Part 2 Data Optimization

Prepared by : TAN JUN YUAN(A22EC0107)

Step 1 : Install and Import Libraries

```
In [ ]: !pip install polars
        import polars as pl
        import pandas as pd
        import re
        import time
        import psutil
        import tracemalloc
```

```
Requirement already satisfied: polars in /usr/local/lib/python3.11/dist-packages (1.21.0)
```

Step 2 : Upload Excel Files

```
In [ ]: from google.colab import files
        uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving polars_cleaned_dataset.csv to polars_cleaned_dataset.csv
```

Step 3 : Load and Display Dataset, Checking on Total Numbers of Rows and Columns

```
In [ ]: %%time

        tracemalloc.start()
        start_time = time.perf_counter()
        total_rows = 0

        filename = list(uploaded.keys())[0]
        df_cleaned = pl.read_csv(filename)

        total_rows = df_cleaned.shape[0]
        display(df_cleaned.head(10))
        print(f"Total rows: {df_cleaned.shape[0]}")
        print(f"Total columns: {df_cleaned.shape[1]}\n\n\n")

        current, peak = tracemalloc.get_traced_memory()
        end_time = time.perf_counter()
        tracemalloc.stop()

        execution_time = end_time - start_time
        throughput = total_rows / execution_time

        print("===================== Performance =====================\n")
        print(f"Total rows processed: {total_rows}")
        print(f"Code Execution time: {execution_time:.4f} seconds")
        print(f"Throughput: {throughput:.2f} rows per second")
        print(f"Current memory usage: {current / 10**6:.4f} MB")
        print(f"Peak memory usage: {peak / 10**6:.4f} MB")

        cpu_usage = psutil.cpu_percent(interval=1)
        print(f"CPU usage: {cpu_usage}%")

        print("=======================================================")

        print("\nTotal time for this cell(Including time to display the performance):")
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"ELECTRIC COOKER HOUSEHOLD DORM...	68.85	"OVERSEAS"	0	0	"HOME APPLIANCES"
"JAYNEBIO 6PCS/BOX ST TIP CAPYB...	5.97	"CHINA"	9	4	"STATIONERY"
"PILOT HI-TEC POINT BXRT-V5 RET...	31.53	"N/A"	18000	0	"STATIONERY"
"SPRING SUMMER NEW STYLE KOREAN...	44.49	"OVERSEAS"	6	0	"WOMEN'S FASHION"
"【2/5 BUNDLES】PENTAVITE COLLAG...	70.0	"SELANGOR"	0	1	"BEAUTY & SKINCARE"
"FRENCH SQUARE COLLAR SWEET GEN...	55.06	"N/A"	157	2	"WOMEN'S FASHION"
"AEROFUME 100% NATURAL SOY WAX ...	39.9	"SELANGOR"	0	0	"HOME & LIVING"
"PILOT G2 GEL PEN REFILL 0.7MM"	5.35	"SELANGOR"	11	1	"STATIONERY"
"AUSTRALIAN AOJIABAO BLACKMORES...	90.59	"CHINA"	0	0	"HEALTH & WELLNESS"
"[VASELINE] LIPSTICK LIP BALM T...	37.4	"SOUTH KOREA"	35	15	"BEAUTY & SKINCARE"

```
Total rows: 113596
Total columns: 6



==================== Performance ====================

Total rows processed: 113596
Code Execution time: 0.8896 seconds
Throughput: 127694.12 rows per second
Current memory usage: 1.1556 MB
Peak memory usage: 4.2618 MB
CPU usage: 36.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 550 ms, sys: 64.3 ms, total: 614 ms
Wall time: 1.89 s
```

Step 4 : Dividing Products into 4 Categories Based on Price using Polars

In [ ]:

