Basic Level Questions

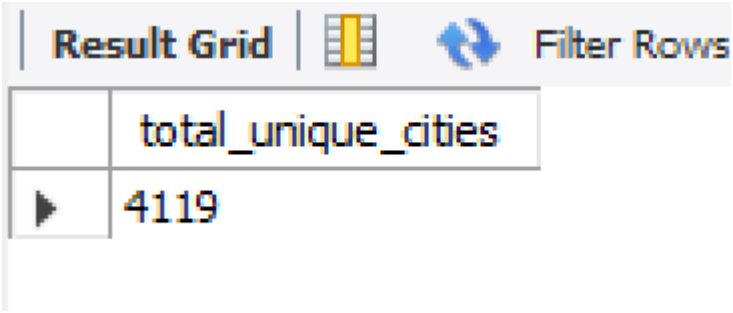
Q1. List all unique cities where customers are located.

Python:

```
unique_cities = customers['customer_city'].unique()
print(f"Total Unique Cities: {len(unique_cities)}")
```

```
Total Unique Cities: 4119
```

SQL:



	total_unique_cities
▶	4119

Q2. Count the number of orders placed in 2017.

Python:

```
orders_2017 = orders[orders['order_purchase_timestamp'].dt.year == 2017].shape[0]
print(f"Orders placed in 2017: {orders_2017}")
```

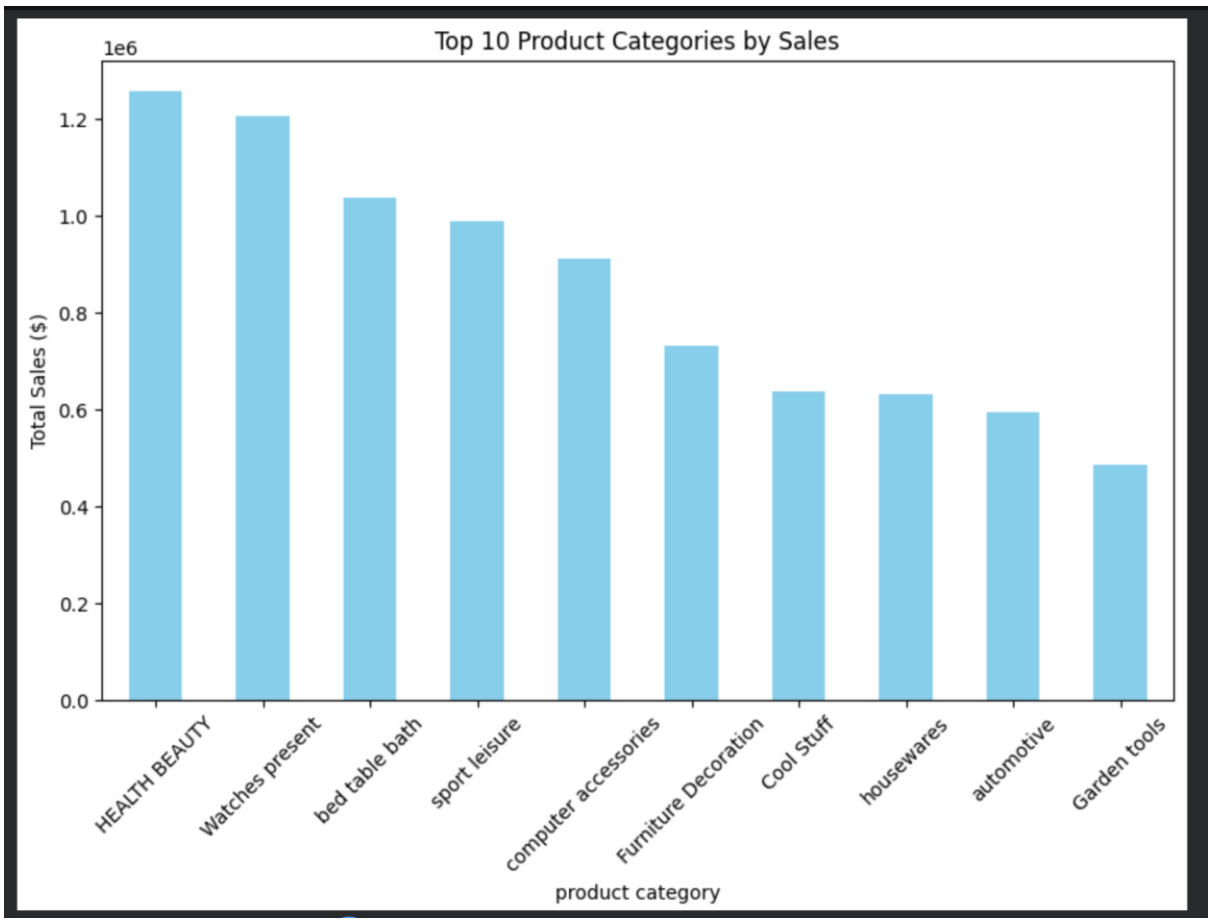
```
Orders placed in 2017: 45101
```

SQL:

	order_year	total_orders
▶	2017	45101
	2018	54011
	2016	329

Q3. Find the total sales per category.

