Measurement and Analysis of Ping-Pong Time

A RStudio Demonstration and Statistical Analysis Draft

Jean-Marc Vincent, Lucas Mello Schnorr

October 2017

This document is used to do a demonstration of Rstudio. It has an introduction, followed by a step-by-step receipt to conduct an analysis following the principles of literate programming.

1 Introduction

An experimental protocol is the description of the experience that has been conducted. It explains how the measurements have been implemented. For this Rstudio demonstration, we have measured the ping-pong time. It corresponds to the time taken by a message of a given size to go from one machine to another and back to the original machine. The following C-based code represents how the ping-pong time is measured, for a given `size` message in an environment with two processes (the `sender` and the `receiver`). We can see that the measured time is the one observed by the sender.

```c
if (sender) {
  atime = get_time();
  send_message(size);
  btime = get_time();
  recv_message(size);
  ctime = get_time();
  //Print (btime - atime) + (ctime - btime)
}else{
  recv_message(size);
  send_message(size);
}
```

Network measurements can be affected by a number of external factors. The most easily observed ones are the operating system complexity, its network layer, and also the wired or wireless hardware. All these factors increase the measurement variability, which should be quantified and taken into account during the analysis. As a consequence, the experiment has been replicated many times (by injecting the code above in a for loop). The computer program generates a textual file in the CSV (Comma-Separated Values) format. Each line of this CSV file contains one measurement, as the output of the Print command above.

2 Synthetic Ping-Pong Measurements (Demonstration)

We use synthetic measurements created as detailed below for this demonstration. For such reason, skip this section is you are playing with the given real measurements.

2.1 Create data

To create the synthetic ping-pong measurements, the code below is executed to generate 50 measurements considering an average of 4 and standard deviation of 1.5. The resulting values are kept in the `times` vector, which is written as a CSV file with the name `pp-synthetic.csv`.

```r
set.seed(42);
times <- rnorm(n = 50, mean = 4, sd = 1.5);
write.csv(times, "pp-synthetic.csv", row.names = FALSE);
```

The `set.seed` call is here only to keep the synthetic analysis fully reproducible.

2.2 Read data

To read the synthetic ping-pong measurements, the code below reads the file `pp-synthetic.csv` and creates the data frame `df`. This data frame has a single column called `time` which contains all measurements that have been previously generated. The call to `head(df)` lists the first rows of the data frame.

```
df <- read.csv("pp-synthetic.csv", header=TRUE, col.names=c("time"))
head(df);
```

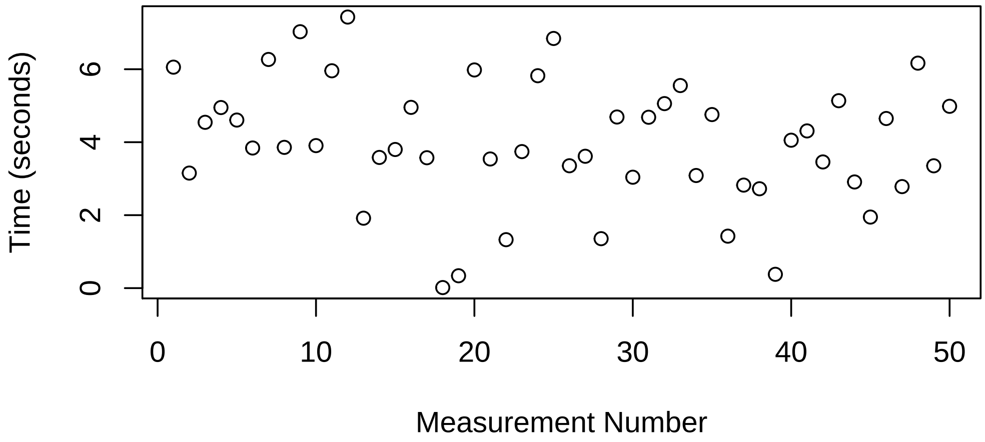
```
##       time
## 1 6.056438
## 2 3.152953
## 3 4.544693
## 4 4.949294
## 5 4.606402
## 6 3.840813
```

2.3 Overview of all the measurements

The best way to get an overview is to plot the data (see below). Unaware of the true mean, define the following metrics, providing a discussion:

- Estimate the ping-pong measurement (mean, median)
- Quantify the variability (sd)

```
plot(df$time, ylab="Time (seconds)", xlab="Measurement Number");
```



2.4 Statistical Summary

A summary of a set of values can be easily obtained in R with (`df$time` is a vector with the measurements – it represents the column `time` of the data frame `df`). So, calling the function `summary` on this vector gives:

```
summary(df$time);
```

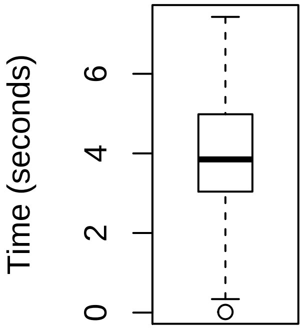
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01532 3.05166 3.84941 3.94649 4.97609 7.42997
```

We can see that the mean estimation is `3.946`, while the median gives `3.849`. Other values are the minimum, the first quartile, the third quartile and the maximum number.

2.5 Boxplot

A much nicer way to represent the metrics of the summary above is using a boxplot. It takes all the values and create a visual representation. Can you explain what represents each element of the boxplot? How it helps you to better understand the data?

```
boxplot(df$time, ylab="Time (seconds)")
```

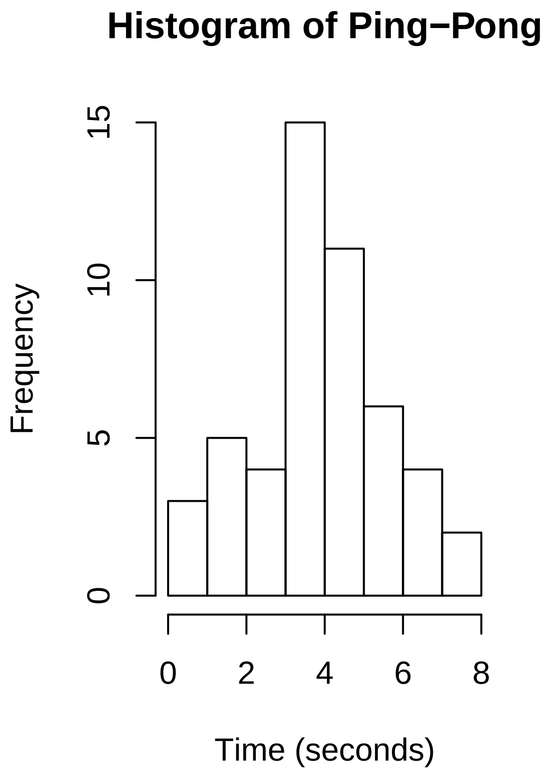


- My answer: I can see a dotted vertical line from the Min to the Max values, the median as a horizontal line in the middle of the boxplot, and the horizontal edges aroun the median are the first and the third quartile. This representation is useful to understand the distribution of the time values.

2.6 Histogram

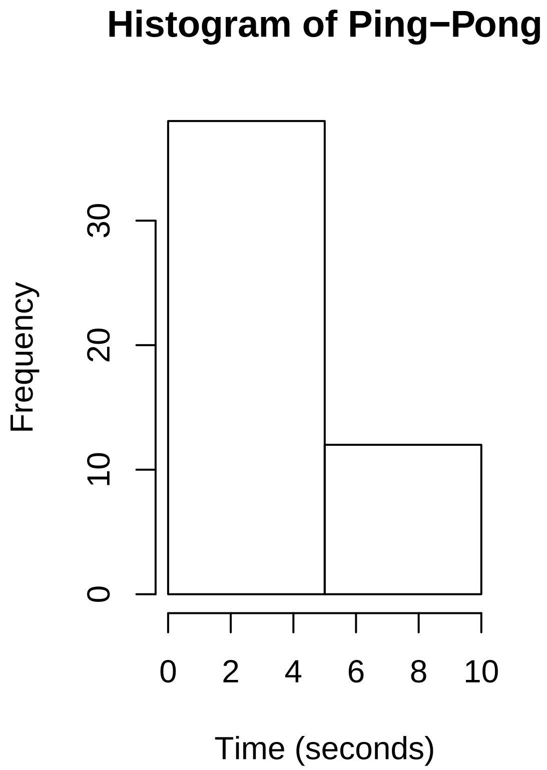
A histogram counts how many times the observed value falls into different bins. For time measurements, a bin is represented by an interval of values, for instance (2,3). Whenever the measurement falls into a bin, a counter is incremented. Below we can see the histogram for the ping-pong synthetic measurement. We have used 7 bins (as explicited by the `breaks` parameter).

