Дипломная работа по теме:

**Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly**.

Выполнил: Жуков Михаил Андреевич

Оглавление дипломной работы:

Введение. 3

Обоснование выбора темы: 4

Введение в библиотеки для визуализации на Python. 4

Библиотека Matplotlib

Библиотека Seaborn

Библиотека Plotly

Анализ и интерпретация результатов.

**Введение**

Сегодня самым ценным ресурсом в мире является уже не нефть, а данные. Визуализация становится все более важным инструментом для понимания миллиардов строк данных. Преобразуя данные в графическое представление, которое легко интерпретировать, визуализация данных помогает излагать историю данных, выделяя соответствующую информацию, закономерности и выбросы. Однако данные и графика должны работать вместе: это искусство объединения качественного анализа с отличным повествованием.

Визуализация данных — это важный инструмент для анализа и представления информации. Python предлагает множество библиотек для создания разнообразных графиков, диаграмм и других визуализаций.

Язык программирования Python часто используют аналитики данных. Для этого в нем существуют расширения — библиотеки, наборы готовых инструментов для более эффективной работы.

Визуализация данных, неотъемлемая часть науки о данных, представляет собой визуальное представление данных. Преобразовывая сложные наборы данных в более понятный графический формат, мы можем эффективно передавать идеи. Войдите в Python, мощный язык программирования, широко известный своим применением в науке о данных, благодаря своей простоте и широкому набору библиотек, предназначенных для обработки и визуализации данных.

**Важность визуализации данных**

Визуализация данных — это не модное дополнение, а необходимость. Это позволяет нам быстро интерпретировать данные и корректировать различные переменные, чтобы увидеть их влияние. Диаграммы и графики облегчают передачу данных, поскольку они могут эффективно рассказать историю. Python играет здесь решающую роль; его разнообразные библиотеки визуализации предоставляют инструменты для создания сложных и информативных изображений, помогая лучше анализировать данные и принимать решения.

**Обоснование выбора темы.**

В текущее время все больше и больше данных накапливает человечество, что порождает спрос на обработку, накопленной информации. Исходя из особенности человеческого восприятия и перефразируя поговорку: «Большое видится на расстоянии», другими словами большие объемы данных человек воспринимает лучше в графическом виде, давайте рассмотрим, как это реализуется на Python.

И для начала приведем наиболее популярные инструменты для этого. Так как данный вопрос очень интересен было принято решение выбрать данную тему диплома «**Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly».**

**Введение в библиотеки для визуализации на Python.**

Python **библиотеки для визуализации данных: Matplotlib, Seaborn и Plotly**.

Python обладает большим числом библиотек для визуализации данных, предлагая инструменты для любых нужд. Эти библиотеки похожи на разные кисти, каждая со своим уникальным стилем и силой, готовые воплотить ваши данные в жизнь. Здесь мы обсудим три популярных:

1. **Matplotlib**. Matplotlib часто считается основой визуализации данных в Python. Это универсальная библиотека. Несмотря на то, что это одна из старейших библиотек Python для этой цели, ее сила заключается в ее гибкости. Это позволяет вам полностью контролировать внешний вид ваших графиков, от простых линейных графиков до сложных трехмерных графиков. Matplotlib предлагает широкий спектр параметров настройки, что дает вам возможность создавать практически любую визуализацию.
2. **Seaborn**: если Matplotlib — это огромный, красивый, но несколько пугающий лес, то Seaborn — это дружелюбный проводник, который проведет вас через него. Seaborn построен на основе Matplotlib, предлагая более удобный интерфейс и эстетическое улучшение. Он специализируется на статистической визуализации, что позволяет легко создавать привлекательные и информативные статистические графики. От простых точечных диаграмм до сложных тепловых карт Seaborn предлагает упрощенный подход к сложной визуализации.
3. **Plotly**. В эпоху интерактивности Plotly сияет. Он позволяет создавать динамические и интерактивные визуальные эффекты, которые можно встраивать в веб-приложения или в отдельный HTML-код. Эта интерактивность предлагает новый уровень взаимодействия, позволяя пользователям масштабировать, панорамировать, наводить курсор и нажимать на графики для получения дополнительной информации. Plotly позволяет не только увидеть, но и испытать ваши данные.

Дополнительно для упрощения работы с большими объемами данных применяют библиотеку Pandas, перед тем как отправить эти данные на визуализацию, формируя понятные и информативные графики и диаграммы. Pandas — популярная библиотека в Python для работы с данными. Её активно используют аналитики данных и ученые.

Библиотека была создана в 2008 году компанией AQR Capital, а в 2009 году она стала проектом с открытым исходным кодом с поддержкой большого комьюнити, что и обеспечило ее популярность и дальнейшее развитие.

Данную библиотеку используют в таких областях как:

**Аналитика данных: продуктовая, маркетинговая и другая.** Работа с любыми данными требует анализа и подготовки: необходимо удалить или заполнить пропуски, отфильтровать, отсортировать или каким-то образом изменить данные. Pandas в Python позволяет быстро выполнить все эти действия, а в большинстве случаев ещё и автоматизировать их.

**Data science и работа с большими данными.** Pandas помогает подготовить и провести первичный анализ данных, чтобы потом использовать их в машинном или глубоком обучении.

**Статистика.** Библиотека поддерживает основные статистические методы, которые необходимы для работы с данными. Например, расчёт средних значений, их распределение по квантилям и другие.

И получившиеся результаты в дальнейшем проще визуализировать.

