

# Sketching Data With T-Digest

**Erik Erlandson**

Red Hat, Inc.

email: [eje@redhat.com](mailto:eje@redhat.com)

twitter: [@manyangled](https://twitter.com/manyangled)

github: [erikerlandson](https://github.com/erikerlandson)

# Why Sketching?



# Why Sketching?



- **Smaller**

# Why Sketching?



- **Smaller**
- **Faster**

# Why Sketching?



- **Smaller**
- **Faster**
- **Essential Features**



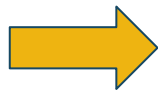
# We All Sketch Data

3.4

6.0

2.5

⋮




Mean = 3.97

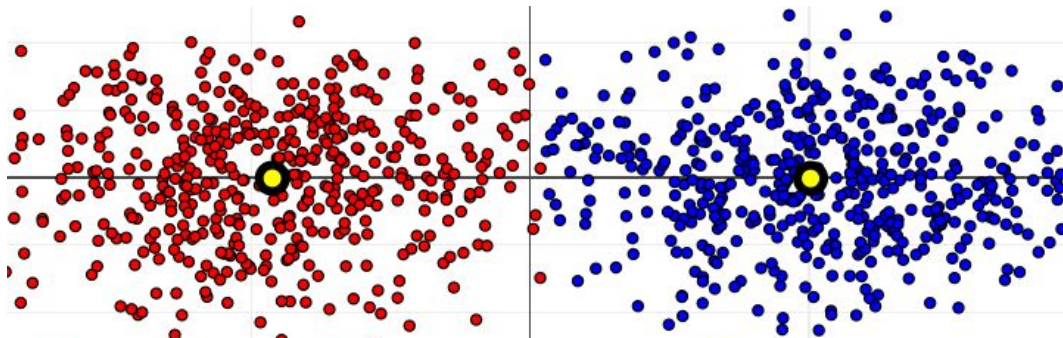
Variance = 3.30

# We All Sketch Data

3.4

6.0  Mean = 3.97  
2.5 Variance = 3.30

⋮



# We All Sketch Data

3.4

6.0

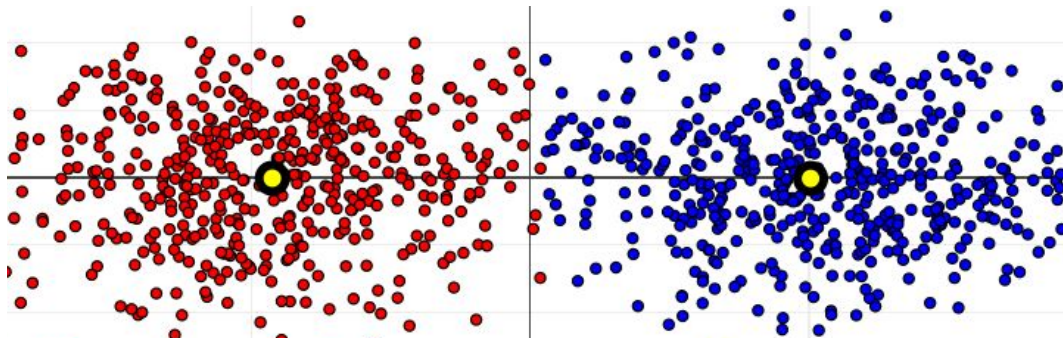
2.5

⋮



Mean = 3.97

Variance = 3.30

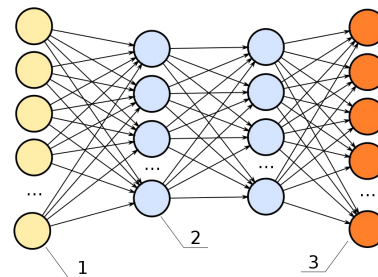


3.4, 5.0, 9.0

6.0, 2.1, 7.7

2.5, 4.4, 3.2

⋮





# T-Digest

*Computing Extremely Accurate Quantiles Using t-Digests*

Ted Dunning & Omar Ertl

Java, Python, R, JS, C++ and Scala

Library for Spark and PySpark

# What is T-Digest Sketching?

3.4

6.0

2.5

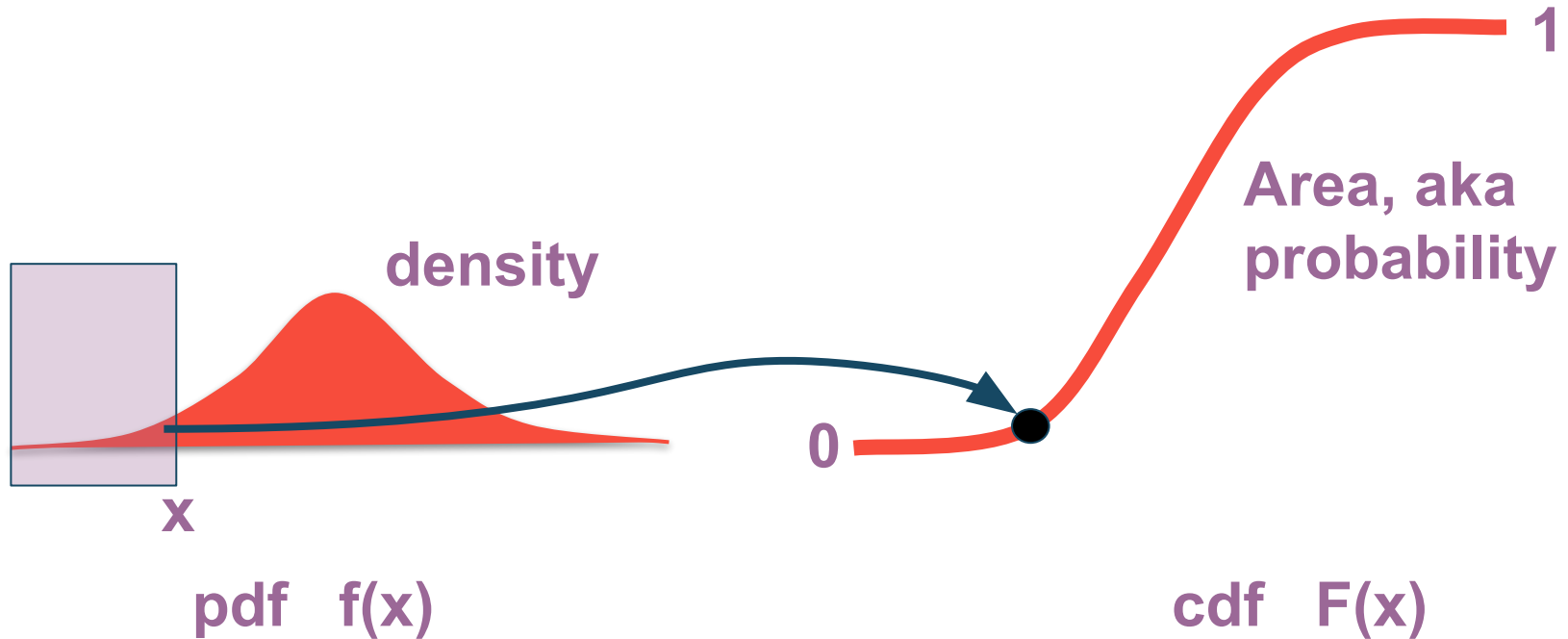
⋮



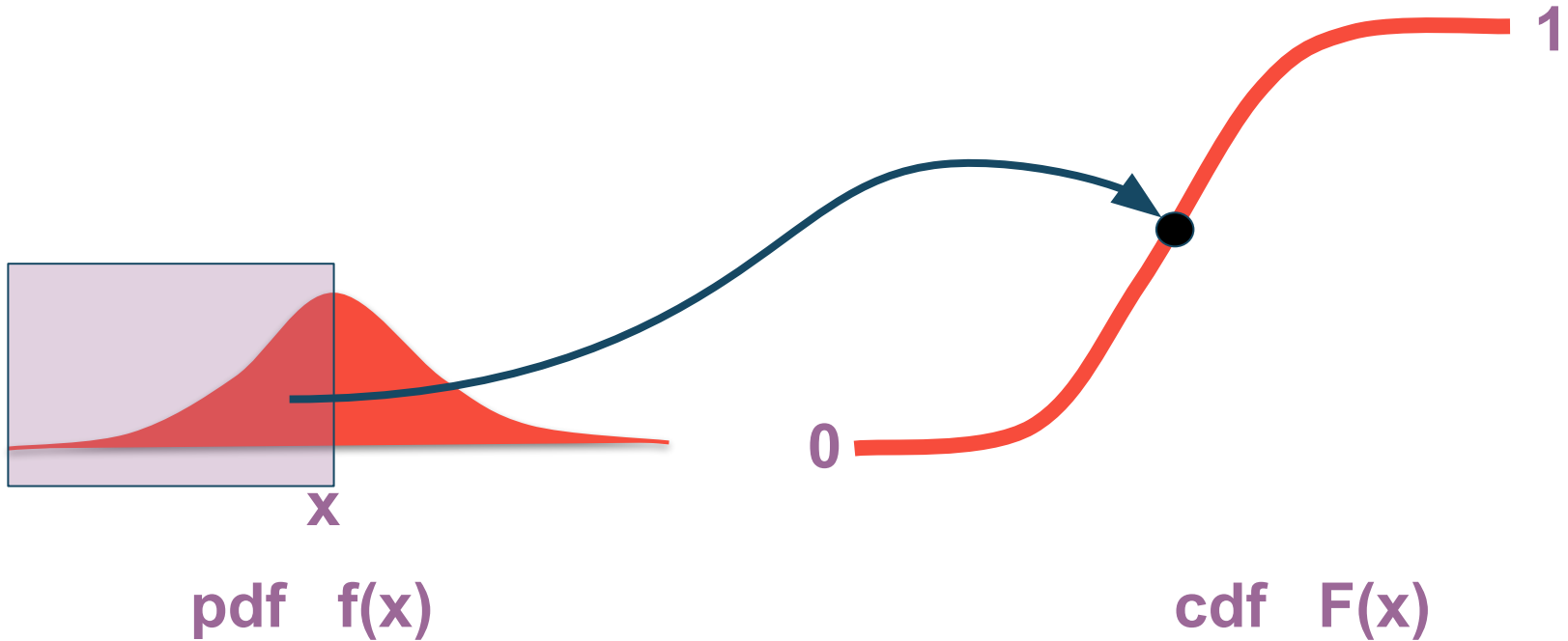
**Estimate of  
CDF**



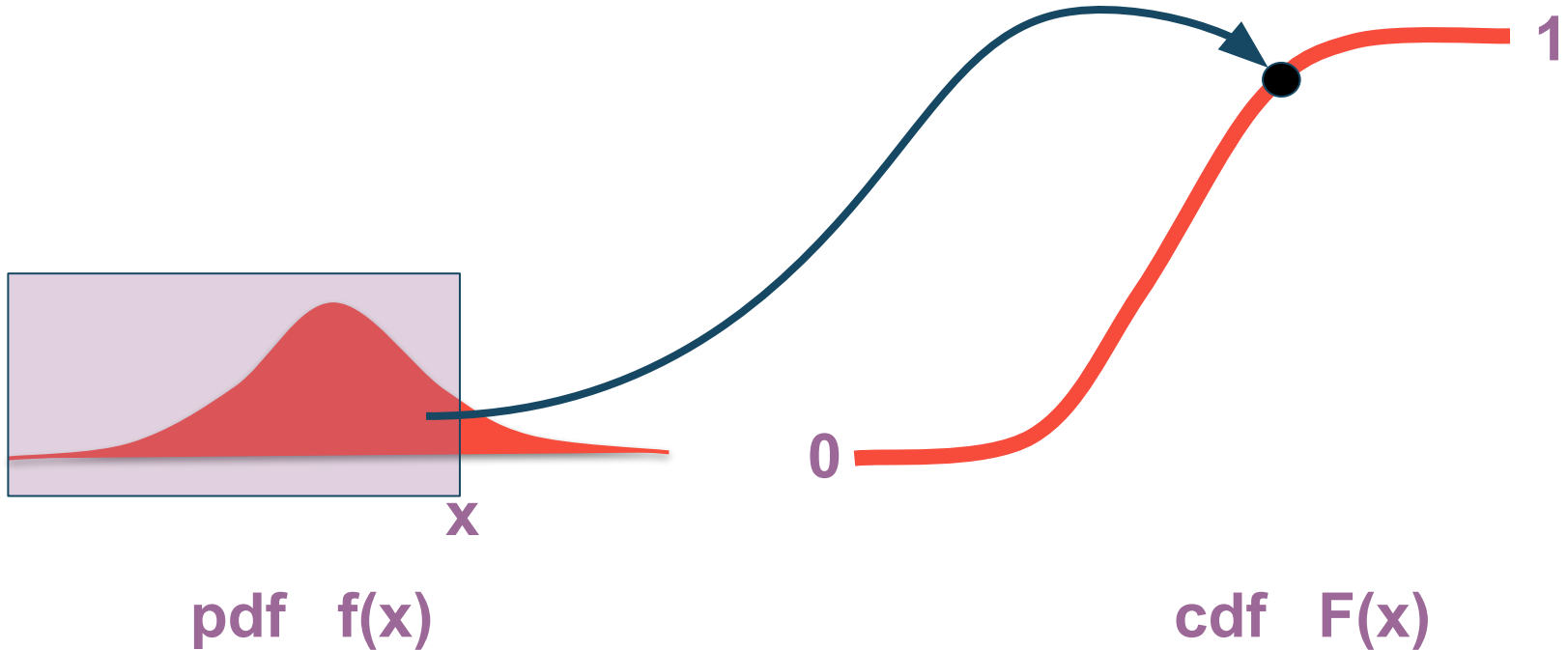
# Cumulative Distribution



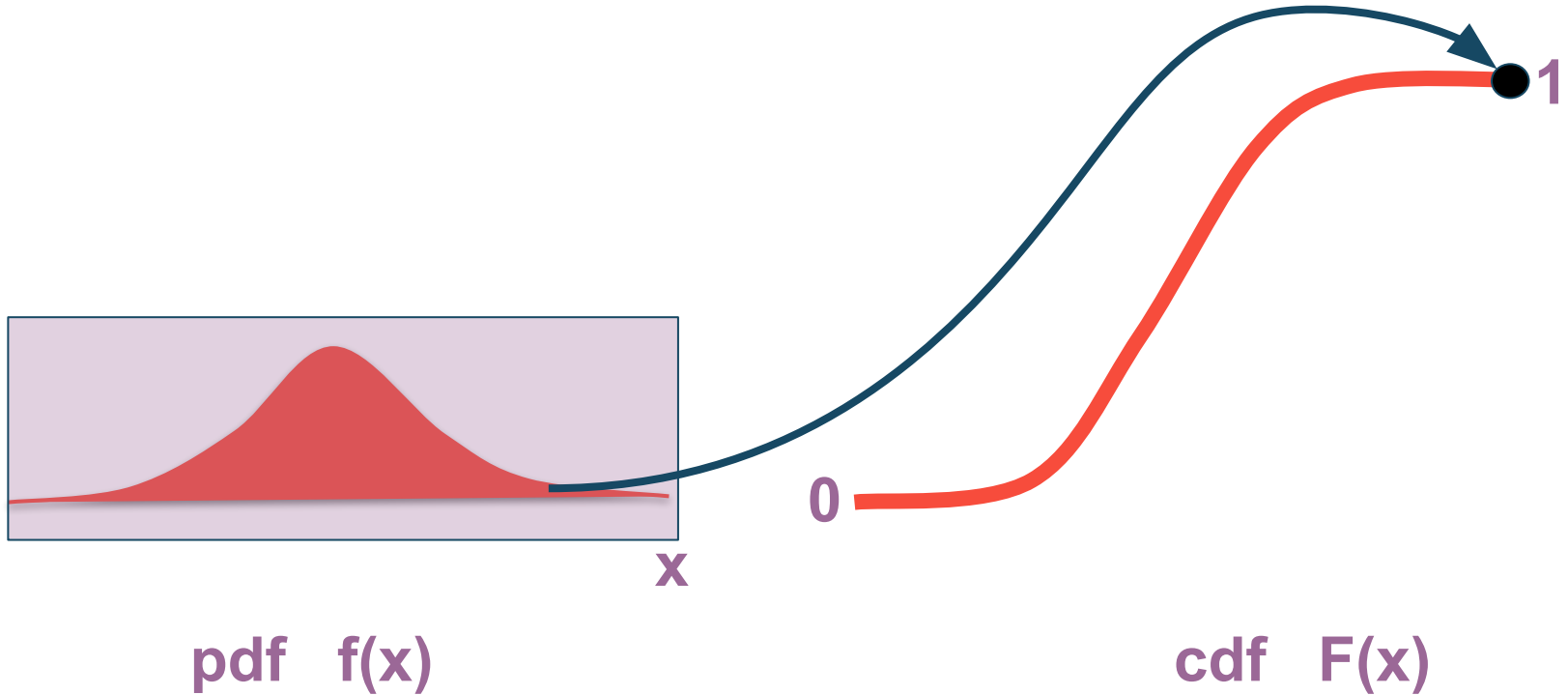
# Cumulative Distribution



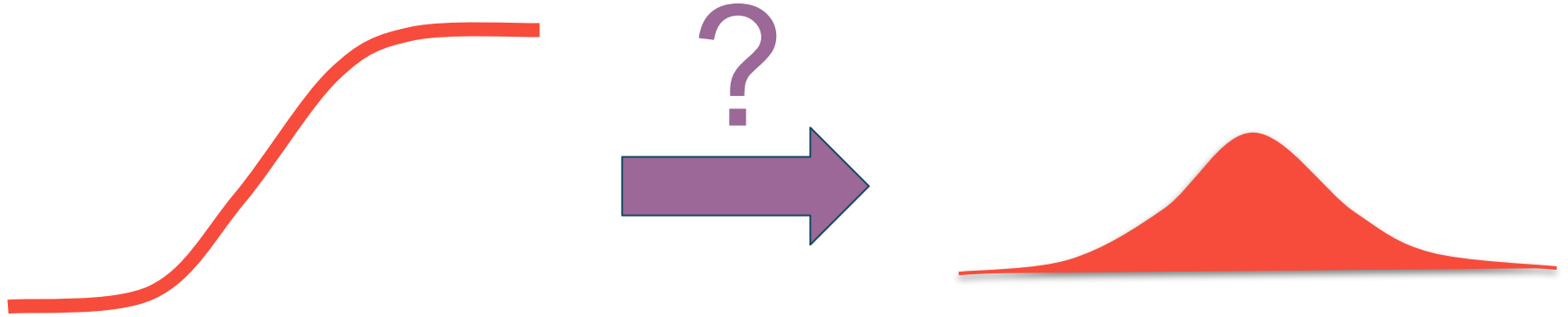
