II2202: Quantitative tools: R and more

G. Q. Maguire Jr.
&
Marilyn E. Noz

2011.09.19

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

Resources

Get it from: 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
- Emmanuel Paradis, **"R for Beginners"** **http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf**
- …

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

$\Rightarrow$ at least: 107,505 RTP packets in each direction

$\Rightarrow$ 429 RTCP packets in one direction

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)

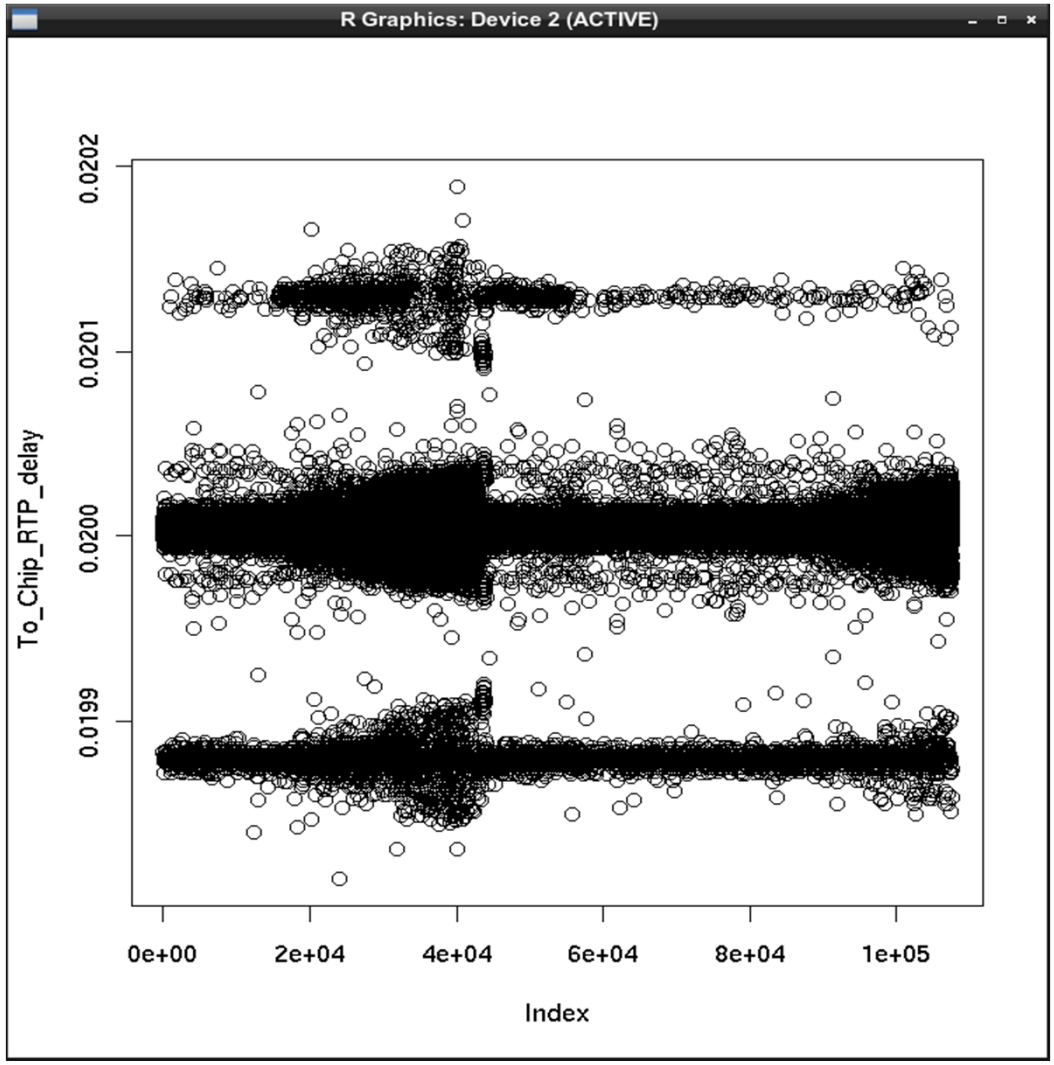
```
   No.                 Time                      Source
Min.   :  1443      Min.   :17686          90.226.255.70 :107515
1st Qu.: 55331      1st Qu.:18223          217.211.xx.xx :     0
Median :109224      Median :18761          41.209.78.223 :     0
Mean   :109223      Mean   :18761          62.20.251.42  :     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

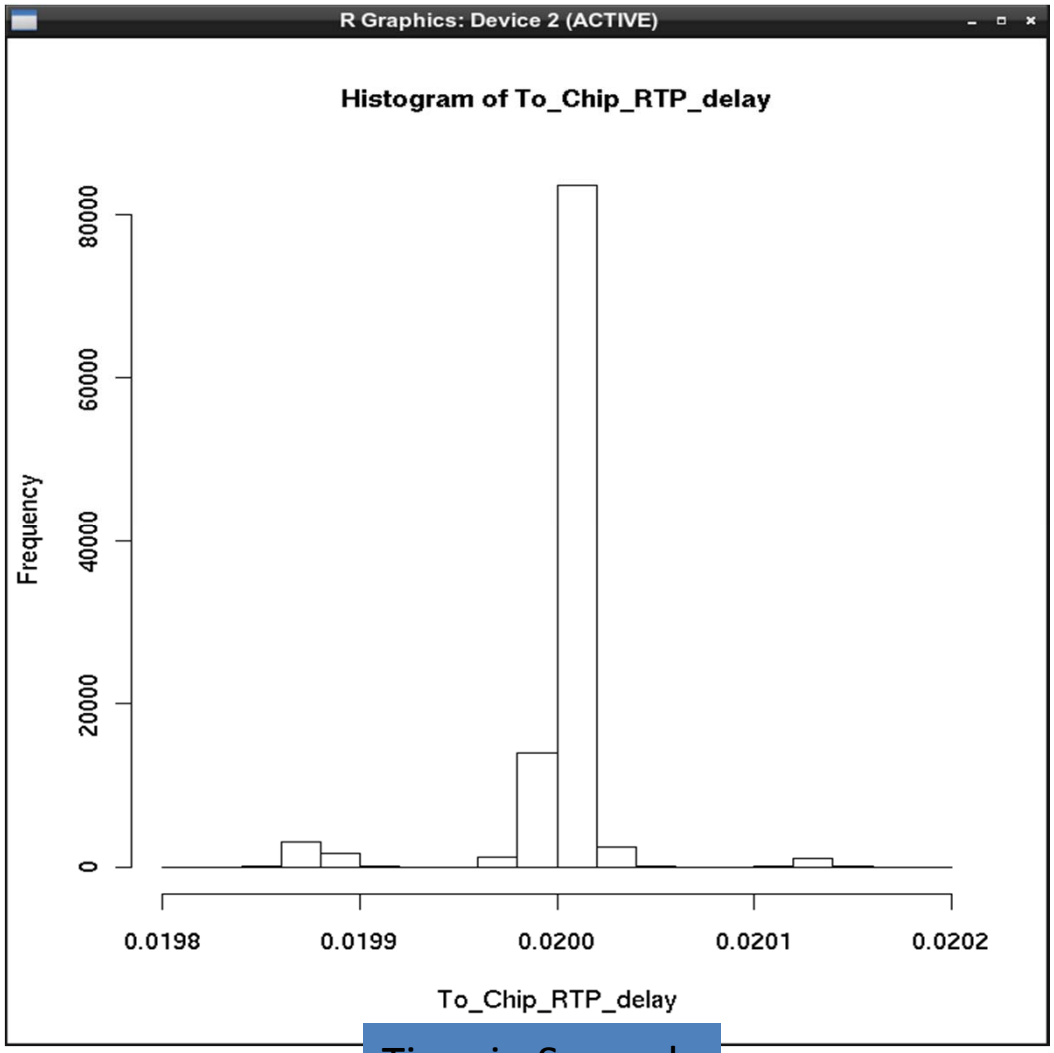
```
lvh<-nrow(To_Chip_RTP)
[1] 107515
lvh<-lvh-1> lvh
[1] 107514
To_Chip_RTP_delay=vector(length=(nrow(To_Chip_RTP)-1))
for (i in 1:lvh) {
To_Chip_RTP_delay[i]<-To_Chip_RTP$Time[i+1]-
   To_Chip_RTP$Time[i]
}
```

```
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)

