Network Analysis and Simulation - Homework 1

Michele Polese, 1100877

April 13, 2015

1 Estimators

The following MATLAB code describes the estimators used in all the scripts of this homework. They are written as functions for code reusability.

Estimator for q-quantiles

The q-quantile estimator of a dataset $\{x_1, ..., x_n\}$ is defined as $\frac{x_{(j)}+x_{(k)}}{2}$ with $\{x^n_{(1)}, ..., x^n_{(n)}\}$ the order statistic, $j = \lfloor qn + (1-q) \rfloor$, $j = \lceil qn + (1-q) \rceil$.

```
function [ quant ] = qquant_est( data, q )
% Estimator of the median
data = sort(data);
n = length(data);
kl = floor(q*n + (1 - q));
ku = ceil(q*n + (1-q));
quant = (data(kl)+data(ku))*0.5;
end
```

Estimator for the mean

The sample mean over a dataset $\{x_1, ..., x_n\}$ is defined as $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$.

```
function [ mean_es ] = mean_est( data )
% Estimator of the mean
    mean_es = sum(data)/length(data);

end
```

Estimator for the variance

The sample variance over a dataset $\{x_1, ..., x_n\}$ is defined as $\hat{s}^2_n = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu}_n)^2$ (unbiased estimator) or $\hat{s}^2_n = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu}_n)^2$ (biased estimator).

```
function [ s_2 ] = var_est( data, option )
% Unbiased or biased estimator for variance
m = mean_est(data);
diff = (data - m).^2;
if option == 0 % unbiased estimator
    s_2 = sum(diff)/(length(data) - 1);
elseif option == 1 % biased estimator
    s_2 = sum(diff)/(length(data));
else
    disp('Error, option is in [0, 1]');

end
```

Estimator for the autocorrelation

Biased estimator for autocorrelation can be found in [1]. Given the dataset $\{x_1, ..., x_n\}$ the sample autocorrelation sequence (ACS) is $\hat{\rho}_t = \hat{\gamma}_t/\hat{\gamma}_0$, with $\hat{\gamma}_t$ the sample autocovariance $\hat{\gamma}_t = \frac{1}{n}\sum_{s=1}^{n-t}(x_{s+t} - \hat{\mu}_n)(x_s - \hat{\mu}_n)$ ($\hat{\mu}_n$ is the sample mean).

```
function [ autoc ] = autocorrelation( x, N_corr )
% This function returns ACS of a signal x as defined in Le Boudec.
% N_corr is the number of desired samples for the ACS
x = x - mean(x);
K = length(x);
autoc = zeros(N_corr + 1, 1);
for n = 1:(N_corr + 1)
    d = x(n:K);
    b = conj(x(1:(K - n + 1)));
    c = K; % - n + 1 for the unbiased estimator
    autoc(n) = d.' * b / c;
end

% rescale
autoc = autoc/autoc(1);

end
```

2 Exercise 1

The following Figures contain plots from Figures 2.1, 2.2, 2.3, 2.7, 2.8 and 2.10 in [1]. In Figure 2 there is a plot of the empirical cumulative distribution function for the two dataset. Given a dataset $\{x_1, ..., x_n\}$ the ECDF is defined as $F(x) = \frac{1}{n}\sum_{i=1}^{n} 1_{x_i \leq x}$.

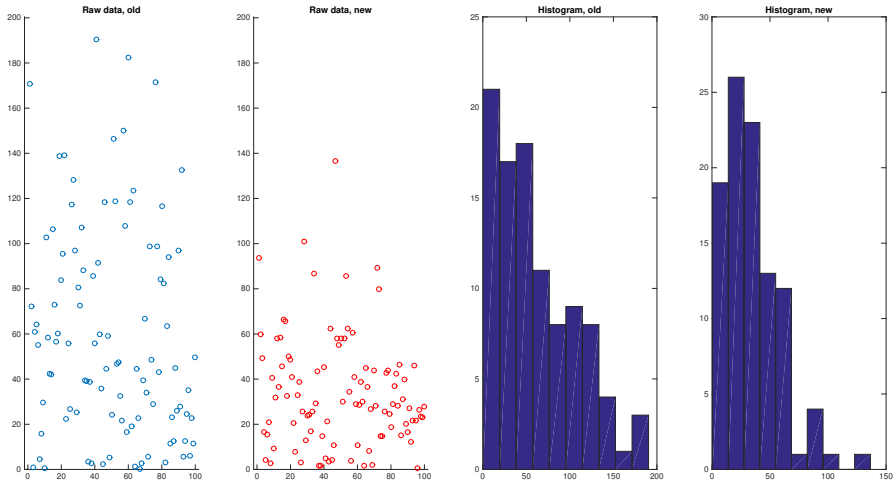

Figure 1: Figure 2.1 in [1]. Different way to compare two datasets.

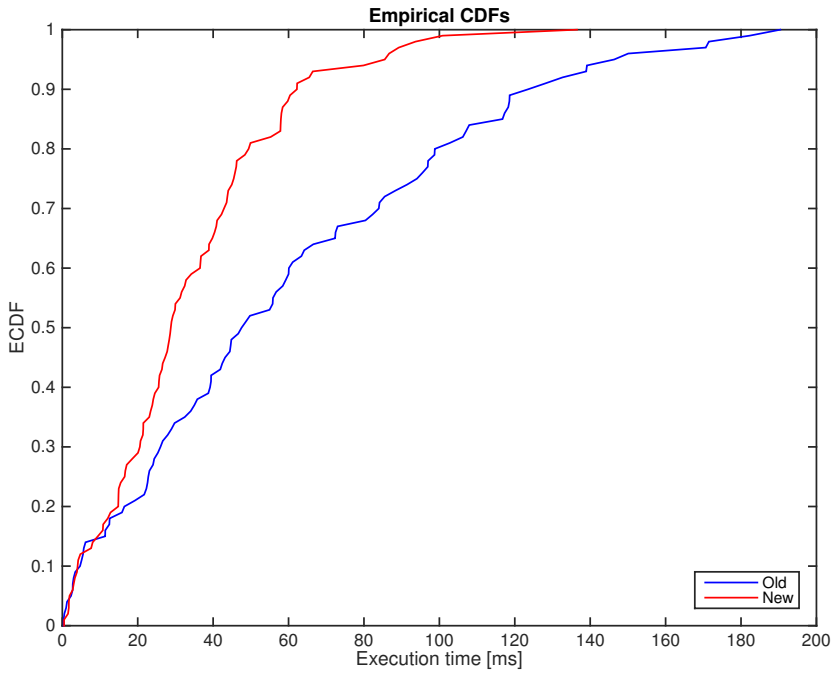

Figure 2: Figure 2.2 in [1]. Empirical CDF for the 2 datasets.

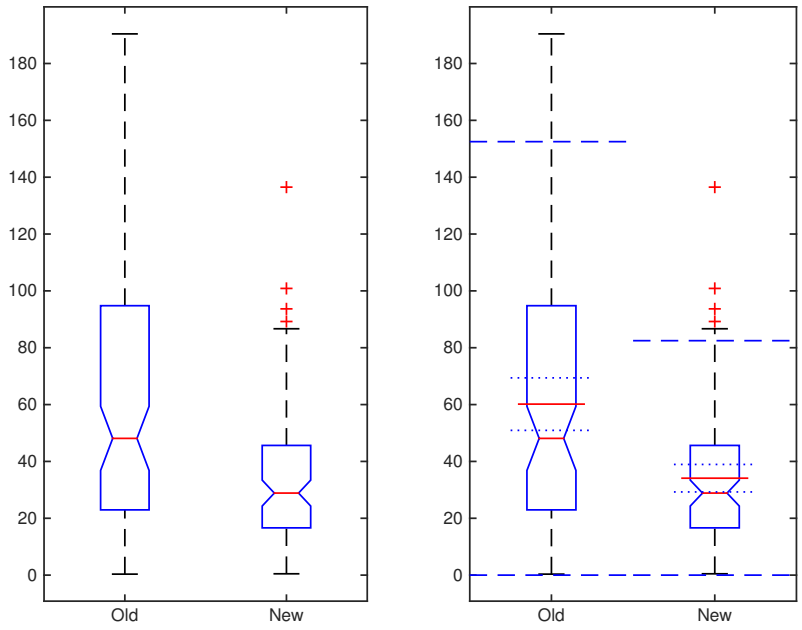

Figure 3: Figure 2.3 in [1]. Boxplots.

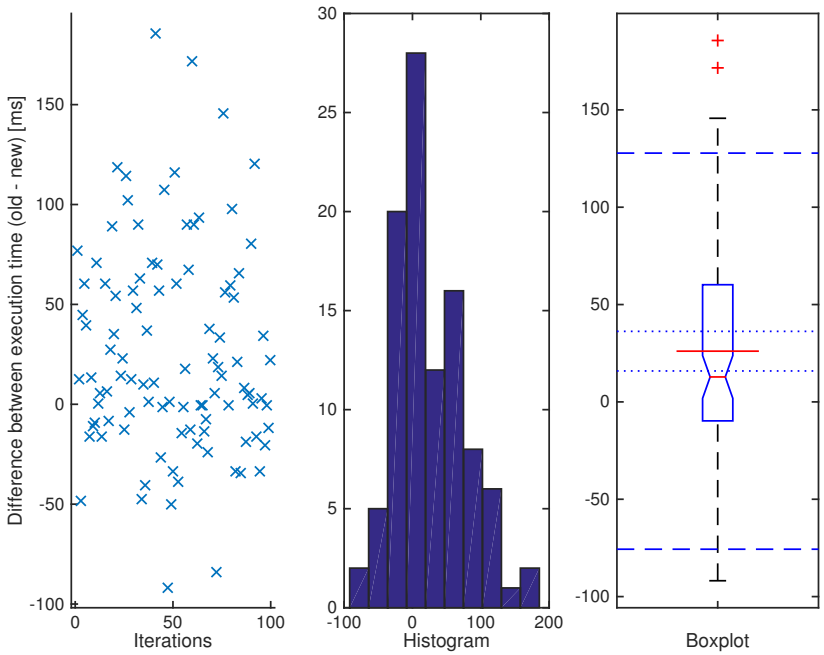

Figure 4: Figure 2.7 in [1]. Difference between the 2 datasets.

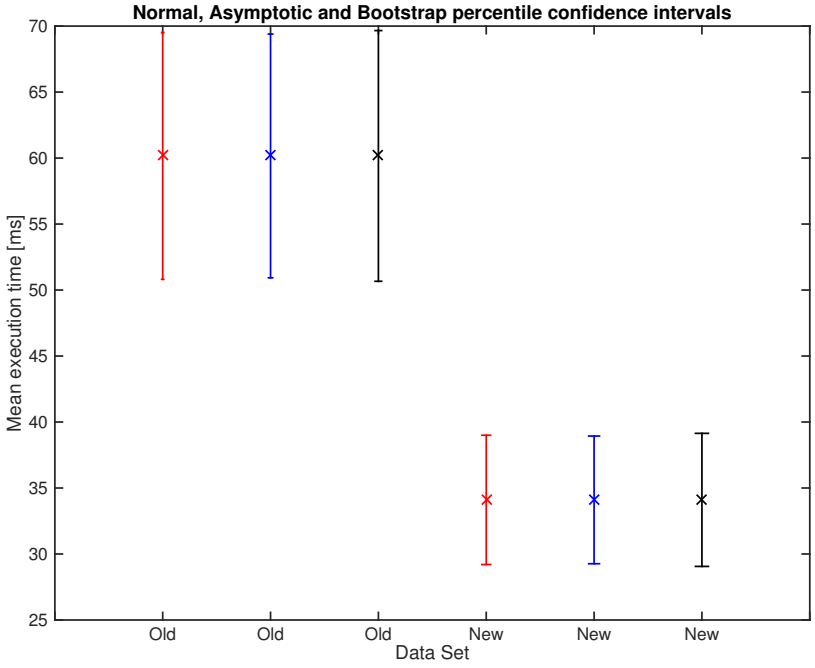

Figure 5: Figure 2.8 in [1]. Different ways to compute confidence intervals for the mean, from left to right, for each dataset: with the assumption that dataset is normal, iid and big dataset, bootstrap method.

