CS8803-MDS: Evaluation Plan

Eric W. Martin
emartin31@gatech.edu

Akshay Iyer
aiyer89@gatech.edu

In addition to submitting your Evaluation section, also re-include your submitted Introduction (Assignment 1 project proposal) section in this Overleaf. You do not need to make any edits to the introduction section at this time (though you can if you want). Also include the correct references in the ref.bib file.

1 INTRODUCTION

Scatter plots are an indispensible, visual tool for analysts and scientists. Specifically, it is adept at displaying correlations between ostensibly independent variables [5]. This property of scatter plots should not be understated since it triggers pertinent, insightful questions guiding the discovery process such as 'Why attributes X are Y correlated?', 'Is there are third unobserved variable explaining the correlation?',' and 'Is the relationship causal?'. Given this property, scatter-plots have duly received much research attention over the decades related to how best to display them [3, 6–8, 10]. In recent years, there has been an explosion in data due to the dropping cost of compute, but the sizes of screens have understandably remained constrained by the limits of human perception. This recent trend places scatter-plots at a disadvantage since plotting millions of data points in a finite area leads to unintelligible data blobs [8]. One may be predisposed to believe that the scatter plots have served their usage and are no longer relevant for exploring large datasets, but more data is not necessarily better. The judgments scatter-plots enable are every bit as valuable as the underlying data they represent and sampling methods should be applied to extend these insights into the era of Big Data.

Ever since Stonebreaker et. al. in a famous 2007 paper advocating for the development of specialized databases in lieu of one-size-fits-all approaches [14], there has been embracement of specialized data engines for handling the unique and large data needs in environments such as data warehouses, streaming, and text search. The improvements in backend processing initiated by this cognitive shift, however, have not been fully realized in visualization software needed by analysts and data scientists. There is a lack of interactive visual tools for these users [1, 2, 9, 16] to explore these potentially rich datasets, and this paucity of tools is glaring considering the exponential growth of data and the leveling off of microprocessor performance. This lack of appropriate UIs can be partly attributed to the obvious fact that while data sets have grown, screens have not. Recent work in the field of human-in-the-loop analytics (HILDA) have begun to address this problem by providing users with graphical statistical summaries to help navigate large spreadsheets [13] and specialized sketches which communicate screen limitations to the backend servers [4] to limit query sizes. These recent approaches, however, presume that summary statistics are the only displays worth presenting to the users. There is a bias to provide a complete, descriptive view of a large dataset which discounts the potential errors such generic views impugn. This bias has led current approaches to eschew visualizing scatter plots for large datasets despite both their intuitive and informative value.

In this paper, we introduce QuickScatter, a new prototype scatterplot system capable of generating multiple sample scatter plots which display correlations between variables in large, multi-dimensional datasets. QuickScatter is a simple system which uses established statistical sampling methods such as uniform sampling to produce sparse but informative scatter plots. As mentioned earlier, recent research avoids scatter plots because plotting millions or even thousands of data points tends to generate unintelligible data blobs. This unintelligibility, however, is an artifact due to too much data at the same time within a finite window. Valuable insights and correlations remain present in the data regardless of a screen's ability to render it. These patterns can be gleaned via well-established statistical sampling methods which not only identify correlations but do so quickly to enable interactive, exploratory analysis. Most importantly, a sparse data representation allows users to work with concrete and intuitive data instances and by displaying multiple sampled views, users can gain an intuitive sense for the level of variability and noise in the data at a fine grain level which may warrant a full statistical analysis later on.

The main challenges to producing multiple scatter plots are 1) selecting sample the data from a potentially unknown distribution, 2) identifying how many sparse scatter plots are needed to illustrate both the correlations and the variability across samples, and finally 3) producing plots and/or updates to plots within 1 second to support interactive click-based interfaces [11]. To confirm the usability of the interface, it was decided to focus on meeting the strict latency requirements first. To that end, a simplified, normally distributed data environment was assumed so that online uniform sampling methods could be used. Furthermore, as an ad-hoc solution, a heuristic of 4 scatter-plots requiring 4 unique sample sets were selected to enable users to observe persistent trends or noise across samples.