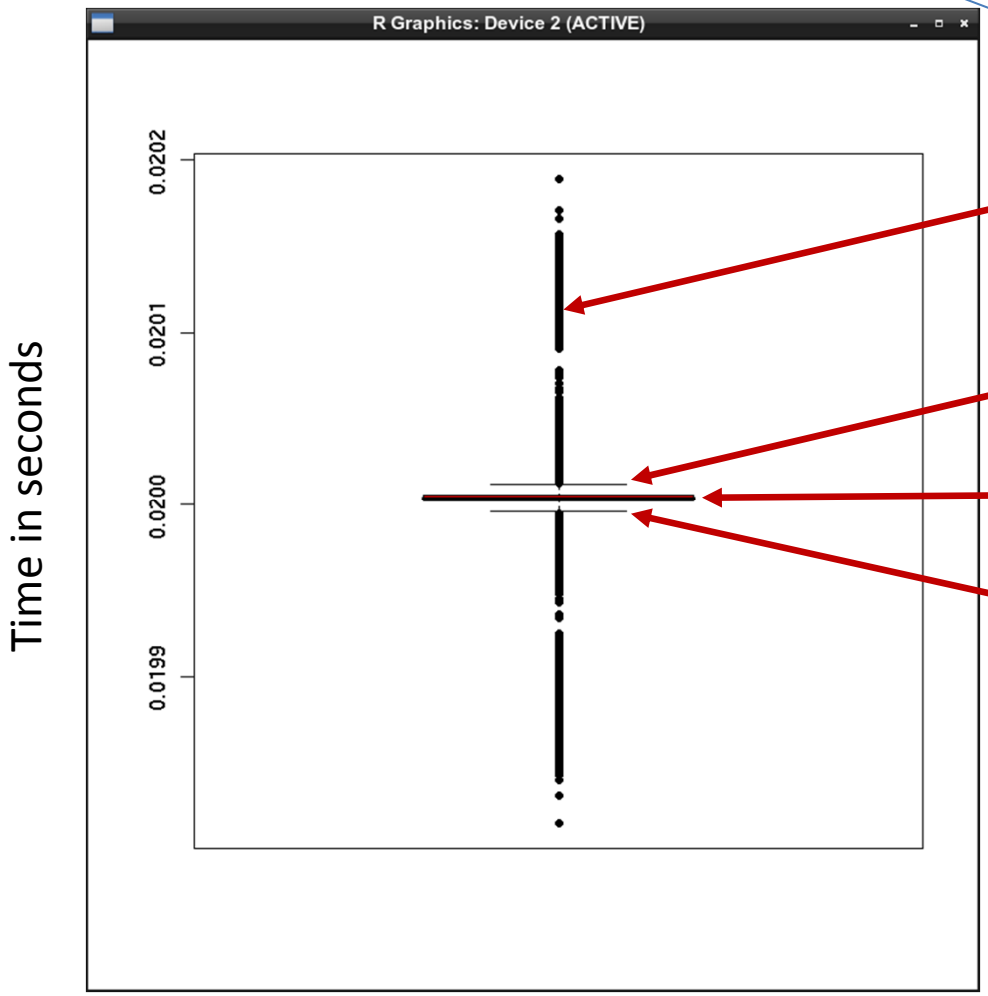


hist(To_Chip_RTP_delay)



Time in Seconds

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



Color 3 = Blue

Symbol = bullett

Outliers

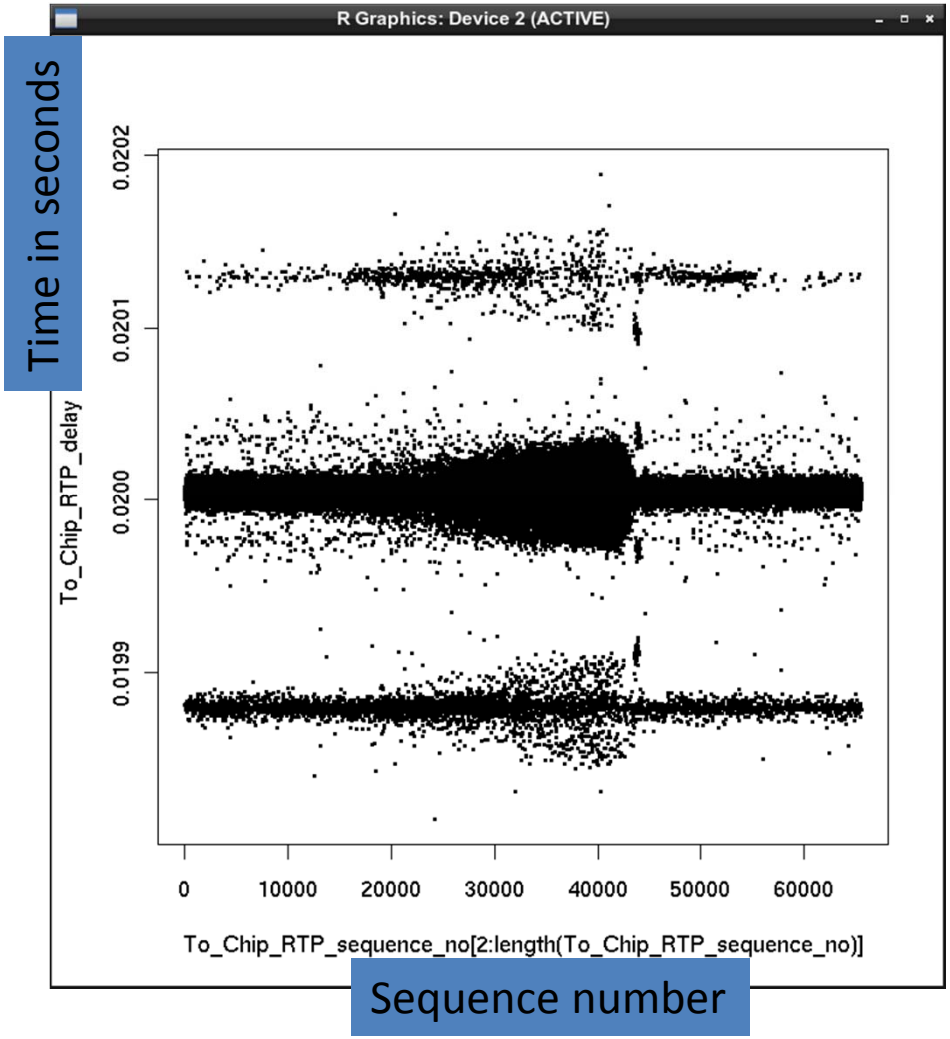
Upper quartile

Median

Lower quartile

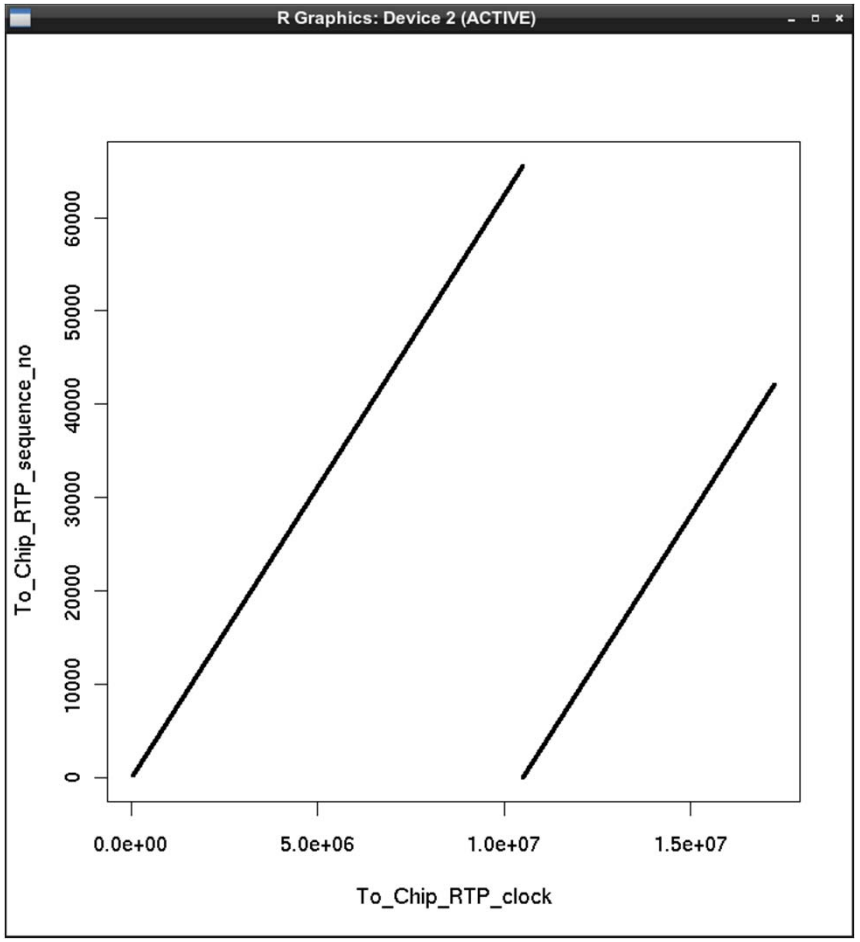
Interarrival delay vs. sequence

```
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:le
    ngth(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)
```



From network to local user agent

Raw output fom Microsoft Excel 2010 (Beta)