Python:

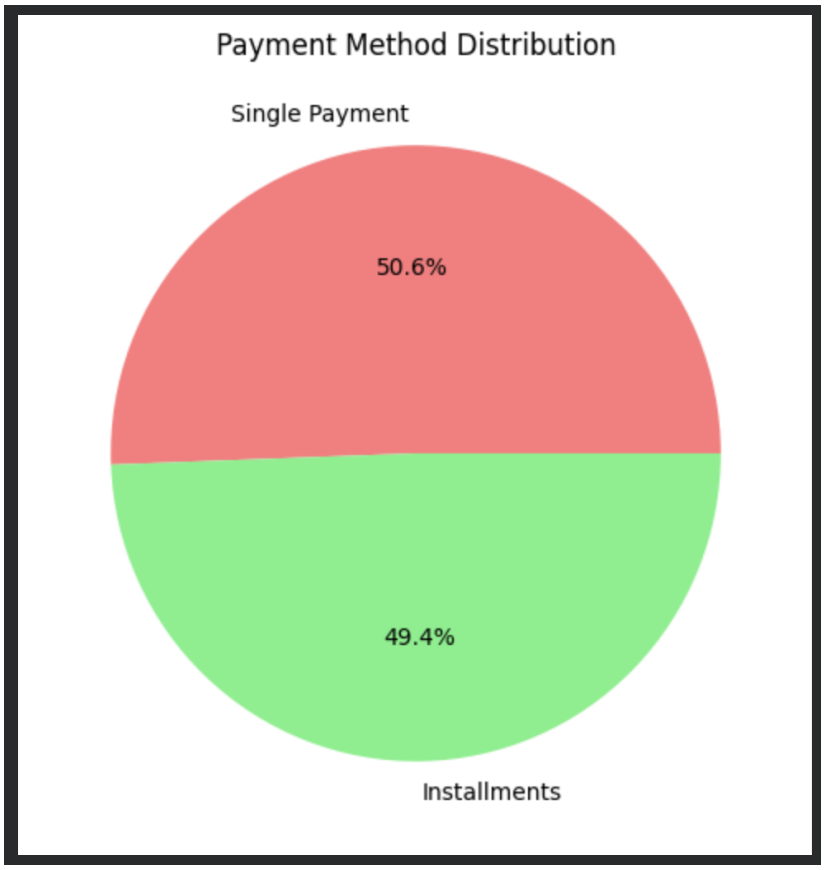


SQL:

	product_category_name	total_revenue
▶	HEALTH BEAUTY	1297355.80
	Watches present	1253143.30
	bed table bath	1092551.02
	sport leisure	1023996.34
	computer accessories	942277.57
	Furniture Decoration	765093.89
	housewares	666587.00
	Cool Stuff	662309.49
	automotive	616752.51
	Garden tools	518217.54

Q4. Calculate the percentage of orders that were paid in more than one installment.

Python:

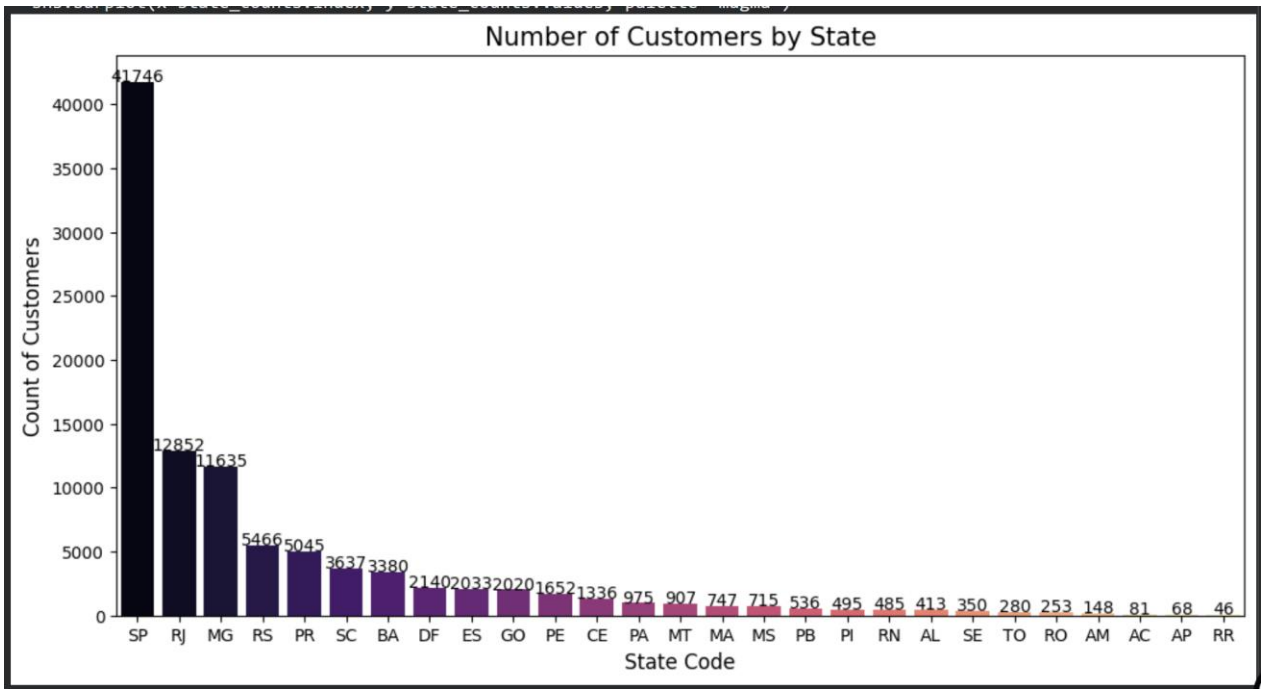


SQL:

	payment_type	total_count	percentage
▶	Installments	51338	49.4
	Single Payment	52548	50.6

5. Count the number of customers from each state.

Python:



