Лабораторная работа №6

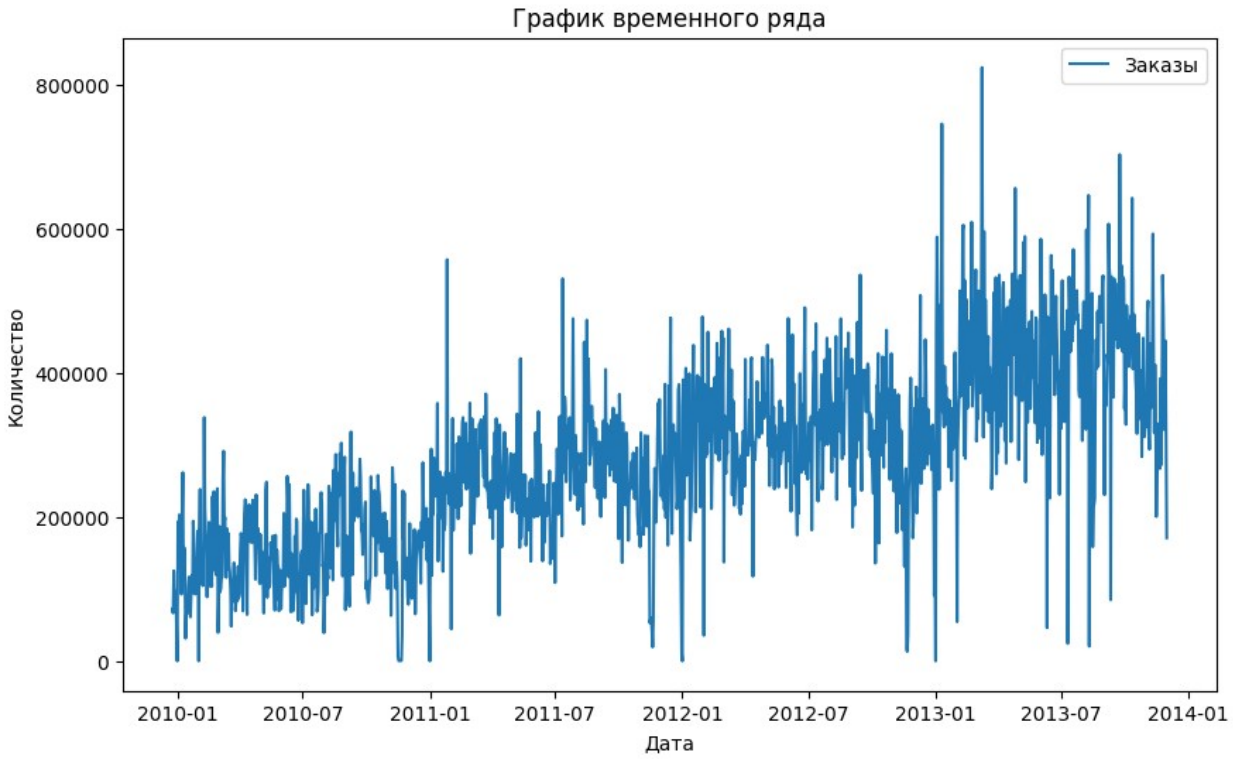
Временные ряды

```
import pandas as pd
import numpy as np
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt
import statsmodels.api as sm

from datetime import datetime

df = pd.read_csv("data/tovar_moving.csv", index_col=['date'],
parse_dates=['date'], dayfirst=True) # считываем датасет
df.head()
```

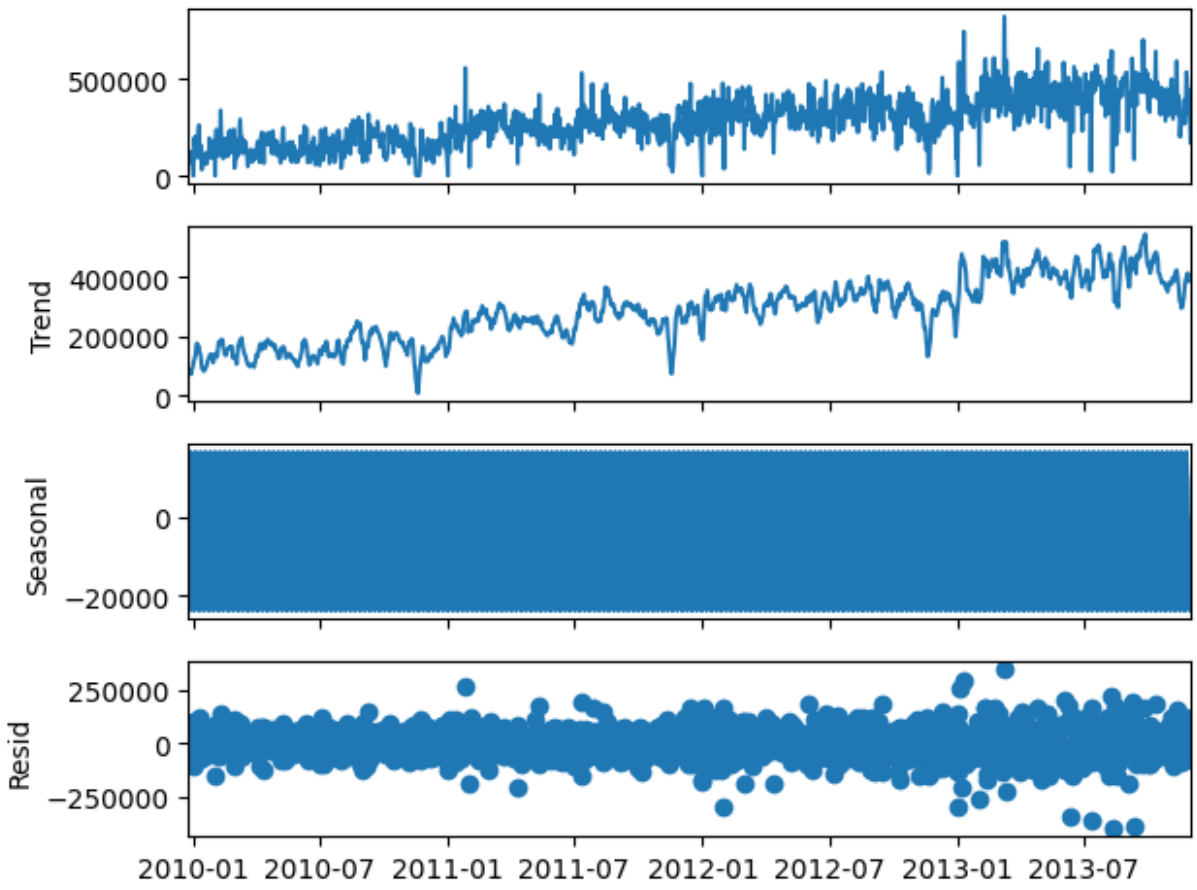
```
C:\Users\marin\AppData\Local\Temp\ipykernel_23536\4252153029.py:9:
UserWarning: Parsing dates in %Y-%m-%d format when dayfirst=True was
specified. Pass `dayfirst=False` or specify a format to silence this
warning.
  df = pd.read_csv("data/tovar_moving.csv", index_col=['date'],
parse_dates=['date'], dayfirst=True) # считываем датасет
```

```
                qty
date
2009-12-25   72314.0
2009-12-26   66586.0
2009-12-27  125199.0
2009-12-28   91544.0
2009-12-29   76995.0
```

```
test_df = df.iloc[-1]
train_df = df.iloc[:-1]

plt.figure(figsize=(10, 6))
plt.plot(train_df.index, train_df['qty'], label='Заказы')  # Access
'date' using index
plt.xlabel('Дата')
plt.ylabel('Количество')
plt.title('График временного ряда')
plt.legend()
plt.show()
```




```
decomposition = seasonal_decompose(train_df, model='additive')
decomposition.plot()
pyplot.show() # любуемся результатом
```




```
from statsmodels.tsa.api import SimpleExpSmoothing
ses = SimpleExpSmoothing(train_df)
alpha = 0.7
model = ses.fit(smoothing_level = alpha, optimized = False)
exp_pred = model.forecast(1)
print("Forecast: {:.1f}".format(exp_pred.iloc[0]))
print("Actual: ", test_df)
```

```
Forecast: 225015.5
Actual:  qty    423846.0
Name: 2013-12-02 00:00:00, dtype: float64
```

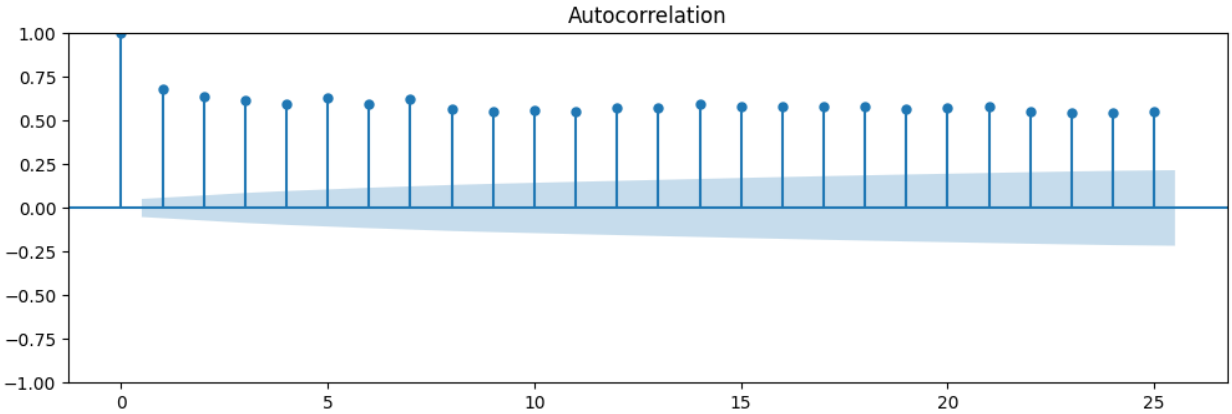
```
c:\Users\marin\AppData\Local\Programs\Python\Python39\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be
used.
  self._init_dates(dates, freq)
```

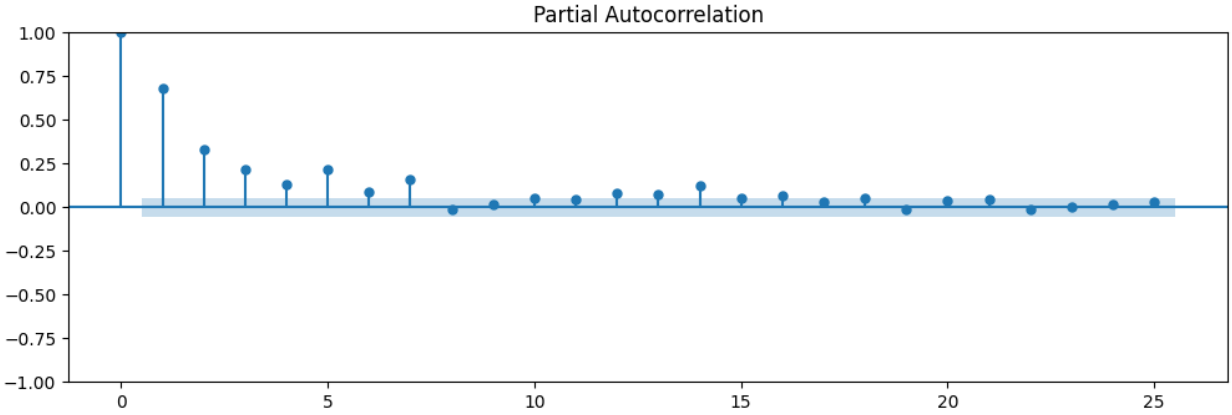
```
def stat_test(df):
    test = sm.tsa.adfuller(df)
    #print ('adf: ', test[0] )
    #print ('p-value: ', test[1])
    #print('Critical values: ', test[4])
```

```
    if test[0]> test[4]['5%']:
        return False
    else:
        return True

test = df
order = 0
while not stat_test(test):
    test = test.diff().dropna()
    order += 1
print(f"Порядок интегрирования: {order}")

if stat_test(test) == False:
    print('есть единичные корни, ряд не стационарен')
else:
    print ('единичных корней нет, ряд стационарен')
```

```
Порядок интегрирования: 1
единичных корней нет, ряд стационарен
```

```
from matplotlib import pyplot as plt
%matplotlib inline

fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(train_df.values.squeeze(), lags=25,
ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(train_df, lags=25, ax=ax2)
```






```
import pandas as pd
import numpy as np
from statsmodels.tsa.ar_model import AutoReg

ar_model = AutoReg(train_df, lags=7).fit()
print(ar_model.summary())

ar_pred = ar_model.predict(start=len(train_df), end=(len(train_df)),
dynamic=False)
ar_pred
```

```
                            AutoReg Model Results

==============================================================================
========
Dep. Variable:                      qty   No. Observations:
1438
Model:                     AutoReg(7)   Log Likelihood                -
18187.904
Method:                Conditional MLE   S.D. of innovations
80096.504
Date:                Thu, 06 Jun 2024   AIC
36393.808
Time:                        13:18:59   BIC
```

```
36441.203
