Unit 2 Cheat Sheet

Tibbles:

- `data.frame(col1 = vector1, col2 = vector2)` - creates and returns a data frame from the given vectors
- `nrow()` - returns the number of rows in a data frame
- `ncol()` - returns the number of columns in a data frame
- `dim()` - returns the number of rows and columns in a data frame
- `colnames()` - returns the names of the columns in a data frame
- `head(n = 5)` - returns the specified number of rows from the beginning of a data frame
- `tail(n = 5)` - returns the specified number of rows from the end of a data frame
- `rbind(data_frame, new_vec)` - adds an extra row or rows onto a data frame
- `data.frame$col1 –` dollar sign used to access columns of data frames, can be used to modify, add, and remove columns
- `as.data.frame(tibble)` - converts a tibble to a data frame
- `tibble(col1 = vector1, col2 = vector2)` - creates and returns a tibble from the given vectors
- `read_csv(file_name)` - reads in a csv file as a tibble
- `read_tsv(file_name)` - reads in a tsv file as a tibble
- `write_csv(tibble, file_name)` - writes a tibble as a csv file to the specified path
- `write_tsv(tibble, file_name)` - writes a tibble as a tsv file to the specified path

Visualizing data:

- `ggplot(data, aes(x=col1, y=col2, color=col3, fill=col4, size=col5))` - create a ggplot using a tibble, using columns from the tibble as data to populate the graph
- `geom_point()` - creates a scatterplot
- `geom_bar()` - creates a one variable bar plot
- `geom_col()` - creates a two variable bar plot
- `geom_histogram()` - creates one variable histogram
- `geom_point()` - creates a scatterplot
- `geom_density()` - creates a density plot
- `geom_line()` - creates a two variable line graph
- `geom_boxplot()` - creates a box plot
- `labs(x="n1", y="n2", color="n3", size="n4")` - renames the labels of graph attributes
- `facet_wrap(col5)` - creates panels of the graph based subsections of the data
- `theme_bw()` - changes the color scheme of the background to black and white

Transforming data:

- `filter(data, col1 >= value)` - eliminates rows from a tibble on the given conditional
- `arrange(data, col1)` - sorts rows in a tibble in ascending order
- `arrange(data, desc(col1))` - sorts rows in a tibble in descending order
- `select(data, col1, col2)` - keeps only the specified columns in the tibble
- `mutate(data, new_col = col1 * col2)` - creates a new column in the tibble
- `mutate(data, col1 = col1 ^ 2)` - modifies an existing column in the tibble
- `group_by(data, col1) %>%`
- `summarize(mean_col = mean(col2))` - group_by() and summarize() are always used in conjunction to create a summary statistic for the tibble based on the specified column in group_by()
- `rename(new_name = old_name)` - renames a column in a tibble
- `vector = pull(data, col1)` - pulls out a column from a tibble as a vector

Strings:

- `str_detect(string, "substring")` - returns a boolean value to detect presence of a substring in a given string
- `str_starts(string, "starting")` - returns boolean value to detect presence of a substring at the beginning of a given string
- `str_ends(string, "ending")` - returns boolean value to detect presence of a substring at the end of a given string
- `str_length(string)` - returns the character length of a string
- `str_to_lower(string)` - converts each character in the string to lower case
- `str_to_upper(string)` - converts each character in the string to upper case
- `str_to_title(string)` - converts the first character in each word to upper case, and the rest to lower case
- `str_replace_all(string, "original", "replace")` - replaces each substring with a replacement string
- `str_replace_na(string, "replace")` - replaces all NA values with a replacement string
- `str_sub(string, 3, 6)` - returns the substring between the passed indices

Factors:

- `factor(vector, levels = c("level1", "level2", "level3"))` - creates a factor from the vector with specified levels
- `combined_factor = fct_c(factor1, factor2)` - combines two factors
- `fct_recode(factor, "new_level1" = "level1", "new_level2" = "level2)` - renames the levels in a factor
- `vector = as.vector(factor)` - converts a factor to a vector
- `factor[1:3]` – subset first 3 levels of a factor

Tidy data:

- `separate(data, existing_column, into = c("new_col1", "new_col2"), sep = "-", convert = False)` - separates a column into two columns on a given character or substring
- `unite(data, new_col, old_col1, old_col2, sep = "-")` - combines two ot more columns into a single column
- `pivot_longer(data, col1:col4, names_to = "new_col1", values_to = "new_col2")` - pivots the tibble such that the specified columns are converted to a single column, creating more rows in the tibble
- `pivot_wider(data, names_from = "col1", values_from = "col2")` - pivots the tibble such that the values from the specified columns are converted to multiple columns, creating more columns in the tibble

Relational data:

- Syntax for all joins: `join(tibble1, tibble2, by = c("col1" = "col2"))`
- `inner_join()` – only keeps rows present in both tibbles
- `left_join()` – only keeps rows present in the left tibble
- `right_join()` – only keeps rows present in the right tibble
- `full_join()` – keeps all rows from both tibbles
- `semi_join()` – only keeps rows present in both tibbles, and only columns from the left tibble
- `anti_join()` – only keeps rows that do not overlap between the two tibbles, and only columns from the left tibble