Data Preparation and Analysis Module 4

Partitioning, Segmenting, and Clustering of Observations

Module 4 Lesson Plan

1. **Lesson 1: Partition Observations for Training Models**
   - Understand the Need to Objectively Validate Models with a Testing Partition
   - Perform Simple Random Sampling, with Optional Stratification Variables
2. **Lesson 2: Create Segments of Observations for Business Reasons**
   - Identify the Most Valuable Customers for a Retail Business
   - Perform the Recency, Frequency, and Monetary (RFM) Analysis
3. **Lesson 3: Put Observations with Similar Feature Values in Clusters**
   - Apply the K-Means and the K-Modes Clustering Algorithms
   - Describe Profiles of Clusters

Lesson 1:

Partition Observations for Training Models

Lesson 1:

Partition Observations for Training Models

What are Objective Results?

- We want to apply our model to different data, in a different environment, at a future time, by someone else, and still generate values and make a real impact.
- Although we cannot claim that our model will work in all scenarios, foreseeable or not, it is our responsibility to evaluate how well our model can perform in other circumstances.
- Our evaluations are based on two criteria, namely, Reproducibility and Replicability.

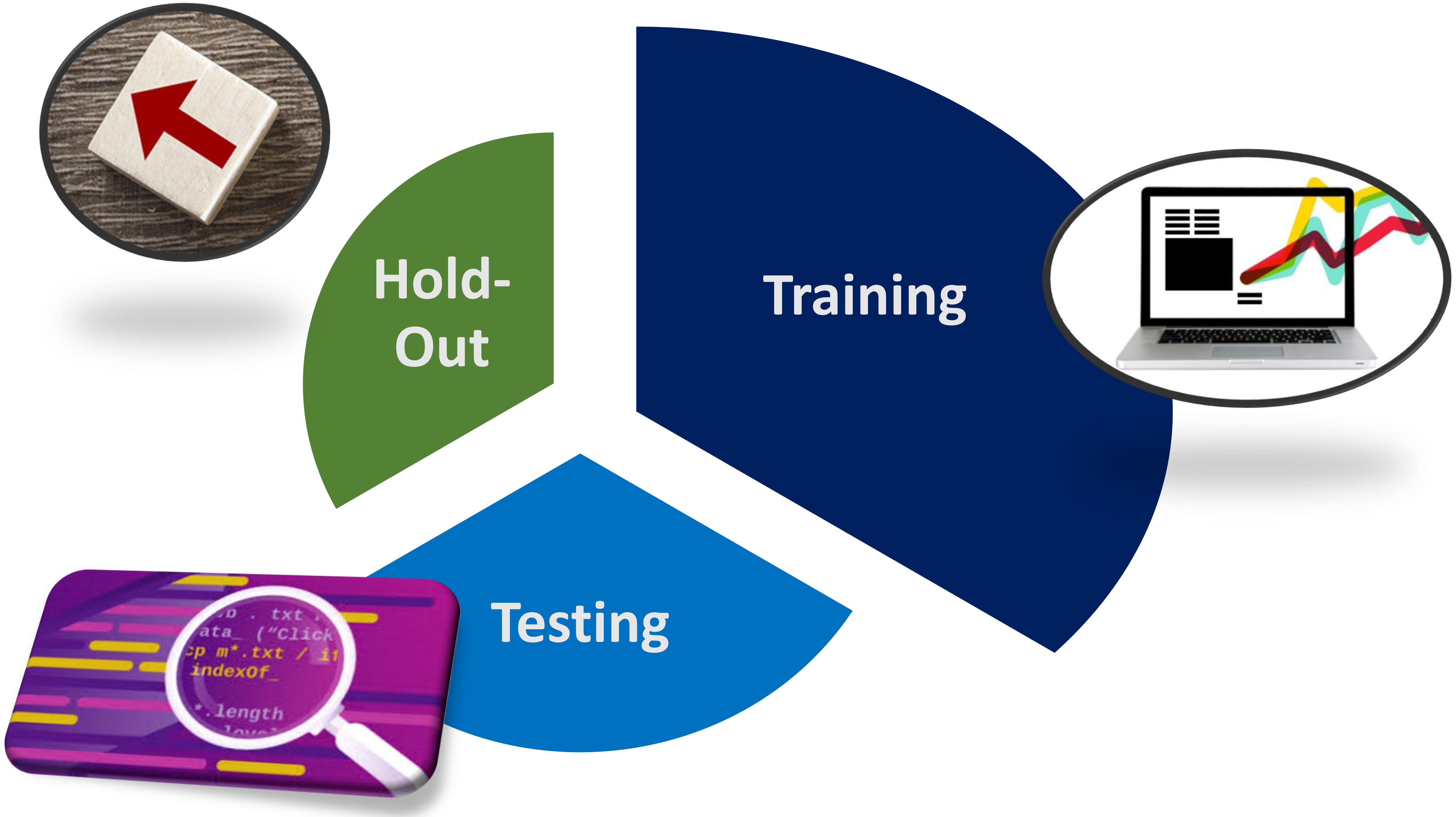
1. Start from the original data
2. Use the same algorithms
3. Execute the same tasks
4. Run on same or compatible machine
5. Reproduce the same results and conclusions

- If the expected results cannot be reproduced, this indicates there are some unexplained (intentional or random) interactions among the data, the algorithm, and the machine.
- Common causes are:
    1. Uninitialized variables in the codes
    2. Misunderstood documentation of the activity

1. Use a different data in same business context
2. Use the same algorithms
3. Execute the same tasks
4. Run on a compatible machine
5. Lead to the similar conclusions

- If the expected conclusions cannot be replicated, this indicates design flaws in the tasks.
- Common causes are:
    1. Correlations among the features not accounted for
    2. Degenerated data not handled properly
    3. Algorithms only work in specialized scenarios (e.g., no missing values).
    4. Software issues (e.g., need a particular hotfix)

A common practice is to separate the original observations into two, or occasionally three, partitions.



Training

- Prepare Data
- Ascertain model structure
- Determine model parameters

Testing

- Assess model performance
- Compare performance among models
- Play the role of a “future data”

Hold-Out

- Obtain an objective estimate of the error size
- Play the role of a “third party” data
- Often used in data challenge

Unless we must report the model performance on a third-party benchmark dataset, we typically do not need the Hold-Out partition

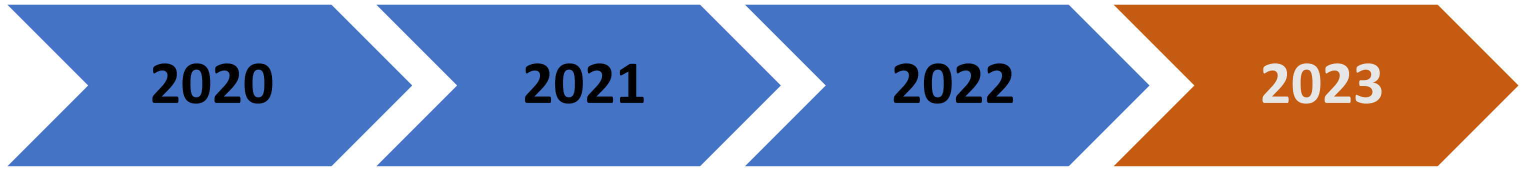
Large or Good Data

Typical Allocation

Small or Noisy Data

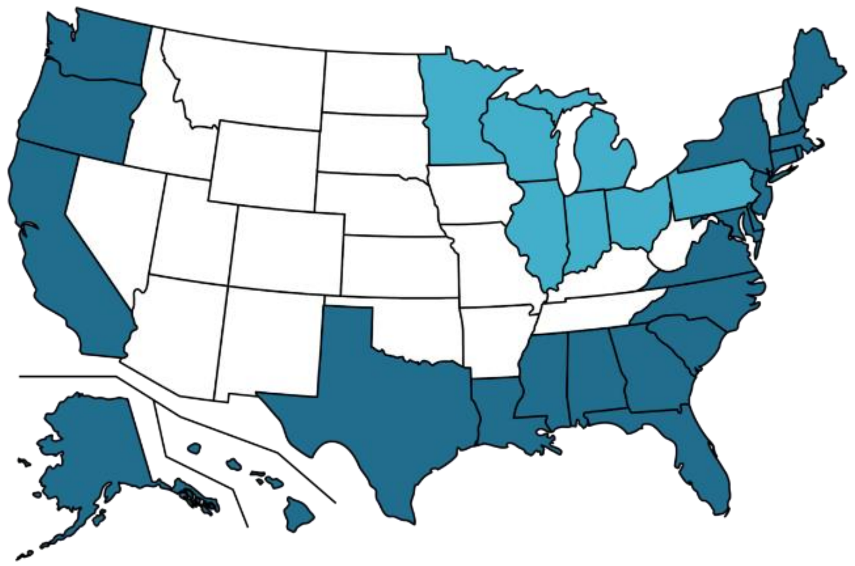
By Chronological Variables

- Train a customer sentiment model using data from 2020 to 2022
- Then apply the model to the "future" data of 2023



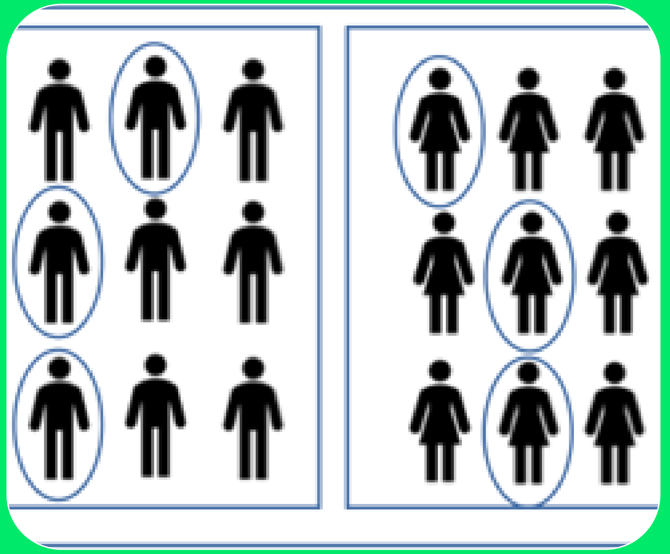
By Geographical Variables

- Train a marketing campaign model based on the U.S. coastal states
- Then apply the model to the U.S. inland states



Simple Random Sampling (SRS)

- Select a proportion of observations without replacement



Stratified Random Sampling

- Perform simple random sampling in each stratum
- Select the same proportion of observations from each stratum

- If k observations are to be selected from a pool of N observations, then any sample of k of observations will have the same probability of being selected.
- Observations are sampled without replacement (i.e., once an observation is selected, it cannot be selected again).
- C. T. Fan, Mervin E. Muller and Ivan Rezucha (1962), "Development of Sampling Plans by Using Sequential (Item by Item) Selection Techniques and Digital Computers", *Journal of the American Statistical Association, Volume 57, Number 298, pages 387-402.*

- The strata must be disjoint groups.
- Each distinct value combination of the stratification variables forms a stratum.
- Perform the simple random sampling within each stratum.
- Select the same proportion of observations from each stratum.
- Collect the sub-samples from each stratum to form the final deliverable.

- **Function**: sklearn.model_selection.train_test_split (*arrays, **options)
- **Description**: Allocate rows of a data frame into the random training and testing subsets. The indices of the original data frame are carried over to the subsets.
- **Reference**: sklearn.model_selection.train_test_split.html

Specify random seed to 60616 so we can reproduce the results.

```
import pandas
from sklearn.model_selection import train_test_split

hmeq = pandas.read_csv('hmeq.csv')
hmeq_train, hmeq_test = train_test_split(hmeq, train_size = 0.7, random_state = 60616)
```

The hmeq.csv has 5,960 observations. Training partition has 5,960 × 70% = 4,172 observations. Testing partition has 5,960 × 30% = 1,788 observations.

Assign 70% of the observations to the Training partition, and the remaining 30% to the Testing partition

- Suppose BAD is our label variable. The distribution of the categories of BAD in the training partition is different from that in the testing partition.
- The model may favor either category as it tried to fit the training partition well.
- As a result, the model may perform poorer in the testing partition.

```
print(hmeq_train['BAD'].value_counts(normalize = True))
BAD
0   0.7955417066
1   0.2044582934

print(hmeq_test['BAD'].value_counts(normalize = True))
BAD
0   0.8120805369
1   0.1879194631
```

```
import pandas
from sklearn.model_selection import train_test_split

hmeq = pandas.read_csv('hmeq.csv')
hmeq_train, hmeq_test = train_test_split(hmeq, stratify = hmeq['BAD'], train_size = 0.7,
                                         random_state = 60616)
```

Each category of BAD forms a stratum.

```
print(hmeq_train['BAD'].value_counts(normalize = True))
BAD
0    0.8005752637
1    0.1994247363
print(hmeq_test['BAD'].value_counts(normalize = True))
BAD
0    0.8003355705
1    0.1996644295
```

Notice the proportions?

- Suppose the original data has n observations where n_0 of them are in the category BAD = 0 and another n_1 observations are in the category BAD = 1.
- Let $0 < p < 1$ is the sampling proportion and BAD is the stratification variable.
- The Training partition consists of $n_0 p$ observations with category BAD = 0 and $n_1 p$ observations with category BAD = 1.
- The Testing partition consists of $n_0(1-p)$ observations with category BAD = 0 and $n_1(1-p)$ observations with category BAD = 1.

- The proportion of category BAD = 0 in the Training and Testing partitions is

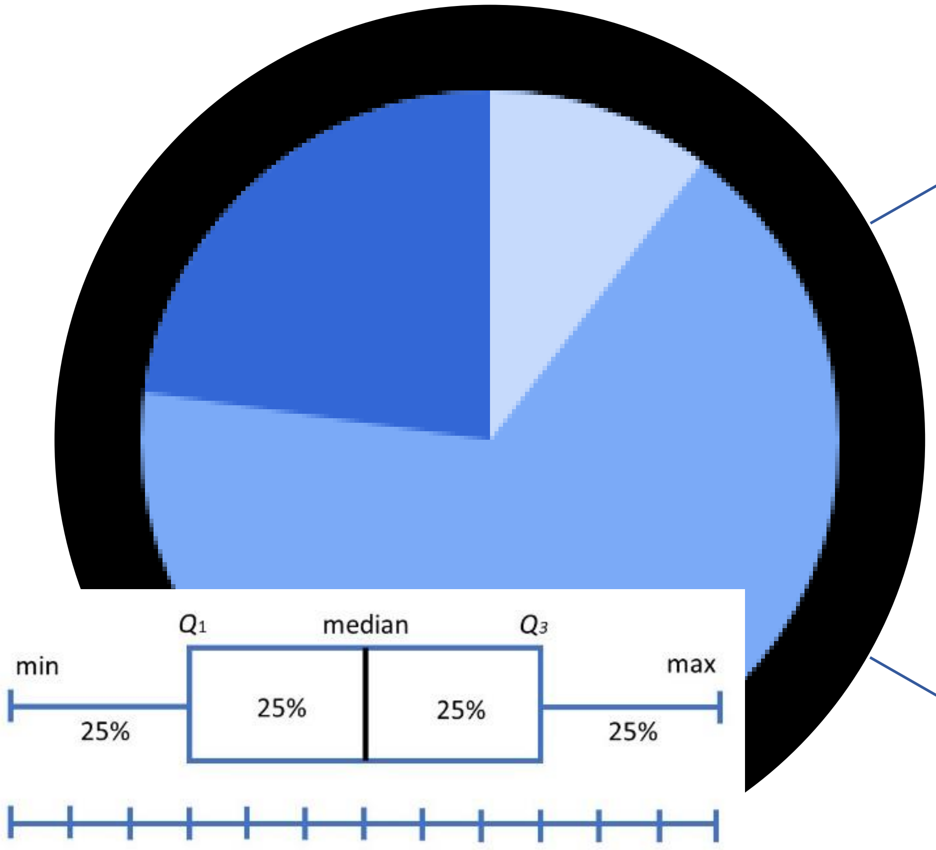
$$\frac{n_0 p}{n_0 p + n_1 p} = \frac{n_0(1-p)}{n_0(1-p) + n_1(1-p)} = \frac{n_0}{n_0 + n_1}$$ which is identical to that in the original data.

- The proportion of category BAD = 1 in the Training and Testing partitions is

$$\frac{n_1 p}{n_0 p + n_1 p} = \frac{n_1(1-p)}{n_0(1-p) + n_1(1-p)} = \frac{n_1}{n_0 + n_1}$$ which is identical to that in the original data.

- Therefore, stratified random sampling maintain the distribution of the label variable the same across both partitions. As a result, we can fairly evaluate the performance of a model in both partitions.

Measurement Level of Label Variable?



Categorical

- Recommend Stratified Random Sampling
- Strata defined by label categories
- Label distribution maintained across partitions
- Watch out some rare label categories

Continuous

- Recommend Simple Random Sampling
- Check if target distributions in partitions are similar afterward

Lesson 2:

Create Segments of Observations for Business Reasons

Return Customers Bring Business

- Retaining an existing customer costs less than acquiring a potential customer (the former has previously generated revenue for you).
- Raising customer retention by a small percentage can bring a considerable profit increase (as costs are mostly fixed).
- The success rate of selling to a customer you already have is always higher than that of selling to a new customer.
- Word-of-mouth from a long-time customer is more effective than a random advertisement.

What Customers Should We Retain?

Come Back Soon

- Customers who purchase something recently based on their business needs. Yesterday or last year.

Come Often

- Customers who purchase very often according to their business cycle. Once per day, or once per year.

Spend A Lot

- Customers who spend a lot of money on many and/or high-end items.

Recency-Frequency-Monetary (RFM) Analysis

- The RFM analysis places customers into tiers of RFM values based on the customers’ transaction history.
- The RFM values reflect the level of customer loyalty to the business.
- The RFM values help businesses to prioritize their resources and attention to existing customers.
- The RFM Analysis originated from Direct Marketing. Marketers believe customers with higher RFM values will be more likely to respond to a new campaign or product offer.

Transaction History for RFM Analysis

Information on each transaction:

1. Customer Identifier (*CustomerID*)
2. Data, Time, or Sequence (*Date*)
3. Transaction Amount (*Amount*)
4. Item Identifiers (*ProductLine* and *ProductNumebr*) are optional.

CustomerID	ProductLine	ProductNumber	Date	Amount
300	B-200	228	1/1/2021	40
347	A-100	171	1/1/2021	36
373	E-500	571	1/1/2021	169
489	E-500	592	1/1/2021	182
507	D-400	438	1/1/2021	142
50	D-400	493	1/2/2021	119
180	D-400	460	1/2/2021	104
204	D-400	469	1/2/2021	149
665	C-300	316	1/2/2021	82
753	B-200	275	1/2/2021	41
810	C-300	324	1/2/2021	87
885	D-400	411	1/2/2021	117
895	B-200	220	1/2/2021	37
297	E-500	592	1/3/2021	156
340	A-100	112	1/3/2021	45

Transaction With Negative or Zero Amount

- The business must decide whether to include transactions with non-positive amounts in the RFM analysis.
- Negative amounts usually indicate merchandise returned by customers.
- Zero amounts suggest complimentary items for customers.
- Including these transactions may adversely affect the RFM analysis. On the other hand, these transactions reflect actual interaction with customers.

Transaction Amount Due To One Customer

- From the business perspective, determine the monetary value of each transaction.
- Roll up the observations to the customer level and calculate the total monetary value of each customer.

Monetary Score

1. For each customer, calculate Monetary as the total value of transactions.
2. Determine the five quintiles of Monetary and divide Monetary into five equal groups observations. Ideally, each group contains 20% of observations.
3. The first quintile group (i.e., the lowest 20%) has a Monetary Score of 1, the second group has 2, the third group has 3, and the fourth group has 4. Finally, the fifth quintile group (i.e., the highest 20%) has a Monetary Score of 5.

Reference For Recency

- We need a reference date for calculating the Recency. This reference date must come before the earliest date in the transaction history.
- Suppose the earliest date in our transaction history is January 1, 2023. To make this date as Day 1 for our RFM analysis, we will specify the reference date as December 31, 2022.
- If the transactions are stamped with another chronological unit (e.g., time), we will apply this concept to specify our reference for recency.

Recency Score

1. Calculate the number of chronological units since our reference.
2. Roll up the observations to the customer level and calculate Recency as the largest number of chronological units for each customer.
3. Determine the five quintiles of Recency and divide Recency into five equal groups of observations. Ideally, each group contains 20% of observations.
4. The first quintile group (i.e., the lowest 20%) has a Recency Score of 1, the second group 2, the third group has 3, and the fourth group has 4. Finally, the fifth quintile group (i.e., the highest 20%) has a Recency Score of 5.

Frequency Score

1. For each customer, calculate Frequency as the number of transactions that are associated with the customer.
2. Determine the five quintiles of Frequency and divide Frequency into five equal groups of observations. Ideally, each group contains 20% of observations.
3. The first quintile group (i.e., the lowest 20%) has a Frequency Score of 1, the second group 2, the third group has 3, and the fourth group has 4. Finally, the fifth quintile group (i.e., the highest 20%) has a Frequency Score of 5.

RFM Score

- A customer's RFM Score is a three-digit integer.
- The hundred position is the Recency Score, the ten position is the Frequency Score, and the unit position is the Monetary Score.
- Mathematically, the RFM Score is 100 × Recency Score + 10 × Frequency Score + Monetary Score.
- The highest RFM Score is 555 and the lowest RFM Score is 111.

Customer Loyalty Tiers

- In theory, we can find 5×5×5=125 tiers of customers with various loyalty levels.
- The most loyal customers ideally have RFM scores of 555. The business should retain them and build strong customer relationships with them.
- Customers on the verge of churning usually have some 1s in their RFM scores. They interact with the business less often, spend little, and/or haven’t visited the business for a very long time. The business should reach out to these customers to listen to their concerns.

Customer Transactions in 2021

Module 4 RFM Analysis.py

Customer Transactions in 2021

- There were 4,907 transactions in 2021 from 995 customers.
- All transactions occurred between January 1 and December 31 of 2021 inclusively. So, we chose our reference date as December 31, 2020.
- There are no transactions with zero or negative amounts.

Quintiles of Recency, Frequency, and Monetary

Statistic	Recency	Frequency	Monetary
Count	995	995	995
Minimum	30	1	12
20%	**228**	**3**	**254.8**
40%	**269**	**4**	**381.0**
60%	**302**	**5**	**505.4**
80%	**336**	**7**	**665.0**
Maximum	365	14	1488

Decision Rules For Assigning Scores

```
if (Recency <= 228):
   Recency_Score = 1
elif (Recency <= 269):
   Recency_Score = 2
elif (Recency <= 302):
   Recency_Score = 3
elif (Recency <= 336):
   Recency_Score = 4
else:
   Recency_Score = 5
```

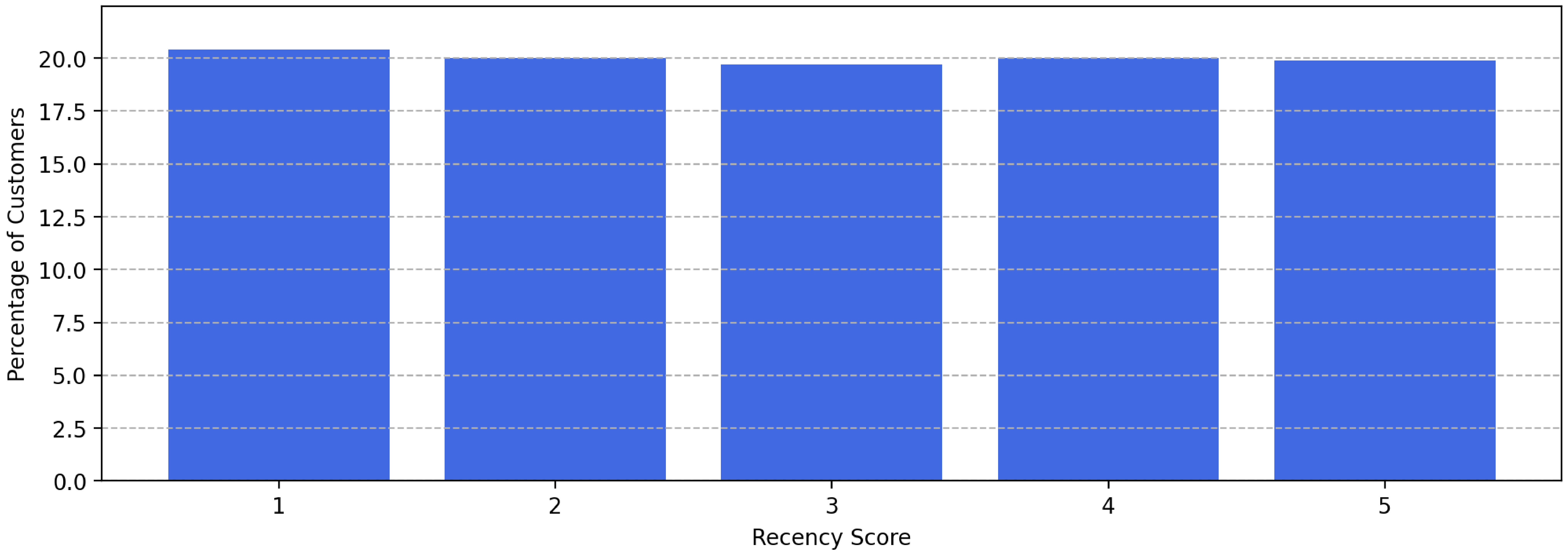
```
if (Frequency <= 3):
   Frequency_Score = 1
elif (Frequency <= 4):
   Frequency_Score = 2
elif (Frequency <= 5):
   Frequency_Score = 3
elif (Frequency <= 7):
   Frequency_Score = 4
else:
   Frequency_Score = 5
```

```
if (Monetary <= 254.8):
   Monetary_Score = 1
elif (Monetary <= 381):
   Monetary_Score = 2
elif (Monetary <= 505.4):
   Monetary_Score = 3
elif (Monetary <= 665):
   Monetary_Score = 4
else:
   Monetary_Score = 5
```

