How to Create a Residual Plot in Python

A residual plot is a type of plot that displays the fitted values against the residual values for a regression model.

This type of plot is often used to assess whether or not a linear regression model is appropriate for a given dataset and to check for heteroscedasticity of residuals.

This tutorial explains how to create a residual plot for a linear regression model in Python.

Example: Residual Plot in Python

For this example we'll use a dataset that describes the attributes of 10 basketball players:

In [1]:

```
import numpy as np
import pandas as pd

#create dataset
df = pd.DataFrame({'rating': [90, 85, 82, 88, 94, 90, 76, 75, 87, 86],
                   'points': [25, 20, 14, 16, 27, 20, 12, 15, 14, 19],
                   'assists': [5, 7, 7, 8, 5, 7, 6, 9, 9, 5],
                   'rebounds': [11, 8, 10, 6, 6, 9, 6, 10, 10, 7]})

#view dataset
df
```

Out[1]:

	rating	points	assists	rebounds
0	90	25	5	11
1	85	20	7	8
2	82	14	7	10
3	88	16	8	6
4	94	27	5	6
5	90	20	7	9
6	76	12	6	6
7	75	15	9	10
8	87	14	9	10
9	86	19	5	7

Residual Plot for Simple Linear Regression

Suppose we fit a simple linear regression model using points as the predictor variable and rating as the response variable:

In [2]:
```
#import necessary libraries
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.formula.api import ols

#fit simple linear regression model
model = ols('rating ~ points', data=df).fit()

#view model summary
print(model.summary())
```

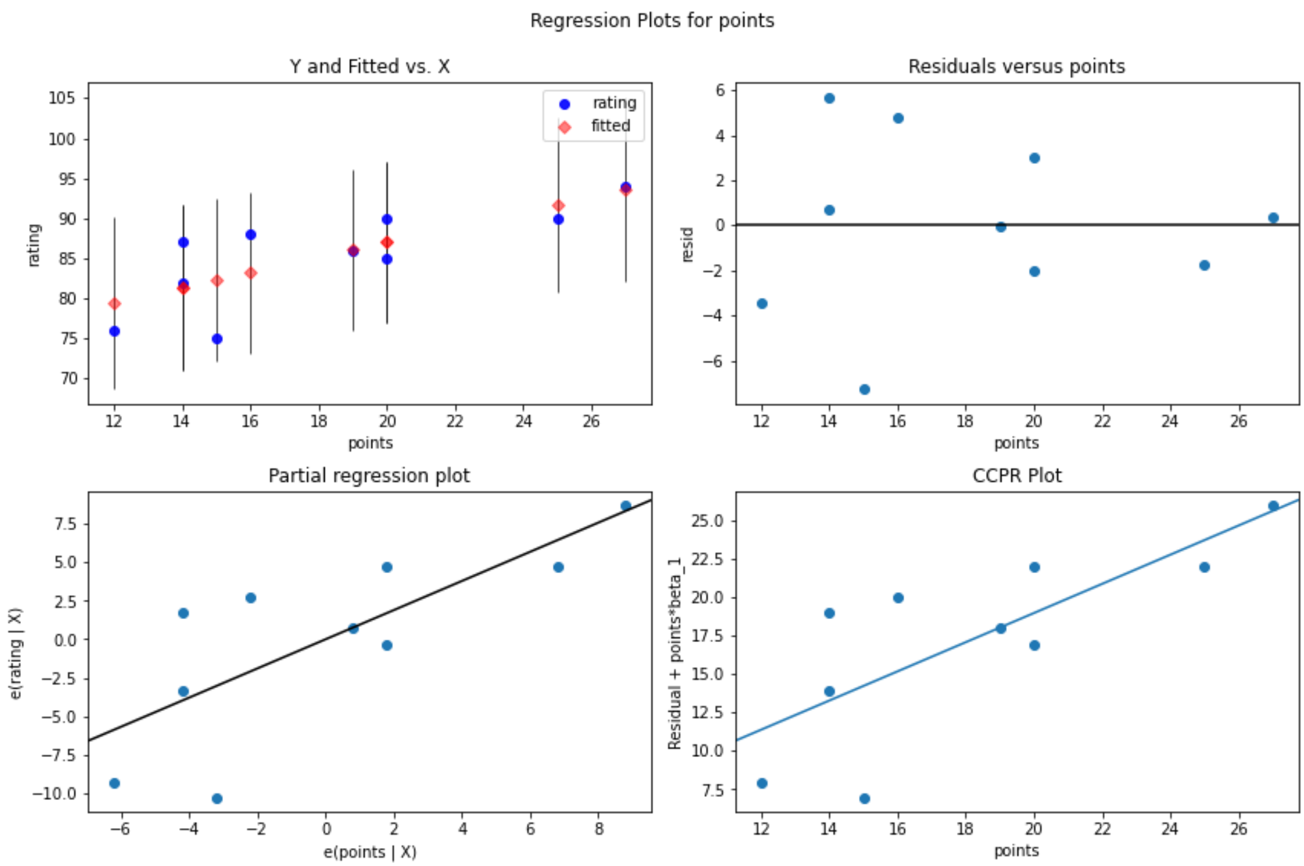
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 rating   R-squared:                       0.592
Model:                            OLS   Adj. R-squared:                  0.541
Method:                 Least Squares   F-statistic:                     11.61
Date:                Fri, 29 Apr 2022   Prob (F-statistic):            0.00927
Time:                        11:23:30   Log-Likelihood:                -27.252
No. Observations:                  10   AIC:                             58.50
Df Residuals:                       8   BIC:                             59.11
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     68.0282      5.235     12.994      0.000      55.956      80.101
points         0.9490      0.279      3.407      0.009       0.307       1.591
==============================================================================
Omnibus:                        0.220   Durbin-Watson:                   2.255
Prob(Omnibus):                  0.896   Jarque-Bera (JB):                0.204
Skew:                          -0.230   Prob(JB):                        0.903
Kurtosis:                       2.473   Cond. No.                         75.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
/Users/Code/opt/anaconda3/lib/python3.9/site-packages/scipy/stats/stats.py:154
1: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=10
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

We can create a residual vs. fitted plot by using the plot_regress_exog() function from the statsmodels library:

In [3]:
```
#define figure size
fig = plt.figure(figsize=(12,8))

#produce regression plots
fig = sm.graphics.plot_regress_exog(model, 'points', fig=fig)
```



Four plots are produced. The one in the top right corner is the residual vs. fitted plot. The x-axis on this plot shows the actual values for the predictor variable points and the y-axis shows the residual for that value.

Since the residuals appear to be randomly scattered around zero, this is an indication that heteroscedasticity is not a problem with the predictor variable.

Residual Plots for Multiple Linear Regression

Suppose we instead fit a multiple linear regression model using assists and rebounds as the predictor variable and rating as the response variable:

In [4]:
```
#fit multiple linear regression model
model = ols('rating ~ assists + rebounds', data=df).fit()

#view model summary
print(model.summary())
```

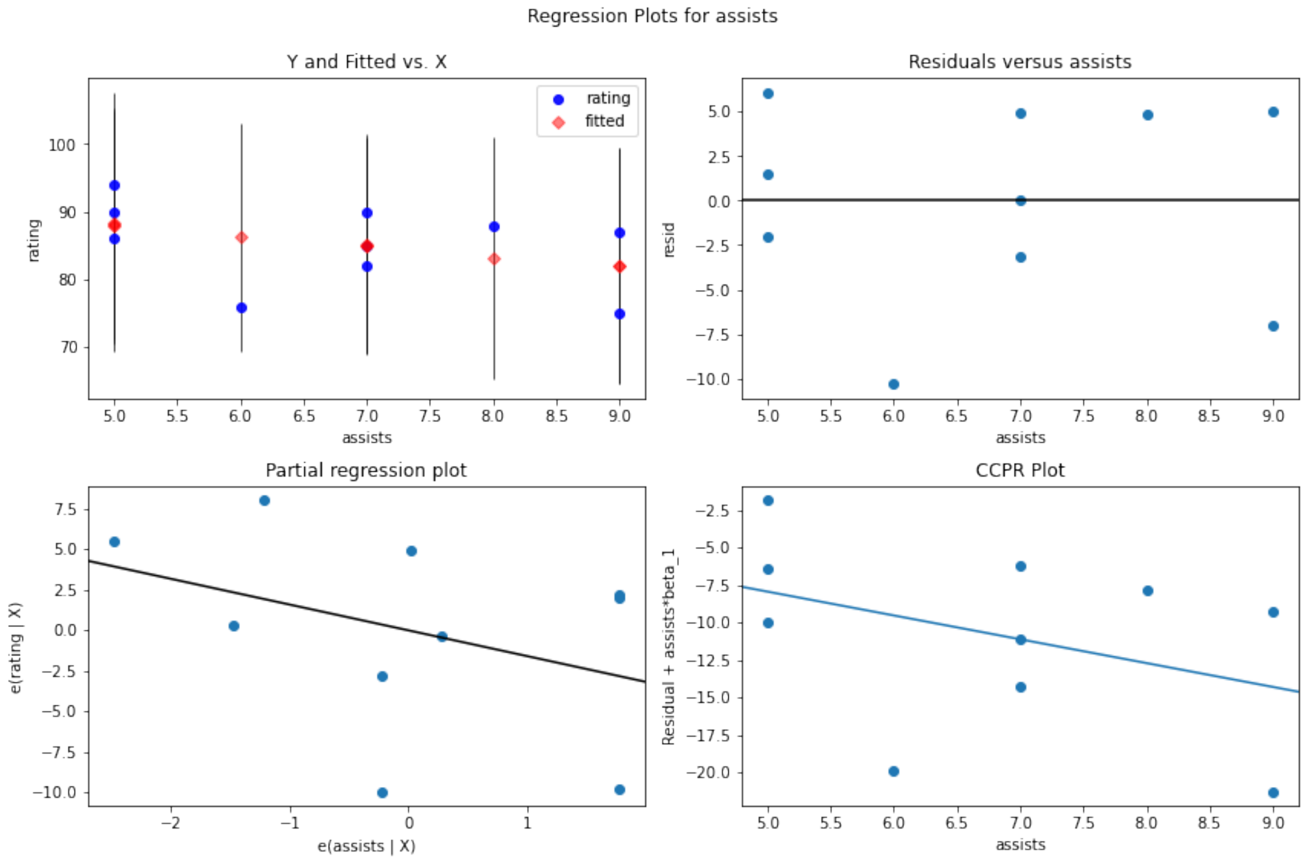
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 rating   R-squared:                       0.156
Model:                            OLS   Adj. R-squared:                 -0.086
Method:                 Least Squares   F-statistic:                    0.6455
Date:                Fri, 29 Apr 2022   Prob (F-statistic):              0.553
Time:                        11:35:26   Log-Likelihood:                -30.887
No. Observations:                  10   AIC:                             67.77
Df Residuals:                       7   BIC:                             68.68
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     95.1953     11.462      8.305      0.000      68.092     122.299
assists       -1.5904      1.440     -1.104      0.306      -4.996       1.815
rebounds       0.1108      1.146      0.097      0.926      -2.599       2.821
==============================================================================
Omnibus:                        1.183   Durbin-Watson:                   1.834
Prob(Omnibus):                  0.553   Jarque-Bera (JB):                0.897
Skew:                          -0.593   Prob(JB):                        0.639
Kurtosis:                       2.137   Cond. No.                         62.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
/Users/Code/opt/anaconda3/lib/python3.9/site-packages/scipy/stats/stats.py:154
1: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=10
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

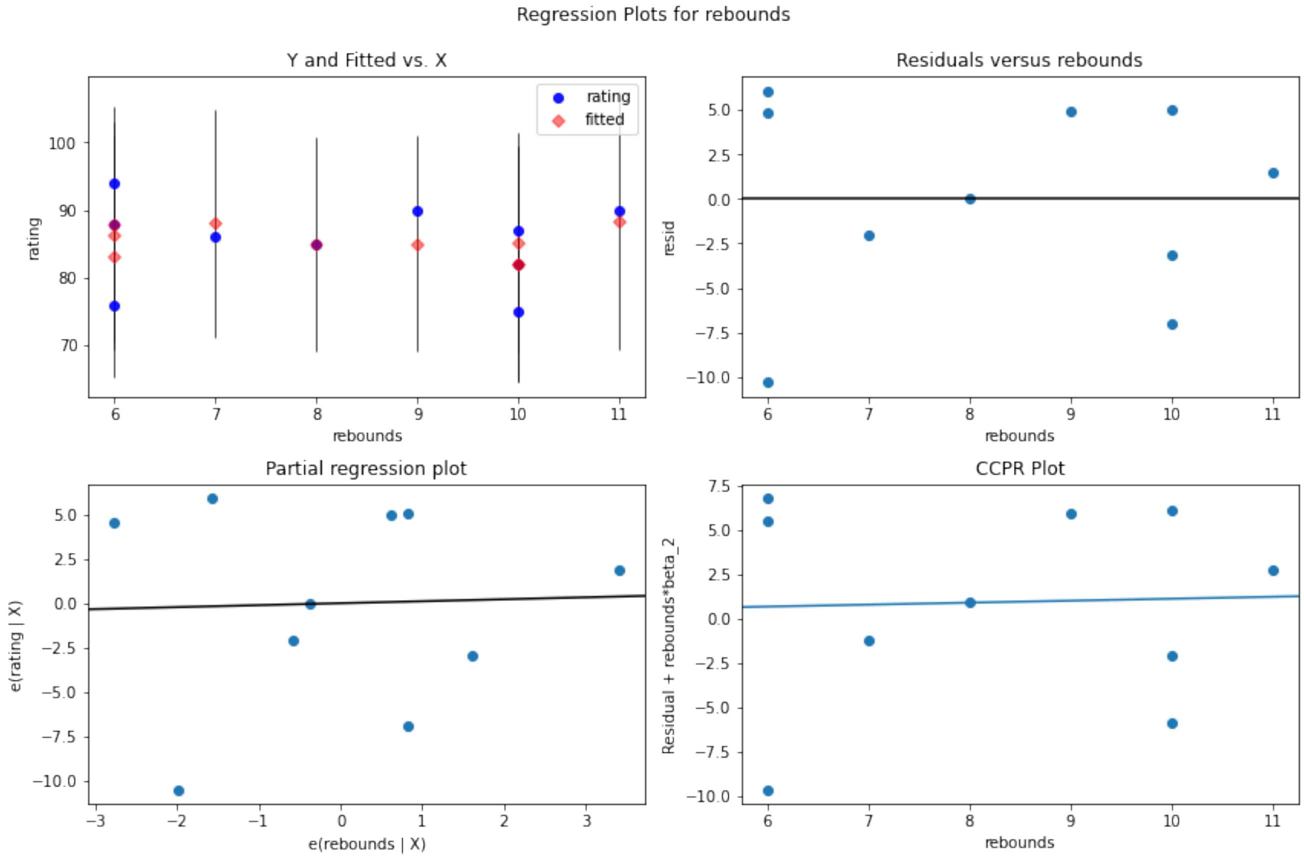
Once again we can create a residual vs. predictor plot for each of the individual predictors using the plot_regress_exog() function from the statsmodels library.

For example, here's what the residual vs. predictor plot looks like for the predictor variable assists:

In [5]:
```
#create residual vs. predictor plot for 'assists'
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'assists', fig=fig)
```



In [6]:
```
#create residual vs. predictor plot for 'assists'
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'rebounds', fig=fig)
```



In both plots the residuals appear to be randomly scattered around zero, which is an indication that heteroscedasticity is not a problem with either predictor variable in the model.

In [ ]: