Aalto-yliopisto
Aalto-universitetet
Aalto University

ELEC-E7130

Internet Traffic Measurement and Analysis

Distributions and Sampling

Assignment #6

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

October 2023

Assignment #6
Aitor Urruticoechea Puig
October 2023

Contents

List of Figures

Acronyms

CDF Cumulative Distribution Function. 2, 3

PDF Probabiltiy Density Function. 2, 3

Task 1: Introduction to distribution and sampling

1.1 What is sampling in statistics, and how does it help us understand data distributions?

In statistics, one can refer to sampling as the process of separating a subset of data point from the whole population to extract insights about the general trends that the data follows overall. It is fundamental to study distributions, get parameters, and draw conclusions without needing to collect data from every individual of the population. It helps in numerous ways to understand the distribution of data, from easing visualization to hypothesis testing; and from estimation to efficiency improvements cost and time-wise.

1.2 Choose at least three distributions from the following options and explain their respective parameters and typical applications.

- **Normal Distribution**: Also known as Gaussian distribution, is one of the most popular distributions. It follows a symmetric, bell-shaped curve. To determine the centre and the spread of the curve, the key parameters are the mean μ and standard deviation σ.
- **Beta Distribution**: It is often used to model the distribution of random variables that represent proportions or probabilities, specially in Bayesian statistics, for it is constrained by definition between 0 and 1. The parameters that define one such distribution are α and β, and depending on their values the distribution can vary wildly, from symmetric to right or left-skewed.
- **Exponential Distribution**: It is commonly used in reliability analysis, queuing theory, and survival analysis to model the time until failure, waiting times between customer arrivals, and time until an event occurs. Its key attribute is that it is a memory-less distribution, meaning that the probability of an event occurring in the next time interval is independent of how much time has already elapsed. The Probabiltiy Density Function (PDF) for one such distribution is given by $f(x) = \lambda \cdot e^{(-\lambda x)}$ for $x \geq 0$; where λ represents the rate at which events occur.

1.3 What are the components of the following goodness-of-fit plots used to validate a model?

- **Probability-Probability (P-P) plot**: In this plot, the x-axis represents the theoretical Cumulative Distribution Function (CDF) values expected under the assumed theoretical distribution; while the y-axis represents the empirical CDF values calculated from the actual data. Thus, each point on the plot corresponds to a data point from the sample. A diagonal line (y = x) represents perfect agreement between the empirical and theoretical CDFs. Deviations from this line indicate differences between the observed data and the theoretical distribution. This kind of plots helps in assessing whether the tails or central part of the data distribution deviates from the theoretical distribution.
- **Quantile-Quantile (Q-Q) plot**: In one such plot, the x-axis represents the quantiles of the theoretical distribution; while the y-axis represents the quantiles of the actual data. Points on the Q-Q plot correspond to pairs of quantiles, one from the theoretical distribution and one from the empirical data. A 45-degree reference line is often included, where a perfect match between the data and the theoretical distribution would fall along this line. Departures from the reference line can reveal deviations between the data distribution and the theoretical distribution.
- **Comparison of probability density (empirical and theoretical)**: This type of plot typically consists of a graph that shows the PDF of the theoretical distribution and a histogram of the empirical data. The x-axis represents the values of the random variable, and the y-axis

represents the PDF values or densities. By overlaying the theoretical PDF with the empirical density estimate, one can visually compare how the two distributions align.

- **Comparison of CDF (empirical and theoretical)**: This plot compares the CDF of the theoretical distribution and the empirical CDF of the data. The x-axis represents the values of the random variable, and the y-axis represents the CDF values. The theoretical CDF is plotted as a curve, and the empirical CDF is represented by steps (a staircase-like pattern). Again, the comparison can help in assessing how well the empirical data follows the expected CDF of the theoretical distribution.