```
hist(df$time, breaks=7, xlab="Time (seconds)", main="Histogram of Ping-Pong")
```



The problem of histograms is that by changing the number of bins, the perception based on the plot can radically change. Let's for instance force the number of bins to 2:

```
hist(df$time, breaks=2, xlab="Time (seconds)", main="Histogram of Ping-Pong")
```



We can see that two bins hardly represents the reality.

2.7 Variability

As we previously mentioned, ping-pong measurements suffer external influences that might affect estimations. As a consequence, we need to evaluate the variability. A common measure of variability is the standard deviation, calculated like this:

```
sd(df$time)
```

```
## [1] 1.727215
```

2.8 Wrap up

We have synthetically generated measurements for ping-pong, by using the `rnorm` function of R. We asked for a true mean of 4, with a standard deviation of 1.5. so, unaware of such values (using only statistics), we can estimate the mean and standard deviation as:

```
mean(df$time)
```

```
## [1] 3.946492
```

```
sd(df$time)
```

```
## [1] 1.727215
```

Which are slightly different from the true values.

3 Real Ping-Pong Measurements (TD)

The measurements detailed in the introduction have been obtained for a number of different message sizes. We provide five of them here, with the following names:

```
ls data/PP_size*.csv
```

```
## data/PP_size_1.csv
## data/PP_size_2.csv
## data/PP_size_3.csv
## data/PP_size_4.csv
## data/PP_size_5.csv
```

From 1 to 5 bytes. You should repeat all the steps of previous section using at least one of these files. Write your stastitical interpretation of the measurements. Here's how you can read one of them.

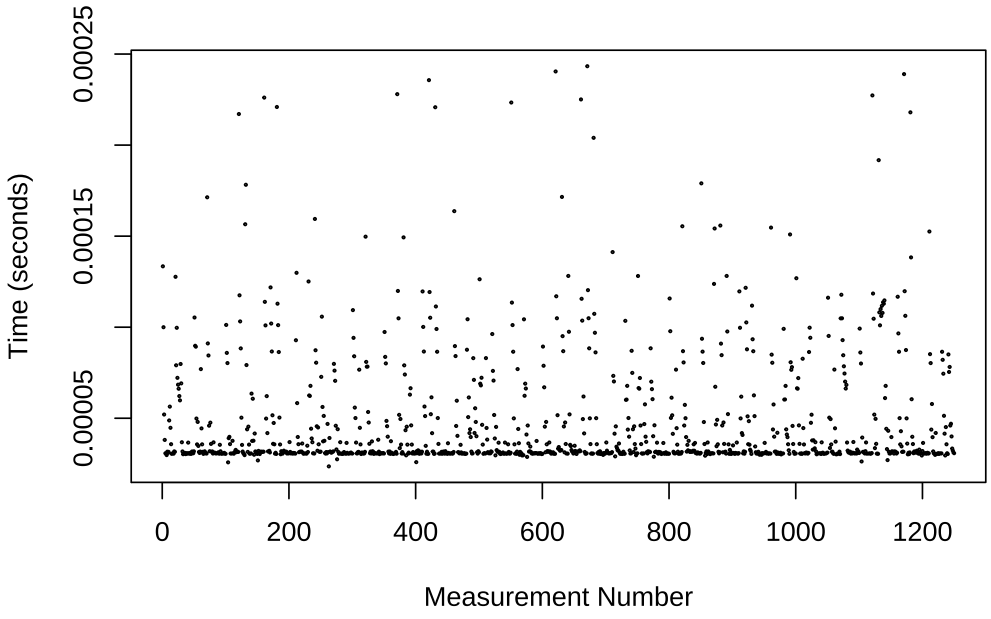
```
df1 <- read.csv("data/PP_size_1.csv");
head(df1);
```

```
##          time
## 1 0.000133434
## 2 0.000099994
## 3 0.000052050
## 4 0.000038092
## 5 0.000030759
## 6 0.000030473
```

3.1 Statistical interpretation for Real Ping-Pong Measurements (PP_size_1.csv data)

- Plot of all data

```
plot(df1$time, ylab="Time (seconds)", xlab="Measurement Number", cex = .2);
```



An overview of the data shows that all the measurements are of less than one second.

- Statistical Summanry

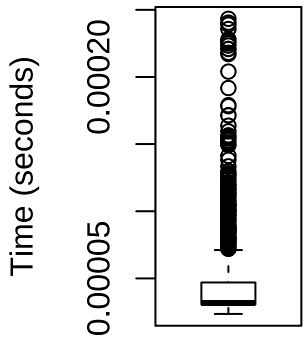
```
summary(df1$time);
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 2.358e-05 3.070e-05 3.164e-05 4.732e-05 4.696e-05 2.433e-04
```

The mean estimation is 4.732e-05, while the median gives 3.164e-05.

- Boxplot

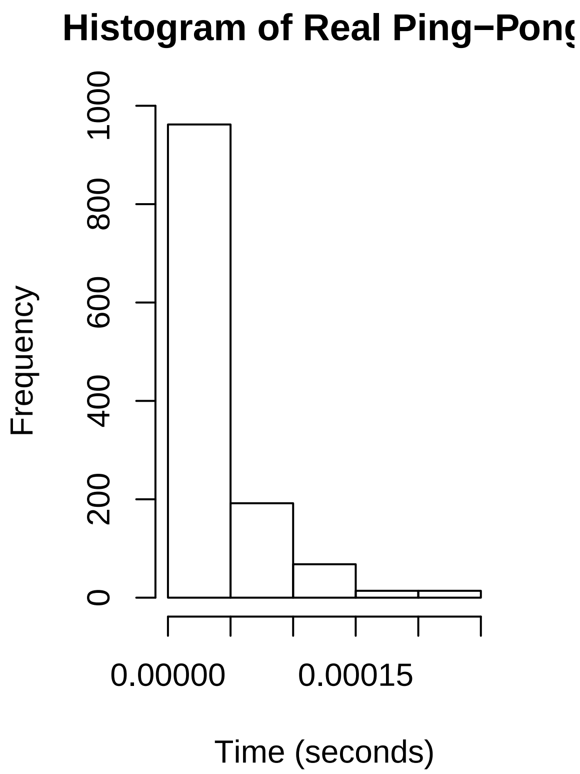
```
boxplot(df1$time, ylab="Time (seconds)")
```



The boxplot shows several outilers.

- Histogram

```
hist(df1$time, breaks=5, xlab="Time (seconds)", main="Histogram of Real Ping-Pong")
```



The histogram confirms the idea that most of points are really close to zero.

- Variability

```
sd(df1$time)
```

```
## [1] 3.286077e-05
```

```
mean(df1$time)
```

```
## [1] 4.732325e-05
```

```
sd(df1$time)
```

```
## [1] 3.286077e-05
```

4 Iteration Duration of a Geophysics Parallel Application

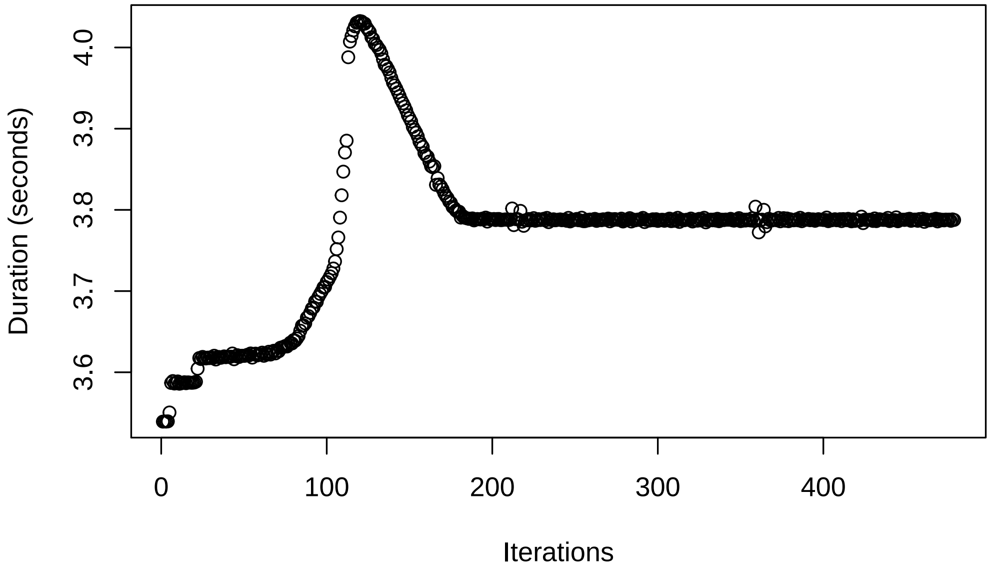
Geophysics parallel applications are generally organized in timesteps. At each timestep, the mathematical model is calculated taking into account the time interval. As the simulation evolves, one can measure how much time it takes to calculate the mathematical model for each iteration. Here we provide a data set that contains the duration of each iteration in a simulation composed of a little less than 500 iterations. Here's the code to read that data set:

```
df <- read.csv("data/IterDuration.csv");
head(df);
```

```
##   duration
## 1 3.539277
## 2 3.539381
## 3 3.539294
## 4 3.539549
## 5 3.550483
## 6 3.586875
```

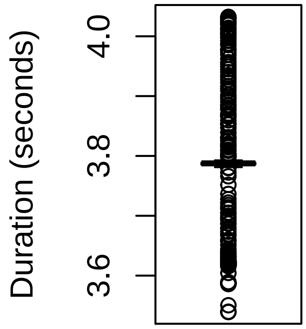
- Plot of Geophysics Parallel Application

```
plot(df$duration, ylab="Duration (seconds)", xlab="Iterations");
```



- Boxplot

```
boxplot(df$duration, ylab="Duration (seconds)")
```



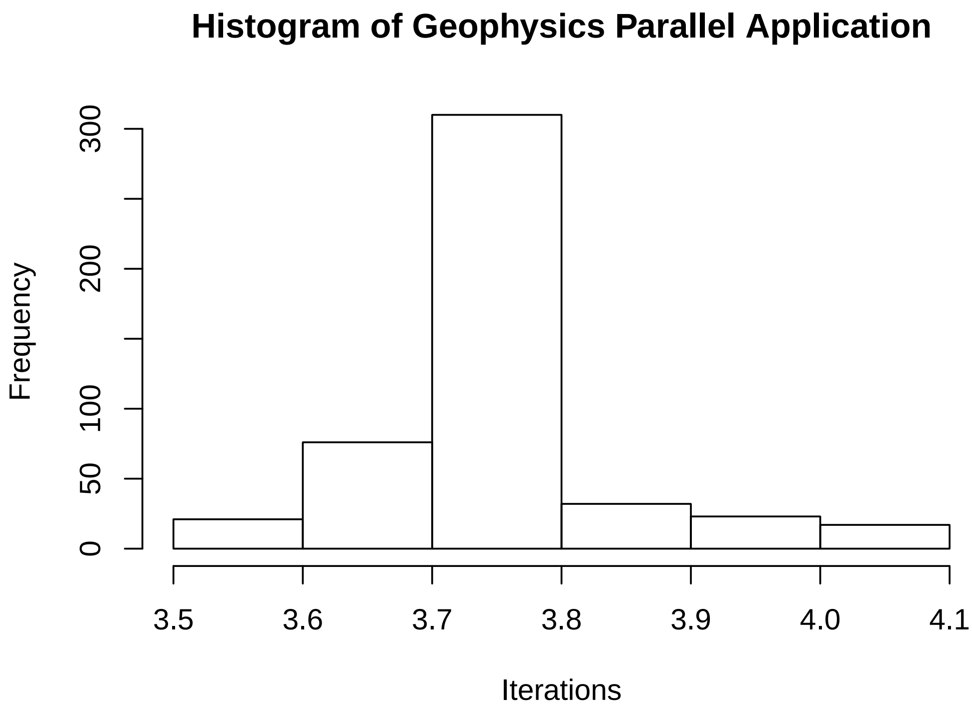
- Statistical Summanry

```
summary(df$duration);
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.539   3.786   3.788   3.773   3.789   4.032
```

- Histogram

```
hist(df$duration, breaks=5, xlab="Iterations", main="Histogram of Geophysics Parallel Applicatio
```


Histogram of Geophysics Parallel Application
Frequency
Iterations