Boosting Students' Programming Interest Using an R Shiny App Rstats in General Education Statistics Courses

Xuemao Zhang
East Stroudsburg University
The 2022 Electronic Conference on Teaching Statistics

05/25/2022

Overview

- Background of building the web app Rstats
  http://esumath.shinyapps.io/rstats
- Rstats web app layout and features
- Students usage information
  - *Students usage statistics*
  - *User experience survey results*
- Teaching Examples
  - *Univariate descriptive statistics*
  - *Simple linear regression models*
  - *Probability calculations*
  - *Quantile calculations*
  - *Statistical inferences*
- Resources and References
- The slides are available on
  https://esumath.shinyapps.io/Rstats_eCOTS

Background

- Learning to code familiarizes people with the values of a digital society: how people collaborate and share information.
- Non-computing majors generally do not take a typical programming course due to their fear of the command line.
- Introduction to Programming has not typically been an option as part of a GE course sequence at most universities before 2015 (Ferguson 2015).
- To overcome the fear of the command line for non-computing students, it is beneficial to introduce computer programming in more GE courses.
    - *Introductory statistics course is a good fit to include programming content since the application of most statistical concepts and formulas requires numerical computations.*
    - *Point-and-click statistical software packages like Minitab, SPSS and Stata are not helpful to improve a student's programming skill.*

Background

- R (R Core Team 2021), Python and SAS are the top three programming languages for statistical data analysis in data science.
  - *Both R and Python are open source*
  - *R is designed for data analysis*
  - *R has a less steep learning curve*
- More than 18,000 R add-on packages (code written to enhance the core language or solve a specific type of problems) have been developed
- The R Shiny package allows one to build a web app straight from R without using any knowledge of CSS and Javascript (Beeley 2013).
  - *Instructors have been developing Shiny web apps to help statistics students learn material more effectively Doi (2016).*
- Idea: development of a Shiny web app with Reproducibility.
  - *One web app I found is intRo: http://intro-stats.com/*
  - *The app Rstats was developed under the support of an FPDC 2021 grant.*

Background - Main techniques used in Rstats web app

- The whole app is powered by shiny.
- Reproducible R code is done with two techniques:
  - *the function interpolate and*
  - *R package shinymeta.*
- The golem framework is used to build the Shiny App backend: **Shiny modules** are used to manage complexity of the app.
  - *Each (sub)menu in the app is a module which is a small shiny app as part of the whole app.*

Rstats web app layout and features

- The R Shiny web app Rstats http://esumath.shinyapps.io/rstats is used in teaching introductory statistics in General Education.
  - *It can conduct all types of data analysis problems in the GE statistics courses with point and click interface.*
  - *It was used in online homework assignments only*
    - WebWork hosted by PASSHE Keystone Library Network https://postulate.klnpa.org/webwork2
- Reproducibility: It captures logic in a Shiny app and generates the R code which can be run on an R console.

Rstats web app layout and features

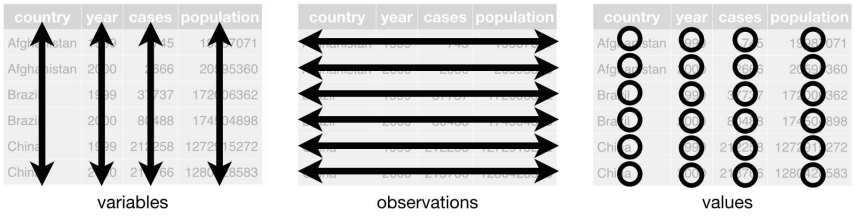
Rstats has the following six menus:

- Distributions.
  - *Calculation of probabilities and quantiles for some typical discrete and continuous distribution.*
- Inferences.
  - *Statistical inferences about population means, proportions, and variances when data or statistics are given.*
- Data Import.
  - *Data upload (Only .csv format files are accepted) or manually data entry (up to 5 variables), and data transformation.*
- Univariate.
  - *Univariate data analysis can be done for the imported data.*
- Multivariate.
  - *Simple and multiple linear regression models, logistic regression models, and contingency analysis.*
- ANOVA.
  - *Analysis of variance with one or two factors.*

Rstats web app layout and features

Data format:

- Data need to be entered as vectors under the menu *Inferences*
- Data imported or manually entered data need to be tidy (Wickham and Grolemund, 2017) data frames:
  - *Each variable must have its own column.*
  - *Each observation must have its own row.*
  - *Each value must have its own cell.*



Students usage statistics

Shiny applications not supported in static R Markdown documents

User experience survey results

Shiny applications not supported in static R Markdown documents

Teaching Examples

What we need today:

- Rstats: http://esumath.shinyapps.io/rstats
- R and Rstudio
  - *Installation of R and Rstudio (for later) - https://stat545.com/install.html*
- Let's use RStudio Cloud if you want to try to run the R code in an R console
  - *Rstudio cloud link:* ***eCOTS_2022****: https://rstudio.cloud/spaces/247851/join?access_code=7Os1gcQobAjYvIrf2SJmkZFg9eKPnZuw6XamxKs1*
  - *Click the button* ***Join Space****, and click Projects->Rstats*
  - *The R code in the slides are in the file examples.Rmd. You can use the keyboard shortcut 'Ctrl+Enter' to run the code line by line or paste the R code from the slides to the R Console.*
  - *The following R packages are installed to the project already*
    - `install.packages("datarium")`
    - `install.packages("dplyr")`
    - `install.packages("ggplot2")`

Teaching Examples - Descriptive Statistics

- **Example 1**: Find the mean,median,variance,standard deviation of the data 5, 15, 25, 35, 45
  - *Data entry: import the data set as a .csv file or enter it manually*
  - *Data analysis:* ***Univariate*** *->* ***Numerical Data Analysis***

```
Data <- data.frame(Y = c(5, 15, 25, 35, 45))
summary(Data$Y)
mean(Data$Y)
var(Data$Y)
sd(Data$Y)
```

Teaching Examples - Descriptive Statistics

- **Example 2**: Find the mean, variance and standard deviation of a frequency/relative frequency table

Y	Freq
5	4
15	9
25	6
35	4
45	2

- Data entry: Let's enter the data as two columns under **Distributions** -> **Probability calculation** -> *Finite*

Teaching Examples - Descriptive Statistics

- Example 2 continued
  - *Data analysis*

```
x = c(5, 15, 25, 35, 45)
w = c(4, 9, 6, 4, 2)
mu <- sum(x * w) / sum(w)
cat("Population/Sample mean:", mu)
sigma2 <- sum((x - mu)^2 * w) / sum(w)
cat("Population variance:", sigma2)
sigma <- sqrt(sigma2)
cat("Population standard deviation:", sigma)
s2 <- sum((x - mu)^2 * w) / (sum(x) - 1)
cat("Sample variance:", s2)
s <- sqrt(s2)
cat("Sample standard deviation:", s)
```

Teaching Examples - SLR model

- **Example**: Use Price as the response variable, draw a scatterplot, calculate the linear correlation coefficient and fit the SLR model

Capacity (in TB)	Price (in $)
0.080	29.95
0.120	35.00
0.200	299.00
0.250	49.95
0.320	69.95
1.0	99.00
2.0	205.00
4.0	449.00

Teaching Examples - SLR model

- Data entry: import the data set as a .csv file or enter it manually
- Data analysis: **Multivariate -> SLR Model**

```
Data = data.frame(Y = c(29.95, 35, 299, 49.95, 69.95, 99, 205,
         449),
                   X = c(0.08, 0.12, 0.2, 0.25, 0.32, 1, 2, 4))
cor(y = Data$Y, x = Data$X, method = "pearson")
ggplot(Data, aes(y = Y, x = X)) +
  geom_point() +
  geom_smooth(method = lm, formula = y ~x,
              se = TRUE, level = 0.95) +
  labs(title = "Plot of fit with confidence band")
fit = lm(Y ~ X, data = Data)
summary(fit)
```

Teaching Examples - Probability calculations

Distributions -> Probability calculation

$X \sim N(\mu = 1, \sigma = 1.5)$. Find the following probabilities

- $P(X < 1.8)$

```
pnorm(1.8, mean = 1L, sd = 1.5, lower.tail = TRUE)
```

- $P(X > 1.8)$

```
pnorm(1.8, mean = 1L, sd = 1.5, lower.tail = FALSE)
```

- $P(0.5 < X < 1.8)$

```
pnorm(1.8, mean = 1L, sd = 1.5, lower.tail = TRUE) -
  pnorm(0.5, mean = 1L, sd = 1.5, lower.tail = TRUE)
```

- You can try other distributions

Teaching Examples - Quantile calculations

Distributions -> Quantile calculation

$X \sim N(\mu = 1, \sigma = 1.5)$.

- Find x such that $P(X < x) = 0.05$

```
qnorm(0.05, mean = 1L, sd = 1.5, lower.tail = TRUE)
```

- Find x such that $P(X > x) = 0.05$

```
qnorm(0.05, mean = 1L, sd = 1.5, lower.tail = FALSE)
```

- Again, you can try other continuous distributions

Teaching Examples - Statistical Inferences

- Method 1: use the menu **Inferences** ->
  - -> ***Statistics*** *if summary statistics of a data set are given*
  - -> ***Data*** *if detailed data set is given; data must be entered as vectors.*
- Method 2: Use menu **Data Import** to import data or manually enter data and use the following three menus to conduct data analysis
  - ***Univariate*** *for descriptive statistics and inferences about a single population parameter*
  - ***Multivariate***
  - ***ANOVA***
- Let's consider statistical inference about population means using Method 1.

