

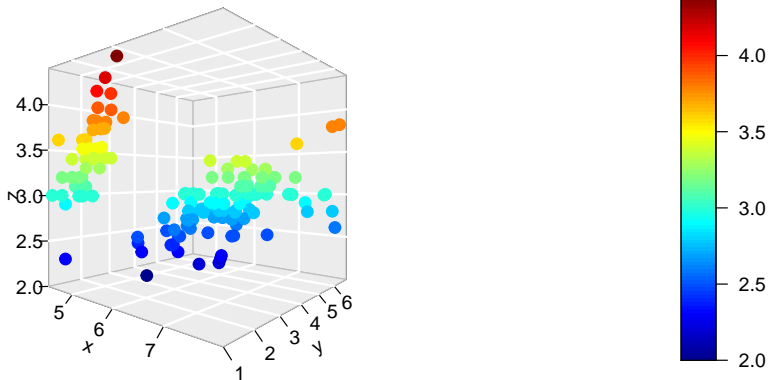
R language and data analysis: ggplot2

Qiang Shen

Dec 30, 2015

Visualization

- ▶ A picture is worth a thousand words



data visualization

- ▶ base package
- ▶ plot for summary statistics

motivation

```
library(gcookbook)
barplot(simpledat, beside=T); barplot(t(simpledat), beside=T)
plot(simpledat[1,], type='l')
lines(simpledat[2,], type='l', col='blue')
```

How is the case in ggplot2

```
library(ggplot2)
simplifiedat_long
ggplot(simplifiedat_long, aes(x=Aval, y=value, fill=Bval)) +
  geom_bar(stat='identity', position='dodge')
ggplot(simplifiedat_long, aes(x=Bval, y=value, fill=Aval)) +
  geom_bar(stat='identity', position='dodge')
ggplot(simplifiedat_long, aes(x=Aval, y=value, colour=Bval,
                             group=Bval)) + geom_line()
```

four graphics system in R.

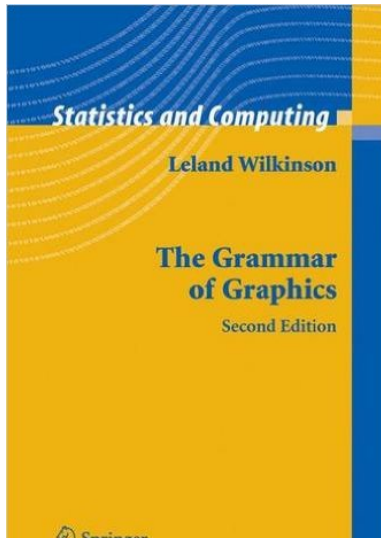
System	Included In base Installation?	Must be explicitly loaded?
base	Yes	No
grid	Yes	Yes
lattice	Yes	Yes
ggplot2	No	Yes

package requirement for the chapter

- ▶ ggplot2
- ▶ car
- ▶ gridExtra
- ▶ ggthemes

ggplot2

ggplot2 is an implementation of Leland Wilkinson's Grammar of Graphics—a general scheme for data visualization which breaks up graphs into semantic components such as scales and layers.



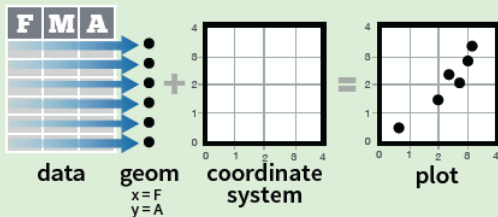
ggplot2

Plot = data + Aesthetics + Geometry.

- ▶ data is a data frame
- ▶ Aesthetics (aes) is used to indicate x and y variables. It can also be used to control the color, the size or the shape of a point. . .
- ▶ Geometry (geom) corresponds to the type of graphics (histogram, box plot, line plot, density plot, dot plot)

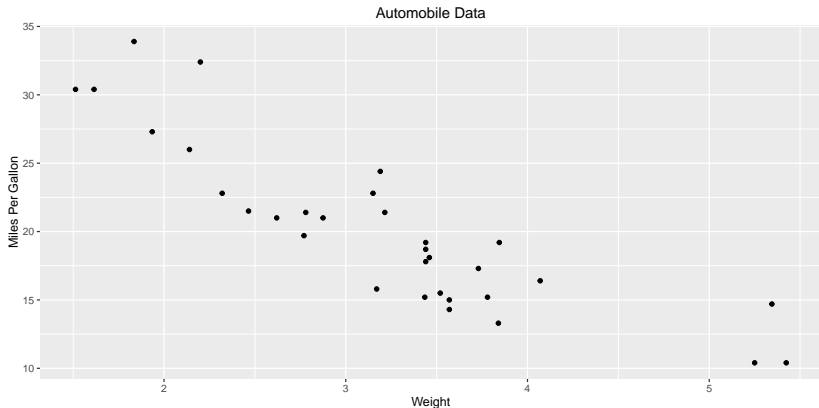
ggplot2

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system**.



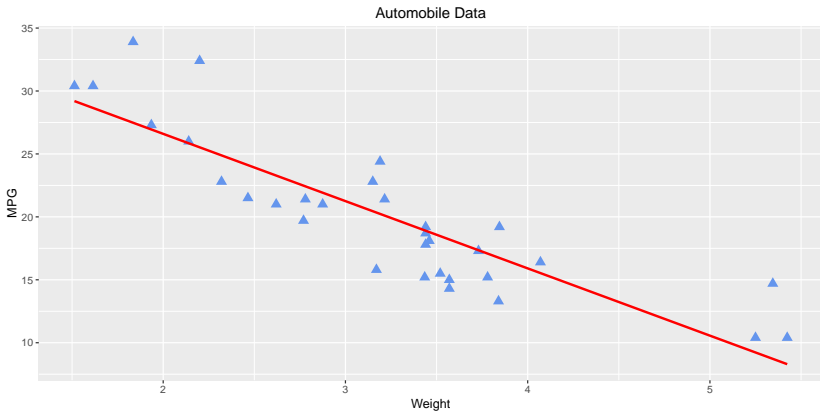
Basic scatterplot

```
library(ggplot2)
# with(mtcars, plot(wt, mpg))
ggplot(data=mtcars, aes(x=wt, y=mpg))+
  geom_point()+labs(title="Automobile Data",
                    x="Weight", y="Miles Per Gallon")
```



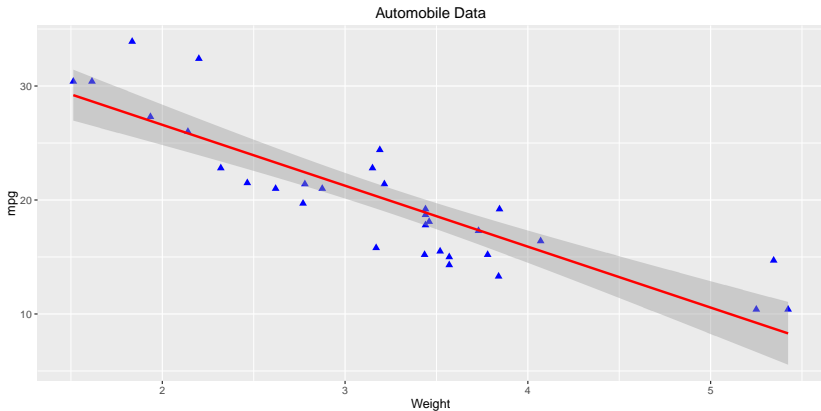
Scatter plot with additional options

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +  
  geom_point(pch=17,color="cornflowerblue",size=3)+  
  geom_smooth(method="lm",color="red",se=F,linetype=1)+  
  labs(title="Automobile Data", x="Weight", y="MPG")
```



ggplot2

```
p<-ggplot(data=mtcars, aes(x=wt, y=mpg))  
p+ geom_point(pch=17, color="blue", size=2)+  
geom_smooth(method="lm",color="red",linetype=1)+  
labs(title="Automobile Data",x="Weight",y="mpg")
```

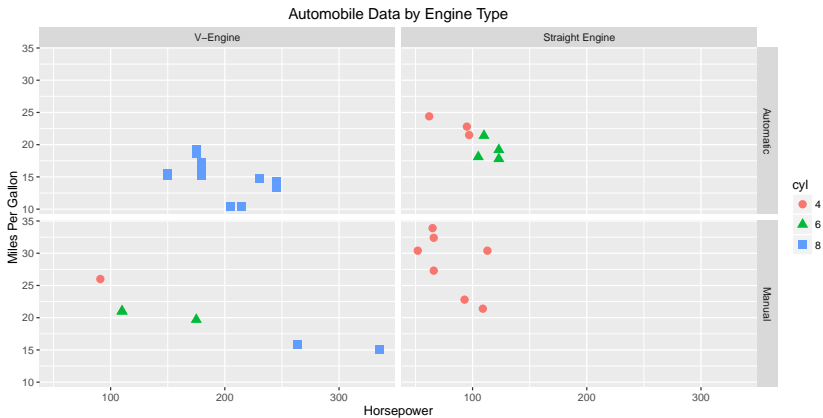


Scatter plot with grouping and faceting

- ▶ grouping (display groups in a single plot)
- ▶ faceting (display observations in separate, side-by-side plots)
- ▶ need to be factors

```
data(mtcars)
mtcars$am <- factor(mtcars$am, levels=c(0,1),
                    labels=c("Automatic", "Manual"))
mtcars$vs <- factor(mtcars$vs, levels=c(0,1),
                    labels=c("V-Engine", "Straight Engine"))
mtcars$cyl <- factor(mtcars$cyl)
```

Scatter plot with grouping and faceting



Scatter plot with grouping and faceting

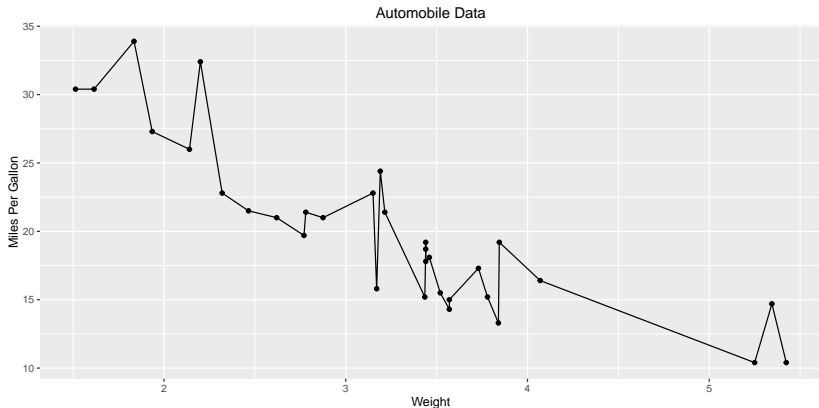
```
library(ggplot2)
ggplot(data=mtcars, aes(x=hp, y=mpg,
                        shape=cyl, color=cyl)) +
  geom_point(size=3) +
  facet_grid(am~vs) +
  labs(title="Automobile Data by Engine Type",
       x="Horsepower", y="Miles Per Gallon")
```


geometric objects (geom)

Function	Adds	Options
<code>geom_bar()</code>	Bar chart	color, fill, alpha
<code>geom_boxplot()</code>	Box plot	color, fill, alpha, notch, width
<code>geom_density()</code>	Density plot	color, fill, alpha, linetype
<code>geom_histogram()</code>	Histogram	color, fill, alpha, linetype, binwidth
<code>geom_hline()</code>	Horizontal lines	color, alpha, linetype, size
<code>geom_jitter()</code>	Jittered points	color, size, alpha, shape
<code>geom_line()</code>	Line graph	color, alpha, linetype, size
<code>geom_point()</code>	Scatterplot	color, alpha, shape, size
<code>geom_rug()</code>	Rug plot	color, side
<code>geom_smooth()</code>	Fitted line	method, formula, color, fill, linetype, size
<code>geom_text()</code>	Text annotations	Many; see the help for this function
<code>geom_violin()</code>	Violin plot	color, fill, alpha, linetype
<code>geom_vline()</code>	Vertical lines	color, alpha, linetype, size

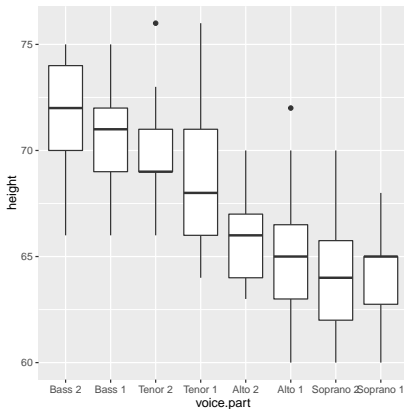
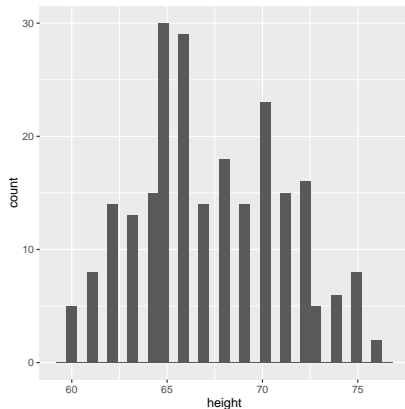
Using geoms: line

