Visualizing and describing data

Chaochen Wang

ZJU-UoE Institute

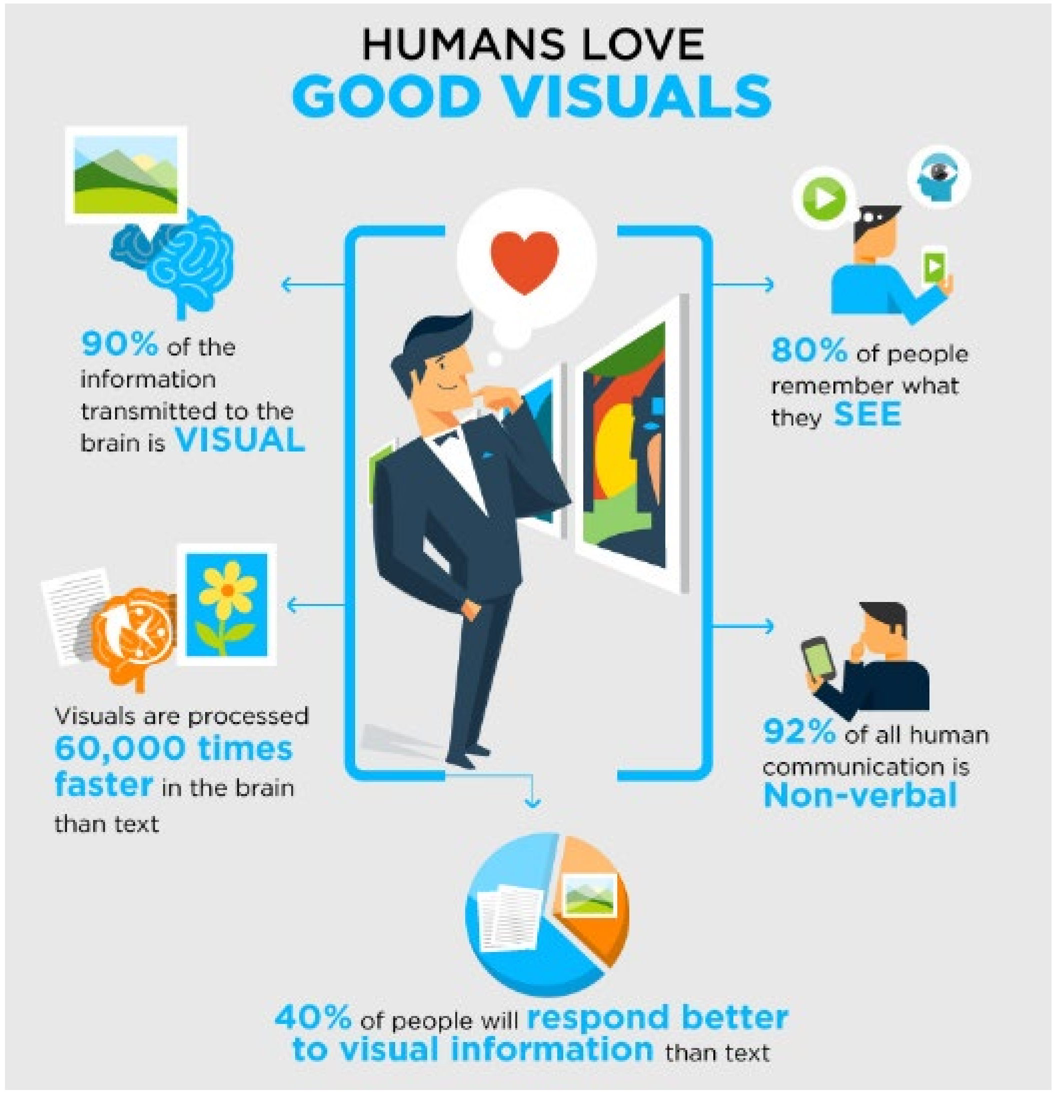
wangchaochen@intl.zju.edu.cn

Oct 21, 2019

Learning Objects-Week 6

- ❖ Use ggplot2 for data visualization
- ❖ Think critically about data visualization choices

Purposes of data visualization


HUMANS LOVE
GOOD VISUALS
90% of the information transmitted to the brain is VISUAL
80% of people remember what they SEE
Visuals are processed 60,000 times faster in the brain than text
92% of all human communication is Non-verbal
40% of people will respond better to visual information than text

https://www.infographicdesignteam.com/blog/data-visualization-best-practices/

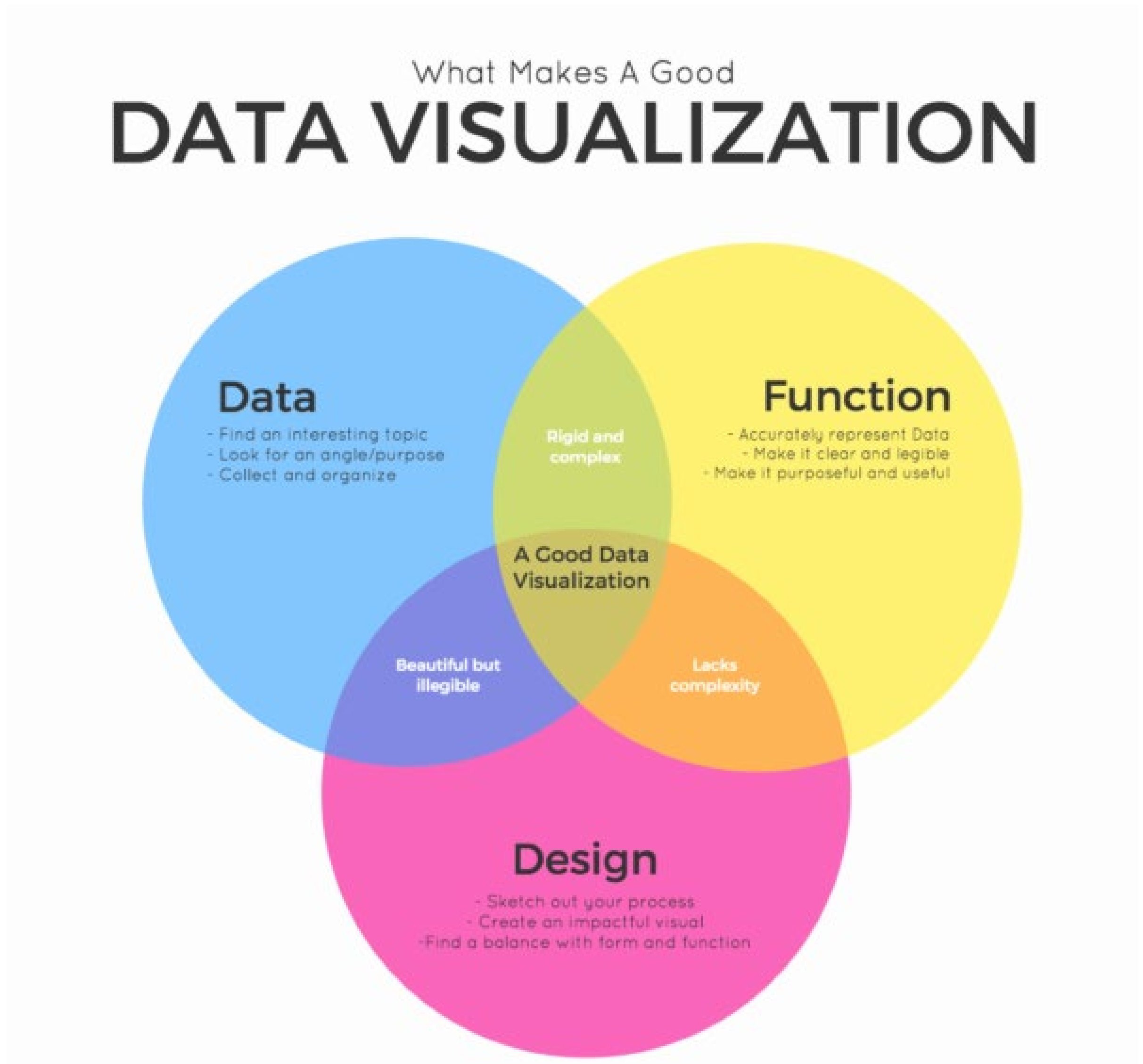
1. Present data
-Straightforward
-Present large data sets in a limited space

2. Provide more information
-Counts, Distribution, Trends, Irregularities…

3. Tell a story
-Relationships among data
-Help find interesting regions
-Help make decisions

Good data visualization

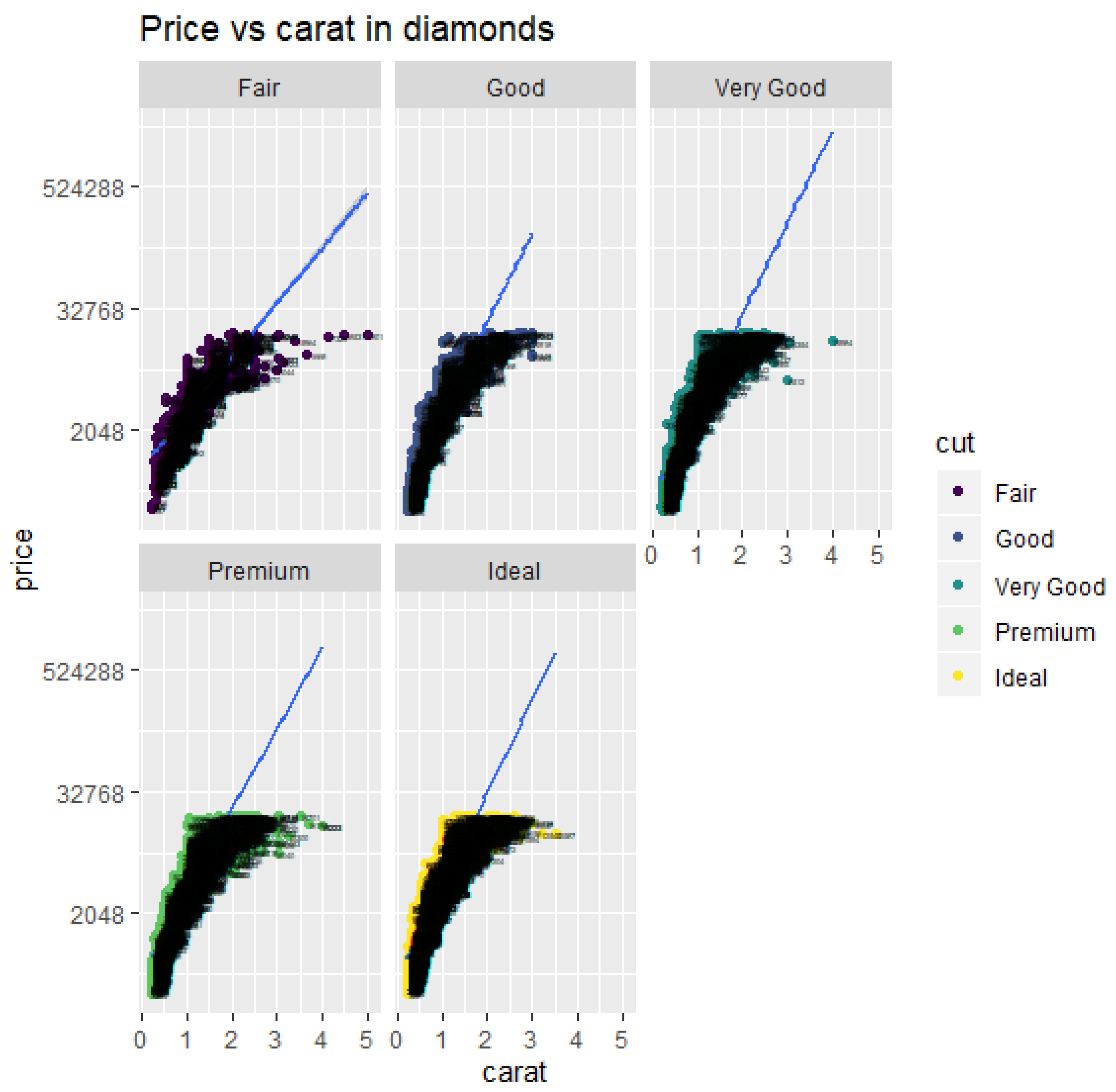
- ❖ Numbers & Stats
- ❖ Timelines
- ❖ Processes
- ❖ Comparisons
- ❖ Lists


What Makes A Good
DATA VISUALIZATION
Data
- Find an interesting topic
- Look for an angle/purpose
- Collect and organize
Function
- Accurately represent Data
- Make it clear and legible
- Make it purposeful and useful
Rigid and complex
A Good Data Visualization
Beautiful but illegible
Lacks complexity
Design
- Sketch out your process
- Create an impactful visual
-Find a balance with form and function

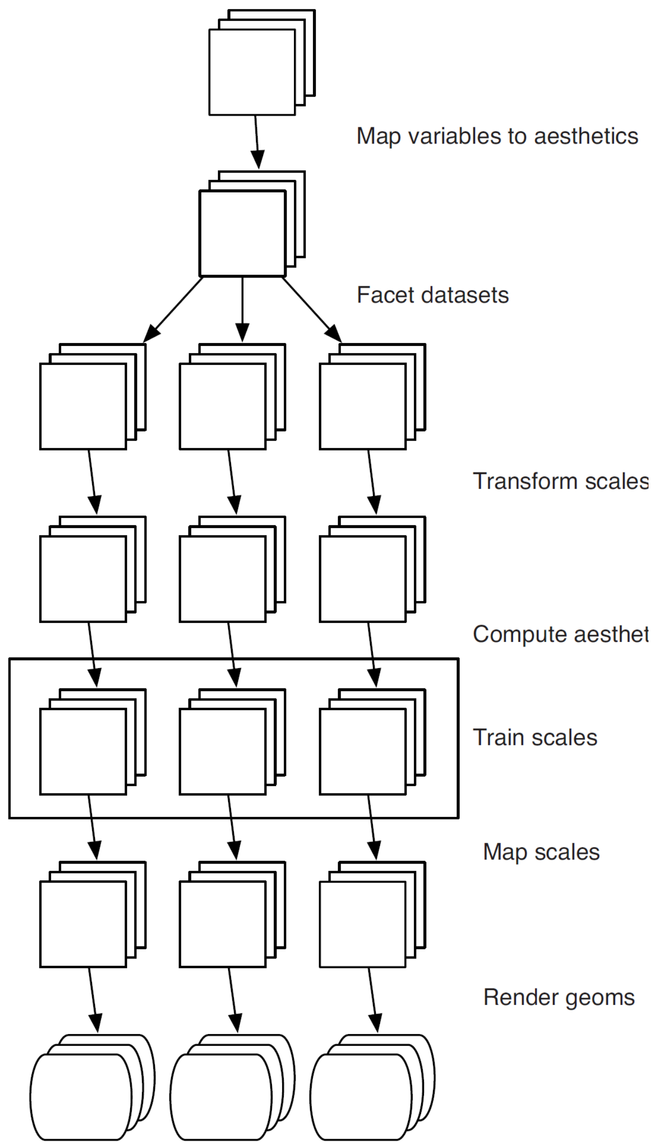
https://hiilite.com/information-visualization/

ggplot2

```
data(“diamonds”)

ggplot(data=diamonds, aes(x=carat, y=price, group= cut))
+ geom_point(stat = "identity", aes(colour = cut))
+ geom_smooth(aes(group=cut), method = "lm")
+ geom_text(aes(label = price),hjust = 0.1, nudge_x = 0.05 ,size=1)
+ scale_y_continuous(trans='log2’)
+ labs(title="Price vs carat in diamonds")
+ facet_wrap(.~cut)
```



Ggplot2 – layered grammar



From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Data(Must be data frame)

↓

Aesthetic mappings

↓

Geometric objects & statistical transformations

↓

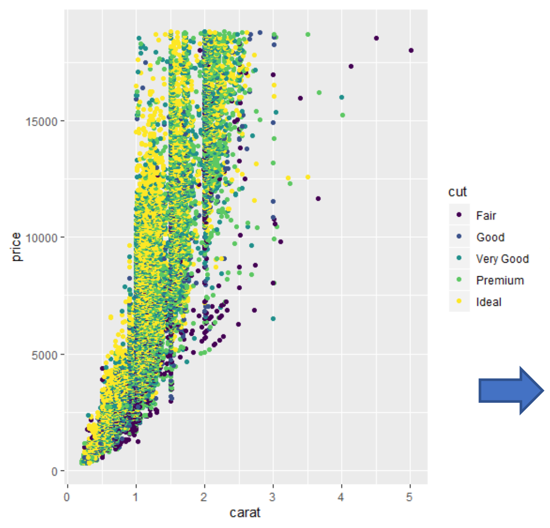
Scales

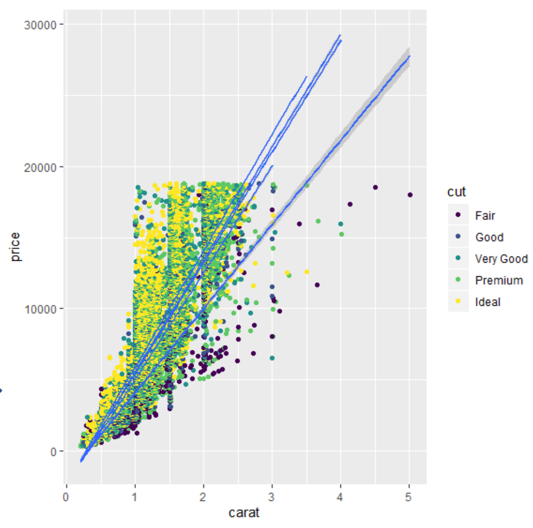
↓

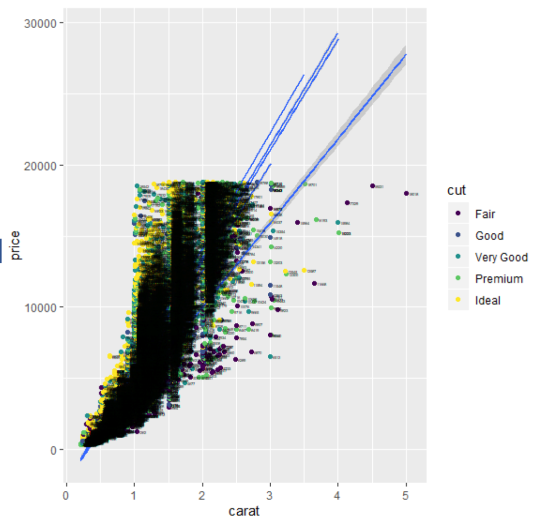
Faceting

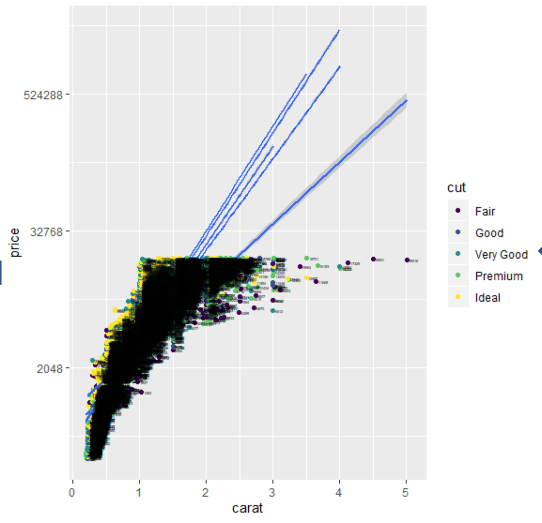
Build layer by layer

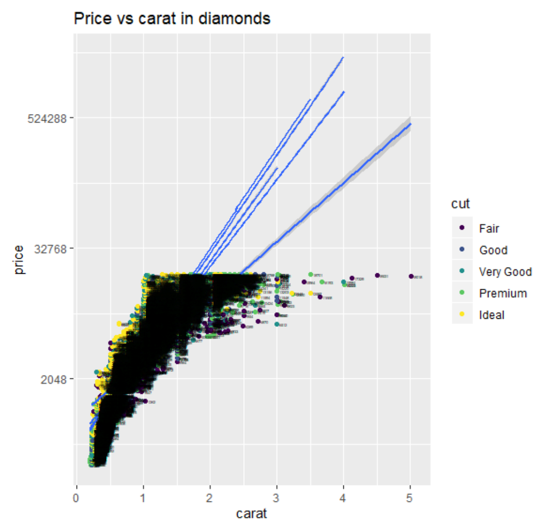
```
g <- ggplot(data=diamonds,aes(x=carat, y=price, group=cut))
g1 <- g + geom_point(stat = "identity", aes(colour = cut))
g2 <- g1+ geom_smooth(aes(group=cut), method = "lm")
g3 <- g2 + geom_text(aes(label = price),hjust = 0.1, nudge_x = 0.05 ,size=1)
g4 <- g3 + scale_y_continuous(trans='log2’)
g5 <- g4 + labs(title="Price vs carat in diamonds")
g6 <- g5 +facet_wrap(.~cut)
```

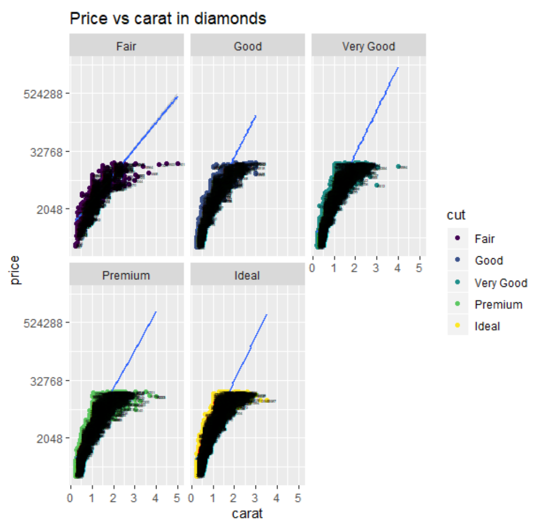












Layer components

Layers are responsible for creating the objects that we perceive on the plot.
A layer is composed of four parts:

- ❖ data and aesthetic mapping,
- ❖ a statistical transformation (stat),
- ❖ a geometric object (geom)
- ❖ and a position adjustment

```
data(“diamonds”)

ggplot(data=diamonds, aes(x=carat, y=price, group= cut))
+ geom_point(stat = "identity", aes(colour = cut), position = “identity” )
```

Multiple layers

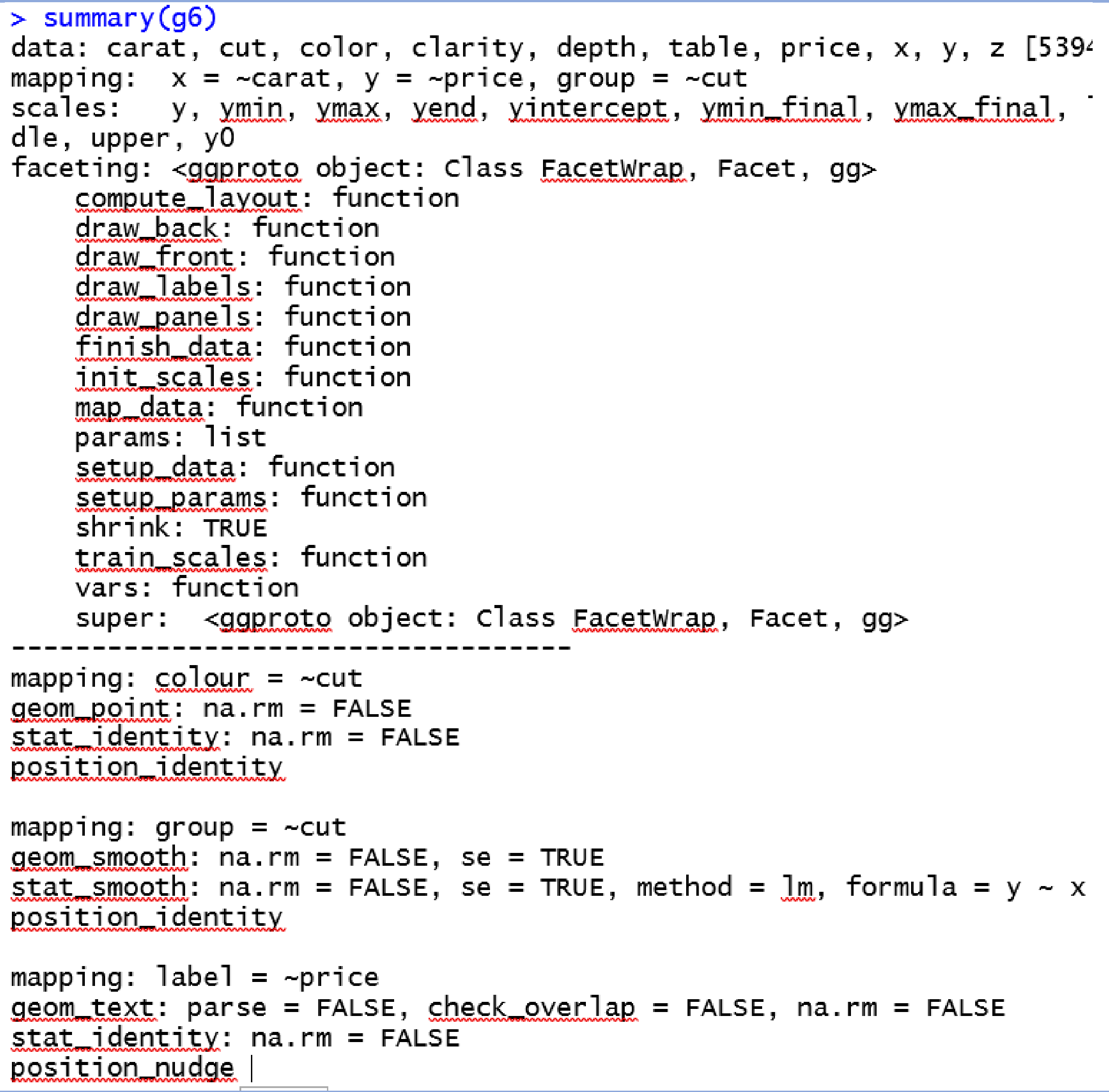
- A default dataset and set of mappings from variables to aesthetics.
- One or more layers, each composed of a geometric object, a statistical transformation, and a position adjustment, and optionally, a dataset and aesthetic mappings.
- One scale for each aesthetic mapping.
- A coordinate system.
- The faceting specification

```
data(“diamonds”)

ggplot(data=diamonds, aes(x=carat, y=price, group= cut))
+ geom_point(stat = "identity", aes(colour = cut))
+ geom_smooth(aes(group=cut), method = "lm")
+ geom_text(aes(label = price), hjust = 0.1, nudge_x = 0.05 ,size=1)
+ scale_y_continuous(trans='log2’)
+ labs(title="Price vs carat in diamonds")
+ facet_wrap(.~cut)
```

Ggplot is a R object

- Can be viewed by *summary*
- Can be saved (*save*) and loaded (*load*)
- Data is stored inside the plot, so that if you change the data outside of the plot, and then redraw a saved plot, it will not be updated.
- Geom can also be saved and apply it to another ggplot object if the aesthetics still exist

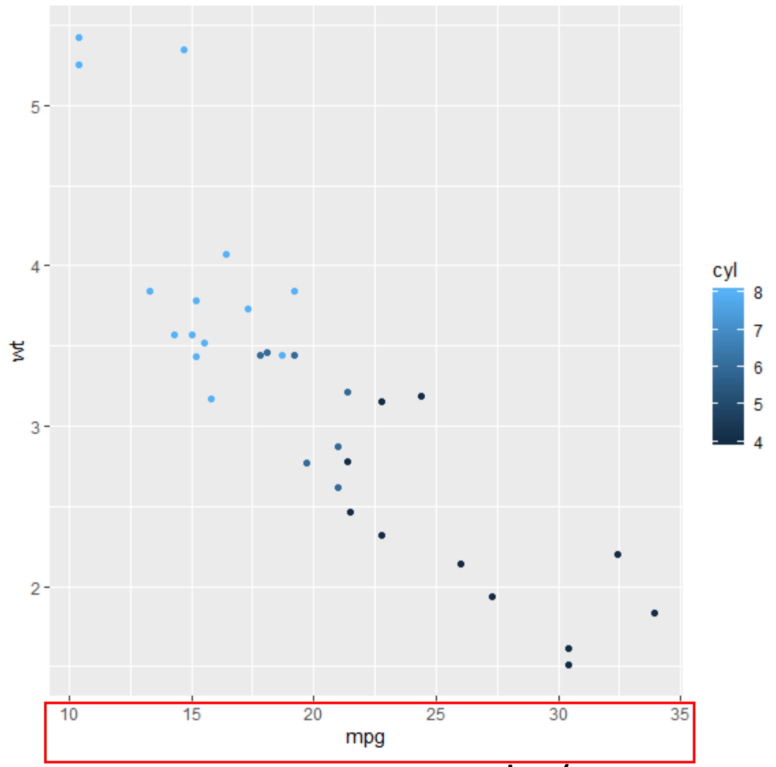
```
r <- ggplot(data=rubies, aes(x=carat, y=price, group=cut))
s <- ggplot(data= sapphires, aes(x=carat, y=price, group=cut))
p <-  geom_point(stat = "identity", aes(colour = cut))
+ geom_smooth(aes(group=cut), method = "lm")
r + p
s + p
```

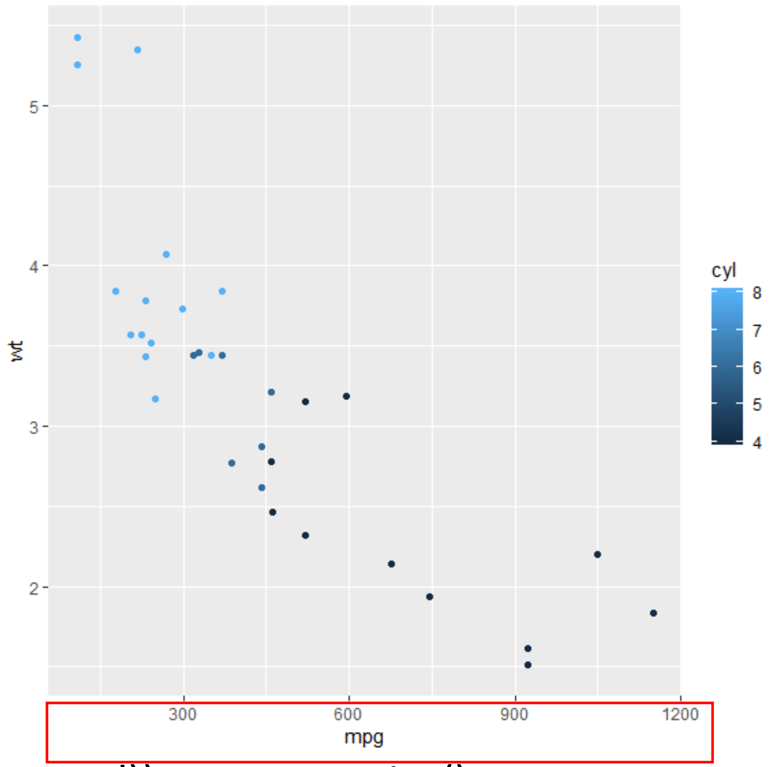


```
> summary(g6)
data: carat, cut, color, clarity, depth, table, price, x, y, z [539
mapping:  x = ~carat, y = ~price, group = ~cut
scales:   y, ymin, ymax, yend, yintercept, ymin_final, ymax_final,
dle, upper, y0
faceting: <ggproto object: Class FacetWrap, Facet, gg>
    compute_layout: function
    draw_back: function
    draw_front: function
    draw_labels: function
    draw_panels: function
    finish_data: function
    init_scales: function
    map_data: function
    params: list
    setup_data: function
    setup_params: function
    shrink: TRUE
    train_scales: function
    vars: function
    super:  <ggproto object: Class FacetWrap, Facet, gg>
-----------------------------------
mapping: colour = ~cut
geom_point: na.rm = FALSE
stat_identity: na.rm = FALSE
position_identity

mapping: group = ~cut
geom_smooth: na.rm = FALSE, se = TRUE
stat_smooth: na.rm = FALSE, se = TRUE, method = lm, formula = y ~ x
position_identity

mapping: label = ~price
geom_text: parse = FALSE, check_overlap = FALSE, na.rm = FALSE
stat_identity: na.rm = FALSE
position_nudge
```

Dataset

- Must be a data frame
- Old dataset can be replaced with %+%





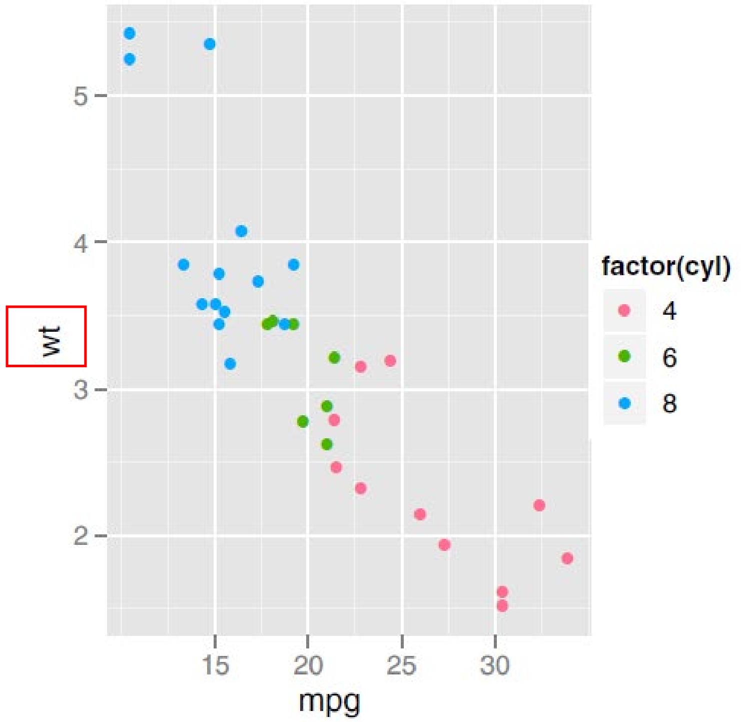
```
p <- ggplot(mtcars, aes(mpg, wt, colour = cyl)) + geom_point()
P
mtcars <- transform(mtcars, mpg = mpg ^ 2)
p %+% mtcars
```

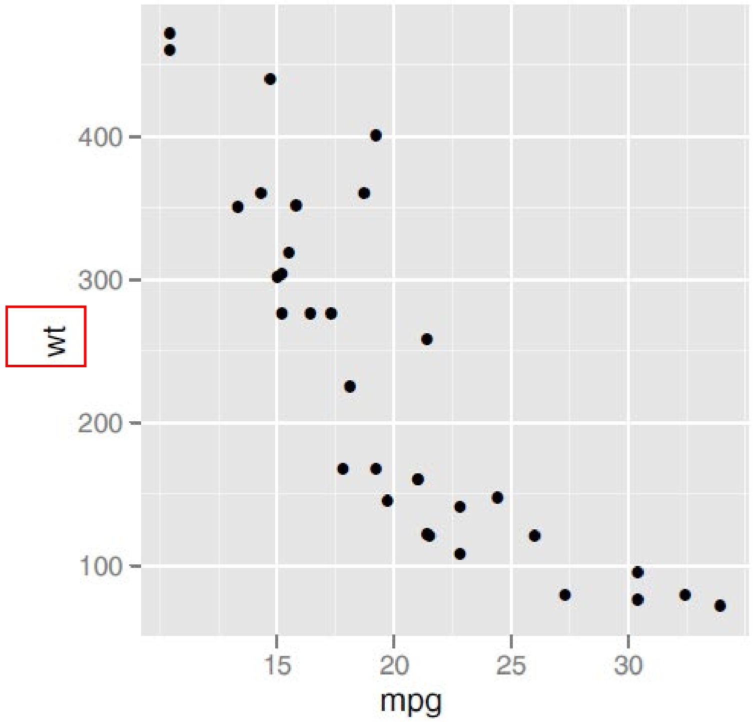
From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Aesthetics mapping

- Any variable in an aes() specification must be contained inside the plot or layer data.
- Mapping can be extended or overridden in the layers.
- Aesthetic mappings specified in a layer affect only that layer.

```
p <- ggplot(mtcars, aes(x = mpg, y = wt))
p + geom_point(aes(colour = factor(cyl)))
p + geom_point(aes(y = disp))
```





From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Setting vs. mapping

map an aesthetic to a variable (e.g., (aes(colour = cut))) or **set** it to a constant (e.g.,colour = "red"), they are different!!!

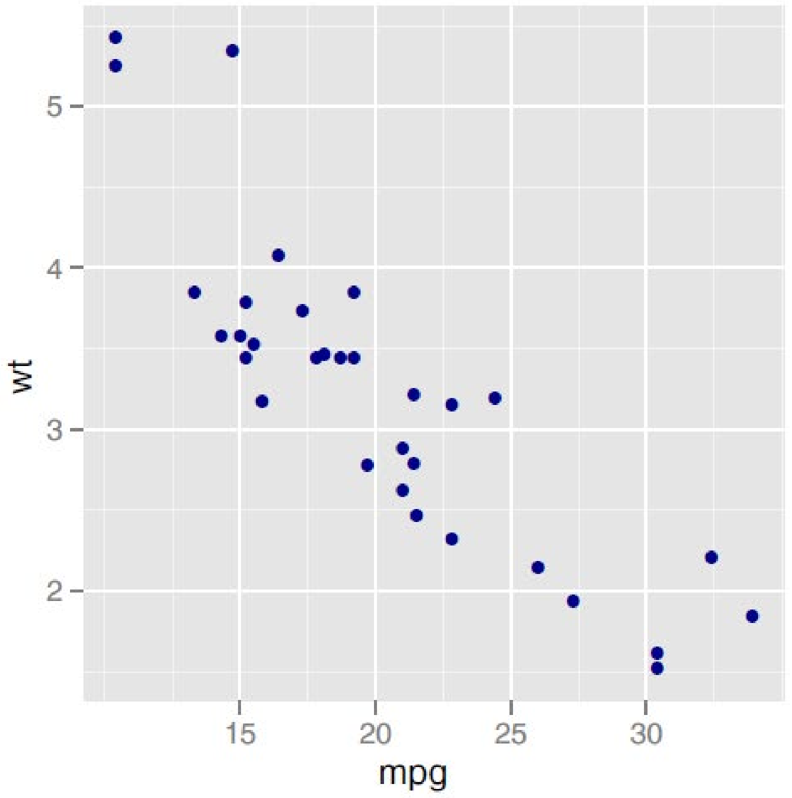
p <- ggplot(mtcars, aes(mpg, wt))

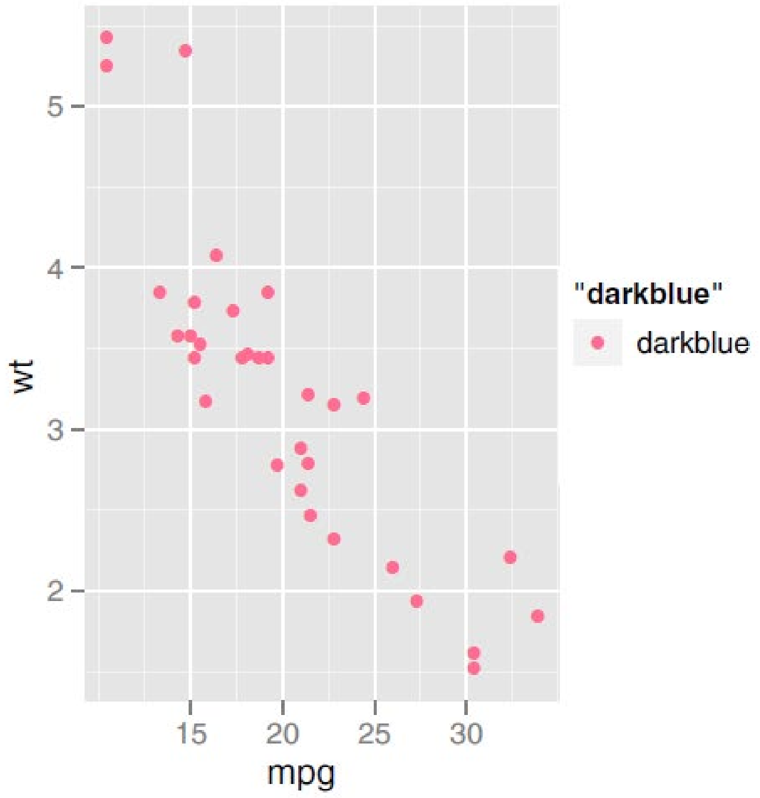
p + geom_point(colour = "darkblue")

It is a parameter of colour

p + geom_point(aes(colour = "darkblue"))

It creates a new variable





From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

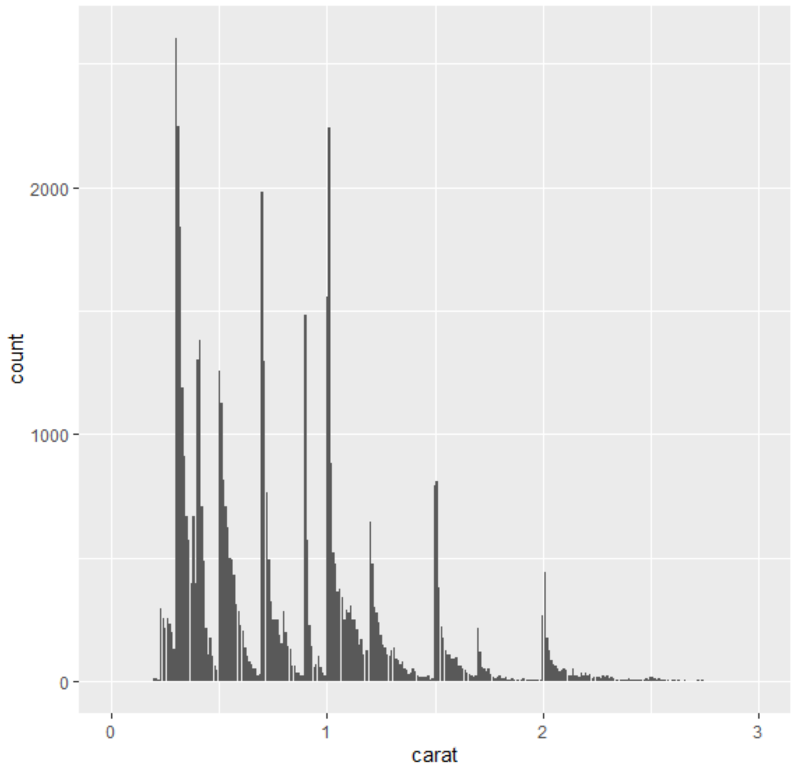
Stat

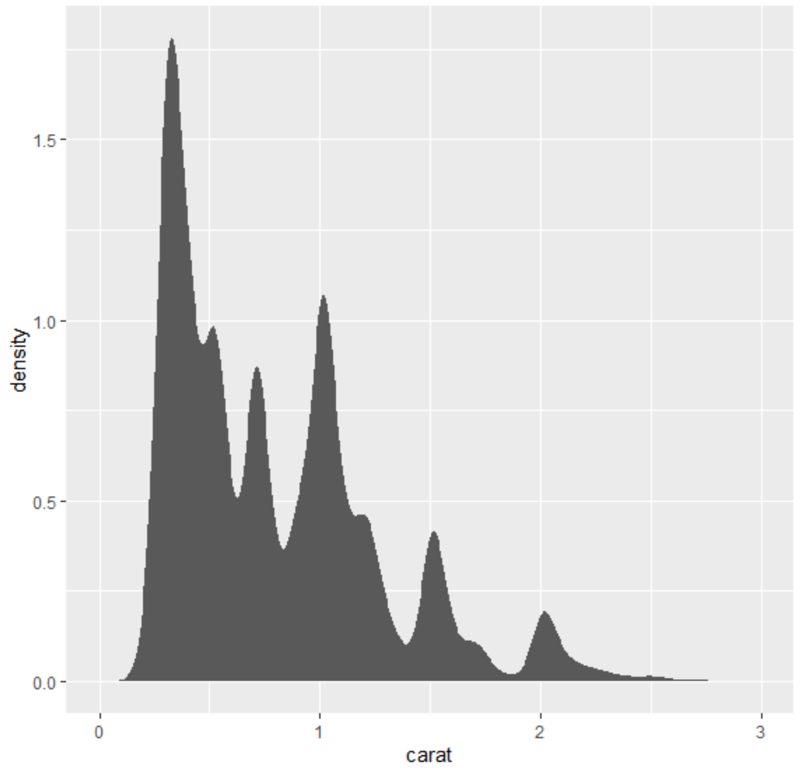
Name	Description
bin	Bin data
boxplot	Calculate components of box-and-whisker plot
contour	Contours of 3d data
density	Density estimation, 1d
density_2d	Density estimation, 2d
function	Superimpose a function
identity	Don't transform data
qq	Calculation for quantile-quantile plot
quantile	Continuous quantiles
smooth	Add a smoother
spoke	Convert angle and radius to xend and yend
step	Create stair steps
sum	Sum unique values. Useful for overplotting on scatter-plots
summary	Summarise y values at every unique x
unique	Remove duplicates

From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

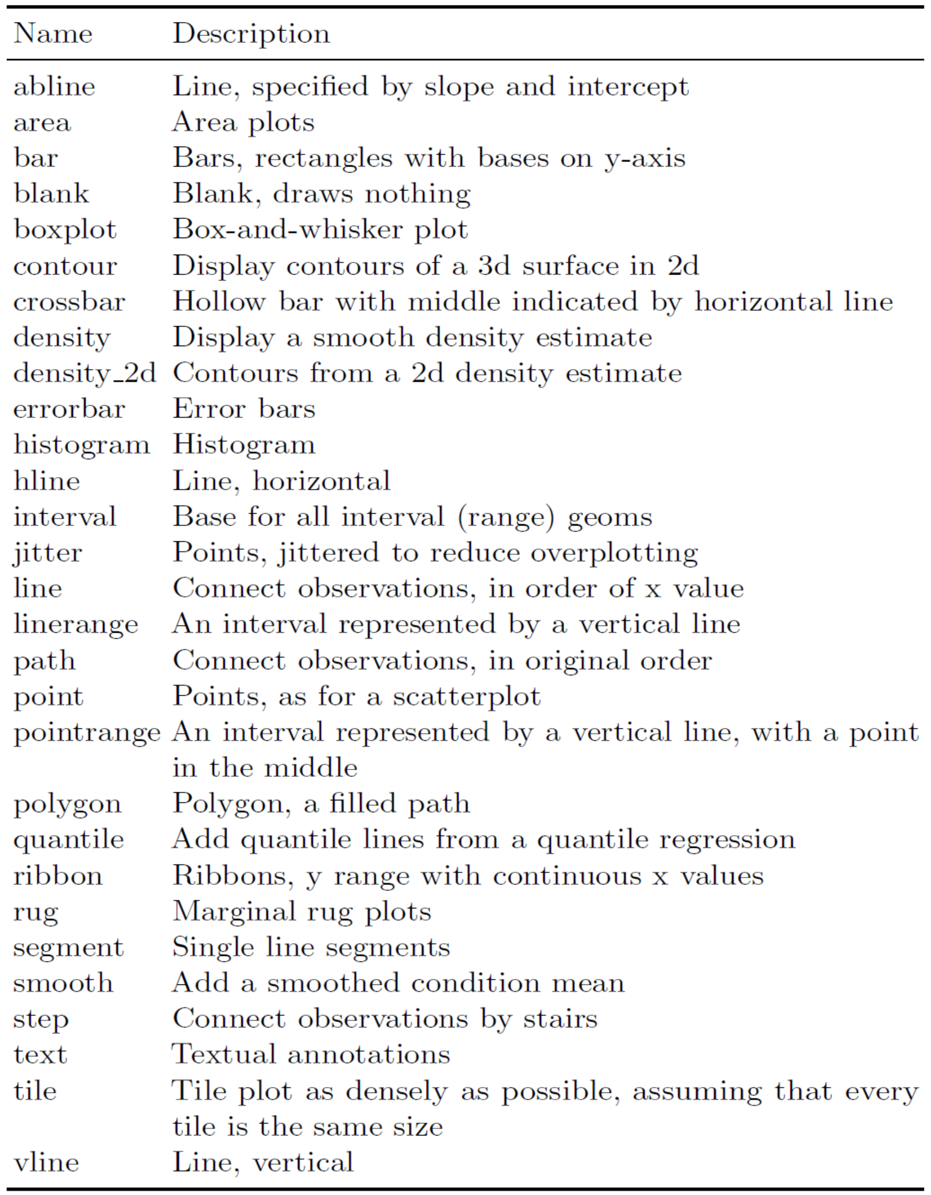
Choose different stat

```
d <- ggplot(diamonds, aes(carat)) + xlim(0, 3)
d+ geom_histogram(stat = "count")
d+ geom_histogram(stat="density")
```





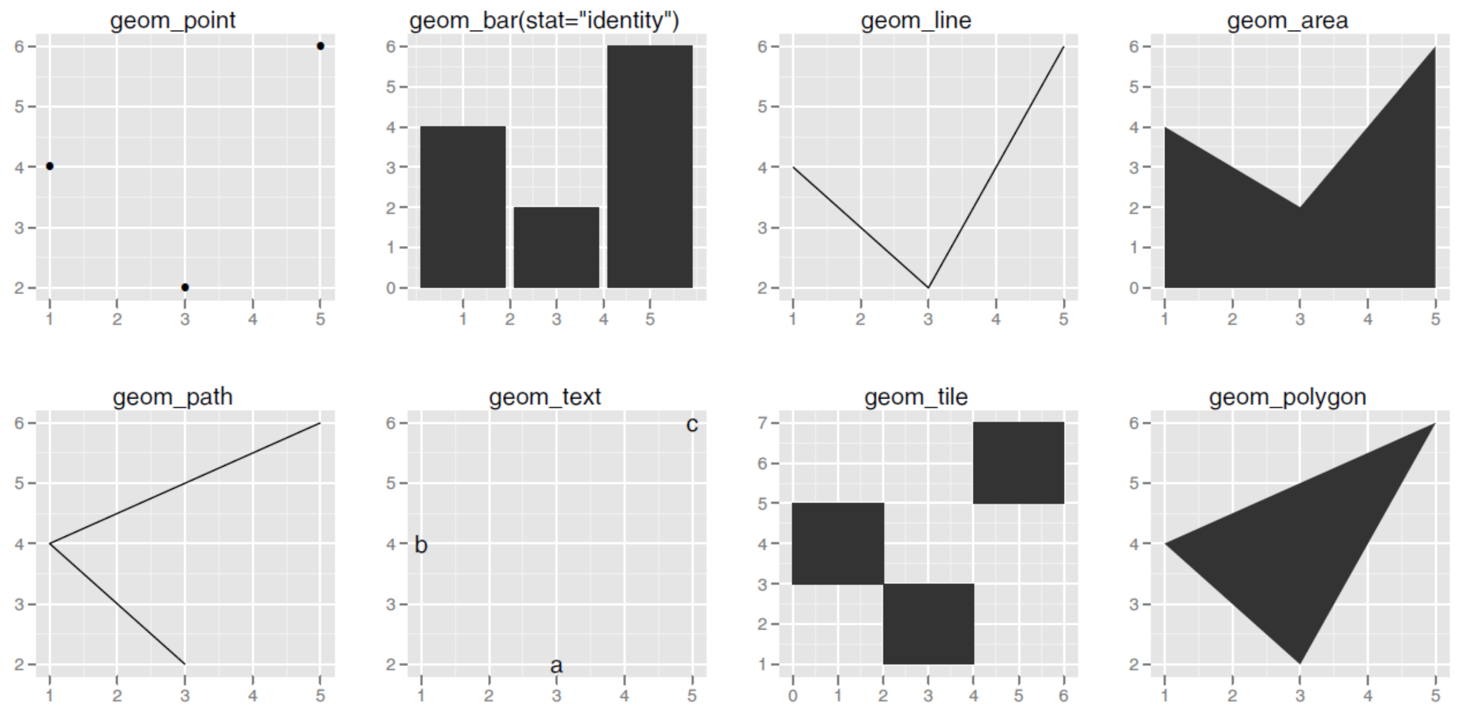
geom



Name	Description
abline	Line, specified by slope and intercept
area	Area plots
bar	Bars, rectangles with bases on y-axis
blank	Blank, draws nothing
boxplot	Box-and-whisker plot
contour	Display contours of a 3d surface in 2d
crossbar	Hollow bar with middle indicated by horizontal line
density	Display a smooth density estimate
density_2d	Contours from a 2d density estimate
errorbar	Error bars
histogram	Histogram
hline	Line, horizontal
interval	Base for all interval (range) geoms
jitter	Points, jittered to reduce overplotting
line	Connect observations, in order of x value
linerange	An interval represented by a vertical line
path	Connect observations, in original order
point	Points, as for a scatterplot
pointrange	An interval represented by a vertical line, with a point in the middle
polygon	Polygon, a filled path
quantile	Add quantile lines from a quantile regression
ribbon	Ribbons, y range with continuous x values
rug	Marginal rug plots
segment	Single line segments
smooth	Add a smoothed condition mean
step	Connect observations by stairs
text	Textual annotations
tile	Tile plot as densely as possible, assuming that every tile is the same size
vline	Line, vertical

From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham
More information in
https://ggplot2.tidyverse.org/reference/

geom

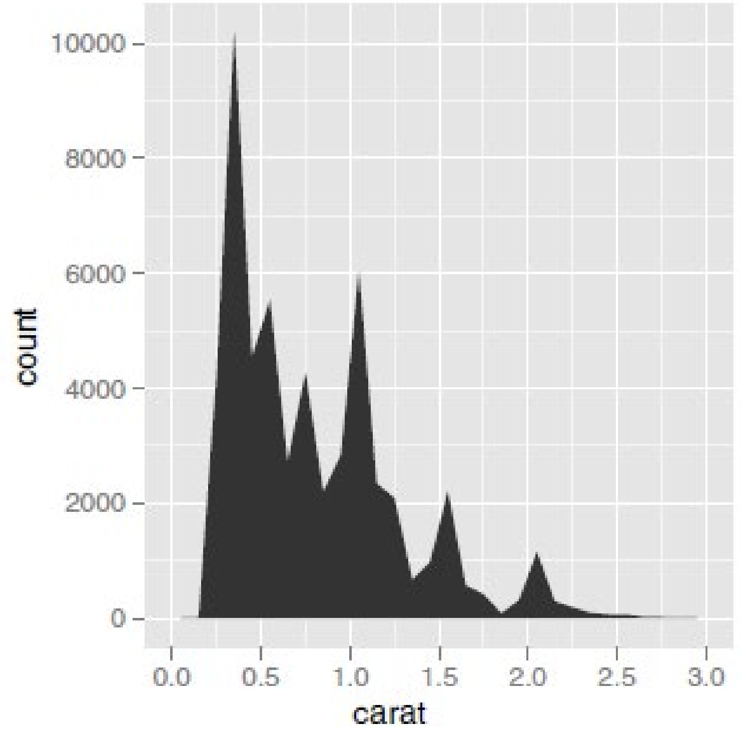


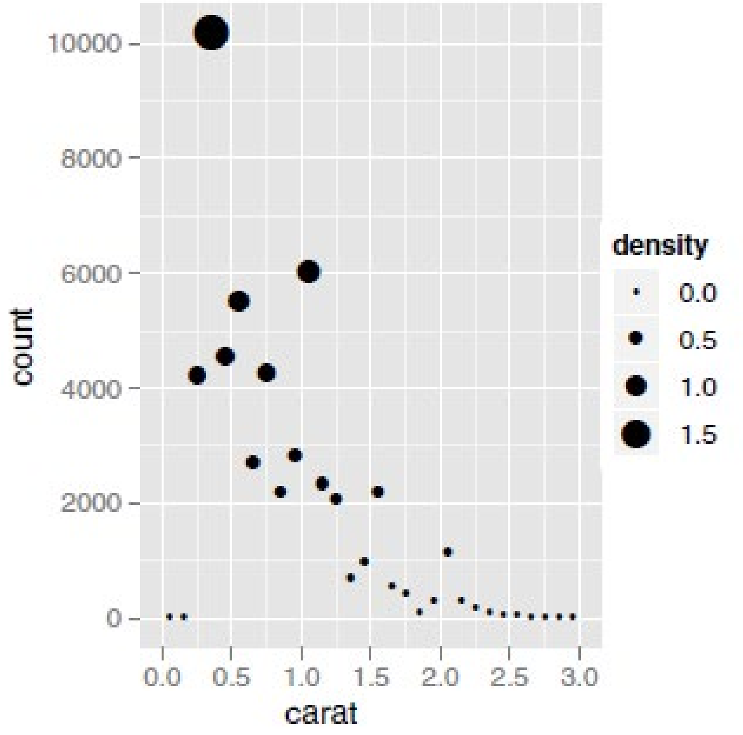
From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Choose different geom

```
d <- ggplot(diamonds, aes(carat)) + xlim(0, 3)
d + stat_bin(aes(ymax = ..count..), binwidth = 0.1, geom = "area")
d + stat_bin(aes(size = ..density..), binwidth = 0.1, geom = "point", position="identity" )
```

Statistical transformation generate new variables, which can be directly used.
To distinguish them from the variable names in the original data, they are surrounded by “..”





From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

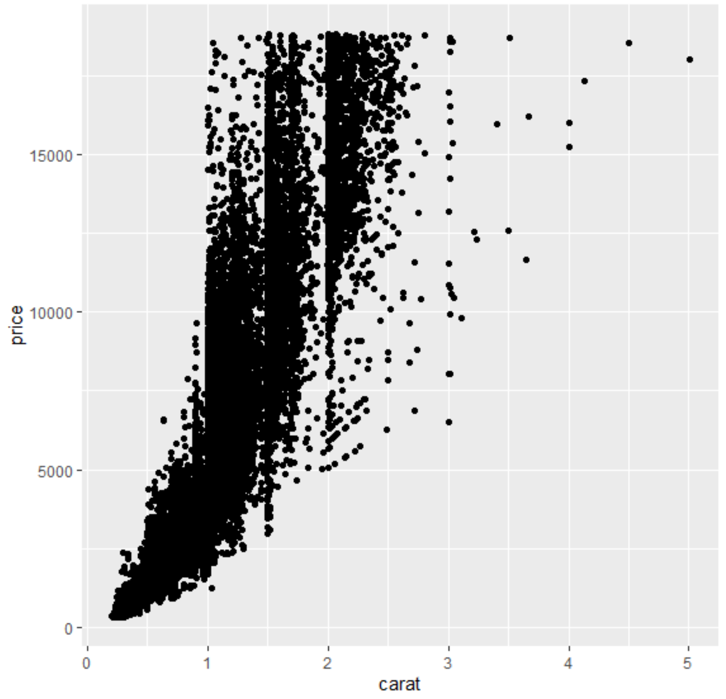
Geom_xxx and stat_xxx are shortcuts for layer

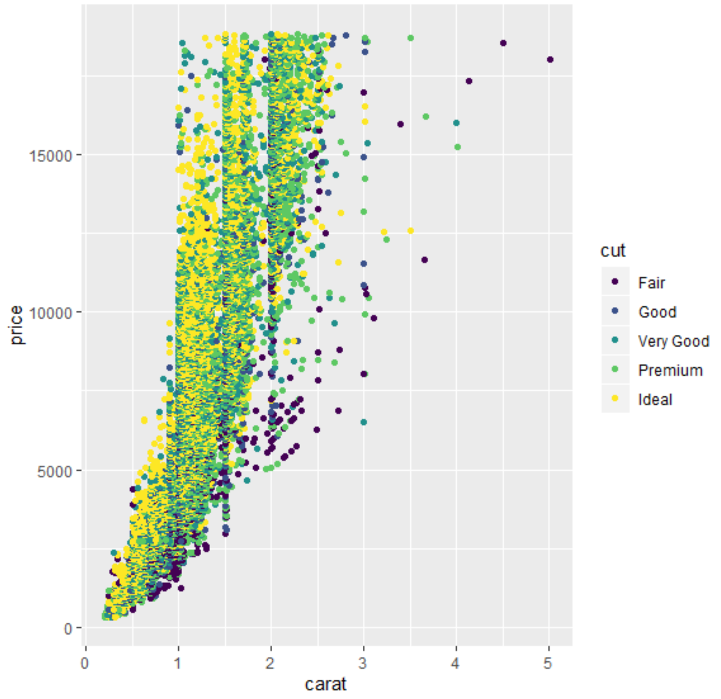
```
p <- ggplot(diamonds, aes(x = carat))

p + layer(geom = "bar", stat = "bin", position = "identity“,params = list(fill = "steelblue", binwidth=0.1))

p + geom_histogram(stat="bin", fill="steelblue", binwidth = 0.1)

p + stat_bin(geom="bar", fill="steelblue", binwidth = 0.1)
```

They are the same!

Grouping

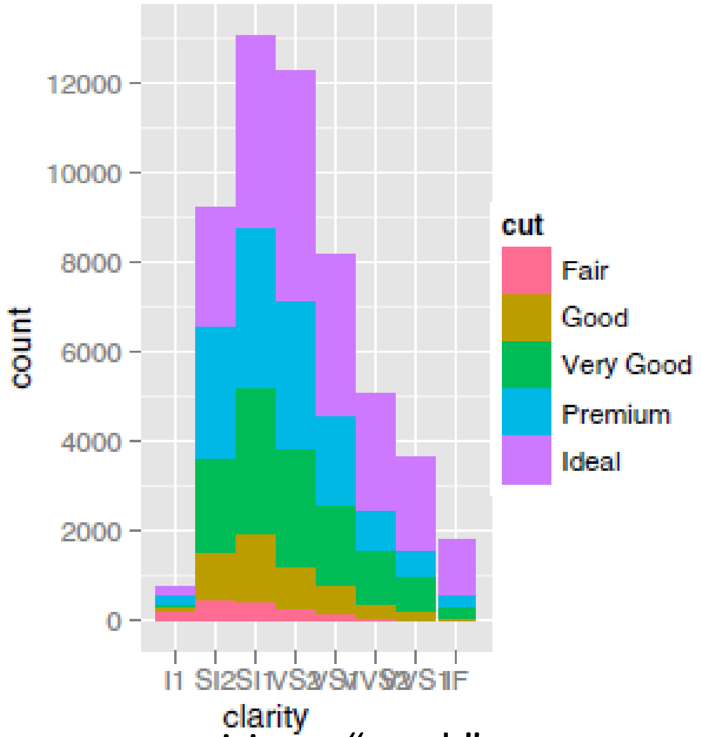
```
g <- ggplot(data=diamonds,aes(x=carat, y=price))
g + geom_point()
g + geom_point(aes(group=cut, colour=cut))
```



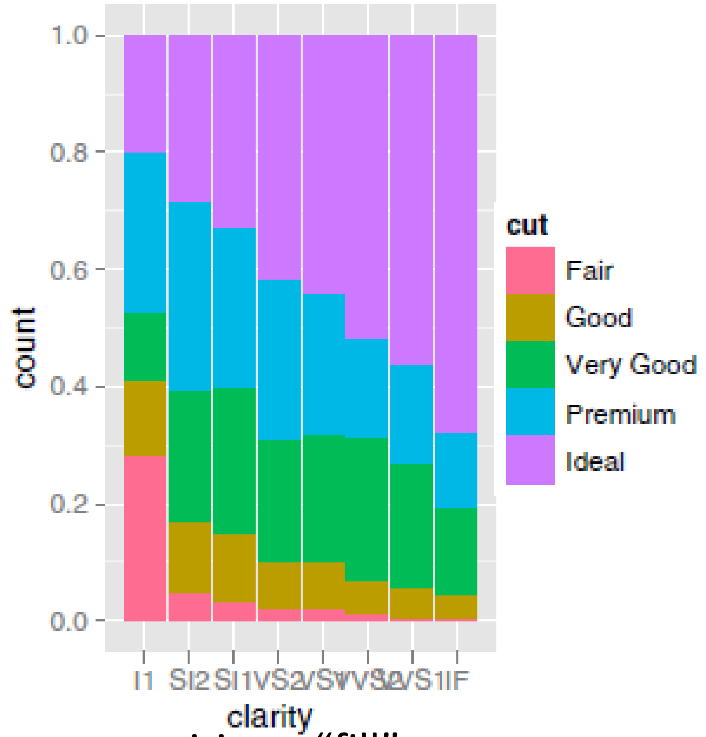


Position adjustment

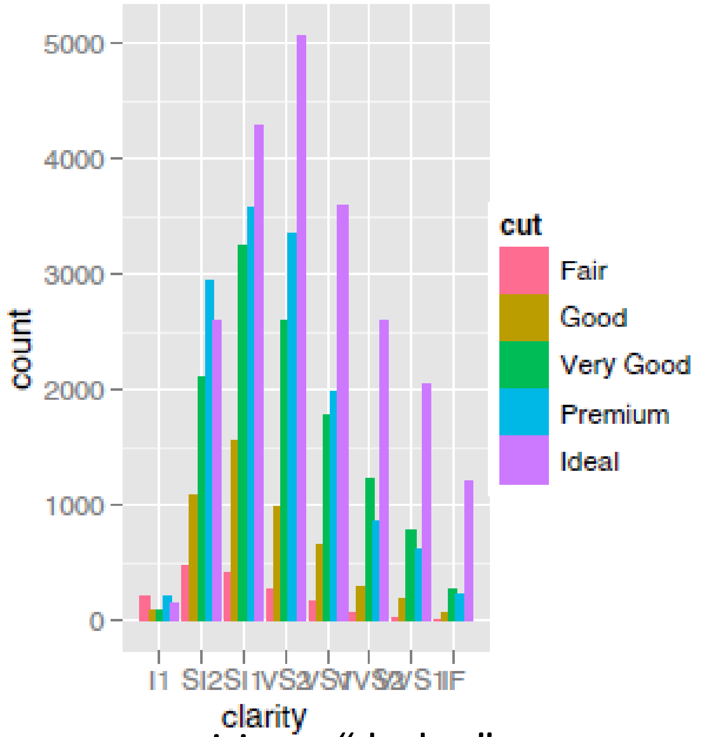
Adjustment	Description
dodge	Adjust position by dodging overlaps to the side
fill	Stack overlapping objects and standardise have equal height
identity	Don't adjust position
jitter	Jitter points to avoid overplotting
stack	Stack overlapping objects on top of one another



position= “stack”



position= “fill”

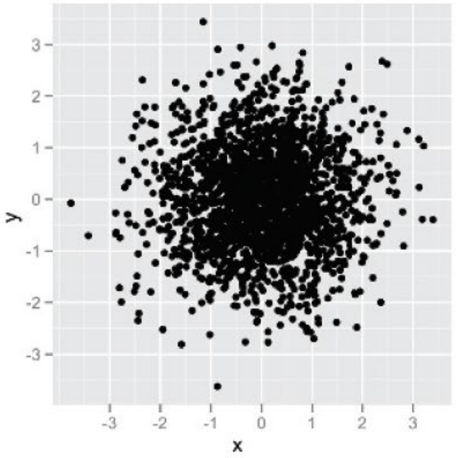


position= “dodge”

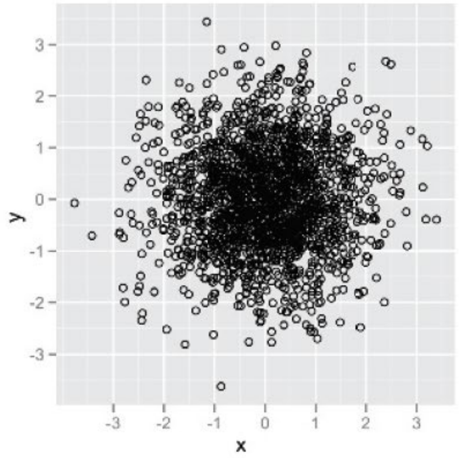
From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Overplotting

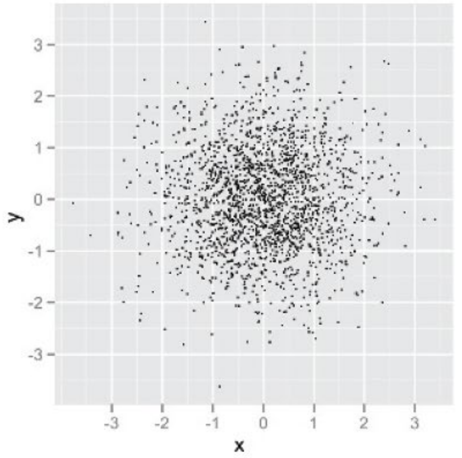
making the points smaller or using hollow glyphs



geom_point()

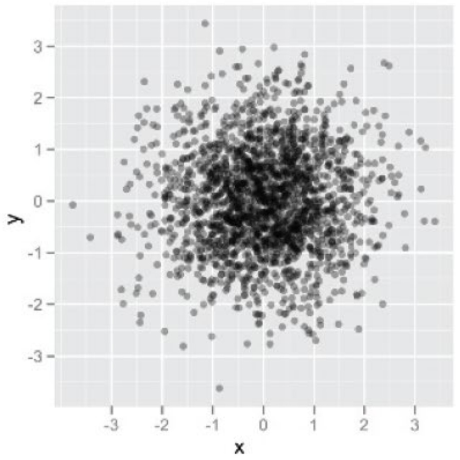


geom_point(shape = 1)

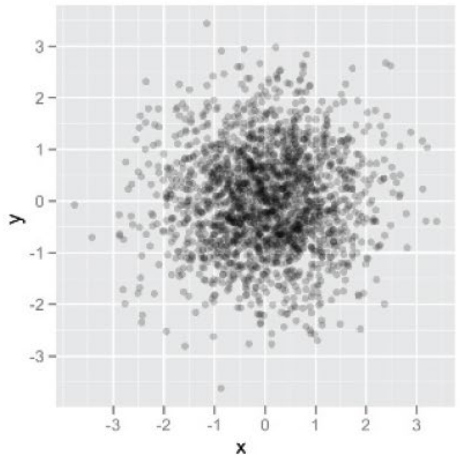


geom_point(shape = ".")

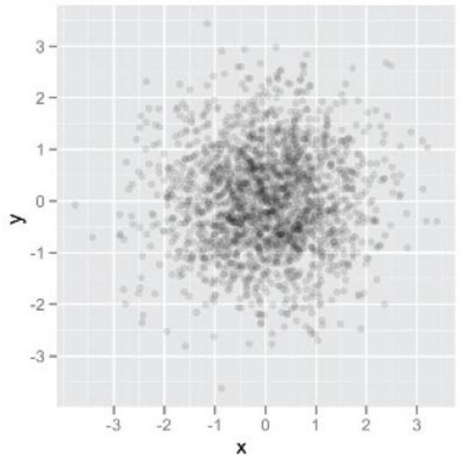
use alpha blending (transparency)



geom_point(colour = alpha("black", 1/3))



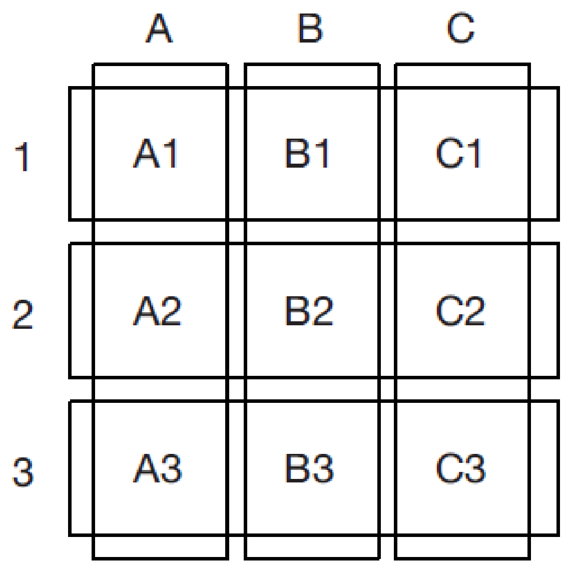
1/5



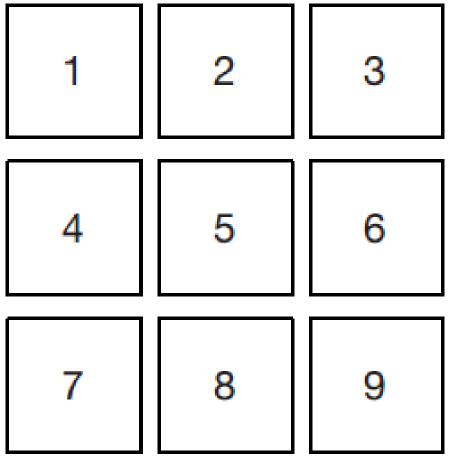
1/10

From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Faceting



facet_grid



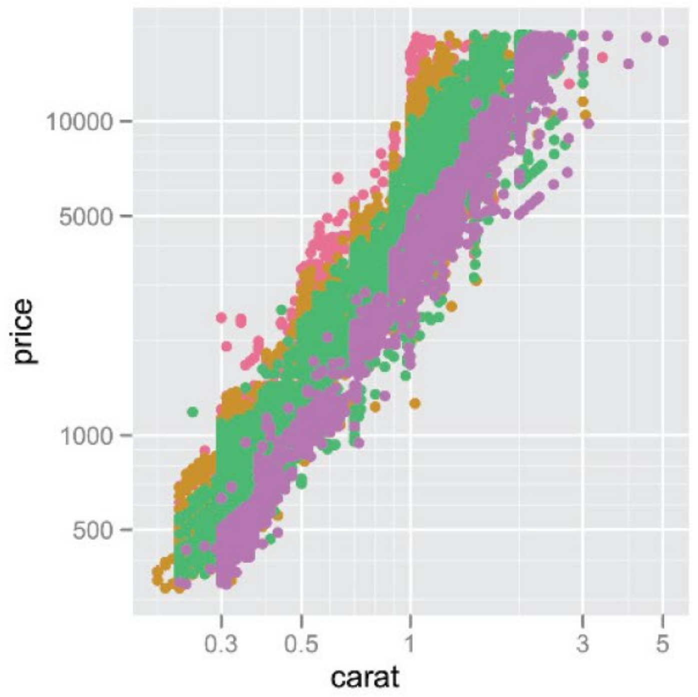
facet_wrap

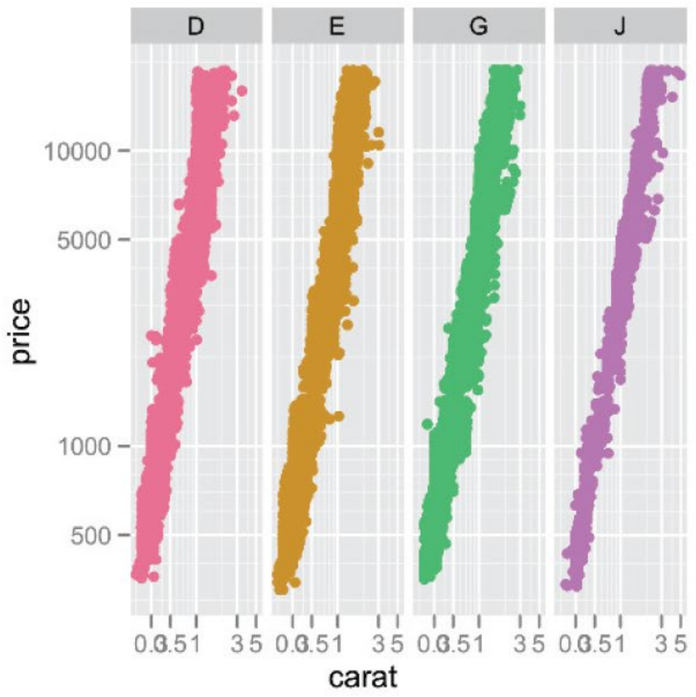
```
p + facet_grid(rows = vars(drv))
p + facet_grid(cols = vars(cyl))
p + facet_grid(vars(drv), vars(cyl))

p + facet_grid(. ~ cyl)
p + facet_grid(drv ~ .)
p + facet_grid(drv ~ cyl)
```

```
p + facet_wrap(vars(drv), nrow=3)
p + facet_wrap(~drv, ncol=3)
```

From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Grouping vs Faceting

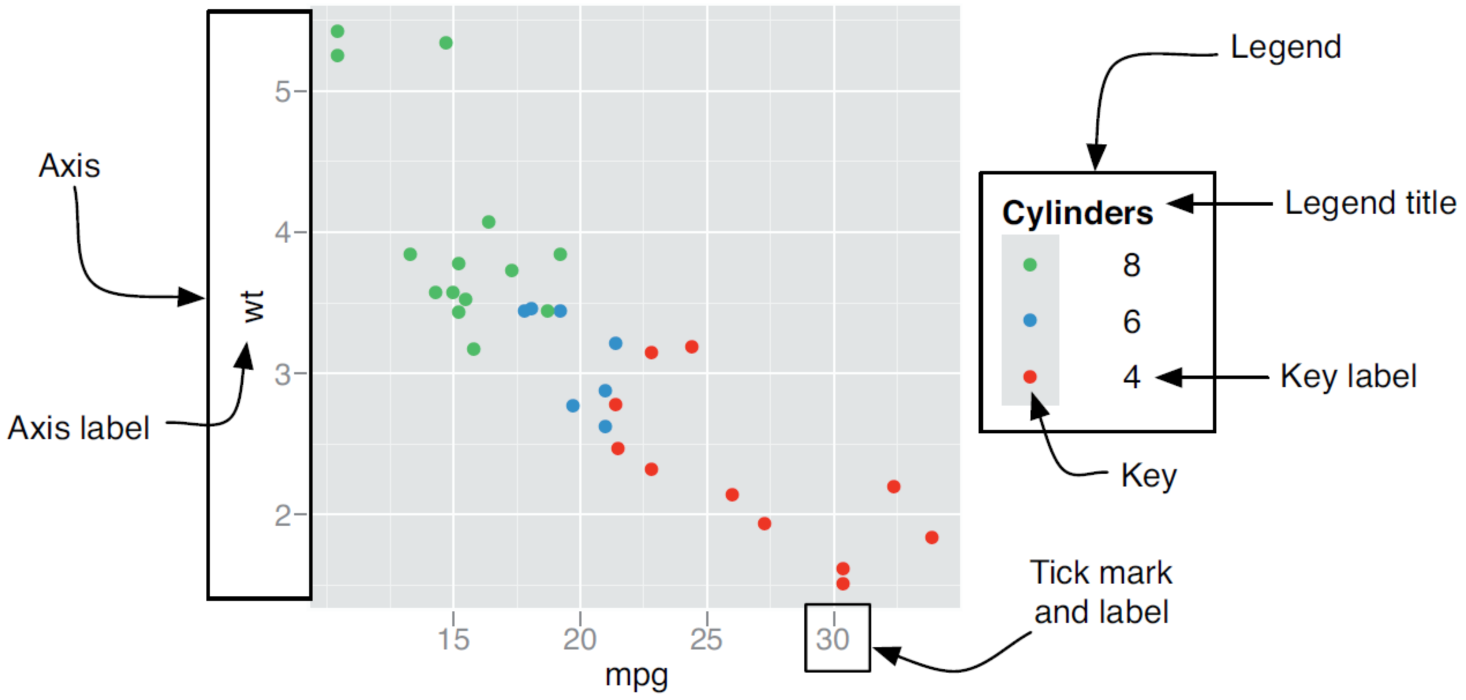




```
dplot + geom_point(aes(colour=color)
dplot + geom_point(aes(colour=color) + facet_grid(. ~ color)
```

From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Axis and legend



From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Scales

❖ Position scales, used to map continuous, discrete and date-time variables onto the plotting region and to construct the corresponding axes.

