OPEN SOURCE INTRODUCTION TO STATISTICS WITH R

Krishna K. Saha
Central Connecticut State University
Department of Mathematical Sciences
New Britain, Connecticut, 06053 USA
sahakrk@ccsu.edu

Abstract

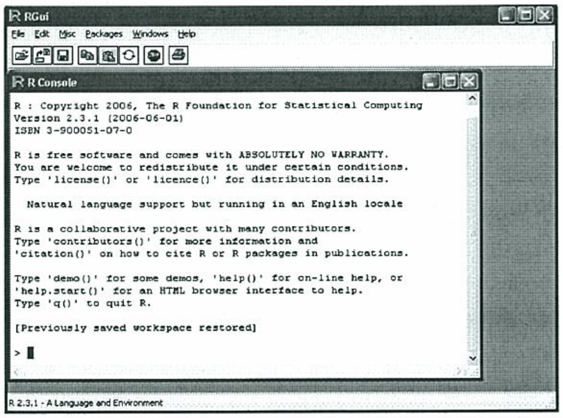
This paper illustrates the use of open source statistical tool and interactive programming language R for introduction to statistics courses. Specifically, we work through examples to demonstrate the use of R functions for data manipulation, graphical presentation, probability distribution description, and traditional topics in statistical inferences, such as confidence intervals and significance tests.

1. **Introduction**

Technology plays a vital role when students who have only pre-calculus skills are first learning statistics. In introductory statistics courses we usually use graphing calculators such as the TI-83/84 introduced by the Texas Lutheran University as well as the statistical software such as SPSS, Minitab, and Excel to avoid cumbersome mathematical and numerical computations. However, students often have difficulty acquiring and using these instruments. In such cases, open source R gains wider acceptance due to simplicity and economic reasons as well as its available documentations on the web site. In addition, the students would be better served as it is relatively easy to use and free, and could be installed campus-wide as well as at home. The goal of this paper is to explore my ideas by demonstrating the use of R functions for an introductory statistics course by working through examples. We will limit our demonstrations for the use of R functions to Windows applications.

2. Start R under Windows

To start R under Microsoft Windows, click on the R icon on the PC, it will open a Window, along with the command prompt subwindow. The following Figure shows the screen snapshot of the command window as it appears. In this R window, > is the *command line prompt*, where we need to type the R command of the problem and press the Enter key to execute the command by R. This will provide the result of the command starting from the next line in the same window. For example, type the command *sqrt(49)* for $\sqrt{49}$ in the *command line prompt* and press the Enter.


RGui
File Edit Misc Packages Windows Help
R Console
R 2.3.1 - A Language and Environment

```
R : Copyright 2006, The R Foundation for Statistical Computing
Version 2.3.1 (2006-06-01)
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> 
```

Then the result of this command will be 7, which will appear on the subsequent line, that is,

```
>sqrt(49)
```

```
[1] 7.
```

Here the [1] refers to the first observation. To exit or quit R, type the command q( ) in the *command line prompt* or click on the File menu in the top of the R window and then click on Exit.

3. **Data Entry in R Workspace**

In many ways data can be entered in R workspace. The R functions c( ), scan( ), read.table( ), read.csv( ), and attach( ) can be used to store different forms of data sets, such as the vector data, columns of data, a TEXT data file or Excel spreadsheet. We will illustrate below how to use these functions to load a data set in an R session.

Example 1: A data set for the number of absences of 7 randomly selected students in a statistics class is: 6, 2, 15, 9, 12, 5, and 8. To store this data set in R workspace, the R function c( ) or scan( ) can be used as follows:

```
> c(6, 2, 15, 9, 12, 5, 8)
[1] 1 0 6 4 2 5
> scan()
1: 1 0 6 4 2 5
7:
Read 6 items
[1] 1 0 6 4 2 5
```

Example 2: Data from the text file, namely trees.txt, or data from the spreadsheet file in CSV format, namely trees.csv in drive "d," can be loaded into R session, respectively, as

```
>read.table("d:/trees.txt", header=T)
Girth Height Volume
1   8.3   70   10.3
2   8.6   65   10.3
…… ….   ….
31  20.6   87   77.0
> read.csv("d:/trees.csv", header=T)
Girth Height Volume
1   8.3   70   10.3
2   8.6   65   10.3
…… ….   ….
31  20.6   87   77.0
```

4. **Graphical Presentation**

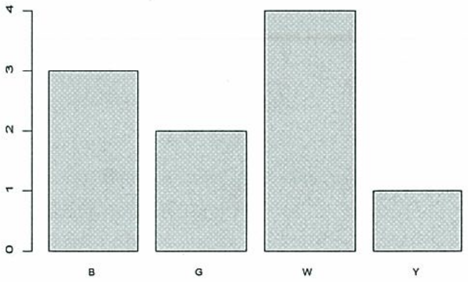
4.1 **Graphs for Categorical Data.**

Categorical data set is summarized in a categorical frequency table and graphical manner. The R function table( ) can be used to create the categorical frequency table. For example, we can construct a categorical frequency table for the data of 10 randomly selected men's dress colors, W, W, B, Y, G, B, G, W, B, and W, as follows:

```
> data<-c("W", "W", "B", "Y", "G", "B",
"G", "W", "B", "W")
> table(data)
data
B G W Y
3 2 4 1
```

