Lecture 0: Overview and Setup

Shengwu Shang

9/2/2021

Overview

- Rank-based testing (classical nonparametric techniques), R-estimators, nonparametric confidence intervals, modern nonparametric (bootstrap), curve fitting (density, regression function), confidence sets, wavelets, Bayesian nonparametric, credible intervals.
- Current interests
  - Special issue in Statistical Sciences (A Review Journal of The Institute of Mathematical Statistics): gives papers for review project (17 articles) [@randles2004].

Software

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 [@chambers2017].
  - Read more about R: here.

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

(optional, if you will render R Markdown to HTML documents and if you'll use some other word processor to write a report for your project)

- Latex, which will enable you to create PDFs directly from the R Markdown in RStudio.
  - Mac users should download macTeX http://www.tug.org/mactex/downloading.html from Safari (not Chrome).
  - Windows users should install MiKTeX https://miktex.org/download.

Basics of R and R Markdown

Vectors

These examples follow **(KM)**: 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
```

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
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)
```

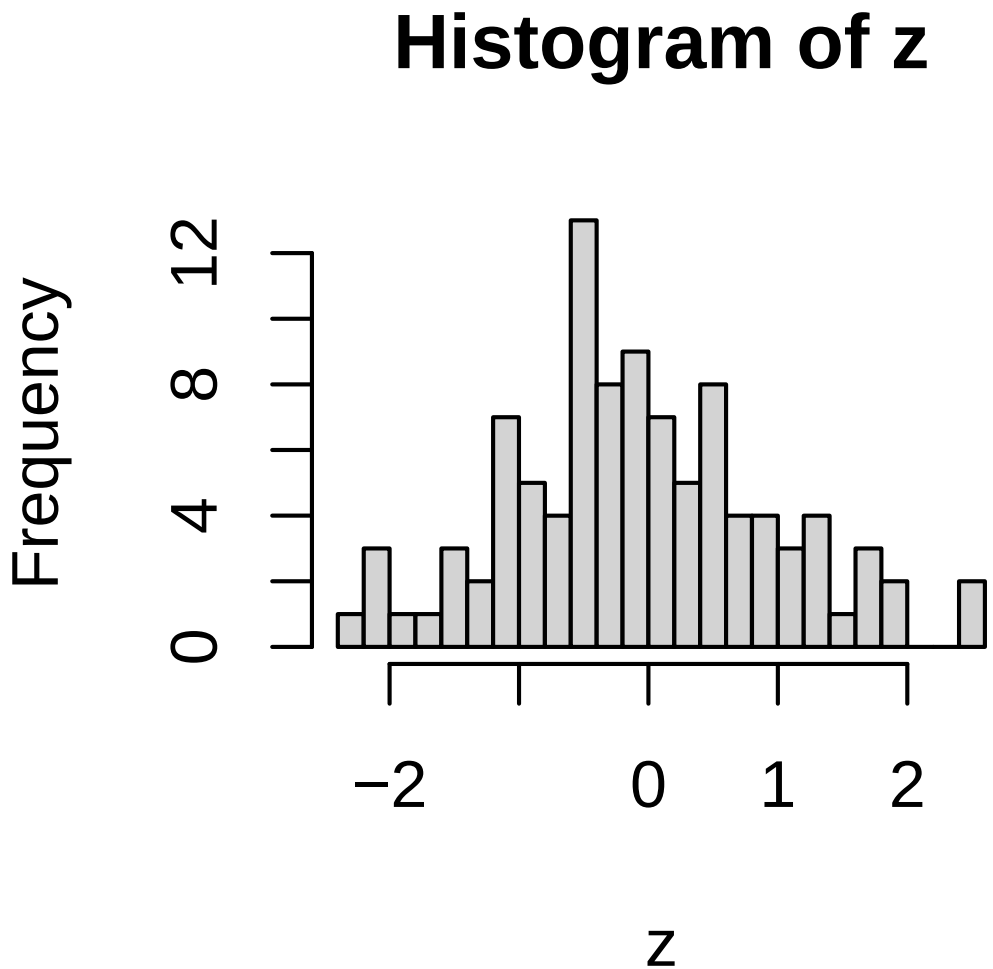
```
summary(z)
```

```
##      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)
```



