

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
import seaborn as sns
import scipy.stats as scp
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import warnings
warnings.filterwarnings("ignore")
```

ДЗ№6: тестируете модель из ДЗ№1 на гомоскедастичность и отсутствие автокорреляции (с помощью подходящих для этого тестов и графического анализа). При необходимости корректируете модель, если обнаружите гетероскедастичность и/или автокорреляцию.

```
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Эконометрика/Домашнее задание 5
x = data['X']
y = data['Y']
data
```

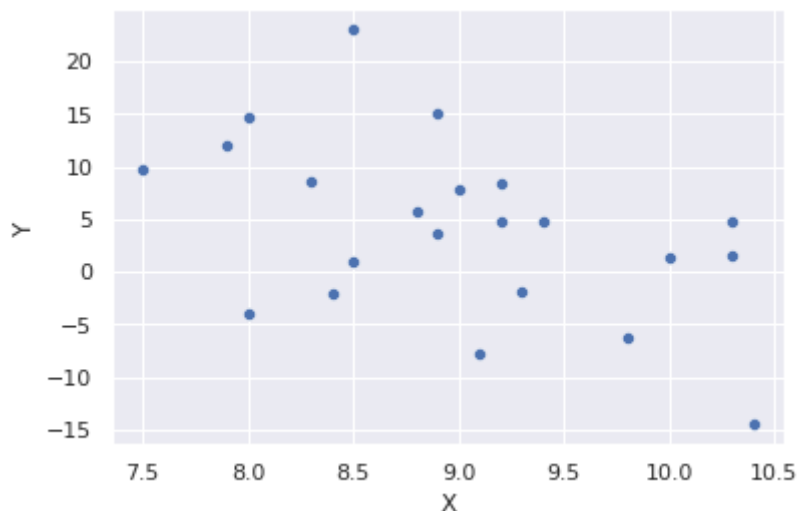
	year	X	Y
0	2000	9.2	4.762676
1	2001	8.5	1.042202
2	2002	8.3	8.643859
3	2003	8.5	23.079495
4	2004	8.9	14.970938
5	2005	8.9	3.694779

```
data = data.drop('year', axis = 1)
data.head()
```

	X	Y	
0	9.2	4.762676	
1	8.5	1.042202	
2	8.3	8.643859	
3	8.5	23.079495	
4	8.9	14.970938	
14	2014	10.5	1.405071

```
sns.set()
sns.scatterplot(x, y)
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pas
warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1c4151190>
```



```
import statsmodels.api as sm
results = sm.OLS.from_formula("Y ~ X", data = data).fit()
```

Основная модель регрессии

```
print(results.summary())
```

```

                                OLS Regression Results
=====
Dep. Variable:                  Y    R-squared:                0.229
Model:                          OLS    Adj. R-squared:           0.191
Method:                        Least Squares    F-statistic:              5.947
Date:                          Sat, 10 Dec 2022    Prob (F-statistic):       0.0242
Time:                          01:29:17    Log-Likelihood:          -74.744
No. Observations:                22    AIC:                     153.5
Df Residuals:                    20    BIC:                     155.7
Df Model:                        1
Covariance Type:                nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	48.2294	18.169	2.654	0.015	10.329	86.130
X	-4.9109	2.014	-2.439	0.024	-9.112	-0.710

```

=====
Omnibus:                        0.053    Durbin-Watson:           1.727
Prob(Omnibus):                  0.974    Jarque-Bera (JB):         0.065
Skew:                           0.016    Prob(JB):                 0.968
Kurtosis:                       2.735    Cond. No.                 103.
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly spe

▼ Считаем остатки

```
reg = LinearRegression().fit(x.values.reshape(-1,1), y.values.reshape(-1,1))
```

```
epsilons = y.values.reshape(-1,1) - reg.predict(x.values.reshape(-1, 1))
```

```
mean_e = epsilons.mean()
```

```
std_e = epsilons.std()
```

```
vec_w = (-1) * (epsilons - mean_e) / std_e
vec_w[:5]
```

```

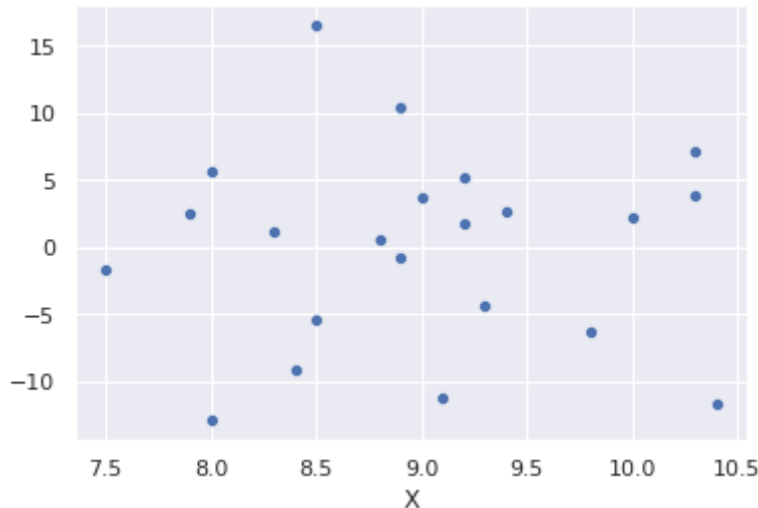
array([[ -0.23691399],
       [  0.75286089],
       [-0.16243865],
       [-2.29431523],
       [-1.44473385]])

```

```
sns.set()
```

```
sns.scatterplot(x, epsilons.reshape(1, -1)[0])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pas
warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1c36008e0>
```



Как следует из графика, нормального распределения остатков нет

```
# тест ранговой корреляции Спирмена
```

```
rho, p = scp.spearmanr(x, abs(epsilons))
rho, p
```

```
(0.11309030242269326, 0.6163076095411506)
```

```
# тест Глейзера
```

```
result = [0, 1, 2, 3, 4, 5]
```

```
data_1 = data
```

```
data_1.columns = ['y', 'x']
```

```
model = ['x', 'np.sqrt(x)', 'np.abs(1/x)', 'np.abs(1 / np.sqrt(x))', 'np.sqrt(x ** 3)']
```

```
result[1] = sm.OLS.from_formula("np.abs(epsilons) ~ x", data = data_1).fit().tvalues['x']
```

```
result[2] = sm.OLS.from_formula("np.abs(epsilons) ~ np.sqrt(x)", data = data_1).fit().tval
```

```
result[3] = sm.OLS.from_formula("np.abs(epsilons) ~ np.abs(1/x)", data = data_1).fit().tva
```

```
result[4] = sm.OLS.from_formula("np.abs(epsilons) ~ np.abs(1 / np.sqrt(x))", data = data_1
```

```
result[5] = sm.OLS.from_formula("np.abs(epsilons) ~ np.sqrt(x ** 3)", data = data_1).fit()
```

```
for i in range(5):
```

```
    print(f'Model {model[i]} t-stats : {np.abs(result[i + 1])}')
print(scp.t.isf(0.05 / 2, 20))
```

```
Model x t-stats : 3.534415069889281
```

```
Model np.sqrt(x) t-stats : 7.2080643450497695
```

```
Model np.abs(1/x) t-stats : 2.61360426639657
```

```
Model np.abs(1 / np.sqrt(x)) t-stats : 3.4610263116630597
```

```
Model np.sqrt(x ** 3) t-stats : 9.193121635057915
```

```
2.085963447265837
```

Из этого критерия следует, что гипотеза H_0 отвергается, и остатки гетероскедастичны

▼ Робустная ковариационная матрица

```
print(results.get_robustcov_results(cov_type = "HC0").summary2())
```

```

Results: Ordinary least squares
=====
Model:                OLS                Adj. R-squared:    0.191
Dependent Variable: Y                AIC:              153.4882
Date:                2022-12-10 01:34 BIC:              155.6703
No. Observations:    22                Log-Likelihood:   -74.744
Df Model:            1                F-statistic:      6.642
Df Residuals:        20                Prob (F-statistic): 0.0180
R-squared:           0.229              Scale:          57.533
-----
                Coef.   Std.Err.    t    P>|t|    [0.025   0.975]
-----+-----
Intercept      48.2294    17.3084    2.7865  0.0114   12.1247   84.3341
X              -4.9109     1.9056   -2.5771  0.0180   -8.8858   -0.9359
-----
Omnibus:            0.053              Durbin-Watson:      1.727
Prob(Omnibus):      0.974              Jarque-Bera (JB):    0.065
Skew:               0.016              Prob(JB):           0.968
Kurtosis:           2.735              Condition No.:      103
=====

```

```
white_test = het_white(results.get_robustcov_results(cov_type = "HC0").resid, results.get_
```

```
#define labels to use for output of White's test
```

```
labels = ['Test Statistic', 'Test Statistic p-value', 'F-Statistic', 'F-Test p-value']
```

```
#print results of White's test
```

```
print(dict(zip(labels, white_test))['Test Statistic p-value'])
```

```
0.9653326305899965
```

Из теста Уайта следует, что после робустной матрицы устранена гетероскедастичность, так как по результатам теста Уайта мы принимаем гипотезу H_0 .

▼ Метод Доступных Взвешенных Наименьших Квадратов

```
data_2 = data_1
```

```
data_2['x'] = 1 / data_2['x']
```

```
data_2['y'] = data_2['y'] * data_2['x']
```

```
fin_model = sm.OLS.from_formula("y ~ x", data = data_2)
result_dmkn = fin_model.fit()
```

```
sum_dm = result_dmkn.summary()
print(sum_dm)
```

```

                                OLS Regression Results
=====
Dep. Variable:                  y      R-squared:                0.992
Model:                        OLS      Adj. R-squared:          0.992
Method:                    Least Squares  F-statistic:            2537.
Date:                Sat, 10 Dec 2022  Prob (F-statistic):      1.51e-22
Time:                  01:34:58      Log-Likelihood:         -3.1805
No. Observations:                22      AIC:                   10.36
Df Residuals:                    20      BIC:                   12.54
Df Model:                        1
Covariance Type:                nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.0322	0.066	0.488	0.631	-0.105	0.170
x	9.1967	0.183	50.369	0.000	8.816	9.578

```

=====
Omnibus:                5.425      Durbin-Watson:          1.749
Prob(Omnibus):          0.066      Jarque-Bera (JB):        3.649
Skew:                   0.519      Prob(JB):                0.161
Kurtosis:               4.704      Cond. No.                 2.96
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly spe

```
y_pred = fin_model.predict(fin_model.fit().model.exog.reshape(2, 22))
y_pred[:5]
```

```

array([[ 1.20996602e+00,  2.35230778e-01,  1.20996602e+00,
         9.25766293e-01,  1.20996602e+00,  1.60026431e-01,
         1.20996602e+00,  1.86584444e-01,  1.20996602e+00,
         5.23307414e-02,  1.20996602e+00,  4.25316566e-01,
         1.20996602e+00,  2.21220514e-01,  1.20996602e+00,
         9.55816094e-02,  1.20996602e+00,  4.43081096e-03,
         1.20996602e+00, -1.82733993e-01,  1.20996602e+00,
        -5.36142910e-01],
       [ 1.95950704e+00,  3.25421453e-01,  1.95950704e+00,
         8.05317870e-01,  1.95950704e+00,  3.18302840e-01,
         1.95950704e+00,  6.97982460e-01,  1.95950704e+00,
         6.92071945e-04,  1.95950704e+00,  9.77440715e-01,
         1.95950704e+00,  3.78598656e-01,  1.95950704e+00,
         1.92271281e-01,  1.95950704e+00, -3.46407210e-01,
         1.95950704e+00, -3.72032971e-01,  1.95950704e+00,
        -4.73454024e-01],
       [ 1.11568908e+00,  2.23886635e-01,  1.11568908e+00,
         9.40916243e-01,  1.11568908e+00,  1.40118494e-01,
         1.11568908e+00,  1.22261027e-01,  1.11568908e+00,

```

