R Tutorial -Part1

Rupali Akerkar
Department of Mathematical Sciences
NTNU, Norway

February 5, 2010

1 Introduction

This page contains introductory information on and about using R for course TMA4275-lifetime analysis. The aim is to help new bee's in R.

1.1 What is R

R is free statistical software. It is an open source implementation of the S language.The Comprehensive R Archive Network web page at http://cran.r-project.org/ is the authoritative source of information about R.
R is powerful software for interacting with data. With R you can create sophisticated graphs, you can carryout statistical analyses, and you can create and run simulations. R is also a programming language with an extensive set of built-in functions, so you can, with some experience, extend the language and write your own code to build your own statistical tools. Advanced users can even incorporate functions written in other languages, such as C, C++, and Fortran.
The S language has been around for more than twenty years and has been the most widely-used statistical software in departments of statistics for most of that time, first as S and then as the commercially available S-PLUS. R is an open source implementation of the S language that is now a viable alternative to S-PLUS, and in fact, has many advantages. A core team of statisticians and many other contributors work to update and improve R and to make versions that run well under all of the most popular operating systems. Most importantly to you, R is free, high-quality statistical software that will be useful as you learn statistics even though it is also a first-rate tool for professional statisticians.

1.2 Installation

Already available on pc in Labs. Installing R on your computer is simple if you have clear directions you can find that tell you exactly what to do in a way that is easy to understand. Directions exist at the R website (http://cran.rproject.org/) for installing R. OR Get information during lecture!!!!!.

2 Working in R

2.1 Basics

Some of the basic functions are given here. When a command is typed and entered, R responds, prints the result if relevant or asks for more input. Thus its a command-line interface.
calculating with numbers

```
> 5+7
[1] 12
> 3*5 +10-sqrt(4)
[1] 23
> 9*5/3+2
[1] 17
> 40/2*5 - 1
[1] 99
> sum(1:5)
[1] 15
> mean(1:5)
[1] 3
> sd(1:5)
[1] 1.581139
> a=1:10
> a
 [1]  1  2  3  4  5  6  7  8  9 10
```

Workspace
By default, R will use as its workspace the folder in which the executable program exists. You will most likely want to change the workspace to a new folder where you might keep data from the textbook and your homework.My advice is to have a folder where you keep work for this course and to change the worspace to this folder each time you start R.
Quitting R
To quit, you can type q() on a command line or you can quit through the File menu. R will prompt you if you want to save your workspace.
Entering data

- Using c

  ```
  >marks=c(34,67,12,45,76,32,44,65,41,68)
  > marks
   [1] 34 67 12 45 76 32 44 65 41 68
  ```

- Using read.table, you can copy a given data file with .txt extension, in same folder as working directory,then in R

```
>data=read.table("filename.txt", header = T)
```

It is necessary to add the argument **header=T** to the command as the first row usually contains the variable names for each variable.
The R commands above create data, an object known in R as a data frame. The name x is arbitrary. You may use any valid variable name (which does not begin with a digit or use characters with other meaning). A data frame is a data matrix, but can include both categorical and numerical variables.

2.2 Exploratory Data Analysis

For demonstration, we will use data from example 11.4, page 472 (H& R).

```
# to read data
> data=read.table("lifetime-data.txt",header=T)
# to check dimensions
> dim(data)
[1] 16  1
# to see first 5 lines
> data[1:5,]
[1] 31.7 39.2 57.5 65.0 65.8
```

