A Context-Free Method of Visualizing Streaming Object Data for the Purpose of
Identifying Known Events: an Implementation and Analysis

by

Quinn Gregory Carver

A thesis submitted to the College of Engineering and Computing at
Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Masters
in
Computer Science

Melbourne, Florida
December, 2017
We the undersigned committee hereby approve the attached thesis, “A Context-Free Method of Visualizing Streaming Object Data for the Purpose of Identifying Known Events: an Implementation and Analysis,” by Quinn Gregory Carver.

__________________________
David W. Mutschler, Ph.D., P.E.
Associate Professor, Major Advisor
School of Computing

__________________________
Jennifer Long, Ph.D.
Assistant Professor
Department of Mechanical and Aerospace Engineering

__________________________
Vernon Gordon, Ph.D.
Associate Professor
Department of Mechanical and Aerospace Engineering

__________________________
Philip Bernhard, Ph.D.
Associate Professor and Director of School of Computing
Abstract

Title: A Context-Free Method of Visualizing Streaming Object Data for the Purpose of Identifying Known Events: an Implementation and Analysis

Author: Quinn Gregory Carver

Advisor: David Mutschler, Ph. D.

This year, every second, five gigabytes of new data will be streamed to storage and yet less than half a percent of all this data will ever be analyzed. Much of this muted data is high-variety object data, unnoticed in a void of sorely needed tools to make it readily understandable. A software tool for visualizing streaming object data in a context-free manner can be built, and this tool would aid in finding predictor data for known events which are functions of the data. The design of the tool to support the hypothesis is presented and field results from over 500 test trials are analyzed. Score distributions from participants using the tool are compared to a random score distribution. Users did correctly identify more known event predictor data than they would have if selections were made at random. Therefore, the software tool is useful.
# Table of Contents

Abstract........................................................................................................................................ iii

Table of Contents............................................................................................................................... iv

List of Figures..................................................................................................................................... vii

List of Tables..................................................................................................................................... viii

List of Equations............................................................................................................................... ix

Acknowledgement............................................................................................................................. x

Dedication......................................................................................................................................... xi

Chapter 1 Introduction .................................................................................................................... 1

Chapter 2 Background ..................................................................................................................... 4

  Relevant Concepts.......................................................................................................................... 4

    Term: Known Event....................................................................................................................... 4

    Definition: Object.......................................................................................................................... 4

    Definition: Polymorphic ................................................................................................................. 6

    Term: Flattening or Un-nesting Objects ....................................................................................... 6

    Definition: Streaming ................................................................................................................... 6

    Definition: Context-Free, BNF and Recursive Grammar ........................................................... 7

    Term: Publish and Subscribe ....................................................................................................... 8

Related Work..................................................................................................................................... 8

  Visualizing Object Data.................................................................................................................. 8
Visualizing Time........................................................................................................... 12
Data Driven Documents (D3) ...................................................................................... 14

Chapter 3 Design of a Context-Free Streaming Object Data Visualization Tool ........ 15

Design............................................................................................................................ 15
Objectives ...................................................................................................................... 15
The Sentinel Use Case ................................................................................................. 15
The Test Use Case ........................................................................................................ 16
Depicting Object Data................................................................................................. 16
Representing the Stream ............................................................................................ 18
Test Use Case Enhancements ...................................................................................... 18
Implementation Decisions ........................................................................................... 19
Browser Based ............................................................................................................. 19
Data Source .................................................................................................................. 20
Rules ............................................................................................................................. 21
Data Driven Documents ............................................................................................ 22
Interactive Interface .................................................................................................... 22
Scoring .......................................................................................................................... 23

Chapter 4 Experiment ............................................................................................... 24
Training ......................................................................................................................... 24
Test Run ........................................................................................................................ 25

Chapter 5 Experiment Findings and Discussion ....................................................... 27
Summary Statistics ....................................................................................................... 27
Test Score Distributions .............................................................................................. 28
Distribution of the Foil ............................................................................................... 29
Evaluating the Hypothesis ......................................................................................... 30
Discussion: .................................................................................................................32
Chapter 6 Future Work & Conclusions ........................................................................34
Better Depiction of Hierarchies....................................................................................35
Trends Between Objects.................................................................................................36
User Interaction................................................................................................................36
Conclusions ....................................................................................................................36
References ......................................................................................................................37
Appendix A Data Collected for Test 1 ...........................................................................40
Appendix B Data Collected for Test 2 ............................................................................43
Appendix C Data Collected for Test 3 ............................................................................45
Appendix D Values Used and Computed for Calculating the Random Distribution .......47
Appendix E Data Visualization Source Code ...................................................................49
List of Figures

Figure 1 A play “firetruck” depicted as an object ................................................................. 5
Figure 2 Tree, Icicle plot, Pie Tree, Cascaded TreeMap, Sunburst, Radial Layout ............. 9
Figure 3 Example of a Treemap and Voronoi Treemap ...................................................... 11
Figure 4 Voronoi Treemap embedded in a 100-gon.............................................................. 11
Figure 5 Example of circle-packing with increasing hierarchy depth................................. 12
Figure 6 Pixel map showing changes over time in column row order.............................. 13
Figure 7 Embedding clock glyphs to show network activity over a 24 hour period .......... 14
Figure 8 Example of a how an object is depicted............................................................... 17
Figure 9 Stream depiction ................................................................................................... 18
Figure 10 Six items in a datastream, a portion of the manifest for the RMS Titanic ......... 21
Figure 11 Example rule for determining if an object is a predictor for a known event....... 21
Figure 12 During the training phase large graphics encourage interaction ...................... 25
Figure 13 Test feedback is presented in the background and on the scoreboard............. 26
Figure 14 Comparisons of random data and test data distributions for all three tests ........ 28
Figure 15 Comparisons of the average random and test data distributions overall .......... 29
Figure 16 Graph comparing the Z-Scores for rejecting H0 with the actual Z-Score......... 31
Figure 17 Triptych of “foot”, “belly” and “head” regions of the graphs ............................... 32
Figure 18 A prototype written by the author for visualizing hierarchies in XML............. 34
Figure 19 Example of a Cluster Graph in D3 ...................................................................... 35
List of Tables

Table 1 Summary statistics for test run data.................................................................27
List of Equations

Equation 1 Example production rules for a context free grammar .................................. 7
Equation 2 Formula for the binomial distribution of N Bernoulli trials ......................... 30
Equation 3 Formula for computing the z score ................................................................. 30
Equation 4 Formula for computing the P-value ............................................................... 31
Acknowledgement

Thanks to my advisor Doctor David Mutschler for his guidance with my education at Florida Institute of Technology and for his assistance along with Doctor Vernon Gordon and Doctor Jennifer Long with this thesis project. Also thanks to the hard working staff at the Patuxent River offsite facility: Doctor Bob Schaller, Anita Kriner and Melissa Meeks. Thanks to Progeny Systems Incorporated and Coherent Technical Services Incorporated who through their work study programs have made this education possible. Lastly, thanks also to Miko Wade (Lieutenant Commander, US Navy) who was tenacious in insisting I return to graduate school.
This paper is dedicated to the first and last women that I have loved. My mother, her 10th grade biology lab full of skeletons, pickled scorpions and compound binocular microscopes may be the reason I am a scientist today, and ultimately to my wife, who has been a constant and a fortress for the family while I have had my nose in all of these books!
Chapter 1
Introduction

With the amount of unique data archived doubling every two years, estimates are that the world will store nearly 145,000 Petabytes of data in 2017 (Rydning 2014). Roughly 5 new gigabytes of new data per second will be streamed to storage and yet less than half a percent of all this data will ever be analyzed (Regalado, 2013)! The challenges of Big-Data are volume, velocity and variety; the "variety challenge" has emerged and object data, recursively defined data types with nested fields, are a major cause (Bean, 2016). These high-variety information formats include: Javascript Object Notation (JSON), relational, non-Structured Query Language (NoSQL), Avro, Parquet, and XML and others. Spotting trends in (flat) parametric data plotted over time, for instance, in x-y plots, pie-charts, or bar graphs is common. Object data, however have no such visualization panacea. In a fashion complementary to parametric data, a purely generic way to visualize and understand time domain object data is sorely needed.

This paper is constructed to evaluate a hypothesis which addresses this challenge. A software tool for visualizing streaming object data in a context-free manner can be built, and this tool would aid in finding predictor data for known events which are functions of the data. This tool offers a unique method for viewing object data. No configuration or a priori context clues are required. It is especially applicable and relevant for uses which require a view of high-variety data before a tailored interface can be constructed. Background information, presentation of the software tool and results from the test trials which support the hypothesis are given in the chapters that follow.

Chapter 2 begins by arming the reader with a kit of terms used throughout the paper: objects, polymorphism, nesting, flattening, streaming, context-free grammars and publish and subscribe patterns are briefly discussed. A discussion of previous related work follows. Techniques for visualizing object data over time, and static techniques which can be adapted, are reviewed for their relevant merits. Techniques reviewed include: trees, treemaps, measures of success for treemaps, Voronoi treemaps, circle-packing, visualizing
Chapter 3 details the use cases, design and implementation of the software tool used to test the hypothesis. For the first use case, the “sentinel” role, a user monitors object data streams as an aid in predicting known events. For example, a sentinel might be a financial analyst monitoring object based market data in real time. The other use case, the “test” use case, provides a role for the participants to quantitatively test the effectiveness of the tool. Objects will be represented by groups of circles, each endowed with visible attributes to indicate information about their ancestry, field names and values. The data stream is depicted as an animation of horizontally staggered objects moving from left to right. Implementation choices selected for the application environment of the tool, data provisions, stream input, the interface, and test scoring are explained. The tool is designed to work with data formatted for JSON. The source code for the software can be found in Appendix E.

Chapter 4 is a walk-through of the experiment used to test the hypothesis. Approximately 184 participants used the tool with three different static sets of data for a total of over 500 test runs. Before each of the three tests in the trial, the participant is presented with an introductory slide and an interactive training period. This training familiarizes the user in how to select objects indicative of known events and what to look for. A brief two minute test run immediately follows training. The tool is scored based on how well the participant can observe the conditions which cause the known event. The score is tallied on a computer server for use in the experiment results.

Chapter 5 gives the experiment results and findings. Summary statistics of the mode, median and mean scores of the participants show that the tool was useful. Distributions of the scores are graphed. Scores are mostly in the greater third of possible test scores, however small distributions of low scores do exist. To confirm that the test data does indeed aid in finding predictor data for known events, a foil and its null hypothesis are proposed to provide evidence to the contrary. The foil is a computed distribution of what the test scores would be if participants had selected answers at random. How the foil distribution is computed is discussed and graphed and the factors and values are provided.
in a table in Appendix A. The Z-score of the data as compared to the foil is determined to be 3.61 which surpasses the Z-score necessary to reject $H_0$ and accept the alternative hypothesis $H_1$. That hypothesis states that the tool indeed does not give random performance. As the data is not random and the mean far exceeds the random behavior we conclude that the software tool is in fact “useful”.

Chapter 6 is an informal discussion of the results of the test run and possible explanations for the slightly different distributions observed. In short, users that were ambivalent in the first test seemed to polarize and either “got it”, or became more confused and exhibited worse than random behavior by the last test run. Word-of-mouth feedback from some users is also given.

Chapter 7 provides three ideas for future work. These follow-on work ideas include: improvements for depiction of hierarchies, a tool which shows trends in objects as a derivative over time and not the specific objects themselves, and lastly work for a tool which would include more user interaction. The chapter concludes with closing remarks which recap the performance of the tool.

References and Appendices follow the closing remarks.
Chapter 2
Background

Background information is presented to aid in understanding the problem in its context. The terms: object, polymorphism, nesting, streaming and context-free grammars are explained. Following these definitions, techniques for visualizing object data over time and static techniques which can be adapted are reviewed for their relevant merits.

Relevant Concepts

Term: Known Event
The term “known event” is used in this body of work to indicate an event which occurs as a result of factors which are known. A known event might be predicted if the event is likely to occur given a certain set of known circumstances. For example, we can predict that an individual will pack an umbrella if the weather report calls for rain. In this case, the known event predicted is that the umbrella is brought. Data which suggests that the weather calls for rain is predictive data which suggests the known event. Real world known events are predictive and occur stochastically to some degree. In a controlled test environment, data might actually be indicative (not predictive) of a known event. This means that a known event can be deterministically known given indicative data. Take for example “2 + 2 = 4”, the sum being four is a known event when both addends are two.

Definition: Object
The word “Object” has a special meaning in the software community. An object is a collection which contains attributes and has relationships with other objects. Attributes within objects, including attributes which are also objects, are ordered like an n-ary tree. The root node and each interior node of the tree is a sub context for child nodes. Each child node is an attribute which further describes the parent node. Leaf nodes contain primitive attribute such as words, numbers, enumerations or Booleans. The object describes a thing which may be tangible or intangible.
Figure 1 An AMF 508 “Fire Fighter” depicted as an object in a partially completed tree diagram. The values in red are leaf nodes which represent primitive attributes. (speedwaymotors 2012)
Definition: Polymorphic

An object is considered “polymorphic” if the design of the object is abstract enough to describe different kinds of similar objects which have some basic attributes in common. Polymorphism happens when the definition of one object is a part of the definition of another object. By analogy, a skateboard, a car and a truck are all types of vehicles. When we define any of these objects, we know that they all have traits of being a vehicle in common. A vehicle is a polymorphic object.

Term: Flattening or Un-nesting Objects

Data within an object can be “un-nested” or “flattened” and expressed in a tabular form, for example like in a spreadsheet. This is often impractical because a spreadsheet would have to include columns for all polymorphic types of every possible instance of every possible object field. Furthermore, more descriptive information would have to be added to show the relationships between columns. Object types, and relationships between objects are themselves metadata describing the object.

Definition: Streaming

Datastreams arrive from remote sources. Internet stream sources include Rich Site Summary (RSS) feeds, or a polled web-service or other streaming service publishers. A stream originates from a sequence of information elements $X_1, X_2, \ldots, X_k$ where $k$ is not known and the order in which the elements are received is not guaranteed (Abdulla, Arrighi, Critchlow 2003). Similar to Ethernet packets, a stream is typically sent in order but the order of receipt may be shuffled in comparison. As $k$ is unknown, datastreams may be of an infinite length, and therefore difficult or impossible to archive (Soares, dos Santos, Naldi 2015). Streaming object data is simply streaming data where each element of data received is an object.
Definition: Context-Free, BNF and Recursive Grammar

A production rule is a symbol substitution that can be performed to generate new symbol sequences. A formal grammar is a set of production rules defining a formal language.

\[ S \rightarrow AA \]
\[ A \rightarrow \alpha \]
\[ A \rightarrow \beta \]

**Equation 1** An example of a production rules for a context free grammar. S can produce the terminal expressions \( a \ a \) or \( \beta \ \beta \) or \( a \ \beta \) or \( a\beta \) or \( \beta a \)

A non-terminal symbol (such as \( S \) and \( A \) in Equation 1 above) is a symbol which has a production rule. A terminal symbol (such as \( \alpha \) and \( \beta \) in Equation 1 above) is a symbol with no production rules and is therefore static. A non-terminal symbol residing on the left can be defined by more non-terminal symbols or a terminal value on the right. This is the case with “Context-Free” grammars, terminal symbols only exist on the right side of the rule.

The Backus-Naur Form (BNF) is a popular example of a Context-Free grammar which is also a recursive grammar (Rosen 2011).

Recursive grammars have the distinction that the production of a non-terminal expression may include the same non-terminal that is produced from. In the example of Equation 1 above, a language which had a production rule of \( S \rightarrow S \) would be a recursive grammar because the term on the left includes itself in the production on the right.

Modern examples of recursive BNF grammars include eXtensible Markup Language (XML) and JSON. These are most typically used for expressing object data in streams. Web feeds, RSS and Atom syndications, and Simple Object Access Protocol (SOAP) all use BNF grammars for data transfer.

To avoid confusion, note that “context-free” as used in the hypothesis of this work is pseudo-synonymous with “universal”. The hypothesis proposes that a tool can be made to view an object in a context-free manner. By this, what is meant is *any* object which can be
defined by a context-free grammar can be visualized.

Term: Publish and Subscribe

“Publish and subscribe” is a software architecture pattern sometimes called “pubsub”. The subscriber requests that the publisher provide information as it becomes available. The publisher provides information to all the subscribers when it has information to share. Publish and subscribe architectures can be used inside components of a single software application, between software applications and over networks. Data streams are often provided by a publisher and consumed by a subscriber.

Related Work

The tool proposed visualizes data arriving in a datastream. Though the passage of time and advances in a datastream are not synonymous, a datastream can contain a historical record. With this minor concession, methods for graphing object data over time are adapted and used as related works. Techniques for graphing static object data are also discussed if they have been a reference for the software tool developed in this work.

Visualizing Object Data

Though analysis of abstract data is not intuitive, (Soares, dos Santos, Naldi 2015) visualization can generically be facilitated by clustering similar attributes and by detecting changes in them. Research by (Schultz, Hadlik 2015) finds that one can indeed make visualization rules to map hierarchical data to enumerated stereotypes of graphs or a blending of those stereotypes to make a new type. Graph types in the study included Cartesian and polar variations of: treemaps, cascaded treemaps, icicle plots, nested treemaps, pie trees, radial trees, sunbursts and n-ary trees. All of these graphs are variations on a “tree” theme where parent elements are symbolically connected to child elements.
Trees

Perhaps the most generic way to visualize hierarchical data is as a tree. To review, a tree consists of a collection of nodes, connected by directional arcs (Budd 1997). At the top of the tree diagram is a single root node. The node which has no parent is referred to as a root node. A node which points downward to another node is referred to as a parent of the child node it points to.

Object data can be visualized with a tree. Parent nodes in hierarchical data trees represent objects which contain other objects. The objects they contain are represented by child nodes in the tree. The primitives, fields which are contained by objects, are the leaf nodes in a tree.

Trees can be stylistically depicted in many ways. An icicle plot is a tree where parent nodes are represented as rectangles with child nodes juxtaposed below them represented by smaller rectangles. A radial tree is a tree which fans out from a central root node, rather than graphing downward and outward. Radial trees child nodes move outward as widening
arcs. A pie tree or sunburst chart is a crossover between an icicle plot and a radial tree. Child nodes fan away from the central root in a circular manner. However, wedge shaped blocks are used in place of lines and nodes.

**TreeMaps**

Ordered Treemaps are space filling tree diagrams wherein root nodes are represented by rectangles which contain child node rectangles. The containment pattern continues recursively from the bounding root node down to leaf node rectangles. The area of the leaf node rectangle relates to the value of it. The areas of parent node rectangles are the result of their contents. Rectangles span alternating axes within their parent rectangle with respect to a pivot. The pivot is typically the largest child node. Node hierarchy is stressed by the boundary thickness of each parent rectangle or a color scheme.

Ideally, treemaps strive to meet three measures of success (Benderson, Shneiderman, Wattenberg 2002).

1. The areas should be as symmetric as possible as it is easier to compare chunky volumes than strips.
2. In the case of dynamic data, the areas in the treemap should move position as little as possible. This is important with temporal based frames or animation of the data where the observer will expect to see the same data in the same place from one instant to the next.
3. The readability, the amount of time the user has to scan the graph to find the data, should be short.

The design of the software tool explained in Chapter 3 of this thesis incorporates these three measures of success in its design. However, the software tool does not render treemaps. Even so, the supposition that these measures could apply to any area-based graphing tool would seem plausible.
Voronoi Treemaps

The Voronoi treemap algorithm produces a nested tessellation of convex polygons rather than squares and rectangles. The result is that a node’s aspect ratios are more isometric (chunkier). Given that isometric n-gons (where n > 4) are more symmetric than rectangles, the Voronoi treemap is superior to standard treemaps for estimating attribute size by the first measure of success for treemaps (see TreeMaps).

Some Voronoi treemap algorithms can place volumes in a stable layout which is a problem for Ordered Treemaps (Hahn et al, 2014). Since Voronoi treemaps are more stable than rectangular treemaps and stable placement is the second measure of success for treemaps, dynamic (animated) Voronoi graphs would be a better method for visualizing object data over time than a dynamic treemap. Unfortunately, computational complexity makes it difficult to use Voronoi treemaps in real time. In practice, rendering takes seconds per frame on web based platforms (Henry, Vines 2014).
Circle Pack

Similar to treemaps, large hierarchical data can also be visualized by circle-packing nested circles (Wang et al 2006). In this paradigm, child nodes are nested within a parent circle and the pattern is recursive. Sibling nodes are presented alongside one another.

![Figure 5 Example of circle-packing with increasing hierarchy depth](image)

Unlike treemaps, circle-pack graphs can use at most 85% (Specht 2016) of the space in the visualization to express attributes. This subtle inefficiency is due to the nature of the geometry of a circle. As per Descartes’ circle theorem (Mutanguha, 2016), given any three circles that are as close as touching, there can be a circle defined between them (and encompassing them for that matter). Both squarified and Voronoi treemaps leave no space between element boundaries.

Though circle-pack graph area is less efficient, Wang et al (Wang et al 2006) show that it is easier to understand the hierarchical structure of data in circle-packing visualizations than in rectangular form factor of tree-maps. Considering a circle as an $\infty$-gon, and therefore a kind of n-gon treemap, circle-pack outperforms other treemaps in the third measure of success for treemaps (readability) in use cases where the user is scanning the graph for hierarchical information.

Visualizing Time

A comprehensive summary paper (Kerracher, Kennedy, Chalmers 2014), which addresses methods for graphing multidimensional data, breaks techniques into a graphing solutions
In this study, the graph offerings for object data over time include time-slicing or embedding techniques.

Time-slicing methods include sequential or juxtapositional views, or stacked overlays depicting the changes in a third dimension. Animation is a sequential view using time itself, not space, to depict time. Pixel-based visualization is an example of a time slicing technique. A heat map represents attribute values at points in time as colors; they are scaled between “cold” and “hot” values. A study by (Krstajić Keim 2013) claim that pixel-based juxtapositional heat maps are useful for identifying recursive changes in attributes over time.

![Pixel map showing changes over time in column row order (Krstajić Keim 2013)](image)

Several suggestions are given for embedding time directly into a graph. Time may be represented as a glyph or small picture (Fuchs 2015). Time can also be embedded into a graph by merging time information with the node with color or some other style. Lastly, time can be embedded into a graph as a node itself or several nodes. Consider a bipartite graph illustrating time nodes on one side connecting to data nodes on another side.
Figure 7 Embedding clock glyphs to show network activity over a 24 hour period

Data Driven Documents (D3)

Unlike the topics of previous subsections in this chapter, Data-Driven Documents (D3) is not a type of graph. An Application Programming Interface (API) is a contract provided by a software component which exposes its resources for use by another software component. D3 is a flexible API written in Javascript for constructing graphs in web pages (Bostock, Ogievetsky, Heer 2011). Other APIs used for web based graphing include Processing, Raphael, Protovis (Bostock, Ogievetsky, Heer 2011) and Flare (J.Davies 2017). D3 is unique in that it is used just for graphing and that it works within the pre-existing HTML Document Object Model (DOM) Framework. D3 developers build visualizations using the standard webpage languages: HTML, Cascading Style Sheets (CSS) and Javascript. Developers include D3 into a page’s Javascript code. The API of this code provides interfaces to build and customize various types of charts or graphs and methods to update them.

A chart is updated as data driven events occur. A data driven event is an event which occurs when there are changes to the data being graphed. There are three such data driven events in D3: events for receipt of new data, events for changes to existing data, and events for the deletion of existing data. These events change the data and D3 handles updates to the graph. This data-driven approach which directly affects the DOM (the document) is the tenet defining D3’s namesake.
Chapter 3
Design of a Context-Free Streaming Object Data Visualization Tool

This chapter focuses on the design and implementation of a software tool developed to test the hypothesis.

Design

The tool must present the object data stream and represent it as feedback to the user. Input is information received through the network and output is a graph that the user can see.

Objectives

The purpose of the software tool is for a user to be able gain an advantage in predicting known events based on what is seen in any object data stream. However, in the context of this study, the objectives are two-fold. First, the tool must serve its purpose. Second, its effectiveness in that purpose must be testable.

The two different objectives underscore two different use cases. A “use case” describes the interactions that a user has in their role with a system. The first use case is important because it is the artifact of the body of work in this thesis. The second use case is simply a derivative of the first use case used to grade it.

The Sentinel Use Case

Design for the first objective should leave the observer open to view whatever object based stream data they would like to observe. This is the “Sentinel” use case. Through exposure to the data stream and observations that follow, users (sentinels) condition themselves to predict known events based on what is seen in the data. By analogy, consider stock market tickers. Ticker readers can make predictions about changes in the market. A known event such as a “Bear Market” may be predicted if certain securities fall below some level. In
this use case, the known event is probabilistic. In the analogy, a day-trader watching securities fall may predict a “Bear Market” with great confidence, but due to other factors, still be wrong.

