



Praktikum 2

Kecerdasan Komputasional

Writing your own R functions

```
sum.of.squares <- function(x,y) {  
  x^2 + y^2  
}
```

You have now created a function called `sum.of.squares` which requires two arguments and returns the sum of the squares of these arguments.

Since you ran the code through the console, the function is now available, like any of the other built-in functions within R.

Running `sum.of.squares(3,4)` will give you the answer 25.

C:/Users/medri_yzv794q/Desktop/mmm,mv r/CI/R/R.MEDRIA - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

ABC.R

```
1 fooABC <- function(x) {  
2   k <- x+1  
3   return(k)  
4 }
```

Environment History

Global Environment

Functions

fooABC function (x)

Files Plots Packages Help Viewer

New Folder Delete Rename More

C:/Users/medri_yzv794q/Desktop/mmm,mv r/CI/R/R.MEDRIA

Name	Size	Modified
..		
R.MEDRIA.Rproj	218 B	Feb 17, 2017, 1:14 PM
ABC.R	51 B	Feb 17, 2017, 1:15 PM

4.2 (Top Level) R Script

Console C:/Users/medri_yzv794q/Desktop/mmm,mv r/CI/R/R.MEDRIA/

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.

```
> source("ABC.R")  
> fooABC(5)  
[1] 6  
>
```

Review: R as a Calculator

```
> 5 + 4
```

```
[1] 9
```

or various calculations in the same row

```
> 2+3; 5*9; 6-6
```

```
[1] 5
```

```
[1] 45
```

```
[1] 0
```

Operation Symbols

Symbol	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%%	Modulo (estimates remainder in a division)
^	Exponential

Assignment

"<-" used to indicate assignment

```
x<-c(1,2,3,4,5,6,7)
```

```
x<-c(1:7)
```

```
x<-1:4
```

Review: R as a Calculator

Mathematical problem :

$x=3, y=4$

a. $x+y$

b. $x-y$

c. 5 modulo 3

d. $2y+2x^2$

How to write this in R language?



Exercise 1

Create a function that will return the sum of 2 integers.


Solution – Exercise 1

```
f.sum <- function (x, y) {  
  r <- x + y  
  r  
}
```

```
f.sum(2, 3)
```

R function for testing if a vector contains a given element


```
v <- c('a', 'b', 'c', 'e')  
'b' %in% v ## returns TRUE  
match('b', v) ## returns the first location of 'b'
```

- 
- Mathematical Problem : vector $a = \{1, 2, 3, 4, 5\}$
 - Check whether 4 is in a
 - Check whether 6 is in a



Exercise 2

Create a function that will return TRUE if a given integer is inside a vector.



```
f.exist<-function(v,a){  
  a %in% v  
}
```



Data Frames

- Another way that information is stored is in data frames. This is a way to take many vectors of different types and store them in the same variable. The vectors can be of all different types. For example, a data frame may contain many lists, and each list might be a list of factors, strings, or

Example of a data frame

```
a <- c(1,2,3,4)
```

```
b <- c(2,4,6,8)
```

```
levels <- factor(c("A","B","A","B"))
```

```
bubba <- data.frame(first=a, second=b, f=levels)
```



>bubba

	first	second	f
--	-------	--------	---

1	1	2	A
---	---	---	---

2	2	4	B
---	---	---	---

3	3	6	A
---	---	---	---

4	4	8	B
---	---	---	---

How to Extract Rows & Columns from Data Frame

```
> df <- data.frame( c( 183, 85, 40), c( 175, 76, 35), c( 178, 79, 38 ))
```

```
> names(df) <- c("Height", "Weight", "Age")
```

```
> df
```

	Height	Weight	Age
--	--------	--------	-----

1	183	175	178
---	-----	-----	-----

2	85	76	79
---	----	----	----

3	40	35	38
---	----	----	----



Exercise

All Rows and All Columns

```
df[,]
```

First row and all columns

```
df[1,]
```

First two rows and all columns

```
df[1:2,]
```

First and third row and all columns

```
df[ c(1,3), ]
```

Exercise

First Row and 2nd and third column

?

First, Second Row and Second and Third Column

?

Just First Column with All rows

?

First and Third Column with All rows

?