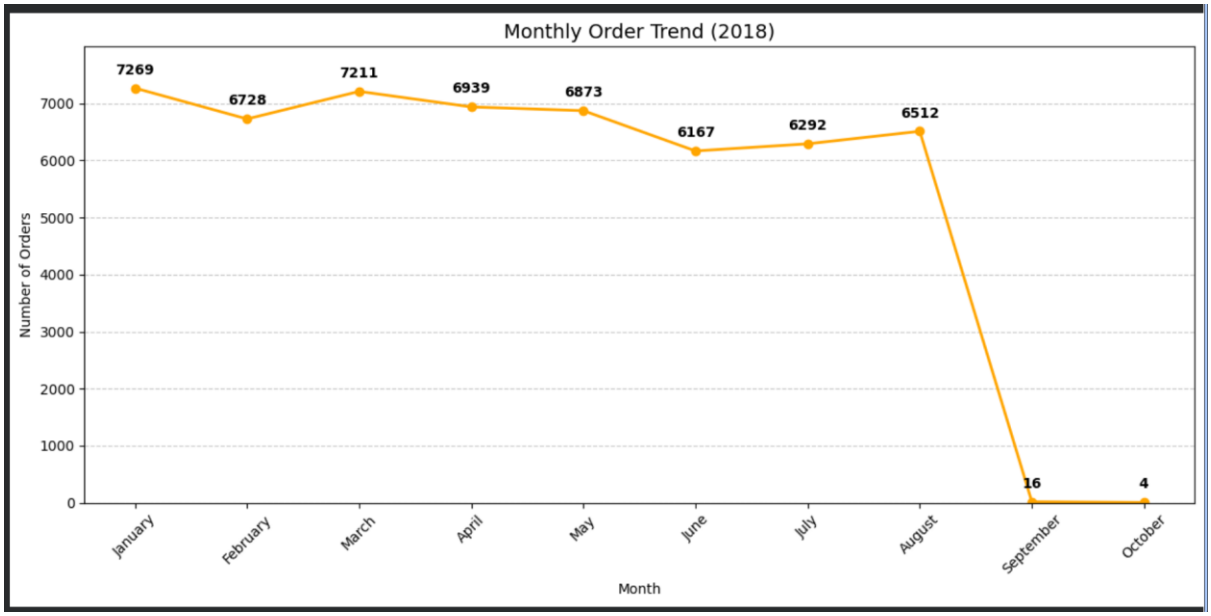
SQL:

customer_state	total_customers
SP	41746
RJ	12852
MG	11635
RS	5466
PR	5045
SC	3637
BA	3380
DF	2140
ES	2033
GO	2020

Intermediate Level Questions

Q1. Orders per Month in 2018.

Python:

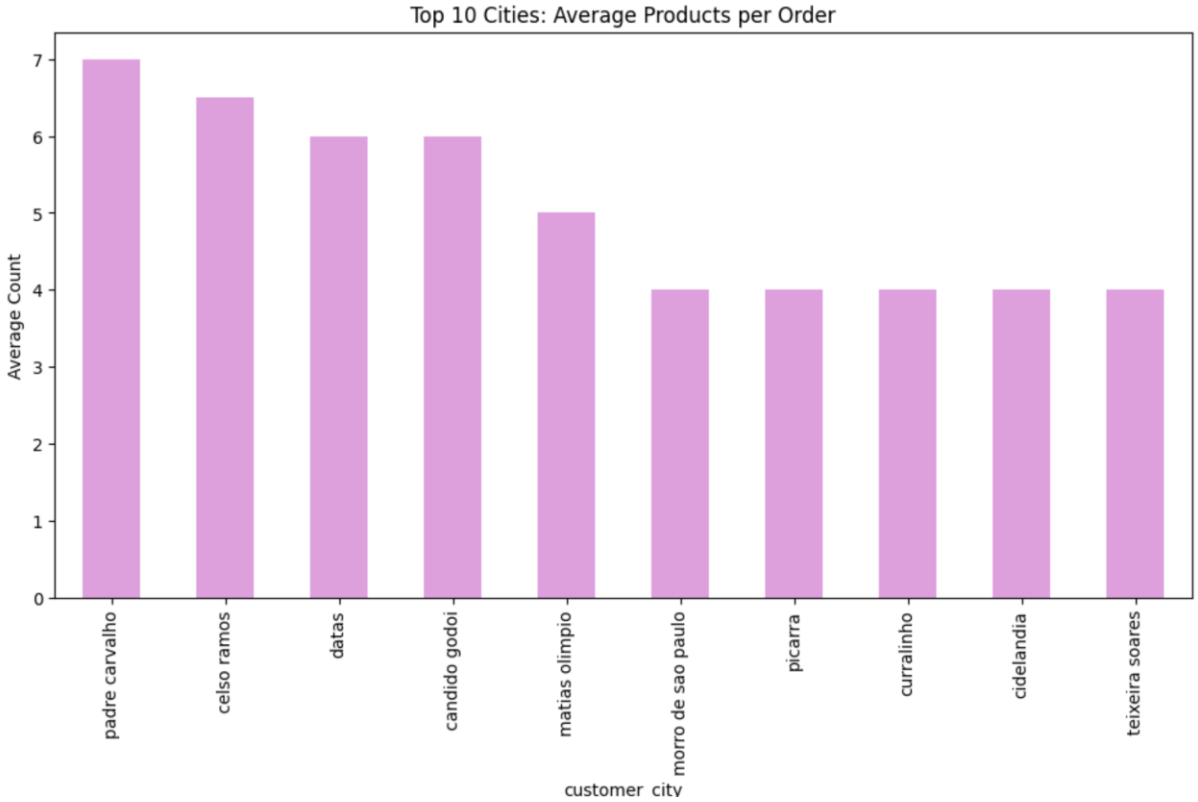


SQL:

month_name	total_orders
January	7269
February	6728
March	7211
April	6939
May	6873
June	6167
July	6292
August	6512
September	16
October	4

Q2. Average Number of Products per Order (Grouped by City).