❖ Colour scales, used to map continuous and discrete variables to colours.

❖ Manual scales, used to map discrete variables to your choice of symbol size, line type, shape or colour, and to create the corresponding legend.

❖ The identity scale, used to plot variable values directly to the aesthetic rather than mapping them.

From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Common arguments

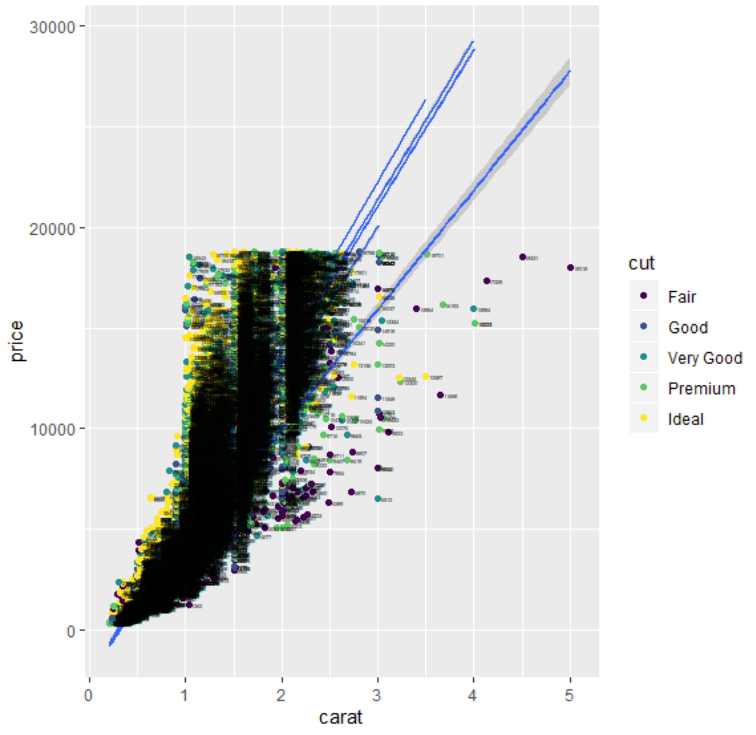
- *name*: sets the label which will appear on the axis or legend.

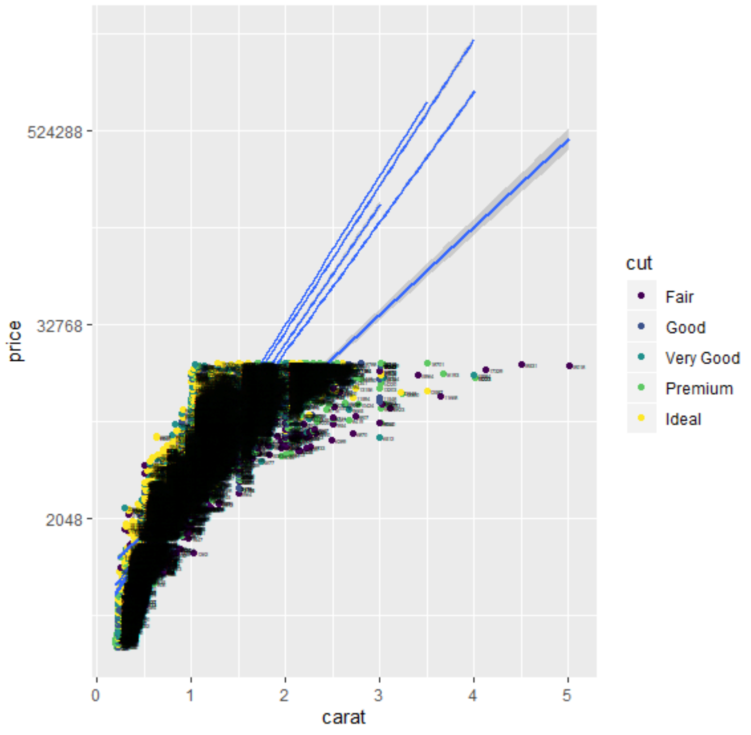
- *limits*: fixes the domain of the scale.

- *breaks* and *labels*: breaks controls which values appear on the axis or
Legend; abels specifies the labels that hould appear at the breakpoints.

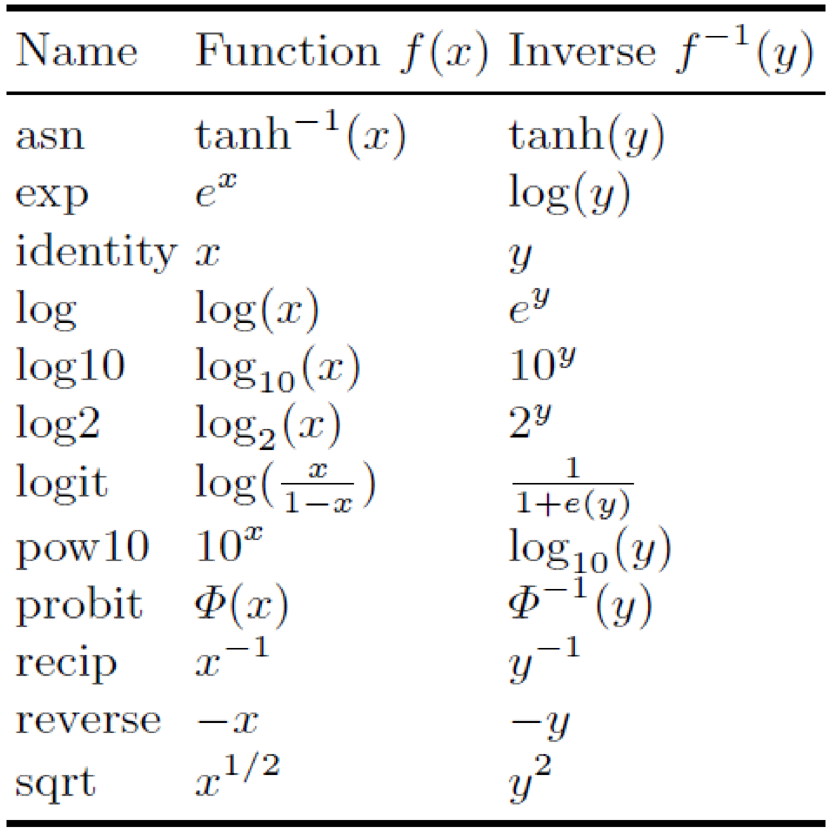
From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Axis scale





```
g3 + scale_y_continuous(trans="log2")
g3 + scale_y_log2()
```

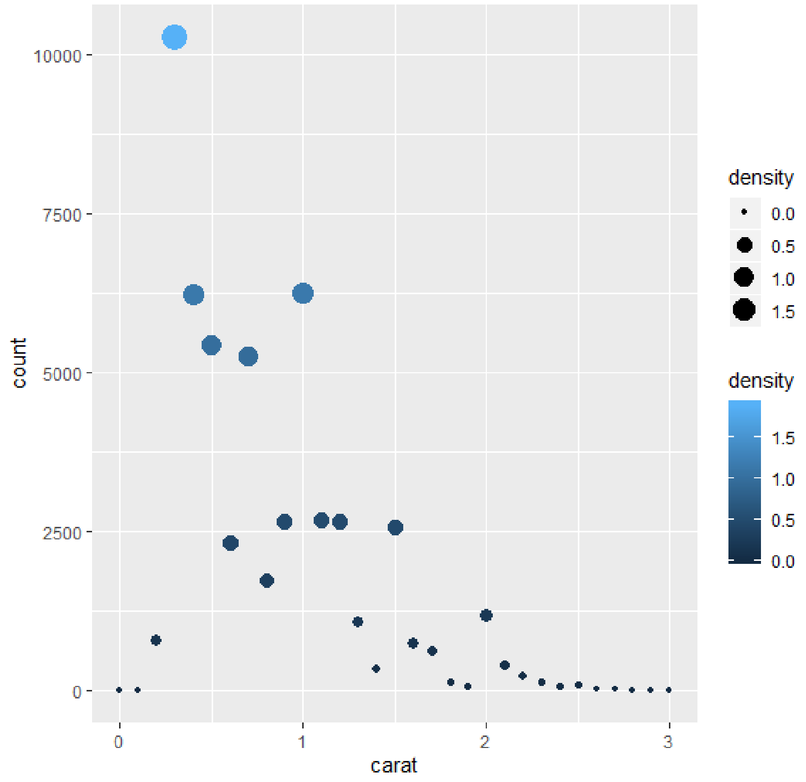


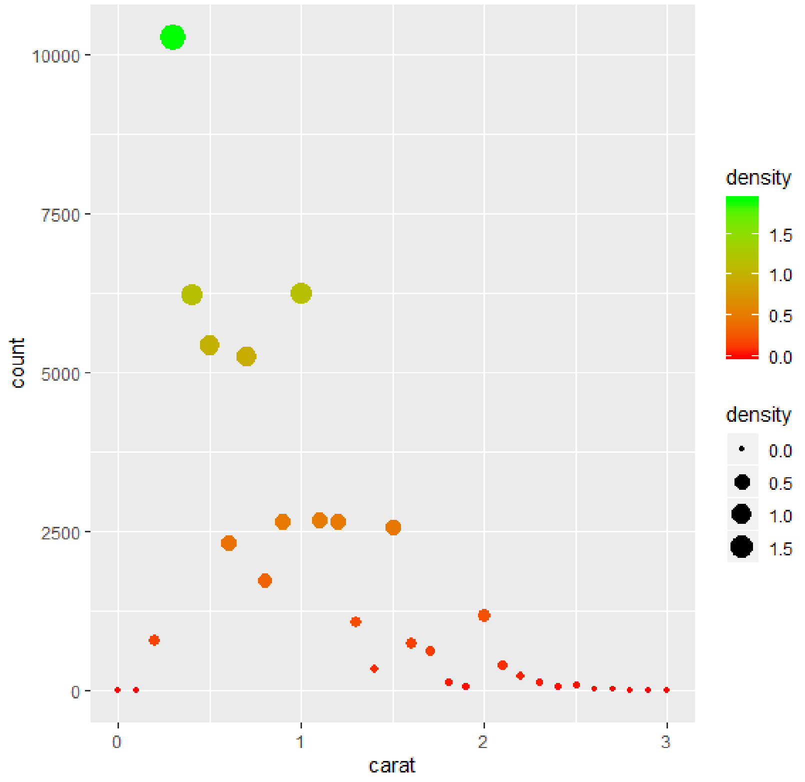
Name	Function $f(x)$	Inverse $f^{-1}(y)$
asn	$\tanh^{-1}(x)$	$\tanh(y)$
exp	e^x	$\log(y)$
identity	x	y
log	$\log(x)$	e^y
log10	$\log_{10}(x)$	10^y
log2	$\log_2(x)$	2^y
logit	$\log(\frac{x}{1-x})$	$\frac{1}{1+e(y)}$
pow10	10^x	$\log_{10}(y)$
probit	$\Phi(x)$	$\Phi^{-1}(y)$
recip	x^{-1}	y^{-1}
reverse	$-x$	$-y$
sqrt	$x^{1/2}$	y^2

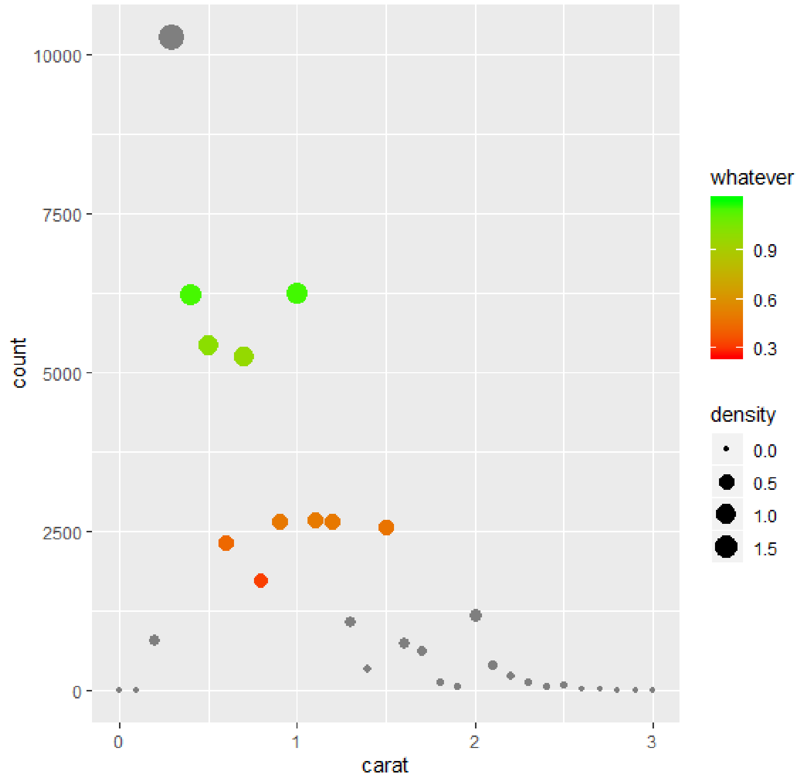
From: ggplot2 -Elegant Graphics for Data Analysis by Hadley Wickham

