# REAL-TIME EXPLORATION OF LARGE SPATIOTEMPORAL DATASETS BASED ON ORDER STATISTICS

Original Work By: Cicero Pahins, Nivan Ferreira, and Joao Comba

Presented By: Vaastav Anand

# OUTLINE

**Background**

Motivation

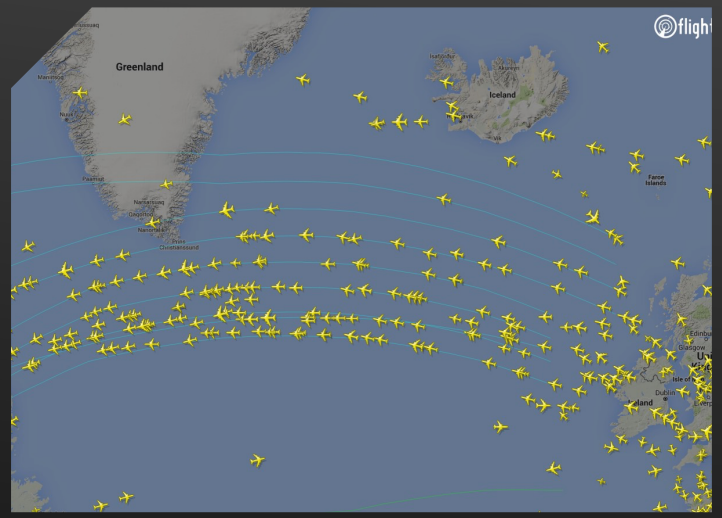
Quantile Datacube Structure

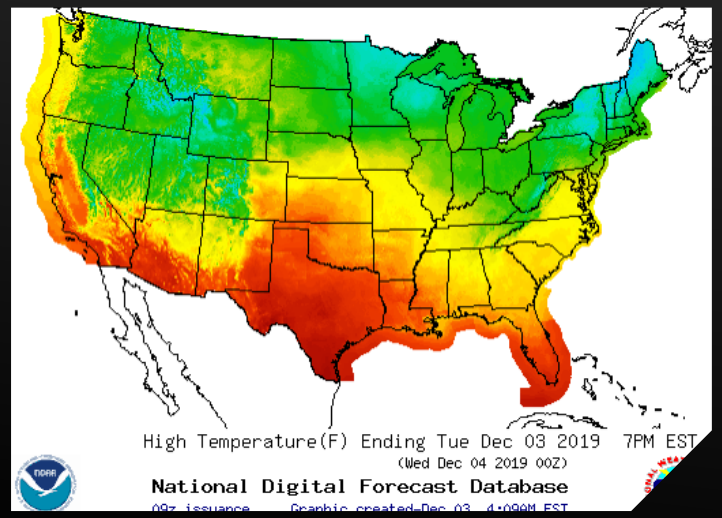
Example Visualizations

Evaluation

Critique

# SPATIOTEMPORAL DATASETS




*" Datasets generated from measuring a set of values across set of locations (spatial dimension) across a time range (temporal dimension)."*

PROBLEM: Dataset is too large to perform interactive analysis.

# EXISTING SOLUTIONS

- Precomputed Indices that store aggregations of a given dataset as solutions to this problem.
  - Doesn't take into account data distribution
- Gaussian Cubes that support interactive data modelling by describing data distribution using parametric Gaussian distributions.
  - Based on non-robust statistics (mean + covariance)
  - Can't assume real-world data has a normal distribution

# OUTLINE


Background


**Motivation**


Quantile Datacube Structure


Example Visualizations


Evaluation


Critique

# PROBLEMS WITH CURRENT SOLUTIONS

- Memory footprint is too high
- Distributions are approximated using non-robust statistics
- Queries are slow enough to disallow interactive experience
- Queries are limited to count queries