Initially, we will look at more rudimentary sampling techniques such as random sampling and will render summary scatter plots on smaller datasets. These initial datasets will likely be on the order of 10000 or so. How well we can observe correlations at this smaller level will give us an indication of whether our solution is scalable or not. Presence of blobs and lack of any patterns may signal the need for different sampling techniques and/or different datasets. At this lower level we will also look at statistics like Chi-squared to ensure that the scatter plot is indeed a good representation of our data. From there, we will apply our scatter plot functions on a larger dataset on the order of 10000000 or higher. After determining optimal sampling techniques and accuracy metrics, we will direct our focus to more user-relevant feedback. This might include visualizations such as confidence intervals and the ability to visualize correlations for specific subsets of the data. One dataset that we found for a smaller-scale scatter-plot can be accessed here: https://www.kaggle.com/datasets/muhammadadiltalay/imdb-video-games. This dataset contains information about the plot, genres, and ratings for various video games. It should provide us with

the ability to understand any correlations between genre and ratings across different years. Using older census data may also help us to truly determine our scatterplot's effectiveness on big data by determining relationships between income and education, race, and/or occupation. One dataset we found that meets the 10 Million row requirement is here: https://www.kaggle.com/datasets/brijeshbmehta/adult-datasets?select=adult10m.

If our scatterplot's results are both accurate and responsive, we can be more confident that sampling based approaches lead to an effective tool for visualization. Especially for big-data systems, removing clutter and helping analysts to visualize patterns is a valuable tool to direct further investigation. Such analysis is particularly useful when trying to decide the scope of machine learning in a project. Overall, if our experimentation is successful, analysts will have an intuitive way to understand patterns in the dataset with varying confidence levels without having to be concerned with low latencies and system crashes.

2 EVALUATION PLAN

2.1 Thesis

The main thesis underlying QuickScatter is that online sampling where the sample size is bounded by the dimensions of the visualization produces scatter plot images which are sufficiently similar to the ground truth scatter plot image produced by the full dataset.

2.2 Claim

The visualization dependent sampling sizes recommended by QuickScatter provide a more usable framework for enabling data scientists and analysts to generate scatter plots for interactive, exploratory analysis. Unlike offline approaches which build idealized samples prior to exploration, QuickScatter does not limit the explorable set of attributes to compare. Furthermore, the QuickScatter sample sizes can be used in a variety of environments such as JupyterNotebooks and for popular graphics libraries such Plotly and Vega. The central claim of QuickScatter is that for many datasets projected to a finite 2D scatter plot, there exists a sampling point K which yields a scatter plot image denoted as Image_K which is similar enough to the corresponding full dataset image denoted as Image_N such that further sampling is unnecessary. This sampling point K represents the max information bandwidth of a fixed scatter plot and given dataset.

2.3 Evaluation Design

Dependent Variable.

There are two main classes of dependent variables in this experiment. One class of dependent variables are the objective measurements of similarity between the sample images and the corresponding ground truth image. Three of these objective measurements are image dependent. They are the $L1$ and $L2$ norms and the structural similarity index (SSIM) [15]. The remaining objective measurement is a data dependent objective loss function denoted in this paper as VAS, introduced by Park, et al. which was used to construct ideal offline samples for scatter plots [12]. The other class of of dependent variables are the generated sample scatter plot images which will be correlated with aforementioned similarity metrics.

Independent Variable.

There are two independent variables in this experiment. One independent variable is the sampling size K which will be progressively increased up to the full dataset size N. The other independent variable (or class of variables) will the simulated data distributions which will be constructed to test two of the primary uses cases of scatter plots 1) trend analysis and 2) cluster detection. The two types of distributions studied will be normal and uniform. For density cluster detection, the number of normal distributions rendered will be independent as well.

Task.

To prove the claim that there exists an Image_{K^*} which is sufficiently similar to Image_N, the following tasks will be performed.

For trend analysis, there will be four types of data distributions explored: 1) a normal distribution correlated with a normal distribution, 2) a uniform distribution correlated with a normal distribution, 3) two uncorrelated normal distributions, and 4) two uncorrelated uniform distributions. For cases 1 and 2, the relations will be linear. For each data distribution, uniform random sampling will be performed to generate samples of increasing size up to the full dataset N. All of these samples will then be rendered on a scatter plot of fixed dimensions; the width and height will be 500 pixels with a marker radius for each data point set to 5 pixels. The Plotly graphics library will be used to generate the scatter plots. Each sample scatter plot image Image_k will be compared to the ground truth image Image_N using the $L1$ Norm, $L2$ Norm, $SSIM$, and VAS to generate multiple loss curves. The relevant regions of the loss curves will then be subjectively compared to each rendered Image_k to confirm the the rate of loss (or equivalently the increase in similarity) aligns with human perception.

