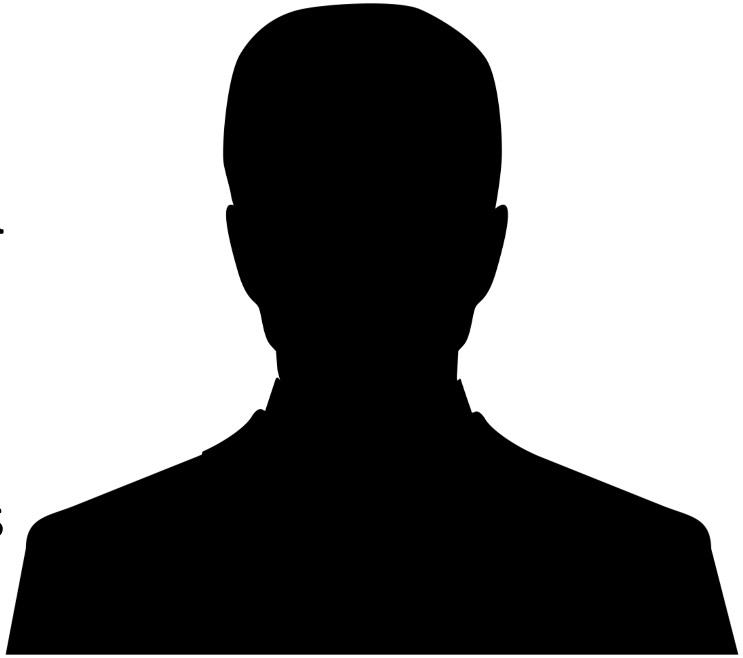
R Programming Language

Lesson Preview

Goals:

- Understand when to use R & when not use it.
- Understand basic syntax & write short programs.
- Understand scalability issues & ways to resolve them.

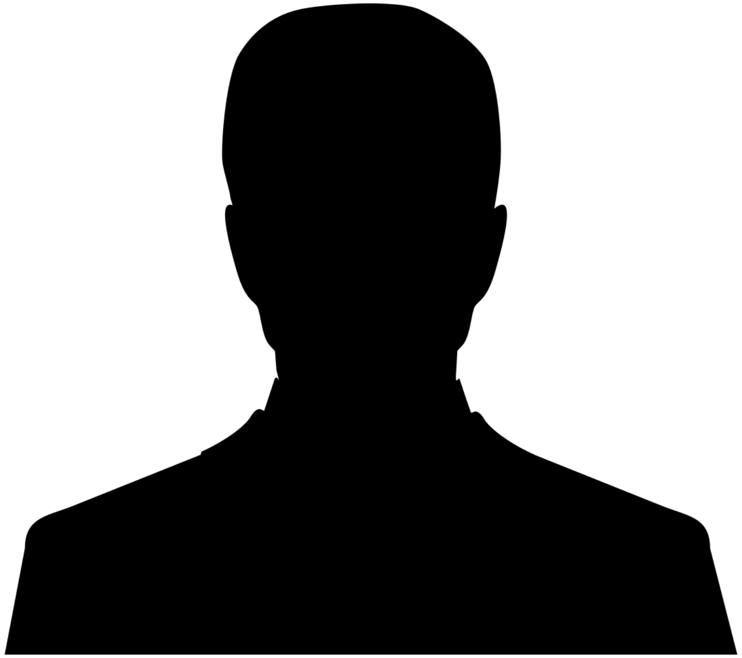


R Programming Language

Lesson Preview

Four parts of this lesson:

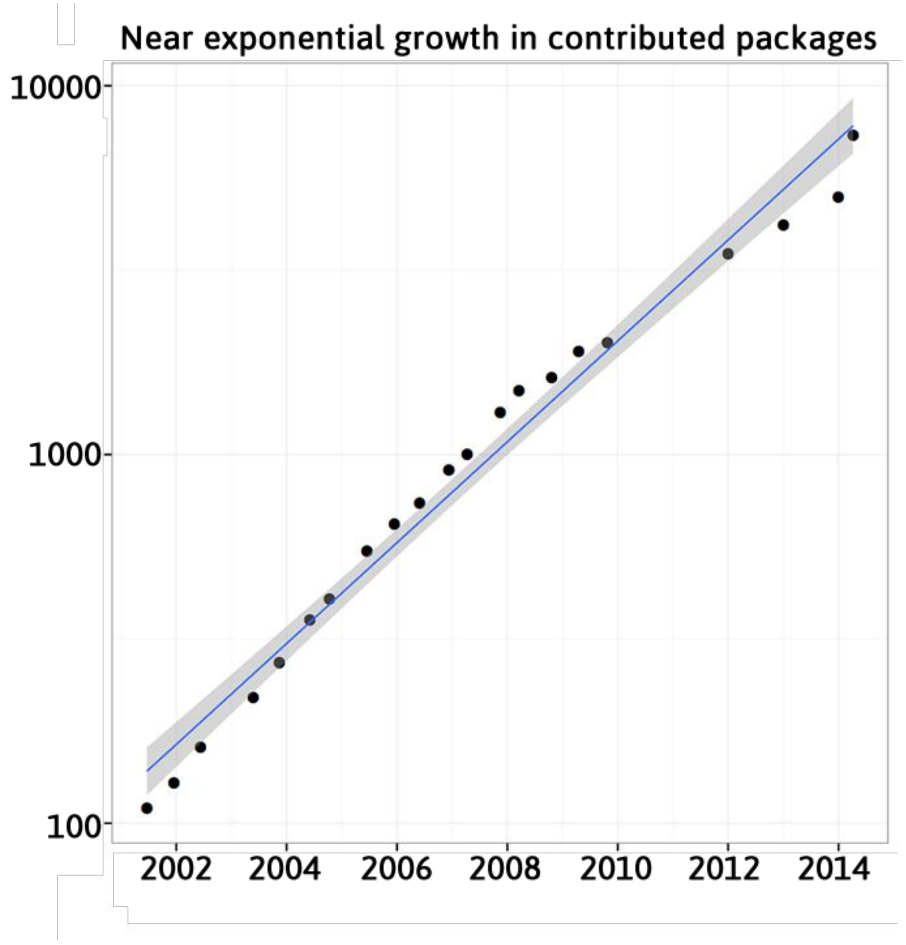
- getting started
- data types
- control flow and functions
- scalability and interfaces



R, Python, and Matlab Similarities

Characteristic	R	Python	Matlab
Run in interactive shell or graphical UI	x	x	x
Store and manipulate data as arrays	x	x	x
Many packages	x	x	x
Slower than C, C++	x	x	x
Interface with C++	x	x	x

