R Programming

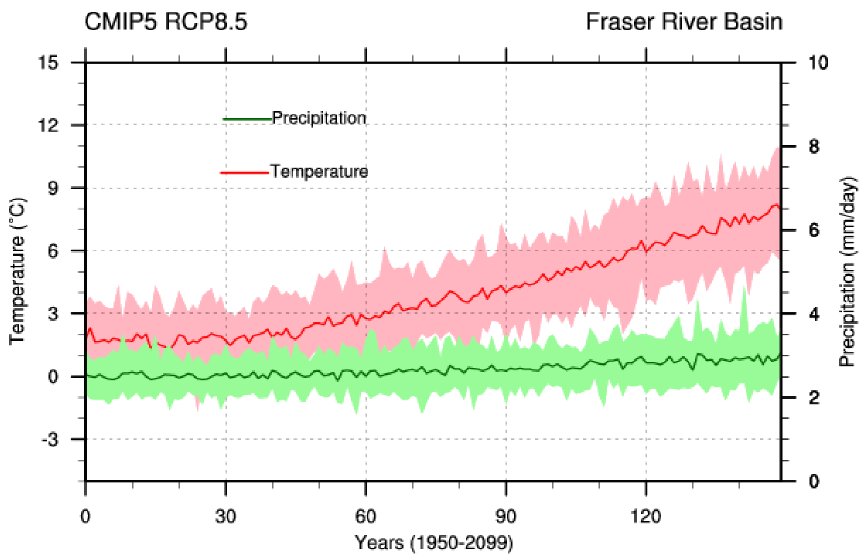
Lecture 2

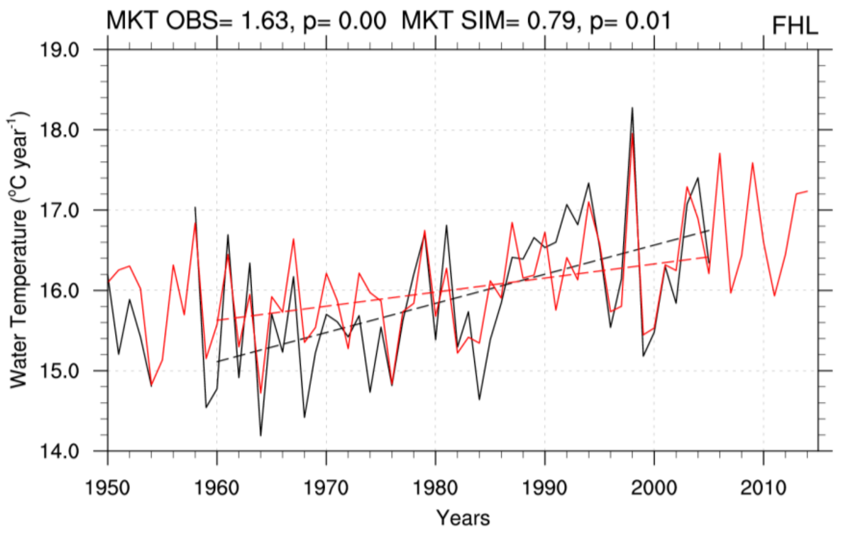
- Objects in R
- Arithmetic functions
- Create and combine vectors
- Extract elements from vectors
- Vectors arithmetic

How to Present Scientific Data Analysis

There are many different ways to present scientific data analysis depending on the nature of the data and the goal of analysis e.g.

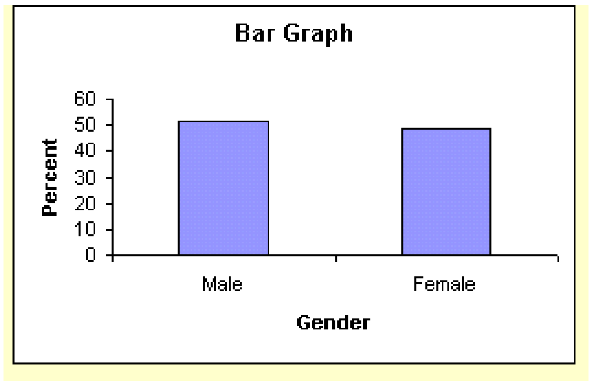
Line graph - A line graph is particularly useful when we want to show the trend of a variable over time. Time is displayed on the horizontal axis (x-axis) and the variable is displayed on the vertical axis (y- axis).