Teaching Examples - Statistical Inferences

Example: A simple random sample of 15-year old boys from one city is obtained and their weights (in pounds) are listed below. Suppose the sample is selected from a normal population

146, 140, 160, 151, 134, 189, 157, 144, 175, 127, 164

Inferences -> Data

- Find a 96% confidence interval (2-sided symmetric) of the population mean

```
(dat = c(146, 140, 160, 151, 134, 189, 157, 144, 175, 127, 164))
test <- t.test(x = dat, mu = 147L,
          alternative = "two.sided", conf.level = 1 - 0.04)
CI <- test$conf.int
cat("Confidence interval", "of level", 1 - 0.04, "is :", CI)
```

Teaching Examples - Statistical Inferences

- **Test 1**: Test the claim that these sample weights come from a population with a mean greater than 147 lb. Use significance level $\alpha = 0.05$.
  - $H_0 : \mu = 147$ *versus* $H_0 : \mu > 147$
  -

```
(dat = c(146, 140, 160, 151, 134, 189, 157, 144, 175, 127,
         164))
test <- t.test(x = dat, mu = 147L,
               alternative = "greater", conf.level = 1 - 0.05)
CI <- test$conf.int
cat("Confidence interval", "of level", 1 - 0.05, "is :",
         CI)
test
```

Teaching Examples - Statistical Inferences

- **Test 2**: Test the claim that these sample weights come from a population with a mean greater than or equal to 147 lb. Use significance level $\alpha = 0.05$.

  - $H_0 : \mu = 147$ *versus* $H_0 : \mu < 147$
  -

```
(dat = c(146, 140, 160, 151, 134, 189, 157, 144, 175, 127,
         164))
test <- t.test(x = dat, mu = 147L,
               alternative = "less", conf.level = 1 - 0.05)
CI <- test$conf.int
cat("Confidence interval", "of level", 1 - 0.05, "is :",
         CI)
test
```

Teaching Examples - Statistical Inferences

- **Test 3**: Test the claim that these sample weights come from a population with a mean equal to 147 lb. Use significance level $\alpha = 0.05$.

  - $H_0 : \mu = 147$ *versus* $H_0 : \mu \neq 147$
  -

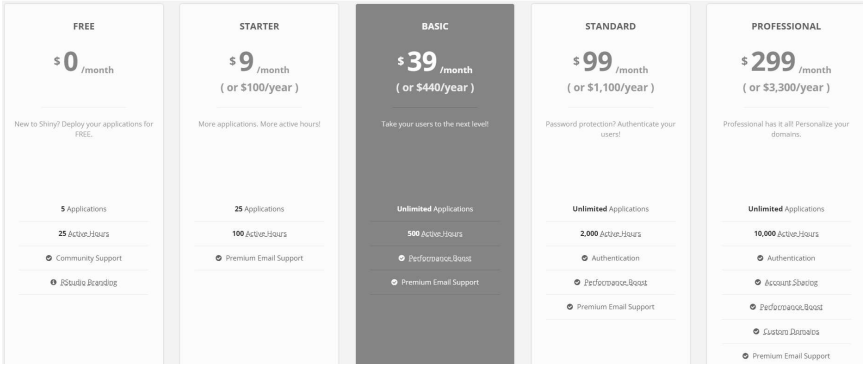
```
(dat = c(146, 140, 160, 151, 134, 189, 157, 144, 175, 127,
         164))
test <- t.test(x = dat, mu = 147L,
               alternative = "two.sided", conf.level = 1 -
         0.05)
CI <- test$conf.int
cat("Confidence interval", "of level", 1 - 0.05, "is :",
         CI)
test
```

Some comments about the app Rstats

- It is for introductory statistics only
- Data manipulation functionality is limited
  - *It can perform several data transformations*
    - Power and log transformation for numerical variables
    - Transform between numerical variables and categorical variables
  - *It cannot perform data cleaning*
    - Missing values can be addressed

Rstats source code

- Rstats source code: https://github.com/esumath/Rstats
- You can download the source code and deploy the web app to a server such as www.shinyapps.io.



References

- Beeley, C. (2013). *Web Application Development with R Using Shiny*, Packt Publishing, Birmingham, UK.
- Doi, J., Potter, G., Wong, J., Alcaraz, I. and Chi, P. (2016). Web Application Teaching Tools for Statistics Using R and Shiny, *Technology Innovations in Statistics Education*, 9(1), 1-32.
- Ferguson, R., Leidig, P. and Reynolds, J. (2015). Including a Programming Course in General Education: Are We Doing Enough?, *Information Systems Education Journal*, 13, 34-42.
- Hare, E. and Kaplan, A. (2017). Designing Modular Software: A Case Study in Introductory Statistics, *Journal of Computational and Graphical Statistics*, 26(3), 493-500.
- Wickham, H. and Grolemund, G. (2017). *R for Data Science*. O'Reilly Media.
- R Core Team (2021). *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria,