https://intro2r.com/ Chapter 4

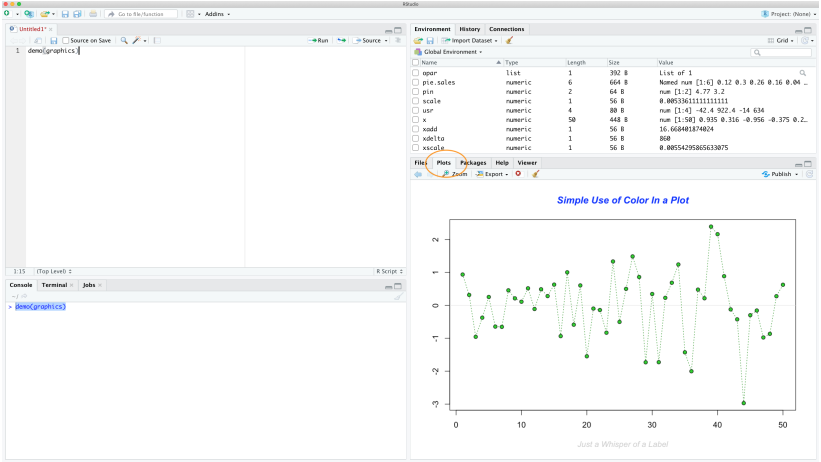
CSCI 297b, Spring 2023

April 25, 2023

Base, lattice, and ggplot2 graphics

- Base graphics: easy, but good style takes work
- Lattice graphics: best with complex multi-dimensional data using panel plots
- Grammar of graphics: logical development, very good defaults

Plot panel in RStudio

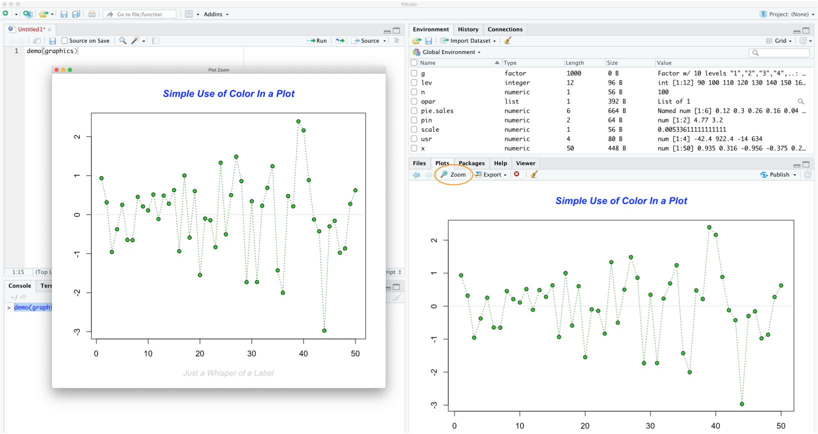


```
demo(graphics)
```

```
> demo(graphics)
```

Name	Type	Length	Size	Value
opar	list	1	392 B	List of 1
pie.sales	numeric	6	664 B	Named num [1:6] 0.12 0.3 0.26 0.16 0.04 …
pin	numeric	2	64 B	num [1:2] 4.77 3.2
scale	numeric	1	56 B	0.00533611111111111
usr	numeric	4	80 B	num [1:4] -42.4 922.4 -14 634
x	numeric	50	448 B	num [1:50] 0.935 0.316 -0.956 -0.375 0.2…
xadd	numeric	1	56 B	16.668401874024
xdelta	numeric	1	56 B	860
xscale	numeric	1	56 B	0.00554295865633075

Plot panel in RStudio, Zoom button

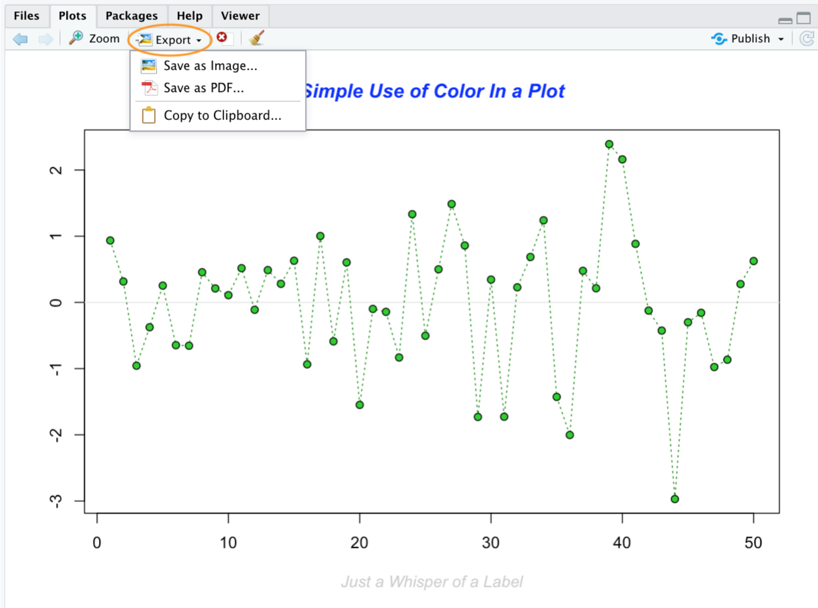


```
1  demo(graphics)
```

```
> demo(graphi
```

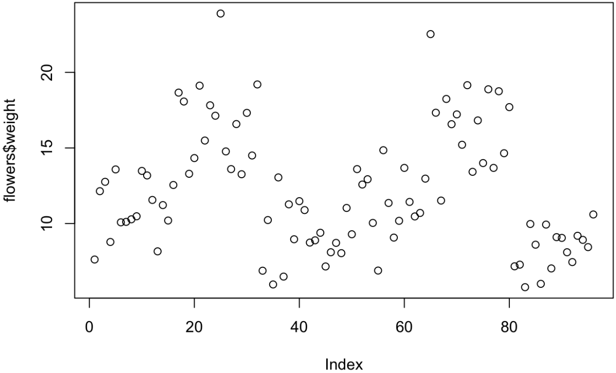
Name	Type	Length	Size	Value
g	factor	1000	0 B	Factor w/ 10 levels "1","2","3","4",..: …
lev	integer	12	96 B	int [1:12] 90 100 110 120 130 140 150 16…
n	numeric	1	56 B	100
opar	list	1	392 B	List of 1
pie.sales	numeric	6	664 B	Named num [1:6] 0.12 0.3 0.26 0.16 0.04 …
pin	numeric	2	64 B	num [1:2] 4.77 3.2
scale	numeric	1	56 B	0.00533611111111111
usr	numeric	4	80 B	num [1:4] -42.4 922.4 -14 634
x	numeric	50	448 B	num [1:50] 0.935 0.316 -0.956 -0.375 0.2…

Plot panel in RStudio, save button



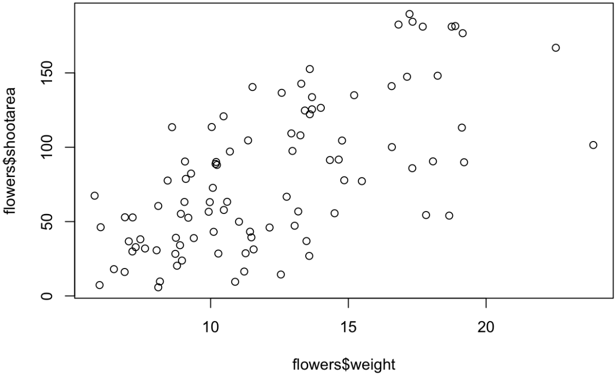
Scatterplots

```
flowers <- read.table(file = 'data/flower.txt',
                       header = TRUE, sep = "\t",
                       stringsAsFactors = TRUE)
plot(flowers$weight)
## or
## with(flowers, plot(weight))
```



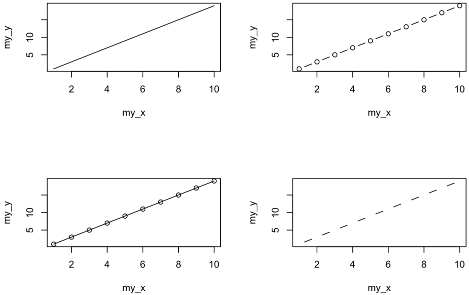
Scatterplots

```
plot(x = flowers$weight, y = flowers$shootarea)
## or
## plot(flowers$shootarea ~ flowers$weight)
```



Scatterplots

```
my_x <- 1:10
my_y <- seq(from = 1, to = 20, by = 2)
par(mfrow = c(2, 2))
plot(my_x, my_y, type = "l")
plot(my_x, my_y, type = "b")
plot(my_x, my_y, type = "o")
plot(my_x, my_y, type = "c")
```

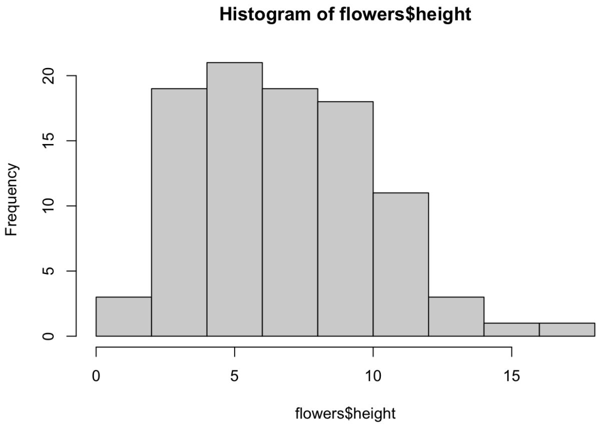


Plot

- `plot` has many options
- Can add more points, lines, text, *etc.*
- `plot` is a generic function: it can change its behavior based on what kind of object it is plotting

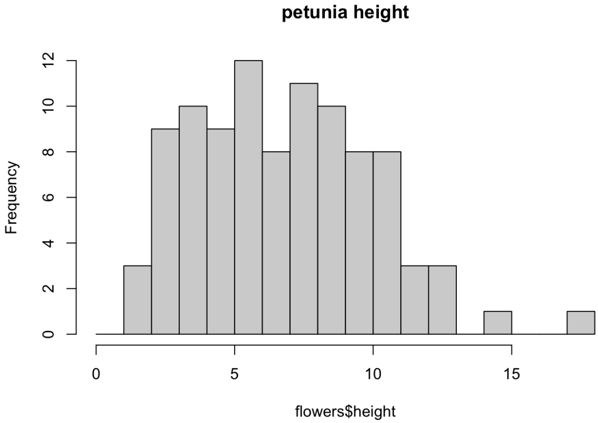
Histograms

```
hist(flowers$height)
```



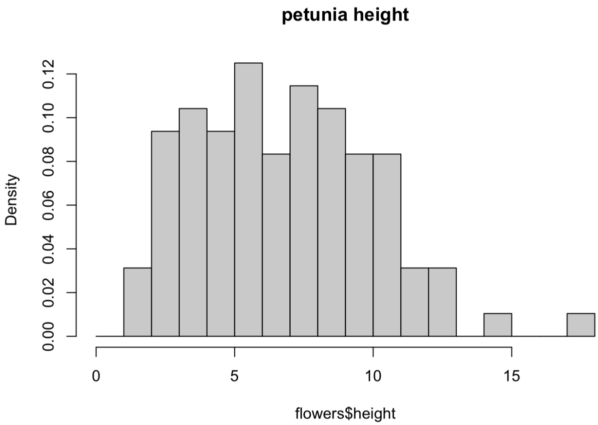
Histograms

```
brk <- seq(from = 0, to = 18, by = 1)
hist(flowers$height, breaks = brk, main = "petunia height")
```



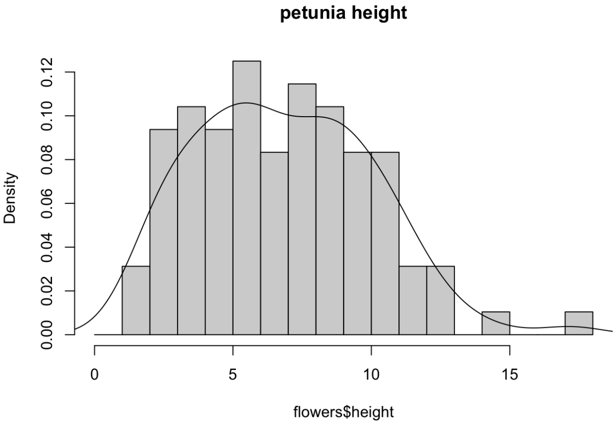
Histograms

```
brk <- seq(from = 0, to = 18, by = 1)
hist(flowers$height, breaks = brk, main = "petunia height",
      freq = FALSE)
```



Histograms

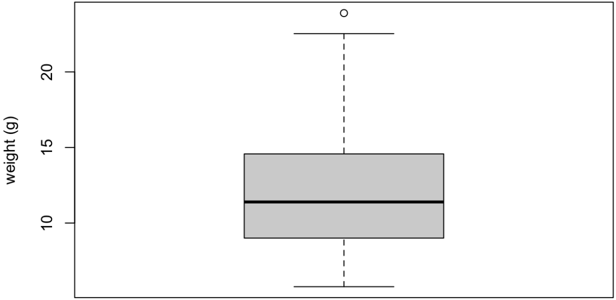
```
dens <- density(flowers$height)
hist(flowers$height, breaks = brk, main = "petunia height",
      freq = FALSE)
lines(dens)
```



Do exercise 4 part 1

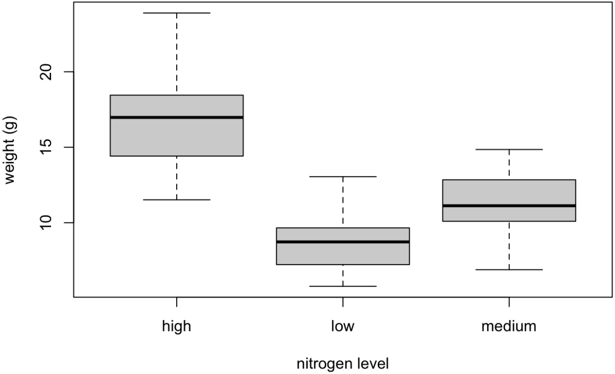
Boxplots

```
boxplot(flowers$weight, ylab = "weight (g)")
```



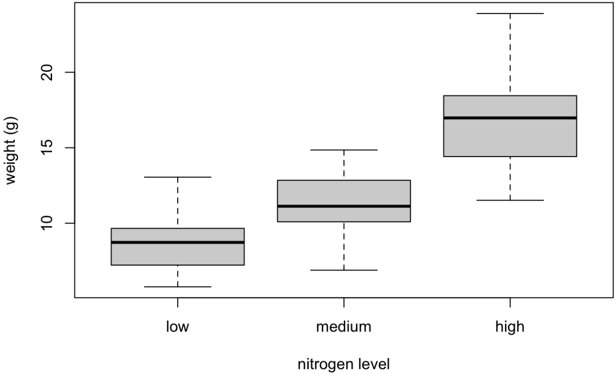
Boxplots

```
boxplot(weight ~ nitrogen, data = flowers,
        ylab = "weight (g)", xlab = "nitrogen level")
```



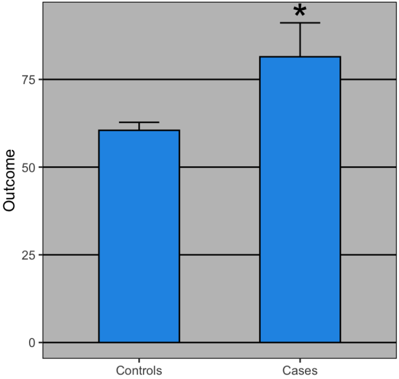
Change order of factor

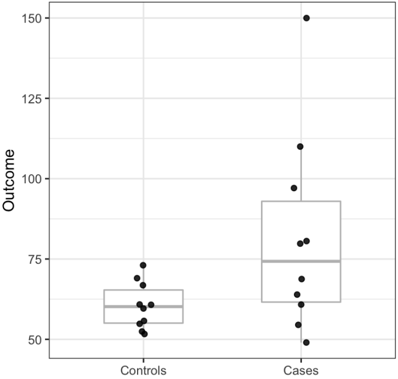
```
flowers$nitrogen <- factor(flowers$nitrogen,
                    levels = c("low", "medium", "high"))
boxplot(weight ~ nitrogen, data = flowers,
         ylab = "weight (g)", xlab = "nitrogen level")
```



Dynamite plots must die

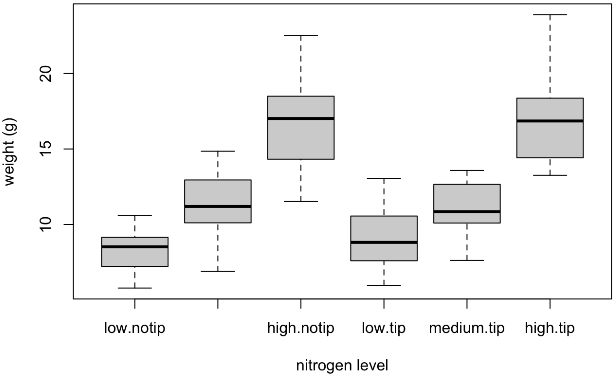
- `https://simplystatistics.org/posts/2019-02-21-dynamite-plots-must-die/`
- Use boxplots in combination with dotplots





Two factors

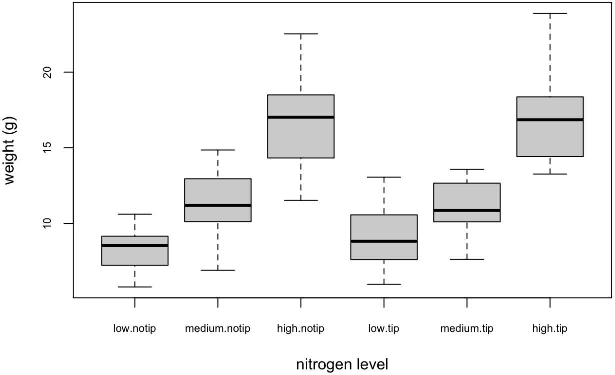
```
boxplot(weight ~ nitrogen * treat, data = flowers,
        ylab = "weight (g)", xlab = "nitrogen level")
```



- Some labels missing

Reduce size of label

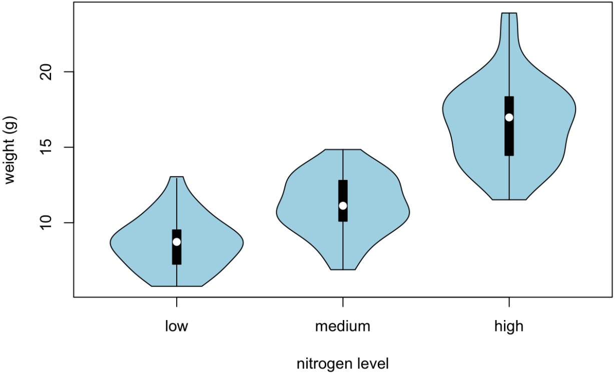
```
boxplot(weight ~ nitrogen * treat, data = flowers,
        ylab = "weight (g)", xlab = "nitrogen level",
        cex.axis = 0.7)
```



Violin plots

```
library(vioplot)
vioplot(weight ~ nitrogen, data = flowers,
         ylab = "weight (g)", xlab = "nitrogen level",
         col = "lightblue")
```

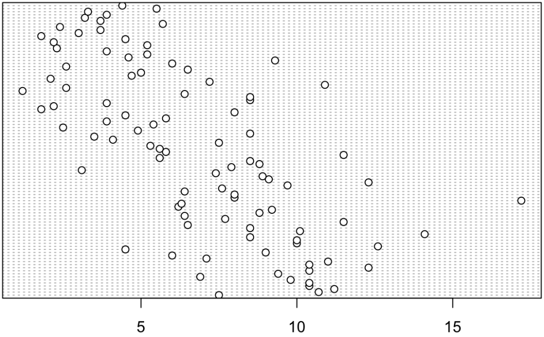
- Kernel density estimate on its side



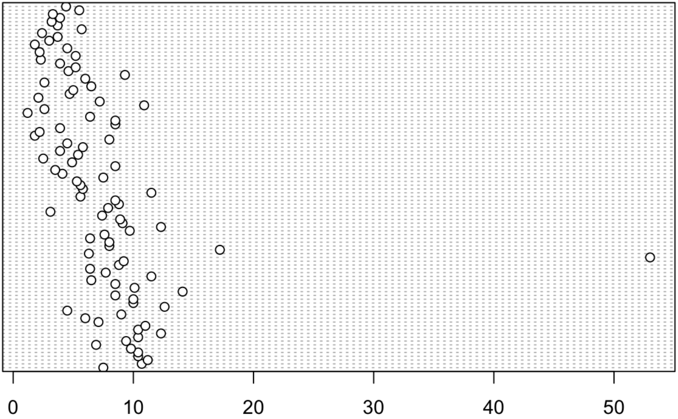
Dot charts

```
dotchart(flowers$height)
```

- x axis is height
- y axis is order in data frame (bottom to top)

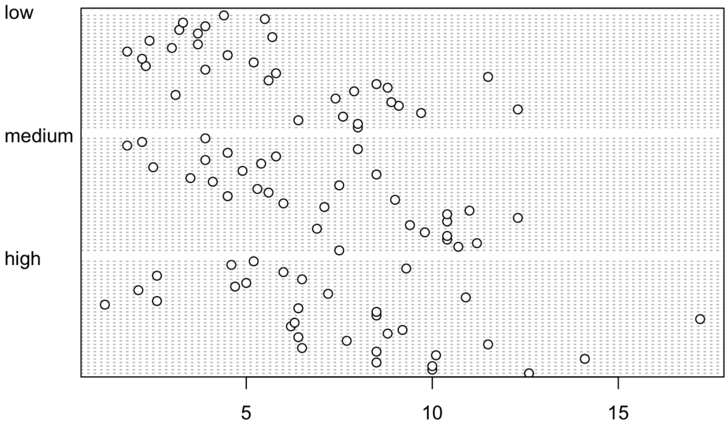


Outliers



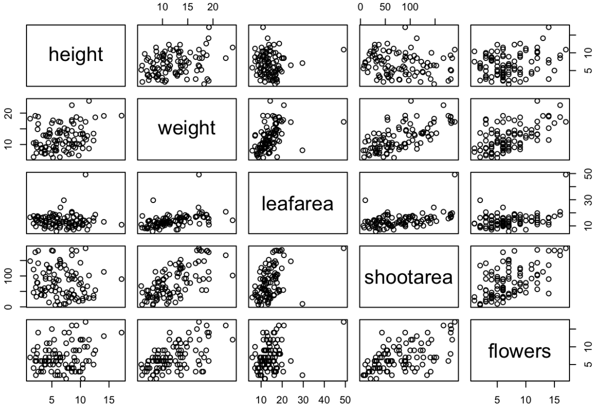
Grouping data

```
dotchart(flowers$height, groups = flowers$nitrogen)
```



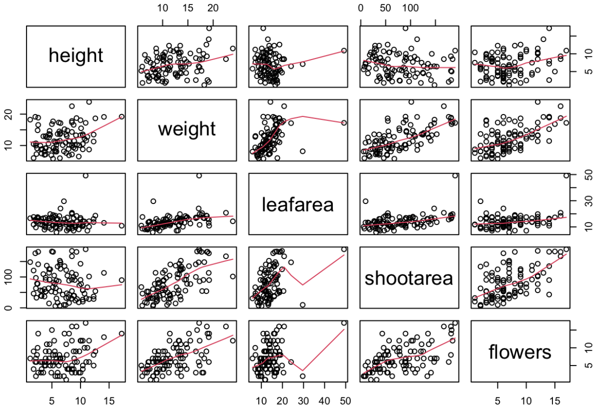
Pairs plots

```
pairs(flowers[, c("height", "weight", "leafarea",
                  "shootarea", "flowers")])
# or we could use the equivalent
# pairs(flowers[, 4:8])
```



Pairs plots, add a LOWESS fit

```
pairs(flowers[, c("height", "weight", "leafarea",
                "shootarea", "flowers")],
                 panel = panel.smooth)
```

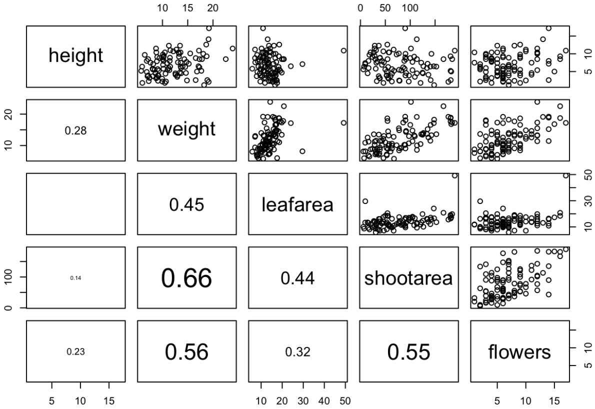


Customized panel function for pairs plots

```
panel.cor <- function(x, y, digits = 2, prefix = "",
                      cex.cor, ...)
{
    usr <- par("usr")
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
```

Panel function in use

```
pairs(flowers[, c("height", "weight", "leafarea",
                  "shootarea", "flowers")],
                   lower.panel = panel.cor)
```

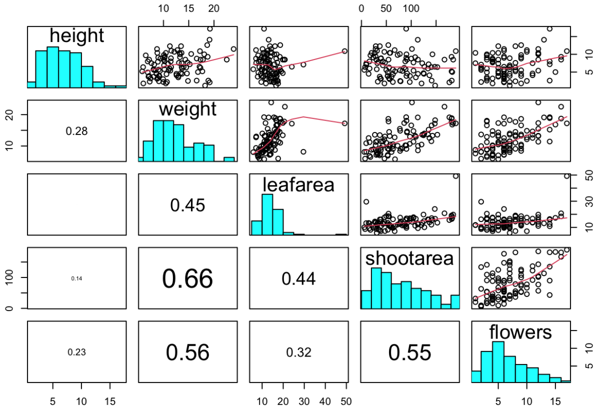


A panel function for the diagonal

```
panel.hist <- function(x, ...)
{
    usr <- par("usr")
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}
```

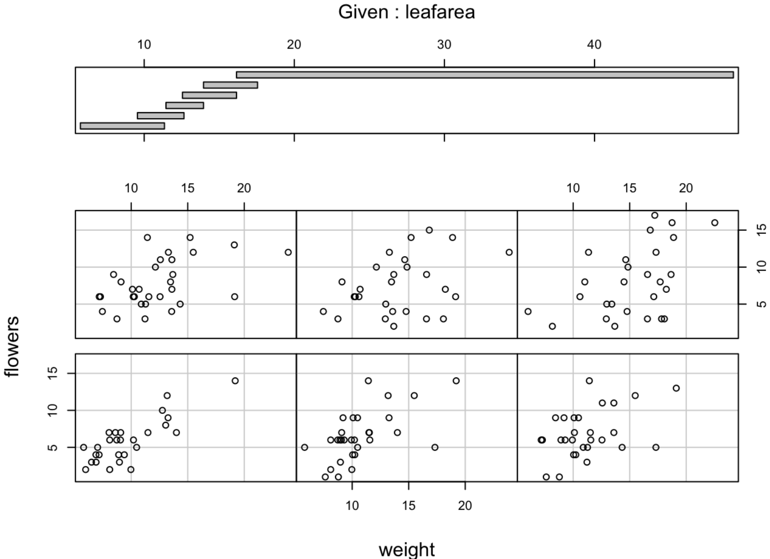
Using 3 panel functions

```
pairs(flowers[, c("height", "weight", "leafarea",
                "shootarea", "flowers")],
                 lower.panel = panel.cor,
                 diag.panel = panel.hist,
                 upper.panel = panel.smooth)
```



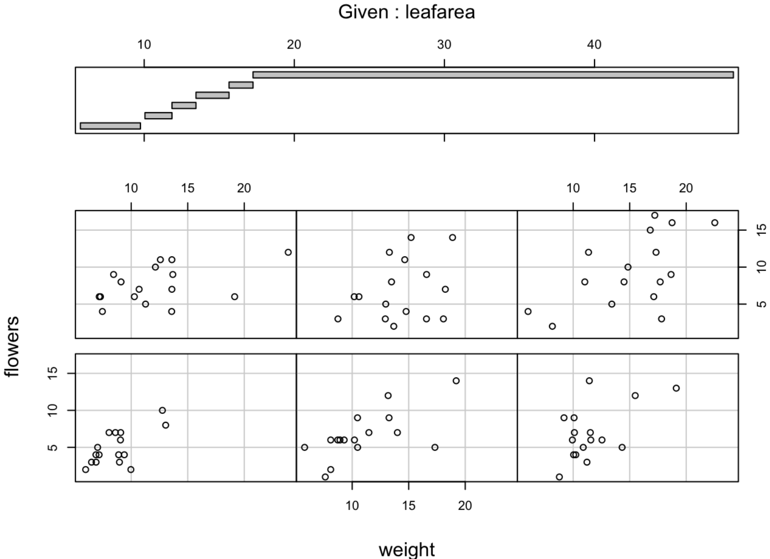
Conditional scatterplot

```
coplot(flowers ~ weight | leafarea, data = flowers)
```



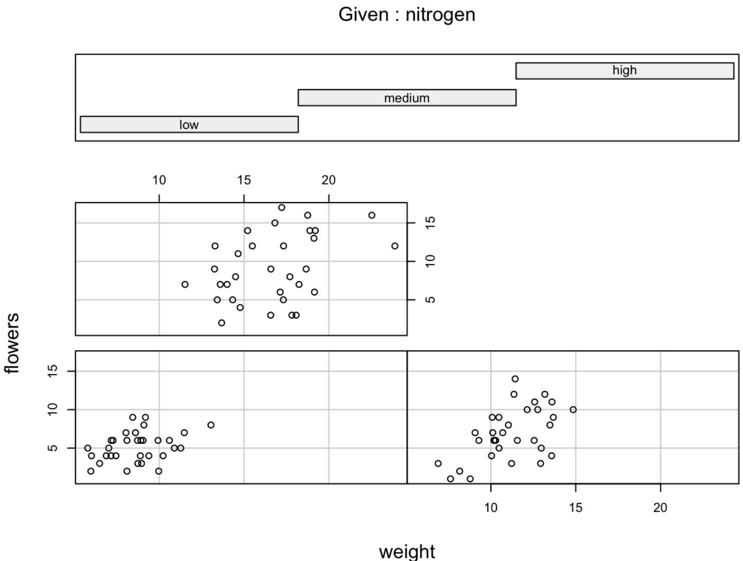
Conditional scatterplot, overlap=0

```
coplot(flowers ~ weight | leafarea, data = flowers, overlap = 0)
```



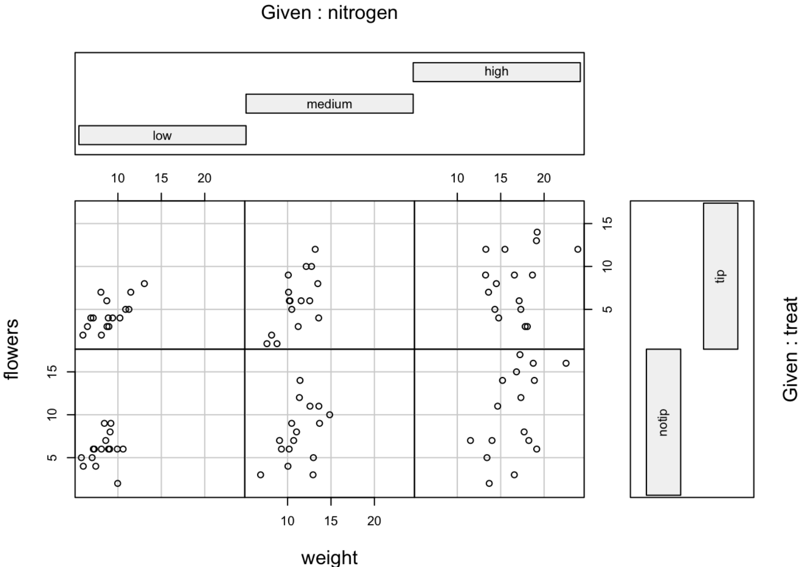
Conditional scatterplot with factors

```
coplot(flowers ~ weight | nitrogen, data = flowers)
```



Conditional scatterplot with two factors

```
coplot(flowers ~ weight | nitrogen * treat, data = flowers)
```

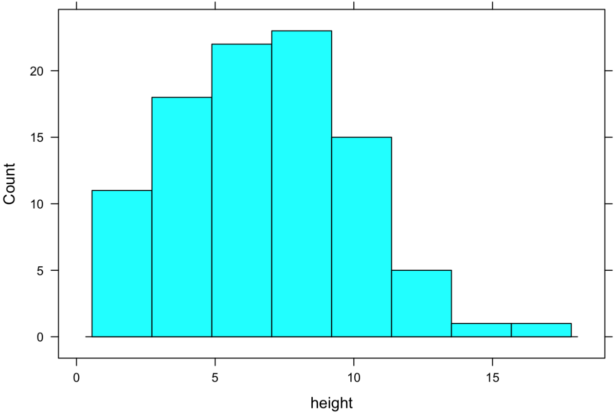


Lattice graphics

- Improved versions of basic graphics
- Use formula notation
- Must load library, `library(lattice)`

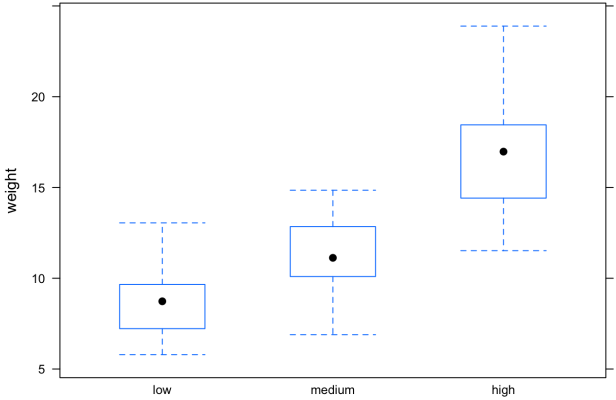
Lattice histogram

```
library(lattice)
histogram(~ height, type = "count", data = flowers)
```



Lattice boxplot (box and whisker plot)

```
bwplot(weight ~ nitrogen, data = flowers)
```

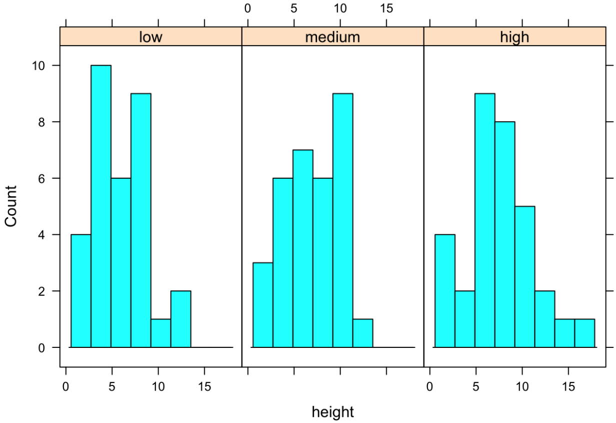


Some lattice functions and their base R equivalents

Graph	Lattice	Base R
scatterplot	xyplot()	plot()
frequency histogram	histogram(type = "count")	hist()
boxplot	bwplot()	boxplot()
Cleveland dotplot	dotplot()	dotchart()
scatterplot matrix	splom()	pairs()
conditioning plot	xyplot(y $\sim$ x $\mid$ z)	coplot()

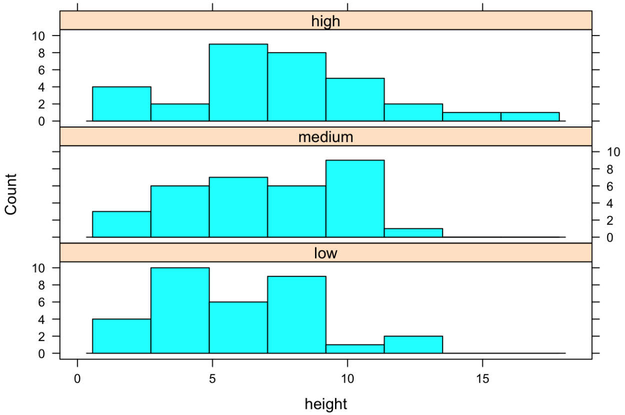
Lattice graphics with multiple panels

```
histogram(~ height | nitrogen, type = "count", data = flowers)
```



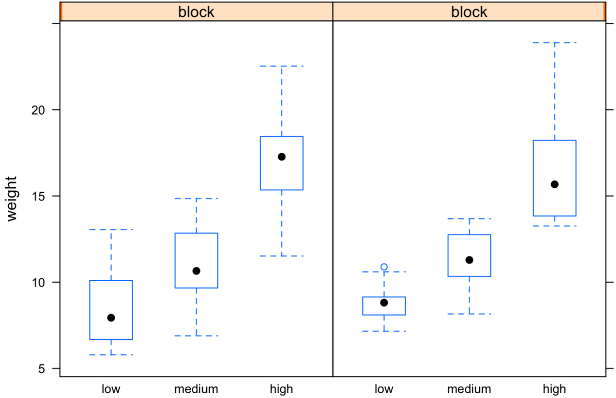
Lattice graphics with multiple panels

```
histogram(~ height | nitrogen, type = "count",
          layout = c(1, 3), data = flowers)
```



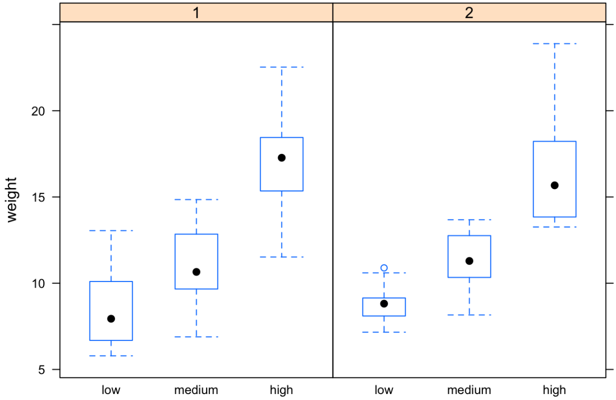
Conditional boxplots

```
bwplot(weight ~ nitrogen | block, data = flowers)
```



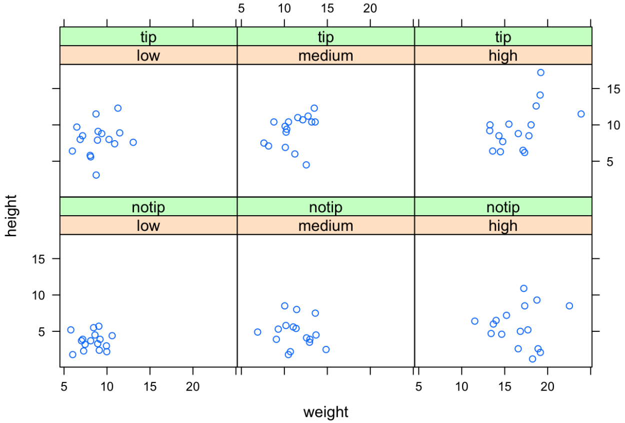
Conditional boxplots, block must be factor

```
bwplot(weight ~ nitrogen | factor(block), data = flowers)
```



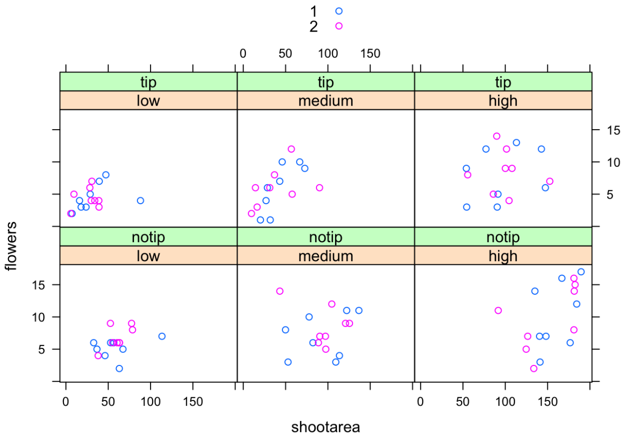
Conditional boxplots

```
xyplot(height ~ weight | nitrogen * treat, data = flowers)
```



Conditional boxplots

```
xyplot(flowers ~ shootarea | nitrogen * treat,
       groups = block, auto.key = TRUE, data = flowers)
```

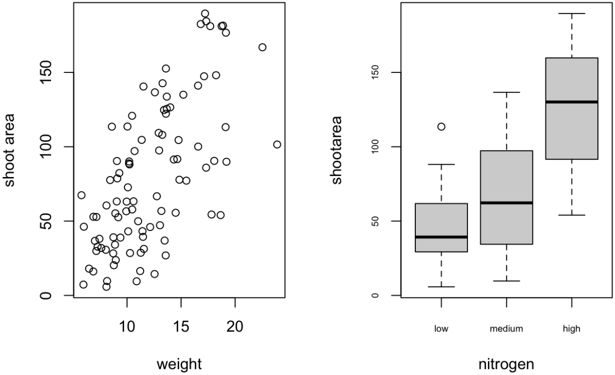


Customizing Base R Graphics

- There are many customizable features.
- Not all customizations are available for all plots.
- Refer to text: `https://intro2r.com/custom_plot.html`

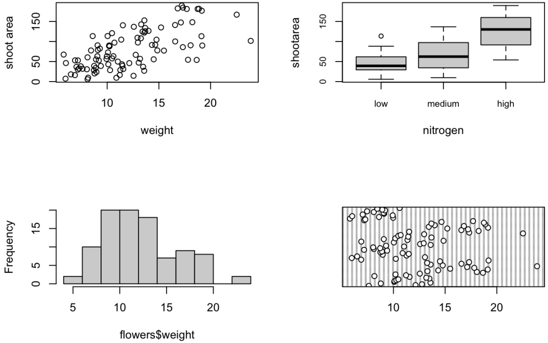
Multiple plots with par(mfrow == ...)

```
par(mfrow = c(1, 2))
plot(flowers$weight, flowers$shootarea, xlab = "weight",
      ylab = "shoot area")
boxplot(shootarea ~ nitrogen, data = flowers, cex.axis = 0.6)
```



Multiple plots with `par(mfrow == ...)`

```
par(mfrow = c(2, 2))
plot(flowers$weight, flowers$shootarea, xlab = "weight",
      ylab = "shoot area")
boxplot(shootarea ~ nitrogen, cex.axis = 0.8, data = flowers)
hist(flowers$weight, main ="")
dotchart(flowers$weight)
```



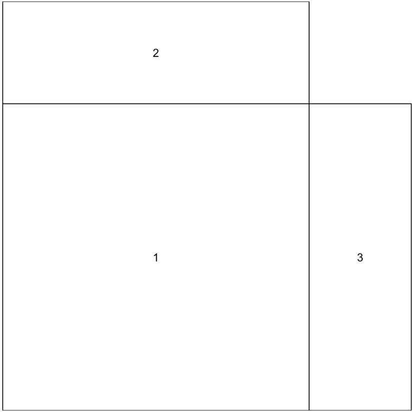
Multiple plots with `layout()`

```
layout_mat <- matrix(c(2, 0, 1, 3), nrow = 2, ncol = 2,
                     byrow = TRUE)
layout_mat
##      [,1] [,2]
## [1,]    2    0
## [2,]    1    3
```

- This gives us two rows and two columns
- The first plot will be lower left
- The second plot will be upper left
- The third plot will be lower right
- Nothing will be upper right

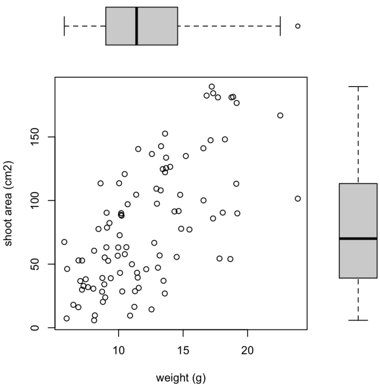
We can specify row and column sizes layout()

```
my_lay <- layout(mat = layout_mat,
                 heights = c(1, 3),
                 widths = c(3, 1), respect =TRUE)
layout.show(my_lay)
```



Adjusting margins generally necessary

```
plot(flowers$weight, flowers$shootarea,
     xlab = "weight (g)", ylab = "shoot area (cm2)")
par(mar = c(0, 4, 0, 0))
boxplot(flowers$weight, horizontal = TRUE, frame = FALSE,
        axes =FALSE)
par(mar = c(4, 0, 0, 0))
boxplot(flowers$shootarea, frame = FALSE, axes = FALSE)
```



Saving plots

- Can use RStudio buttons
- Can also save in code with `pdf`, `png`, `jpeg`, `tiff`, `bmp`

```
pdf("output/myplot.pdf")
plot(...)
dev.off()

png("output/myplot.png")
plot(...)
dev.off()
```

Do exercise 4 part 2