```
library(ggplot2)
ggplot(data=mtcars, aes(x=wt, y=mpg))+geom_line()+
  geom_point() + labs(title="Automobile Data",
    x="Weight", y="Miles Per Gallon")
```



Using geoms: histogram, geom_boxplot

```
data(singer, package="lattice")  
p1<-ggplot(singer, aes(x=height)) + geom_histogram()  
p2<-ggplot(singer, aes(x=voice.part, y=height)) + geom_boxplot()  
library(gridExtra); grid.arrange(p1, p2, ncol=2)
```

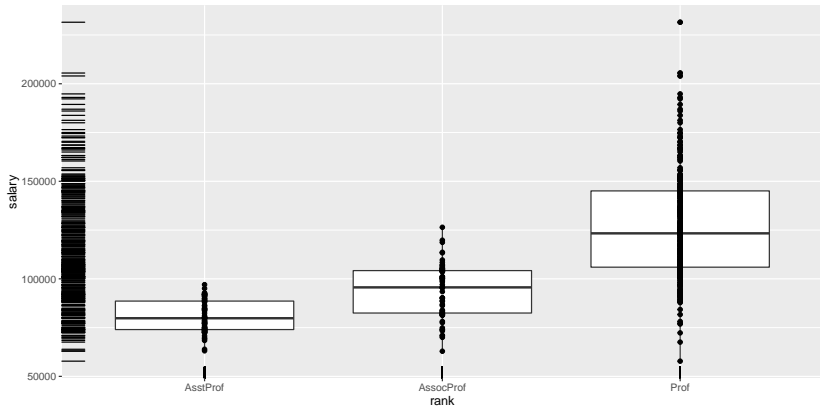


common options for geom functions

Option	Specifies
<code>color</code>	Color of points, lines, and borders around filled regions.
<code>fill</code>	Color of filled areas such as bars and density regions.
<code>alpha</code>	Transparency of colors, ranging from 0 (fully transparent) to 1 (opaque).
<code>linetype</code>	Pattern for lines (1 = solid, 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash).
<code>size</code>	Point size and line width.
<code>shape</code>	Point shapes (same as <code>pch</code> , with 0 = open square, 1 = open circle, 2 = open triangle, and so on). See figure 3.4 for examples.
<code>position</code>	Position of plotted objects such as bars and points. For bars, "dodge" places grouped bar charts side by side, "stacked" vertically stacks grouped bar charts, and "fill" vertically stacks grouped bar charts and standardizes their heights to be equal. For points, "jitter" reduces point overlap.
<code>binwidth</code>	Bin width for histograms.
<code>notch</code>	Indicates whether box plots should be notched (TRUE/FALSE).
<code>sides</code>	Placement of rug plots on the graph ("b" = bottom, "l" = left, "t" = top, "r" = right, "bl" = both bottom and left, and so on).
<code>width</code>	Width of box plots.

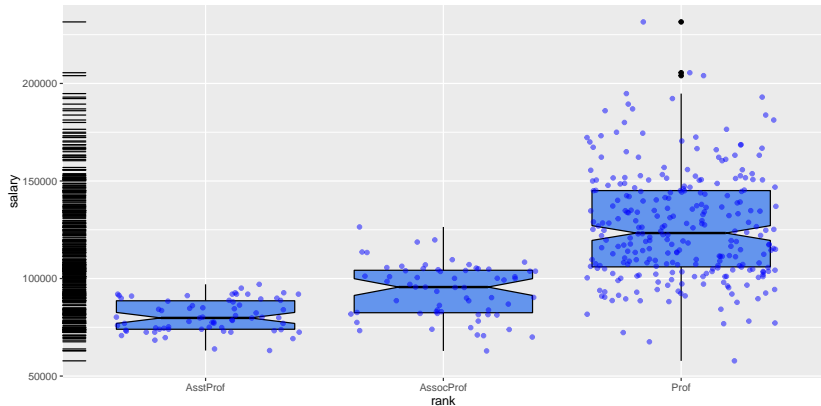
common options for geom functions

```
data(Salaries, package="car")  
library(ggplot2)  
ggplot(Salaries, aes(x=rank, y=salary)) +  
  geom_boxplot()+ geom_point()+ geom_rug()
```



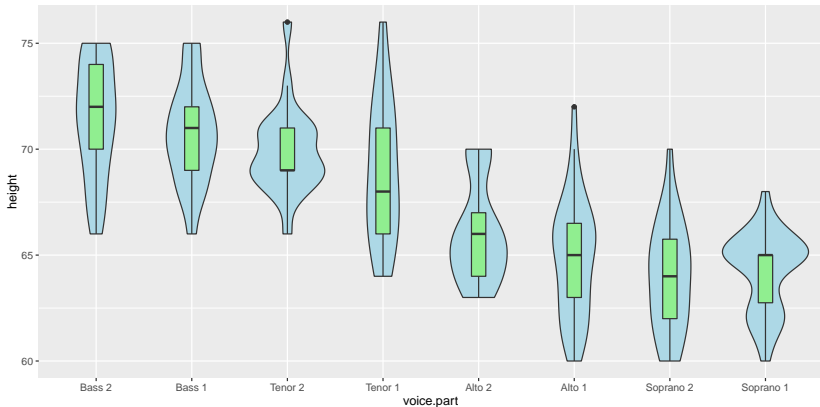
common options for geom functions

```
data(Salaries, package="car")
ggplot(Salaries, aes(x=rank, y=salary)) +
  geom_boxplot(fill="cornflowerblue",color="black",notch=T)+
  geom_point(position='jitter',color="blue", alpha=.5)+
  geom_rug(side="l", color="black")
```



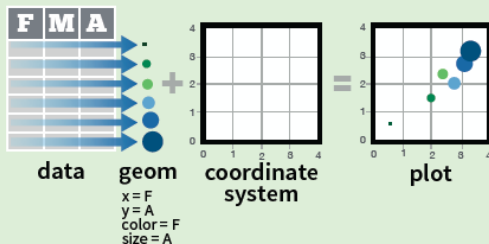
common options for geom functions

```
library(ggplot2)
data(singer, package="lattice")
ggplot(singer, aes(x=voice.part,y=height))+
  geom_violin(fill="lightblue")+
  geom_boxplot(fill="lightgreen", width=0.2)
```



Grouping

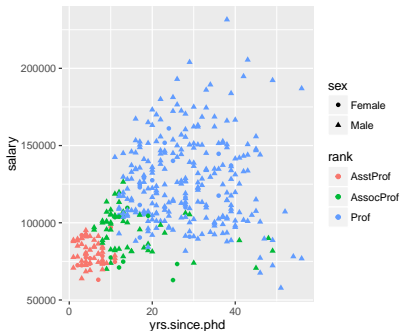
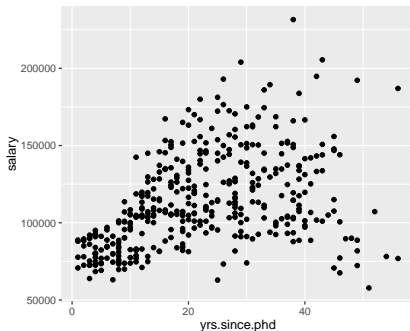
To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Grouping

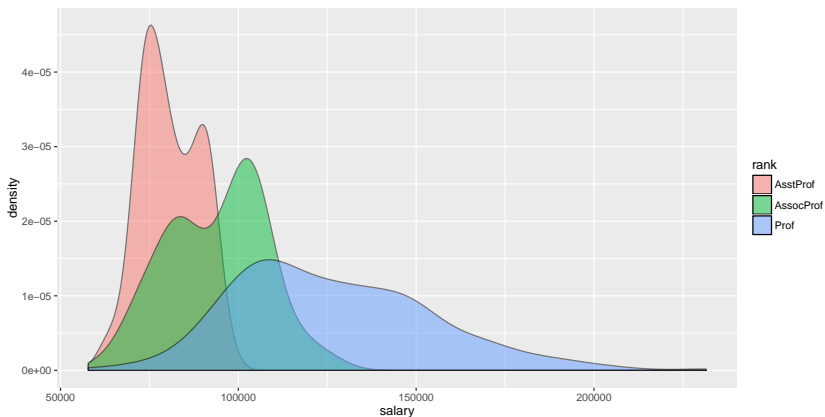
- color, fill, size, shape or combination.

```
data(Salaries, package="car")  
p<-ggplot(Salaries,aes(x=yrs.since.phd,y=salary))+geom_point()  
q<-ggplot(Salaries,aes(x=yrs.since.phd,y=salary,  
shape=sex,color=rank))+geom_point()  
library(gridExtra);grid.arrange(p,q,ncol=2)
```



Grouping: kernel density plot

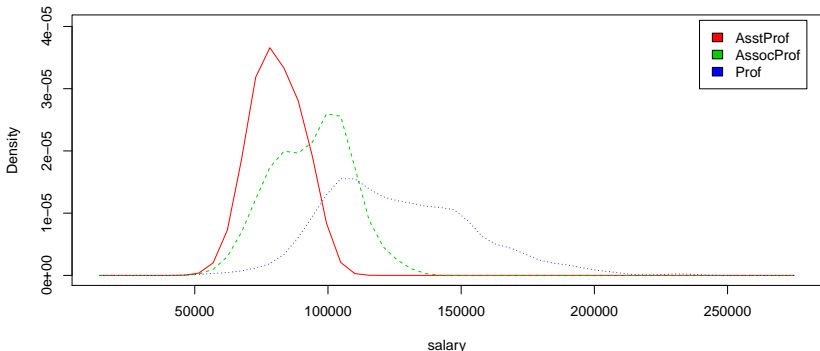
```
data(Salaries, package="car")  
ggplot(data=Salaries, aes(x=salary, fill=rank)) +  
  geom_density(alpha=.5, col='black')
```



```
# +facet_grid(rank~.)
```

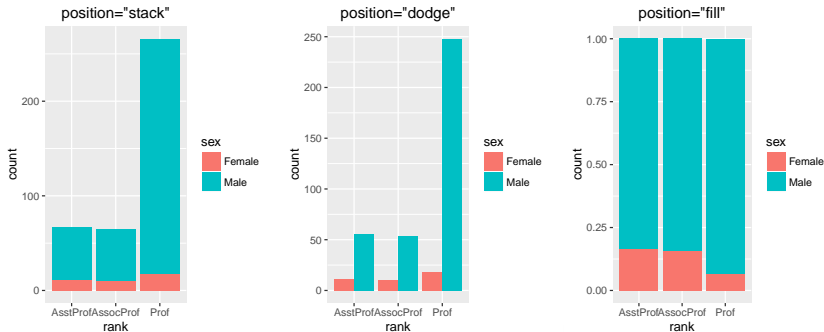
Grouping in base graphics

```
library(sm);data(Salaries, package="car")
attach(Salaries);sm.density.compare(salary,rank)
colfill<-c(2:(2+length(levels(rank))))
legend('topright',levels(rank),fill=colfill,inset=0.02)
```



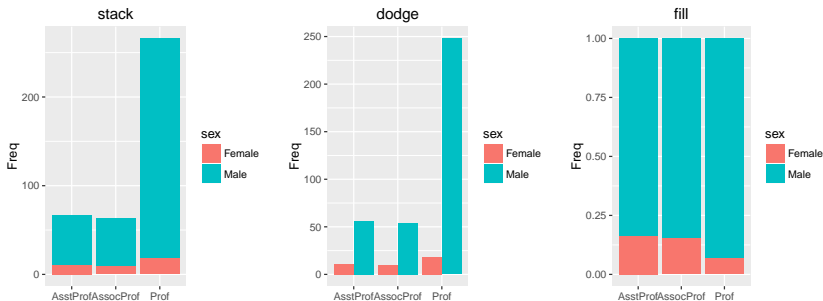
Grouping: barplot

```
a<-ggplot(Salaries, aes(x=rank, fill=sex))+  
geom_bar(position="stack")+labs(title='position="stack"')  
b<-ggplot(Salaries, aes(x=rank, fill=sex))+  
geom_bar(position="dodge")+labs(title='position="dodge"')  
c<-ggplot(Salaries, aes(x=rank, fill=sex))+  
geom_bar(position="fill")+labs(title='position="fill"')  
library(gridExtra);grid.arrange(a,b,c,ncol=3)
```



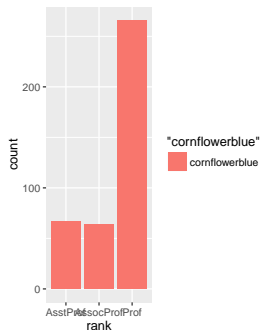
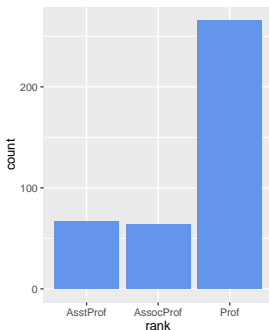
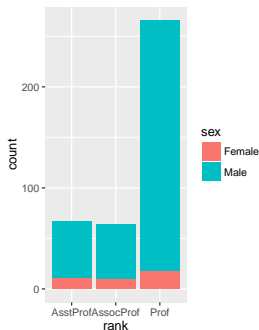
how to cope with summarized data: barplot

```
data<-data.frame(with(Salaries,table(sex,rank)))  
a<-ggplot(data, aes(x=rank,y=Freq,fill=sex))+geom_bar(  
  stat='identity',position="stack")+labs(title='stack')  
b<-ggplot(data, aes(x=rank,y=Freq,fill=sex))+geom_bar(  
  stat='identity',position="dodge")+labs(title='dodge')  
c<-ggplot(data, aes(x=rank,y=Freq,fill=sex))+geom_bar(  
  stat='identity',position="fill")+labs(title='fill')  
library(gridExtra);grid.arrange(a,b,c,ncol=3)
```



Placing options: barplot

```
a<-ggplot(Salaries,aes(x=rank,fill=sex))+geom_bar()  
b<-ggplot(Salaries,aes(x=rank))+ geom_bar(fill="cornflowerblue")  
c<-ggplot(Salaries,aes(x=rank,fill="cornflowerblue"))+  
geom_bar()  
library(gridExtra);grid.arrange(a,b,c,ncol=3)
```



Facet:trellis graphs

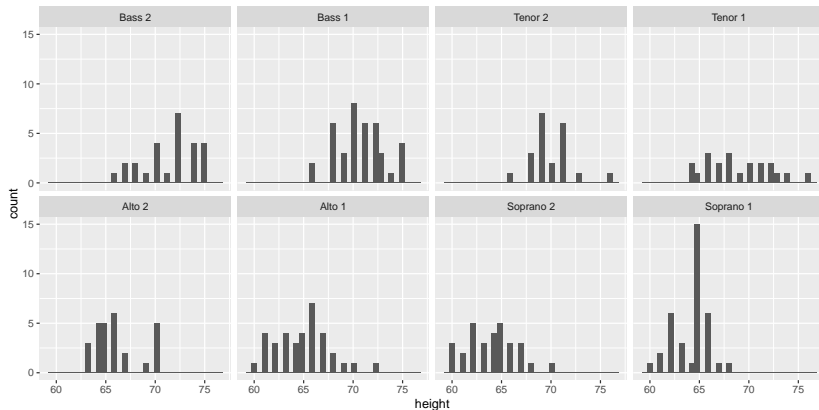
side-by-side graphs vs. single graph (grouping) - `facet_wrap` - `facet_grid`

Syntax	Results
<code>facet_wrap(~var, ncol=n)</code>	Separate plots for each level of <i>var</i> arranged into <i>n</i> columns
<code>facet_wrap(~var, nrow=n)</code>	Separate plots for each level of <i>var</i> arranged into <i>n</i> rows

Syntax	Results
<code>facet_grid(rowvar~colvar)</code>	Separate plots for each combination of <i>rowvar</i> and <i>colvar</i> , where <i>rowvar</i> represents rows and <i>colvar</i> represents columns
<code>facet_grid(rowvar~.)</code>	Separate plots for each level of <i>rowvar</i> , arranged as a single column
<code>facet_grid(.~colvar)</code>	Separate plots for each level of <i>colvar</i> , arranged as a single row

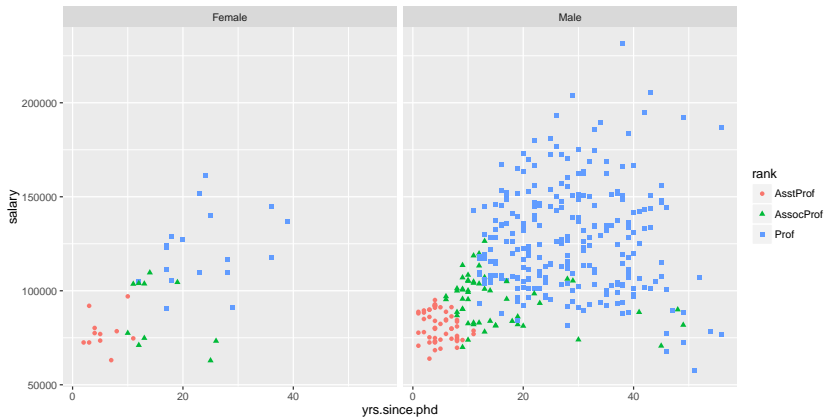
Faceting: facet_wrap

```
data(singer, package="lattice")  
ggplot(data=singer, aes(x=height)) +  
  geom_histogram() + facet_wrap(~voice.part, nrow=2)
```



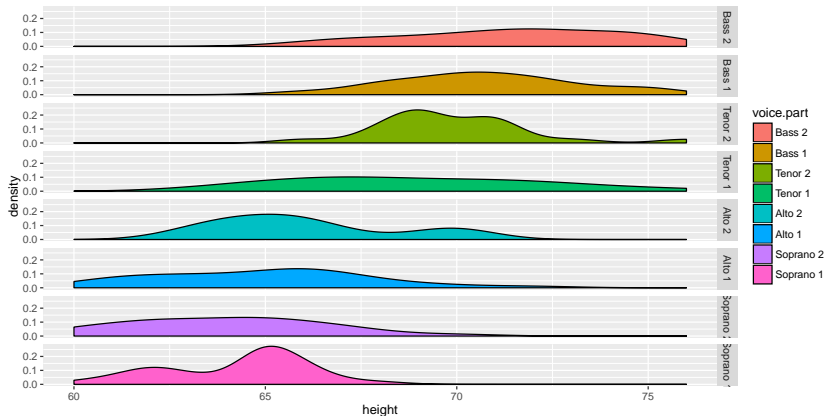
Faceting: facet_grid

```
ggplot(Salaries, aes(x=yrs.since.phd, y=salary,  
color=rank,shape=rank)) + geom_point() +  
facet_grid(.~sex)
```



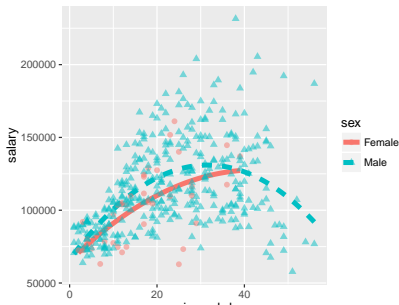
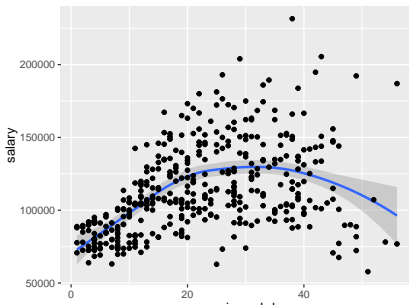
facet_grid

```
data(singer, package="lattice")  
library(ggplot2)  
ggplot(data=singer, aes(x=height, fill=voice.part))+  
  geom_density()+ facet_grid(voice.part~.)
```



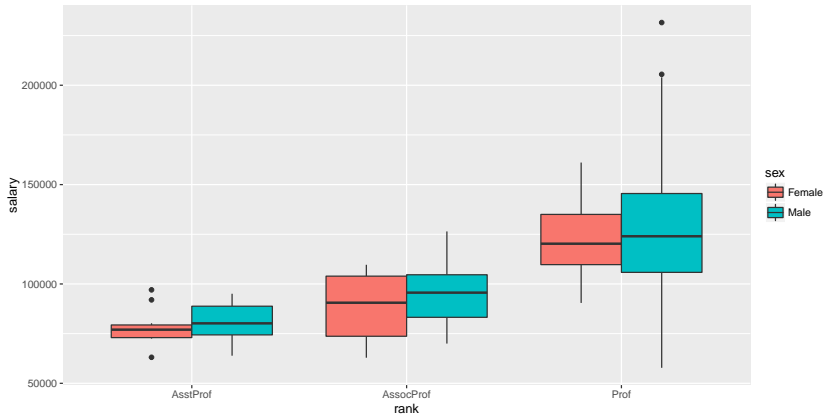
Adding smoothed lines

```
data(Salaries, package="car")
a<-ggplot(data=Salaries, aes(x=yrs.since.phd, y=salary)) +
  geom_smooth() + geom_point()
b<-ggplot(data=Salaries, aes(x=yrs.since.phd, y=salary,
  linetype=sex, shape=sex, color=sex)) +
  geom_smooth(method=lm, formula=y~poly(x,2),se=F,size=2)+
  geom_point(size=2,alpha=0.5) # quadratic
library(gridExtra);grid.arrange(a,b,ncol=2)
```



Modifying axes

```
data(Salaries, package="car")  
library(ggplot2)  
ggplot(Salaries, aes(x=rank, y=salary, fill=sex)) +  
  geom_boxplot()
```



Modifying axes



Modifying axes

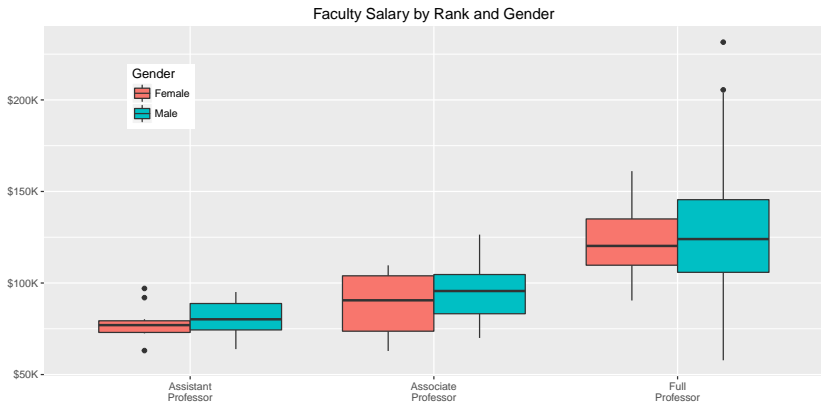
Function	Options
<code>scale_x_continuous()</code> , <code>scale_y_continuous()</code>	<code>breaks=</code> specifies tick marks, <code>labels=</code> specifies labels for tick marks, and <code>limits=</code> controls the range of the values displayed.

Function	Options
<code>scale_x_discrete()</code> , <code>scale_y_discrete()</code>	<code>breaks=</code> places and orders the levels of a factor, <code>labels=</code> specifies the labels for these levels, and <code>limits=</code> indicates which levels should be displayed.
<code>coord_flip()</code>	Reverses the x and y axes.

Modifying axes

```
data(Salaries,package="car")
library(ggplot2)
ggplot(data=Salaries, aes(x=rank, y=salary,fill=sex)) +
  geom_boxplot() +
  scale_x_discrete(breaks=c("AsstProf", "AssocProf", "Prof"),
                  labels=c("Assistant\nProfessor",
                           "Associate\nProfessor",
                           "Full\nProfessor")) +
  scale_y_continuous(breaks=c(50000, 100000, 150000, 200000),
                    labels=c("$50K", "$100K", "$150K", "$200K")) +
  labs(title="Faculty Salary by Rank and Sex", x="", y="")
```

Legends

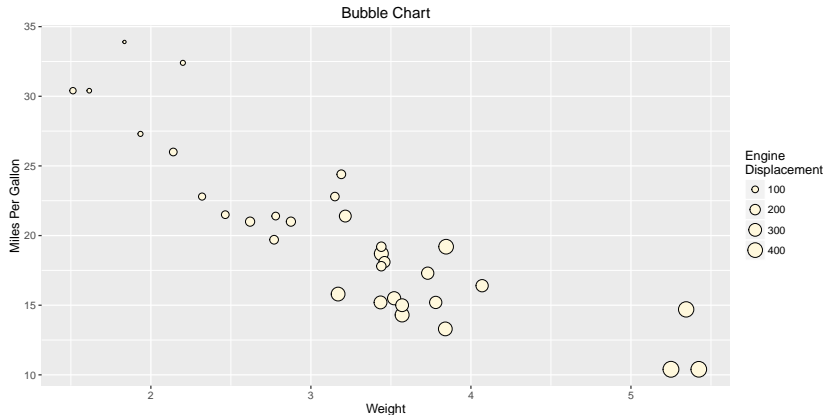


Legends

```
data(Salaries,package="car")
ggplot(data=Salaries, aes(x=rank, y=salary, fill=sex)) +
  geom_boxplot() +
  scale_x_discrete(breaks=c("AsstProf", "AssocProf", "Prof"),
                  labels=c("Assistant\nProfessor",
                          "Associate\nProfessor",
                          "Full\nProfessor")) +
  scale_y_continuous(breaks=c(50000, 100000, 150000, 200000),
                    labels=c("$50K", "$100K", "$150K", "$200K")) +
  labs(title="Faculty Salary by Rank and Gender",
       x="", y="", fill="Gender") +
  theme(legend.position=c(.15,.8))
```

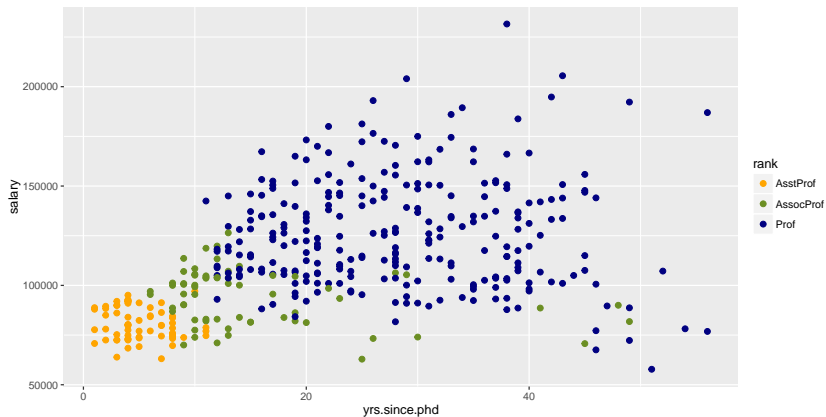
Scale

```
# table(mtcars$disp)
ggplot(mtcars, aes(x=wt, y=mpg, size=disp)) +
  geom_point(shape=21, color="black", fill="cornsilk") +
  labs(x="Weight", y="Miles Per Gallon",
  title="Bubble Chart", size="Engine\nDisplacement")
```



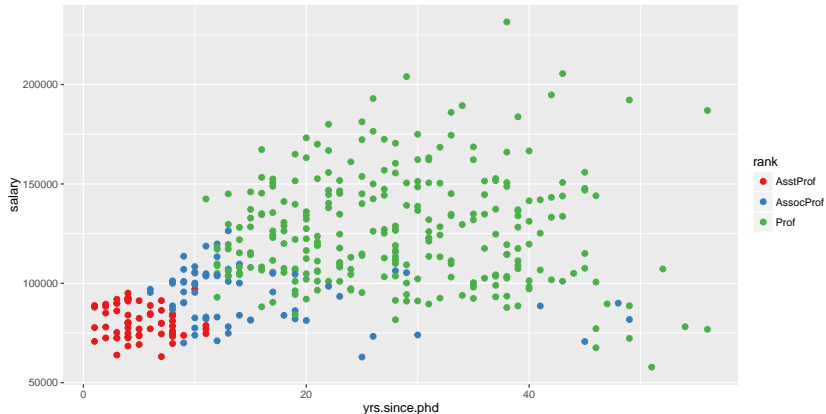
Scales

```
data(Salaries, package="car")  
ggplot(data=Salaries, aes(x=yrs.since.phd, y=salary, color=rank)) +  
  geom_point(size=2) +  
  scale_color_manual(values=c("orange", "olivedrab", "navy"))
```



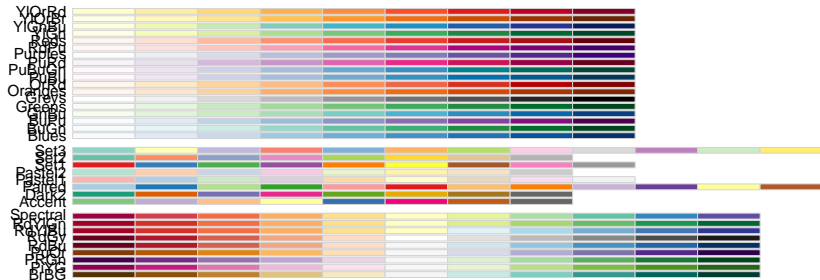
Scales

```
data(Salaries, package="car")  
ggplot(Salaries, aes(x=yrs.since.phd, y=salary, color=rank)) +  
  scale_color_brewer(palette="Set1") +  
  geom_point(size=2)
```



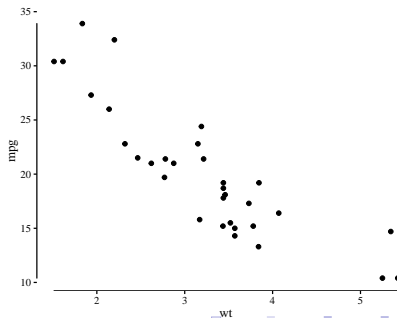
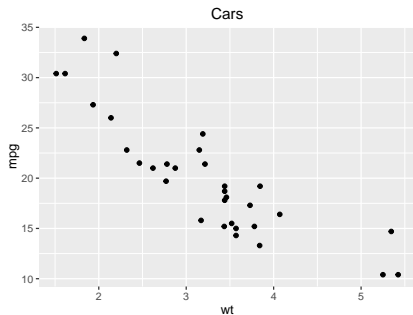
Scales

```
library(RColorBrewer);display.brewer.all()
```



Themes

```
library(ggthemes)
p1<-ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +labs(title='Cars')
p2<-ggplot(mtcars, aes(wt, mpg)) +
  geom_point() + geom_rangeframe() +
  theme_tufte()
library(gridExtra);grid.arrange(p1,p2,ncol=2)
```

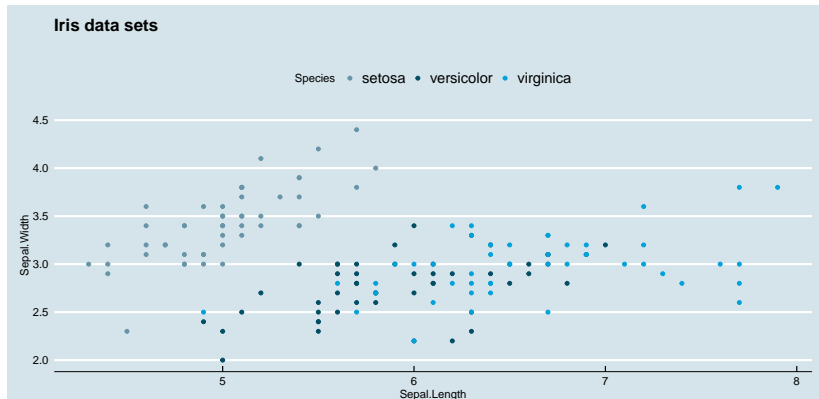


Themes



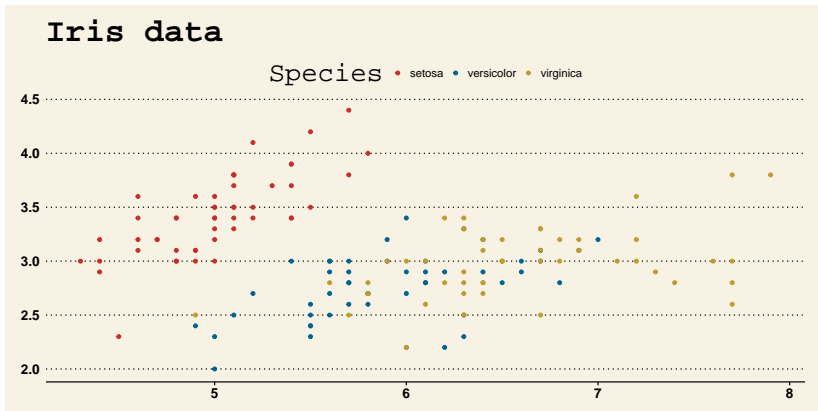
Economist template

```
p<-ggplot(iris,aes(Sepal.Length,Sepal.Width,colour = Species))+ geom_point(size=1.3)
p + theme_economist() +
scale_color_economist()+
ggtitle("Iris data sets")
```



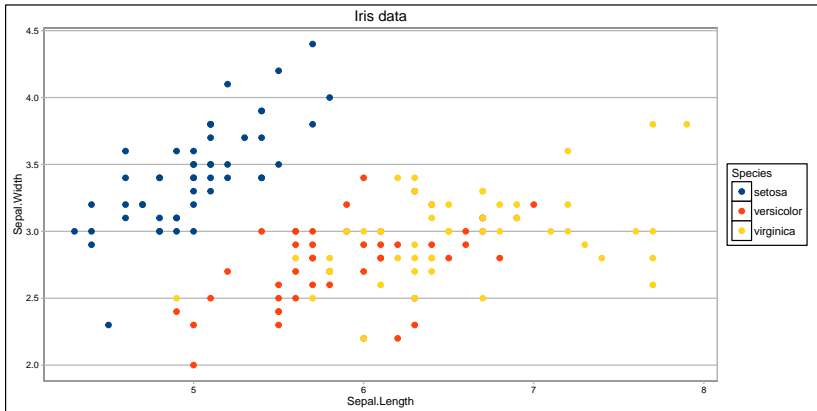
Wall Street Journal template

```
p<-ggplot(iris,aes(Sepal.Length,Sepal.Width,colour  
= Species))+ geom_point(size=1.3)  
p + theme_ws() + scale_colour_ws("colors6")+  
ggtitle("Iris data")# rgby, red_green, black_green, dem_req
```



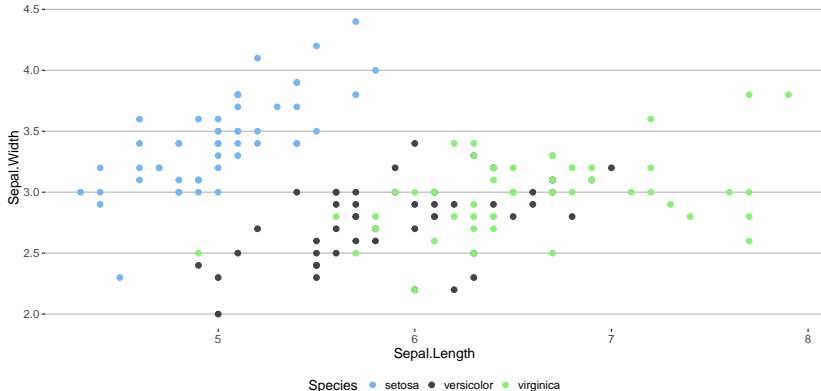
Google Docs

```
p<-ggplot(iris,aes(Sepal.Length,Sepal.Width,colour
= Species))+ geom_point(size=2)
p + theme_calc()+ scale_colour_calc()+
ggtitle("Iris data")
```



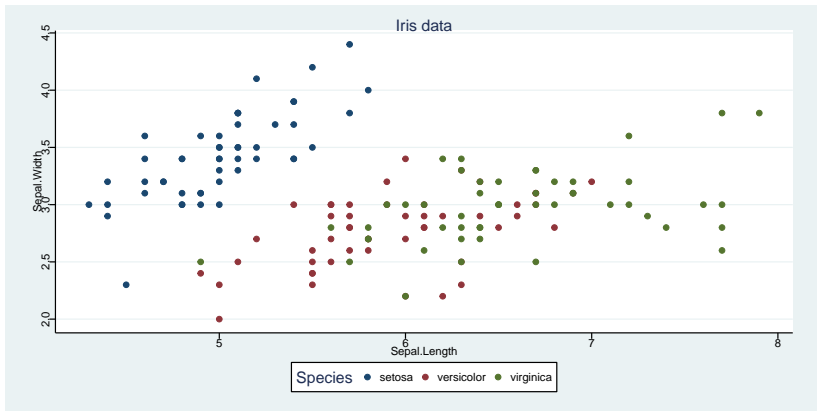
Highcharts JS

```
p<-ggplot(iris,aes(Sepal.Length,Sepal.Width,colour = Species))+ geom_point(size=2)  
p + theme_hc()+ scale_colour_hc()
```



Stata template

```
p<-ggplot(iris,aes(Sepal.Length,Sepal.Width,colour= Species  
p + theme_stata() + scale_color_stata() +  
ggtitle("Iris data"))
```



ggplot2 books

- ▶ R Graphics Cookbook (Winston Chang)
- ▶ ggplot2: Elegant Graphics for Data Analysis (Hadley Wickham)