Multiple Regression Model

We can create a regression model using more than one explanatory variables. Let's load a dataset and do it.

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from scipy import stats
from scipy.stats import probplot
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

In [3]:

```
df = pd.read_csv("HeartDiseaseTrain.csv")
df.head(n=6)
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
0	63	1	1	145	233	1	2	150	0	2.3	3	2	3	(
1	67	1	4	160	286	0	2	108	1	1.5	2	5	2	'
2	67	1	4	120	229	0	2	129	1	2.6	2	4	4	'
3	37	1	3	130	250	0	0	187	0	3.5	3	2	2	(
4	41	0	2	130	204	0	2	172	0	1.4	1	2	2	(
5	56	1	2	120	236	0	0	178	0	0.8	1	2	2	(

Understanding the dataset using Data analysis

In [4]:

```
#checking the data types of the columns
df.dtypes
```

Out[4]:
```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [5]:
```
#checking if there is a missing row or observation in the dataset
df.isnull().values.any()
```

Out[5]:
```
False
```

In [6]:
```
#checking the number of missing values for each of the column
df.isnull().sum()
```

Out[6]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [7]:
```
print(df.describe(include='all'))
```

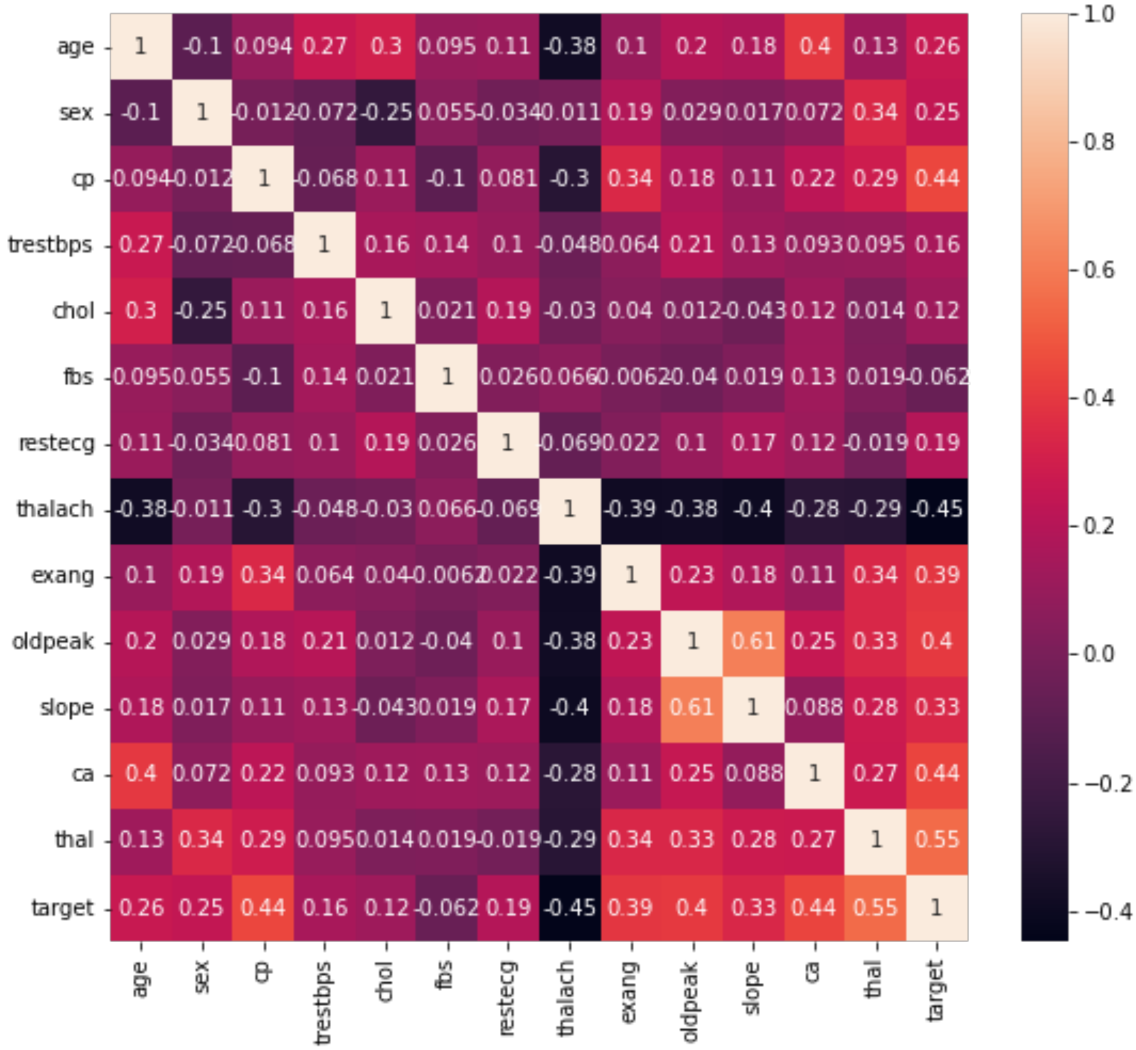
```
              age         sex          cp    trestbps        chol         fbs  \
count  200.000000  200.000000  200.000000  200.000000  200.000000  200.000000
mean    54.745000    0.710000    3.155000  132.565000  252.655000    0.170000
std      8.800981    0.454901    0.956845   18.025269   54.316086    0.376575
min     29.000000    0.000000    1.000000   94.000000  126.000000    0.000000
25%     48.750000    0.000000    3.000000  120.000000  218.500000    0.000000
50%     56.000000    1.000000    3.000000  130.000000  248.000000    0.000000
75%     61.000000    1.000000    4.000000  140.000000  282.250000    0.000000
max     77.000000    1.000000    4.000000  200.000000  564.000000    1.000000

          restecg     thalach       exang     oldpeak       slope          ca  \
count  200.000000  200.000000  200.000000  200.000000  200.000000  200.000000
mean     1.120000  151.105000    0.330000    1.116000    1.620000    2.695000
std      0.995265   22.244506    0.471393    1.171669    0.638465    0.972989
min      0.000000   88.000000    0.000000    0.000000    1.000000    2.000000
25%      0.000000  139.000000    0.000000    0.000000    1.000000    2.000000
50%      2.000000  154.500000    0.000000    0.800000    2.000000    2.000000
75%      2.000000  166.000000    1.000000    1.650000    2.000000    3.000000
max      2.000000  202.000000    1.000000    6.200000    3.000000    5.000000

             thal      target
count  200.000000  200.000000
mean     2.900000    0.450000
std      0.971969    0.498742
min      2.000000    0.000000
25%      2.000000    0.000000
50%      2.000000    0.000000
75%      4.000000    1.000000
max      4.000000    1.000000
```

