Intro to data visualization with R

Winston Chang
RStudio

March 2015

http://bit.ly/1x4zPql

Getting started

```
install.packages("ggplot2")

# Load ggplot2 in your R session
library(ggplot2)
```

http://bit.ly/1x4zPql

Inspecting data

Built-in datasets

```
faithful          # Show whole data set
head(faithful) # Just the first 6 rows
str(faithful)  # Show structure
```

```
> faithful
    eruptions waiting
1       3.600      79
2       1.800      54
3       3.333      74
4       2.283      62
5       4.533      85
6       2.883      55
7       4.700      88
8       3.600      85
9       1.950      51
10      4.350      85
11      1.833      54
12      3.917      84
13      4.200      78
14      1.750      47
15      4.700      83
16      2.167      52
17      1.750      62
18      4.800      84
19      1.600      52
20      4.250      79
21      1.800      51
22      1.750      47
```

```
> head(faithful)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
6     2.883      55
>
```

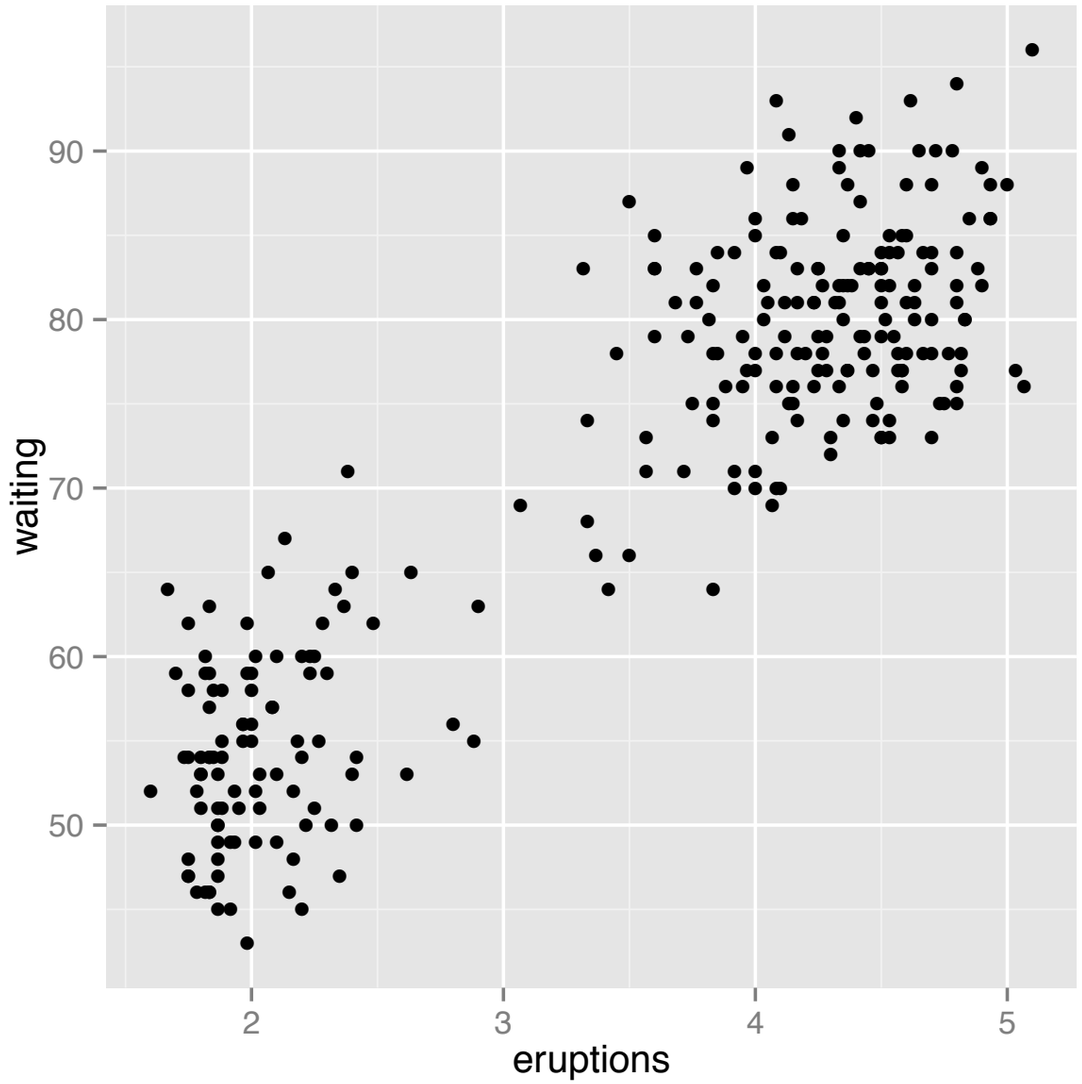
```
> str(faithful)
'data.frame':   272 obs. of  2 variables:
 $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
 $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
>
```

```
> head(mpg)
  manufacturer model displ year cyl      trans drv cty hwy fl   class
1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
```

```
> str(mpg)
'data.frame':   234 obs. of  11 variables:
 $ manufacturer: Factor w/ 15 levels "audi","chevrolet",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ model       : Factor w/ 38 levels "4runner 4wd",..: 2 2 2 2 2 2 2 3 3 3 ...
 $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
 $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
 $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
 $ trans       : Factor w/ 10 levels "auto(av)","auto(l3)",..: 4 9 10 1 4 9 1 9 4 10 ...
 $ drv         : Factor w/ 3 levels "4","f","r": 2 2 2 2 2 2 2 1 1 1 ...
 $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
 $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
 $ fl          : Factor w/ 5 levels "c","d","e","p",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ class       : Factor w/ 7 levels "2seater","compact",..: 2 2 2 2 2 2 2 2 2 2 ...
```

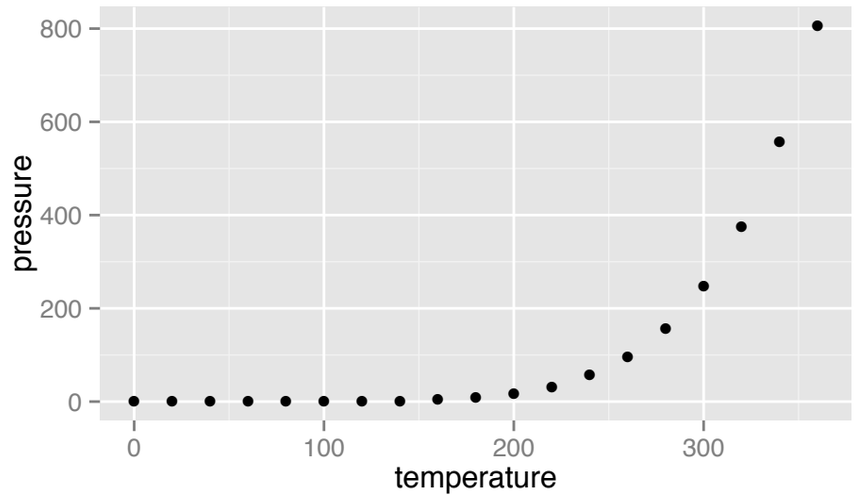
Getting data into R

```
mydata <- read.csv("myfile.csv")

mydata
head(mydata)
str(mydata)
```

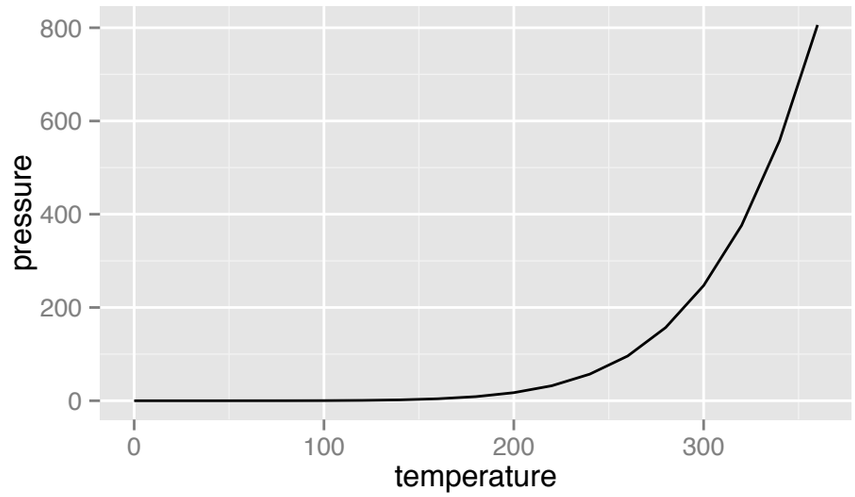



```
ggplot(data=faithful, mapping=aes(x=eruptions, y=waiting)) +
  geom_point()

# More concisely:
ggplot(faithful, aes(x=eruptions, y=waiting)) + geom_point()

qplot(eruptions, waiting, data=faithful)
```

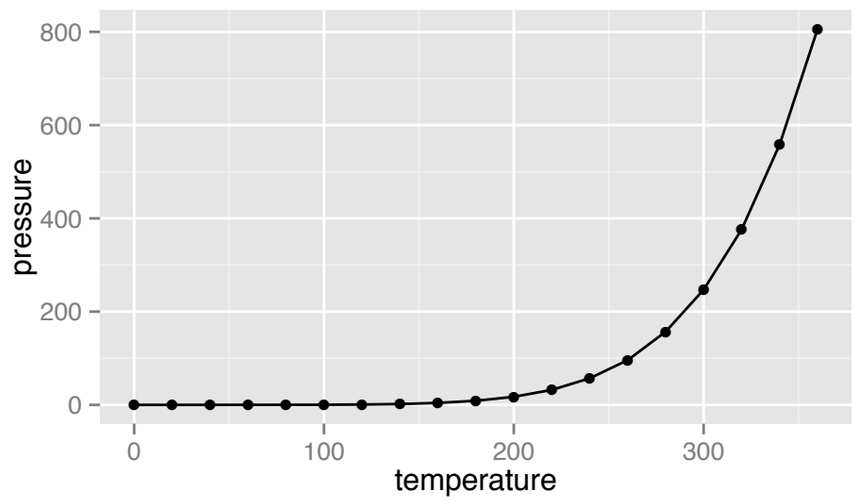
```
p <- ggplot(pressure, aes(x=temperature, y=pressure))

# Points
p + geom_point()
```

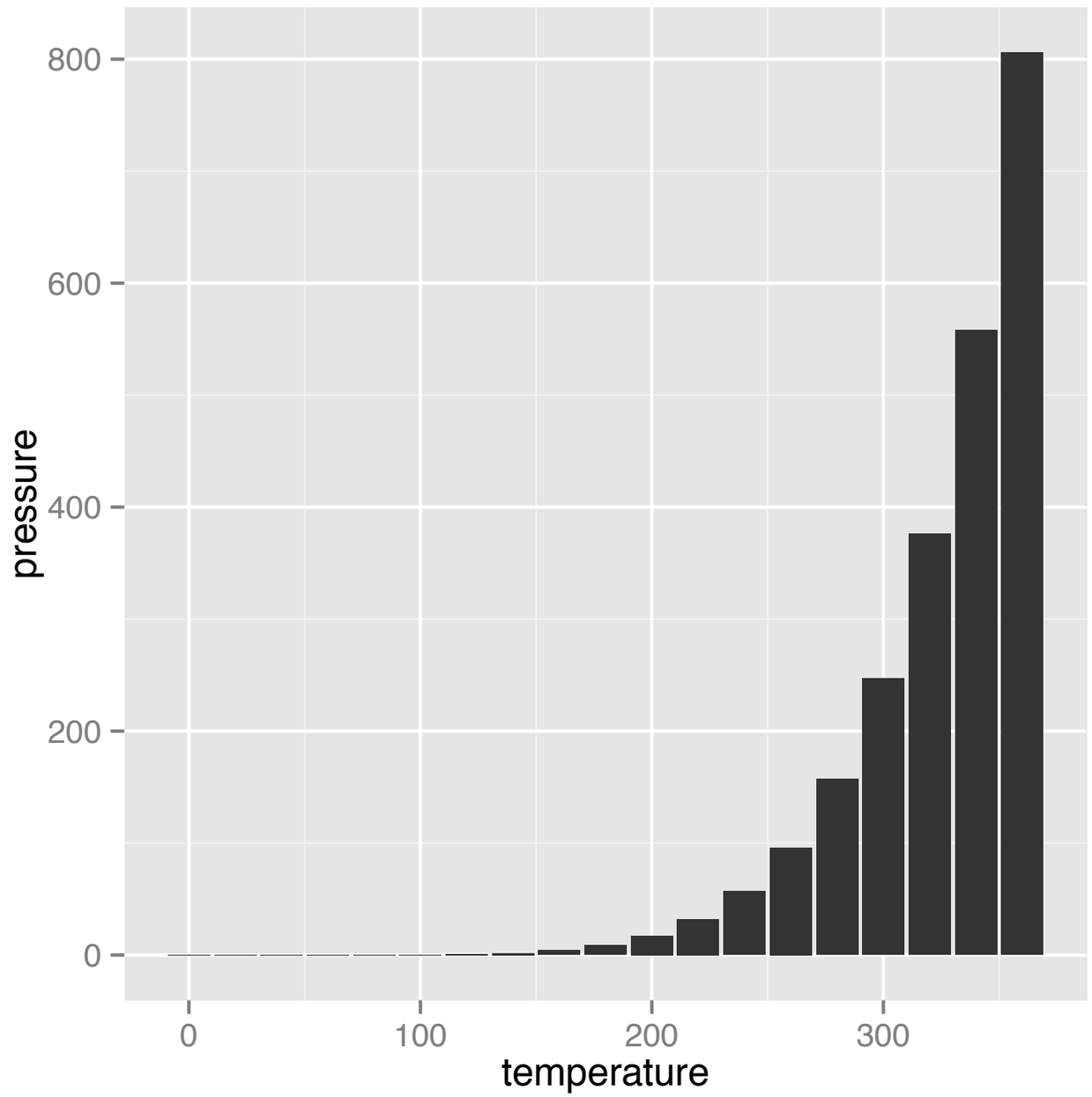



```
# Lines
p + geom_line()
```

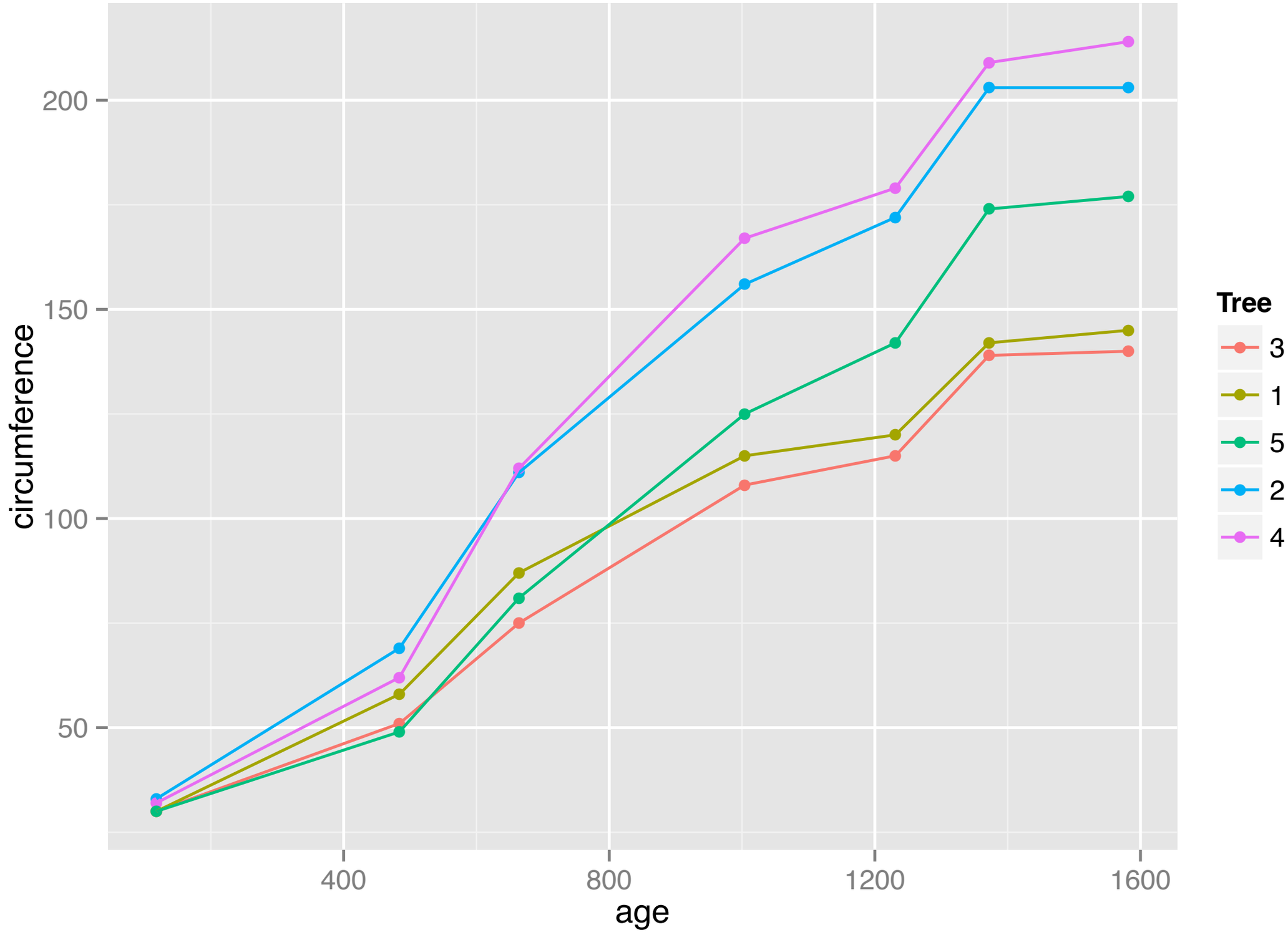



```
# Points with lines
p + geom_line() + geom_point()
```

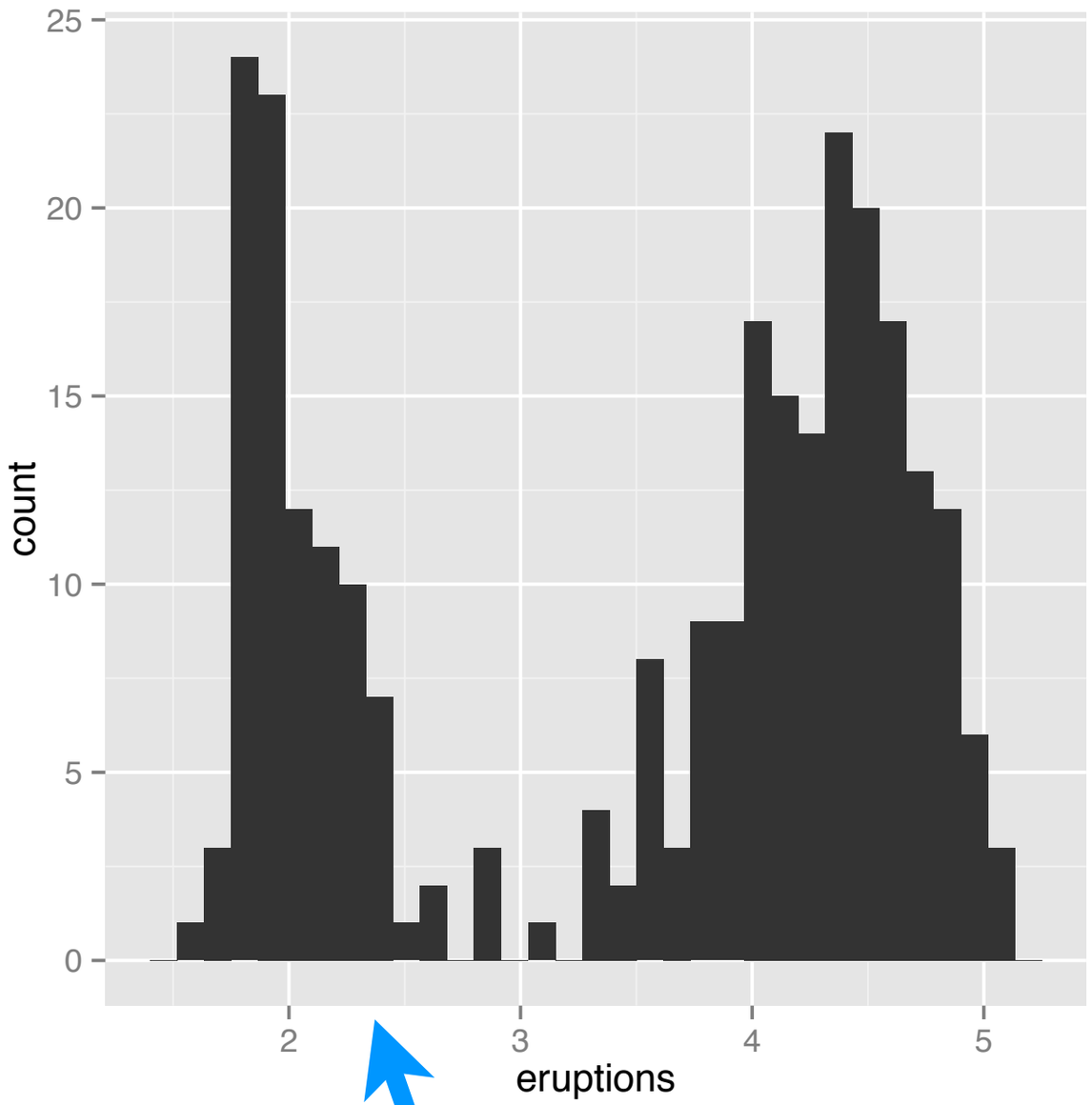


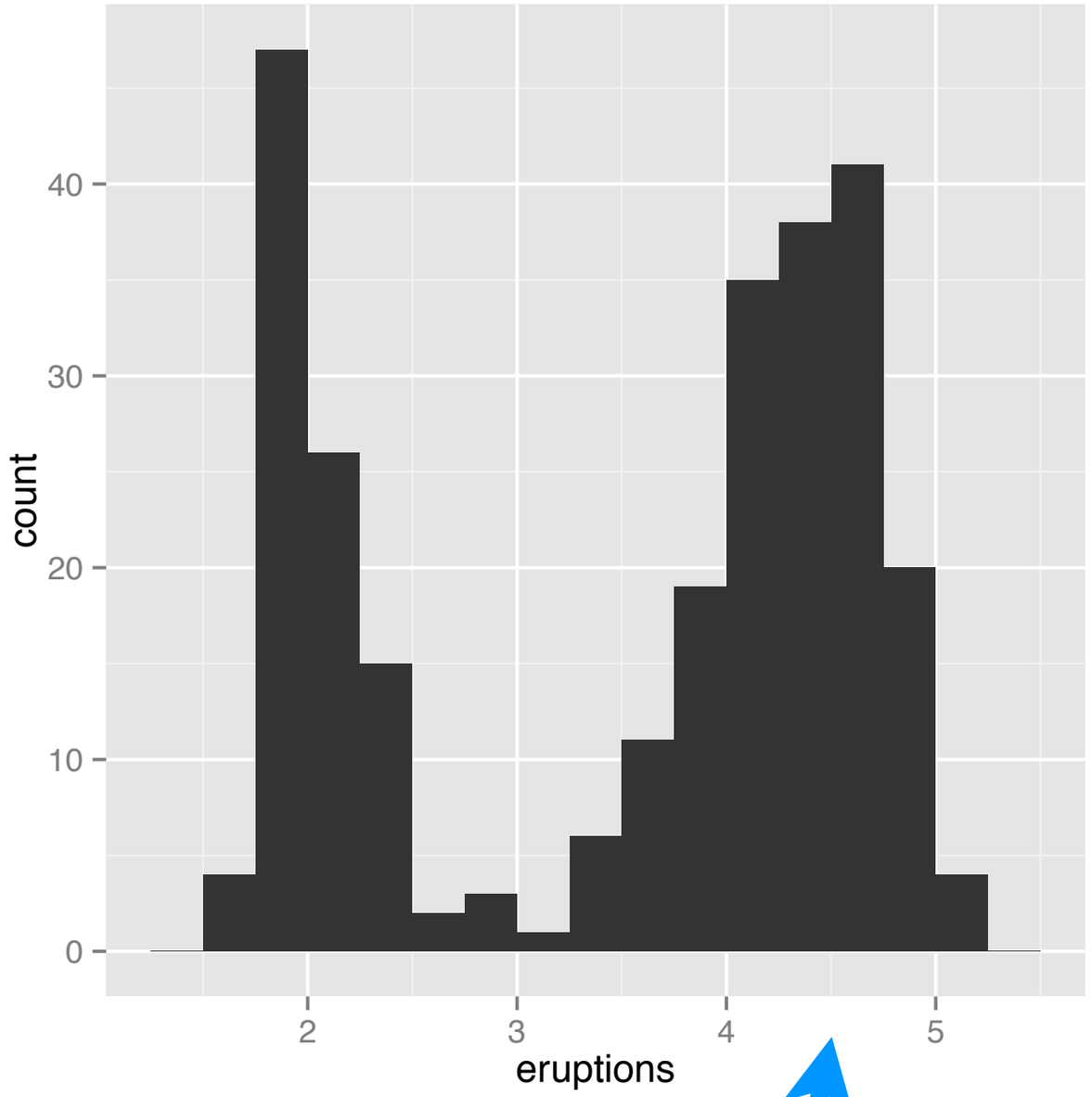



```
p + geom_bar(stat = "identity")
```




```
ggplot(Orange, aes(x=age, y=circumference, colour=Tree)) +
  geom_line() + geom_point()
```



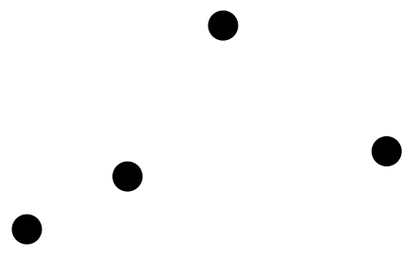



```
ggplot(faithful, aes(x=eruptions)) + geom_histogram()

ggplot(faithful, aes(x=eruptions)) + geom_histogram(binwidth=.25)
```

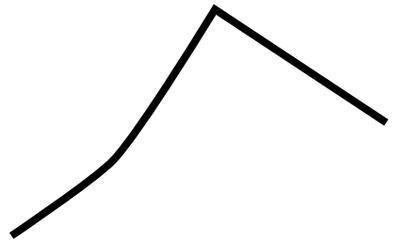
ggplot2 concepts

Geoms

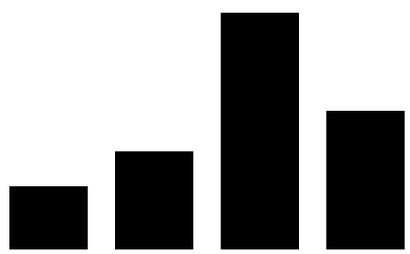
Points



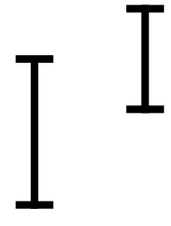
Lines



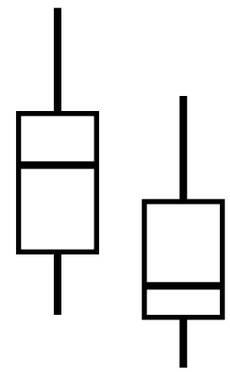
Bars



Error bars



Box plot

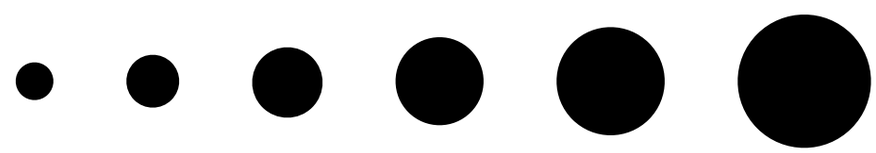


Aesthetics

Y position

Color

Size



X position

Example data

```
> dat <- data.frame(var1 = c(2, 3, 5, 7),
                    var2 = c(2, 4, 8, 5),
                    var3 = c(5, 0, 4, 1))
```

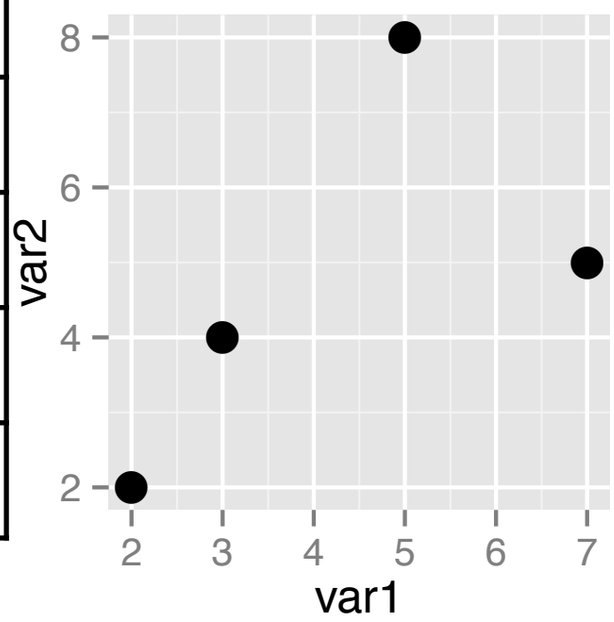
```
> dat
  var1 var2 var3
1    2    2    5
2    3    4    0
3    5    8    4
4    7    5    1
```

```
> dat2 <- data.frame(var1 = c("A", "B", "A", "B", "A", "B"),
                     var2 = c("G1", "G0", "G2", "G1", "G0", "G2"),
                     var3 = c(5, 0, 4, 1, 6, 3))
```

