

```

In [1]: import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split

# Загрузка данных
data = pd.read_csv('airline-passengers.csv')

# Преобразование столбца 'Month' в тип datetime и установка его в качестве индекса
data['Month'] = pd.to_datetime(data['Month'])
data.set_index('Month', inplace=True)

# Фильтрация данных по заданному временному периоду
data = data['1952-01-01':'1957-12-01']

# Декомпозиция временного ряда
result = seasonal_decompose(data['Passengers'], model='multiplicative')
result.plot()
plt.show()

# Автокорреляция
plot_acf(data['Passengers'], lags=24)
plt.show()

# Функция для создания новых признаков
def create_features(df, lags, rolling_windows):
    for lag in lags:
        df[f'lag_{lag}'] = df['Passengers'].shift(lag)
    for window in rolling_windows:
        df[f'rolling_mean_{window}'] = df['Passengers'].rolling(window=window).mean()
    return df

# Применение функции для создания новых признаков
data = create_features(data, lags=[1, 2, 12], rolling_windows=[3, 6, 12])
data.dropna(inplace=True) # Удаление строк с пропусками из-за создания признаков

# Разделение данных на признаки и целевую переменную
X = data.drop('Passengers', axis=1)
y = data['Passengers']

# Разделение на обучающую и валидационную выборки
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, shuffle=False)

# Обучение линейной модели
model = LinearRegression()
model.fit(X_train, y_train)

# Предсказание и оценка модели
y_pred_train = model.predict(X_train)
y_pred_val = model.predict(X_val)
