Lecture 2 Supplementary: Some Tips on R

Donghyeon Yu

Materials in this lecture notes come from th lecture notes written by Pratheepa Jeganathan (Stanford Univ.).

Recall

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

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.
  - Read more about R: https://www.r-project.org/about.html.

Introduction to R Markdown

- Follow this https://posit.co/download/rstudio-desktop/ to install R Studio (The newest version of R Studio is highly recommended (v2025.05.1+513)): 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

Vectors

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

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
```

```r
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)
```

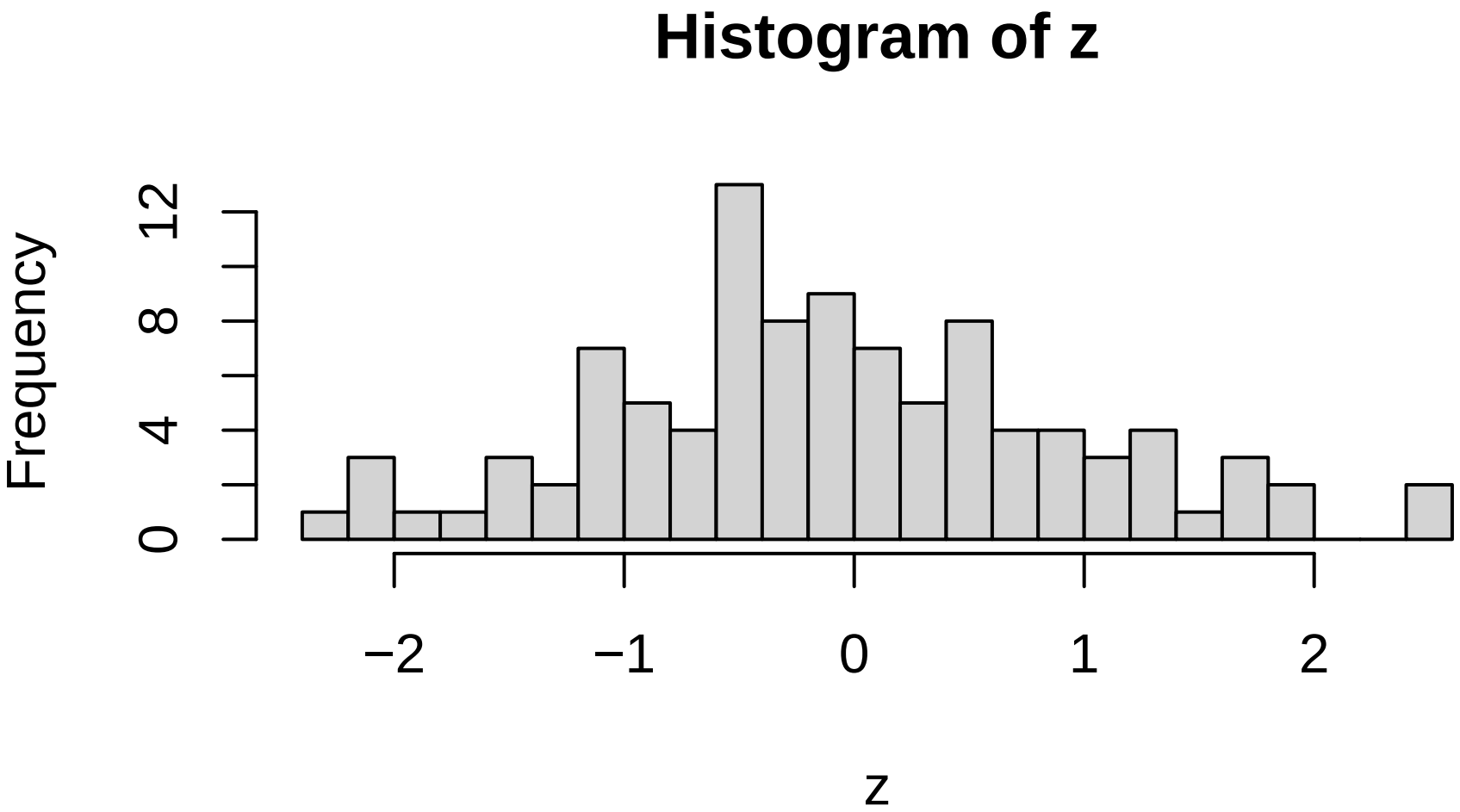
```
summary(z)
```

