

# R workshop

Action Again!

(Operators and Functions Part 2)

Load dataset1

# Working Directory Review

- `getwd()`
- `setwd( ' /yourpath/goes/here' )`

# Do it yourself!

- Save datasets to a folder on your computer
- Set your working directory to folder with your datasets
- Load the dataset
- Attach your dataset ('attach' function)

# REMOVE POST-ITS!!



(and save!)

# Indexing Review

- Indexing a vector (1 dimension)  
> `subject[1]`
- The numbers gives us the item in the first location of this vector.
  
- Indexing a data frame (2 dimensions)  
> `dataset1[1,2:3]`

This gives us the first row and 2<sup>nd</sup> thru 3<sup>rd</sup> column of this dataset.

# Logical Operator Review

- Returns a value of TRUE or FALSE (boolean)

==	equality
!=	inequality
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

# Logical Operator Review

- Multiple conditions

&	and
	or
!	not

```
                Is this true?      AND      Is this true?  
> (10 < 100 & 24 == 23 + 1)  
[1] TRUE  
> (5 > 4 & 5 > 10)  
[1] FALSE  
> (5 > 4 | 5 > 10)  
[1] TRUE
```

# Exercise

- Identify which subjects have anxiety greater than 6
- `>anxiety > 6`
- `>which(anxiety > 6)`

# Which

- **which()** tells us the locations of data which meet certain criteria.
- Why is this useful?

# Exercise

```
> anxiety > 6
```

```
[1] 5 6 9
```

```
> dataset1[ c(5,6,9), ]
```

```
> dataset1[ which(anxiety>6), ]
```

```
install.packages("babynames")  
install.packages("ggplot2")
```

```
library(babynames)  
library(ggplot2)
```

```
MyName <- "Sara"  
birthday <- 1990  
MySex <- "F"
```

```
data("babynames")  
colnames(babynames)  
myName.df <- subset(babynames, name == MyName)
```

```
ggplot(myName.df, aes(x = year, y = prop, color=sex)) +  
  geom_line() +  
  geom_point(aes(x = birthday,  
                 y = myName.df[myName.df$name == MyName &  
                               myName.df$year == birthday &  
                               myName.df$sex == MySex, "prop"]),  
            color="black") +  
  ggtitle(paste("Popularity of", MyName))
```

# REMOVE POST-ITS!!



(and save!)

# Subset your data

- Sometimes, you will want to perform functions on only some of your data points
- You can subset your data to identify subjects in a certain subgroup (e.g., females, persons over 40)

# Exercise

```
> subset(x = dataset1,  
        subset = anxiety > 6)
```

```
> subset(x = dataset1,  
        subset = anxiety > 6 &  
          gender == "female")
```

```
install.packages("babynames")  
install.packages("ggplot2")
```

```
library(babynames)  
library(ggplot2)
```

```
MyName <- "Sara"  
birthday <- 1990  
MySex <- "F"
```

```
data("babynames")  
colnames(babynames)
```

```
myName.df <- subset(babynames, name == MyName)
```

```
ggplot(myName.df, aes(x = year, y = prop, color=sex)) +  
  geom_line() +  
  geom_point(aes(x = birthday,  
                 y = myName.df[myName.df$name == MyName &  
                               myName.df$year == birthday &  
                               myName.df$sex == MySex, "prop"]),  
            color="black") +  
  ggtitle(paste("Popularity of", MyName))
```

# REMOVE POST-ITS!!



(and save!)

# What if I don't remember the arguments?

- Check the R documentation.  
`>?subset`
- When in doubt, you can always search the “Help” tab or the Internet.

# Back to Arguments

- Most functions take more than one argument.
- Separate arguments with commas.

```
> round (x = 2.30467, digits = 3)  
[1] 2.305
```

Number of  
digits to round  
to.



# Arguments have Names

- Most arguments in functions have names.
- It is recommended that you use those names when using a function.

```
> round (x = 2.30467, digits = 3)  
[1] 2.305
```

# Exercise

- Look up the `subset()` function.
- What is the difference between arguments called `subset =` and `select =` ?

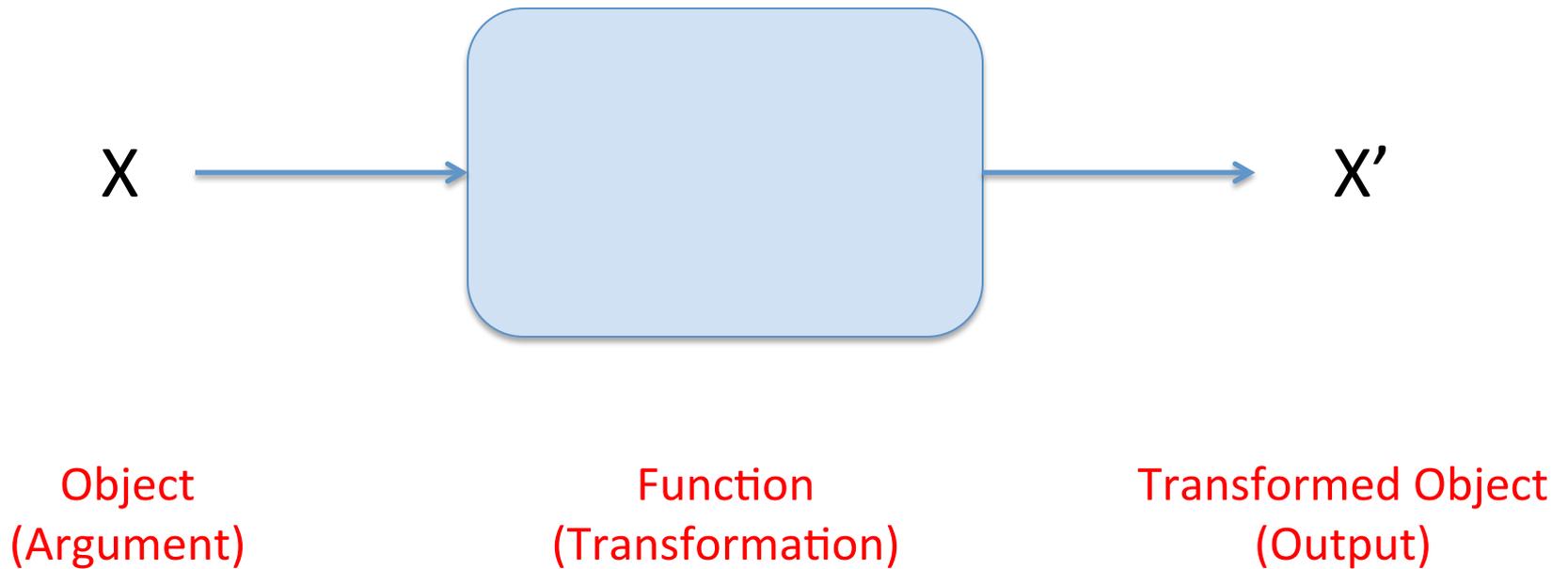
# Functions within Functions

- Often you can wrap functions within functions.
- Also called “nesting.”

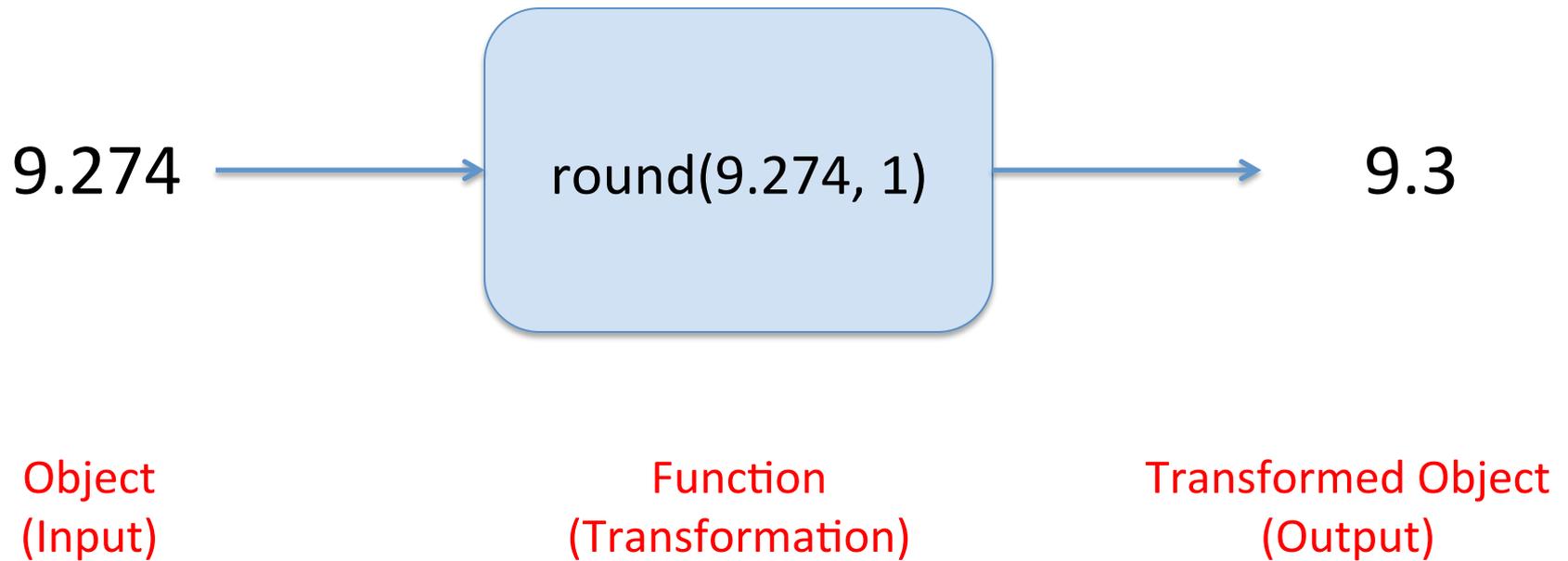
```
> round( sqrt(86) )
```

```
[1] 9
```

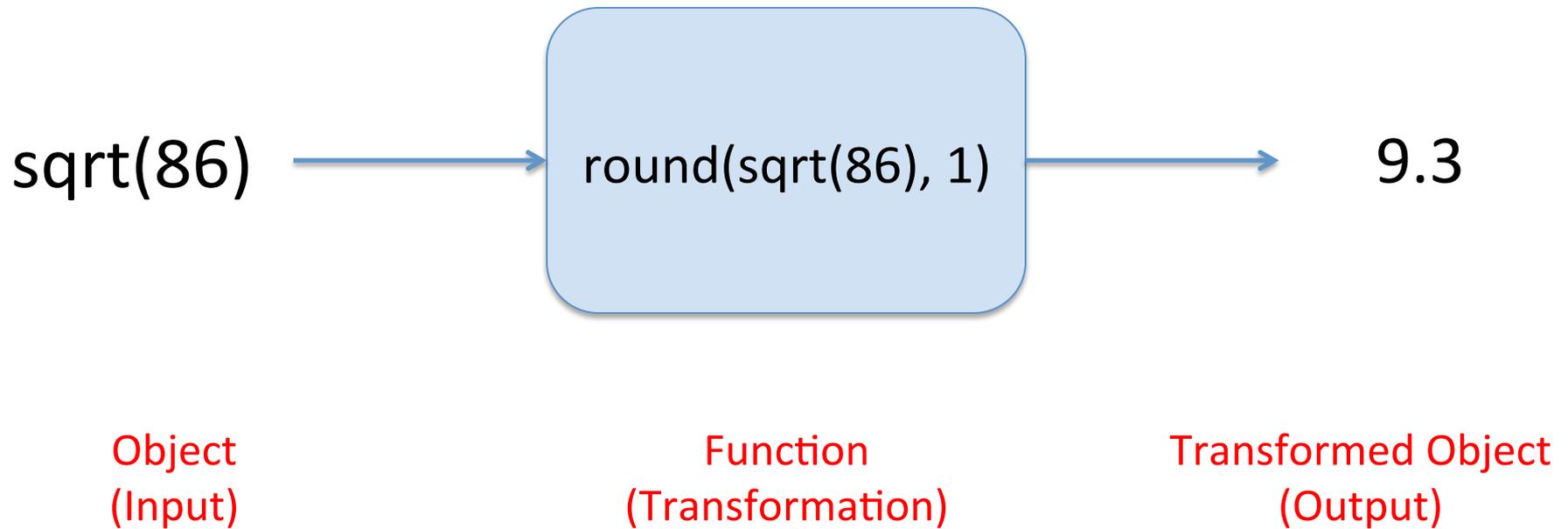
# Visualizing Functions



# Visualizing Functions

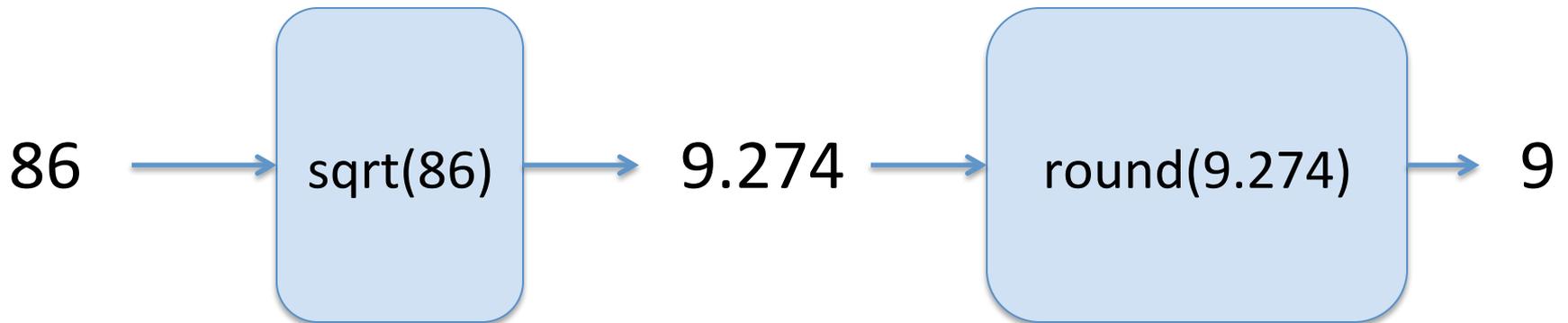


# Visualizing Functions



# Visualizing Nested Functions

round(sqrt(86))



Object  
(Input)

Transformed Object  
(Output)

# Exercise

- Find the mean of the variable anxiety.
- Identify which participants have anxiety scores above the mean.
- Use nested functions to create a subset of the dataset which contains only participants who have anxiety score above the mean.

# Exercise solution

```
> mean(anxiety)
```

```
> which(anxiety > mean(anxiety))
```

```
> subset(x = dataset1,  
         subset = anxiety > mean(anxiety))
```

# What if I don't remember how to use a function?

- Use the internet

# Favorite Websites

- Quick-R
- Cookbook for R
- Institute for Digital Research and Education (UCLA)
- Stack Overflow