



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана»  
(национальный исследовательский университет)  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Фундаментальные науки

КАФЕДРА Вычислительная математика и математическая физика (ФН-11)

## КУРСОВАЯ РАБОТА

на тему:

Применение рекуррентных нейронных сетей  
для прогнозирования временных рядов.

Дисциплина:

Численные методы

Студент группы ФН11-62Б

\_\_\_\_\_  
(Подпись, дата)

**Очкин Н.В.**  
(И.О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

**Зубарев К.М.**  
(И.О. Фамилия)

Оценка: \_\_\_\_\_

Москва, 2024

## СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель НИР

\_\_\_\_\_

Подпись, дата

Зубарев К.М.

Исполнители:

Студент группы

ФН11-62Б

\_\_\_\_\_

Подпись, дата

Очкин Н.В.

Нормоконтролёр

ст. преп. каф. ФН11

\_\_\_\_\_

Подпись, дата

Прозоровский А.А.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Прогнозирование временных рядов</b>	<b>5</b>
<b>1.1 Временные ряды</b>	<b>5</b>
1.1.1 Прогнозирование временного ряда [1]	6
1.1.2 Главная особенность временных рядов	7
1.1.3 Компоненты временных рядов	7
<b>1.2 Автокорреляция</b>	<b>8</b>
1.2.1 Автокорреляция, её вычисление	9
1.2.2 Коррелограммы	10
1.2.2.1 График задержек	10
1.2.2.2 Матрица корреляций	11
1.2.2.3 Автокорреляционная функция	13
<b>2 Практическая часть</b>	<b>14</b>
<b>ЗАКЛЮЧЕНИЕ</b>	<b>15</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>16</b>
<b>ПРИЛОЖЕНИЕ А</b>	<b>17</b>
<b>ПРИЛОЖЕНИЕ Б</b>	<b>18</b>
<b>ПРИЛОЖЕНИЕ В</b>	<b>19</b>
<b>ПРИЛОЖЕНИЕ Г</b>	<b>27</b>

## **ВВЕДЕНИЕ**

Курсовая работа представляет собой вид учебной работы, рассчитанный на получение и закрепление необходимых навыков, развитие умения работать со сбором и анализом материала, полученного из изученных источников, а также написанию текста работы с соблюдением стандарта оформления ( в данной работе используется ГОСТ 7.32).

# 1 Прогнозирование временных рядов

## 1.1 Временные ряды

Временным рядом называется последовательность значений признака  $y$ , измеряемого через постоянные временные интервалы:

$$y_1, \dots, y_T, \dots, \quad y_t \in \mathbb{R}.$$

В этом определении нужно обратить внимание на то, что временные интервалы между измерениями признака постоянны.

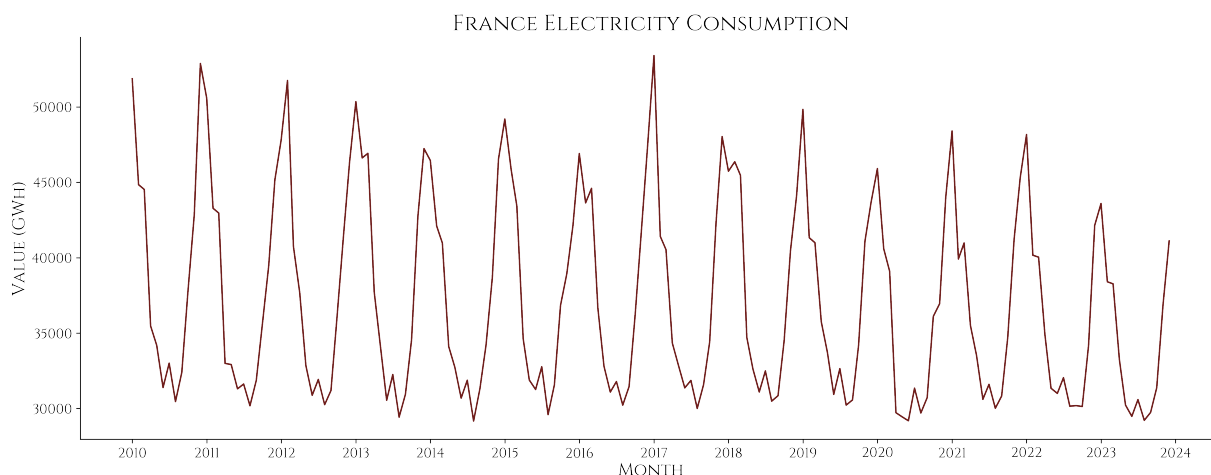


Рис. 1.1: Ежемесячное потребление электричества во Франции. Построено программой по адресу (листинг 1).