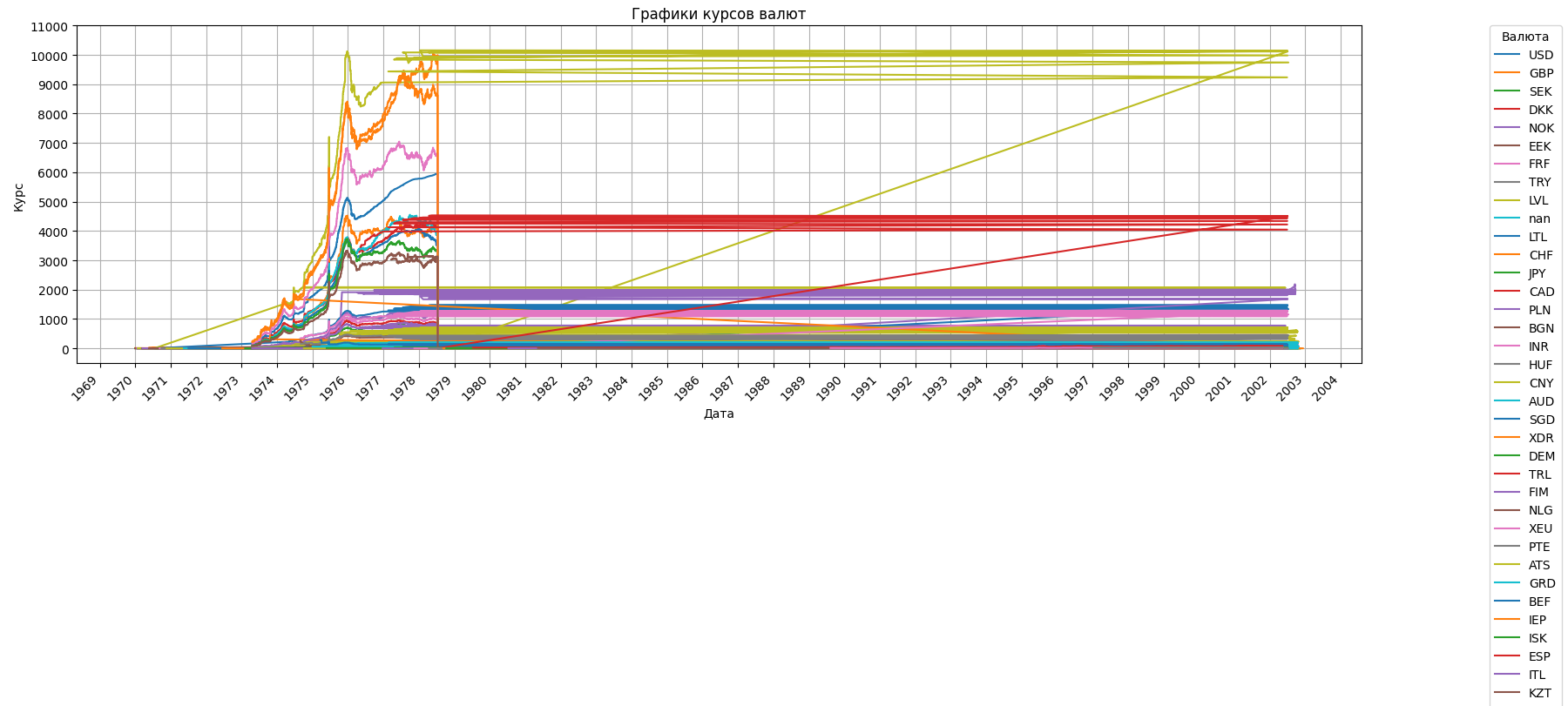
Определение цели и задач исследования

Возможно вы часто в последнее время слышите слова: котировки, биржа, фьючерсы и т. п., но если присмотреться, то с точки зрения программиста это будут наборы переменных числа, и строки. Причем эти данные могут как изменяться в реальном времени так и накапливаться где либо для дальнейшего анализа. Беря для примера график курса RUB-EUR мы получаем в случае упрощения набор на единичный график где у вас две координаты стоимость и дата, и он достаточно прост. Но что будет если взать более 30 пар валют и период времени не один два месяца, где количество данных будет небольшим. А период в несколько десятилетий, и при этом у нас возможны в данном наборе вариант, когда данные не полные на каком-то периоде, при недостатке данных и их переизбытке как поведут себя библиотеки для визуализации.

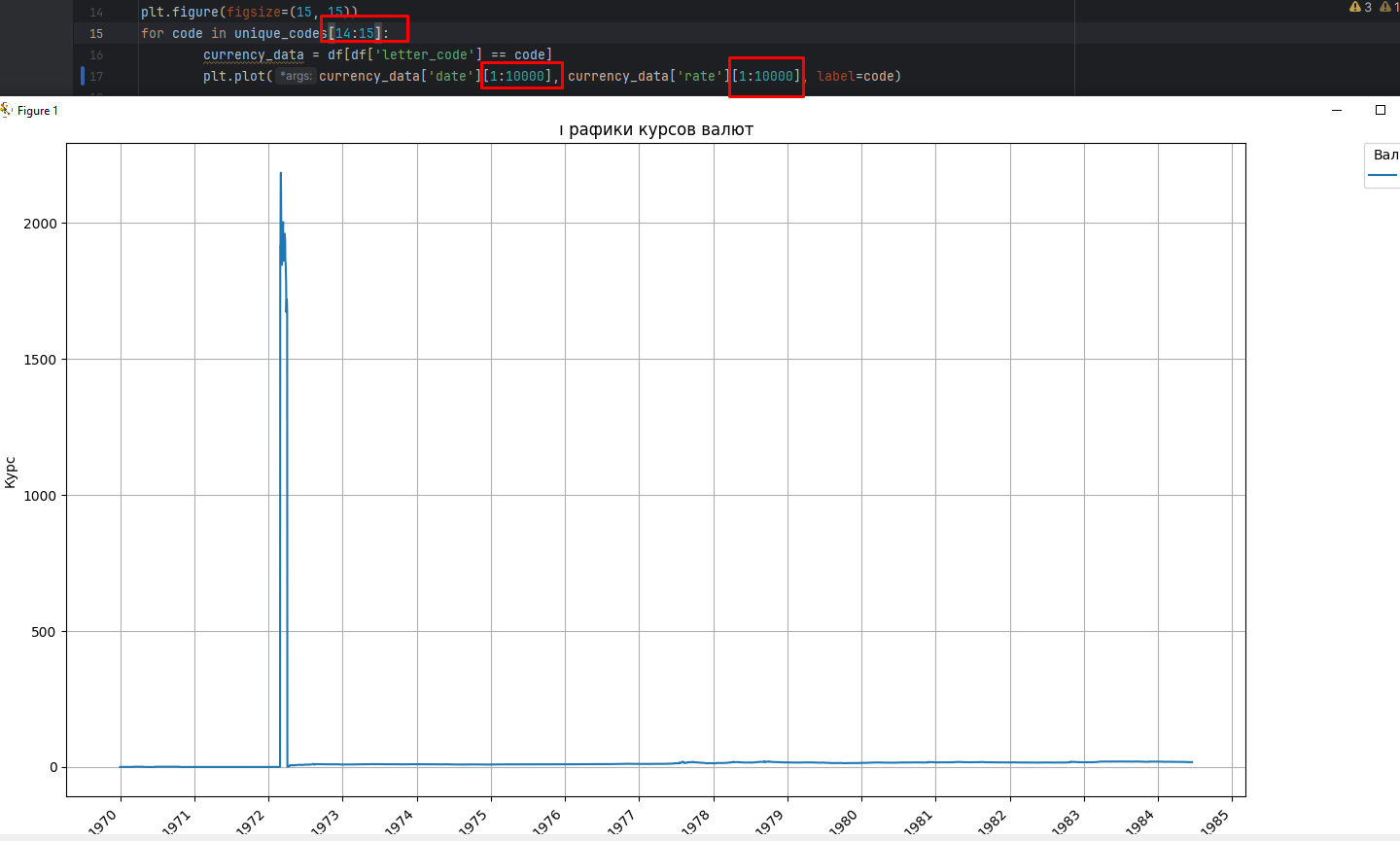
Возьмем наборы данных и проведём для него набор визуализаций с использованием разных библиотек Matplotlib, Seaborn и Plotly, чтобы сравнить их функциональность и удобство использования. Возьмем набор из 320 тысяч строк и более 30 валют их стоимость относительно рубля.

Библиотека Matplotlib

Первое, что замечаешь при работе с таким объемом данных это требуемое время для обработки всех строк и при использовании данной библиотеки оно ощутимо, а второе - возникающие артефакты, связанные с неполнотой данных по отдельным валютам.

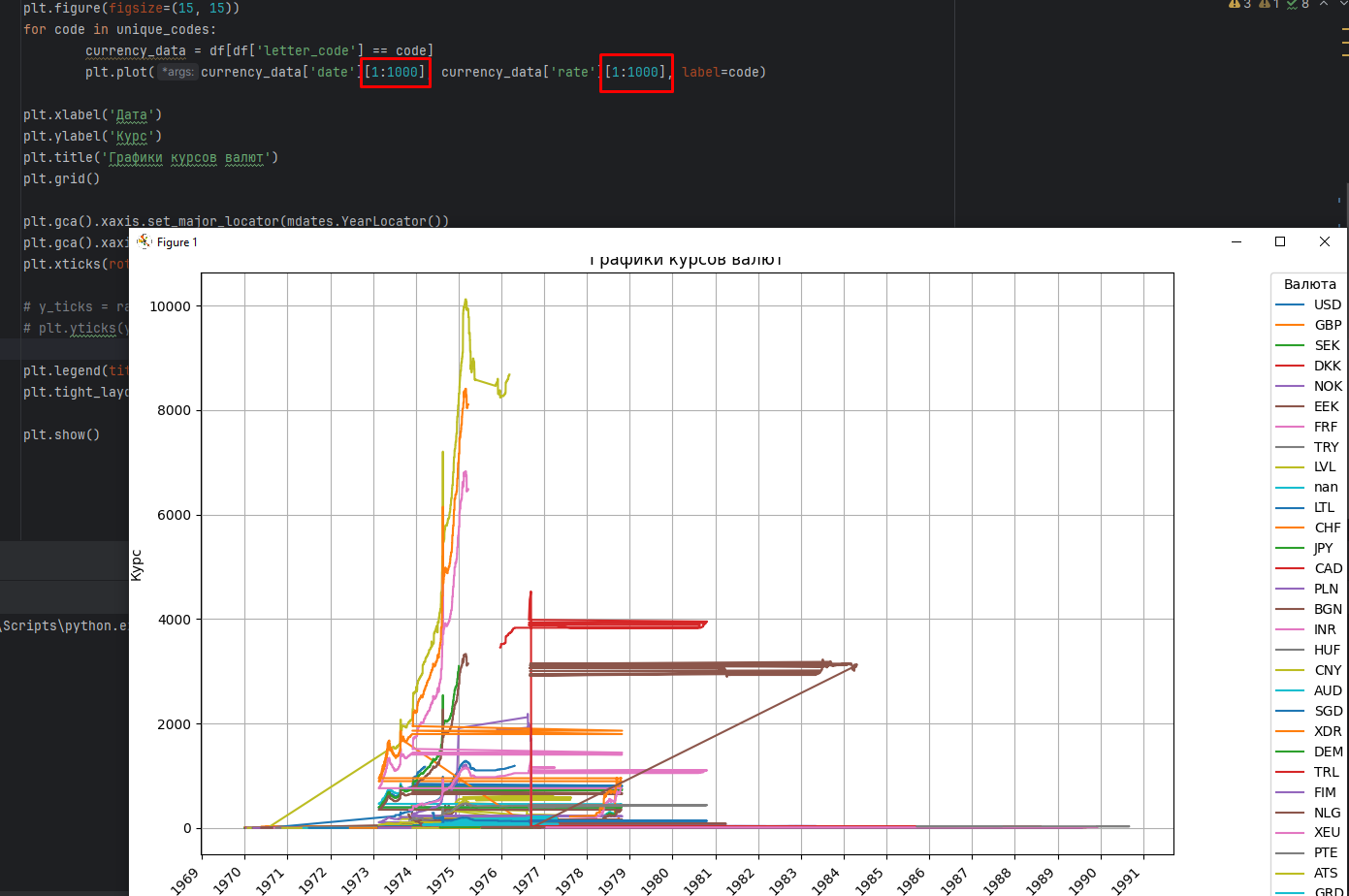


Далее при попытке понять с чем связаны артефакты проводил эксперименты по изменению количества отображаемых данных. Например единичный график.

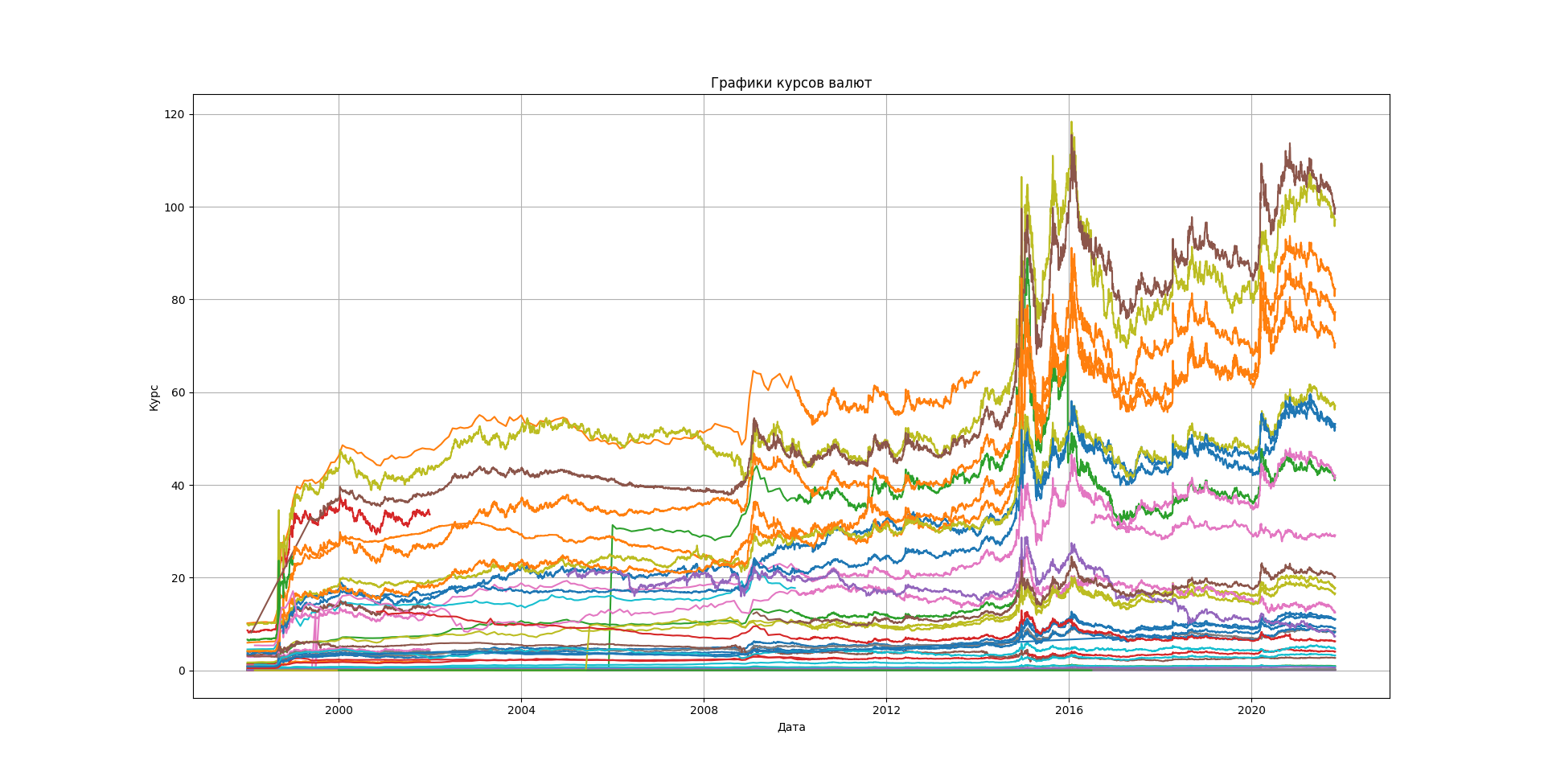


Как видим отрисовывается без проблем что говорит, о том, что артефакты возникают не по причине отдельных данных

Вернув количество графиков, но оставив ограничение по времени отображения получили:



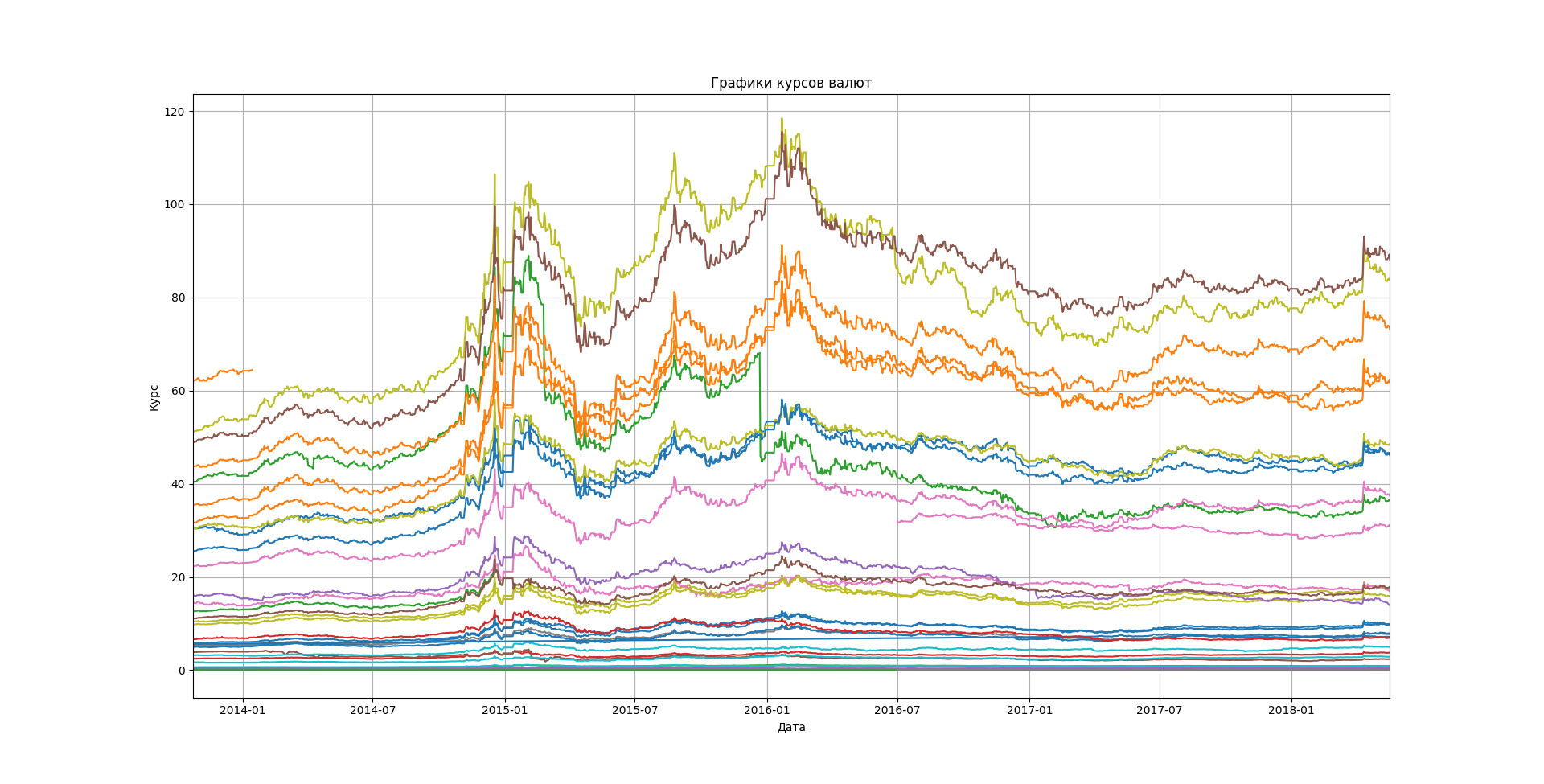
Проведя анализ исходных данных получил результат: что качество исходных данных также требуется корректировать, введя ограничение по периоду когда были данные по всем валютам



Так же для ускорения работы с построением такого объема данных в виде графика в программу импортируем не всю библиотеку а только часть ее

import matplotlib.pyplot as plt

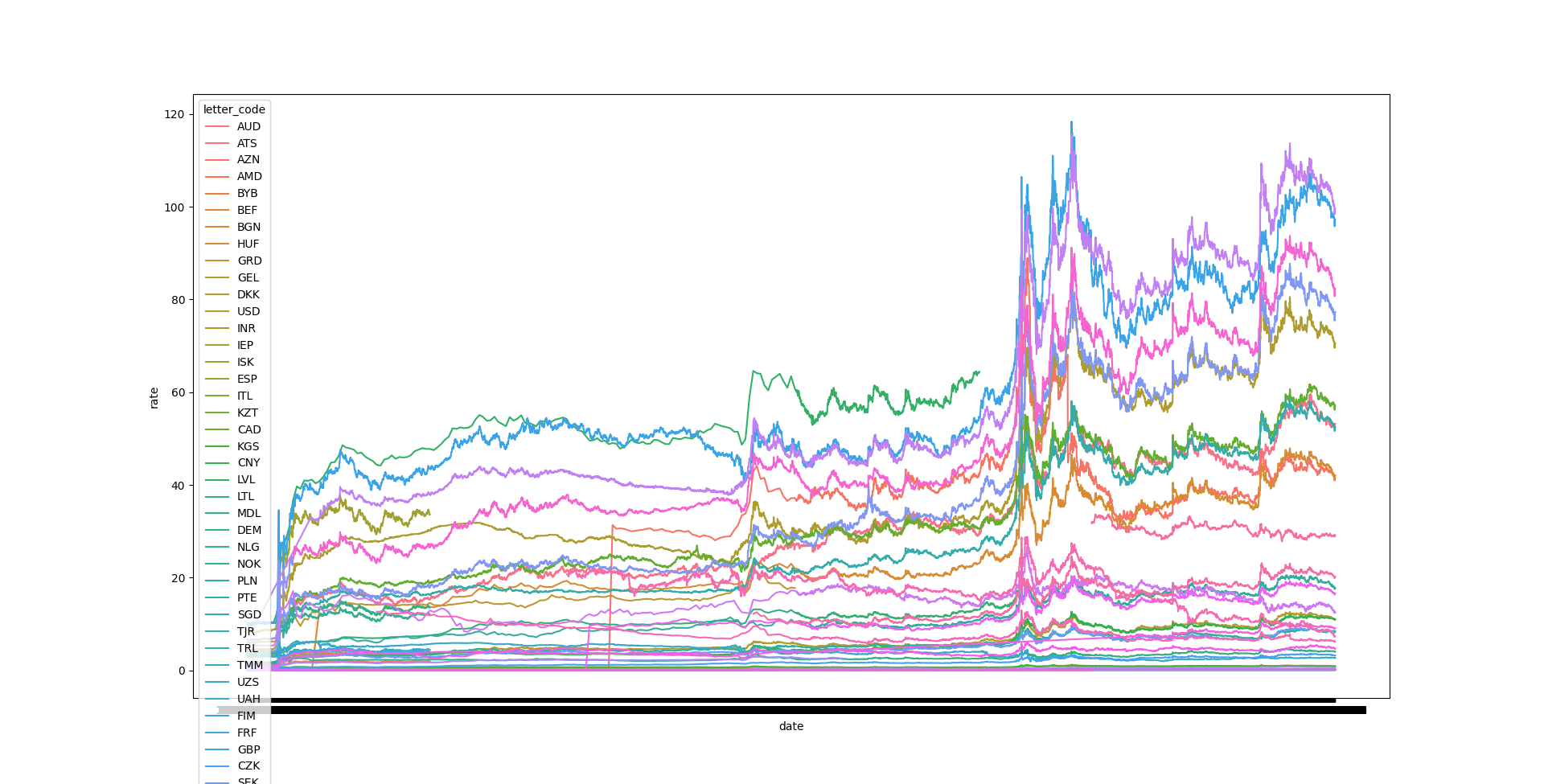
Так же при просмотре построенных данных мы моем получить более детальную информацию просто увеличив график не закрываю программу



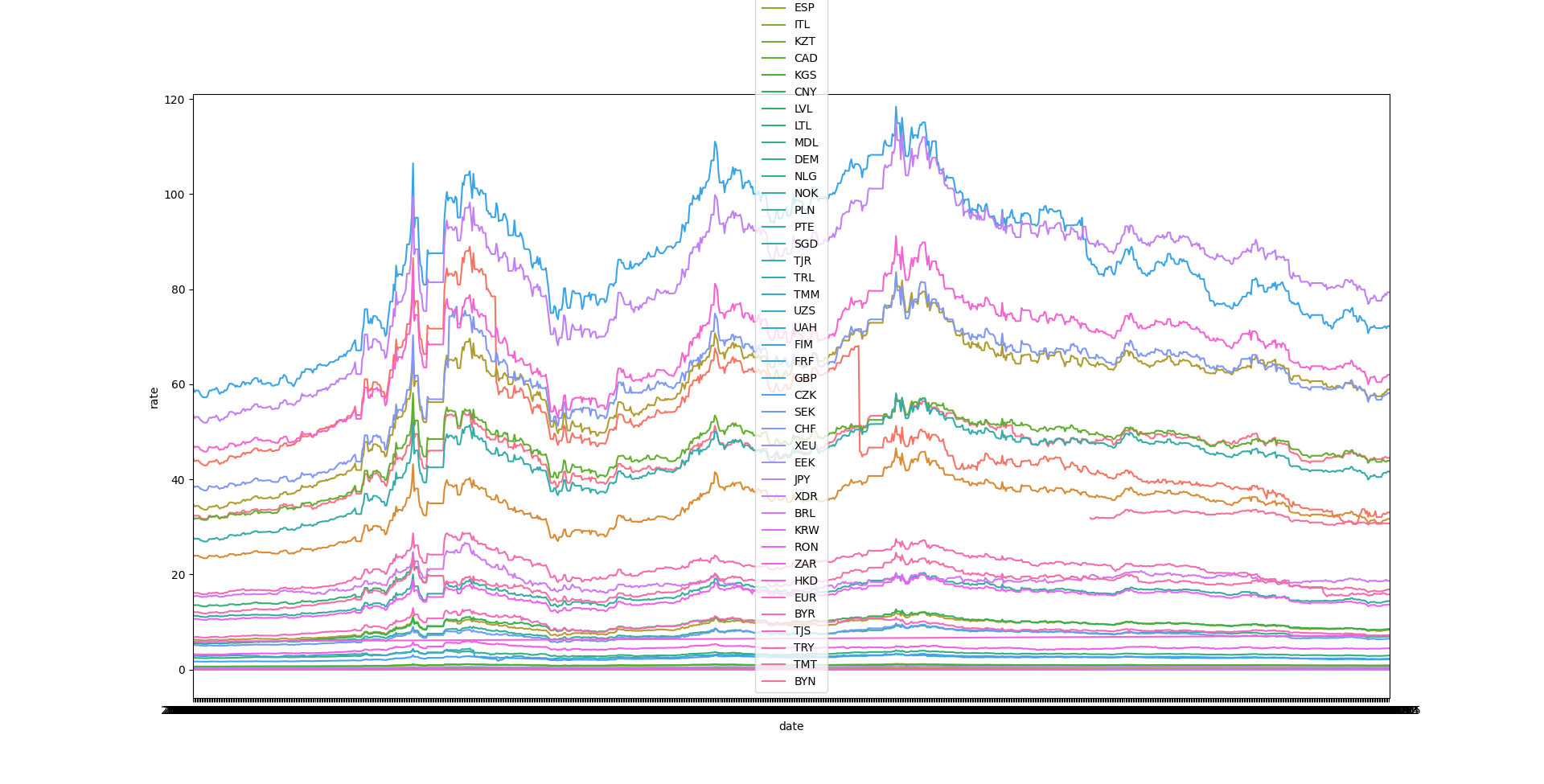
Как видим мы наблюдаем по всем валютам рост стоимости валют в период с 2014-09 месяца по примерно 2016-03 месяц. Скорее всего в данный период возникли мировые события повлиявшие на стоимость валют, но скорее всего даже ранее, так как рынок имеет инерцию реагирования.

Библиотека Seaborn

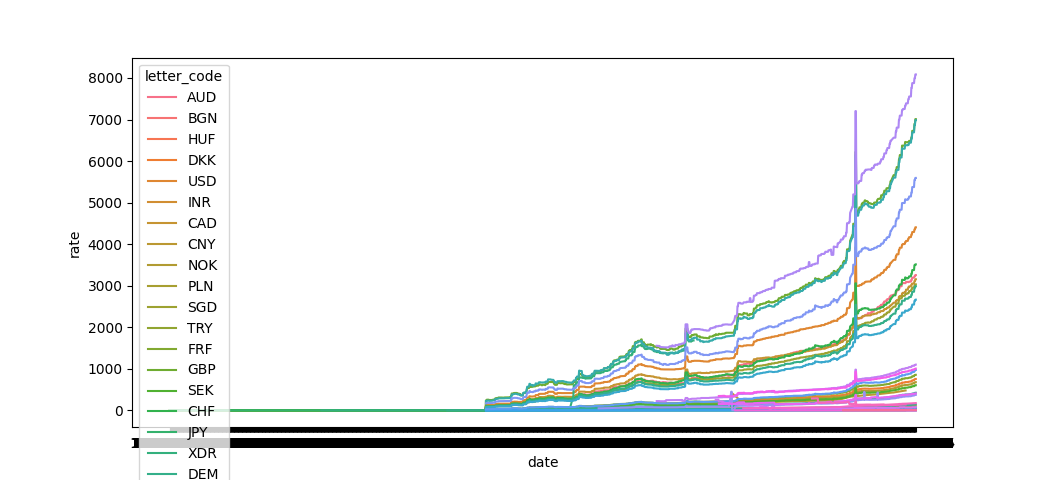
Посмотрим как измениться отображение данных на графике с использованием библиотеки более высокого уровня но основанная на matplotlib с учетом ограничения по датам



Как видим достаточно похоже выглядят графики в том числе и в увеличении



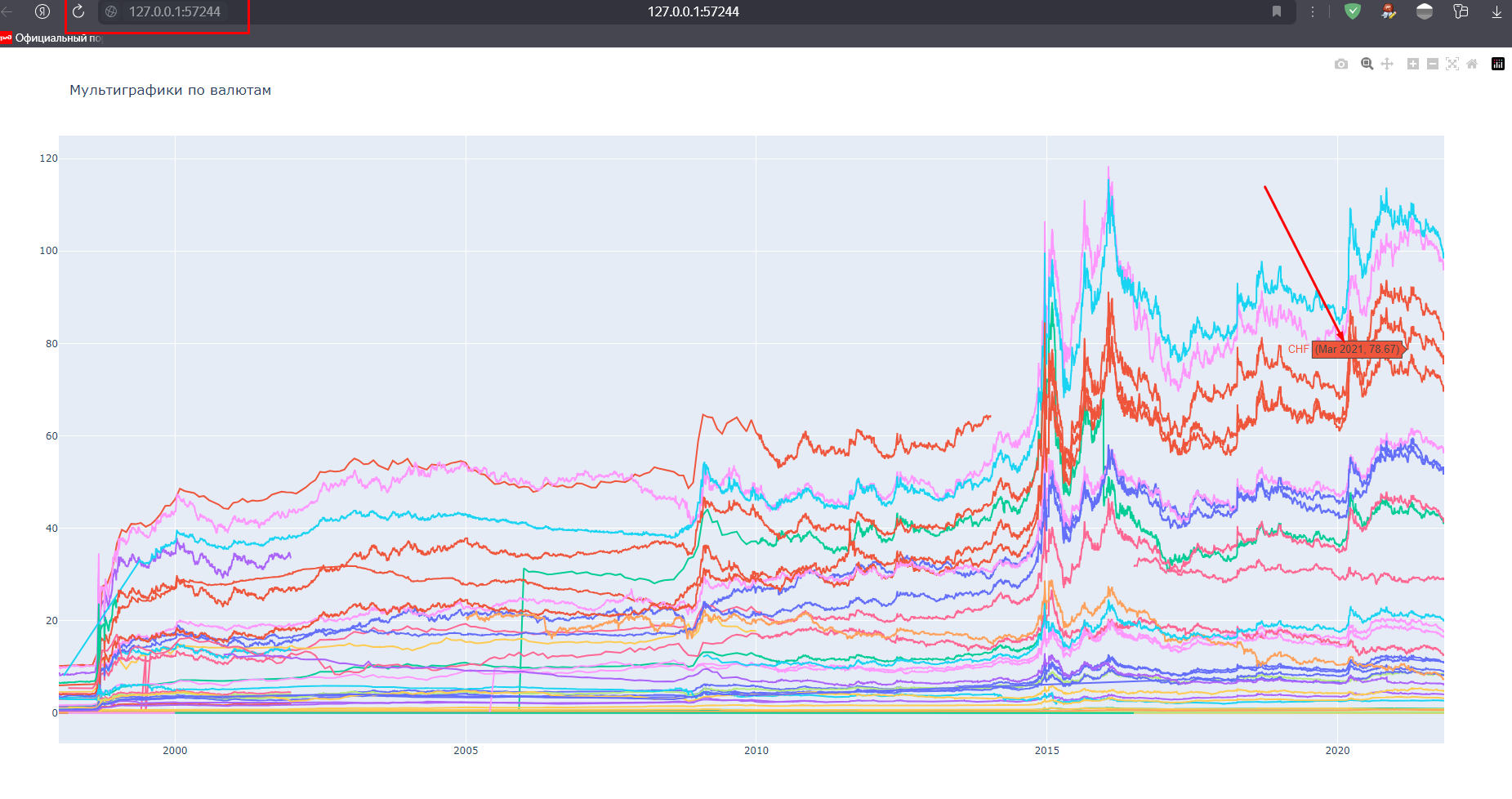
Но как можно видеть при подкладывании полного объема данных с пробелами в данных артефактов стало меньше



