

In [12]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf
from scipy.interpolate import CubicSpline
from scipy.stats import t
from IPython.display import Math
from sympy import *
alpha = 0.95
pd.set_option('future.no_silent_downcasting', True)
```

ФУНКЦИИ СОЗДАНИЯ МОДЕЛЕЙ

In [13]:

```
def addition_model_auto_reg(origin_data, n_new, text_show = False, show_tables =
    def display_txt(lst):
        if type(lst)==list:

            for text in lst:
                try:
                    display(Math(text.replace(' ', '~')))
                except:
                    display(Math(str(text)))
        else:
            try:
                display(Math(lst.replace(' ', '~')))
            except:
                display(Math(str(lst)))

def smooth_plot(datas = [], indices = [], xfroms = [], markers = [], last = True):

    if xfroms == []:
        xfroms = [1]*len(indices)

    lines = []

    for i in range(len(indices)):
        # Данные
        if dropna:
            y = datas[i][indices[i]].dropna().values
        else:
            y = datas[i][indices[i]].values
        x = np.arange(xfroms[i], y.size+xfroms[i], dtype = float)
        # Создание слаживающей функции
        cs = CubicSpline(x, y)

        # Генерация новых точек для плавного графика
        x_smooth = np.linspace(x.min(), x.max(), 100)
        y_smooth = cs(x_smooth)

        if len(markers):
            plt.plot(x, y, markers[i] + 'o') # Оригинальные точки
            lines.append(plt.plot(x_smooth, y_smooth, markers[i] + '-', alpha
                if last:
                    plt.legend()
```

```

plt.grid()
plt.show()

def bar_plot_dict(r:dict):
    # Данные для коррелограммы
    lags = r.keys()
    correlations = r.values()

    # Создание графика
    fig, ax = plt.subplots()
    bars = ax.bar(lags, correlations, color='lightblue')

    # Добавление значений на вершины столбцов
    for bar in bars:
        yval = bar.get_height()
        ax.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 8), ha='center')

    # Настстройка осей и заголовков
    ax.set_xlabel('Лаг')
    ax.set_ylabel('Корреляция')
    ax.set_ylim(0, 0.6)

    # Отображение графика
    plt.show()

y = origin_data.values.ravel()

n_new = origin_data.shape[1] * n_new
display_txt(n_new)
data = pd.DataFrame(origin_data.values.ravel()).reshape(-1,1), index=pd.RangeIndex(start=0, stop=n_new, step=1)

if show_tables:
    display(data)

data['(y_t - _y)**2'] = (data['y_t'] - data['y_t'].mean())**2
r = dict()
for i in range(1,n_new+1):
    data[f'y_t-{i}'] = data['y_t'].shift(i)
    data[f'{i}'] = (data['y_t'] - data['y_t'].mean())*(data[f'y_t-{i}'] - data['y_t'])
    r[f'{i}'] = data[f'{i}'].sum()/data['(y_t - _y)**2'].sum()

if text_show:
    display_txt(
        '''Для выявления структуры имеющегося временного ряда проведем автокорреляцию. Пользуясь данными итоговой строки таблицы, рассчитаем коэффициенты автокорреляции 1-го, 2-го, 3-го и 4-го порядков.'''
    )
if show_tables:
    display(data)

bar_plot_dict(r)

txt1_1 = f'''Из графика коррелограммы и значений коэффициентов автокорреляции видно, что наиболее тесная связь наблюдается при временном лаге {n_new} при умеренном значении коэффициента автокорреляции 1-го порядка. Из этого следует, что во временном ряду наряду с

```

Т присутствуют сезонные колебания S с периодичностью в $\{n_{new}\}$ квартала, то динамики инвестиций в основной капитал ежегодно повторяется.

Данные предположения также подтверждаются графиком динамики наблюдаемых значений исследуемого показателя ''''

```

if text_show:
    display_txt(txt1_1.split('\n'))
```

smooth_plot([data],['y_t'],[1],['b'])

txt1 = '''Из графика поквартальной динамики инвестиций в основной капитал видно что амплитуда колебаний постепенно увеличивается, следовательно, для моделирования такого временного ряда, целесообразнее использовать мультипериодную модель. Впоследствии данный факт будет доказан сравнением показателей качества аддитивной и мультипликативной моделей проекции. Как было сказано выше, построение аддитивной и мультипликативной моделей проекции для начала построим аддитивную модель. '''

```

if text_show:
    display_txt(txt1.split('\n'))
```

if text_show:
 display_txt('Произведем выравнивание исходного временного ряда с помощью')
data2 = pd.DataFrame(y.reshape(-1,1), columns=['y_t'])

k4 = []
yc_t = []
for i in range(y.size - (n_new-1)):
 k4.append(sum([y[i+j] for j in range(n_new)]))
 yc_t.append(sum([y[i+j] for j in range(n_new)])/n_new)

data2['4k'] = [pd.NA]*(n_new//2) + k4 + [pd.NA]*(n_new - n_new//2 - 1)
data2['yc_t'] = [pd.NA]*(n_new//2) + yc_t + [pd.NA]*(n_new - n_new//2 - 1)
data2['_ycc_t'] = [pd.NA]*(n_new//2) + [(data2['yc_t'][i]+data2['yc_t'][i+1])/2]
data2['mark'] = (data2['y_t'] - data2['_ycc_t'])

if show_tables:
 display(data2)
if text_show:
 display_txt('Произведя подобные преобразования, мы сгладили имеющиеся в')
smooth_plot([data,data2],['y_t','yc_t'], [1,3],['b','m'])

txt2='''Шаг 2. Найдем оценки сезонной компоненты как разность между фактическими и центрированными скользящими средними с циклом t у .

Далее перенесем полученные разности в таблицу согласно соответствующему году

```

if text_show:
    display_txt(txt2.split('\n'))
marks = data2['mark'].fillna(0).values.reshape(*origin_data.shape).astype(float)
if show_tables:
    display(marks)
```

_S_i = marks.sum(axis=0)/np.count_nonzero(marks,0)

_S_i
txt2_1_5 = 'Затем рассчитаем средние оценки сезонных колебаний S по всем годам'
display_txt([txt2_1_5]+list(_S_i))

_S = (_S_i).sum()

```

txt2_l_4 = """При моделировании периодических колебаний обычно предполагается что сезонные воздействия за период взаимопогашаются. В аддитивной модели это что сумма значений сезонной компоненты по всем кварталам должна быть равна н
display_txt(txt2_l_4.split('\n')+[S])

k = S/origin_data.shape[0]

txt2_l_3 = '''Для устранения данного противоречия скорректируем полученные средние значения оценок сезонной компоненты S таким образом. Величина корректирующего коэффициента будет следующей: '''
display_txt(txt2_l_3.split('\n')[k])

d_S_i = S_i - k
d_S_i

txt2_l_2 = 'Теперь рассчитаем скорректированные значения сезонной компоненты'
display_txt(txt2_l_2.split('&&&') + list(d_S_i))

txt2_l_1 = 'Проверим условие взаимопогашаемости значений сезонной компоненты'
display_txt([txt2_l_1,f'{d_S_i.sum()} +r' '\approx 0'])

txt2_l = 'Перенесем полученные значения S в таблицу напротив соответствующих'
display_txt(txt2_l)

data3 = pd.DataFrame(data.loc[:, 'y_t'])
data3['S'] = list(d_S_i)*origin_data.shape[0]
if show_tables:
    display(data3)

txt3='''Шаг 3. Исключим влияние сезонной компоненты S, вычитая ее соответственно. Таким образом получим: T + E = Yt - S.
Эти значения рассчитываются для каждого момента времени и содержат только те ...
if text_show:
    display_txt(txt3.split('\n'))

data3['T+E'] = data3['y_t'] - data3['S']
if show_tables:
    display(data3)

b0, b1 = symbols('b0 b1')

# Запись уравнений системы
eq1 = Eq(data3.index.size * b0 + pd.Series(data3.index).sum() * b1, data3['T+E'].sum())
eq2 = Eq(pd.Series(data3.index).sum() * b0 + (pd.Series(data3.index)**2).sum(), b1**2)

# Решение системы уравнений
solution = solve((eq1, eq2), (b0, b1))

# Вывод решения
print("b0 =", solution[b0])
print("b1 =", solution[b1])
b0 = float(solution[b0])
b1 = float(solution[b1])

```

```

data3['T'] = pd.Series(b0 + np.arange(0,data.shape[0] +1)*b1)
data3['T+S'] = data3['T']+data3['S']

if show_tables:
    display(data3)

smooth_plot([data3]*4,['y_t','T+E','T','T+S'],markers=['b','m','y','c'])

data['e_t'] = data['y_t'] - data3['T+S']
data4 = pd.DataFrame(data.loc[:, 'e_t'])
data4['(dyt - y_t)**2'] = (data3['T+S'] - data['y_t'].mean())**2
data4['(y_t - dyt)**2'] = (data3['T+S'] - data['y_t'])**2
data4['|y_t - dyt|/y_t'] = np.abs(data3['T+S'] - data['y_t'])/data['y_t'] *

if show_tables:
    display(data4)

if text_show:
    display_txt('В оцененном виде аддитивную модель изучаемого временного ря')

display_txt(f'y = {b0} + {b1}' + r'\times t + \hat{S_i} + e_t')

display_txt(r'\hat{S_i} = ' + f'{list(d_S_i)}')

def hat_y(t):
    return b0 + b1*t + d_S_i[t%len(d_S_i)-1]

h_y = np.vectorize(hat_y)

predicted = h_y(range(data.shape[0] +1 ,data.shape[0] +1 + n_new))

display_txt(['\nПредсказанные значения:\n', *predicted])

S_e = np.sqrt(data4['(y_t - dyt)**2'].sum()/(data.shape[0] - 1 - 1))

display_txt(f'\nРассчитаем стандартную ошибку регрессии: {S_e}\n')

def s_e(t):
    return S_e * np.sqrt(1 + 1/data.shape[0] + (t - np.mean(data.index))**2)

std = np.vectorize(s_e)

errors = std(range(data.shape[0] +1 ,data.shape[0] +1 + n_new))

display_txt(['\nСтандартные ошибки прогноза будут следующими:\n', *errors])

# Нахождение критического значения t
degrees_of_freedom = data.shape[0] - 2
t_critical = t.ppf(1 - (1 - alpha) / 2, degrees_of_freedom)

lower_boarder = (h_y(range(data.shape[0] +1 ,data.shape[0] +1 + n_new)) - st
upper_boarder = (h_y(range(data.shape[0] +1 ,data.shape[0] +1 + n_new)) + st

```

```

display_txt(f"\nКритическое значение t_{{{0.95;{degrees_of_freedom}}}} = {t_cr}

data5 = pd.DataFrame(np.hstack((data3.iloc[:,4].values,predicted)),columns = [
    'y_t','predict','lower_boarder','upper_boarder'])

data6 = pd.DataFrame(np.vstack((lower_boarder, upper_boarder)).T, columns=['y_t'])
if show_tables:
    display(data6)

smooth_plot([data3,data5,data6,data6],[ 'y_t','predict','lower_boarder','upper_boarder'])

R2 = data4['(dyt - _y_t)**2'].sum()/(data4['(dyt - _y_t)**2'].sum() + data4['y_t'].var())
display_txt(['- коэффициент детерминации',R2])

A_ = data4['|y_t - dyt|/y_t'].mean()
display_txt(['- средняя ошибка аппроксимации', f'{A_}%'])

```

```

In [14]: def multiplication_model_auto_reg(origin_data, n_new, text_show = False, show_table=False):
    def display_txt(lst):
        if type(lst)==list:
            for text in lst:
                try:
                    display(Math(text.replace(' ', '~')))
                except:
                    display(Math(str(text)))
        else:
            try:
                display(Math(lst.replace(' ', '~')))
            except:
                display(Math(str(lst)))

    def smooth_plot(datas = [], indices = [], xfroms = [], markers = [],last = True):
        if xfroms == []:
            xfroms = [1]*len(indices)

        lines = []

        for i in range(len(indices)):
            # Данные
            if dropna:
                y = datas[i][indices[i]].dropna().values
            else:
                y = datas[i][indices[i]].values
            x = np.arange(xfroms[i], y.size+xfroms[i], dtype = float)
            # Создание сглаживающей функции
            cs = CubicSpline(x, y)

            # Генерация новых точек для плавного графика
            x_smooth = np.linspace(x.min(), x.max(), 100)
            y_smooth = cs(x_smooth)

            if len(markers):
                plt.plot(x, y, markers[i] + 'o') # Оригинальные точки
                lines.append(plt.plot(x_smooth, y_smooth, markers[i] + '-', alpha=0.5))
            else:
                plt.plot(x, y, 'o')
                plt.plot(x_smooth, y_smooth, '-')

    return lines

```

```

if last:
    plt.legend()
    plt.grid()
    plt.show()

def bar_plot_dict(r:dict):
    # Данные для коррелограммы
    lags = r.keys()
    correlations = r.values()

    # Создание графика
    fig, ax = plt.subplots()
    bars = ax.bar(lags, correlations, color='lightblue')

    # Добавление значений на вершины столбцов
    for bar in bars:
        yval = bar.get_height()
        ax.text(bar.get_x() + bar.get_width()/2, yval, round(yval, 8), ha='center')

    # Настстройка осей и заголовков
    ax.set_xlabel('Лаг')
    ax.set_ylabel('Корреляция')
    ax.set_ylim(0, 0.6)

    # Отображение графика
    plt.show()

y = origin_data.values.ravel()

n_new = origin_data.shape[1] * n_new
display_txt(n_new)
data = pd.DataFrame(origin_data.values.ravel()).reshape(-1,1), index=pd.RangeIndex(0, n_new)

if show_tables:
    display(data)

data['(y_t - _y)**2'] = (data['y_t'] - data['y_t'].mean())**2
r = dict()
for i in range(1,n_new+1):
    data[f'y_t-{i}'] = data['y_t'].shift(i)
    data[f'{i}'] = (data['y_t'] - data['y_t'].mean())*(data[f'y_t-{i}'] - data[f'{i}'])
    r[f'{i}'] = data[f'{i}'].sum()/data['(y_t - _y)**2'].sum()

if text_show:
    display_txt(
        '''Для выявления структуры имеющегося временного ряда проведем автокорреляцию. Пользуясь данными итоговой строки таблицы, рассчитаем коэффициенты автокорреляции 1-го, 2-го, 3-го и 4-го порядков.'''
    )
if show_tables:
    display(data)

bar_plot_dict(r)

```

```

txt1_1 = f'''Из графика коррелограммы и значений коэффициентов автокорреляции что наиболее тесная связь наблюдается при временном лаге {n_new} при умеренном автокорреляции 1-го порядка. Из этого следует, что во временном ряду наряду с Т присутствуют сезонные колебания S с периодичностью в {n_new} квартала, то динамики инвестиций в основной капитал ежегодно повторяется.

Данные предположения также подтверждаются графиком динамики наблюдаемых значений исследуемого показателя'''

if text_show:
    display_txt(txt1_1.split('\n'))


smooth_plot([data],[ 'y_t'],[1],[ 'b'])

txt1 = '''Из графика поквартальной динамики инвестиций в основной капитал видно что амплитуда колебаний постепенно увеличивается, следовательно, для моделирования такого временного ряда, целесообразнее использовать мультиплексную модель. Впоследствии данный факт будет доказан сравнением показателей качества аддитивной и мультипликативной моделей.

Как было сказано выше, построение аддитивной и мультипликативной моделей проще. Для начала построим аддитивную модель.'''
if text_show:
    display_txt(txt1.split('\n'))


if text_show:
    display_txt('Произведем выравнивание исходного временного ряда с помощью')
data2 = pd.DataFrame(y.reshape(-1,1), columns=[ 'y_t'])

k4 = []
yc_t = []
for i in range(y.size - (n_new-1)):
    k4.append(sum([y[i+j] for j in range(n_new)]))
    yc_t.append(sum([y[i+j] for j in range(n_new)]) / n_new)

data2['4k'] = [pd.NA] * (n_new // 2) + k4 + [pd.NA] * (n_new - n_new // 2 - 1)
data2['yc_t'] = [pd.NA] * (n_new // 2) + yc_t + [pd.NA] * (n_new - n_new // 2 - 1)
data2['_ycc_t'] = [pd.NA] * (n_new // 2) + [(data2['yc_t'][i] + data2['yc_t'][i+1]) / 2]
data2['mark'] = (data2['y_t'] / data2['_ycc_t'])

if show_tables:
    display(data2)
if text_show:
    display_txt('Произведя подобные преобразования, мы сгладили имеющиеся в')
    smooth_plot([data,data2],[ 'y_t','yc_t'], [1,3],[ 'b','m'])

txt2 = '''Шаг 2. Найдем оценки сезонной компоненты как частное от деления фактического ряда t у на центрированные скользящие средние с центром t у.

Далее рассчитаем средние значения сезонных колебаний для каждого квартала i. Для этого перенесем имеющиеся оценки в таблицу'''

if text_show:
    display_txt(txt2.split('\n'))
marks = data2['mark'].fillna(0).values.reshape(*origin_data.shape).astype(float)
if show_tables:
    display(marks)

_S_i = marks.sum(axis=0) / np.count_nonzero(marks, 0)

```

```

txt2_l_5 = '''Взаимопогашаемость сезонных воздействий в мультиплексивной мо
что сумма значений i S сезонной компоненты S должна быть равна числу периода
то есть для данного примера - четырем. Суммируя i S , получаем: '''

_S = (_S_i).sum()

display_txt(txt2_l_5.split('\n')+list(_S_i) +['\n',_S])

k = origin_data.shape[0]/_S

txt2_l_3 = '''Рассчитаем корректирующий коэффициент:
'''

display_txt(txt2_l_3.split('\n')+ [k])

d_S_i = _S_i*k

display_txt(
    '''Определим скорректированные значения сезонной компоненты,
    умножив ее средние оценки i S на корректирующий коэффициент k: '''.split
)

display_txt('''Проверим выполнение свойства взаимопогашаемости сезонных коле
display_txt('''Для выполнения последующих расчетов перенесем полученные знач

data3 = pd.DataFrame(data.loc[:, 'y_t'])
data3['S'] = list(d_S_i)*origin_data.shape[0]
if show_tables:
    display(data3)

txt3='''Шаг 3. Исключим из исходного временного ряда сезонные колебания,
разделив каждый уровень t у на соответствующее значение i S.
В итоге получим временной ряд, содержащий только тенденцию T и случайную ком
if text_show:
    display_txt(txt3.split('\n'))

data3['TE'] = data3['y_t']/data3['S']
if show_tables:
    display(data3)

b0, b1 = symbols('b0 b1')

# Запись уравнений системы
eq1 = Eq(data3.index.size * b0 + pd.Series(data3.index).sum() * b1, data3['T']
eq2 = Eq(pd.Series(data3.index).sum() * b0 + (pd.Series(data3.index)**2).sum()

# Решение системы уравнений
solution = solve((eq1, eq2), (b0, b1))

# Вывод решения
print("b0 =", solution[b0])
print("b1 =", solution[b1])
b0 = float(solution[b0])
b1 = float(solution[b1])

data3['T'] = pd.Series(b0 + np.arange(0,data.shape[0]+1)*b1)
data3['TS'] = data3['T']*data3['S']
smooth_plot([data3]*4,['y_t','TE','T','TS'],markers=['b','m','y','c'])

data['e_t'] = data['y_t']-data3['TS']

```

```

if show_tables:
    display(data3)
if text_show:
    display_txt('Таким образом, мультипликативную модель изучаемого временно')

display_txt(f'y = ({b0} + {b1}' + r'\times t)\times \hat{S_i} \times e_t')

data4 = pd.DataFrame(data.loc[:, 'e_t'])
data4['(dyt - y_t)**2'] = (data3['TS'] - data['y_t'].mean())**2
data4['(y_t - dyt)**2'] = (data3['TS'] - data['y_t'])**2
data4['|y_t - dyt|/y_t'] = np.abs(data3['TS'] - data['y_t'])/data['y_t'] * 1

display_txt(
    '''Для расчета показателей качества модели значения отн т е
    относительной ошибки Е не могут быть использованы.
    Поэтому для оценки качества модели необходимо рассчитать значения
    абсолютной ошибки по формуле Еабс=YT - (T·S), то есть )
    Значения остатков t e представлены в графике 1.
    '''.split('\n')
)
if show_tables:
    display(data4)

display_txt(r'\hat{S_i} = ' + f'{list(d_S_i)}')

def hat_y(t):
    return (b0 + b1*t)* d_S_i[t%len(d_S_i)-1]

h_y = np.vectorize(hat_y)

predicted = h_y(range(data.shape[0] +1 ,data.shape[0] +1 + n_new))

display_txt(['\nПредсказанные значения:\n', *predicted])

S_e = np.sqrt(data4['(y_t - dyt)**2'].sum()/(data.shape[0] - 1 - 1))

display_txt(f'\nРассчитаем стандартную ошибку регрессии: {S_e}\n')

def s_e(t):
    return S_e * np.sqrt(1 + 1/data.shape[0] + (t - np.mean(data.index))**2)

std = np.vectorize(s_e)

errors = std(range(data.shape[0] +1 ,data.shape[0] +1 + n_new))

display_txt(['\nСтандартные ошибки прогноза будут следующими:\n', *errors])

# Нахождение критического значения t
degrees_of_freedom = data.shape[0] - 2
t_critical = t.ppf(1 - (1 - alpha) / 2, degrees_of_freedom)

lower_boarder = (h_y(range(data.shape[0] +1 ,data.shape[0] +1 + n_new)) - st
upper_boarder = (h_y(range(data.shape[0] +1 ,data.shape[0] +1 + n_new)) + st

display_txt(f"\nКритическое значение t_{degrees_of_freedom} = {t_cr}

```

```

data5 = pd.DataFrame(np.hstack((data3.iloc[:,4].values,predicted)),columns = []

data6 = pd.DataFrame(np.vstack((lower_boarder, upper_boarder)).T, columns=[

if show_tables:
    display(data6)

smooth_plot([data3,data5,data6],['y_t','predict','lower_boarder','upper_boarder'])

R2 = data4['(dyt - _y_t)**2'].sum()/(data4['(dyt - _y_t)**2'].sum() + data4['(y_t - dyt)**2'].sum())
display_txt(['- коэффициент детерминации',R2])

A_ = data4['|y_t - dyt|/y_t'].mean()
display_txt(['- средняя ошибка аппроксимации', f'{A_}%'])

```

Доход

In [15]:

```

revenue = '''5,7 6,8 8,7
5,8 6,7 8,7
5,9 6,8 8,9
6 7 9
6 7,1 9,1
6,1 7,3 9,3
6,2 7,5 9,4
6,3 7,6 9,6
6,4 7,8 9,7
6,5 8 9,9
6,6 8,2 10,1
6,7 8,4 10,1'''
```

```

revenue = np.array(list(map(lambda x: list(map(float,(x.replace(',','.')).split())))
revenue = pd.DataFrame(revenue, columns=[2005,2006,2007],index=list(range(1,13)))
revenue = revenue.T
revenue
```

Out[15]:

	1	2	3	4	5	6	7	8	9	10	11	12
2005	5.7	5.8	5.9	6.0	6.0	6.1	6.2	6.3	6.4	6.5	6.6	6.7
2006	6.8	6.7	6.8	7.0	7.1	7.3	7.5	7.6	7.8	8.0	8.2	8.4
2007	8.7	8.7	8.9	9.0	9.1	9.3	9.4	9.6	9.7	9.9	10.1	10.1

In []: #revenue.to_excel('revenue.xlsx')

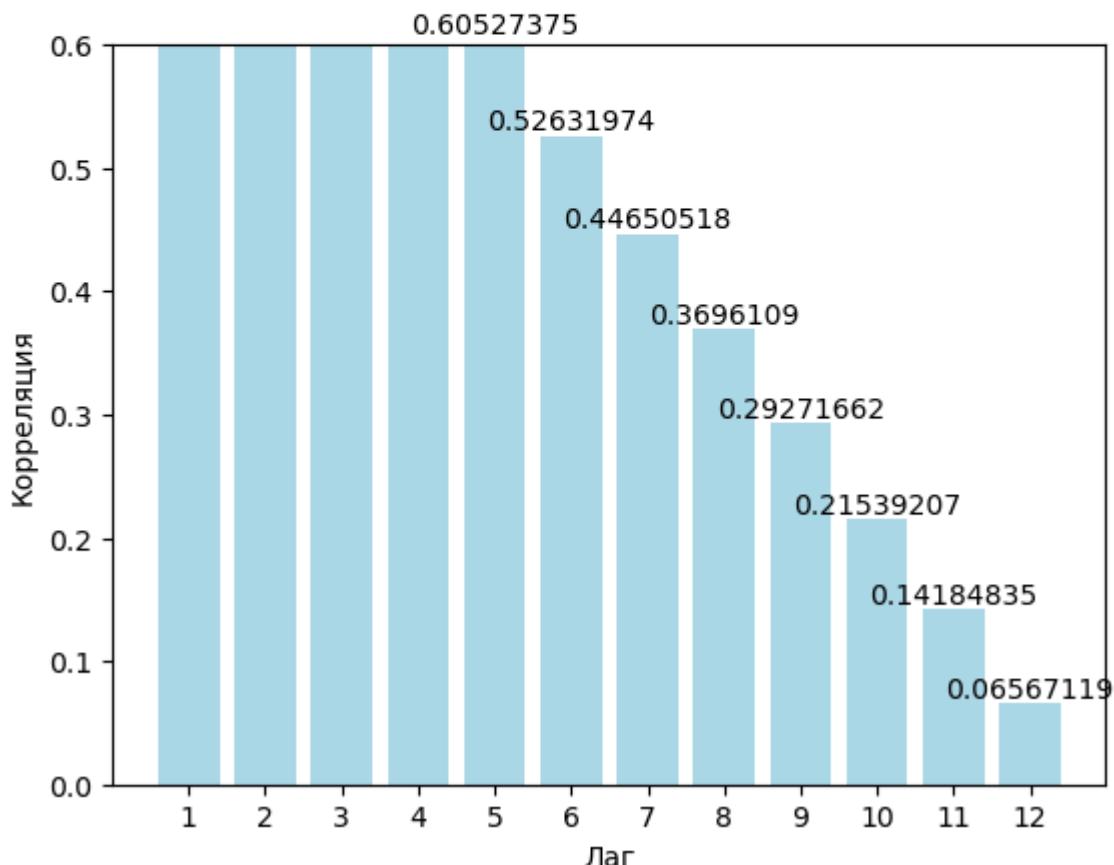
In [17]: addition_model_auto_reg(revenue, 1, False)

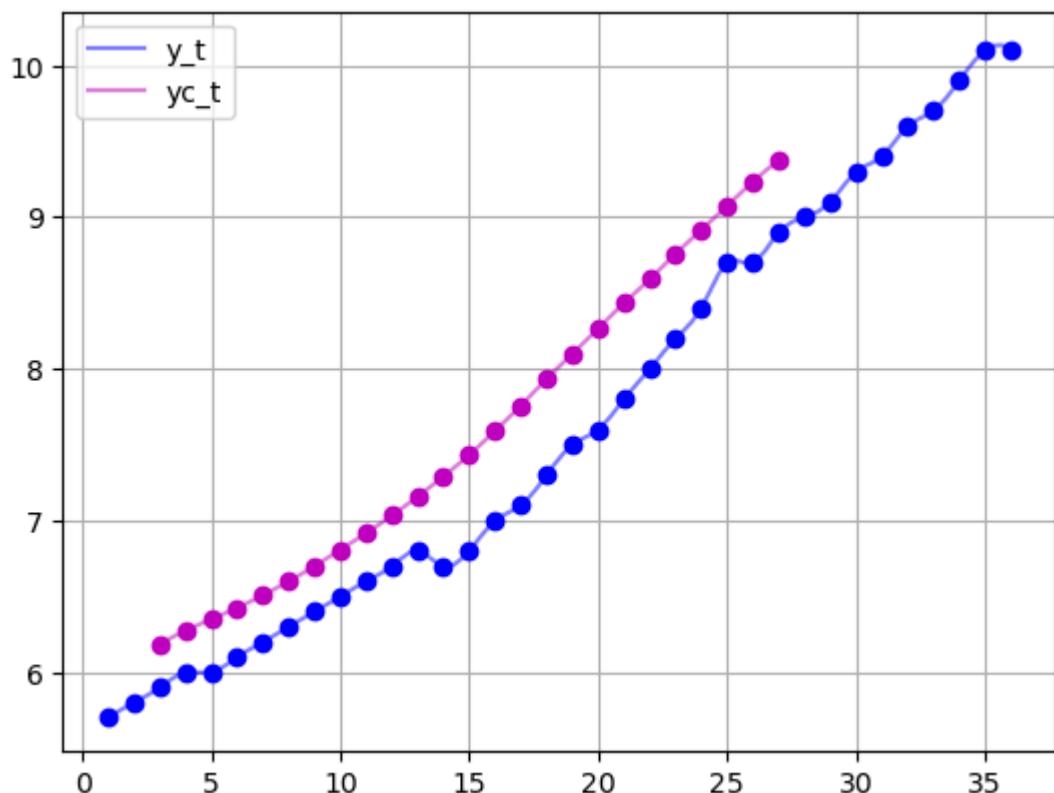
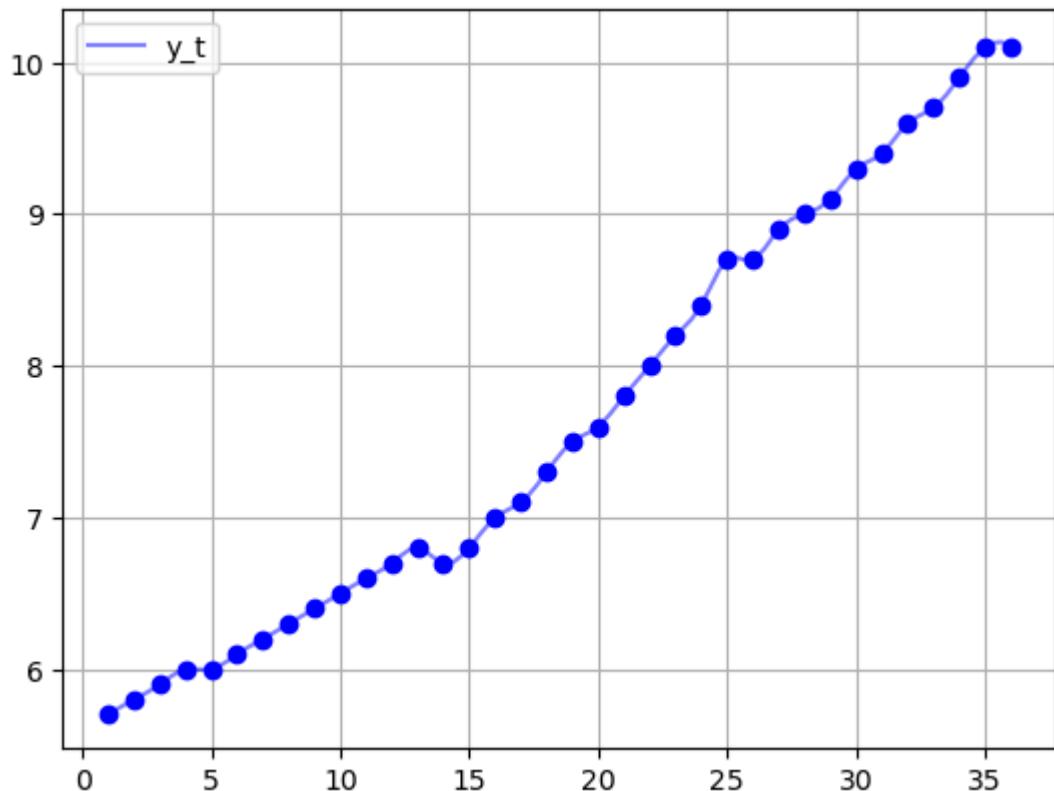
0.9243328

0.84345052

0.76419771

0.68657435





З а т е м р а с с ч и т а е м с р е д н и е о ц е н к и с е з о н н ы х к о ю

◀ ▶

0.1166666666666667

-0.0687499999999964

-0.0562499999999902

-0.0458333333333295

-0.08958333333333446

-0.03333333333333366

-0.02083333333333348

-0.04375000000000062

-0.01666666666666705

0.0083333333333286

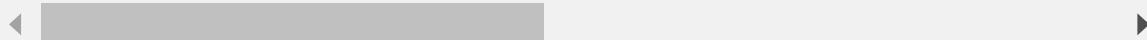
0.03124999999999112

0.050000000000000266

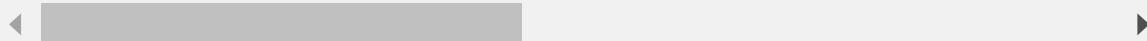
При моделировании периодических колебаний



что сезонные воздействия за период взаимопо-



что сумма значений сезонной компоненты по в-



-0.16875000000000195

Для устранения данного противоречия скоррек-



средние значения оценок сезонной компонент



Величина корректирующего коэффициента бы-



-0.05625000000000654

Теперь рассчитаем скорректированные значения



разность между ее средней оценкой S и коррект



0.1729166666666736

-0.012499999999998991

$1.6306400674181987e - 15$

0.01041666666667705

-0.03333333333333805

0.02291666666666995

0.0354166666666717

0.012500000000000032

0.0395833333333336

0.0645833333333352

0.0874999999999977

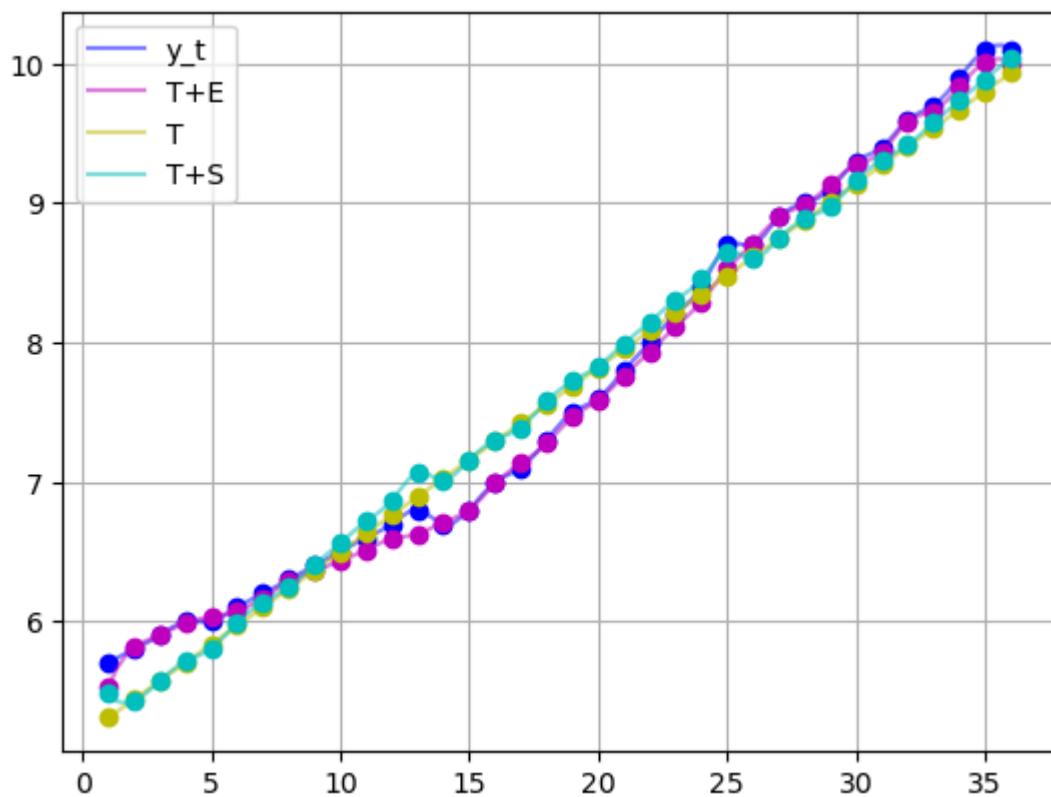
0.1062500000000093

Проверим условия взаимопогашаемости значен

0.506250000000006 ≈ 0

Перенесем полученные значения S в таблицу на

$b_0 = 5.17609126984127$
 $b_1 = 0.132195141570142$



$$\hat{y} = 5.17609126984127 + 0.13219514157014156 \times t + S_i + e_t$$

$$S_i = [0.1729166666666736, -0.01249999999998991, 1.6306400674181987e-15, 0.010416666]$$

Предсказанные значения:

10.240228174603175

10.18700664950665

10.331701791076792

10.4743135993136

10.56275874088374

10.751203882453883

10.895899024024024

11.005177498927498

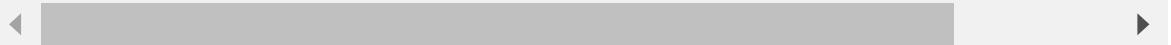
11.164455973830972

11.321651115401115

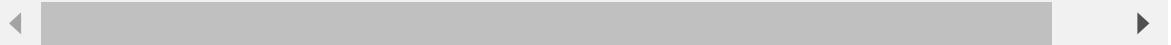
11.476762923637922

11.627708065208067

Рассчитаем стандартную ошибку регрессии: 0.204



Стандартные ошибки прогноза будут следующими:



0.21578717181625764

0.21673085199682812

0.21771978075092516

0.21875334440720862

0.21983091339617078

0.22095184388458639

0.2221154793875261

0.2233211523507131

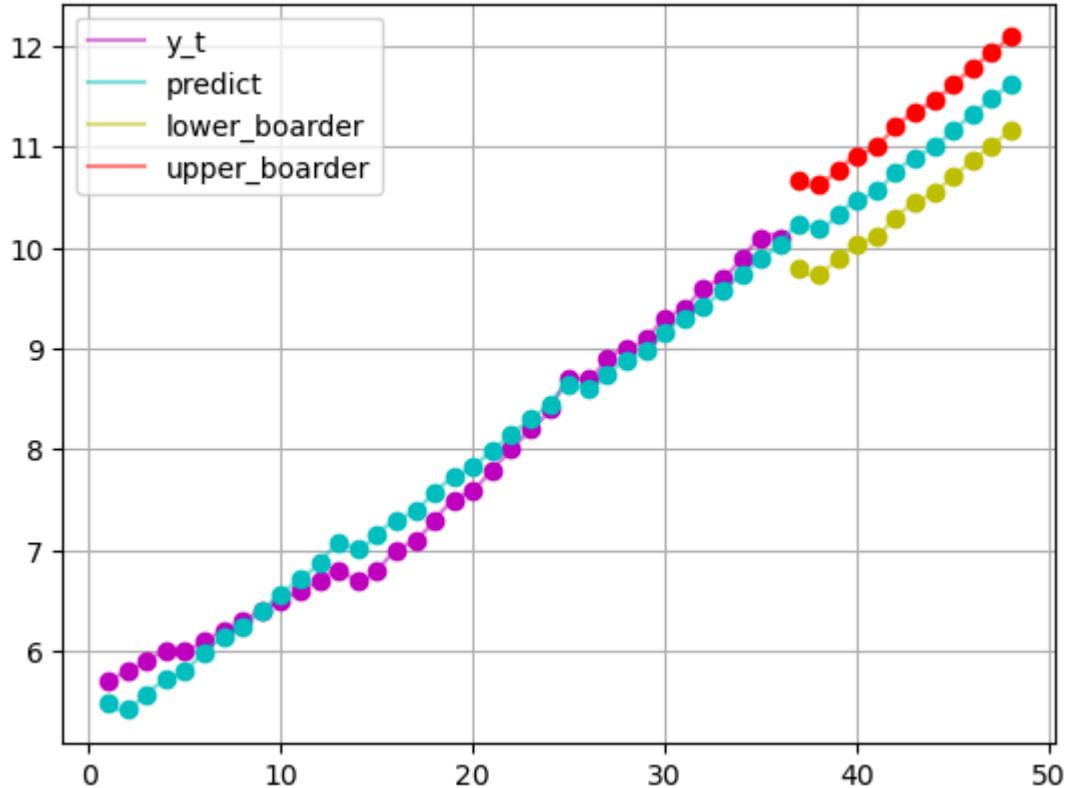
0.2245681856966794

0.22585589432887104

0.22718358658855184

0.22855056566005233

Критическое значение $t_{0.95;34} = 2.03$



– коэффициент детерминации

0.9796688573856972

– средняя ошибка аппроксимации

2.4192557112558037

