CptS 575 Data Science: Assignment 3

Md Muhtasim Billah

9/21/2020

Question 1

Reading in the data in R as a data frame and printing the first few values of the columns with a header including "sleep".

```
library(dplyr)
msleep = read.csv(url("https://scads.eecs.wsu.edu/wp-content/uploads/2017/10/msleep_ggplot2.csv"),
                  header = TRUE)
sleep_columns = select(msleep,contains("sleep"))
head(sleep_columns)
```

```
##   sleep_total sleep_rem sleep_cycle
## 1        12.1        NA          NA
## 2        17.0       1.8          NA
## 3        14.4       2.4          NA
## 4        14.9       2.3   0.1333333
## 5         4.0       0.7   0.6666667
## 6        14.4       2.2   0.7666667
```

1 (a)

The number of animals which weigh under 1 kilogram and sleep more than 14 hours a day.

```
library(dplyr)
anim_1kg_14h = filter(msleep, bodywt<1 & sleep_total>14)
anim_1kg_14h
```

```
##                             name        genus    vore           order
## 1                    Owl monkey        Aotus    omni        Primates
## 2     Greater short-tailed shrew      Blarina    omni    Soricomorpha
## 3                  Big brown bat    Eptesicus insecti      Chiroptera
## 4      Western american chipmunk     Eutamias   herbi        Rodentia
## 5           Thick-tailed opposum   Lutreolina   carni Didelphimorphia
## 6               Mongolian gerbil     Meriones   herbi        Rodentia
## 7                 Golden hamster Mesocricetus   herbi        Rodentia
## 8               Little brown bat       Myotis insecti      Chiroptera
## 9           Round-tailed muskrat     Neofiber   herbi        Rodentia
## 10    Northern grasshopper mouse    Onychomys   carni        Rodentia
## 11         Arctic ground squirrel Spermophilus   herbi        Rodentia
## 12 Golden-mantled ground squirrel Spermophilus   herbi        Rodentia
## 13     Eastern american chipmunk       Tamias   herbi        Rodentia
## 14                         Tenrec       Tenrec    omni    Afrosoricida
##    conservation sleep_total sleep_rem sleep_cycle awake brainwt bodywt
## 1          <NA>        17.0       1.8          NA   7.0 0.01550  0.480
```

```
## 2            lc        14.9       2.3   0.1333333   9.1 0.00029  0.019
## 3            lc        19.7       3.9   0.1166667   4.3 0.00030  0.023
## 4          <NA>        14.9        NA          NA   9.1      NA  0.071
## 5            lc        19.4       6.6          NA   4.6      NA  0.370
## 6            lc        14.2       1.9          NA   9.8      NA  0.053
## 7            en        14.3       3.1   0.2000000   9.7 0.00100  0.120
## 8          <NA>        19.9       2.0   0.2000000   4.1 0.00025  0.010
## 9            nt        14.6        NA          NA   9.4      NA  0.266
## 10           lc        14.5        NA          NA   9.5      NA  0.028
## 11           lc        16.6        NA          NA   7.4 0.00570  0.920
## 12           lc        15.9       3.0          NA   8.1      NA  0.205
## 13         <NA>        15.8        NA          NA   8.2      NA  0.112
## 14         <NA>        15.6       2.3          NA   8.4 0.00260  0.900
```

```
count(anim_1kg_14h)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    14
```

1 (b)

The name, order, sleep time and bodyweight of the animals with the 6 longest sleep times, in order of sleep time.

```
head(arrange(select(msleep, name, order, sleep_total, bodywt),desc(sleep_total)),n=6)
```

```
##                     name           order sleep_total bodywt
## 1       Little brown bat      Chiroptera        19.9  0.010
## 2          Big brown bat      Chiroptera        19.7  0.023
## 3   Thick-tailed opposum Didelphimorphia        19.4  0.370
## 4        Giant armadillo       Cingulata        18.1 60.000
## 5 North American Opossum Didelphimorphia        18.0  1.700
## 6   Long-nosed armadillo       Cingulata        17.4  3.500
```

1 (c)