```python
%%time

tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_cleaned.shape[0]

df_price = df_cleaned.filter(pl.col("Price") > 0)

q1 = df_price.select(pl.col("Price").quantile(0.25)).item()
q3 = df_price.select(pl.col("Price").quantile(0.75)).item()
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

df_cleared = df_price.filter((pl.col("Price") >= lower_bound) & (pl.col("Price") <= upper_bound))

min_price = df_cleared.select(pl.col("Price").min()).item()
max_price = df_cleared.select(pl.col("Price").max()).item()
print(f"Min Price: {min_price}")
print(f"Max Price: {max_price}")

price_range = (max_price - min_price) / 4
bound1 = min_price + price_range
bound2 = min_price + 2 * price_range
bound3 = min_price + 3 * price_range

group1 = df_price.filter(pl.col("Price") <= bound1).sort("Price")
group2 = df_price.filter((pl.col("Price") > bound1) & (pl.col("Price") <= bound2)).sort("Price")
group3 = df_price.filter((pl.col("Price") > bound2) & (pl.col("Price") <= bound3)).sort("Price")
group4 = df_price.filter(pl.col("Price") > bound3).sort("Price")

print(f"\nGroup 1 (Budget Friendly Price): {group1.shape[0]} products")
print(f"Group 2 (Affordable Price): {group2.shape[0]} products")
print(f"Group 3 (Mid-Range Price): {group3.shape[0]} products")
print(f"Group 4 (Premium Price): {group4.shape[0]} products")

def print_category_counts(df_group):
    print(f"📊 Category count:")
    for row in df_group.group_by("Category").len().sort("len", descending=True).iter_rows():
        category, count = row
        print(f"- {category}: {count} products")

print("\nGroup 1 (Budget Friendly Price):")
display(group1.head(10))
print(f"Total rows: {group1.shape[0]}")
print(f"Total columns: {group1.shape[1]}\n")
print_category_counts(group1)

print("\nGroup 2 (Affordable Price):")
display(group2.head(10))
print(f"Total rows: {group2.shape[0]}")
print(f"Total columns: {group2.shape[1]}\n")
print_category_counts(group2)

print("\nGroup 3 (Mid-Range Price):")
display(group3.head(10))
print(f"Total rows: {group3.shape[0]}")
print(f"Total columns: {group3.shape[1]}\n")
print_category_counts(group3)

print("\nGroup 4 (Premium Price):")
display(group4.head(10))
print(f"Total rows: {group4.shape[0]}")
print(f"Total columns: {group4.shape[1]}\n")
print_category_counts(group4)
print("\n\n\n")
```

```python
current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=======================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

```
Min Price: 0.05
Max Price: 172.82

Group 1 (Budget Friendly Price): 65992 products
Group 2 (Affordable Price): 22626 products
Group 3 (Mid-Range Price): 10327 products
Group 4 (Premium Price): 14651 products

Group 1 (Budget Friendly Price):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"【MALAYSIA 3PIN PLUG】3L ELECTRI…	0.05	"SELANGOR"	0	0	"HOME APPLIANCES"
"LIVE TRACKING #NOT FOR SALES"	0.05	"KELANTAN"	0	1	"HOME & LIVING"
"SPOT BESEEN PLUS EYE CARE + BR…	0.1	"JOHOR"	0	0	"HEALTH & WELLNESS"
"1PCS 0.5MM BALL GEL INK PEN MA…	0.1	"SELANGOR"	33500	1354	"STATIONERY"
"HEALTH TREE REISSUE PRODUCT LI…	0.1	"CHINA"	10	3	"HEALTH & WELLNESS"
"BEFREE BESEEN PLUS VITAMIN EYE…	0.1	"JOHOR"	0	0	"HEALTH & WELLNESS"
"READY STOCK MICKEY MOUSE MASCO…	0.11	"WP KUALA LUMPUR"	49	1	"HOME & LIVING"
"✨ HARI RAYA HIASAN GANTUNG KRA…	0.12	"WP KUALA LUMPUR"	24	1	"HOME & LIVING"
"PENUTUP BOTTLE CAPS ARTEMIA,AR…	0.13	"PENANG"	7	0	"HEALTH & WELLNESS"
"《**MILD EXFOLIATION**》 SOAP FOAM…	0.14	"PERAK"	922	20	"HEALTH & WELLNESS"

