

# Cpts575 Hw3

Mengxiao

## Part 1

```
library(dplyr)
#msleep = read.csv('https://scads.eecs.wsu.edu/wp-content/uploads/2017/10/msleep_ggplot2.csv')
msleep = read.csv('msleep_ggplot2.csv')
#heads = head(select(msleep, contains('sleep')))
msleep %>%
  select(contains('sleep'))%>%
  head()
```

```
##   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_numbers = count(filter(msleep, bodywt<50, sleep_total>16))
msleep %>%
  filter(bodywt<50, sleep_total>16) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     7
```

b.

```
msleep %>%
  select(name, sleep_total, bodywt) %>%
  arrange(desc=sleep_total) %>%
  top_n(5)
```

```
## Selecting by bodywt
```

```
##           name sleep_total  bodywt
## 1      Giraffe         1.9 899.995
```

```
## 2      Pilot whale      2.7  800.000
## 3 African elephant      3.3 6654.000
## 4   Asian elephant      3.9 2547.000
## 5           Cow        4.0  600.000
```

c.

```
msleep = msleep %>%
  mutate(wt_ratio=brainwt/bodywt,
         rem_ratio=sleep_rem/sleep_total)
```

d.

```
order_sum = data.frame(
  msleep %>%
    group_by(order)%>%
    summarise(avg=mean(sleep_total),
              max=max(sleep_total),
              min=min(sleep_total))
)
order_sum
```

```
##      order      avg  max  min
## 1  Afrosoricida 15.600000 15.6 15.6
## 2  Artiodactyla  4.516667  9.1  1.9
## 3    Carnivora 10.116667 15.8  3.5
## 4    Cetacea  4.500000  5.6  2.7
## 5  Chiroptera 19.800000 19.9 19.7
## 6    Cingulata 17.750000 18.1 17.4
## 7 Didelphimorphia 18.700000 19.4 18.0
## 8  Diprotodontia 12.400000 13.7 11.1
## 9  Erinaceomorpha 10.200000 10.3 10.1
## 10   Hyracoidea  5.666667  6.3  5.3
## 11   Lagomorpha  8.400000  8.4  8.4
## 12   Monotremata  8.600000  8.6  8.6
## 13 Perissodactyla  3.466667  4.4  2.9
## 14     Pilosa 14.400000 14.4 14.4
## 15    Primates 10.500000 17.0  8.0
## 16  Proboscidea  3.600000  3.9  3.3
## 17    Rodentia 12.468182 16.6  7.0
## 18   Scandentia  8.900000  8.9  8.9
## 19  Soricomorpha 11.100000 14.9  8.4
```

e.

```
#mutate(msleep, avg_ratio=order_sum[order_sum$order == order,]$avg)
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  wt_ratio
## 1      12.1      NA      NA  11.9      NA  50.000      NA
## 2      17.0      1.8      NA   7.0 0.01550   0.480 0.03229167
## 3      14.4      2.4      NA   9.6      NA   1.350      NA
## 4      14.9      2.3  0.1333333   9.1 0.00029   0.019 0.01526316
## 5       4.0      0.7  0.6666667  20.0 0.42300 600.000 0.00070500
## 6      14.4      2.2  0.7666667   9.6      NA   3.850      NA
##  rem_ratio
## 1      NA
## 2 0.1058824
## 3 0.1666667
## 4 0.1543624
## 5 0.1750000
## 6 0.1527778
```

```
msleep_copy2 = data.frame(
  msleep %>%
    group_by(order) %>%
    mutate(brainwt)
)
```

## Part 2

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

```
## # A tibble: 6 x 60
##   country iso2 iso3  year new_sp_m014 new_sp_m1524 new_sp_m2534
##   <chr>   <chr> <chr> <int>      <int>      <int>      <int>
## 1 Afghan~ AF    AFG   1980         NA         NA         NA
## 2 Afghan~ AF    AFG   1981         NA         NA         NA
## 3 Afghan~ AF    AFG   1982         NA         NA         NA
## 4 Afghan~ AF    AFG   1983         NA         NA         NA
## 5 Afghan~ AF    AFG   1984         NA         NA         NA
## 6 Afghan~ AF    AFG   1985         NA         NA         NA
## # ... with 53 more variables: 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>
```

```
mywho = 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)
head(mywho)
```

```
## # A tibble: 6 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
```

a.

This line tries to replace all the strings name “newrel” to “new\_rel”. So it tidy the key name of the data. If I skip this line, we will have one more key called “newrel” and it’s data cannot be select when we use the key “new\_rel”.

b.

```
removed = count(gather(mywho)) - count(gather(mywho, na.rm=1))
removed
```

```
##   n
## 1 0
```

c.

1. Explicit missing value means the data is ‘NA’, ‘NAN’ or some other Null value.
2. Implicit missing value means the data is just doesn’t appear on the table.

d.

```
mywho2 = mywho %>%  
  select(country, year, var, sex, age)  
  filter(mywho2, var != 'sp')
```

```
## # A tibble: 31,226 x 5  
##   country      year var    sex  age  
##   <chr>      <int> <chr> <chr> <chr>  
## 1 Afghanistan  2007 sn    m    014  
## 2 Afghanistan  2011 sn    m    014  
## 3 Afghanistan  2012 sn    m    014  
## 4 Albania      2006 sn    m    014  
## 5 Albania      2007 sn    m    014  
## 6 Albania      2008 sn    m    014  
## 7 Albania      2009 sn    m    014  
## 8 Albania      2010 sn    m    014  
## 9 Albania      2011 sn    m    014  
## 10 Albania     2012 sn    m    014  
## # ... with 31,216 more rows
```