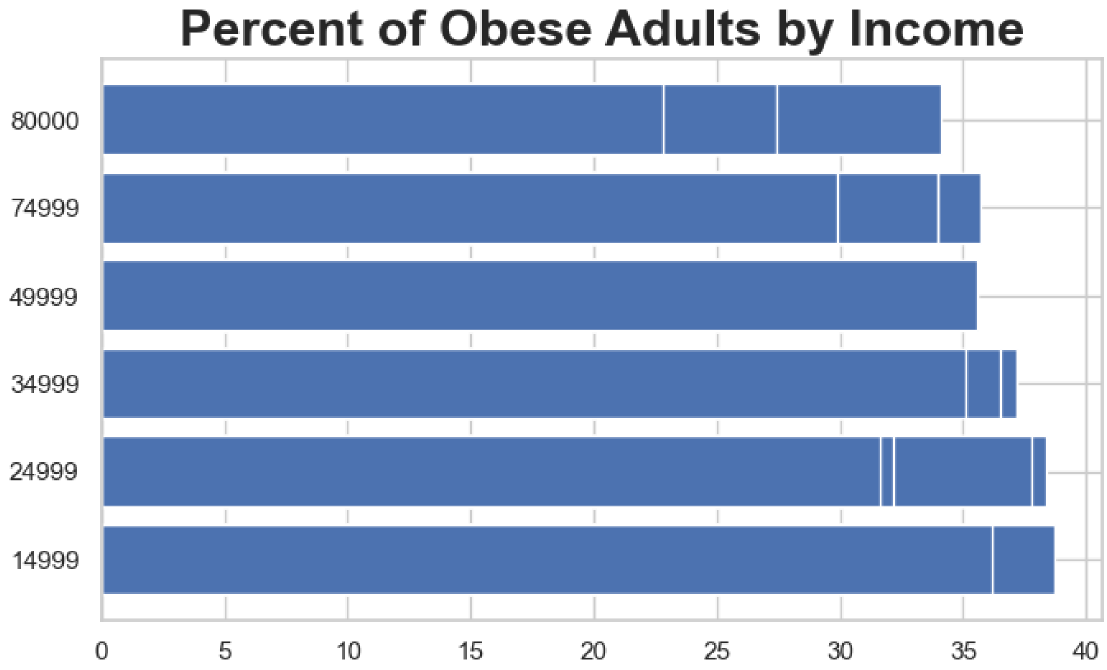
[126]:
```
32.5
```

[127]:
```
#nice distribution which is needed for normalacy.
#we know its a nice distribution because my mean and my median are close, if␣
 ↪they were far apart it would be skewed.
```

[128]:
```
#Create Graphs to Identify Relationships
```

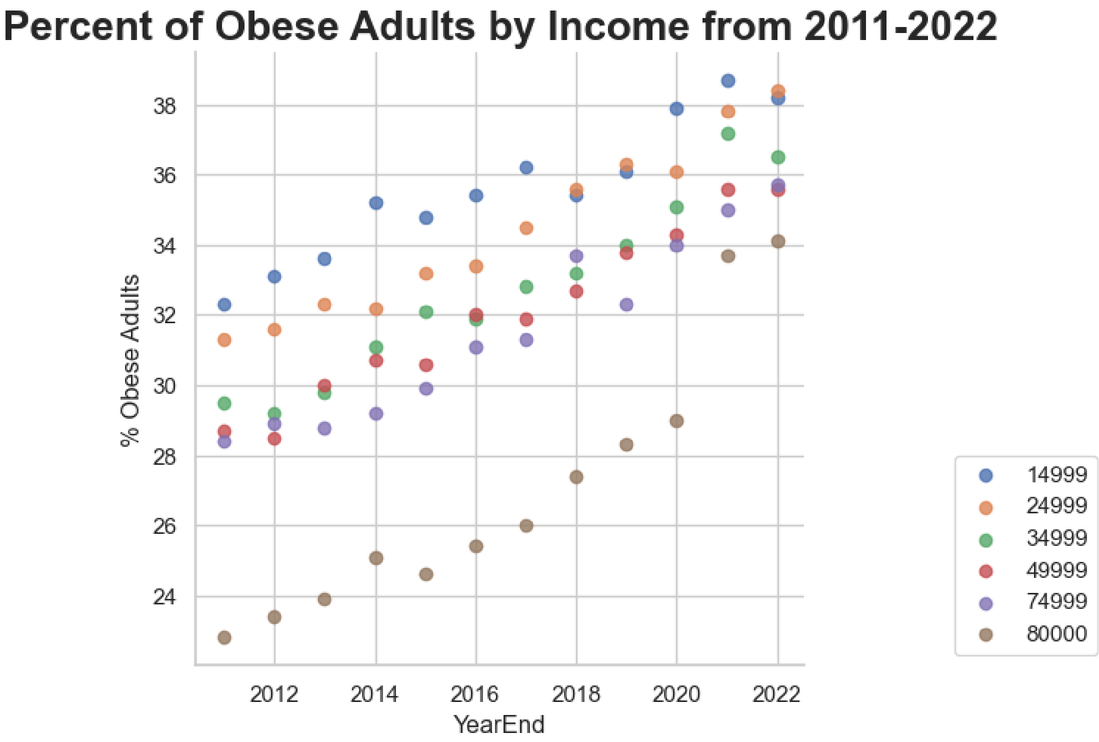
[129]:
```
# Data for bar chart
values = (OBIN['Income'])
categories = (OBIN['Data_Value'])
```

