For Loops in R

Additional reading/source for this handout

https://r4ds.had.co.nz/iteration.html

The setting: we want to repeat a task many times

Two general approaches:

1. imperative (`for` loops and similar, the topic of this handout)
2. functional (`map` and related functions from `purrr` package)
    - write a function that does the task once
    - `map` (or similar) calls that function many times and assembles the results

R is a functional programming language, the functional paradigm is a natural fit. (R also has systems for object-oriented programming.)

Often, loops are a perfectly good way of doing things.

Overall structure

```
1. Allocate space where results will be stored, if necessary
2. for(object in vector_of_objects) {
    a. do some stuff based on object
    b. potentially, save some results in the space allocated in step 1
}
```

Example: Mean of each variable in a data frame

The `mtcars` data set comes with R and contains observations of 11 numeric variables for 32 cars.

```r
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```r
dim(mtcars)
```

```
## [1] 32 11
```

```r
ncol(mtcars)
```

```
## [1] 11
```

Let's find the mean of each of those variables.

Pseudo Code:

```
1. Allocate space to store the column means
2. For each column in the mtcars data frame,
    a. Find the mean for that column
    b. Store the column mean in the appropriate entry of the space allocated in step 1.
```

The point here is to write down the steps using not-quite-code so that you have thought through what your code will do before you start trying to code. The above is probably a minimal level of detail for pseudo code; you could add in some more detail if you want. Below are three ways of actually implementing the algorithm above in R.

Approach 1:

```r
var_means <- vector("numeric", ncol(mtcars)) # 1. allocate space for results
for(i in seq_len(ncol(mtcars))) { # 2. set up for loop
  var_means[[i]] <- mean(mtcars[[i]]) # a. and b. do something based on i, save results
}

var_means
```

```
##  [1]  20.090625   6.187500 230.721875 146.687500   3.596563   3.217250
##  [7]  17.848750   0.437500   0.406250   3.687500   2.812500
```

Approach 2:

Basically the same thing, a few details changed:

```r
#1. allocate space to store results
var_means_df <- data.frame(
  var_name = rep(NA, ncol(mtcars)),
  var_mean = rep(NA, ncol(mtcars))
)

for(i in seq_along(mtcars)) { # 2. set up for loop
  var_means_df$var_name[i] <- colnames(mtcars)[i] # a. and b. do something based on i, save results
  var_means_df$var_mean[i] <- mean(mtcars[[i]]) # a. and b. do something based on i, save results
}

var_means_df
```

```
##    var_name   var_mean
## 1       mpg  20.090625
## 2       cyl   6.187500
## 3      disp 230.721875
## 4        hp 146.687500
## 5      drat   3.596563
## 6        wt   3.217250
## 7      qsec  17.848750
## 8        vs   0.437500
## 9        am   0.406250
## 10     gear   3.687500
## 11     carb   2.812500
```

Approach 3

More of the same!

```
#1. allocate space to store results
var_means_df_2 <- data.frame(
  var_name = colnames(mtcars),
  var_mean = rep(NA, ncol(mtcars))
)

for(var_name in colnames(mtcars)) { # 2. set up for loop
  save_ind <- which(var_means_df_2$var_name == var_name) # a. and b. do something based on i, save results

  var_means_df_2[save_ind, "var_mean"] <- mean(mtcars[[var_name]]) # a. and b. do something based on i, save
}

var_means_df_2
```

```
##    var_name   var_mean
## 1       mpg  20.090625
## 2       cyl   6.187500
## 3      disp 230.721875
## 4        hp 146.687500
## 5      drat   3.596563
## 6        wt   3.217250
## 7      qsec  17.848750
## 8        vs   0.437500
## 9        am   0.406250
## 10     gear   3.687500
## 11     carb   2.812500
```