STA 141 Worksheet 6

Richard McCormick

October 24, 2023

Due Date: Tuesday, October 31, 2023 before 11:00am.

Instructions

Worksheets must be turned in as a PDF file through Canvas. The worksheet is worth a total of **15 points**, which is 3 percent of your overall grade.

Exercises

Begin by running the following code block to add the packages we need to use to our library.

Exercise 1

(a) Import the Flagstaff Weather dataset from the following URL and save as weather: `https://github.com/dere`

```
weather <- read_csv( 'https://github.com/dereksonderegger/141/raw/master/data-raw/FlagMaxTemp.
```

```
## New names:
## Rows: 365 Columns: 34
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (34): ...1, Year, Month, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ...
## 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.
## * `` -> `...1`
```

```
head( weather )
```

```
## # A tibble: 6 x 34
##    ...1  Year Month   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`  `10`
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1  1985     5  71.1  71.1  68    68    64.9  64.0  64.0  64.9  69.1  66.0
## 2     2  1985     6  63.0  63.0  64.9  60.1  69.1  75.9  82.0  86    84.9  84.0
## 3     3  1985     7  81.0  86    90.0  88.0  91.9  91.9  89.1  88.0  90.0  87.1
## 4     4  1985     8  77    68    78.1  80.1  82.0  81.0  82.9  82.9  80.1  80.1
## 5     5  1985     9  82.9  75.0  73.9  72.0  66.9  63.0  63.0  64.0  68    64.9
## 6     6  1985    10  64.0  60.1  64.0  71.1  71.1  75.0  69.1  54.0  51.1  55.0
## # i 21 more variables: `11` <dbl>, `12` <dbl>, `13` <dbl>, `14` <dbl>,
## #   `15` <dbl>, `16` <dbl>, `17` <dbl>, `18` <dbl>, `19` <dbl>, `20` <dbl>,
## #   `21` <dbl>, `22` <dbl>, `23` <dbl>, `24` <dbl>, `25` <dbl>, `26` <dbl>,
## #   `27` <dbl>, `28` <dbl>, `29` <dbl>, `30` <dbl>, `31` <dbl>
```

(b) Is this data in a wide or a long format and why?

The data appears to be in a mix of wide and long formats. It is in long format for the years and months, but in wide format for the days.

(c) How many rows and how many columns are in this dataset (at present)?

```
dim( weather )
```

```
## [1] 365  34
```

There are 365 rows, with 34 columns.

(d) Reshape the data into the long format where it has only four columns: `Year, Month, Day, Max.temp`

```
weather <- weather %>%
    pivot_longer(
    4:34,                # which columns to apply this to
    names_to  = 'Day',   # What should I call the column of old column names
    values_to = 'Max.Temp')

weather <- subset( weather, select = c( "Year", "Month", "Day", "Max.Temp" ) )

head( weather )
```

```
## # A tibble: 6 x 4
##    Year Month Day   Max.Temp
##   <dbl> <dbl> <chr>    <dbl>
## 1  1985     5 1         71.1
## 2  1985     5 2         71.1
## 3  1985     5 3         68
## 4  1985     5 4         68
## 5  1985     5 5         64.9
## 6  1985     5 6         64.0
```

(e) How many observations and how many columns are in this dataset now?

```
dim( weather )
```

```
## [1] 11315     4
```

There are now 11,315 observations and 4 columns in the dataset.

(f) Filter the dataset so you only have measurements from the Year 2012.

```
weather.2012 <- weather %>% filter( Year == 2012 )
head( weather.2012 )
```

```
## # A tibble: 6 x 4
##    Year Month Day   Max.Temp
##   <dbl> <dbl> <chr>    <dbl>
## 1  2012     1 1         53.1
## 2  2012     1 2         48.0
## 3  2012     1 3         57.0
## 4  2012     1 4         51.1
## 5  2012     1 5         57.0
## 6  2012     1 6         48.0
```

(g) How many `NA` values are there in the 2012 dataset?

```
sum( is.na( weather.2012$Max.Temp ) )
```

```
## [1] 21
```

Exercise 2

(a) Download the file named `wide.data.csv` from Canvas and read it into R.

```
wide.data <- read_csv( 'wide.data.csv' )
```

```
## Rows: 4 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Country.Name
## dbl (4): 2017, 2018, 2019, 2020
##
## 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.
```

```
head( wide.data )
```

```
## # A tibble: 4 x 5
##   Country.Name    `2017`   `2018`   `2019`   `2020`
##   <chr>            <dbl>    <dbl>    <dbl>    <dbl>
## 1 United Kingdom 2.70e12  2.90e12  2.88e12 2.76e12
## 2 United States  1.95e13  2.05e13  2.14e13 2.19e13
## 3 Mexico         1.16e12 NA       NA       1.09e12
## 4 Canada         2.10e13  2.05e13  1.98e13 1.94e13
```

(b) Is this data in a wide format or is the file named incorrectly?

