Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: ТПпУБД

Отчёт по лабораторной работе на тему

**Обработка данных курсов валют**

Выполнил:

Магистрант гр.

956241 Шуба И.А.

Проверил:

Доцент, кандидат технических наук

Стержанов М.В.

Минск 2020

**СОДЕРЖАНИЕ**

[1. Постановка задачи 2](#_Toc40817091)

[2. Получение данных 2](#_Toc40817092)

[3. Работа с АPI 3](#_Toc40817093)

[4. Функционал программы. 5](#_Toc40817094)

[5. Анализ данных 9](#_Toc40817095)

[Вывод 15](#_Toc40817096)

## Постановка задачи

Целью данного проекта является обработка данных курсов валют национального банка НБ РБ. Для достижения данной цели были поставлены следующие задачи:

* Реализация сбора данных;
* Организация хранения и доступа к собранным данным;
* Построение графов курсов валют.

## Получение данных

Официальный курс белорусского рубля по отношению к доллару США, евро и российскому рублю устанавливается по данным биржевых торгов в режиме торгов с установлением фиксинга или как средневзвешенный курс по сделкам, совершенным на биржевых торгах в режиме непрерывного двойного аукциона, проводимых [открытым акционерным обществом "Белорусская валютно-фондовая биржа"](http://www.bcse.by/) по доллару США, евро и российскому рублю. Национальный банк РБ, предоставляет доступ к своему сайту посредством API. Это очень удобно, поскольку нет необходимости скачивать HTML страницы и парсить их. Вместо этого мы можем получить доступ к информации сразу в подходящем формате, а именно в виде JSON файла.

## Работа с АPI

Национальный банк РБ предоставляет API для получения данных официального курса белорусского рубля по отношению к иностранным валютам, устанавливаемого Национальным банком Республики Беларусь:

Полный перечень иностранных валют, по отношению к которым Национальным банком устанавливается официальный курс белорусского рубля.

Официальный курс белорусского рубля по отношению к иностранным валютам, устанавливаемый Национальным банком на конкретную дату.

Получение динамики официального курса белорусского рубля по отношению к заданной иностранной валюте (не более чем за 365 дней).

API — описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой. Обычно входит в описание какого-либо интернет-протокола, программного каркаса или стандарта вызовов функций операционной системы.

Ниже описан API получения официального курса белорусского рубля по отношению к иностранным валютам, устанавливаемого Национальным банком Республики Беларусь.

Описание запроса для получения полного перечня иностранных валют, по отношению к которым Национальным банком устанавливается официальный курс белорусского рубля:

**Адрес запроса:** http://www.nbrb.by/API/ExRates/Currencies[/{Cur\_ID}]

**Результат:** массив объектов класса Currency. Если указан Cur\_ID, то возвращается один объект Currency.

Класс **Currency** содержит следующие свойства:

* Cur\_ID – внутренний код
* Cur\_ParentID – этот код используется для связи, при изменениях наименования, количества единиц к которому устанавливается курс белорусского рубля, буквенного, цифрового кодов и т.д. фактически одной и той же валюты\*.
* Cur\_Code – цифровой код
* Cur\_Abbreviation – буквенный код
* Cur\_Name – наименование валюты на русском языке
* Cur\_Name\_Bel – наименование на белорусском языке
* Cur\_Name\_Eng – наименование на английском языке
* Cur\_QuotName – наименование валюты на русском языке, содержащее количество единиц
* Cur\_QuotName\_Bel – наименование на белорусском языке, содержащее количество единиц
* Cur\_QuotName\_Eng – наименование на английском языке, содержащее количество единиц
* Cur\_NameMulti – наименование валюты на русском языке во множественном числе
* Cur\_Name\_BelMulti – наименование валюты на белорусском языке во множественном числе\*
* Cur\_Name\_EngMulti – наименование на английском языке во множественном числе\*
* Cur\_Scale – количество единиц иностранной валюты
* Cur\_Periodicity – периодичность установления курса (0 – ежедневно, 1 – ежемесячно)
* Cur\_DateStart – дата включения валюты в перечень валют, к которым устанавливается официальный курс бел. рубля
* Cur\_DateEnd – дата исключения валюты из перечня валют, к которым устанавливается официальный курс бел. рубля

Описание запроса для получения официального курса белорусского рубля по отношению к иностранным валютам, устанавливаемый Национальным банком на конкретную дату:

**Адрес запроса:** http://www.nbrb.by/API/ExRates/Rates[/{Cur\_ID}]

**Параметры (GET):**

* onDate\*\* – дата, на которую запрашивается курс (если не задана, то возвращается курс на сегодня)
* Periodicity – периодичность установления курса (0 – ежедневно, 1 – ежемесячно)
* ParamMode – формат аргумента Cur\_ID: 0 – внутренний код валюты, 1 – трехзначный цифровой  код валюты в соответствии со стандартом ИСО 4217, 2 – трехзначный буквенный код валюты (ИСО 4217). По умолчанию = 0

При использовании буквенного или цифрового кода валюты (ИСО 4217) учитывайте его значение на запрашиваемую дату.

**Результат:** массив объектов класса Rate. Если указан Cur\_ID, то возвращается один объект Rate.

Класс **Rate** содержит следующие свойства:

* Cur\_ID – внутренний код
* Date – дата, на которую запрашивается курс
* Cur\_Abbreviation – буквенный код
* Cur\_Scale – количество единиц иностранной валюты
* Cur\_Name – наименование валюты на русском языке во множественном, либо в единственном числе, в зависимости от количества единиц
* Cur\_OfficialRate – курс\*

Описание запроса для получения динамики официального курса белорусского рубля по отношению к заданной иностранной валюте (не более чем за 365 дней):

**Адрес запроса:** http://www.nbrb.by/API/Rates/Dynamics/{Cur\_ID}

**Параметры (GET):**

* startDate\*\* – дата начала запрашиваемого периода
* endDate\*\* – дата окончания запрашиваемого периода

**Результат:** массив объектов класса RateShort.

Класс **RateShort** содержит следующие свойства:

* Cur\_ID – внутренний код
* Date – дата, на которую запрашивается курс
* Cur\_OfficialRate – курс\*

**Примечание.** Если в любой из перечисленных методов передается неверный код валюты, то возвращается 404 HTTP статус-код. Если же на запрашиваемую дату курс не установлен, то возвращается пустой массив.