First Row and 2nd and third column

`df[1, 2:3]`

First, Second Row and Second and Third COlumn

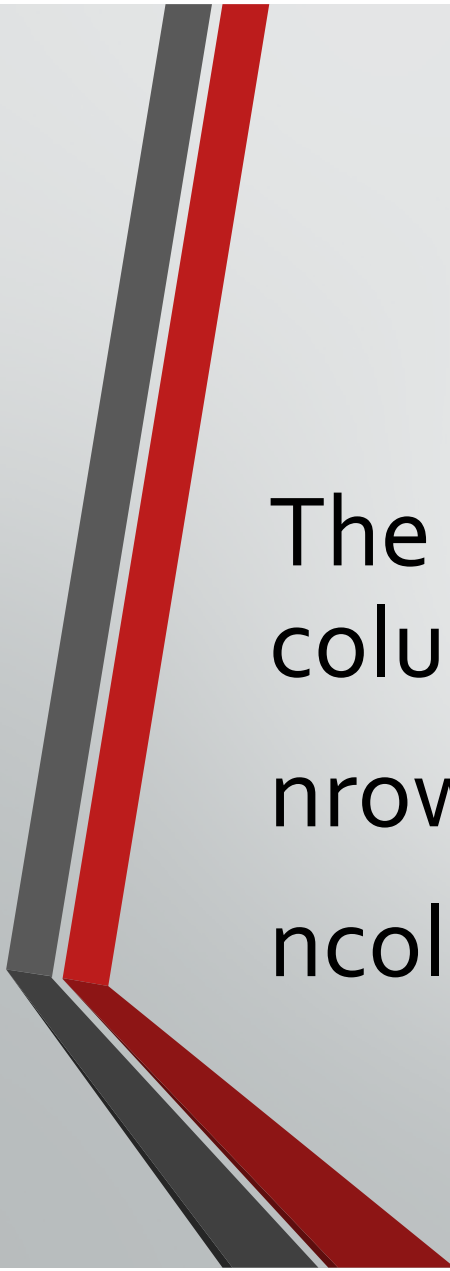
`df[1:2, 2:3]`

Just First Column with All rows

`df[, 1]`

First and Third Column with All rows

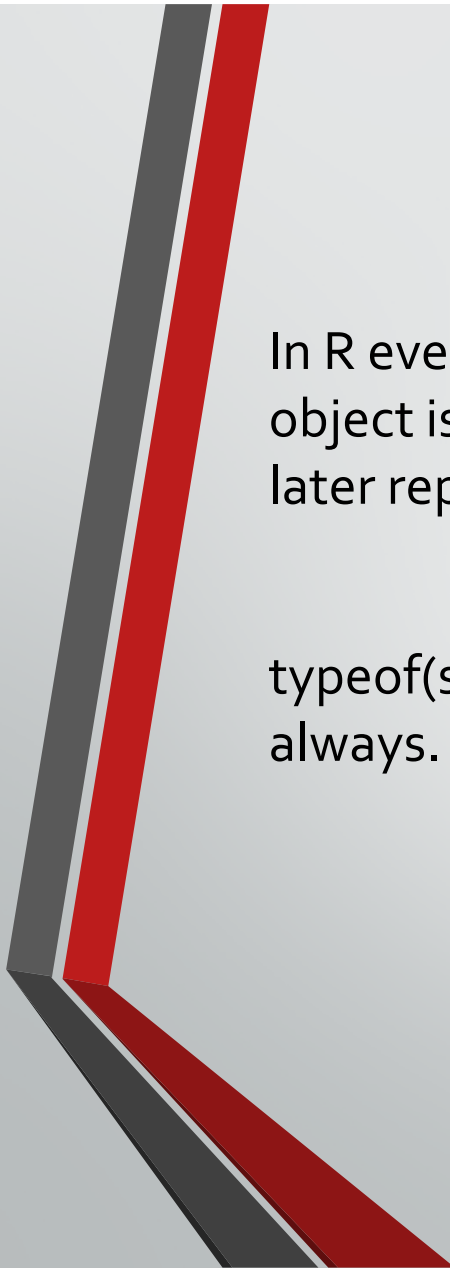
`df[,c(1,3)]`



The number of data rows and data
column :

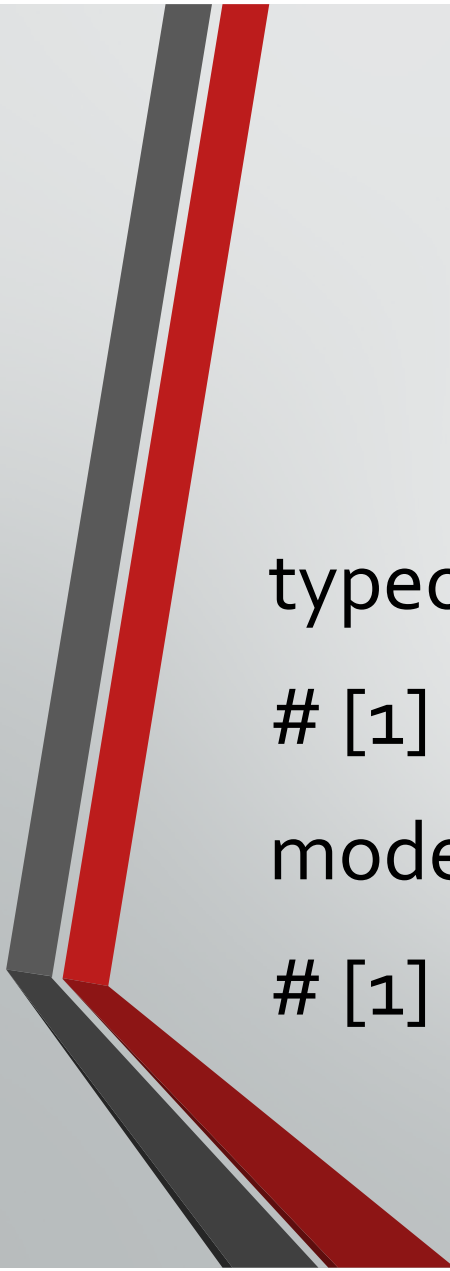
`nrow(df)`

`ncol(df)`



In R every "object" has a **mode** and a **class**. The former represents how an object is stored in memory (numeric, character, list and function) while the later represents its abstract type.

`typeof(storage.mode)` will usually give the same information as mode but not always. e



```
typeof(c(1,2))
```


```
# [1] "double"
```

```
mode(c(1,2))
```

```
# [1] "numeric"
```

`names(df[1])`

- How can we get the heading of each column?



```
names(df[1])
```

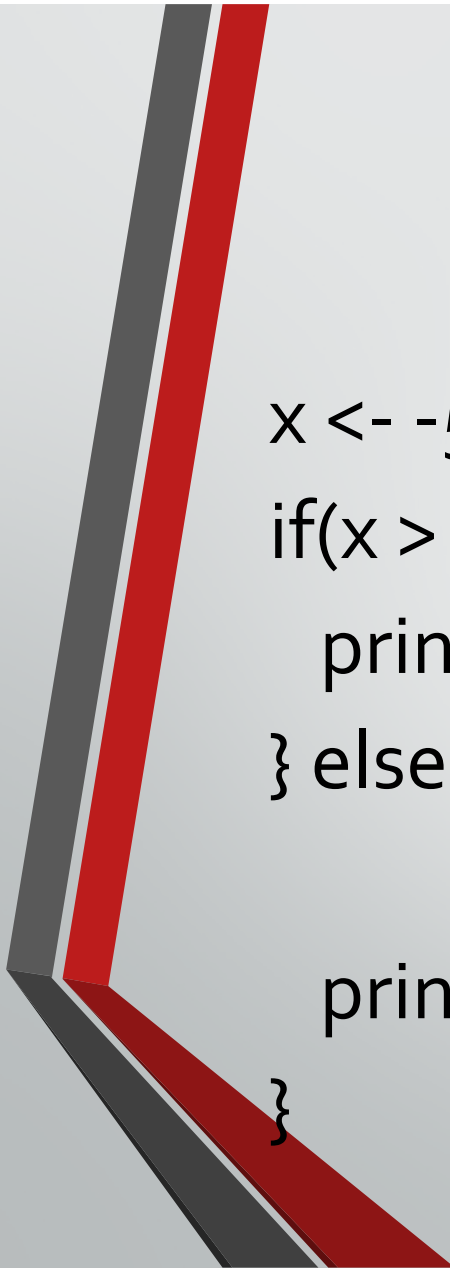
```
names(df[2])
```

```
names(df[3])
```

R if...else Statement

The syntax of if statement is:

```
if (test_expression) {  
  statement  
}
```



```
x <- -5  
if(x > 0){  
  print("Non-negative number")  
} else {  
  
  print("Negative number")  
}
```

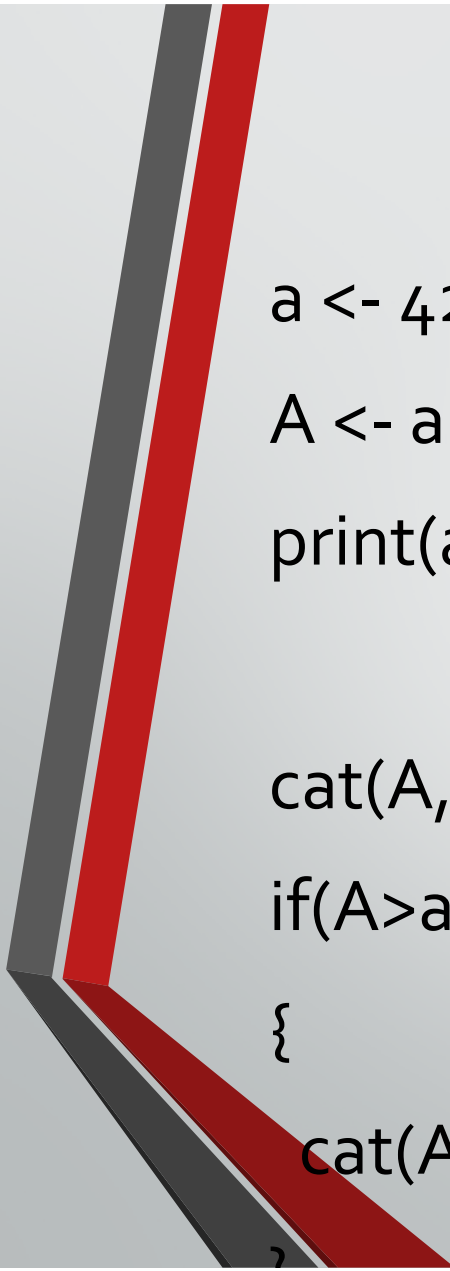
Example of nested if...else

```
x <- 0  
if (x < 0) {  
  print("Negative number")  
} else if (x > 0) {  
  print("Positive number")  
} else  
  print("Zero")
```



Print and Cat

- Print automatically appends a new line character to the output.
- With Cat you have to append it manually



```
a <- 42
```

```
A <- a * 2 # R is case sensitive
```

```
print(a)
```

```
cat(A, "\n") # "84" is concatenated with "\n"
```

```
if(A>a) # true, 84 > 42
```

```
{
```

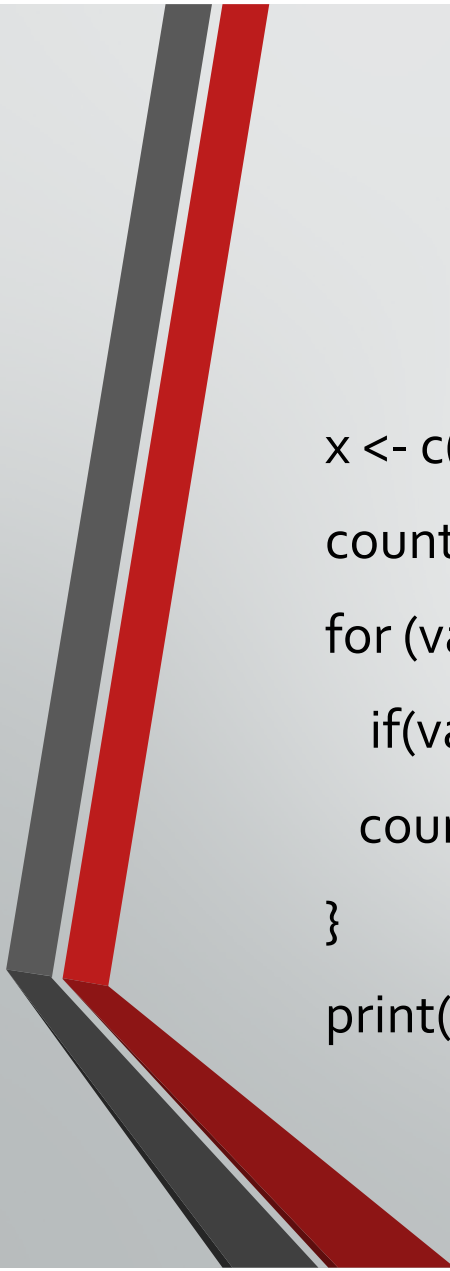
```
  cat(A, ">", a, "\n")
```

For Loop in R

Syntax of for loop

```
for (val in sequence)
{
  statement
}
```

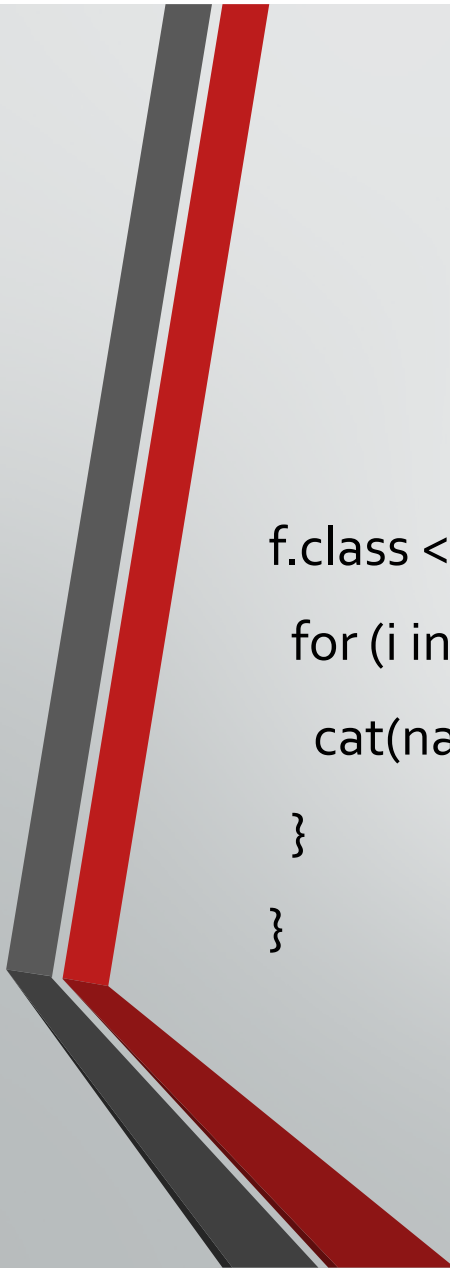
- 
- Example to count the number of even numbers in a vector!



```
x <- c(2,5,3,9,8,11,6)
count <- 0
for (val in x) {
  if(val %% 2 == 0)
    count = count+1
}
print(count)
```