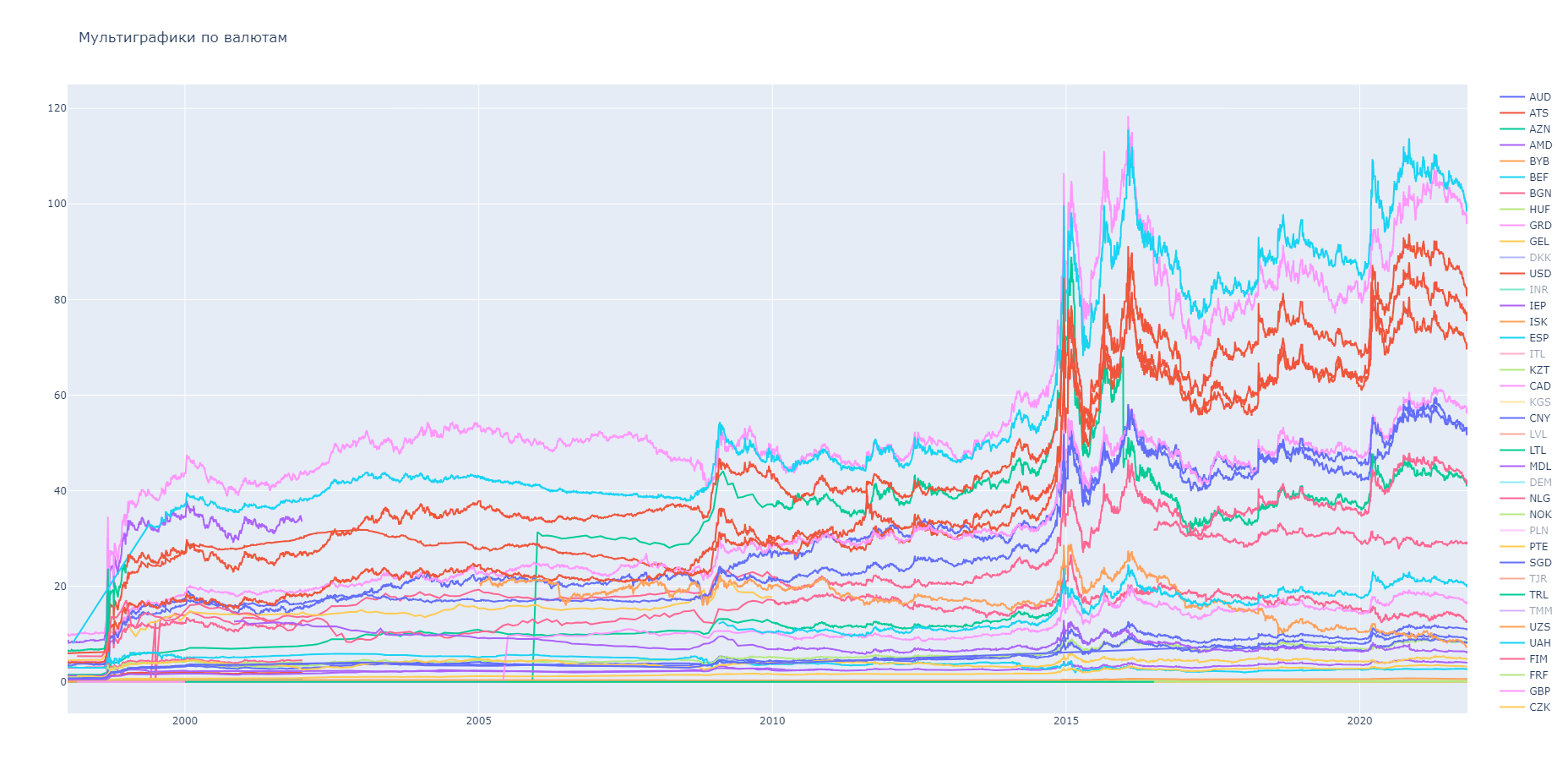
Так же заметно что график отрисовывается быстрее, кода в программе меньше 16 строк против 23 строк с базовой библиотекой и он более читабельный.