simple data summary

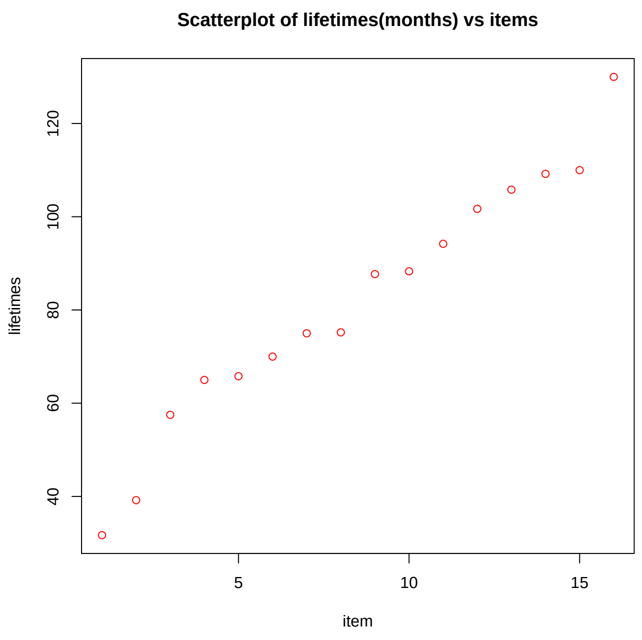
```
> summary(data)
# This will give mean, median, minimum, maximum, quartiles.
 life.time
 Min.   : 31.70
 1st Qu.: 65.60
 Median : 81.45
 Mean   : 81.64
 3rd Qu.:102.72
 Max.   :130.00

# for standard deviation
> sd(data)
life.time
 26.77588
```

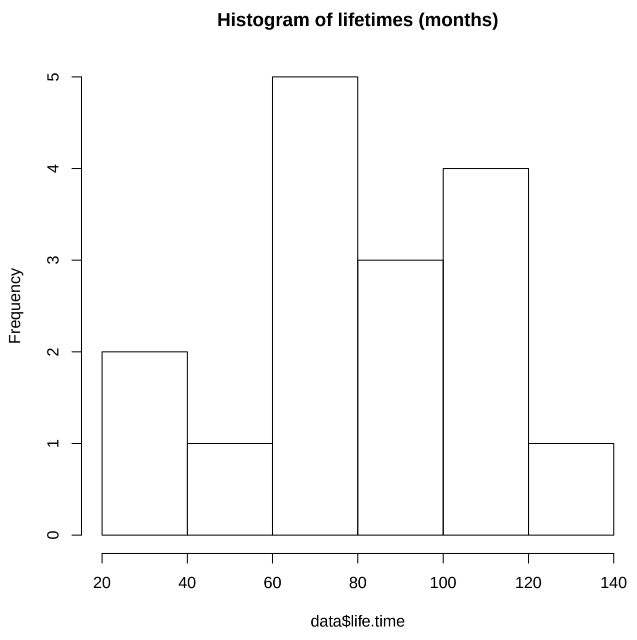
Scatter plot

```
# for x-axis we need index,so let
>x=1:16
> plot(x,data$life.time,xlab="item",ylab="lifetimes",
main="Scatterplot of lifetimes(months) vs items", col="red")
```

Histogram



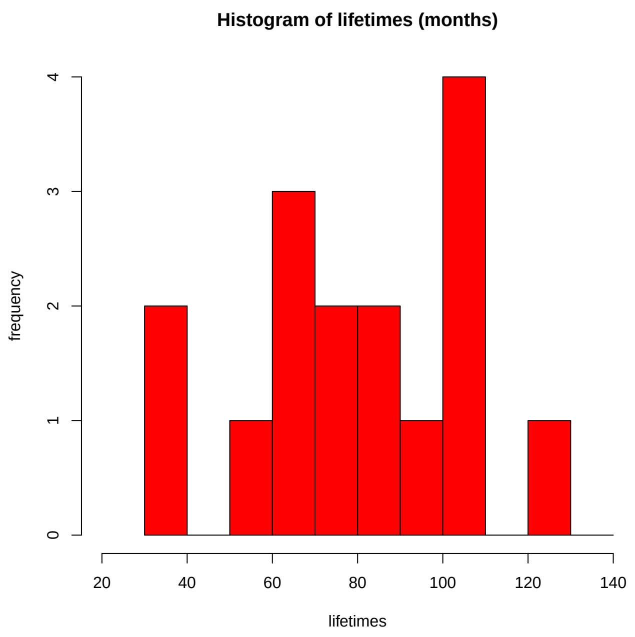
```
# to plot histogram
> hist(data$life.time)
# Since data is a data frame, we cannot use it directly
for drawing histogram, rather we need to give proper path.
```



