Exploring data using R

Kamarul Imran Musa, Wan Nor Arifin

2017-07-05

Contents

1 Introduction to R **5**

1.1 R and RStudio . . . 5

1.2 Working with Data . . . 6

2 Textual **9**

2.1 Basic descriptive statistics . . . 9

2.2 More on descriptive statistics . . . 13

2.3 More on tables . . . 19

3 Graphical **21**

4 Reporting results **23**

5 Final Words **25**

Chapter 1

Introduction to R

This chapter introduces readers to the basics of working with data in R. We will start with installing R in your computer and getting familiar with RStudio interface. These will be followed by the basics of handling data in R.

1.1 R and RStudio

1.1.1 Installing R and RStudio

Install R base package: http://www.r-project.org/

Install RStudio: http://www.rstudio.com/

1.1.2 Getting familiar with the interface

Consists of 4 tabs:

1. Source
2. Console
3. Environment & History
4. Misc. Most important Plots, Packages & Help

1.1.3 R script

source tab

- important
- everything done here
- keep track what's going on
- not recommended to type in console

1.1.4 Working with packages

what is package/library

1.1.4.1 Installing packages

```
install.packages("package.name")
```

1.1.4.2 Loading libraries

```
library("package.name")
```

1.2 Working with Data

1.2.1 Setting working directory

general steps

- codes
- point-and-click

1.2.2 Data management

concerns reading data from data set, displaying data.

advanced, direct input in the code, esp. useful for tables.

1.2.2.1 Reading data set

Easiest is to read .csv file.

```
read.csv("file.name")
```

For SPSS file, need `foreign` package

```
library("foreign")
read.spss("file.name")
```

Can read data in table format from text file. From text file

```
read.table("file.name", header = TRUE)
```

1.2.2.2 Viewing data set

Easy, just type the name,

```
data
```

Nicer, using `View()`

```
View(data)
```

Important tasks

```
dim(data)
str(data)
names(data)
```

1.2.3 More about data management

- subsetting
- new variable
- recoding
- direct input for table

Chapter 2

Textual

In this chapter, we will go through a number of R functions for basic statistics. The focus will be on the results that are presented in form of numbers in text or tables (textual). We will mostly use the builtin functions (from R standard library). Extra packages will be introduced whenever necessary.

2.1 Basic descriptive statistics

In this part, we are going to use the functions as applied to a variable. For this purpose, we are going to use builtin datasets in R. You can view the available datasets by

```
data()
```

```
## Data sets in package ‘datasets’:

## AirPassengers                     Monthly Airline Passenger Numbers 1949-1960
## BJsales                           Sales Data with Leading Indicator
## BJsales.lead (BJsales)            Sales Data with Leading Indicator
## BOD                               Biochemical Oxygen Demand
## CO2                               Carbon Dioxide Uptake in Grass Plants
## ...
```

We can view any dataset description by appending “?” to the dataset name. For example,

```
?chickwts
```

We will start by using `chickwts` dataset that contains both numerical (`weight`) and categorical (`feed`) variables. We can view the first six observations,

```
head(chickwts)
```

```
##   weight      feed
## 1    179 horsebean
## 2    160 horsebean
## 3    136 horsebean
## 4    227 horsebean
## 5    217 horsebean
## 6    168 horsebean
```

the last six observations,

```
tail(chickwts)
```

```
##    weight   feed
## 66    352 casein
## 67    359 casein
## 68    216 casein
## 69    222 casein
## 70    283 casein
## 71    332 casein
```

and the dimension of the data (row and column).

```
dim(chickwts)
```

```
## [1] 71  2
```

Here we have 71 rows (71 subjects) and two columns (two variables).

Next, view the names of the variables,

```
names(chickwts)
```

```
## [1] "weight" "feed"
```

and view the details of the data,

```
str(chickwts)
```

```
## 'data.frame':    71 obs. of  2 variables:
##  $ weight: num  179 160 136 227 217 168 108 124 143 140 ...
##  $ feed  : Factor w/ 6 levels "casein","horsebean",..: 2 2 2 2 2 2 2 2 2 2 ...
```

which shows that `weight` is a numerical variable and `feed` is a factor, i.e. a categorical variable. `feed` consists of six categories or levels.

We can view the levels in `feed`,

```
levels(chickwts$feed)
```

```
## [1] "casein"    "horsebean" "linseed"   "meatmeal"  "soybean"   "sunflower"
```

2.1.1 Describing a numerical variable

A numberical variable is described by a number of descriptive statistics below.

To judge the central tendency of the `weight` variable, we obtain its mean,

```
mean(chickwts$weight)
```

```
## [1] 261.3099
```

and median,

```
median(chickwts$weight)
```

```
## [1] 258
```

To judge its spread and variability, we can view its minimum, maximum and range

```
min(chickwts$weight)
```

```
## [1] 108
```

```r
max(chickwts$weight)
```

```
## [1] 423
```

```r
range(chickwts$weight)
```

```
## [1] 108 423
```

and obtain its standard deviation (SD)

```r
sd(chickwts$weight)
```

```
## [1] 78.0737
```

variance,

```r
var(chickwts$weight)
```

```
## [1] 6095.503
```

quantile,

```r
quantile(chickwts$weight)
```

```
##    0%   25%   50%   75%  100% 
## 108.0 204.5 258.0 323.5 423.0
```

and interquartile range (IQR)

```r
IQR(chickwts$weight)
```

```
## [1] 119
```

There are nine types of quantile algorithms in R (for `quantile` and `IQR`), the default being type 7. You may change this to type 6 (Minitab and SPSS),

```r
quantile(chickwts$weight, type = 6)
```

```
##   0%  25%  50%  75% 100% 
##  108  203  258  325  423
```

```r
IQR(chickwts$weight, type = 6)
```

```
## [1] 122
```

In addition to SD and IQR, we can obtain its median absolute deviation (MAD),

```r
mad(chickwts$weight)
```

```
## [1] 91.9212
```

It is actually simpler to obtain most these in a single command,

```r
summary(chickwts$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   108.0   204.5   258.0   261.3   323.5   423.0
```

even simpler, obtain all of the statistics using `describe` in the `psych` package

```r
install.packages("psych")
```

```r
library(psych)
describe(chickwts$weight)
```

```
##    vars  n   mean    sd median trimmed   mad min max range  skew kurtosis
## X1    1 71 261.31 78.07    258     261 91.92 108 423   315 -0.01    -0.97
##      se
## X1 9.27
```

2.1.2 Describing a categorical variable

A categorical variable is described by its count, proportion and percentage by categories.

We obtain the count of the `feed` variable,

```
summary(chickwts$feed)
```

```
##    casein horsebean   linseed  meatmeal   soybean sunflower
##        12        10        12        11        14        12
```

```
table(chickwts$feed)
```

```
##
##    casein horsebean   linseed  meatmeal   soybean sunflower
##        12        10        12        11        14        12
```

both `summary` and `table` give the same result.

`prop.table` gives the proportion of the result from the count.

```
prop.table(table(chickwts$feed))
```

```
##
##    casein horsebean   linseed  meatmeal   soybean sunflower
## 0.1690141 0.1408451 0.1690141 0.1549296 0.1971831 0.1690141
```

the result can be easily turned into percentage,

```
prop.table(table(chickwts$feed))*100
```

```
##
##    casein horsebean   linseed  meatmeal   soybean sunflower
##  16.90141  14.08451  16.90141  15.49296  19.71831  16.90141
```

To view the count and the percentage together, we can use `cbind`,

```
cbind(n = table(chickwts$feed), "%" = prop.table(table(chickwts$feed))*100)
```

```
##            n        %
## casein    12 16.90141
## horsebean 10 14.08451
## linseed   12 16.90141
## meatmeal  11 15.49296
## soybean   14 19.71831
## sunflower 12 16.90141
```

We need the quotation marks " " around the percentage sign %, because % also serves as a mathematical operator in R.

2.2 More on descriptive statistics

Just now, we viewed all the statistics as applied to a variable. In this part, we are going to view the statistics on a number of variables. This includes viewing a group of numerical variables or categorical variables, or a mixture of numerical and categorical variables. This is relevant in a sense that, most of the time, we want to view everything in one go (e.g. the statistics of all items in a questionnaire), compare the means of several groups and obtain cross-tabulation of categorical variables.

