# Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»	
Кафедра ИУ5 «Системы обработки информации и управления	<b>&gt;&gt;</b>

Отчет по лабораторной работе №6 «Анализ и прогнозирование временного ряда»

Выполнил:

студент группы ИУ5-63Б Кащеев Максим Проверил:

преподаватель каф. ИУ5 Гапанюк Ю.Е.

Москва, 2022 г.

# Описание задания:

- 1. Выберите набор данных (датасет) для решения задачи прогнозирования временного ряда.
- 2. Визуализируйте временной ряд и его основные характеристики.
- 3. Разделите временной ряд на обучающую и тестовую выборку.
- 4. Произведите прогнозирование временного ряда с использованием как минимум двух методов.
- 5. Визуализируйте тестовую выборку и каждый из прогнозов.
- 6. Оцените качество прогноза в каждом случае с помощью метрик.

# Лабораторная работа №6: "Анализ и прогнозирование временного ряда".

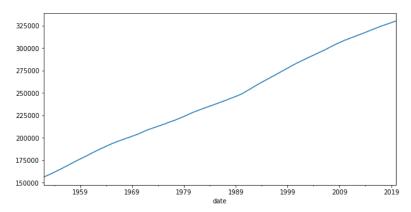
```
In [2]:
         import numpy as np
          import pandas as pd
          from matplotlib import pyplot
          import matplotlib.pyplot as plt
          from statsmodels.tsa.seasonal import seasonal_decompose
         from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from statsmodels.tsa.arima.model import ARIMA
          from sklearn.model_selection import GridSearchCV
          from gplearn.genetic import SymbolicRegressor
          from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
        Использован датасет, содержащий данные об изменении численности населения: https://www.kaggle.com/datasets/census/population-time-series-data?
        datasetId=51748&sortBy=voteCount
In [4]:
         data = pd.read_csv('POP.csv')
         data.head()
Out[4]: realtime_start value
                                   date realtime_end
             2019-12-06 156309.0 1952-01-01
                                             2019-12-06
         1 2019-12-06 156527.0 1952-02-01 2019-12-06
         2 2019-12-06 156731.0 1952-03-01 2019-12-06
             2019-12-06 156943.0 1952-04-01
             2019-12-06 157140.0 1952-05-01 2019-12-06
        Проигнорируем данные о реальном времени, поскольку мы концентрируемся только на диапазоне дат, в котором меняется население.
In [5]:
         data = data.drop(['realtime_start','realtime_end'],axis=1)
In [6]:
         """Преобразование столбца даты в объект datetime и установка его в качестве индекса"""
         data['date'] = pd.to_datetime(data['date'])
data.set_index('date',inplace=True)
          data.head()
                       value
Out[6]:
               date
         1952-01-01 156309.0
         1952-02-01 156527.0
         1952-03-01 156731.0
         1952-04-01 156943.0
         1952-05-01 157140.0
In [7]: | data.describe()
Out[7]:
                       value
         count
         mean 243847.767826
           std 50519.140567
           min 156309.000000
          25% 201725.250000
          50% 239557.500000
          75% 289364.250000
          max 330309.946000
        Визуализация временного ряда
In [8]:
```

 $\label{eq:fig_start} \text{fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))}$ 

fig.suptitle('Временной ряд в виде графика') data.plot(ax=ax, legend=False)

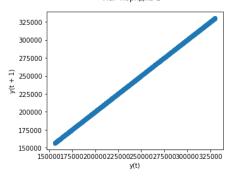
pyplot.show()

#### Временной ряд в виде графика

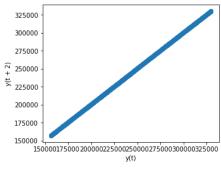


```
In [9]: for i in range(1, 5):
    fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(5,4))
    fig.suptitle(f', lar ποραμκα {i}')
    pd.plotting.lag_plot(data, lag=i, ax=ax)
    pyplot.show()
```

