Problem Set 1: Manipulation of mammalian sleep data

Your name and student ID

Today's date

Instructions

- Solutions will be released by Friday, July 5th.
- This semester, problem sets are for practice only and will not be turned in for marks.

Helpful hints:

- Every function you need to use was taught during lecture! So you may need to revisit the lecture code to help you along by opening the relevant files on Datahub. Alternatively, you may wish to view the code in the condensed PDFs posted on the course website. Good luck!

- Knit your file early and often to minimize knitting errors! If you copy and paste code for the slides, you are bound to get an error that is hard to diagnose. Typing out the code is the way to smooth knitting! We recommend knitting your file each time after you write a few sentences/add a new code chunk, so you can detect the source of knitting errors more easily. This will save you and the GSIs from frustration!

- It is good practice to not allow your code to run off the page. To avoid this, have a look at your knitted PDF and ensure all the code fits in the file. If it doesn't look right, go back to your .Rmd file and add spaces (new lines) using the return or enter key so that the code runs onto the next line.

Begin by knitting this document by pushing the "Knit" button above. As you fill in code and text in the document, you can re-knit (push the button again) and see how the document changes. It is important to re-knit often, because if there is any error in your code, the file will not generate a PDF, so our advice is to knit early and often!

Using `dplyr` to investigate sleep times in mammals

The data file `sleep.csv` contains the sleep times and weights for a set of mammals. Hit the green arrow icon in the line below to execute the lines of code in the code chunk, or execute them line by line by placing your cursor on the first line and hitting cmd + enter on Mac or ctrl + enter on PC.

```
library(dplyr)
```

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

```
## The following object is masked from 'package:testthat':
##
##     matches
```

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

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

```
library(readr)
```

```
##
## Attaching package: 'readr'
```

```
## The following objects are masked from 'package:testthat':
##
##     edition_get, local_edition
```

```
sleep <- read_csv("data/sleep.csv")
```

```
## Rows: 83 Columns: 11
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): name, genus, vore, order, conservation
## dbl (6): sleep_total, sleep_rem, sleep_cycle, awake, brainwt, bodywt
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

- The `library` command loads the library `dplyr` into memory.
- The `readr` library contains functions to read in the dataset.
- The `dplyr` library contains functions we will use to manipulate data.

Notice that an object called `sleep` appears in the Environment tab under "Data".

1. [2 points] Use four useful functions discussed in lecture to examine the sleep dataset:

```
# Text inside a code chunk that begins with "#" is called a comment.
# We sometimes use comments to explain code to you in plain English.
# Write your four functions below these comments, replacing the placeholder
# text "<<<<YOUR CODE HERE>>>>". Remember, code does *not* begin with a "#"

"<<<<YOUR CODE HERE>>>>"
```

```
## [1] "<<<<YOUR CODE HERE>>>>"
```

```
"<<<<YOUR CODE HERE>>>>"
```

```
## [1] "<<<<YOUR CODE HERE>>>>"
```

```
"<<<<YOUR CODE HERE>>>>"
```

```
## [1] "<<<<YOUR CODE HERE>>>>"
```

```
"<<<<YOUR CODE HERE>>>>"
```

```
## [1] "<<<<YOUR CODE HERE>>>>"
```

```
# Then, assign p1 to a vector of your function names, in alphabetical order.
# For example, assigning p0 to a vector of fruits looks like this:
# p0 <- c("apple", "banana", "orange")

p1 <- c("dim", "head", "names", "str")
p1
```

```
## [1] "dim"   "head"  "names" "str"
```

```
. = ottr::check("tests/p1.R")
```

```
##
## All tests passed!
```

Description of the variables found in the sleep dataset:

Column name	Description
name	common name
genus	taxonomic rank
vore	carnivore, omnivore or herbivore?
order	taxonomic rank
conservation	the conservation status of the mammal
sleep_total	total amount of sleep, in hours
sleep_rem	Rapid eye movement (REM) sleep, in hours
sleep_cycle	length of sleep cycle, in hours
awake	amount of time spent awake, in hours
brainwt	brain weight in kilograms
bodywt	body weight in kilograms

2. [2 points] Write code to select a set of columns. Specifically select the `awake`, `brainwt`, and `bodywt` columns. Assign this smaller dataset to a dataframe called `sleep_small`.

```r
sleep_small <- select(sleep, awake, brainwt, bodywt)
sleep_small
```

```
## # A tibble: 83 x 3
##     awake  brainwt  bodywt
##     <dbl>    <dbl>   <dbl>
##  1  11.9 NA          50
##  2   7    0.0155      0.48
##  3   9.6 NA           1.35
##  4   9.1  0.00029     0.019
##  5  20    0.423     600
##  6   9.6 NA           3.85
##  7  15.3 NA          20.5
##  8  17   NA           0.045
##  9  13.9  0.07       14
## 10  21    0.0982     14.8
## # i 73 more rows
```

```r
. = ottr::check("tests/p2.R")
```

```
##
## All tests passed!
```

3. [1 point] To select a range of columns by name, use the ':' (colon) operator. Redo the selection for question 1, but use the colon operator. Assign this to `sleep_small_colon`. Note that this returns the same data frame as the previous problem, but is not recommended in practice because it depends on the ordering of the columns and isn't explicit in the columns that are selected, whereas selecting columns by name offers much higher readability for someone else looking at your code later on.

```r
sleep_small_colon <- sleep %>% select(awake:bodywt)
sleep_small_colon
```

```
## # A tibble: 83 x 3
##    awake  brainwt  bodywt
##    <dbl>    <dbl>   <dbl>
##  1  11.9 NA           50
##  2   7    0.0155     0.48
##  3   9.6 NA          1.35
##  4   9.1  0.00029   0.019
##  5  20     0.423     600
##  6   9.6 NA          3.85
##  7  15.3 NA          20.5
##  8  17    NA         0.045
##  9  13.9  0.07        14
## 10  21     0.0982    14.8
## # i 73 more rows
```

```r
. = ottr::check("tests/p3.R")
```

```
##
## All tests passed!
```

4. [1 point] Select all of the columns except for the `vore` variable from the original `sleep` dataset. Assign this to `sleep_no_vore`.

```
sleep_no_vore <- sleep %>% select(-vore)
sleep_no_vore
```

```
## # A tibble: 83 x 10
##     name          genus order conservation sleep_total sleep_rem sleep_cycle awake
##     <chr>         <chr> <chr> <chr>               <dbl>     <dbl>       <dbl> <dbl>
## 1 Cheetah        Acin~ Carn~ lc                   12.1      NA          NA    11.9
## 2 Owl monkey     Aotus Prim~ <NA>                 17         1.8        NA     7
## 3 Mountain be~   Aplo~ Rode~ nt                   14.4       2.4        NA     9.6
## 4 Greater sho~   Blar~ Sori~ lc                   14.9       2.3         0.133 9.1
## 5 Cow            Bos   Arti~ domesticated          4         0.7         0.667 20
## 6 Three-toed ~   Brad~ Pilo~ <NA>                 14.4       2.2         0.767 9.6
## 7 Northern fu~   Call~ Carn~ vu                    8.7       1.4         0.383 15.3
## 8 Vesper mouse   Calo~ Rode~ <NA>                  7        NA          NA    17
## 9 Dog            Canis Carn~ domesticated         10.1       2.9         0.333 13.9
## 10 Roe deer       Capr~ Arti~ lc                   3        NA          NA    21
## # i 73 more rows
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```
# YOUR CODE HERE
```

```
. = ottr::check("tests/p4.R")
```

```
##
## All tests passed!
```

5. [1 point] Run the following chunk of code.

```r
select(sleep, starts_with("sl"))
```

```
## # A tibble: 83 x 3
##    sleep_total sleep_rem sleep_cycle
##          <dbl>     <dbl>       <dbl>
##  1        12.1      NA          NA
##  2        17         1.8        NA
##  3        14.4       2.4        NA
##  4        14.9       2.3         0.133
##  5         4         0.7         0.667
##  6        14.4       2.2         0.767
##  7         8.7       1.4         0.383
##  8         7        NA          NA
##  9        10.1       2.9         0.333
## 10         3        NA          NA
## # i 73 more rows
```

What does it return? Copy your choice and assign it to p5.

```r
# p5 <- "returns all columns that start with sl"
```

```r
p5 <- "returns all columns that start with sl"
p5
```

```
## [1] "returns all columns that start with sl"
```

```r
# YOUR CODE HERE
```

```r
. = ottr::check("tests/p5.R")
```

```
##
## All tests passed!
```

6. [1 point] Rewrite the chunk of code that selects columns starting with "sl" (in question 5) using the pipe operator. Assign this to `sleep_sl`.

```r
sleep_sl <- sleep %>% select(starts_with("sl"))
sleep_sl
```

```
## # A tibble: 83 x 3
##    sleep_total sleep_rem sleep_cycle
##          <dbl>     <dbl>       <dbl>
##  1        12.1        NA          NA
##  2        17         1.8          NA
##  3        14.4        2.4          NA
##  4        14.9        2.3       0.133
##  5         4         0.7       0.667
##  6        14.4        2.2       0.767
##  7         8.7        1.4       0.383
##  8         7          NA          NA
##  9        10.1        2.9       0.333
## 10         3          NA          NA
## # i 73 more rows
```

```r
. = ottr::check("tests/p6.R")
```

```
##
## All tests passed!
```

7. **[1 point]** Filter the original `sleep` dataset to include rows with mammals that sleep a total of more than 16 hours. Assign this to `sleep_over16`.

```
sleep_over16 <- sleep %>% filter(sleep_total > 16)
sleep_over16
```

```
## # A tibble: 8 x 11
##    name     genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##    <chr>    <chr> <chr> <chr> <chr>               <dbl>     <dbl>       <dbl> <dbl>
## 1 Owl mo~ Aotus omni  Prim~ <NA>                 17        1.8      NA         7
## 2 Long-n~ Dasy~ carni Cing~ lc                   17.4      3.1       0.383     6.6
## 3 North ~ Dide~ omni  Dide~ lc                   18        4.9       0.333     6
## 4 Big br~ Epte~ inse~ Chir~ lc                   19.7      3.9       0.117     4.3
## 5 Thick-~ Lutr~ carni Dide~ lc                   19.4      6.6      NA         4.6
## 6 Little~ Myot~ inse~ Chir~ <NA>                 19.9      2         0.2       4.1
## 7 Giant ~ Prio~ inse~ Cing~ en                   18.1      6.1      NA         5.9
## 8 Arctic~ Sper~ herbi Rode~ lc                   16.6     NA        NA         7.4
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```
. = ottr::check("tests/p7.R")
```

```
##
## All tests passed!
```

8. [2 points] Filter the rows to include mammals that sleep a total of more than 16 hours and have a body weight of greater than 1 kilogram. Assign this to `sleep_mammals`.

```r
sleep_mammals <- sleep %>% filter(sleep_total > 16 & bodywt > 1)
sleep_mammals
```

```
## # A tibble: 3 x 11
##   name      genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <chr>     <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Long-n~ Dasy~ carni Cing~ lc                  17.4       3.1       0.383   6.6
## 2 North ~ Dide~ omni  Dide~ lc                  18         4.9       0.333   6
## 3 Giant ~ Prio~ inse~ Cing~ en                  18.1       6.1       NA      5.9
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```r
. = ottr::check("tests/p8.R")
```

```
##
## All tests passed!
```

9. [1 point] Suppose you are specifically interested in the sleep times of horses and giraffes. Use the original `sleep` dataset and assign `sleep_hg` to a dataframe that only includes horses and giraffes.

```r
sleep_hg <- sleep %>% filter(name %in% c("Horse", "Giraffe"))
sleep_hg
```

```
## # A tibble: 2 x 11
##   name    genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <chr>   <chr> <chr> <chr> <chr>               <dbl>     <dbl>       <dbl> <dbl>
## 1 Horse   Equus herbi Peri~ domesticated          2.9       0.6           1  21.1
## 2 Giraffe Gira~ herbi Arti~ cd                    1.9       0.4          NA  22.1
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```r
. = ottr::check("tests/p9.R")
```

```
##
## All tests passed!
```

10. [1 point] Order the original dataset from shortest sleep time to longest sleep time. Assign this to `sleep_time`.