2.2.1 Describing numerical variables

Let us use `women` dataset,

```
head(women)
```

```
##   height weight
## 1     58    115
## 2     59    117
## 3     60    120
## 4     61    123
## 5     62    126
## 6     63    129
```

```
names(women)
```

```
## [1] "height" "weight"
```

```
str(women)
```

```
## 'data.frame':    15 obs. of  2 variables:
##  $ height: num  58 59 60 61 62 63 64 65 66 67 ...
##  $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
```

which consists of `weight` and `height` numerical variables.

The variables can be easily viewed together by `summary`,

```
summary(women)
```

```
##      height         weight     
##  Min.   :58.0   Min.   :115.0  
##  1st Qu.:61.5   1st Qu.:124.5  
##  Median :65.0   Median :135.0  
##  Mean   :65.0   Mean   :136.7  
##  3rd Qu.:68.5   3rd Qu.:148.0  
##  Max.   :72.0   Max.   :164.0
```

even better using `describe` (`psych`),

```
describe(women)
```

```
##        vars  n   mean    sd median trimmed   mad min max range skew
## height    1 15  65.00  4.47     65   65.00  5.93  58  72    14 0.00
## weight    2 15 136.73 15.50    135  136.31 17.79 115 164    49 0.23
##        kurtosis   se
## height    -1.44 1.15
## weight    -1.34 4.00
```

2.2.2 Describing categorical variables

Let us use `infert` dataset,

```
head(infert)
```

```
##   education age parity induced case spontaneous stratum pooled.stratum
## 1    0-5yrs  26      6       1    1           2       1              3
## 2    0-5yrs  42      1       1    1           0       2              1
## 3    0-5yrs  39      6       2    1           0       3              4
## 4    0-5yrs  34      4       2    1           0       4              2
## 5   6-11yrs  35      3       1    1           1       5             32
## 6   6-11yrs  36      4       2    1           1       6             36
```

```
names(infert)
```

```
## [1] "education"      "age"            "parity"         "induced"       
## [5] "case"           "spontaneous"    "stratum"        "pooled.stratum"
```

```
str(infert)
```

```
## 'data.frame':    248 obs. of  8 variables:
##  $ education     : Factor w/ 3 levels "0-5yrs","6-11yrs",..: 1 1 1 1 2 2 2 2 2 2 ...
##  $ age           : num  26 42 39 34 35 36 23 32 21 28 ...
##  $ parity        : num  6 1 6 4 3 4 1 2 1 2 ...
##  $ induced       : num  1 1 2 2 1 2 0 0 0 0 ...
##  $ case          : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ spontaneous   : num  2 0 0 0 1 1 0 0 1 0 ...
##  $ stratum       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ pooled.stratum: num  3 1 4 2 32 36 6 22 5 19 ...
```

We notice that `induced`, `case` and `spontaneous` are not yet set as categorical variables, thus we need to `factor` the variables. We view the value labels in the dataset description,

```
?infert
```

We label the values in the variables according to the description as

```
infert$induced = factor(infert$induced, levels = 0:2, labels = c("0", "1", "2 or more"))
infert$case = factor(infert$case, levels = 0:1, labels = c("control", "case"))
infert$spontaneous = factor(infert$spontaneous, levels = 0:2, labels = c("0", "1", "2 or more"))
str(infert)
```

```
## 'data.frame':    248 obs. of  8 variables:
##  $ education     : Factor w/ 3 levels "0-5yrs","6-11yrs",..: 1 1 1 1 2 2 2 2 2 2 ...
##  $ age           : num  26 42 39 34 35 36 23 32 21 28 ...
##  $ parity        : num  6 1 6 4 3 4 1 2 1 2 ...
##  $ induced       : Factor w/ 3 levels "0","1","2 or more": 2 2 3 3 2 3 1 1 1 1 ...
##  $ case          : Factor w/ 2 levels "control","case": 2 2 2 2 2 2 2 2 2 2 ...
##  $ spontaneous   : Factor w/ 3 levels "0","1","2 or more": 3 1 1 1 2 2 1 1 2 1 ...
##  $ stratum       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ pooled.stratum: num  3 1 4 2 32 36 6 22 5 19 ...
```

and we now all these variables are turned into factors.

