

Quantitative tools: R and more



ROYAL INSTITUTE
OF TECHNOLOGY

prof. Gerald Q. Maguire Jr.
School of Information and Communication Technology (ICT)
KTH Royal Institute of Technology

<http://web.ict.kth.se/~maguire>

&

Prof. em. Marilyn E. Noz, Ph. D.
School of Medicine
New York University

II2202 Fall 2012

2012.09.11

© 2012 G. Q. Maguire Jr. All rights reserved.

Some statistical concepts

Independent versus dependant variables

Independent variable – a variable that you can change

Dependant variable:

- A response or outcome
- This is what you will **measure**

Types of data

- **Nominal data**: unordered groups, e.g., male/female, left-handed/right-handed, ...
- **Ordinal data**: rank ordered; the difference between item numbered n and $n+i$ does not tell you anything other than that one is ranked ahead of the other, e.g. Top 500 Universities, top 10 protocols in bytes, ...
- **Interval data**: continuous ranges mapped to some scale, without a clear zero
- **Ratio data**: like interval data but with a clear absolute zero value

Metrics

Type of data	Example Metrics	Common statistics
Nominal data	Success/failure	Frequencies, Chi-square
Ordinal data	Ranking	Frequencies, Chi-square, Wilcoxon rank sum tests, Spearman rank correlation
Interval data	Likert scale, System Useability Scale,	All descriptive statistics (average, median, std. dev., ...), Student's t-test, ANOVA, correlation, regression, ...
Ratio data	Task completion time, packet inter-arrival time, ...	All of the previous + geometric mean

Adapted from Table 2.3 on page 23 of [1]

Measures of Central Tendency

Three most common measures are:

Mean arithmetic average

Median mid point of the distribution
(half the values are larger and half are smaller)

Mode most common value

Selecting participants

- **Random** sampling
- **Systematic** sampling – e.g. every 3rd person
- **Stratified** sampling – based upon a representative subset
- **Samples of convenience**
 - Who can you get?
 - Are they representative of the target population?

Sample size

- What is the goal?
 - Is the difference expected to be large or small?
- What is an acceptable margin of error?

Within-subjects versus between-subjects

- Within-subjects
 - Also known as repeated-measures
 - The same subject, but repeated measurements
- Between-subjects
 - Comparing results of subject_i with subject_k
 - Avoids carry-over effects (where the subject learns from one trial and this causes a difference in subsequent trials)
- Mixed design

Counterbalancing

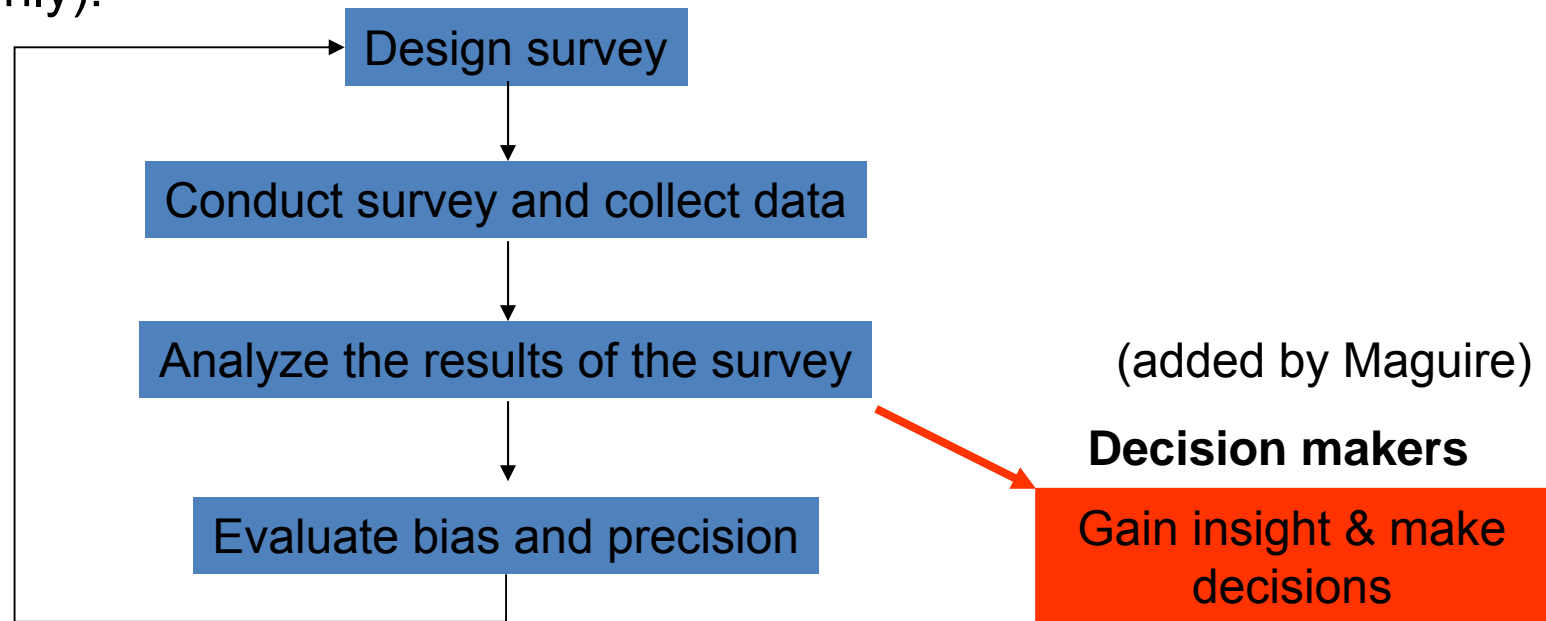
To avoid carryover effects vary the order of the tasks:

- Randomize order
- Sets of predefined orders – subject is randomly assigned to one of these sets

(Starting) Quantitative analysis of survey data

Overview

Gillian Raab, Professor of Applied Statistics at Napier University, shows the process of carrying out surveys as viewed by a statistician (roughly):



Adapted from the figure on his slide 7 in "Background to P|E|A|S project", 9 September 2004, <http://www2.napier.ac.uk/depts/fhls/peas/workshops/workshop1presentationGR.ppt>

Objective

- What is the object of the survey?
 - Finding a **predictive** model
 - Finding **hidden** relationships
 - Segmenting a population into **strata**
 - **Visualizing** responses
(e.g., Distance from a park versus frequency of visits to this park)
 - Making a **decision** (e.g., where to put a park)
- What is (are) the research question(s)?

Considerations when designing studies

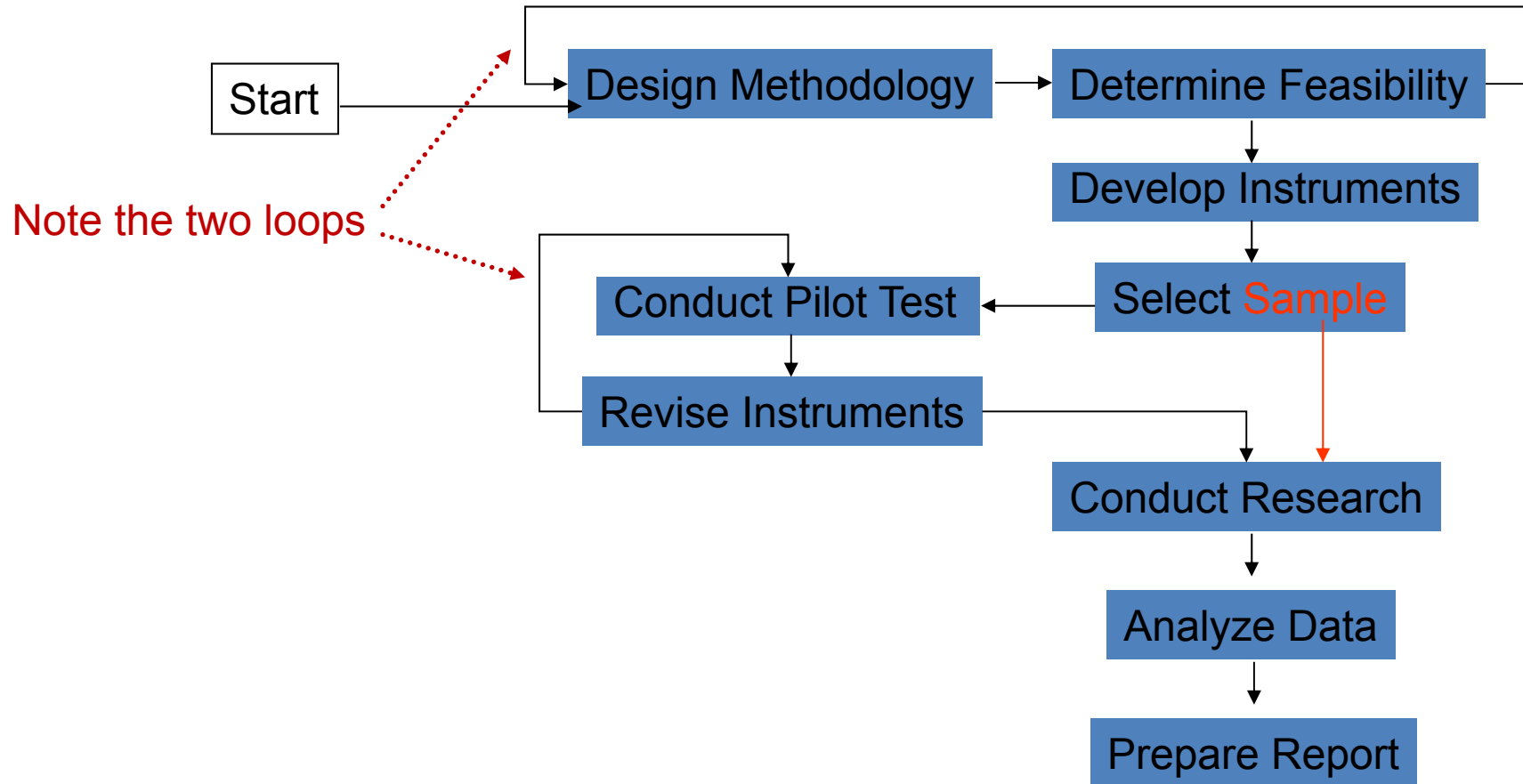
Ken Kelley and Scott E. Maxwell state:

“At a minimum, the following points must be considered when designing studies in the behavioral, educational, and social sciences:

- (a) the question(s) of interest must be determined;
- (b) the population of interest must be identified;
- (c) a sampling scheme must be devised;
- (d) selection of independent and dependent measures must occur;
- (e) a decision regarding experimentation versus observation must be made;
- (f) statistical methods must be chosen so that the question(s) of interest can be answered in an appropriate and optimal way;
- (g) sample size planning must occur so that an appropriate sample size given the particular scenario, as defined by points a through f, can be used;
- (h) the duration of the study and number of measurement occasions need to be considered;
- (i) the financial cost (and feasibility) of the proposed study calculated.”

Ken Kelley and Scott E. Maxwell, Sample Size Planning with Applications to Multiple Regression: Power and Accuracy for Omnibus and Targeted Effects, In P. Alasuuta, L. Bickman, & J. Brannen (Editors), Hand book of social research methods. Sage, Newbury Park, CA, USA, 2008, pp. 166-192
http://nd.edu/~kkelley/publications/chapters/Kelley_Maxwell_Chapter_SSMR_2008.pdf

Questionnaire Research Flow Chart



Adapted from pg. 3 of David S. Walonick, *A Selection from Survival Statistics*, StatPac, Inc. Bloomington, MN, USA, 14 August 2010, ISBN 0-918733-11-1, <http://www.statpac.com/surveys/>

Sampling methods

- *Probability*
 - Random sampling & systematic sampling (every Nth person) \Rightarrow equal probability of selection
 - Sampling proportional to size (PPS) – concentrates on the largest segments of the population
 - Stratified sampling (members of each *stratum* (a sub-population) share some characteristic)
 - Advantage: can calculate sampling error
- *Nonprobability*
 - Accidental, Haphazard, *convenience sampling* \Rightarrow these might not be representative of the target population
 - *Purposeful – sampling* with a purpose in mind
 - *Modal instance sampling* – focused on ‘typical’ case
 - *Expert sampling* – choosing experts for your samples
 - *Quota sampling* - proportional vs. non-proportional
 - *Heterogeneity sampling* – to achieve diversity in samples
 - *Snowball sampling* – get recommendations of other to sample, from your samples

For further details of Nonprobability sampling see: William M.K. Trochim, The Research Methods Knowledge Base, 2nd Edition, webpage: Nonprobability Sampling, Last Revised: 10/20/2006 <http://www.socialresearchmethods.net/kb/sampron.php>

Sample size

Choosing the size of your sample is related to your expected signal to noise ratio and your desired confidence.

Statisticians speak about **statistical power**, for details see <http://www.socialresearchmethods.net/kb/power.php>

See also:

Ken Kelley and Scott E. Maxwell, Sample Size Planning with Applications to Multiple Regression: Power and Accuracy for Omnibus and Targeted Effects, In P. Alasuuta, L. Bickman, & J. Brannen (Editors), Hand book of social research methods. Sage, Newbury Park, CA, USA, 2008, pp. 166-192

http://nd.edu/~kkelley/publications/chapters/Kelley_Maxwell_Chapter_SSMR_2008.pdf

S. E. Maxwell, K. Kelley, and J. R. Rausch. Sample size planning for statistical power and accuracy in parameter estimation. Annual Review of Psychology, 59, 2008, pages 537-563.

http://nd.edu/~kkelley/publications/articles/Maxwell_Kelley_Rausch_2008.pdf

K. Kelley and S.E. Maxwell, Sample size for multiple regression: Obtaining regression coefficients that are accurate, not simply significant. Psychological Methods, 8(3), 2003, pages 305-321.

http://nd.edu/~kkelley/publications/articles/Kelley_Maxwell_2003.pdf

K. Kelley, S.E. Maxwell, and J.R. Rausch, Obtaining power or obtaining precision: Delineating methods of sample-size planning. Evaluation and the Health Professions, 26(3), 2003, pages 258-287.

http://nd.edu/~kkelley/publications/articles/Kelley_Maxwell_Rausch_2003.pdf

K. Kelley, K. Lai, and P-J Wu. Using R for data analysis: A best practice for research. In J. Osbourne (Ed.), Best practices in quantitative methods, Sage, Newbury Park, CA, USA, 2008, pages 535-572.

http://nd.edu/~kkelley/publications/chapters/Kelley_Lai_Wu_Using_R_2008.pdf

Getting started with data analysis

Assuming that the survey has already be conducted and that the data has been entered into a computer system, what is the next step?

- **Preliminary analysis**
 - Descriptive statistics
- **Exploratory data analysis**
 - Plots (points, lines, scatterplots, ...), histograms, ...

