Assignment3

Jaswanth Gade

September 27, 2018

Question 1. For this question you will be using the dplyr package to manipulate and clean up a dataset called msleep (mammals sleep) that is available on the course webpage (at https://scads.eecs.wsu.edu/wp-content/uploads/2017/10/msleep_ggplot2.csv). The dataset contains the sleep times and weights for a set of mammals. It has 83 rows and 11 variables. Here is a description of the variables:

Load the data into R, and check the first few rows for abnormalities. You will likely notice several.

```
ysleep <- read.csv("https://scads.eecs.wsu.edu/wp-content/uploads/2017/10/msleep_ggplot2.csv")
```

Below are the tasks to perform. Use select() to print the head of the columns with a title including "sleep".

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
sleepData <- select(ysleep, name, sleep_total)
head(sleepData)
```

```
##                         name sleep_total
## 1                    Cheetah        12.1
## 2                 Owl monkey        17.0
## 3            Mountain beaver        14.4
## 4 Greater short-tailed shrew        14.9
## 5                        Cow         4.0
## 6           Three-toed sloth        14.4
```

1(a).Use filter() to count the number of animals which weigh over 50 kilograms and sleep more than 6 hours a day

```
filter(ysleep, bodywt>50, sleep_total >6)
```

```
##              name        genus    vore        order conservation
## 1       Gray seal Halichoerus   carni    Carnivora           lc
## 2           Human         Homo    omni     Primates         <NA>
## 3      Chimpanzee          Pan    omni     Primates         <NA>
## 4           Tiger     Panthera   carni    Carnivora           en
## 5          Jaguar     Panthera   carni    Carnivora           nt
## 6            Lion     Panthera   carni    Carnivora           vu
## 7 Giant armadillo   Priodontes insecti    Cingulata           en
## 8             Pig          Sus    omni Artiodactyla domesticated
##   sleep_total sleep_rem sleep_cycle awake brainwt  bodywt
## 1         6.2       1.5          NA  17.8   0.325  85.000
## 2         8.0       1.9    1.500000  16.0   1.320  62.000
```

```
## 3         9.7       1.4    1.416667  14.3    0.440  52.200
## 4        15.8        NA          NA   8.2       NA 162.564
## 5        10.4        NA          NA  13.6    0.157 100.000
## 6        13.5        NA          NA  10.5       NA 161.499
## 7        18.1       6.1          NA   5.9    0.081  60.000
## 8         9.1       2.4    0.500000  14.9    0.180  86.250
```

1(b).Use piping (%>%), select() and arrange() to print the name, order, sleep time and bodyweight of the animals with the top 6 sleep times, in order of sleep time.

```
ysleep %>%
    select(name, order, sleep_total,bodywt) %>%
    arrange(desc(sleep_total)) %>%
    top_n(6,sleep_total)
```

```
##                     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).Use mutate to add 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. If you think they might be useful, feel free to extract more features than these, and describe what they are? SOlution:

```
ysleep %>%
    mutate(wt_ratio = brainwt / bodywt ,
           rem_ratio = sleep_rem/sleep_total,
           total_time= sleep_total + awake)
```

```
##                                name         genus    vore           order
## 1                           Cheetah      Acinonyx   carni       Carnivora
## 2                        Owl monkey         Aotus    omni        Primates
## 3                   Mountain beaver    Aplodontia   herbi        Rodentia
## 4        Greater short-tailed shrew       Blarina    omni    Soricomorpha
## 5                               Cow           Bos   herbi    Artiodactyla
## 6                  Three-toed sloth     Bradypus   herbi          Pilosa
## 7                 Northern fur seal   Callorhinus   carni       Carnivora
## 8                      Vesper mouse       Calomys    <NA>        Rodentia
## 9                               Dog         Canis   carni       Carnivora
## 10                         Roe deer     Capreolus   herbi    Artiodactyla
## 11                             Goat         Capri   herbi    Artiodactyla
## 12                       Guinea pig         Cavis   herbi        Rodentia
## 13                           Grivet Cercopithecus    omni        Primates
## 14                       Chinchilla    Chinchilla   herbi        Rodentia
## 15                  Star-nosed mole      Condylura    omni    Soricomorpha
## 16        African giant pouched rat    Cricetomys    omni        Rodentia
## 17        Lesser short-tailed shrew     Cryptotis    omni    Soricomorpha
## 18             Long-nosed armadillo       Dasypus   carni       Cingulata
## 19                       Tree hyrax   Dendrohyrax   herbi      Hyracoidea
## 20           North American Opossum     Didelphis    omni Didelphimorphia
## 21                   Asian elephant       Elephas   herbi     Proboscidea
## 22                    Big brown bat     Eptesicus insecti      Chiroptera
## 23                            Horse         Equus   herbi  Perissodactyla
```

```
## 24                         Donkey         Equus   herbi  Perissodactyla
## 25              European hedgehog     Erinaceus    omni  Erinaceomorpha
## 26                   Patas monkey  Erythrocebus    omni        Primates
## 27       Western american chipmunk      Eutamias   herbi        Rodentia
## 28                   Domestic cat         Felis   carni       Carnivora
## 29                         Galago        Galago    omni        Primates
## 30                        Giraffe       Giraffa   herbi    Artiodactyla
## 31                    Pilot whale Globicephalus   carni         Cetacea
## 32                      Gray seal  Haliochoerus   carni       Carnivora
## 33                     Gray hyrax   Heterohyrax   herbi      Hyracoidea
## 34                          Human          Homo    omni        Primates
## 35                 Mongoose lemur         Lemur   herbi        Primates
## 36               African elephant     Loxodonta   herbi     Proboscidea
## 37            Thick-tailed opposum    Lutreolina   carni Didelphimorphia
## 38                        Macaque        Macaca    omni        Primates
## 39               Mongolian gerbil      Meriones   herbi        Rodentia
## 40                 Golden hamster  Mesocricetus   herbi        Rodentia
## 41                          Vole       Microtus   herbi        Rodentia
## 42                    House mouse           Mus   herbi        Rodentia
## 43               Little brown bat        Myotis insecti      Chiroptera
## 44            Round-tailed muskrat      Neofiber   herbi        Rodentia
## 45                     Slow loris     Nyctibeus   carni        Primates
## 46                           Degu       Octodon   herbi        Rodentia
## 47      Northern grasshopper mouse     Onychomys   carni        Rodentia
## 48                         Rabbit   Oryctolagus   herbi      Lagomorpha
## 49                          Sheep          Ovis   herbi    Artiodactyla
## 50                     Chimpanzee           Pan    omni        Primates
## 51                          Tiger      Panthera   carni       Carnivora
## 52                         Jaguar      Panthera   carni       Carnivora
## 53                           Lion      Panthera   carni       Carnivora
## 54                         Baboon         Papio    omni        Primates
## 55                Desert hedgehog   Paraechinus    <NA>  Erinaceomorpha
## 56                          Potto  Perodicticus    omni        Primates
## 57                     Deer mouse    Peromyscus    <NA>        Rodentia
## 58                      Phalanger     Phalanger    <NA>   Diprotodontia
## 59                    Caspian seal         Phoca   carni       Carnivora
## 60                Common porpoise      Phocoena   carni         Cetacea
## 61                        Potoroo      Potorous   herbi   Diprotodontia
## 62                Giant armadillo    Priodontes insecti       Cingulata
## 63                     Rock hyrax      Procavia    <NA>      Hyracoidea
## 64                 Laboratory rat        Rattus   herbi        Rodentia
## 65           African striped mouse     Rhabdomys    omni        Rodentia
## 66                 Squirrel monkey       Saimiri    omni        Primates
## 67           Eastern american mole      Scalopus insecti     Soricomorpha
## 68                     Cotton rat      Sigmodon   herbi        Rodentia
## 69                       Mole rat        Spalax    <NA>        Rodentia
## 70          Arctic ground squirrel  Spermophilus   herbi        Rodentia
## 71 Thirteen-lined ground squirrel  Spermophilus   herbi        Rodentia
## 72 Golden-mantled ground squirrel  Spermophilus   herbi        Rodentia
## 73                     Musk shrew        Suncus    <NA>    Soricomorpha
## 74                            Pig           Sus    omni    Artiodactyla
## 75             Short-nosed echidna  Tachyglossus insecti     Monotremata
## 76       Eastern american chipmunk        Tamias   herbi        Rodentia
## 77                 Brazilian tapir       Tapirus   herbi  Perissodactyla
```

