

1. Импорт данных и визуализация исходного ряда

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
import numpy as np
import sys
import io

sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')

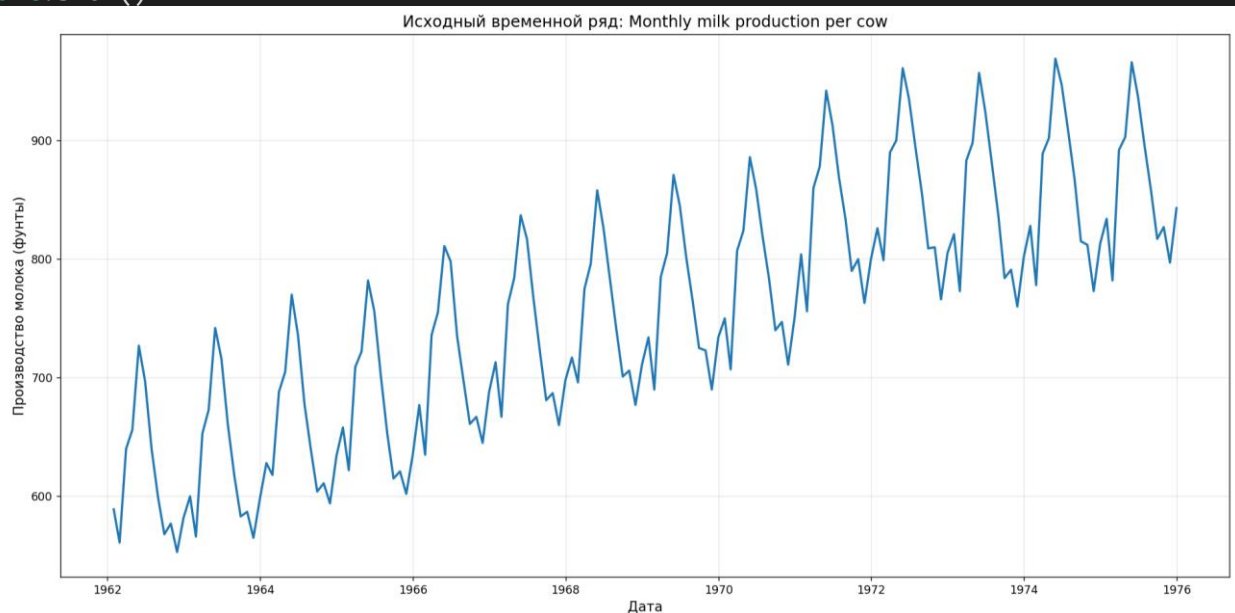
# 1. Импорт данных
df = pd.read_csv('D:\\VSU\\sem7\\AI\\lab2\\milk.csv')
# Создаем временную шкалу для 14 лет (168 месяцев)
df['Дата'] = pd.date_range(start='1962-01-01', periods=len(df), freq='M')
```

Информация о временном ряде:

- Период: 1962-01 - 1975-12
- Количество наблюдений: 168
- Среднее значение: 754.71
- Стандартное отклонение: 102.20

2. Визуализация и анализ временного ряда

```
# 2. Визуализация исходного временного ряда
plt.figure(figsize=(14, 8))
plt.plot(df['Дата'], df['milk'], linewidth=2)
plt.xlabel('Дата', fontsize=12)
plt.ylabel('Производство молока (фунты)', fontsize=12)
plt.title('Исходный временной ряд: Monthly milk production per cow', fontsize=14)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```



3. Построение прогностической модели

```
# 3. Построение прогностической модели
model = ExponentialSmoothing(df['milk'],
                             trend='add',
                             seasonal='add',
                             seasonal_periods=12)

model_fit = model.fit()

# Получение подогнанных значений
fitted_values = model_fit.fittedvalues
```

4. Прогноз на заданный промежуток времени

```
# 4. Прогноз на 8 месяцев
forecast_steps = 8
forecast = model_fit.forecast(steps=forecast_steps)

# Создаем даты для прогноза
last_date = df['Дата'].iloc[-1]
forecast_dates = pd.date_range(start=last_date + pd.DateOffset(months=1),
                               periods=forecast_steps, freq='M')
```