```
In [18]: multiplication_model_auto_reg(revenue, 1, False)
```

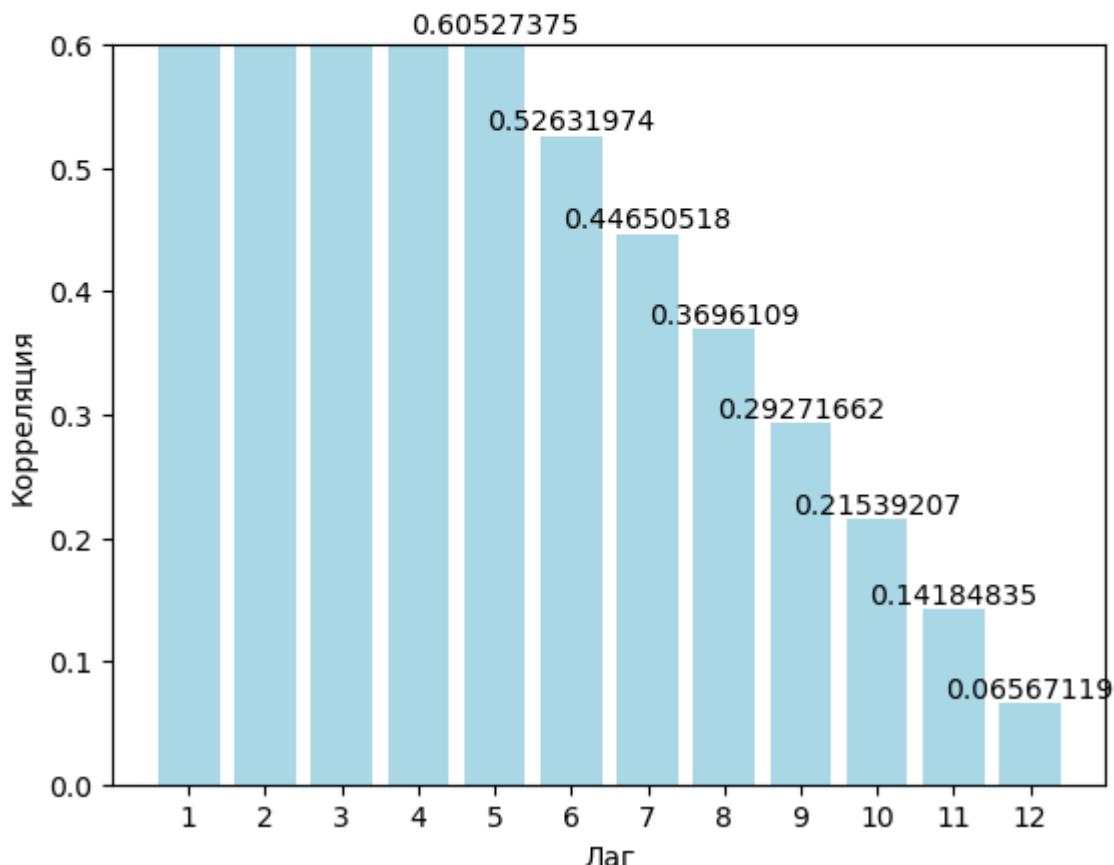
12

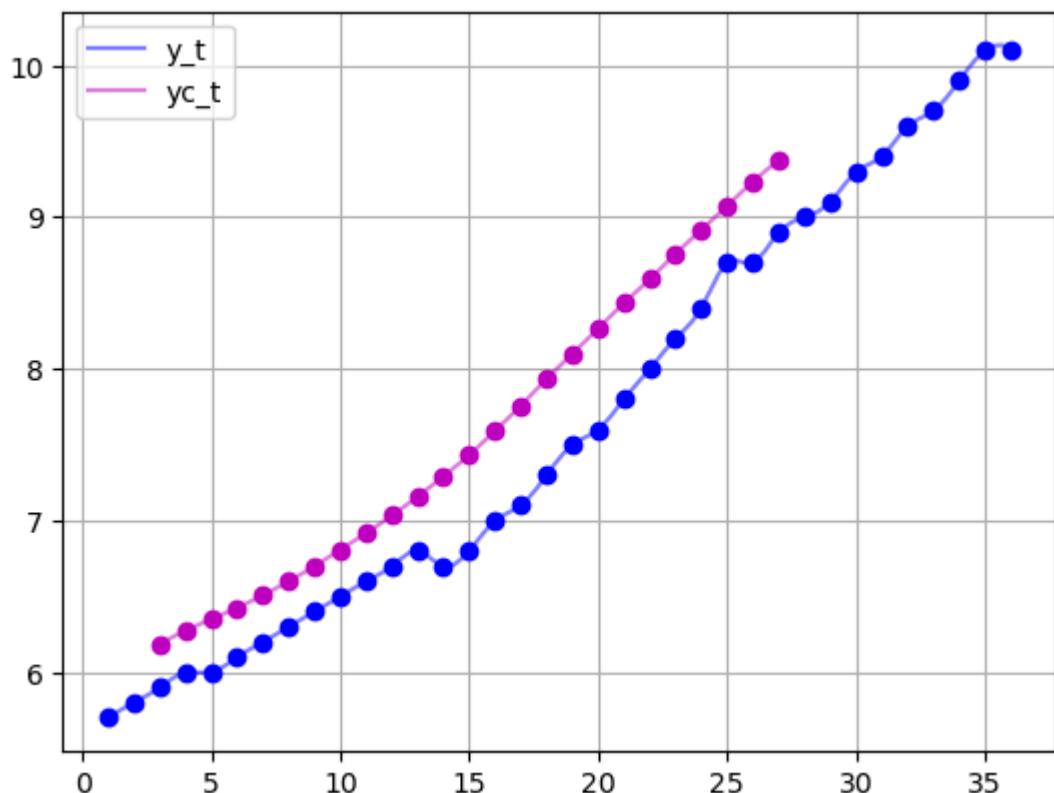
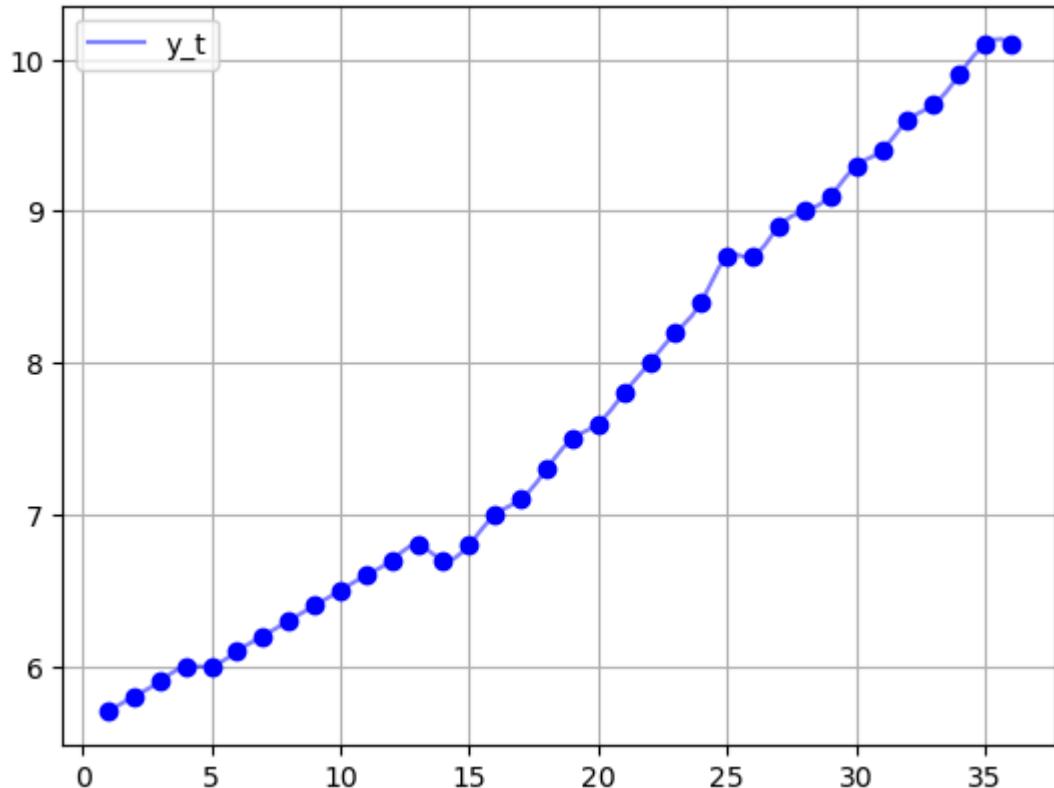
0.9243328

0.84345052

0.76419771

0.68657435





Взимопогашаемость сезонных воздействий в



что сумма значений iS сезонной компоненты S д



то есть для данного примера – четырем. Сумми



1.014406183163164

0.9896012136291817

0.99099126502512

0.9934787995029343

0.9883909010872012

0.9955316038098077

0.9968269161208465

0.994123907504757

0.9980576026306025

1.0015378185481278

1.0045148334354037

1.0067534104722886

11.974214454929434

Рассчитаем корректирующий коэффициент:

0.25053835567184013

Определим скорректированные значения сезона



умножив ее средние оценки iS на корректирую

0.2541476571130466

0.24793306083351258

0.2482813220245503

0.2489045448222989

0.24762983111939577

0.24941885103785902

0.2497433764543482

0.2490661691203063

0.25005171062884995

0.25092363820220975

0.25166949461687843

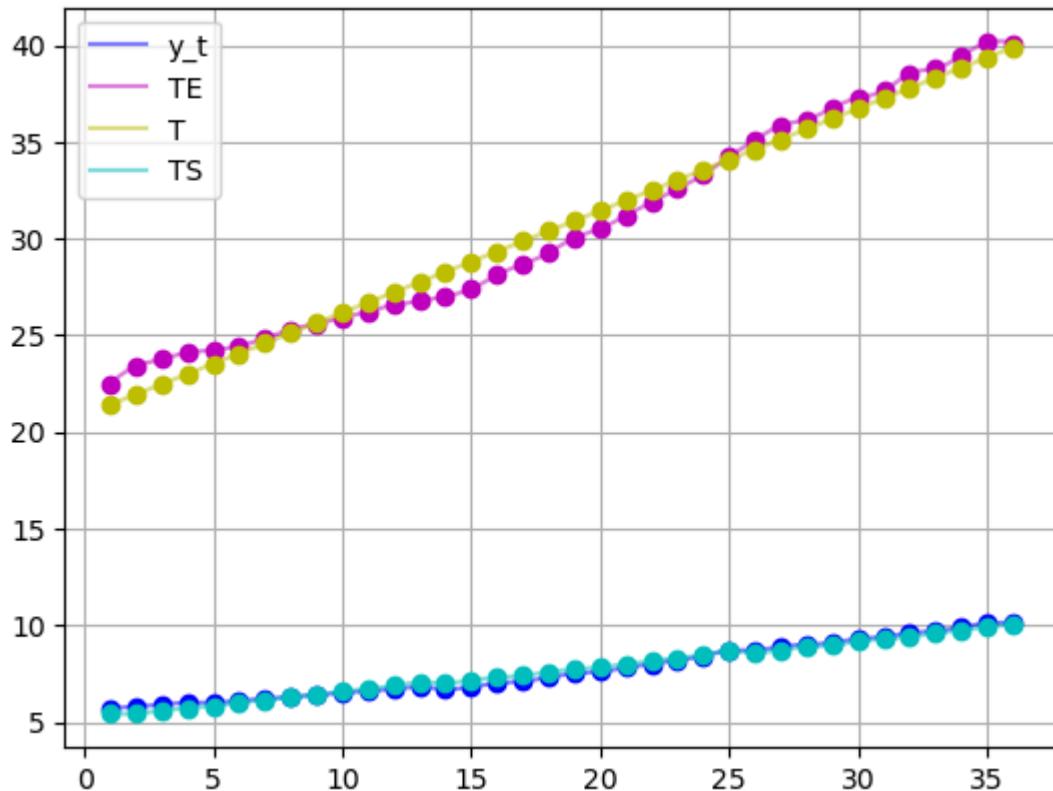
0.2522303440267443

Проверим выполнение свойства взаимопогаша-

3.0

Для выполнения последующих расчетов пере-

$b_0 = 20.8858922386229$
 $b_1 = 0.527919084310927$



$$y = (20.88589223862286 + 0.5279190843109266 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели знач

относительной ошибки E не могут быть исполь

Поэтому для оценки качества модели необходимо

абсолютной ошибки по формуле $E_{abs} = Y_t - (T \cdot S)$, т

значения остатков te представлены в графике 1.

^

$S_i = [0.2541476571130466, 0.24793306083351258, 0.2482813220245503, 0.2489045448222989, 0$



П р е д с к а з а н н ы е з н а ч е н и я :

10.27236832080838

10.152069779894731

10.297402415481146

10.454651876198648

10.53183902922553

10.739600045949027

10.885417916186775

10.987387661665522

11.162871237027812

11.334263423951677

11.500815014232199

11.659601972846739

Р а с с ч и т а е м с т а н д а р т н у ю о ш и б к у р е г р е с с и и : 0.203



С т а н д а р т н ы е о ш и б к и п р о г н о з а б у д у т с л е д у ю щ и



0.21518460039845896

0.21612564541448484

0.21711181265038576

0.21814249014845827

0.21921705009722595

0.2203348504613252

0.22149523658901749

0.22269754279012724

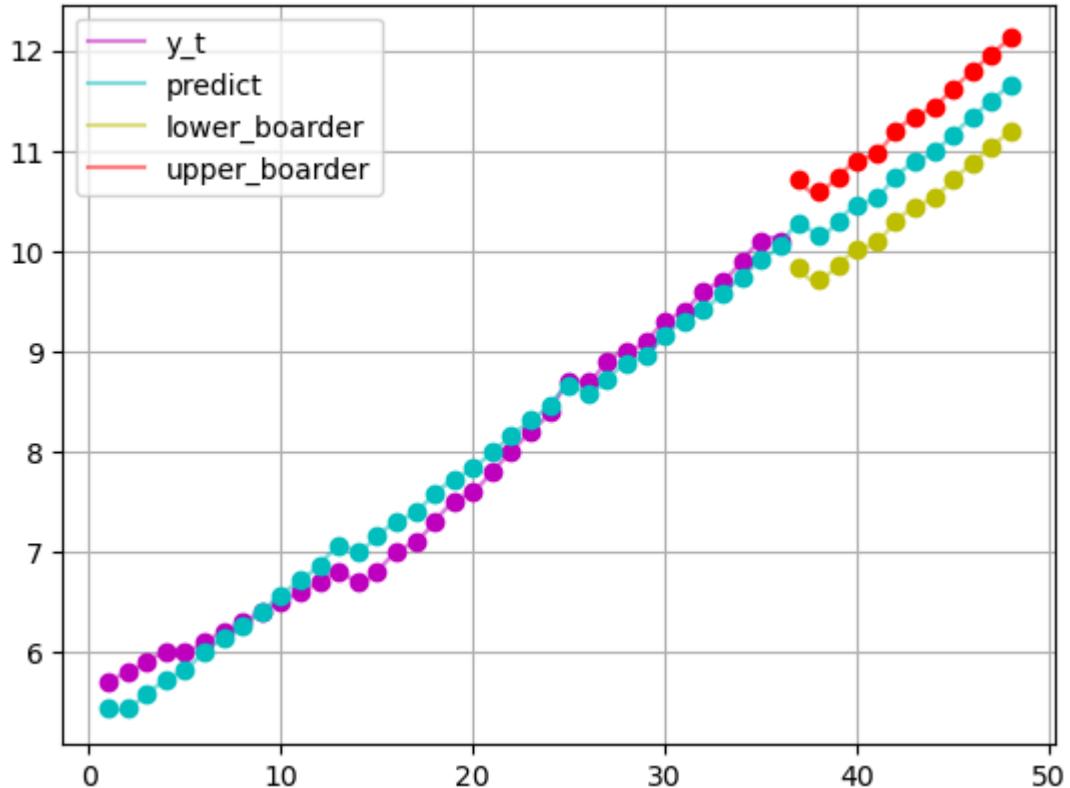
0.22394109387788053

0.22522520666881024

0.22654919143559152

0.2279123533083666

Критическое значение $t_{0.95;34} = 2.03$



– коэффициент детерминации

0.9797763320338649

– средняя ошибка аппроксимации

2.419278613698379

Организации

```
In [19]: organizations = '''244 207 141
241 202 136
237 196 131
235 190 126
232 184 121
229 179 116
227 173 112
222 167 108
219 162 104
215 156 100
212 151 96
210 145 93'''

organizations
```

```
organizations = np.array(list(map(lambda x: list(map(float,x.replace(',','.').split())), organizations))
organizations = pd.DataFrame(organizations, columns=[2005,2006,2007],index=list(range(1, len(organizations)+1)))
organizations = organizations.T
organizations
```

Out[19]:

	1	2	3	4	5	6	7	8	9	10	11	12
2005	244.0	241.0	237.0	235.0	232.0	229.0	227.0	222.0	219.0	215.0	212.0	210.0
2006	207.0	202.0	196.0	190.0	184.0	179.0	173.0	167.0	162.0	156.0	151.0	145.0
2007	141.0	136.0	131.0	126.0	121.0	116.0	112.0	108.0	104.0	100.0	96.0	93.0

In []: `#organizations.to_excel('organizations.xlsx')`In [21]: `addition_model_auto_reg(organizations, 1, False)`

12

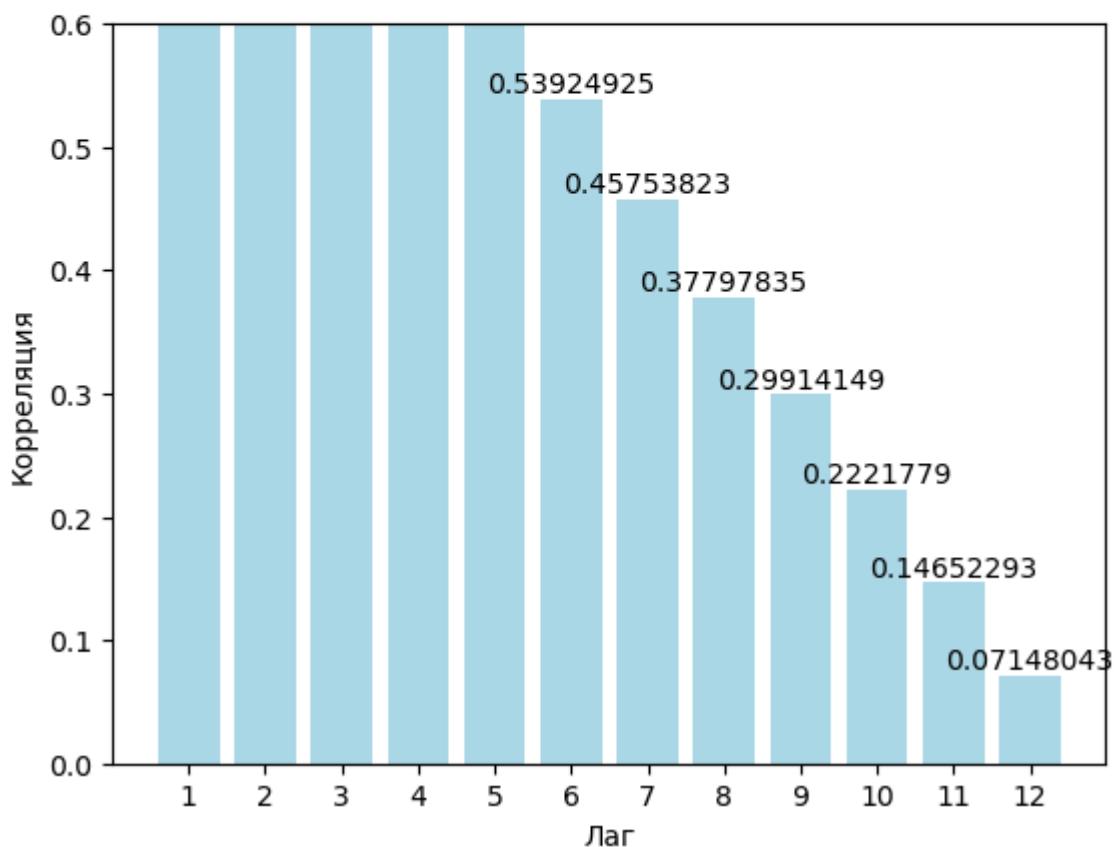
0.92744923

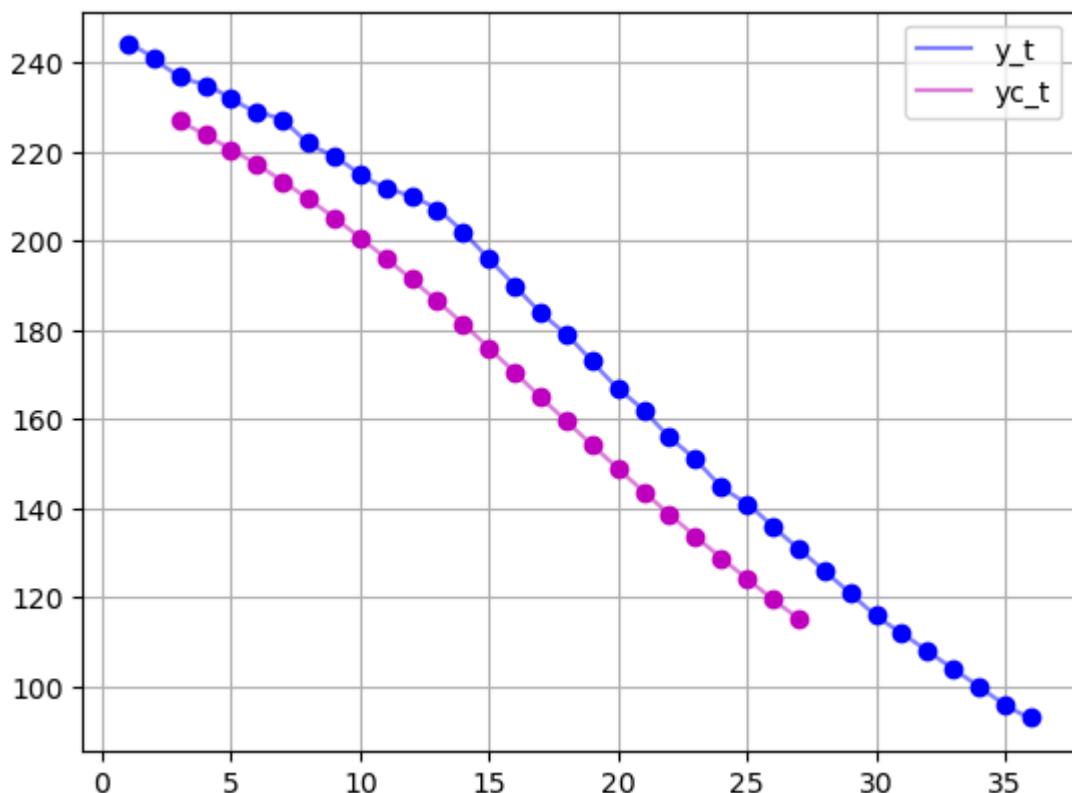
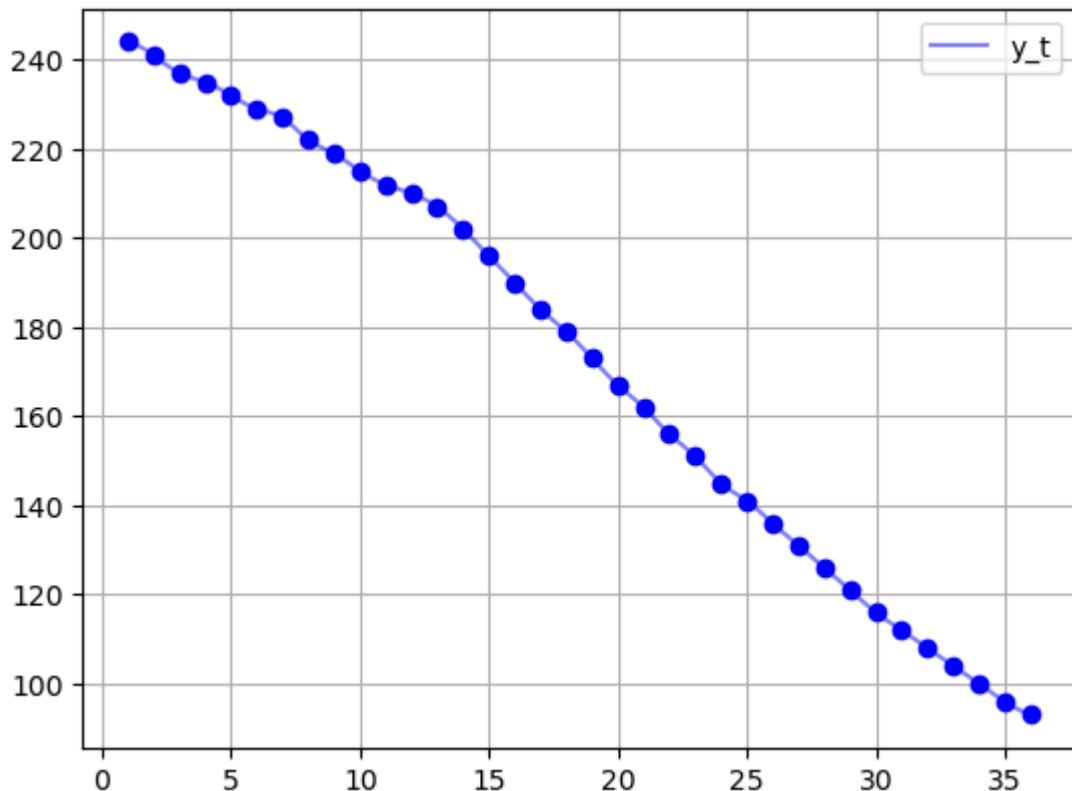
0.8522454

0.77638427

0.69837798

0.61930509





З а т е м р а с с ч и т а е м с р е д н и е о ц е н к и с е з о н н ы х к о ю

◀ ▶

1.8958333333333428

1.6666666666666856

0.9375000000000142

0.229166666666785

-0.4583333333333286

-0.6041666666666572

0.6875

-0.4791666666666714

-0.0833333333334281

-0.6041666666666714

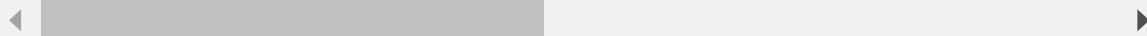
-0.02083333333328596

0.645833333333428

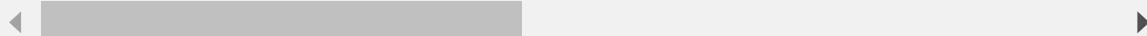
При моделировании периодических колебаний



что сезонные воздействия за период взаимопо-



что сумма значений сезонной компоненты по в-



3.812500000000064

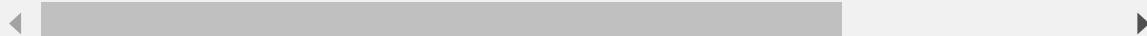
Для устранения данного противоречия скоррек-



средние значения оценок сезонной компонент



Величина корректирующего коэффициента бы-

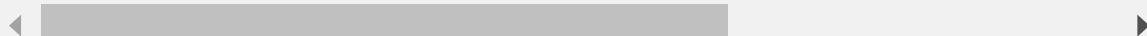


1.270833333333546

Теперь рассчитаем скорректированные значения



разность между ее средней оценкой S и коррект



0.624999999999882

0.3958333333333104

-0.3333333333334036

-1.041666666666676

-1.729166666666832

-1.8750000000000118

-0.583333333333546

-1.7500000000000026

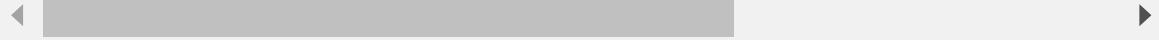
-1.3541666666666974

-1.875000000000026

-1.2916666666666832

-0.6250000000000118

Проверим условия взаимопогашаемости значен



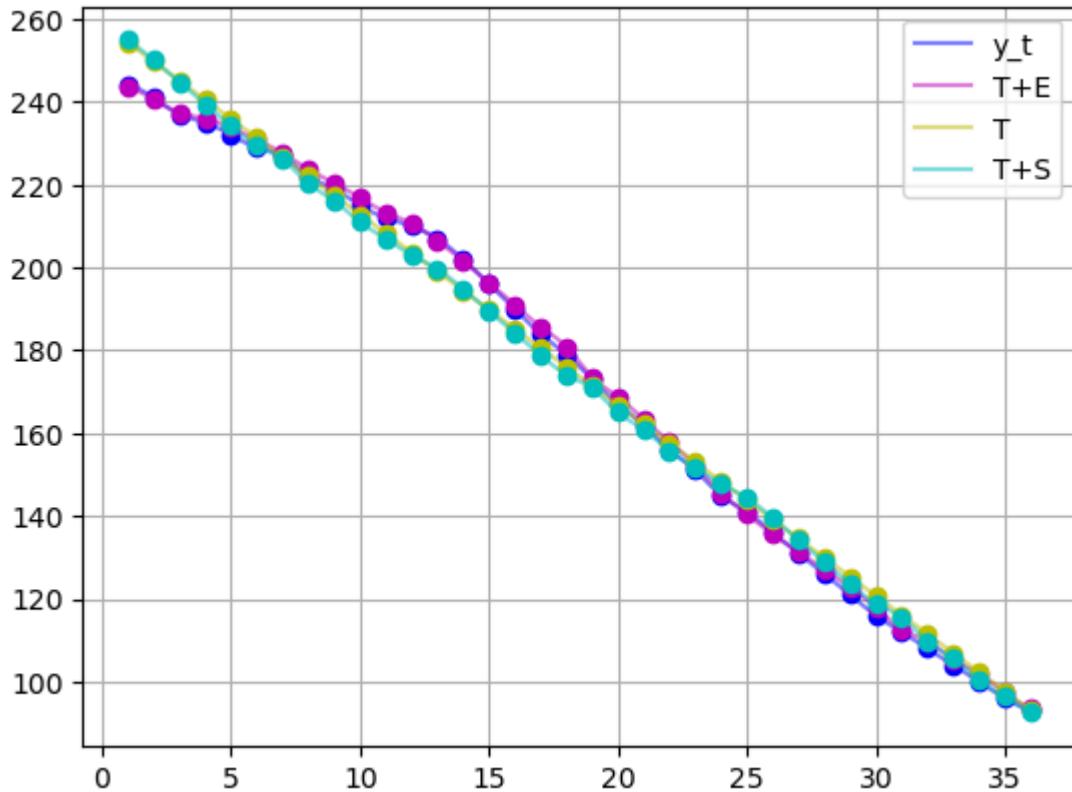
-11.437500000000192 ≈ 0

Перенесем полученные значения S в таблицу на



$b_0 = 258.924107142857$

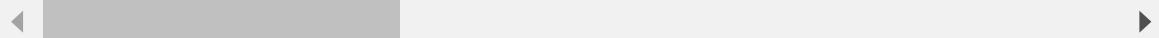
$b_1 = -4.60653957528958$



$$y = 258.92410714285717 + -4.606539575289576 \times t + S_i + e_t$$

^

$$S_i = [0.6249999999999882, 0.3958333333333104, -0.3333333333334036, -1.041666666666666]$$



Предсказанные значения:

89.10714285714285

84.27143661518663

78.93573037323037

73.62085746460748

68.32681788931788

63.57444498069496

60.25957207207206

54.4863658301158

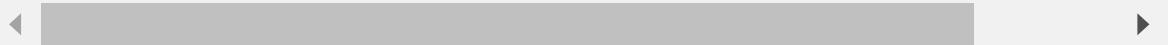
50.27565958815957

45.14828667953665

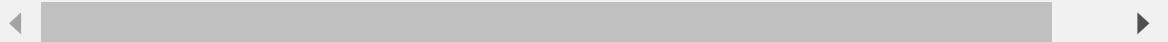
41.125080437580436

37.18520752895752

Рассчитаем стандартную ошибку регрессии: 4.588



Стандартные ошибки прогноза будут следующими



4.846883832909386

4.868080219015872

4.890292951820089

4.913508247399331

4.937711964735653

4.9628896424279265

4.989026534899923

5.016107647942282

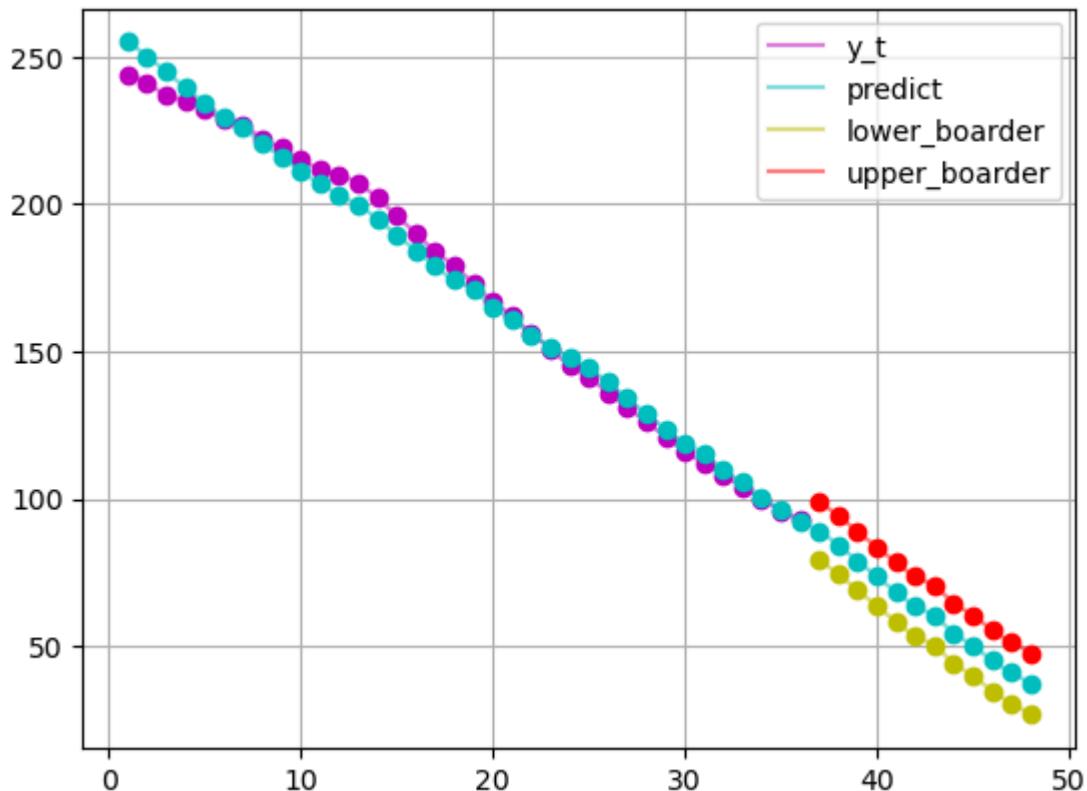
5.044117773441354

5.0730415231635195

5.102863361479309

5.133567636927261

Критическое значение $t_{0.95;34} = 2.03$



– коэффициент детерминации

0.9914515904639293

– средняя ошибка аппроксимации

1.9818659684374842