Временные ряды изобилуют в таких областях, как экономика, бизнес, инженерия, естественные науки (особенно геофизика и метеорология), а также социальные науки.

Примеры временных рядов - это ряды с ежемесячной последовательностью объемов отгруженных с завода товаров, еженедельными данными о количестве дорожно-транспортных происшествий, ежедневными объемами осадков, почасовыми наблюдениями за химическими выбросами. Ещё один пример временного ряда (показан на рисунке 1.1) — это реальное ежемесячное потребление электричества во Франции.

### 1.1.1 Прогнозирование временного ряда [1]

Информация о доступных наблюдениях в момент времени  $t$  временного ряда, используемая для предсказания его значения в некотором будущем  $t + l$  может предоставить фундамент для планирования в бизнесе и экономике, контроля продукции, оптимизации индустриальных процессов и так далее. Здесь  $l$  - это, так называемый, *горизонт прогнозирования*, который меняется от задачи к задаче.

Предположим, что наблюдения - *дискретные*, равноудаленные друг от друга величины. Например в задаче прогнозирования продаж, продажи  $z_t$  в текущий месяц  $t$  и продажи  $z_{t-1}, z_{t-2}, z_{t-3}, \dots$  в предыдущие месяцы можно использовать для прогнозирования горизонта в  $l = 1, 2, 3, \dots, 12$  месяцев. Обозначим за  $\hat{z}_t(l)$  прогноз, составленный в момент времени  $t$  для продаж  $z_{t+l}$  в некотором будущем  $t + l$ , т.е. с *горизонтом прогнозирования*  $l$ . Функция  $\hat{z}_t(l)$ , прогнозирующая в момент времени  $t$  для всех будущих горизонтов прогнозирования, на основе доступной информации о текущем и предыдущих значениях  $z_t, z_{t-1}, z_{t-2}, z_{t-3}, \dots$  на протяжении времени  $t$ , называется *функцией прогнозирования* в момент времени  $t$ . Основная задача прогнозирования временных рядов заключается в том, чтобы найти функцию прогнозирования с наименьшим средним квадратом отклонений  $z_{t+l} - \hat{z}_t(l)$  между истинными и спрогнозированными значениями для каждого из *горизонтов прогнозирования*  $l$ .

В добавок к нахождению наилучших прогнозов, также необходимо указать их точность, так чтобы, например, можно было рассчитать риски, ассоциированные с решениями, принятыми на их основе. Точность прогнозов можно выразить в качестве *доверительных интервалов* с каждой стороны прогноза. Эти интервалы могут быть рассчитаны для любого удобного набора вероятностей, например, 50 и 95%. Они указывают вероятность попадания спрогнозированной величины в данный интервал. За иллюстрацией обратимся к рис. 1.2, на котором изображены последние 20 значений временного ряда, кульминирующие в момент времени  $t$ . Также представлены прогнозы, сделанные в момент времени  $t$  с горизонтами прогнозирования  $l = 1, 2, \dots, 13$ , вместе с 50% доверительными интервалами.