5. Построение графика исходного ряда и подогнанных данных с учетом прогноза

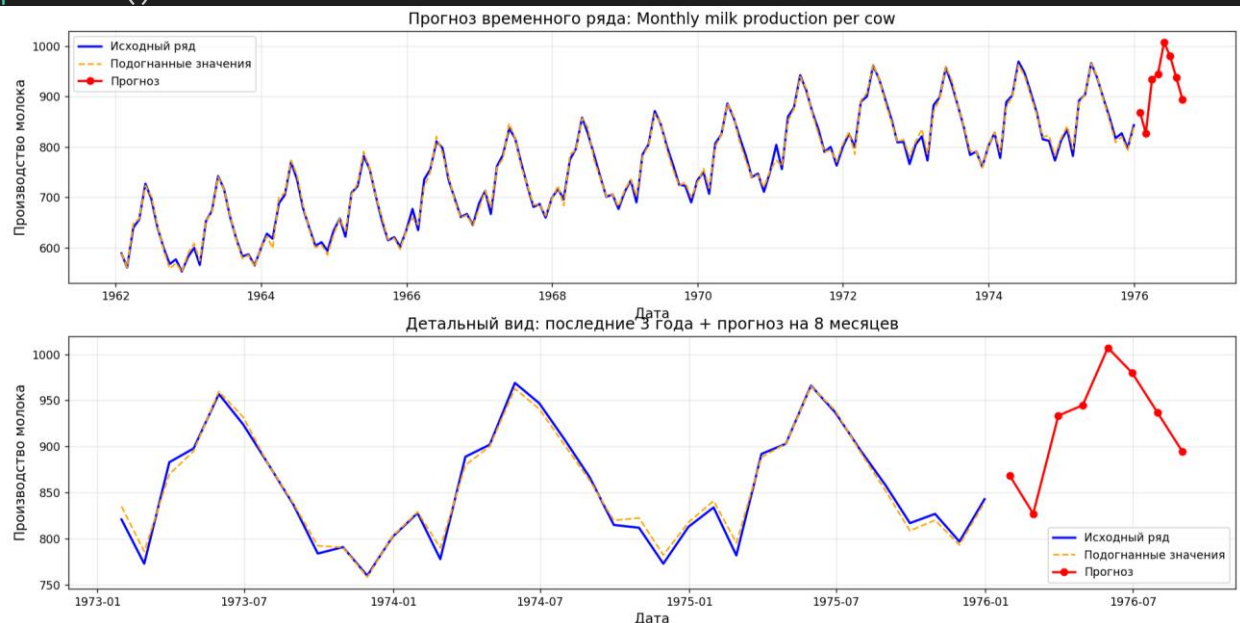
```
# 5. Построение графиков исходного ряда и подогнанных данных с учетом прогноза
print("\n5. Построение графиков с прогнозом")
plt.figure(figsize=(15, 10))

# Основной график
plt.subplot(2, 1, 1)
plt.plot(df['Дата'], df['milk'], label='Исходный ряд', linewidth=2, color='blue')
plt.plot(df['Дата'], fitted_values, label='Подогнанные значения', linewidth=1.5,
color='orange', linestyle='--')
plt.plot(forecast_dates, forecast, label='Прогноз', linewidth=2, color='red',
marker='o')
plt.xlabel('Дата', fontsize=12)
plt.ylabel('Производство молока', fontsize=12)
plt.title('Прогноз временного ряда: Monthly milk production per cow',
fontsize=14)
plt.legend()
plt.grid(True, alpha=0.3)

# График только последних 3 лет + прогноз для лучшей видимости
plt.subplot(2, 1, 2)
last_3_years = 36 # 3 года * 12 месяцев
plt.plot(df['Дата'].iloc[-last_3_years:], df['milk'].iloc[-last_3_years:],
label='Исходный ряд', linewidth=2, color='blue')
```

```
plt.plot(df['Дата'].iloc[-last_3_years:], fitted_values.iloc[-last_3_years:],
        label='Подогнанные значения', linewidth=1.5, color='orange',
        linestyle='--')
plt.plot(forecast_dates, forecast, label='Прогноз', linewidth=2, color='red',
        marker='o')
plt.xlabel('Дата', fontsize=12)
plt.ylabel('Производство молока', fontsize=12)
plt.title('Детальный вид: последние 3 года + прогноз на 8 месяцев', fontsize=14)
plt.legend()
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```



6. Набор чисел – спрогнозированных значений рассматриваемой величины для каждого из будущих моментов времени

```
# 6. Вывод спрогнозированных значений
```

```
print("Дата\t\t\t\t\tПрогноз")
for i, (date, value) in enumerate(zip(forecast_dates, forecast), 1):
    print(f"{date.strftime('%Y-%m-%d')}\t\t\t\t\t{value:.2f}")

print("\nМассив прогнозированных значений:")
forecast_array = np.array(forecast)
print("[", end="")
for i, value in enumerate(forecast_array):
    if i == len(forecast_array) - 1:
        print(f"{value:.2f}", end="")
    else:
        print(f"{value:.2f} ", end="")
print("]")
```

Дата	Прогноз
1976-01-31	868.61
1976-02-29	827.16
1976-03-31	933.53
1976-04-30	944.77
1976-05-31	1007.12
1976-06-30	979.88
1976-07-31	937.10
1976-08-31	894.43

Визуализация тренда и сезонности