Python:

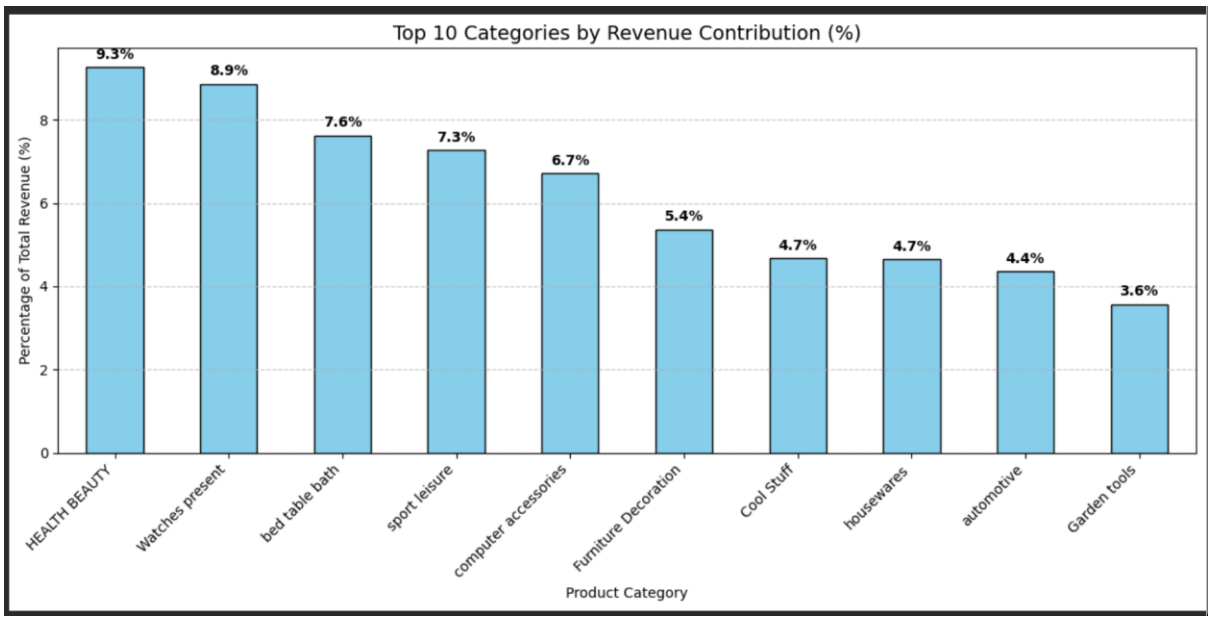


SQL:

customer_city	avg_products_per_order
padre carvalho	7.00
celso ramos	6.50
candido godoi	6.00
datas	6.00
matias olimpio	5.00
picarra	4.00
cidelandia	4.00
morro de sao paulo	4.00
teixeira soares	4.00
curralinho	4.00

Q3. Percentage of Total Revenue by Product Category.

Python:

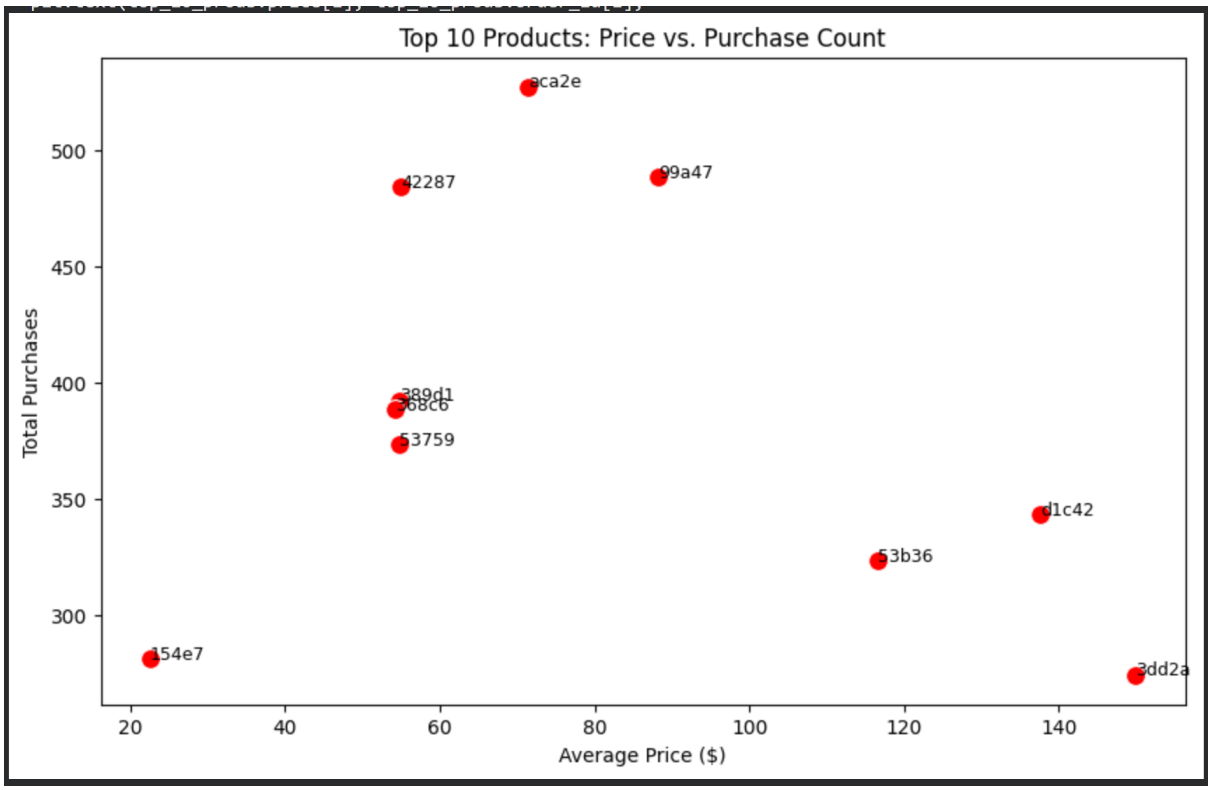


SQL:

	product_category_name	revenue_percentage
▶	HEALTH BEAUTY	9.3
	Watches present	8.9
	bed table bath	7.6
	sport leisure	7.3
	computer accessories	6.7
	Furniture Decoration	5.4
	housewares	4.7
	Cool Stuff	4.7
	automotive	4.4
	Garden tools	3.6

Q4. Correlation Between Product Price and Purchase Count.