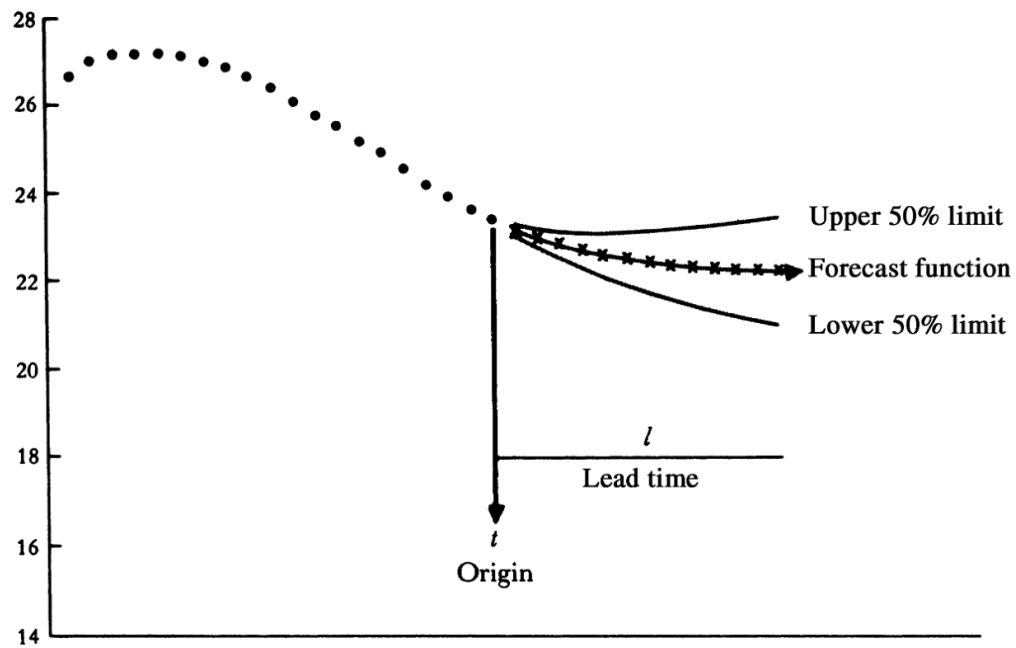


Рис. 1.2: Значения временного ряда с прогнозирующей функцией и 50% доверительными интервалами.

### 1.1.2 Главная особенность временных рядов

Часто, в задачах регрессионного анализа и машинного обучения, в качестве анализируемых данных берутся простые выборки, построенные из независимо одинаково распределенных наблюдений. В задаче анализа временных рядов всё с точностью наоборот: предполагается, что данные в прошлом каким-то образом связаны с данными в будущем. Чем сильнее они связаны, тем больше имеется информации о поведении временного ряда в будущем и тем точнее можно сделать прогноз.

Полезно снова рассмотреть данные о реальном ежемесячном потреблении электричества во Франции (рис. 1.1). Видно, что на графике изображена не простая выборка (измерения не являются независимыми и одинаково распределёнными), а сложный, структурированный процесс. Выявив структуру этого процесса, можно учесть её в прогнозирующей модели и построить действительно точный прогноз.

### 1.1.3 Компоненты временных рядов

Полезно рассмотреть несколько понятий, которыми можно описать поведение временных рядов:

- Тренд — плавное долгосрочное изменение уровня ряда. Эту характе-

ристику можно получить, наблюдая ряд в течение достаточно долгого времени.

- Сезонность — циклические изменения уровня ряда с постоянным периодом. В данных о ежемесячном потреблении электричества во Франции (рис. 1.1) очень хорошо видны подобные сезонные колебания: признак всегда принимает максимальное значение зимой, а минимальное — летом. Это легко объяснить тем, что летом электричества необходимо меньше всего, это самый тёплый сезон во Франции. В целом профиль изменения потребления электричества внутри года остаётся более-менее постоянным.
- Цикл — изменение уровня ряда с переменным периодом. Такое поведение часто встречается в рядах, связанных с продажами, и объясняется циклическими изменениями экономической активности. В экономике выделяют циклы длиной 4 - 5 лет, 7 - 11 лет, 45 - 50 лет и т. д. Другой пример ряда с такой характеристикой — это солнечная активность, которая соответствует, например, количеству солнечных пятен за день. Она плавно меняется с периодом, который составляет несколько лет, причём сам период также меняется во времени.
- Ошибка — непрогнозируемая случайная компонента ряда. Сюда включены все те характеристики временного ряда, которые сложно измерить (например, слишком слабые).

## 1.2 Автокорреляция

Одной из важнейших характеристик временного ряда является автокорреляция. Автокорреляцией называется математическая репрезентация степени «схожести» между исходным рядом и его версией, сдвинутой на некоторый интервал, называемый лагом. Концептуально это похоже на корреляцию Пирсона между двумя временными рядами, только автокорреляция рассматривает один и тот же ряд дважды.

Стоит напомнить, что коэффициент корреляции, а значит и автокорреляции может быть равен 0 при наличии сильной, но нелинейной



зависимости (рис. 1.3) [2].