# OUTLINE


Background


Motivation


**Quantile Datacube Structure**

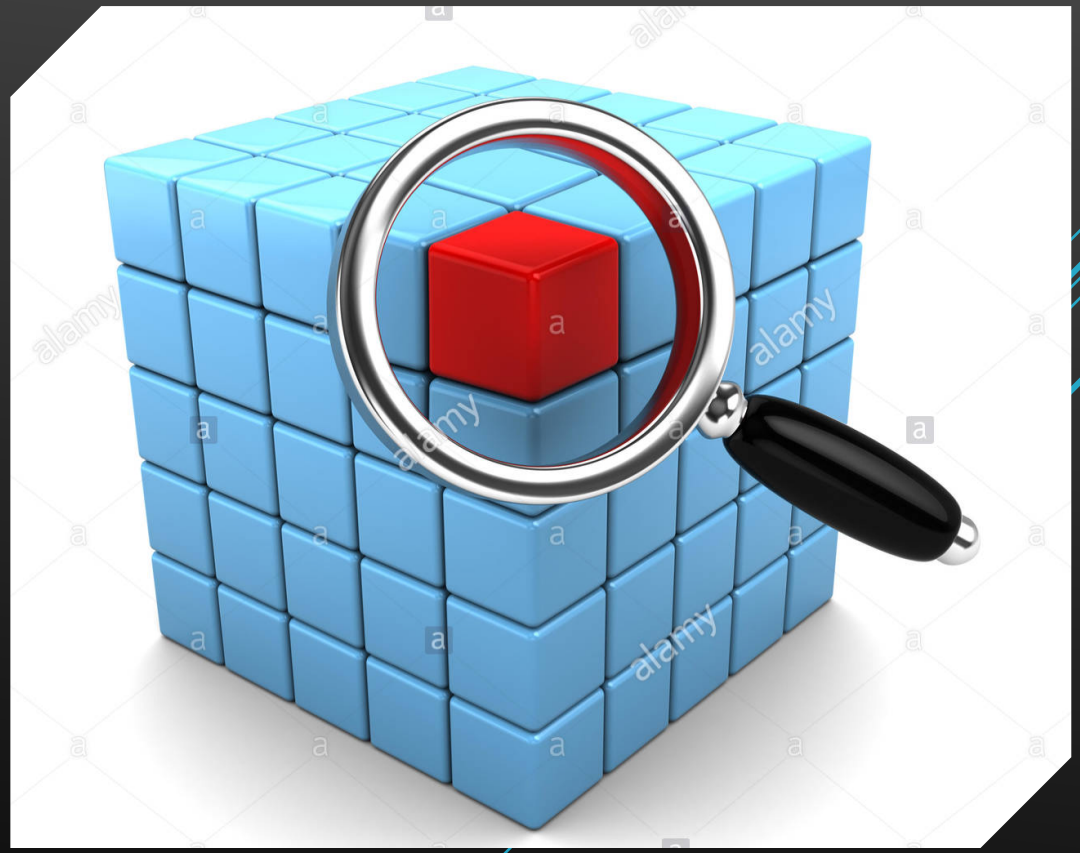

Example Visualizations


Evaluation


Critique

# QUANTILE DATACUBE STRUCTURE

- A novel datastructure
- Encode data distributions based on robust statistics
- Uses a non-parametric modelling technique called p-digest
- A novel indexing structure that reduces the large memory footprint



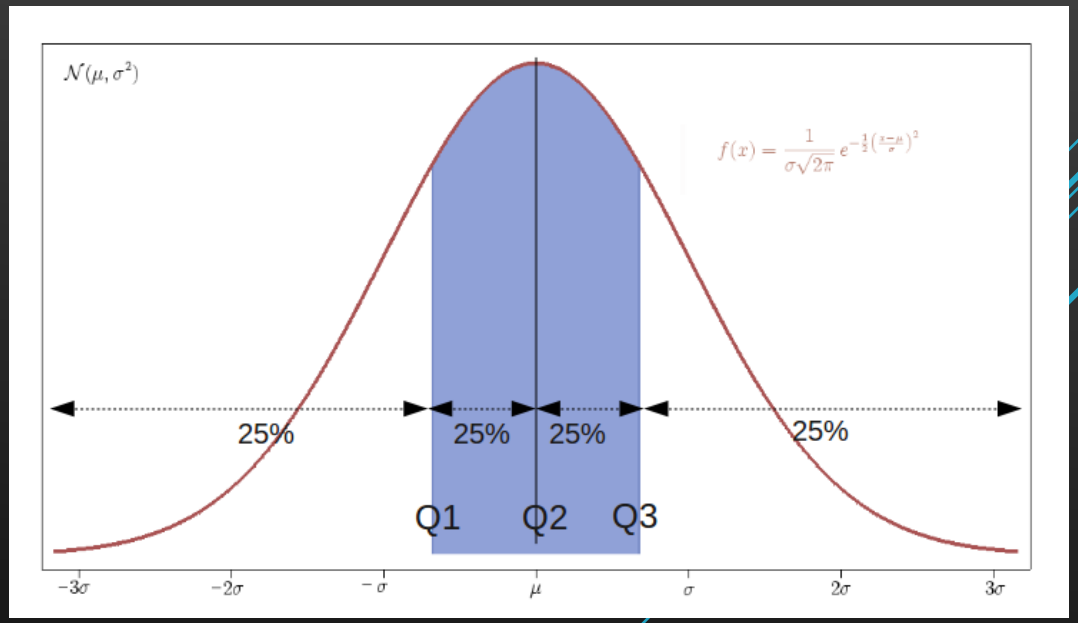
# EXAMPLE QUESTIONS

- *“How likely is a flight operated by Delta Airlines to be delayed more than 10 minutes at JFK airport?”*
- *“How does the distribution of flight delays for two airports compare to each other in the past month?”*
- *“How unusual are the delays experienced by Delta flights on January 29th, 2017?”*

# INDEXING SCHEME

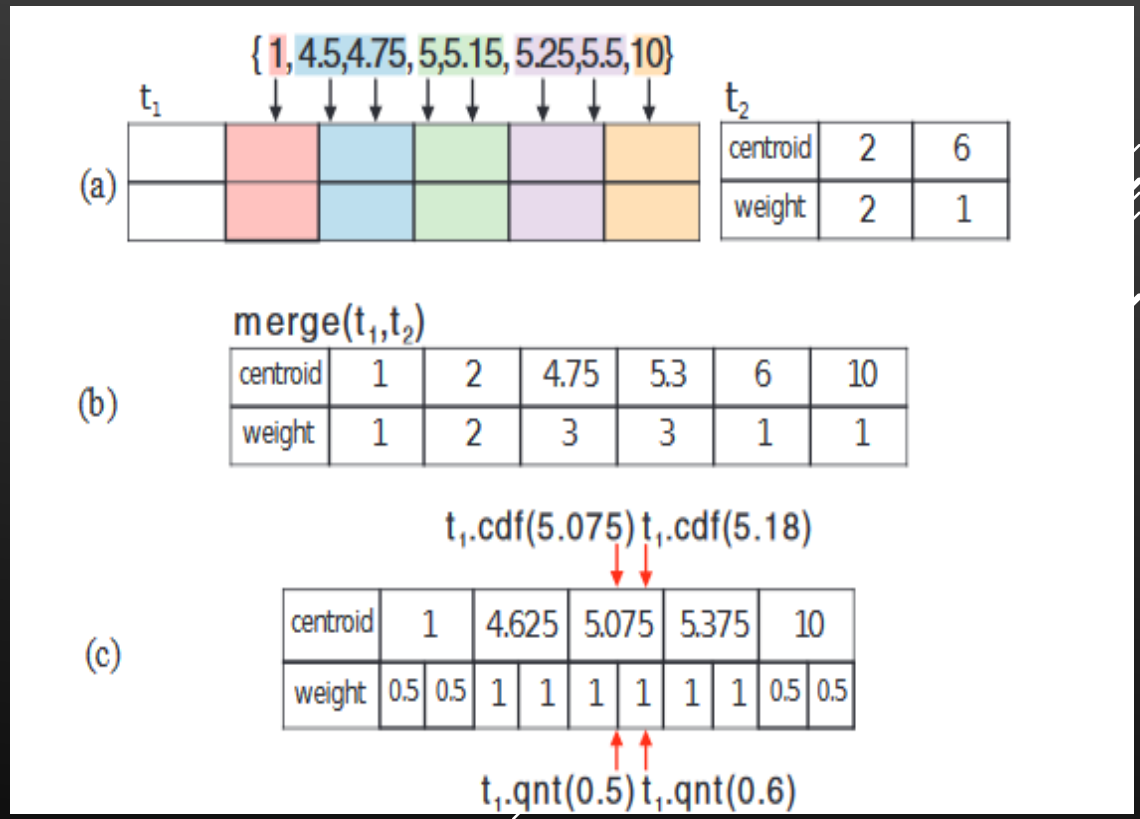
# P-DIGEST DATA SKETCH : T-DIGEST DATA SKETCH

- “Data Sketch”: Data Structure that can be easily updated with new or modified data and supports a set of queries whose results approximate queries on the full dataset.
- An optimized version of t-digest data sketch
  - Quantile sketch that supports queries of quantile and cdf estimation.
  - Summarizes the empirical cdf of an input dataset by a set of weighted values called centroids.



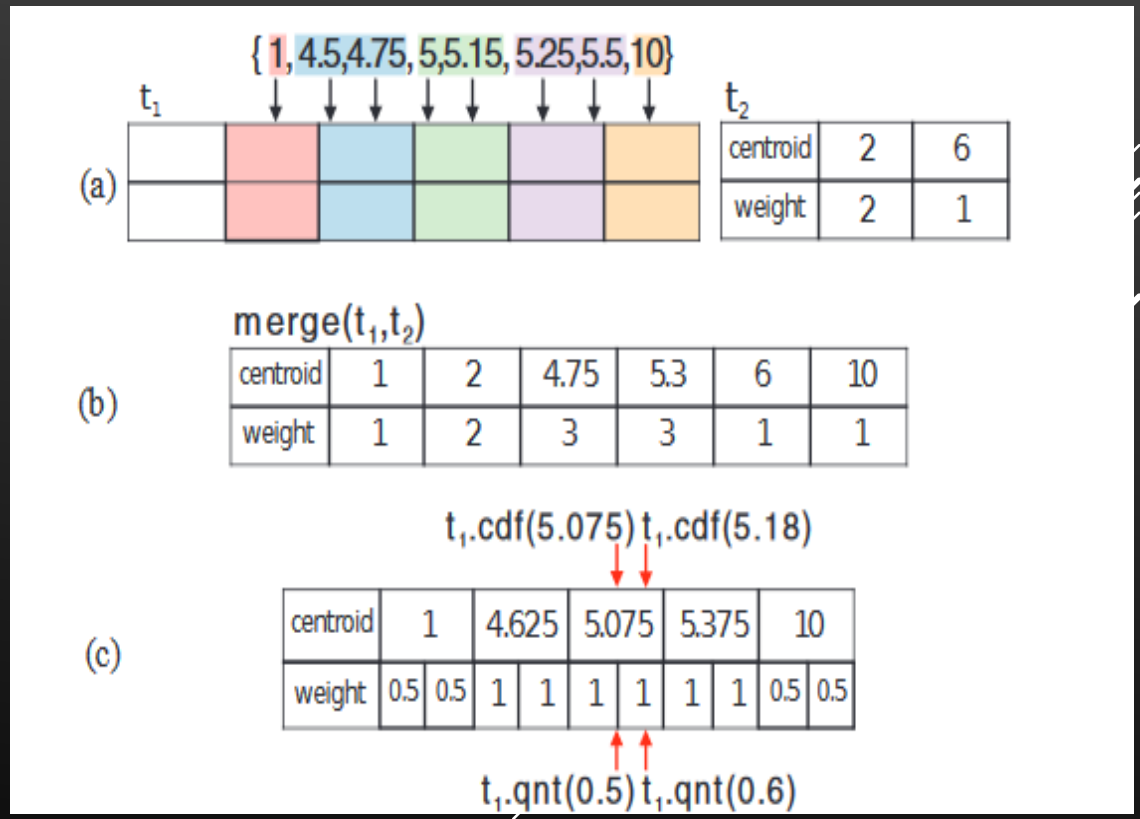
# P-DIGEST DATA SKETCH : T-DIGEST DATA SKETCH

- “Data Sketch”: Data Structure that can be easily updated with new or modified data and supports a set of queries whose results approximate queries on the full dataset.
- An optimized version of t-digest data sketch
  - Quantile sketch that supports queries of quantile and cdf estimation.
  - Summarizes the empirical cdf of an input dataset by a set of weighted values called centroids.
  - Compression Parameter defines the number of centroids
  - Queries for extreme quantiles are made more accurate



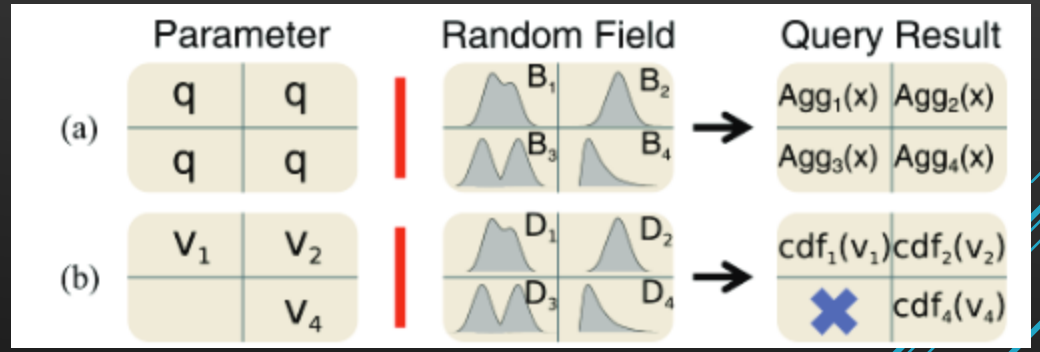
# P-DIGEST DATA SKETCH

- “Data Sketch”: Data Structure that can be easily updated with new or modified data and supports a set of queries whose results approximate queries on the full dataset.
- An optimized version of t-digest data sketch
  - Reduce centroid storage from 40-80 bytes per centroid array to at most 8 bytes for each of the centroid and weighted arrays. Stored as single chunk of floats
  - If all weight values are 1 then weights are not stored.



# QUERY TYPES

- 3 types
  - Quantile queries
  - CDF queries: Modeled as inverse of quantile queries
  - Pipeline queries: Result of a given query is used as a parameter to another query




Primary Query: **select AGGR from QDR where CONSTRAINTS [group by G]**

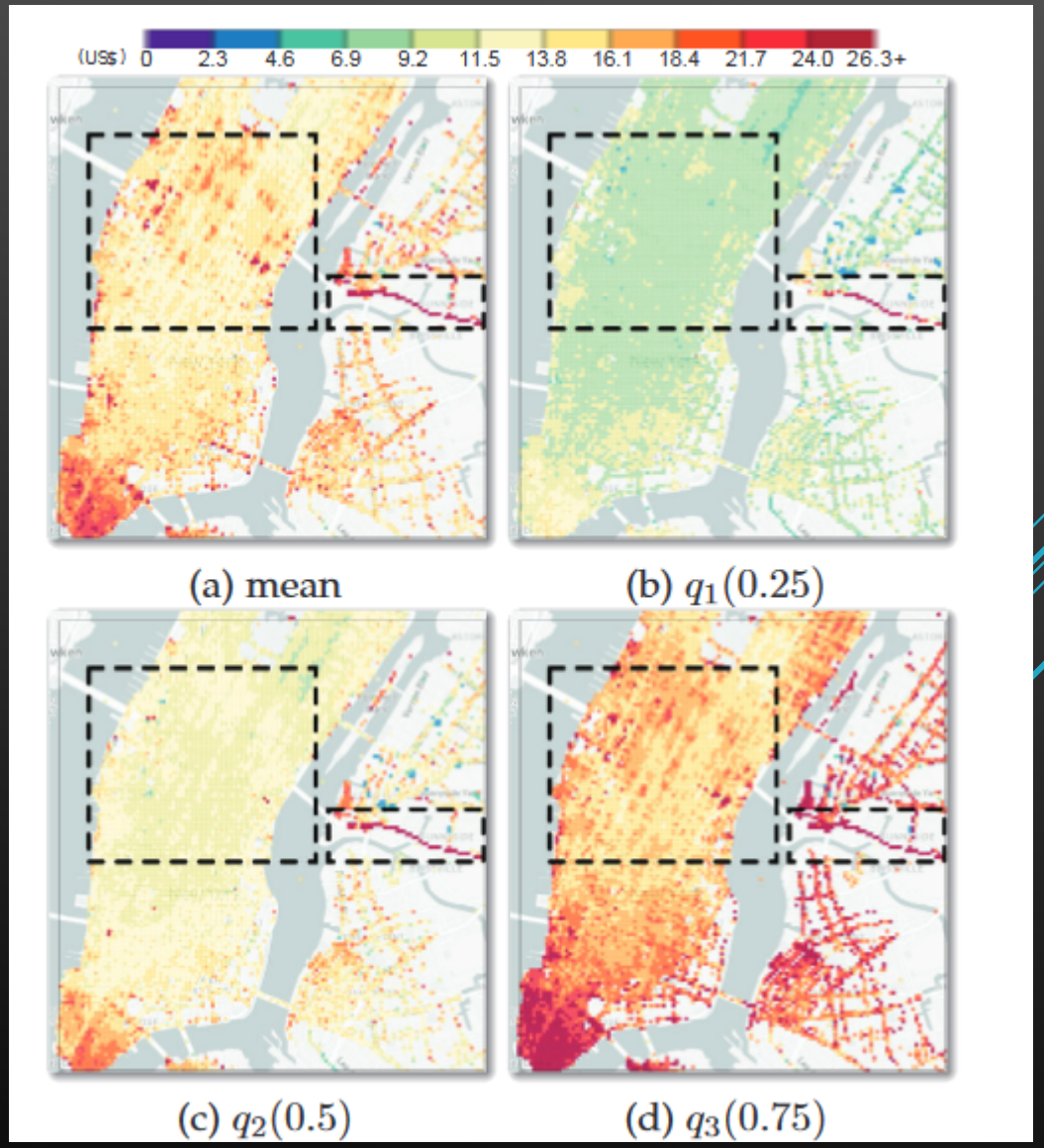
# QUERY ALGORITHM

# OUTLINE

- Background
- Motivation
- Quantile Datacube Structure
- **Example Visualizations**
- Evaluation
- Critique

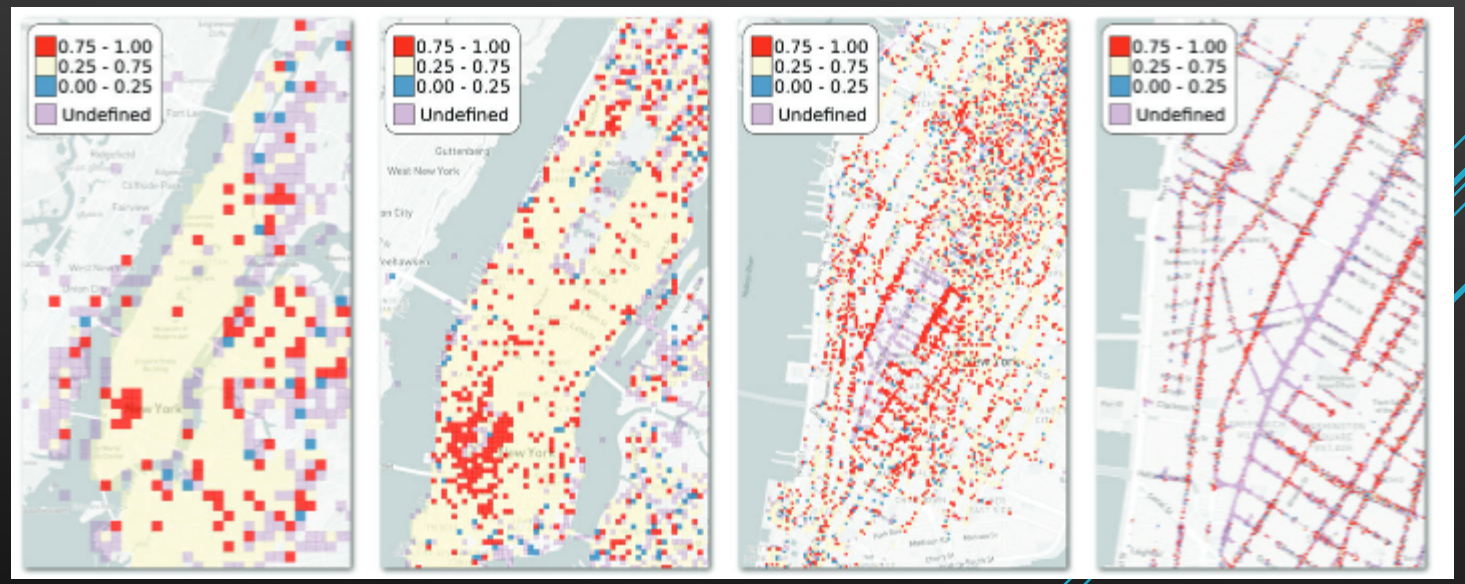
# QUANTILE HEATMAPS

- Instead of using the mean for a given location, use the specified quantile at the location as the aggregate measure
- Quantiles are not sensitive to outliers whereas mean is.
- Powered by QDS's quantile queries



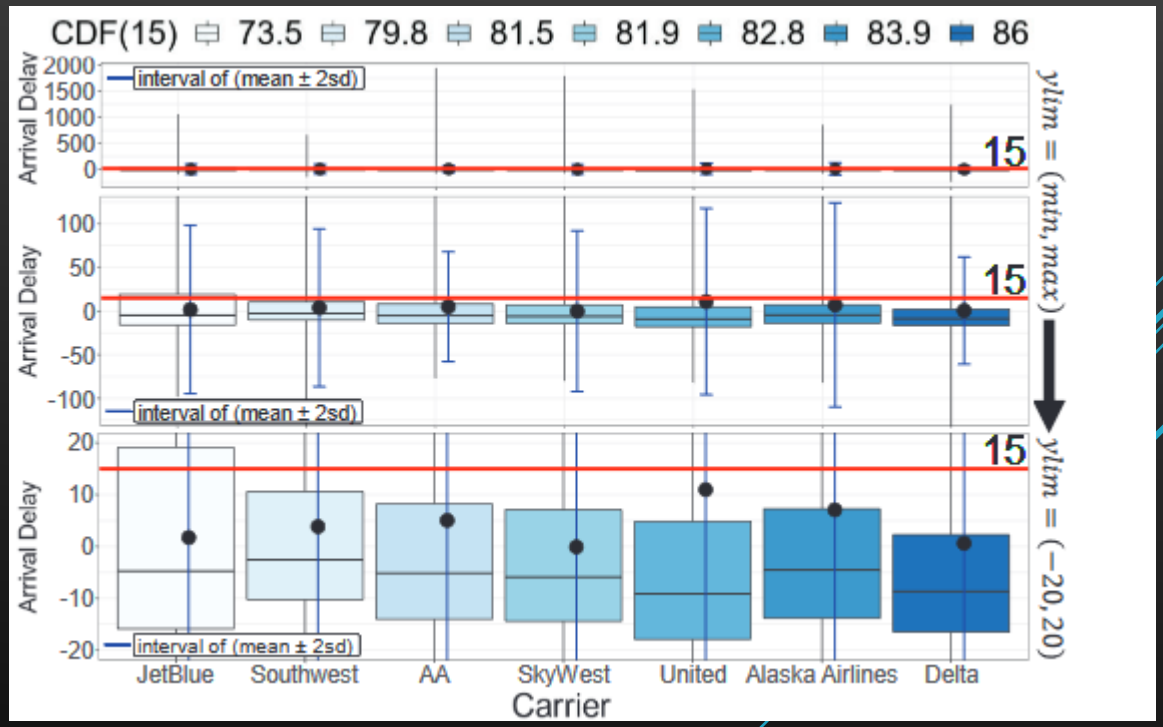
# CDF HEATMAPS

- Instead of showing mean, they show high likely a distribution in a given location is to be smaller than a certain value
- Powered by QDS's cdf query



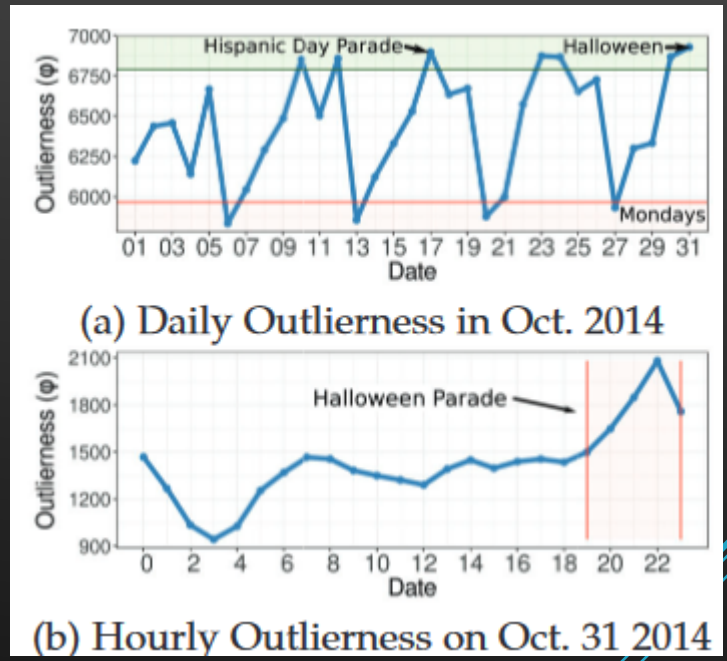
# READABLE UNCERTAINTY VISUALIZATIONS

- Interpreting uncertainty visualizations is not easy.
  - Can have high cognitive load
  - Viz researchers have idioms to shift the cognitive load
  - But existing interactive techniques are not efficient
- QDS's cdf query can allow for this interactivity by computing the result for each box plot.



# OUTLIER EXPLORATION

- Finding outliers requires users to inspect a large number of data slices over time and space.
- QDS can retrieve approx. distributions over an arbitrary portion of the data very quickly.
- Authors define an outlierness measure supported by QDS's pipeline query.



# OUTLINE

- Background
- Motivation
- Quantile Datacube Structure
- Example Visualizations
- **Evaluation**
- Critique

# MEMORY FOOTPRINT

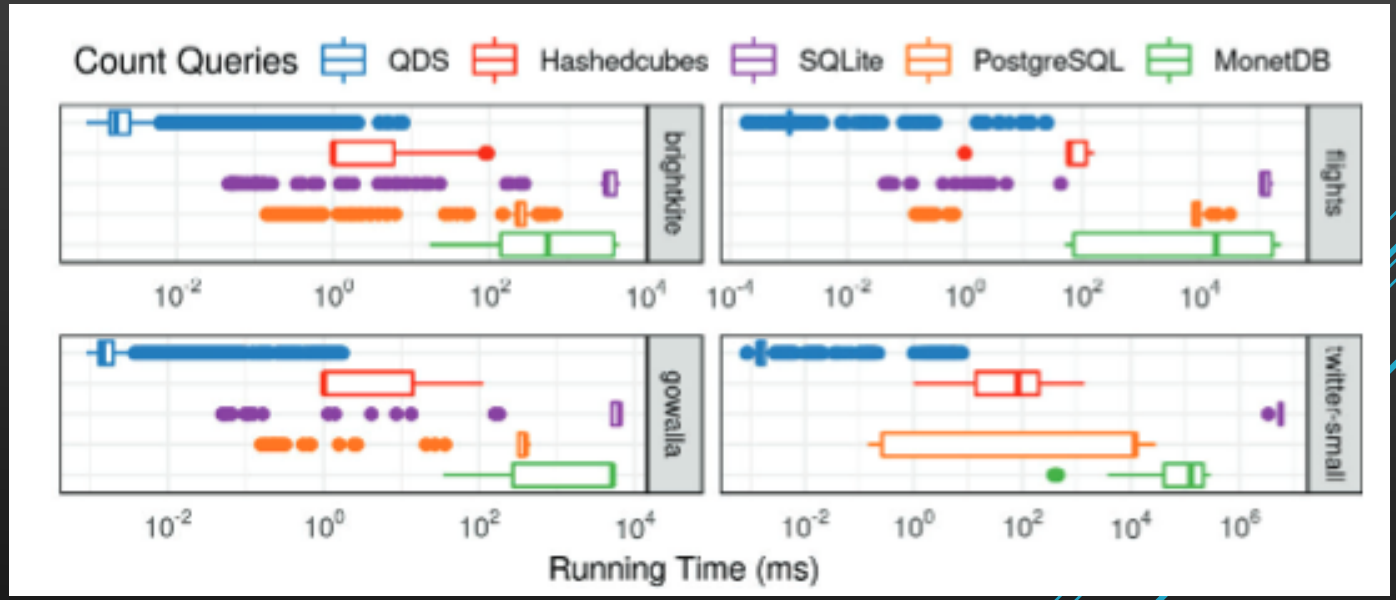
- Comparable memory footprint with HashedCubes.
- Better for some datasets, Worse for other datasets

Overall summary of the relevant information for building QDS.

| dataset | size | index schema(bits) | payload schema | QDS Memory/Time | | HC Memory/Time |
|---|---|---|---|---|---|---|
| | | | | leaf-size = 1 | leaf-size = 32 or 64 | leaf-size = 32 or 64 |
| brightkite | 4.5 M | dayOfWeek (3), hourOfDay (5), time (16), lat (25), lon (25) | NA | 455 MB/9s | 276 MB/7s | 366 MB/7s |
| gowalla | 6.4 M | dayOfWeek (3), hourOfDay (5), time (16), lat (25), lon (25) | NA | 711 MB/13s | 367 MB/11s | 743 MB/13s |
| twitter-small | 210.6 M | device (3), time (16), lat (17), lon (17) | NA | 3.1 GB/05:55m | 2.7 GB/05:54m | 4.9 GB/10:53m |
| twitter | 210.6 M | app (2), device (3), language (5), time (16), lat (17), lon (17) | NA | 4.6 GB/06:39m | 4.2 GB/06:37m | 9.4 GB/12:04m |
| flights | 121.2 M | dep. delay (4), carrier (11), dep. time (16), lat (25), lon (25) | arrDelay, depDelay | 1.4 GB/02:50m | 1.4GB/02:50m | 457 MB/03:56m |
| green-taxis-small | 42 M | pickupDateTime (16), lat (22), lon (22) | ttlAmount, distance | 1.3 GB/01:24m | 1.2 GB/01:16m | 788 MB/03:56m |
| green-taxis | 42 M | dayOfWeek (3), hourOfDay (5), pickupTime (16), lat (22), lon (22) | ttlAmount, distance | 1.3 GB/01:16m | 1.2 GB/01:15m | 3.0 GB/01:49m |
| yellow-taxis-small | 706 M | pickupDateTime (16), lat (22), lon (22) | ttlAmount, distance | 9.7 GB/27:53m | 9.3 GB/28:04m | 7.0 GB/18:14m |
| yellow-taxis | 706 M | dayOfWeek (3), hourOfDay (5), pickupTime (day), lat (22), lon (22) | ttlAmount, distance | 9.7 GB/31:37m | 9.3 GB/31:33m | 12.6 GB/20:38m |

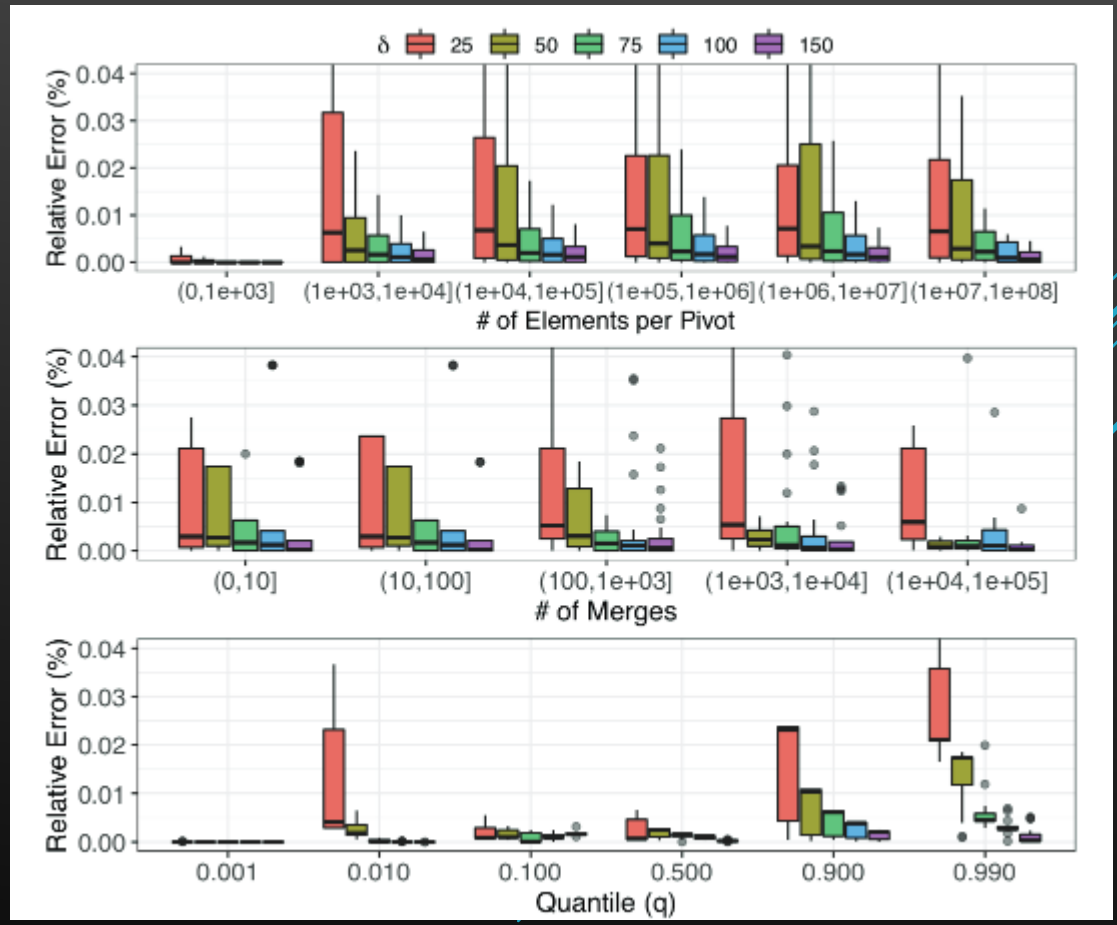
# PERFORMANCE

- Measures the performance of count queries.
- QDS is faster than HashedCubes, MonetDB, SQLite, PostgreSQL
- MonetDB, SQLite, PostgreSQL don't provide effective mechanisms for spatial filtering with temporal and categorical constraints.



# P-DIGEST APPROXIMATION ERROR

- The error rate is fairly low for low quantiles, high merges, and low # of elements per pivot
  - More accurate when data broken down into more parts
  - Accurate for small input data
  - Large values of compression parameter better for higher quantile estimation.
- The error measured is the relative error of the estimated quantile to the actual empirical quantile



# OUTLINE


Background


Motivation


Quantile Datacube Structure

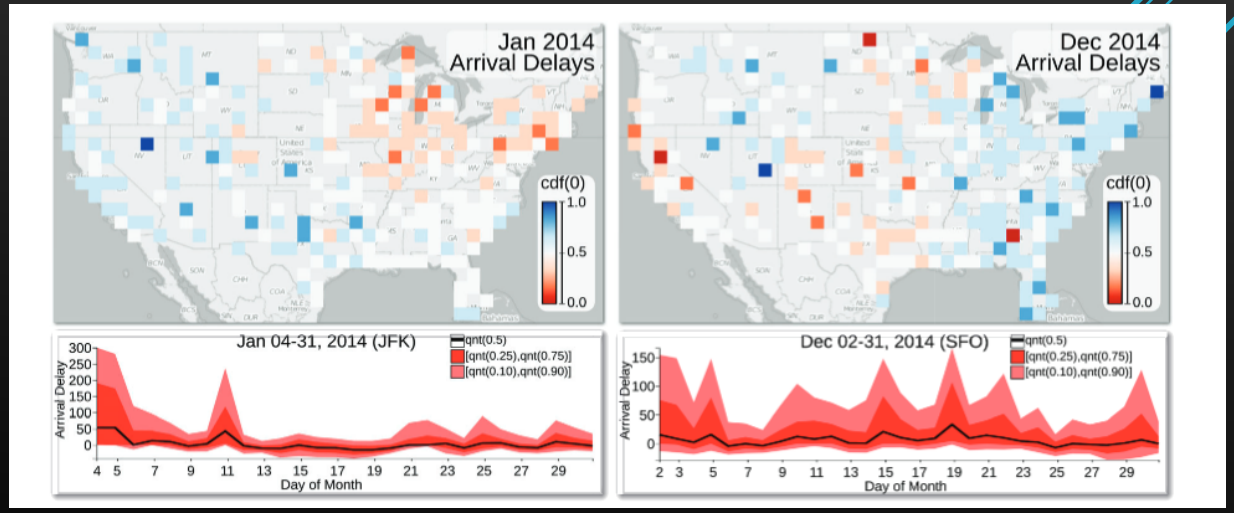

Example Visualizations


Evaluation


**Critique**

# WHAT-WHY-HOW FRAMEWORK

- What?
  - Large Static Spatiotemporal datasets
- Why?
  - Better Memory Footprint
  - Quicker Results -> More Interactive exploration
  - Robust Metrics + Visualizing Uncertainty
- How?
  - QDS – New Datastructure for storing indices of spatiotemporal datasets
  - New queries that can approximate quantiles + cdf.



# STRENGTHS

- Thorough performance eval
  - Good comparisons vs believable baselines
- Built on/Promotes usage of Robust Statistics
- Allows for exploration of uncertainty in datasets.

# WEAKNESSES

- Lack of Validation: No User Study
  - One of the goals was more interactivity. Can’t validate w/o user study.
- Lack of Error Estimation
  - The distributions are approximations but currently the user has no way of knowing how good the approximation is
- Lack of Detail: Building QDS with a dataset. (minor)
  - Needs more detail about how to specify the schema
- Meta Comment: Should use a spellchecker. (minor)

# CONCLUSION

- Presents QDS: a fast in-memory data structure
  - Supports uncertainty exploration + data distribution estimation
  - Designed for large static spatiotemporal datasets
- Source Code: https://github.com/cicerolp/qds
- Video: https://vimeo.com/262669555