

Coding Basics

Part 2

More on Data Structure

Subsetting Vectors

Command: *name_of_vector[position/s you want to extract]*

Example 1: For the following vector:

`vector_1 <- c(2, 3, 4, 5, 5, 6)`

- What code do you need to write to extract the third element?

Code: `vector_1[3]`

output: 4

- What code do you need to write to extract the fourth through sixth element?

Code: `vector_1[4:6]`

output: 5 5 6

Subsetting Matrices

Since matrices are two-dimensional organization of data, to subset a matrix you need a number for the row and a number for the column.

Command: *name_of_matrix[row number, column number]*

Example 2:

- The following code defines a matrix, write out what the matrix will “look like” if we ran that code.

```
matrix_1 <-  
matrix(c(5,6,7,2,3,6), ncol = 2, byrow = FALSE)
```

$$\begin{bmatrix} 5 & 2 \\ 6 & 3 \\ 7 & 6 \end{bmatrix}$$

- Extract the element in row 3, column 1.

Code: `matrix_1[3,1]`

Output: **7**

5	2
6	3
7	6

- Extract the element in column 2, row 1.

Code: `matrix[1,2]`

Output: **2**

5	2
6	3
7	6

- Extract elements in column 2

Code: `matrix_1[,2]`

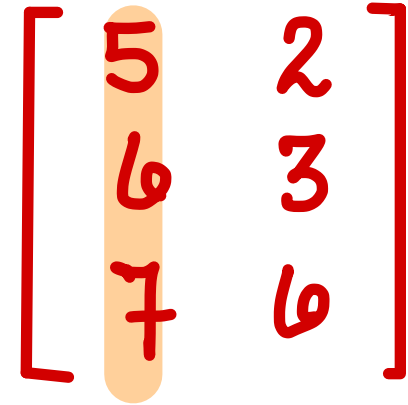
Output: **2 3 6**

5	2
6	3
7	6

- Extract all elements in column 1

Code: `matrix_1[:,1]`

Output: 5 6 7

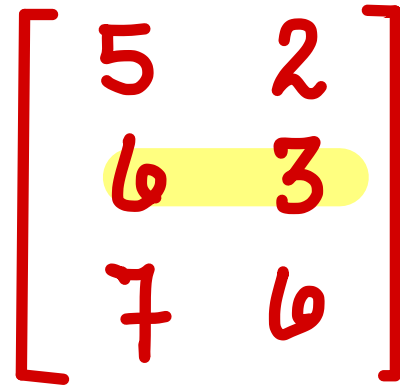


5	2
6	3
7	6

- Extract all elements in row 2


Code: `matrix_1[2,]`

Output: 6 3



5	2
6	3
7	6

Classes (Data Type)

1. **numeric**: any real number(s)
2. **character**: strings or individual characters, quoted 
3. **integer**: any integer(s)/whole numbers
4. **factor**: categorical/qualitative variables
5. **logical**: variables composed of **TRUE** or **FALSE**
6. **Date/POSIXct**: represents calendar dates and times

How to check the class of a vector?

Command: *class(name of the vector)*

Example 3: In the following table, I define 5 different vectors. Type the code you would use to check the class of each vector and in the third column, type what the output would be if you ran that code.

Vector	Code to Check each vector's class	Output
<code>x <- c(1,2,3,4,5)</code>	<code>class(x)</code>	numeric
<code>y <- c("mary", "joe", "john")</code>	<code>class(y)</code>	character
<code>k <- c("1", "2", "3")</code>	<code>class(k)</code>	character
<code>z <- factor(c("A", "B", "C", "A", "A", "B"))</code>	<code>class(z)</code>	factor
<code>w <- c(FALSE, TRUE, TRUE)</code>	<code>class(w)</code>	logical

How to change a vector from one class to another class?

Use commands the following commands:

- To change to a numeric vector: `as.numeric(name of the vector)`
- To change to a character vector: `as.character(name of the vector)`
- To change to a factor vector: `as.factor(name of the vector)`
- To change to a logical vector: `as.logical(name of vector)`

Example 4: Suppose you have the following vector

```
vector_2 <- c("3", "5", "7", "9")
```

- What is the class of this vector as it is?

Code: `class(vector_2)`

Output: `character`

- How would you change it to numerical?

Code: `vector_2 <- as.numeric(vector_2)`

Dataframes

Dataframe: is a rectangular collection of variables (in the columns) and observations (in the rows). You can think of every column in a dataframe as a vector.

age	month	year
18	1	2006
18	2	2006
19	5	2005
20	12	2004
18	1	2006

How to define dataframe?

Example 5: Define the following dataset as a dataframe.

age	month	year
18	1	2006
18	2	2006
19	5	2005
20	12	2004
18	1	2006

Code:

```
demographics <- data.frame(  
  age = c(18, 18, 19, 20, 18),  
  month = c(1, 2, 5, 12, 1),  
  year = c(2006, 2006, 2005, 2004, 2006))
```

Example 6: Define the following dataset as a dataframe.

V1	V2	V3
1	1	1
2	1	3
3	1	5
4	1	7
5	1	9

Code: `df <- data.frame(V1 = 1:5,
V2 = rep(1,5),
V3 = seq(1,9,2))`

replicates
1, 5 times

Sequence
from 1
to 9 with
increases
of 2.

Subsetting dataframes

Command: *name_of_dataframe\$name_of_column*

Example 7: What code would you write to pull/subset the column year from the dataframe in Example 5.

Code: *demographics\$year*

Output: *2006 2006 2005 2004 2006*

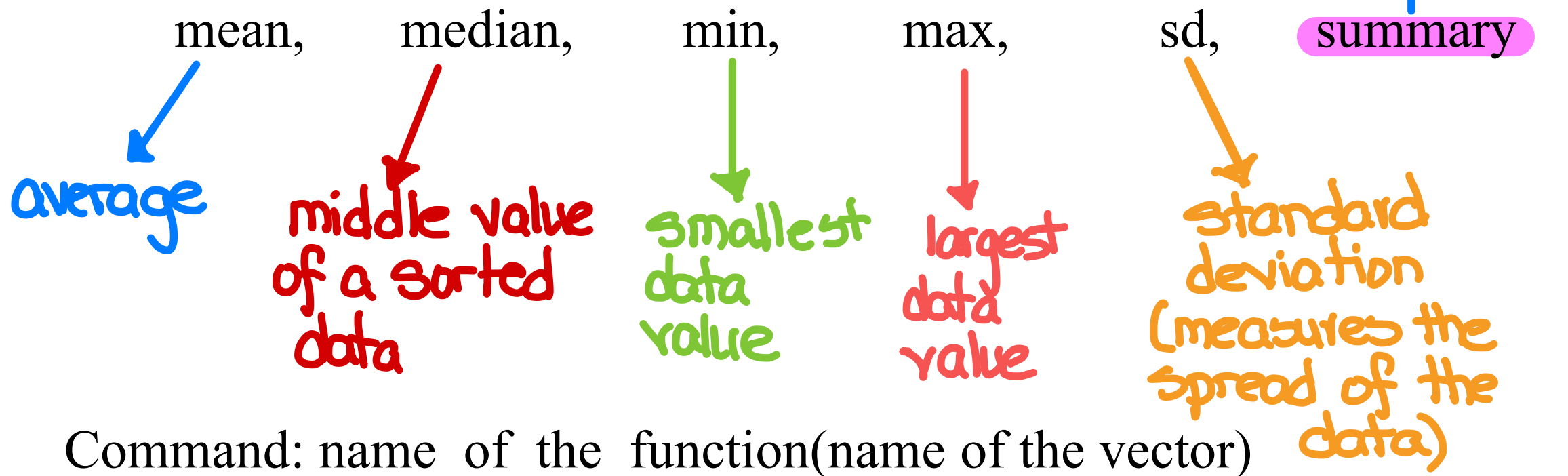
Example 8: What code would you write to check the class of the second column from the dataframe in Example 5.

Code: `class(demographics$ month)`

Output: `numeric`

Functions in R

For today we will study the following functions:



Example 9: Suppose you have the following vector:

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

- How would you compute the mean?

Code: `mean(x)`

Output: `4`

- How would you compute the minimum value?

Code: `min(x)`

Output: `1`

- How would you compute the standard deviation?

Code: `sd(x)`

Output: `2.38`

Boolean Expressions

Boolean expressions: expressions that are either true or false

Commands: use `==`, `!=`, `>`, `<`, `>=`, or `<=` to compare two expressions.

checks
for
equality

checks
for
"not
equal to"

Smaller
than

greater
than or
equal to.

greater than

Smaller
than or
equal to

Example 10: Suppose you defined the following vectors:

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

```
y <- 4:7
```

```
z <- c(4, 5, 7, 9)
```

What are the outputs if I ran the following lines of codes?

Code	Output
<code>x == y</code>	TRUE TRUE TRUE TRUE
<code>x == z</code>	TRUE TRUE FALSE FALSE
<code>x != z</code>	FALSE FALSE TRUE TRUE
<code>x > 6</code>	FALSE FALSE FALSE TRUE
<code>x <= 5</code>	TRUE TRUE FALSE FALSE