Inter-arrival times of RTP packets

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

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]: "0.020004999984913“

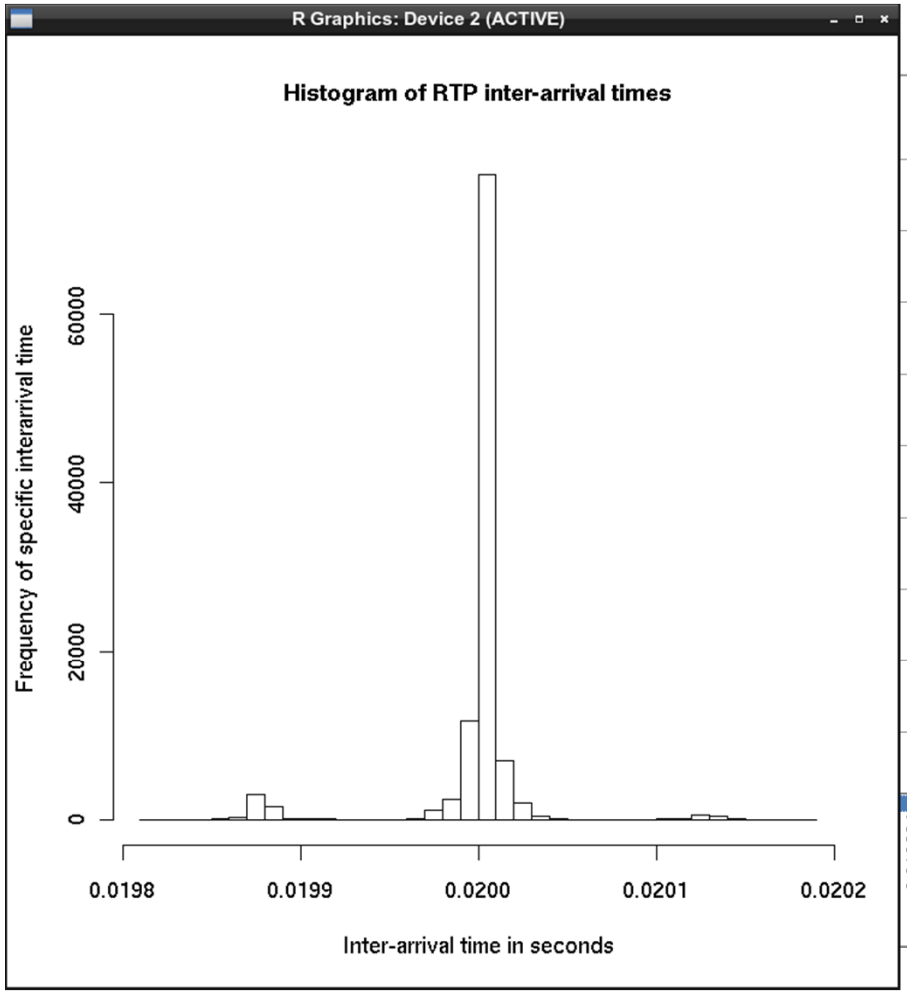
sd(To_Chip_RTP_delay): 3.044357e-05
var(To_Chip_RTP_delay): 9.268109e-10

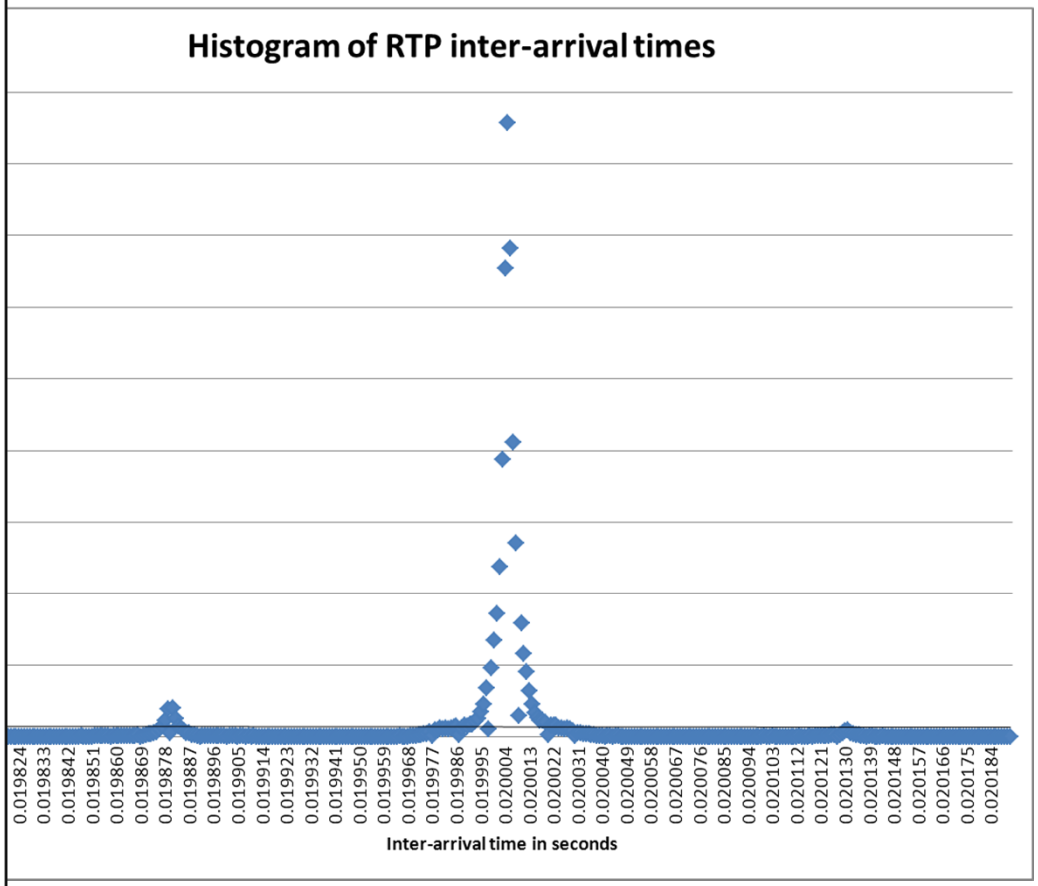
library(moments);
kurtosis(To_Chip_RTP_delay): 15.36689
skewness(To_Chip_RTP_delay): -2.054706

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

http://www.r-tutor.com/elementary-statistics/numerical-measures/kurtosis
http://www.r-tutor.com/elementary-statistics/interval-estimation/interval-estimate-population-proportion

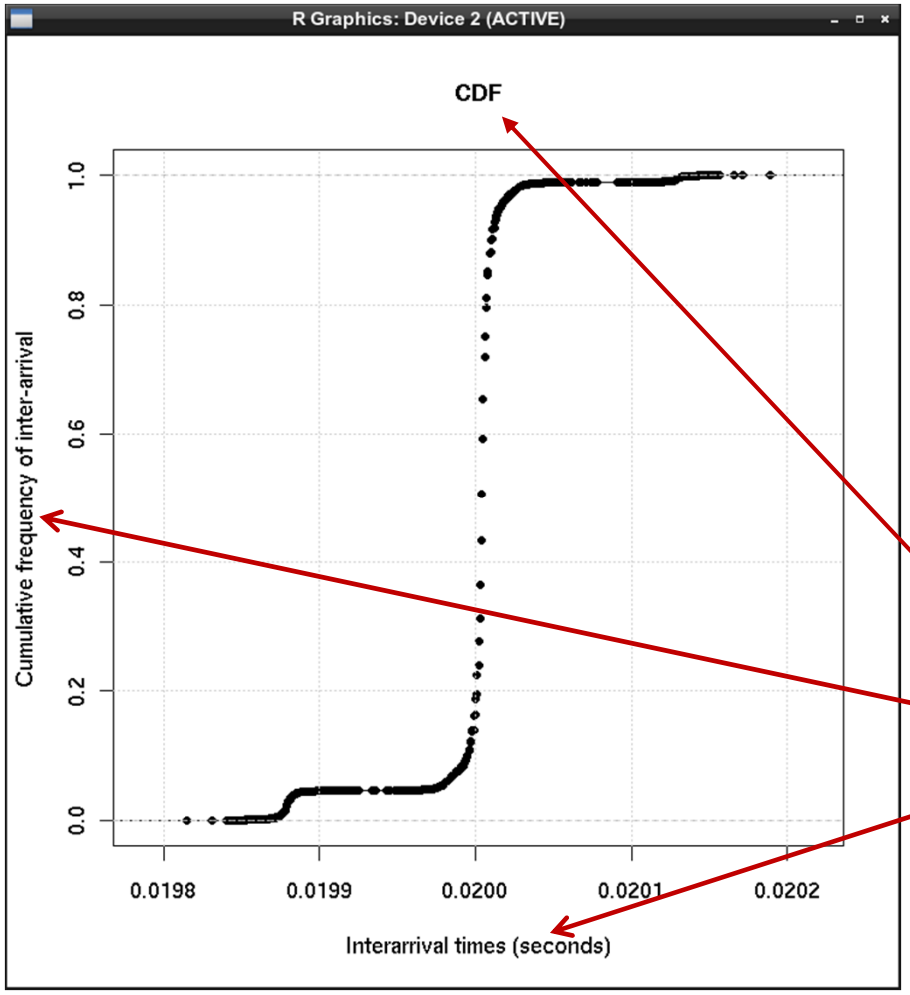
R vs. Excel histogram






```
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)
```

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				02
Standard Error				08
Median				04
Mode				05
Standard Deviation				05
Sample Variance				.0
Kurtosis				08
Skewness				75
Range				74
Minimum				.5
Maximum				39
Sum				51
Count				00
Confidence Level(95.0%)				07

```
foo<-function(n){
v <-1:12
v[1]=mean(To_Chip_RTP_delay[1:n])
v[2]=std.error(To_Chip_RTP_delay[1:n])
v[3]=names(sort(-table(To_Chip_RTP_delay[1:n])))[1]
v[4]=sd(To_Chip_RTP_delay[1:n])
v[5]=var(To_Chip_RTP_delay[1:n])
v[6]=kurtosis(To_Chip_RTP_delay[1:n])
v[7]=skewness(To_Chip_RTP_delay[1:n])
v[8]=min(To_Chip_RTP_delay[1:n])
v[9]=max(To_Chip_RTP_delay[1:n])
v[10]=sum(To_Chip_RTP_delay[1:n])
v[11]=length(To_Chip_RTP_delay[1:n])
v[12]=qnorm(0.965)*std.error(To_Chip_RTP_delay[1:n])
return(v)}
seq1<-c(foo(100),foo(1000),foo(10000),foo(100000))
mat1<-matrix(seq1,  ncol=4)
```

Applying a function to a list of arguments

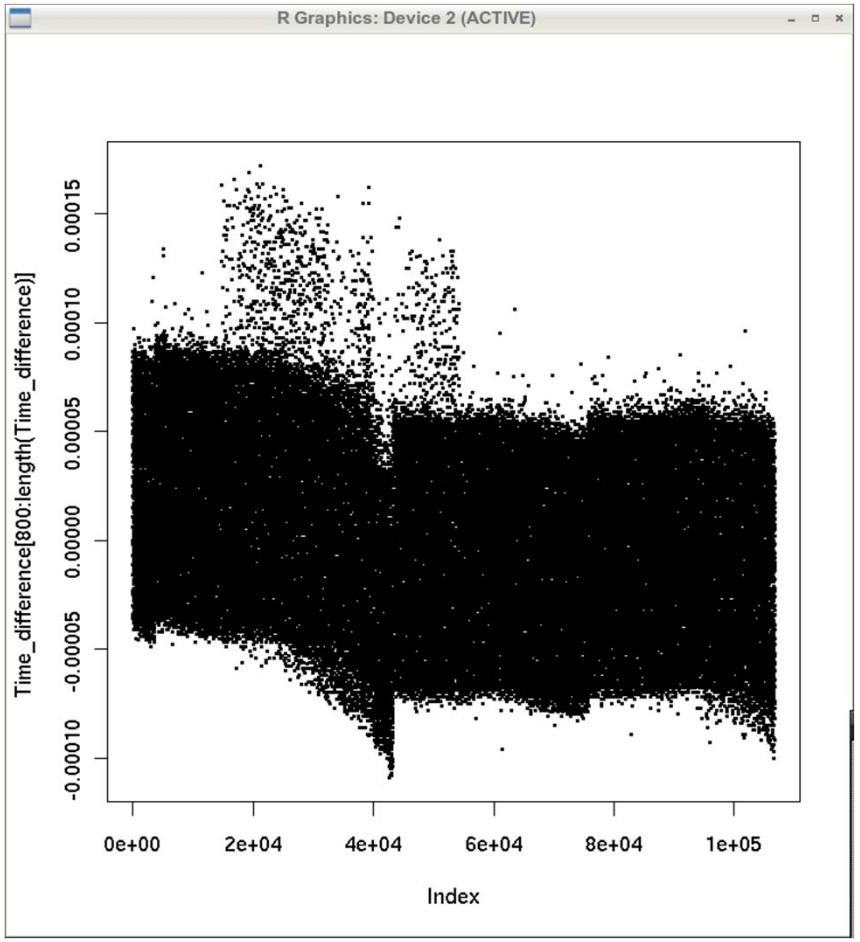
Descriptive Statistics	First 100	First 1K	First 10K	First 100K
Mean				
Standard Error				
Median				
Mode				
Standard Deviation				
Sample Variance				
Kurtosis				
Skewness				
Range				
Minimum				
Maximum				
Sum				
Count				
Confidence Level(95.0%)				

```
foo<-function(m,n){v <- 1:12
v[1]=mean(m[1:n])
v[2]=std.error(m[1:n])
v[3]=names(sort(-table(m[1:n])))[1]
v[4]=sd(m[1:n])
v[5]=var(m[1:n])
v[6]=kurtosis(m[1:n])
v[7]=skewness(m[1:n])
v[8]=min(m[1:n])
v[9]=max(m[1:n])
v[10]=sum(m[1:n])
v[11]=length(m[1:n])
v[12]=qnorm(0.965)*std.error(m[1:n])
return(v)}

fee<-function(n) {foo(To_Chip_RTP_delay, 10^n)}

lapply(c(2:5), fee)
[[1]] [1] "0.0200006800000119"   "2.12697347407497e-06" "0.0200049999984913“
      [4] "2.12697347407497e-05" "4.52401615941855e-10" "30.3672958382318“ …
```

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

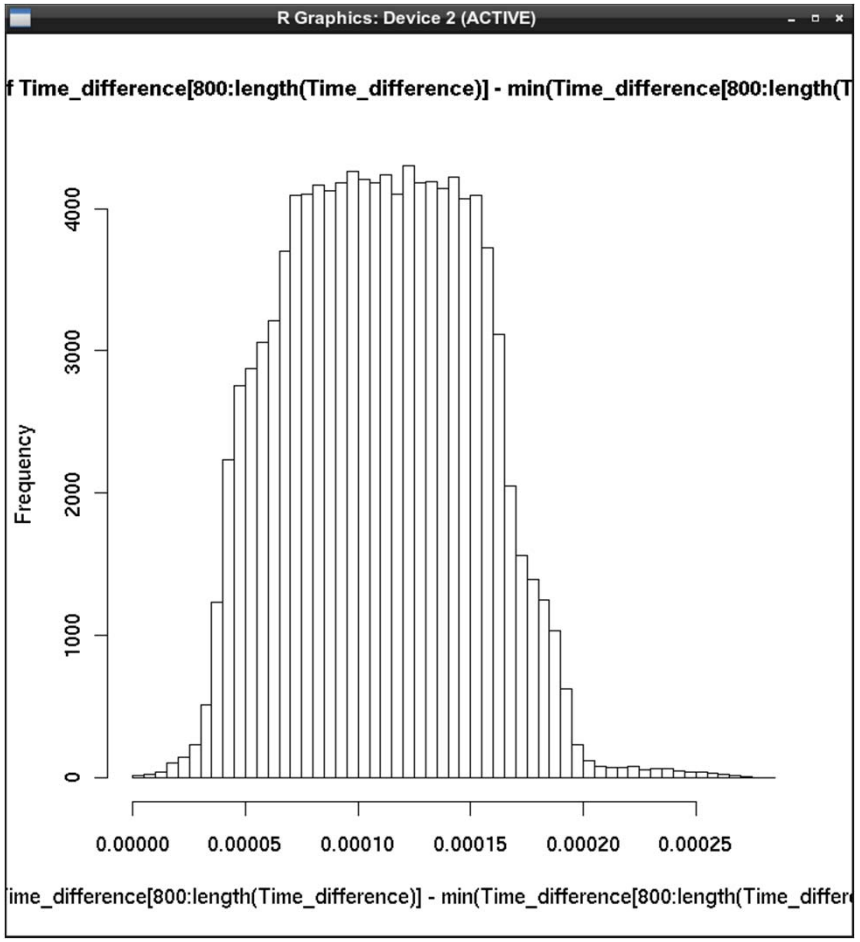


How does the measured data differ from the expected data?

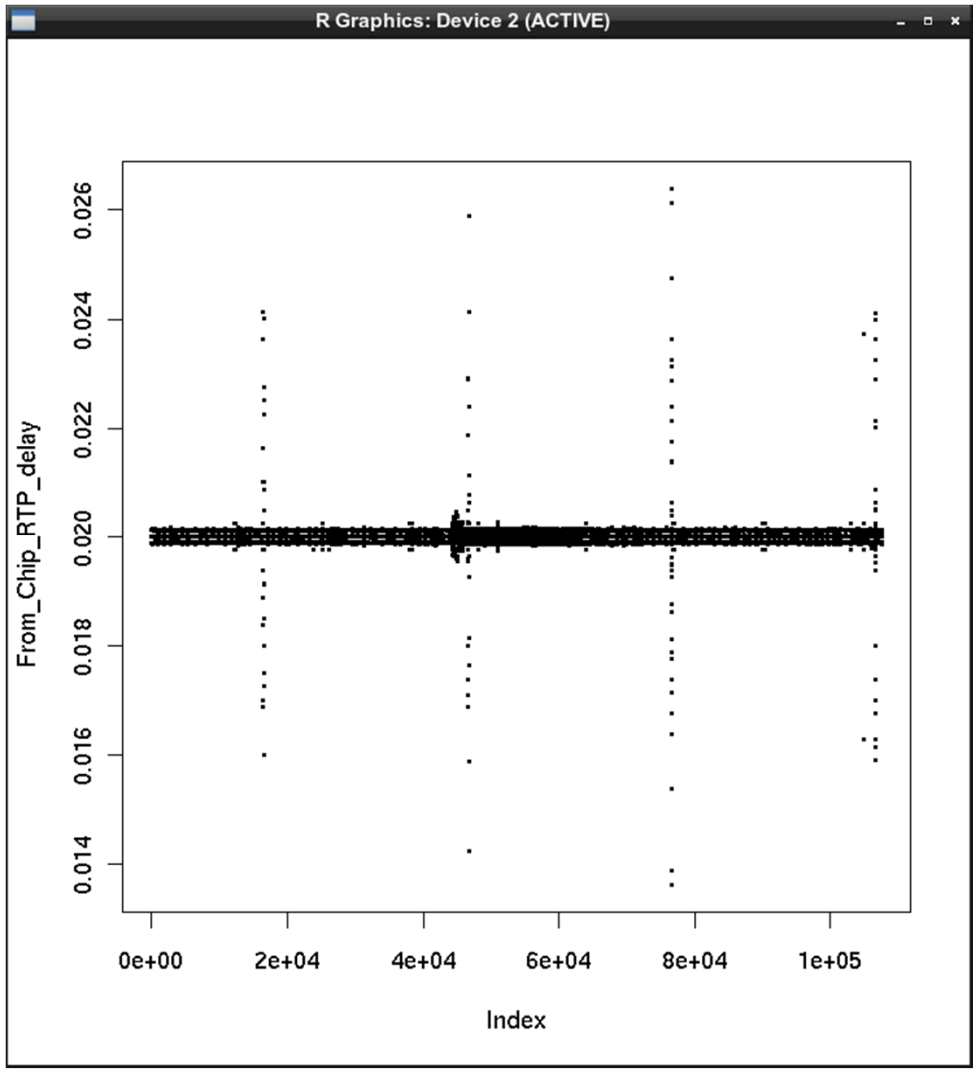
Note that since delay can not be negative, the real difference can be found by adding the min() ⇒

```
hist(
    Time_difference[800:length(Time_
    difference)]-
    min(Time_difference[800:length(Ti
    me_difference)] ),
    breaks=100)
```

