Introduction to Histograms

Presented By:

Laukik Chitnis
(lchitnis@cise.ufl.edu)

Introduction to Histograms

- This presentation is mostly based on the following work done by Yannis Ioannidis and Viswanath Poosala {yannis,viswanath}@cs.wisc.edu
  - Histogram Based Solutions to Diverse Database Estimation Problems (1995)
  - Improved Histograms for Selectivity Estimation of Range Predicates (1996) (with IBM Almaden research guys Peter Haas and Eugene Shekita)
  - History of Histograms (presented by Yannis)

Motivation and Expected Features

- Why Histograms?
  - Size estimates needed for
    - Estimating the costs of access plans
  - Query profiling for user feedback
  - Load balancing for parallel join execution
- Expected features
  - Should produce estimates with small errors
  - Inexpensive to construct, use and maintain
  - Utility across diverse estimation problems

Data Distribution

- Histograms as approximations of data distribution
- Data distribution is a set of (attribute value, frequency) pairs

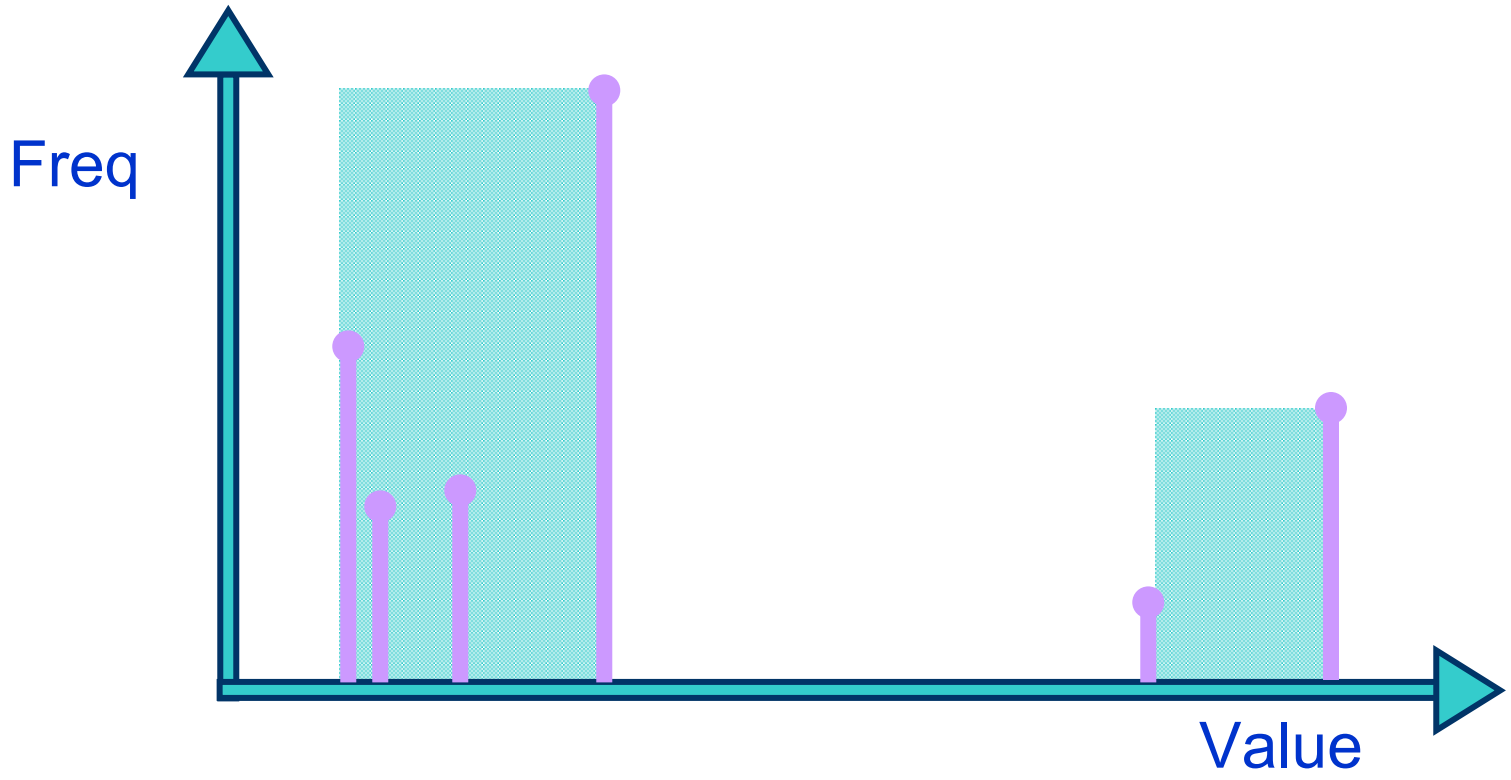
Name	Salary	Department
Zeus	100K	General Management
Poseidon	80K	Defense
Pluto	80K	Justice
Aris	50K	Defense
Ermis	60K	Commerce
Apollo	60K	Energy
Hefestus	50K	Energy
Hera	90K	General Management
Athena	70K	Education
Aphrodite	60K	Domestic Affairs
Demeter	60K	Agriculture
Hestia	50K	Domestic Affairs
Artemis	60K	Energy

Department	Frequency
General Management	2
Defense	2
Education	1
Domestic Affairs	2
Agriculture	1
Commerce	1
Justice	1
Energy	3

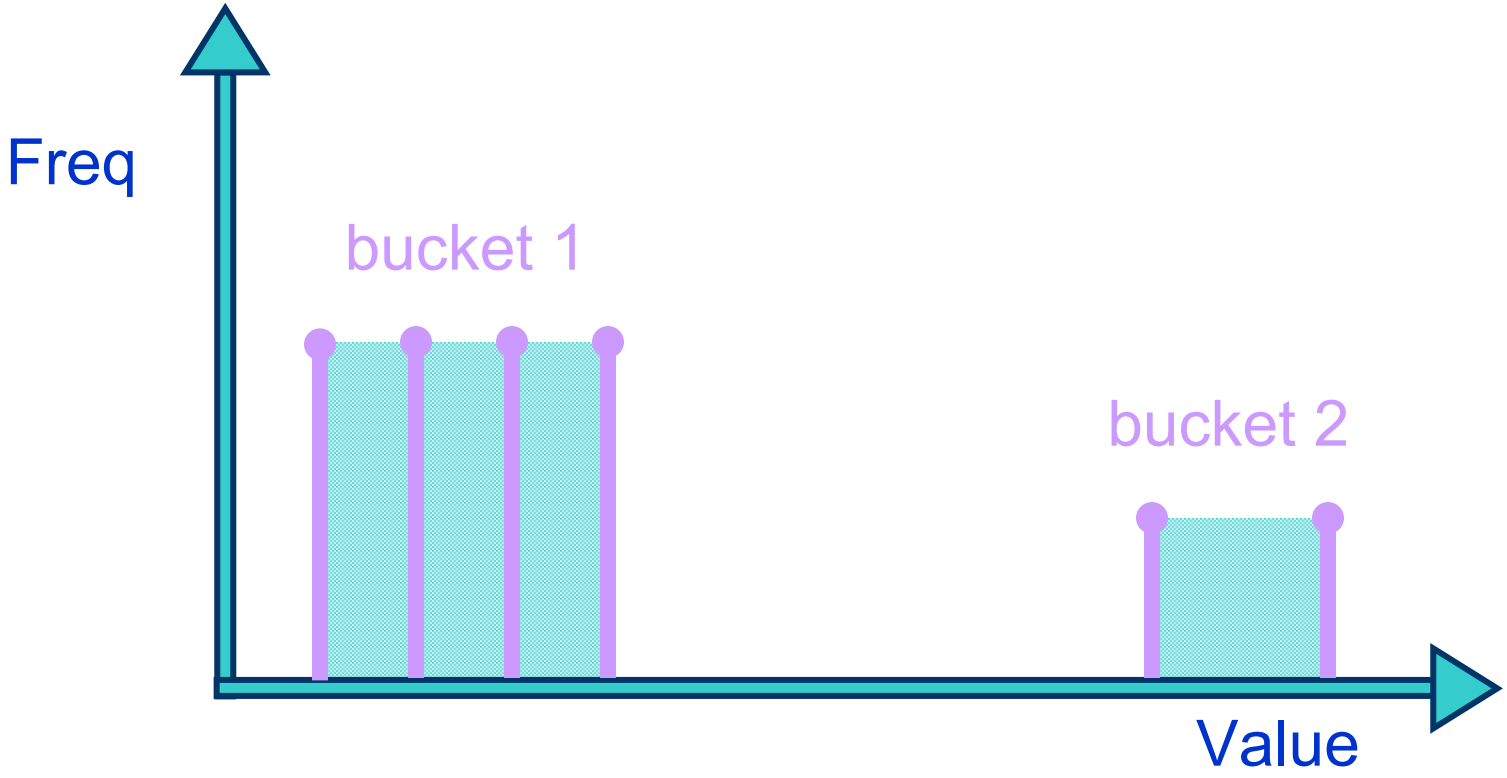
Data Distribution

- Set of (attribute value, frequency) pairs
- This data distribution has all the information required to answer query (count, join, aggregate,..)
- But, it is too bulky!
- So, we “approximate” it to a histogram!

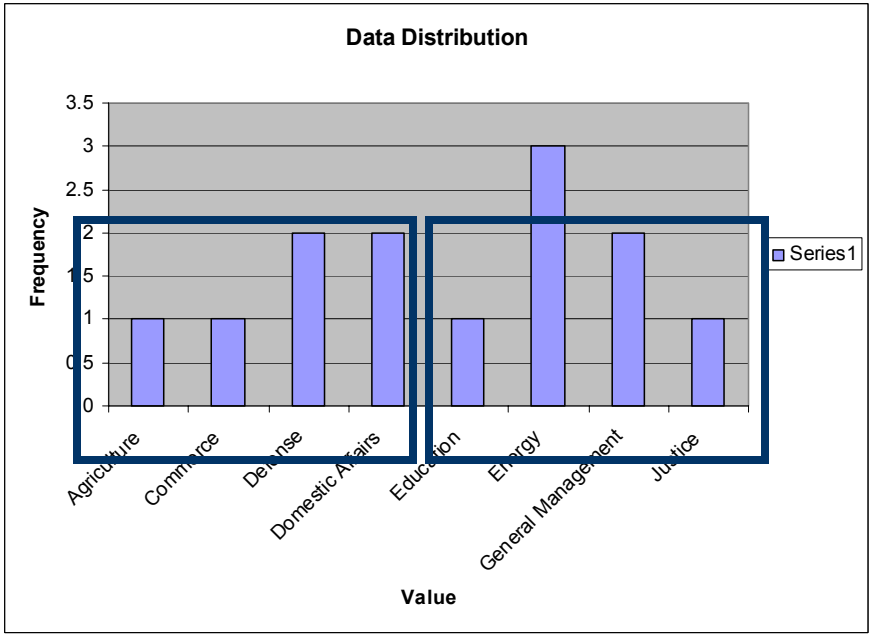
From Data Distribution to Histogram

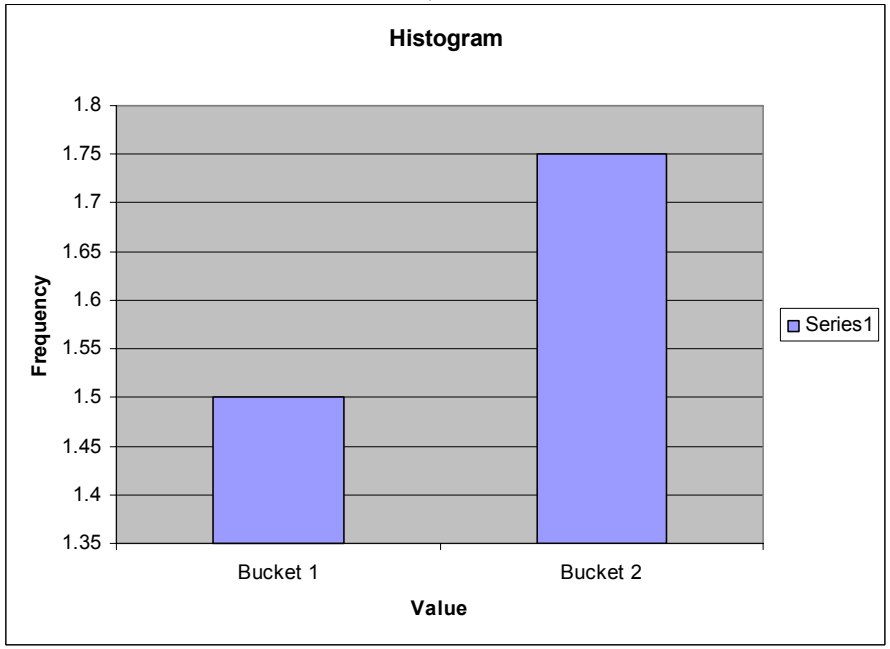


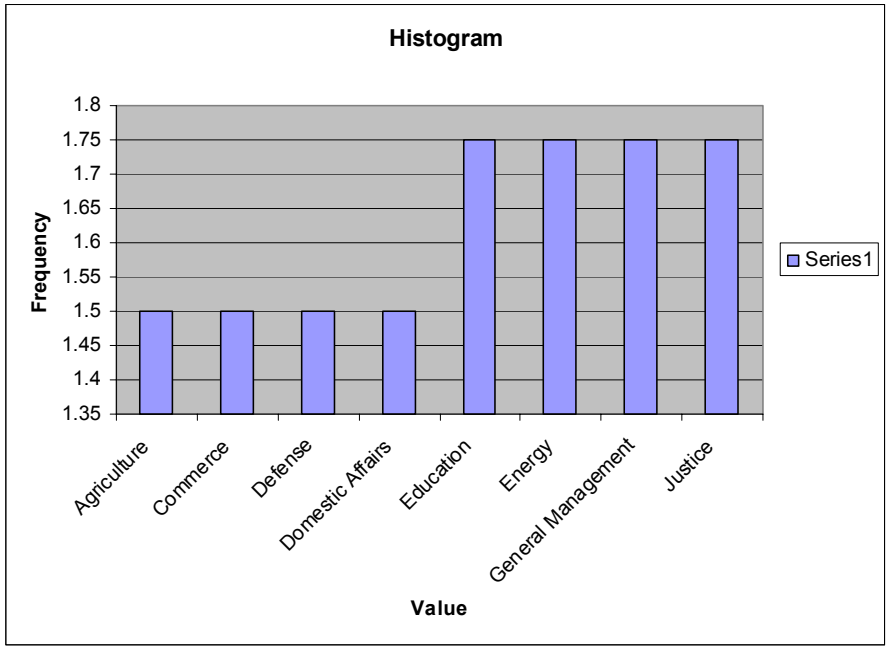
From Data Distribution to Histogram



Department	Histogram H1	
	Frequency in Bucket	Approximate Frequency
Agriculture	1	1.5
Commerce	1	1.5
Defense	2	1.5
Domestic Affairs	2	1.5
Education	1	1.75
Energy	3	1.75
General Management	2	1.75
Justice	1	1.75



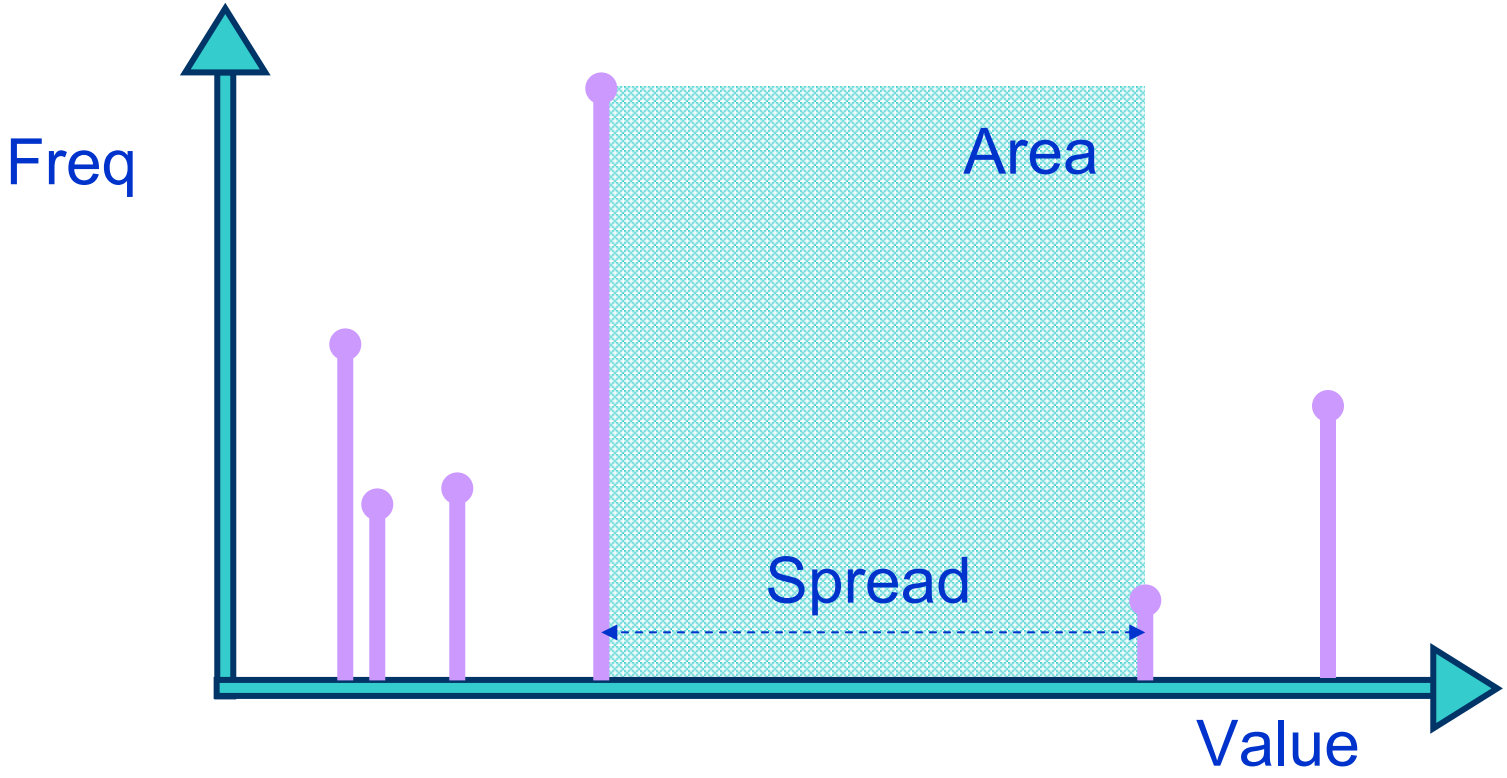




Definitions

- equi-width histograms
  - A histogram type wherein the ‘spread’ of each bucket is same.
- Spread S and Area A
  - $S_i = V_{i+1} – V_i$
  - $A_i = f_i * S_i$
  - where V_i is an attribute value f_i is its frequency

Spread S and Area A



Equi-width Histograms

- Divide the value axis into buckets of equal ‘width’ (range equalized)
- **Advantages**:
  - natural order of the attribute-values is preserved.
  - Light on storage requirements
- **Example**: Count of tuples having x<5

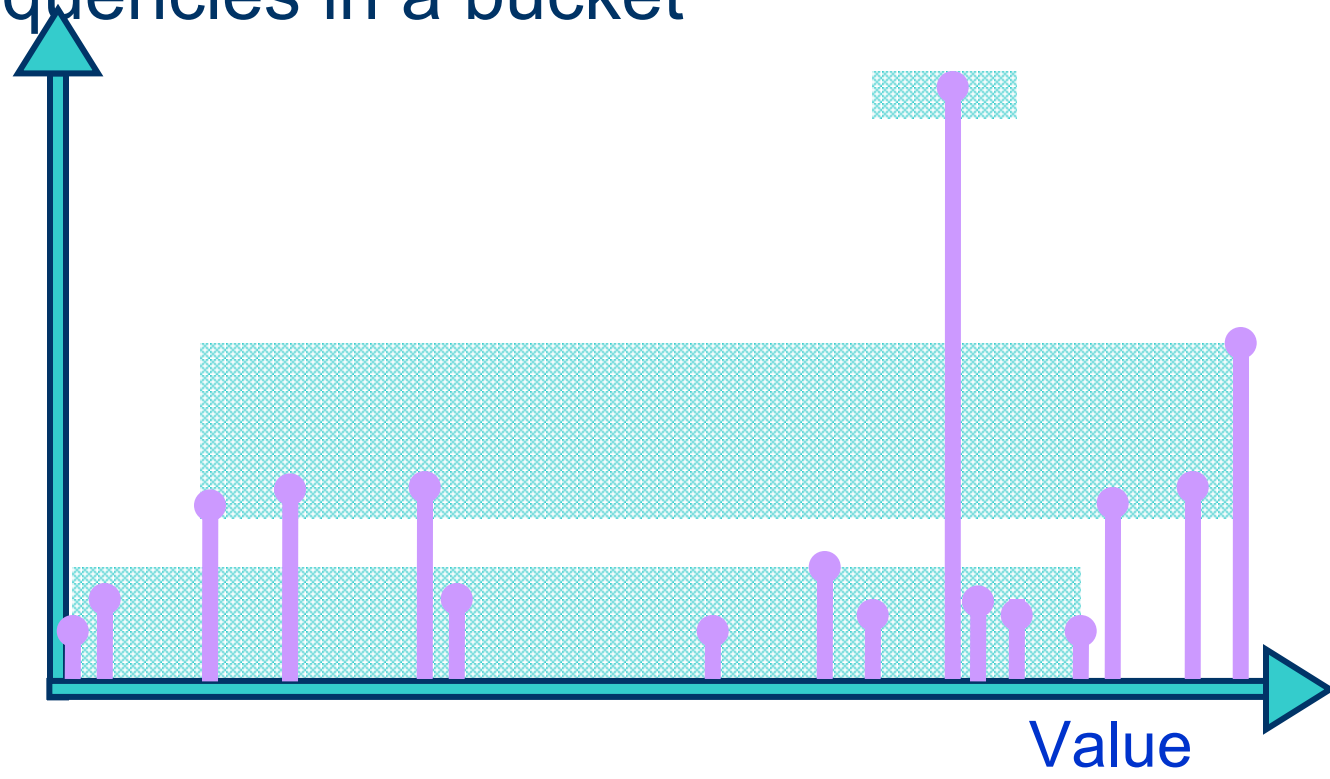
- Another example: What if the query range boundary does not match the bucket boundary?
- Scale the last bucket!
- **Assumption**: **uniform distribution** within a bucket!
- **Disadvantages**: High variance!
  - Difficult to estimate errors
  - example: Our previous graph was ‘continuous’. What if we get crazy graphs? Big guys besides short guys!
  - Consider self-join with this histogram

Alternative histogram

- Don’t equalize ranges of values but number of tuples in bucket
- equi-depth histograms
- Disadvantages:
  - Variance within a bucket may be still very high
  - Storage requirement same as equi-width, but more complex to maintain
- Work well for range queries only when the data distribution has low skew

Frequency based histograms

- Minimize variance – group the items having similar frequencies in a bucket



Serial Histogram

Serial Histogram H2

Department	Frequency
General Management	2
Defense	2
Education	1
Domestic Affairs	2
Agriculture	1
Commerce	1
Justice	1
Energy	3

Department	Histogram H1		Histogram H2	
	Frequency in Bucket	Approximate Frequency	Frequency in Bucket	Approximate Frequency
Agriculture	1	1.5	1	1.33
Commerce	1	1.5	1	1.33
Defense	2	1.5	2	1.33
Domestic Affairs	2	1.5	(2)	2.5
Education	(1)	1.75	1	1.33
Energy	(3)	1.75	(3)	2.5
General Management	(2)	1.75	2	1.33
Justice	(1)	1.75	1	1.33

Serial Histogram

- The frequencies of the attribute values associated with each bucket are either all greater than or all less than the frequencies of the attribute values associated with other bucket
- Advantage: Optimal for reducing errors in estimation
- Disadvantage: Storage requirement high
  - No order-correlation between values and frequencies → index required leading to approximate frequency of every individual attribute value
  - Optimal for equality joins and selection only when attribute value list is maintained for each bucket
- Can we reduce the storage requirement?

End-biased histograms

- Some of the highest frequencies and some lowest frequencies are explicitly and accurately maintained in separate individual buckets
- Remaining (middle) frequencies are all approximated together in a single bucket
- Storage requirement: very little and no index required
  - Only store attribute values in individual buckets and their corresponding frequencies

Histograms… formally

- Definition
  - A histogram on attribute is constructed by partitioning the data distribution D into mutually disjoint β subsets called buckets and approximating the frequencies f and values V in each bucket in some common fashion.

Key properties

- **that characterize histograms and determine their effectiveness in query result size estimation.**
  - Partition class
  - Sort parameter
  - Partition constraint
  - Source parameter
- **These properties are mutually orthogonal and form the basis for a general taxonomy of histograms.**

Orthogonal Properties of Histogram

- Partition Class
  - Indicates restrictions on partitioning
    - Serial: non-overlapping ranges of sort parameter values
    - End-biased: at most one non-singleton bucket
- Partition Constraint
  - The mathematical constraint that uniquely identifies the histogram within its partition class.
    - Equi-depth: sum of number of tuples in a bucket should be equal

Orthogonal Properties of Histogram…

- Sort parameter and Source parameter
  - Derivative of data distribution element (its value and/or frequency)
    - Attribute values (V)
    - Frequencies (F)
    - Areas (A) = spread x frequency
  - Serial: buckets must contain contiguous sort parameter values

Orthogonal Properties of Histogram…

- Approximation of values within a bucket
  - The assumptions that determine the approximate values within a bucket
    - Uniform distribution of values within a bucket
- Approximation of frequency of a value within a bucket
  - assumptions that determine the approximate frequency of each value within a bucket
    - Frequency of each value assumed to be arithmetic mean

Histogram Taxonomy

SORT PARAMETER	SOURCE PARAMETER		
	SPREAD (S)	FREQUENCY (F)	CUM. FREQ(C)
VALUE(V)	EQUI-SUM	EQUI-SUM	SPLINE-BASED
FREQUENCY(F)		V-OPTIMAL	

- Each histogram is primarily identified by its partition constraint and its sort and source parameters.
- If the choice in the above three properties is p, s, and u, respectively, then the histogram is named **p(s,u).**

Histogram Taxonomy

- Trivial Histogram
- Equi-sum(V,S) alias Equi-width
- Equi-sum(V,F) alias Equi-depth
- V-optimal (F,F)
- V-optimal-end-biased (F,F)

- Spline-based(V,C)

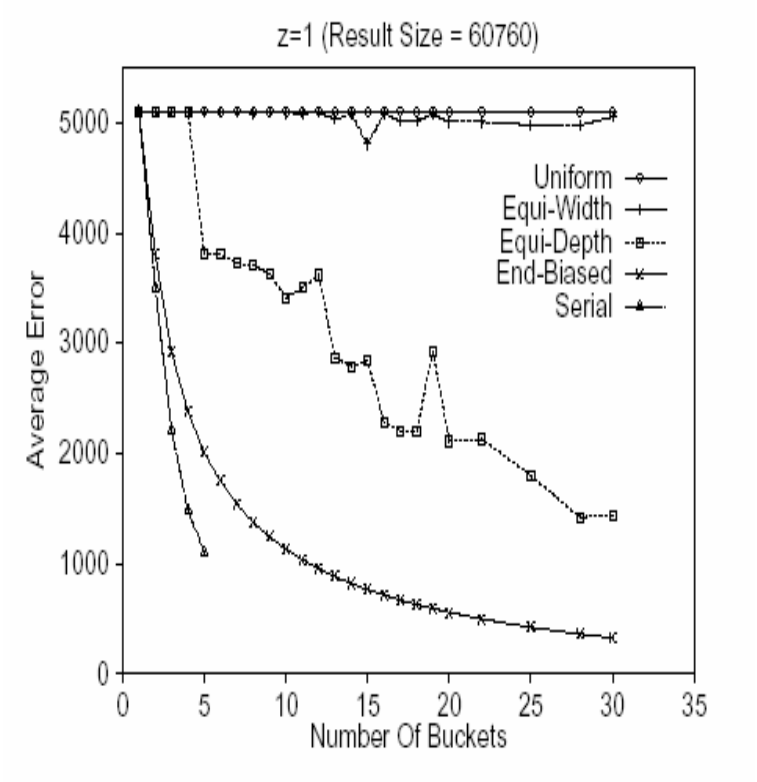
Histogram Taxonomy…

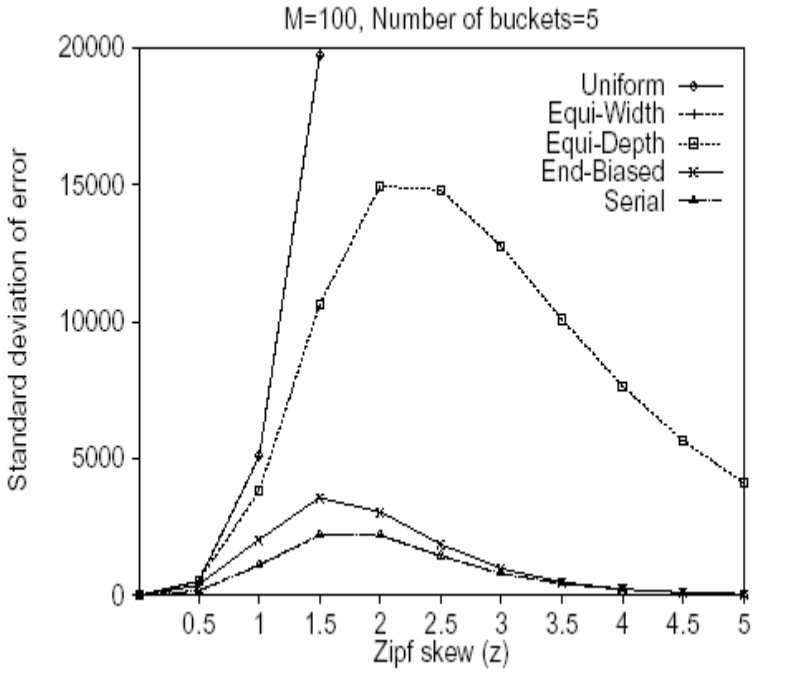
- Trivial Histogram
  - All values in a single bucket
- Equi-sum(V,S) alias Equi-width
  - group contiguous ranges of attribute values into buckets, and the sum of the spreads in each bucket (i.e., the maximum minus the minimum value in the bucket) is approximately equal to times 1/ β times the domain range

Histogram Taxonomy…

- Equi-sum(V,F) alias Equi-depth
  - are like equi-width histograms but have the sum of the frequencies in each bucket be equal rather than the sum of the spreads
- V-optimal(F,F) histograms
  - group contiguous sets of frequencies into buckets so as to minimize the variance of the overall frequency approximation.
  - Were simply called serial histograms before

Performance Comparison





Observations

- Performance
  - Frequency-based much better than traditional
  - End-based almost as good as serial
- Construction cost
  - Serial histograms are exponential in the number of buckets whereas end-based are almost linear
- Complexity in storage and usage
  - Serial has much higher complexity than end-based

More Histograms

- Max-diff and compressed as partition constraint

SORT PARAMETER	*SOURCE PARAMETER*			
	SPREAD (S)	FREQUENCY (F)	AREA (A)	CUM. FREQ (C)
VALUE (V)	**EQUI–SUM**	**EQUI–SUM** **V–OPTIMAL** **MAX–DIFF** **COMPRESSED**	**V–OPTIMAL** **MAXDIFF** **COMPRESSED**	**SPLINE–BASED** **V–OPTIMAL**
FREQUENCY (F)		**V–OPTIMAL** **MAXDIFF**		
AREA (A)			**V–OPTIMAL** **MAXDIFF**	

Figure 3: Augmented Histogram Taxonomy.

More partition constraints

- Max-diff
  - a bucket boundary between two source parameter values that are adjacent (in sort parameter order) if the difference between these values is one of the β -1 largest such differences.
  - Goal is to avoid grouping attribute values with vastly different source parameter values into a bucket
  - Efficiently constructed by first computing differences between adjacent* source parameters
- *Compressed*
  - Highest source values are stored separately in singleton buckets; the rest are partitioned as in an equi-sum histogram.
  - Achieve great accuracy in approximating skewed frequency distributions and/or non-uniform spreads

Any (more) Questions?

Thank You!