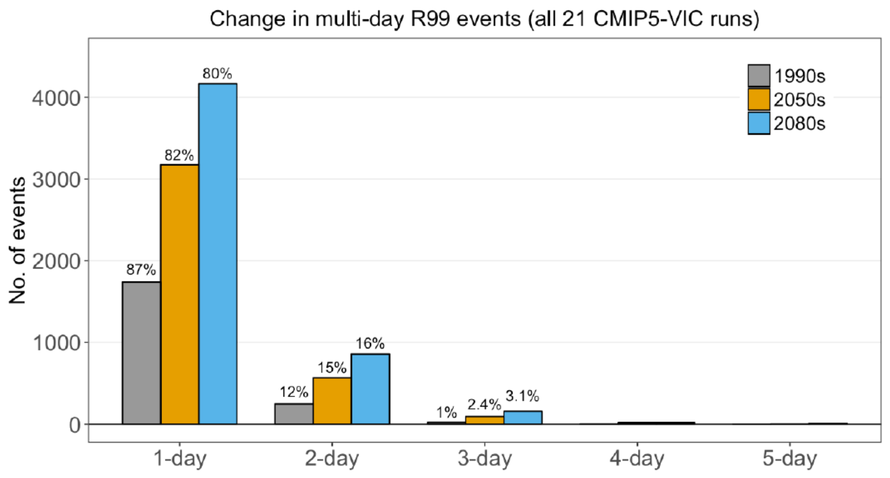
Bar Graph

A bar graph is a way of summarizing a set of categorical data. It displays the data using a number of rectangles, of the same width, each of which represents a particular category. Bar graphs can be displayed horizontally or vertically and they are usually drawn with a gap between the bars (rectangles).



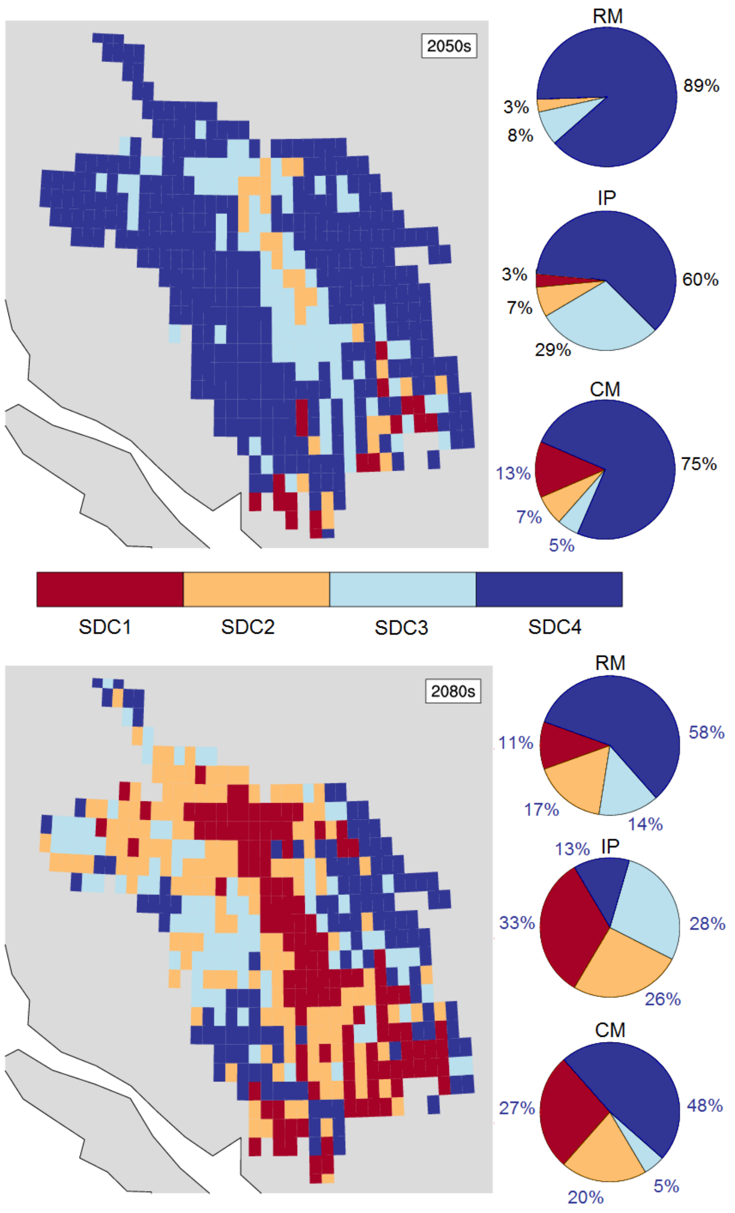
Histogram

A histogram is a way of summarizing data that are measured on an interval scale (either discrete or continuous). It is often used in exploratory data analysis to illustrate the features of the distribution of the data in a convenient form.



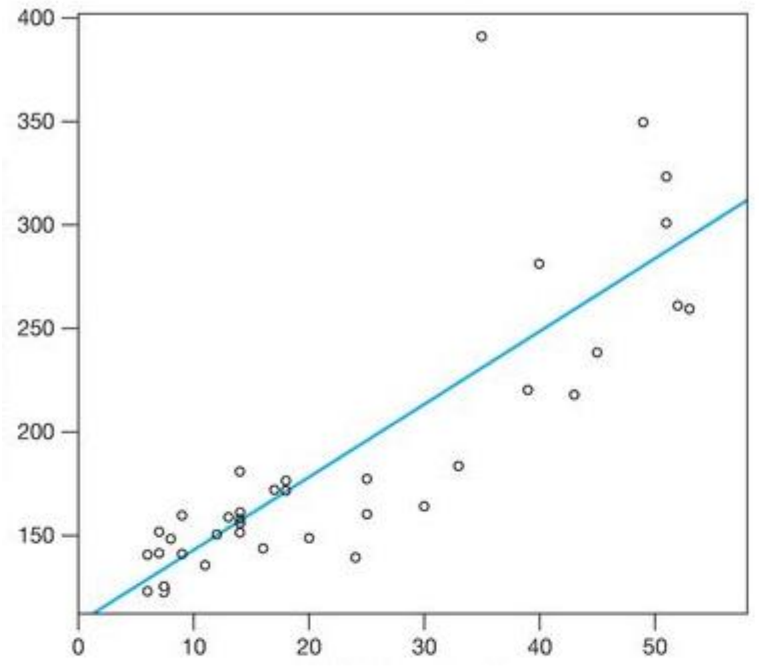
Pie Chart

A pie chart is used to display a set of categorical data. It is a circle, which is divided into segments. Each segment represents a particular category. The area of each segment is proportional to the number of cases in that category.



Scatter Plots

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation .



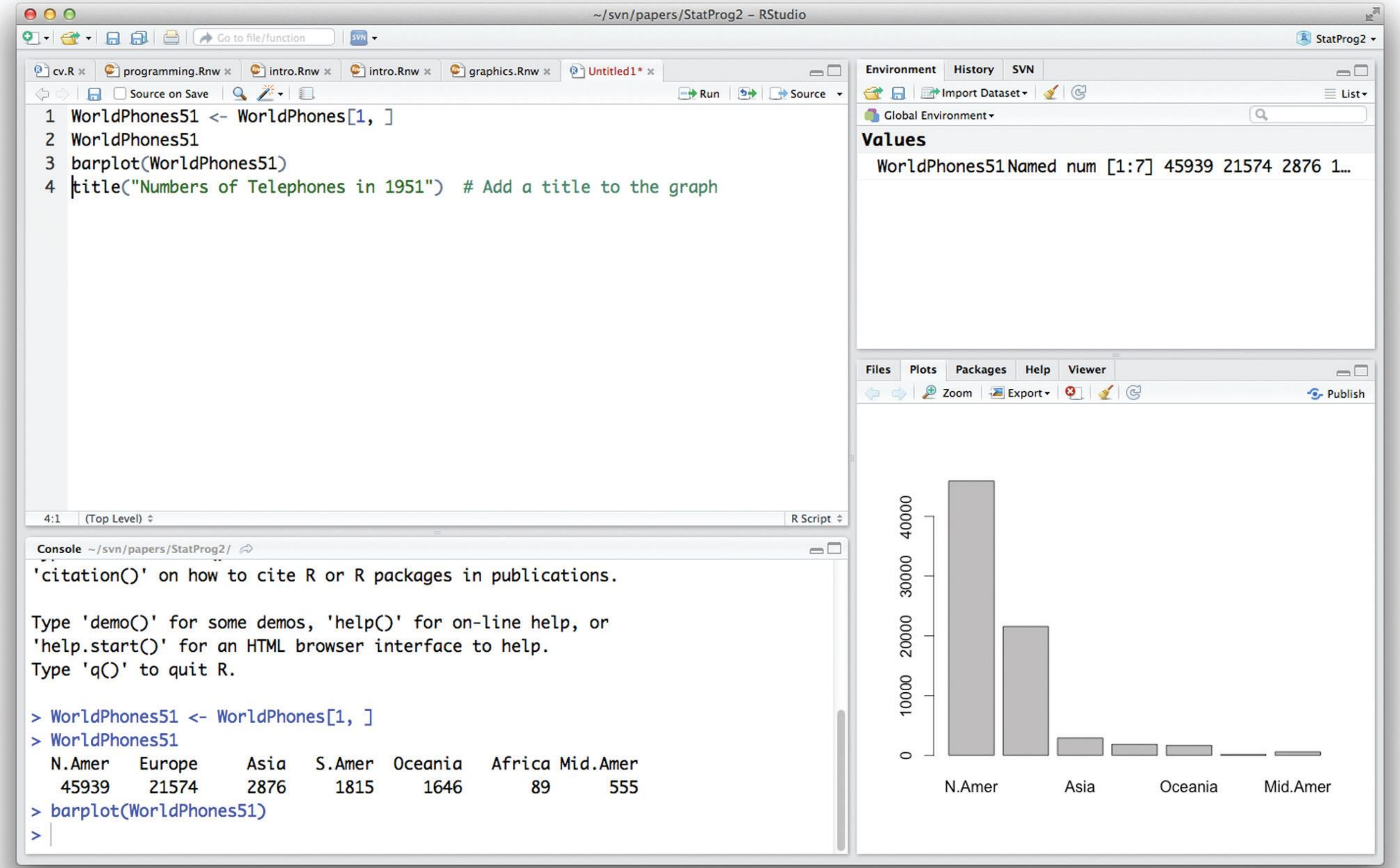
Correlation

Correlation is another way to determine how two variables are related. In addition to identify whether variables are positively or negatively related, it also inform about the degree to which the variables tend to move together.

Correlation standardizes the measure of interdependence between two variables and, consequently, tells you how closely the two variables move. The correlation measurement, called a correlation coefficient, will always take on a value between 1 and – 1.

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

RStudio



```r
WorldPhones51 <- WorldPhones[1, ]
WorldPhones51
barplot(WorldPhones51)
title("Numbers of Telephones in 1951")  # Add a title to the graph
```

```
Values
WorldPhones51 Named num [1:7] 45939 21574 2876 1…
```

```
'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.

> WorldPhones51 <- WorldPhones[1, ]
> WorldPhones51
 N.Amer   Europe     Asia   S.Amer  Oceania   Africa Mid.Amer 
  45939    21574     2876     1815     1646       89      555 
> barplot(WorldPhones51)
> 
```

Objects in R

- types of objects: vector, factor, array, matrix, data.frame, ts, list
- attributes
  - mode: numeric, character, complex, logical
  - length: number of elements in object
- creation
  - assign a value
  - create a blank object

Source: A first course in statistical programming with R - W. John Braun and Duncan J. Murdoch, Cambridge, 2016.

Naming Convention

- must start with a letter (A-Z or a-z)
- can contain letters, digits (0-9), and/or periods "."
- case-sensitive
  - `mydata` different from `MyData`
- do not use use underscore "_"

Objects in R

A <- c(1,2,3,4,5,6,7) or A <- 1:7

```
17:58

##  [1] 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
## [23] 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58
```

The first line starts with the first return value, so it is labeled `[1]`; the second line starts with the 23rd, so it is labeled `[23]`.

Everything that you type after a # sign is assumed to be a comment and is ignored by R.

```
5:(2*3 + 10)    # the result is  the same as 5:16

##  [1]  5  6  7  8  9 10 11 12 13 14 15 16

(7:10) + pi        # pi is a stored constant

## [1] 10.14159 11.14159 12.14159 13.14159
```

Source: A first course in statistical programming with R - W. John Braun and Duncan J. Murdoch, Cambridge, 2016.

