Assignment 3

Tayyab Munir - 11716089

9/23/2020

Question 1

```
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
```

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

Before you begin, print the first few values of the columns with a header including "sleep". (head(), head())

```
head(msleep)
```

```
##                         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
## 1        12.1        NA          NA  11.9      NA  50.000
## 2        17.0       1.8          NA   7.0 0.01550   0.480
## 3        14.4       2.4          NA   9.6      NA   1.350
## 4        14.9       2.3   0.1333333   9.1 0.00029   0.019
## 5         4.0       0.7   0.6666667  20.0 0.42300 600.000
## 6        14.4       2.2   0.7666667   9.6      NA   3.850
```

```
head(select(msleep, starts_with("sleep")))
```

```
##   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
```

a. Count the number of animals which weigh under 1 kilogram and sleep more than 14 hours a day. (filter(), query())

```
light_animals1 = msleep%>%
  select("name", "genus", "order", "sleep_total", "bodywt", "brainwt")%>%
  filter(bodywt<1 & sleep_total>14)
light_animals1
```

```
##                             name        genus           order sleep_total
## 1                     Owl monkey        Aotus        Primates        17.0
## 2      Greater short-tailed shrew      Blarina    Soricomorpha        14.9
## 3                    Big brown bat    Eptesicus      Chiroptera        19.7
## 4       Western american chipmunk     Eutamias        Rodentia        14.9
## 5            Thick-tailed opposum   Lutreolina Didelphimorphia        19.4
## 6                Mongolian gerbil     Meriones        Rodentia        14.2
## 7                  Golden hamster Mesocricetus        Rodentia        14.3
## 8                Little brown bat       Myotis      Chiroptera        19.9
## 9            Round-tailed muskrat     Neofiber        Rodentia        14.6
## 10     Northern grasshopper mouse    Onychomys        Rodentia        14.5
## 11         Arctic ground squirrel Spermophilus        Rodentia        16.6
## 12 Golden-mantled ground squirrel Spermophilus        Rodentia        15.9
## 13      Eastern american chipmunk       Tamias        Rodentia        15.8
## 14                         Tenrec       Tenrec    Afrosoricida        15.6
##    bodywt brainwt
## 1   0.480 0.01550
## 2   0.019 0.00029
## 3   0.023 0.00030
## 4   0.071      NA
## 5   0.370      NA
## 6   0.053      NA
## 7   0.120 0.00100
## 8   0.010 0.00025
## 9   0.266      NA
## 10  0.028      NA
## 11  0.920 0.00570
## 12  0.205      NA
## 13  0.112      NA
## 14  0.900 0.00260
```

Number of animals: 14

```
count(light_animals1)
```

```
##    n
## 1 14
```

```
light_animals2 = msleep%>%
  select("name", "genus", "order", "sleep_total", "bodywt", "brainwt")%>%
```

```
  filter((bodywt+brainwt)<1 & sleep_total>14)
light_animals2
```

```
##                          name        genus        order sleep_total bodywt
## 1                  Owl monkey        Aotus     Primates        17.0  0.480
## 2 Greater short-tailed shrew      Blarina Soricomorpha        14.9  0.019
## 3              Big brown bat    Eptesicus   Chiroptera        19.7  0.023
## 4             Golden hamster Mesocricetus     Rodentia        14.3  0.120
## 5           Little brown bat       Myotis   Chiroptera        19.9  0.010
## 6     Arctic ground squirrel Spermophilus     Rodentia        16.6  0.920
## 7                     Tenrec       Tenrec Afrosoricida        15.6  0.900
##   brainwt
## 1 0.01550
## 2 0.00029
## 3 0.00030
## 4 0.00100
## 5 0.00025
## 6 0.00570
## 7 0.00260
```