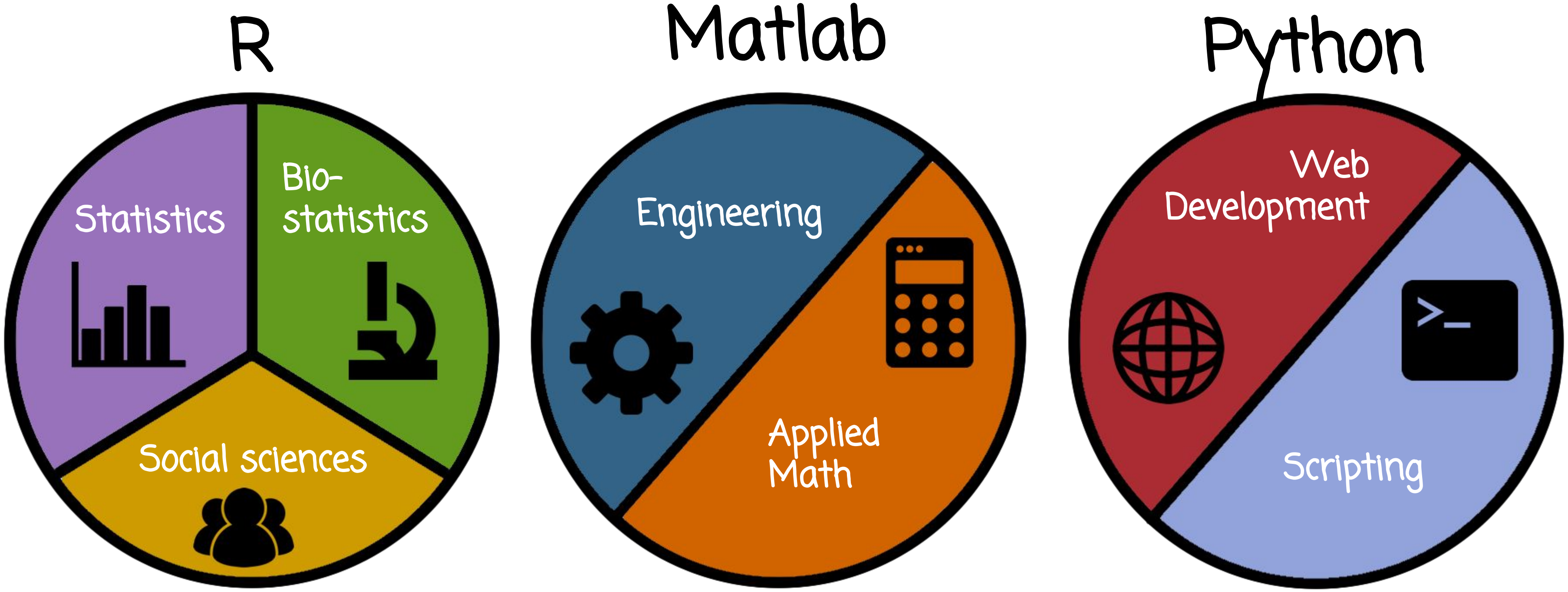
R, Python, and Matlab Differences



R, Python, and Matlab Differences

Characteristic	R	Python	Matlab
Open source	x	x	
Ease of Contribution	x		
Quality of Contributions	x		
Suitable for Statistics	x		
Better Graphics Capabilities	x		

R, Python, and Matlab Differences



Running R

Interactively

Non-Interactively

Running R - Interactively

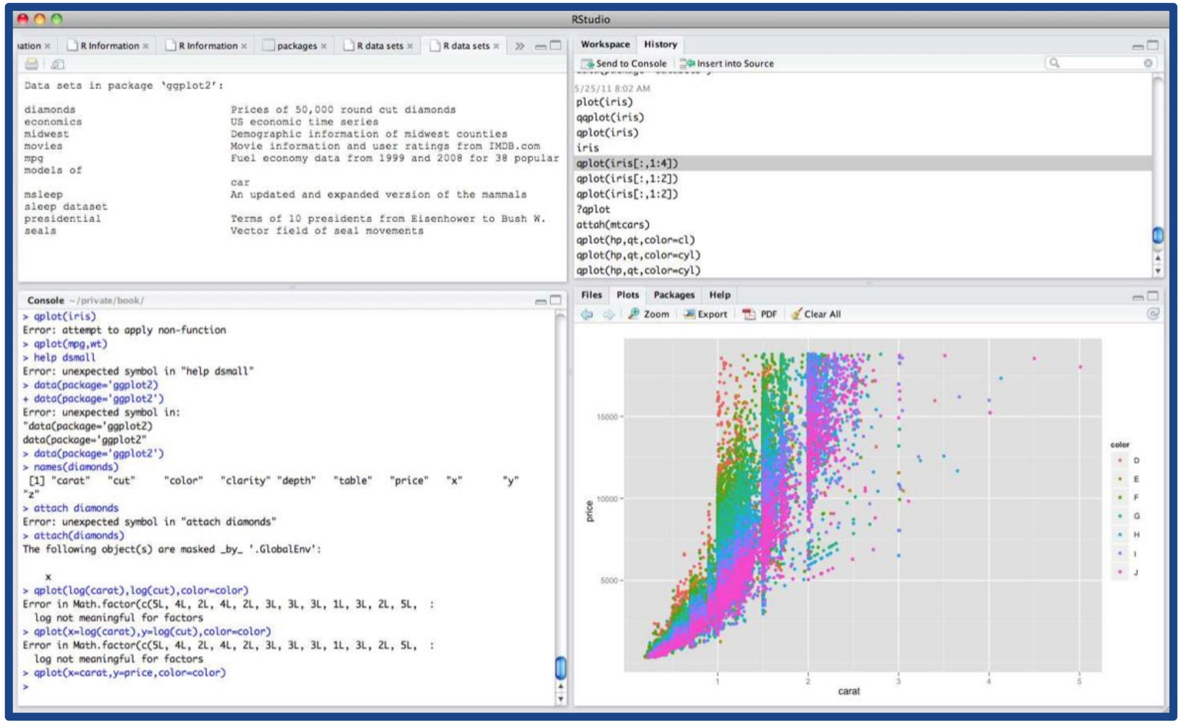
RStudio

Running R - Interactively

From the terminal:

MAC	`$ open -a RStudio .`
Linux	`$ rstudio`

Running R - Interactively

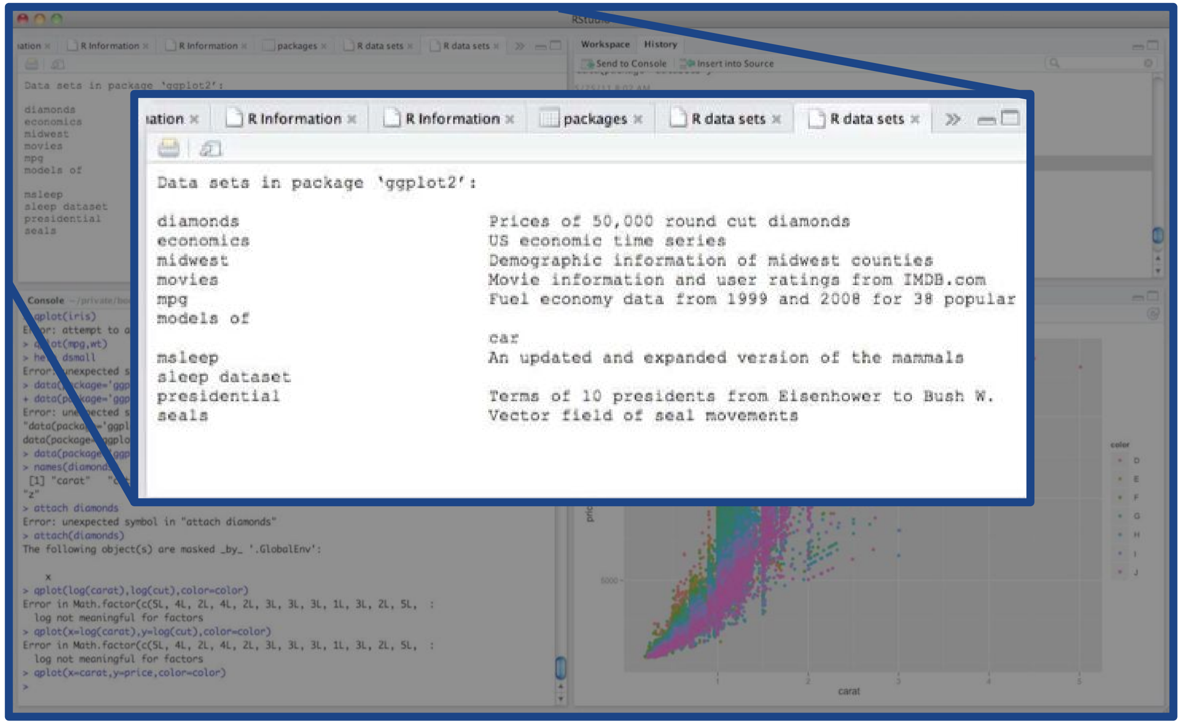


```
Data sets in package 'ggplot2':

diamonds                          Prices of 50,000 round cut diamonds
economics                         US economic time series
midwest                           Demographic information of midwest counties
movies                            Movie information and user ratings from IMDB.com
mpg                               Fuel economy data from 1999 and 2008 for 38 popular
models of
                                  car
msleep                            An updated and expanded version of the mammals
sleep dataset
presidential                      Terms of 10 presidents from Eisenhower to Bush W.
seals                             Vector field of seal movements
```

```
5/25/11 8:02 AM
plot(iris)
qqplot(iris)
qplot(iris)
iris
qplot(iris[:,1:4])
qplot(iris[:,1:2])
qplot(iris[:,1:2])
?qplot
attah(mtcars)
qplot(hp,qt,color=cl)
qplot(hp,qt,color=cyl)
qplot(hp,qt,color=cyl)
```

```
> qplot(iris)
Error: attempt to apply non-function
> qplot(mpg,wt)
> help dsmall
Error: unexpected symbol in "help dsmall"
> data(package='ggplot2)
+ data(package='ggplot2')
Error: unexpected symbol in:
"data(package='ggplot2)
data(package='ggplot2"
> data(package='ggplot2')
> names(diamonds)
 [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"   "x"       "y"
"z"
> attach diamonds
Error: unexpected symbol in "attach diamonds"
> attach(diamonds)
The following object(s) are masked _by_ '.GlobalEnv':

    x
> qplot(log(carat),log(cut),color=color)
Error in Math.factor(c(5L, 4L, 2L, 4L, 2L, 3L, 3L, 3L, 1L, 3L, 2L, 5L,  :
  log not meaningful for factors
> qplot(x=log(carat),y=log(cut),color=color)
Error in Math.factor(c(5L, 4L, 2L, 4L, 2L, 3L, 3L, 3L, 1L, 3L, 2L, 5L,  :
  log not meaningful for factors
> qplot(x=carat,y=price,color=color)
>
```

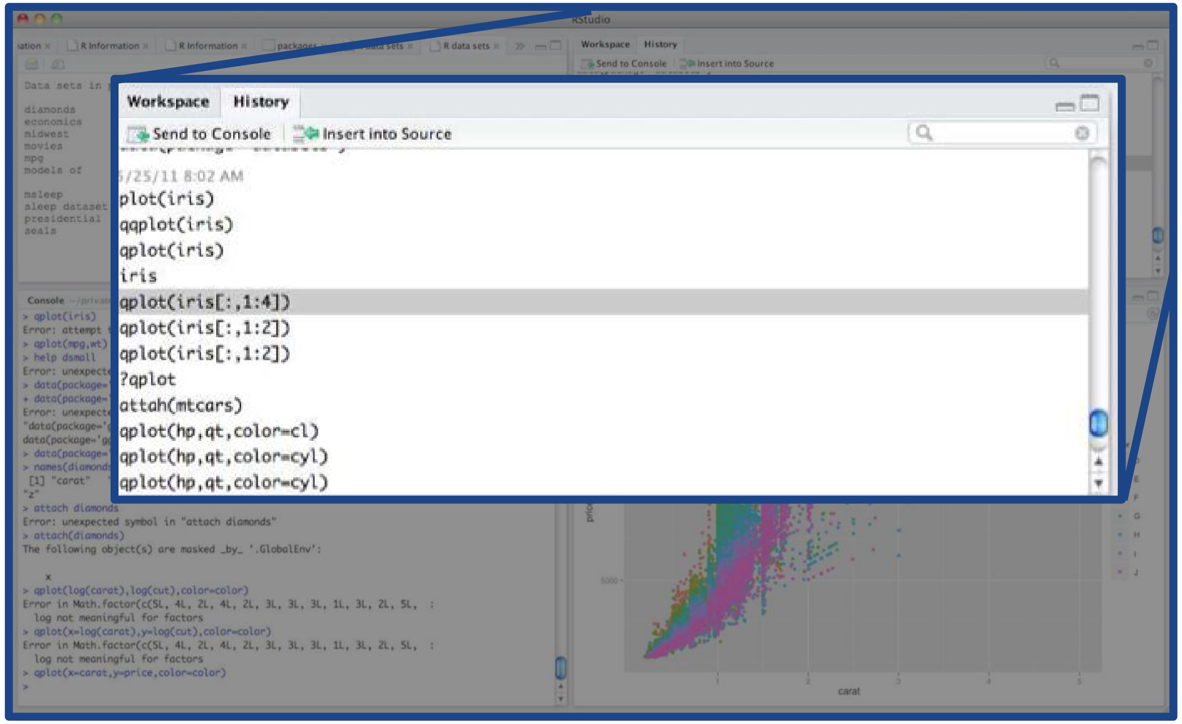
Running R - Interactively



```
Data sets in package 'ggplot2':

diamonds                       Prices of 50,000 round cut diamonds
economics                      US economic time series
midwest                        Demographic information of midwest counties
movies                         Movie information and user ratings from IMDB.com
mpg                            Fuel economy data from 1999 and 2008 for 38 popular
models of
                               car
msleep                         An updated and expanded version of the mammals
sleep dataset
presidential                   Terms of 10 presidents from Eisenhower to Bush W.
seals                          Vector field of seal movements
```

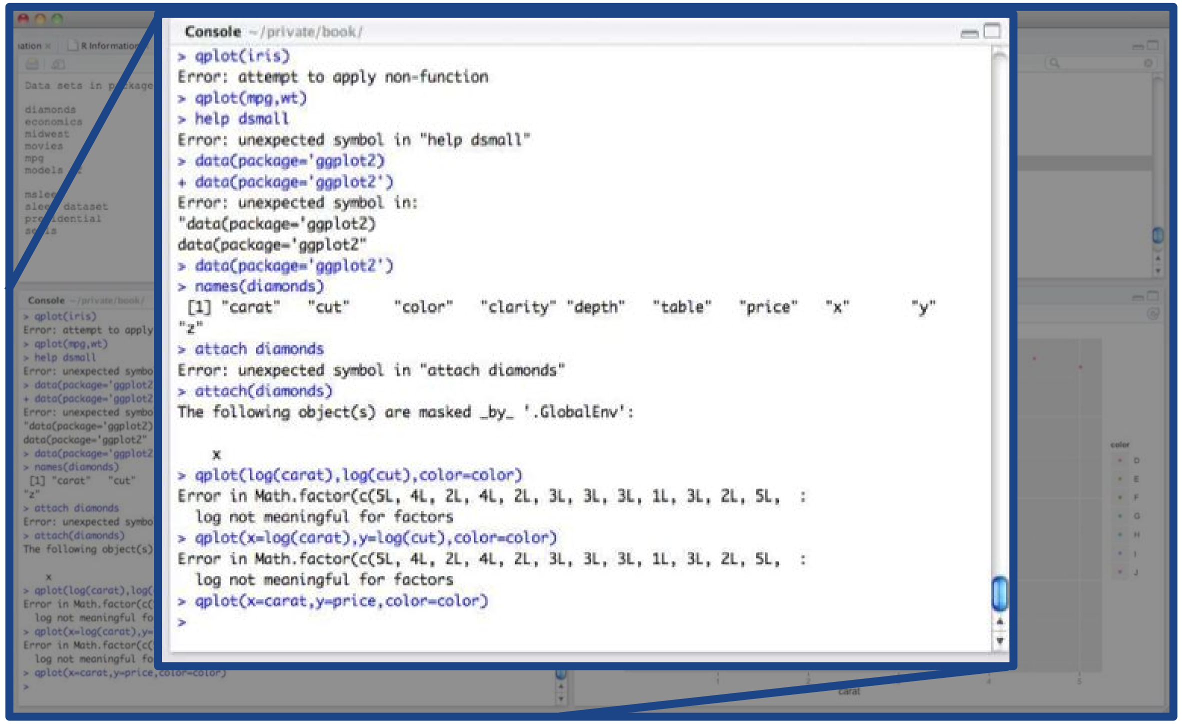
```
> attach diamonds
Error: unexpected symbol in "attach diamonds"
> attach(diamonds)
The following object(s) are masked _by_ '.GlobalEnv':

    x
> qplot(log(carat),log(cut),color=color)
Error in Math.factor(c(5L, 4L, 2L, 4L, 2L, 3L, 3L, 3L, 1L, 3L, 2L, 5L,  :
  log not meaningful for factors
> qplot(x=log(carat),y=log(cut),color=color)
Error in Math.factor(c(5L, 4L, 2L, 4L, 2L, 3L, 3L, 3L, 1L, 3L, 2L, 5L,  :
  log not meaningful for factors
> qplot(x=carat,y=price,color=color)
>
```

Running R - Interactively



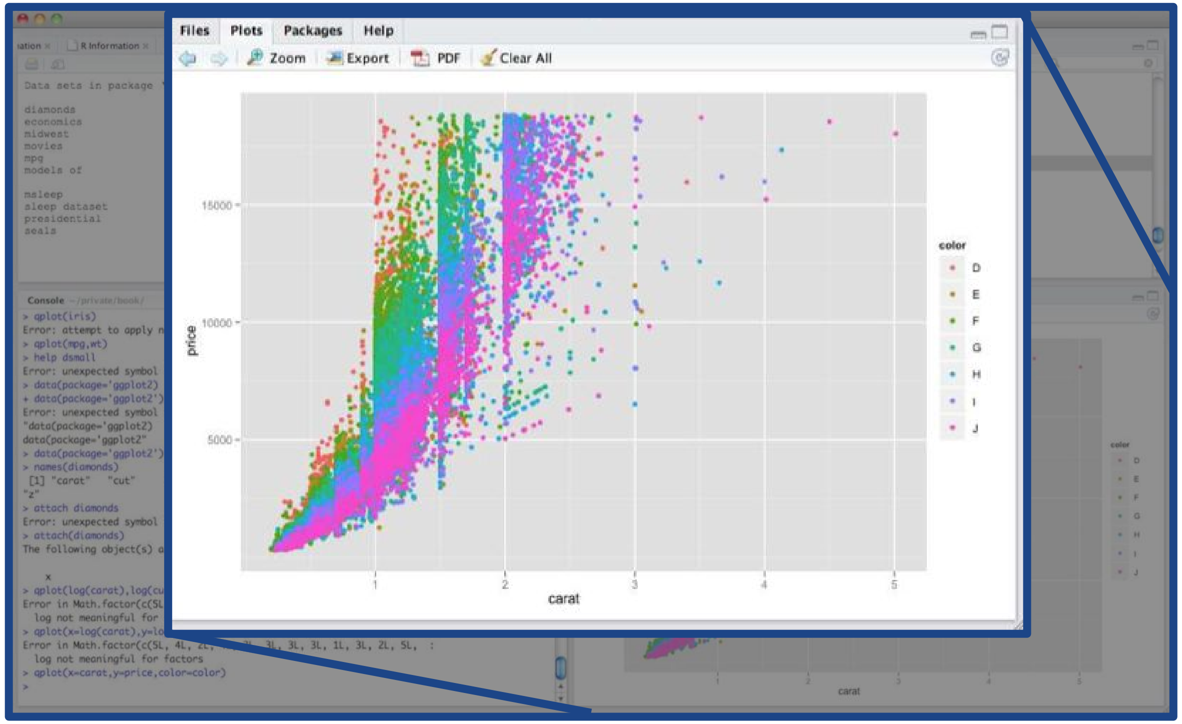
```
5/25/11 8:02 AM
plot(iris)
qqplot(iris)
qplot(iris)
iris
qplot(iris[:,1:4])
qplot(iris[:,1:2])
qplot(iris[:,1:2])
?qplot
attah(mtcars)
qplot(hp,qt,color=cl)
qplot(hp,qt,color=cyl)
qplot(hp,qt,color=cyl)
```

Running R - Interactively



```
> qplot(iris)
Error: attempt to apply non-function
> qplot(mpg,wt)
> help dsmall
Error: unexpected symbol in "help dsmall"
> data(package='ggplot2)
+ data(package='ggplot2')
Error: unexpected symbol in:
"data(package='ggplot2)
data(package='ggplot2"
> data(package='ggplot2')
> names(diamonds)
 [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"   "x"       "y"
"z"
> attach diamonds
Error: unexpected symbol in "attach diamonds"
> attach(diamonds)
The following object(s) are masked _by_ '.GlobalEnv':

    x
> qplot(log(carat),log(cut),color=color)
Error in Math.factor(c(5L, 4L, 2L, 4L, 2L, 3L, 3L, 3L, 1L, 3L, 2L, 5L,  :
  log not meaningful for factors
> qplot(x=log(carat),y=log(cut),color=color)
Error in Math.factor(c(5L, 4L, 2L, 4L, 2L, 3L, 3L, 3L, 1L, 3L, 2L, 5L,  :
  log not meaningful for factors
> qplot(x=carat,y=price,color=color)
>
```

