```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn as sns

data=pd.read_csv('File name'). (*reads the file and stores it in the data frame data*)
data (*You should get the complete data*)
data.head() (*You should get the top 5 rows of the data*)
data.describe (*Will give the various description of the data such as number of
observations, mean, std, etc.*)

y=data['GPA']. (*the entries in the column of GPA is stored under the dependent variable
y*)
x1=data['SAT'] (*the entries in the column of GPA is stored under the dependent variable
x1*)

(*Plot a scatter graph*)
plt.scatter(x1,y)
plt.xlabel('SAT',fontsize=20)
plt.ylabel('GPA',fontsize=20)
plt.show()

(*Fitting a regression line*)
x=sm.add_constant(x1). (*this adds the y-intercept*)
results=sm.OLS(y,x).fit() (*Ordinary Least squares model for y on x*)
results.summary()

(*Plotting the linear line on the scatter graph*)
plt.scatter(x1,y)
yhat=0.0017*x1+0.275
fig=plt.plot(x1,yhat,lw=4,c='orange',label='regression line')
plt.xlabel('SAT',fontsize=20)
plt.ylabel('GPA',fontsize=20)
plt.show()

sns.set(). (*overrides the mathsplotlib*)

(*How to read the table*)
The table is divided into three parts: model summary, coefficients table, some other tests
The lower the standard error, better the results
Null Hypothesis is: Coefficient of x = 0 that is no dependance
P-value below 0.05 means the variable is significant

Sum of squares Total: $\sum_{i=1}^{n}(y_i - \bar{y})^2$ – Measure of the total variability of the dataset (SST or
TSS)
```

Sum of squares regression: $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$ – Measure of the explained variability in line (SSR or ESS)
Sum of square errors: $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ – Measures the variability by regression (SSE or RSS)
SST = SSR + SSE

$$0 < R^2 = \frac{SSR}{SST} < 1$$

For multiple regression, change

x1=data[['SAT','Rand 1,2,3]]

For multiple regression we prefer the adjusted R^2 which is given as $R^2_{adj} = 1 - (1 - R^2) * \frac{n-1}{n-p-1}$ where *n* is the number of observations and *p* is the number of predictors.

Converting Dummy Variables

raw_data=pd.read_csv('Filename')
data=raw.data.copy()
data['Attendance']=data['Attendance'].map({'Yes':1,'No':0}) (*maps the attendance entry of yes to 1 and no to 0*)
data.describe()