```
In [22]: multiplication_model_auto_reg(organizations, 1, False)
```

12

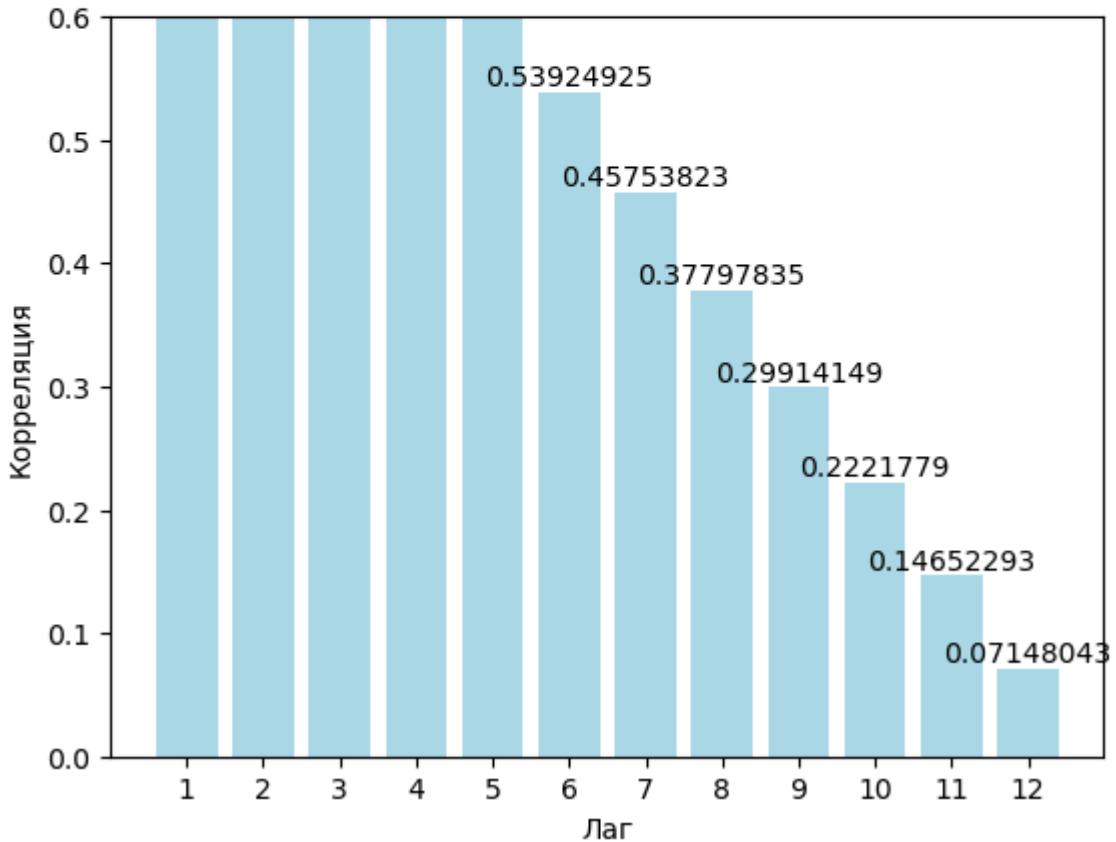
0.92744923

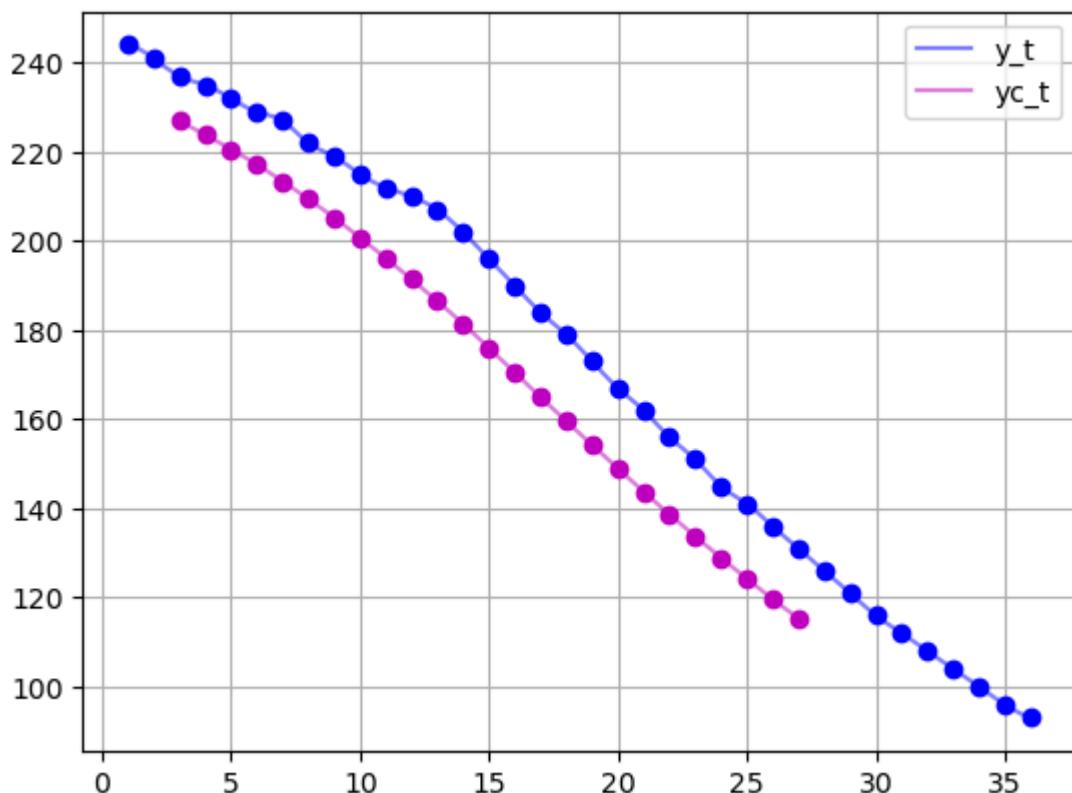
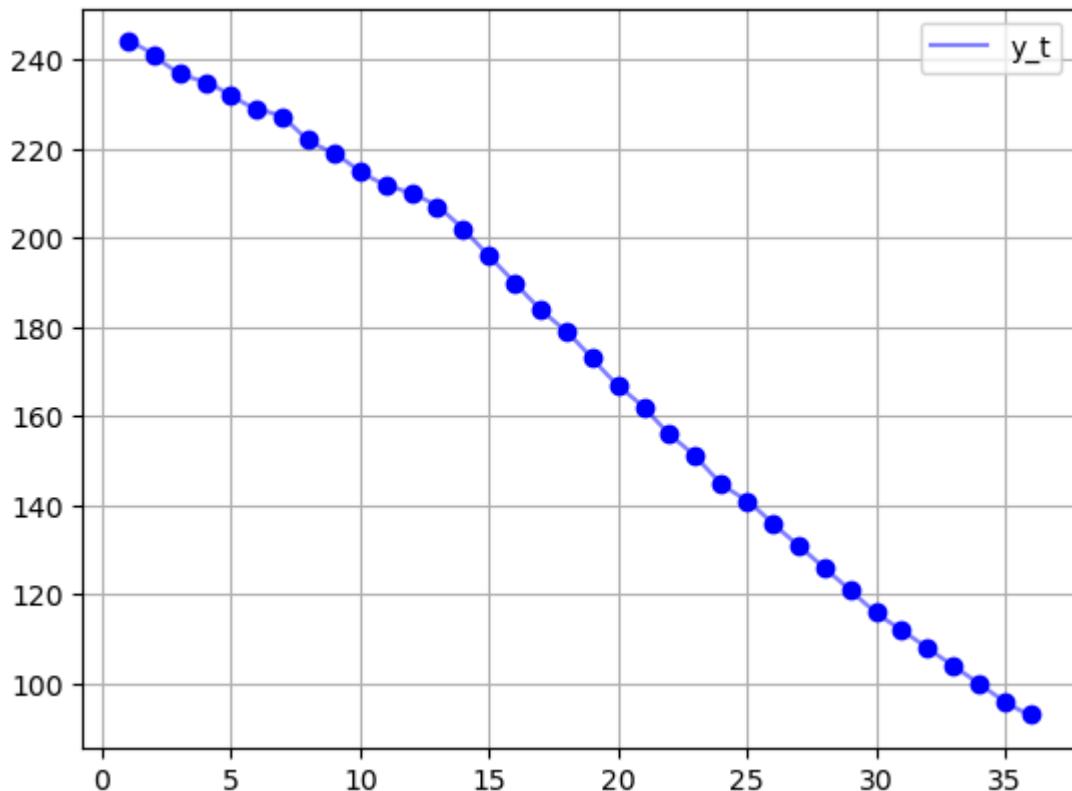
0.8522454

0.77638427

0.69837798

0.61930509





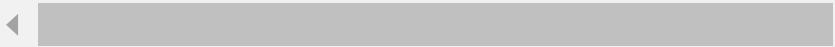
Взимопогашаемость сезонных воздействий в



что сумма значений $i S$ сезонной компоненты S д



то есть для данного примера – четырем. Сумми



1.0091145360169893

1.0081581878566217

1.0044286638864848

1.0004521921155987

0.9961843128044787

0.9944330623118021

1.0028836018852658

0.9972957512527096

0.9993869631804015

0.9964017533962055

0.9993185764128654

1.001734027859773

12.009791628979197

Рассчитаем корректирующий коэффициент:

0.24979617404527713

Определим скорректированные значения сезона



умножив ее средние оценки iS на корректирую



0.25207295027051896

0.2518340581590039

0.2509024373402535

0.24990912990568712

0.24884302998248237

0.24840557430961682

0.25051648676368626

0.24912066305453726

0.24964303979319255

0.24889734581037784

0.24962595704030668

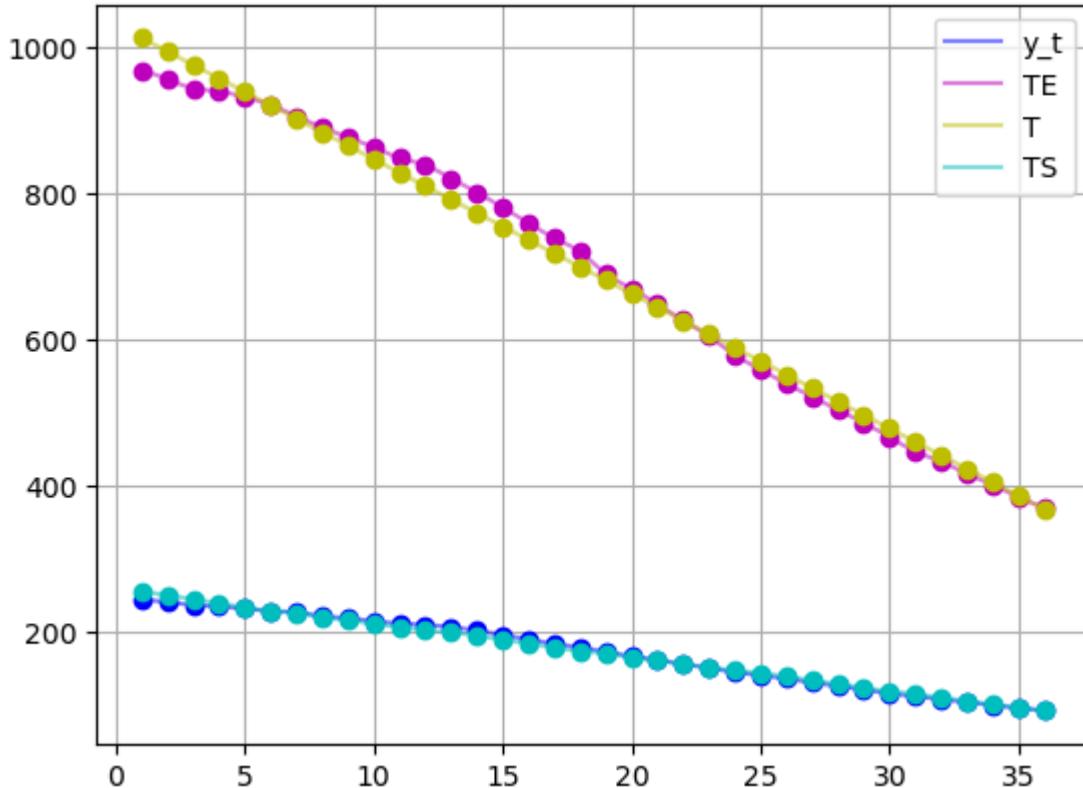
0.25022932757033634

Проверим выполнение свойства взаимопогаша-

2.9999999999999996

Для выполнения последующих расчетов пере-

$b_0 = 1031.80681088522$
 $b_1 = -18.4297811073839$



$$y = (1031.8068108852242 + -18.429781107383885 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели знач-

относительной ошибки E не могут быть исполь-

Поэтому для оценки качества модели необходимо

абсолютной ошибки по формуле $E_{abs} = Yt - (T \cdot S)$, т

значения остатков te представлены в графике 1.

^

$S_i = [0.25207295027051896, 0.2518340581590039, 0.2509024373402535, 0.24990912990568712,$



П р е д с к а з а н н ы е з н а ч е н и я :

88.20156295566315

83.47672686565016

78.5438407352678

73.62711990314513

68.72690769755343

64.02802829854907

59.9551646434501

55.02986813737045

50.544392722036264

45.806290949674406

41.33983054014975

36.82808124938327

Р а с с ч и т а е м с та н д арт н у ю о ш и б к у р е г р ес с и и : 4.655



С та н д арт н ы е о ш и б ки п р о г н о з а б у д у т с л е д у ю щ и



4.917906735706269

4.939413719079367

4.961951941995827

4.98550741855282

5.010065800519649

5.035612414587752

5.062132299109372

5.089610240160313

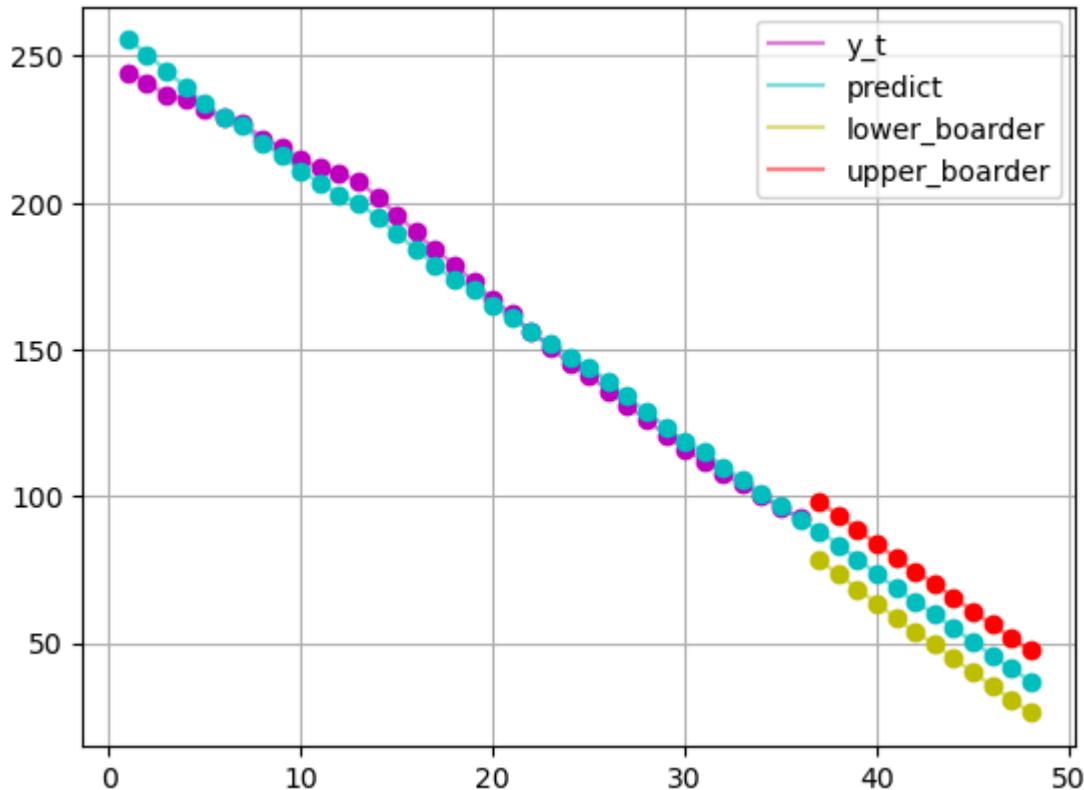
5.118030806777684

5.147378385239275

5.177637212267181

5.208791407054191

Критическое значение $t_{0.95;34} = 2.03$



- коэффициент детерминации

0.9912043584595985

- средняя ошибка аппроксимации

2.003322777496183

Поквартальная динамика оборота розничной торговли региона

```
In [23]: organizations = '''317,1 380,6 444 507,4 570,8 634,3  
481,6 578 674,3 770,6 866,9 963,3  
644,4 773,3 902,1 1031 1159,9 1288,8  
792,1 950,6 1109 1267,4 1425,8 1584,3  
425 510 595 680 765 850  
601 721,2 841,4 961,6 1081,8 1202  
762,8 915,3 1067,9 1220,4 1373 1525,5  
972,3 1166,7 1361,2 1555,6 1750,1 1944,5  
545,6 654,8 763,9 873 982,1 1091,3  
751,5 901,8 1052,1 1202,4 1352,7 1503  
968,8 1162,5 1356,3 1550 1743,8 1937,5  
1256 1507,2 1758,4 2009,6 2260,8 2512  
647,3 776,7 906,2 1035,6 1165,1 1294,5  
647,3 776,7 906,2 1035,6 1165,1 1294,5  
1237,5 1485 1732,5 1980 2227,5 2475  
1641,9 1970,3 2298,6 2627 2955,4 3283,8'''  
  
organizations = np.array(list(map(lambda x: list(map(float,x.replace(',','.')),  
organizations = pd.DataFrame(organizations, columns = range(1,7))
```

```
#organizations = organizations.T
```

```
organizations
```

Out[23]:

	1	2	3	4	5	6
0	317.1	380.6	444.0	507.4	570.8	634.3
1	481.6	578.0	674.3	770.6	866.9	963.3
2	644.4	773.3	902.1	1031.0	1159.9	1288.8
3	792.1	950.6	1109.0	1267.4	1425.8	1584.3
4	425.0	510.0	595.0	680.0	765.0	850.0
5	601.0	721.2	841.4	961.6	1081.8	1202.0
6	762.8	915.3	1067.9	1220.4	1373.0	1525.5
7	972.3	1166.7	1361.2	1555.6	1750.1	1944.5
8	545.6	654.8	763.9	873.0	982.1	1091.3
9	751.5	901.8	1052.1	1202.4	1352.7	1503.0
10	968.8	1162.5	1356.3	1550.0	1743.8	1937.5
11	1256.0	1507.2	1758.4	2009.6	2260.8	2512.0
12	647.3	776.7	906.2	1035.6	1165.1	1294.5
13	647.3	776.7	906.2	1035.6	1165.1	1294.5
14	1237.5	1485.0	1732.5	1980.0	2227.5	2475.0
15	1641.9	1970.3	2298.6	2627.0	2955.4	3283.8

In []:

```
table = 80

#organizations.to_excel('organizations2.xlsx')

for i in organizations.columns:
    print(f"\n{'='*table}\n{f'Данные задания {i}':^{table}}\n{'='*table}")
    data = pd.DataFrame(organizations.loc[:,i].values.reshape(4,4), columns=[1,2])
    display()

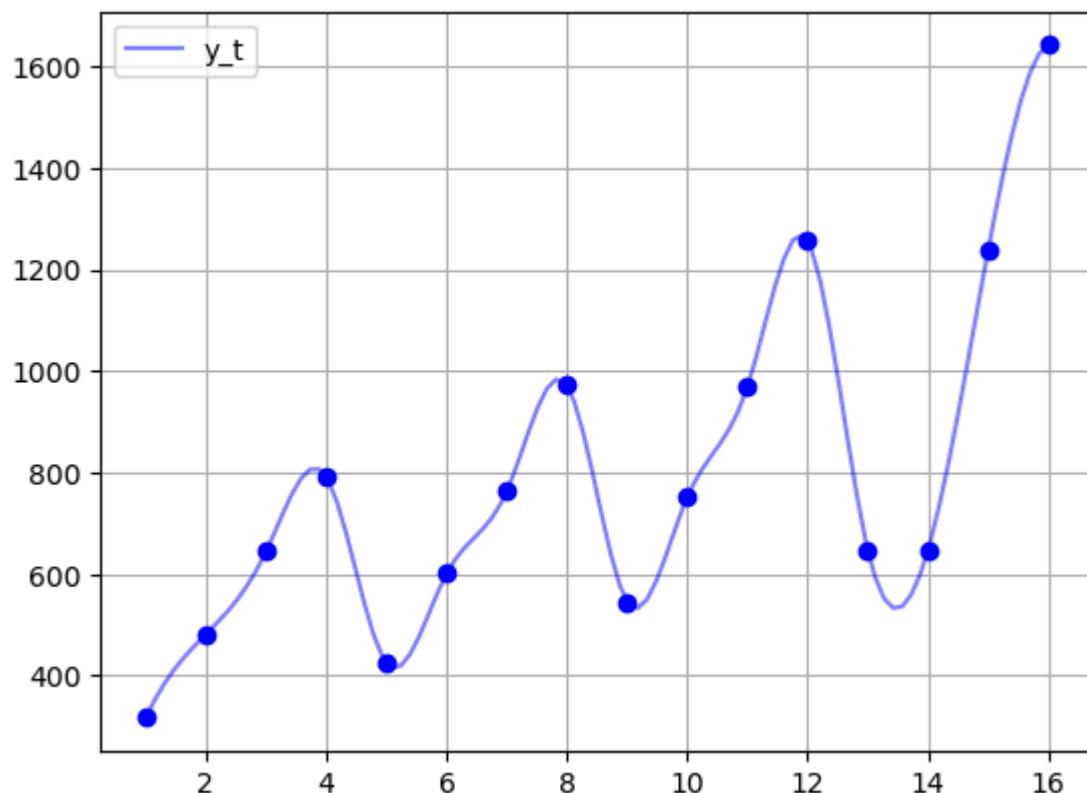
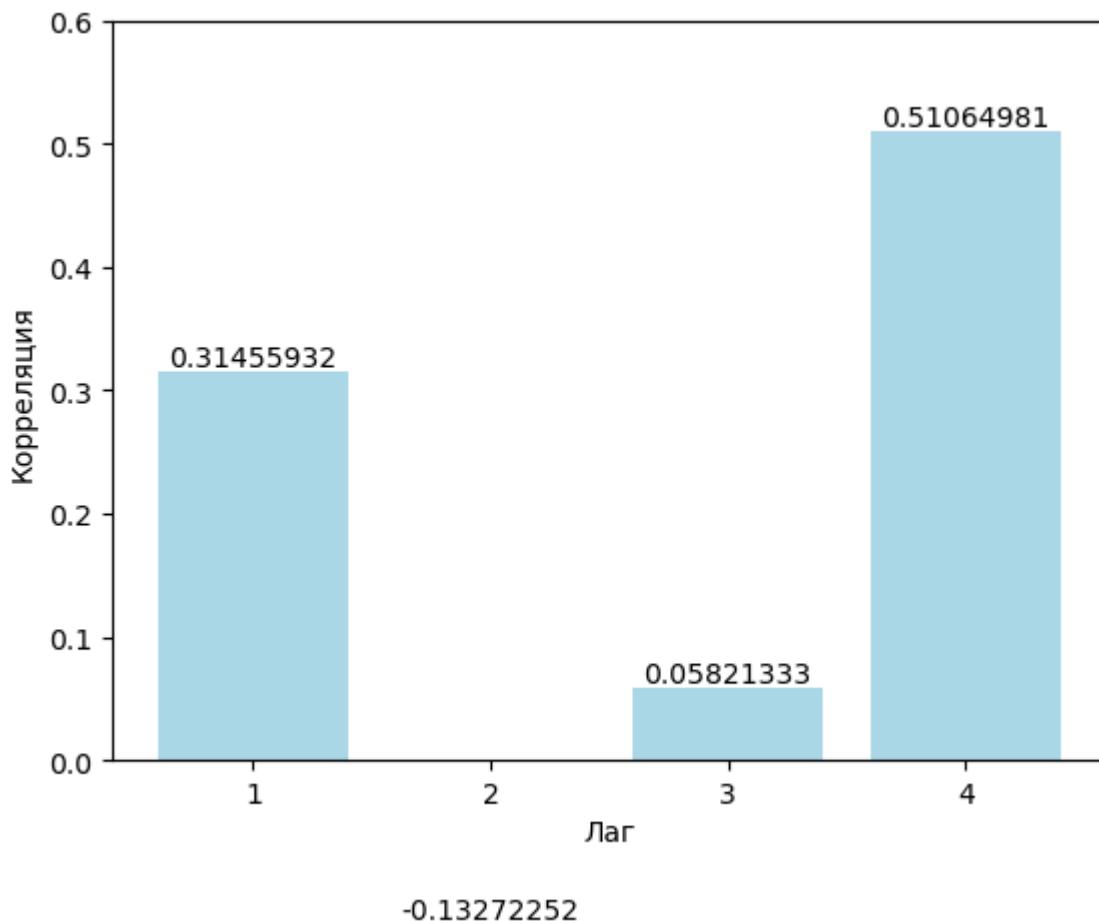
    print(f"\n--- Аддитивная модель {i}_1 ---")
    addition_model_auto_reg(data, 1, False)

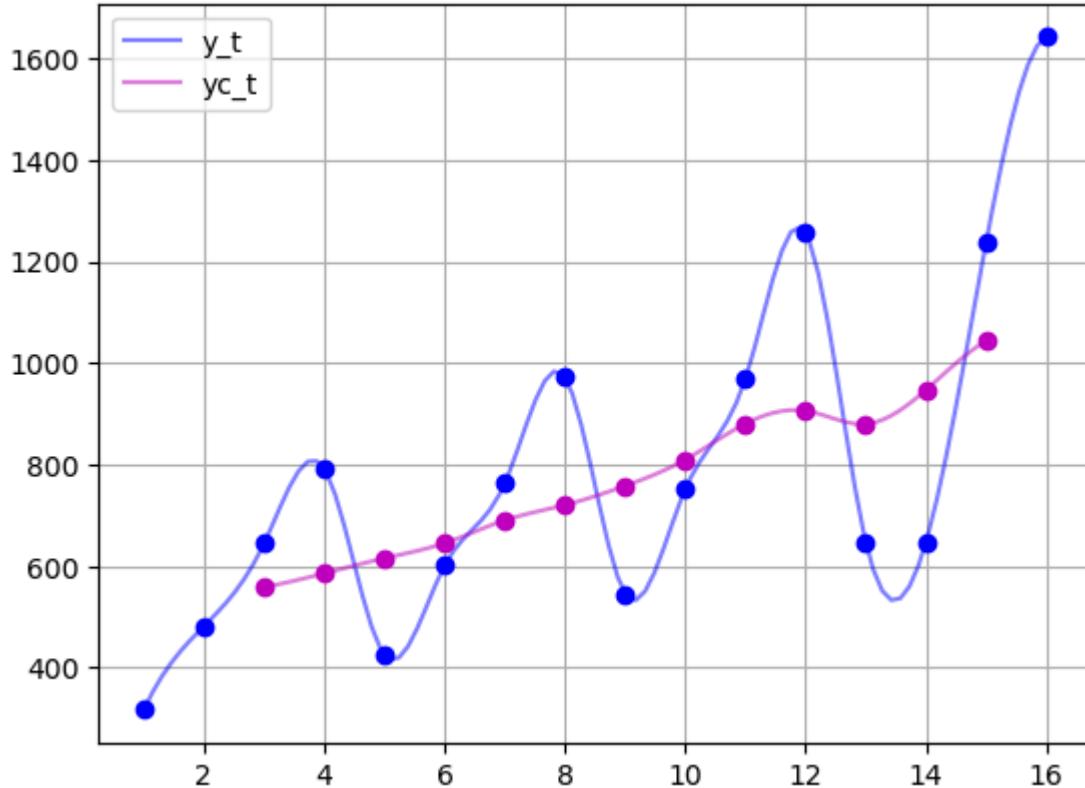
    print(f"\n--- Мультипликативная модель {i}_2 ---")
    multiplication_model_auto_reg(data, 1, False)
```