mae_train = mean_absolute_error(y_train, y_pred_train)
mae_val = mean_absolute_error(y_val, y_pred_val)

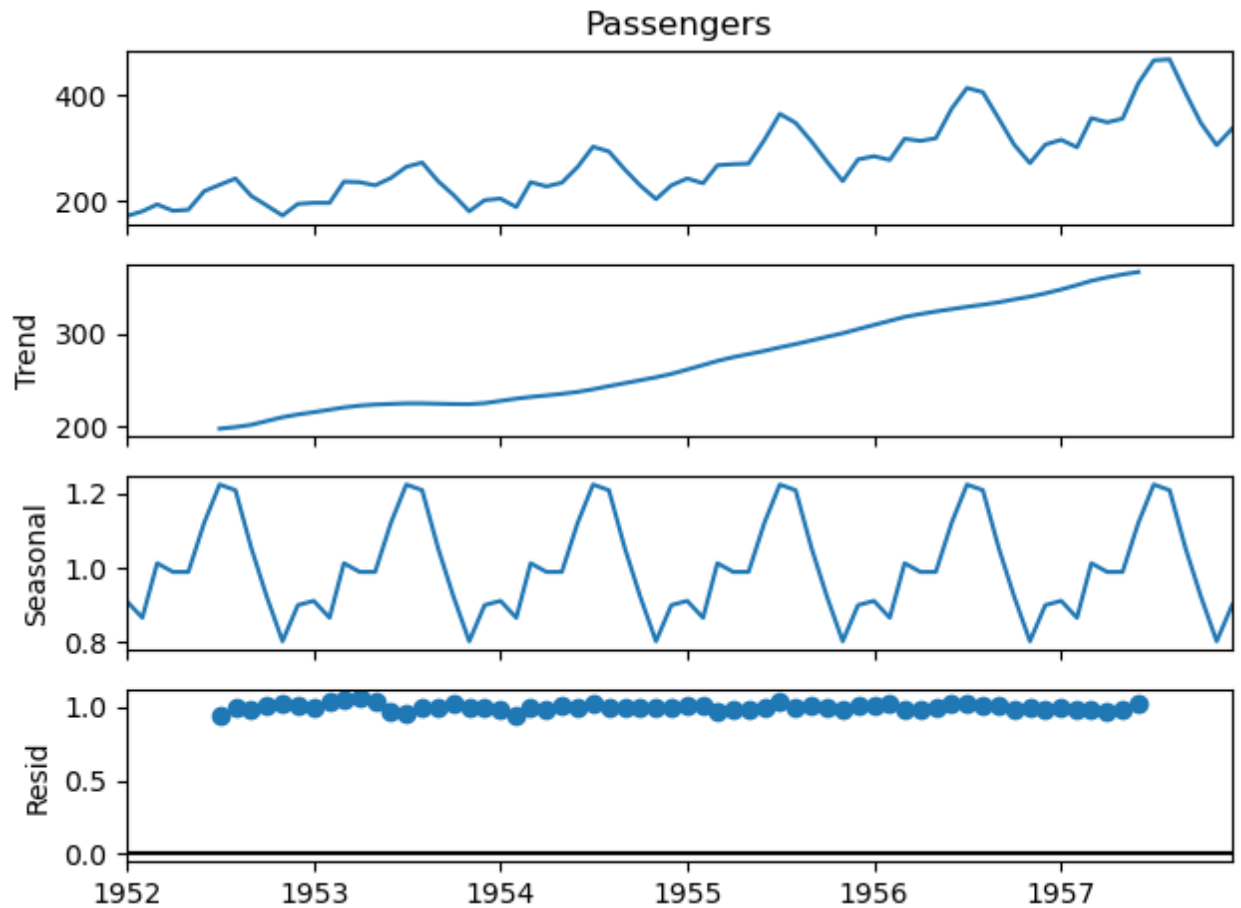
```

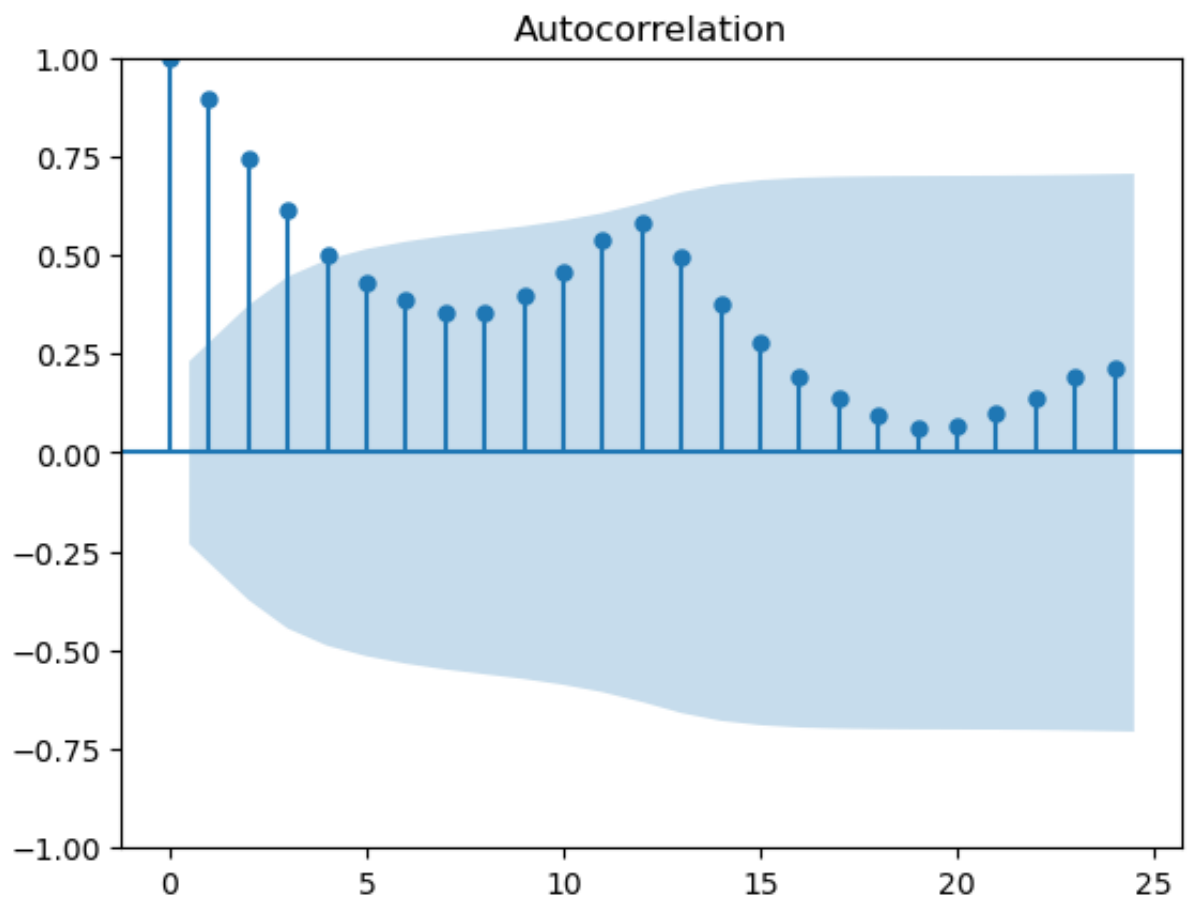
```

print(f"MAE на обучающей выборке: {mae_train}")
print(f"MAE на валидационной выборке: {mae_val}")

# Визуализация результатов
plt.figure(figsize=(14, 7))
plt.plot(data.index, data['Passengers'], label='Исходные данные')
plt.plot(X_train.index, y_pred_train, label='Прогноз на обучении')
plt.plot(X_val.index, y_pred_val, label='Прогноз на валидации', color='r')
plt.legend()
plt.show()

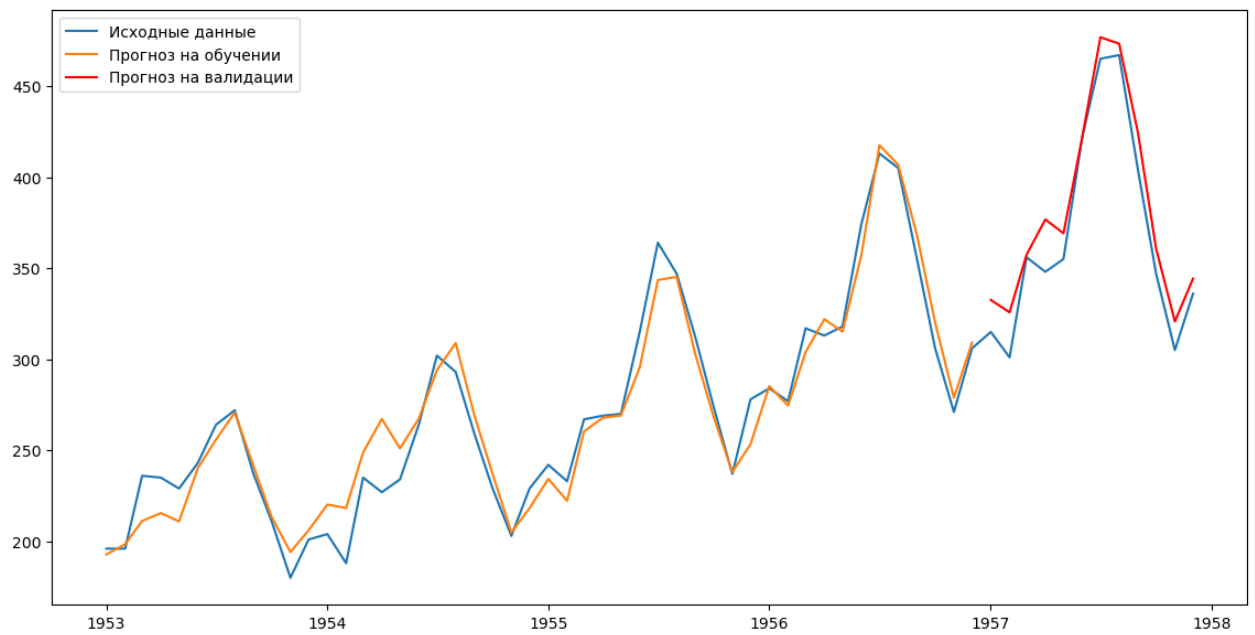
```





MAE на обучающей выборке: 10.038510087517126

MAE на валидационной выборке: 13.562528613951494



In [ ]: