HW4_R_TEAM_GA

2022-09-30

HW-ggplot 2-dplyr 1

February 10, 2022

Instructions
- After completing the questions, upload both the .RMD and PDF files to Canvas.
- Use dplyr functions wherever possible.

Notes
- For this homework, we will use the msleep dataset from ggplot2.
- Useful packages: ggplot2, ggthemes
- Some useful functions: scale fill colorblind(), scale y log10(), facet wrap(), scale color discrete()

Problem 1

Data: Use msleep dataset from ggplot2.

(i) Load the tidyverse and ggthemes packages and the msleep data set.

```
library(ggthemes)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("ggthemes")
library(describer)
library(ggplot2)
library(dplyr)
data(msleep)
```

(ii) How many mammals are in the msleep data frame? How many variables? Use two R functions to get this information.

```
ncol(msleep) ## number of variables
```

```
## [1] 11
```

```
names(msleep) ## name of variables
```

```
##  [1] "name"         "genus"        "vore"         "order"        "conservation"
##  [6] "sleep_total"  "sleep_rem"    "sleep_cycle"  "awake"        "brainwt"
## [11] "bodywt"
```

```
nrow(msleep) # number of mammals
```

```
## [1] 83
```

```
msleep$name ## names of the mammals (83 mammals,hence it is a mammals data set, no filtering is require
```

```
##  [1] "Cheetah"                         "Owl monkey"
##  [3] "Mountain beaver"                 "Greater short-tailed shrew"
##  [5] "Cow"                             "Three-toed sloth"
##  [7] "Northern fur seal"               "Vesper mouse"
##  [9] "Dog"                             "Roe deer"
## [11] "Goat"                            "Guinea pig"
## [13] "Grivet"                          "Chinchilla"
## [15] "Star-nosed mole"                 "African giant pouched rat"
## [17] "Lesser short-tailed shrew"       "Long-nosed armadillo"
## [19] "Tree hyrax"                      "North American Opossum"
## [21] "Asian elephant"                  "Big brown bat"
## [23] "Horse"                           "Donkey"
## [25] "European hedgehog"               "Patas monkey"
## [27] "Western american chipmunk"       "Domestic cat"
## [29] "Galago"                          "Giraffe"
## [31] "Pilot whale"                     "Gray seal"
## [33] "Gray hyrax"                      "Human"
## [35] "Mongoose lemur"                  "African elephant"
## [37] "Thick-tailed opposum"            "Macaque"
## [39] "Mongolian gerbil"                "Golden hamster"
## [41] "Vole "                           "House mouse"
## [43] "Little brown bat"                "Round-tailed muskrat"
## [45] "Slow loris"                      "Degu"
## [47] "Northern grasshopper mouse"      "Rabbit"
## [49] "Sheep"                           "Chimpanzee"
## [51] "Tiger"                           "Jaguar"
## [53] "Lion"                            "Baboon"
## [55] "Desert hedgehog"                 "Potto"
## [57] "Deer mouse"                      "Phalanger"
## [59] "Caspian seal"                    "Common porpoise"
## [61] "Potoroo"                         "Giant armadillo"
## [63] "Rock hyrax"                      "Laboratory rat"
## [65] "African striped mouse"           "Squirrel monkey"
## [67] "Eastern american mole"           "Cotton rat"
## [69] "Mole rat"                        "Arctic ground squirrel"
## [71] "Thirteen-lined ground squirrel"  "Golden-mantled ground squirrel"
## [73] "Musk shrew"                      "Pig"
## [75] "Short-nosed echidna"             "Eastern american chipmunk"
## [77] "Brazilian tapir"                 "Tenrec"
## [79] "Tree shrew"                      "Bottle-nosed dolphin"
## [81] "Genet"                           "Arctic fox"
## [83] "Red fox"
```

(iii) You want to explore if total sleep time has a relationship with mammal body weight.

- Write out a question about the relationship.

```
# What is the effect of "total_sleep" on Mammals "bodywt"? or
# What is the effect of "bodywt" on mammals "total_sleep"?  Either way we can explore the relationship
# # e.gg: "lm(msleep$bodywt~msleep$sleep_total)" or otherwise# The easiest be a linear regression
```

- What is your response variable and what type is it?

Note: The answer varies depend on the question we ask.

```
    # If we ask: What is the effect of "total_sleep" on Mammals "bodywt"
    #
    # Response variable is "bodywt".
    #
    # If we ask: What is the effect of "bodywt" on mammals "total_sleep"?
    #
    # Response variable is "total_sleep".
```

- What is your explanatory variable and what type is it?

```
     # If we ask: What is the effect of "total_sleep" on Mammals "bodywt"
     #
     # Explanatory Variable is "total_sleep".
     #
     # If we ask: What is the effect of "bodywt" on mammals "total_sleep"?
     #
     # Explanatory Variable is "bodywt".
```

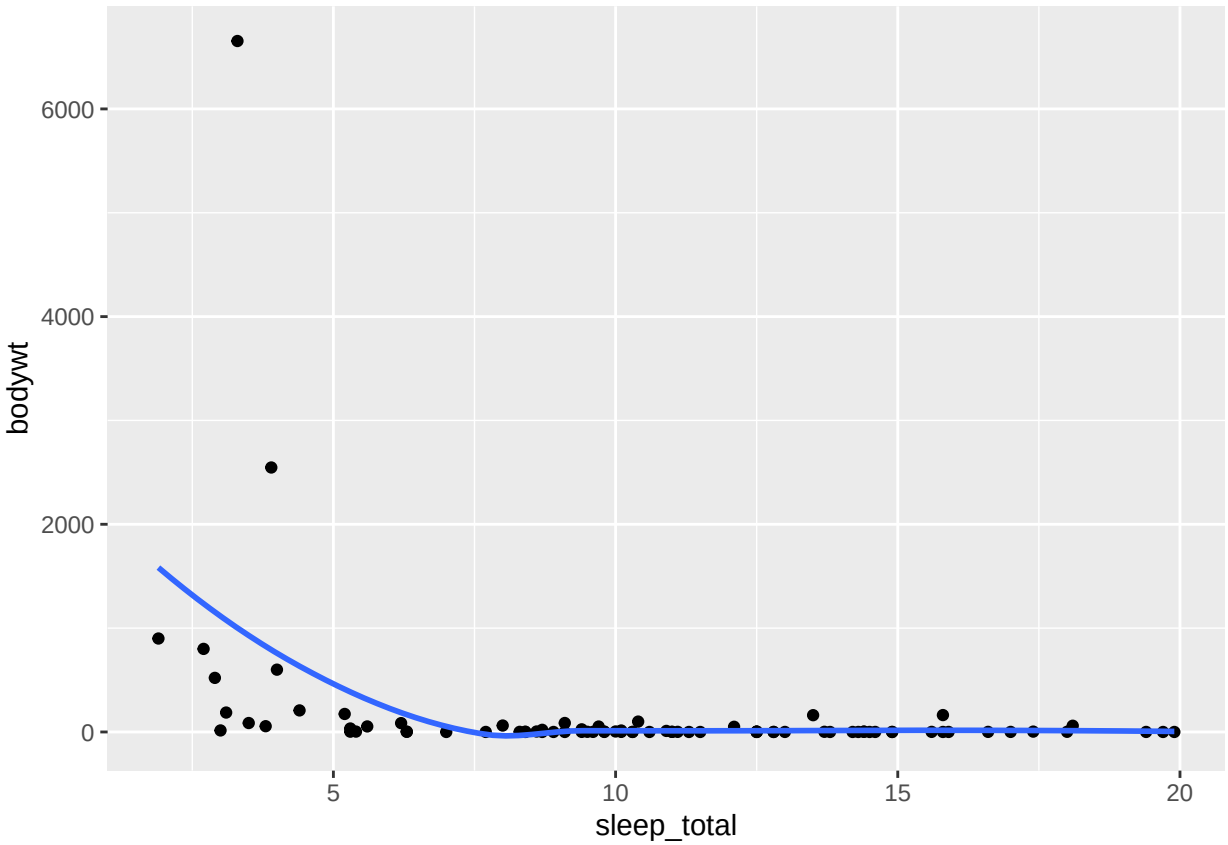
- What is the appropriate type of plot given the types of variables?

```
# Hence the variables are numbers, would be better to see in scatter plot (geom_point). And I think Lin
```

- Create the appropriate plot with body weight against the total amount of sleep.

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total)) +
geom_point() +
geom_smooth(se=FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
#geom_smooth(se = FALSE)
```

- Interpret the plot in one sentence: what does the shape tell you about the relationship?

The linear regression method show that it is hard to see the relationship.

```
### trying the linear regression model to see the actual relationship (prediction)

lm(msleep$bodywt~msleep$sleep_total) #If we ask: What is the effect of "total_sleep" on Mammals "bodywt
```

```
##
## Call:
## lm(formula = msleep$bodywt ~ msleep$sleep_total)
##
## Coefficients:
##        (Intercept)  msleep$sleep_total
##             741.71              -55.16
```

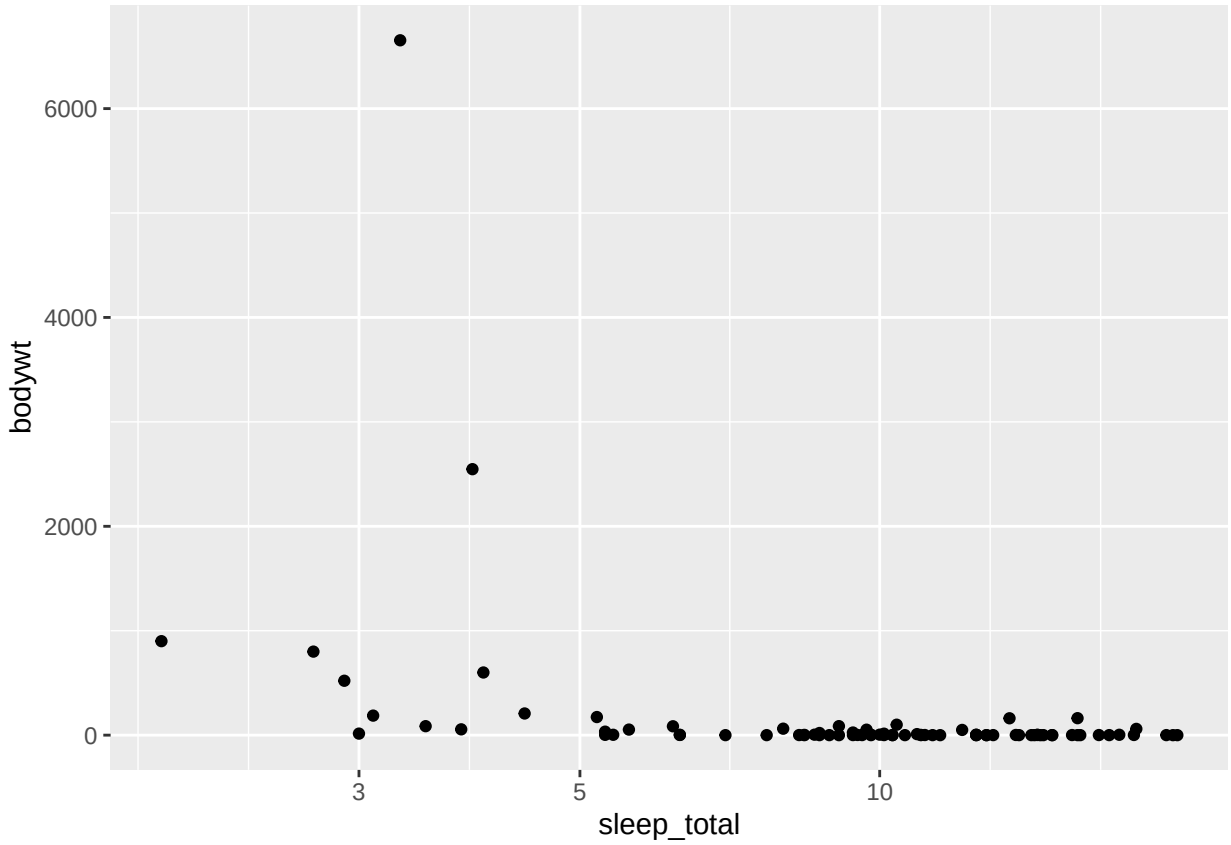
The lm predict that there is a negative relationship b/n total sleep and body weight. To be precise, a one unit change of total sleep would make the body weight to drop by 55 unit.

(iv) When you see a curved or skewed relationship in a plot, you can often get rid of the curve or skew by taking a log transformation of either the explanatory or the response variable or both.

- Create three plots:

(a) log(x): when only the predictor transformed to log.

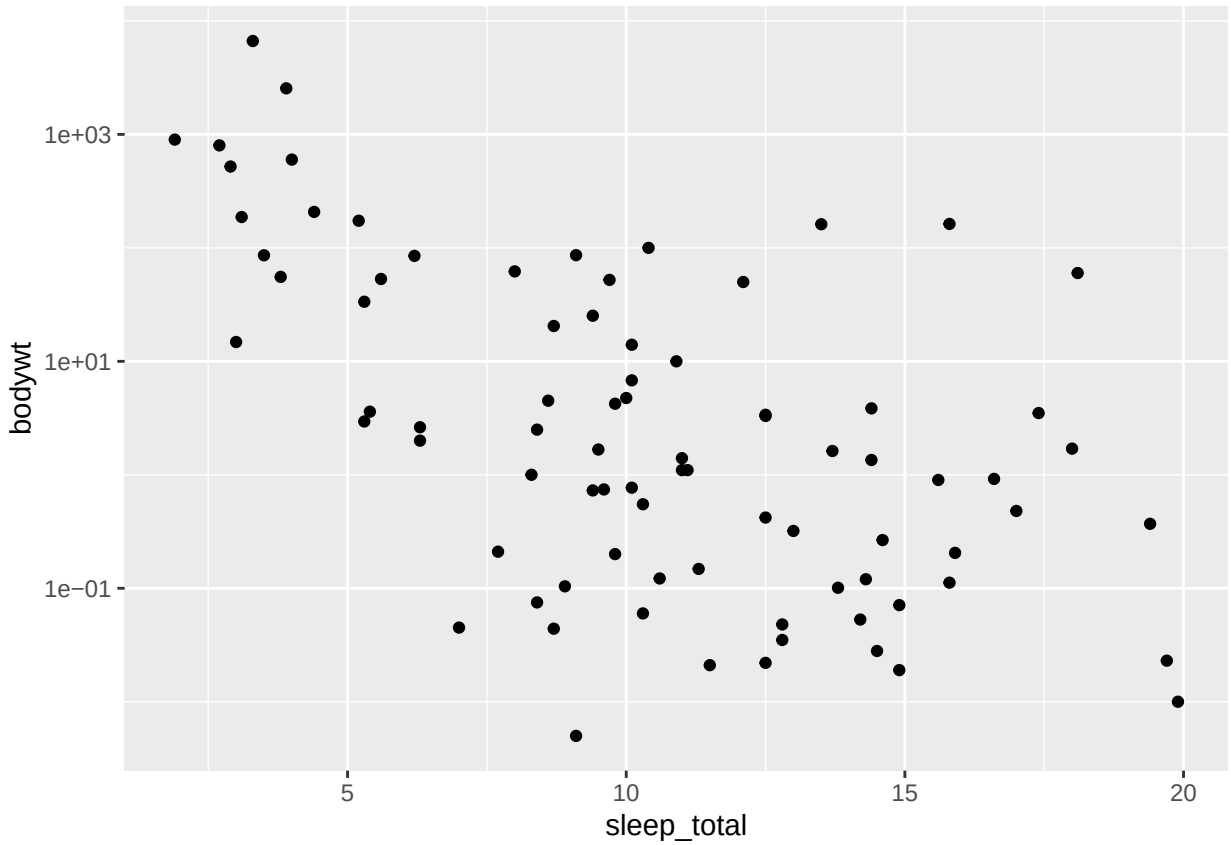
```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total)) +
scale_x_log10() +
geom_point()
```




```
#geom_smooth(se = FALSE)
```

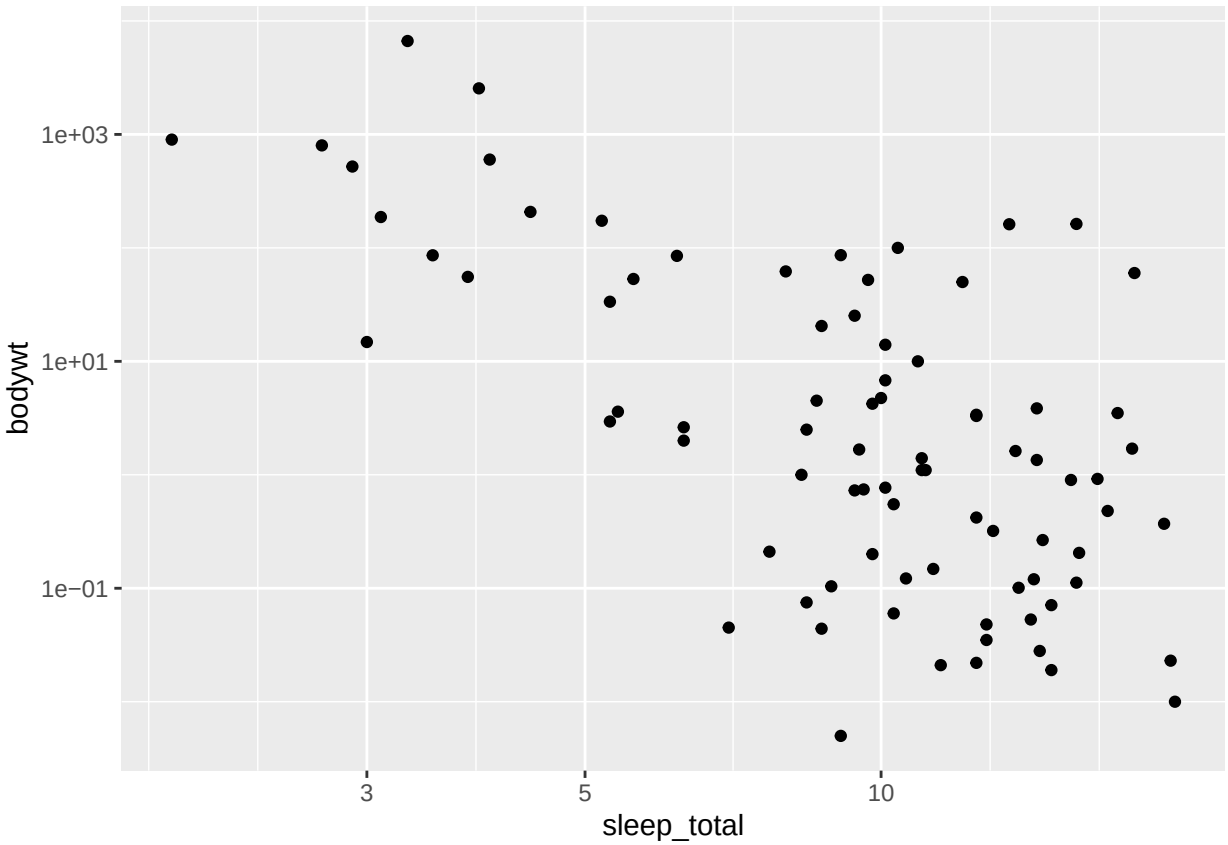
(b) log(y)and: when only the response variable transformed to log.

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total)) +
scale_y_log10() +
geom_point()
```



(c) log(x) and log (y): When both variables are transformed to log.

