Role of the computer

Deborah Nolan
University of California, Berkeley

Overview

- Background & Motivation for change
- Examples of how to use R in teaching
- Feedback from students
- Introductory lessons in R

Background & Motivation

Critical point in statistics.

- Computing is becoming increasingly vital part of statistical in this era of
  - Ubiquitous data availability & sources.
  - Increased volume and complexity of data.
  - New and ever-evolving Web technologies.
  - Increased relevance of data analysis in all fields, done by non-statisticians
  - Communicating results in new ways

Computational Science

- "Computation is now regarded as an equal and indispensable partner, along with theory and experiment, in the advance of scientific knowledge" (SIAM Working Group Computational Science & Engineering Education, 2001).
- Computing is an essential, foundational skill for modern data analysis and statistics research
- Friedman ('97): Statistics is defined by a set of tools
  - Probability, real analysis, asymptotics,..
  - Computing has been the most glaring omission from the set of tools

Preparation for work/research

- Do our students have the essential skills needed to engage in collaborative research, data-driven decision making, and problem solving?
- Do our students have the confidence needed to overcome computational challenges to carry out a comprehensive data analysis?
- Are our students ready to engage in and succeed at statistical inquiry?

How to use computing in the introductory course?

Tool for understanding concepts

- Probability calculations
  - Probabilities for known distributions
- Simulation study
  - Approximate distributions
  - Comparison observed phenomena against model
  - Study properties of statistics
- Bootstrapping
  - Use observed data to study sampling distribution

Tool for data analysis

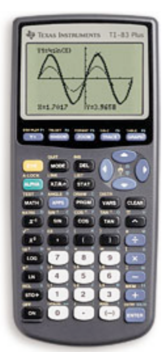
- Exploratory data analysis (Graphics slides)
- Presentation Graphics (Graphics slides)
- Hypothesis Testing – permutation tests
- Bootstrap Confidence Intervals
- Modeling

Why R ?

- Allows custom analysis
- High-level scripting language
- Statistical programming language
- Interactive exploratory data analysis
- Easy to replicate analysis
- Sound numerical methods
- Large Community of contributors

Secondary School Statistics Education

- Calculators harder to use than R
- Graphical capabilities include histograms, boxplots, scatter plots
- Most college students have [pre]-calculus background

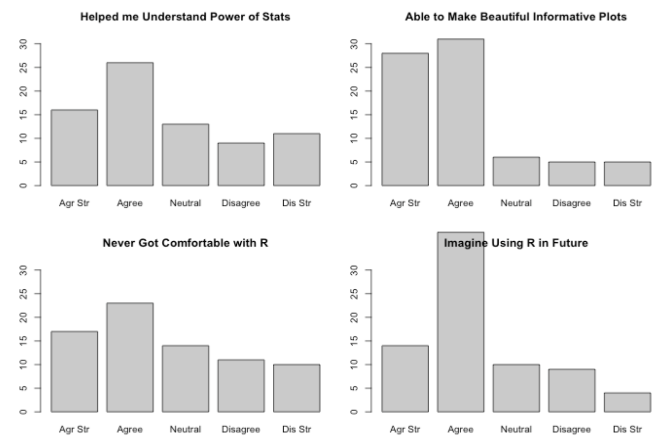


Care with R taught

- Pre-Calculus: familiar with functions f(x) g(x,y)
- Emphasize connections to statistics
  - Vector as a variable
  - Factor represents nominal/ordinal data
  - Missing data and NA
- Connect R simulations to physical examples
  - replicate, sample
- Avoid programming
- Use base graphics

Student Feedback

Usefulness of R (75 of 111)

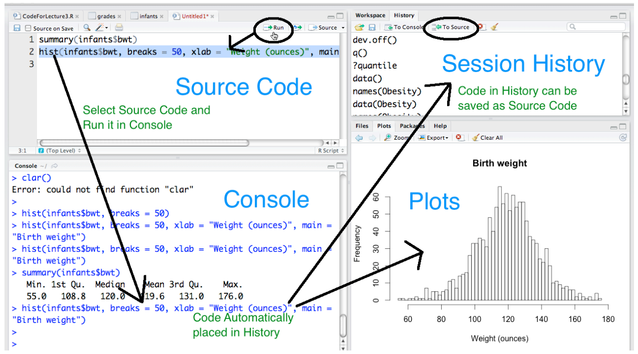


Main Topics in Teaching R

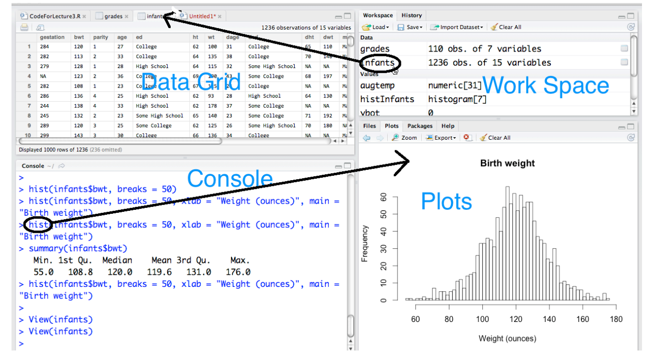
- Using R as a calculator
- Measurements on a variable are stored in a vector;
  - Vector operations
  - Data types
  - Missing values
- Calling functions
- Organizing variables into a data frame
- Subsetting vectors and data frames

Teach from the Statistician's perspective:
Computer a Tool to Work with data

Rstudio Environment



RStudio



Statistician's perspective:
Want an interactive environment for exploration

Using R as a calculator

The Prompt

- The R prompt is: >
- At the prompt, type an ***expression***
- Hit the return/enter key
- R ***evaluates*** the expression (performs a ***computation***)
- R returns a value

```
> 2 + 3
[1] 5
Returns 5
```

```
> 2 * 4
[1] 8
Returns 8
```

What do expressions look like?

```
2 + 3
9 - 8
4 * 5
10 / 3
2 + (7 ^ 2)/3
```

Order of operations

Order of operations is what you expect, i.e. exponentiation first, followed by multiplication and division, then addition and subtraction; left to right; parentheses override order

```
> 1 + 2*3
[1] 7
> (1+ 2) * 3
[1] 9
```

Functions in R

R has some arithmetic functions, e.g. log, sin

```
> log(100)
[1] 4.60517

> log10(100)
[1] 2
```

Variables in R

Variable

- A variable contains measurements, e.g. daily temperature (degrees Farenheit) in June

```
> junetemp
 [1] 81 73 86 74 84 75 70 73 66 68 62 64 65
[14] 62 61 66 70 73 72 82 72 75 69 70 66 69
[27] 73 71 68 67
```

- We call a variable in R a vector
  - They are ordered containers.
  - There are 30 values in `junetemp` the first is 81 and 30^{th} is 67

Vectors

- A vector is an ordered container of a set of values/measurements
- The values must be all the same type of information

junetemp
Vector

81
73
.
.
.
68
67

Vector calculations

- Convert temperature from Farenheit to Celsius
- Formula:

C = (F – 32) * 5/9

- Element-wise calculation

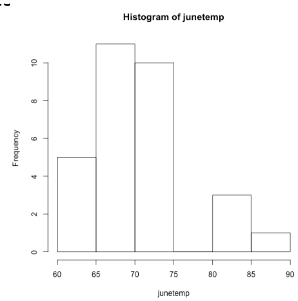
junetemp
Vector

junetempC = ([81, 73, . . . , 68, 67] - 32) * 5/9

Operating on Vectors

> junetempC = (junetemp – 32) * 5/ 9
> junetempC
[1] 27.22 22.77 30.00 …. 20.00 19.44

> min(junetemp)
[1] 61
> mean(junetemp)
[1] 70.9
> hist(junetemp)



Statistician's perspective:
x is a Variable is a Vector of values

Data Types

Vectors

- We have data on a 14-member family –
- For each person we have his/her
  - name, age, gender, weight, height, and whether or not he/she is over weight (BMI > 25)

> name
[1] "Tom" "May" "Joe" "Bob" "Sue" "Liz" "Jon" "Sal"
[9] "Tim" "Tom" "Ann" "Dan" "Art" "Zoe"
> age
[1] 77 33 79 47 27 33 67 52 59 27 55 24 46 48

Family information

> gender

[1] m f m m f f m f m m f m m f

Levels: m f

> overWt

```
 [1]  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
[8]  FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
```

These Variables have different Data Types

- `age`: numeric
- `name`: character string
- `overWt`: A *logical* vector contains values that are either TRUE or FALSE.
- `gender`*: factor* vector is a special type used for qualitative data. The values are stored as integers but each integer corresponds to a *level*, which is a character string

> levels(gender)

[1] "m" "f"

Missing Values

- The notion of a Missing value is important in statistics
- The missing value symbol in R is NA
- It stands for "Not Available"
- NA can be an element of a vector of any type

Statistician:
Data types reflect the differences a statistician cares about for a data analysis

Data Frames

The Family

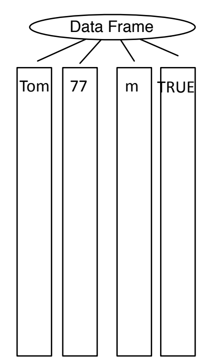
- We have all sorts of information about our family, height, weight, first name, gender, …
- Each set of measurements is stored in a vector, e.g., all names are in `name`
- The first value of each vector is a measurement on the same person in the family, in this case Tom, the second value is a measurement on May, and so on.

```
> family
   firstName gender age height weight      bmi overWt
1        Tom      m  77     70    175 25.16239   TRUE
2        May      f  33     64    125 21.50106  FALSE
3        Joe      m  79     73    185 24.45884  FALSE
4        Bob      m  47     67    156 24.48414  FALSE
5        Sue      f  27     64    105 18.06089  FALSE
6        Liz      f  33     68    190 28.94981   TRUE
7        Jon      m  67     68    185 28.18797   TRUE
8        Sal      f  52     65    124 20.67783  FALSE
9        Tim      m  59     68    175 26.66430   TRUE
10       Tom      m  27     71    215 30.04911   TRUE
11       Ann      f  55     67    166 26.05364   TRUE
12       Dan      m  24     66    140 22.64384  FALSE
13       Art      m  46     66    150 24.26126  FALSE
14       Zoe      f  48     62    125 22.91060  FALSE
```

Vectors

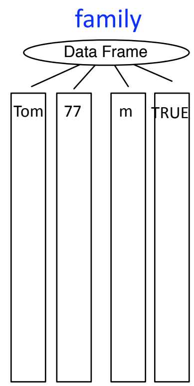
The data frame gives us a way to collect all of these variables (vectors) into one object.

```
> name
 [1] "Tom" "May" "Joe"...
> age
 [1] 77 33 79 ...
> gender
 [1] m f m ...
> overWt
 [1]  TRUE FALSE FALSE ...
```



Data Frame

- *Ordered* container of vectors
- Vectors must all be the *same length*
- Vectors in a data frame can be *different types*



dataframe$vector

We can refer to a vector in the data frame as follows:

> **family$gender**

[1] m f m m f f m f m m f m m f

Levels: m f

> **mean(family$height)**

[1] 67.07143

Missing Values

- Important concept of "missing" in statistics.
- Represented as the literal/constant NA
- Why is 1 + NA an NA?

```
> 1 + NA
[1] NA
```

- Why is the average value for sex and NA?

```
> mean(sex)
[1] NA
Warning message:
In mean.default(sex) : argument is not numeric or logical:
  returning NA
```

Statistician's perspective:
A matrix and data frame are different concepts
With data frames, rows and columns have different meanings, columns are not same type

Functions

Calling Functions

If you understand functions in math, then functions in R are easy. The syntax for calling a function is:

`functioname( argument)`

To add up all of the elements in `junetemp`:

```
> sum(junetemp)
[1] 2127
```

To average all of the elements in `junetemp`:

```
> mean(junetemp)
[1] 70.9
```

Calling Functions

Summary of age:

```
> summary(age)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
24.00   33.00   47.50   48.14   58.00   79.00
```

Summary of gender:

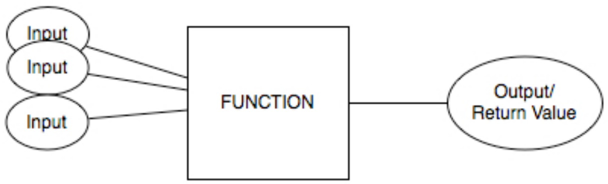
```
>  summary(gender)
m f
8 6
```

Why does the summary function behave differently for age and gender?