# Cumulative Distribution



# Cumulative Distribution



# But What About The Density?

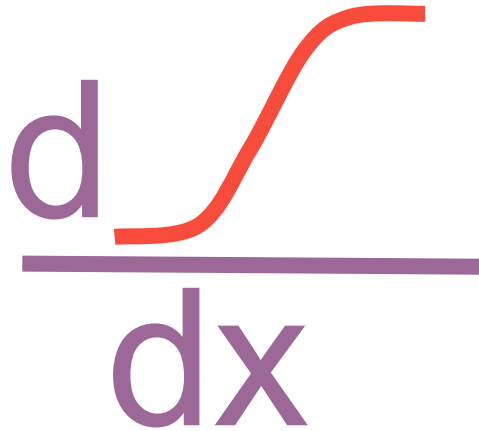


# Cumulative Distribution

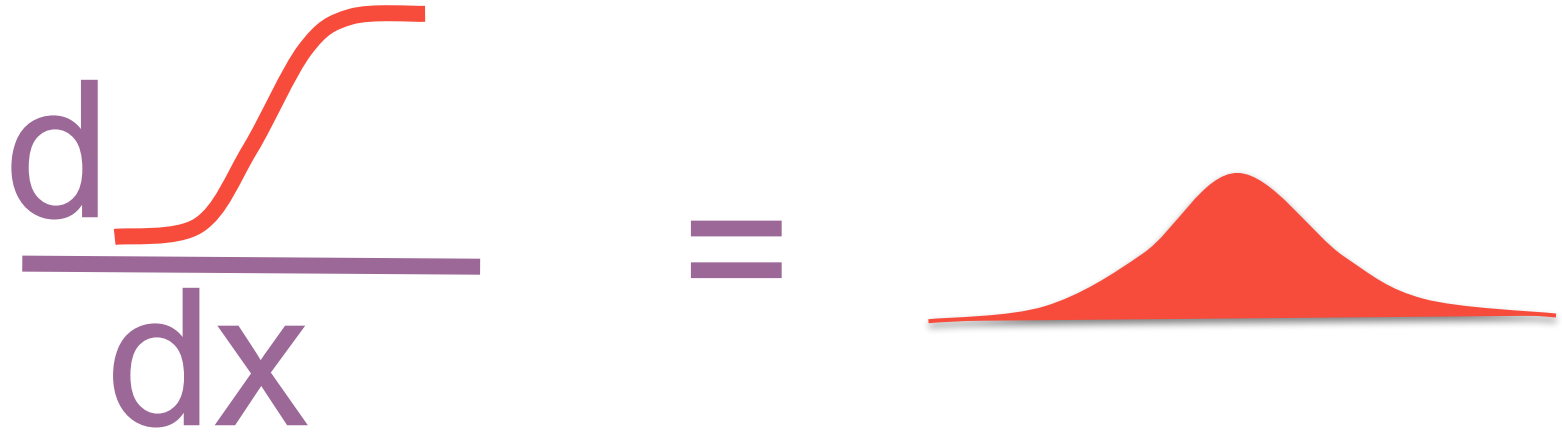




# Cumulative Distribution


$$\frac{d}{dx}$$

# Cumulative Distribution



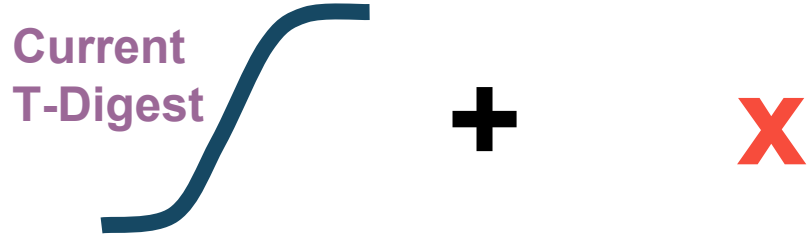
# Incremental Updates

Current  
T-Digest



# Incremental Updates

Current  
T-Digest



The diagram illustrates the formula for incremental updates. It consists of the text "Current T-Digest" in purple, followed by a dark blue S-shaped curve. To the right of the curve is a black plus sign, and further right is a large red X.

$$\text{Current T-Digest} + X$$

# Incremental Updates

Current  
T-Digest



+

X

=



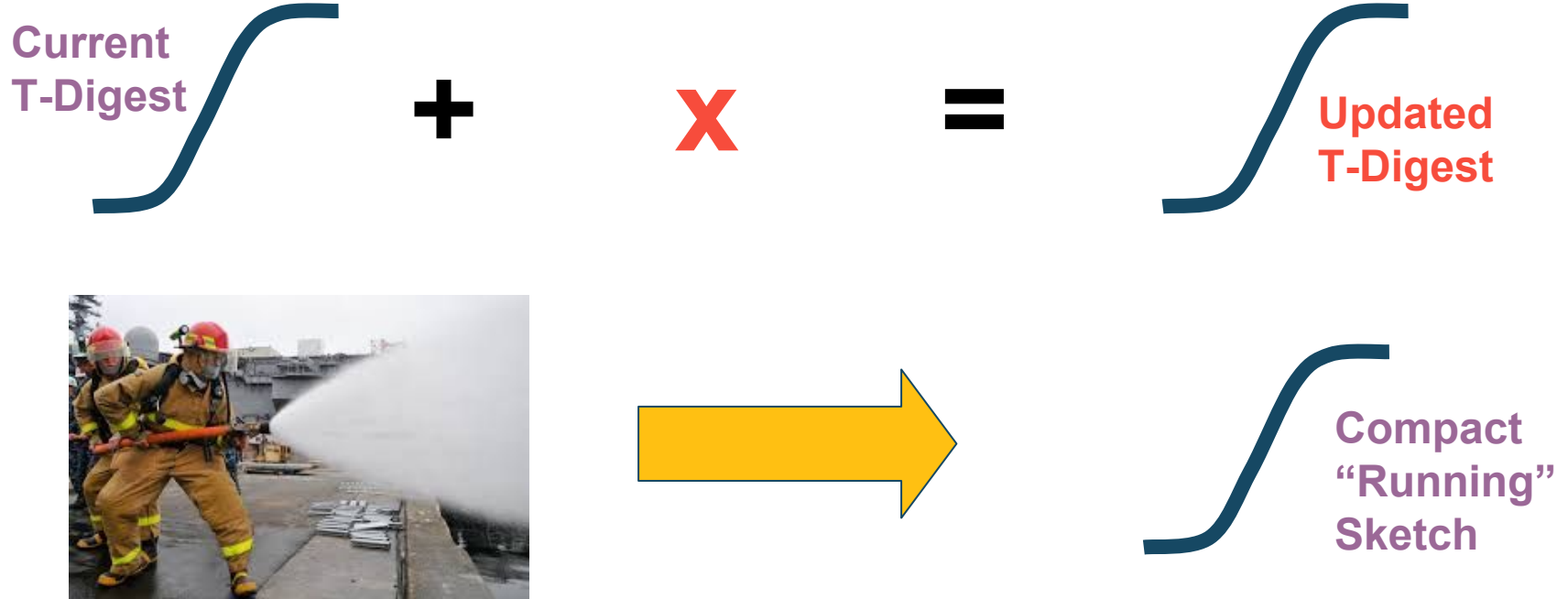
Updated  
T-Digest

# Incremental Updates

$$\text{Current T-Digest} \int + \times = \int \text{Updated T-Digest}$$



# Incremental Updates



# Payoff



**Query  
Latencies**





# Payoff



**Query  
Latencies**



**What does my  
latency distribution  
look like?**

# Payoff



**Query  
Latencies**

**Are 90% of my  
latencies under 1  
second?**

**What does my  
latency distribution  
look like?**

# Payoff



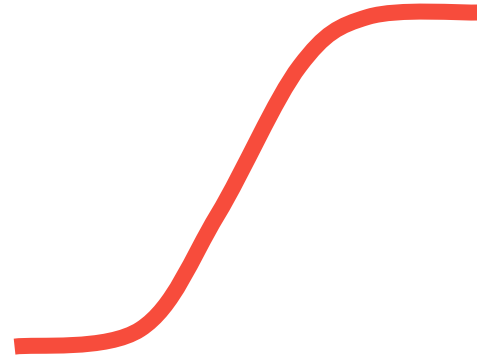
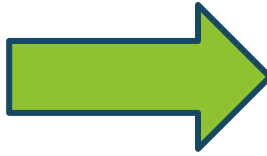
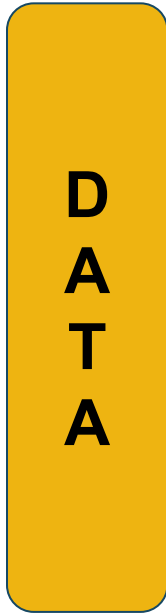
**Query  
Latencies**

**Are 90% of my  
latencies under 1  
second?**

**What does my  
latency distribution  
look like?**

**I want to simulate  
my latencies!**

# Even More Payoff



# Even More Payoff

**DATA 1**



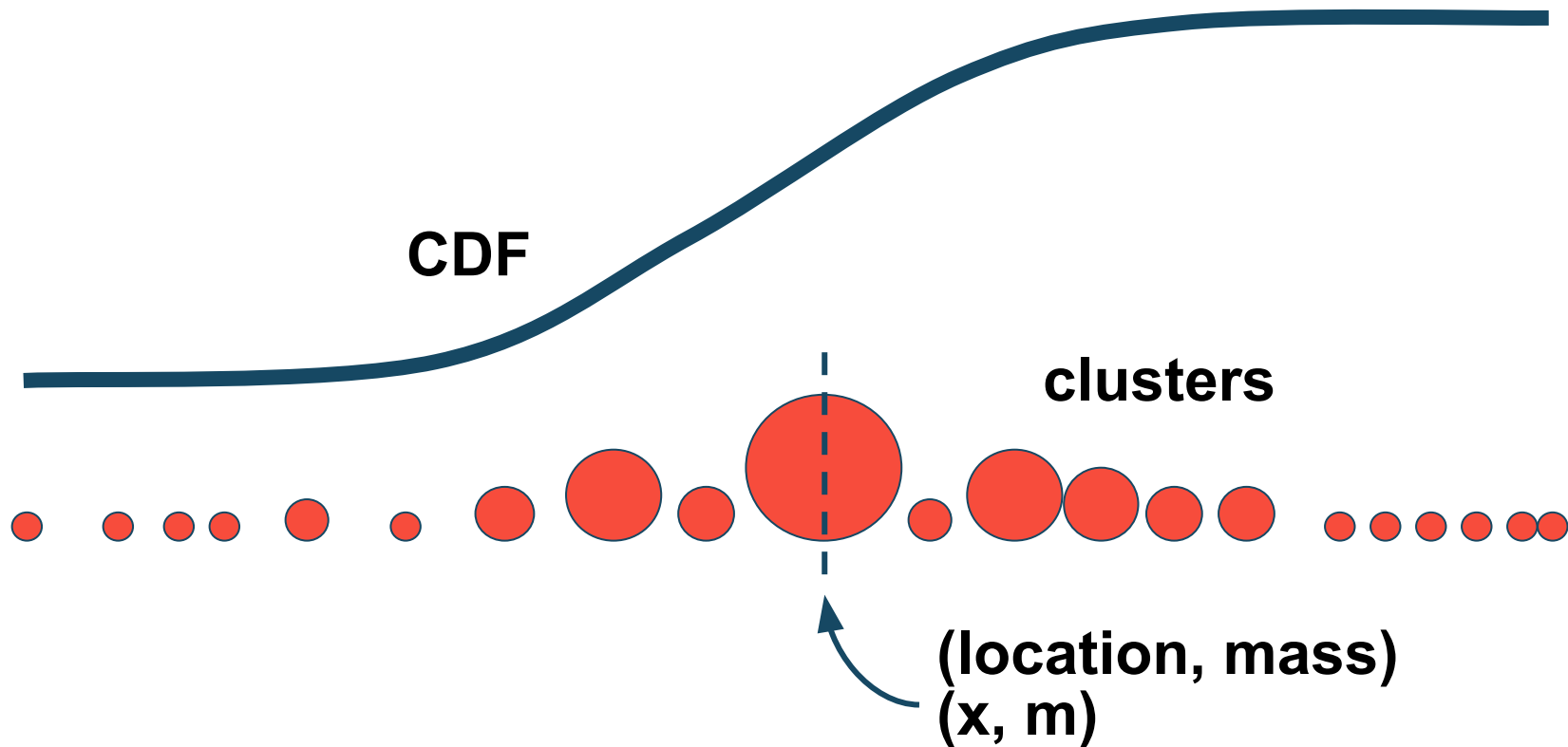
**DATA 2**



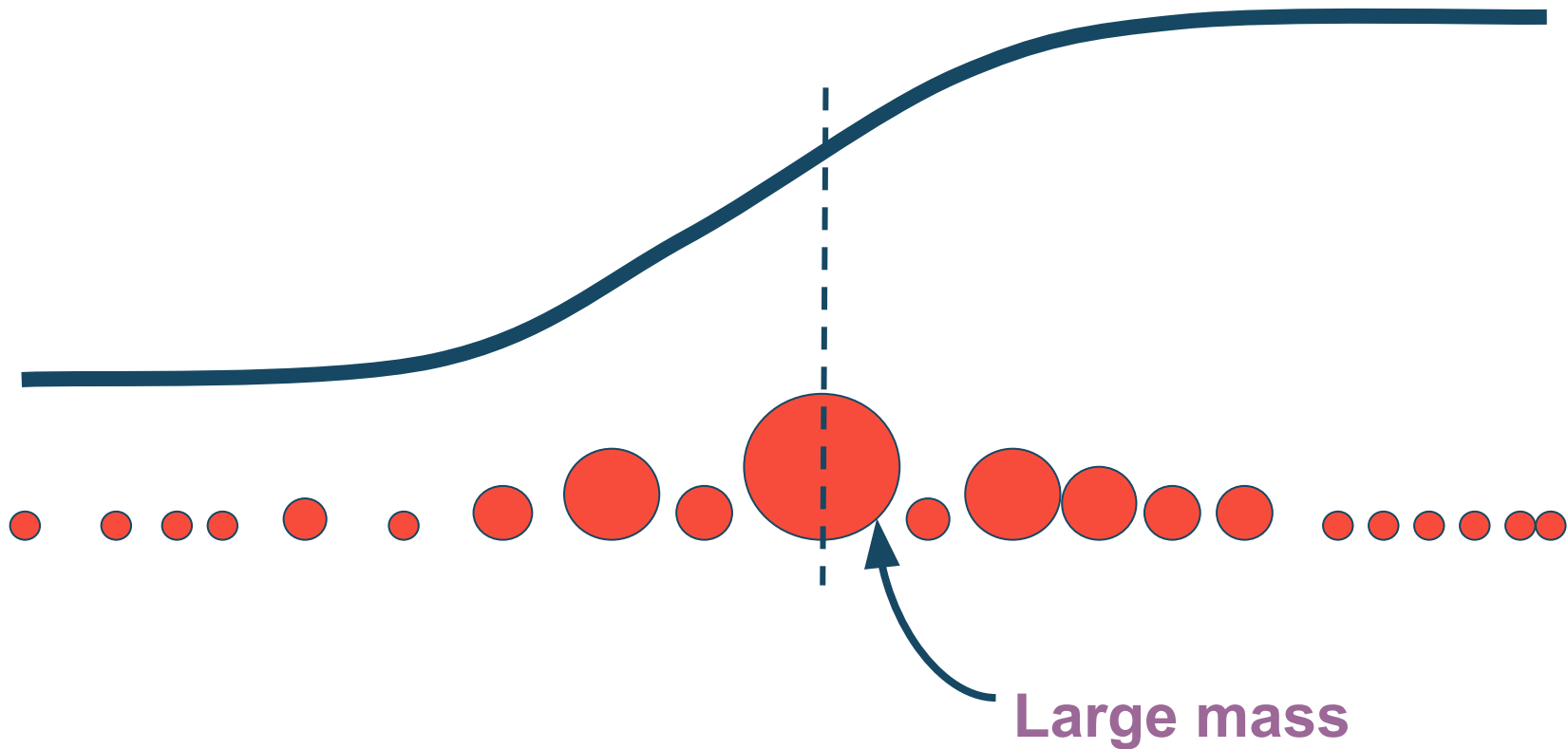
**DATA N**



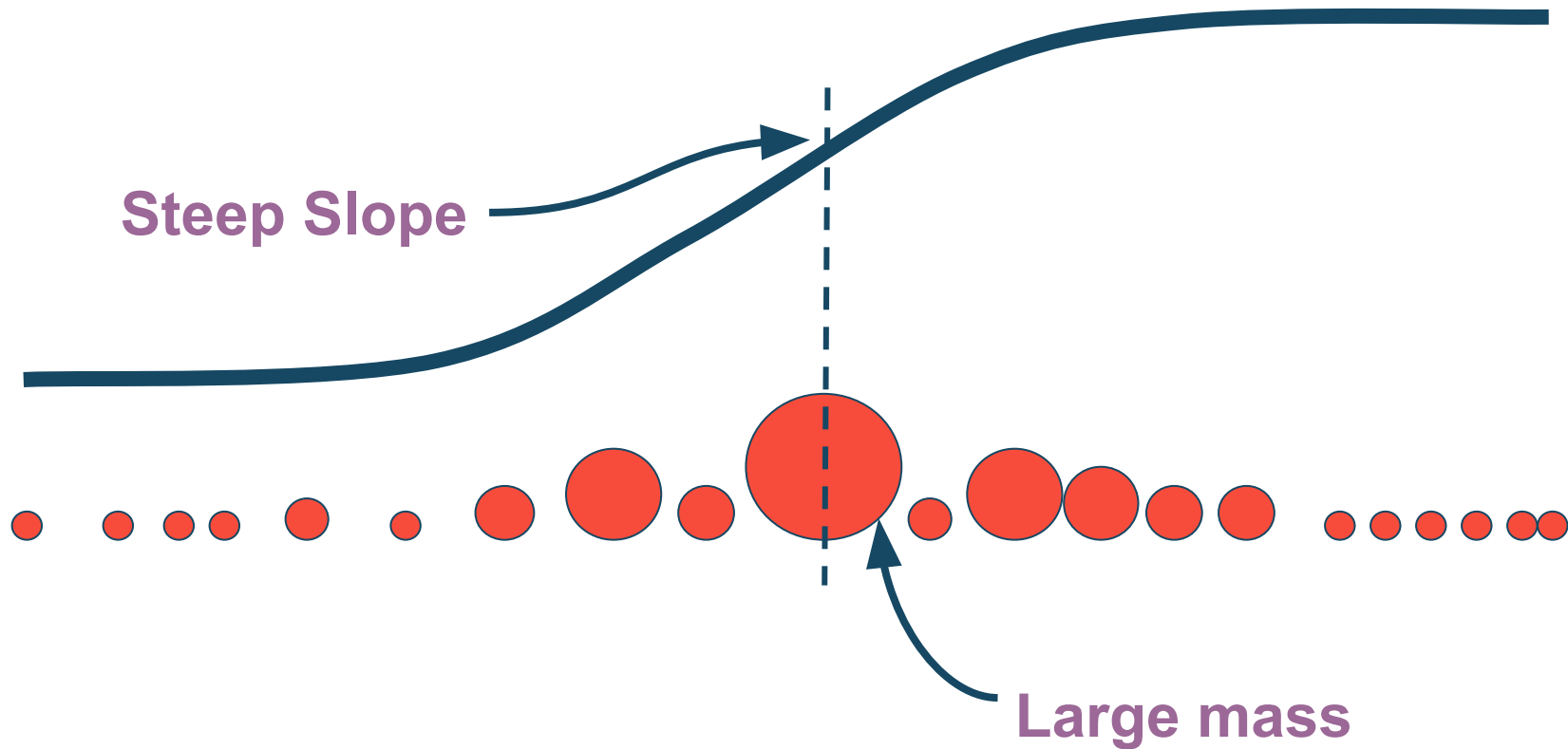
# Representation



# Representation

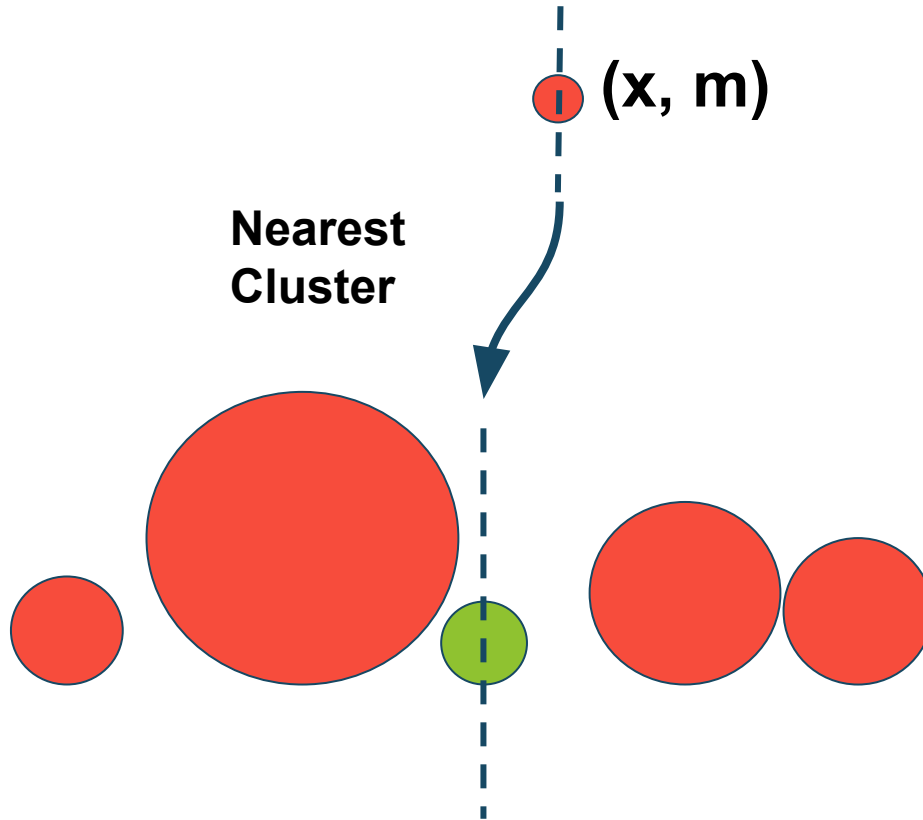


# Representation

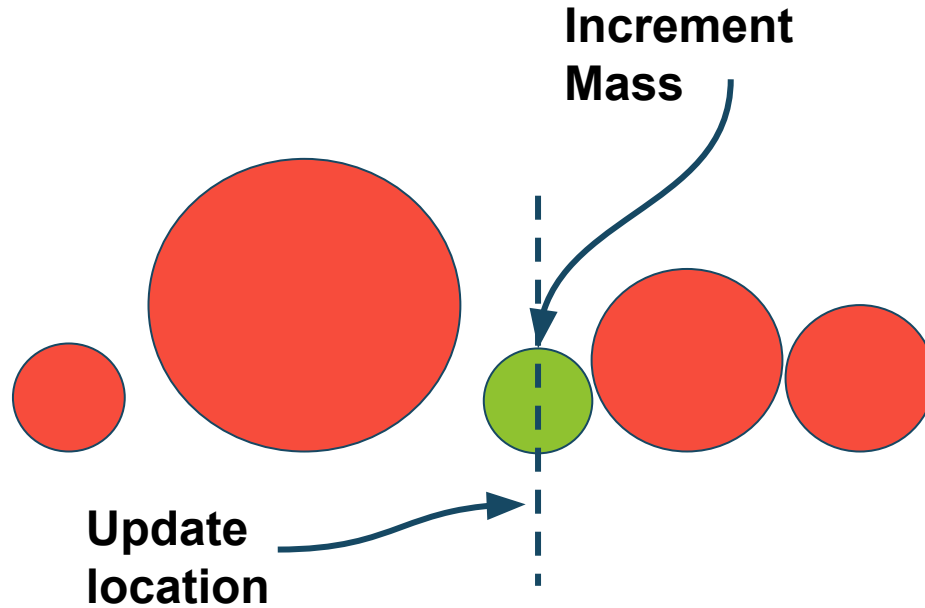




# Update

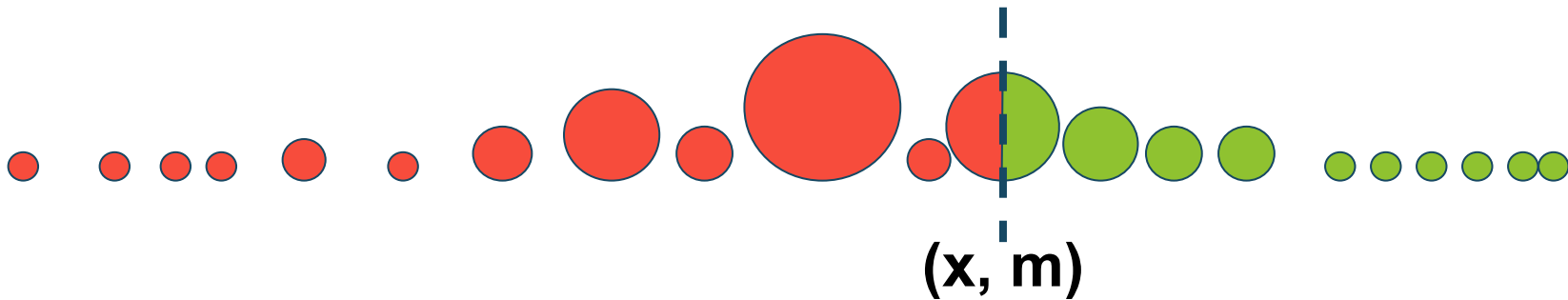


# Update



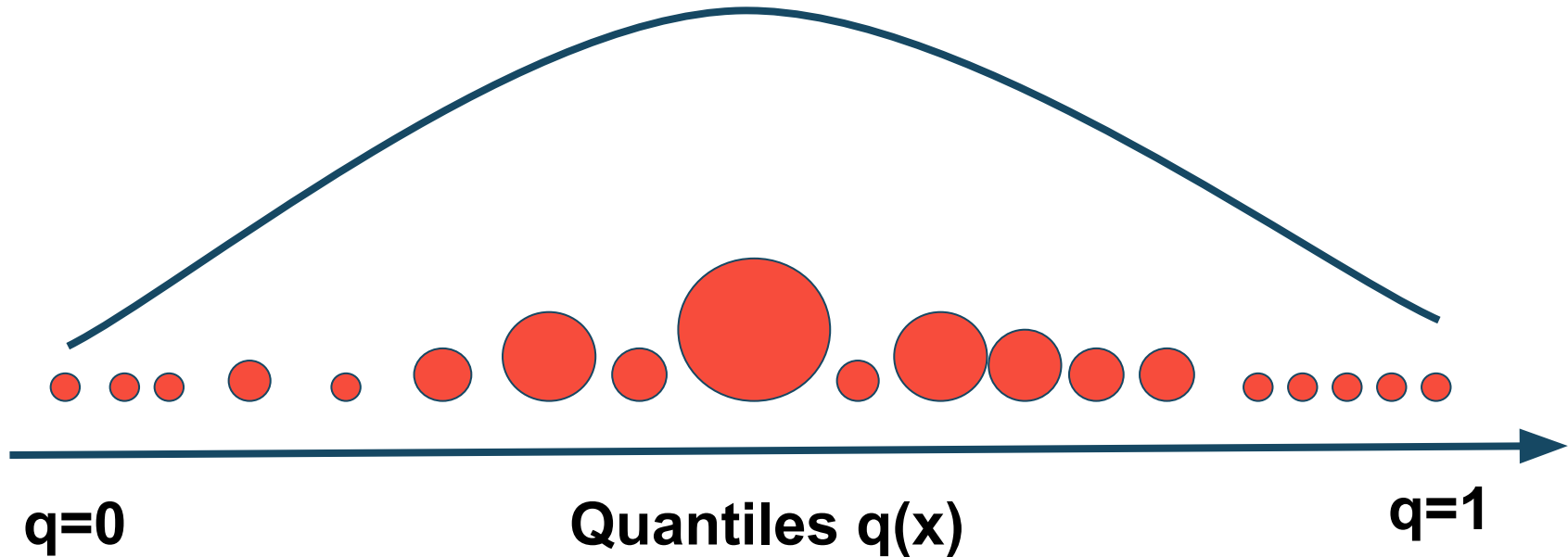
# Cluster Quantile

$$q(x) = \frac{\sum \text{red circles}}{\sum \text{red circles} + \sum \text{green circles}}$$

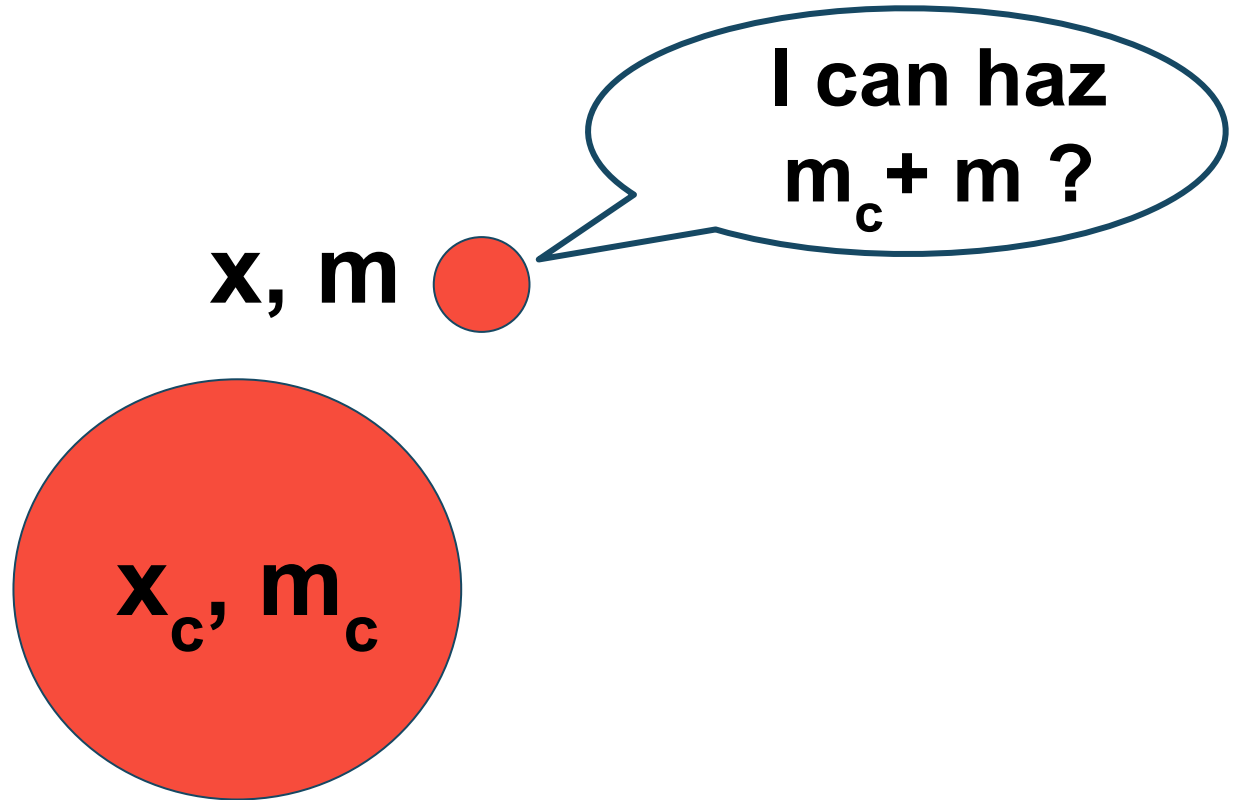


# Cluster Mass Bounds

$$B(x) = C \cdot M \cdot q(x) \cdot (1 - q(x))$$



# Bounds Force New Clusters



# Bounds Force New Clusters

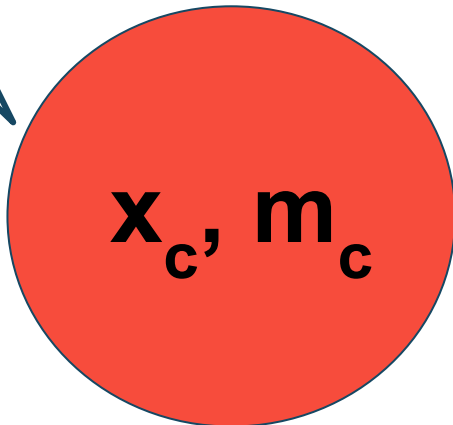
**Sorry!**

$$m_c + m > B(x_c)$$

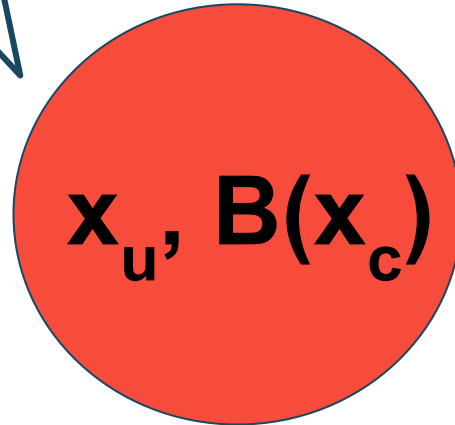
**$x, m$**



**$x_c, m_c$**



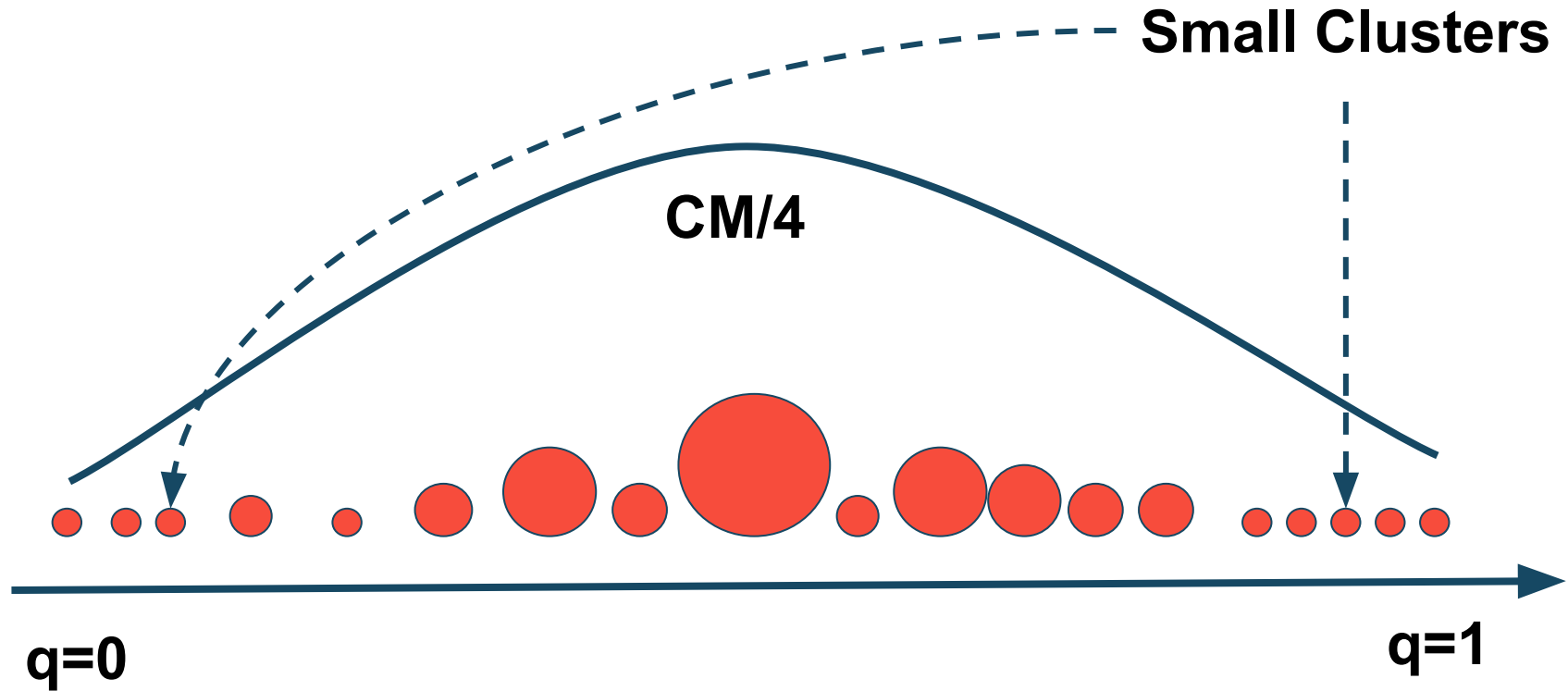
# Bounds Force New Clusters



$x, m_c + m - B(x_c)$

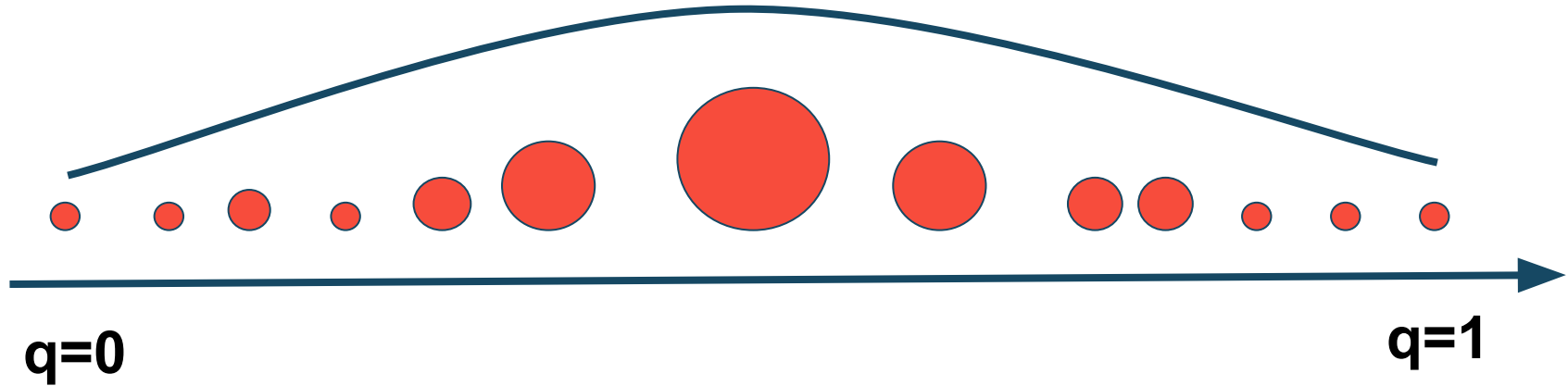


$$B(x) = C \cdot M \cdot \underline{q(x) \cdot (1 - q(x))}$$

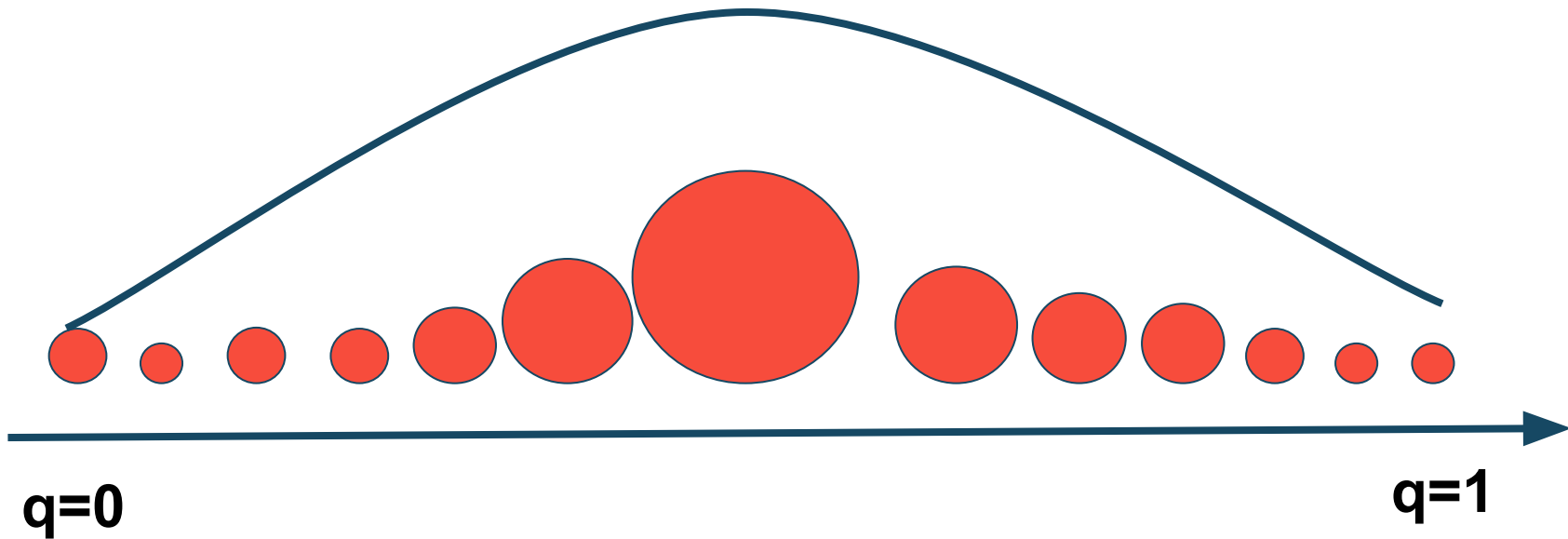




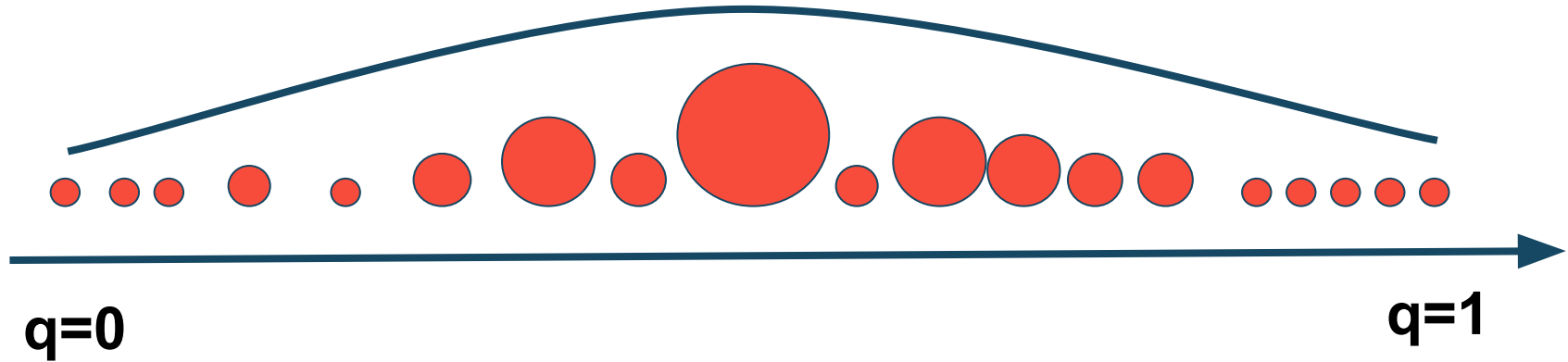
$$B(x) = C \cdot \underline{M} \cdot q(x) \cdot (1 - q(x))$$



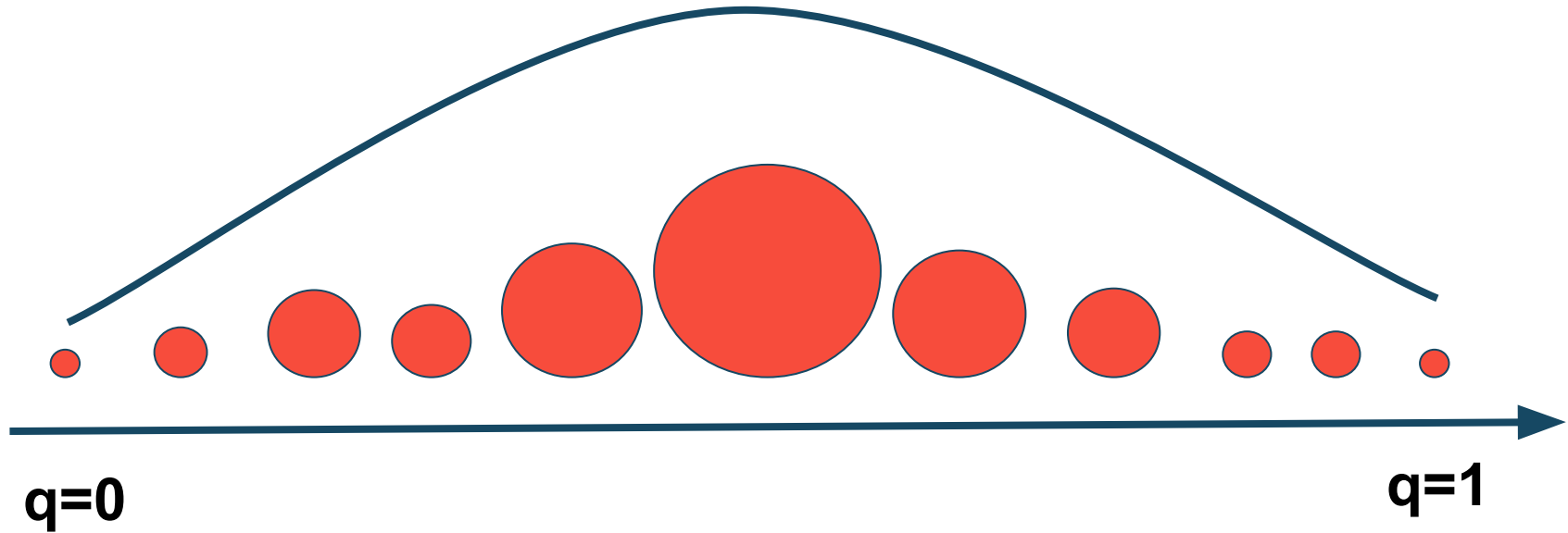
$$B(x) = C \cdot \underline{M} \cdot q(x) \cdot (1 - q(x))$$



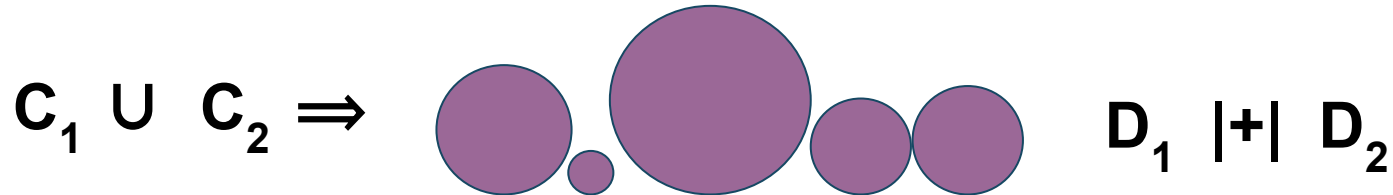
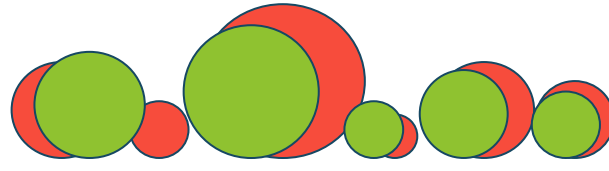
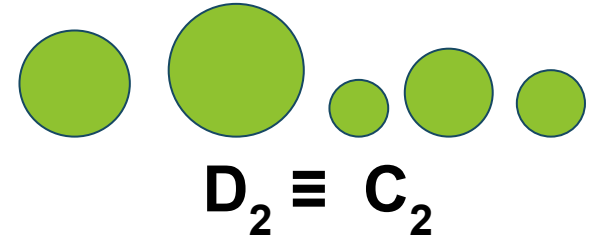
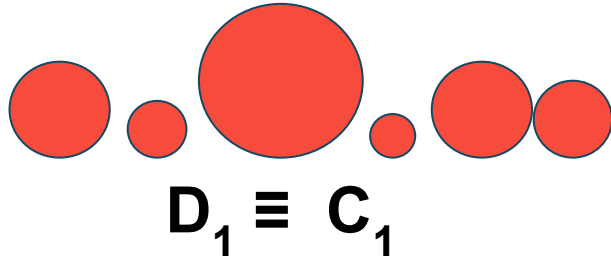
$$B(x) = \underline{C} \cdot M \cdot q(x) \cdot (1 - q(x))$$



$$B(x) = \underline{C} \cdot M \cdot q(x) \cdot (1 - q(x))$$



# T-Digests are Mergeable



# Flashback

**DATA 1**



**DATA 2**



**DATA N**



# Mergeable => Scale-Out

Data Partitions

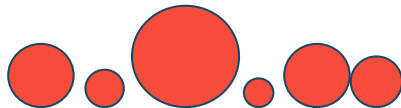
t-digests

P1

P2

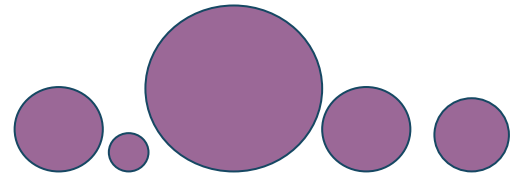
Pn

Map

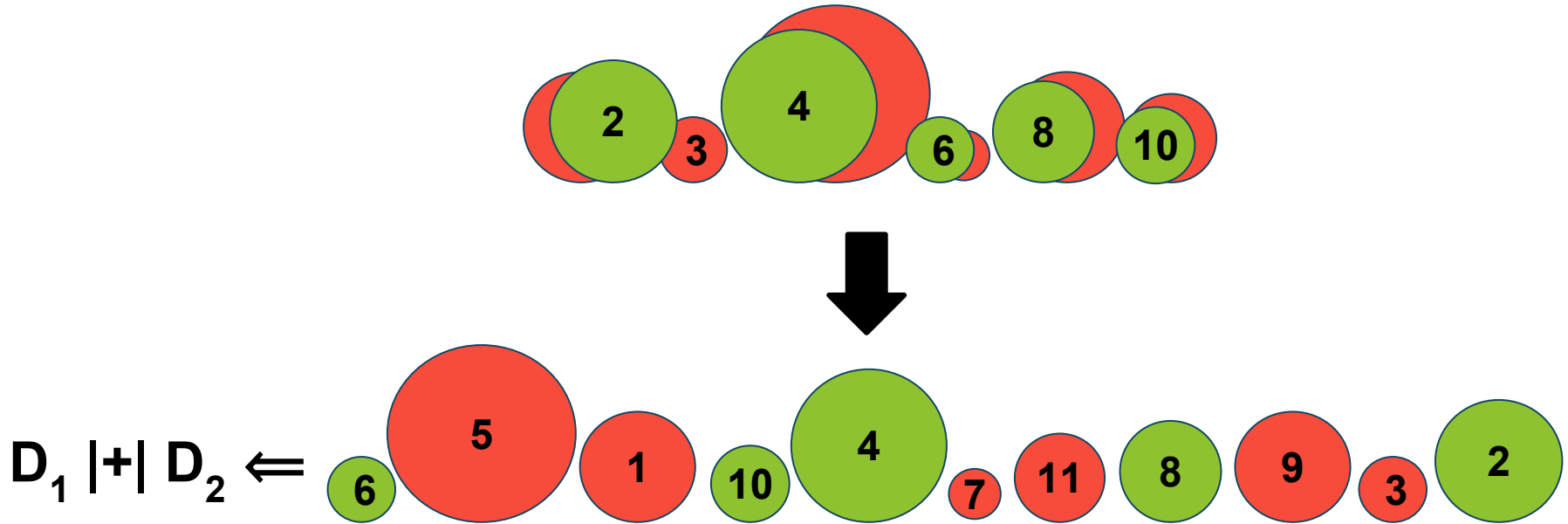


|+|

result

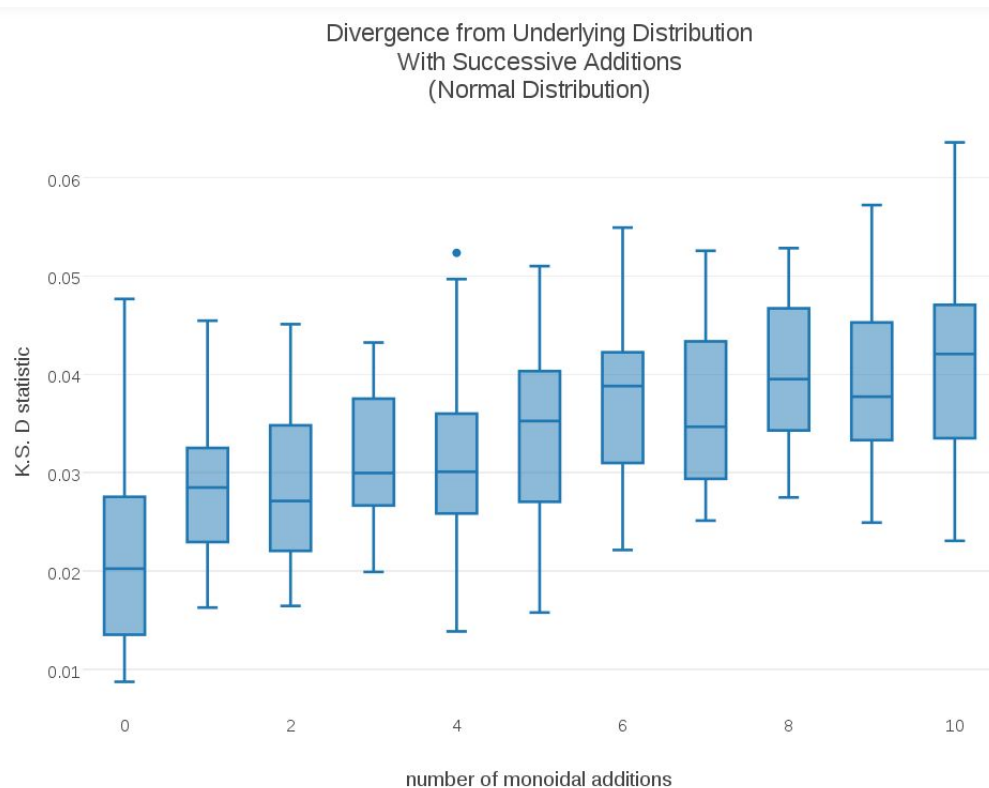


# Merge: Randomized Order

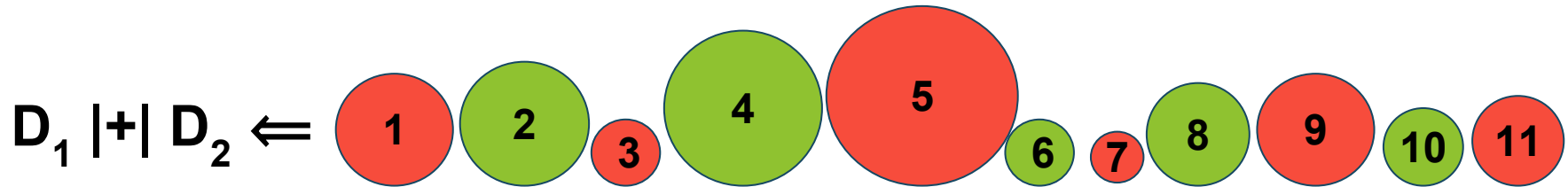
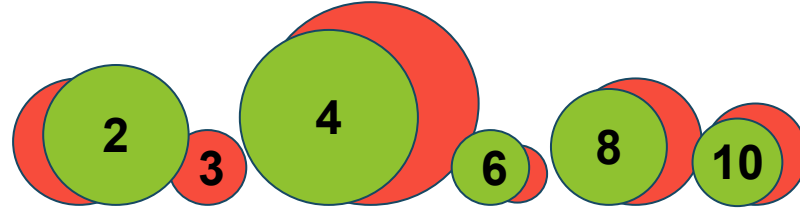




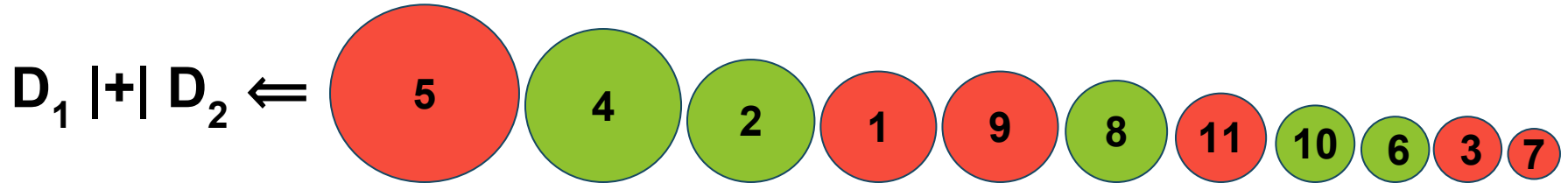
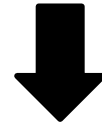
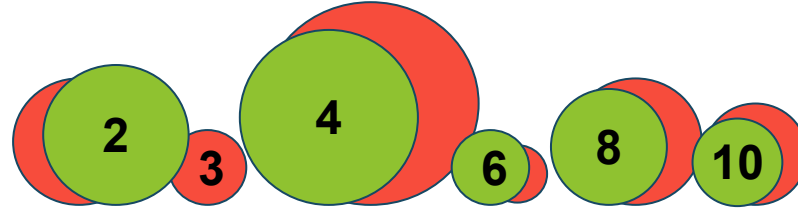
# Random Merging Diverges



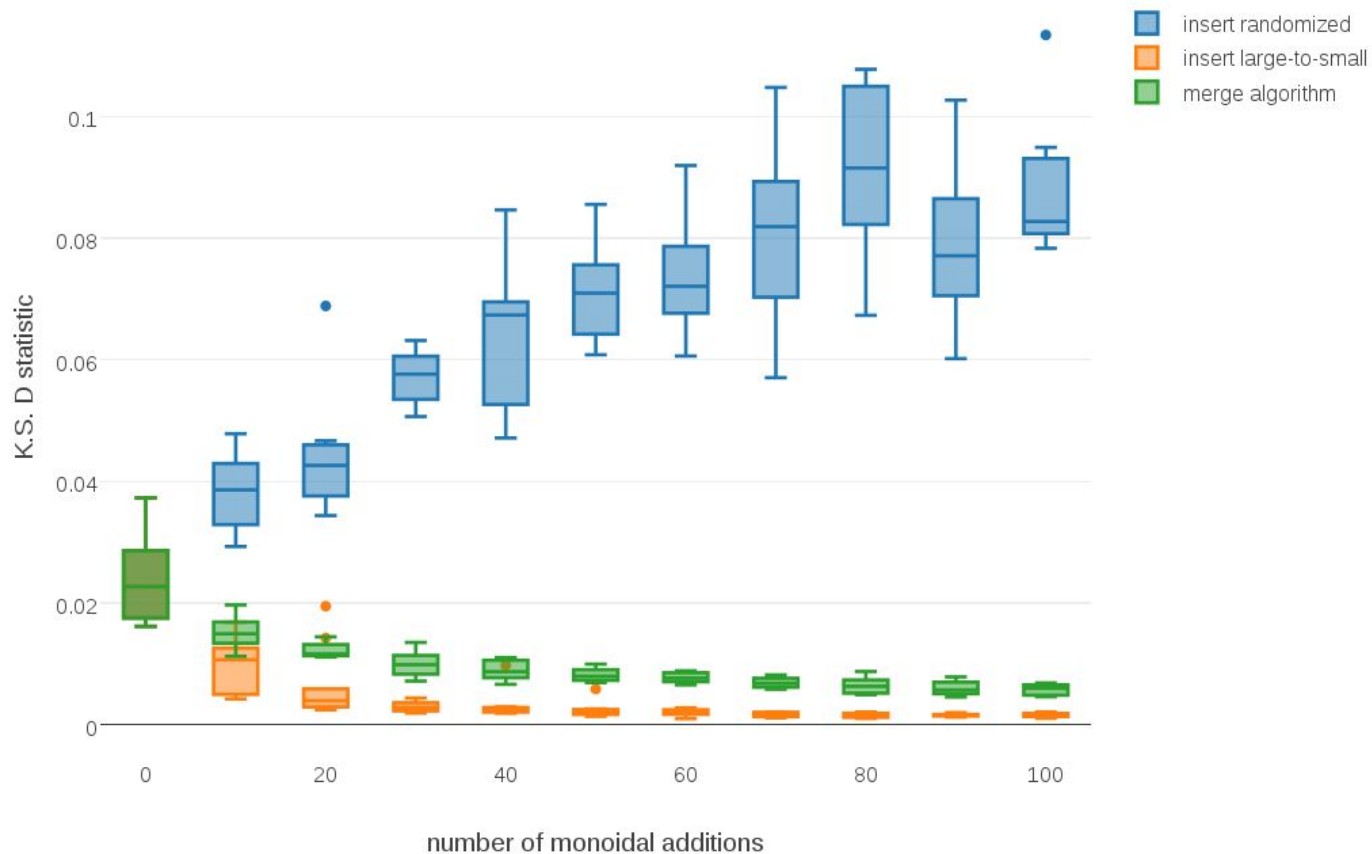
# Merge: Location Order



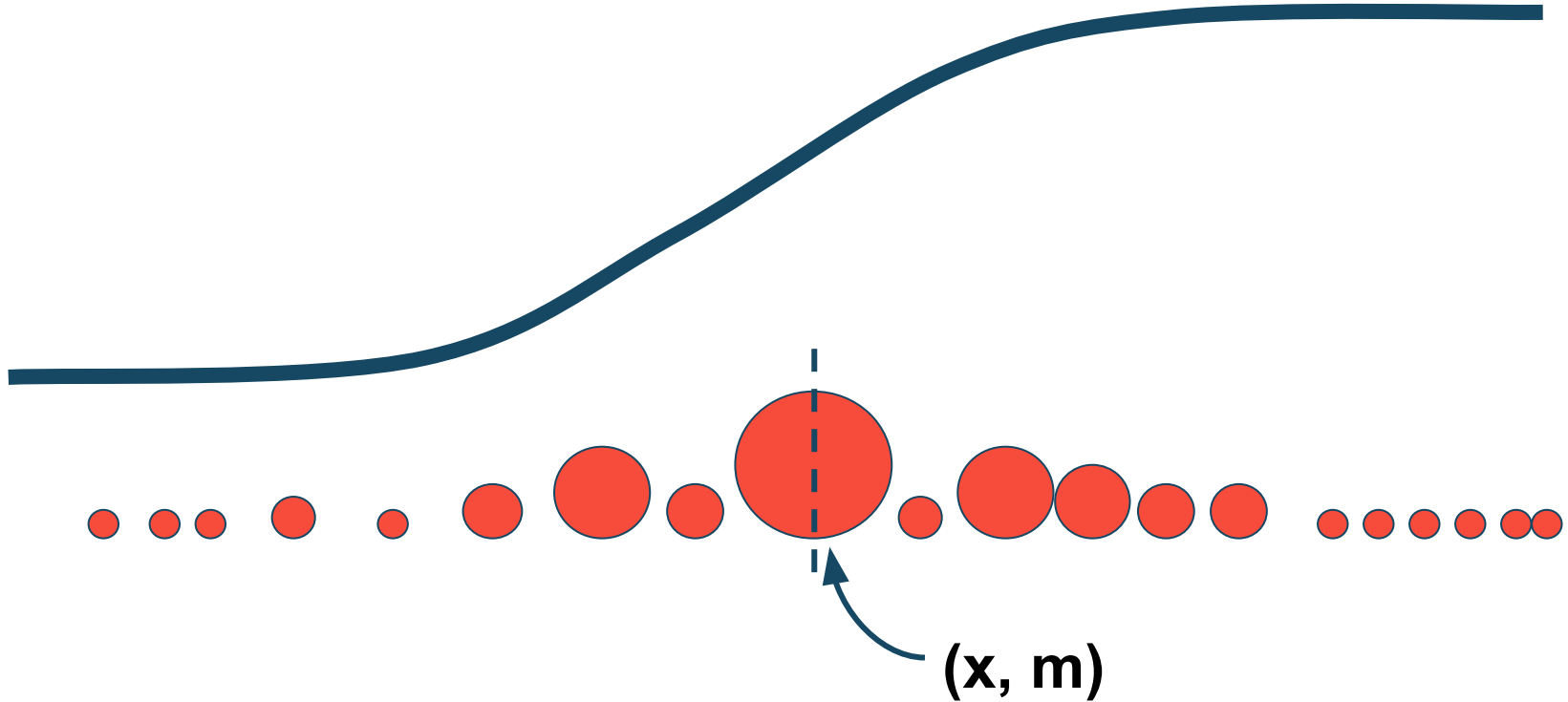
# Merge - Large to Small



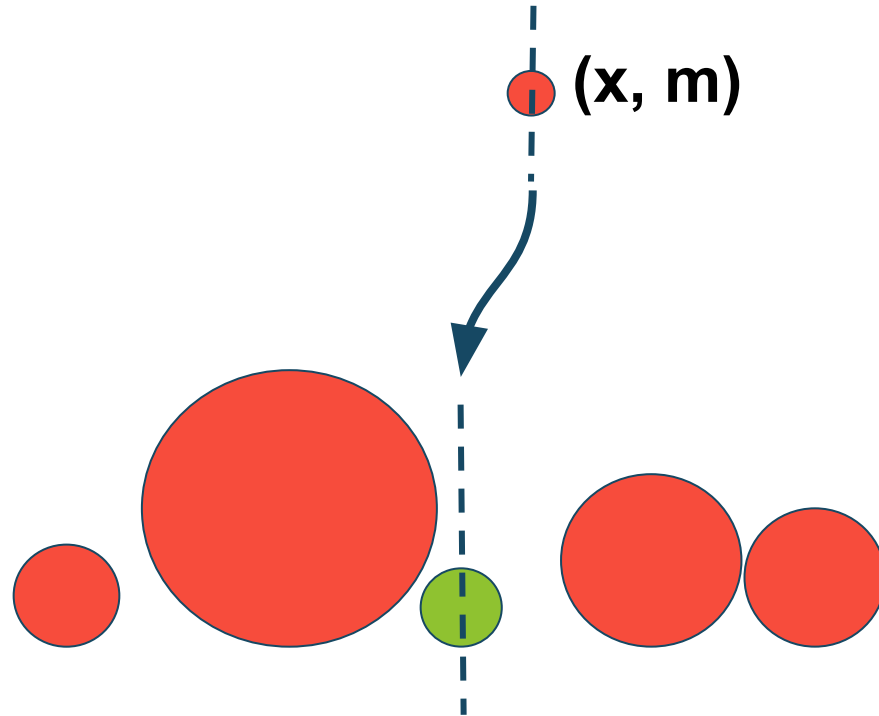
# Comparing Merge Definitions



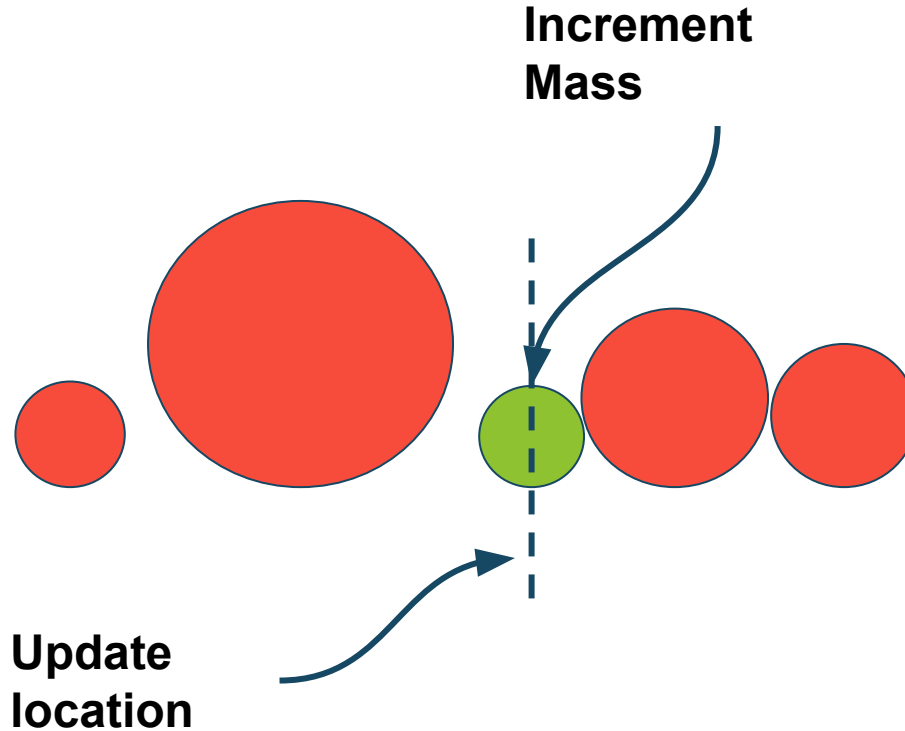
# Clusters Maintained in Order



# Query the Nearest Cluster

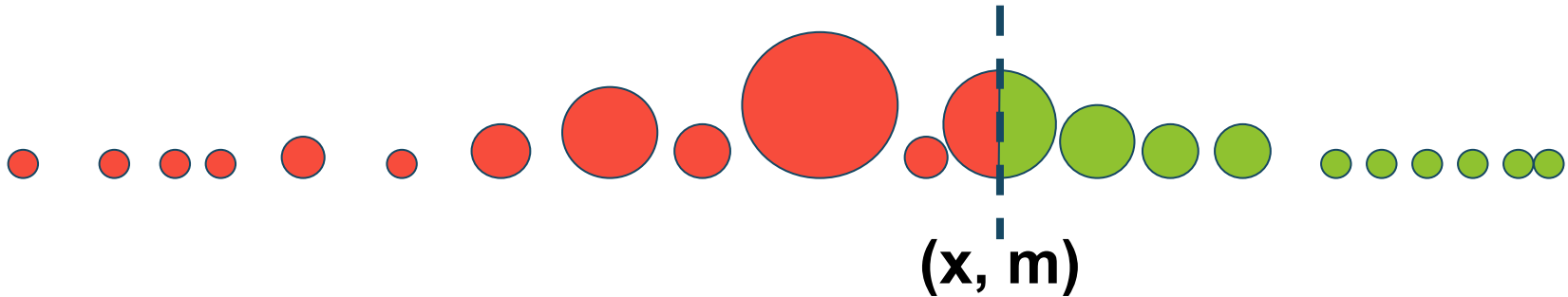


# Insert and Update Clusters



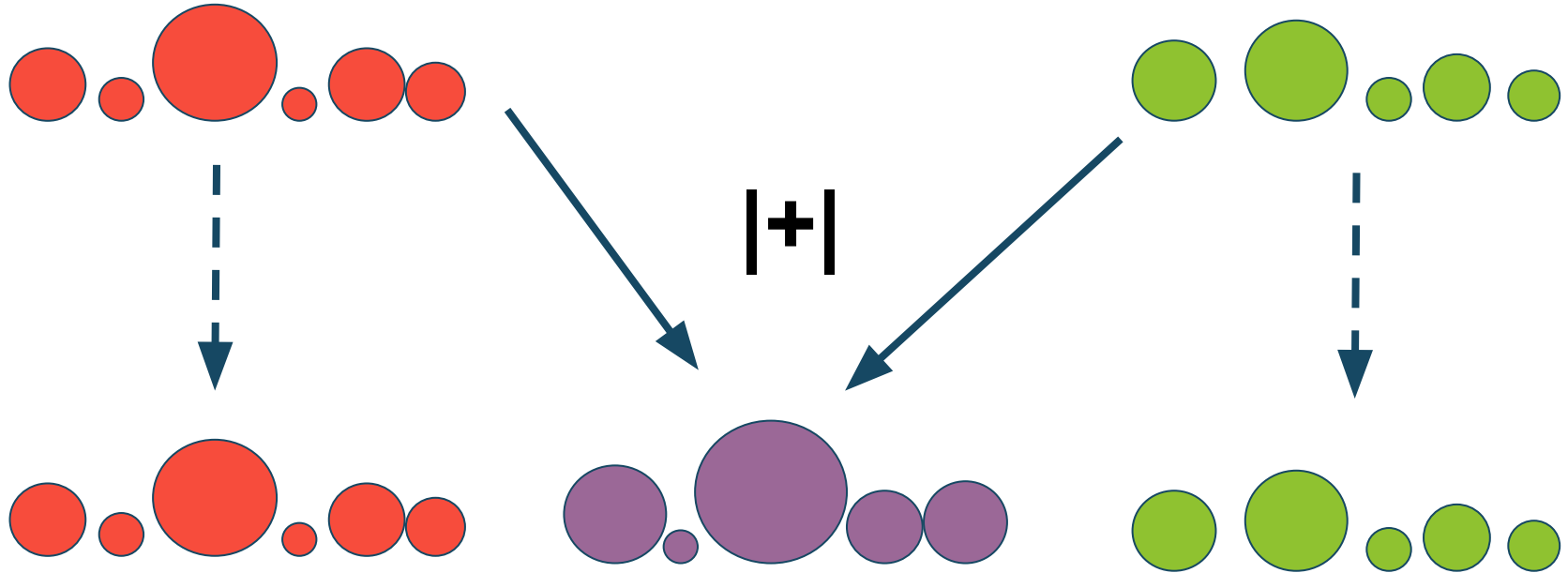
# Compute “Prefix Sums”

$$q(x) = \frac{\sum \text{red circles}}{\sum \text{red circles} + \sum \text{green circles}}$$





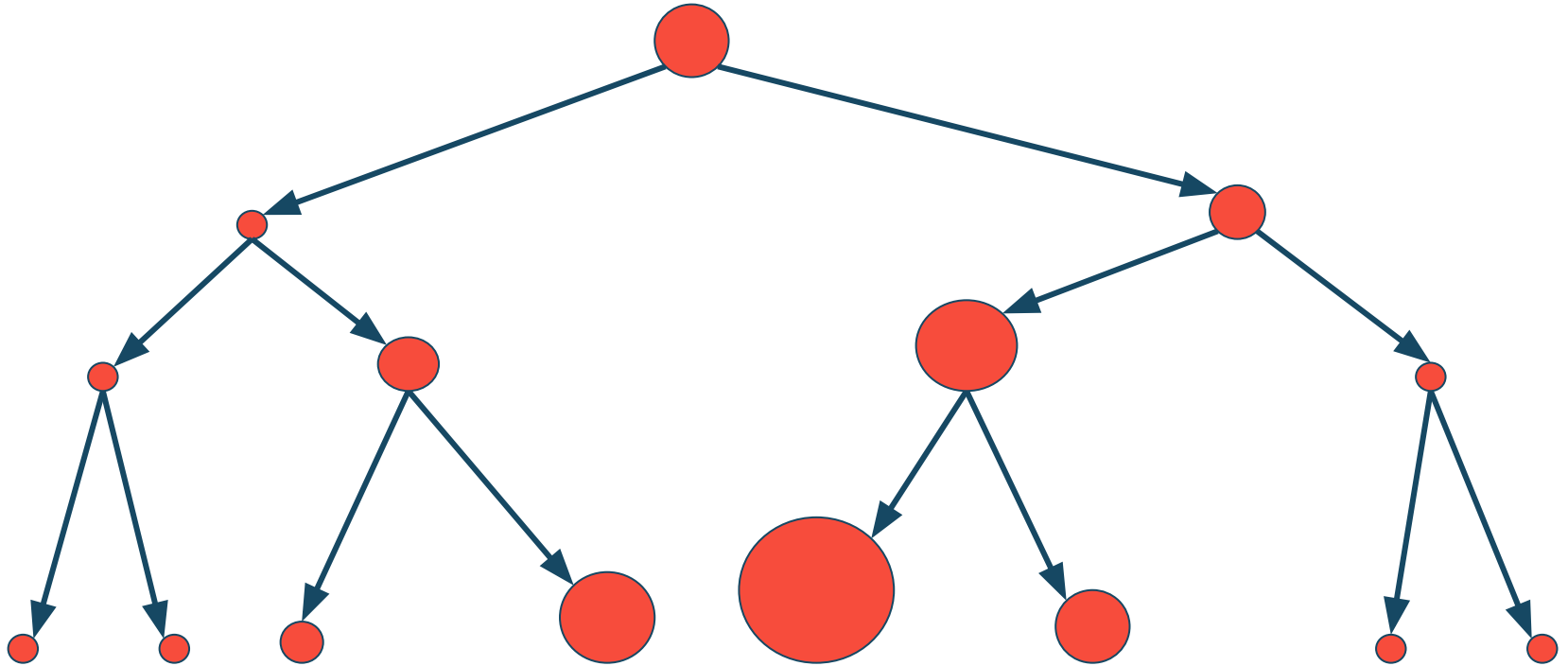
# Immutable Data Structure



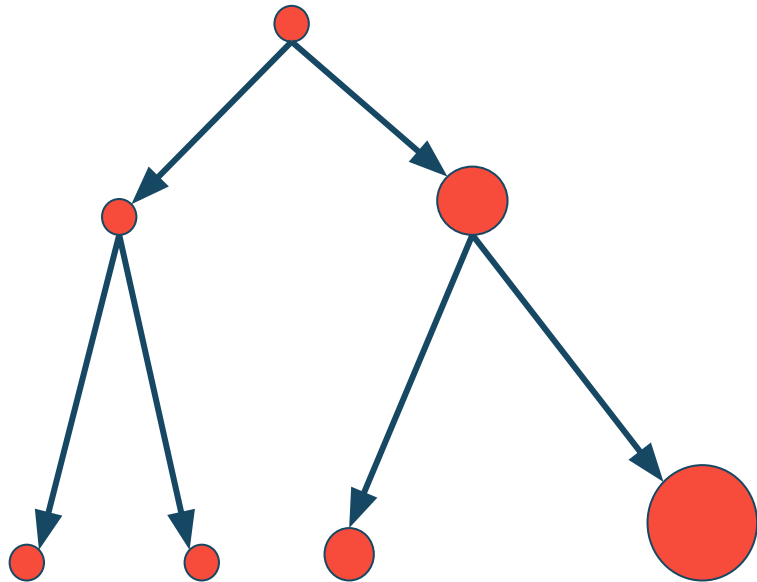
**Fast!**

$O(\log n)$

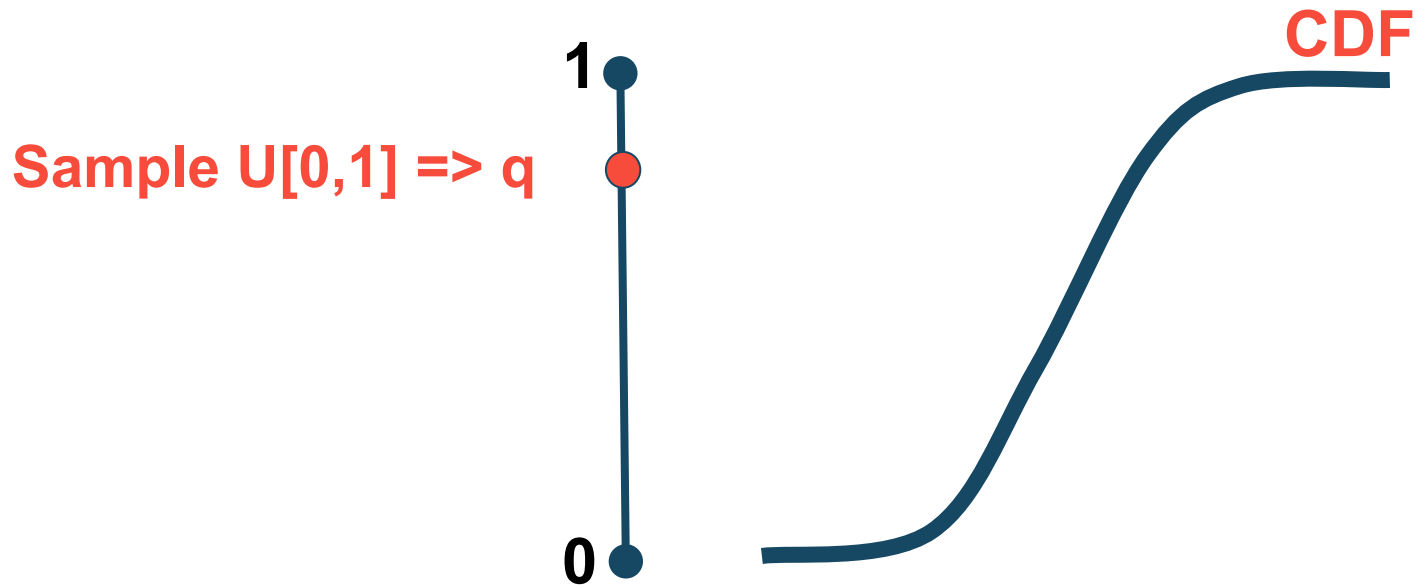
# Balanced Immutable Tree



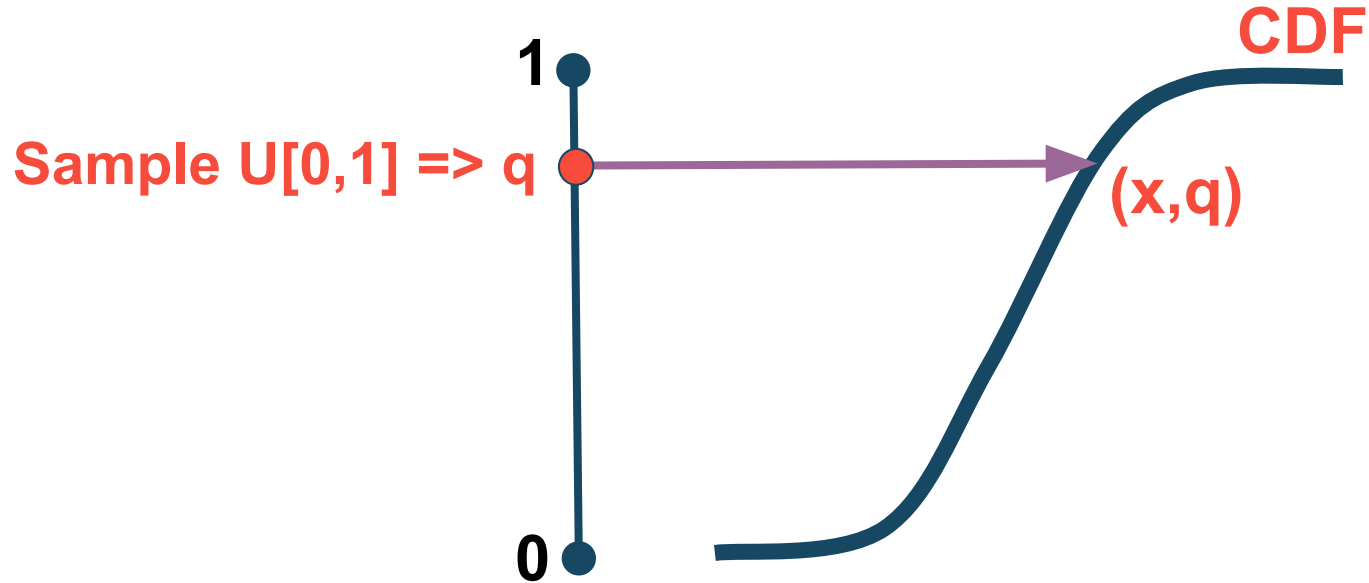
# Explore



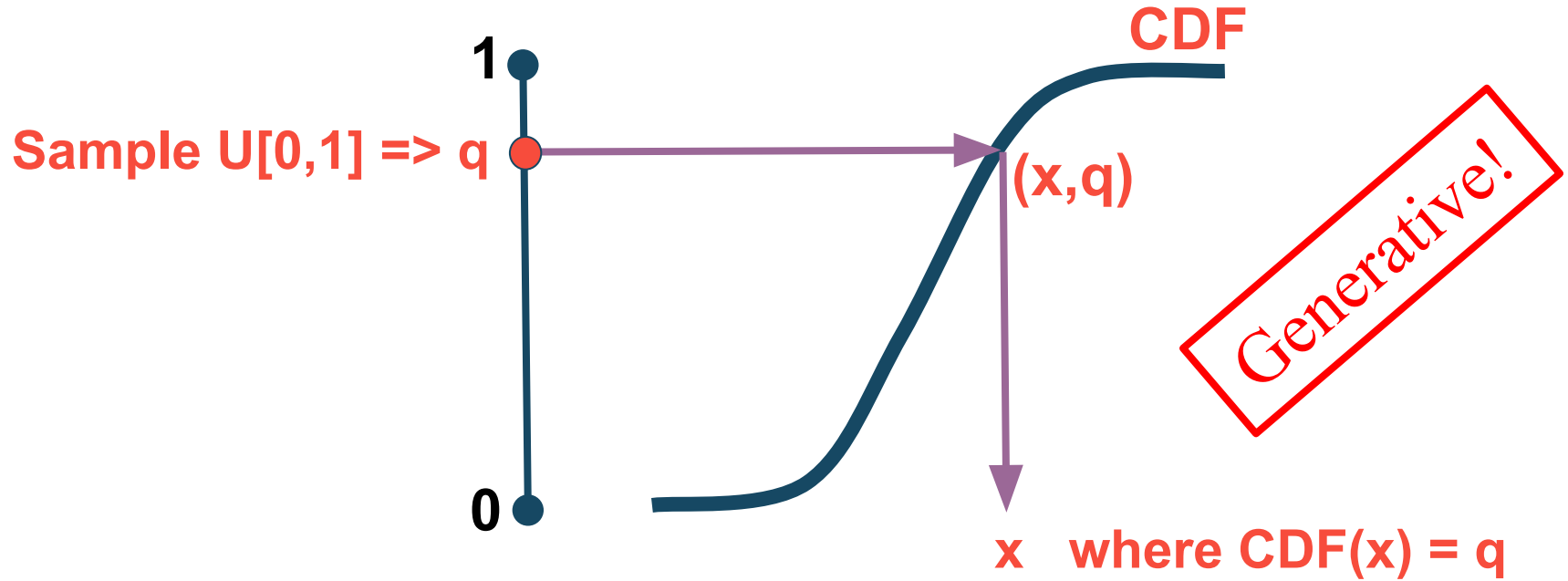
# Inverse Transform Sampling (ITS)



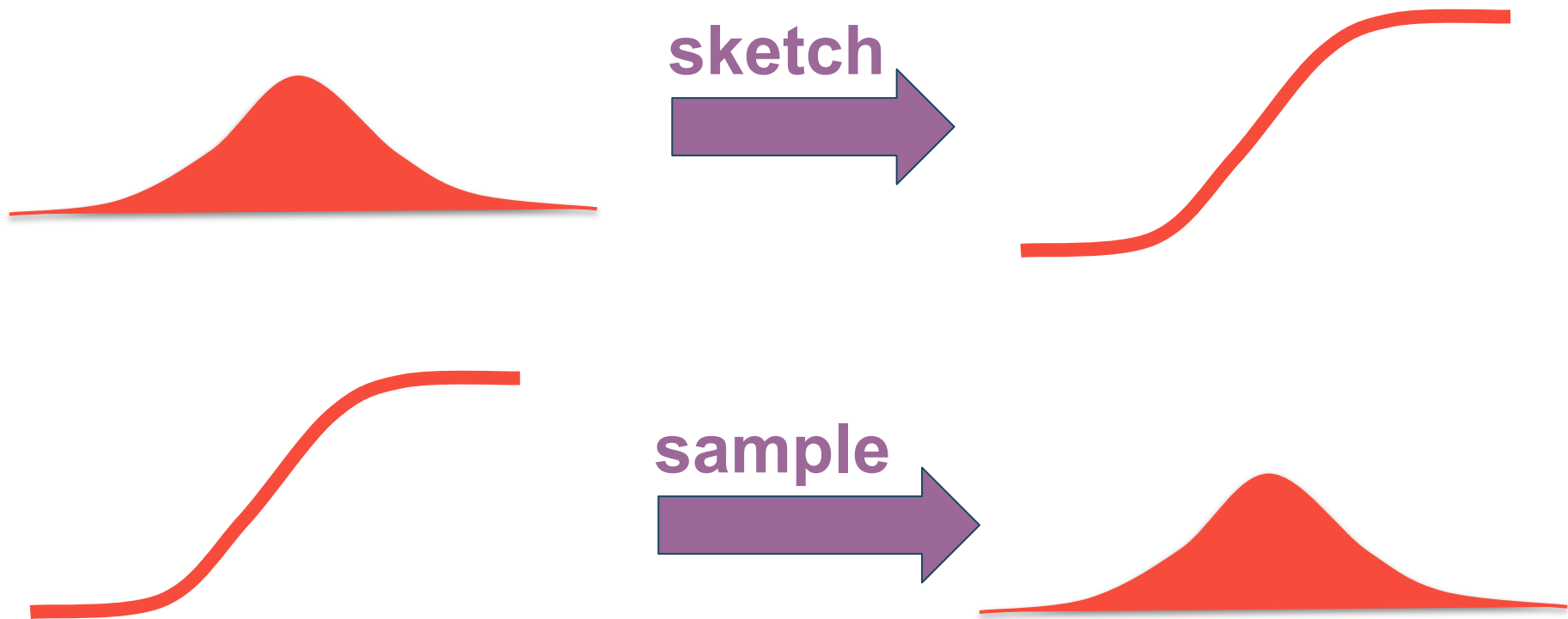
# Inverse Transform Sampling (ITS)



# Inverse Transform Sampling (ITS)

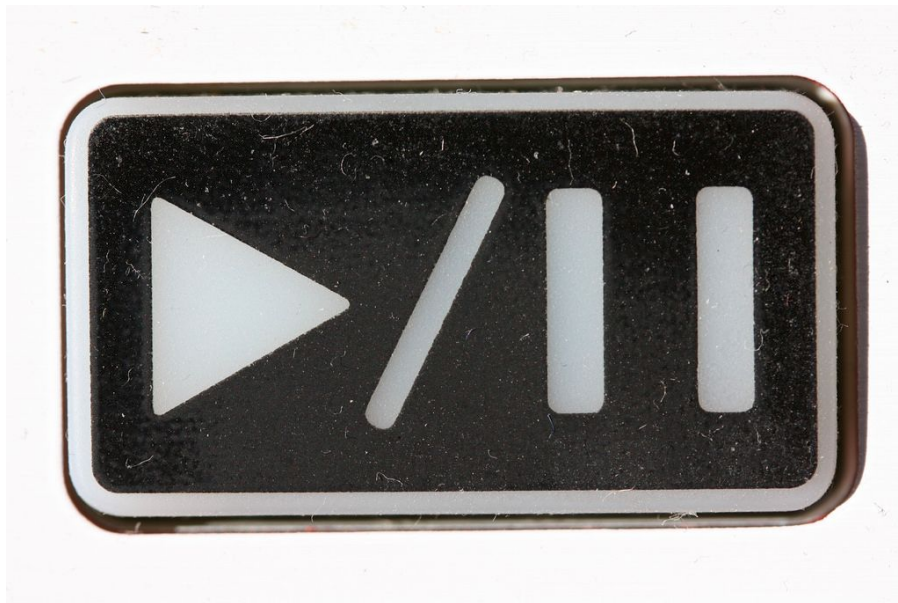


# Generative Sampling





# Demo



# Explore



[T-Digests and Feature Importance for Spark](#)



[Feature Reduction With T-Digests & RF](#)



[Data Science with Generative T-Digests](#)



[Probabilistic Structures for Scalable Computing](#)



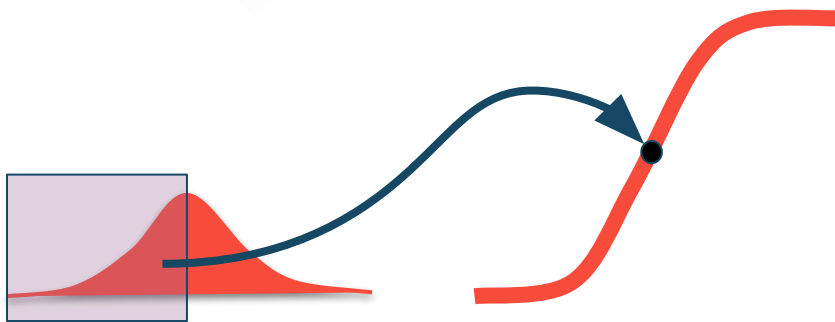
[Demo Notebook for This Talk](#)

# Get This Deck

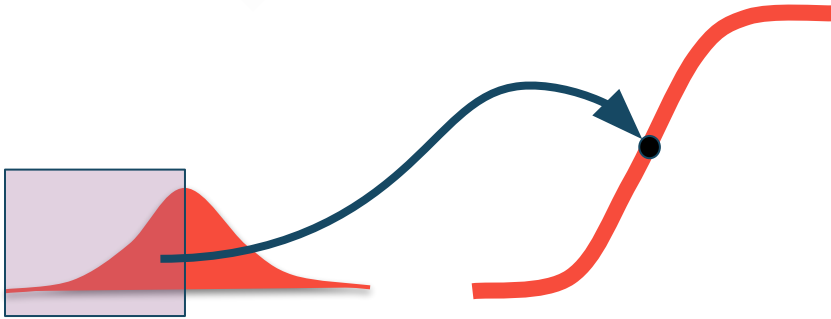
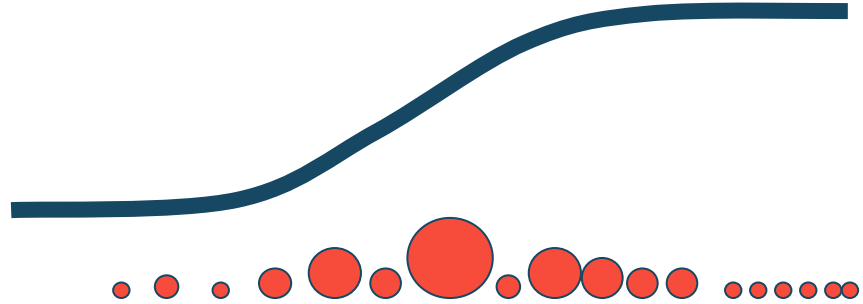
# Review



# Review



# Review



# Review