Sophisticated plots

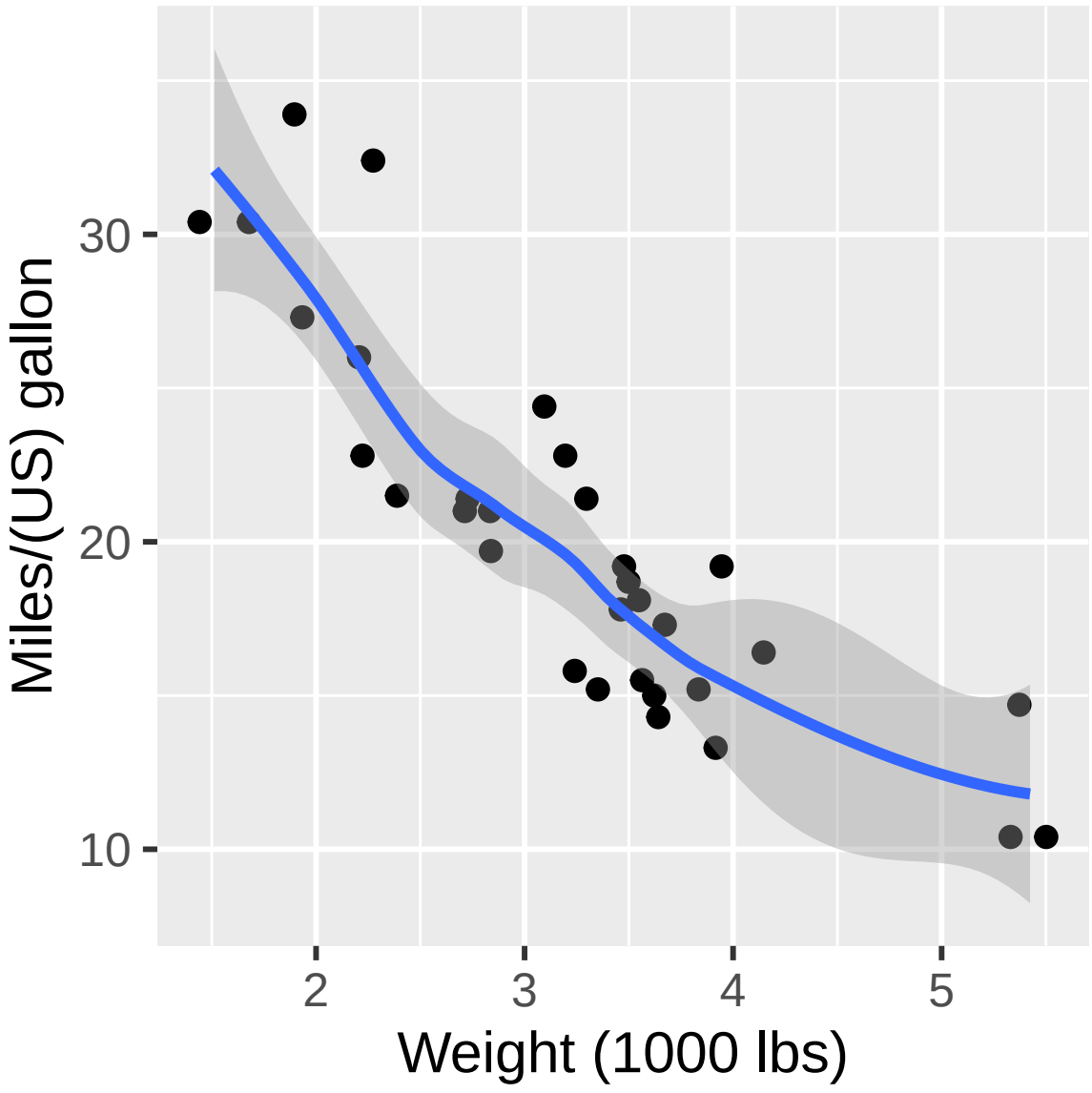
The ggplot2 package is very popular to make more sophisticated plots

```
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)
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)

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 335.6666
```

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)
  list(med=median(x),iqr=q3-q1)
}
xsamp = 1:13
mSummary(xsamp)
```

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

Monte Carlo simulations

Generate a dataset 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. In order to have access to all of the functions used throughout the text **HWC**, we need to install and load `NSM3` package.

```
install.packages("NSM3")
```

```
library(NSM3)
```

```
## Warning: package 'NSM3' was built under R version 4.1.1
```

```
## Loading required package: combinat
```

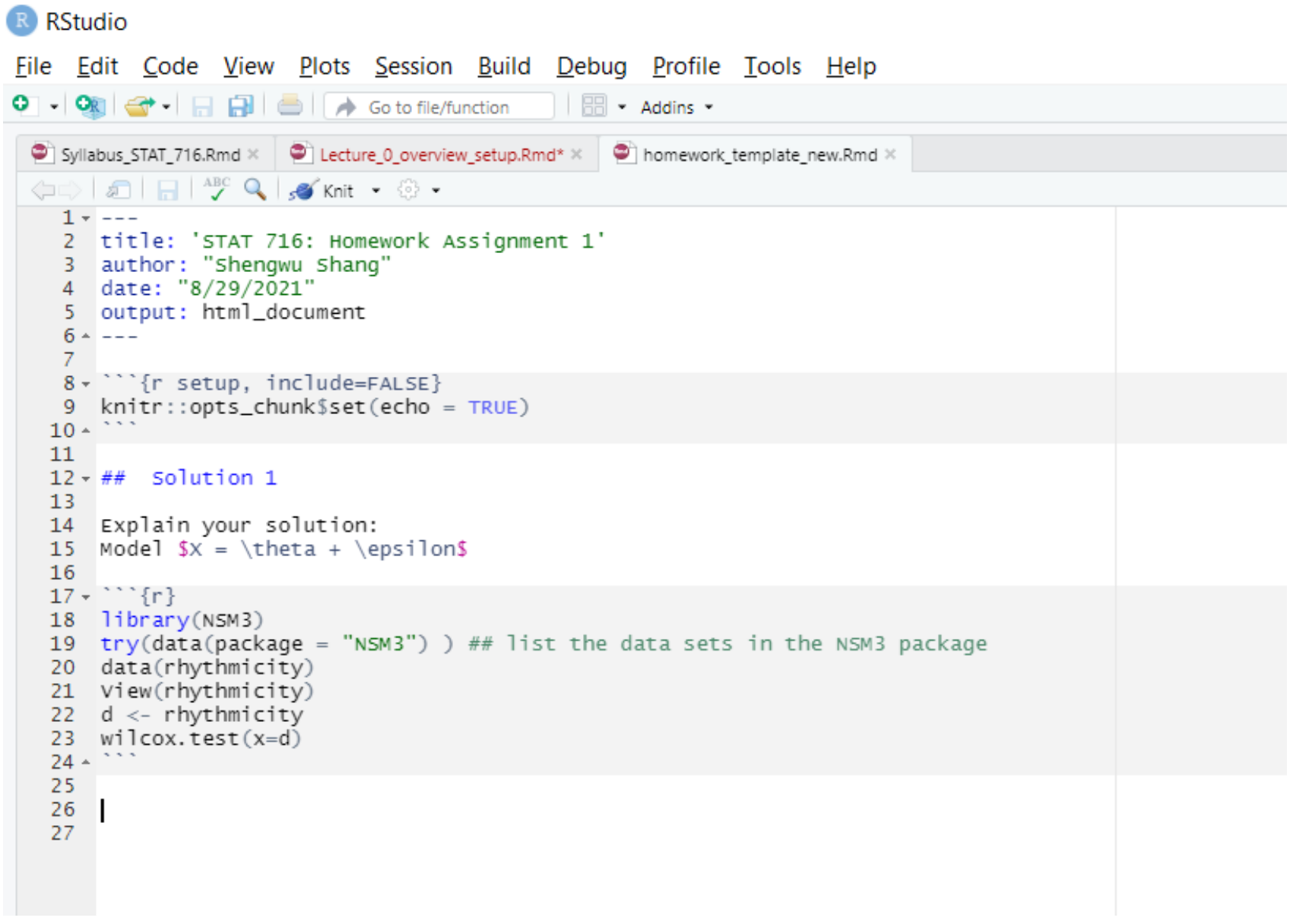
```
##
## Attaching package: 'combinat'
```

```
## The following object is masked from 'package:utils':
##
##     combn
```

Templates

Homework template

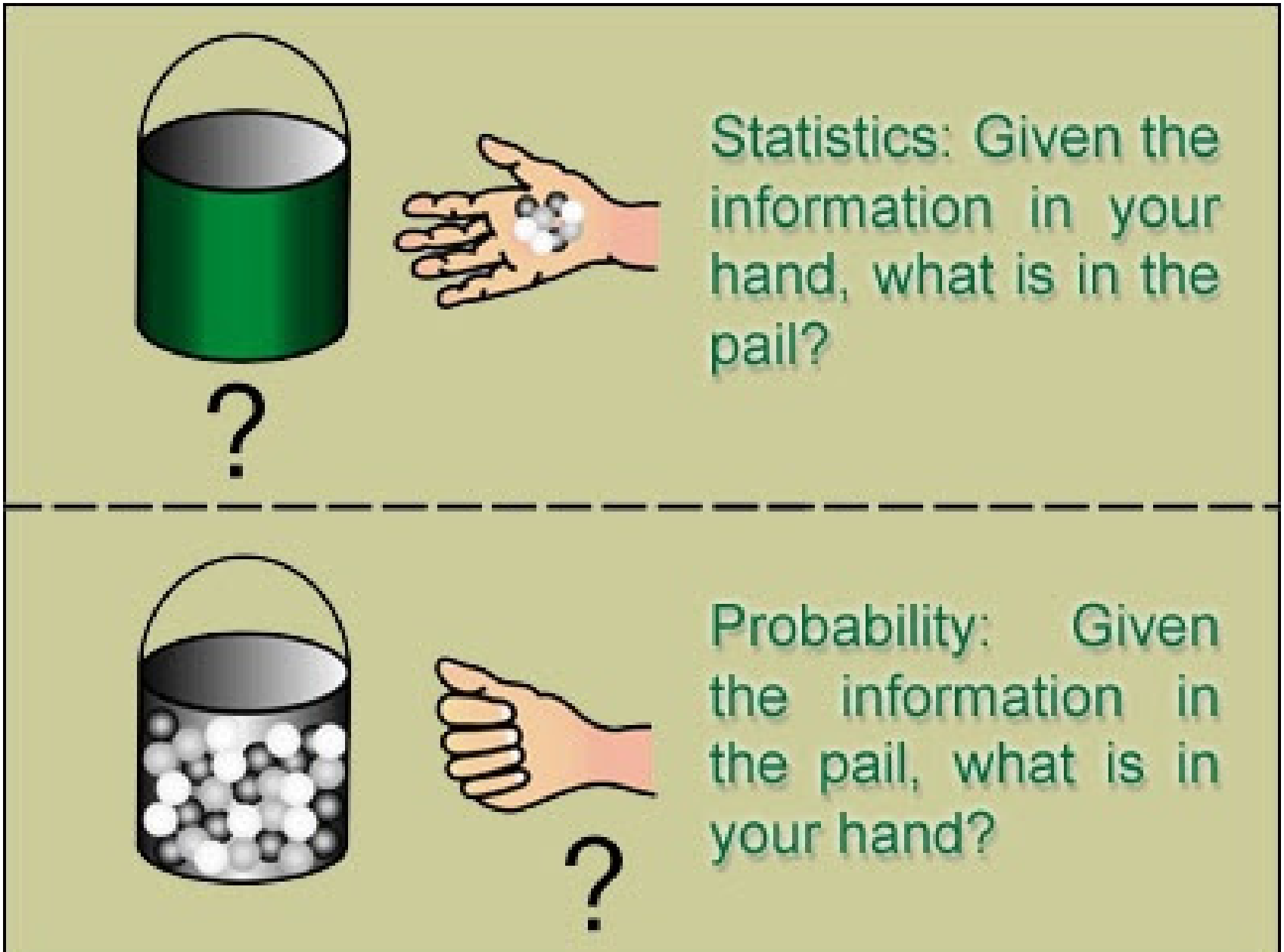
- See the template in Canvas/Files/Templates
- See the following link for a further outline of using R markdown for reporting.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Syllabus_STAT_716.Rmd
Lecture_0_overview_setup.Rmd*
homework_template_new.Rmd
Knit
---
title: 'STAT 716: Homework Assignment 1'
author: "Shengwu Shang"
date: "8/29/2021"
output: html_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
Solution 1
Explain your solution:
Model $X = \theta + \epsilon$
```{r}
library(NSM3)
try(data(package = "NSM3") ) ## list the data sets in the NSM3 package
data(rhythmicity)
View(rhythmicity)
d <- rhythmicity
wilcox.test(x=d)
```

Review

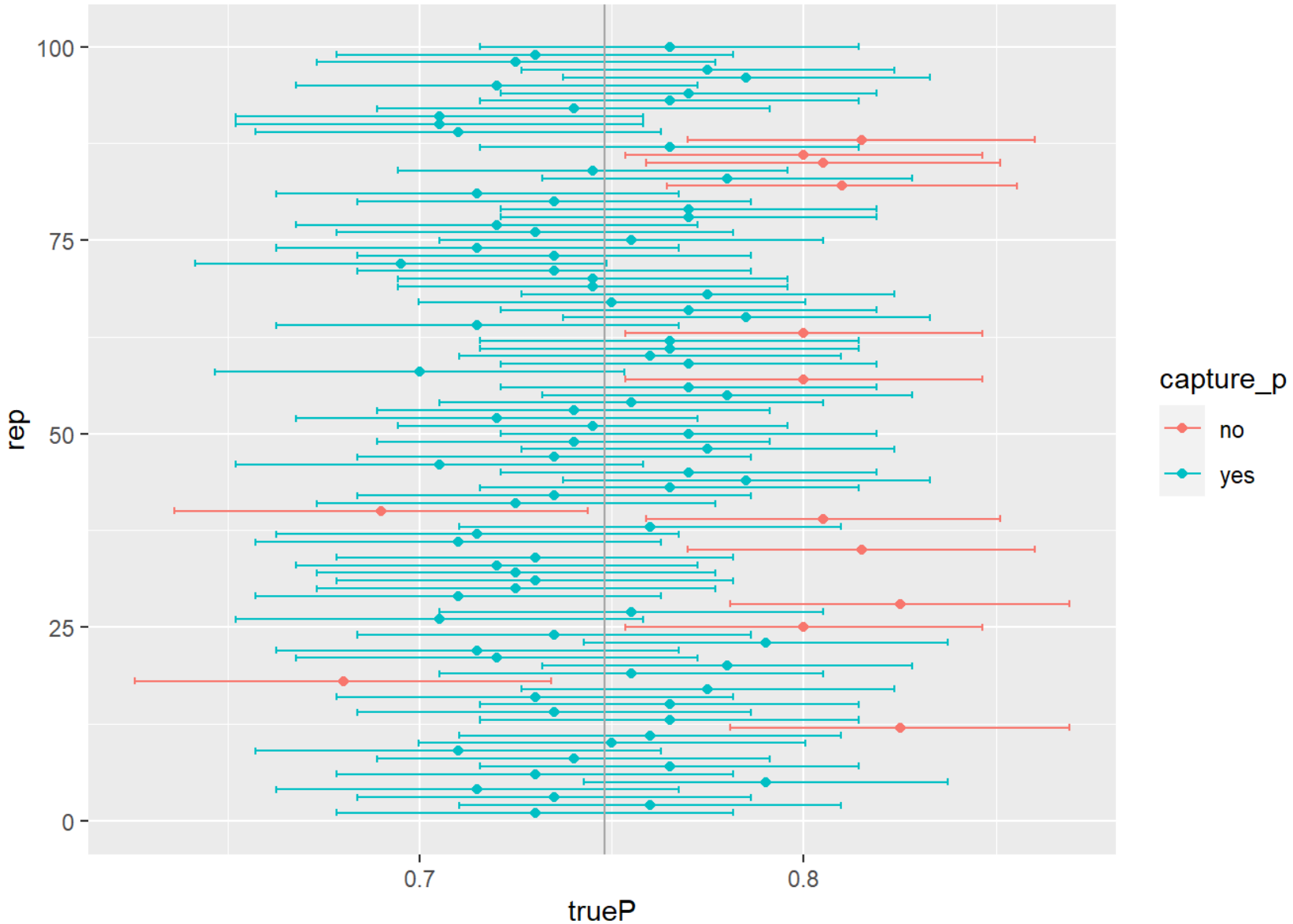
Prob. vs. Stat.: Big Picture



Terminologies

- Model vs. Method/Algorithm: Is there suach a thing as OLS/LSE model?
- statistical inference:

Confidence Interval interpretation



Read ch1 of HWC for next class!