We can also check the correlation between the variables using a correlation plot

In [8]:

```
# Draw a heatmap for correlation matrix
plt.figure(figsize=(9,8))
plt.subplot(1,1,1)
sns.heatmap(df.corr(), annot=True)
```

Out[8]: <AxesSubplot:>



Now let's create a model taking chol as our response variable and age and trestbps as our explanatory variable.

In [9]:
```
model = smf.ols('chol~trestbps+age', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.097
Model:                            OLS   Adj. R-squared:                  0.088
Method:                 Least Squares   F-statistic:                     10.64
Date:                Mon, 25 Apr 2022   Prob (F-statistic):           4.10e-05
Time:                        15:49:10   Log-Likelihood:                -1072.0
No. Observations:                 200   AIC:                             2150.
Df Residuals:                     197   BIC:                             2160.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    125.4077     31.767      3.948      0.000      62.760     188.055
trestbps       0.2446      0.212      1.156      0.249      -0.173       0.662
age            1.7322      0.433      3.998      0.000       0.878       2.587
==============================================================================
Omnibus:                       64.293   Durbin-Watson:                   2.125
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              256.300
Skew:                           1.212   Prob(JB):                     2.21e-56
Kurtosis:                       7.988   Cond. No.                     1.25e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 1.25e+03. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

Now let's ass the variable thalach in the model.

In [10]:

```
model = smf.ols('chol~trestbps+age+thal', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.098
Model:                            OLS   Adj. R-squared:                  0.085
Method:                 Least Squares   F-statistic:                     7.126
Date:                Mon, 25 Apr 2022   Prob (F-statistic):           0.000144
Time:                        15:49:10   Log-Likelihood:                -1071.9
No. Observations:                 200   AIC:                             2152.
Df Residuals:                     196   BIC:                             2165.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    128.3651     32.542      3.945      0.000      64.188     192.543
trestbps       0.2505      0.212      1.179      0.240      -0.168       0.669
age            1.7525      0.437      4.013      0.000       0.891       2.614
thal          -1.6756      3.829     -0.438      0.662      -9.227       5.876
==============================================================================
Omnibus:                       65.751   Durbin-Watson:                   2.125
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              268.767
Skew:                           1.234   Prob(JB):                     4.35e-59
Kurtosis:                       8.115   Cond. No.                     1.28e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 1.28e+03. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

Adding Interaction term

we can add interaction term in two ways one is interaction term and the original term using * and the other is using :: with just the interaction term.

In [11]:

```
model = smf.ols('chol~trestbps*age', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.109
Model:                            OLS   Adj. R-squared:                  0.096
Method:                 Least Squares   F-statistic:                     8.015
Date:                Mon, 25 Apr 2022   Prob (F-statistic):           4.58e-05
Time:                        15:49:10   Log-Likelihood:                -1070.7
No. Observations:                 200   AIC:                             2149.
Df Residuals:                     196   BIC:                             2163.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
==
                   coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
Intercept     -194.1734    200.754     -0.967      0.335    -590.089     201.7
42
trestbps         2.7245      1.553      1.755      0.081      -0.338       5.7
87
age              7.3871      3.534      2.090      0.038       0.417      14.3
57
trestbps:age    -0.0437      0.027     -1.612      0.109      -0.097       0.0
10
==============================================================================
Omnibus:                       61.076   Durbin-Watson:                   2.122
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              223.726
Skew:                           1.176   Prob(JB):                     2.62e-49
Kurtosis:                       7.616   Cond. No.                     4.12e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 4.12e+05. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

In [12]:

```
model = smf.ols('chol~trestbps:age', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.080
Model:                            OLS   Adj. R-squared:                  0.076
Method:                 Least Squares   F-statistic:                     17.26
Date:                Mon, 25 Apr 2022   Prob (F-statistic):           4.85e-05
Time:                        15:49:10   Log-Likelihood:                -1073.9
No. Observations:                 200   AIC:                             2152.
Df Residuals:                     198   BIC:                             2158.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==
                   coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
Intercept      187.6823     16.070     11.679      0.000     155.991     219.3
73
trestbps:age     0.0089      0.002      4.154      0.000       0.005       0.0
13
==============================================================================
Omnibus:                       71.861   Durbin-Watson:                   2.135
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              332.678
Skew:                           1.316   Prob(JB):                     5.75e-73
Kurtosis:                       8.744   Cond. No.                     3.26e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 3.26e+04. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

Categorical data

Categorical data is very useful in data science. We will see now how to manipulate categorical data and also how to use categorical data to build regression model.

We can make a column of a dataframe from numerical to categorical if feasible.

In [13]:

```
df['sex']=df['sex'].astype('category')
##we can rename the column values as male and female
df['sex'].replace({1:"M",0:"F"},inplace=True)
df['sex']=df['sex'].astype('category')
print(df.describe(include='category'))
```

```
        sex
count   200
unique    2
top       M
freq    142
```

In [14]:

```
df.head(n=5)
```

Out[14]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
0	63	M	1	145	233	1	2	150	0	2.3	3	2	3	(
1	67	M	4	160	286	0	2	108	1	1.5	2	5	2	'
2	67	M	4	120	229	0	2	129	1	2.6	2	4	4	'
3	37	M	3	130	250	0	0	187	0	3.5	3	2	2	(
4	41	F	2	130	204	0	2	172	0	1.4	1	2	2	(

Now we can see that the sex is now categorical as M and F

In [15]:

```
df.dtypes
```

Out[15]:

```
age            int64
sex         category
cp             int64
trestbps       int64
chol           int64
fbs            int64
restecg        int64
thalach        int64
exang          int64
oldpeak      float64
slope          int64
ca             int64
thal           int64
target         int64
dtype: object
```

Building model with categorical data

In [16]:

```
model = smf.ols('chol~sex', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.064
Model:                            OLS   Adj. R-squared:                  0.060
Method:                 Least Squares   F-statistic:                     13.63
Date:                Mon, 25 Apr 2022   Prob (F-statistic):           0.000287
Time:                        15:49:10   Log-Likelihood:                -1075.6
No. Observations:                 200   AIC:                             2155.
Df Residuals:                     198   BIC:                             2162.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    274.1724      6.916     39.644      0.000     260.534     287.811
sex[T.M]     -30.3062      8.208     -3.692      0.000     -46.492     -14.121
==============================================================================
Omnibus:                       49.018   Durbin-Watson:                   2.161
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              165.853
Skew:                           0.949   Prob(JB):                     9.67e-37
Kurtosis:                       7.038   Cond. No.                         3.48
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
```

In the summary table the sex[T.M] means that the type specified for sex is Male. T goes for type.

