



# PROGRAMMING WITH R, PART 1

Dartmouth College | Research Computing

# OVERVIEW

- Info & Workshops
- What is R?
- R as a programming language
- The R Console
- R Studio
- Data Management
- Packages to extend R
- Hands-on programming in R!

# INFO & WORKSHOPS (I)

- Data Visualization using R
  - James Adams, Baker-Berry Library, [James.L.Adams@dartmouth.edu](mailto:James.L.Adams@dartmouth.edu)
- Statistical Consulting (R, Stata, SAS)
  - Jianjun Hua from Ed Tech provides consulting support for statistics-related questions. Jianjun can be contacted at 603-646-6552 or by emailing [jianjun.hua@dartmouth.edu](mailto:jianjun.hua@dartmouth.edu)
- R for High Performance Computing, parallel computing, GIS
  - [Research.computing@Dartmouth.edu](mailto:Research.computing@Dartmouth.edu) and <http://rc.dartmouth.edu/>
- R Club
  - Katja Koeppen, Microbiology Department organizes an R Club, [Katja.Koeppen@Dartmouth.edu](mailto:Katja.Koeppen@Dartmouth.edu)
- Programming n' Pizza #4 – Tonight 6:30pm – 8:30pm, Carson 61
  - <http://rc.dartmouth.edu/index.php/programming-n-pizza/>
- Departmental Courses at Dartmouth, Math, Quantitative Social Sciences, etc
  - Math 10, Math 50 <https://math.dartmouth.edu/courses/by-term/> , <http://qss.dartmouth.edu/>
  - Math 10, Online Stats book “Online Statistics Education: A Multimedia Course of Study” (<http://onlinestatbook.com/> ). David M. Lane, Rice University.

# INFO & WORKSHOPS (II)

- RC Workshops, Training and Consulting
  - GIS
  - Python
  - Matlab
  - Database Design for Research
- High-Performance Computing
  - Using the high-performance compute resources at Dartmouth
- <http://rc.dartmouth.edu/>

The screenshot shows the website for RC Information Technology & Consulting at Dartmouth. The header includes the RC logo and navigation links for 'Request an Account', 'Contact Us', and 'Request Help'. Below the header is a navigation bar with links for 'High Performance Computing', 'Services', 'Help', 'Training' (highlighted), 'Partnership', 'News', and 'About Us'. The main content area is titled 'Training' and contains a paragraph about research computing help, a search bar, and a calendar for December 2017. The calendar shows dates from 3 to 31. Below the calendar is a section for 'Upcoming Events' listing several workshops and tutorials with their dates and times. On the right side, there are sections for 'Recent Posts' and 'Keep in Touch'.

rc.dartmouth.edu/index.php/training/

**RC** INFORMATION, TECHNOLOGY & CONSULTING  
Research Computing AT DARTMOUTH

Request an Account Contact Us Request Help

High Performance Computing Services Help **Training** Partnership News About Us

## Training

Research Computing helps you learn how to use research software and systems through our live training sessions and online classes.

Browse the following calendar to learn more about the classes/trainings/workshops already scheduled. Click on the title to obtain more information and to register.

December 2017						
S	M	T	W	T	F	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

**Upcoming Events:**

- Workshop: Introduction to Geographic Information Systems  
1/17/2018 2:00pm
- Programming with R, part 1  
1/23/2018 2:00pm
- Programming with R, part 2  
1/25/2018 2:00pm
- Introduction to Matlab  
2/6/2018 10:00am
- Hands-on Tutorial: Introduction to Database Design and Implementation  
2/6/2018 2:00pm
- Show All

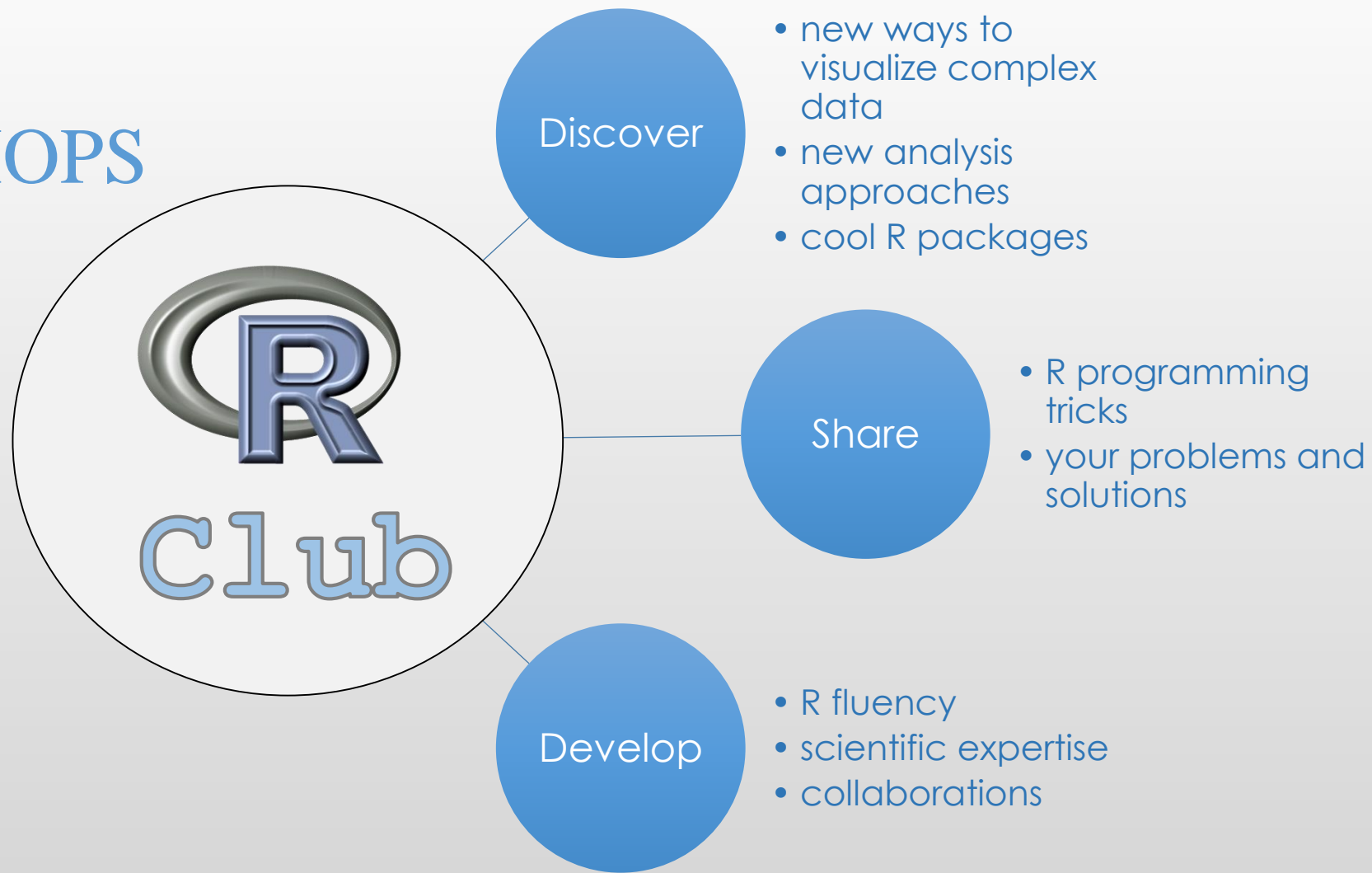
### Recent Posts

- Upcoming Workshops and Training
- Geospatial Community Luncheon
- Programming, Arduinos N' Pizza — November '17
- UPCOMING Programming N'Pizza
- Programming N' Pizza

### Keep in Touch

E-mail us

# INFO & WORKSHOPS (III)



Mondays at Noon in Vail 513

Contact: [Katja.Koeppen@Dartmouth.edu](mailto:Katja.Koeppen@Dartmouth.edu)

# WHAT IS R?

- R is a free software environment used for computing, graphics and statistics. It comes with a robust programming environment that includes tools for data analysis, data visualization, statistics, high-performance computing and geographic analysis. Visit <https://www.r-project.org/> for more
- R has been around for more than 20 years and it has become popular at universities, research labs and federal and state government offices in the last ten years for many applications
- R consists of base packages but also includes hundreds of add-on packages that greatly extend the capabilities of the programming environment.
- These capabilities include data manipulation, data visualization, statistics, geographic data tools, web publishing, workflow

# R AS A PROGRAMMING LANGUAGE (I)

- The R project has a robust programming language that includes basic and advanced programming tools.
- R has the ability to load and manipulate many types of data
- R has programming constructs such as loops and conditionals
- R code can be saved and run at command lines across platforms: PC, Mac and Linux, and it can be optimized to run in High-performance computing environments as well.
- R has packages that allow for connection to databases, advanced visualization, web app design, and more

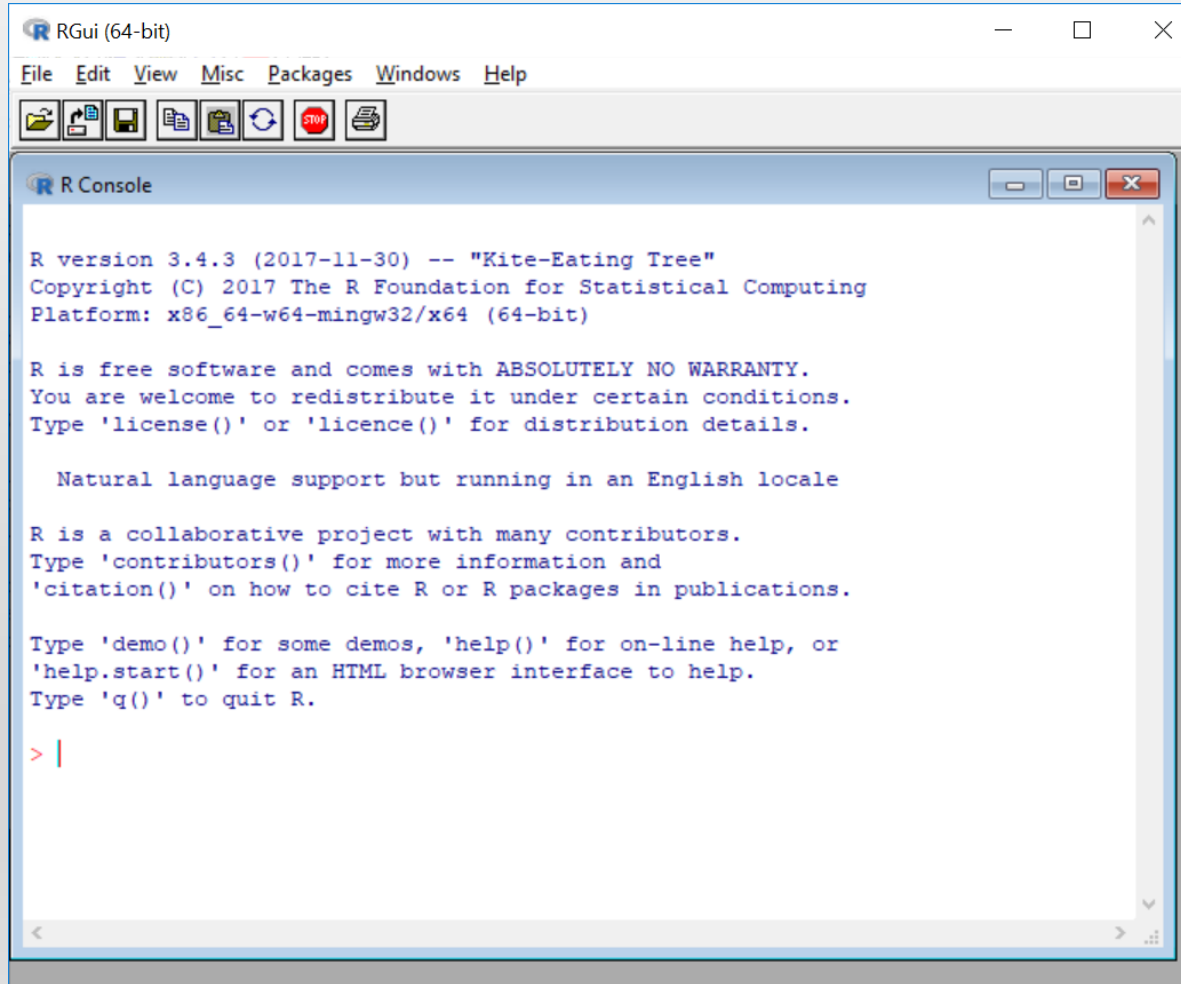
# R AS A PROGRAMMING LANGUAGE (II)

- To work and program effectively with R, we will cover some of the basics:
  - Declaring variables
  - Importing data
  - Running basic functions
  - Creating new functions
  - Programming loops and conditionals
  - Working with Data Frames
  - Extracting data by rows and columns



# THE R CONSOLE

- The R console is a quick, light, multiplatform install

A screenshot of the RGui (64-bit) window. The window has a menu bar with 'File', 'Edit', 'View', 'Misc', 'Packages', 'Windows', and 'Help'. Below the menu bar is a toolbar with icons for file operations. The main area is the 'R Console' window, which displays the R startup message. The text in the console is as follows:

```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

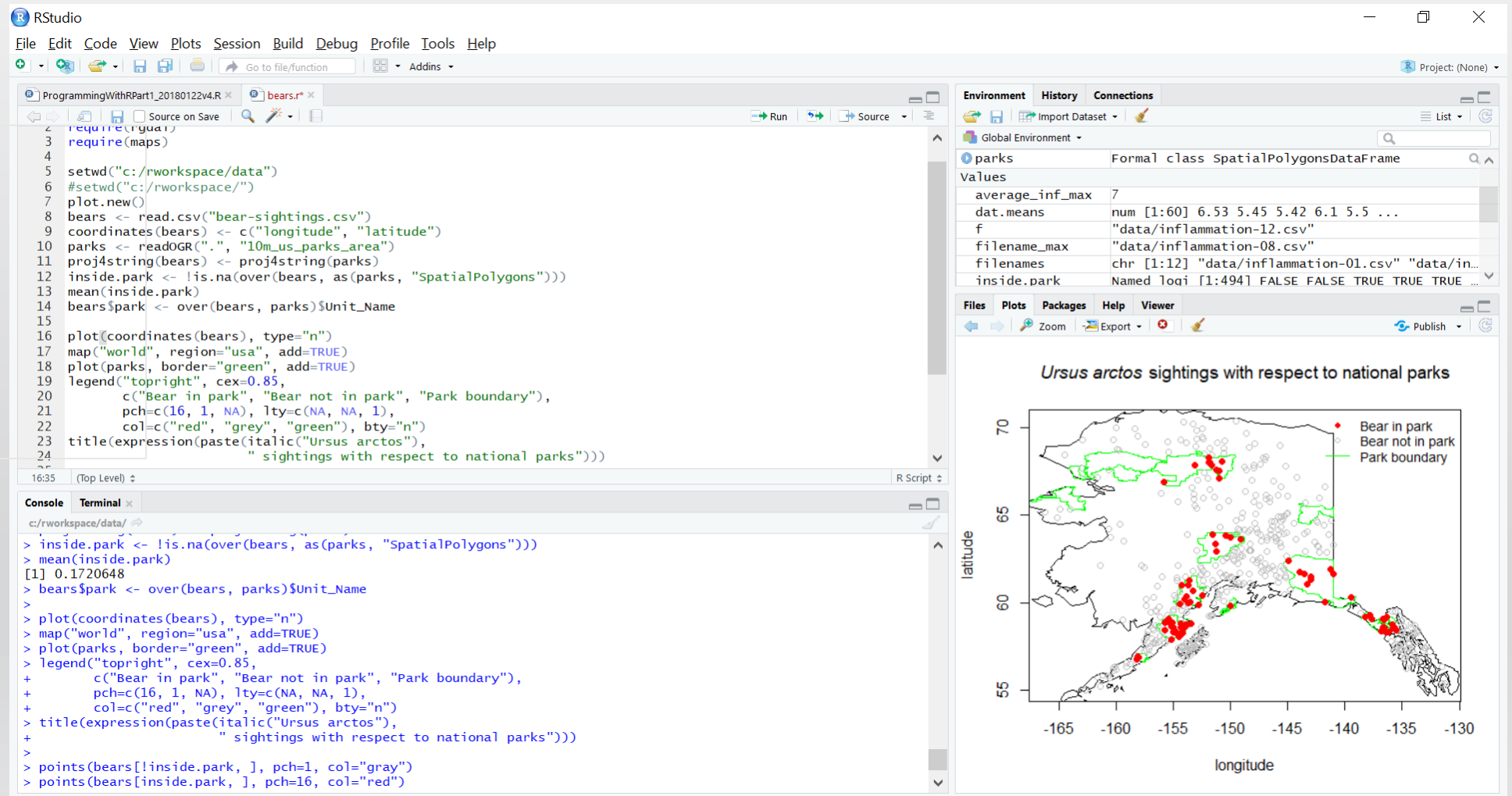
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

# THE R STUDIO IDE

- RStudio is an interactive development environment

- Console
- Terminal
- Script Editor
- Variables
- Plots, Graphics
- Exports
- Package import

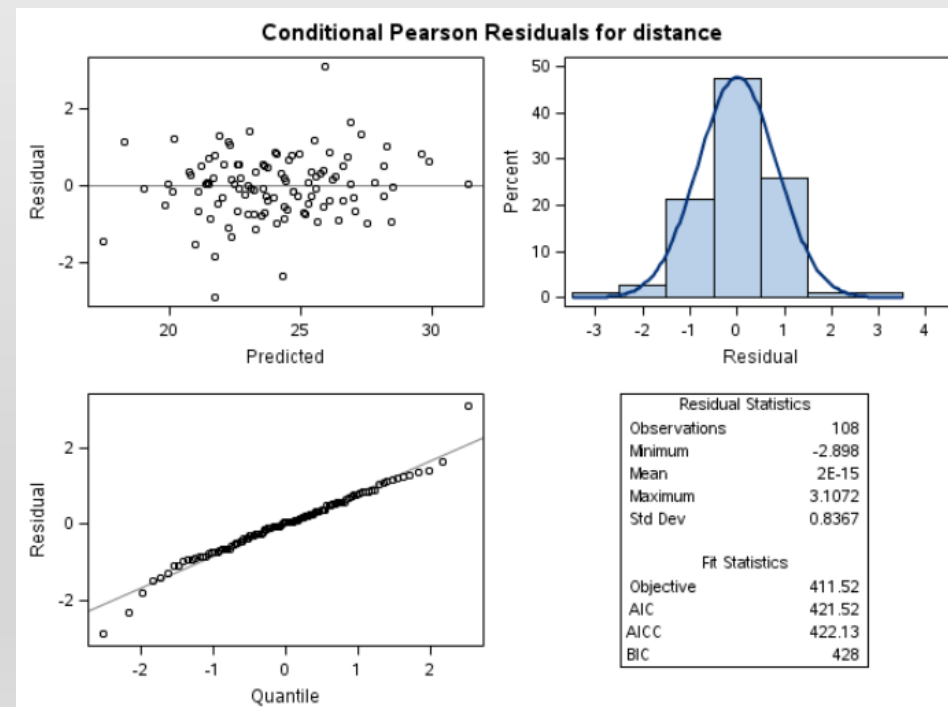
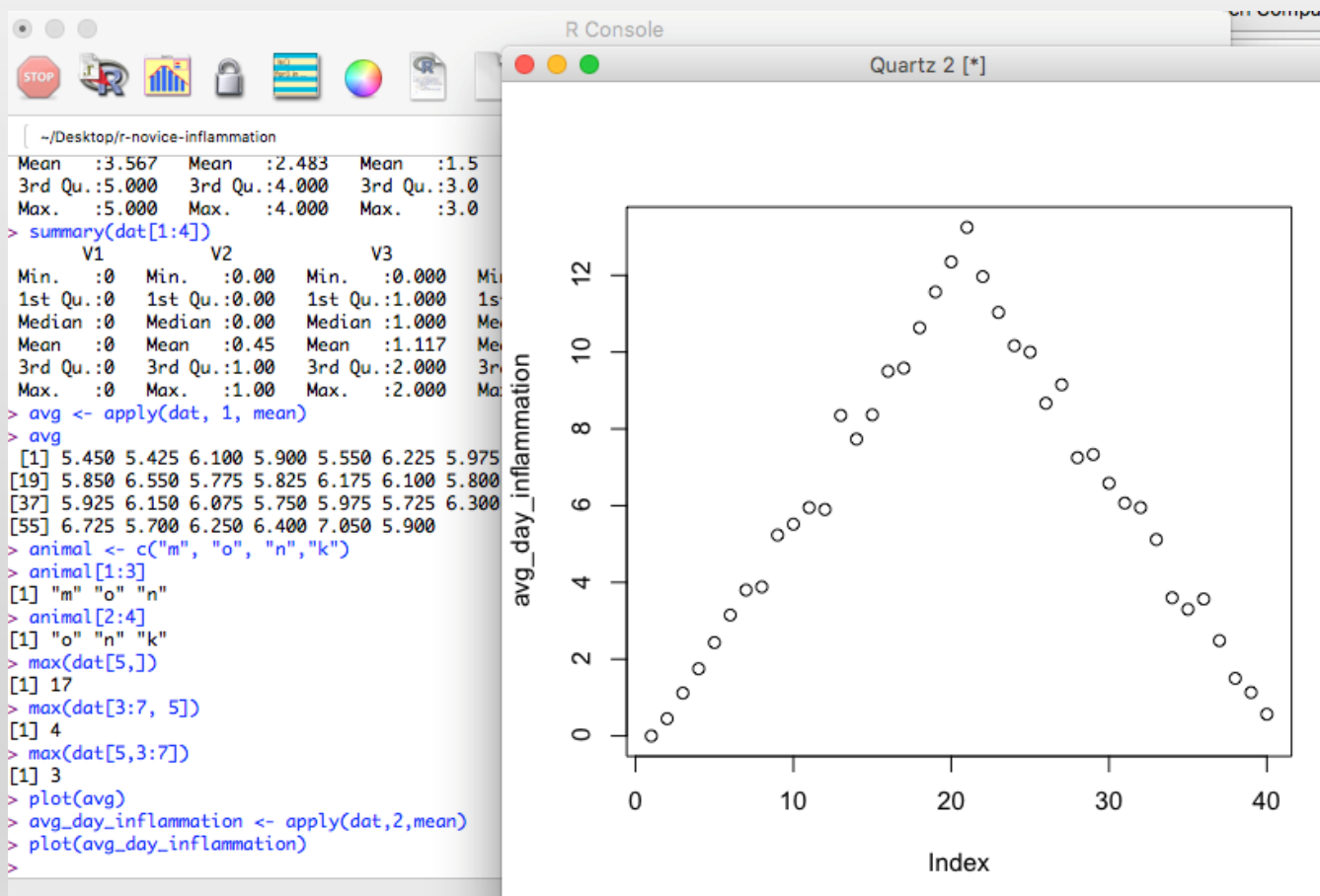


# SOME PACKAGES TO EXTEND R

- <https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages>
- Tidyrr
- Ggplot2
- Dpylr
- xlsx
- Maps
- Sp
- Rgdal
- Parallel

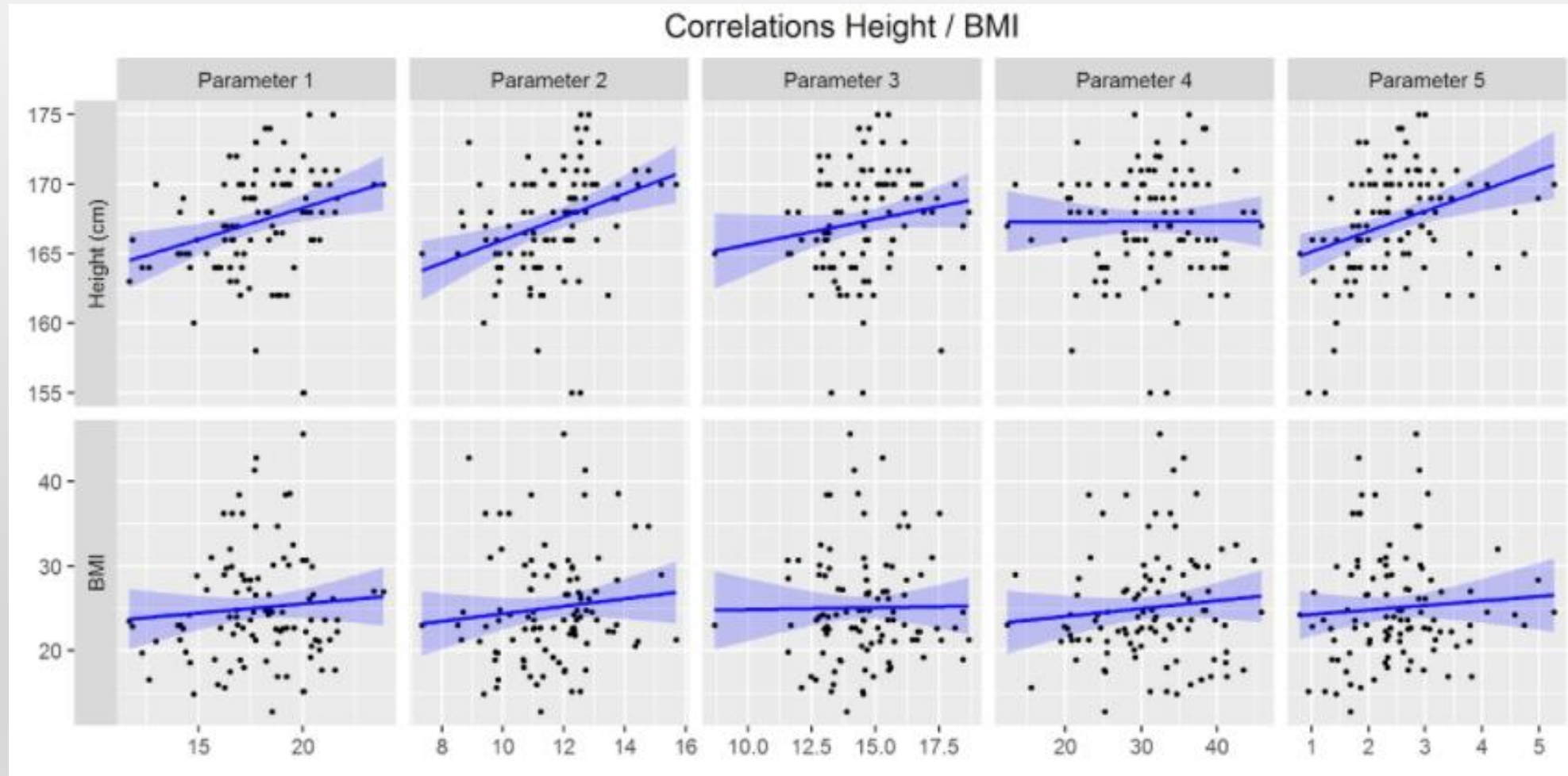
# WHAT CAN R DO?

## SCATTER PLOTS, HISTOGRAMS, RESIDUALS



# WHAT CAN R DO?

## CORRELATION GRAPHS

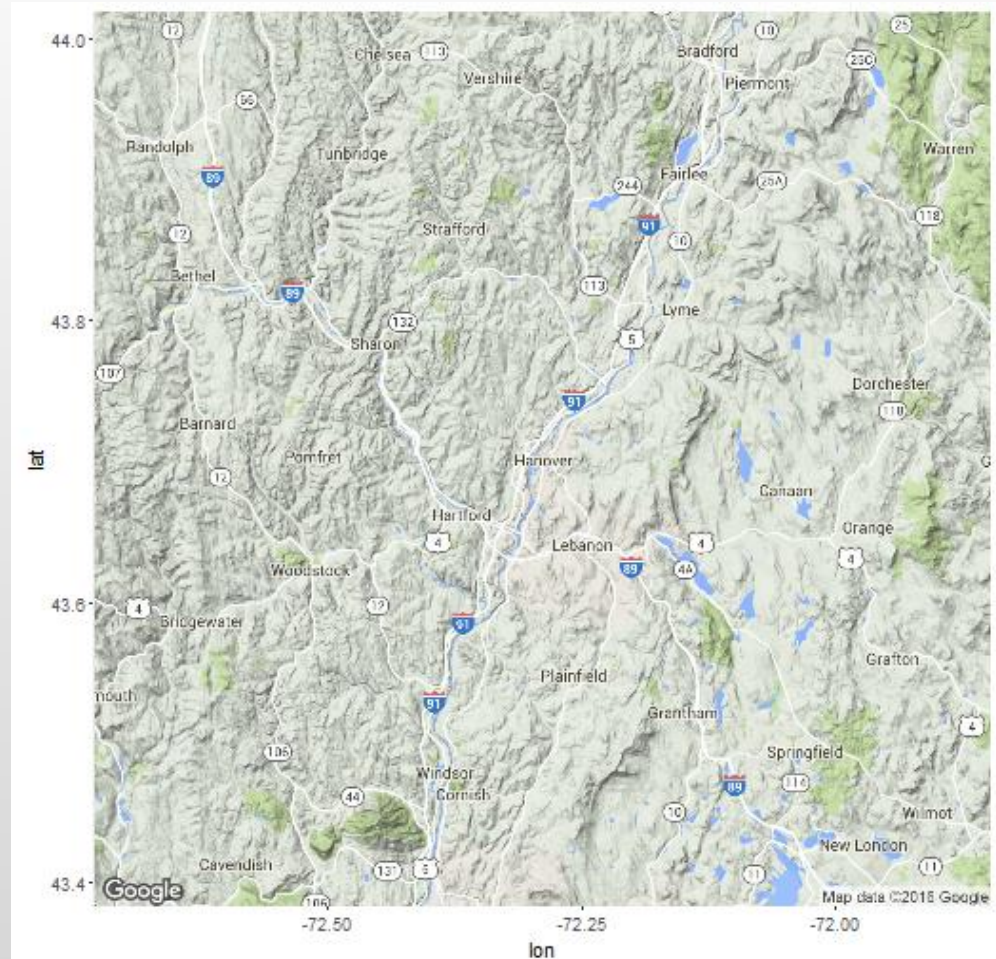


# WHAT CAN R DO?

## MAP EXAMPLE

- Put a Google base map right in your plot window, overlay spatial data on to the map plot

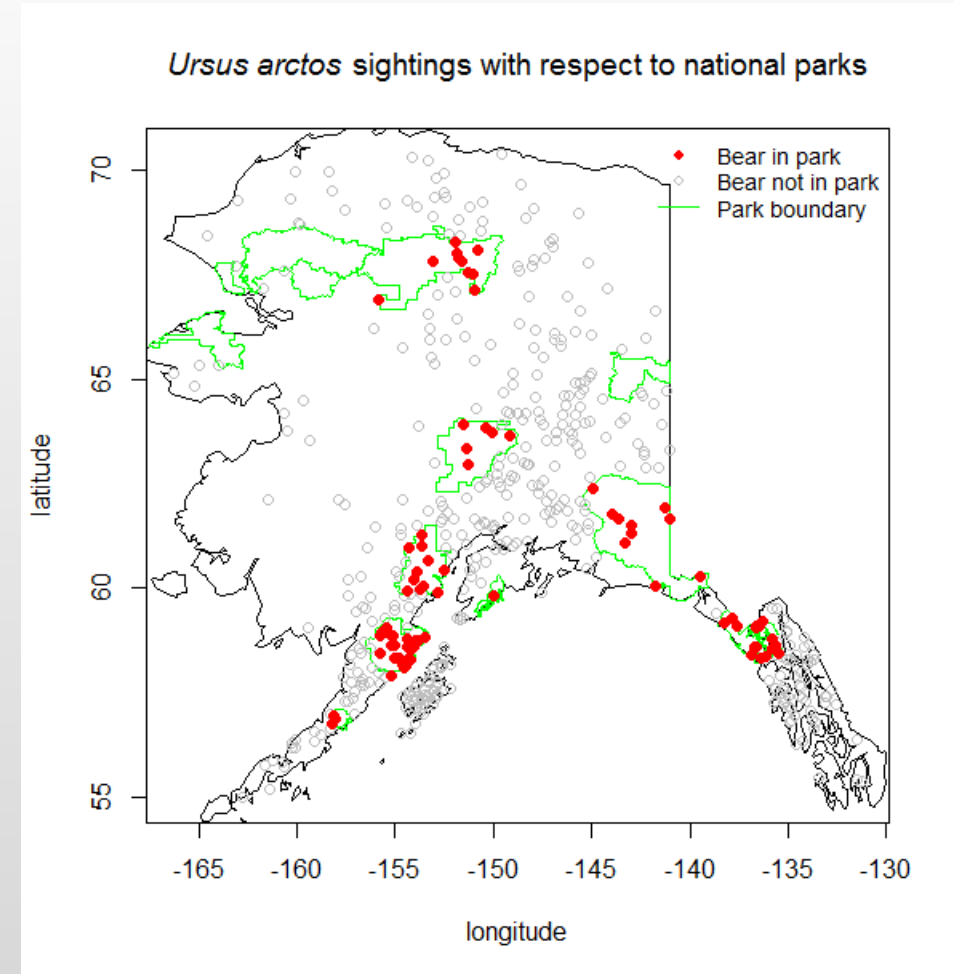
```
install.packages("ggplot2")  
install.packages("ggmap")
```





# WHAT CAN R DO? GEOGRAPHIC INFORMATION ANALYSIS

