An Introduction to Statistical Computing in R

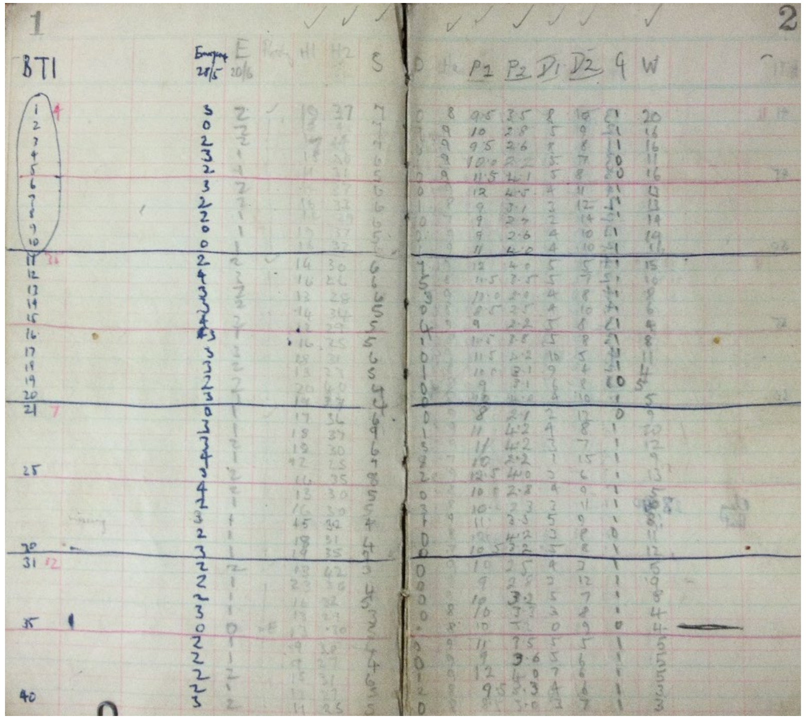
Heather Kropp

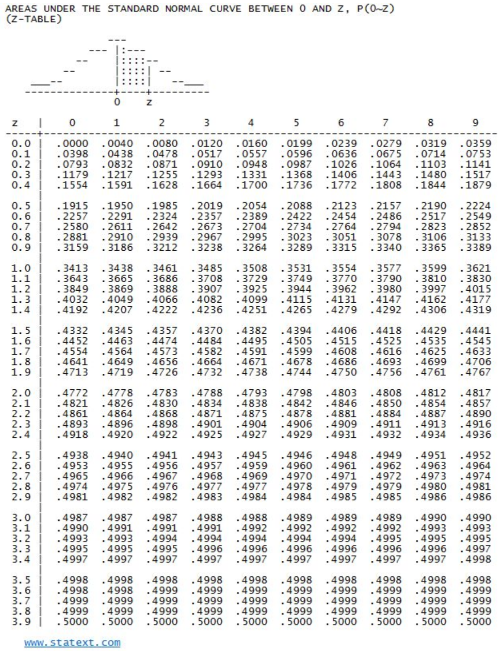
Spring 2019

GEOG 401

Data & Physical Geography

- Environmental data has rapidly changed with evolving technology
  - Satellites, sensors, sUAS, and citizen science produce more data than ever
  - Many datasets becoming "big data"
- Rapidly improving computational power offers:
  - data management & storage
  - more sophisticated statistics




AREAS UNDER THE STANDARD NORMAL CURVE BETWEEN 0 AND Z, P(0~Z)
(Z-TABLE)

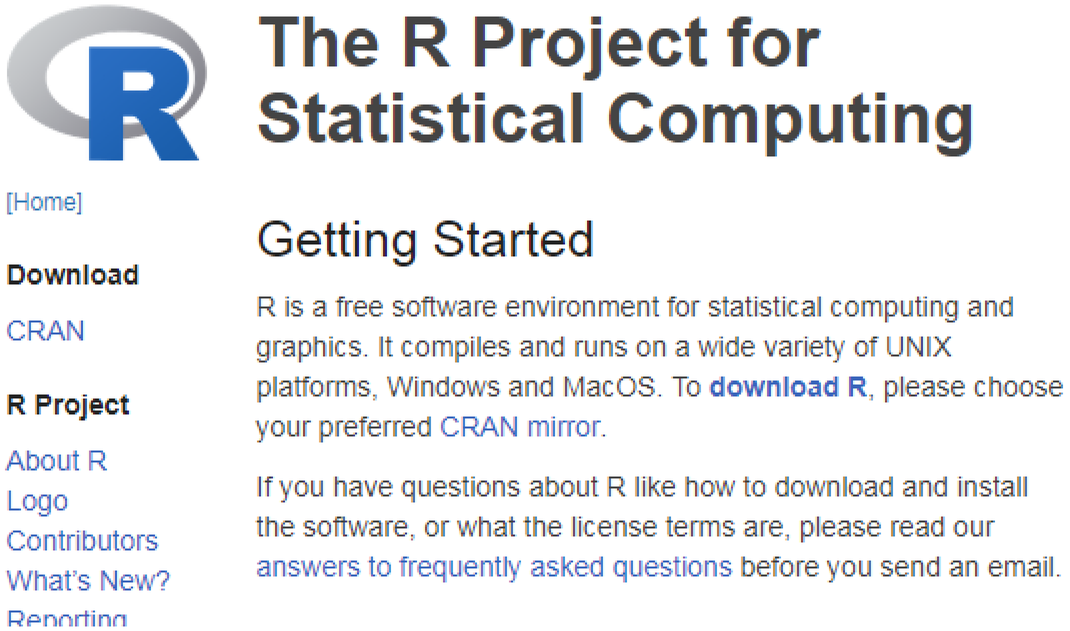
```
             ---
         --- |:---
       --    |::::--
     --      |::::| --
 ___--       |::::|   --___
-------------+----+---------
             0    z
```

z	0	1	2	3	4	5	6	7	8	9
0.0	.0000	.0040	.0080	.0120	.0160	.0199	.0239	.0279	.0319	.0359
0.1	.0398	.0438	.0478	.0517	.0557	.0596	.0636	.0675	.0714	.0753
0.2	.0793	.0832	.0871	.0910	.0948	.0987	.1026	.1064	.1103	.1141
0.3	.1179	.1217	.1255	.1293	.1331	.1368	.1406	.1443	.1480	.1517
0.4	.1554	.1591	.1628	.1664	.1700	.1736	.1772	.1808	.1844	.1879
0.5	.1915	.1950	.1985	.2019	.2054	.2088	.2123	.2157	.2190	.2224
0.6	.2257	.2291	.2324	.2357	.2389	.2422	.2454	.2486	.2517	.2549
0.7	.2580	.2611	.2642	.2673	.2704	.2734	.2764	.2794	.2823	.2852
0.8	.2881	.2910	.2939	.2967	.2995	.3023	.3051	.3078	.3106	.3133
0.9	.3159	.3186	.3212	.3238	.3264	.3289	.3315	.3340	.3365	.3389
1.0	.3413	.3438	.3461	.3485	.3508	.3531	.3554	.3577	.3599	.3621
1.1	.3643	.3665	.3686	.3708	.3729	.3749	.3770	.3790	.3810	.3830
1.2	.3849	.3869	.3888	.3907	.3925	.3944	.3962	.3980	.3997	.4015
1.3	.4032	.4049	.4066	.4082	.4099	.4115	.4131	.4147	.4162	.4177
1.4	.4192	.4207	.4222	.4236	.4251	.4265	.4279	.4292	.4306	.4319
1.5	.4332	.4345	.4357	.4370	.4382	.4394	.4406	.4418	.4429	.4441
1.6	.4452	.4463	.4474	.4484	.4495	.4505	.4515	.4525	.4535	.4545
1.7	.4554	.4564	.4573	.4582	.4591	.4599	.4608	.4616	.4625	.4633
1.8	.4641	.4649	.4656	.4664	.4671	.4678	.4686	.4693	.4699	.4706
1.9	.4713	.4719	.4726	.4732	.4738	.4744	.4750	.4756	.4761	.4767
2.0	.4772	.4778	.4783	.4788	.4793	.4798	.4803	.4808	.4812	.4817
2.1	.4821	.4826	.4830	.4834	.4838	.4842	.4846	.4850	.4854	.4857
2.2	.4861	.4864	.4868	.4871	.4875	.4878	.4881	.4884	.4887	.4890
2.3	.4893	.4896	.4898	.4901	.4904	.4906	.4909	.4911	.4913	.4916
2.4	.4918	.4920	.4922	.4925	.4927	.4929	.4931	.4932	.4934	.4936
2.5	.4938	.4940	.4941	.4943	.4945	.4946	.4948	.4949	.4951	.4952
2.6	.4953	.4955	.4956	.4957	.4959	.4960	.4961	.4962	.4963	.4964
2.7	.4965	.4966	.4967	.4968	.4969	.4970	.4971	.4972	.4973	.4974
2.8	.4974	.4975	.4976	.4977	.4977	.4978	.4979	.4979	.4980	.4981
2.9	.4981	.4982	.4982	.4983	.4984	.4984	.4985	.4985	.4986	.4986
3.0	.4987	.4987	.4987	.4988	.4988	.4989	.4989	.4989	.4990	.4990
3.1	.4990	.4991	.4991	.4991	.4992	.4992	.4992	.4992	.4993	.4993
3.2	.4993	.4993	.4994	.4994	.4994	.4994	.4994	.4995	.4995	.4995
3.3	.4995	.4995	.4995	.4996	.4996	.4996	.4996	.4996	.4996	.4997
3.4	.4997	.4997	.4997	.4997	.4997	.4997	.4997	.4997	.4997	.4998
3.5	.4998	.4998	.4998	.4998	.4998	.4998	.4998	.4998	.4998	.4998
3.6	.4998	.4998	.4999	.4999	.4999	.4999	.4999	.4999	.4999	.4999
3.7	.4999	.4999	.4999	.4999	.4999	.4999	.4999	.4999	.4999	.4999
3.8	.4999	.4999	.4999	.4999	.4999	.4999	.4999	.4999	.4999	.4999
3.9	.5000	.5000	.5000	.5000	.5000	.5000	.5000	.5000	.5000	.5000

www.statext.com

Statistical computing

- Interaction between statistics, numerical analysis, and data manipulation
- R: program for statistical computing



The R Project for Statistical Computing

[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting

Getting Started

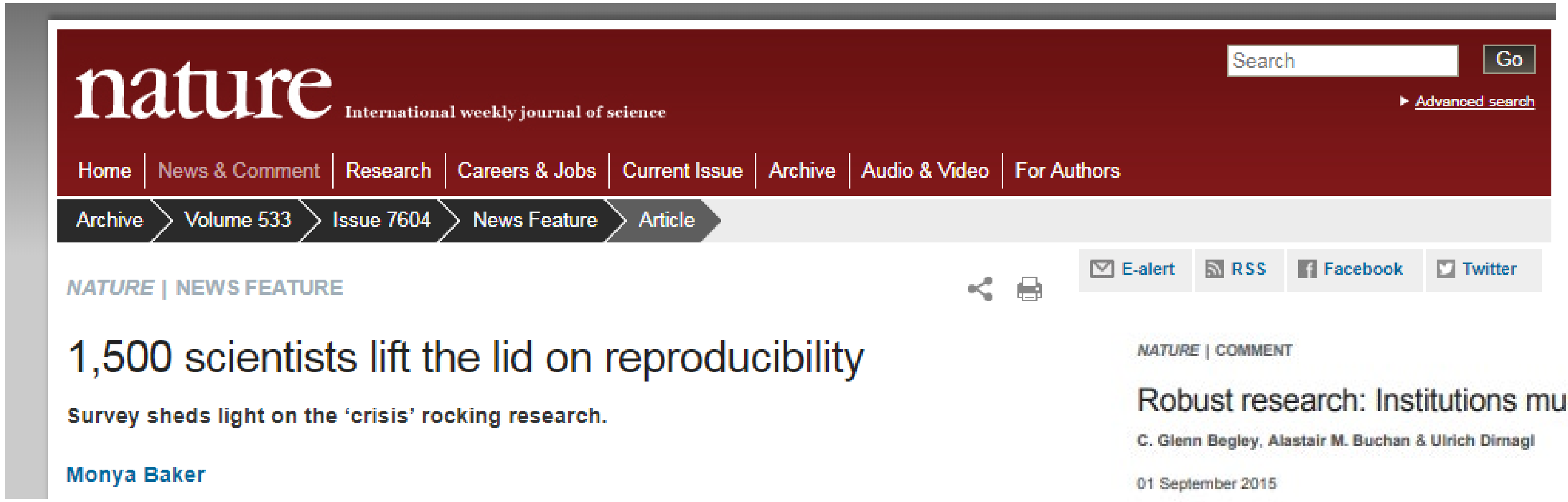
R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

Why use a programming language?

- Powerful resource for handling large or complex data
- Integrates statistics, data manipulation, and GIS in a single framework
- **Reproducibility**

Reproducibility crisis?

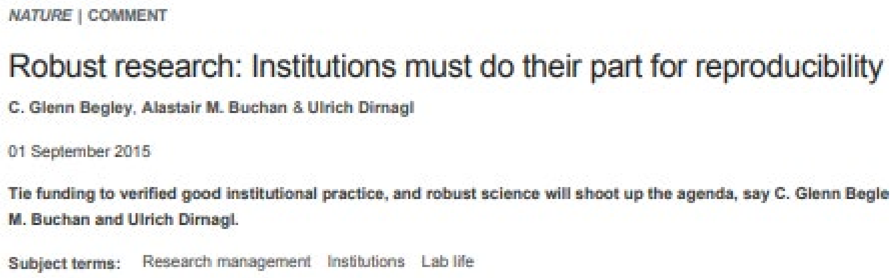


NATURE | NEWS FEATURE

1,500 scientists lift the lid on reproducibility

Survey sheds light on the 'crisis' rocking research.

Monya Baker



NATURE | COMMENT

Robust research: Institutions must do their part for reproducibility

C. Glenn Begley, Alastair M. Buchan & Ulrich Dirnagl

01 September 2015

Tie funding to verified good institutional practice, and robust science will shoot up the agenda, say C. Glenn Begle
M. Buchan and Ulrich Dirnagl.

Subject terms: Research management Institutions Lab life

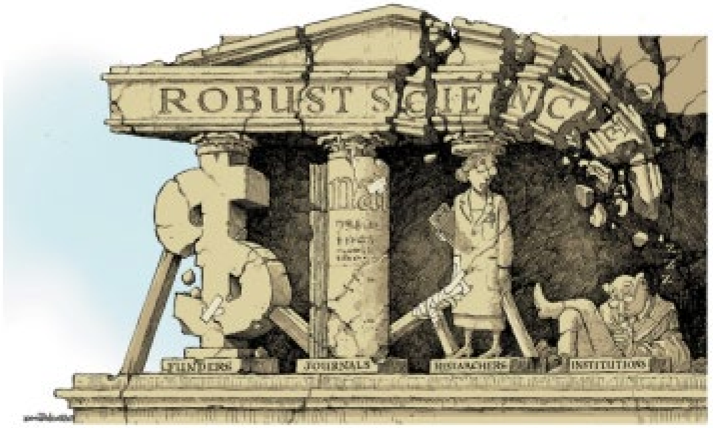


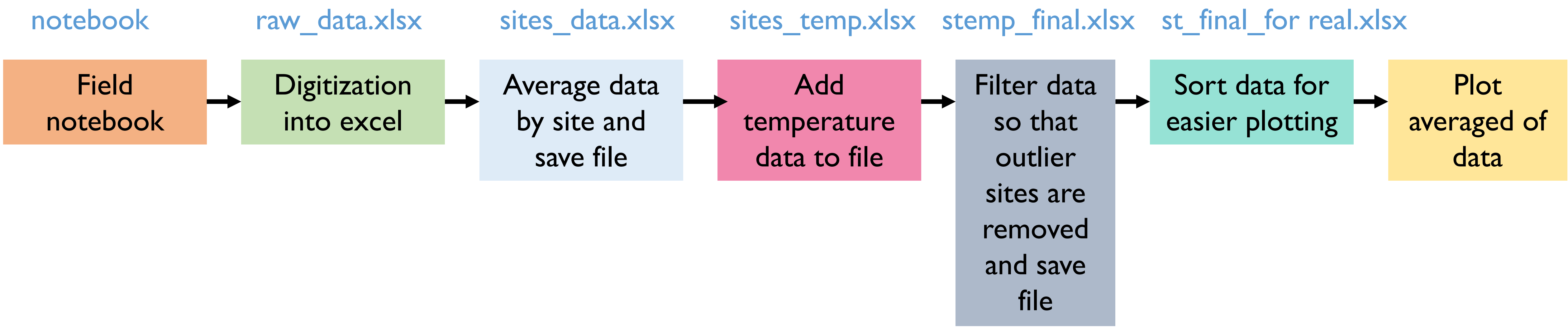
Illustration by David Parkins

Reproducibility issues: two factors

- Statistical uncertainty and limitations
    - Inherent issues with describing the natural world
- Poor data handling and analysis
    - “Fixable”
    - Open data can make analysis transparent
    - Using a programing language allows analyses to be replicated and documented

Data provenance

- The trajectory of data from its collection to final visualization and analysis



notebook → Field notebook

raw_data.xlsx → Digitization into excel

sites_data.xlsx → Average data by site and save file

sites_temp.xlsx → Add temperature data to file

stemp_final.xlsx → Filter data so that outlier sites are removed and save file

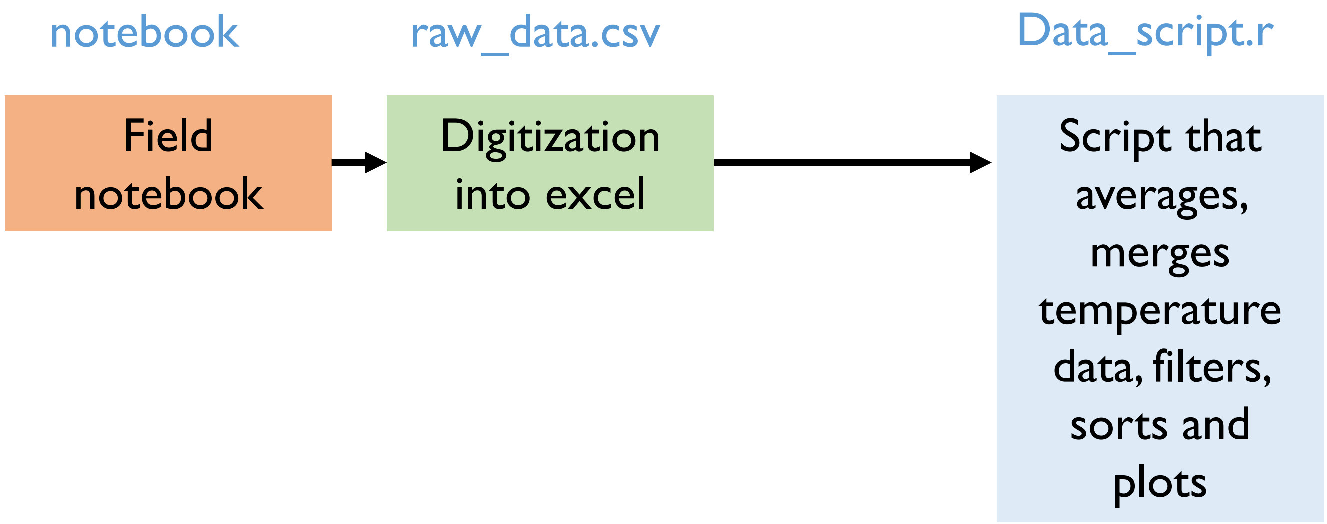
st_final_for real.xlsx → Sort data for easier plotting

Plot averaged of data

- The management of data can readily become unwieldy and easily clouded

Data provenance

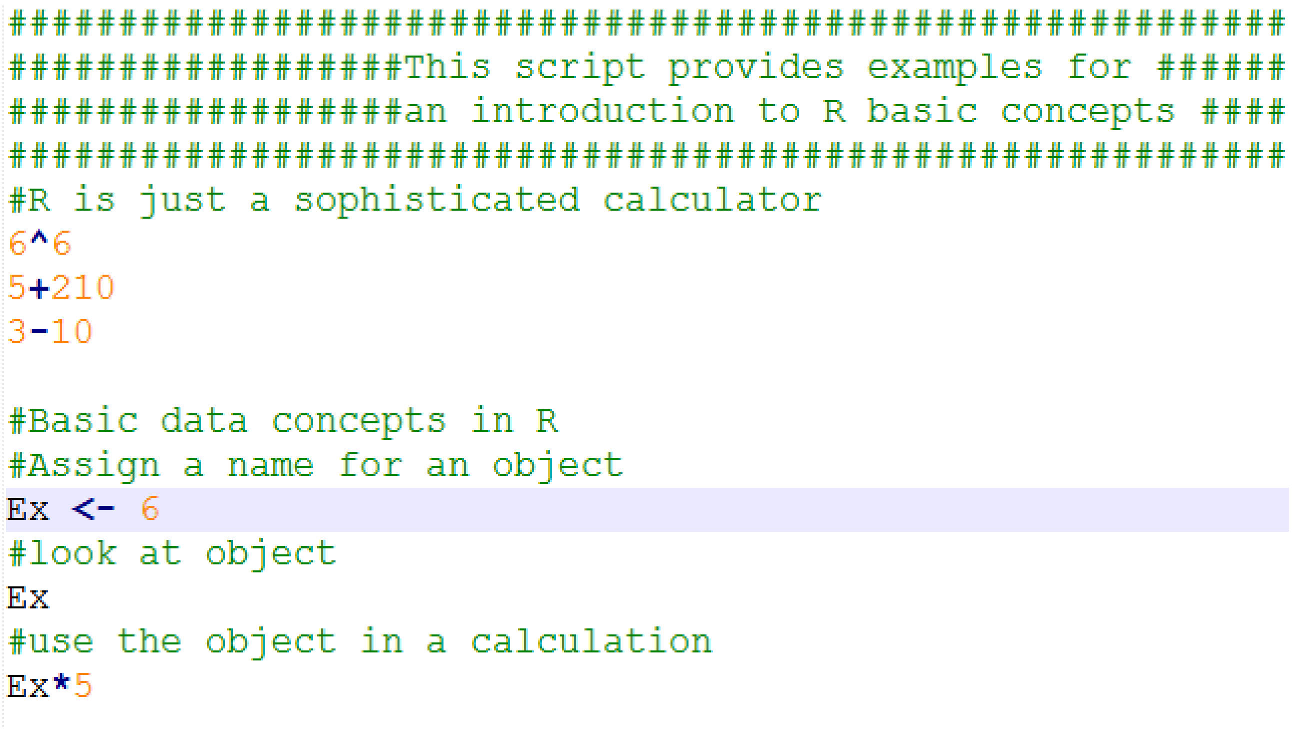
- Using a programming language for data manipulation provides clear documentation and data provenance


notebook
raw_data.csv
Data_script.r
Field notebook
Digitization into excel
Script that averages, merges temperature data, filters, sorts and plots

- It still requires careful management of data files and scripts

Reproducibility and R scripts

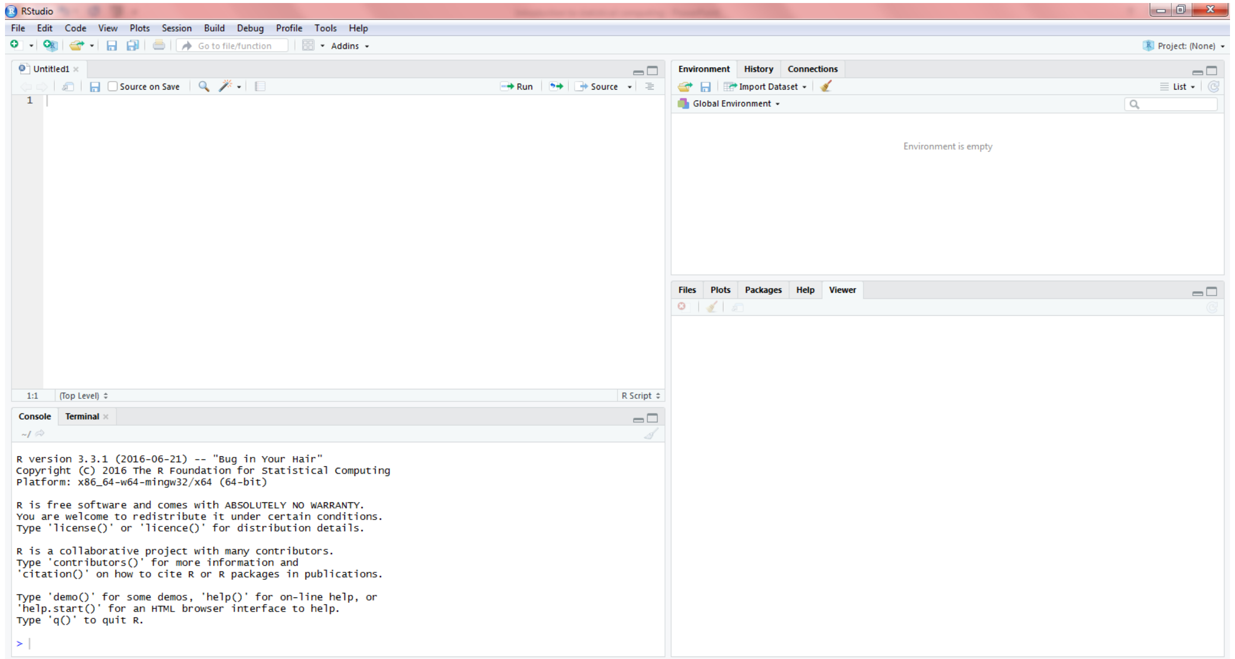
- A script is a file that stores all of the code for a project or operation
- A good script is written so that it can be run and the results will be the same every time
- It is a good idea to keep data workflows simple. Don't generate a lot of scripts for a project.
- Comments in a script are lines the program ignores. These are used for documentation
- COMMENT, COMMENT COMMENT!!!!! (yes, I am yelling at you in all caps)



```
##########################################################
##################This script provides examples for ######
##################an introduction to R basic concepts ####
##########################################################
#R is just a sophisticated calculator
6^6
5+210
3-10

#Basic data concepts in R
#Assign a name for an object
Ex <- 6
#look at object
Ex
#use the object in a calculation
Ex*5
```

Many different ways to interface with R

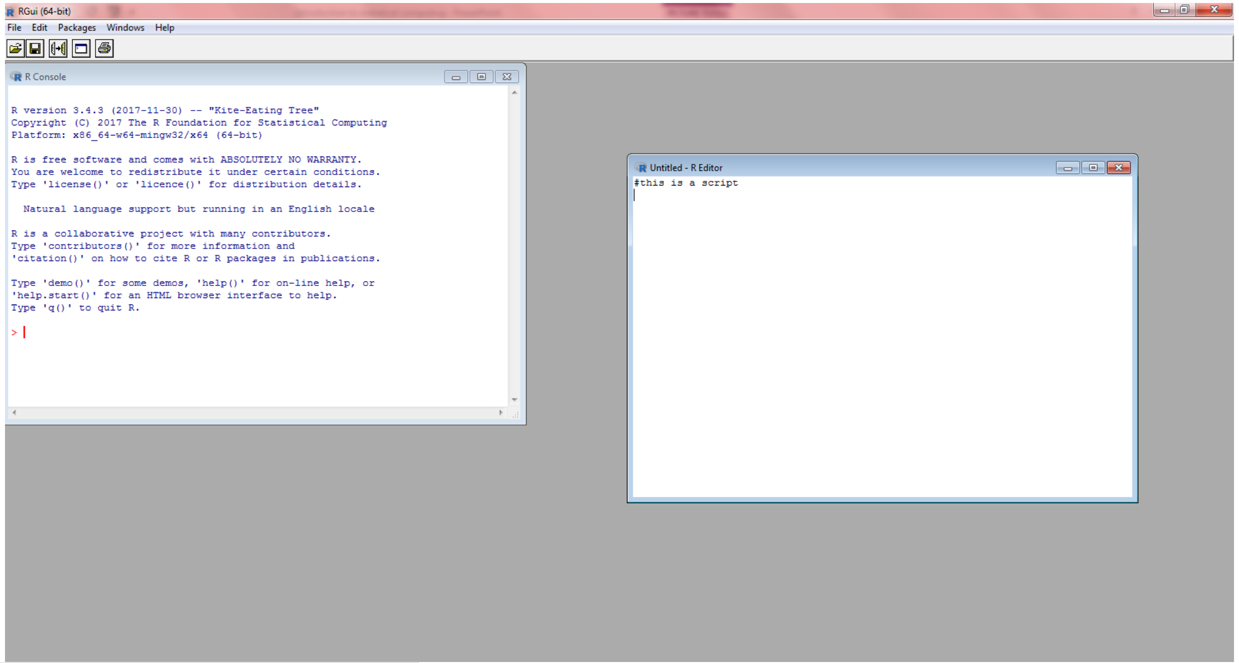
Rstudio



```
1
```

```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 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.

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.

>
```

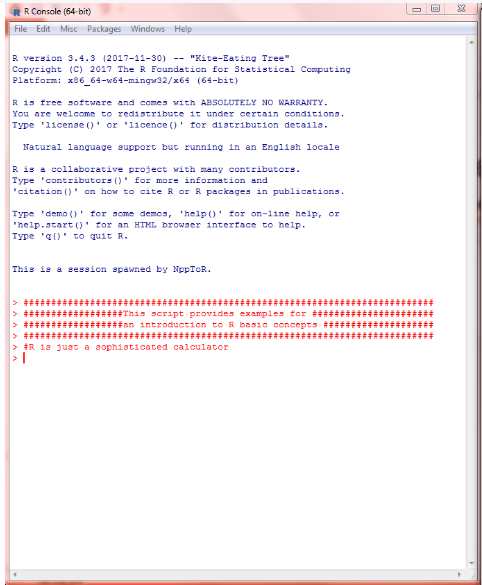
R



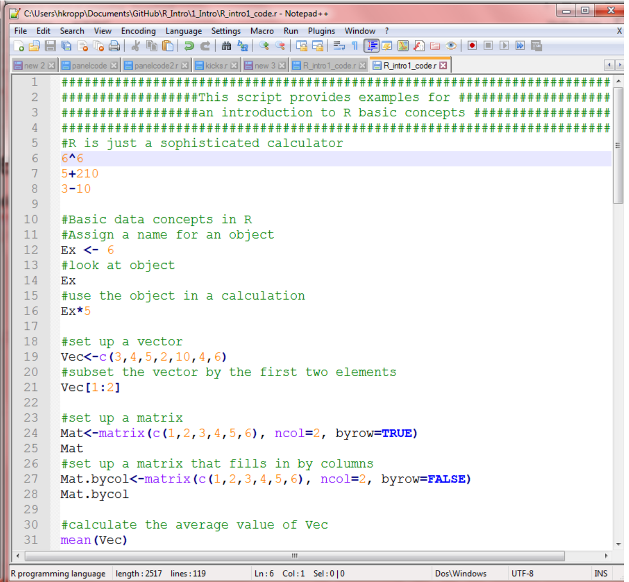
```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

>
```

```
#this is a script
```

Text editors that can send code to R



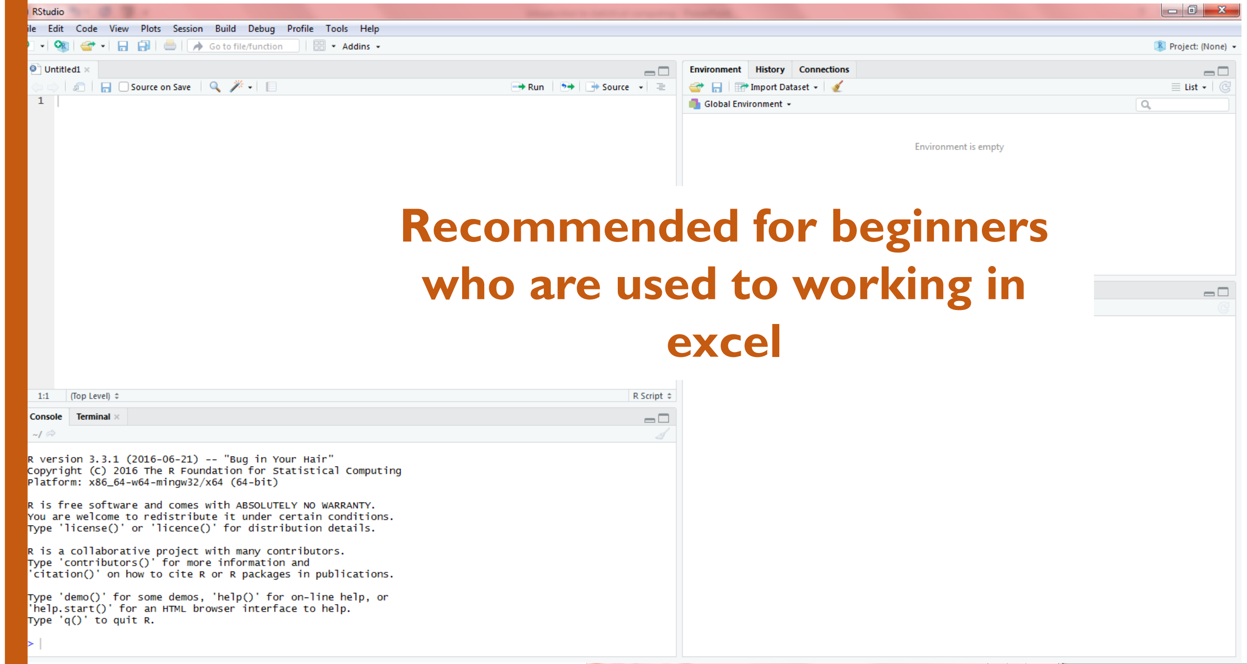
```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.


This is a session spawned by NppToR.


> ##########################################################################
> ##################This script provides examples for ######################
> ##################an introduction to R basic concepts ####################
> ##########################################################################
> #R is just a sophisticated calculator
>
```


C:\Users\hkropp\Documents\GitHub\R_Intro\1_Intro\R_intro1_code.r - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 2
panelcode
panelcode2.r
kicks.r
new 3
R_intro1_code.r
R_intro1_code.r
R programming language
length : 2517 lines : 119
Ln : 6 Col : 1 Sel : 0 | 0
Dos\Windows
UTF-8
INS

```
########################################################################
##################This script provides examples for ####################
##################an introduction to R basic concepts ##################
########################################################################
#R is just a sophisticated calculator
6^6
5+210
3-10

#Basic data concepts in R
#Assign a name for an object
Ex <- 6
#look at object
Ex
#use the object in a calculation
Ex*5

#set up a vector
Vec<-c(3,4,5,2,10,4,6)
#subset the vector by the first two elements
Vec[1:2]

#set up a matrix
Mat<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=TRUE)
Mat
#set up a matrix that fills in by columns
Mat.bycol<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=FALSE)
Mat.bycol

#calculate the average value of Vec
mean(Vec)
```

Many different ways to interface with R

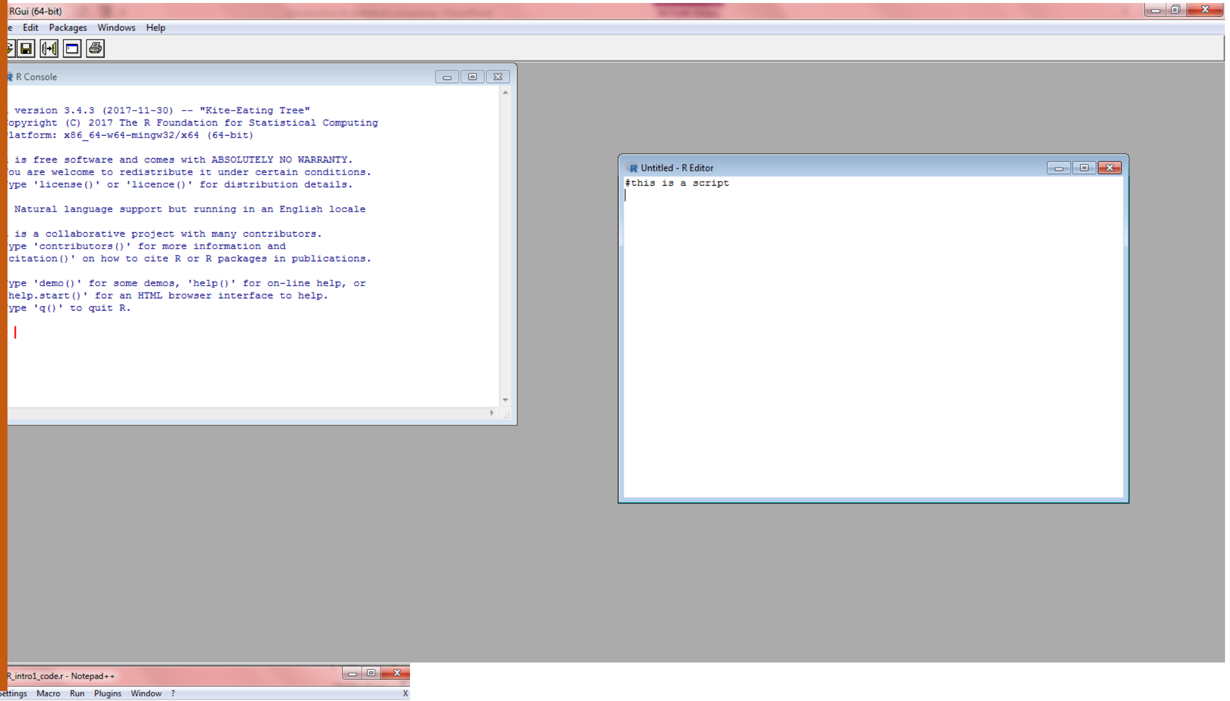
Rstudio



```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 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.

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.

>
```

Recommended for beginners who are used to working in excel

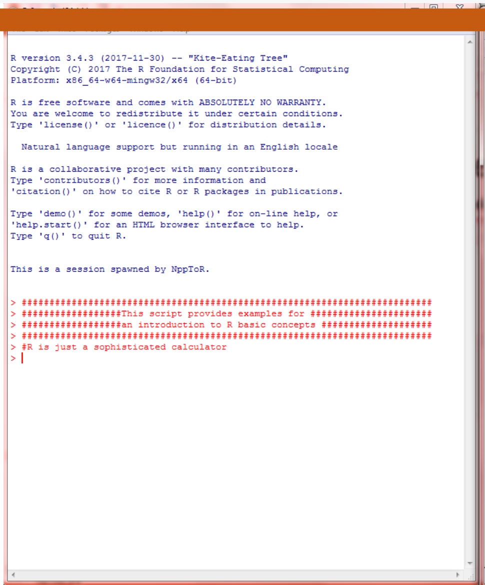
R



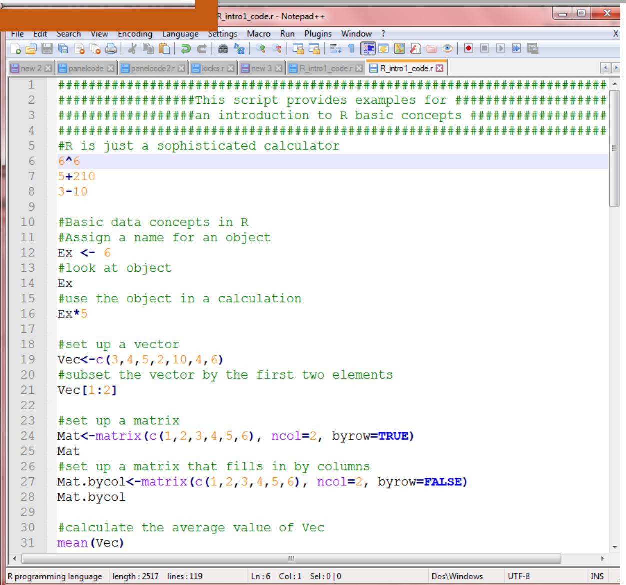
```
 version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
opyright (C) 2017 The R Foundation for Statistical Computing
latform: x86_64-w64-mingw32/x64 (64-bit)

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

 Natural language support but running in an English locale

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

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

```
#this is a script
```

Text editors that can send code to R



```
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.


This is a session spawned by NppToR.


> ##########################################################################
> ##################This script provides examples for #####################
> ##################an introduction to R basic concepts ###################
> ##########################################################################
> #R is just a sophisticated calculator
>
```


R_intro1_code.r - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 2
panelcode
panelcode2.r
kicks.r
new 3
R_intro1_code.r
R_intro1_code.r
R programming language length : 2517 lines : 119 Ln : 6 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 INS

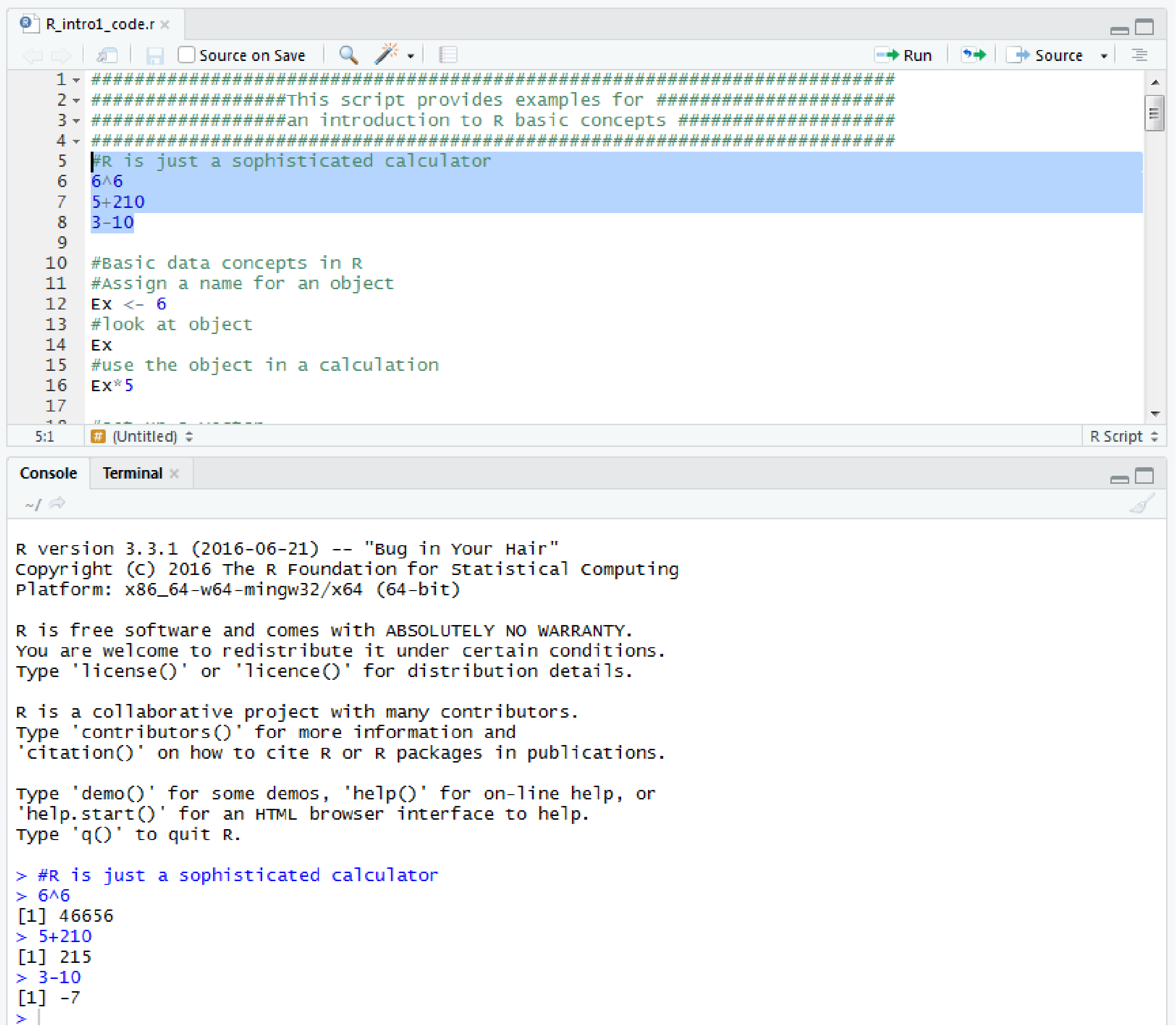
```r
##########################################################################
##################This script provides examples for #####################
##################an introduction to R basic concepts ###################
##########################################################################
#R is just a sophisticated calculator
6^6
5+210
3-10

#Basic data concepts in R
#Assign a name for an object
Ex <- 6
#look at object
Ex
#use the object in a calculation
Ex*5

#set up a vector
Vec<-c(3,4,5,2,10,4,6)
#subset the vector by the first two elements
Vec[1:2]

#set up a matrix
Mat<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=TRUE)
Mat
#set up a matrix that fills in by columns
Mat.bycol<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=FALSE)
Mat.bycol

#calculate the average value of Vec
mean(Vec)
```

The core of R: console and your script

- R console does all of the work. It is basically a giant calculator.
- Your script is where you write code and save it. You need to send it to the console to run it
- The session that you have started in R is called your working environment. It contains everything that you have run in the console.

script

console

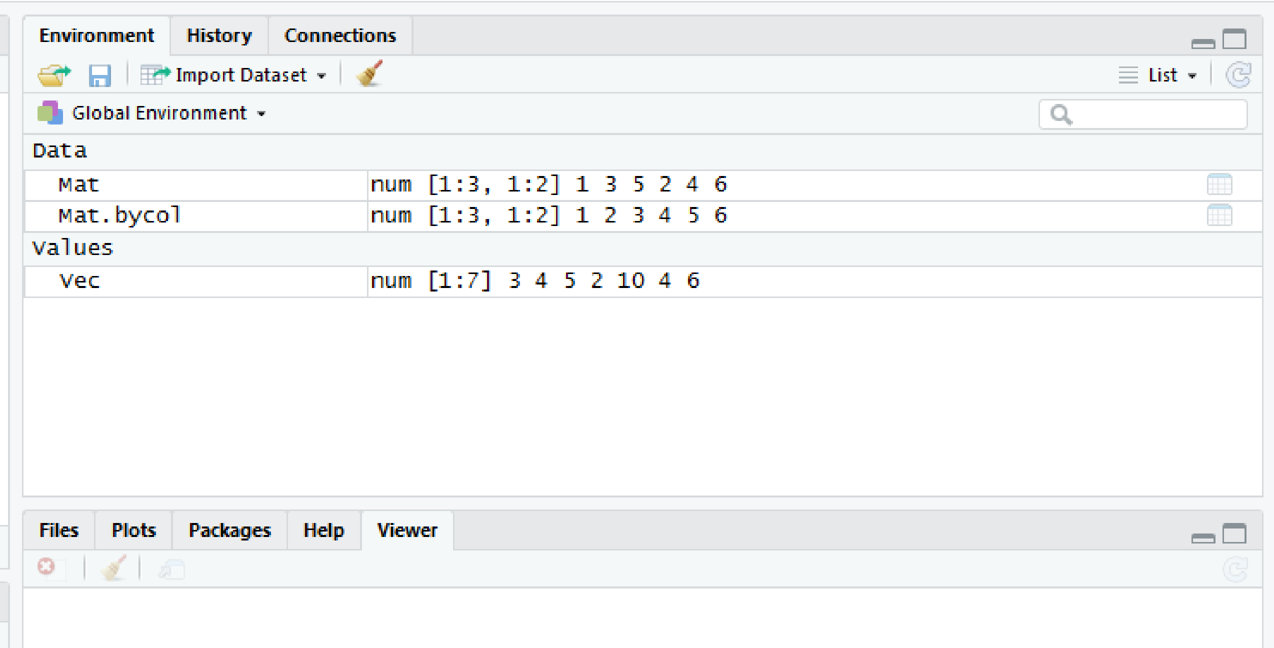


```
##########################################################################
##################This script provides examples for ######################
##################an introduction to R basic concepts ####################
##########################################################################
#R is just a sophisticated calculator
6^6
5+210
3-10

#Basic data concepts in R
#Assign a name for an object
Ex <- 6
#look at object
Ex
#use the object in a calculation
Ex*5

```

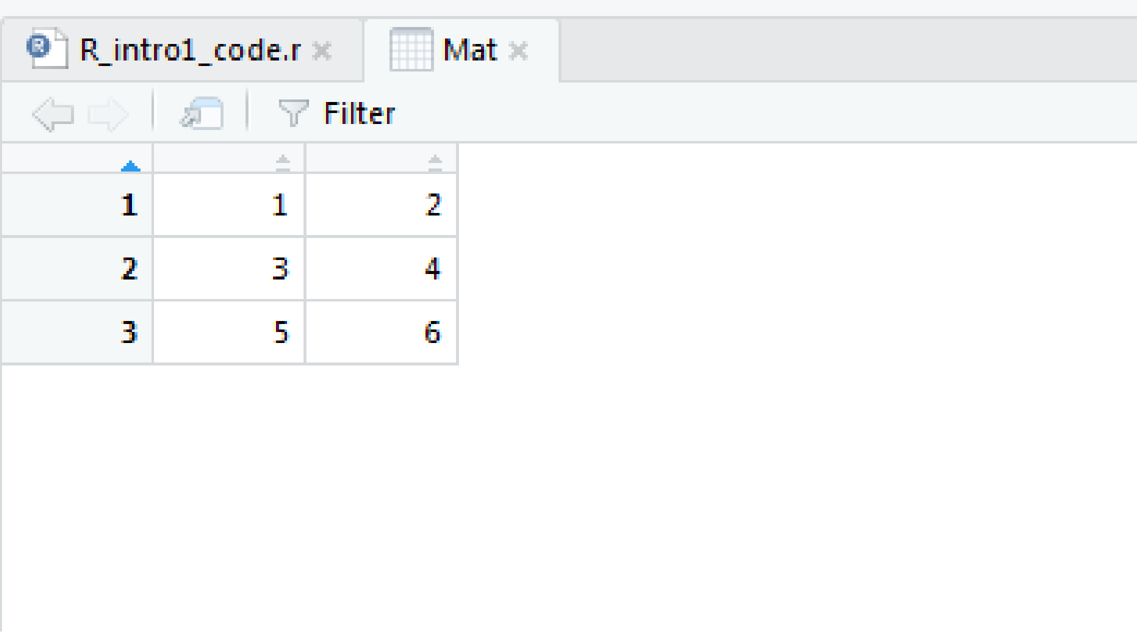
```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

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.

> #R is just a sophisticated calculator
> 6^6
[1] 46656
> 5+210
[1] 215
> 3-10
[1] -7
> 
```

Rstudio basics

- Rstudio also allows you to view the objects you are working with.
- This can provide a more user friendly interface that is more similar to excel



```
Data
  Mat            num [1:3, 1:2] 1 3 5 2 4 6
  Mat.bycol      num [1:3, 1:2] 1 2 3 4 5 6
Values
  Vec            num [1:7] 3 4 5 2 10 4 6
```

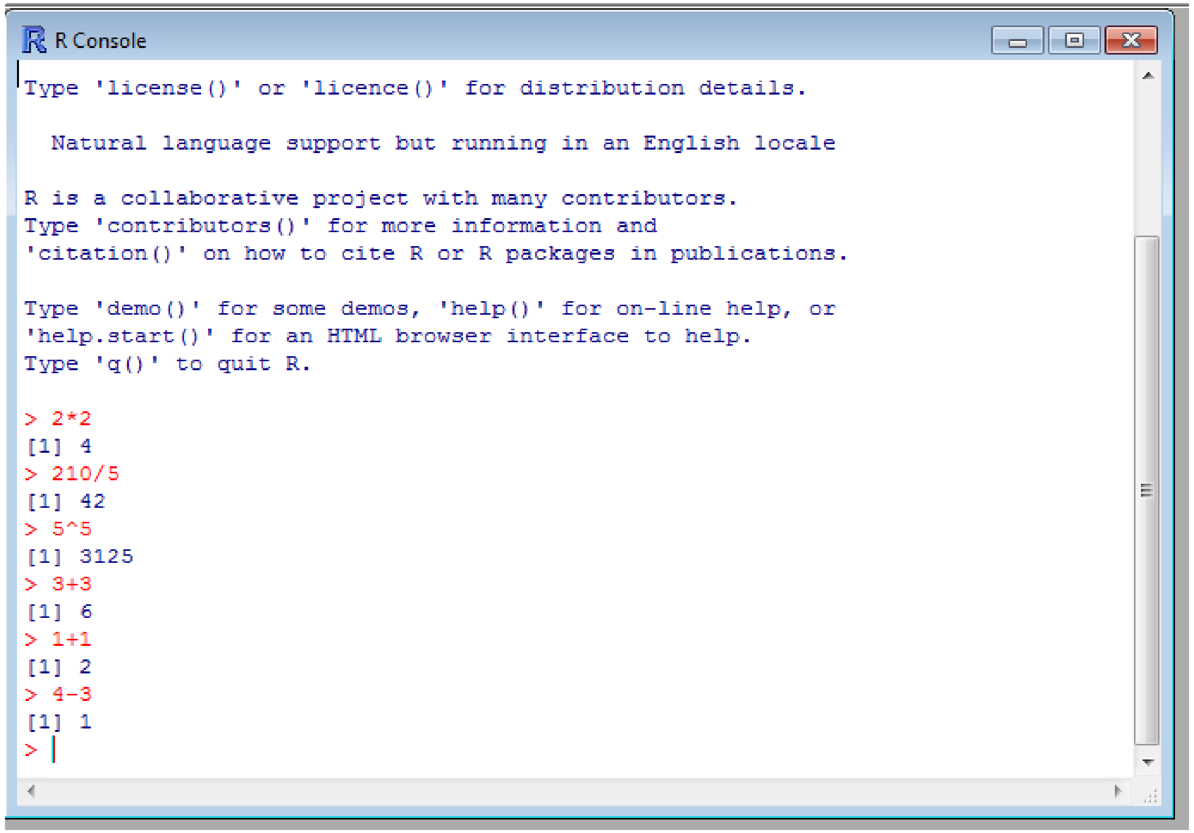


1	1	2
2	3	4
3	5	6

Working with numbers in R

- R acts like a calculator:

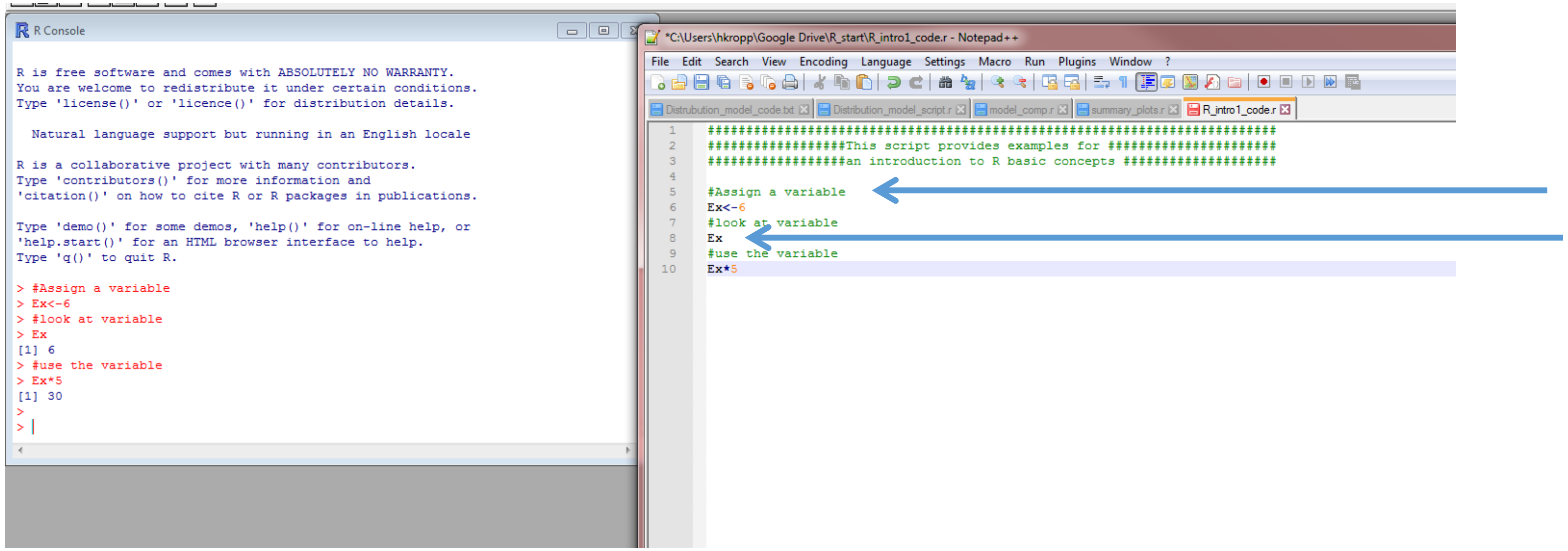
Red=inputs from your script
Blue= output from R



```
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.

> 2*2
[1] 4
> 210/5
[1] 42
> 5^5
[1] 3125
> 3+3
[1] 6
> 1+1
[1] 2
> 4-3
[1] 1
> |
```

Assigning variables in R

- Give an item a name using the <- followed by the item
  - This allows you to refer back to items without having to remember them or write huge amounts of code
- Comment your code using # in front of the line
  - Commenting allows you to keep track of what you are doing and provide reminders for later


R Console
*C:\Users\hkropp\Google Drive\R_start\R_intro1_code.r - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Distrubution_model_code.txt
Distribution_model_script.r
model_comp.r
summary_plots.r
R_intro1_code.r

```
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.

> #Assign a variable
> Ex<-6
> #look at variable
> Ex
[1] 6
> #use the variable
> Ex*5
[1] 30
>
> 
```

```
###########################################################################
##################This script provides examples for ######################
##################an introduction to R basic concepts ####################

#Assign a variable
Ex<-6
#look at variable
Ex
#use the variable
Ex*5
```

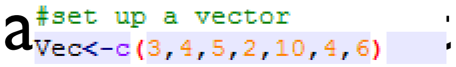
comments

variable

Vectors and matrix in R

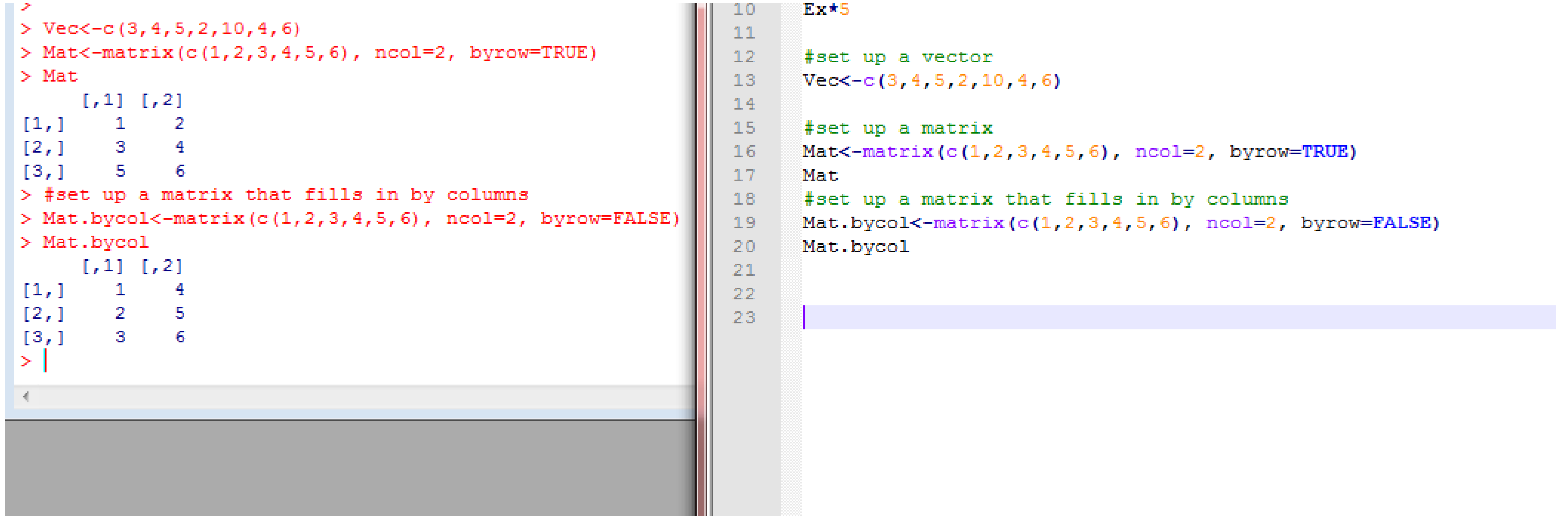
- R automatically treats inputs like they are vectors.
- Create a vector using c()

- Set up a ma



```
#set up a vector
Vec<-c(3,4,5,2,10,4,6)
```

rix()

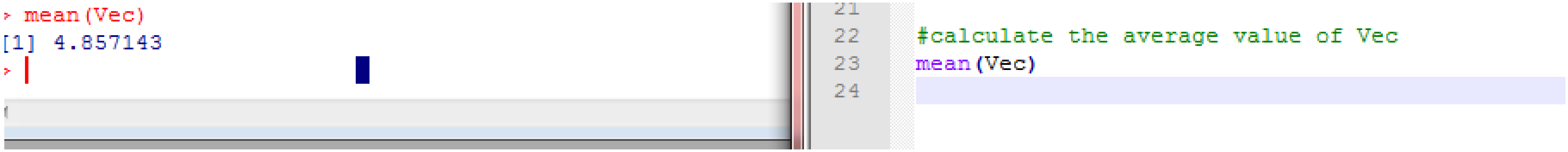


```
>
> Vec<-c(3,4,5,2,10,4,6)
> Mat<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=TRUE)
> Mat
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> #set up a matrix that fills in by columns
> Mat.bycol<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=FALSE)
> Mat.bycol
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> 
```

```
Ex*5

#set up a vector
Vec<-c(3,4,5,2,10,4,6)

#set up a matrix
Mat<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=TRUE)
Mat
#set up a matrix that fills in by columns
Mat.bycol<-matrix(c(1,2,3,4,5,6), ncol=2, byrow=FALSE)
Mat.bycol
```

Functions in R

- Using matrix() is an example of a function in R
- There are a lot of “built in” functions in R that make it easier to work with data or statistics
- For example, calculating an average using mean():

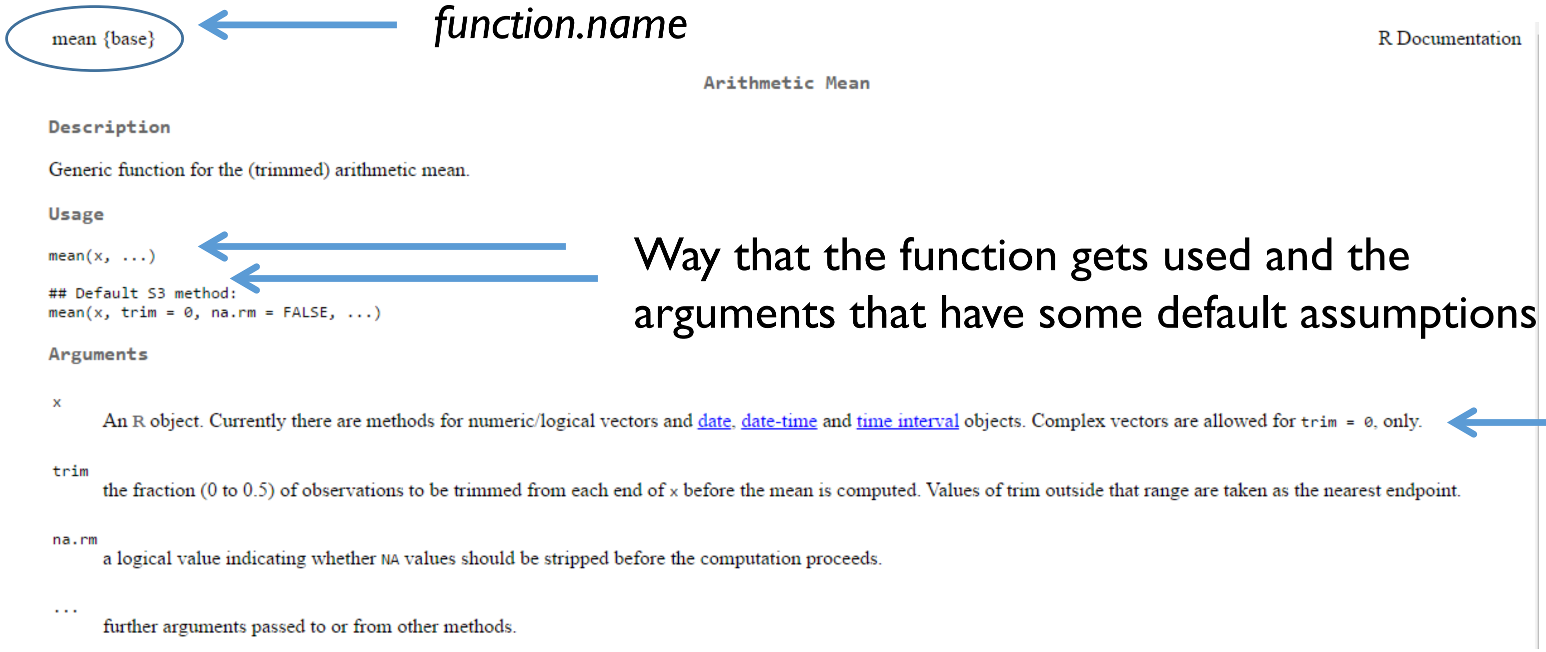


```
> mean(Vec)
[1] 4.857143
> 
```

```
#calculate the average value of Vec
mean(Vec)
```

Functions in R

- A **function** typically gives you an **output** based on the inputs you give it
- The inputs needed for a function are often called **arguments**
- R will have a description of the arguments and output for its function
  - google the *function.name*
  - type help(*function.name*) in the console



mean {base}

R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

`x`
An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.

`trim`
the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

`na.rm`
a logical value indicating whether `NA` values should be stripped before the computation proceeds.

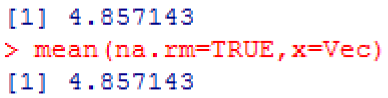
`...`
further arguments passed to or from other methods.

function.name

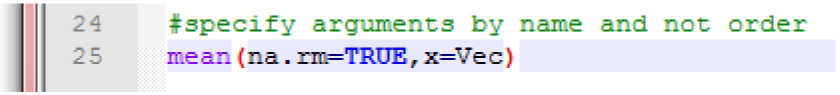
Way that the function gets used and the arguments that have some default assumptions

Detailed description

- Here *x*, *trim*, and *na.rm* are the names of arguments
- The order here is important because if the names of the arguments aren't used, then they are assumed to be in this default order.
- If the default arguments are sufficient there is no need to include them in the function
- If you don't want to use the default order than you can specify



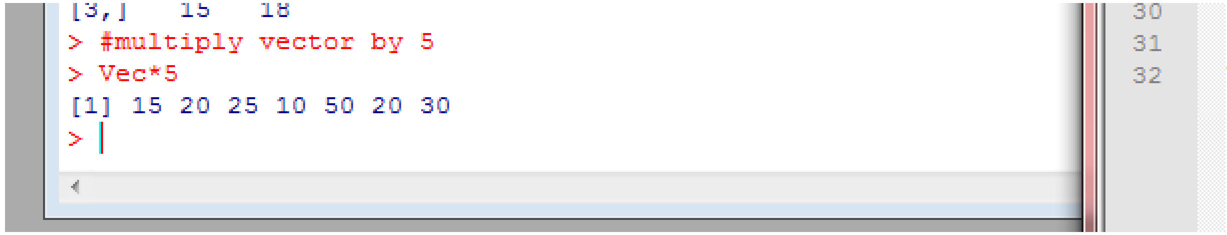
```
[1] 4.857143
> mean(na.rm=TRUE,x=Vec)
[1] 4.857143
```



