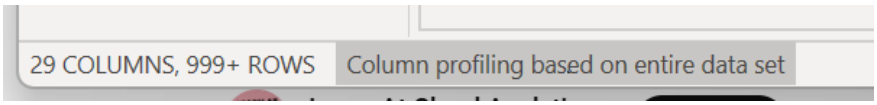
ASSIGNMENT-4

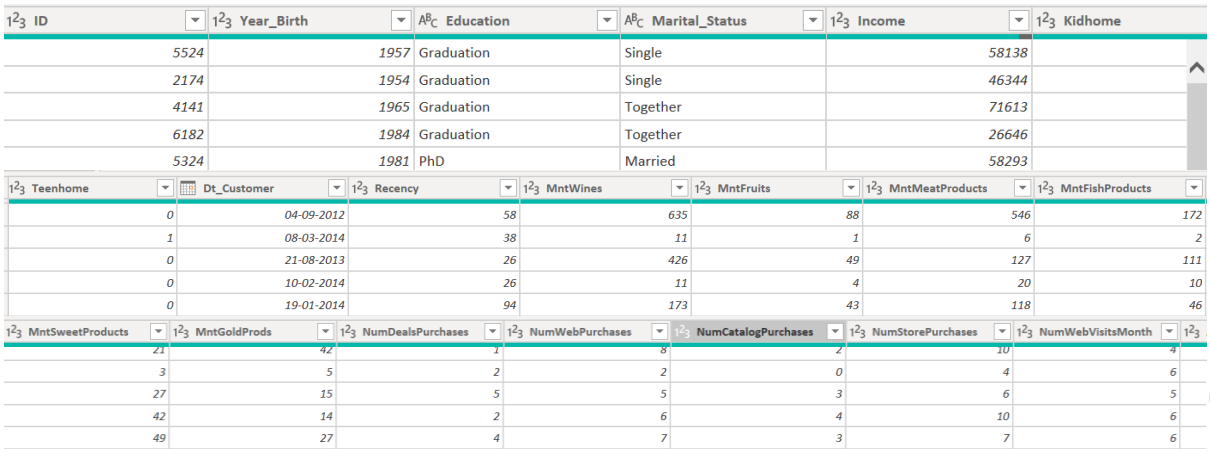
TASK-1: DATA VISUALIZATION AND PRE-PROCESSING USING POWER BI

DATA PRE-PROCESSING:



Changed the column profiling to Column profile not just on 1000 rows but on the entire dataset.

1. Ensure data type of each column: Correct

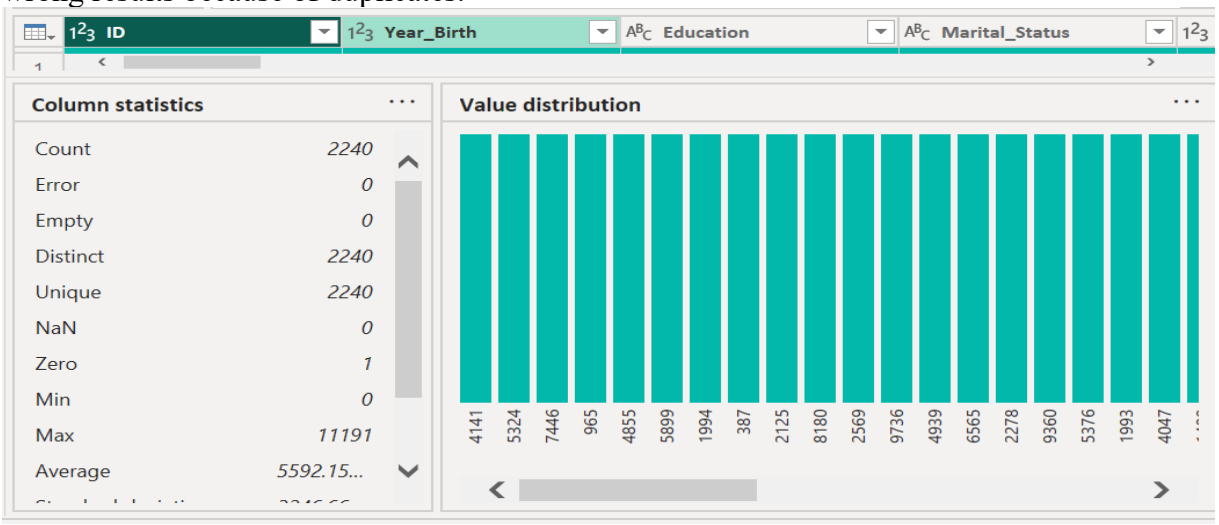


ID	Year_Birth	Education	Marital_Status	Income	Kidhome
5524	1957	Graduation	Single	58138	
2174	1954	Graduation	Single	46344	
4141	1965	Graduation	Together	71613	
6182	1984	Graduation	Together	26646	
5324	1981	PhD	Married	58293	

Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts
0	04-09-2012	58	635	88	546	172
1	08-03-2014	38	11	1	6	2
0	21-08-2013	26	426	49	127	111
0	10-02-2014	26	11	4	20	10
0	19-01-2014	94	173	43	118	46

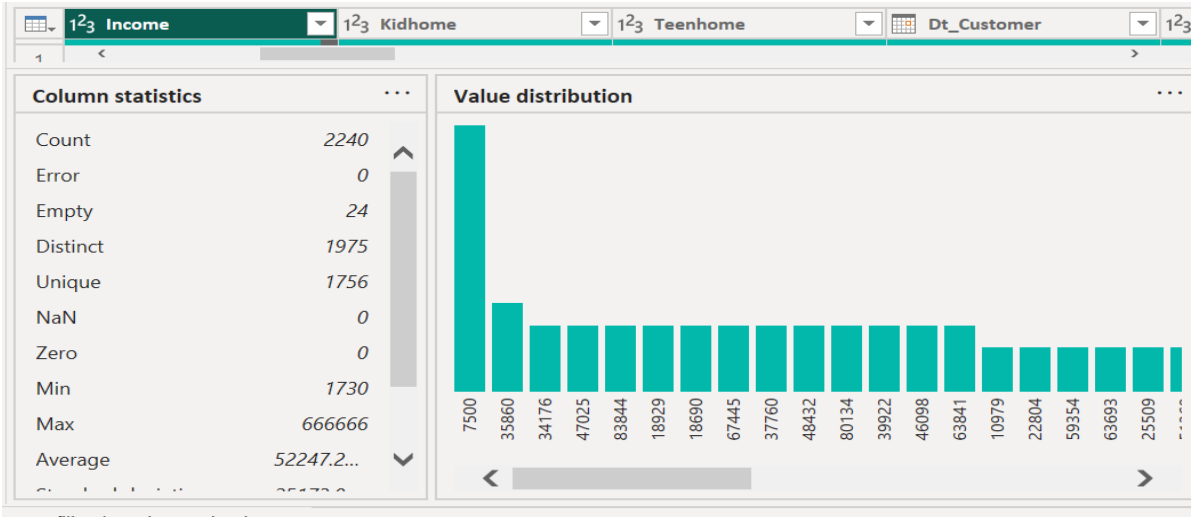
MntSweetProducts	MntGoldProds	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth
21	42	1	8	2	10	4
3	5	2	2	0	4	6
27	15	5	5	3	6	5
42	14	2	6	4	10	6
49	27	4	7	3	7	6

2. Remove duplicates: Applied this on ID column, no duplicates found. No other column will give wrong results because of duplicates.

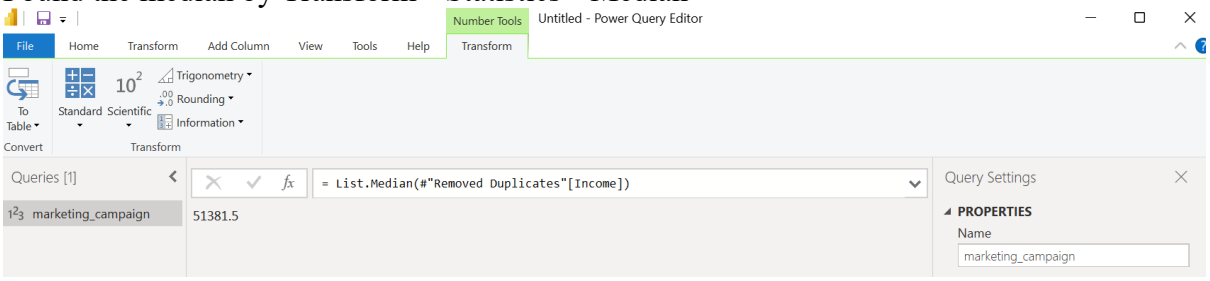


All unique.

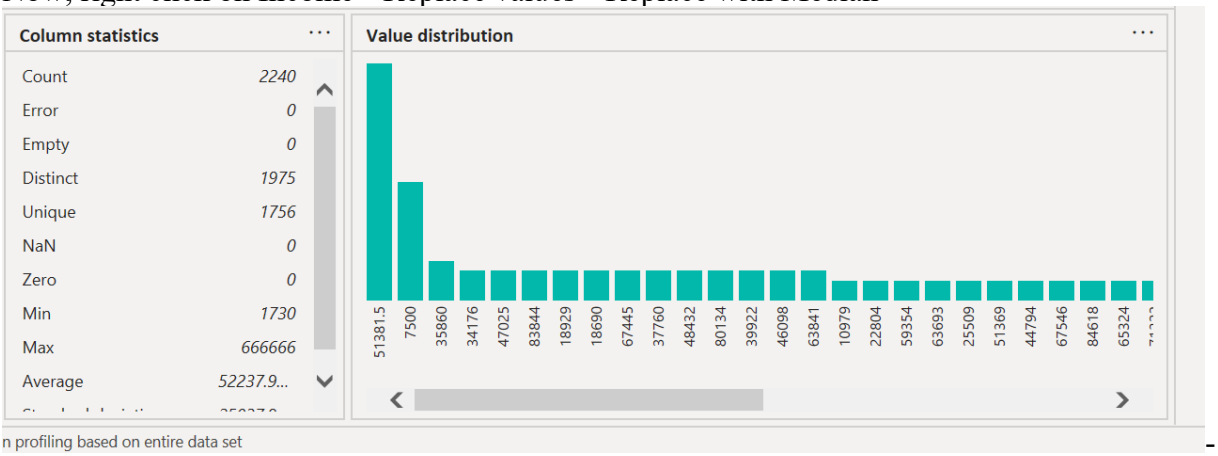
3. Substitute/remove NULL values:
   a. 24 cells empty in Income-> substitute with Median values of the column



Found the median by Transform->Statistics->Median



Now, right click on Income->Replace values->Replace with Median

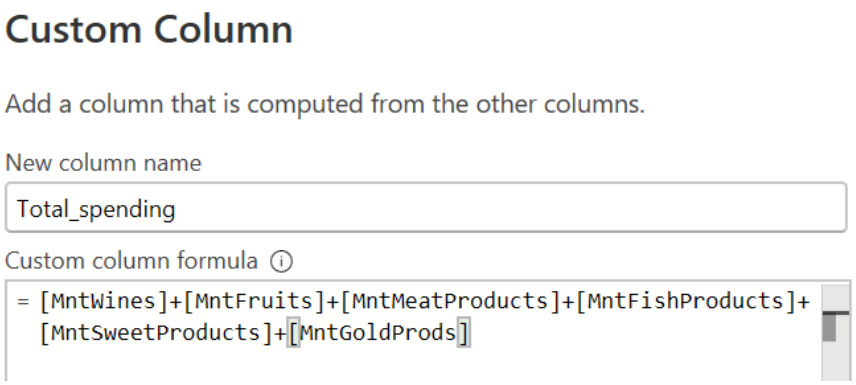


No other column having EMPTY/ NULL values.