Python:

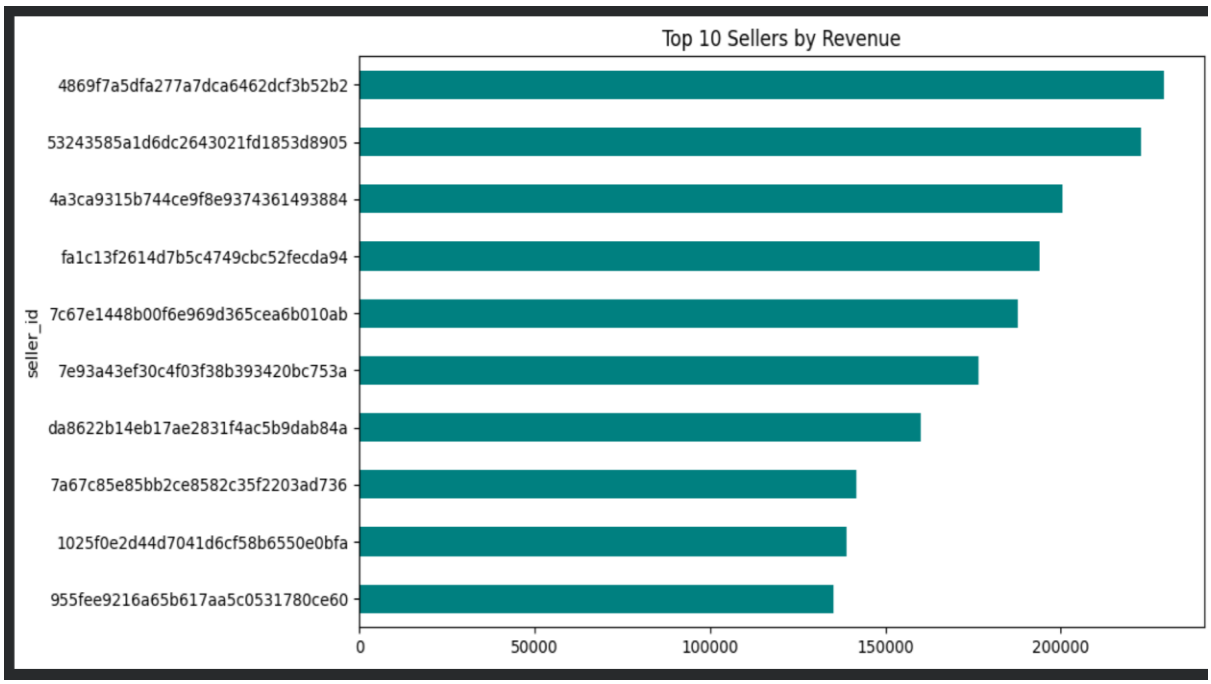


SQL:

product_id	product_category_name	mean_price	purchase_count
aca2eb7d00ea1a7b8ebd4e68314663af	Furniture Decoration	71.364137	527
99a4788cb24856965c36a24e339b6058	bed table bath	88.167131	488
422879e10f46682990de24d770e7f83d	Garden tools	54.911612	484
389d119b48cf3043d311335e499d9c6b	Garden tools	54.695383	392
368c6c730842d78016ad823897a372db	Garden tools	54.270103	388
53759a2ecddad2bb87a079a1f1519f73	Garden tools	54.657373	373
d1c427060a0f73f6b889a5c7c61f2ac4	computer accessories	137.651633	343
53b36df67ebb7c41585e8d54d6772e08	Watches present	116.666935	323
154e7e31ebfa092203795c972e5804a6	HEALTH BEAUTY	22.509573	281
3dd2a17168ec895c781a9191c1e95ad7	computer accessories	149.936496	274

Q5. Total Revenue Generated by Each Seller (Ranked).

Python:



SQL:

seller_id	total_revenue	seller_rank
4869f7a5dfa277a7dca6462dcf3b52b2	229472.63	1
53243585a1d6dc2643021fd1853d8905	222776.05	2
4a3ca9315b744ce9f8e9374361493884	200472.92	3
fa1c13f2614d7b5c4749cbc52fecda94	194042.03	4
7c67e1448b00f6e969d365cea6b010ab	187923.89	5
7e93a43ef30c4f03f38b393420bc753a	176431.87	6
da8622b14eb17ae2831f4ac5b9dab84a	160236.57	7
7a67c85e85bb2ce8582c35f2203ad736	141745.53	8
1025f0e2d44d7041d6cf58b6550e0bfa	138968.55	9
955fee9216a65b617aa5c0531780ce60	135171.70	10

Advanced Level Question

Q1. Moving Average of Order Values.