The Test Use Case

The second use case is the “Test” use case. It is more rigidly constructed because it needs to have quantifiable results. The test use case provides an artificial known event determined by a rule in a dataset which is provided. In this case, the known event is completely deterministic. Determinism is necessary to avoid penalizing the user for making a good guess. For example, if the data set visualizes trending data toward heavy rain and the known event is whether or not to bring an umbrella, the user is graded on the decision to pack an umbrella, not on the rain outcome. Training and scoring features are implemented for this use case.

The following design discussions on the depiction of object data and how the stream of data is represented apply to both the sentinel use case and the test use case.

Depicting Object Data

The tool uses the areas of circles rather than rectangles to associate attributes with values. This design incorporates the first measure of success for treemaps, ease of estimating volume (see TreeMaps). Circle boundaries are isometrically perfect and so they are better for area estimation.

Unlike circle pack diagrams, the tool only plots attribute circles, not parents within parents. This is to overcome wasted space inherent to circle pack diagrams (see Circle Pack). Instead, lineage to parent nodes is shown through the color of the circle. The circle’s color is unique to the lineage of the node and any other sibling nodes. If the circles represent nodes that are part of an array, they are colored as though they are siblings. Only circles with the same parent will have the same color. Clusters of circles together represent the entire object.
Each circle has indicators to distinguish the fields they represent. The name of the field for an attribute is given a glyph unique to that field. Chat based emoji are used as glyphs.

Attribute values are rendered in one of three ways, numerically, as text, or as a Boolean.

Numeric values (see blue arrow in Figure 8 above) are distinguishable by:

- The radius of the circle. A bigger radius is a larger number.
- Internal regularly spaced rings are also used to help compare radius sizes.

Text attributes (see orange arrow in Figure 8 above) are distinguishable by:

- The radius of the circle. A bigger radius indicates a longer expression.
- Some characters from the expression are shown around the radius of the circle.

Boolean attributes (see black arrow in Figure 8 above) are represented by

- A smaller constant size circle
- If a Boolean value is true, the circle is present. Otherwise, it is not.

Figure 8 Example of a how an object is depicted

```json
{"person":{
    "name":"Mrs.John Bradley (Florence)",
    "sex":"female",
    "age":38},
"relationships":{
    "sibsp":true,
    "parch":false},
"ticket":{
    "fare":71.2833,
    "cabin":"C85",
    "embarked":"C"}}
```
Representing the Stream

Clusters of circles represent the datastream of objects that the visualization is subscribed to. Circles side-scroll from left to right across the browser window. Animation is used to show the objects as they travel and juxtaposition is used to show values of neighboring objects in the screen.

Figure 9 Stream depiction (arrows added to indicate movement)

The stream of data consumed as a sequence of objects is presented at a rate suitable for viewing. Each group of circles presented in the animation represents exactly one object in the datastream. The vertical placement of the objects as they scroll is not a function of the data. Objects are staggered vertically to take advantage of space that might otherwise clutter or overlap neighboring clusters to clutter or overlap.

The data’s graphic is placed off-screen to the left and an animation begins which moves the object to the right. Objects do not change appearance as they cross the screen because the tool only constructs the objects visualization once. Data leaves the visualization at the completion of the object’s transit off the right end of the screen.

Test Use Case Enhancements

The design described so far accommodates all the functionality needed to watch object data streams go by and for users to make decisions about the data they are viewing. However, more functionality is needed to gather test metrics used to evaluate the effectiveness of the tool. To control the test environment and record the user’s perceptions, the following
additions are made: provisions for repeatable data sets, deterministic rules for identifying a synthetic known event, an interactive interface to gather user feedback, a training mode to familiarize the user, and a score tracking mechanism.

Rather than a published stream, the test use cases uses data from a file to keep test runs consistent. Though the contents of the file itself are streamed from a place store on the internet the data is static unlike a real world sentinel use case.

A deterministic rule provided with the data configures a pattern that an object must have in order to be considered as an indicator for a known event. These set the pattern for the type of object the user is looking for.

The interactive interface allows the user to select the object with the qualities that identify it as a known event indicator. By selecting the object, the user gives insight into what she perceives as the indicating object. These selections can be scored as correct or incorrect and provide measures of the effectiveness of the tool.

As the trial participant has never seen the interface before, a short introductory training session is required. The training session needs to be automated in order to be universal in all trials. The training session should familiarize the user with how to use the tool in each test and coach the user as to what pattern they are looking for.

Finally, all the interactions of the user must be scored and archived. Scores from all trials are collected and post-processed and used to determine the effectiveness of the tool.

Implementation Decisions

Browser Based

The tool is internet browser based, not hosted in an application on a local machine. There are several reasons for this. The tool should be accessible to a wide variety of machines. Internet connectivity makes the application readily available to received data streams;
Universal Resource Identifiers (URIs) to stream publishers are readily used for input. Web based graphing framework tools already exist. Lastly, the web makes it easy to gather data to test the tool’s effectiveness.

There are also disadvantages to a web based design decision. A platform independent tool implies it is not compiled for a target; therefore, a scripted language is used. This makes an application slower. Operations in graphics APIs are also more limited. Lastly, with non-native applications, it is likely that the user experience will be somewhat inconsistent between any two platforms.

Data Source

JSON was selected as the data format because it is a context-free grammar and it is human readable. In the context of BNF languages, a schema is a predefinition of what an object can be. For spontaneity, JSON data has the advantage that it can be schema-less. XML “tags” are expressions that come before and after a value or attribute definition. JSON is more efficient than XML to transmit as it does not require tags which occupy bandwidth. JSON is also a very popular format currently and as its name implies, it works well with JavaScript

For a sentinel use case, the tool receives input data and registers as a subscriber to an object publisher. The publisher is inconsequential as long as it has a URI and publishes objects in JSON format as an array. This makes the tool available to visualize data universally from any publicly available internet data source that can publish a JSON array. For the sentinel case, the data publisher “PubNub” was used in the development of the tool. Pubnub (Blum, Greene 2010) is an internet service which provides a convenient way to connect private publishers and subscribers over the internet.

```json
{"person":{"name":"Braund Mr. Owen Harris","sex":"male","age":22},"relationships":{"sibsp":1,"parch":0},"ticket":{"fare":7.25,"cabin":"","embarked":"S"}},
{"person":{"name":"Cumings Mrs. John Bradley (Florence Briggs"
For the test use case, data is streamed from specific premade JSON arrays read from a server on the internet. The test use case was for 500 trials. Streaming from static data makes the trial run data identical and easier to grade the effectiveness of the tool.

Rules

For the test use case, each static data file contains an array of objects to publish. Preceding the array is rule (for example Figure 11) to use for determining the known event predictor. The rule is written in the syntax of a Javascript method. After receipt by the tool, the rule is dynamically interpreted in Javascript to return a Boolean value for the object. The rule is dynamically loaded to demonstrate that the use case is generic.

```
"eval_function":"return (obj.person.sex == 'female' &&
obj.person.age <= 26 && obj.ticket.fare > 30.0)"
```

Figure 11 Example rule for determining if an object is a predictor for a known event

When the rule is compared against an incoming object, a value of true marks the object as a known event predictor. For example, given a data set, there could be a rule “O.A.n > c and O.B.b = true and O.C.a != s” in order to qualify an object as a predictor. In this rule, O is an object, A,B, and C are subcontexts in the object and n, a, and c are fields. The fields are
of type number (n), string (s) and Boolean (b). These fields are compared to c (a constant) and s (a string) which are constants.

The example above (Figure 11) is from the RMS Titanic survivability test. The known event is the survival of the passenger. The rule in this context states that if the gender of the passenger is female and she is under 26 and paid over thirty dollars for her ticket, then she is survivor. A user clicking on an object matching this description would receive a point for choosing correctly.

**Data Driven Documents**

D3 was selected as the graphing “engine” to use for the tool. There are hundreds of D3 examples (Bostock 2017) and source code available from enthusiastic members of the community. D3 does not require that a “plug-in” be installed in a browser for use. Although D3 is an API that contains strong paradigms, it is a Javascript implementation and not a new language. Pages using D3 are written with the established languages HTML, CSS and Javascript. Lastly, D3 is very customizable. A developer that understands D3 and Javascript can develop entirely new kinds of charts as is needed by the tool we present.

A D3 circle pack graph was selected as the basic graph type from which to derive extra functionality. Features to use interactions with the mouse, side-scroll, disable circle packing, hide the root nodes, markup for field names, markup for field values and hierarchy colors were added by the author with Javascript code through the course of this project.

**Interactive Interface**

Input from the user is made possible by Javascript mouse handlers. When the mouse is clicked, the rule is evaluated for the object the user has clicked on (if any). Feedback is given to the user by painting a large red “X” for incorrect answers or a large green “✔” for correct answers. These items also appear on the right of the screen as objects leave the
visualization to give the user feedback on objects that may have been overlooked or were correctly not clicked on. In each case, correct or incorrect, the score for correct and incorrect answers are provided in the top middle of the viewing area.

Scoring

Scores are prepared as a JSON object. The object contains data about the correct and incorrect selections made by the user as well as the date, time, time zone, and a unique number identifying the test participant's computer (see appendices B, C, and D). The purpose of the last parameter is to ensure that a user had not taken a test more than once. The JSON object is published through pubnub from the participant's computer. Typical pub sub design requires the subscriber to be listening at all times for publications. A missed message is lost. Fortunately, during the course of this project, pubnub, added a history feature. This was useful as a score capturing (subscribing) computer did not have to be left on at all times to record scores. Scores publications are archived on the pubnub computer for subscribers up to one month later. A separate Javascript which is not part of the tool was written to subscribe to the history of scores.

The complete source code for the software tool is included in Appendix E. The tool is an internet served, browser based, Cascading Style Sheets (CSS), HTML page which uses custom Javascript implementation and D3 to interpret JSON array data.
Chapter 4
Experiment

Approximately 184 individuals were asked to participate in trials of the tool. Each participant used the tool in three short (two minute) test runs each with a different dataset, a total of over 500 test runs among all users. The first test featured data for prediction of cancer in patients. The second data set featured data for prediction of traffic accidents. The third data set featured data for survivability as a passenger on the RMS Titanic.

Figure 7 — Example of the introductory slide seen before beginning a test run

Training

For each of three tests, an introductory slide is given. The user is coached as to what
known event indicator data will look like. Following this description, the participant is shown five examples as they will appear in the stream. During this phase, large graphics are presented which encourage the participant to click on the examples of predictor objects (objects which comply with a rule for a known event). Clicking on the object will be used during the test run to indicate when the participant recognizes an object as a predictor. The training is complete when the user has made five correct choices. Following training the extra graphics are removed and the test begins immediately.

Figure 12 During the training phase large graphics encourage interaction

Test Run

The test run lasts approximately 2 minutes. During the test run, new objects are presented. They enter the view area from the left and move gradually off the view area to the right. Object visualizations continue to appear and move rightward as more objects from the stream arrive.

Ideally, the participant will interact by clicking new arriving objects that look similar to
known event predictor objects they have been trained to identify. Objects disappear from the view screen once they are clicked, or if they exit to the right. If the participant clicks on a matching object or leaves a non-matching object exit un-clicked positive feedback is given. If the converse occurs then negative feedback is given. As described in the Implementation section, feedback is presented to the user in the background as a large green “✔” or a large red “X”. A red and green scoreboard visible top and center is also updated.

![Score: 10: 6](image)

Figure 13 Test feedback is presented in the background and on the scoreboard

At the end of the test round, the score is published to a retrievable place. At the end of the first and second test round, the user is immediately taken to a slide formatted in the same manner as the slide before the first test. That slide tells the user what to look for in the next test. Following the slide, the pattern repeats. There is a short training period and a two minute test. At the end of the last test, the user is brought to a terminal screen, thanked for their participation, and a button is made available to forward an invitation to a friend or colleague to take part in the study.
Chapter 5
Experiment Findings and Discussion

Table 1 Summary statistics for test run data

<table>
<thead>
<tr>
<th></th>
<th>Test1</th>
<th>Test2</th>
<th>Test3</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sample Size</strong></td>
<td>184</td>
<td>163</td>
<td>160</td>
<td>507</td>
</tr>
<tr>
<td><strong>Mean %Score</strong></td>
<td>79.64%</td>
<td>87.60%</td>
<td>76.78%</td>
<td>81.30%</td>
</tr>
<tr>
<td><strong>Median %Score</strong></td>
<td>90%</td>
<td>95%</td>
<td>85%</td>
<td>90%</td>
</tr>
<tr>
<td><strong>Mode %Score</strong></td>
<td>100%</td>
<td>100%</td>
<td>95%</td>
<td>100%</td>
</tr>
<tr>
<td><strong>Standard Dev</strong></td>
<td>24.41%</td>
<td>16.49%</td>
<td>22.67%</td>
<td>22.01%</td>
</tr>
<tr>
<td><strong>Std Est of Mean</strong></td>
<td>1.79%</td>
<td>1.29%</td>
<td>1.79%</td>
<td>0.98%</td>
</tr>
<tr>
<td><strong>Conf Interval</strong></td>
<td>3.53%</td>
<td>2.53%</td>
<td>3.51%</td>
<td>1.92%</td>
</tr>
</tbody>
</table>

**Summary Statistics**

For trials, there were three test instances available for each participant to take. As a percentage, the mode score was 100 in most tests (95 on the third test). The median scores for each instance were 90, 95 and 85 respectively. It is interesting then that the mean scores were 79, 87 and 76. These summary statistics show that participants did much better identifying known events with the tool than they would have if they had selected answers at random.

The difference of the mode and median scores as compared to the mean might be trivial in a study with a smaller sample size. However, there were over 500 test runs, and the Standard Error of the Mean (SEM) is less than two percent. Scores are not distributed in a standard manner. In fact, in all tests, only the middle volume of the curves at best, resemble a standard distribution. Perhaps the explanation is that participants either understood how to use the tool or they did not. Hence, there was little middle ground.
Test Score Distributions

The test findings are based on the distribution of the frequency of a test score. Each test score is a normalized number, the ratio of correct over incorrect items.

Figure 14 Comparisons of random data and test data distributions for all three tests
The total number of items in each test is always 20. Each answer is correct or incorrect, so there are 20 possible score values. The distribution is the number of occurrences of a test score over the number of tests taken overall. Put another way, it is a graduated measure of how prolific a test score is.

Each of the three tests had a different rule to qualify a selection. In each case, there was a different ratio of items which meet the rule and therefore a different probability of success or failure. Results show each test instances’ distribution of the frequency of each test score, and also a summary average of the distributions of all test scores.

![Figure 15 Comparisons of the average random and test data distributions overall](image)

**Distribution of the Foil**

To prove or disprove the hypothesis that the software tool is “useful”, we compare the average of all test score distributions against its foil, a random distribution. The random distribution is a Bernoulli trial of a binomial probability expansion output to 20 possible score bins.
In typical binomial expansion fashion, each bin is the number of combinations which result in the bin score \( \binom{N}{r} \) multiplied by the probability of the number of success for that score bin \( p^r \) multiplied by the probability for the number of failures for that score bin \( q^{(N-r)} \).

To be clear, because the events are independent, \( q \) the chance of failure, is simply \( 1 - p \).

Simply put, the random distribution is a graduated measure of how prolific a test scores would be if answers are selected completely at random.

\[
\binom{N}{r} \times p^r \times q^{(N-r)}
\]

Equation 2 Formula for calculating the binomial distribution of \( N \) Bernoulli trials

Given that \( N \) is always 20, \( r \) the correct number of items selected, \( p \) the probability of correctly selecting and \( q \) the probability of unsuccessfully selecting we compute the random distribution table found in Appendix D.

Evaluating the Hypothesis

To evaluate the hypothesis that the tool is useful, a null hypothesis is asserted. The null hypothesis \((H_0)\) is that the average foil score is actually the same as the average test score. The alternative hypothesis \((H_1)\) is that the average foil score is less than that the average user score. The test used to prove \( H_0 \) or affirm the directionality of \( H_1 \) is a single-tailed \( Z \)-test.