[130]:
```
# Sort the table
OBIN = OBIN.sort_values(by=['Income'])
# Create horizontal bars
plt.barh(y=OBIN['Income'], width=OBIN['Data_Value'])

# 1. Adjust horizontal padding
# 2. Decrease both left and right margins
# 3. Customize room in bottom and top.
plt.subplots_adjust(wspace=0.1, left=0.025, right=0.975, bottom=0.11, top=0.82)

# Add title
plt.title("Percent of Obese Adults by Income", fontsize=22, fontweight="bold",␣
 ↪fontname="Arial")


# Show graphic
plt.show()
```



[131]:
```
#Use a Scatter Plot to see how Obesity Percentage increases thru the years
#Scatter plot highlights income disparitys
```

[132]:
```
# Use the 'hue' argument to provide a factor variable
sns.lmplot( x="YearEnd", y="Data_Value", data=OBIN, fit_reg=False,␣
 ↪hue='Income', legend=False)

# Move the legend to an empty part of the plot
plt.legend(loc='lower right', bbox_to_anchor=(1.5,0))

# Add title and rename y
plt.ylabel('% Obese Adults')
plt.title("Percent of Obese Adults by Income from 2011-2022", fontsize=20,␣
 ↪fontweight="bold", fontname="Arial")

plt.show()
```



