Individual Contribution			
CWID	Name	Contribution	Percent Contribution
A20563430	Yaxuan Zhang	Bar chart completion and report writing	33.3%
A20563468	Lexin Li	Complete the scatter chart and line chart related content	33.3%
A20563435	Siyu Lu	Stack area map related content and description data set	33.3%

Catalogue

1 Describe the dataset

This dataset contains detailed specifications and official launch prices of various mobile phone models from different companies. It provides insights into smartphone hardware, pricing trends, and brand competitiveness across multiple countries. The dataset includes key features such as RAM, camera specifications, battery capacity, processor details, and screen size.

One important aspect of this dataset is the pricing information. The recorded prices represent the official launch prices of the mobile phones at the time they were first introduced in the market. Prices vary based on the country and the launch period, meaning older models reflect their original launch prices, while newer models include their most recent launch prices. This makes the dataset valuable for studying price trends over time and comparing smartphone affordability across different regions.

Features:

Company Name: The brand or manufacturer of the mobile phone.

Model Name: The specific model of the smartphone.

Mobile Weight: The weight of the mobile phone (in grams).

RAM: The amount of Random Access Memory (RAM) in the device (in GB).

Front Camera: The resolution of the front (selfie) camera (in MP).

Back Camera: The resolution of the primary rear camera (in MP).

Processor: The chipset or processor used in the device.

Battery Capacity: The battery size of the smartphone (in mAh).

Screen Size: The display size of the smartphone (in inches).

Launched Price: (Pakistan, India, China, USA, Dubai): The official launch price of the mobile in the respective country at the time of its release. Prices vary based on the year the mobile was launched.

Launched Year: The year the mobile phone was officially launched.

2 Line chart

2.1 The library used

Matplotlib library： Create a line chart to achieve visualization.

Pandas library: Read dataset data.

matplotlib.pyplot library: Create subgraphs to present multiple graphics together.

```
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib as mpl
```

2.2 Methods for creating visualizations

Firstly, use pandas' read_csv function to read the CSV file from the specified path and store the read data in the DataFrame object df.

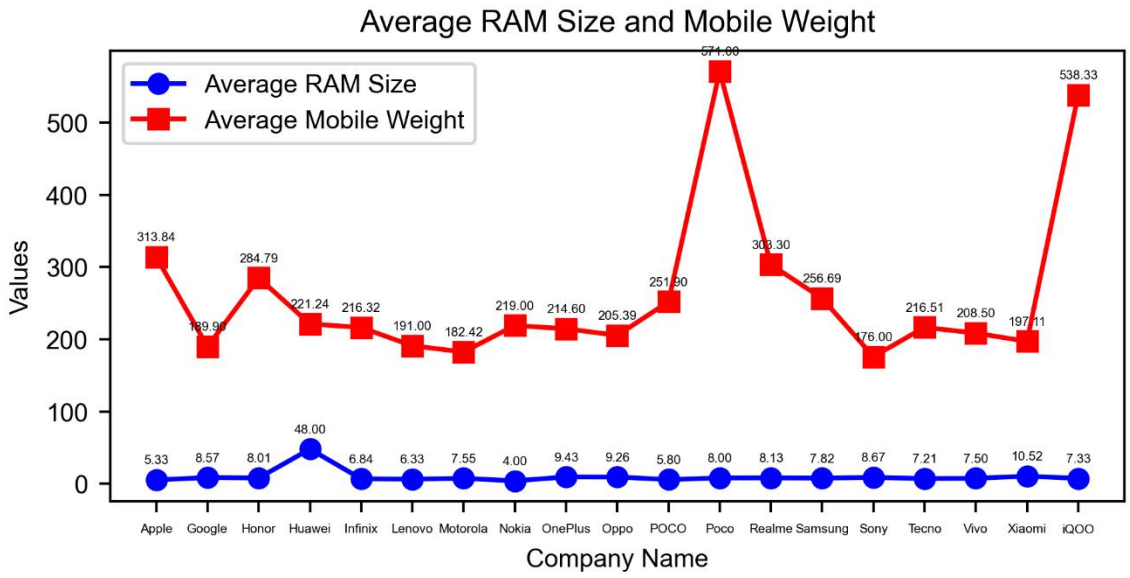
Next, perform data preprocessing. Define the extract_number function, which is used to extract the numerical part from a string and convert it to a floating-point number; Apply the extract_number function to the RAM column and Mobile Weight column using the apply method, and process the data in these two columns to ensure that it is of numerical type (to avoid interference from units)

Next, perform data grouping and computation. Use the groupby method to group companies along the horizontal axis (i.e., company name), calculate the average RAM size and Mobile Weight size for each company, and store the results in company ram and company weight.

Finally, create a line chart. Create a new graphics window using plt.figure and set the size of the graphics; use the plt.plot function to draw line charts of the average RAM size and average phone weight, and set different marker styles, colors, and labels; use the plt.annotate function to add data labels to each data point, displaying values with two decimal places; use the plt.show function to display the drawn graph.

2.3 Result and Analysis

The data results are presented as follows:



Average RAM size: The average RAM size of mobile phones from various companies is generally at a low level and the numerical difference is not significant, with most companies below 40GB.

Average mobile phone weight: There is a significant difference in the average weight of mobile phones among different companies. POCO's average mobile phone weight is the highest, reaching 571.00 grams; Google is relatively lightweight, weighing 189.90 grams. The average weight of most companies fluctuates around 200 grams.

Analysis:

The relationship between RAM and companies: The average RAM of most companies' mobile phones is relatively low, and the differences between different companies are not significant, indicating that within the scope of data coverage, there is not a significant gap in mobile phone RAM configuration among companies. But Huawei is an exception, with an average RAM significantly higher than other companies, reaching about 48.00GB, reflecting that the company may have a unique strategy in mobile phone memory configuration, more inclined to provide users with products with relatively large memory.

The relationship between weight and company: The average weight distribution range of mobile phones in various companies is relatively wide, indicating that there are significant differences in mobile phone weight design among different companies. However, despite the wide distribution, the average mobile phone weight of most companies is concentrated around 200 grams, indicating that this weight range may be a common and widely accepted mobile phone weight range in the market. Companies will refer to this range when designing.

The relationship between RAM and weight: It is difficult to see a clear correlation between the average RAM size and average weight of a mobile phone from the graph, and there is no significant consistency in the trend of the two changes.

3 Scatter plot

3.1 The library used

Pandas library: Read dataset data.

matplotlib.pyplot library: Create subgraphs to present multiple graphics together.

3.2 Methods for creating visualizations

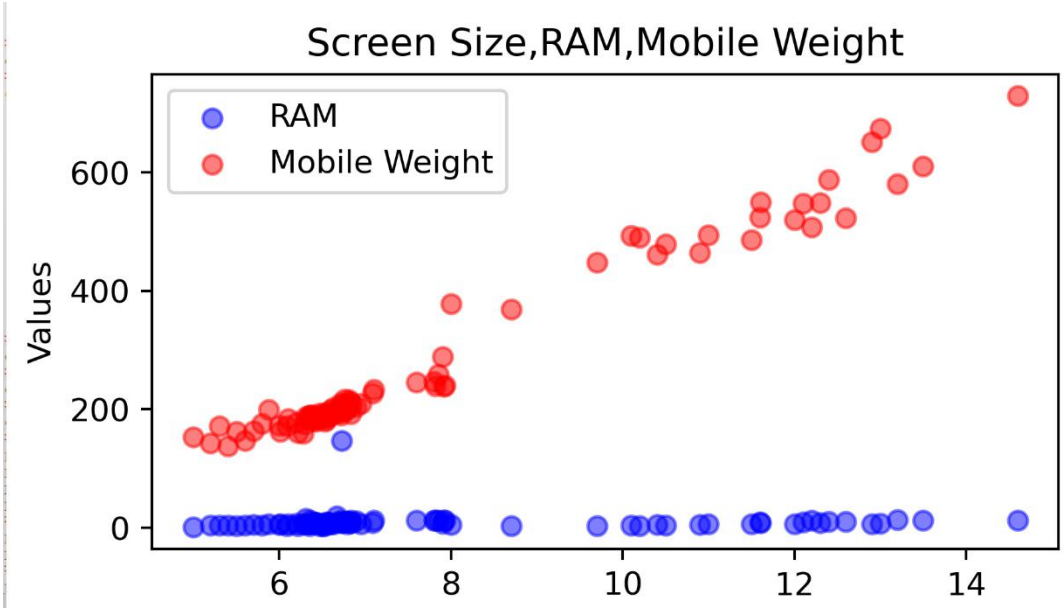
Firstly, use pd.read_csv to read CSV files and store the read data in df.

