Assignment 2 Question 2

Morgan Baccus

```
#Read in and tidy the data set

library(tidyr)

tidyr::who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  ) %>%
  mutate(
    key = stringr::str_replace(key, "newrel", "new_rel")
  ) %>%
  separate(key, c("new", "var", "sexage")) %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1)
```

```
## # A tibble: 76,046 x 6
##    country      year var   sex   age   cases
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1997 sp    m     1524     10
##  3 Afghanistan  1997 sp    m     2534      6
##  4 Afghanistan  1997 sp    m     3544      3
##  5 Afghanistan  1997 sp    m     4554      5
##  6 Afghanistan  1997 sp    m     5564      2
##  7 Afghanistan  1997 sp    m     65        0
##  8 Afghanistan  1997 sp    f     014       5
##  9 Afghanistan  1997 sp    f     1524     38
## 10 Afghanistan  1997 sp    f     2534     36
## # ... with 76,036 more rows
```

Question 2.a the line “mutate(key = stringr::str_replace(key,”newrel“,”new_rel“))” is necessary to properly tidy the data so that the names are consistent. If you skip this line then later when you need to separate the variable names at each underscore, it won’t work for this variable.

```
#Question 2.b
#How many entries are gone after setting values_drop_na to true

who1 <- tidyr::who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  )
who1
```

```
## # A tibble: 76,046 x 6
##    country     iso2  iso3   year key          cases
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1997 new_sp_m014      0
##  2 Afghanistan AF    AFG    1997 new_sp_m1524    10
##  3 Afghanistan AF    AFG    1997 new_sp_m2534     6
##  4 Afghanistan AF    AFG    1997 new_sp_m3544     3
##  5 Afghanistan AF    AFG    1997 new_sp_m4554     5
##  6 Afghanistan AF    AFG    1997 new_sp_m5564     2
##  7 Afghanistan AF    AFG    1997 new_sp_m65       0
##  8 Afghanistan AF    AFG    1997 new_sp_f014      5
##  9 Afghanistan AF    AFG    1997 new_sp_f1524    38
## 10 Afghanistan AF    AFG    1997 new_sp_f2534    36
## # ... with 76,036 more rows
```

```
who2 <-tidyr::who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = FALSE
  )
who2
```

```
## # A tibble: 405,440 x 6
##    country     iso2  iso3   year key          cases
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1980 new_sp_m014     NA
##  2 Afghanistan AF    AFG    1980 new_sp_m1524    NA
##  3 Afghanistan AF    AFG    1980 new_sp_m2534    NA
##  4 Afghanistan AF    AFG    1980 new_sp_m3544    NA
##  5 Afghanistan AF    AFG    1980 new_sp_m4554    NA
##  6 Afghanistan AF    AFG    1980 new_sp_m5564    NA
##  7 Afghanistan AF    AFG    1980 new_sp_m65      NA
##  8 Afghanistan AF    AFG    1980 new_sp_f014     NA
##  9 Afghanistan AF    AFG    1980 new_sp_f1524    NA
## 10 Afghanistan AF    AFG    1980 new_sp_f2534    NA
## # ... with 405,430 more rows
```

When values_drop_na = FALSE, there are 404,440 observations. When values_drop_na = TRUE, there are only 76,046 observations. This means that 329,394 entries were removed.

Question 2.c Explicit missing values: a value that is marked as na Implicit missing values: a value that is simply not present in the data

There are implicit missing values for the variable cases and are shown as zeros in the dataset.

Question 2.d I believe that country, year, var, sex, and cases are all typed appropriately.It seems as if age could be reworked so that each age range corresponded with a letter and the letter is what appeared in the data. That would make a chr the best type. Reading the lower age and the upper range as one number is confusing and unnecessary as they are stored as chrs and can't even be used in calculations.

```
#Question 2.e
#Generate an informative data visualization

who1 %>%
  group_by(country) %>%
  summarise(average_cases = mean(cases, na.rm=TRUE)) %>%
  top_n(10) %>%
  arrange(desc(average_cases))
```

```
## Selecting by average_cases
```

```
## # A tibble: 10 x 2
##    country                          average_cases
##    <chr>                                    <dbl>
##  1 India                                   27729.
##  2 China                                   23049.
##  3 South Africa                             7414.
##  4 Indonesia                                6928.
##  5 Philippines                              4537.
##  6 Bangladesh                               4011.
##  7 Viet Nam                                 3832.
##  8 Democratic Republic of the Congo         3612.
##  9 Pakistan                                 3457.
## 10 Nigeria                                  2471.
```

```
who1 <- who1 %>% mutate(average_cases = mean(cases, na.rm=TRUE))
```

This chart shows the top ten countries with the highest average cases in descending order. This is interesting to look at because it shows that the top countries are all countries that do not have well established health care systems. If a country that does have an established health care system was in the top ten countries with the highest average number of cases, that would suggest other issues going on. It is also interesting to see the large difference between the number of average cases for the first two countries and then the next eight. The number of cases almost triples from the third country, South Africa, to the second country, China.

```
#Question 2.f
#Create a table and use pivot_longer()/gather() and separate()/pivot_wider() to alter it

qtrRev <- data.frame(Group=rep(c('1', '2', '3'), each=4),
                     Year=rep(c('2006', '2007', '2008', '2009'), times=3),
                     Qtr.1=rep(c(15, 12, 22, 10, 12, 16, 13, 23, 11, 13, 17, 14)),
                     Qtr.2=rep(c(16, 13, 22, 14, 13, 14, 11, 20, 12, 11, 12, 9)),
                     Qtr.3=rep(c(19, 27, 24, 20, 25, 21, 29, 26, 22, 27, 23, 31)),
                     Qtr.4=rep(c(17, 23, 20, 16, 18, 19, 15, 20, 16, 21, 19, 24)))

qtrRev %>%
  gather(Quarter, Revenue, Qtr.1:Qtr.4) %>%
  separate(Quarter, c("Time_Interval", "Interval_ID"))
```

```
##    Group Year Time_Interval Interval_ID Revenue
## 1      1 2006           Qtr           1      15
## 2      1 2007           Qtr           1      12
## 3      1 2008           Qtr           1      22
## 4      1 2009           Qtr           1      10
## 5      2 2006           Qtr           1      12
## 6      2 2007           Qtr           1      16
## 7      2 2008           Qtr           1      13
## 8      2 2009           Qtr           1      23
## 9      3 2006           Qtr           1      11
## 10     3 2007           Qtr           1      13
## 11     3 2008           Qtr           1      17
## 12     3 2009           Qtr           1      14
## 13     1 2006           Qtr           2      16
## 14     1 2007           Qtr           2      13
## 15     1 2008           Qtr           2      22
## 16     1 2009           Qtr           2      14
## 17     2 2006           Qtr           2      13
## 18     2 2007           Qtr           2      14
## 19     2 2008           Qtr           2      11
## 20     2 2009           Qtr           2      20
## 21     3 2006           Qtr           2      12
## 22     3 2007           Qtr           2      11
## 23     3 2008           Qtr           2      12
## 24     3 2009           Qtr           2       9
## 25     1 2006           Qtr           3      19
## 26     1 2007           Qtr           3      27
## 27     1 2008           Qtr           3      24
## 28     1 2009           Qtr           3      20
## 29     2 2006           Qtr           3      25
## 30     2 2007           Qtr           3      21
## 31     2 2008           Qtr           3      29
## 32     2 2009           Qtr           3      26
## 33     3 2006           Qtr           3      22
## 34     3 2007           Qtr           3      27
## 35     3 2008           Qtr           3      23
## 36     3 2009           Qtr           3      31
## 37     1 2006           Qtr           4      17
## 38     1 2007           Qtr           4      23
## 39     1 2008           Qtr           4      20
## 40     1 2009           Qtr           4      16
## 41     2 2006           Qtr           4      18
## 42     2 2007           Qtr           4      19
## 43     2 2008           Qtr           4      15
## 44     2 2009           Qtr           4      20
## 45     3 2006           Qtr           4      16
## 46     3 2007           Qtr           4      21
## 47     3 2008           Qtr           4      19
## 48     3 2009           Qtr           4      24
```