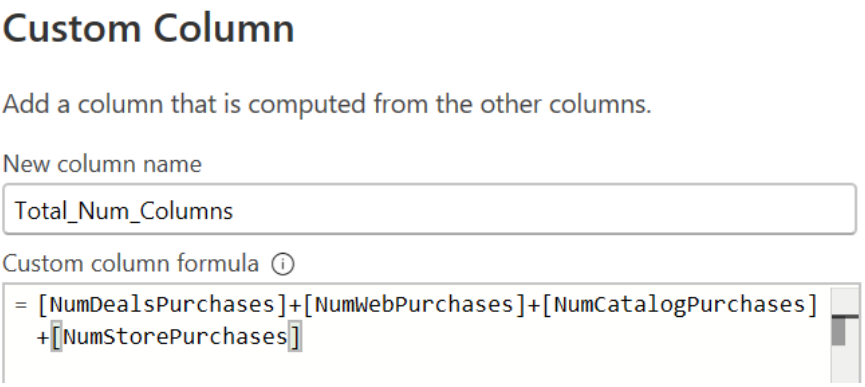
4. Derived Columns
   a. Age=2025-[Year_Birth]

And remove the column Year_Birth

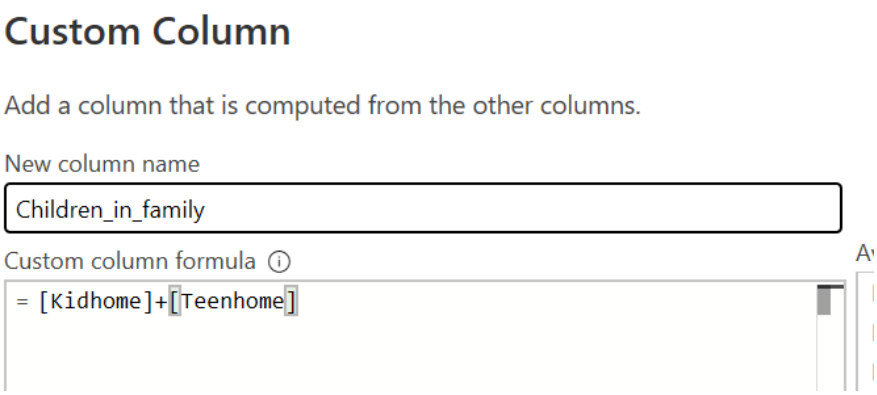
   b. Total_Spending



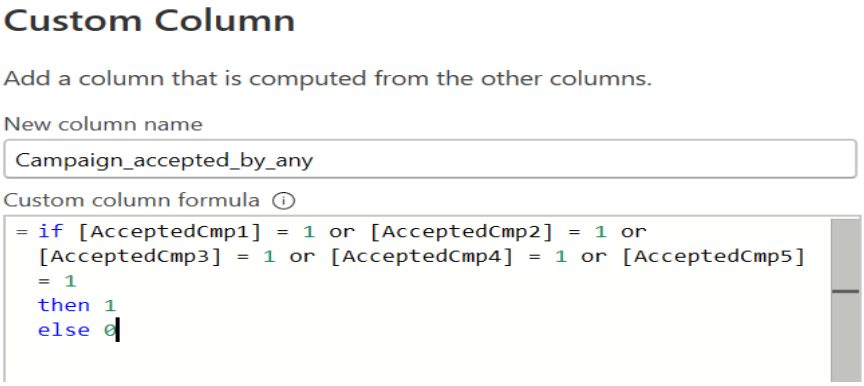
c. Total_Num_Columns



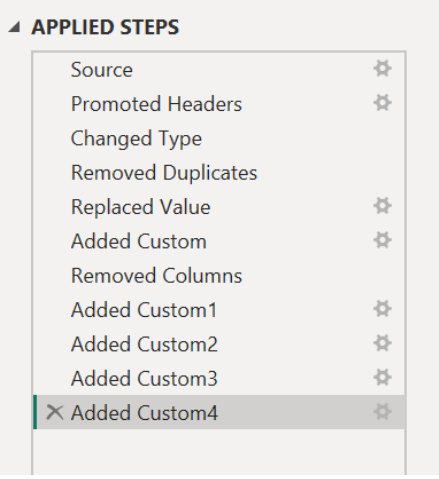
d. Children_in_family



e. Campaign_Accpeted_by_any



5. Trim/clean: Eg. Changing case so that the whole data is similar (Already constant as seen in column profile so not necessary)
6. Binning numerical columns (Done for histogram visualisation afterwards)

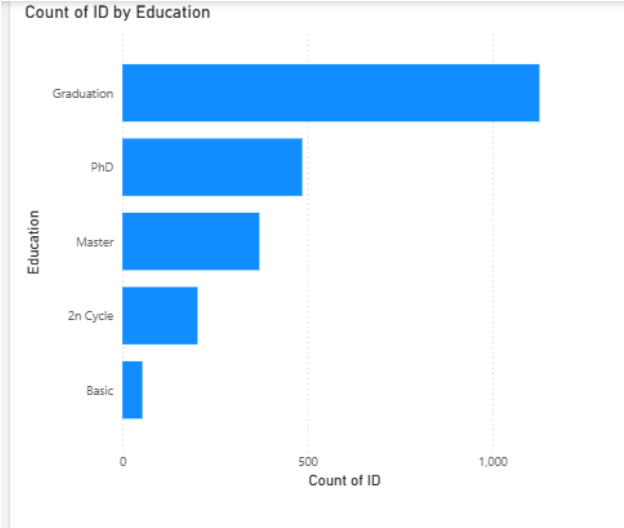


Changed the data type of newly created columns to Whole numbers after this.

VISUALISATION:

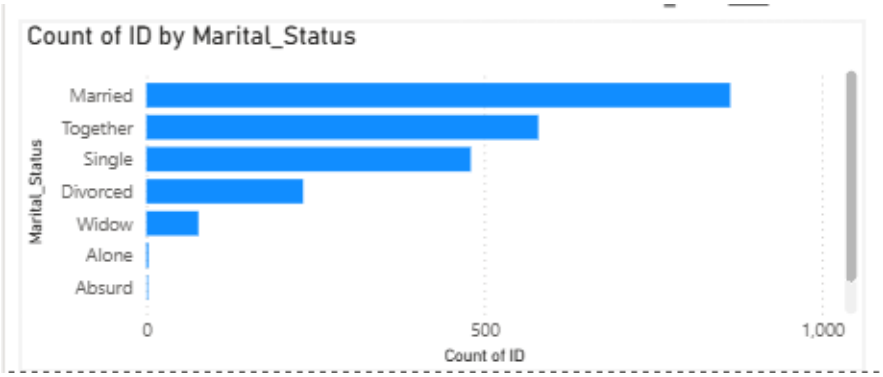
CATEGORICAL VARIABLES:

1. Education



Customer base in the dataset is highly educated with most of them being graduates.

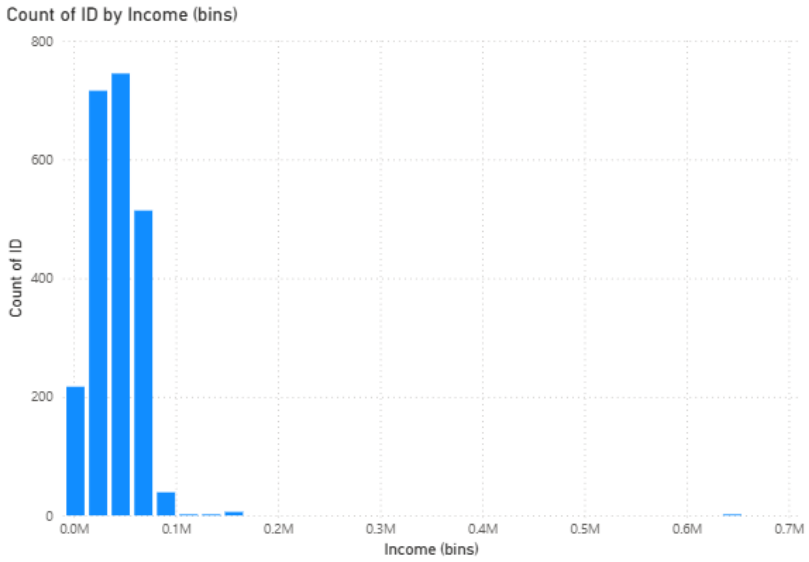
2. Marital status



The majority being married suggests that spending decision maybe family oriented. Absurd should have been cleaned from the dataset.

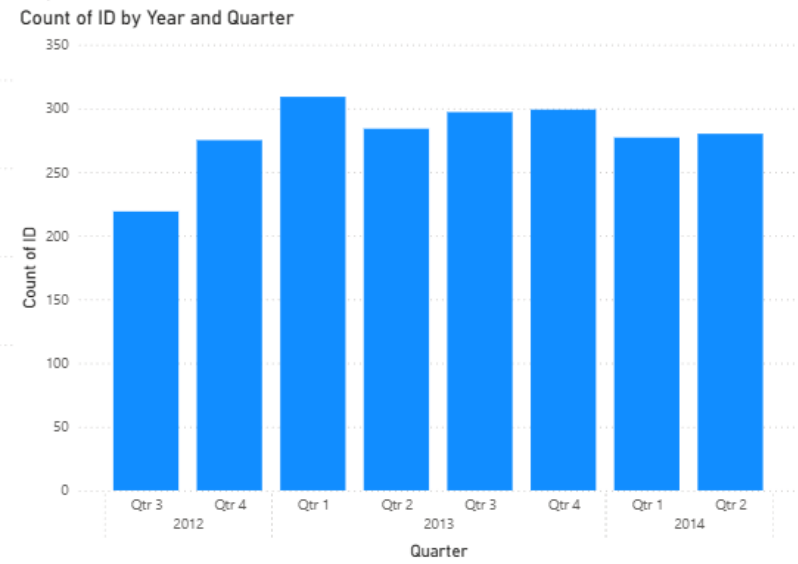
NUMERICAL VARIABLES:

1. Income (Bin size=2666.66, because of equal sized bins property by default)



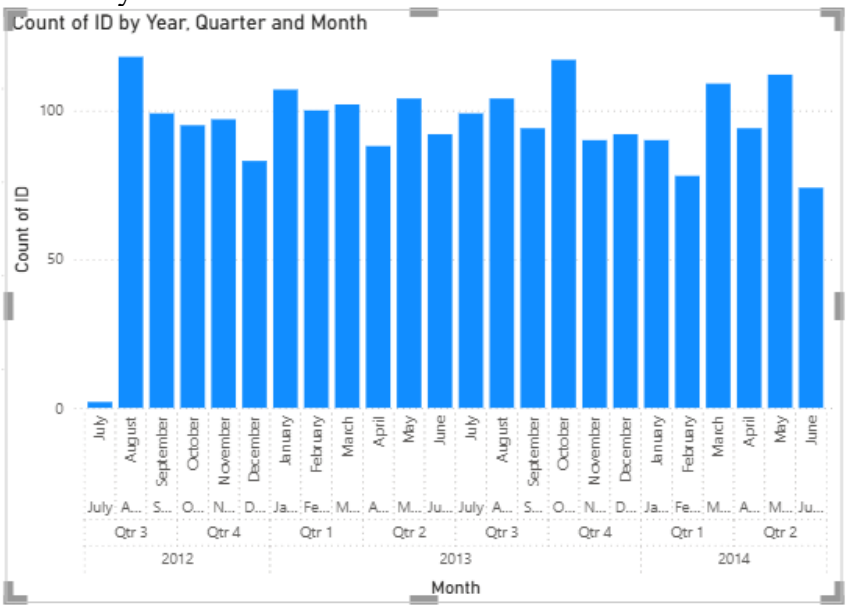
Most customers fall between 20k to 80k income range with the highest group being 40k-60k. A tiny number of people have extremely high income (600K-700k). These are outliers but need not be removed as they do exist in society and removing them might give wrong results and analysis.

2. Year_birth
   By quarter:



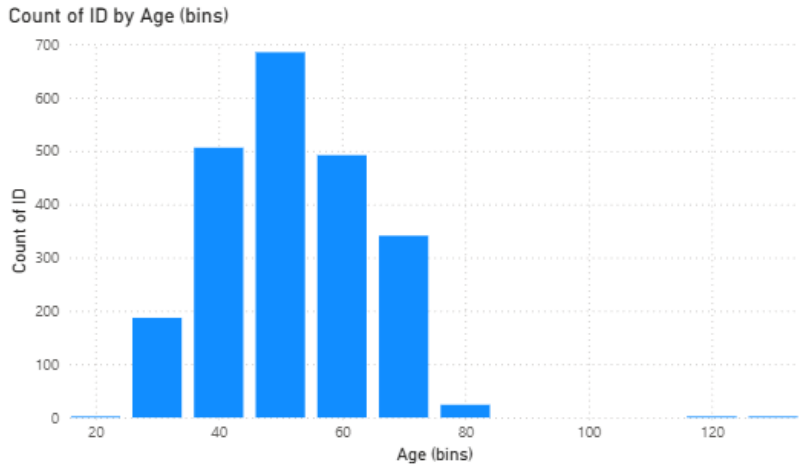
2012 Quarter 3 has lowest count, whilt 2013 first quarter has highest. Hence, 2013 overall is more stable and customers peaked more in early 2013. 2014 is slightly declining from peak but still stable.

By months:



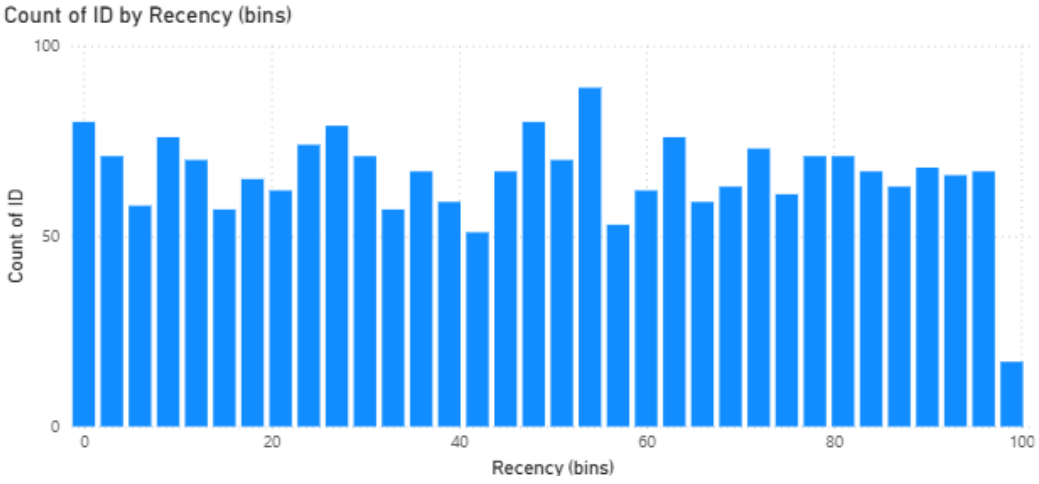
July 2012 has almost 0 count... that is very less customers. Can observe that spikes are often happening in the early month of each quarter (eg. Jan, Apr, Jul, Oct).

3. Age (binsize=10):



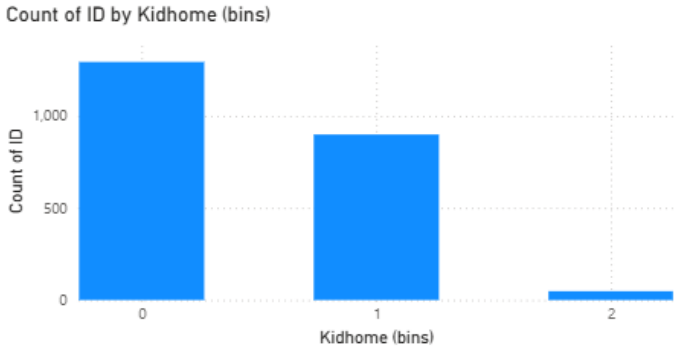
Peak exists betwen 40-60 age, hence most of the customers belong to this age. 140 is an outlier, we will remove that.

4. Recency (Number of days since the customer's last purchase) (Bin size=3.66)



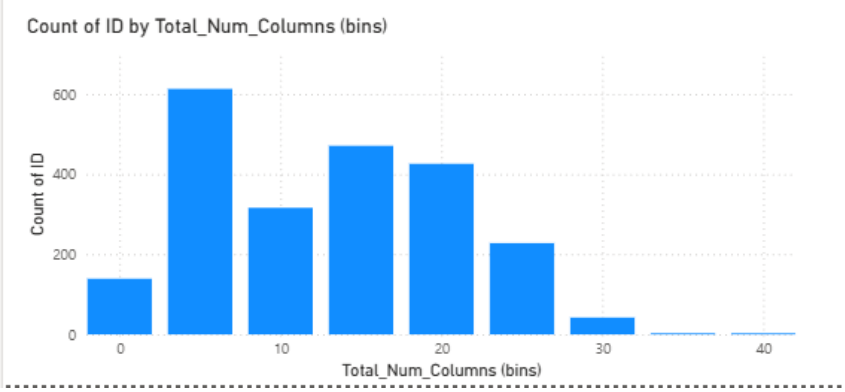
Almost consistent distribution of IDs across the recency.

5. Kidhome (Number of kids at home) (Bin size=1)



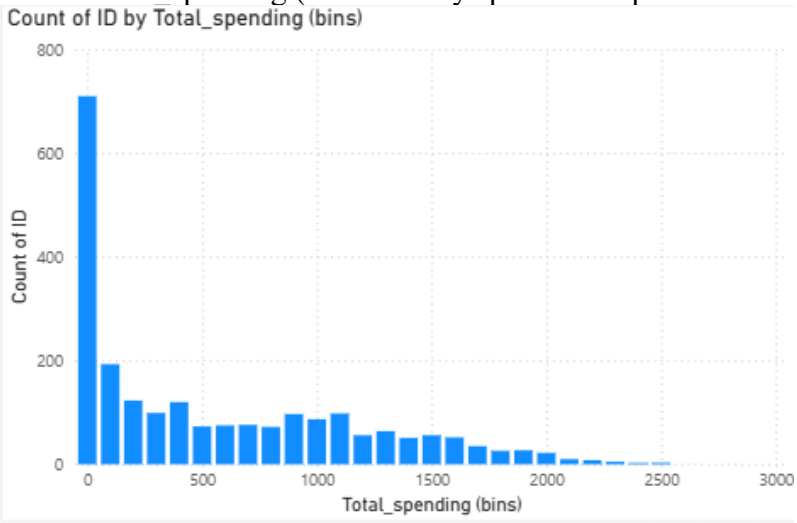
No. Of kids being 0,1,2 in the home. They might be very small in age or teen. This suggests that individuals without children at home dominate the dataset.

6. Total_Num_Columns (Total number of purchases) (Binsize=5)



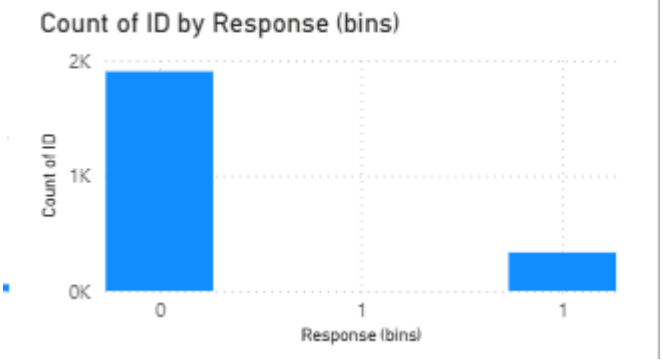
Most of the users fall in 5-25 number of purchases range. Hence, a majority of customers interact with a moderate number of product categories. Only a small segment interacts with a very high number of categories.

7. Total_spending (Total money spent on all products combined) (Binsize=100)



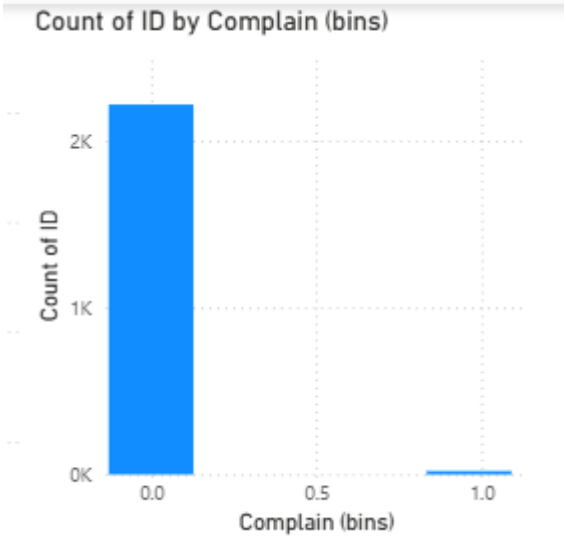
Right-skewed distribution. High concentrations of users with low spending, some customers do spend over 2k but they are extremely rare. These are high spending outliers but I believe they are necessary for interpretation and hence won't remove them, as they can represent bulk buyers.

8. Response (whether they accepted the last campaign (binary 0/1)) (Bin size=1)



0->Did not respond, 1->Did respond. Response rate is very low.

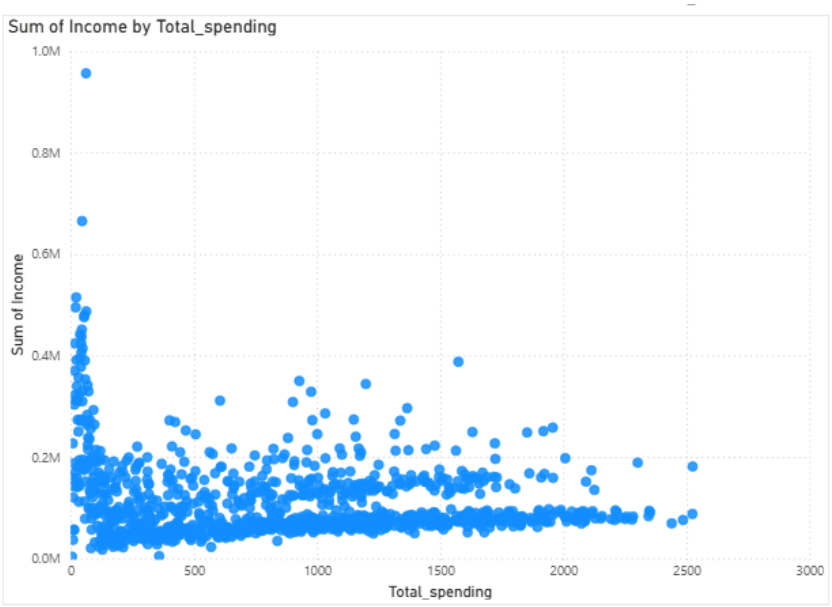
9. Complain (whether they complained in last 2 years (binary 0/1)) (Binsize=0.5)



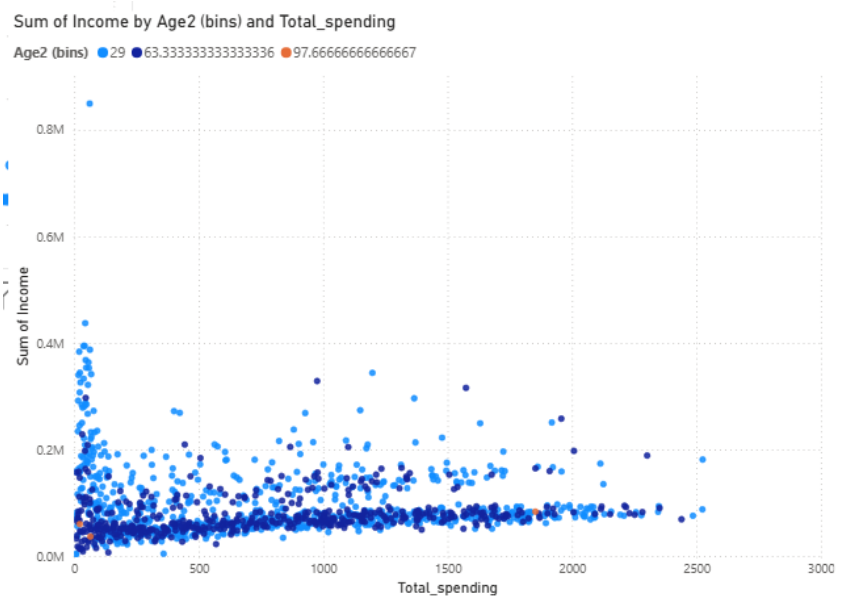
0-> Did not complain, 1->Complaied. Majority of the customers complained.

SCATTER PLOTS:

1. Income v/s Total_spending

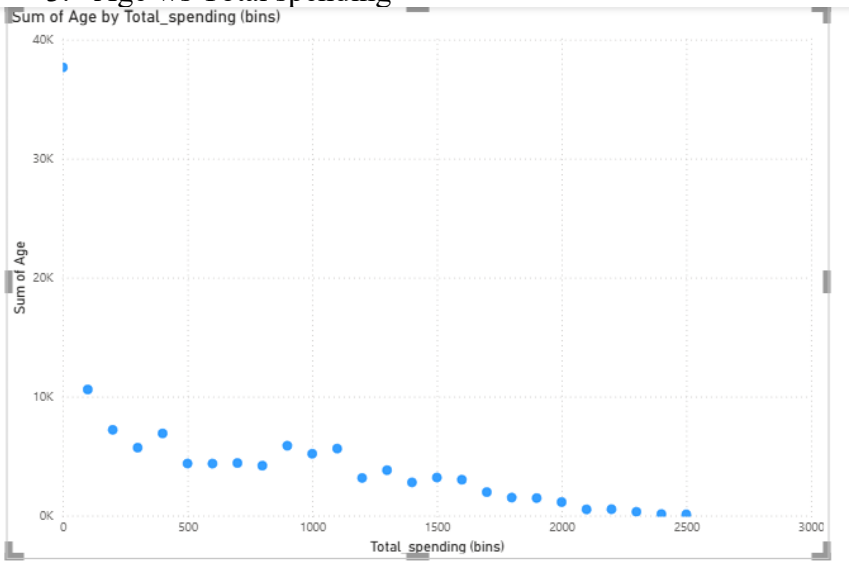


Each point = one customer. Customers will high income do not necessarily spend more. Most of them spend below 0.2M only. And customers spending more are less in number. Very weak positive correlation.



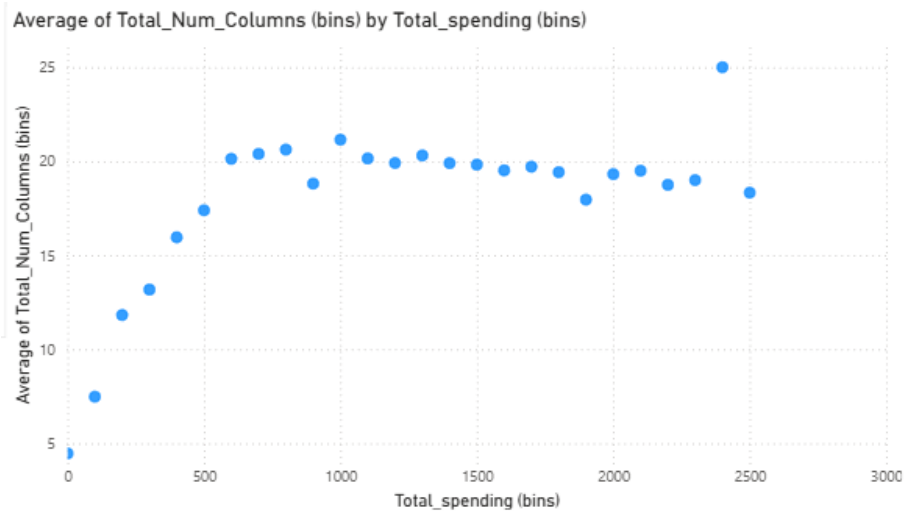
We can see that most of the high spending outliers are people aged on an average 29 while average or low spending are averaged 63 in age.