```
=====
Данные задания 1
=====
```

```
--- Аддитивная модель 1_1 ---
```

4





Затем рассчитаем средние оценки сезонных колебаний

-236.58749999999998

-169.4083333333333

68.39166666666667

262.52916666666664

При моделировании периодических колебаний

что сезонные воздействия за период взаимопо-

что сумма значений сезонной компоненты по-

-75.0749999999993

Для устранения данного противоречия скоррек-

средние значения оценок сезонной компонент

величина корректирующего коэффициента буде-

-18.768749999999983

Теперь рассчитаем скорректированные значения

разность между ее средней оценкой S и коррект

-217.81875

-150.63958333333335

87.16041666666668

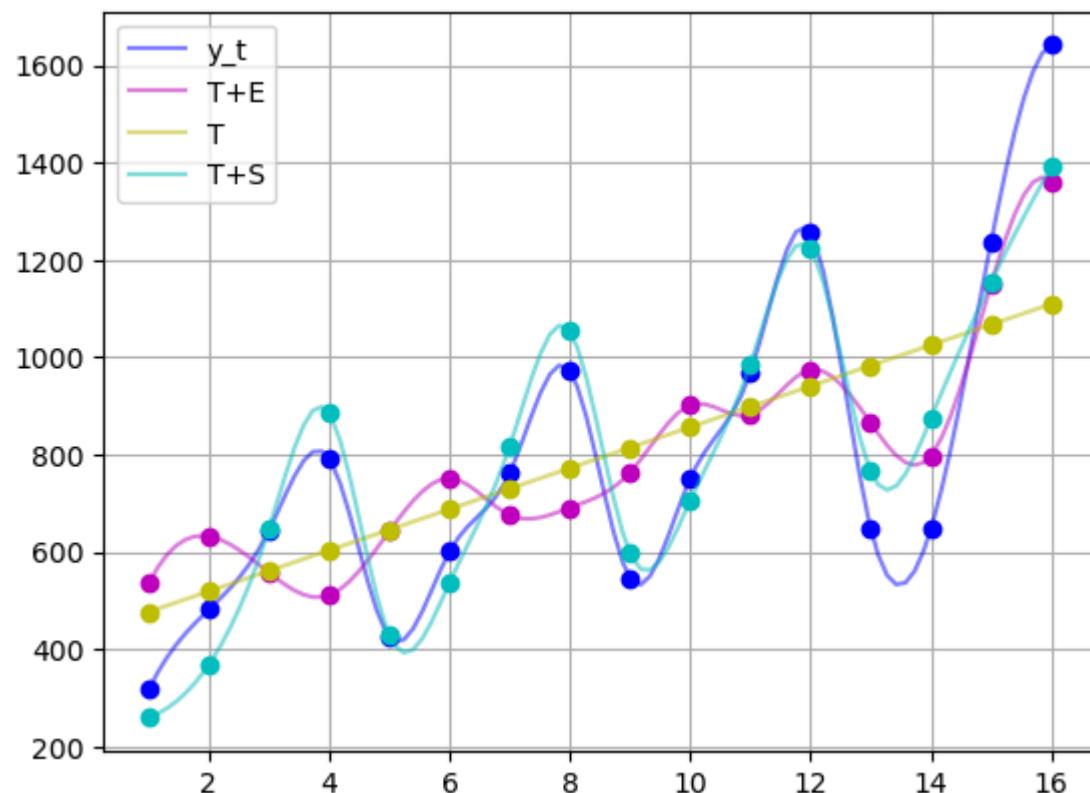
281.29791666666665

Проверим условие взаимопогашаемости значений

$-5.684341886080802e - 14 \approx 0$

Перенесем полученные значения S в таблицу на

$b_0 = 434.735000000000$
 $b_1 = 42.1797058823529$



$$\hat{y} = 434.735 + 42.17970588235294 \times t + S_i + e_t$$

$$S_i = [-217.81875, -150.63958333333335, 87.16041666666668, 281.29791666666665]$$

Предсказанные значения:

933.971249999999

1043.3301225490197

1323.3098284313724

1559.6270343137255

Рассчитаем стандартную ошибку регрессии: 113.3



Стандартные ошибки прогноза будут следующими

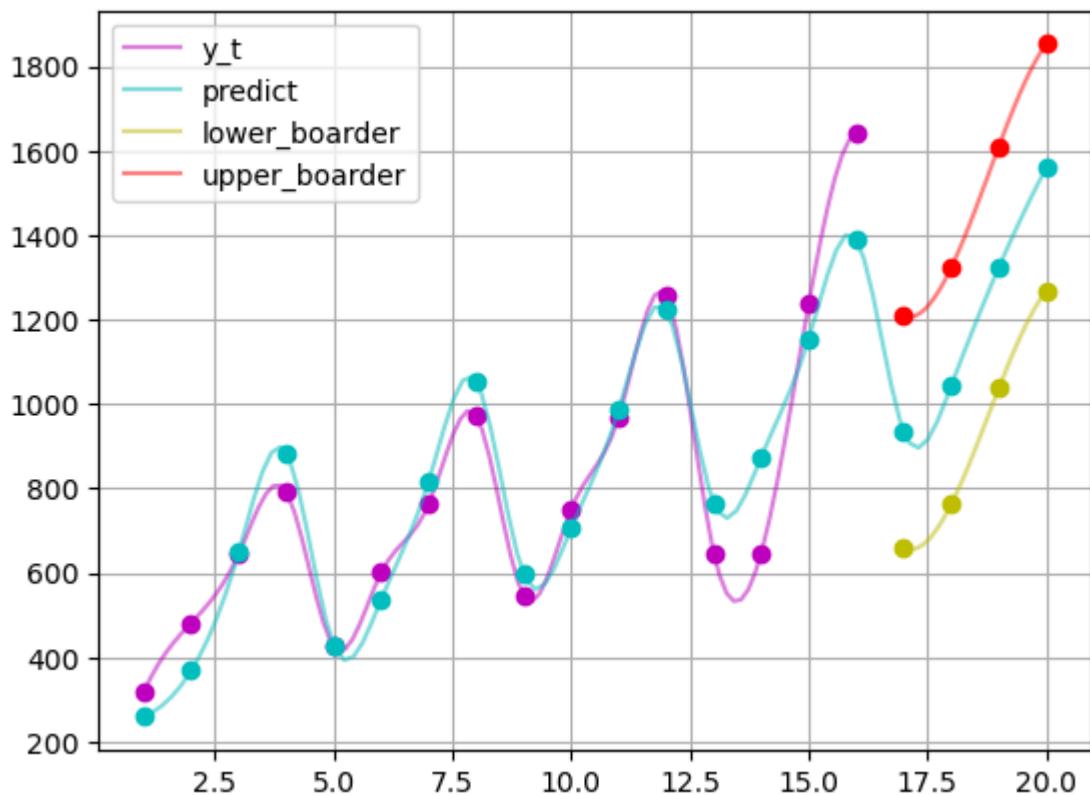


128.01840020422148

130.64919017311342

133.51150556181148

136.59079201879612

Критическое значение $t_{0.95;14} = 2.14$ 

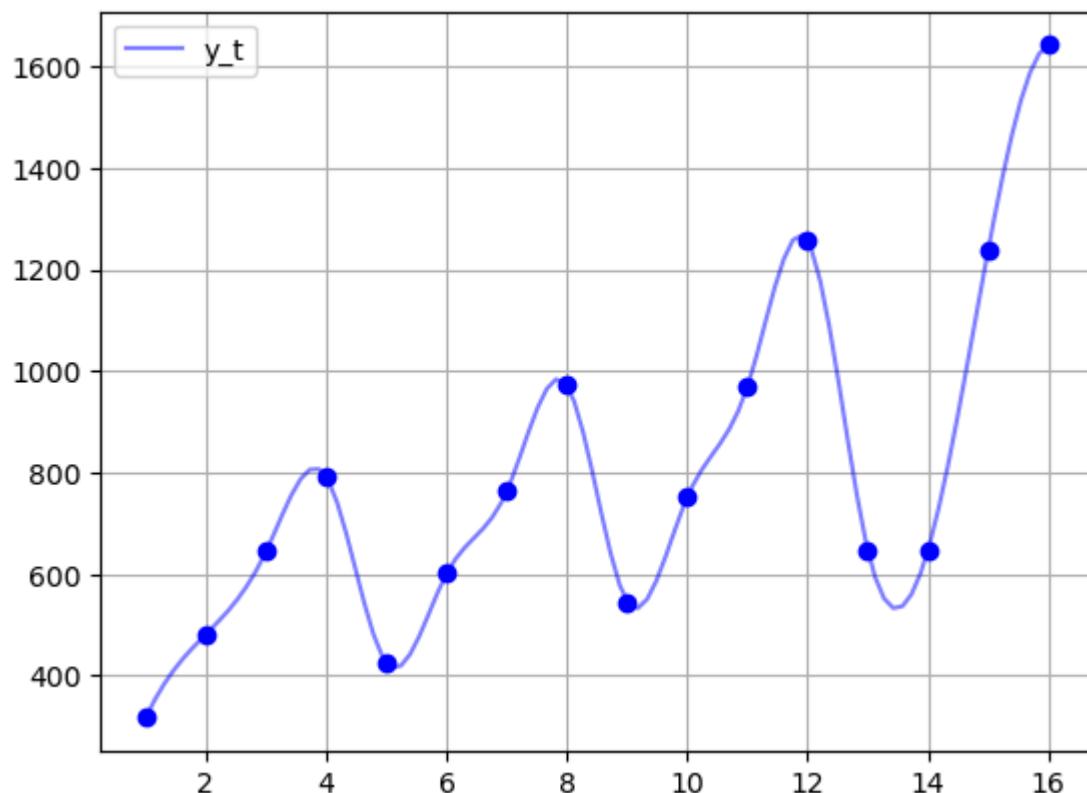
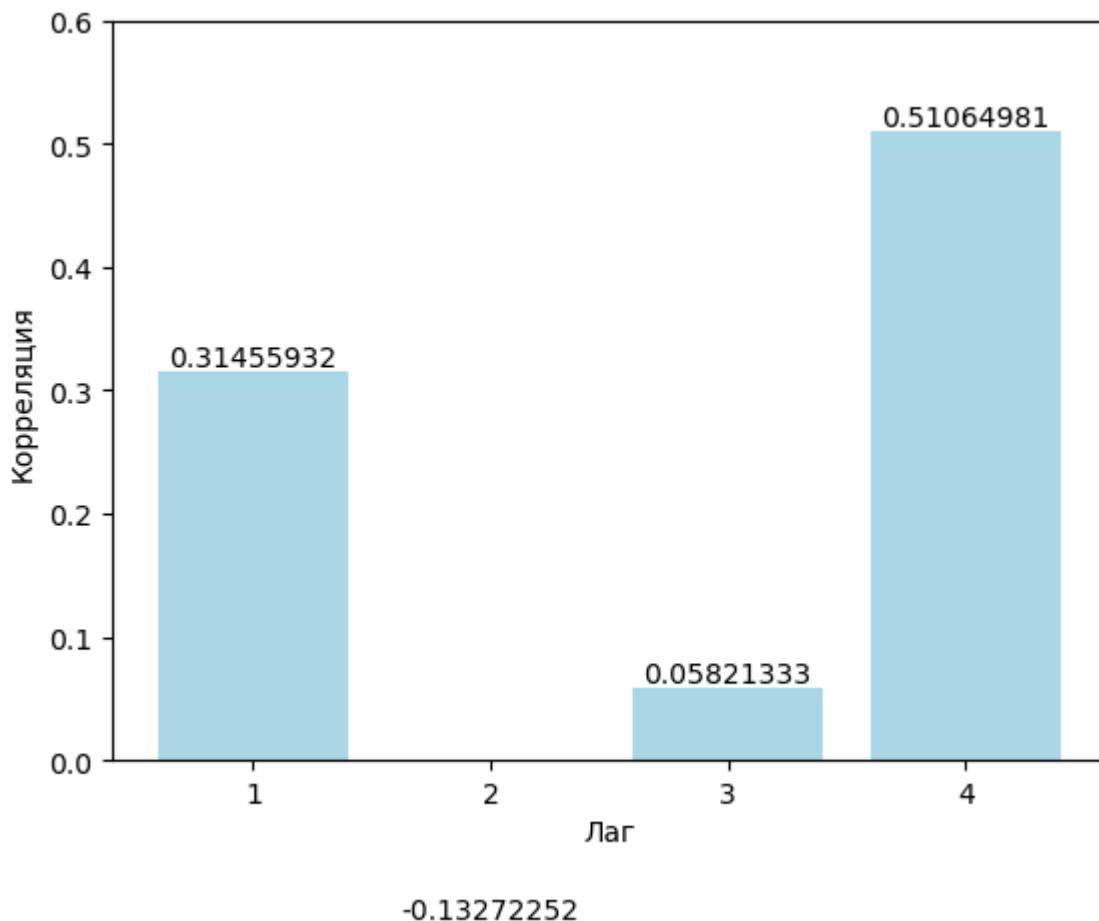
- коэффициент детерминации

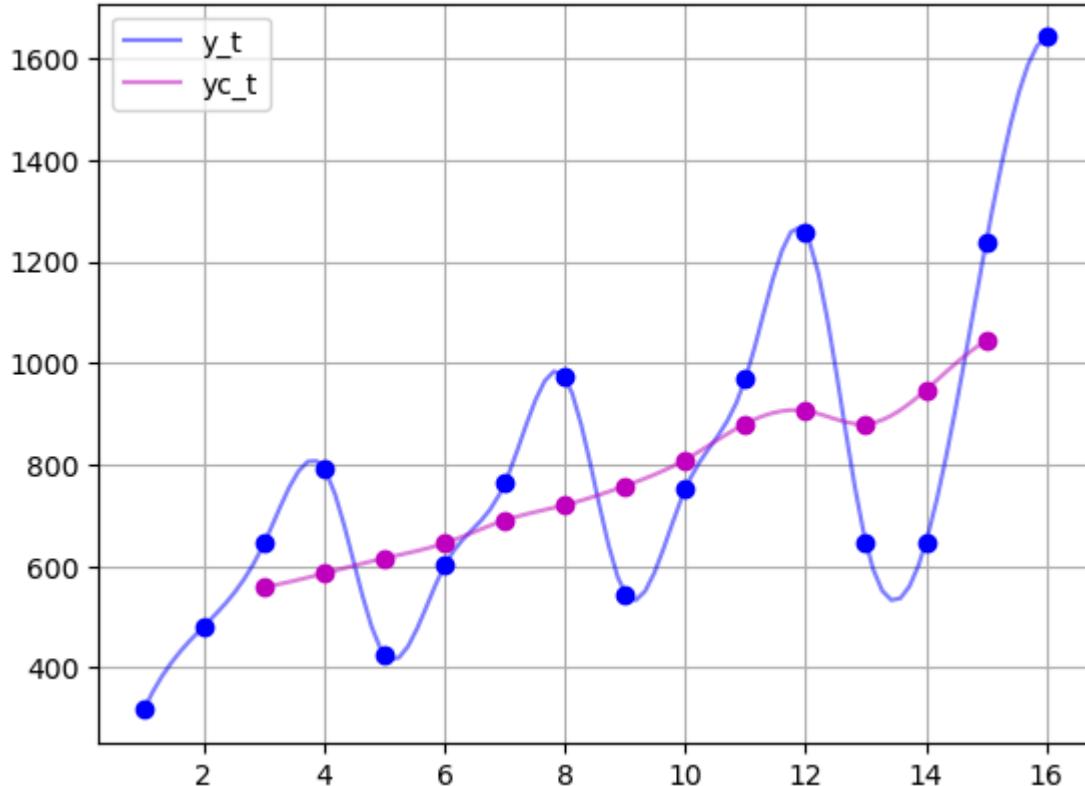
0.8944586506469873

- средняя ошибка аппроксимации

11.003038061712864

--- Мультипликативная модель 1_2 ---





Взаимопогашаемость сезонных воздействий в м



>

что сумма значений iS сезонной компоненты S д



>

то есть для данного примера – четырем. Сумми



>

0.6929620470545647

0.8132515255247674

1.0973703579519745

1.3468647725775842

3.950448703108891

Рассчитаем корректирующий коэффициент:

1.0125432072696232

Определим скорректированные значения сезон



>

умножив ее средние оценки iS на корректирую



>

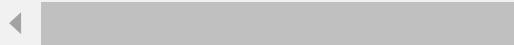
0.7016540136407524

0.8234523079717618

1.1111349018033068

1.3637587765841788

Проверим выполнение свойства взаимопогаша



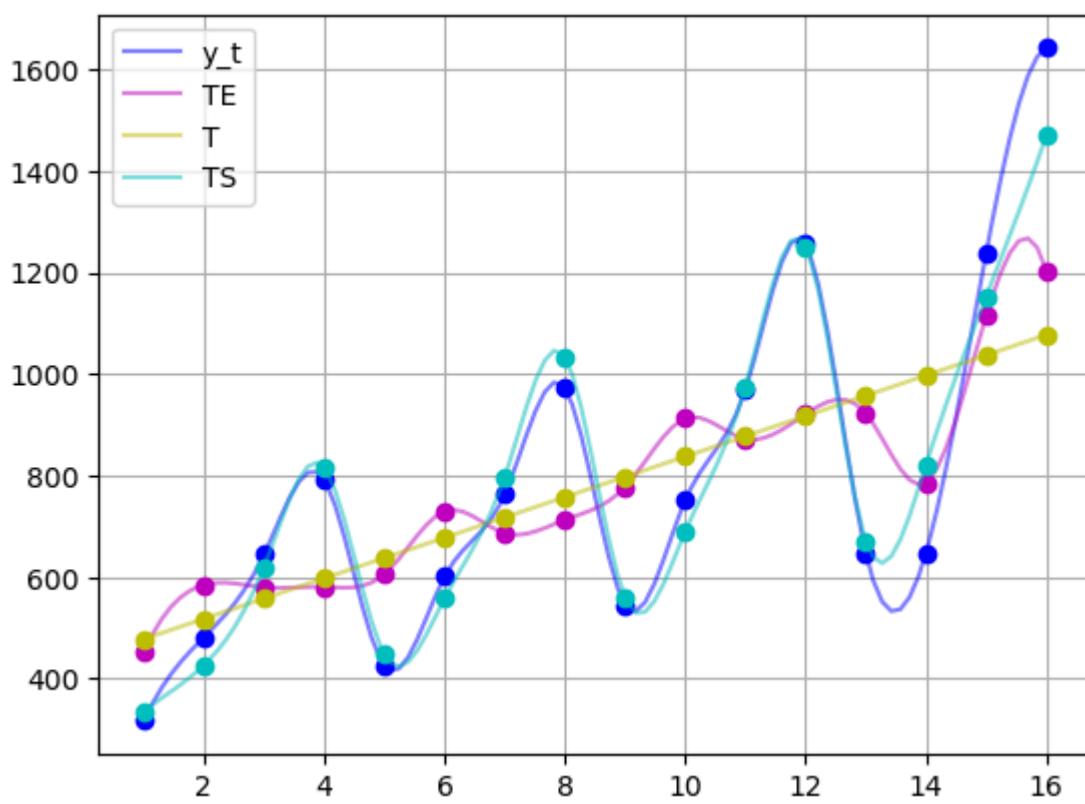
4.0

Для выполнения последующих расчетов перене



$b_0 = 438.140418558105$

$b_1 = 39.9391685664597$



^

$$y = (438.140418558105 + 39.939168566459706 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели знач



относительной ошибки E не могут быть исполь



Поэтому для оценки качества модели необходимо



абсолютной ошибки по формуле $E_{abs} = Y_t - (T \cdot S)$, т



Значения остатков в t_e представлены в графике.

^

$$S_i = [0.7016540136407524, 0.8234523079717618, 1.1111349018033068, 1.3637587765841788]$$

Предсказанные значения:

$$783.822107963761$$

$$952.7717484988261$$

$$1330.0113896713992$$

$$1686.8656744245689$$

Рассчитаем стандартную ошибку регрессии: 77.37



Стандартные ошибки прогноза будут следующими



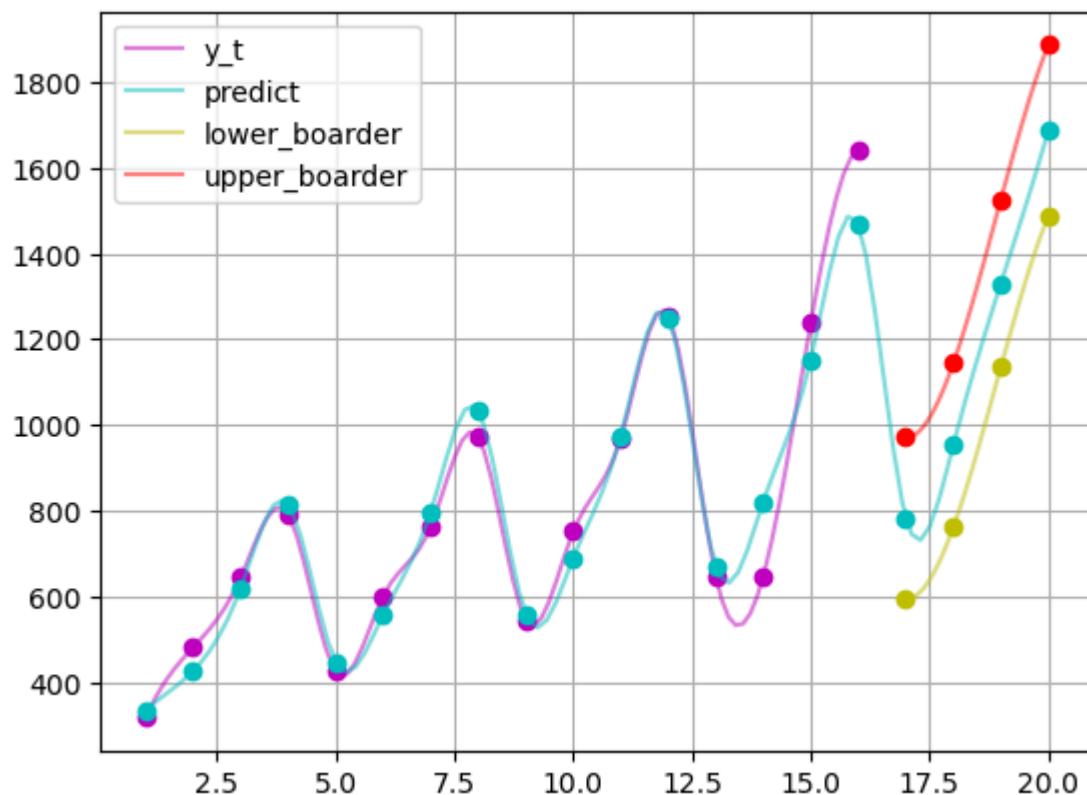
$$87.3695216685764$$

$$89.16497342258917$$

$$91.11843578406389$$

$$93.21997575330342$$

Критическое значение $t_{0.95;14} = 2.14$



- коэффициент детерминации

0.9476147837859924

- средняя ошибка аппроксимации

6.691774161844011

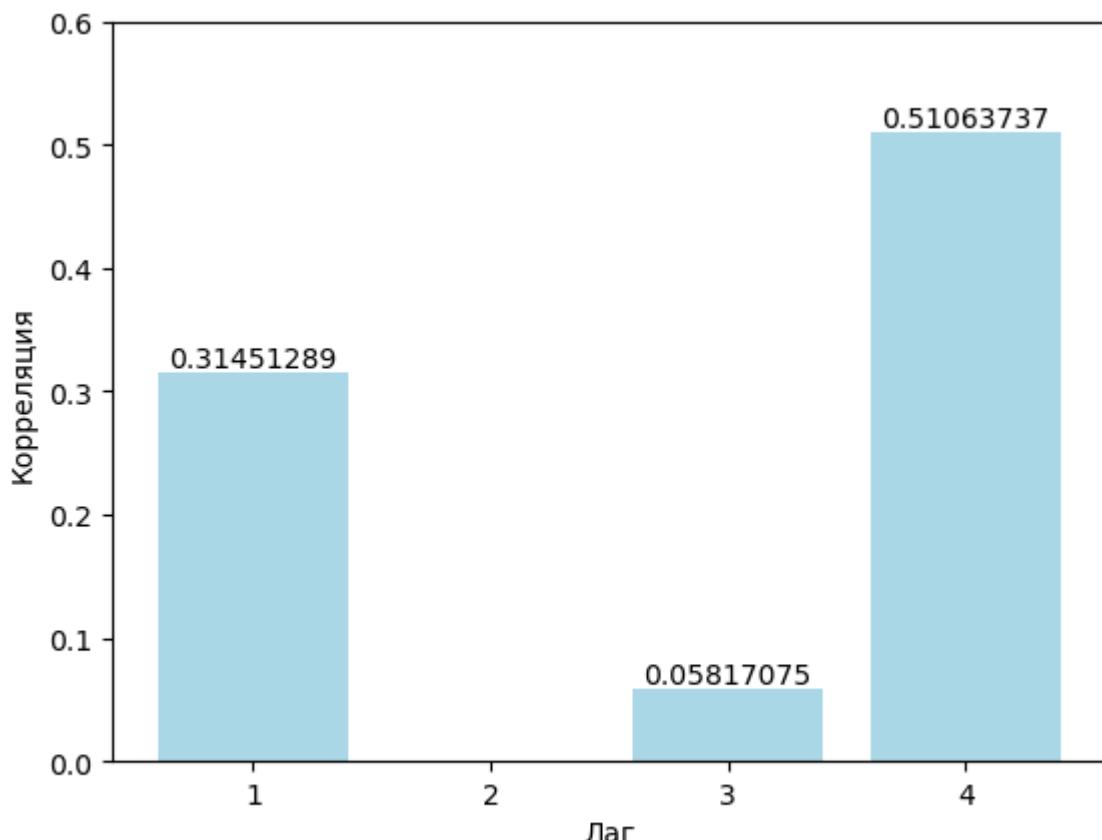
=====

Данные задания 2

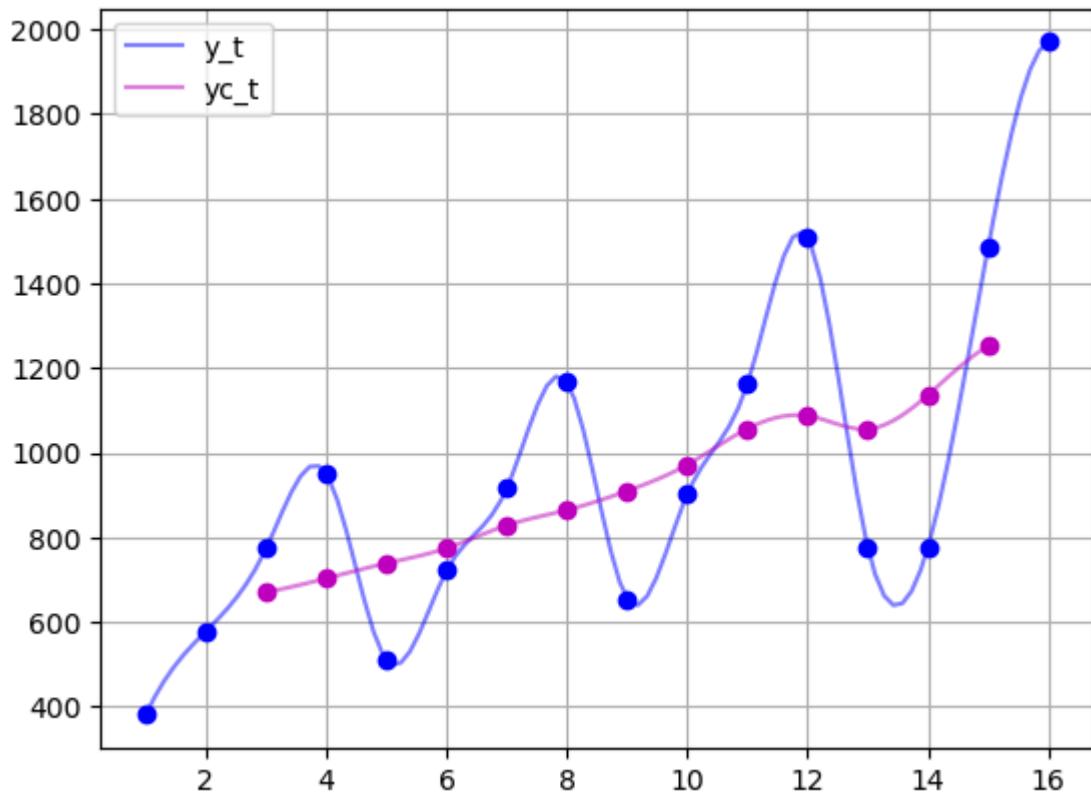
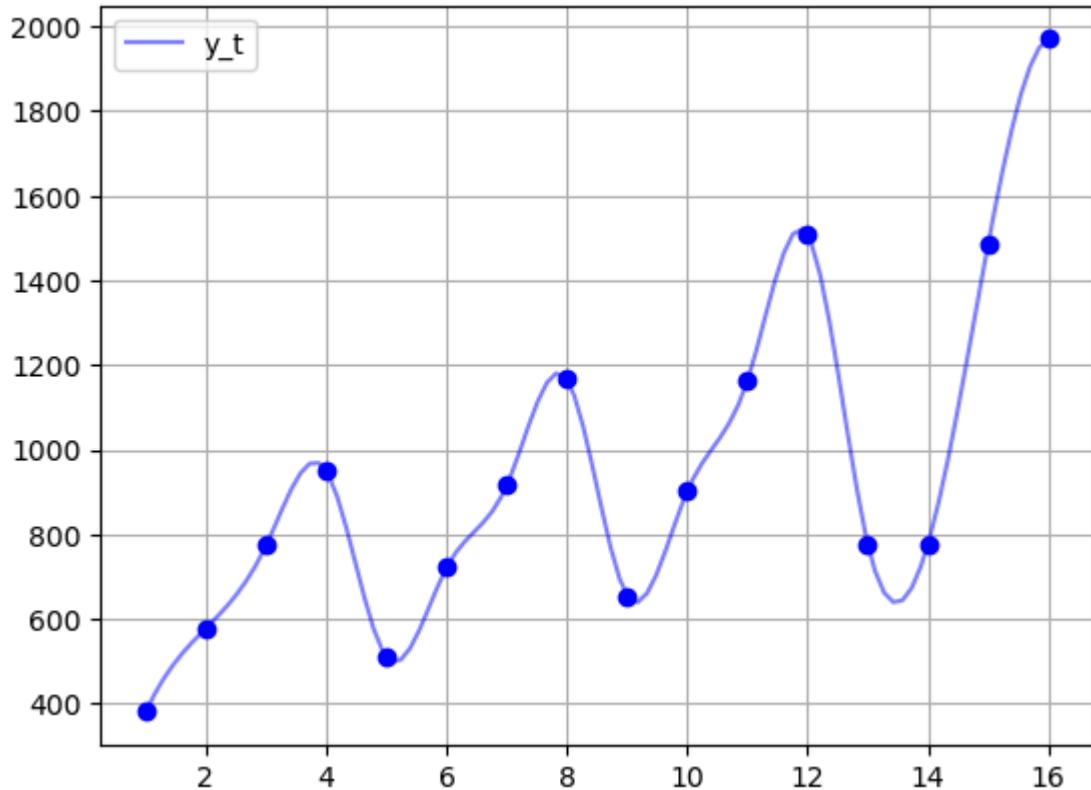
=====

--- Аддитивная модель 2_1 ---

4



-0.13277688



З а т е м р а с с ч и т а е м с р е д н и е о ц е н к и с е з о н н ы х к о л

-283.88750000000001

-203.2958333333332

82.02916666666665

315.045833333334

П р и м о д е л и р о в а н и и п е р и о д и ч е с к и х к о л е б а н и й



ч т о с е з о н н ы е в о з д е й с т в и я з а п е р и о д в з а и м о п о



ч т о с у м м а з н а ч е н и й с е з о н н о й к о м п о н е н т ы п о в



-90.1083333333335

Д л я у с т р а н е н и я д а н н о г о п р o t i v o r e c h i я с к o r r e k



с р e d n i e з n a ч e n i j o c e n o k с e z o n n o й k o m p o n e n t



В e l i c h i n a k o r r e k t i r u y u щ e g o k o e ф f i c i e n t a b u



-22.52708333333337

Т e p e r y r a s c c i t a e m s k o r r e k t i r o v a n n y e z n a c h e n



r a z n o s t ь m e j d u e e c r e d n e j o c e n k o й S i k o r r e k t



-261.36041666666677

-180.76874999999998

104.55624999999999

337.57291666666674

П r o v e r i m u s l o v i e v z a i m o p o g a s h a e m o s t i z n a c h e n



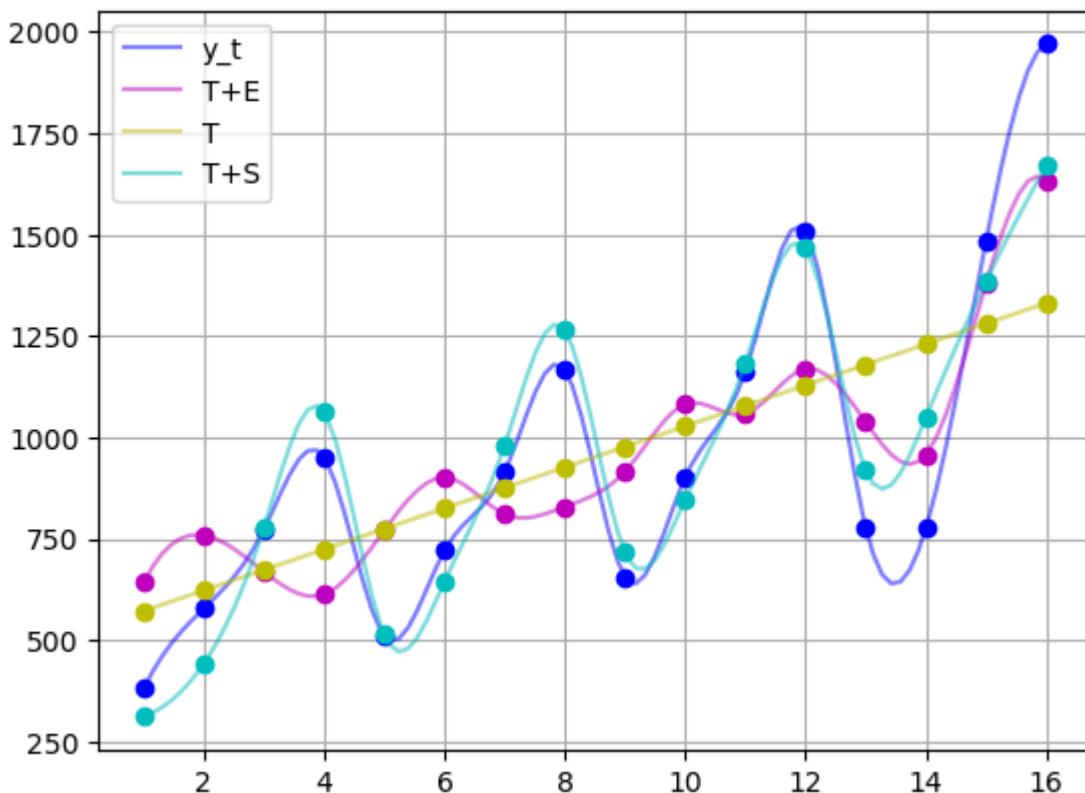
$0.0 \approx 0$

П e r e n e s e m p o l u c h e n n y e z n a c h e n i j S v t a b l i c u n a g



$b0 = 521.733750000000$

$b1 = 50.6100000000000$



$$\hat{y} = 521.73375 + 50.61 \times t + S_i + e_t$$

$$S_i = [-261.36041666666677, -180.76874999999998, 104.55624999999999, 337.57291666666667]$$

Предсказанные значения:

1120.743333333333

1251.945

1587.88

1871.506666666667

Рассчитаем стандартную ошибку регрессии: 136.0

Стандартные ошибки прогноза будут следующими

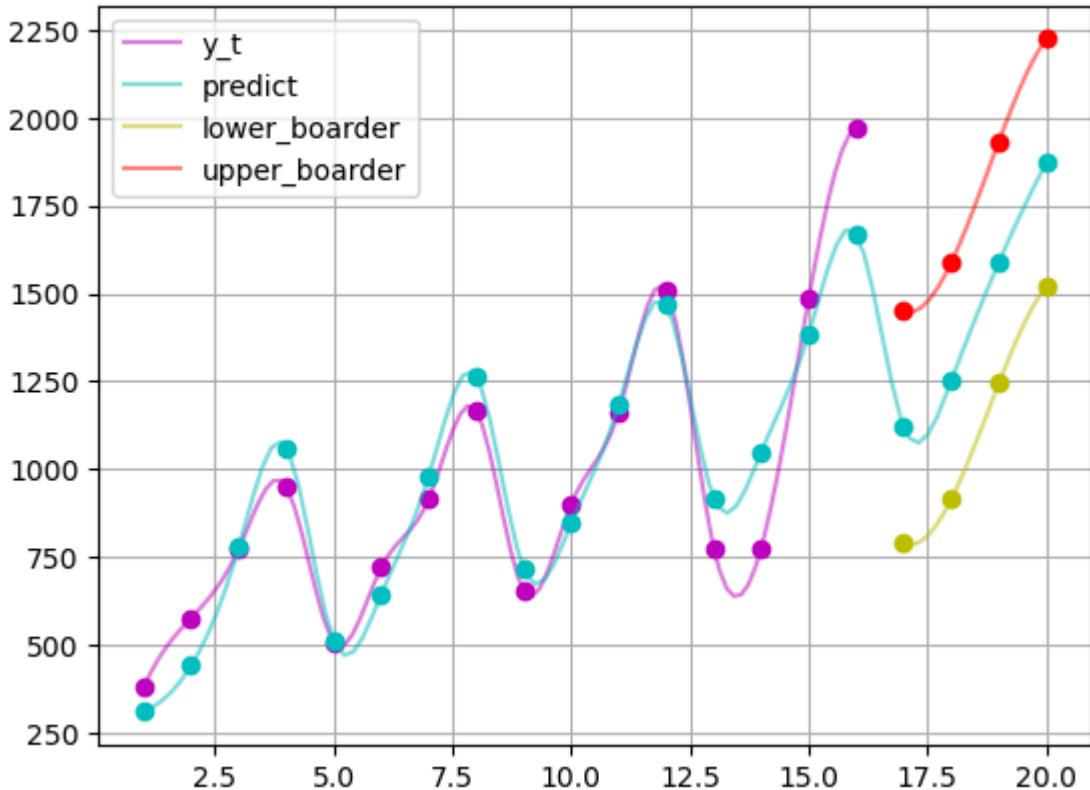
153.6491760695293

156.80668085389937

160.2420651457122

163.93786064263284

Критическое значение $t_{0.95;14} = 2.14$



– коэффициент детерминации

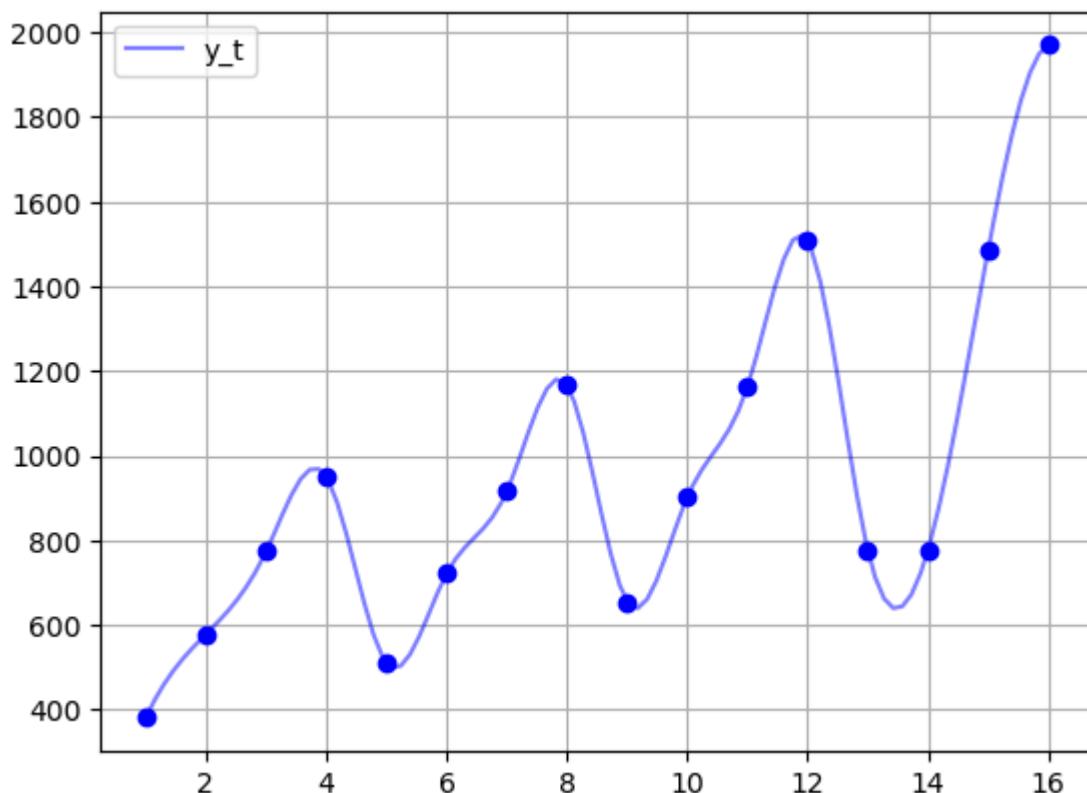
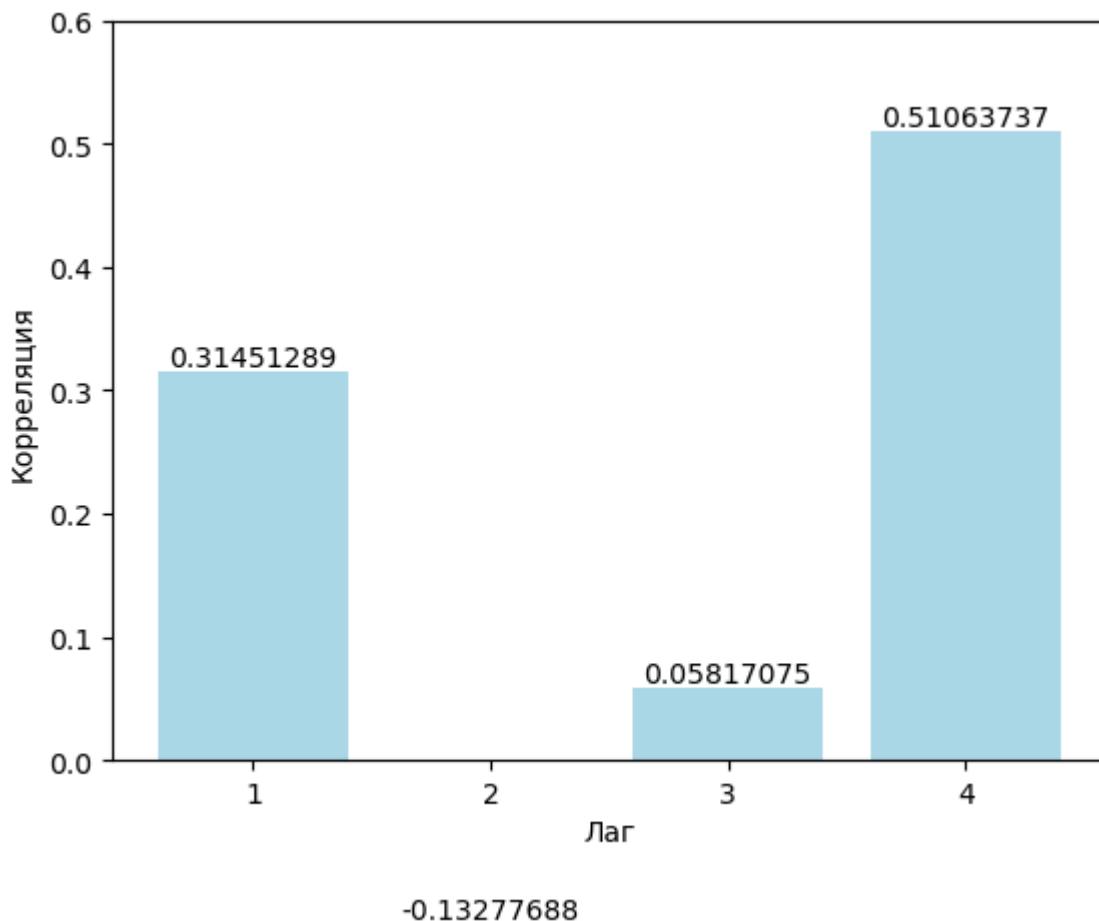
0.8944130669514643

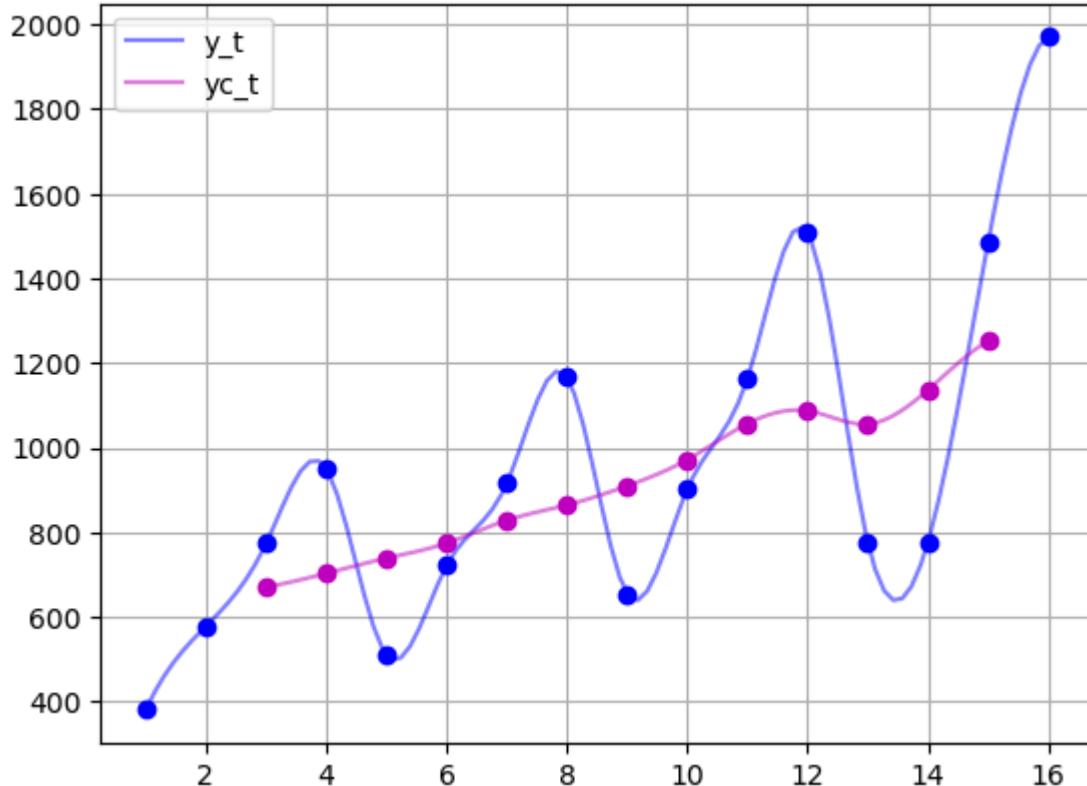
– средняя ошибка аппроксимации

11.004838114522919

--- Мультипликативная модель 2_2 ---

4





Взаимопогашаемость сезонных воздействий в м



что сумма значений iS сезонной компоненты S д



то есть для данного примера – четырем. Сумми



0.6929782454562253

0.8132451821264898

1.0973204524846973

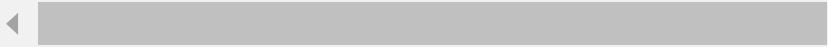
1.3468792319391794

3.950423112006592

Рассчитаем корректирующий коэффициент:

1.0125497665915149

Определим скорректированные значения сезон



умножив ее средние оценки iS на корректирую



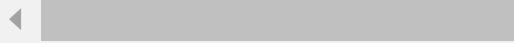
0.7016749606896985

0.8234512193438512

1.1110915680394757

1.3637822519269749

Проверим выполнение свойства взаимопогаша



▶

4.0

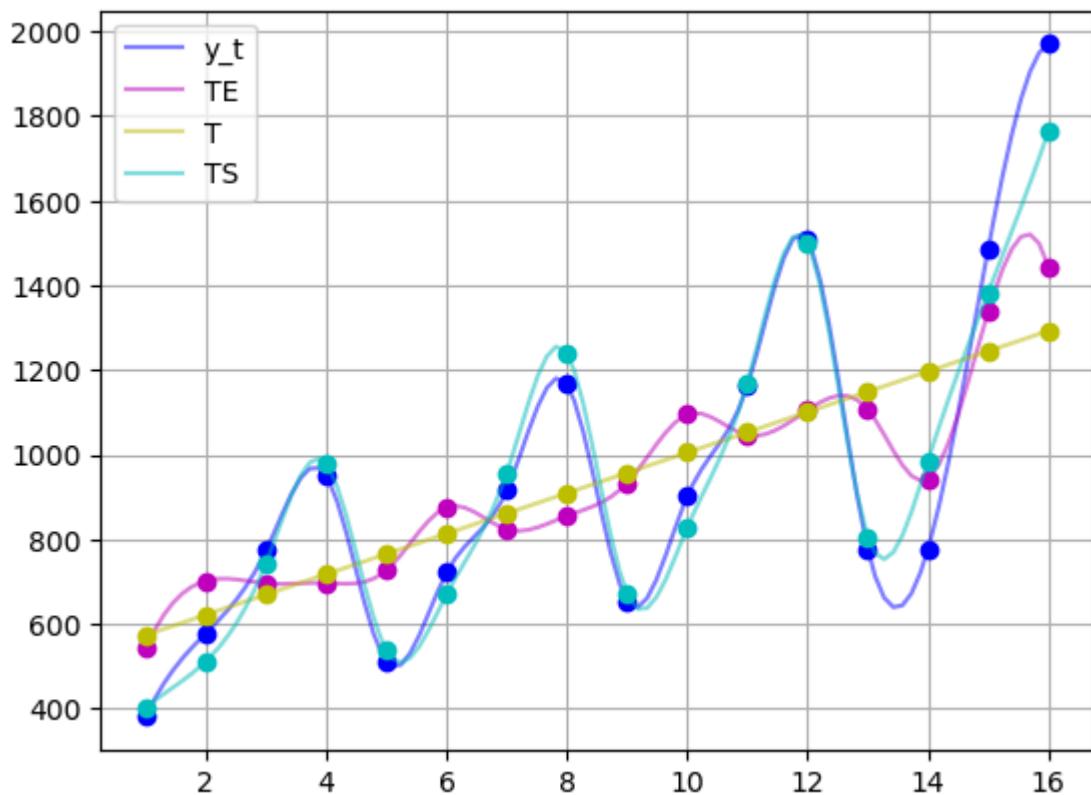
Для выполнения последующих расчетов перене



▶

$b_0 = 525.834701072655$

$b_1 = 47.9199031902259$



^

$$y = (525.834701072655 + 47.91990319022588 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели знач



▶

относительной ошибки E не могут быть исполь



▶

Поэтому для оценки качества модели необходимо



▶

абсолютной ошибки по формуле $E_{abs} = Y_t - (T \cdot S)$, т



▶

Значения остатков в t е предstawлены в графике 1.

^

$$S_i = [0.7016749606896985, 0.8234512193438512, 1.1110915680394757, 1.3637822519269749]$$

Предсказанные значения:

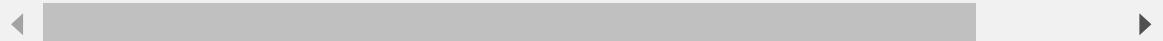
940.576378387785

1143.2738746025418

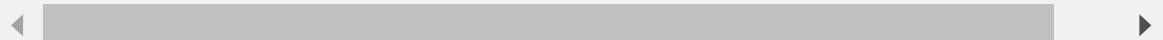
1595.8751096870162

2024.1703024679905

Рассчитаем стандартную ошибку регрессии: 92.88



Стандартные ошибки прогноза будут следующими



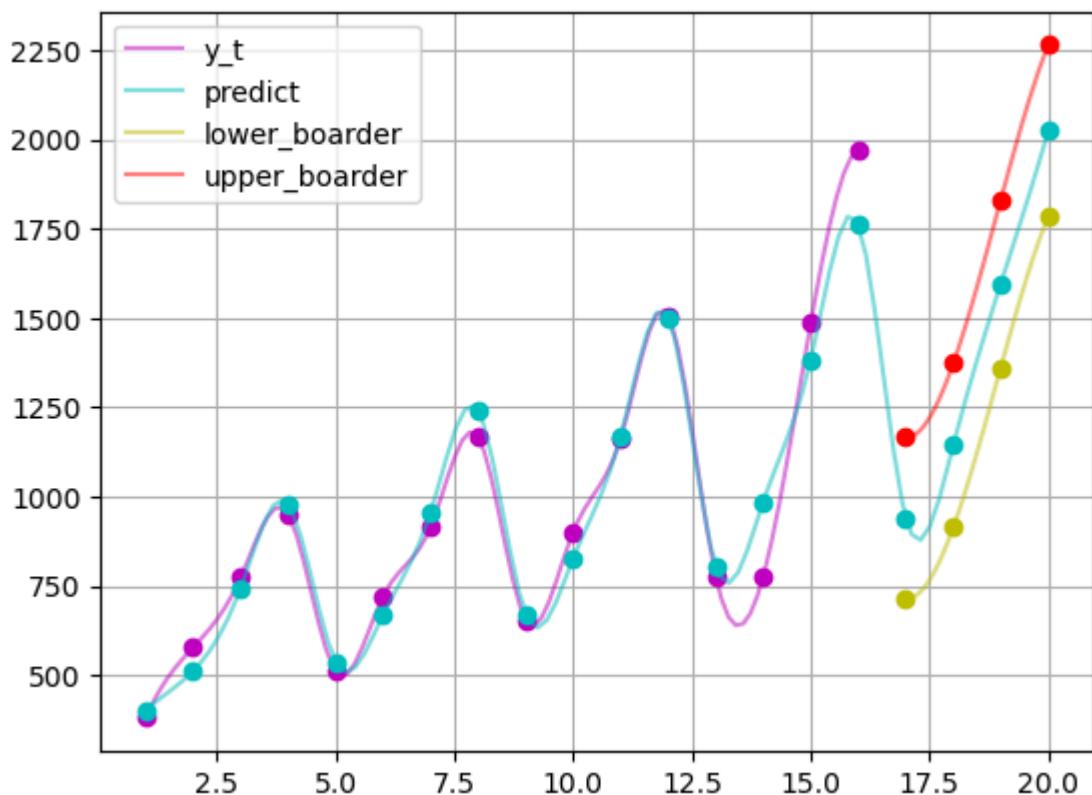
104.87768499760362

107.03293112794238

109.3778519457372

111.90052395646251

Критическое значение $t_{0.95;14} = 2.14$



- коэффициент детерминации

0.9475754363313096

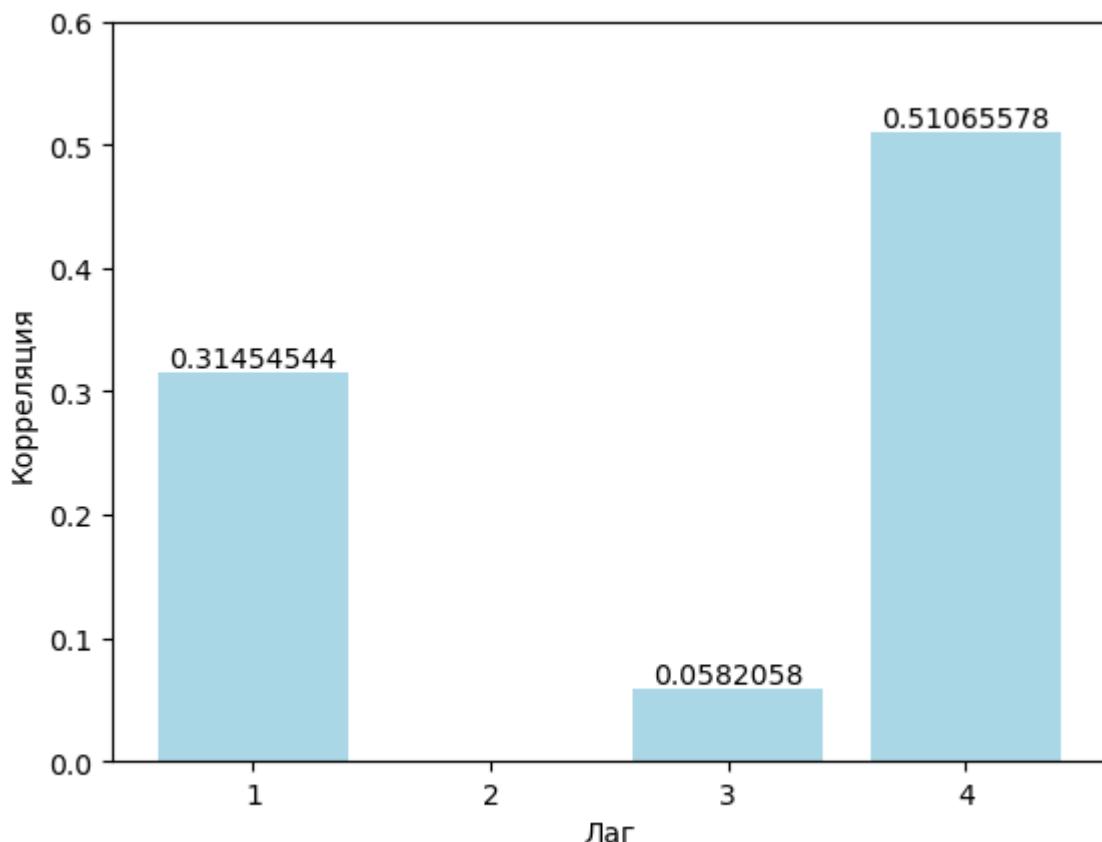
— средняя ошибка аппроксимации

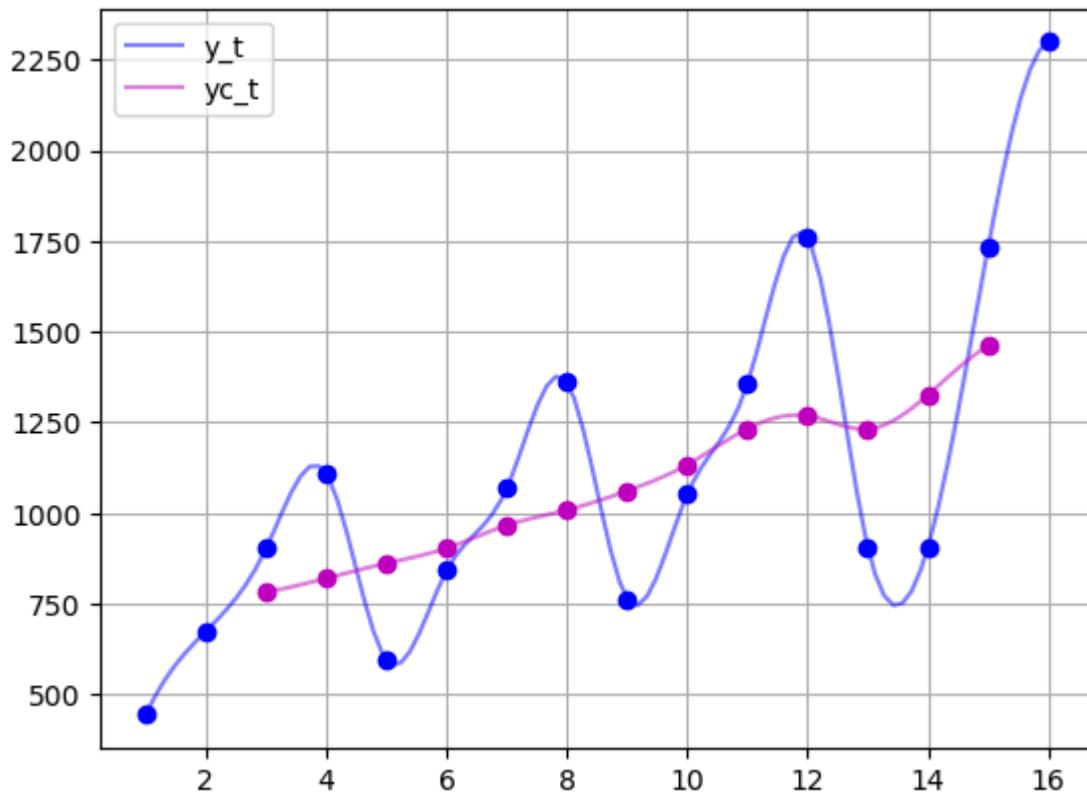
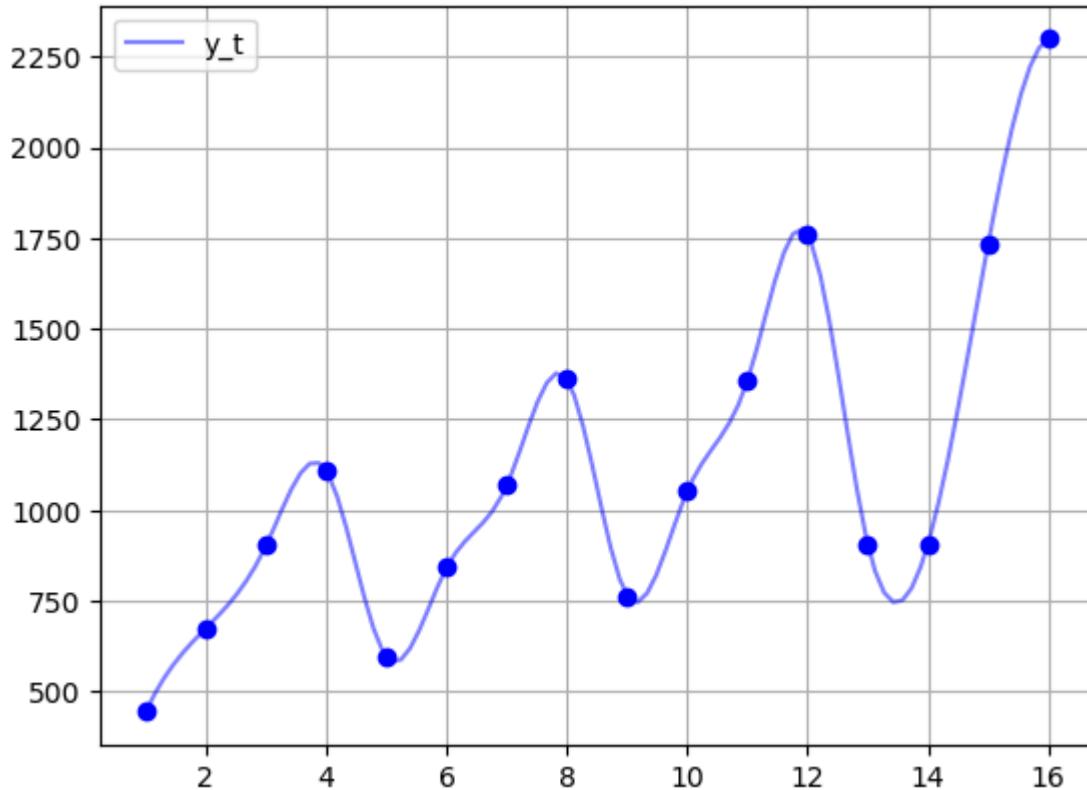
6.693568798025706

Данные задания 3

--- Аддитивная модель З 1 ---

4





З а т е м р а с с ч и т а е м с р е д н и е о ц е н к и с е з о н н ы х к оэ

◀ ▶

-331.2083333333333

-237.17500000000004

95.7083333333333

367.5541666666668

П р и м о д е л и р о в а н и и п е р и о д и ч е с к и х к о л е б а н и й



ч т о с е з о н н ы е в о з д е й с т в и я з а п е р и о д в з а и м о п о



ч т о с у м м а з н а ч е н и й с е з о н н о й к о м п о н е н т ы п о в



-105.1208333333322

Д л я у с т р а н е н и я д а н н о г о п р o t i v o r e c h i я с к o r r e k



с р e d n i e з n a ч e n i j o c e n o k с e z o n n o й k o m p o n e n t



В e l i c h i n a k o r r e k t i r u y u щ e g o k o e ф i c i e n t a b u



-26.280208333333306

Т e p e r y r a s c c i t a e m s k o r r e k t i r o v a n n y e z n a c h e n



r a z n o s t ь m e j d u e e c r e d n e j o c e n k o й S i k o r r e k t



-304.928125

-210.8947916666672

121.98854166666663

393.8343750000001

П r o v e r i m u s l o v i e v z a i m o p o g a s h a e m o s t i z n a c h e n



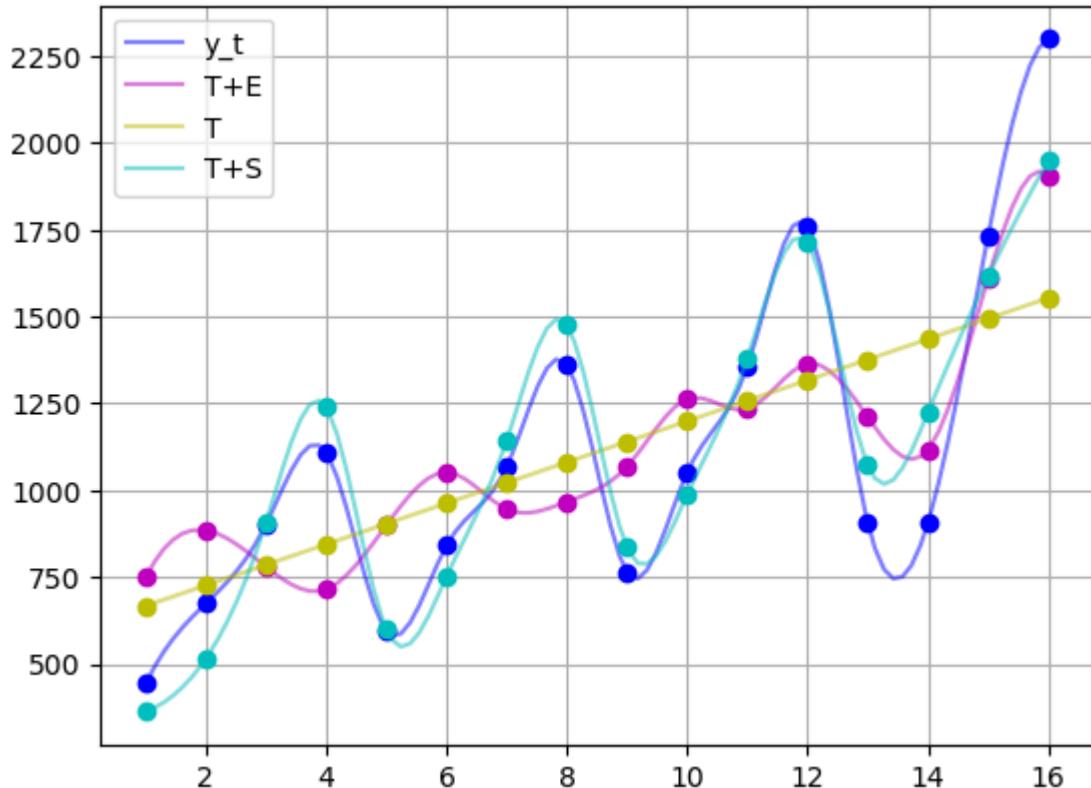
$-5.684341886080802e - 14 \approx 0$

П e r e n e s e m p o l u c h e n n y e z n a c h e n i j S v t a b l i c u n a g



$b0 = 608.663541666675$

$b1 = 59.0476715686265$



$$\hat{y} = 608.663541666675 + 59.04767156862647 \times t + S_i + e_t$$

$$S_i = [-304.928125, -210.8947916666672, 121.98854166666663, 393.8343750000001]$$

Предсказанные значения:

1307.545833333325

1460.6268382352846

1852.5578431372446

2183.4513480392043

Рассчитаем стандартную ошибку регрессии: 158.7



Стандартные ошибки прогноза будут следующими



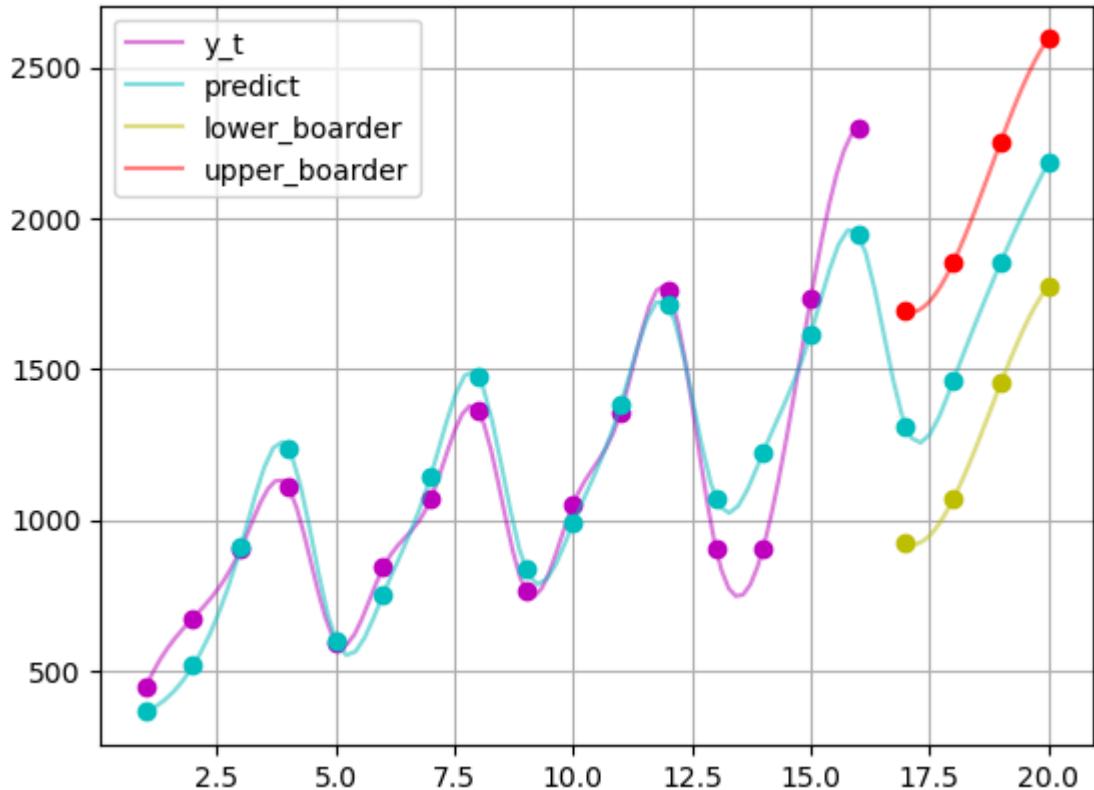
179.22377720613468

182.90684241011056

186.91403961535102

191.22499295518412

Критическое значение $t_{0.95;14} = 2.14$



– коэффициент детерминации

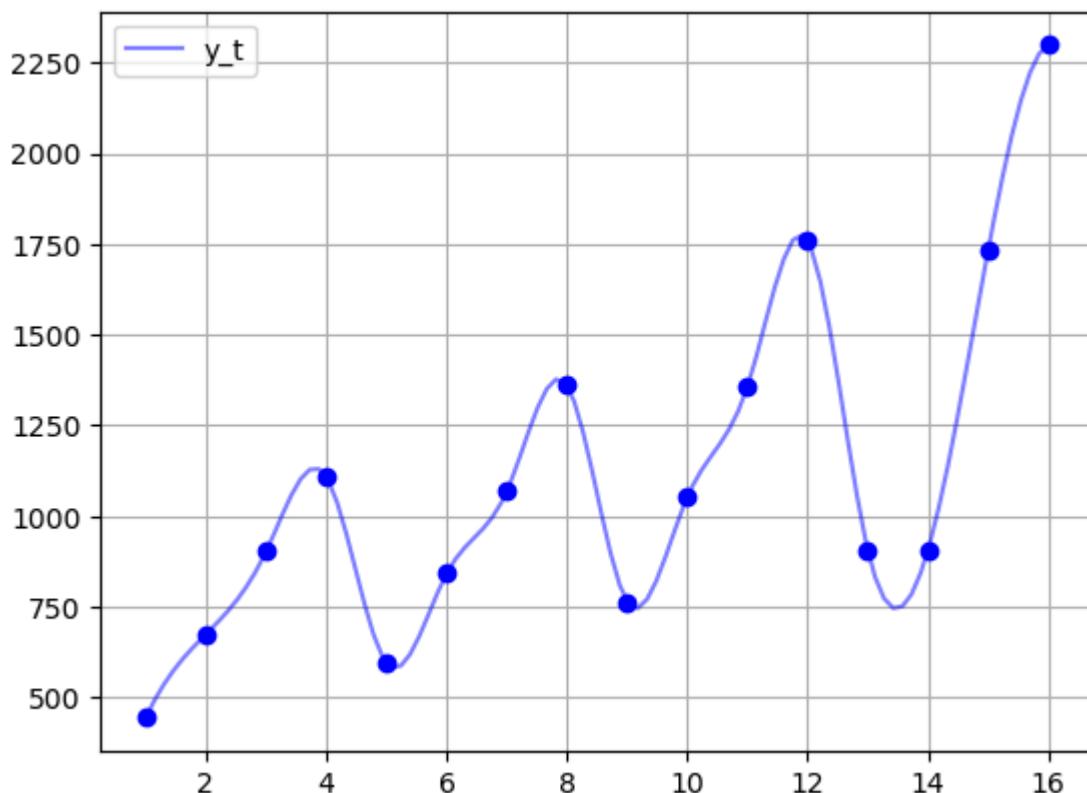
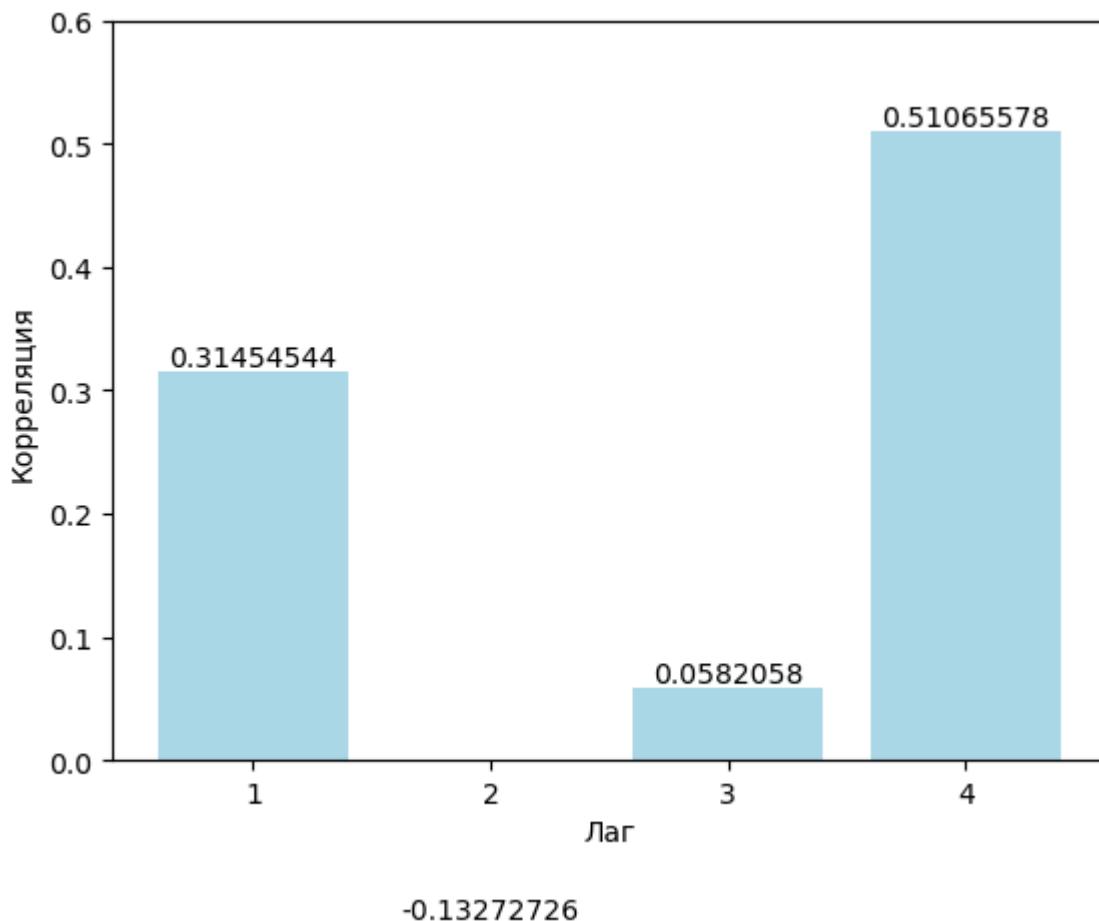
0.8944534584179158

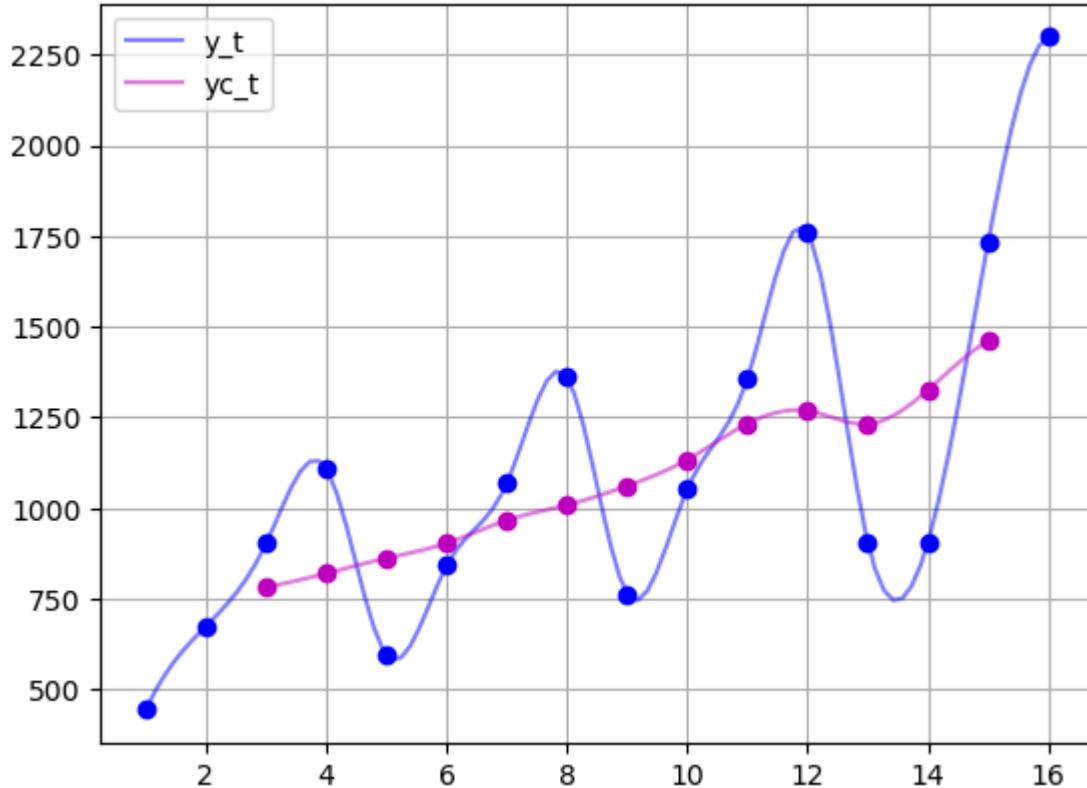
– средняя ошибка аппроксимации

11.003612497419681

--- Мультипликативная модель 3_2 ---

4





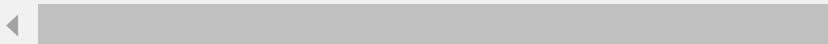
Взаимопогашаемость сезонных воздействий в м



что сумма значений iS сезонной компоненты S д



то есть для данного примера – четырем. Сумми



0.6929750386191768

0.8132475843796773

1.0973236861199558

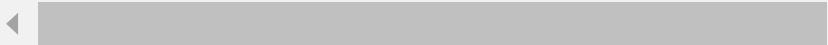
1.346880763454618

3.9504270725734276

Рассчитаем корректирующий коэффициент:

1.0125487514427849

Определим скорректированные значения сезон



умножив ее средние оценки iS на корректирую



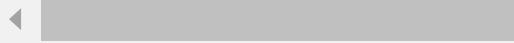
0.7016710101348631

0.823452826177503

1.1110937283093556

1.3637824353782784

Проверим выполнение свойства взаимопогаша



▶

4.0

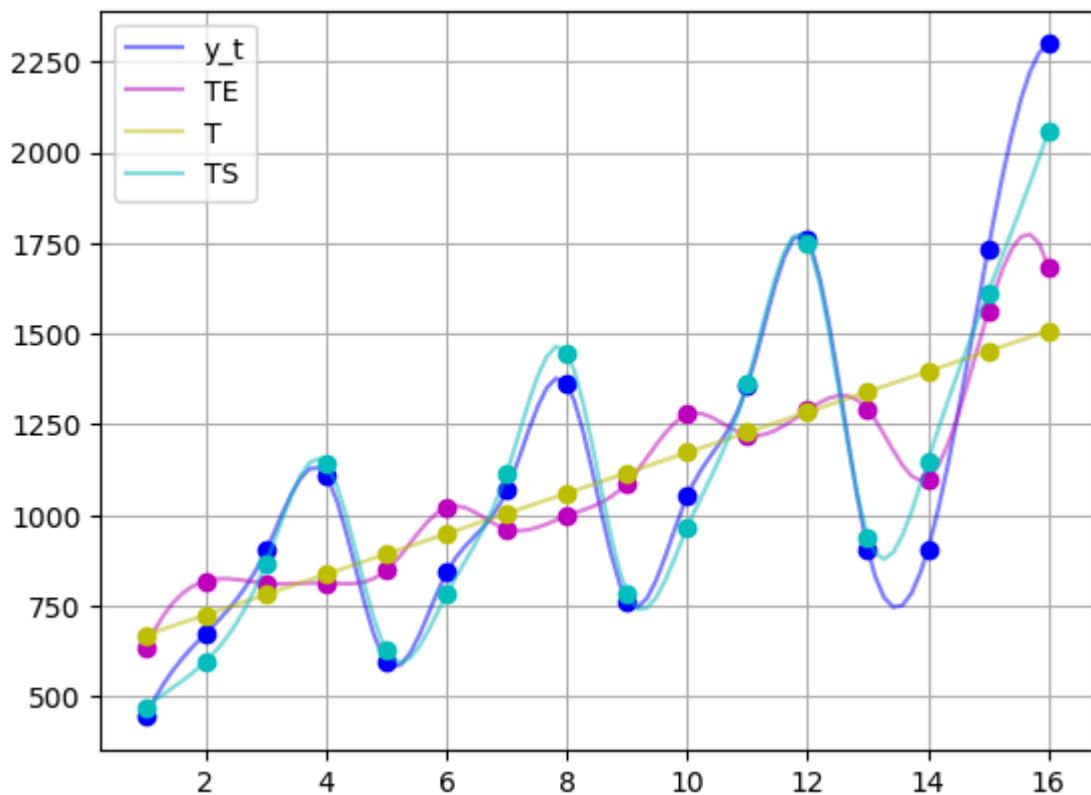
Для выполнения последующих расчетов перене



▶

$b_0 = 613.438795386443$

$b_1 = 55.9104285285413$



^

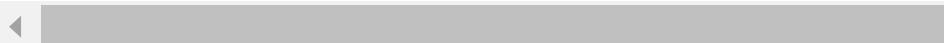
$$y = (613.4387953864425 + 55.910428528541324 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели знач



▶

относительной ошибки E не могут быть исполь



▶

Поэтому для оценки качества модели необходимо



▶

абсолютной ошибки по формуле $E_{abs} = Y_t - (T \cdot S)$, т



▶

Значения остатков в t_e представлены в графике.

^

$$S_i = [0.7016710101348631, 0.823452826177503, 1.1110937283093556, 1.3637824353782784]$$

Предсказанные значения:

1097.3545758805278

1333.8507166710967

1861.9008014733865

2361.590261961585

Рассчитаем стандартную ошибку регрессии: 108.3



Стандартные ошибки прогноза будут следующими



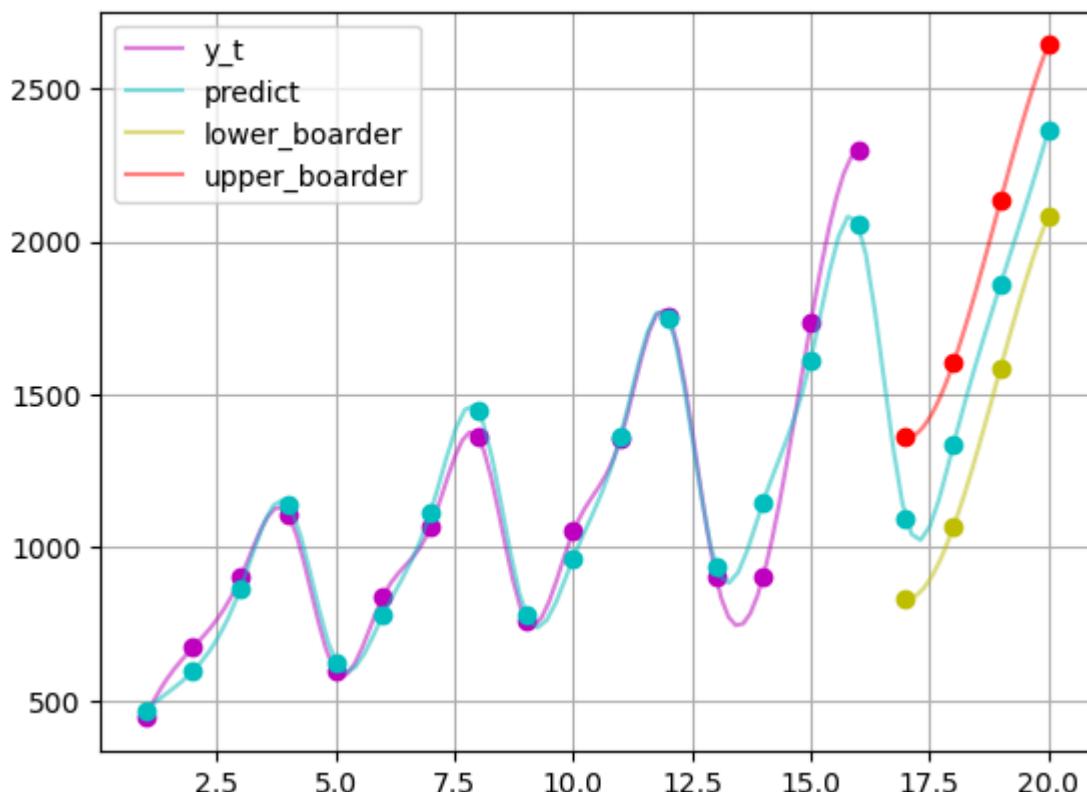
122.31850091941277

124.83215743062796

127.56703090932784

130.50921502285334

Критическое значение $t_{0.95;14} = 2.14$



- коэффициент детерминации

0.9476103573282315

- средняя ошибка аппроксимации

6.69153226716468

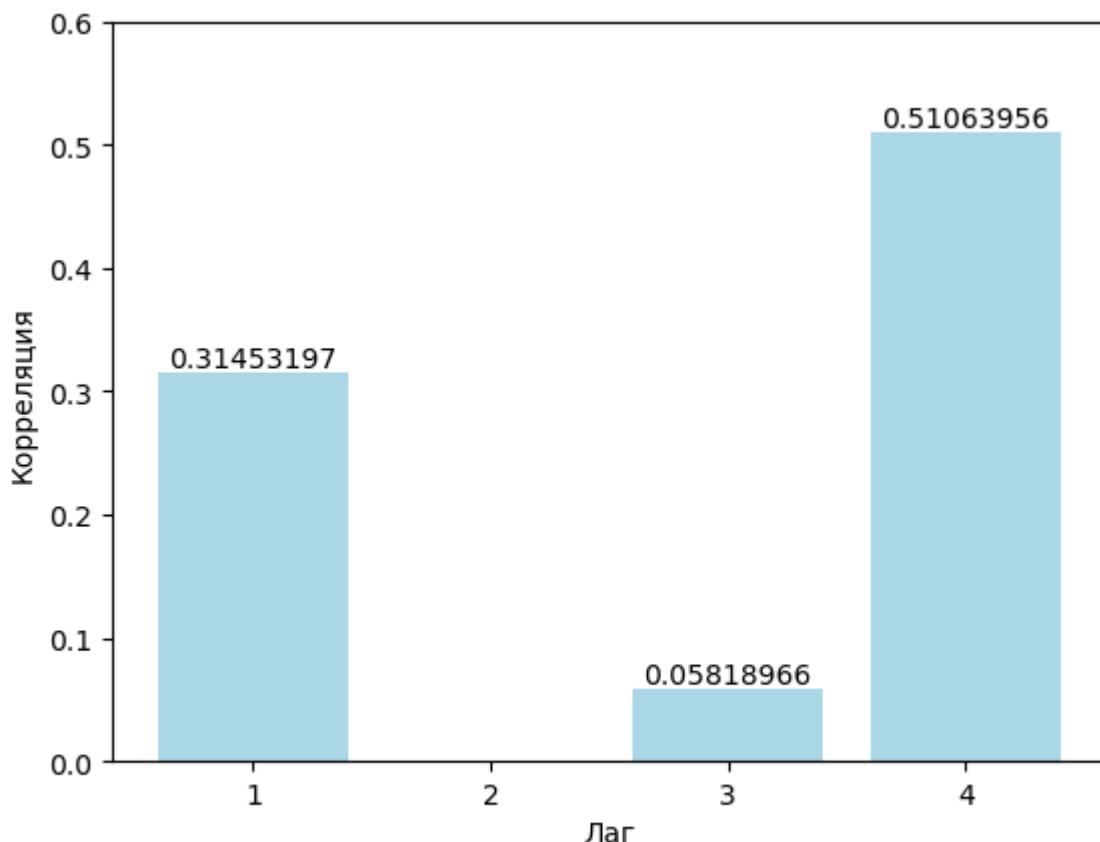
=====

Данные задания 4

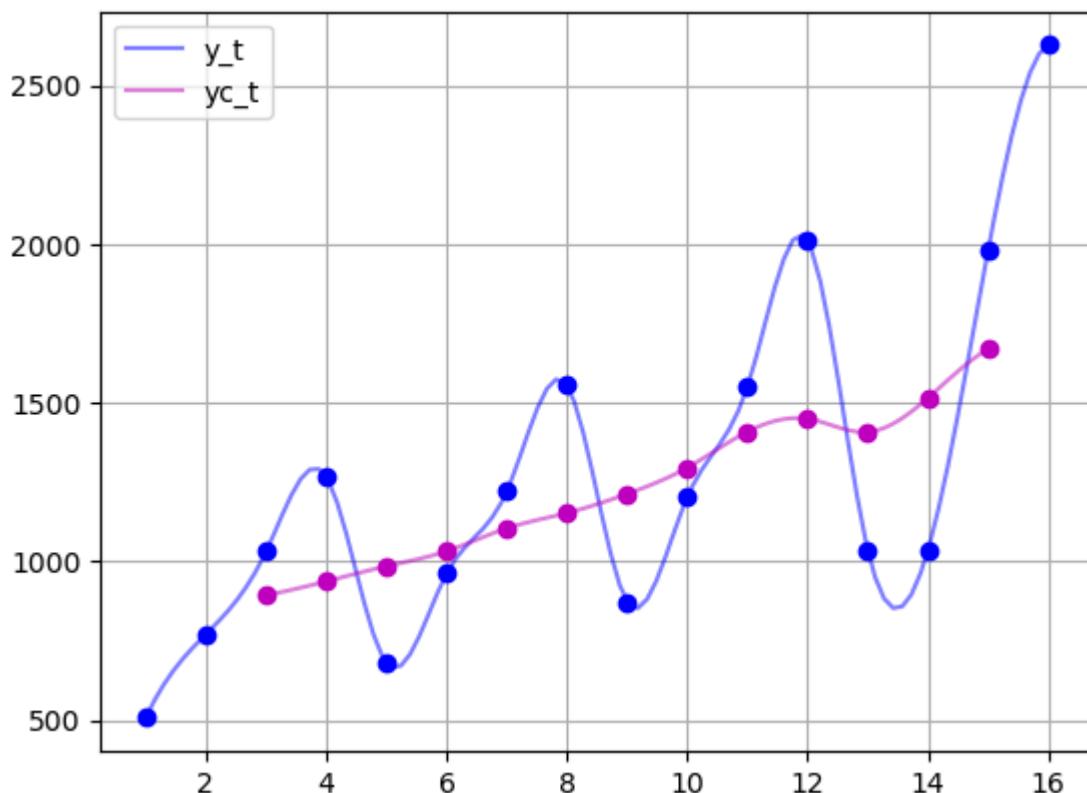
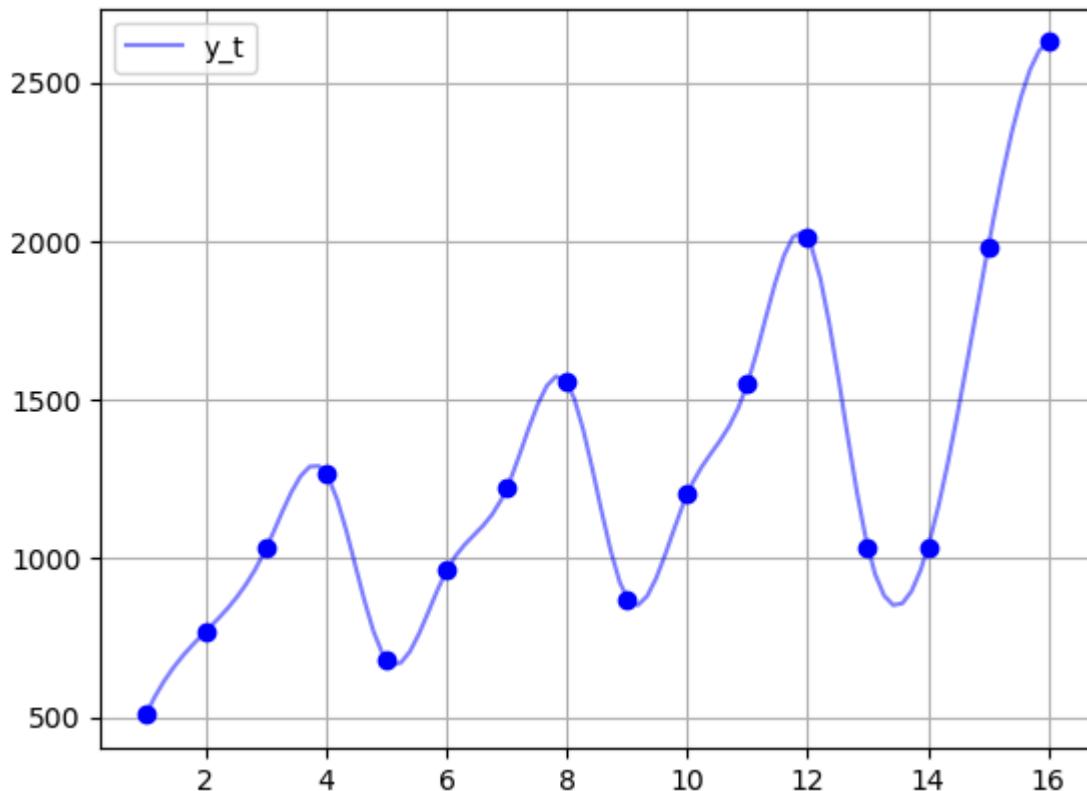
=====

--- Аддитивная модель 4_1 ---

4



-0.13274484



З а т е м р а с с ч и т а е м с р е д н и е о ц е н к и с е з о н н ы х к о ј

◀ ▶

-378.5249999999999

-271.05

109.375

420.0583333333334

П р и м о д е л и р о в а н и и п е р и о д и ч е с к и х к о л е б а н и й



ч т о с е з о н н ы е в о з д е й с т в и я з а п е р и о д в з а и м о п о



ч т о с у м м а з н а ч е н и й с е з о н н о й к о м п о н е н т ы п о в



-120.1416666666666

Д л я у с т р а н е н и я д а н н о г о п р o t i v o r e c h i я с к o r r e k



с р e d n i e з n a ч e n i j o c e n o k с e z o n n o й k o m p o n e n t



В e l i c h i n a k o r r e k t i r u y u щ e g o k o e ф i c i e n t a b u



-30.03541666666665

Т e p e r y r a s c c i t a e m s k o r r e k t i r o v a n n y e z n a c h e n



r a z n o s t ь m e j d u e e c r e d n e j o c e n k o й S i k o r r e k t



-348.4895833333326

-241.0145833333335

139.4104166666666

450.09375

П r o v e r i m u s l o v i e v z a i m o p o g a s h a e m o s t i z n a c h e n



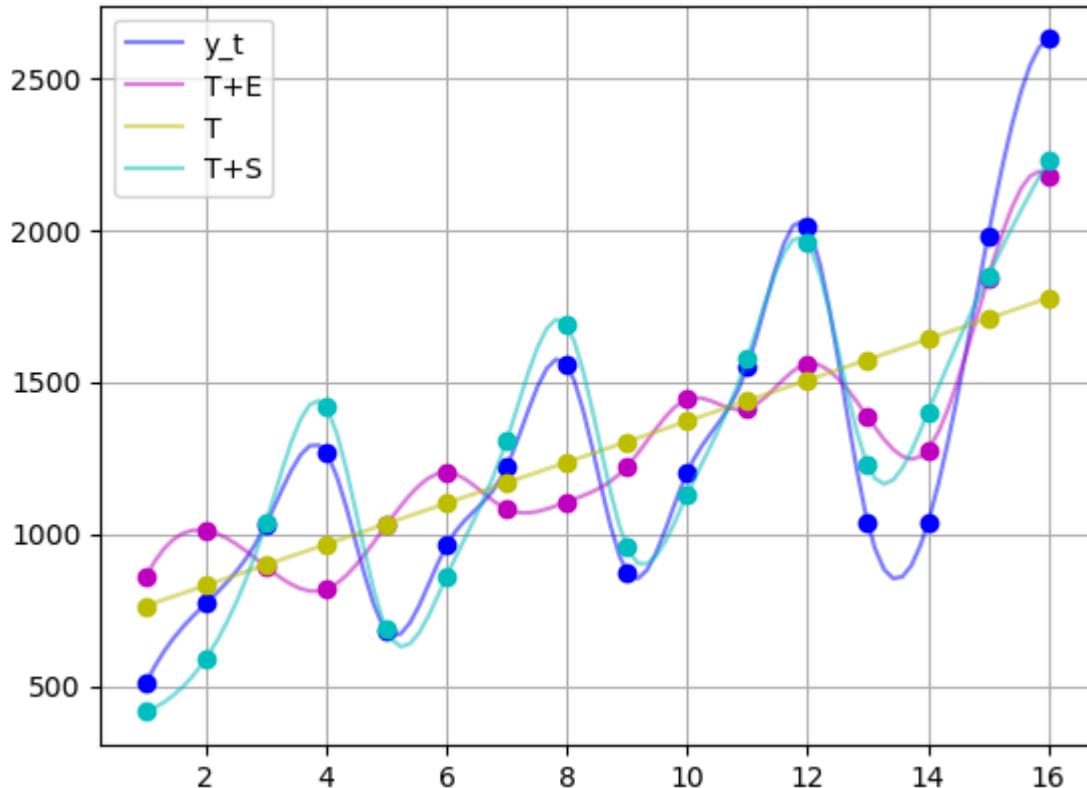
$5.684341886080802e - 14 \approx 0$

П e r e n e s e m p o l u c h e n n y e z n a c h e n i j S v t a b l i c u n a g



$b0 = 695.593750000000$

$b1 = 67.4830882352941$



$$\hat{y} = 695.59375 + 67.48308823529412 \times t + S_i + e_t$$

^

$$S_i = [-348.4895833333326, -241.0145833333335, 139.41041666666666, 450.09375]$$

Предсказанные значения:

1494.3166666666668

1669.2747549019607

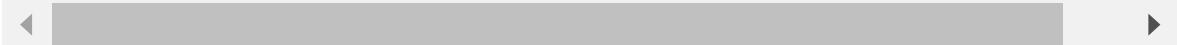
2117.182843137255

2495.3492647058824

Рассчитаем стандартную ошибку регрессии: 181.4



Стандартные ошибки прогноза будут следующими



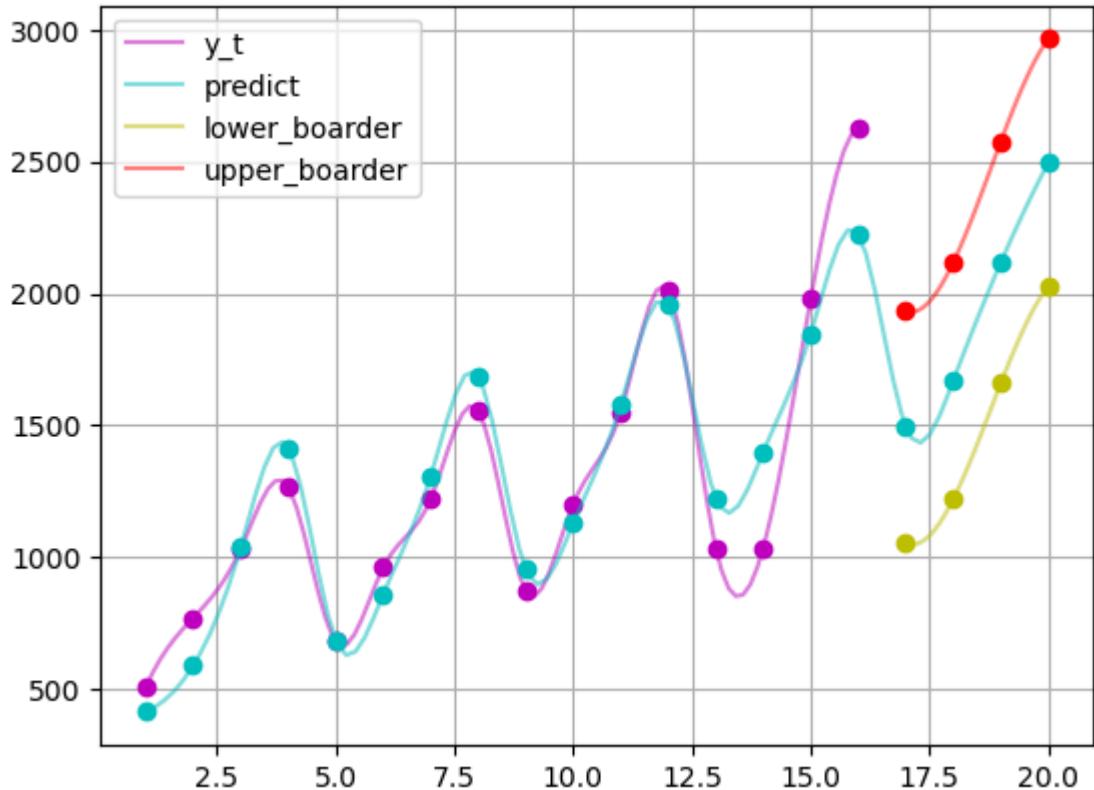
204.85211275563262

209.06184262637092

213.6420541726204

218.56945785433194

Критическое значение $t_{0.95;14} = 2.14$



– коэффициент детерминации

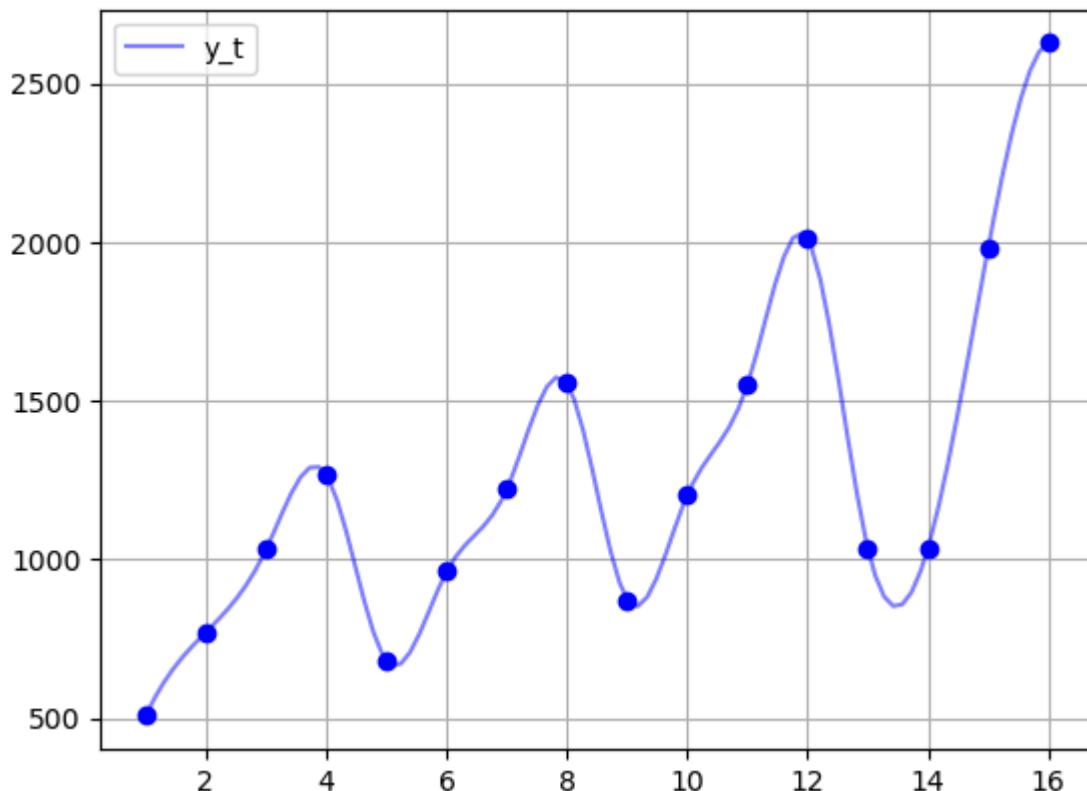
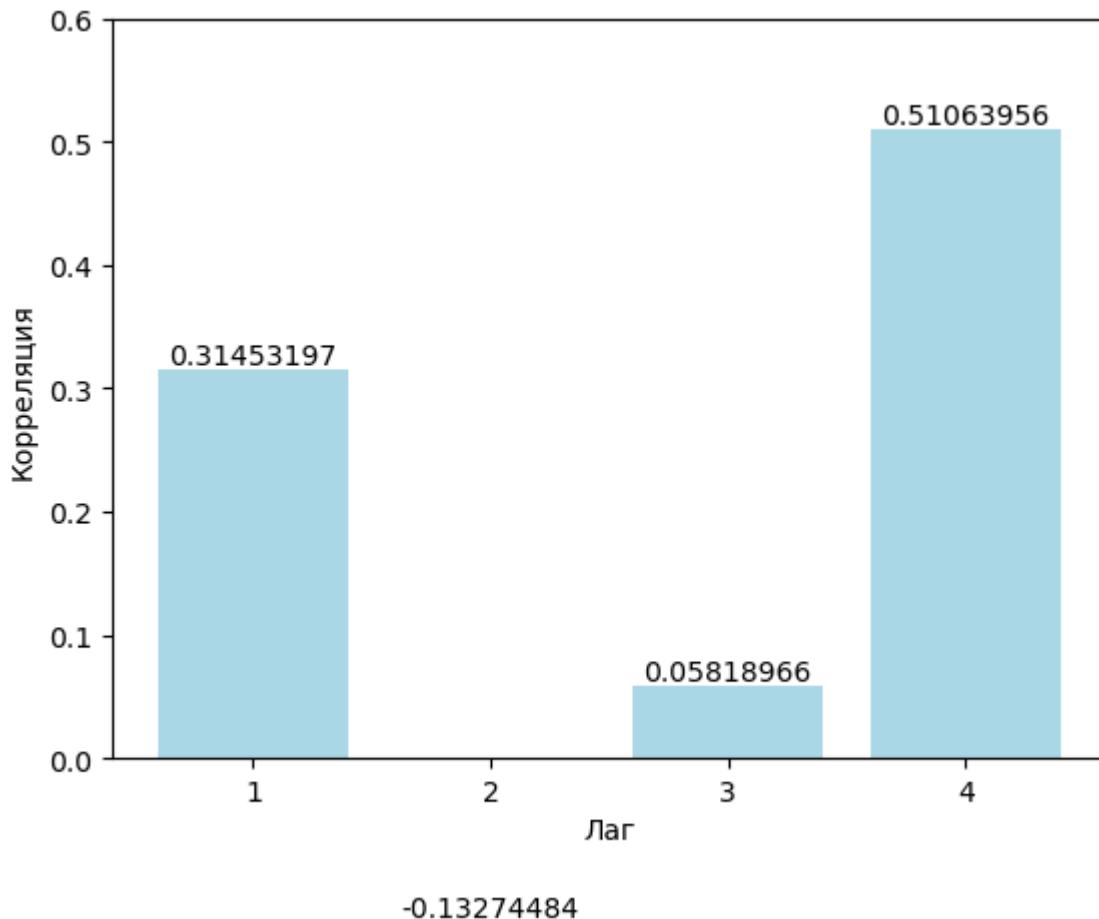
0.8944296603062589

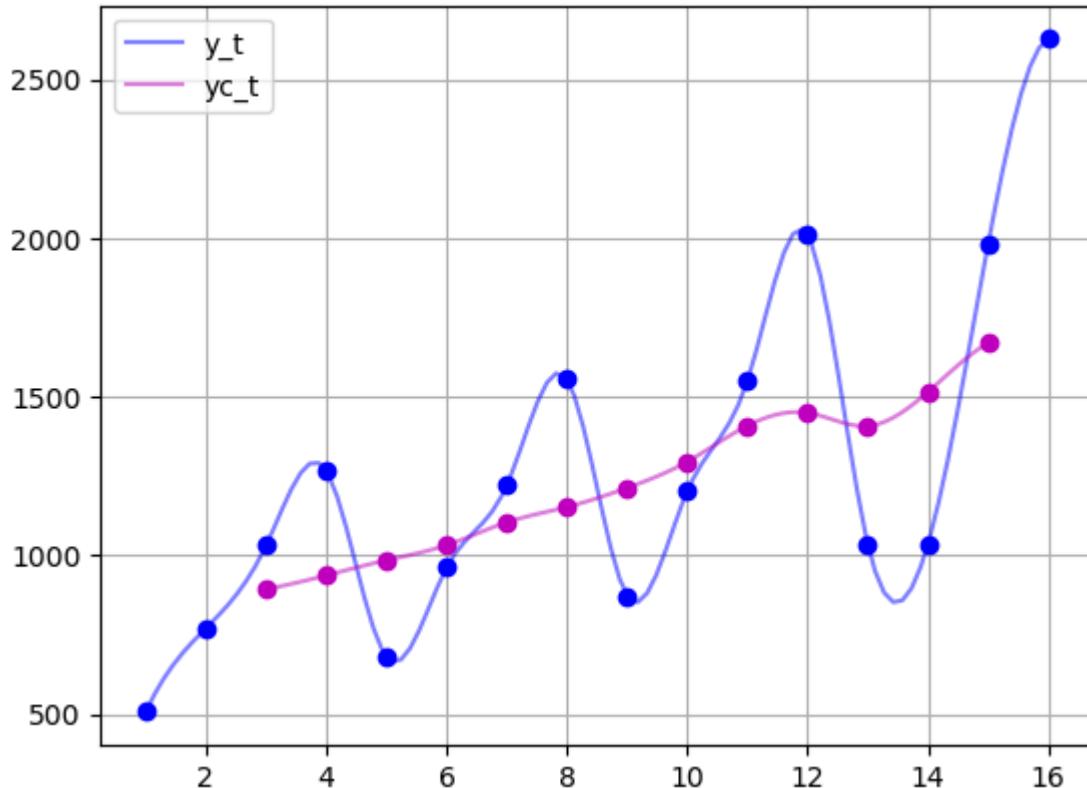
– средняя ошибка аппроксимации

11.00464809476734

--- Мультипликативная модель 4_2 ---

4





Взаимопогашаемость сезонных воздействий в м



что сумма значений iS сезонной компоненты S д



то есть для данного примера – четырем. Сумми



0.6929691787470257

0.8132523111557814

