

# 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(msleep$sleep_total)) %>%
  head(5)
```

```
##           name sleep_total bodywt
## 1 Little brown bat      19.9  0.010
## 2   Big brown bat      19.7  0.023
## 3 Thick-tailed opossum    19.4  0.370
## 4   Giant armadillo     18.1 60.000
## 5 North American Opossum  18.0  1.700
```

c.

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

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.

```
msleep1 <- msleep %>%
  group_by(order) %>%
  mutate(brainwt = ifelse(is.na(brainwt), mean(wt_ratio, na.rm = 1)*bodywt, brainwt))
head(select(msleep1, name, order, brainwt), n = 8)
```

```
## # A tibble: 8 x 3
## # Groups:   order [6]
##   name                order      brainwt
##   <fct>              <fct>      <dbl>
## 1 Cheetah            Carnivora    0.371
## 2 Owl monkey        Primates     0.0155
## 3 Mountain beaver   Rodentia     0.0189
## 4 Greater short-tailed shrew Soricomorpha 0.00029
## 5 Cow               Artiodactyla 0.423
## 6 Three-toed sloth   Pilosa      NaN
## 7 Northern fur seal  Carnivora    0.152
## 8 Vesper mouse       Rodentia     0.000631
```

```
msleep2 <- msleep %>%
  group_by(order) %>%
  mutate(brainwt = ifelse(is.na(brainwt), mean(brainwt, na.rm = 1), brainwt))
head(select(msleep2, name, order, brainwt), n = 8)
```

```
## # A tibble: 8 x 3
## # Groups:   order [6]
##   name                order      brainwt
##   <fct>              <fct>      <dbl>
## 1 Cheetah            Carnivora    0.0986
## 2 Owl monkey        Primates     0.0155
## 3 Mountain beaver   Rodentia     0.00357
## 4 Greater short-tailed shrew Soricomorpha 0.00029
## 5 Cow               Artiodactyla 0.423
## 6 Three-toed sloth   Pilosa      NaN
## 7 Northern fur seal  Carnivora    0.0986
## 8 Vesper mouse       Rodentia     0.00357
```

In my opinion, the first way is better. Since it has considered the difference of animal's body weight. I believe that animals in the same order would have different brain size with different body size.

## 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.

```
unremoved = who %>%  
  gather(new_sp_m014:newrel_f65, key="key", value="cases")  
Delete_Number = nrow(unremoved) - nrow(mywho)  
Delete_Number
```

```
## [1] 329394
```

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. From the data we can find that the data after tidy is begin from 1997, but it was begin from 1980. The data between 1980 and 1997 is missing.

d.

```
mywho2 = mywho %>%  
  select(country, year, var, sex, age, value)
```

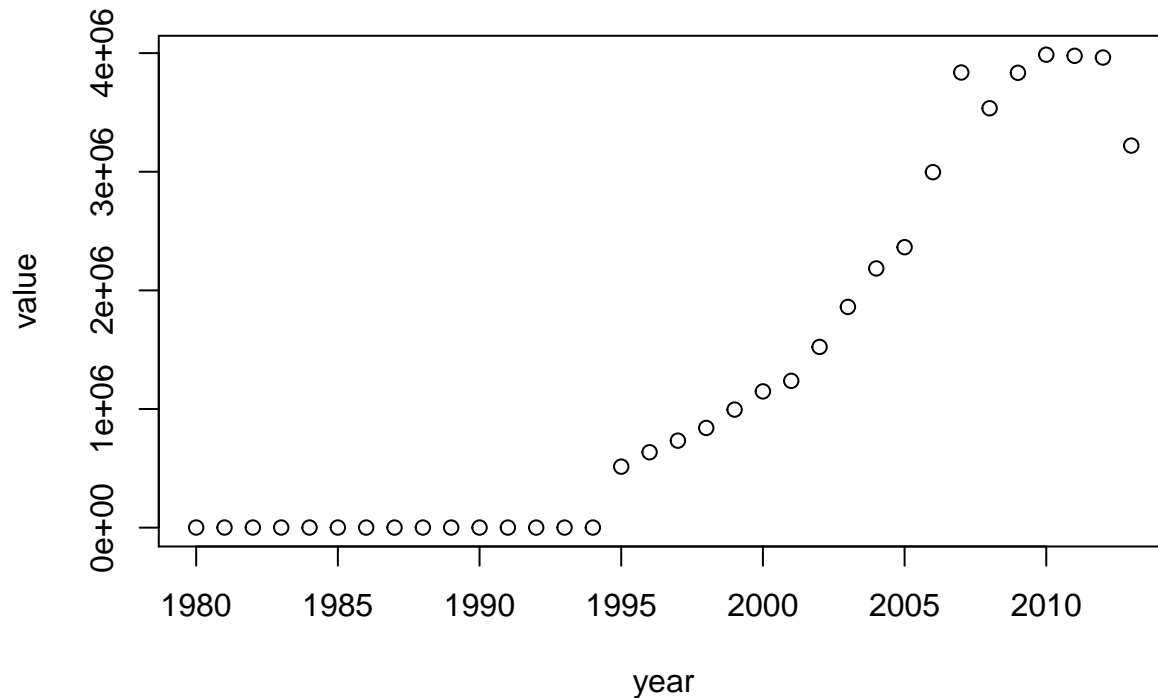
I think the ‘age’ should change it’s type to ‘int’, since ages are all int numbers.

e.

In my opinion, gather is the operation that would help people focus on the characteristics of one or some typical object. For example, when we have the data of a lot of cars, when we want to compare the details of two cars, then we can gather two cars. Also, spread do the similar things as gather, but it would catch the different objects’ one or some typical characteristics, then compare the characteristics of different object to know the difference between different unities.

f.

```
mywho3 <- mywho %>%  
  group_by(year) %>%  
  summarise(value=sum(value))  
plot(mywho3)
```



```
c(mywho3['year'])
```

```
## $year
## [1] 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993
## [15] 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
## [29] 2008 2009 2010 2011 2012 2013
```

From this scatter I can know the amount of cases is increasing from 1995, but is decrease in the 2013. I want to know whether the cases decrease with the development of the modern medicine.

g.

```
siteDemo <- data.frame(Site = c("facebook", "myspace", "snapchat", "twitter"),
  U30.F = c(32, 1, 6, 17),
  U30.M = c(31, 5, 4, 23),
  O30.F = c(60, 3, 3, 12),
  O30.M = c(58, 6, 2, 17))
siteDemo_new <- siteDemo %>%
  gather('U30.F', 'U30.M', 'O30.F', 'O30.M', key="AGG", value = "Count")
siteDemo_new
```

```
##      Site  AGG Count
## 1 facebook U30.F   32
## 2  myspace U30.F    1
## 3 snapchat U30.F    6
## 4  twitter U30.F   17
## 5 facebook U30.M   31
## 6  myspace U30.M    5
```

```
## 7  snapchat U30.M      4
## 8  twitter  U30.M     23
## 9  facebook 030.F     60
##10  myspace  030.F      3
##11  snapchat 030.F      3
##12  twitter  030.F     12
##13  facebook 030.M     58
##14  myspace  030.M      6
##15  snapchat 030.M      2
##16  twitter  030.M     17
```

```
siteDemo_new <- siteDemo_new %>%
  separate('AGG', into = c("AgeGroup", "Gender"))
siteDemo_new
```

##	Site	AgeGroup	Gender	Count
## 1	facebook	U30	F	32
## 2	myspace	U30	F	1
## 3	snapchat	U30	F	6
## 4	twitter	U30	F	17
## 5	facebook	U30	M	31
## 6	myspace	U30	M	5
## 7	snapchat	U30	M	4
## 8	twitter	U30	M	23
## 9	facebook	030	F	60
##10	myspace	030	F	3
##11	snapchat	030	F	3
##12	twitter	030	F	12
##13	facebook	030	M	58
##14	myspace	030	M	6
##15	snapchat	030	M	2
##16	twitter	030	M	17