Number of bins to use



Uplink inter-arrival times



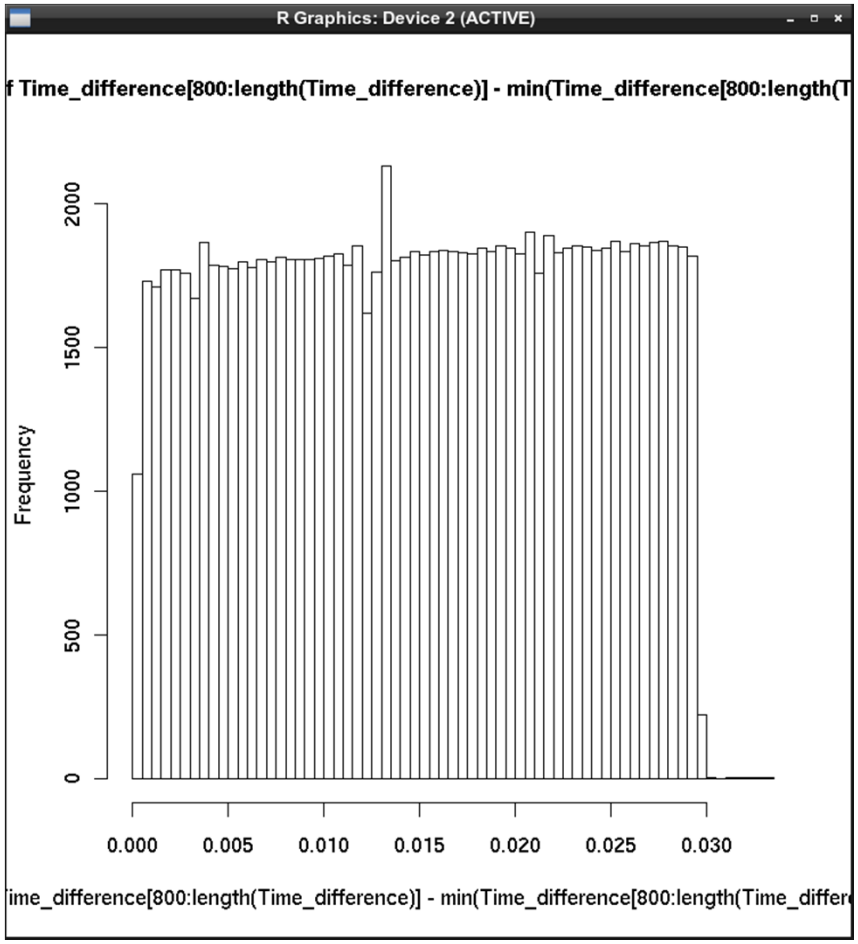
Uplink inter-arrival times stats

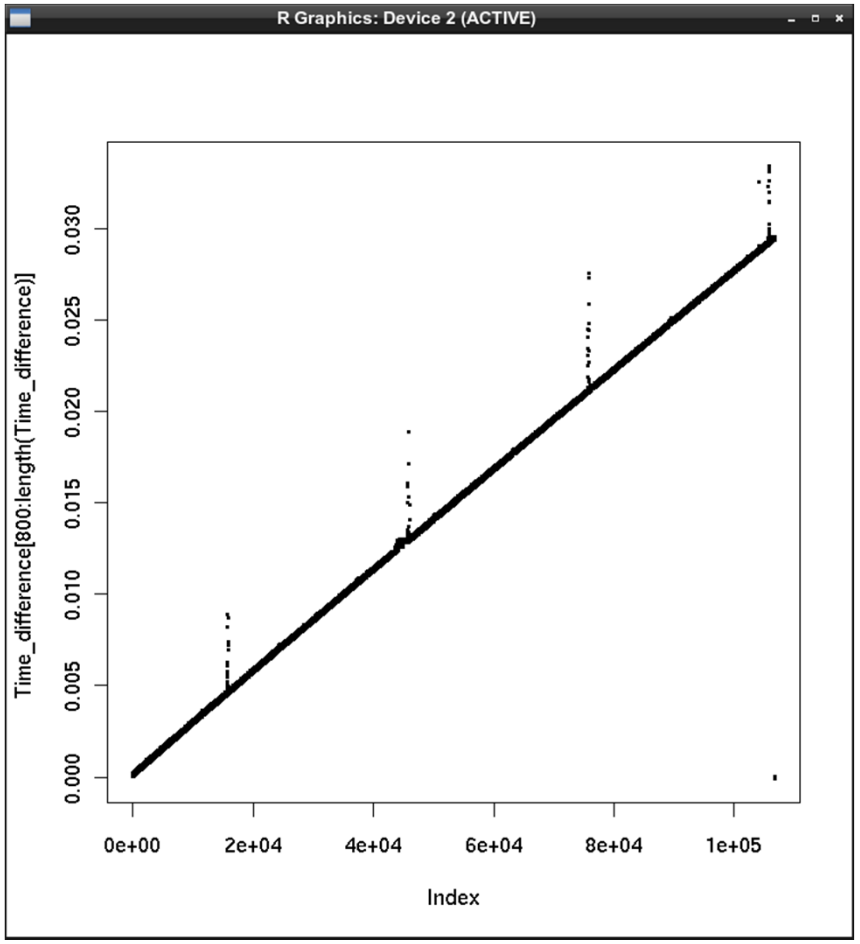
```
library(plotrix);library(moments)foo
<-function(m,n){v <- 1:12
v[1]=mean(m[1:n])
v[2]=std.error(m[1:n])
v[3]=names(sort(-table(m[1:n])))[1]
v[4]=sd(m[1:n])
v[5]=var(m[1:n])
v[6]=kurtosis(m[1:n])
v[7]=skewness(m[1:n])
v[8]=min(m[1:n])
v[9]=max(m[1:n])
v[10]=sum(m[1:n])
v[11]=length(m[1:n])
v[12]=qnorm(0.965)*std.error(m[1:n])
return(v)}
```

foo(From_Chip_RTP_delay, 10^5)	What to put into a report:
"0.0200027577"	0.020000 s
"3.63331229733734e-0	3.63e-07 s
"0.020004999984913"	0.020005 s
"0.000114895423102 8 4	0.000115 s
"1.32009582499827e-0	1.32e-08 s
"742.581556664333"	742.58
"0.63365800721361 5"	0.634
"0.013624999995925"	0.013625 s
"0.026384000006894"	0.026384 s
"2000.027577"	2000.027577 s
"100000"	100000
"6.583237329715 44e-0	6.58e-07 s

Truncated to meaningful number of digits, added units, decimal align the numbers, set in fixed width font (Courier)

For traffic in the opposite direction





Difference histogram and difference plot ⇒ the clock is drifting wrt the Wireshark clock

DNS lookup times

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

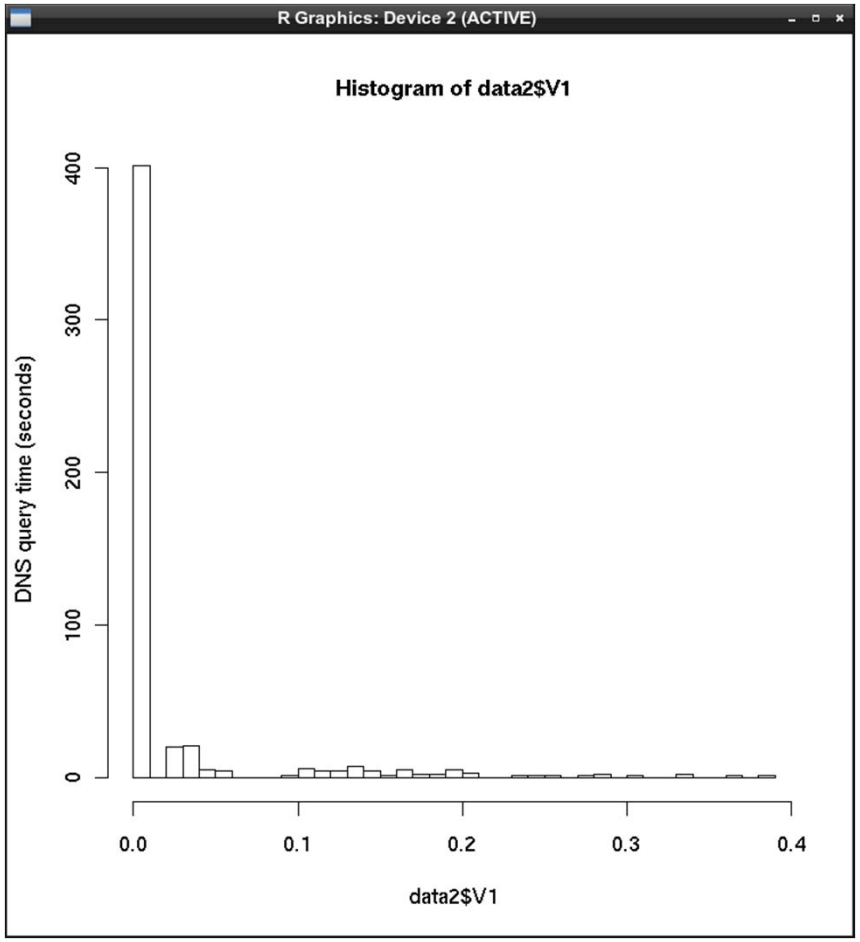
```
   Min.    :0.000710
   1st Qu.:0.000896
   Median :0.001066
   Mean    :0.023868
   3rd Qu.:0.003329
   Max.    :0.389880
```

