CHAPITRE 6 - DATA VISUALIZATION

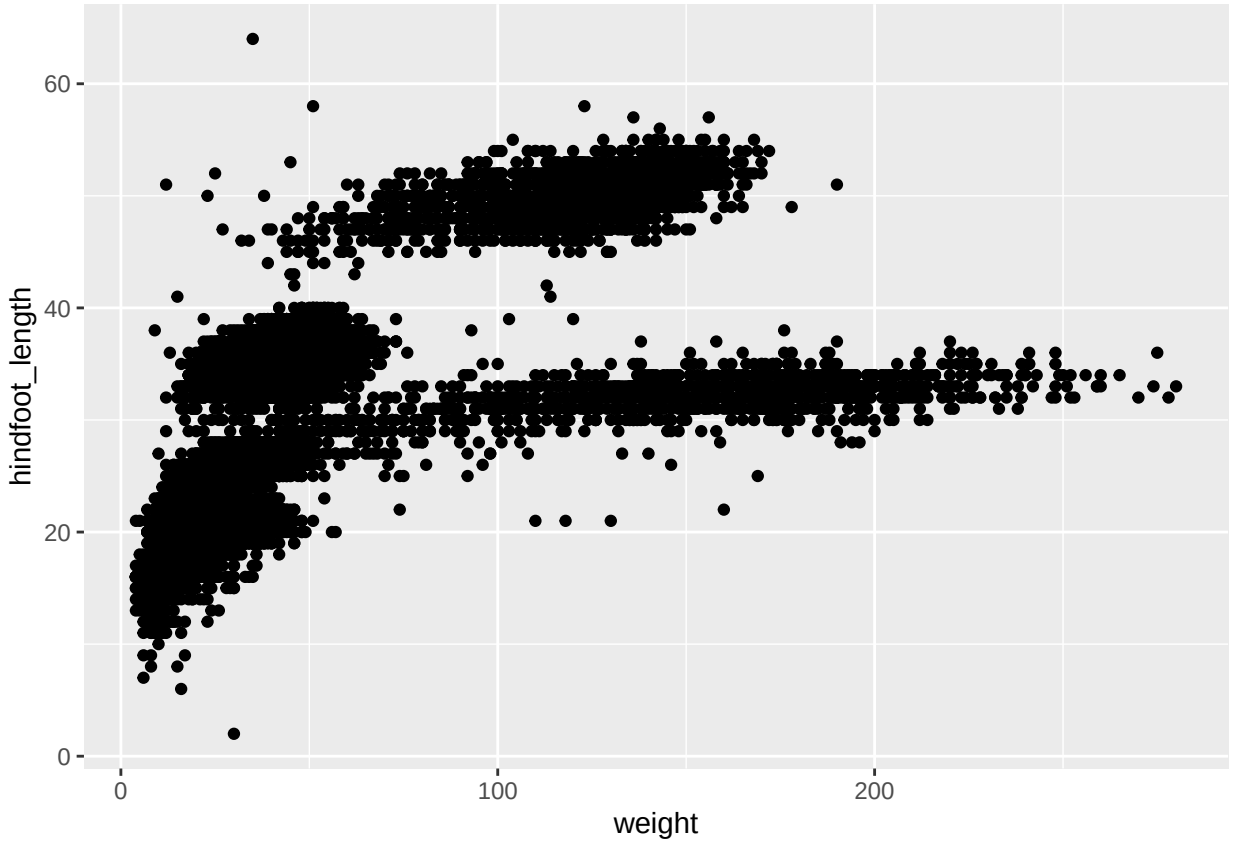
Armelle le Court

19/05/2021

```
library("tidyverse") #ggplot2 is in tidyverse
surveys_complete <- read_csv("data_output/surveys_complete.csv")
library("hexbin")
```

1 : PLOTTING WITH GGPLOT2

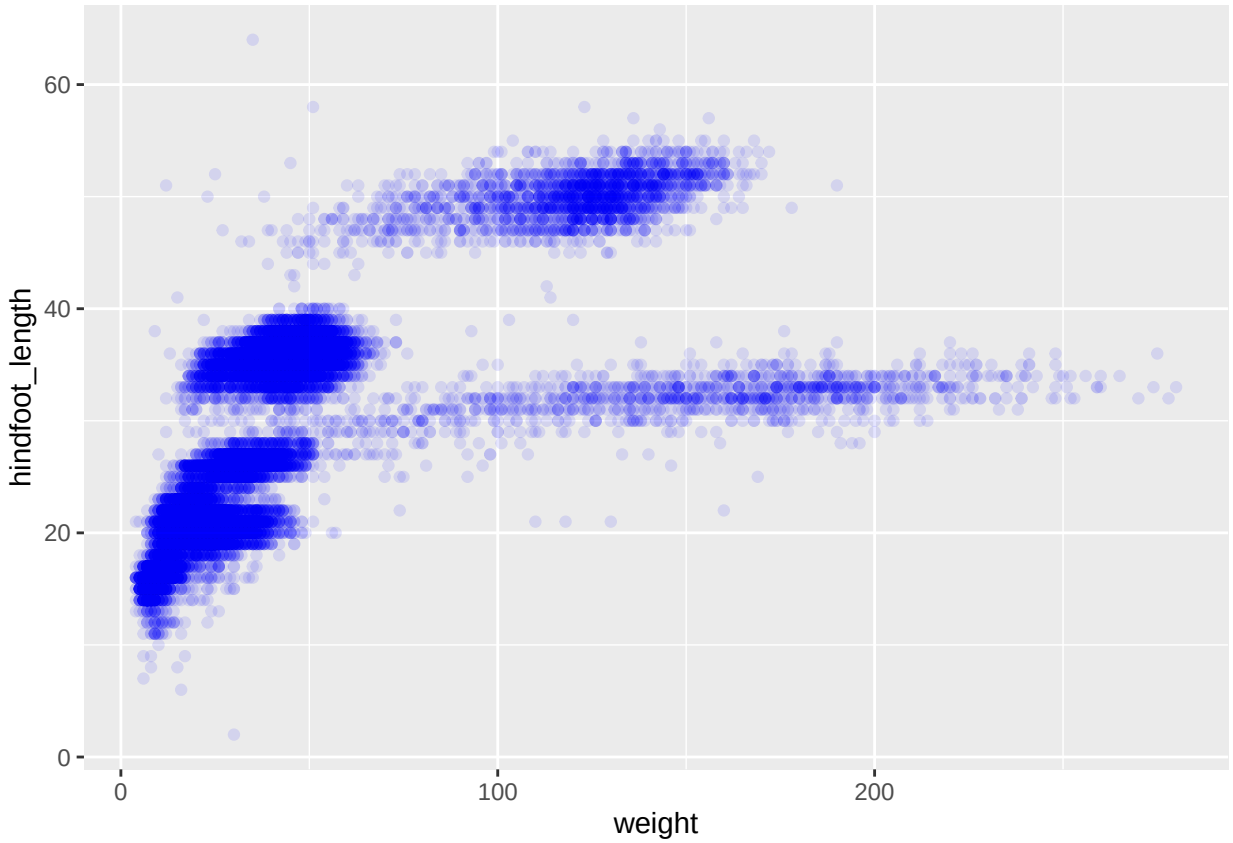
```
#ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +  <GEOM_FUNCTION>() is how we construct
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length)) +
  geom_point()
```



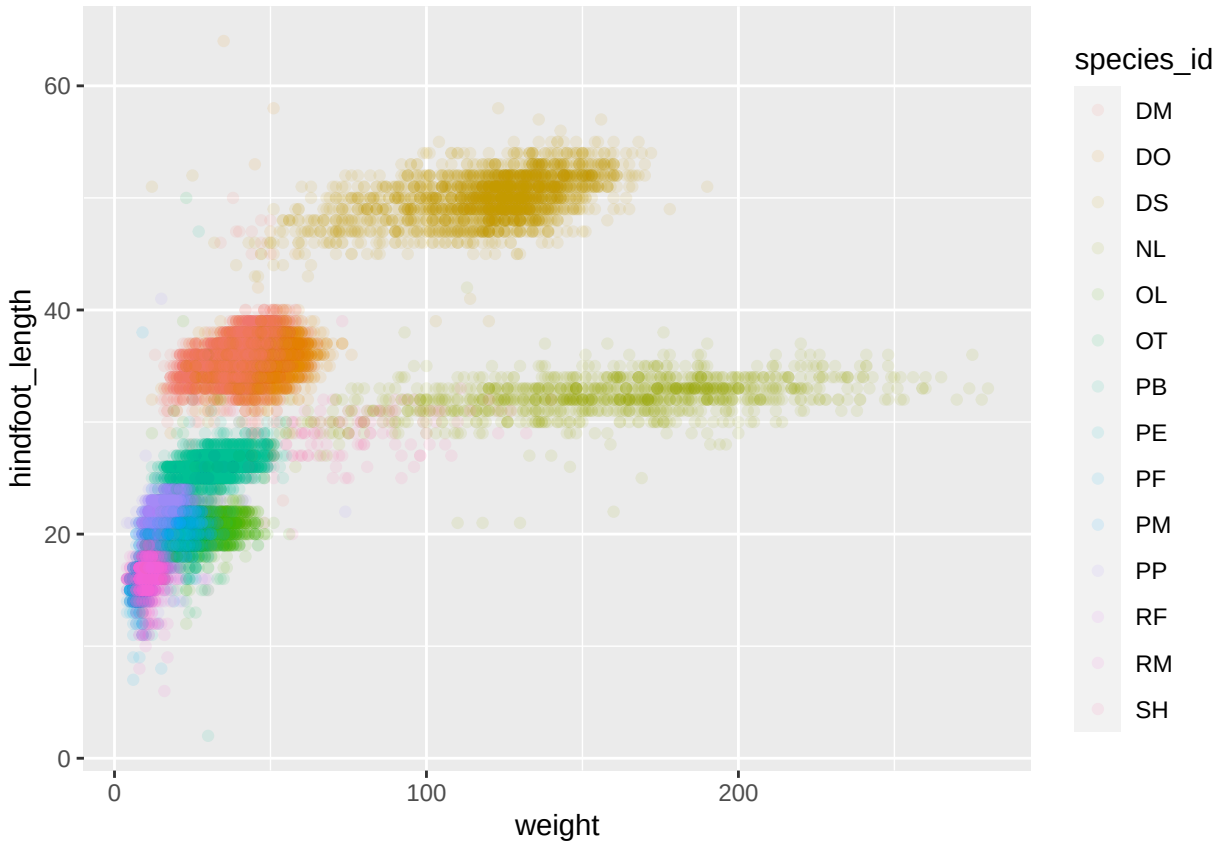

```
#geom_point for dots, geom_boxplot for boxplot, and geom_line for trend lines
```