Secondly, define data processing functions and perform data preprocessing. Apply the extract_number function to each element in the RAM column and Mobile Weight column of the DataFrame to convert the strings into numbers.

Next, create a canvas and draw a scatter plot. Create a new graphic window using the plt.figure function; Using plt.scatter to draw a scatter plot (df ['Model Name '] is the x-axis data of the scatter plot, representing the phone model; Df ['RAM '] and df ['Mobile Weight'] are the y-axis data of the scatter plot, representing the RAM size and weight of the phone; And add titles, labels, and colors).

Finally, use plt.legend to display legends for distinguishing different scatter plots; Use plt.show to display the drawn graphics.

3.3 Result and Analysis



The relationship between screen size and RAM:

The overall level is low and the fluctuation is small, so even if there is a change in screen size, the change in RAM value is not significant. However, there are some points where the screen size is relatively large but the RAM is relatively small, indicating that there is no absolute correspondence between screen size and RAM, which may be influenced by various factors such as phone model positioning and cost control.

The relationship between screen size and phone weight:

The red dots show an overall increase in phone weight as the screen size increases. Screen size is an important factor affecting the weight of a mobile phone. Generally, the larger the screen, the more materials are required, and the overall weight of the phone also increases. But the scatter distribution is not completely concentrated on a straight line, indicating that besides screen size, there are other factors that affect the weight of the phone, such as battery capacity, body material, internal component configuration, etc.

The relationship between RAM and phone weight:

It is difficult to directly see a clear correlation between RAM and phone weight from the graph. This indicates that there is not a strong correlation between RAM size and phone weight in this dataset, and both may be determined by different design and manufacturing factors.

4 Bar chart

4.1 The library used

matplotlib.pyplot: Used to plot charts

pandas: Used to read, process, and analyze data.

re: Used to handle string matching and replacement operations.

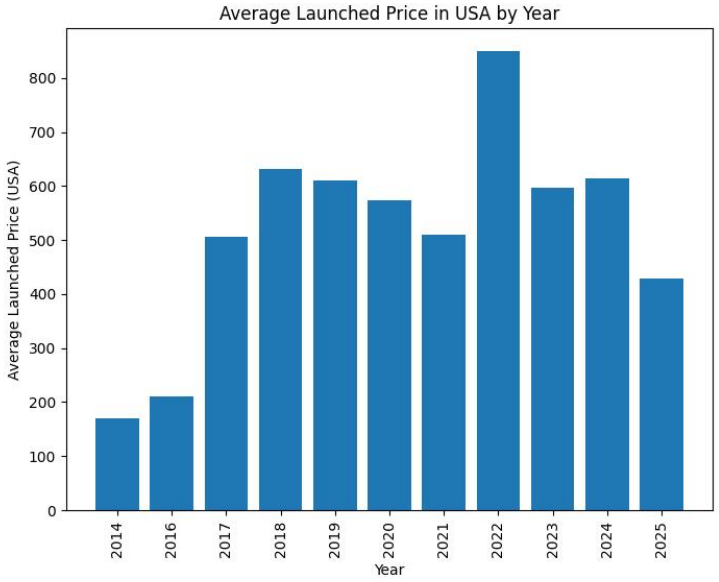
4.2 Methods for creating visualizations

The file is first read using pandas' read_csv function, then the non-numeric and non-decimal characters in the price string are removed and converted to floating-point numbers. Finally, the average is calculated.