Number of animals(brainwt): 7

```
count(light_animals2)
```

```
##   n
## 1 7
```

b. Print the name, order, sleep time and bodyweight of the animals with the 6 longest sleep times, in order of sleep time. (select(), arrange(), loc(), sort_values())

```
msleep%>%
  select(name,order,sleep_total,bodywt)%>%
  arrange(desc(sleep_total))%>%
  head()
```

```
##                      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
```

c. 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. (mutate(), assign())

```
new_msleep = msleep%>%
  mutate(
    wt_ratio = brainwt/bodywt,
    rem_ratio = sleep_rem/sleep_total,
    awake_ratio = awake/sleep_total,
```

```
    total_wt = brainwt+bodywt
  )
new_msleep
```

```
##                              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 awake_ratio   total_wt
## 1            NA         NA   0.9834711         NA
## 2  0.0322916667 0.10588235   0.4117647    0.49550
## 3            NA 0.16666667   0.6666667         NA
## 4  0.0152631579 0.15436242   0.6107383    0.01929
## 5  0.0007050000 0.17500000   5.0000000  600.42300
## 6            NA 0.15277778   0.6666667         NA
## 7            NA 0.16091954   1.7586207         NA
## 8            NA         NA   2.4285714         NA
## 9  0.0050000000 0.28712871   1.3762376   14.07000
## 10 0.0066351351         NA   7.0000000   14.89820
## 11 0.0034328358 0.11320755   3.5283019   33.61500
## 12 0.0075549451 0.08510638   1.5531915    0.73350
## 13           NA 0.07000000   1.4000000         NA
## 14 0.0152380952 0.12000000   0.9200000    0.42640
## 15 0.0166666667 0.21359223   1.3300971    0.06100
## 16 0.0066000000 0.24096386   1.8915663    1.00660
## 17 0.0280000000 0.15384615   1.6373626    0.00514
## 18 0.0030857143 0.17816092   0.3793103    3.51080
## 19 0.0041694915 0.09433962   3.5283019    2.96230
## 20 0.0037058824 0.27222222   0.3333333    1.70630
## 21 0.0018072242         NA   5.1538462 2551.60300
## 22 0.0130434783 0.19796954   0.2182741    0.02330
## 23 0.0012571977 0.20689655   7.2758621  521.65500
## 24 0.0022406417 0.12903226   6.7419355  187.41900
## 25 0.0045454545 0.34653465   1.3762376    0.77350
## 26 0.0115000000 0.10091743   1.2018349   10.11500
## 27           NA         NA   0.6107383         NA
```

```
## 28 0.0077575758 0.25600000   0.9200000    3.32560
## 29 0.0250000000 0.11224490   1.4489796    0.20500
## 30           NA 0.21052632  11.6315789         NA
## 31           NA 0.03703704   7.9074074         NA
## 32 0.0038235294 0.24193548   2.8709677   85.32500
## 33 0.0046742857 0.09523810   2.8095238    2.63727
## 34 0.0212903226 0.23750000   2.0000000   63.32000
## 35           NA 0.09473684   1.5263158         NA
## 36 0.0008584310         NA   6.2727273 6659.71200
## 37           NA 0.34020619   0.2371134         NA
## 38 0.0263235294 0.11881188   1.3762376    6.97900
## 39           NA 0.13380282   0.6901408         NA
## 40 0.0083333333 0.21678322   0.6783217    0.12100
## 41           NA         NA   0.8750000         NA
## 42 0.0181818182 0.11200000   0.9200000    0.02240
## 43 0.0250000000 0.10050251   0.2060302    0.01025
## 44           NA         NA   0.6438356         NA
## 45 0.0089285714         NA   1.1818182    1.41250
## 46           NA 0.11688312   2.1168831         NA
## 47           NA         NA   0.6551724         NA
## 48 0.0048400000 0.10714286   1.8571429    2.51210
## 49 0.0031531532 0.15789474   5.3157895   55.67500
## 50 0.0084291188 0.14432990   1.4742268   52.64000
## 51           NA         NA   0.5189873         NA
## 52 0.0015700000         NA   1.3076923  100.15700
## 53           NA         NA   0.7777778         NA
## 54 0.0071329503 0.10638298   1.5531915   25.41500
## 55 0.0043636364 0.26213592   1.3300971    0.55240
## 56           NA         NA   1.1818182         NA
## 57           NA         NA   1.0869565         NA
## 58 0.0070370370 0.13138686   0.7518248    1.63140
## 59           NA 0.11428571   5.8571429         NA
## 60           NA         NA   3.2946429         NA
## 61           NA 0.13513514   1.1621622         NA
## 62 0.0013500000 0.33701657   0.3259669   60.08100
## 63 0.0058333333 0.09259259   3.4444444    3.62100
## 64 0.0059375000 0.18461538   0.8461538    0.32190
## 65           NA         NA   1.7586207         NA
## 66 0.0269179004 0.14583333   1.5000000    0.76300
## 67 0.0160000000 0.25000000   1.8571429    0.07620
## 68 0.0079729730 0.09734513   1.1238938    0.14918
## 69 0.0245901639 0.22641509   1.2641509    0.12500
## 70 0.0061956522         NA   0.4457831    0.92570
## 71 0.0396039604 0.24637681   0.7391304    0.10500
## 72           NA 0.18867925   0.5094340         NA
## 73 0.0068750000 0.15625000   0.8750000    0.04833
## 74 0.0020869565 0.26373626   1.6373626   86.43000
## 75 0.0055555556         NA   1.7906977    4.52500
## 76           NA         NA   0.5189873         NA
## 77 0.0008144539 0.22727273   4.4545455  207.67000
```

```
## 78 0.0028888889 0.14743590   0.5384615    0.90260
## 79 0.0240384615 0.29213483   1.6966292    0.10650
## 80           NA         NA   3.6153846         NA
## 81 0.0087500000 0.20634921   2.8095238    2.01750
## 82 0.0131656805         NA   0.9200000    3.42450
## 83 0.0119148936 0.24489796   1.4489796    4.28040
```

Description: wt_ratio is Ratio of Brain Size and Body Weight rem_ratio is Ratio of Rem sleep and Sleep Time awake_ratio is Ratio of Awake Time and Sleep Time total_wt is Body Weight plus Brain Weight

d. Display the average, min and max sleep times for each order. (group_by(), summarise(), groupby(), agg())

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

## `summarise()` ungrouping output (override with `.groups` argument)

animal_sleep_stats

## # A tibble: 19 x 5
##    order           avg_sleep min_sleep max_sleep total
##    <chr>               <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
```

e. 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 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. (group_by(), mutate(), groupby(),assign())

```
missing_average_weight_ratio = msleep%>%
  group_by(order)%>%
  mutate(brainwt = ifelse(is.na(brainwt), mean(brainwt / bodywt, na.rm =
TRUE) * bodywt, brainwt))%>%
  ungroup()
missing_average_weight_ratio
```

```
## # A tibble: 83 x 11
##    name  genus vore  order conservation sleep_total sleep_rem sleep_cycle
awake
##    <chr> <chr> <chr> <chr> <chr>              <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
##  7 Nort~ Call~ carni Carn~ vu                   8.7       1.4       0.383
15.3
##  8 Vesp~ Calo~ <NA>  Rode~ <NA>                 7        NA        NA
17
##  9 Dog   Canis carni Carn~ domesticated        10.1       2.9       0.333
13.9
## 10 Roe ~ Capr~ herbi Arti~ lc                   3        NA        NA
21
## # ... with 73 more rows, and 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```
na.omit(missing_average_weight_ratio[,"brainwt"])
```

```
## # A tibble: 79 x 1
##     brainwt
##       <dbl>
##  1 0.371
##  2 0.0155
##  3 0.0189
##  4 0.00029
##  5 0.423
##  6 0.152
##  7 0.000631
##  8 0.07
##  9 0.0982
```

```
## 10 0.115
## # ... with 69 more rows

missing_average_weight = msleep%>%
  group_by(order) %>%
  mutate(brainwt = ifelse(is.na(brainwt), mean(brainwt,na.rm = TRUE),
brainwt))  %>%
  ungroup()
missing_average_weight

## # A tibble: 83 x 11
##    name  genus vore  order conservation sleep_total sleep_rem sleep_cycle
awake
##    <chr> <chr> <chr> <chr> <chr>              <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
##  7 Nort~ Call~ carni Carn~ vu                   8.7       1.4       0.383
15.3
##  8 Vesp~ Calo~ <NA>  Rode~ <NA>                 7        NA        NA
17
##  9 Dog   Canis carni Carn~ domesticated        10.1       2.9       0.333
13.9
## 10 Roe ~ Capr~ herbi Arti~ lc                   3        NA        NA
21
## # ... with 73 more rows, and 2 more variables: brainwt <dbl>, bodywt <dbl>

na.omit(missing_average_weight[,"brainwt"])

## # A tibble: 79 x 1
##    brainwt
##      <dbl>
##  1 0.0986
##  2 0.0155
##  3 0.00357
##  4 0.00029
##  5 0.423
##  6 0.0986
##  7 0.00357
##  8 0.07
##  9 0.0982
```

```
## 10 0.115
## # ... with 69 more rows

missing_sleep_rem = msleep%>%
  select("name","genus","vore","order","sleep_rem","sleep_cycle")%>%
  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()
missing_sleep_rem

## # A tibble: 83 x 6
##    name                       genus       vore  order       sleep_rem
sleep_cycle
##    <chr>                      <chr>       <chr> <chr>           <dbl>
<dbl>
##  1 Cheetah                    Acinonyx    carni Carnivora        1.87
NA
##  2 Owl monkey                 Aotus       omni  Primates         1.8
NA
##  3 Mountain beaver            Aplodontia  herbi Rodentia         2.4
NA
##  4 Greater short-tailed shr~ Blarina     omni  Soricomorp~      2.3
0.133
##  5 Cow                        Bos         herbi Artiodacty~      0.7
0.667
##  6 Three-toed sloth           Bradypus    herbi Pilosa           2.2
0.767
##  7 Northern fur seal          Callorhinus carni Carnivora        1.4
0.383
##  8 Vesper mouse               Calomys     <NA>  Rodentia         2.02
NA
##  9 Dog                        Canis       carni Carnivora        2.9
0.333
## 10 Roe deer                   Capreolus   herbi Artiodacty~      0.94
NA
## # ... with 73 more rows

missing_sleep_cycle = msleep%>%
  select("name","genus","vore","order","sleep_rem","sleep_cycle")%>%
  group_by(order)%>%

mutate(sleep_cycle=ifelse(is.na(sleep_cycle),ifelse(is.nan(mean(sleep_cycle,n
a.rm = TRUE)),0,mean(sleep_cycle,na.rm = TRUE)),sleep_cycle))%>%
  ungroup()
missing_sleep_cycle

## # A tibble: 83 x 6
##    name                       genus       vore  order       sleep_rem
sleep_cycle
##    <chr>                      <chr>       <chr> <chr>           <dbl>
```

```
<dbl>
##  1 Cheetah                    Acinonyx    carni Carnivora        NA
0.371
##  2 Owl monkey                 Aotus       omni  Primates          1.8
0.977
##  3 Mountain beaver            Aplodontia  herbi Rodentia          2.4
0.181
##  4 Greater short-tailed shr~ Blarina     omni  Soricomorp~       2.3
0.133
##  5 Cow                        Bos         herbi Artiodacty~       0.7
0.667
##  6 Three-toed sloth           Bradypus    herbi Pilosa            2.2
0.767
##  7 Northern fur seal          Callorhinus carni Carnivora         1.4
0.383
##  8 Vesper mouse               Calomys     <NA>  Rodentia         NA
0.181
##  9 Dog                        Canis       carni Carnivora         2.9
0.333
## 10 Roe deer                   Capreolus   herbi Artiodacty~      NA
0.583
## # ... with 73 more rows
```

The best way to replace missing values is by taking the mean of the brain weight since replacing the mean will not affect the data when we perform statistical operations on it. Therefore the missing values of sleep_rem and sleep_cycle has been replaced by the corresponding mean of their orders.

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.html#case-study). Grab the dataset from the tidyr package (tidyr::who), and tidy it as shown in the case study before answering the following questions. Note: if you are using pandas you can perform these same operations, just replace the pivot_longer() function with melt() and the pivot_wider() function with pivot(). However, you may prefer to use R for this question, as the dataset is from an R package.

```
library(tidyr)
who = tidyr::who
```

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?

This line is used to replace all the strings which contain newrel as column name to new_rel. This is done because when we try to separate the data using separate(key, c(“new”, “type”, “sexage”), sep = "_"), if we retain the column name as newrel itself, both the details of whether is a new case of TB (new) and the type of TB (rel) will both be present in the new column itself. If we change newrel to new_rel there will be consistency in all the column

names which not only makes the separate function execute as expected other functions will also work as expected.

b. How many entries are removed from the dataset when you set values_drop_na to true in the pivot_longer command (in this dataset)?

To check the number of entries that are removed from the dataset when we removed the NA values in the gather command, first let check how much entries were there before removing the NA's.

```
who1 = who %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm = FALSE)

who1
```

```
## # 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    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
```

It shows that arround 405,440 entries are there in total having both numeric data for some entities and NA data for the rest. The following lines of codes shows the number of data left after removing NA from the data set.

```
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
```

It shows that we had entries more than 300,000 that includes NA, and afteer removing that we are just left with 76 thousand entries. If we were asked to do the same task without using na.rm = TRUE, I will use na.omit() to remove the rows having NA as input. It will give the same result arround 76 thousand rows.

```
who1 = who %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases")%>%
  na.omit(who1)

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
```

I think that na.omit is a good way to handle these missing values from the dataset because na.rm is mostly used in mathematical operations whereas, na.omit is used to omit all the rows that contains missing value.

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?

Explicit missing values means there is a specific representation that will indicate the row has missing value (row=NA) Implicit missing values means the value is not present(row=" ") or might be represented differently (row = 0)

```
whonew = who
implictCount=whonew %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm =
TRUE)%>%
  filter(cases == 0) %>%
  nrow()

whonew
```

```
## # A tibble: 7,240 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
##  7 Afghan~ AF    AFG    1986          NA           NA           NA
NA
##  8 Afghan~ AF    AFG    1987          NA           NA           NA
NA
##  9 Afghan~ AF    AFG    1988          NA           NA           NA
NA
## 10 Afghan~ AF    AFG    1989          NA           NA           NA
NA
## # ... with 7,230 more rows, and 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>
```

Total Implicit Missing Value = 11080

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?

```
who_tidied_data=who %>%
  gather(key, value, new_sp_m014:newrel_f65, na.rm = 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)

who_tidied_data
```

```
## # A tibble: 76,046 x 6
##    country      year var   sex   age   value
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1998 sp    m     014      30
##  3 Afghanistan  1999 sp    m     014       8
##  4 Afghanistan  2000 sp    m     014      52
##  5 Afghanistan  2001 sp    m     014     129
##  6 Afghanistan  2002 sp    m     014      90
##  7 Afghanistan  2003 sp    m     014     127
##  8 Afghanistan  2004 sp    m     014     139
##  9 Afghanistan  2005 sp    m     014     151
## 10 Afghanistan  2006 sp    m     014     193
## # ... with 76,036 more rows
```

