Exam 02

COMP/STAT112 (Spring 2025)

Name: Pablo Jimenez Section: 01

1 Instructions

- You will have **20** minutes to solve the exam on your own. After that, we will discuss the exam as a group. So, focus on the concepts not the R code.
- After class,
  1. answer the research questions using R then add the answers to your portfolio website as new page named **Food Consumption 2** under the **Best Work** section.
  2. add your *approved* summary sheet to your portfolio website as a new page named **Wrangling** under the **Summary** section. If you do not have a Summary section yet, add one first.
  3. reflect on your work in your Progress Tracker.

2 Background

You will work with the Food Consumption and CO2 Emissions dataset used in week 8 of year 2020 (Feb 2, 2020) in the TidyTuesday project. The dataset contains one file as described in the Data Dictionary below.

2.1 Data Dictionary

`food_consumption.csv`

variable	class	description
country	character	Country Name
food_category	character	Food Category
consumption	double	Consumption (kg/person/year)
co2_emmission	double	Co2 Emission (Kg CO2/person/year)

2.2 Load Packages

```
library(tidytuesdayR)
library(tidyverse)
library(rnaturalearth)  # for country boundaries
library(sf)             # for spatial viz
```

2.3 Load Data

```
tuesdata <- tt_load('2020-02-18')
fc <- tuesdata$food_consumption
```

2.4 Inspect Data

```
str(fc)
```

```
'data.frame':   1430 obs. of  4 variables:
 $ country      : chr  "Argentina" "Argentina" "Argentina" "Argentina" ...
 $ food_category: chr  "Pork" "Poultry" "Beef" "Lamb & Goat" ...
 $ consumption  : num  10.51 38.66 55.48 1.56 4.36 ...
 $ co2_emmission: num  37.2 41.53 1712 54.63 6.96 ...
```

```
head(fc, 22)
```

```
    country          food_category consumption co2_emmission
1 Argentina                   Pork       10.51         37.20
2 Argentina                Poultry       38.66         41.53
3 Argentina                   Beef       55.48       1712.00
4 Argentina             Lamb & Goat        1.56         54.63
```

5	Argentina	Fish	4.36	6.96
6	Argentina	Eggs	11.39	10.46
7	Argentina	Milk - inc. cheese	195.08	277.87
8	Argentina	Wheat and Wheat Products	103.11	19.66
9	Argentina	Rice	8.77	11.22
10	Argentina	Soybeans	0.00	0.00
11	Argentina	Nuts inc. Peanut Butter	0.49	0.87
12	Australia	Pork	24.14	85.44
13	Australia	Poultry	46.12	49.54
14	Australia	Beef	33.86	1044.85
15	Australia	Lamb & Goat	9.87	345.65
16	Australia	Fish	17.69	28.25
17	Australia	Eggs	8.51	7.82
18	Australia	Milk - inc. cheese	234.49	334.01
19	Australia	Wheat and Wheat Products	70.46	13.44
20	Australia	Rice	11.03	14.12
21	Australia	Soybeans	0.19	0.09
22	Australia	Nuts inc. Peanut Butter	8.73	15.45

2.5 Inspect country Variable

The dataframe contains information about **130** countries.

2.6 Inspect `food_category` Variable

	food_category
1	Pork
2	Poultry
3	Beef
4	Lamb & Goat
5	Fish
6	Eggs
7	Milk - inc. cheese
8	Wheat and Wheat Products
9	Rice
10	Soybeans
11	Nuts inc. Peanut Butter

The above table is a print out of the unique values of the `food_category` variable.

Which function was used to produce it?

mutate

What is the problem with these values?

They are clearly organized by groups

2.7 Fix `food_category` Variable

List the necessary wrangling functions to shorten the food category names shown in the table below. The resulting dataframe should be stored to a variable called `fcc`.