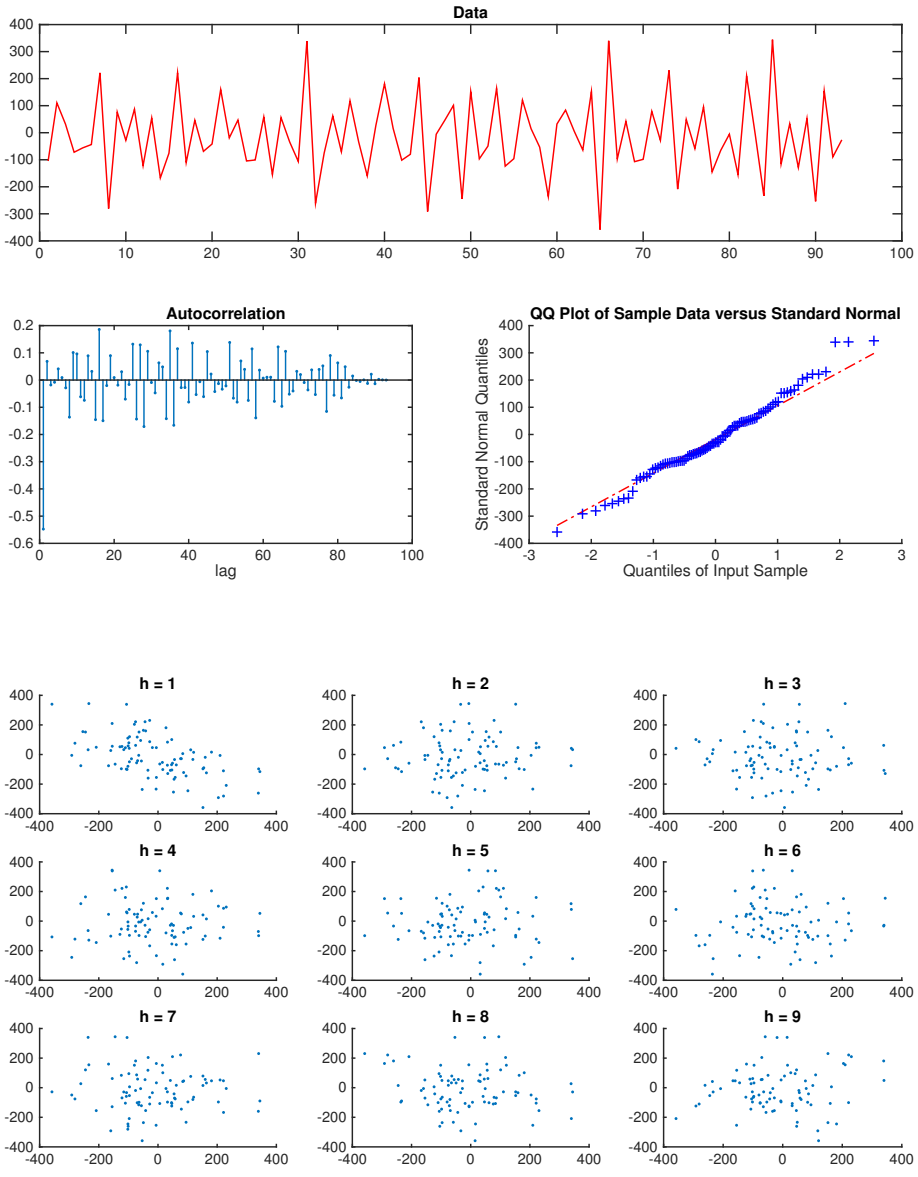

Figure 6: Figure 2.10 in [1]. Different ways to visualize iid and normality assumptions.

3 Exercise 2

Sample mean and standard deviation are estimated with formulas in Section 1 ($\hat{s}_n = \sqrt{\hat{s}_n^2}$). Confidence interval for the mean of the dataset $\{x_1, ..., x_n\}$ is computed as in Theorem 2.2 of [1], which can be applied if the dataset is composed of iid samples, the underlying distribution has a finite variance and the number of samples is large. Under these assumptions, given $\hat{\mu}_n$ and $\hat{s}_n$ of the dataset, then the confidence interval is $\hat{\mu}_n \pm \eta \frac{\hat{s}_n}{\sqrt{n}}$ with η such that $N_{0,1}(\eta) = \frac{1+\gamma}{2}$.
Actually $n = 48$ samples are not too many. However the theorem provides a good approximation of the confidence interval for the mean. Let's consider the following experiment: draw for 1000 times, independently, 48 iid $U[0, 1]$ samples at each time and compute the confidence interval at 95% level as above ($\eta = 1.96$). In Figure 7 there's the plot of the 1000 confidence intervals ordered by increasing lower bound. The number of confidence intervals that don't contain the true mean is 49, this means that in $49/1000 \approx 0.05\%$ of the cases the true mean is outside the confidence interval. This respects the definition of confidence interval at 95% with a little approximation due to the fact that each dataset isn't too big and data is not normal.
If instead the experiment is carried out using 48 iid random variables distributed according to a $N[0, 1]$, the confidence interval is given by an exact result from Theorem 2.3 in [1]. The number of confidence intervals that don't contain the true mean in 1000 trials is 43 (0.043%), as it can be seen in Figure 8.
However if we increase the number of 48-samples dataset generated to 10^6 then it is possible to see that the percentage of intervals that doesn't contain the true mean grows to expected results, which are 5.378% for U[0,1] (it is still an approximation with few non normal samples) and 4.9978% for N[0,1] ($\approx 5\%$ according to the definition).

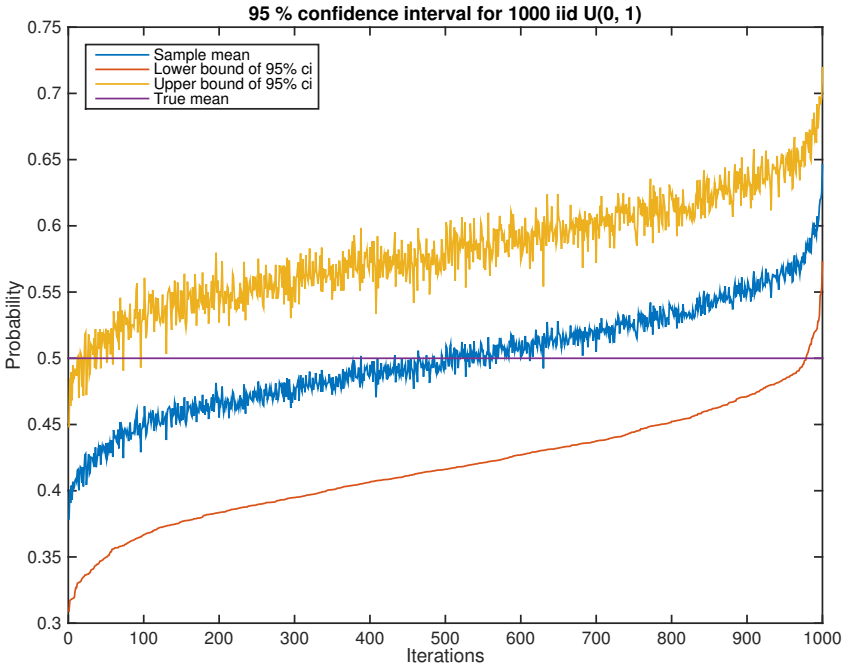

Figure 7: 1000 confidence intervals, each computed for 48 iid rv U[0,1]

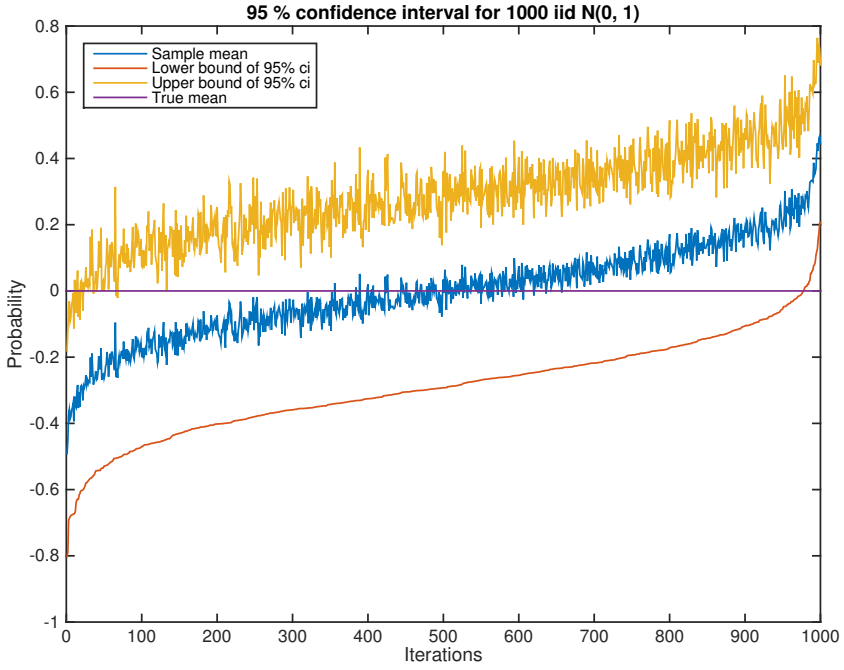

Figure 8: 1000 confidence intervals, each computed for 48 iid rv N[0,1]

4 Exercise 3

Consider a dataset $\{X_1, ..., X_n\}$ and its order statistic $X^n_{(1)}, ..., X^n_{(n)}$. $X^n_{(1)}$ is the lowest value in $\{X_1, ..., X_n\}$, $X^n_{(i)}$ is the $i-th$ lowest value in $\{X_1, ..., X_n\}$ and so on. If the $i-th$ element in the order statistic is equal to y, then there are $i-1$ samples lower or equal than y (each with probability $P[X \leq y] = F_X(y)$) and $n-i$ samples greater than y (each with probability $P[X > y] = 1 - F_X(y)$). This can be described with a multinomial distribution (see [2]):

$$P[X^n_{(i)} = y] = f_{X^n_{(i)}}(y) = \frac{n!}{(i-1)!1!(n-i)!}(P[X \leq y]^{(i-1)})P[X = y](P[X > y]^{(n-i)}) =$$

$$= \frac{n!}{(i-1)!(n-i)!}F_X(y)^{(i-1)}f_X(y)[1 - F_X(y)]^{(n-i)} \quad (1)$$

where the first term is a coefficient of the multinomial distribution, the second is the probability that $i-1$ samples are lower or equal than y, the third is the density of the i-th sample itself, and eventually the fourth is the probability that $n-i$ samples are greater than y.
If X is $U[0,1]$ then $f_X(y) = 1$ and $F_X(y) = y$ with $y \in [0,1]$ so Equation 1 becomes

$$f_{X^n_{(i)}}(y) = \frac{n!}{(i-1)!(n-i)!}y^{i-1}(1-y)^{n-i} \quad (2)$$

which is a beta distribution with $\alpha = i$ and $\beta = n - i + 1$. So $E[X^n_{(i)}] = \frac{i}{n+1}$ as reported in [2].

5 Exercise 4

This experiment is designed in order to study how does the accuracy of estimates of mean and variance improve by increasing the size of the dataset on which they are computed, both in the $U[0,1]$ and $N[0,1]$ cases. For the sample mean of a dataset $\{x_1, ..., x_n\}$ the variance of the estimator is $\frac{\hat{s}^2_n}{\sqrt{n}}$ so we should expect that the accuracy increases as n gets bigger.

This behavior can be observed in Figures 9 and 10, with the plot of sample mean and sample variance as functions of the dataset size n, computed with estimators of Section 1. For small n the estimate is bad, while it tends to the true value of the mean and standard deviation for higher values of n.

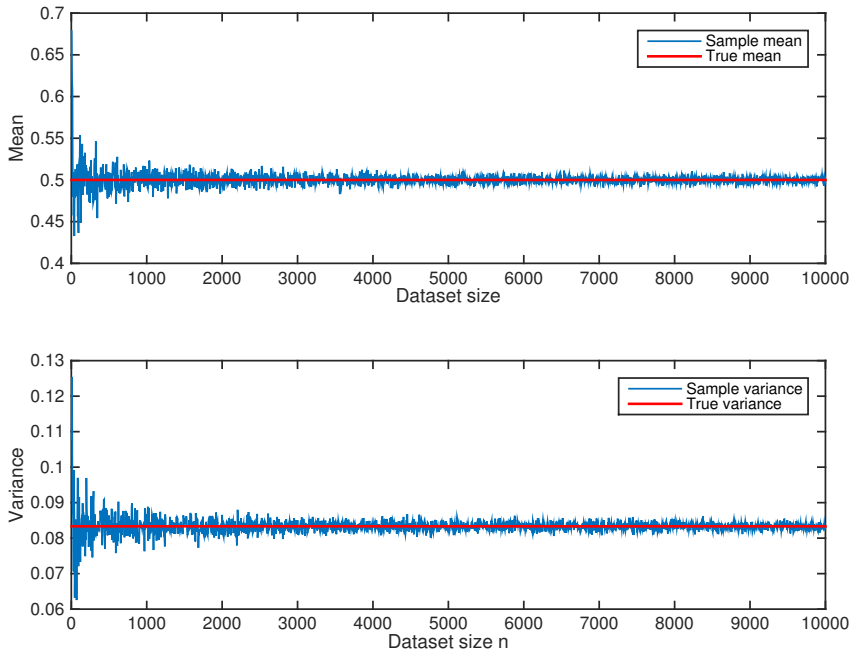

Figure 9: Sample mean and sample variance for $U[0,1]$ as a function of dataset size n