Assignment

- "<-" used to indicate assignment
  - `x<-c(1,2,3,4,5,6,7)`
  - `x<-c(1:7)`
  - `x<-1:7`

- *note: as of version 1.4 "=" is also a valid assignment operator*

Functions

- actions can be performed on objects using functions (note: a function is itself an object)
- have arguments and options, often there are defaults
- provide a result
- parentheses () are used to specify that a function is being called

Workspace

- during an R session, all objects are stored in a temporary, working memory
- list objects
  - `ls()`
- remove objects
  - `rm()`

Two most common object types for statistics:

matrix
data frame

Matrix

- a matrix is a vector with an additional attribute (dim) that defines the number of columns and rows
- only one mode (numeric, character, complex, or logical) allowed
- can be created using `matrix()`

```
x<-matrix(data=0,nr=2,nc=2)
          or
x<-matrix(0,2,2)
```

Data Frame

- several modes allowed within a single data frame
- can be created using `data.frame()`

```
L<-LETTERS[1:4] #A B C D
x<-1:4          #1 2 3 4
data.frame(x,L) #create data frame
```

- `attach()` and `detach()`
  - the database is attached to the R search path so that the database is searched by R when it is evaluating a variable.
  - objects in the database can be accessed by simply giving their names

Extract Elements of Vector

```
x <- c(0, 2, 4, 6, 8, 10)
x[3] # access 3rd element
4
x[c(2, 4)] # access 2nd and 4th element
2 6
```

Negative indices can be used to avoid certain elements. For example, we can select all but the second and third elements of x as

```
 x[-c(2,3)]
```