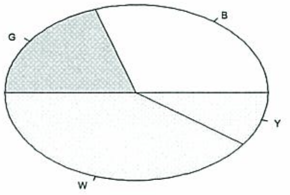
To find a bar graph for this data set, use the R function barplot( ) as

```
> barplot(table(data))
```



Also, to find a pie graph for this data set, use the R function pie( ) as

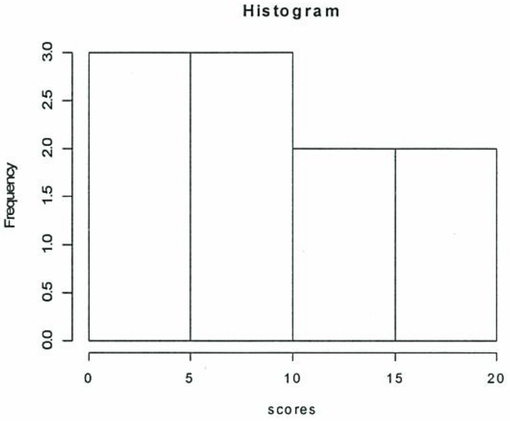
```
> pie(table(data))
```



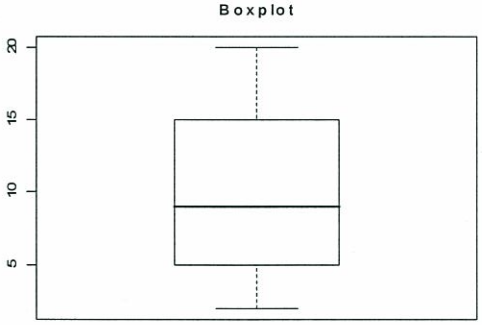
4.2 **Graphs for Categorical Data**

Graphical presentation can also be used to describe the numerical data set. The graphs most commonly used to analyze such data sets are the histogram, box plot, and stem-and-leaf plot. To illustrate these graphs using the R functions, we consider a set of data for the scores of 10 students on a 20-point quiz in a statistics course shown here:

```
> scores<-c(18,15,12,6,8,2,3,5,20,10)
> hist(scores)
```



```
> boxplot(scores)
```



```
> stem(scores)
The decimal point is 1 digit(s) to the right
of the |
 0 | 23
 0 | 568
 1 | 02
 1 | 58
 2 | 0
```

5. Descriptive Statistics using R

In this section, we discuss how to obtain descriptive statistics, such as measures of average, measures of variation, and measures of position for the same score data set used above, as follows:

```
> mean(scores) #mean of the scores
[1] 9.9
> median(scores) #median of the scores
[1] 9
> length(scores) #n=length of the scores
[1] 10
> range(scores) #range of the scores
[1]  2 20
> var(scores) #variance of the scores
[1] 38.98889
> sd(scores) #standard deviation
[1] 6.244108
> quantile(scores,0.25) #1st quantile
 25%
5.25
> quantile(scores,0.95) #95th percentile
 95%
19.1
> IQR(scores) #IQR of the scores
[1] 9
```

Here # is the comment character. All text in the line above started by # is treated as a comment.

6. R as a Calculator

R can be used to evaluate the result of any expression. For example, we can compute the mean, variance and standard deviation of the same scores data using the formulas:

$\bar{x} = \sum x/n \, ; s^2 = \sum (x_i - \bar{x})^2 /(n-1);$

$s = \sqrt{s^2}$ as follows:

```
> x<-c(18,15,12,6,8,2,3,5,20,10)
> mean<-sum(x)/length(x)
> mean
[1] 9.9
> variance<-sum((x-mean)^2)/(length(x)-1)
> variance
[1] 38.98889
> st.deviation<-sqrt(variance)
> st.deviation
[1] 6.244108
```

Also, we can use R to perform basic arithmetic, as we would do with a calculator. For example

```
> 2+3*4+sqrt(5)+exp(6)-7/8
[1] 418.7899
> log(10)
[1] 2.302585
> sin(pi)
[1] 1.224606e-16
```

7. Probability Distributions

There are a number of basic distributions that are usually used in many different probability models, such as binomial, Poisson, uniform, normal, t, χ^2, and F distributions. Four kinds of R functions are available to analyze the data using any of these models, which are combined "d", "p", "q", and "r" with the name of the distributions in R. For example, as the name of the binomial distribution (BD) in R is "binom" so these R functions would be dbinom( ), pbinom( ), qbinom( ), and rbinom( ). Here dbinom( ) returns the probability distribution function of the BD, whereas pbinom( ) returns the cumulative distribution function of the BD. Similarly, qbinom() and rbinom( ) return the quantile and the random sample from a distribution, respectively. Below we show the use of these functions with example.