```
# Разложение ряда на компоненты (для демонстрации)
from statsmodels.tsa.seasonal import seasonal_decompose

decomposition = seasonal_decompose(df['milk'], model='additive', period=12)

plt.figure(figsize=(15, 12))

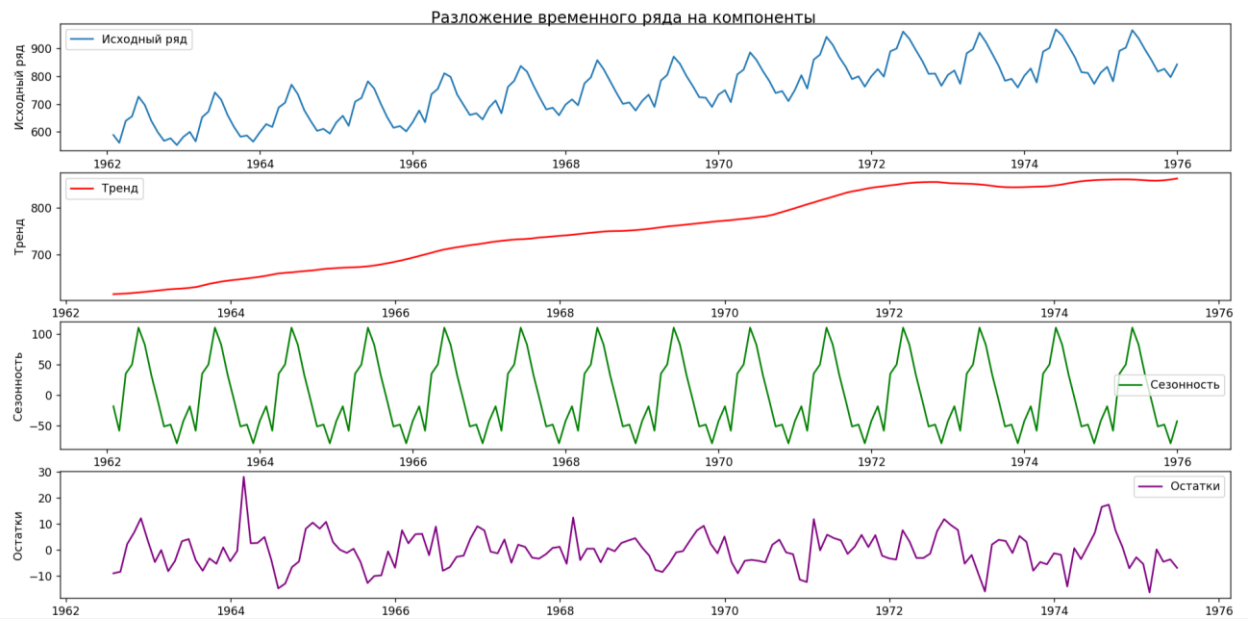
plt.subplot(4, 1, 1)
plt.plot(df['Дата'], df['milk'], label='Исходный ряд')
plt.ylabel('Исходный ряд')
plt.legend()

plt.subplot(4, 1, 2)
plt.plot(df['Дата'], decomposition.trend, label='Тренд', color='red')
plt.ylabel('Тренд')
plt.legend()

plt.subplot(4, 1, 3)
plt.plot(df['Дата'], decomposition.seasonal, label='Сезонность', color='green')
plt.ylabel('Сезонность')
plt.legend()

plt.subplot(4, 1, 4)
plt.plot(df['Дата'], decomposition.resid, label='Остатки', color='purple')
plt.ylabel('Остатки')
plt.legend()

plt.suptitle('Разложение временного ряда на компоненты', fontsize=14)
plt.tight_layout()
plt.show()
```



Итоговые выводы

- Ряд имеет восходящий тренд - производство молока постепенно увеличивается
- Выраженная сезонность - пик производства весной, спад осенью
- Прогноз показывает продолжение роста производства молока
- Сезонная компонента стабильна - паттерн повторяется ежегодно

Дата	Прогноз
1976-01-31	868.61
1976-02-29	827.16
1976-03-31	933.53
1976-04-30	944.77
1976-05-31	1007.12
1976-06-30	979.88
1976-07-31	937.10
1976-08-31	894.43