2 : BUILDING YOUR PLOTS ITERATIVELY

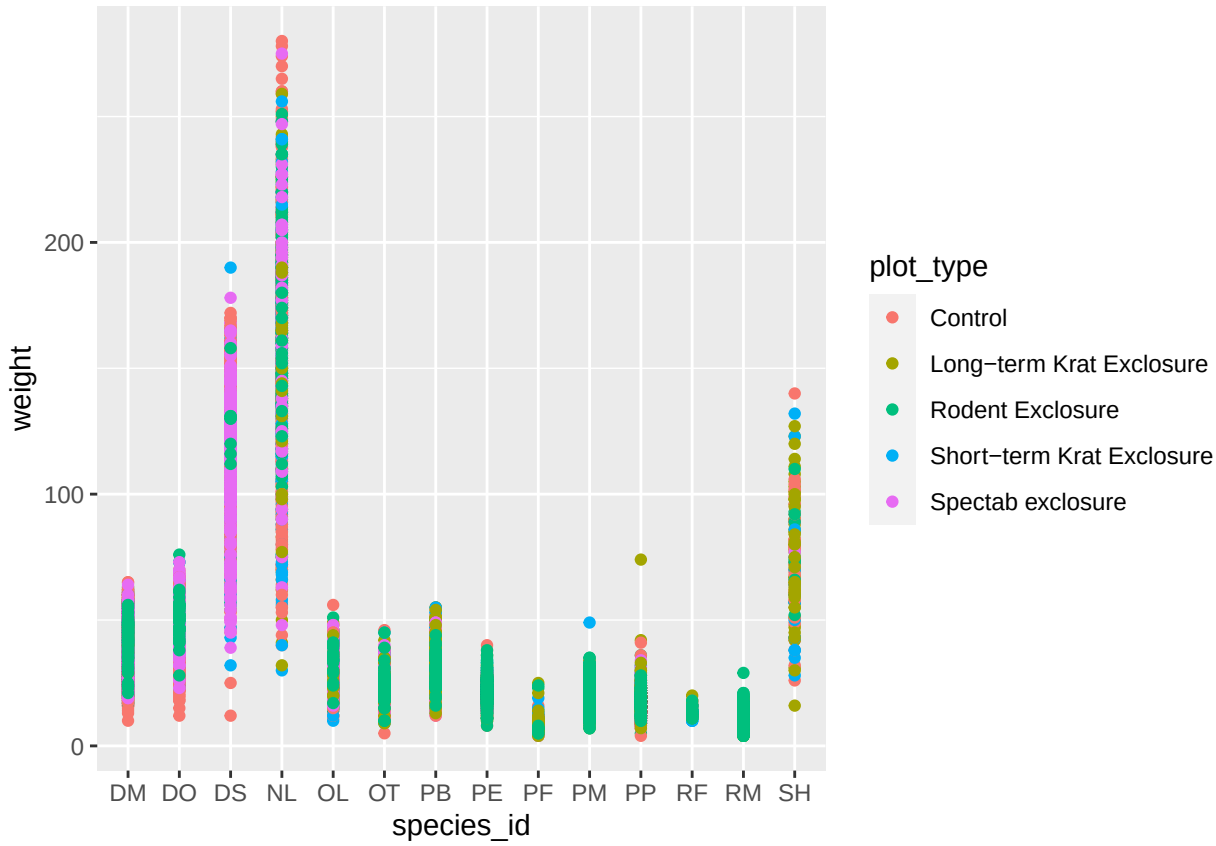
```
#we cand add "alpha" for transparency and colors with "color".
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length)) +
  geom_point(alpha = 0.1, color = "blue")
```




```
#to color each species with different colors
ggplot(data = surveys_complete, mapping = aes(x = weight, y = hindfoot_length)) +
  geom_point(alpha = 0.1, aes(color = species_id))
```



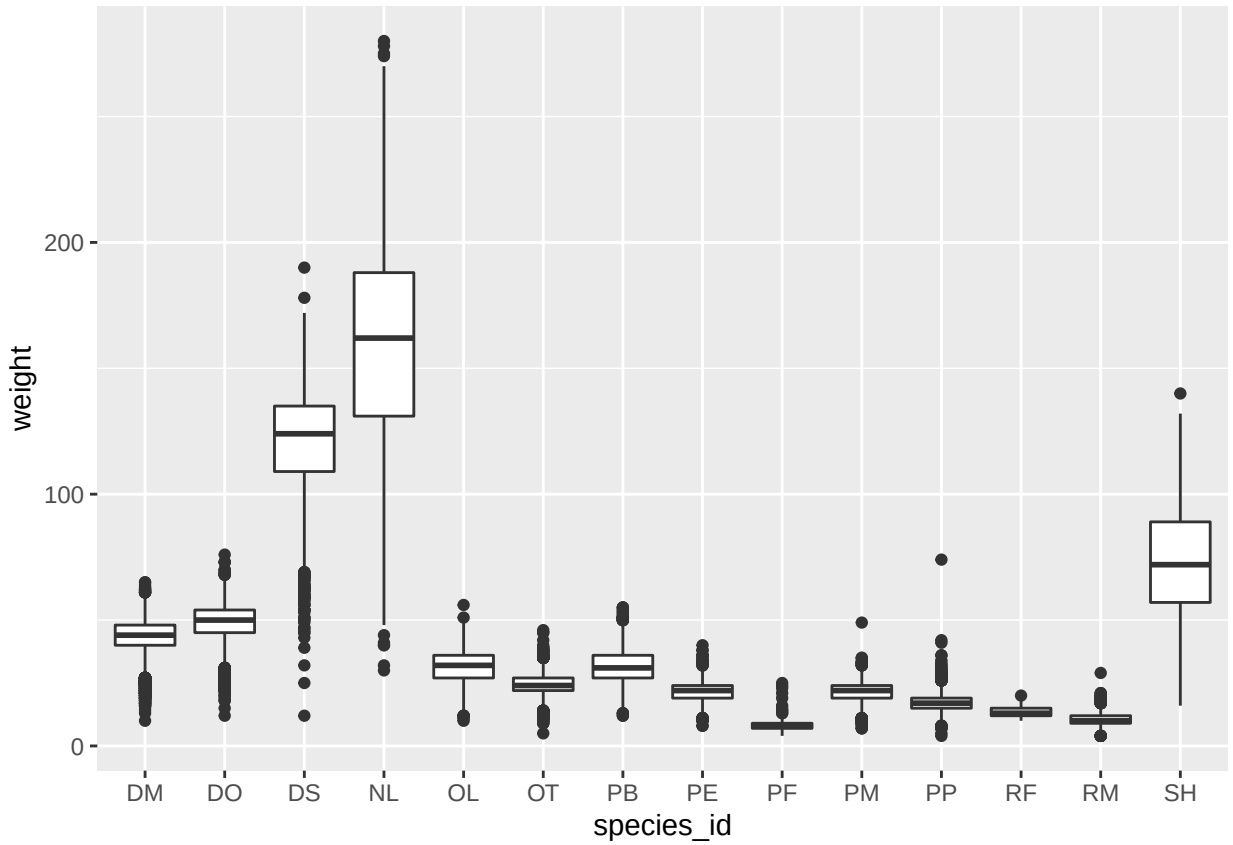

```
#we can specify colors in the mapping directly also

#QUESTION
#Use what you just learned to 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?
ggplot(data = surveys_complete, mapping = aes(x = species_id, y = weight)) +
  geom_point(aes(color = plot_type))
```



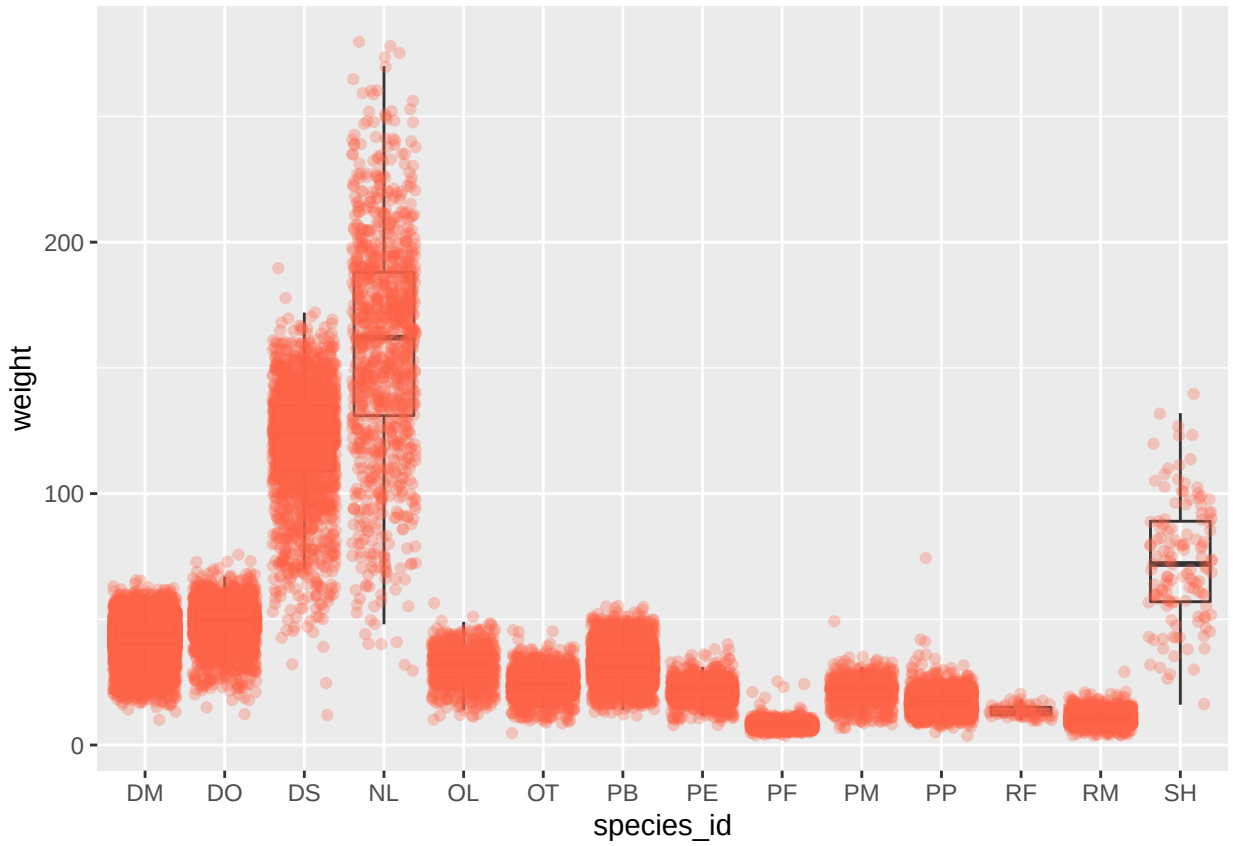
3 : BOXPLOT

to visualize distributions

```
ggplot(data = surveys_complete, mapping = aes(x = species_id, y = weight)) +
    geom_boxplot()
```




```
#adding points can give a better idea of the nmbr of measurement and their distribution

ggplot(data = surveys_complete, mapping = aes(x = species_id, y = weight)) +
  geom_boxplot(alpha = 0) +
  geom_jitter(alpha = 0.3, color = "tomato")
```

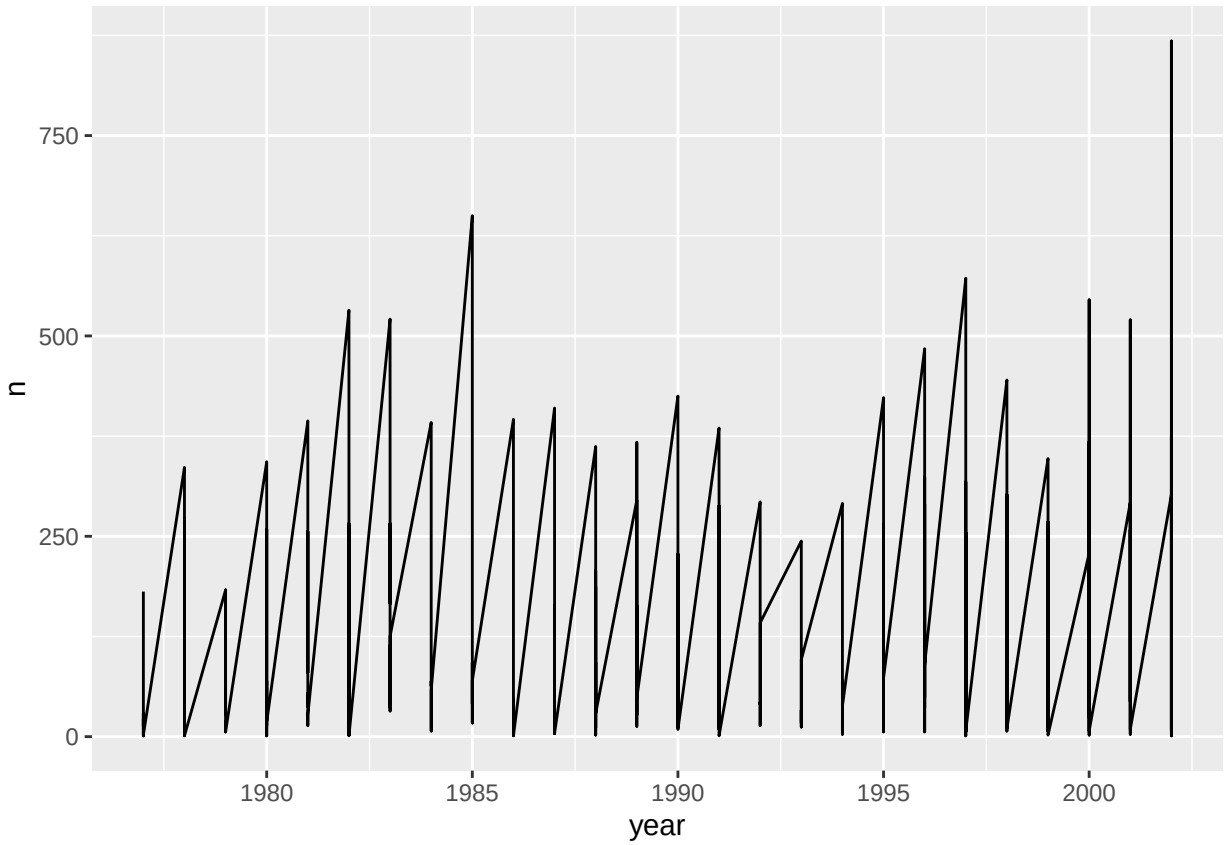


4 : PLOTTING TIME SERIES DATA

```
yearly_counts <- surveys_complete %>%
  count(year, species_id)
yearly_counts
```

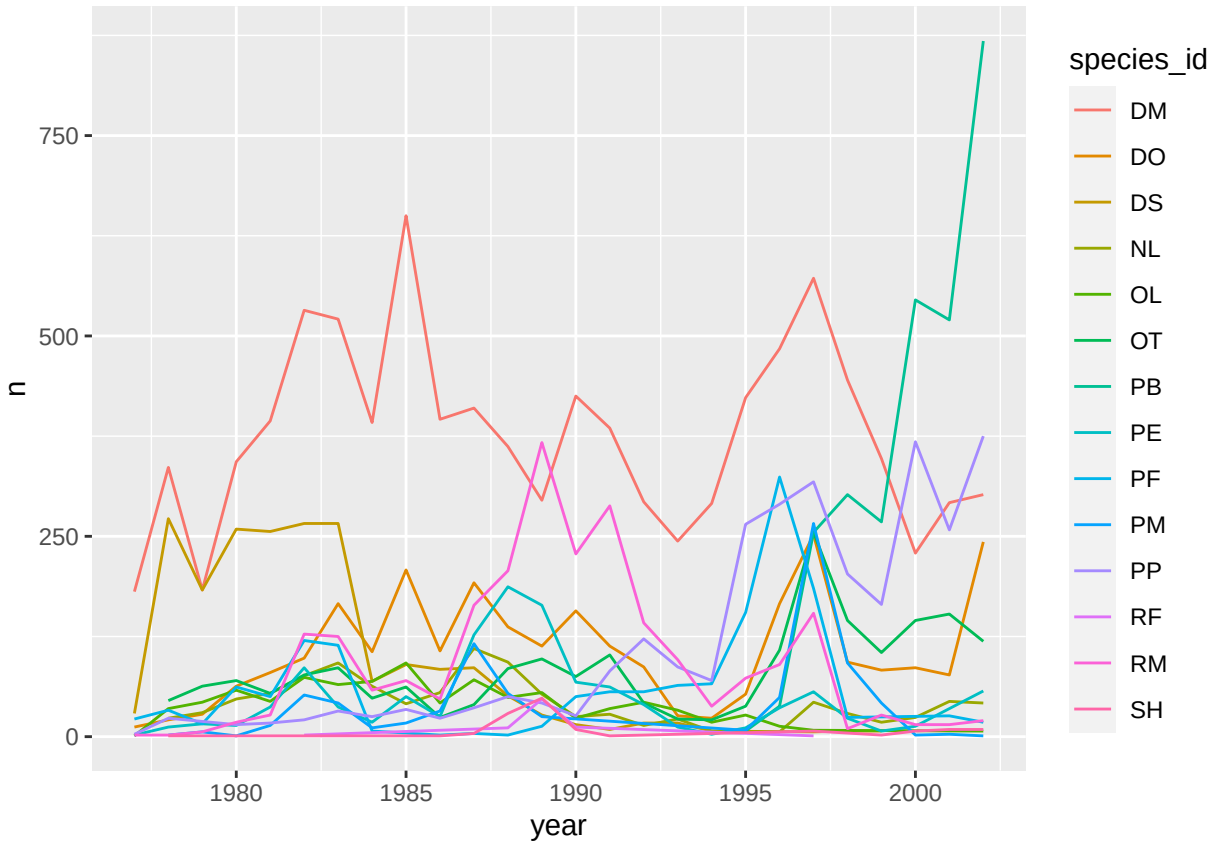
```
## # A tibble: 292 x 3
##     year species_id     n
##    <dbl> <chr>      <int>
##  1  1977 DM           181
##  2  1977 DO            12
##  3  1977 DS            29
##  4  1977 OL             1
##  5  1977 PE             2
##  6  1977 PF            22
##  7  1977 PP             3
##  8  1977 RM             2
##  9  1978 DM           336
## 10  1978 DO            21
## # ... with 282 more rows
```

```
#visualize time series as a line plot
ggplot(data = yearly_counts, mapping = aes(x = year, y = n)) +
     geom_line()
```




```
#its doesnt work bc we plotted for all the species together
#we need to draw a line for each species
ggplot(data = yearly_counts, mapping = aes(x = year, y = n, group = species_id)) +
    geom_line()
```

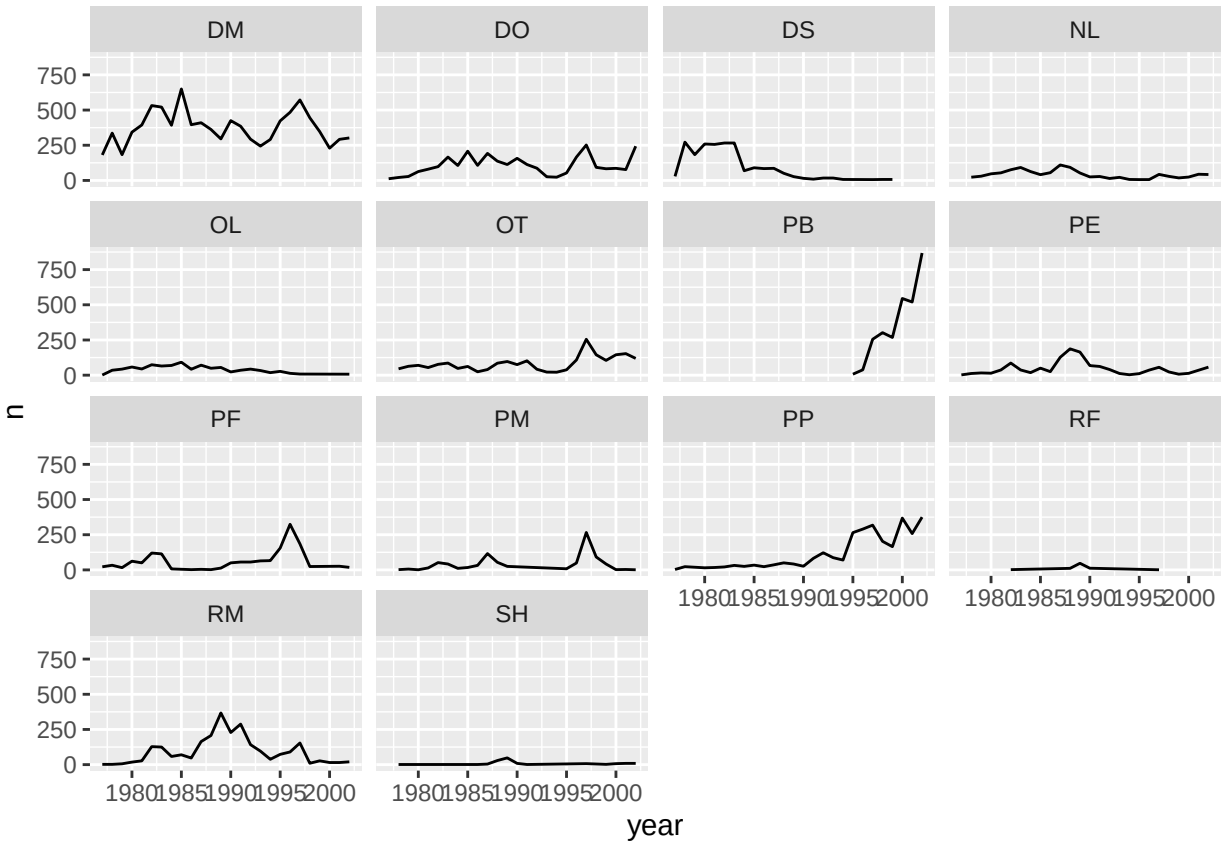

```
#for better visu, we can add colors
ggplot(data = yearly_counts, mapping = aes(x = year, y = n, color = species_id)) +
    geom_line()
```



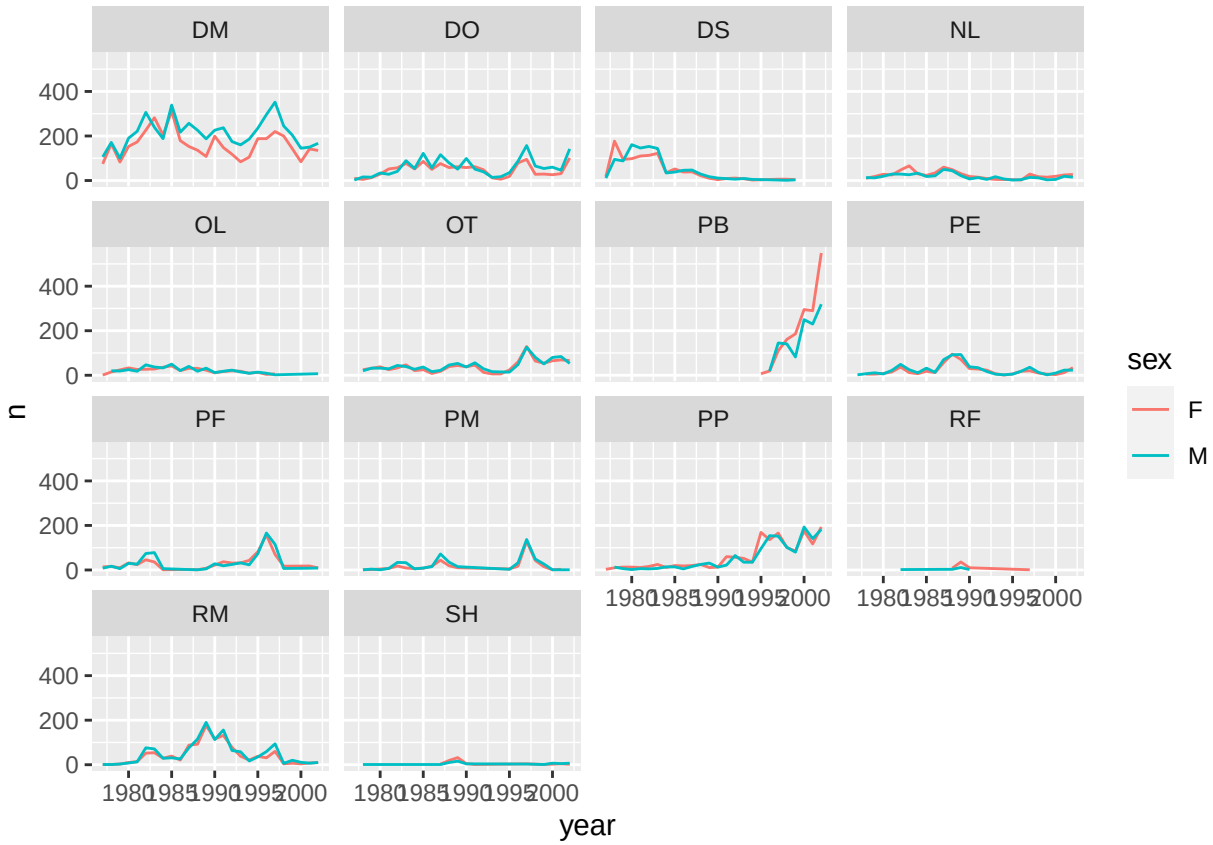
5 : FACETING

to split one plot into multiple plots

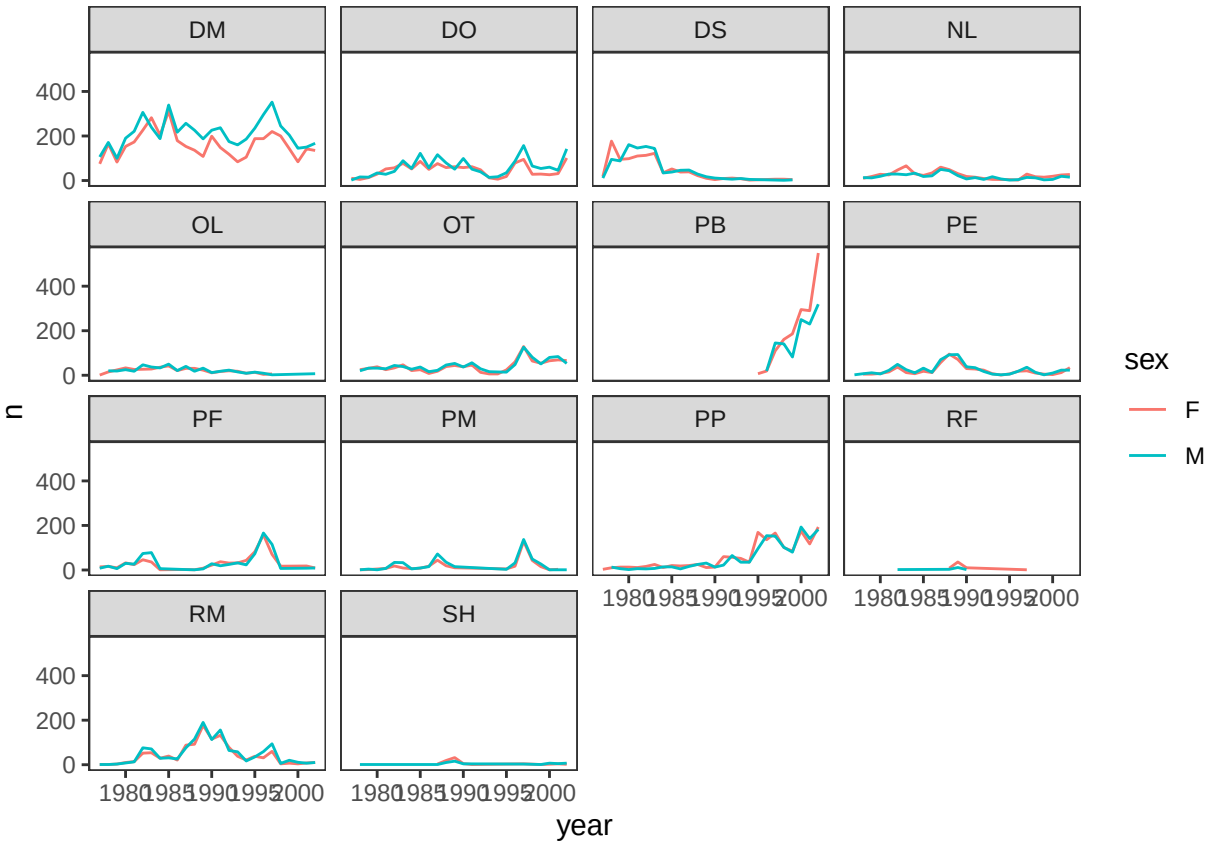
```
# to make a time series plot for each species
ggplot(data = yearly_counts, mapping = aes(x = year, y = n)) +
    geom_line() +
    facet_wrap(~ species_id)
```




```
# to split each line into the 2 sex
#for that we need to make counts in the data frame grouped by year, species_id, and sex:
yearly_sex_counts <- surveys_complete %>%
  count(year, species_id, sex)
#and now we can facet what we want
ggplot(data = yearly_sex_counts, mapping = aes(x = year, y = n, color = sex)) +
  geom_line() +
  facet_wrap(~ species_id)
```




```
#We can set the background to white using the function theme_bw() + remove the grid
ggplot(data = yearly_sex_counts, mapping = aes(x = year, y = n, color = sex)) +
  geom_line() +
  facet_wrap(~ species_id) +
  theme_bw() +
  theme(panel.grid = element_blank())
```



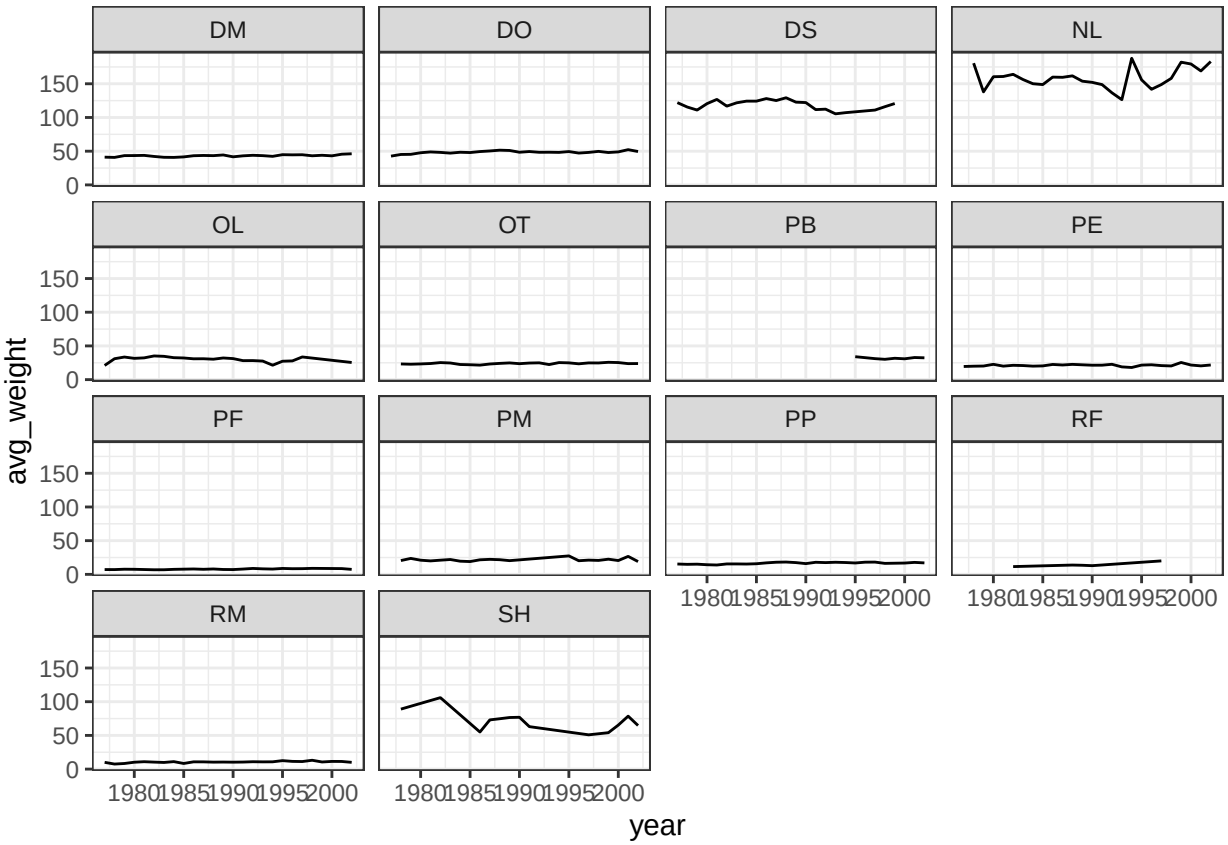
6 : GGPLOT2 THEMES

###QUESION : Use what you just learned to create a plot that depicts how the average

```
#weight of each species changes through the years.
yearly_weight <- surveys_complete %>%
  group_by(year, species_id) %>%
  summarize(avg_weight = mean(weight))
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

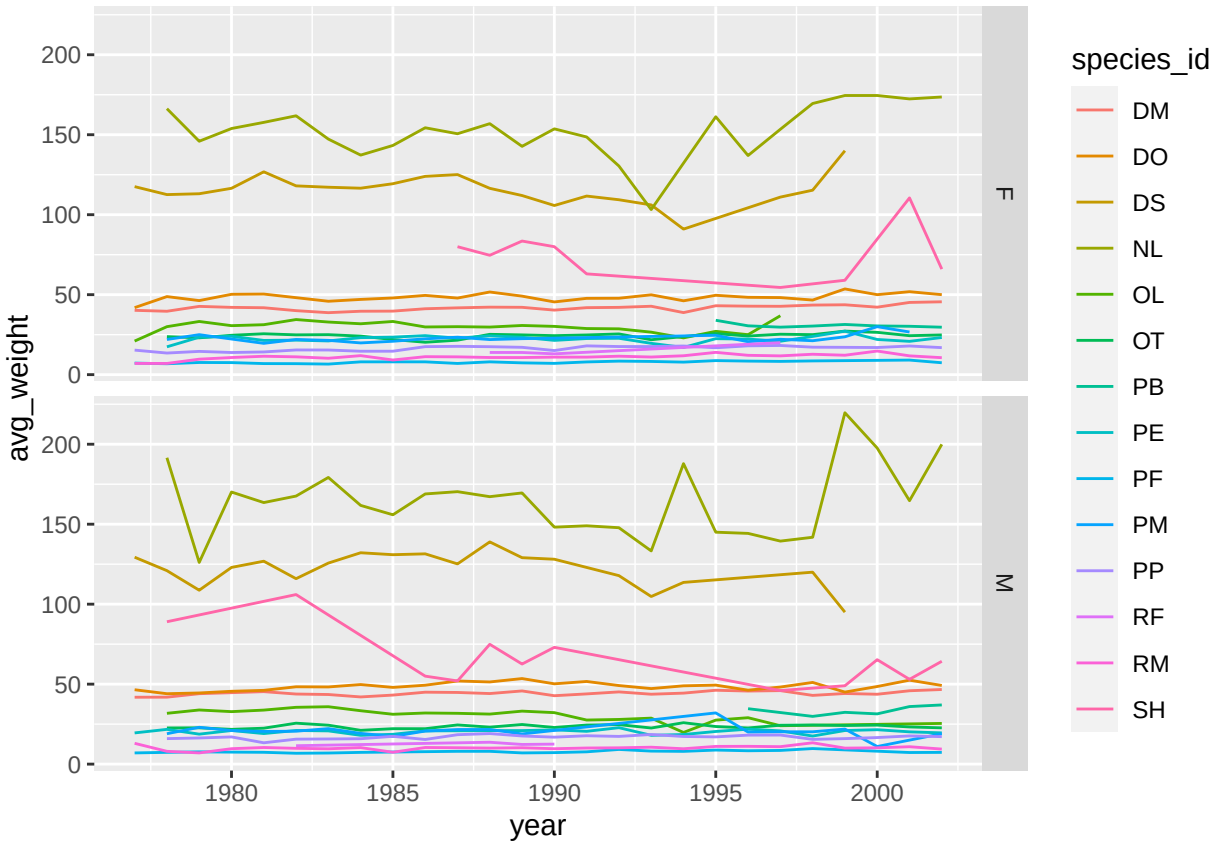
```
ggplot(data = yearly_weight, mapping = aes(x=year, y=avg_weight)) +
    geom_line() +
    facet_wrap(~ species_id) +
    theme_bw()
```



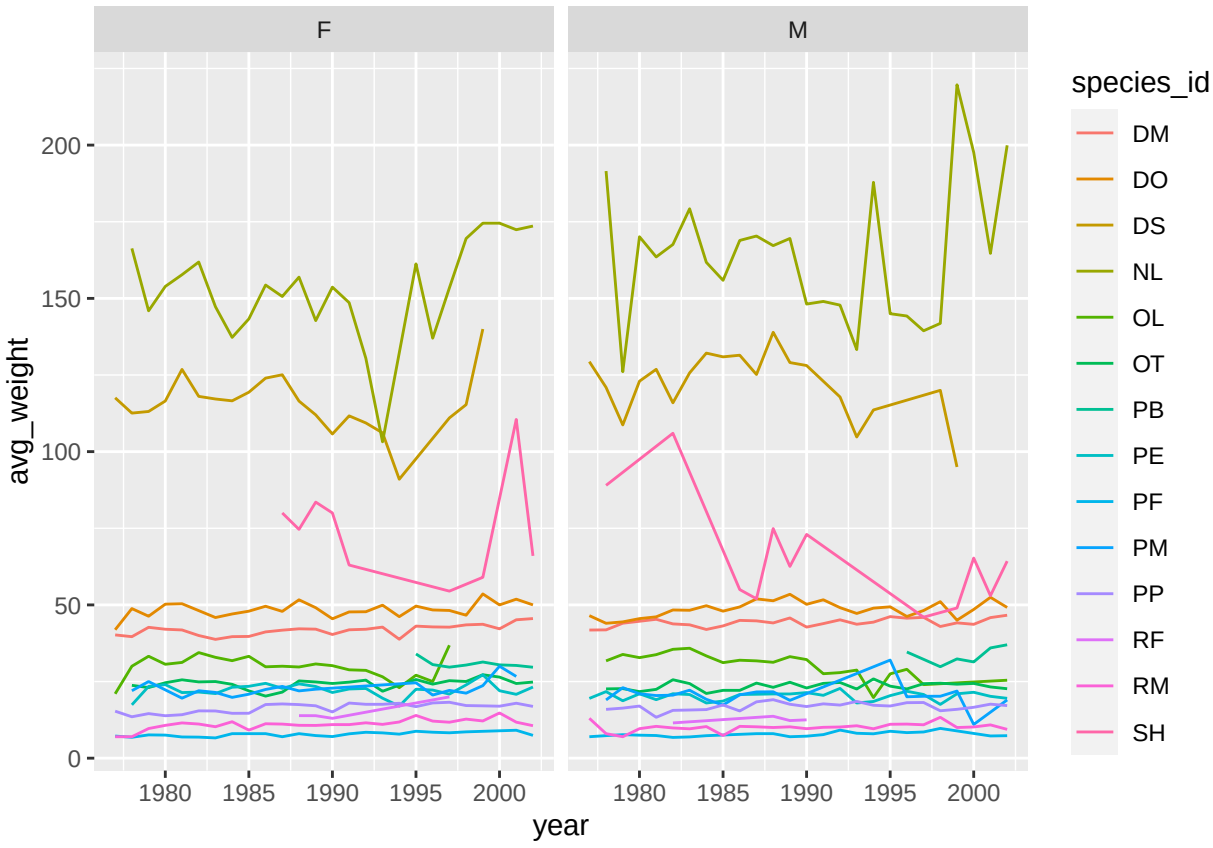

```
#Let's modify the previous plot to compare how the weights of males and females has changed through tim

# One column, facet by rows
yearly_sex_weight <- surveys_complete %>%
  group_by(year, sex, species_id) %>%
  summarize(avg_weight = mean(weight))
```

```
## `summarise()` has grouped output by 'year', 'sex'. You can override using the `.groups` argument.
```

```
ggplot(data = yearly_sex_weight,
       mapping = aes(x = year, y = avg_weight, color = species_id)) +
  geom_line() +
  facet_grid(sex ~ .)
```



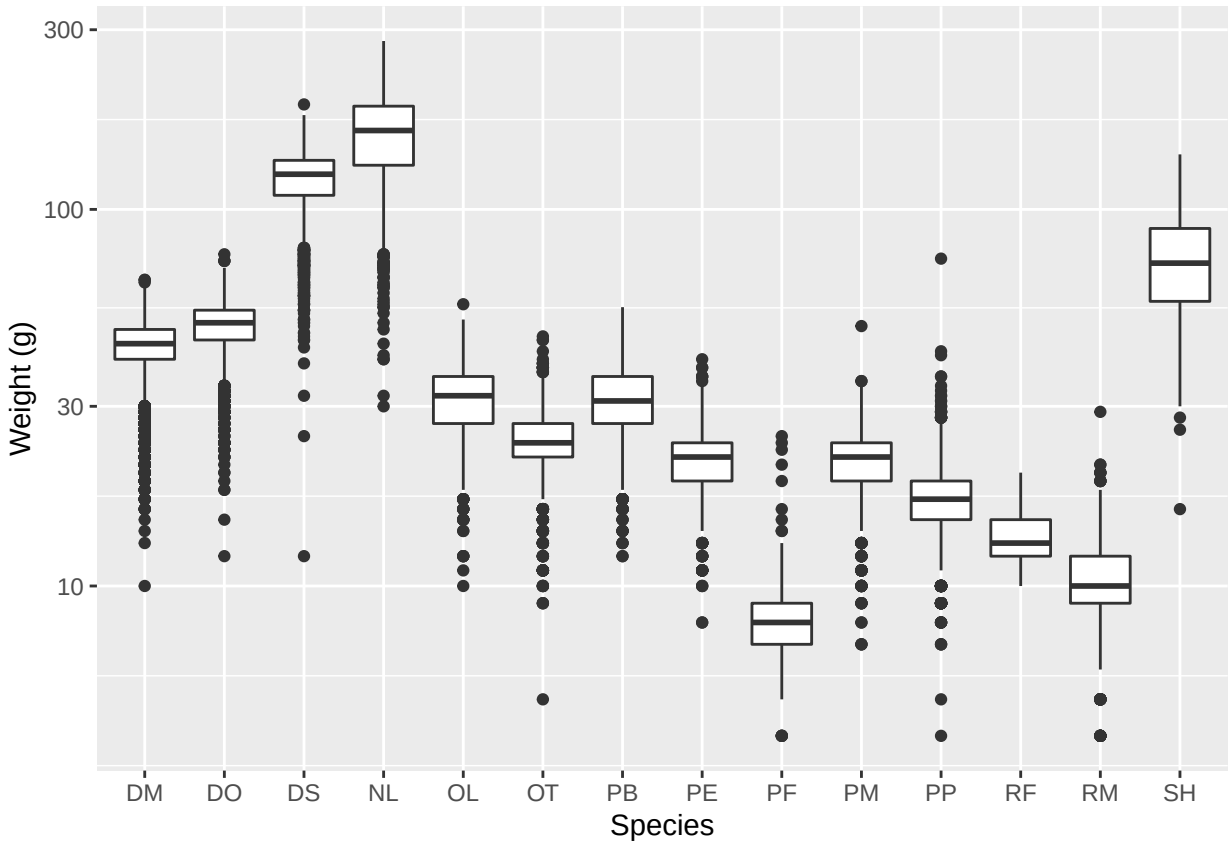

```
# One row, facet by column
ggplot(data = yearly_sex_weight,
       mapping = aes(x = year, y = avg_weight, color = species_id)) +
    geom_line() +
    facet_grid(. ~ sex)
```



8 : COMPOSING PLOTS

to produce a single figure that contains multiple independent plots

```
spp_weight_boxplot <- ggplot(data = surveys_complete,
                             mapping = aes(x = species_id, y = weight)) +
  geom_boxplot() +
  xlab("Species") + ylab("Weight (g)") +
  scale_y_log10()
spp_weight_boxplot
```



9 : EXPORTING DATAS

```
my_plot <- ggplot(data = yearly_sex_counts,
                  mapping = aes(x = year, y = n, color = sex)) +
    geom_line() +
    facet_wrap(~ species_id) +
    labs(title = "Observed species in time",
        x = "Year of observation",
        y = "Number of species") +
    theme_bw() +
    theme(axis.text.x = element_text(colour = "grey20", size = 12, angle = 90, hjust = 0.5, vjust = 0.5
          axis.text.y = element_text(colour = "grey20", size = 12),
          text=element_text(size = 16))
ggsave("fig_output/yearly_sex_counts.png", my_plot, width = 15, height = 10)
```

10 : ADDITIONAL EXERCISES

question 1,reproduce the figure (cfr cours)

```
library(rWSBIM1207)
beers<- data(beers)
library("tidyverse")
```

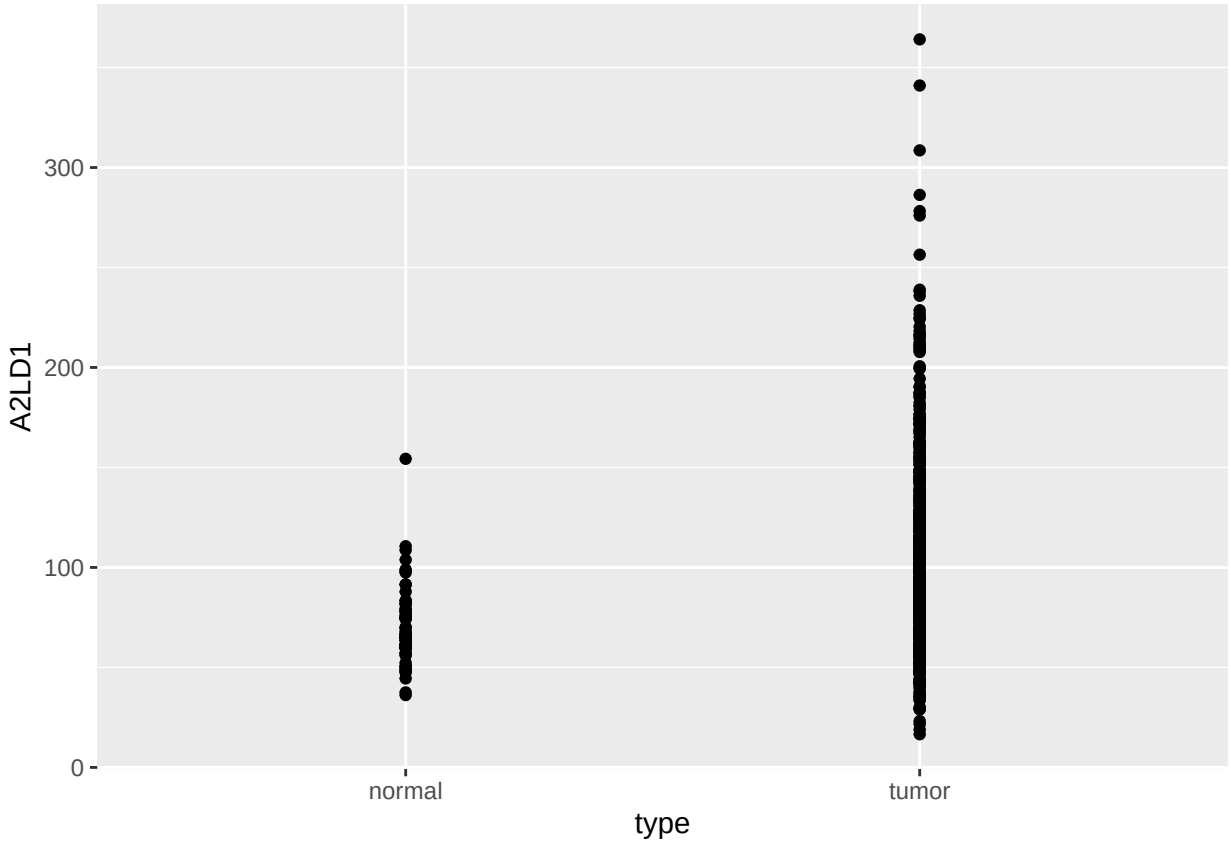
question 2

###1 Using geom_point, draw a plot showing distribution of expression levels of A2LD1 in normal tissue samples and in primary tumor samples

```
expression <- read_csv(expression.csv())
expression
```

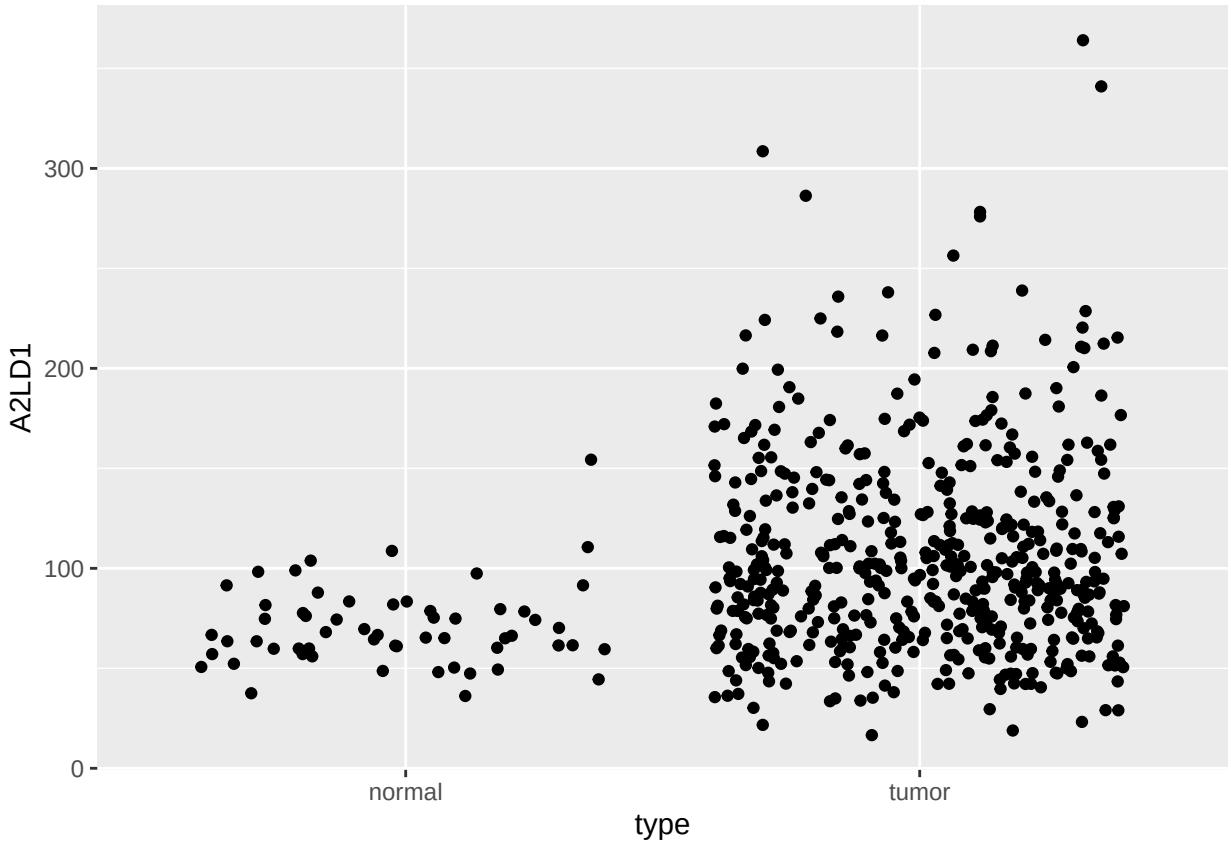
```
## # A tibble: 570 x 8
##    sampleID         patient      type   A1BG   A1CF A2BP1 A2LD1   A2ML1
##    <chr>            <chr>        <chr> <dbl>  <dbl> <dbl> <dbl>   <dbl>
##  1 TCGA-05-4244-01A TCGA-05-4244 tumor  26.0  0      1.75 136.    0.349
##  2 TCGA-05-4249-01A TCGA-05-4249 tumor 120.   0.322  1.61  89.1   1.61
##  3 TCGA-05-4250-01A TCGA-05-4250 tumor  50.9  0      0    151.    0
##  4 TCGA-05-4382-01A TCGA-05-4382 tumor 146.   0      0    112.    4.79
##  5 TCGA-05-4384-01A TCGA-05-4384 tumor 127.   0      0     87.6   0
##  6 TCGA-05-4389-01A TCGA-05-4389 tumor  67.1 36.2    0    112.   36.6
##  7 TCGA-05-4390-01A TCGA-05-4390 tumor 165.   0     97.7   64.9   0.639
##  8 TCGA-05-4395-01A TCGA-05-4395 tumor  22.0  0      0    181.  391.
##  9 TCGA-05-4396-01A TCGA-05-4396 tumor  17.4 15.9    0    134.    0.758
## 10 TCGA-05-4397-01A TCGA-05-4397 tumor 127.   0      0    229.   19.2
## # ... with 560 more rows
```

```
ggplot(data=expression, mapping = aes (x=type, y = A2LD1))+
  geom_point()
```



###2 Repeat this visualisation using this time the geom_jitter

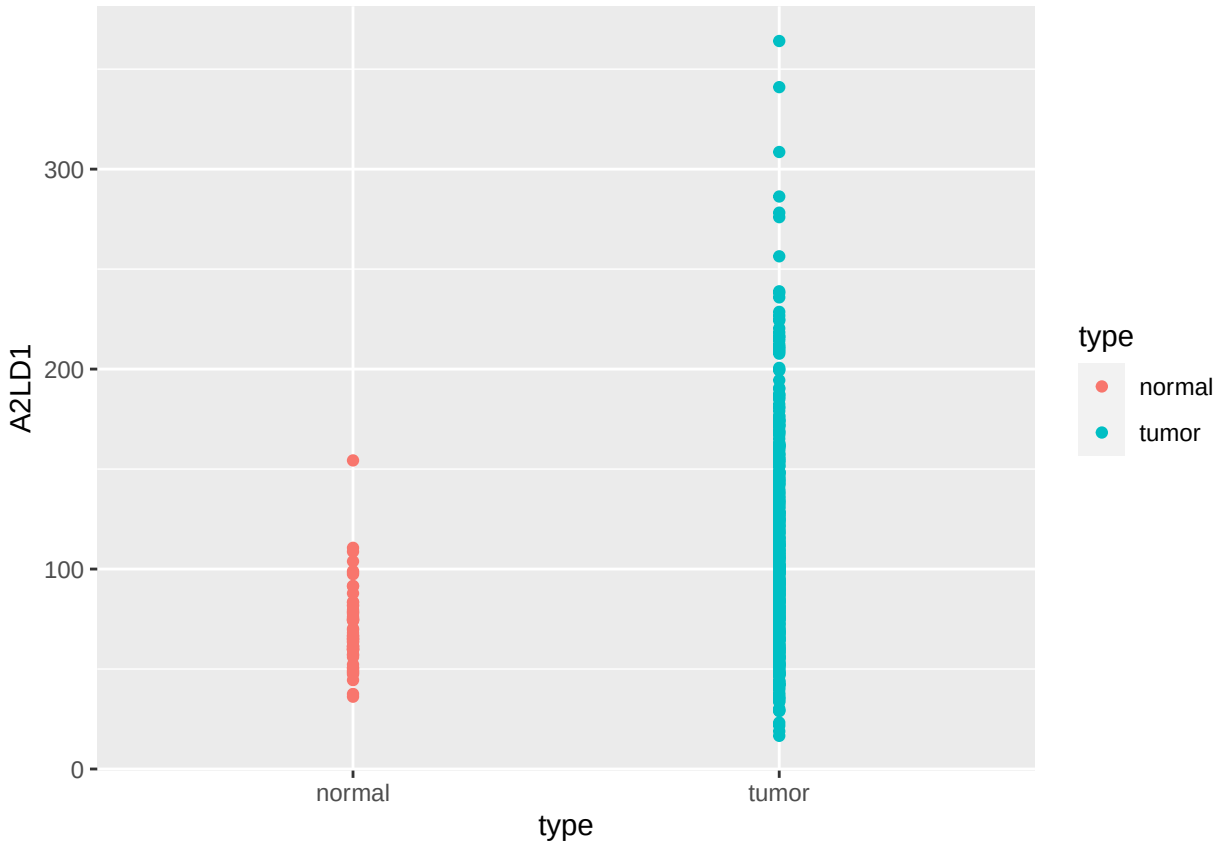
```
ggplot(data=expression, mapping = aes (x=type, y = A2LD1))+
  geom_jitter()
```



this is not a good visualisation

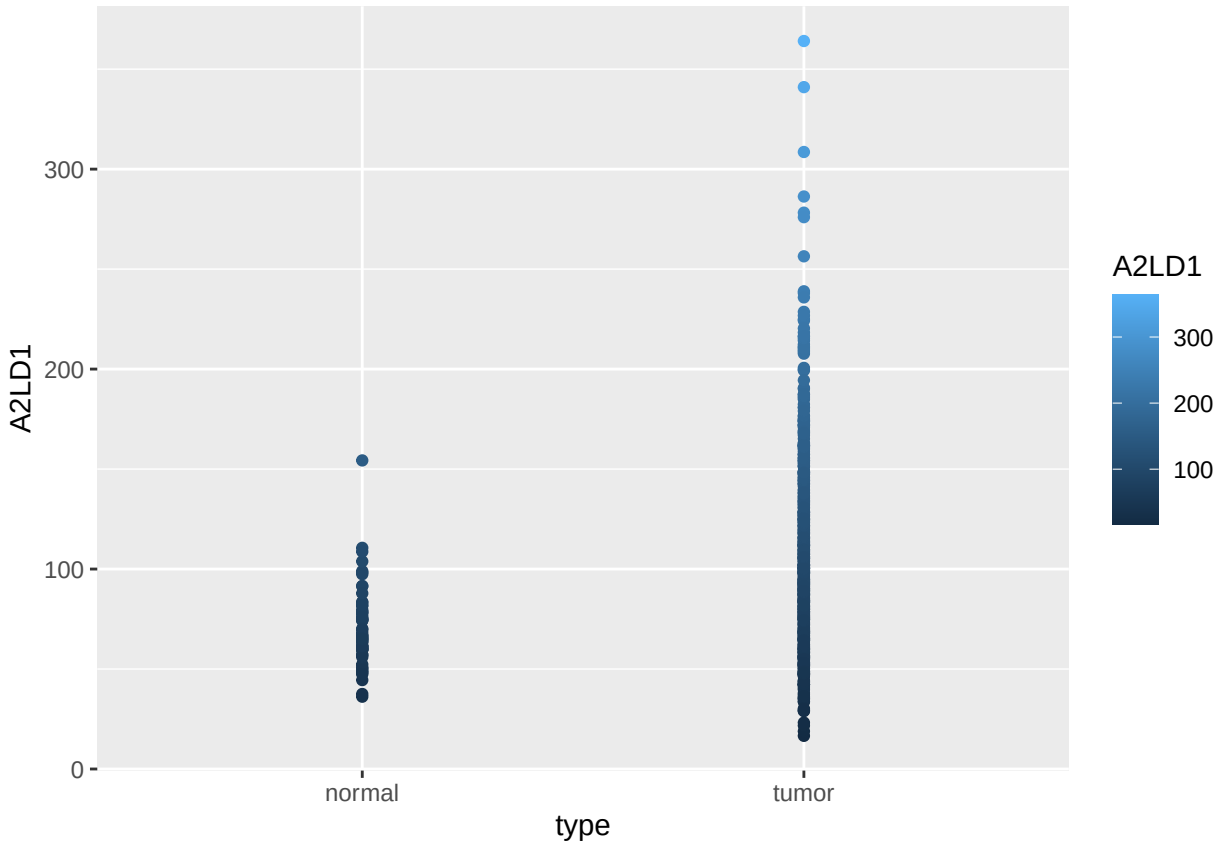
###3 Colour the samples according to the tissue type

```
ggplot(data=expression, mapping = aes (x=type, y = A2LD1, color = type))+
  geom_point()
```



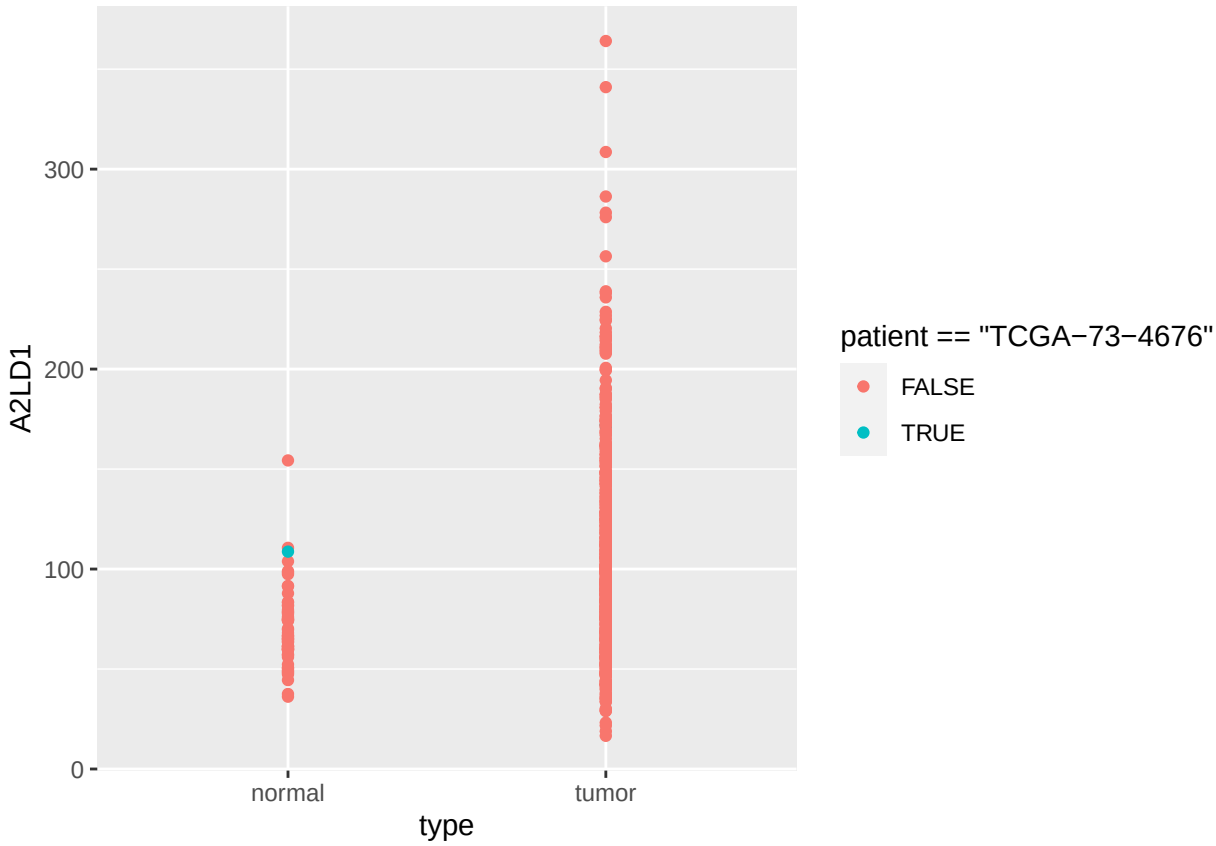
###4 Colour the samples according to their expression level in A2ML1

```
ggplot(data=expression, mapping = aes (x=type, y = A2LD1, color = A2LD1))+
  geom_point()
```



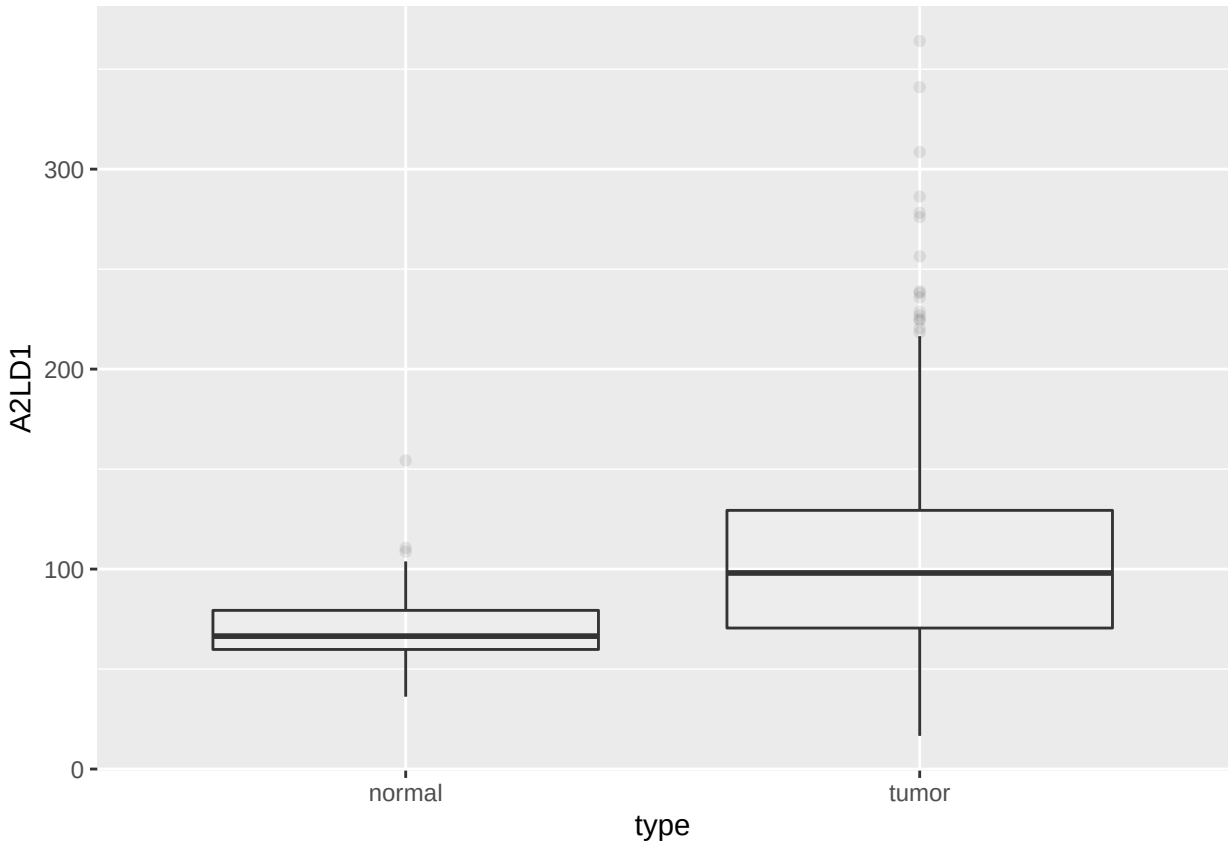
5 Highlight the points corresponding to patient "TCGA-73-4676"

```
ggplot(data=expression, mapping = aes (x=type, y = A2LD1, colour = patient == "TCGA-73-4676"))+
  geom_point()
```



6 Add a transparent boxplot to the graph

```
ggplot(data=expression, mapping = aes (x=type, y = A2LD1))+
  geom_boxplot(alpha = 0.1)
```

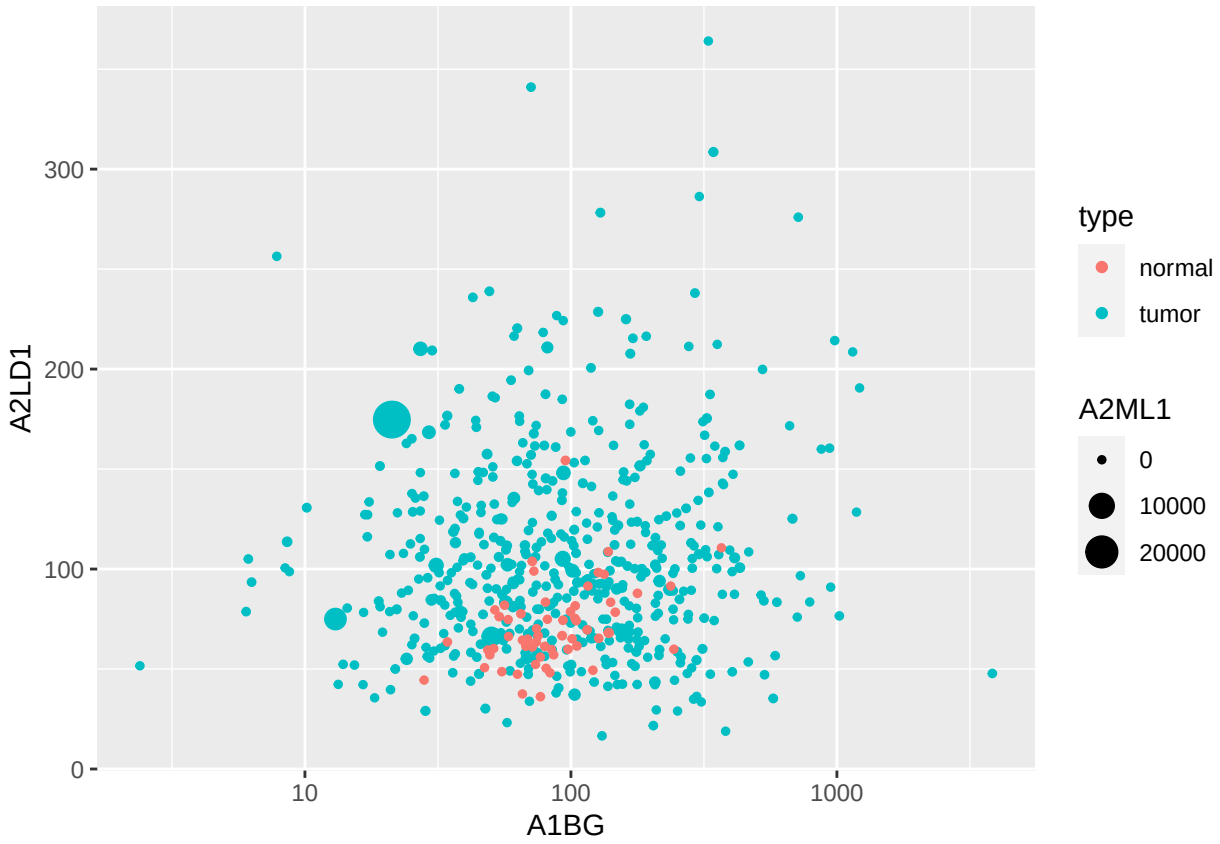


7 Change the y scale to log10 scale

```
#?
```

8 Visualise the expression of A1BG against that of A2LD1 setting the x axis on the log10 scale. Colours the observations based on their type and resize the points according to the expression level of the A2ML1 gene

```
ggplot(data = expression, mapping = aes(x = A1BG , y = A2LD1, color = type, size = A2ML1)) +
  geom_point()+
  scale_x_log10()
```

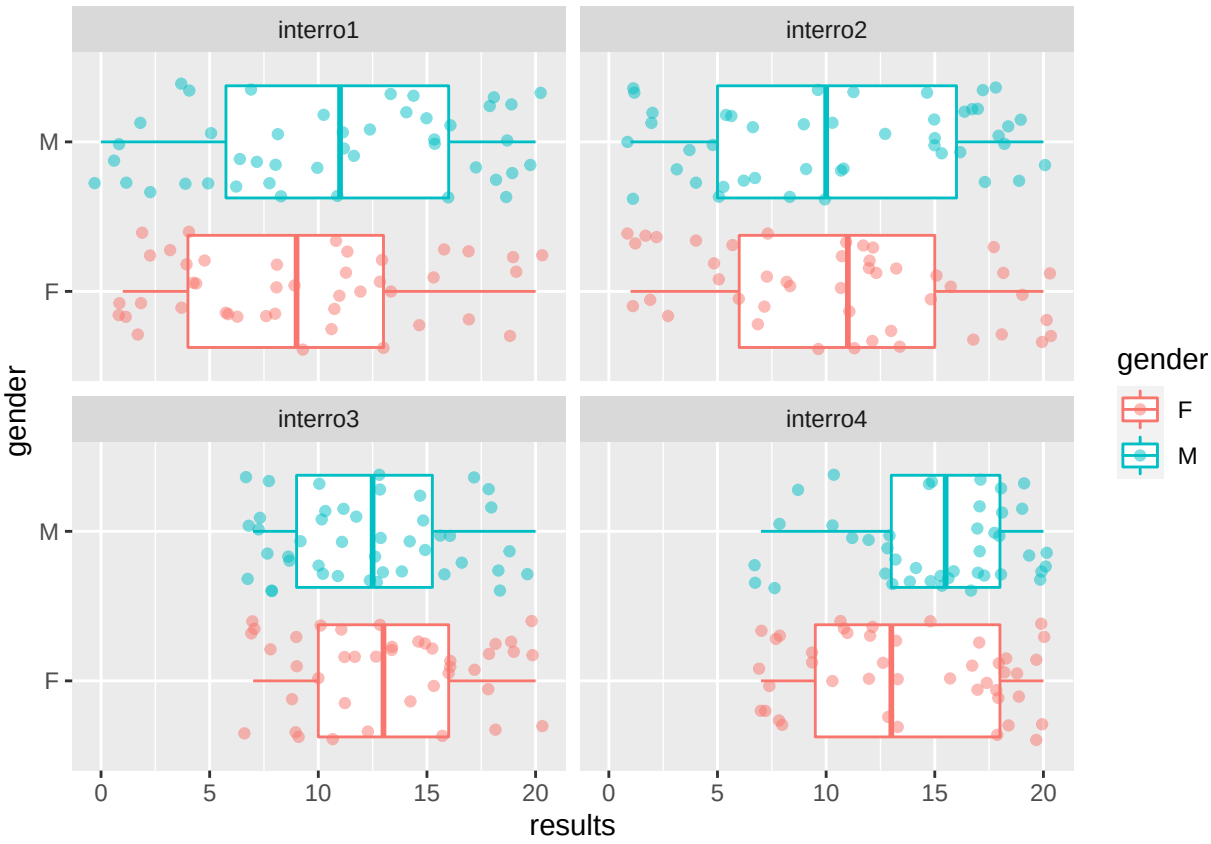


question 3

After gathering the interroA data from the rWSBIM1207 package in a long table format (see additional exercise chapter 5), visualise the result distributions for each test and male/female students group.

```
interroA <- read_csv(interroA.csv())
longer <- interroA %>% pivot_longer(cols = c(interro1, interro2, interro3, interro4),
                                    names_to= "interros", values_to = "results")
longer <- na.omit(longer)
```

```
ggplot(longer, aes(x= results , y = gender , colour = gender)) + facet_wrap(~interros) +
  geom_boxplot() + geom_jitter(alpha = 0.5)
```


interro1
interro2
interro3
interro4
gender
M
F
0
5
10
15
20
results
gender
F
M