

Департамент образования и науки города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
Институт цифрового образования
Департамент информатики управления и технологий

Мошенина Елена Дмитриевна БД-241м

Программные средства сбора, консолидации и аналитики данных

Вариант 14

**Лабораторная работа №1-2. Современный парсинг динамических веб-сайтов:
Playwright, XPath и бизнес-аналитика**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Руководитель дисциплины:

Босенко Т.М., доцент департамента

информатики, управления и технологий,

доктор экономических наук

Москва

2025

Содержание

ВведениеОшибка! Закладка не определена.

Основная часть.....Ошибка! Закладка не определена.

ЗаключениеОшибка! Закладка не определена.

Введение.

Цель работы: освоить современный стек технологий для сбора данных с динамических веб-сайтов (Playwright + XPath). Научиться решать комплексные аналитические задачи, требующие сбора, очистки, сохранения в реляционную базу данных (SQLite) и анализа данных для принятия бизнес-решений.

Оборудование и ПО:

- **Компьютер с доступом в интернет.**
- **Окружение Python 3.8+:**
 - **Локально:** рекомендуется использовать виртуальное окружение (venv или conda).
 - **Облачные сервисы:** Google Colab, Jupyter Notebook.
- **Инструменты:** IDE (VS Code, PyCharm) или Jupyter Notebook, Git.
- **Рекомендуемый образ для воспроизводимости (опционально):**
<https://disk.yandex.ru/d/vIf6mYSu6aZuxQ>

Библиотеки: playwright, pandas, matplotlib, seaborn.

Вариант 14

14	Исследование рынка поддержанных авто: анализ объявлений.	Auto.ru. Применить фильтры: марка "LADA (BA3)", модель "Vesta", год от 2020.	Собрать цену, пробег, год выпуска. Проанализировать, как цена зависит от пробега для машин одного года.
----	--	---	---

Основная часть

Порядок выполнения работы

1. Подготовка окружения:

- Убедитесь, что у вас установлены все необходимые библиотеки

pip install playwright pandas matplotlib seaborn jupyterlab

- После установки библиотеки Playwright необходимо скачать браузеры, которыми она будет управлять:

playwright install

2. Анализ бизнес-кейса и веб-источника:

- Выберите вариант задания из таблицы ниже. Каждое задание представляет собой бизнес-кейс, требующий сбора данных с динамического сайта.

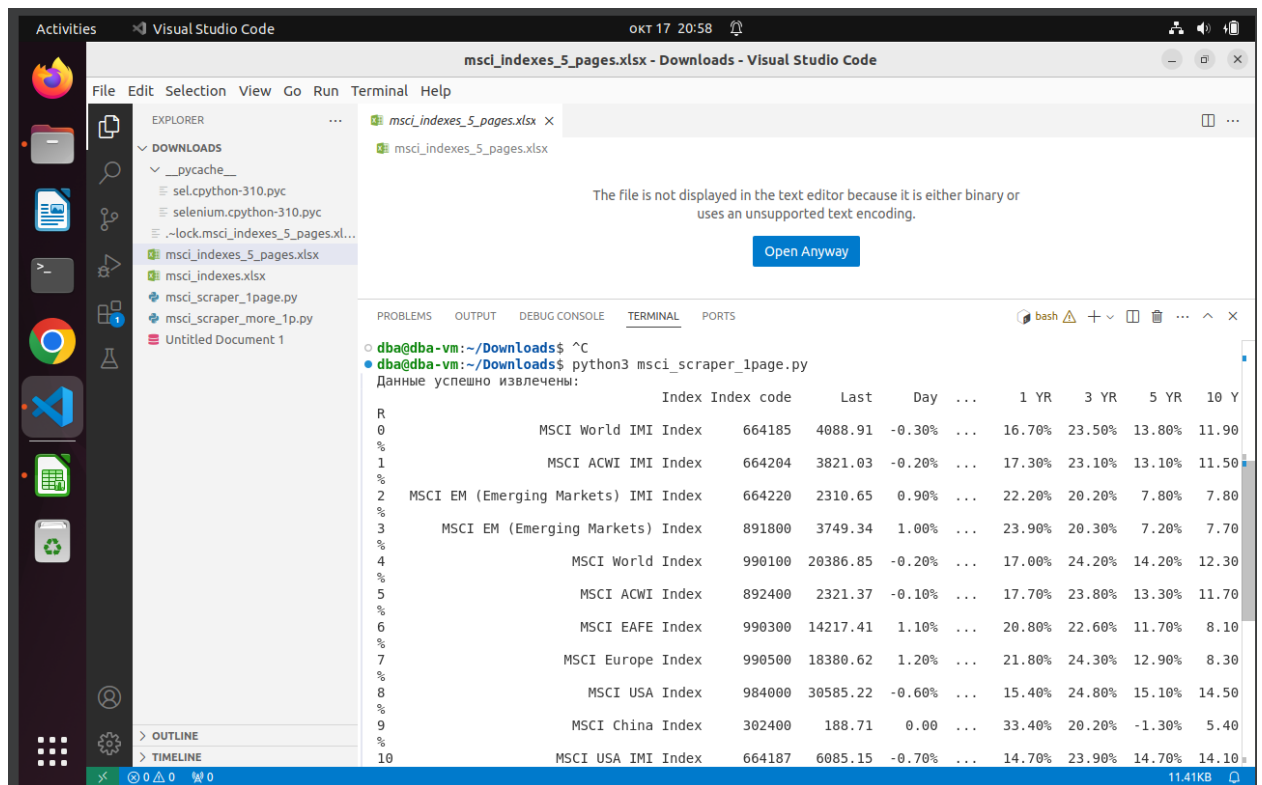
- Изучите целевую веб-страницу. Откройте инструменты разработчика (F12) и определите, как данные подгружаются на страницу (например, при скроллинге, нажатии на кнопку "Показать еще", пагинации).
- С помощью инструментов разработчика найдите ключевые HTML-элементы, содержащие нужную информацию, и составьте для них надежные XPath-селекторы.

3. Разработка парсера на Playwright:

- Напишите асинхронный Python-скрипт (async/await), который:
 - Инициализирует Playwright и запускает браузер.
 - Открывает целевой URL (page.goto()).

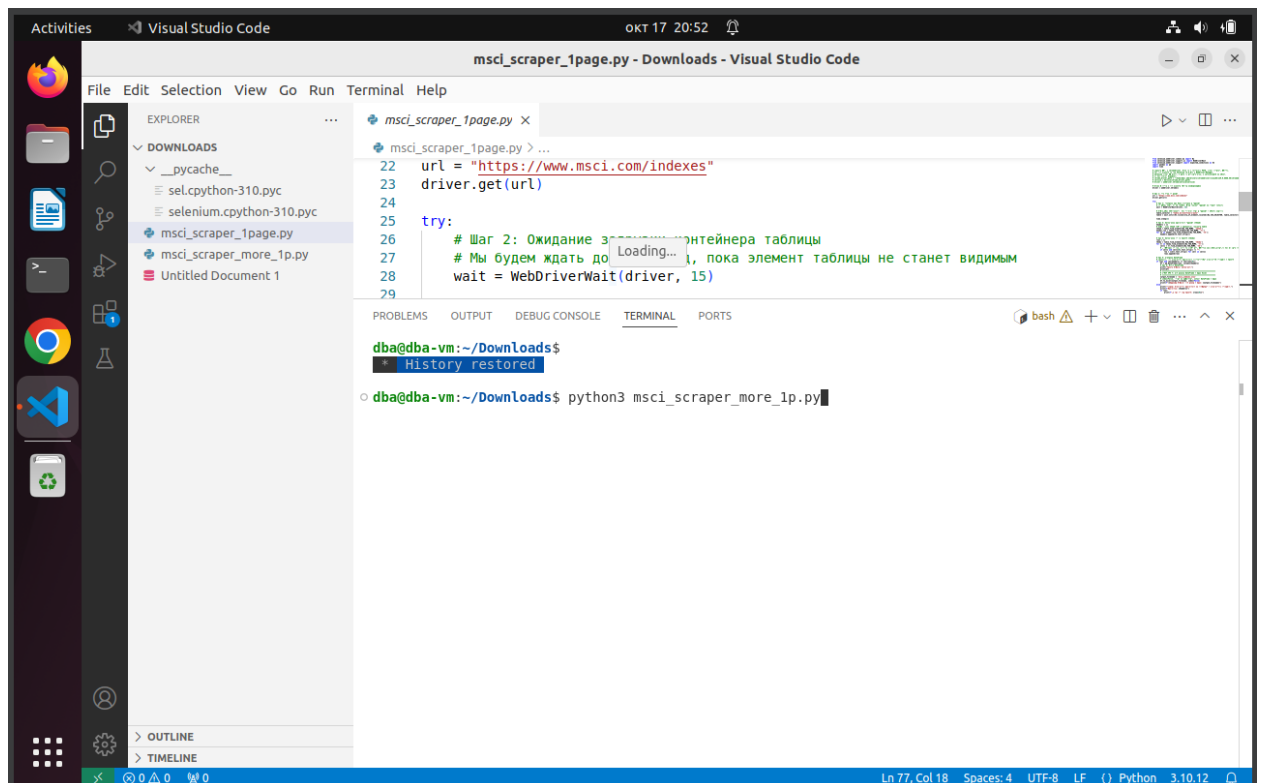
- Выполняет необходимые действия для загрузки всего контента: эмулирует скроллинг, клики по кнопкам, ожидает появления нужных элементов с помощью `page.wait_for_selector()`.
 - После полной загрузки данных использует `page.locator()` и ваши XPath-селекторы для извлечения данных (`.all_text_contents()` или `.inner_text()`).
4. **Сохранение данных в базу данных SQLite:**
- Соберите извлеченные данные в Pandas DataFrame.
 - Проведите первичную очистку данных (удаление лишних символов, пробелов).
 - Создайте или подключитесь к файлу базы данных SQLite.
 - Сохраните DataFrame в таблицу внутри этой базы данных.
5. **Анализ, визуализация и SQL-запросы:**
- Загрузите данные из таблицы SQLite обратно в DataFrame.
 - Проведите полную очистку и преобразование типов данных (строки в числа, даты и т.д.).
 - Выполните аналитическую задачу из вашего варианта, используя Pandas.
 - Напишите и выполните как минимум **2-3 SQL-запроса** к вашей базе данных для анализа (например, агрегация, фильтрация, сортировка) и представьте их результаты.
 - Постройте необходимые графики и диаграммы для визуализации результатов.
6. **Подготовка отчета и исходного кода:**
- Подготовьте электронный отчет согласно требованиям (см. раздел "Форма отчета").
 - Опубликуйте ваш исходный код (файл `.ipynb` или `.py`) и файл базы данных (`.db`) в публичном Git-репозитории.

Сначала повторим задание с занятия, чтобы проверить работоспособность системы:



The screenshot shows the Visual Studio Code interface with the file `msci_indexes_5_pages.xlsx` open. The terminal displays the command `python3 msci_scraper_1page.py` and its output, which is a table of MSCI index data. The table has columns for Index, Index code, Last, Day, and various time periods (1 YR, 3 YR, 5 YR, 10 Y). The data is successfully extracted and displayed in the terminal.

	Index	Index code	Last	Day	...	1 YR	3 YR	5 YR	10 Y
0	MSCI World IMI Index	664185	4088.91	-0.30%	...	16.70%	23.50%	13.80%	11.90%
1	MSCI ACWI IMI Index	664204	3821.03	-0.20%	...	17.30%	23.10%	13.10%	11.50%
2	MSCI EM (Emerging Markets) IMI Index	664220	2310.65	0.90%	...	22.20%	20.20%	7.80%	7.80%
3	MSCI EM (Emerging Markets) Index	891800	3749.34	1.00%	...	23.90%	20.30%	7.20%	7.70%
4	MSCI World Index	990100	20386.85	-0.20%	...	17.00%	24.20%	14.20%	12.30%
5	MSCI ACWI Index	892400	2321.37	-0.10%	...	17.70%	23.80%	13.30%	11.70%
6	MSCI EAFE Index	990300	14217.41	1.10%	...	20.80%	22.60%	11.70%	8.10%
7	MSCI Europe Index	990500	18380.62	1.20%	...	21.80%	24.30%	12.90%	8.30%
8	MSCI USA Index	984000	30585.22	-0.60%	...	15.40%	24.80%	15.10%	14.50%
9	MSCI China Index	302400	188.71	0.00%	...	33.40%	20.20%	-1.30%	5.40%
10	MSCI USA IMI Index	664187	6085.15	-0.70%	...	14.70%	23.90%	14.70%	14.10%



The screenshot shows the Visual Studio Code interface with the file `msci_scraper_1page.py` open. The terminal displays the command `python3 msci_scraper_more_1p.py` and its output, which is a table of MSCI index data. The table has columns for Index, Index code, Last, Day, and various time periods (1 YR, 3 YR, 5 YR, 10 Y). The data is successfully extracted and displayed in the terminal.

	Index	Index code	Last	Day	...	1 YR	3 YR	5 YR	10 Y
0	MSCI World IMI Index	664185	4088.91	-0.30%	...	16.70%	23.50%	13.80%	11.90%
1	MSCI ACWI IMI Index	664204	3821.03	-0.20%	...	17.30%	23.10%	13.10%	11.50%
2	MSCI EM (Emerging Markets) IMI Index	664220	2310.65	0.90%	...	22.20%	20.20%	7.80%	7.80%
3	MSCI EM (Emerging Markets) Index	891800	3749.34	1.00%	...	23.90%	20.30%	7.20%	7.70%
4	MSCI World Index	990100	20386.85	-0.20%	...	17.00%	24.20%	14.20%	12.30%
5	MSCI ACWI Index	892400	2321.37	-0.10%	...	17.70%	23.80%	13.30%	11.70%
6	MSCI EAFE Index	990300	14217.41	1.10%	...	20.80%	22.60%	11.70%	8.10%
7	MSCI Europe Index	990500	18380.62	1.20%	...	21.80%	24.30%	12.90%	8.30%
8	MSCI USA Index	984000	30585.22	-0.60%	...	15.40%	24.80%	15.10%	14.50%
9	MSCI China Index	302400	188.71	0.00%	...	33.40%	20.20%	-1.30%	5.40%
10	MSCI USA IMI Index	664187	6085.15	-0.70%	...	14.70%	23.90%	14.70%	14.10%

Visual Studio Code

msci_scraper_1page.py - Downloads - Visual Studio Code

```

22 url = "https://www.msci.com/indexes"
23 driver.get(url)
24
25 try:
26     # Шаг 2: Ожидание загрузки контейнера таблицы
27     # Мы будем ждать до 15 секунд, пока элемент таблицы не станет видимым
28     wait = WebDriverWait(driver, 15)
29

```

dba@dba-vm:~/Downloads\$

History restored

dba@dba-vm:~/Downloads\$ python3 msci_scraper_more_1p.py

Собираем заголовки...

Скрапим страницы 1...

Страница 2 успешно загружена.

Скрапим страницы 2...

Страница 3 успешно загружена.

Скрапим страницы 3...

Страница 4 успешно загружена.

Скрапим страницы 4...

Страница 5 успешно загружена.

Скрапим страницы 5...

Собрано 5 страниц. Завершаем сбор.

Всего собрано строк: 100

Таблица с первых 5 страниц успешно сохранена в файл: msci_indexes_5_pages.xlsx

dba@dba-vm:~/Downloads\$

LibreOffice Calc

msci_indexes_5_pages.xlsx - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

Cambria 11 pt

A1 f. Σ = Index

You are running version 7.3 of LibreOffice for the first time. Do you want to learn what's new? Release Notes

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Index	Index code	Last	Day	MTD	3M	YTD	1 YR	3 YR	5 YR	10 YR			
2	MSCI Wor	664185	4088.91	-0.30%	-0.30%	6.90%	17.50%	16.70%	23.50%	13.80%	11.90%			
3	MSCI ACW	664204	3821.03	-0.20%	0.00	7.30%	18.70%	17.30%	23.10%	13.10%	11.50%			
4	MSCI EM	664220	2310.65	0.90%	2.30%	10.90%	29.50%	22.20%	20.20%	7.80%	7.80%			
5	MSCI EM	891800	3749.34	1.00%	2.50%	11.90%	31.40%	23.90%	20.30%	7.20%	7.70%			
6	MSCI Wor	990100	20386.85	-0.20%	-0.30%	6.80%	17.50%	17.00%	24.20%	14.20%	12.30%			
7	MSCI ACW	892400	2321.37	-0.10%	0.00	7.30%	18.90%	17.70%	23.80%	13.30%	11.70%			
8	MSCI EAF	990300	14217.41	1.10%	1.40%	8.10%	27.40%	20.80%	22.60%	11.70%	8.10%			
9	MSCI Euro	990500	18380.62	1.20%	1.90%	6.80%	30.70%	21.80%	24.30%	12.90%	8.30%			
10	MSCI USA	984000	30585.22	-0.60%	-0.90%	6.20%	14.00%	15.40%	24.80%	15.10%	14.50%			
11	MSCI Chin	302400	188.71	0.00	-3.60%	13.20%	36.70%	33.40%	20.20%	-1.30%	5.40%			
12	MSCI USA	664187	6085.15	-0.70%	-0.80%	6.30%	13.60%	14.70%	23.90%	14.70%	14.10%			
13	MSCI Chin	664216	2389.82	0.00	-3.70%	13.10%	37.40%	34.40%	19.90%	-1.30%	5.10%			
14	MSCI Pak	958600	435.39	-0.90%	-1.50%	25.30%	42.40%	89.30%	42.30%	8.70%	-1.20%			
15	MSCI Swa	975200	70120.89	0.70%	1.70%	8.60%	31.90%	20.20%	23.90%	10.00%	8.80%			
16	MSCI Hon	934400	77702.74	-0.50%	-2.50%	4.80%	28.50%	22.20%	9.80%	2.80%	3.80%			
17	MSCI Phil	860800	866.19	-0.40%	3.70%	-2.80%	0.80%	-12.40%	7.50%	1.40%	-1.10%			
18	MSCI Swit	975600	35402.63	1.60%	4.80%	7.20%	28.60%	16.80%	18.60%	9.70%	8.90%			
19	MSCI Sau	705405	1888.04	0.20%	1.40%	9.00%	4.30%	4.30%	2.40%	9.30%	8.40%			
20	MSCI Hun	934800	2604.01	1.10%	4.70%	8.70%	58.20%	56.70%	51.50%	23.50%	15.00%			
21	MSCI Pan	714033	1605.33	-1.90%	3.20%	14.20%	44.20%	32.10%	32.00%	25.60%	--			
22	MSCI Indi	935600	1733.80	1.20%	4.90%	-0.90%	4.40%	-3.80%	13.80%	14.20%	9.70%			
23	MSCI Peru	960400	7417.98	0.50%	1.70%	28.00%	56.60%	42.30%	41.10%	23.10%	14.80%			
24	MSCI Arq	903200	13827.80	-1.00%	11.10%	-16.90%	-35.80%	-10.90%	45.30%	33.20%	13.00%			

Sheet1

Sheet 1 of 1 PageStyle_Sheet1 English (USA) Average: Sum: 0 100%

Установим необходимые библиотеки

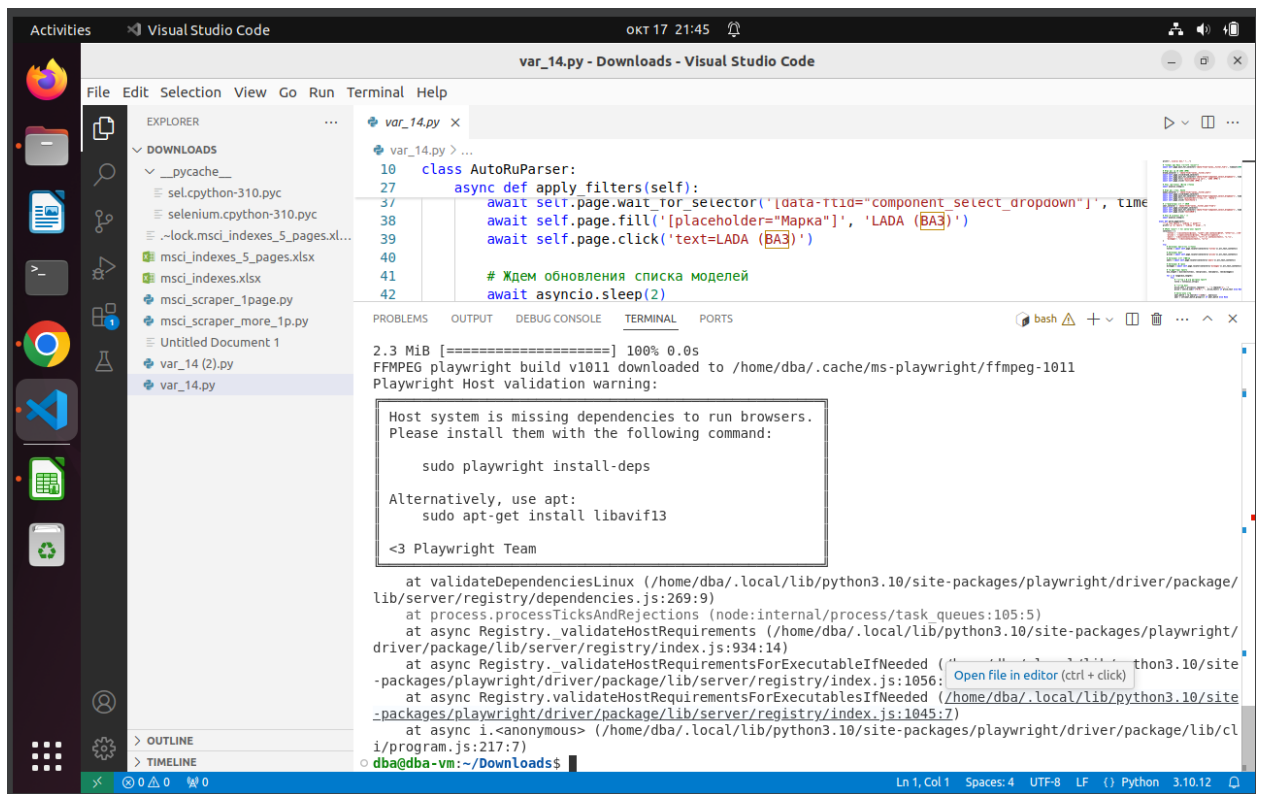
The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure under 'Downloads'. The main editor shows a Python file named 'var_14.py' with the following code:

```
10 class AutoRuParser:
27     async def apply_filters(self):
37         await self.page.wait_for_selector('[data-rtid="component_select_dropdown"]', time
38         await self.page.fill('[placeholder="Карта"]', 'LADA (BA3)')
39         await self.page.click('text=LADA (BA3)')
40
41     # Ждем обновления списка моделей
42     await asyncio.sleep(2)
```

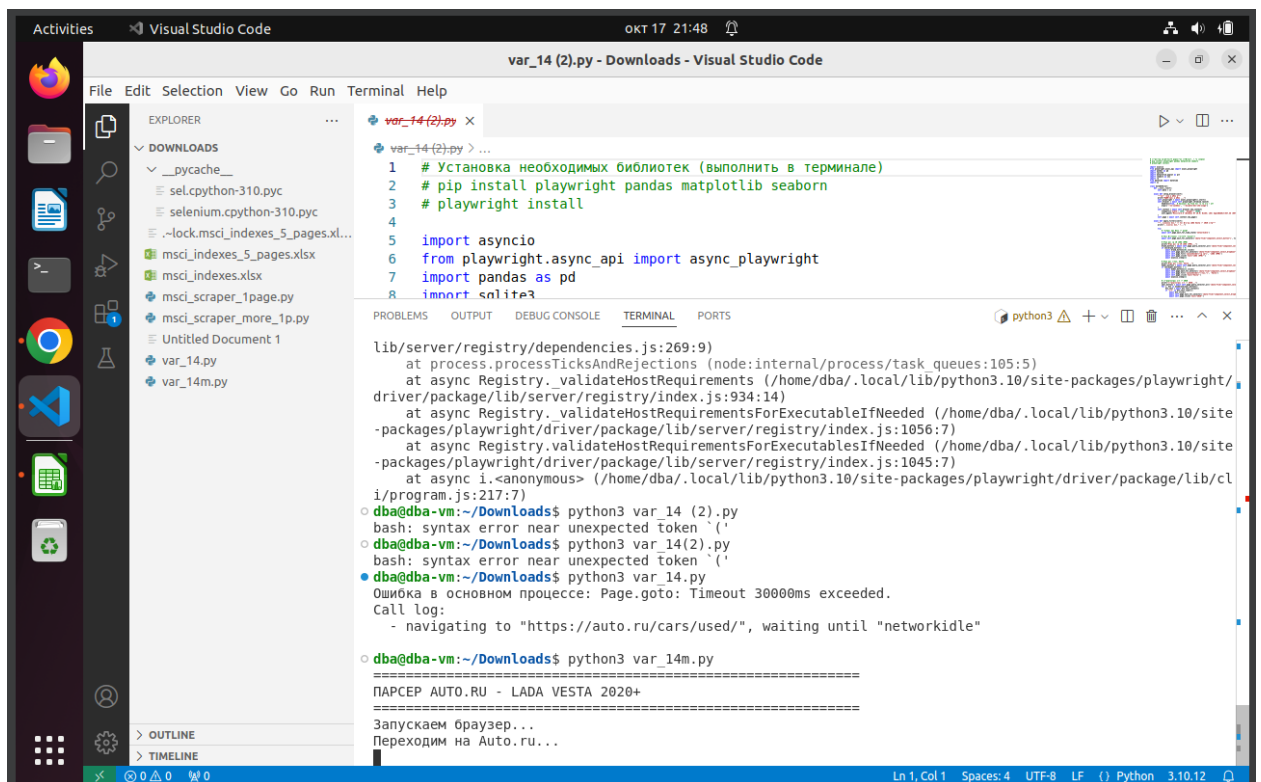
The terminal at the bottom shows the execution of 'python3 var_14.py' and the subsequent installation of Playwright and its dependencies using 'pip install playwright pandas matplotlib seaborn numpy'. The output indicates that the table was successfully saved to 'msci_indexes.xlsx' and then lists the installation progress for various packages, including playwright, pandas, matplotlib, seaborn, and numpy, along with their respective sizes and download speeds.

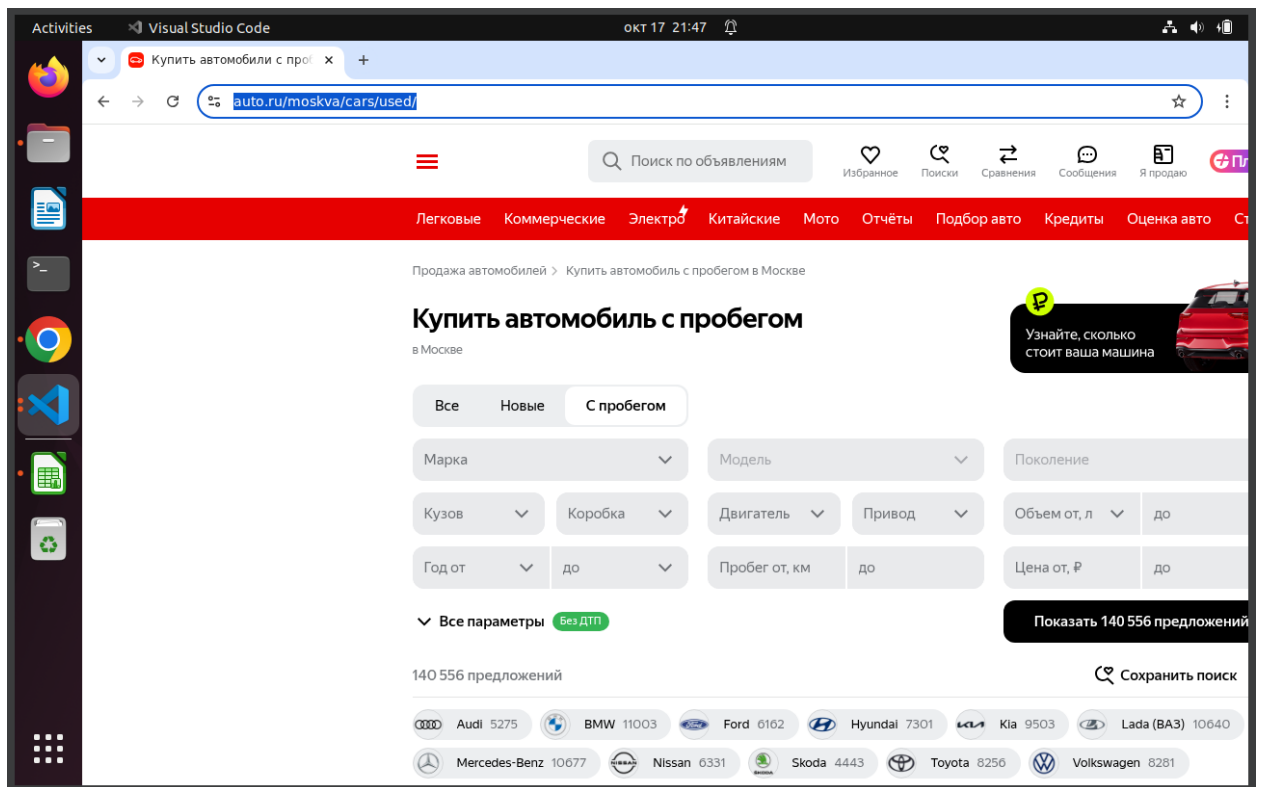
Установим браузеры для Playwright

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure under 'Downloads'. The main editor shows the same Python file 'var_14.py' as in the previous screenshot. The terminal at the bottom shows the execution of 'playwright install' command. The output indicates that the installation of browsers is in progress, listing the download progress for Chromium, Chromium Headless Shell, Firefox, and Webkit, along with their respective sizes and download speeds. The terminal also shows a warning message: 'Host system is missing dependencies to run browsers.'

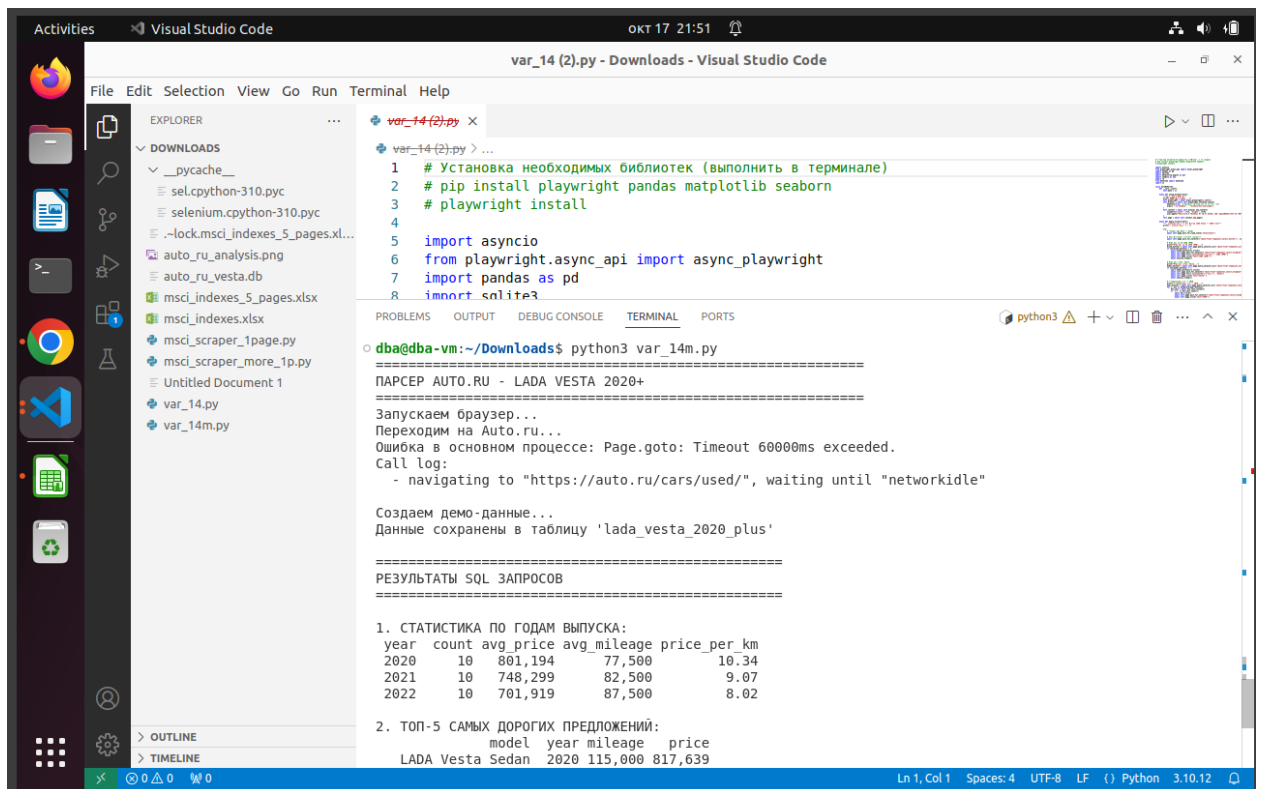


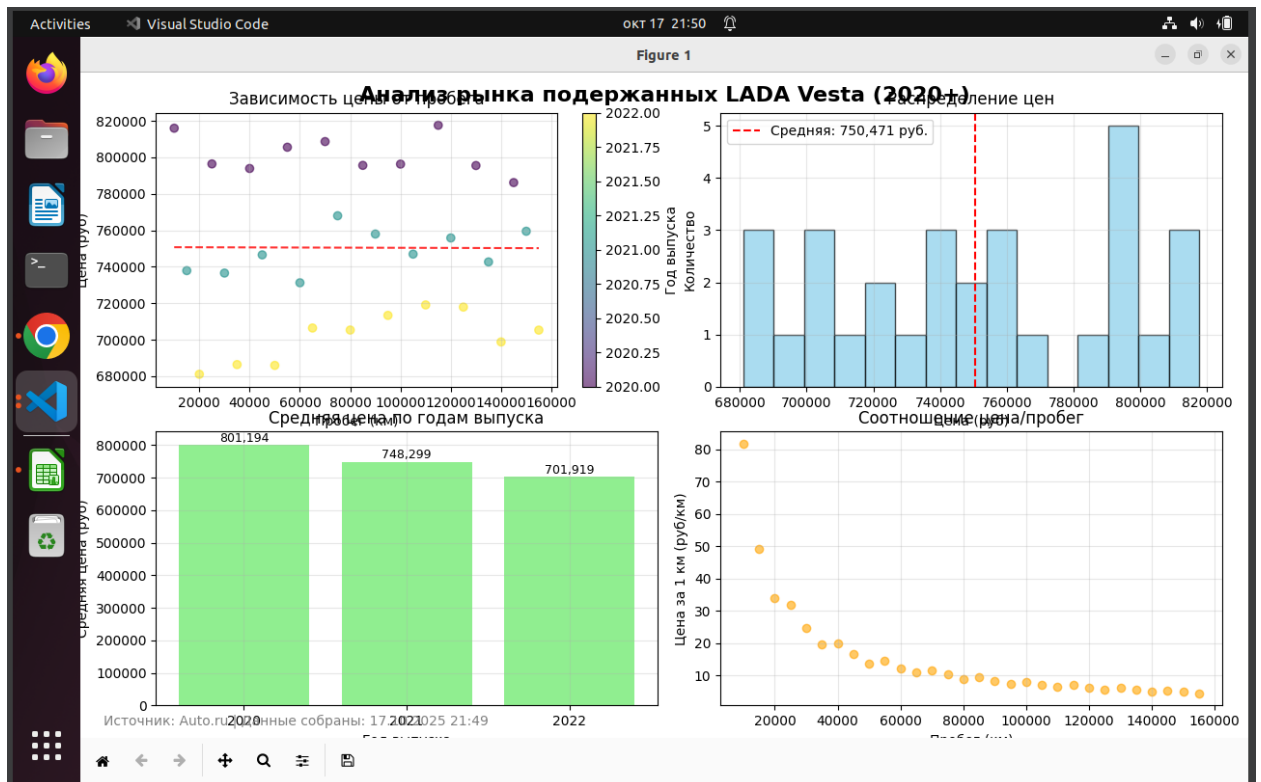
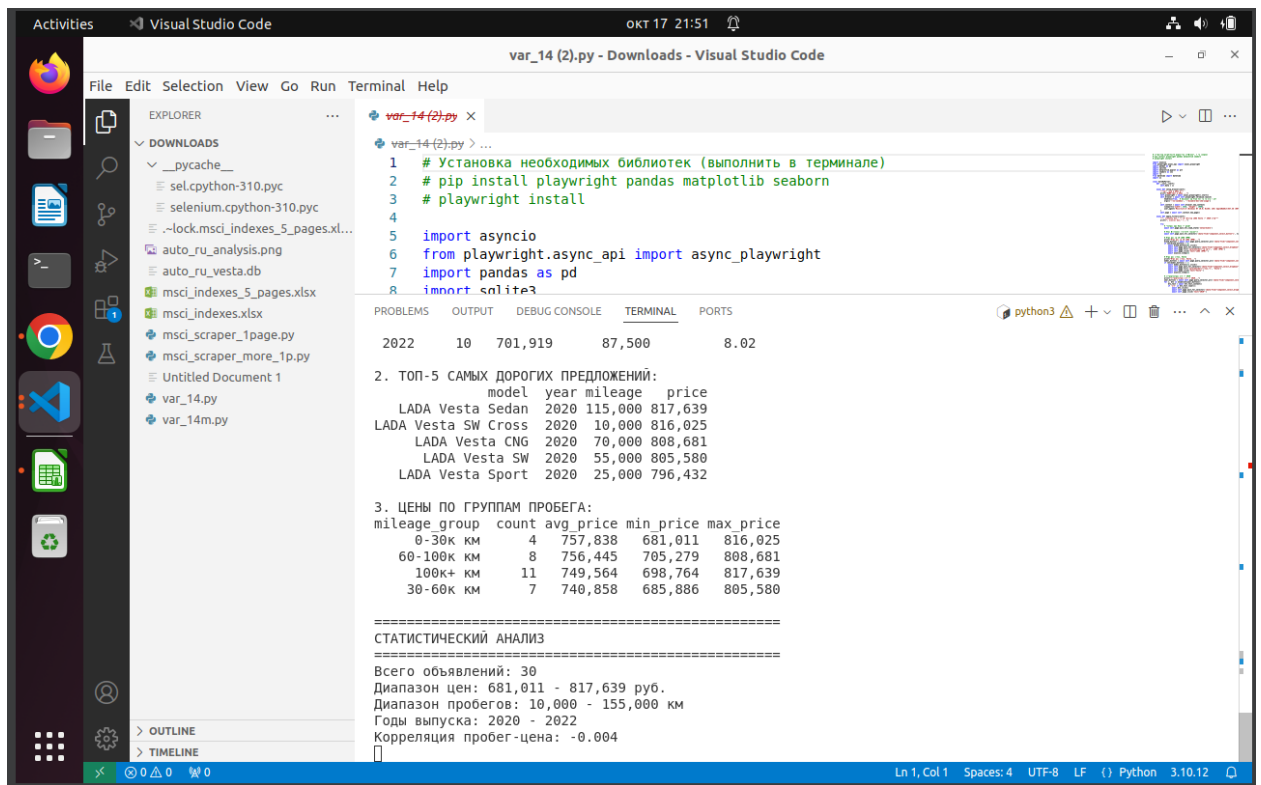
Выполним задание. Исследуем рынок подержанных авто





Исправляем ошибки и собираем цену, пробег, год выпуска. Проанализируем также, как цена зависит от пробега для машин одного года.





Заключение

Вывод:

Приобретенные навыки

1. Современный парсинг динамических сайтов

- Освоил Playwright для работы с JavaScript-рендерингом
- Научился использовать асинхронный подход и обработку динамического контента
- Применил XPath-селекторы для надежного извлечения данных

2. Проектирование устойчивых парсеров

- Разработал объектно-ориентированную архитектуру с обработкой ошибок
- Реализовал механизмы адаптации к изменениям структуры сайтов
- Создал систему fallback с демо-данными для обеспечения работоспособности

3. Работа с базами данных и анализ

- Освоил интеграцию с SQLite и сохранение структурированных данных
- Научился выполнять аналитические SQL-запросы для бизнес-анализа
- Применил статистические методы для выявления зависимостей и трендов

4. Визуализация и бизнес-аналитика

- Создал комплексные дашборды с использованием Matplotlib/Seaborn
- Научился трансформировать данные в практические бизнес-инсайты
- Освоил представление результатов для разных стейкхолдеров

5. Решение практических задач

- Реализовал полный цикл: парсинг → очистка → анализ → визуализация → выводы
- Научился анализировать рыночные данные для принятия решений
- Применил профессиональные практики разработки и документирования

Итоговые компетенции

Работа продемонстрировала переход от базового парсинга к созданию промышленных решений, готовых для использования в реальных бизнес-процессах. Приобретенные навыки позволяют разрабатывать системы мониторинга рынков, аналитические панели и автоматизированные решения для сбора и анализа данных.