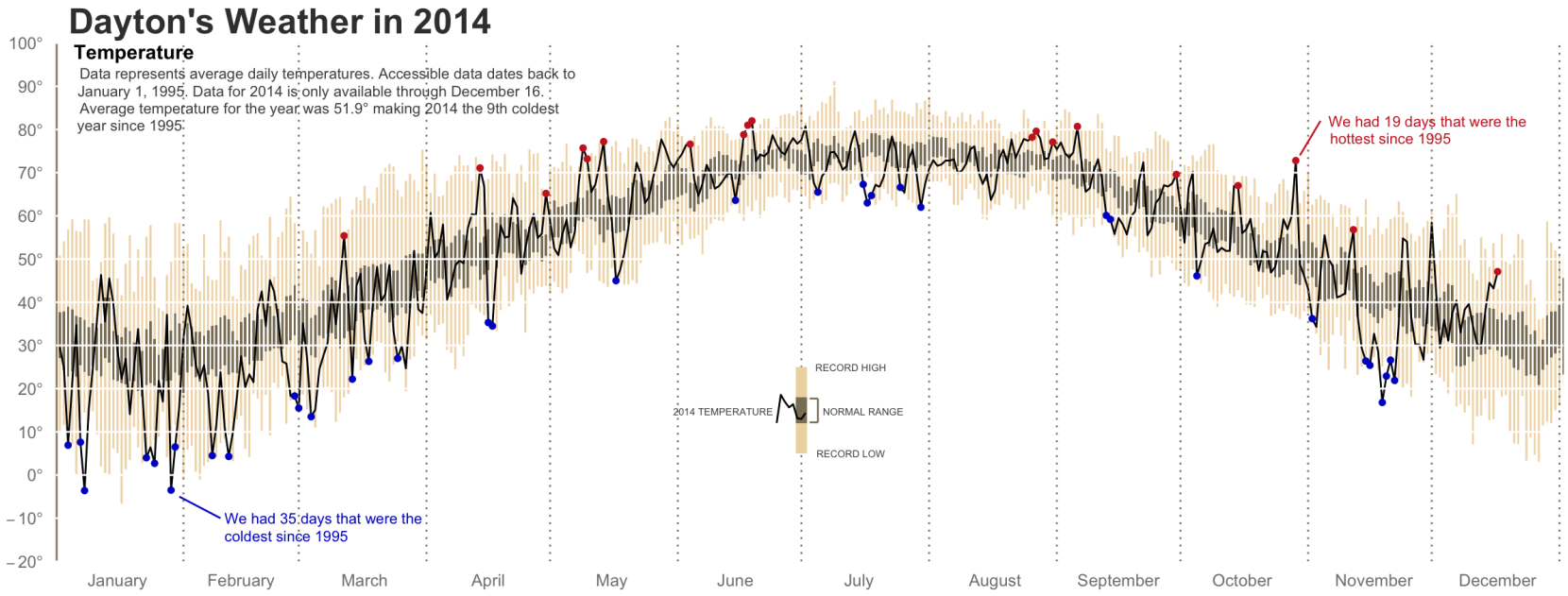
Basics for Enhanced Visualization: 3D/Data Grammar of graphics



Rodrigo Cabral

Polytech Nice - Data Science

cabral@unice.fr

Source: `https://rpubs.com/bradleyboehmke/weather_graphic`

Outline

These slides were partially inspired from

- Wickham, H. (2010). A layered grammar of graphics. Journal of Computational and Graphical Statistics, 19(1), 3-28.

How to build a plot

- Draw scatter plot of *A vs C* indicating categorical variable *D*:

Data

A	B	C	D
2	3	4	a
1	2	1	a
4	5	15	b
9	10	80	b

How to build a plot

- Draw scatter plot of *A vs C* indicating categorical variable *D*:

Data

A	B	C	D
2	3	4	a
1	2	1	a
4	5	15	b
9	10	80	b

- One solution:
  - Draw a point for each observation.
  - Horizontal position given by A, vertical position by B.
  - D specifies the shape of the point.

How to build a plot

- First step:
  create a new dataset with the mappings of the visual **aesthetics** variables and the values of the variables to be displayed.

x	y	Shape
2	4	a
1	1	a
4	15	b
9	80	b

- All other variables are deleted.

How to build a plot

- Second step:
  choose **geometries** for rendering visually the data.

- In our case the geometry is point. So that we have a scatter plot.
- But we can use lines
  bars
  texts
  paths
  ribbons

How to build a plot

- Third step:
  give **scales** to the variables from data units to physical units so that the computer know how to display it:

  size (in pixels or mm), angles, symbols, colors.

How to build a plot

- Third step:
  give **scales** to the variables from data units to physical units so that the computer know how to display it:

  size (in pixels or mm), angles, symbols, colors.

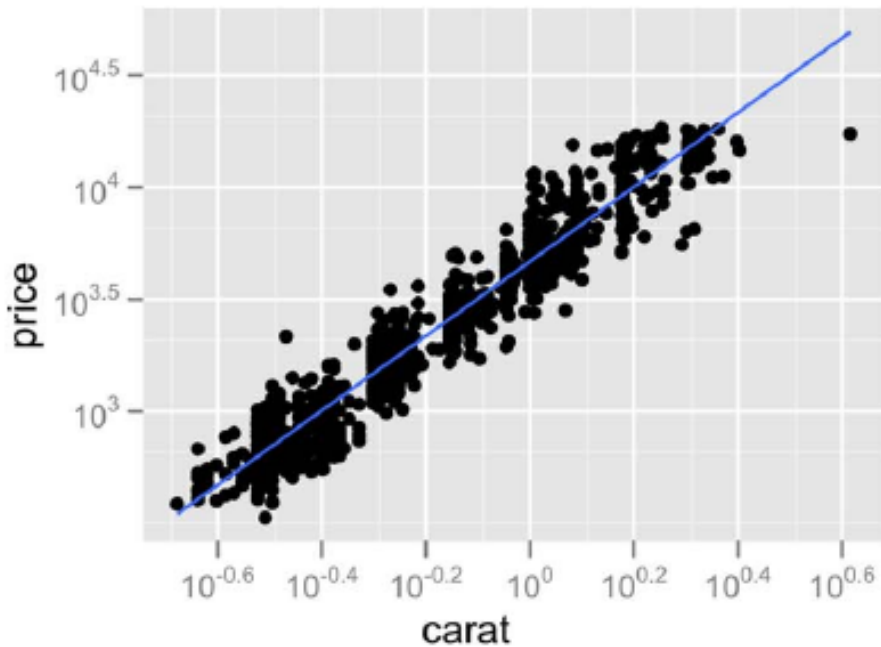
- For example: $x := \text{round}\left(\frac{x - \min(x)}{\text{range}(x)} \times \text{graphic width}\right)$ pixels
- Scale x to $[0, 200]$, y to $[0, 300]$, “a" to “•" and “b" to “■"

x	y	Shape
25	11	•
0	0	•
75	53	■
200	300	■

How to build a plot

- Third step:
  give **scales** to the variables from data units to physical units so that a computer know how to display it:
  length (in pixels or mm), area, angle, symbol, color.

- We can also use non linear scales such as log on each variable:



How to build a plot

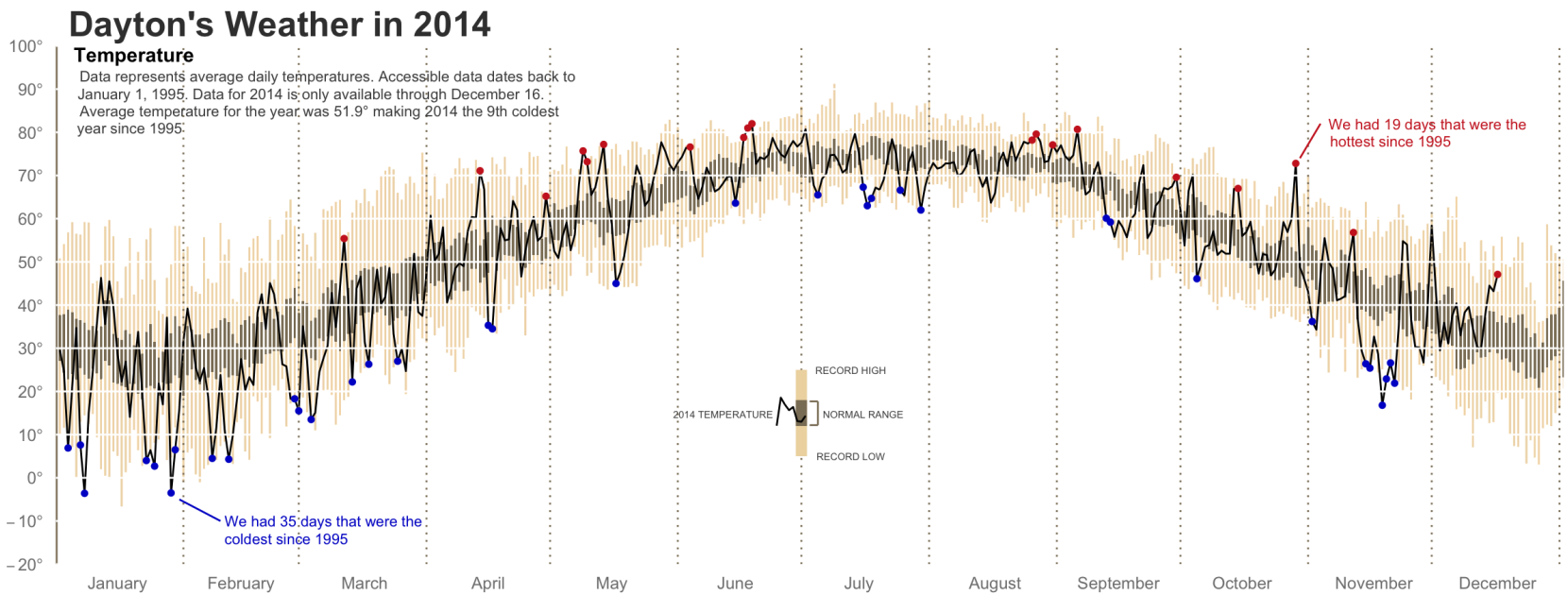
- Fourth step:
  use **statistical transformations** to generate new variables.
- This can be, for example, binning or aggregation:

  counts, maximum, minimum, average, standard deviation *etc.*
- In this simple example we will show directly the data:
  transformation is identity.

Introduction

How to build a plot

- A graphic with both data and statistical transformations:

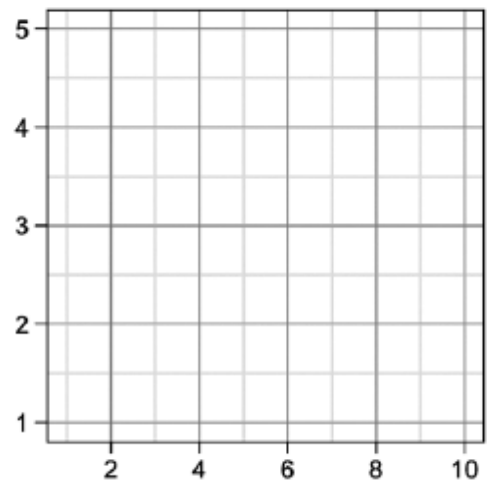


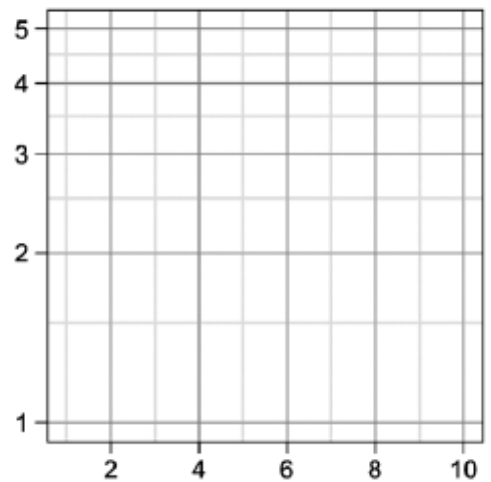
How to build a plot

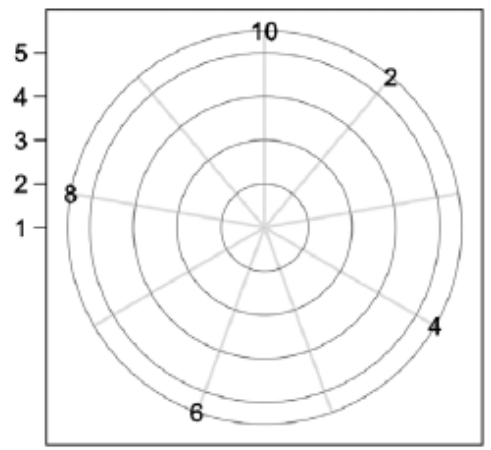
- Fifth step:
  choose **coordinates** system of the graphic:

  Cartesian, log, semi-log, polar, map projection *etc*.

- Cartesian, semi-log and polar coordinates:

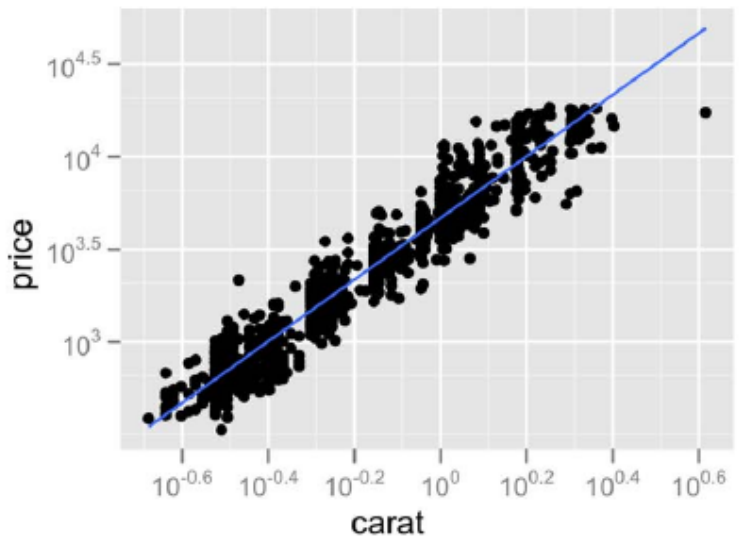


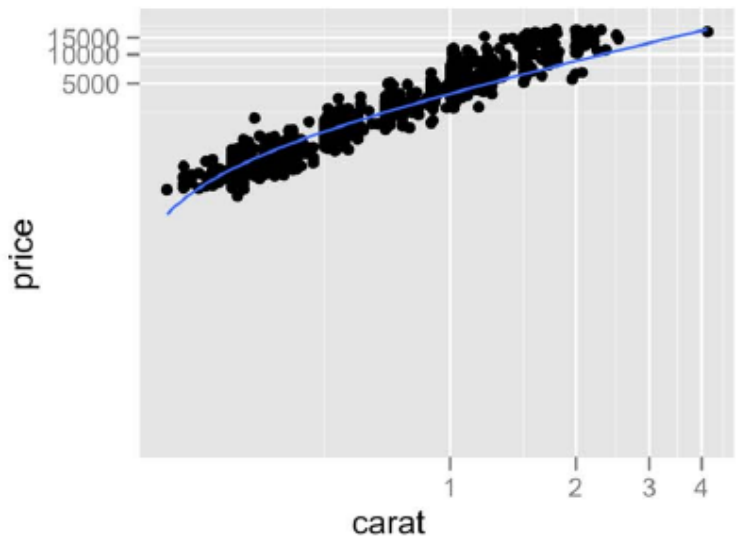




How to build a plot

- Note that in some cases changing the scale (left) or the coordinates (right) have the same effect on displaying data.





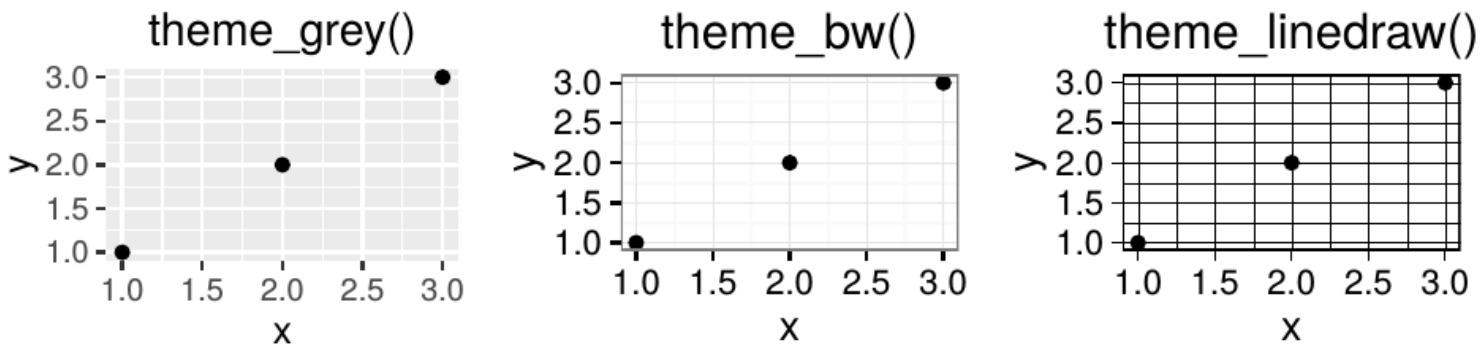
- The difference appears on the way the scales are shown.

How to build a plot

- Sixth step:
  indicate additional non-data information

  title, small description of the context, names of the axes, units *etc*.

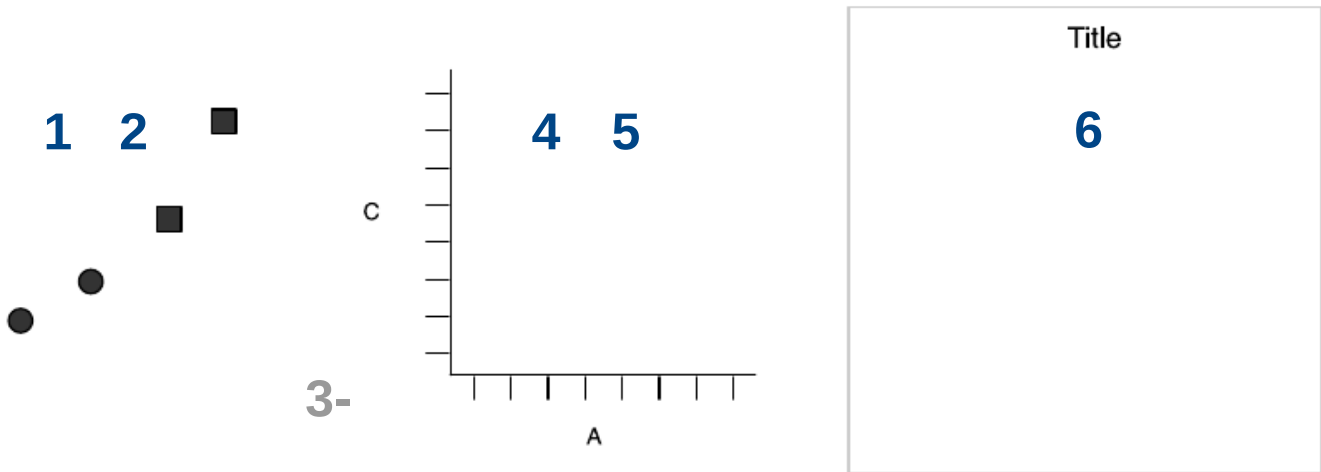
- The non-data part of the graphic is called a **theme**.



Source: Wickham, H. (2016). ggplot2: elegant graphics for data analysis. Springer.

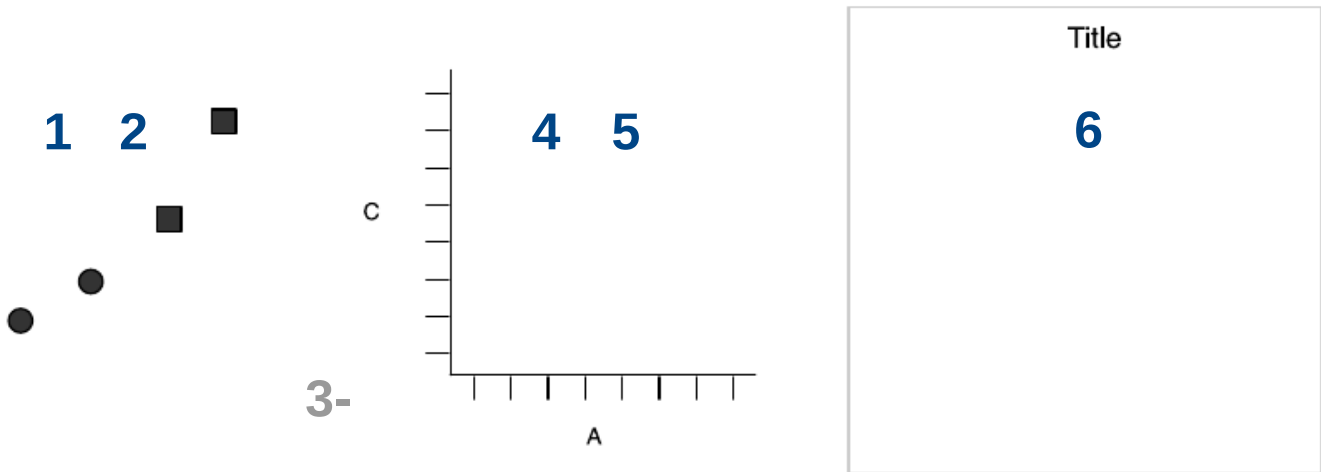
How to build a plot

- All six steps together: 1- aesthetics, 2- geometries, 3- statistics, 4- scales, 5- coordinates, 6- themes

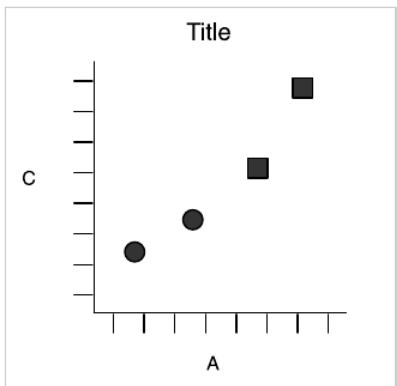


How to build a plot

- All six steps together: 1- aesthetics, 2- geometries, 3- statistics, 4- scales, 5- coordinates, 6- themes



- generate a graphic



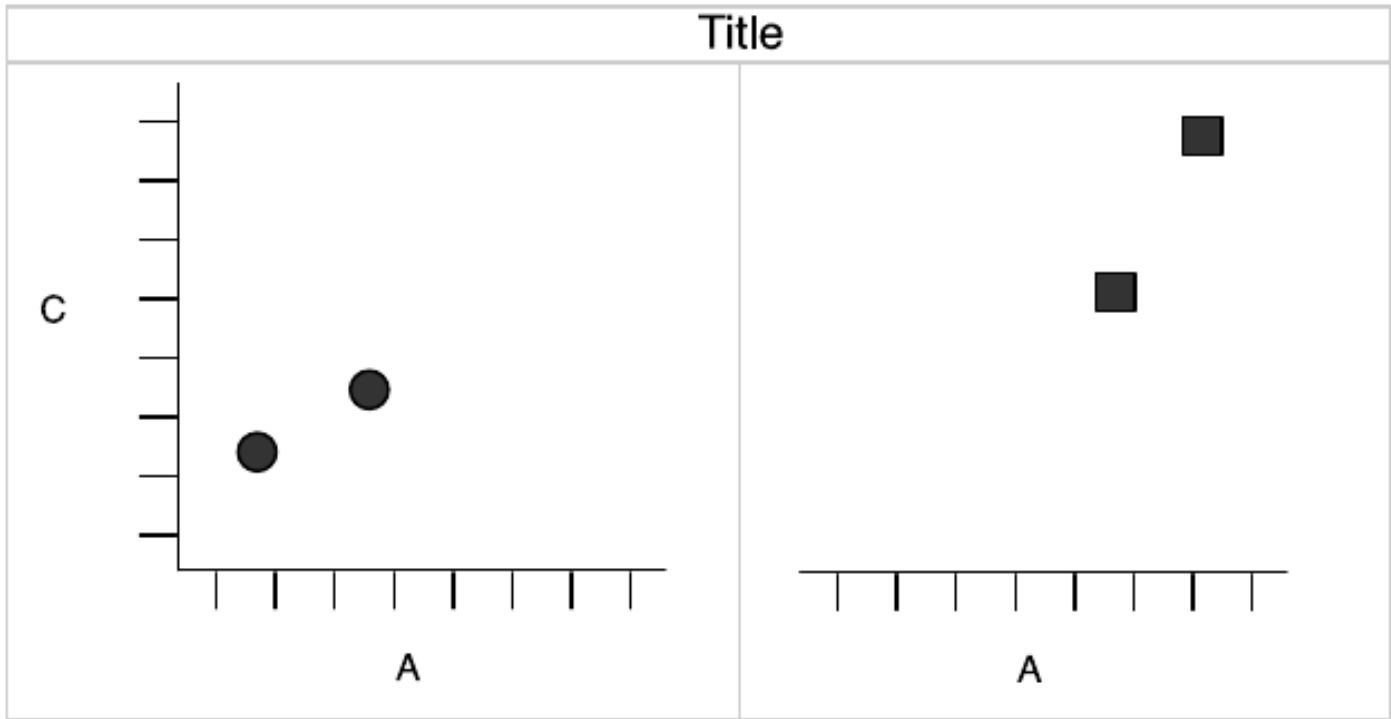
A more complex plot

- Consider that we want to generate different plots of the previous data conditioned on the variable D.
- This will lead us to create two **facets** of the data, *i.e.* two plots using subsets of the data set. See the table:

	x	y	Shape
Facet 1	25	11	•
	0	0	•
Facet 2	75	53	■
	200	300	■

A more complex plot

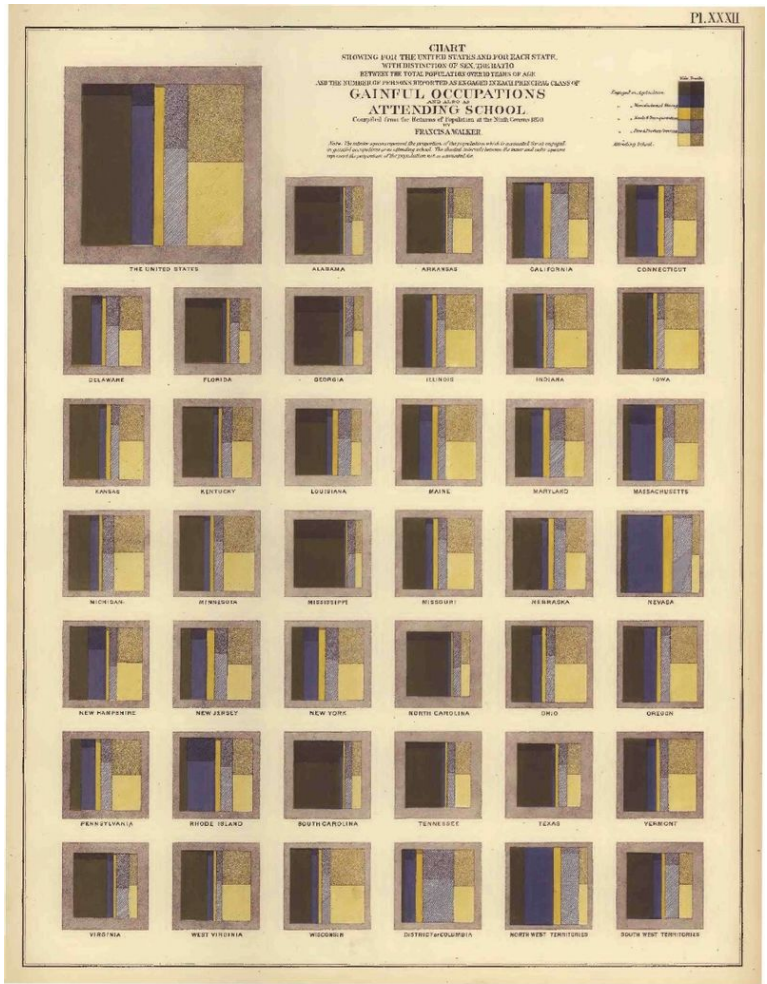
- The plot with faceting:



- Note that all previous specifications can remain unchanged.
- Scales are normally specified globally, so that the facets can be compared.

A more complex plot

- Scales are normally specified globally, so that the facets can be compared.
- This is a way to describe **small multiples**: the same graphic pattern is used to different data.



Source: Francis A. Walker - Statistical Atlas Of The United States based on the results of the Ninth Census 1870, as reproduced in Fraser's Federal Reserve Archive. https://fraser.stlouisfed.org/scribd/?title_id=64&filepath=/docs/publications/stat1870/Stat_Atlas1870.pdf#scribd-open, Public Domain, https://commons.wikimedia.org/w/index.php?curid=48396812

A grammar of graphics

The words of a graphic

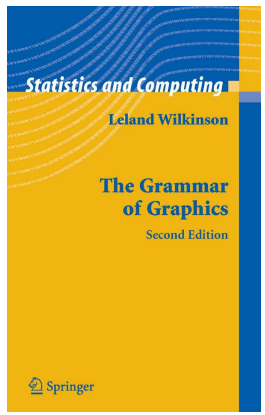
- Joining data, we can design a plethora of graphics.
  - aesthetics
  - geometries
  - statistics
  - scales
  - coordinates
  - facets
  - themes

- These 8 elements can be changed almost independently one from the other. Similarly to a sentence construction with words:

The red bird is sitting in a tree.
The blue bird is sitting on a branch of the tree.
The yellow bird is singing on the top the tree.

The grammar of graphics

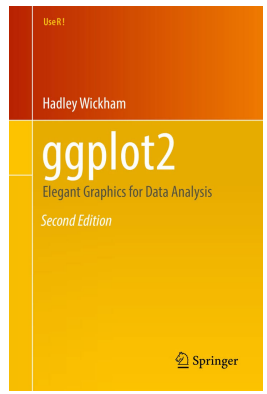
- These 8 elements can be changed almost independently one from the other.
- This is the reason this approach to construct graphics is called **grammar of graphics**.
- It was proposed originally in
  The Grammar of Graphics. L. Wilkinson. Springer Science & Business Media, 1999.



Source: `books.google.com`

The grammar of graphics

- Further developed as the 8 elements approach presented here in the book
  ggplot2 : Elegant Graphics for Data Analysis. H. Wickham. Springer, 2009.



- With the corresponding R package.
- It is also the base for *Tableau*, *IBM SPSS GPL* and Python's *Bokeh* package.

Source: `books.google.com`

A grammar of graphics

Examples

Pie chart

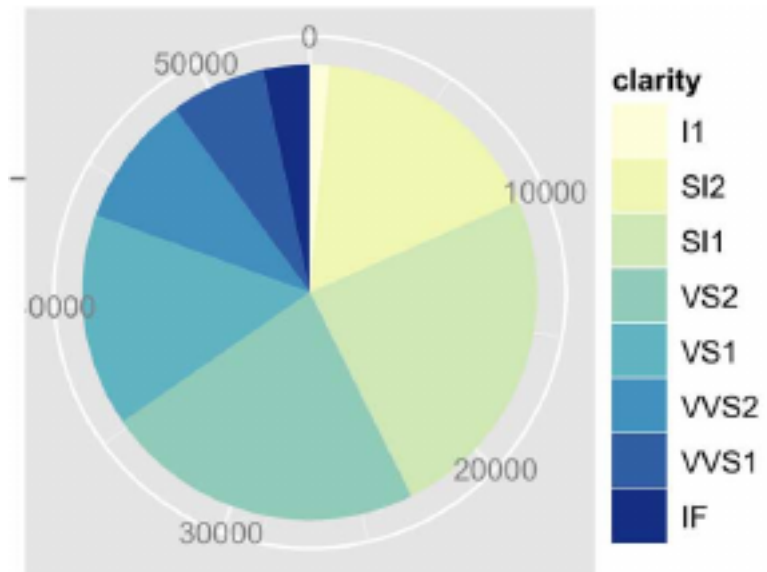
data	object type $\times$ quantities
aesthetics	object type - color quantities - y
geometries	stacked bar vertical
statistics	identity
scales	y - linear color - integers
coordinates	polar y - θ
facets	none
themes	none

A grammar of graphics

Examples

Pie chart

data	object type $\times$ quantities
aesthetics	object type - color quantities - y
geometries	stacked bar vertical
statistics	identity
scales	y - linear color - integers
coordinates	polar y - θ
facets	none
themes	none



Examples

Bullseye chart

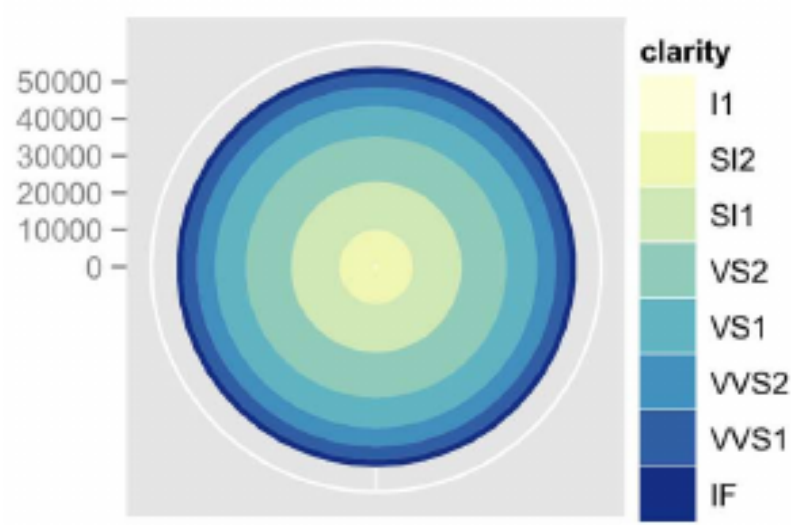
data	object type $\times$ quantities
aesthetics	object type - color quantities - y
geometries	stacked bar vertical
statistics	identity
scales	y - linear color - integers
coordinates	polar y - *r*
facets	none
themes	none

A grammar of graphics

Examples

Bullseye chart

data	object type × quantities
aesthetics	object type - color quantities - y
geometries	stacked bar vertical
statistics	identity
scales	y - linear color - integers
coordinates	polar y - *r*
facets	none
themes	none



Examples

Bar chart

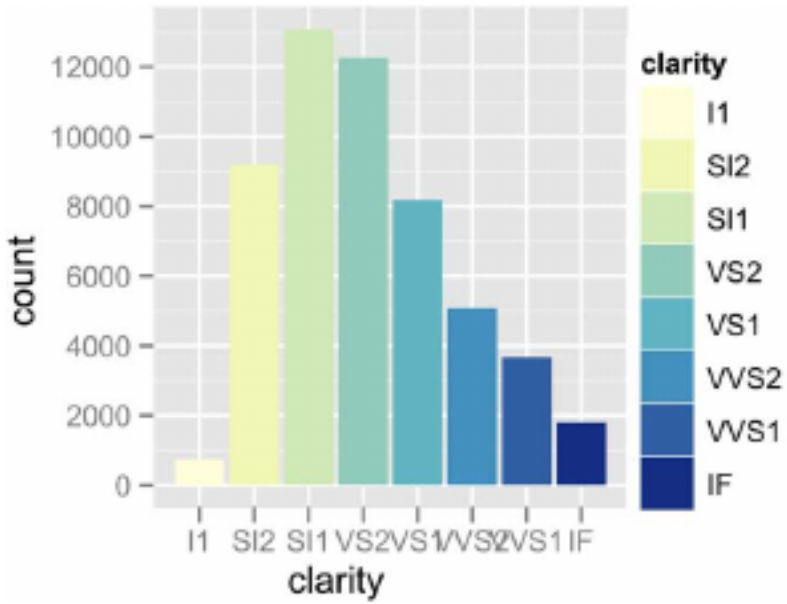
data	object type $\times$ quantities
aesthetics	object type - x quantities - y object type - color
geometries	bars vertical
statistics	identity
scales	x - integers y - linear color - integers
coordinates	Cartesian
facets	none
themes	none

A grammar of graphics

Examples

Bar chart

data	object type × quantities
aesthetics	object type - x quantities - y object type - color
geometries	bars vertical
statistics	identity
scales	x - integers y - linear color - integers
coordinates	Cartesian
facets	none
themes	none



Examples

Coxcomb chart

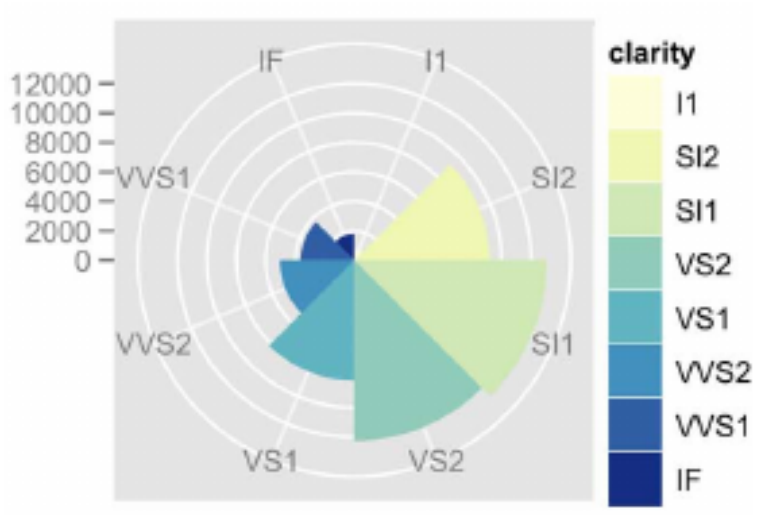
data	object type $\times$ quantities
aesthetics	object type - x quantities - y object type - color
geometries	bars vertical
statistics	identity
scales	x - integers y - linear color - integers
coordinates	Polar y - *r*
facets	none
themes	none

A grammar of graphics

Examples

Coxcomb chart

data	object type × quantities
aesthetics	object type - x quantities - y object type - color
geometries	bars vertical
statistics	identity
scales	x - integers y - linear color - integers
coordinates	Polar y - *r*
facets	none
themes	none



Examples

Histogram

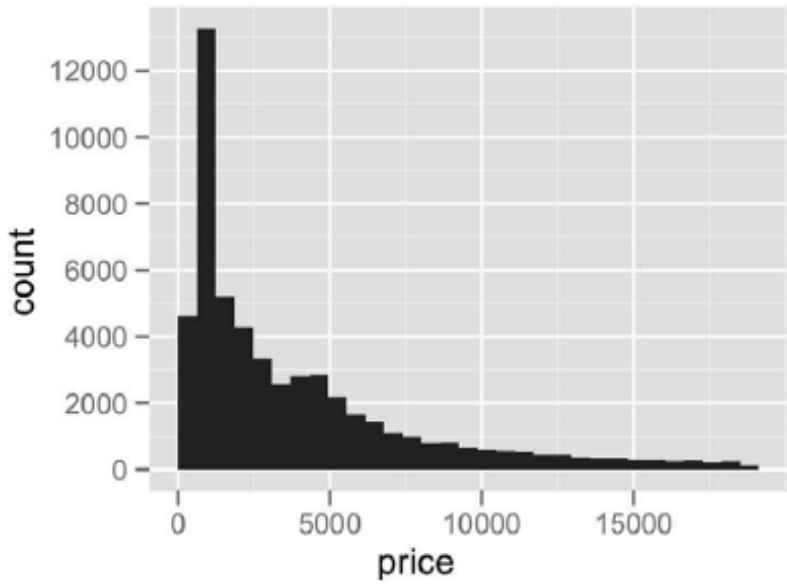
data	object × price
aesthetics	price - x statistic - y
geometries	bars vertical
statistics	count 30 uniform bins
scales	x - linear y - linear
coordinates	Cartesian
facets	none
themes	none

A grammar of graphics

Examples

Histogram

data	object × price
aesthetics	price - x statistic - y
geometries	bars vertical
statistics	count 30 uniform bins
scales	x - linear y - linear
coordinates	Cartesian
facets	none
themes	none



Examples

Density plot

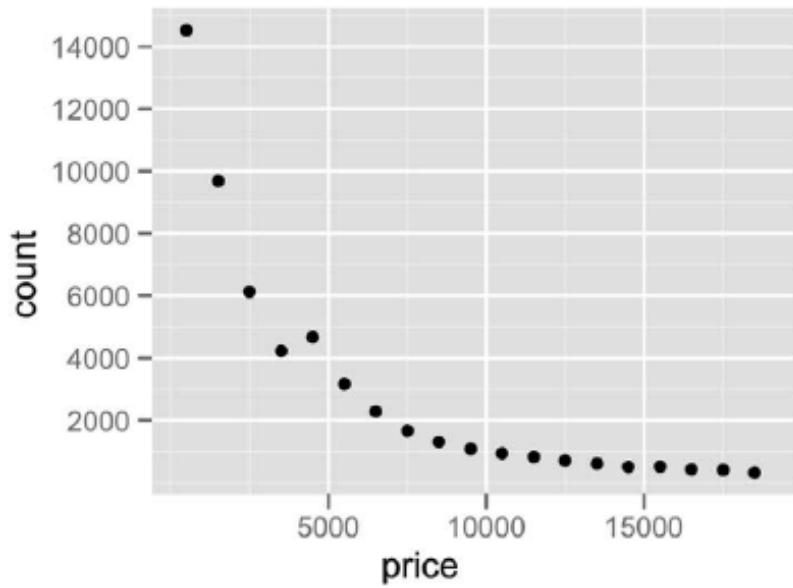
data	object × price
aesthetics	price - x
	statistic - y
geometries	points
statistics	density
	19 points
scales	x - linear
	y - linear
coordinates	Cartesian
facets	none
themes	none

A grammar of graphics

Examples

Density plot

data	object × price
aesthetics	price - x
	statistic - y
geometries	points
statistics	density
	19 points
scales	x - linear
	y - linear
coordinates	Cartesian
facets	none
themes	none



Examples

Filled density plot

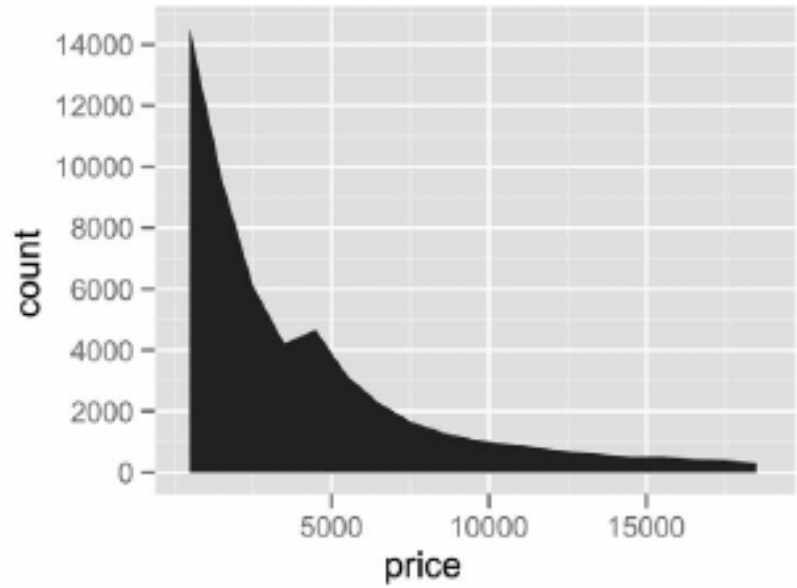
data	object × price
aesthetics	price - x statistic - y
geometries	ribbon vertical starting at 0
statistics	density 19 points
scales	x - linear y - linear
coordinates	Cartesian
facets	none
themes	none

A grammar of graphics

Examples

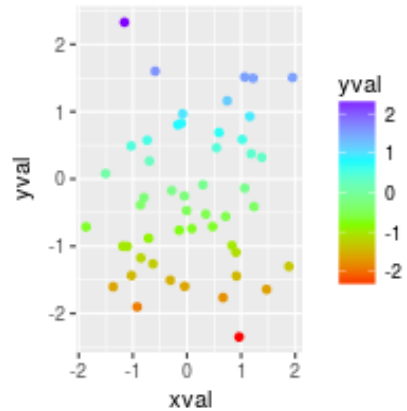
Filled density plot

data	object × price
aesthetics	price - x statistic - y
geometries	ribbon vertical starting at 0
statistics	density 19 points
scales	x - linear y - linear
coordinates	Cartesian
facets	none
themes	none



Axes and legends

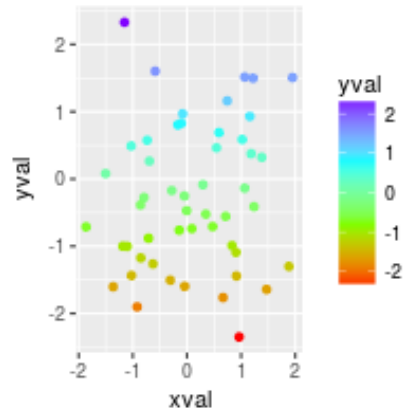
- Axes: guides indicating the inverse functions of scales of continuous variables or the spatial coordinates.



Source: `http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/`

Axes and legends

- Axes: guides indicating the inverse functions of scales of continuous variables or the spatial coordinates.



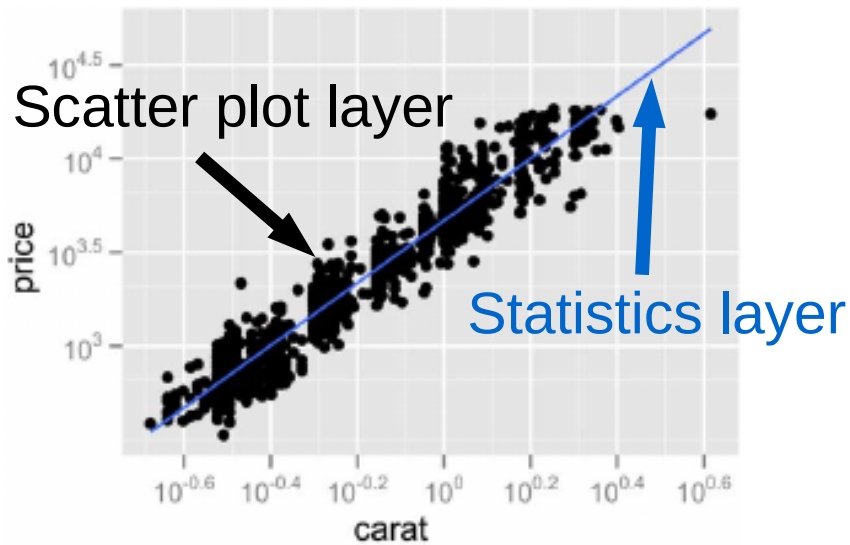
- Legends: guides indicating the inverse of scales of discrete variables.

Source: `http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/`

ggplot2

A layered grammar of graphics

- Implementation of the grammar of graphics in R.
- It is a layered implementation: we can add different plots to the same graphic as layers:



R code: ggplot and layer

- We start by defining a ggplot object:

  ggplot()+...

- then we add layers

  layer(geom=points, stat="identity", data=diamonds,
  mapping=aes(x = carat, y = price), position= "identity"**)**

- A layer has standard parameters:
  - geom: geometries
  - stat: statistics
  - position: positioning specification for geometric objects

R code: ggplot specification

- Data (data) and aesthetics mapping (mapping) can be either defined on the layer or passed directly to the ggplot object at the beginning.

  ggplot(data=diamonds,mapping=aes(x=carat,y=price)**)**+...

R code: ggplot specification

- Data (data) and aesthetics mapping (mapping) can be either defined on the layer or passed directly to the ggplot object at the beginning.

  ggplot(data=diamonds,mapping=aes(x=carat,y=price)**)**+...

- All subsequent layers will inherit the data and aesthetics used in ggplot.
- We can overwrite the default for a layer by respecifying data and aesthetics.

R code: ggplot and layer

- We finish by adding scales and coordinates:

```
scale_y_continuous() +
scale_x_continuous() +
  coord_cartesian()
```

R code: ggplot and layer

- We finish by adding scales and coordinates:

```
scale_y_continuous() +
scale_x_continuous() +
  coord_cartesian()
```

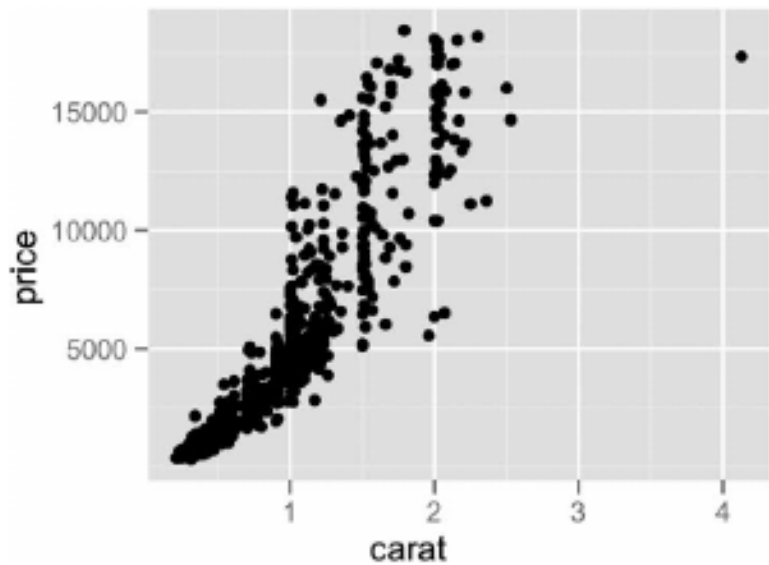
- ggplot2 has many default options and shortcuts to simplify plotting.
- All previous code can be rewritten using defaults and shortcuts:

ggplot(diamonds, aes(carat, price)**)** + **geom_point()**

Examples

Scatter plot

```
ggplot() +
layer(data = diamonds,
mapping = aes(x = carat, y = price),
geom = "point", stat = "identity",
position = "identity" ) +
scale_y_continuous() +
scale_x_continuous() +
coord_cartesian()
```



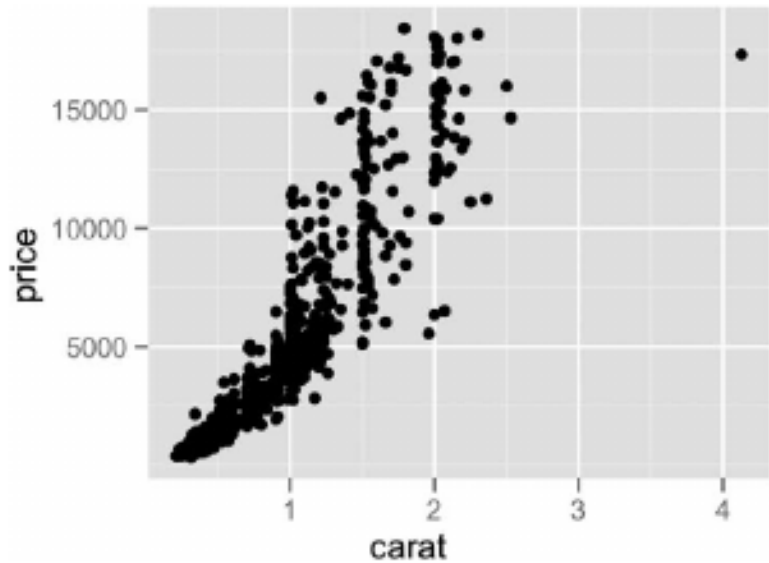
Examples

Scatter plot

```
ggplot() +
layer(data = diamonds,
mapping = aes(x = carat, y = price),
geom = "point", stat = "identity",
position = "identity" ) +
scale_y_continuous() +
scale_x_continuous() +
coord_cartesian()
```

or

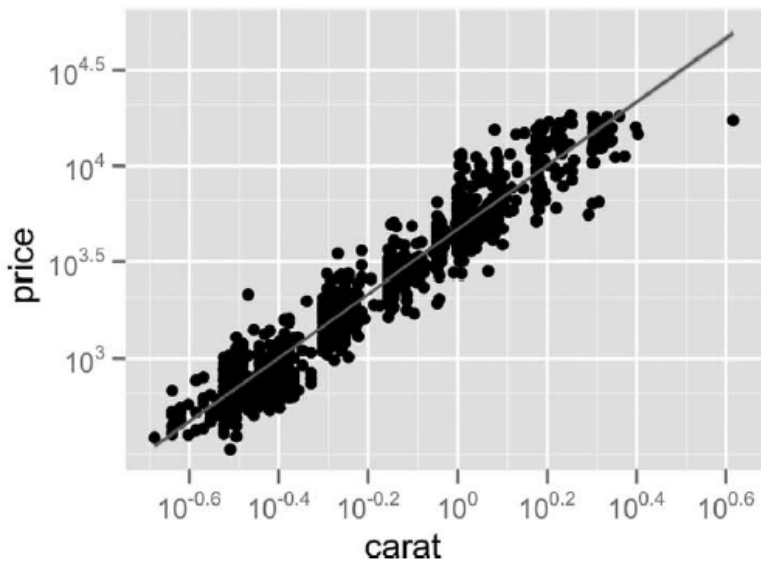
```
ggplot(diamonds, aes(carat, price)) +
geom_point()
```



Examples

Loglog scatter plot with linear regression

```
ggplot() +
layer(data = diamonds,
mapping = aes(x = carat, y = price),
geom = "point", stat = "identity",
position = "identity") +
layer(data = diamonds,
mapping = aes(x = carat, y = price),
geom = "smooth", position = "identity",
stat = "smooth", method = lm) +
scale_y_log10() + scale_x_log10() +
coord_cartesian()
```



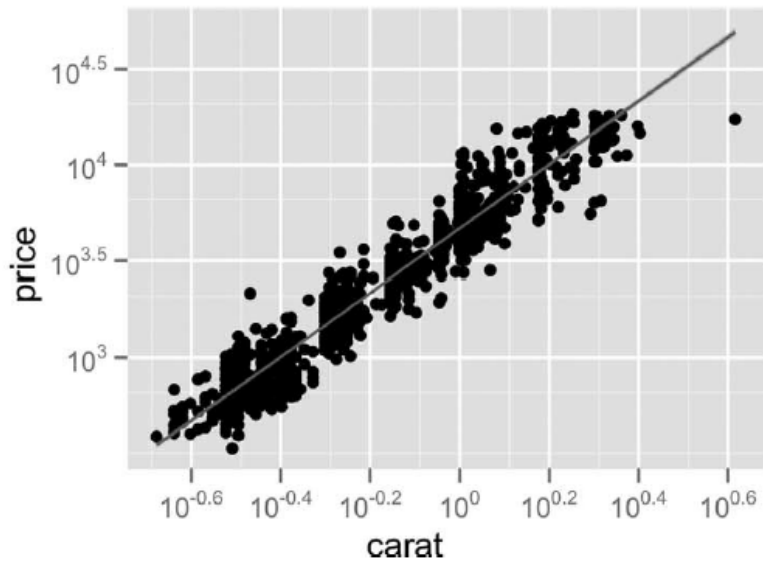
Examples

Loglog scatter plot with linear regression

```
ggplot() +
layer(data = diamonds,
mapping = aes(x = carat, y = price),
geom = "point", stat = "identity",
position = "identity") +
layer(data = diamonds,
mapping = aes(x = carat, y = price),
geom = "smooth", position = "identity",
stat = "smooth", method = lm) +
scale_y_log10() + scale_x_log10() +
coord_cartesian()
```

or

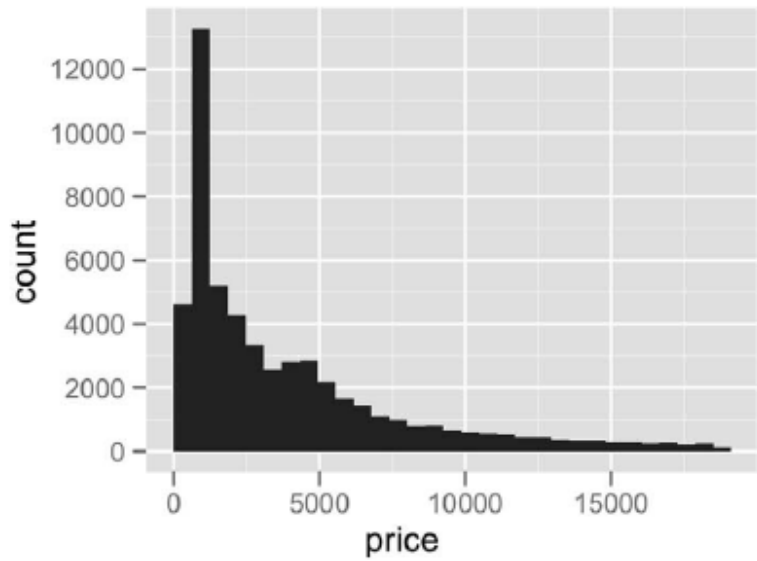
```
ggplot(diamonds, aes(carat, price)) +
geom_point() +
stat_smooth(method = lm) +
scale_x_log10() + scale_y_log10()
```



Examples

Histogram

ggplot(data = diamonds, mapping = aes(price)**)** +
layer(geom = "bar", stat = "bin", mapping = aes(y = ..count..)**)**



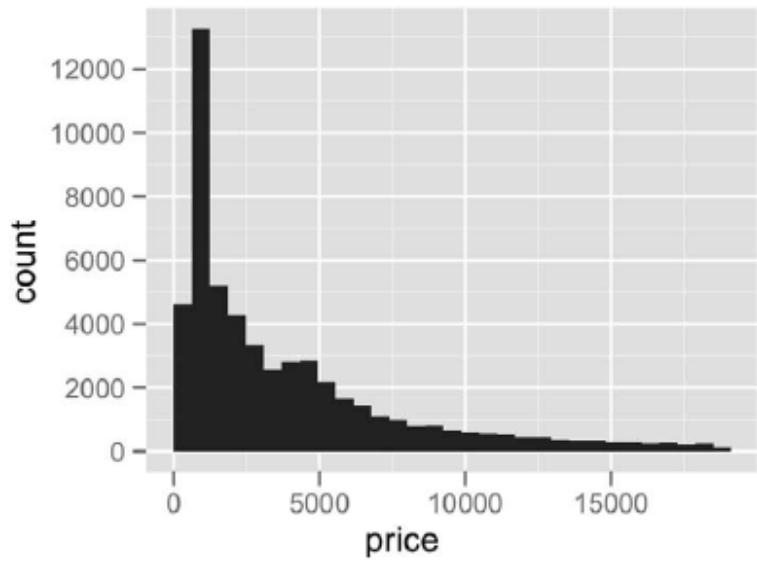
Examples

Histogram

ggplot(data = diamonds, mapping = aes(price)**)** +
layer(geom = "bar", stat = "bin", mapping = aes(y = ..count..)**)**

or

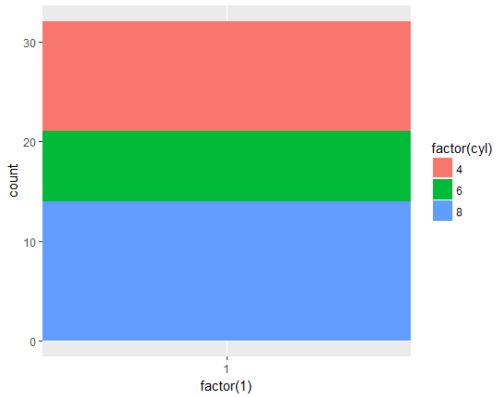
ggplot(diamonds, aes(x = price)**)** +
geom_histogram()



Examples

Stacked bar chart

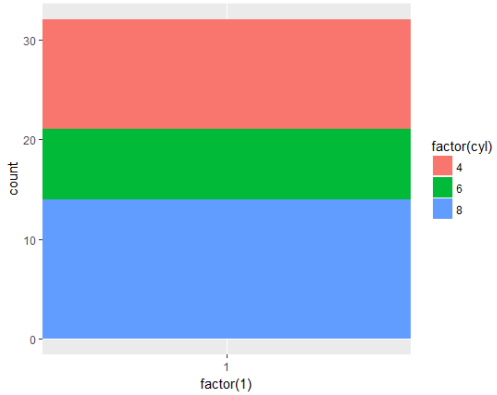
```
sbar <- ggplot(mtcars,
aes(x = factor(1),
fill = factor(cyl))) +
geom_bar(width = 1)
```



Examples

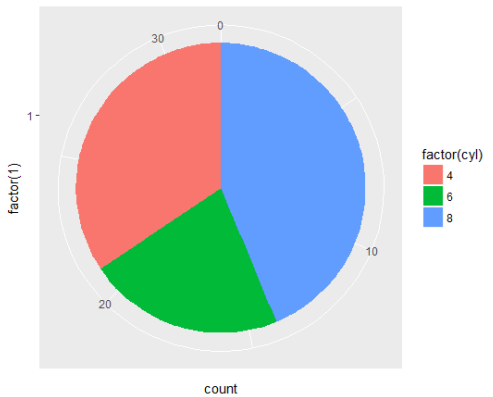
Stacked bar chart

```
sbar <- ggplot(mtcars,
aes(x = factor(1),
fill = factor(cyl))) +
geom_bar(width = 1)
```



Pie chart

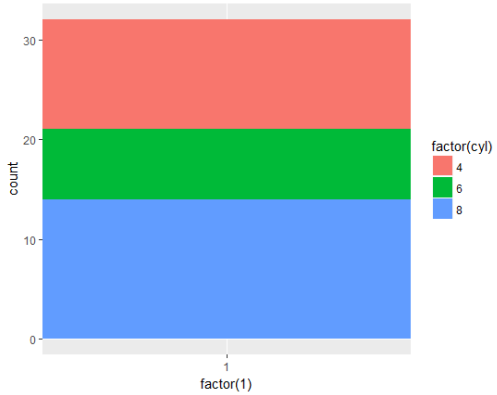
```
pie <- sbar + coord_polar("y")
```



Examples

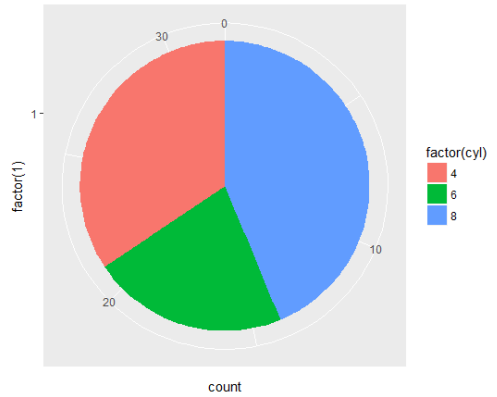
Stacked bar chart

```
sbar <- ggplot(mtcars,
aes(x = factor(1),
fill = factor(cyl))) +
geom_bar(width = 1)
```



Pie chart

```
pie <- sbar + coord_polar("y")
```



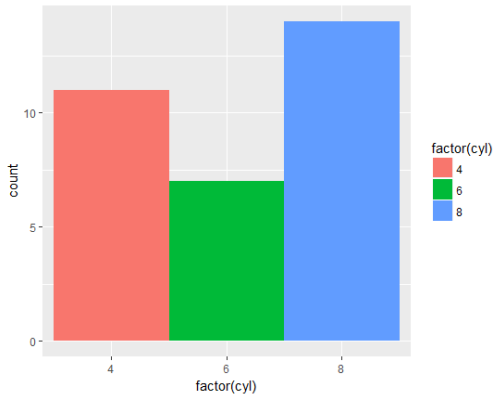
Bullseye chart

```
bull <- sbar + coord_polar("x")
```

Examples

Bar chart

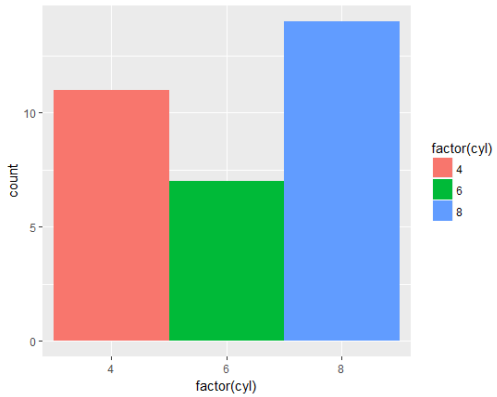
```
bar <- ggplot(mtcars,
aes(x = factor(cyl),
fill = factor(cyl))) +
geom_bar(width = 1)
```



Examples

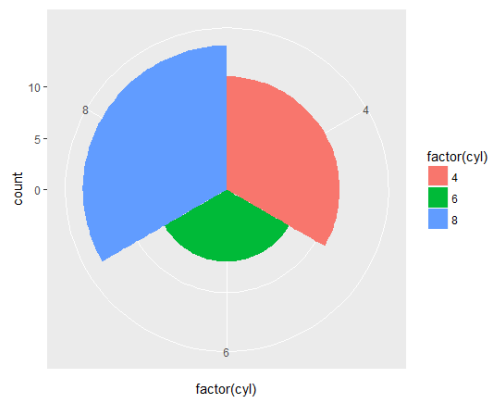
Bar chart

```
bar <- ggplot(mtcars,
aes(x = factor(cyl),
fill = factor(cyl))) +
geom_bar(width = 1)
```



Coxcomb chart

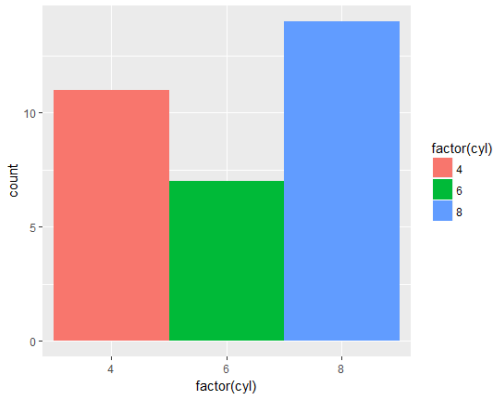
```
cxc <- bar + coord_polar("x")
```



Examples

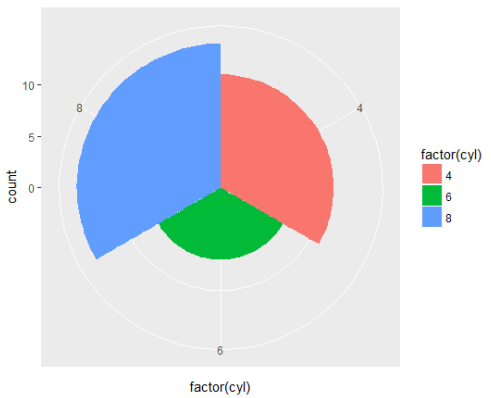
Bar chart

```
bar <- ggplot(mtcars,
aes(x = factor(cyl),
fill = factor(cyl))) +
geom_bar(width = 1)
```



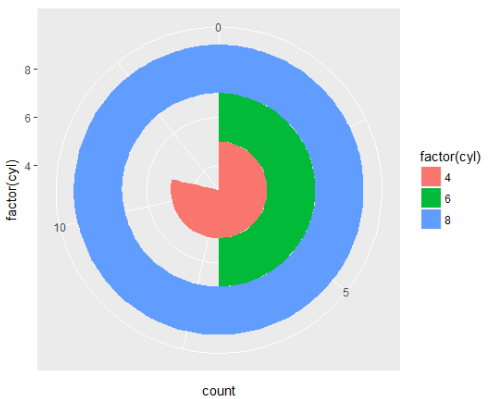
Coxcomb chart

```
cxc <- bar + coord_polar("x")
```



?????? chart

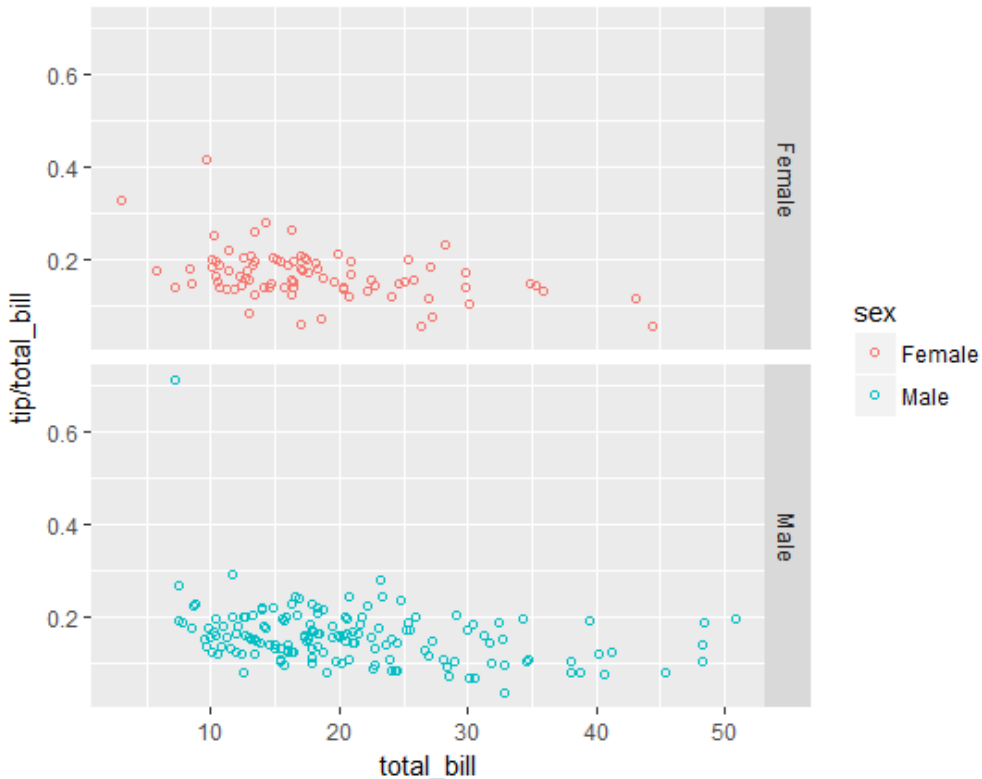
```
xxc <- bar + coord_polar("y")
```



Examples

Small multiples

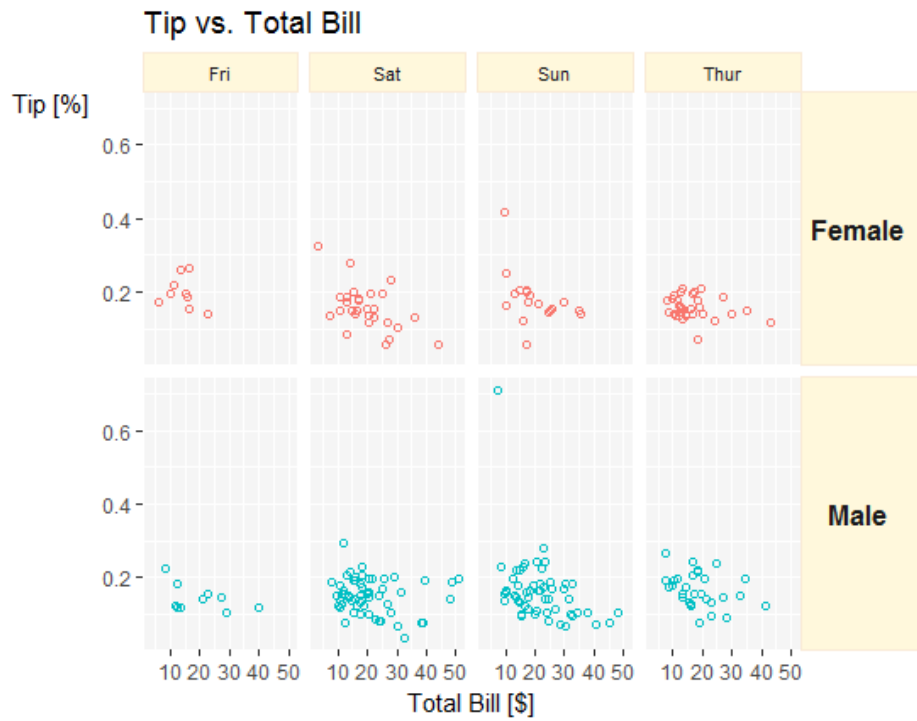
```
tips <- ggplot(tips,
aes(x=total_bill,
y=tip/total_bill,color=sex))=+
geom_point(shape=1) +
facet_grid(sex ~ .)
```



Examples

Theme

```
tips + facet_grid(sex ~ day) +
ggtitle("Tip vs. Total Bill") +
labs(x="Total Bill [$]",y="Tip [%]") +
theme(strip.text.x =
    element_text(size=8, angle=0),
strip.text.y =
    element_text(size=12, face="bold",
angle=0),
strip.background =
    element_rect(colour="antiquewhite",
        fill="cornsilk"),
panel.background =
    element_rect(fill = 'gray96',
        colour = 'gray100'),
axis.title.y =
    element_text(angle = 0),
legend.position="none")
```



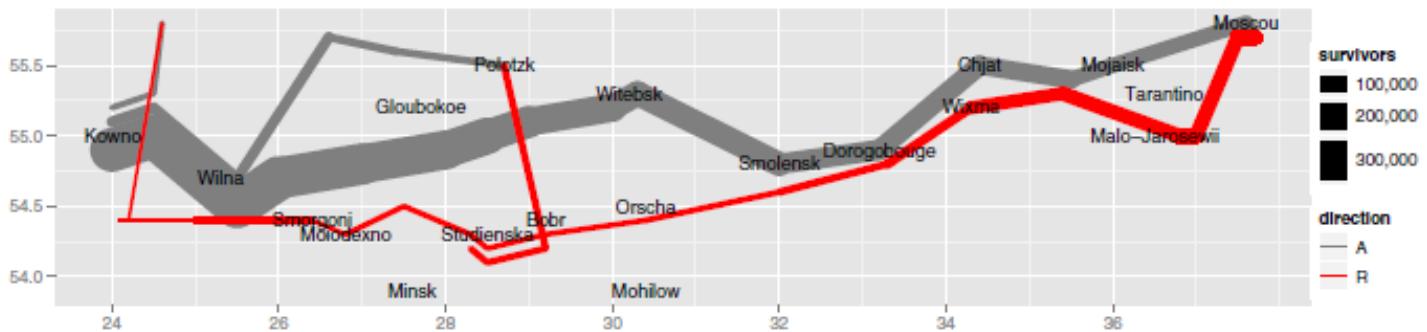
Examples

Napoleon's march

```
plot_troops <- ggplot(troops, aes(long, lat)) +
    geom_path(aes(size = survivors, color = direction, group = group))

plot_both <- troops_plot + geom_text(aes(label = city), size = 4, data = cities)

plot_polished <- both + scale_size(to = c(1, 10), breaks = c(1, 2, 3) * 10^5,
    labels = comma(c(1, 2, 3) * 10^5))+
    scale_color_manual(values = c("grey50","red")) +
xlab(NULL) + ylab(NULL)
```



Other resources

- R studio cheat sheet (available on Jalon):
  `https://www.rstudio.com/wp-content/uploads/2016/11/ggplot2-cheatsheet-2.1.pdf`
- R cookbook (Graphs):
  `http://www.cookbook-r.com/Graphs/`
- ggplot2 book website:
  `http://ggplot2.org/book/`

Conclusions

Conclusions

- Grammar of graphics ≠“zoo of charts" which is often proposed in any data processing software (Excel charts).
- Grammar approach: extend the effectiveness of our written communication to graphics.
- Fundamental idea: richness of the communication system comes from the structure, not only from the quantity of different symbols.

Conclusions

- First popular implementation: ggplot2, but others are appearing (Tableau).
- Still in its infancy: ggplot2 still has some “zoo of charts" and it does not take into account interaction.
- As any grammar it does not prevent us from constructing uggly phrases (graphics).