Unit 1 Cheat Sheet

General Tips:

- `Ctrl` + `/` (Windows) or `Cmd` + `/` (Mac) Comments out multiple lines of code at once.
- `?any_function` Look up documentation for any function.
- `?dplyr::arrange` Looks up arrange() function in dplyr package. Can substitute dplyr with other package names.
- (Windows only) If cursor changes from a vertical line to a blinking box/block, the editor has switched from insert mode to overtype mode. Hit `Insert` or `Ins` key to get back to normal.

Variables:

- `variable_name = value` or `variable_name -> value` assigning variables.
- `paste0()` Converts its arguments to character strings and concatenates them.
- `typeof()` Returns the type of an object.
- `is.character()` Returns `TRUE` if the provided object is a character. Otherwise, returns `FALSE`.
- `is.integer()` Returns `TRUE` if the provided object is an integer. Otherwise, returns `FALSE`.
- `is.numeric()` Returns `TRUE` if the provided object is numeric. Otherwise, returns `FALSE`.
- `is.logical()` Returns `TRUE` if the provided object is a logical value. Otherwise, returns `FALSE`.
- `as.character()` Returns the argument as a character.
- `as.integer()` Returns the argument as an integer.
- `as.double()` Returns the argument as a double.
- `as.numeric()` Returns the argument as a numeric value.
- `as.logical()` Returns the argument as a logical value.

Math:

- `+` Addition.
- `-` Subtraction.
- `*` Multiplication.
- `/` Division.
- `^` Exponentiation.
- `%%` Modulus, remainder after division.
- `%/%` Quotient, quotient after division.
- `ceiling(x)` Returns the smallest integer that is greater than or equal to x.
- `floor(x)` Returns the largest integer that is less than or equal to x.
- `round(x, n)` Rounds the values in the first argument x to the specific number of decimal places, n (default is 0).
- `sqrt(x)` Returns the square root of x.

- `log(x, base)` Returns the logarithm of x, by default the base is *e*.
- `abs(x)` Returns the absolute value of x.
- `log2(x)` Returns the logarithm of x with base 2.

Descriptive Statistics

- Syntax for all following functions: `function(x, na.rm = TRUE)` by default `na.rm` is set to `FALSE`
- `sum()` Returns the sum of the elements in the vector.
- `min()` Returns the minimum element in the vector.
- `max()` Returns the maximum element in the vector.
- `mean()` Returns the mean of the elements in the vector.
- `median()` Returns the median of the elements in the vector.
- `var()` Returns the variation of the elements in the vector.
- `sd()` Returns the standard deviation of the elements in the vector.

Vectors:

- `c()` Concatenates the arguments into a vector.
- `seq(x, y, n)` Returns a vector from x to y incrementing by n.
- `rep(x, n)` Returns a vector with x repeated n times.
- `length(x)` Returns the length of the vector x.
- `sort(x)` Returns x in sorted order.
- `table(x)` Returns a table with the elements in x and their frequencies.
- `unique(x)` Returns all the unique elements in x.
- `intersect(x, y)` Returns the elements that are in both x and y.
- `setdiff(x, y)` Returns the elements that are in x but not in y.
- `x[4]` Selects the fourth element.
- `x[2:4]` Selects elements two to four.
- `x[c(1,5)]` Selects elements one and five.
- `x[-4]` Selects all elements but the fourth.
- `x[x < 0]` Selects all elements that meet the given condition. In this example, selects all elements less than 0.
- `x[x %in% c(1,2,5)]` Selects elements that are in the set 1, 2, 5.

Lists:

- `list(name1 = value1, name2 = value2)` Creates a list.
- `my_list[[name1]]` Retrieves the value that corresponds to the given name.
- `my_list$name1` Retrieves the value that corresponds to the given name.
- `names(my_list)` Returns all names in the given list.
- `my_list$name1 = NULL` - Removes the given name and its corresponding value from the list.

Programming:

If statements

```
if (condition) {
    <Do something>
} else if (other condition) {
    <Do something different>
} else {
    <Do something else>
}
```

Relational Operators

- `>` Greater than.
- `<` Less than.
- `>=` Greater than or equal to.
- `<=` Less than or equal to.
- `==` Are equal.
- `!=` Not equal.
- `|` Or.
- `&` And.
- `5 %in% c(1, 2, 5)` Returns `TRUE` if the value is in the vector. Otherwise, returns `FALSE`. In this example, it returns `TRUE`.
- `any(my_vector < value)` Returns `TRUE` if any element in the vector meets the given condition. Otherwise, returns `FALSE`.
- `all(my_vector < value)` Returns `TRUE` if all elements in the vector meet the given condition. Otherwise, returns `FALSE`.
- `which(my_vector < value)` Returns the indices at which the elements in the vector meet the given condition.
- `is.na()` Returns `TRUE` if the argument is `NA`. Otherwise, returns `FALSE`. If a vector is given as the argument, returns a vector with a logical value for each element in the given vector. `TRUE` if the element is `NA` and `FALSE` otherwise.

This document was created by Aidan Cardall, Ethan Bang, and Megan Rodabough at Brigham Young University.