In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st
from sympy import *
import time as dt
from sklearn.linear_model import LinearRegression
import warnings
from IPython.display import Math,Latex
from pandas.errors import SettingWithCopyWarning
from statsmodels.api import *
from sklearn.metrics import r2_score
from sklearn.preprocessing import PolynomialFeatures,StandardScaler

init_printing(use_unicode=True,use_latex=True)

def sm_model_outputs(model,untrust):
    r2=model.rsquared
    params=model.params
    nobs=model.nobs
    df=model.df_model
    tvalues=model.tvalues
    pvalues=model.pvalues
    f_pvalue=model.f_pvalue
    fvalue=model.fvalue
    untrust=0.05
    t_crit = st.t.ppf(1 - untrust/2, nobs - (df + 1))
    print(f'Коэффициент Детерминаци равен: {str(round(r2*100,3)).replace(".",","
    print(f'Коэфициенты b при каждом члене равны соответственно: {("; ".join(lis
    for i in range(len(params)):
        if pvalues.iloc[i]<untrust:
            print(f'Коэффициент {str(params.iloc[i]).replace(".",",")} стат. зна
        else:
            print(f'Коэффициент {str(params.iloc[i]).replace(".",",")} стат. нез
    print (f'Регрессия стат. значима, т.к. F-значение критерия фишера = {str(rou
```

Задание:

1. Оцените качество спецификации модели.
2. Проверьте адекватность модели.
3. Запишите оценённую модель в стандартной форме и дайте характеристику её качества.

$$\ln w_t = a + b \ln u_t + \varepsilon_t$$

Построим модель на изначальных данных

In [2]:
```
t1=pd.read_excel('Дз 3 нелинейная.xlsx','Лист2',usecols=[0,1,2],index_col=0,name

t1['1']=[1]*t1.shape[0]
t1['lnw']=np.log(t1['w'])
```

```
t1['lnu']=np.log(t1['u'])



md1 = LinearRegression()
md1.fit(t1[['lnu']].values,t1[['lnw']].values)
xx= np.linspace(t1[['u']].min(), t1[['u']].max(), 100)
yy= np.exp(md1.predict(np.log(xx).reshape(-1, 1)))
plt.scatter(t1[['u']],t1[['w']])
plt.plot(xx, yy , c='y')

m1 = OLS(t1[['lnw']],t1[['1','lnu']]).fit()
display(m1.summary())


sm_model_outputs(m1,0.05)
a,b,u, E = symbols('a b u \Epsilon')
w = exp(a) * u**b + E
print(f'Уравнение нелинейной регрессии:')
display(Math('w = '+latex(w)))
w = w.subs(a,round(m1.params.iloc[0],4))
w = w.subs(b,round(m1.params.iloc[1],4))
display(Math('w = '+latex(w)))

prediction2 = np.e** m1.predict(t1[['1','lnu']].values)
r2 = r2_score(t1[['w']].values,prediction2)
print(f'Коэффициент Детерминаци равен: {str(round(r2*100,3)).replace(".",",")}%

plt.show()
```

OLS Regression Results

Dep. Variable:	lnw	**R-squared:**	0.682
Model:	OLS	**Adj. R-squared:**	0.668
Method:	Least Squares	**F-statistic:**	47.26
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	6.64e-07
Time:	13:24:41	**Log-Likelihood:**	-0.056948
No. Observations:	24	**AIC:**	4.114
Df Residuals:	22	**BIC:**	6.470
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
1	2.2997	0.186	12.366	0.000	1.914	2.685
lnu	-0.7791	0.113	-6.875	0.000	-1.014	-0.544

Omnibus:	4.052	**Durbin-Watson:**	1.103
Prob(Omnibus):	0.132	**Jarque-Bera (JB):**	2.290
Skew:	-0.477	**Prob(JB):**	0.318
Kurtosis:	4.174	**Cond. No.**	7.97

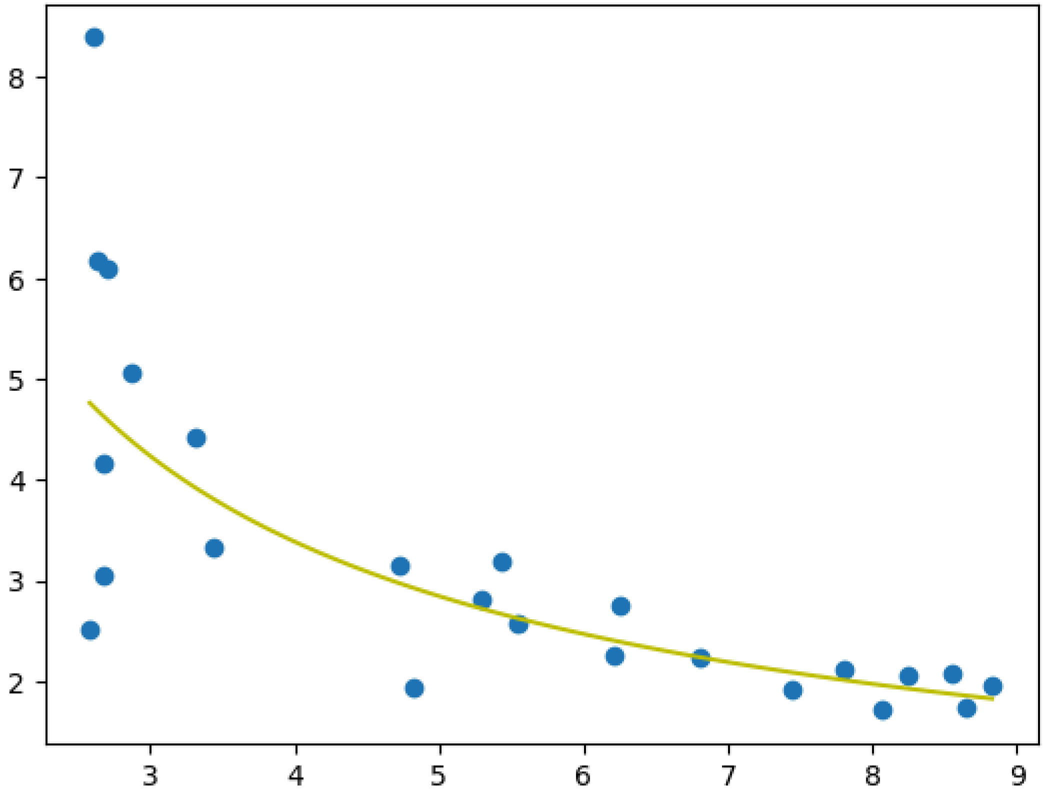
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
Коэффициент Детерминаци равен: 68,235% .
Коэфициенты b при каждом члене равны соответственно: 2,2997037728451235; -0,77905
63496663773 .
Коэффициент 2,2997037728451235 стат. значим, т.к. значение t = 12,3658 больше t_к
ритического = 2,0739 <=> pvalue=2,228541583865526e-11 < 0,05
Коэффициент -0,7790563496663773 стат. значим, т.к. значение t = 6,8746 больше t_к
ритического = 2,0739 <=> pvalue=6,638587637386962e-07 < 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 47,2597 больше F_крити
ческого <=> fvalue=6,638587637386955e-07 < 0,05
Уравнение нелинейной регрессии:
```

$$w = \Epsilon + u^b e^a$$

$$w = \Epsilon + \frac{9.97119064887161}{u^{0.7791}}$$

```
Коэффициент Детерминаци равен: 57,327% .
```



Стандартизируем модель

In [3]:
```
s1=StandardScaler()
t1['ln_std_u']=s1.fit_transform(t1['lnu'].values.reshape(-1,1))
t1['ln_std_w']=s1.fit_transform(t1['lnw'].values.reshape(-1,1))
display(t1)

m1 = OLS(t1[['ln_std_w']],t1[['ln_std_u']]).fit()
display(m1.summary())
sm_model_outputs(m1,0.05)


b,u, E = symbols('beta_1 u \Epsilon')
w =u**b + E
print(f'Уравнение нелинейной регрессии:')
display(Math('w = '+latex(w)))
w = w.subs(b,round(m1.params.iloc[0],4))
display(Math('w = '+latex(w)))


b,u, E = symbols('beta_1 ln(u) \Epsilon')
w =b*u + E
print(f'Уравнение линейной регрессии:')
display(Math('ln(w) = '+latex(w)))
w = w.subs(b,round(m1.params.iloc[0],4))
display(Math('ln(w) = '+latex(w)))


xx= np.linspace(t1[['ln_std_u']].min(), t1[['ln_std_u']].max(), 100)
yy= m1.predict(xx).reshape(-1, 1)

plt.scatter(t1[['ln_std_u']].values,t1[['ln_std_w']].values)
```

```
plt.plot(xx,yy,c='y')
plt.show()
```

t	w	u	1	lnw	lnu	ln_std_u	ln_std_w
1	1.73	8.65	1	0.548121	2.157559	1.273720	-1.216489
2	1.94	4.82	1	0.662688	1.572774	-0.007838	-0.950272
3	3.05	2.67	1	1.115142	0.982078	-1.302347	0.101089
4	4.17	2.67	1	1.427916	0.982078	-1.302347	0.827879
5	2.52	2.58	1	0.924259	0.947789	-1.377492	-0.342463
6	1.71	8.07	1	0.536493	2.088153	1.121617	-1.243509
7	1.95	8.83	1	0.667829	2.178155	1.318855	-0.938325
8	2.57	5.54	1	0.943906	1.711995	0.297264	-0.296810
9	5.06	2.87	1	1.621366	1.054312	-1.144047	1.277398
10	2.81	5.29	1	1.033184	1.665818	0.196069	-0.089354
11	4.43	3.31	1	1.488400	1.196948	-0.831460	0.968424
12	3.19	5.44	1	1.160021	1.693779	0.257345	0.205374
13	2.23	6.80	1	0.802002	1.916923	0.746364	-0.626551
14	2.06	8.25	1	0.722706	2.110213	1.169961	-0.810809
15	3.33	3.44	1	1.202972	1.235471	-0.747036	0.305180
16	2.12	7.80	1	0.751416	2.054124	1.047041	-0.744096
17	3.15	4.72	1	1.147402	1.551809	-0.053783	0.176053
18	1.92	7.45	1	0.652325	2.008214	0.946429	-0.974352
19	2.26	6.21	1	0.815365	1.826161	0.547460	-0.595499
20	6.18	2.64	1	1.821318	0.970779	-1.327110	1.742023
21	2.07	8.55	1	0.727549	2.145931	1.248237	-0.799556
22	8.39	2.60	1	2.127041	0.955511	-1.360569	2.452426
23	2.75	6.25	1	1.011601	1.832581	0.561531	-0.139507
24	6.10	2.70	1	1.808289	0.993252	-1.277861	1.711747

OLS Regression Results

Dep. Variable:	ln_std_w	**R-squared (uncentered):**	0.682
Model:	OLS	**Adj. R-squared (uncentered):**	0.669
Method:	Least Squares	**F-statistic:**	49.41
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	3.66e-07
Time:	13:24:41	**Log-Likelihood:**	-20.293
No. Observations:	24	**AIC:**	42.59
Df Residuals:	23	**BIC:**	43.76
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
ln_std_u	-0.8260	0.118	-7.029	0.000	-1.069	-0.583

Omnibus:	4.052	**Durbin-Watson:**	1.103
Prob(Omnibus):	0.132	**Jarque-Bera (JB):**	2.290
Skew:	-0.477	**Prob(JB):**	0.318
Kurtosis:	4.174	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
Коэффициент Детерминаци равен: 68,235% .
Коэффициенты b при каждом члене равны соответственно: -0,8260476877422089 .
Коэффициент -0,8260476877422089 стат. значим, т.к. значение t = 7,0291 больше t_к
ритического = 2,0739 <=> pvalue=3,6637058596786566e-07 < 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 49,4078 больше F_крити
ческого <=> fvalue=3,6637058596786704e-07 < 0,05
Уравнение нелинейной регрессии:
```

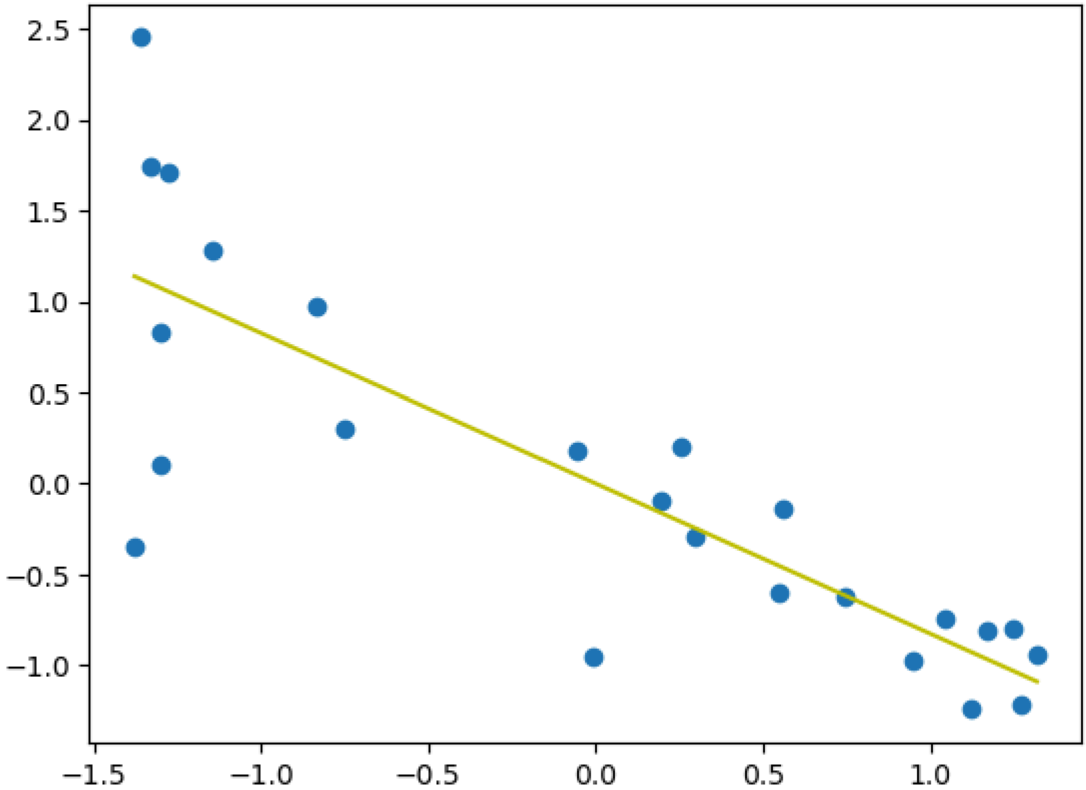
$$w = \Epsilon + u^{\beta_1}$$

$$w = \Epsilon + u^{-0.826}$$

```
Уравнение линейной регрессии:
```

$$ln(w) = \Epsilon + \beta_1 ln(u)$$

$$ln(w) = \Epsilon - 0.826 ln(u)$$



Запишите оценённую модель в стандартной форме и дайте характеристику её качества.

Оцените модель $\Delta K_t = \alpha I_t^{\beta} \cdot \varepsilon_t$.

$$\Delta K_t = \alpha I_t^{\beta} \times \epsilon_t$$
$$\ln(\Delta K_t) = \ln(\alpha) + \beta \times \ln(I_t) + \epsilon_t$$

Построим модель на изначальных данных

In [4]:
```
t2=pd.read_excel('Дз 3 нелинейная.xlsx','Лист3',usecols=[0,1,2],index_col=0,name

t2['1']=[1]*t2.shape[0]
t2['lnK']=np.log(t2['K'])
t2['lnI']=np.log(t2['I'])

m2 = OLS(t2[['lnK']].iloc[:,0].values,t2[['lnI','1']]).fit()
display(m2.summary())


xx= np.linspace(t2[['I']].min(), t2[['I']].max(), 100).reshape((-1,1))
xx = np.concatenate([xx,np.ones(xx.shape[0]).reshape((-1,1))],axis=1)
prediction = np.e** m2.predict(xx)
prediction2 = np.e** m2.predict(t2[['lnI','1']].values)

sm_model_outputs(m2,0.05)
```

```
r2 = (r2_score(t2[['K']].iloc[:,0].values,prediction2))
print(f'Коэффициент Детерминаци равен: {str(round(r2*100,3)).replace(".",",")}%
```

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.982
Model:	OLS	**Adj. R-squared:**	0.981
Method:	Least Squares	**F-statistic:**	1359.
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	2.58e-23
Time:	13:24:41	**Log-Likelihood:**	41.209
No. Observations:	27	**AIC:**	-78.42
Df Residuals:	25	**BIC:**	-75.83
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
lnI	0.9726	0.026	36.869	0.000	0.918	1.027
1	0.0363	0.117	0.311	0.759	-0.204	0.277

Omnibus:	18.786	**Durbin-Watson:**	0.522
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	27.506
Skew:	-1.489	**Prob(JB):**	1.06e-06
Kurtosis:	6.947	**Cond. No.**	51.6

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

```
Коэффициент Детерминаци равен: 98,194% .
Коэфициенты b при каждом члене равны соответственно: 0,9726013794191091; 0,036260
64833517084 .
Коэффициент 0,9726013794191091 стат. значим, т.к. значение t = 36,8692 больше t_к
ритического = 2,0595 <=> pvalue=2,5764166737745987e-23 < 0,05
Коэффициент 0,03626064833517084 стат. незначим, т.к. значение t = 0,3107 меньше t
_критического = 2,0595 <=> pvalue=0,7585750524336108 > 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 1359,3374 больше F_кри
тического <=> fvalue=2,5764166737745728e-23 < 0,05
Коэффициент Детерминаци равен: 96,498% .
```

In [5]:
```
m2 = OLS(t2[['lnK']].iloc[:,0].values,t2[['lnI']]).fit()
display(m2.summary())


xx= np.linspace(t2[['I']].min(), t2[['I']].max(), 100).reshape((-1,1))
prediction = np.e** m2.predict(xx)
prediction2 = np.e** m2.predict(t2[['lnI']].values)
sm_model_outputs(m2,0.05)
```

```
r2 = (r2_score(t2[['K']].iloc[:,0].values,prediction2))
print(f'Коэффициент Детерминаци равен: {str(round(r2*100,3)).replace(".",",")}%

b,I, E= symbols('beta, I_t, epsilon')
K = I**b + E
print(f'Уравнение нелинейной регрессии:')
display(Math('K = '+latex(K)))
K = K.subs(b,round(m2.params.iloc[0],4))
display(Math('K = '+latex(K)))

xx= np.linspace(t2[['I']].min(), t2[['I']].max(), 100)
yy= np.exp(m2.predict(np.log(xx).reshape(-1, 1)))

plt.scatter(t2[['I']].values,t2[['K']].values)
plt.plot(xx,yy,c='y')
plt.show()
```

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	1.000
Model:	OLS	**Adj. R-squared (uncentered):**	1.000
Method:	Least Squares	**F-statistic:**	1.763e+05
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	2.42e-51
Time:	13:24:41	**Log-Likelihood:**	41.157
No. Observations:	27	**AIC:**	-80.31
Df Residuals:	26	**BIC:**	-79.02
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
lnI	0.9808	0.002	419.829	0.000	0.976	0.986

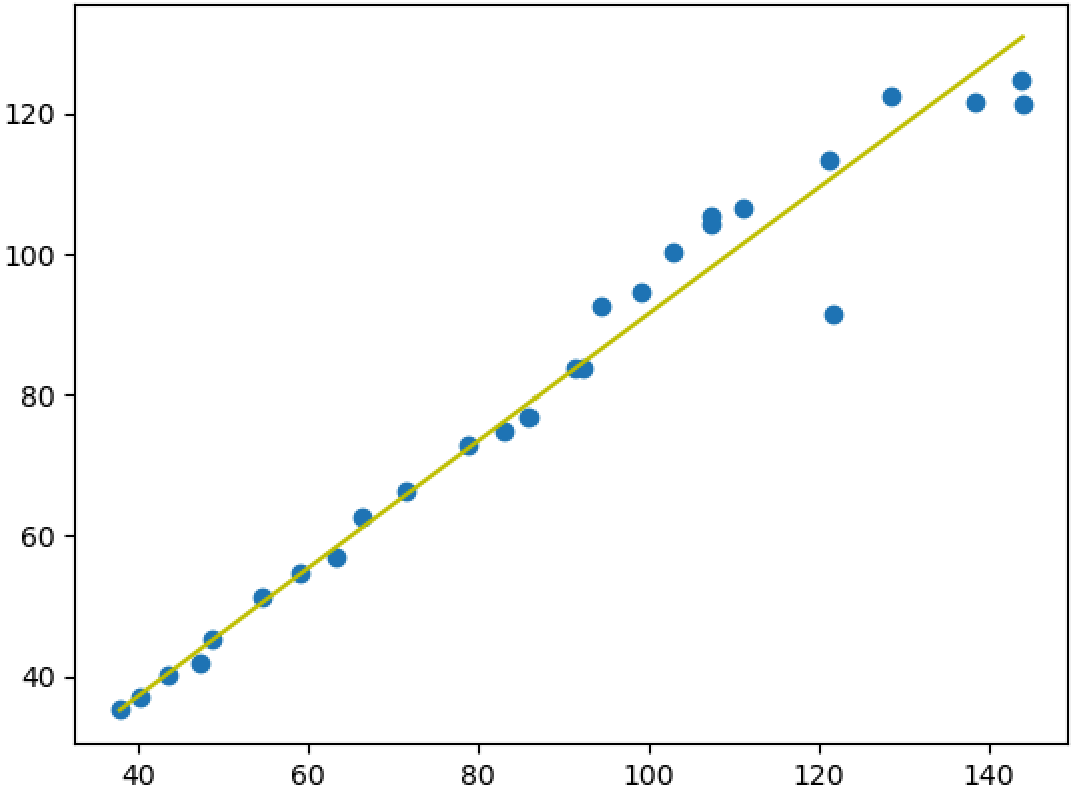
Omnibus:	20.825	**Durbin-Watson:**	0.518
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	32.803
Skew:	-1.629	**Prob(JB):**	7.53e-08
Kurtosis:	7.306	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
Коэффициент Детерминаци равен: 99,985% .
Коэфициенты b при каждом члене равны соответственно: 0,9807652414335774 .
Коэффициент 0,9807652414335774 стат. значим, т.к. значение t = 419,829 больше t_к
ритического = 2,0595 <=> pvalue=2,4224054494664673e-51 < 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 176256,354 больше F_кр
итического <=> fvalue=2,4224054494664916e-51 < 0,05
Коэффициент Детерминаци равен: 96,428% .
Уравнение нелинейной регрессии:
```

$$K = I_t^{\beta} + \epsilon$$

$$K = I_t^{0.9808} + \epsilon$$



Стандартизируем модель

In [6]:
```
s1=StandardScaler()
t2['ln_std_K']=s1.fit_transform(t2['lnK'].values.reshape(-1,1))
t2['ln_std_I']=s1.fit_transform(t2['lnI'].values.reshape(-1,1))

display(t2)

m2 = OLS(t2[['ln_std_K']],t2[['ln_std_I']]).fit()
display(m2.summary())
sm_model_outputs(m2,0.05)

b,I, E= symbols('beta, I_t, epsilon')
K = I**b + E
print(f'Уравнение нелинейной регрессии:')
display(Math('K = '+latex(K)))
K = K.subs(b,round(m2.params.iloc[0],4))
display(Latex('K = '+latex(K)))


b,I, E= symbols('beta, ln(I_{t}), epsilon')
```

```
K =b*I + E
print(f'Уравнение линейной регрессии:')
display(Math('ln(K) = '+latex(K)))
K = K.subs(b,round(m2.params.iloc[0],4))
display(Math('ln(K) = '+latex(K)))


xx= np.linspace(t2[['ln_std_I']].min(), t2[['ln_std_I']].max(), 100)
yy= m2.predict(xx).reshape(-1, 1)

plt.scatter(t2[['ln_std_I']].values,t2[['ln_std_K']].values)
plt.plot(xx,yy,c='y')
plt.show()
```

Год	K	I	1	lnK	lnI	ln_std_K	ln_std_I
1994	35.474	37.759	1	3.568800	3.631224	-1.922270	-1.941944
1995	37.096	40.131	1	3.613509	3.692149	-1.808026	-1.789143
1996	40.319	43.412	1	3.696823	3.770736	-1.595136	-1.592046
1997	41.816	47.094	1	3.733279	3.852146	-1.501980	-1.387869
1998	45.239	48.579	1	3.811960	3.883191	-1.300929	-1.310006
1999	51.176	54.564	1	3.935271	3.999374	-0.985835	-1.018617
2000	54.609	58.884	1	4.000199	4.075569	-0.819926	-0.827518
2001	56.980	63.251	1	4.042700	4.147111	-0.711322	-0.648091
2002	62.604	66.325	1	4.136829	4.194567	-0.470797	-0.529070
2003	66.280	71.378	1	4.193888	4.267990	-0.324995	-0.344925
2004	72.866	78.824	1	4.288622	4.367218	-0.082924	-0.096059
2005	74.958	82.968	1	4.316928	4.418455	-0.010594	0.032445
2006	76.798	85.849	1	4.341179	4.452590	0.051373	0.118056
2007	83.654	91.336	1	4.426689	4.514545	0.269876	0.273440
2008	83.814	92.301	1	4.428600	4.525055	0.274759	0.299800
2009	92.488	94.299	1	4.527079	4.546471	0.526400	0.353510
2010	94.493	99.074	1	4.548526	4.595867	0.581203	0.477397
2011	100.106	102.840	1	4.606230	4.633174	0.728652	0.570965
2012	105.455	107.293	1	4.658284	4.675563	0.861666	0.677277
2013	104.244	107.297	1	4.646734	4.675601	0.832153	0.677371
2014	106.429	110.957	1	4.667478	4.709143	0.885159	0.761495
2015	113.260	121.181	1	4.729686	4.797285	1.044118	0.982558
2016	122.398	128.362	1	4.807278	4.854854	1.242387	1.126942
2017	121.601	138.250	1	4.800745	4.929064	1.225694	1.313060
2018	124.794	143.895	1	4.826664	4.969084	1.291924	1.413432
2019	121.350	143.971	1	4.798679	4.969612	1.220414	1.414756
2020	91.500	121.676	1	4.516339	4.801362	0.498956	0.992782

OLS Regression Results

Dep. Variable:	ln_std_K	**R-squared (uncentered):**	0.982
Model:	OLS	**Adj. R-squared (uncentered):**	0.981
Method:	Least Squares	**F-statistic:**	1414.
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	3.40e-24
Time:	13:24:41	**Log-Likelihood:**	15.879
No. Observations:	27	**AIC:**	-29.76
Df Residuals:	26	**BIC:**	-28.46
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
ln_std_I	0.9909	0.026	37.599	0.000	0.937	1.045

Omnibus:	18.786	**Durbin-Watson:**	0.522
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	27.506
Skew:	-1.489	**Prob(JB):**	1.06e-06
Kurtosis:	6.947	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
Коэффициент Детерминаци равен: 98,194% .
Коэфициенты b при каждом члене равны соответственно: 0,9909292700664827 .
Коэффициент 0,9909292700664827 стат. значим, т.к. значение t = 37,5993 больше t_к
ритического = 2,0595 <=> pvalue=3,396455335888668e-24 < 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 1413,7109 больше F_кри
тического <=> fvalue=3,39645533588866e-24 < 0,05
Уравнение нелинейной регрессии:
```

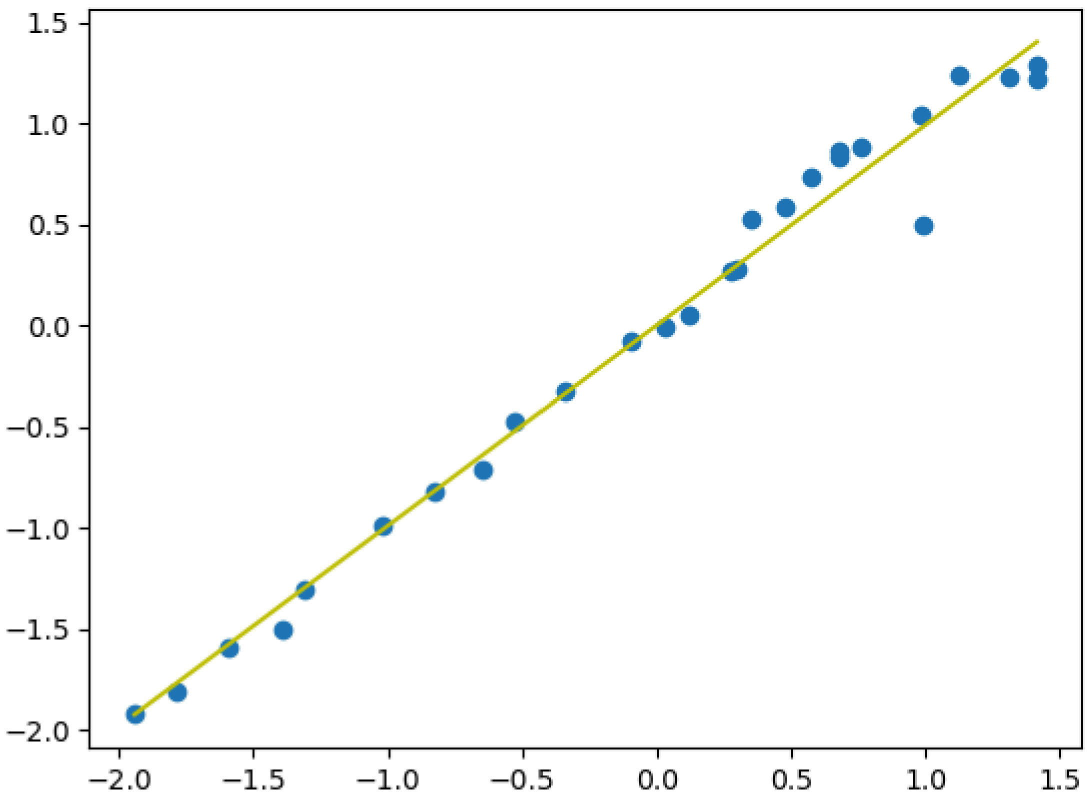
$$K = I_t^{\beta} + \epsilon$$

K = I_{t}^{0.9909} + \epsilon

```
Уравнение линейной регрессии:
```

$$ln(K) = \beta ln(I_t) + \epsilon$$

$$ln(K) = \epsilon + 0.9909 ln(I_t)$$



1. Оцените параметры модели $Q = \alpha L^{\beta_1} K^{\beta_2}$.
2. Оцените качество спецификации модели с помощью теста Фишера. Дайте интерпретацию коэффициента детерминации R^2.
3. Запишите оценённую модель в стандартной форме.
4. Проверьте адекватность линеаризованной модели.

Построим модель на изначальных данных

In [7]:
```
t3=pd.read_excel('Дз 3 нелинейная.xlsx','Лист4',usecols=[0,1,2,3],index_col=0,he

for i in t3.columns:
    t3[f'ln{i}']=np.log(t3[f'{i}'])
t3['1']=[1]*t3.shape[0]
display(t3)
m3 = OLS(t3[['lnQ']],t3[['1','lnL','lnK']]).fit()
display(m3.summary())

a,b_1,b_2,K,L,E= symbols('alpha,beta_1,beta_2, K,L, epsilon')
Q = exp(a)*L**b_1*K**b_2 + E
print(f'Уравнение нелинейной регрессии:')
display(Math('Q = '+latex(Q)))
Q = Q.subs(a,round(m3.params.iloc[0],4))
Q = Q.subs(b_1,round(m3.params.iloc[1],4))
Q = Q.subs(b_2,round(m3.params.iloc[2],4))
```

```
display(Math('Q = '+latex(Q)))


sm_model_outputs(m3,0.05)




print('\n\nКоэффициент при ln(K) не значим по t-критерию стьдента, значит его не
m4 = OLS(t3[['lnQ']],t3[['1','lnL']]).fit()
display(m4.summary())
sm_model_outputs(m4,0.05)

a,b_1,K,L, E= symbols('alpha,beta_1, K,L, epsilon')
Q = exp(a)*L**b_1 + E
print(f'Уравнение нелинейной регрессии:')
display(Math('Q = '+latex(Q)))
Q = Q.subs(a,round(m4.params.iloc[0],4))
Q = Q.subs(b_1,round(m4.params.iloc[1],4))

display(Math('Q = '+latex(Q)))


a,b_1,K,L, E= symbols('alpha,beta_1, K,ln(L), epsilon')
Q = a + L*b_1 + E
print(f'Уравнение линейной регрессии:')
display(Math('ln(Q) = '+latex(Q)))
Q = Q.subs(a,round(m4.params.iloc[0],4))
Q = Q.subs(b_1,round(m4.params.iloc[1],4))
display(Math('ln(Q) = '+latex(Q)))





plt.scatter(np.exp(m4.predict(t3[['1','lnL']].values)),t3[['Q']].values)
plt.plot([t3[['Q']].min(),t3[['Q']].max()],[t3[['Q']].min(),t3[['Q']].max()],col
plt.show()
```

	Q	L	K	lnQ	lnL	lnK	1
Фирма							
1	2350	2334	1570	7.762171	7.755339	7.358831	1
2	2470	2425	1850	7.811973	7.793587	7.522941	1
3	2110	2230	1150	7.654443	7.709757	7.047517	1
4	2560	2463	1940	7.847763	7.809135	7.570443	1
5	2650	2565	2450	7.882315	7.849714	7.803843	1
6	2240	2278	1340	7.714231	7.731053	7.200425	1
7	2430	2380	1700	7.795647	7.774856	7.438384	1
8	2530	2437	1860	7.835975	7.798523	7.528332	1
9	2550	2446	2446	7.843849	7.802209	7.802209	1
10	2450	2403	2403	7.803843	7.784473	7.784473	1
11	2290	2301	2301	7.736307	7.741099	7.741099	1
12	2160	2253	2253	7.677864	7.720018	7.720018	1
13	2400	2367	2367	7.783224	7.769379	7.769379	1
14	2490	2430	2430	7.820038	7.795647	7.795647	1
15	2590	2470	2470	7.859413	7.811973	7.811973	1

```
c:\Users\ivant\anaconda3\Lib\site-packages\scipy\stats\_axis_nan_policy.py:531: U
serWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=15
  res = hypotest_fun_out(*samples, **kwds)
```

OLS Regression Results

Dep. Variable:	lnQ	**R-squared:**	0.965
Model:	OLS	**Adj. R-squared:**	0.959
Method:	Least Squares	**F-statistic:**	163.6
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	1.96e-09
Time:	13:24:41	**Log-Likelihood:**	44.745
No. Observations:	15	**AIC:**	-83.49
Df Residuals:	12	**BIC:**	-81.37
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
1	-5.3417	0.829	-6.443	0.000	-7.148	-3.535
lnL	1.6809	0.116	14.477	0.000	1.428	1.934
lnK	0.0077	0.019	0.415	0.685	-0.033	0.048

Omnibus:	6.644	**Durbin-Watson:**	2.148
Prob(Omnibus):	0.036	**Jarque-Bera (JB):**	3.999
Skew:	-1.241	**Prob(JB):**	0.135
Kurtosis:	3.483	**Cond. No.**	2.58e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.58e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

```
Уравнение нелинейной регрессии:
```

$$Q = K^{\beta_2} L^{\beta_1} e^{\alpha} + \epsilon$$

$$Q = 0.00478772465619674 K^{0.0077} L^{1.6809} + \epsilon$$

```
Коэффициент Детерминаци равен: 96,462% .
Коэфициенты b при каждом члене равны соответственно: -5,341670376619618; 1,680943
2632307555; 0,007704622816963713 .
Коэффициент -5,341670376619618 стат. значим, т.к. значение t = 6,4432 больше t_кр
итического = 2,1788 <=> pvalue=3,1919952969638325e-05 < 0,05
Коэффициент 1,6809432632307555 стат. значим, т.к. значение t = 14,4774 больше t_к
ритического = 2,1788 <=> pvalue=5,826128816733424e-09 < 0,05
Коэффициент 0,007704622816963713 стат. незначим, т.к. значение t = 0,4153 меньше
t_критического = 2,1788 <=> pvalue=0,6852734964090719 > 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 163,6068 больше F_крит
ического <=> fvalue=1,9599620513876233e-09 < 0,05


Коэффициент при ln(K) не значим по t-критерию стьдента, значит его не стоит учиты
вать. Построим модель без него.
```

```
c:\Users\ivant\anaconda3\Lib\site-packages\scipy\stats\_axis_nan_policy.py:531: U
serWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=15
  res = hypotest_fun_out(*samples, **kwds)
```

OLS Regression Results

Dep. Variable:	lnQ	**R-squared:**	0.964
Model:	OLS	**Adj. R-squared:**	0.961
Method:	Least Squares	**F-statistic:**	349.3
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	8.92e-11
Time:	13:24:41	**Log-Likelihood:**	44.638
No. Observations:	15	**AIC:**	-85.28
Df Residuals:	13	**BIC:**	-83.86
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
1	-5.5010	0.711	-7.736	0.000	-7.037	-3.965
lnL	1.7090	0.091	18.689	0.000	1.511	1.907

Omnibus:	7.273	**Durbin-Watson:**	2.073
Prob(Omnibus):	0.026	**Jarque-Bera (JB):**	4.414
Skew:	-1.295	**Prob(JB):**	0.110
Kurtosis:	3.598	**Cond. No.**	1.64e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.64e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

```
Коэффициент Детерминаци равен: 96,412% .
Коэфициенты b при каждом члене равны соответственно: -5,501021987061762; 1,708957
7240081084 .
Коэффициент -5,501021987061762 стат. значим, т.к. значение t = 7,7359 больше t_кр
итического = 2,1604 <=> pvalue=3,223205444753363e-06 < 0,05
Коэффициент 1,7089577240081084 стат. значим, т.к. значение t = 18,6889 больше t_к
ритического = 2,1604 <=> pvalue=8,920073563460382e-11 < 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 349,2753 больше F_крит
ического <=> fvalue=8,920073563460092e-11 < 0,05
Уравнение нелинейной регрессии:
```

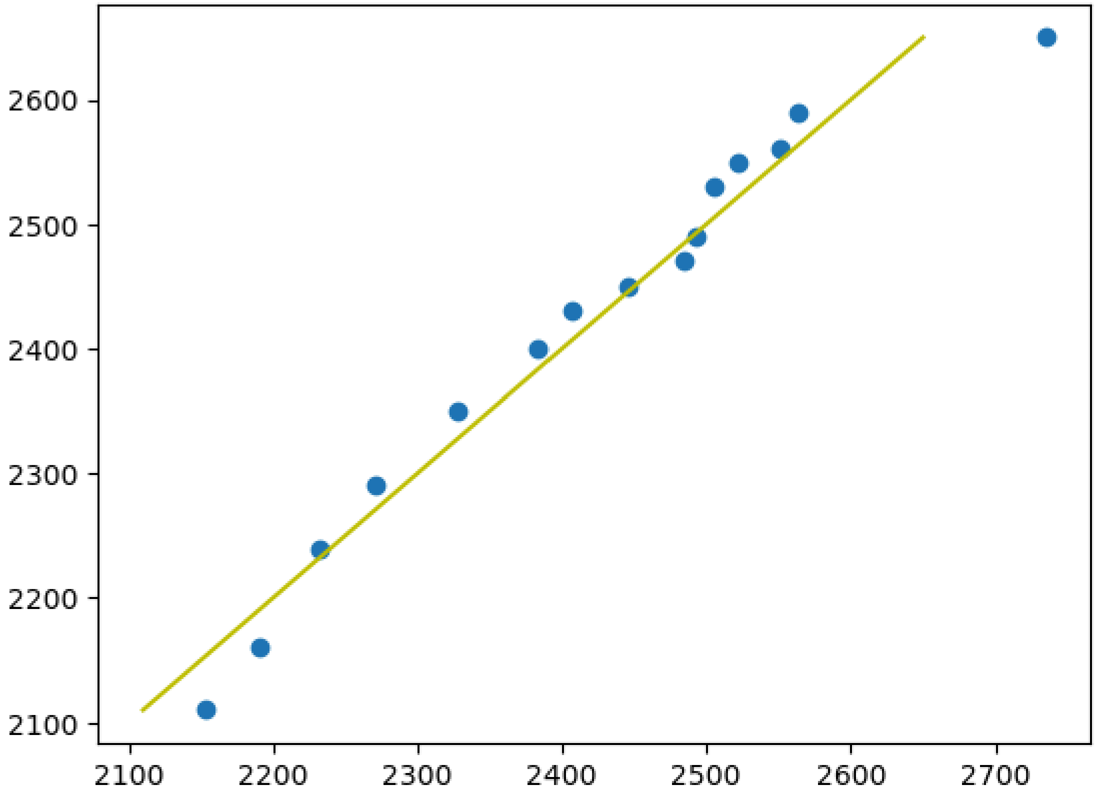
$$Q = L^{\beta_1} e^{\alpha} + \epsilon$$

$$Q = 0.00408268670973036 L^{1.709} + \epsilon$$

```
Уравнение линейной регрессии:
```

$$ln(Q) = \alpha + \beta_1 ln(L) + \epsilon$$

$$ln(Q) = \epsilon + 1.709 ln(L) - 5.501$$



Стандартизируем модель

In [8]:
```
s1=StandardScaler()
t3['ln_std_Q']=s1.fit_transform(t3['lnQ'].values.reshape(-1,1))
t3['ln_std_L']=s1.fit_transform(t3['lnL'].values.reshape(-1,1))
t3['ln_std_K']=s1.fit_transform(t3['lnK'].values.reshape(-1,1))

display(t3)

m5 = OLS(t3[['ln_std_Q']],t3[['ln_std_L','ln_std_K']]).fit()
display(m5.summary())
sm_model_outputs(m5,0.05)

print('\n\nКоэффициент при стандартизированном ln(K) не значим по t-критерию сть
```

```
m6 = OLS(t3[['ln_std_Q']],t3[['ln_std_L']]).fit()
display(m6.summary())
sm_model_outputs(m6,0.05)

b_1,K,L, E= symbols('beta_1, K,L, epsilon')
Q = L**b_1 + E
print(f'Уравнение нелинейной регрессии:')
display(Math('Q = '+latex(Q)))
Q = Q.subs(b_1,round(m6.params.iloc[0],4))

display(Math('Q = '+latex(Q)))
b_1,K,L, E= symbols('beta_1, K,ln(L), epsilon')
Q = L*b_1 + E
print(f'Уравнение линейной регрессии:')
display(Math('ln(Q) = '+latex(Q)))
Q = Q.subs(b_1,round(m6.params.iloc[0],4))
display(Math('ln(Q) = '+latex(Q)))

xx= np.linspace(t3[['ln_std_L']].min(), t3[['ln_std_L']].max(), 100)
yy= m6.predict(xx).reshape(-1, 1)

plt.scatter(t3[['ln_std_L']].values,t3[['ln_std_Q']].values)
plt.plot(xx,yy,c='y')
plt.show()
```

Фирма	Q	L	K	lnQ	lnL	lnK	1	ln_std_Q	ln_std_L	ln_std_
1	2350	2334	1570	7.762171	7.755339	7.358831	1	-0.405738	-0.564020	-0.99980
2	2470	2425	1850	7.811973	7.793587	7.522941	1	0.358718	0.457800	-0.29922
3	2110	2230	1150	7.654443	7.709757	7.047517	1	-2.059317	-1.781771	-2.32879
4	2560	2463	1940	7.847763	7.809135	7.570443	1	0.908069	0.873190	-0.09644
5	2650	2565	2450	7.882315	7.849714	7.803843	1	1.438437	1.957267	0.89993
6	2240	2278	1340	7.714231	7.731053	7.200425	1	-1.141593	-1.212827	-1.67603
7	2430	2380	1700	7.795647	7.774856	7.438384	1	0.108106	-0.042612	-0.66020
8	2530	2437	1860	7.835975	7.798523	7.528332	1	0.727128	0.589675	-0.27621
9	2550	2446	2446	7.843849	7.802209	7.802209	1	0.847992	0.688156	0.89296
10	2450	2403	2403	7.803843	7.784473	7.784473	1	0.233924	0.214325	0.81724
11	2290	2301	2301	7.736307	7.741099	7.741099	1	-0.802734	-0.944443	0.63208
12	2160	2253	2253	7.677864	7.720018	7.720018	1	-1.699824	-1.507640	0.54209
13	2400	2367	2367	7.783224	7.769379	7.769379	1	-0.082576	-0.188938	0.75281
14	2490	2430	2430	7.820038	7.795647	7.795647	1	0.482507	0.512827	0.86494
15	2590	2470	2470	7.859413	7.811973	7.811973	1	1.086902	0.949010	0.93464

```
c:\Users\ivant\anaconda3\Lib\site-packages\scipy\stats\_axis_nan_policy.py:531: U
serWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=15
  res = hypotest_fun_out(*samples, **kwds)
```

OLS Regression Results

Dep. Variable:	ln_std_Q	**R-squared (uncentered):**	0.965
Model:	OLS	**Adj. R-squared (uncentered):**	0.959
Method:	Least Squares	**F-statistic:**	177.2
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	3.69e-10
Time:	13:24:41	**Log-Likelihood:**	3.7788
No. Observations:	15	**AIC:**	-3.558
Df Residuals:	13	**BIC:**	-2.142
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
ln_std_L	0.9658	0.064	15.069	0.000	0.827	1.104
ln_std_K	0.0277	0.064	0.432	0.673	-0.111	0.166

Omnibus:	6.644	**Durbin-Watson:**	2.148
Prob(Omnibus):	0.036	**Jarque-Bera (JB):**	3.999
Skew:	-1.241	**Prob(JB):**	0.135
Kurtosis:	3.483	**Cond. No.**	1.94

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
Коэффициент Детерминаци равен: 96,462% .
Коэфициенты b при каждом члене равны соответственно: 0,9657980186094612; 0,027702
992918669705 .
Коэффициент 0,9657980186094612 стат. значим, т.к. значение t = 15,0685 больше t_к
ритического = 2,1788 <=> pvalue=1,3050315502238635e-09 < 0,05
Коэффициент 0,027702992918669705 стат. незначим, т.к. значение t = 0,4322 меньше
t_критического = 2,1788 <=> pvalue=0,6726602487227937 > 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 177,2407 больше F_крит
ического <=> fvalue=3,6863930154352747e-10 < 0,05


Коэффициент при стандартизированном ln(K) не значим по t-критерию стьдента, значи
т его не стоит учитывать. Построим модель без него.
```

```
c:\Users\ivant\anaconda3\Lib\site-packages\scipy\stats\_axis_nan_policy.py:531: U
serWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=15
  res = hypotest_fun_out(*samples, **kwds)
```

OLS Regression Results

Dep. Variable:	ln_std_Q	**R-squared (uncentered):**	0.964
Model:	OLS	**Adj. R-squared (uncentered):**	0.962
Method:	Least Squares	**F-statistic:**	376.1
Date:	Sun, 20 Oct 2024	**Prob (F-statistic):**	1.63e-11
Time:	13:24:41	**Log-Likelihood:**	3.6718
No. Observations:	15	**AIC:**	-5.344
Df Residuals:	14	**BIC:**	-4.636
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
ln_std_L	0.9819	0.051	19.394	0.000	0.873	1.090

Omnibus:	7.273	**Durbin-Watson:**	2.073
Prob(Omnibus):	0.026	**Jarque-Bera (JB):**	4.414
Skew:	-1.295	**Prob(JB):**	0.110
Kurtosis:	3.598	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
Коэффициент Детерминаци равен: 96,412% .
Коэфициенты b при каждом члене равны соответственно: 0,9818939281520398 .
Коэффициент 0,9818939281520398 стат. значим, т.к. значение t = 19,3944 больше t_к
ритического = 2,1604 <=> pvalue=1,630760045990652e-11 < 0,05
Регрессия стат. значима, т.к. F-значение критерия фишера = 376,1426 больше F_крит
ического <=> fvalue=1,6307600459906518e-11 < 0,05
Уравнение нелинейной регрессии:
```

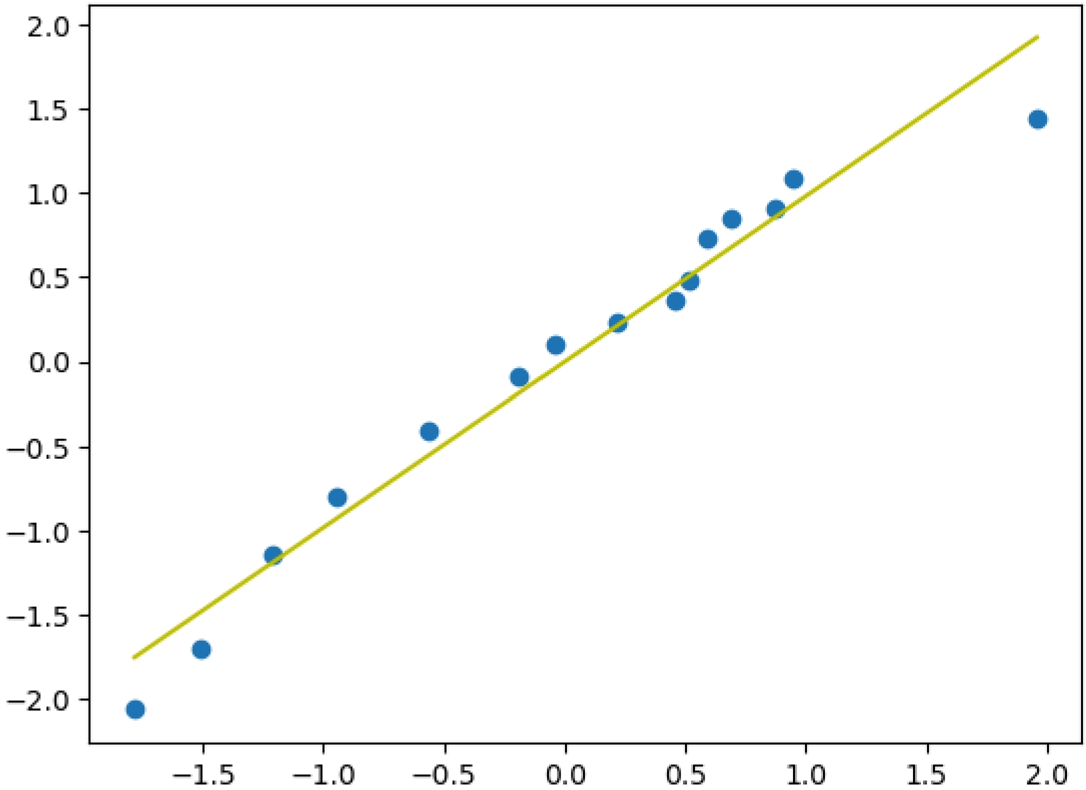
$$Q = L^{\beta_1} + \epsilon$$

$$Q = L^{0.9819} + \epsilon$$