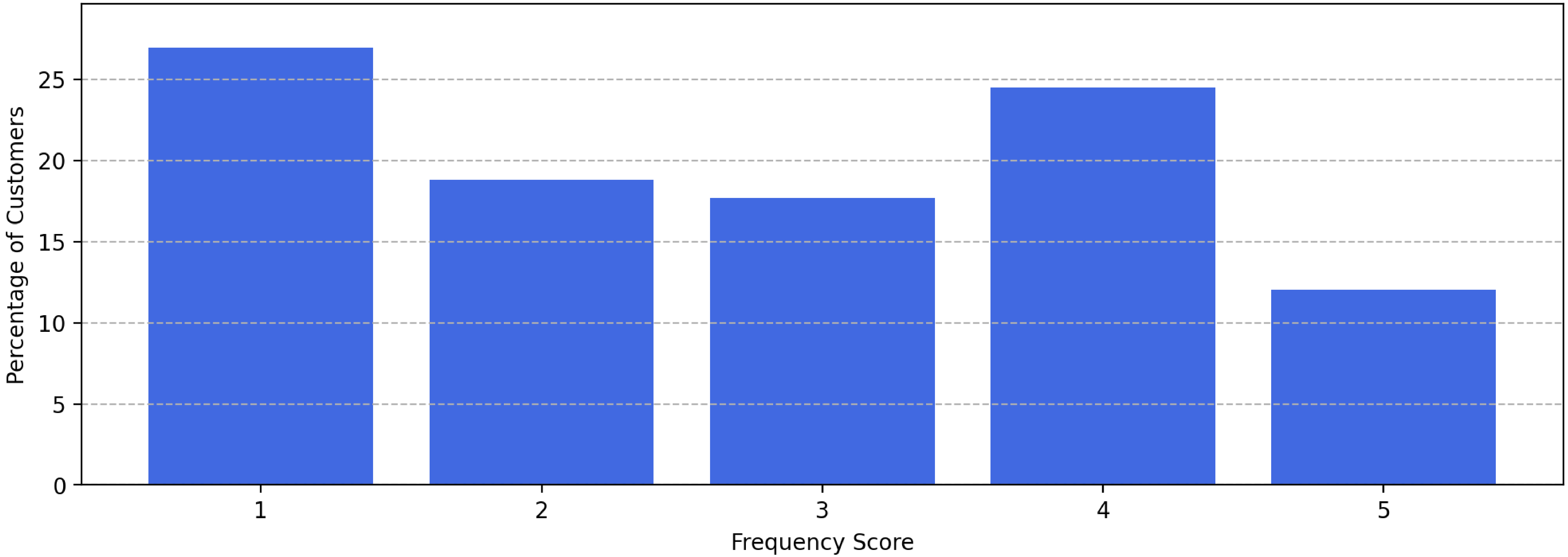
Inspect Group Assignments

- Do the groups have an equal number of observations?
- Are there any groups severely under-represented?
- What is the average monetary value in each Recency-Frequency group?

Distribution of Recency Score



Distribution of Frequency Score

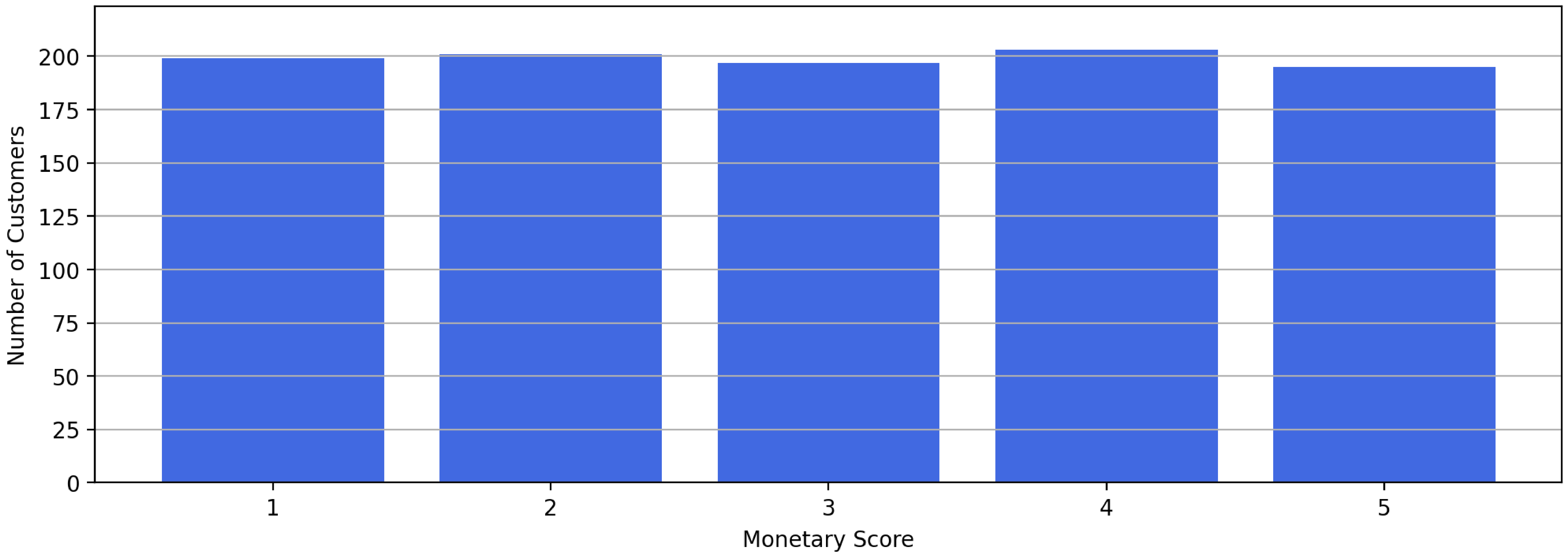


- Customers shop with very irregular frequencies!

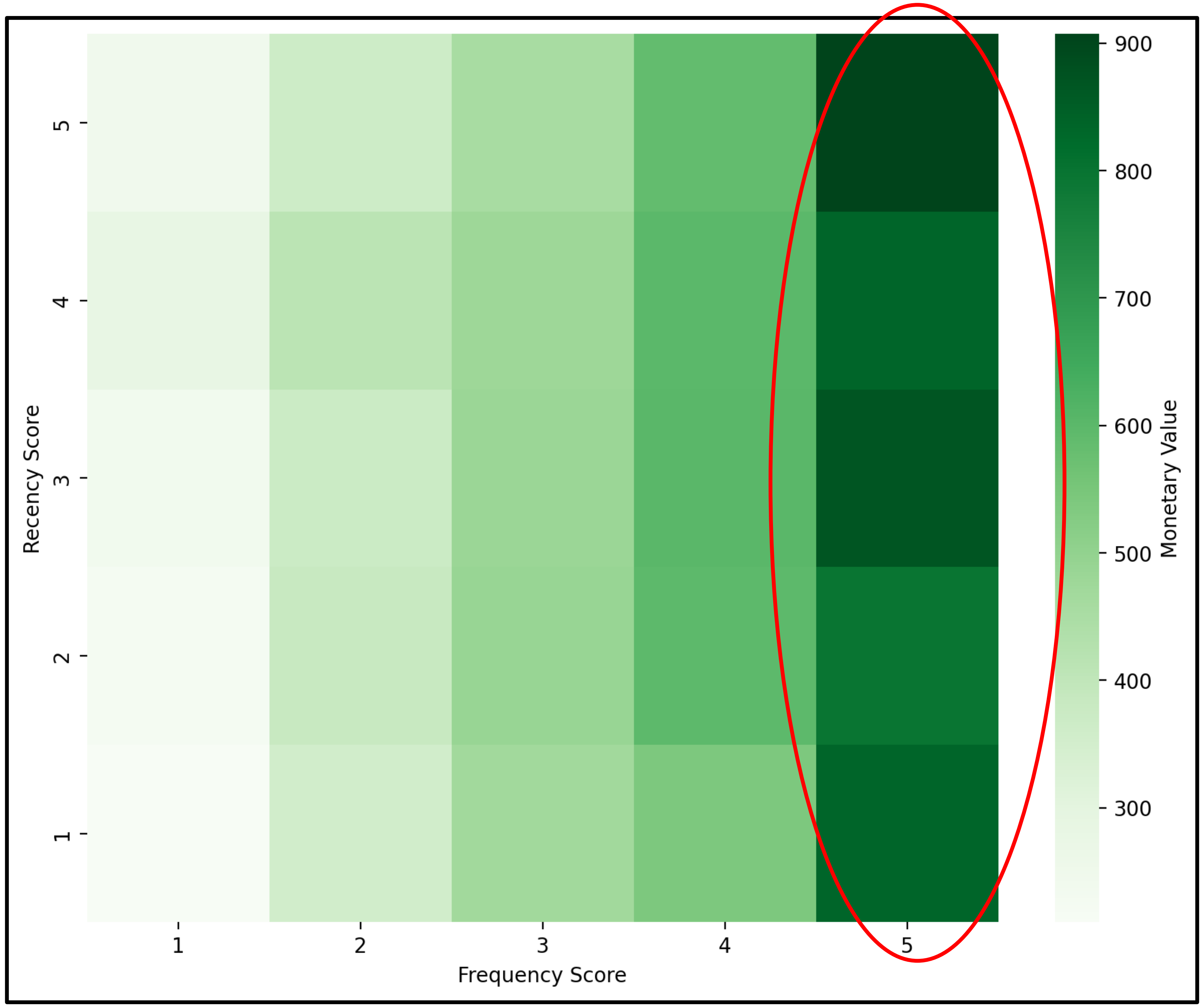
=== In-Video Questions For Slide 46 ===

1. Can you offer some explanations on why customers shop with irregular frequencies?
2. What kinds of businesses will have these customers?

Distribution of Monetary Score



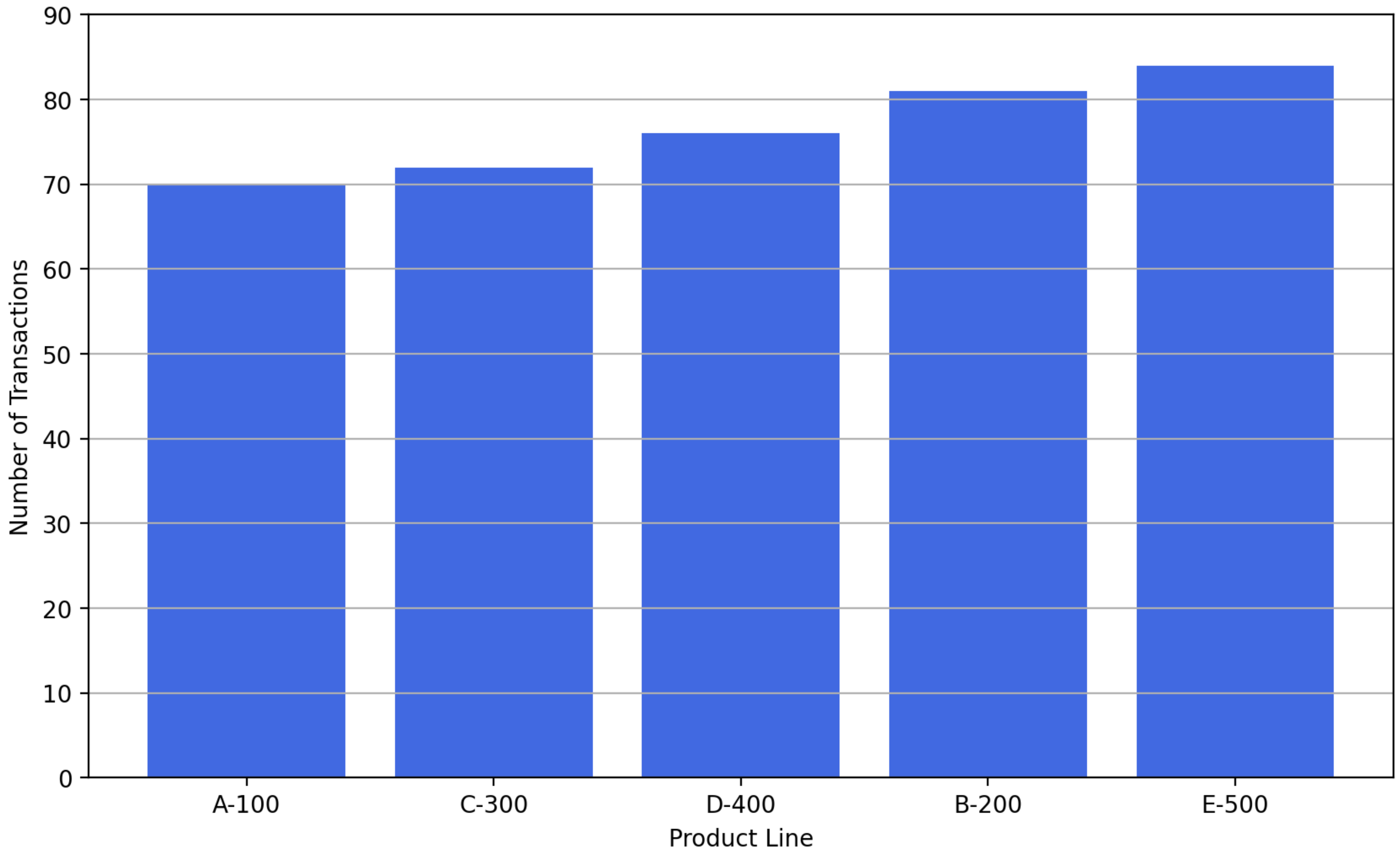
Customers With High Monetary Values



Monetary seems to be driven primarily by Frequency

Products Purchased by the 555 Group

- The product E-500 is the most popular!
- Well, the product A-100 is also very liked!



Lesson 3:
Put Observations with Similar Feature Values in Clusters

Put These Numbers Into Groups

1, 2, 3, 4, 5, 11, 12, 13, 14, 15, 101, 102, 103, 104, 105

Put These Numbers Into Groups

- According to Numeric Magnitudes
  1. {1, 2, 3, 4, 5}
  2. {11, 12, 13, 14, 15}
  3. {101, 102, 103, 104, 105}
- According to Odd / Even Types
  1. {1, 3, 5, 11, 13, 15, 101, 103, 105}
  2. {2, 4, 12, 14, 102, 104}
- According to Numeric Magnitudes and Odd / Even Type
  1. {1, 3, 5}
  2. {11, 13, 15}
  3. {101, 103, 105}
  4. {2, 4}
  5. {12, 14}
  6. {102, 104}

Put These Motorized Vehicles into Groups

=== In-Video Questions For Slide 56 ===

1. How would you put these motorized vehicles into groups?
2. What criteria did you use?
3. Are any criteria considered continuous features?

Possible Grouping Criteria

By Physical Attributes

- **Number of Wheels?** 0, 2, 4, …
- **Weight?** 300 lb. (motorcycle) to Infinity
- **Top Speed?** 23 mph (cruise ship) to 17,500 mph (space shuttle)
- **Payload?** 200 lb. (motorcycle), 50 tons (space shuttle), 248 tons (747), 50,000 tons (cruise ship) to Infinity (freight train)

By Soft Attributes

- **Number of Passengers?** 1, 2, 3, 4, …
- **Sticker Price?** $20,000 (car) to Infinity
- **Personal Ownership?** 0 or 1
- **Travel Environment?** Sea, Land, Air, or Space
- **Satisfaction of Owning The Vehicle?** High, Medium, and Low

Group By Common Sense…

Aerospace Group

Land Group – Four Wheels

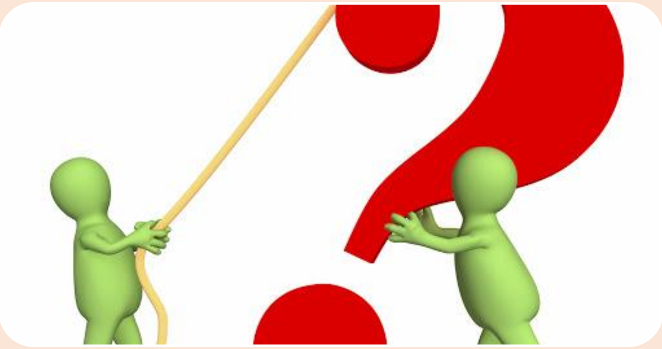
Land Group – Many Wheels

Marine Group

Land Group – Two Wheels

A Cluster is Different From a Clutter

What is Cluster Analysis?



Hypothesis – Assume There Are Clusters

- The observations are drawn from different populations.
- There are more than one population, otherwise, why find clusters?

Goals – Identify the Clusters

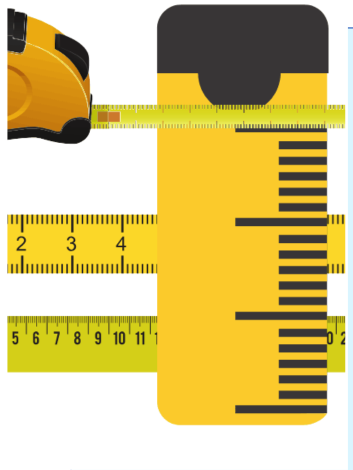
- Objects within the same cluster are as *similar* as possible.
- Objects from different clusters are as *dissimilar* as possible.

Tasks – Construct the Clusters

- Determine the number of disjoint clusters
- Must assign *similar* observations to the same cluster

Issues in Finding Clusters

How to process attributes?



How to measure “similarity”?

How to assign observations into clusters?

How to determine the number of clusters?

Cluster Centroids

- A centroid is the *centerpiece* and the *spokesperson* of a cluster.
- If the observations have p features, then a centroid is a p-dimensional array.
- Each array element is a location statistic (e.g., mean, median, or mode but not necessarily numeric) of the respective feature in the cluster.
- Therefore, a centroid may not be an observation in a cluster.

Cluster Identifier

- We identified clusters using consecutive non-negative integers.
- The Cluster Identifiers are merely integer labels.
- **Disclaimers**. The Cluster Identifiers do not indicate the discovery order of the clusters, the relative magnitudes of the centroids, or any relationships among the clusters.

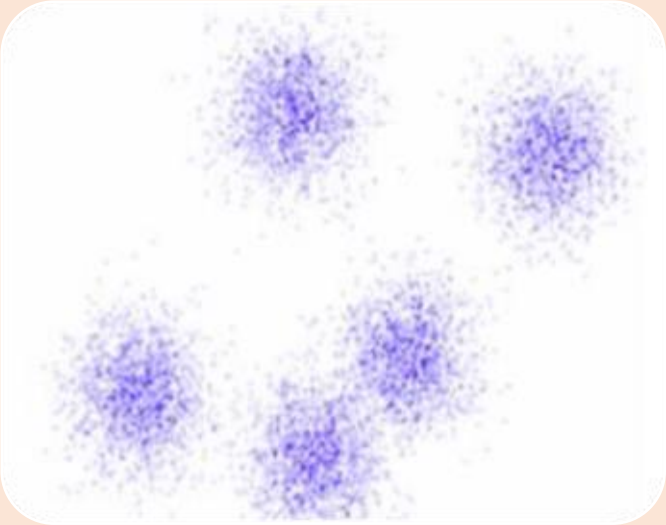
Measure Similarity with Distance Metric

- Suppose $\mathbf{x}_r$ and $\mathbf{x}_s$ are the r^{th} and the s^{th} observations, respectively.
- Both observations consists of p variables.
- We measure the distance between the observations $\mathbf{x}_r$ and $\mathbf{x}_s$, denoted as $d(\mathbf{x}_r, \mathbf{x}_s)$. The distance will indicate the similarity between the two observations.
- The smaller the distance, the more *similar* the two observations are. The larger the distance, the more *dissimilar* the two observations are.

Four Requirements for Distance Metric

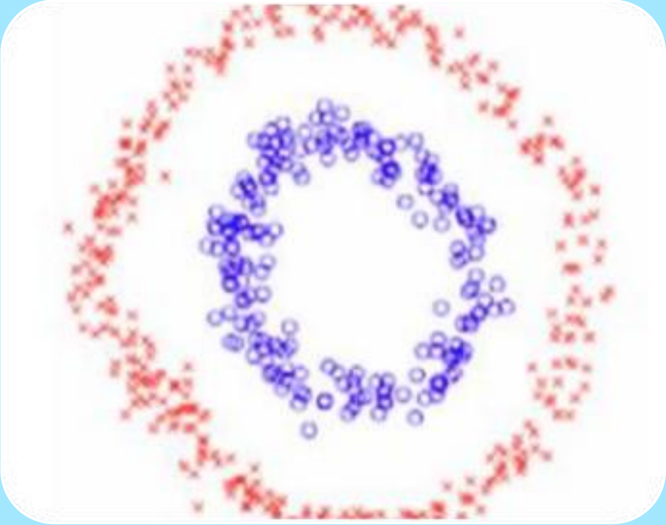
1. **Non-negativity**. $d(\mathbf{x}_r, \mathbf{x}_s\,) \geq 0$.
2. **Symmetry**. $d(\mathbf{x}_r, \mathbf{x}_s\,) = d(\mathbf{x}_s, \mathbf{x}_r\,)$.
3. **Coincidence**. $d(\mathbf{x}_r, \mathbf{x}_s\,) = 0$ if and only if $\mathbf{x}_r = \mathbf{x}_s$.
4. **Subadditivity**. $d(\mathbf{x}_r, \mathbf{x}_t\,) + d(\mathbf{x}_t, \mathbf{x}_s\,) \geq d(\mathbf{x}_r, \mathbf{x}_s\,)$ where $\mathbf{x}_t$ is another observation.

Compact vs. Connected Cluster Structures



Compact Clusters

- Compare the distance an observation to its centroid (intra-cluster distance) to other clusters’ centroids (inter-cluster distances).
- Imagine enclosing observations in a cluster by a circle.



Connected Clusters

- Defined by distance an observation to its neighbors and the density of other observations around it.
- Think about connecting the observations in a cluster by a curve.

Clustering Algorithms

Number of Categorical Features	Number of Continuous Features	Clustering Algorithm
0	> 0	k-Means or k-Medians
> 0	0	k-Modes
> 0	> 0	k-Prototypes

In the interest of time, we will only cover the k-Means algorithm here.

The k-Means Algorithm for Continuous Features

Rescale, if Deemed Necessary

Standardize

- $y = (x - \bar{x})/s_x$
- The mean of x is $\bar{x}$ and the standard deviation of x is s_x
- Then, $\bar{y} = 0$ and $s_y = 1$.
- The centroids are usually scattered around zero.

Range

- $y = A \times \left(x - x_{[1]}\right)/\left(x_{[n]} - x_{[1]}\right)$
- The minimum of x is $x_{[1]}$ and the maximum of x is $x_{[n]}$
- Then, $y_{[1]} = 0$ and $y_{[n]} = A$.
- Variations within clusters will be bounded within $[0, A]$

Common Distance for Continuous Features

- Euclidean
- Manhattan
- Chebyshev
- Cosine

The $\boldsymbol{k}$-Means Cluster Algorithm

- Centroids are the sample mean of observations in a cluster

$$\mathbf{c}_i \equiv \bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{\mathbf{x}_{ij} \in C_i} \mathbf{x}_{ij} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_{ij}$$

- Total Within-Cluster Variation (TWCV)

$$\sum_{i=1}^{K} \sum_{\mathbf{x}_{ij} \in C_i} d^2\left(\mathbf{x}_{ij}, \bar{\mathbf{x}}_i\right)$$

The $\boldsymbol{k}$-Means Algorithm

1. For a fixed $k > 1$ number of clusters
2. Specify k arrays with p elements as the initial centroids
3. Repeat the following four sub-tasks
   a. calculate the distance of each observation to all the centroids
   b. assigning each observation to the cluster with the shortest distance
   c. update (i.e., re-compute) the centroids of all clusters
   d. exit if the Total Within-Cluster variation converges (in practice, check for no changes in cluster memberships)

Multiple Trials for Initial Centroids

- Since it is a *de-facto* standard to limit the number of iterations in an iterative algorithm such as the k-Means algorithm, an iteration may terminate too early resulting in a non-optimal solution.
- Therefore, a common strategy is to rerun the k-Means algorithm with different initial centroids. Finally, return the solution that produces the most compact clusters.

What is the Number of Clusters?

- How many disjoint segments the data exhibits?
- What is the number of clusters that best separated the data?
- We must have this is a piece of information that we usually do not know.
- The common practice is to use the Elbow value and the Silhouette Index to help us decide.

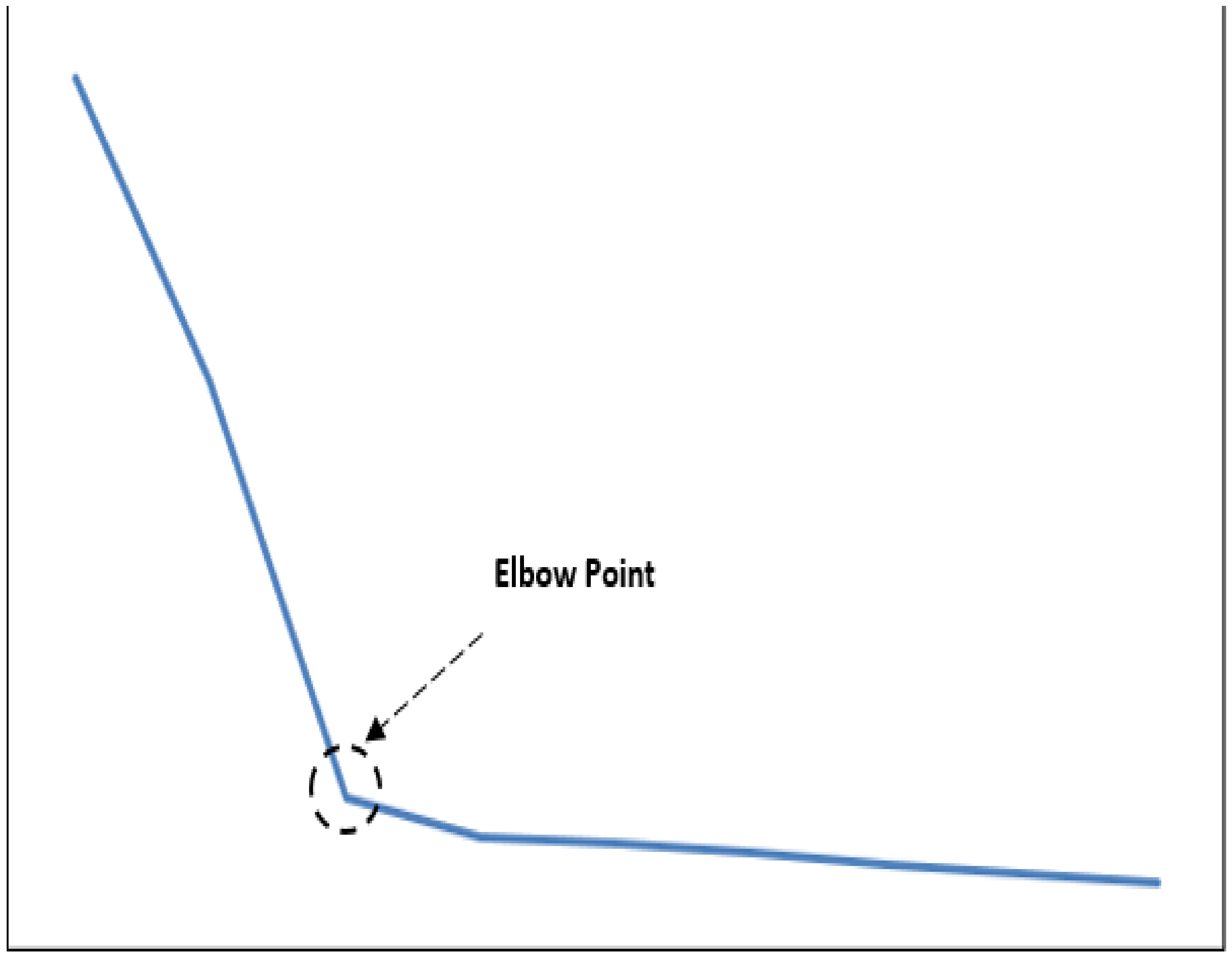
The Elbow Method

- The Within-Cluster Variation $\mathrm{WCV}_i = \sum_{\mathbf{x}_{ij} \in C_i} d^2(\mathbf{x}_{ij}, \bar{\mathbf{x}}_i)$
- The WCV is usually larger for a cluster with many observations, thus we need to account for the size of a cluster.
- Let n_i be the number of observations in the cluster
- For K number of clusters, the measure is:

$$W_K = \sum_{i=1}^{K} \frac{1}{n_i} \left(\sum_{\mathbf{x}_{ij} \in C_i} d^2(\mathbf{x}_{ij}, \bar{\mathbf{x}}_i) \right) = \sum_{i=1}^{K} \frac{\mathrm{WCV}_i}{n_i}$$

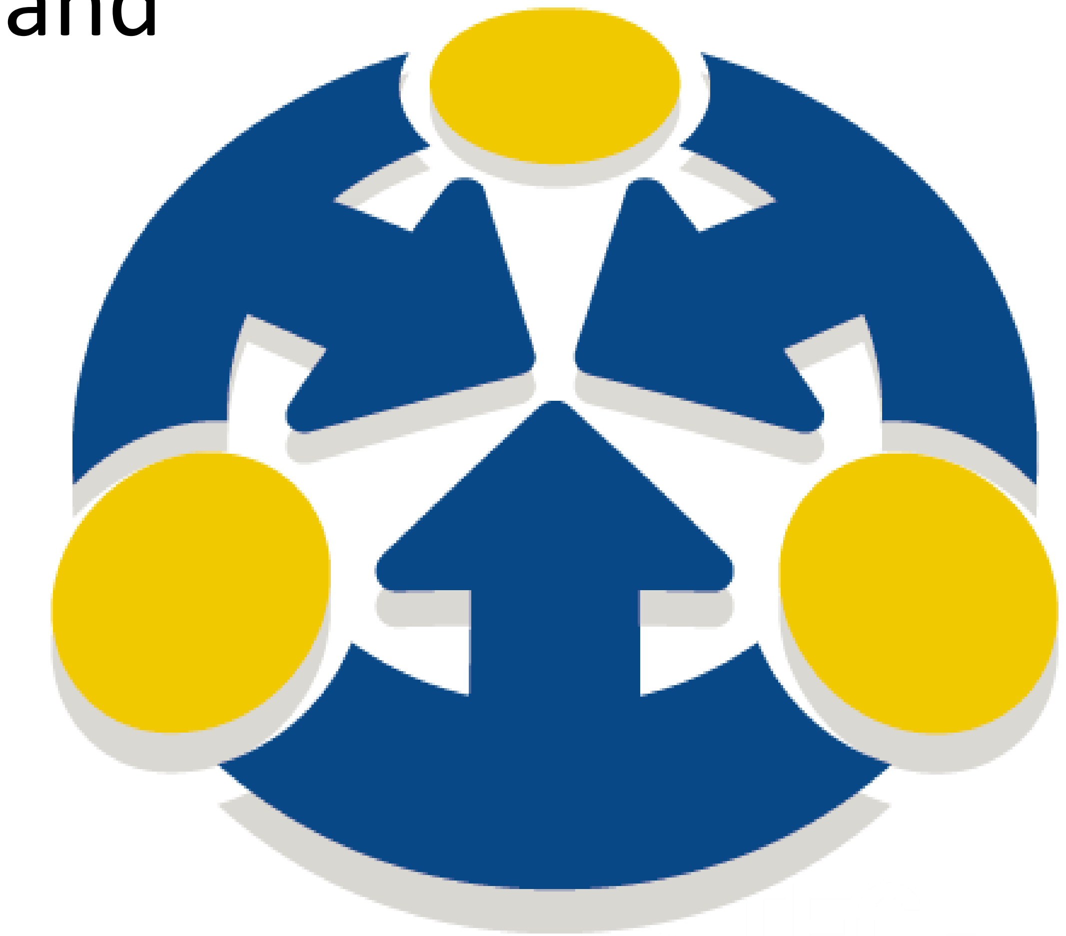
The Elbow Method

- Create clusters for $k = 1, 2, \ldots$ and up to a conventionally specified upper limit
- Plot W_K versus k
- The curve is decreasing *in theory*
- Select k that corresponds to the ***first elbow*** in the L-curve



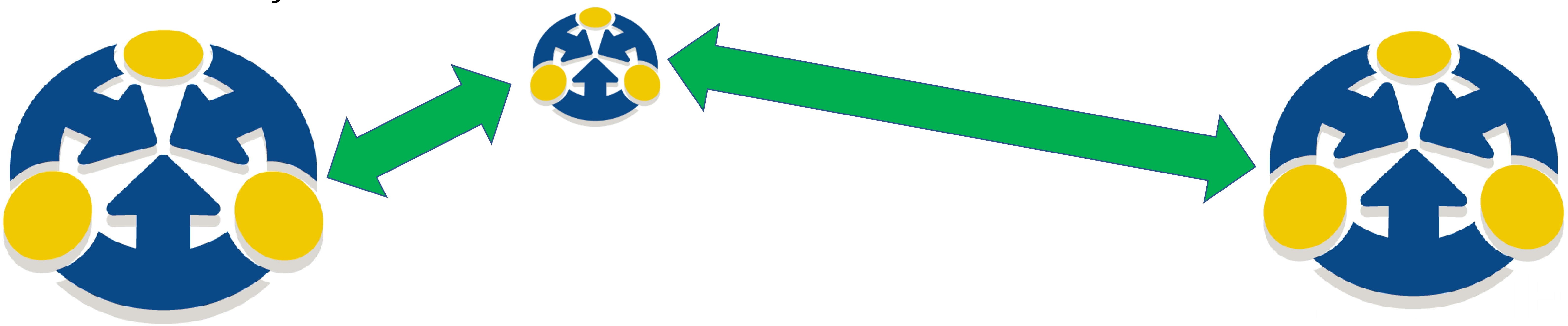
The Silhouette Index

- Define $a_{ij} = \sum_{\mathbf{x}_{ij}, \mathbf{x}_{is} \in C_i, j \neq s} d\left(\mathbf{x}_{ij}, \mathbf{x}_{is}\right) / (n_i - 1)$
- a_{ij} is the average distance between the observation $\mathbf{x}_{ij}$ and all other $n_{C_i} - 1$ observations in the same cluster.
- If $n_{C_i} = 1$, then $a_{ij} = 0$ (by definition).
- These a_{ij} indicates how compact a cluster is.



The Silhouette Index

- Define $d_{ij,C_r} = \sum_{\mathbf{x}_{ij} \in C_i, \mathbf{x}_{rs} \in C_r} d(\mathbf{x}_{ij}, \mathbf{x}_{rs})/n_r$
- d_{ij,C_r} is the average distance between the observation $\mathbf{x}_{ij}$ in cluster C_i and all n_r observations in the cluster C_r.
- Finally, define $b_{ij} = \min(d_{ij,C_r} : r \neq i)$ which is the average distance of the observation $\mathbf{x}_{ij}$ to its *nearest neighboring* cluster.



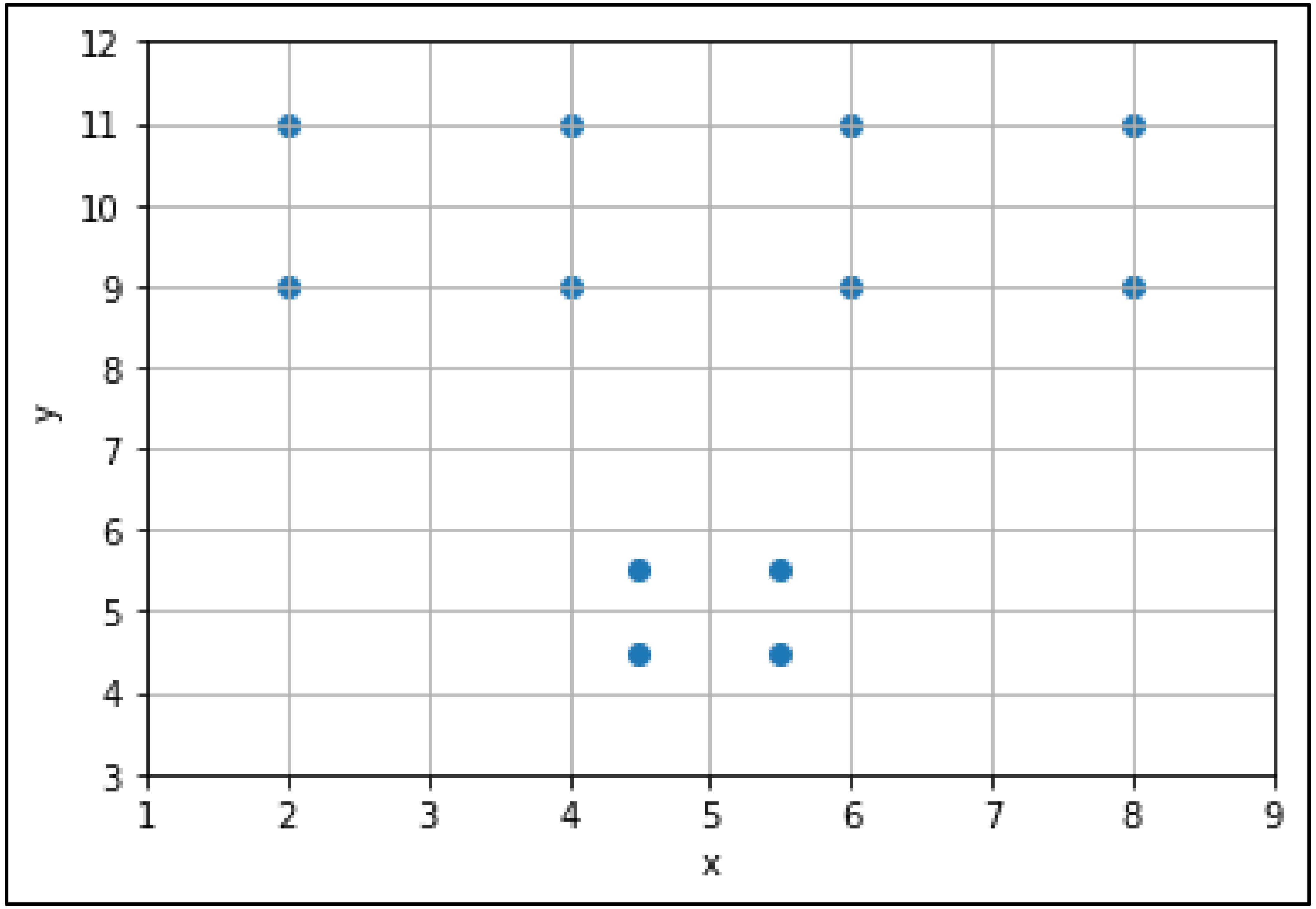
The Silhouette Index

- The Silhouette width of the observation $\mathbf{x}_{ij}$ is $s_{ij} = \frac{b_{ij} - a_{ij}}{\max(a_{ij}, b_{ij})}$
- The Silhouette Index $\sum_{i=1}^{K} \sum_{j=1}^{n_i} s_{ij} / \sum_{i=1}^{K} n_i$.
- The Silhouette Index is undefined when there is only one cluster.
- The Silhouette Index has a range of [-1, 1].
    - A larger value is better
    - +1 indicates a perfect clustering result
    - -1 indicates the worst clustering result

- **Function**: sklearn.cluster.KMeans(n_clusters=, init = 'random', n_init='auto', random_state=None)¶
- **Description**:
  - Discover *n_clusters* clusters with continuous features.
  - Choose initial centroids randomly with multiple trials.
  - To replicate the results, you must specify a positive integer for *random_state*.
  - The only distance metric offered is Euclidean as of version 1.3.2.
- **Reference**: sklearn.cluster.KMeans.html

How Many Clusters Are There?

x	y
2	11
4	11
2	9
4	9
6	11
8	11
6	9
8	9
4.5	5.5
5.5	5.5
5.5	4.5
4.5	4.5

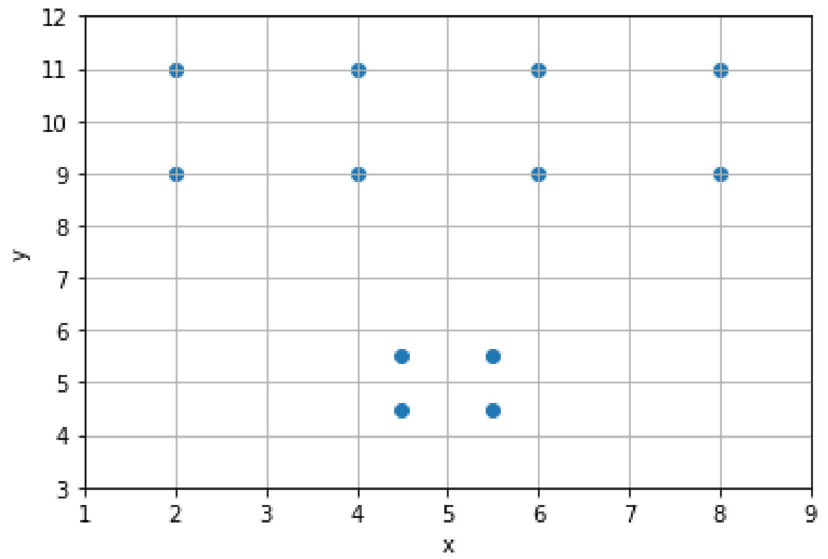


=== In-Video Questions For Slide 81 ===

1. How many clusters do you see based on the chart?
2. What is the minimum number of clusters?
3. What is the maximum number of clusters?

How Many Clusters Are There?

- We surely see *at least* two clusters.
  - The four points in the lower half of the chart are clearly apart from the re
  - Thus, there are at least two clusters.
- We will try number of clusters from 1 to 11.
  - Start at one anyway for the sake of completeness.
  - There are twelve observations. The KMeans implementation can go as high as one fewer than the number of observations.
- We use a random seed of 5712023.
- We will try ten random sets of initial centroids.



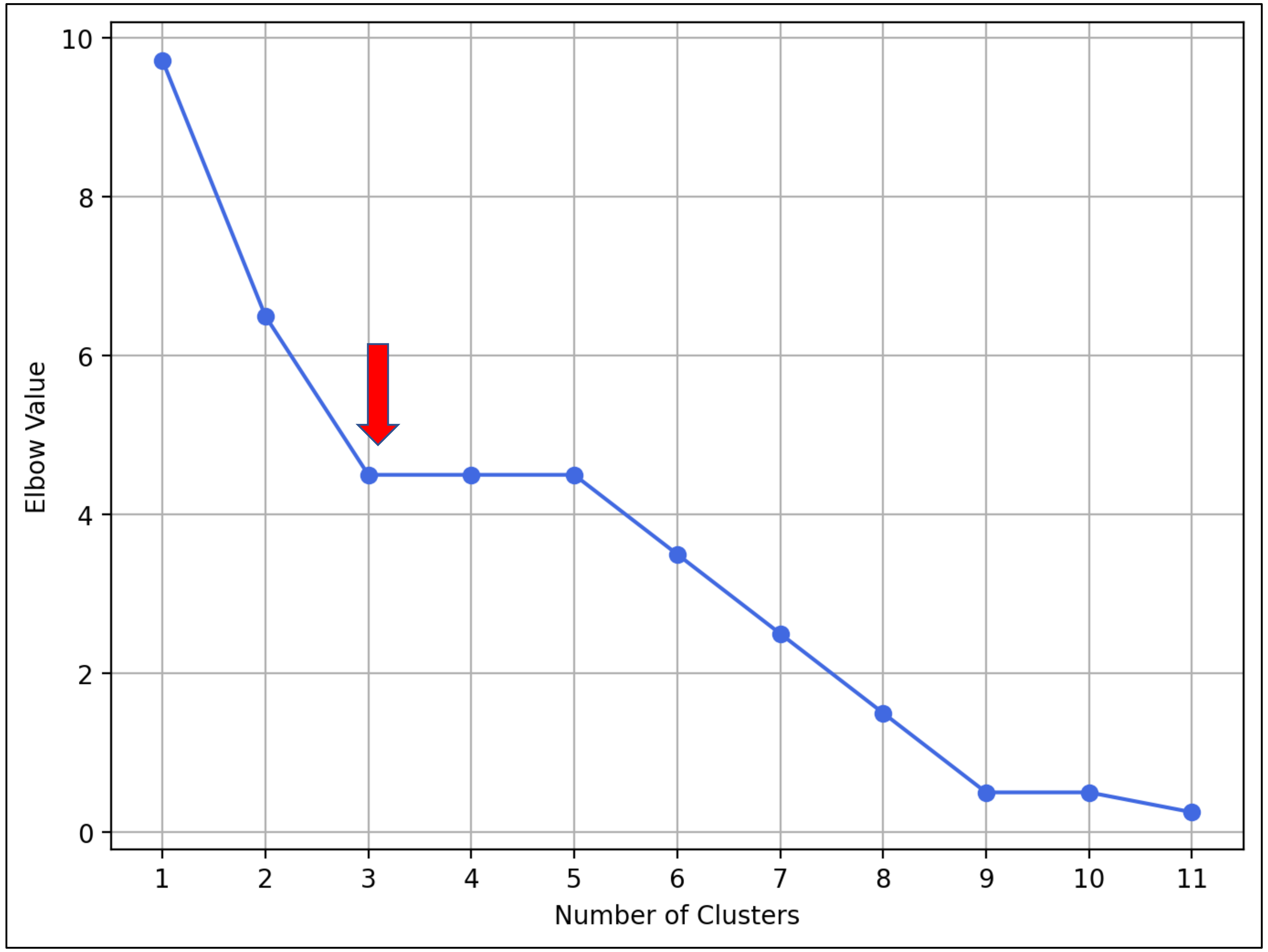
Two-Dimensional Example

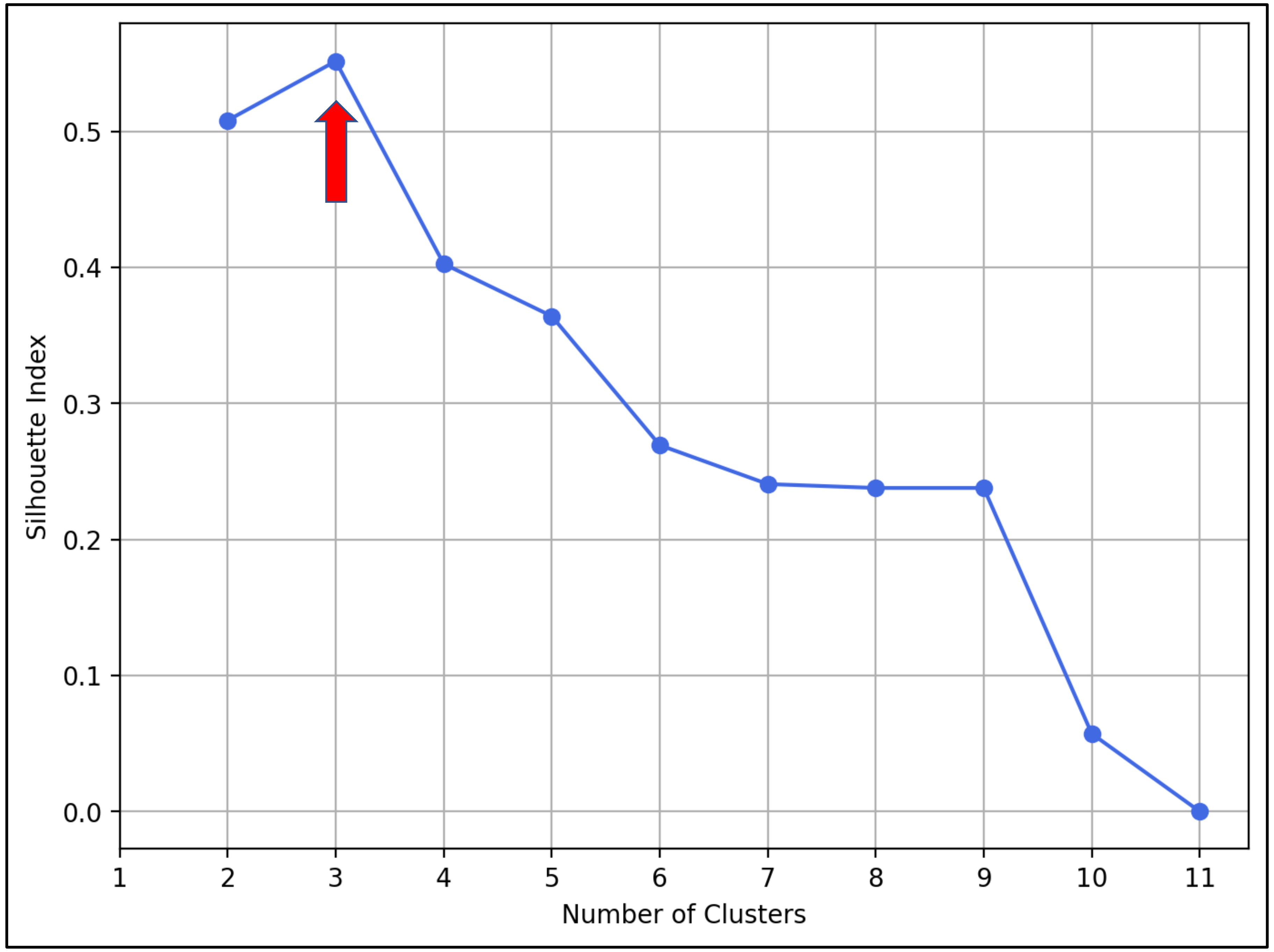
Module 4 KMeans 2D Example.py

Results For Various Number of Clusters

1	116.6667	9.7222	
2	50	6.5	0.5080
3	18	4.5	0.5515
4	14	4.5	0.4025
5	10	4.5	0.3641
6	8	3.5	0.2693
7	6	2.5	0.2407
8	4	1.5	0.2378
9	2	0.5	0.2378
10	1	0.5	0.0572
11	0.5	0.25	0

Elbow Value and Silhouette Index





Locate The Elbow Programmatically

- In a typical Elbow chart, the Elbow values decrease as we increase the number of clusters. The Elbow is the point where the decrease slows down.
- If we calculate the slope (i.e., the decrease per additional cluster) and then the acceleration (i.e., the change in slope per additional cluster), the Elbow is the point where the acceleration is the highest.
- We may use our best judgment to take the number of clusters (or less one) at the Elbow to be our optimal number of clusters.