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total)) +
scale_x_log10() +
scale_y_log10()+
geom_point()
```



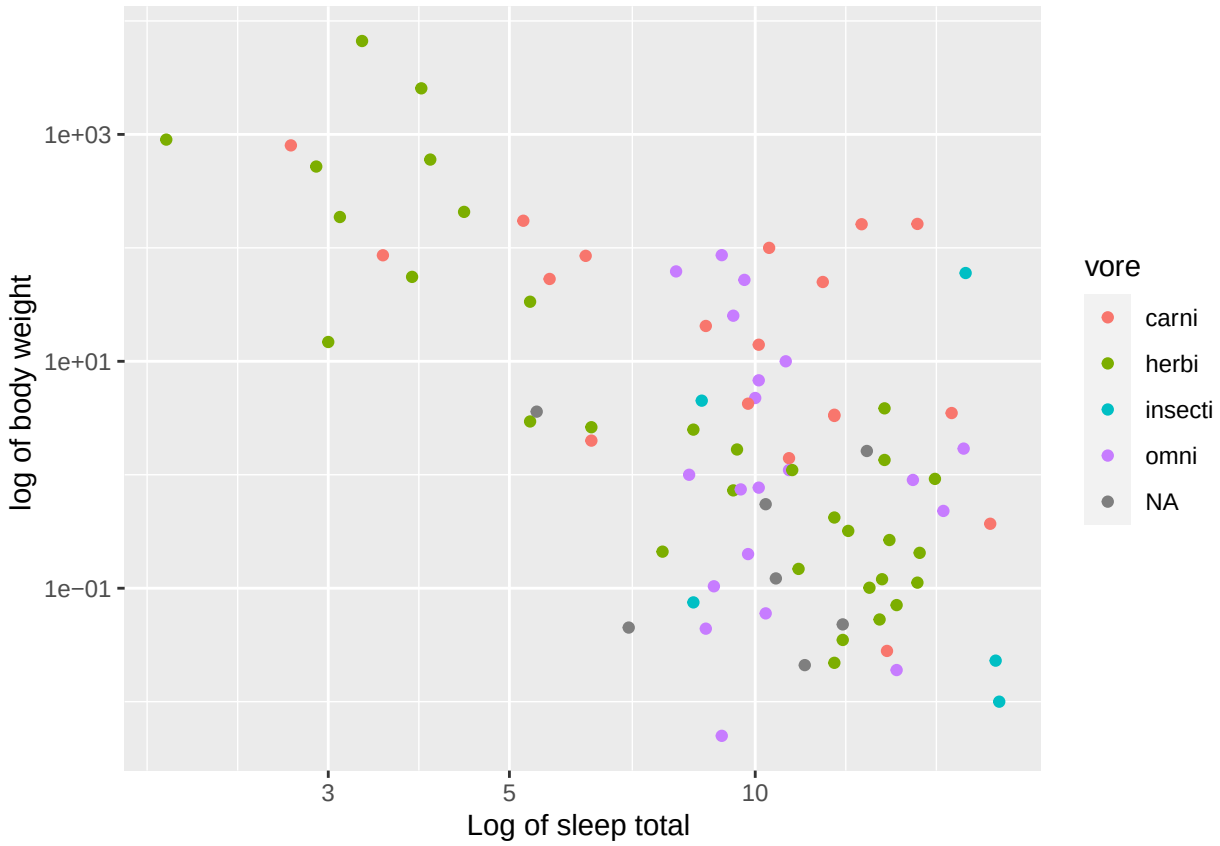
- Which plot appears best (most linear) to you and why?

When both variables are transformed to log is most linear. From the graph, the first one (when we take the log of sleep_total) seems most linear but that is due to scaling issue and outliers. But, insteady of log transforamtion, we could take the outliers out as it is and do the plottig.

(v) Color code the plot in part 4 by the diet of the animals (vore).

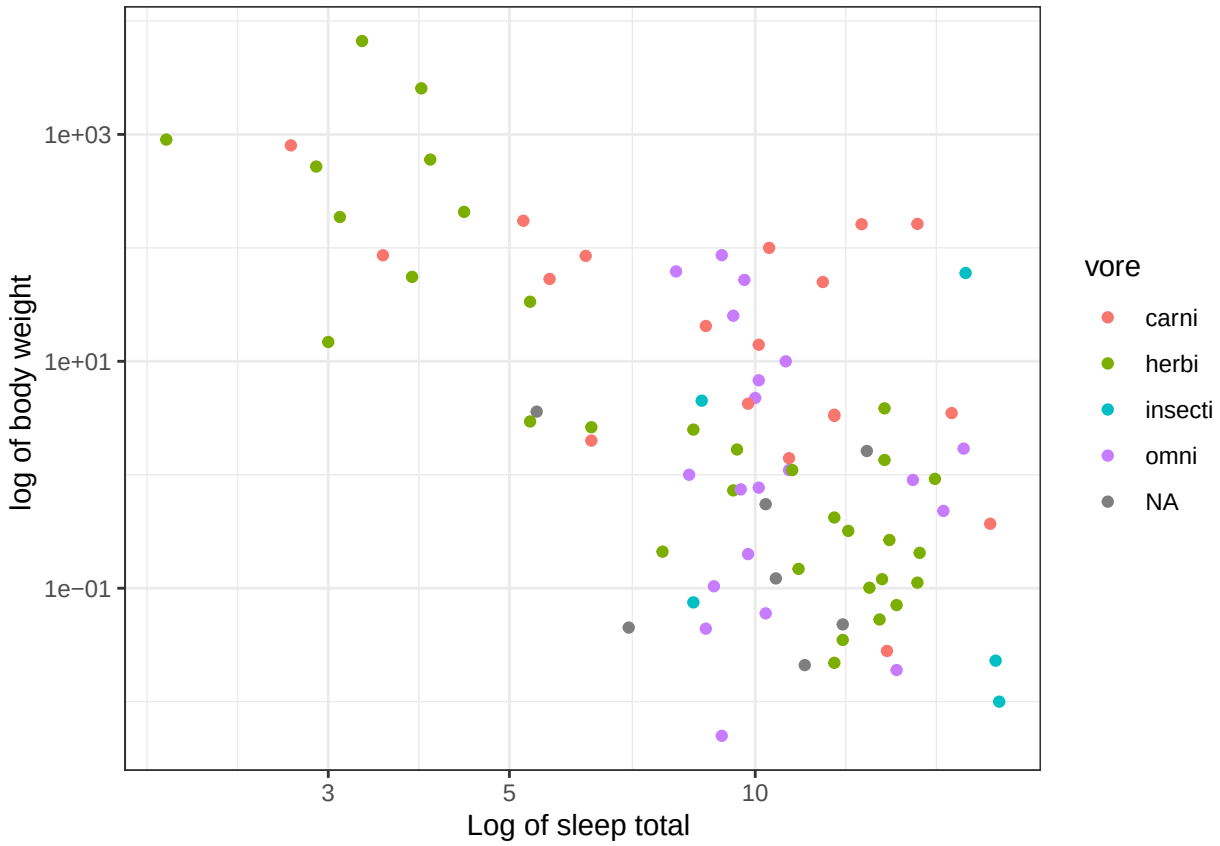
- Make the axis labels nice,

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total, color= vore)) +
scale_x_log10() +
scale_y_log10()+
xlab("Log of sleep total ")+
ylab("log of body weight ")+
geom_point()
```