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

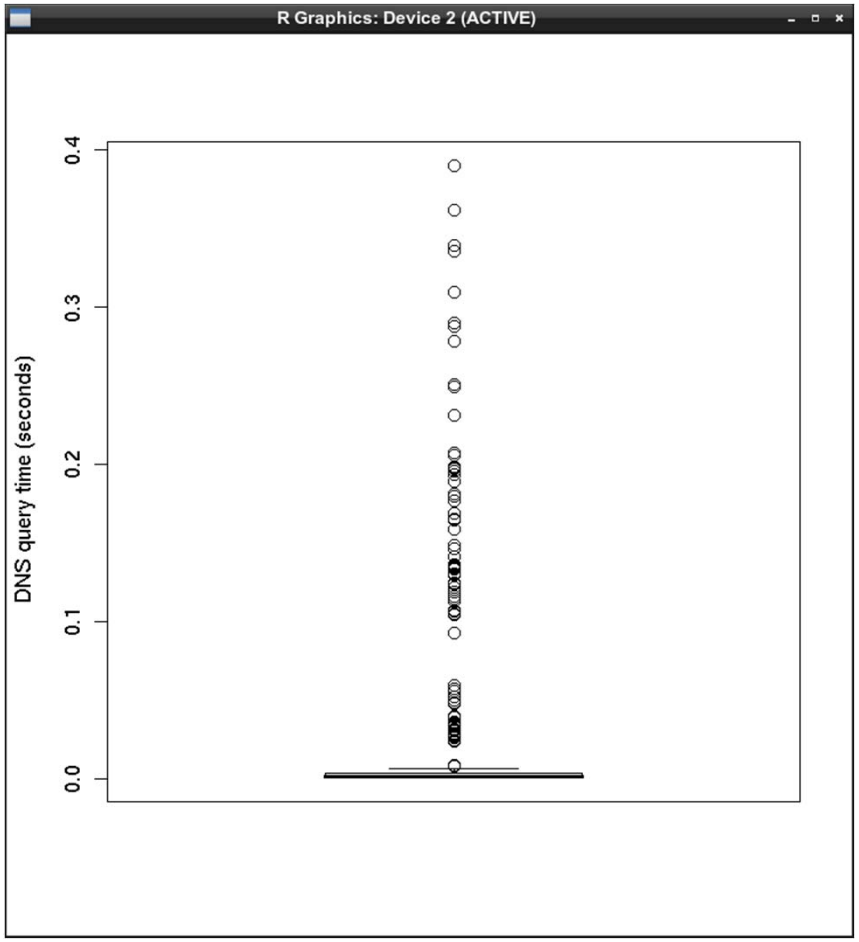
```
Mean:        0.023868 s
std.error:   0.002669 s
Mode:        0.000896 s
Sd:          0.060045 s
Var:         0.003605 s
Kurtosis: 14.3
Skewness:  3.3
Min:         0.00071  s
Max:         0.38988  s
Sum:        12.077197 s
Count:    506
Conf (95%) 0.004837  s
```

DNS lookup time graphs

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



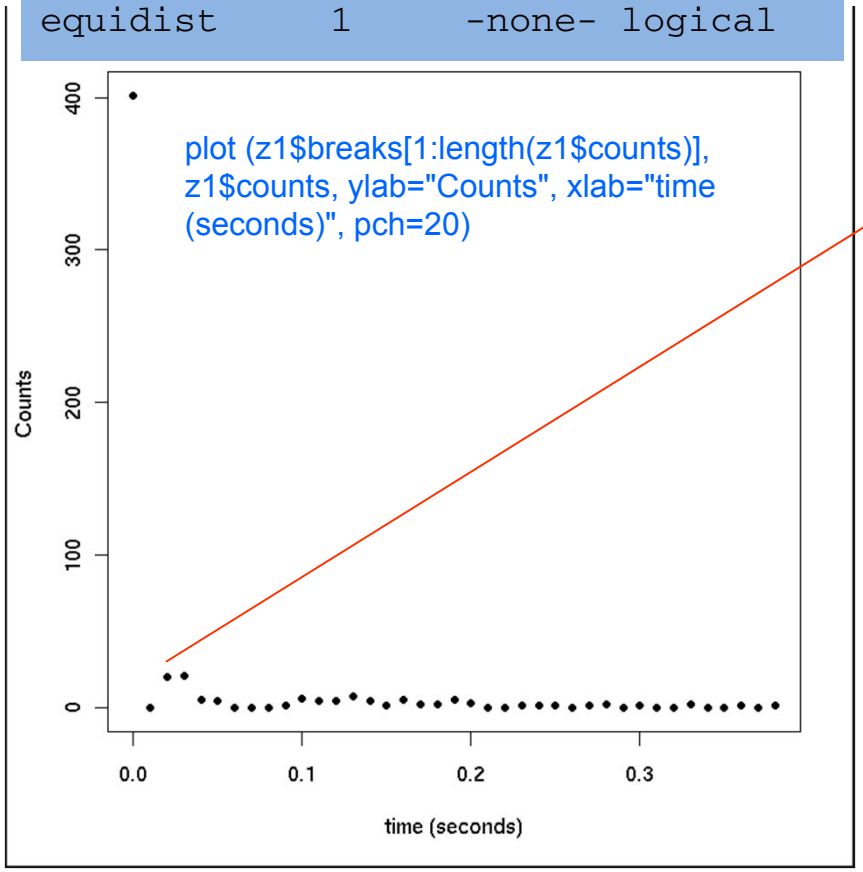
boxplot(data2$V1, ylab="DNS query time (seconds)")



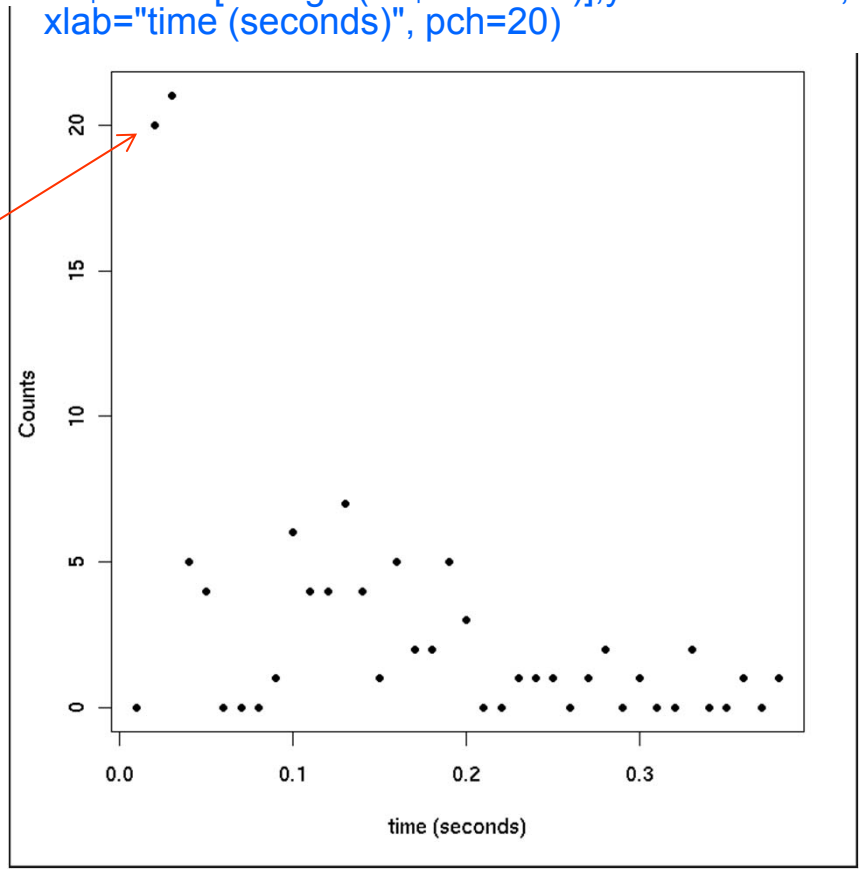
More graphs:
change scale

```
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
```

plot (z1$breaks[1:length(z1$counts)], z1$counts, ylab="Counts", xlab="time (seconds)", pch=20)

