# Expected answers

2023-09-27

## Part 1: body mass and metabolic rate

**Ex.1)**

```
#install the tidyverse package here
#after you do this, you can comment the line so it doesn't reinstall this every time you
↪  knit your document.
#install.packages("tidyverse")
#then, load the tidyverse package here
library(tidyverse)
```

**Ex.2)**

Expected output

```
# type your code here
mass_metabolism<-read_csv("mass_metabolism.csv")
```

```
## Rows: 25 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): family
## dbl (2): body_mass, metabolic_rate
##
## 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(mass_metabolism)
```

```
## # A tibble: 6 x 3
##   body_mass metabolic_rate family
##       <dbl>          <dbl> <chr>
## 1     32000           50.0 Antilocapridae
## 2     37800           52.0 Antilocapridae
## 3    347000          307.  Bovidae
## 4      4200           10.1 Bovidae
## 5    196500          230.  Bovidae
## 6    100000          149.  Bovidae
```

Answer: It's a tibble. There are three columns.

**Ex.3)**

```
# type your code here
dim(mass_metabolism)
```

```
## [1] 25   3
```

Answer: 25 rows, 3 columns

**Ex.4)**

```
# type your code here
is.tibble(mass_metabolism)
```

```
## Warning: `is.tibble()` was deprecated in tibble 2.0.0.
## i Please use `is_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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

Answer: we had seen with `head` that this is a tibble. The `is.tibble` command confirms this by returning TRUE.

**Ex.5)**

```
# type your code here
str(mass_metabolism)
```

```
## spc_tbl_ [25 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ body_mass     : num [1:25] 32000 37800 347000 4200 196500 ...
##  $ metabolic_rate: num [1:25] 50 52 306.8 10.1 230.1 ...
##  $ family        : chr [1:25] "Antilocapridae" "Antilocapridae" "Bovidae" "Bovidae" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   body_mass = col_double(),
##   ..   metabolic_rate = col_double(),
##   ..   family = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Answer:

- body_mass: numeric
- metabolic_rate: numeric
- family: character

**Ex.6)**

```
# type your code here
mass_metabolism2<-mass_metabolism %>% mutate(family=factor(family))
levels(mass_metabolism2$family)
```

```
## [1] "Antilocapridae" "Bovidae"        "Camelidae"      "Canidae"
## [5] "Cervidae"       "Suidae"         "Tayassuidae"    "Tragulidae"
```

```
mass_metab_levels<-levels(mass_metabolism2$family)
```

Answer: The levels for `mass_metabolism2$family` are:

"Antilocapridae" "Bovidae" "Camelidae" "Canidae"
"Cervidae" "Suidae" "Tayassuidae" "Tragulidae"

**Ex.7)**

```
# type your code here
summary(mass_metabolism)
```

```
##    body_mass      metabolic_rate      family
## Min.   :  1613   Min.   :  4.865   Length:25
## 1st Qu.: 21500   1st Qu.: 46.347   Class :character
## Median : 65320   Median :106.670   Mode  :character
## Mean   : 99358   Mean   :112.935
## 3rd Qu.:133300   3rd Qu.:148.949
## Max.   :407000   Max.   :306.770
```

```
summary(mass_metabolism2)
```

```
##    body_mass      metabolic_rate              family
## Min.   :  1613   Min.   :  4.865   Bovidae       :11
## 1st Qu.: 21500   1st Qu.: 46.347   Cervidae      : 5
## Median : 65320   Median :106.670   Antilocapridae: 2
## Mean   : 99358   Mean   :112.935   Camelidae     : 2
## 3rd Qu.:133300   3rd Qu.:148.949   Tragulidae    : 2
## Max.   :407000   Max.   :306.770   Canidae       : 1
##                                    (Other)       : 2
```

Answer: `summary` is meaningless for a character column. Once we converted into factor, `summary` gives us counts within each category. For example, there are 11 rows where `family=='Bovidae'`.

**Ex.8)**

```
# type your code here
table(is.na(mass_metabolism2))
```

```
##
## FALSE
##    75
```

```
#the command above shows there are no missing data
mass_metab_na<-FALSE
```

Answer: there are no NAs.

**Ex.9)**

```
range_mass<-diff(range(mass_metabolism2$body_mass))
range_mass_bovidae<-mass_metabolism %>% filter(family=="Bovidae") %>%
↪    summarise(range=diff(range(body_mass)))
```
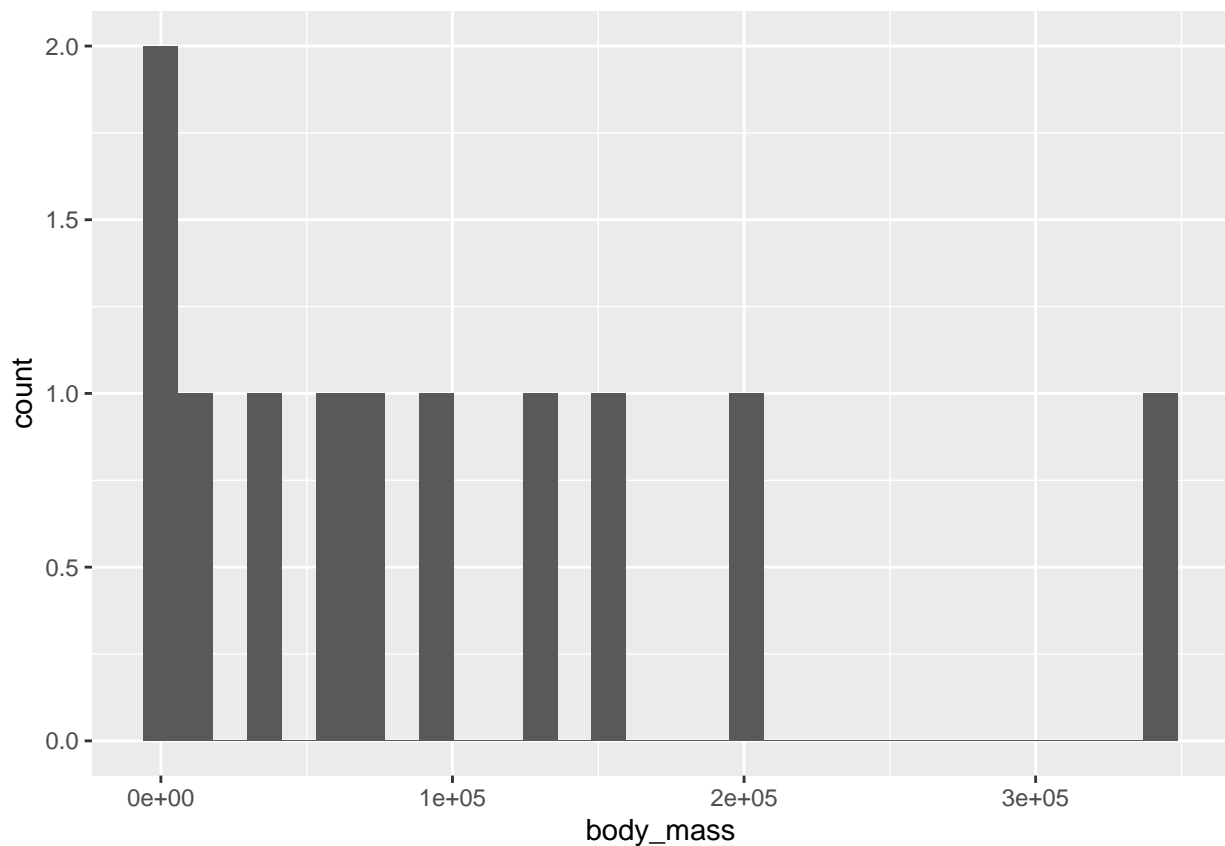
Answer: the range in mass overall is pretty huge (405387 kg!). But the range within Bovidae alone is also very impressive: 342800, so yes, I would say yes.