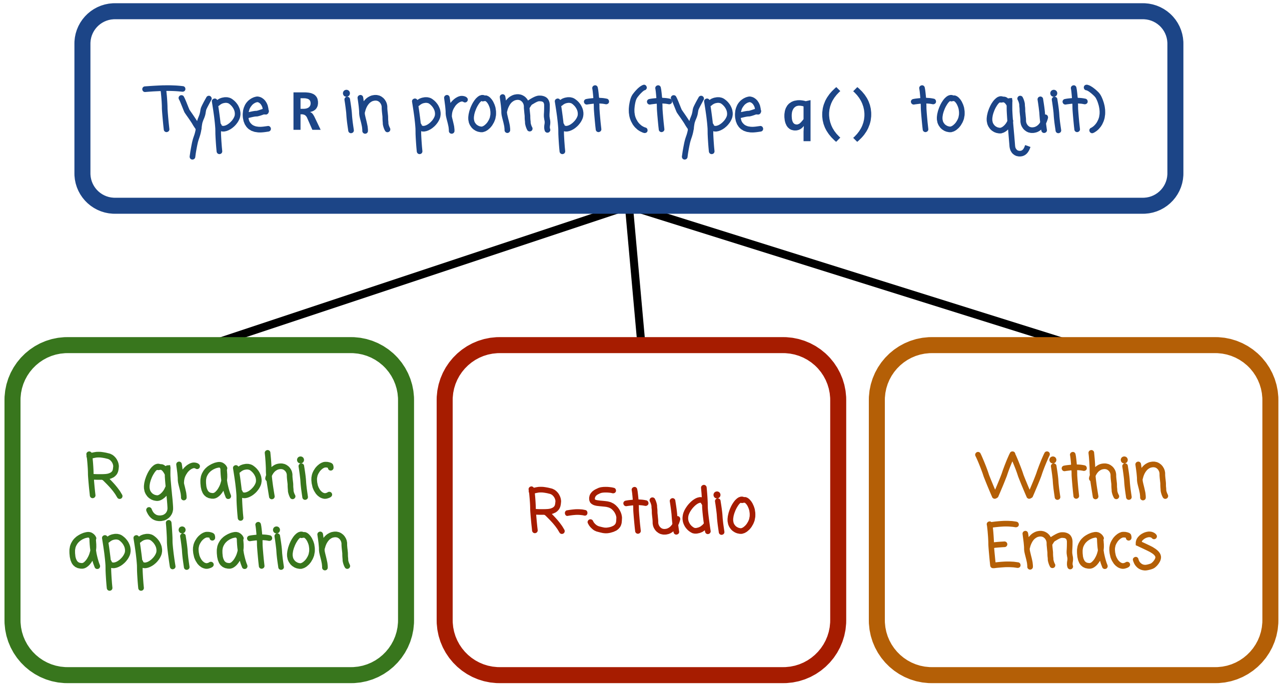
Running R - Interactively



```
Data sets in package '

diamonds
economics
midwest
movies
mpg
models of

msleep
sleep dataset
presidential
seals
```

```
> qplot(iris)
Error: attempt to apply n
> qplot(mpg,wt)
> help dsmall
Error: unexpected symbol
> data(package='ggplot2)
+ data(package='ggplot2')
Error: unexpected symbol
"data(package='ggplot2)
data(package='ggplot2"
> data(package='ggplot2')
> names(diamonds)
 [1] "carat"    "cut"
"z"
> attach diamonds
Error: unexpected symbol
> attach(diamonds)
The following object(s) a

    x
> qplot(log(carat),log(cu
Error in Math.factor(c(5L
  log not meaningful for
> qplot(x=log(carat),y=lo
Error in Math.factor(c(5L, 4L, 2L,          3L, 3L, 3L, 1L, 3L, 2L, 5L,  :
  log not meaningful for factors
> qplot(x=carat,y=price,color=color)
>
```

4 Ways to Run R



Running R - Non-Interactively

- call script from R: `source("foo.R")`
- call script from shell: `R CMD BATCH foo.R`
- call script from shell: `Rscript foo.R`
- executable script, prefixed by `#!/usr/bin/Rscript`, followed by `./foo.R < inFile > outFile`

R Language Quiz

Check all statements that are **true**:

- [✓] white spaces are dropped
- [ ] semicolons are required at the ends of all commands
- [ ] comments are denoted with '%'
- [ ] statically typed

R Language Quiz

Check all statements that are **true**:

- [✓] white spaces are dropped
- [ ] semicolons are required at the ends of all commands

Example of when semicolons are required:

```
a = "a string"; b = 2
```

R Language Quiz

Check all statements that are **true**:

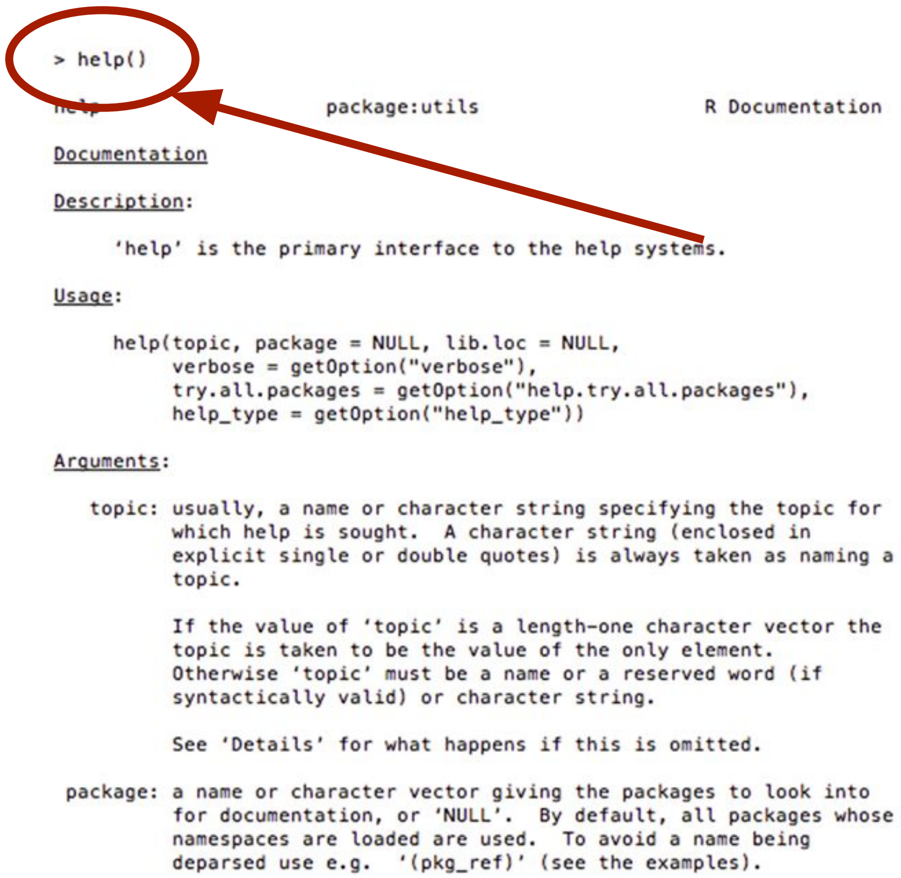
- [✓] white spaces are dropped
- [ ] semicolons are required at the ends of all commands
- [ ] comments are denoted with '%'

Comments are denoted by:

```
# This is a comment
```

R Help Documentation

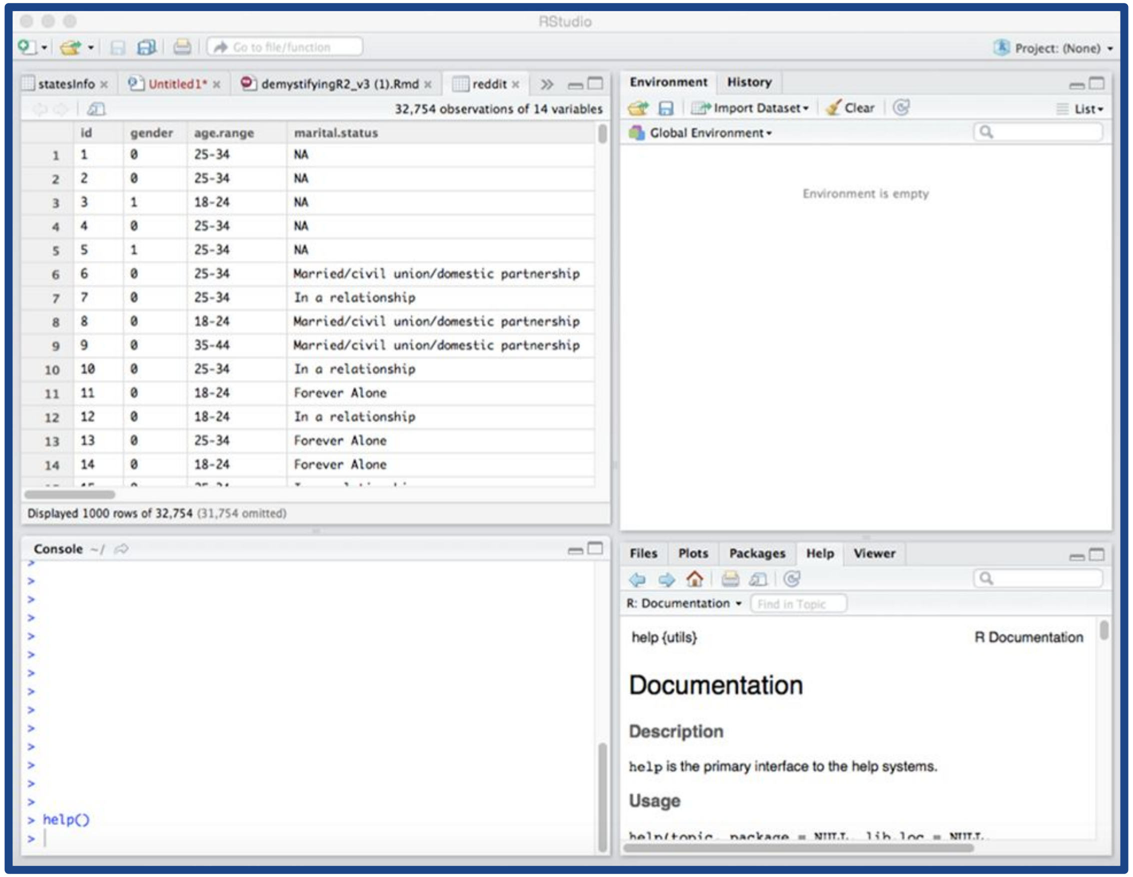
Typing **`help()`** in the terminal:



```
> help()

help                  package:utils                  R Documentation

Documentation

Description:

     'help' is the primary interface to the help systems.

Usage:

     help(topic, package = NULL, lib.loc = NULL,
          verbose = getOption("verbose"),
          try.all.packages = getOption("help.try.all.packages"),
          help_type = getOption("help_type"))

Arguments:

   topic: usually, a name or character string specifying the topic for
          which help is sought.  A character string (enclosed in
          explicit single or double quotes) is always taken as naming a
          topic.

          If the value of 'topic' is a length-one character vector the
          topic is taken to be the value of the only element.
          Otherwise 'topic' must be a name or a reserved word (if
          syntactically valid) or character string.

          See 'Details' for what happens if this is omitted.

 package: a name or character vector giving the packages to look into
          for documentation, or 'NULL'.  By default, all packages whose
          namespaces are loaded are used.  To avoid a name being
          deparsed use e.g.  '(pkg_ref)' (see the examples).
```

R Help Documentation

Typing `help()` in the terminal:



32,754 observations of 14 variables

	id	gender	age.range	marital.status
1	1	0	25-34	NA
2	2	0	25-34	NA
3	3	1	18-24	NA
4	4	0	25-34	NA
5	5	1	25-34	NA
6	6	0	25-34	Married/civil union/domestic partnership
7	7	0	25-34	In a relationship
8	8	0	18-24	Married/civil union/domestic partnership
9	9	0	35-44	Married/civil union/domestic partnership
10	10	0	25-34	In a relationship
11	11	0	18-24	Forever Alone
12	12	0	18-24	In a relationship
13	13	0	25-34	Forever Alone
14	14	0	18-24	Forever Alone

Displayed 1000 rows of 32,754 (31,754 omitted)

```
> help()
>
```

help {utils} R Documentation

Documentation

Description

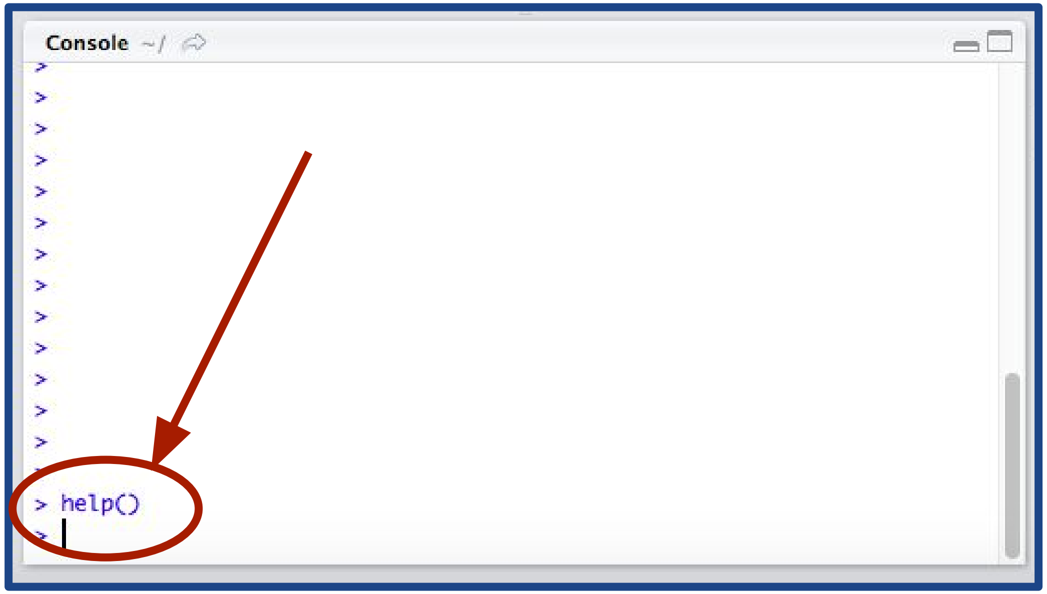
`help` is the primary interface to the help systems.

Usage

```
help(topic, package = NULL, lib.loc = NULL,
```

R Help Documentation

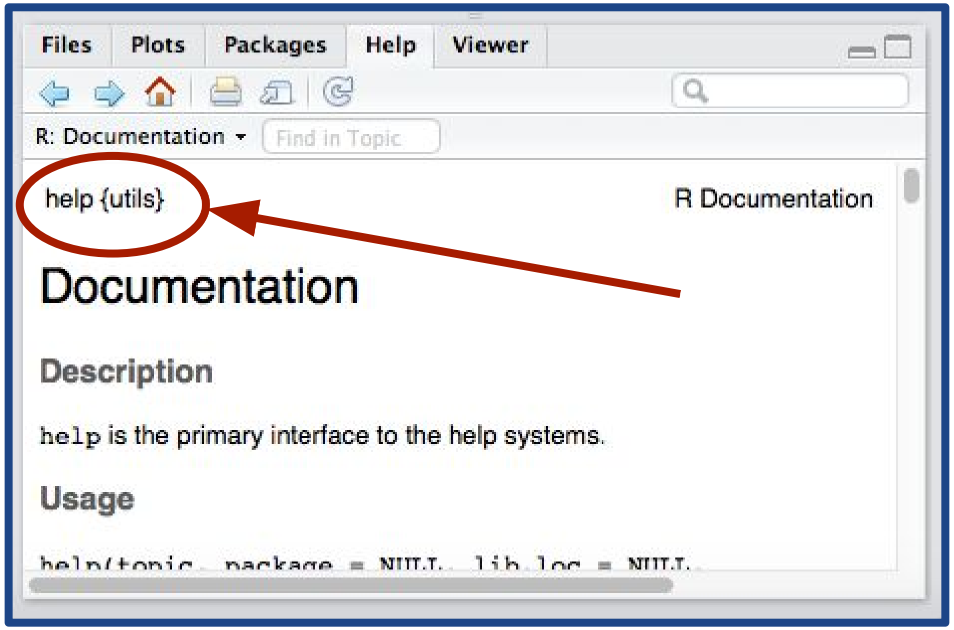
Typing **`help()`** in the terminal:



```
>
>
>
>
>
>
>
>
>
>
>
>
>
> help()
>
```

R Help Documentation

Typing **`help()`** in the terminal:



help {utils} R Documentation

Documentation

Description

`help` is the primary interface to the help systems.

Usage

```
help(topic, package = NULL, lib.loc = NULL,
```

R Help Documentation

To get help on a specific command, type:

```
help(“specific-command”)
```

For example: `help(“load”)`

R Commands

- `ls()` - list variable names in workspace memory

- `save.image(file=”R_workspace”)` - Saving variables to a file
- `save(new.var, legal.var.name, file = “R_workspace”)` - save specified variables

- `load(“R_workspace”)` - load variables saved in a file

R Commands

Environment Commands:

- `install.packages("ggplot2")` - install the ggplot2 package
- `library(ggplot2)` - load the ggplot2 package

System Commands:

- `system("ls -al")` - executes a command in the shell, for example ls -al

Scalars

Major Scalar Types:

Type	Example	Result of Command
numeric	a = 3.2; b = 3	a: num 3.2 b: num 3
integer	c = as.integer(b)	c: int 3
logical	d = TRUE	d: logi TRUE
	e = as.numeric	e: num 1
string	f = “This is a string”	f: chr “This is a string”

Factors

Factors

factors are variables in R which take on a **limited number of different values**

Ordered Factor

```
current.season = factor("summer", levels = c
("summer", "fall", "winter", "spring"), ordered =
TRUE)
```

UnOrdered Factor:

```
my.eye.color = factor("brown", levels = c
("brown", "blue", "green"), ordered = FALSE)
```

Vectors Quiz 1

Fill in the blanks with the outcome of each 'R' command.

Purpose	Example	Outcome
concatenate	x = c(4,3,3,4,3,1)	x = 4 3 3 4 3 1
get length of vector or array	length(x)	length = 6
assign a boolean vector	y = vector(mode = "logical", length = 4)	y = FALSE FALSE FALSE FALSE
assign a numeric vector	z = vector(length = 3, mode = "numeric")	z = 0 0 0

Vectors Quiz 2

Fill in the blanks with the outcome of each 'R' command.

Purpose	Example	Outcome
repeat value multiple times	q = rep(3.2, times = 10)	q = 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2
load values in increments	w = seq(0, 1, by = 0.1)	w = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
load values in equally spaced increments	w = seq(0, 1, length.out = 11)	w = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Comparison Commands Quiz

Fill in the boxes with the result of each example command.

Purpose	Example	Outcome
Boolean vector	w <= 0.5	w = TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE
Checking for true elements	any(w <= 0.5)	TRUE
Checking for all true elements	all(w <= 0.5)	FALSE
Which elements are true	which(w <= 0.5)	1 2 3 4 5 6

Subset Commands Quiz

Fill in the boxes with the result of each example command.

Purpose	Example	Outcome
Extracting entries	w[w <= 0.5]	0.0 0.1 0.2 0.3 0.4 0.5
Subset function	subset(w, w <= 0.5)	0.0 0.1 0.2 0.3 0.4 0.5
Zero out components	w[w <= 0.5] = 0	w = 0.0 0.0 0.0 0.0 0.0 0.0 0.6 0.7 0.8 0.9 1.0

Creating Arrays Quiz

```
z = seq(1, 20,length.out = 20)
x = array(data = z, dim = c(4, 5))
```

Fill in the boxes with the values stored in the array.

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	5	9	13	17
[2,]	2	6	10	14	18
[3,]	3	7	11	15	19
[4,]	4	8	12	16	20

Reading Arrays Quiz

Given the following array, fill in the blanks with the results of each command.

```
      [,1] [,2] [,3] [,4] [,5]
[,1]     1    5    9   13   17
[,2]     2    6   10   14   18
[,3]     3    7   11   15   19
[,4]     4    8   12   16   20
```

```
x[2,3] = 10
x[2,] =  2  6 10 14 18
```

`x[-1,] =`

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	2	6	10	14	18
[2,]	3	7	11	15	19
[3,]	4	8	12	16	20

`y = x[c(1,2), c(1,2)]`

	[,1]	[,2]
[1,]	1	5
[2,]	2	6

Manipulating Arrays Quiz

Given the following array, **determine the outcomes** of the following commands.

```
y =        [,1] [,2]
     [,1]     1    5
     [,2]     2    6
```

`2 * y + 1`

	[,1]	[,2]
[1,]	3	11
[2,]	5	13

`y %*% y`

	[,1]	[,2]
[1,]	11	35
[2,]	14	46

Inner Product and Transpose Quiz

Given the array 'x', **determine the outcome** of the following commands.

```
      [,1] [,2] [,3] [,4] [,5]
[,1]     1    5    9   13   17
[,2]     2    6   10   14   18
[,3]     3    7   11   15   19
[,4]     4    8   12   16   20
```

`x[1,] %*% x[1,]`

	[,1]
[1,]	565

`t(x)`

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	2	3	4	
[2,]	5	6	7	8	
[3,]	9	10	11	12	
[4,]	13	14	15	16	
[5,]	17	18	19	20	

Outer Product Quiz

Given the array 'x', **determine the outcome** of the following commands.

```
outer(x[,1], x[,1])
```

```
      [,1] [,2] [,3] [,4] [,5]
[,1]     1    5    9   13   17
[,2]     2    6   10   14   18
[,3]     3    7   11   15   19
[,4]     4    8   12   16   20
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	2	3	4	
[2,]	2	4	6	8	
[3,]	3	6	9	12	
[4,]	4	8	12	16	
[5,]					

Concatenation Quiz

Determine the outcome of the following commands.