```
Уравнение линейной регрессии:
```

$$ln(Q) = \beta_1 ln(L) + \epsilon$$

$$ln(Q) = \epsilon + 0.9819 ln(L)$$



Задача 18

По 30 заводам, выпускающим продукцию А, изучается зависимость потребления электроэнергии y (тыс. кВт · ч) от производства продукции – x_1 (тыс. ед.) и уровня механизации труда – x_2 (%). Данные приведены в табл. 2.13.

Таблица 2.13

Признак	Среднее значение	Среднее квадратическое отклонение	Парный коэффициент корреляции
y	1000	27	$r_{yx_1} = 0{,}77$
x_1	420	45	$r_{yx_2} = 0{,}43$
x_2	41,5	18	$r_{x_1x_2} = 0{,}38$

Задание

1. Постройте уравнение множественной регрессии в стандартизованном и натуральном масштабе.
2. Определите показатели частной и множественной корреляции.
3. Найдите частные коэффициенты эластичности и сравните их с β-коэффициентами.
4. Рассчитайте общий и частные F-критерии Фишера.

$$\mathbb{E}(Y) = 1000$$
$$\mathbb{E}(X1) = 420$$
$$\mathbb{E}(X2) = 41,5$$

Построим матрицу парных корреляций :

1		
$r_{Y\ X1}$	1	
$r_{Y\ X2}$	$r_{X1\ X2}$	1

1		
$0,77$	1	
$0,43$	$0,38$	1

Построим матрицы :

$$R_X = \begin{pmatrix} 1 & r_{X1\ X2} \\ r_{X1\ X2} & 1 \end{pmatrix}$$

$$r_Y = \begin{pmatrix} r_{Y\ X1} \\ r_{Y\ X2} \end{pmatrix}$$

In [9]:
```
np.set_printoptions(suppress=True)
################################################################################
#ДАННЫЕ
################################################################################
n_res=30
R = np.matrix([[1,0.77,0.43],[0.77,1,0.38],[0.43,0.38,1]])
S= np.matrix([[27],[45],[18]])
E = np.matrix([[1000],[420],[41.5]])
################################################################################
#ПОДСЧЕТ ИСХОДЯ ИЗ ДАННЫХ
################################################################################
RX = R[1:,1:]
r= R[1:,0]
#1
b = (RX**-1)@r
beta = (RX**-1)@r
b=np.multiply(b*S[0],1/S[1:])
b_0=E[0]-np.sum(np.multiply(b[0:],E[1:]))
b = np.matrix(np.vstack((np.array(b_0),np.array(b))))
#2
partial_corr_matrix=np.matrix([[ ((-R**-1)[i,j]/np.sqrt((R**-1)[i,i]*(R**-1)[j,j
R_mnoj=np.sqrt(r.T@(RX**-1)@r)[0,0]
#3
elas=np.multiply(b[1:]/E[0],E[1:])
compares=np.where(elas>beta[1:])
#4
fisher=np.array([[ partial_corr_matrix[i,j]**2*(n_res-E.shape[0])/((1-partial_cc
```

```
############################################################################
#ВЫВОД ИНФОРМАЦИИ
############################################################################
display(Latex('$№1$'))
display(Latex('$Построим~уравнение~регрессии~в~натуральном~масштабе:$'))
y1 = '$y = '+''.join(['+' + str(b[i,0])+f'x_{i}' if str(b[i,0])[0]!='-' else str
display(Latex(y1))
display(Latex('$Построим~уравнение~регрессии~в~стандартизированном~масштабе:$'))
y = '$y = '+''.join(['+' + str(beta[i,0])+f'x_{i+1}' if str(beta[i,0])[0]!='-' e
display(Latex(y))
############################################################################
display(Latex('$№2$'))
display(Latex('''
Формула для частной корреляции между переменными $X_i $ и $ X_j $ с учётом всех

$r_{ij \cdot other} = -\Omega_{ij}/\sqrt{\Omega_{ii} \Omega_{jj}}$

Где $ \Omega $ — это обратная корреляционная матрица.

По этой формуле посчитаем все частные корреляции:

'''))
for i in range(partial_corr_matrix.shape[0]):
    for j in range(i,partial_corr_matrix.shape[1]):
        if i!=j:
            if i==0:
                display(Math(('r_{Y'+f'X_{j}'+'}=') + latex(partial_corr_matrix[
            else:
                display(Math(('r_{'+f'X_{i}'+f'X_{j}'+'}=') + latex(partial_corr
display(Latex(f'$Коэффициент~множественной~корреляции~равен:~~{R_mnoj}~$'))
############################################################################
display(Latex('$№3$'))
for i in range(elas.shape[0]):
    display(Math(f'Частный~коэффициент~элатичности~при~X_{i+1}~=~'+ latex(elas[i
    display(Latex(f'$Этот~коэффициент~{"больше" if compares[0] else "меньше"}~че
############################################################################
display(Latex('$№4$'))
for i in range(fisher.shape[0]):
    if i==0:
        display(Math(('F_{общий}=') + latex(fisher[i])))
    else:
        display(Math(('F_{'+f'X_{i}'+'}=') + latex(fisher[i])))
```

№1

Построим уравнение регрессии в натуральном масштабе :

$y = 811.3413394109397 + 0.4253856942496494x_1 + 0.24088359046283303x_2$

Построим уравнение регрессии в стандартизированном масштабе :

$y = 0.708976157082749x_1 + 0.16058906030855535x_2$

№2

Формула для частной корреляции между переменными X_i и X_j с учётом всех остальных переменных выглядит следующим образом: $r_{ij \cdot other} = -\Omega_{ij}/\sqrt{\Omega_{ii}\Omega_{jj}}$ Где Ω — это обратная корреляционная матрица. По этой формуле посчитаем все частные корреляции:

$r_{YX_1} = 0.72637614236143$

$r_{YX_2} = 0.232809551177815$

$r_{X_1X_2} = 0.084889282207865$

$\text{Коэффициент множественной корреляции равен :} \quad 0.7841970013245367$

№3

Частный коэффициент элатичности при $X_1 \; = \; 0.178661991584853$

Этот коэффициент меньше чем коэффициент β_{X_1}

Частный коэффициент элатичности при $X_2 \; = \; 0.00999666900420757$

Этот коэффициент меньше чем коэффициент β_{X_2}

№4

$F_{\text{общий}} = 21.5617418861327$

$F_{X_1} = 15.0788258113867$

$F_{X_2} = 0.773635121844607$