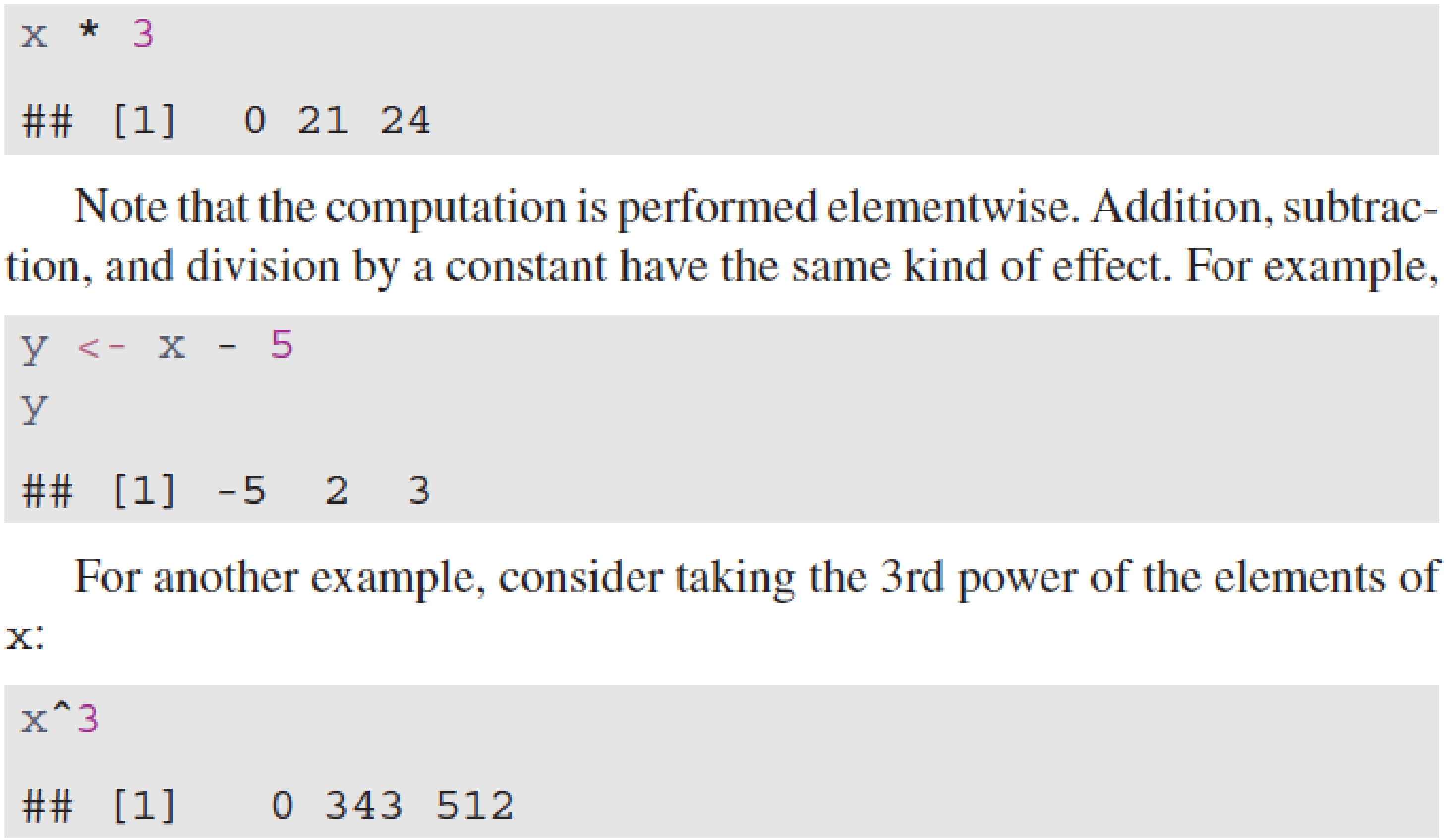
Data Elements

- select only one element
  - `x[2]`
- select range of elements
  - `x[1:3]`
- select all but one element
  - `x[-3]`
- slicing: including only part of the object
  - `x[c(1,2,5)]`
- select elements based on logical operator
  - `x(x>3)`

Vectors

- Vectors can be joined together (i.e. concatenated) with the c function.
- For example, note what happens when we type
- s <- c(x, a) # x and a are vectors.

Vectors Arithmetic



```
x * 3

## [1]  0 21 24
```

Note that the computation is performed elementwise. Addition, subtraction, and division by a constant have the same kind of effect. For example,

```
y <- x - 5
y

## [1] -5  2  3
```

For another example, consider taking the 3rd power of the elements of x:

```
x^3

## [1]   0 343 512
```

Character Vectors

```
a <- c(”I”, “am”, “learning”, ”R")
??
```

Scalars and vectors can be made up of strings of characters instead of numbers.

```
colors <- c("red", "yellow", "blue")
more.colors <- c(colors, "green", "magenta", "cyan")
# this appended some new elements to colors
z <- c("red", "green", 1)
```

Practice Examples

Script.R

Create a vector filled with random numbers

```
U1 <- rnorm(30)
```

Create a 30 x 30 matrix i.e. (30 rows and 30 columns)

```
mymat <- matrix(nrow=30, ncol=30)
```

Just show the upper left 10x10 chunk

```
mymat[1:10, 1:10]
```

Source: www.datacamp.com

Practice Examples

Calculate mean.

```
M = sum(xx)/length(xx)
M
```

Covert temperature Fahrenheit to Celsius.

```
temp_C <- (temp_F - 32) * 5 / 9
temp_C
```

Covert temperature Celsius to Kelvin.

```
temp_K <- temp_C + 273.15
 temp_K
```

Making Functions

Lets make a function to calculate mean.

```
#make a new R file and save it a my_mean.R
myMean = function(xx) {
M = sum(xx)/length(xx)
return(M)
}
```

Making Functions

Lets make a function to covert temperature Fahrenheit to Celsius.

```
#make a new R file and save it a F2C.R
fahrenheit_to_celsius <- function(temp_F) {
  temp_C <- (temp_F - 32) * 5 / 9
  return(temp_C)
}
```

Practice Functions

# Write R program to calculate the correlation.

- X <- c(65.21, 64.75, 65.26, 65.76, 65.96)
- Y <- (67.25, 66.39, 66.12, 65.70, 66.64)

- We can use built-in functions such as Sum(), mean(), cor()

- my.cor<-(1/(length(x)-1))*(sum(((x-mean(x))/sd(x))*(((y-mean(y))/sd(y)))))

R: Example Problem

- For this analysis, we will use the cars dataset that comes with R by default.

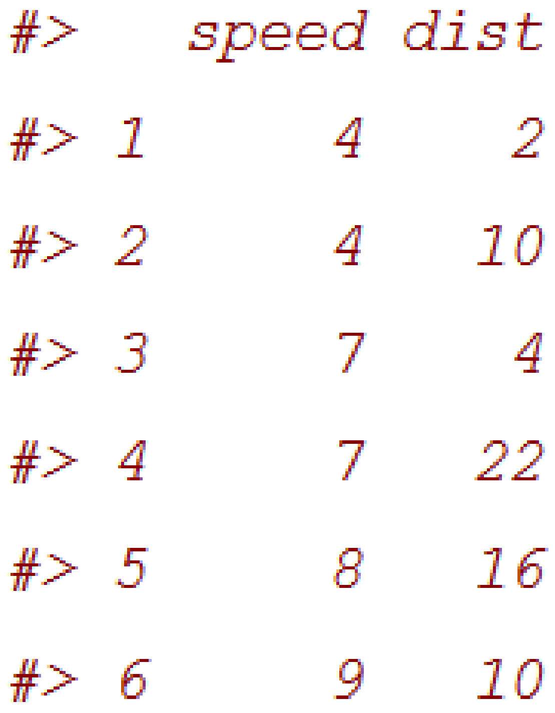
- cars is a standard built-in dataset, that makes it convenient to show analysis in a simple and easy to understand fashion.

- You can access this dataset by typing in cars in your R console.

- You will find that it consists of 50 observations(rows) and 2 variables (columns) – dist and speed. Lets print out the first six observations here.

R: Example Problem

- head(cars)

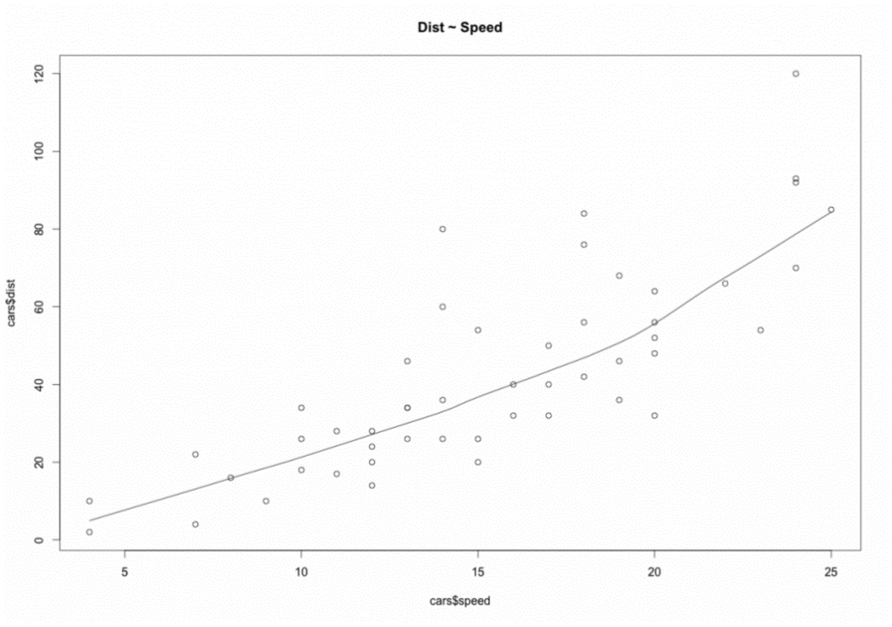


```
#>    speed dist
#> 1      4    2
#> 2      4   10
#> 3      7    4
#> 4      7   22
#> 5      8   16
#> 6      9   10
```

R: Example Problem

```
# Calculate correlation between speed and distance
R <- cor(cars$speed, cars$dist)
```

R ~ 0.807



The scatter plot along with the smoothing line above suggests a linear and positive relationship between the ‘dist’ and ‘speed’.

Source: https://www.machinelearningplus.com/machine-learning

Practice Example

- Example 2.1 :An individual wishes to take out a loan, today, of P at a monthly interest rate i. The loan is to be paid back in n monthly installments of size R, beginning one month from now. The problem is to calculate R.

- In R, we define variables as follows: principal to hold the value of P, intRate to hold the interest rate, and n to hold the number of payments. We will assign the resulting payment value to an object called payment.

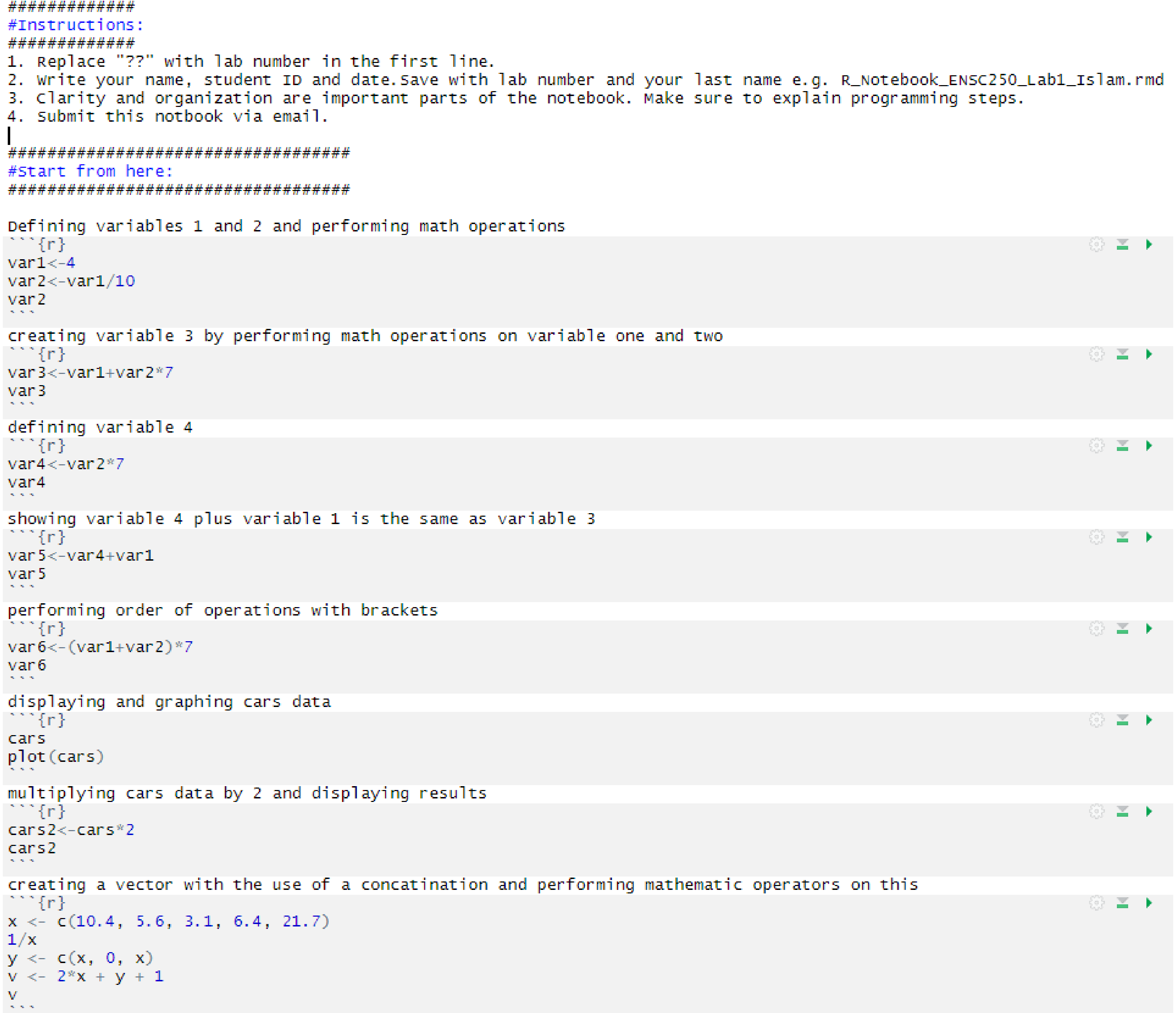
$$R = P\frac{i}{1-(1+i)^{-n}}$$

R Script for Interest Rate

Suppose that the loan amount is $1500, the interest rate is 1% and the number of payments is 10.

```r
intRate <- 0.01
n <- 10
principal <- 1500
payment <- principal * intRate / (1 - (1 + intRate)^(-n))
payment

## [1] 158.3731
```



~~~
#############
#Instructions:
#############
1. Replace "??" with lab number in the first line.
2. Write your name, student ID and date.Save with lab number and your last name e.g. R_Notebook_ENSC250_Lab1_Islam.rmd
3. Clarity and organization are important parts of the notebook. Make sure to explain programming steps.
4. Submit this notbook via email.

###################################
#Start from here:
###################################

Defining variables 1 and 2 and performing math operations
```{r}
var1<-4
var2<-var1/10
var2
~~~
creating variable 3 by performing math operations on variable one and two
```{r}
var3<-var1+var2*7
var3
```
defining variable 4
```{r}
var4<-var2*7
var4
```
showing variable 4 plus variable 1 is the same as variable 3
```{r}
var5<-var4+var1
var5
```
performing order of operations with brackets
```{r}
var6<-(var1+var2)*7
var6
```
displaying and graphing cars data
```{r}
cars
plot(cars)
```
multiplying cars data by 2 and displaying results
```{r}
cars2<-cars*2
cars2
```
creating a vector with the use of a concatination and performing mathematic operators on this
```{r}
x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
1/x
y <- c(x, 0, x)
v <- 2*x + y + 1
v
```
```
```