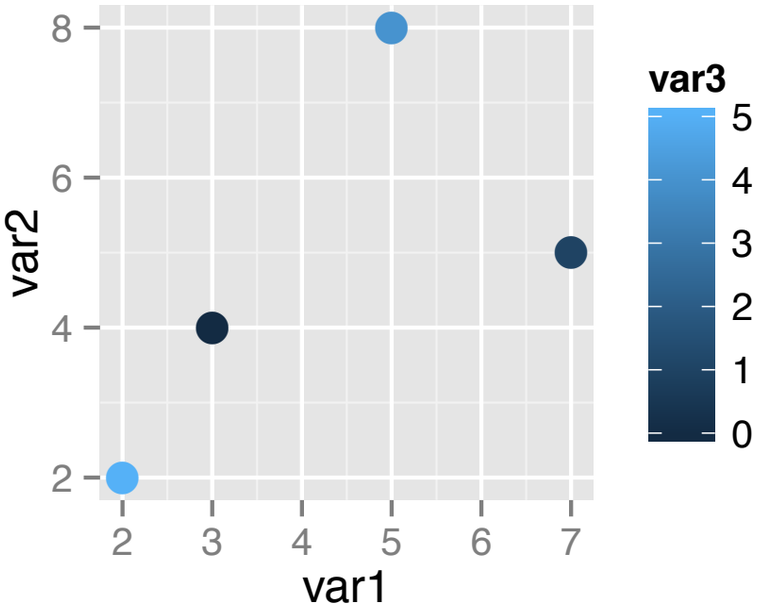
```
> dat2
  var1 var2 var3
1    A   G1    5
2    B   G0    0
3    A   G2    4
4    B   G1    1
5    A   G0    6
6    B   G2    3
```

Mapping data to aesthetics

var1	var2	var3
2	2	5
3	4	0
5	8	4
7	5	1



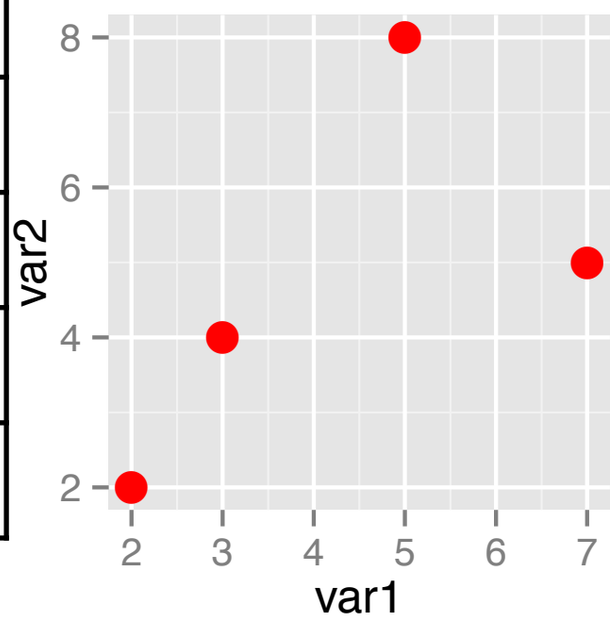

```
> ggplot(dat, aes(x=var1,
  y=var2)) +
  geom_point()
```



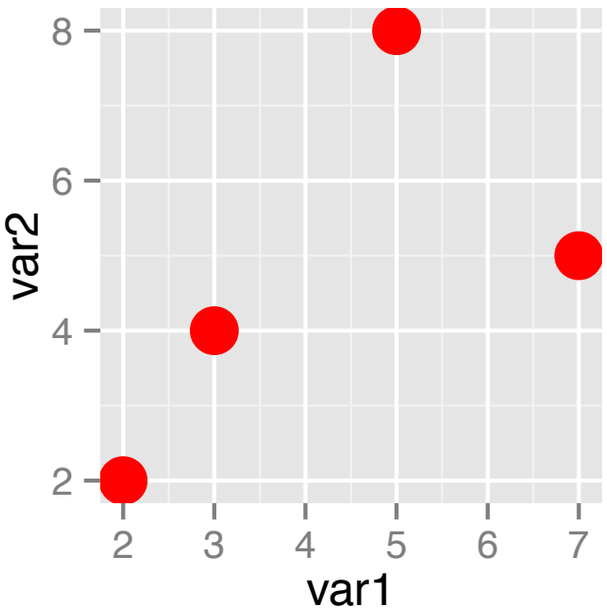

```
> ggplot(dat, aes(x=var1,
  y=var2, colour=var3)) +
  geom_point()
```

Setting aesthetics

var1	var2	var3
2	2	5
3	4	0
5	8	4
7	5	1

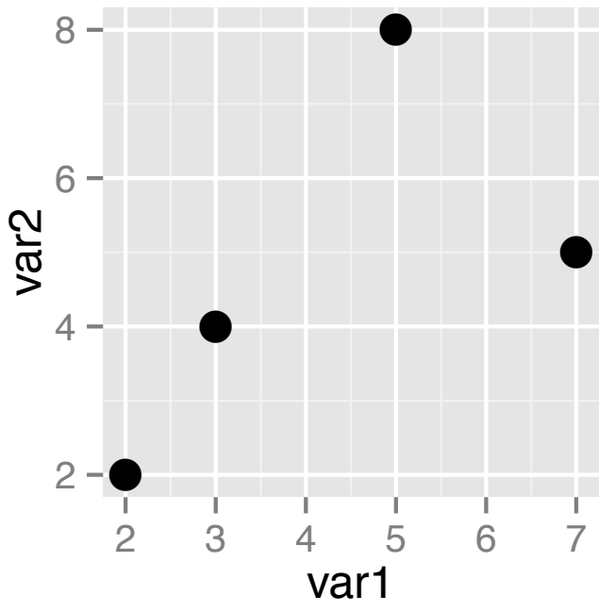



```
> ggplot(dat, aes(x=var1,
    y=var2)) +
    geom_point(colour="red")
```



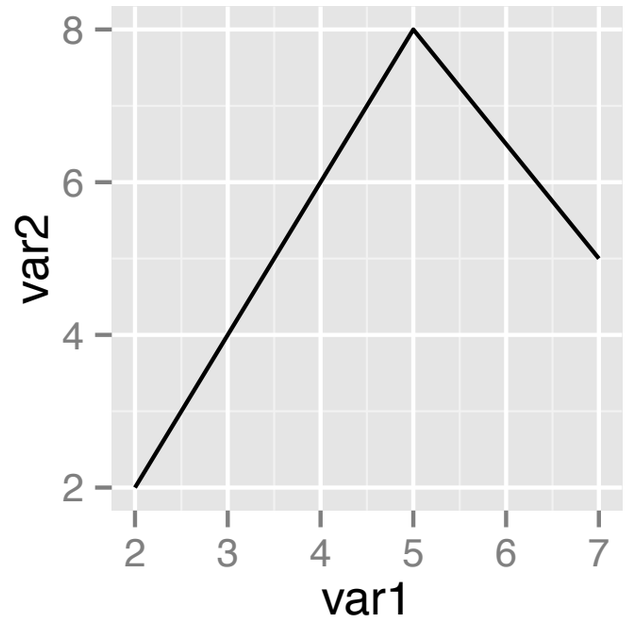

```
> ggplot(dat, aes(x=var1, y=var2)) +
    geom_point(colour="red", size=6)
```

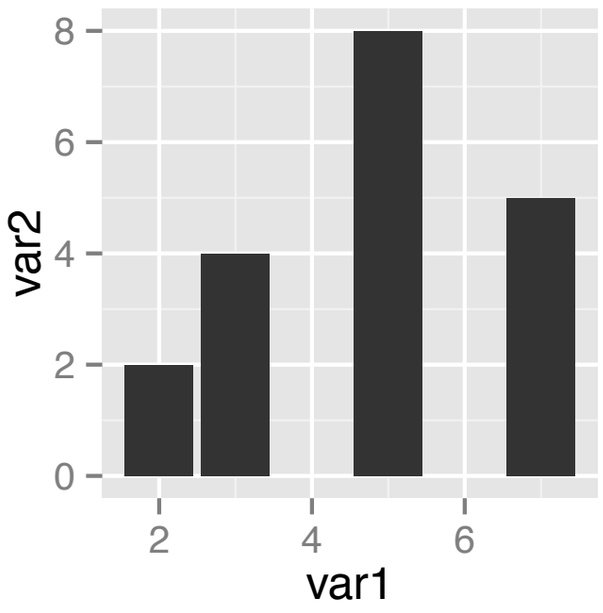
Different geoms




```
> ggplot(dat, aes(x=var1, y=var2)) +
  geom_point()
```

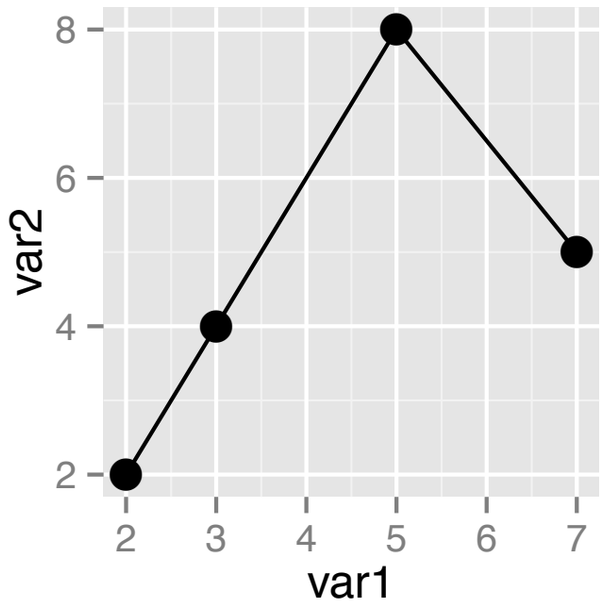
```
> ggplot(dat, aes(x=var1, y=var2)) +
    geom_line()
```






```
> ggplot(dat, aes(x=var1, y=var2)) +
  geom_bar(stat="identity")
```

Using multiple geoms



Default data

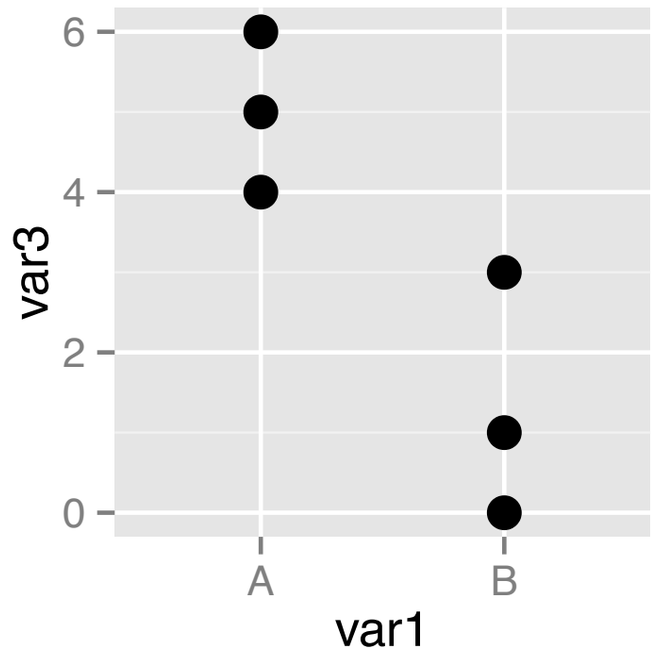
Default mapping

Override defaults in each geom

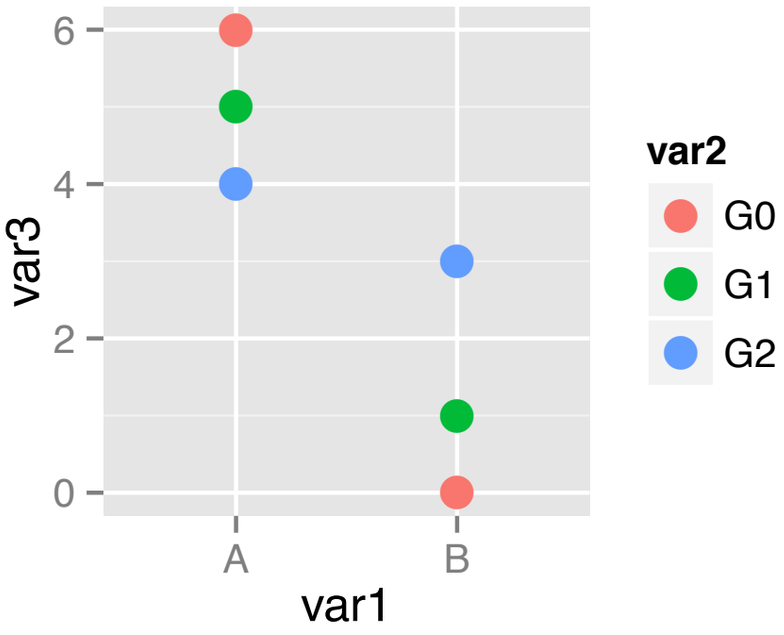
```
> ggplot(dat, aes(x=var1, y=var2)) +
    geom_point() + geom_line()

# Equivalent to
> ggplot(dat) +
    geom_point(aes(x=var1, y=var2)) +
    geom_line(aes(x=var1, y=var2))

> ggplot() +
   geom_point(aes(x=var1, y=var2), data=dat) +
   geom_line(aes(x=var1, y=var2), data=dat)
```

Mapping discrete variables

var1	var2	var3
A	G1	5
B	G0	0
A	G2	4
B	G1	1
A	G0	6
B	G2	3

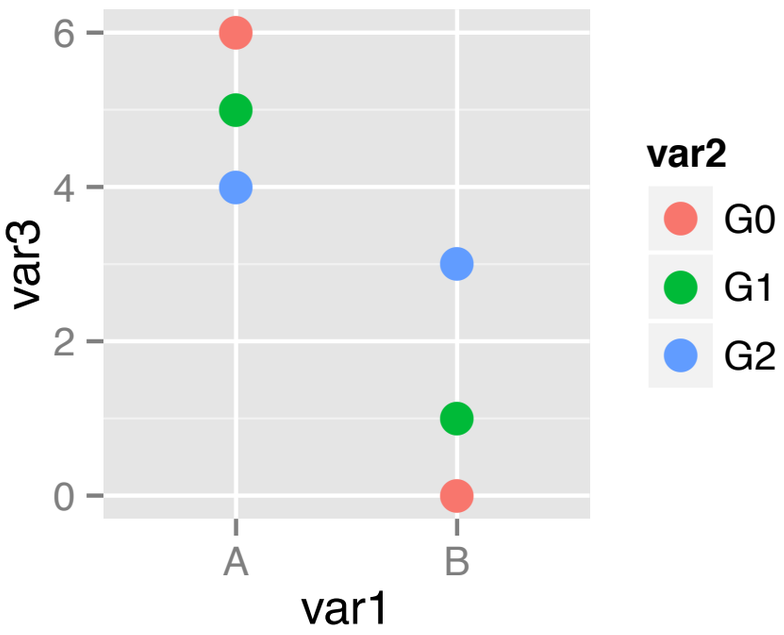



```
> ggplot(dat2, aes(x=var1,
   y=var3)) +
   geom_point()
```




```
> ggplot(dat2, aes(x=var1, y=var3,
   colour=var2)) + geom_point()
```

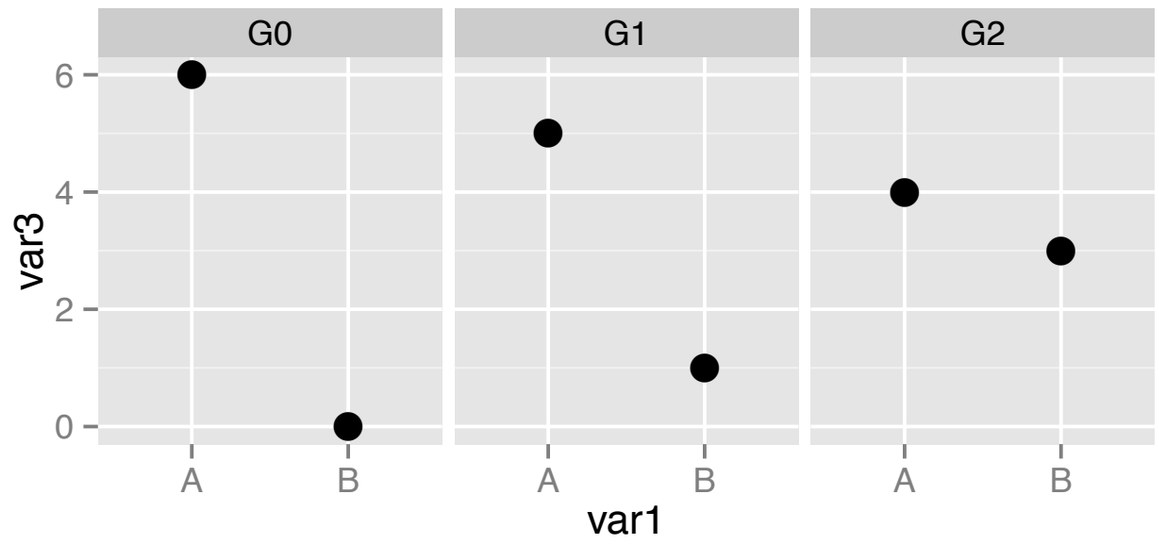
Facets




```
> ggplot(dat2, aes(x=var1, y=var3,
    colour=var2)) + geom_point()
```

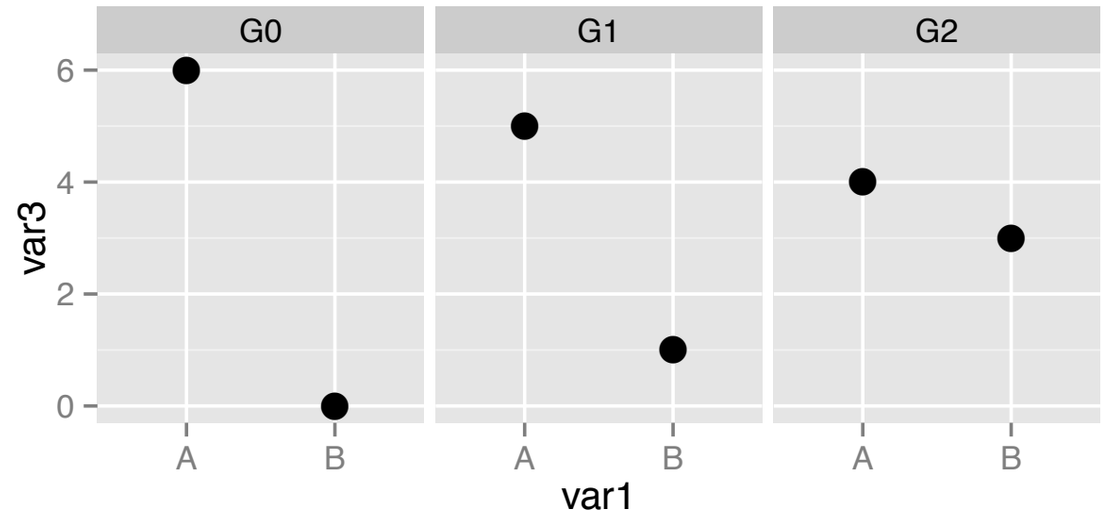
var1	var2	var3
A	G1	5
B	G0	0
A	G2	4
B	G1	1
A	G0	6
B	G2	3

```
> ggplot(dat2, aes(x=var1, y=var3)) +
    geom_point() +
    facet_wrap( ~ var2)
```

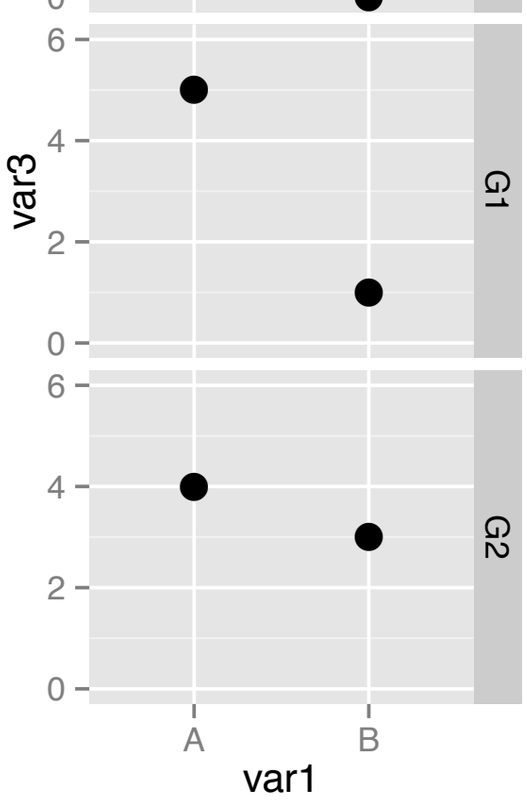


Facets

```
> ggplot(dat2, aes(x=var1, y=var3)) +
    geom_point() + facet_grid(. ~ var2)
```



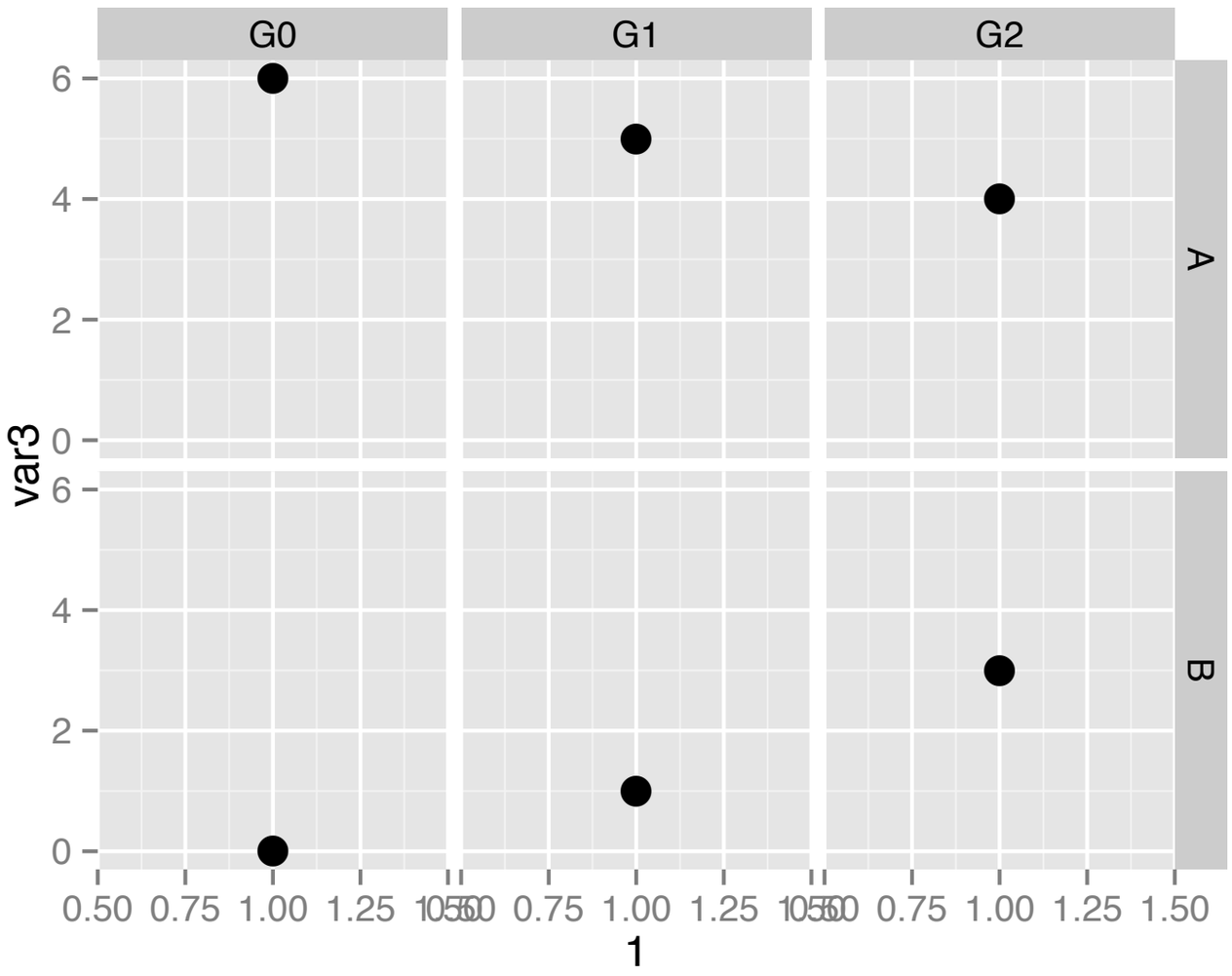
var1	var2	var3
A	G1	5
B	G0	0
A	G2	4
B	G1	1
A	G0	6
B	G2	3

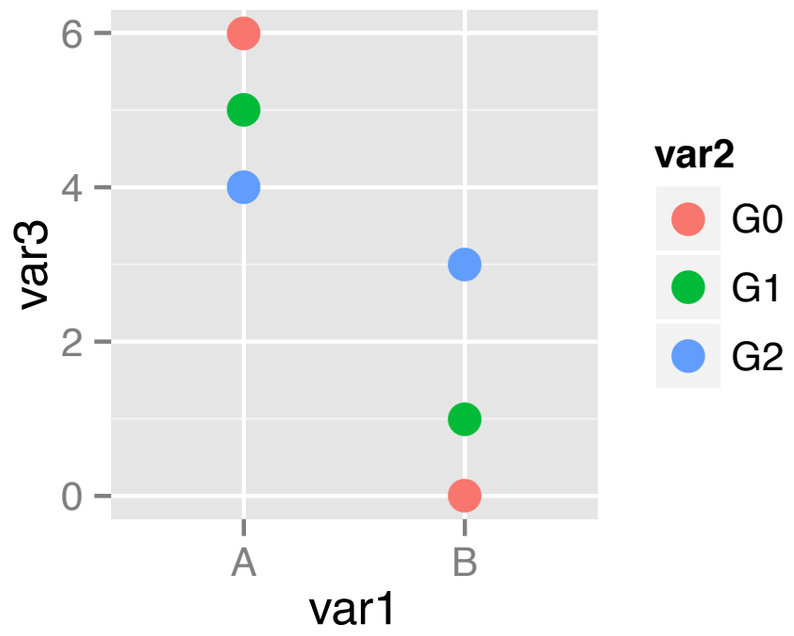



```
> ggplot(dat2, aes(x=var1, y=var3)) +
    geom_point() + facet_grid(var2 ~ .)
```

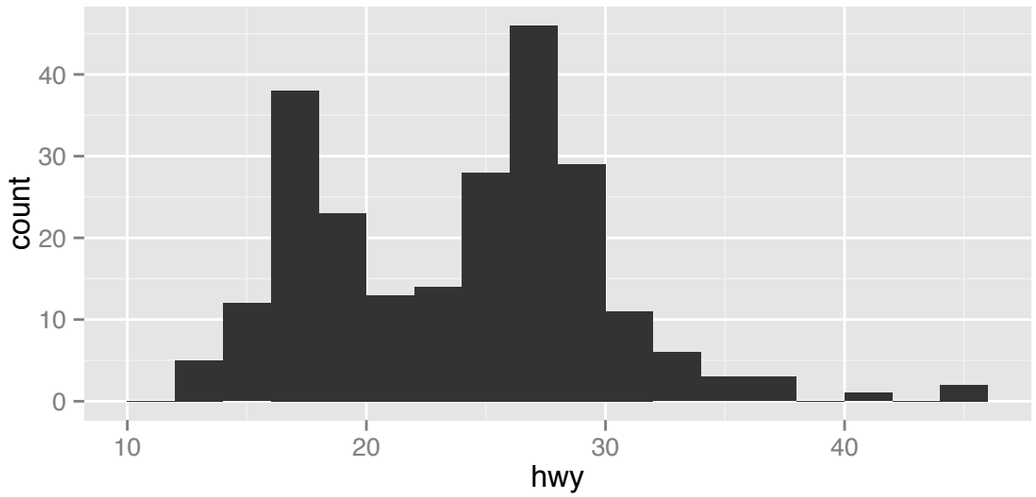
Facets

```
> ggplot(dat2, aes(x=1, y=var3)) + geom_point() +
  facet_grid(var1 ~ var2)
```

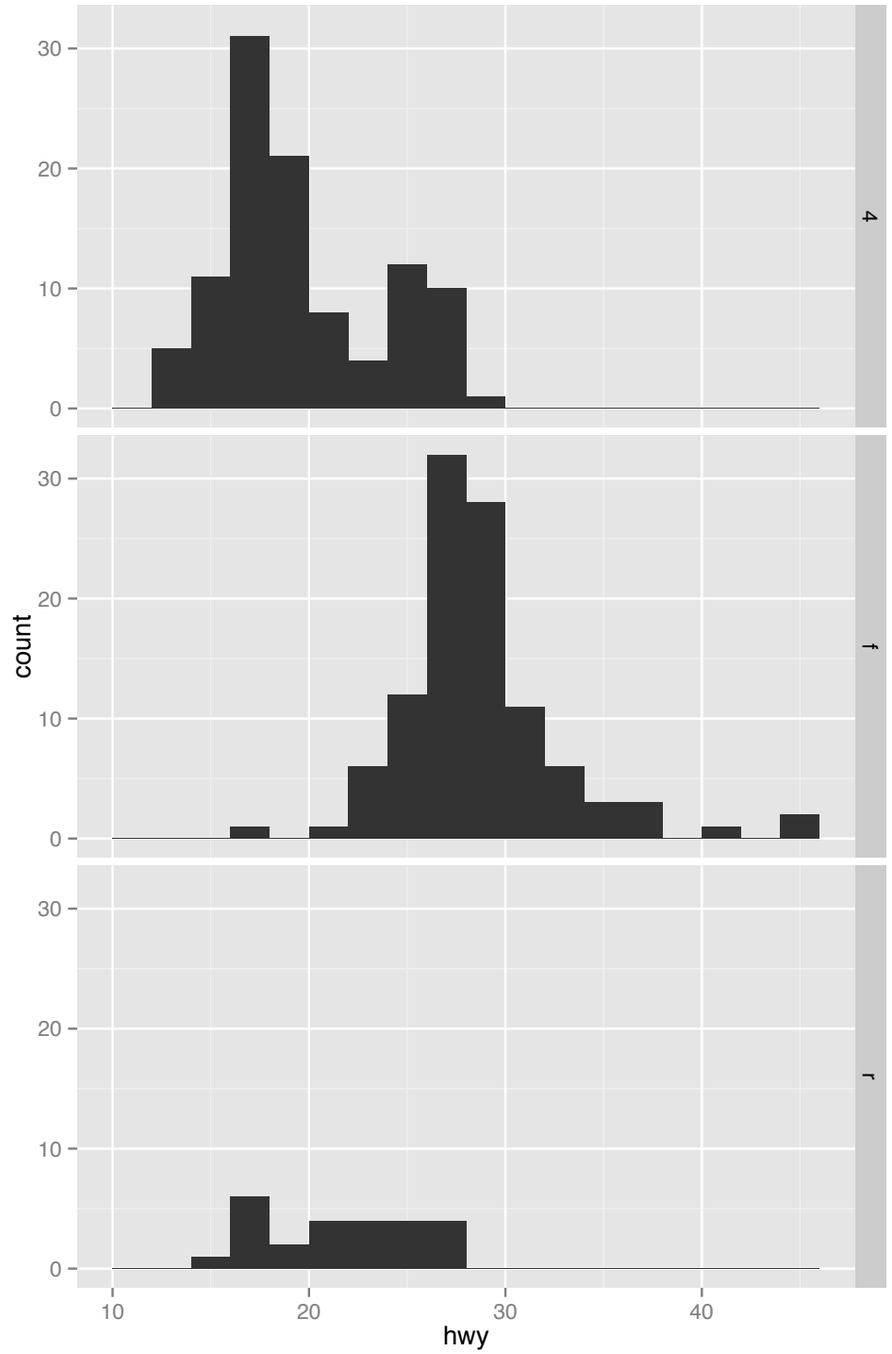




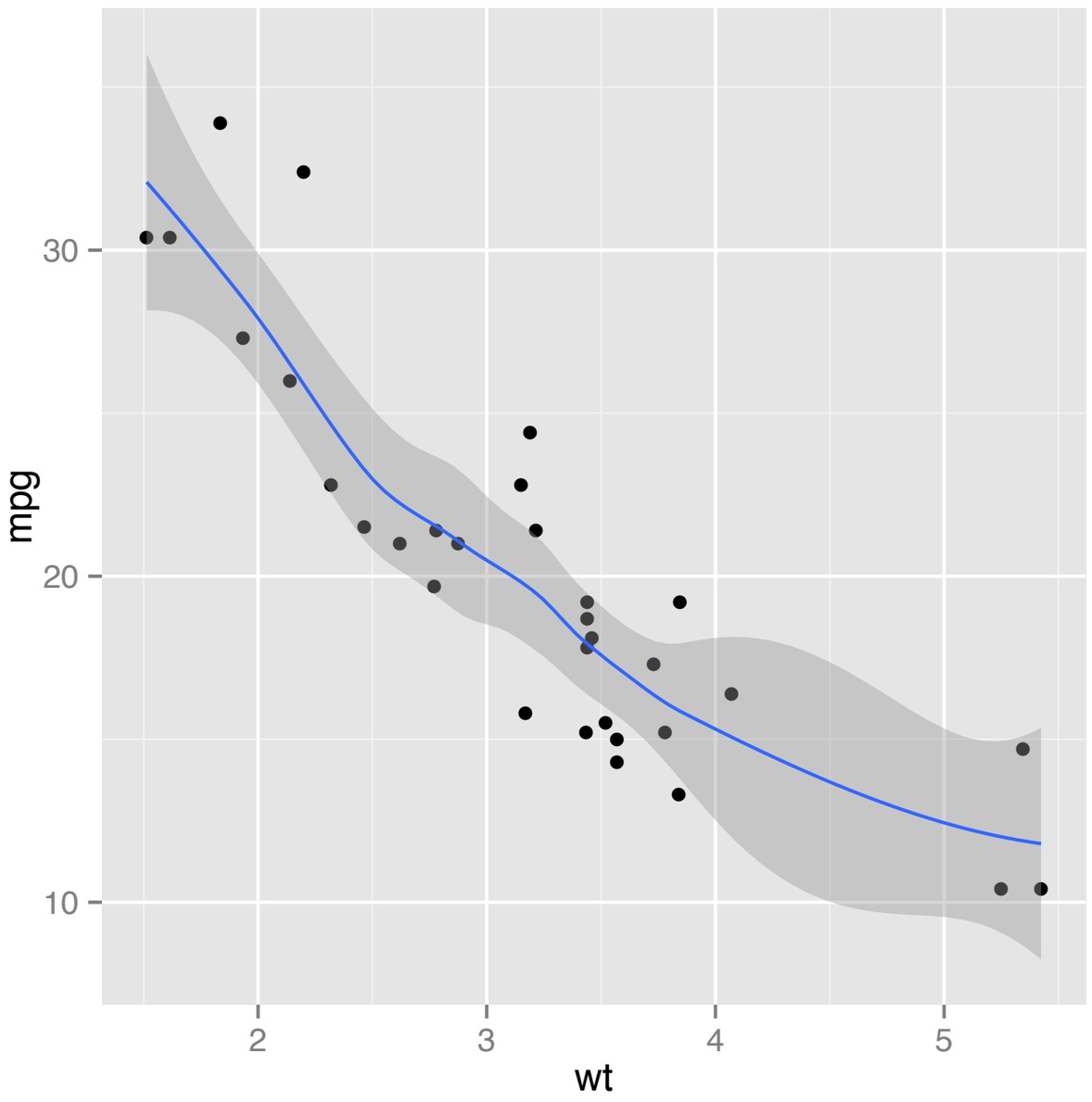
More advanced graphics

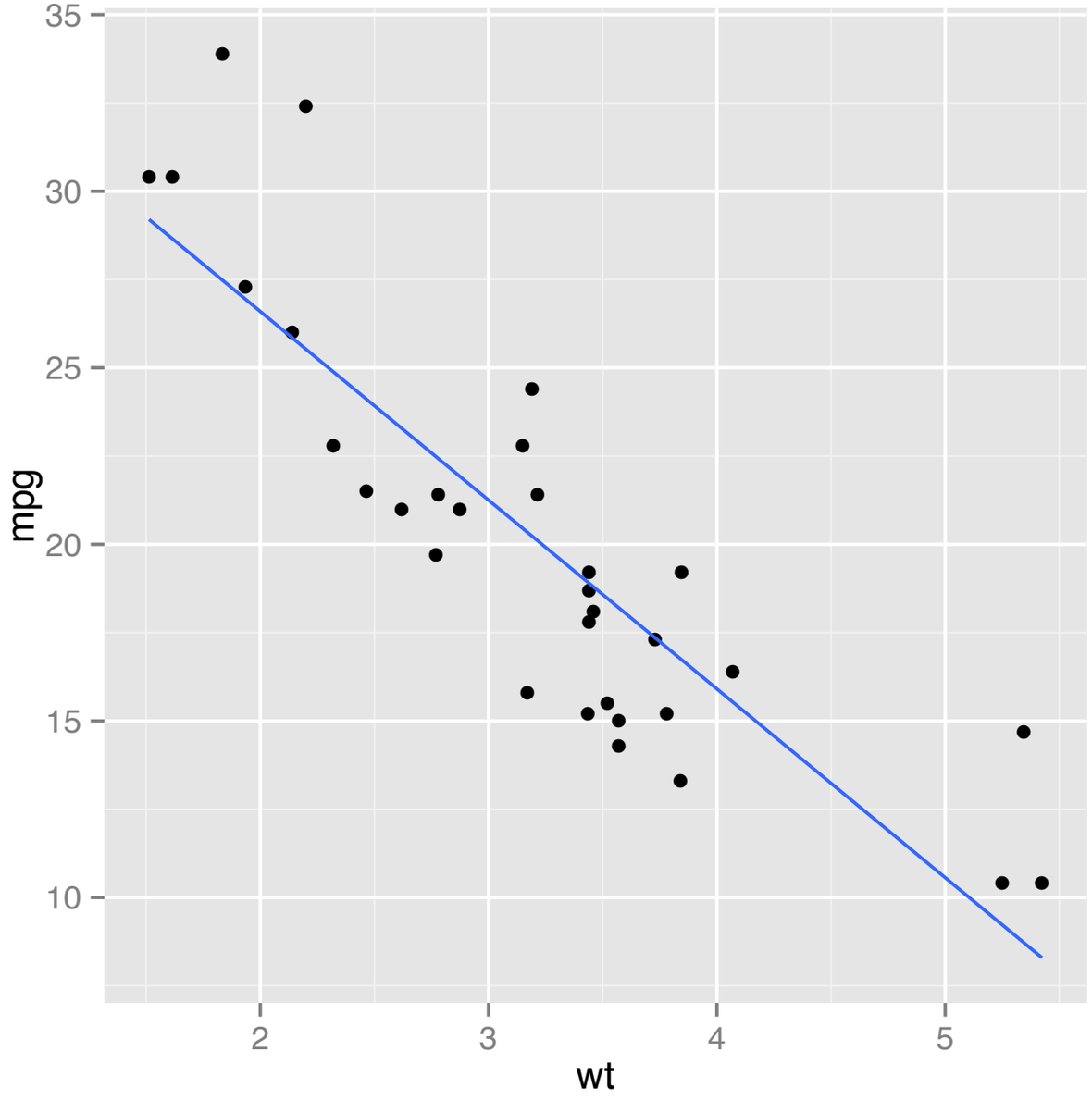



```
ggplot(mpg, aes(x=hwy)) +
  geom_histogram(binwidth=2)
```

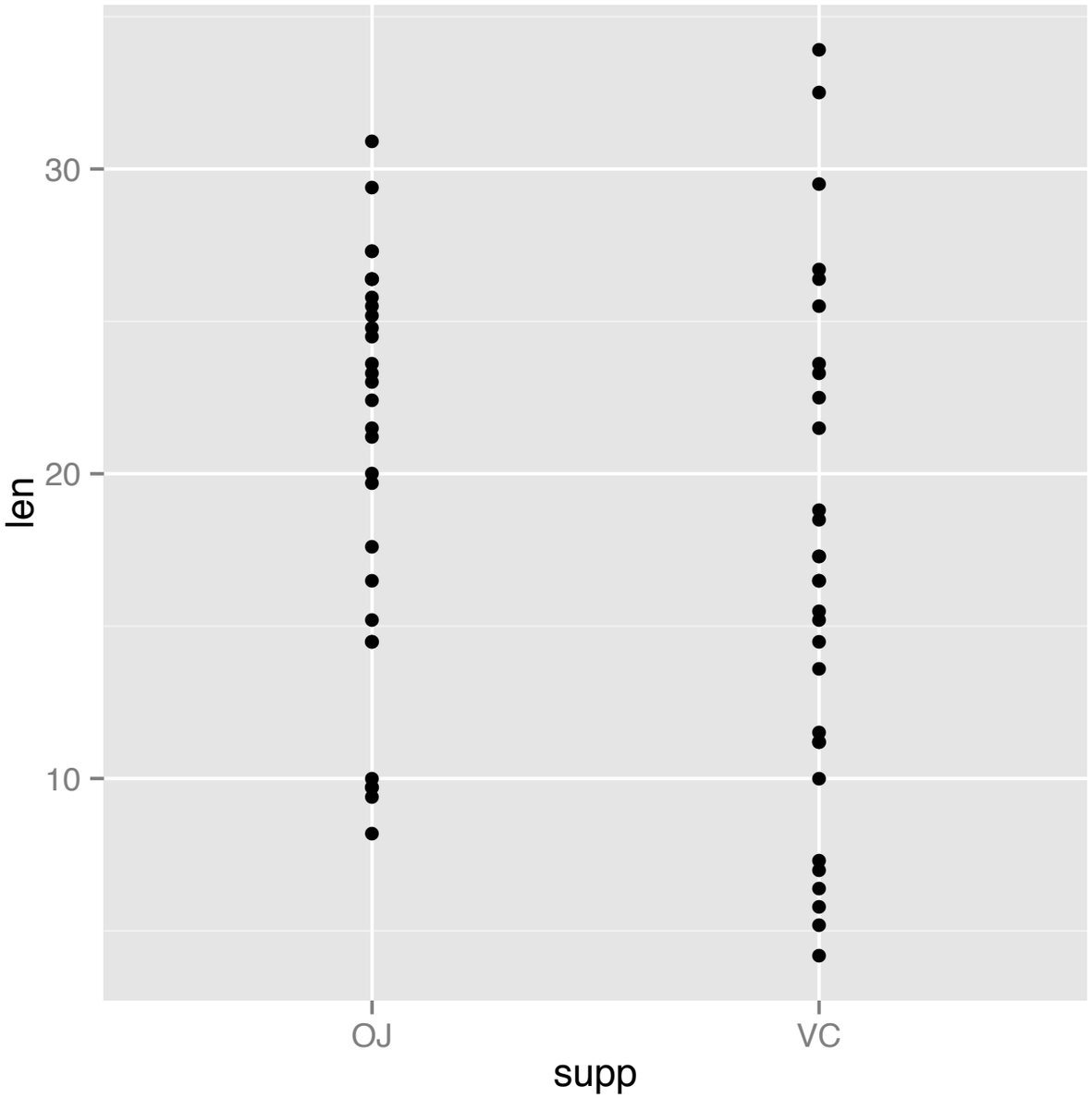


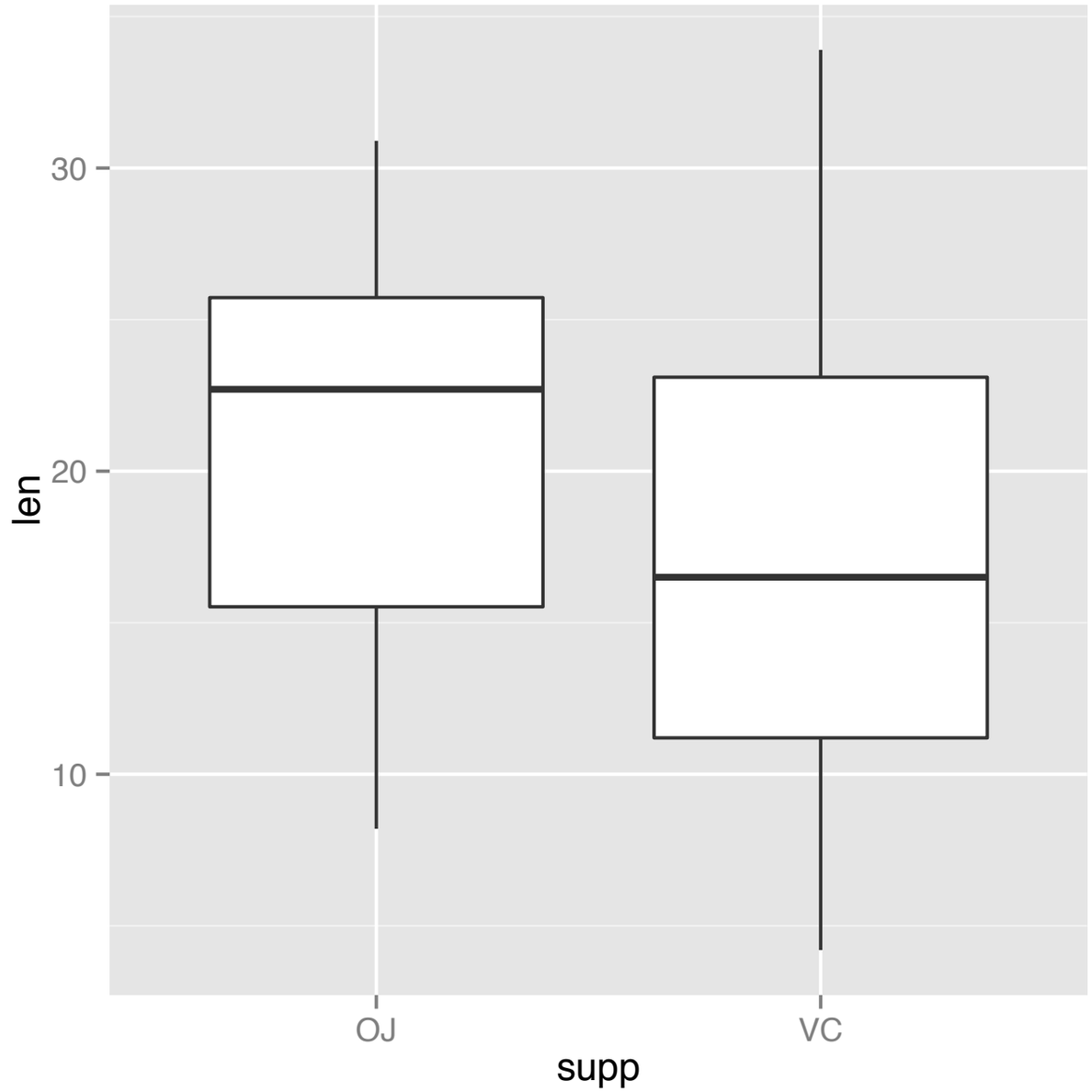

```
ggplot(mpg, aes(x=hwy)) +
  geom_histogram(binwidth=2) +
  facet_grid(drv ~ .)
```





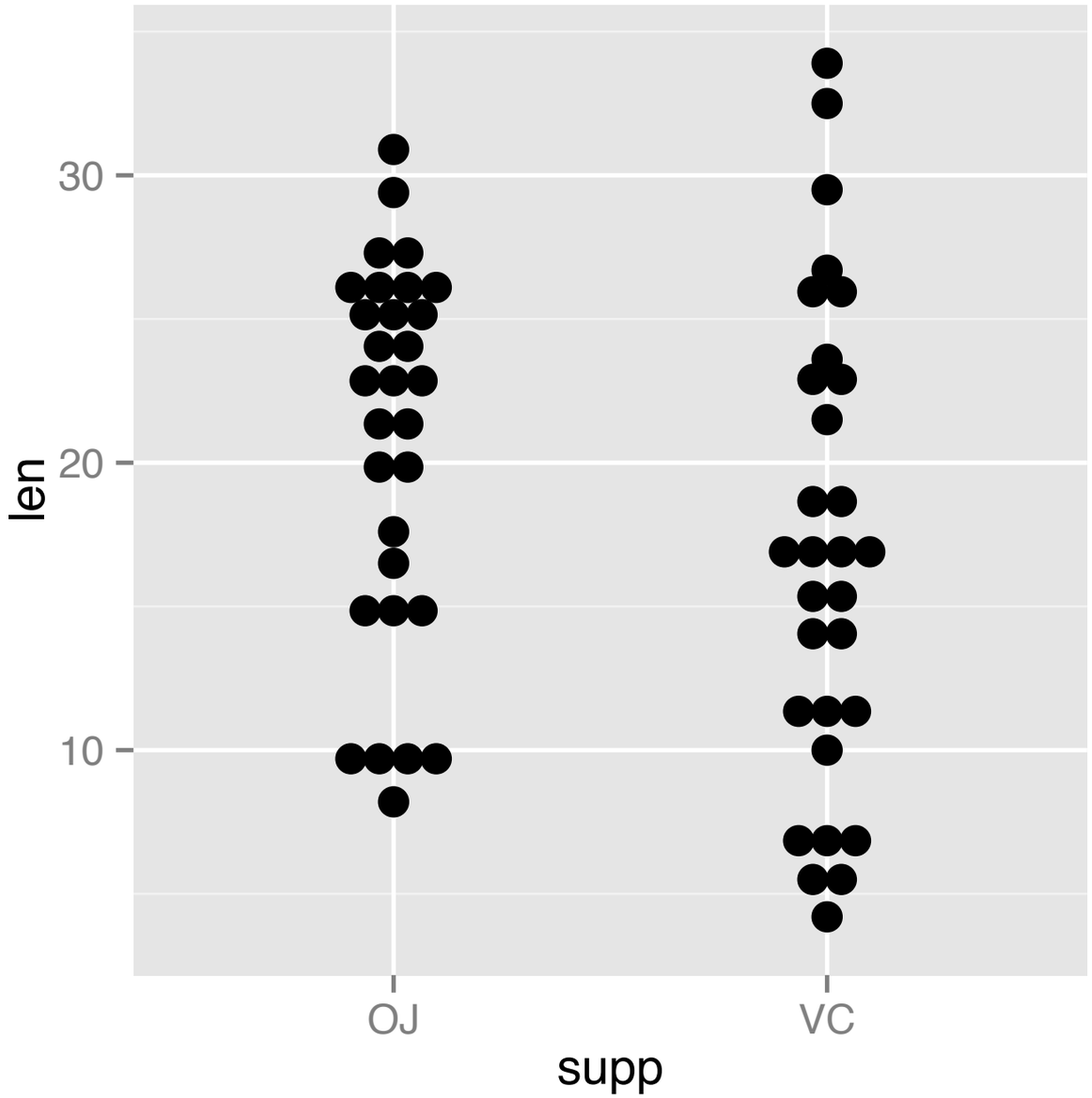

```
p <- ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()
p + geom_smooth()

p + geom_smooth(method=lm, se=FALSE)
```

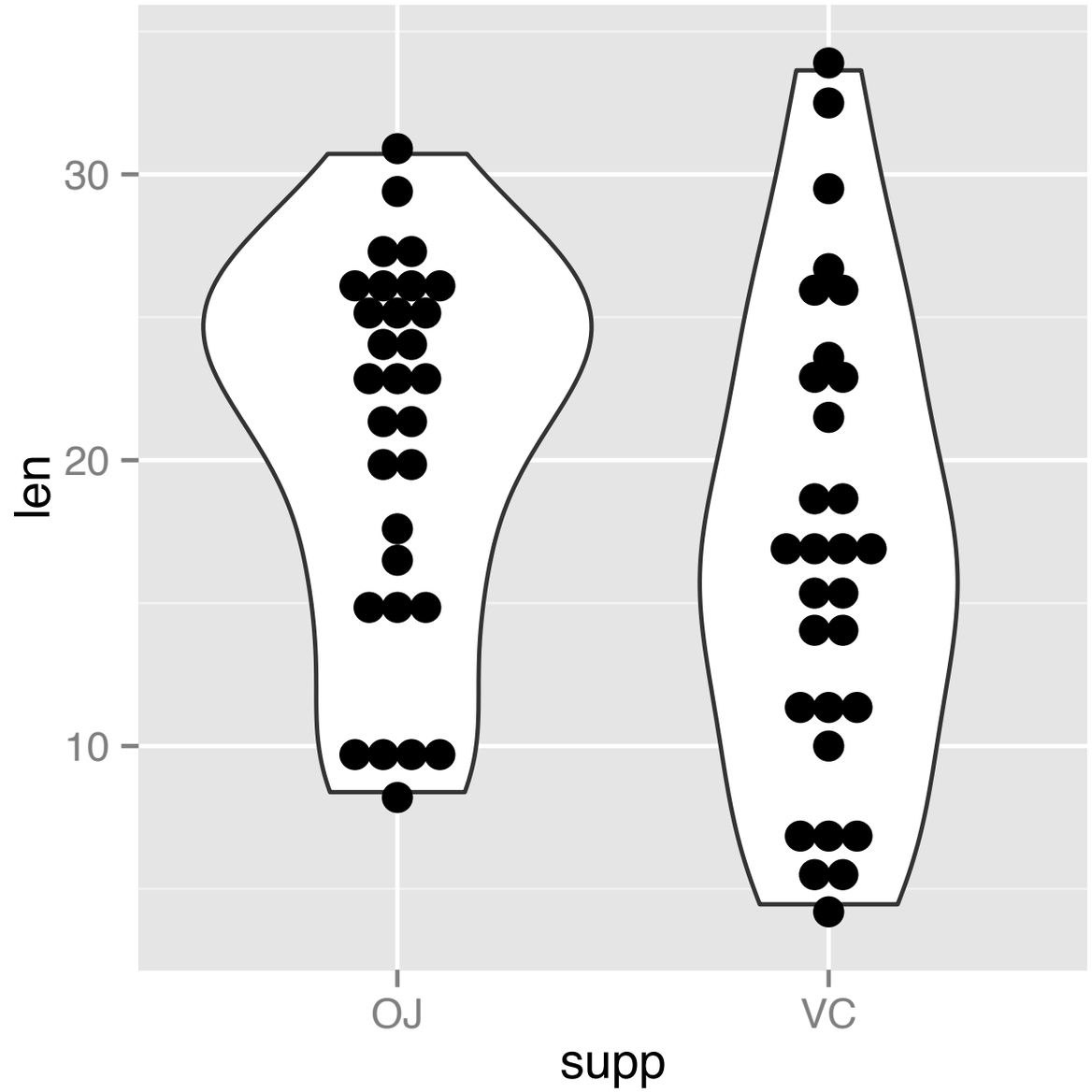




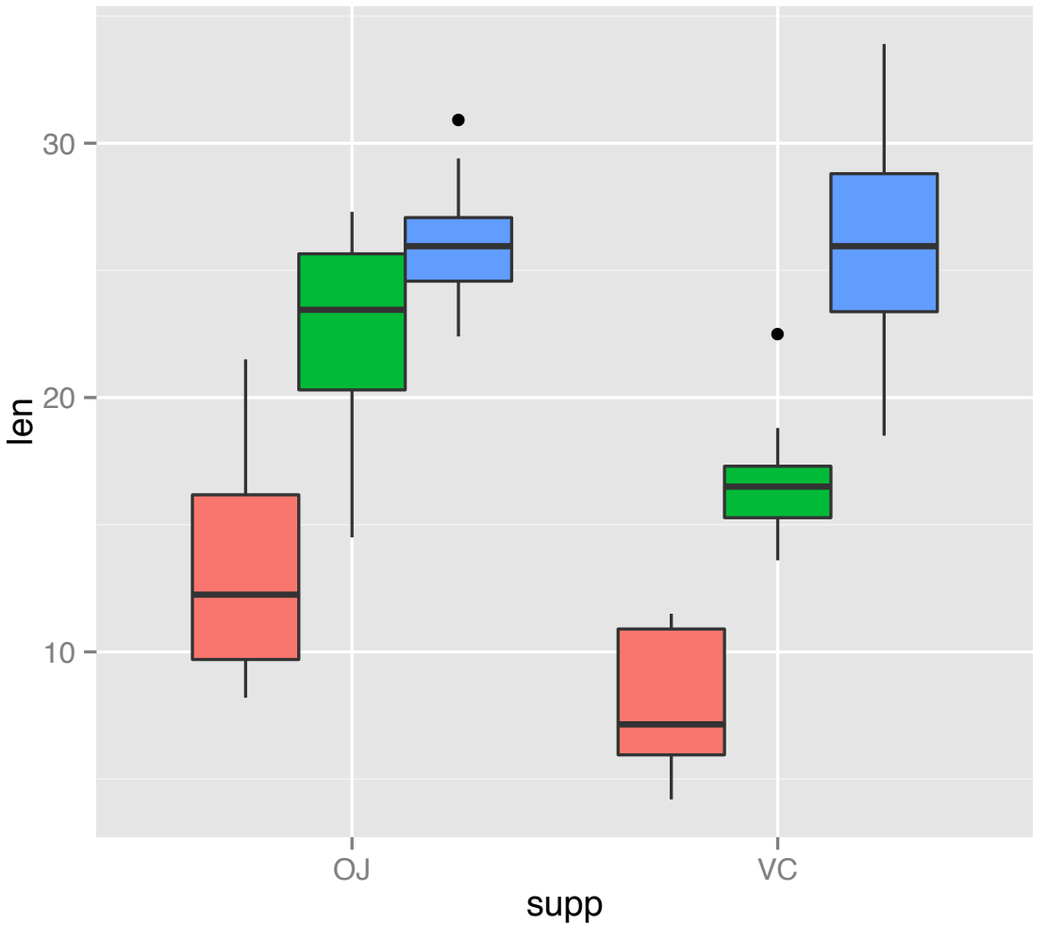

```
ggplot(ToothGrowth, aes(x=supp, y=len)) + geom_point()
```

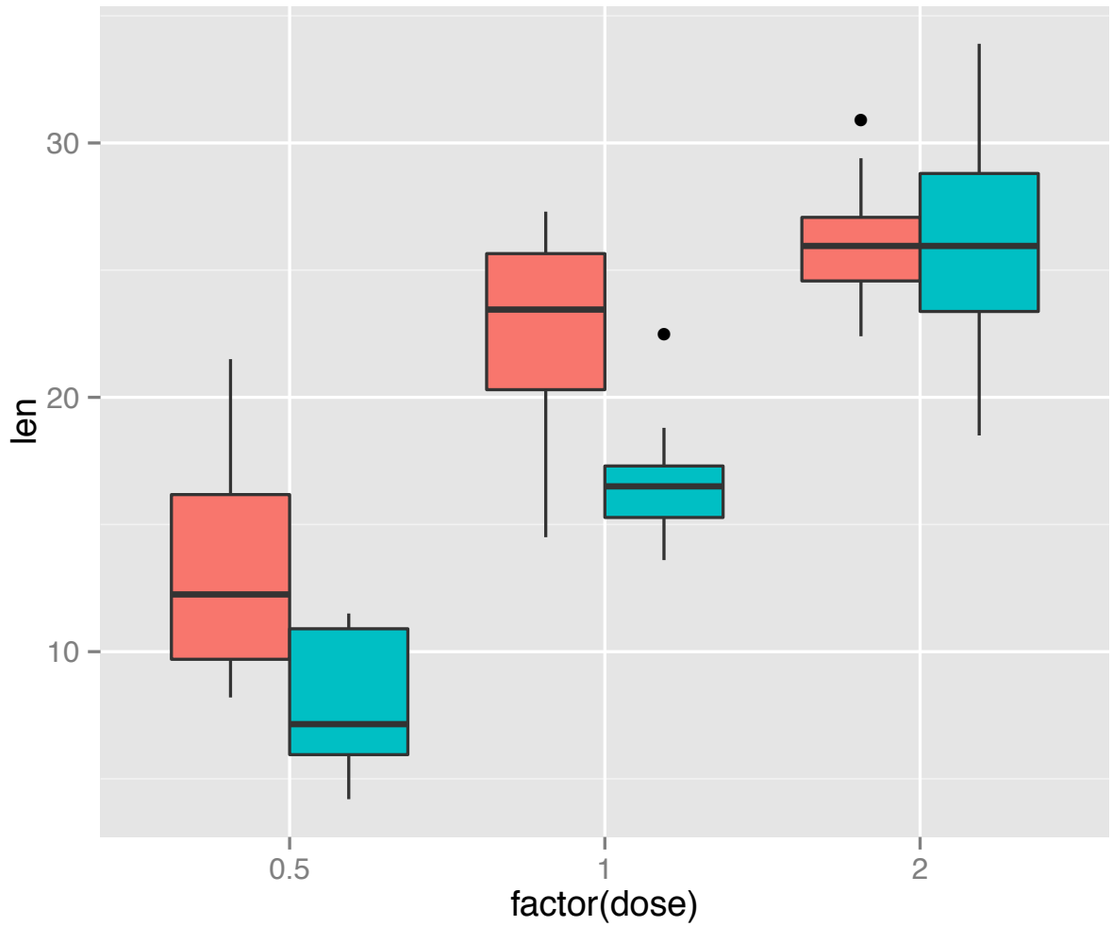
```
ggplot(ToothGrowth, aes(x=supp, y=len)) + geom_boxplot()
```



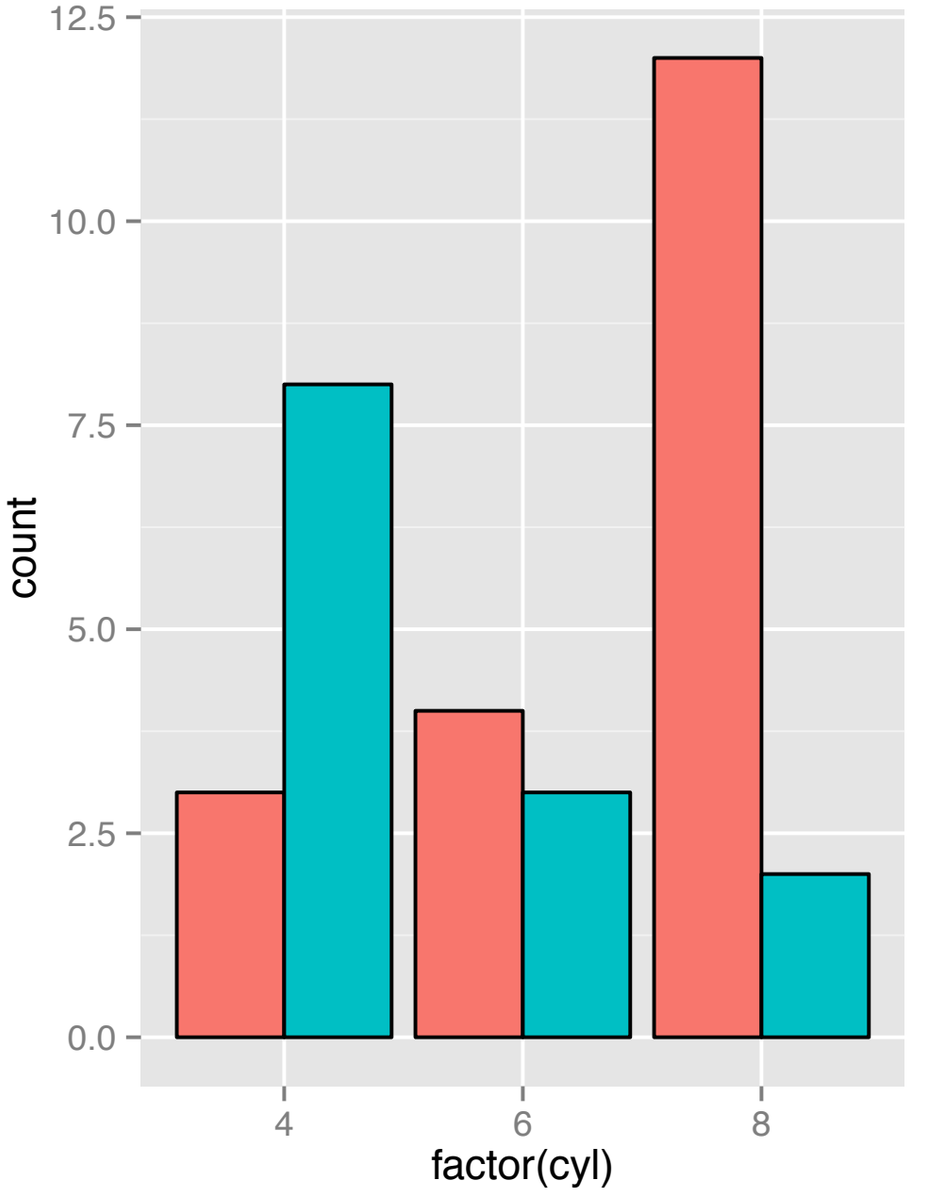


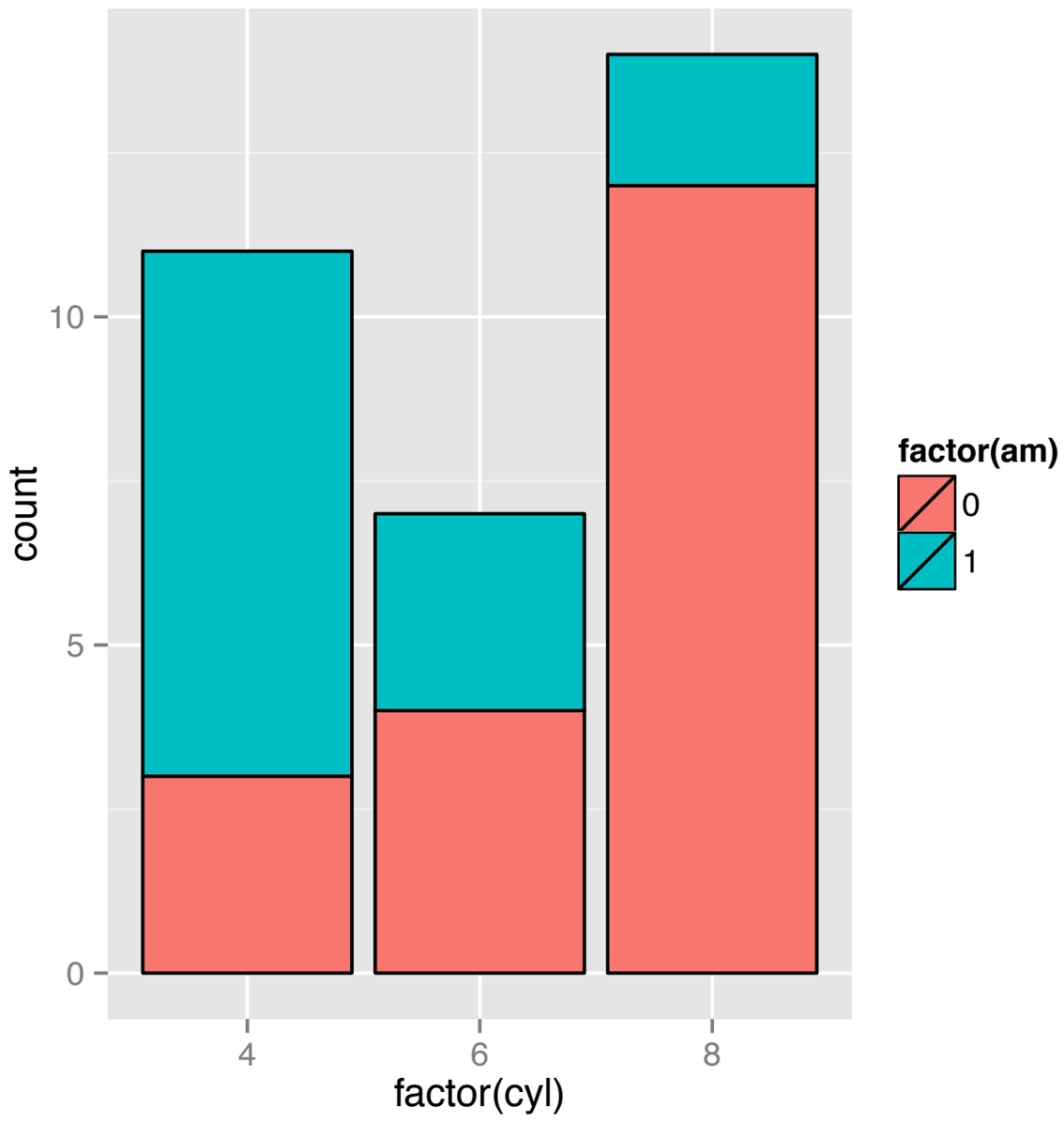

```
ggplot(ToothGrowth, aes(x=supp, y=len)) +
  geom_dotplot(binaxis="y", stackdir="center")

ggplot(ToothGrowth, aes(x=supp, y=len)) + geom_violin()
```



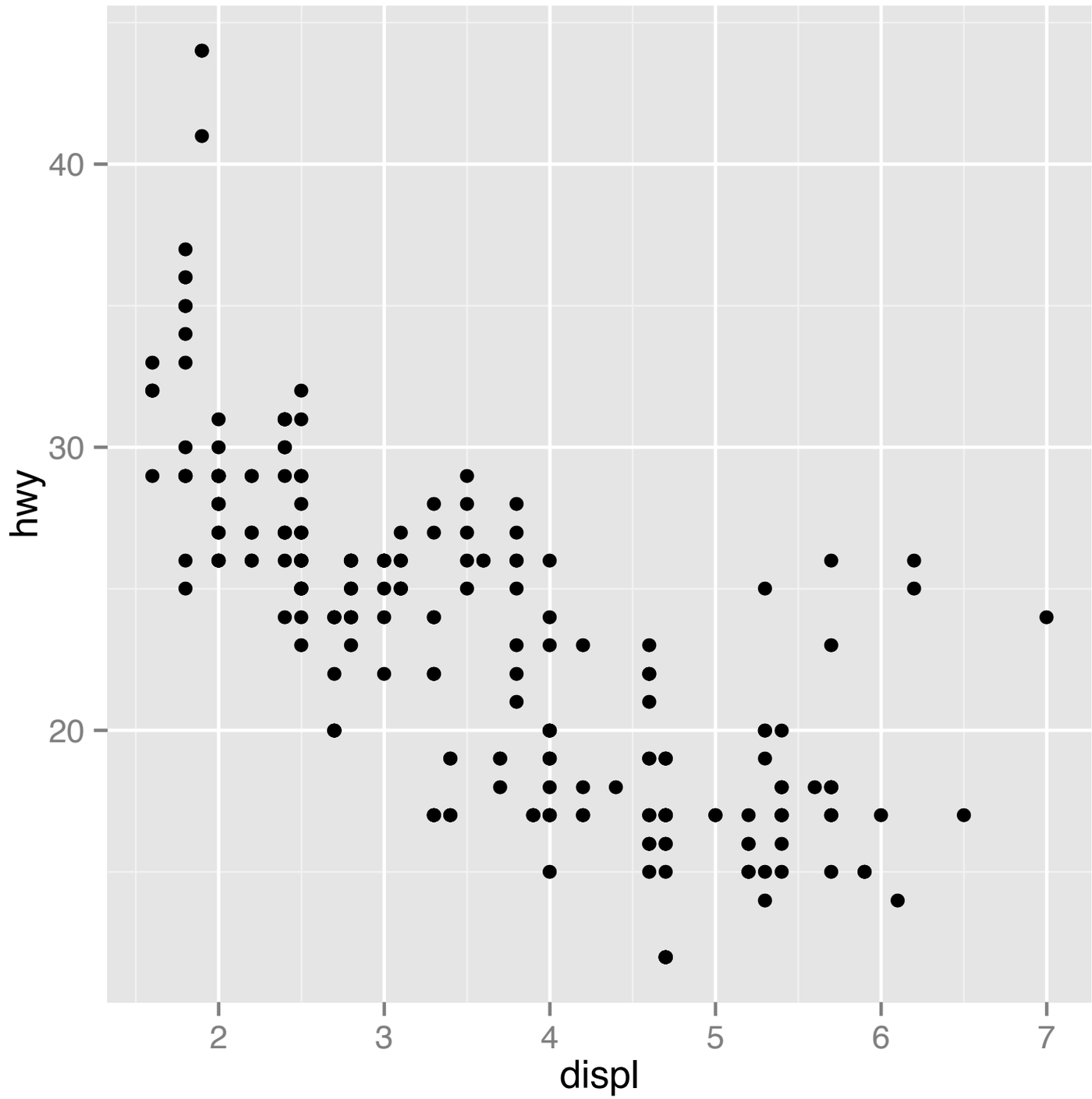



```
ggplot(ToothGrowth, aes(x=supp, y=len, fill=factor(dose))) +
  geom_boxplot()

ggplot(ToothGrowth, aes(x=factor(dose), y=len, fill=supp)) +
  geom_boxplot()
```

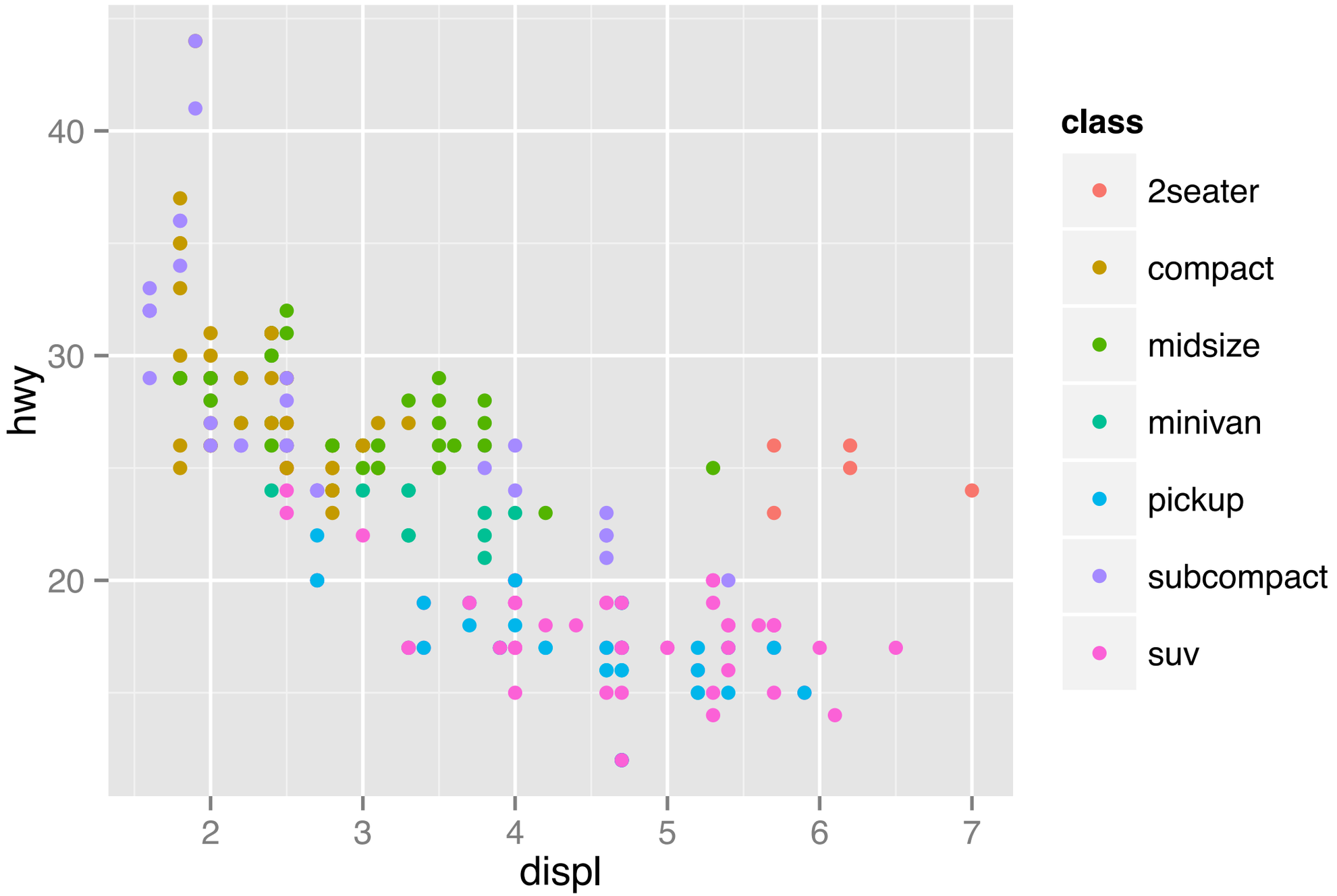





```
p <- ggplot(mtcars, aes(x=factor(cyl), fill=factor(am))) +
p + geom_bar(position="dodge", colour="black")

p + geom_bar(position="stack", colour="black")
```




```
ggplot(mpg, aes(x=displ, y=hwy)) +
  geom_point()
```




```
ggplot(mpg, aes(x=displ, y=hwy, colour=class)) +
  geom_point()
```

Saving output

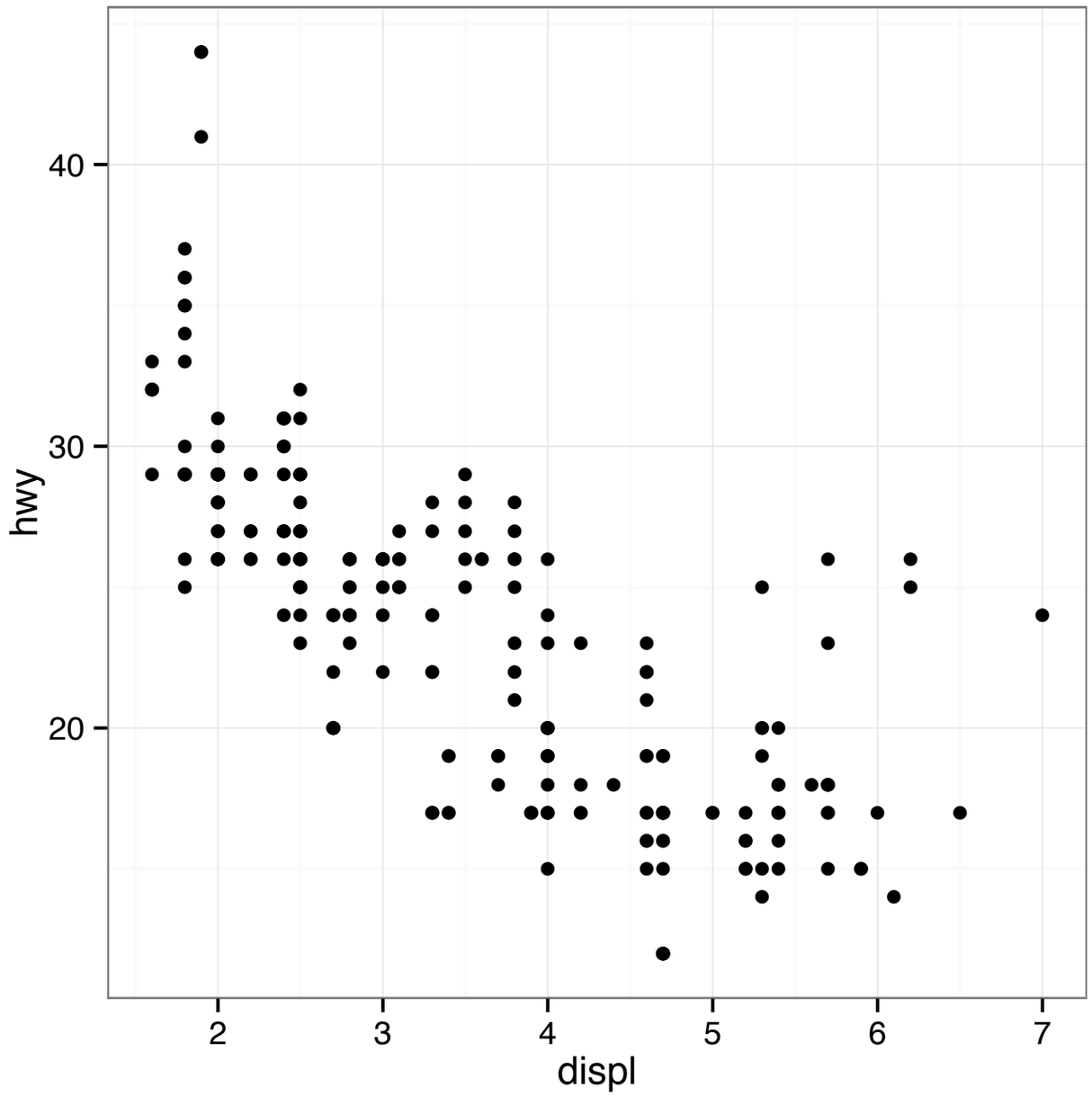
```
ggplot(mpg, aes(x=displ, y=hwy)) + geom_point()
ggsave("scatter.png")
```

```
p <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point()
ggsave("scatter.png", p, width=4, height=4)
```

```
p <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point()
png("scatter.png")  # Or you can use pdf
print(p)
dev.off()
```

Customizing appearance

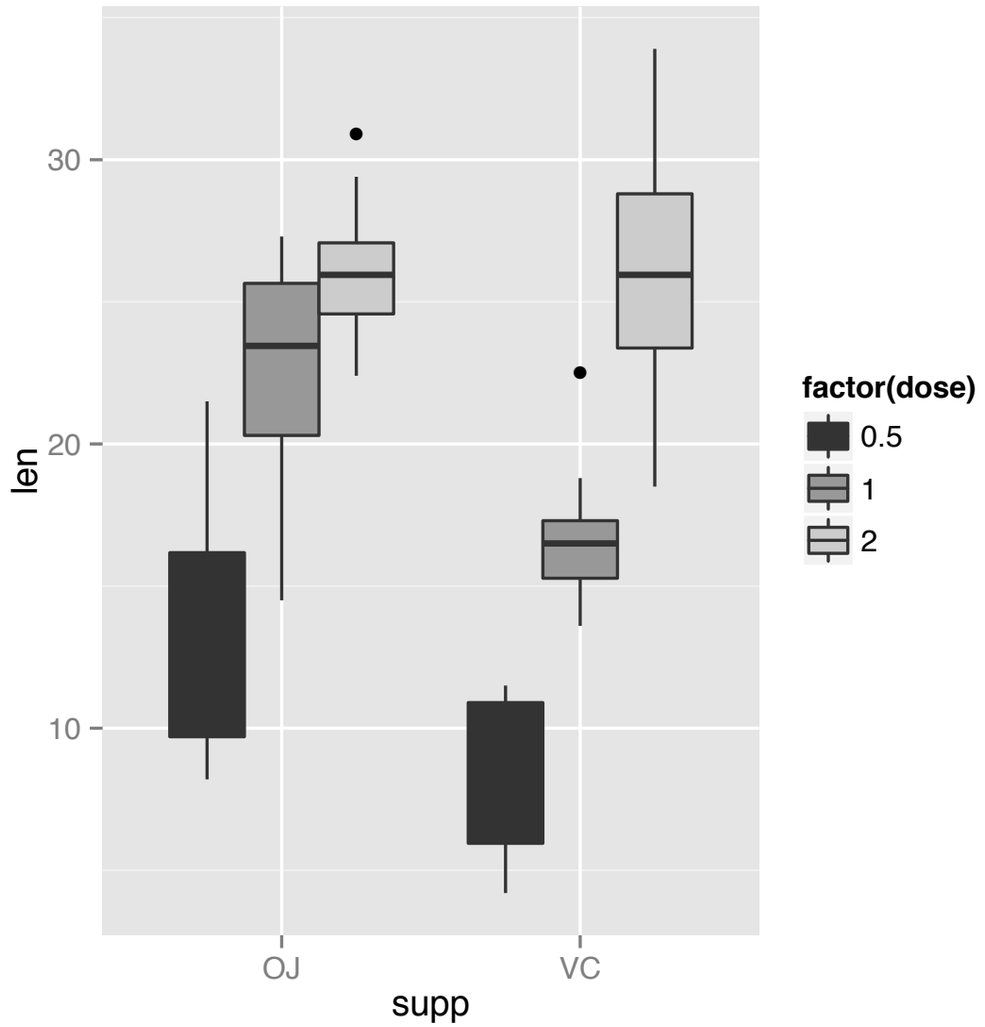
Themes



Themes control the appearance of non-data elements of the graph.

```
ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() +
  theme_bw()
```

Scales



Scales control the mapping from data values to aesthetic properties.

```
ggplot(ToothGrowth, aes(x=supp, y=len, fill=factor(dose))) +
  geom_boxplot() +
  scale_fill_grey()
```

Wide vs long data

```
# install.packages("gcookbook")
library(gcookbook)
simpledat
   A1 A2 A3
B1 10  7 12
B2  9 11  6

simpledat_long
  Aval Bval value
1   A1   B1    10
2   A1   B2     9
3   A2   B1     7
4   A2   B2    11
5   A3   B1    12
6   A3   B2     6
```

ggplot2 always uses "long" data!

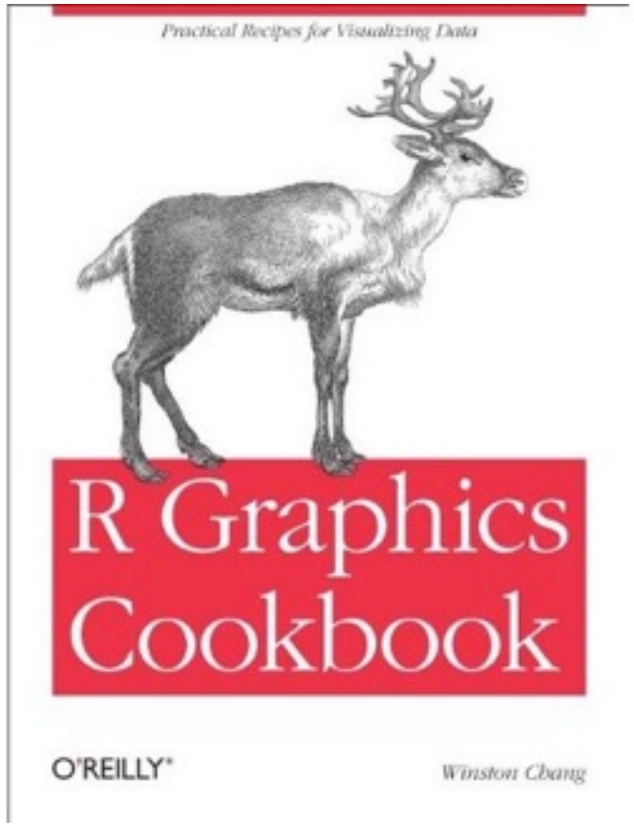
```
plum_wide
  length       time dead alive
1   long   at_once   84   156
2   long in_spring  156    84
3  short   at_once  133   107
4  short in_spring  209    31

plum
  length       time survival count
1   long   at_once      dead    84
2   long in_spring      dead   156
3  short   at_once      dead   133
4  short in_spring      dead   209
5   long   at_once     alive   156
6   long in_spring     alive    84
7  short   at_once     alive   107
8  short in_spring     alive    31
```

Use tidyr or reshape2 package to convert between long and wide formats.

http://www.cookbook-r.com/Manipulating_data/

Useful resources

Cookbook for R:
http://www.cookbook-r.com/



Official ggplot2 documentation:
http://docs.ggplot2.org/current/

Stack Overflow:
http://stackoverflow.com/

ggplot2 mailing list