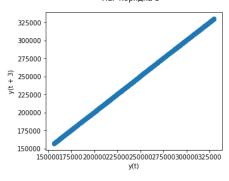
#### Лаг порядка 1



#### Лаг порядка 2



#### Лаг порядка З



# Лаг порядка 4 325000 - 300000 - 275000 - 2500000 - 2500000 - 2500000 - 2500000 - 2500000 - 250000 - 250000 -

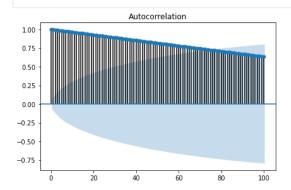
```
In [10]:

fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))
fig.suptitle('Автокорреляционная диаграмма')
pd.plotting.autocorrelation_plot(data, ax=ax)
pyplot.show()
```

#### 

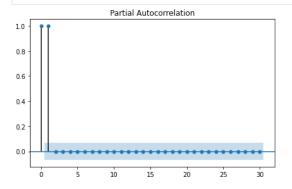
#### Автокорреляционная функция

In [42]: plot\_acf(data, lags=100)
 plt.tight\_layout()



## Частичная автокорреляционная функция

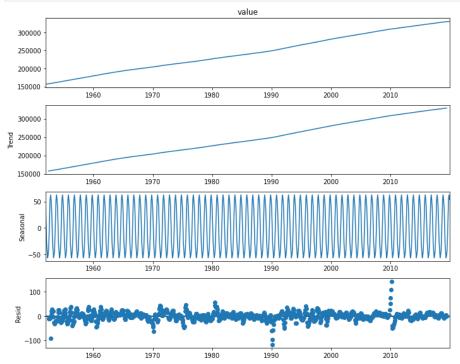
In [40]: plot\_pacf(data, lags=30)
 plt.tight\_layout()



#### Декомпозиция временного ряда

```
In [14]:
    decomposed = seasonal_decompose(data['value'], model = 'add')
    fig = decomposed.plot()
```





Наблюдается положительная динамика с 1952 по 2019 год.

# Разделение временного ряда на обучающую и тестовую выборку

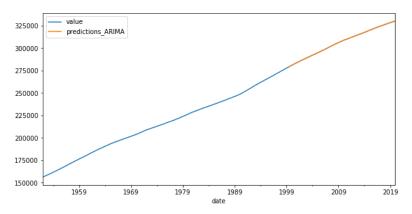
```
In [15]: data_2 = data.copy()

In [16]: # Целочисленная метка шкалы θремени xnum = list(range(data_2.shape[0])) # Pasdeneнue θωθορκи на οδυγακωμγο и тестовую Y = data_2['value'].values train_size = int(len(Y) * 0.7) xnum_train, xnum_test = xnum[0:train_size], xnum[train_size:] train, test = Y[0:train_size], Y[train_size:] history_arima = [x for x in train]
```

## Прогнозирование временного ряда авторегрессионным методом (ARIMA)

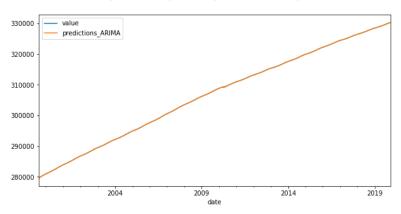
```
In [17]:
            # Параметры модели (p,d,q) arima_order = (2,1,0)
            # Формирование предсказаний
            predictions arima = list()
            for t in range(len(test)):
                 model_arima = ARIMA(history_arima, order=arima_order)
                model_arima_fit = model_arima.fit()
yhat_arima = model_arima_fit.forecast()[0]
predictions_arima.append(yhat_arima)
                 \verb|history_arima.append(test[t])|\\
            # Вычисление метрики RMSE
            error_arima = mean_squared_error(test, predictions_arima, squared=False)
In [18]:
            # Ошибка прогноза
            np.mean(Y), error_arima
           (243847.7678259804, 24.173499535797916)
Out[18]:
In [19]:
            # Записываем предсказания в DataFrame
            data_2['predictions_ARIMA'] = (train_size * [np.NAN]) + list(predictions_arima)
In [20]:
            \label{eq:fig_power}  \mbox{fig, ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10,5))} 
            fig.suptitle('Предсказания временного ряда')
            data_2.plot(ax=ax, legend=True)
            pyplot.show()
```