Current Name	Change to
Lamb & Goat	Lamb
Milk - inc. cheese	Diary
Wheat and Wheat Products	Wheat
Nuts inc. Peanut Butter	Nuts

1. group-by recod
2. mutate
3. Summerize
4. filter
5. str

mutate →
recode

2.8 Re-inpect `food_category` Variable

Make sure the new values of the `food_category` variable are as expected.

```
   food_category
1           Pork
2        Poultry
3           Beef
4           Lamb
5           Fish
6           Eggs
7           Milk
8          Wheat
9           Rice
10      Soybeans
11          Nuts
```

summerize
distinct
select

3 Most Consuming Countries

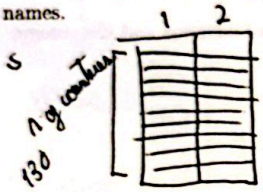
Research Question 1

Which 5 countries consume the most food are?

- Show the answer as a 2-column table and as a graph.
- Sort the countries based on consumption from largest to smallest.
- Use the cleaned dataframe.

3.1 Expected Shape

Sketch the optimal shape of the needed dataframe. Show the number of rows and columns and the column names.



3.2 Steps

List the necessary steps to produce the needed dataframe.

summerize
key

select
filter !
head or group_by (country)
arrange
desc

1. notate
2. desc
3. summerize
4. filter
5. group by
   arrange

Total

3.3 Viz

Which graph types can be used to visualize the produced dataframe? Star most appropriate one.

1. a barplot ★
2.
3.

4 Most Consuming Countries of Each Food

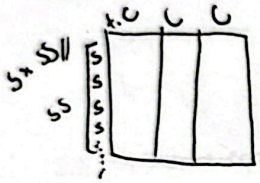
Research Question 2

Which top 5 countries consume each food are?

- Show the answer as a 3-column table (show the food category as the first column followed by country and consumption) and as a graph.
- For each food, sort the countries based on consumption from largest to smallest.
- Use the cleaned dataframe.

4.1 Expected Shape

Sketch the optimal shape of the needed dataframe. Show the number of rows and columns and the column names.



4.2 Steps

List the steps necessary to compute the required dataframe.

1. ~~desc~~ group by (food categories)
2. ~~head(5)~~ arrange (desc
3. slice max (available to slice the max)
4. relocate (to move one column)
5.

4.3 Viz

Which graph types can be used to visualize the produced dataframe? Star most appropriate one.

1. box, column ☆
2. } facet
3. }

5 Food Consumption

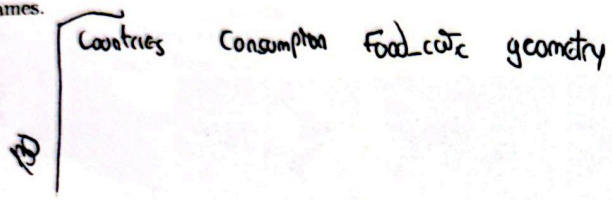
Research Question 3

What does the consumption of each food look like?

- Show a choropleth map for each food.
- Use the cleaned dataframe.

5.1 Expected Shape

Sketch the optimal shape of the needed dataframe. Show the number of rows and columns and the column names.



5.2 Steps

List the necessary steps to produce the needed dataframe.

1. ~~merge~~ boundary
2. group-by left join the datasets
3. Facet
4. ~~left_go~~
5.

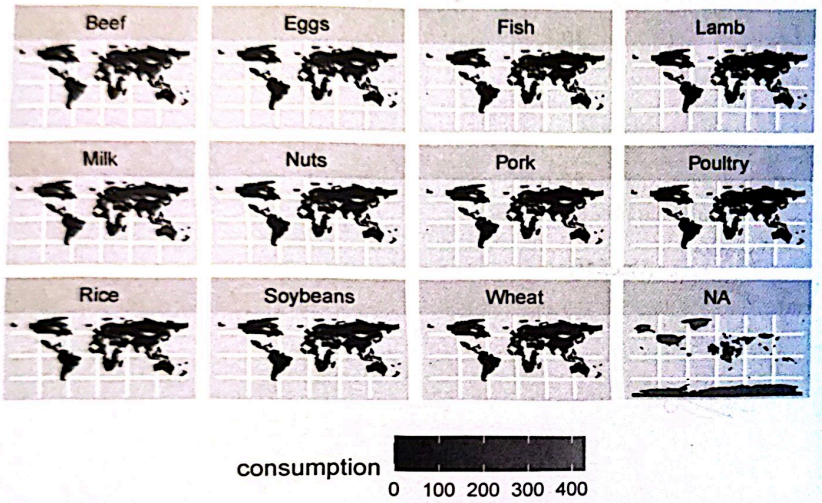
5.4 Countries with Missing Consumption Data

