

hw_2

Andrew Mikolinski

2023-09-24

```
library(dplyr) library(tidyverse)
data("mtcars")
```

Question 1

This code appears to be attempting a subset, the proper code for a subset is as follows

```
mtcars[mtcars$cyl < 6, ]
```

Avoid the -x:y situation, theres a few options for fixing this depending on what you're trying to accomplish but this is one of them

```
mtcars[-1:0 & 0:3, ]
```

This command just needed a second "=" sign

```
mtcars[mtcars$cyl == 8, ]
```

This command needed more specificity because it was basically saying "either 4 cylinders or a 6 anywhere in the data" so if you're looking for either 4 or 6 cylinders this is the code for it

```
mtcars[mtcars$cyl == 4 | mtcars$cyl == 6, ]
```

Question 2

NA Represents a missing value, none of the values in `x = 1:5` are missing so the command `x[NA]` will return 5 NAs.

```
x = 1:5 x[NA]
```

Question 3

This returns an error because without the comma the command doesn't let you keep all rows or columns in the matrix/array. The comma tells R that you're looking for just rows 1-15, if you were to plug in a value after the comma like 7, R would return rows 1-15 and only column 7. In a multi dimensional array/matrix, the commas specify the values for specific dimensions that you're looking for.

```
mtcars[1:15]
```

Question 4

The first line of this code sets up a matrix comprised of 1, 2, 3, 5, 6, 7, and 3 missing values, and tells R that the values should be split into 3 rows in the order in which each of the values appear in the code.

The second line is checking if there are any missing values in the provided matrix. Normally, without the "`= 0`" this would return 3 NA values, and the command "`is.na(x)`" would return a 3 row 3 column table with 6 "False" and 3 "True" values. With "`= 0`", you're telling R that even though there's a few missing values in the set it should not consider them missing values. So now if you put the command `is.na(x)` it will return a table with 3 rows and 3 columns and all of them will say "False".

```
x = matrix(c(1:3, NA, 5:7, NA, NA), nrow = 3) x[is.na(x)]
```

Question 5

```
data("mtcars") mtcars$mpg$mpg_2 <- ifelse(mtcars$mpg < 16, "Low", ifelse(mtcars$mpg >= 16 | mtcars$mpg < 21, "Low_intermediate", ifelse(mtcars$mpg >= 21 | mtcars$mpg < 26, "Intermediate_high", ifelse(mtcars$mpg >= 26, "High")))) mpg_2
```