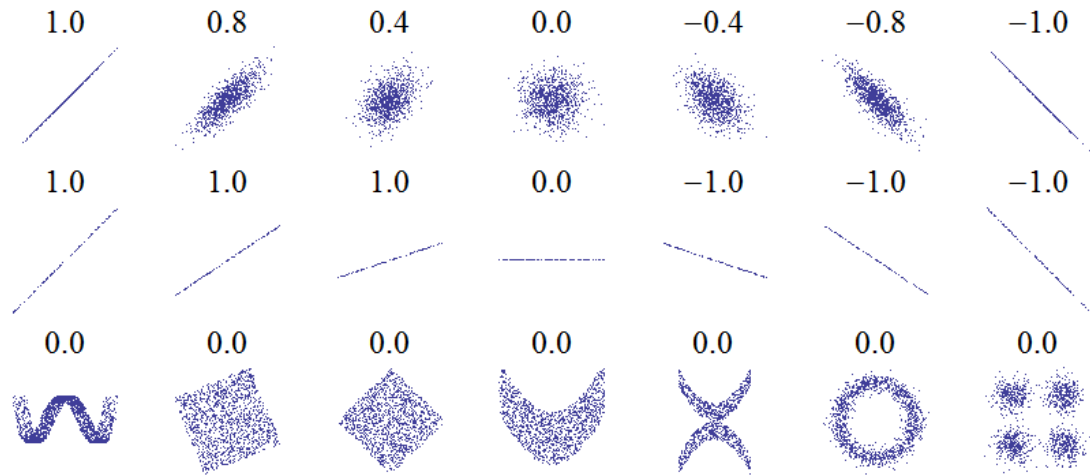


Рис. 1.3: Несколько наборов точек  $(x, y)$  и коэффициент корреляции  $x$  и  $y$  для каждого набора. Отметим, что корреляция отражает зашумленность и направление линейной зависимости (верхний ряд), но не отражает ее угловой коэффициент (средний ряд), а также многие аспекты нелинейных связей (нижний ряд). (Примечание: на рисунке в центре угловой коэффициент равен 0, но в этом случае коэффициент корреляции не определен, потому что дисперсия  $Y$  нулевая.)

### 1.2.1 Автокорреляция, её вычисление

Количественной характеристикой сходства между значениями ряда в соседних точках является автокорреляционная функция (или просто автокорреляция), которая задаётся следующим соотношением:

$$r_l \stackrel{\text{def}}{=} \frac{\mathbb{E}[(y_t - \mathbb{E}_y)(y_{t+l} - \mathbb{E}_y)]}{\mathbb{V}_y},$$

где количество отсчётов, на которое сдвинут ряд, называется лагом автокорреляции ( $l$ ).

Можно заметить сходство с корреляцией Пирсона:

$$\rho \stackrel{\text{def}}{=} \text{corr}[X, Y] \stackrel{\text{def}}{=} \frac{\text{Cov}[X, Y]}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}}, \quad \text{Cov}[X, Y] \stackrel{\text{def}}{=} \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Значения, принимаемые автокорреляцией такие же, как и у коэффициента Пирсона:  $r_l \in [-1, 1]$ . Вычислить автокорреляцию по выборке можно,

заменяя в формуле математическое ожидание на выборочное среднее, а дисперсию — на выборочную дисперсию.

### 1.2.2 Коррелограммы

Анализировать величину автокорреляции при разных значениях лагов удобно с помощью графиков. Далее различные методы графического анализа автокорреляции будут демонстрироваться на примере данных о суммарном объёме продаж красного вина в Австралии за месяц на протяжении 15 лет (рис. 1.4).



Рис. 1.4: Месячный объём продаж красного вина в Австралии, в бутылках. Построено программой по адресу (листинг 2).

Этот ряд обладает ярко выраженной годовой сезонностью: максимум продаж за год приходится на декабрь, а затем, в январе, происходит существенное падение.

#### 1.2.2.1 График задержек

Графиком задержек (Lag Plot) называется особый вид диаграммы рассеяния (scatter plot), на которой по одной из ее осей откладывается значение временного ряда в момент времени  $t$ , а по другой - значение в момент времени  $t + l$ , где  $l$  - значение лага.

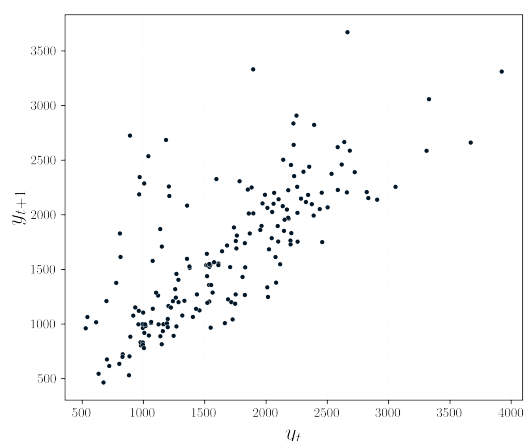


Рис. 1.5: Связь между значениями объёма продаж красного вина в соседние месяцы, по горизонтали отложен объём продаж в месяц  $t$ , по вертикали — в следующий месяц,  $t + 1$ . Построено программой по адресу (листинг 3).

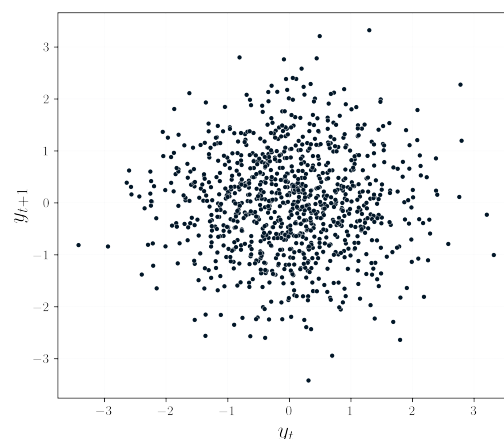


Рис. 1.6: График задержек случайных величин, взятых из нормального распределения с лагом  $l = 1$ . Построено программой по адресу (листинг 4).

