Exo des chapitres

Chapitre 3

```
### 1
# Using this vector of heights in inches, create a new vector with the NAs removed.
heights <- c(63, 69, 60, 65, NA, 68, 61, 70, 61, 59, 64, 69, 63, 63, NA, 72, 65, 64, 70, 63, 65)
heights2 <- na.omit(heights)

# Use the function median() to calculate the median of the heights vector.
med <- median(heights, na.rm = TRUE)

# Use R to figure out how many people in the set are taller than 67 inches.
table(heights > 67)
```

```
##
## FALSE  TRUE
##    13     6
```

```
### 2
# What if we wanted to repeat the values 1, 2 and 3 five times, but obtain five 1s,
# five 2s and five 3s in that order? There are two possibilities - see ?rep or ?sort for help.
repet <- rep(c(1, 2, 3), each = 5)
```

Chapitre 4

```
download.file(url="https://ndownloader.figshare.com/files/2292169",
              destfile = "data/portal_data_joined.csv")
surveys <- read.csv("data/portal_data_joined.csv")

### 1
  # What is the class of the object surveys?
class(surveys)
```

```
## [1] "data.frame"
```

```
  # How many rows and how many columns are in this object?
dim(surveys)
```

```
## [1] 34786    13
```

```
  # How many species (as defined by the species_id variable) have been recorded during
    # these surveys?
table(surveys$species_id)
```

```
##
##    AB    AH    AS    BA    CB    CM    CQ    CS    CT    CU    CV    DM    DO
```

```
##   303   437     2    46    50    13    16     1     1     1     1 10596  3027
##    DS    DX    NL    OL    OT    OX    PB    PC    PE    PF    PG    PH    PI
##  2504    40  1252  1006  2249    12  2891    39  1299  1597     8    32     9
##    PL    PM    PP    PU    PX    RF    RM    RO    RX    SA    SC    SF    SH
##    36   899  3123     5     6    75  2609     8     2    75     1    43   147
##    SO    SS    ST    SU    UL    UP    UR    US    ZL
##    43   248     1     5     4     8    10     4     2
```

```
### 2
  # Create a data.frame (surveys_200) containing only the data in row 200 of the
    # surveys dataset.
surveys_200 <- surveys[200, ]

  # Notice how nrow() gave you the number of rows in a data.frame?
nrow(surveys)
```

```
## [1] 34786
```

```
    # Use that number to pull out just that last row in the inital surveys data frame.
surveys[34786, ]
```

```
##       record_id month day year plot_id species_id sex hindfoot_length weight
## 34786     30986     7   1 2000       7         PX                  NA     NA
##             genus species   taxa        plot_type
## 34786 Chaetodipus     sp. Rodent Rodent Exclosure
```

```
    # Compare that with what you see as the last row using tail() to make sure it's
        # meeting expectations.
tail(surveys)
```

```
##       record_id month day year plot_id species_id sex hindfoot_length weight
## 34781     26787     9  27 1997       7         PL   F              21     16
## 34782     26966    10  25 1997       7         PL   M              20     16
## 34783     27185    11  22 1997       7         PL   F              21     22
## 34784     27792     5   2 1998       7         PL   F              20      8
## 34785     28806    11  21 1998       7         PX                  NA     NA
## 34786     30986     7   1 2000       7         PX                  NA     NA
##             genus  species   taxa        plot_type
## 34781  Peromyscus leucopus Rodent Rodent Exclosure
## 34782  Peromyscus leucopus Rodent Rodent Exclosure
## 34783  Peromyscus leucopus Rodent Rodent Exclosure
## 34784  Peromyscus leucopus Rodent Rodent Exclosure
## 34785 Chaetodipus      sp. Rodent Rodent Exclosure
## 34786 Chaetodipus      sp. Rodent Rodent Exclosure
```

```
    # Pull out that last row using nrow() instead of the row number.
surveys[nrow(surveys), ]
```

```
##       record_id month day year plot_id species_id sex hindfoot_length weight
## 34786     30986     7   1 2000       7         PX                  NA     NA
##             genus species   taxa        plot_type
## 34786 Chaetodipus     sp. Rodent Rodent Exclosure
```

```
    # Create a new data frame (surveys_last) from that last row.
surveys_last <- surveys[nrow(surveys), ]

 # Use nrow() to extract the row that is in the middle of the surveys dataframe.
```

```
  # Store the content of this row in an object named surveys_middle.
surveys_middle <- surveys[nrow(surveys)/2, ]

 # Combine nrow() with the - notation above to reproduce the behavior of head(surveys),
   # keeping just the first through 6th rows of the surveys dataset.
head <- surveys[-7:-(nrow(surveys)), ]
head == head(surveys)
```

```
##   record_id month  day year plot_id species_id  sex hindfoot_length weight
## 1      TRUE  TRUE TRUE TRUE    TRUE       TRUE TRUE            TRUE     NA
## 2      TRUE  TRUE TRUE TRUE    TRUE       TRUE TRUE            TRUE     NA
## 3      TRUE  TRUE TRUE TRUE    TRUE       TRUE TRUE              NA     NA
## 4      TRUE  TRUE TRUE TRUE    TRUE       TRUE TRUE              NA     NA
## 5      TRUE  TRUE TRUE TRUE    TRUE       TRUE TRUE              NA     NA
## 6      TRUE  TRUE TRUE TRUE    TRUE       TRUE TRUE              NA     NA
##   genus species taxa plot_type
## 1  TRUE    TRUE TRUE      TRUE
## 2  TRUE    TRUE TRUE      TRUE
## 3  TRUE    TRUE TRUE      TRUE
## 4  TRUE    TRUE TRUE      TRUE
## 5  TRUE    TRUE TRUE      TRUE
## 6  TRUE    TRUE TRUE      TRUE
```

```
### 3
 # There are a few mistakes in this hand-crafted data.frame.
 # Can you spot and fix them? Don't hesitate to experiment!
animal_data <- data.frame(
       animal = c("dog", "cat", "sea cucumber", "sea urchin"),
       feel = c("furry", "furry", "squishy", "spiny"),
       weight = c(45, 8, 1.1, 0.8))


### 4
 # Using the function installed.packages(), verify if the lubridate package is installed.
 # If not, install it from CRAN.
ip <- rownames(installed.packages())

"lubridate" %in% ip
```

```
## [1] TRUE
```

```
### 5
 # Construct a matrix of dimension 1000 by 3 of normally distributed
  # data (mean 0, standard deviation 1).
set.seed(123)
m <- matrix(rnorm(3000), ncol = 3)
dim(m)
```

```
## [1] 1000    3
```

Chapitre 5

```r
library(tidyverse)
### 1
 # Using pipes, subset the surveys data to include animals collected before 1995
  # and retain only the columns year, sex, and weight.
x <- surveys %>% filter(year < "1995") %>%
  select(year, sex, weight)


### 2
 # Create a new data frame from the surveys data that meets the following criteria:
   # contains only the species_id column
   # and a new column called hindfoot_half containing values that are
      # half the hindfoot_length values.
   # In this hindfoot_half column, there are no NAs and all values are less than 30.
hindfoot <- surveys %>%
  select(species_id) %>%
  mutate(hindfoot_half = surveys$hindfoot_length/2) %>%
  filter(!is.na(hindfoot_half)) %>% filter(hindfoot_half < "30")


### 3
 # How many animals were caught in each plot_type surveyed?
nb_animals_plot_type <- surveys %>% count(plot_type)

 # Use group_by() and summarize() to find the mean, min,
   # and max hindfoot length for each species (using species_id).
   # Also add the number of observations (hint: see ?n).
Moy_Med_species_id <- surveys %>% filter(!is.na(hindfoot_length),
                                         !is.na(species_id)) %>%
  group_by(species_id) %>%
  summarise(Moyenne = mean(hindfoot_length),
            Minimum = min(hindfoot_length),
            Maximum = max(hindfoot_length),
            Number_of_Observations = n())

# What was the heaviest animal measured in each year?
   # Return the columns year, genus, species_id, and weight.
Max_weight <- surveys %>%
  filter(!is.na(weight),
         !is.na(year),
         !is.na(genus),
         !is.na(species_id)) %>%
  select(year, genus, species_id, weight) %>%
  group_by(year) %>%
  summarise(Maximum_weight = max(weight))

      # Correction
MAX_weight <- surveys %>%
    filter(!is.na(weight)) %>%
    group_by(year) %>%
    filter(weight == max(weight)) %>%
    select(year, genus, species, weight) %>%
    arrange(year)
```

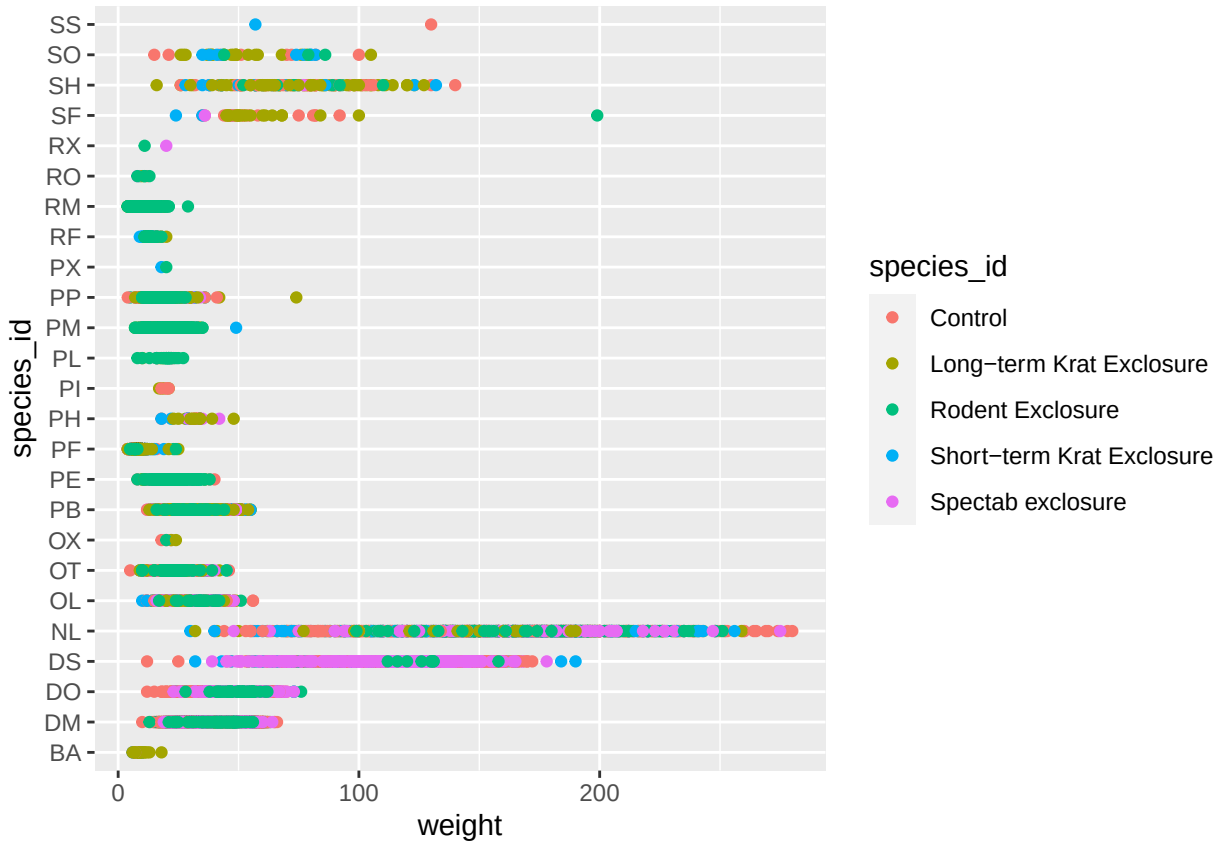
```r
### 4
 # The surveys data set has two measurement columns: hindfoot_length and weight.
 # Use pivot_longer() to create a dataset where we have a key column called measurement
 # and a value column that takes on the value of either hindfoot_length or weight.
longer <- surveys %>% pivot_longer(cols = c(weight, hindfoot_length),
                                   names_to = "measurement",
                                   values_to = "value")

 # With this new data set, calculate the average of each measurement in each year
  # for each different plot_type.
  # Then use pivot_wider() to generate a data set with a column for hindfoot_length and weight.
l2 <- longer %>% filter(!is.na(year), !is.na(plot_type)) %>%
  group_by(measurement, year, plot_type) %>%
  summarise(Moyenne = mean(value, na.rm = TRUE)) %>%
  pivot_wider(names_from = measurement,
                   values_from = Moyenne)
```

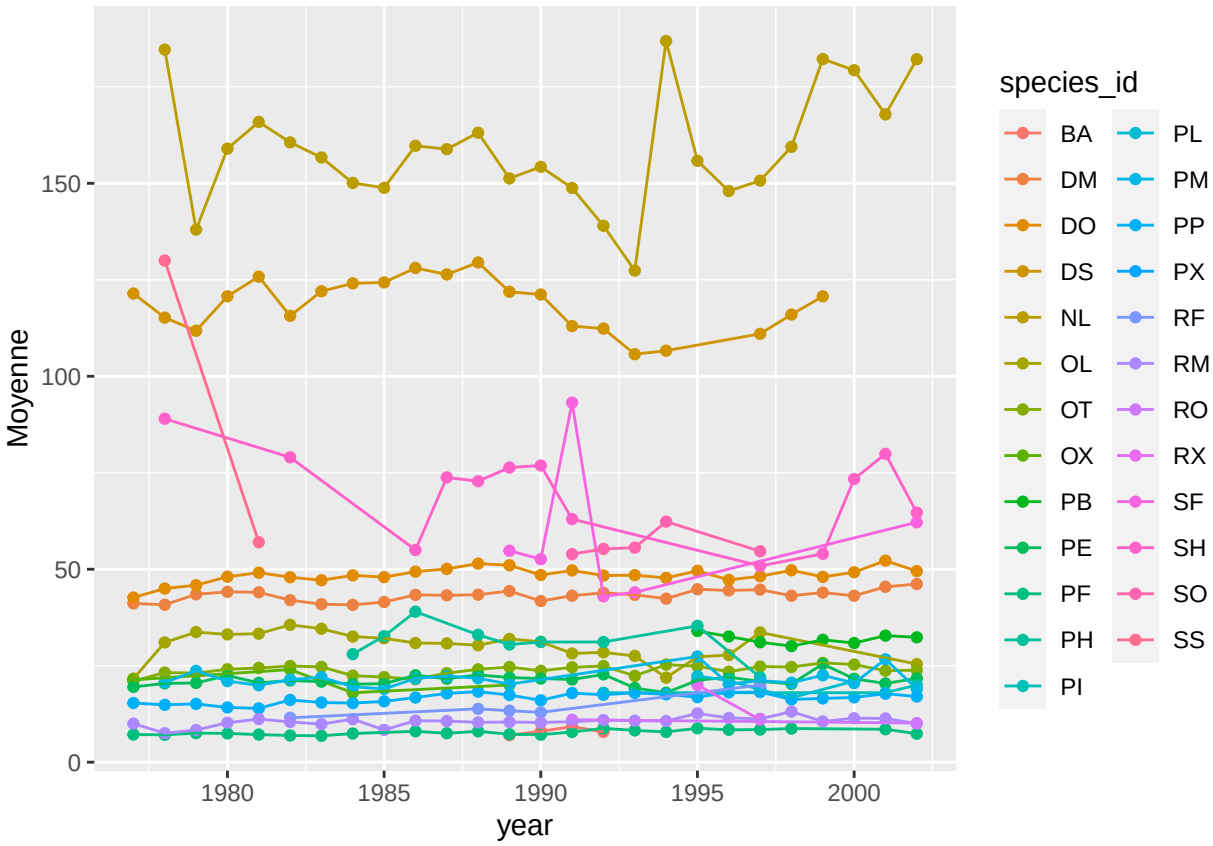
Chapitre 6

```r
library(ggplot2)

### 1
  # Create a scatter plot of weight over species_id with the plot types
    # showing in different colors. Is this a good way to show this type of data?
gg <- surveys %>% filter(!is.na(weight), !is.na(species_id))
gg %>% ggplot(aes( x = weight,
                       y = species_id,
                       colour = species_id)) + geom_point(aes(color = plot_type))
```




```
### 2
 # Use what you just learned to create a plot that depicts how the average weight
  # of each species changes through the years.
g2 <- surveys %>% filter(!is.na(year), !is.na(weight)) %>%
  group_by(year, species_id) %>%
  summarise(Moyenne = mean(weight))
g2 %>% ggplot(aes(x = year,
                  y = Moyenne,
                  colour = species_id)) +
  geom_point() + geom_line()
```



Chapitre 7

```
### 1
# Using the full_join function demonstrated above, join tables jdf4 and jdf5.
  # What has happened for observations P26039 and P02468?
library("rWSBIM1207")
data(jdf)
full_join(jdf4, jdf5) %>% filter(uniprot %in% c("P26039", "P02468"))
```

```
## # A tibble: 2 x 6
##   uniprot organelle          entry     gene_name description            organism
##   <chr>   <chr>              <chr>     <chr>     <chr>                  <chr>
## 1 P26039  Actin cytoskeleton TLN1_MOU~ <NA>      <NA>                   <NA>
## 2 P02468  <NA>               <NA>      Lamc1     Laminin subunit gamma~ Mmus
```

```
### 2
 # Join tables jdf4 and jdf5, keeping only observations in jdf4.
left_join(jdf4, jdf5)
```

```
## # A tibble: 10 x 6
##    uniprot  organelle                entry    gene_name description     organism
##    <chr>    <chr>                    <chr>    <chr>     <chr>           <chr>
##  1 P26039   Actin cytoskeleton       TLN1_MO~ <NA>      <NA>            <NA>
##  2 Q99PL5   Endoplasmic reticulum/G~ RRBP1_M~ <NA>      <NA>            <NA>
##  3 Q6PB66   Mitochondrion            LPPRC_M~ <NA>      <NA>            <NA>
##  4 P11276   Extracellular matrix     FINC_MO~ <NA>      <NA>            <NA>
```

```
##  5 Q6PR54   Nucleus - Chromatin      RIF1_MO~ <NA>      <NA>            <NA>
##  6 Q05793   Extracellular matrix     PGBM_MO~ <NA>      <NA>            <NA>
##  7 P19096   Cytosol                  FAS_MOU~ Fasn      Fatty acid syn~ Mmus
##  8 Q9JKF1   Plasma membrane          IQGA1_M~ <NA>      <NA>            <NA>
##  9 Q9QZQ1-2 Plasma membrane          AFAD_MO~ <NA>      <NA>            <NA>
## 10 Q6NS46   Nucleus - Non-chromatin  RRP5_MO~ <NA>      <NA>            <NA>
```

```
# Join tables jdf4 and jdf5, keeping only observations in jdf5.
right_join(jdf4, jdf5)
```

```
## # A tibble: 5 x 6
##   uniprot organelle entry   gene_name description                        organism
##   <chr>   <chr>     <chr>   <chr>     <chr>                              <chr>
## 1 P19096  Cytosol   FAS_MO~ Fasn      Fatty acid synthase                Mmus
## 2 P02468  <NA>      <NA>    Lamc1     Laminin subunit gamma-1            Mmus
## 3 P08113  <NA>      <NA>    Hsp90b1   Endoplasmin                        Mmus
## 4 Q8BI84  <NA>      <NA>    Mia3      Melanoma inhibitory activity pro~ Mmus
## 5 Q6P5D8  <NA>      <NA>    Smchd1    Structural maintenance of chromo~ Mmus
```

```
# Join tables jdf4 and jdf5, keeping observations observed in both tables.
inner_join(jdf4, jdf5)
```

```
## # A tibble: 1 x 6
##   uniprot organelle entry     gene_name description         organism
##   <chr>   <chr>     <chr>     <chr>     <chr>               <chr>
## 1 P19096  Cytosol   FAS_MOUSE Fasn      Fatty acid synthase Mmus
```

Chapitre 10