VizCertify

A framework for secure visual data exploration

Lorenzo De Stefani (Brown U.), **Leonhard F. Spiegelberg** (Brown U.),
Eli Upfal (Brown U.) and Tim Kraska (MIT CSAIL)

IEEE DSAA - October 8th, 2019

Introduction

New data arrives → What now?

Introduction

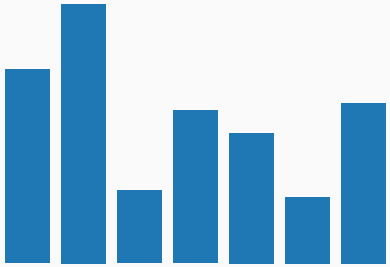
New data arrives → What now?

⇒ Natural approach is to explore data via *visualizations* to understand its statistical behavior.

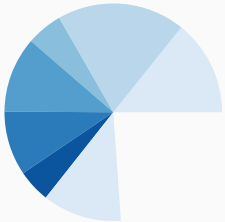
⇒ Particularly useful are *histograms*.

Visualizations

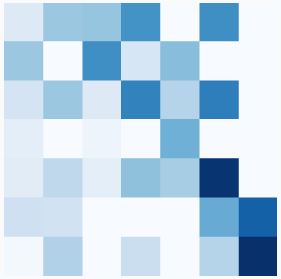
We look at histograms in the broader sense, i.e. any visualization that can be derived from a histogram over discrete data.



bar charts



pie charts



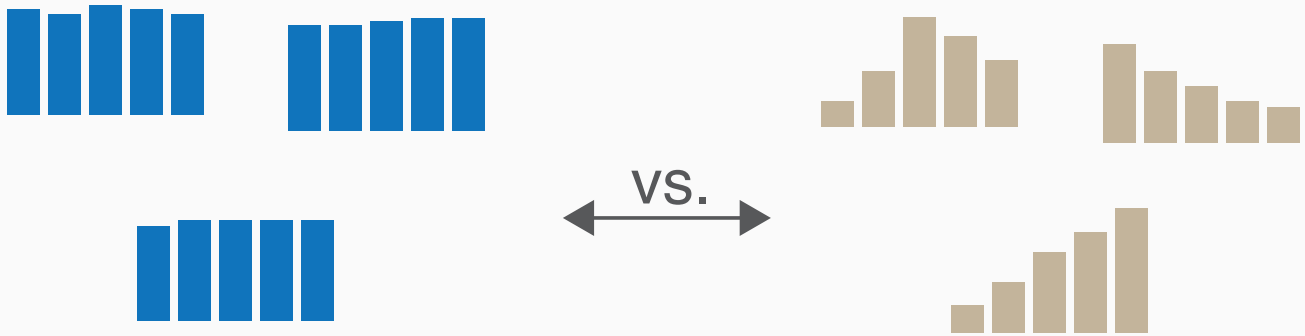
heatmaps

Why care about visualizations?

$\Longrightarrow$ visualizations allow to gain fast insight into data. Many existing systems like **SeeDB** (2014)[5], **Data Polygamy**(2016)[1], ..., **VizML**(2019)[3] use visualizations as primary tool for exploration.

“Can exploration be automated?”

⇒ Which visualizations are interesting?
Visualizations whose data distribution *differs* are interesting.



⇒ automate filtering to retrieve *interesting* subpopulations of the data whose data distribution *differ*.

⇒ base search on a *reference visualization* for one attribute, e.g. a flat prior or a subpopulation selected via a *reference filter*.

Why do I need recommendation?

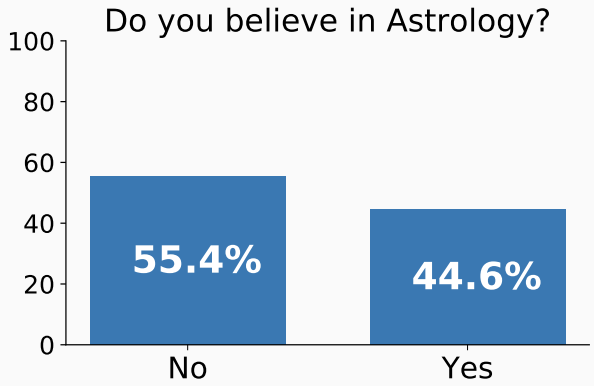
Automatic exploration leads to a very high number of visualizations in the worst case.
$\Longrightarrow$ Recommend a curated list with only the most *interesting* ones.

General approach to explore data:

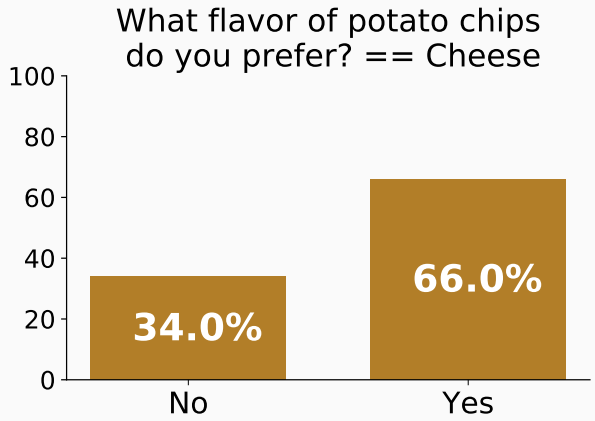
1. Generate all visualizations upfront and step through them to find insight.
2. **Interactive exploration:** Iteratively step through and generate visualizations.

Example

Survey of 2,644 participants from the U.S., 32 questions.



Reference visualization over full data sample



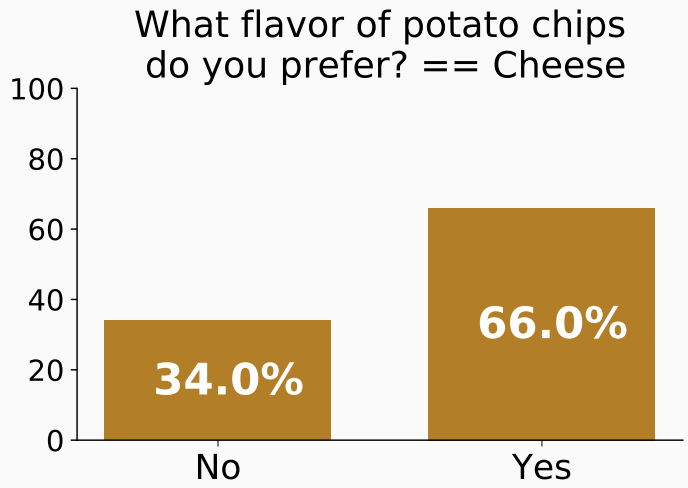
Top **recommended visualization** after filtering down on Cheese flavour

filter predicate: boolean expression rows have to satisfy to be selected for visualization.

⇒ recommend *interesting* visualizations by adding **filter predicates** to the *reference visualization*. E.g., here $\text{chip\_flavor} = \text{Cheese}$

Histogram queries

In this work: Visualizations are based on discrete random variables, or those which can be equipped with a natural metric. Visualizations can be obtained through a SQL query with F being a filter predicate, Y the random variable to study and X the attribute under which Y is partitioned via



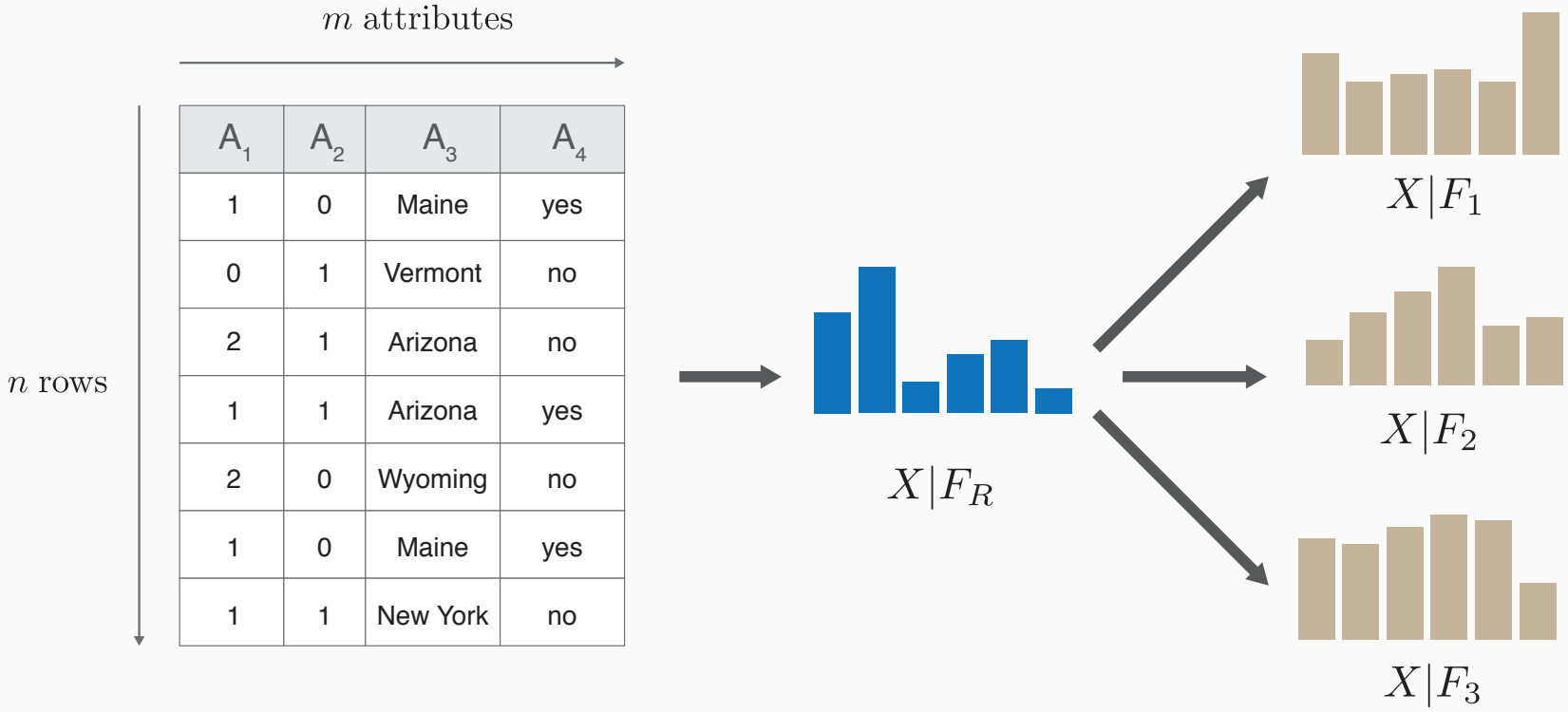

```
SELECT X, COUNT(Y) FROM D WHERE F GROUP BY X.
```

⇒ **Example:**

```
SELECT astrology, COUNT(*) FROM survey WHERE
chip_flavor=cheese
```

⇒ recommend a candidate visualization when its histograms differs from the reference.

Recommending visualizations



A_1	A_2	A_3	A_4
1	0	Maine	yes
0	1	Vermont	no
2	1	Arizona	no
1	1	Arizona	yes
2	0	Wyoming	no
1	0	Maine	yes
1	1	New York	no

Starting from a reference visualization based on the distribution of an attribute X under an initial filter predicate F_R, a system recommends candidates based on $X|F_i$ ranked by a distance metric $d(X|F_R, X|F_i)$.

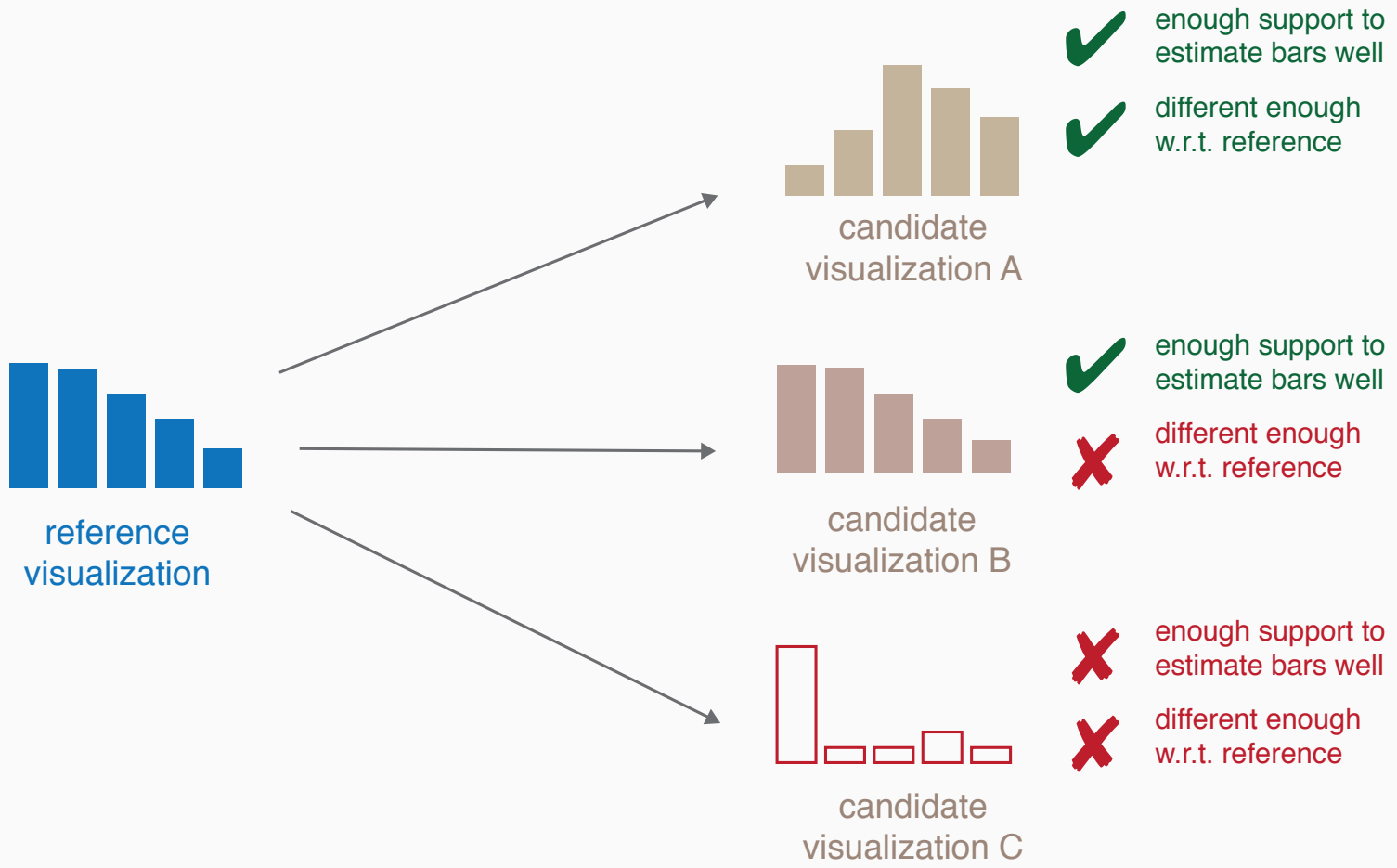
Problem

False discovery: A recommendation for something which is not actually true, i.e. a type I error.

Problems when recommending visualizations:

1. How to avoid *false discoveries*?
2. How to make sure only visualizations with *enough data support* are recommended?
3. How to allow for *interactive exploration*?

Core idea of VizCertify



Core idea behind VizCertify: Recommend a pair of visualizations if and only if it is visually distinguishable, estimates are accurate and query is performed within a safe exploration space to avoid false discoveries.

Estimating bars

p_k true, underlying probability of k-th value, i.e. $p_k = Pr[X = k]$

$\hat{p}_k$ is the estimator of the probability of the k-th value.

$\Rightarrow$ $\hat{p}_k^R$ estimates p_k^R for reference visualization R.

$\Rightarrow$ $\hat{p}_k^C$ estimates p_k^C for candidate visualization C.

Computing estimates for a visualization corresponding to a filter F:

$$\hat{p}_k^F := \frac{\sum_{i=1}^{n} \mathbb{1}_{\{X_i = k, X_i \in F\}}}{\sum_{i=1}^{n} \mathbb{1}_{\{X_i \in F\}}}$$

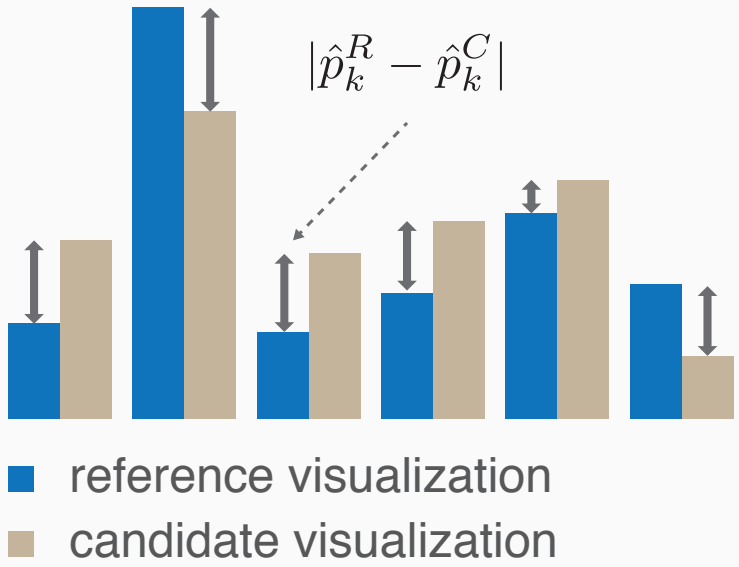
$\Longrightarrow$ i.e. relative frequency of records satisfying filter predicate F.

Chebyshev distance

VizCertify uses the Chebyshev distance (i.e. maximum difference between corresponding bars) to recommend visualizations.

$$d_k := |\hat{p}_k^R - \hat{p}_k^C|$$

$$d(X|F_R, X|F_C) := \max_{k=1,\ldots,K} d_k$$

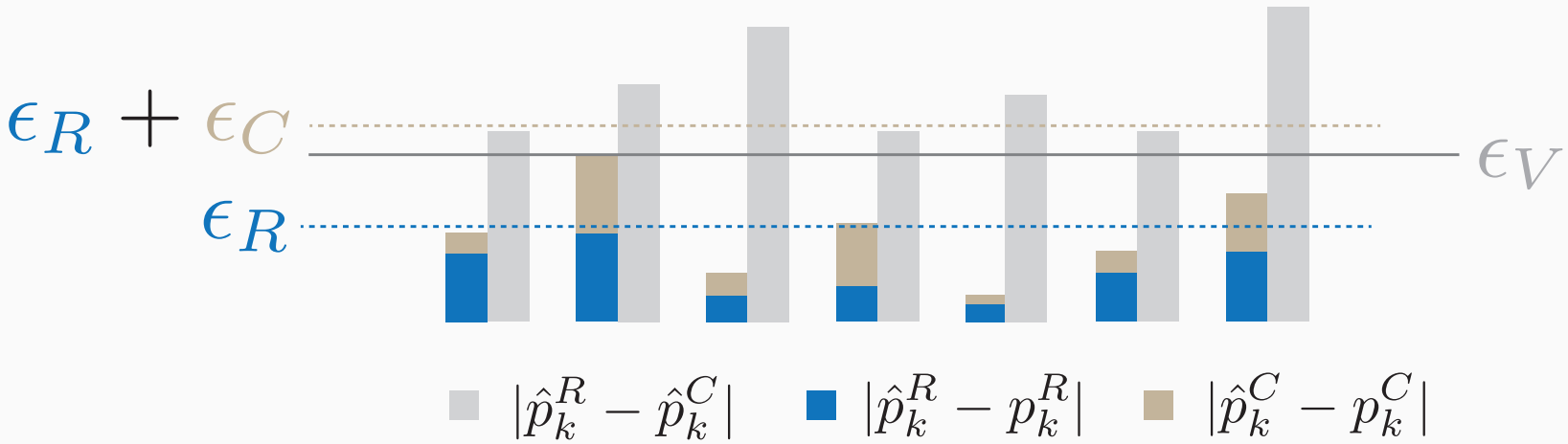


F_R, F_C is the filter predicate corresponding to the reference visualization R and candidate visualization C.

$\implies$ R, C are visually distinguishable if $d(X|F_R, X|F_C) > \epsilon_V$ for some threshold ϵ_V.

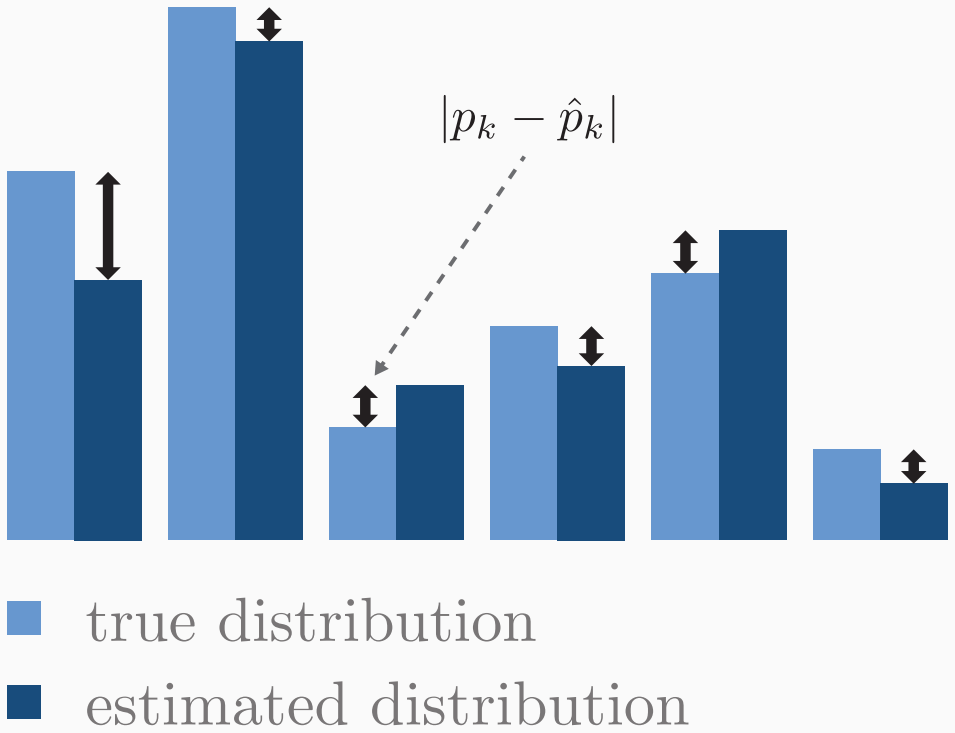
Recommending visualizations

Criterion: Recommend a candidate $X|F_C$ iff observed distance is larger than combined estimation error on $X|F_R$ and $X|F_C$ as well as a minimum observable visual distance ϵ_V.



Recommend candidate C iff for each bar $d_k > \max\{\epsilon_C + \epsilon_R, \epsilon_V\}$.

Controlling estimation error



Error between estimated $\hat{p}_k$ and true underlying probabilities p_k.

Control estimation error at $\delta \in (0, 1)$ confidence level.

$$\forall k = 1, ..., K: \quad \Pr[|p_k - \hat{p}_k| > \epsilon] < \delta$$

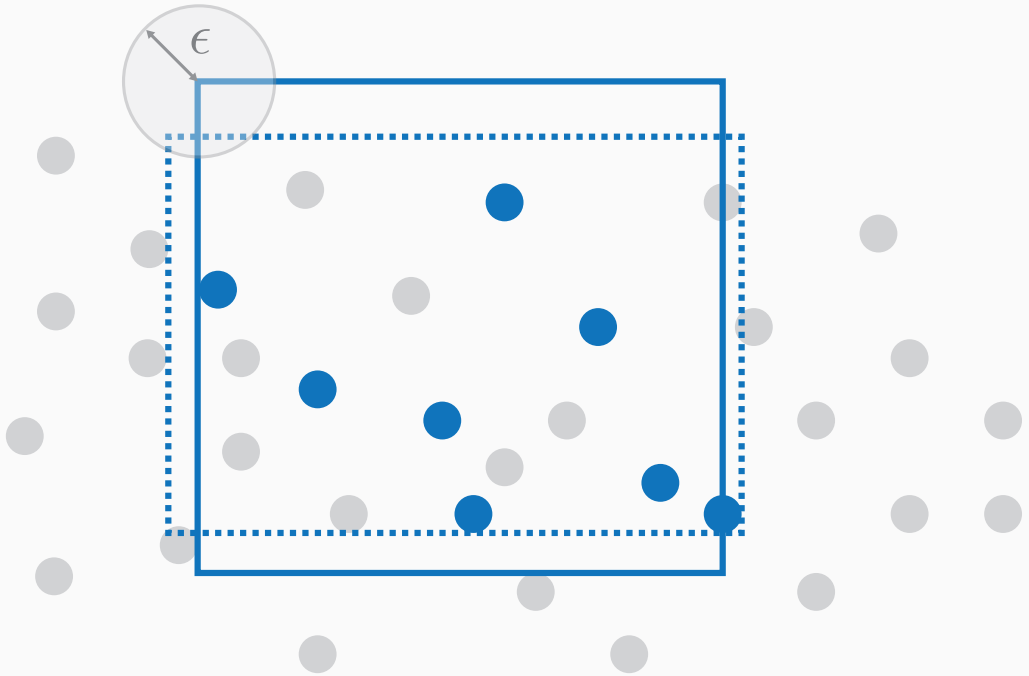
Bounding the estimation error

$\Rightarrow$ We could use e.g. Chernoff-bounds to bound the estimation error on a single bar and use the union bound to control across multiple bars (and visualizations).

$$\Pr\left[\bigcup_{k=1}^{K}\{|p_k - \hat{p}_k| > \epsilon\}\right] \leq \sum_{k=1}^{K}\Pr[|p_k - \hat{p}_k| > \epsilon] < \delta$$

$\Rightarrow$ Better bounds can be obtained using results from Vapnik and Chervonenkis' theory. Instead of bounding by the cardinality of the family of functions to estimate, bound by the VC dimension of the family of estimation functions!

Bounds via VC dimension



$\Rightarrow$ VC-theory allows us to get a bound on how well a sample approximates a function depending on the VC-dimension of the family of estimation functions.

$\Rightarrow$ **Core idea:** Learn filter queries, and use VC dimension of the query class to bound the estimation error.

VC dimension bounds on sample complexity

Estimation error for *all* $k = 1, \ldots, K$, i.e.

$$Pr\left[\left|p_k^F - \hat{p}_k^F\right| \geq \epsilon\right] < \delta$$

is not larger than ϵ at δ confidence level iff

$$\epsilon \geq \sqrt{\frac{c}{|D|F|}\left(d + \log_2 \frac{1}{\delta}\right)}.$$

with $c < 0.5$ [4] and $|D|F|$ being the number of rows of the sample D satisfying the filter predicate F.

VizCertify criterion

$\Longrightarrow$ Based on the number of samples satisfying F_R and F_C we can compute ϵ_R and ϵ_C for a chosen confidence level δ:

$$\epsilon_R := \frac{d + \log_2 \frac{1}{\delta}}{2|\mathcal{D}|F_R|}$$
$$\epsilon_C := \frac{d + \log_2 \frac{1}{\delta}}{2|\mathcal{D}|F_C|}$$

$\Rightarrow$ with probability of at least $1 - \delta$ we have that $\forall x_i \in dom\, X$: $|p^R(X = x_i) - \hat{p}^R(X = x_i)| < \epsilon_R$ and $|p^C(X = x_i) - \hat{p}^C(X = x_i)| < \epsilon_C$

$\Rightarrow$ if $|\hat{p}^R(x_i) - \hat{p}^C(x_i)| > \epsilon_C + \epsilon_R$ then with probability of at least $1 - \delta$ it holds $p^R(x_i) \neq p^R(x_i)$ for at least one x_i.

Controlling FWER

Family-wise error rate: Probability of making at least one false discovery.

$\Longrightarrow$ VizCertify controls FWER at level δ, because

$$\mathrm{FWER} = \Pr\left[\{|\hat{p}_k^R - p_k^R| \geq \epsilon_R\} \cup \{|\hat{p}_k^C - p_k^C| \geq \epsilon_C\} \cup \{|\hat{p}_k^R - \hat{p}_k^C| \geq \epsilon_V\}\right]$$

but when ϵ_R and ϵ_C are picked according to the VizCertify procedure of the previous slide

$$\mathrm{FWER} < \delta.$$

VC dimension of query class

For practical use, sufficient to bound VC dimension of query class Q.

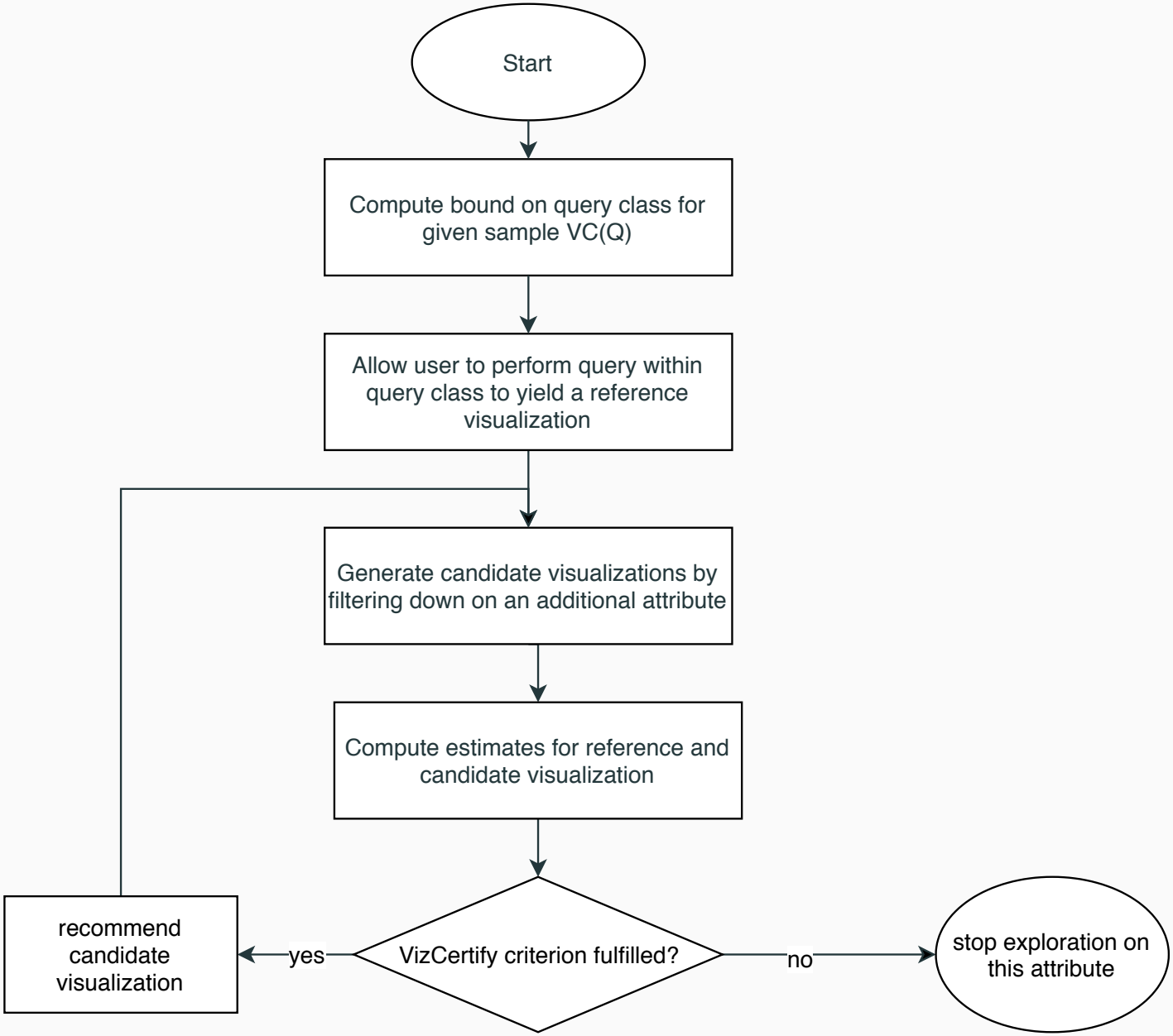
$\Rightarrow$ Bound depends on the type of queries which are supported within a system.

$\Rightarrow$ **Example:** Query class which only allows to filter on A_1, A_2 and queries of the form

```
SELECT X, COUNT(*) FROM survey WHERE A_1 < a_1 AND
A_2 BETWEEN a_2^l AND a_2^h
```

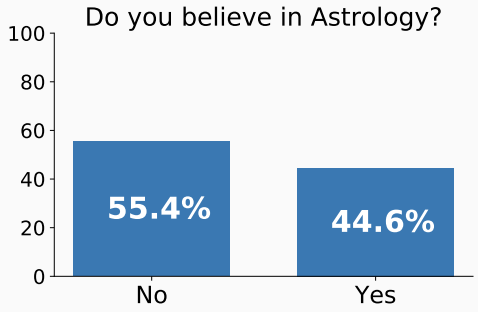
has a VC dimension bound by $VC(Q) \leq 1 + 2$

VizCertify algorithm

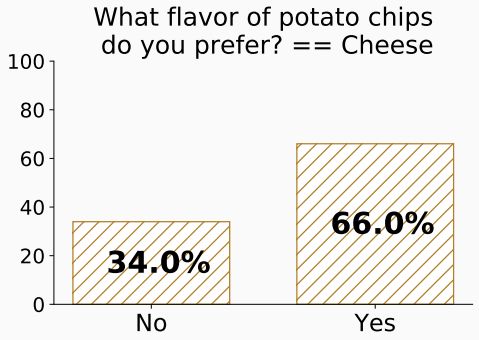


Example

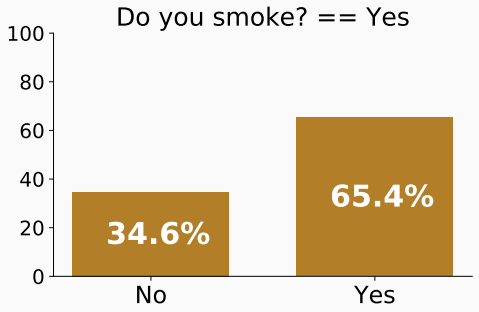
Amazon mechanical turk survey with $2,644$ participants from the U.S., 32 (mostly) unrelated questions ($VC(Q) = 50$).



reference visualization



top 1 (unsafe)

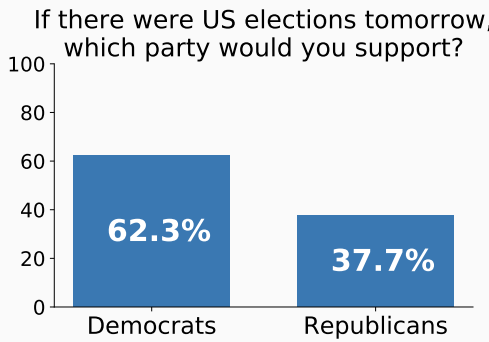


top2 (safe)

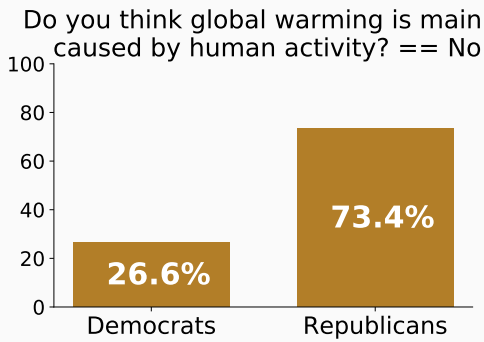
$\Longrightarrow$ VizCertify flags the top recommendation as unsafe and would not recommend it!

Another example

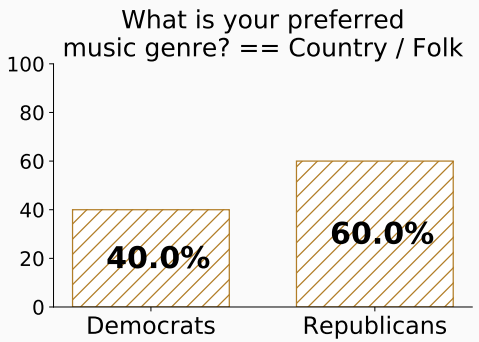
Amazon mechanical turk survey with $2,644$ participants from the U.S., 32 (mostly) unrelated questions ($VC(Q) = 50$).



reference visualization



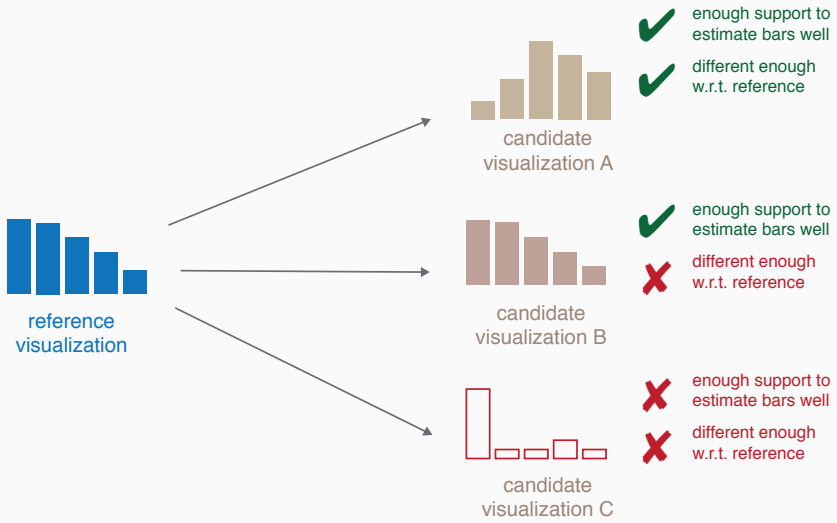
top 1 (safe)



top2 (unsafe)

$\Longrightarrow$ Can't conclude that taste in Country music is influential to party support, however belief in global warming is.

VizCertify



⇒ allows to ensure that a recommended visualization is *safe* to recommend by ensuring that

1. estimates are accurate.
2. differences are discernible.

⇒ allows to prune the search space.

⇒ allows safe interactive exploration in an upfront defined query space.

Thank you!

References i

F. Chirigati, H. Doraiswamy, T. Damoulas, and J. Freire.
Data polygamy: The many-many relationships among urban spatio-temporal data sets.
In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 1011–1025, New York, NY, USA, 2016.

S. Har-Peled and M. Sharir.
Relative (p, ε)-approximations in geometry.
Discrete & Computational Geometry, 45(3):462–496, 2011.

References ii


K. Z. Hu, M. A. Bakker, S. Li, T. Kraska, and C. A. Hidalgo.
Vizml: A machine learning approach to visualization recommendation.
In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, page 128, 2019.


M. Löffler and J. M. Phillips.
Shape fitting on point sets with probability distributions.
CoRR, abs/0812.2967, 2008.


M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis.
Seedb: automatically generating query visualizations.
Proceedings of the VLDB Endowment, 7(13):1581–1584, 2014.

Backup

Bounding sample error using VC dimension

Definition range space: A range space is a pair (X, R) where X is a (finite or infinite) set and R is a (finite or infinite) family of subsets of X. Members of X are called *points* and those of R are called *ranges*.

Absolute approximation: For a range space (X, R) and $0 \leq \epsilon \leq 1$ a finite subset $S \subset X$ is an absolute ϵ-approximation for X if for all $r \in R$:

$$\left| \frac{|r|}{|X|} - \frac{|S \cap r|}{|S|} \right| \leq \epsilon$$

$\implies$ Using $p_k = \frac{|F|}{|\Omega|}$ and $\hat{p}_k = \frac{|\mathcal{D} \cap F|}{|\mathcal{D}|}$ with $\mathcal{D}$ being the sample available from a universe Ω and F a range corresponding to a filter allows to bound the sampling error.

Sample complexity

Theorem [2]: Let (X, R) be a range-space of VC-dimension at most d, and let $0 < \epsilon, \delta < 1$. Then, there exists an absolute positive constant $c > 0$ s.t. for any random subset $S \subseteq X$ it holds:

$$|S| \geq \frac{c}{\epsilon^2}\left(d + \log_2 \frac{1}{\delta}\right) \implies \left|\frac{|r|}{|X|} - \frac{|S \cap r|}{|S|}\right| \leq \epsilon$$

for any $r \in R$, i.e. S is an ϵ-approximation of X.

$\implies$ experimentally shown that $c < 0.5$ [4].

VC dimension

Let (Ω, Q) denote the range space with a global domain Ω and Q being the query range space, whose VC-dimension is bound by d.

Corollary: Let $\mathcal{D} \subseteq \Omega$ be a random sample from Ω, then for given $0 < \delta < 1$ for any visualization $\mathcal{V}$ corresponding to a filter predicate F it holds $\exists c > 0$ s.t.

$$Pr\left[\left|p_k^F - \hat{p}_k^F\right| \geq \epsilon\right] < \delta$$

when

$$\epsilon \geq \sqrt{\frac{c}{|D|F|}\left(d + \log_2 \frac{1}{\delta}\right)}.$$

VC dimension of query class

For practical use, sufficient to bound VC dimension of query class Q:

$$VC(Q) = \sum_{i=1,\ldots,m} 2\alpha_i + \beta_i$$

- $\Rightarrow$ α_i is the maximum number of (non-redundant) closed intervals, β_i is the maximum number of open intervals for attribute i.
- $\Rightarrow$ Can use SQL queries with filter predicates over m attributes using $\geq, \leq, =$ and $\neq$ which can be combined with *and* or *or*.
- $\Rightarrow$ to activate/deactivate a feature use dummy value $+\infty$.

Example: Query class which only allows to filter on A_1, A_2 and queries of the form `SELECT X, COUNT(*) FROM survey WHERE A_1 < a_1 AND A_2 BETWEEN a_2^l AND a_2^h` has a VC dimension bound by $VC(Q) \leq 1 + 2$