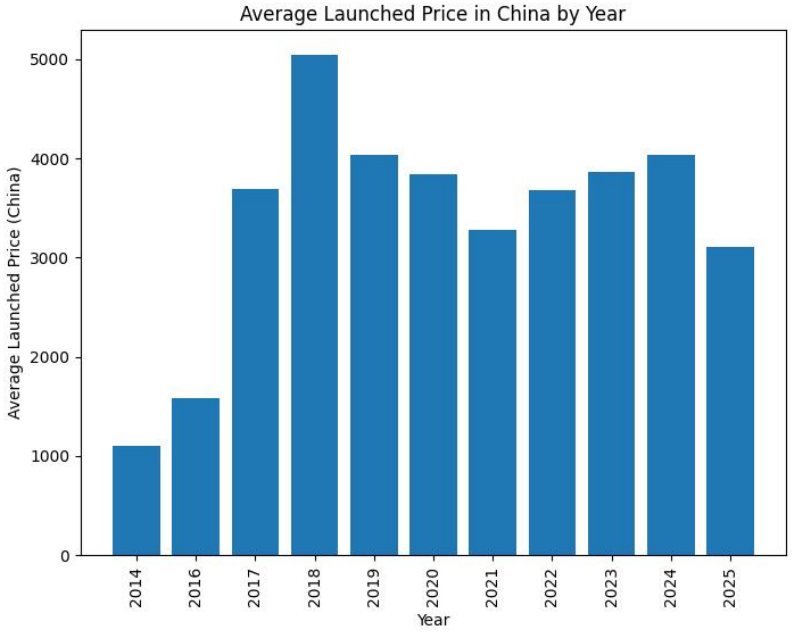
Creation of visualizations:

First create a new graph object with plt.figure and set the size of the graph, then use the plt.bar function to create a histogram with the year as the horizontal coordinate and the price as the vertical coordinate, then use the plt.xticks function to set the scale label on the X-axis, and then add the title. Finally, use the plt.show function to display the created visualizations.

4.3 Result and Analsis



As can be seen from the graph, the average release price of mobile phones was relatively low in 2014, followed by a significant increase in 2018. The average release price in 2022 reached the highest value in the chart, exceeding 800. The average release price for 2025 is one of the lower levels in the chart, around 400 +. Overall, there is a certain fluctuation in the average release price of mobile phones between various years, but the price has a downward trend after 2022.



As can be seen from the data in the figure, the average release price of mobile phones in 2014 was relatively low, about 1000. In 2018, the price reached the highest value in the chart, over 5000. Prices fluctuated in subsequent years, and in 2025, the average release price was around just over 3,000. As a whole, it shows a trend of first rising and then fluctuating.

5 Stack area diagram

5.1 The library used

pandas (for data processing)

numpy (for numerical computation)

matplotlib.pyplot (for plotting)

seaborn (for beautifying charts)

5.2 Methods for creating visualizations

Read data

The program first uses pandas to read the smartphone price data stored in a CSV file and select the columns related to the price (price information for different regions).

Data clearing

Since price data may contain currency symbols or other non-numeric characters, the program defines the clean_price function:

Keep only digits and decimal points and remove unnecessary characters.

Is set to the missing value (NaN) if the conversion fails.

Apply the cleanup function to price columns for all regions to ensure that the data format is consistent.

Calculate the average price by region each year

The program grouped smartphones into categories "Launched Year" and calculated the average price of smartphones in each region each year to get a picture

of price trends over time.

Normalization processing

To make the data more comparable, the program normalizes all price data using the earliest year as the base year:

Normalization formula: Price per region ÷ price in base year.

In this way, we can observe the relative price changes in different regions, regardless of the specific price numbers.