```
#specify arguments by name and not order
mean(na.rm=TRUE,x=Vec)
```

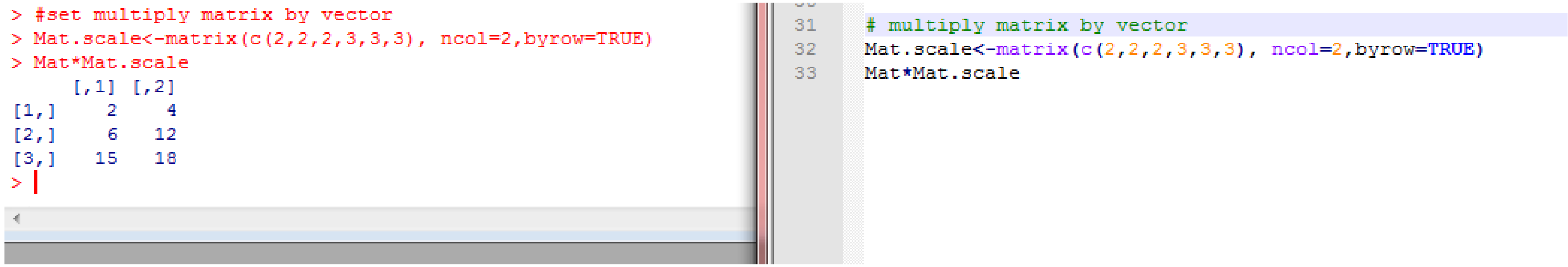
Working with vectors and matrices:

- Keep in mind that R automatically does vector/matrix math:



```
[3,]    15    18
> #multiply vector by 5
> Vec*5
[1] 15 20 25 10 50 20 30
> 
```

```
#multiply vector by 5
Vec*5
```



```
> #set multiply matrix by vector
> Mat.scale<-matrix(c(2,2,2,3,3,3), ncol=2,byrow=TRUE)
> Mat*Mat.scale
     [,1] [,2]
[1,]    2    4
[2,]    6   12
[3,]   15   18
> 
```

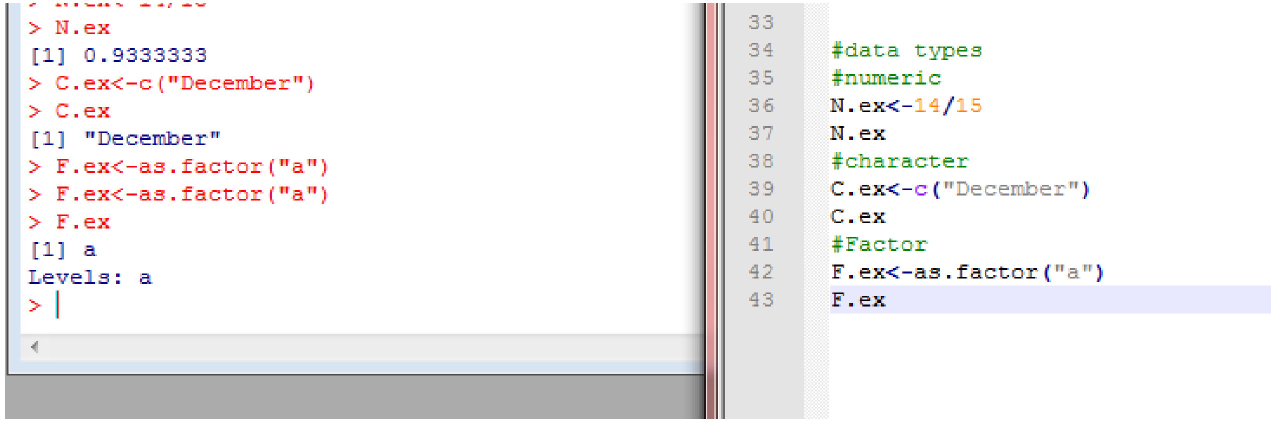
```
# multiply matrix by vector
Mat.scale<-matrix(c(2,2,2,3,3,3), ncol=2,byrow=TRUE)
Mat*Mat.scale
```

Packages in R

- People have created thousands of **packages** to add more functions to R
- Packages allow you to download only the ones you want to use (it would take up a lot of space)
- Some functions may have the same name in different packages so be sure to note potential overlap in packages
  - only load the ones you are going to use

Data Types

- Numeric: number can have any number of decimals
- Character: text
- Factor: text but a short identifying category name

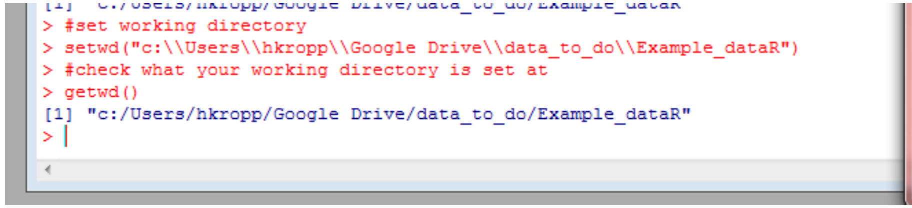


```
> N.ex<-14/15
> N.ex
[1] 0.9333333
> C.ex<-c("December")
> C.ex
[1] "December"
> F.ex<-as.factor("a")
> F.ex<-as.factor("a")
> F.ex
[1] a
Levels: a
> 
```

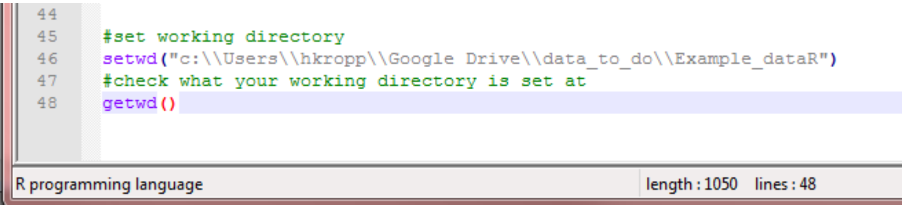
```

#data types
#numeric
N.ex<-14/15
N.ex
#character
C.ex<-c("December")
C.ex
#Factor
F.ex<-as.factor("a")
F.ex
```

Reading in Data

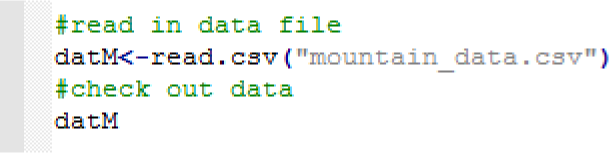
- The easiest and most consistent way to read in data in R is through a comma separated text file (.csv)
- You need to tell R where to find the data
  - Set a working directory to always get files from one folder
  - Or specify the file path with the csv name
- File/File paths always need to be in quotes and file paths always have \\ between folders



```
[1] "c:/Users/hkropp/Google Drive/data_to_do/Example_dataR"
> #set working directory
> setwd("c:\\Users\\hkropp\\Google Drive\\data_to_do\\Example_dataR")
> #check what your working directory is set at
> getwd()
[1] "c:/Users/hkropp/Google Drive/data_to_do/Example_dataR"
> 
```

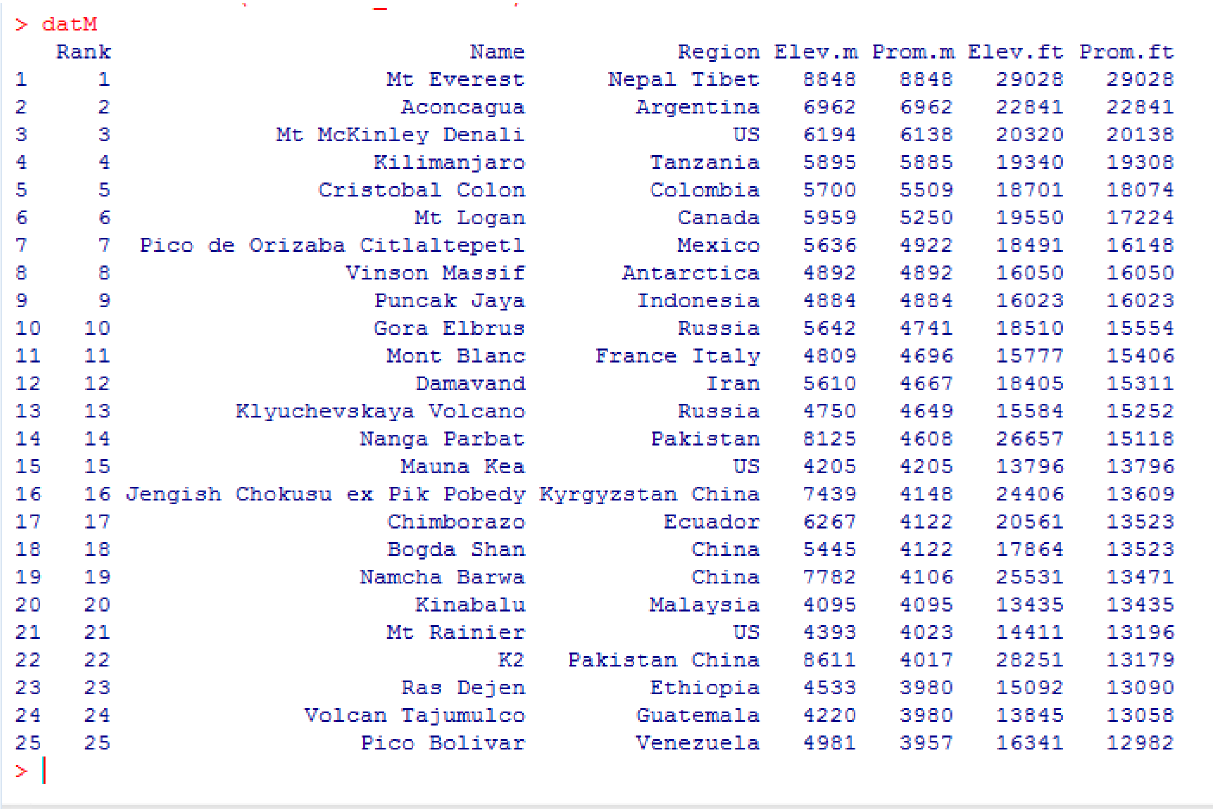


```
#set working directory
setwd("c:\\Users\\hkropp\\Google Drive\\data_to_do\\Example_dataR")
#check what your working directory is set at
getwd()
```



```
#read in data file
datM<-read.csv("mountain_data.csv")
#check out data
datM
```

- Always think about your names
  - Length
  - Clarity
- Capitalization matters!



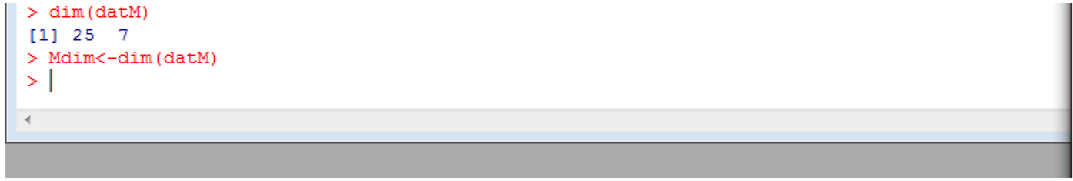
```
> datM
   Rank                          Name          Region Elev.m Prom.m Elev.ft Prom.ft
1     1                    Mt Everest     Nepal Tibet   8848   8848   29028   29028
2     2                     Aconcagua       Argentina   6962   6962   22841   22841
3     3             Mt McKinley Denali              US   6194   6138   20320   20138
4     4                   Kilimanjaro        Tanzania   5895   5885   19340   19308
5     5                Cristobal Colon        Colombia   5700   5509   18701   18074
6     6                      Mt Logan          Canada   5959   5250   19550   17224
7     7  Pico de Orizaba Citlaltepetl          Mexico   5636   4922   18491   16148
8     8                 Vinson Massif      Antarctica   4892   4892   16050   16050
9     9                   Puncak Jaya       Indonesia   4884   4884   16023   16023
10   10                   Gora Elbrus          Russia   5642   4741   18510   15554
11   11                    Mont Blanc    France Italy   4809   4696   15777   15406
12   12                      Damavand            Iran   5610   4667   18405   15311
13   13            Klyuchevskaya Volcano          Russia   4750   4649   15584   15252
14   14                   Nanga Parbat        Pakistan   8125   4608   26657   15118
15   15                     Mauna Kea              US   4205   4205   13796   13796
16   16 Jengish Chokusu ex Pik Pobedy Kyrgyzstan China   7439   4148   24406   13609
17   17                    Chimborazo         Ecuador   6267   4122   20561   13523
18   18                    Bogda Shan           China   5445   4122   17864   13523
19   19                  Namcha Barwa           China   7782   4106   25531   13471
20   20                      Kinabalu        Malaysia   4095   4095   13435   13435
21   21                    Mt Rainier              US   4393   4023   14411   13196
22   22                            K2  Pakistan China   8611   4017   28251   13179
23   23                     Ras Dejen        Ethiopia   4533   3980   15092   13090
24   24              Volcan Tajumulco       Guatemala   4220   3980   13845   13058
25   25                  Pico Bolivar       Venezuela   4981   3957   16341   12982
> 
```

Properties of a data frame

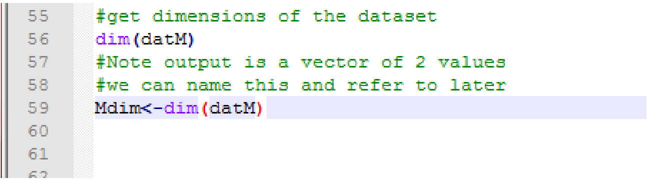
- Typically the columns have names
- All columns are the same length
- There can be different types of data in each column

Basic Info about a data frame

- Get dimensions

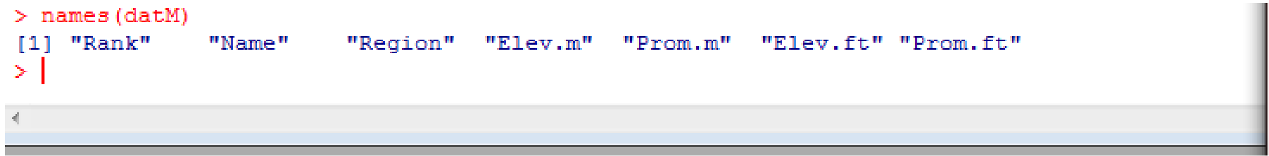