```
## 78                        Tenrec       Tenrec    omni    Afrosoricida
## 79                    Tree shrew       Tupaia    omni      Scandentia
## 80           Bottle-nosed dolphin     Tursiops   carni         Cetacea
## 81                         Genet      Genetta   carni       Carnivora
## 82                    Arctic fox       Vulpes   carni       Carnivora
## 83                       Red fox       Vulpes   carni       Carnivora
##    conservation sleep_total sleep_rem sleep_cycle awake brainwt   bodywt
## 1            lc        12.1        NA          NA 11.90      NA   50.000
## 2          <NA>        17.0       1.8          NA  7.00 0.01550    0.480
## 3            nt        14.4       2.4          NA  9.60      NA    1.350
## 4            lc        14.9       2.3   0.1333333  9.10 0.00029    0.019
## 5  domesticated         4.0       0.7   0.6666667 20.00 0.42300  600.000
## 6          <NA>        14.4       2.2   0.7666667  9.60      NA    3.850
## 7            vu         8.7       1.4   0.3833333 15.30      NA   20.490
## 8          <NA>         7.0        NA          NA 17.00      NA    0.045
## 9  domesticated        10.1       2.9   0.3333333 13.90 0.07000   14.000
## 10           lc         3.0        NA          NA 21.00 0.09820   14.800
## 11           lc         5.3       0.6          NA 18.70 0.11500   33.500
## 12 domesticated         9.4       0.8   0.2166667 14.60 0.00550    0.728
## 13           lc        10.0       0.7          NA 14.00      NA    4.750
## 14 domesticated        12.5       1.5   0.1166667 11.50 0.00640    0.420
## 15           lc        10.3       2.2          NA 13.70 0.00100    0.060
## 16         <NA>         8.3       2.0          NA 15.70 0.00660    1.000
## 17           lc         9.1       1.4   0.1500000 14.90 0.00014    0.005
## 18           lc        17.4       3.1   0.3833333  6.60 0.01080    3.500
## 19           lc         5.3       0.5          NA 18.70 0.01230    2.950
## 20           lc        18.0       4.9   0.3333333  6.00 0.00630    1.700
## 21           en         3.9        NA          NA 20.10 4.60300 2547.000
## 22           lc        19.7       3.9   0.1166667  4.30 0.00030    0.023
## 23 domesticated         2.9       0.6   1.0000000 21.10 0.65500  521.000
## 24 domesticated         3.1       0.4          NA 20.90 0.41900  187.000
## 25           lc        10.1       3.5   0.2833333 13.90 0.00350    0.770
## 26           lc        10.9       1.1          NA 13.10 0.11500   10.000
## 27         <NA>        14.9        NA          NA  9.10      NA    0.071
## 28 domesticated        12.5       3.2   0.4166667 11.50 0.02560    3.300
## 29         <NA>         9.8       1.1   0.5500000 14.20 0.00500    0.200
## 30           cd         1.9       0.4          NA 22.10      NA  899.995
## 31           cd         2.7       0.1          NA 21.35      NA  800.000
## 32           lc         6.2       1.5          NA 17.80 0.32500   85.000
## 33           lc         6.3       0.6          NA 17.70 0.01227    2.625
## 34         <NA>         8.0       1.9   1.5000000 16.00 1.32000   62.000
## 35           vu         9.5       0.9          NA 14.50      NA    1.670
## 36           vu         3.3        NA          NA 20.70 5.71200 6654.000
## 37           lc        19.4       6.6          NA  4.60      NA    0.370
## 38         <NA>        10.1       1.2   0.7500000 13.90 0.17900    6.800
## 39           lc        14.2       1.9          NA  9.80      NA    0.053
## 40           en        14.3       3.1   0.2000000  9.70 0.00100    0.120
## 41         <NA>        12.8        NA          NA 11.20      NA    0.035
## 42           nt        12.5       1.4   0.1833333 11.50 0.00040    0.022
## 43         <NA>        19.9       2.0   0.2000000  4.10 0.00025    0.010
## 44           nt        14.6        NA          NA  9.40      NA    0.266
## 45         <NA>        11.0        NA          NA 13.00 0.01250    1.400
## 46           lc         7.7       0.9          NA 16.30      NA    0.210
## 47           lc        14.5        NA          NA  9.50      NA    0.028
```

```
## 48 domesticated         8.4       0.9   0.4166667 15.60 0.01210    2.500
## 49 domesticated         3.8       0.6          NA 20.20 0.17500   55.500
## 50         <NA>         9.7       1.4   1.4166667 14.30 0.44000   52.200
## 51           en        15.8        NA          NA  8.20      NA  162.564
## 52           nt        10.4        NA          NA 13.60 0.15700  100.000
## 53           vu        13.5        NA          NA 10.50      NA  161.499
## 54         <NA>         9.4       1.0   0.6666667 14.60 0.18000   25.235
## 55           lc        10.3       2.7          NA 13.70 0.00240    0.550
## 56           lc        11.0        NA          NA 13.00      NA    1.100
## 57         <NA>        11.5        NA          NA 12.50      NA    0.021
## 58         <NA>        13.7       1.8          NA 10.30 0.01140    1.620
## 59           vu         3.5       0.4          NA 20.50      NA   86.000
## 60           vu         5.6        NA          NA 18.45      NA   53.180
## 61         <NA>        11.1       1.5          NA 12.90      NA    1.100
## 62           en        18.1       6.1          NA  5.90 0.08100   60.000
## 63           lc         5.4       0.5          NA 18.60 0.02100    3.600
## 64           lc        13.0       2.4   0.1833333 11.00 0.00190    0.320
## 65         <NA>         8.7        NA          NA 15.30      NA    0.044
## 66         <NA>         9.6       1.4          NA 14.40 0.02000    0.743
## 67           lc         8.4       2.1   0.1666667 15.60 0.00120    0.075
## 68         <NA>        11.3       1.1   0.1500000 12.70 0.00118    0.148
## 69         <NA>        10.6       2.4          NA 13.40 0.00300    0.122
## 70           lc        16.6        NA          NA  7.40 0.00570    0.920
## 71           lc        13.8       3.4   0.2166667 10.20 0.00400    0.101
## 72           lc        15.9       3.0          NA  8.10      NA    0.205
## 73         <NA>        12.8       2.0   0.1833333 11.20 0.00033    0.048
## 74 domesticated         9.1       2.4   0.5000000 14.90 0.18000   86.250
## 75         <NA>         8.6        NA          NA 15.40 0.02500    4.500
## 76         <NA>        15.8        NA          NA  8.20      NA    0.112
## 77           vu         4.4       1.0   0.9000000 19.60 0.16900  207.501
## 78         <NA>        15.6       2.3          NA  8.40 0.00260    0.900
## 79         <NA>         8.9       2.6   0.2333333 15.10 0.00250    0.104
## 80         <NA>         5.2        NA          NA 18.80      NA  173.330
## 81         <NA>         6.3       1.3          NA 17.70 0.01750    2.000
## 82         <NA>        12.5        NA          NA 11.50 0.04450    3.380
## 83         <NA>         9.8       2.4   0.3500000 14.20 0.05040    4.230
##        wt_ratio  rem_ratio total_time
## 1            NA         NA      24.00
## 2  0.0322916667 0.10588235      24.00
## 3            NA 0.16666667      24.00
## 4  0.0152631579 0.15436242      24.00
## 5  0.0007050000 0.17500000      24.00
## 6            NA 0.15277778      24.00
## 7            NA 0.16091954      24.00
## 8            NA         NA      24.00
## 9  0.0050000000 0.28712871      24.00
## 10 0.0066351351         NA      24.00
## 11 0.0034328358 0.11320755      24.00
## 12 0.0075549451 0.08510638      24.00
## 13           NA 0.07000000      24.00
## 14 0.0152380952 0.12000000      24.00
## 15 0.0166666667 0.21359223      24.00
## 16 0.0066000000 0.24096386      24.00
## 17 0.0280000000 0.15384615      24.00
```

```
## 18 0.0030857143 0.17816092      24.00
## 19 0.0041694915 0.09433962      24.00
## 20 0.0037058824 0.27222222      24.00
## 21 0.0018072242         NA      24.00
## 22 0.0130434783 0.19796954      24.00
## 23 0.0012571977 0.20689655      24.00
## 24 0.0022406417 0.12903226      24.00
## 25 0.0045454545 0.34653465      24.00
## 26 0.0115000000 0.10091743      24.00
## 27           NA         NA      24.00
## 28 0.0077575758 0.25600000      24.00
## 29 0.0250000000 0.11224490      24.00
## 30           NA 0.21052632      24.00
## 31           NA 0.03703704      24.05
## 32 0.0038235294 0.24193548      24.00
## 33 0.0046742857 0.09523810      24.00
## 34 0.0212903226 0.23750000      24.00
## 35           NA 0.09473684      24.00
## 36 0.0008584310         NA      24.00
## 37           NA 0.34020619      24.00
## 38 0.0263235294 0.11881188      24.00
## 39           NA 0.13380282      24.00
## 40 0.0083333333 0.21678322      24.00
## 41           NA         NA      24.00
## 42 0.0181818182 0.11200000      24.00
## 43 0.0250000000 0.10050251      24.00
## 44           NA         NA      24.00
## 45 0.0089285714         NA      24.00
## 46           NA 0.11688312      24.00
## 47           NA         NA      24.00
## 48 0.0048400000 0.10714286      24.00
## 49 0.0031531532 0.15789474      24.00
## 50 0.0084291188 0.14432990      24.00
## 51           NA         NA      24.00
## 52 0.0015700000         NA      24.00
## 53           NA         NA      24.00
## 54 0.0071329503 0.10638298      24.00
## 55 0.0043636364 0.26213592      24.00
## 56           NA         NA      24.00
## 57           NA         NA      24.00
## 58 0.0070370370 0.13138686      24.00
## 59           NA 0.11428571      24.00
## 60           NA         NA      24.05
## 61           NA 0.13513514      24.00
## 62 0.0013500000 0.33701657      24.00
## 63 0.0058333333 0.09259259      24.00
## 64 0.0059375000 0.18461538      24.00
## 65           NA         NA      24.00
## 66 0.0269179004 0.14583333      24.00
## 67 0.0160000000 0.25000000      24.00
## 68 0.0079729730 0.09734513      24.00
## 69 0.0245901639 0.22641509      24.00
## 70 0.0061956522         NA      24.00
## 71 0.0396039604 0.24637681      24.00
```

```
## 72           NA 0.18867925      24.00
## 73 0.0068750000 0.15625000      24.00
## 74 0.0020869565 0.26373626      24.00
## 75 0.0055555556         NA      24.00
## 76           NA         NA      24.00
## 77 0.0008144539 0.22727273      24.00
## 78 0.0028888889 0.14743590      24.00
## 79 0.0240384615 0.29213483      24.00
## 80           NA         NA      24.00
## 81 0.0087500000 0.20634921      24.00
## 82 0.0131656805         NA      24.00
## 83 0.0119148936 0.24489796      24.00
```