```
# Map overlay & spatial statistics  
# packages sp, rgdal and maps can turn your R in to a  
# GIS: read, write and analyze spatial data, map overlay  
install.packages("sp")  
install.packages("rgdal")  
install.packages("maps")
```



# READY TO DIVE IN?

- We'll use the **R CONSOLE** today
- Data for this session can be downloaded at:

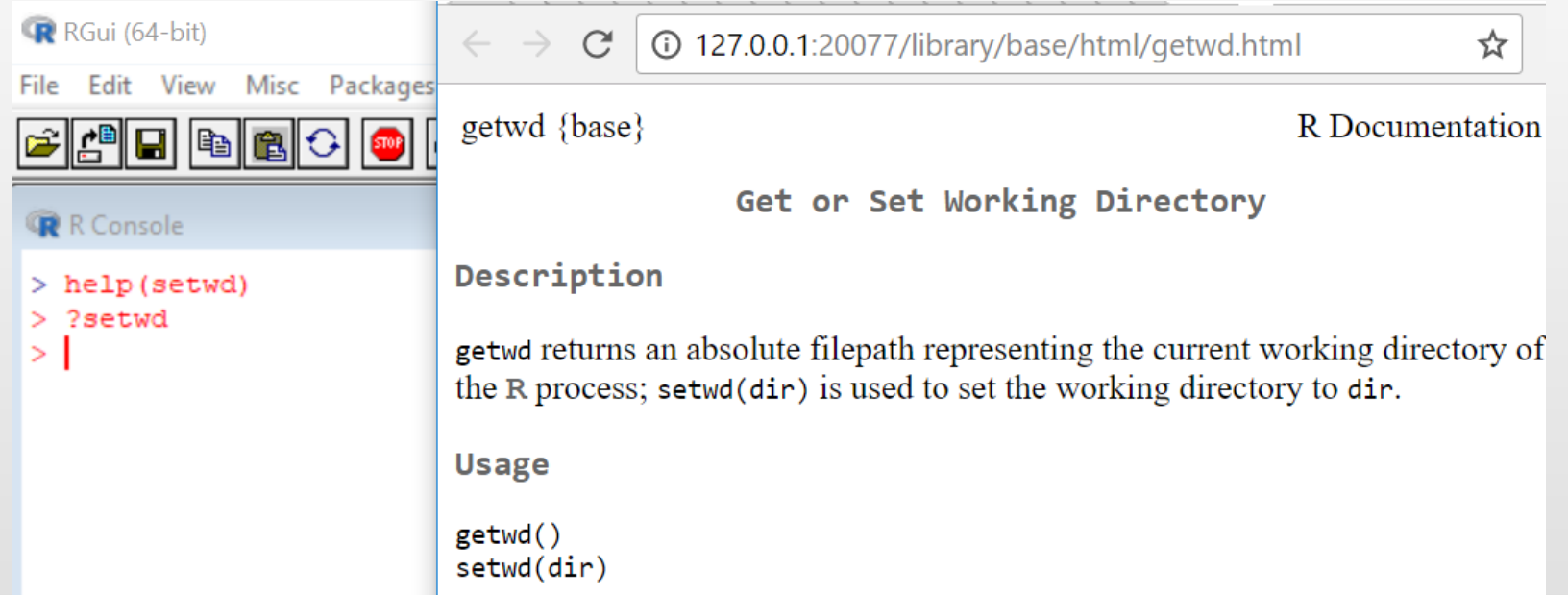
[dartgo.org/programwithr1](http://dartgo.org/programwithr1)

Materials: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.

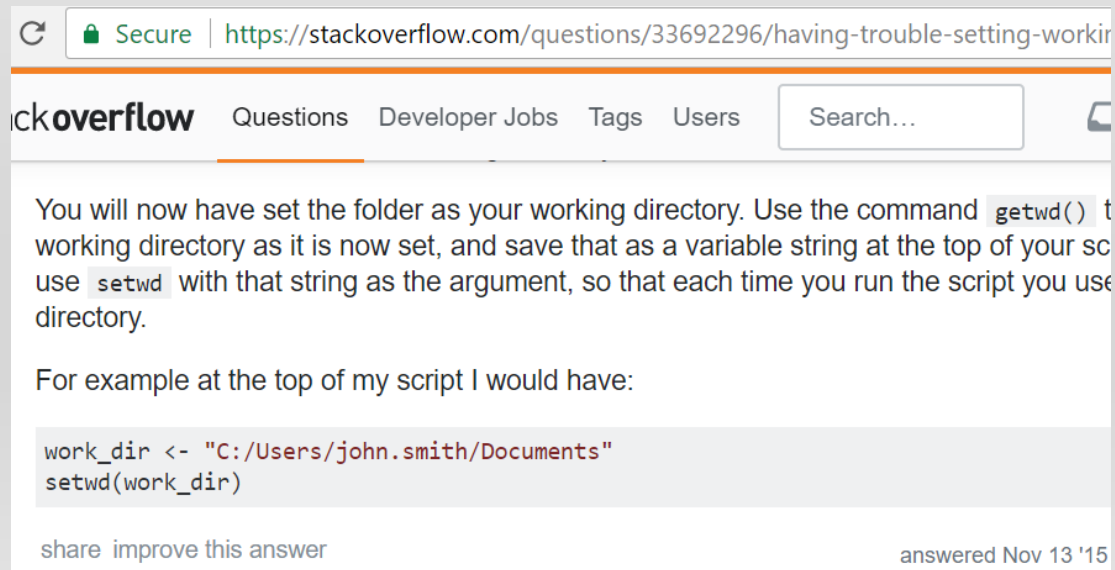


# HELP IN R

- `?setwd`
- `Help(setwd)`
- Web Searches
  - Google 'r set working directory'
  - Stack Overflow 'r set working directory stack overflow'



The screenshot shows the RGui (64-bit) interface. The R Console on the left displays the commands `> help(setwd)`, `> ?setwd`, and a cursor. The web browser on the right shows the R Documentation for `getwd` and `setwd`. The browser address bar shows `127.0.0.1:20077/library/base/html/getwd.html`. The documentation page has a title `getwd {base}` and `R Documentation`. The main heading is `Get or Set Working Directory`. The **Description** section states: `getwd` returns an absolute filepath representing the current working directory of the R process; `setwd(dir)` is used to set the working directory to `dir`. The **Usage** section shows `getwd()` and `setwd(dir)`.



The screenshot shows a Stack Overflow question page. The URL in the address bar is `https://stackoverflow.com/questions/33692296/having-trouble-setting-working-directory`. The page title is `having-trouble-setting-working-directory`. The question text is: "You will now have set the folder as your working directory. Use the command `getwd()` to get the current working directory as it is now set, and save that as a variable string at the top of your script. Then use `setwd` with that string as the argument, so that each time you run the script you use the correct working directory." The answer text is: "For example at the top of my script I would have:" followed by a code block: 