Python:

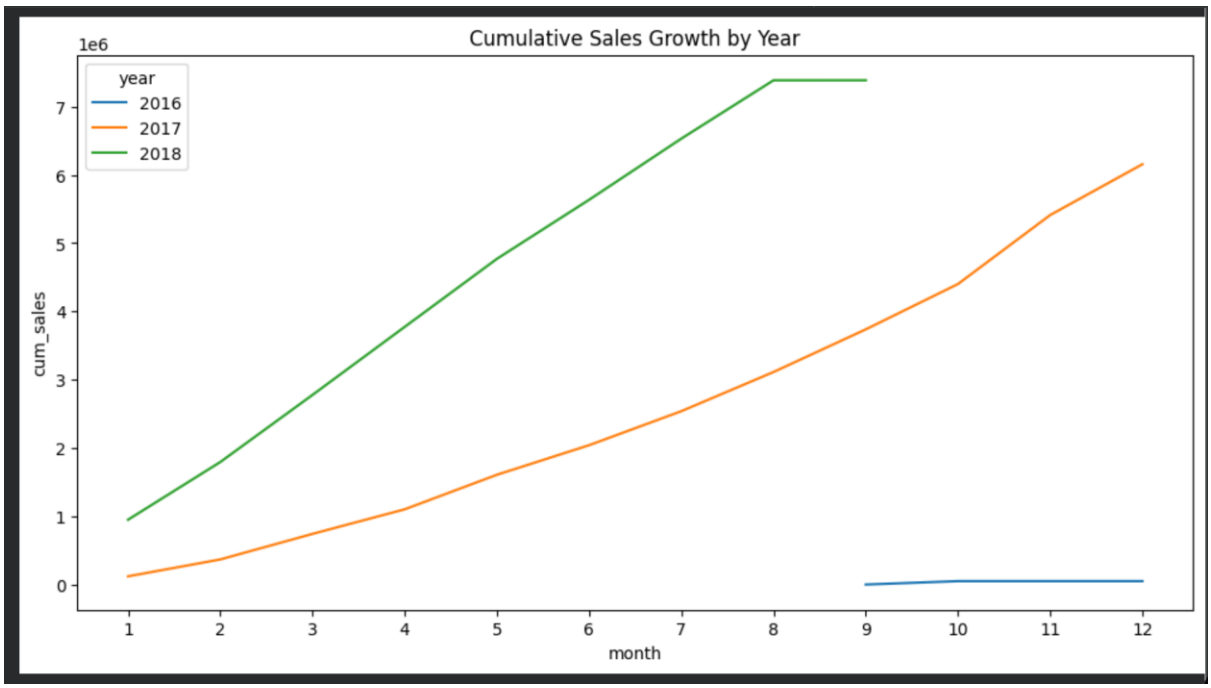
```
...                                customer_id order_purchase_timestamp  \
    71587  00012a2ce6f8dcda20d059ce98491703      2017-11-14 16:08:26
    10466  000161a058600d5901f007fab4c27140      2017-07-16 09:40:32
    68795  0001fd6190edaaf884bcaf3d49edf079      2017-02-28 11:06:43
    45159  0002414f95344307404f0ace7a26f1d5      2017-08-16 13:09:20
    6119   000379cdec625522490c315e70c7a9fb      2018-04-02 13:42:17
    76895  0004164d20a9e969af783496f3408652      2017-04-12 08:35:12
    48295  000419c5494106c306a97b5635748086      2018-03-02 17:47:40
    62657  00046a560d407e99b969756e0b10f282      2017-12-18 11:08:30
    82776  00050bf6e01e69d5c0fd612f1bcfb69c      2017-09-17 16:04:44
    83947  000598caf2ef4117407665ac33275130      2018-08-11 12:14:35

           payment_value  moving_avg
    71587         114.74      114.74
    10466          67.41       67.41
    68795         195.42      195.42
    45159         179.35      179.35
    6119          107.01      107.01
    76895          71.80       71.80
    48295          49.40       49.40
    62657         166.59      166.59
    82776          85.23       85.23
    83947        1255.71     1255.71
```

SQL:

customer_id	order_purchase_timestamp	payment_value	moving_avg
00012a2ce6f8dcda20d059ce98491703	2017-11-14 16:08:26	114.74	114.740000
000161a058600d5901f007fab4c27140	2017-07-16 09:40:32	67.41	67.410000
0001fd6190edaaf884bcaf3d49edf079	2017-02-28 11:06:43	195.42	195.420000
0002414f95344307404f0ace7a26f1d5	2017-08-16 13:09:20	179.35	179.350000
000379cdec625522490c315e70c7a9fb	2018-04-02 13:42:17	107.01	107.010000
0004164d20a9e969af783496f3408652	2017-04-12 08:35:12	71.80	71.800000
000419c5494106c306a97b5635748086	2018-03-02 17:47:40	49.40	49.400000
00046a560d407e99b969756e0b10f282	2017-12-18 11:08:30	166.59	166.590000
00050bf6e01e69d5c0fd612f1bcfb69c	2017-09-17 16:04:44	85.23	85.230000
000598caf2ef4117407665ac33275130	2018-08-11 12:14:35	1255.71	1255.710000

Q2. Cumulative Sales per Month for Each Year.

