

# Lab2

SAF

2025-02-22

## Part I: Introduction to modeling using basic R syntax

*Congratulations, you have puppy fever! As someone who is inflicted with puppy fever, you would like to buy as many puppies as you perceivable can. To help you determine if you can afford all of the puppies you want or to determine how many puppies you can afford, you opt to use R to help you. In this exercise, please print the contents of each variable after you declare it.*

**Integer:** *puppies* variable holds the number of puppies you'd like to have.

```
## puppies = 12
```

**Integer:** *max\_puppies* variable holds Qty of puppies you can afford for \$1,000.

```
## max_puppies = 1
```

**Numeric:** *puppy\_price* variable holds the price of a single puppy.

```
## puppy_price = 1.01
```

**Numeric:** *total\_cost* variable holds the total price of all puppies.

```
## total_cost = 1.01
```

**Logical-Bool** *too\_expensive* Variables return TRUE if the cost is greater than \$1,000.

```
## too_expensive= FALSE
```

## Part II: Manipulating variables and learning how to use new functions

You work as a data analyst for a new company and are asked to create id tags for everyone at work. Your goal is to make it informative as well as personal to help facilitate collaboration in the work place. To do this, you first want to gather information about each employee.

**2a.Character** *my\_name* This assigns your name to the variable. Assign *my\_name* to a variable in (4) different ways.

```
## -----
```

```
## [1] "Sal F"
```

```
## Sal F
```

```
## [1] "Sal F"
```

```
## Sal F
```

**2b.Character** *favorite\_day* holds your favorite day of the week.

```
## -----
```

```
## My favorite day is: Thursday
```

**2c.Integer** *my\_height* Assigns your height in whole inches values.

```
## -----
```

```
## My height is: 71 inches
```

**2d.Character** *favorite\_quote* Holds your favorite quote.

```
## -----
```

```
## My favorite quote is about water: Whiskey is for drinking; water is for fighting over. It has 52 characters
```

**2e.Type of Data Objects** Verify what type *my\_name*, *my\_height*, *favorite\_day*, and *favorite\_quote* are.

```
## -----
```

```
## my_name type is a character
```

```
## favorite_day type is a character
```

```
## my_height type is an integer
```

```
## favorite_quote type is a character
```

**2f.Coerce these variables** to a *numeric* and describe what happens.

```
## -----
```

```
## Warning in cat("my_name as a numeric:", as.numeric(my_name)): NAs introduced by coercion
```

```
## my_name as a numeric: NA
```

```
## Warning in cat("favorite_day as numeric:", as.numeric(favorite_day)): NAs introduced by coercion
```

```
## favorite_day as numeric: NA
```

```
## my_height as numeric: 71
```

```
## Warning in cat("favorite_quote as numeric:\n", as.numeric(favorite_quote)): NAs introduced by coercion
```

```
## favorite_quote as numeric:
```

```
## NA
```

**2g.Create of Vector** named "id" that contains *my\_name*, *my\_height*, *favorite\_day*, and *favorite\_quote*.

```
## -----
```

```
## id vector: Sal F 71 Thursday Whiskey is for drinking; water is for fighting over.
```

**2h.Class of Vector** What class is "id"? Did the classes change for the variables themselves?

```
## -----
```

```
## The vector id is a type of:
```

```
## character
```

```
## Below are the data types for each variable. Verify if variables weren't altered
```

```
## my_name type: character
```

```
## favorite_day type: character
```

```
## my_height type: integer
```

```
## favorite_quote type: character
```

**2i.Employee's Information** Your employer wants you to be able to print each employee's id while displaying each variable of information line by line. As a beginner with R, however, you are unfamiliar with how to do this so your employer gives you a hint to use the functions `cat` and `paste`. Try using `cat` and `paste` with `id` as a function argument. How do the results differ? What happens when we use `cat` and `paste` at the same time (*i.e.*  $f(g(x))$ )? What happens if we change the order we use them (*i.e.*  $g(f(x))$ )?

```
## cat function(id):, Sal F, 71, Thursday, Whiskey is for drinking; water is for fighting over.,
## paste function(id):
## [1] "Sal F"
## [2] "71"
## [3] "Thursday"
## [4] "Whiskey is for drinking; water is for fighting over."

## f(g(x))-Using paste function: Sal F 71 Thursday Whiskey is for drinking; water is for fighting over
## g(f(x))-Using cat function:
## Sal F 71 Thursday Whiskey is for drinking; water is for fighting over.
## character(0)
```

**2j.Difference between cat and paste** How would you determine the difference between `cat` and `paste` using R documentation (from within RStudio)?

What is a great internet resource to use as discussed in the book? RStudio <https://forum.posit.co/>  
<https://www.rdocumentation.org/>

What do `sep` and `collapse` arguments for `paste` do? They control the spacing between each string fragment (word). Where `collapse` removes all spaces and the `Sep` command allows the user to dictate the char that will be used to speratarte the string sements.

If we wanted to append each character variable in our vector `id` with a new line (*i.e.* `\n`) would we use `sep` or `collapse`? To add a new line (`\n`) to each character variable in a vector, you would first add the new line character to each element using `paste()` and then `collapse` them into a single string

**2k. cat and paste** Display the contents of the `id` function using a combination of `cat` and `paste` with the appropriate arguments for `paste`.

```
## 1. Sal F
## 2. 71
## 3. Thursday
## 4. Whiskey is for drinking; water is for fighting over.
```

## Part III: Accessing data in GitHub and mastering order of operations

```
## -----
```