```
Total rows: 65992
Total columns: 6

📊 Category count:
- STATIONERY: 20460 products
- BEAUTY & SKINCARE: 17957 products
- HEALTH & WELLNESS: 10315 products
- HOME & LIVING: 8293 products
- WOMEN'S FASHION: 4653 products
- HOME APPLIANCES: 3837 products
- MOTHER & BABY: 477 products

Group 2 (Affordable Price):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"CERAVE PM FACIAL LOTION 89ML F…	43.25	"JOHOR"	0	0	"BEAUTY & SKINCARE"
"NATURAL HAIRLINE BANGS WIG PIE…	43.25	"N/A"	0	0	"WOMEN'S FASHION"
"CERAVE ULTRA-LIGHT MOISTURIZIN…	43.25	"JOHOR"	0	0	"BEAUTY & SKINCARE"
"【READY STOCK】FREEPLUS MILD SOA…	43.25	"SELANGOR"	0	0	"BEAUTY & SKINCARE"
"BERRYFULL ORIGINAL INAYAH BEAU…	43.26	"KELANTAN"	0	0	"HEALTH & WELLNESS"
"DONG KOOK 东国四代水光精华 SKIN （1 BO…	43.26	"SELANGOR"	7	1	"HEALTH & WELLNESS"
"WHITENING SUPPLEMENT L-GLUTATH…	43.26	"SELANGOR"	0	1	"HEALTH & WELLNESS"
"(EXPIRY 112026) HIMALAYA TRIPH…	43.26	"SELANGOR"	0	0	"HEALTH & WELLNESS"
"KUALA LUMPUR SPOT 最夯瘦身 FLASHY …	43.26	"WP KUALA LUMPUR"	344	10	"HEALTH & WELLNESS"
"HIKARI PREMIUM JAPAN L-GLUTATH…	43.26	"SELANGOR"	0	0	"HEALTH & WELLNESS"

```
Total rows: 22626
Total columns: 6

📊 Category count:
- HEALTH & WELLNESS: 6380 products
- BEAUTY & SKINCARE: 5373 products
- WOMEN'S FASHION: 3623 products
- HOME APPLIANCES: 3053 products
- HOME & LIVING: 2309 products
- STATIONERY: 1170 products
- MOTHER & BABY: 718 products

Group 3 (Mid-Range Price):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"URIAGE BARIEDERM CLEANSING CIC…	86.44	"NEGERI SEMBILAN"	0	1	"BEAUTY & SKINCARE"
"EXP:10/2026 NUTRITION CARE GUT…	86.45	"SELANGOR"	383	109	"HEALTH & WELLNESS"
"JERGENS COCONUT MOISTURIZING B…	86.45	"PENANG"	0	1	"BEAUTY & SKINCARE"
"[BEST DFFER] ONLINE EXCLUSIVE…	86.46	"WP KUALA LUMPUR"	0	0	"HOME & LIVING"
"[EROM] (FREE SHAKER INCLUDED) …	86.46	"OVERSEAS"	51	18	"HEALTH & WELLNESS"
"[BEST DFFER] BATA HALIL WOMEN…	86.46	"WP KUALA LUMPUR"	0	1	"HOME & LIVING"
"⭐ ⭐READY STOCK⭐ ⭐ ☞ LUNA JELLY…	86.48	"WP KUALA LUMPUR"	0	0	"BEAUTY & SKINCARE"
"SWISSE BEAUTY BRIGHT SKIN BOOS…	86.49	"SELANGOR"	85	30	"BEAUTY & SKINCARE"
"SWISSE BEAUTY BRIGHT SKIN BOOS…	86.49	"SELANGOR"	85	30	"HEALTH & WELLNESS"
"EXCELAB 12 PROBIOTICS VEGE CAP…	86.5	"SELANGOR"	23	4	"HEALTH & WELLNESS"