#### Предсказания временного ряда



```
In [21]:
    fig, ax = pyplot.subplots(1, 1, sharey='col', sharey='row', figsize=(10,5))
    fig.suptitle('Предсказания временного ряда (тестовая выборка)')
    data_2[train_size:].plot(ax=ax, legend=True)
    pyplot.show()
```

#### Предсказания временного ряда (тестовая выборка)



# Прогнозирование временного ряда методом символьной регресии

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

y = column or 1d(y, warn=True)

y =	column_or	_1d(y, warn=True)				
	Populati	ion Average		Best Individual	I	
Gen	Length	Fitness	Length	Fitness	00B Fitness	Time Left
0	263.65	2.43463e+63	23	7.14077e+09	N/A	2.77m
1	130.98	5.77055e+16	43	6.06688e+09	N/A	1.13m
2	53.10	4.58992e+15	34	3.54847e+09	N/A	39.70s
3	34.28	1.99853e+19	13	1.42699e+09	N/A	32.05s
4	35.05	2.10424e+16	38	1.04052e+09	N/A	31.71s
5	30.47	2.56729e+16	36	4.29436e+08	N/A	29.49s
6	31.30	3.00498e+16	50	6.39791e+07	N/A	30.52s
7	38.37	8.59782e+15	35	1.51165e+07	N/A	30.47s
8	43.37	5.29474e+15	47	4.76034e+06	N/A	30.80s
9	37.70	8.42452e+15	35	4.14545e+06	N/A	27.96s
10	40.68	5.69103e+15	32	3.65059e+06	N/A	30.63s
11	45.38	5.71108e+15	29	3.65015e+06	N/A	29.93s
12	41.36	5.72894e+15	29	3.65015e+06	N/A	29.92s
13	35.07	3.58233e+15	29	3.65015e+06	N/A	25.72s
14	33.33	8.46569e+15	35	3.53261e+06	N/A	25.03s
15	31.43	3.14997e+19	35	3.53261e+06	N/A	24.63s
16	30.19	1.42657e+16	35	3.53261e+06	N/A	22.80s
17	30.81	2.81228e+15	35	3.53261e+06	N/A	24.42s
18	33.31	5.72757e+15	35	3.53261e+06	N/A	23.06s
19	33.71	1.26632e+16	35	3.50395e+06	N/A	22.17s
20	34.95	1.70198e+16	35	3.50395e+06	N/A	22.53s
21	42.21	6.70957e+15	35	3.50395e+06	N/A	24.04s
22	54.68	6.78469e+15	35	3.50395e+06	N/A	24.09s
23	50.99	6.47928e+18	102	3.50387e+06	N/A	23.79s
24	42.69	8.57551e+15	71	3.50376e+06	N/A	20.67s
25	59.07	6.73374e+21	85	3.49756e+06	N/A	23.04s
26	89.07	1.51918e+25	85	3.49756e+06	N/A	26.59s
27	100.70	2.98833e+18	91	3.48956e+06	N/A	28.13s
28	120.58	7.92131e+23	91	3.48956e+06	N/A	31.64s
29	142.26	1.91023e+18	127	3.48498e+06	N/A	33.27s
30	116.37	6.9315e+21	54	3.46676e+06	N/A	35.69s

```
103.96
                    2.33782e+22
                                                3.46676e+06
  31
                                       54
                                                                           N/A
                                                                                   34.645
                     2.82439e+18
                                        54
        107.16
                                                3.46676e+06
                                                                           N/A
  33
       110.56
                    4.95099e+26
                                       112
                                                3.45858e+06
                                                                           N/A
                                                                                   25.78s
  34
        94.20
                    1.96986e+18
                                       114
                                                3.45249e+06
                                                                           N/A
                                                                                   22.815
  35
         77.71
                      6.0703e+15
                                       133
                                                3.43034e+06
                                                                           N/A
                                                                                   19.76s
       111.25
                     5.62717e+15
                                       79
                                                3.42948e+06
                                                                           N/A
                                                                                   23.33s
  37
38
        142.44
                      1.4552e+18
                                       246
                                                3.41658e+06
                                                                           N/A
                                                                                   24.405
                                       187
                                                                                   26.25s
       171.28
                    3.11029e+19
                                                3.36822e+06
                                                                           N/A
  39
                     2.8446e+16
                                       187
                                                                           N/A
        213.08
                     1.12226e+16
  40
                                       212
                                                3.35931e+06
                                                                           N/A
                                                                                   28.03s
  41
       193.33
                     7.07447e+17
                                       181
                                                3.35563e+06
                                                                           N/A
                                                                                   25.74s
                    9.48793e+19
                                                3.25166e+06
  42
        200.58
                                       308
                                                                           N/A
                                                                                   25.51s
                      6.9535e+17
                                       308
                                                3.24914e+06
                                                                           N/A
  44
        271.65
                    2.48275e+15
                                       434
                                                3.17665e+06
                                                                                   28.32s
                                                                           N/A
  45
        340.95
                    1.45248e+18
                                       434
                                                3.17665e+06
                                                                           N/A
                                                                                   31.87s
        407.23
                      2.9286e+14
                                      874
                                                3.13466e+06
                                                                                    34.04s
  46
                                                                           N/A
                                                3.13086e+06
