Linear Regression model

```
from statsmodels.datasets import longley
from statsmodels.formula.api import ols
dta = longley.load_pandas().data
formula·=·'TOTEMP·~·GNPDEFL·+·GNP·+·UNEMP·+·ARMED·+·POP·+·YEAR'
results·=·ols(formula,·dta).fit()
hypotheses = '(GNPDEFL = GNP), (UNEMP = 2), (YEAR/1829 = 1)'
f_test = results.f_test(hypotheses)
print(f_test)
print(results.params)
print(results.summary())
```

```
<F test: F=array([[144.17976065]]), p=6.322026217368697e-08, df_denom=9, df_num=3>
Intercept   -3.482259e+06
GNPDEFL      1.506187e+01
GNP         -3.581918e-02
UNEMP       -2.020230e+00
ARMED       -1.033227e+00
POP         -5.110411e-02
YEAR         1.829151e+03
dtype: float64
                            OLS Regression Results
==============================================================================
Dep. Variable:                 TOTEMP   R-squared:                       0.995
Model:                            OLS   Adj. R-squared:                  0.992
Method:                 Least Squares   F-statistic:                     330.3
Date:                Sun, 19 Sep 2021   Prob (F-statistic):           4.98e-10
Time:                        01:02:55   Log-Likelihood:                -109.62
No. Observations:                  16   AIC:                             233.2
Df Residuals:                       9   BIC:                             238.6
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept  -3.482e+06    8.9e+05     -3.911      0.004    -5.5e+06   -1.47e+06
GNPDEFL       15.0619     84.915      0.177      0.863    -177.029     207.153
GNP           -0.0358      0.033     -1.070      0.313      -0.112       0.040
UNEMP         -2.0202      0.488     -4.136      0.003      -3.125      -0.915
ARMED         -1.0332      0.214     -4.822      0.001      -1.518      -0.549
POP           -0.0511      0.226     -0.226      0.826      -0.563       0.460
YEAR        1829.1515    455.478      4.016      0.003     798.788    2859.515
==============================================================================
Omnibus:                        0.749   Durbin-Watson:                   2.559
Prob(Omnibus):                  0.688   Jarque-Bera (JB):                0.684
Skew:                           0.420   Prob(JB):                        0.710
Kurtosis:                       2.434   Cond. No.                     4.86e+09
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.86e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
/usr/local/lib/python3.7/dist-packages/scipy/stats/stats.py:1535: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=16
  "anyway, n=%i" % int(n))
```

Log-log regression model

```
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf

# Fit regression model (using the natural log of one of the regressors)
formula = 'np.log(TOTEMP) ~ np.log(GNPDEFL) + np.log(GNP) + np.log(UNEMP) + np.log(ARMED) + np.log(POP) + YEAR'
results = ols(formula, dta).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         np.log(TOTEMP)   R-squared:                       0.993
Model:                            OLS   Adj. R-squared:                  0.989
Method:                 Least Squares   F-statistic:                     222.9
Date:                Sun, 19 Sep 2021   Prob (F-statistic):           2.89e-09
Time:                        01:11:55   Log-Likelihood:                 64.580
No. Observations:                  16   AIC:                            -115.2
Df Residuals:                       9   BIC:                            -109.8
Df Model:                           6
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept         -45.9649     19.651     -2.339      0.044     -90.418      -1.512
np.log(GNPDEFL)    -0.1773      0.139     -1.278      0.233      -0.491       0.137
np.log(GNP)        -0.0487      0.157     -0.310      0.764      -0.404       0.307
np.log(UNEMP)      -0.0794      0.019     -4.219      0.002      -0.122      -0.037
np.log(ARMED)      -0.0333      0.010     -3.174      0.011      -0.057      -0.010
np.log(POP)        -1.3184      0.510     -2.588      0.029      -2.471      -0.166
YEAR                0.0383      0.014      2.792      0.021       0.007       0.069
==============================================================================
Omnibus:                        0.086   Durbin-Watson:                   2.145
Prob(Omnibus):                  0.958   Jarque-Bera (JB):                0.301
Skew:                          -0.092   Prob(JB):                        0.860
Kurtosis:                       2.353   Cond. No.                     2.70e+07
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.7e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
/usr/local/lib/python3.7/dist-packages/scipy/stats/stats.py:1535: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=16
  "anyway, n=%i" % int(n))
```

Robust regression model (Huber) The stackloss dataset has four variables (airflow, water temperature, concentration of acid, and stack loss). Obtained from 21 days of operation of a plant for the oxidation of ammonia (NH(_3)) to nitric acid (HNO(_3)). The nitric oxides produced are absorbed in a countercurrent absorption tower". (Brownlee, cited by Dodge, slightly reformatted by MM.)

Air Flow represents the rate of operation of the plant. Water Temp is the temperature of cooling water circulated through coils in the absorption tower. Acid Conc. is the concentration of the acid circulating, minus 50, times 10: that is, 89 corresponds to 58.9 per cent acid. stack.loss (the dependent variable) is 10 times the percentage of the ingoing ammonia to the plant that escapes from the absorption column unabsorbed; that is, an (inverse) measure of the over-all efficiency of the plant.

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
data = sm.datasets.stackloss.load()
data.exog = sm.add_constant(data.exog)
huber_t = sm.RLM(data.endog, data.exog, M=sm.robust.norms.HuberT())
hub_results = huber_t.fit()
print(hub_results.params)
print(hub_results.bse)
print(
    hub_results.summary(
        yname="y", xname=["var_%d" % i for i in range(len(hub_results.params))]
    )
)
```

```
[-41.02649835   0.82938433   0.92606597  -0.12784672]
[9.79189854 0.11100521 0.30293016 0.12864961]
                    Robust linear Model Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                   21
Model:                            RLM   Df Residuals:                       17
Method:                          IRLS   Df Model:                            3
Norm:                          HuberT
Scale Est.:                       mad
Cov Type:                          H1
Date:                Sun, 19 Sep 2021
Time:                        01:28:08
No. Iterations:                    19
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
var_0        -41.0265      9.792     -4.190      0.000     -60.218     -21.835
var_1          0.8294      0.111      7.472      0.000       0.612       1.047
var_2          0.9261      0.303      3.057      0.002       0.332       1.520
var_3         -0.1278      0.129     -0.994      0.320      -0.380       0.124
==============================================================================

If the model instance has been used for another fit with different fit
parameters, then the fit options might not be the correct ones anymore .
/usr/local/lib/python3.7/dist-packages/statsmodels/datasets/utils.py:337: FutureWarning: load will return datasets containing pandas DataFrames and Serie
  FutureWarning)
```

❗ 0s completed at 9:42 PM