Lecture 5b: Computer Problem Solving with R

Irucka Embry, EIT

Fall 2015
Freshman Engineering Seminar ENGR 1020

Tennessee State University College of Engineering
Department of Civil & Architectural Engineering
Outline

- Data Import and Export with R
- Variable Indexing with R
- R Script Files
- R Function Files
- FE Reference Handbook
- Works Cited
Data Import (.csv)

- csv (comma-separated delimiter) file
- MillCreekcsv <- read.table("03431000_MillCreek_AntiochTN_revised.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)
Data Import (.xlsx)

- `library(openxlsx)`
- `MillCreekxlsx <- read.xlsx(xlsxFile = "03431000_MillCreek_AntiochTN_revised.xlsx", detectDates = TRUE)`
```
library(psych)

# Excerpt of MillCreekcsv (rows 112 - 125) found in 03431000_MillCreek_AntiochTN_revised.csv

<table>
<thead>
<tr>
<th>agency_cd</th>
<th>site_no</th>
<th>datetime</th>
<th>o2_00060_0003</th>
<th>o2_00060_0003_cd</th>
</tr>
</thead>
<tbody>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-20</td>
<td>2670</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-21</td>
<td>1120</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-22</td>
<td>1300</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-23</td>
<td>360</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-24</td>
<td>225</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-25</td>
<td>156</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-26</td>
<td>127</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-27</td>
<td>183</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-28</td>
<td>98</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-01-29</td>
<td>78</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-02-01</td>
<td>66</td>
<td>A</td>
</tr>
<tr>
<td>USGS</td>
<td>03431000</td>
<td>1954-02-02</td>
<td>59</td>
<td>A</td>
</tr>
</tbody>
</table>
```
Data Import (clipboard) 2

- `MillCreekclip <- read.clipboard.csv() # copy the comma-separated table [03431000_MillCreek_AntiochTN_revised.csv opened in an advanced text editor], including the table header [column names]`
write.csv(MillCreekcsv, file = "03431000_MillCreek_AntiochTN_revisedClassification.csv", row.names = FALSE)
Data Export (.xlsx)

- `write.xlsx(MillCreekxlsx, file = "03431000_MillCreek_AntiochTN_revisedClass.xlsx")`
Variable Indexing 1

- Return to Example 5 from Lecture Notes 5a (next slide)
- \( m \leftarrow c(83.6, 60.2, 72.1, 91.1, 92.9, 65.3, 80.9) \) \# kg
- \( vt \leftarrow c(53.4, 48.5, 50.9, 55.7, 54, 47.7, 51.1) \) \# m/s
- \( g \leftarrow 9.81 \) \# m/s^2
- What if we only wanted to find the cd (drag coefficient) for the following masses: 72.1, 91.1, and 92.9 kg and the corresponding velocities: 50.9, 55.7, 54 m/s?
  - \( cd \leftarrow g \times m / (vt^2) \) \# kg/m (from Lecture 5)
  - \( cd \leftarrow g \times m[3:5] / (vt[3:5]^2) \) \# kg/m (only certain masses)
- 72.1 is 3rd; 91.1 is 4th; and 92.9 is 5th (mass in kg)
- 50.9 is 3rd; 55.7 is 4th; and 54 is 5th (velocity in m/s)
Example 5 (Chapra 20-21)

- Use the following equation and table to determine the drag coefficient

\[ c_d = \frac{mg}{v_t^2} \]

<table>
<thead>
<tr>
<th>m, kg</th>
<th>83.6</th>
<th>60.2</th>
<th>72.1</th>
<th>91.1</th>
<th>92.9</th>
<th>65.3</th>
<th>80.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_t ), m/s</td>
<td>53.4</td>
<td>48.5</td>
<td>50.9</td>
<td>55.7</td>
<td>54</td>
<td>47.7</td>
<td>51.1</td>
</tr>
</tbody>
</table>
Variable Indexing 2

- # What if you wanted to only see the Date and the corresponding Mean Discharge [02_00060_00003 is the column name] (ft^3/s or cfs) for rows 100 to 250?
- View(MillCreekcsv[100:250, 3:4]) # rows 100 to 250, columns 3 to 4 only
R Script Files

• A script file is a series of R commands that are saved to a file.

• A script file is useful for retaining a series of commands that you want to execute on more than one occasion.

• The script file can be executed by typing `source("file name")` in the command window.
Problem statement: Develop a script file to compute the velocity of the free-falling bungee jumper (Example 4 from Lecture 5a).

Solution: In an advanced text editor (ex. Notepad++) or the RStudio file editor, type the following statements:

```r
g <- 9.81; m <- 68.1; t <- 12; cd <- 0.25; v <- sqrt(g * m / cd) * tanh(sqrt(g * cd / m) * t)
```

Save file as scriptdemo.R

Type `source("scriptdemo.R")` in the command window # use the full file path

Type `v` in the command window

Output:

- `v`
  - [1] 50.61748 # m/s
R Function Files

• Function files are R-files that have the word `function` on the first line.

• In contrast to script files, they can accept input arguments and return outputs.