Where \( \bar{x} \) is the average test score, \( \mu \) is average random score, \( \sigma \) is the standard deviations from the mean, \( n \) is the population size, the following formula for \( Z \), the number of standard deviations from the mean of the foil, can be applied.

\[
Z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}} = \frac{81 - 38.33}{22.01 / \sqrt{507}} = 1.894
\]

Equation 3 the formula for computing the \( z \) score
The probability that the test score is the same as the average random distribution score is

\[
P(1 - Z)_{1-tailed} = \frac{P(1 - Z)}{2} = \frac{P(1 - .97)}{2} = 0.015 < \alpha = 0.05
\]

**Equation 4 Formula for computing the P-value**

As 0.015 is less than the threshold value of a Type-I error, that of rejecting a correct null hypothesis, the null hypothesis is safely determined to be false. Interestingly, this would also be true for any Z score over 1.28.

![Distributions of the Foil and Test Scores](image)

**Figure 16 Graph comparing the Z-Scores for rejecting H0 with the actual Z-Score**

The null hypothesis is rejected so the alternative hypothesis H1 that using the tool, the scores are better than random is accepted. As the mean of the scores is higher than the random score the conclusion is that the tool provides some utility and therefore is useful in
identifying known events.

Discussion:

![Triptych of “foot”, “belly” and “head” regions of the graphs](image)

By “rack of eye”, we observe that the distributions for each test seem to have three regions. For ease of discussion, consider that the leftmost region is the “foot”. The foot of the graph contains data that is below the mean of the random distribution. Consider also that the graph has a second region, a “belly”, wherein the test results align with the mean of the random distribution. Lastly, the “head” of the graph is the portion of the graph which supports the hypothesis; those scores are greater than the random distribution.

The first and second test runs exhibit no noticeable foot region in their graphs. Curiously, the first test shows a pooling in the belly region of the graph. Perhaps this would indicate that some users initially did not understand the instructions yet and so they exhibited random behavior.

Unfortunately, no formal narrative feedback was afforded in the design of the test to support this supposition. However, some informal feedback from a small number of participants offered that the user was looking for a known event rule for the trend between objects and not in the object instances themselves. As a result, the user clicked on very little and most of the objects transitioned off screen and so, was sorted by default. This would support a random trend behavior.
Notice in the second and third test runs, how the belly of the graph does two things. First, the graph begins an upward trend toward the head, perhaps suggesting better familiarization. Second, the belly region is cleaving, pinching off directly below the random mean. By and large the mass is moving toward the head. However, a new group starts scuttling off toward the developing foot region. An interpretation of this is perhaps that users are polarizing. While one sub-group is starting to “get it”, the group in the foot of the graph is functioning at worse than random behavior, they are misreading the visualization grossly.

Informal feedback provided by six tests participants was that instructions which used the words “few” and “many” were not specific enough. Perhaps, “few” rings and “many” rings became a source of worse than random scoring.

Considering the head region, two subregions can be observed. To tax the analogy further, entertain that the “face” is the area of data lying below a hypothetical standard distribution created from the mean of the test data. The “nose” then juts into the vertical region at the end of the graph. Recall that it is the nose region which contains both the mode (most frequent score of 100% correct!) and the median (above 90% correct on all tests). It is plausible that with more in depth instruction, this area may grow even more.
Chapter 6
Future Work & Conclusions

Initial prototypes of the data visualization tool leveraged non-web based techniques in the Processing API of Java. In the non-web environment, it was possible to do many more complicated depictions including ones that more clearly show the hierarchies within objects.

Figure 18 A prototype written by the author for visualizing hierarchies in XML. This example was generated from data about plants sold in a home-garden store.

Indeed, initial attempts wrestled with a compromise between three things: what could be done on the web, what could be done to show hierarchy within the object, and what could be done to make the best use of space. Ultimately, this work has been a compromise of these factors and others and many aspects deserve to be revisited in future works.
Better Depiction of Hierarchies

Though the tool in this study does truly visualize object hierarchy by uniquely identifying an object’s parent color, it essentially “flattens” the object to the point of the parent. The D3 API used in this study does offer a “Cluster Graph” which was not used in this tool for two reasons: complications making more than one instance of an object at a time, and complications getting the cluster to move away from a pre-initialized central point in the view window.

Figure 19 Example of a Cluster Graph by D3 creator, Mike Bostock

Given more resources, it might be worthwhile to investigate a multi-instance, animated, hybrid of a cluster graph and circle-packing diagram. The result would be hierarchical animation more space efficient than a circle pack perhaps more similar to a “Soap Bubble Diagram” where bubbles nest within bubbles. The focus of this further study would be to determine if hierarchy is useful in data visualization.
Trends Between Objects

A second related study might be to increase the animation rate analogous to the data stream throughput by using objects to graph the changes between data points rather than the data points themselves. Perhaps one object could represent changes over all the objects in the last unit of time, or perhaps the objects themselves could animate in a “breathing” sort of way that depicts a trend for the period they represent. In the latter example, animated objects would march along with other objects - each responsible for depicting their own time sub-period. The focus of this further study would be to determine how much throughput can usefully be depicted.

User Interaction

Lastly, and perhaps most obviously, feedback from users was resoundingly in favor of some kind of interaction with the graph. The ability to revisit the classification of an object after more data has been observed would likely be helpful. Another idea would be to allow the user to pick apart an object to look for something more specific, for example “what does six rings around a Panda really mean?” Object presentation customization, like suggestions for emoji labels instead of just using a random hash might be helpful. The focus of this further study would become much more human systems integration (HSI) related and observe the benefits of differing interfaces.

Conclusions

Foremost, the conclusion that a tool can be made to visualize streaming object data can be built. This is demonstrated in the presentation of the tool. Secondly, such a tool is useful in predicting known events. This is demonstrated statistically by disproving that using the tool is not significantly different than classifying known events at random. Softer arguments could also be made that the tool would improve with better subject familiarization and clearer representation of field hierarchies and values. These are the subjects of future works.
References


https://www.jasondavies.com/voronoi-treemap/.


J. Mutanguha, 2016, “From Heron’s formula to Descartes’ circle theorem.”

N. Kerracher1, J. Kennedy1 and K. Chalmers1, 2014 “The Design Space of Temporal Graph Visualisation.” Eurographics Conference on Visualization, p. 1–5 N. Elmqvist, M. Hlawitschka, and J. Kennedy (Editors) Edinburgh Napier University, United Kingdom.


Appendix A
Data Collected for Test 1

<table>
<thead>
<tr>
<th>TIME (GMT)</th>
<th>Score</th>
<th>TimeZone</th>
<th>UID</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 Sep, 00:52</td>
<td>20</td>
<td>EDT</td>
<td>3079398289</td>
</tr>
<tr>
<td>27 Sep, 17:23</td>
<td>20</td>
<td>EDT</td>
<td>1712135843</td>
</tr>
<tr>
<td>27 Sep, 23:58</td>
<td>20</td>
<td>EDT</td>
<td>3079398289</td>
</tr>
<tr>
<td>29 Sep, 16:19</td>
<td>17</td>
<td>EDT</td>
<td>2340715489</td>
</tr>
<tr>
<td>29 Sep, 16:19</td>
<td>20</td>
<td>PDT</td>
<td>2429806661</td>
</tr>
<tr>
<td>29 Sep, 16:22</td>
<td>18</td>
<td>EDT</td>
<td>1504394411</td>
</tr>
<tr>
<td>29 Sep, 16:22</td>
<td>20</td>
<td>EDT</td>
<td>1893097619</td>
</tr>
<tr>
<td>29 Sep, 16:22</td>
<td>21</td>
<td>EDT</td>
<td>1893097619</td>
</tr>
<tr>
<td>29 Sep, 16:23</td>
<td>14</td>
<td>EST</td>
<td>1829384513</td>
</tr>
<tr>
<td>29 Sep, 16:24</td>
<td>8</td>
<td>EDT</td>
<td>2318687957</td>
</tr>
<tr>
<td>29 Sep, 16:32</td>
<td>19</td>
<td>EDT</td>
<td>1060207719</td>
</tr>
<tr>
<td>29 Sep, 16:43</td>
<td>20</td>
<td>EDT</td>
<td>3600256895</td>
</tr>
<tr>
<td>29 Sep, 16:49</td>
<td>20</td>
<td>EDT</td>
<td>2690223618</td>
</tr>
<tr>
<td>29 Sep, 16:52</td>
<td>20</td>
<td>EDT</td>
<td>2690223618</td>
</tr>
<tr>
<td>29 Sep, 16:55</td>
<td>20</td>
<td>EDT</td>
<td>4287442072</td>
</tr>
<tr>
<td>29 Sep, 16:58</td>
<td>20</td>
<td>EDT</td>
<td>866887957</td>
</tr>
<tr>
<td>29 Sep, 16:59</td>
<td>20</td>
<td>CDT</td>
<td>582574536</td>
</tr>
<tr>
<td>29 Sep, 16:59</td>
<td>18</td>
<td>EST</td>
<td>1029277691</td>
</tr>
<tr>
<td>29 Sep, 17:00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29 Sep, 17:04</td>
<td>15</td>
<td>EDT</td>
<td>2690223618</td>
</tr>
<tr>
<td>29 Sep, 17:10</td>
<td>20</td>
<td>EST</td>
<td>2912851114</td>
</tr>
<tr>
<td>29 Sep, 17:11</td>
<td>20</td>
<td>EDT</td>
<td>3694918326</td>
</tr>
<tr>
<td>29 Sep, 17:54</td>
<td>16</td>
<td>EDT</td>
<td>2033481748</td>
</tr>
<tr>
<td>29 Sep, 17:59</td>
<td>20</td>
<td>CDT</td>
<td>2032997866</td>
</tr>
<tr>
<td>29 Sep, 20:15</td>
<td>16</td>
<td>EST</td>
<td>1842361090</td>
</tr>
<tr>
<td>29 Sep, 20:20</td>
<td>10</td>
<td>EDT</td>
<td>2859308761</td>
</tr>
<tr>
<td>29 Sep, 20:49</td>
<td>21</td>
<td>EDT</td>
<td>3312860073</td>
</tr>
<tr>
<td>29 Sep, 20:49</td>
<td>20</td>
<td>EDT</td>
<td>3312860073</td>
</tr>
<tr>
<td>30 Sep, 02:35</td>
<td>13</td>
<td>EDT</td>
<td>2051024117</td>
</tr>
<tr>
<td>30 Sep, 02:53</td>
<td>13</td>
<td>EDT</td>
<td>725442502</td>
</tr>
<tr>
<td>30 Sep, 04:57</td>
<td>14</td>
<td>EDT</td>
<td>727630581</td>
</tr>
<tr>
<td>30 Sep, 06:15</td>
<td>16</td>
<td>PDT</td>
<td>4215200070</td>
</tr>
<tr>
<td>30 Sep, 06:24</td>
<td>4</td>
<td>HST</td>
<td>3735854158</td>
</tr>
<tr>
<td>30 Sep, 09:55</td>
<td>20</td>
<td>EDT</td>
<td>1065172767</td>
</tr>
<tr>
<td>30 Sep, 10:59</td>
<td>20</td>
<td>EDT</td>
<td>3020795877</td>
</tr>
<tr>
<td>30 Sep, 12:44</td>
<td>18</td>
<td>EDT</td>
<td>2456849288</td>
</tr>
<tr>
<td>30 Sep, 12:55</td>
<td>18</td>
<td>EDT</td>
<td>3250046562</td>
</tr>
<tr>
<td>30 Sep, 13:09</td>
<td>18</td>
<td>EDT</td>
<td>3691860990</td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Zone</td>
<td>Time</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>02 Oct, 20:42</td>
<td>19 CEST</td>
<td>16 EDT</td>
<td>3694918326</td>
</tr>
<tr>
<td>02 Oct, 20:48</td>
<td>16 EDT</td>
<td>242980661</td>
<td>06 Oct, 15:06</td>
</tr>
<tr>
<td>02 Oct, 21:33</td>
<td>20 PDT</td>
<td>1616565473</td>
<td>06 Oct, 15:48</td>
</tr>
<tr>
<td>02 Oct, 21:33</td>
<td>20 CEST</td>
<td>3463823138</td>
<td>06 Oct, 15:48</td>
</tr>
<tr>
<td>02 Oct, 21:53</td>
<td>20 CDT</td>
<td>242980661</td>
<td>06 Oct, 18:37</td>
</tr>
<tr>
<td>02 Oct, 22:15</td>
<td>21 PDT</td>
<td>2352502674</td>
<td>06 Oct, 18:49</td>
</tr>
<tr>
<td>02 Oct, 22:29</td>
<td>5 PDT</td>
<td>3416149062</td>
<td>06 Oct, 18:56</td>
</tr>
<tr>
<td>02 Oct, 22:32</td>
<td>20 ACST</td>
<td>3463823138</td>
<td>06 Oct, 18:56</td>
</tr>
<tr>
<td>02 Oct, 22:32</td>
<td>21 ACST</td>
<td>2352502674</td>
<td>06 Oct, 20:29</td>
</tr>
<tr>
<td>02 Oct, 22:51</td>
<td>20 CDT</td>
<td>79910624</td>
<td>06 Oct, 21:25</td>
</tr>
<tr>
<td>02 Oct, 23:05</td>
<td>20 EDT</td>
<td>1020304390</td>
<td>07 Oct, 06:16</td>
</tr>
<tr>
<td>02 Oct, 23:34</td>
<td>12 CDT</td>
<td>1342817524</td>
<td>07 Oct, 18:42</td>
</tr>
<tr>
<td>03 Oct, 00:00</td>
<td>20 EDT</td>
<td>3694918326</td>
<td>08 Oct, 03:53</td>
</tr>
<tr>
<td>03 Oct, 01:28</td>
<td>16 EDT</td>
<td>3682696649</td>
<td>09 Oct, 18:56</td>
</tr>
<tr>
<td>03 Oct, 01:38</td>
<td>20 EDT</td>
<td>2690223618</td>
<td>10 Oct, 02:14</td>
</tr>
<tr>
<td>03 Oct, 11:54</td>
<td>5 EST</td>
<td>1003818262</td>
<td>11 Oct, 18:55</td>
</tr>
<tr>
<td>03 Oct, 12:34</td>
<td>15 MST</td>
<td>1090359293</td>
<td>13 Oct, 15:01</td>
</tr>
<tr>
<td>03 Oct, 13:02</td>
<td>19 AST</td>
<td>4242446255</td>
<td>13 Oct, 15:02</td>
</tr>
<tr>
<td>03 Oct, 13:38</td>
<td>20 EDT</td>
<td>2075897107</td>
<td>13 Oct, 15:09</td>
</tr>
<tr>
<td>03 Oct, 14:24</td>
<td>20 CDT</td>
<td>701710182</td>
<td>13 Oct, 15:09</td>
</tr>
<tr>
<td>03 Oct, 19:24</td>
<td>20 EDT</td>
<td>610367834</td>
<td>13 Oct, 16:17</td>
</tr>
<tr>
<td>04 Oct, 19:24</td>
<td>20 EDT</td>
<td>2690223618</td>
<td>13 Oct, 16:23</td>
</tr>
<tr>
<td>04 Oct, 19:24</td>
<td>21 EDT</td>
<td>2690223618</td>
<td>13 Oct, 16:53</td>
</tr>
<tr>
<td>05 Oct, 00:09</td>
<td>16 CST</td>
<td>1821703250</td>
<td>13 Oct, 16:53</td>
</tr>
<tr>
<td>05 Oct, 00:24</td>
<td>15 CST</td>
<td>1821703250</td>
<td>13 Oct, 17:09</td>
</tr>
<tr>
<td>05 Oct, 01:13</td>
<td>6 MDT</td>
<td>1471988582</td>
<td>13 Oct, 17:11</td>
</tr>
<tr>
<td>05 Oct, 01:37</td>
<td>14 CDT</td>
<td>4004643020</td>
<td>13 Oct, 17:18</td>
</tr>
<tr>
<td>05 Oct, 01:37</td>
<td>15 CDT</td>
<td>4004643020</td>
<td>13 Oct, 20:29</td>
</tr>
<tr>
<td>05 Oct, 01:47</td>
<td>13 CDT</td>
<td>3287301202</td>
<td>13 Oct, 20:34</td>
</tr>
<tr>
<td>05 Oct, 03:40</td>
<td>4 CDT</td>
<td>1684210396</td>
<td>13 Oct, 23:27</td>
</tr>
<tr>
<td>05 Oct, 08:08</td>
<td>16 EDT</td>
<td>3694918326</td>
<td>13 Oct, 23:52</td>
</tr>
<tr>
<td>05 Oct, 11:41</td>
<td>14 CDT</td>
<td>1174942141</td>
<td>14 Oct, 13:20</td>
</tr>
<tr>
<td>05 Oct, 12:26</td>
<td>14 CDT</td>
<td>833730990</td>
<td>15 Oct, 19:29</td>
</tr>
<tr>
<td>05 Oct, 13:22</td>
<td>17 CDT</td>
<td>78283959</td>
<td>16 Oct, 16:42</td>
</tr>
<tr>
<td>05 Oct, 13:52</td>
<td>6 CDT</td>
<td>2182935843</td>
<td>18 Oct, 20:32</td>
</tr>
<tr>
<td>05 Oct, 14:09</td>
<td>16 CDT</td>
<td>3001544070</td>
<td>20 Oct, 18:12</td>
</tr>
<tr>
<td>05 Oct, 14:40</td>
<td>4 CDT</td>
<td>4138248994</td>
<td>20 Oct, 18:20</td>
</tr>
<tr>
<td>05 Oct, 14:48</td>
<td>15 CDT</td>
<td>1684210396</td>
<td>20 Oct, 18:22</td>
</tr>
<tr>
<td>05 Oct, 19:32</td>
<td>19 CDT</td>
<td>1842410558</td>
<td>20 Oct, 18:28</td>
</tr>
<tr>
<td>05 Oct, 23:55</td>
<td>15 CDT</td>
<td>644847362</td>
<td>20 Oct, 22:36</td>
</tr>
<tr>
<td>06 Oct, 01:15</td>
<td>5 EDT</td>
<td>866887957</td>
<td>21 Oct, 11:32</td>
</tr>
<tr>
<td>06 Oct, 01:40</td>
<td>20 EDT</td>
<td>2690223618</td>
<td>23 Oct, 20:54</td>
</tr>
<tr>
<td>06 Oct, 12:30</td>
<td>4 EDT</td>
<td>3863720030</td>
<td>24 Oct, 18:38</td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Zone</td>
<td>Phone Number</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>27 Oct, 16:56</td>
<td>14 EDT</td>
<td>3694918326</td>
<td>13 Nov, 20:16</td>
</tr>
<tr>
<td>27 Oct, 19:00</td>
<td>20 CDT</td>
<td>1419745570</td>
<td>13 Nov, 20:16</td>
</tr>
<tr>
<td>27 Oct, 19:00</td>
<td>21 CDT</td>
<td>1419745570</td>
<td>13 Nov, 20:24</td>
</tr>
<tr>
<td>02 Nov, 18:56</td>
<td>20 HST</td>
<td>699353939</td>
<td>13 Nov, 20:29</td>
</tr>
<tr>
<td>03 Nov, 02:31</td>
<td>16 EDT</td>
<td>3944941889</td>
<td>13 Nov, 20:39</td>
</tr>
<tr>
<td>10 Nov, 19:04</td>
<td>10 EST</td>
<td>2554404583</td>
<td>13 Nov, 20:39</td>
</tr>
<tr>
<td>13 Nov, 20:14</td>
<td>17 EST</td>
<td>1419745570</td>
<td>13 Nov, 20:51</td>
</tr>
<tr>
<td>13 Nov, 20:15</td>
<td>19 EST</td>
<td>2814479625</td>
<td>14 Nov, 13:11</td>
</tr>
</tbody>
</table>
Appendix B
Data Collected for Test 2

<table>
<thead>
<tr>
<th>TIME (GMT)</th>
<th>Score</th>
<th>TimeZone</th>
<th>UID</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 Sep, 00:55</td>
<td>20</td>
<td>EDT</td>
<td>3079398289</td>
</tr>
<tr>
<td>27 Sep, 17:27</td>
<td>19</td>
<td>EDT</td>
<td>1712135843</td>
</tr>
<tr>
<td>27 Sep, 23:59</td>
<td>20</td>
<td>EDT</td>
<td>3079398289</td>
</tr>
<tr>
<td>29 Sep, 16:21</td>
<td>19</td>
<td>EDT</td>
<td>2340715489</td>
</tr>
<tr>
<td>29 Sep, 16:22</td>
<td>17</td>
<td>PDT</td>
<td>242980661</td>
</tr>
<tr>
<td>29 Sep, 16:25</td>
<td>18</td>
<td>EDT</td>
<td>1893097619</td>
</tr>
<tr>
<td>29 Sep, 16:25</td>
<td>19</td>
<td>EDT</td>
<td>1893097619</td>
</tr>
<tr>
<td>29 Sep, 16:25</td>
<td>20</td>
<td>EDT</td>
<td>1504394411</td>
</tr>
<tr>
<td>29 Sep, 16:25</td>
<td>19</td>
<td>EST</td>
<td>1829384513</td>
</tr>
<tr>
<td>29 Sep, 16:26</td>
<td>19</td>
<td>EDT</td>
<td>2318268989</td>
</tr>
<tr>
<td>29 Sep, 16:34</td>
<td>19</td>
<td>EDT</td>
<td>1060207719</td>
</tr>
<tr>
<td>29 Sep, 16:46</td>
<td>20</td>
<td>EDT</td>
<td>3600256895</td>
</tr>
<tr>
<td>29 Sep, 16:52</td>
<td>20</td>
<td>EDT</td>
<td>2690223618</td>
</tr>
<tr>
<td>29 Sep, 16:55</td>
<td>20</td>
<td>EDT</td>
<td>2690223618</td>
</tr>
<tr>
<td>29 Sep, 16:58</td>
<td>19</td>
<td>EDT</td>
<td>4287442072</td>
</tr>
<tr>
<td>29 Sep, 16:58</td>
<td>20</td>
<td>EDT</td>
<td>4287442072</td>
</tr>
<tr>
<td>29 Sep, 17:01</td>
<td>18</td>
<td>EDT</td>
<td>866887957</td>
</tr>
<tr>
<td>29 Sep, 17:01</td>
<td>17</td>
<td>CDT</td>
<td>582574536</td>
</tr>
<tr>
<td>29 Sep, 17:02</td>
<td>20</td>
<td>EST</td>
<td>1029277691</td>
</tr>
<tr>
<td>29 Sep, 17:06</td>
<td>20</td>
<td>EDT</td>
<td>2690223618</td>
</tr>
<tr>
<td>29 Sep, 17:09</td>
<td>20</td>
<td>EDT</td>
<td>2690223618</td>
</tr>
<tr>
<td>29 Sep, 17:12</td>
<td>20</td>
<td>EDT</td>
<td>2912851114</td>
</tr>
<tr>
<td>29 Sep, 17:13</td>
<td>20</td>
<td>EDT</td>
<td>3694918326</td>
</tr>
<tr>
<td>29 Sep, 17:57</td>
<td>20</td>
<td>EDT</td>
<td>2033481748</td>
</tr>
<tr>
<td>29 Sep, 18:01</td>
<td>19</td>
<td>CDT</td>
<td>2032997866</td>
</tr>
<tr>
<td>29 Sep, 20:17</td>
<td>16</td>
<td>EST</td>
<td>1842361090</td>
</tr>
<tr>
<td>29 Sep, 20:32</td>
<td>15</td>
<td>EDT</td>
<td>2859308761</td>
</tr>
<tr>
<td>29 Sep, 20:51</td>
<td>20</td>
<td>EDT</td>
<td>3312860073</td>
</tr>
<tr>
<td>30 Sep, 02:38</td>
<td>17</td>
<td>EDT</td>
<td>2051024117</td>
</tr>
<tr>
<td>30 Sep, 02:39</td>
<td>19</td>
<td>EDT</td>
<td>725442502</td>
</tr>
<tr>
<td>30 Sep, 04:59</td>
<td>16</td>
<td>HKT</td>
<td>727630581</td>
</tr>
<tr>
<td>30 Sep, 06:18</td>
<td>20</td>
<td>PDT</td>
<td>4215200070</td>
</tr>
<tr>
<td>30 Sep, 09:58</td>
<td>20</td>
<td>EDT</td>
<td>1065172767</td>
</tr>
<tr>
<td>30 Sep, 11:01</td>
<td>19</td>
<td>EDT</td>
<td>3020795877</td>
</tr>
<tr>
<td>30 Sep, 12:51</td>
<td>9</td>
<td>EDT</td>
<td>2456849288</td>
</tr>
<tr>
<td>30 Sep, 12:57</td>
<td>19</td>
<td>EAT</td>
<td>3250046562</td>
</tr>
<tr>
<td>30 Sep, 13:11</td>
<td>20</td>
<td>EDT</td>
<td>3691860990</td>
</tr>
<tr>
<td>30 Sep, 13:34</td>
<td>20</td>
<td>EDT</td>
<td>7799927776</td>
</tr>
</tbody>
</table>

30 Sep, 15:18  20 HST  660522047
30 Sep, 16:30  20 EDT  1221580124
30 Sep, 17:14  19 EDT  397218467
30 Sep, 17:41  17 PDT  4264092582
30 Sep, 17:54  17 PDT  361291739
30 Sep, 18:03  20 PDT  361291739
30 Sep, 18:06  20 MST  2718425632
30 Sep, 18:11  20 EDT  3833591170
30 Sep, 18:48  20 EDT  3079398289
30 Sep, 18:42  20 EST  2017840801
30 Sep, 21:03  20 EDT  3020795877
02 Oct, 09:29  17 IST  1023112317
02 Oct, 11:33  19 EDT  500986437
02 Oct, 12:15  20 EDT  516098537
02 Oct, 12:34  20 EDT  3682696649
02 Oct, 12:37  20 EDT  1208493539
02 Oct, 12:43  19 EDT  3694918326
02 Oct, 13:31  19 EDT  2690223618
02 Oct, 13:53  20 EST  2789866446
02 Oct, 14:53  18 CDT  4254683
02 Oct, 15:21  20 EDT  2827125153
02 Oct, 15:38  19 EDT  2690223618
02 Oct, 15:42  19 PDT  4133812499
02 Oct, 15:56  17 EDT  1060207719
02 Oct, 16:02  13 EDT  3865005430
02 Oct, 16:14  20 EDT  125619029
02 Oct, 16:54  20 EST  3218770930
02 Oct, 18:27  18 EDT  3682696649
02 Oct, 20:33  11 EDT  1302259463
02 Oct, 20:41  18 PDT  128232182
02 Oct, 20:45  11 CEST  1616565473
02 Oct, 20:51  20 EDT  3694918326
02 Oct, 21:55  20 CDT  3463823138
02 Oct, 22:17  20 PDT  242980661
02 Oct, 22:32  17 PDT  2352502674
02 Oct, 22:34  20 ACST 2236879688
02 Oct, 22:53  20 CDT  79910624
## Appendix C

Data Collected for Test 3

<table>
<thead>
<tr>
<th>TIME (GMT), Score, TimeZone, UID</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 Sep, 00:58</td>
<td>20</td>
</tr>
<tr>
<td>27 Sep, 17:30</td>
<td>15</td>
</tr>
<tr>
<td>28 Sep, 00:01</td>
<td>17</td>
</tr>
<tr>
<td>29 Sep, 15:46</td>
<td>20</td>
</tr>
<tr>
<td>29 Sep, 15:46</td>
<td>17</td>
</tr>
<tr>
<td>29 Sep, 15:46</td>
<td>15</td>
</tr>
<tr>
<td>29 Sep, 16:24</td>
<td>13</td>
</tr>
<tr>
<td>29 Sep, 16:27</td>
<td>19</td>
</tr>
<tr>
<td>29 Sep, 16:27</td>
<td>12</td>
</tr>
<tr>
<td>29 Sep, 16:28</td>
<td>20</td>
</tr>
<tr>
<td>29 Sep, 16:28</td>
<td>19</td>
</tr>
<tr>
<td>29 Sep, 16:48</td>
<td>20</td>
</tr>
<tr>
<td>29 Sep, 16:55</td>
<td>20</td>
</tr>
<tr>
<td>29 Sep, 16:59</td>
<td>19</td>
</tr>
<tr>
<td>29 Sep, 17:00</td>
<td>20</td>
</tr>
<tr>
<td>29 Sep, 17:04</td>
<td>18</td>
</tr>
<tr>
<td>29 Sep, 17:05</td>
<td>17</td>
</tr>
<tr>
<td>29 Sep, 17:09</td>
<td>17</td>
</tr>
<tr>
<td>29 Sep, 17:15</td>
<td>19</td>
</tr>
<tr>
<td>29 Sep, 17:15</td>
<td>19</td>
</tr>
<tr>
<td>29 Sep, 17:33</td>
<td>19</td>
</tr>
<tr>
<td>29 Sep, 17:59</td>
<td>18</td>
</tr>
<tr>
<td>29 Sep, 20:20</td>
<td>14</td>
</tr>
<tr>
<td>29 Sep, 20:54</td>
<td>19</td>
</tr>
<tr>
<td>29 Sep, 21:20</td>
<td>9</td>
</tr>
<tr>
<td>30 Sep, 02:40</td>
<td>13</td>
</tr>
<tr>
<td>30 Sep, 02:42</td>
<td>13</td>
</tr>
<tr>
<td>30 Sep, 05:01</td>
<td>9</td>
</tr>
<tr>
<td>30 Sep, 06:20</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 10:00</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 11:03</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 11:03</td>
<td>20</td>
</tr>
<tr>
<td>30 Sep, 13:00</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 13:13</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 13:37</td>
<td>16</td>
</tr>
<tr>
<td>30 Sep, 15:20</td>
<td>13</td>
</tr>
<tr>
<td>30 Sep, 16:32</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 17:30</td>
<td>15</td>
</tr>
<tr>
<td>30 Sep, 17:44</td>
<td>20</td>
</tr>
<tr>
<td>30 Sep, 17:56</td>
<td>13</td>
</tr>
<tr>
<td>30 Sep, 18:05</td>
<td>15</td>
</tr>
<tr>
<td>30 Sep, 18:09</td>
<td>16</td>
</tr>
<tr>
<td>30 Sep, 18:51</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 18:51</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 19:33</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 21:06</td>
<td>11</td>
</tr>
<tr>
<td>30 Sep, 09:31</td>
<td>20</td>
</tr>
<tr>
<td>30 Sep, 11:35</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 12:17</td>
<td>13</td>
</tr>
<tr>
<td>30 Sep, 12:36</td>
<td>17</td>
</tr>
<tr>
<td>30 Sep, 12:39</td>
<td>14</td>
</tr>
<tr>
<td>30 Sep, 14:46</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 13:4:</td>
<td>16</td>
</tr>
<tr>
<td>30 Sep, 13:55</td>
<td>14</td>
</tr>