[133]:

```
#Better Scatter plot with regression line
```

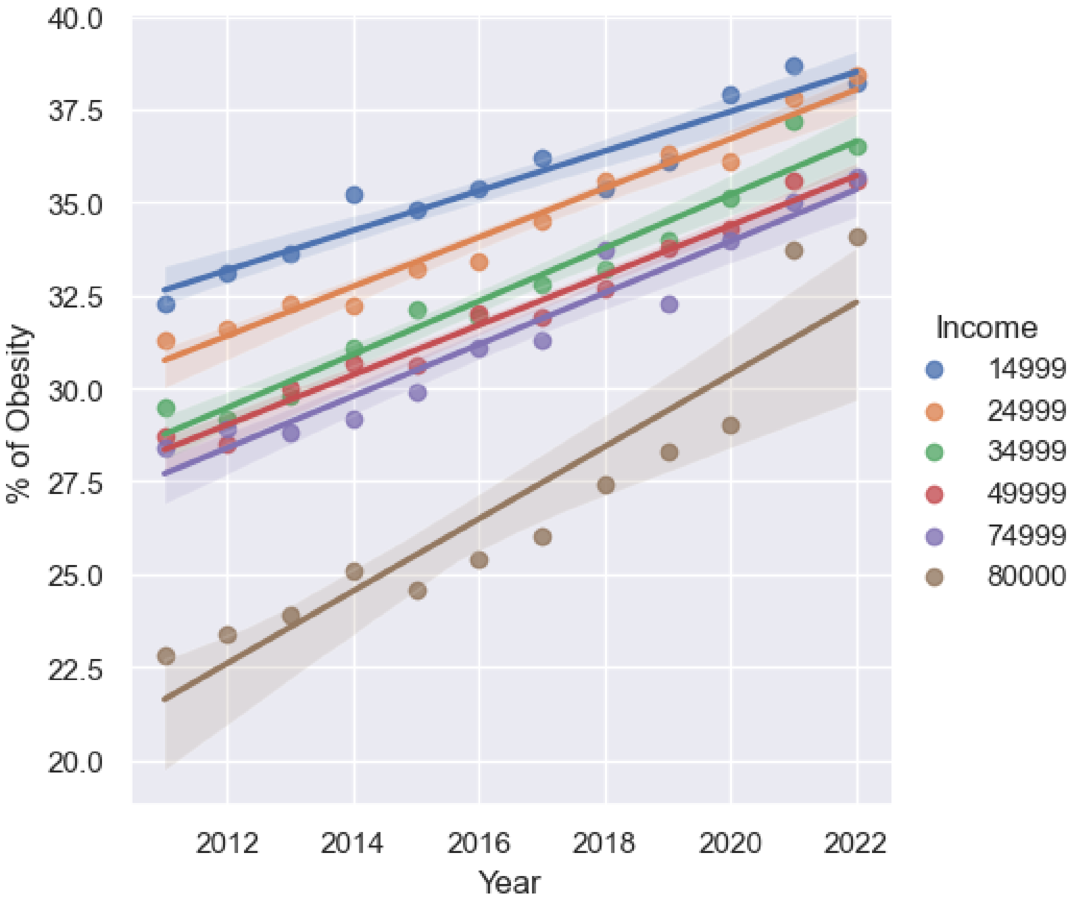
[134]:

```
import seaborn as sns
sns.set_theme()
# Define the order of income levels
income_order = sorted(OBIN["Income"].unique())

# Convert "Income" column to categorical with the defined order
OBIN["Income"] = pd.Categorical(OBIN["Income"], categories=income_order,␣
 ↪ordered=True)

# Plot Income across years
g = sns.lmplot(
    data=OBIN,
    x="YearEnd", y="Data_Value", hue="Income",
    height=5
)
```

```
# Use more informative axis labels than are provided by default
g.set_axis_labels("Year", "% of Obesity")
```

[134]:

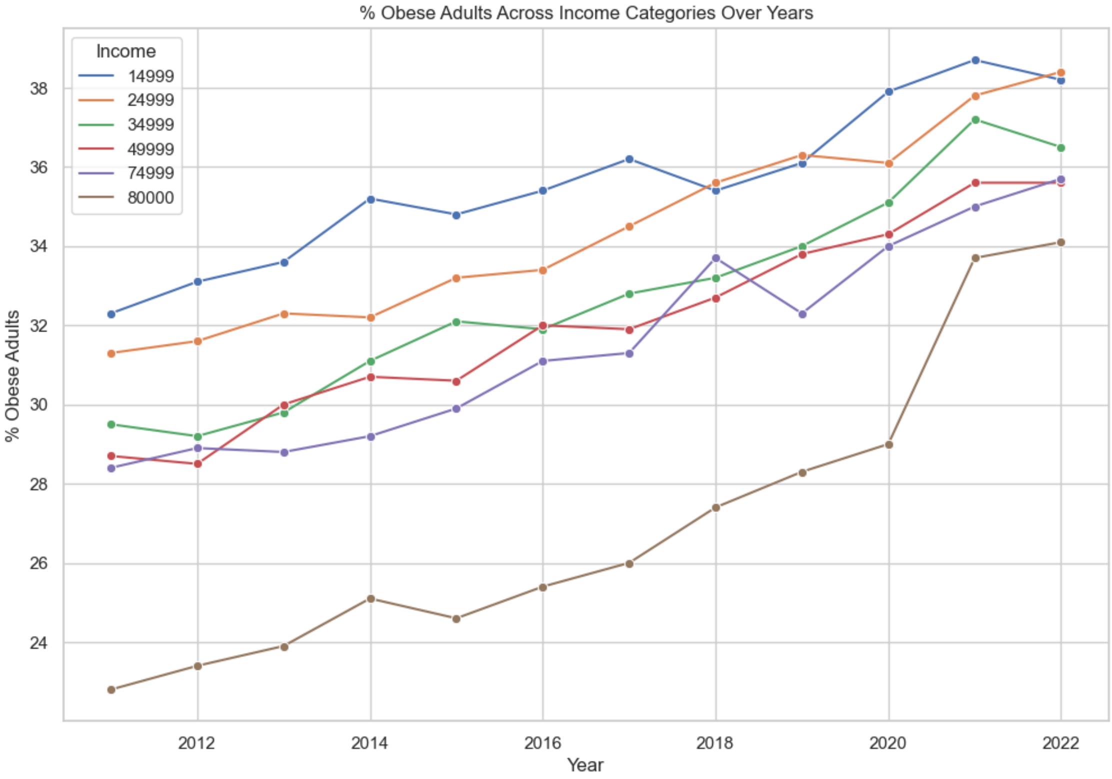
```
<seaborn.axisgrid.FacetGrid at 0x29b00817c10>
```



