# K-Anonymity & Outliers: EDA and Discussion

CS1050 Final Project

Eva Harris

# Table of Contents:

1 - Abstract
2 - Background
3 - EDA Experiment
4 - Outlier Detection (Research Analysis)
5 - Discussion
6 - Conclusion
7 - Code
8 - Works Cited

# 1 - Abstract

This paper addresses the shortcomings of k-anonymity and the challenges associated with outlier detection in real datasets. The analysis is divided into three main sections: an exploratory data analysis (EDA), a discussion centered around a case study on Massive Online Open Courseware (MOOCs), and a research analysis of outlier detection methods.

The first section involves performing an Exploratory Data Analysis (EDA) using a Medical Cost Personal Dataset (Medical Cost Personal Datasets) that I downloaded from Kaggle. A Python script is employed to analyze the data and calculate the level of k-anonymity. Generalization and suppression techniques are applied to the dataset, and their effects on k-anonymity are observed. Additionally, this section explores how the choice of different quasi-identifiers influences k-anonymity. The analysis provides insights into how various de-identification techniques impact privacy and anonymity levels in a dataset. [3]

The second section examines a case study on Massive Online Open Courseware (MOOCs) to explore the implications of de-identifying datasets. The case study evaluates the distortion in data analysis that results from de-identification, highlighting the trade-offs between preserving privacy and maintaining data utility. This section discusses the practical challenges of balancing these competing priorities.

The third section is a research analysis on outlier detection methods. This part examines various outlier detection techniques, with a focus on the challenges posed by outliers and their relationship to k-anonymity. Outliers, or anomalies in the data, can distort data distributions and undermine k-anonymity by making certain individuals more identifiable within a dataset. This section builds on classroom concepts and extends the discussion with further research to examine these implications in detail.

Together, these sections provide a comprehensive examination of the interplay between privacy-preserving techniques and outlier detection. By analyzing real datasets, this paper offers practical insights into addressing privacy and data integrity challenges in modern data analysis workflows.

# 2 - Background

## Key Terms

- **k-Anonymity**: A dataset is considered k-anonymous if each person's quasi-identifiers are the same as at least k-1 other people in the dataset.
- **Quasi-Identifiers**: Attributes or characteristics of a subject that, although they do not uniquely identify the subject on their own, can be combined with information from another dataset to reveal the subject's identity.
- **Outlier**: A data point that is significantly different from the majority of other values in a dataset. Outliers can skew data analysis results by being much higher or lower than other values.
- **Generalization**: A method in which data is categorized into groups, making values harder to re-identify and improving anonymity.
- **Suppression**: A method where entire columns are dropped or replaced with asterisks to remove sensitive data from analysis.

# 3 - EDA Experiment

In the EDA portion of the project, I used a medical insurance dataset to observe the effects of generalization and suppression on k-anonymity. I also analyzed the impact that the choice of quasi-identifiers would have on k-anonymity. I used a function to calculate the value of k-anonymity for each new version of the dataset, and with every revision, created an entirely new CSV that implemented the change that I wanted to observe.

## 3.1 - Functions:

### K-Anon Function

To calculate the k-anonymity of the original dataset, I implemented a function that accepts two parameters: the filepath (as a string) and the quasi-identifiers of the dataset (as a list of strings). This function loads the CSV file into a pandas dataframe, groups the data by the quasi-identifiers, and counts the rows in each group. The value for k-anonymity is determined by the size of the smallest group, which is then returned by the function.

### Generalization Functions

The **generalize_age** function takes an age as input and returns a corresponding age range category as a string. It uses a series of conditional checks to determine the appropriate category. If the age is between 18 (inclusive) and 25 (exclusive), it returns "18-25." If the age falls between 25 (inclusive) and 35 (exclusive), it returns "25-35." Similarly, ages between 35 and 45 map to "35-45," ages between 45 and 55 map to "45-55," and ages between 55 and 65 map to "55-65." For any age 65 or older, the function returns "65+." If the input age is below 18, the function returns "Under 18."

