Aalto-yliopisto
Aalto-universitetet
Aalto University

ELEC-E7130

Internet Traffic Measurement and Analysis

Sampling

Assignment #7

Aitor Urruticoechea Puig
aitor.urruticoecheapuig@aalto.fi
Student N°101444219
November 2023

Contents

List of Figures

Task 1: Sampling and Distributions

1.1 Historiogram

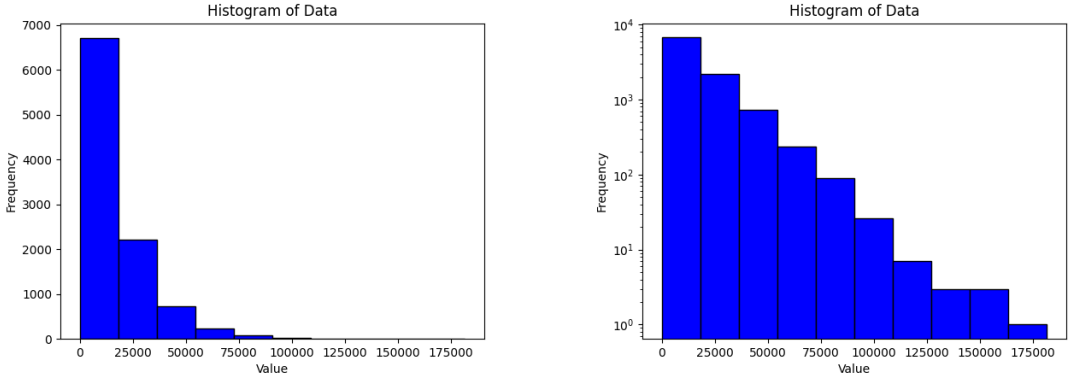
Since this task will require multiple plottings of historiograms, means, and other statistical data, a function to be called every time that is needed has been devised:

```python
def histmean(data):
    plt.hist(data, bins=10, color='blue', edgecolor='black')
    plt.title('Histogram of Data')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.show()

    plt.hist(data, bins=10, color='blue', edgecolor='black')
    plt.title('Histogram of Data')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.yscale('log')
    plt.show()

    mean_value = statistics.mean(data)
    print(f"Mean: {mean_value}")
    std_value = statistics.stdev(data)
    print(f"STD: {std_value}")
    return mean_value, std_value
```

For the whole data set, the mean is of 16432.250672819187. Figure 1 can be consulted for the historiogram in different scales.



(a) Linear scale.

(b) Logarithmic scale.

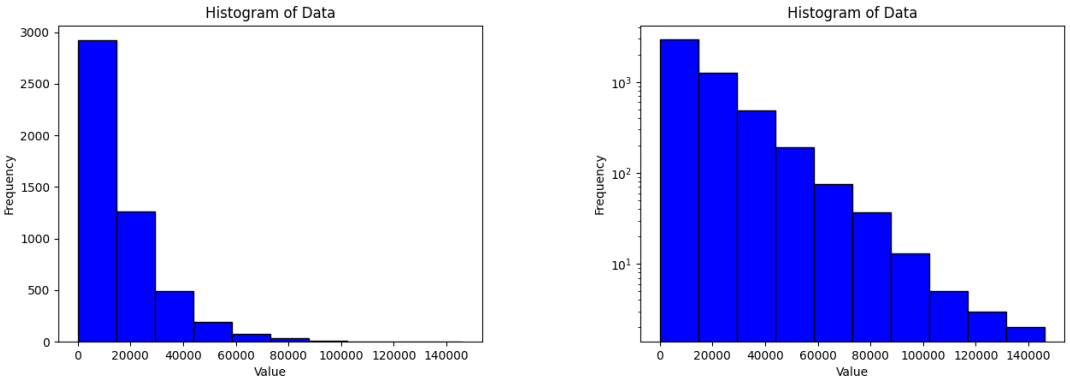
Figure 1: Historiogram of the whole data set.

1.2 5000 random samples

After selecting 5000 random samples using python's random module, a mean of 16433.736662148083 has been obtained. The historiograms for this case can be consulted in Figure 2.

1.3 10000 times n random elements

Computing the required parameters for this section is quite straight forward using python's statistics library. Three different cases of N have been analysed: 10, 100, and 1000.