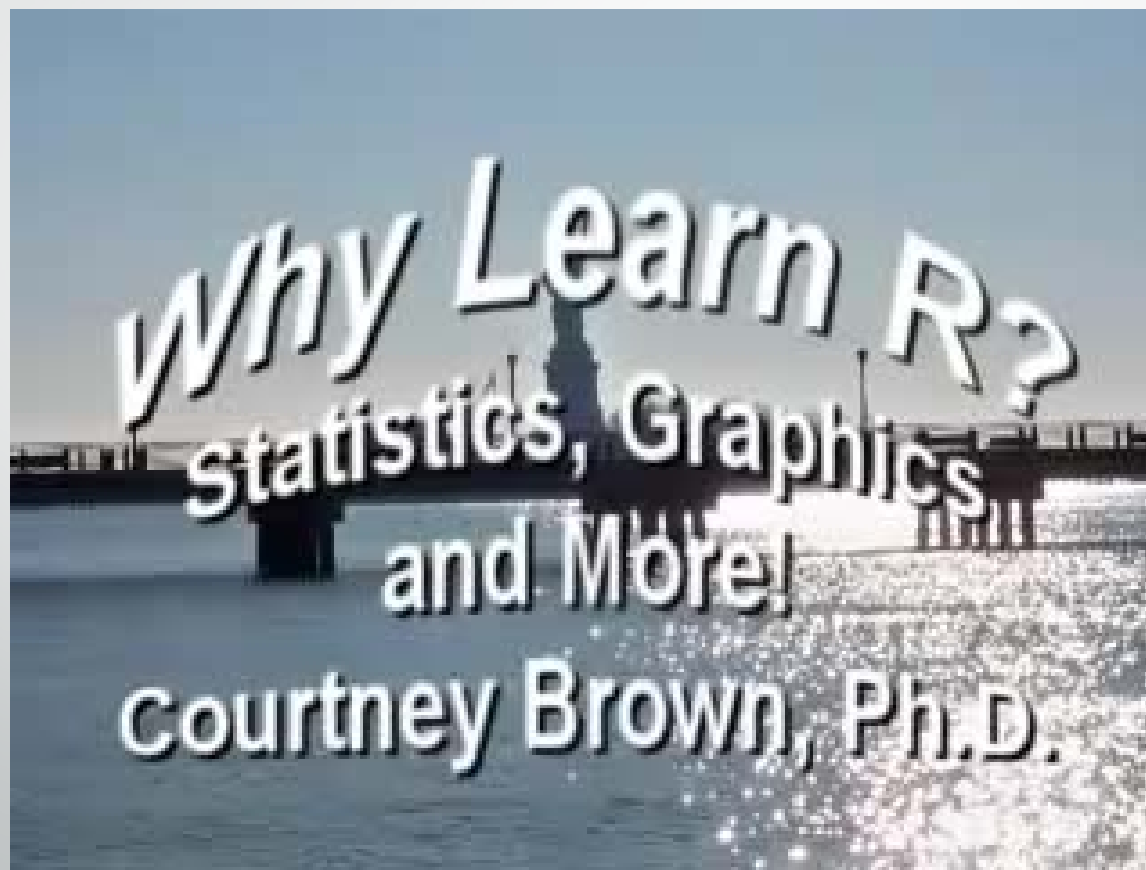
Exercise 3

- Create a function that given a data frame will print by screen the name of the column and the class of data it contains (e.g. Variable1 is Numeric).



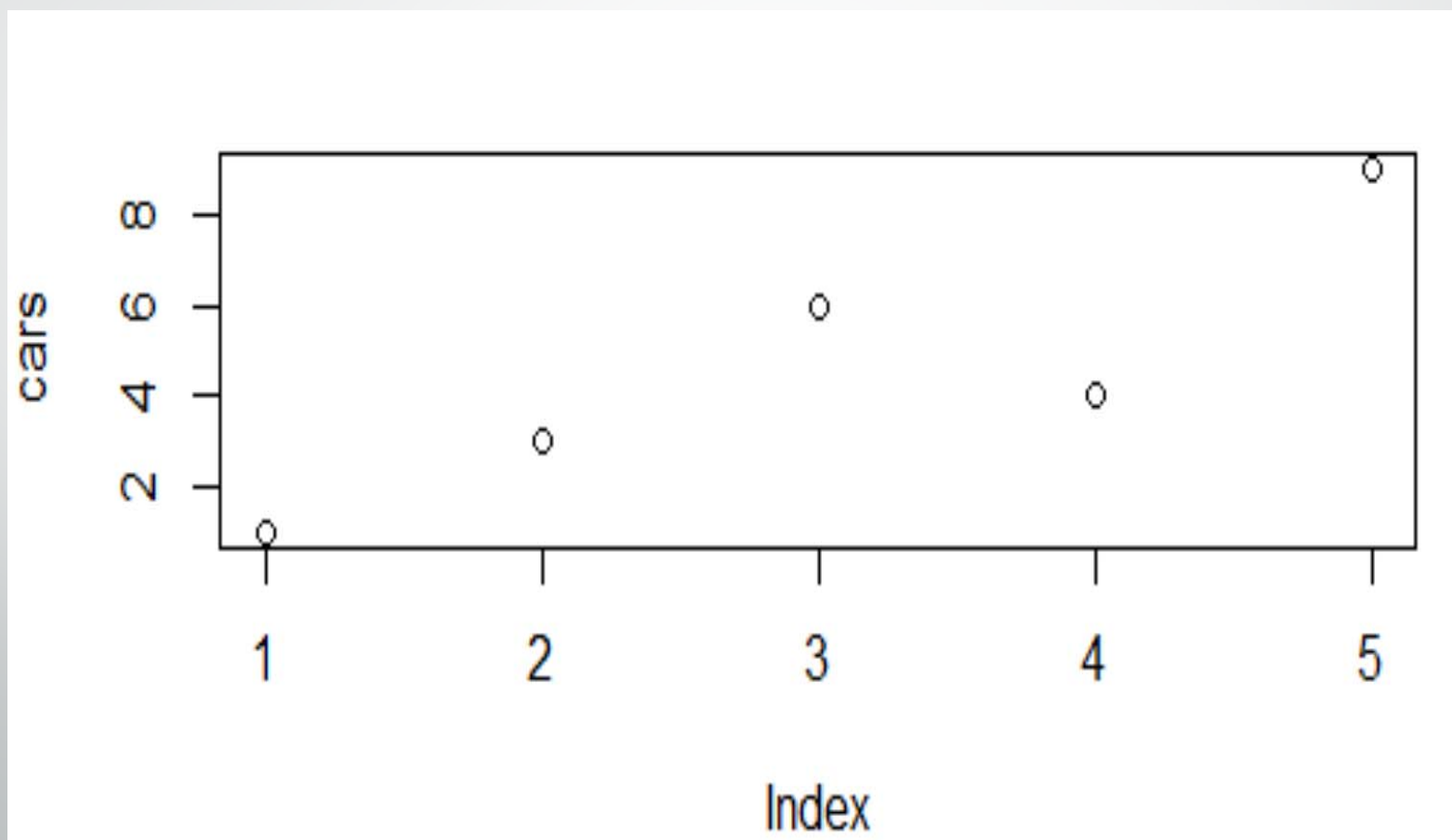
```
f.class <- function (df) {  
  for (i in 1:ncol(df)) {  
    cat(names(df)[i], "is", class(df[, i]), "\n")  
  }  
}
```

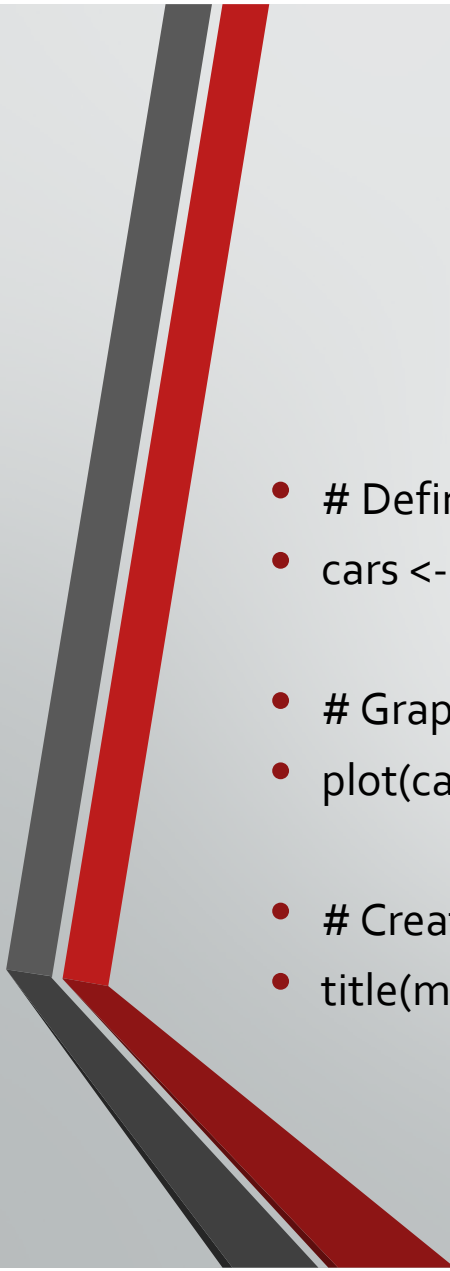
Reasons to learn the R Programming Language



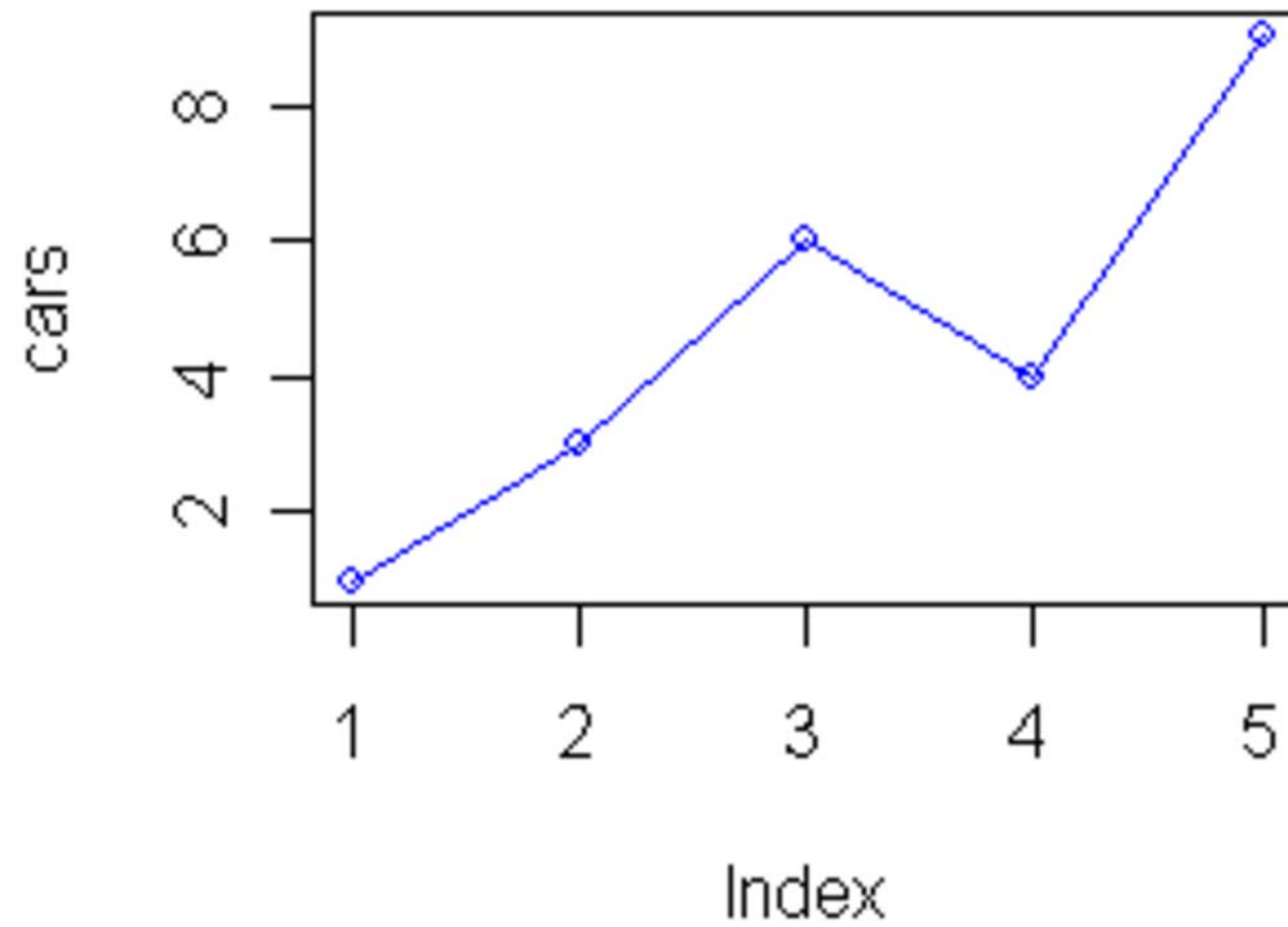
Producing Simple Graphs with R

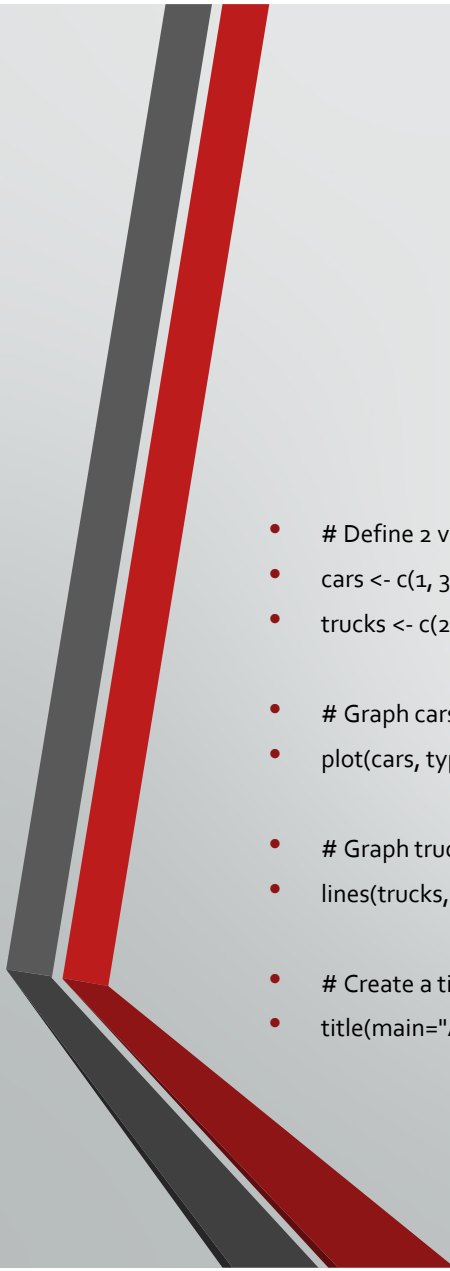
- `# Define the cars vector with 5 values`
- `cars <- c(1, 3, 6, 4, 9)`
- `# Graph the cars vector with all defaults`
- `plot(cars)`