Sample:                     01-01-2010   HQIC
36411.506
                          - 12-01-2013

==============================================================================
                 coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
const       2.606e+04   6125.181      4.254      0.000    1.41e+04
3.81e+04
qty.L1         0.2917      0.026     11.169      0.000       0.241
0.343
qty.L2         0.1248      0.027      4.585      0.000       0.071
0.178
qty.L3         0.1008      0.027      3.727      0.000       0.048
0.154
qty.L4         0.0199      0.027      0.731      0.465      -0.033
0.073
qty.L5         0.1659      0.027      6.128      0.000       0.113
0.219
qty.L6         0.0374      0.027      1.374      0.169      -0.016
0.091
qty.L7         0.1692      0.026      6.466      0.000       0.118
0.220
                                    Roots

=============================================================================
                  Real          Imaginary           Modulus
Frequency
-----------------------------------------------------------------------------
AR.1            1.0270           -0.0000j            1.0270
-0.0000
AR.2            0.7834           -1.0303j            1.2943
-0.1465
AR.3            0.7834           +1.0303j            1.2943
0.1465
AR.4           -1.1247           -0.6422j            1.2952
-0.4174
AR.5           -1.1247           +0.6422j            1.2952
0.4174
AR.6           -0.2828           -1.4029j            1.4311
-0.2817
AR.7           -0.2828           +1.4029j            1.4311
0.2817
```

```
------------------------------------------------------------------------
-------

c:\Users\marin\AppData\Local\Programs\Python\Python39\lib\site-
packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No
frequency information was provided, so inferred frequency D will be
used.
  self._init_dates(dates, freq)
c:\Users\marin\AppData\Local\Programs\Python\Python39\lib\site-
packages\statsmodels\tsa\deterministic.py:308: UserWarning: Only
PeriodIndexes, DatetimeIndexes with a frequency set, RangesIndexes,
and Index with a unit increment support extending. The index is set
will contain the position relative to the data length.
  fcast_index = self._extend_index(index, steps, forecast_index)

2013-12-02    345269.605384
Freq: D, dtype: float64

pred = ar_model.predict(start=len(train_df), end=(len(df)-1),
dynamic=False)
print(pred)

2013-12-02    345269.605384
Freq: D, dtype: float64

c:\Users\marin\AppData\Local\Programs\Python\Python39\lib\site-
packages\statsmodels\tsa\deterministic.py:308: UserWarning: Only
PeriodIndexes, DatetimeIndexes with a frequency set, RangesIndexes,
and Index with a unit increment support extending. The index is set
will contain the position relative to the data length.
  fcast_index = self._extend_index(index, steps, forecast_index)
```