In [17]:

```
#Adding numerical variable with categorical
model = smf.ols('chol~trestbps+sex', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.083
Model:                            OLS   Adj. R-squared:                  0.074
Method:                 Least Squares   F-statistic:                     8.961
Date:                Mon, 25 Apr 2022   Prob (F-statistic):           0.000188
Time:                        15:49:10   Log-Likelihood:                -1073.5
No. Observations:                 200   AIC:                             2153.
Df Residuals:                     197   BIC:                             2163.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    218.1654     28.576      7.635      0.000     161.811     274.520
sex[T.M]     -29.1116      8.166     -3.565      0.000     -45.216     -13.007
trestbps       0.4161      0.206      2.019      0.045       0.010       0.823
==============================================================================
Omnibus:                       56.894   Durbin-Watson:                   2.148
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              228.726
Skew:                           1.051   Prob(JB):                     2.15e-50
Kurtosis:                       7.799   Cond. No.                     1.04e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 1.04e+03. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

In [18]:

```
#adding interaction term with categorical data
model = smf.ols('chol~trestbps*sex', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.083
Model:                            OLS   Adj. R-squared:                  0.069
Method:                 Least Squares   F-statistic:                     5.946
Date:                Mon, 25 Apr 2022   Prob (F-statistic):           0.000669
Time:                        15:49:10   Log-Likelihood:                -1073.5
No. Observations:                 200   AIC:                             2155.
Df Residuals:                     196   BIC:                             2168.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
=======
                        coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
Intercept           214.8519     48.513      4.429      0.000     119.178
310.526
sex[T.M]            -24.1618     59.054     -0.409      0.683    -140.624
92.301
trestbps              0.4407      0.357      1.235      0.218      -0.263
1.144
trestbps:sex[T.M]    -0.0370      0.438     -0.085      0.933      -0.900
0.826
==============================================================================
Omnibus:                       57.224   Durbin-Watson:                   2.145
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              231.410
Skew:                           1.056   Prob(JB):                     5.62e-51
Kurtosis:                       7.828   Cond. No.                     3.30e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 3.3e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

We should remember that if we use the categorical columns as numerical like 0 and 1 we will get the same results. We can also use C() in the model to factor them inside the model.

In [19]:

```
model = smf.ols('chol~trestbps+C(cp)', data = df)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   chol   R-squared:                       0.039
Model:                            OLS   Adj. R-squared:                  0.019
Method:                 Least Squares   F-statistic:                     1.978
Date:                Mon, 25 Apr 2022   Prob (F-statistic):             0.0994
Time:                        15:49:10   Log-Likelihood:                -1078.3
No. Observations:                 200   AIC:                             2167.
Df Residuals:                     195   BIC:                             2183.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    168.8236     33.850      4.987      0.000     102.065     235.582
C(cp)[T.2]     9.8141     16.945      0.579      0.563     -23.605      43.233
C(cp)[T.3]    17.1428     15.375      1.115      0.266     -13.180      47.466
C(cp)[T.4]    22.2369     14.750      1.508      0.133      -6.853      51.327
trestbps       0.5038      0.216      2.334      0.021       0.078       0.930
==============================================================================
Omnibus:                       75.125   Durbin-Watson:                   2.170
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              372.089
Skew:                           1.360   Prob(JB):                     1.59e-81
Kurtosis:                       9.104   Cond. No.                     1.32e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
[2] The condition number is large, 1.32e+03. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

In [20]:

```
##we can predict for a new value
preds = results.predict(pd.DataFrame({"trestbps":[100],"cp":[1]}))
print(preds)
```

```
0    219.206839
dtype: float64
```

In [21]:

```
#predicting for train
preds1 = results.predict(df)

df["predicted"] = preds1
```

In [22]:

```
df.head()
```

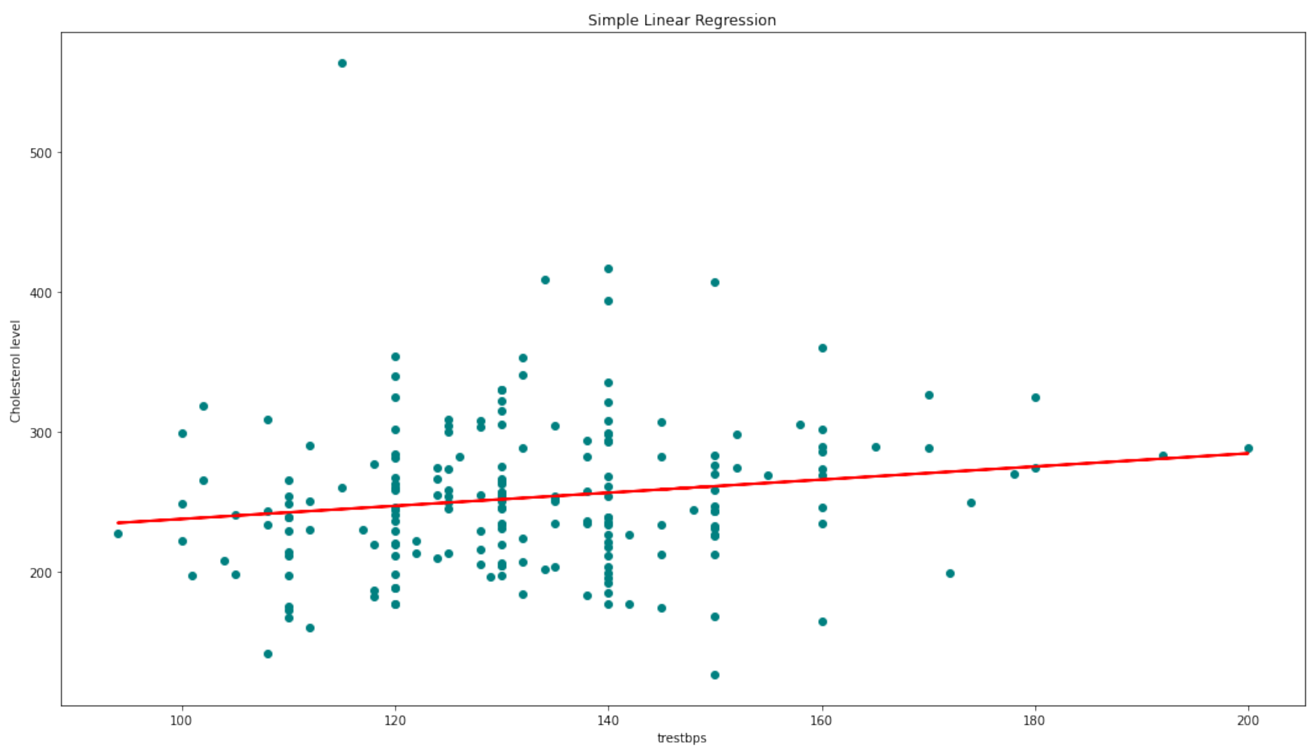
Out[22]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
0	63	M	1	145	233	1	2	150	0	2.3	3	2	3	(
1	67	M	4	160	286	0	2	108	1	1.5	2	5	2	
2	67	M	4	120	229	0	2	129	1	2.6	2	4	4	
3	37	M	3	130	250	0	0	187	0	3.5	3	2	2	(
4	41	F	2	130	204	0	2	172	0	1.4	1	2	2	(

In [23]:

```
model = smf.ols('chol~trestbps', data = df)
results = model.fit()
preds = results.predict()
# visualizing the results.
plt.figure(figsize=(18, 10))
# Scatter plot of input and output values
plt.scatter(df.trestbps, df.chol, color='teal')
# plot of the input and predicted output values
plt.plot(df.trestbps, results.predict(), color='Red', linewidth=2 )
plt.title('Simple Linear Regression')
plt.xlabel('trestbps')
plt.ylabel('Cholesterol level')
```

Out[23]: Text(0, 0.5, 'Cholesterol level')



In [ ]: