https://intro2r.com/ Chapter 5

CSCI 297b, Spring 2023

April 26, 2023

R graphics

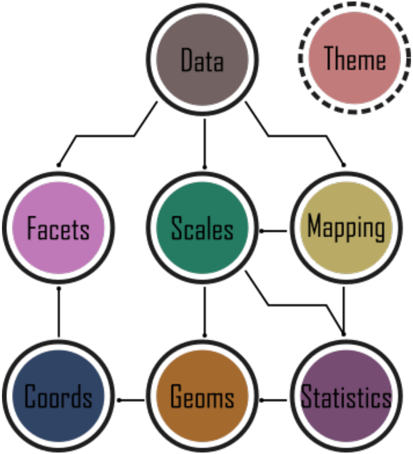
- R provides easy, quick and dirty graphics for work and exploration
- R also provides sophisticated, publication-ready graphics
- R graphics are highly customizable, feeding our creative side
- This enables better expression and communication
- Opposite of “click scatterplot button”

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

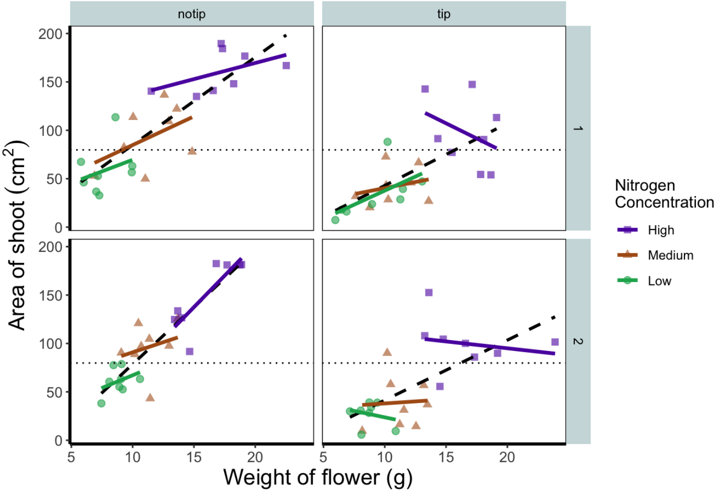
- The user should be in control of all components (i.e. layers) and produce a tailor-made figure fit for their specific needs.



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.

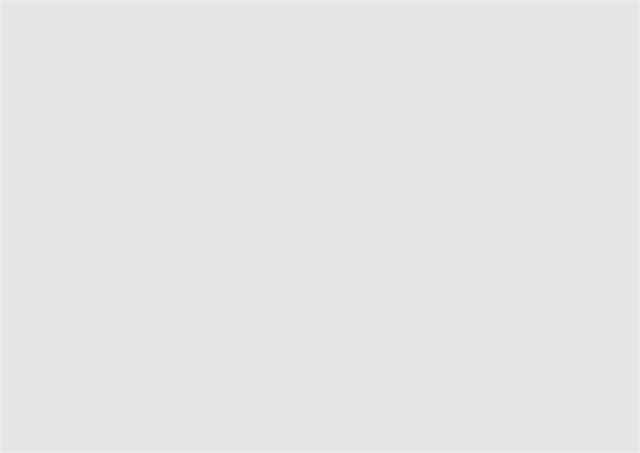
ggplot2 library

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

- Installing the package is not necessary on RStudio Workbench

The base ggplot

```
ggplot()
```



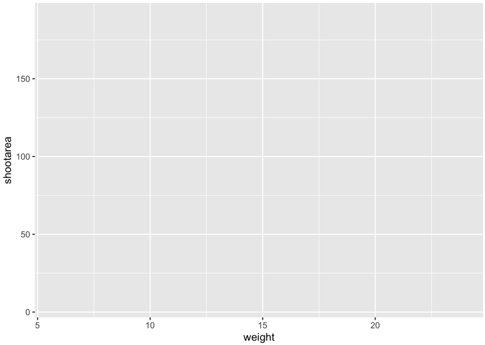
A reminder of the flower data set

```
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)
```

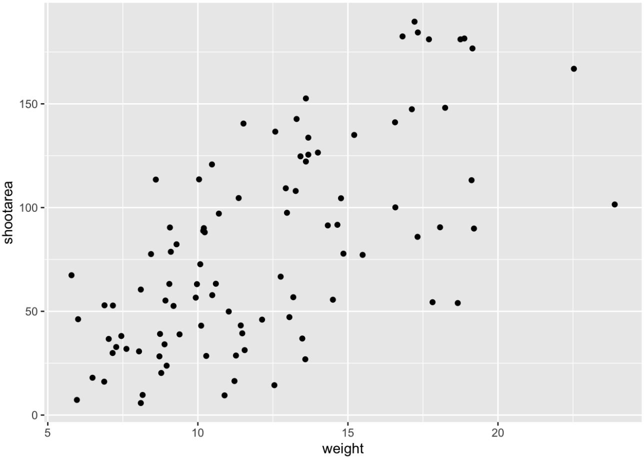


Data

Mapping

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

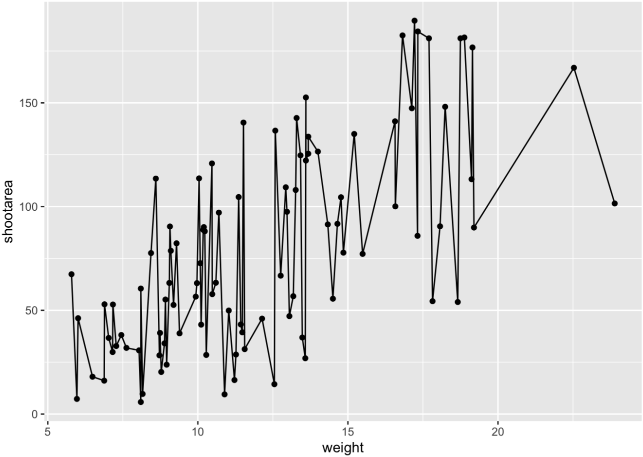
```
ggplot(aes(x = weight, y = shootarea), data = flower) +
        geom_point()     # Adding a geom to display data as point data
```

- There are three essential layers

- Other layers are optional, defaults handle most things.

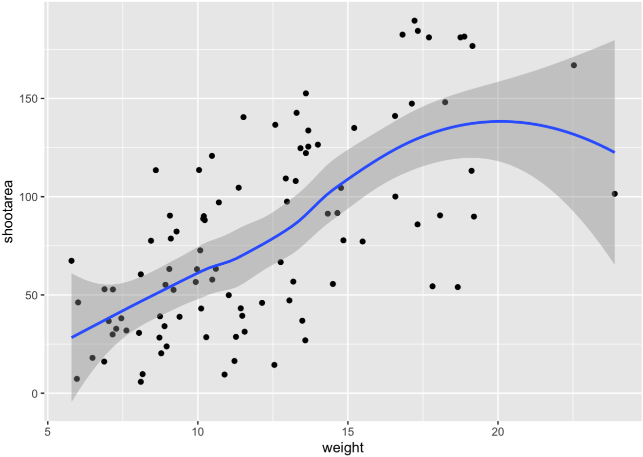
Add line geometry layer

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

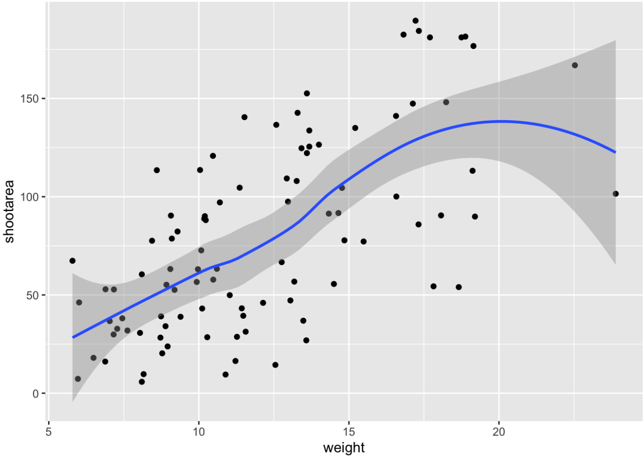


Add smooth geometry layer

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



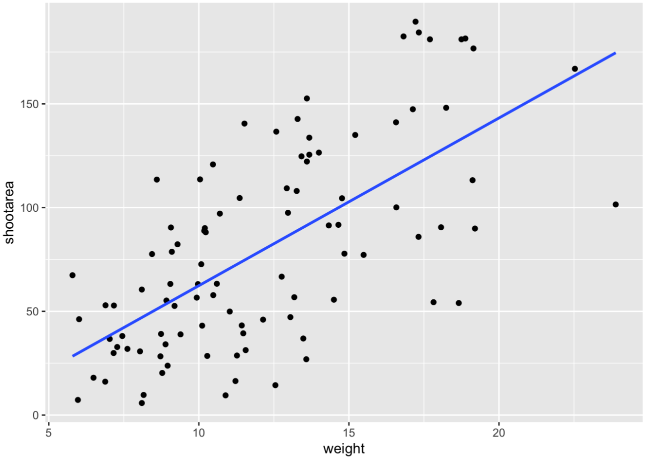
LOESS default



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

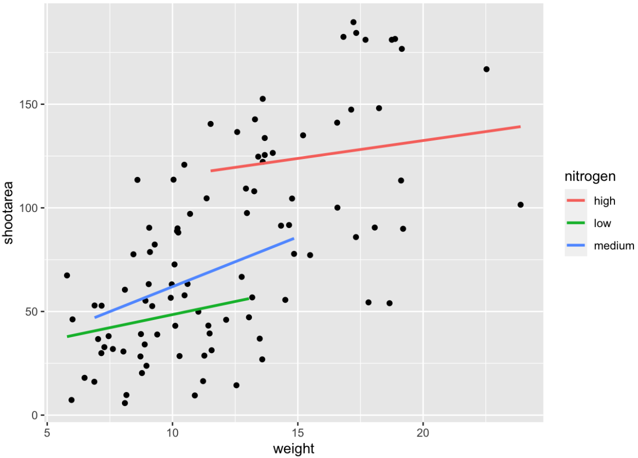
Add linear model line layer

```
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

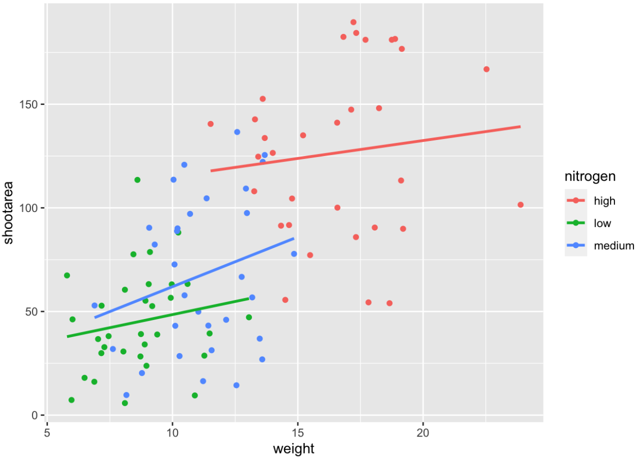
```
ggplot(aes(x = weight, y = shootarea), data = flower) +
  geom_point() +
  # Including colour argument in aes()
  geom_smooth(aes(color = nitrogen), method = "lm", se = FALSE)

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)
```


shootarea
weight
nitrogen
high
low
medium