

Lecture 3: Some tips on R

Pratheepa Jeganathan

09/27/2019

Recall

- ▶ What is a regression model?
- ▶ Descriptive statistics – graphical
- ▶ Descriptive statistics – numerical
- ▶ Inference about a population mean
- ▶ Difference between two population means

Today (Some tips on R)

Introduction to R

We will use R and R Markdown for this course (highly recommended). The examples in the lecture notes and homework assignments will be written in R. Choosing R for your homework solutions and project is highly recommended.

- ▶ Follow this <https://www.r-project.org/> to install R:
 - ▶ R is an interpreted language, which means you will not have to compile your code and your actual code will be executed.
 - ▶ R is interactive for data analysis.
 - ▶ R includes interfaces to other programming languages (Python, Julia, C++), which means you can adapt R to big data analysis or computationally intensive procedures (Chambers 2017).
 - ▶ Read more about R: <https://www.r-project.org/about.html>.

Introduction to R Markdown

- ▶ Follow this <https://www.rstudio.com/> to install R Studio (The newest version of R Studio is highly recommended (v1.1.463)): we will use R Markdown from R Studio to
 - ▶ track data analysis.
 - ▶ produce high-quality documents that can be shared with your collaborators.
 - ▶ reproduce the results.
 - ▶ Read more about R Markdown: [here](#).

Introduction to Latex

- ▶ Latex, which will enable you to create PDFs directly from the R Markdown in RStudio.

```
install.packages("tinytex")
```

- ▶ After installing TinyTex, close RStudio.
- ▶ Reopen RStudio.
- ▶ Run the following:

```
tinytex::install_tinytex()
```

Basics of R and R Markdown

These examples follow Kloeke and McKean (2015): Nonparametric Statistical Methods Using R. Chapter 1 (Kloeke and McKean 2014).

Matrices and data frames

Make vectors:

```
x = c(11,218,123,36,1001)
y = rep(1,5)
z = seq(1,5,by =1)
```

Vector operations:

```
y + z
```

```
## [1] 2 3 4 5 6
```

```
u = y + z # comments: assign the value to variable u
u
```

```
## [1] 2 3 4 5 6
```

- Some more operations

```
sum(x)
```

```
## [1] 1389
```

```
c(mean(x),sd(x),var(x),median(x))
```

```
## [1]      277.8000      412.3733 170051.7000      123.0000
```

```
length(x)
```

```
## [1] 5
```

- **A word of caution.** In R you can overwrite built-in functions so try not to call variables `c`.
- Other variables to be careful are the aliases `T` for `TRUE` and `F` for `FALSE`. Since we compute t and F statistics it is natural to also have variables named `T` so when you are expecting `T` to be `TRUE` you might get a surprise.

Generate a random sample

Ex: coin tossing

```
coin = c("H", "T")  
set.seed(100)  
samples = sample(x= coin, size =100,  
  replace = TRUE)
```

the number times H shows up

```
sum(samples == "H")
```

```
## [1] 50
```

Matrices

combine vectors of same data type into matrices

```
X = cbind(x,y,z)
```

```
X
```

```
##           x y z
## [1,]    11 1 1
## [2,]   218 1 2
## [3,]   123 1 3
## [4,]    36 1 4
## [5,]  1001 1 5
```

create a matrix using R function from the base package

```
Y = matrix(data = c(2,3,4,5,6,7),  
  nrow = 2, ncol =3, byrow = TRUE)
```

Y

```
##      [,1] [,2] [,3]  
## [1,]    2    3    4  
## [2,]    5    6    7
```

Data frame

combine vectors of different data types

```
subjects = c('Jim', 'Jack', 'Joe', 'Mary', 'Jean')
score = c(85, 90, 75, 100, 70)
D = data.frame(subjects = subjects, score = score)
D
```

```
##   subjects score
## 1      Jim    85
## 2     Jack    90
## 3      Joe    75
## 4     Mary   100
## 5     Jean    70
```

```
D$class = c("Jun", "Sopho", "Sopho", "Sopho", "Jun")  
D
```

```
##    subjects score class  
## 1      Jim     85   Jun  
## 2     Jack     90 Sopho  
## 3      Joe     75 Sopho  
## 4     Mary    100 Sopho  
## 5     Jean     70   Jun
```

Generating random variables

R provides numerous functions for random number generation

Ex: generate standard normal random variable

```
z = rnorm(n = 100, mean = 0, sd = 1)
```

```
summary(z)
```

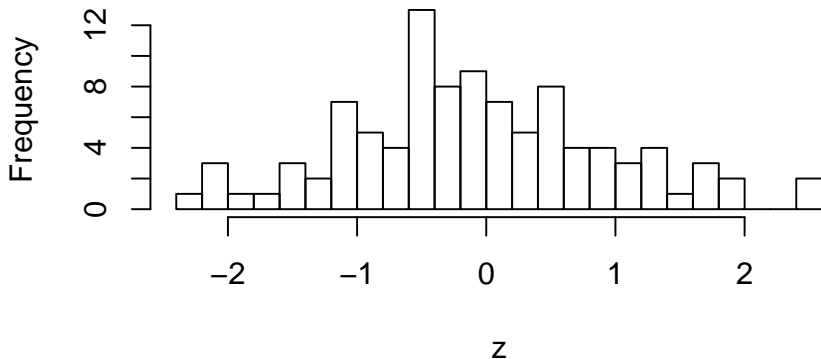
```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -2.27193 -0.72820 -0.12918 -0.08774  0.45056  2.58196
```


Graphics

Basic plotting Ex: histogram of Z

```
hist(z,breaks = 30)
```

Histogram of z



Sophisticated plots

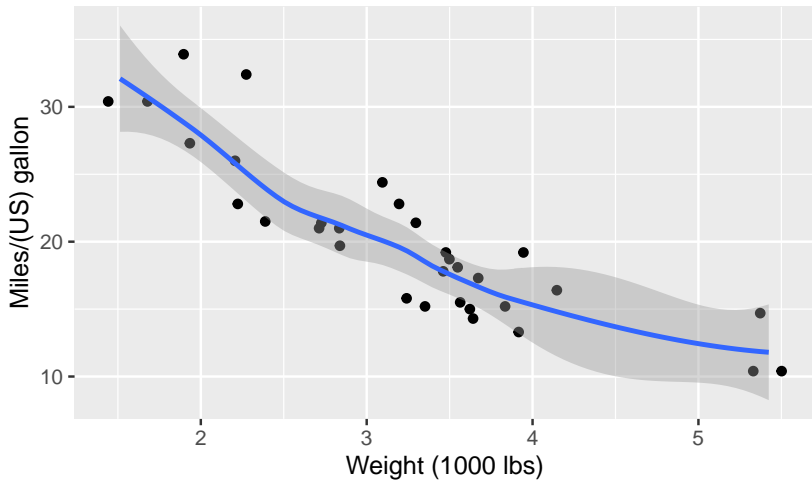
The ggplot2 package is very popular to make more sophisticated plots

```
library(ggplot2)
```

You are encouraged to learn the grammar of ggplot. There are many tutorials online. Here is one example [link](#).

Let's see how to use ggplot2 for scatter plots on automobile data

```
data(mtcars)
p = ggplot(mtcars, aes(x=wt,y=mpg)) +
  geom_point(position=position_jitter(w=0.1,h=0)) +
  geom_smooth() + xlab('Weight (1000 lbs)') +
  ylab("Miles/(US) gallon")
```



```
ggsave("example_plot.eps", p, width = 6, height = 5)
```

- Add an external image and write a caption

```
knitr::include_graphics("example_plot.eps")
```

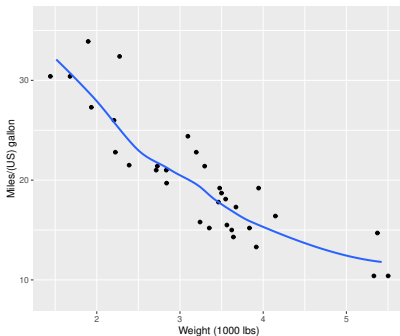


Figure 1: External image

Repeating tasks

- In addition to for loop, R provides `apply` and `tapply` functions to replicate code a number of times

```
X
```

```
##           x y z
## [1,]    11 1 1
## [2,]   218 1 2
## [3,]   123 1 3
## [4,]    36 1 4
## [5,]  1001 1 5
```

- row-wise mean

```
apply(X,1,mean)
```

```
## [1]    4.333333  73.666667  42.333333  13.666667 335.666667
```

- ▶ column-wise mean

```
apply(X,2,mean)
```

```
##      x      y      z
## 277.8   1.0   3.0
```

```
D
```

```
## subjects score class
## 1      Jim     85   Jun
## 2      Jack    90 Sopho
## 3      Joe     75 Sopho
## 4      Mary   100 Sopho
## 5      Jean    70   Jun
```

```
tapply(D$score,D$class,mean)
```

```
##      Jun      Sopho
## 77.50000 88.33333
```

User defined functions

```
mSummary = function(x) {  
  q1 = quantile(x,.25)  
  q3 = quantile(x,.75)  
  lt = list(med=median(x),iqr=q3-q1)  
  return(lt)  
}  
xsamp = 1:13  
mSummary(xsamp)
```

```
## $med  
## [1] 7  
##  
## $iqr  
## 75%  
##    6
```


- Read data set from the website

```
readCSVFromCANVAS = function(url, sep = "\t"){  
  read.table(url, header = T, sep = sep)  
}  
  
groundhog = readCSVFromCANVAS("http://web.stanford.edu/class/STAT316/groundhog.csv")  
head(groundhog)
```

	##	year	mintemp	shadow
	## 1	1990	24	N
	## 2	1991	23	Y
	## 3	1992	22	Y
	## 4	1993	16	Y
	## 5	1994	12	Y
	## 6	1995	13	N

Monte Carlo simulations

Generate a data set with 100 rows and 10 columns. Each row is from a standard normal distribution.

```
set.seed(1000)  
X = matrix(rnorm(10*100),ncol=10)
```

Sample mean of each of the 100 samples:

```
xbar = apply(X, MARGIN = 1, FUN = mean)
```

Variance of sample mean:

```
var(xbar)
```

```
## [1] 0.1013805
```

compared to theoretical results: $\frac{\sigma^2}{n}$

```
1/10
```

```
## [1] 0.1
```

R packages

Two distribution site: CRAN and Bioconductor

In addition to commonly used functions in R, some other functions are available from developers.

For example, to use functions in dplyr package, we need to install the package.

```
install.packages("dplyr")
```

```
library(dplyr)  
head(iris)
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa

Distributions in R

- ▶ In practice, we will often be using the distribution (CDF), quantile (inverse CDF) of standard random variables like the T , F , chi-squared and normal.
- ▶ The standard 1.96 (about 2) standard deviation rule for $\alpha = 0.05$: (note that $1 - 0.05/2 = 0.975$)

```
qnorm(0.975)
```

```
## [1] 1.959964
```

- ▶ We might want the $\alpha = 0.05$ upper quantile for an F with 2,40 degrees of freedom:

```
qf(0.95, 2, 40)
```

```
## [1] 3.231727
```

- ▶ So, any observed F greater than 3.23 will get rejected at the $\alpha = 0.05$ level.

- ▶ Alternatively, we might have observed an F of 5 with 2, 40 degrees of freedom, and want the p-value

```
1 - pf(5, 2, 40)
```

```
## [1] 0.01152922
```

- ▶ Let's compare this p-value with a chi-squared with 2 degrees of freedom, which is like an F with infinite degrees of freedom in the denominator (send 40 to infinity).
- ▶ We also should multiply the 5 by 2 because it's divided by 2 (numerator degrees of freedom) in the F .