## Функционал программы.

После запуска программы, первым делом начинается валидация переданных скрипту параметров (требуемые параметры: --currency, --start\_date, --end\_date). Инициализация и валидация аргументов выполняется при помощи библиотеки argparse. Для этого вызывается метод validate\_args() модуля utils. В этом методе инициализируется объект класса ArgumentParser(), у которого в свою очередь есть метод add\_argument(\*args, \*\*kwargs), который и используется для объявления и описания всех возможных аргументов, которые может принять скрипт:

arg\_parser.add\_argument(

"-c", "--currency", type=str, required=True,

help="Код валюты для сбора данных. (USD, EUR, RUB)"

)

arg\_parser.add\_argument(

"--start\_date", type=valid\_date, required=True,

help="Дата начала периода для аналитики, в формате дд-мм-гггг",

)

arg\_parser.add\_argument(

"--end\_date", type=valid\_date, required=True,

help="Дата конца периода для аналитики, в формате дд-мм-гггг"

)

Так, передавая в этот метод специальные аргументы, можно объявить, как именно будут приниматься аргументы на входе в скрипт. Первые два позиционных аргумента метода add\_argument() – это короткое имя аргумента, которое нужно указать перед его значением в консоли при запуске скрипта, а также полное. Например, для аргумента c именем валюты указываем -c и –currency. Таким образом, для того, чтобы указать скрипту имя валюты:

python main.py -c USD или python main.py –currency=USD

Остальные используемые именованные аргументы, которые используются для метода add\_argument():

* type: тип принимаемого агумента
* help: описание аргумента, которое используется при вызове скрипта с флагом –help
* required: если True, аргумент считается обязательным при валидации.

Стоит отметить, что параметром type можно указать не только стандартные типы объектов python, но и любую функцию-валидатор, в которую передается пришедший в скрипт аргумент. Эта функция должна выполнить проверку аргумента на валидность, а затем вернуть его, либо же выбросить исключение argparse.ArgumentParseError(). В коде программы это используется для валидации параметров –start\_date и –end\_date:

def valid\_date(date: str):

try:

return datetime.strptime(date, DEFAULT\_DATEFORMAT)

except ValueError:

msg = f"Введена неккоректная дата: '{date}'."

raise argparse.ArgumentTypeError(msg)

Валидация описанных аргументов начинается с вызовом функции args =argparser.parse\_args(), которая возвращает отвалидированные аргументы, или выбрасывает исключение, описанное выше. Таким образом, переданные в скрипт аргументы currency, start\_date, end\_date возвращаются в модуль main.

Далее происходит генерация имени файла для сохранения и чтения данных. Это происходит в методе generate\_filename(currency, start\_date, end\_date). Генерация имени файла производится путем форматирования с конкатенацией аргументов currency, start\_date, end\_date.

После этого производится проверка на то, существует ли уже файл. Если же файл с таким именем существует, значит данные для анализа по конкретно этой валюте за конкретный промежуток времени уже были ранее сохранены. Так как данные за прошедший промежуток времени статичны и не могут поменяться со временем – выполнять запрос к API НБРБ было бы избыточным решением, поэтому можно считать их из ранее сохраненного файла.

Если файл с данными не найден, необходимо выполнить запрос к API НБРБ. Для этого используется модуль nbrb\_api и его метод get\_rates(currency, start\_date, end\_date). В этом модуле происходит составление и отправка GET запроса к API НБРБ. Для отправки запроса используется requests.Session(). Иногда API НБРБ может возвращать ошибки 5xx по тем или иным причинам на валидные запросы, поэтому необходимо было реализовать механизм переотправки запросов к API в случае ошибок несколько раз, прежде чем выбросить исключение. В программе была реализована функция-конструктор сессии с описанным механизмом переотправки при помощи requests.adapters.HTTPAdapter и urllib3.util.retry.Retry. Таким образом, запрос к API в коде программы выполняется при помощи сессии, полученной при помощи следующей функции-конструктора:

def construct\_session():

session = requests.Session()

retries = Retry(total=5,

backoff\_factor=0.1,

status\_forcelist=[500, 502, 503, 504])

session.mount("http://", HTTPAdapter(max\_retries=retries))

return session

В этом же модуле был объявлен класс-наследник базового класс python Exception:

class NBRBApiException(Exception):

def \_\_init\_\_(self, \*args):

self.message = args[0] if args else None

def \_\_str\_\_(self):

return self.message or "Error while working with NBRB API."

Исключения этого типа будут выброшены при:

* попытке запросить данные из API по валюте, которой нет в списке разрешенных валют;
* неудачно завершенной серии запросов после нескольких попыток переотправки с политикой Retry;
* получении ответа от API со статус-кодов не равным 200 (ОК);

Эти исключения перехватываются в модуле main, и в таком случае выполнение программы завершается со статусом 1.

Если же исключений не возникает, get\_rates() преобразует ответ от API в словарь python при помощи response.json() и вернёт результат в модуль main.

После получения данных от API, в модуле main происходит сохранение данных в файл при помощи помощи библиотеки pandas, которая помогает словарь python в формат для записи csv-файла.

После того как данные получены и сохранены, начинается обработка данных и построение графиков. Графики строятся благодаря библиотеке pyplot. В модуле plot\_graphs описаны функции для построения графиков и гистограмм. Каждая из таких функций обёрнута в декоратор, который помогает добавить к графику общие сведения, такие как надписи на осях, заголовок, а также показать график на экран при помощи метода plt.show() после создания. Наличие этого декоратора позволяет соблюдать правило хорошего тона DRY (Don’t Repeat Yourself), описав общую логику приложения в одном месте и пере используя её:

def plot\_constructor(func):

@wraps(func)

def wrapper(\*args, title=None, xlabel=None, ylabel=None, \*\*kw):

func(\*args, \*\*kwargs)

plt.title(title)

plt.legend()

plt.ylabel(ylabel)

plt.xlabel(xlabel)

plt.show()

return wrapper

Таким образом, для обёртки функции построения графика этим конструктором:

@plot\_constructor

def plot\_changes(df, label):

plt.plot((df["rate"] - df["rate"].shift(1))[1:], "-", linewidth=1, label=label)

plt.xticks(shorten\_dates(df.index.values), rotation=90)

* label - подпись к графику.
* plt.plot(x, y, "-", linewidth=1, label=label) с помощью matplotlib рисуется график, где "-" - соединить точки линией, linewidth - ширина линии.
* plt.xticks(rotation=90) - повернуть надписи для оси x.
* plt.xticks(shorten\_dates(x)) – сокращение количества отметок на оси x при помощи функции shorten\_dates(), которая оставляет в списке только каждое пятое значение, если их количество больше 31.
* plt.legend()  - добавляет надпись
* plt.show() - выводит на экран график.

## Анализ данных

Для построения графиков был использован инструмент matplotlib.pyplot.plot. На рисунке 1 представлена официальный курс белорусского рубля по отношению к евро, устанавливаемый Национальным банком Республики Беларусь за месяц.

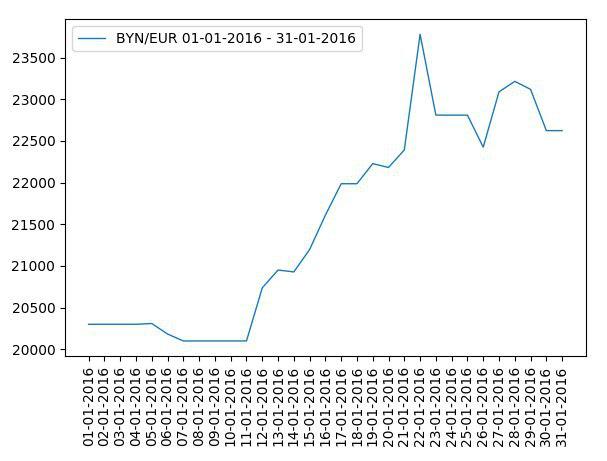


Рисунок 1 - Курс белорусского рубля по отношению к евро за месяц

На рисунке 2 представлена официальный курс белорусского рубля по отношению к евро, устанавливаемый Национальным банком Республики Беларусь за месяц.

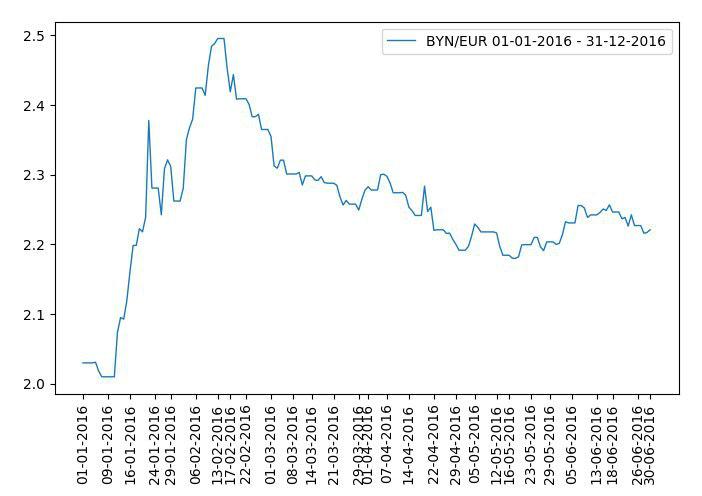


Рисунок 2 - Курс белорусского рубля по отношению к евро за выбранный период

На рисунке 3 представлена официальный курс белорусского рубля по отношению к евро, устанавливаемый Национальным банком Республики Беларусь за месяц.

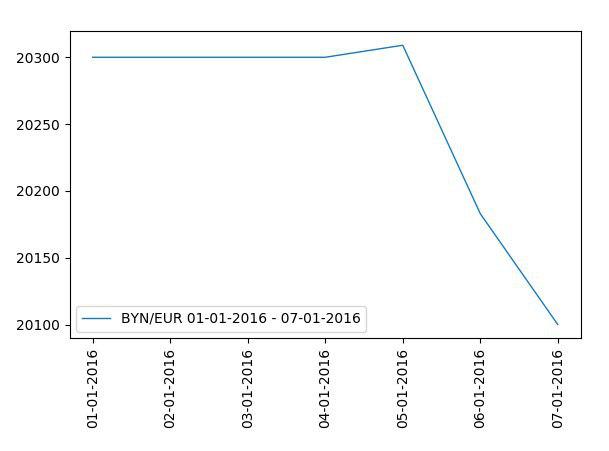


Рисунок 3 - Курс белорусского рубля по отношению к евро за полгода

На рисунке 4 представлена официальный курс белорусского рубля по отношению к евро, устанавливаемый Национальным банком Республики Беларусь за месяц.

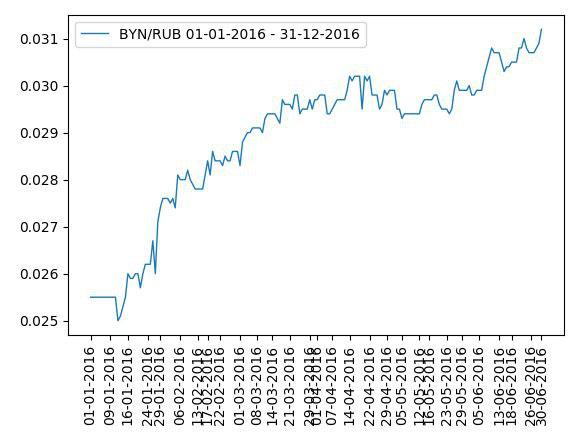


Рисунок 4 - Курс белорусского рубля по отношению к российскому

рублю за год

Рассмотрим первичный анализ данных на примере данных курса доллара к белорусскому рублю в период с 1 января 2019 по 31 декабря 2019.

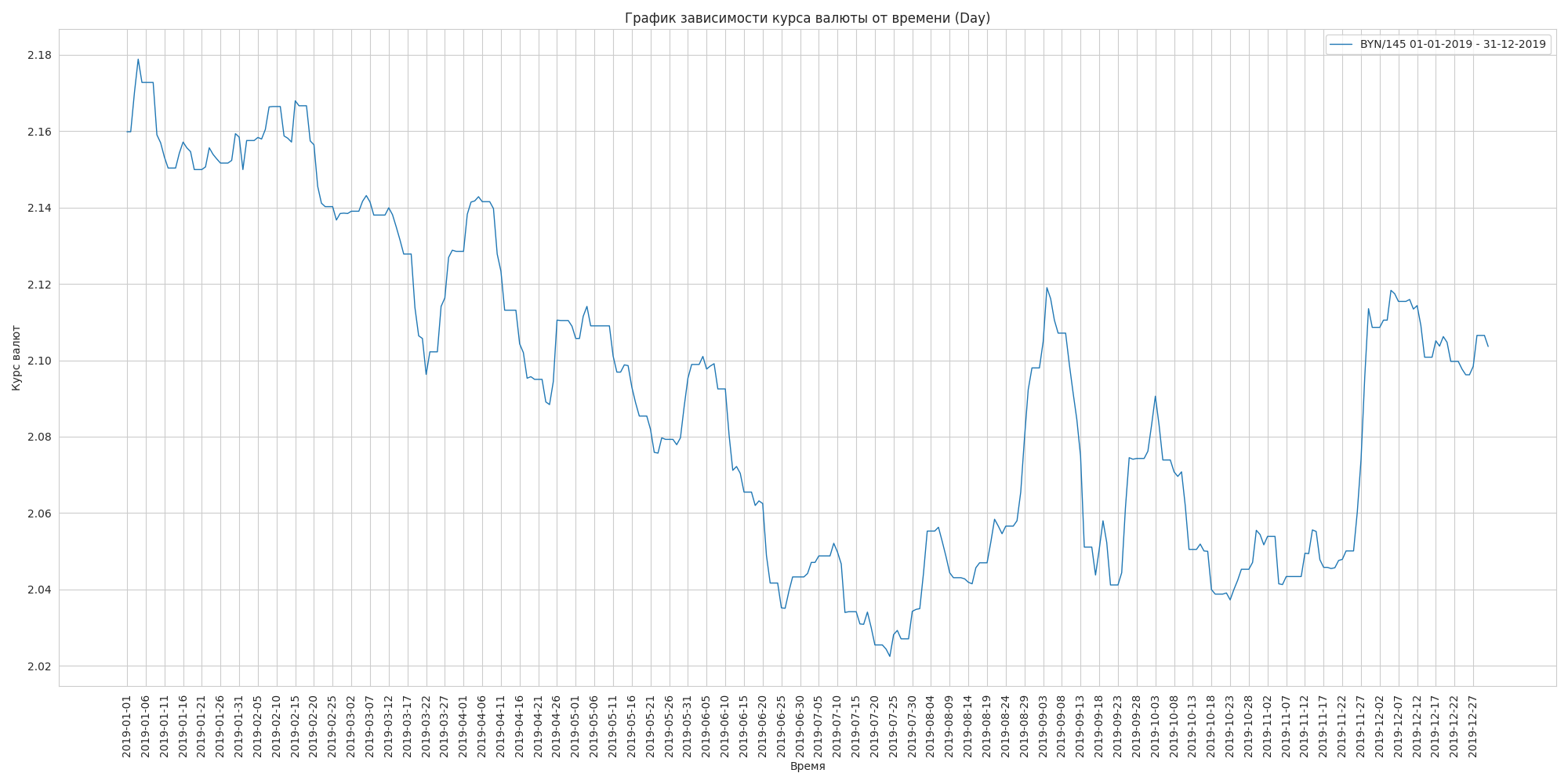


Рисунок 5 – График зависимости курса

валюты от времени за один день

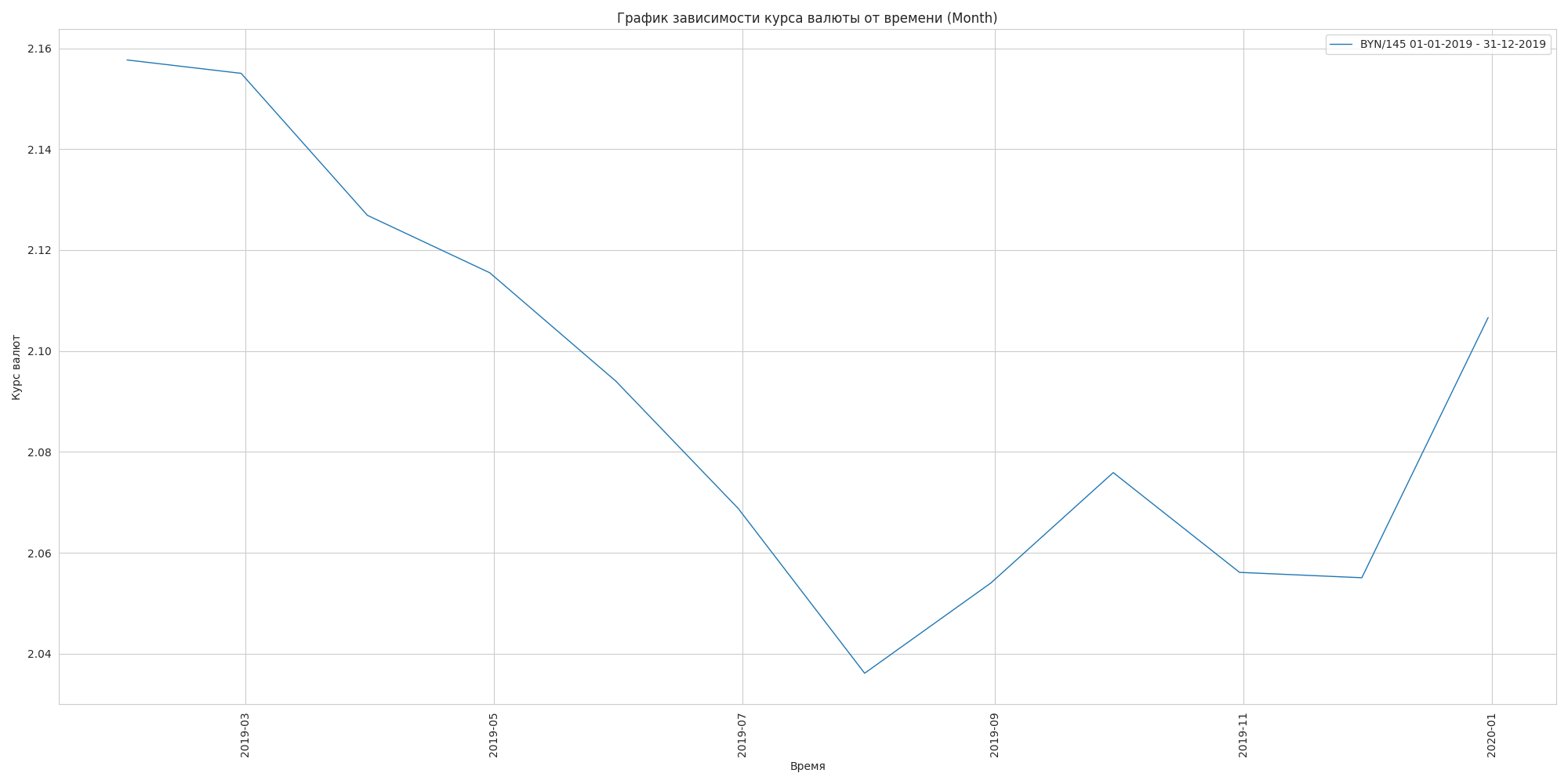


Рисунок 6 – График зависимости курса

валюты от времени помесячно

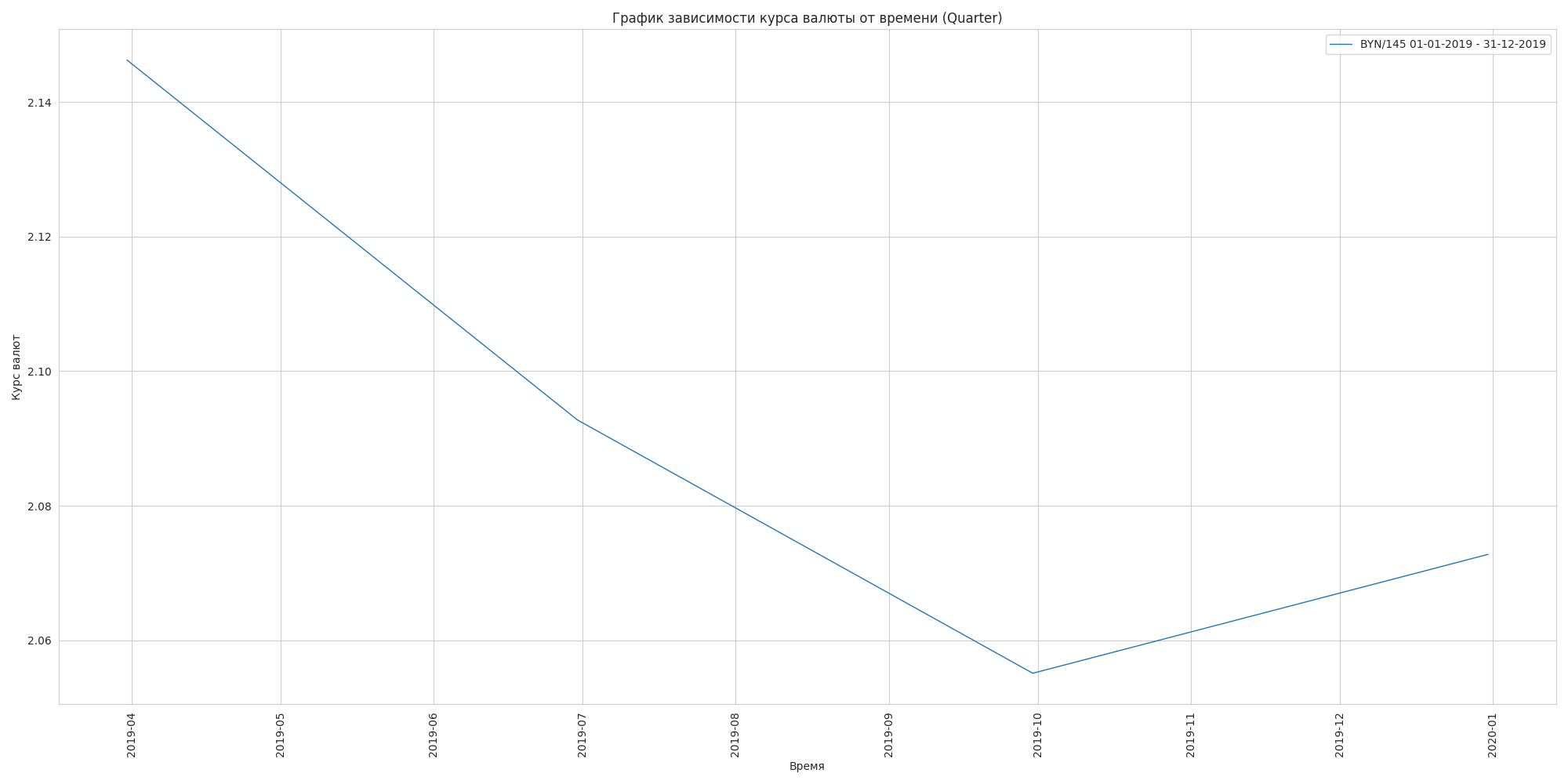


Рисунок 7 – График зависимости курса

валюты от времени поквартально

Декомпозируем временной ряд на составляющие: тренд, сезонность и остаток.

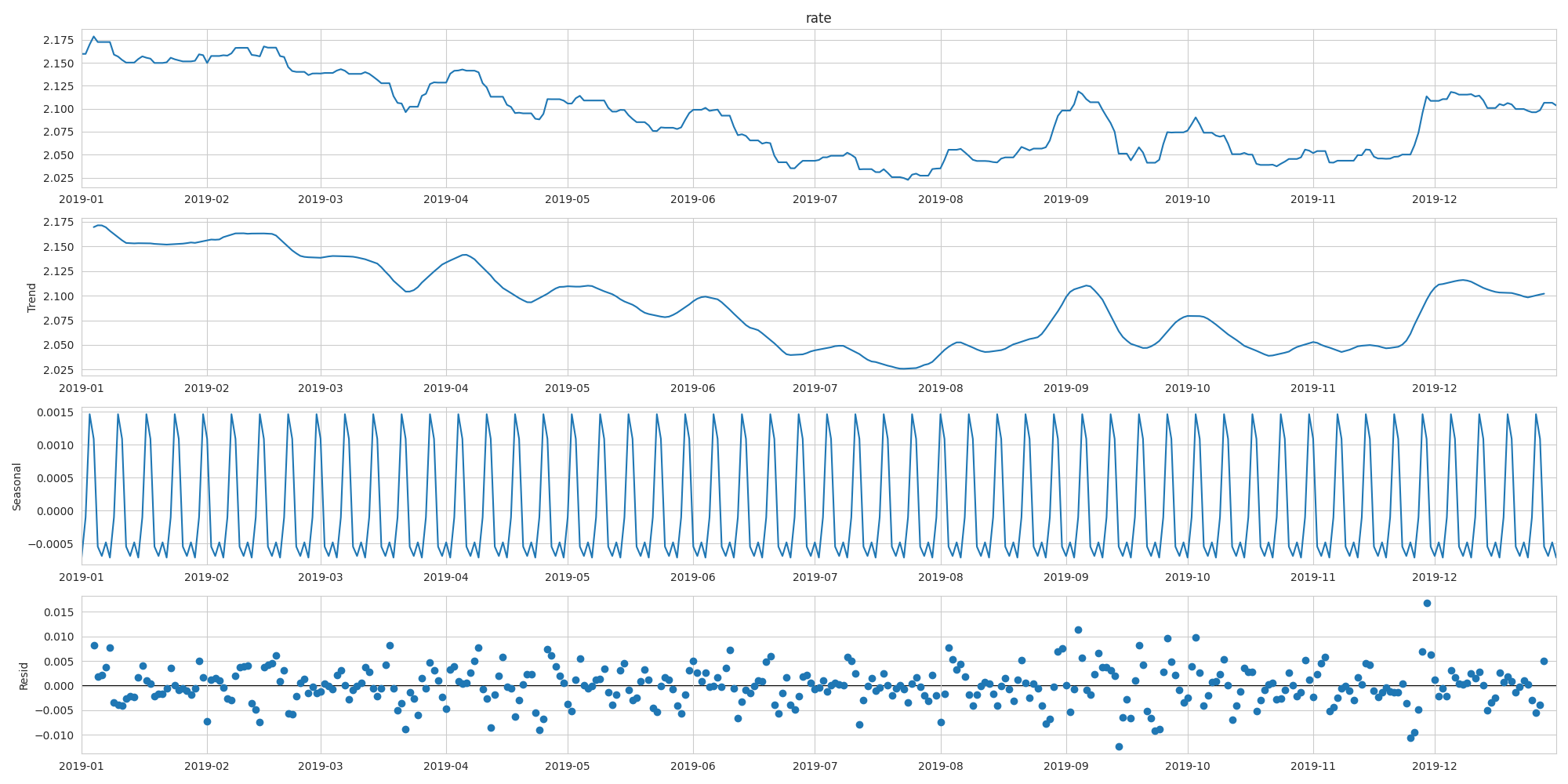


Рисунок 8 – Декомпозиция временного ряда

Найдем изменение курса валюты за предыдущий день.

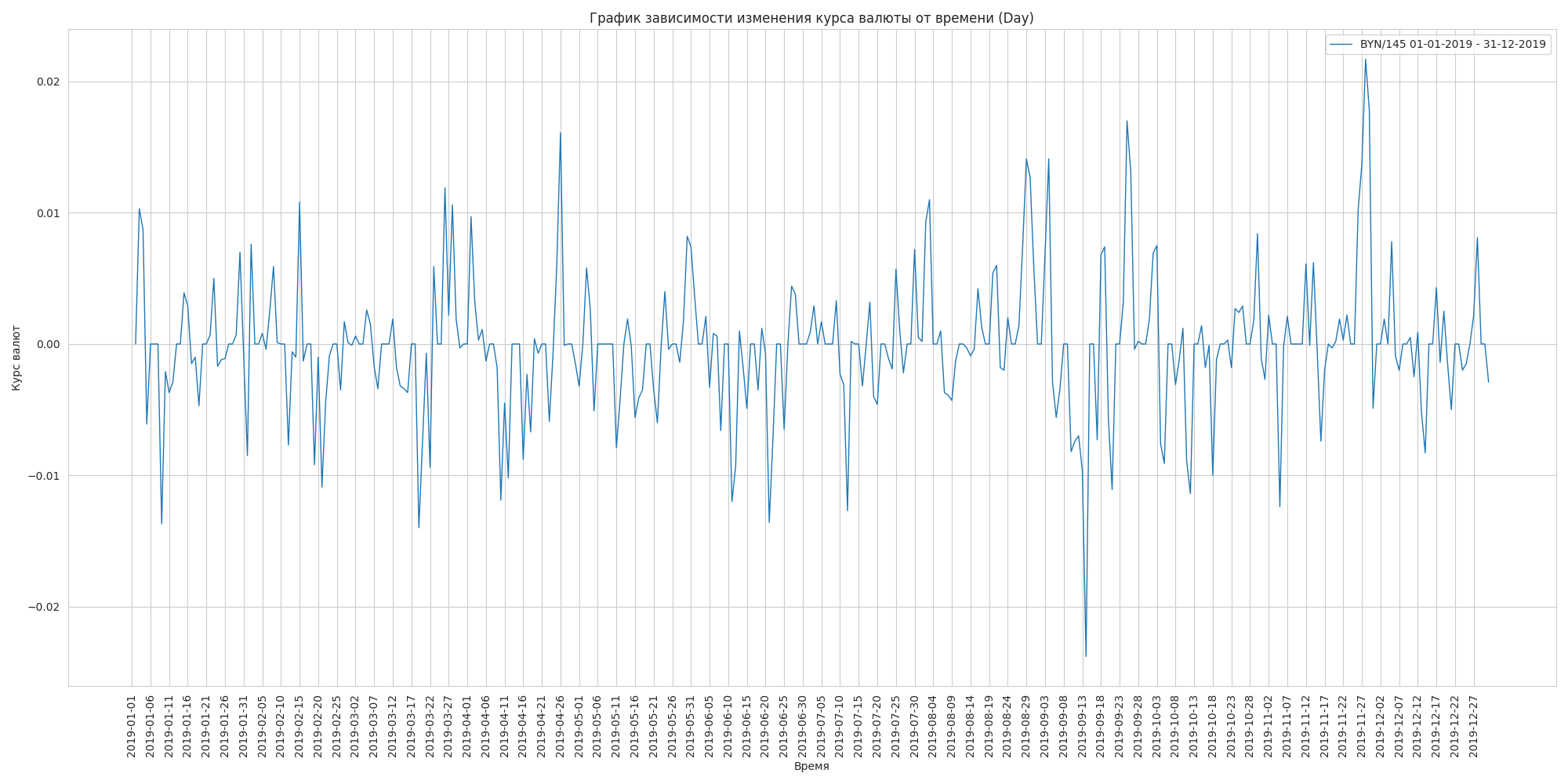


Рисунок 9 – Изменение курса валюты за предыдущий день

Найдем среднеквадратическое отклонение курса валюты за каждые 5 дней.

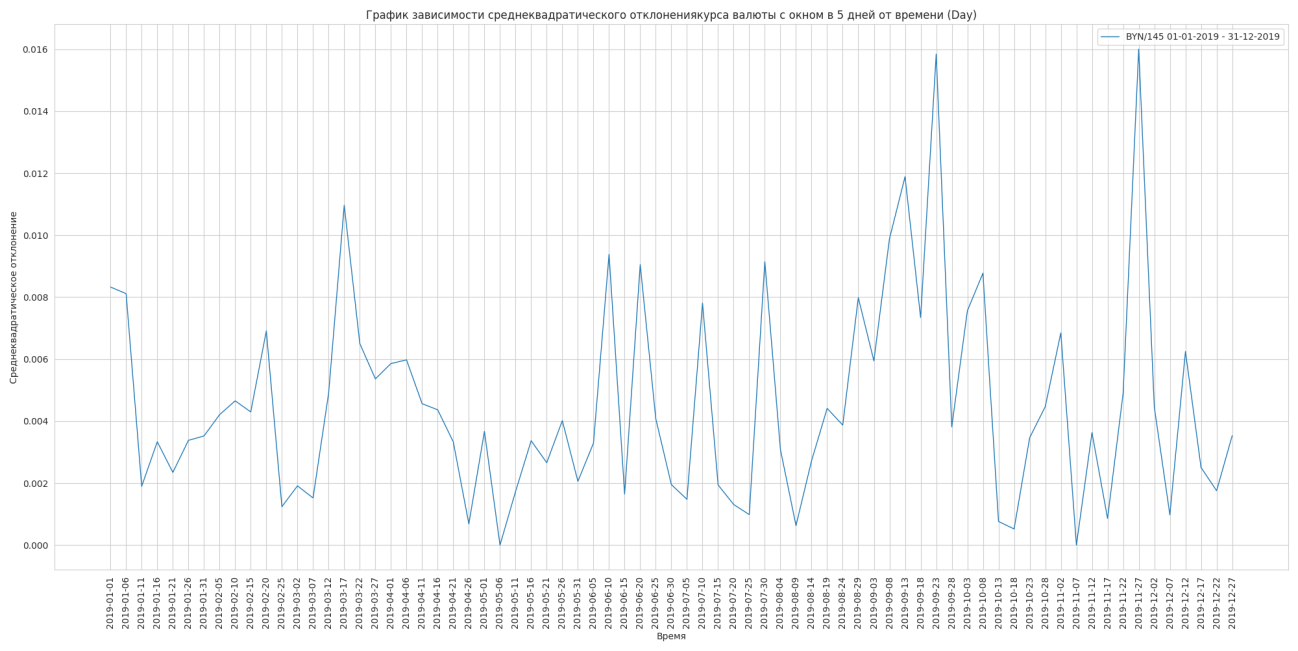


Рисунок 10 – среднеквадратическое отклонение курса

валюты за каждые 5 дней

По приведенным выше графикам видно, что в период с 9 по 28 сентября, а также с 22 по 28 ноября были наибольшие выбросы. По рисунку 9 определяем, что на первом интервале было наибольшие снижение курса, а на втором – наибольшие возрастание.

Построим гистограмму исходного временного ряда.

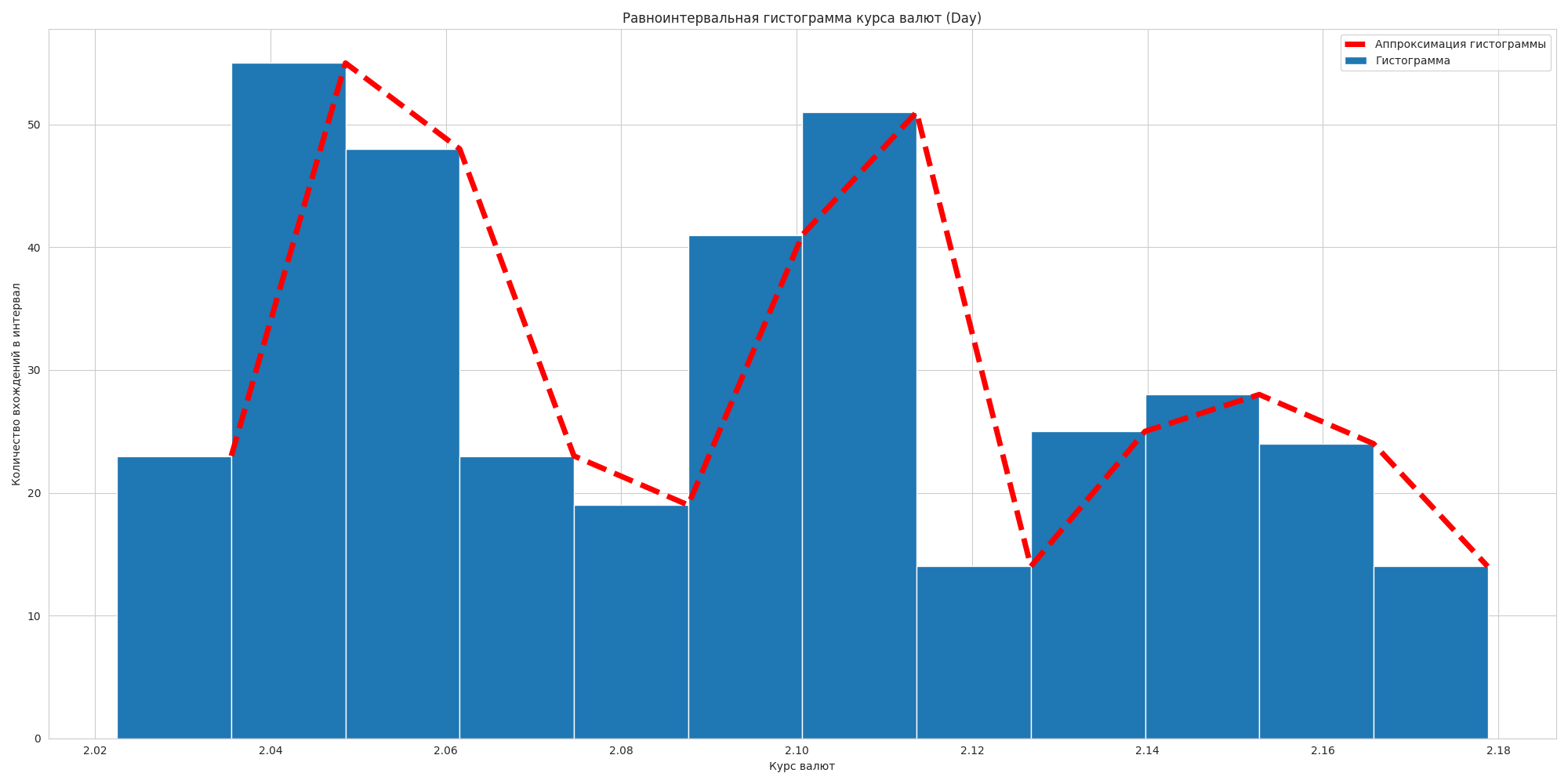


Рисунок 11 – Равноинтервальная гистограмма курса валюты

Из рисунка 11 видно, что существуют 3 более-менее различимые гаусианы, это значит, что на протяжении рассматриваемого периода это были наиболее характерные зоны курса валюты.

## Вывод

В ходе данной лабораторной работы была разработана программа для обработки данных курсов валют Национального банка Республики Беларусь. Для этого использовался язык программирования Python. Официальный курс брался с веб-ресурса <http://www.nbrb.by/>. Веб-ресурс предоставляет удобное API для получения данных в удобном формате, благодаря чему имеется возможность построить графики для проведения анализа.

Были проанализированы и построены графики по следующим пунктам:

* Курс белорусского рубля по отношению к иностранной валюте за год;
* Курс белорусского рубля по отношению к иностранной валюте за месяц;
* Курс белорусского рубля по отношению к иностранной валюте за заданный период;
* Использовалась следующая валюта: доллары, евро, российский рубль.