https://intro2r.com/ Chapter 5

CSCI 297b, Spring 2023

May 1, 2023

R graphics

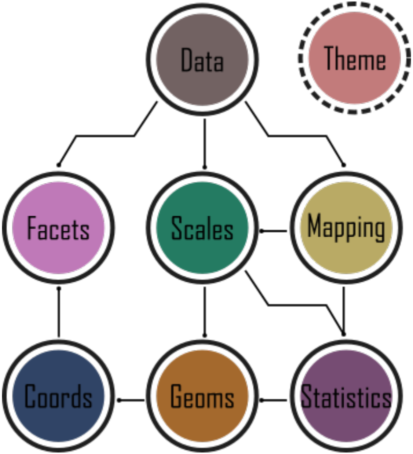
- R provides easy, quick and dirty graphics for work and exploration
- R also provides sophisticated, publication-ready graphics
- R graphics are highly customizable, feeding our creative side
- This enables better expression and communication
- Opposite of “click scatterplot button”

The grammar of graphics

- https://link.springer.com/book/10.1007/0-387-28695-0
- https://www.jstatsoft.org/article/view/v017b03/v17b03.pdf
- https://ggplot2-book.org/

The grammar of graphics

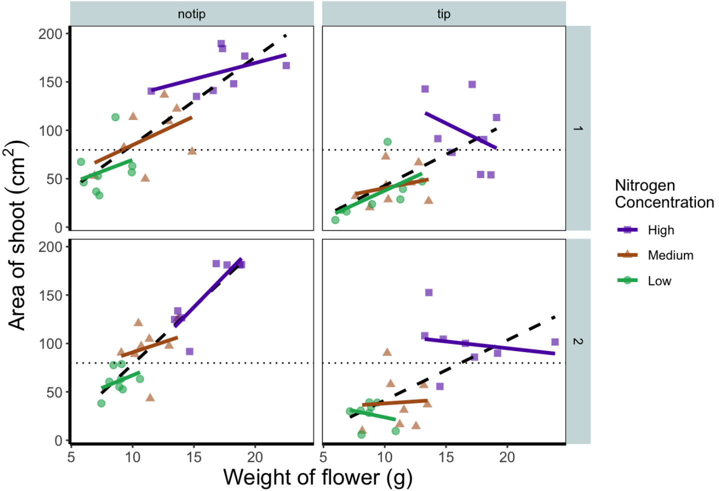
- The user should be in control of all components (i.e. layers) and produce a tailor-made figure fit for their specific needs.



ggplot2

- In 2007 ggplot2 was released by Hadley Wickham.
- By 2017 the package had reportedly been downloaded 10 million times.
- `ggplot2` is not required for publication quality graphics.
- Base graphics can do the job, it's just easier with `ggplot2`

A final figure



Note: Regression assumptions are unvalidated

- We will develop this figure step-by-step.

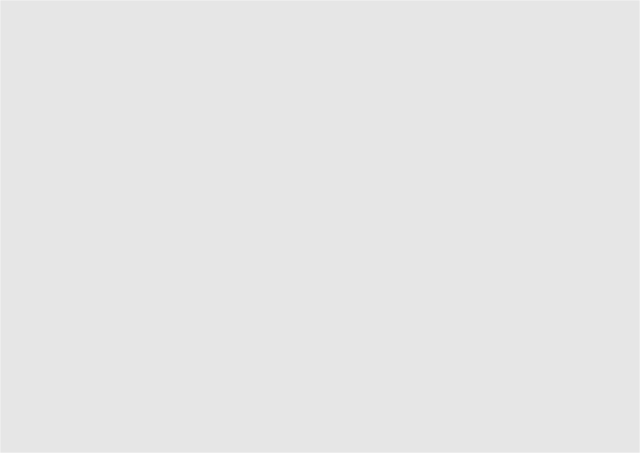
ggplot2 library

```
install.packages("ggplot2")
library(ggplot2)
```

- Installing the package is not necessary on RStudio Workbench

The base ggplot

```
ggplot()
```



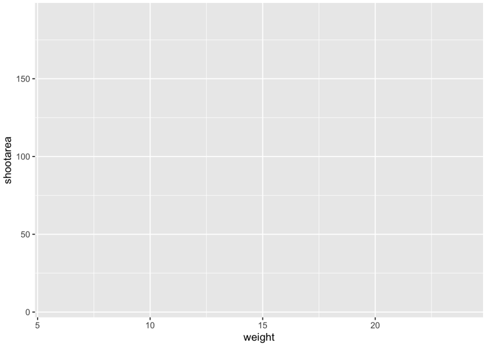
A reminder of the flower data set

```
flower <- read.table("data/flower.csv",
                          stringsAsFactors = TRUE,
                          header = TRUE, sep = ",")
str(flower)
## 'data.frame':    96 obs. of  8 variables:
##  $ treat    : Factor w/ 2 levels "notip","tip": 2 2 2 2 2
##  $ nitrogen : Factor w/ 3 levels "high","low","medium": 3
##  $ block    : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ height   : num  7.5 10.7 11.2 10.4 10.4 9.8 6.9 9.4 10
##  $ weight   : num  7.62 12.14 12.76 8.78 13.58 ...
##  $ leafarea : num  11.7 14.1 7.1 11.9 14.5 12.2 13.2 14 1
##  $ shootarea: num  31.9 46 66.7 20.3 26.9 72.7 43.1 28.5
##  $ flowers  : int  1 10 10 1 4 9 7 6 5 8 ...
```

Getting started

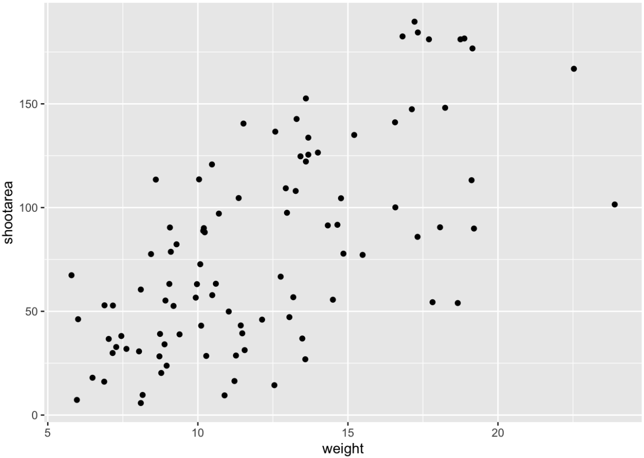
- We want `shootarea` on the *y* axis and `weight` on the *x* axis.
- To do this we specify a `mapping` which is an `aesthetic`.

```
# Including aesthetics for x and y axes as well as specifying the dataset
ggplot(mapping = aes(x = weight, y = shootarea), data = flower)
```



To see something we need geometry layers

```
ggplot(aes(x = weight, y = shootarea), data = flower) +
          geom_point()      # Adding a geom to display data as point data
```



ggplot is like painting in layers

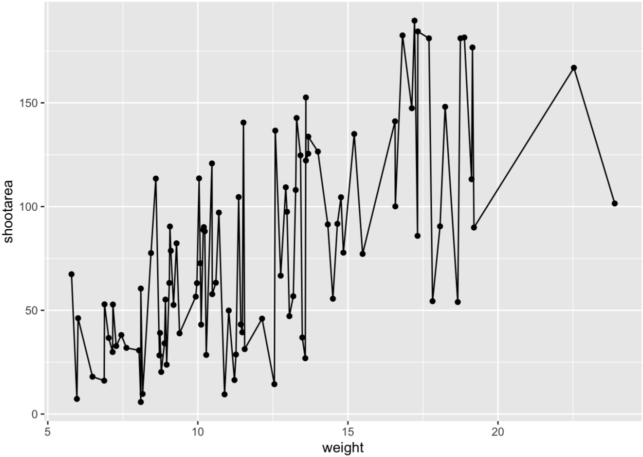
```
ggplot(aes(x = weight, y = shootarea), data = flower) +
       geom_point()     # Adding a geom to display data as point data
```

- There are three essential layers

- Other layers are optional, defaults handle most things.

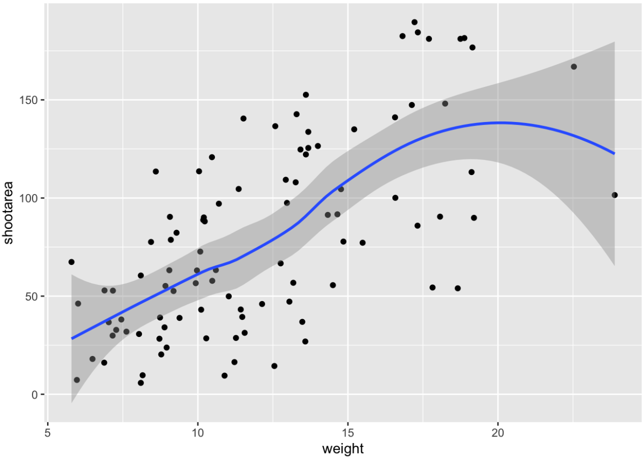
Add line geometry layer

```
ggplot(aes(x = weight, y = shootarea), data = flower) +
    geom_point() +
    geom_line()     # Adding geom_line
```

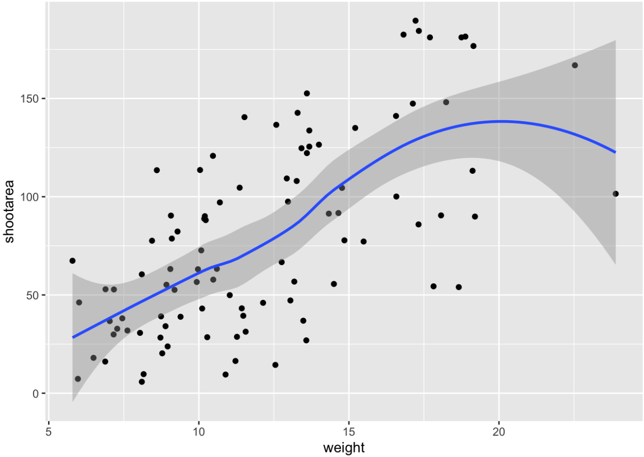


Add smooth geometry layer

```
ggplot(aes(x = weight, y = shootarea), data = flower) +
    geom_point() +
    geom_smooth()     # Adding geom_line
```



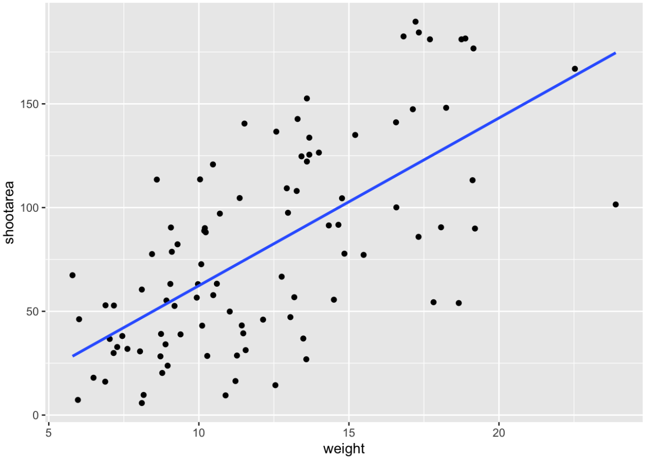
LOESS default



- locally estimated scatterplot smoothing
- We want a simple linear fit, from a linear model
- Also, remove the confidence interval

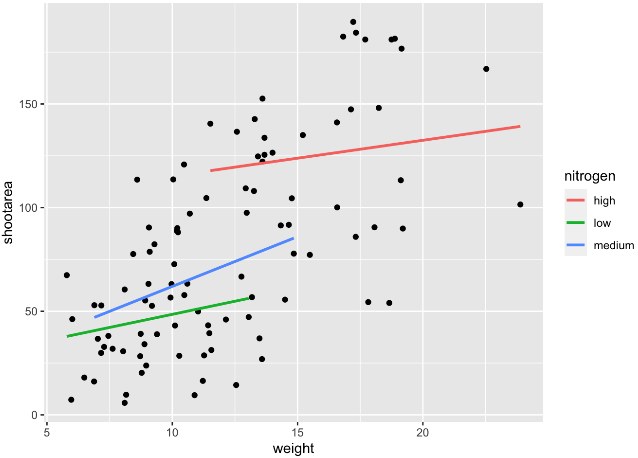
Add linear model line layer

```
ggplot(aes(x = weight, y = shootarea), data = flower) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE)    # method and se
```



Add linear models for each level of nitrogen

```
ggplot(aes(x = weight, y = shootarea), data = flower) +
  geom_point() +
  # Including colour argument in aes()
  geom_smooth(aes(color = nitrogen), method = "lm", se = FALSE)
```



Where to put information

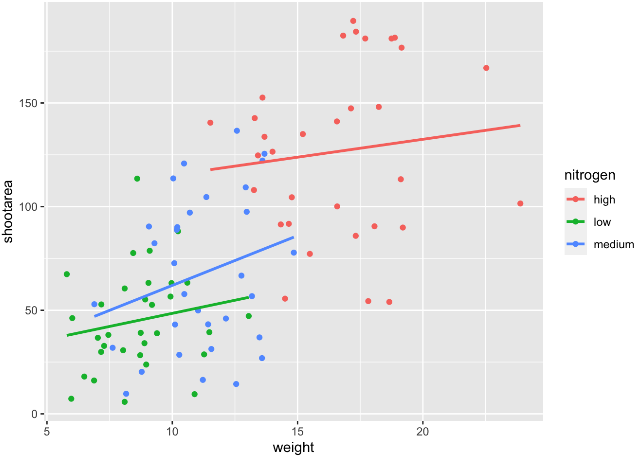
```
ggplot(aes(x = weight, y = shootarea), data = flower) +
  geom_point() +
  # Including colour argument in aes()
  geom_smooth(aes(color = nitrogen), method = "lm", se = FALSE)

ggplot() +
  # Moved aes() and data into geoms
  geom_point(aes(x = weight, y = shootarea), data = flower) +
  geom_smooth(aes(x = weight, y = shootarea, colour = nitrogen),
                           data = flower, method = "lm", se = FALSE)
```

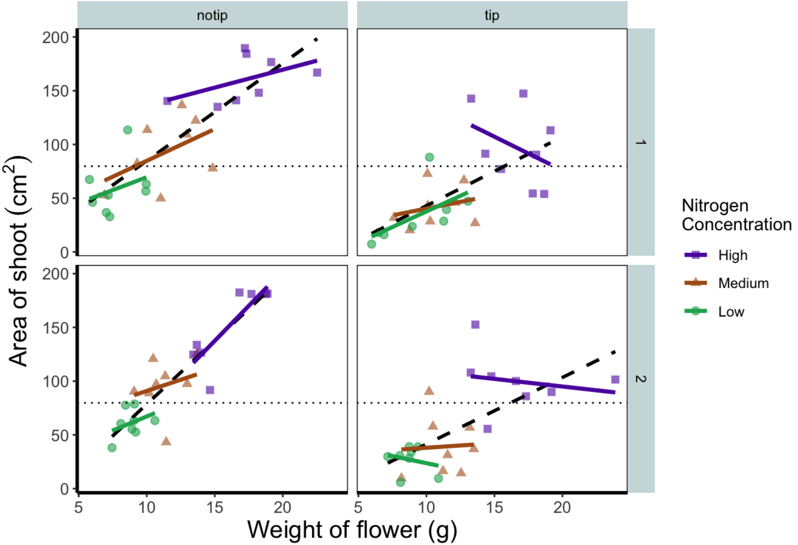
- When we include information such as `data =` and `aes()` in `ggplot()` we are setting those as the default, universal values which all subsequent geoms use.
- Whereas if we were to include that information within a geom, only that geom would use that specific information.

Moving color into ggplot colors points, too

```
# Moved colour = nitrogen into the universal ggplot()
ggplot(aes(x = weight, y = shootarea, color = nitrogen), data = flower) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



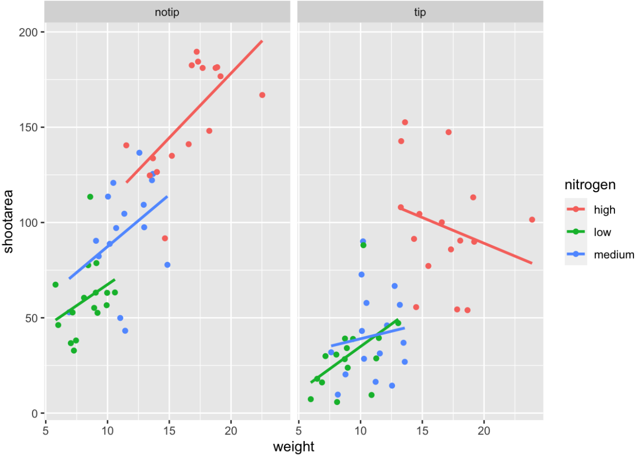
Wrapping grids



Note: Regression assumptions are unvalidated

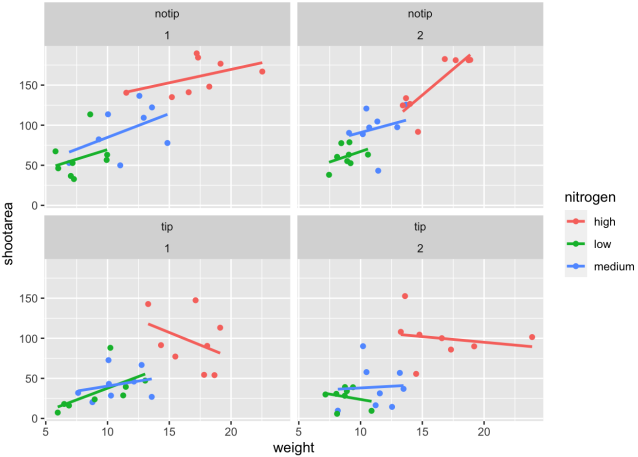
Facets

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
    # Splitting the single figure into multiple depending on treatment
  facet_wrap(~ treat)
```



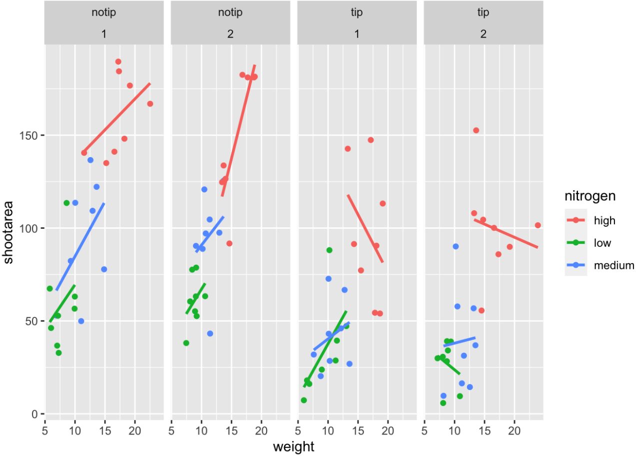
Facets

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
    # Adding "block" to formula
  facet_wrap(~ treat + block)
```



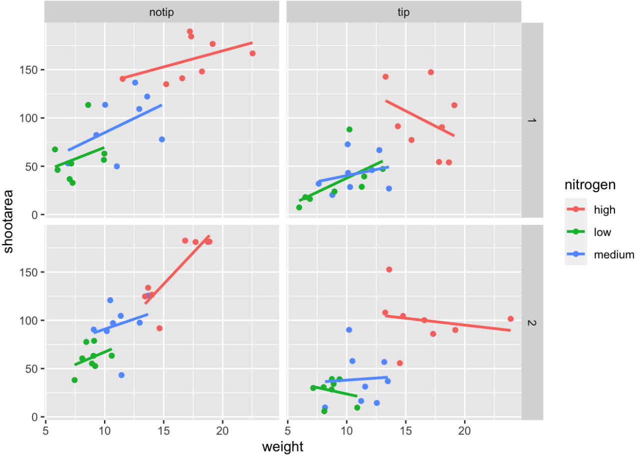
Facets

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
    # Changing to facet_grid
  facet_grid(~ treat + block)
```



Facets

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
    # Rearranging formula, block in relation to treatment
  facet_grid(block ~ treat)
```



Multiple figures using `patchwork`

```
library(patchwork)
```

- Arrange figures above and below each other with /
- Arrange figures side by side with + or |

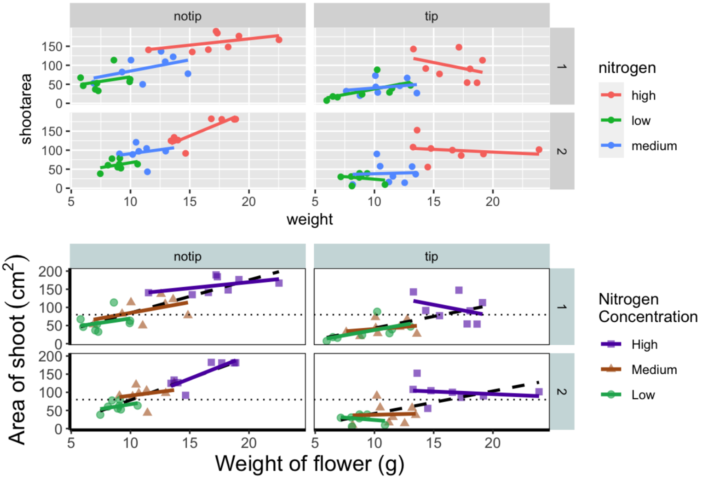
Assign plots to objects

```
rbook_figure <- ggplot(aes(x = weight,
                     y = shootarea,
                     colour = nitrogen), data = flower) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_grid(block ~ treat)
```

- The figure we just constructed
- Nothing is plotted if we assign it to an object
- To see the figure, enter its name at the console

Arrange named figures with patchwork

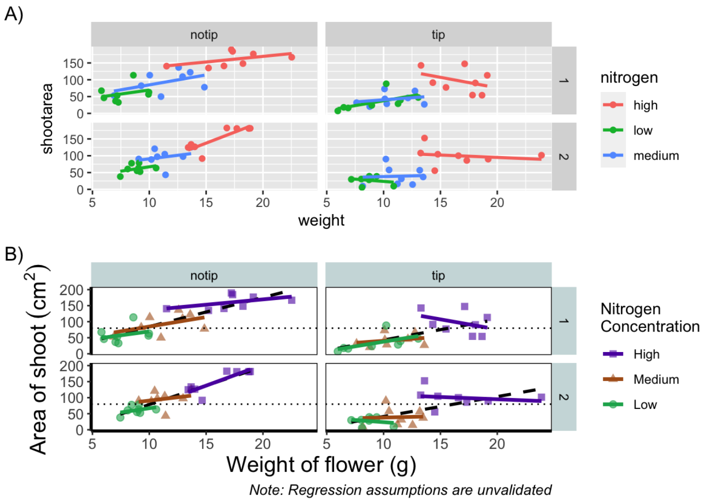
```
rbook_figure / final_figure
```



Note: Regression assumptions are unvalidated

Caption individual figures with patchwork

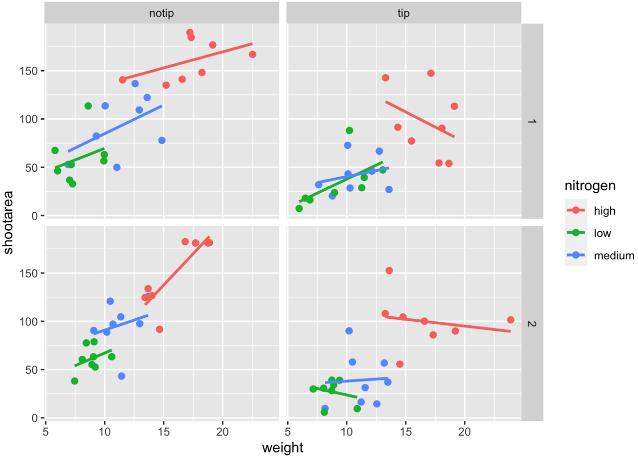
```
nested_compare <- rbook_figure / final_figure
nested_compare +
  plot_annotation(tag_levels = "A", tag_suffix = ")")
```



Note: Regression assumptions are unvalidated

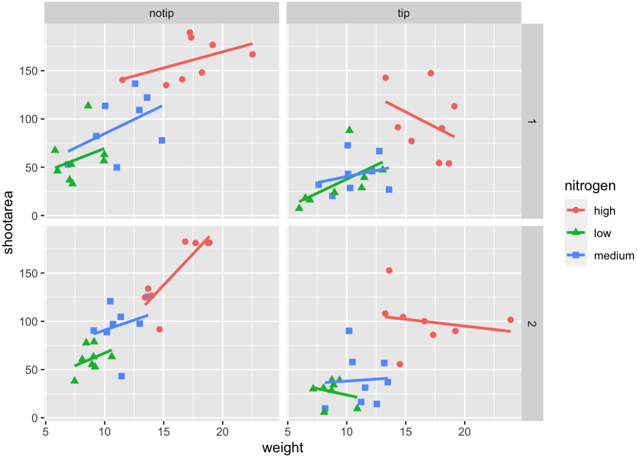
Changing point size

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
    geom_point(size = 2) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(block ~ treat)
```



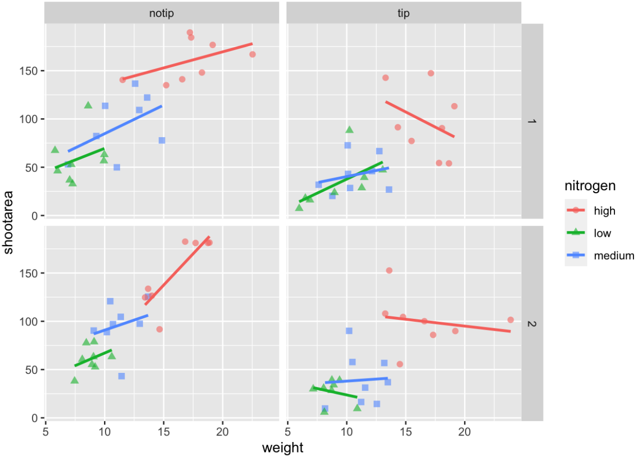
Changing point shape

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
    geom_point(aes(shape = nitrogen), size = 2) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(block ~ treat)
```



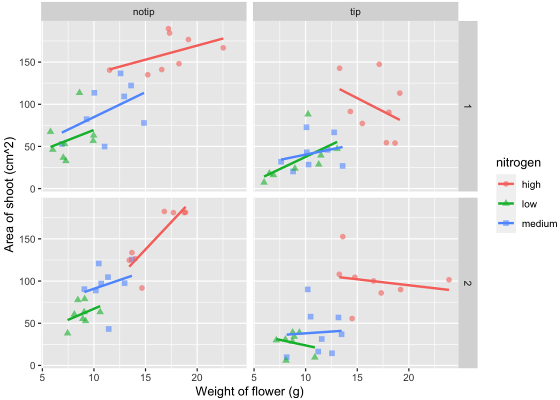
Changing point transparency

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
    geom_point(aes(shape = nitrogen), size = 2, alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(block ~ treat)
```



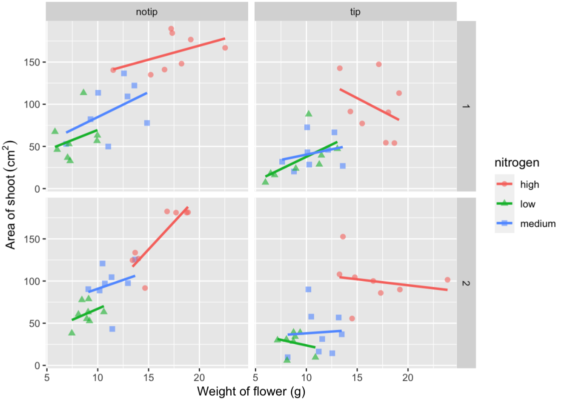
Changing axis labels

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
    geom_point(aes(shape = nitrogen), size = 2, alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(block ~ treat) +
    xlab("Weight of flower (g)") +
    ylab("Area of shoot (cm^2)")
```



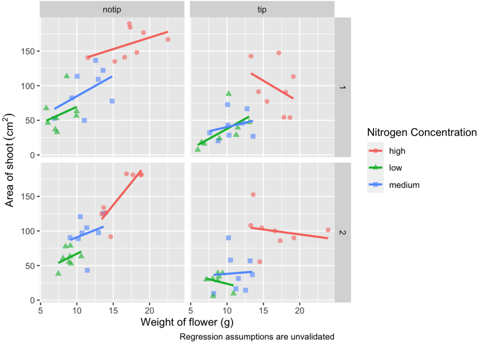
Special character interpretation with bquote

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
    geom_point(aes(shape = nitrogen), size = 2, alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(block ~ treat) +
    xlab("Weight of flower (g)") +
    ylab(bquote("Area of shoot"~(cm^2)))
```



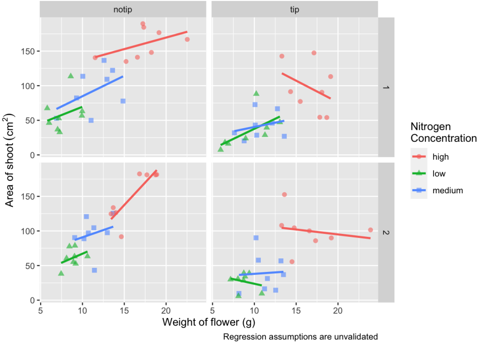
Fix some labels

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
    geom_point(aes(shape = nitrogen), size = 2, alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(block ~ treat) +
    xlab("Weight of flower (g)") +
    ylab(bquote("Area of shoot"~(cm^2))) +
    labs(shape = "Nitrogen Concentration", color = "Nitrogen Concentration",
             caption = "Regression assumptions are unvalidated")
```



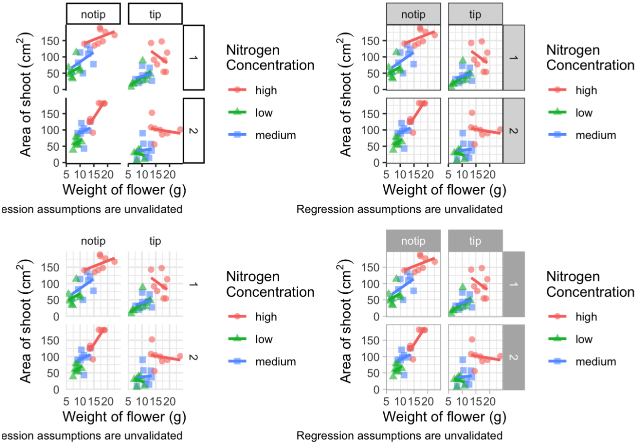
Using \n for newlines

```
ggplot(aes(x = weight, y = shootarea, color = nitrogen), data = flower) +
    geom_point(aes(shape = nitrogen), size = 2, alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_grid(block ~ treat) +
    xlab("Weight of flower (g)") +
    ylab(bquote("Area of shoot"~(cm^2))) +
    labs(shape = "Nitrogen\nConcentration", color = "Nitrogen\nConcentration",
             caption = "Regression assumptions are unvalidated")
```



Themes

```
classic <- rbook_figure  + theme_classic()
bw <- rbook_figure       + theme_bw()
minimal <- rbook_figure  + theme_minimal()
light <- rbook_figure    + theme_light()
(classic | bw) / (minimal | light)
```

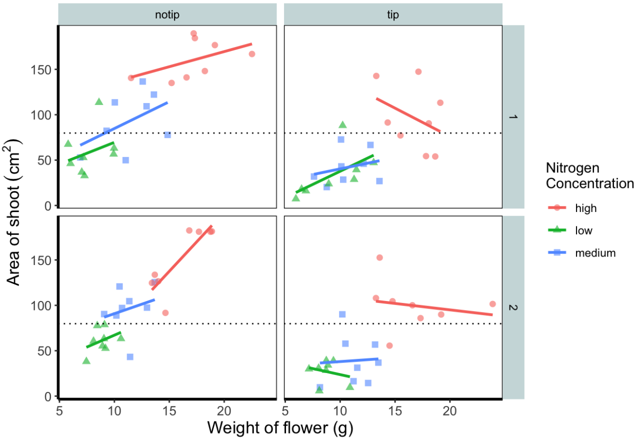


More themes

- Package `ggthemes`
- Write your own theme (see text)

Add horizontal line

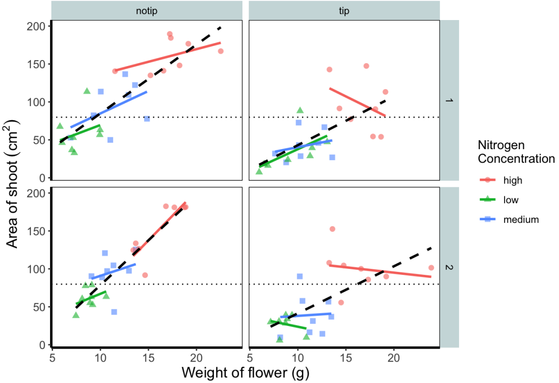
```
rbook_figure +
    geom_hline(aes(yintercept = mean(shootarea)),
          linewidth = 0.5, colour = "black", linetype = 3)
```



Regression assumptions are unvalidated

Adding a second geom_smooth

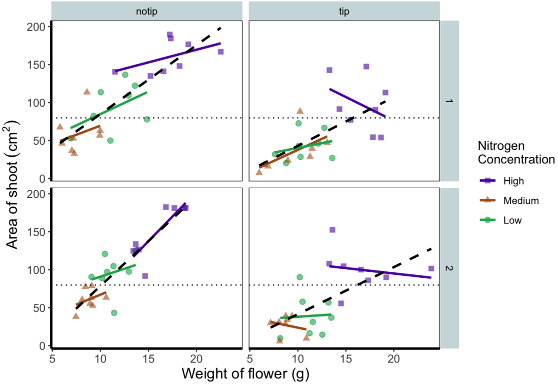
```
ggplot(aes(x = weight, y = shootarea, color = nitrogen), data = flower) +
    geom_point(aes(shape = nitrogen), size = 2, alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE) +
    # Adding a SECOND geom_smooth :O
    geom_smooth(method = "lm", se = FALSE, linetype = 2, alpha = 0.6,
                          color = "black")  # overrides the default
    ....
```



Regression assumptions are unvalidated

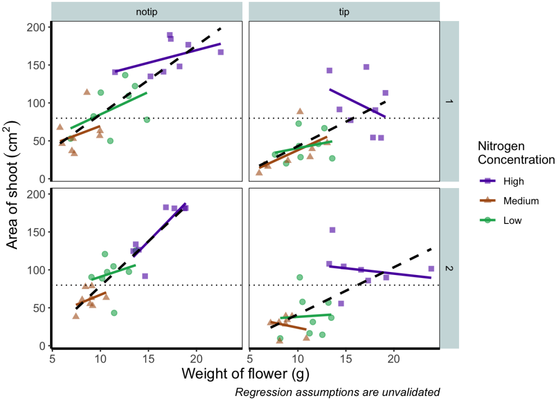
Changing colors and scale labels manually

```
ggplot(aes(x = weight, y = shootarea, colour = nitrogen), data = flower) +
    ...
    # Setting colour and associated labels
    scale_colour_manual(values = c("#5C1AAE", "#AE5C1A", "#1AAE5C"),
       labels = c("High", "Medium", "Low")) +
    # Setting shape and associated labels
    scale_shape_manual(values = c(15,17,19),
       labels = c("High", "Medium", "Low"))
```



Regression assumptions are unvalidated

Labels are incorrect



Regression assumptions are unvalidated

- High, Medium and Low were arbitrary
- Fix with:

```
flower$nitrogen <- factor(flower$nitrogen, levels = c("high", "medium", "low"))
```

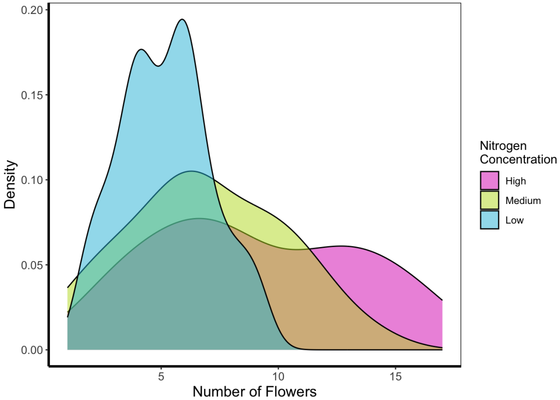
Saving ggplots

```
ggsave(filename = "areashoot_weight_facet.pdf",
       plot = rbook_figure,
       device = "pdf",
       path = "output",
       width = 250,
       height = 150,
       units = "mm",
       dpi = 500,
       limitsize = TRUE)
```

- If `plot` is not specified, last thing plotted is saved.

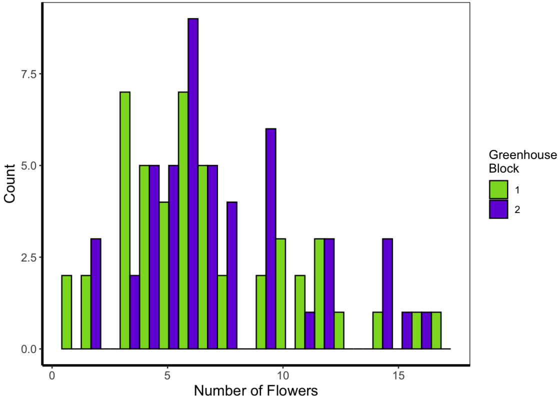
Density plot

```
ggplot(flower) +
    geom_density(aes(x = flowers, fill = nitrogen), alpha = 0.5) +
    labs(y = "Density", x = "Number of Flowers", fill = "Nitrogen\nConcentratio
    scale_fill_manual(labels = c("High", "Medium", "Low"),
        values = c("#DB24BC", "#BCDB24", "#24BCDB")) +
    theme_rbook()
```



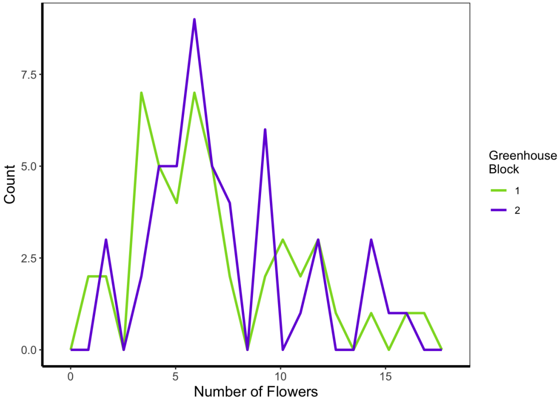
Histogram

```
ggplot(flower) +
    geom_histogram(aes(x = flowers, fill = factor(block)), colour = "black", bi
                 position = "dodge") +
    labs(y = "Count", x = "Number of Flowers", fill = "Greenhouse\nBlock") +
    scale_fill_manual(labels = c("1", "2"),
          values = c("#8CD926", "#7326D9")) +
    theme_rbook()
```



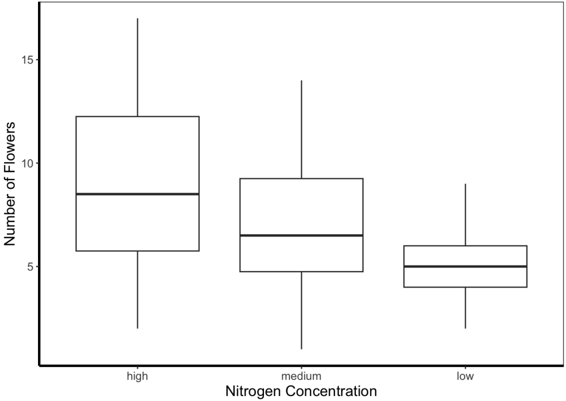
Frequency polygons

```
ggplot(flower) +
    geom_freqpoly(aes(x = flowers, colour = factor(block)), size = 1, bins = 20
    labs(y = "Count", x = "Number of Flowers", colour = "Greenhouse\nBlock") +
    scale_colour_manual(labels = c("1", "2"),
         values = c("#8CD926", "#7326D9")) +
    theme_rbook()
```



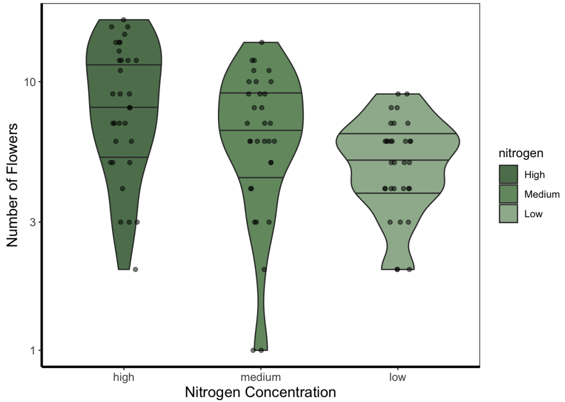
Boxplot

```
ggplot(flower) +
    geom_boxplot(aes(y = flowers, x = nitrogen)) +
    labs(y = "Number of Flowers", x = "Nitrogen Concentration") +
    theme_rbook()
```



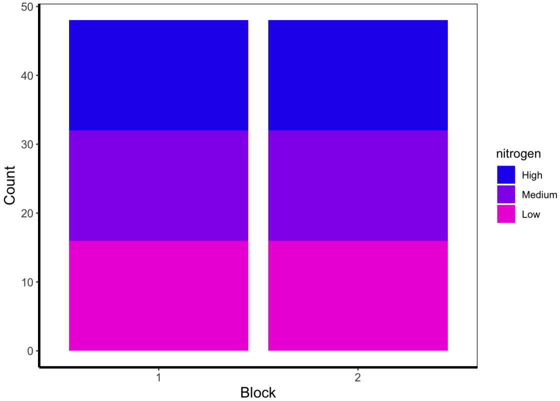
Violin plots

```
ggplot(flower) +
  geom_violin(aes(y = flowers, x = nitrogen, fill = nitrogen),
              draw_quantiles = c(0.25, 0.5, 0.75)) +
  geom_jitter(aes(y = flowers, x = nitrogen), colour = "black", height = 0,
              width = 0.1, alpha = 0.5) +
    scale_fill_manual(labels = c("High", "Medium", "Low"),
         values = c("#5f7f5c", "#749770", "#9eb69b")) +
    labs(y = "Number of Flowers", x = "Nitrogen Concentration") + scale_y_log10
```



Barchart

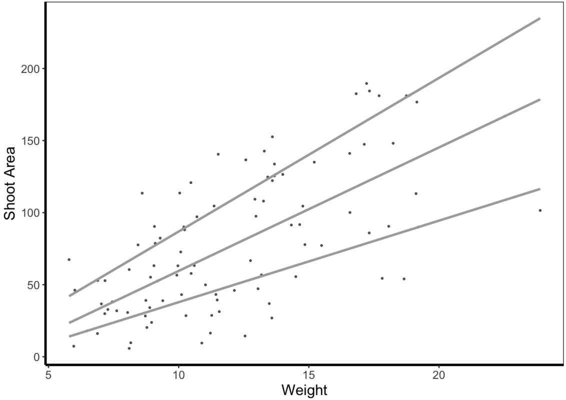
```
ggplot(flower) +
  geom_bar(aes(x = factor(block), fill = nitrogen)) +
    scale_fill_manual(labels = c("High", "Medium", "Low"),
         values = c("#2613EC", "#9313EC", "#EC13D9")) +
    labs(y = "Count", x = "Block") +
    theme_rbook()
```



Quantile lines

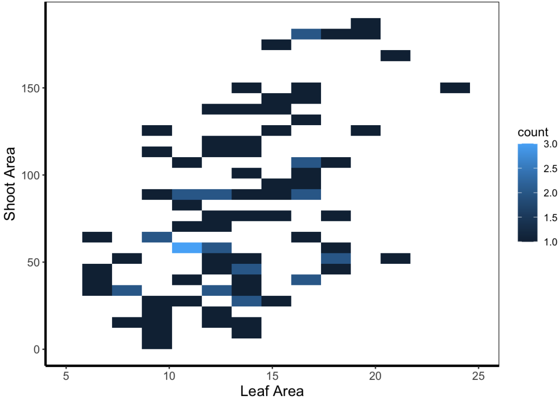
```
library(quantreg)

ggplot(aes(x = weight, y = shootarea), data = flower) +
    geom_point(size = 0.5, alpha = 0.6) +
    geom_quantile(colour = "darkgrey", size = 1) +
    labs(y = "Shoot Area", x = "Weight") +
    theme_rbook()
```



Heatmap

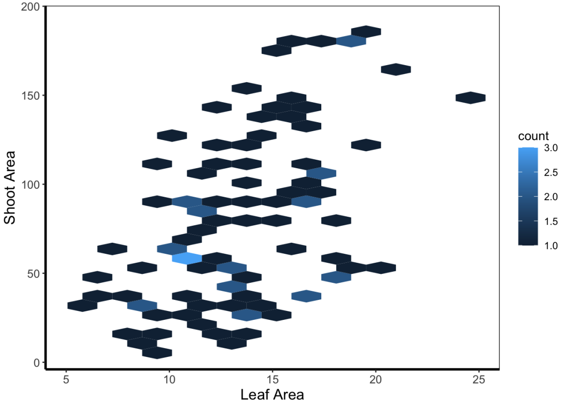
```
ggplot(aes(x = leafarea, y = shootarea), data = flower) +
    geom_bin2d() +
    labs(y = "Shoot Area", x = "Leaf Area") +
    coord_cartesian(xlim = c(5,25)) +
    theme_rbook()
```



Hex map

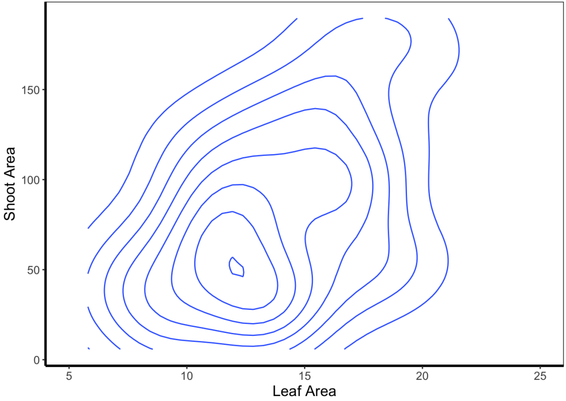
```
library(hexbin)

ggplot(aes(x = leafarea, y = shootarea), data = flower) +
    geom_hex() +
    labs(y = "Shoot Area", x = "Leaf Area") +
    coord_cartesian(xlim = c(5,25)) +
    theme_rbook()
```



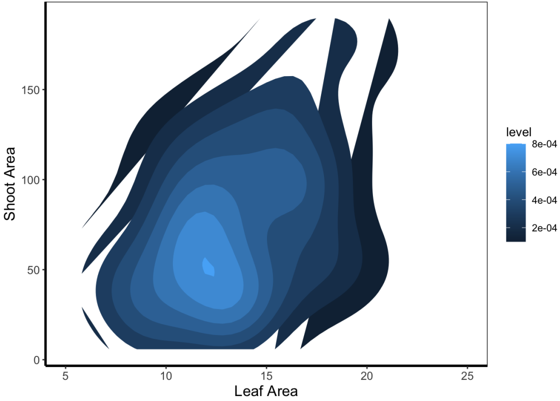
Contour map

```
ggplot(aes(x = leafarea, y = shootarea), data = flower) +
    geom_density2d() +
    labs(y = "Shoot Area", x = "Leaf Area") +
    coord_cartesian(xlim = c(5,25)) +
    theme_rbook()
```



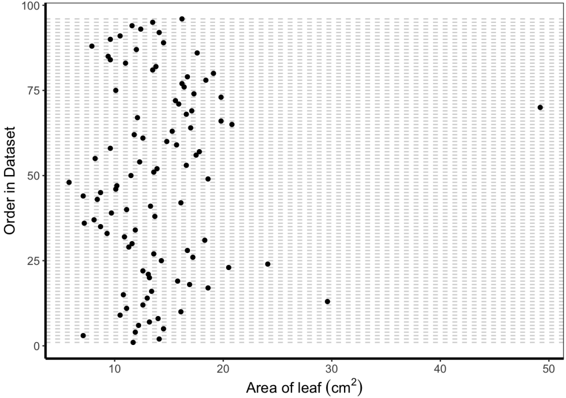
Contour map

```
ggplot(aes(x = leafarea, y = shootarea), data = flower) +
    stat_density_2d(aes(fill = stat(level)), geom = "polygon") +
    labs(y = "Shoot Area", x = "Leaf Area") +
    coord_cartesian(xlim = c(5,25)) +
    theme_rbook()
```



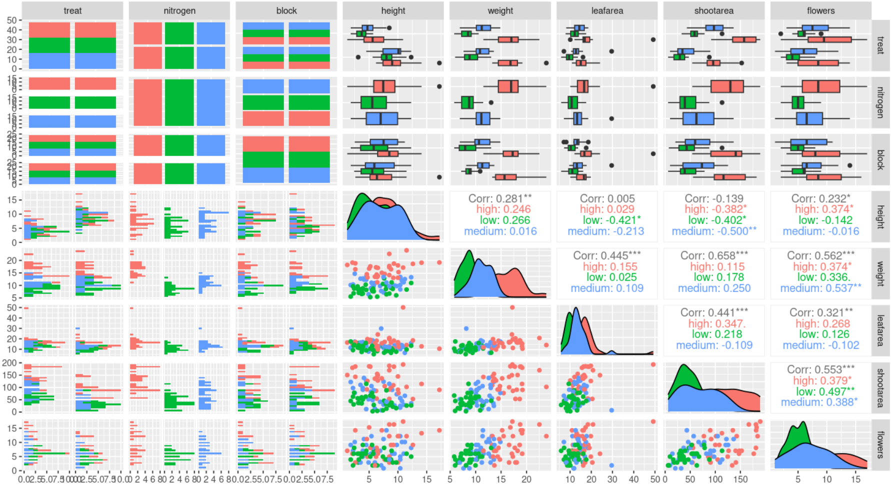
Cleveland dotplot

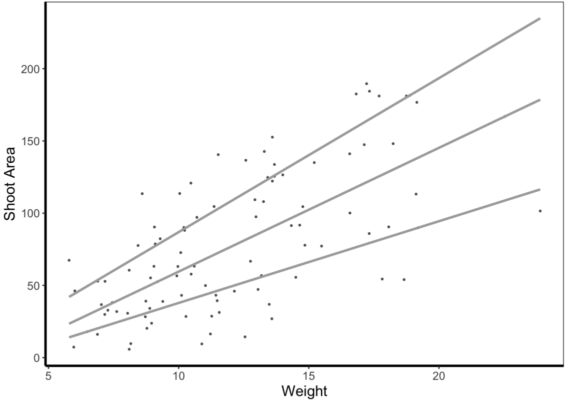
```
ggplot(flower) +
    geom_hline(aes(yintercept = as.numeric(rownames(flower))), linetype = 2,
             colour = "lightgrey") +
  geom_point(aes(x = leafarea, y = as.numeric(rownames(flower)))) +
  labs(y = "Order in Dataset", x = bquote("Area of leaf"~(cm^2))) +
    theme_rbook()
```

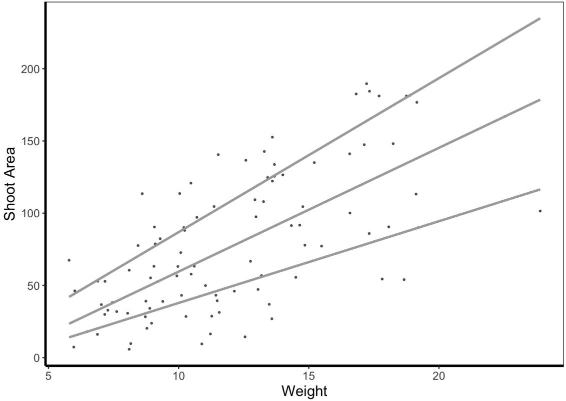


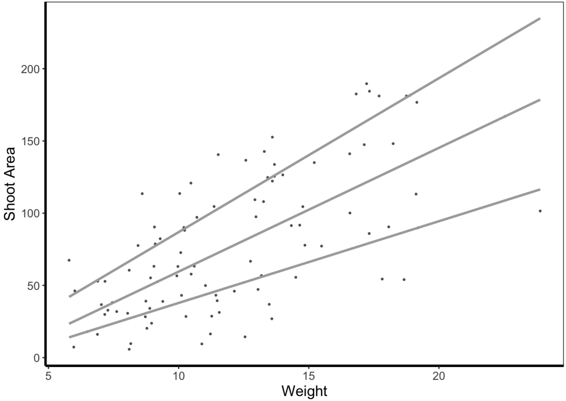
Pairs plot

```
library(GGally)
flower$block <- factor(flower$block)
ggpairs(flower, aes(colour = nitrogen))
```

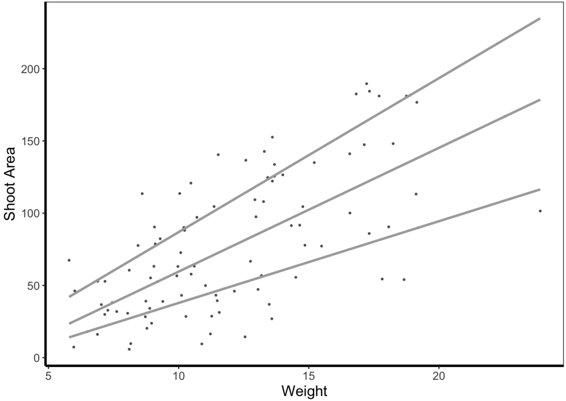


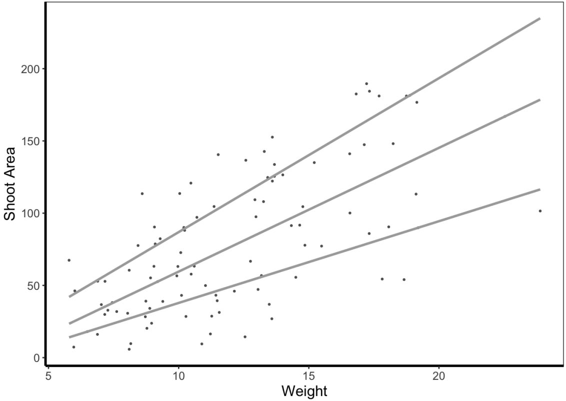

Shoot Area
200
150
100
50
0
5
10
15
20
Weight


Shoot Area
200
150
100
50
0
5
10
15
20
Weight



Shoot Area
200
150
100
50
0
5
10
15
20
Weight


Shoot Area
0
50
100
150
200
5
10
15
20
Weight



Shoot Area
200
150
100
50
0
5
10
15
20
Weight