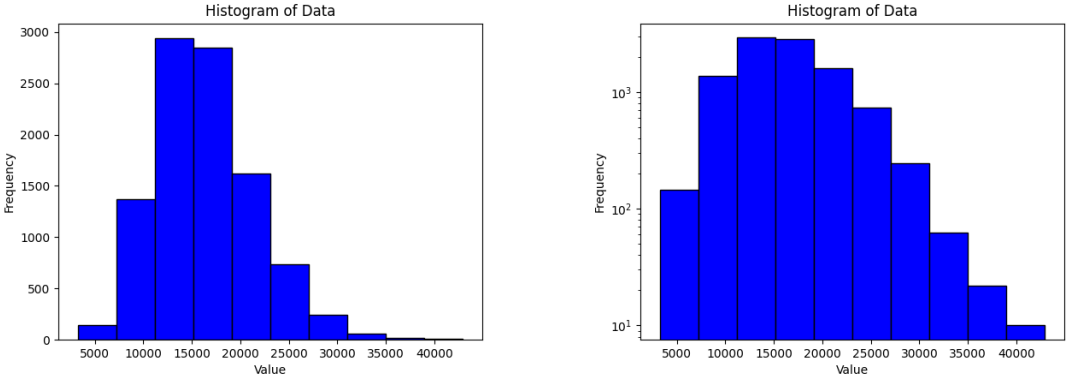


(a) Linear scale. (b) Logarithmic scale.

Figure 2: Historiogram of a random set containing 5000 samples.

For $n = 10$, the historiogram can be consulted in Figure 3, while the Q-Q plot is available in Figure 4. The obtained statistical values are:

- Mean: 16351.83936422154
- STD: 5274.579541224985
- Sample Error Mean: -80.4113085976478
- Variance: 27821189.33670918



(a) Linear scale. (b) Logarithmic scale.

Figure 3: Historiogram of a random set containing 10 samples.

For $n = 100$, the historiogram can be consulted in Figure 5, while the Q-Q plot is available in Figure 6. The obtained statistical values are:

- Mean: 16390.382651740147
- STD: 1648.6838041061508
- Sample Error Mean: -41.86802107904077
- Variance: 2718158.2859219285

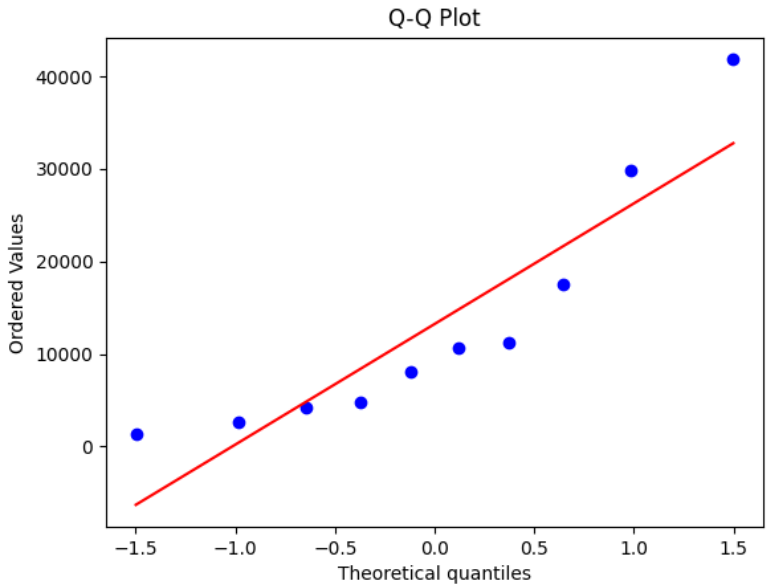
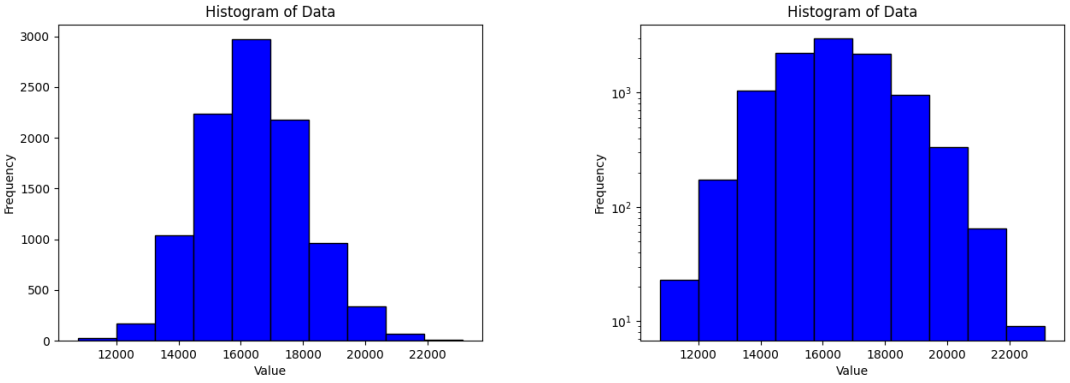


Figure 4: Q-Q plot for the random sample of 10 data points.



(a) Linear scale.

(b) Logarithmic scale.

Figure 5: Historiogram of a random set containing 100 samples.

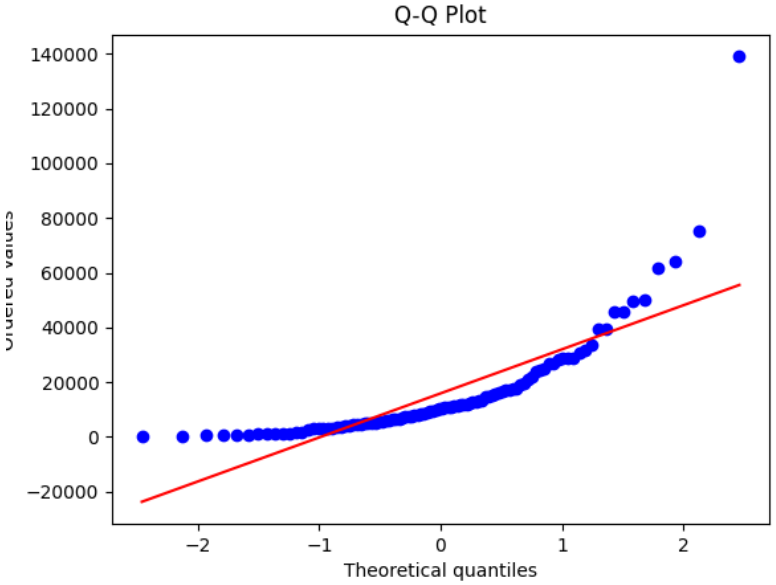
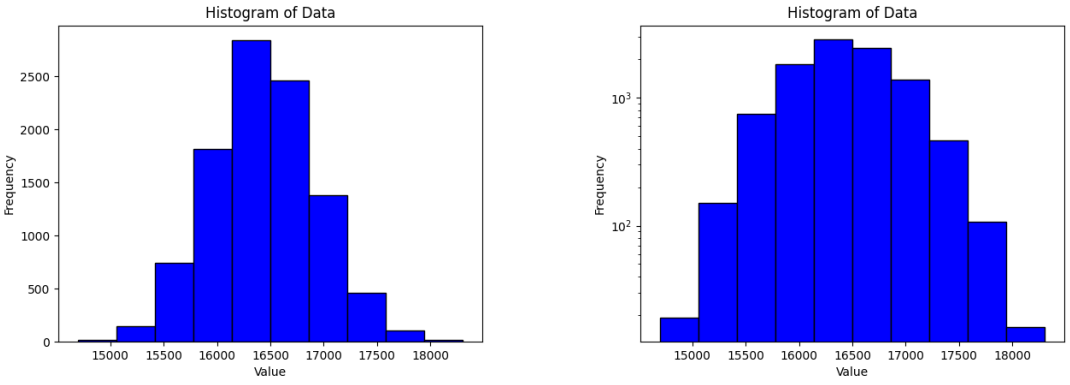


Figure 6: Q-Q plot for the random sample of 100 data points.

Finally, for $n = 1000$, the historiogram can be consulted in Figure 7, while the Q-Q plot is available in Figure 8. The obtained statistical values are:

- Mean: 16437.491651164473
- STD: 497.93193182946237
- Sample Error Mean: 5.240978345285839
- Variance: 247936.20873542034



(a) Linear scale.

(b) Logarithmic scale.

Figure 7: Historiogram of a random set containing 1000 samples.

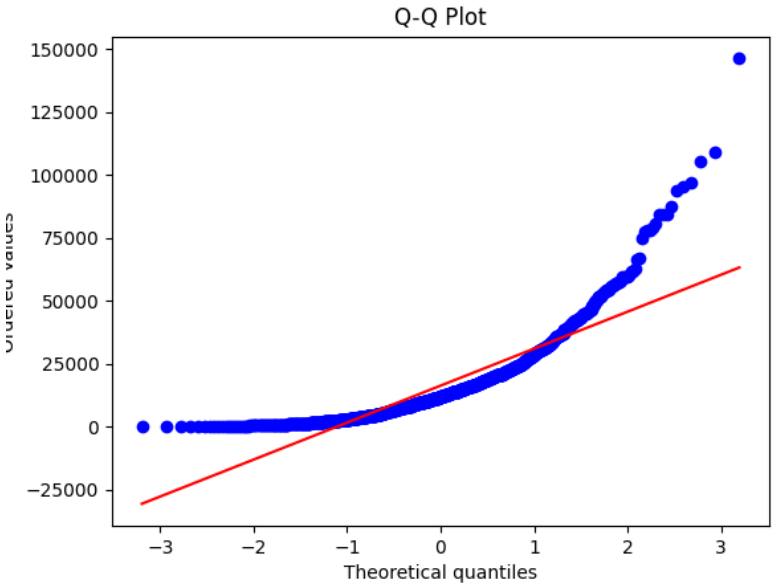


Figure 8: Q-Q plot for the random sample of 1000 data points.