The **generalize_age_2** function takes an age as input and returns a corresponding age range category as a string. This function is a broader version of the original function to generalize age. It uses a series of conditional checks to determine the appropriate category. If the age falls between 35 (inclusive) and 55 (exclusive), it returns "35-55." Ages between 55 (inclusive) and 85 (exclusive) map to "55-85." For any age 85 or older, the function returns "85+." If the input age is below 35, the function returns "Under 35."

The **generalize_bmi** function takes a Body Mass Index (BMI) value as input and returns a corresponding health classification based on standard guidelines. If the BMI is less than 18.5, it returns "Underweight." If the BMI falls between 18.5 (inclusive) and 24.9 (exclusive), it classifies the BMI as "Normal weight." For BMI values between 25 (inclusive) and 29.9 (exclusive), it returns "Overweight." If the BMI is 30 or higher, it classifies the individual as "Obese."

The **generalize_charges** function takes a value from ‘charges’ as input and returns a corresponding charge range category as a string. It uses a series of conditional checks to determine the appropriate category. If the charges are less than 1000, it returns "less than 1000." If the charges fall between 1000 (inclusive) and 2000 (exclusive), it returns "1000-2000." Charges between 2000 (inclusive) and 3000 (exclusive) are categorized as "2000-3000," while charges between 3000 (inclusive) and 4000 (exclusive) map to "3000-4000." For charges between 4000 (inclusive) and 5000 (exclusive), it returns "4000-5000." If the charges are 5000 or higher, the function returns "5000+."

## 3.2 - My Experiment

First, I downloaded the initial dataset from Kaggle and installed the necessary packages to perform the analysis. The schema of the dataset showed that its variables had the following datatypes: ‘age’ (int64), ‘sex’ (object), ‘bmi’ (float64), ‘children’ (int64), ‘smoker’ (object), ‘region’ (object), and 'charges' (float64). There are 1338 rows in total in the dataset. Thus the highest possible value of k-anonymity that I could achieve in this experiment is 1338. However, this would entail suppressing all the columns, resulting in a dataset without data. This would be a quite useless dataset to work with.

The original dataset (**Version 0**) began with a k-anonymity value of 1. A k-anonymity of 1 means that each combination of quasi-identifiers is completely unique. There is absolutely no privacy protection. I used ‘age’, ‘sex’, ‘region’ and ‘BMI’ as my initial list of quasi-identifiers since these variables were most common to be found in other datasets.

Despite it being a unique float value, I did not include 'charges' in my initial list of quasi-identifiers because it is not a common piece of information to have access to on the internet. It is not normal for a dataset to have the exact amount of a person’s insurance charge because this is very personal information. However, if a bad actor obtains this exact value from the dark web and the dataset contains unique float values, the anonymity of the entire dataset could be compromised. In **Version 7** through **Version 10**, I experimented with the impact that the choice of my quasi-identifiers would have on the k-anonymity of the dataset. Over the course

of this experiment as a whole, I revised this dataset 10 times and observed the effect that each revision had on the k-anonymity. Below are the results of these trials.

| Version | Generalizations | Suppressions | Quasi-identifiers | K-Anon |
|---|---|---|---|---|
| 1 | Age (Version 1) | None | ['age', 'sex', 'region', 'bmi'] | 1 |
| 2 | Age (Version 1), BMI | None | ['age', 'sex', 'region', 'bmi'] | 1 |
| 3 | Age (Version 2), BMI | None | ['age', 'sex', 'region', 'bmi'] | 1 |
| 4 | Age (Version 2), BMI | Region | ['age', 'sex', 'region', 'bmi'] | 1 |
| 5 | BMI | Region, Age | ['age', 'sex', 'region', 'bmi'] | 8 |
| 6 | Age (Version 2), BMI | Region, Sex | ['age', 'sex', 'region', 'bmi'] | 2 |
| 7 | BMI | Region, Age, Sex | ['age', 'sex', 'region', 'bmi'] | 20 |
| 8 | BMI, charges | Region, Age, Sex | ['age', 'sex', 'region', 'bmi', 'charges'] | 2 |
| 9 | BMI | Region, Age, Sex | ['age', 'sex', 'region', 'bmi', 'charges'] | 1 |
| 10 | BMI, charges | Region, Age, Sex | ['age', 'sex', 'region', 'bmi'] | 20 |

In my initial dataset, I noticed that there were a lot of float values. By nature, float values are often unique, which inhibits k-anonymity. Thus, in **Version 1**, I began by simply generalizing age with the **generalize_age** function. After just generalizing age, there was no direct reflection in the k-anonymity. I realized that this was likely because **BMI** was still a float value that needed to be generalized. However, when I made this revision in **Version 2**, the k-anonymity remained at 1.

In **Version 3**, I tried making the generalization for age more broad, and expanding the bin size with **generalize_age_2**. There was still no effect. The k-anonymity remained at 1. In an effort to get a reaction out of the dataset, I began suppressing columns. In **Version 4**, I suppressed the region while keeping the generalizations of age **(generalize_age_2)** and BMI **(generalize_bmi).** I still observed no effect on the k-anonymity.

Finally, **Version 5** was when I started to see progress. I decided to try suppressing age entirely. In **Version 5**, I suppressed age and region suppressed age entirely, while keeping the generalization of BMI **(generalize_bmi).** This made the k-anonymity jump to 8, however, suppressing the age entirely would eliminate the possibility for many valuable analytics. I decided that in **Version 6**, I would attempt to retain a higher k-anonymity by implementing a generalized version of age (**generalize_age_2**) and suppressing the sex column entirely. In **Version 6**, the k-anonymity dropped to 2.

In retrospect, I realized that because sex column had 2 possible values, the values for sex were already not very unique. There was a high probability that the values were already the same, especially since most of my dataset were males. Because of this, suppression barely had an effect at all when comparing **Version 6** with **Version 4**. In the next part of my experiment, I explored the importance of choice in quasi-identifiers.

In **Version 7**, I will now see what value I will achieve if I suppressed all quasi identifiers (age, region, sex) except for BMI **(generalize_bmi)**. This made the K-anonymity jump to 20. This is a drastic shift, but it's a shift that makes sense intuitively. After all, all of our quasi-identifiers have been suppressed except BMI, and even that has been generalized **(generalize_bmi)**. Let's see what happens if I generalize 'charges' and include it as a quasi identifier, but keep everything else the same. Thus, in **Version 8**, I suppressed all quasi identifiers (age, region, sex) except for BMI **(generalize_bmi)** and 'charges' **(generalize_charges).** My list of quasi-identifiers was ['age', 'sex', 'region', 'bmi', 'charges']. This made the K-anonymity drop back to 2.

Notice that I didn't even touch 'charges' until **Version 8**. This whole time, I left it in its original form, which was a float data type. It had entirely unique values (which you can clearly see by looking at the CSV), yet it did not change any of my previous results because it was not included in my list of quasi-identifiers. Yet as mentioned previously, because it is a precise value, it would hinder the overall privacy of the data.

Logically speaking, if an additional column was generalized, it should make the entire dataset more protected and this should be reflected in the metric used to quantify data privacy. In **Version 9**, I demonstrated that if I did not generalize this value and kept it in my list of quasi-identifiers, the k-anonymity of the dataset would return to 1. The actual dataset was the

same as **Version 7**, but now ‘charges’, a unique value, was included as a quasi-identifier. Further, in **Version 10**, I found that when I generalized a variable that was not a quasi-identifier, it did not make any impact on the k-anonymity, even though it made the data itself more private. In **Version 10**, I maintained the generalization of 'charges', but left it out of the list of quasi-identifiers. This made the k-anonymity go back to 20, which is the same as in **Version 7**. Intuitively, this makes sense when considering the method used to calculate k-anonymity. In **Version 7** and **Version 10**, the function is looking at the same columns for the calculation. The only thing that changed was that in **Version 10** I generalized 'charges' with **generalize_charges**, and this did not affect the k-anonymity because 'charges' was not a quasi-identifier.

This reveals a flaw, suggesting that the effectiveness of k-anonymity depends on the completeness of the chosen quasi-identifiers. Bad actors could easily take advantage of a poorly constructed list of quasi-identifiers. In this case, k-anonymity does not show how protected a dataset is. From my experiments in **Version 7** through **Version 10**, I conclude that a strong value of k-anonymity can be as much a result of an insufficient list of quasi-identifiers as it is of a legitimately, well-protected dataset.

## 3.3 - Results Summary

This experiment analyzed a dataset with 1,338 rows, aiming to assess the impact of different revisions on k-anonymity. The highest possible k-anonymity value (which was 1,338) would require suppressing all columns, rendering the dataset unusable. The original dataset (Version 0) began with a k-anonymity of 1, indicating that every combination of quasi-identifiers was unique and offered no privacy protection. Using 'age,' 'sex,' 'region,' and 'BMI' as the initial quasi-identifiers, I demonstrated that suppressing key variables like 'age' demonstrated the trade-off between enhancing privacy and preserving data utility. Suppressing less unique variables, such as sex,' had minimal effect due to inherent overlaps in the data, particularly given the dataset’s male-dominated entries.

The exclusion of the 'charges' variable from the list of quasi-identifiers offered insights into the limitations of k-anonymity. While 'charges' contained entirely unique float values, it did not influence k-anonymity unless explicitly added to the quasi-identifier list. Generalizing

non-quasi-identifiers did not impact the k-anonymity metric, even though it improved the overall privacy of the dataset as a whole. These findings underscore a key weakness of k-anonymity: its reliance on a complete and accurate list of quasi-identifiers. Poorly chosen quasi-identifiers can create a false sense of security, leaving datasets vulnerable to reidentification. Ultimately, this experiment demonstrated that a strong k-anonymity value could result from an insufficient quasi-identifier list rather than robust privacy protection, highlighting the importance of thoughtful quasi-identifier selection for effective data anonymization.

In this next section of my paper, I will be conducting a research analysis on outlier detection methods. Following my EDA, it occurred to me that by definition, there is a continuous tradeoff between outliers and k-anonymity. First, outliers challenge k-anonymity due to their inherent uniqueness. As explored more thoroughly in my EDA, the presence of unique values will invariably lead to a k-anonymity of 1. However, there are plenty of research uses to exploring unique cases. A blog post by “Stats and R” illustrates that deciding whether to remove or keep an outlier depends on the context of your analysis, how thorough your statistical tests are to outliers, and the degree to which the outlier deviates from other observations. Second of all, generalization and suppression might address data privacy concerns, but will hide or absorb outliers if applied in the context of k-anonymity. Following this, one can reason that over-generalization risks losing valuable insights from unique outliers.

# 4 - Outlier Detection

In terms of detecting outliers, let’s begin with the descriptive statistics. The following methods do not require that the data be normally distributed. Some outliers are simply erroneous values that are so far-fetched, that they are actually quite unrealistic. These would be easy to detect with a min/max function, or a plot. Either a **histogram** or a **boxplot** would suffice, however, a box plot would calculate (and represent) the outlier differently than the rest of the data. The boxplot calculates the data based on the interquartile range (IQR). The IQR is the difference between the first and third quartile. The data points that lie outside 1.5 times the IQR, are determined to be outliers and are plotted as points in the box plot. An interesting fact to note is that the interval used by the boxplot to determine outliers is equivalent to finding values outside of the 1st and 99th percentiles of the data. [8]

The **Hampel Identifier (or Hampel Filter)** is another method used to identify outliers and does not require the data to be normally distributed. This method only considers the median and the median absolute deviation (MAD). Intuitively, the datapoint is determined to be an outlier by the Hampel Identifier if the absolute difference between the datapoint of interest and the median is greater than three times the MAD.[5][8]

Alternatively, there are some outlier detection methods that depend on normally distributed data. To test if the data is normally distributed, it's common to either plot a histogram or examine a QQ plot. Three statistical methods that require the data to be normally distributed come to mind: the **Z-score method,** the **Grubbs' Test,** the **Tietjen-Moore Test,** and the **Rosner's Test** (which requires the data to be approximately normal). An outlier is determined by the **Z-score method** if the standardized score for each datapoint is outside of 3 standard deviations from the mean. The **Grubbs' Test** will determine if there is an outlier present in the dataset. It is not suitable for detecting multiple outliers. The null hypothesis is that there are no outliers in the dataset, while the alternative hypothesis is that there is one outlier in the dataset. The test statistic basically takes the largest absolute deviation from the sample mean, and puts it in units of the sample standard deviation. [4][7]

In the case of testing for multiple outliers, either the **Tietjen-Moore Test** or the **Rosner's Test** would be a better test. It's useful to note that in the case of one outlier, the **Tietjen-Moore Test** is actually equivalent to the **Grubbs' Test**. If the number of outliers is known, then the **Tietjen-Moore Test** is ideal. The null hypothesis is that there are no outliers in the dataset, while the alternative hypothesis is that there is $k$ number of outliers in the dataset (with $k$ being the specified number of outliers). However, if this value is unknown, then the **Rosner's Test,** (aka the generalized extreme student used deviation test), is a better test. In the **Rosner's Test,** aka the generalized extreme student used deviation test, we only need to know a ceiling value (maximum) for the number of outliers  The null hypothesis is that there are no outliers in the dataset, while the alternative hypothesis is that there is up to $r$ number of outliers in the dataset (with r being the maximum specified number of outliers). [4][7][9]

These are some of the most common outlier detection methodologies used today. As a whole, outliers play a complex role in data analysis, particularly in the context of k-anonymity.

Their inherent uniqueness challenges privacy protections while offering valuable insights for research, depending on the context and statistical rigor applied. Balancing the need for privacy with the retention of valuable information from outliers is critical for achieving robust and insightful data analysis.

# 5 - Discussion

The industries that utilize k-anonymity the most are often those that deal with highly sensitive personal information that needs to be protected, while still allowing for data analysis and research. There are particular privacy regulations that come with sharing data. In particular, the Health Insurance Portability and Accountability Act (HIPAA) and the Federal Educational Right to Privacy Act (FERPA) come to mind. HIPAA simply states that all identifying information (directory information) must be removed from the dataset. FERPA states that the dataset must have a total of at least 5 data points that exhibit the same characteristics. To put this into the perspective of this paper, FERPA requires a 5-anonymity. By definition, making a dataset k-anonymous would remove the directory information (thus, meeting HIPAA's standards as well). [11]

However, there is often a significant amount of bias that results from anonymization techniques such as generalization and suppression. As examined in *"Privacy, Anonymity, and Big Data in the Social Sciences"*, the de-anonymization of a dataset from Massive Online Open Courseware (MOOCs) led to compromised reproducibility of the analysis from the original dataset. The de-identified dataset simply produced different results. In an ideal case, whether or not a result is statistically significant would be enough to determine if a result is actually valid or simply a figment of statistical bias. Yet for example, the de-identified dataset cut the number of students that were certified as a result of the course, by half. It would be impossible to realize that this erroneous result was a matter of statistical bias due to the de-identifying process had the same analysis not been executed on the original dataset. [1]

Similarly, this data was also used for the piece *"How to De-Identify your Data"*. it was found that increasing the k-anonymity would significantly reduce the mean grade of the students in the dataset. However, this distortion could be combated by deleting quasi-identifier columns

whose values' frequency of occurrence were highly correlated with numeric attributes. After coming to this realization, the theory was then tested by manually increasing the correlations. When correlations were manually increased, further distortion was found in the numeric data. In short, stronger correlations had led to a greater distortion of the data following de-identification. [6]

Another finding from this paper was in the inverse relationship between distortion in the mean of certain attributes and distortion of the correlation between quasi-identifiers and numeric attributes. Generalization can reduce distortion by preventing records with rare quasi-identifier values from being suppressed. When bin sizes were increased, fewer rows required suppression, and the mean grade approached its true value. However, this made the distortion of the correlation between quasi-identifiers and numeric attributes increase. [6]

Generalization particularly impacts values that have a heavy skew because values that are less frequent yet highly variable would be lost. These values are more commonly seen as outliers. For example, frequent forum contributors were grouped in a way that introduced statistical bias when analyzing the correlation between forum posts and grades. Connecting this to my EDA, it is important to mention that one of the biggest risks in attempts to anonymize data is if there is an outlier of interest. By the nature of k-anonymity, the goal is to eliminate values that stand out. There is no way to really "generalize" an outlier while retaining its uniqueness. By doing this, it remains a privacy risk even after being de-identified. In **Version 5**, I converted all my variables to categorical values except for 'children' and 'charges. Then by **Version 8**, all the variables were categorical except for 'children'. These string values make it significantly more difficult to discern outliers, but still gives the data protection. The piece *"How to De-Identify your Data"* considered this phenomena, and came up with a potential solution: to implement varying bin sizes. The main idea is that smaller bins were used for frequently occurring values and larger bins were used for less frequent values. This was designed to preserve the integrity of outliers or extreme values without undermining the accuracy of more common data points, thus balancing the trade-off between privacy and data utility. [6]

Currently the primary methods used for data privacy are k-anonymity and differential privacy. In differential privacy, the dataset itself is kept secure, while the researcher queries the data for values without accessing it directly. By using this method, the data provider can restrict

the types of queries allowed and add statistical noise to the results. This protects individual privacy, and when applied correctly, will make it mathematically certain that it is not possible for individuals in the dataset to be re-identified. However, making this system a reality is quite difficult and still poses a significant risk for erroneous data analysis. Unfortunately, both k-anonymity and differential privacy are not immune to the tradeoff that results between data privacy and statistical bias. [11]

# 6 - Conclusion

By definition, outliers stand out from a dataset. The premise of k-anonymity is for values to “hide in a group”, which would make it impossible to have occurrence of outliers in a k-anonymous dataset. My EDA explored the flaws of k-anonymity. I used direct application to observe the impacts of impossible  unique nature, outliers directly

This paper explores the intricate relationship between k-anonymity, outlier detection, and data de-identification techniques through a detailed experimental analysis. Using a dataset of 1,338 rows, it was demonstrated that achieving the highest possible k-anonymity would require suppressing all columns, rendering the dataset unusable. Through multiple revisions, it became evident that thoughtful selection of quasi-identifiers is critical for achieving effective data anonymization. For instance, variables like "charges," which contained entirely unique float values, did not impact k-anonymity unless explicitly included in the quasi-identifier list. This revealed a significant limitation of k-anonymity: a strong value might reflect an insufficient quasi-identifier list rather than genuine privacy protection. The analysis also underscored the delicate trade-off between generalization and suppression to enhance k-anonymity while preserving data utility and addressing the unique role of outliers in privacy frameworks.

The analysis explored the inherent challenge outliers pose to k-anonymity, as their uniqueness fundamentally contradicts the premise of values "hiding in a group." A research analysis of statistical tests for outlier detection provided insights into techniques researchers could employ. These were: Min/max function, Histogram, Boxplot, Interquartile range (IQR), Hampel Identifier (or Hampel Filter), Median absolute deviation (MAD), Z-score method, Grubbs’ Test, Tietjen-Moore Test and Rosner’s Test. The discussion expanded on the impact of

de-identification, using a case study on MOOCs to demonstrate how generalization and suppression could introduce statistical bias and distort data utility. For instance, de-identifying datasets halved the number of students certified by a course, underscoring the risks to analysis accuracy. While generalization reduced distortions in mean values by grouping rare quasi-identifiers, it often increased distortions in correlations between quasi-identifiers and numeric attributes.

Subsequently, the discussion delved into the consequences of de-identifying datasets, particularly through a case study on MOOCs. The findings highlighted how de-identification could compromise the accuracy and reproducibility of analyses, as seen when de-identified datasets produced significantly distorted results. For example, the number of students certified by a course was halved in a de-identified dataset, showcasing the statistical bias introduced during the process. Further exploration showed that generalization could reduce distortion in mean values by grouping rare quasi-identifiers, though this often increased distortion in correlations between quasi-identifiers and numeric attributes.

