STAT112 - Exam 2 Sheet

Joining:

Factors:

- functions for changing the **order** of factor levels
  - `fct_relevel()` = *manually* reorder levels
  - `fct_reorder()` = reorder levels according to values of another *variable*
  - `fct_infreq()` = order levels from highest to lowest frequency
  - `fct_rev()` = reverse the current order
- functions for changing the **labels** or values of factor levels
  - `fct_recode()` = *manually* change levels
  - `fct_lump()` = *group together* least common levels

Strings:

`str_replace(x, pattern, replacement)` - finds the *first* part of x that matches the pattern and replaces it with replacement

`str_replace_all(x, pattern, replacement)` - finds *all* instances in x that matches the pattern and replaces it with replacement

`str_to_lower(x)` - converts all upper case letters in x to lower case

`str_sub(x, start, end)` - only keeps a subset of characters in x, from start (a number indexing the first letter to keep) to end (a number indexing the last letter to keep)

`str_length(x)` - records the number of characters in x

`str_detect(x, pattern)` - is TRUE if x contains the given pattern and FALSE otherwise

Data Import:

- Using mac follow this format

```r
```{r}
library(tidyverse)
 imdb_messy <- read_csv("../data/imdb_5000_messy.csv")
```
```

Function	Data file type
`read_csv()`	.csv - you can save Excel files and Google Sheets as .csv
`read_delim()`	other delimited formats (tab, space, etc.)
`read_sheet()`	Google Sheet
`st_read()`	spatial data shapefile

STAT112 - Exam 2 Sheet

Wrangling:

`arrange` - arrange the *rows* according to some *column*

`filter` - filter out or obtain a subset of the *rows*

`select` - select a subset of *columns*

`mutate` - mutate or create a *column*

`summarize` - calculate a numerical summary of a *column*

`group_by` - group the *rows* by a specified *column*

```{r}
# Keep only data on counties in Hawaii
elections_small |>
  filter(state_name %in% c("Hawaii"))
```

```{r}
# Arrange the counties in elections_small from lowest to highest percentage of 2020
Democrat support
# Show the top ten rows of that
elections_small |>
  arrange(repub_pct_20) |>
  head()
```

```{r}
elections_small |>
  filter(state_name=="Minnesota") |>
  select(county_name, dem_pct_20) |>
  arrange(desc(dem_pct_20))
```

```{r}
# What does this do?
# Shows the summary of the repub_pct_2 column.
elections_small |>
  summarize(median(repub_pct_20))
```

```{r}
# What did this code do?
#It created a new column called diff_20 that was the repub_pct_20 (-) the dem_pct_20. It
shows the difference between the columns.
elections_small |>
  mutate(diff_20 = repub_pct_20 - dem_pct_20) |>
  head()
```

```{r}
elections_small |>
  group_by(state_name)
```

Dates:

symbol	meaning
==	equal to
!=	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
%in% c("???", "???")	a list of multiple values

Shaping:

```{r}
# NOTE the use of is.na()
#Shoes number that are missing body mass.
penguins |>
  summarize(sum(is.na(body_mass_g)))
```

In the *very rare case* that we need complete information on every variable for the specific task at hand, we can use `na.omit()` to get rid of *any* penguin that's missing info on *any* variable:

```{r}
penguins_complete <- penguins |>
  na.omit()
```

- `-` *removes* a given variable and keeps all others (e.g. `select(-island)`)
- `starts_with("___")`, `ends_with("___")`, or `contains("___")` selects only the columns that either start with, end with, or simply contain the given string of characters

```
# A tibble: 6 × 8
  species island    bill_length_mm bill_depth_mm flipper_l<
  <chr>   <chr>              <dbl>         <dbl>
1 Adelie  Torgersen           39.1          18.7
2 Adelie  Torgersen           39.5          17.4
3 Adelie  Torgersen           40.3          18
4 Adelie  Torgersen           NA            NA
5 Adelie  Torgersen           36.7          19.3
6 Adelie  Torgersen           39.3          20.6
# ℹ 2 more variables: sex <chr>, year <dbl>
```

v/s

```
# A tibble: 6 × 3
# Groups:   species [3]
  species   sex    avg_body_mass
  <chr>     <chr>          <dbl>
1 Adelie    female         3369.
2 Adelie    male           4043.
3 Chinstrap female         3527.
4 Chinstrap male           3939.
5 Gentoo    female         4680.
6 Gentoo    male           5485.
```

pivot_longer()
pivot_wider()

```
sleep_wide |>
  pivot_longer(cols = starts_with("day"), names_to = "day", values_to = "reaction_time")
```

```{r}
 sleep_long |>
   pivot_wider(names_from = day, values_from = reaction_time, names_prefix = "day") |>
   head()
```