Statistician's perspective:
same function applied to different data type may behave differently,
NA values need care

Functions

Syntax: FuntionName(input, input, input)



Assign the Return value to an R object:

```
x = function(input, input, input)
```

Inputs to a function

- The inputs are called the **arguments** to the function
- Some arguments are required.
- Some arguments are optional, meaning if the input is not provided then a default value is used
- Arguments have names.

`mean()`

Let's take a look at the function definition

```
mean(x, trim = 0, na.rm = FALSE, …)
```

There are three arguments

x – is required because it has no default value

trim – is not required; it's default is 0

na.rm – is not required; it's default is FALSE

Arguments to `mean()`

- When you read the help information for mean, you find what the function expects for each input
- x – is a numeric type; the function takes the mean of this information
- trim – a fraction between 0 and 0.5 that specifies how much of the data to trim away before taking the mean
- na.rm – tells the function whether to remove the Nas in x before taking the mean or not

Invoke the function

- We **call** the function, to find the average time:

```
> mean(x = junetemp)
```
[1] 70.9

- Call it again, and this time trim away the largest and smallest 10% of the data before taking the means

```
> mean(x = time, trim = 0.1)
```
[1] 70.4   Why does the mean get smaller?

Argument Matching

Can pass arguments by name or by position (order)

```
> mean(temp)    # equivalent to mean(x = temp)
[1] 70.9
> mean(temp, 0.1)
# same as mean(x = temp, trim = 0.1)
[1] 70.4
> mean(trim = 0.1, temp)
[1] 70.4
# mix named and unnamed arguments
# named arguments are assigned first, then
unnamed arguments are matched by position
```

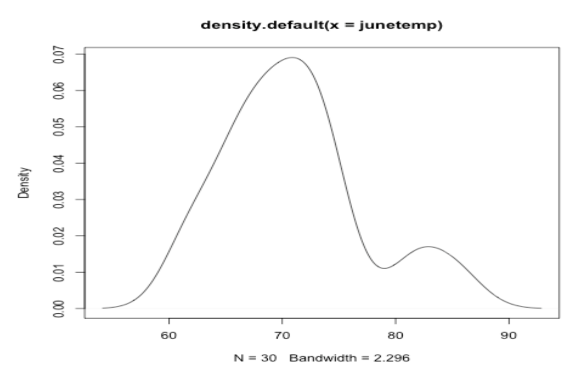
Compound functions

We can take the return value from one function and pass it as an input to another function.

```
> dens = density(junetemp)
> plot(dens)
```
OR, equivalently

```
> plot(density(junetemp))
```
Note: be careful not to use function names for your variables. It can confuse you (and R)

plot(density(junetemp))



Example plotting function

Arguments to plot()

- main: title for the plot
- xlab: x axis label
- xlim: upper and lower bound for x axis
- col: color for plotting symbol
- ylab, ylim

We will cover these in greater detail tomorrow

Subsetting

Subgroups

- Suppose we want to compare the BMI of the men and women in our family
- Create a logical expression that identifies the women in the family

```
> family$gender == "f"
 [1] FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE
[8] TRUE FALSE FALSE TRUE FALSE FALSE  TRUE
```

- Use this logical expression to subset the weights

```
> subset(family$weight, family$gender == "f"]
[1] 125 105 190 124 166 125
```

Comparing subgroups

- Suppose we want to compare the men and women
- Use this logical expression to subset the vector of fweight
- Now we have two data frames, one for each subgroup:

```
> females = subset(family, gender == "f")
> males = subset(family, gender != "f")
```

tapply()

This function is useful to apply a function to subgroups

```
> tapply(family$weight, family$gender, mean)
 f        m
139.1667 172.6250
```

Subsetting with []

- BMI of the 10th person in the family

> family$bmi[10]

Subset by position

[1] 30.04911

- Ages of all but the first person in the family

> family$age[-1]

[1] 33 79 47 27 33 67 52 59 27 55 24 46 48

Subset by exclusion

Suppose we want:

- Genders of the family members who are over weight

Subset by logical

```
> family$gender[family$overWt]
[1] m f m m m f
```

- Heights of female family members

```
> family$height[family$gender ==
"f"]
[1] 64 64 68 65 67 62
```

Subsetting a data frame with [ ]

```
> family[family$weight > 180,]
```

We subset the rows using a **logical** vector

Statistician's perspective:
The method of comparison is a key concept in statistics

R's graphics model

- There are two models in R – painter and object-oriented
- We will use the painter's model
- The other is easy to get started but hard to tweak
- Painter's model – start with a blank canvas, add/paint on it in multiple passes

A Few R Plotting Functions

- `hist()` histogram
- `boxplot()` boxplot
- `dotchart()` dotchart
- `stripchart()`
- `plot()` for scatter plots, line plots, density plots
- `smoothScatter()`
- `barchart()`
- `pie()`
- `mosaicplot()`
- `map('county','Colorado')`
- `abline()` add line to canvas
- `points()` add points to canvas
- `lines()` add line segments to canvas
- `text()` add text to canvas
- `legend()` add legend
- `jitter()` add noise to points

A Few Plot Arguments

`?plot.default`

- `type = "l"` "p" for points, "l" for lines, "n" for nothing
- `ylim = c(0, 1)` the range for the scale of the axis; xlim for x-axis
- `xlab = "x axis label" xlab for x-axis`
- `main = "plot title"`
- `col` = vector of colors for each point
- `log = "y"` use log scale on y axis, can be "x" or "xy"
- `lwd = 2` thickness of line
- `pch = 19` plotting character
- `cex = 0.5` character magnification
- `lty = 2` type of line – check other numbers
- `las = 1` 0,1,2, or 3 style of tick mark labels

Reading data into R

- In the introductory class, all data was given to the students in R format, i.e. in an .Rda file
- They simply load the data into R with

```
> load("BRFSS.rda")
> load(url("
http://www...data/BRFSS.rda"))
```

Resources

- R videos for the introductory course:

http://www.stat.berkeley.edu/share/rvideos/R_Videos/R_Videos.html

- Mosaic project: http://mosaic-web.org/
- “Using R for Data Analysis and Graphics - Introduction, Examples and Commentary” by John Maindonald http://cran.r-project.org/doc/contrib/usingR.pdf

With these skills what can a student do?

Training Statisticians

Reasons:

- Good computing skills are essential to good data analysis
- Computing provides insight and understanding for statistical concepts in a constructive and tangible manner
- Students need to express ideas through computation with the same facility as math

Topics

- Problems with data (real, large, problem driven)
- EDA in modern era ***with*** computing
- Programming concepts (using R)
- Data technologies - regular expressions, databases, XML
- Computer intensive statistical methods
- Simulation studies

Goals

- Basic computing vocabulary & skills
- Express computational tasks in programming language
  - Correctly
  - Efficiently (in terms of the student's time)
- Reason about different approaches to computational tasks
- Learn how to learn about new technologies

Today's Workplace

http://www.youtube.com/watch?v=pi472Mi3VLw&feature=mfu_in_order&list=UL

Statistical Skills

- Complementary Scarce Factor: Ability to understand data and extract value from it
- Skills needed:
  - Access Data, Process Data,
  - Extract Value from Data,
  - Visualize and Communicate
- Managers need data skills
- Information access empowers knowledge workers to work more effectively

Preparation for work/research

- Our students need the essential skills to engage in collaborative research and problem solving
- Our students must have the confidence to overcome computational challenges to carry out a comprehensive data analysis
- Our students should be ready to engage in and succeed at statistical inquiry