Task 2: Distributions

In order to analyse the three given datasets, a single fitting process has been devised so that the best distribution can be found no matter the distribution at hand. Thus, the same code has been applied to the three sets. The code is, in actuality, quite simple, for the main tool used is python's Fitter library.

```python
# File task2.py
# Aitor Urruticoechea 2023
from matplotlib import pyplot as plt
import numpy as np
from fitter import Fitter, get_common_distributions, get_distributions

choice = False
while not(choice):
    file = input("File to analyse (a/b/c): ")
    if file == "a":
        file = "sampling-data/distr_a.txt"
        choice = True
    elif file == "b":
        file = "sampling-data/distr_b.txt"
        choice = True
    elif file == "c":
        file = "sampling-data/distr_c.txt"
        choice = True
    else:
        print("Invalid choice")
data = np.loadtxt(file)

f = Fitter(data)
f.fit()
print(f.summary())
print(f.get_best())

plt.show()
```

This allows for switching between datasets, and no matter what Fitter's methods will return the best possible fits along with the statistical data to corroborate it.

2.1 Set A

For set A, the results are very tied between half and fold Cauchy and beta distribution; all three with remarkably low error and p-value results (Table 1). By a very small margin, the winner is the Half-Cauchy distribution defined by a centre at $-3.0713813432179294 \cdot 10^{-09}$ and a scale of 0.21569503589803207. The fitting can be visually checked in Figure 1.

Distribution	Error (sum square)	AIC	BIC	Statistic KS	P-value
halfcauchy	0.170020	1290.704533	1304.137933	0.203062	3.824105e-221
foldcauchy	0.170064	1292.764790	1312.914888	0.203037	4.334850e-221
beta	0.178155	5568.699858	5595.566656	0.235664	1.105389e-298
genpareto	0.214928	1164.286303	1184.436402	0.147104	9.417682e-116
invgauss	0.275493	1191.221441	1211.371539	0.183880	4.130902e-181

Table 1: Best fitted distributions for Set A.

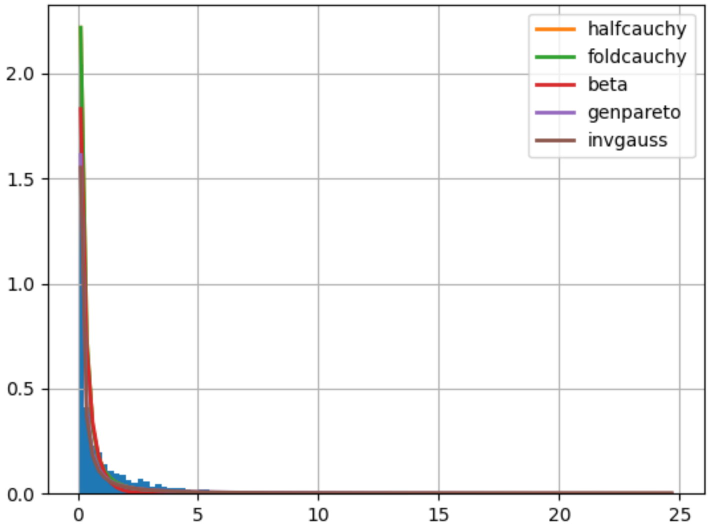

Figure 1: Set A: Best fitted distributions.

2.2 Set B

In Set B, Python's fitter library seems to fail, for the data is too discrete for it to function properly, thus resulting in quite large error (Table 2, Figure 2). Instead, the code has been slighlty modified to try to fit a normal distribution, which makes a lot more sense than the proposed ones when looking at the data. This can be very easily implemented with Python's SciPy library. The result is a normal distribution (Figure 3) with a μ of 6.05 and a standard deviation (σ) of 2.54.

```python
from scipy.stats import norm
# code skip
mu, std = norm.fit(data)
plt.hist(data, bins=30, density=True, alpha=0.6, color='g', label='Data Histogram')
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2, label='Fitted Normal Distribution')
plt.title("Fit results: mu = %.2f,  std = %.2f" % (mu, std))
plt.show()
```

Distribution	Error (sum square)	AIC	BIC	Statistic KS	P-value
dgamma	3.690515	758.796255	775.745333	0.146716	6.442808e-40
dweibull	3.696865	765.521282	782.470360	0.132078	2.098272e-32
exponnorm	3.720900	713.026952	729.976030	0.088806	7.345967e-15
invgamma	3.720995	717.155247	734.104325	0.083181	4.345369e-13
betaprime	3.720996	719.155295	741.754066	0.083180	4.347845e-13

Table 2: Best fitted distributions for Set B.

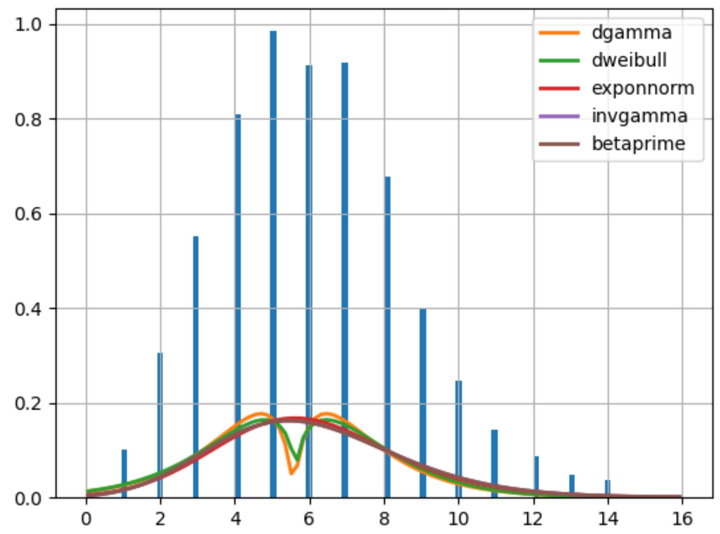

Figure 2: Set B: Best fitted distributions according to Fitter.

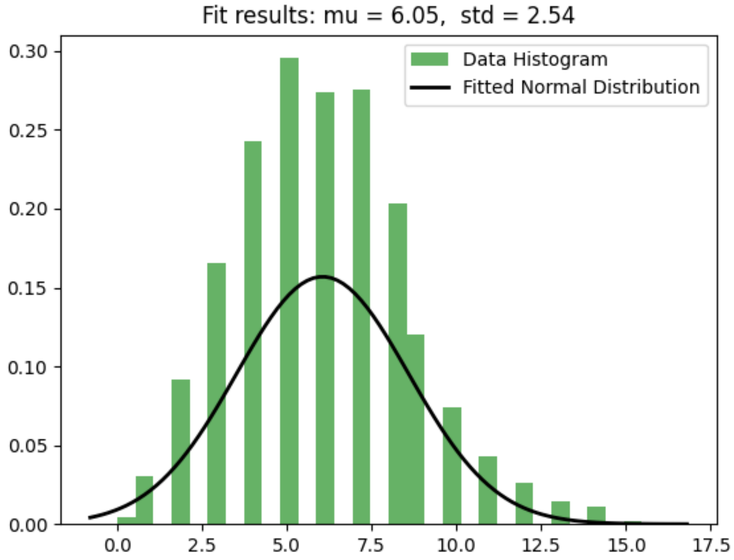

Figure 3: Set B: Fitting a normal distribution.

2.3 Set C

For the last dataset, the data is very heavily concentrated in quite few points, which has resulted in computationally-zero P-values for many distributions (Table 3). Nonetheless, the one with the lowest error is the half logistic, with a location point at 0.005912795687705464 and a scale value of 6580304512.607342. How well it fits can be checked out in Figure 4.

Distribution	Error (sum square)	AIC	BIC	Statistic KS	P-value
halflogistic	6.346631e-23	53003.647641	53015.693435	0.839741	0.0
exponnorm	2.734752e-22	51483.533427	51501.602117	0.814573	0.0
expon	2.755491e-22	51674.960185	51687.005979	0.815477	0.0
laplace_asymmetric	2.756170e-22	51681.250738	51699.319428	0.815452	0.0
gumbel_r	3.055286e-22	52877.839625	52889.885419	0.509870	0.0

Table 3: Best fitted distributions for Set C.

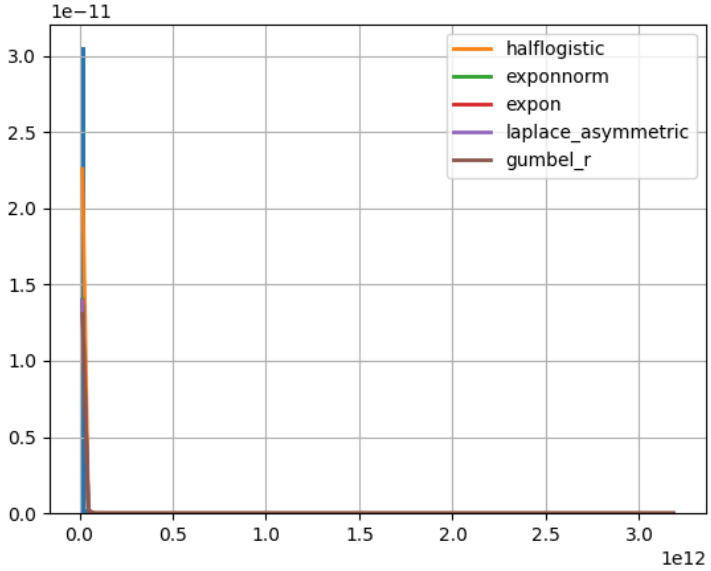

Figure 4: Set C: Best fitted distributions.

