Comparative market analysis of Apple products in US and Switzerland

Final report

Group 51
Xuekai Li
Dmytro Rudyka
Sasa Ljubisavljevic

HSLU
Master in Applied Information and Data Science
Module: Data collection, Integration and Pre-processing

April 2023, Lucerne

Table of contents

Introduction

Subject area

Understanding price differences for goods and services in different markets is crucial for businesses and customers. This is particularly relevant in the technology industry, where Apple products are highly popular and often carry a premium price tag. In this project, we aimed to compare the prices of Apple products between the US, Switzerland and Germany, important markets for Apple. To achieve this we used data integration, preparation and processing techniques to scrape and extract data from two typical retailers in each market, ABT.com in the US, FUST.ch in Switzerland and cyberport.de in Germany.

We have transformed and loaded this data into a suitable data storage system and performed exploratory data analysis to answer to the stated questions. We gained insights into the pricing of Apple products in different markets and are hoping to provide valuable information.

Why this topic

Everyone on our team loves Apple. Well, not everyone. But no one can deny the huge influence of this brand on the entire industry. Statistically, every second phone in Switzerland is an iPhone, and two of the team members have a MacBook. And we have all often heard that the cheapest place to buy it is in the USA. But how true is that? How much difference is there for different types of gadgets? This is very interesting to find out! That's why we decided to do this.

It's also interesting how much of the difference is influenced by the popularity of the products, which assessed using ratings, quantity and quality. We understand that when looking at products of the same brand a big contribution to the price is made by the manufacturer. But we investigated at independent shops to reduce this effect.

Sources

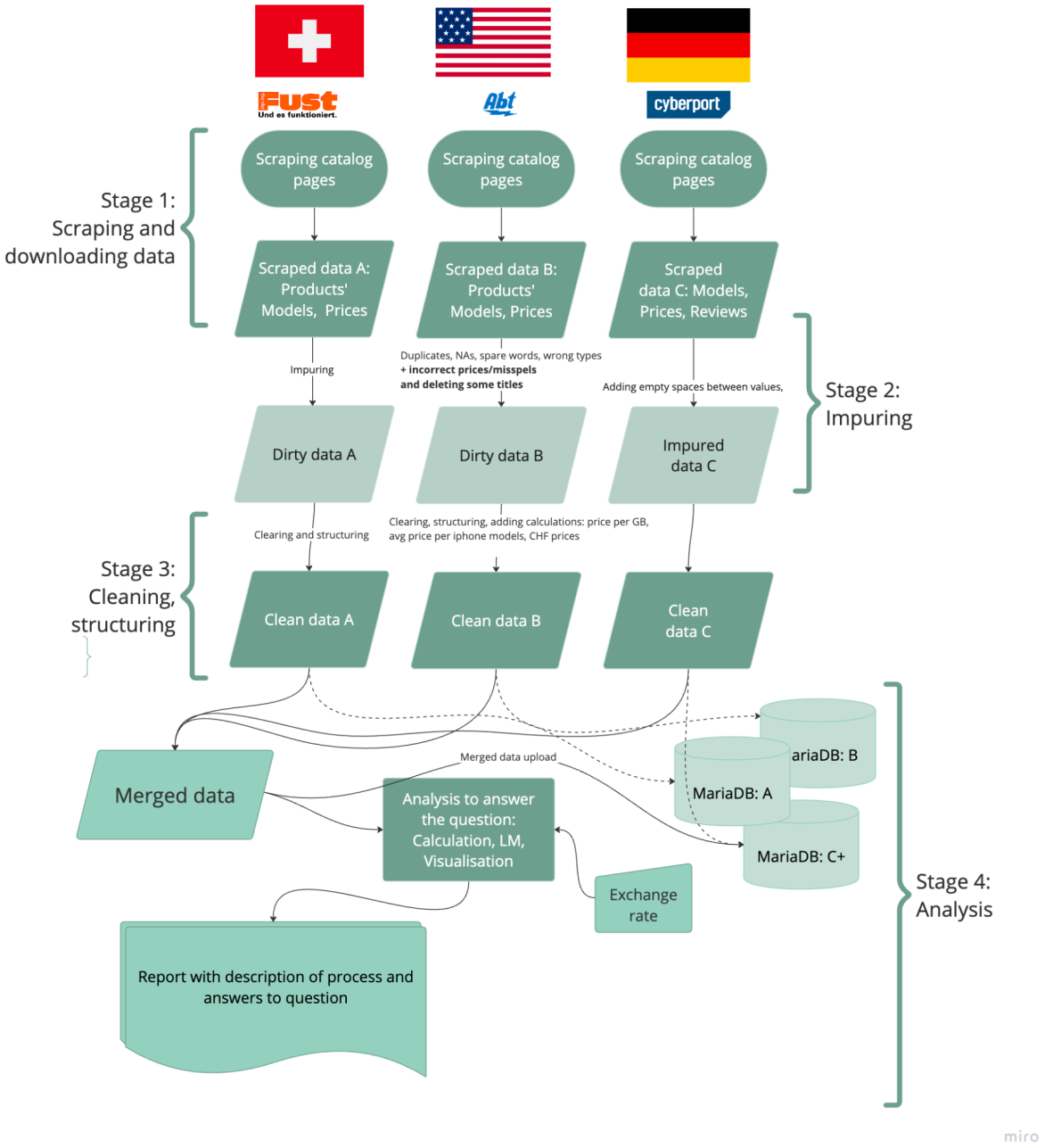
- Source A: scraping from www.fust.ch
- Source B: scraping from www.abt.com
- Source C: scraping from www.cyberport.de

Research questions

By analyzing the selected markets, Switzerland, US, and Germany we aim to answer the following questions:

1. Which country has higher prices for Apple products, the USA, Switzerland or Germany? And by how much?
2. Between Apple mobile phones, laptops and tablets, which category has the biggest price difference?
3. Is there a correlation between the popularity (in terms of number of reviews and overall rating) and the price difference of Apple products in the country with the lowest and highest prices?

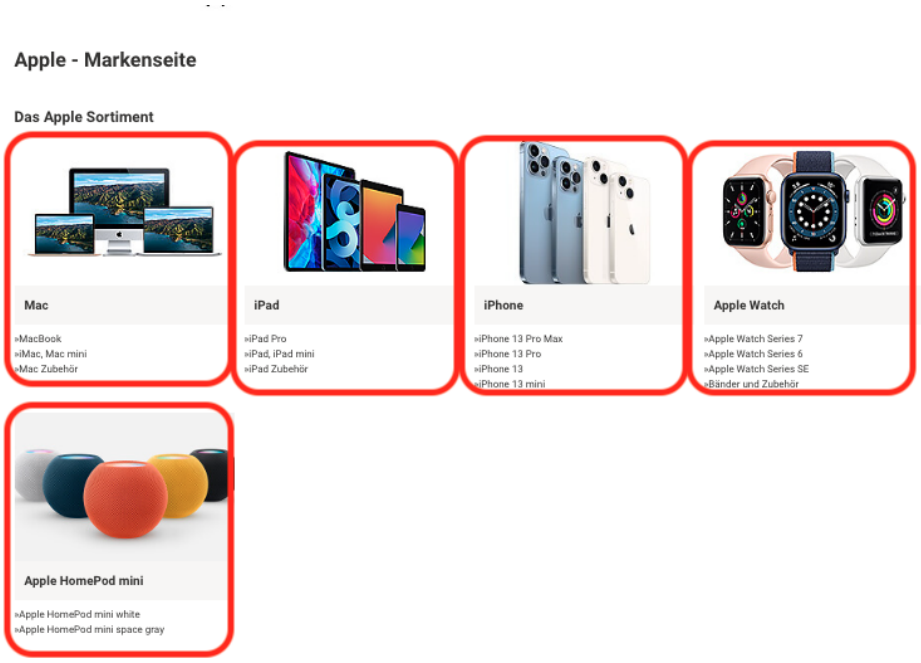
Workflow diagram



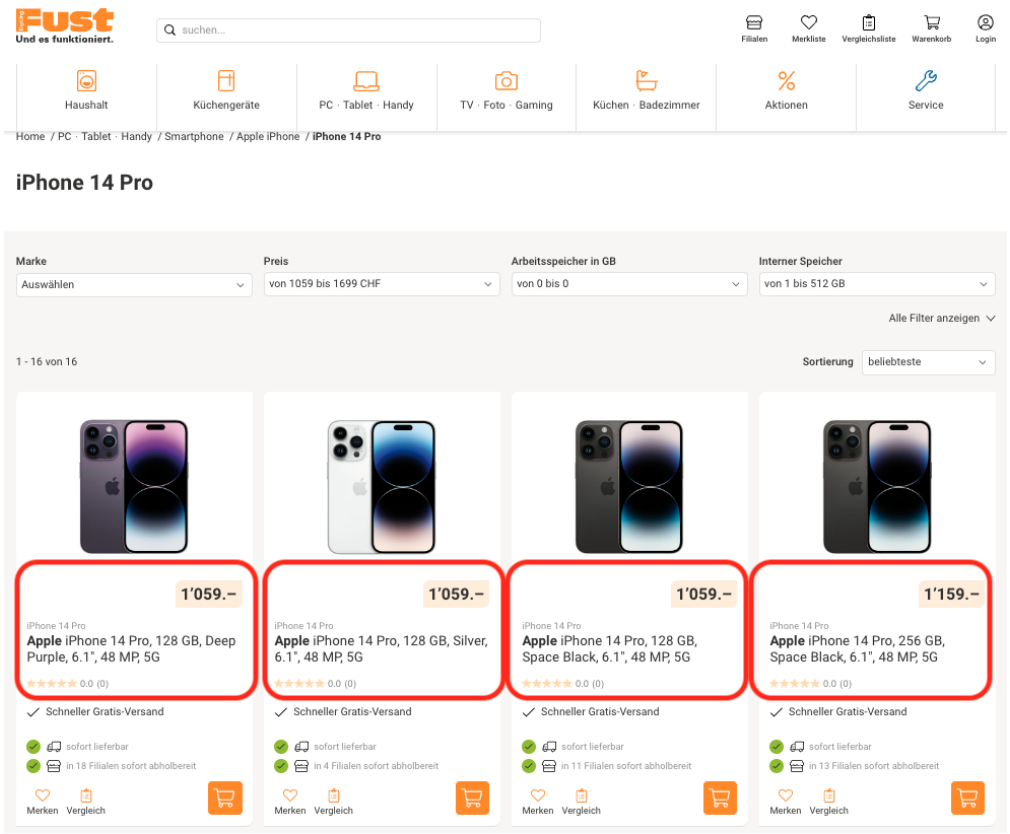
Description of students' tasks

Student A: Xuekai Li

Xuekai scraped data from web source A: https://www.fust.ch/de/marken/apple.html
1. category list:



2. item details.



- Eligibility check: www.fust.ch/robots.txt

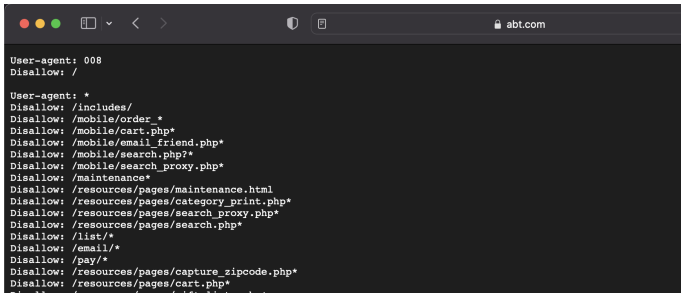
```
# robots.txt for https://www.fust.ch/

User-agent:*
Disallow:/admin/
#Disallow:/*jsessionid
```

- target attribute: category, model, price, ratings
- made dirty/cleans/enriches the data with Python/Pandas.
- uploaded last stage of personal data before merge to personal Maria DB.

Student B: Dmytro Rudyka

Dmytro scraped catalog pages of Apple products of US online shop atb.com, Aple products are here: https://www.abt.com/brand/apple

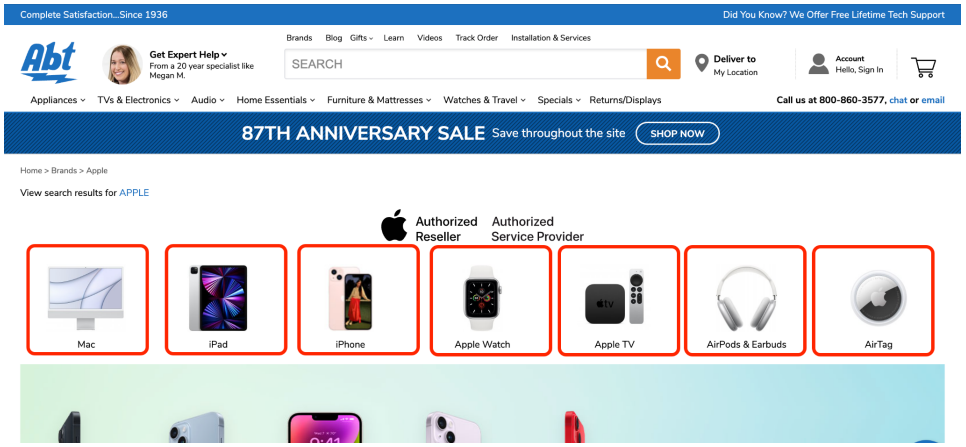
Eligibility check: After investigating the website robots txt it was not found any prohibitions for scraping needed pages:



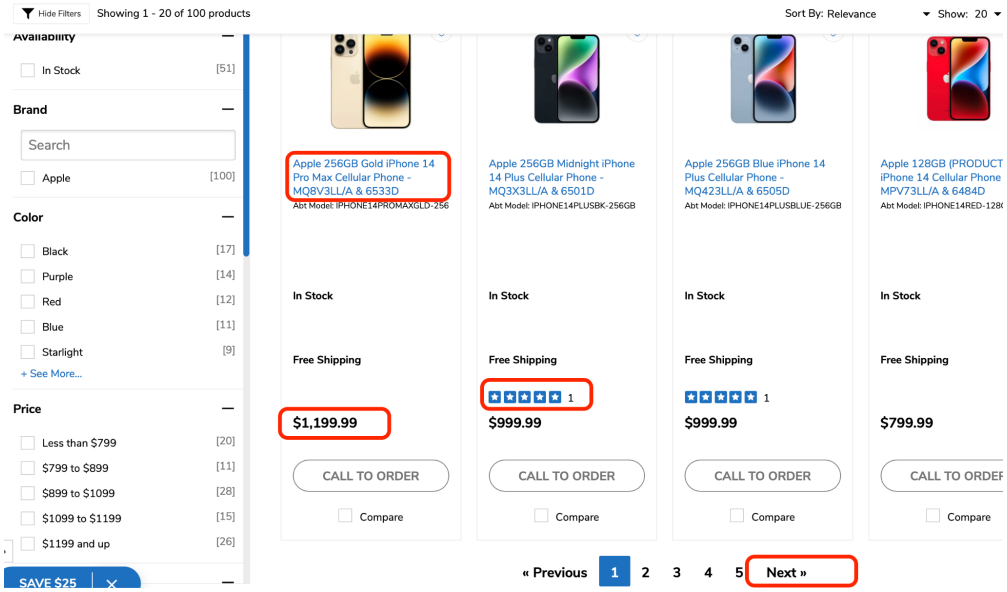
It was not scraped any of disallowed URLs or folders above.

Scraping procedure:

1. Brand catalog with categories is starting point which has the links to the listing with list of the products with the categories



After that (for some categories through interim page with subcategories) was got the list of pages with the products, for example:



2. Went through pages and scrapefollowing **List of of attributes**:
    - Scraped URL

- Category
- Title with model ID
- Price
- URL of product
- Number of reviews
- Rating

3. Impured
4. Cleaned, extracted and uploaded to MariaDB

Student C: Sasa Ljubisavljevic

> Initial plan was to download the Amazon data from BrightData.com. But that could not be achieved in four weeks. After long and slow communication with the vendors and consultants there, we were never able to get an adequate response.
>
> So we decided to scrape one more website with the same data but in new country, Germany. It was www.cyberport.de. The whole procedure is described below.

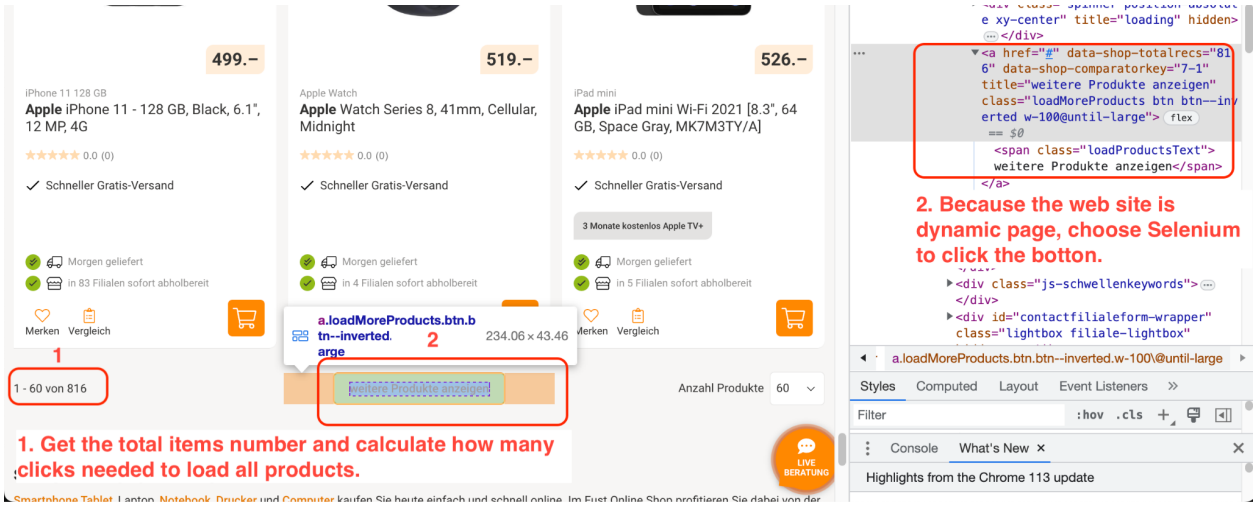
Description of the process

The processes of scraping each site and processing the information to answer the questions are described below.

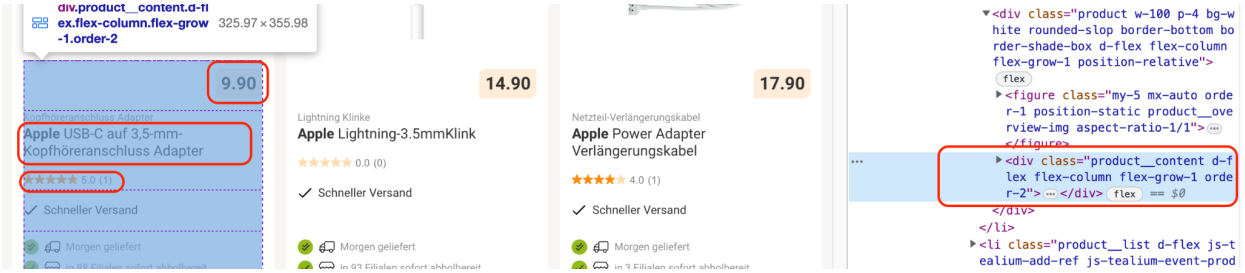
Student A: Xuekai Li — www.fust.ch

Extract

After I filtered out the product by apple brand and investigated the website(screenshot below), I found out the page is dynamic, so choose selenium to scrape the data.



After loading all products, I can use selenium to get the product content of each product, and use a for loop to extract the key attributes(screenshot below).



Finally I transpose the output as panda's data frame, and save as 'stage1.csv' file. From the screenshot below we know it is dirty and price and reviews need to normalize. you will see more details in the next phase.

	brand	title	reviews	rating	price
806	Apple	iPad Pro Wi-Fi + 5G 2021 [12.9", 1 TB, Space G...	(0)	0.0	1'999.–
807	Apple	iPad Pro Wi-Fi 2022 [11", 2 TB, Silver, MNXN3T...	(0)	0.0	2'139.–
808	Apple	iPad Pro Wi-Fi 2022 [11", 2 TB, Space Gray, MN...	(0)	0.0	2'139.–
809	Apple	iPad Pro Wi-Fi + 5G 2022 [11", 2 TB, Silver, M...	(0)	0.0	2'309.–
810	Apple	iPad Pro Wi-Fi + 5G 2022 [11", 2 TB, Space Gra...	(0)	0.0	2'309.–
811	Apple	iPad Pro Wi-Fi 2022 [12.9", 2 TB, Silver, MNY0...	(0)	0.0	2'458.–
812	Apple	iPad Pro Wi-Fi 2022 [12.9", 2 TB, Space Gray, ...	(0)	0.0	2'458.–
813	Apple	iPad Pro Wi-Fi + 5G 2022 [12.9", 2 TB, Silver,...	(0)	0.0	2'624.–
814	Apple	Mac Studio 2022 [M1 Ultra Chip, 64 GB RAM, 1 T...	(0)	0.0	4'369.–
815	Apple	Gift Card CHF 30.00	(0)	0.0	33.90

Transform

On 'fust' website didn't provide the product number for 'iPhone', so I need to enrich additional information from the title. That information will be used in the merge phase. I will follow the steps:

Enrichment

1. extract the category from the title.
2. extract the product_number, model, storage,(chip/size/color) from each category.

Cleaning:

1. drop the other category(accessory).
2. drop the outliers

Transformation:

1. normalize the reviews.
2. normalize the price, from string to numeric, and remove "'" and "-" from "2'199.-".
3. transform the storage format(from "64 GB" to "64GB").

the cleaned data will look like this:

	brand	title	reviews	rating	price	category	storage	model	sub	prod_n
798	Apple	iPhone 13 Pro Max - 1 TB, Silver, 6.7", 12 MP, 5G	0	0.0	1599.0	iPhone	1TB	13 Pro Max	Silver	-1
799	Apple	iPad Pro Wi-Fi 2022 [11", 1 TB, Silver, MNXL3T...	0	0.0	1689.0	iPad	1TB	Pro	11"	MNXL3TY/A
800	Apple	iPhone 13 Pro - 1 TB, Alpine Green, 6.1", 12 M...	0	0.0	1737.0	iPhone	1TB	13 Pro	Alpine Green	-1
801	Apple	iPad Pro Wi-Fi + 5G 2022 [12.9", 512 GB, Silve...	0	0.0	1744.0	iPad	512GB	Pro	12.9"	MP233TY/A
802	Apple	iMac 2020 [27", Intel Core i5, 16 GB RAM, 256 ...	0	0.0	1749.0	iMac	256GB	2020	i5	-1
803	Apple	iPhone 13 Pro Max - 1 TB, Gold, 6.7", 12 MP, 5G	0	0.0	1749.0	iPhone	1TB	13 Pro Max	Gold	-1
804	Apple	iPhone 13 Pro Max - 1 TB, Alpine Green, 6.7", ...	0	0.0	1837.0	iPhone	1TB	13 Pro Max	Alpine Green	-1
805	Apple	iPad Pro Wi-Fi + 5G 2022 [11", 1 TB, Silver, M...	0	0.0	1869.0	iPad	1TB	Pro	11"	MNYK3TY/A
806	Apple	iPad Pro Wi-Fi + 5G 2021 [12.9", 1 TB, Space G...	0	0.0	1999.0	iPad	1TB	Pro	12.9"	MHRA3TY/A
807	Apple	iPad Pro Wi-Fi 2022 [11", 2 TB, Silver, MNXN3T...	0	0.0	2139.0	iPad	2TB	Pro	11"	MNXN3TY/A

Uploading

In the final phase, I will use python to upload the data from dataframe. after I connected the 'CIP' database in mariaDB, I can create the table as 'fust_stage3'.

```
MariaDB [sys]> use CIP;
Reading table information fo
You can turn off this featur

Database changed
MariaDB [CIP]> show tables;
+----------------+
| Tables_in_CIP  |
+----------------+
| fust_stage3    |
| tbl_kontakt    |
+----------------+
```

```
MariaDB [CIP]> show columns from fust_stage3;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| brand    | varchar(255) | YES  |     | NULL    |                |
| title    | varchar(255) | YES  |     | NULL    |                |
| reviews  | int(11)      | YES  |     | NULL    |                |
| rating   | float        | YES  |     | NULL    |                |
| price    | float        | YES  |     | NULL    |                |
| category | varchar(255) | YES  |     | NULL    |                |
| storage  | varchar(255) | YES  |     | NULL    |                |
| model    | varchar(255) | YES  |     | NULL    |                |
| sub      | varchar(255) | YES  |     | NULL    |                |
| prod_n   | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
```

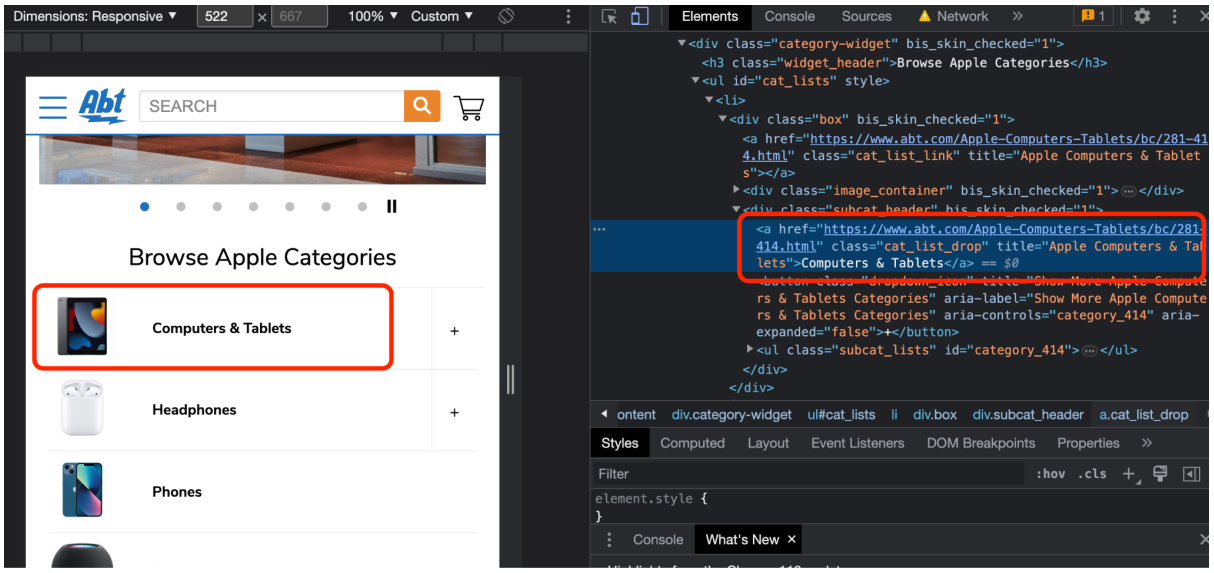
Finally use a for loop to insert each row from the dataframe. and the result is:

```
MariaDB [CIP]> select * from fust_stage3;
+----+-------+-----------------------------------------------------------------------+---------+--------+-------+----------+---------+---------+-----------+-----------+
| id | brand | title                                                                 | reviews | rating | price | category | storage | model   | sub       | prod_n    |
+----+-------+-----------------------------------------------------------------------+---------+--------+-------+----------+---------+---------+-----------+-----------+
|  1 | Apple | Watch SE, 44mm, Midnight                                              |       0 |      0 | 294.9 | Watch    | 44mm    | SE      | Midnight  | -1        |
|  2 | Apple | iPad Wi-Fi 2021 [10.2", 64 GB, Silver, MK2L3TY/A]                     |       0 |      0 |   298 | iPad     | 64GB    | 2021    | 10.2"     | MK2L3TY/A |
|  3 | Apple | iPad Wi-Fi 2021 [10.2", 64 GB, Space Gray, MK2K3TY/A]                 |       0 |      0 |   298 | iPad     | 64GB    | 2021    | 10.2"     | MK2K3TY/A |
|  4 | Apple | Watch Series 8, 41mm, Midnight                                        |       0 |      0 |   423 | Watch    | 41mm    | Series 8| Midnight  | -1        |
|  5 | Apple | Watch Series 8, 45mm, Silver                                          |       0 |      0 |   449 | Watch    | 45mm    | Series 8| Silver    | -1        |
|  6 | Apple | iPad Wi-Fi 2022 [10.9", 64 GB, Silver, MPQ03TY/A]                     |       0 |      0 |   466 | iPad     | 64GB    | 2022    | 10.9"     | MPQ03TY/A |
|  7 | Apple | iPad Wi-Fi + 4G 2021 [10.2", 64 GB, Space Gray, MK473TY/A]            |       0 |      0 |   479 | iPad     | 64GB    | 2021    | 10.2"     | MK473TY/A |
|  8 | Apple | iPad Wi-Fi 2021 [10.2", 256 GB, Space Gray, MK2N3TY/A]                |       0 |      0 |   498 | iPad     | 256GB   | 2021    | 10.2"     | MK2N3TY/A |
|  9 | Apple | Watch Series 8, 41mm, Cellular, Midnight                              |       0 |      0 |   519 | Watch    | 41mm    | Series 8| Midnight  | -1        |
| 10 | Apple | iPad mini Wi-Fi 2021 [8.3", 64 GB, Space Gray, MK7M3TY/A]             |       0 |      0 |   526 | iPad     | 64GB    | mini    | -1        | MK7M3TY/A |
| 11 | Apple | iPad Air Wi-Fi 2022 [10.9", 64 GB, Blue, MM9E3TY/A]                   |       0 |      0 |   534 | iPad     | 64GB    | Air     | 10.9"     | MM9E3TY/A |
| 12 | Apple | iPad Air Wi-Fi 2022 [10.9", 64 GB, Space Gray, MM9C3TY/A]             |       0 |      0 |   534 | iPad     | 64GB    | Air     | 10.9"     | MM9C3TY/A |
| 13 | Apple | iPad Air Wi-Fi 2022 [10.9", 64 GB, Purple, MME23TY/A]                 |       0 |      0 |   534 | iPad     | 64GB    | Air     | 10.9"     | MME23TY/A |
| 14 | Apple | iPad Wi-Fi 2022 [10.9", 256 GB, Silver, MPQ83TY/A]                    |       0 |      0 |   648 | iPad     | 256GB   | 2022    | 10.9"     | MPQ83TY/A |
| 15 | Apple | iPad Air Wi-Fi 2022 [10.9", 256 GB, Purple, MME63TY/A]                |       0 |      0 |   773 | iPad     | 256GB   | Air     | 10.9"     | MME63TY/A |
```

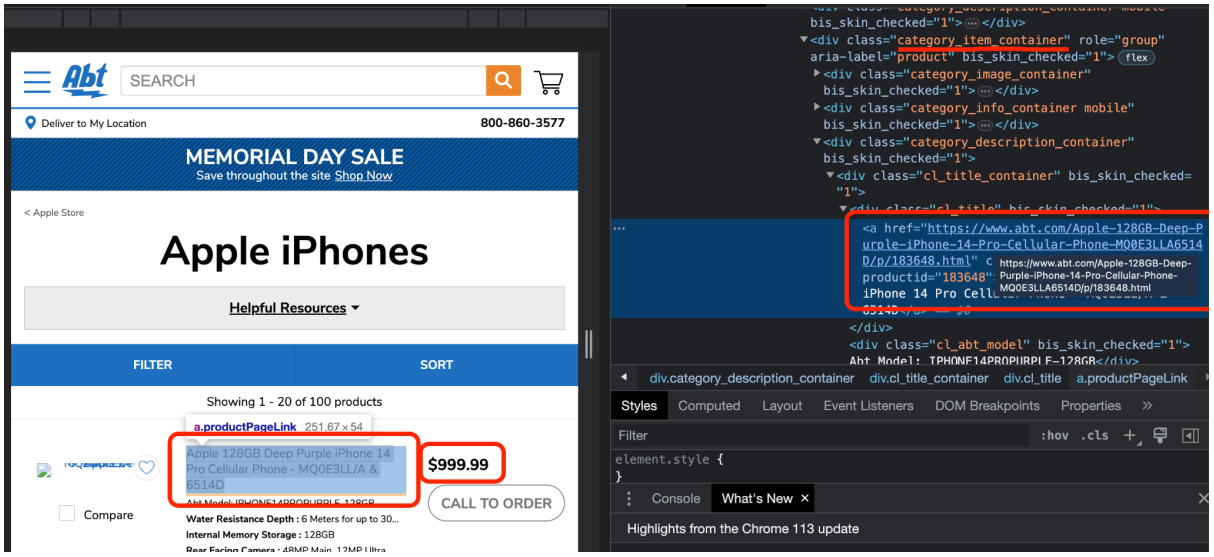
Student B: Dmytro Rudyka — www.abt.com

Extraction

Started from the main page of Apple https://www.abt.com/brand/apple. From which URLs to scrape was gathered.



After gathering URLs from the main page and pages with subcategories was founded proper classes to scrape products properties:



Scrape the products within all category pages the following function with recursion was applied:

```
# function to scrape products and next url altogether, and write the result at once in CSV

def scrape_prods_to_csv(url, category): #category will be stored in CSV as well, because it will be used on next s
    print(category)
    with open('../data/abt_rudyka_stage1.csv', 'a', newline='') as csvfile: #method is in order to just mini
        writer = csv.writer(csvfile, delimiter=';', quotechar='"', quoting=csv.QUOTE_MINIMAL) #quoting= just in ca
        sleep = random.randint(1,5)
        print('Waiting for', sleep, 'and then scrape', url) # to be informed of the process
        time.sleep(sleep)
        r = requests.get(url)
        doc = BeautifulSoup(r.text, "html.parser")
        products = doc.select(".category_item_container")
        for product in products:
            title = product.select_one('.cl_title').text.strip() #title of the product
            price = product.select_one('.pricing-item-price').text.strip() #all the content of tag with price
            product_url = product.select_one('.productPageLink').attrs['href'] #url of product page
            try:
                rating = product.select_one('.abt-reviews-rating .sr-only').text.strip()
                review = product.select_one('.abt-reviews-num-reviews').text.strip()
            except AttributeError:
                rating = review = None
            writer.writerow([url, category, title, price, product_url, rating, review])
    nextbutton = "Next »"  #scraping URL of next page
    links = doc.find_all('a', string=nextbutton) #list of urls with only one element
    if links:
        url = links[0].attrs['href']
        print('naxt page to scrape: ', url)
        scrape_prods_to_csv(url, category) #simple recursion. Because why not
```

This procedure gathered all needed attributes of products (title with model ID, price, number of reviews and average rating) and the URL of next page. Except that function saved all the data into a file straight off.

Impurifying

Data already had natural impurities: duplicates, NA values, incorrect types, irrelevant data or contaminated data for some values, strings with words in prices (e.g. 'Your Price: $799.00'), reviews and ratings ('5 out of 5 stars') where only numbers needed. But this is only 4 differnt types of impurites. So it was add two additional impurifications:

1. Removing the dot from prices to get outliers (float prices, in this case, could be treated as multiplied to 100) ot misspels. Example 999.00$ → 99900$
2. Deleting titles. In case of this website, URL contains the same full title of the Product so the goal was to delete it and then recreate out of URL. Example of url: https://www.abt.com/Apple-iPad-9th-Generation-10.2-Inch-64GB-Wi-Fi-Space-Grey-2021-MK2K3LLA/p/168532.html

Cleaning, Transformation and Uploading

During cleaning was removed artificial impurities, extracted correct prices, ratings and reviews, set to correct types. And then excessive columns were dropped,

During the transformation phase was added three additional calculations:

1. Price per GB
2. The average price of the same mode in different colours. Yes, they can differ.
3. Added price in CHF

Then prepared for merging data was uploaded to MariaDB.

Student C: Sasa Ljubisavljevic — www.cyberport.de

Extract

In the extraction phase, data was collected from the website https://www.cyberport.de/apple-und-zubehoer.html using the web scraper browser extension tool.

This tool allowed me to define selectors and create a structured scraping process. Templates were created to navigate the website's clear structure and target the desired Apple products. Subcategories were utilized to ensure precise item targeting and facilitate integration with other datasets.

Impurifying and cleaning

To ensure data quality and remove unnecessary elements, impurities were addressed, and the data was cleaned.
Columns containing links for each element were dropped as they were not relevant to our analysis.
Rows with category values unrelated to our project were also removed. The data underwent thorough cleaning to standardize its format and eliminate unnecessary text and characters. This included cleaning newly obtained columns such as storage and rating, ensuring a consistent format for analysis.

Transform

Following the extraction and cleaning stages, the data underwent transformation to make it suitable for analysis. Columns were modified, rearranged, and structured to facilitate further processing. Unnecessary text and characters were removed to create a refined dataset. Textual data was converted to numerical values to enable subsequent analysis and formatting.

Uploading

After the data was transformed and cleaned, it was uploaded to a MariaDB database. A table named "cyber_port" was created in the database to store the scraped data. This allowed for efficient storage and management of the dataset. Prior to uploading, a validation process was conducted to ensure the accuracy and integrity of the data. This step ensured that the uploaded data was ready for analysis and met the requirements of the project.

Merging

We could merge the data by following steps:
1. prepare the data from 3 source by same process:
    a. get the model ‘number / storage / color’ for the Iphone category.

	brand	CH_title	CH_reviews	CH_rating	CH_price	category	storage	model	sub	prod_n	prod_n5
33	Apple	iPhone 13 - 512 GB, Red, 6.1", 12 MP, 5G	0	0.0	949.0	iPhone	512GB	13	Red	-1	-1
114	Apple	iPhone 11 - 64 GB, Black, 6.1", 12 MP, 4G	0	0.0	439.0	iPhone	64GB	11	Black	-1	-1
115	Apple	iPhone SE 3. Gen. - 64 GB, Midnight, 4.7" 12 M...	0	0.0	469.0	iPhone	64GB	SE 3	Midnight	-1	-1

b. create a column for the first 5 digits of the product number .

	brand	CH_title	CH_reviews	CH_rating	CH_price	category	storage	model	sub	prod_n	prod_n5
1	Apple	iPad Wi-Fi 2021 [10.2", 64 GB, Silver, MK2L3TY/A]	0	0.0	298.0	iPad	64GB	2021	10.2"	MK2L3TY/A	MK2L3
2	Apple	iPad Wi-Fi 2021 [10.2", 64 GB, Space Gray, MK2...	0	0.0	298.0	iPad	64GB	2021	10.2"	MK2K3TY/A	MK2K3
5	Apple	iPad Wi-Fi 2022 [10.9", 64 GB, Silver, MPQ03TY/A]	0	0.0	466.0	iPad	64GB	2022	10.9"	MPQ03TY/A	MPQ03

2. merge 3 data frame by the same process:

a. merge the CH and US data with the above 3 attributes for iPhone.

```
# merge the iphone data by  storage, model, and color.
iphone_merged = pd.merge(df1,df2, left_on=['storage','model','sub'], right_on=['iphone_storage','iphone_model','iphone_color'])
iphone_merged
```

Python

	brand	CH_title	CH_reviews	CH_rating	CH_price	category	storage	model	sub	prod_n	prod_n5	US_title	model_id	iphone_storage	iphone_model	i
0	Apple	iPhone 12 - 64 GB, Black, 6.1", 12 MP, 5G	0	0.0	619.0	iPhone	64GB	12	Black	-1	-1	Apple 64GB Black iPhone 12 Cellular Phone - MG...	MGEF3LL/A	64GB	12	

b. merge the other category by short product number.

```
# merge the other product by first 5 model number.
other_merged = pd.merge(df1,df2, left_on=['prod_n5'], right_on=['model_id5'])
other_merged
```

Python

	brand	CH_title	CH_reviews	CH_rating	CH_price	category	storage	model	sub	prod_n	prod_n5	US_title
0	Apple	iPad Wi-Fi 2021 [10.2", 64 GB, Silver, MK2L3TY/A]	0	0.0	298.0	iPad	64GB	2021	10.2"	MK2L3TY/A	MK2L3	Apple iPad (9th Generation) 10.2-Inch 64GB Wi-...

c. concat the ‘iphone_merged’ and ‘other_merged’ together.

3. subset the merged data with necessary columns. We are only interested in the price / reviews / ratings of 3 countries.

4. finally we create 2 columns to keep the ‘CH_price_in_dallor’ and ‘’DE_price_in_dallor’.

```
# take the exchange rate to exchange the price to dollar.
exchangerate = {'USD_CH':0.9046, 'EUR_USD': 1.0752}

# create two columns to keep the price in dollar.
final_merged['CH_price_in_dallor'] = (final_merged['CH_price']/exchangerate['USD_CH']).round(2)
final_merged['DE_price_in_dallor'] = (final_merged['DE_price'] * exchangerate['EUR_USD']).round(2)
final_merged
```

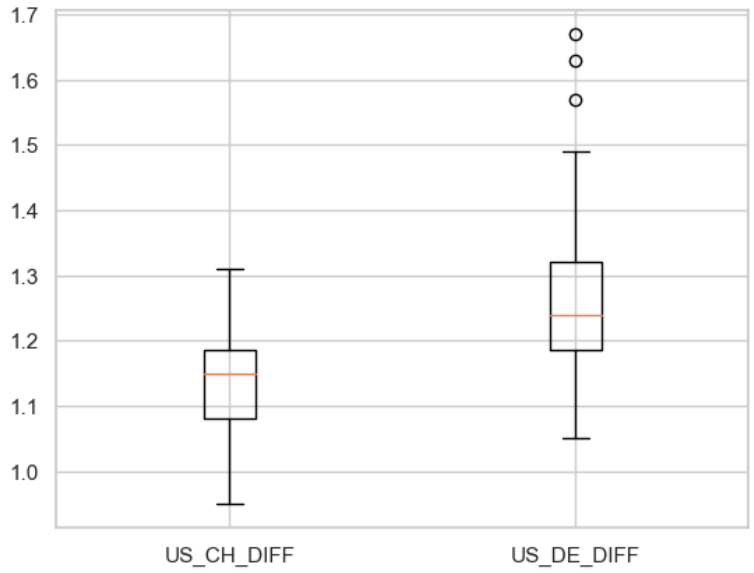
Analysis and answers to questions

Question 1: Which country has higher prices for Apple products, the USA or Switzerland? And by how much?

First, I use ‘US_price’ as the benchmark to normalize the price difference by following the code. and create 2 new columns to keep the results.

```
# nomalize the price different by persentage.

final_merged['US_CH_DIFF'] = (final_merged['CH_price_in_dallor']/final_merged['US_price']).round(2)
final_merged['US_DE_DIFF'] = (final_merged['DE_price_in_dallor']/final_merged['US_price']).round(2)
```

Then we could plot the boxplot of those two columns to get our result.



results: the price in CH and DE both higher than the price in the US. On average, CH price is 1.15 times higher than price in the US; DE price is 1.25 times higher than the price in US.

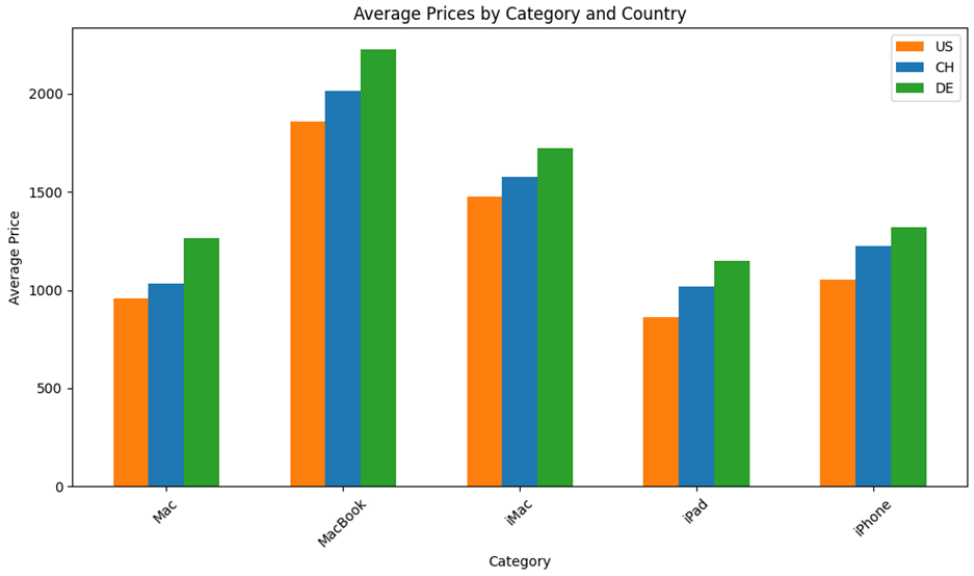
Question 2: Between Apple products categories, which has the biggest price difference?

