In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

Загрузка датасета

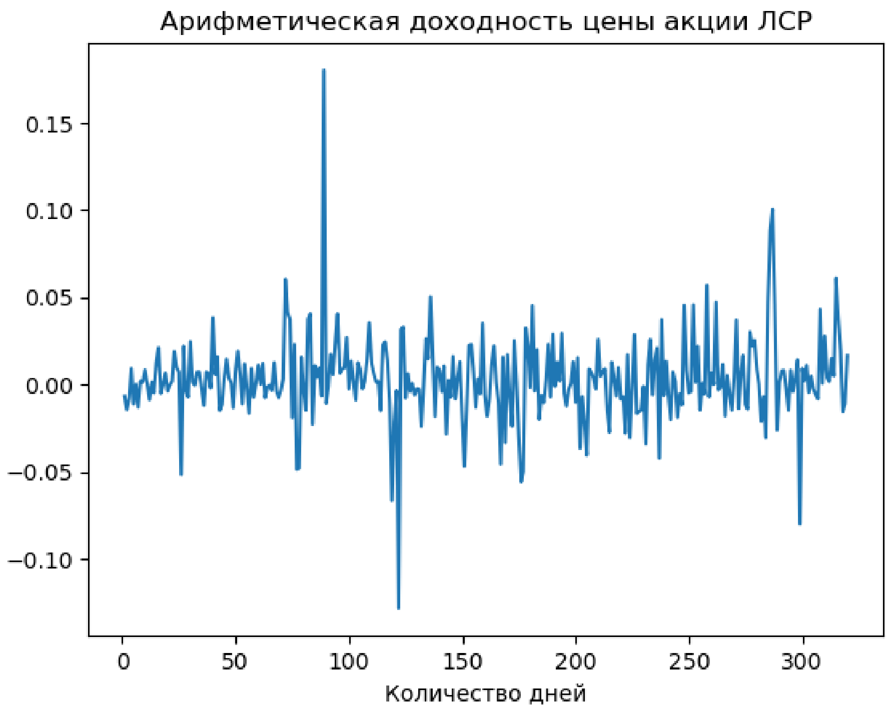
In [26]:
```
data = pd.read_excel('данные для дз.xlsx')
data = data.drop('Unnamed: 5',axis=1)
data
```

Out[26]:

	Дата	ЛСР	USD_RUB	ср.знач	дисп	ареф.ЛСР	геом.ЛСР	ареф.дол	геом.долл
0	2023-01-11	472.8	70.3002	NaN	NaN	NaN	NaN	NaN	NaN
1	2023-01-12	469.6	69.6094	471.200000	0.238602	-0.006768	-0.006791	-0.009826	-0.009875
2	2023-01-13	462.8	69.0202	468.400000	0.410460	-0.014480	-0.014586	-0.008464	-0.008500
3	2023-01-14	459.0	67.7775	466.050000	1.143911	-0.008211	-0.008245	-0.018005	-0.018169
4	2023-01-17	463.2	67.5744	465.480000	1.371487	0.009150	0.009109	-0.002997	-0.003001
...	...	...	...	...	...	...	...	...	...
316	2024-04-10	1126.0	92.7463	686.326183	72.979393	0.038362	0.037645	0.001785	0.001784
317	2024-04-11	1148.0	93.2198	687.777987	72.885484	0.019538	0.019350	0.005105	0.005092
318	2024-04-12	1130.0	93.7196	689.164263	72.812663	-0.015679	-0.015804	0.005362	0.005347
319	2024-04-13	1117.6	93.4419	690.503125	72.727344	-0.010973	-0.011034	-0.002963	-0.002967
320	2024-04-16	1136.0	93.5891	691.890966	72.647923	0.016464	0.016330	0.001575	0.001574

321 rows × 9 columns

In [36]:
```
plt.plot(data['ареф.ЛСР'])
plt.title('Арифметическая доходность цены акции ЛСР')
plt.xlabel('Количество дней')
plt.show()
```



Судя по графику, наш ряд стационарен. Докажем это

In [59]:
```
from statsmodels.tsa.stattools import adfuller

print('Результат теста:')
df_result = adfuller(data['ареф.ЛСР'][1:])
print(f'Значение p_value: {df_result[1]}')

if df_result[1] <= 0.05:
    print("ряд является стационарным.")
else:
    print("ряд не является стационарным.")
```

Результат теста:
Значение p_value: 1.7242023195479813e-29
ряд является стационарным.

In [38]:
```
plt.plot(data['ЛСР'])
plt.title('Изменение цены акции ЛСР')
plt.xlabel('Количество дней')
```

```
plt.ylabel('Цена акции')
plt.show()
```

ACF и PACF

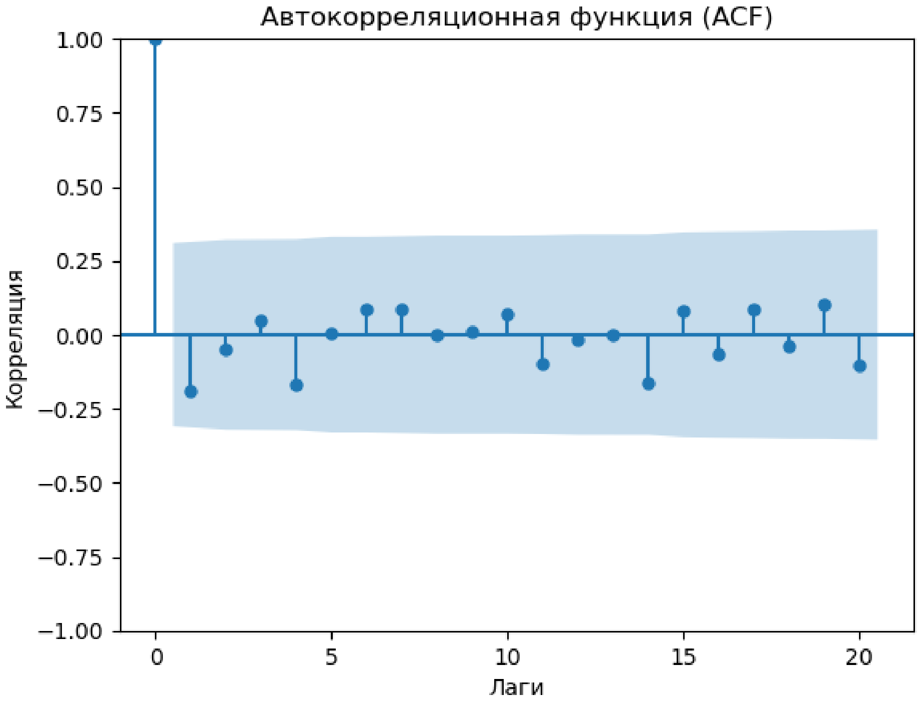
In [27]:
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

In [47]:
```
data_areth = data['ареф.ЛСР'][1:41]
data_areth
```

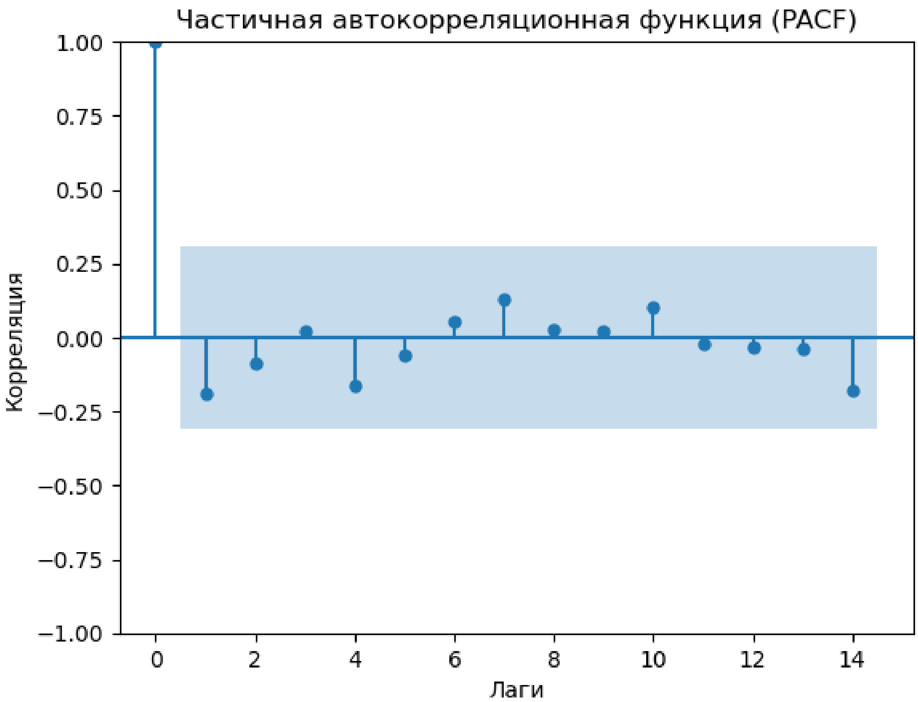
Out[47]:
```
1    -0.006768
2    -0.014480
3    -0.008211
4     0.009150
5    -0.011226
6     0.000000
7    -0.013100
8     0.001770
9     0.001325
10    0.008381
11    0.000437
12   -0.008745
13    0.001323
14   -0.004846
15    0.012395
16    0.020988
17   -0.005139
18   -0.000430
19    0.006460
20   -0.003851
21   -0.000430
22    0.001719
23    0.018876
24    0.009684
25    0.007089
26   -0.051760
27    0.021834
28   -0.003419
29   -0.007290
30    0.024622
31    0.001265
32   -0.000421
33    0.007161
34    0.007110
35   -0.002076
36   -0.012068
37    0.007161
38    0.006274
39   -0.002078
40    0.038317
Name: ареф.ЛСР, dtype: float64
```

In [48]:
```
plot_acf(data_areth, lags = 20)
plt.title('Автокорреляционная функция (ACF)')
plt.xlabel('Лаги')
```

```
plt.ylabel('Корреляция')
plt.show()
```



In [49]:
```
plot_pacf(data_areth, lags=14, method='ywm')
plt.title('Частичная автокорреляционная функция (PACF)')
plt.xlabel('Лаги')
plt.ylabel('Корреляция')
plt.show()
```



Построение моделей

MA

In [50]:
```
from statsmodels.tsa.arima.model import ARIMA
```

In [51]:
```
lsr_data = data['ареф.ЛСР'][1:]
lsr_data
```

```
Out[51]:1     -0.006768
        2     -0.014480
        3     -0.008211
        4      0.009150
        5     -0.011226
                 ...
        316    0.038362
        317    0.019538
        318   -0.015679
        319   -0.010973
        320    0.016464
        Name: ареф.ЛСР, Length: 320, dtype: float64
```

```
In [60]: model_ma1 = ARIMA(lsr_data, order=(0, 0, 1)) # Параметры p,d,q. p-порядок авторегресии, d-порядок интегрирования, q-порядок скользяще
         result_ma1 = model_ma1.fit()

         result_ma1.summary()
```

Out[60]:

SARIMAX Results

Dep. Variable:	ареф.ЛСР	**No. Observations:**	320
Model:	ARIMA(0, 0, 1)	**Log Likelihood**	733.107
Date:	Tue, 30 Apr 2024	**AIC**	-1460.215
Time:	22:35:27	**BIC**	-1448.910
Sample:	0	**HQIC**	-1455.701
	- 320		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0030	0.001	2.026	0.043	9.85e-05	0.006
ma.L1	0.0663	0.043	1.528	0.127	-0.019	0.151
sigma2	0.0006	1.88e-05	31.789	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1767.64
Prob(Q):	0.98	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.06	**Skew:**	0.84
Prob(H) (two-sided):	0.78	**Kurtosis:**	14.39

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:

$y_t = 0.0030 + u_t + 0.0663*u_{t-1}$

```
In [64]: model_ma2 = ARIMA(lsr_data, order=(0, 0, 2))
         result_ma2 = model_ma2.fit()

         result_ma2.summary()
```

Out[64]: SARIMAX Results

Dep. Variable:	apeф.ЛСР	**No. Observations:**	320
Model:	ARIMA(0, 0, 2)	**Log Likelihood**	733.164
Date:	Tue, 30 Apr 2024	**AIC**	-1458.329
Time:	22:41:56	**BIC**	-1443.256
Sample:	0	**HQIC**	-1452.310
	- 320		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0030	0.002	1.980	0.048	3.1e-05	0.006
ma.L1	0.0633	0.044	1.426	0.154	-0.024	0.150
ma.L2	0.0257	0.060	0.430	0.667	-0.091	0.143
sigma2	0.0006	1.93e-05	31.047	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.01	**Jarque-Bera (JB):**	1747.47
Prob(Q):	0.94	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.05	**Skew:**	0.84
Prob(H) (two-sided):	0.78	**Kurtosis:**	14.32

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:
$y_t = 0.0030 + u_t + 0.0663*u_{t-1} + 0.0257 * u_{t-2}$

AR модели

In [67]:
```
model_ar1 = ARIMA(lsr_data, order=(1, 0, 0))
result_ar1 = model_ar1.fit()

result_ar1.summary()
```

Out[67]: SARIMAX Results

Dep. Variable:	apeф.ЛСР	**No. Observations:**	320
Model:	ARIMA(1, 0, 0)	**Log Likelihood**	733.132
Date:	Tue, 30 Apr 2024	**AIC**	-1460.263
Time:	22:47:19	**BIC**	-1448.958
Sample:	0	**HQIC**	-1455.749
	- 320		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0030	0.002	2.011	0.044	7.78e-05	0.006
ar.L1	0.0689	0.042	1.626	0.104	-0.014	0.152
sigma2	0.0006	1.89e-05	31.724	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1762.75
Prob(Q):	0.99	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.06	**Skew:**	0.84
Prob(H) (two-sided):	0.78	**Kurtosis:**	14.37

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:
$y_t = 0.0030 + 0.0689*y_{t-1} + \epsilon_t$

In [68]:
```
model_ar2 = ARIMA(lsr_data, order=(2, 0, 0))
result_ar2 = model_ar2.fit()

result_ar2.summary()
```

Out[68]: SARIMAX Results

Dep. Variable:	ареф.ЛСР	**No. Observations:**	320
Model:	ARIMA(2, 0, 0)	**Log Likelihood**	733.163
Date:	Tue, 30 Apr 2024	**AIC**	-1458.327
Time:	22:51:25	**BIC**	-1443.253
Sample:	0	**HQIC**	-1452.308
	- 320		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0030	0.002	1.976	0.048	2.48e-05	0.006
ar.L1	0.0671	0.044	1.519	0.129	-0.019	0.154
ar.L2	0.0141	0.060	0.236	0.814	-0.103	0.132
sigma2	0.0006	1.95e-05	30.689	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1749.83
Prob(Q):	0.99	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.05	**Skew:**	0.84
Prob(H) (two-sided):	0.78	**Kurtosis:**	14.33

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:

$y_t = 0.0030 + 0.0671*y_{t-1} + 0.0141*y_{t-2} + \epsilon_t$

ARMA модели

In [71]:
```
model_arma11 = ARIMA(lsr_data, order=(1, 0, 1))
result_arma11 = model_arma11.fit()

result_arma11.summary()
```

Out[71]: SARIMAX Results

Dep. Variable:	ареф.ЛСР	**No. Observations:**	320
Model:	ARIMA(1, 0, 1)	**Log Likelihood**	733.148
Date:	Tue, 30 Apr 2024	**AIC**	-1458.296
Time:	22:57:40	**BIC**	-1443.223
Sample:	0	**HQIC**	-1452.277
	- 320		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0030	0.002	1.946	0.052	-2.25e-05	0.006
ar.L1	0.2387	0.784	0.305	0.761	-1.297	1.774
ma.L1	-0.1724	0.797	-0.216	0.829	-1.735	1.390
sigma2	0.0006	1.99e-05	30.089	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1750.24
Prob(Q):	0.97	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.06	**Skew:**	0.85
Prob(H) (two-sided):	0.78	**Kurtosis:**	14.33

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:

$y_t = 0.0030 + 0.2387*y_{t-1} + u_t - 0.1724*u_{t-1}$

In [72]:
```
model_arma12 = ARIMA(lsr_data, order=(1, 0, 2))
result_arma12 = model_arma12.fit()

result_arma12.summary()
```

Out[72]: SARIMAX Results

Dep. Variable:	apeф.ЛСР	**No. Observations:**	320
Model:	ARIMA(1, 0, 2)	**Log Likelihood**	733.108
Date:	Tue, 30 Apr 2024	**AIC**	-1456.215
Time:	22:57:41	**BIC**	-1437.374
Sample:	0	**HQIC**	-1448.692
	- 320		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0030	0.002	1.978	0.048	2.77e-05	0.006
ar.L1	0.5962	41.604	0.014	0.989	-80.947	82.139
ma.L1	-0.5298	41.611	-0.013	0.990	-82.086	81.027
ma.L2	-0.0405	2.730	-0.015	0.988	-5.391	5.310
sigma2	0.0006	2.01e-05	29.875	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1769.17
Prob(Q):	0.98	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.06	**Skew:**	0.84
Prob(H) (two-sided):	0.78	**Kurtosis:**	14.40

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:

$y_t = 0.0030 + 0.5962*y_{t-1} + u_t - 0.5298*u_{t-1} - 0.0405*u_{t-2}$

In [73]:

```
model_arma21 = ARIMA(lsr_data, order=(2, 0, 1))
result_arma21 = model_arma21.fit()

result_arma21.summary()
```

Out[73]: SARIMAX Results

Dep. Variable:	apeф.ЛСР	**No. Observations:**	320
Model:	ARIMA(2, 0, 1)	**Log Likelihood**	733.140
Date:	Tue, 30 Apr 2024	**AIC**	-1456.280
Time:	22:59:52	**BIC**	-1437.438
Sample:	0	**HQIC**	-1448.756
	- 320		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0030	0.002	1.890	0.059	-0.000	0.006
ar.L1	-0.4263	6.786	-0.063	0.950	-13.726	12.873
ar.L2	0.0406	0.450	0.090	0.928	-0.841	0.922
ma.L1	0.4952	6.793	0.073	0.942	-12.819	13.810
sigma2	0.0006	1.91e-05	31.356	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1757.01
Prob(Q):	0.98	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.05	**Skew:**	0.84
Prob(H) (two-sided):	0.79	**Kurtosis:**	14.36

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:

$y_t = 0.0030 - 0.4263*y_{t-1} + 0.0406*y_{t-2} + u_t + 0.4952*u_{t-1}$

In [74]:

```
model_arma22 = ARIMA(lsr_data, order=(2, 0, 2))
result_arma22 = model_arma22.fit()
```

```
result_arma22.summary()
```

Out[74]:

SARIMAX Results

Dep. Variable:	ареф.ЛСР	**No. Observations:**	320
Model:	ARIMA(2, 0, 2)	**Log Likelihood**	735.480
Date:	Tue, 30 Apr 2024	**AIC**	-1458.959
Time:	23:02:34	**BIC**	-1436.349
Sample:	0	**HQIC**	-1449.931
	- 320		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	0.0030	0.002	1.954	0.051	-8.86e-06	0.006
ar.L1	1.2130	0.135	8.997	0.000	0.949	1.477
ar.L2	-0.8895	0.122	-7.318	0.000	-1.128	-0.651
ma.L1	-1.1506	0.137	-8.371	0.000	-1.420	-0.881
ma.L2	0.8566	0.130	6.571	0.000	0.601	1.112
sigma2	0.0006	2.8e-05	21.058	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.01	**Jarque-Bera (JB):**	1529.91
Prob(Q):	0.94	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.05	**Skew:**	0.78
Prob(H) (two-sided):	0.81	**Kurtosis:**	13.60

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:
$y_t = 0.0030 + 1.2130*y_{t-1} - 0.8895*y_{t-2} + u_t - 1.1506*u_{t-1} + 0.8566*u_{t-2}$

ARIMA модели

In [80]:
```
!pip install pmdarima
```

```
Collecting pmdarima
  Downloading pmdarima-2.0.4-cp39-cp39-win_amd64.whl (614 kB)
     ------------------------------------ 615.0/615.0 kB 921.9 kB/s eta 0:00:00
Requirement already satisfied: scipy>=1.3.2 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (1.9.1)
Requirement already satisfied: pandas>=0.19 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (1.4.4)
Requirement already satisfied: Cython!=0.29.18,!=0.29.31,>=0.29 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (0.29.32)
Requirement already satisfied: setuptools!=50.0.0,>=38.6.0 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (63.4.1)
Requirement already satisfied: packaging>=17.1 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (21.3)
Requirement already satisfied: urllib3 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (1.26.11)
Requirement already satisfied: scikit-learn>=0.22 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (1.0.2)
Requirement already satisfied: joblib>=0.11 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (1.1.0)
Requirement already satisfied: statsmodels>=0.13.2 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (0.13.2)
Requirement already satisfied: numpy>=1.21.2 in c:\users\arsen\anaconda3\lib\site-packages (from pmdarima) (1.21.5)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\arsen\anaconda3\lib\site-packages (from packaging>=17.1->pmdarima) (3.0.9)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\arsen\anaconda3\lib\site-packages (from pandas>=0.19->pmdarima) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\arsen\anaconda3\lib\site-packages (from pandas>=0.19->pmdarima) (2022.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\arsen\anaconda3\lib\site-packages (from scikit-learn>=0.22->pmdarima) (2.2.0)
Requirement already satisfied: patsy>=0.5.2 in c:\users\arsen\anaconda3\lib\site-packages (from statsmodels>=0.13.2->pmdarima) (0.5.2)
Requirement already satisfied: six in c:\users\arsen\anaconda3\lib\site-packages (from patsy>=0.5.2->statsmodels>=0.13.2->pmdarima) (1.16.0)
Installing collected packages: pmdarima
Successfully installed pmdarima-2.0.4
```

In [95]:
```
import pmdarima as pm
# Библиотека pmdarima автоматически подбирает лучшую модель
model = pm.auto_arima(data['ЛСР'],seasonal=False,trace=True)
model
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=1.23 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2758.382, Time=0.04 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2755.946, Time=0.14 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2756.151, Time=0.27 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2760.644, Time=0.03 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=2757.860, Time=0.29 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=2757.887, Time=0.33 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=2759.846, Time=0.99 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=2757.292, Time=0.08 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0] intercept
Total fit time: 3.420 seconds
```

Out[95]:ARIMA(order=(1, 1, 0), scoring_args={}, suppress_warnings=True)

В предыдущих моделях мы использовали уже готовый стационарный ряд (арифметическая доходность ЛСР), а в модели ARIMA я использовал изначальный ряд, то есть цена акции ЛСР, так как модель ARIMA сама по себе делает из ряда стационарный ряд

In [96]:
```
model.summary()
```

Out[96]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	321
Model:	SARIMAX(1, 1, 0)	**Log Likelihood**	-1374.973
Date:	Tue, 30 Apr 2024	**AIC**	2755.946
Time:	23:28:17	**BIC**	2767.251
Sample:	0	**HQIC**	2760.460
	- 321		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
intercept	1.8402	1.021	1.802	0.072	-0.162	3.842
ar.L1	0.1173	0.037	3.181	0.001	0.045	0.190
sigma2	315.9224	10.842	29.138	0.000	294.672	337.173

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	1108.51
Prob(Q):	0.98	**Prob(JB):**	0.00
Heteroskedasticity (H):	2.26	**Skew:**	0.11
Prob(H) (two-sided):	0.00	**Kurtosis:**	12.12

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Модель:
$y_t = 1.8402 + 1.2130*y_{t-1}$

Критерий Акаике. Лучшая модель.

In [104]:
```
import numpy as np
n=321
ak_ma1= np.log(0.0006) + 2*(1+0+1)/n
ak_ma2 = np.log(0.0006) + 2*(2+0+1)/n
ak_ar1 = np.log(0.0006) + 2*(0+1+1)/n
ak_ar2 = np.log(0.0006) + 2*(0+2+1)/n
ak_arma11 = np.log(0.0006) + 2*(1+1+1)/n
ak_arma12 = np.log(0.0006) + 2*(1+2+1)/n
ak_arma21 = np.log(0.0006) + 2*(2+1+1)/n
ak_arma22 = np.log(0.0006) + 2*(2+2+1)/n
akaike = pd.DataFrame({'Модель': ['AR(1)','AR(2)', 'MA(1)', 'MA(2)','ARMA(1,1)','ARMA(1,2)',"ARMA(2,1)",'ARMA(2,2)'],
                       'AIC': [ak_ar1,ak_ar2,ak_ma1,ak_ma2,ak_arma11,ak_arma12,ak_arma21,ak_arma22]})
akaike
```

Out[104]:

	Модель	AIC
0	AR(1)	-7.406120
1	AR(2)	-7.399889
2	MA(1)	-7.406120
3	MA(2)	-7.399889
4	ARMA(1,1)	-7.399889
5	ARMA(1,2)	-7.393659
6	ARMA(2,1)	-7.393659
7	ARMA(2,2)	-7.387428

In [109]:
```
akaike.loc[akaike["AIC"].idxmin()]
```

Out[109]:
```
Модель      AR(1)
AIC      -7.40612
Name: 0, dtype: object
```

По результатам критерия Акаике, можно сделать вывод, что AR(1) является лучшей моделью

Прогноз данных

In [111]:
```
from statsmodels.tsa.arima.model import ARIMA

forecast = result_ar1.forecast(steps=5)



# Вывод прогноза
print("Прогноз цен акций на следующие 5 дней:")
print(forecast)
```

```
Прогноз цен акций на следующие 5 дней:
321    0.003963
322    0.003101
323    0.003042
324    0.003038
325    0.003037
Name: predicted_mean, dtype: float64
```

In [112]:
```
# Первая цена акции (используется для прогноза)
initial_price = data['ЛСР'].iloc[-1]

# Преобразование прогноза арифметической доходности в цены акций
forecast_prices = [initial_price * (1 + return_) for return_ in forecast]

# Вывод прогноза цен акций на следующие 5 дней
print("Прогноз цен акций на следующие 5 дней:")
for i, price in enumerate(forecast_prices, start=1):
    print(f"День {i}: {price}")
```

```
Прогноз цен акций на следующие 5 дней:
День 1: 1140.5018561293696
День 2: 1139.5229528543214
День 3: 1139.4554754717894
День 4: 1139.4508241469146
День 5: 1139.4505035235845
```

И так, мы спрогнозировали цены акций ЛСР на последуюшие 5 дней, то есть с 17.04.2024 на 21.04.2024

In [ ]: