# Name Collision Analysis Data Exploration

# Agenda

**“Can we predict what strings are going to make the Internet go boom and is there a way to mitigate any of these if we do discover them?”**

1. Examine data attributes for evaluating collision strings.
2. Some basic “Data Science” - exploring patterns, similarity, and clustering.
3. Start deliberating how we incorporate this data exploration & case studies back into guidance that we must provide for the Board’s 10 questions.

# Data Attributes When Evaluating Collision Strings

**Traffic Properties:**

- Network diversity
  - Number of unique ASNs, /24s, etc.
  - Distribution of traffic (e.g. heavily weighted in a few ASNs)
- Geographical diversity
- Qtype distribution
- Query volume
- Longitudinal trends

**Qname and Labels:**

- Distinct SLDs
  - Distribution of traffic over SLDs
- Amount of “noise” (e.g. Chromium)
- SLDs appear to be delegated TLDs
- First label features
  - DNS-SD
  - Common protocols
- Qname Minimization effect

**Other Attributes:**

- The string’s context
- OSINT of string being used
- Data sensitivity and catchment of data collector

# Patterns, Similarity, and Clustering

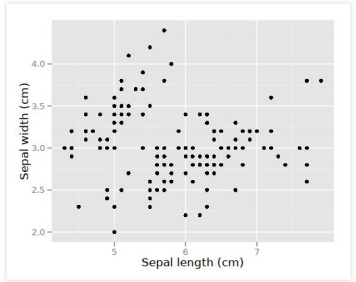
- Do collision strings exhibit some type of common pattern?
  - If string X demonstrates a propensity to incur risk are there other strings similar to X?
  - If string Y was known to cause collisions are there other strings similar to Y?
- Use data to measure features/attributes
  - This is the tricky part. What to measure? How important is that measurement (information gain)? How can that measurement be used to compute similarity?
- Define a distance function and measure how close (i.e. similar) or distant (i.e. dissimilar) are strings.
  - This distance measurement then lets us perform grouping or clustering algorithms (e.g. K-means, Heir aglom cluster, etc.)
  - Warning: You always get an answer with unsupervised learning...It’s the qualitative assessment of the answer that we need to consider.

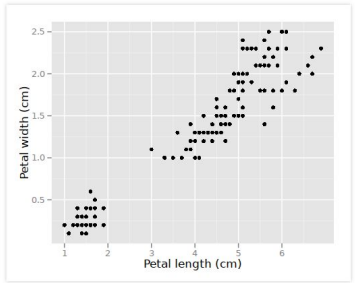
# K-Means Cluster Overview

- Clustering is used for finding groups or "clusters" of data for which the true groups/labels are unknown.
- k-means is an iterative algorithm which assigns cluster "centroids" (an average of the points that make up a cluster) and then reassigns points to the new cluster-centroids. The algorithm stops when points don't change their cluster assignments.
- k-means requires deciding upfront the value of *k*

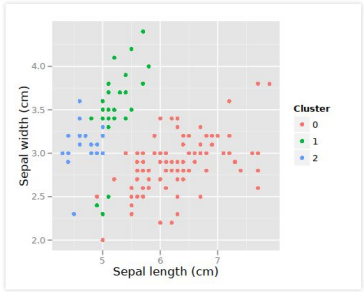
| Sepal length (cm) | Sepal width (cm) | Petal length (cm) | Petal width (cm) |
|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 |
| 4.9 | 3.0 | 1.4 | 0.2 |
| 4.7 | 3.2 | 1.3 | 0.2 |
| ... | ... | ... | ... |

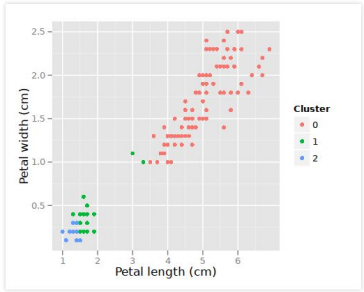
Measurements of Various Iris Flowers




K-Means Cluster (K=3)




Images from: http://web.cse.ohio-state.edu/~stiff.4/cse3521/k-means.html

# Collision String Data Features

- Based on data attributes we discussed, the following data features were calculated on a per string basis for a ~80 top traffic TLDs (delegated and non-delegated)
- In total this will contain 32 measurements for a given string.

1. Fraction of Qtypes: "A", "AAAA", "MX", "PTR", "SRV", "SOA", "DS", "DNSKEY", "NS", "Other", "RD"
2. Fraction of single queries (e.g. the qname was only received once)
3. Cumulative fraction of traffic accounted by top ASN, top ten ASNS, and top hundred ASNs
4. Fraction of query labels containing 1, 2, 3, and 4 labels
5. Fraction of queries for WPAD, ISATAP, MSOID, WWW, NS1, NS2
6. Fraction of queries that exhibit Chromium pattern [a-z]{7,15}
7. Fraction of queries with SLDs that are delegated TLDs
8. Fraction of SLD traffic percentiles at 50th, 80th, and 95th percentile

# Collision String Similarity/Distance Measurements

- The classification of observations into groups requires some methods for computing the distance or the (dis)similarity between each pair of observations.
- Define clusters such that the total intra-cluster variation is minimized

**Euclidean distance:**

$$d_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

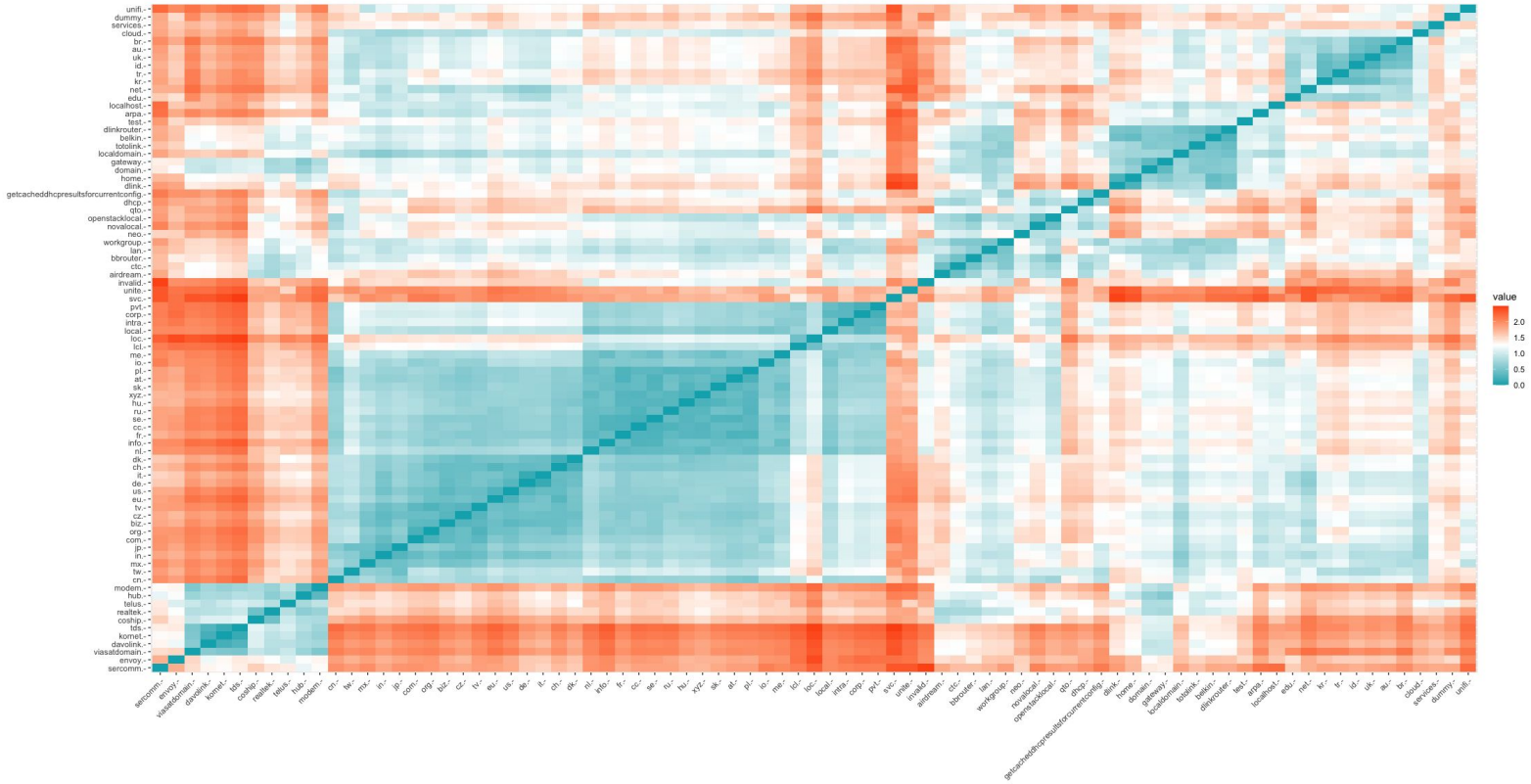
**Manhattan distance:**

$$d_{man}(x,y) = \sum_{i=1}^{n}|(x_i - y_i)|$$

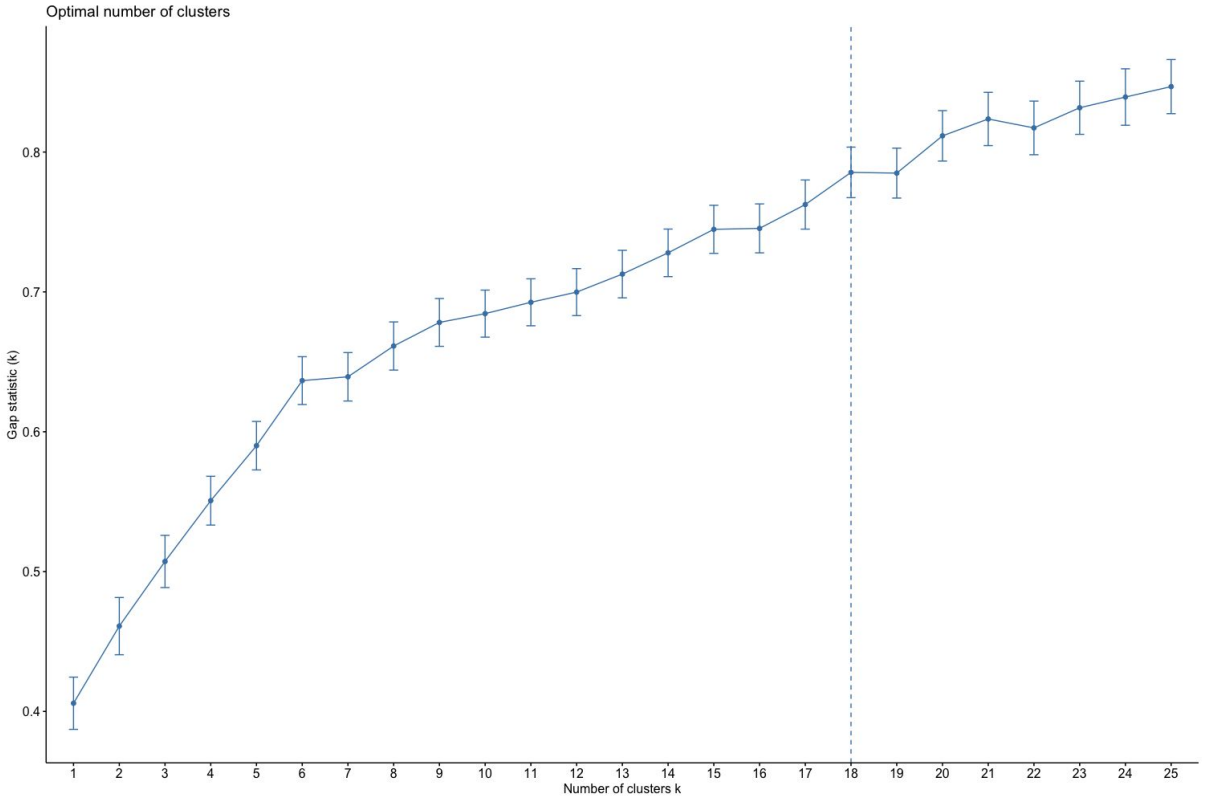
$$W(C_k) = \sum_{x_i \in C_k}(x_i - \mu_k)^2$$

$$minimize\left(\sum_{k=1}^{k} W(C_k)\right)$$

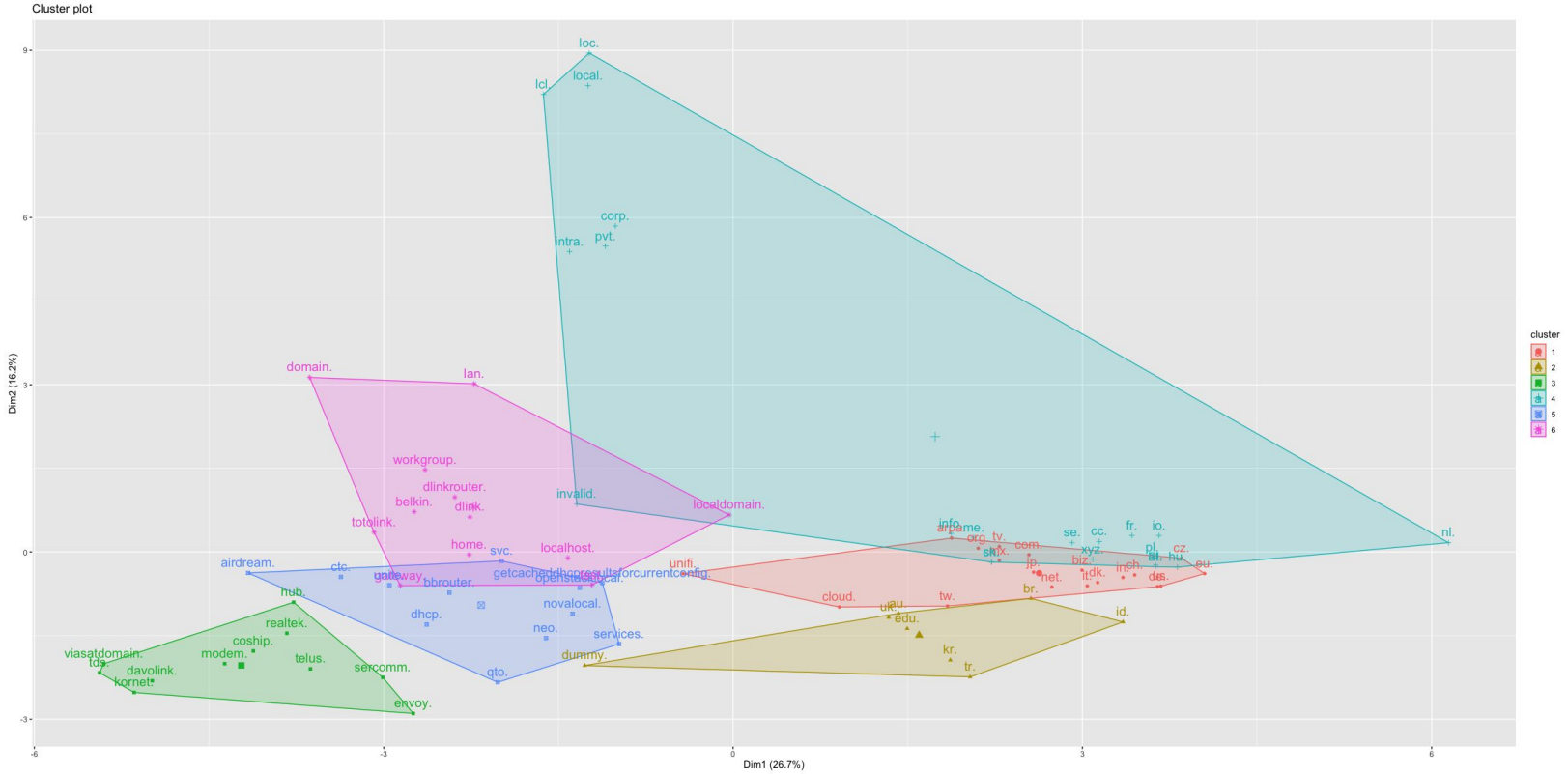
# Collision String Similarity/Distance Measurements



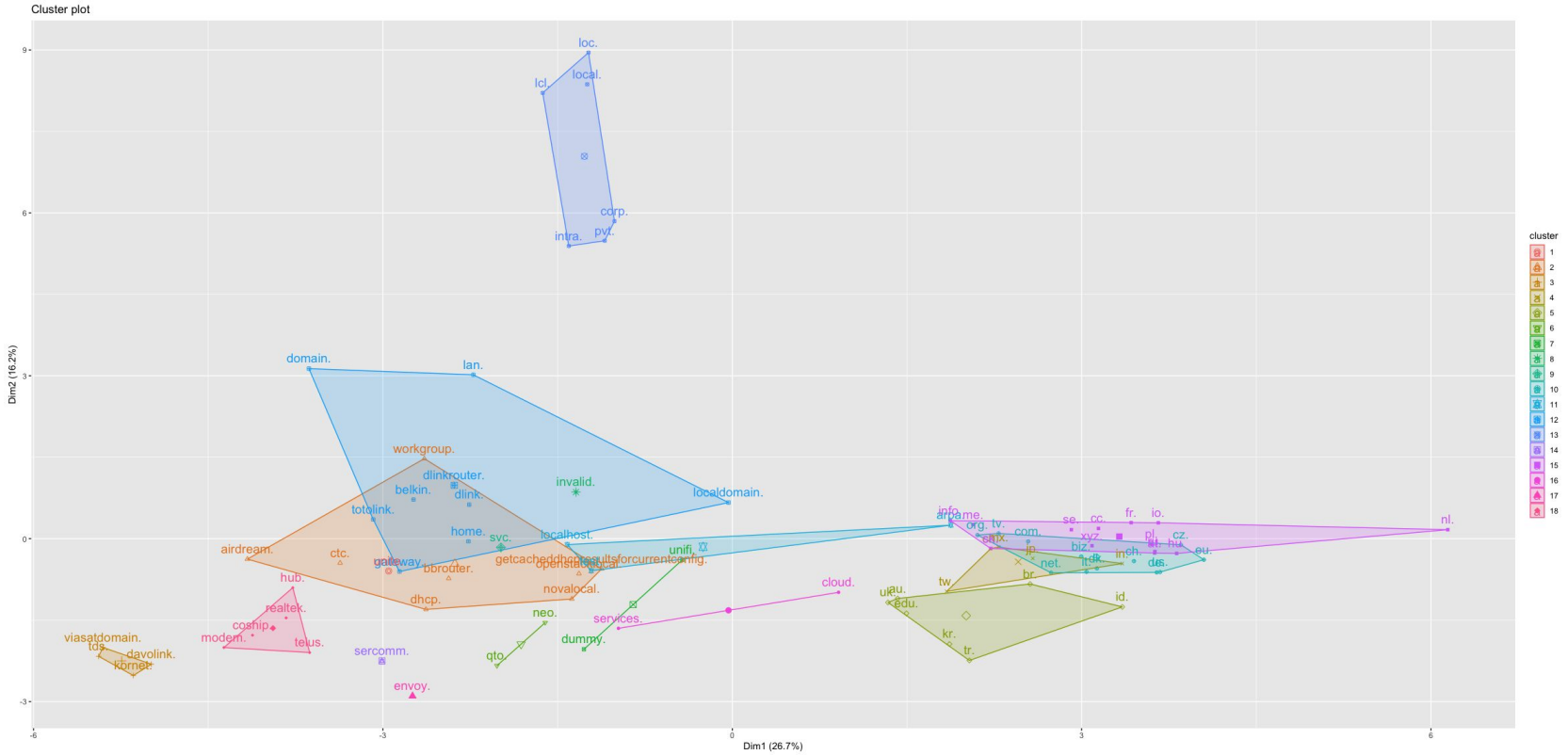
# Collision String Elbow Test



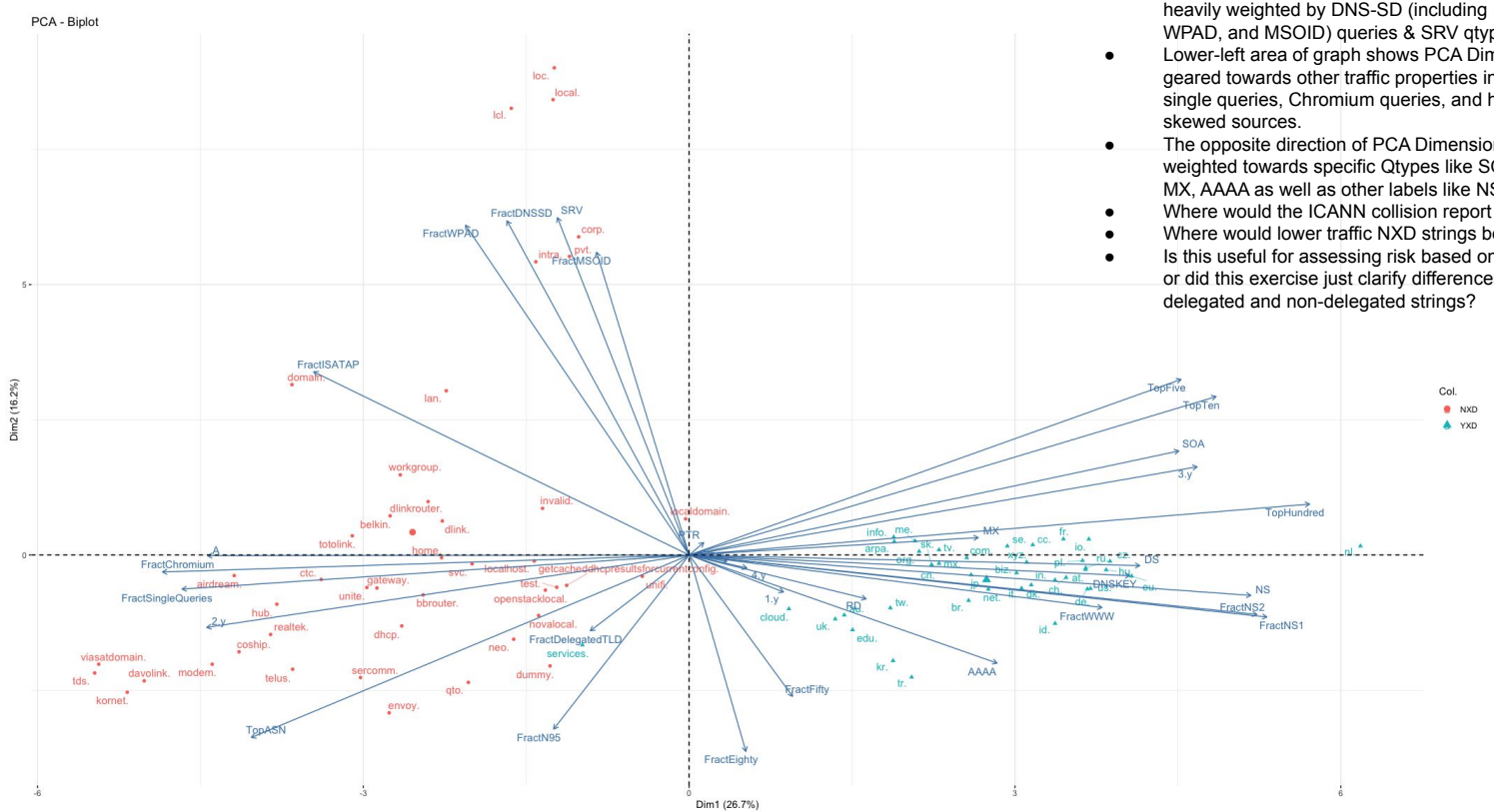
# Collision String K-Means



# Collision String K-Means

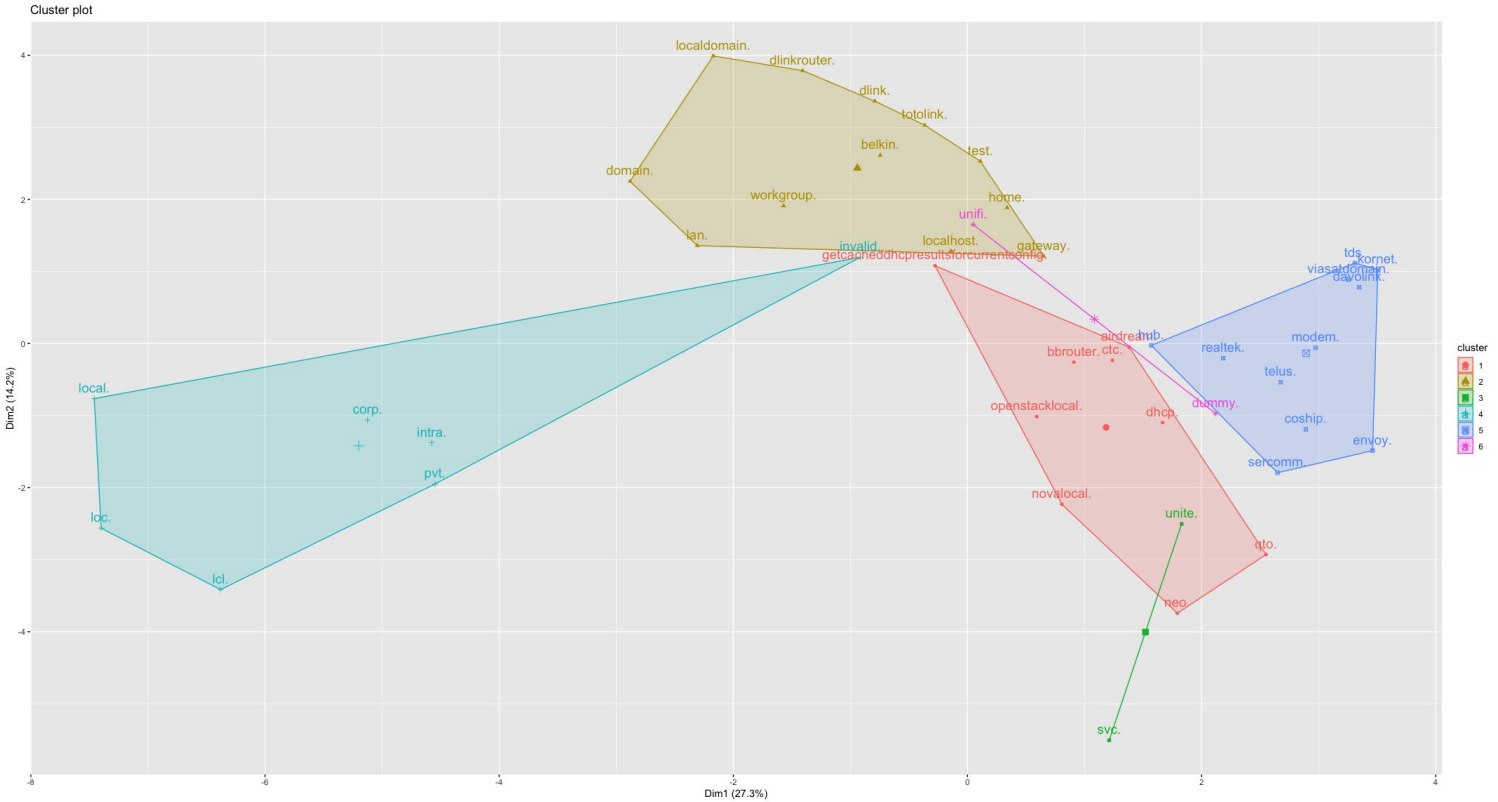


# Collision String Prin. Comp. Analysis

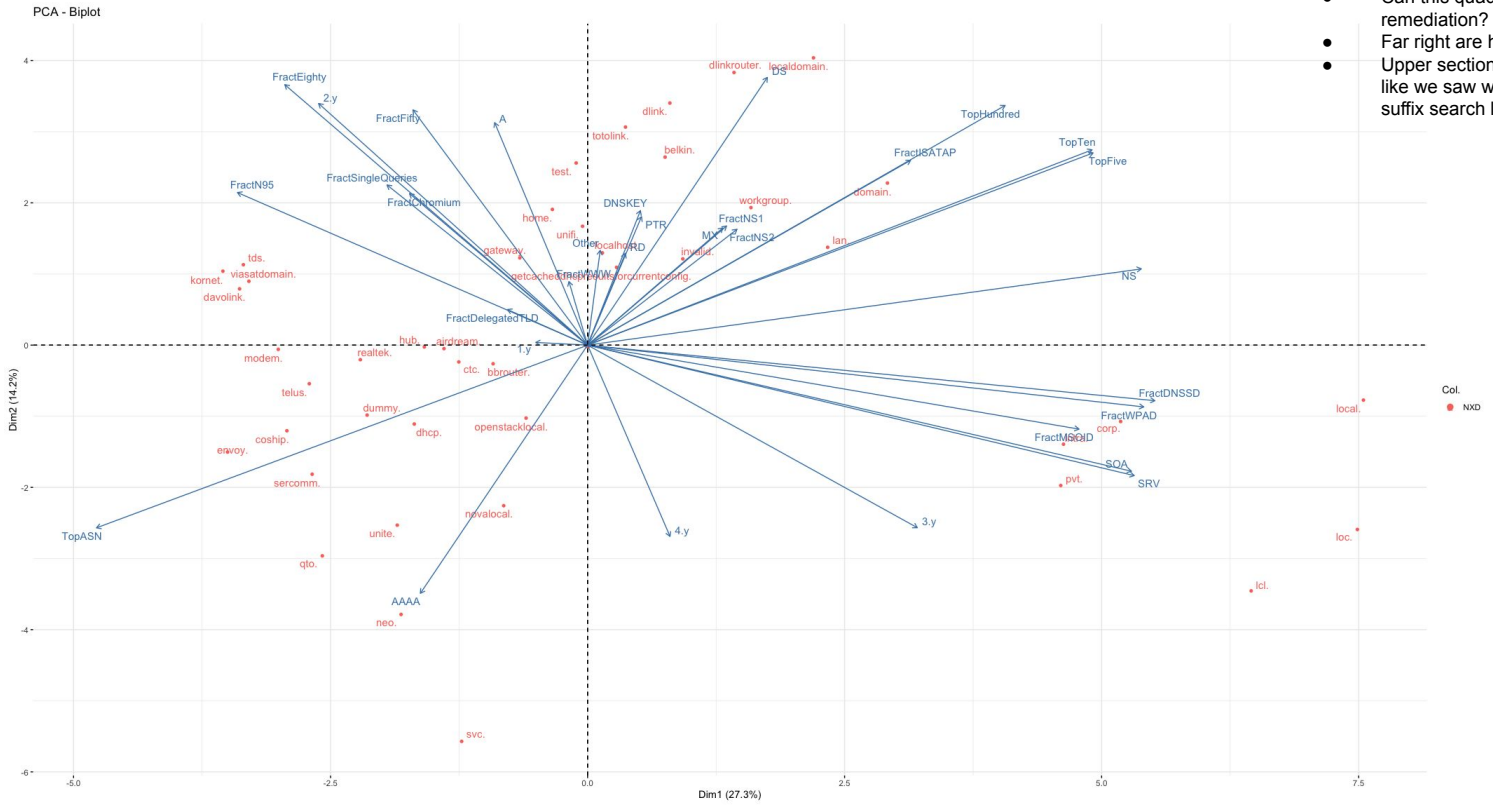



- Arrow magnitude represents how important (e.g how much variance it can explain)
- Red are non-delegated TLDs. Blue are delegated.
- Upper-left area of graph shows PCA Dimension-2 is heavily weighted by DNS-SD (including ISATAP, WPAD, and MSOID) queries & SRV qtype.
- Lower-left area of graph shows PCA Dimension-1 is geared towards other traffic properties including single queries, Chromium queries, and heavily skewed sources.
- The opposite direction of PCA Dimension-1 is weighted towards specific Qtypes like SOA, DS, MX, AAAA as well as other labels like NS1, WWW.
- Where would the ICANN collision report strings be?
- Where would lower traffic NXD strings be?
- Is this useful for assessing risk based on proximity or did this exercise just clarify differences between delegated and non-delegated strings?

# Collision String K-Means - Undelegated Strings



# Collision String Prin. Comp. Analysis




- Heavily skewed “single ASN originating” strings in lower left.
- Have contacted one of the ISP related strings already based.
- Can this quadrant be used for outreach / remediation?
- Far right are heavily focused on DNSSD
- Upper section contains more DNS query structure like we saw with delegated TLDs.  Is this due to suffix search lists?

# Board Questions

**Board Questions**

(1) a proper definition for name collision and the underlying reasons why strings that manifest name collisions are so heavily used;
(2) the role that negative answers currently returned from queries to the root for these strings play in the experience of the end user, including in the operation of existing end systems;
(3) the harm to existing users that may occur if Collision Strings were to be delegated, including harm due to end systems no longer receiving a negative response and additional potential harm if the delegated registry accidentally or purposely exploited subsequent queries from these end systems, and any other types of harm; (Mainly reliant at this point on the summary of Study 1 Report … Other thoughts?)
(4) possible courses of action that might mitigate harm; (Study 3)
**(5) factors that affect potential success of the courses of actions to mitigate harm (also Study 3);**
**(6) potential residual risks of delegating Collision Strings even after taking actions to mitigate harm;**
**(7) suggested criteria for determining whether an undelegated string should be considered a string that manifest name collisions, (i.e.) placed in the category of a Collision String;**
**(8) suggested criteria for determining whether a Collision String should not be delegated, and suggested criteria for determining how remove an undelegated string from the list of Collision Strings; and**
**(9) measures to protect against intentional or unintentional creation of situations, such as queries for undelegated strings, which might cause such strings to be placed in a Collision String category, and research into risk of possible negative effects, if any, of creation of such a collision string list.**

(10) to present data, analysis and points of view, and provide advice to the Board regarding the risks posed to users and end systems if .CORP, .HOME, .MAIL strings were to be delegated in the root, as well as possible courses of action that might mitigate the identified risks. (We did case studies….what does the group think?)