Technology Help: RStudio

Luc Hens

In Statistics for Business and Economics I and II, in Econometrics, and in other courses you will use RStudio and R to display data and do statistical computations. This document complements the end-of-chapter Technology Help sections in Sharpe et al. (2015) by providing information on how to use RStudio and R to do the most common statistical computations in Statistics I. Print the document (one-sided), cut out the sections for each chapter, and paste them after the introduction (which you should read) of the Technology Help section in each chapter of your textbook.

Technology Help: Data (Ch. 1)

RStudio

To open a text file with data in RStudio:

- Select **File** > **Import Dataset** > **From Text (base)...** (Or select in the right panel the tab **Environment** and select: **Import Dataset** > **From Text (base)...**)
- Browse to locate the file you want to import. The data sets of the textbook are available as plain text files that use tabs to separate variables (such files usually have a `.txt` extension). Often commas (or other characterS such as a semicolon) are used to separate variables. In that case the files are called *comma-separated values-* or *character-separated values*-text files, and the extension `.csv` is used.
- RStudio opens a dialog window. The panel at the bottom right shows how RStudio will import the data. The variable names are in boldface. If the data table in the panel at the bottom right does not look as you expected, you may have to change one or more of the options shown in the left panel.
- Click the **Import** button at the bottom. The panel at the top left displays the data table.

The R-instructions that we will use in the following chapters are shorter if we attach the name of the data set to the search path. Type in the console:

```
attach(name.of.data.set)
```

and press the Enter of Return key. (Instead of `name.of.data.set` use the name of the data set you can see in the tab on the top left of the data table panel.)

Technology Help: Displaying categorical data (Ch. 2)

RStudio

To make a frequenty table, bar chart, or pie chart:

- Download the SuperBowl.txt data file that contains the data shown in the table on top of p. 49 from the learning platform and import the data using **File** > **Import Dataset** > **From Text (base)...**. Then type in the console:

  ```
  attach(SuperBowl)
  ```

  and press the Enter or Return key.

- Frequency table: type in the console:

  ```
  table(my.data)
  ```

  and press the Enter or Return key.

- Bar chart: type in the console:

  ```
  barplot(table(my.data), las=1)
  ```

  and press the Enter or Return key. The bar chart appears in the bottom right panel in the **Plots** tab. (The `las=1` option is used to display the labels on the vertical axis in a reader-friendly orientation.) Select **Zoom** to see the chart at full size. To order the bars from small to large, type:

  ```
  barplot(sort(table(my.data)), las=1)
  ```

  To make a bar chart of the relative frequencies (expressed as percentages), first convert the frequency table to relative frequencies:

  ```
  my.data.as.percentages <- table(my.data)/sum(table(my.data))*100
  my.data.as.percentages
  barplot(my.data.as.percentages, las=1)
  ```

- Pie chart: type in the console:

  ```
  pie(table(my.data))
  ```

  and press the Enter or Return key. The chart appears in the bottom right panel in the **Plots** tab. Select **Zoom** to see the chart at full size.

The commands are similar if you don't have the data table but do have a frequency table (such as table 2.1 on p. 48). First use the assignment operator `<-` and the concatenate operator `c( )` to create lists of the categories and the corresponding frequencies:

```
Source <- c("Google", "Direct", "E-mail", "Bing", "Yahoo", "Facebook", "Mobile", "Other")
Visits <- c( 130158 ,  52969  ,  16084   , 9581 ,  7439  ,  2253     ,  1701   ,  6740  )
```

and press the Enter or Return key.

- Bar chart:

  ```
  barplot(Visits, names.arg = Source, las=1)
  ```

- Bar chart of relative frequencies:

  ```
  Visits.as.percentage.of.total <- (Visits/sum(Visits))*100
  Visits.as.percentage.of.total
  barplot(Visits.as.percentage.of.total, names.arg = Source,
           las=1, ylab="Percentage")
  ```

- Pie chart:

  ```
  pie(Visits.as.a.percentage.of.total, labels = Source)
  ```

Technology Help: Displaying and summarizing quantitative data (Ch. 3)

RStudio

To make a histogram of the variable `x`, type in the Console window:

```
hist(x,las=1, col="Gray")
```

and press the Enter or Return key. (The `las=1` option is used to display the labels on the vertical axis in a reader-friendly orientation.) This gives a histogram with the count on the vertical axis. To make a histogram of the variable `x` with the relative frequency (expressed as a percentage), type in the Console window:

```
h <- hist(x, plot=FALSE)
h$counts <- 100*h$counts / sum(h$counts)
plot(h, freq=TRUE, ylab="Relative Frequency (%)", las=1, col="Gray")
```

To find the mean of `x`, type in the Console window:

```
mean(x)
```

and press the Enter or Return key.

To find the standard deviation of `x`, type in the Console window:

```
sd(x)
```

and press the Enter or Return key.

To find the five-number summary (plus the mean) of `x`, type in the Console window:

```
summary(x)
```

and press the Enter or Return key.

to make a boxplot of `x`, type in the Console window:

```
boxplot(x)
```

and press the Enter or Return key.

To make side-by-side boxplots of two variables (`x.1` and `x.2`), type in the Console window:

```
boxplot(x.1,x.2)
```

and press the Enter or Return key.

Technology Help: Correlation and Regression (Ch. 4)

RStudio

To make a scatter plot of two variables (`x` on the x-axis and `y` on the y-axis), typ in the Console window:

```
plot(x, y, las=1)
```

and press the Enter of Return key. (The option `las=1` ensures that the labels on the vertical axis are oriented correctly.)

To get the correlation coefficient between two variables (`x` and `y`) type in the Console window:

```
cor(x, y)
```

and press the Enter of Return key.