The data is named correctly. This data is in wide format.

(e) Tidy the data so that it is in the long format. Give the new columns names that you believe are appropriate.

```r
long.data <- wide.data %>%
    pivot_longer(
    2:5,                # which columns to apply this to
    names_to  = 'Year',   # What should I call the column of old column names
    values_to = 'Value')

head( long.data )
```

```
## # A tibble: 6 x 3
##   Country.Name   Year    Value
##   <chr>          <chr>   <dbl>
## 1 United Kingdom 2017  2.70e12
## 2 United Kingdom 2018  2.90e12
## 3 United Kingdom 2019  2.88e12
## 4 United Kingdom 2020  2.76e12
## 5 United States  2017  1.95e13
## 6 United States  2018  2.05e13
```

(f) Remove any rows from the dataset that have missing values.

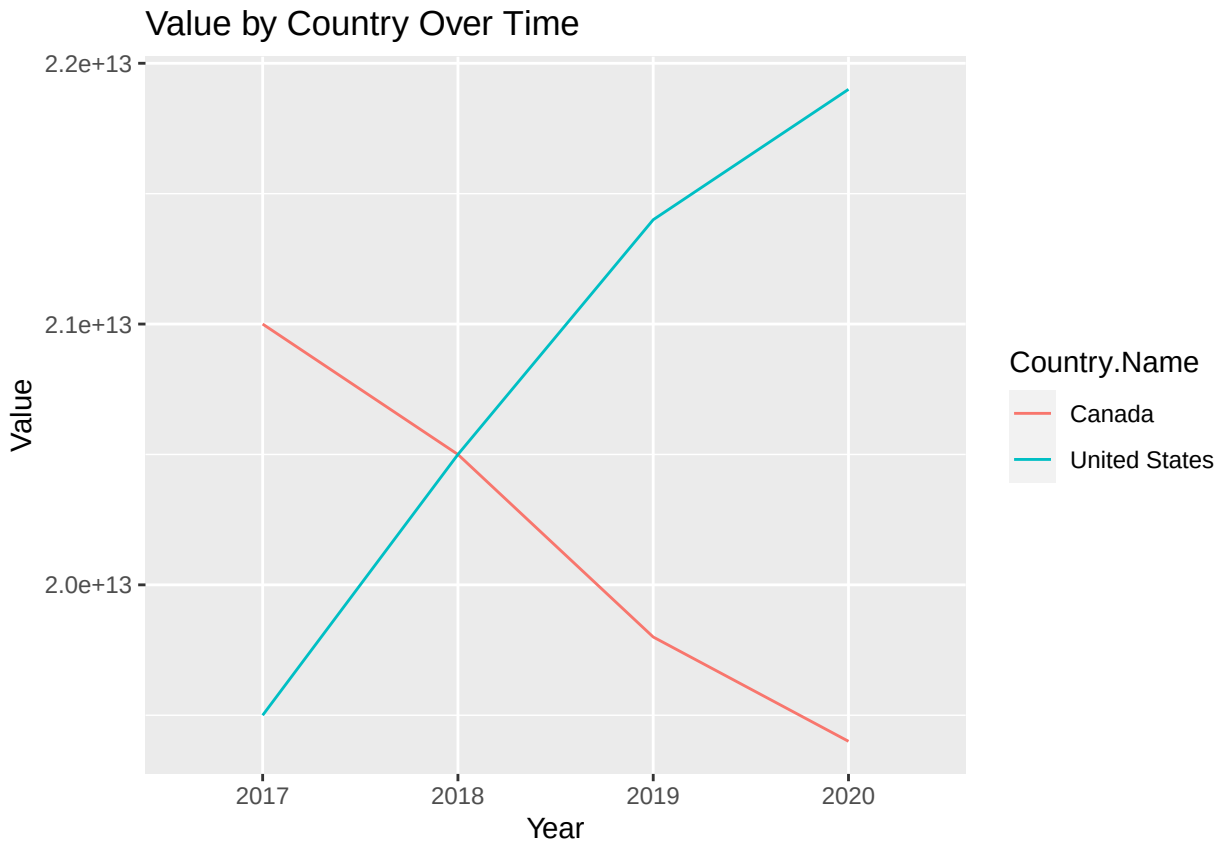
```r
long.data <- long.data %>% drop_na()
long.data
```

```
## # A tibble: 14 x 3
##    Country.Name   Year    Value
##    <chr>          <chr>   <dbl>
##  1 United Kingdom 2017  2.70e12
##  2 United Kingdom 2018  2.90e12
##  3 United Kingdom 2019  2.88e12
##  4 United Kingdom 2020  2.76e12
##  5 United States  2017  1.95e13
##  6 United States  2018  2.05e13
##  7 United States  2019  2.14e13
##  8 United States  2020  2.19e13
##  9 Mexico         2017  1.16e12
## 10 Mexico         2020  1.09e12
## 11 Canada         2017  2.10e13
## 12 Canada         2018  2.05e13
## 13 Canada         2019  1.98e13
## 14 Canada         2020  1.94e13
```

(g) Filter the data so that you only have the United States and Canada's values. Plot a line graph of each of the values against the year.

```r
long.data <- long.data %>% filter( Country.Name == "United States" |
                                   Country.Name == "Canada" )
```

```
long.plot <- ggplot( data=long.data ) +
  geom_line( aes( x = Year, y = Value, group=Country.Name, color=Country.Name ) ) +
  labs( title="Value by Country Over Time", x="Year", y="Value" )

long.plot
```



(h) Are the two countries following the same trend in this (unknown) variable?

No. The two countries are following opposite trends - the United States is going up, while Canada is going down.

Exercise 3

(a) Download the file named `msleep.csv` from Canvas and read it into R. This is an edited version of the dataset that is available freely in R.

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

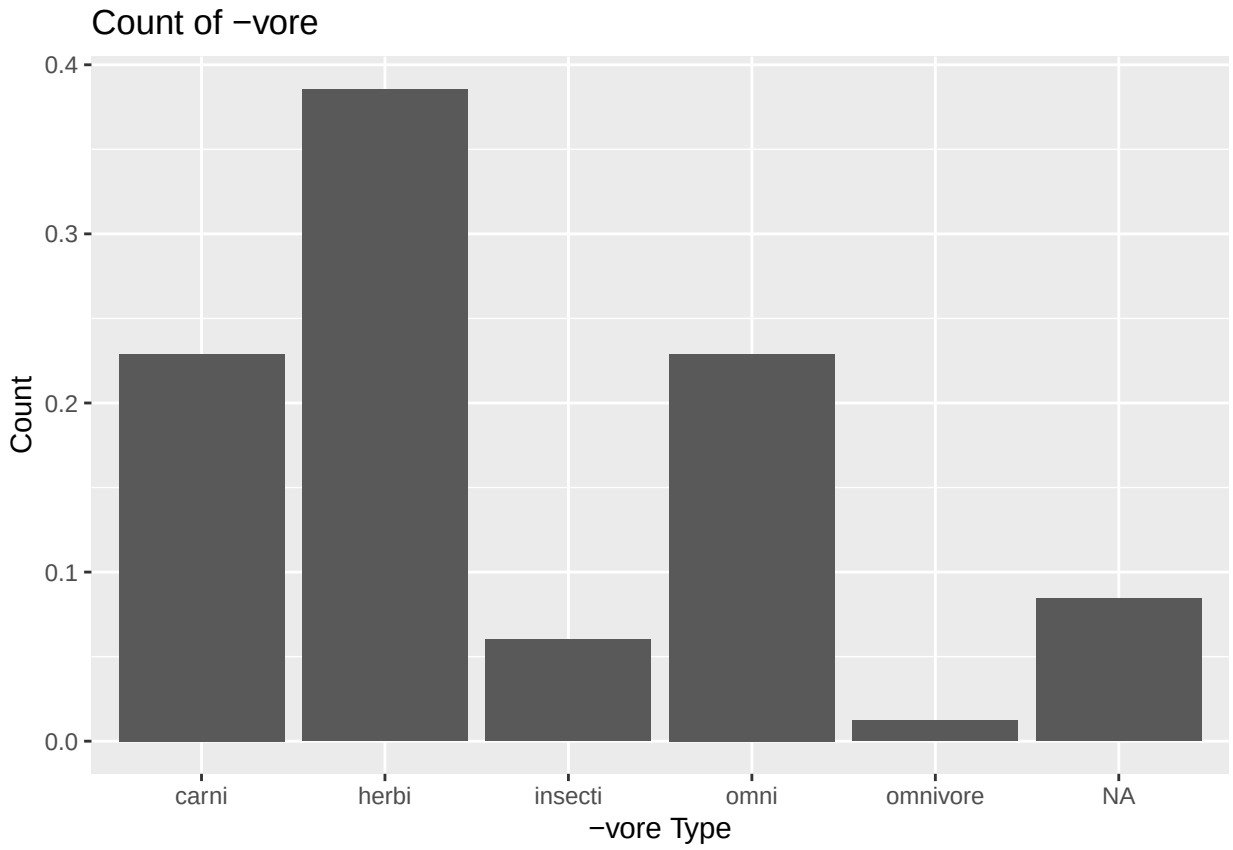
```
## Rows: 83 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): name, genus, vore, order, conservation
## dbl (6): sleep_total, sleep_rem, sleep_cycle, awake, brainwt, bodywt
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

(b) One of mammals has a NA value for the `awake` variable. Find it and replace it with the correct value. Hint: Once you locate it you can replace the value using slicing: `data[row,column] <- value`

```
sleep.data[ 64, 'awake' ] <- 11
```

(c) Plot a bar chart of the `vore` variable. Are there any issues with the levels?

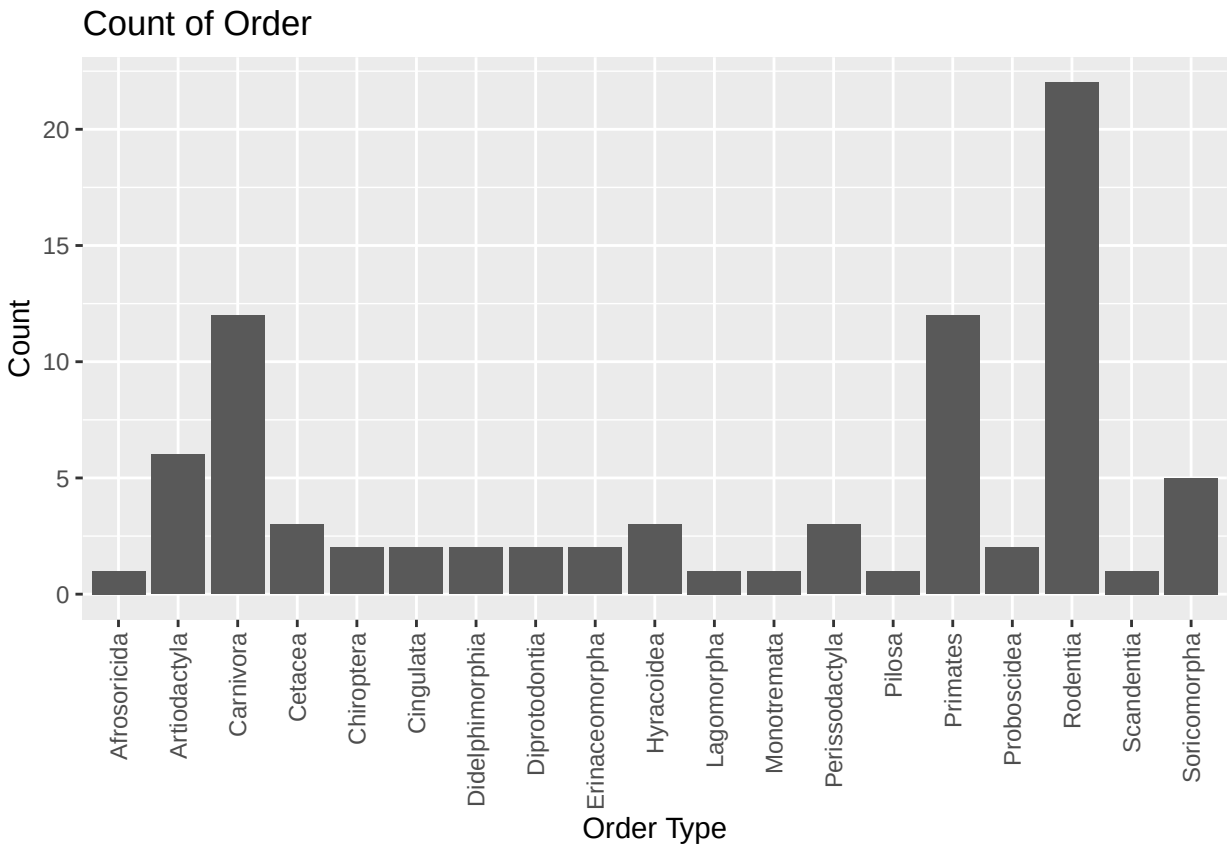
```
sleep.plot <- ggplot( data = sleep.data ) +
  geom_bar( aes( x=vore,
                 y=after_stat( count )/sum(after_stat( count ) ) ) ) +
  labs( title="Count of -vore", y="Count", x="-vore Type" )

sleep.plot
```



There seems to be an issue that there is a significant amount of NA-vores. This means that the data is not complete.

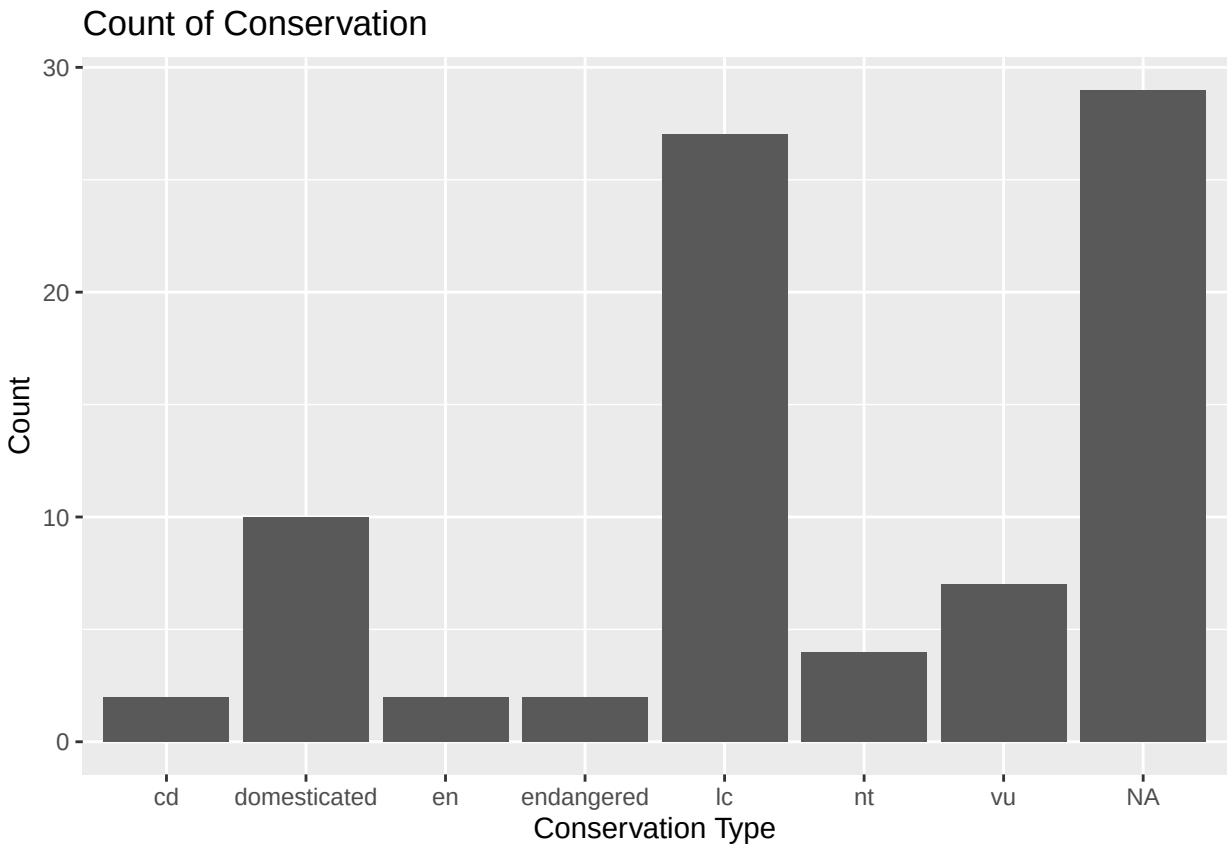
(d) Do the same for the `order` and `conservation` variables. Do there appear to be any issues with these variables?

```
sleep.plot <- ggplot( data = sleep.data ) +
  geom_bar( aes( x=order,
                 y=after_stat( count ) ) ) +
  labs( title="Count of Order", y="Count", x="Order Type" ) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

sleep.plot
```



There doesn't appear to be any issue with the Order variable.

```
sleep.plot <- ggplot( data = sleep.data ) +
  geom_bar( aes( x=conservation,
                 y=after_stat( count ) ) ) +
  labs( title="Count of Conservation", y="Count", x="Conservation Type" )

sleep.plot
```

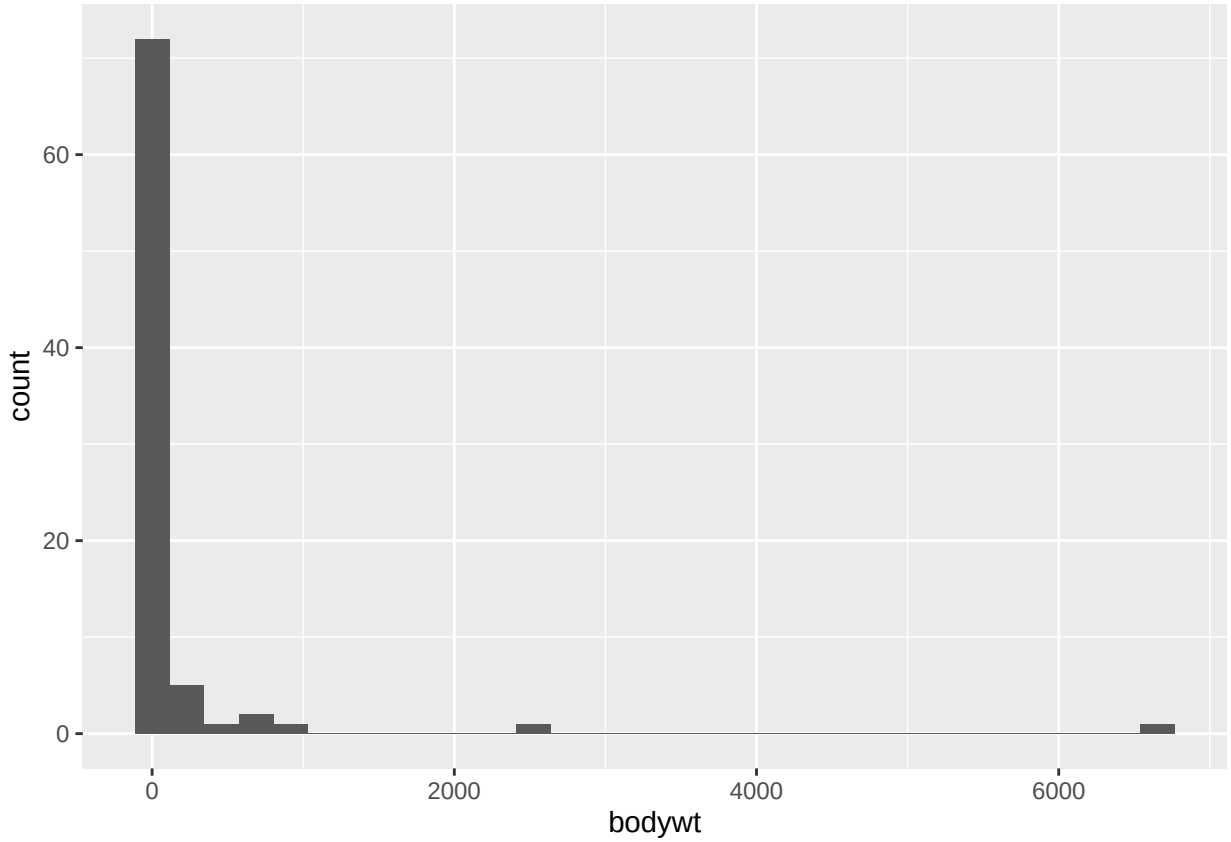


There appears to be a very large number of NA in the Conservation variable, which indicates that the data is not complete.

(e) Finally, plot a histogram of the `body weight` variable. Does there appear to be any issue with the values of this variable?

```
sleep.plot <- ggplot( data=sleep.data ) +
  geom_histogram( aes( x=bodywt ) )

sleep.plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The issue with this data is that the majority of the data is clustered around a very low body weight, but there are a few outliers with very high body weight. This makes it very difficult to accurately assess the data, as we can't really see the majority of the data.