        475.59
                     8.20919e+13
                                       857
                                                                                    36.51s
  48
       698.39
                     6.58531e+17
                                     1124
                                                 3.1245e+06
                                                                           N/A
                                                                                   47.91s
  49
       871.75
                     5.67064e+14
                                     1140
                                                 3.1232e+06
                                                                           N/A
                                                                                   54.96s
                     1.44739e+18
                                      1126
                                                3.11533e+06
                                                                           N/A
       1040.20
                     8.00984e+13
                                                                                    57.75s
  51
                                     1337
                                                 3.1087e+06
                                                                           N/A
  52
       1087.90
                     4.8939e+10
                                     1352
                                                3.10262e+06
                                                                           N/A
                                                                                   56.91s
  53
       1212.74
                    6.88053e+18
                                     1338
                                                3.09244e+06
                                                                           N/A
                                                                                   58.22s
       1332.59
                     9.76027e+14
                                                3.09015e+06
                                                                           N/A
  55
       1375.70
                     4.2908e+14
                                                3.08045e+06
                                                                                    57.03s
                                     1361
                                                                           N/A
  56
       1400.24
                     4.21109e+14
                                     1622
                                                3.07579e+06
                                                                           N/A
                                                                                   53.435
  57
       1485.49
                     3.56926e+14
                                     1361
                                                 3.0712e+06
                                                                           N/A
                                                                                    56.32s
                     5.99454e+17
                                     1379
                                                3.06568e+06
                                                                                    58.10s
  59
       1519.96
                     1.18494e+10
                                     1452
                                                3.05779e+06
                                                                           N/A
                                                                                    44.295
  60
       1441.96
                     3.61367e+14
                                     1452
                                                3.05779e+06
                                                                           N/A
                                                                                    37.905
  61
       1484.91
                    3.60553e+14
                                     1441
                                                3.04915e+06
                                                                           N/A
                                                                                   36.40s
       1502.33
                     8.71965e+13
                                      1441
                                                3.04637e+06
                                                                           N/A
  63
      1499.87
                     5.11657e+12
                                     1470
                                                3.04262e+06
                                                                           N/A
                                                                                   27.785
                                                3.03789e+06
  64
      1457.59
                    3.97956e+14
                                     1453
                                                                                   21.46s
                                                                           N/A
  65
      1514.59
                     3.44098e+14
                                     1735
                                                3.03427e+06
                                                                                    20.76s
                                                                           N/A
       1597.79
                     6.70466e+14
                                                3.02874e+06
                                     1728
                                                                           N/A
                                                                                    14.12s
  67
      1652.89
                     3.52673e+14
                                     1753
                                                3.02842e+06
                                                                           N/A
                                                                                    9.83s
                                                3.02504e+06
  68
      1696.85
                     3.44312e+14
                                     1728
                                                                           N/A
                                                                                    4.99s
       1756.59
                     5.96419e+17
                                     1817
                                                3.01702e+06
```

Out [23]. Symbol

function\_set=['add', 'sub', 'mul', 'div', 'sin'],
generations=70, init\_depth=(4, 10), metric='mse',
population\_size=500, random\_state=0, stopping\_criteria=0.01,
verbose=1)

