

R you ready?

**IntRo to RStudio and R Markdown
for open data and reproducibility**

Unit 3:

**Baby steps: Basics of coding in
RStudio, part 2**

Mason A. Wirtz



WHEN YOUR R CODE RUNS



WITHOUT ERRORS

memehackerator.net

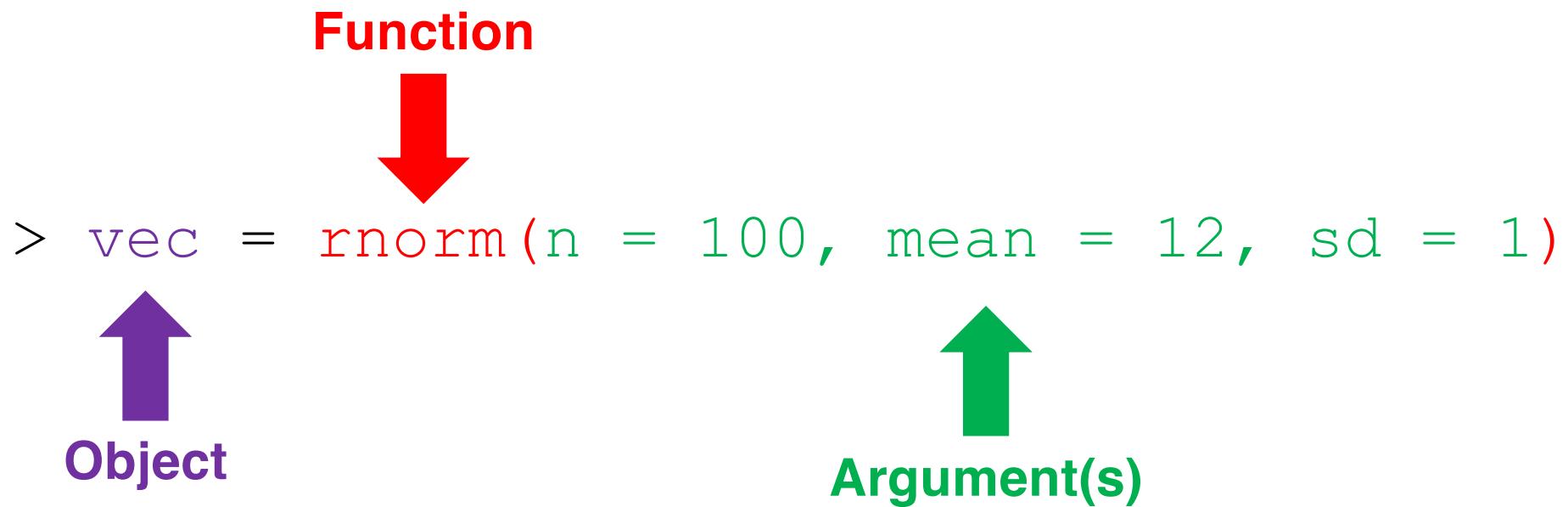
Some more on data frames

Working with data frames

Most of us will work almost exclusively with data frames, so we need to have a good grasp on how we handle these in R

For this, we need functions

Structure of R expressions



Basic functions in R

Everyone, go ahead and type in the following code:

```
> vec = rnorm(n = 100, mean = 12, sd = 1)
```

Now, run the following code

```
> ?rnorm
```



**What did we just do?
What does the function `rnorm()` do?**

Basic functions in R

We can then use other functions to explore this vector

```
> mean(vec)  
> sd(vec)  
> median(vec)
```

Our output should be similar to

```
[1] 11.98811  
[1] 0.9585754  
[1] 11.97012
```

Basic functions in R

We can transform/manipulate this data however we want using base R functions

Let's log transform the data

```
> log(vec)
```

This function log transforms each of the 100 values in the vector

Basic functions in R

If we are uncertain of which data structure we have, we can check this

This requires the `class()` function

```
> class(vec)
```

Our output should be

```
[1] "numeric"
```

Exploring data frames

Exploring data frames

In our `Vampires` **data frame**, we want to know which variables we have

We can use the `colnames()` function

```
> colnames(Vampires)
```

Our output should be

```
[1] "idVampire"      "gender"          "ageOfVampire"  
[4] "deadOrAlive"    "hasFangs"        "bornIn"  
[7] "visitedCities"   "numberOfChildren" "numberChangedToVamp"
```

Exploring data frames

Call the following functions.

What do they do?

```
> head(Vampires)  
> tail(Vampires)  
> str(Vampires)  
> summary(Vampires)
```



What is the \$ operator?

e.g. > Vampires\$ageOfVampire

Exploring data frames

We can call functions for specific variables/vectors

How?

```
> head(Vampires$ageOfVampire)  
> tail(Vampires$ageOfVampire)  
> str(Vampires$ageOfVampire)  
> summary(Vampires$ageOfVampire)
```

Loading in packages in R

Loading in packages in R

But what we can actually do with the data frames is limited in base R.

We need **external packages**

e.g. if I want to run a **frequentist mixed-effects model**, I would need the package **lme4**, or if I wanted to do some **Bayesian modeling**, I would need the package **brms**, or if I want to **plot like a champ**, I would need **ggplot2**

Exploring data frames

We need two very special functions to load in and call external packages

Install **NEW PACKAGES** (i.e. packages you haven't ever installed)

```
> install.packages ("")
```



The package you are loading in **MUST BE UNDER QUOTATION MARKS**

LOAD PACKAGES (i.e. packages you have already installed)

```
> library()
```

Exploring data frames

Let's go ahead and try an example

We want to load in the package `ggplot2`, how do we do this?

```
> install.packages("ggplot2")
> library(ggplot2)
```



Once you have installed a package via
`install.packages("")`, **YOU DO NOT NEED TO
DO IT AGAIN**

**You do need to load in your packages via
`library()` in every R session.**

LET'S GET OUR HANDS DIRTY



makeameme.org