Confidence Intervals for variance

If dataset isn't normal there is not a theoretical result that can be used to exactly compute confidence intervals for variance. The only way is to use the bootstrap method, which involves a simulation and the usage of 2 percentiles as confidence intervals, as it can be seen in the following code.

```
function [ ci_boot ] = bootstrap_var( data, gamma, r0 )
% Bootstrap algorithm for ci for the mean
% gamma is the confidence level
% r0 is a level of accuracy of the algorithm, tipically r0 = 50 for gamma = 0.95

R = ceil(2*r0/(1-gamma)) - 1;
var_R = zeros(R, 1);
for r = 1:R
    x_r = datasample(data, length(data)); % draw n number with replacement
    var_R(r) = var_est(x_r, 0);
end
var_R = sort(var_R);
ci_boot = [var_R(r0), var_R(R+1-r0)];

end
```

The extraction with replacement of the samples from the dataset could be performed using a uniform random variable in the interval $[0, n]$ with n the size of the dataset, however MATLAB provides function `datasample` with the same functionality.
The confidence interval at 95% level gets closer to the value of the true variance as n increases as it can be seen in 11, this means that the estimate is more accurate in the 95% of the cases in which the true variance is included in the confidence interval.

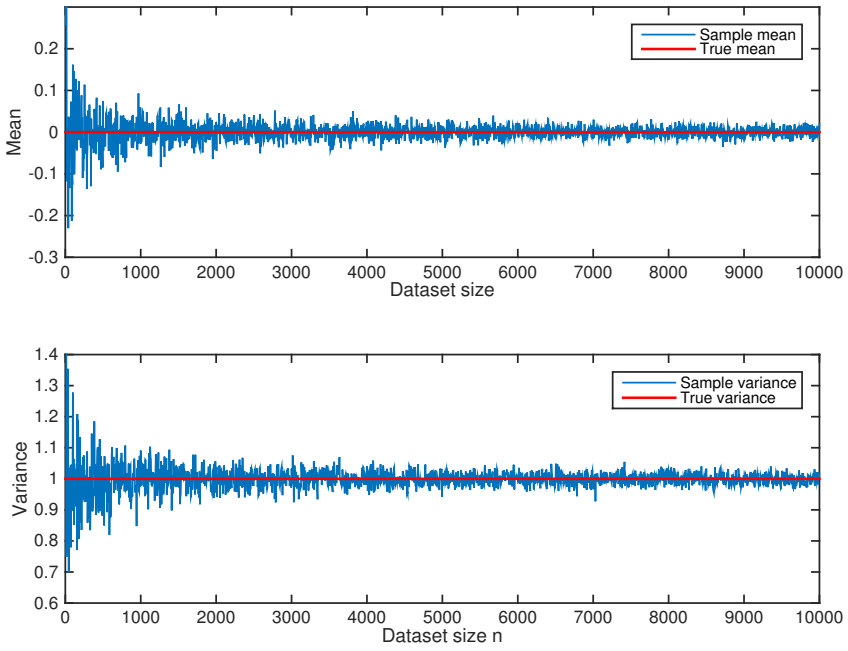

Figure 10: Sample mean and sample variance for $N[0,1]$ as a function of dataset size n

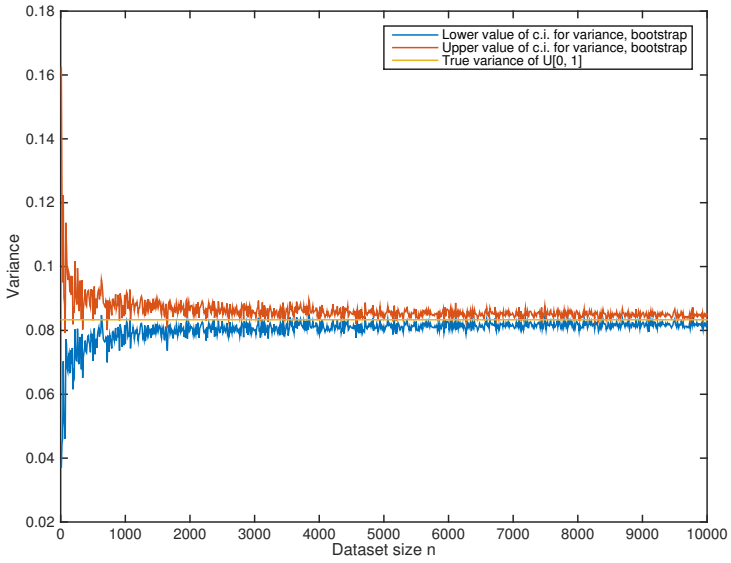

Figure 11: Confidence intervals for the variance of $U[0,1]$ as a function of dataset size n

If the data is normal, instead, the distribution of $(n-1)\frac{\hat{s}_n^2}{\sigma^2}$ is χ^2_{n-1} and confidence interval for level γ can be computed exactly as $\left[\hat{s}_n\sqrt{\frac{n-1}{\xi}}, \hat{s}_n\sqrt{\frac{n-1}{\zeta}}\right]$ with ξ, ζ such that $\chi^2_{n-1}(\xi) = \frac{1-\gamma}{2}$ and $\chi^2_{n-1}(\zeta) = \frac{1+\gamma}{2}$.
The bootstrapped and theoretically computed confidence intervals for a $N[0,1]$ are in Figures 12 and 13.

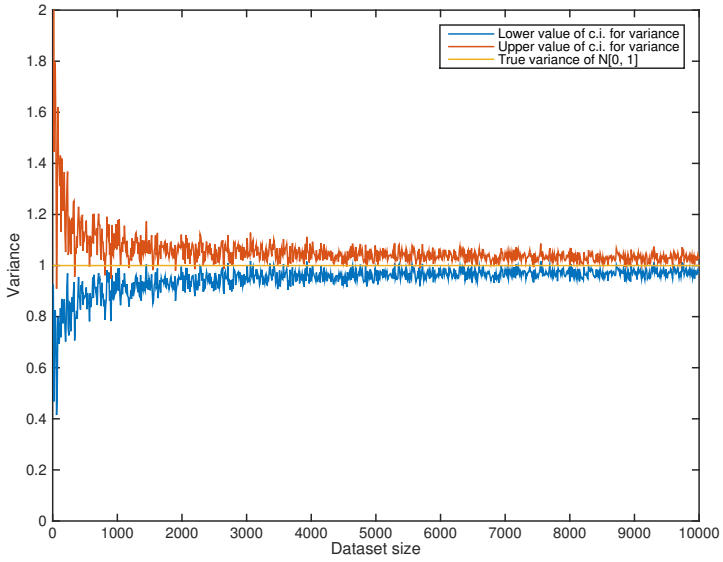

Figure 12: Bootstrapped confidence intervals for the variance of $N[0,1]$ as a function of dataset size n

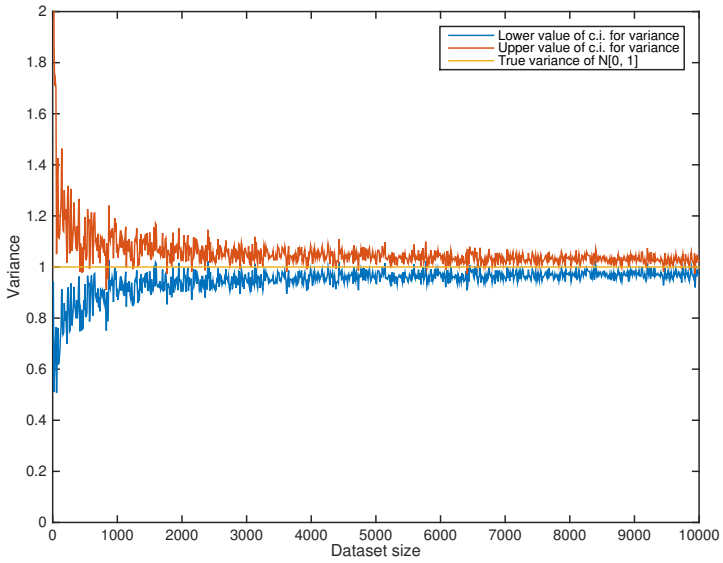

Figure 13: Theoretically computed confidence intervals for the variance of $N[0,1]$ as a function of dataset size n

Prediction Intervals

While a confidence interval describes the accuracy of an estimator, a prediction interval is the interval in which with a certain level of confidence γ the sample X_{n+1} of the sequence $\{X_1, ..., X_{n+1}\}$ can be found, given that $X_1, ..., X_n$ are known and X_{n+1} hasn't been observed yet.
For iid datasets there's a result which is based on order statistic, similar to the bootstrap, but exact. In particular Theorem 2.5 in [1] states that given the order statistic $X^n_{(1)}, ..., X^n_{(n)}$, for $1 \leq j \leq k \leq n$, $P(X^n_{(j)} \leq X_{n+1} \leq X^n_{(k)}) = \frac{k-j}{n+1}$. For $1 - \gamma = \alpha \geq \frac{2}{n+1}$ then the prediction interval is $[X_{(j)}, X_{(k)}]$ with $j = \lfloor (n+1)\frac{\alpha}{2} \rfloor$ and $k = \lceil (n+1)(1-\frac{\alpha}{2}) \rceil$.
Figure 14 plots $\alpha = 0.05$ (for a 95% prediction level) against $\frac{2}{n+1}$ and it can be seen that for $n \geq 39$ it is possible to compute exactly the prediction interval, which can be found in Figure 15. It is compared with the dataset size n.

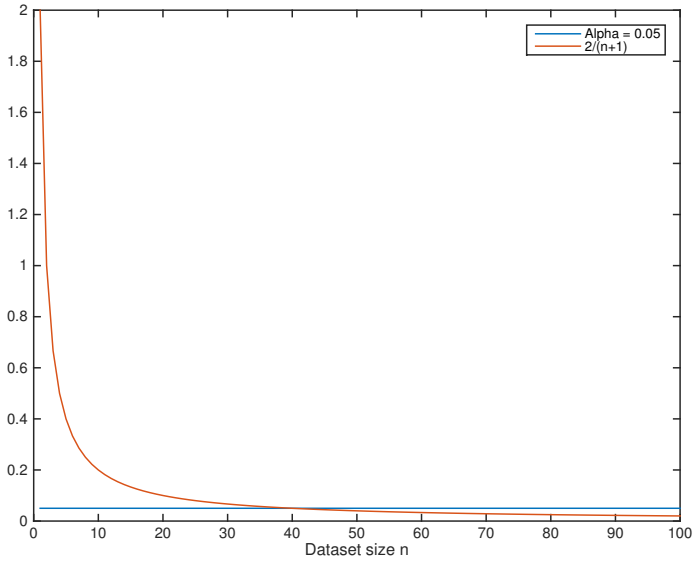

Figure 14: Allowed dataset size n for 95% prediction interval computation with order statistic

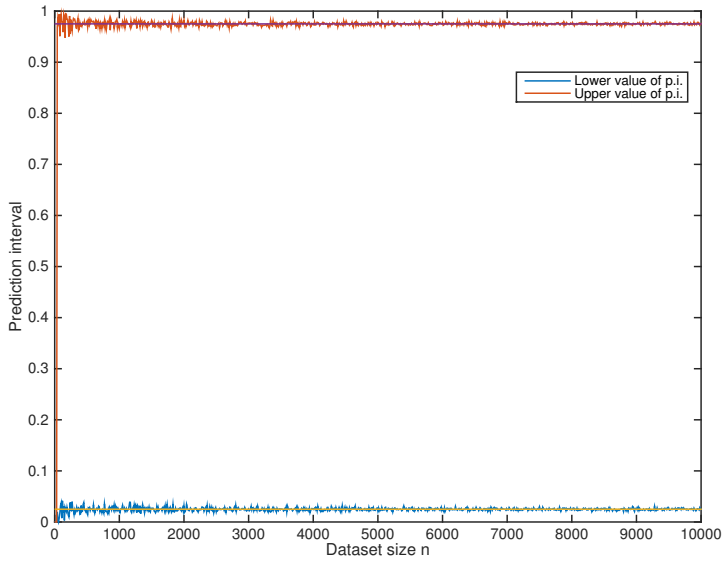

Figure 15: 95% prediction interval for a U[0,1] dataset as a function of n

The prediction interval can be computed also with an alternative version of the bootstrap method, described in the code below and based once again on Theorem 2.5 in [1]. However this is an approximation of the prediction interval, because Theorem 2.5 gives an exact result on the original dataset, while the bootstrap computes the same result on a resampling of the dataset. The result is in Figure 16. In Figure 17 there's a comparison between the two methods for small values of n.

```
function [ pi_boot ] = bootstrap_pi( data, gamma, r0 )
% Bootstrap algorithm for prediction intervals

alpha = 1 - gamma;
R = ceil(2*r0/(1-gamma)) - 1;
pi = zeros(R, 2);
n = length(data);
for r = 1:R
    x_r = datasample(data, length(data)); % draw n number with replacement
    x_r = sort(x_r);
    pi(r, 1) = x_r(floor((n+1)*alpha/2));
    pi(r, 2) = x_r(ceil((n+1)*(1 - alpha/2)));

end
pi_boot = [mean_est(pi(:, 1)), mean_est(pi(:, 2))];

end
```

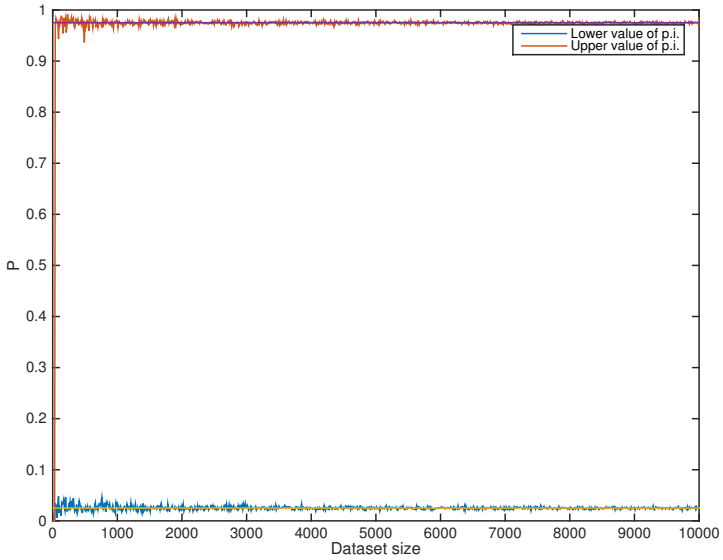

Figure 16: 95% prediction interval for a U[0,1] dataset as a function of n with bootstrap method

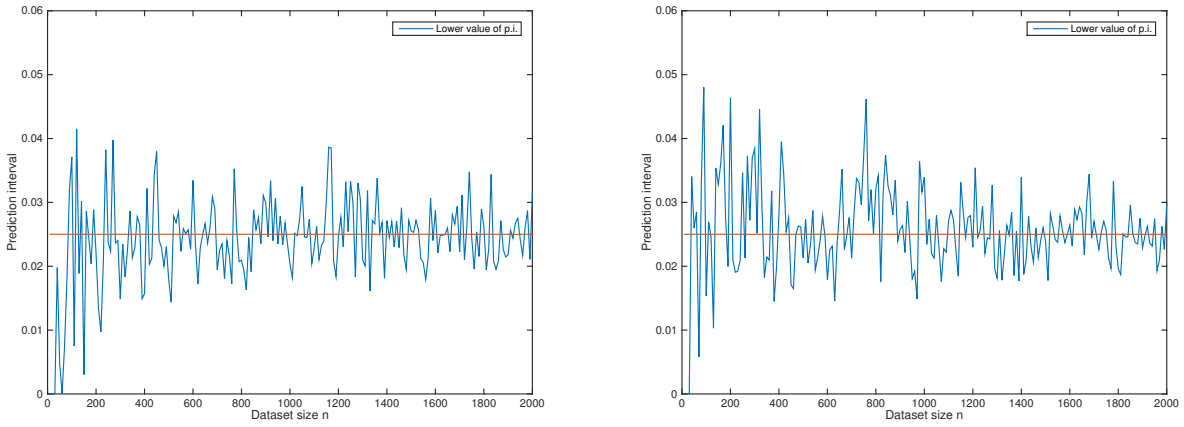

Figure 17: Lower value of prediction interval for small n computed with Theorem 2.5 and bootstrap method

For iid normal $N[0,1]$ datasets instead it is possible to use another result, described in Theorem 2.6 in [1]. The 95% prediction interval for normal dataset $\{x_1, ..., x_n\}$ is $\hat{\mu}_n \pm \eta \hat{s}_n \sqrt{1 + \frac{1}{n}}$ with η the $(1 - \frac{\alpha}{2})$ quantile of t_{n-1} student distribution.
In Figures 18, 19 and 20 there's a comparison between the 95% prediction interval computed for a dataset of size n with the general method, with the method for normal datasets and with bootstrap. As it can be seen in Figure 21 the second one presents better results for small n, since it is based on the gaussianity of the dataset, while for larger n the 3 are approximately the same. Note that formula provided by Theorem 2.6 has a weak dependence on the dataset size n, and for large n it is approximated by $\hat{\mu}_n \pm \eta \hat{s}_n$ with η such that $N_{0,1}(\eta) = \frac{1+\gamma}{2}$. Note also that the formula recalls the approximation for confidence interval for the mean of Theorem 2.2 in [1], without the $\sqrt{n}$ factor that divides the standard deviation, so it doesn't change a n gets larger. The grassy behavior which is present in Figure 19 for small n is due to the less accurate estimate of sample mean and sample standard deviation used to perform the calculation.

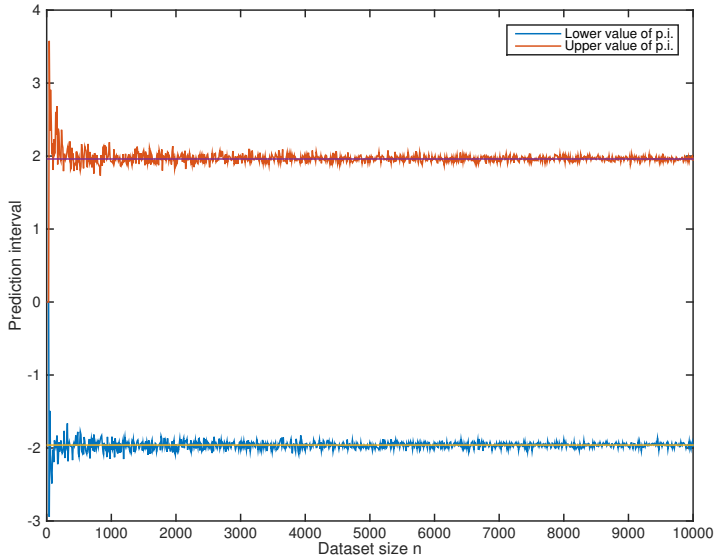

Figure 18: 95% prediction interval for a N[0,1] dataset as a function of n, from Theorem 2.5 in [1]

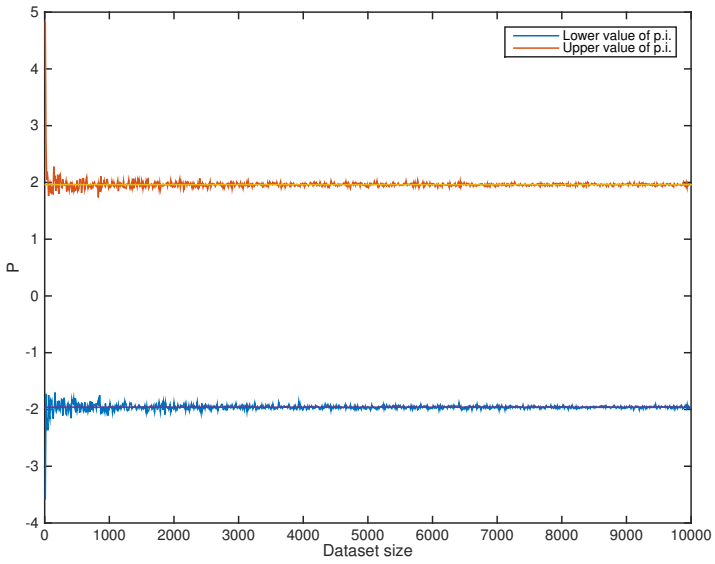

Figure 19: 95% prediction interval for a N[0,1] dataset as a function of n, from Theorem 2.6 in [1]

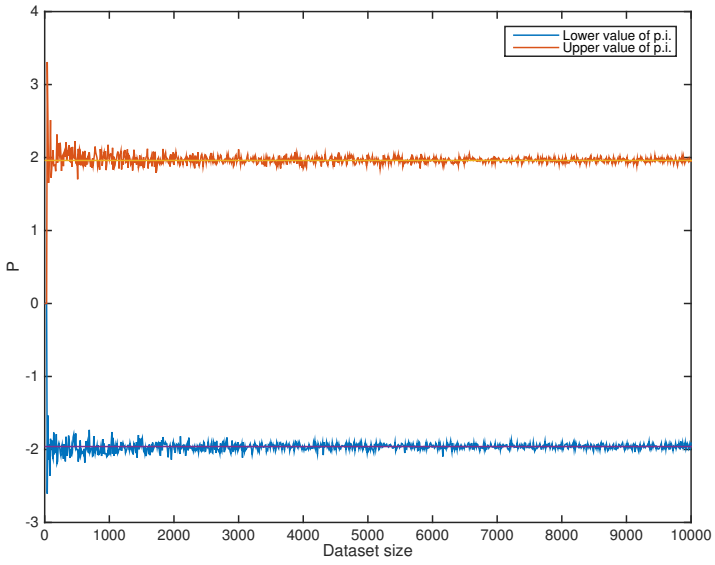

Figure 20: 95% prediction interval for a N[0,1] dataset as a function of n with bootstrap method

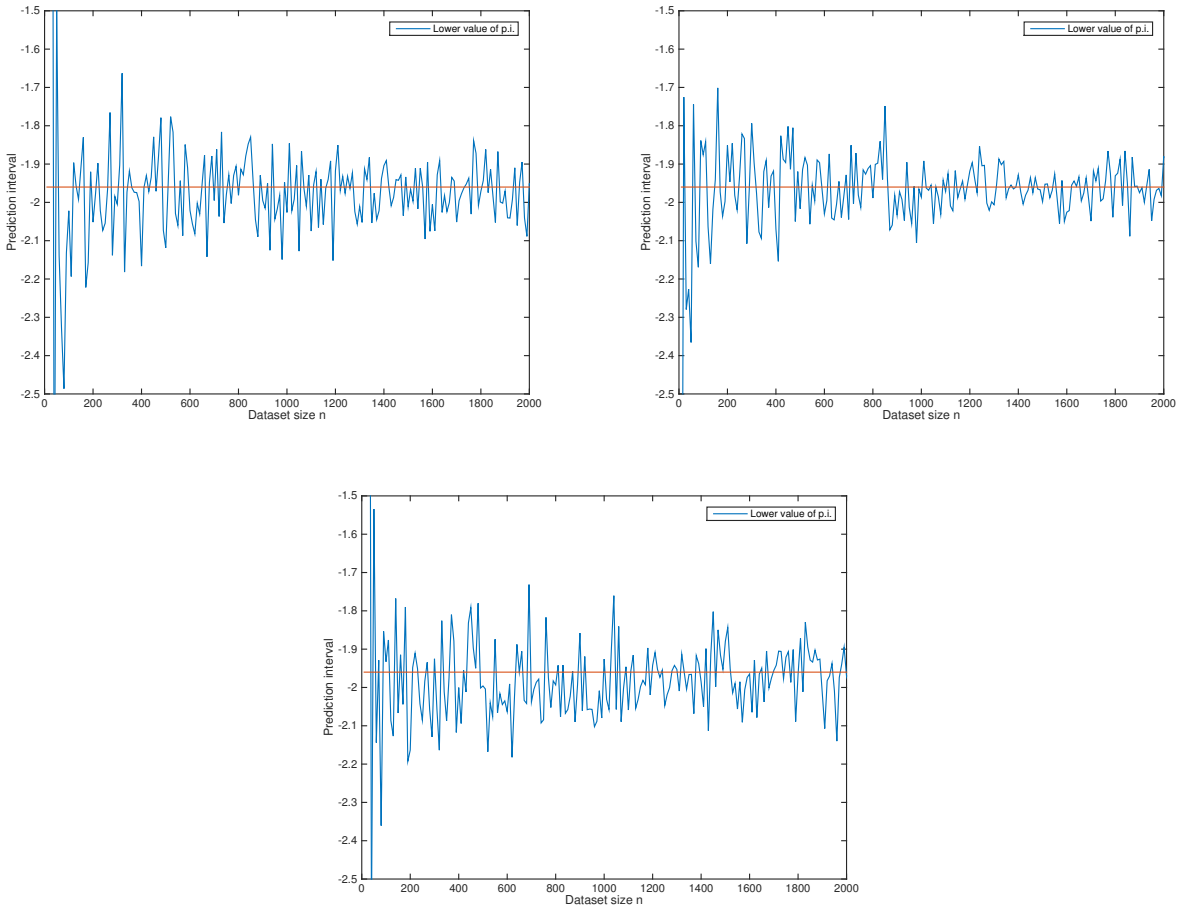

Figure 21: Lower value of prediction interval for small n computed with Theorem 2.5, the exact result for normal distributions and bootstrap method

References

[1] Y. Le Boudec, Performance Evaluation of Computer and Communications Systems, EPFL, 2015

[2] M. Pinsky, S. Karlin, An Introduction to Stochastic Modeling, 4^{th} edition, Elsevier, 2011