R programming for beginners

Ni Shuai

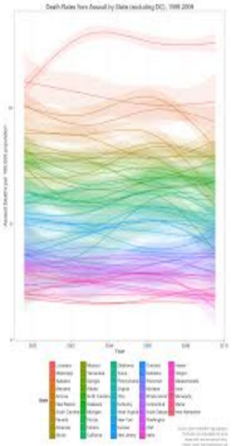
Computational Genome Biology
German Cancer Research Center (DKFZ)

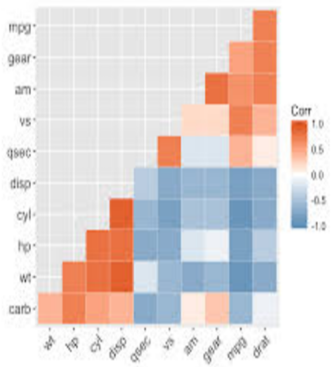
November, 2016

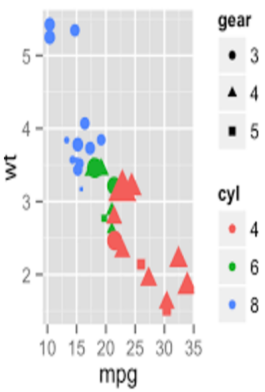
ggplot

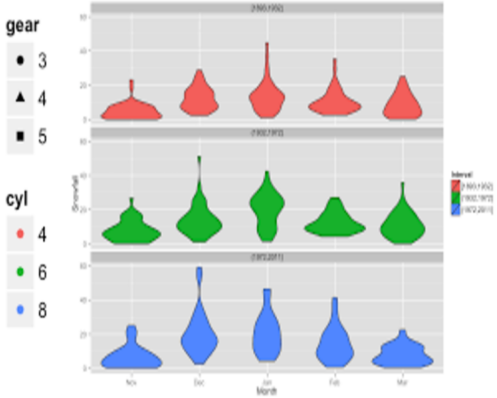
ggplot - a new plotting system in R

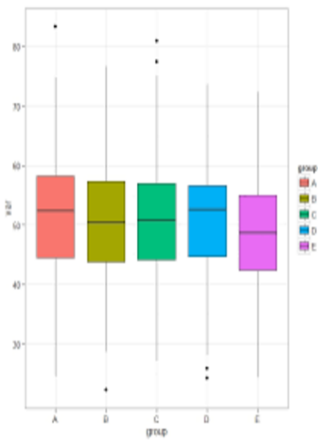
- Intrinsically nice looking
- Powerful and smart
- Complete plot system and consistent gramma
- Complicated with simple plot, simple with complicated plot
- Activly matained and developed











ggplot

Essencial components in a ggplot

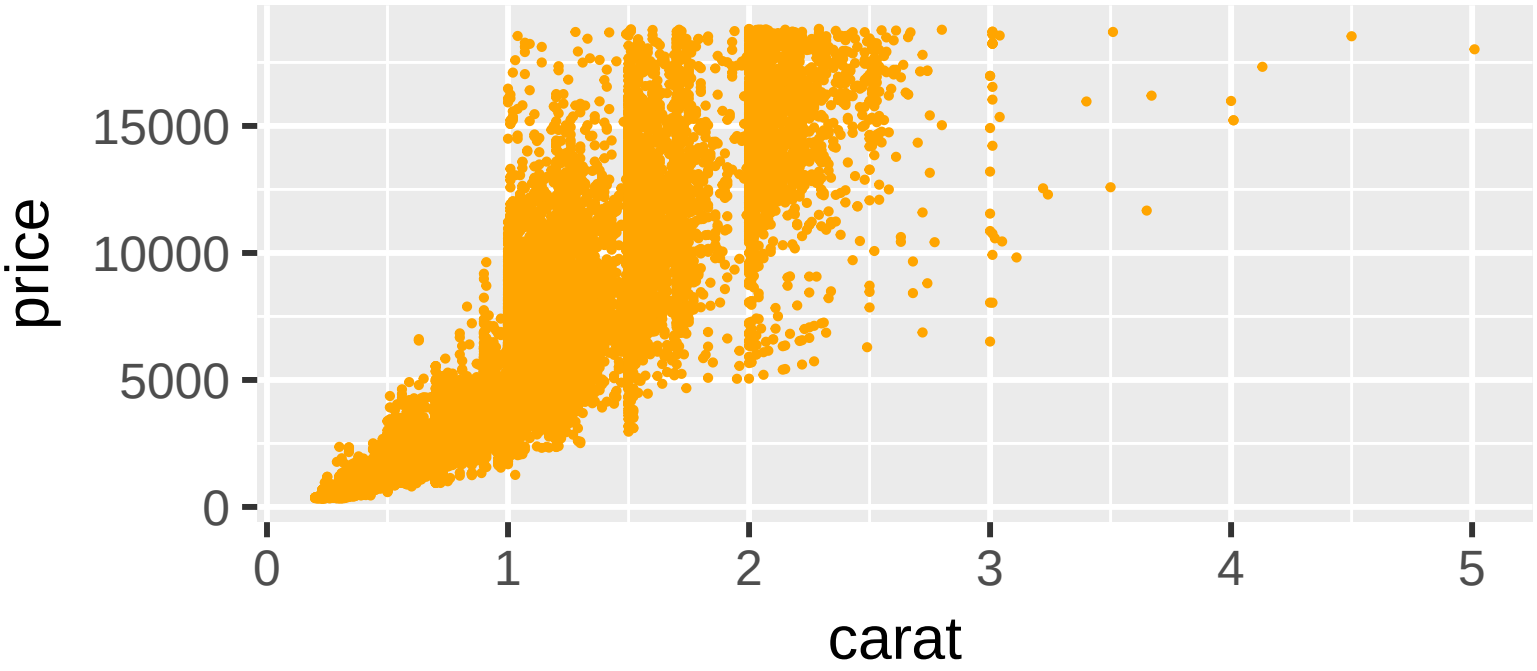
- Input dataset, should always be a data.frame
- X and Y axis - mapping, grouping, coloring
- Layer: the geometric object (plot type)
- Layer: statistical representation of the data
- Position adjustment: dodge, jitter, stack
- Annotataions: addons, lines, borders
- Scales: axis, limits, colors
- Themes: existing themes

The layer geom and stat is always exchangable in different situations based on the emphasis

Gramma

All ggplot2 plots with a call to ggplot(), supplying default data and aesthethic mappings, specified by aes(). You then add layers, scales, coords and facets with +. To save a plot to disk, use ggsave().

```
library(ggplot2)
ggplot(data=diamonds, aes(x=carat, y=price)) +
    geom_point(color='orange', size= 0.2)
## ggsave('Myggplot.pdf')
## ggsave("Myggplot.png")
```



Aesthetic mapping

Aesthetic mappings is the central part of the plot, it describes what variable in the data to be represented and should the plot elements be grouped by variables.

```
##city miles per gallon and highway miles per gallon
ggplot(data=mpg, aes(x=cty, hwy))
p = ggplot(data=mpg, aes(x=cty, hwy))
summary(p)

p + geom_point()
```

Aesthetic mappings can be set in ggplot() and in every geom() layers

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

Aesthetic mapping

aes() is also used to set color and size by variables in dataset

```
ggplot(data=mpg, aes(x=cty, hwy))+
     geom_point(aes(color=cty))

ggplot(data=mpg, aes(x=cty, hwy))+
     geom_point(aes(color=factor(cyl), size=cyl))+
    ggtitle('title of my first graph')
```

You can also map aesthetics to functions of variables

```
ggplot(data=mpg, aes( x = cty^ 2, y = hwy / cyl))+
    geom_point()
```

Aesthetic mapping

There can be only one variable in aes(), with suitable geom method

```
ggplot(mtcars, aes(mpg))
ggplot(mtcars, aes(mpg))+geom_histogram(binwidth=5)

ggplot(mtcars, aes(mpg))+geom_point()
```

geom lables will override color sceme from ggplot call

```
ggplot(mtcars, aes(x=wt, y=mpg, color=cyl))+
    geom_point(size=5, color='green')
ggplot(mtcars, aes(x=wt, y=mpg, color=factor(cyl)))+
    geom_point(size=5)
```

Layers

We can set another layer of statistical representation, variable names should always indide aes()

```
ggplot(mpg, aes(x=cty, y=hwy))+
   geom_point(aes(color=factor(year), size=displ))+
   stat_smooth()
```

```
ggplot(mpg, aes(x=cty, y=hwy))+
   geom_point(aes(color=factor(year), size=displ),
              alpha=0.5, position= 'jitter')+
   stat_smooth()+
   scale_color_manual(values=c('gold','lightblue'))
```

Aesthetic mapping will not be shared between added layers

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

Exercise

- Box plot

```
ggplot(mtcars, aes(x=factor(cyl), y=mpg, fill=cyl))+
    geom_boxplot()
ggplot(mtcars, aes(x=factor(cyl), y=mpg, fill=factor(cyl)))+
    geom_boxplot()
```

- Violin plot

```
ggplot(mtcars, aes(x=factor(cyl), y=mpg, fill=factor(cyl)))+
    geom_violin()
```

- Jitter plot

```
###jitter plot
ggplot(diamonds, aes(x=clarity, y=price,color=clarity))+
    geom_jitter()
head(diamonds)
```

Exercise

- Line chart

```
scale_fill_manual(values=rep(brewer.pal(8, 'Pastel1')[c(1,2,5)], 5))
set.seed(100)
rainfall=data.frame(matrix(rnorm(48), 8, 6))
rainfall=rainfall+5
names(rainfall)=c('Jan','Feb','Mar','Apr','May','Jun')
rainfall[1:4,]=rainfall[1:4,] + rep(seq(0,9,length.out = 8), each=6)
rainfall$city=c('Beijing','Bangkok','Delhi',
                'Moscow','Suzhou','Lima', 'Berlin','Madrid')
rainfall=melt(rainfall)
ggplot(rainfall, aes(variable, value, color=city))+
    geom_line(aes(group=city), size=2 )+
    geom_point(size=3)+
     theme(panel.grid.minor=element_blank(),
           panel.grid.major=element_blank(),
           panel.background=element_blank(),
           panel.border=element_blank())+
    scale_fill_manual(values=brewer.pal(8, 'set2'))
```