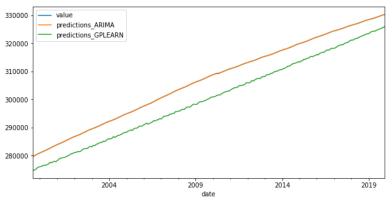
In [24]:

print(SR.\_program)

dd(add(add(add(X0, 51.302), add(add(X0, X0), X0)), add(X0, X0)), add(X0, X0))))), sin(mul(sub(sub(sub(sub(add(X0, X0), Sin(mul(sub(-36.019, -77.64 tal(add(add(add(x0, 51.302), add(add(x0, x0), x0)), add(x0, x0)), add(x0, x0)), sin(mul(sub(-36.019, -77.644), add(add(add(x0, 51.302), add(add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0)), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0)), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0), sin(x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0), sin(x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0), sin(x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0)))), sin(mul(sub(-36.019, -77.644), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0)))), sin(mul(sub(-36.019, -77.644), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(x0, x0), x0)))), sin(mul(sub(-36.019, -77.644), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0)))), sin(mul(sub(-36.019, -77.644), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0), x0))))), sin(mul(sub(-36.019, x0)))), sin(mul(sub(-36.019, x0))))), sin(mul(sub(-36.019, x0)))), sin(mul(sub(-36.019, x0)))), si 0))))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, 51.302), add(add(X0, X0), X0)), add(X0, X0)), add(X0, X0))))), sin(mul(sub(-36.019, 67)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), sub(add(x0, x0)), add(add(add(x0, x0)), sin(x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0)))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0)))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, x0)), add(x0, x0))))))))))))))))))))))))) 1.302), add(add(X0, X0), X0)), add(X0, X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, sub(mul(sub(-36.019, -77.644), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(X0, X0), add(add(add(X0, X0), X0), sin(x0))))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, X0), X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(X0, X0), X0)), add(X0, X0)), x dd(add(add(add(X0, 51.302), add(add(X0, X0), X0)), add(X0, X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, sub(mul(sub(-36.019, -77.644), add(add(add(X0, sub(mul(sub(-36.019, -77.644), add(add(add(X0, sub(mul(sub(-36.019, -77.644), add(add(add(X0, sub(mul(sub(-36.019, -77.644), add(add(x0, sub(-36.019, -77.644), add(add(x0, sub(-36.019, -77.644), add(x0, sub(-36.019, -77.644), add(add(x0, sub(-36.019, -77.644), add(x0, sub( add(add(add(add(add(add(add(x0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(add(x0, 51.302), add(x0, X0)), add(X0, X0)))), add(X0, X0)), add(X0, X0), x0)), add(X0, X0), x0)), add(X0, X0), x0), add(x0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, X0), X0)), add(X0, X0)), sub(add(X0, X0), x0), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(add(x0, X0), X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, X0), X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, X0), X0)), add(x0, X0)))), sin(x0)))), sin(x0) 0)))))), \$1.302), add(X0, X0)))), \$in(mul(sub(-36.019, -77.644), add(add(add(add(add(X0, \$1.302), add(X0, X0)), x0)), add(X0, X0)), add(X0, X0)))), \$in(mul(sub(-36.019, -77.644), add(add(add(x0, \$1.302), add(X0, X0)))), \$in(mul(sub(-36.019, -77.644), add(add(add(X0, \$1.302), add(X0, X0)), add(X0, X0)), add(X0, X0)))), \$in(mul(sub(-36.019, -77.644), add(add(add(X0, \$1.302), add(X0, \$1.302), add( in(mul(sub(-36.019, -77.644), add(add(add(x0, 51.302), add(add(X0, X0), X0)), add(X0, X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), 4), add(add(x0, sub(mul(sub(-36.019, -77.644), add(X0, X0)), sub(add(X0, X0), add(add(add(X0, X0), X0), sin(X0))))), 51.302), add(X0, X 0))))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, 51.302), add(add(X0, X0), X0)), add(X0, X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(X0, 51.302), add(add(X0, X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(X0, X0)), add(X0, X0))), add(X0, X0))), sin(mul(sub(-36.019, -77.644), add(add(x0, 