Task 3: Sampling

3.1 Overview of the data set

By selecting 1000 random data points from the whole dataset, a parallel plot can be easily done using matplotlib library. For this initial overview, the number of packets transmitted has been chosen as the colour column (see Figure 5).

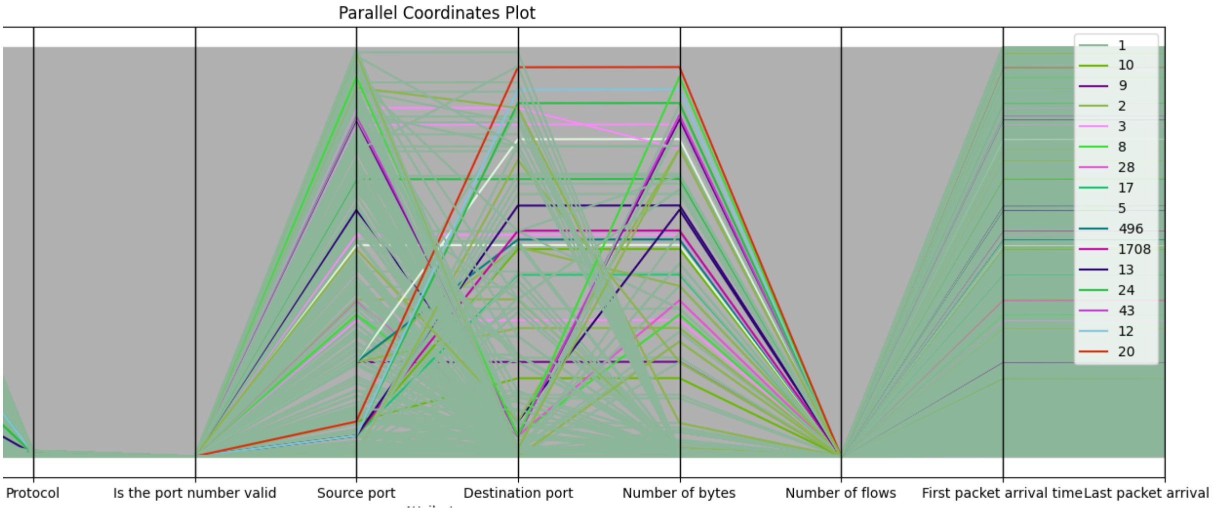

Figure 5: Overview: Parallel plot.

3.2 Number of bytes against packets

The scatter plots are easily created (as seen in previous assignments) using matplotlib library. Figures 6 and 7 show these, from which one can clearly see the overall pattern is not lost when only taking a small sample; but the nuance regarding the actual dispersion of the data is.

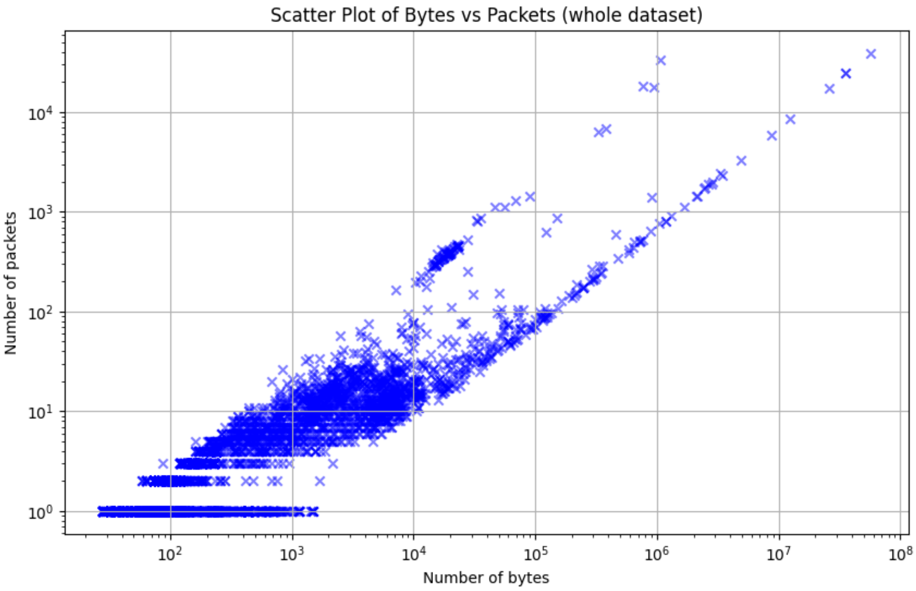

Figure 6: Scatter plot: whole dataset.

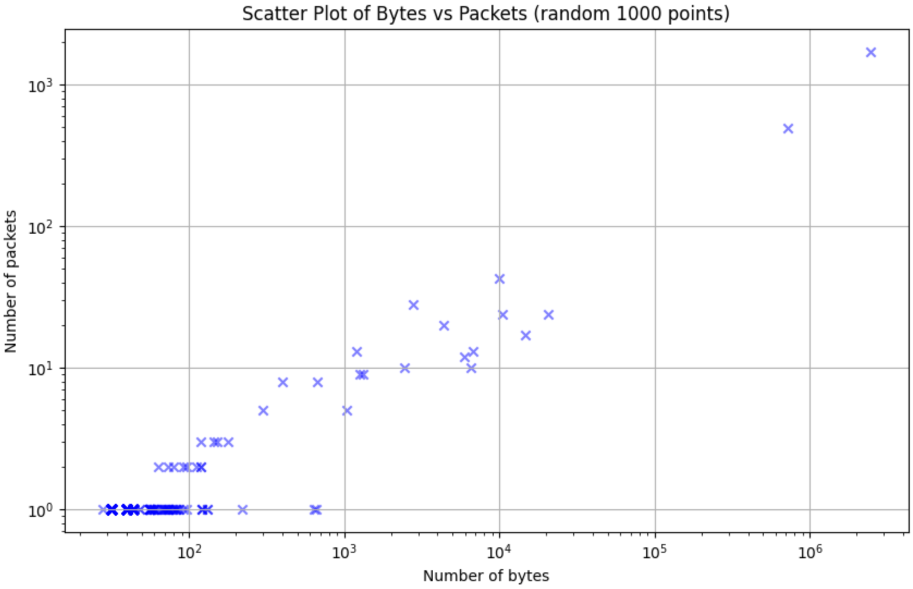

Figure 7: Scatter plot: random sample.

This nuance issue is exacerbated when one is to calculate the average packet size for the two datasets. Following the given formula,

$$Average\ packet\ size\ of\ a\ flow = \frac{total\ bytes\ of\ a\ flow}{total\ packets\ of\ a\ flow}$$

results in an average packet size of $611,192\ \text{bits}/\text{packet}$ when looking at the totality of the dataset, yet the number skyrockets to $953,806\ \text{bits}/\text{packet}$ when only using the sample. The difference is notable enough to be worrying; and showcases the limitations of sampling data instead of considering the whole set. In that sense, though, computationally-light measurements like averages should be performed using the whole dataset, while plots and other computationally-heavy operations are probably fine with only a sample.

3.3 Average throughput

Calculating the average throughput is an easy operation with a non-trivial problem. The average throughput of a flow is the number of bytes transferred divided by the transfer time; yet in many cases this transfer time results in zero. While flows with zero or very small time differences likely represent real-time data transfers, such as a single small packet or near-instantaneous communication (in such cases, the extremely high throughput value may be accurate and indicative of the nature of the connection); it can also be the result of practical limitations. Clock resolution and measurement limitations can lead to recorded time differences that are very close to zero, even if there is a small but real time difference. In such situations, the calculated throughput would be an overestimation.

The results on average throughput are of $3649.00\ \text{bits}/\text{s}$ when taking into account the whole dataset; and of $5572.62\ \text{bits}/\text{s}$ when looking only at the 1000 random points. This again, comes to show how outliers may overrepresented when looking only at a snippet of the whole data; and thus it is easy to obtain average data that is far from the real deal.