



GRAPHICS WITH GGLOT2

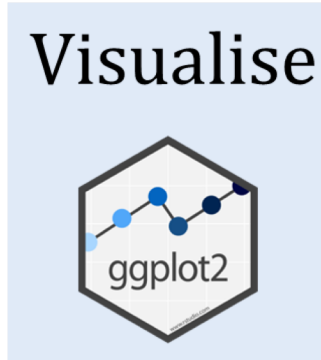
Coral del Val Muñoz

Dpt. Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada

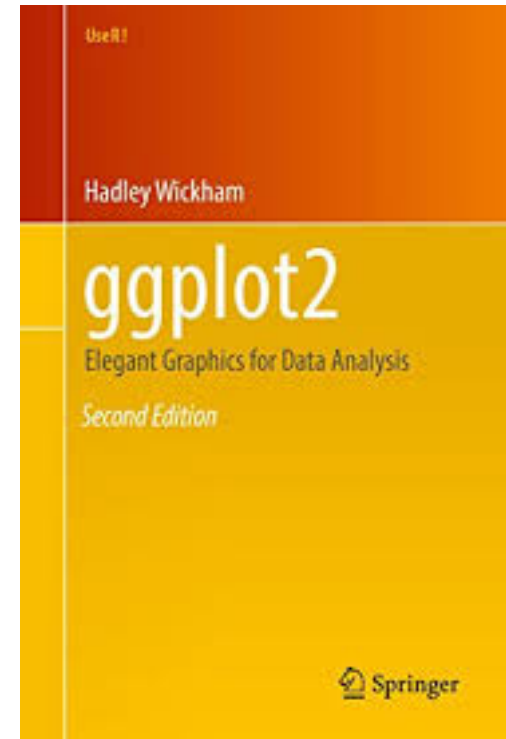
Two plotting systems in R

- The Base plotting system
 - Pre-installed (no need to install extra packages)
 - Plotting function & parameters structure
 - Different plotting functions may have different parameters

What is ggplot2?



<http://www.seec.uct.ac.za/r-tidyverse>



ggplot2

A grammaticalized plotting system

Different plots have the similar grammar structure

Names of the parameters are relatively more user-friendly

How to work

- Data
 - Usually a data frame
 - Can be vectors
- Aesthetic
 - Mapping data to coordinate system(x, y)
 - Distinguish between different groups of data(color, size, shape)
- Geometry
 - Choose the right geometry

How to work

- **data:** in ggplot2, data must be stored as an R dataframe
- coordinate system: describes 2-D space that data is projected onto, for example, Cartesian coordinates, polar coordinates, map projections, ...
- **aesthetics:** describe visual characteristics that represent data-for example, position, size, color, shape, transparency, fill
- **scales:** for each aesthetic, describe how visual characteristic is converted to display values-for example, log scales, color scales, size scales, shape scales, ...
- **stats:** describe statistical transformations that typically summarize data-for example, counts, means, medians, regression lines, ...
- **facets:** describe how data is split into subsets and displayed as multiple small graphs
- **geoms:** describe type of geometric objects that represent data-for example, points, lines, polygons, ...

geoms

name	description
geom_point	Scatterplot
geom_bar	Bar plot
geom_histogram	Histogram
geom_density	Probability distribution plot
geom_boxplot	Box and whiskers plot
geom_text	Textual annotations in a plot
geom_errorbar	Error bars

ggplot2 Packages

```
### install & load ggplot library
```

```
install.packages("ggplot2")  
library("ggplot2")
```

Scatterplot

```
p <- ggplot(iris, aes(x=Sepal.Length, y=Petal.Length))
p <- p + geom_point()
p
# Remove elements outside limits
p + xlim(c(4,6)) + ylim(c(1,2))

# Customizing
p + labs(title="Iris", subtitle="flowers", y="Petal length",
x="Sepal Length", caption="Fig. 1")

# Alternative
p <- ggplot(iris, aes(x=Sepal.Length, y=Petal.Length)) +
geom_point(col="red", size=1)
```

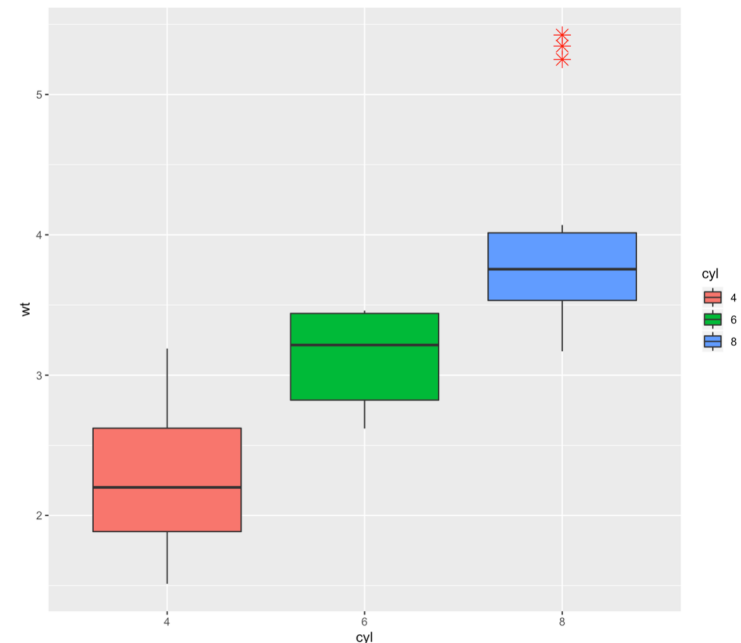
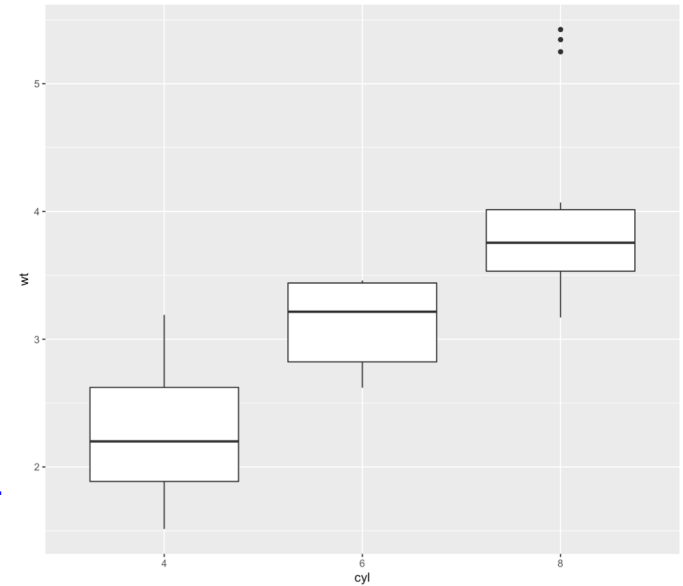

Boxplot

```
# Load the data
data(mtcars)
library(ggplot2)
# Basic box plot
p <- ggplot(mtcars, aes(x=cyl, y=wt)) +
  geom_boxplot()

p
```

```
# Change outlier, color, shape and size
ggplot(mtcars, aes(x=cyl, y=wt, fill=cyl)) +
  geom_boxplot(outlier.colour="red",
    outlier.shape=8,
    outlier.size=4)
```

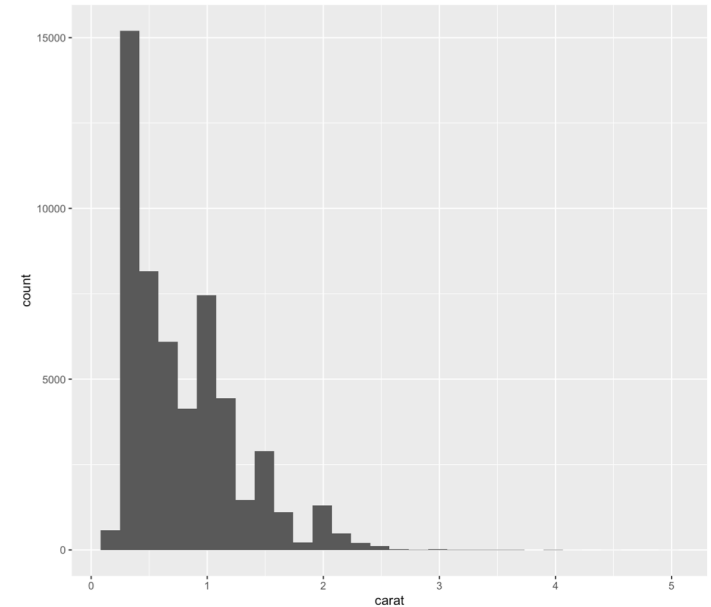
```
# To change the orientation
coord_flip()
```



Histogram

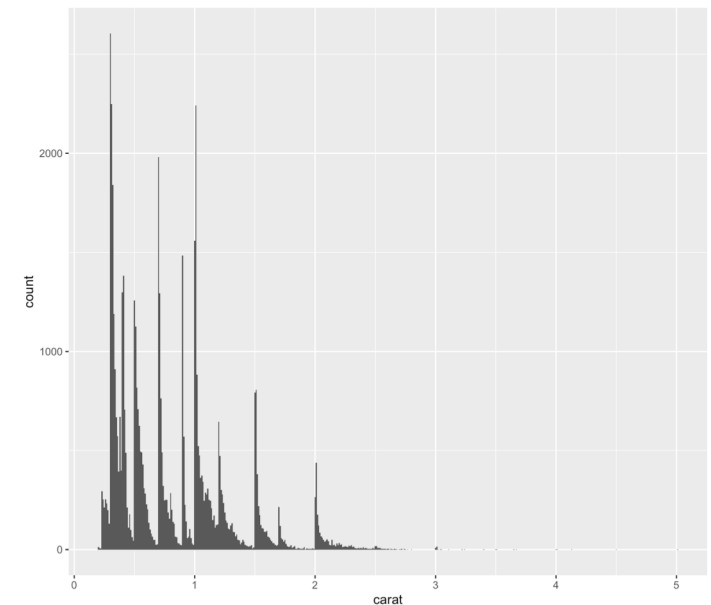
```
# Basic histogram
```

```
ggplot(diamonds, aes(carat)) +  
  geom_histogram()
```



```
# Change the width of bins
```