Acceleration = Rate of Slope Changes

Slope[i] = (Elbow[i] – Elbow[i-1]) / (NCluster[i] – NCluster[i-1])

Acceleration[i] = (Slope[i] – Slope[i-1]) / (NCluster[i] – NCluster[i-1])

1	9.7222		
2	6.5	-3.2222	
3	4.5	-2	1.2222
4	4.5	0	2
5	4.5	0	0
6	3.5	-1	-1
7	2.5	-1	0
8	1.5	-1	0
9	0.5	-1	0
10	0.5	0	1
11	0.25	-0.25	-0.25

My Rule of Thumb is to pick the choice before the largest acceleration

Choose Optimal Number of Clusters by Common Sense

Driving Distances in Miles From Chicago

Data

- Driving distances (in miles) from Chicago to 59 cities
- DistanceFromChicago.csv

Clustering

- Discover up to 15 clusters
- DrivingMilesFromChicago is a continuous feature

CityState	StateCode	City	DrivingMilesFromChicago
Albany, NY	NY	Albany	820
Albuquerque, NM	NM	Albuquerque	1341
Atlanta, GA	GA	Atlanta	712
Baltimore, MD	MD	Baltimore	704
Billings, MT	MT	Billings	1247
Birmingham, AL	AL	Birmingham	661
Boise, ID	ID	Boise	1702
Boston, MA	MA	Boston	986
Buffalo, NY	NY	Buffalo	531
Charleston, WV	WV	Charleston	484
Charleston, SC	SC	Charleston	911
Charlotte, NC	NC	Charlotte	770
Cheyenne, WY	WY	Cheyenne	968
Cleveland, OH	OH	Cleveland	342
Columbia, SC	SC	Columbia	802
Columbus, OH	OH	Columbus	352
Dallas, TX	TX	Dallas	933
Denver, CO	CO	Denver	1009
Des Moines, IA	IA	Des Moines	333
Detroit, MI	MI	Detroit	278
EL Paso, TX	TX	EL Paso	1488
Fargo, ND	ND	Fargo	644
Grand Junction, CO	CO	Grand Junction	1252
Hartford, CT	CT	Hartford	903
Houston, TX	TX	Houston	1089
Indianapolis, IN	IN	Indianapolis	179
Jackson, MS	MS	Jackson	747
Jacksonville, FL	FL	Jacksonville	1058
Kansas City, MO	MO	Kanas City	529
Las Vegas, NV	NV	Las Vegas	1755
Little Rock, AR	AR	Little Rock	655
Los Angeles, CA	CA	Los Angeles	2028
Louisville, KY	KY	Louisville	297
Memphis, TN	TN	Memphis	536
Miami, FL	FL	Miami	1373
Milwaukee, WI	WI	Milwaukee	92
Minneapolis, MN	MN	Minneapolis	407
Nashville, TN	TN	Nashville	472
New Orleans, LA	LA	New Orleans	927
New York, NY	NY	New York	811
Norfolk, VA	VA	Norfolk	891
Oklahoma City, OK	OK	Oklahoma City	796
Omaha, NE	NE	Omaha	469
Orlando, FL	FL	Orlando	1152
Philadelphia, PA	PA	Philadelphia	761
Phoenix, AZ	AZ	Phoenix	1804
Pittsburgh, PA	PA	Pittsburgh	460
Portland, ME	ME	Portland	1087
Portland, OR	OR	Portland	2122
Rapid City, SD	SD	Rapid City	909
Reno, NV	NV	Reno	1924
Saint Louis, MO	MO	Saint Louis	300
Salt Lake City, UT	UT	Salt Lake City	1404
San Antonio, TX	TX	San Antonio	1210
San Diego, CA	CA	San Diego	2088
San Francisco, CA	CA	San Francisco	2148
Seattle, WA	WA	Seattle	2070
Washington, DC	DC	Washington	705
Wichita, KS	KS	Wichita	725

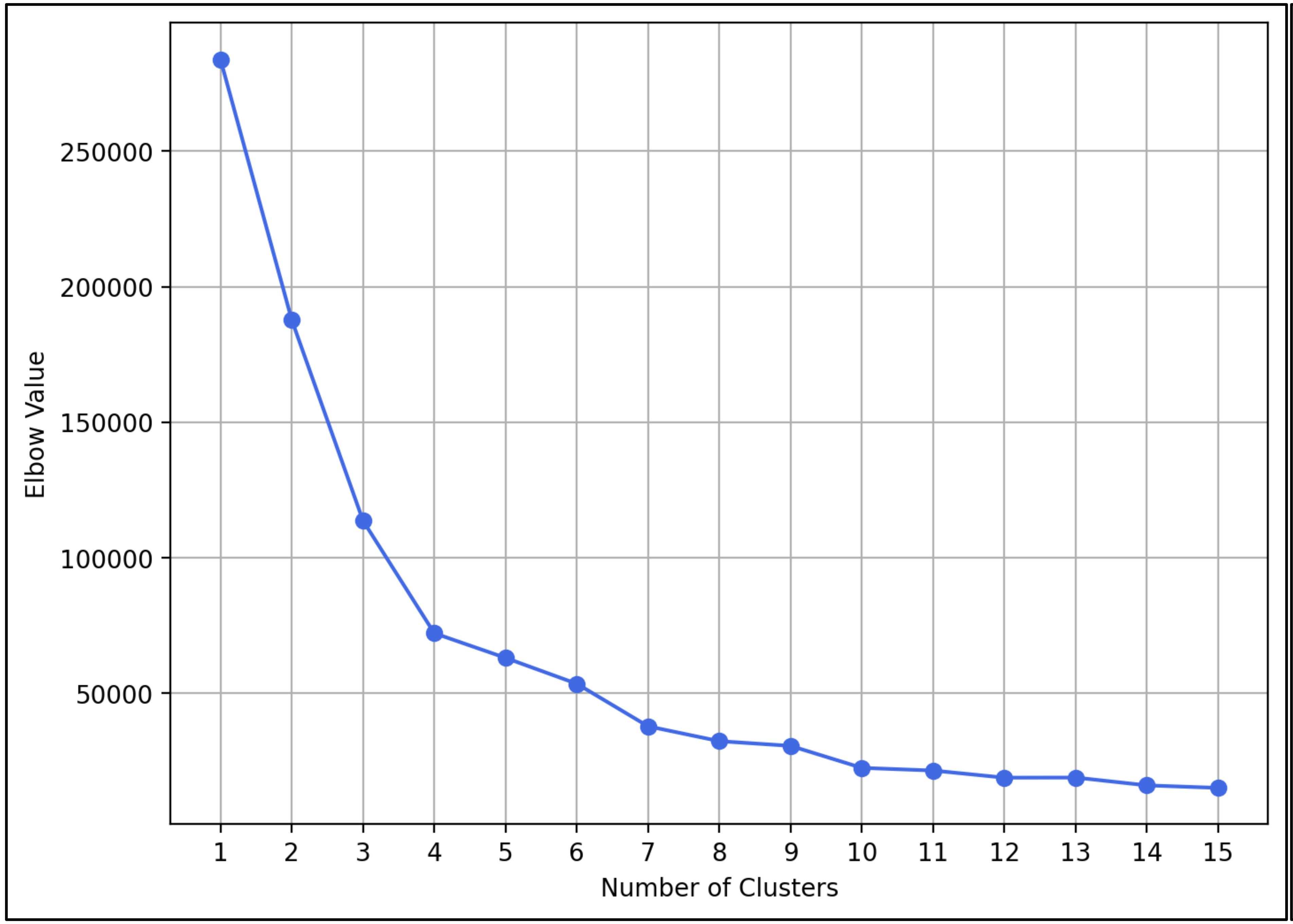
Driving Distances in Miles From Chicago

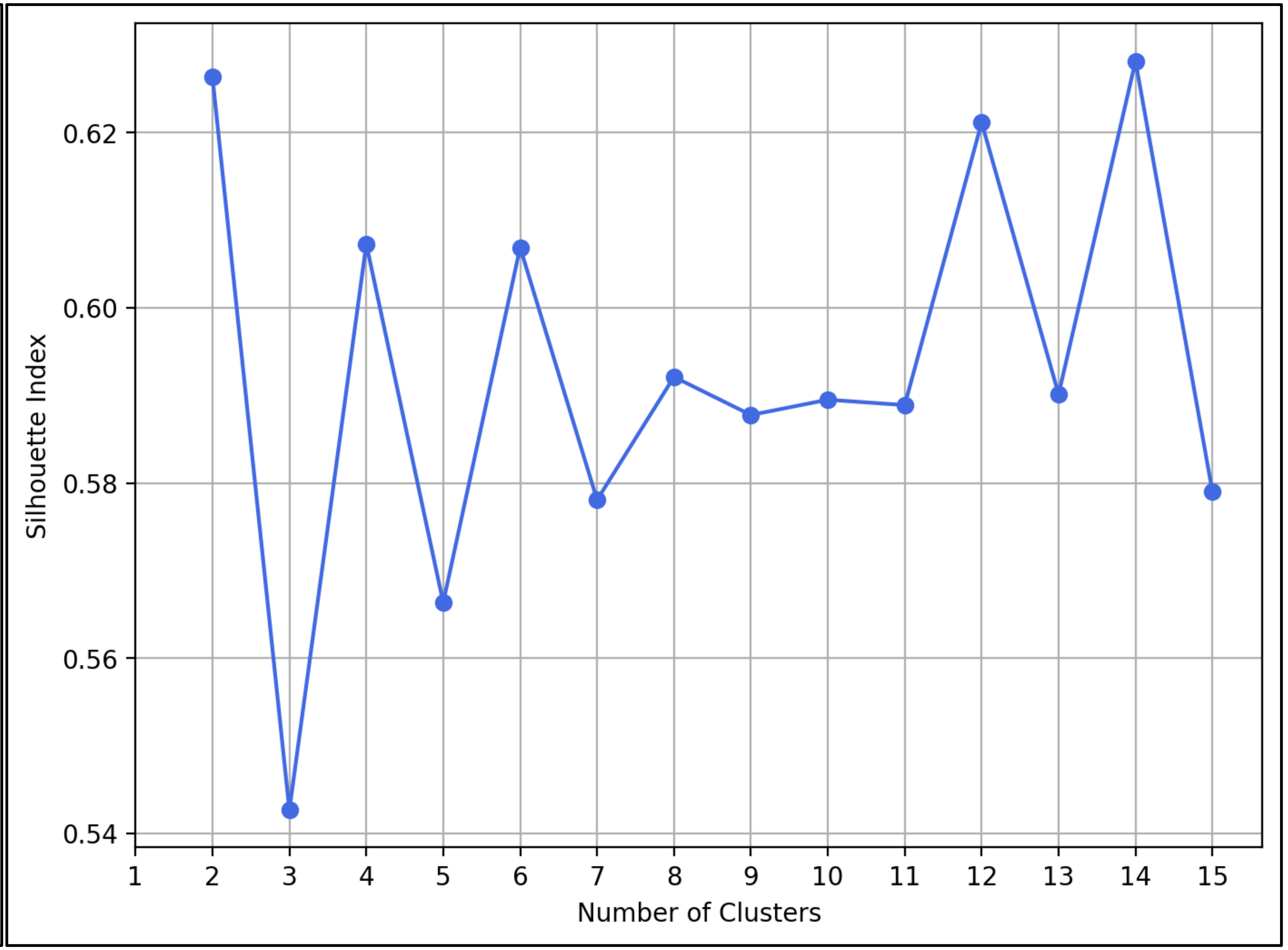
Module 4 Distance From Chicago.py

Cluster Results

1	16,749,775.1864	283,894.4947	
2	4,954,271.0000	187,758.4433	0.6264
3	2,163,889.1592	113,637.3584	0.5426
4	968,307.9667	72,117.1928	0.6073
5	692,893.1466	63,014.6480	0.5663
6	492,246.7698	53,459.1751	0.6069
7	347,433.4845	37,758.5304	0.5781
8	223,938.4948	32,275.5589	0.5921
9	188,838.4234	30,537.3497	0.5878
10	158,787.0476	22,415.4062	0.5895
11	111,339.5317	21,426.0054	0.5889
12	90,462.8214	18,799.3326	0.6212
13	82,098.3714	18,834.7563	0.5901
14	75,276.5095	15,956.2620	0.6282
15	55,819.5095	15,024.4009	0.5790

Elbow Value and Silhouette Index





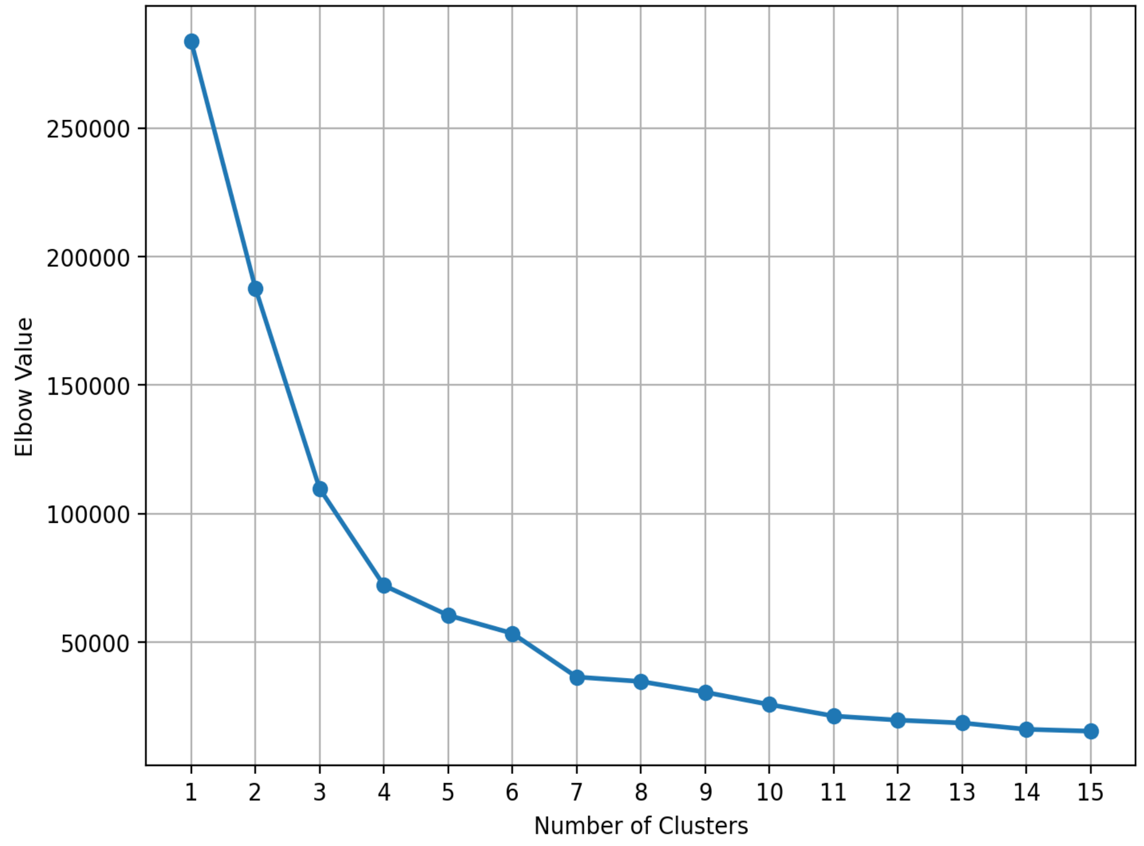
=== In-Video Questions For Slide 93 ===

1. If you can locate an elbow in the Elbow chart, where is it?
2. The Silhouette Index chart has multiple local peaks. Which peak will you choose for the number of clusters?