[135]:

```
#Bigger Line Graph to help show data
# Setting the plot style
sns.set(style="whitegrid")

# Creating the line plot
plt.figure(figsize=(12, 8))
sns.lineplot(data=OBIN, x='YearEnd', y='Data_Value', hue='Income', marker='o')

# Adding titles and labels
plt.title('% Obese Adults Across Income Categories Over Years')
plt.xlabel('Year')
plt.ylabel('% Obese Adults')
plt.legend(title='Income')
```

```
# Show the plot
plt.show()
```



[136]:
```
#Create a Model, Split data into training and test sets
```

[137]:
```
# Create the model here:
model = sm.OLS.from_formula('Data_Value ~ Income', data = OBIN)
# Fit the model here:
results = model.fit()
# Print the coefficients here:
print(results.params)
```

```
Intercept           35.575000
Income[T.24999]     -1.183333
Income[T.34999]     -2.875000
Income[T.49999]     -3.541667
Income[T.74999]     -4.050000
Income[T.80000]     -8.600000
dtype: float64
```

[138]:
```
OBINEx = OBIN[['Income', 'Data_Value', 'YearEnd']]
```

[139]:

```
x = OBIN['Income']
y = OBIN['Data_Value']
```

[140]:

```
print("Length of X:", len(x))
print("Length of y:", len(y))
```

```
Length of X: 72
Length of y: 72
```

[141]:

```
#drop na values
x = x.dropna()
y = y.dropna()
```

[142]:

```
# Reshape to make it a two-dimensional array
x = x.values.reshape(-1, 1)
```

[143]:

```
# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=0)
```

[144]:

```
model_obese= LinearRegression()
```

[145]:

```
model_obese.fit (X_train, y_train)
```

[145]:

```
LinearRegression()
```

[146]:

```
print("Data type of X_test:", type(X_test))
print("Shape of X_test:", X_test.shape)
print("Data type of model coefficients:", type(model_obese.coef_))
print("Shape of model coefficients:", model_obese.coef_.shape)
print("Data type of model intercept:", type(model_obese.intercept_))
```

```
Data type of X_test: <class 'pandas.core.arrays.categorical.Categorical'>
Shape of X_test: (15, 1)
Data type of model coefficients: <class 'numpy.ndarray'>
Shape of model coefficients: (1,)
Data type of model intercept: <class 'numpy.float64'>
```

[147]:

```
# Predictions on testing set
y_pred = model_obese.predict(X_test)

# Print out the shapes of y_test and y_pred to ensure they are compatible
print("Shape of y_test:", y_test.shape)
print("Shape of y_pred:", y_pred.shape)

# Evaluating the model
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y_test, y_pred)
```

```
r2 = r2_score(y_test, y_pred)

# Displaying the results
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
```

```
Shape of y_test: (15,)
Shape of y_pred: (15,)
Mean Squared Error: 9.68809675024846
R-squared Score: 0.2504309450136162
```

[148]:

```
# Predictions on testing set
y_pred = model_obese.predict(X_test)

# Evaluating the model
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Displaying the results
mse, r2
```

[148]:

```
(9.68809675024846, 0.2504309450136162)
```

[ ]:

```
```

[ ]:

```
```