```
work_dir <- "C:/Users/john.smith/Documents"
setwd(work_dir)
```

 The answer is marked as "answered Nov 13 '15".

# R AND THE R CONSOLE

- Install R and the R Console on your Mac/Win PC.
  - R: <https://cran.r-project.org/>
- Material:
  - <http://swcarpentry.github.io/r-novice-inflammation/>
- RStudio
  - RStudio (free): <https://www.rstudio.com/products/rstudio/download/#download>

# DATA MANAGEMENT

- Data Frames
- CSV format (clean csv)
- Tidy Data
- Other formats - Reading out of databases (SQL), Geographic data constructs

# QUESTIONS?



# EXERCISE #1

## Subsetting More Data

Suppose you want to determine the maximum inflammation for patient 5 across days three to seven. To do this you would extract the relevant subset from the data frame and calculate the maximum value. Which of the following lines of R code gives the correct answer?

1. `max(dat[5, ])`
2. `max(dat[3:7, 5])`
3. `max(dat[5, 3:7])`
4. `max(dat[5, 3, 7])`

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R."  
Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.

# EXERCISE #1 SOLN

- Answer: 3

Explanation: You want to extract the part of the dataframe representing data for patient 5 from days three to seven. In this dataframe, patient data is organised in columns and the days are represented by the rows. Subscripting in R follows the [i,j] principle, where i=columns and j=rows. Thus, answer 3 is correct since the patient is represented by the value for i (5) and the days are represented by the values in j, which is a slice spanning day 3 to 7.

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.

# EXERCISE #2

## Using the Apply Function on Patient Data

Challenge: the apply function can be used to summarize datasets and subsets of data across rows and columns using the MARGIN argument. Suppose you want to calculate the mean inflammation for specific days and patients in the patient dataset (i.e. 60 patients across 40 days).

Please use a combination of the apply function and indexing to:

1. calculate the mean inflammation for patients 1 to 5 over the whole 40 days
2. calculate the mean inflammation for days 1 to 10 (across all patients).
3. calculate the mean inflammation for every second day (across all patients).

Think about the number of rows and columns you would expect as the result before each apply call and check your intuition by applying the mean function.

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.

# EXERCISE #2 - SOLN

# 1.

```
apply(dat[1:5, ], 1, mean)
```

# 2.

```
apply(dat[, 1:10], 2, mean)
```

# 3.

```
apply(dat[, seq(1,40, by=2)], 2, mean)
```

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.



# EXERCISE #3

## Create a Function

In the last lesson, we learned to concatenate elements into a vector using the `c` function, e.g. `x <- c("A", "B", "C")` creates a vector `x` with three elements. Furthermore, we can extend that vector again using `c`, e.g. `y <- c(x, "D")` creates a vector `y` with four elements. Write a function called `fence` that takes two vectors as arguments, called `original` and `wrapper`, and returns a new vector that has the wrapper vector at the beginning and end of the original:

```
best_practice <- c("Write", "programs", "for", "people", "not", "computers")
asterisk <- "****" # R interprets a variable with a single value as a vector
                # with one element.
fence(best_practice, asterisk)
```

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.

# EXERCISE #3 - SOLN

```
# write a function to surround or "fence" a variable with another
variable:
# define the vector
best_practice <- c("write", "programs", "for", "people", "not",
"computers")
# define the fence
fence <- "|||"
# define the buildfence function
buildfence <- function(original, wrapper) {
  answer <- c(wrapper, original, wrapper)
  return(answer)
}
# use the buildfence function to build the fence around the vector
buildfence (best_practice, fence)
```

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.

# EXERCISE #4

## Functions to Create Graphs

Write a function called `analyze` that takes a filename as a argument and displays the three graphs produced in the [previous lesson](#) (average, min and max inflammation over time). `analyze("data/inflammation-01.csv")` should produce the graphs already shown, while `analyze("data/inflammation-02.csv")` should produce corresponding graphs for the second data set. Be sure to document your function with comments.

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.

## EXERCISE #4

```
analyze <- function(filename) {  
  # Plots the average, min, and max inflammation over time.  
  # Input is character string of a csv file.  
  dat <- read.csv(file = filename, header = FALSE)  
  avg_day_inflammation <- apply(dat, 2, mean)  
  plot(avg_day_inflammation)  
  max_day_inflammation <- apply(dat, 2, max)  
  plot(max_day_inflammation)  
  min_day_inflammation <- apply(dat, 2, min)  
  plot(min_day_inflammation)  
}
```

Source: John Blischak, Daniel Chen, Harriet Dashnow, and Denis Haine (eds): "Software Carpentry: Programming with R." Version 2016.06, June 2016, <https://github.com/swcarpentry/r-novice-inflammation>, 10.5281/zenodo.57541.