```
Total rows: 10327
Total columns: 6

📊 Category count:
- HEALTH & WELLNESS: 3866 products
- BEAUTY & SKINCARE: 2008 products
- HOME APPLIANCES: 1938 products
- HOME & LIVING: 977 products
- WOMEN'S FASHION: 774 products
- STATIONERY: 451 products
- MOTHER & BABY: 313 products

Group 4 (Premium Price):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"1PCS BLENDER CUP COVER SUITABL...	129.66	"CHINA"	0	0	"HOME APPLIANCES"
"JKLIV JAPANESE-STYLE WABI-SABI...	129.66	"CHINA"	0	0	"HOME & LIVING"
"CREATIVE GIFT FOR NURSES DAY P...	129.7	"N/A"	207	5	"HOME APPLIANCES"
"YOUNG WOMEN'S PURPLE CHEONGSAM...	129.7	"N/A"	600	0	"WOMEN'S FASHION"
"PENTEL VICUNA EX3 - 2 COLOR (B...	129.7	"SELANGOR"	5	2	"STATIONERY"
"CREATIVE GIFT FOR NURSES DAY P...	129.7	"N/A"	195	5	"HEALTH & WELLNESS"
"THICKENED XPE BABY CRAWLING MA...	129.7	"CHINA"	0	0	"HOME & LIVING"
"【LOCAL DELIVERY】XIAO.MI 1000ML...	129.71	"KEDAH"	52	23	"HEALTH & WELLNESS"
"【HOT SALES】[VTCOSMETICS] FOAM ...	129.71	"SELANGOR"	0	0	"BEAUTY & SKINCARE"
"【HOT SALES】[VT COSMETICS] SUPE...	129.71	"SELANGOR"	0	0	"BEAUTY & SKINCARE"

```
Total rows: 14651
Total columns: 6

📊 Category count:
- HOME APPLIANCES: 4700 products
- HEALTH & WELLNESS: 3581 products
- BEAUTY & SKINCARE: 2388 products
- HOME & LIVING: 1642 products
- MOTHER & BABY: 923 products
- STATIONERY: 794 products
- WOMEN'S FASHION: 623 products



==================== Performance ====================

Total rows processed: 113596
Code Execution time: 0.5189 seconds
Throughput: 218918.57 rows per second
Current memory usage: 0.0536 MB
Peak memory usage: 0.1225 MB
CPU usage: 90.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 181 ms, sys: 88.5 ms, total: 269 ms
Wall time: 1.52 s
```

Step 5 : Filtering Based on 'Total Reviews' to Determine Popularity of Products

In [ ]:

```python
%%time

tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_cleaned.shape[0]

Q1 = df_cleaned.select(pl.col("Total Reviews").quantile(0.25)).item()
Q3 = df_cleaned.select(pl.col("Total Reviews").quantile(0.75)).item()
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df_filtered = df_cleaned.filter((pl.col("Total Reviews") >= lower_bound) & (pl.col("Total Reviews") <= upper_bound))

min_ratings = df_filtered.select(pl.col("Total Reviews").min()).item()
max_ratings = df_filtered.select(pl.col("Total Reviews").max()).item()
print(f"Minimum Number of Total Reviews: {min_ratings}")
print(f"Maximum Number of Total Reviews: {max_ratings}")

rating_range = round((max_ratings - min_ratings) / 4)
bound1 = min_ratings + rating_range
bound2 = min_ratings + 2 * rating_range
bound3 = min_ratings + 3 * rating_range

group1 = df_filtered.filter(pl.col("Total Reviews") <= bound1).sort("Total Reviews", descending=True)
group2 = df_filtered.filter((pl.col("Total Reviews") > bound1) & (pl.col("Total Reviews") <= bound2)).sort("Total Reviews", descending=True)
group3 = df_filtered.filter((pl.col("Total Reviews") > bound2) & (pl.col("Total Reviews") <= bound3)).sort("Total Reviews", descending=True)
group4 = df_filtered.filter(pl.col("Total Reviews") > bound3).sort("Total Reviews", descending=True)

def print_category_counts(df_group):
    print(f"📊 Category count:")
    for row in df_group.group_by("Category").len().sort("len", descending=True).iter_rows():
        category, count = row
        print(f"- {category}: {count} products")

print(f"\nGroup 1 (Least popular): {group1.shape[0]} products")
print(f"Group 2 (Below Average Popularity): {group2.shape[0]} products")
print(f"Group 3 (Above Average Popularity): {group3.shape[0]} products")
print(f"Group 4 (Most popular): {group4.shape[0]} products")

print("\nGroup 1 (Least popular):")
display(group1.head(10))
print(f"Total rows: {group1.shape[0]}")
print(f"Total columns: {group1.shape[1]}\n")
print_category_counts(group1)

print("\nGroup 2 (Below Average Popularity):")
display(group2.head(10))
print(f"Total rows: {group2.shape[0]}")
print(f"Total columns: {group2.shape[1]}\n")
print_category_counts(group2)

print("\nGroup 3 (Above Average Popularity):")
display(group3.head(10))
print(f"Total rows: {group3.shape[0]}")
print(f"Total columns: {group3.shape[1]}\n")
print_category_counts(group3)

print("\nGroup 4 (Most popular):")
display(group4.head(10))
print(f"Total rows: {group4.shape[0]}")
print(f"Total columns: {group4.shape[1]}\n")
print_category_counts(group4)
print("\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("===================== Performance =====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")

cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=======================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

```
Minimum Number of Total Reviews: 0
Maximum Number of Total Reviews: 27

Group 1 (Least popular): 80729 products
Group 2 (Below Average Popularity): 7845 products
Group 3 (Above Average Popularity): 4653 products
Group 4 (Most popular): 2783 products

Group 1 (Least popular):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"UNICORN STATIONERY 0.5 MM RUB-...	2.0	"SELANGOR"	68	7	"STATIONERY"
"HOME LIVING ROOM BEDROOM FLOOR...	55.0	"NEGERI SEMBILAN"	28	7	"HOME & LIVING"
"ELBA 5.0L ELECTRIC KETTLE STAI...	119.0	"WP KUALA LUMPUR"	19	7	"HOME APPLIANCES"
"BURTS BEES 100% NATURAL MOISTU...	35.9	"PENANG"	44	7	"BEAUTY & SKINCARE"
"SOLID ADHESIVE NAIL GLUE SUPER...	2.86	"MELAKA"	31	7	"HEALTH & WELLNESS"
" 🔥 SHIP 24H 🔥 1PCS CURSIVE WRITING...	10.74	"CHINA"	16	7	"STATIONERY"
"QUICK EXTENDED GLUE MANICURE A...	4.56	"N/A"	81	7	"STATIONERY"
"** （笔芯0.5MM）抄经金笔/ GOLD PEN / 5支...	3.5	"SELANGOR"	37	7	"STATIONERY"
"GLUMONY KAPSUL PEMUTIH BADAN W...	24.03	"WP KUALA LUMPUR"	20	7	"HEALTH & WELLNESS"
"FABER CASTLE 48 TRI COLOUR PEN...	28.8	"JOHOR"	27	7	"STATIONERY"

```
Total rows: 80729
Total columns: 6

📊 Category count:
- BEAUTY & SKINCARE: 18080 products
- HEALTH & WELLNESS: 16519 products
- STATIONERY: 15505 products
- HOME & LIVING: 10196 products
- HOME APPLIANCES: 9757 products
- WOMEN'S FASHION: 8441 products
- MOTHER & BABY: 2231 products

Group 2 (Below Average Popularity):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"PARKER IM BRUSHED METAL CT STA…	83.0	"SELANGOR"	46	14	"STATIONERY"
"LED FULL-BODY MIRROR MODERN SM…	269.0	"PENANG"	38	14	"HOME & LIVING"
"[READY STOCK EXP:11/2026] COMV…	207.7	"PAHANG"	44	14	"HEALTH & WELLNESS"
"NEW CHINESE STYLE HIGH WAIST S…	42.97	"N/A"	98	14	"WOMEN'S FASHION"
"ZY·HT KOREAN VERSION FASHION B…	168.0	"CHINA"	46	14	"WOMEN'S FASHION"
"现货✨晨光中性笔1支 M&G GEL INK PEN 1PC…	1.13	"PERAK"	144	14	"STATIONERY"
"EGO QV CREAM 100ML/500ML [SKI…	32.8	"PERAK"	42	14	"BEAUTY & SKINCARE"
"MALTOFER FOL 100MG CHEWABLE TA…	29.8	"SELANGOR"	36	14	"HEALTH & WELLNESS"
"SPINNING PEN TRAINING BALANCE …	2.2	"SELANGOR"	65	14	"STATIONERY"
"P 3.5 / 4 INCH CERAMIC ASHTRAY…	1.7	"NEGERI SEMBILAN"	128	14	"HOME & LIVING"