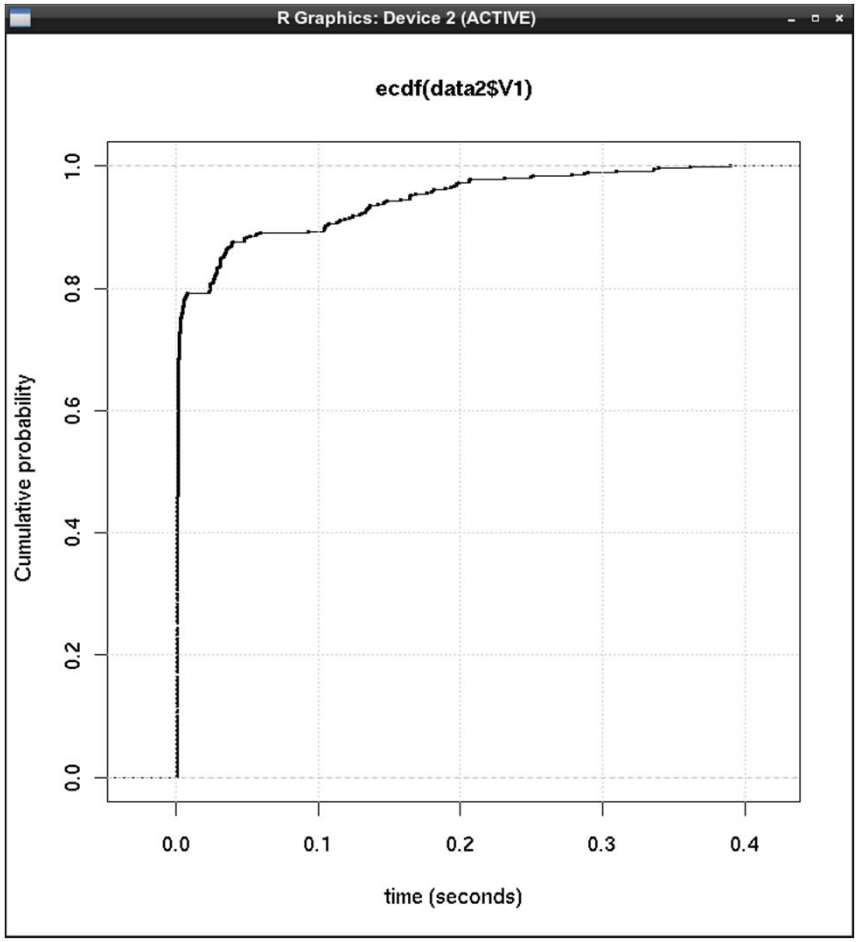


plot (z1$breaks[2:length(z1$counts-1)], z1$counts[2:length(z1$counts-1)],ylab="Counts", xlab="time (seconds)", pch=20)



DNS response CDF

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



Importing Data into R

From a comma separated file:

DataD1 ← read.csv(file=”table.csv”,header=TRUE, ...)

help(read.cvs) for all the options which include reading row names

DataD1 ← read.table(file=”table.cvs”, sep=”,” …)

help(read.table) for all the options

library(gdata) (Note: the package is called gdata – see Packages slide)

DataD4 <- read.xls(“table.xls", sheet=4. ...)

help(read.xls) for all the options

The resulting files are 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
# this just produces 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**

www.r-project.org

About R
What is R?
Contributors
Screenshots
What's new?

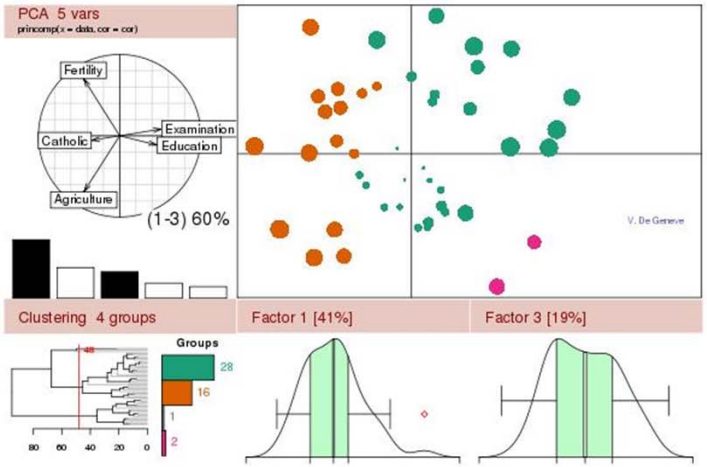
Download, Packages
CRAN

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

The R Project for Statistical Computing



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 2.12.0 prerelease versions** will appear starting September 17. Final release is scheduled for October 15, 2010.
- The R Journal Vol.2/1 is available
- **useR! 2010**, the R user conference, has been held at NIST, Gaithersburg, Maryland, USA, July 21-23, 2010.
- **useR! 2011**, will take place at the University of Warwick, Coventry, UK, August 16-18, 2011.

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
Related Projects
User Groups
Links

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.

Australia		
	http://cran.ms.unimelb.edu.au/	University of Melbourne
Austria		
	http://cran.at.r-project.org/	Wirtschaftsuniversitaet Wien
Belgium		
	http://www.freestatistics.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.parentinginformed.com/	iWeb, Montreal
Chile		
	http://dirichlet.mat.puc.cl/	Pontificia Universidad Catolica de Chile, Santiago
China		
	http://ftp.ctex.org/mirrors/CRAN/	CTEX.ORG
	http://cran.csdb.cn/	Computer Network Information Center, CAS, Beijing
	http://mirrors.geoexpat.com/cran/	GeoExpat.Com
Colombia		
	http://www.laqee.unal.edu.co/CRAN/	National University of Colombia
Denmark		
	http://mirrors.dotsrc.org/cran/	dotsrc.org, Aalborg
France		
	http://cran.cict.fr/	CICT, Toulouse
	http://cran.univ-lyon1.fr/	Dept. of Biometry & Evol. Biology, University of Lyon
	http://mirror.ibcp.fr/pub/CRAN/	CNRS IBCP, Lyon

Choose mirror site near your location, for example:

http://ftp.sunet.se/lang/CRAN

Swedish University Computer Network

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

Frequently used pages

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:

- Linux
- MacOS X
- Windows

Source Code for all Platforms

Windows and Mac users most likely want 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** (2010-05-31): R-2.11.1.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.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN mirror nearest to you to minimize network load.

R for Windows

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 for Windows

This directory contains 32-bit binaries for a base distribution and packages to run on i386/x64 Windows.
See here for a 64-bit Windows port.

Note: CRAN does not have Windows systems and cannot check these binaries for viruses. Use the normal precautions with downloaded executables.

Subdirectories:

base	Binaries for base distribution (managed by Duncan Murdoch)
contrib	Binaries of contributed packages (managed by Uwe Ligges)

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.

Last modified: April 4, 2004, by Friedrich Leisch

R for Macs

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 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) can be found in the old directory.

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

Universal R 2.11.1 released on 2010/05/31

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

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-2.11.1.pkg`
in the *Terminal* application to print the MD5 checksum for the R-2.11.1.pkg image.

Files:

R-2.11.1.pkg (latest version) MD5-hash: ce4de47e58efb9a69573b86ba0cb5b3d (ca. 38MB)	Three-way universal binary of **R 2.11.1** for Mac OS X 10.5 (Leopard) and higher. Contains R 2.11.1 framework, R.app GUI 1.34 in 32-bit and 64-bit. 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.**
Mac-GUI-1.34.tar.gz MD5-hash: c2da4540149ff1c1575da4f074002181d	Sources for the R.app GUI 1.34 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.
NEWS (for Mac GUI)	News features and changes in the R.app Mac GUI

The new R.app Cocoa GUI has been written by Simon Urbanek and Stefano Iacus with contributions from many developers and translators world-wide, see "About R" in the GUI.

Mac Extras

Subdirectories:

tools	Additional tools necessary for building R for Mac OS X: Universal GNU Fortran compiler for Mac OS X (see R for Mac tools page for details).
leopard	Binaries of universal (32-bit and 64-bit) package builds for Mac OS X 10.5 or higher (Leopard build)
universal	Legacy binaries of universal (32-bit) package builds for Mac OS X 10.4 (Tiger build)
old	Previously released R versions for Mac OS X

You may also want to read the R FAQ and R for Mac OS X FAQ. For discussion of Mac-related topics and reporting Mac-specific bugs, please use the R-SIG-Mac mailing list.

Information, tools and most recent **daily builds of the R GUI, R-patched and R-devel** can be found at http://R.research.att.com/. Please visit that page especially during beta stages to help us test the Mac OS X binaries before final release!

Package maintainers should visit CRAN check summary page to see whether their package is compatible with the current build of R for Mac OS X.

Sources for dependencies not present here are available from http://R.research.att.com/src.

Last modified: 2010/05/31, by Simon Urbanek

Linux 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

Home
Search
News
About
Contact

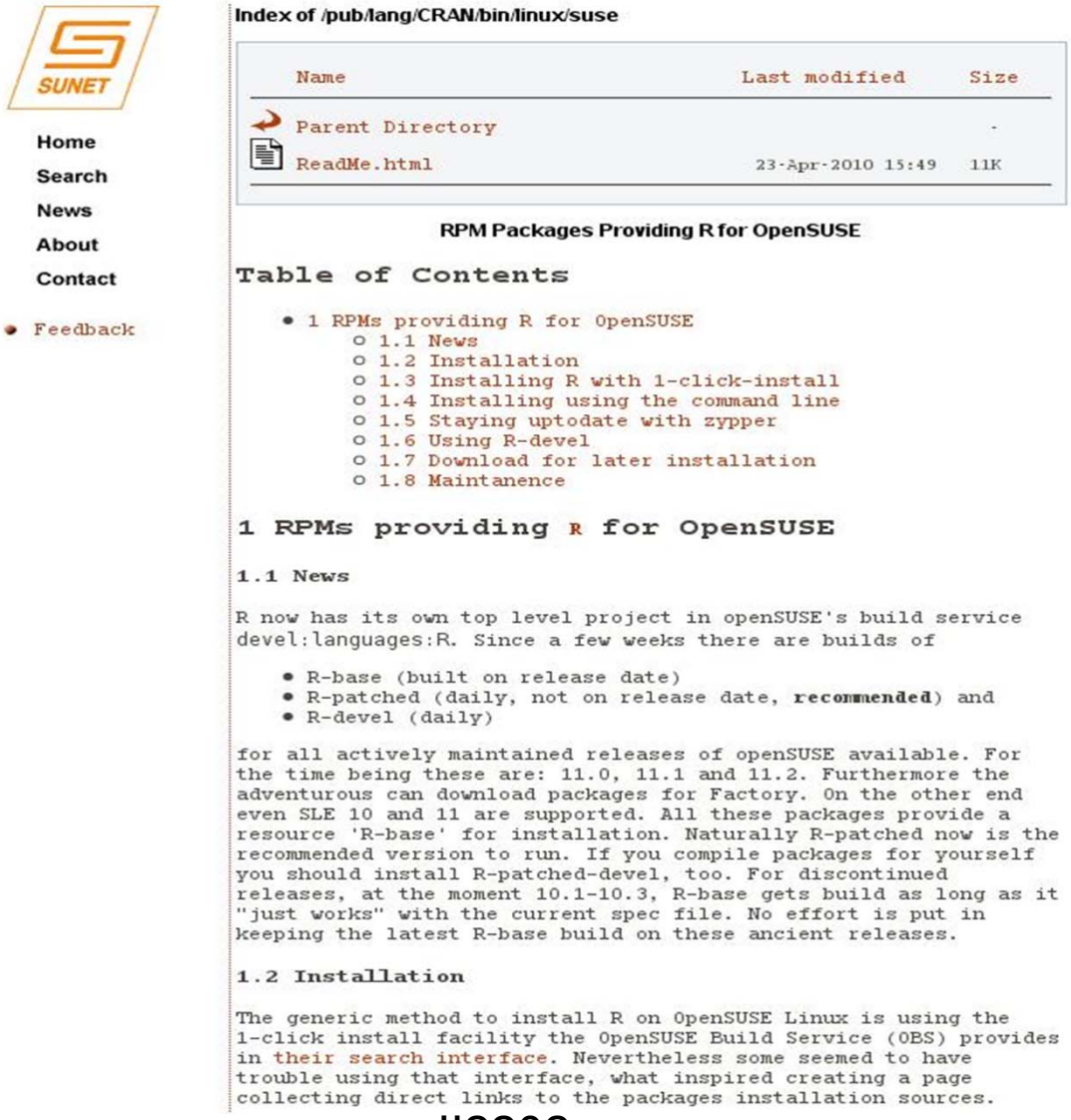
Feedback

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

Name	Last modified	Size
Parent Directory		-
debian/	06-Jun-2010 21:36	-
redhat/	25-Nov-2009 18:01	-
suse/	18-Dec-2009 11:11	-
ubuntu/	05-May-2010 20:41	-

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

For example for SuSE



Installation - SuSE

1.3 Installing R with 1-click-install

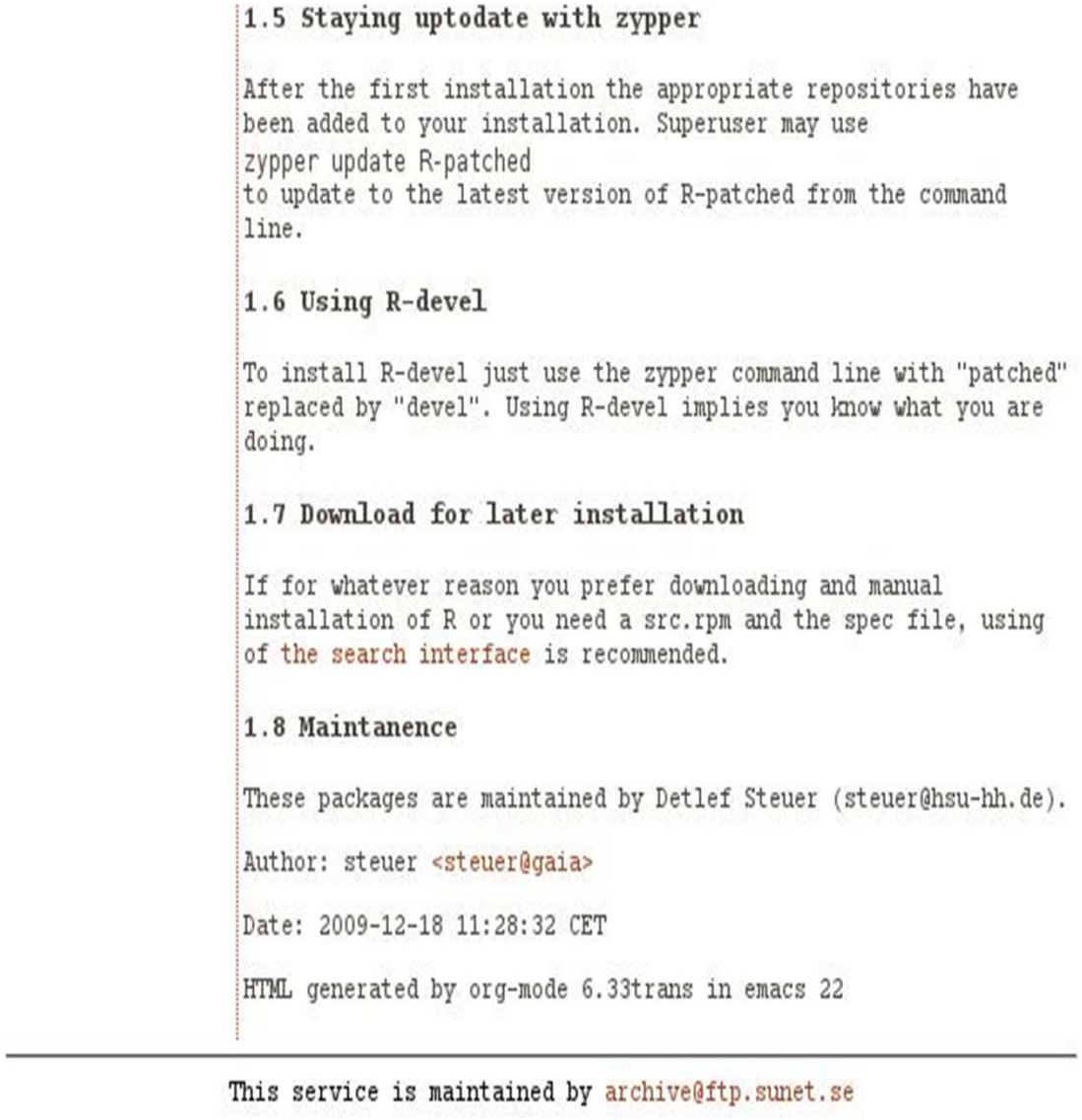
The links in the table below invoke yast2 directly to provide access to OpenSUSE's 1-click-install. Architecture and dependencies are handled automatically. You must be root to do this, resp. you will be asked to give the superuser password during installation. For maintained releases you'll receive current R-patched, otherwise R-base.

The devel package is needed, if you want to install add-on packages. You'll probably need to install a compiler, too. Some packages will need additional dependencies. See install instructions for the package, if an installation fails.

OpenSuSE Version	R-patched-2.13.0 1-click-install	R-patched-devel-2.13.0 1-click-install
10.1	Install	Install
10.2	Install	Install
10.3	Install	Install
SLE 10	Install	Install
11.0	Install	Install
11.1	Install	Install
11.2	Install	Install
11.3	Install	Install
11.4	Install	Install
SLE 11	Install	Install
Factory	Install	Install

The version provided at the moment is R-2.13.0 (patched). It includes the recommended packages and uses R's internal blas library.

More Information - SuSE



1.5 Staying uptodate with zypper

After the first installation the appropriate repositories have been added to your installation. Superuser may use

```
zypper update R-patched
```

to update to the latest version of R-patched from the command line.

1.6 Using R-devel

To install R-devel just use the zypper command line with "patched" replaced by "devel". Using R-devel implies you know what you are doing.

1.7 Download for later installation

If for whatever reason you prefer downloading and manual installation of R or you need a src.rpm and the spec file, using of the search interface is recommended.

1.8 Maintanence

These packages are maintained by Detlef Steuer (steuer@hsu-hh.de).

Author: steuer <steuer@gaia>

Date: 2009-12-18 11:28:32 CET

HTML generated by org-mode 6.33trans in emacs 22

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

Manuals

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 R Manuals

edited by the R Development Core Team.

Current Version: 2.11.1 (May 2010)

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, 15MB, approx. 3000 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()`).

Packages

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

Contributed Packages

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this directory. The manual R Installation and Administration (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, 28 views are available.

Daily 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 at 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 (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

Available Packages

Currently, the CRAN package repository features 2521 available packages.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Obtain a Package

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

gplots: Various R programming tools for plotting data

Various R programming tools for plotting data

Version:	2.8.0
Depends:	gtools, gdata, stats, caTools, grid
Suggests:	gtools
Published:	2010-06-15
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_2.8.0.tar.gz
MacOS X binary:	gplots_2.8.0.tgz
Windows binary:	gplots_2.8.0.zip
Reference manual:	gplots.pdf
News/ChangeLog:	NEWS ChangeLog
Old sources:	gplots archive

Reverse dependencies:

Reverse depends:	MADAM, MAMA, MetabolAnalyze, ROCR, RPPanalyzer, ResearchMethods, TIMP, ares, cellVolumeDist, genefu, ghyp, gregmisc, nnDiag, sisus, made4
Reverse imports:	MADAM, scapeMCMC, GeneticsBase, HTqPCR, flagme, tigre
Reverse suggests:	MKmisc, gmodels, rattle, simba, Agi4x44PreProcess, AgiMicroRna, plateCore

Install Package

Under Linux type the command:

R CMD INSTALL package.tar.gz

(no need to ungzip or untar the package)

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://addictedtor.free.fr/graphiques/

[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