• The function syntax is below:
  
  ```r
  functionname <- function (inputarguments)
  {
    # comments
    R commands
  }
  ```
- `functionname = function's name`
- `inputarguments = function's argument list (comma-delimited values that are passed into the function)`
- `comments = text that provides the user with information regarding the function (this information can also include the software license)`
- The R-file should be saved as `functionname.R`
- The function can then be executed by typing:
  ```r
  source("functionname")
  functionname(inputarguments)
  ```
- in the command window
Script versus Function Files

- Variables within a function are local and are erased after the function is executed.
- Variables in a script file retain their existence after the script is executed.
Example Function 1

volumesphere <- function (r)
{
  # volumesphere: computes the volume of a sphere using a given radius in length units
  #
  # input:
  # "r = radius of sphere which is the integral of the surface area of a sphere" (length units)
  #
  # output
  # v = volume of sphere (volumetric length units)
  #
  #
  volumesphere <- (4 / 3) * pi * r ^ 3
  cat("The sphere has a volume of approximately", paste(round(volumesphere)), "units^3.")
  cat("\n")
  return(volumesphere)
}
Example Function 1 cont

Save as volumesphere.R

Type `source("volumesphere.R")`

`volumesphere(r)` in the command line

Replace `r` with the radius of the sphere

Input: `volumesphere(50)`

Output: The sphere has a volume of approximately 523599 units^3.

[1] 523598.8
pythagorean <- function (a, b) 
{
  # pythagorean: computes the length of the hypotenuse of a right triangle using the Pythagorean theorem assuming that the other 2 sides are known
  #
  # input:
  # a = known side 1 (length units)
  # b = known side 2 (length units)
  #
  # output
  # c = hypotenuse (length units)
  #
  #
  csquared <- a ^ 2 + b ^ 2; # csquared = c ^ 2
  c <- sqrt(csquared) # sqrt(csquared) = sqrt(c ^ 2) = c (the missing hypotenuse)
  cat("The hypotenuse has a length of approximately", paste(round(c)), "units.")
  cat("\n")
  return(c)
}
Example Function 2 cont

Save as pythagorean.R

Type `source("pythagorean.R")`

`pythagorean(a, b)` in the command line

Replace `a` & `b` with lengths of known sides

Input: `pythagorean(3, 4)`

Output: The hypotenuse has a length of approximately 5 units.

[1] 5
Example Function 3

righttri <- function (a, b) {
  # righttri: computes 1) the length of the hypotenuse of a right triangle using the Pythagorean theorem assuming that the other 2 sides are known
  # 2) the area of the right triangle
  # 3) the altitude of the right triangle
  # 4) the angle associated with the side named a
  # 5) the angle associated with the side named b
  
  # input:
  # a = known side 1 (length units)
  # b = known side 2 (length units)
  
  # output
  # c = hypotenuse (length units)
  # triarea = area of the right triangle (length units^2)
  
  # alphaangle = the angle associated with the side named a (degrees)
  # betaangle = the angle associated with the side named b (degrees)
  library(pracma) # pracma is needed for the asec functions (below)
  
  csquared <- a^2 + b^2; # csquared = c^2
  c <- sqrt(csquared) # sqrt(csquared) = sqrt(c^2) = c (the missing hypotenuse)
  triarea <- 0.5 * a * b # length units^2
  altitude <- (a * b) / c # length units
  alphaanglerad <- asec(c / a); # radians
  alphaangle <- alphaanglerad * (180 / pi) # degrees
  betaanglerad <- asec(c / b); # radians
  betaangle <- betaanglerad * (180 / pi) # degrees
  
  # check that the interior angles of the right triangle are equal to 180 degrees
  if (alphaangle + betaangle + 90 == 180) {
    cat("True, the interior angles equal 180 degrees and this is a right triangle.\n")
    cat("The hypotenuse has a length of approximately", paste(round(c)), "units.\n")
  } else {
    cat("False, the interior angles do not equal 180 degrees so check that the triangle is actually a right triangle.\n")
  }
}
Save as righttri.R

Type `source("righttri.R")`
righttri(a, b) in the command line

Replace `a` & `b` with lengths of known sides

Input: righttri(3, 4)

Output:

True, the interior angles equal 180 degrees and this is a right triangle. The hypotenuse has a length of approximately 5 units.

[1] 5
Problem statement: Compute the velocity of the free-falling bungee jumper using a function file.

Solution: type the following in the text editor:

```r
freefallvel <- function (t, m, cd) {

  # freefallvel: bungee velocity with second-order drag
  #
  # input:
  # t = time (s)
  # m = mass (kg)
  # cd = second-order drag coefficient (kg/m)
  #
  # output:
  # v = downward velocity (m/s)
  # g <- 9.81; # acceleration due to gravity (m/s^2)
  v <- sqrt(g * m / cd) * tanh(sqrt(g * cd / m) * t) # m/s
  cat("The downward velocity is approximately", paste(round(v)), "m/s.")
  cat("\n")
  return(v)
}
```

Save as freefallvel.R

Type `source("freefallvel.R")`
`freefallvel(t, m, cd)` in the command line

Replace t, m, and cd with the time, mass, and drag coefficient values, respectively

Input: `freefallvel(12, 68.1, 0.25)`

Output:

The downward velocity is approximately 51 m/s.
[1] 50.61748
Script and Function files from today's lecture can be found online:

Data Import and Export files from today’s lecture can be found online:

- http://www.ecoccs.com/03431000_MillCreek_AntiochTN_revised.xlsx
- http://www.ecoccs.com/03431000_MillCreek_AntiochTN_revised.csv
- http://www.ecoccs.com/03431000_MillCreek_AntiochTN_dv.csv
Reviewing the following pages from the *Fundamentals of Engineering Reference Handbook* (posted online at http://www.ecoccs.com/tsuteach.html#engr1020) will be helpful for the remainder of this semester

- iii – 17,
- 19 – 24,
- 28 – 29,
- 40 – 48,
- 109, and
- 114 – 120
Works Cited