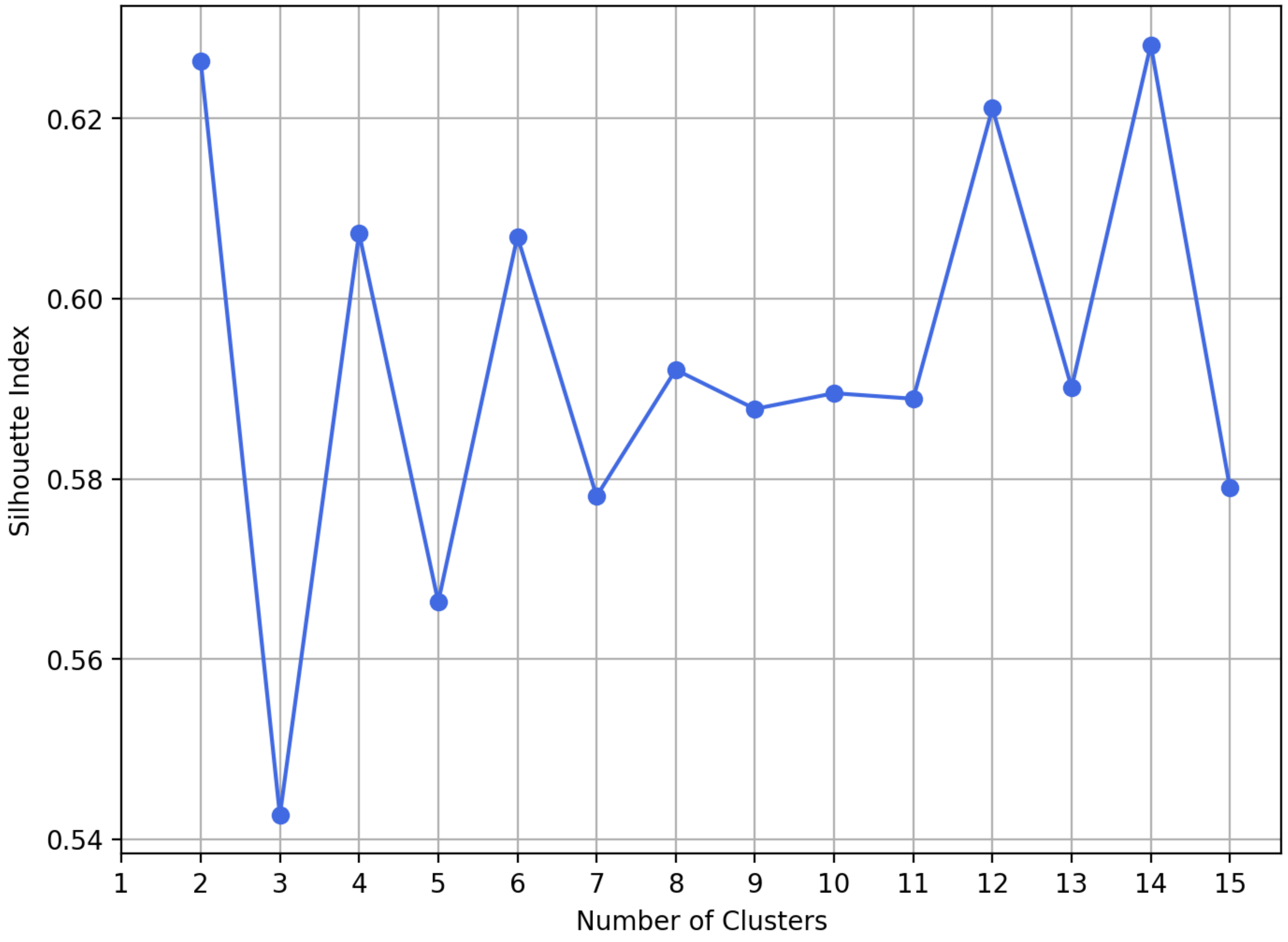
Locate the Largest Acceleration

1	283,894.4947		
2	187,758.4433	-96,136.0514	
3	113,637.3584	-74,121.0849	22,014.9665
4	72,117.1928	-41,520.1656	32,600.9194
5	63,014.6480	-9,102.5448	32,417.6207
6	53,459.1751	-9,555.4729	-452.9281
7	37,758.5304	-15,700.6447	-6,145.1718
8	32,275.5589	-5,482.9715	10,217.6732
9	30,537.3497	-1,738.2092	3,744.7623
10	22,415.4062	-8,121.9435	-6,383.7343
11	21,426.0054	-989.4008	7,132.5427
12	18,799.3326	-2,626.6728	-1,637.2720
13	18,834.7563	35.4237	2,662.0965
14	15,956.2620	-2,878.4943	-2,913.9180
15	15,024.4009	-931.8611	1,946.6332

My Rule of Thumb chooses three clusters. But the four clusters also look interesting.



Locate the Local Maximum of Silhouette



- The Silhouette chart shows a local valley at the three-cluster solution, so my Rule of Thumb choice does not work!
- But there is a local peak at the four-cluster solution. So, let’s study the four clusters.

Common Sense Suggests Four Clusters

Cluster ID	Number of Observations	Centroid	Within-Cluster Sum of Squares
3	16	378 miles	255,120.4
1	23	815 miles	280,147.9
2	11	1246 miles	204,122.7
0	9	1960 miles	228,916.9
Overall	**59**	**951 miles**	**16,749,775.0**

Members of the Four Clusters

City, State	Distance
Milwaukee, WI	92
Indianapolis, IN	179
Detroit, MI	278
Louisville, KY	297
Saint Louis, MO	300
Des Moines, IA	333
Cleveland, OH	342
Columbus, OH	352
Minneapolis, MN	407
Pittsburgh, PA	460
Omaha, NE	469
Nashville, TN	472
Charleston, WV	484
Kansas City, MO	529
Buffalo, NY	531
Memphis, TN	536

3

City, State	Distance
Fargo, ND	644
Little Rock, AR	655
Birmingham, AL	661
Baltimore, MD	704
Washington, DC	705
Atlanta, GA	712
Wichita, KS	725
Jackson, MS	747
Philadelphia, PA	761
Charlotte, NC	770
Oklahoma City, OK	796
Columbia, SC	802
New York, NY	811
Albany, NY	820
Norfolk, VA	891
Hartford, CT	903
Rapid City, SD	909
Charleston, SC	911
New Orleans, LA	927
Dallas, TX	933
Cheyenne, WY	968
Boston, MA	986
Denver, CO	1009

1

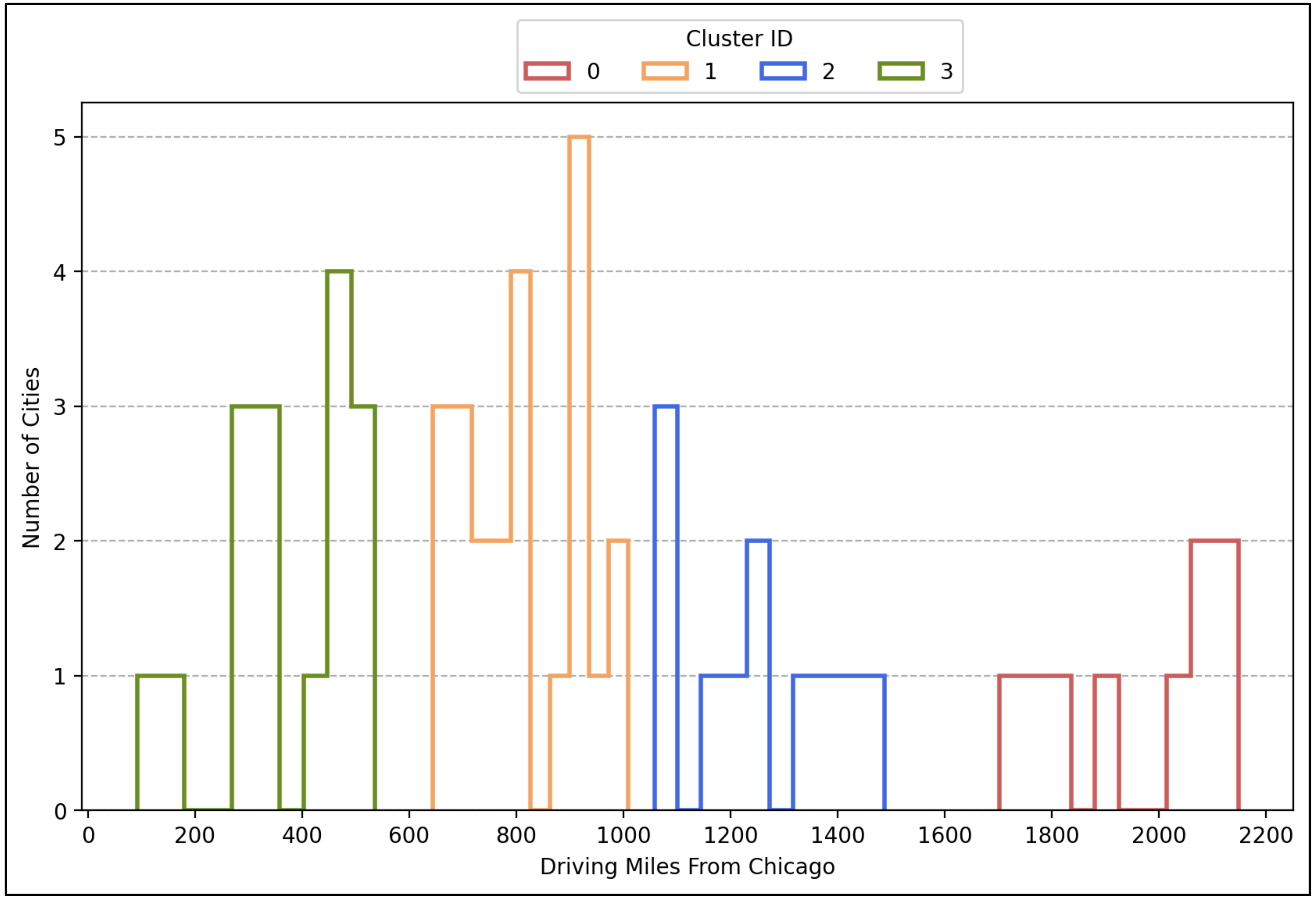
CityState	Distance
Jacksonville, FL	1058
Portland, ME	1087
Houston, TX	1089
Orlando, FL	1152
San Antonio, TX	1210
Billings, MT	1247
Grand Junction, CO	1252
Albuquerque, NM	1341
Miami, FL	1373
Salt Lake City, UT	1404
EL Paso, TX	1488

2

City, State	Distance
Boise, ID	1702
Las Vegas, NV	1755
Phoenix, AZ	1804
Reno, NV	1924
Los Angeles, CA	2028
Seattle, WA	2070
San Diego, CA	2088
Portland, OR	2122
San Francisco, CA	2148

0

Visualize How Well Clusters are Separated



- **Cluster 0** is well-separated from the other three clusters.
- **Cluster 3** is fairly separated from the other three clusters
- **Cluster 1** and **Cluster 2** narrowly separated from each other
- Unfilled histogram bars allow for overlapping clusters