A similar process will be repeated for cluster detection; the main difference will be the data distributions explored. In these experiments, each data distribution will have a different amount of clusters each of which will be normally distributed. The test cases to be explored will be 2, 3, 4, 5, and 10 clusters. With these test cases, the same procedure described for tend analysis will be repeated with the only difference being each cluster will have a different color coding.

Threats.

There are multiple threats facing this evaluation. First, to prove K^* is optimal, the ideal loss curve to observe across all test cases would be a large initial loss for very small sample sizes which precipitously drops to a sample size $K << N$ where the loss of K and N are approximately equal. It will likely prove difficult to observe an ideal loss curve or for one metric ($L1$ $L2$, $SSIM$, or VAS) to generate such a curve for all possible data distributions. Furthermore, the computer vision metrics being used to compare the sample scatter plot images to the ground truth scatter plot image may not be sensitive to small changes in the sample size for low values of K. These metrics were originally developed to compare photographic images and may not extend to scatter plot especially if the scatter plot has many sparse regions which the aforementioned metrics evaluate as equivalent. On the other end of the spectrum, VAS may be too volatile since it performs a pairwise comparison between the all the sample data points and the data points in the full dataset. VAS may be inappropriate for random

online sampling since it was designed to generate the best samples for specific scatter plots in an offline setting. Secondly, the if there exists an ideal sample size K^*, it may not be significantly less than N which would preclude QuickScatter's from being used in interactive environments. Finally, the generated scatter plot images may not align with the generated loss curve. For instance, it could be the case that a scatter plot image is highly similar to the ground truth which captures the correlation between two variables quite well, but the sparsity between points in the sample generates a high loss value with respect to the ground truth. This would especially be true if the ground truth has one or more dense regions in the case of clustering.

Why will this design directly test your thesis.

The design elaborated above directly tests that stated thesis that there exists an ideal sample point K^* generated from random online sampling such that $Image_{K^*}$ is similar enough the ground truth image $Image_N$ to support the scatter plot tasks of trend analysis and cluster detection. The key element of this experimental design is that the full datasets are simulated which means that the ground truth is known for all test cases. For trend analysis, this means the true correlation is known and the sampling points at which the correlation converges can be compared to generated images and the rate of change in the loss curves. Likewise for the cluster experiments, the number of clusters is known before hand which allows the perceived number of clusters in the images to be compared to the rates in change of the loss curves. If it is the case that measurable K^* does not exist, then the experiment will fail to produce a loss curve with a flat region between K^* and N where $K << N$. This could occur if the sparsity (prevalence of background) in the image dominates the distance measurements between samples images and the ground truth. Furthermore, it could be case that there does not exist a loss function which can capture such a relation or loss the functions have to be mapped to specific scatter plot tasks.

3 EVALUATION

In the following evaluation, we will test for the existence of the optimal sampling point K^* using simulated datasets. For each simulated dataset, uniform random samples of increasing size will be taken up to the full dataset. At each sampling point, the distance to each sample image will be measured with respect to the ground truth scatter plot image generated by the full dataset using similarity metrics from computer vision or data based metrics from offline sampling literature. The metrics used will be discussed in the following section. The simulated datasets are designed to validate the existence of K^* for two popular scatter plot use cases for scatter plots: trend analysis and cluster detection. The results of each use case will be discussed individually in subsequent sections.

3.1 Distance Metrics

To compare the sample images to the ground truth dataset, four different metrics were used. Two images of pixel width w and pixel height h can be easily compared using the Manhattan Distance $d = \frac{1}{wh}\sum_{x=1}^{w}\sum_{y=1}^{h}|I_{xy}(V_1) - I_{xy}(V_2)|$ and the Root Mean Square Error $d = \frac{1}{wh}\sum_{x=1}^{w}\sum_{y=1}^{h}\|I_{xy}(V_1) - I_{xy}(V_2)\|$ where $I_{xy}(V)$ is the luminance value of a pixel in the scatter plot image V. In addition to Euclidean measures, the Structural Similarity Index was used [15]. This index accounts for the structure of an image which is relevant for comparing clusters with similar shapes. The range of values for SSIM are bounded between 0 and 1 with 0 implying no similarity and 1 implying high similarity.