```
sapply(who_tidied_data, class)
```

```
##     country        year         var         sex         age       value
## "character"   "integer" "character" "character" "character"   "integer"
```

The column age can be changed from character to integer since it contains only integer values.

e. Generate an informative visualization, which shows something about the data. Give a brief description of what it shows, and why you thought it would be interesting to investigate.

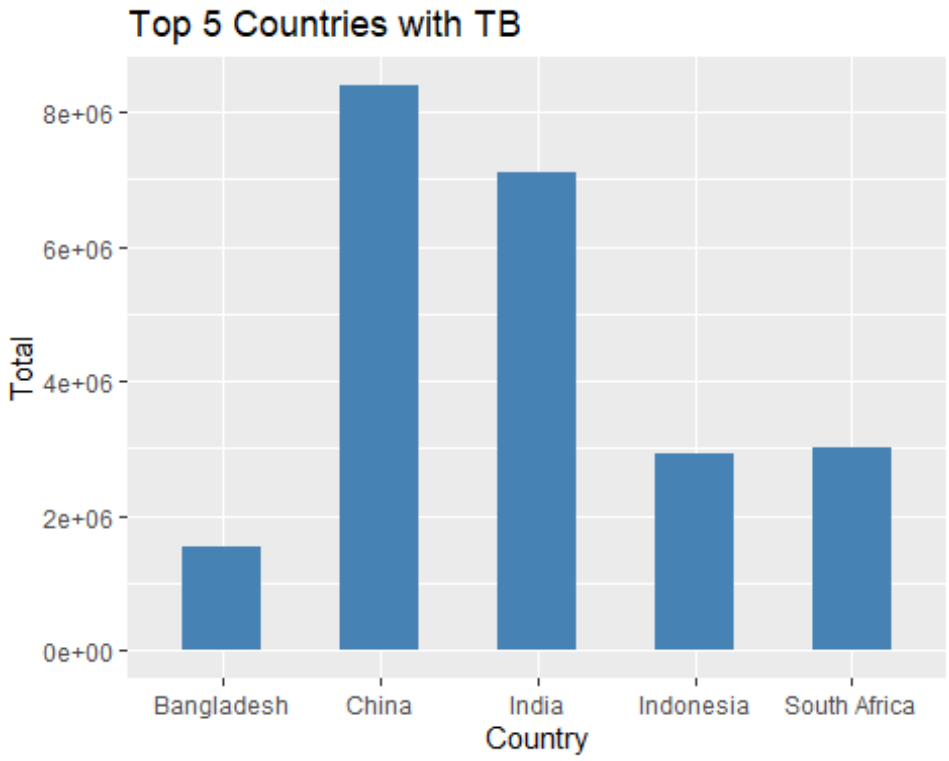
```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     msleep
```

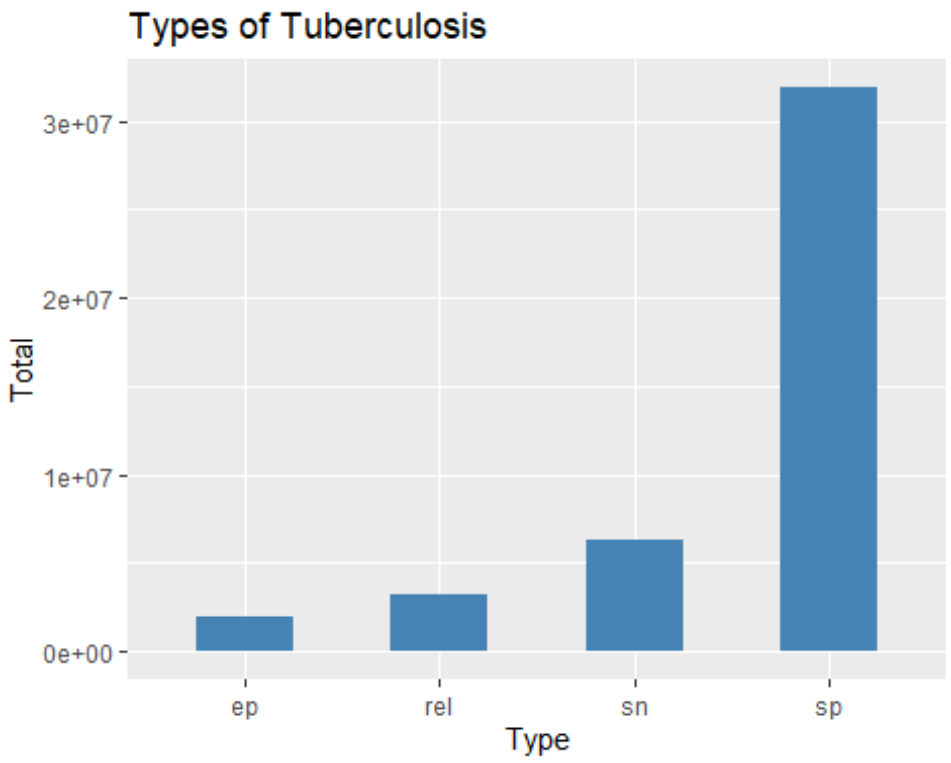
```
library(ggpubr)

who_v_is= who %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm =
TRUE)%>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel"))%>%
  separate(key, c("new", "type", "sexage"), sep = "_")%>%
  select(-new, -iso2, -iso3)%>%
  separate(sexage, c("sex", "age"), sep = 1)
```

```
country_cases=who_v_is%>%
group_by(country)%>%
tally(cases)%>%
top_n(5)

## Selecting by n

ggplot(data=country_cases, aes(x=country,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Top 5 Countries with TB")+
  xlab("Country")+ylab("Total")
```



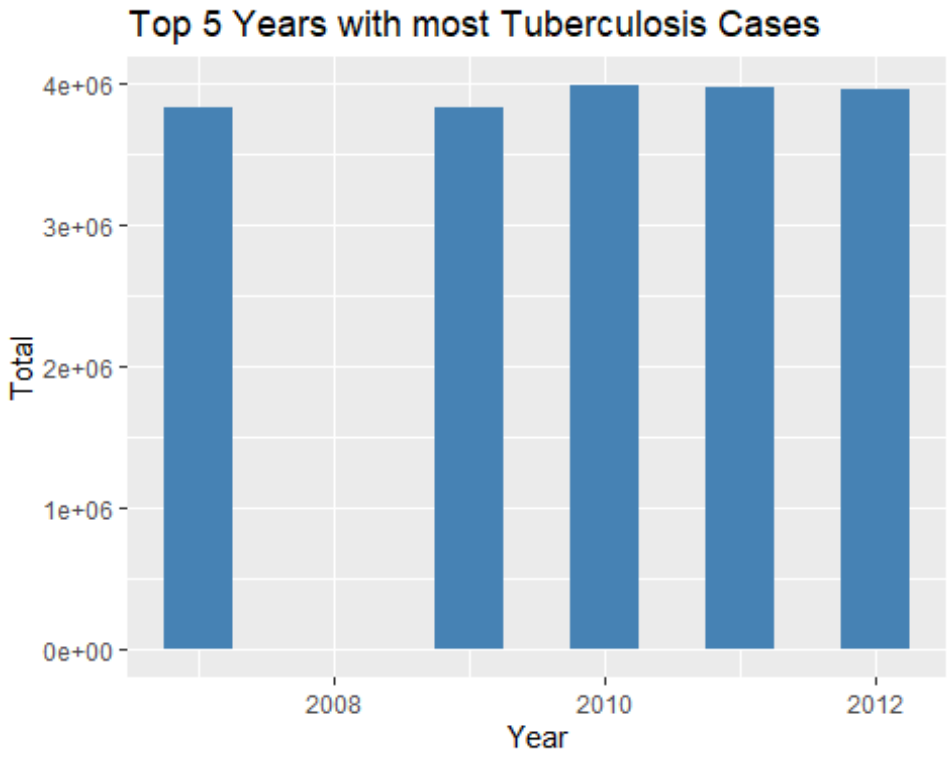
The Above Graph Shows the countries with most TB cases. From the plot we can understand Asia and South Africa are the most affected.

```
type_cases=who_v_is%>%
  group_by(type)%>%
  tally(cases)

ggplot(data=type_cases, aes(x=type,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Types of Tuberculosis")+
  xlab("Type")+ylab("Total")
```



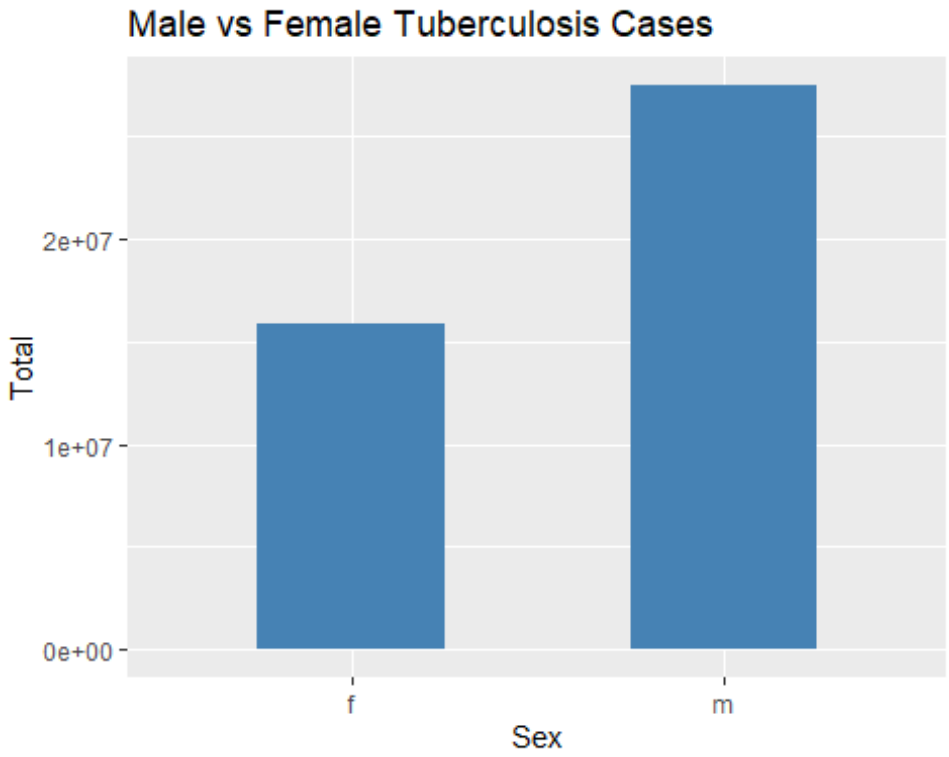
The Above Graph Shows the different types of TB. From the plot we can find that sp is the most common type of TB.

```
year_cases=who_v_is%>%
  group_by(year)%>%
  tally(cases)%>%
  top_n(5)

## Selecting by n

ggplot(data=year_cases, aes(x=year,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Top 5 Years with most Tuberculosis Cases")+
  xlab("Year")+ylab("Total")
```



The Above Graph Shows the years with most TB cases. From the plot we can see that from 2007-2012 has seen the most TB cases and the number of TB cases in these years are almost same.

```
sex_cases=who_v_is%>%
  group_by(sex)%>%
  tally(cases)

ggplot(data=sex_cases, aes(x=sex,y=n))+
  geom_bar(stat="identity",width=0.5,fill="steelblue")+
  ggtitle("Male vs Female Tuberculosis Cases")+
  xlab("Sex")+ylab("Total")
```



The Above Graph Shows the number of TB cases for men and women. From the plot we know that men are more prone to TB when compared to women.

f. Suppose you have the following dataset called siteDemo:

You know that the U30.F column is the number of female users under 30 on the site, O30.M denotes the number of male users 30 or older on the site, etc. Construct this table, and show the code you would use to tidy this dataset (using gather()/pivot_longer() and separate()/pivot_wider() or melt() and pivot()) such that the columns are organized as: Site, AgeGroup, Gender and Count.

```
siteDemo = data.frame(Site = c("facebook",
"myspace","snapchat","twitter","tiktok"),
                      U30.F = c(30,1,6,18,44),
                      U30_M = c(35,2,5,23,60),
                      O30.F = c(66,3,3,12,2),
                      O30.M = c(58,6,2,28,7))

siteDemo
```

```
##       Site U30.F U30_M O30.F O30.M
## 1 facebook    30    35    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
```

```r
library(tidyr)

new_siteDemo = gather(siteDemo, key = "Age_Group", value = "Number_Of_Users",
-Site)

new_siteDemo
```

```
##        Site Age_Group Number_Of_Users
## 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              35
## 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
```

```r
final_siteDemo = separate(new_siteDemo, col = Age_Group, into = c("Age
Group", "Gender"), sep = "([\\.\\_])")

final_siteDemo
```

```
##        Site Age Group Gender Number_Of_Users
## 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              35
## 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
```