Colour scale

```
d <- ggplot(diamonds, aes(carat)) + xlim(0, 3)
d + stat_bin(aes(size = ..density.., colour=..density..), binwidth = 0.1, geom = "point", position="identity" )
d + stat_bin(aes(size = ..density.., colour=..density..), binwidth = 0.1, geom = "point", position="identity" ) \
+ scale_color_distiller(palette = 8)
d + stat_bin(aes(size = ..density.., colour=..density..), binwidth = 0.1, geom = "point", position="identity" ) \
+ scale_color_gradient(low="red",high = "green", breaks = c(0.3,0.6,0.9), name = "whatever", limit =c(0.25,1.2) )
```

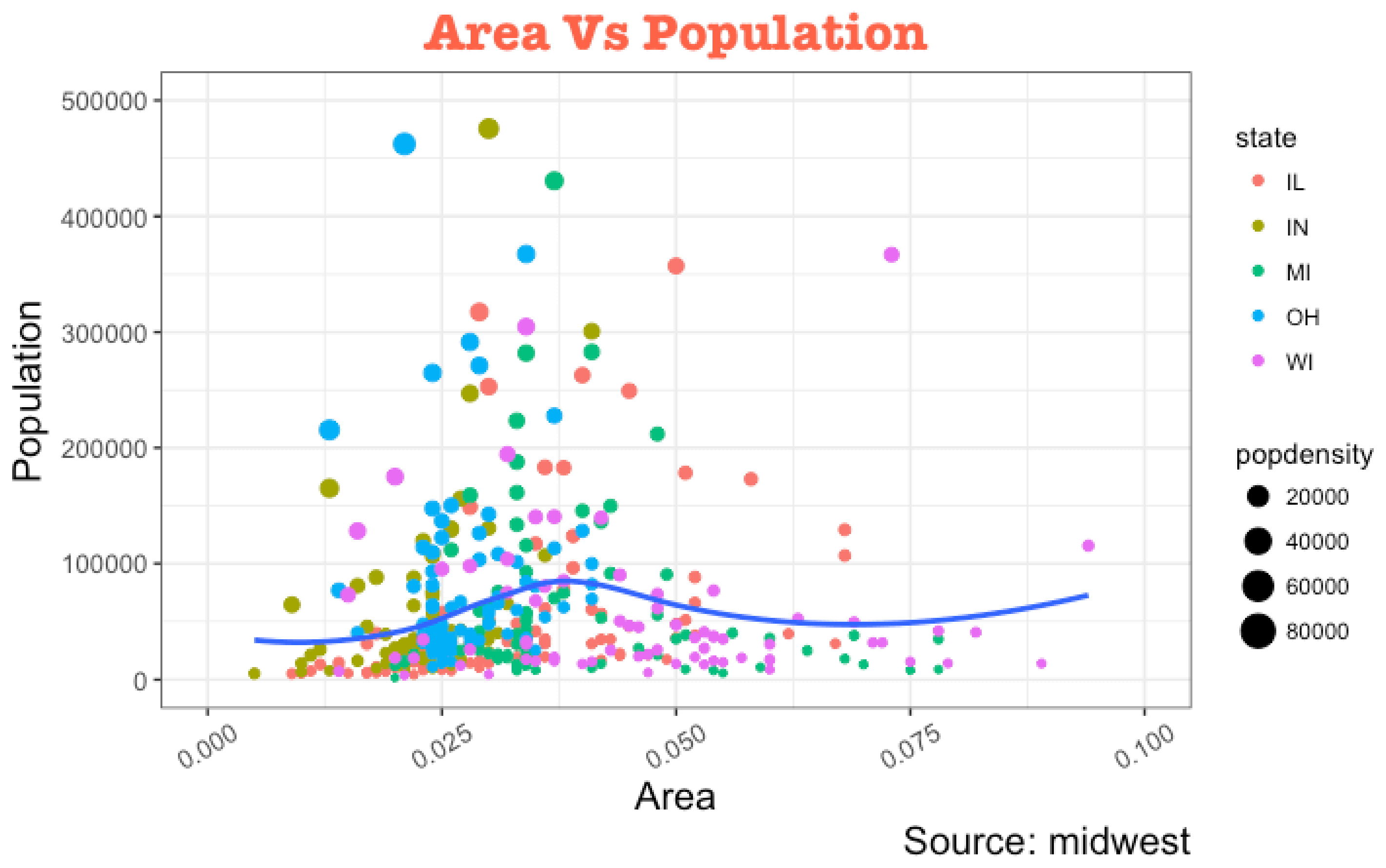






Titles, subtitles and captions

```
plot + labs(title="Area Vs Population", y="Population", x="Area", caption="Source: midwest")
```



http://r-statistics.co/Complete-Ggplot2-Tutorial-Part2-Customizing-Theme-With-R-Code.html

Theme (*theme*)

The appearance of non-data elements of the plot is controlled by the theme system.

❖ Background (color, linetype….)

❖ Title(font, size, position, angle…)

…

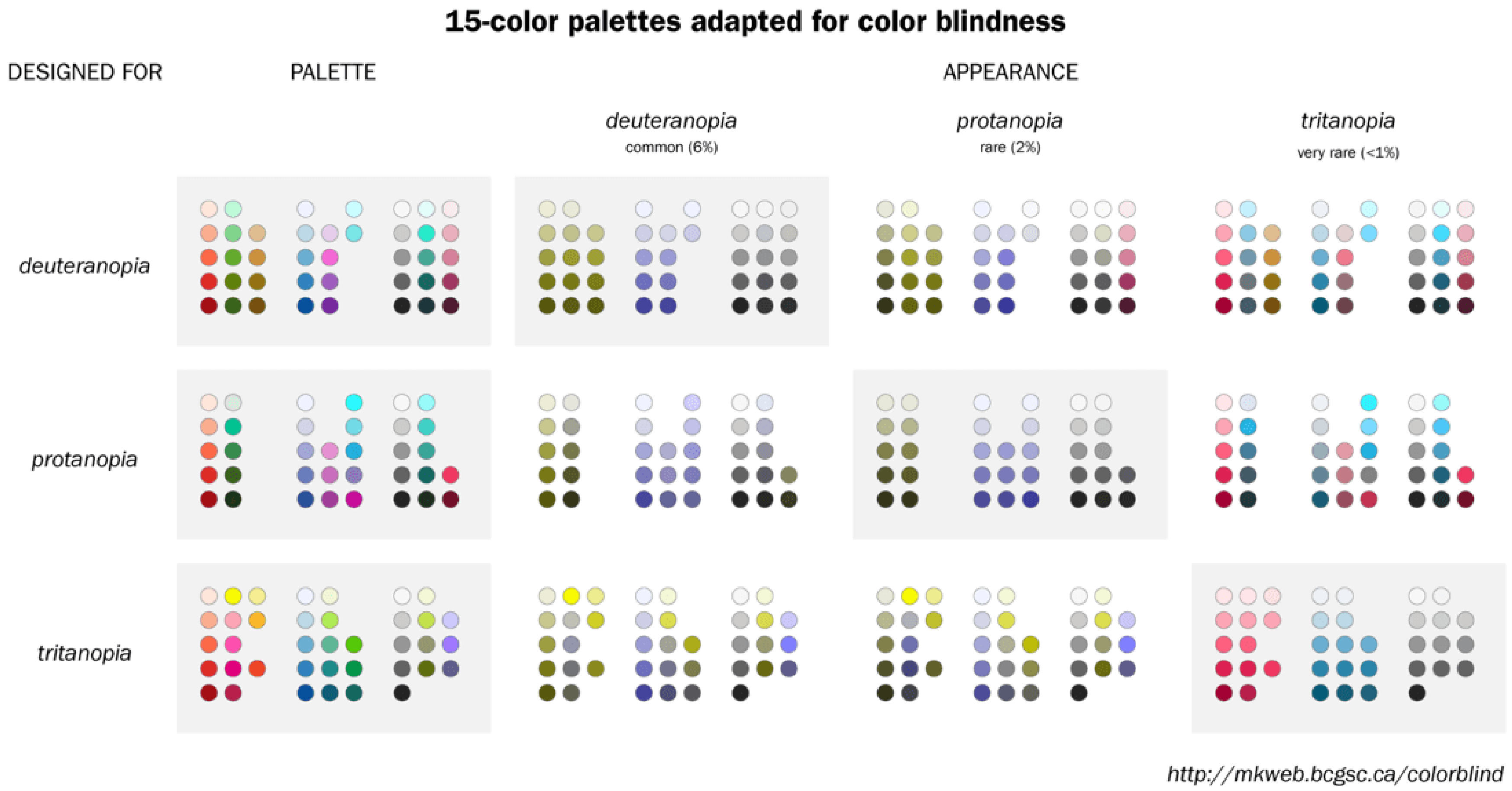
Save a plot to a file

```
png(file=“my_plot.png”, width=500, height=500, units=“px”)
d <- ggplot(diamonds, aes(carat)) + xlim(0, 3)
d + stat_bin(aes(size = ..density.., colour=..density..), binwidth =
0.1, geom = "point", position="identity" )
dev.off()
```

You can also save to bmp, jpeg, tiff, pdf formats by
bmp, *jpeg*, *tiff*, *pdf*

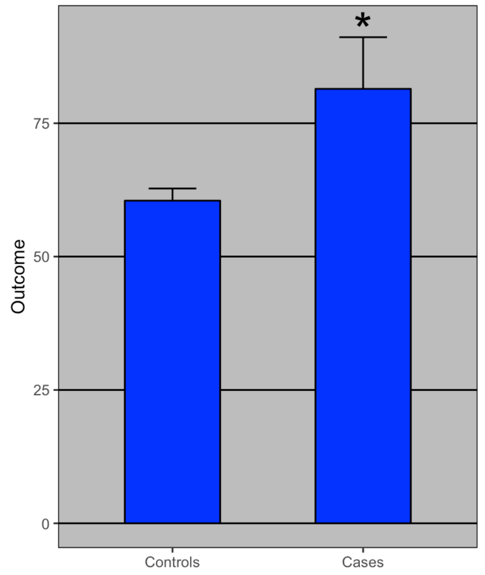
More about data visualization – using colours

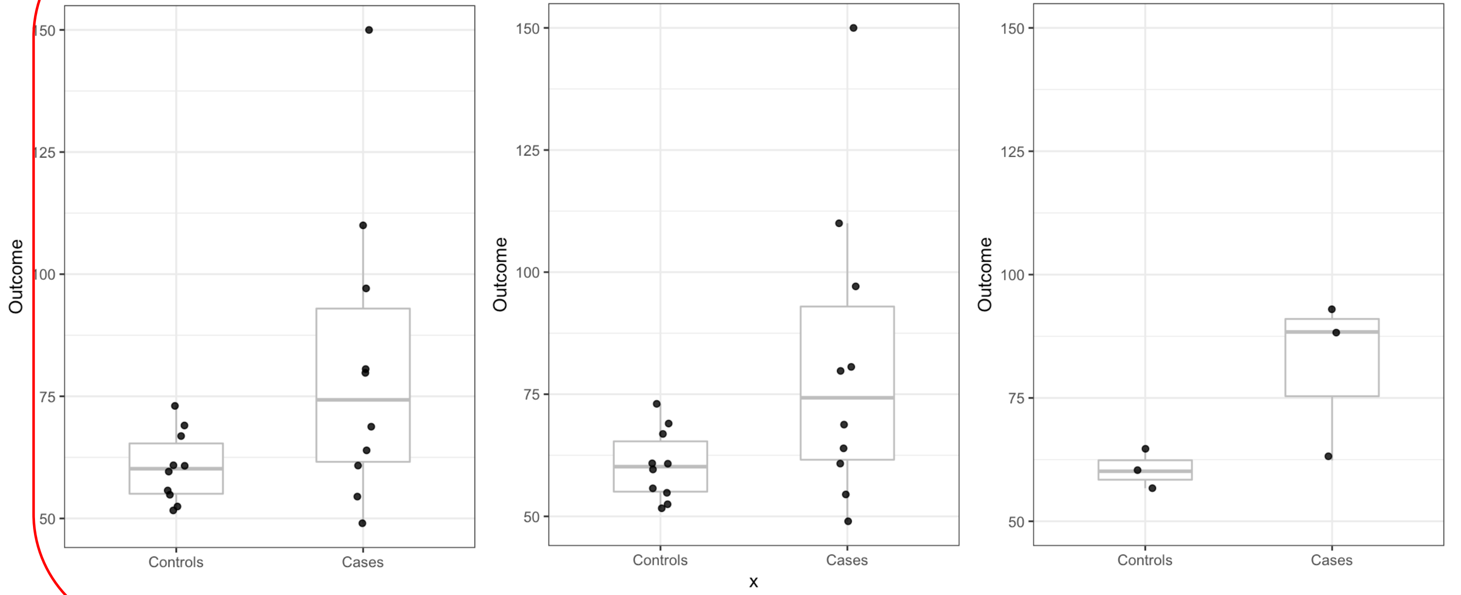
Colour (**color**) **blindness** (**colour** vision deficiency, or CVD) affects approximately
1 in 12 men (8%) and 1 in 200 women in the world.
Try to be color blind friendly.



More about data visualization – using plots

Plots can be misleading or hiding more information





https://simplystatistics.org/2019/02/21/dynamite-plots-must-die/

They produce the same barplot

More about data visualization

- ❖ Make the design of your data visualization fit the data, not the other way around
- ❖ Don't manipulate the data to make it fit your argument
- ❖ Cite the sources of your data
- ❖ Tell a story from the data

Suggested readings

❖ ***ggplot2 -Elegant Graphics for Data Analysis*** by Hadley Wickham

❖ ***R Graphic Cookbook*** by Winston Chang