In addition to these image dependent metrics, a data dependent loss function developed by Park, et al. was adopted as a distance measure [12]. This loss function, refereed to as VAS in this paper, is a point loss objective function which compares all the sampled data points to the full dataset. The loss function penalizes samples which do not include some data points s near a data point x of the full data set when projected to the xy-plane. Furthermore, it only provides diminishing improvements if x already has neighbors s near it thus under-weighting dense regions in the scatter plot. The VAS loss of a given sample S is evaluated via the following formula:

$$S = \text{Sample} \qquad D = \text{Full Dataset}$$

$$x, s \qquad \text{Projections of data points in D and S}$$

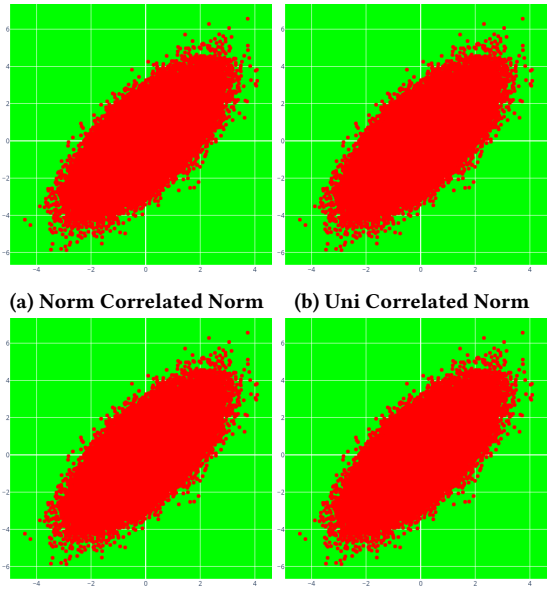
$$\text{Loss}(S) = \int \frac{1}{\sum_{s_i \in S} k(x, s_i)}\, dx, \qquad k(x, s_i) = \exp\frac{-\|x - s_i\|^2}{\epsilon^2}$$

3.2 Trend Analysis

3.2.1 Simulated Datasets. For trend analysis, four different datasets were generated. The datasets were selected to investigate the location of K^* for multiple scenarios which may occur when an analyst is comparing two variables. These data sets are 1) a normal distribution correlated with another normal distribution, 2) a uniform distribution correlated with normal distribution, 3) two uncorrelated normal distributions, and 4) 2 uncorrelated uniform distributions. Note the datasets account for various levels of randomness which may be present in the real world with dataset 1 being the least random and dataset 4 being the most random.

Each dataset contained 100,000 data points and was rendered to a fixed size scatter plot with width and height dimensions of 500 pixels each and marker radius of 5 using the Plotly Python Graphics Library. It was originally intended to test with 1 million data points but Plotly imposed memory constraints which prevented such figures from being created. It is believed that the results generated from this evaluation would extend to 1 million data points, but we will leave this investigation for future work. (TODO: Generate other 4 ground truth images.)

3.2.2 Results. (Note to reader: This is rough draft. Not all results are present.) The normalized loss curves observed for the Manhattan, RMSE, and SSIM measurements across the four test cases are displayed in Figure 2. The y-axis represents the normalized distance of each sample scatter plot image to the corresponding ground truth image. The distances were normalized in order to fairly compare changes in distance measurements between each metric. Note that for SSIM, the image quality is inverted with $1 - SSIM$ so that all three metrics measure the same value for the terminal case $K = N$. For test cases 1-3, small sample sizes (less than 1% of the data) exhibited highly variable distance values. As the sample sizes approached 10% of the full dataset ($K = 10,000$), the measured distances were resistant to change until over 50% of the data was sampled. At greater than 50% of the data, the samples were able