1.4 Discussion

From the obtained figures, it is clear that the accuracy of the result notably grows the larger the sample is. As more data points are added, the more it resembles a normal distribution, and the more clear the Q-Q plot becomes. Similarly, the bias from outliers and less-frequent data points becomes negligible once N is big enough, which can be corroborated by the decreasing values obtained for sample error mean and variance.

Task 2: High variability (on-line sampling)

2.1 Original Data

Relevant statistical data has been computed for both packet and byte flow length:

- Mean Flow Length (Packets): 7.761738086904345
- Median Flow Length (Packets): 1
- Mean Flow Length (Bytes): 6015.98789939497
- Median Flow Length (Bytes): 40

An historiogram for both packets and bytes has been included in Figure 9.

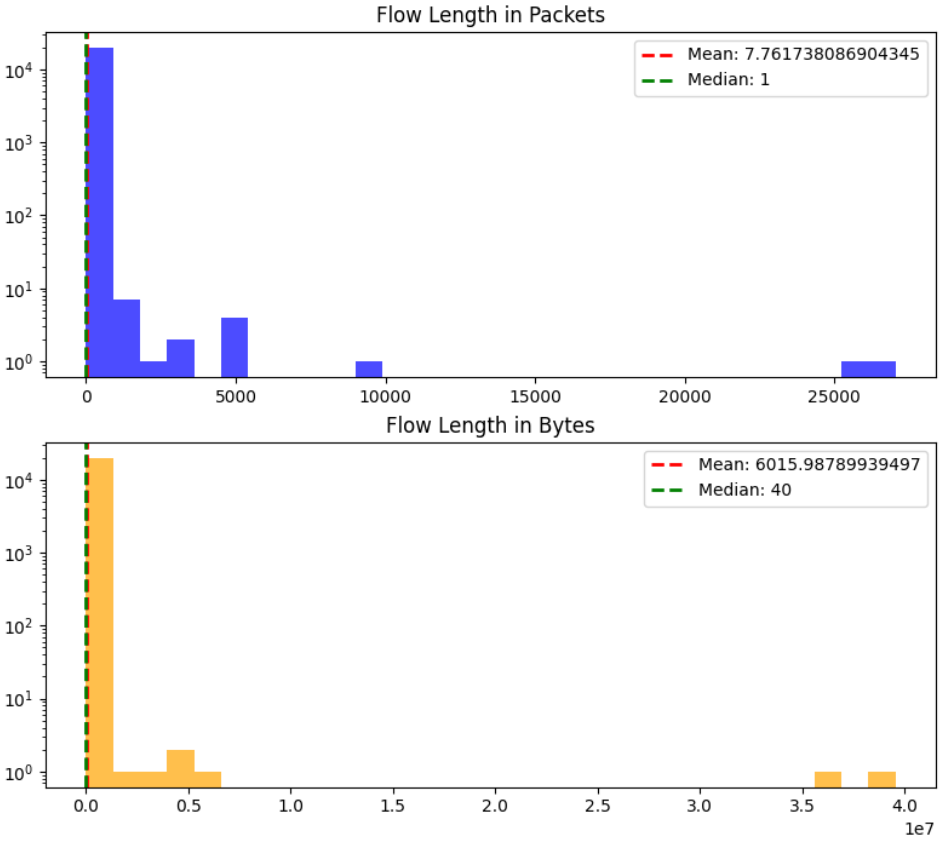


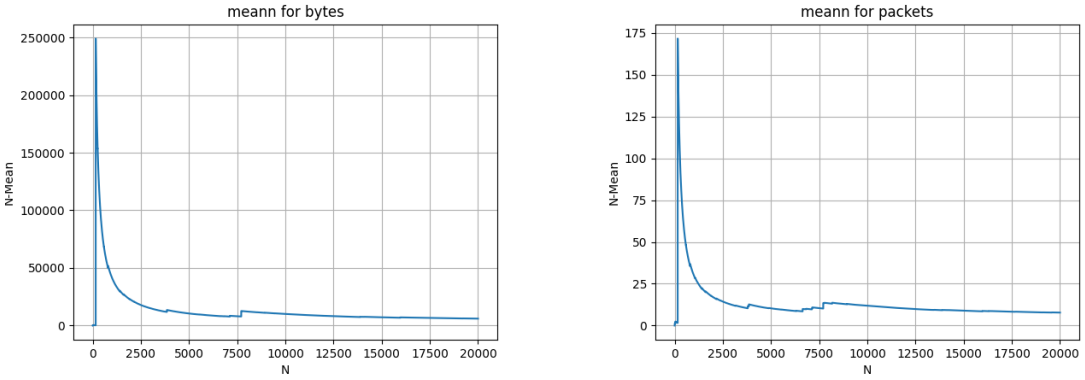
Figure 9: Flow length historiogram, for both byte and packet length

2.2 On-line measurement

As requested, a function named `running_mean` has been implemented. It takes two inputs, the data set and the n value, and it outputs the mean for the first n points of the dataset.

```
def running_mean(data, n):
    run = data.head(n)
    try:
        return mean(run)
    except:
        return 0
```

This function can then be called for ever-increasing numbers of n, which so that the outputted mean ever more resembles the actual mean of the whole data set. This behaviour can be clearly seen in Figure 10 for both packet and byte length.



(a) Case: Bytes.

(b) Case: Packets.

Figure 10: N value against obtained average.

If, rather than the mean, the median was the point of the study, the function could be more easily optimize. One could, for instance, use a data structure like a sorted list or a priority queue to efficiently update the median as new data points arrive. However, this would still have a time complexity of $O(n \log n)$ due to the having to perform a sorting operation for each window of data points. Methods more efficient could be devised for this exact purpose like, for example, a balanced binary search tree.

2.3 Conclusions

The most clear conclusion one can extract is that, when analysing data from a fixed, off-line, data set one has the advantage of it being fixed. No new updates to the data mean the conclusions to be extracted are not bound to change, and one can believe, for instance, the data obtained from the mean. In the case for a constantly-updating, on-line data set like the case for the running mean, one needs to necessarily reach a safe amount of N data points before computing any statistical conclusions. Outliers early in the measurement can have a huge impact if not enough flows are analysed to compensate for them. Of course, alternatively, one can rather use the mean to have a more resilient metric to such outliers.

Task 3: Data pre-processing for ML purposes

3.1 Preparing the dataset

The provided skeleton code has been completed using the required python methods. The result is a completely prepared, standardized and normalized data set. The output of the code is as follows:

```
        srcip   srcport     dstip   dstport   proto  duration
79   0.758621  0.002268  0.166667  0.001299  1.0000  0.000000
193  0.482759  0.823541  0.250000  0.000000  1.0000  0.578431
135  0.482759  0.748106  0.183333  0.152807  0.3125  0.000005
8    0.482759  0.554423  0.200000  0.000886  0.3125  0.000002
119  0.482759  0.650846  0.550000  0.005961  0.3125  0.000071
(200, 6)
```

3.2 Questions

3.2.1 Mention three types of probability sampling applied in ML apart from the one already mentioned.

- **Stratified sampling**: Stratified sampling involves dividing the population into distinct subgroups or strata based on certain characteristics, and then random samples are taken from each stratum. This ensures that each subgroup is represented in the sample proportionally to its presence in the overall population.
- **Cluster Sampling**: Cluster sampling involves dividing the population into clusters and then randomly selecting entire clusters to be included in the sample. Once the clusters are chosen, all members within those clusters are included in the sample.
- **Multistage Sampling**: Multistage sampling is a combination of various sampling methods. It involves multiple stages of sampling, where different techniques may be applied at each stage.

3.2.2 What is the purpose of encoding the values in ML?

The purpose of encoding is to enable the algorithm to effectively interpret and learn patterns from the data. Machine learning models are based on mathematical equations and operations, and they work best with numerical inputs. Proper encoding ensures that the model can effectively capture relationships and patterns in the data, allowing it to generalize well to unseen examples and make accurate predictions.

3.2.3 What are the differences between standardization and normalization in terms of feature scaling in ML?

Standardization, also known as z-score normalization, involves transforming the features so that they have a mean of 0 and a standard deviation of 1. On the flip side, and not to be confused, normalization scales the features to a specific range, often between 0 and 1.

Standardization is useful when the features in the dataset have different units or scales, and it helps algorithms that rely on distance measures. By contrast, normalization is beneficial when the features have varying ranges, and it is particularly useful for algorithms that are sensitive to the magnitude of the features, such as neural networks.