Types of analysis

- **Design-based analysis**

- In this approach randomness is **induced** by the random selection of sample or the assignment of samples to a subset
- Choice of a statistical model can be used for model-based inference

- **Model-based analysis**

In this approach randomness is because of the **innate** randomness in the measurements (in the case of surveys – these are the responses)

Modeling techniques

- Prediction, classification (using neural networks, bayesian networks, trees, ...), regression
- Clustering, segmentation
- Fitting to an *apriori* model
- Factor analysis, principle components analysis

Weights

When we have samples, we need to make sure that these samples are representative of the total population – to do this we may need to establish weights

For details of weights see:

James R. Chromy and Savitri Abeyasekera, "Statistical analysis of survey data", Chapter XIX, In Household Sample Surveys in Developing and Transition Countries. United Nations, New York, NY, 2005.

<http://www.cpc.unc.edu/projects/addhealth/data/guides/weight1.pdf>

Significance

Significance is a statistical term indicating *your confidence* in your conclusion that a real difference exists or that a relationship actually exists, i.e., **that the result is unlikely to be due simply to chance.**

If your hypothesis states a direction of this difference – use a **One-Tailed** significance test, otherwise use a **Two-Tailed** significance test.

Note: Significant **does not** imply important, interesting, or meaningful!

Similarly not all observations that are not statistically significant are unimportant, uninteresting, ...

Testing for significance

1. Decide on your significance level α
2. Calculate your statistical value p
3. If $p < \alpha$, then the result is significant, else it is not significant

An alternative view is:

$$\text{confidence} = (\text{signal/noise}) * \sqrt{\text{sample size}}$$

For details of the above equation see: David L. Sackett, Why randomized controlled trials fail but needn't: 2. Failure to employ physiological statistics, or the only formula a clinician-trialist is ever likely to need (or understand!).

Canadian Medical Association Journal (CMAJ), 165(9):1226-37, 30 October 2001 PubMedID (PMID): 11706914

<http://www.cmaj.ca/cgi/content/full/165/9/1226>

See also: Understanding Hypothesis Testing: Example #1, Department of Statistics, West Virginia University, last modified 4 April 2000

<http://www.stat.wvu.edu/SRS/Modules/HypTest/exam1.html>

Next steps

1. Search the literature and read extensively
2. Consult a statistician to get help with your statistical analysis
(In most cases this is going to cost you money, but can save you a lot of time and effort.)
3. Doing some statistical analysis yourself

R

R is an open source successor to the statistics package S and Splus

S was developed by the statisticians at Bell Labs **to help them help others with their problems**

Josef Freuwald (a graduate student in Linguistics at the University of Pennsylvania) said: **“Quite simply, R is the statistics software paradigm of our day.”**

<http://www.ling.upenn.edu/~joseff/rstudy/week1.html#why>

And its free! Additionally, it supports Windows, Linux, and Mac OS

Commercial alternatives to R

- Microsoft's Excel – we will return to this in a later lecture
- MathWorks' MATLAB – Statistics Toolbox™
<http://www.mathworks.se/products/statistics/>
- Statistical Analysis with SAS/STAT® Software
<https://www.sas.com/technologies/analytics/statistics/stat/index.html>
- IBM® SPSS® Advanced Statistics
<http://www-01.ibm.com/software/analytics/spss/products/statistics/advanced-statistics/>
- Stata <http://stata.com/>
- ...

R Resources

Comprehensive R Archive Network (CRAN)

<http://cran.r-project.org/>

Lots of tutorials:

- <http://www.r-tutor.com/>
- <http://cran.r-project.org/doc/manuals/R-intro.html>
- <http://heather.cs.ucdavis.edu/~matloff/r.html>
- ...

R Packages

Lots of libraries called **packages**:

- Basic packages (included with the distribution): base, datasets, grDevices, graphics, grid, methods, splines, stats, stats4, tcltk, tools, utils

http://cran.r-project.org/doc/FAQ/R-FAQ.html#Which-add_002don-packages-exist-for-R_003f

- Add-on packages from lots of others

...

Why use a programming language versus using a spreadsheet?

- When you want to do something over and over again
- When you want to do something **systematically**

Experiment 1

Captured packets using Wireshark during a long (2150.12 second) VoIP call

⇒ at least: 107,505 RTP packets in each direction

⇒ 429 RTCP packets in one direction

<http://www.Wireshark.org>

Load the data, then extract relevant RTP packets

Starting with a tab separated file of the form:

```
"No."   "Time" "Source"      "Destination"  "Protocol"
      "RSSI" "Info"
"1443"  "17685.760952" "90.226.255.70" "217.211.xx.xx" "RTP"  ""
      "PT=ITU-T G.711 PCMA, SSRC=0x6E21893F, Seq=183, Time=46386 "
```

```
data1<-read.table("one-call.tab", sep="\t",
  header=TRUE, stringsAsFactors = FALSE)
```

Extract the traffic going to me:

```
To_Chip<-subset(data1, Source == "90.226.255.70",
  drop=TRUE)
```

Extract only the RTP protocol packets:

```
To_Chip_RTP<-subset(To_Chip, Protocol == "RTP",
  drop=TRUE)
```

Summary

summary(To_Chip_RTP)

No.	Time	Source
Min. : 1443	Min. :17686	90.226.255.70 :107515
1st Qu.: 55331	1st Qu.:18223	217.211.xx.xx : 0
Median :109224	Median :18761	41.209.78.223 : 0
Mean :109223	Mean :18761	62.20.251.42 : 0
3rd Qu.:163110	3rd Qu.:19298	81.228.11.66 : 0
Max. :217022	Max. :19836	90.226.251.20 : 0
		(Other) : 0

Destination	Protocol	RSSI
217.211.47.125:107515	RTP :107515	Mode:logical
41.209.78.223 : 0	ARP : 0	NA's:107515
62.20.251.42 : 0	DHCP : 0	
81.228.11.66 : 0	ICMP : 0	
90.226.251.20 : 0	NTP : 0	
90.226.255.70 : 0	RTCP : 0	
(Other) : 0	(Other): 0	

Info

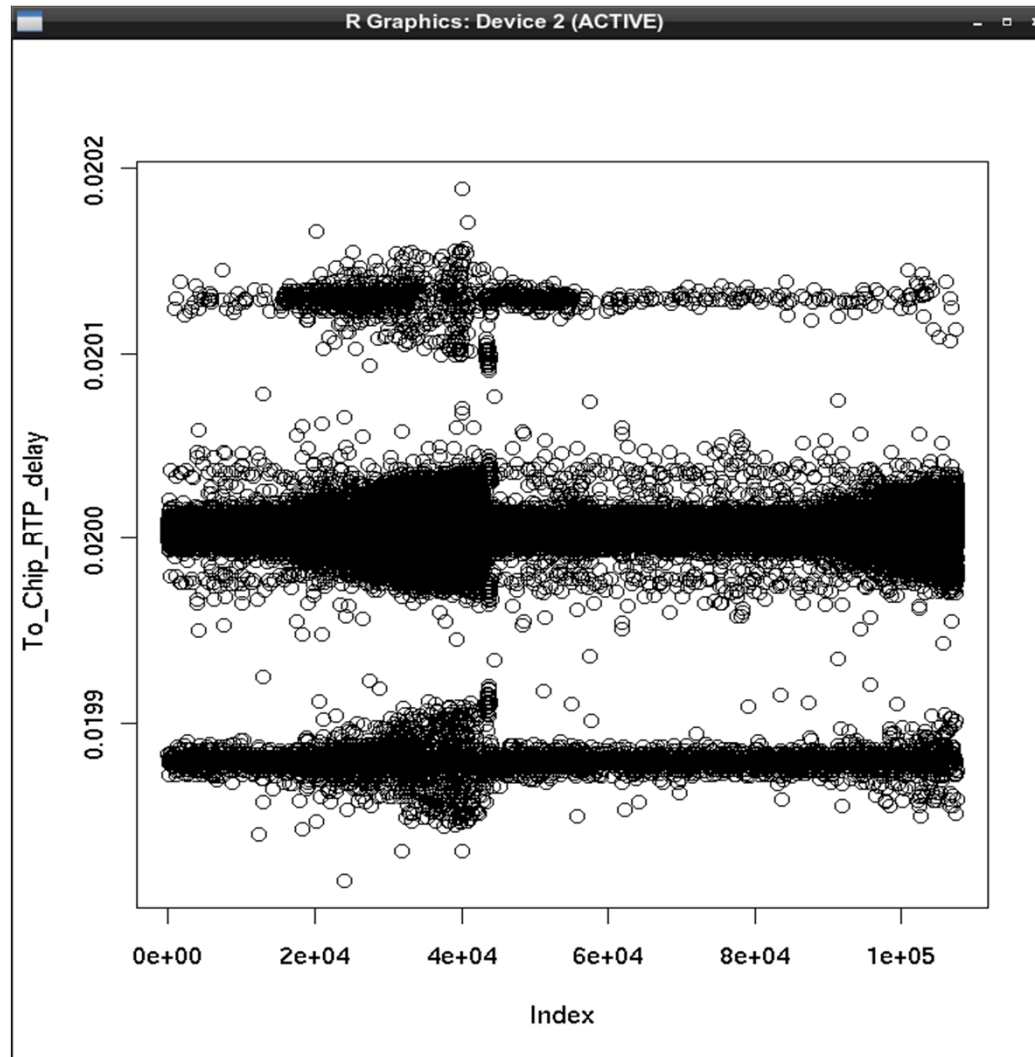
PT=ITU-T G.711 PCMA, SSRC=0x6E21893F, Seq=0, Time=10502866	: 1
PT=ITU-T G.711 PCMA, SSRC=0x6E21893F, Seq=10000, Time=12102866	: 1
PT=ITU-T G.711 PCMA, SSRC=0x6E21893F, Seq=10000, Time=1617106	: 1
PT=ITU-T G.711 PCMA, SSRC=0x6E21893F, Seq=10001, Time=12103026	: 1
PT=ITU-T G.711 PCMA, SSRC=0x6E21893F, Seq=10001, Time=1617266	: 1
PT=ITU-T G.711 PCMA, SSRC=0x6E21893F, Seq=10002, Time=12103186	: 1
(Other)	:107509

Inter-arrival delays

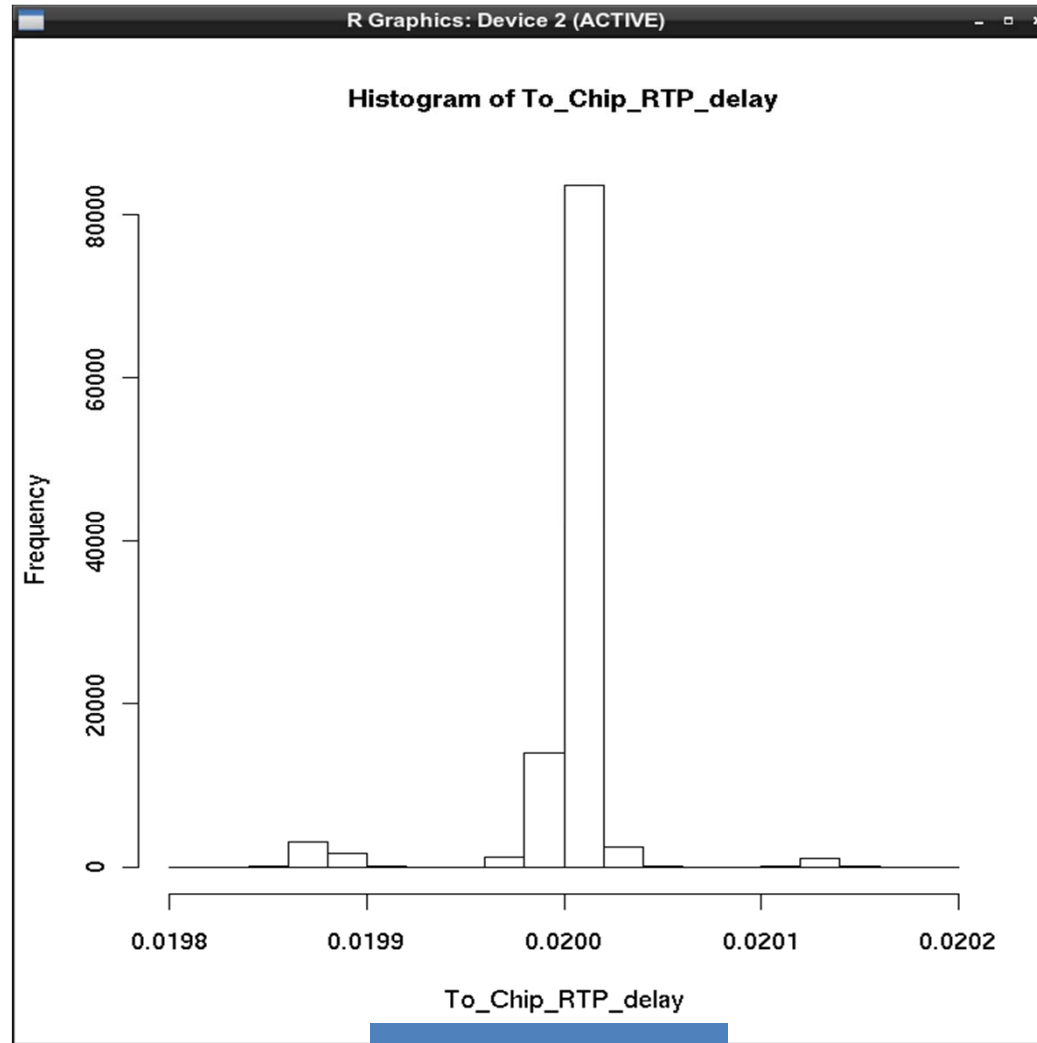
```
lvh<-nrow(To_Chip_RTP)
[1] 107515
lvh<-lvh-1> lvh
[1] 107514
To_Chip_RTP_delay=vector(length=(nrow(To_Chip_RTP) -
  1))
for (i in 1:lvh) {
To_Chip_RTP_delay[i]<-To_Chip_RTP$Time[i+1]-
  To_Chip_RTP$Time[i]
}
```

```
summary(To_Chip_RTP_delay)
  Min.    1st Qu.  Median    Mean    3rd Qu.  Max.
0.01981  0.02000  0.02000  0.02000  0.02001  0.02019
```

plot(To_Chip_RTP_delay)



hist(To_Chip_RTP_delay)