Again, the variables can be easily viewed together by `summary`,

```
summary(infert[c("education", "induced", "case", "spontaneous")])
```

```
##    education        induced         case        spontaneous
##  0-5yrs : 12   0        :143   control:165   0        :141
##  6-11yrs:120   1        : 68   case   : 83   1        : 71
##  12+ yrs:116   2 or more: 37                 2 or more: 36
```

We do not use `table` here in form of `table(infert[c("education", "induced", "case", "spontaneous")])` because `table` used in this form will give us 3-way cross-tabulation instead of count per categories. Cross-tabulation of categorical variables will be covered later.

To obtain the proportion and percentage results, we have to use `lapply`,

```
lapply(infert[c("education", "induced", "case", "spontaneous")], function(x) summary(x)/length(x))
```

```
## $education
##    0-5yrs   6-11yrs   12+ yrs
## 0.0483871 0.4838710 0.4677419
##
## $induced
##         0         1 2 or more
## 0.5766129 0.2741935 0.1491935
##
## $case
##   control      case
## 0.6653226 0.3346774
##
## $spontaneous
##         0         1 2 or more
## 0.5685484 0.2862903 0.1451613
```

```
lapply(infert[c("education", "induced", "case", "spontaneous")], function(x) summary(x)/length(x)*100)
```

```
## $education
##   0-5yrs  6-11yrs  12+ yrs
##  4.83871 48.38710 46.77419
##
## $induced
##         0         1 2 or more
##  57.66129  27.41935  14.91935
##
## $case
##  control     case
## 66.53226 33.46774
##
## $spontaneous
##         0         1 2 or more
##  56.85484  28.62903  14.51613
```

because we need `lappy` to obtain the values for each of the variables. `lappy` goes through each variable and performs this particular part,

```
function(x) summary(x)/length(x)
```

`function(x)` is needed to specify some extra operations to any basic function in R, in our case `summary(x)` divided by `length(x)`, in which the summary results (the counts) are divided by the number of subjects (`length(x)` gives us the "length" of our dataset).

Now, since we already learned about `lapply`, we may also obtain the same results by using `summary` (within `lapply`), `table` and `prop.table`.

```
lapply(infert[c("education", "induced", "case", "spontaneous")], summary)
```

```
## $education
##  0-5yrs 6-11yrs 12+ yrs
##      12     120     116
##
## $induced
##         0         1 2 or more
##       143        68        37
##
## $case
## control    case
##     165      83
##
## $spontaneous
##         0         1 2 or more
##       141        71        36
```

```
lapply(infert[c("education", "induced", "case", "spontaneous")], table)
```

```
## $education
##
##  0-5yrs 6-11yrs 12+ yrs
##      12     120     116
##
## $induced
##
##         0         1 2 or more
##       143        68        37
##
## $case
##
## control    case
##     165      83
##
## $spontaneous
##
##         0         1 2 or more
##       141        71        36
```

```
lapply(infert[c("education", "induced", "case", "spontaneous")], function(x) prop.table(table(x)))
```

```
## $education
## x
##    0-5yrs   6-11yrs   12+ yrs
## 0.0483871 0.4838710 0.4677419
##
## $induced
## x
##         0         1 2 or more
## 0.5766129 0.2741935 0.1491935
##
## $case
## x
##   control      case
## 0.6653226 0.3346774
##
## $spontaneous
## x
##         0         1 2 or more
## 0.5685484 0.2862903 0.1451613
```

```
lapply(infert[c("education", "induced", "case", "spontaneous")], function(x) prop.table(table(x))*100)
```

```
## $education
## x
##   0-5yrs  6-11yrs  12+ yrs
##  4.83871 48.38710 46.77419
##
## $induced
## x
##         0         1 2 or more
##  57.66129  27.41935  14.91935
##
## $case
## x
##  control     case
## 66.53226 33.46774
##
## $spontaneous
## x
##         0         1 2 or more
##  56.85484  28.62903  14.51613
```

Notice here, whenever we do not need to specify extra operations on a basic function, e.g. `summary` and `table`, all we need to write after the comma in `lapply` is the basic function without `function(x)` and `(x)`.

2.2.3 Describing the variables together

In the preceeding sections, we intentionally went through the descriptive statistics of a variable, followed by a number of variables of the same type. This will give you the basics in dealing with the variables. Most commonly, the variables are described by groups or in form cross-tabulated counts/percentages.

2.2.3.1 By groups

To obtain all the descriptive statistics by group, we can use `by` with the relevant functions

```
by(infert[c("age", "parity")], infert$case, summary)
```

```
## infert$case: control
##       age            parity     
##  Min.   :21.00   Min.   :1.000  
##  1st Qu.:28.00   1st Qu.:1.000  
##  Median :31.00   Median :2.000  
##  Mean   :31.49   Mean   :2.085  
##  3rd Qu.:35.00   3rd Qu.:3.000  
##  Max.   :44.00   Max.   :6.000  
## --------------------------------------------------------
## infert$case: case
##       age            parity     
##  Min.   :21.00   Min.   :1.000  
##  1st Qu.:28.00   1st Qu.:1.000  
##  Median :31.00   Median :2.000  
##  Mean   :31.53   Mean   :2.108  
##  3rd Qu.:35.50   3rd Qu.:3.000  
##  Max.   :44.00   Max.   :6.000  
```

```r
by(infert[c("age", "parity")], infert$case, describe)
```

```
## infert$case: control
##        vars   n  mean   sd median trimmed  mad min max range skew kurtosis
## age       1 165 31.49 5.25     31    31.34 5.93  21  44    23 0.23    -0.72
## parity    2 165  2.08 1.24      2     1.88 1.48   1   6     5 1.32     1.42
##          se
## age    0.41
## parity 0.10
## -------------------------------------------------------- 
## infert$case: case
##        vars  n  mean   sd median trimmed  mad min max range skew kurtosis
## age       1 83 31.53 5.28     31    31.39 5.93  21  44    23 0.21    -0.77
## parity    2 83  2.11 1.28      2     1.90 1.48   1   6     5 1.32     1.34
##          se
## age    0.58
## parity 0.14
```

```r
by(infert[c("education", "induced", "spontaneous")], infert$case, summary)
```

```
## infert$case: control
##    education        induced      spontaneous 
##  0-5yrs : 8   0        :96   0        :113  
##  6-11yrs:80   1        :45   1        : 40  
##  12+ yrs:77   2 or more:24   2 or more: 12  
## -------------------------------------------------------- 
## infert$case: case
##    education        induced      spontaneous
##  0-5yrs : 4   0        :47   0        :28  
##  6-11yrs:40   1        :23   1        :31  
##  12+ yrs:39   2 or more:13   2 or more:24  
```

```r
by(infert[c("education", "induced", "spontaneous")], infert$case, function(x) lapply(x, table))
```

```
## infert$case: control
## $education
## 
##  0-5yrs 6-11yrs 12+ yrs 
##       8      80      77 
## 
## $induced
## 
##         0         1 2 or more 
##        96        45        24 
## 
## $spontaneous
## 
##         0         1 2 or more 
##       113        40        12
## 
## --------------------------------------------------------
## infert$case: case
## $education
## 
##  0-5yrs 6-11yrs 12+ yrs 
##       4      40      39 
## 
## $induced
## 
##         0         1 2 or more 
##        47        23        13 
## 
## $spontaneous
## 
##         0         1 2 or more 
##        28        31        24
```

Please note that simply replacing `table` for `summary` as in `by(infert[c("education", "induced", "spontaneous")], infert$case, table)` will not work as intended. `education` will be nested in `induced`, which is nested in `spontaneous`, listed by `case` instead.

We can also use `describeBy`, which is an the extension of `describe` in the `psych` package.

```
describeBy(infert[c("age", "parity")], group = infert$case)
```

```
## 
##  Descriptive statistics by group 
## group: control
##        vars   n  mean   sd median trimmed  mad min max range skew kurtosis
## age       1 165 31.49 5.25     31   31.34 5.93  21  44    23 0.23    -0.72
## parity    2 165  2.08 1.24      2    1.88 1.48   1   6     5 1.32     1.42
##          se
## age    0.41
## parity 0.10
## --------------------------------------------------------
## group: case
##        vars  n  mean   sd median trimmed  mad min max range skew kurtosis
## age       1 83 31.53 5.28     31   31.39 5.93  21  44    23 0.21    -0.77
## parity    2 83  2.11 1.28      2    1.90 1.48   1   6     5 1.32     1.34
##          se
## age    0.58
## parity 0.14
```

which is limited to numerical variables only.

2.2.3.2 Simple cross-tabulation

2.3 More on tables

This requires special attention.

Chapter 3

Graphical

Test GIT Test GIT 2 - commit

Chapter 4

Reporting results

Chapter 5

Final Words

We have finished a nice book.