```
> dim(datM)
[1] 25  7
> Mdim<-dim(datM)
> 
```

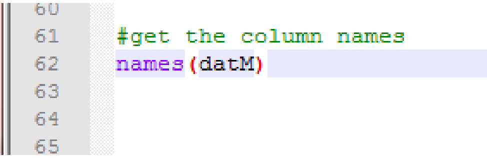


```
#get dimensions of the dataset
dim(datM)
#Note output is a vector of 2 values
#we can name this and refer to later
Mdim<-dim(datM)
```

- Names of columns

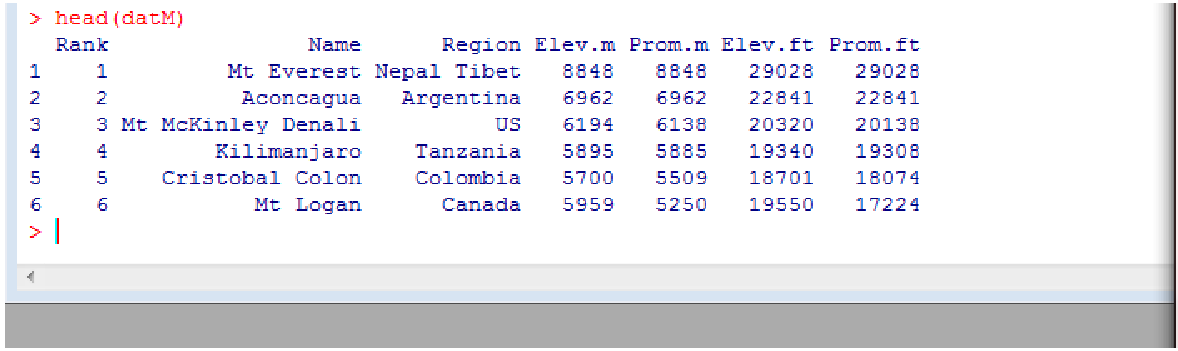


```
> names(datM)
[1] "Rank"    "Name"    "Region"  "Elev.m"  "Prom.m"  "Elev.ft" "Prom.ft"
> 
```

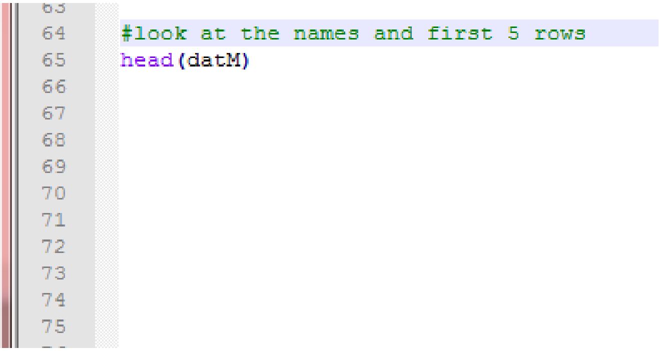


```
#get the column names
names(datM)
```

- See what it looks like



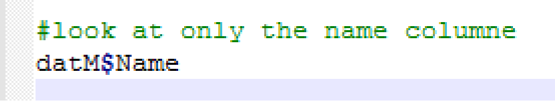
```
> head(datM)
  Rank              Name       Region Elev.m Prom.m Elev.ft Prom.ft
1    1        Mt Everest Nepal Tibet   8848   8848   29028   29028
2    2         Aconcagua    Argentina   6962   6962   22841   22841
3    3 Mt McKinley Denali           US   6194   6138   20320   20138
4    4       Kilimanjaro     Tanzania   5895   5885   19340   19308
5    5    Cristobal Colon     Colombia   5700   5509   18701   18074
6    6          Mt Logan       Canada   5959   5250   19550   17224
> 
```



```
#look at the names and first 5 rows
head(datM)
```

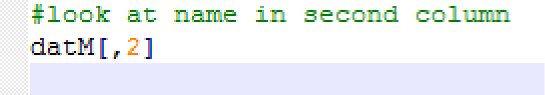
Referring to data in data frames

- A column can be used by: *data.frame$column*



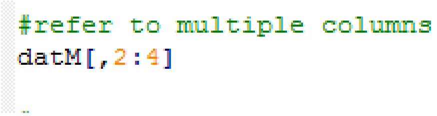
```
#look at only the name columne
datM$Name
```

- Data frames can also be refered to like matrix where [rows,columns] notation is used
  - Refer to a column without calling its name:



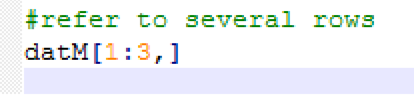
```
#look at name in second column
datM[,2]
```

  - Multiple columns



```
#refer to multiple columns
datM[,2:4]
```

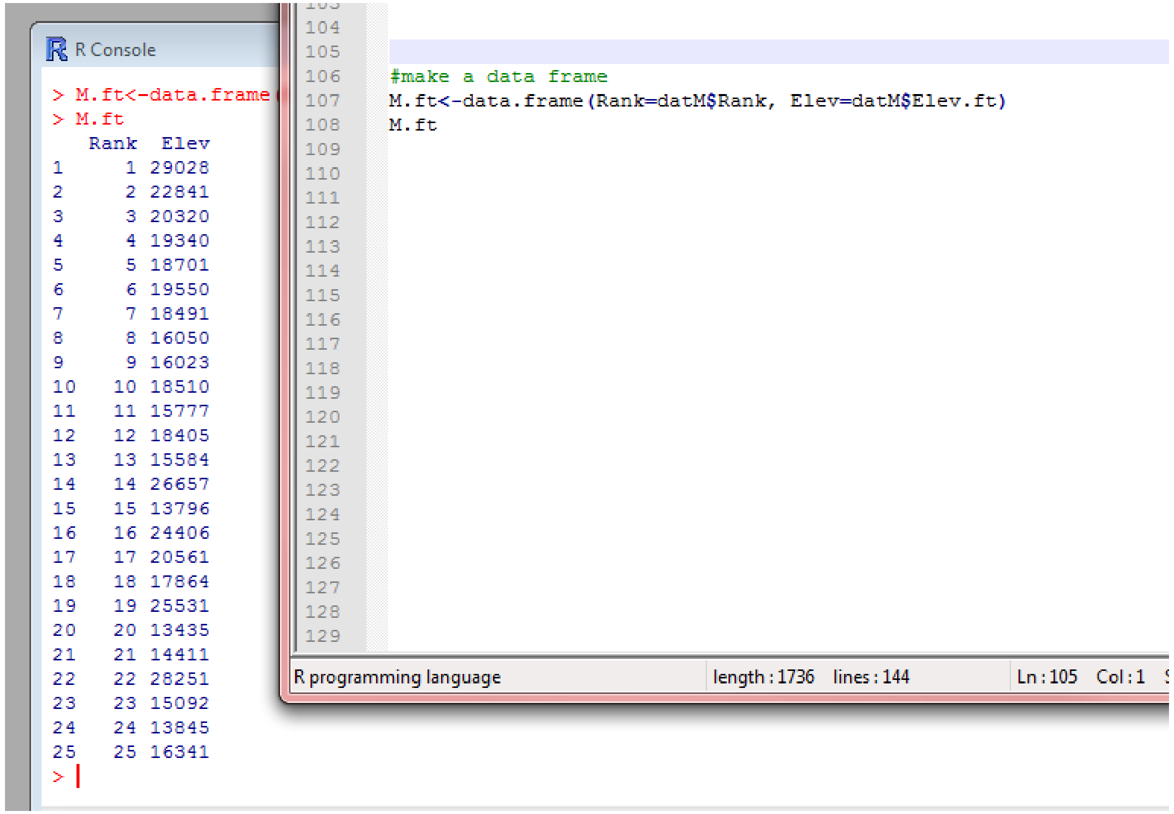
  - Rows:



```
#refer to several rows
datM[1:3,]
```

Creating a data frame

- Use the function data.frame
  - Note: all vectors must be equal lengths

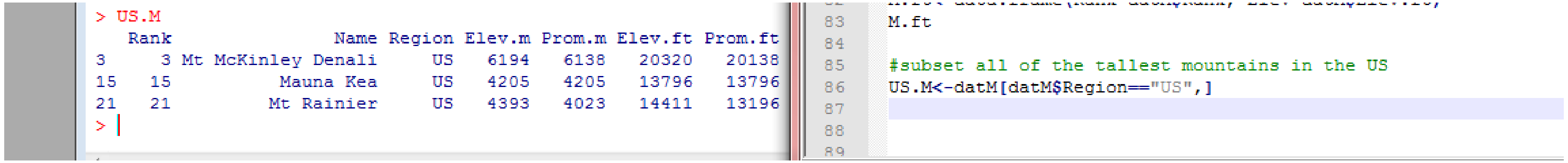


```
#make a data frame
M.ft<-data.frame(Rank=datM$Rank, Elev=datM$Elev.ft)
M.ft
```

```
> M.ft<-data.frame(
> M.ft
   Rank  Elev
1     1 29028
2     2 22841
3     3 20320
4     4 19340
5     5 18701
6     6 19550
7     7 18491
8     8 16050
9     9 16023
10   10 18510
11   11 15777
12   12 18405
13   13 15584
14   14 26657
15   15 13796
16   16 24406
17   17 20561
18   18 17864
19   19 25531
20   20 13435
21   21 14411
22   22 28251
23   23 15092
24   24 13845
25   25 16341
> 
```

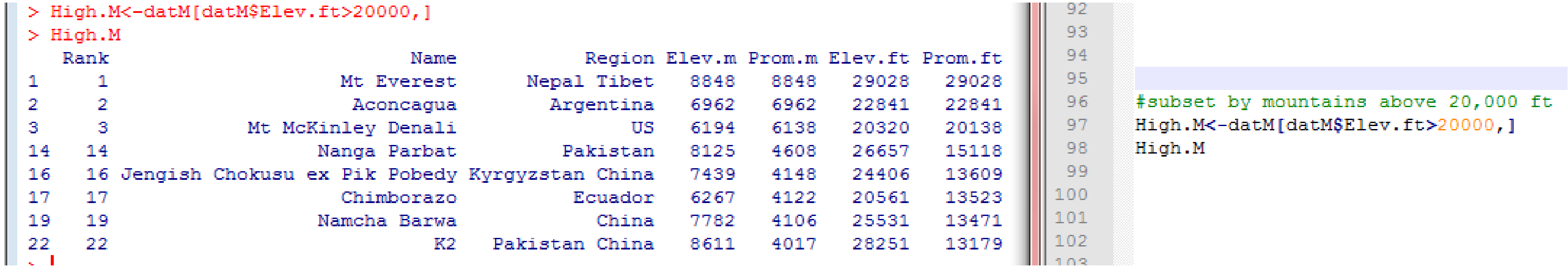
Subset data

- Data can be subset by a characteristic
- This is done using logical expressions (see R's guide for logical expressions.
    - https://www.r-bloggers.com/logical-operators-in-r/
- Subset with brackets:
    - Mountains in the US



```
> US.M
   Rank                Name Region Elev.m Prom.m Elev.ft Prom.ft
3     3 Mt McKinley Denali     US   6194   6138   20320   20138
15   15          Mauna Kea     US   4205   4205   13796   13796
21   21         Mt Rainier     US   4393   4023   14411   13196
> 
```

```
M.ft

#subset all of the tallest mountains in the US
US.M<-datM[datM$Region=="US",]
```



```
> High.M<-datM[datM$Elev.ft>20000,]
> High.M
   Rank                           Name           Region Elev.m Prom.m Elev.ft Prom.ft
1     1                     Mt Everest      Nepal Tibet   8848   8848   29028   29028
2     2                      Aconcagua        Argentina   6962   6962   22841   22841
3     3             Mt McKinley Denali               US   6194   6138   20320   20138
14   14                   Nanga Parbat         Pakistan   8125   4608   26657   15118
16   16 Jengish Chokusu ex Pik Pobedy Kyrgyzstan China   7439   4148   24406   13609
17   17                     Chimborazo          Ecuador   6267   4122   20561   13523
19   19                  Namcha Barwa             China   7782   4106   25531   13471
22   22                             K2   Pakistan China   8611   4017   28251   13179
```

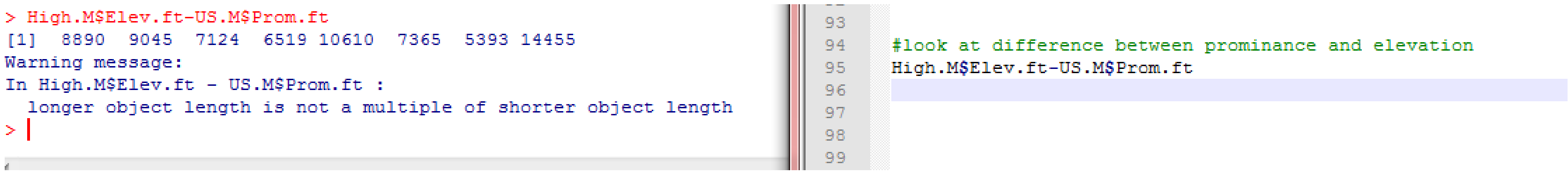
```