To get the Spearman rank correlation coefficient between two variables `x` and `y` type in the Console window:

```
cor(x, y, method="spearman")
```

and press the Enter of Return key.

To compute a regression, use the `lm` function in R (`lm` stands for linear model). To find the regression line (line of best fit) between the dependent variable `y` and the independent variable `x` and store the results in an object called `my.model`, type in the Console window:

`my.model <- lm(y ~ x)`

and press the Enter of Return key. The $\sim$ symbol between `y` and `x` is called a tilde. To get a summary of the results, type in the Console window:

```
summary(my.model)
```

and press the Enter of Return key. The intercept is reported in the `Coefficients` table in the `Estimate` column next to `(Intercept)`. The slope is reported in the `Coefficients` table in the `Estimate` column next to the name of the variable. The R^2 is `Multiple R-squared`. The residuals are stored as `my.model$residuals`, the predicted (or fitted) values for the dependent variable as `my.model$fitted.values`.

To make a scatter plot of two variables (the independent variable `x` on the x-axis and the dependent variable or response variable `y` on the y-axis) with the regression line, type in the Console window:

```
plot(x, y, las=1)
abline(lm(y ~ x))
```

and press the Enter of Return key. (The option `las=1` ensures that the labels on the vertical axis are oriented correctly.)

To take the natural logarithm of a variable `x` do: `log(x)`.

Technology Help: Generating random numbers (Ch. 5)

RStudio

R can generate random numbers with the `runif()` function. To generate a random number in R:

- In the Console window, type `runif(1)` and press the Enter or Return key. A random number between 0 and 1 (a real number to 7 decimal places) appears in the Console window.
- To generate more (for instance, 10) random numbers, type `runif(10)`
- To generate a random number within a range [**a**, **b**], where **a** is the number at the low end of the range and **b** is number at the high end of the range, type `runif()*(b-a)+a`. *E.g.*, to generate a random number between 100 and 200 type:

  ```
  a <- 100
  b <- 200
  runif(1)*(b-a)+a
  ```

- You can also use the function `sample(a:b,1,replace=TRUE)` to generate an integer between **a** and **b**.

Random numbers are re-generated every time you repeat the `runif()`-command. To avoid this, you can store the generated random numbers in a list:

```
list.of.random.numbers <- runif(10)
```

To display the list:

```
list.of.random.numbers
```

Technology Help: Random variables and probability models (Ch. 6)

RStudio

- `dgeom(x, prob=...)` for the geometric probability density function (pdf) (we skipped the geometric distribution)
- `dbinom(k, size=..., prob=...)` for the binomial probability density function (pdf)
- `pbinom(k, size=..., prob=...)` for the binomial cumulative distribution function (cdf)
- `dpois(x, lambda=...)` for the Poisson probability density function (pdf) (we skipped the Poisson distribution)
- `ppois(x, lambda=...)` for the Poisson cumulative distribution function (cdf) (we skipped the Poisson distribution)

Technology Help: Probabilility calculations and plots (Ch. 7)

RStudio

R can make Normal probability plots (R calls these Q-Q plots, which stands for quantile-quantile plots). First import your data and attach the data set (see the Technology Help for Chapter 1). To make a Normal probability plot of variable `x`, type in the Console:

```
qqnorm(x, las=1)
```

and press the Enter or Return key. (The option `las=1` ensures that the labels on the vertical axis are oriented correctly.)

To calculate continuous distribution probabilities in R:

- `dnorm(x)` for the standard normal probability density function (pdf)
- `pnorm(x)` for the standard normal cumulative distribution function (cdf)
- `qnorm(p)` for the p^{th} quantile of the standard normal distribution. Express p as a decimal fraction, e.g., to find the 4^{th} percentile use `qnorm(0.04)`.

- To find the area under the standard normal curve between **a** and **b**, where **a** is the number at the low end of the range and **b** is number at the high end of the range, type `pnorm(b)-pnorm(a)`
- `dexp(x,lambda=...)` for the exponential probability density function (pdf) (we skipped the exponential distribution)
- `pexp(x,lambda=...)` for the exponential cumulative distribution function (cdf) (we skipped the exponential distribution)

Technology Help: Random sampling (Ch. 8)

RStudio

The R function `sample(a:b,1)` draws an integer without replacement from a uniform distribution between **a** and **b**. To generate a list of integers, change the 1 into how many random numbers you want to draw. *E.g.*, to draw 60 random integers between 1 and 1000 without replacement and store them in a list called `my.list`, type in the Console window:

```
my.list <- sample(1:1000,60)
```

and press the Enter or Return key. To display the list, type in the Console window:

```
my.list
```

and press the Enter or Return key. To sort the list from small to large, type in the Console window:

```
sort(my.list)
```

and press the Enter or Return key.

To draw (without replacement) a sample of size n from a variable `x` in RStudio, type in the Console window:

```
sample(x, n, replace = FALSE)
```

and press the Enter or Return key.

To draw with replacement, add the option `replace=TRUE` in the `sample()` function.

Technology Help: Confidence intervals for proportions (Ch. 9)

RStudio

To find a 95% confidence interval for the true proportion when there were k successes in a sample of size n, type in the Console window:

```
binom.test(k, n=..., conf.level=0.95)
```

and press the Enter or Return key.

Comments

The method shown here will work for summarized data. When working with raw data, use the `table()` function to count values to compute the sample proportion.

Technology Help: Confidence intervals for means (Ch. 11)

RStudio

To find a 95% confidence interval for the true mean of variable `x`, type in the Console window:

```
t.test(x, conf.level=0.95)
```

and press the Enter or Return key.

References

Sharpe, N. R., De Veaux, R., and Velleman, P. (2015). *Business Statistics*. Pearson Education, 3rd edition.