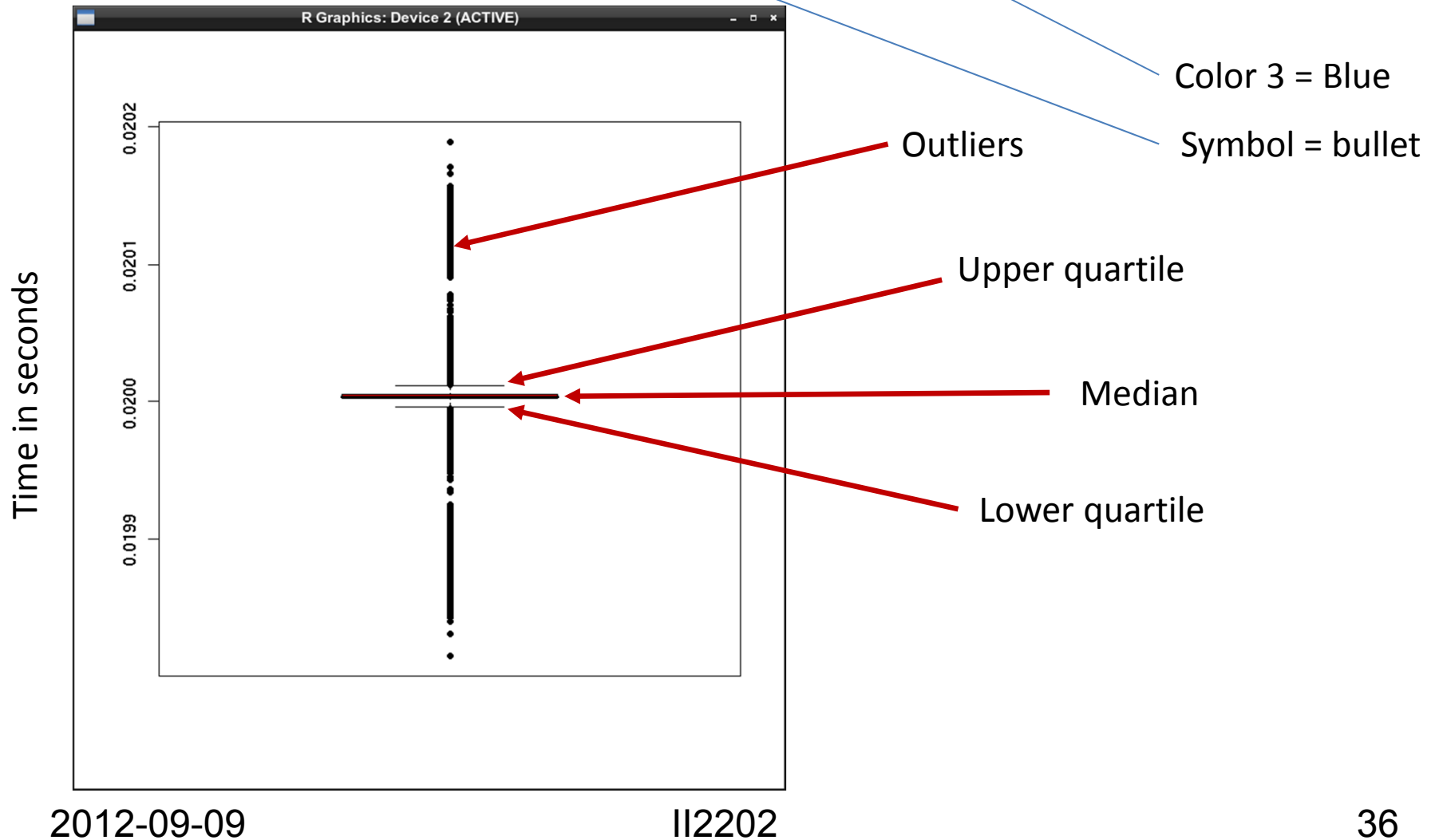


2012-09-09

Time in Seconds

35

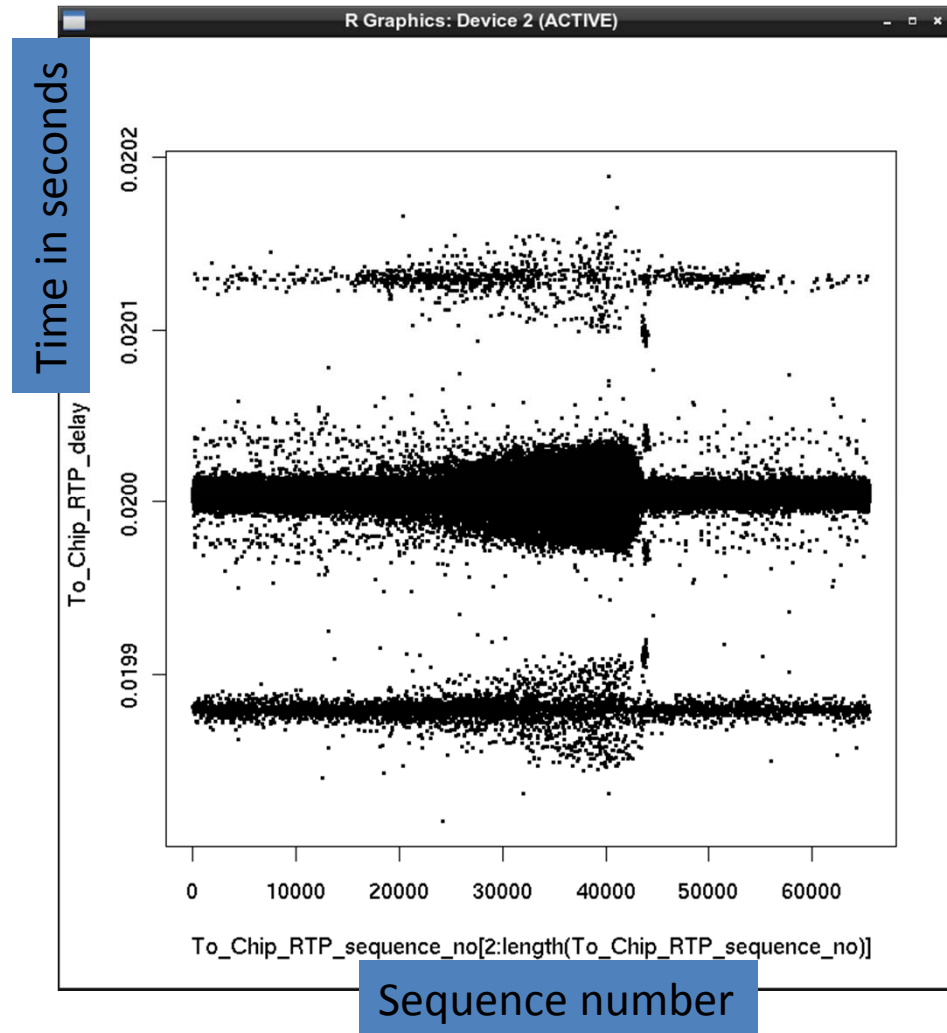
```
boxplot(To_Chip_RTP_delay,  
pch=20, col=3 )
```



Interarrival delay vs. sequence

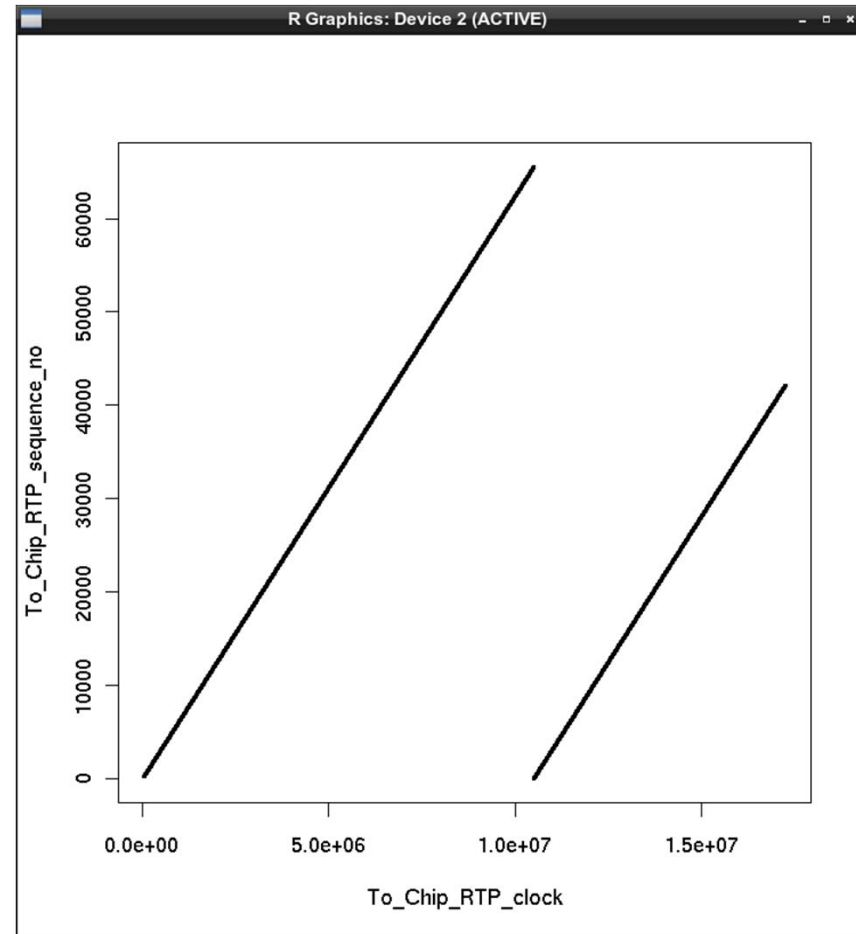
```
for (i in
  1:length(To_Chip_RTP$Info
)) {
z1<-
  strsplit(To_Chip_RTP$Info
[i], ",")
z2<-strsplit(z1[[1]][3],
  "=")
To_Chip_RTP_sequence_no[i]<
  -z2[[1]][2]
}

plot(To_Chip_RTP_sequence_no[2
:length(To_Chip_RTP_sequence
_no)],To_Chip_RTP_delay,
pch=20, cex=0.25)
```



RTP Clock vs. sequence

```
To_Chip_RTP_clock<-1
for (i in
  1:length(To_Chip_RTP$Info))
  {
z1<-
  strsplit(To_Chip_RTP$Info[i]
  , ",")
z2<-strsplit(z1[[1]][4], "=")
To_Chip_RTP_clock[i] <-
  z2[[1]][2]
}
plot ( To_Chip_RTP_clock,
  To_Chip_RTP_sequence_no,
  pch=20, cex=0.25)
```



Inter-arrival times of RTP packets: From network to local user agent

Using Excel:

Mean	0.019999999
Standard Error	9.28526E-08
Median	0.020004
Mode	0.020005
Standard Deviation	3.04446E-05
Sample Variance	9.26874E-10
Kurtosis	12.36652501
Skewness	-2.054662184
Range	0.000374
Minimum	0.019815
Maximum	0.020189
Sum	2150.11991
Count	107506
Confidence Level(95.0%)	1.8199E-07

Raw output from Microsoft Excel 2010 (Beta)

Note: count \neq length and the two programs get a different value for kurtosis

Using R functions:

```
mean(To_Chip_RTP_delay): 0.02
```

```
library(plotrix); std.error(To_Chip_RTP_delay):  
9.284597e-08
```

The mode is the most frequently occurring value
(hence via <https://stat.ethz.ch/pipermail/r-help/1999-December/005668.html>):

```
names(sort(-table(To_Chip_RTP_delay)))[1]:  
"0.0200049999984913"
```

```
sd(To_Chip_RTP_delay): 3.044357e-05
```

```
var(To_Chip_RTP_delay): 9.268109e-10
```

```
library(moments);
```

```
kurtosis(To_Chip_RTP_delay): 15.36689
```

```
skewness(To_Chip_RTP_delay): -2.054706
```

```
min(To_Chip_RTP_delay): 0.019815
```

```
max(To_Chip_RTP_delay): 0.020189
```

```
sum(To_Chip_RTP_delay): 2150.28
```

```
length(To_Chip_RTP_delay): 107514
```

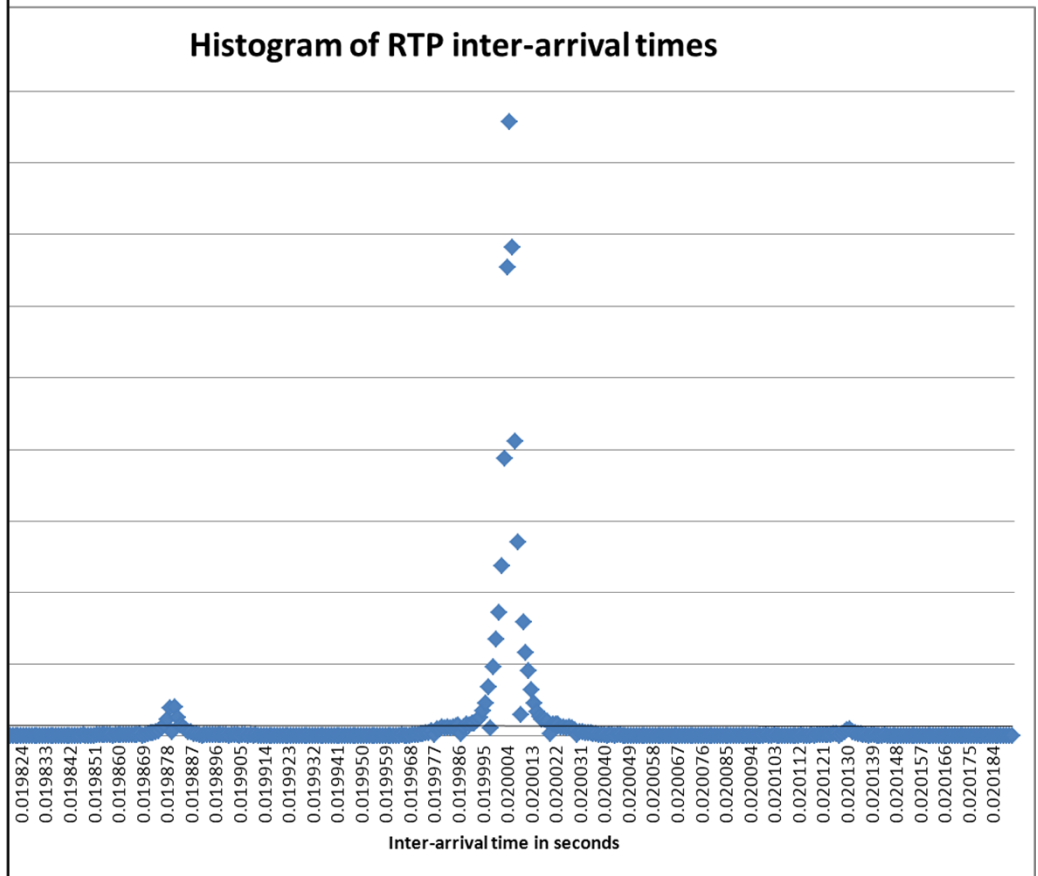
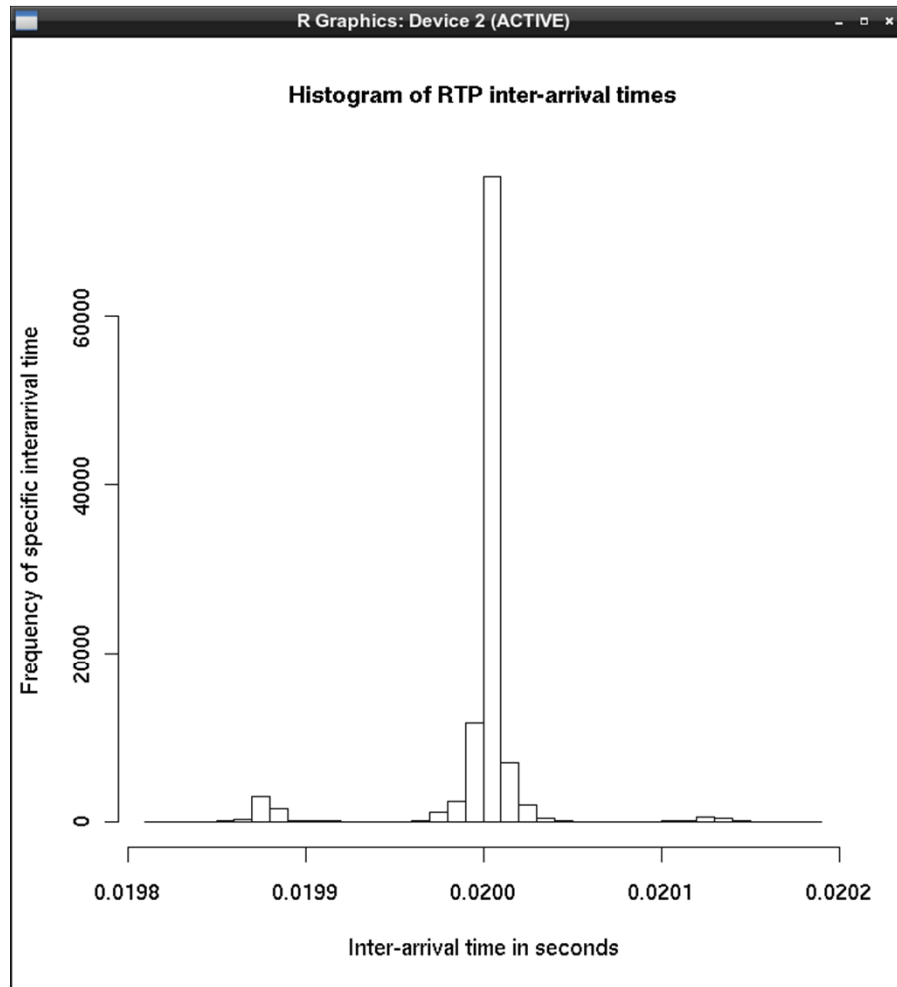
```
qnorm(0.975)*std.error(To_Chip_RTP_delay):  
1.819748e-07
```

2012-09-09

I12202

39

R vs. Excel histogram



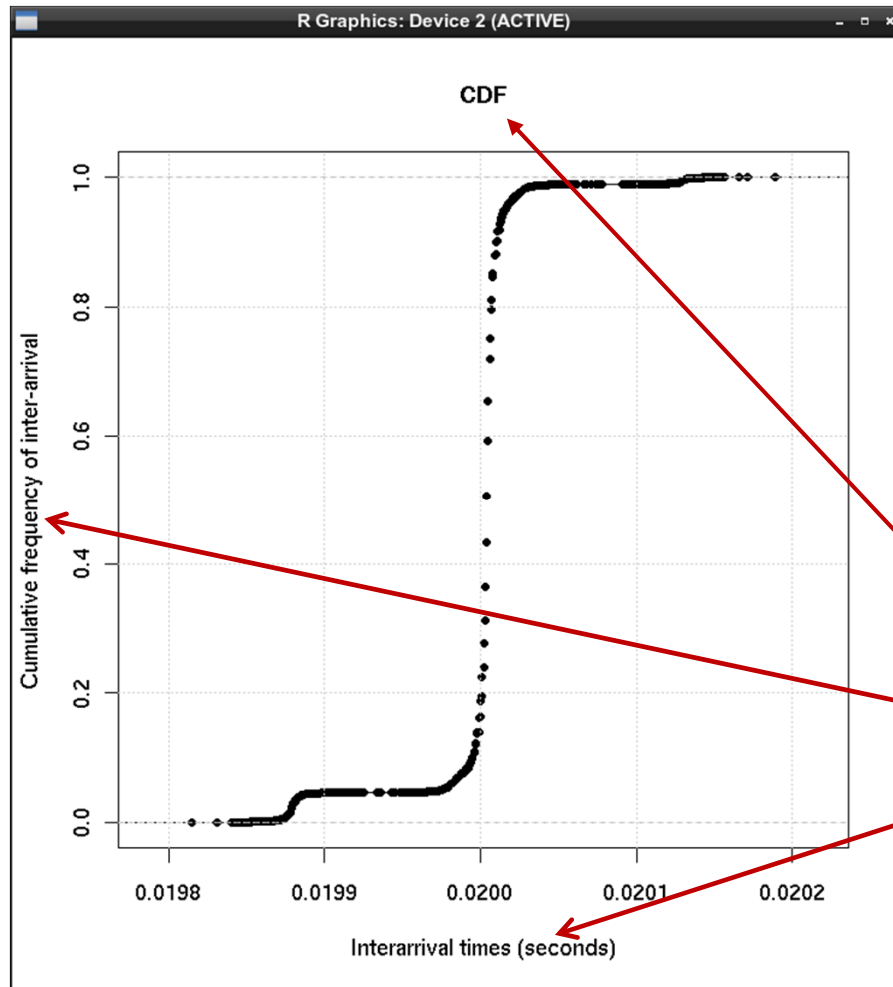
```
hist(To_Chip_RTP_delay, ylab="Frequency of specific interarrival time",  
xlab="Inter-arrival time in seconds", main="Histogram of RTP inter-arrival times", breaks=46)
```

2012-09-09

II2202

40

Plot as a Cumulative Distribution



```
plot(ecdf(To_Chip_RTP_delay),  
pch=20, cex=1, main="CDF",  
xlab="Interarrival times  
(seconds)", ylab="Cumulative  
frequency of inter-arrival"); grid()
```

cex = size of text or symbol for plot
1 = default

main = major label

ylab = y label

xlab = x label

grid() adds the grid in the background

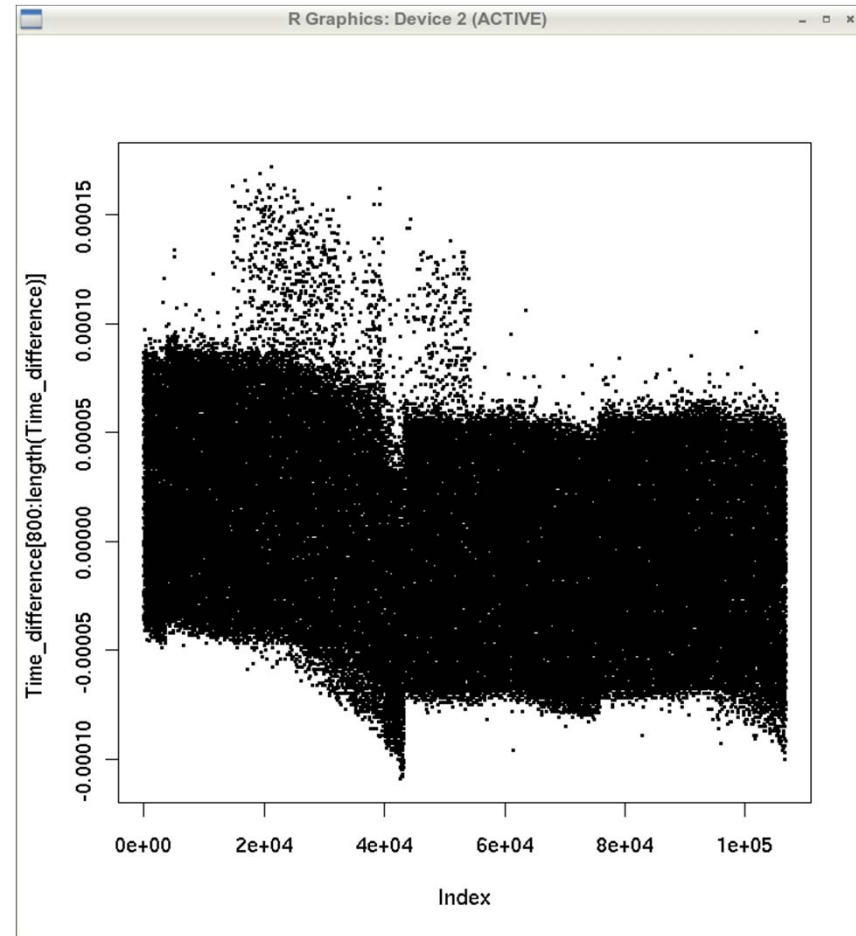
With varying numbers of samples

Descriptive Statistics	First 100	First 1K	First 10K	First 100K
Mean	0.02000071	0.020000066	0.020000004	0.02
Standard Error	2.12714E-06	7.53406E-07	2.51164E-07	9.69855E-08
Median	0.020005	0.020004	0.020004	0.020004
Mode	0.020005	0.020005	0.020005	0.020005
Standard Deviation	2.12714E-05	2.38248E-05	2.51164E-05	3.06695E-05
Sample Variance	4.52471E-10	5.67621E-10	6.30831E-10	9.40618E-10
Kurtosis	28.87137928	21.46428225	19.07376827	12.23083198
Skewness	-5.453831468	-4.509853108	-3.831289593	-2.003065575
Range	0.000135	0.000252	0.000277	0.000374
Minimum	0.01988	0.019872	0.019868	0.019815
Maximum	0.020015	0.020124	0.020145	0.020189
Sum	2.000071	20.000066	200.000044	1999.999951
Count	100	1000	10000	100000
Confidence Level(95.0%)	4.2207E-06	1.47844E-06	4.92331E-07	1.9009E-07

How does the measured data differ from the expected data?

```
for (i in
      1:length(To_Chip_RTP$Time)) {
Time_difference[i]=
  (To_Chip_RTP$Time[i]-To_Chip_RTP$Time[1]) -
  ((as.numeric(To_Chip_RTP_clock[i]) -
    as.numeric(To_Chip_RTP_clock[1]))/8000)
}
plot( Time_difference[800:
      length(Time_difference)]
      , pch=20, cex=0.25)
```

Scale the bullet
to ¼ size

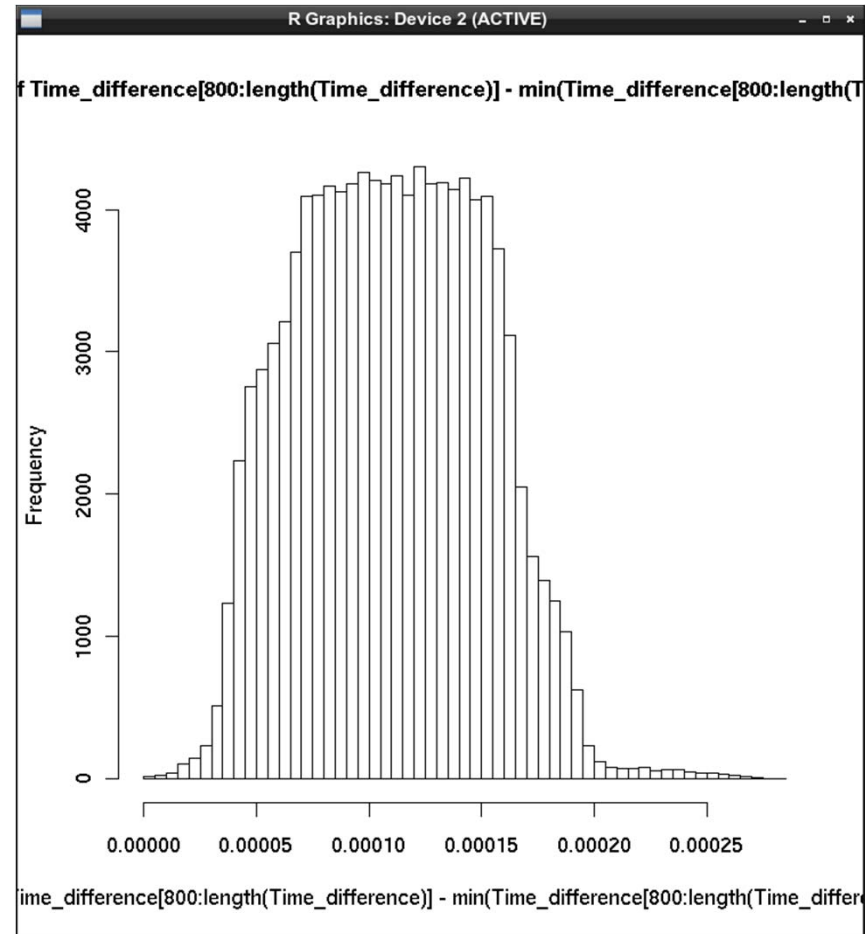


How does the measured data differ from the expected data?

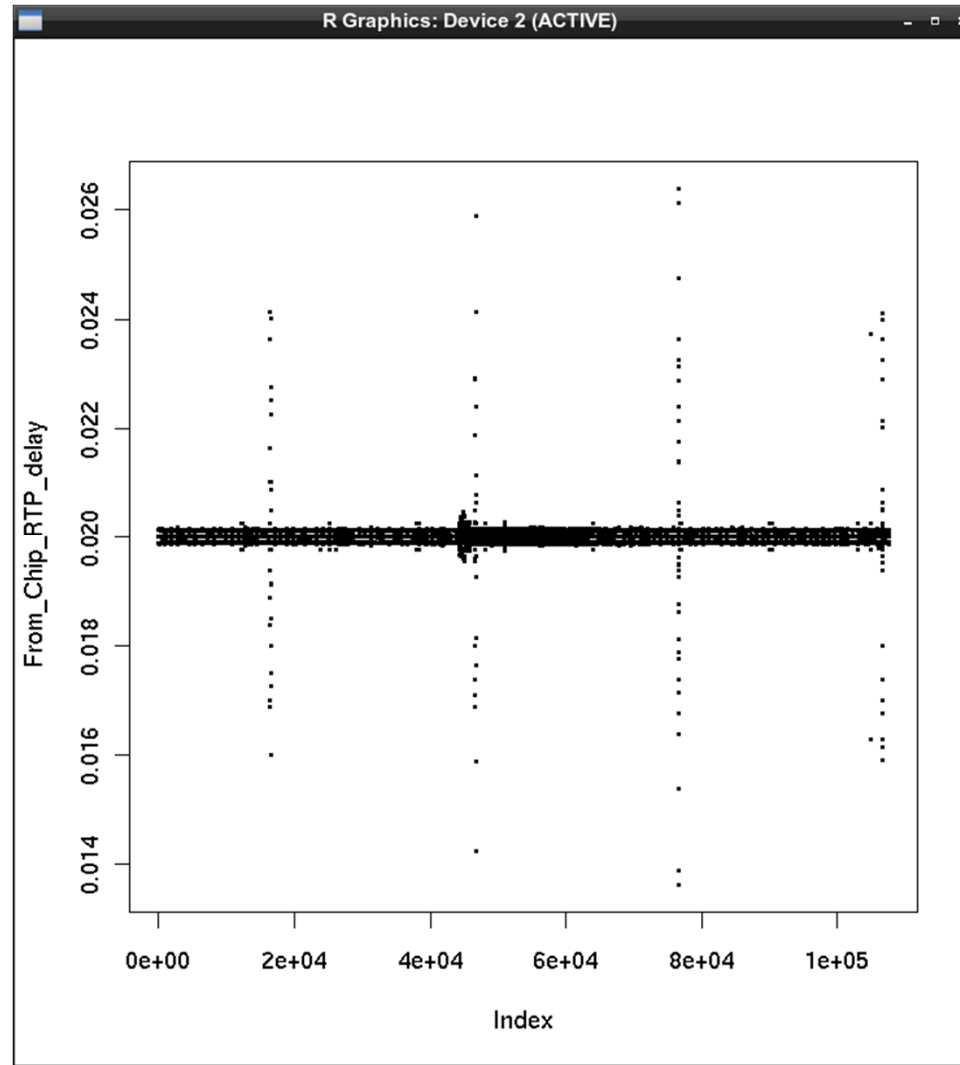
Since delay can not be negative, the real difference can be found by subtracting the min()

