

Lab 4: Functions (Solutions)

Jacob J. Coutts

Question 1

a.) Write a function called `convert_temps()` that converts temperatures between Fahrenheit and Celsius in both directions.

```
convert_temps <- function(temp, unit){
  if(is.numeric(temp) == FALSE & is.integer(temp) == FALSE){
    stop("You must enter a number for temp.")
  } else{
    if(unit == "F"){
      new_temp = (temp - 32)*(5/9)
      new_unit = "C"
    } else if (unit == "C"){
      new_temp = temp*(9/5)+32
      new_unit = "F"
    } else{
      stop("You may only enter 'C' for Celsius or 'F' for Fahrenheit.")
    }
  }

  return(paste("You entered ", temp, unit, ". Your new temperatures is, ", new_temp, new_unit))
}
```

b.) Test your function with 0 C, 100 C, 0 F and 100 F.

```
convert_temps(temp = 0, unit = "C")
```

```
[1] "You entered 0C. Your new temperatures is, 32F"
```

```
convert_temps(temp = 100, unit = "C")
```

```
[1] "You entered 100C. Your new temperatures is, 212F"
```

```
convert_temps(temp = 0, unit = "F")
```

```
[1] "You entered 0F. Your new temperatures is, -17.7777777777778C"
```

```
convert_temps(temp = 100, unit = "F")
```

```
[1] "You entered 100F. Your new temperatures is, 37.7777777777778C"
```

c.) Already done above

d.) Test your new function with the following, and for the cases where you get an error message explain in a sentence or two below the code chunk why it didn't work (you may need to retype the quotation marks because of formatting differences between Word and R).

```
convert_temps(55, "F")
```

```
[1] "You entered 55F. Your new temperatures is, 12.7777777777778C"
```

```
convert_temps(unit = "C", temp = 20)
```

```
[1] "You entered 20C. Your new temperatures is, 68F"
```

```
#convert_temps("C", 20)
# arguments in wrong order
#convert_temps(32, "Celsius")
# unit not specified correctly
#convert_temps(85, F)
# F not a string
```

Question 2

Write a function called **col_na** that checks whether each column of a data frame has missing values. It should have one argument, **df**, that is a data frame input. There should be a check to make sure that **df** is a data frame. If not, generate the error, "Input must be a data frame."

If there is missing data, you should print the column number and how many values are missing. For instance, "Column 1 has two missing value(s), Column 3 has one missing value(s)" and so on. (NOTE: There are a few approaches here. `NA %in% object_name` will tell you whether there is at least one missing value in an object. `is.na(vector_name)` will print a vector of TRUE/FALSE for whether every element of a vector is missing (i.e., NA) or not. Hint: You will need to loop through the columns one by one; consider using a `paste()` statement to get the output described above. Test the function on the data frame below:

```
mis_check <- data.frame(scores = c(81,83,95, NA, 100), ID = c("a1","a2","a3","a4","a5"), height = c(170,175,180,185,190))

col_na <- function(df){
  if(class(df) == "data.frame"){
    for(i in 1:ncol(df)){
      if(sum(is.na(df[,i])) > 0){
        print(paste("Column", i, "has", sum(is.na(df[,i])), "missing value(s)."))
      }
    }
  } else{
    stop("Input must be a data frame.")
  }
}

col_na(df = mis_check)
```

```
[1] "Column 1 has 1 missing value(s)."
```

```
[1] "Column 3 has 2 missing value(s)."
```

End of Script