R Basics

STAT201

Winter 2025

Upcoming tentative schedule

	2/17/2025	Monday	R: Basics	Chapter 8	Assignment 5
7	2/19/2025	Wedesday	R: Control flow tools	Chapter 9	
	2/24/2025	Monday	R: Functions	Chapter 10	Assignment 6
8	2/26/2025	Wedesday	R: Data Structures	Chapter 11	
	3/3/2025	Monday	R: Data Structures	Chapter 11	
9	3/5/2025	Wedesday	Github		Assignment 7
	3/10/2025	Monday	Github & Final Review		
10	3/12/2025	Wedesday	**WCAS Reading Period**		
11	3/20/2025	Thursday	**Final Exam (3-5pm)**		

Small Assignments

Grading

Grading Scale	
93 - 100%	A
89 - 93%	A-
86 - 89%	B+
83 - 86%	B
79 - 83%	B-
76 - 79%	C+
73 - 76%	C
69 - 73%	C-
60 - 69%	D
Below 60%	F

Category	Weight
Small assignments/assessments	15%
Assignment	30%
Midterm Exam	25%
Final Exam	25%
Participation	5%

15 quizzes will be considered for your final grading

Complete all the quizzes will earn you an additional 2% participation credit

Shortcuts in RStudio

- **Insert a new code chunk (R Markdown):**
  - Windows/Linux: `Ctrl + Alt + I`
  - Mac: `Cmd + Option + I`

- **Run current line or selection:**
  - Windows/Linux: `Ctrl + Enter`
  - Mac: `Cmd + Enter`

R comments

- Comments in R are written using the `#` symbol.
- Anything after `#` is ignored by the R interpreter.
- Example:

```
# This is a comment
print("Hello, World!") # This prints a message
```

Data Types in R

- Common data types:
  - **Numeric:** `x <- 10.5`
  - **Integer:** `y <- as.integer(10)`
  - **Character:** `name <- "R Language"`
  - **Logical:** `is_r_fun <- TRUE`
- Example:

```r
x <- 5.5
typeof(x) # Output: "double"
```

Variables in R

- Variables store data values (information)
- Here are variable naming rules:
  - A variable name must start with a letter and can be a combination of letters, digits, period(.) and underscore(_). If it starts with period(.), it cannot be followed by a digit.
  - A variable name cannot start with a number or underscore (_)
  - Variable names are case-sensitive (age, Age and AGE are three different variables)
  - Reserved words cannot be used as variables (TRUE, FALSE, NULL, if...)

The assignment Operators

8.3.1.1 Using `<-` (**Preferred Operator**)

The `<-` operator is the standard way to assign values in R:

```
x <- 10
y <- "Hello, R!"
z <- TRUE
```

Using the -> operator

```
```{r}
5.1 -> x
typeof (x)
```
```

8.3.1.2 Using `=` (**Not Recommended**)

Although `=` can be used for assignment, it is generally not recommended because it can cause issues in function arguments:

```
x = 10  # Works, but `<-` is preferred
```

💡 **Best Practice:**

- Always use `<-` for assignments to avoid ambiguity.

How to check variable type in R

```
a <- 1
b <- 2.4
c <- "Set of characters"
d <- TRUE
i <- as.integer(a)
```

```
#checking data type
print(typeof(a))
print(typeof(b))
print(typeof(c))
print(typeof(d))
print(typeof(i))
```

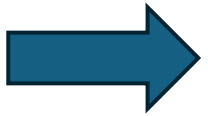
```
Output:
[1] "double"
[1] "double"
[1] "character"
[1] "logical"
[1] "integer"
```

```
#checking data type
print(class(a))
print(class(b))
print(class(c))
print(class(d))
print(class(i))
```

```
Output:
[1] "numeric"
[1] "numeric"
[1] "character"
[1] "logical"
[1] "integer"
```

- Using typeof()
- Using class()
- Using is.datatype()

```
Code:
#checking data type
print(is.numeric(a))
print(is.complex(b))
print(is.character(c))
print(is.logical(d))
print(is.integer(i))
```



```
Output:
[1] TRUE
[1] FALSE
[1] TRUE
[1] TRUE
[1] TRUE
```

Converting data types

From → To	Conversion Function	Example Usage	Notes
Numeric → Character	`as.character(x)`	`as.character(42)` → `"42"`	Converts numbers to strings
Numeric → Logical	`as.logical(x)`	`as.logical(0)` → `FALSE`	`0` is `FALSE`, non-zero is `TRUE`
Character → Numeric	`as.numeric(x)`	`as.numeric("3.14")` → `3.14`	Returns `NA` if conversion fails
Character → Logical	`as.logical(x)`	`as.logical("TRUE")` → `TRUE`	Case-sensitive, `"TRUE"` and `"FALSE"` work
Logical → Numeric	`as.numeric(x)`	`as.numeric(TRUE)` → `1`	`TRUE = 1`, `FALSE = 0`
Logical → Character	`as.character(x)`	`as.character(FALSE)` → `"FALSE"`	Converts logical values to text

Display information in R

- Use `print()` to display output:

```r
print("Hello, R!")
```

- Use `cat()` to print multiple values:

```r
cat("Hello", "R", "Programming!\n")
```

- Use `paste()` to concatenate and print:

```r
message <- paste("Hello", "R", "World!")
print(message)
```

- Use `message()` for warning messages:

```r
message("This is a message in R")
```

- Use `sprintf()` for formatted strings:

```r
name <- "R"
version <- 4.0
sprintf("Welcome to %s version %.1f", name, version)
```

Take input in R

- Use `readline()` for user input:

```
name <- readline(prompt="Enter your name: ")
print(paste("Hello,", name))
```

- Convert input to numeric:

```
age <- as.numeric(readline(prompt="Enter your age: "))
print(age + 1)
```

Arithmetic operators

Operation	Symbol	Example	Result
Addition	`+`	`5 + 3`	`8`
Subtraction	`-`	`10 - 4`	`6`
Multiplication	`*`	`6 * 2`	`12`
Division	`/`	`8 / 2`	`4`
Exponentiation	`^` or `**`	`3^2` or `3**2`	`9`
Integer Division	`%/%`	`10 %/% 3`	`3`
Modulo (Remainder)	`%%`	`10 %% 3`	`1`

Logical Operators

Logical operators are used to combine conditions.

Operator	Symbol	Example	Result
AND	`&`	`(5 > 3) & (2 < 4)`	`TRUE`
OR	`|`	`(5 > 3) | (2 > 4)`	`TRUE`
NOT	`!`	`!(5 > 3)`	`FALSE`