Draw a stack area diagram

The program uses matplotlib and seaborn to generate visual charts:

kind="area" generates the area chart and sets the stacked=True to stack.

Color transparency (alpha=0.7) and color scheme (colormap="tab10") improve readability.

Add titles, axis labels, legends, and grids to make your charts clearer.

Result analysis

The resulting chart shows relative price trends over time in different regions and can be used to compare changes in affordability across regions.

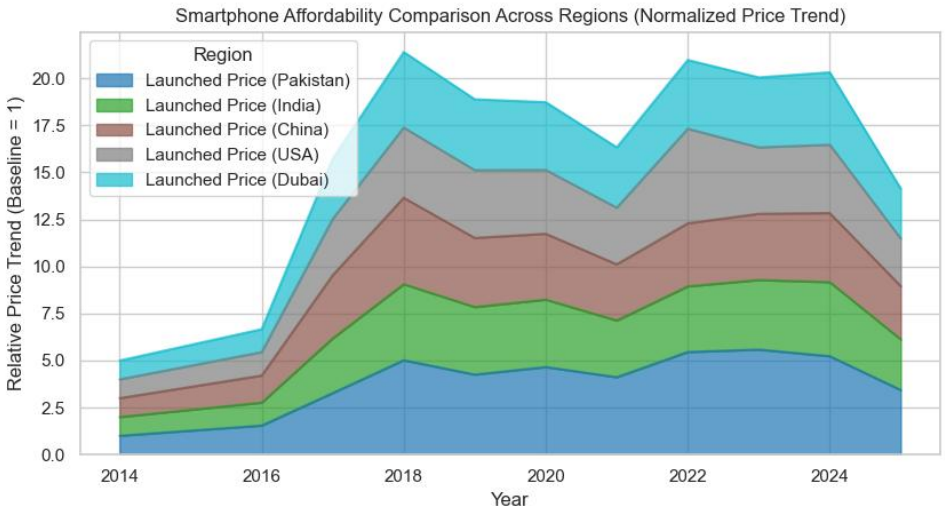
For example:

If the size of an area increases, the relative price of the area becomes higher and the affordability decreases.

If the area of a certain area is reduced, it means that the relative price of the area is reduced and the affordability is increased.

This code can help analyze price trends in the global smartphone market and is of great value for market research and economic analysis.

5.3 Result and Analysis



Summary of results

Based on the selling price data of smartphones in different regions, the analysis uses Stacked Area Chart to show the relative price trends in different regions. All prices have been normalized to the base year (earliest year) to visually compare price changes over time in different regions.

Visual analysis

According to the stack area diagram drawn, we can observe the following trends:

Global trends:

If the total area increases over time, it indicates that the overall price of smart phones is rising, and the purchase burden of consumers may increase.

If the total area is stable or declining, the overall price is relatively stable or declining, and the phone is more affordable.

Differences in price changes between regions:

Faster relative price growth in some regions means that smartphone prices in these regions have increased more, which may be related to the exchange rate, import taxes, market demand and other factors.

The slower price growth in some regions indicates that price adjustments are

relatively stable and may be influenced by market competition, government subsidies or local manufacturing industries.

Price volatility:

If a region has a significant increase in the percentage of the area on the chart, it means that the price of smartphones in that region is rising faster than in other regions.

If the proportion of area in a region decreases, the price of smartphones in that region will increase less, and may even decline.

Key conclusion

Economic impact: Areas with faster price increases are likely to face higher living costs, while areas with stable or declining prices are relatively affordable to consumers.

Market strategy: Manufacturers can optimize pricing strategies according to price trends in different regions, such as introducing higher-end products in regions with slower price growth and providing more cost-effective models in regions with faster price growth.

Policy implications: Governments or regulators can take these trends into account to develop policies that support local manufacturing or reduce import taxes to improve the affordability of electronics.

If more detailed analysis is needed, such as for inflation adjustments or per capita income comparisons, the data processing can be further optimized to gain deeper insights.