Food shortage?

Tasks day 3 - Possible solutions

Solveig Bjørkholt

1. Make an R Markdown file. Title the document: "Food shortage?"

2. Make a chunk where you `library` in the *tidyverse* package.

```
library(tidyverse)
```

3. Download the dataset `global_food_prices.csv` from Canvas. Place it in a folder on your computer and find the path to the file. With this path, read the dataset into R using the function `read_csv`. Assign the dataset to an object called `foodprices`.

```
foodprices <- read.csv("../../../datafolder/global_food_prices.csv")
```

The data is gathered from this site, which again has gotten it from the World Food Programme and Humanitarian Data Exchange.

The dataset contains Global Food Prices data from the World Food Programme covering foods such as maize, rice, beans, fish, and sugar for 76 countries and some 1500 markets. The data goes back as far as 1992 for a few countries, although many countries started reporting from 2003 or thereafter. It includes these main variables: country, locality, market, goods purchased, price & currency used, quantity exchanged, and month/year of purchase.

All the names of the variables are given below:

- adm0_id: country id
- adm0_name: country name
- adm1_id: locality id
- adm1_name: locality name
- mkt_id: market id
- mkt_name: market name
- cm_id: commodity purchase id
- cm_name: commodity purchased
- cur_id: currency id
- cur_name: name of currency
- pt_id: market type id
- pt_name: market type (Retail/Wholesale/Producer/Farm Gate)
- um_id: measurement id
- um_name: unit of goods measurement
- mp_month: month recorded
- mpyear: year recorded
- mpprice: price paid

- mp_commoditysource: Source supplying price information

4. How many variables are there in this dataset? How many rows? Comment briefly on the size of the dataset in your R Markdown report. Why is it so big?

```
glimpse(foodprices)
```

```
## Rows: 2,050,638
## Columns: 19
## $ X                  <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ~
## $ adm0_id            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, NA, 1, 1, 1, 1, 1,~
## $ adm0_name          <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afgha~
## $ adm1_id            <int> 272, 272, 272, 272, 272, 272, 272, 272, 272, 272, 2~
## $ adm1_name          <chr> "Badakhshan", "Badakhshan", "Badakhshan", "Badakhsh~
## $ mkt_id             <int> 266, 266, 266, 266, 266, 266, 266, 266, NA, 266, 26~
## $ mkt_name           <chr> "Fayzabad", "Fayzabad", "Fayzabad", "Fayzabad", "Fa~
## $ cm_id              <int> 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, NA, 55,~
## $ cm_name            <chr> NA, "Bread - Retail", "Bread - Retail", "Bread - Re~
## $ cur_id             <int> 0, 0, 0, NA, 0, 0, 0, 0, NA, 0, NA, 0, 0, 0, 0, 0, ~
## $ cur_name           <chr> "AFN", "AFN", "AFN", NA, "AFN", "AFN", NA, "AFN", "~
## $ pt_id              <int> 15, NA, NA, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15,~
## $ pt_name            <chr> "Retail", "Retail", NA, "Retail", "Retail", "Retail~
## $ um_id              <int> 5, 5, 5, 5, 5, 5, 5, 5, NA, 5, NA, 5, 5, 5, 5, 5, 5~
## $ um_name            <chr> "KG", "KG", "KG", "KG", NA, "KG", "KG", "KG", "KG",~
## $ mp_month           <int> 1, 2, 3, 4, 5, 6, 7, NA, 9, 10, 11, 12, 1, 2, 3, 6,~
## $ mp_year            <int> 2014, 2014, 2014, NA, 2014, 2014, 2014, 2014, NA, 2~
## $ mp_price           <dbl> NA, 50.00, 50.00, 50.00, 50.00, 50.00, 50.00, 50.00~
## $ mp_commoditysource <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

The dataset has about 2 million observations and 18 variables. It's a very big dataset. The reason why this dataset is so big, is because it covers many countries that contain many markets, for many different commodities. It does this over many years. What happens then, is that we get many different units, for example "Bread-Retail in the market Fayzabad in Afghanistan in 2014". The combination of all these variables to make up so many units, results in a very big dataset.

5. To do some preliminary tests on the data, you decide to subset only a few observations and variables. First, use `filter` to get all observations where the year equals 2015 and 2020. Write it into a new object that you call `foodprices_subset`. How many observations and variables does this dataset have?

```
foodprices_subset <- foodprices %>%
    filter(mp_year %in% c("2015", "2020"))
```

The dataset has about 450000 observations and still 18 variables.

6. Second, use `select` to fetch the variables `adm0_name`, `mp_year`, `adm1_name`, `mkt_name`, `cm_name`, `cur_name`, and `mp_price`. Overwrite the old object by adding an arrow in and calling the object the same, `foodprices_subset`. How many observations and variables does the dataset have now?

```
foodprices_subset <- foodprices_subset %>%
    select(adm0_name, mp_year, adm1_name, mkt_name, cm_name, cur_name, mp_price)
```

Now the dataset has about 450000 observations and 7 variables.

7. Give the variables some names that work better for you to remember what they mean. You are free to choose which names you want, take a look over in the document to see what the different variables contain of information. Below is an example of some new names for the variables:

- adm0_name - country
- mp_year - year
- adm1_name - locality
- mkt_name - market
- cm_name - commodity
- cur_name - currency
- mp_price - price

Remember that when you use `rename`, the new name comes before the old name.

```
foodprices_subset <- foodprices_subset %>%
    rename(country = adm0_name, year = mp_year, locality = adm1_name, market = mkt_name,
        commodity = cm_name, currency = cur_name, price = mp_price)
```

8. What is the extent of missing values (`NA`) in our dataset? Use `is.na` and `table` to figure it out.

```
foodprices_subset %>%
    is.na() %>%
    table()
```

```
## .
##   FALSE    TRUE
## 2756535  407605
```

9. Use some maths in R to figure out what the percentage of missing values is. Recall that percentages are calculated by the number of observations that are missing, divided by all the observations, and multiplied by one hundred. Write the number in your report.

```
407605/(407605 + 2756535) * 100
```

```
## [1] 12.88202
```

About 13 percent of the observations in the dataset has missing values.

10. Use `group_by` and `summarise` to figure out what the sum of the prices for food was in each country for each year. Recall the the function used to find the sum is `sum`, and that to avoid trouble because of missing values, you have to add `na.rm = TRUE`.

```
foodprices_subset %>%
    group_by(country, year) %>%
    summarise(food = sum(price, na.rm = TRUE))
```

```
## ‘summarise()‘ has grouped output by ’country’. You can override using the
## ‘.groups‘ argument.
## # A tibble: 182 x 3
## # Groups:   country [95]
##    country      year    food
##    <chr>       <int>   <dbl>
##  1 Afghanistan  2015   67117.
##  2 Afghanistan  2020  303845.
##  3 Algeria      2015   75366.
##  4 Angola       2015   46395.
##  5 Angola       2020  138418.
##  6 Argentina    2015      30.5
##  7 Argentina    2020    5271.
##  8 Armenia      2015  580823.
##  9 Armenia      2020 1893159.
## 10 Bangladesh   2015  125160.
## # ... with 172 more rows
```

11. What was the total price for food for Kenya? Use the code you wrote above and add a row using `filter` to figure it out.

```
foodprices_subset %>%
    group_by(country, year) %>%
    summarise(food = sum(price, na.rm = TRUE)) %>%
    filter(country == "Kenya")
```

```
## 'summarise()' has grouped output by 'country'. You can override using the
## '.groups' argument.
```

```
## # A tibble: 2 x 3
## # Groups:   country [1]
##   country  year   food
##   <chr>   <int>  <dbl>
## 1 Kenya    2015 499154.
## 2 Kenya    2020 182921.
```

12. What's the average price for food for all countries in 2015 and 2020 respectively? Remember that the function `mean` gives you the average.

```
foodprices_subset %>%
    group_by(year) %>%
    summarise(foodprice = mean(price, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##    year foodprice
##   <int>     <dbl>
## 1  2015     5662.
## 2  2020     9330.
```

11. What does the code below do? Comment each line to explain what the different lines do. Remember that comments are made by setting a hashtag in the code chunk and writing your comment after that; # comment

Which country traded most spices in our dataset?

```
foodprices_subset %>%
  group_by(country, year) %>% # grouping by country and year to get statistics by these variables
  count(commodity, name = "commodity_number") %>% # counting the number of commodities per country and
  na.omit() %>% # removing rows with missing values
  mutate(spices_commodity = ifelse(commodity %in% c("Salt - Retail", "Sugar - Retail"), # making a new
                                   "spice", # ... the new variable "spices_commodity" should have the v
                                   "other")) %>% # ... otherwise it should have the name "other"
  filter(spices_commodity == "spice") %>% # filter out only the rows that have the value "spice" on our
  ungroup() %>% # ungrouping the dataset
  arrange(desc(commodity_number)) # arranging the dataset so that the values with the highest number on
```

```
## # A tibble: 104 x 5
##    country                        year commodity commodity_number spices_commodity
##    <chr>                         <int> <chr>                <int> <chr>
##  1 Indonesia                      2020 Sugar - ~             1244 spice
##  2 Bassas da India                2020 Sugar - ~              508 spice
##  3 Myanmar                        2020 Salt - R~              466 spice
##  4 Bassas da India                2015 Sugar - ~              421 spice
##  5 Syrian Arab Republic           2020 Sugar - ~              413 spice
##  6 Burundi                        2020 Salt - R~              350 spice
##  7 Libya                          2020 Salt - R~              275 spice
##  8 Libya                          2020 Sugar - ~              256 spice
##  9 Democratic Republic of the~    2020 Sugar - ~              254 spice
## 10 Democratic Republic of the~    2020 Salt - R~              251 spice
## # ... with 94 more rows
```

Indonesia traded most spices.