3. Age v/s Total spending



Same result as above. Negative correlation: As age increases, spending decreases.

4. Total_num_customer (total purchases) v/s Total spending



Positive correlation for the spending till 1000rs. After that its almost constant or -ve. As total spending increase, total purchases also increase. But after some time, even though total spending is increasing, customers might be buying higher cost products and hence total number of purchases is remaining constant only.

CORRELATION PLOT/MATRIX PLOT/HEATMAP

1. Matrix plot (Row->income, Column->age, Value->total spending)

Income (bins)	20	30	40	50	60	70	80	120	130
1,730.00	2	45	76	56	28	10			
23,894.53		59	206	263	109	76	2	1	
46,059.07		23	114	222	222	151	12		1
68,223.60		51	98	133	128	95	8	1	
90,388.13		9	8	8	5	8	1		
1,12,552.67							1		
1,34,717.20				1					
1,56,881.73			3	2		1			
6,44,501.47			1						

People aged 40-60 with incomes around 23-68k spend the most. Very high incomes show low values which means fewer customers or lower purchases. Conclusion-> Spending peaks in middle-income and middle-aged groups.

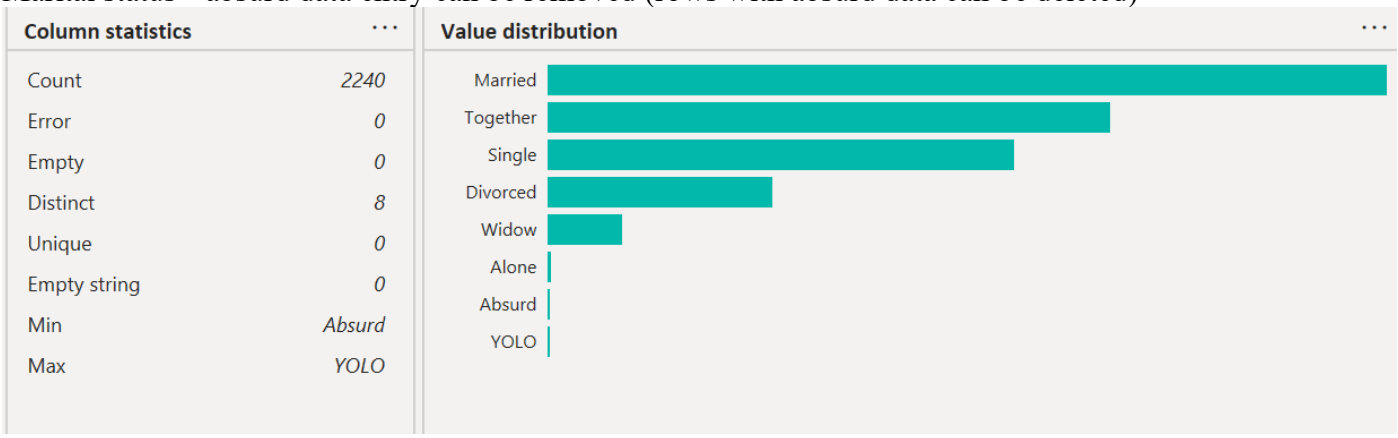
2. Rows->Age, Columns->Education, Value->Total_new_columns(purchases)

Age (bins)	2n Cycle	Basic	Graduation	Master	PhD	**Total**
20	1	1				**2**
30	23	16	105	21	22	**187**
40	68	18	274	69	77	**506**
50	51	10	342	118	164	**685**
60	41	6	238	91	116	**492**
70	17	3	163	65	93	**341**
80			5	6	13	**24**
120	1				1	**2**
130	1					**1**
Total	**203**	**54**	**1127**	**370**	**486**	**2240**

Graduates aged 40-60 make the highest number of purchases. Basic and low education segments contribute very little. Basic and low education segments also contribute little only.

Hence, observation regarding **ANAMOLIES** from the above visualisation are:

1. Marital status->absurd data entry can be removed (rows with absurd data can be deleted)



Removed 2 rows with absurd and 2 with YOLO.

2. Age=140 outlier in the age column and practically not easily possible as well.
   Removed>100 rows

   New dimensions: 33*2233

TASK-2: K-MEANS CLUSTERING IMPLEMENTATION

1. CUSTOMER DEMOGRAPHICS:

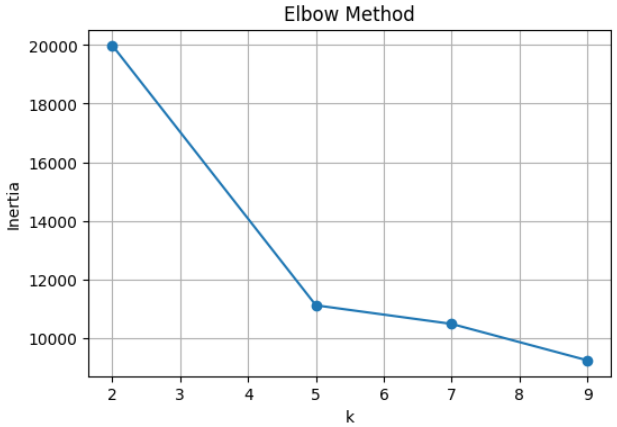
Features: Age, Income, Children_in_family, Marital_Status

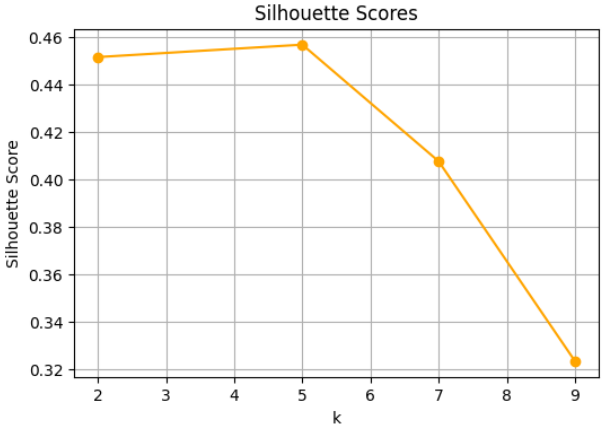
Outcome:

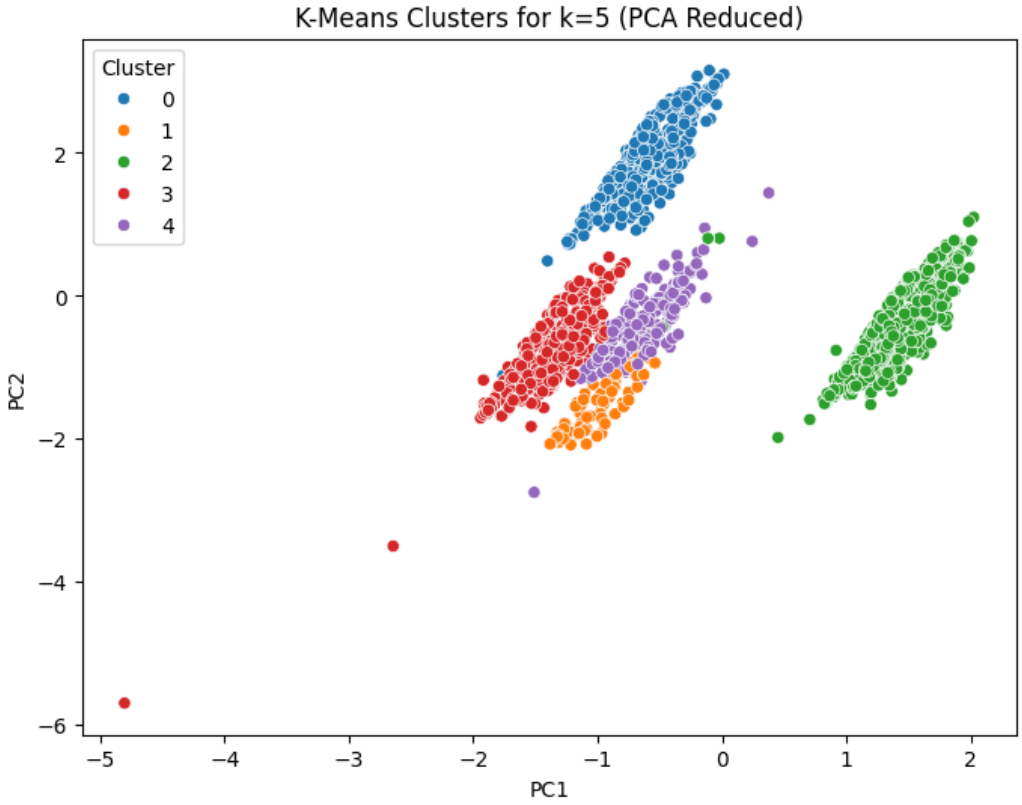
Elbow method is changing at 5 so k=5 is appropriate according to elbow method. Silhouette score as well is highest at k=5. Hence, k=5 is the optimal choice for the number of clusters here.

```
Evaluating K values...

K=2 | Inertia=19989.47 | Silhouette=0.4515
K=5 | Inertia=11107.68 | Silhouette=0.4568
K=7 | Inertia=10472.63 | Silhouette=0.4075
K=9 | Inertia=9227.23 | Silhouette=0.3232
```







PCA here is Principal component analysis.

PCA creates patterns where:

- Points that lie close together = people who have similar values for the 4 features
- Points far apart = people who differ significantly in those features

Cluster 0 (Blue group): Higher income

- Middle or slightly younger age
- Less income
- Children count moderate
- Similar marital status category

Cluster 1 (Green group): Higher income than cluster 0, slightly older, similar number of children, Possibly a distinct marital status grouping. Most separated cluster

Cluster 2 (Orange group):

- Their Age/Income/Children/Marital_Status combination is very different from others.
- Maybe marital status different from others.

Cluster 3 (Red group):

This group overlaps partly with purple and orange, meaning:

- They share some characteristics with those clusters
- Age and income levels are not extreme

Cluster 4 (Purple group, have few outliers, middle cluster)

This cluster sits between others, meaning:

- They have "average" values
- Age and income not extreme
- Moderate children count
- Common marital status

	PC1	PC2
Age	-0.177816	-0.454656
Income	-0.146700	-0.217314
Children_in_family	0.074829	-0.070504
Marital_Status_Alone	0.003974	0.019244
Marital_Status_Divorced	-0.139448	-0.091412
Marital_Status_Married	0.769486	-0.215300
Marital_Status_Single	-0.191812	0.740265
Marital_Status_Together	-0.526975	-0.306856
Marital_Status_Widow	-0.123254	-0.208696
Marital_Status_YOLO	-0.002218	0.017435

Higher the PCA score for the column, more they are contributing in the cluster formation and distinction.

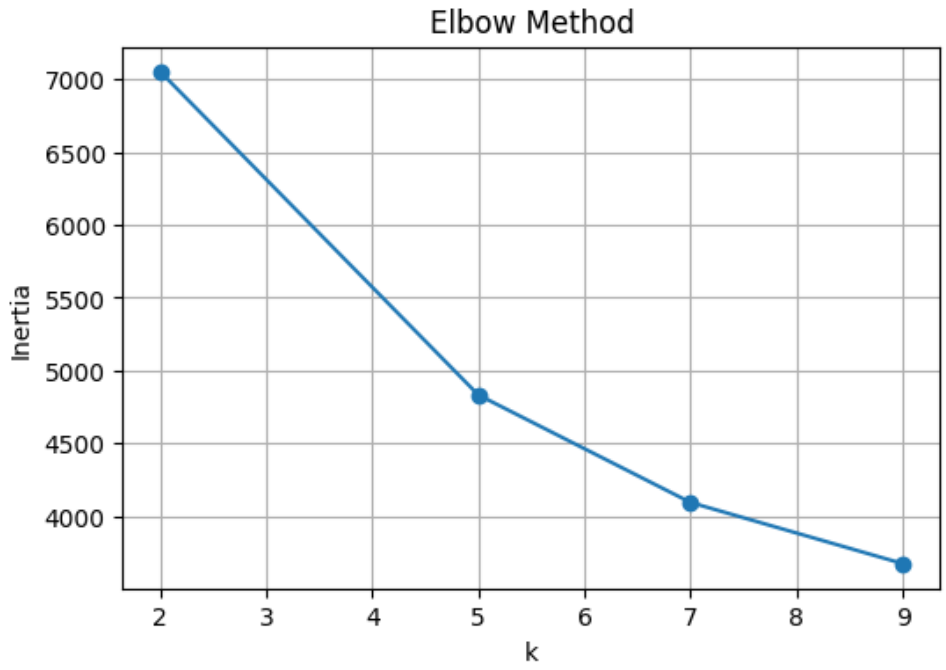
2. CUSTOMER SPENDING AND RECENCY

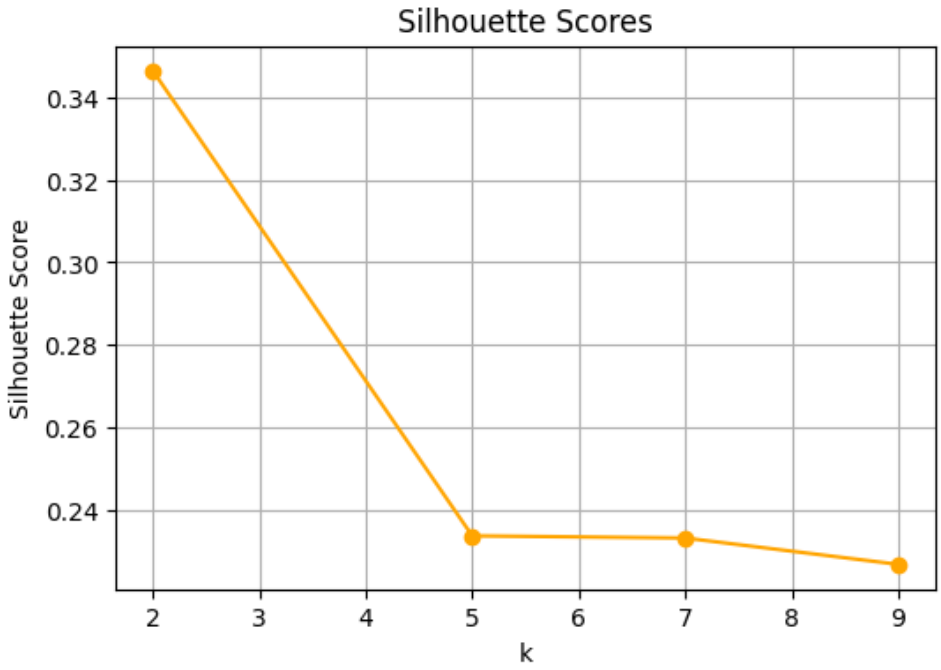
Features: Age, Income, Recency, Total_spending, Total_Num_Columns

Outcome:

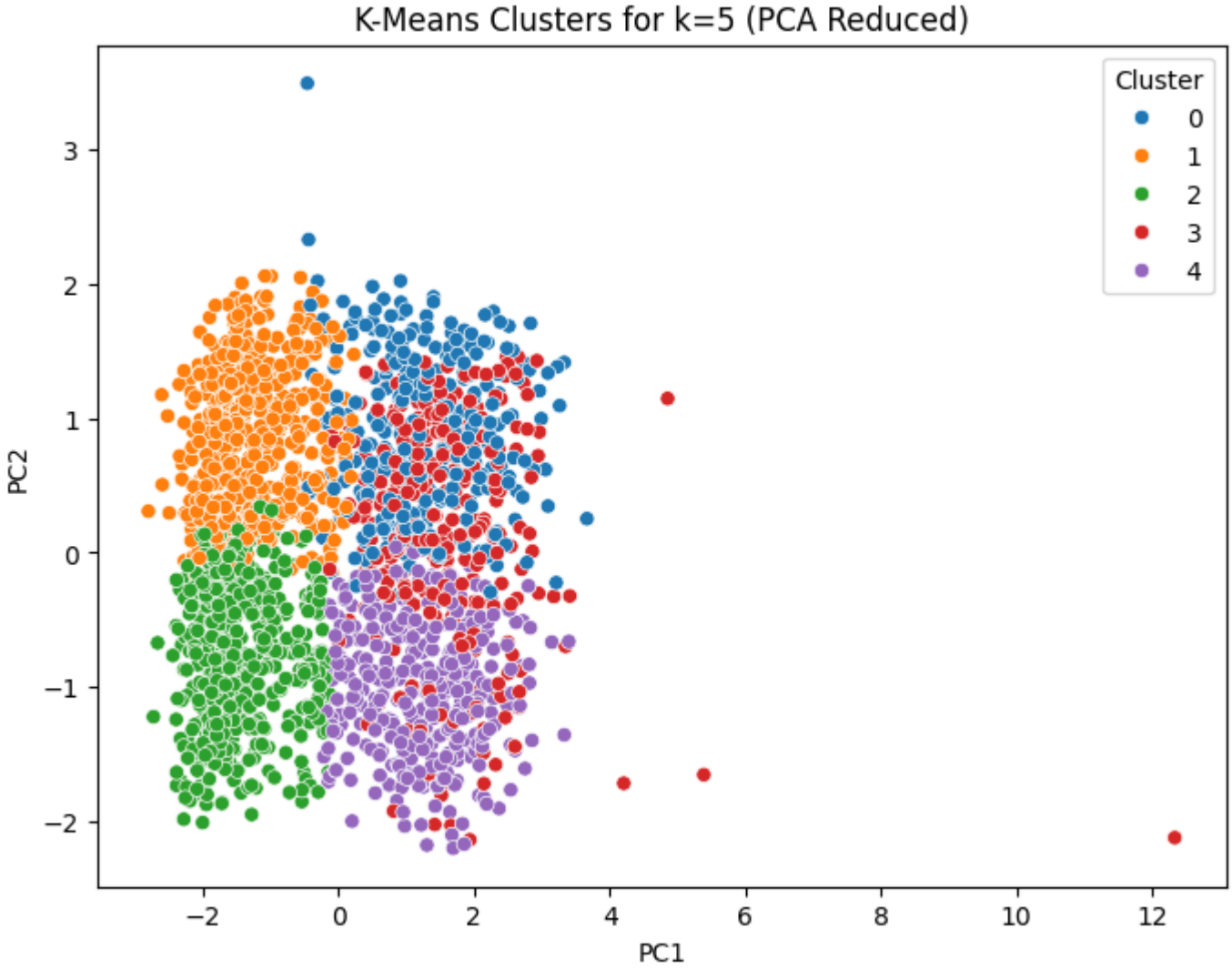
```
Evaluating K values...

K=2 | Inertia=7054.52 | Silhouette=0.3463
K=5 | Inertia=4831.90 | Silhouette=0.2338
K=7 | Inertia=4093.30 | Silhouette=0.2333
K=9 | Inertia=3673.01 | Silhouette=0.2269
```



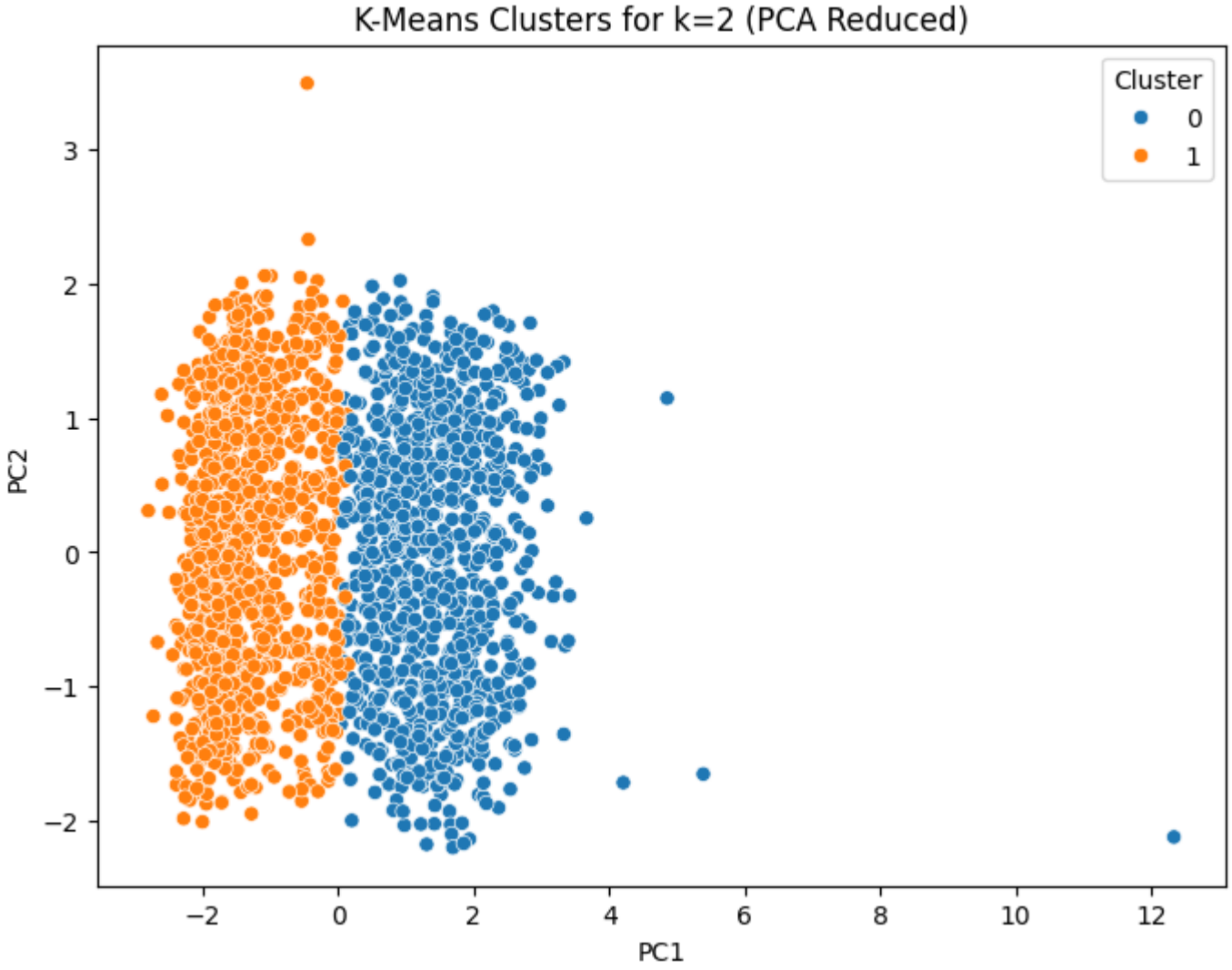


Acc to elbow method, k=5 has the steep but acc to silhoutte k=2 has the highest score. We give more importance to silhoutte score and hence k=2.



K=5 is not a good choice clearly.

Total spending varies variedly between both the clusters and i believe is the main factor concerning the clusters here.



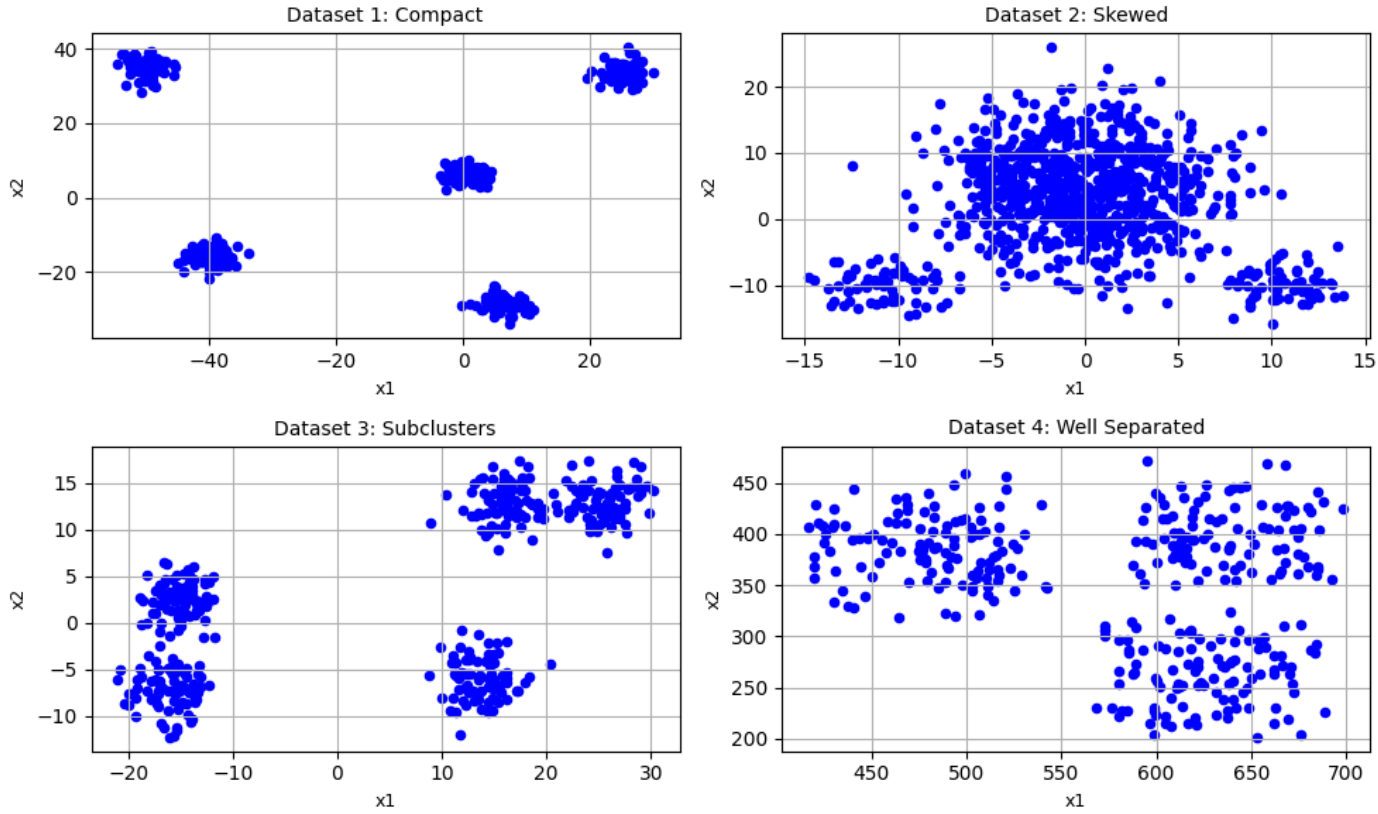
PCA Analysis:

	PC1	PC2
Age	0.181597	0.305290
Income	0.541693	-0.045111
Recency	0.012213	0.949750
Total_spending	0.590702	-0.041899
Total_Num_Columns	0.569661	-0.031340

PC1 is dominated by total_spending feature and PC2 is dominated by Recency.

TASK-3: CLUSTERING WITH DIFFERENT ALGORITHMS

VISUALIZATION USING SCATTER PLOTS



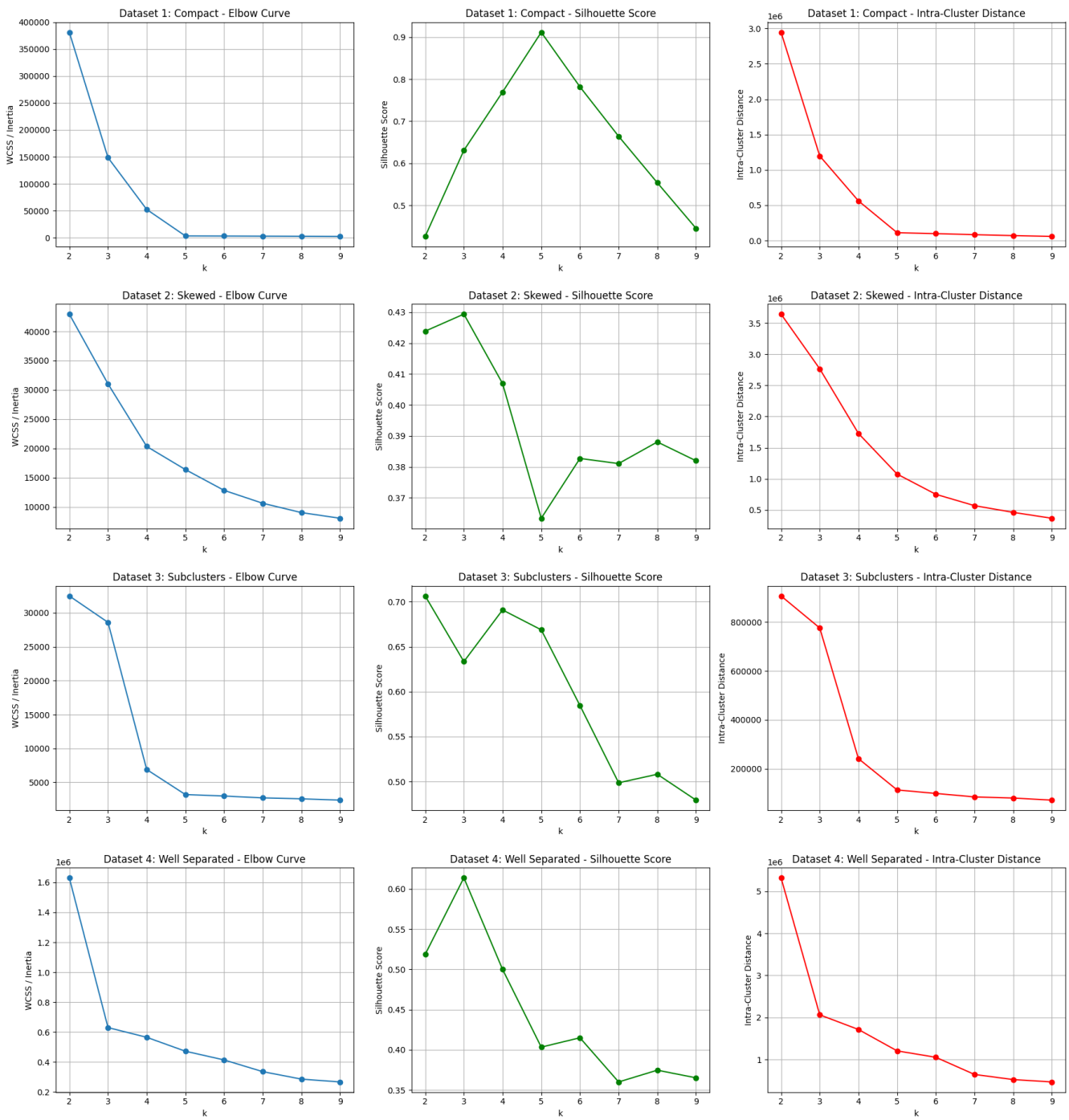
Dataset1: 5 well separated clusters (k=5 can be seen)
Dataset 2: Data is highly skewed and dense in centre.
Dataset 3: Some clusters are close and may merge.
Dataset 4: 3 large clusters, some points being close to other clusters as well.

Elbow, Silhoutte score, and Intra cluster score for each graph:

Elbow score: Sharp drop means clusters are becoming much better. Flattening curve means adding more clusters does not improve much further.

Silhoutte score: On a 0->1 scale, closer to 0 means very well separated clusters and <0.5 means not great separation.

Intra cluster distance: Lower the better.



From the above graphs:

Dataset 1: Compact

- Elbow at **k = 5**
- Silhouette highest at **k = 5**
- Intra-cluster distance flattens at **k = 5**

Hence, Best k = **5**

Dataset 2: Skewed

- Silhouette peaks around **k = 3**

- Elbow bend also around **k = 4**
- Intra-cluster distance flattening begins near **k = 4-6**

Best k = **3**

Dataset 3: Subclusters

- Elbow bends around **k = 4**
- Silhouette peaks around **k = 2 or k=4**
- Intra-cluster distance flattening begins at **k=5**

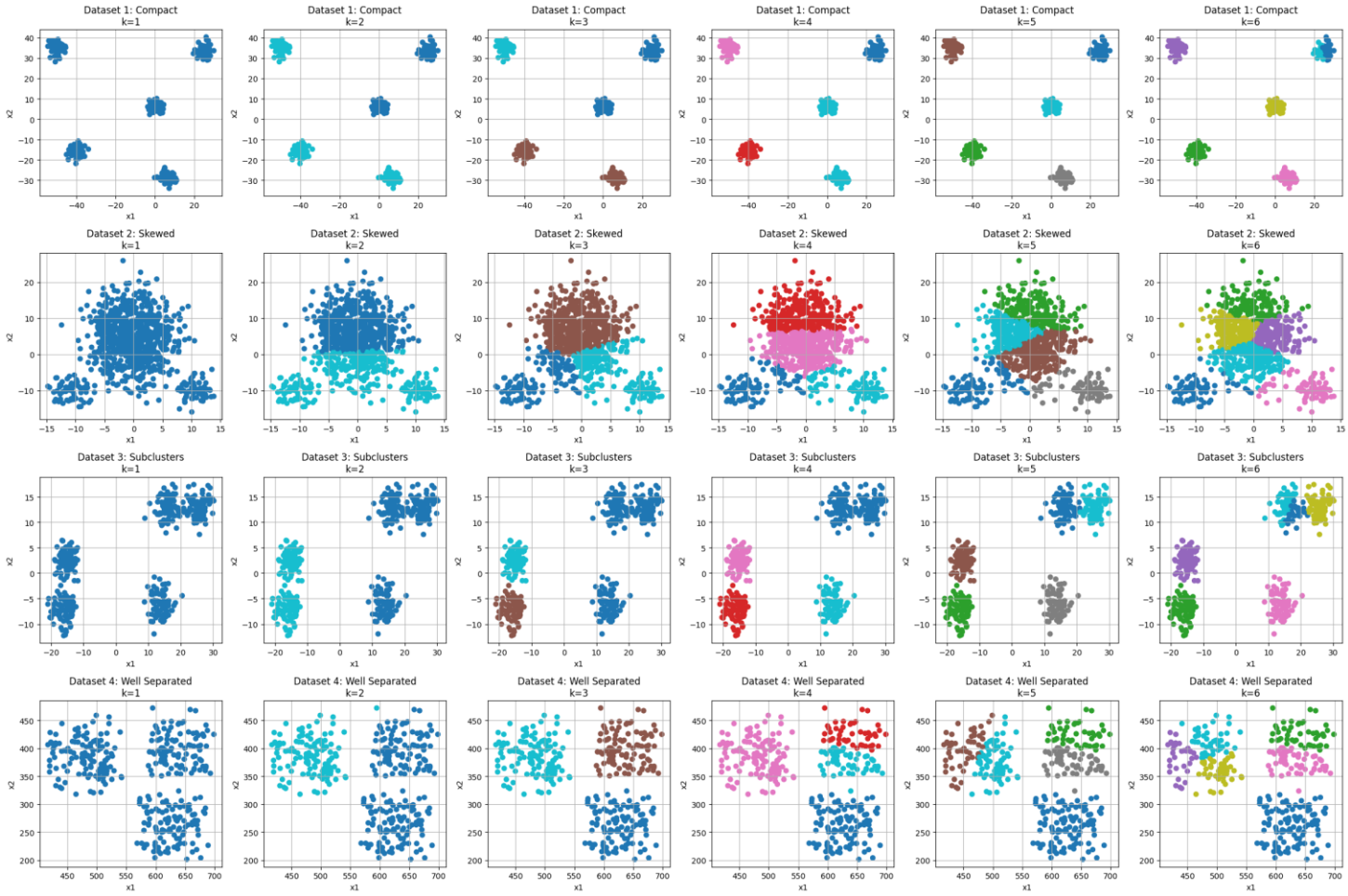
Best k = **4**

Dataset 4: Well-Separated

- Elbow bend around **k = 4**
- Silhouette highest around **k = 3**, but **4 also high**
- Intra-cluster distance flattening at 3

Best k = 3

APPLYING K MEANS



Dataset1: k<5 merges clusters. **K=5** captures all natural clusters perfectly.

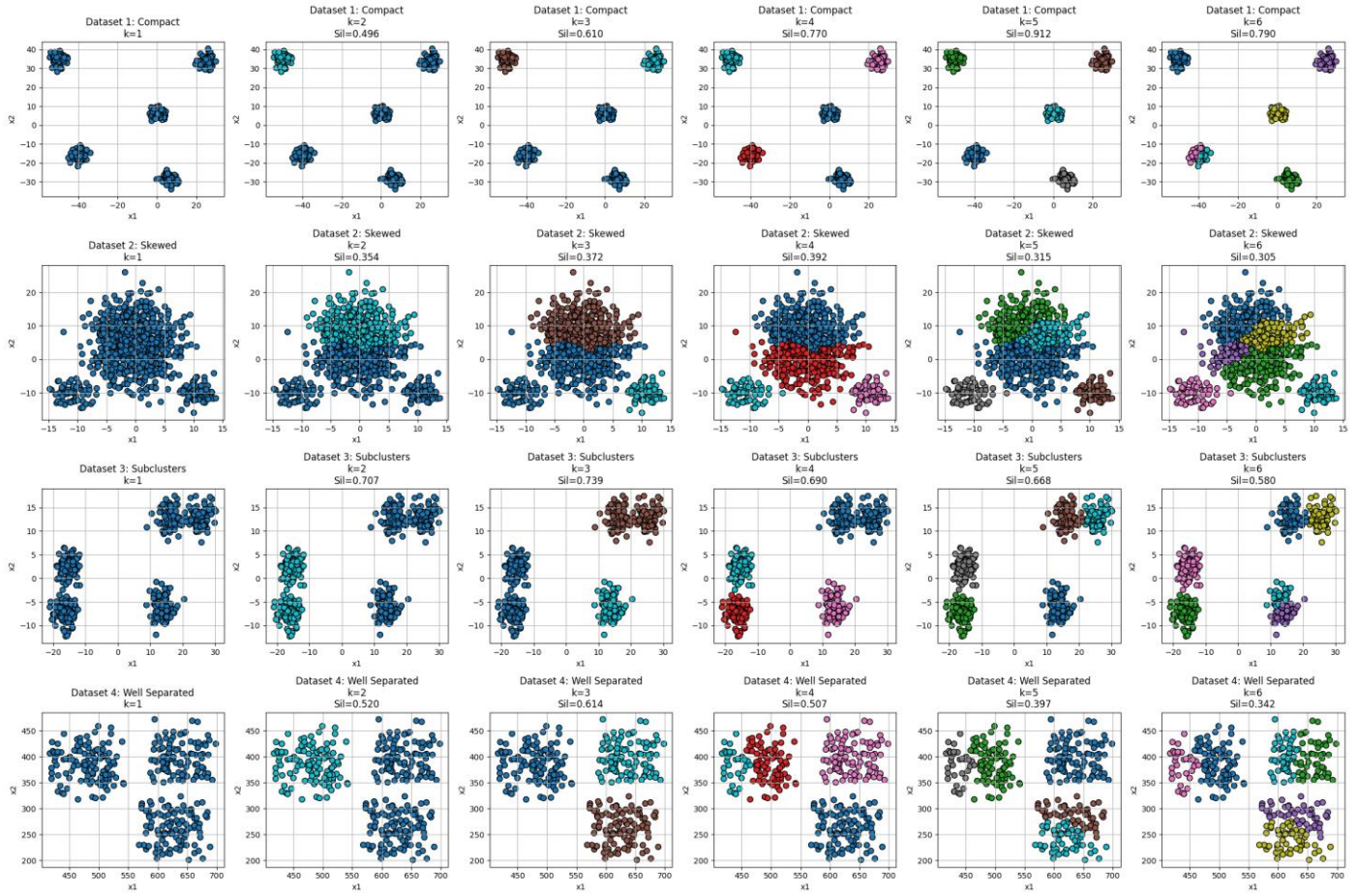
Dataset2: Data has a dense centre and small side clusters. K=3 keeps some clusters merged. K=4 or 5 starts separating the main skewed cluster to meaningful subclusters. **K=4**

Dataset3: **k=3** (silhouette score best) shows two opposing side clusters. K=2 or k=4 can be a good choice as well.

Dataset4: Widely separated points inside each cluster. **k=3** captures the distinction best.

```
Agglomerative Silhouette Scores:
Dataset 1: Compact: ['NA', '0.496', '0.610', '0.770', '0.912', '0.790']
Dataset 2: Skewed: ['NA', '0.354', '0.372', '0.392', '0.315', '0.305']
Dataset 3: Subclusters: ['NA', '0.707', '0.739', '0.690', '0.668', '0.580']
Dataset 4: Well Separated: ['NA', '0.520', '0.614', '0.507', '0.397', '0.342']
```

APPLYING AGGLOMERATIVE CLUSTERING




```
Agglomerative Silhouette Scores:
Dataset 1: Compact: ['NA', '0.496', '0.610', '0.770', '0.912', '0.790']
Dataset 2: Skewed: ['NA', '0.354', '0.372', '0.392', '0.315', '0.305']
Dataset 3: Subclusters: ['NA', '0.707', '0.739', '0.690', '0.668', '0.580']
Dataset 4: Well Separated: ['NA', '0.520', '0.614', '0.507', '0.397', '0.342']
```

Dataset1: k=5 (K means effective as well)

Dataset2: k=4 (Better than k means as it was considering outliers at the bottom of the densed cluster in below clusters increasing confusion and blurred boundary).

Dataset3: k=3 (Better from k means as two closeby clusters are merged)

Dataset4: k=3 (K means effective as well)

DBSCAN CLUSTERING:

Eps: Maximum distance between two points for them to be considered neighbours.

If eps too small, many points may be considered noise because they don't have enough neighbours.

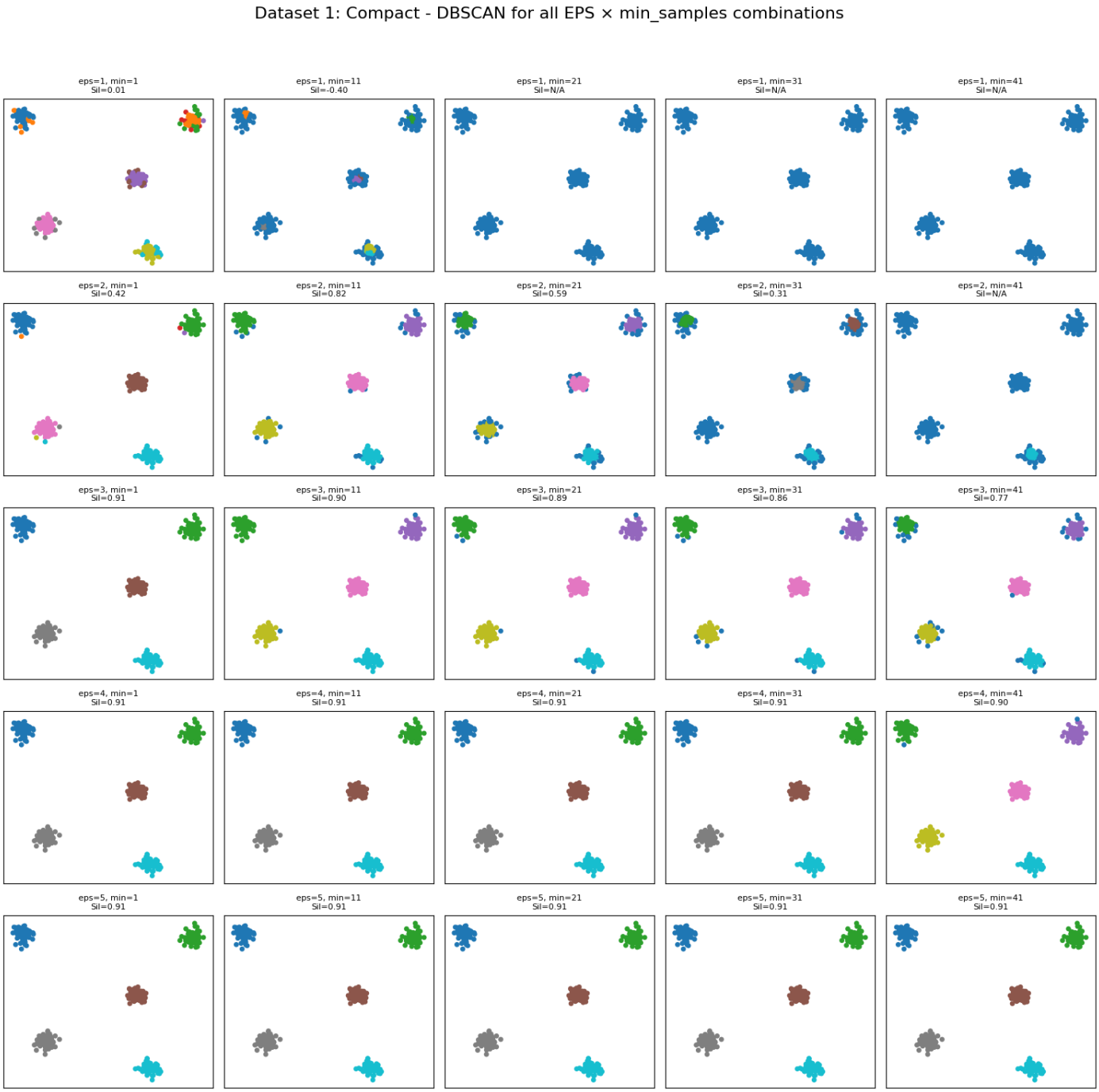
If eps too large, clusters may merge, resulting into fewer, bigger clusters.

Min_samples: Minimum number of points (including the point itself) required in a neighbourhood to form a dense region (a cluster).

If min_samples too small, even sparse areas can form clusters.

If min_samples too large, more noise points because small clusters wont satisfy the threshold.

Dataset 1:



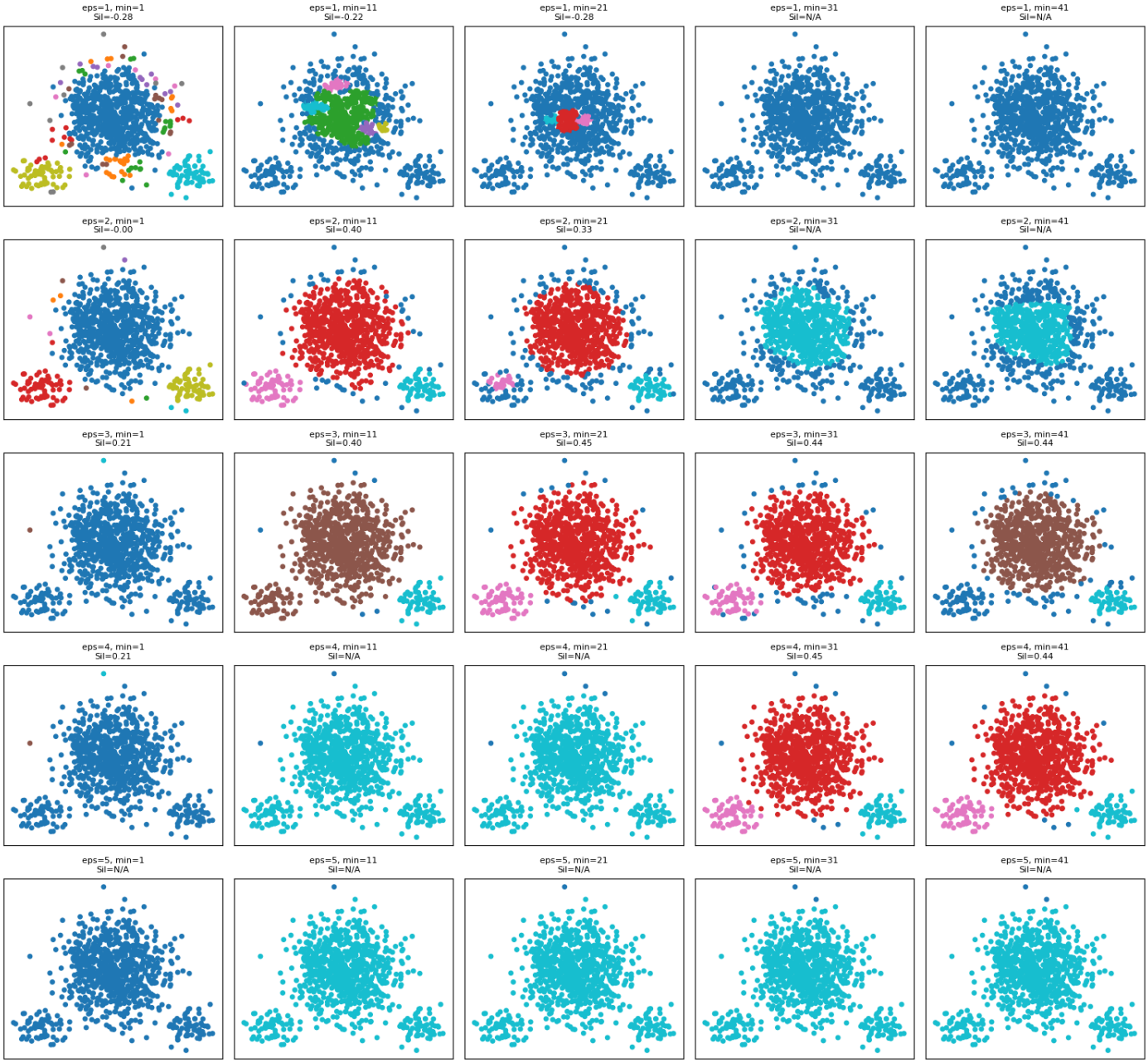
eps ↓ / min_samples →	1	11	21	31	41
1	Si≈0.01 – Tiny micro-clusters, very poor	N/A – All noise	N/A – No clusters	N/A – No clusters	N/A – No clusters
2	Si≈0.42 – Some splitting + noise	Si≈0.82 – Good clusters, small noise	Si≈0.59 – Noise↑ due to strict min_samples	Si≈0.31 – Too strict, many noise points	N/A – All noise
3	Si≈0.91 – Perfect 4 clusters	Si≈0.90 – Excellent, slight border noise	Si≈0.89 – Very good, small noise	Si≈0.86 – Noise↑, clusters thin	Si≈0.77 – Too strict, incomplete clusters
4	Si≈0.91 – Perfect, stable	Si≈0.91 – Almost perfect	Si≈0.91 – Still perfect	Si≈0.90 – Few noise points	Si≈0.90 – Slight noise but stable
5	Si≈0.91 – Perfect clusters	Si≈0.91 – Perfect	Si≈0.91 – Perfect	Si≈0.91 – Perfect	Si≈0.91 – Perfect

FINAL SUMMARY:

- **eps = 1 → too small, fails**
- **eps = 2 → only works okay at min_samples = 11**
- **eps = 3 → excellent, best with min_samples = 1**
- **eps = 4 & 5 → consistently best; silhouette ≈ 0.91 for almost all min_samples**

Dataset 2:

Dataset 2: Skewed - DBSCAN for all EPS × min_samples combinations



eps ↓ / min_samples →	1	11	21	31	41
1	Si=0.28 — Over-fragmented, many micro-clusters	Si=0.22 — Big cluster fragmented, small clusters partial	Si=0.28 — Slightly better but still too many tiny clusters	N/A — eps too small → all noise	N/A — all noise
2	Si=0.60 — Big cluster detected, small OK, some noise	Si=0.60 — Good separation, stable	**Si=0.61 — BEST**; best tradeoff, shapes preserved	Si=0.44 — Too strict → noise↑ → silhouette drops	N/A — no clusters
3	Si=0.21 — Over-merging; big blob swallows small clusters	Si=0.40 — Some structure recovered	Si=0.45 — Better but still mixed edges	Si=0.34 — Too strict → noise↑	Si=0.44 — Slight recovery
4	Si=0.21 — Severe merging → almost one cluster	N/A — mostly noise	Si=0.25 — Very poor separation, blob dominates	Si=0.45 — Still merged heavily	Si=0.40 — Over-merged
5	N/A — everything merged into one → silhouette undefined	N/A — unusable	N/A — unusable	N/A — unusable	N/A — unusable

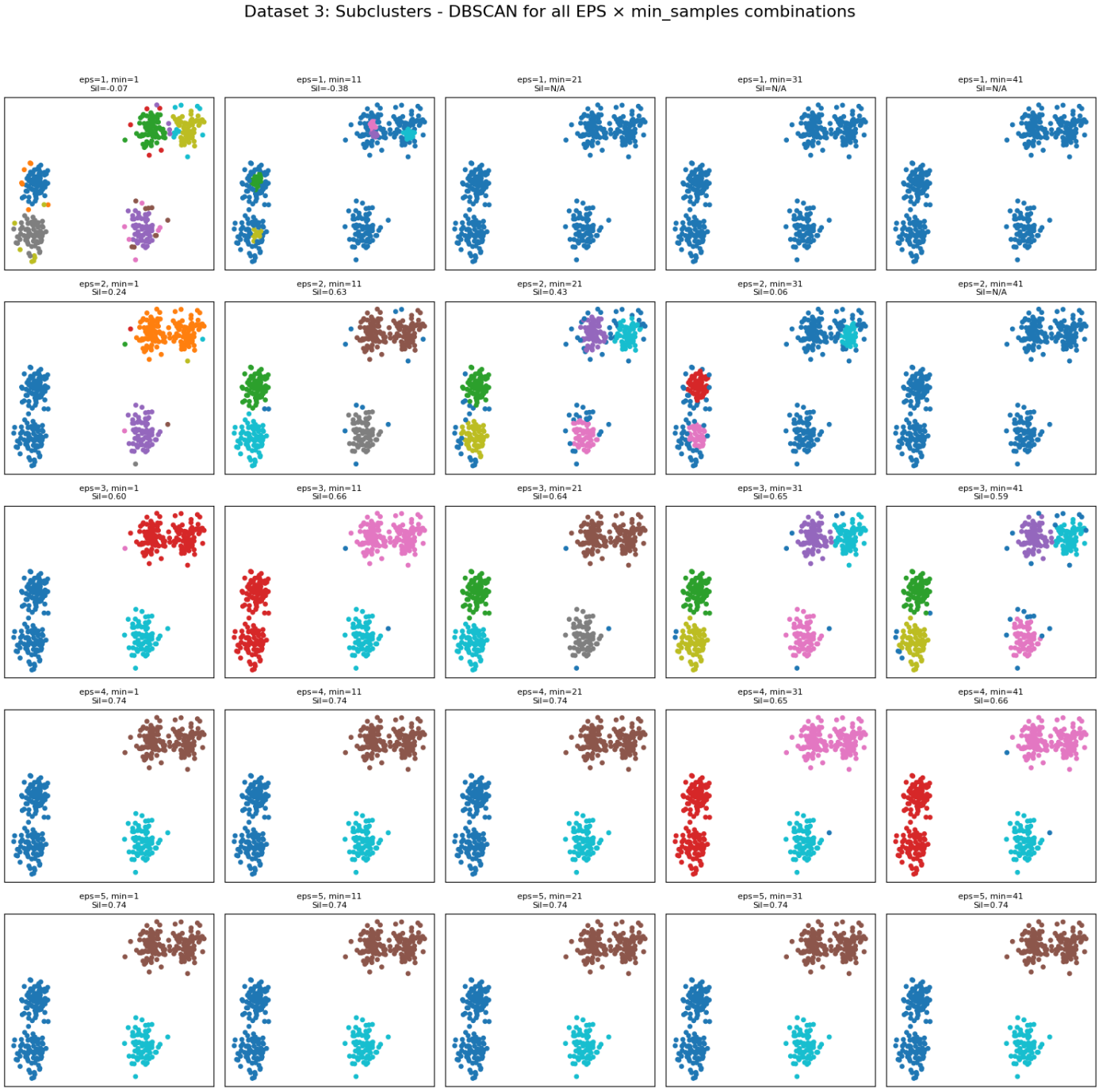
BEST COMBINATION FOR THIS DATASET:

- eps = 2, min_samples = 21

Why:

- Good separation of big & small clusters
- Minimal noise
- No merging
- Highest silhouette (~0.61)

DATASET3:



eps ↓ / min_samples →	1	11	21	31	41
1	Si=0.07 — Heavy fragmentation, many micro-clusters	Si=0.38 — Still fragmented; many small clusters	N/A — too strict → mostly noise	N/A — mostly noise	N/A — mostly noise
2	Si=0.24 — Too many subclusters	Si=0.58 — Subclusters begin merging	**Si=0.66 — Best; correct clusters formed**	Si=0.45 — Too strict → noise↑	N/A — no clusters
3	Si=0.60 — Very good; subclusters almost fully merged	**Si=0.66 — One of the best; clean 2 clusters**	Si=0.62 — Slight noise	Si=0.55 — Too strict → noise↑	Si=0.59 — Some fragmentation
4	**Si=0.74 — BEST; perfect 2 clusters**	**Si=0.74 — Same perfect result**	Si=0.71 — Very good, small noise	Si=0.55 — Boundary breaks	Si=0.66 — Better but not best
5	Si=0.74 — Excellent	Si=0.74 — Excellent	Si=0.74 — Excellent	Si=0.74 — Excellent	Si=0.74 — Excellent

BEST COMBINATION FOR DATASET 3:

eps = 4 & min_samples = 1 or 11

(or)

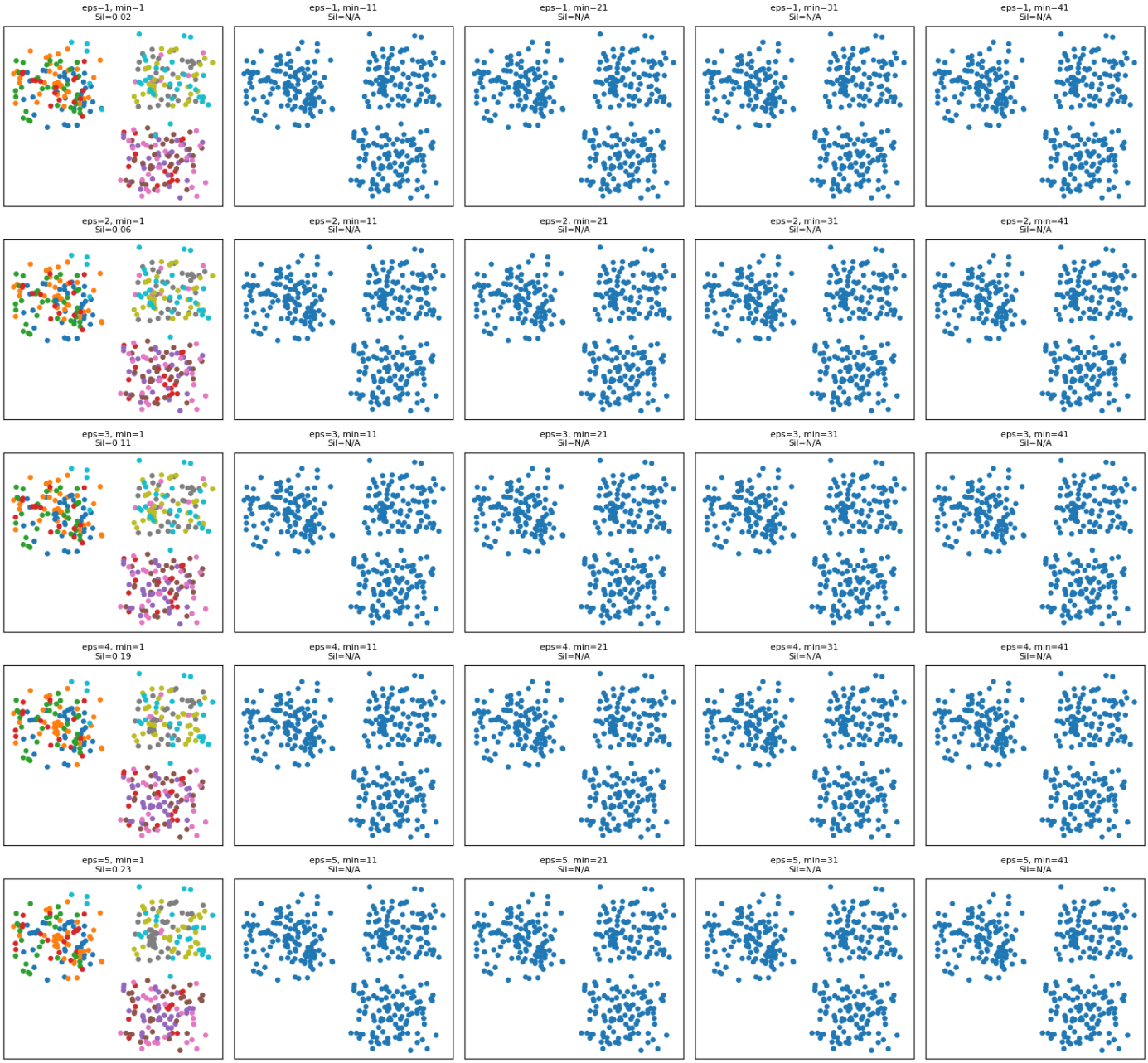
eps = 5 with any min_samples

Why:

- Subclusters merge correctly
- Only 2 clusters found (true structure)
- Highest silhouette (~0.74)
- Very stable
- No noise or over-merging

DATASET4:

Dataset 4: Well Separated - DBSCAN for all EPS × min_samples combinations



eps ↓ / min_samples →	1	11	21	31	41
1	Si=0.22 — Heavy fragmentation; many tiny clusters	N/A — too sparse → all noise	N/A — all noise	N/A — all noise	N/A — all noise
2	Si=0.06 — Extreme fragmentation	N/A — no clusters	N/A — no clusters	N/A — no clusters	N/A — no clusters
3	Si=0.11 — Slightly better but still fragmented	N/A — not enough neighbors	N/A	N/A	N/A
4	Si=0.19 — Some merges but still many mini-clusters	N/A — insufficient density	N/A	N/A	N/A
5	**Si=0.23 — BEST; correct 2 clusters detected**	N/A — cluster too sparse for min ≥ 11	N/A	N/A	N/A

- **eps = 1 → 4** all fail (too small, too sparse → everything becomes noise)
- **eps = 5, min_samples = 1** → **ONLY effective setting**, recovers correct 2-cluster structure
- Larger min_samples fail because the dataset is **very sparse**, so points do not have enough neighbors even with eps=5.

BEST COMBINATION FOR DATASET 4:

eps = 5, min_samples = 1

Why?

- Only combination that connects sparse cluster points
- Recovers the true 2 cluster structure
- Highest silhouette (0.23, and all other valid ones are lower)

FINAL ANALYSIS:

DATASET 1:

Best: K-Means OR Agglomerative (Ward / Complete)

Why:

- Clusters are **spherical** and **well-separated**.
- K-Means performs very well because its assumptions match the dataset.
- Agglomerative with Ward/complete works equally well.

DBSCAN:

- Also works but gives no advantage over simpler methods.

DATASET 2: Non-spherical / Crescent / Moons (Non-Convex Shapes)

Best: DBSCAN

Why:

- DBSCAN can model **arbitrary shapes**, like curves, spirals, moons.
- It groups points based on density, so shape does not matter.

K-Means:

- **Fails** — forces circular clusters → wrong boundaries.

Agglomerative:

- **Single linkage** can work somewhat (captures chaining),

- but DBSCAN is **more robust** and cleaner.

DATASET 3: Overlapping Clusters + Noise

Best: Agglomerative

Why:

- Handles overlapping clusters better by merging hierarchical structure.
- Less sensitive to outliers compared to K-Means.
- Linkage choice (complete/average) gives stable separation.

K-Means:

- Struggles when cluster boundaries overlap → misassignments.

DBSCAN:

- Often **fails**:
  - Too much noise → marks many points as "-1"
  - Or breaks one cluster into many small ones
- Cannot correctly separate clusters with similar density.

DATASET 4: (Well-Separated but Sparse Clusters)

Best: K-Means OR Agglomerative

Why:

- Clusters are well-separated and roughly spherical.
- Very easy for centroid-based or hierarchical methods.

DBSCAN:

- **Fails for most (eps, min_samples) combinations**:
  - Distances between clusters are large
  - Points are sparse → DBSCAN marks **almost everything as noise** unless eps is extremely large
- Only very extreme eps settings work, but that merges clusters incorrectly.