(a) Norm Correlated Norm (b) Uni Correlated Norm

(c) Norm Uncorrelated (d) Uni Uncorrelated

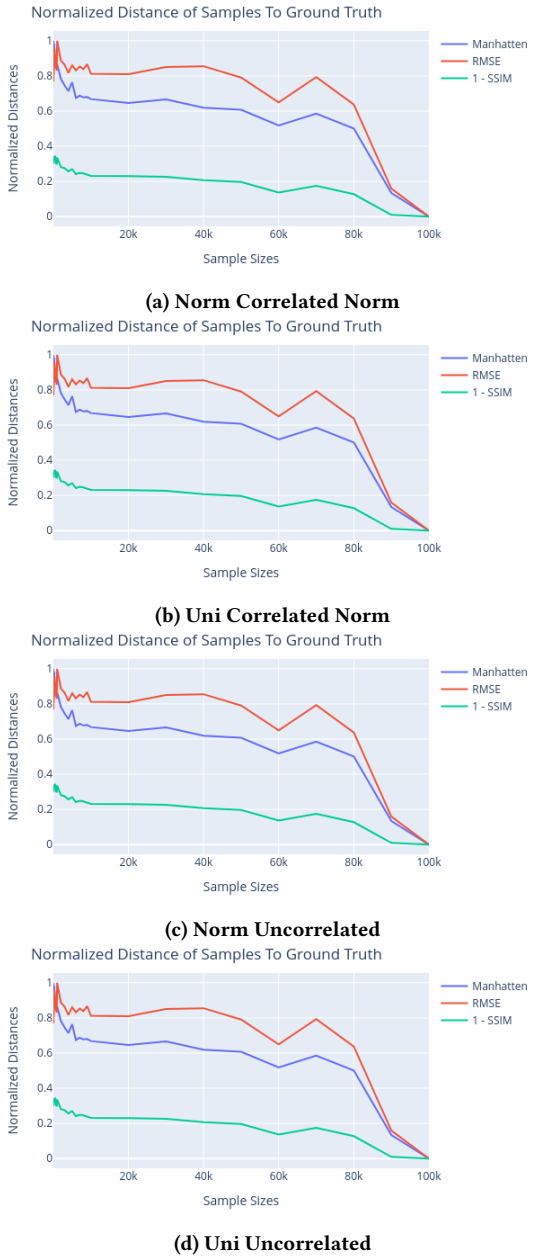
Figure 1: Ground Truth Images for Trend Analysis

to include more and more outliers. These results suggest that if an analyst is exploring trends in large datasets, no more than 10% of the data should be collected to generate a representative scatter plot. This claim assumes the analyst has no interest in outliers. The completely random dataset represented by test case 4, displayed a steady rate in decreasing loss values until the entire 2D space was filled. This point occurred after the 10% threshold indicating that the for highly entropic datasets, further sampling is needed to visually represent a completely random, uncorrelated dataset.

Figure 3 shows the changes in VAS distance over all four test cases above. Since VAS is a data dependent point-wise comparison between sampled and full dataset points, the largest sample size which could be accommodated before memory crashes was $K = 10000$. Furthermore, the values produced by VAS varied by orders of magnitude since the VAS was developed as an objective function to select the best K points for offline sampling. This means different sample selections at a fixed K were by design measurably different. This leads to a highly variable loss function for the online sampling use cases which needs to be displayed on a log scale.

(TODO: Show VAS loss for all 4 test cases. Show variance measures from random sample at 1% and 10% sample rates.) Despite it's variance in values, VAS displays the same general trend as the image dependent metrics indicating that samples sizes should not exceed 10% of the data. Furthermore, VAS indicates that further data reductions are possible form 10% as indicated by the relatively small decrease in distance values from $K = 1000$ to $K = 10000$ data points.

To confirm the 10% claim above, the sample scatter plot images for 0.1%, 1.0%, and 10.0% were subjective compared to there ground truth image. These samples for test case 1 are displayed in Figure 4. As expected, the at 1% we obtain a sparse representation of the



(a) Norm Correlated Norm

(b) Uni Correlated Norm

(c) Norm Uncorrelated

(d) Uni Uncorrelated

Figure 2: Ground Truth Images for Trend Analysis

true correlation between attributes X and Y. At 10%, we are able to obtain the density in addition to the general shape of the data.

3.3 Cluster Detection

3.3.1 Simulated Datasets. When multiple clusters were observed, we were also interested in finding a lower sampling bound K. Moreover, we wanted to analyze how the lower sample bound K in a multiple cluster scenario compares with the K value found in a singular cluster case. We applied the same random sampling methodology as discussed previously and our metrics were consistent with the trend