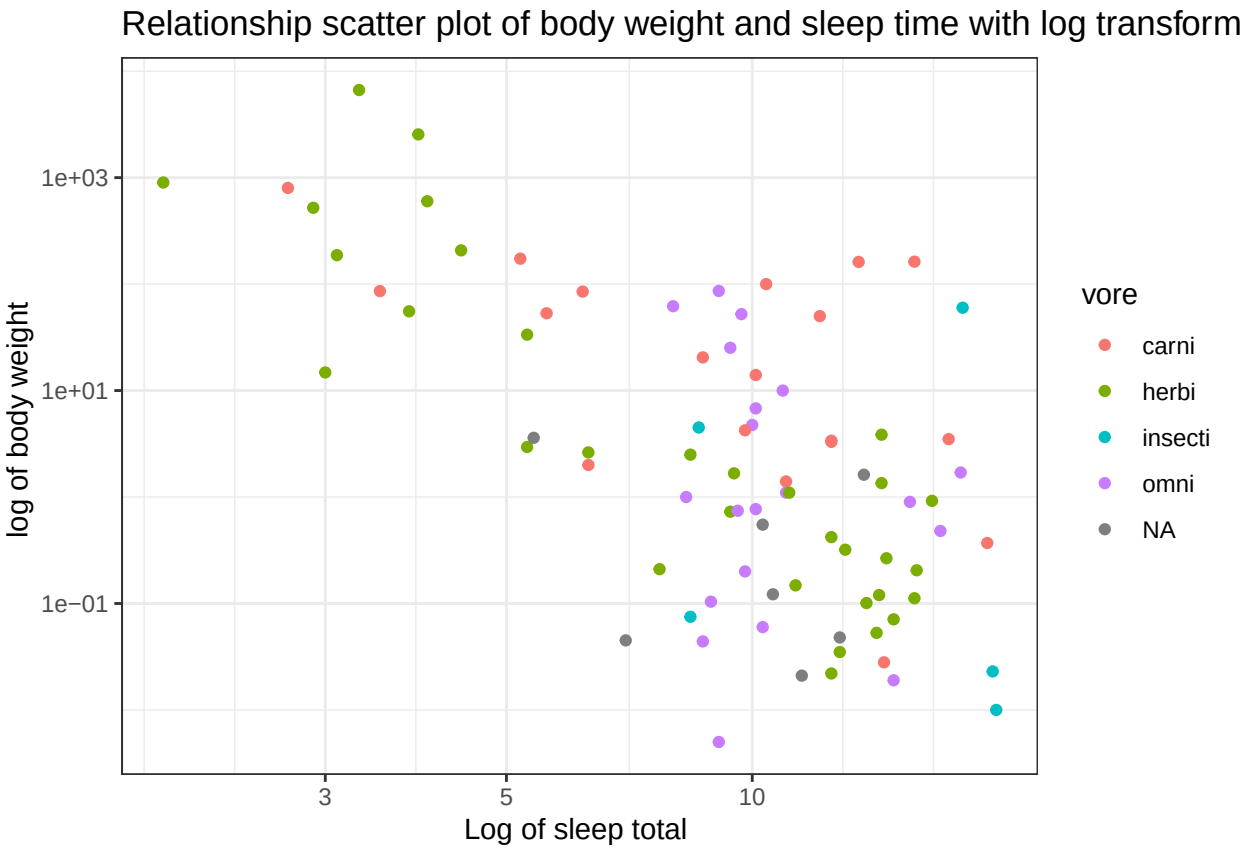
- Change the theme to black and white, and