График задержек помогает понять являются ли значения во временном ряду случайными. Если данные случайны, то на графике мы не увидим никакой определенной структуры. Однако если же данные не случайны, то на графике задержек можно будет увидеть ярко выраженную форму.

Например на рисунке 1.5 видно, что большая часть точек сгруппирована вокруг главной диагонали, что говорит о схожести продаж в соседние месяцы. В свою очередь на рисунке 1.6 нет никакой ярко выраженной структуры, что логично, так как это просто белый шум.

### 1.2.2.2 Матрица корреляций

Еще одним широко распространенным методом графического анализа автокорреляции временного ряда является матрица корреляций (корреляционная матрица). Она отображает в себе корреляцию между всеми возможными заданными парами значений.

Рассмотрим матрицу корреляций на примере еженедельного общего объема продаж авокадо за 3 года (рис. 1.7).

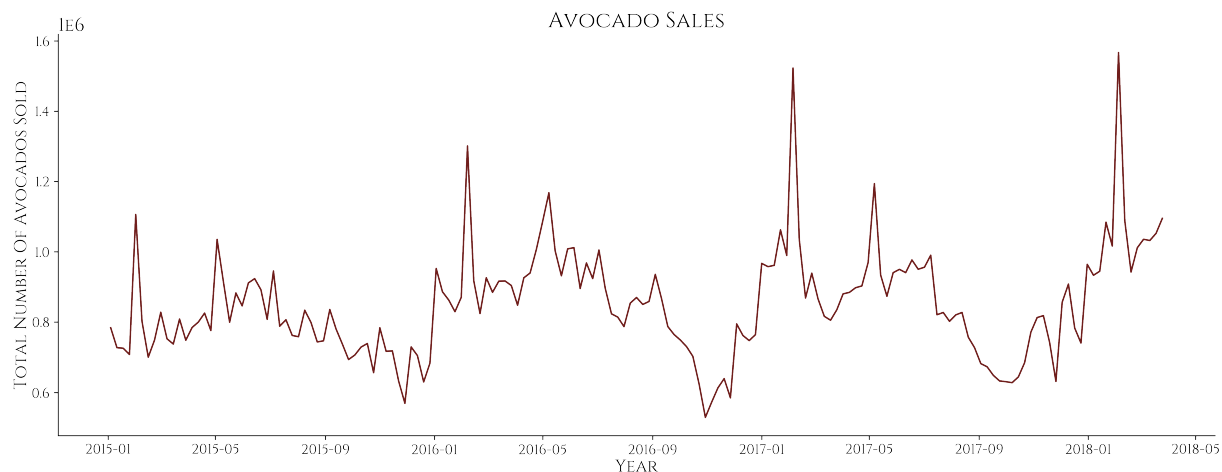


Рис. 1.7: Еженедельный объем продаж авокадо. Построено программой по адресу (листинг 5).

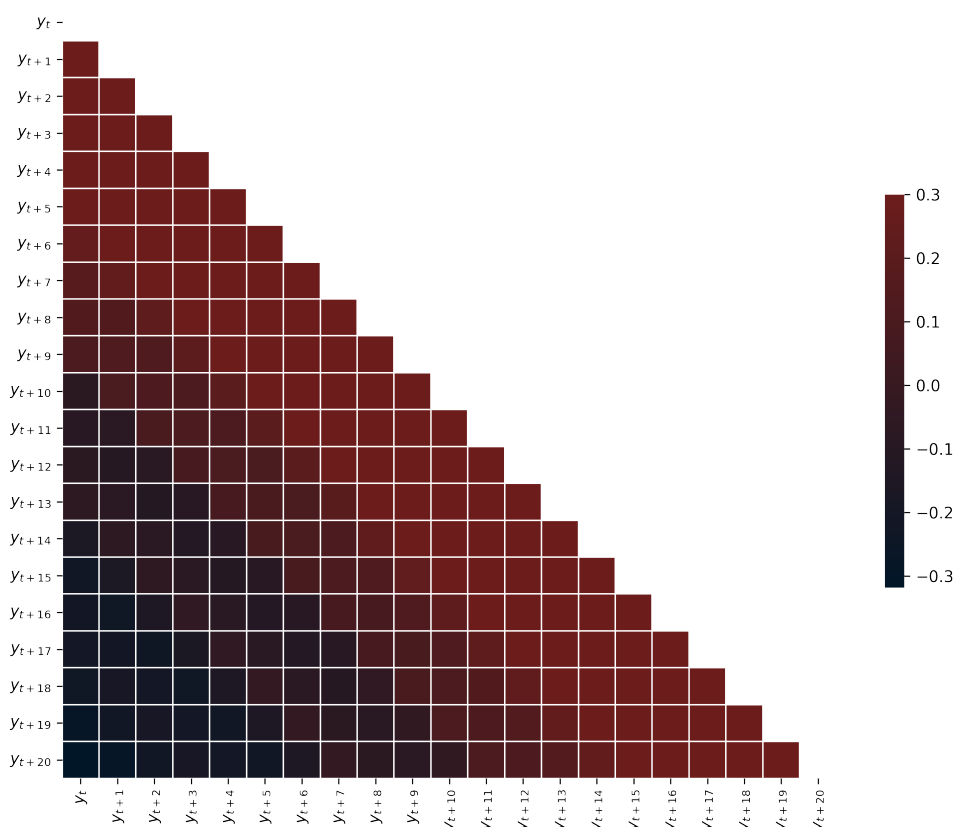


Рис. 1.8: Диагональная корреляционная матрица объема продаж авокадо. Построено программой по адресу (листинг 6).

На рисунке 1.8 видно, что данные тем сильнее скоррелированы, чем ближе они находятся. Так например значения в соседних точках временного ряда  $y_t$  и  $y_{t+1}$  имеют корреляцию равную 0.3, а  $y_t$  и  $y_{t+20}$ , в свою очередь:

−0.3.

### 1.2.2.3 Автокорреляционная функция

График автокорреляции временного ряда от лага называется автокорреляционной функцией (ACF). Такой график также часто называют коррелограммой. По оси ординат на нём откладывается автокорреляция, а по оси абсцисс — размер лага  $l$ .

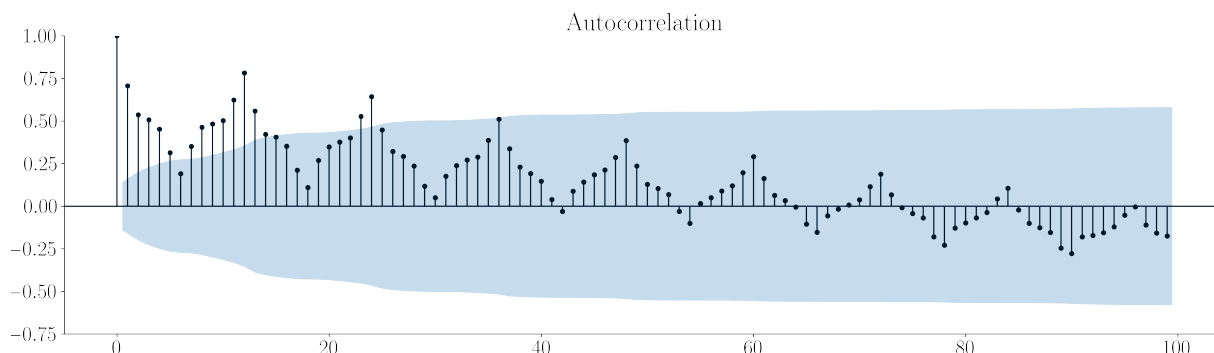


Рис. 1.9: Коррелограмма для объёма продаж красного вина в Австралии. Построено программой по адресу (листинг 7).

На рисунке 1.9 показан пример коррелограммы для исследуемых ранее данных о месячных продажах красного вина в Австралии (рисунок 1.4). На графике видно, что автокорреляция принимает большие значения в лагах, кратных сезонному периоду. Такой вид коррелограммы типичен для данных с выраженной сезонностью.

Стоит отметить, что автокорреляция может начать колебаться вокруг горизонтальной оси, соответствующей её нулевому значению, как это и произошло на рисунке 1.9, а также, что автокорреляция в точке с лагом  $l = 0$  всегда равна единице, так как в ней вычисляется корреляция значения ряда с самим собой.

Также на рисунке 1.9 изображён синий коридор вокруг горизонтальной оси. Это коридор значимости отличия корреляции от нуля. Фактически, все автокорреляции, которые изображены вне этого коридора, значимо отличаются от нуля.

## **2 Практическая часть**

## **ЗАКЛЮЧЕНИЕ**

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Gregory C. Reinsel George E. P. Box, Gwilym M. Jenkins and Greta M. Ljung. Time Series Analysis: Forecasting and Control. Wiley, fifth edition edition, 2015.
- [2] Kevin P. Murphy. Probabilistic Machine Learning: An introduction. MIT Press, 2022.