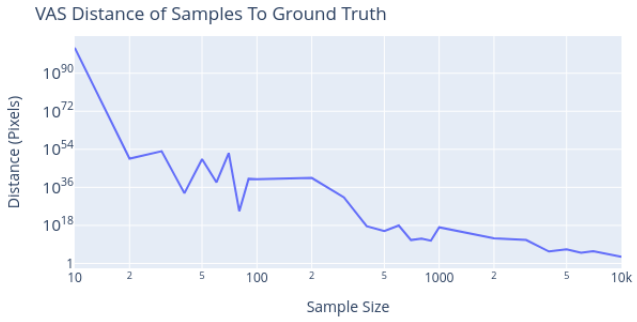
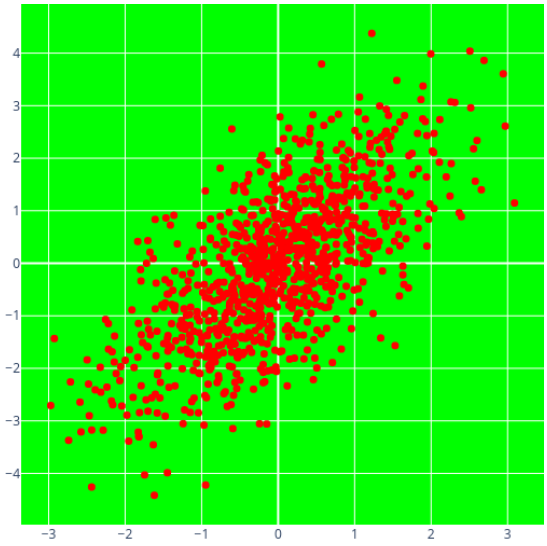


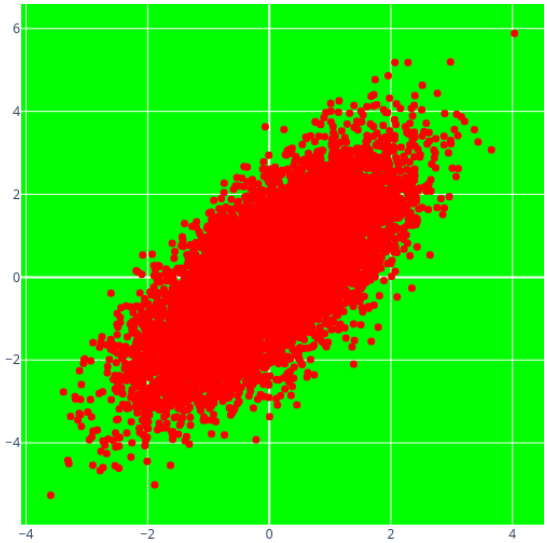
Figure 3: VAS Loss Curves

analysis. Once we established a baseline for our upper sampling bound K, we experimented with other methods to show that this initial result could be improved. To elaborate, we applied DBSCAN on random samples of size K' <= K and then showed that these smaller sample sizes in fact captured the number of clusters present in the larger dataset when all the points were visualized. The DBSCAN method was applied here because it allows for proximity-based cluster detection. Given that our use case concerns a large number of points on a relatively small canvas, we felt this would be a good approach to use. After fine tuning the DBSCAN parameters such as epsilon(distance between samples to be considered neighbors) and min samples(the number of neighboring points needed for the current point to be considered a "core point"), we were able to compare our results and determine that our initial K value could in fact be reduced. The results section contains more explicit details, but essentially, running DB Scan on this new smaller K' value obtained the same number of clusters as the original dataset and the centers of clusters were visually consistent as well.

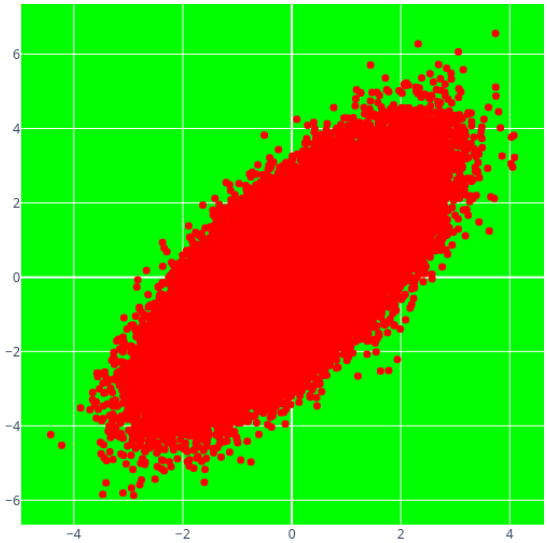
3.3.2 Results. (Please note that this is a rough draft and not all results are final. We may also experiment with additional clustering methods such KMeans/TSNE to determine which method best reduces the sampling size). As noted from the above discussion on trend analysis, the K value found previously remained in the range of 1-10 percent of N, where N is the total number of points. We observed that when multiple clusters are found, this value is generally around 10-15 percent of N. After this mark, the distance similarities only marginally improved. When we applied DBSCAN at this point, we found that we could discard about 1/3 of the sampled points and still obtain clusters that effectively captured the original scatterplot. So DBSCAN showed us that we can use random uniform sampling in the range [6.5-10] percent and still get desired results. While some downsampling in this form led to a lower similarity score, the clarity of the new scatterplot made up for this difference. In particular, the lack of distracting blobs made navigation between clusters substantially easier. To further verify the effectiveness of reducing sampling size via DBSCAN, we examined the proximity of outputted clusters' centroids with the cluster centers whose coordinates we determined visually based on a 90 percent confidence interval. We found that the cluster centers from the DBSCAN output matched up closely with our own user determined clusters and all the outputted clusters were in range