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total, color= vore)) +
scale_x_log10() +
scale_y_log10()+
xlab("Log of sleep total ")+
ylab("log of body weight ")+
theme_bw()+
geom_point()
```



- Add a meaningful title to the plot.

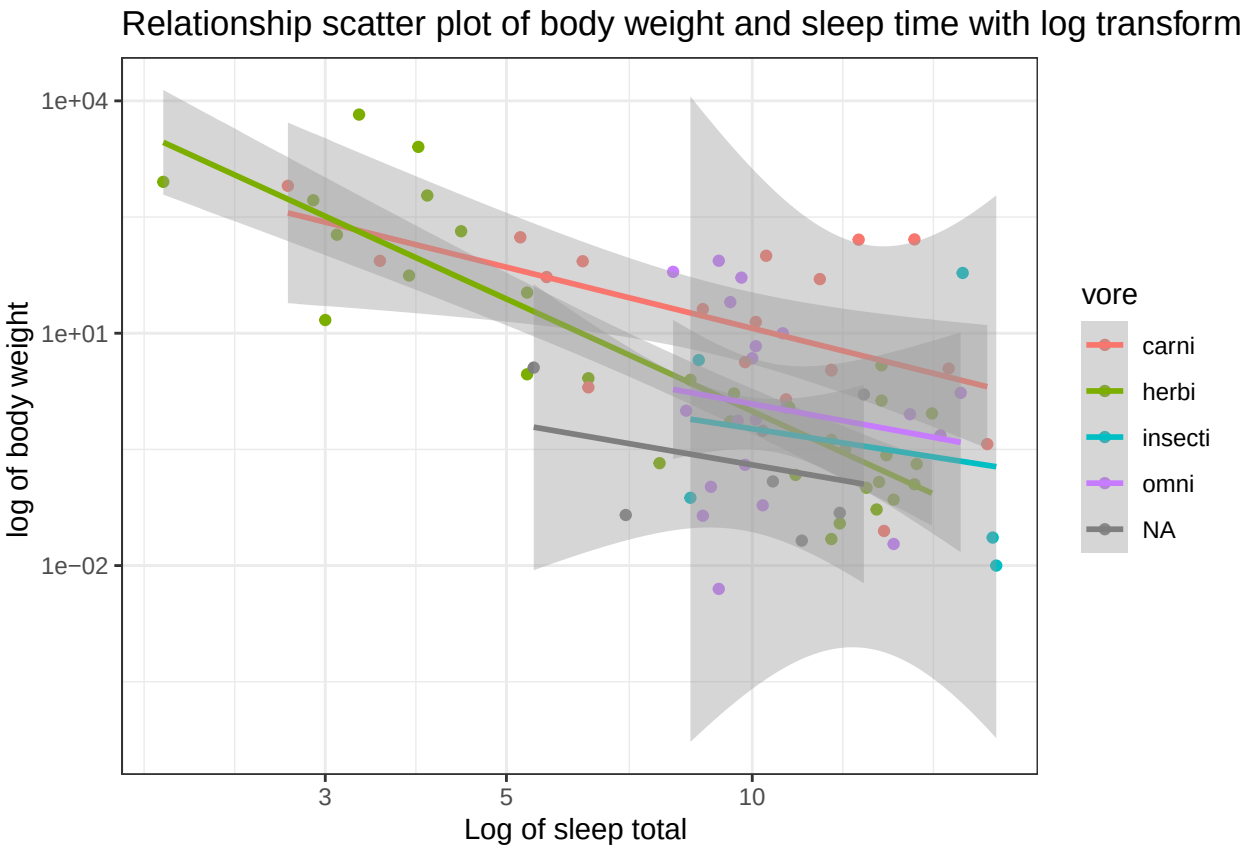
```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total, color= vore)) +
scale_x_log10() +
scale_y_log10()+
xlab("Log of sleep total ")+
ylab("log of body weight ")+
ggtitle("Relationship scatter plot of body weight and sleep time with log transformation")+
theme_bw()+
geom_point()
```



(vi) Copy the plot from part 5 and add an OLS (Ordinary Least Square) line (without standard errors) for each vore category.

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total, color= vore)) +
scale_x_log10() +
scale_y_log10()+
xlab("Log of sleep total ")+
ylab("log of body weight ")+
ggtitle("Relationship scatter plot of body weight and sleep time with log transformation")+
theme_bw()+
geom_point() +
geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



- Does the effect of body weight on sleep total appear larger for some diets?

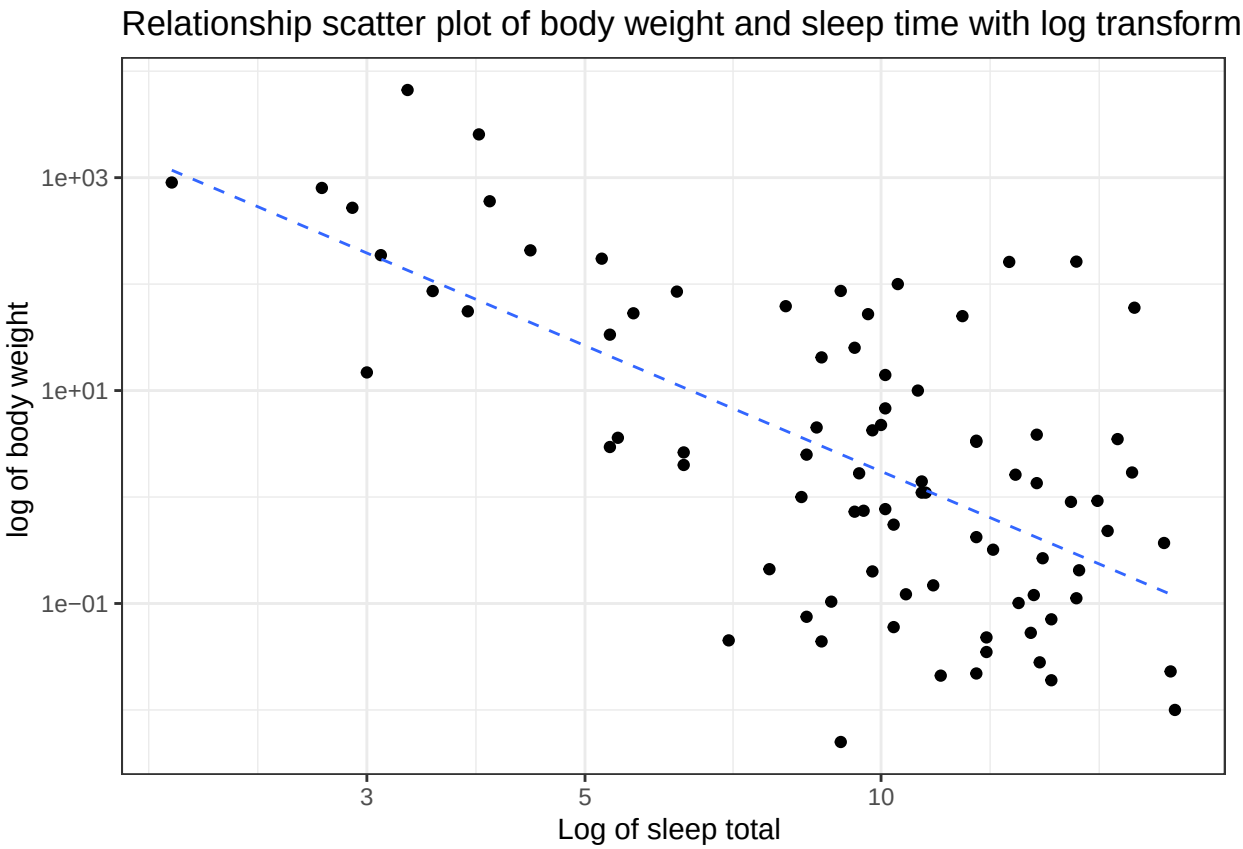
```
#Yes, on Herbivore Mammals.
```

(vii) Copy the plot from 6 and add the overall (across all vore types) OLS line (without standard errors) to the above plot.

- Make sure this new line is dashed, and has width of 0.5.

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total)) +
scale_x_log10() +
scale_y_log10()+
xlab("Log of sleep total ")+
ylab("log of body weight ")+
ggtitle("Relationship scatter plot of body weight and sleep time with log transformation")+
theme_bw()+
geom_point() +
geom_smooth(se= FALSE, size = 0.5, linetype = "dashed", method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



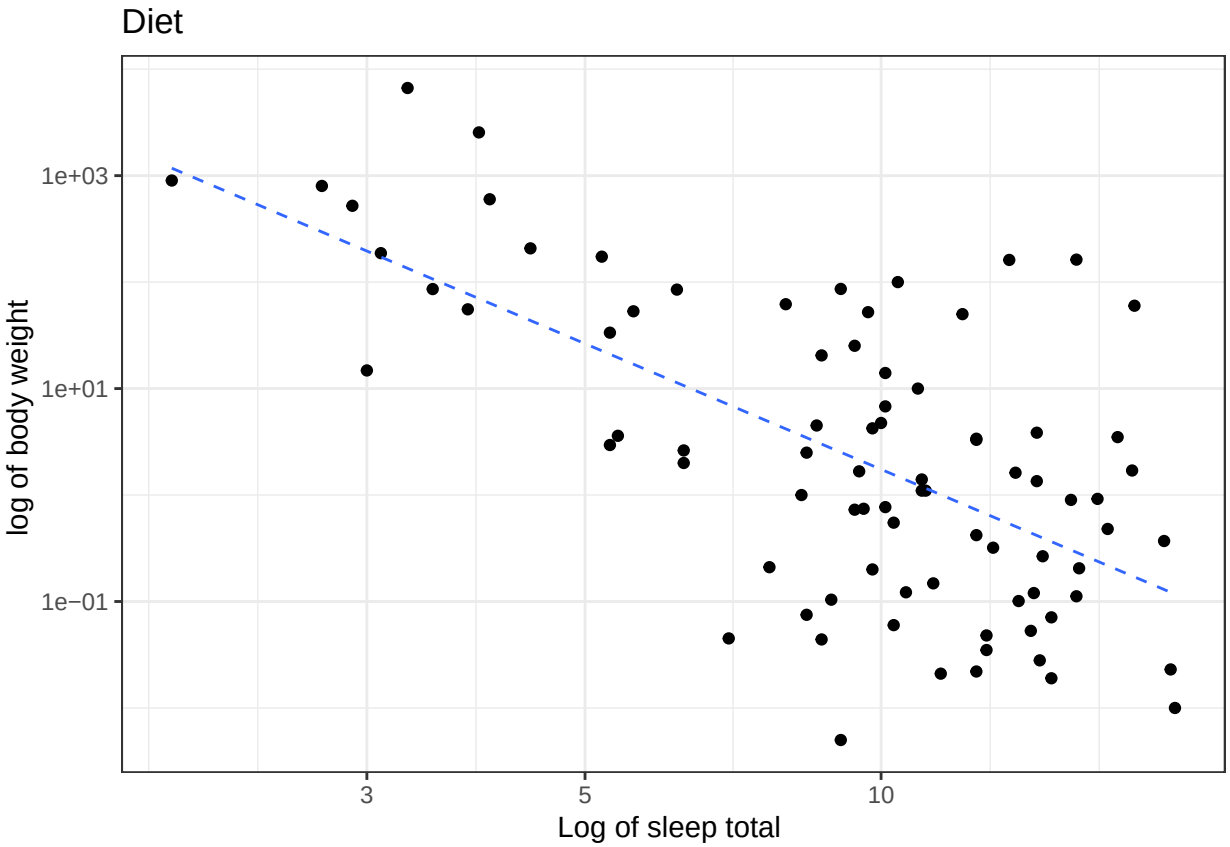
- In one sentence, how does this line compare to the individual lines?

```
#The individual lines are fitted to the vore group and varies for each group. It seems possible to see
```

viii) Copy the plot from 7 and change the title of the legend to "Diet".

```
ggplot(data = msleep, mapping = aes(y = bodywt, x = sleep_total)) +
scale_x_log10() +
scale_y_log10()+
xlab("Log of sleep total ")+
ylab("log of body weight ")+
ggtitle("Diet")+
theme_bw()+
geom_point() +
geom_smooth(se= FALSE, size = 0.5, linetype = "dashed", method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Problem 2

Data: flights data frame from the nycflights13 package.

(i) Load and review the data

- Load the tidyverse nycflights 13 packages

```
library(nycflights13)
```

- Load the flights data frame

```
data("flights")
```

- What are the variables

```
names(flights) # 19 variables
```

```
##  [1] "year"           "month"          "day"            "dep_time"
##  [5] "sched_dep_time" "dep_delay"      "arr_time"       "sched_arr_time"
##  [9] "arr_delay"      "carrier"        "flight"         "tailnum"
## [13] "origin"         "dest"           "air_time"       "distance"
## [17] "hour"           "minute"         "time_hour"
```

- How many observations (rows) are there?

```
dim(flights) #336776 rows
```

```
## [1] 336776     19
```

- Look at the first three rows

```
head(flights,3)
```

```
## # A tibble: 3 x 19
##    year month   day dep_time sched_dep~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##   <int> <int> <int>    <int>       <int>   <dbl>   <int>   <int>   <dbl> <chr>
## 1  2013     1     1      517         515       2     830     819      11 UA
## 2  2013     1     1      533         529       4     850     830      20 UA
## 3  2013     1     1      542         540       2     923     850      33 AA
## # ... with 9 more variables: flight <int>, tailnum <chr>, origin <chr>,
## #   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>, and abbreviated variable names 1: sched_dep_time,
## #   2: dep_delay, 3: arr_time, 4: sched_arr_time, 5: arr_delay
```

(ii) Worst Plane to Fly

- Which planes (tailnum) have the three worst (highest) average departure delay record?

```
l=arrange(flights, desc(dep_delay))

head(l$tailnum,3)
```

```
## [1] "N384HA" "N504MQ" "N517MQ"
```

(a) How many flights did each make?

```
head(l[, c(12,11)],3)
```

```
## # A tibble: 3 x 2
##   tailnum flight
##   <chr>    <int>
## 1 N384HA      51
## 2 N504MQ    3535
## 3 N517MQ    3695
```

```
# "N384HA"= 51;
# "N504MQ"=3535;
# "N517MQ"=3695
```

(b) Now only look tailnums where each flew more than 12 flights and find the three tailnums with the hi

```
# filtering all flights with flight more than 12: k dataframe
k <-flights%>%
  filter(flight>12)

# arranging the k dataframe desc
l=arrange(k, desc(dep_delay))

#the top three delayed flight with tailnum, flight, and dep_delay(column 12, 11, 6)
head(l[, c(12,11,6)],3)
```

```
## # A tibble: 3 x 3
##   tailnum flight dep_delay
##   <chr>    <int>     <dbl>
## 1 N384HA      51      1301
## 2 N504MQ    3535      1137
## 3 N517MQ    3695      1126
```

(iii) Best Time of Day to Fly

- Use a plot to see what hour of the day you should fly to minimize your expected (average) delay time?

```
#The question is not clear by average (either (dep_delay + arr_delay)/2 or just dep_delay average). I a


Flight_dep_time_delay= flights[,c(4,6)]
dep_delay_mean = mean(flights$dep_delay, na.rm = TRUE)
dep_delay_mean
```

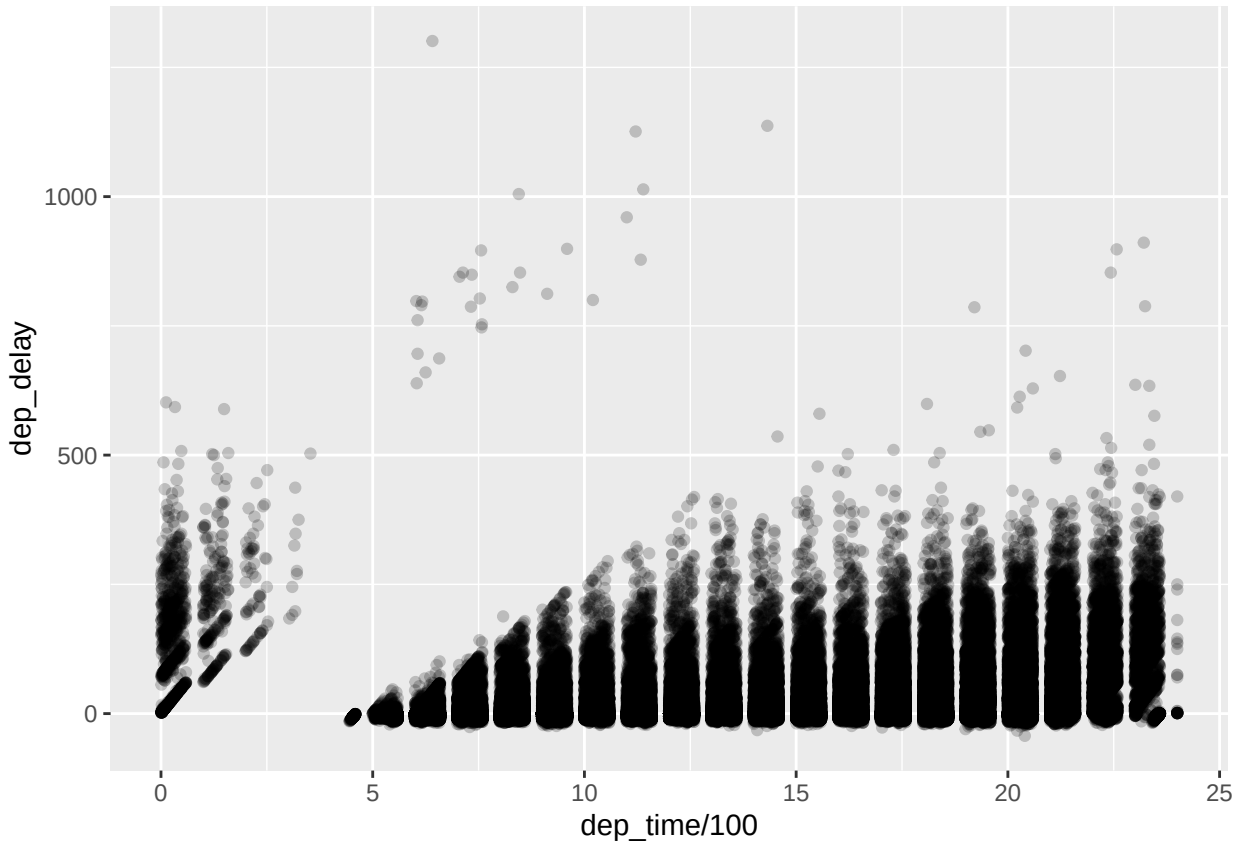
```
## [1] 12.63907
```

```
Flight_dep_time_delay
```

```
## # A tibble: 336,776 x 2
##    dep_time dep_delay
##       <int>     <dbl>
##  1      517         2
##  2      533         4
##  3      542         2
##  4      544        -1
##  5      554        -6
##  6      554        -4
##  7      555        -5
##  8      557        -3
##  9      557        -3
## 10      558        -2
## # ... with 336,766 more rows
```

```
ggplot(data =flights, mapping = aes(x = dep_time/100, y = dep_delay)) +
  geom_point(alpha = 0.2)
```

```
## Warning: Removed 8255 rows containing missing values (geom_point).
```




```
## there seems to be two blocks of time where flights depart.

# *** In the dataset 2400 is midnight and on average, early morning flights has fewer dep_time. (400 -1
```