	name
1	Afghanistan
2	Antarctica
3	Azerbaijan
4	Benin
5	Bhutan
6	Bosnia and Herz.
7	Brunei
8	Burkina Faso
9	Burundi
10	Central African Rep.
11	Chad
12	Czechia
13	Côte d'Ivoire
14	Dem. Rep. Congo
15	Djibouti
16	Dominican Rep.
17	Eq. Guinea
18	Eritrea
19	Falkland Is.
20	Fr. S. Antarctic Lands
21	Gabon
22	Greenland
23	Guinea-Bissau
24	Guyana
25	Haiti
26	Iraq
27	Kosovo
28	Kyrgyzstan
29	Laos
30	Lebanon
31	Lesotho
32	Libya
33	Mali
34	Mauritania
35	Moldova
36	Mongolia
37	Montenegro
38	N. Cyprus
39	North Korea
40	North Macedonia
41	Palestine
42	Papua New Guinea
43	Puerto Rico
44	Qatar
45	S. Sudan
46	Solomon Is.
47	Somalia
48	Somaliland
49	Sudan
50	Suriname
51	Syria

5.3 Viz: Attempt 1

```
ne_countries(returnclass = "sf") |>                                    ①
  select(name, geometry) |>                                            ②
  left_join(fcc |> select(-co2_emmission),                             ③
            join_by(name == country)) |>                               ④
  ggplot() +
  geom_sf(aes(fill = consumption)) +
  facet_wrap(~food_category) +
  theme(legend.position = "bottom")
```

① ?
② ?
③ ?
④ ?



What does each annotated line in the above code chunk do? Use the numbers below the code chunk for your answers.

What problems does the above viz suffer from?

1. Not precisse
2. labs not labeled
3. too small
4.
5.

52	Taiwan
53	Tajikistan
54	Timor-Leste
55	Turkmenistan
56	United States of America
57	Uzbekistan
58	Vanuatu
59	W. Sahara
60	Yemen
61	eSwatini

Some of the countries do not have consumption data as shown in the above table and by the `NA` facet in the above choropleth map. What steps were used to find these countries in the above print out. **HINT**: To remove the `geometry` column, look at the documentation of the `st_drop_geometry` of the `sf` package.

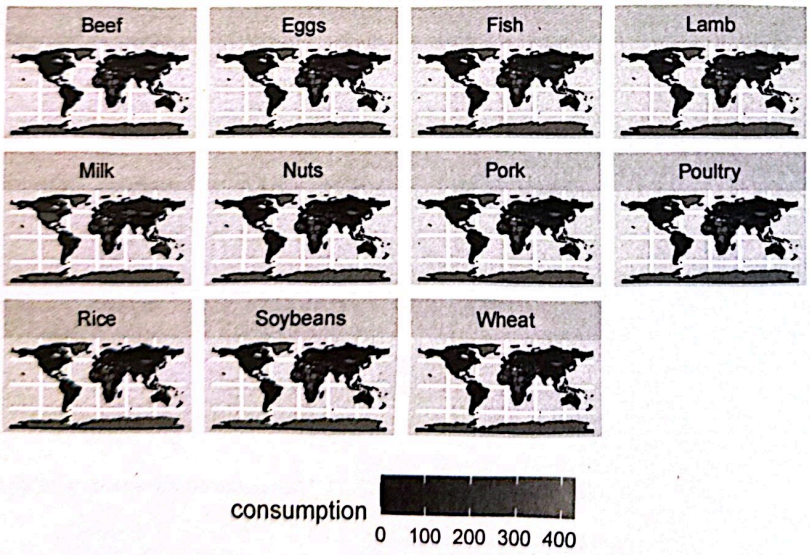
1. ~~Remov~~ Checking for NS
2. Removing NA data
3. Not including it as part of our points of interest
4.
5.

5.5 Viz: Attempt 2

We can fix the discrepancy in names either in the boundary data or the consumption data. The code chunk below fixes the discrepancy in the boundary data for some of the countries then plot the data again.

```
ne_countries(returnclass = "sf") |>
  select(name, geometry) |>
  mutate(name = ifelse(name == "United States of America", "USA", name)) |>                ①
  mutate(name = ifelse(name == "Bosnia and Herz.", "Bosnia and Herzegovina", name)) |>     ②
  mutate(name = ifelse(name == "Czechia", "Czech Republic", name)) |>                      ③
  mutate(name = ifelse(name == "Taiwan", "Taiwan. ROC", name)) |>                          ④
  left_join(fcc |> select(-co2_emmission),
            join_by(name == country)) |>
  pivot_wider(names_from = food_category,                                                  ⑤
              values_from = consumption) |>
  select(-"NA") |>                                                                         ⑥
  pivot_longer(cols = c(-name, -geometry),                                                 ⑦
               names_to = "food_category",
               values_to = "consumption") |>
  ggplot() +
  geom_sf(aes(fill = consumption)) +
  facet_wrap(~food_category) +
  theme(legend.position = "bottom")
```

① ?
② ?
③ ?
④ ?
⑤ ?
⑥ ?
⑦ ?



52	Taiwan
53	Tajikistan
54	Timor-Leste
55	Turkmenistan
56	United States of America
57	Uzbekistan
58	Vanuatu
59	W. Sahara
60	Yemen
61	eSwatini

Some of the countries do not have consumption data as shown in the above table and by the `NA` facet in the above choropleth map. What steps were used to find these countries in the above print out. **HINT:** To remove the `geometry` column, look at the documentation of the `st_drop_geometry` of the `sf` package.

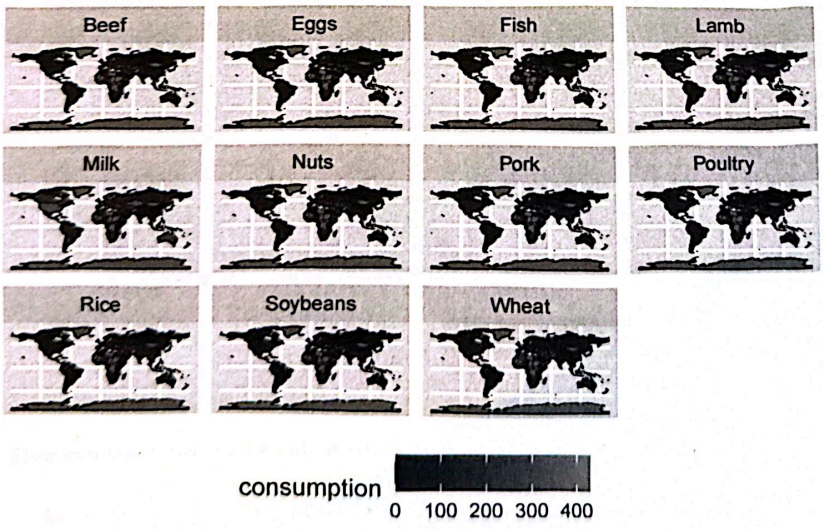
1. ~~Remov~~ Checking for NA
2. Removing NA data
3. Not including it as part of our points of interest
4.
5.

5.5 Viz: Attempt 2

We can fix the discrepancy in names either in the boundary data or the consumption data. The code chunk below fixes the discrepancy in the boundary data for some of the countries then plot the data again.

```
ne_countries(returnclass = "sf") |>
  select(name, geometry) |>
  mutate(name = ifelse(name == "United States of America", "USA", name)) |>                    ①
  mutate(name = ifelse(name == "Bosnia and Herz.", "Bosnia and Herzegovina", name)) |>         ②
  mutate(name = ifelse(name == "Czechia", "Czech Republic", name)) |>                          ③
  mutate(name = ifelse(name == "Taiwan", "Taiwan. ROC", name)) |>                              ④
  left_join(fcc |> select(-co2_emmission),
            join_by(name == country)) |>
  pivot_wider(names_from = food_category,                                                      ⑤
              values_from = consumption) |>
  select(-"NA") |>                                                                             ⑥
  pivot_longer(cols = c(-name, -geometry),                                                     ⑦
               names_to = "food_category",
               values_to = "consumption") |>
  ggplot() +
  geom_sf(aes(fill = consumption)) +
  facet_wrap(~food_category) +
  theme(legend.position = "bottom")
```

① ?
② ?
③ ? } Mutate the data so these countries are the main data included
④ ?
⑤ ? adding data while maintaining the former data
⑥ ? take out NA
⑦ ? Add values to food category & consumption



What does each annotated line in the above code chunk do? Use the numbers below the code chunk for your answer.

What problems does the above viz still suffer from?

1. It still has missing labels
2. Its data is still not recognizable
3.
4.
5.

5.6 Food Consumption Statistics

```
# A tibble: 11 x 4
   food_category   min   max range
   <fct>         <dbl> <dbl> <dbl>
 1 Milk           3.04 431.  428.
 2 Wheat          2.74 198.  195.
 3 Fish           0.24 180.  179.
 4 Rice           0.95 172.  171.
 5 Pork           0     67.1  67.1
 6 Poultry        0.47  62.5  62.0
 7 Beef           0.78  55.5  54.7
 8 Nuts           0.18  23.0  22.8
 9 Lamb           0     21.1  21.1
10 Eggs           0.16  19.2  19.0
11 Soybeans       0     17.0  17.0
```

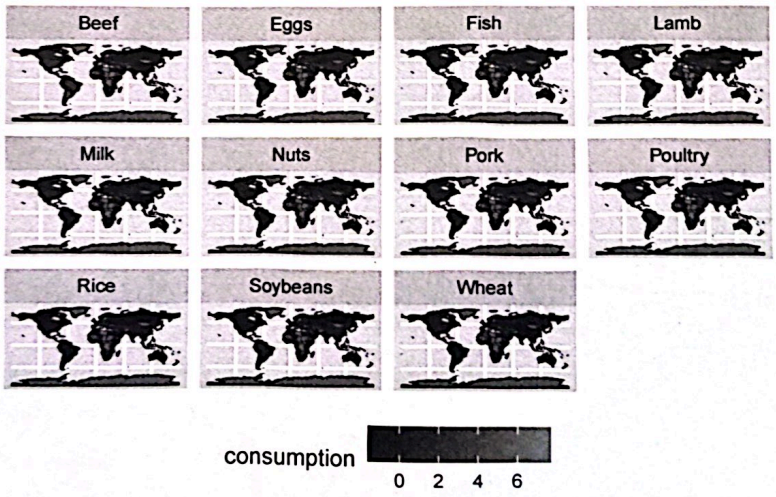
The above table shows the min, max, and range of consumption for each food. What steps were used to produce it?

1. Select
2. Summarize
3. arrange
4.
5.

5.7 Viz: Attempt 3

The food consumption ranges vary a lot making the choropleth maps less useful. The code chunk below fixes the problem by standardizing the consumption for foods.

```
ne_countries(returnclass = "sf") |>
  select(name, geometry) |>
  mutate(name = ifelse(name == "United States of America", "USA", name)) |>
  mutate(name = ifelse(name == "Bosnia and Herz.", "Bosnia and Herzegovina", name)) |>
  mutate(name = ifelse(name == "Czechia", "Czech Republic", name)) |>
  mutate(name = ifelse(name == "Taiwan", "Taiwan. ROC", name)) |>
  left_join(
    fcc |>
      select(-co2_emmission) |>
      group_by(food_category) |>
      mutate(consumption = (consumption - mean(consumption))/sd(consumption)),
    join_by(name == country)) |>
  pivot_wider(names_from = food_category, values_from = consumption) |>
  select(-"NA") |>
  pivot_longer(cols = c(-name, -geometry),
               names_to = "food_category",
               values_to = "consumption") |>
  ggplot() +
  geom_sf(aes(fill = consumption)) +
  facet_wrap(~food_category) +
  theme(legend.position = "bottom")
```



How can the above viz be enhanced?

1. Adding labels
2. Having clearer points of reference

6.1 Manage Plot Size

The default dimensions used by RStudio to produce large plots does not work most of the times. To control these dimensions, use the code chunk YAML options `fig-height` and `fig-width` at the top of the code chunk producing the plot as shown below:

```
#| fig-height: 22
#| fig-width: 11

ggplot(...) +
  geom_point(...)
```

6.2 Faceting

- To produce cleaner viz when using the `facet_wrap` function, look at the documentation of the `scale` argument.
- To control the number of facets per row, look at the documentation of the `ncol` argument of `facet_wrap` function.