Библиотека Plotly

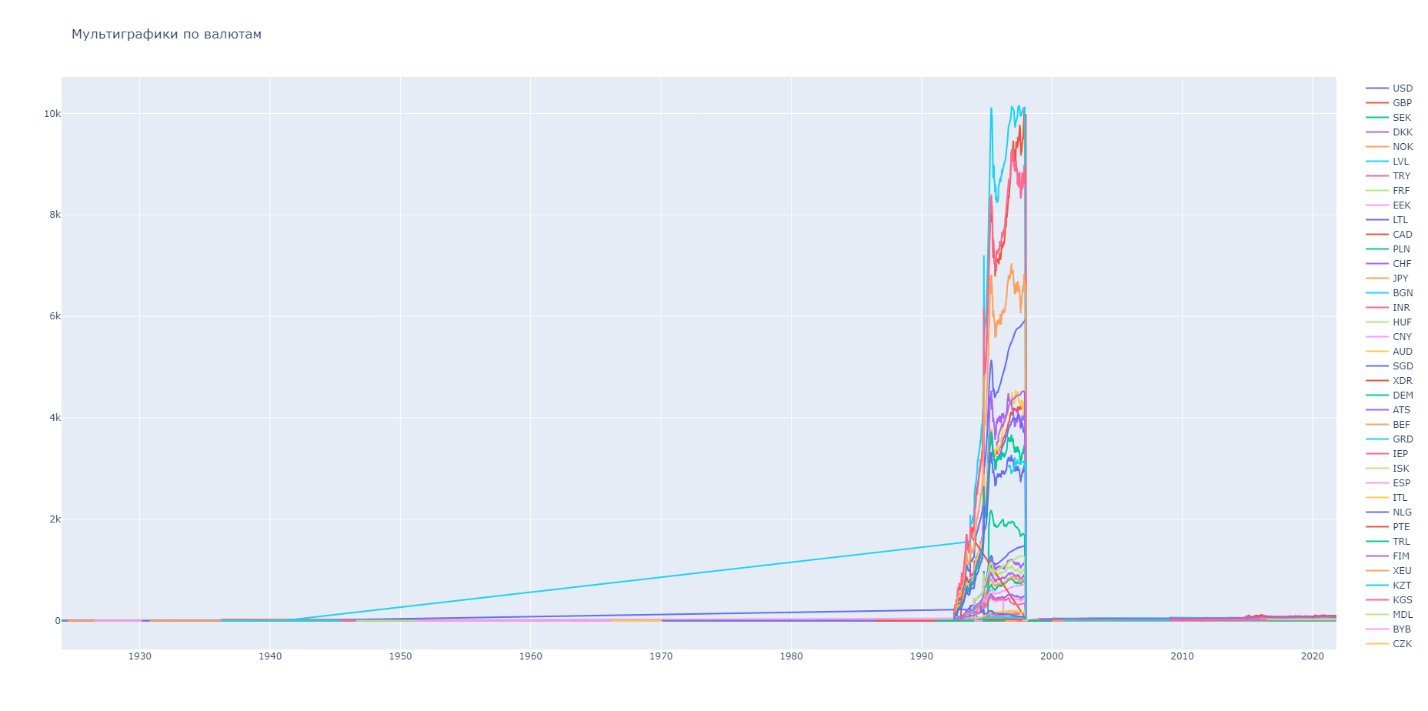
Работа с данной библиотекой сразу отличается от предыдущих библиотек, начиная, с того что формирует сразу из «коробки» график, не как окно, а как html документ, открывая в браузере как новую страницу с локальным адресом. А так же, всплывающими аннотациями при наведении курсора на точку графика.



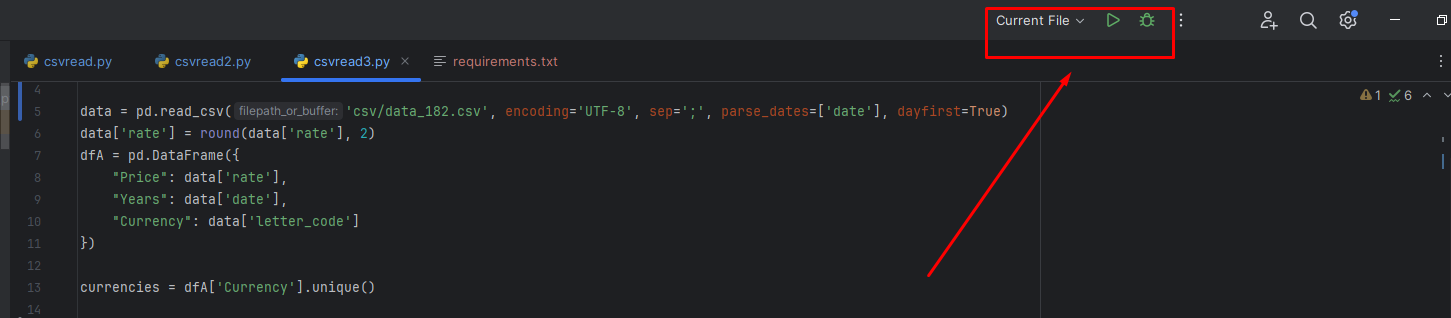
Так же при включении “легенды” можно управлять отображаем ли какой то из графиков нажатием на нужную легенду.



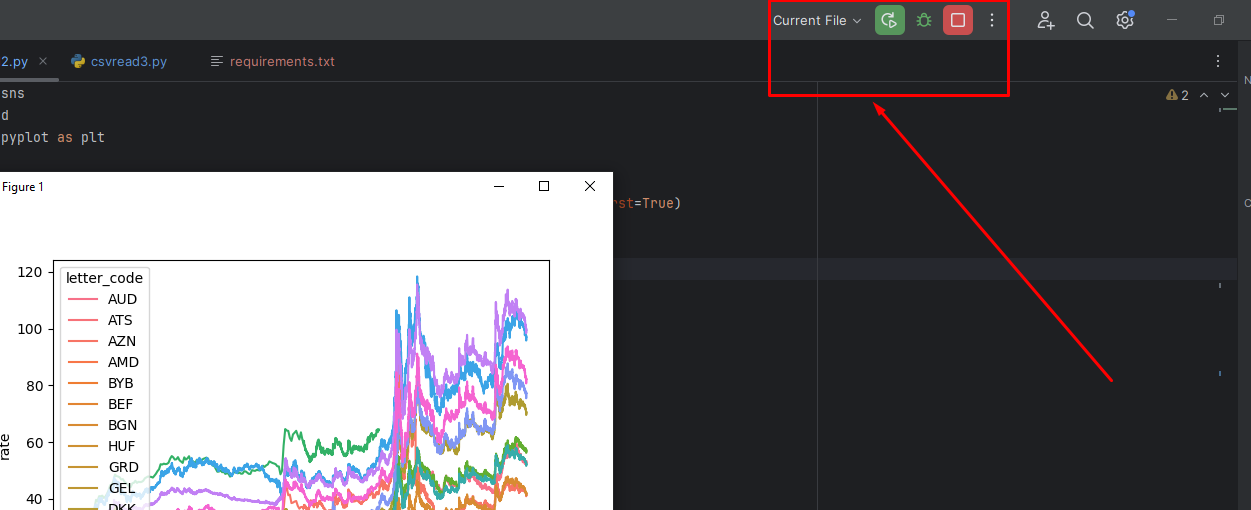
Так же при просмотре сырых данных видим, что данная библиотека так же требует контроля изначальных данных если данных на каком-то участке не хватает или есть где-то неупорядоченность данных.



Но что достаточно удобно в отличии от двух предыдущих библиотек мы наблюдаем результат выполнения и после закрытия программы



В отличии от двух предыдущих библиотек, результат выполнения, с помощью которых, возможен, лишь при запущенной программе.



Анализ и интерпретация результатов.

Не смотря на то что для получения аналогичного графика для экспериментального сравнения удобства библиотек потребовалось 27 строк, время обработки данных близко к библиотеке Seaborn, и также быстрее обработала массив данных, чем matplotlib.

Согласно полученным результатам нельзя сказать? что каждый из представленных библиотек имеет выраженные недостатки. Скорее можно сказать что каждая из представленных библиотек имеет свою область применения.