Suppose we toss a fair coin 100 times and let X be the number of heads. Then X follows a binomial distribution with n = 10 and p = 0.5. The probability that there are exactly 7 heads, P(X = 7) can be obtained as

```
> dbinom(7, size = 10, prob = 0.5)
[1] 0.1172
```

The probability that there are 7 or fewer heads, $P(X \leq 7)$ is:

```
> pbinom(7, size = 10, prob = 0.5)
[1] 0.9453
```

The second quantile of X is given by

```
 > qbinom(0.5, size = 10, prob = 0.5)
[1] 5
```

Finally, a random sample of size 10 can be drawn from the BD (10, 0.5) by

```
> rbinom(10, size = 10, prob = 0.5)
 [1] 4 5 6 6 6 7 3 7 4 4
```

Similarly, one can use these types of functions for other distributions. The names of Poisson, uniform, normal, t, χ^2, and F distributions in R are, respectively, "pois", "unif", "norm", t, chisq, and f.

8. Confidence Intervals using R

In this section, we discuss how to construct confidence intervals for unknown parameters, such as population mean (μ) and population proportion (P) in R. Motivating examples are used to find confidence intervals for a population mean or proportion.

8.1 Confidence Interval for μ

If a sample of size n is drawn from a normal population with mean μ and unknown variance σ^2, then a $(1-\alpha)100\%$ confidence interval for μ is given by

$$\bar{x} \pm t_{\alpha/2} \, {}^{s}\!/_{\sqrt{n}} \, .$$

R provides a function called t.test( ) to obtain this confidence interval. Here is an example for a random sample of the daily salaries of substitute teachers for eight local school districts:

```
> x<-c(60,56,60,55,70,55,60,55) #data set
```

Now, the 90% confidence interval of the mean for the salaries of substitute teachers in the region can be obtained by

```
> t.test(x,conf.level=0.90)
       One Sample t-test
data:  x
t = 32.7594, df = 7, p-value = 6.392e-09
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 55.47007 62.27993
sample estimates:
mean of x    58.875
```

So, based on this sample, one can be 90% confident that the mean for the salaries of substitute teachers in the region is between \$55.47 and \$62.28.

8.2 Confidence Interval for P

The function prop.test( ) can be used to obtain a confidence interval for a population proportion. For example, in a study of 150 households, only 54 had central air conditioning. The 95% confidence interval of the true proportion of households with central air conditioning can be found as:

```
> prop.test(54,150,conf.level=0.95)
      1-sample proportions test with
continuity correction data: 54 out of 150,
null probability 0.5
X-squared = 11.2067, df = 1, p-value =
0.000815
alternative hypothesis: true p is not equal
to 0.5
95 percent confidence interval:
 0.2844673 0.4428236
sample estimates:
   p
0.36
```

That is, $0.2845 < P < 0.4428$.

9. Significant Tests

Tests of significance are used to investigate the hypotheses concerning parameters of interest, such as population means and populations. There are two specific statistical tests used for the hypotheses concerning means: the z test and the t test. The following example will explain in detail the testing procedure with t test in R.

Example: The average daily salary for substitute teachers was reported to be \$63.75. To verify this report, a random sample of the daily salaries of substitute teachers for eight local school districts was taken and given in section 8.1. Assume that X = the daily salary has $N(\mu, \sigma^2)$. The t.test( ) function in R can be used to test $H_0 : \mu = 63.75$ vs $H_a : \mu = \$63.75$ as follows:

```
> x<-c(60,56,60,55,70,55,60,55) #data set
>t.test(x,alternative="two.sided",mu=63.75)
      One Sample t-test
data: x
t = -2.7126, df = 7, p-value = 0.03009
alternative hypothesis: true mean is not
equal to 63.75
95 percent confidence interval:
 54.62531 63.12469
sample estimates:
mean of x
  58.875
```

Here t = -2.7126 with p-value = 0.0301 indicates that at the 5% level of significance the average daily salary for substitute teachers was not \$63.75. Note that this function can also be used to test the hypothesis about the two population means $H_0 : \mu_1 = \mu_2$ vs $H_a : \mu_1 \neq \mu_2$. In addition, the function prop.test( ) can be used to test the hypothesis about the population proportions.

Conclusion: Open source R is an excellent statistical tool for the beginning statistics student. It is very simple to use and free to install from the online R home page: http://www.r-project.org. This tool provides students with the opportunity to learn statistics by utilizing their existing computer skills.

References

Maindonald, J. and Braun, J. (2003). Data Analysis and Graphics Using R, Cambridge University Press.

Verzani, J. (2005). Using R for Introductory Statistics, Chapman & Hall/CRC.