Adding two new columns to the dataframe; "wt_ratio" with the ratio of brain size to body weight, "rem_ratio" with the ratio of rem sleep to sleep time.

```
msleep2 = mutate(msleep, wt_ratio = brainwt/bodywt, rem_ratio = sleep_rem/sleep_total)
head(msleep2)
```

```
##                        name      genus  vore        order conservation
## 1                   Cheetah   Acinonyx carni    Carnivora           lc
## 2                 Owl monkey      Aotus  omni     Primates         <NA>
## 3            Mountain beaver Aplodontia herbi     Rodentia           nt
## 4 Greater short-tailed shrew    Blarina  omni Soricomorpha           lc
## 5                       Cow        Bos herbi Artiodactyla domesticated
```

```
## 6           Three-toed sloth   Bradypus herbi       Pilosa         <NA>
##   sleep_total sleep_rem sleep_cycle awake brainwt  bodywt   wt_ratio rem_ratio
## 1        12.1        NA          NA  11.9      NA  50.000         NA        NA
## 2        17.0       1.8          NA   7.0 0.01550   0.480 0.03229167 0.1058824
## 3        14.4       2.4          NA   9.6      NA   1.350         NA 0.1666667
## 4        14.9       2.3   0.1333333   9.1 0.00029   0.019 0.01526316 0.1543624
## 5         4.0       0.7   0.6666667  20.0 0.42300 600.000 0.00070500 0.1750000
## 6        14.4       2.2   0.7666667   9.6      NA   3.850         NA 0.1527778
```

1 (d)

Displaying the average, min and max sleep times for each order.

```
msleep %>% group_by(order) %>% summarise(sleep_avg = mean(sleep_total), sleep_min = min(sleep_total),
                                         sleep_max = max(sleep_total))
```

```
## # A tibble: 19 x 4
##    order           sleep_avg sleep_min sleep_max
##    <fct>               <dbl>     <dbl>     <dbl>
##  1 Afrosoricida        15.6       15.6      15.6
##  2 Artiodactyla         4.52       1.9       9.1
##  3 Carnivora           10.1        3.5      15.8
##  4 Cetacea              4.5        2.7       5.6
##  5 Chiroptera          19.8       19.7      19.9
##  6 Cingulata           17.8       17.4      18.1
##  7 Didelphimorphia     18.7       18        19.4
##  8 Diprotodontia       12.4       11.1      13.7
##  9 Erinaceomorpha      10.2       10.1      10.3
## 10 Hyracoidea           5.67       5.3       6.3
## 11 Lagomorpha           8.4        8.4       8.4
## 12 Monotremata          8.6        8.6       8.6
## 13 Perissodactyla       3.47       2.9       4.4
## 14 Pilosa              14.4       14.4      14.4
## 15 Primates            10.5        8        17
## 16 Proboscidea          3.6        3.3       3.9
## 17 Rodentia            12.5        7        16.6
## 18 Scandentia           8.9        8.9       8.9
## 19 Soricomorpha        11.1        8.4      14.9
```

1 (e)

Imputing the missing brain weights as the average wt_ratio for that animal's order times the animal's weight.

```
missingBrainWtRatio = msleep %>%
  group_by(order) %>%
  mutate(
    brainwt=ifelse(
      is.na(brainwt),
      ifelse(is.nan(mean(brainwt,na.rm=TRUE)),0,
             mean(brainwt,na.rm=TRUE))/mean(bodywt,na.rm=TRUE),brainwt)) %>%
  ungroup(order)
head(missingBrainWtRatio)
```

```
## # A tibble: 6 x 11
##   name  genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <fct> <fct> <fct> <fct> <fct>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Chee... Acin... carni Carn... lc                  12.1      NA        NA      11.9
## 2 Owl ... Aotus omni  Prim... <NA>               17         1.8      NA       7
## 3 Moun... Aplo... herbi Rode... nt                  14.4       2.4      NA       9.6
## 4 Grea... Blar... omni  Sori... lc                  14.9       2.3       0.133   9.1
## 5 Cow   Bos   herbi Arti... domesticated         4         0.7       0.667  20
## 6 Thre... Brad... herbi Pilo... <NA>                14.4       2.2       0.767   9.6
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

Making a second copy of the dataframe, but this time imputing missing brain weights with the average brain weight for that animal's order.

```
missingBrainWtRatio2 = msleep %>%
  group_by(order) %>%
  mutate(
    brainwt = ifelse(
      is.na(brainwt),
      ifelse(is.nan(mean(brainwt,na.rm = TRUE)),0,
             mean(brainwt,na.rm = TRUE)), brainwt)) %>%
  ungroup(order)
head(missingBrainWtRatio2)
```

```
## # A tibble: 6 x 11
##   name  genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <fct> <fct> <fct> <fct> <fct>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Chee... Acin... carni Carn... lc                  12.1      NA        NA      11.9
## 2 Owl ... Aotus omni  Prim... <NA>               17         1.8      NA       7
## 3 Moun... Aplo... herbi Rode... nt                  14.4       2.4      NA       9.6
## 4 Grea... Blar... omni  Sori... lc                  14.9       2.3       0.133   9.1
## 5 Cow   Bos   herbi Arti... domesticated         4         0.7       0.667  20
## 6 Thre... Brad... herbi Pilo... <NA>                14.4       2.2       0.767   9.6
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

The best way to impute the data is by replacing the null values with the mean. Thus, even if the observations are removed from the dataset, the mean value will remain the same which will provide some statistical advantages.

The above procedure can be applied for filling the missing values of other columns such as "sleep_rem" and "sleep_cycle" which are shown below (one at a time).

```
missingSleepRem = msleep %>%
  group_by(order) %>%
  mutate(
    sleep_rem = ifelse(
      is.na(sleep_rem),
      ifelse(is.nan(mean(sleep_rem,na.rm = TRUE)),0,
             mean(sleep_rem,na.rm = TRUE)), sleep_rem)) %>%
  ungroup(order)
head(missingSleepRem)
```

```
## # A tibble: 6 x 11
```

```
##   name  genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <fct> <fct> <fct> <fct> <fct>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Chee... Acin... carni Carn... lc                 12.1      1.87      NA      11.9
## 2 Owl ... Aotus omni  Prim... <NA>               17        1.8       NA       7
## 3 Moun... Aplo... herbi Rode... nt                 14.4      2.4       NA       9.6
## 4 Grea... Blar... omni  Sori... lc                 14.9      2.3        0.133   9.1
## 5 Cow   Bos   herbi Arti... domesticated       4        0.7        0.667  20
## 6 Thre... Brad... herbi Pilo... <NA>               14.4      2.2        0.767   9.6
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```
missingSleepCycle = msleep %>%
  group_by(order) %>%
  mutate(
    sleep_cycle = ifelse(
      is.na(sleep_cycle),
      ifelse(is.nan(mean(sleep_cycle,na.rm = TRUE)),0,
             mean(sleep_cycle,na.rm = TRUE)), sleep_cycle)) %>%
  ungroup(order)
head(missingSleepCycle)
```

```
## # A tibble: 6 x 11
##   name  genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <fct> <fct> <fct> <fct> <fct>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Chee... Acin... carni Carn... lc                 12.1      NA         0.371  11.9
## 2 Owl ... Aotus omni  Prim... <NA>               17         1.8       0.977   7
## 3 Moun... Aplo... herbi Rode... nt                 14.4       2.4       0.181   9.6
## 4 Grea... Blar... omni  Sori... lc                 14.9       2.3       0.133   9.1
## 5 Cow   Bos   herbi Arti... domesticated       4         0.7       0.667  20
## 6 Thre... Brad... herbi Pilo... <NA>               14.4       2.2       0.767   9.6
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

Question 2.

Loading the who dataset from the "tidyr" package.

```
library(tidyr)
head(who)
```

```
## # A tibble: 6 x 60
##   country iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
##   <chr>   <chr> <chr> <int>       <int>        <int>        <int>        <int>
## 1 Afghan... AF    AFG    1980          NA           NA           NA           NA
## 2 Afghan... AF    AFG    1981          NA           NA           NA           NA
## 3 Afghan... AF    AFG    1982          NA           NA           NA           NA
## 4 Afghan... AF    AFG    1983          NA           NA           NA           NA
## 5 Afghan... AF    AFG    1984          NA           NA           NA           NA
## 6 Afghan... AF    AFG    1985          NA           NA           NA           NA
## # ... with 52 more variables: new_sp_m4554 <int>, new_sp_m5564 <int>,
## #   new_sp_m65 <int>, new_sp_f014 <int>, new_sp_f1524 <int>,
## #   new_sp_f2534 <int>, new_sp_f3544 <int>, new_sp_f4554 <int>,
## #   new_sp_f5564 <int>, new_sp_f65 <int>, new_sn_m014 <int>,
## #   new_sn_m1524 <int>, new_sn_m2534 <int>, new_sn_m3544 <int>,
## #   new_sn_m4554 <int>, new_sn_m5564 <int>, new_sn_m65 <int>,
## #   new_sn_f014 <int>, new_sn_f1524 <int>, new_sn_f2534 <int>,
## #   new_sn_f3544 <int>, new_sn_f4554 <int>, new_sn_f5564 <int>,
## #   new_sn_f65 <int>, new_ep_m014 <int>, new_ep_m1524 <int>,
## #   new_ep_m2534 <int>, new_ep_m3544 <int>, new_ep_m4554 <int>,
## #   new_ep_m5564 <int>, new_ep_m65 <int>, new_ep_f014 <int>,
## #   new_ep_f1524 <int>, new_ep_f2534 <int>, new_ep_f3544 <int>,
## #   new_ep_f4554 <int>, new_ep_f5564 <int>, new_ep_f65 <int>,
## #   newrel_m014 <int>, newrel_m1524 <int>, newrel_m2534 <int>,
## #   newrel_m3544 <int>, newrel_m4554 <int>, newrel_m5564 <int>,
## #   newrel_m65 <int>, newrel_f014 <int>, newrel_f1524 <int>,
## #   newrel_f2534 <int>, newrel_f3544 <int>, newrel_f4554 <int>,
## #   newrel_f5564 <int>, newrel_f65 <int>
```

Tidying the dataset according to the case study provided here: http://r4ds.had.co.nz/tidy-data.html#case-study

```
#considering the columns "new_sp_m014" to "newrel_f65" as values, they are put under the column name "k
#and their number of appearences are put under "cases"

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

#counting the values in the "key" column
who1 %>% count(key)
```

```
## # A tibble: 56 x 2
```

```
##    key              n
##    <chr>        <int>
##  1 new_ep_f014   1032
##  2 new_ep_f1524  1021
##  3 new_ep_f2534  1021
##  4 new_ep_f3544  1021
##  5 new_ep_f4554  1017
##  6 new_ep_f5564  1017
##  7 new_ep_f65    1014
##  8 new_ep_m014   1038
##  9 new_ep_m1524  1026
## 10 new_ep_m2534  1020
## # ... with 46 more rows
```

```
#replacing "newrel" with "new_rel"
who2 = who1 %>%
  mutate(names_from = stringr::str_replace(key, "newrel", "new_rel"))
#separating the "key" column into 3 columns ("new", "type", "sexage")
who3 = who2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
#dropping "new column since it is constant and
#dropping "iso2" and "iso3" since they are redundant
who4 = who3 %>%
  select(-new, -iso2, -iso3)
#separating sex and age, splitting after the first character
who5 <- who4 %>%
  separate(sexage, c("sex", "age"), sep = 1)
```

2(a)

The line <mutate(key = stringr::str_replace(key, “newrel”, “new_rel”))> is necessary to maintain the consistency of the values of the column “key” in the dataset. Specially, when we apply the “separate()” method to create three new columns (“new”, “type”, “sexage”), if the previous line is not executed, “newrel” will remain under the “new column rather than being splitted into two different column (since”_" is used as the separator). Thus, it is required to properly tidy the data.

2(b)

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

WithoutNA = who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
```

```
    values_drop_na = TRUE
  )
count(WithNA)
```

```
## # A tibble: 1 x 1
##         n
##     <int>
## 1 405440
```

```
count(WithoutNA)
```

```
## # A tibble: 1 x 1
##        n
##    <int>
## 1 76046
```

```
EntriesRemoved = count(WithNA) - count(WithoutNA)
EntriesRemoved
```

```
##         n
## 1 329394
```

2(c)

Explicit missing value is recognized by a specific representation of the value such as row=NA which indicates that a certain row has a missing value. On the other hand, implicit missing value is represented differently such as row="" or row=0.

```
implicitMissingVals = who5 %>%
  filter(cases==0) %>%
  nrow()
implicitMissingVals
```

```
## [1] 11080
```

2(d)

Looking at the features (country, year, var, sex, age, cases) in the tidied data, it seems that the feature "age" is typed as a character variable. It would be more reasonable to treat this as an integer.

```
head(who5)
```

```
## # A tibble: 6 x 7
##   country      year type  sex   age   cases names_from
##   <chr>       <int> <chr> <chr> <chr> <int> <chr>
## 1 Afghanistan  1997 sp    m     014       0 new_sp_m014
## 2 Afghanistan  1997 sp    m     1524     10 new_sp_m1524
## 3 Afghanistan  1997 sp    m     2534      6 new_sp_m2534
## 4 Afghanistan  1997 sp    m     3544      3 new_sp_m3544
## 5 Afghanistan  1997 sp    m     4554      5 new_sp_m4554
## 6 Afghanistan  1997 sp    m     5564      2 new_sp_m5564
```

2(e)

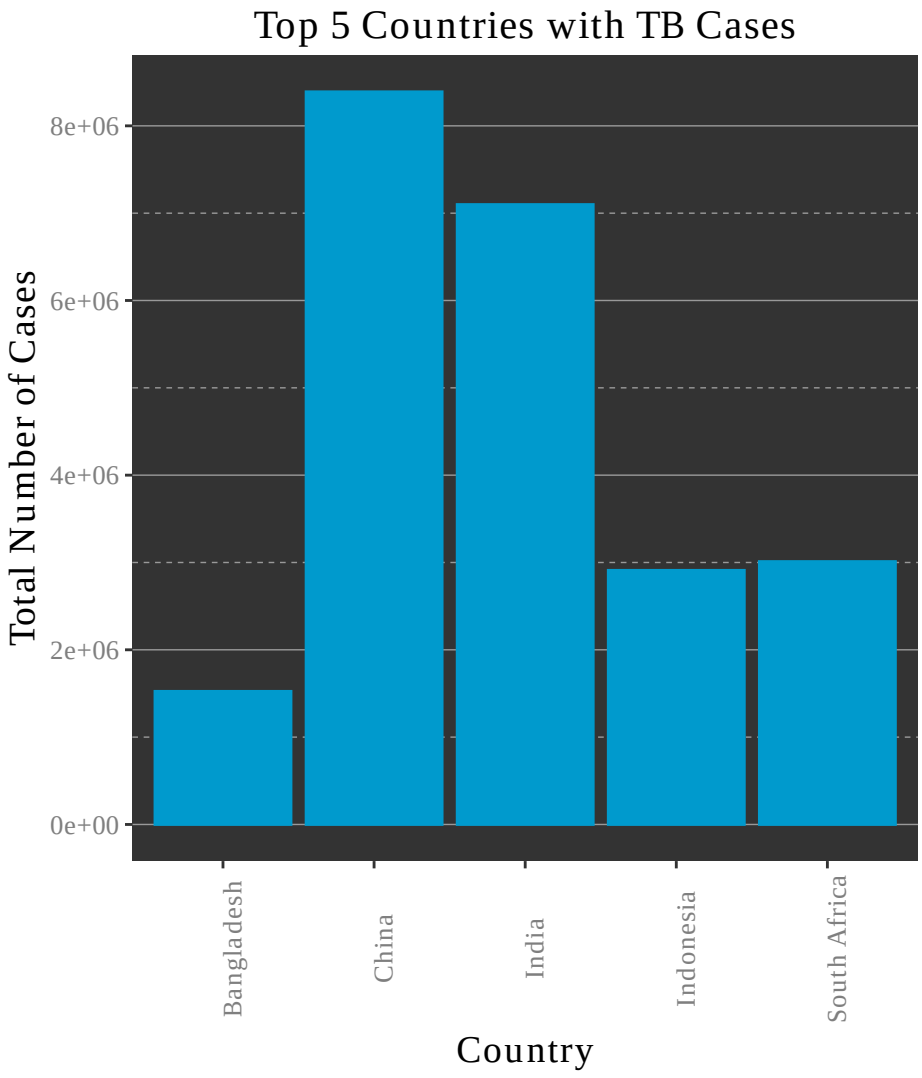
The top 5 countries with the highest number of TB cases are shown in the following bar chart. It is interesting to see that 4 of these countries belong to Asia.

```
top5countries = who5 %>%
  group_by(country) %>%
  tally(cases) %>%
  top_n(5)

library(ggplot2)

top5plot = ggplot(data=top5countries, aes(x=country, y=n)) +
  geom_bar(stat="identity", colour = "deepskyblue3", fill = "deepskyblue3") +
  ggtitle("Top 5 Countries with TB Cases") + ylab("Total Number of Cases") +   xlab("Country") + labs(f
  theme(
    panel.background = element_rect(fill = "grey20"),
    panel.grid.major = element_line(colour = "grey60", size=0.25),
    panel.grid.minor = element_line(colour = "grey60", linetype = "dashed"),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    plot.title = element_text(size=15, hjust=0.5, vjust = 1.5, family = "Palatino", colour = "Black", ma
    axis.title.x = element_text(size=14, vjust = -0.3, family = "Palatino", colour = "Black", margin = m
    axis.text.x = element_text(size = 10, family = "Palatino", colour = "grey50", angle = 90),
    axis.title.y = element_text(size=14, vjust = 1.5, family = "Palatino", colour = "Black", margin = ma
    axis.text.y = element_text(size = 10, family = "Palatino", colour = "grey50"),
  )

top5plot
```



2(f)

Constructing the table.

```
Site = c("facebook", "myspace", "snapchat", "twitter", "tiktok")
U30.F = c(30,1,6,18,44)
U30_M = c(25,2,5,23,60)
O30.F = c(66,3,3,12,2)
O30.M = c(58,6,2,28,7)
siteDemo = data.frame(Site,U30.F,U30_M,O30.F,O30.M)
siteDemo
```

```
##       Site U30.F U30_M O30.F O30.M
## 1 facebook    30    25    66    58
## 2  myspace     1     2     3     6
## 3 snapchat     6     5     3     2
## 4  twitter    18    23    12    28
## 5   tiktok    44    60     2     7
```

Tidying the dataset.

```
tidysiteDemo = siteDemo %>%
  gather(U30.F:O30.M,key = "UsersAge_Sex",value = "Count", na.rm = TRUE) %>%
  mutate(UsersAge_Sex = stringr::str_replace(UsersAge_Sex, "U30_M", "U30.M")) %>%
  separate(UsersAge_Sex, c("AgeGroup","Gender"))
tidysiteDemo
```

```
##         Site AgeGroup Gender Count
## 1   facebook      U30      F    30
## 2    myspace      U30      F     1
## 3   snapchat      U30      F     6
## 4    twitter      U30      F    18
## 5     tiktok      U30      F    44
## 6   facebook      U30      M    25
## 7    myspace      U30      M     2
## 8   snapchat      U30      M     5
## 9    twitter      U30      M    23
## 10    tiktok      U30      M    60
## 11 facebook      O30      F    66
## 12   myspace      O30      F     3
## 13 snapchat      O30      F     3
## 14   twitter      O30      F    12
## 15    tiktok      O30      F     2
## 16 facebook      O30      M    58
## 17   myspace      O30      M     6
## 18 snapchat      O30      M     2
## 19   twitter      O30      M    28
## 20    tiktok      O30      M     7
```