Python:

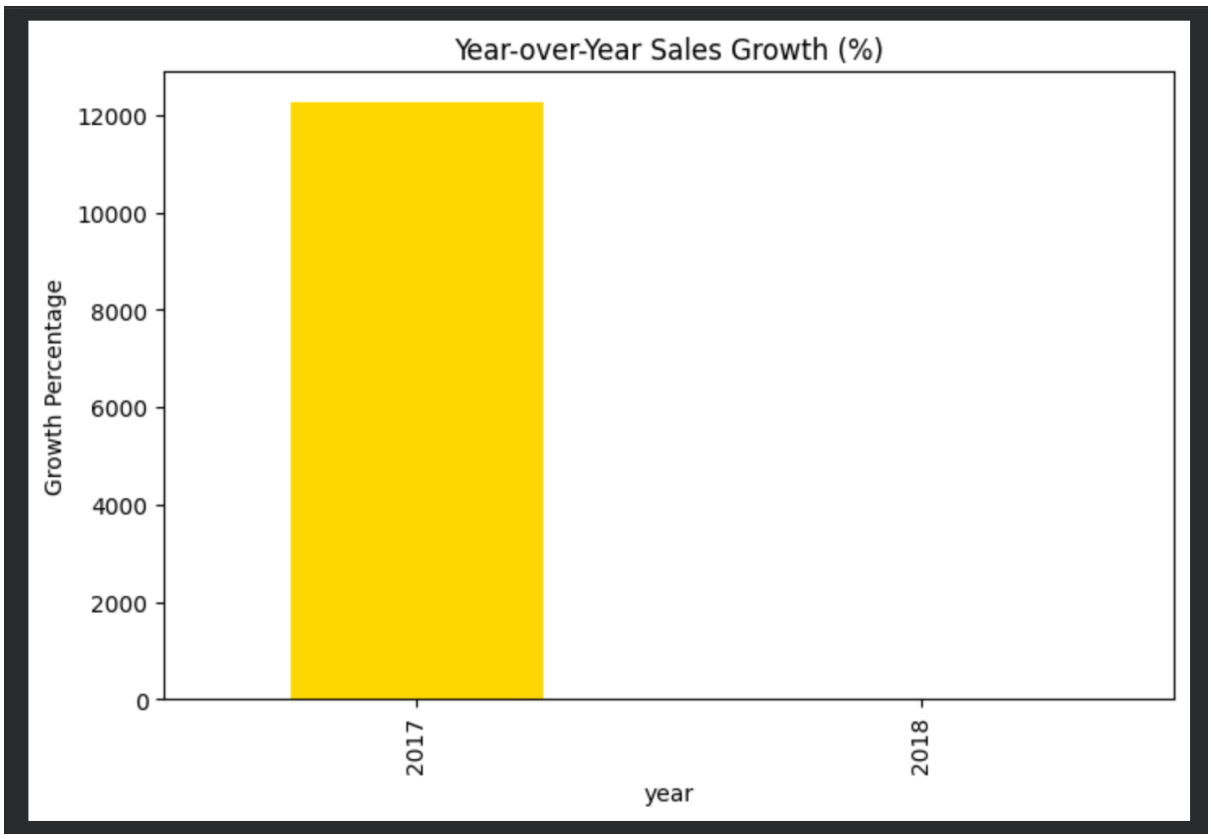


SQL:

	sales_year	sales_month	monthly_revenue	cumulative_sales
▶	2016	9	252.24	252.24
	2016	10	59090.48	59342.72
	2016	12	19.62	59362.34
	2017	1	138488.04	138488.04
	2017	2	291908.01	430396.05
	2017	3	449863.60	880259.65
	2017	4	417788.03	1298047.68
	2017	5	592918.82	1890966.50
	2017	6	511276.38	2402242.88
	2017	7	592382.92	2994625.80
	2017	8	674396.32	3669022.12
	2017	9	727762.45	4396784.57
	2017	10	779677.88	5176462.45
	2017	11	1194882.80	6371345.25
	2017	12	878401.48	7249746.73

Q3. Year-over-Year (YoY) Growth Rate.

Python:

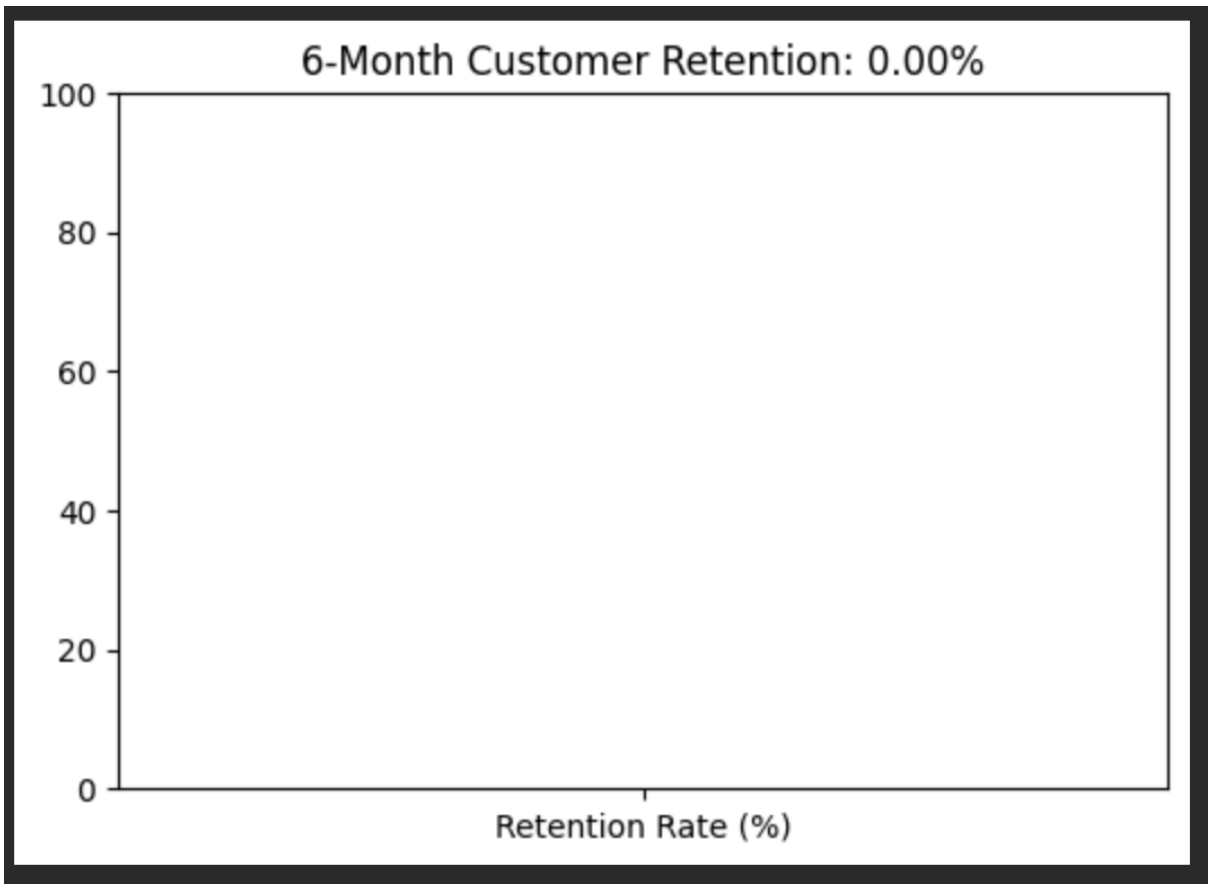


SQL:

year	total_revenue	previous_year_revenue	yoy_growth
2016	59362.34	NULL	NULL
2017	7249746.73	59362.34	12112.703761
2018	8699763.05	7249746.73	20.000924

Q4. Retention Rate (Repeat Purchase within 6 Months).

Python:

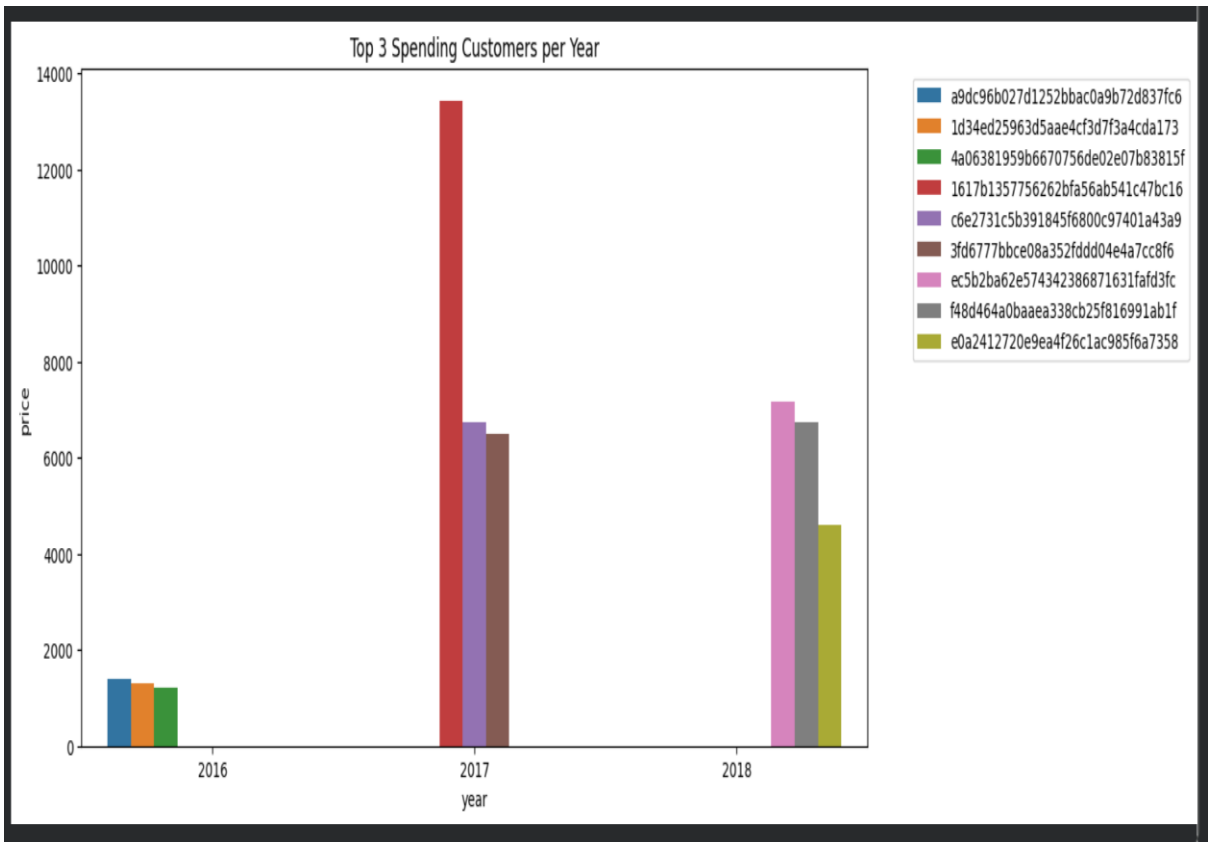


SQL:

	retention_rate
▶	0.0000

Q5. Top 3 Customers Who Spent the Most Each Year.

Python:



SQL:

year	customer_id	total_spent	spend_rank
2016	a9dc96b027d1252bbac0a9b72d837fc6	1423.55	1
2016	1d34ed25963d5aae4cf3d7f3a4cda173	1400.74	2
2016	4a06381959b6670756de02e07b83815f	1227.78	3
2017	1617b1357756262bfa56ab541c47bc16	13664.08	1
2017	c6e2731c5b391845f6800c97401a43a9	6929.31	2
2017	3fd6777bbce08a352fddd04e4a7cc8f6	6726.66	3
2018	ec5b2ba62e574342386871631fafd3fc	7274.88	1
2018	f48d464a0baaea338cb25f816991ab1f	6922.21	2
2018	e0a2412720e9ea4f26c1ac985f6a7358	4809.44	3