⇒

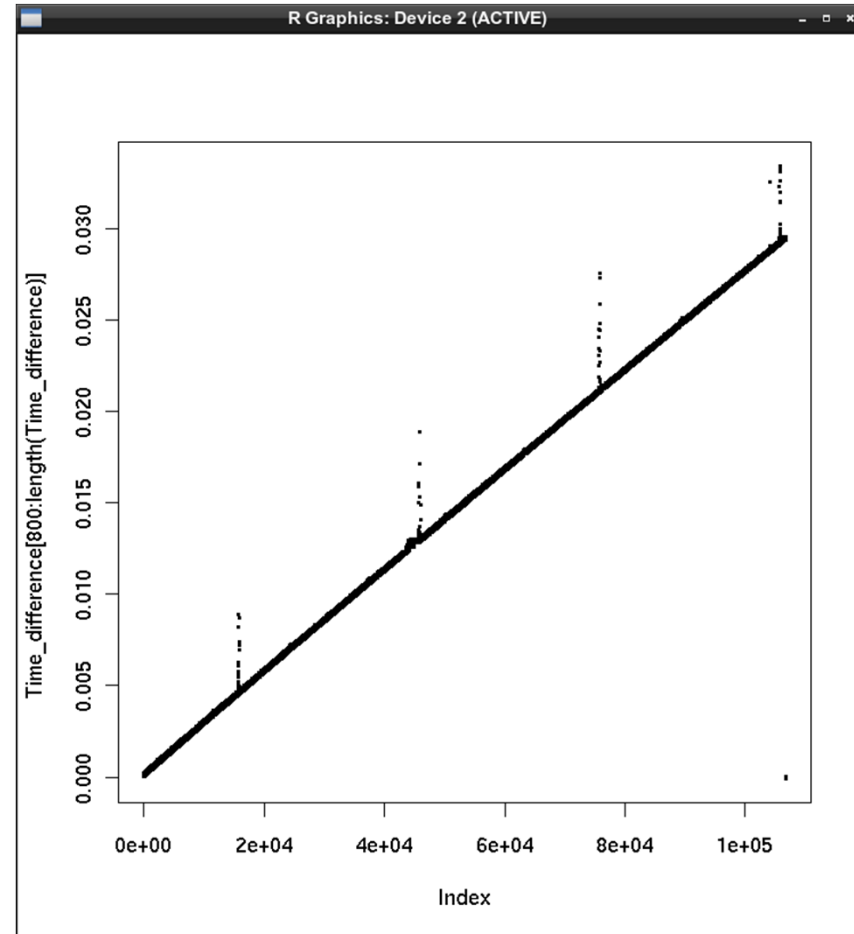
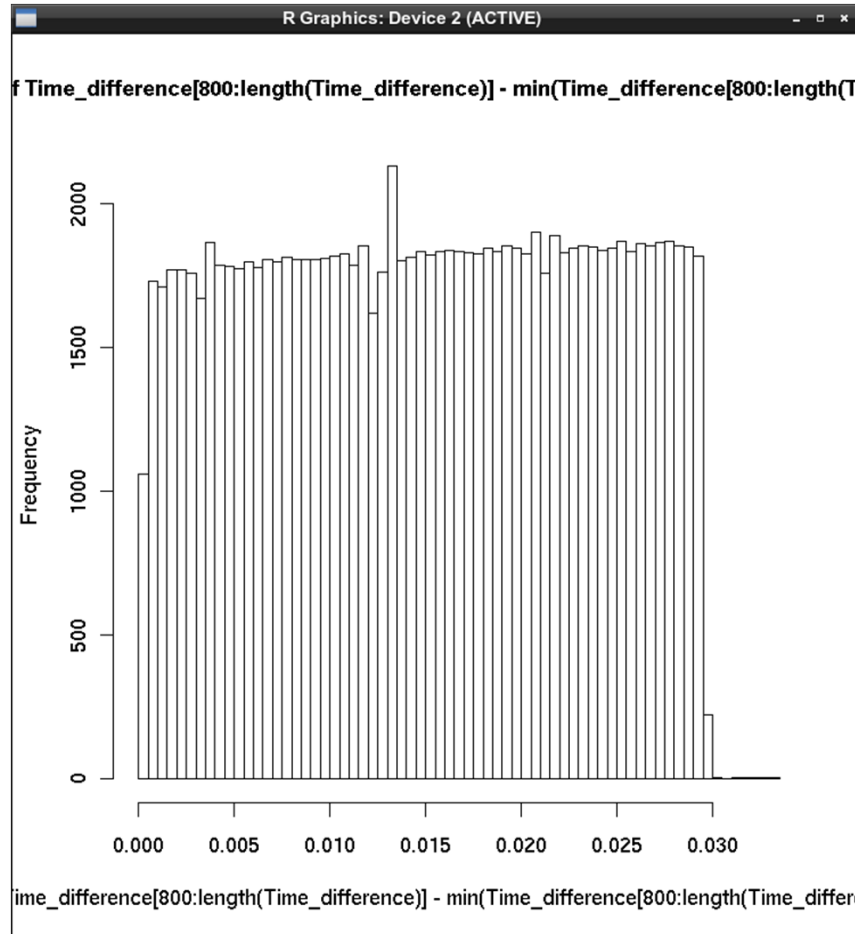
```
hist(  
  Time_difference[800:length(  
    Time_difference)] -  
  min(Time_difference[800:  
    length(Time_difference)])  
  , breaks=100)  
  Number of bins to use
```



Uplink inter-arrival times



For traffic in the opposite direction



Difference histogram and difference plot \Rightarrow the clock is drifting wrt the Wireshark clock

Experiment 2: DNS lookup

Captured DNS traffic with Wireshark using filter `udp.port==53` then exported in PDML format producing a file

`dns-capture-20100915a.pdml`

Using Emacs filtered out all lines except those containing `dns.time` fields

```
data2<-read.table("dns-capture-20100915a-a.txt",  
header=FALSE)
```

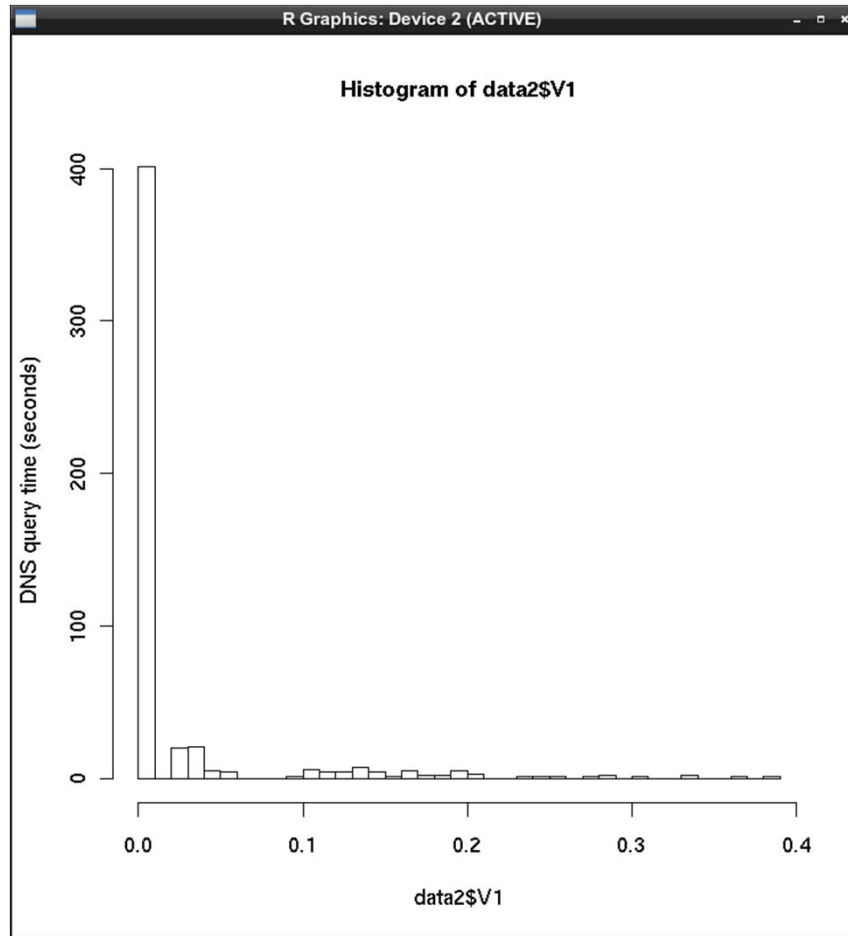
```
summary(data2)    V1  
  Min.   :0.000710  
 1st Qu.:0.000896  
  Median:0.001066  
   Mean  :0.023868  
 3rd Qu.:0.003329  
   Max.   :0.389880
```

```
foo(data2$V1, length(data2$V1))
```

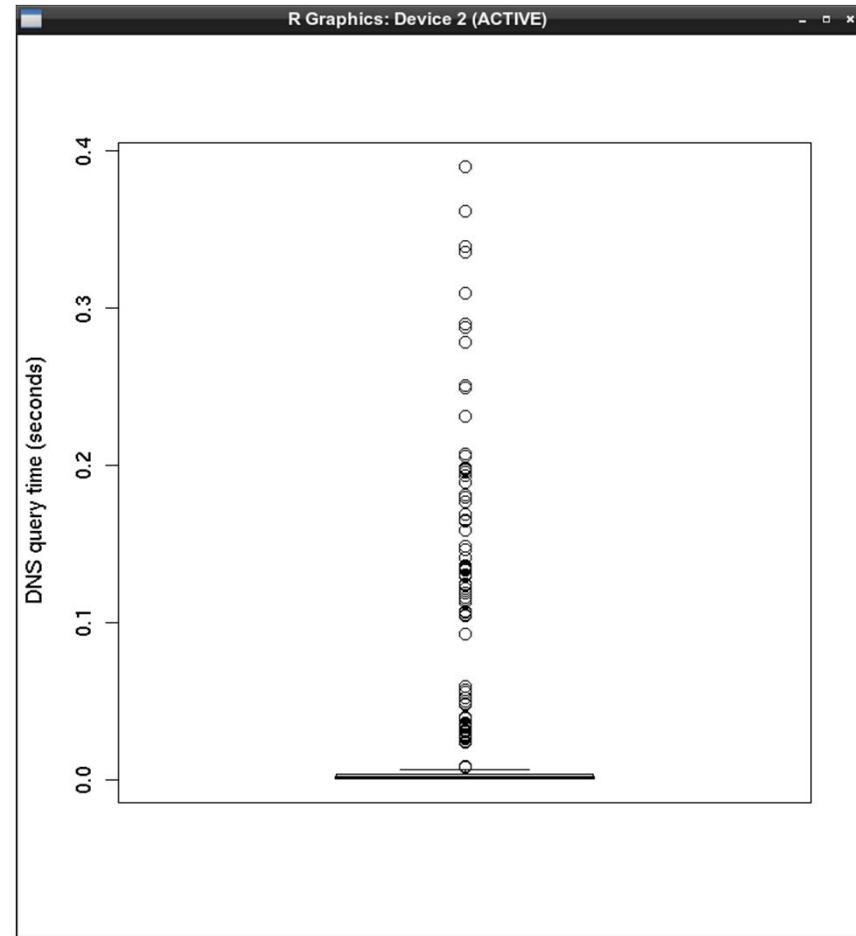
```
Mean:      0.023868 s  
std.error: 0.002669 s  
Mode:      0.000896 s  
Sd:        0.060045 s  
Var:       0.003605 s  
Kurtosis:  14.3  
Skewness:  3.3  
Min:       0.00071  s  
Max:       0.38988  s  
Sum:       12.077197 s  
Count:     506  
Conf (95%) 0.004837 s
```

DNS lookup time graphs

```
hist(data2$V1, ylab="DNS query time (seconds)", breaks=40)
```



```
boxplot(data2$V1, ylab="DNS query time (seconds)")
```

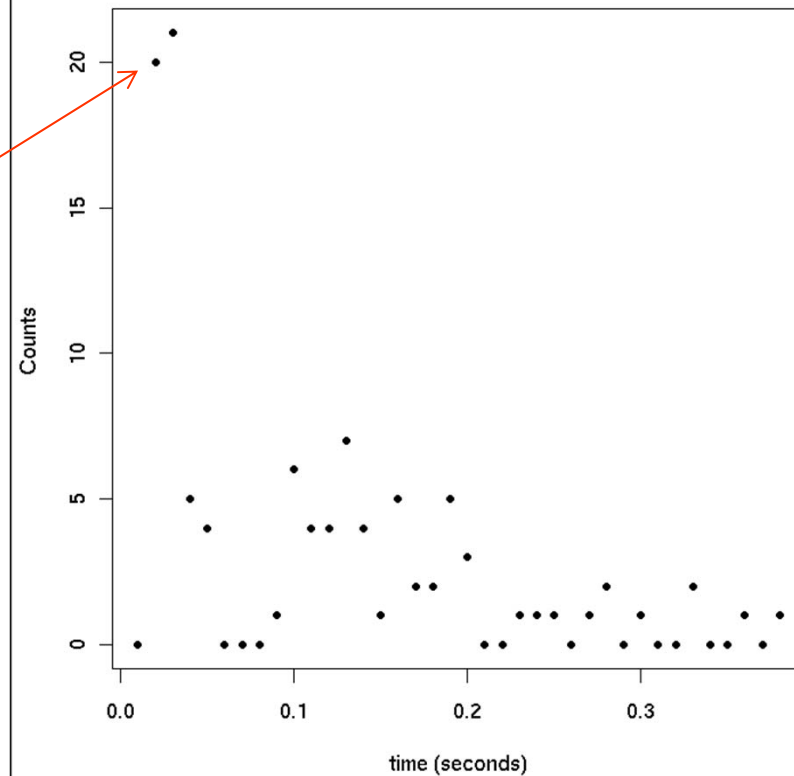
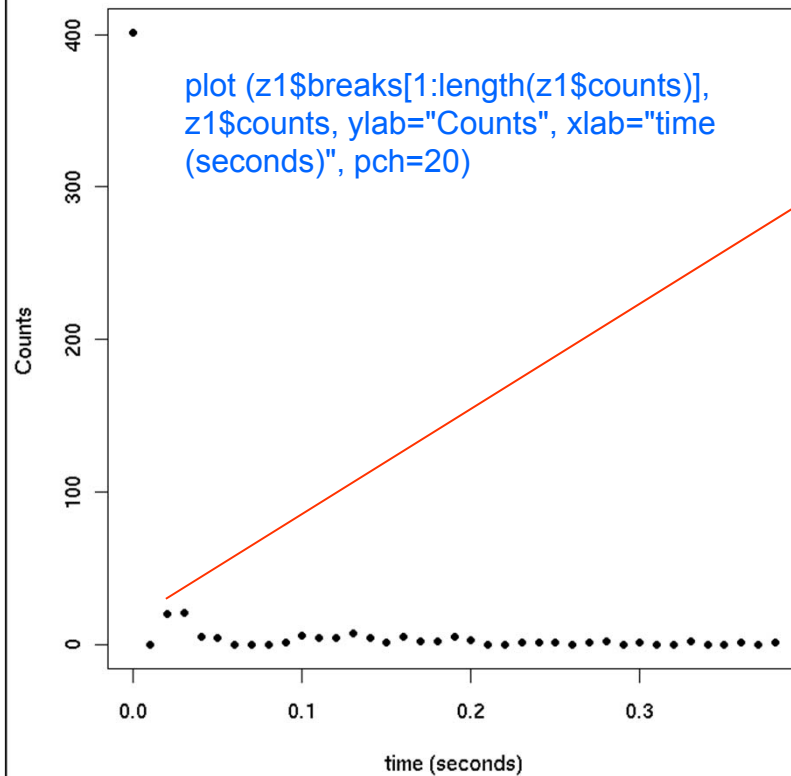



```
z1<-hist(data2$V1, breaks=40)
summary(z1)
```

	Length	Class	Mode
breaks	40	-none-	numeric
counts	39	-none-	numeric
intensities	39	-none-	numeric
density	39	-none-	numeric
mids	39	-none-	numeric
xname	1	-none-	character
equidist	1	-none-	logical

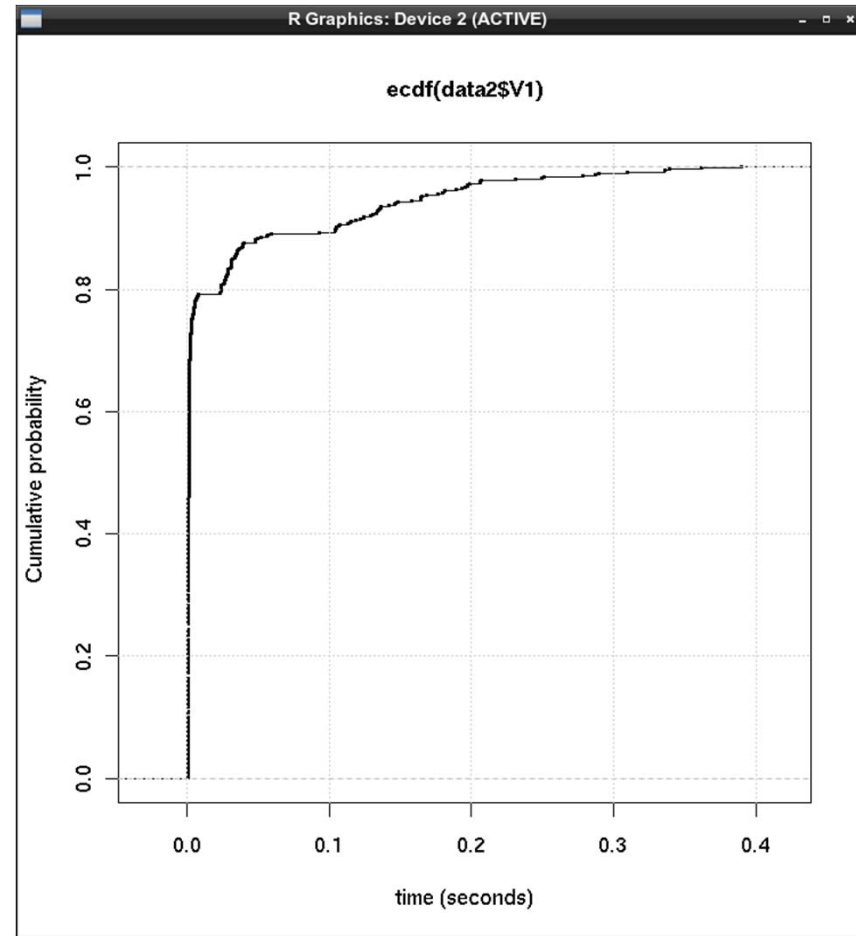
More graphs: change scale

```
plot(z1$breaks[2:length(z1$counts)],
     z1$counts[2:length(z1$counts)],ylab="Counts",
     xlab="time (seconds)", pch=20)
```



DNS response CDF

```
plot  
  (ecdf(data2$V1),  
   xlab="time  
   (seconds)",  
   ylab="Cumulative  
   probability", pch=20,  
   cex=0.25)  
grid()
```



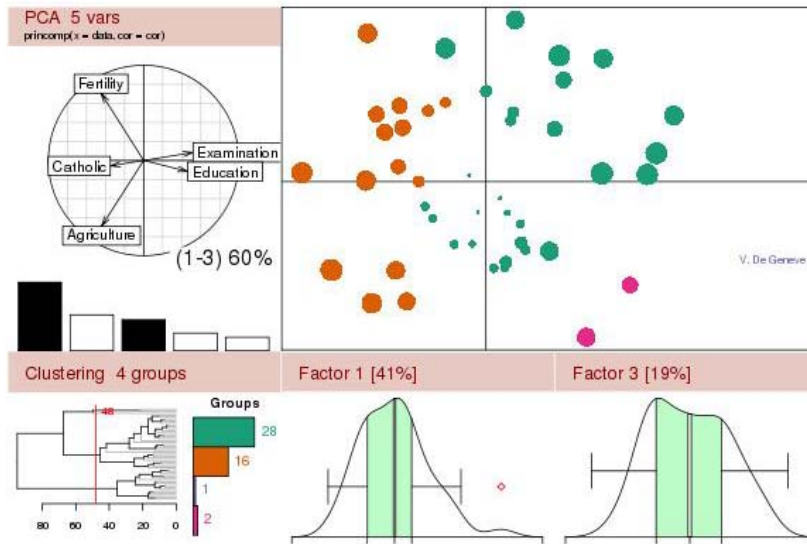
So how do **you** get started using R?

www.r-project.org



The R Project for Statistical Computing

- About R
 - [What is R?](#)
 - [Contributors](#)
 - [Screenshots](#)
 - [What's new?](#)
- Download, Packages
 - [CRAN](#)
- R Project Foundation
 - [Members & Donors](#)
 - [Mailing Lists](#)
 - [Bug Tracking](#)
 - [Developer Page](#)
 - [Conferences](#)
 - [Search](#)
- Documentation
 - [Manuals](#)
 - [FAQs](#)
 - [The R Journal](#)
 - [Wiki](#)
 - [Books](#)
 - [Certification](#)
 - [Other](#)
- Misc
 - [Bioconductor](#)
 - [Related Projects](#)
 - [User Groups](#)
 - [Links](#)



Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#)
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News:

- **R version 2.15.1** (Roasted Marshmallows) has been released on 2012-06-22.
- [The R Journal Vol.4/1](#) is available.
- [useR! 2012](#), took place at Vanderbilt University, Nashville Tennessee, USA, June 12-15, 2012.
- [useR! 2013](#), will take place at the University of Castilla-La Mancha, Albacete, Spain, July 10-12 2013.

This server is hosted by the [Institute for Statistics and Mathematics](#) of the [WU Wien](#)

Press CRAN for mirror sites



About R
[What is R?](#)
[Contributors](#)
[Screenshots](#)
[What's new?](#)

Download, Packages
[CRAN](#)

R Project
[Foundation](#)
[Members & Donors](#)
[Mailing Lists](#)
[Bug Tracking](#)
[Developer Page](#)
[Conferences](#)
[Search](#)

Documentation
[Manuals](#)
[FAQs](#)
[The R Journal](#)
[Wiki](#)
[Books](#)
[Certification](#)
[Other](#)

Misc
[Bioconductor](#)

CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: [main page](#), [windows release](#), [windows old release](#).

Argentina	http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
	http://r.mirror.mendoza-conicet.gob.ar/	CONICET Mendoza
Australia	http://cran.csiro.au/	CSIRO
	http://cran.ms.unimelb.edu.au/	University of Melbourne
Austria	http://cran.at.r-project.org/	Wirtschaftsuniversitaet Wien
Belgium	http://www.freeststatistics.org/cran/	K.U.Leuven Association
Brazil	http://cran-r.c3sl.ufpr.br/	Universidade Federal do Parana
	http://cran.fiocruz.br/	Oswaldo Cruz Foundation, Rio de Janeiro
	http://www.vps.fmvz.usp.br/CRAN/	University of Sao Paulo, Sao Paulo
	http://brieger.esalq.usp.br/CRAN/	University of Sao Paulo, Piracicaba
Canada	http://cran.stat.sfu.ca/	Simon Fraser University, Burnaby
	http://mirror.its.dal.ca/cran/	Dalhousie University, Halifax
	http://probability.ca/cran/	University of Toronto
	http://cran.skazkaforyou.com/	iWeb, Montreal
	http://cran.parentingamerica.com/	iWeb, Montreal
Chile	http://dirichlet.mat.puc.cl/	Pontificia Universidad Catolica de Chile, Santiago
China	http://ftp.ctex.org/mirrors/CRAN/	CTEX.ORG

Choose mirror site near your location, for
example:

<http://ftp.sunet.se/pub/lang/CRAN/>

Swedish University Computer Network
(SUNET)

R Distributions



CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

Documentation

[Manuals](#)

[FAQs](#)

[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for MacOS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2012-06-22, Roasted Marshmallows): [R-2.15.1.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

R for Windows

R for Windows

Subdirectories:

[base](#)

Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).

[contrib](#)

Binaries of contributed packages (managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

[Rtools](#)

Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

R for Macs

R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.5 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) [here](#). Releases for old Mac OS X systems (through Mac OS X 10.4) and PowerPC Macs can be found in the [old](#) directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 2.15.1 released on 2012/06/22

This binary distribution of R and the GUI supports Intel (32-bit and 64-bit) based Macs on Mac OS X 10.5 (Leopard), 10.6 (Snow Leopard) and 10.7 (Lion).

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type
md5 R-2.15.1.pkg
in the *Terminal* application to print the MD5 checksum for the R-2.15.1.pkg image.

Files:

[R-2.15.1-signed.pkg](#) (latest version)
MD5-hash: c63e2efbd4aadcd8ab9f0edda8887e3
(ca. 64MB)

R 2.15.1 binary for Mac OS X 10.5 (Leopard) and higher, signed package. Contains R 2.15.1 framework, R.app GUI 1.52 in 32-bit and 64-bit for Intel Macs. The above file is an Installer package which can be installed by double-clicking. Depending on your browser, you may need to press the control key and click on this link to download the file.

This package **only** contains the R framework, 32-bit GUI (R.app) and 64-bit GUI (R64.app). **For Tcl/Tk libraries (needed if you want to use tcltk) and GNU Fortran (needed if you want to compile packages from sources that contain FORTRAN code) please see [the tools directory](#).**

[R-2.15.1.pkg](#)
MD5-hash: f0ab42c814dc4bfa26ab7cb8e38356e
(ca. 64MB)

Same package as above but not signed. This is provided only as a historical reference to the original R 2.15.1 release which was not signed, but some installation policies in Mac OS X 10.8 (Mountain Lion) may require signed packages and thus the above supersedes the original unsigned release. The actual content is identical in both packages.

[Mac-GUI-1.51.tar.gz](#)
MD5-hash: 598dd66bd9d421657e3b660d82501504

Sources for the R.app GUI 1.51 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not intended for regular users. Read the INSTALL file for further instructions.

Linux Distributions



- [Home](#)
- [Search](#)
- [News](#)
- [About](#)
- [Contact](#)

[Feedback](#)







Index of /pub/lang/CRAN/bin/linux

Name	Last modified	Size
↶ Parent Directory		-
debian/	29-Apr-2012 11:06	-
redhat/	25-Nov-2009 18:01	-
suse/	16-Feb-2012 15:09	-
ubuntu/	30-May-2012 04:02	-

This service is maintained by archive@ftp.sunet.se

For example for Ubuntu

Index of /pub/lang/CRAN/bin/linux/ubuntu

Name	Last modified	Size
 Parent Directory		-
 hardy/	27-Aug-2012 04:01	-
 lucid/	27-Aug-2012 04:01	-
 natty/	27-Aug-2012 04:02	-
 oneiric/	27-Aug-2012 04:02	-
 precise/	27-Aug-2012 04:02	-

UBUNTU PACKAGES FOR R

Announcement: Due to sever issues, the Ubuntu CRAN packages have been signed with a new pgp key. See SECURE APT below.

R packages for Ubuntu on i386 and amd64 are available for all stable Desktop releases of Ubuntu until their official end of life date. However, only the latest Long Term Support (LTS) release is fully supported. As of April 28, 2011, the supported releases are Precise Pangolin (12.04; LTS), Oneiric Ocelot (11.10), Natty Nawwhal (11.04), Lucid Lynx (10.04; LTS) and Hardy Heron (8.04; LTS).

See <https://wiki.ubuntu.com/Releases> for details.

The previous LTS release, Hardy Heron (8.04), will remain supported as long as backporting of packages does not involve too much additional work (as is currently the case).

openSUSE – includes R

RPM Packages Providing R for OpenSUSE

Table of Contents

- [1 RPMs providing R for OpenSUSE](#)
 - [1.1 News](#)
 - [1.2 Installation](#)
 - [1.3 Installing R with 1-click-install](#)
 - [1.4 Installing using the command line](#)
 - [1.5 Staying uptodate with zypper](#)
 - [1.6 Using R-devel](#)
 - [1.7 Download for later installation](#)
 - [1.8 Maintenance](#)

Note that this text was missing from the SUNET web site as of 2012.08.27 at 8:35, this version is from http://watson.nci.nih.gov/cran_mirror/

1 RPMs providing [R](#) for OpenSUSE

1.1 News

R has its own top level project in openSUSE's build service `devel:languages:R`.

R is included in the latest releases of openSUSE since 11.4, so it may be installed without adding any repositories directly from yast. Obviously you only get the version of R that was current, when a openSUSE release got frozen. I.e. 12.1 contains R-2.13.2 but we already have R-2.14.1 out by now. Read on to find out how to stay current with R!

Below `devel:languages:R` you find a few subprojects:

- R-base (built on release date)
- R-patched (daily, **recommended**) and
- R-devel (daily)

for all actively maintained releases of openSUSE available. For the time being these are: 11.3 and 12.1. Furthermore the adventurous can download packages for Factory and Tumbleweed. On the other end even SLE 10, 11 and 11SP1 are supported.

All these packages provide a resource 'R-base' for installation.

R Manuals

The R Manuals

edited by the R Development Core Team.

Current Version: 2.15.1 (Roasted Marshmallows, 2012-06-22)

The following manuals for R were created on Debian Linux and may differ from the manuals for Mac or Windows on platform-specific pages, but most parts will be identical for all platforms. The correct version of the manuals for each platform are part of the respective R installations. Here they can be downloaded as PDF files or directly browsed as HTML:

- **An Introduction to R** is based on the former "Notes on R", gives an introduction to the language and how to use R for doing statistical analysis and graphics. [[browse HTML](#) | [download PDF](#)]
- A draft of **The R language definition** documents the language *per se*. That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions. [[browse HTML](#) | [download PDF](#)]
- **Writing R Extensions** covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces. [[browse HTML](#) | [download PDF](#)]
- **R Data Import/Export** describes the import and export facilities available either in R itself or via packages which are available from CRAN. [[browse HTML](#) | [download PDF](#)]
- **R Installation and Administration** [[browse HTML](#) | [download PDF](#)]
- **R Internals**: a guide to the internal structures of R and coding standards for the core team working on R itself. [[browse HTML](#) | [download PDF](#)]
- **The R Reference Index**: contains all help files of the R standard and recommended packages in printable form. [[download PDF](#), [SMB](#), [approx. 3500 pages](#)]

Translations of manuals into other languages than English are available from the [contributed documentation](#) section (only a few translations are available).

The latex or texinfo sources of the latest version of these documents are contained in every R source distribution (in the subdirectory `doc/manual` of the extracted archive). Older versions of the manual can be found in the respective [archives of the R sources](#). The HTML versions of the manuals are also part of most R installations (accessible using function `help.start()`).



CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

Documentation

[Manuals](#)

[FAQs](#)

[Contributed](#)

R Packages

Contributed Packages

Available Packages

Currently, the CRAN package repository features 4002 available packages.

[Table of available packages, sorted by date of publication](#)

[Table of available packages, sorted by name](#)

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository. The manual [R Installation and Administration \[PDF\]](#) (also contained in the R base sources) explains the process in detail.

[CRAN Task Views](#) allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 29 views are available.

Package Check Results

All packages are tested regularly on machines running [Debian GNU/Linux](#), [Fedora](#) and Solaris. Packages are also checked under MacOS X and Windows, but typically only on the day the package appears on CRAN.

The results are summarized in the [check summary](#) (some [timings](#) are also available). Additional details for Windows checking and building can be found in the [Windows check summary](#)

Writing Your Own Packages

The manual [Writing R Extensions \[PDF\]](#) (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

Repository Policies

The manual [CRAN Repository Policy \[PDF\]](#) describes the policies in place for the CRAN package repository.

Related Directories

Obtain an R Package

gplots: Various R programming tools for plotting data

Various R programming tools for plotting data

Version: 2.11.0
Depends: R (\geq 2.10), [gtools](#), [gdata](#), stats, [caTools](#), grid, [KernSmooth](#), [MASS](#), datasets
Suggests: [gtools](#)
Published: 2012-06-08
Author: Gregory R. Warnes. Includes R source code and/or documentation contributed by (in alphabetical order): Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber, Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz, Bill Venables
Maintainer: Gregory R. Warnes <greg at warnes.net>
License: [GPL-2](#)
In views: [Graphics](#)
CRAN checks: [gplots results](#)

Downloads:

Package source: [gplots_2.11.0.tar.gz](#)
MacOS X binary: [gplots_2.11.0.tgz](#)
Windows binary: [gplots_2.11.0.zip](#)
Reference manual: [gplots.pdf](#)
News/ChangeLog: [NEWS](#) [ChangeLog](#)
Old sources: [gplots archive](#)

Reverse dependencies:

Reverse depends: [allanvar](#), [ares](#), [ARTIVA](#), [bayesMCCLust](#), [cellVolumeDist](#), [DandEFA](#), [ddepn](#), [DiagTest3Grp](#), [FLLat](#), [ghyp](#), [GMD](#), [gregmisc](#), [HumMeth27QCReport](#), [iCluster](#), [lmSupport](#), [MADAM](#), [Meth27QC](#), [muma](#), [pbkrtest](#), [qat](#), [rehh](#), [ResearchMethods](#), [ROCR](#), [rsgcc](#), [sisus](#), [swamp](#), [TIMP](#)
Reverse imports: [GPvam](#), [MADAM](#), [scapeMCMC](#)
Reverse suggests: [gmodels](#), [heplots](#), [HistData](#), [MKmisc](#), [opm](#), [PerformanceAnalytics](#), [rattle](#), [RGraphics](#), [simba](#), [spartan](#)
Reverse enhances: [GMD](#)

Install an R Package (linux)

type Linux the command:

```
R CMD INSTALL package.tar.gz
```

(No need to ungzp or untar the package.)

Importing Data into R

From a comma separated file:

```
DataD1 ← read.csv(file="table.csv", header=TRUE, ...)  
help(read.csv) for all the options which include reading row names
```

```
DataD1 ← read.table(file="table.csv", sep="," ...)  
help(read.table) for all the options
```

library(gdata) (load the package **gdata**)

```
DataD4 ← read.xls("table.xls", sheet=4, ...)  
help(read.xls) for all the options
```

In each case above the file is put into a “data frame” which can be referenced by row and column.

Example using a csv File

```
cup.diameters <- function()  
{  
phant1 <-  
read.csv(file="hip_stats1.csv",header=TRUE,sep=",");  
diameter1 <- ((phant1[2:15, 10])*2)  
  
phant1a <-  
read.csv(file="hip_stats1a.csv",header=TRUE,sep=",");  
diameter1a <- ((phant1a[2:15, 10])*2)  
  
phant2 <-  
read.csv(file="hip_stats2.csv",header=TRUE,sep=",");  
diameter2 <- ((phant2[2:15, 10])*2)  
  
total_cup <- c(diameter1, diameter1a, diameter2)  
print("total cup diameter is")  
print(total_cup)  
total.cup <- total_cup
```

Importing Any File

Using the function **scan** any style file can be read, e.g.,

```
invitro.cals -> function(string)
{
# string is the directory path to all the files to be used
# paste() adds a file name to the directory path
# what is the type of file to be used
# The result is an unformatted string of numbers in R
thalf <- scan(paste(string, "std.decay.time",
  sep = " "), what=numeric())
}
```

See “help(scan)” for a complete list of parameters than can be read.

Exporting Any File

Use the R function **cat** to write out a text file just as the data is in R.

Use the R function **dput** to write out a file so that it can be directly read using the R function **dget**.

Plot Formatting

```
cup.measures <- function()
{
phant1 <- read.csv(file="hip_stats1.csv",header=TRUE,sep=",")
diameter1 <- ((phant1[2:15, 10])*2)

# Plot the numbers 1-14 (on x) against the diameter (on y)
# choose labels on the x and y axis
# choose limits for the x and y axis
# choose a main and sub title
# choose a plotting type – lines “l”, symbols “p”, or both “b”
# choose a symbol type – a number indicates a built in symbol
# or one can indicate a symbol by pch="sym", e.g., pch="ö"
# choose a line type – a number of line types are available by number

plot(c(1:14),diameter1,xlab="Individual Scans",ylab="Diameter in mm",
+ylim=c(54.18, 54.27), xlim=c(0,15),main="Acetabular Cup Diameter",
+sub="Experimental Data", type="b",pch=7, lty=1,axes=F)
```

Add Labels to the Points

```
# load library to plot labels
```

```
library(plotrix)
```

```
# Get labels
```

```
plotlabels <- phant1$labelr[2:15]
```

```
# plot labels
```

```
thigmophobe.labels(c(1:14),
```

```
diameter1, plotlabels, col="darkblue", font=2)
```

```
# label color is darkblue
```

```
# label font is bold
```

Add Another Plot to This One

```
phant1a <-  
  read.csv(file="hip_stats1a.csv",header=TRUE,sep=",")  
diameter1a <- ((phant1a[2:15, 10])*2)  
  
# Plot  
# Note: different symbol and different line type  
points(c(1:14), diameter1a, type="b", pch=9, lty=2)  
  
# Get labels  
plotlabels <- phant1a$labelr[2:15]  
  
# plot labels  
thigmophobe.labels(c(1:14), diameter1a, plotlabels, col="darkgreen",  
  +font=2)  
# continue adding as many plots as wanted  
# note that one can minutely control every aspect of a plot  
# use 'help(par)' for all the gory details
```

Do Some Statistics and Add to Plot

do mean and SD *2

```
total_cup <- c(diameter1, diameter1a, ...)
meanc <- mean(total_cup)
medianc <- median(total_cup)
SD <- sqrt(var(total_cup))
SD2 <- SD * 2
meanplus <- meanc + SD2
meanminus <- meanc - SD2
```

add to plot

```
ylmean<-meanc + 0.003
text(0.2,ylmean,"Mean", srt=0, crt=0)
points(c(0:41),rep(meanc,42),type="l", lty = 1)

ylmedian<-medianc - 0.003
text(0.4,ylmedian,"Median", srt=0, crt=0)
points(c(0:41),rep(medianc,42),type="l", lty = 1)

ylup <- meanplus + 0.004
Text(1.3,ylup,"Mean Plus 2SD", srt=0, crt=0)
points(c(0:41),rep(meanplus,42),type="l", lty = 1)

yldn <- meanminus + 0.004
text(1.5,yldn,"Mean Minus 2SD", srt=0, crt=0)
points(c(0:41),rep(meanminus,42),type="l", lty = 1)
```


Finish Plot

```
# fix the axes and tick marks
```

```
# first draw a box
```

```
box()
```

```
# Now fix the x-axis indicated by "1"
```

```
# indicate where to draw the tick marks
```

```
# indicate the labels to be used
```

```
# indicate the orientation of the labels – parallel, horizontal,
```

```
# perpendicular, vertical,
```

```
axis(1, at=c(0:15), labels=c(0:15), las=1)
```

las = 1 sets orientation parallel

```
# Now fix the y-axis
```

```
axis(2, at=seq(54.18, 54.27, 0.01),
```

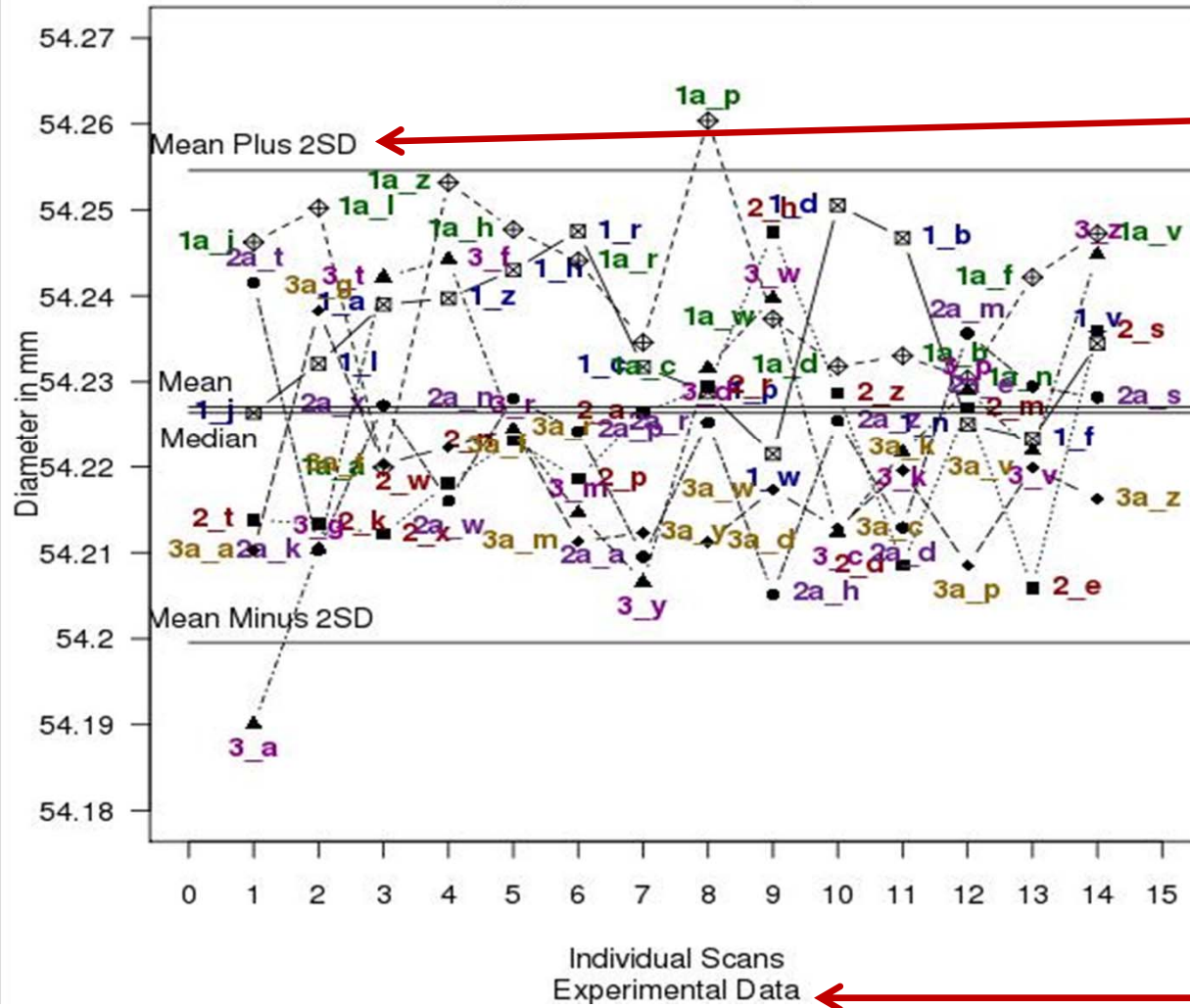
```
  +labels=round(seq(54.18, 54.27, 0.01), digits=2), las=2)
```

las = 2 sets orientation horizontal

Example Finished Plot

Acetabular Cup Diameter - Hip Phantom Scan Series:
One Trial One, One Trial Two, Two Trial One,
Two Trial Two, Three Trial One, Three Trial Two