To address the challenges posed by current data privacy methods like k-anonymity and differential privacy, it may be more effective to focus on the intent of the researcher when de-identifying a dataset. A potential solution could involve researchers submitting a statement outlining the specific aspects of the dataset they intend to analyze before anonymization techniques are applied. This proactive approach would allow data providers to tailor the anonymized dataset to preserve the columns essential for the intended analysis while suppressing only the most unnecessary or privacy-sensitive information.

In this framework, unnecessary columns could be suppressed entirely, while columns critical to maintaining data utility but potentially inhibiting privacy could be generalized. One limitation of this method could be the risk of overlooking unexpectedly important columns, which could go undetected by the researcher during the initial planning stage. However, ideally, this would be circumvented by the data provider since they would likely see an important column when analyzing correlations during the process of refining the anonymized dataset.

# 7 - Code

Link to my github repo: https://github.com/Erow4/K-Anonymity_EDA.git

# 8 - Works Cited


[1] Alley. “Privacy, Anonymity, and Big Data in the Social Sciences.” *Communications of the ACM*, *Communications of the ACM*, 31 Aug. 2023, cacm.acm.org/practice/privacy-anonymity-and-big-data-in-the-social-sciences/#T1.

[2] Angiuli, Olivia, and Jim Waldo. “Statistical Tradeoffs between Generalization and Suppression in the De-Identification of Large-Scale Data Sets.” *2016 IEEE 40th Annual Computer Software and Applications Conference*, IEEE, 2016, pp. 589–593, doi:10.1109/COMPSAC.2016.198.

[3] Choi, Miri. “Medical Cost Personal Datasets.” *Kaggle*, 21 Feb. 2018, www.kaggle.com/datasets/mirichoi0218/insurance.

[4] “Grubbs’ Test.” *Grubbs’ Test - Data Processing Compendium - Workflows for Knowledge Exploitation in the Process Industries 2.0*, dataprocessing.aixcape.org/Algorithms/GrubbsTest/index.html. Accessed 15 Dec. 2024.

[5] “Hampel Identifier.” *Hampel Identifier - Data Processing Compendium - Workflows for Knowledge Exploitation in the Process Industries 2.0*, dataprocessing.aixcape.org/Algorithms/HampelIdentifier/index.html. Accessed 15 Dec. 2024.

[6] “How to De-Identify Your Data.” *How to De-Identify Your Data - ACM Queue*, queue.acm.org/detail.cfm?id=2838930. Accessed 15 Dec. 2024.

[7] National Institute of Standards and Technology (NIST). *NIST/SEMATECH e-Handbook of Statistical Methods*. National Institute of Standards and Technology, www.itl.nist.gov/div898/handbook/index.htm. Accessed 15 Dec. 2024.

[8] “Outliers Detection in R.” *Stats and R*, statsandr.com/blog/outliers-detection-in-r/. Accessed 14 Dec. 2024.

[9] “Rosnertest: Rosner’s Test for Outliers.” *RDocumentation*, www.rdocumentation.org/packages/EnvStats/versions/2.3.1/topics/rosnerTest. Accessed 15 Dec. 2024.

[10] “Tietjen-Moore Test.” *Tietjen-Moore Test - Data Processing Compendium - Workflows for Knowledge Exploitation in the Process Industries 2.0*, dataprocessing.aixcape.org/Algorithms/TietjenMooreTest/index.html. Accessed 15 Dec. 2024.

[11] Waldo, Michael Smith, and Jim Waldo. “Anonymity, de-Identification, and the Accuracy of Data.” *Harvard Online*, www.harvardonline.harvard.edu/blog/anonymity-de-identification-accuracy-data#:~:text=The%20basis%20is%20that%20if,the%20accuracy%20of%20statistical%20analyses. Accessed 15 Dec. 2024.