Summary Sheet

AUTHOR
Julia Gallucci

Class 3

Factors:

- `factor()`
- type of data object used to represent categorical variables.
- useful when working with data that has a limited number of distinct values or levels that can be ordered (i.e., Sex, education level, any rating scale values etc.)
- `fct_recode()` from the forcats library used to change factor levels by hand

Dates:

- dates can be represented and manipulated using the `lubridate` library i.e.,
    - `today()` to get today's date
    - `now()` to get today's date and time
    - can extract specific components : `year(), month(), hour(), minute()` ...
    - can get specific time spans: `as.duration()`

Missing data:

Detect missing data using the `is.na()` function

Data manipulation:

- `glimpse()` to view your data as columns down the page, rows across; easier to view entire columns in data set
- `filter()` to subset data set and retain only those columns that meet a certain condition
    - i.e., `filter(data_set, certain_column < certain value)`
- `arrange()` to order rows in a data set by a specific column, default as ascending order, use `-` for descending

- i.e., `arrange(data_set, -certain_column)` to order descending
- i.e., `arrange(data_set, certain_column)` to order ascending

- `select()` to extract only a specified column from data set, can use `-` ro remove a certain column from data set
    - i.e., `select(data_set, certain_column)` to pick only that column
    - i.e., `arrange(data_set, -certain_column)` to pick all columns BUT that one
- `mutate()` to create a new column from existing columns or modify an existing column
    - i.e., `mutate(data_set,new_column_name = existing_column_name + some value)`

Pipe operator

`%>%` is used to combine multiple operations at once

i.e., `data_set %>%`

```
filter(``certain_column < certain value``) %>%

arrange(certain_column) %>%

select(certain_column)
```

- filter the data set to only view rows with a specific column under a specific value
- arrange the data set to view rows of a certain column in ascending
- select only specified columns to extract from data set

Data summary:

- `summary()` provides an overview of data; i.e., central tendencies, dispersion and distribution
- `pull()` to extract a specific column from a data frame as a numeric vector
    - can combine `pull` with a mathematical operation

i.e., `data_set %>%`

```
pull(certain_column) %>%

mean() #can also be things like median, variance, sd etc.
```

Note: use na.rm = TRUE when column observations contain NA

- `summarise()` to create a new data table summarizing observations
    - i.e., `data_set %>%`

      ```
      summarise(name_column = mean(certain_column),

      name_column2 = sd(certain_column))
      ```

- can also `group_by()` prior to summarizing to get a summary table based on categories
    - i.e., `data_set %>%`

      `group_by(certain_categorical_column) %>% summarise(n_column = n(),`

      `name_column = mean(certain_column),`

      `name_column = sd(certain_column))`