← main title



← y lup Text

← subtitle

Figure Legends

For some plots it might be necessary to add a legend. This can be placed inside or outside the actual plot. The format of a legend can be:

place legend at x,y where these coordinates are derived from the graph

```
legend(x=tmp.u[1], y=tmp.u[4], legend=list("Scan Series One - Trial One", "Scan Series One - Trial +Two", "Scan Series Two - Trial One", "Scan Series Two - Trial Two", "Scan Series Three - Trial +One", "Scan Series Three - Trial Two"), pch=c(7,9,15,16,17,18))
```

break the above legend into two pieces and place outside the graph

```
legend(x=0.0, y=54.14, legend=list("Scan Series One - Trial One", "Scan Series One - Trial Two", "Scan Series Two - Trial One"), pch=c(7,9,15))
```

```
legend(x=8.0, y=54.14, legend=list("Scan Series Two - Trial Two", "Scan Series Three - Trial One", "Scan Series Three - Trial Two"), pch=c(16,17,18))
```

place the legend at an interactive point

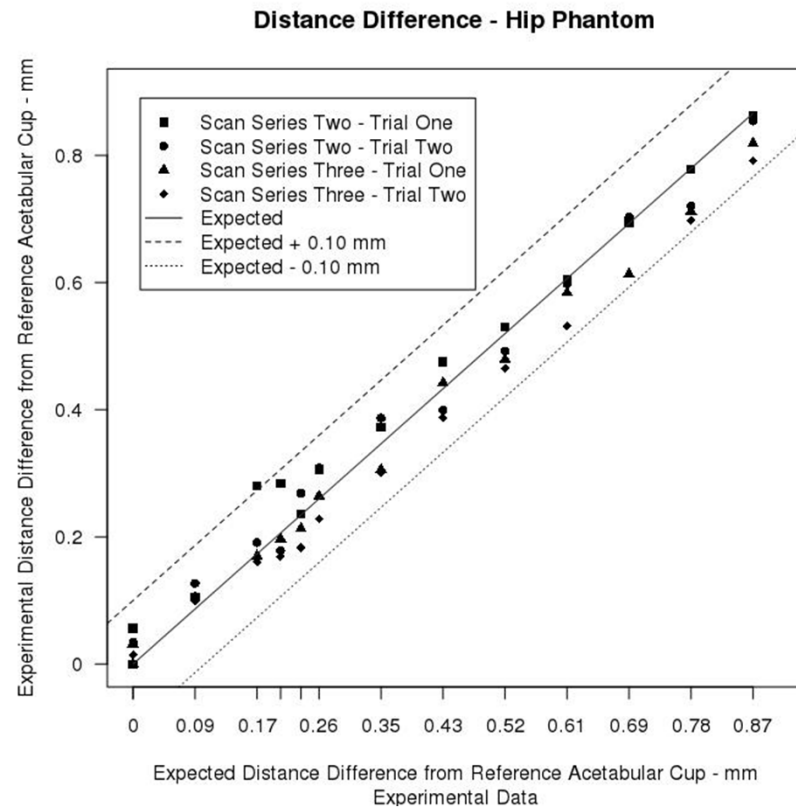
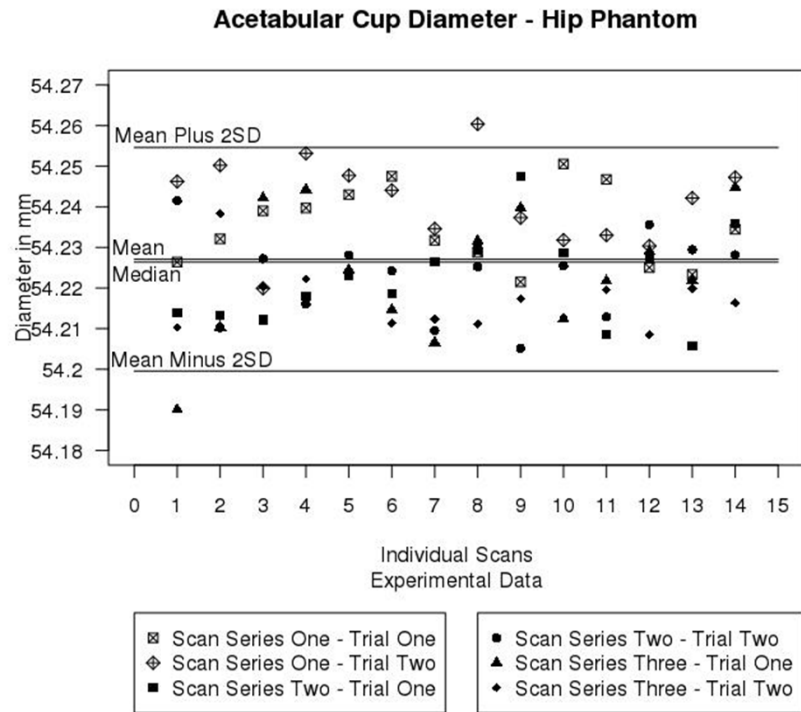
locator reads the position of the graphics cursor when the (first) mouse button is pressed

```
legend(locator(), legend=list("Scan Series One - Trial One", "Scan Series One - Trial Two", "Scan +Series Two - Trial One", "Scan Series Two - Trial Two", "Scan Series Three - Trial One", "Scan +Series Three - Trial Two"), pch=c(7,9,15,16,17,18))
```

use lines and points in graph and indicate which is which:

```
legend(x=0.01, y = 0.89, legend=list("Scan Series Two - Trial One", "Scan Series Two - Trial Two", "Scan Series Three - Trial One", "Scan Series Three - Trial Two", "Expected", "Expected + 0.10 +mm", "Expected - 0.10 mm"), lty=c(-1,-1,-1,-1,1,2,3), pch=c(15,16,17,18,-1,-1,-1))
```

Example Plots



Remarks

Notice that in the previous set of slides, the example functions were just a set of functions which already existed in R.

It is convenient to work in an editor like emacs, try things out, find all the components needed to do the job and then save the set as an R function (e.g., `cup.measures`).

Error bars

Why show error bars?

- To convey to the viewer the expected range of values that might be expected
- Between the whiskers is the total **confidence interval** (CI) within which you are working:
 - This might be: 90%, 95%, or 99%
 - These correspond to 10%, 5%, and 1% probability that the true value is **outside** this range

Error Bars in R

Use the package “gplots”

Reference manual “gplots.pdf” gives instructions for using plotCI - also available from help(plotCI) after the package has been loaded.

CI = confidence interval

For a good set of example code with plots drawn – the plots are at the end – see

http://rgm2.lab.nig.ac.jp/RGM2/func.php?rd_id=plotrix:plotCI

Example of Error Bars in R – read in data and format for finding CI

```
# error in distance difference in scans 2 and 3 (both trials):
# normal distribution
# error difference from expected

errorbars1 <- function() (create simple function)
{
library(gplots)

expected1 <- read.csv(file="Hip-phantom-scan-procedure-series-
1-2-3a.csv",header=TRUE,sep=",");
phant2 <- read.csv(file="hip_stats2.csv",header=TRUE,sep=",");
phant2a <-
  read.csv(file="hip_stats2a.csv",header=TRUE,sep=",");
phant3 <- read.csv(file="hip_stats3.csv",header=TRUE,sep=",");
phant3a <-
  read.csv(file="hip_stats3a.csv",header=TRUE,sep=",");
```

```
x1 <- expected1[3:15,9] - phant2[3:15, 20]
x2 <- expected1[3:15,9] - phant2a[3:15, 20]
x3 <- expected1[3:15,9] - phant3[3:15, 20]
x4 <- expected1[3:15,9] - phant3a[3:15, 20]
```

First make an x5 that exists as a vector

```
x5<-c(1:28)
x5[1:7] <- x1[1:7]
x5[8:14] <- x2[1:7]
x5[15:21] <- x3[1:7]
x5[22:28] <- x4[1:7]

print(x5)
```

Error Bars in R – find 99% CI

```
meanex <- mean(x5)
print("mean")
print(meanex)
SDex <- (sqrt(var(x5)))
print("SD")
print(SDex)
upperCI <- meanex +(2.58*SDex/(sqrt(length(x5))))
print("upperCI")
print(upperCI)
lowerCI <- meanex - (2.58*SDex/(sqrt(length(x5))))
print("lowerCI")
print(lowerCI)
totalCI <- upperCI - lowerCI
print("totalCI")
print(totalCI)
```

Error Bars in R – Plot Graph with error bars

```
plot(expected1[3:15,9], x1, type="b", ylab="Error in Experimental Distance Difference from  
Reference Acetabular Cup - mm", xlab="Expected Distance Difference from  
Reference Acetabular Cup - mm", ylim=c(-0.1, 0.1), xlim=c(0.0,0.9), main="Error in  
Distance Difference - Hip Phantom Scans",sub="Experimental Data", axes=F,pch=15,  
lty=1)
```

```
points(expected1[3:15,9], x2, type="b",pch=17, lty=2)
```

```
points(expected1[3:15,9], x3, type="b",pch=17, lty=3)
```

```
points(expected1[3:15,9], x4, type="b",pch=18, lty=4)
```

```
plotCI(expected1[3:15,9], x1, totalCI, pch=21, pt.bg=par("bg"), add=TRUE)
```

```
plotCI(expected1[3:15,9], x2, totalCI, pch=21, pt.bg=par("bg"), add=TRUE)
```

```
plotCI(expected1[3:15,9], x3, totalCI, pch=21, pt.bg=par("bg"), add=TRUE)
```

```
plotCI(expected1[3:15,9], x4, totalCI, pch=21, pt.bg=par("bg"), add=TRUE)
```

```
par(mfrow = c(1, 1))
```

```
# Note for docs on plotCI see gplots.pdf and web site given above
```

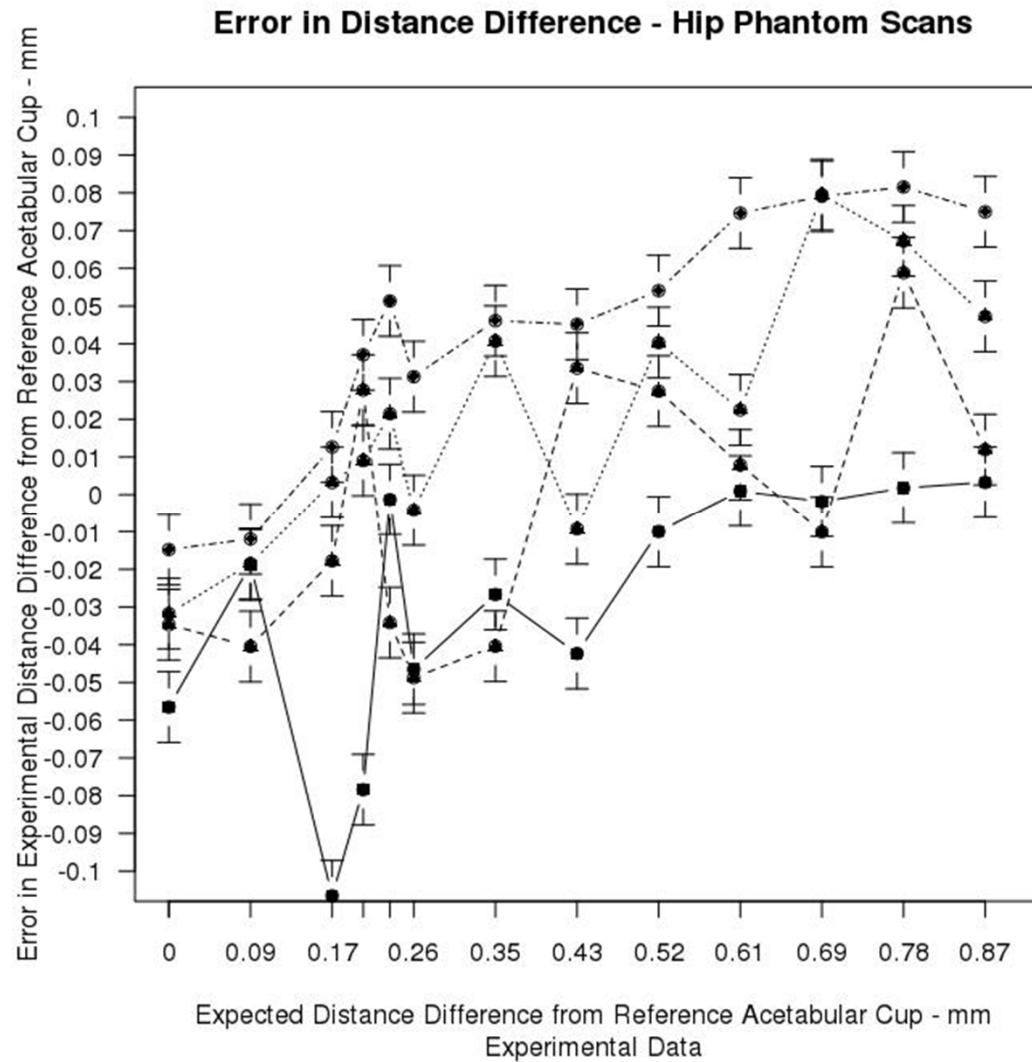
```
box()
```

```
axis(1, at=expected1[2:15,9], labels=round(expected1[2:15,9],digits=2), las=1)
```

```
axis(2, at=seq(-0.1, 0.1, 0.01),labels=round(seq(-0.1, 0.1, 0.01), digits=2), las=2)
```

```
}
```

Error Bars in R – Resulting Plot



References

- [1] Tom Tullis and Bill Albert, "Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics", Morgan-Kaufmann, 2008, ISBN 978-0-12-373558-4
- [2] R Graphics Gallery, <http://gallery.r-enthusiasts.com/>
- [3] Hadley Wickham, ggplot2: Elegant Graphics for Data Analysis (Use R), Springer; 2nd Printing. August 7, 2009, 216 pages, ISBN-10: 0387981403 and ISBN-13: 978-0387981406, website for the book: <http://had.co.nz/ggplot2/book/>
- [4] Hadley Wickham, website of Hadley Wickham, Rice University, Houston TX, USA, 2010, last accessed Wed 15 Sep 2010 04:51:27 PM CEST, <http://had.co.nz/>
- [5] Dong-Yun Kim, "MAT 356 R Tutorial, Spring 2004", web page, Department of Mathematics, Illinois State University, Normal, IL, USA, last modified: 14 January 2004 07:51:38 AM CET, <http://math.illinoisstate.edu/dhkim/rstuff/rtutor.html>
- [6] Frank McCown, Producing Simple Graphs with R, web page, Computer Science Department, Harding University, Searcy, AR, USA, last modified: 06/08/2008 01:06:21, <http://www.harding.edu/fmccown/r/>
- [7] Michael Wexler, R GUIs, web page, last modified Wed 08 Sep 2010 05:02:06 PM CEST, <http://www.nettakeaway.com/tp/?s=R> (VP of Web Analytics at Barnes and Noble.com)
- [8] Dennis R. Mortensen, Yahoo! Web Analytics 9.5 Launched. Visual.revenue blog, New York City, Tuesday, April 28, 2009, <http://visualrevenue.com/blog/2009/04/yahoo-web-analytics-95-launched.html>
- [9] Julian J. Faraway, "Linear Models with R" Chapman & Hall/CRC Texts in Statistical Science, 2005, 242 pages, ISBN 0-203-50727-4
- [10] Dov Goldvasser, Marilyn E Noz, G Q Maguire Jr., Henrik Olivecrona, Charles R Bragdon, and Henrik Malchau, 'A New Technique for Measuring Wear in Total Hip Arthroplasty Using Computed Tomography', The Journal of arthroplasty, May 2012, DOI:10.1016/j.arth.2012.03.053, Available at <http://www.ncbi.nlm.nih.gov/pubmed/22658429>.

¿Questions?