**Ex.10)**

```
mass_metabolism2 %>% filter(family=="Bovidae") %>% ggplot(aes(x=body_mass)) +
↪    geom_histogram()
```

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



Answer: skewed right.

**Ex.11)**

4

```
range_metab<-mass_metabolism %>% summarise(range=diff(range(metabolic_rate)))
range_metab_bovidae<-mass_metabolism %>% filter(family=="Bovidae") %>%
↪   summarise(range=diff(range(metabolic_rate)))
range_metab
```

```
## # A tibble: 1 x 1
##    range
##    <dbl>
## 1  302.
```
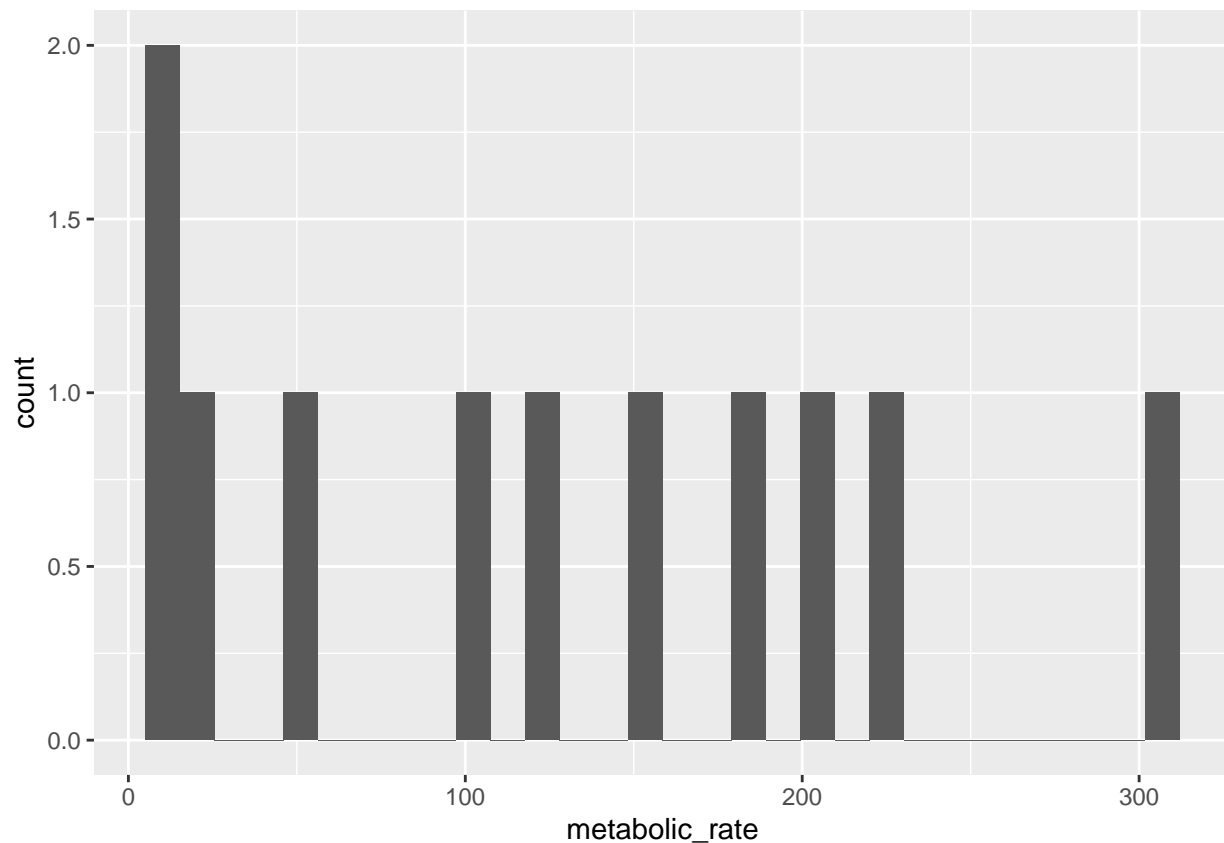
```
range_metab_bovidae
```

```
## # A tibble: 1 x 1
##    range
##    <dbl>
## 1  297.
```

Answer: Yes, the range in Bovidae alone (297) is almost as high as the overall range (302).

**Ex.12)**

```
mass_metabolism2 %>% filter(family=="Bovidae") %>% ggplot(aes(x=metabolic_rate)) +
↪   geom_histogram()
```
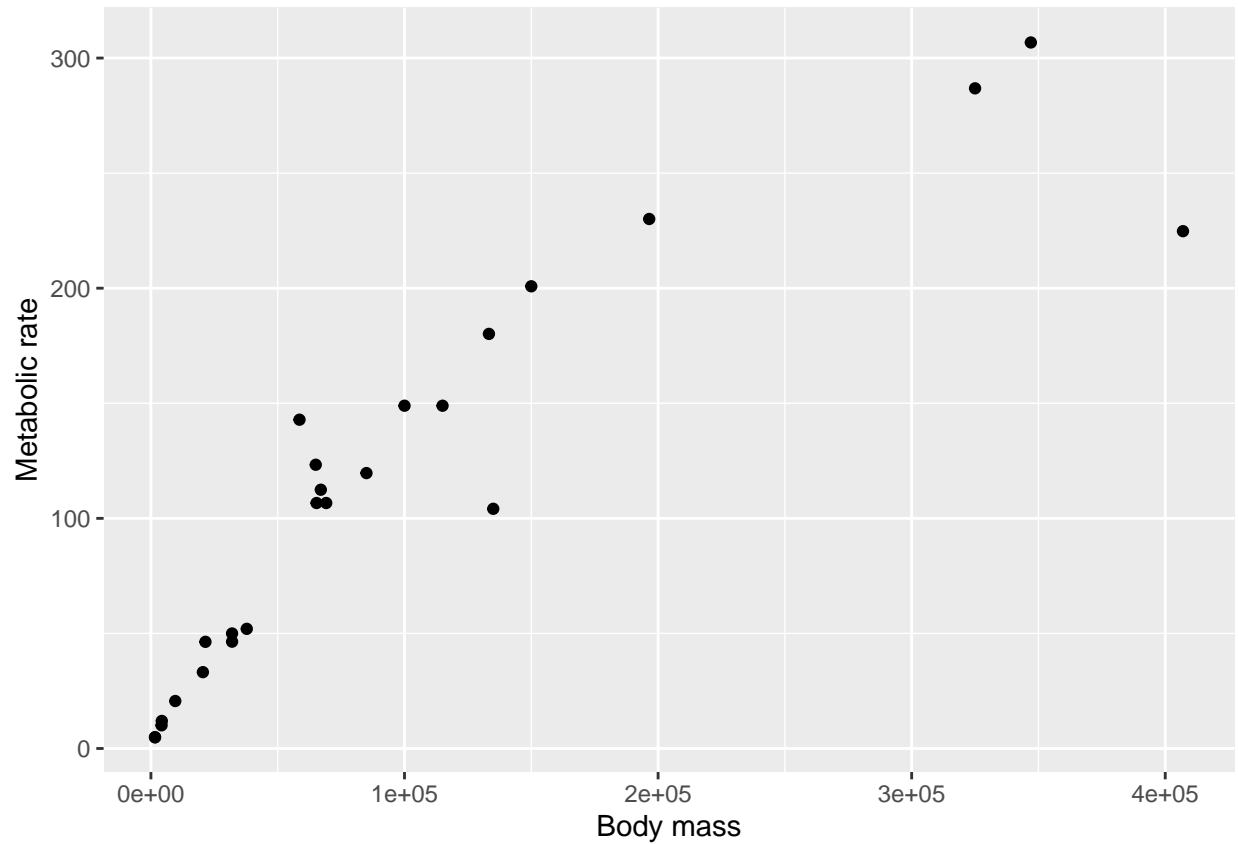
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

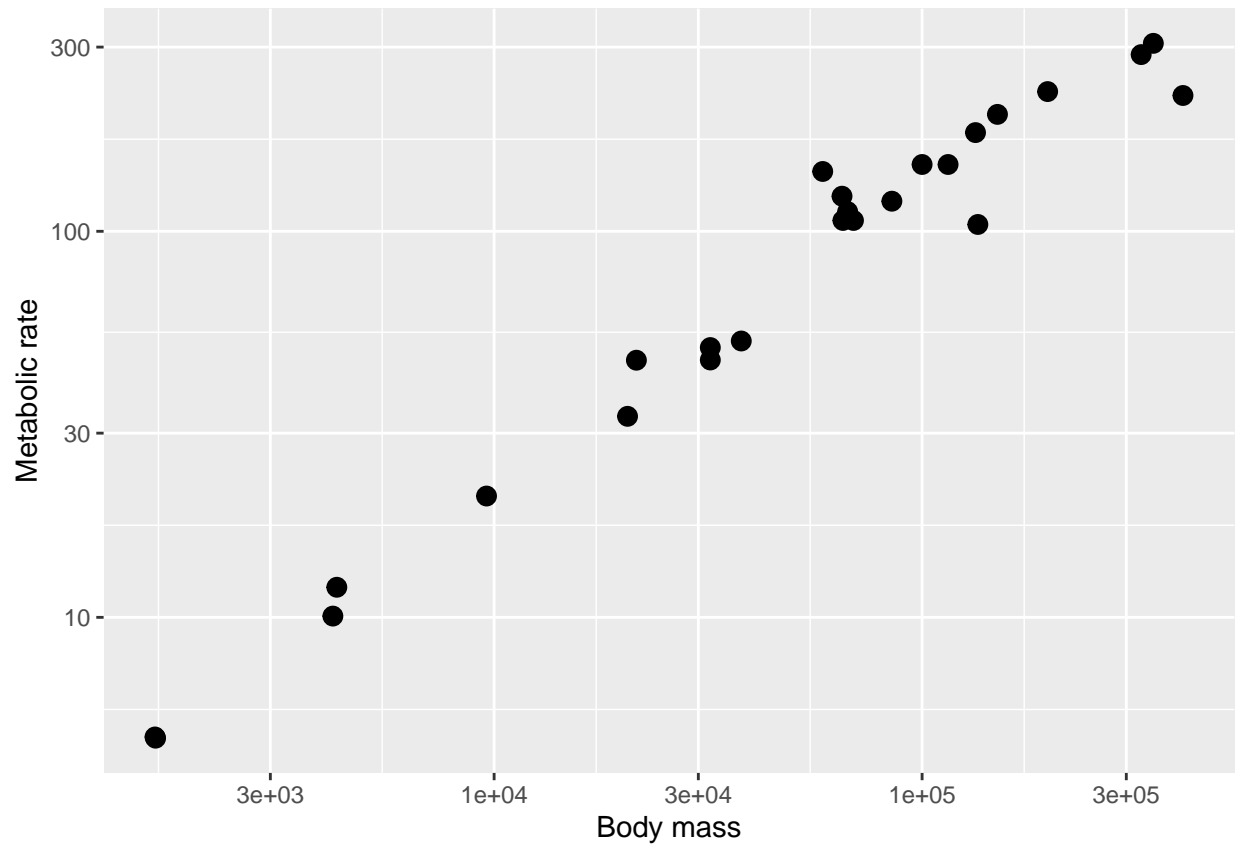Answer: skewed right

**Ex.13)**

```
ggplot(mass_metabolism2, aes(x=body_mass, y=metabolic_rate)) + geom_point() +
↪   ylab("Metabolic rate") + xlab("Body mass")
```



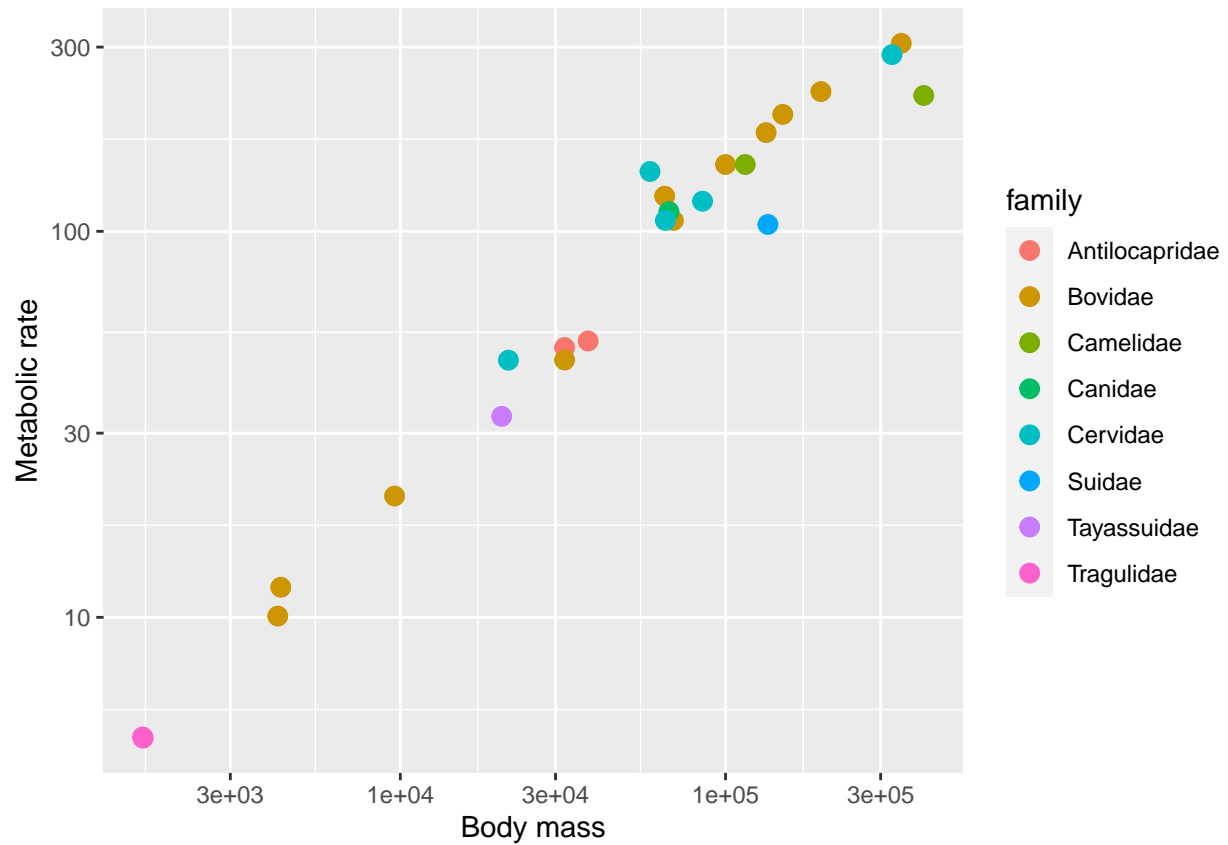Answer: body mass (explanatory), metabolic rate (response).

**Ex.14)**

```
ggplot(mass_metabolism2, aes(x=body_mass, y=metabolic_rate)) + geom_point(size=3) +
↪   ylab("Metabolic rate") + xlab("Body mass") + scale_y_continuous(trans='log10') +
↪   scale_x_continuous(trans='log10')
```
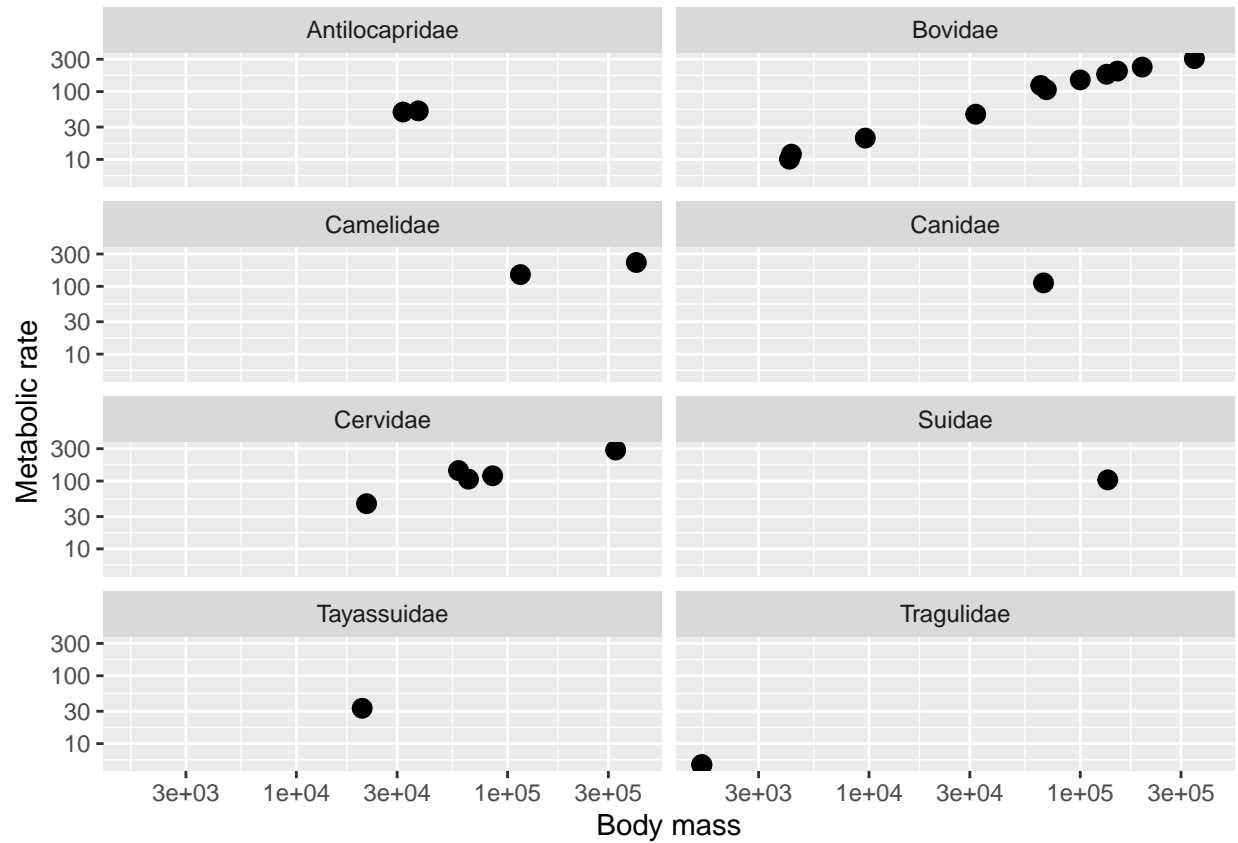
**Ex.15)**

```
ggplot(mass_metabolism2, aes(x=body_mass, y=metabolic_rate, color=family)) +
↪  geom_point(size=3) + ylab("Metabolic rate") + xlab("Body mass") +
↪  scale_y_continuous(trans='log10') + scale_x_continuous(trans='log10')
```

**Ex.16)**

```
#type your code here
ggplot(mass_metabolism2, aes(x=body_mass, y=metabolic_rate)) + geom_point(size=3) +
↪    ylab("Metabolic rate") + xlab("Body mass") + scale_y_continuous(trans='log10') +
↪    scale_x_continuous(trans='log10') + facet_wrap(~family, ncol=2)
```

```
ggplot(mass_metabolism2, aes(x=body_mass, y=metabolic_rate)) + geom_point(size=3) +
↪  ylab("Metabolic rate") + xlab("Body mass") + scale_y_continuous(trans='log10') +
↪  scale_x_continuous(trans='log10') + facet_wrap(~family, ncol=2)
```
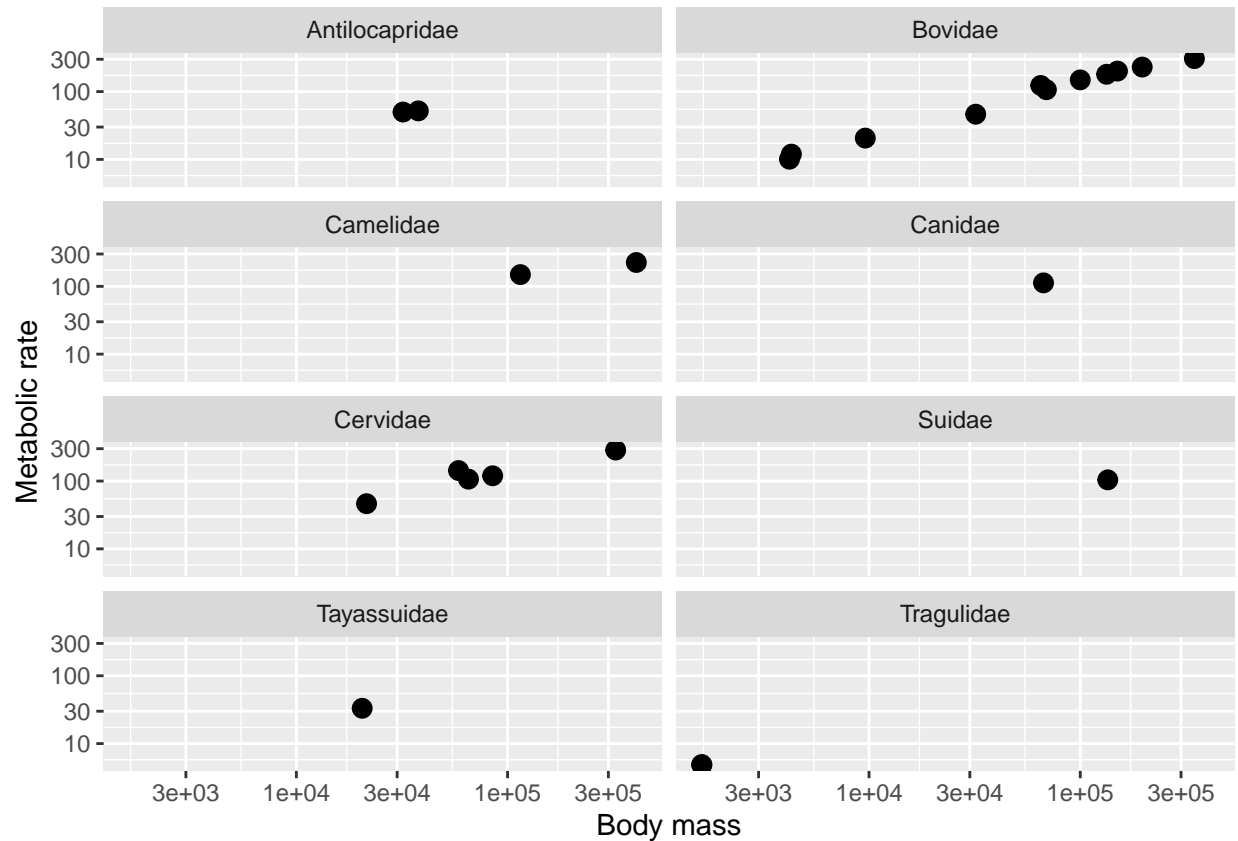
**Ex.17)**

```
body_mass_summary<-mass_metabolism2 %>% group_by(family) %>% summarise(
  N=n(),
  Range_body_mass=diff(range(body_mass))
)
body_mass_summary
```

```
## # A tibble: 8 x 3
##   family           N Range_body_mass
##   <fct>        <int>           <dbl>
## 1 Antilocapridae   2            5800
## 2 Bovidae         11          342800
## 3 Camelidae        2          292000
## 4 Canidae          1               0
## 5 Cervidae         5          303500
## 6 Suidae           1               0
## 7 Tayassuidae      1               0
## 8 Tragulidae       2               5
```

Answer: because there is only one value so range cannot really be calculated.

---

## Part 2: Carbon storage in shrubs

**Ex.18)**

```r
# type your code here
shrubs<-read_csv("shrub-volume-data.csv")
```

```
## Rows: 12 Columns: 5
## -- Column specification ----------------------------------------------------
## Delimiter: ","
## dbl (5): site, experiment, length, width, height
##
## 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.
```

```r
head(shrubs)
```

```
## # A tibble: 6 x 5
##    site experiment length width height
##   <dbl>      <dbl>  <dbl> <dbl>  <dbl>
## 1     1          1    2.2   1.3    9.6
## 2     1          2    2.1   2.2    7.6
## 3     1          3    2.7   1.5    2.2
## 4     2          1    3     4.5    1.5
## 5     2          2    3.1   3.1    4
## 6     2          3    2.5   2.8    3
```

```r
dim(shrubs)
```

```
## [1] 12  5
```

```r
str(shrubs)
```

```
## spc_tbl_ [12 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ site      : num [1:12] 1 1 1 2 2 2 3 3 3 4 ...
##  $ experiment: num [1:12] 1 2 3 1 2 3 1 2 3 1 ...
##  $ length    : num [1:12] 2.2 2.1 2.7 3 3.1 2.5 1.9 1.1 3.5 2.9 ...
##  $ width     : num [1:12] 1.3 2.2 1.5 4.5 3.1 2.8 1.8 0.5 2 2.7 ...
##  $ height    : num [1:12] 9.6 7.6 2.2 1.5 4 3 4.5 2.3 7.5 3.2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   site = col_double(),
##   ..   experiment = col_double(),
##   ..   length = col_double(),
##   ..   width = col_double(),
##   ..   height = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
shrubs2<-shrubs %>% mutate(site=factor(site), experiment=factor(experiment))
summary(shrubs)
```

```
##       site          experiment      length          width           height
## Min.   :1.00   Min.   :1      Min.   :1.100   Min.   :0.500   Min.   :1.50
## 1st Qu.:1.75   1st Qu.:1      1st Qu.:2.050   1st Qu.:1.725   1st Qu.:2.60
## Median :2.50   Median :2      Median :2.600   Median :2.100   Median :3.60
## Mean   :2.50   Mean   :2      Mean   :2.558   Mean   :2.417   Mean   :4.55
## 3rd Qu.:3.25   3rd Qu.:3      3rd Qu.:3.025   3rd Qu.:2.875   3rd Qu.:6.75
## Max.   :4.00   Max.   :3      Max.   :4.500   Max.   :4.800   Max.   :9.60
```

```
summary(shrubs2)
```

```
## site   experiment      length          width           height
## 1:3    1:4        Min.   :1.100   Min.   :0.500   Min.   :1.50
## 2:3    2:4        1st Qu.:2.050   1st Qu.:1.725   1st Qu.:2.60
## 3:3    3:4        Median :2.600   Median :2.100   Median :3.60
## 4:3               Mean   :2.558   Mean   :2.417   Mean   :4.55
##                   3rd Qu.:3.025   3rd Qu.:2.875   3rd Qu.:6.75
##                   Max.   :4.500   Max.   :4.800   Max.   :9.60
```
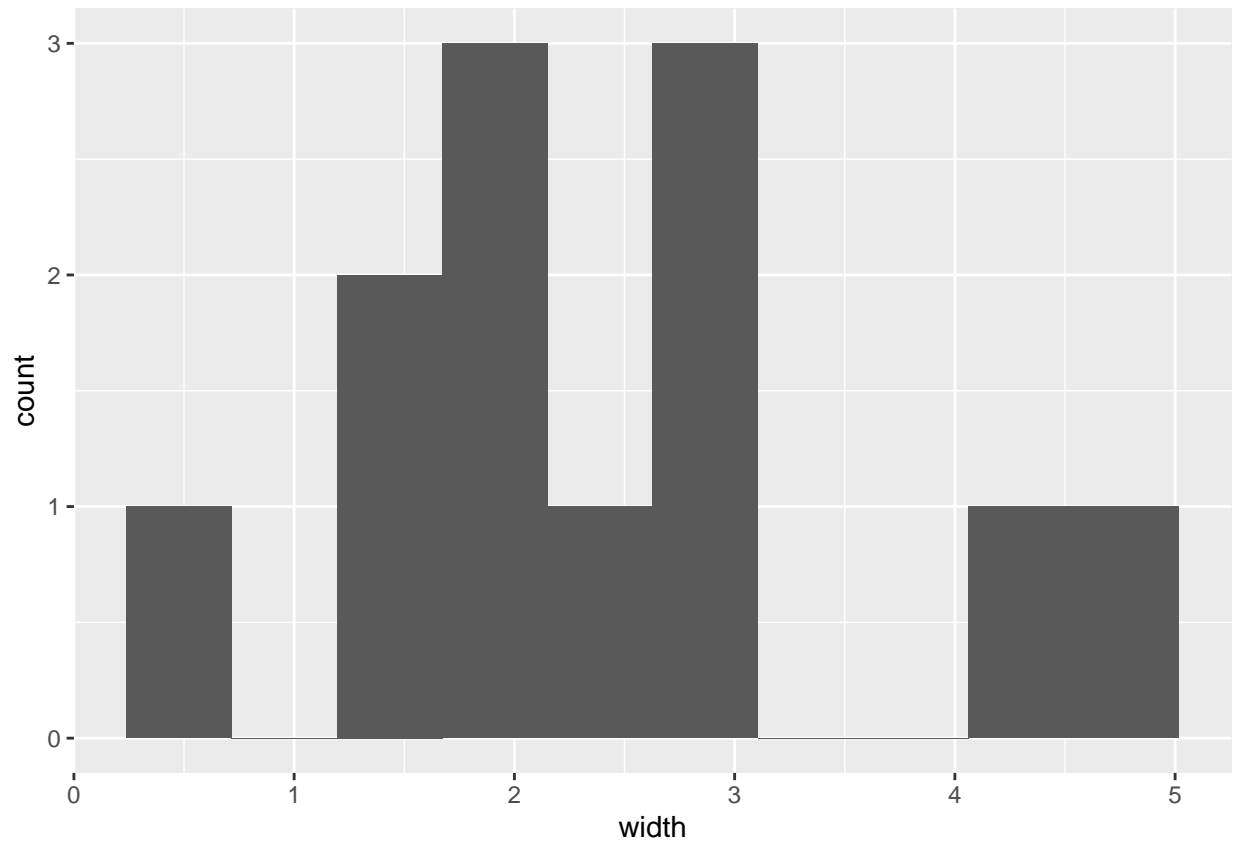
Answer: site and experiment were coded as `numeric` which does make sense since they are categorical variables. We change them to factors instead.
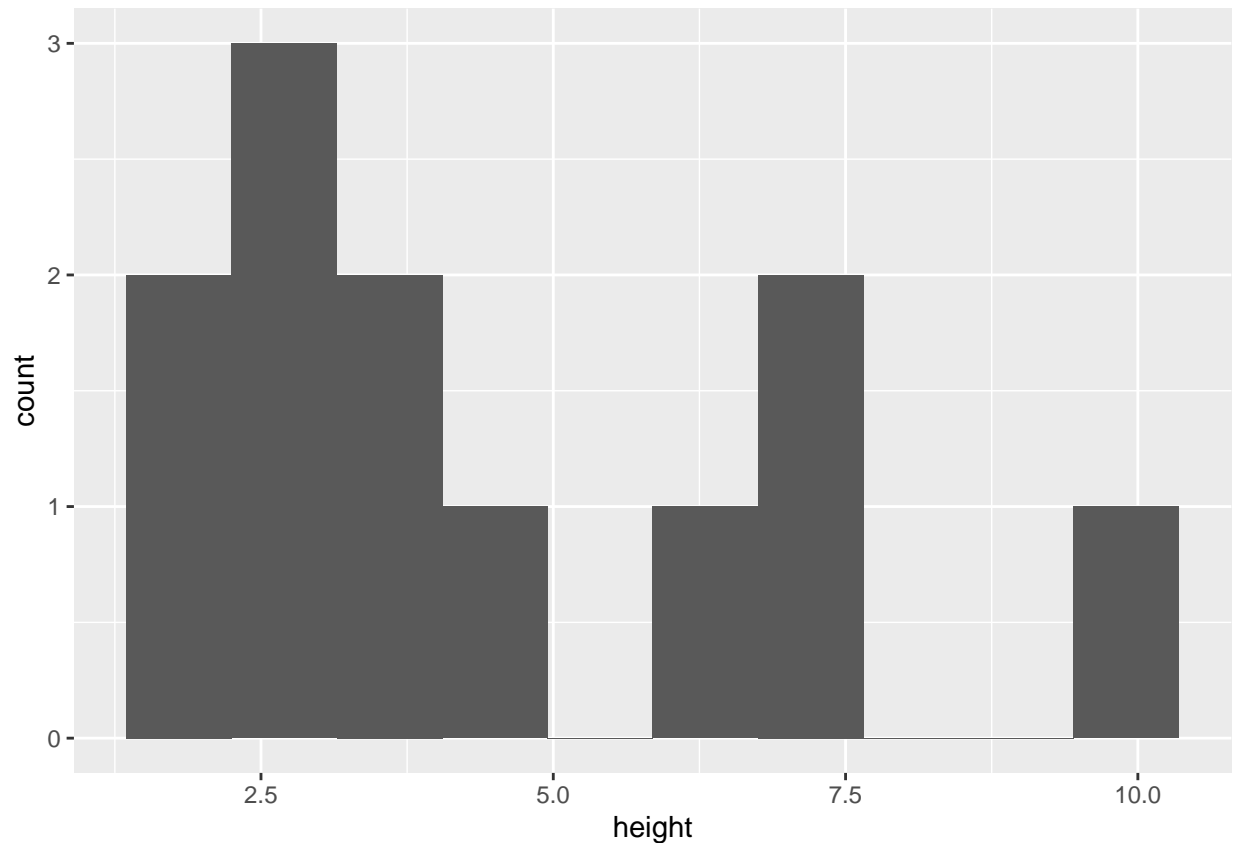
**Ex.19)**

```
#create shrubs3
shrubs3<-shrubs2 %>% na.omit()
#length
shrubs3 %>% ggplot(aes(x=length)) + geom_histogram(bins=10)
```
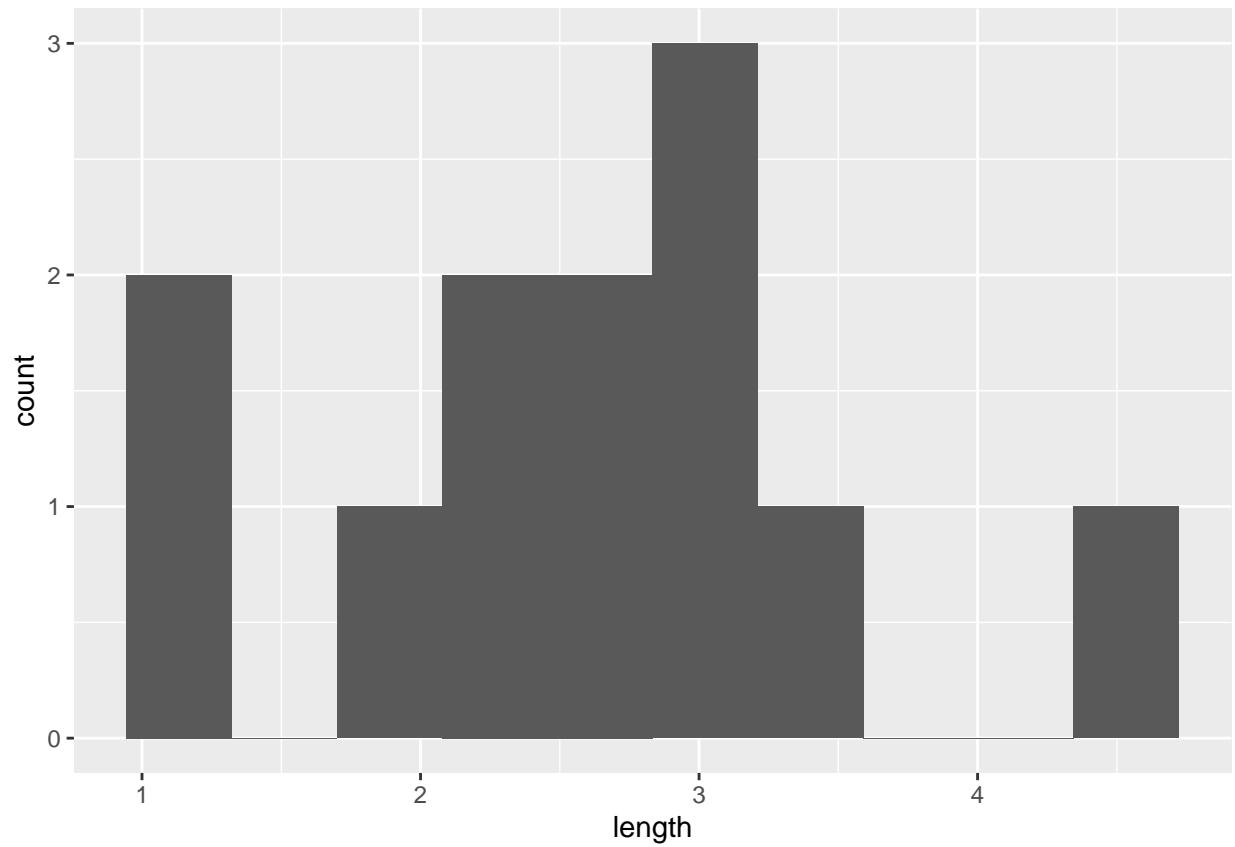
```
#width
shrubs3 %>% ggplot(aes(x=width)) + geom_histogram(bins=10)
```

```
# height
shrubs3 %>% ggplot(aes(x=height)) + geom_histogram(bins=10)
```

**Ex.20)**

```
shrubs3_summ<-shrubs3 %>% group_by(experiment) %>% summarise(N=n(),
→  Range_length=diff(range(length)), Range_width=diff(range(width),
→  Range_height=diff(range(height))))
shrubs3_summ
```

```
## # A tibble: 3 x 4
##   experiment     N Range_length Range_width
##   <fct>      <int>        <dbl>       <dbl>
## 1 1              4          1.1         3.2
## 2 2              4          3.4         4.3
## 3 3              4          2.3         1.3
```

**Ex.21)**

```
#type your code here
shrubs4<-shrubs3 %>% mutate(area=width * length) %>% filter(height > 5) %>%
→  arrange(length)
shrubs4
```

```
## # A tibble: 4 x 6
##   site  experiment length width height  area
##   <fct> <fct>       <dbl> <dbl>  <dbl> <dbl>
```

```
## 1 1      2              2.1   2.2    7.6  4.62
## 2 1      1              2.2   1.3    9.6  2.86
## 3 3      3              3.5   2      7.5  7
## 4 4      2              4.5   4.8    6.5 21.6
```

**Ex.22)**

```
#type your code here
shrubs5<-shrubs3 %>% filter(height>4 & width>2)
shrubs5
```

```
## # A tibble: 2 x 5
##    site  experiment length width height
##    <fct> <fct>       <dbl> <dbl>  <dbl>
## 1 1      2             2.1   2.2    7.6
## 2 4      2             4.5   4.8    6.5
```

**Ex.23)**

```
#type your code here

shrubs6<-shrubs3 %>% filter(experiment ==1 | experiment == 2)

shrubs6
```

```
## # A tibble: 8 x 5
##    site  experiment length width height
##    <fct> <fct>       <dbl> <dbl>  <dbl>
## 1 1      1             2.2   1.3    9.6
## 2 1      2             2.1   2.2    7.6
## 3 2      1             3     4.5    1.5
## 4 2      2             3.1   3.1    4
## 5 3      1             1.9   1.8    4.5
## 6 3      2             1.1   0.5    2.3
## 7 4      1             2.9   2.7    3.2
## 8 4      2             4.5   4.8    6.5
```

**Ex.24)**

```
shrubs %>% filter(!is.na(height))
```

```
## # A tibble: 12 x 5
##     site experiment length width height
##    <dbl>      <dbl>  <dbl> <dbl>  <dbl>
## 1    1          1     2.2   1.3    9.6
## 2    1          2     2.1   2.2    7.6
## 3    1          3     2.7   1.5    2.2
## 4    2          1     3     4.5    1.5
## 5    2          2     3.1   3.1    4
## 6    2          3     2.5   2.8    3
## 7    3          1     1.9   1.8    4.5
```

```
## 8       3           2   1.1   0.5    2.3
## 9       3           3   3.5   2      7.5
## 10      4           1   2.9   2.7    3.2
## 11      4           2   4.5   4.8    6.5
## 12      4           3   1.2   1.8    2.7
```

**Ex.25)**

```
shrub_volumes<-shrubs %>% mutate(area=width*length, volume=length*width*height)
shrub_volumes
```

```
## # A tibble: 12 x 7
##      site experiment length width height  area volume
##     <dbl>      <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl>
## 1      1          1    2.2   1.3    9.6  2.86   27.5
## 2      1          2    2.1   2.2    7.6  4.62   35.1
## 3      1          3    2.7   1.5    2.2  4.05    8.91
## 4      2          1    3     4.5    1.5 13.5    20.2
## 5      2          2    3.1   3.1    4    9.61   38.4
## 6      2          3    2.5   2.8    3    7      21
## 7      3          1    1.9   1.8    4.5  3.42   15.4
## 8      3          2    1.1   0.5    2.3  0.55    1.26
## 9      3          3    3.5   2      7.5  7      52.5
## 10     4          1    2.9   2.7    3.2  7.83   25.1
## 11     4          2    4.5   4.8    6.5 21.6   140.
## 12     4          3    1.2   1.8    2.7  2.16    5.83
```

**Ex.26)**

```
write_csv(shrub_volumes, file="shrub-volumes.csv")
```