51.302), add(add(X0, X0)), x0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(x0, 51.302), add(x0, X0)), x0)), add(x0, X0)), x0)), add(x0, X0)), add(x0, X0)), x0)), add(x0, X0)), add(x0, X0)), add(x0, X0)), x0)), add(x0, X0)), add(x0, X0)), x0))), x0)), x0)), x0))), x0)), x0)), x0)) 0)), add(X0, X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, sub(mul(sub(-36.019, -77.644), add(X0, X0))), sub(add(X0, X dd(add(add(X0, X0), X0), X0), X0), X0), Sin(X0))))), 51.302), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, 51.302), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(sin(X0), X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(x0, X0), X0))), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, X0), X0))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, X0), X0))), sin(mul(sub(-36.019, -77.644), add(add(add(X0, X0), X0))), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(add(X0, X0), X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(add(X0, X0), X0)), add(X0, X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(add(X0, X0), X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(X0, X0), X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(x0, X0), X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(add(X0, S1.302), add(add(X0, X0), X0)), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(add(x0, S1.302), add(add(X0, X0), X0))), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(x0, S1.302), add(add(x0, X0), X0))), add(X0, X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(add(x0, S1.302), add(add(x0, X0), X0)))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, S1.302), add(add(x0, X0), X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, S1.302), add(add(x0, X0), X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, S1.302), add(add(x0, X0), X0)))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, S1.302), add(add(x0, X0), X0))))), sin(mul(sub(-36.019, -77.644), add(add(add(x0, S1.302), add(add(x0, S1.302), add(add(x0, S1.302), add(add(x0, S1.302), add(x0, X0)))))), sin(mul(sub(-36.019, -77.644), add(add(x0, S1.302), add(add(x0, S1.302), add(x0, X0)))))), sin(mul(sub(-36.019, -77.644), add(x0, X0))))), sin(mul(sub(-36.019, -77.644), add(x0, X0))))), sin(mul(sub(-36.019, -77.644), add(x0, X0))))), sin(mul(sub(-36.019, -77.644), add(x0, X0))))), sin(mul(sub(-36.019, -77 

```
In [25]:
          # Предсказания
          y_sr = SR.predict(np.array(xnum_test).reshape(-1, 1))
          y_sr[:10]
         array([274891.75793852, 274817.36307349, 275068.42349884, 275594.91553188,
Out[25]:
                275909.57465535, 276033.9204471 , 276192.3291504 , 276368.95663777,
                276651.56236873, 276542.01774132])
In [26]:
          # Записываем предсказания в DataFrame
          data_2['predictions_GPLEARN'] = (train_size * [np.NAN]) + list(y_sr)
In [27]:
          fig. ax = pyplot.subplots(1, 1, sharex='col', sharey='row', figsize=(10.5))
          fig.suptitle('Предсказания временного ряда (тестовая выборка)')
          data_2[train_size:].plot(ax=ax, legend=True)
          pyplot.show()
```

#### Предсказания временного ряда (тестовая выборка)



```
In [28]: error_SR = mean_squared_error(test, y_sr, squared=False)

In [29]: # Οωμόκα προ≥μοσα np.mean(Y), error_SR

Out[29]: (243847.7678259804, 6510.330169456957)
```

#### Качество прогноза моделей

```
In [30]:

def print_metrics(y_test, y_pred):
    print(f"R^2: {r2_score(y_test, y_pred)}")
    print(f"MSE: {mean_squared_error(y_test, y_pred, squared=False)}")
    print(f"MAE: {mean_absolute_error(y_test, y_pred)}")
```

```
In [31]: print("ARIMA")
print_metrics(test, predictions_arima)
print("\nGPLEARN")
print_metrics(test, y_sr)
```

ARIMA R^2: 0.9999973075905872 MSE: 24.173499535797916 MAE: 16.034435631401305

GPLEARN R^2: 0.8047153645391025 MSE: 6510.330169456957

має: 6443.710113418146

Вывод: Обе модели, ARIMA и GPLEARN, показали хороший результат. Лучшей по всем используемым метрикам оказалась модель ARIMA.