

Ch 2 Workflow: basics

2.1 Coding basics

Can use R to do basic math

```
1 / 200 * 30
```

```
## [1] 0.15
```

```
(59 + 73 + 2) / 3
```

```
## [1] 44.66667
```

```
sin(pi / 2)
```

```
## [1] 1
```

Create new objects with assignment operator

```
x <- 3 * 4  
x
```

```
## [1] 12
```

Combine multiple elements into a vector with `c()`

```
primes <- c(2, 3, 5, 7, 11, 13)
```

Basic arithmetic on vectors is applied to every element

```
primes * 2
```

```
## [1] 4 6 10 14 22 26
```

```
primes - 1
```

```
## [1] 1 2 4 6 10 12
```

All assignment statements have same form

```
#object_name <- value
```

Shortcut for assignment operator is Option + - (minus sign)

2.2 Comments

```
# Create vector of primes
primes <- c(2, 3, 5, 7, 11, 13)

# Multiply primes by 2
primes * 2
```

```
## [1]  4  6 10 14 22 26
```

Use comments to explain why, not the how or what The how or what can be reconstructed from code itself, but not the why Use comments to explain plan of attack and record important insights

2.3 What's in a name?

Posit recommends using snake_case to name objects

```
#i_use_snake_case
#otherPeopleUseCamelCase
#some.people.use.periods
#And_aFew.People_RENOUNCEconvention
```

Inspect an object by typing its name

```
x
```

```
## [1] 12
```

Make another assignment

```
this_is_a_really_long_name <- 2.5
```

Can use tab to autocomplete long names Can type “this” and hit cmd up arrow in console to list commands that start with those letters

Make another assignment

```
r_rocks <- 2^3
```

Try to inspect it

```
#r_rock
#R_rocks
```

Must be completely precise in instructions, case sensitive and no typos

2.4 Calling functions

Built-in functions are called like this

```
#function_name(argument1 = value1, argument2 = value2, ...)
```

Try using seq()

```
seq(from = 1, to = 10)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

Often omit the names of first several arguments in function calls

```
seq(1, 10)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

R provides assistance pairing quotation marks

```
x <- "hello world"
```

If mismatch, R will show the continuation character “+”

```
#x <- "hello
```

Add the missing pair or press escape to try again

2.5 Exercises

1. Why does this code not work?

```
my_variable <- 10  
# my_variable
```

There is a typo when trying to inspect the variable, so the object is not found. You have to type “my_variable” with an “i” instead of a “l” or R will try to look for a variable called “my_var1able” which was not created.

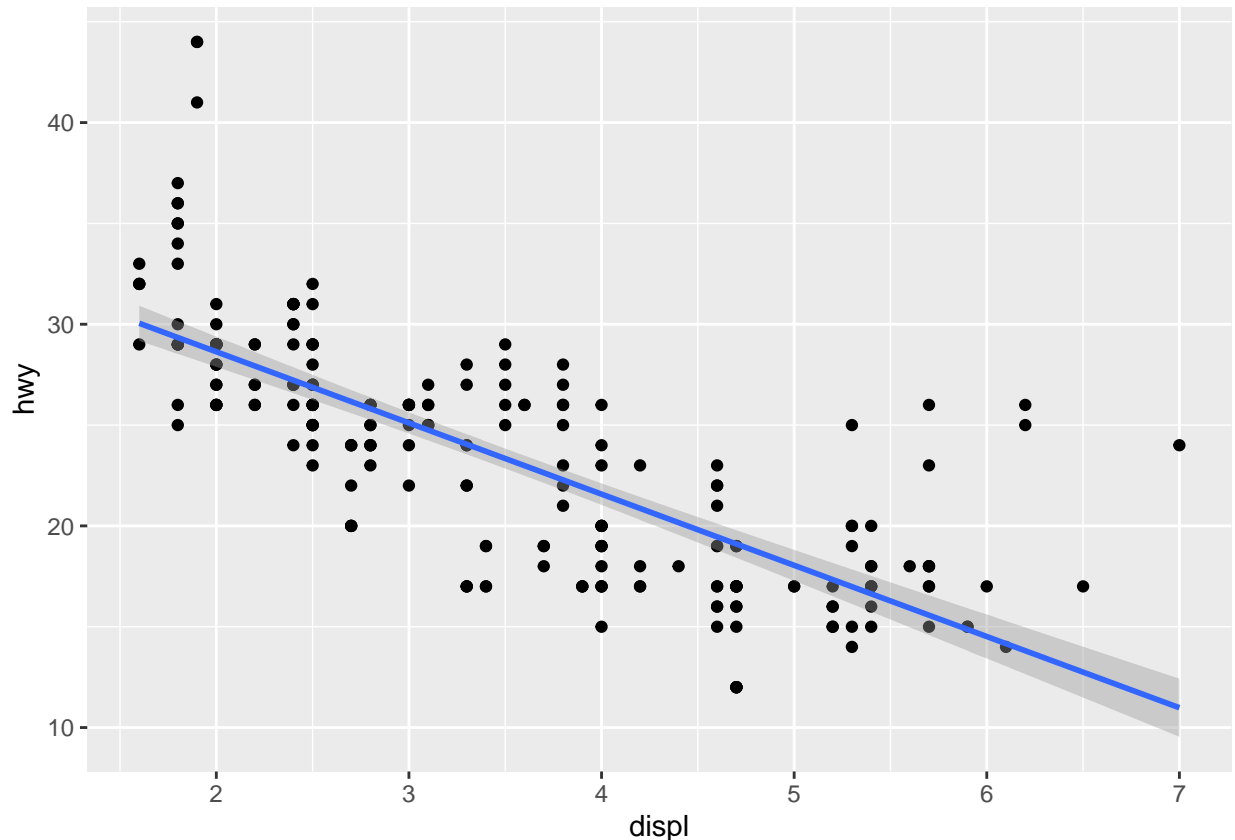
2. Tweak each of the following R commands so they run correctly

```
library(tidyverse) # From library and tidyverse
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v dplyr      1.1.4      v readr      2.1.5  
## v forcats    1.0.0      v stringr    1.5.1  
## v ggplot2    3.5.2      v tibble     3.2.1  
## v lubridate  1.9.4      v tidyr      1.3.1  
## v purrr      1.0.4  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()     masks stats::lag()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
ggplot(
  data = mpg, # From dTA
  mapping = aes(x = displ, y = hwy) # Add comma between arguments
) + # From mapping
  geom_point() + # Moved aesthetics used in geom_smooth to ggplot call
  geom_smooth(method = "lm") # Closed quotation marks
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



3. Press Option + Shift + K, what happens? Get keyboard shortcut quick reference page To get same page from menus, click help on the menubar and it is right under cheat sheets

4. Revisit an exercise from 1.6, run the following lines of code

```
my_bar_plot <- ggplot(mpg, aes(x = class)) +
  geom_bar()
my_scatter_plot <- ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point()
ggsave(filename = "mpg-plot.png", plot = my_bar_plot)
```

```
## Saving 6.5 x 4.5 in image
```

Which of the two plots is saved and why? The bar plot is saved because while ggsave() by default uses the last created plot, you can specify which plot to save with the plot parameter