# Week 2 Exercise Solutions

## Brad McNeney

1. Predict the outcome of the following:

```
c(1, FALSE)
c("a",1)
c(list(1),"a")
c(TRUE,1)
```

*Solution* You can see what they return by typing each line into the R console. The key point is that combining different data types coerces the less flexible type to the more flexible type. For atomic vectors, logicals are least flexible, then numeric, then character. As vectors, list is more flexible than atomic.

2. If `vv <- list(a=1,b=2)`, why doesn't `as.vector(vv)` work to coerce vv into an atomic vector?

*Solution* vv is already a vector

3. What do `dim()`, `nrow()` and `ncol()` return when applied to a 1-dimensional vector? What about `NROW()` or `NCOL()`?

*Solution* Try `x <- c(1,2,3); dim(x); nrow(x); ncol(x)` to see that each returns NULL, because a 1-d vector has no dim attribute. NROW and NCOL treat a vector as a 1-column matrix.

4. What is `dim(cbind(A,A))` if A = `matrix(1:4,2,2)`?

*Solution* Type into the R console to find out. The cbind creates a 2x4 matrix by putting 2 copies of A side-by-side.

5. What do the following return? Understand why.

```
TRUE | FALSE
c(TRUE,TRUE) & c(FALSE,TRUE)
c(TRUE,TRUE) && c(FALSE,TRUE)
```

*Solution* The key here is that `&` is vectorized while `&&` only checks the first element of the two vectors.

6. What sort of object does `table()` return? What is its type? What attributes does it have? How does the dimensionality change as you tabulate more variables?

*Solution* An array with class attribute "table". The array is of type int. The dimension increases with each variable. Tabulating 2 variables returns a 2-d array (matrix), 3 variables returns a 3-d array, etc. Try the following in R

```
tt <- table(c(1,1,2,2,3,3),c(1,1,1,3,3,3),c(1,1,1,1,2,2))
tt
str(tt)
```

7. What happens to a factor when you modify its levels? How do f2 and f3 differ from f1?

```
f1 <- factor(letters)
levels(f1) <- rev(levels(f1))
f2 <- rev(factor(letters))
f3 <- factor(letters, levels = rev(letters))
```

*Solution* The labels on the underlying integer values change, but the integer values themselves don't. Try the following:

```
f1 <- factor(letters)
f1
str(f1)
levels(f1) <- rev(levels(f1))
f1
str(f1)
f2 <- rev(factor(letters))
f2
str(f2)
f3 <- factor(letters, levels = rev(letters))
f3
str(f3)
 data.frame(f1,as.numeric(f1),f2,as.numeric(f2),f3,as.numeric(f3))
```

You can see that f2 and f1 print the same but the integer-to-label correspondence is backwards. Conversely, f3 and f3 have the same integer-to-label correspondence, but the labels are in reverse order.

8. Fix each of the following common data frame subsetting errors:

```
mtcars[mtcars$cyl = 4, ]
mtcars[-1:4, ]
mtcars[mtcars$cyl <= 5]
mtcars[mtcars$cyl == 4 | 6, ]
```

*Solution*

```
mtcars[mtcars$cyl == 4,]
mtcars[-(1:4),]
mtcars[mtcars$cyl <= 5,]
mtcars[mtcars$cyl ==4 | mtcars$cyl == 6 | ]
```

9. Consider the following data frame and tibble. Suppose you want to extract the number from the first row and first column and multiply this number by the vector `1:3`. How would you do this for the data frame? How would you do this for the tibble?

```
dd <- data.frame(x=1:3,y=c("a","b","c"))
library(tibble)
tt <- as.tibble(dd) # Note: we are supposed to use as_tibble() now
```

```
## Warning: 'as.tibble()' was deprecated in tibble 2.0.0.
## Please use 'as_tibble()' instead.
## The signature and semantics have changed, see '?as_tibble'.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.
```

*Solution*

```
dd[1,1]*(1:3)
```

```
## [1] 1 2 3
```

```
as.numeric(tt[1,1])*(1:3)
```

```
## [1] 1 2 3
```

10. Read the help file `help("%in%")`. Copy the following data frame into R.

```
students <- data.frame(
        major=c("Stat","Stat","Econ","Bus","Phys","ActSci","ActSci"),
        year=c(3,4,4,5,3,3,4))
```

Use `%in%` to create a logical vector of students who major in Stat or ActSci and use this to subset `students` to Stat or ActSci majors.

*Solution*

```
students[students$major %in% c("Stat","ActSci"),]
```

```
##     major year
## 1    Stat    3
## 2    Stat    4
## 6 ActSci    3
## 7 ActSci    4
```