Practice for the practice Quiz

Using Problem 12.2.1 Exercise 2 as a guide, use the ideas from Chapter 13 to answer the questions for `table2`.

1. Compute the rate and include it in a final dataframe with the years as columns.

Answer:
The first answer approaches the problem by splitting the dataset into two and then joining the two dataset.

```r
library(tidyverse)

table2

# A tibble: 12 x 4
#  country year type count
#  <chr> <int> <chr> <int>
# 1 Afghanistan 1999 cases 745
# 2 Afghanistan 1999 population 19987071
# 3 Afghanistan 2000 cases 2666
# 4 Afghanistan 2000 population 20595360
# 5 Brazil 1999 cases 37737
# 6 Brazil 1999 population 172006362
# 7 Brazil 2000 cases 80488
# 8 Brazil 2000 population 174504898
# 9 China 1999 cases 212258
# 10 China 1999 population 1272915272
# 11 China 2000 cases 213766
# 12 China 2000 population 1280428583

table2 %>% arrange(type)

# A tibble: 12 x 4
#  country year type count
#  <chr> <int> <chr> <int>
# 1 Afghanistan 1999 cases 745
# 2 Afghanistan 2000 cases 2666
# 3 Brazil 1999 cases 37737
# 4 Brazil 2000 cases 80488
# 5 China 1999 cases 212258
# 6 China 2000 cases 213766
# 7 Afghanistan 1999 population 19987071
# 8 Afghanistan 2000 population 20595360
# 9 Brazil 1999 population 172006362
# 10 Brazil 2000 population 174504898
# 11 China 1999 population 1272915272
# 12 China 2000 population 1280428583

# A tibble: 6 x 3
#  country year cases
#  <chr> <int> <int>
```

```r
table2_cases <- table2 %>% filter(type == "cases") %>%
  select(country, year, count) %>%
  rename(cases = count)
table2_cases

# A tibble: 6 x 3
#  country year cases
#  <chr> <int> <int>
```
```r
library(stringr)

table2_pop <- table2 %>% filter(type == "population") %>%
  select(country, year, count) %>%
  rename(population = count)

table2_pop

# A tibble: 6 x 3
#  country year population
#  <chr> <int>     <int>
# 1 Afghanistan 1999    19987071
# 2 Afghanistan 2000    20595360
# 3 Brazil      1999    172006362
# 4 Brazil      2000    174504898
# 5 China       1999   1272915272
# 6 China       2000   1280428583

Now join the two datasets using two variables as the unique key.

table2_join <- table2_cases %>% inner_join(table2_pop, by=c("country", "year"))

table2_join

# A tibble: 6 x 4
#  country year cases population
#  <chr> <int> <int>     <int>
# 1 Afghanistan 1999   745    19987071
# 2 Afghanistan 2000  2666    20595360
# 3 Brazil      1999  37737    172006362
# 4 Brazil      2000  80488    174504898
# 5 China       1999 212258   1272915272
# 6 China       2000 213766   1280428583

Create the new column.

table2_new <- table2_join %>% mutate(rate = cases / population * 10000)

table2_new

# A tibble: 6 x 5
#  country year cases population rate
#  <chr> <int> <int>     <int>    <dbl>
# 1 Afghanistan 1999   745    19987071   0.373
# 2 Afghanistan 2000  2666    20595360   1.29
# 3 Brazil      1999  37737    172006362  2.19
# 4 Brazil      2000  80488    174504898  4.61
# 5 China       1999 212258   1272915272 1.67
# 6 China       2000 213766   1280428583 1.67

Now spread the data out into two columns.
```
table2_new_spread <- table2_new %>% select(country, year, rate) %>%
  spread(year, rate)

## A tibble: 3 x 3
## country   year  rate
##          <chr> <dbl> <dbl>
## 1 Afghanistan 1999  0.373
## 2 Brazil   2000  4.61
## 3 China      1.67  1.67

Now try the new function `pivot_wider()`. Note new this function is from the `tidyr` 1.0 package.

table2_new_spread2 <- table2_new %>% select(country, year, rate) %>%
  pivot_wider(country, names_from = year, values_from = rate)

## A tibble: 3 x 3
## country   1999  2000
##          <chr> <dbl> <dbl>
## 1 Afghanistan 0.373 1.29
## 2 Brazil 2.19 4.61
## 3 China 1.67 1.67

Are the two files the same. Let’s give the `comparedf()` function a try. It is from the `arsenal` R package.

```r
library(arsenal)
comparedf(table2_new_spread, table2_new_spread2)
```

## Compare Object
##
## Function Call:
## comparedf(x = table2_new_spread, y = table2_new_spread2)
##
## Shared: 3 non-by variables and 3 observations.
## Not shared: 0 variables and 0 observations.
##
## Differences found in 0/3 variables compared.
## 0 variables compared have non-identical attributes.

**Alternative Solution:**

Can we use `spread` from the beginning? Yes.

```r
table2 %>%
  spread(key = type, value = count) %>%
  mutate(rate = cases/population) %>%
  select(-cases, -population) %>%
  spread(key = year, value = rate)
```

## A tibble: 3 x 3
## country   `1999` `2000`
##          <chr> <dbl> <dbl>
## 1 Afghanistan 0.0000373 0.000129
## 2 Brazil 0.000219 0.000461
## 3 China 0.000167 0.000167
Or

```r
table2 %>% pivot_wider(names_from = type, values_from = count) %>%
  mutate(rate = cases/population) %>%
  select(-cases, -population) %>%
  pivot_wider(names_from = year, values_from = rate)
```

```r
## # A tibble: 3 x 3
## country     1999   2000
## <chr>       <dbl>  <dbl>
## 1 Afghanistan 0.0000373 0.000129
## 2 Brazil 0.000219 0.000461
## 3 China 0.000167 0.000167
```

2. Now make a clustered bar graph. Question, which table is the one to use, `table2_new` or `table2_new_spread`?

**Answer:** The one to use is in tidy format. So `table2_new`. Note the use of `as.factor()` function. This is our next topic of discussion.

```r
table2_new %>% ggplot(aes(x = country, y = rate, fill = as.factor(year))) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_light()
```

Or you can make the plot using year to group the bars.

```r
table2_new %>% ggplot(aes(x = as.factor(year), y = rate, fill = country)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_light()
```