```
c(1 - pchisq(5*2, 2), 1 - pf(5, 2, 4000))
```

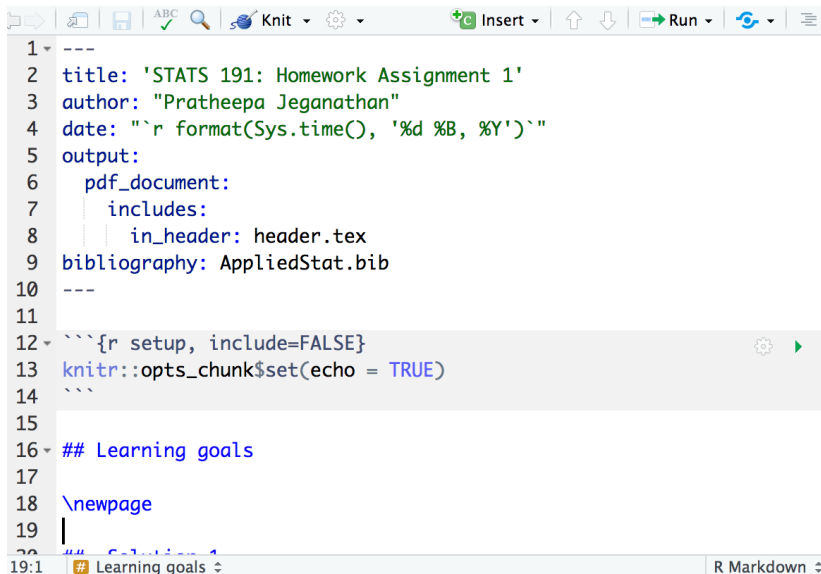
```
## [1] 0.006737947 0.006780121
```

Other common distributions used in applied statistics are `norm` and `t`.

Templates

Homework template

- ▶ See the homework template in [Canvas/Files/Templates](#).
- ▶ More templates:
 - ▶ You will have access to R markdown files for making lecture slides and homework assignments.
- ▶ See the following [link](#) for a further outline of using R markdown for reporting.



The image shows a screenshot of an R Markdown editor interface. The top toolbar includes icons for file operations, a search bar, a Knit button, a settings gear, an Insert button, and Run/Refresh buttons. The main text area contains R Markdown code for a homework assignment template. The code includes metadata (title, author, date), output settings (pdf_document), bibliography (AppliedStat.bib), and a chunk for learning goals. The status bar at the bottom indicates the current line is 19:1 and the document is in R Markdown format.

```
1 ---
2 title: 'STATS 191: Homework Assignment 1'
3 author: "Pratheepa Jeganathan"
4 date: "`r format(Sys.time(), '%d %B, %Y')`"
5 output:
6   pdf_document:
7     includes:
8       in_header: header.tex
9 bibliography: AppliedStat.bib
10 ---
11
12 ```{r setup, include=FALSE}
13 knitr::opts_chunk$set(echo = TRUE)
14 ```
15
16 ## Learning goals
17
18 \newpage
19 |
20 ## Solution 1
21
22 ## Learning goals
```

Figure 2: Homework Template

Other references

- ▶ [An Introduction to R](#)
- ▶ [R for Beginners](#)
- ▶ [Modern Applied Statistics with S](#)
- ▶ [Practical ANOVA and Regression in R](#)
- ▶ [simpleR](#)
- ▶ [R Reference Card](#)
- ▶ [R Manuals](#)
- ▶ [R Wiki](#)
- ▶ [Modern Statistics for Modern Biology](#)
- ▶ [R Studio Education](#)

References for this lecture

- ▶ Based on the lecture notes of [Jonathan Taylor](#) .

Chambers, John M. 2017. *Extending R*. Chapman; Hall/CRC.

Kloke, John, and Joseph W McKean. 2014. *Nonparametric Statistical Methods Using R*. Chapman; Hall/CRC.