

R refresher for:

Introduction to



R-Ladies Nijmegen 21/02/19

Hiya! :)

Do I **have to go through these slides?**

Nope! This material is for those who're not yet comfortable with:

1. using the functions `hist()` and `paste()`
2. subsetting a data frame (or list), eg. `df[["colname"]]`
3. assigning and retrieving values, eg. `x <- "colname"` then `df[[x]]`
4. writing a function with two arguments

Using `hist()`

- `hist()` plots a histogram
- We'll mainly use 3 arguments:
 - `x` is a vector of values, eg.
`mtcars$cyl` (ie. all the values from the `cyl` column of the `mtcars` data frame)
 - `main` is the title of the plot
 - `xlab` is the label of the x-axis

```
# try this example and your own! :)

hist(
  x = mtcars$cyl,
  main = "Histogram of cyl column of
the mtcars data frame",
  xlab = "cyl"
)
```

Using `paste()`

- `paste()` turns as many arguments as you want into text and combines them
- By default, a space is put between each argument

```
# try this example and your own! :)
```

```
paste(  
    "Paste puts",  
    "bits of text together!",  
    "Check it out! :D"  
)
```

Assigning with <-

- <- can be used to save a value in a variable to be reused later

```
# try this example and your own! :)
```

```
x <- "Hi! :)"
```

```
x
```

```
# the two results are identical:
```

```
identical(x, "Hi! :)")
```

Subsetting a data frame or list

- There are two ways we'll get the values from a column in a data frame or a part of list during the workshop:
 1. `df$colname` or `list$part`
 2. `df[["colname"]]` or `list[["part"]]`

```
# try this example and your own! :)  
  
df <- data.frame(col1 = 1:3, col2 = 4:6)  
  
df$col1  
  
df[["col1"]]  
  
  
# the two results are identical:  
  
identical(df$col1, df[["col1"]])
```

Writing a function with two arguments

- **function()** is used to (surprise, surprise) define functions!
- Arguments can be passed into and used in the function
- Defined functions can be assigned to a name for later use

```
# try this example and your own! :)

add_numbers <- # assign function

  function(arg1, arg2) {

    # where the action happens, eg.

    arg1 + arg2

  }

add_numbers(1+2) # use function later
```