```
sleep_time <- sleep %>% arrange(sleep_total)
sleep_time
```

```
## # 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 Giraf~  Gira~ herbi Arti~ cd                    1.9       0.4          NA  22.1
##  2 Pilot~  Glob~ carni Ceta~ cd                    2.7       0.1          NA  21.4
##  3 Horse   Equus herbi Peri~ domesticated          2.9       0.6           1  21.1
##  4 Roe d~  Capr~ herbi Arti~ lc                    3        NA            NA  21
##  5 Donkey  Equus herbi Peri~ domesticated          3.1       0.4          NA  20.9
##  6 Afric~  Loxo~ herbi Prob~ vu                    3.3      NA            NA  20.7
##  7 Caspi~  Phoca carni Carn~ vu                    3.5       0.4          NA  20.5
##  8 Sheep   Ovis  herbi Arti~ domesticated          3.8       0.6          NA  20.2
##  9 Asian~  Elep~ herbi Prob~ en                    3.9      NA            NA  20.1
## 10 Cow     Bos   herbi Arti~ domesticated          4         0.7       0.667 20
## # i 73 more rows
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```
. = ottr::check("tests/p10.R")
```

```
##
## All tests passed!
```

11. [1 point] Now order the original dataset from longest to shortest sleep time. Assign this to `sleep_rev`.

```
sleep_rev <- sleep %>% arrange(-sleep_total)
sleep_rev
```

```
## # 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 Littl~ Myot~ inse~ Chir~ <NA>                19.9       2          0.2     4.1
##  2 Big b~ Epte~ inse~ Chir~ lc                  19.7       3.9        0.117   4.3
##  3 Thick~ Lutr~ carni Dide~ lc                  19.4       6.6       NA       4.6
##  4 Giant~ Prio~ inse~ Cing~ en                  18.1       6.1       NA       5.9
##  5 North~ Dide~ omni  Dide~ lc                  18         4.9        0.333   6
##  6 Long-~ Dasy~ carni Cing~ lc                  17.4       3.1        0.383   6.6
##  7 Owl m~ Aotus omni  Prim~ <NA>                17         1.8       NA       7
##  8 Arcti~ Sper~ herbi Rode~ lc                  16.6      NA         NA       7.4
##  9 Golde~ Sper~ herbi Rode~ lc                  15.9       3         NA       8.1
## 10 Tiger  Pant~ carni Carn~ en                  15.8      NA         NA       8.2
## # i 73 more rows
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```
. = ottr::check("tests/p11.R")
```

```
##
## All tests passed!
```

12. [2 points] Suppose you are interested in the order of sleep time (longest to shortest), but according to whether the animal is a carnivore, herbivore, or omnivore. Write the code that orders sleep time according to the animal's type of `-vore`. Call this `sleep_time_rev`.

```
sleep_time_rev <-  sleep %>% arrange(vore, -sleep_total)
sleep_time_rev
```

```
## # 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 Thick~ Lutr~ carni Dide~ lc                   19.4       6.6        NA     4.6
##  2 Long-~ Dasy~ carni Cing~ lc                   17.4       3.1         0.383  6.6
##  3 Tiger  Pant~ carni Carn~ en                   15.8       NA         NA     8.2
##  4 North~ Onyc~ carni Rode~ lc                   14.5       NA         NA     9.5
##  5 Lion   Pant~ carni Carn~ vu                   13.5       NA         NA    10.5
##  6 Domes~ Felis carni Carn~ domesticated         12.5       3.2         0.417 11.5
##  7 Arcti~ Vulp~ carni Carn~ <NA>                 12.5       NA         NA    11.5
##  8 Cheet~ Acin~ carni Carn~ lc                   12.1       NA         NA    11.9
##  9 Slow ~ Nyct~ carni Prim~ <NA>                 11         NA         NA    13
## 10 Jaguar Pant~ carni Carn~ nt                   10.4       NA         NA    13.6
## # i 73 more rows
## # i 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```
. = ottr::check("tests/p12.R")
```

```
##
## All tests passed!
```

13. [1 point] Create a new column called `rem_proportion` which is the ratio of rem sleep to total amount of sleep. Assign this new dataframe to `sleep_ratio`.

```
sleep_ratio <- sleep %>% mutate(rem_proportion = sleep_rem/sleep_total)
sleep_ratio
```

```
## # A tibble: 83 x 12
##     name    genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##     <chr>   <chr> <chr> <chr> <chr>               <dbl>     <dbl>       <dbl> <dbl>
##  1 Cheet~  Acin~ carni Carn~ lc                   12.1      NA         NA      11.9
##  2 Owl m~  Aotus omni  Prim~ <NA>                 17         1.8       NA       7
##  3 Mount~  Aplo~ herbi Rode~ nt                   14.4       2.4       NA       9.6
##  4 Great~  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 Three~  Brad~ herbi Pilo~ <NA>                 14.4       2.2        0.767   9.6
##  7 North~  Call~ carni Carn~ vu                    8.7       1.4        0.383  15.3
##  8 Vespe~  Calo~ <NA>  Rode~ <NA>                  7        NA         NA      17
##  9 Dog     Canis carni Carn~ domesticated         10.1       2.9        0.333  13.9
## 10 Roe d~  Capr~ herbi Arti~ lc                    3        NA         NA      21
## # i 73 more rows
## # i 3 more variables: brainwt <dbl>, bodywt <dbl>, rem_proportion <dbl>
```

```
. = ottr::check("tests/p13.R")
```

```
##
## All tests passed!
```

14. [1 point] Add another column to the `sleep_ratio` dataset called `bodywt_grams` which is the bodywt column in grams. Call this new dataframe `sleep_bw`.

```
sleep_bw <- sleep %>% mutate(rem_proportion = sleep_rem/sleep_total, bodywt_grams = bodywt * 1000)
sleep_bw
```

```
## # A tibble: 83 x 13
##     name    genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##     <chr>   <chr> <chr> <chr> <chr>               <dbl>     <dbl>       <dbl> <dbl>
##  1 Cheet~ Acin~ carni Carn~ lc                  12.1      NA          NA     11.9
##  2 Owl m~ Aotus omni  Prim~ <NA>                17         1.8        NA      7
##  3 Mount~ Aplo~ herbi Rode~ nt                  14.4       2.4        NA      9.6
##  4 Great~ 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 Three~ Brad~ herbi Pilo~ <NA>                14.4       2.2         0.767  9.6
##  7 North~ Call~ carni Carn~ vu                   8.7       1.4         0.383 15.3
##  8 Vespe~ Calo~ <NA>  Rode~ <NA>                 7        NA          NA     17
##  9 Dog    Canis carni Carn~ domesticated        10.1       2.9         0.333 13.9
## 10 Roe d~ Capr~ herbi Arti~ lc                   3        NA          NA     21
## # i 73 more rows
## # i 4 more variables: brainwt <dbl>, bodywt <dbl>, rem_proportion <dbl>,
## #   bodywt_grams <dbl>
```

```
. = ottr::check("tests/p14.R")
```

```
##
## All tests passed!
```

15. [1 point] Calculate the average sleep time across all the animals in the dataset using a `dplyr` function and label this value `sleep_avg`. Assign this one value to a dataframe called `avg_sleep_time`.

```
avg_sleep_time <- sleep %>% summarize(sleep_avg = mean(sleep_total))
avg_sleep_time
```

```
## # A tibble: 1 x 1
##   sleep_avg
##       <dbl>
## 1      10.4
```

```
. = ottr::check("tests/p15.R")
```

```
##
## All tests passed!
```

16. [2 points] Calculate the average sleep time for each type of "-vore". Hint: you'll need to use two `dplyr` functions! Name the columns of the dataframe `vore` and `sleep_avg` and assign the dataframe to `avg_by_vore`.

```r
avg_by_vore <- sleep %>%
                group_by(vore) %>%
                summarize(sleep_avg = mean(sleep_total))
avg_by_vore
```

```
## # A tibble: 5 x 2
##   vore    sleep_avg
##   <chr>       <dbl>
## 1 carni        10.4
## 2 herbi         9.51
## 3 insecti      14.9
## 4 omni         10.9
## 5 <NA>         10.2
```

```r
. = ottr::check("tests/p16.R")
```

```
##
## All tests passed!
```

END