```
Total rows: 7845
Total columns: 6

📊 Category count:
- BEAUTY & SKINCARE: 2205 products
- HEALTH & WELLNESS: 1784 products
- STATIONERY: 1752 products
- HOME APPLIANCES: 864 products
- HOME & LIVING: 791 products
- WOMEN'S FASHION: 408 products
- MOTHER & BABY: 41 products

Group 3 (Above Average Popularity):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"INSTANT ELECTRIC STAINLESS STE…	197.0	"SELANGOR"	117	21	"HOME APPLIANCES"
"PILOT PERMANENT SCA MARKER 100…	3.24	"WP KUALA LUMPUR"	87	21	"STATIONERY"
"TRAVEL MAKEUP STORAGE ORGANIZE…	66.9	"SELANGOR"	66	21	"BEAUTY & SKINCARE"
"EYES CLEAR LEGASI PERMATA SAKI…	18.0	"SELANGOR"	49	21	"BEAUTY & SKINCARE"
"TOSHIBA BLENDER BL-60PHNMY MOT…	70.9	"SELANGOR"	49	21	"HOME APPLIANCES"
"CORNELL 1.8L CHOPPER BLENDER E…	89.0	"SELANGOR"	63	21	"HOME APPLIANCES"
"YLMEI MEAT GRINDER ICE SHAVER …	44.25	"KEDAH"	61	21	"HOME APPLIANCES"
"360° ROTATING KITCHEN STORAGE …	9.98	"PERAK"	61	21	"STATIONERY"
"【READY STOCK】KIEHL'S CREAMY EY…	96.0	"PENANG"	57	21	"BEAUTY & SKINCARE"
"SCANDINAVIAN MODERN MINIMALIST…	31.1	"N/A"	2500	21	"HOME & LIVING"

```
Total rows: 4653
Total columns: 6

📊 Category count:
- BEAUTY & SKINCARE: 1279 products
- STATIONERY: 1083 products
- HEALTH & WELLNESS: 1031 products
- HOME APPLIANCES: 537 products
- HOME & LIVING: 469 products
- WOMEN'S FASHION: 231 products
- MOTHER & BABY: 23 products

Group 4 (Most popular):
```

shape: (10, 6)