To modify the histogram, can give

```
>hist(data$life.time, breaks= seq(30,140,by=10),xlab="lifetimes", ylab="frequency"
```

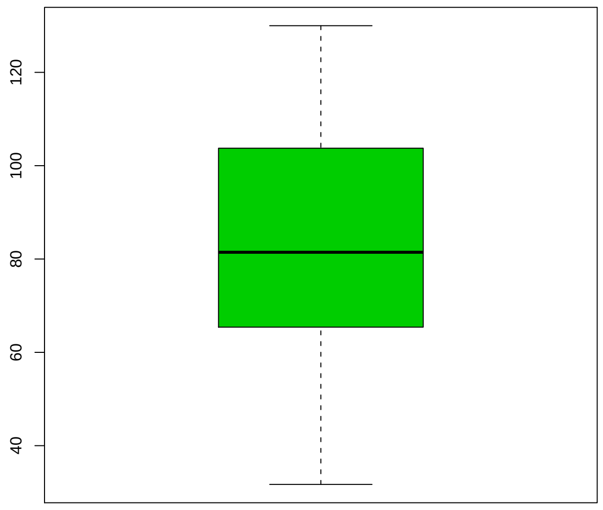
This will give, a cloured histogram with modified classes and labels. Note here breaks (classes) are from 30-40, 40-50, and so on, as defined in the command.



Boxplot

Boxplots are constructed using the boxplot function. If the argument is one vector, a single boxplot will be drawn. Parallel boxplots may be drawn by providing a list of variables, either directly using the list function or as the output of the split function which partitions one variable according to the categories of a second (categorical) variable. Here is a boxplot of the lifetimes for the data with the box shaded green.

```
> boxplot(data$life.time,col="3")
```



Probability distributions:

confidence interval: recall that the construction of confidence intervals requires the calculation of the sample mean, t-score and standard error. to compute 95% confidence interval for the mean of some data, simply apply the appropriate formula in R,

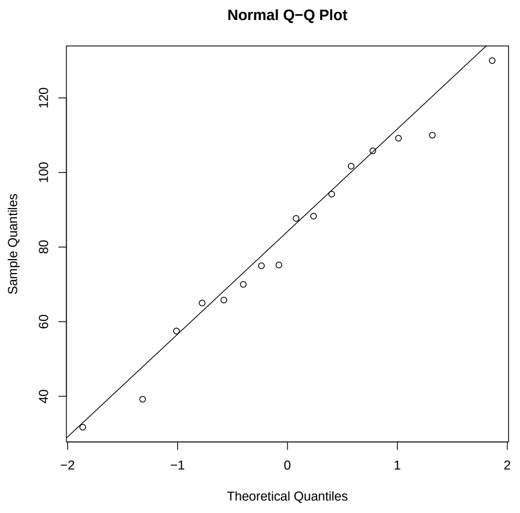
```
>n=length(data$life.time)
>data.ci.l=mean(data$life.time)-qt(0.975,n-1)*sd(data$life.time)/sqrt(n)
>data.ci.u=mean(data$life.time)+qt(0.975,n-1)*sd(data$life.time)/sqrt(n)
>c(data.ci.l,data.ci.u)
```

CDF

```
>x=data$life.time
>hist(x, probability=TRUE)
>mean=mean(data$life.time)
>sd=sd(data$life.time)
>xx <- seq(min(x), max(x), length=100)
>lines(xx, dnorm(xx, mean=mean, sd=sd)) #for f(t)
plot(xx, pnorm(xx, mean=mean, sd=sd), type="line") # for cdf
```

QQ-plots(normal probability plots)

```
# A normal QQ-plot of lifetimes are made as follows:
>qqnorm(data$life.time)
> qqline(data$life.time)
# if the data points are normally distributed, then the
 points should scatter around straight line. The second
 command plotsthe straight line corresponding to the normal
 distribution with mean equal to th esmaple mean and standard
 deviation equal to sample standard deviation.
```



NOTE: For general **QQ-plots** check slides on course web page.

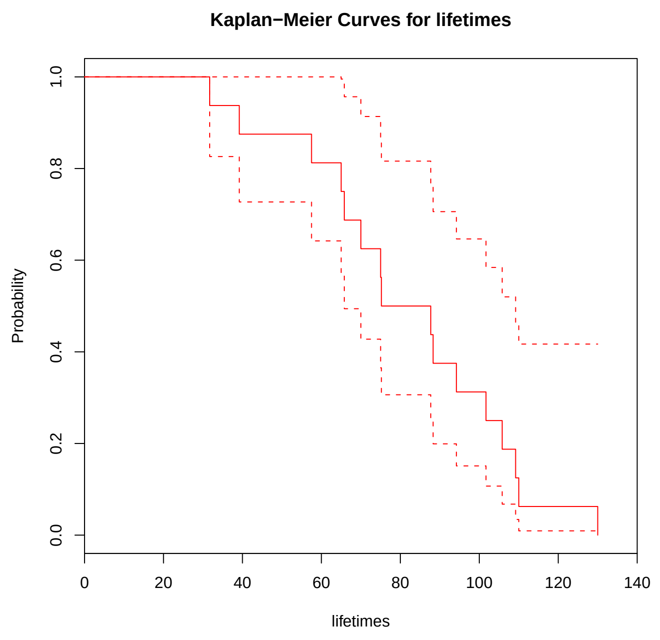
Kaplan Meier estimators

```
# data must be a data frame and must include status(failure),
information about censored or failed, hree the data is uncensored,
>status=rep(1,16)
>data1=data.frame(data$life.time,status)
# can check the dim of data1
>dim(data1)
[1] 16  2
# to estimate Kaplan-Meier function
>fit=survfit(Surv(data$life.time, status)~1, data=data1)
# to get information about it
>summary(fit)
Call: survfit(formula = Surv(data$life.time, status) ~ 1, data = data1)

  time n.risk n.event survival std.err lower 95% CI upper 95% CI
  31.7     16       1   0.9375  0.0605      0.82609        1.000
  39.2     15       1   0.8750  0.0827      0.72707        1.000
  57.5     14       1   0.8125  0.0976      0.64209        1.000
  65.0     13       1   0.7500  0.1083      0.56520        0.995
  65.8     12       1   0.6875  0.1159      0.49409        0.957
  70.0     11       1   0.6250  0.1210      0.42761        0.914
  75.0     10       1   0.5625  0.1240      0.36513        0.867
  75.2      9       1   0.5000  0.1250      0.30632        0.816
  87.7      8       1   0.4375  0.1240      0.25101        0.763
  88.3      7       1   0.3750  0.1210      0.19921        0.706
  94.2      6       1   0.3125  0.1159      0.15108        0.646
 101.7      5       1   0.2500  0.1083      0.10699        0.584
 105.8      4       1   0.1875  0.0976      0.06761        0.520
 109.2      3       1   0.1250  0.0827      0.03419        0.457
 110.0      2       1   0.0625  0.0605      0.00937        0.417
 130.0      1       1   0.0000     NaN           NA           NA
# to draw the curves,
>  plot(fit,lty=1:3,xlab="lifetimes",ylab="Probability",
main="Kaplan-Meier Curves for lifetimes",xlim=c(0,140),col="2")
```