#subset by mountains above 20,000 ft
High.M<-datM[datM$Elev.ft>20000,]
High.M
```

Missing data

- NA indicates that the data is missing in R
- If there are blank cells in a data file R will automatically fill them in with NA
- You can also designate what marks an NA if it differs in a data file:

Errors

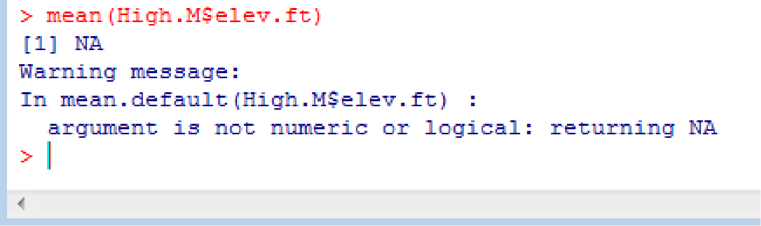
- Error messages look intimidating at first in R, but they are actually very useful
- Some kinds of examples:
  - Trying to do something where vectors are different lengths



```
> High.M$Elev.ft-US.M$Prom.ft
[1]  8890  9045  7124  6519 10610  7365  5393 14455
Warning message:
In High.M$Elev.ft - US.M$Prom.ft :
  longer object length is not a multiple of shorter object length
> 
```

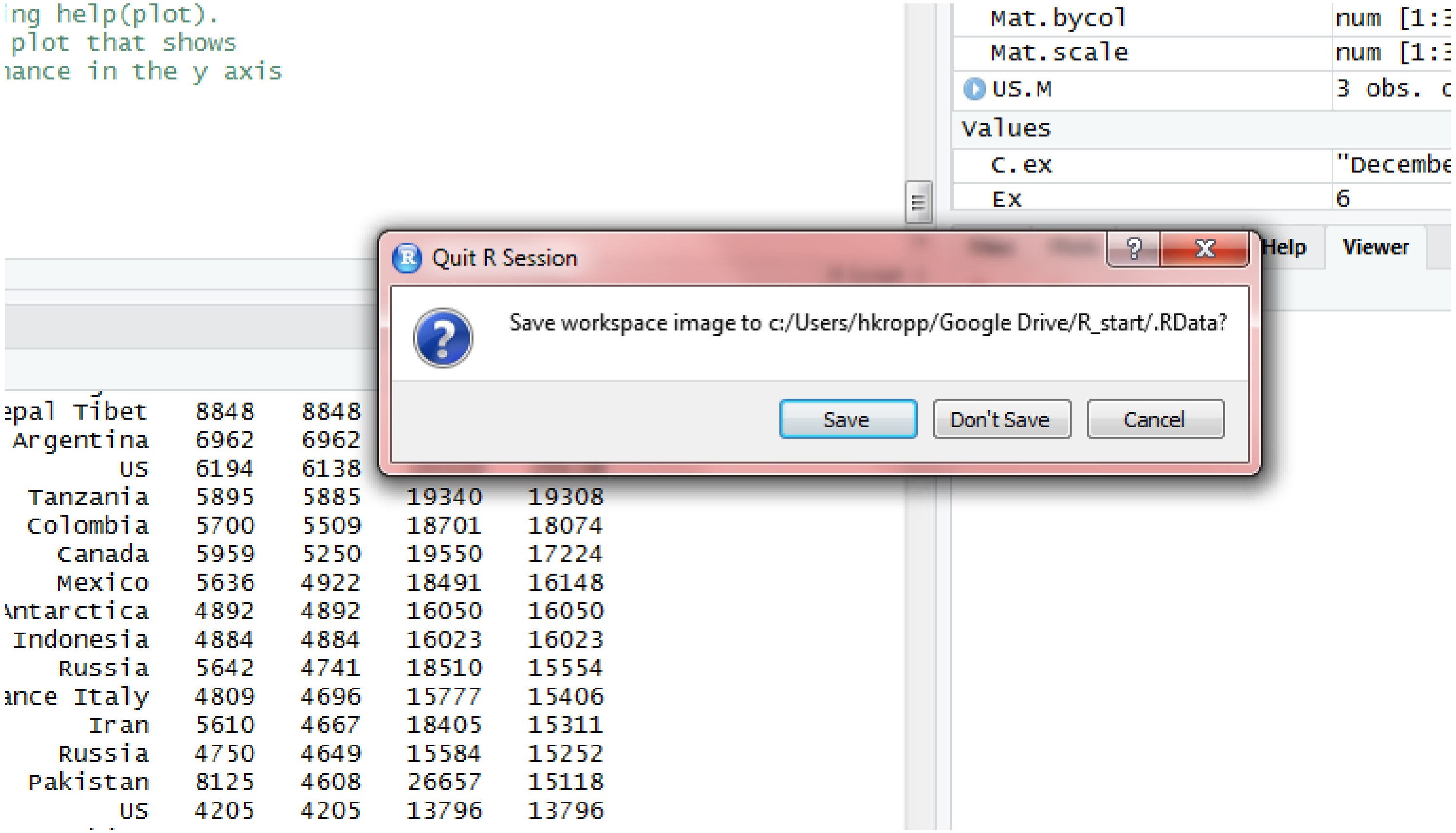
```
#look at difference between prominance and elevation
High.M$Elev.ft-US.M$Prom.ft
```

  - Referring to names incorrectly (capitalization counts!)



```
> mean(High.M$elev.ft)
[1] NA
Warning message:
In mean.default(High.M$elev.ft) :
  argument is not numeric or logical: returning NA
> 
```

- You will get this message when you close out Rstudio.
- The answer is ALWAYS DON'T SAVE!!!!!!!!!!!!!!!!!!!!!!
  - I cannot repeat this enough



```
ing help(plot).
 plot that shows
iance in the y axis
```

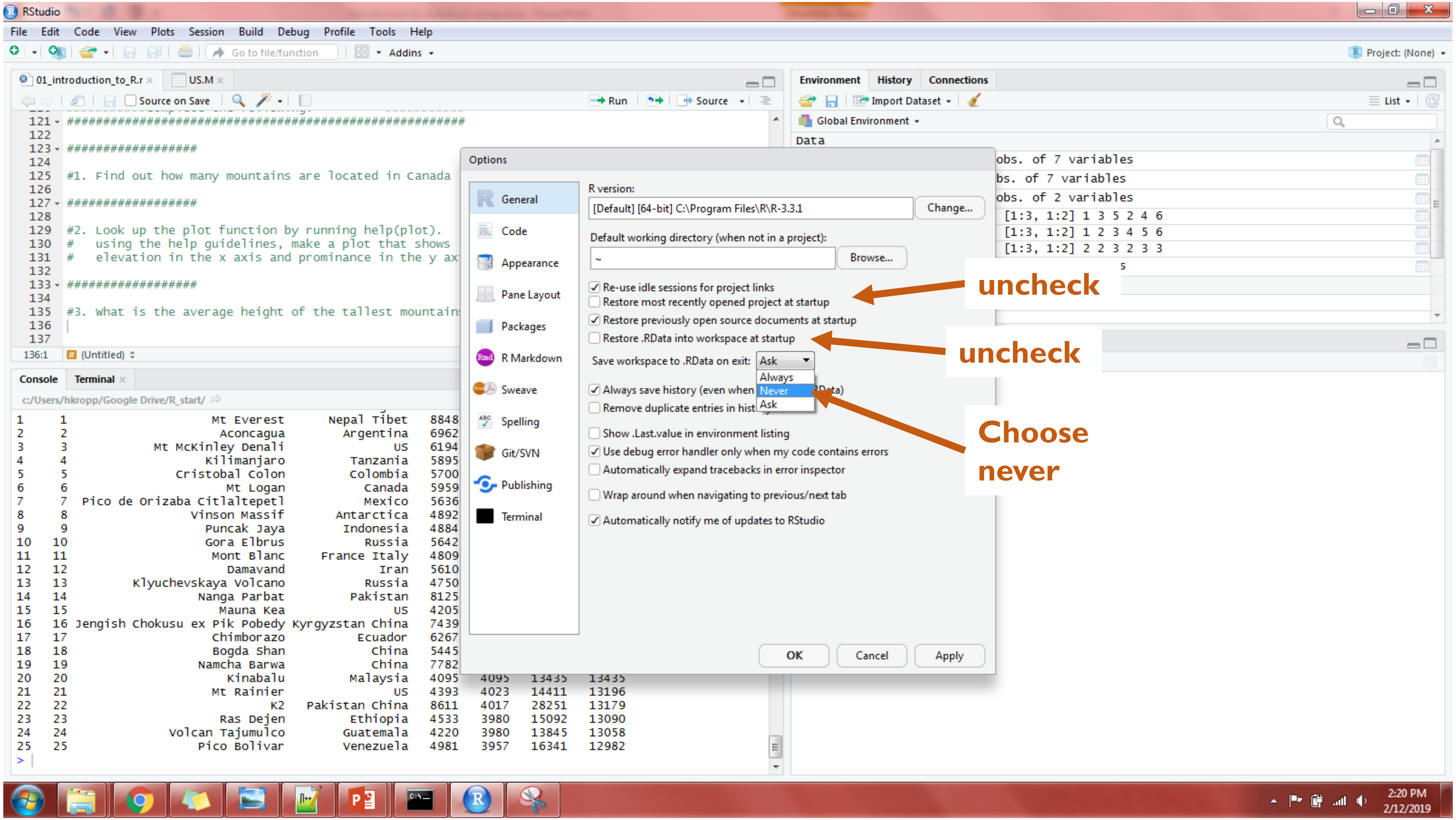
```
  Mat.bycol        num [1:3
  Mat.scale        num [1:3
  US.M             3 obs. c
Values
  C.ex             "Decembe
  Ex               6
```

Quit R Session

Save workspace image to c:/Users/hkropp/Google Drive/R_start/.RData?

```
epal Tibet   8848   8848
 Argentina   6962   6962
        US   6194   6138
  Tanzania   5895   5885   19340   19308
  Colombia   5700   5509   18701   18074
    Canada   5959   5250   19550   17224
    Mexico   5636   4922   18491   16148
Antarctica   4892   4892   16050   16050
 Indonesia   4884   4884   16023   16023
    Russia   5642   4741   18510   15554
ance Italy   4809   4696   15777   15406
      Iran   5610   4667   18405   15311
    Russia   4750   4649   15584   15252
  Pakistan   8125   4608   26657   15118
        US   4205   4205   13796   13796
```

In fact:



```
#############################################################

##################

#1. Find out how many mountains are located in Canada

##################

#2. Look up the plot function by running help(plot).
#   using the help guidelines, make a plot that shows
#   elevation in the x axis and prominence in the y ax

##################

#3. what is the average height of the tallest mountain
```

136:1 (Untitled)

Data

```
obs. of 7 variables
bs. of 7 variables
obs. of 2 variables
 [1:3, 1:2] 1 3 5 2 4 6
 [1:3, 1:2] 1 2 3 4 5 6
 [1:3, 1:2] 2 2 3 2 3 3
```

Options

- General
- Code
- Appearance
- Pane Layout
- Packages
- R Markdown
- Sweave
- Spelling
- Git/SVN
- Publishing
- Terminal

R version:

[Default] [64-bit] C:\Program Files\R\R-3.3.1 Change...

Default working directory (when not in a project):

~ Browse...

☑ Re-use idle sessions for project links

☐ Restore most recently opened project at startup

☑ Restore previously open source documents at startup

☐ Restore .RData into workspace at startup

Save workspace to .RData on exit: Ask ▾ (Always / Never / Ask)

☑ Always save history (even when ...RData)

☐ Remove duplicate entries in hist...

☐ Show .Last.value in environment listing

☑ Use debug error handler only when my code contains errors

☐ Automatically expand tracebacks in error inspector

☐ Wrap around when navigating to previous/next tab

☑ Automatically notify me of updates to RStudio

OK Cancel Apply

uncheck

uncheck

Choose never

c:/Users/hkropp/Google Drive/R_start/

```
1     1                      Mt Everest     Nepal Tibet   8848
2     2                       Aconcagua       Argentina   6962
3     3            Mt McKinley Denali              US   6194
4     4                     Kilimanjaro        Tanzania   5895
5     5                  Cristobal Colon        Colombia   5700
6     6                        Mt Logan          Canada   5959
7     7  Pico de Orizaba Citlaltepetl          Mexico   5636
8     8                   Vinson Massif      Antarctica   4892
9     9                     Puncak Jaya       Indonesia   4884
10   10                     Gora Elbrus          Russia   5642
11   11                      Mont Blanc    France Italy   4809
12   12                        Damavand            Iran   5610
13   13             Klyuchevskaya Volcano          Russia   4750
14   14                    Nanga Parbat        Pakistan   8125
15   15                       Mauna Kea              US   4205
16   16 Jengish Chokusu ex Pik Pobedy Kyrgyzstan China   7439
17   17                      Chimborazo         Ecuador   6267
18   18                      Bogda Shan           China   5445
19   19                    Namcha Barwa           China   7782
20   20                        Kinabalu        Malaysia   4095   4095   13435   13435
21   21                      Mt Rainier              US   4393   4023   14411   13196
22   22                              K2  Pakistan China   8611   4017   28251   13179
23   23                       Ras Dejen        Ethiopia   4533   3980   15092   13090
24   24                 Volcan Tajumulco       Guatemala   4220   3980   13845   13058
25   25                    Pico Bolivar       Venezuela   4981   3957   16341   12982
>
```

Why??

- This saves everything you ran in your previous session and loads it as if you just ran it
- It is easy to loose control of the variables and data in your workspace. You may have things you tested out and didn't work still in the environment.
- It is the easiest way to lead to confusion and problems with reproducing your analyses
- You should be writing nice, clean scripts that can be rerun upon opening. If something works, it's not magic. You wrote your script correctly.