Theme 1: Data Wrangling Essentials - Assignments

Libraries

Note: if you haven't installed the packages yet, remove the `#` sign before `install.packages()` function. Packages need to be installed ***only once***, but they need to be ***activated*** with `library(packagename)` each time you start a new R session or restart RStudio.

```
# install.packages("tidyverse")
# install.packages("here")

library(tidyverse)
library(here)
```

Import the Data

For this time, I'll give the needed piece of code to import your data to R environment. However, in future exercises, I expect you to know how to import the data to R *by yourself* (generally speaking, the amount of "spoon-feeding" i.e. readymade code decreases when we proceed with the themes).

```
vdem <- here("vdem_subset.csv") %>%
  read.csv()
```

Assignment 1: Explore the data

Take your first look at the data with the `glimpse()` function. Answer the following question:

Task 🧐

How many variables and observations there are in the `vdem` data?

Support code 👩‍💻

This time the code is given to you. Just run the piece of code below and answer the question above.

```
# Print the first 6 observations with the following code
vdem %>%
  glimpse()
```

Answer: There are ___ variables and ___ observations in the `vdem` data.

Assignment 2: Subset the data

Task 🧐

Subset the `vdem` data to include only the variables `country_name`, `year`, and `v2x_libdem` and observations from year 2022. Create a new data frame for these data and name it as `vdem_libdem_2022`

Support code 👩‍💻

- Subset the `vdem` data by selecting only the variables `country_name`, `year`, and `v2x_libdem` with the `select()` function.
- Next, use `filter()` to include only observations from the year 2022 (NB: remember to use `%>%` operator!).
- Assign i.e. name this new data as "`vdem_libdem_2022`. Recall from the lecture slides and example materials the use of `<-` operator!

```
vdem_libdem_2022 <-
  vdem %>%
  select(___, ___, ___) %>%
  filter(year == "___")
```

Assignment 3: Explore the Data

Task 🧐

Find out which five countries score highest on liberal democracy index (column: `v2x_libdem`) in the `vdem_libdem_2022` data frame.

Support code 👩‍💻

You can use `arrange()` and `desc()` functions for this.*

```
vdem_libdem_2022 %>%
  arrange(desc(___))
```

Answer: top five countries on liberal democracy index are…

1.
2.
3.
4.
5.

Task 🧐

Find out the number of unique liberal democracy index scores in `vdem_libdem_2022` data frame.

Support code 👩‍💻

You can use the `distinct()` to explore how many unique `v2x_libdem` values there are in the `vdem_libdem` data:

```
vdem_libdem_2022 %>%
  distinct(___)
```

Answer: there are ___ distinct liberal democracy index scores in `vdem_libdem_2022` data.

Assignment 4: Create new columns with `mutate()`

Task 🧐

Let's imagine a following scenario: you're working in a research project dealing with comparing the state of democracy across the world. You have discovered the V-Dem dataset and want to use the liberal democracy index data for year 2022 in the final report of the project.

Your boss likes the idea, but wants this scale to vary from **0 to 10** *instead of* **0 to 1**, *as in the original V-Dem data because other indices in the final report also vary from 0 to 10. Create a new variable `libdem_0_to_10` that captures the liberal democracy index, but ranging from 0 to 10 instead of 0 to 1.*

Support code 👩‍💻

The formula for creating a new liberal democracy index variable is `10 * v2x_libdem`. Create a new variable using this formula within `mutate()` function, and name it as `libdem_0_to_10`. Finally, arrange this new `libdem_0_to_10` variable in descending order:

```
vdem_libdem_2022 %>%
  mutate(libdem_0_to_10 = ___) %>%
  arrange(desc(libdem_0_to_10))
```

Compare the original `v2x_libdem` index with the new `libdem` index. Does your new variable make sense to you? No need to write the answer, you may just think it.

Assignment 5: Summarise data with `summarise()`

Task 🧐

Calculate the mean and median for liberal democracy index in 2022.

Support code 👩‍💻

Use `summarise()` function together with `mean()` and `median()` functions to calculate the mean and median for liberal democracy index `v2x_libdem` in `year` 2022.

```
vdem %>%
  filter(year == "___") %>%
  summarise(mean_libdem = mean(___),
            median_libdem = median(___))
```

Answer:

- The global mean liberal democracy index in 2022 was ___
- The global median liberal democracy index in 2022 was ___

Task 🧐

Calculate minimum and maximum years present in the `vdem` data.

Support code 👩‍💻

This time, use `summarise()` function together with `min()` and `max()` functions to calculate the earliest and most recent `year` with observations:

```
vdem %>%
  summarise(min_year = min(___),
            max_year = max(___))
```

Answer here: the first year (i.e. minimum value) with observations in the `vdem` data is ___ and the last year with observations is ___ .

Assignment 6: calculate by groups with `group_by()`

Task 🧐

Calculate mean liberal democracy index, `v2x_libdem`, by countries. Find out which are the top and bottom three countries on liberal democracy index.

Support code 👩‍💻

- Start with the `vdem` data frame
- Use `group_by()` to group following calculations by `country_name`
- Use `summarise()` and `mean()` to calculate `mean_libdem`
- Arrange the `mean_libdem` column in descending order with `arrange()` and `desc()`
- Remove the missing values with `drop_na()` (you can just leave it as it is in the given code)
- Assign these results as a new `data frame` called `vdem_mean_libdem_historical`

Finally, take a look at the top and bottom 6 observations in `vdem_mean_libdem_historical` data frame with `head()` and `tail()` functions, and answer these questions:

Answer: The top 3 liberal democracies in `vdem_mean_libdem_historical` are …, …, and … Answer: The 3 least liberal democratic countries in `vdem_mean_libdem_historical` are …, …, and …

```
vdem_mean_libdem_historical <-
  vdem %>%
  group_by(___) %>%
  summarise(mean_libdem = mean(___)) %>%
  arrange(desc(mean_libdem)) %>%
  drop_na()

vdem_mean_libdem_historical %>%
  head()

vdem_mean_libdem_historical %>%
  tail()
```

Note: since we have many observations for same countries from different years, we are dealing here with "overall" historical means of liberal democracy in a global scope.

Assignment 7: count unique values of `country_name` with `count()`

Task 🧐

Find out which six countries have the least observations in the whole `vdem` data.

Support code 👩‍💻

You can count unique values for desired variable with `count()`. Here, you want to use `country_name` within the `count()` function.

```
vdem %>%
  count(___) %>%
  arrange(desc(___)) %>%
  tail()
```

Which six countries have the least observations in the whole `vdem` data?

Answer:

1. Country name 1
2. Country name 2
3. Country name 3
4. Country name 4
5. Country name 5
6. Country name 6