```

5.88258305e-02, 1.11568908e+00, 3.55870634e-01,
1.11568908e+00, 2.01425560e-01, 1.11568908e+00,
8.34200251e-02, 1.11568908e+00, 4.85590624e-02,
1.11568908e+00, -1.58924051e-01, 1.11568908e+00,
-5.44027890e-01],
[ 1.04332851e+00, 2.15179642e-01, 1.04332851e+00,
9.52544314e-01, 1.04332851e+00, 1.24838515e-01,
1.04332851e+00, 7.28907451e-02, 1.04332851e+00,
6.38110194e-02, 1.04332851e+00, 3.02568658e-01,
1.04332851e+00, 1.86232300e-01, 1.04332851e+00,
7.40856196e-02, 1.04332851e+00, 8.24289086e-02,
1.04332851e+00, -1.40649158e-01, 1.04332851e+00,
-5.50079865e-01],
[ 1.06679608e+00, 2.18003445e-01, 1.06679608e+00,
9.48773163e-01, 1.06679608e+00, 1.29794033e-01,
1.06679608e+00, 8.89022393e-02, 1.06679608e+00,
6.21942508e-02, 1.06679608e+00, 3.19855257e-01,
1.06679608e+00, 1.91159693e-01, 1.06679608e+00,
7.71129019e-02, 1.06679608e+00, 7.14444292e-02,
1.06679608e+00, -1.46575969e-01, 1.06679608e+00,
-5.48117122e-01]])

```

▼ 95% доверительный интервал :

[0.025	0.975]
-0.105	0.170
8.816	9.578

```
scf.f.ppf(q = 1 - 0.05, dfn = 1, dfd = len(data_2) - 2)
```

```
4.351243503329288
```

▼ Значимость параметров регрессии:

$b_0 \Rightarrow p\text{-value} > 0.05$, следовательно гипотеза H_0 принимается и параметр не значим

$b_1 \Rightarrow p\text{-value} < 0.05$, следовательно гипотеза H_0 отвергается и параметр значим

Значимость модели регрессии:

$F\text{-stat} = 2537 > F\text{-table} = 4.351$, следовательно гипотеза H_0 отвергается и модель значима

▼ Качество модели

$R^2 = 0.992$

$R^2(\text{adj}) = 0.992$

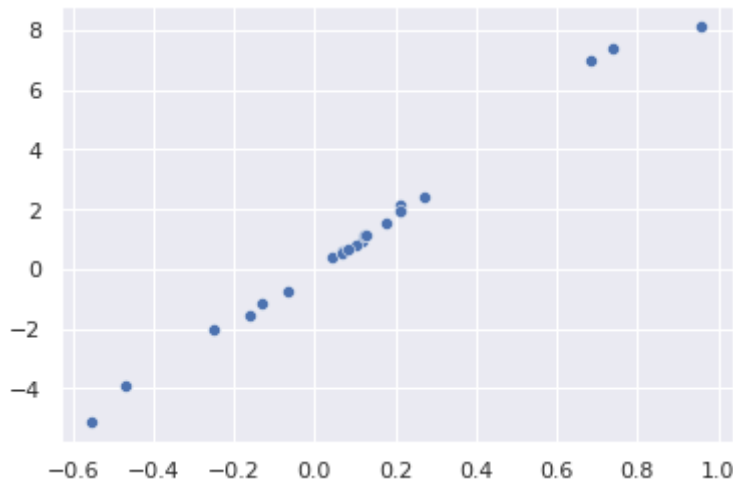
Качество модели отличное - значение R^2 больше 0.8, кроме того различие между скорректированным коэффициентом и обычным достаточно близки между собой

Считаем остатки скорректированной модели

```
reg = LinearRegression().fit(data_2['x'].values.reshape(-1,1), data_2['y'].values.reshape(-1,1))
epsilons = data_2['y'].values.reshape(-1,1) - reg.predict(data_2['x'].values.reshape(-1,1))
```

```
sns.set()
sns.scatterplot(data_2['x'].values, data_2['y'].values)
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pas
warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1c0a38b50>
```



```
sns.set()
sns.scatterplot(data_2['x'].values, epsilons.reshape(1, -1)[0])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1c0afd820>
```



После преобразования нет нормального распределения

▼ Тест Уайта

```
from statsmodels.stats.diagnostic import het_white
```



```
#perform White's test
white_test = het_white(fin_model.fit().resid, fin_model.fit().model.exog)

#define labels to use for output of White's test
labels = ['Test Statistic', 'Test Statistic p-value', 'F-Statistic', 'F-Test p-value']

#print results of White's test
print(dict(zip(labels, white_test))['Test Statistic p-value'])

9.111863761196062e-05
```

▼ Тест Голфреда-Кванта

```
from statsmodels.stats import api
new_x = np.concatenate((np.ones((len(data['y'])), 1)), data_2['x'].values.reshape(-1, 1)),

api.het_goldfeldquandt(epsilons, new_x)

(4.559595336066555, 0.016916729201749102, 'increasing')
```

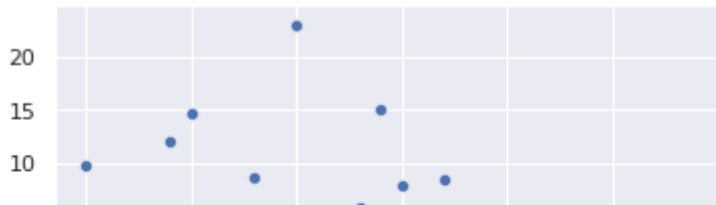
$p\text{-value} = 10^{-5} < 0.05$ следовательно гипотеза H_0 отвергается и модель показывает гетероскедастичность после тестов Уайта и Голфреда Кванта

После применения метода с робустной ковариационной матрицей качество и адекватность модели не меняется, при устранении гетероскедастичности. При этом метод ДМНК, сильно улучшил качество модели, но при этом не устранил гетероскедастичность.

▼ «Автокорреляция»

```
sns.set()
sns.scatterplot(x, y)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7ff1c081a1f0>



```
results = sm.OLS.from_formula("Y ~ X", data = data).fit()
print(results.summary())
```

```

                                OLS Regression Results
=====
Dep. Variable:                  Y      R-squared:                0.229
Model:                        OLS      Adj. R-squared:           0.191
Method:                    Least Squares  F-statistic:                5.947
Date:                Sat, 10 Dec 2022  Prob (F-statistic):        0.0242
Time:                  01:57:43      Log-Likelihood:           -74.744
No. Observations:                22      AIC:                     153.5
Df Residuals:                    20      BIC:                     155.7
Df Model:                        1
Covariance Type:                nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	48.2294	18.169	2.654	0.015	10.329	86.130
X	-4.9109	2.014	-2.439	0.024	-9.112	-0.710

```

=====
Omnibus:                0.053      Durbin-Watson:           1.727
Prob(Omnibus):          0.974      Jarque-Bera (JB):        0.065
Skew:                   0.016      Prob(JB):                0.968
Kurtosis:               2.735      Cond. No.                103.
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly spe



```
from statsmodels.stats.stattools import durbin_watson
```

```
#perform Durbin-Watson test
durbin_watson(results.resid)
```

```
1.7269779188844552
```

По результатам теста Дарбина-Уотсона гипотеза об отсутствии антикорреляции скорее всего принимается

[Платные продукты Colab](#) - [Отменить подписку](#)

✓ 0 сек. выполнено в 04:59