(a) 1% **Sample Rate**



(b) 10% **Sample Rate**



(c) 100% **Sample Rate**

Figure 4: Ground Truth Images for Trend Analysis

of the 90 percent confidence interval of our determined clusters. Thus, the DBSCAN approach allows for downsampling such that the positive aspects of visualization outweight the loss of distance similarity.

REFERENCES

[1] Leilani Battle, Philipp Eichmann, Marco Angelini, Tiziana Catarci, Giuseppe Santucci, Yukun Zheng, Carsten Binnig, Jean-Daniel Fekete, and Dominik Moritz. 2020. Database benchmarking for supporting real-time interactive querying of large data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1571–1587.

[2] Leilani Battle and Carlos Scheidegger. 2020. A structured review of data management technology for interactive visualization and analysis. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1128–1138.

[3] Enrico Bertini and Giuseppe Santucci. 2006. Give chance a chance: modeling density to enhance scatter plot quality through random data sampling. *Information Visualization* 5, 2 (2006), 95–110.

[4] Mihai Budiu, Parikshit Gopalan, Lalith Suresh, Udi Wieder, Han Kruiger, and Marcos K Aguilera. 2019. Hillview: A trillion-cell spreadsheet for big data. *arXiv preprint arXiv:1907.04827* (2019).

[5] William S Cleveland and Robert McGill. 1984. The many faces of a scatterplot. *Journal of the American statistical association* 79, 388 (1984), 807–822.

[6] Stephen G Eick. 1994. Data visualization sliders. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*. 119–120.

[7] Stephen G Eick, Joseph L Steffen, Eric E Sumner, et al. 1992. Seesoft-a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering* 18, 11 (1992), 957–968.

[8] Geoffrey Ellis and Alan Dix. 2007. A taxonomy of clutter reduction for information visualisation. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1216–1223.

[9] Danyel Fisher. 2016. Big data exploration requires collaboration between visualization and data infrastructures. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–5.

[10] Daniel A Keim, Ming C Hao, Umeshwar Dayal, Halldor Janetzko, and Peter Bak. 2010. Generalized scatter plots. *Information Visualization* 9, 4 (2010), 301–311.

[11] Zhicheng Liu and Jeffrey Heer. 2014. The effects of interactive latency on exploratory visual analysis. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2122–2131.

[12] Yongjoo Park, Michael Cafarella, and Barzan Mozafari. 2016. Visualization-aware sampling for very large databases. In *2016 ieee 32nd international conference on data engineering (icde)*. IEEE, 755–766.

[13] Sajjadur Rahman, Mangesh Bendre, Yuyang Liu, Shichu Zhu, Zhaoyuan Su, Karrie Karahalios, and Aditya G Parameswaran. 2021. NOAH: interactive spreadsheet exploration with dynamic hierarchical overviews. *Proceedings of the VLDB Endowment* 14, 6 (2021), 970–983.

[14] M Stonebraker, S Madden, DJ Abadi, S Harizopoulos, N Hachem, and P Helland. 2007. other: The end of an architectural era:(it's time for a complete rewrite). *33rd VLDB* (2007), 1150.

[15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[16] Eugene Wu, Leilani Battle, and Samuel R Madden. 2014. The case for data visualization management systems: vision paper. *Proceedings of the VLDB Endowment* 7, 10 (2014), 903–906.