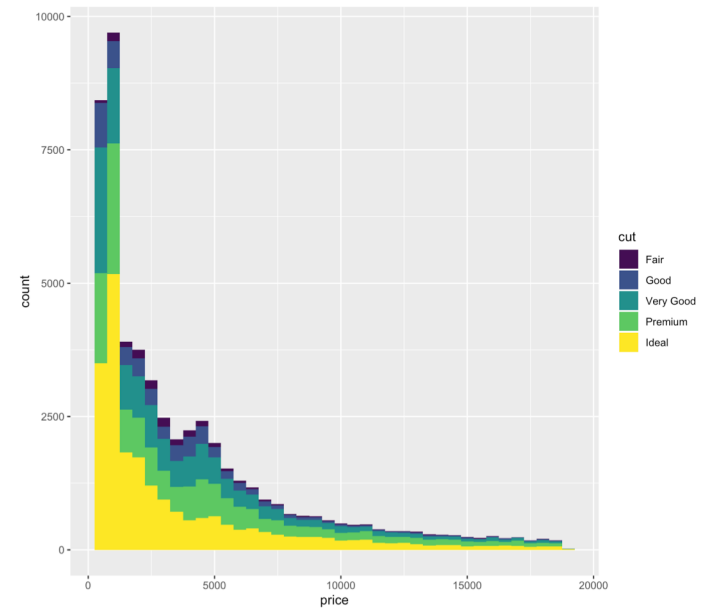
```
ggplot(diamonds, aes(carat)) +  
  geom_histogram(binwidth = 0.01)
```



Histogram

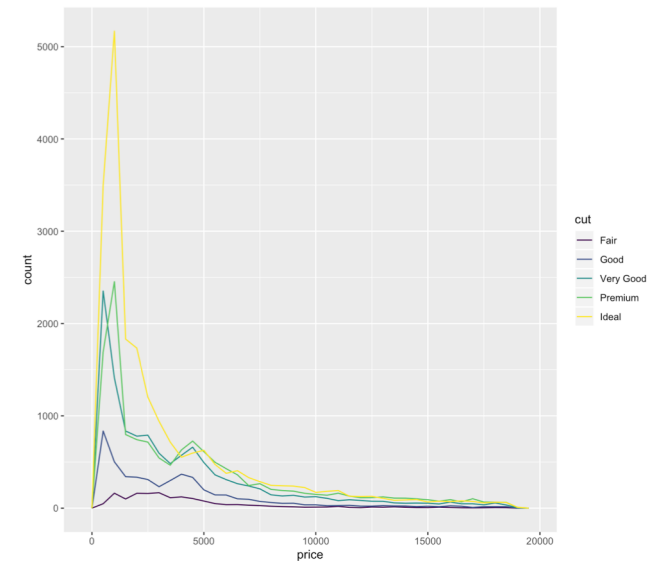
stacking histograms

```
ggplot(diamonds, aes(price, fill = cut)) +  
  geom_histogram(binwidth = 500)
```



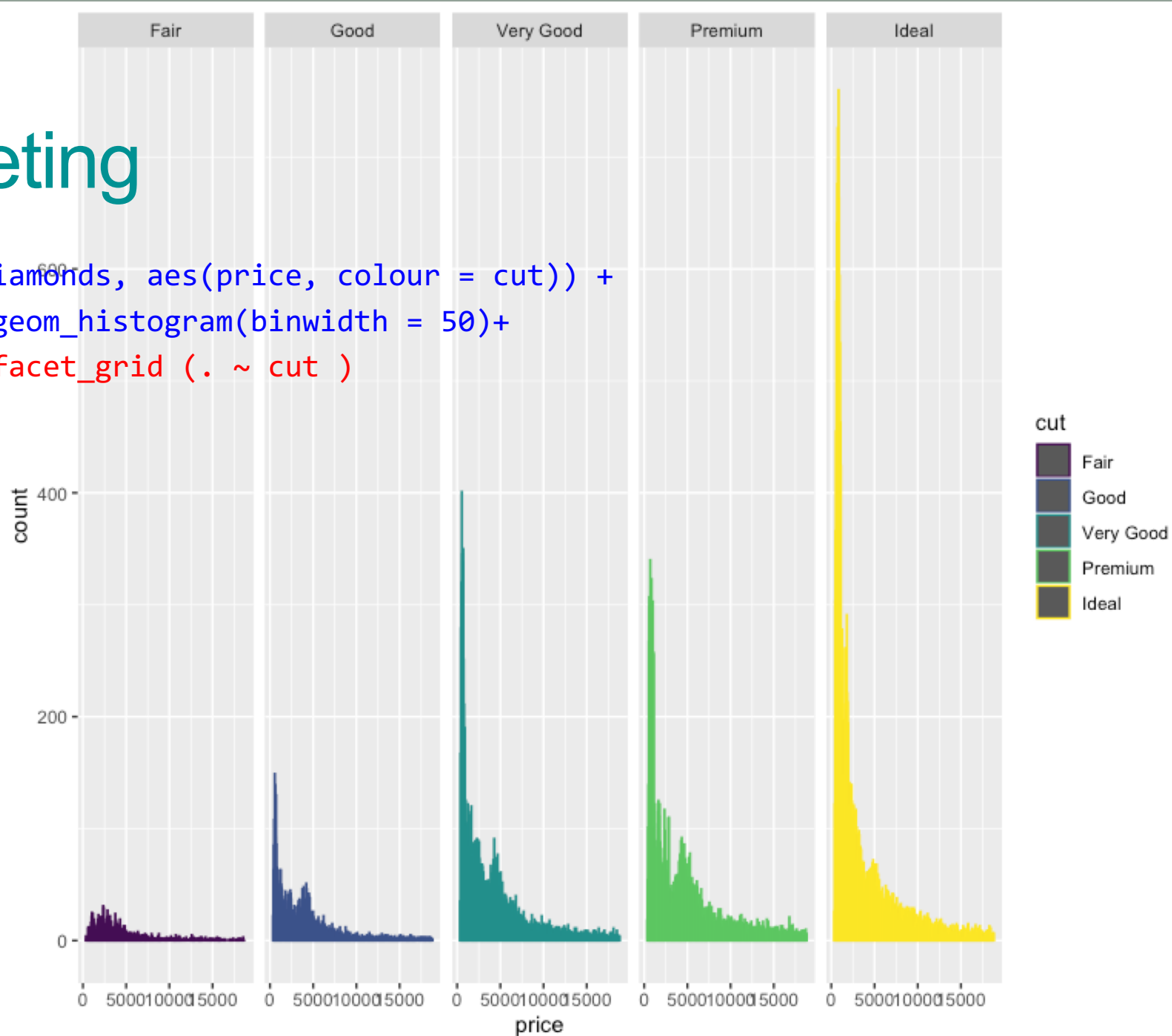
frequency polygons

```
ggplot(diamonds, aes(price, colour = cut)) +  
  geom_freqpoly(binwidth = 500)
```



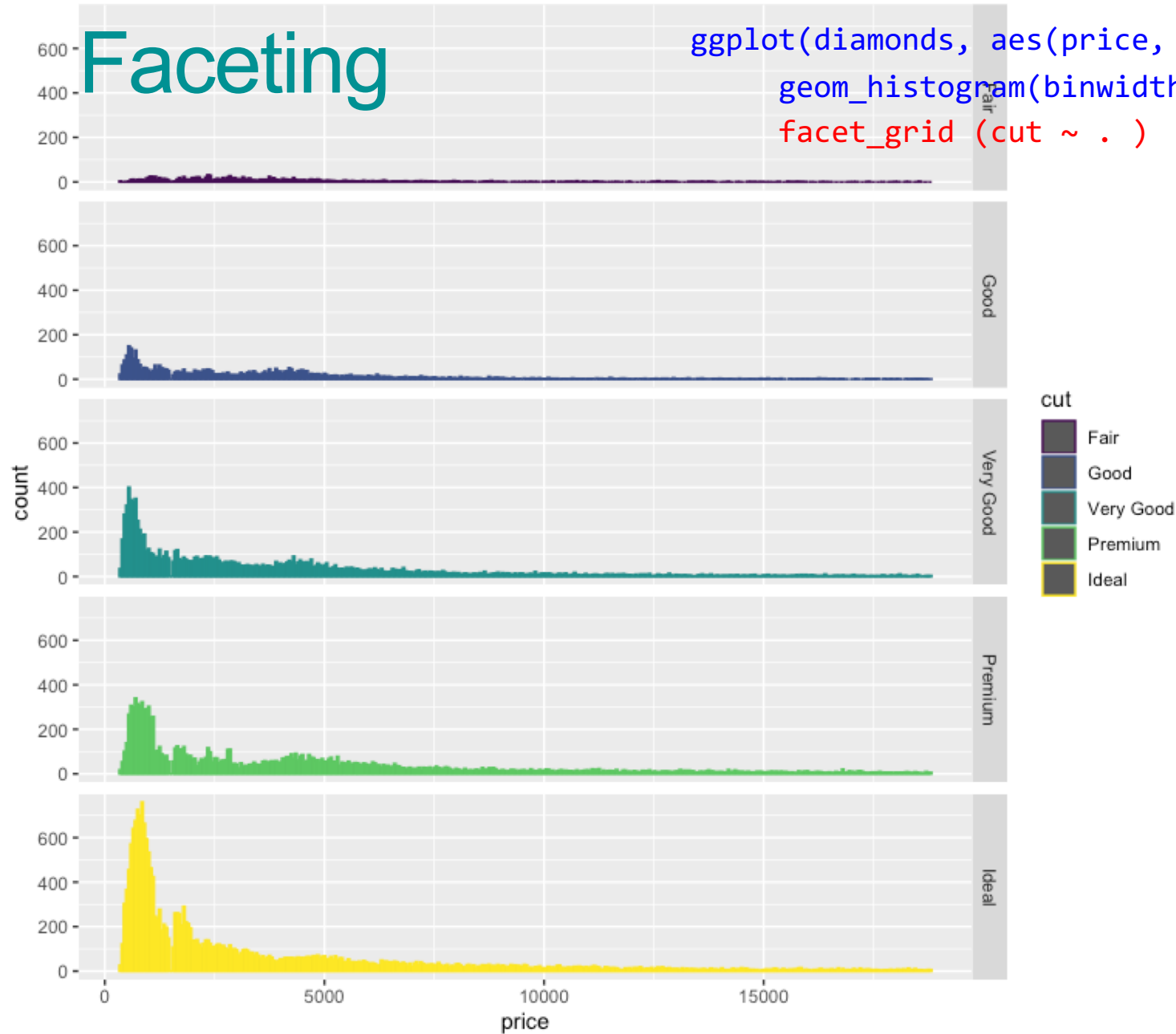
Faceting

```
ggplot(diamonds, aes(price, colour = cut)) +  
  geom_histogram(binwidth = 50)+  
  facet_grid (. ~ cut )
```



Faceting

```
ggplot(diamonds, aes(price, colour = cut)) +  
  geom_histogram(binwidth = 50) +  
  facet_grid (cut ~ . )
```



Change of coordinates

```
p<-ggplot(diamonds, aes(price, colour = cut)) +
  geom_histogram(binwidth = 50)+
  facet_grid (cut ~ . )+
  coord_flip()
```

Change the legend position

```
p + theme(legend.position="top")
p + theme(legend.position="bottom")
```

Remove legend

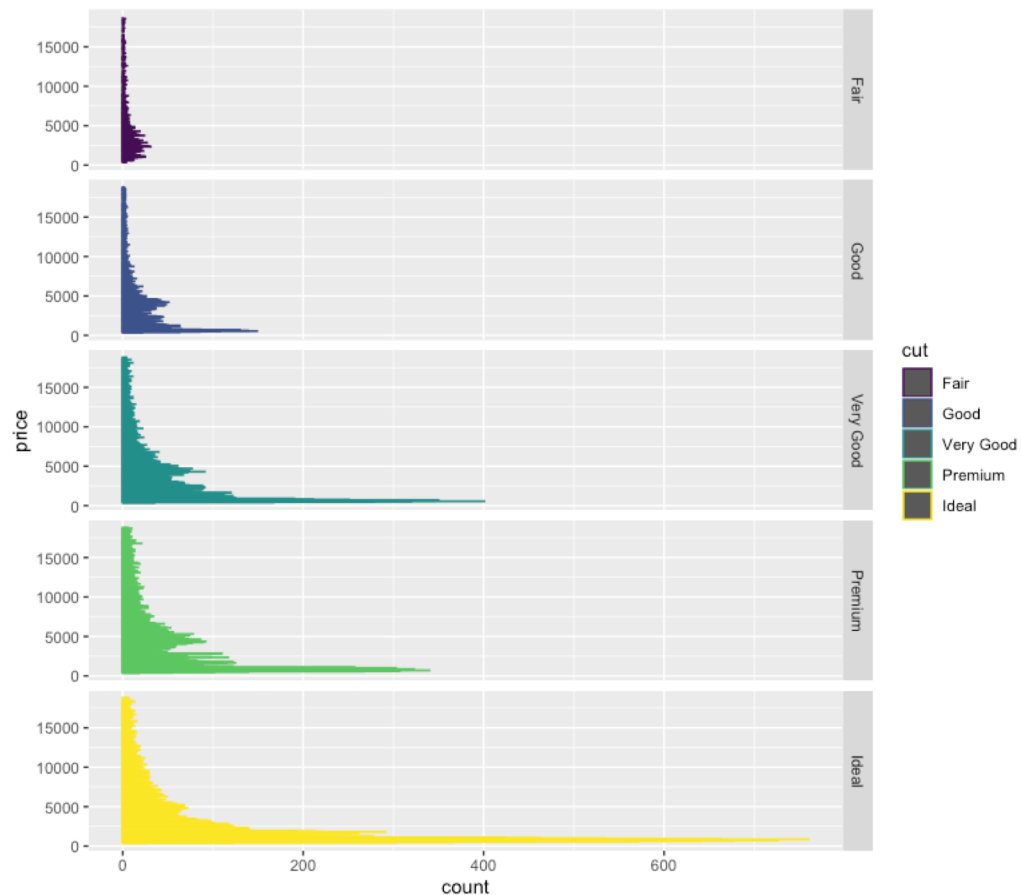
```
p + theme(legend.position="none")
```

Add title

```
p + labs(title="Precio del
diamante\n según el tallado", y =
"Miles")
```

Add formato

```
p + theme_classic()
```



geom_bar()

```
### comparison qplot vs ggplot
# qplot histogram
qplot(clarity, data=diamonds, fill=cut, geom="bar")

# ggplot histogram -> same output
ggplot(diamonds, aes(clarity, fill=cut)) +
  geom_bar()
```

Exercise

- Usa el dataset diamonds para generar un diagrama de barras del numero de diamantes según el tallado “cut”. Usa la geometría `geom_bar()`

```
ggplot(diamonds, aes(x = cut, fill = cut)) +  
geom_bar()
```

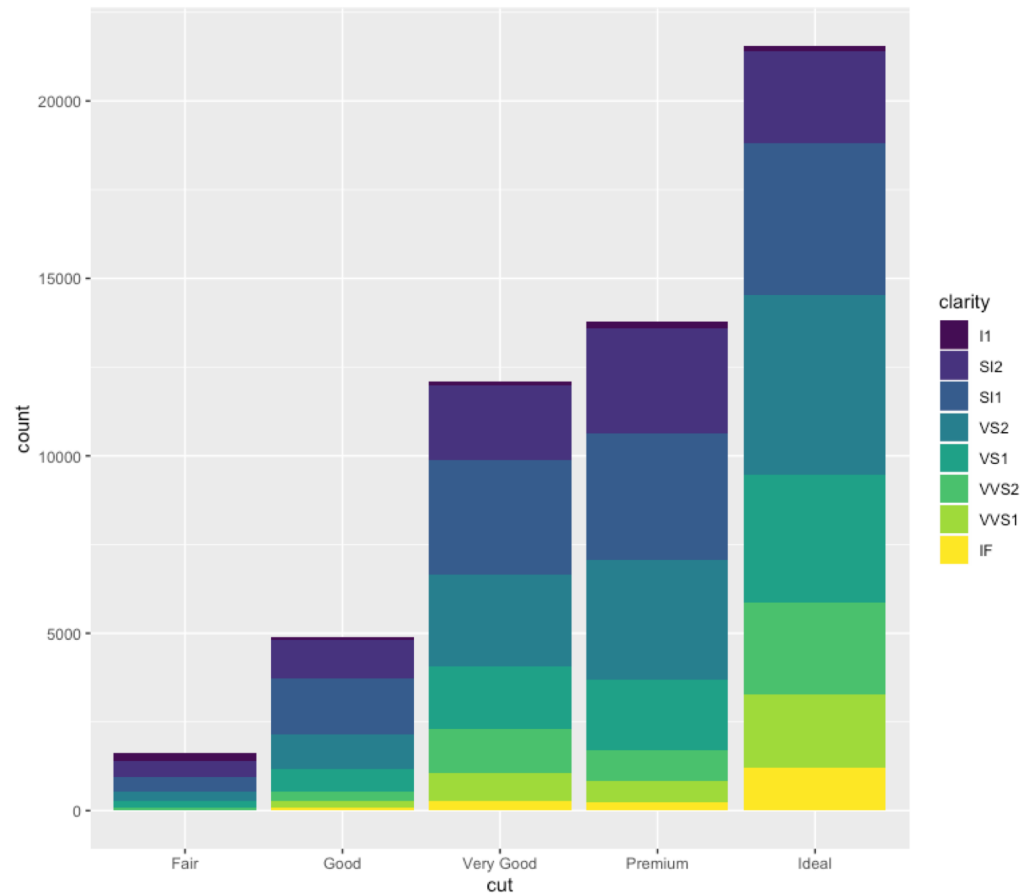
```
ggplot(diamonds, aes(x = cut, color = cut)) +  
geom_bar()
```


Exercise

Añade una segunda variable al tallado “cut”, por ejemplo claridad. Usa la geometría `geom_bar()`

The stacking is performed automatically by the *position adjustment* specified by the *position* argument.

If you don't want a stacked bar chart, you can use one of three other options: **"identity"**, **"dodge"** or **"fill"**:



```
ggplot(diamonds, aes(x = cut, fill  
= clarity)) +  
geom_bar()
```

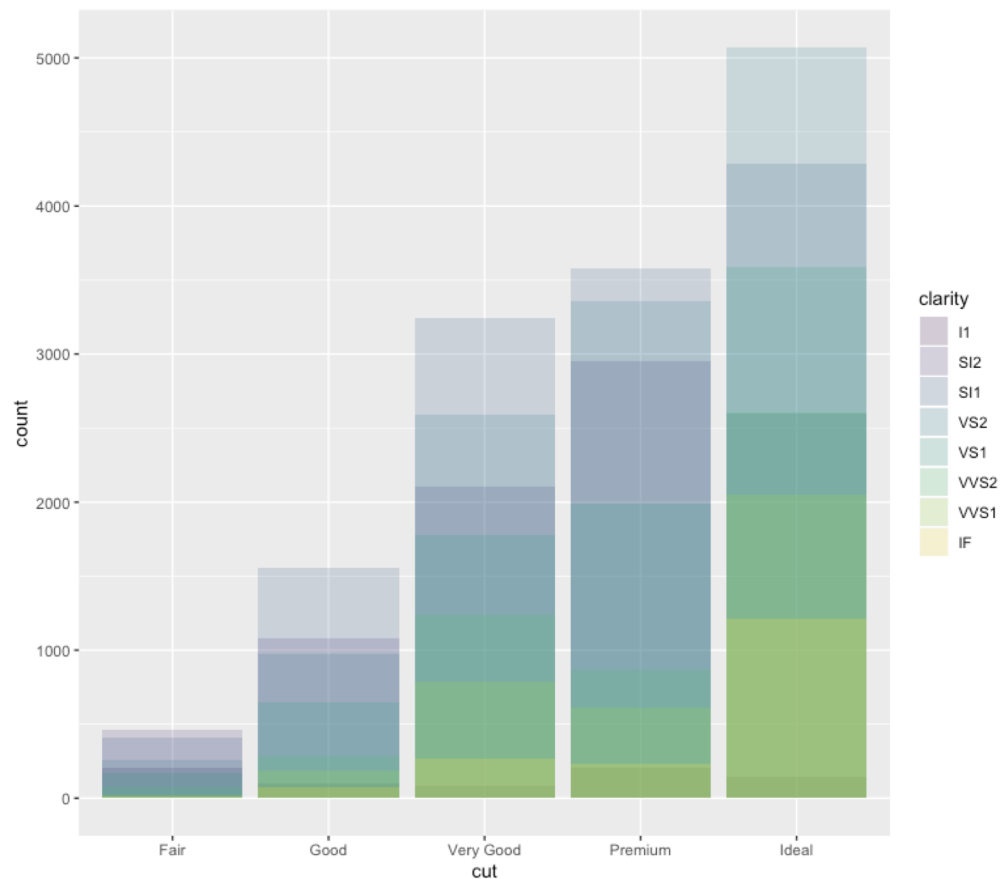
Exercise

- `position = "identity"`
- will place each object exactly where it falls in the context of the graph.
- This is not very useful for bars, because it overlaps them.
- To see that overlapping we either need to make the bars slightly transparent by setting `alpha` to a small value

```
ggplot(diamonds, aes(x = cut, fill = clarity)) +  
  geom_bar(alpha = 1/5, position = "identity")
```

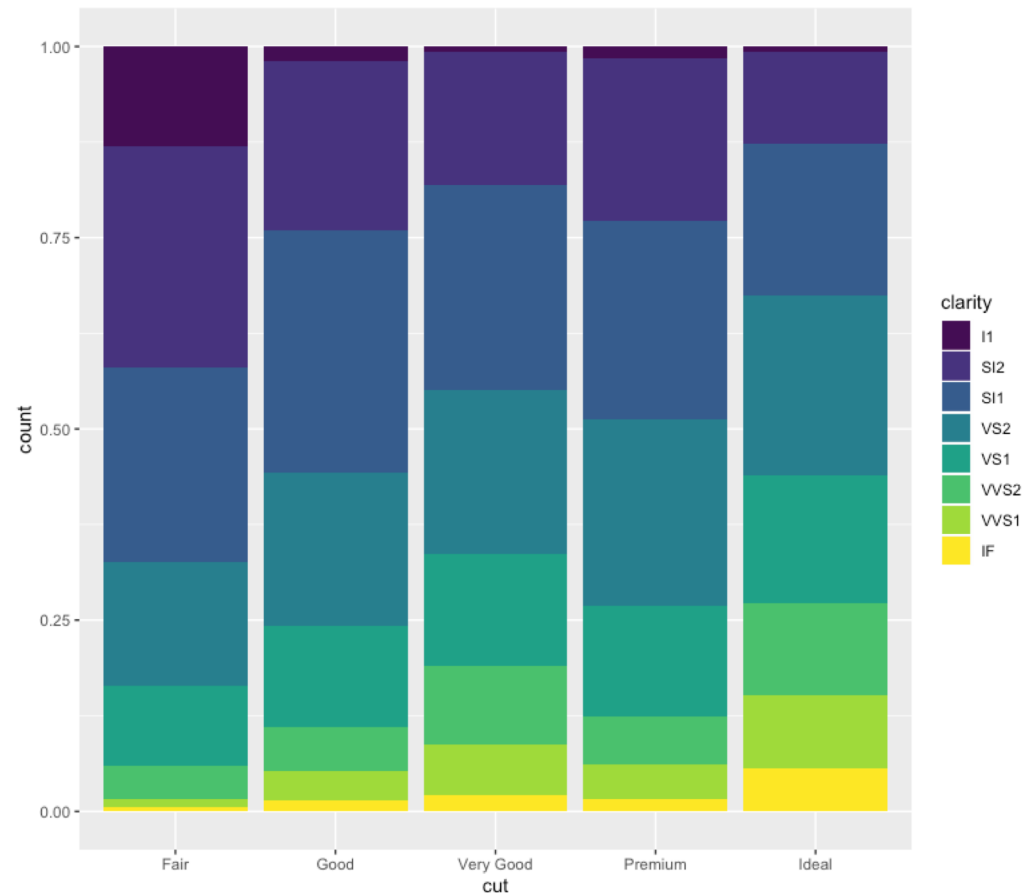
Exercise

```
ggplot(diamonds, aes(x = cut, fill = clarity)) +  
  geom_bar(alpha = 1/5, position = "identity")
```



Exercise

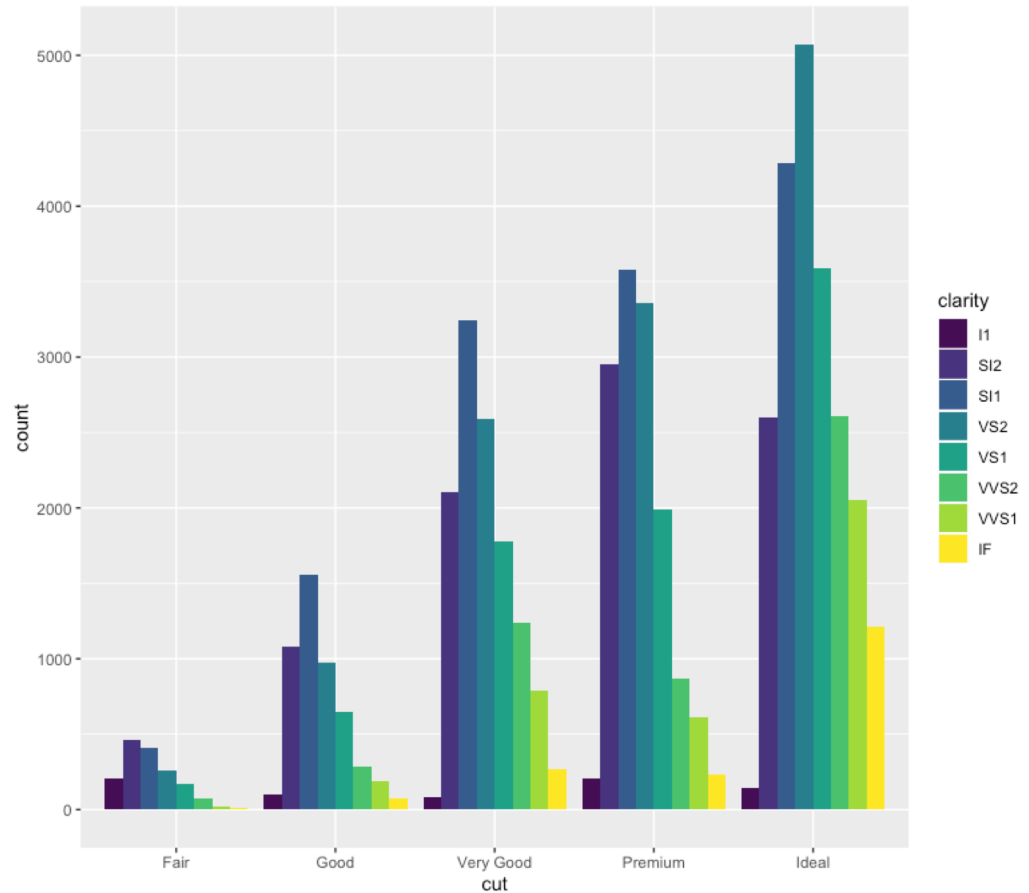
- `position = "fill"`
- works like stacking, but makes each set of stacked bars the same height.
- This makes it easier to compare proportions across groups



```
ggplot(diamonds, aes(x = cut, fill = clarity)) +  
  geom_bar(position = "fill")
```

Exercise

- `position = "dodge"`
- places overlapping objects directly *beside* one another.
- This makes it easier to compare individual values



```
ggplot(diamonds, aes(x = cut, fill = clarity)) +  
  geom_bar(position = "dodge")
```

Customizing plots. Use of themes()

```
theme(line, rect, text, title, aspect.ratio, axis.title, axis.title.x,  
axis.title.x.top, axis.title.x.bottom, axis.title.y, axis.title.y.left,  
axis.title.y.right, axis.text, axis.text.x, axis.text.x.top,  
axis.text.x.bottom, axis.text.y, axis.text.y.left, axis.text.y.right,  
axis.ticks, axis.ticks.x, axis.ticks.x.top, axis.ticks.x.bottom,  
axis.ticks.y, axis.ticks.y.left, axis.ticks.y.right, axis.ticks.length,  
axis.line, axis.line.x, axis.line.x.top, axis.line.x.bottom, axis.line.y,  
axis.line.y.left, axis.line.y.right, legend.background, legend.margin,  
legend.spacing, legend.spacing.x, legend.spacing.y, legend.key,  
legend.key.size, legend.key.height, legend.key.width, legend.text,  
legend.text.align, legend.title, legend.title.align, legend.position,  
legend.direction, legend.justification, legend.box, legend.box.just,  
legend.box.margin, legend.box.background, legend.box.spacing,  
panel.background, panel.border, panel.spacing, panel.spacing.x,  
panel.spacing.y, panel.grid, panel.grid.major, panel.grid.minor,  
panel.grid.major.x, panel.grid.major.y, panel.grid.minor.x,  
panel.grid.minor.y, panel.ontop, plot.background, plot.title, plot.subtitle,  
plot.caption, plot.tag, plot.tag.position, plot.margin, strip.background,  
strip.background.x, strip.background.y, strip.placement, strip.text,  
strip.text.x, strip.text.y, strip.switch.pad.grid, strip.switch.pad.wrap,  
..., complete = FALSE, validate = TRUE)
```

Export graphs

```
pdf("archivo.pdf")  
# R sentences  
ggplot(niris, aes(x=variable, y=value,  
fill=Species)) + geom_bar(stat="identity",  
position=position_dodge())  
dev.off()
```

PDF
PNG
SVG
JPEG
BMP
TIFF

Getting help

- ggplot2 is well documented at <http://docs.ggplot2.org/current/>
- <http://www.sthda.com/english/wiki/ggplot2-essentials>
- Google
- Data visualization: <https://medium.com/@EvanSinar/7-data-visualization-types-you-should-be-using-more-and-how-to-start-4015b5d4adf2>)
- books:
 - ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham. Springer, 2009.-
 - R Graphics Cookbook by Winston Chang. O'Reilly, 2012.
 - The Grammar of Graphics by Leland Wilkinson. Springer, 2005.

Gracias...