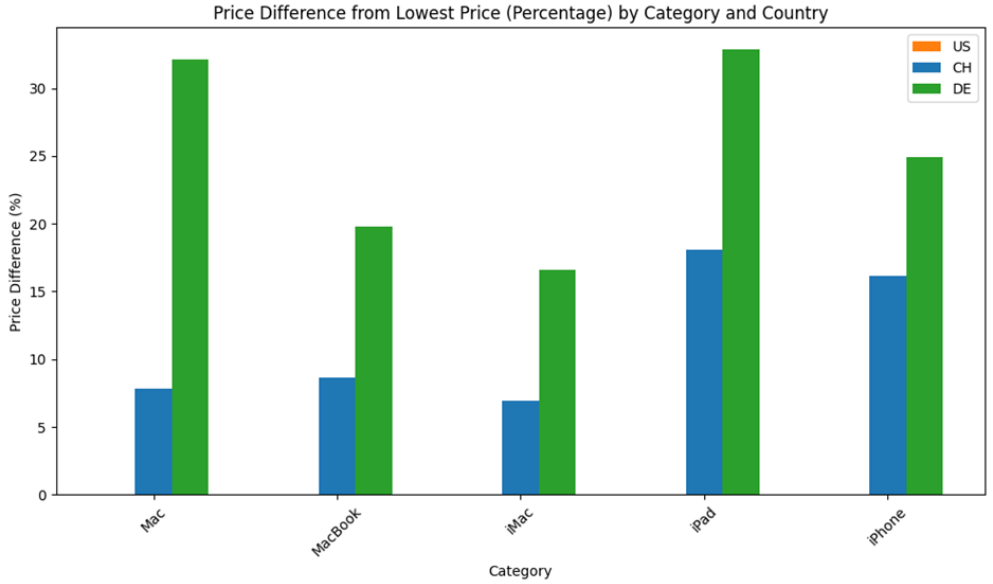
In order to answer the second question, we approached in the following way: we grouped all articles from the final dataset and calculated their average value for each country.
Then, in relation to the cheapest price from each category, we calculated the difference. Then we calculated their differences in relative values.

	Category	Country	Average Price	Price Difference (Abs)	Price Difference (%)
13	Mac	US	956.200000	0.000000	0.000000
12	Mac	CH	1030.954000	74.754000	7.817821
14	Mac	DE	1263.358000	307.158000	32.122778
7	MacBook	US	1856.200000	0.000000	0.000000
6	MacBook	CH	2016.272800	160.072800	8.623683
8	MacBook	DE	2223.682400	367.482400	19.797565
10	iMac	US	1476.777778	0.000000	0.000000
9	iMac	CH	1579.087778	102.310000	6.927921
11	iMac	DE	1721.633333	244.855556	16.580393
4	iPad	US	863.890364	0.000000	0.000000
3	iPad	CH	1020.321091	156.430727	18.107706
5	iPad	DE	1147.626545	283.736182	32.844004
1	iPhone	US	1055.596061	0.000000	0.000000
0	iPhone	CH	1225.871515	170.275455	16.130740
2	iPhone	DE	1318.958939	263.362879	24.949210

When comparing the average prices of Apple products across countries, the US generally offers the lowest prices for all categories. Switzerland tends to have moderately higher prices, while Germany consistently has the highest prices.

We then visualized the obtained results, which gives us a clear insight into the price differences between individual groups of items.





Results: The "Mac" and "iPad" category has the biggest price difference among Apple products. The price difference indicates the percentage difference in prices compared to the average price in the United States (US). The price difference value for the "Mac" and "iPad" category in Germany (DE) around 32%, which is the highest among all the categories listed.

Question 3: Is there a correlation or dependency between the popularity (in terms of Rating and number of reviews) and price differences of Apple products in the US and Germany?

We found some evidence that the difference in prices is dependent on Rating and number of reviews. P–value for both of them is less than 0.05. So within our scope with given shops, we found that:

- the larger rating the larger the relative difference in price. WIth every additional score the difference is growing on 13%.

- the larger the number of reviews the smaller relative difference in price we have. For one additional review we have drop in difference in prices for 0,18%

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             diff_price   R-squared:                       0.175
Model:                            OLS   Adj. R-squared:                  0.154
Method:                 Least Squares   F-statistic:                     8.352
Date:                Fri, 26 May 2023   Prob (F-statistic):           0.000512
Time:                        23:45:24   Log-Likelihood:                 72.140
No. Observations:                  82   AIC:                            -138.3
Df Residuals:                      79   BIC:                            -131.1
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          -0.3115      0.299     -1.040      0.301      -0.908       0.285
all_reviews    -0.0018      0.000     -3.626      0.001      -0.003      -0.001
avg_rating      0.1307      0.061      2.139      0.036       0.009       0.252
==============================================================================
Omnibus:                        4.493   Durbin-Watson:                   0.744
Prob(Omnibus):                  0.106   Jarque-Bera (JB):                3.748
Skew:                           0.420   Prob(JB):                        0.154
Kurtosis:                       2.376   Cond. No.                         889.
==============================================================================
```

Limitations

We know that only three shops have been analysed and it is quite hard to draw conclusions about the whole country. Moreover, there was not so big number of reviews for chosen products in the shops analysed, so it is unlikely that the conclusion and answer to the third question can be extrapolated to the whole countries and shops. But the approach itself and the code created can be applied almost without changes to other shops and datasets, and conclusions can be drawn from these results.

Tools used

- Chrome and Developer tools in the Browser
- For scraping was used Jupyter Notebook. Main packages use:
    - Pandas
    - BeautifulSoup
    - Selenium
- Web scraper browser extension tool
- VM with Ubuntu
- MariaDB

Lesson learned / self-reflection

We learned, or rather confirmed the assumption, that it is most expensive to buy Apple products in Germany and cheapest in the USA. Found the category with the largest difference in prices. We found unexpected relationship between reviews and price differences, of course this data

needs to be double-checked. But it did not coincide with our hypothesis, and that in itself is interesting.

The next step in the development of the project could be scraping additional sites with more reviews and possibly in new countries. This would allow us to generalise the conclusions.

We have learnt how the basic tools for scrpaing and data processing work, faced many challenges and solved many problems.

We have also seen the importance of starting work as soon as possible. And that even the seemingly easiest tasks can take an enormous amount of time, and one should be prepared for this. Additional study of supporting tools such as video editing applications and file editors can be very useful.

Overall, we found the project useful and instructive. Thanks for it!