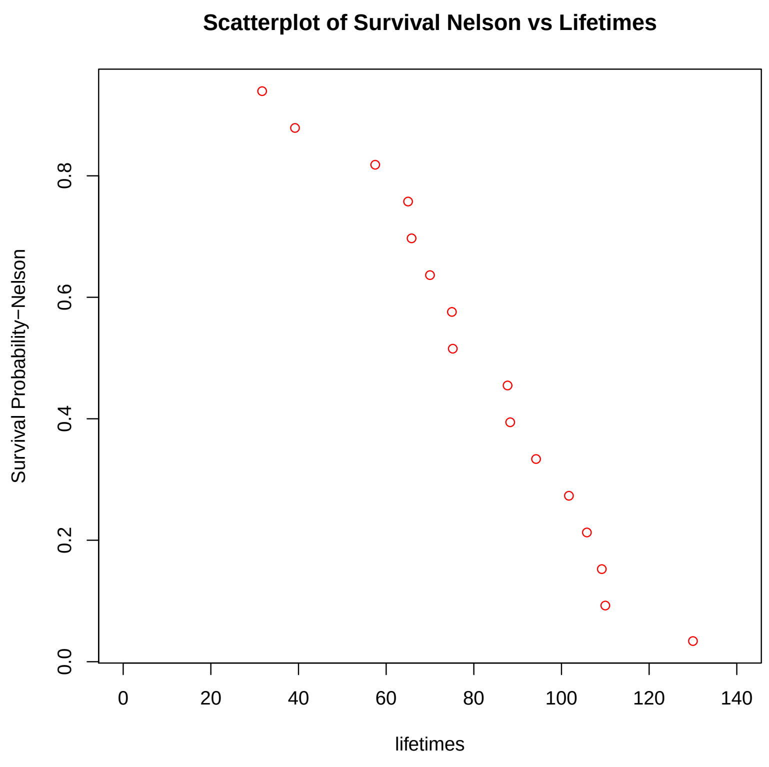
Nelson-Aalen estimators:

```
# The easiest way to get Nelson-Aalen estimator is via cox
 regression using coxph function.
> fit1=survfit(coxph(Surv(data$life.time,status)~1), type="aalen")
#to get the information
>summary(fit1)
```



```
Call: survfit.coxph.null(formula = coxph(Surv(data$life.time, status) ~
    1), type = "aalen")

  time n.risk n.event survival std.err lower 95% CI upper 95% CI
  31.7     16       1   0.9394  0.0587      0.83111        1.000
  39.2     15       1   0.8788  0.0803      0.73472        1.000
  57.5     14       1   0.8182  0.0949      0.65186        1.000
  65.0     13       1   0.7577  0.1054      0.57678        0.995
  65.8     12       1   0.6971  0.1131      0.50723        0.958
  70.0     11       1   0.6365  0.1184      0.44210        0.916
  75.0     10       1   0.5759  0.1216      0.38076        0.871
  75.2      9       1   0.5154  0.1230      0.32287        0.823
  87.7      8       1   0.4548  0.1225      0.26826        0.771
  88.3      7       1   0.3943  0.1202      0.21690        0.717
  94.2      6       1   0.3337  0.1160      0.16890        0.659
 101.7      5       1   0.2732  0.1095      0.12453        0.600
 105.8      4       1   0.2128  0.1005      0.08429        0.537
 109.2      3       1   0.1525  0.0882      0.04909        0.474
 110.0      2       1   0.0925  0.0707      0.02067        0.414
 130.0      1       1   0.0340  0.0428      0.00289        0.401
# to plot Nelson-Aalen estimates,
>  plot(data$life.time,fit1$surv,col="2",xlab="lifetimes",
ylab="Survival Probability-Nelson",main="Scatterplot of
Survival Nelson vs Lifetimes",xlim=c(0,140))
```



3 References

Excellant detailed introduction is given on these sites:

- http://cran.r-project.org/doc/manuals/R-intro.pdf