1.0973248633992776

1.3468813444045722

3.9504276977066572

Рассчитаем корректирующий коэффициент:

1.012548591212572

Определим скорректированные значения сезон



умножив ее средние оценки iS на корректирую



0.7016649656940338

0.8234574819611548

1.1110947445374666

1.3637828078073446

Проверим выполнение свойства взаимопогаша



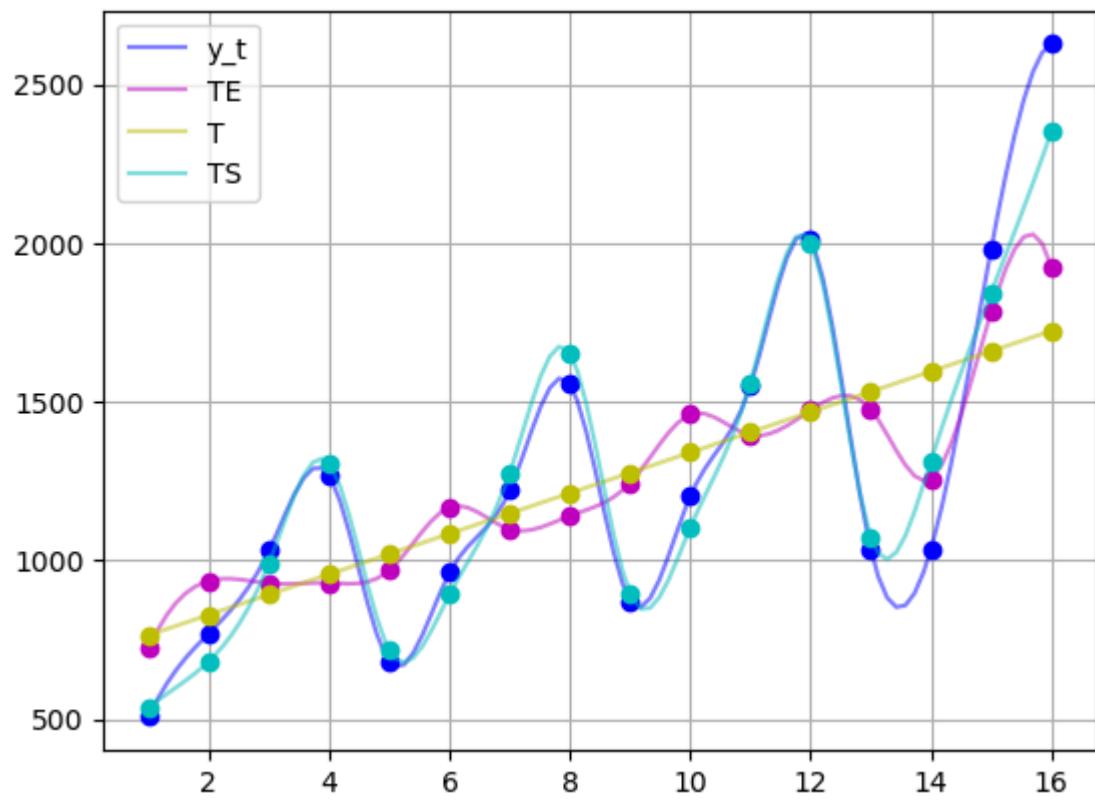
3.9999999999999996

Для выполнения последующих расчетов перене



$b_0 = 701.053657057197$

$b_1 = 63.8970922807628$



$$y = (701.0536570571975 + 63.89709228076279 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели знач



относительной ошибки E не могут быть исполь



Поэтому для оценки качества модели необходимо



абсолютной ошибки по формуле $E_{abs} = Y_t - (T \cdot S)$, т



Значения остатков в t_e представлены в графике.

^

$S_i = [0.7016649656940338, 0.8234574819611548, 1.1110947445374666, 1.3637828078073446]$

Предсказанные значения:

1254.0887583019241

1524.3855760147953

2127.8557790582636

2698.9200432727457

Рассчитаем стандартную ошибку регрессии: 123.8

◀ ▶

Стандартные ошибки прогноза будут следующими

◀ ▶

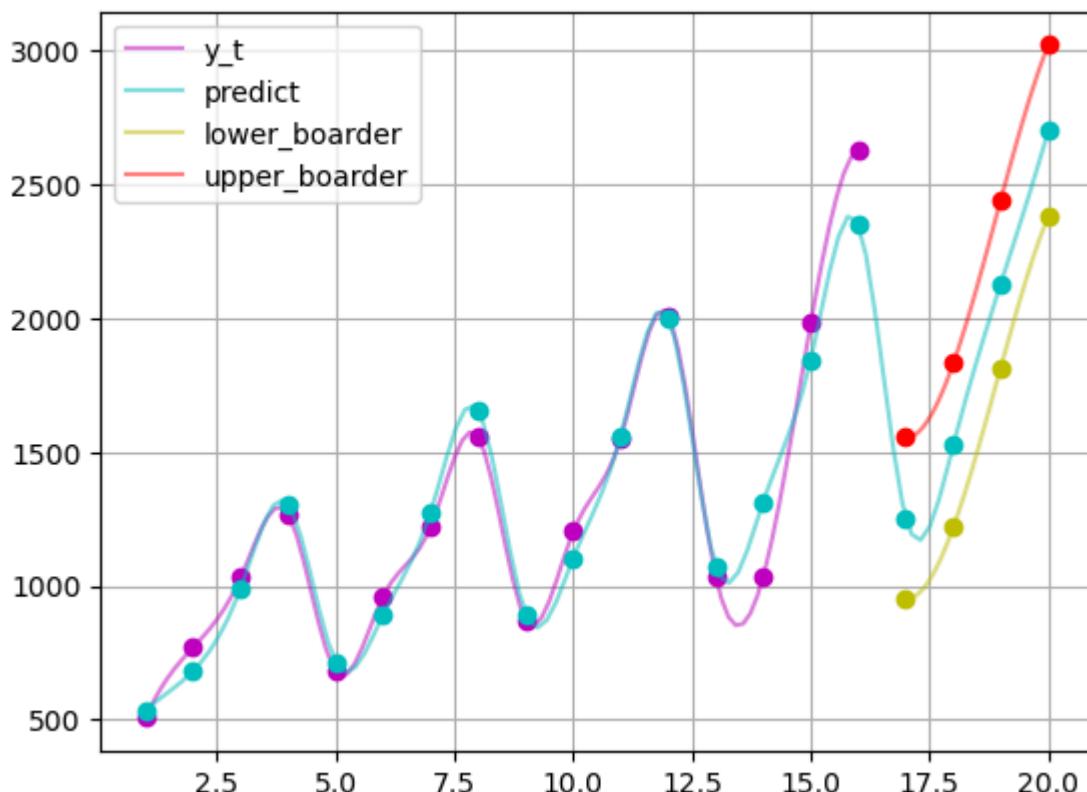
139.82027006861344

142.69358955516753

145.81977852511932

149.18294087793257

Критическое значение $t_{0.95;14} = 2.14$



- коэффициент детерминации

0.947589528503249

- средняя ошибка аппроксимации

6.69318609966898

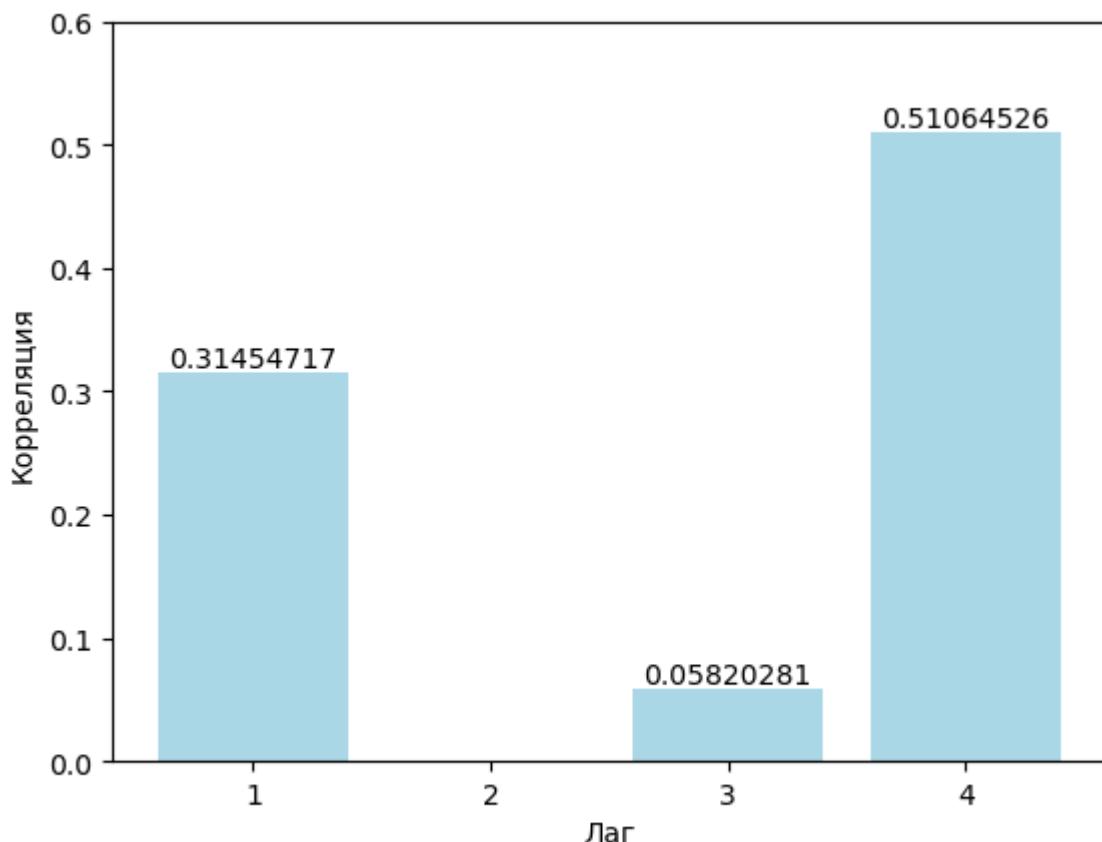
=====

Данные задания 5

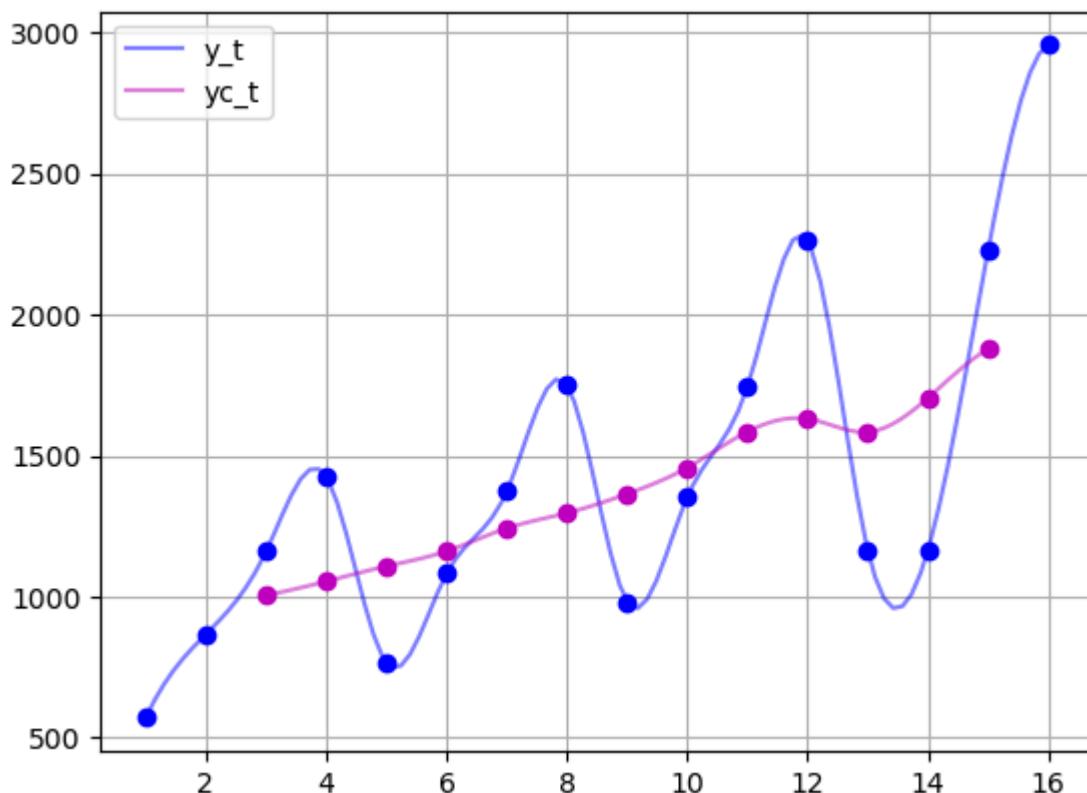
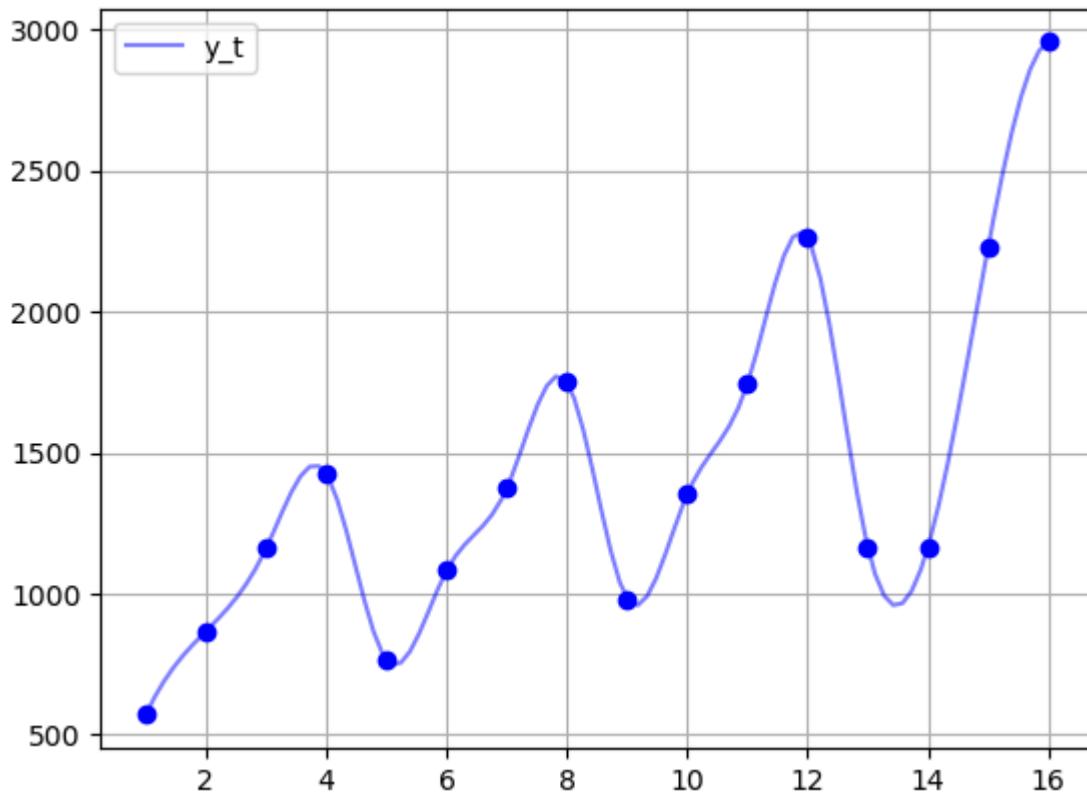
=====

--- Аддитивная модель 5_1 ---

4



-0.13273244



З а т е м р а с с ч и т а е м с р е д н и е о ц е н к и с е з о н н ы х к оэ

-425.85000000000001

-304.933333333334

123.07916666666672

472.5583333333334

П р и м о д е л и р о в а н и и п е р и о д и ч е с к и х к о л е б а н и й



ч т о с е з о н н ы е в о з д е й с т в и я з а п е р и о д в з а и м о п о



ч т о с у м м а з н а ч е н и й с е з о н н о й к о м п о н е н т ы п о в



-135.1458333333343

Д л я у с т р а н е н и я д а н н о г о п р o t i v o r e c h i я с к o r r e k



с р e d n i e з n a ч e n i j o c e n o k с e z o n n o й k o m p o n e n t



В e l i c h i n a k o r r e k t i r u y u щ e g o k o e ф f i c i e n t a b u



-33.78645833333336

Т e p e r y r a s c c i t a e m s k o r r e k t i r o v a n n y e z n a c h e n



r a z n o s t ь m e j d u e e c r e d n e j o c e n k o й S i k o r r e k t



-392.0635416666667

-271.146875

156.86562500000008

506.3447916666667

П r o v e r i m u s l o v i e v z a i m o p o g a s h a e m o s t i z n a c h e n



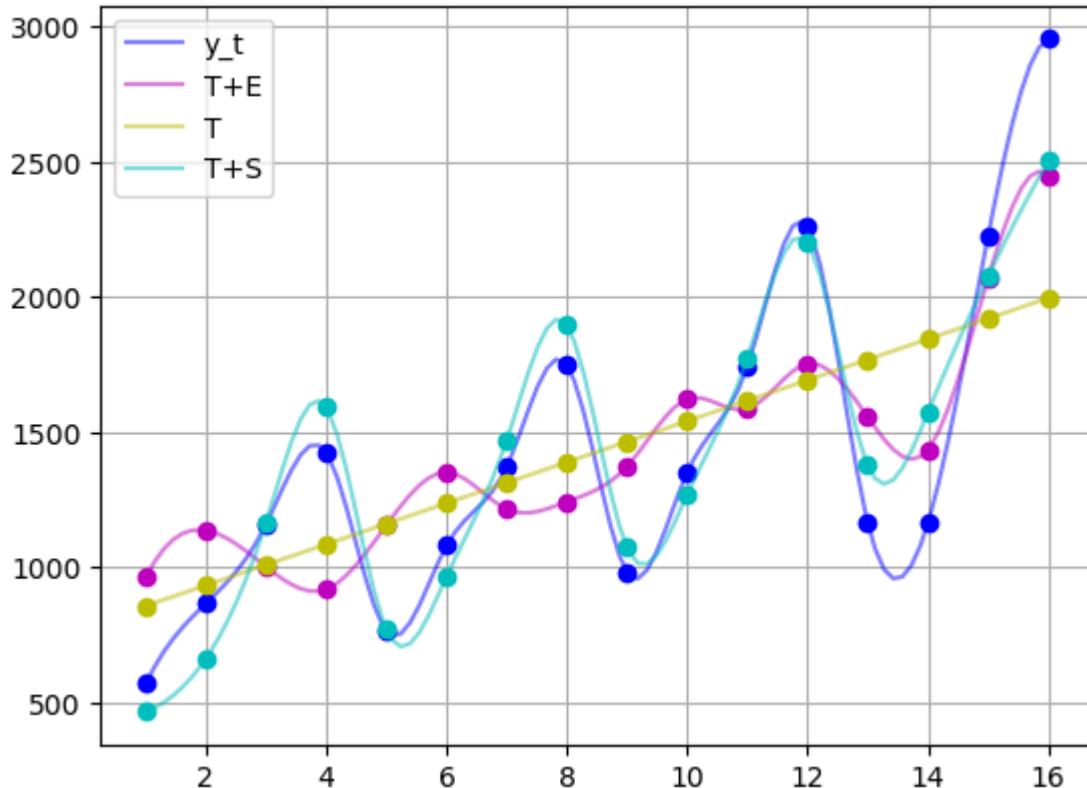
0.0 \approx 0

П e r e n e s e m p o l u c h e n n y e z n a c h e n i j S v t a b l i c u n a g



b0 = 782.531875000000

b1 = 75.9212500000000



$$\hat{y} = 782.531875 + 75.92125 \times t + S_i + e_t$$

$$\hat{S}_i = [-392.0635416666667, -271.146875, 156.86562500000008, 506.3447916666667]$$

Предсказанные значения:

1681.129583333336

1877.9675000000002

2381.90125

2807.301666666667

Рассчитаем стандартную ошибку регрессии: 204.0



Стандартные ошибки прогноза будут следующими



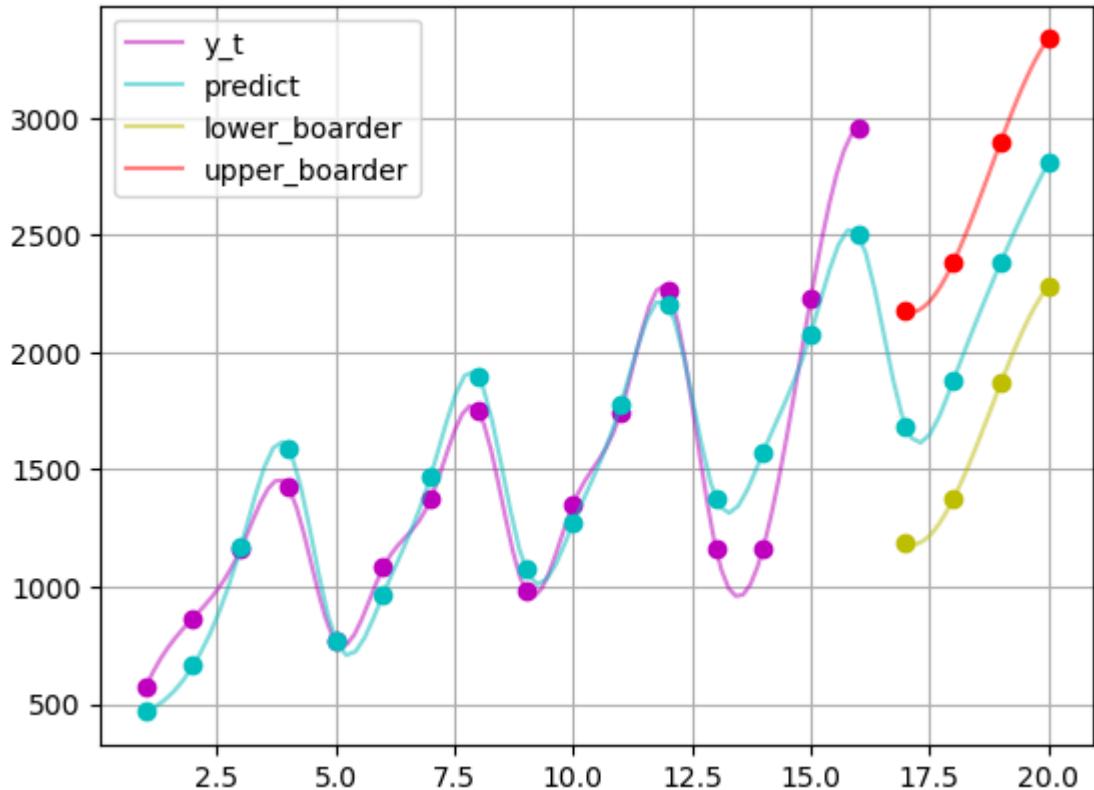
230.444455379149

235.18011025869063

240.33253139361847

245.87551966252525

Критическое значение $t_{0.95;14} = 2.14$



– коэффициент детерминации

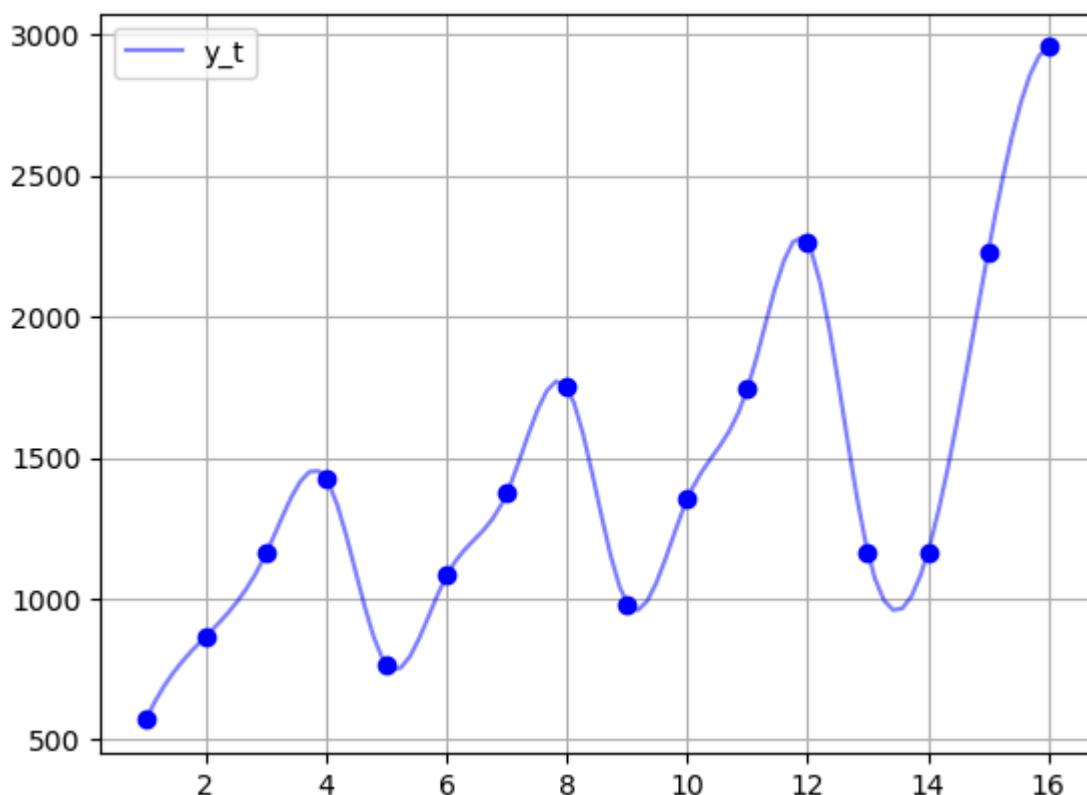
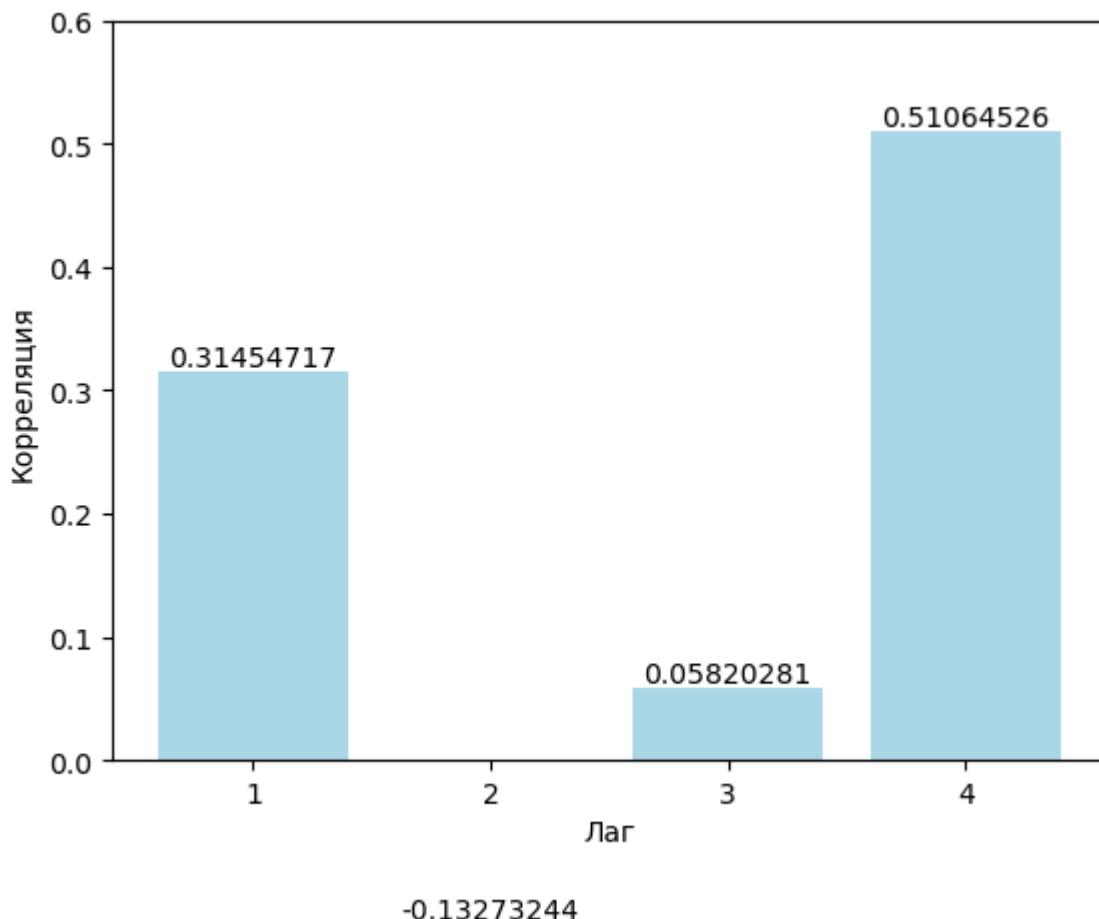
0.8944457675869272

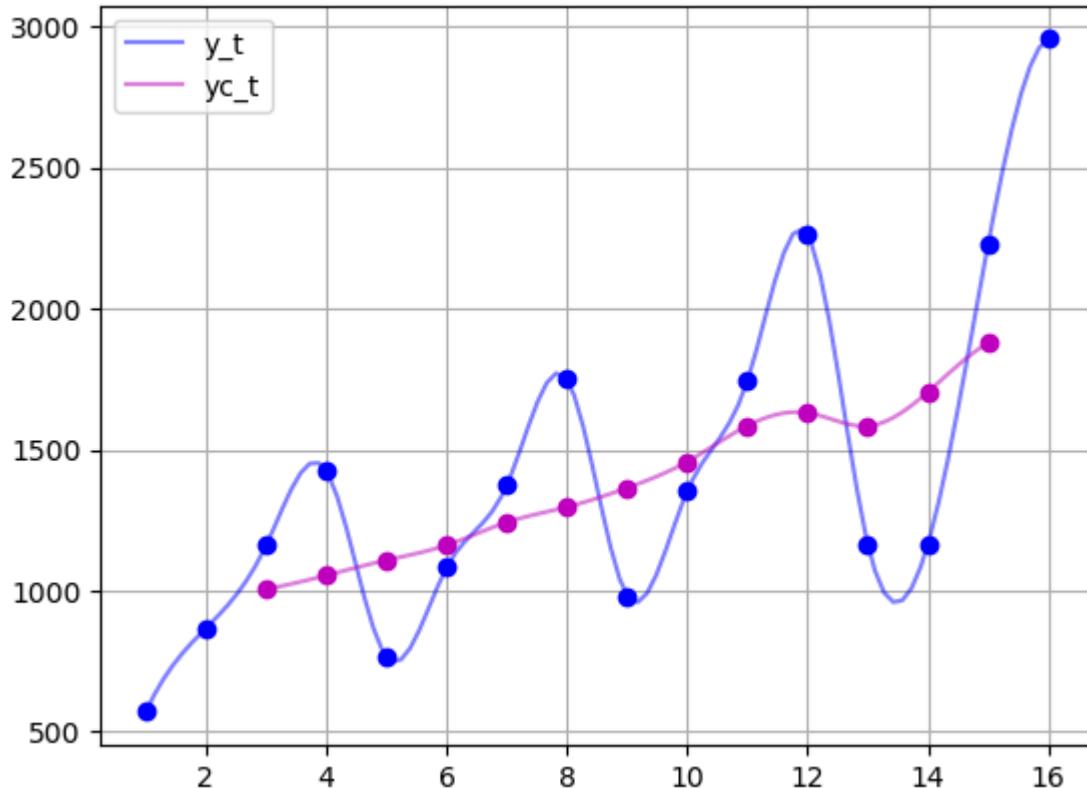
– средняя ошибка аппроксимации

11.003753604650605

--- Мультипликативная модель 5_2 ---

4





Взаимопогашаемость сезонных воздействий в м



>

что сумма значений iS сезонной компоненты S д



>

то есть для данного примера – четырем. Сумми



>

0.6929652166845762

0.81325187471163

1.0973501382156245

1.3468721377786348

3.950439367390466

Рассчитаем корректирующий коэффициент:

1.012545600122012

Определим скорректированные значения сезон



>

умножив ее средние оценки iS на корректирую



>

0.7016588811915643

0.8234546075302386

1.1111170542435123

1.3637694570346848

Проверим выполнение свойства взаимопогаша-



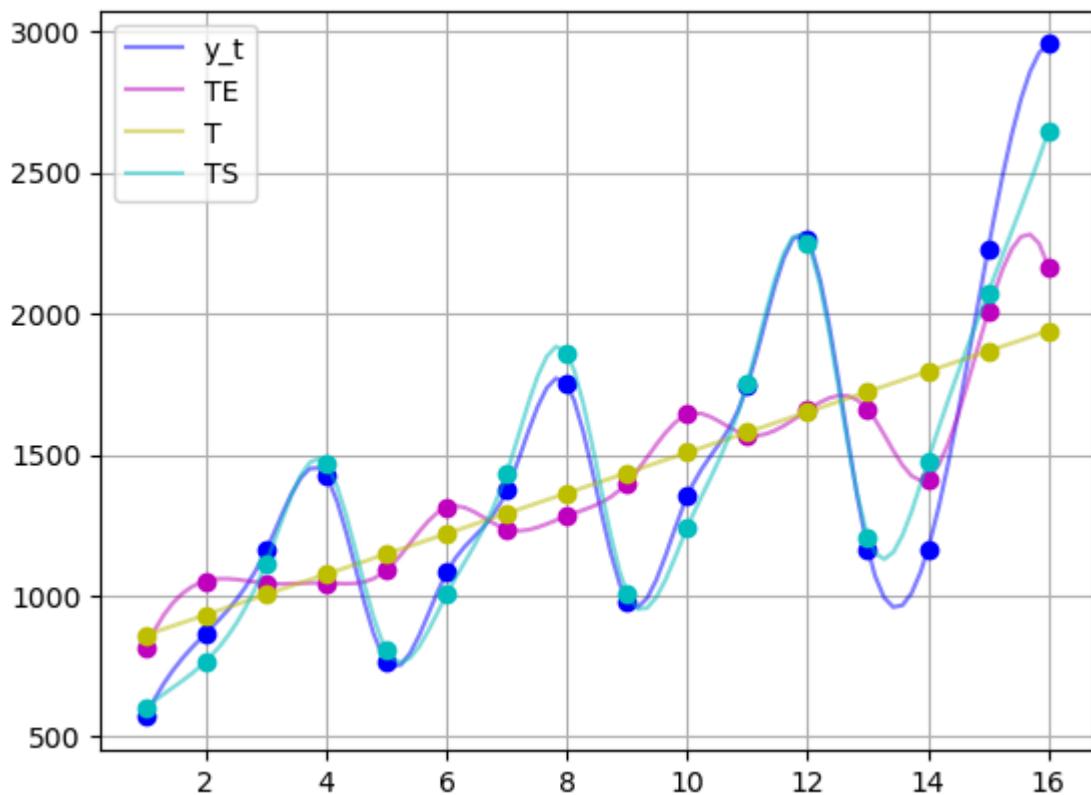
3.9999999999999996

Для выполнения последующих расчетов перене-



$b_0 = 788.667247314310$

$b_1 = 71.8877146704665$



$$y = (788.66724731431 + 71.88771467046647 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели значи-



относительной ошибки E не могут быть исполь-



Поэтому для оценки качества модели необходимо,



абсолютной ошибки по формуле $E_{abs} = Y_t - (T \cdot S)$, т



Значения остатков в t_e представлены в графике.

^

$S_i = [0.7016588811915643, 0.8234546075302386, 1.1111170542435123, 1.3637694570346848]$

Предсказанные значения:

1410.8664869836539

1714.964536273024

2393.9392780722055

3036.3256957230037

Рассчитаем стандартную ошибку регрессии: 139.2

◀ ▶

Стандартные ошибки прогноза будут следующими

◀ ▶

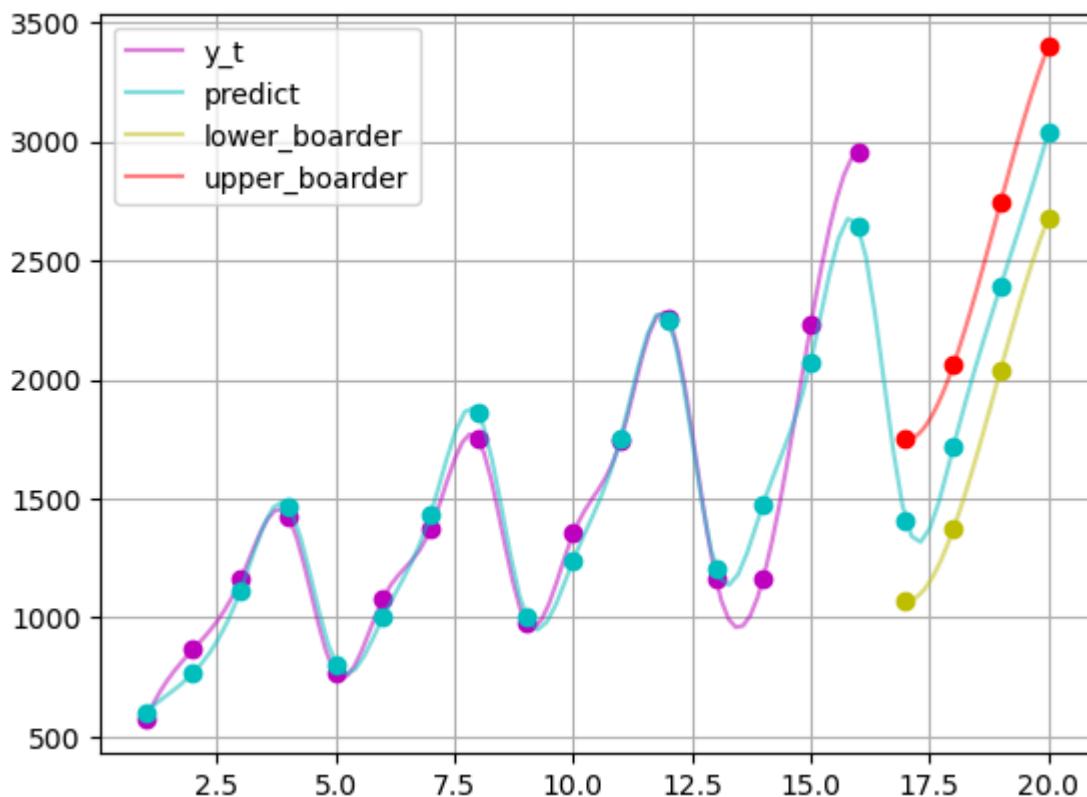
157.27965446759583

160.5117659189503

164.02832271542297

167.81144380730268

Критическое значение $t_{0.95;14} = 2.14$



- коэффициент детерминации

0.9476035614573897

- средняя ошибка аппроксимации

6.69240165962322

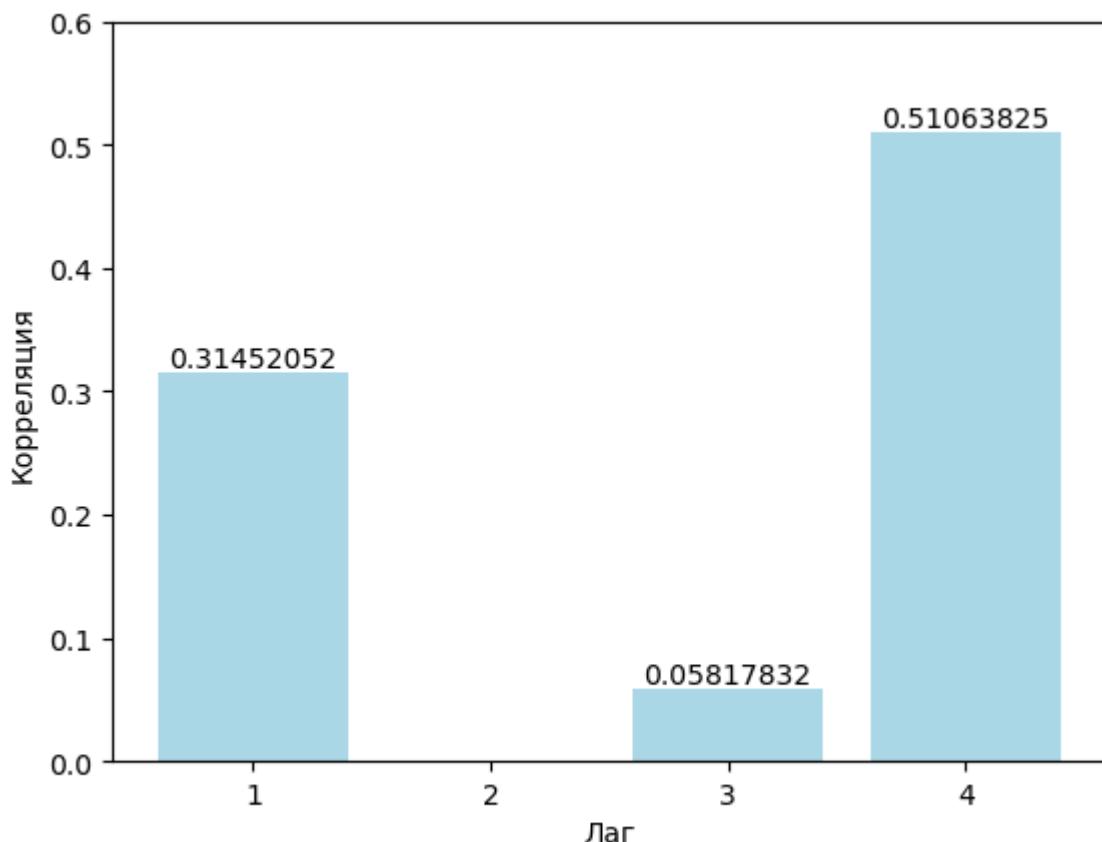
=====

Данные задания 6

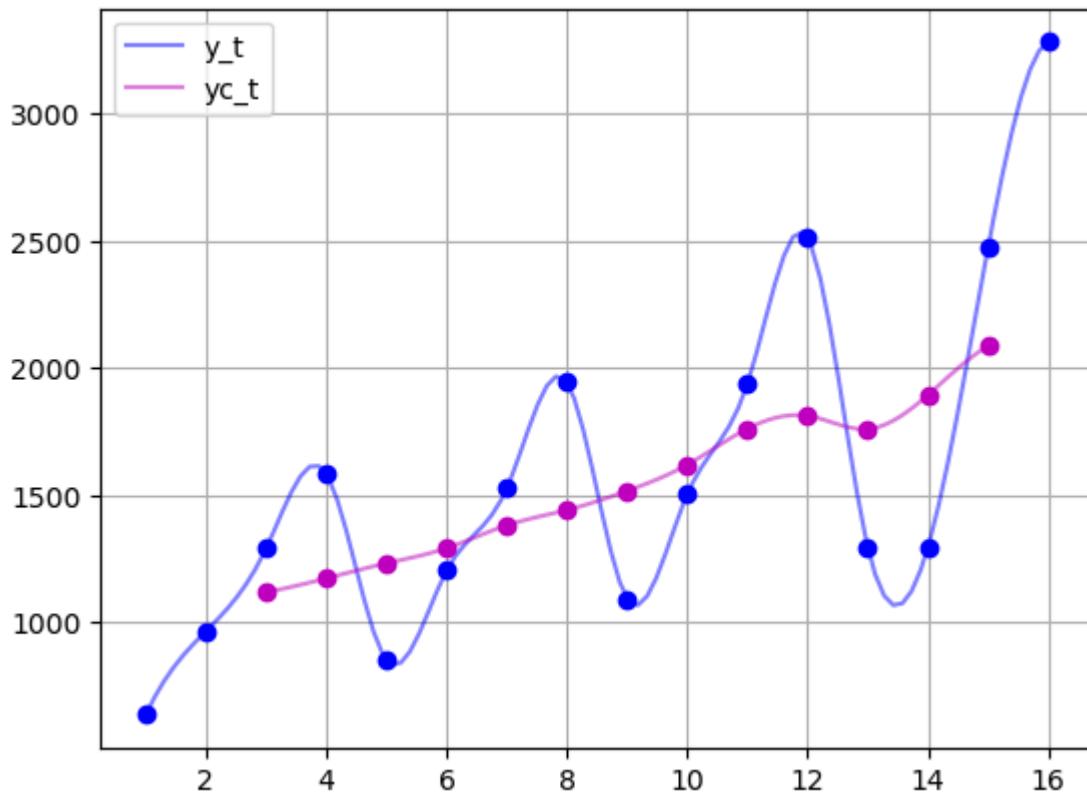
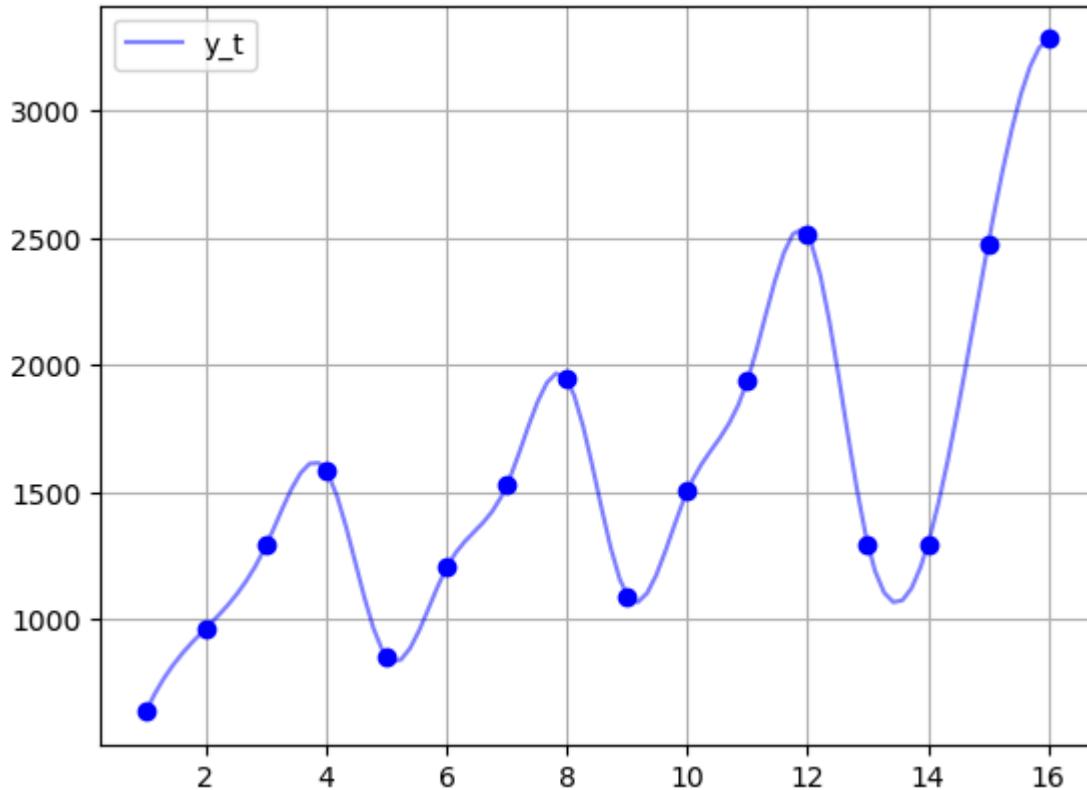
=====

--- Аддитивная модель 6_1 ---

4



-0.13276406



З а т е м р а с с ч и т а е м с р е д н и е о ц е н к и с е з о н н ы х к о ю

◀ ▶

-473.15000000000001

-338.8208333333327

136.71666666666678

525.0749999999999

П р и м о д е л и р о в а н и и п е р и о д и ч е с к и х к о л е б а н и й



ч т о с е з о н н ы е в о з д е й с т в и я з а п е р и о д в з а и м о п о



ч т о с у м м а з н а ч е н и й с е з о н н о й к о м п о н е н т ы п о в



-150.1791666666656

Д л я у с т р а н е н и я д а н н о г о п р o t i v o r e c h i я с к o r r e k



с р e d n i e з n a ч e n i j o c e n o k с e z o n n o й k o m p o n e n t



В e l i c h i n a k o r r e k t i r u y u щ e g o k o e ф i c i e n t a b u



-37.54479166666664

Т e p e r y r a c s c i t a e m s k o r r e k t i r o v a n n y e z n a c h e n



r a z n o s t ь m e j d u e e s r e d n e j o c e n k o й S i k o r r e k t



-435.6052083333345

-301.2760416666663

174.2614583333342

562.6197916666665

П r o v e r i m u s l o v i e v z a i m o p o g a s h a e m o s t i z n a c h e n



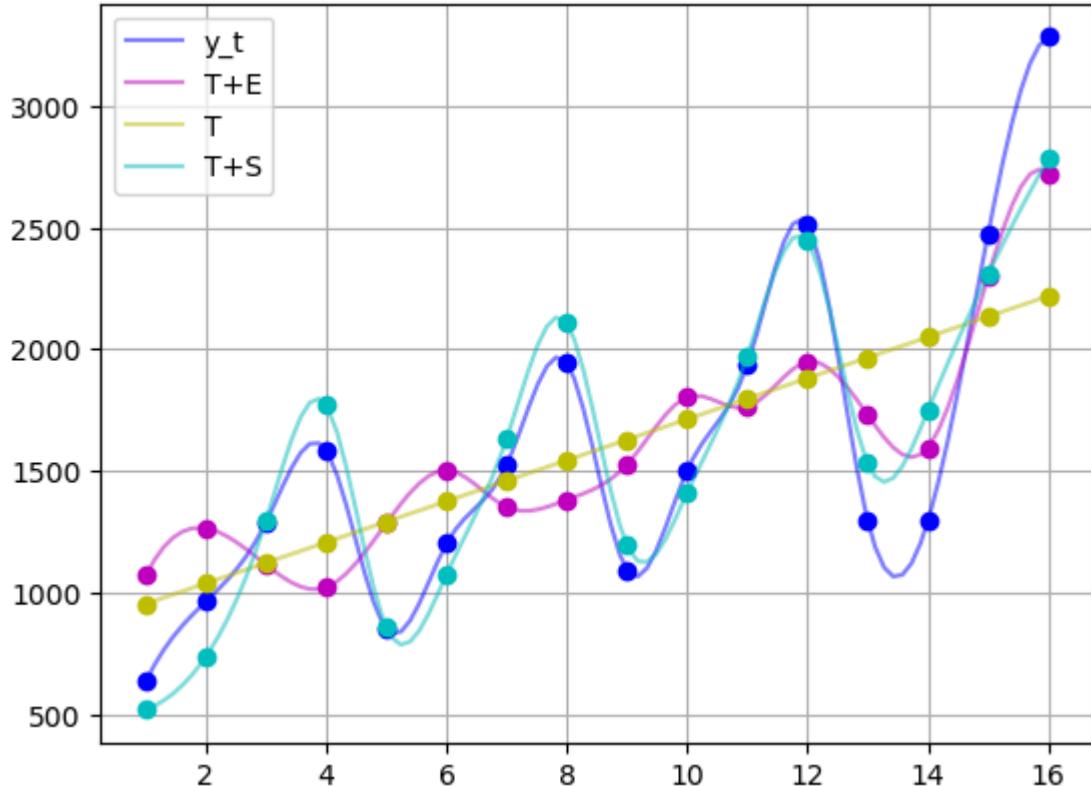
$-2.2737367544323206e - 13 \approx 0$

П e r e n e s e m p o l u c h e n n y e z n a c h e n i j S v t a b l i c u n a g



$b0 = 869.530625000000$

$b1 = 84.3515441176471$



$$y = 869.530625 + 84.35154411764705 \times t + S_i + e_t$$

^

$$S_i = [-435.6052083333345, -301.27604166666663, 174.2614583333342, 562.6197916666665]$$



Предсказанные значения:

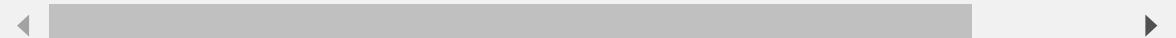
1867.901666666666

2086.5823774509804

2646.471421568627

3119.1812990196076

Рассчитаем стандартную ошибку регрессии: 226.7



Стандартные ошибки прогноза будут следующими



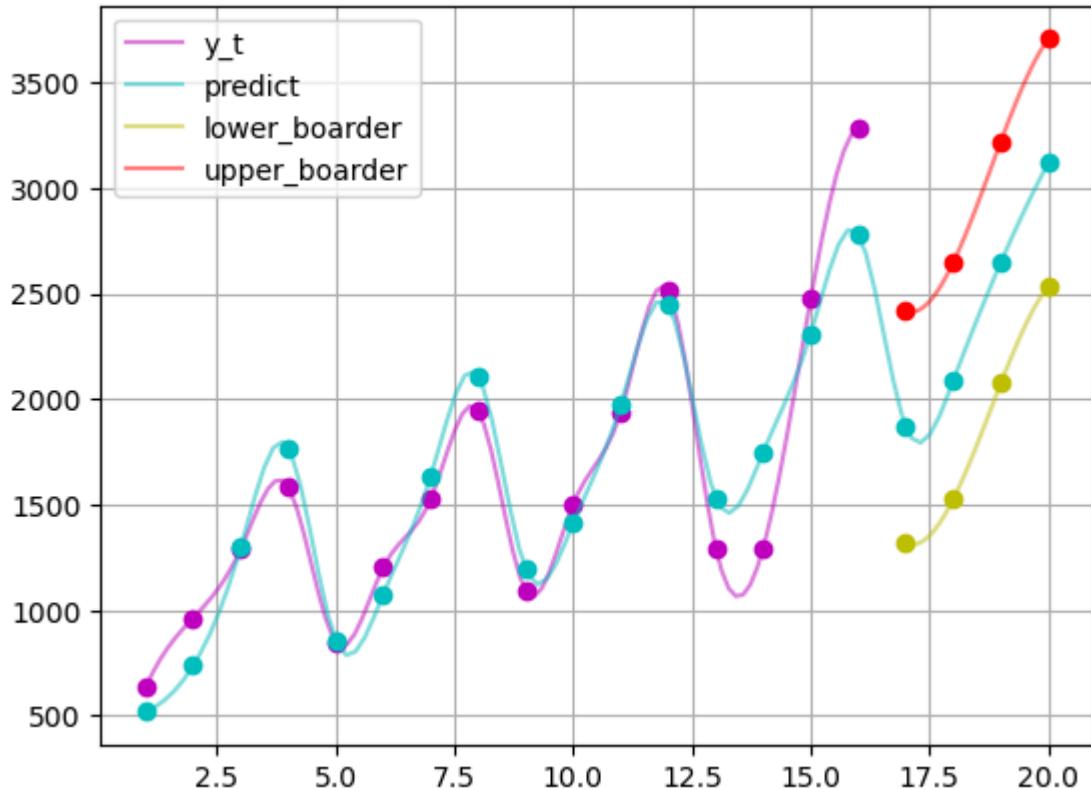
256.0752317408373

261.3376014460577

267.0630914951987

273.22258881598117

Критическое значение $t_{0.95;14} = 2.14$



– коэффициент детерминации

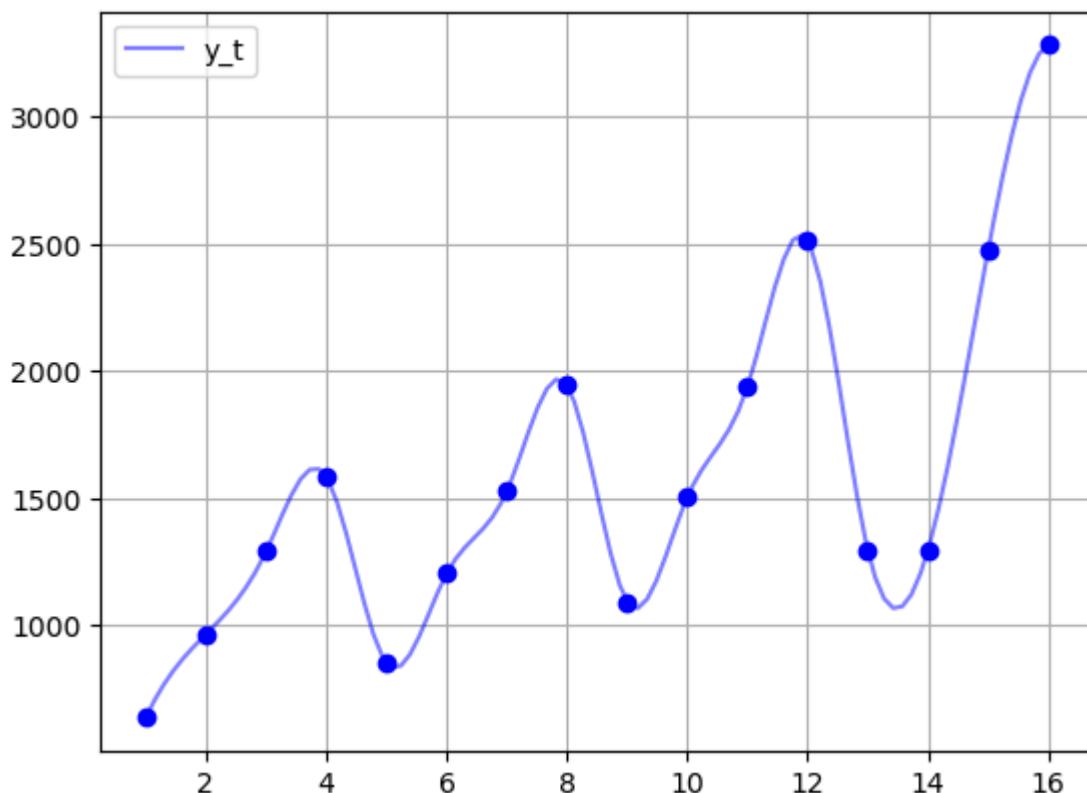
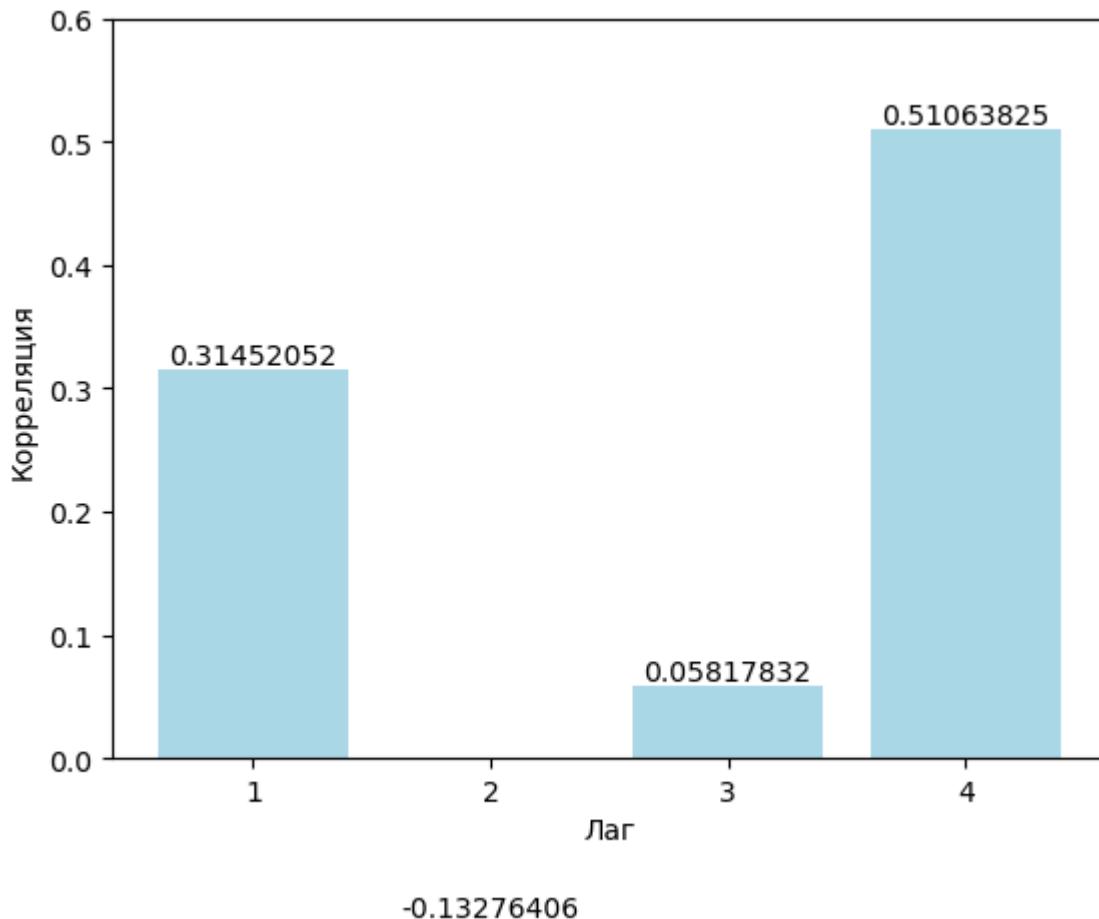
0.8944197049907777

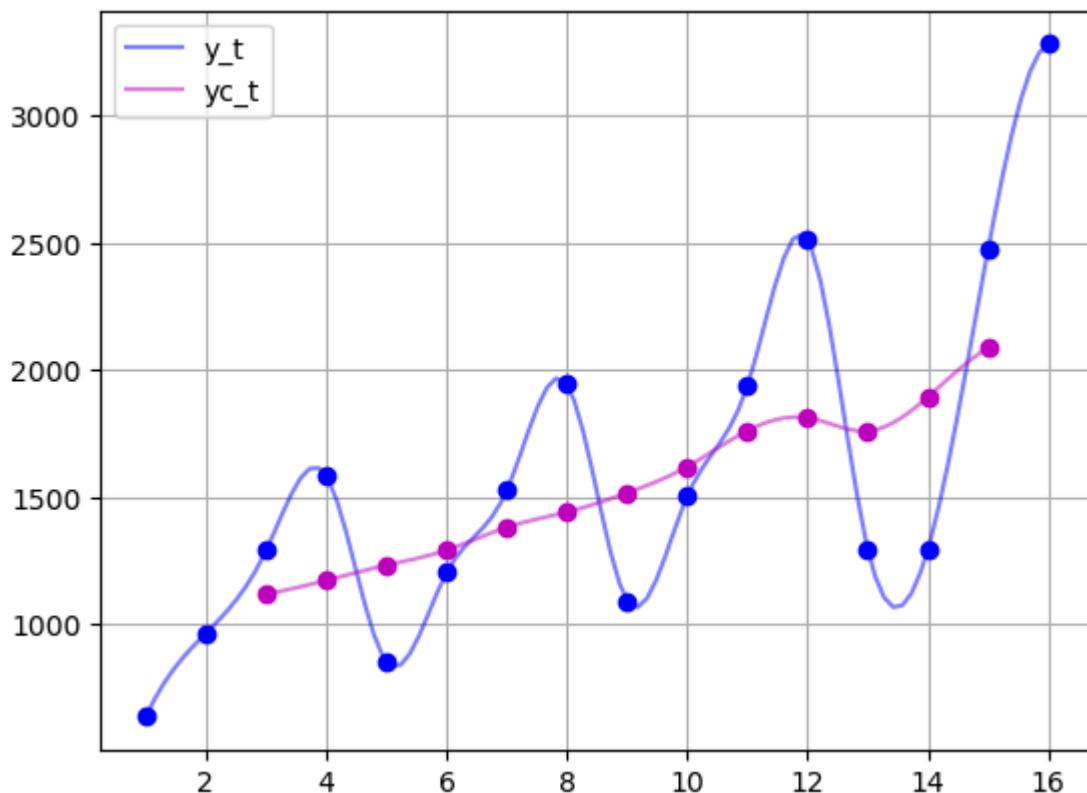
– средняя ошибка аппроксимации

11.004762099203653

--- Мультипликативная модель 6_2 ---

4





Взаимопогашаемость сезонных воздействий в м



>

что сумма значений iS сезонной компоненты S д



>

то есть для данного примера – четырем. Сумми



>

0.6929746187861007

0.8132480337215892

1.0973222168483519

1.34688007694674

3.950424946302782

Рассчитаем корректирующий коэффициент:

1.0125492964354672

Определим скорректированные значения сезон



>

умноживе средние оценки iS на корректирую



>

0.7016709626995024

0.8234537243723222

1.1110928386328058

1.3637824742953695

Проверим выполнение свойства взаимопогаша-



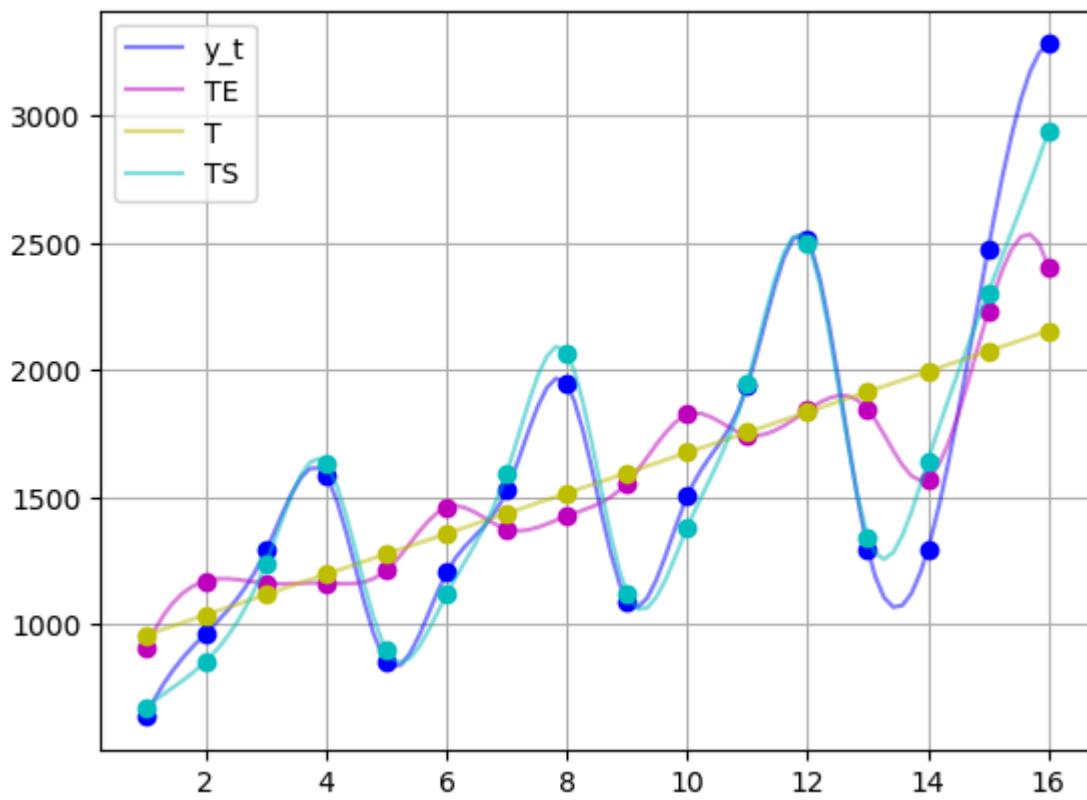
4.0

Для выполнения последующих расчетов перене-



$b_0 = 876.361529651370$

$b_1 = 79.8684493263521$



$$y = (876.36152965137 + 79.86844932635206 \times t) \times S_i \times e_t$$

Для расчета показателей качества модели значи-



относительной ошибки E не могут быть исполь-



Поэтому для оценки качества модели необходимо,



абсолютной ошибки по формуле $E_{abs} = Y_t - (T \cdot S)$, т



Значения остатков в t_e представлены в графике.

^

$$S_i = [0.7016709626995024, 0.8234537243723222, 1.1110928386328058, 1.3637824742953695]$$

Предсказанные значения:

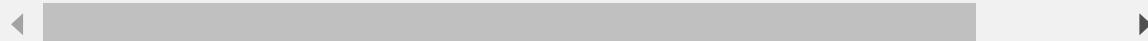
1567.620757561629

1905.4666625253265

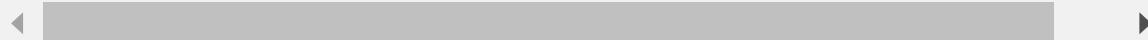
2659.80299915405

3373.630324093755

Рассчитаем стандартную ошибку регрессии: 154.7



Стандартные ошибки прогноза будут следующими



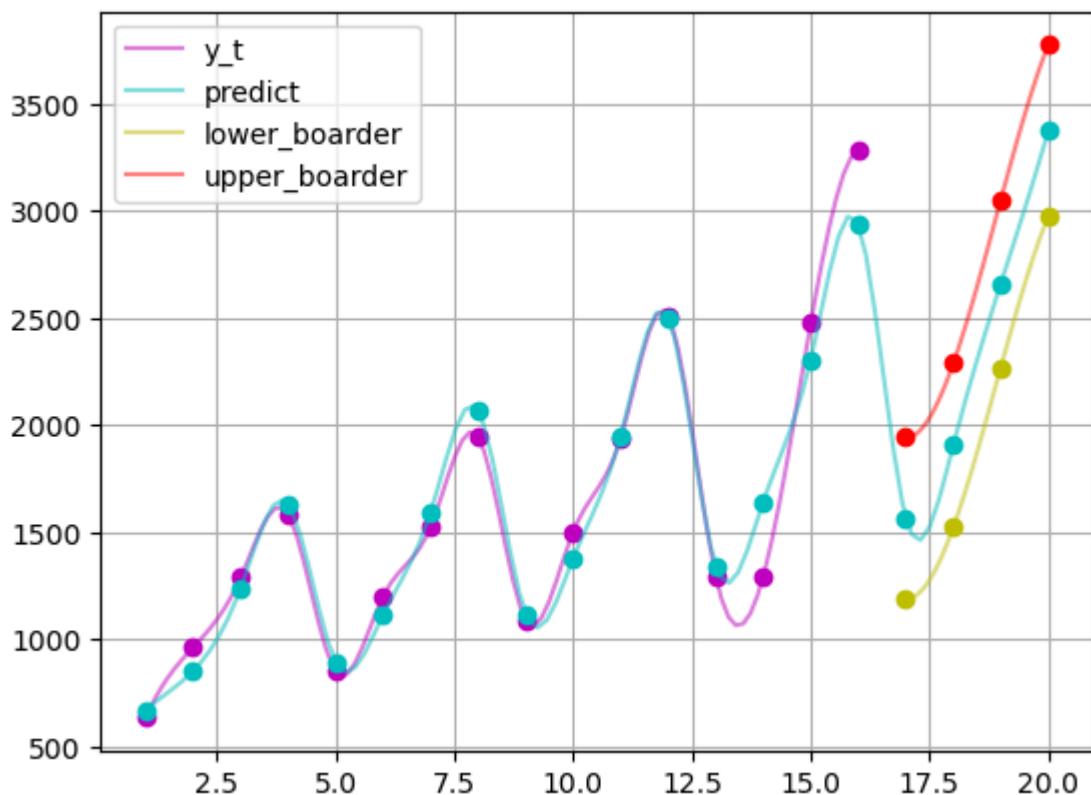
174.7878189867493

178.37972483888697

182.28774011828932

186.4919932802815

Критическое значение $t_{0.95;14} = 2.14$



— коэффициент детерминации

0.9475810740505479

– средняя ошибка аппроксимации

6.693415706339161