```
##      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)
```



Sophisticated plots

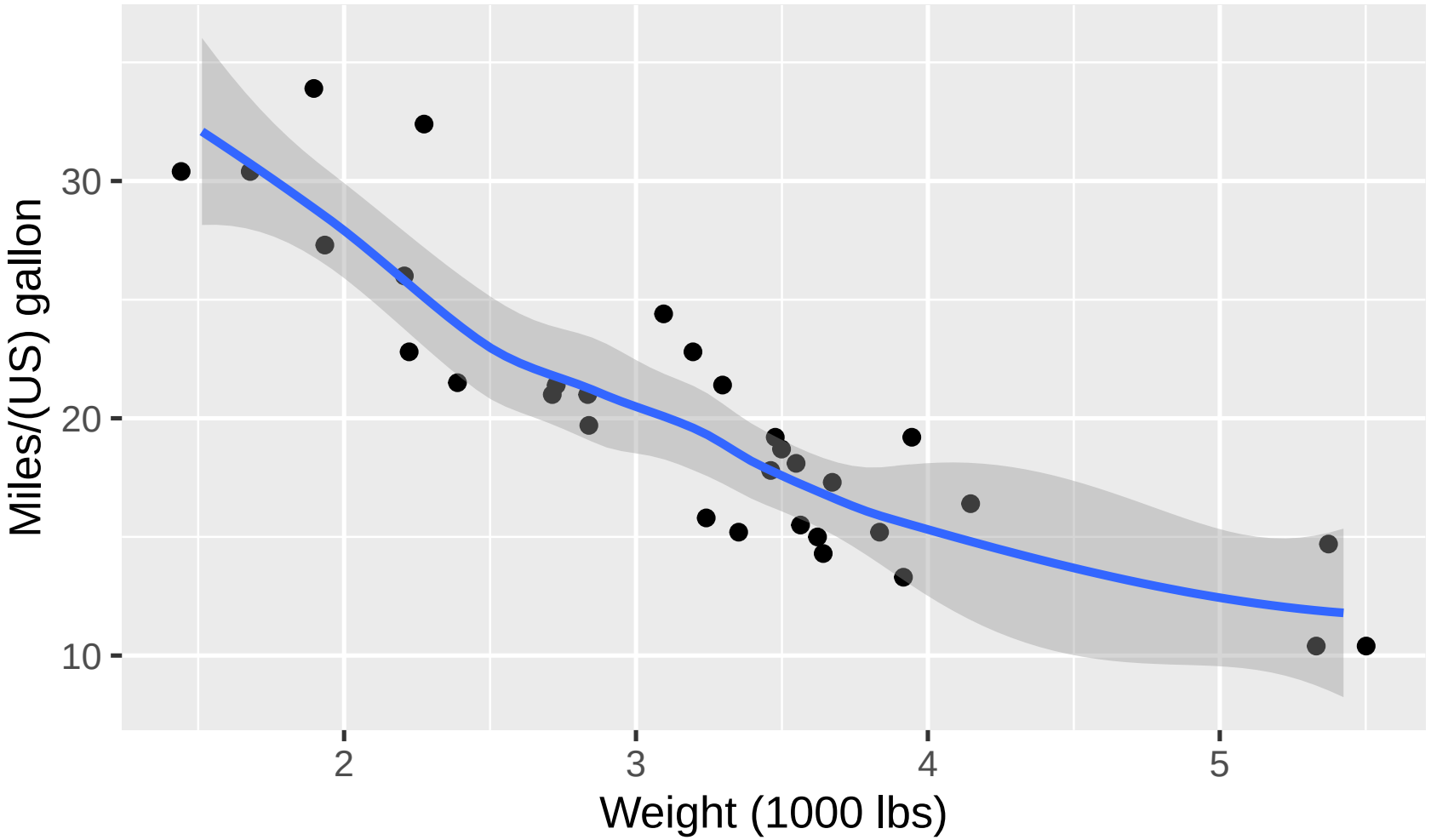
The ggplot2 package is very popular to make more sophisticated plots

```
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")
```


Miles/(US) gallon
30
20
10
2
3
4
5
Weight (1000 lbs)

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

- Add an external image and write a caption

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

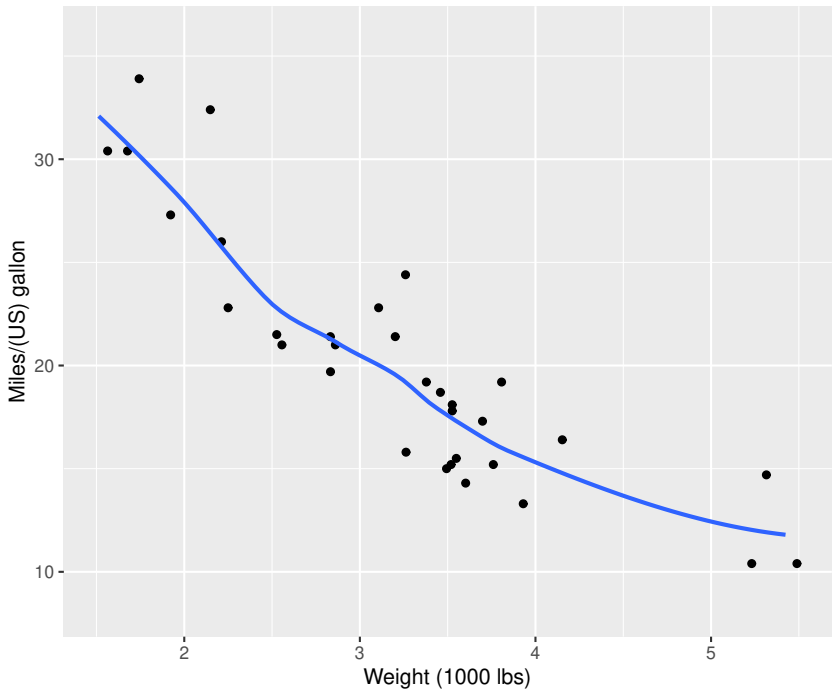


Figure 1: External image

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

```
readCSVFromWeb = function(url, sep = "\t"){
  read.table(url, header = T, sep = sep)
}

rtw  = readCSVFromWeb(url = "http://www1.aucegypt.edu/faculty/hadi/RABE
 sep = ",")
head(rtw)
```

```
##                City.COL.PD.URate.Pop.Taxes.Income.RTWL
## 1    Atlanta\t169\t414\t13.6\t1790128\t5128\t2961\t1
## 2        Austin\t143\t239\t11\t396891\t4303\t1711\t1
## 3 Bakersfield\t339\t43\t23.7\t349874\t4166\t2122\t0
## 4    Baltimore\t173\t951\t21\t2147850\t5001\t4654\t0
## 5    Baton Rouge\t99\t255\t16\t411725\t3965\t1620\t1
## 6    Boston\t363\t1257\t24.4\t3914071\t4928\t5634\t0
```

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, 1)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
```

```
dplyr::filter(iris, Species== "setosa")[1,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  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`.

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 Pratheepa Jeganathan
- Based on the lecture notes of Jonathan Taylor .