```
rbind(x[1,], x[1,])
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	5	9	13	17
[2,]	1	5	9	13	17

```
cbind(x[1,], x[1,])
```

	[,1]	[,2]
[1,]	1	1
[2,]	5	5
[3,]	9	9
[4,]	13	13
[5,]	17	17

Lists Quiz

Given the following list command, **fill in the blanks** with the result of each command.

```
L=list(name = 'John', age = 55, no.children = 2, children.
ages = c(15, 18))
```

`names(L)`	name age no.children children.ages
`L[[2]]`	55
`L$name`	John
`L['name']`	John
`L$children.ages[2]`	18
`L[[4]][2]`	18

Dataframes Quiz

Assume the following commands have been executed, **fill in the blanks** with the corresponding outputs

```
vecn = c("John Smith","Jane Doe")
veca = c(42, 45)
vecs = c(50000, 55000)
R = data.frame(name = vecn, age = veca, salary = vecs)
```

R

	name	age	salary
1	John Smith	42	50000
2	Jane Doe	45	55000

Dataframes Modification Quiz

Given the following dataframe called 'R', **fill in the blanks** to reflect the changes made by the command:

```
names(R) = c("NAME", "AGE", "SALARY")
```

	name	age	salary
1	John Smith	42	50000
2	Jane Doe	45	55000

	NAME	AGE	SALARY
1	John Smith	42	50000
2	Jane Doe	45	55000

Datasets Quiz 1

Write the 'R' command that will perform the listed task

Task	Command
List the dimension (column) names	names(iris)
Show the first four rows	head(iris,4)
Show the first row	iris[1]
Sepal length of the first 10 samples	iris$Sepal.Length[1:10]
Allow replacing iris$Sepal.Length with shorter Sepal.Length	attach(iris, warn.conflicts = FALSE)

Datasets Quiz 2

Write the 'R' command that will perform the listed task

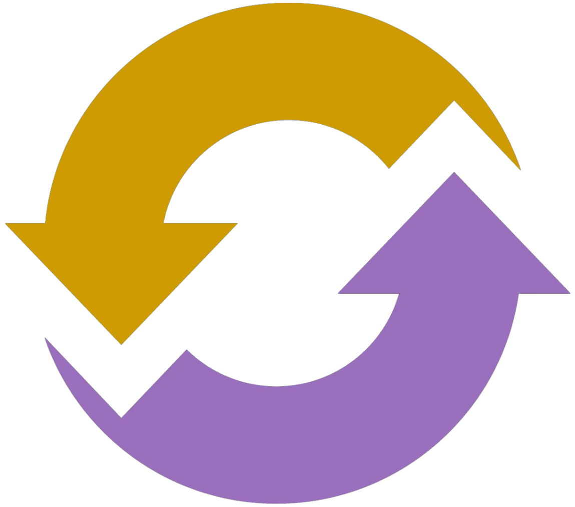
Task	Command
Average of Sepal.Length across all rows	mean(Sepal.Length)
Means of all four numeric columns	colMeans(iris[,1:4])
Create a subset of sepal lengths less than 5 in the setosa species	subset(iris, Sepal.Length < 5 & Species == "setosa")
number of rows corresponding to setosa species	dim(subset(iris, Species == "setosa"))[1]
summary of the dataset iris	summary(iris)

If-Else

```
a = 10; b = 5; c = 1
if (a < b) {
     d = 1
} else if (a == b) {
     d = 2
} else {
     d = 3
}
print(d)
```

If-Else

AND: &&, OR: ||, equality: ==, inequality: !=



Loops Quiz

Use a 'for; loop to write an 'R' program that adds the numbers (num) 1 to 100 and stores it in a variable called 'sum'

```
sum=0
```

```
# repeat for 100 iteration, with num taking values 1:100

for (num in seq(1, 100, by = 1)) {
      sum = sum + num
}
```

Repeat Loops Quiz

Using a repeat loop, write an 'R' program that subtracts the numbers (num) 100 to 1 from a variable called sum. If the sum becomes '0' or less, exit the repeat loop. Use a variable called 'num' for the numbers, and 'sum' for the sum.

```
sum = 5050
```

```
repeat {
    sum = sum - num
    num = num - 1
    if (sm == 0) break
}
```

While Loops Quiz

Given two variables (a, b) and a sum = 0, **write a while loop to perform the following task:** While b > a, increment the variables sum and 'a', and decrement the variable 'b'.

```
a = 1; b = 10
```

```
while (b>a) {
    sm = sm + 1
    a = a + 1
    b = b - 1
}
```

Functions Quiz

The given function is expecting variables to be in the order x,y,z.
Fill in the blanks to call the function for each situation.

```
Assume x=10, y=20, z=30
```

Call foo with the variables in x,y,z, order	foo(10,20,30)
Call foo with the variables in y,x,z order	foo(y=20, x=10, z= 30)
Call foo with the variables x and y set to default, z = 30	foo(z = 30)

Functions

```
myPower = function(
     bas = 10, pow = 2) {
     res = bas^pow
     return(res)
}
```

Functions

```
myPower(2, 3)

myPower(pow = 3, bas = 2)

myPower(bas = 3)
```

Vectorized Code

```
a = 1:10000000; res = 0
system.time(for (e in a) res = res + e^2)

## user system elapsed
## 3.742 0.029 3.800
```

Vectorized Code

```
system.time(sum(a^2))

## user system elapsed
## 0.180 0.032 0.250
```

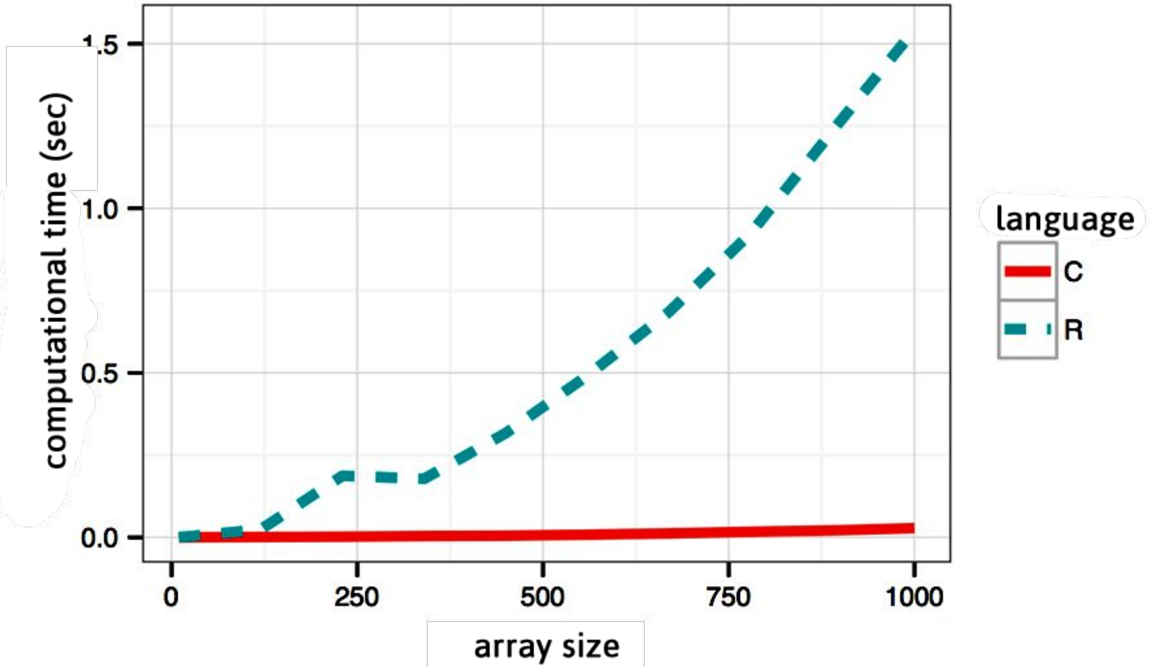
External/Native API

```
dyn.load("fooC2.so") # load compiled C code

A = seq(0, 1, length = 10)
B = seq(0, 1, length = 10)
.Call("fooC2", A, B)
```

Newer packages: Rcpp, RcppArmadillo, RcppEigen

```
##   [1] 13.34 17.48 21.21 24.71 28.03 31.24 34.34 37.37 40.33 43
##   [1] 13.34 17.48 21.21 24.71 28.03 31.24 34.34 37.37 40.33 43
```

External/
Native API


computational time (sec)
array size
language
C
R