Product Name	Price	Location	Quantity Sold	Total Reviews	Category
str	f64	str	i64	i64	str
"ORGANIC PSYLLIUM HUSK SUPPLEME…	10.73	"SELANGOR"	82	27	"HEALTH & WELLNESS"
"HONG KONG STYLE VINTAGE BACKLE…	52.82	"N/A"	605	27	"WOMEN'S FASHION"
"SWEET STYLE KNITTED FABRIC SPO…	33.0	"N/A"	132	27	"WOMEN'S FASHION"
"1/2PCS MIRROR MARKER PEN DIY R…	5.17	"CHINA"	1300	27	"STATIONERY"
"MENTHOLATUM MEN ICY CHARCOAL F…	21.45	"SELANGOR"	68	27	"BEAUTY & SKINCARE"
"OASIS LIVELY GREEN AIR REFRESH…	8.19	"SELANGOR"	130	27	"HOME & LIVING"
"CHOCO INCH ORIGINAL HQ (COKELA…	170.0	"KEDAH"	37	27	"HEALTH & WELLNESS"
"HEAD AND SHOULDER SHAMPOO SMOO…	16.77	"SELANGOR"	283	27	"HEALTH & WELLNESS"
"LIVINGMALL 17CM RESIN GESTURE …	17.6	"N/A"	42	27	"HOME & LIVING"
"NECK MASSAGER ADJUSTABLE TRAVE…	29.98	"SELANGOR"	91	27	"HEALTH & WELLNESS"

```
Total rows: 2783
Total columns: 6

📊 Category count:
- BEAUTY & SKINCARE: 758 products
- HEALTH & WELLNESS: 636 products
- STATIONERY: 609 products
- HOME APPLIANCES: 329 products
- HOME & LIVING: 287 products
- WOMEN'S FASHION: 142 products
- MOTHER & BABY: 22 products




==================== Performance ====================

Total rows processed: 113596
Code Execution time: 0.2981 seconds
Throughput: 381122.98 rows per second
Current memory usage: 0.0196 MB
Peak memory usage: 0.0664 MB
CPU usage: 100.0%
=====================================================

Total time for this cell(Including time to display the performance):
CPU times: user 121 ms, sys: 77.4 ms, total: 198 ms
Wall time: 1.3 s
```

Step 6 : Ranking Location Based on Market Performance

In [ ]:

```
%%time

tracemalloc.start()
start_time = time.perf_counter()
total_rows = df_cleaned.shape[0]

df_location_sales = df_price.group_by("Location").agg(
    pl.col("Quantity Sold").sum().alias("Total Quantity Sold"),
    pl.col("Price").mean().alias("Average Price")
)

df_location_sales = df_location_sales.with_columns(
    (pl.col("Total Quantity Sold") * pl.col("Average Price")).alias("Market Performance")
)


df_location_sales = df_location_sales.sort("Market Performance", descending=True)
display(df_location_sales.head(10))
print("\n\n\n")

current, peak = tracemalloc.get_traced_memory()
end_time = time.perf_counter()
tracemalloc.stop()

execution_time = end_time - start_time
throughput = total_rows / execution_time

print("==================== Performance ====================\n")
print(f"Total rows processed: {total_rows}")
print(f"Code Execution time: {execution_time:.4f} seconds")
print(f"Throughput: {throughput:.2f} rows per second")
print(f"Current memory usage: {current / 10**6:.4f} MB")
print(f"Peak memory usage: {peak / 10**6:.4f} MB")
```

```
cpu_usage = psutil.cpu_percent(interval=1)
print(f"CPU usage: {cpu_usage}%")

print("=======================================================")

print("\nTotal time for this cell(Including time to display the performance):")
```

shape: (10, 4)

Location	Total Quantity Sold	Average Price	Market Performance
str	i64	f64	f64
"SELANGOR"	10562173	100.849751	1.0652e9
"N/A"	18524452	43.647313	8.0854e8
"CHINA"	6341944	97.622898	6.1912e8
"WP KUALA LUMPUR"	1227922	101.519881	1.2466e8
"JOHOR"	1499826	78.073532	1.1710e8
"OVERSEAS"	2083568	48.480572	1.0101e8
"PENANG"	962481	100.382696	9.6616e7
"PERAK"	1029478	80.170026	8.2533e7
"HONG KONG"	376652	138.300619	5.2091e7
"KEDAH"	595993	80.050847	4.7710e7

```
===================== Performance =====================

Total rows processed: 113596
Code Execution time: 0.1121 seconds
Throughput: 1013159.21 rows per second
Current memory usage: 0.0090 MB
Peak memory usage: 0.0222 MB
CPU usage: 100.0%
=======================================================

Total time for this cell(Including time to display the performance):
CPU times: user 42 ms, sys: 15.1 ms, total: 57.1 ms
Wall time: 1.11 s
```

End of Part 2 Data Optimization