## **ПРИЛОЖЕНИЕ А**

## **ПРИЛОЖЕНИЕ Б**

## ПРИЛОЖЕНИЕ В

Программный код, используемый в данной работе.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import matplotlib.font_manager as fm
from matplotlib.colors import LinearSegmentedColormap

import seaborn as sns

import statsmodels as sm

font_path = '../extra/Cinzel-VariableFont_wght.ttf'

cinzel_font = fm.FontProperties(fname=font_path)
fm.fontManager.addfont(font_path)

RED = '#6F1D1B'
RICH_BLACK = '#011627'

plt.rcParams.update({
    "text.use_tex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern Roman"],
})

def decorate_plot(ax, xname, yname, loc):
    SIZE_TICKS = 12

    # Eliminate upper and right axes
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')

    # Show ticks in the left and lower axes only
    ax.xaxis.set_ticks_position('bottom')
```

```

ax.yaxis.set_ticks_position('left')

# x axis name
ax.set_xlabel(xname, fontsize=15)

# y axis name
ax.set_ylabel(yname, fontsize=15)

# Adjust the font size of the tick labels
ax.tick_params(axis='both', which='major', labelsize=SIZE_TICKS)

if loc:
    plt.legend(fontsize=10, loc=loc)

# Update font settings
plt.rcParams.update({
    "font.family": cinzel_font.get_name(),
    "font.size": 16
})

# Adjust layout
plt.tight_layout()

def decorate_lag_plot(ax, xname, yname, loc):
    SIZE_TICKS = 12

    # x and y axis labels
    ax.set_xlabel(xname, fontsize=20)
    ax.set_ylabel(yname, fontsize=20)

    # Adjust tick labels size
    ax.tick_params(axis='both', which='major', labelsize=SIZE_TICKS)

    if loc:
        ax.legend(fontsize=10, loc=loc)

    plt.grid(True, linestyle='--', linewidth=0.05)
    plt.tight_layout()

def decorate_acf(ax, loc):

```

```

SIZE_TICKS = 24

# Eliminate upper and right axes
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

# Show ticks in the left and lower axes only
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')

# Adjust the font size of the tick labels
ax.tick_params(axis='both', which='major', labelsize=SIZE_TICKS)

if loc:
    plt.legend(fontsize=10, loc=loc)

# Update font settings
plt.rcParams.update({
    "font.family": cinzel_font.get_name(),
    "font.size": 24
})

# Adjust layout
plt.tight_layout()

def lag_plot(df, value_name, l=1, save=''):
    lag_df = pd.DataFrame(df[value_name])
    lag_df['lag'] = df[value_name].shift(l)

    lag_df.dropna(inplace=True)

    _, ax = plt.subplots(figsize=(7, 6))

    decorate_lag_plot(ax, r'$y_t$', f'$y_{\{t + \{l\}\}}$', '')

    sns.scatterplot(data=lag_df,
                    x=value_name,
                    y="lag",
                    color=RICH_BLACK,
                    size=100,

```

```

        legend=False,
        ax=ax)

if save:
    plt.savefig(f'../images/{save}.png', dpi=300, transparent=True)

plt.show()

def corr_matrix_plot(df, value_name, n=10, save=''):
    lag_df = pd.DataFrame(df[value_name]).rename(columns={value_name: '$y_t$'})
    for l in range(1, n+1):
        lag_df[f'$y_{{t + {l}}}$'] = df[value_name].shift(1)

    lag_df.dropna(inplace=True)

    # Compute the correlation matrix
    corr = lag_df.corr()

    # Generate a mask for the upper triangle
    mask = np.triu(np.ones_like(corr, dtype=bool))

    # Set up the matplotlib figure
    _, _ = plt.subplots(figsize=(11, 9))

    # Generate a custom diverging colormap
    cmap = LinearSegmentedColormap.from_list('custom_cmap', [RICH_BLACK, RED])

    # Draw the heatmap with the mask and correct aspect ratio
    sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5})

    if save:
        plt.savefig(f'../images/{save}.png', dpi=300, transparent=True)

    plt.show()

```

Листинг 1: Ежемесячное потребление электричества во Франции.

```

df = pd.read_csv('data/global_electricity_production_data.csv')

France_df = df[(df['country_name'] == 'France') &

```

```

(df['product'] == 'Electricity') &
(df['parameter'] == 'Final Consumption (Calculated)')].copy()

start_date = pd.to_datetime('01/01/2010', format='%m/%d/%Y')
end_date    = pd.to_datetime('12/01/2023', format='%m/%d/%Y')

France_df['date'] = pd.to_datetime(df['date'], format='%m/%d/%Y')

France_df = France_df[(France_df['date'] >= start_date) &
                      (France_df['date'] <= end_date)]

France_df.sort_values('date', inplace=True)

# Create the plot
fig, ax = plt.subplots(figsize=(15, 6))
ax.plot(France_df['date'], France_df['value'], linestyle='-', color=RED)

ax.set_title('France Electricity Consumption')

decorate_plot(ax, 'Month', 'Value (GWh)', '')

year_locator = mdates.YearLocator()
year_formatter = mdates.DateFormatter('%Y')
ax.xaxis.set_major_locator(year_locator)
ax.xaxis.set_major_formatter(year_formatter)

plt.savefig(f'../images/time_series_example_France.png',
            dpi=300,
            transparent=True)

plt.show()

```

Листинг 2: Месячный объём продаж красного вина в Австралии.

```

wine_df = pd.read_csv('../data/AusWineSales.csv')

wine_df = wine_df[['YearMonth', 'Red']].copy()
wine_df['YearMonth'] = pd.to_datetime(wine_df['YearMonth'], format='%Y-%m')

wine_df.sort_values('YearMonth', inplace=True)

```

```

# Create the plot
fig, ax = plt.subplots(figsize=(15, 6))
ax.plot(wine_df['YearMonth'], wine_df['Red'], linestyle='--', color=RED)

ax.set_title('Australian Wine Sales')

decorate_plot(ax, 'Year', 'Wine Sales (Volume)', '')

plt.savefig(f'../images/time_series_example_wine.png',
            dpi=300,
            transparent=True)

plt.show()

```

Листинг 3: График задержек продаж вина в Австралии.

```

wine_df = pd.read_csv('../data/AusWineSales.csv')

wine_df = wine_df[['YearMonth', 'Red']].copy()
wine_df['YearMonth'] = pd.to_datetime(wine_df['YearMonth'], format='%Y-%m')

wine_df.sort_values('YearMonth', inplace=True)

lag_plot(wine_df, 'Red', 1, 'australia_wine_lag_plot')

```

Листинг 4: График задержек белого шума.

```

sample_size=1000
random_series = pd.DataFrame(np.random.normal(size=sample_size))
lag_plot(random_series, 0, 1=1, save='random_series_lag_plot')

```

Листинг 5: Еженедельный объём продаж авокадо

```

avocado_df = pd.read_csv('../data/avocado.csv')

avocado_df = avocado_df.groupby('Date')['Total Volume'].mean().reset_index()

avocado_df['Date'] = pd.to_datetime(avocado_df['Date'], format='%Y-%m-%d')

avocado_df.sort_values('Date', inplace=True)

```



```

fig, ax = plt.subplots(figsize=(15, 6))
ax.plot(avocado_df['Date'], avocado_df['Total Volume'], linestyle='-', color=RED)

ax.set_title('Avocado Sales')

decorate_plot(ax, 'Year', 'Total Number Of Avocados Sold', '')

plt.savefig(f'../images/time_series_example_avocado.png', dpi=300, transparent=True)

plt.show()

```

Листинг 6: Диагональная корреляционная матрица объема продаж авокадо.

```

avocado_df = pd.read_csv('../data/avocado.csv')

avocado_df = avocado_df.groupby('Date')['Total Volume'].mean().reset_index()

avocado_df['Date'] = pd.to_datetime(avocado_df['Date'], format='%Y-%m-%d')

avocado_df.sort_values('Date', inplace=True)

corr_matrix_plot(avocado_df, 'Total Volume', 20, save='correlation_matrix_avocado')

```

Листинг 7: Коррелограмма для объёма продаж красного вина в Австралии.

```

wine_df = pd.read_csv('../data/AusWineSales.csv')

wine_df = wine_df[['YearMonth', 'Red']].copy()
wine_df['YearMonth'] = pd.to_datetime(wine_df['YearMonth'], format='%Y-%m')

wine_df.sort_values('YearMonth', inplace=True)

_, ax = plt.subplots(figsize=(20, 6))

decorate_acf(ax, '')

sm.graphics.tsaplots.plot_acf(wine_df['Red'],
                               lags=np.arange(0, 100, 1),
                               ax=ax,
                               color=RICH_BLACK,
                               vlines_kwargs={'colors': RICH_BLACK,

```

```
                                'linewidth': 1.5});  
  
ax.set_ylim(-0.75, 1)  
ax.grid(False)  
  
plt.savefig(f'../images/acf_wine.png', dpi=300, transparent=True)
```

## **ПРИЛОЖЕНИЕ Г**