I extracted an another column to check if the "sleep_total" and "awake" variable together make up to 24 hours.That is checking if the animal is in observation for 24 hours or not.This variable gives out the information of the time of which the mammal is in observatoin. This calculation is stored in the variable "Total_time".

1(d).Use group_by() and summarize() to display the average, min and max sleep times for each order. Remember to use ungroup() when you are done.

```
ysleep %>%
    group_by(order) %>%
    summarise(avg_sleep = mean(sleep_total),
              min_sleep = min(sleep_total),
              max_sleep = max(sleep_total),
              total = n())%>%
ungroup()
```

```
## # A tibble: 19 x 5
##    order           avg_sleep min_sleep max_sleep total
##    <fct>               <dbl>     <dbl>     <dbl> <int>
##  1 Afrosoricida        15.6       15.6      15.6     1
##  2 Artiodactyla         4.52       1.9       9.1     6
##  3 Carnivora           10.1        3.5      15.8    12
##  4 Cetacea              4.5        2.7       5.6     3
##  5 Chiroptera          19.8       19.7      19.9     2
##  6 Cingulata           17.8       17.4      18.1     2
##  7 Didelphimorphia     18.7       18        19.4     2
##  8 Diprotodontia       12.4       11.1      13.7     2
##  9 Erinaceomorpha      10.2       10.1      10.3     2
## 10 Hyracoidea           5.67       5.3       6.3     3
## 11 Lagomorpha           8.4        8.4       8.4     1
## 12 Monotremata          8.6        8.6       8.6     1
## 13 Perissodactyla       3.47       2.9       4.4     3
## 14 Pilosa              14.4       14.4      14.4     1
## 15 Primates            10.5        8        17      12
## 16 Proboscidea          3.6        3.3       3.9     2
## 17 Rodentia            12.5        7        16.6    22
## 18 Scandentia           8.9        8.9       8.9     1
## 19 Soricomorpha        11.1        8.4      14.9     5
```

1(e).Make a copy of your dataframe, and use group_by() and mutate() to impute the missing brain weights as the average wt_ratio for that animal's order times the animal's weight. Make a second copy of your dataframe, but this time use group_by() and mutate() to impute missing brain weights with the average brain weight for that animal's order. What assumptions do these data filling methods make? Which is the best way to impute the data, or do you see a better way, and why? You may impute or remove other variables as you find appropriate. Briefly explain your decisions. Solution: Missing values which are imputed by the average brainwt of the same order mammals is more appropriate compared to the wt_ratio variable. Replacing the missing values with the same kind of feature valu by assuming their average is often a better choice compared to the wt_ratio variable.

```
xsleep = ysleep
xsleep1 = xsleep %>%
   group_by(order) %>%
   mutate( brainwt= ifelse(is.na(brainwt), ((mean(brainwt, na.rm  = TRUE)/mean(bodywt, na.rm = TRUE)) *

  xsleep2 = xsleep %>%
   group_by(order) %>%
   mutate(brainwt= ifelse(is.na(brainwt),mean(brainwt, na.rm= TRUE), brainwt))
```

Question 2. For this question, you will first need to read section 12.6 in the R for Data Science book, here (http://r4ds.had.co.nz/tidy-data.htmlcase-study). Grab the dataset from the tidyr package, and tidy it as shown in the case study before answering the following questions.

```
library(tidyr)
who <-tidyr::who
who1 <- who %>%
gather(key, value, new_sp_m014:newrel_f65, na.rm = FALSE) %>%
mutate(key = stringr::str_replace(key, "newrel", "new_rel"))
who1
```

```
## # A tibble: 405,440 x 6
##    country     iso2  iso3   year key          value
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1980 new_sp_m014     NA
##  2 Afghanistan AF    AFG    1981 new_sp_m014     NA
##  3 Afghanistan AF    AFG    1982 new_sp_m014     NA
##  4 Afghanistan AF    AFG    1983 new_sp_m014     NA
##  5 Afghanistan AF    AFG    1984 new_sp_m014     NA
##  6 Afghanistan AF    AFG    1985 new_sp_m014     NA
##  7 Afghanistan AF    AFG    1986 new_sp_m014     NA
##  8 Afghanistan AF    AFG    1987 new_sp_m014     NA
##  9 Afghanistan AF    AFG    1988 new_sp_m014     NA
## 10 Afghanistan AF    AFG    1989 new_sp_m014     NA
## # ... with 405,430 more rows
```

2(a).Explain why this line mutate(key = stringr::str_replace(key, "newrel", "new_rel")) is necessary to properly tidy the data. What happens if you skip this line?}

```
who2 <- who1 %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel"))
who2
```

```
## # A tibble: 405,440 x 6
##    country     iso2  iso3   year key          value
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1980 new_sp_m014     NA
##  2 Afghanistan AF    AFG    1981 new_sp_m014     NA
##  3 Afghanistan AF    AFG    1982 new_sp_m014     NA
##  4 Afghanistan AF    AFG    1983 new_sp_m014     NA
##  5 Afghanistan AF    AFG    1984 new_sp_m014     NA
##  6 Afghanistan AF    AFG    1985 new_sp_m014     NA
```

```
##   7 Afghanistan AF    AFG    1986 new_sp_m014    NA
##   8 Afghanistan AF    AFG    1987 new_sp_m014    NA
##   9 Afghanistan AF    AFG    1988 new_sp_m014    NA
##  10 Afghanistan AF    AFG    1989 new_sp_m014    NA
## # ... with 405,430 more rows
```

Solution: skipping this line would make it difficult to figure out the patients with the disease being relapsed. To gain a consistency in the data by making it easier to diiferentiate between the other factors. The variable newrel violates consistency, so it has been modified to follow a single pattern and also to maintain consistency among the data. new_sp= smear positive new_sn= smear negative new_ep= extrapulmonry new_rel= relapsed

2(b). How many entries are removed from the dataset when you set na.rm to true in the gather command (in this dataset). How else could those NA values be handled? Among these options, which do you think is the best way to handle those missing values for this dataset, and why?}

```r
who1 <- who %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm = 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    1998 new_sp_m014     30
##  3 Afghanistan AF    AFG    1999 new_sp_m014      8
##  4 Afghanistan AF    AFG    2000 new_sp_m014     52
##  5 Afghanistan AF    AFG    2001 new_sp_m014    129
##  6 Afghanistan AF    AFG    2002 new_sp_m014     90
##  7 Afghanistan AF    AFG    2003 new_sp_m014    127
##  8 Afghanistan AF    AFG    2004 new_sp_m014    139
##  9 Afghanistan AF    AFG    2005 new_sp_m014    151
## 10 Afghanistan AF    AFG    2006 new_sp_m014    193
## # ... with 76,036 more rows
```

```r
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
```

```r
ncol(who)
```

```
## [1] 60
```

```r
ncol(who1)
```

```
## [1] 6
```

solution: There are total of 60 columns in "who" and 6 columns "who1". Here we removed 56 columns and there is an addition of 2 columns to who1 which makes the number of columns in who1 to 6.

2(c). Explain the difference between an explicit and implicit missing value, in general. Can you find any implicit missing values in this dataset, if so where?

Solution:

```r
sum(complete.cases(who))
```

```
## [1] 0
```

```r
who[complete.cases(who),]
```

```
## # A tibble: 0 x 60
## # ... with 60 variables: country <chr>, iso2 <chr>, iso3 <chr>,
## #   year <int>, new_sp_m014 <int>, new_sp_m1524 <int>, new_sp_m2534 <int>,
## #   new_sp_m3544 <int>, 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>
```

I cannot find any mnissing values as the complete function gives the output same as the number of rows in dataset. Also the code in the chunk returns value zero that says there are no implicit values. Explicit values (shown with the variables NA) or implicit values (not even present in the dataset).It is that a explicit value is the presence of an absence and an implicit value is the absence of a presence. The spread() function can turn implicit values explicit (year, return for columns) while the gather() function turns explicit values implicit. Explicitly, i.e. flagged with NA. Implicitly, the absence of the data

2(d). Looking at the features (country, year, var, sex, age, cases) in the tidied data, are they all appropriately typed? Are there any features you think would be better suited as a different type? Why or why not?}

Solution:

```r
who3 <- who2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
who3
```

```
## # A tibble: 405,440 x 8
##    country     iso2  iso3   year new   type  sexage value
##    <chr>       <chr> <chr> <int> <chr> <chr> <chr>  <int>
```

```
##   1 Afghanistan AF    AFG    1980 new   sp    m014       NA
##   2 Afghanistan AF    AFG    1981 new   sp    m014       NA
##   3 Afghanistan AF    AFG    1982 new   sp    m014       NA
##   4 Afghanistan AF    AFG    1983 new   sp    m014       NA
##   5 Afghanistan AF    AFG    1984 new   sp    m014       NA
##   6 Afghanistan AF    AFG    1985 new   sp    m014       NA
##   7 Afghanistan AF    AFG    1986 new   sp    m014       NA
##   8 Afghanistan AF    AFG    1987 new   sp    m014       NA
##   9 Afghanistan AF    AFG    1988 new   sp    m014       NA
##  10 Afghanistan AF    AFG    1989 new   sp    m014       NA
## # ... with 405,430 more rows
```

```r
who4 <- who3 %>%
  select(-new, -iso2, -iso3)
  who5 <- who4 %>%
  separate(sexage, c("sex", "age"), sep = 1)
who5
```

```
## # A tibble: 405,440 x 6
##    country      year type  sex   age   value
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1980 sp    m     014      NA
##  2 Afghanistan  1981 sp    m     014      NA
##  3 Afghanistan  1982 sp    m     014      NA
##  4 Afghanistan  1983 sp    m     014      NA
##  5 Afghanistan  1984 sp    m     014      NA
##  6 Afghanistan  1985 sp    m     014      NA
##  7 Afghanistan  1986 sp    m     014      NA
##  8 Afghanistan  1987 sp    m     014      NA
##  9 Afghanistan  1988 sp    m     014      NA
## 10 Afghanistan  1989 sp    m     014      NA
## # ... with 405,430 more rows
```

Almost all the features are apprpriately typed except the age. The "age" feature here is character type which can be modified to an integer type.

2(e). Explain in your own words what a gather operation is, and give an example of a situation when it might be useful. Do the same for spread.}

Solution: Gather()- Reshaping rows to columns

Description: There are situations our data is considered unstacked and a common attribute is spread out across columns. To reform and keeping these common attributes gathered together into a single variable, the gather() function will take multiple columns and combine them into key-value pairs, forming duplicates of all other columns as needed.

EXAMPLE: mpg cyl disp hp drat wt qsec vs am gear carb

Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4

Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4

Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1

Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1

Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2

Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

After gather

car attribute value

1 Mazda RX4 mpg 21.0

2 Mazda RX4 Wag mpg 21.0

3 Datsun 710 mpg 22.8

4 Hornet 4 Drive mpg 21.4

5 Hornet Sportabout mpg 18.7

6 Valiant mpg 18.1

Spread()- Reshaping columns to rows

Description: There are situations when we are required to turn long formatted data into wide formatted data. The spread() function spreads a key-value pair across multiple columns.

EXAMPLE: After Gather

0 country quarter growth

1 A q1__2017 0.03

2 B q1__2017 0.05

3 C q1__2017 0.01

4 A q2__2017 0.05

5 B q2__2017 0.07

6 C q2__2017 0.02

7 A q3__2017 0.04

8 B q3__2017 0.05

9 C q3__2017 0.01

10 A q4__2017 0.03

11 B q4__2017 0.02

12 C q4__2017 0.04

After Spread

0 country q1__2017 q2__2017 q3__2017 q4__2017

1 A 0.03 0.05 0.04 0.03

2 B 0.05 0.07 0.05 0.02

3 C 0.01 0.02 0.01 0.04

f) Generate an informative visualization, which shows something about the data. Give a brief description of what it shows, and why you thought it was interesting.

```r
count(who5, sex)
```

```
## # A tibble: 2 x 2
##   sex        n
##   <chr>  <int>
## 1 f     202720
## 2 m     202720
```

```r
count(who5, type)
```

```
## # A tibble: 4 x 2
##   type       n
##   <chr>  <int>
## 1 ep    101360
## 2 rel   101360
## 3 sn    101360
## 4 sp    101360
```