- 
- `# Define the cars vector with 5 values`
 - `cars <- c(1, 3, 6, 4, 9)`
 - `# Graph cars using blue points overlayed by a line`
 - `plot(cars, type="o", col="blue")`
 - `# Create a title with a red, bold/italic font`
 - `title(main="Autos", col.main="red", font.main=4)`

Autos



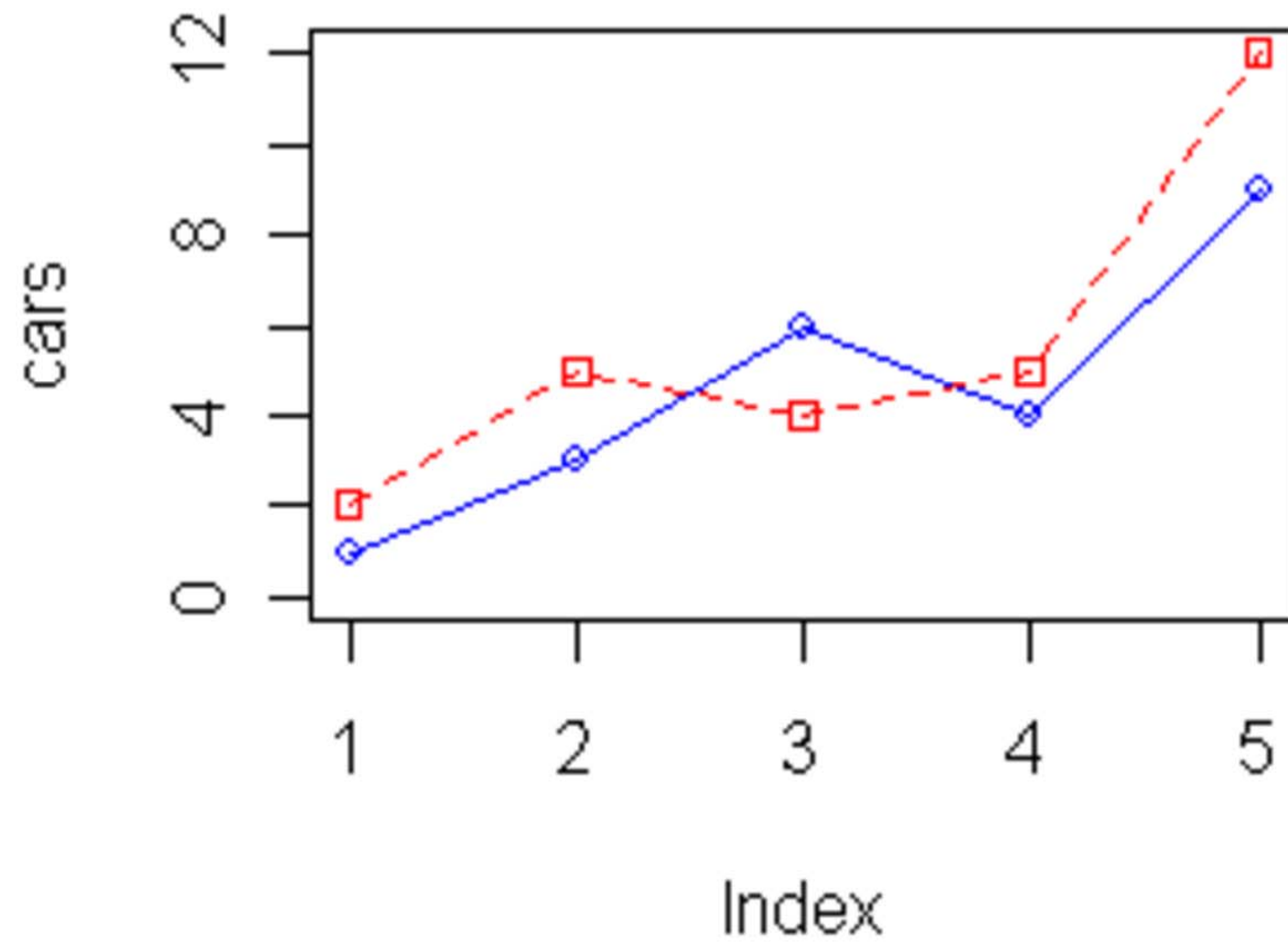
- 
- `# Define 2 vectors`
 - `cars <- c(1, 3, 6, 4, 9)`
 - `trucks <- c(2, 5, 4, 5, 12)`


 - `# Graph cars using a y axis that ranges from 0 to 12`
 - `plot(cars, type="o", col="blue", ylim=c(0,12))`

 - `# Graph trucks with red dashed line and square points`
 - `lines(trucks, type="o", pch=22, lty=2, col="red")`

 - `# Create a title with a red, bold/italic font`
 - `title(main="Autos", col.main="red", font.main=4)`

Autos





References

- Various Sources.



Thank you!