<tr>
<td>30 Sep, 15:24</td>
<td>17</td>
</tr>
<tr>
<td>30 Sep, 15:42</td>
<td>20</td>
</tr>
<tr>
<td>30 Sep, 15:45</td>
<td>16</td>
</tr>
<tr>
<td>30 Sep, 15:58</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 15:58</td>
<td>17</td>
</tr>
<tr>
<td>30 Sep, 15:58</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 16:00</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 16:04</td>
<td>4</td>
</tr>
<tr>
<td>30 Sep, 16:04</td>
<td>4</td>
</tr>
<tr>
<td>30 Sep, 16:17</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 16:57</td>
<td>13</td>
</tr>
<tr>
<td>30 Sep, 18:30</td>
<td>19</td>
</tr>
<tr>
<td>30 Sep, 20:35</td>
<td>7</td>
</tr>
<tr>
<td>30 Sep, 20:44</td>
<td>15</td>
</tr>
<tr>
<td>30 Sep, 20:44</td>
<td>16</td>
</tr>
<tr>
<td>30 Sep, 20:50</td>
<td>14</td>
</tr>
<tr>
<td>30 Sep, 20:53</td>
<td>13</td>
</tr>
<tr>
<td>30 Sep, 21:33</td>
<td>20</td>
</tr>
<tr>
<td>30 Sep, 21:57</td>
<td>18</td>
</tr>
<tr>
<td>30 Sep, 22:20</td>
<td>20</td>
</tr>
</tbody>
</table>
### Appendix D

Values Used and Computed for Calculating the Random Distribution

<table>
<thead>
<tr>
<th>r</th>
<th>nC'</th>
<th>p_1</th>
<th>q_(n-r)_1</th>
<th>RndDist_1</th>
<th>p_2</th>
<th>q_(n-r)_2</th>
<th>RndDist_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1.00E+00</td>
<td>3.17E-03</td>
<td>3.17E-03</td>
<td>1.00E+00</td>
<td>1.81E-04</td>
<td>1.81E-04</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>2.50E-01</td>
<td>4.23E-03</td>
<td>2.11E-02</td>
<td>3.50E-01</td>
<td>2.79E-04</td>
<td>1.95E-03</td>
</tr>
<tr>
<td>2</td>
<td>190</td>
<td>6.25E-02</td>
<td>5.64E-03</td>
<td>6.69E-02</td>
<td>1.23E-01</td>
<td>4.29E-04</td>
<td>9.98E-03</td>
</tr>
<tr>
<td>3</td>
<td>1140</td>
<td>1.56E-02</td>
<td>7.52E-03</td>
<td>1.34E-01</td>
<td>4.29E-02</td>
<td>6.60E-04</td>
<td>3.23E-02</td>
</tr>
<tr>
<td>4</td>
<td>4845</td>
<td>3.91E-03</td>
<td>1.00E-02</td>
<td>1.90E-01</td>
<td>1.50E-02</td>
<td>1.02E-03</td>
<td>7.38E-02</td>
</tr>
<tr>
<td>5</td>
<td>15504</td>
<td>9.77E-04</td>
<td>1.34E-02</td>
<td>2.02E-01</td>
<td>5.25E-03</td>
<td>1.56E-03</td>
<td>1.27E-01</td>
</tr>
<tr>
<td>6</td>
<td>38760</td>
<td>2.44E-04</td>
<td>1.78E-02</td>
<td>1.69E-01</td>
<td>1.84E-03</td>
<td>2.40E-03</td>
<td>1.71E-01</td>
</tr>
<tr>
<td>7</td>
<td>77520</td>
<td>6.10E-05</td>
<td>2.38E-02</td>
<td>1.12E-01</td>
<td>6.43E-04</td>
<td>3.70E-03</td>
<td>1.84E-01</td>
</tr>
<tr>
<td>8</td>
<td>125970</td>
<td>1.53E-05</td>
<td>3.17E-02</td>
<td>6.09E-02</td>
<td>2.25E-04</td>
<td>5.69E-03</td>
<td>1.61E-01</td>
</tr>
<tr>
<td>9</td>
<td>167960</td>
<td>3.81E-06</td>
<td>4.22E-02</td>
<td>2.71E-02</td>
<td>7.88E-05</td>
<td>8.75E-03</td>
<td>1.16E-01</td>
</tr>
<tr>
<td>10</td>
<td>184756</td>
<td>9.54E-07</td>
<td>5.63E-02</td>
<td>9.92E-03</td>
<td>2.76E-05</td>
<td>1.35E-02</td>
<td>6.86E-02</td>
</tr>
<tr>
<td>11</td>
<td>167960</td>
<td>2.38E-07</td>
<td>7.51E-02</td>
<td>3.01E-03</td>
<td>9.65E-06</td>
<td>2.07E-02</td>
<td>3.36E-02</td>
</tr>
<tr>
<td>12</td>
<td>125970</td>
<td>5.96E-08</td>
<td>1.00E-01</td>
<td>7.52E-04</td>
<td>3.38E-06</td>
<td>3.19E-02</td>
<td>1.36E-02</td>
</tr>
<tr>
<td>13</td>
<td>77520</td>
<td>1.49E-08</td>
<td>1.33E-01</td>
<td>1.54E-04</td>
<td>1.18E-06</td>
<td>4.90E-02</td>
<td>4.49E-03</td>
</tr>
<tr>
<td>14</td>
<td>38760</td>
<td>3.73E-09</td>
<td>1.78E-01</td>
<td>2.57E-05</td>
<td>4.14E-07</td>
<td>7.54E-02</td>
<td>1.21E-03</td>
</tr>
<tr>
<td>15</td>
<td>15504</td>
<td>9.31E-10</td>
<td>2.37E-01</td>
<td>3.43E-06</td>
<td>1.45E-07</td>
<td>1.16E-01</td>
<td>2.61E-04</td>
</tr>
<tr>
<td>16</td>
<td>4845</td>
<td>2.33E-10</td>
<td>3.16E-01</td>
<td>3.57E-07</td>
<td>5.07E-08</td>
<td>1.79E-01</td>
<td>4.39E-05</td>
</tr>
<tr>
<td>17</td>
<td>1140</td>
<td>5.82E-11</td>
<td>4.22E-01</td>
<td>2.80E-08</td>
<td>1.77E-08</td>
<td>2.75E-01</td>
<td>5.56E-06</td>
</tr>
<tr>
<td>18</td>
<td>190</td>
<td>1.46E-11</td>
<td>5.63E-01</td>
<td>1.56E-09</td>
<td>6.21E-09</td>
<td>4.23E-01</td>
<td>4.99E-07</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>3.64E-12</td>
<td>7.50E-01</td>
<td>5.46E-11</td>
<td>2.17E-09</td>
<td>6.50E-01</td>
<td>2.83E-08</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>9.09E-13</td>
<td>1.00E+00</td>
<td>9.09E-13</td>
<td>7.61E-10</td>
<td>1.00E+00</td>
<td>7.61E-10</td>
</tr>
<tr>
<td>r</td>
<td>NC^r</td>
<td>p^3</td>
<td>q[n_r]</td>
<td>RndDist_3</td>
<td>p^Av</td>
<td>q[n_r]-Av</td>
<td>RndDist_Av</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1.00E+00</td>
<td>1.16E-07</td>
<td>1.16E-07</td>
<td>1.00E+00</td>
<td>6.32E-05</td>
<td>6.32E-05</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>5.50E-01</td>
<td>2.58E-07</td>
<td>2.83E-06</td>
<td>3.83E-01</td>
<td>1.03E-04</td>
<td>7.86E-04</td>
</tr>
<tr>
<td>2</td>
<td>190</td>
<td>3.03E-01</td>
<td>5.73E-07</td>
<td>3.29E-05</td>
<td>1.47E-01</td>
<td>1.66E-04</td>
<td>4.64E-03</td>
</tr>
<tr>
<td>3</td>
<td>1140</td>
<td>1.66E-01</td>
<td>1.27E-06</td>
<td>2.41E-04</td>
<td>5.63E-02</td>
<td>2.70E-04</td>
<td>1.73E-02</td>
</tr>
<tr>
<td>4</td>
<td>4845</td>
<td>9.15E-02</td>
<td>2.83E-06</td>
<td>1.25E-03</td>
<td>2.16E-02</td>
<td>4.37E-04</td>
<td>4.58E-02</td>
</tr>
<tr>
<td>5</td>
<td>15504</td>
<td>5.03E-02</td>
<td>6.28E-06</td>
<td>4.90E-03</td>
<td>8.28E-03</td>
<td>7.09E-04</td>
<td>9.10E-02</td>
</tr>
<tr>
<td>6</td>
<td>38760</td>
<td>2.77E-02</td>
<td>1.40E-05</td>
<td>1.50E-02</td>
<td>3.17E-03</td>
<td>1.15E-03</td>
<td>1.41E-01</td>
</tr>
<tr>
<td>7</td>
<td>77520</td>
<td>1.52E-02</td>
<td>3.10E-05</td>
<td>3.66E-02</td>
<td>1.22E-03</td>
<td>1.86E-03</td>
<td>1.76E-01</td>
</tr>
<tr>
<td>8</td>
<td>125970</td>
<td>8.37E-03</td>
<td>6.90E-05</td>
<td>7.27E-02</td>
<td>4.66E-04</td>
<td>3.02E-03</td>
<td>1.78E-01</td>
</tr>
<tr>
<td>9</td>
<td>167960</td>
<td>4.61E-03</td>
<td>1.53E-04</td>
<td>1.19E-01</td>
<td>1.79E-04</td>
<td>4.90E-03</td>
<td>1.47E-01</td>
</tr>
<tr>
<td>10</td>
<td>184756</td>
<td>2.53E-03</td>
<td>3.41E-04</td>
<td>1.59E-01</td>
<td>6.85E-05</td>
<td>7.95E-03</td>
<td>1.01E-01</td>
</tr>
<tr>
<td>11</td>
<td>167960</td>
<td>1.39E-03</td>
<td>7.57E-04</td>
<td>1.77E-01</td>
<td>2.63E-05</td>
<td>1.29E-02</td>
<td>5.69E-02</td>
</tr>
<tr>
<td>12</td>
<td>125970</td>
<td>7.66E-04</td>
<td>1.68E-03</td>
<td>1.62E-01</td>
<td>1.01E-05</td>
<td>2.09E-02</td>
<td>2.65E-02</td>
</tr>
<tr>
<td>13</td>
<td>77520</td>
<td>4.21E-04</td>
<td>3.74E-03</td>
<td>1.22E-01</td>
<td>3.86E-06</td>
<td>3.39E-02</td>
<td>1.01E-02</td>
</tr>
<tr>
<td>14</td>
<td>38760</td>
<td>2.32E-04</td>
<td>8.30E-03</td>
<td>7.46E-02</td>
<td>1.48E-06</td>
<td>5.50E-02</td>
<td>3.15E-03</td>
</tr>
<tr>
<td>15</td>
<td>15504</td>
<td>1.27E-04</td>
<td>1.85E-02</td>
<td>3.65E-02</td>
<td>5.67E-07</td>
<td>8.92E-02</td>
<td>7.84E-04</td>
</tr>
<tr>
<td>16</td>
<td>4845</td>
<td>7.01E-05</td>
<td>4.10E-02</td>
<td>1.39E-02</td>
<td>2.17E-07</td>
<td>1.45E-01</td>
<td>1.52E-04</td>
</tr>
<tr>
<td>17</td>
<td>1140</td>
<td>3.86E-05</td>
<td>9.11E-02</td>
<td>4.01E-03</td>
<td>8.33E-08</td>
<td>2.35E-01</td>
<td>2.23E-05</td>
</tr>
<tr>
<td>18</td>
<td>190</td>
<td>2.12E-05</td>
<td>2.03E-01</td>
<td>8.16E-04</td>
<td>3.19E-08</td>
<td>3.80E-01</td>
<td>2.31E-06</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>1.17E-05</td>
<td>4.50E-01</td>
<td>1.05E-04</td>
<td>1.22E-08</td>
<td>6.17E-01</td>
<td>1.51E-07</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>6.42E-06</td>
<td>1.00E+00</td>
<td>6.42E-06</td>
<td>4.69E-09</td>
<td>1.00E+00</td>
<td>4.69E-09</td>
</tr>
</tbody>
</table>
<!doctype html>
<head>
<title></title>
<style>
body { margin: 0;
    font-family: "Helvetica Neue Light", "Helvetica Neue Light",
    "Helvetica Neue", "Roboto Light", "Segoe UI Web Light", "Segoe UI Light", "Segoe UI
    Web Regular", "Segoe UI", Helvetica, Arial, sans-serif;
    margin: 1em;
    background: #293950;
    color: #ecf0f0;
}

text { font-weight: normal;
    font-size: 2.4em;
    text-align: center;
}

misses {
    color: red;
}
hits { 
    color: green;
}

.fill { 
    fill: "none";
    stroke: green;
    stroke-width: 5;
    stroke-opacity: 1;
}
fill-opacity: 0;
letter-spacing: 1em;
}
</style>
</head>
<body>
<p id="score" class="score">Training... </p>
<script src=https://cdn.pubnub.com/sdk/javascript/pubnub.4.0.11.min.js></script>
<script src="pubnubkeys.js"></script>
<script src="fingerprint.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.4.11/d3.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.2/rollups/aes.js"></script>
<script src="d3moji.js"></script>
<script>
var width = window.innerWidth,
    height = window.innerHeight,
    vOffset = 0,
    setupTime = 250,
    layoutSize = (Math.min(width, height)) / 3,
    minRadius = layoutSize / 6,
    maxRadius = minRadius * 6,
    emojiSize = minRadius * .75,
    transitTime = 30000;

var topic = "thesis";
pubkey = ";

var hits = 0;
var misses = 0;

var evalFunction = function () {
    return false;
};

// D3 Bubble Chart

var art = d3.select('body').append('svg')
    .attr('width', width)
    .attr('height', height)
    .append('g');
art.append('rect')
    .attr('class', 'overlay')
    .attr('width', width)
    .attr('height', height);

var chart = d3.select('body').append('svg')
    .attr('width', width)
    .attr('height', height)
    .append('g');

chart.append('rect')
    .attr('class', 'overlay')
    .attr('width', width)
    .attr('height', height);

var g = art.append("g");

function drawCluster(m) {
    if (testLen < 0)
        return;
    var object = JSON.parse(m);
    var jsonData = processData(m);
    layoutSize = 10;
    jsonData.children.forEach(function (o) {
        if ((o.size * 2) > layoutSize) {
            // roughly the radius of the cluster
            layoutSize += o.size;
        }
    });
    if (layoutSize < height / 2)
        layoutSize = height / 2;
    setVOffset();
    var cluster = d3.layout.pack()
        .size([layoutSize, layoutSize])
        .value(function (d) {
            return d.size;
        }) // new data is loaded to bubble layout
        .radius(this.value)
        .padding(3);
    if ((inTraining) && (evalFunction(object))) {
        jsonData.name = "training_circle";
        jsonData.className = "train";
jsonData.contents = "click me!";
jsonData.uid = Math.floor((1 + Math.random()) * 0x10000);

if (jsonData.children.length > 0) {
    jsonData.children[0].sentinel = true;
}

var nodes = cluster.nodes(jsonData).filter(function (d) {
    if (!d.children || d.name === "training_circle") {
        return true;
    }
});

//DATA JOIN: assign new data to existing DOM
var vis = chart.selectAll("g")
    .data(nodes, function (d) {
        return d.uid;
    });

//UPDATE: //there is no UPDATE in this design

//ENTER + UPDATE
vis.enter()
    .append("g").attr('transform', function (d) {
        return 'translate(' + (d.x - width / 10) + ',' +
        (d.y + vOffset) + ')';
    }).attr('class', function (d) {
        return d.className;
    }).transition("enter").duration(transitTime / 2).attr('transform',
    function (d) {
        return 'translate(' + (d.x + (width * .9)) + ',' +
        (d.y + vOffset) + ')';
    }).ease('linear').each("end", function (d) {
        if (d.sentinel === true) {
            console.log("d.sentinel is true ")
            processClick(!evalFunction(object), true);
        }
    }).remove();

// enter - only applies to incoming elements (once emptying data)
vis.append('circle').attr('r', function (d) {
    if (d.name === "training_circle") {
        d.r += 75;
    }
});
return d.r; //the radius circle pack thinks this should be
}).attr('stroke-opacity', function (d) {
  if (d.name === "training_circle") {
    denominator = Math.max(1, hits);
    return 1 / denominator;
  } else
    return 1;
}).attr("id", function (d) {
  return "circle-" + d.uid;
});

vis.append('emoji').filter(function (d) {
  if (!d.children) {
    return true;
  }
}).attr('symbol', function (d) {
  return emojiHash(d.name);
}).attr('width', function (d) {
  return emojiSize; //d.r * 1.5;
}).attr('height', function (d) {
  return emojiSize; //d.r * 1.5;
}).attr('opacity', 255)
  .attr("transform", function (d) {
    return "translate(" + (-.5 * emojiSize) + "," + (-.5 * emojiSize) + ")";
  });

vis.filter(function (d) {
  return isNaN(d.contents);
}).each(function (d) {
  console.log("adding content svg text for field named: " + d.name);
  this_uid = d.uid;

  var svgContainer = chart.selectAll("g").filter(function (d) {
    return d.uid === this_uid;
  }).append('svg').filter(function (d) {
    return isNaN(d.content);
  })
    .attr('width', function (d) {
      return 2 * d.r;
    })
    .attr('height', function (d) {
      return 2 * d.r;
    })
    .attr('stroke', function (d) {
      return "#fff";
    })
    .attr('fill', function (d) {
      return "#000";
    });

  svgContainer.append('text').filter(function (d) {
    return !isNaN(d.content);
  }).text(d.content);
vis.filter(function (d) {
    return !isNaN(d.contents);
}).each(function (d) {
    //draw rings for number size
    console.log("###d.r is :" + d.r + " for field named: " + d.name);
    this_uid = d.uid;
    var numRings = Math.min(d.contents, 210);
    for (var i = 0; i < numRings; i++) {
        if (i % 8 === 0) {
            chart.selectAll("g").filter(function (d) {
                return d.uid === this_uid;
            }).append('circle')
                .attr('r', function (d) {
                    return ((i * (d.r - emojiSize) / numRings) + (emojiSize)); //the radius circle pack thinks this should be
                })(emojiSize)); //the radius circle pack thinks this should be
                .style("stroke-width", 1) // set the stroke
                .style("fill-opacity", 0)
                .style("stroke-opacity", 1);
            }
        }
    }
}

//this is the clickable region, transparent and bigger than the object
vis.append("circle")
    .attr('r', function (d) {
        return d.r + 60; // bigger than radius of visible objects
    })(emojiSize))
    .style("fill", "white")
    .style("opacity", 0)
    .on('click', function (d) {
        processClick(evalFunction(object), false);
        chart.selectAll("g")
            .data(nodes, function (d) {
                return d.uid;
                //MUST make transition w/ same name to CANX it
            }).transition("remove").remove();
    });
function processClick(hit, onRemove) {
  console.log("click!");
  hOffset = (onRemove) ? width : width * .6;
  hOffset -= height / 2;

  if (hit === true) {
    g.append("svg:image")
      .attr("xlink:href", "Green_check.svg")
      .attr("id", "feedback-1")
      .attr("width", height / 2)
      .attr("height", height / 2)
      .attr("x", hOffset)
      .attr("y", height / 4);
    hits++;
    trainingLen--;
    if (inTraining) {
      if (trainingLen < 0) {
        inTraining = false;
        hits = 0;
        misses = 0;
        chart.selectAll("g").transition("remove").remove();
        document.getElementById('score').innerHTML = "Click on groups that match the pattern from training";
        console.log("Training is over");
      }
    } else {
      g.append("svg:image")
        .attr("xlink:href", "Red_x.svg")
        .attr("id", "feedback-1")
        .attr("width", height / 2)
        .attr("height", height / 2)
        .attr("x", hOffset)
        .attr("y", height / 4);
      misses++;
    }
  }
  setTimeout(function () {
    d3.select("#feedback-1").remove();
  }, 100);
  if (inTraining === false) {
    testLen--;
    document.getElementById('score').innerHTML = "Score: " +
"<m style='color:red;'>" + misses + ": </m>" +
"<h style='color:green;'>" + hits + "</h>";

if (testLen < 0) {
    //test is over
    publishScore();
    console.log("Test is over");
    clearTimeout(paceFunction);
    test_data = null;
    window.location.href =
    getNextPage();
    //"https://qcarver.github.io";
}
}
}

function processData(data) {
    if (!data)
        return;
    var dataSet = [];
    var obj = JSON.parse(data);
    traverse(obj, dataSet, "root");
    return {children: dataSet};
}

function traverse(o, dataSet, parentName) {
    for (var i in o) {
        if (o[i] !== null) {
            //skip over container objects
            if (typeof (o[i]) === "object") {
                //going one step down in the object tree!!
                traverse(o[i], dataSet, i);
            } else {
                //sometimes the data file has a number in quotes
                if (!isNaN(Number(o[i]))) {
                    o[i] = Number(o[i])
                }
                //represent false booleans simply by not showing them
                if (includeFieldWithValue(o[i]) === true) {
                    //only push on fields (not objects)
                    var size = circleSize(o[i]);
                    var uid = Math.floor((1 + Math.random()) * 0x10000);
// dataSet.push({name: i, className: "root", size: size

dataSet.push({name: i, className: 

getClassName(parentName), size: size,

, uid: (uid), vOffset: vOffset, clicked: false,

contents: o[i]);

    console.log("uid: " + uid + " + parentName + "." + i + " : " + o[i] + " = " + size);

    }
}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}
function getTrainingClass() {
    var name = "root";
    if (document.getElementsByClassName(name).length === 0) {
        var style = document.createElement('style');
        style.type = 'text/css';
        style.innerHTML = '.' + name + ' { fill: none; color: white; stroke: green; }';
        document.getElementsByTagName('head')[0].appendChild(style);
    }
    return name;
}

function getHashColor(name) {
    var hash = 0, i, chr;
    if (name.length === 0)
        return hash;
    for (i = 0; i < name.length; i++) {
        chr = name.charCodeAt(i);
        //try and distribute the char a-z or A-Z to a # between 0&255
        chr = (chr < 97) ? chr - 64 : chr - 70;
        chr += (52 * (chr % 5)) + i % 5;
        //shift determines which of the 3 bytes we want to update
        var shift = (name.length > 2) ? (i % 3) : (chr % 3);
        //update the byte by adding to it
        hash += ((chr | (hash & 0xFF)) << 8 * shift);
    }
    //we used addition so.. number might exceed 3 bytes now, mask it
    hash &= 0xFFFFFF;
    //this gets our number to a six character hex representation of hash
    var pad = "000000";
    return pad.substring(0, pad.length - hash.toString(16).length) + hash.toString(16);
}

function circleSize(value) {
    if ((value.toString().toUpperCase() === "TRUE") ||
        (value.toString().toUpperCase() === "T")) {
        value = 1;
    }
    var d = (isNaN(value) ? value.length : value);
    f = d3.scale.log().base(2).domain([Math.exp(0),
Math.exp(7.5)).range([minRadius, maxRadius])
    return f(d);
}

///////////////////////////other utilities
function setVOffset() {
    // Rnd vertical offset for new group that differs from last vOffset
    vOffset = (Math.random() * (height - layoutSize) + vOffset) % (height - layoutSize);
}

// hash any given string to an emoji name.. gets the icon from eg: https://twemoji.maxcdn.com/svg/1f4af.svg
function emojiHash(hashMe) {
    return Object.keys(emojiMap)[Math.abs(hashMe.hashCode()) % Object.keys(emojiMap).length];
}

// add a function to the native String type to emulate java's hashCode()
String.prototype.hashCode = function () {
    var hash = 0;
    if (this.length === 0)
        return hash;
    for (var i = 0; i < this.length; i++) {
        var character = this.charCodeAt(i);
        hash = ((hash << 5) - hash) + character;
        hash = hash & hash; // Convert to 32bit integer
    }
    return hash;
};

function publishScore() {
    var fingerPrint = new Fingerprint().get();
    var date = new Date();
    var scoreString = "Test: " + currTest + ", " + hits + ":" + misses + ", " +
        date.getMonth() + "/" + date.getDay() + "/" +
    date.getFullYear() + ", " +
    date.getHours() + ":" + date.getMinutes() + "." +
    date.getSeconds() +
        " TZoffset: " + date.getTimezoneOffset() / 60 + ", uid: " +
    fingerPrint;
if (pubnub !== "empty") {
    console.log("Publishing: " + scoreString);
    var publishConfig = {
        channel: "thesis",
        message: scoreString
    };
    pubnub.publish(publishConfig, function (status, response) {
        console.log(status, response);
    });
}

// Pass in hex number as 6 char string
function getInvertedColor(hex) {
    // Invert color components
    var r = (255 - parseInt(hex.slice(0, 2), 16)).toString(16),
        g = (255 - parseInt(hex.slice(2, 4), 16)).toString(16),
        b = (255 - parseInt(hex.slice(4, 6), 16)).toString(16);
    // Pad each with zeros and return
    return padZero(r) + padZero(g) + padZero(b);
}

function padZero(str, len) {
    len = len || 2;
    var zeros = new Array(len).join('0');
    return (zeros + str).slice(-len);
}

function getDataFile() {
    if (currTest === "Test_1_of_3")
        return "MOCK_PPV_DATA.json";
    if (currTest === "Test_2_of_3")
        return "CRASH_DATA.json";
    if (currTest === "Test_3_of_3")
        return "TITANIC_MORT_DATA.json";
    return "MOCK_PPV_DATA.json";
}

function getNextTest() {
    if (currTest === "Test_1_of_3")
        return "/CrashIntro.html?";
    if (currTest === "Test_2_of_3")
        return "/TitanicIntro.html?";
function getNextPage() {
    var b1 = location.href.lastIndexOf("/");
    var baseUrl = location.href.substring(0, b1);
    var nextTestUrl = baseUrl + getNextTest() + pdub;
    console.log("next page will be: " + nextTestUrl);
    return nextTestUrl;
}

//main:
var pubnub = "empty";
var datafile = "CRASH_DATA.json";
var paceFunction;
var BreakException = {};
var trainingLen = 5;
var testLen = 20;
var inTraining = true;
var pdub = "empty";
var currTest = "empty";
var request = new XMLHttpRequest();
var p1 = location.href.indexOf("?");
var p2 = location.href.indexOf("-");
if ((p1 !== -1) && (p2 !== -1)) {
    //make the key w/ whats between ? and -
    pdub = location.href.substring(p1 + 1, p2);
    console.log("This was passed in: " + pdub);
    var encryptedPublishKey = "U2FsdGVkX1/g6q5ZZ02ZGnWFmOm30ocglx0sUMZs44CfHUcGHXZdxkyqLGGdX9U750JRjWtp7y3zC+gRbde
rBDQ==";
    var encryptedSubscribeKey = "U2FsdGVkX1/vbZ2215lR64jaxTtryrVL5aWIrZ4v/DmQFMcu8tB8dRHiOLxwNOSRxgXkAIU1IbhTuw/vyF0
3LHQ==";
    var decryptedPublishKey = CryptoJS.AES.decrypt(encryptedPublishKey.toString(), pdub);
    var decryptedSubscribeKey = CryptoJS.AES.decrypt(encryptedSubscribeKey.toString(), pdub);
    var pu
bnu
b = new PubNub({
    publishKey: decryptedPublishKey.toString(CryptoJS.enc.Utf8),
    subscribeKey: decryptedSubscribeKey.toString(CryptoJS.enc.Utf8)
});
    currTest = location.href.substring(p2 + 1)
datafile = getDataFile();
} else if ((p1 !== -1) && (p2 === -1)) {
    // no publishing but get test data set name
    currTest = location.href.substr(p1 + 1);
    datafile = getDataFile();
}
request.open("GET", datafile, false);
request.send(null);
var JSON_object = JSON.parse(request.responseText);
evalFunction = new Function("obj", JSON_object.eval_function);
var i = 0;
var test_data = [];
test_data = JSON_object.test_data;
test_data.forEach(function (td, index) {
    setTimeout(function () {
        objAsString = JSON.stringify(td);
        drawCluster(objAsString);
        console.log("sending " + objAsString + " in " + index * (2000) + " ms");
    },
        index * transitTime / 8);
});