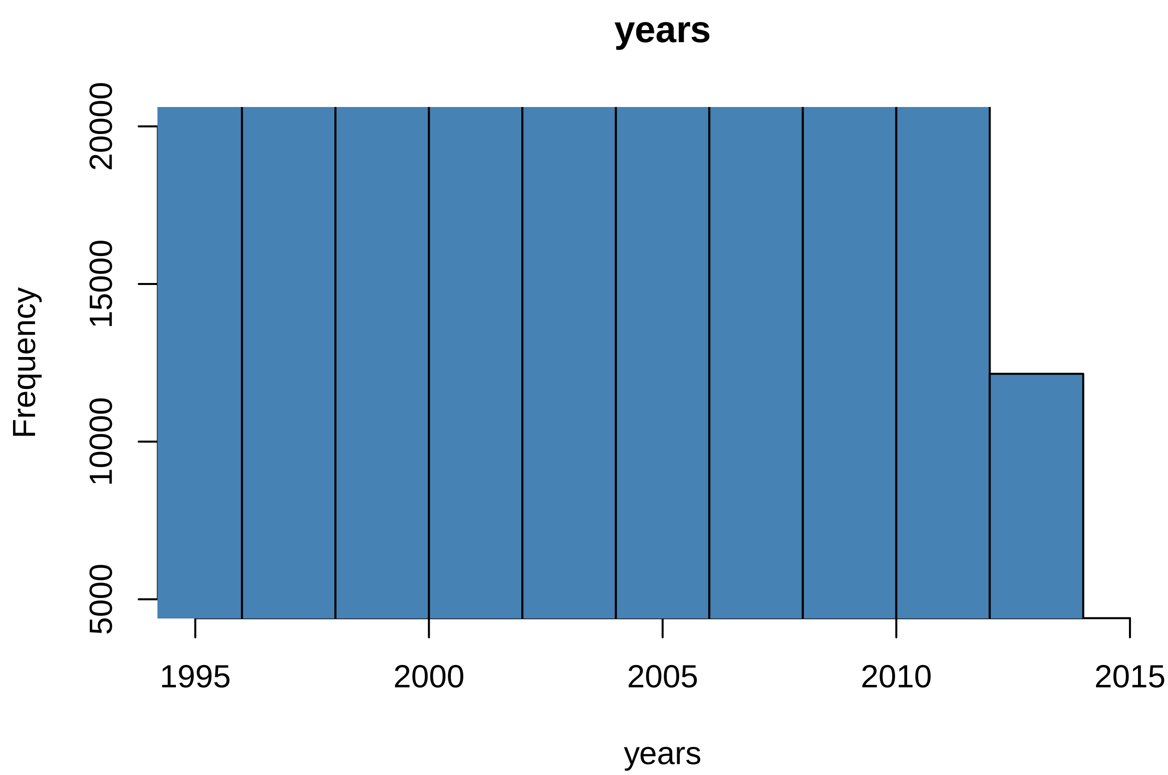
```r
count(who5, country)
```

```
## # A tibble: 219 x 2
##    country                 n
##    <chr>               <int>
##  1 Afghanistan          1904
##  2 Albania              1904
##  3 Algeria              1904
##  4 American Samoa       1904
##  5 Andorra              1904
##  6 Angola               1904
##  7 Anguilla             1904
##  8 Antigua and Barbuda  1904
##  9 Argentina            1904
## 10 Armenia              1904
## # ... with 209 more rows
```

```r
count(who5, country,type)
```

```
## # A tibble: 876 x 3
##    country     type      n
##    <chr>       <chr> <int>
##  1 Afghanistan ep      476
##  2 Afghanistan rel     476
##  3 Afghanistan sn      476
##  4 Afghanistan sp      476
##  5 Albania     ep      476
##  6 Albania     rel     476
##  7 Albania     sn      476
##  8 Albania     sp      476
##  9 Algeria     ep      476
## 10 Algeria     rel     476
## # ... with 866 more rows
```

```r
hist(who5$year, col='steelblue',main='years',xlim = c(1995,2015),ylim=c(5000,20000), xlab='years')
```



In the above observations, i have taken the number of variables that are present in the dataset by using the count() function. The following variables are considered in the above chunk:

1. The number of male and female persons that are effected. As seen, the count of male is slightly higher than the female count.
2. The number of people that have gone through smear positive, smear negative, extrapulmonary, relapse. As we can observe in the table, the count of people that have smear positive is way higher than any variable.
3. In the next visualization, i have taken the count of country through which we can tell the number of people that have gone through the effect from eaxh country.
4. In the fourth observation i have combined the 2nd and 3rd observations. I have shown you the count of the people in each country with respect to the four types available.
5. In the fifth observation that i have taken, i have plotted a histogram to the years and the total count of cases. That is, here i have shown the total number of people around the world that are effected in each year. As we can observe, the count of people effected is high in between the years 2010 to 2012.