Exploratory vs explanatory graphics & R graphics systems

Introdution to Data Visualization

Marcus Birkenkrahe

October 21, 2024

Contents

- What's a snail?
- Exploratory vs explanatory graphics

- Plot arrays and their use
- Base graphics plot functions
- R packages: `grid`, `lattice`, `ggplot`
- Which graphics package should you use?

Image: Matisse, The Snail/L'Escargot (1953). Gouache on paper, cut and pasted,mounted on canvas, 268.4 x 287 cm). Tate Gallery, London.

1 What's a snail?

- What is the purpose of "graphics". Give some examples!
- What does "The Snail" by Matisse achieve as graphics?
- See also: The Swimming Pool, Matisse and the Chapelle du Rosaire
- Graphics: beauty/aesthetics, abstract representation of data
- From Greek "" to draw, write, or scratch (into stone)
- Technique: color, line, medium, object, artist (subject)
- Origin: something written or drawn (also: basic unit of speech)
- John 8:6-8 "Jesus bent down and started writing on the ground with his finger." The Pharisees leave and Jesus is alone with the woman whom he saved from being stoned to death for adultery.
- Graphics examples: artistic, architecture, marketing, product design

2 'Infographics' vs. 'data visualization'

How would you describe the difference? Got an example?

INFOGRAPHICS DEFINITION	DATA VISUALIZATION
Gives you information	Gives you info (data)
Contains graphics	Can be without graphics
Information without data	Data are central

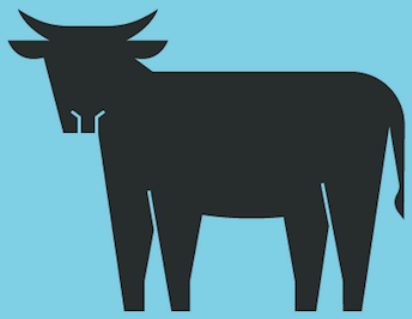
3 Infographics

- Aesthetically rich (i.e. beautiful/nice to look at)
- Manually drawn (i.e. hard to create/alter/check)
- Represents specific data source
- Easy to interpret (depending on quality)

Example: 8-business-model Cow (Leadem, 2017)

ONE COW DESCRIBES 8 BUSINESS MODELS

Business theory can be tricky to understand. Leave it to the power of a simple cow to help you grasp the workings of these popular business models.



DIRECT SALES MODEL

You have one cow.

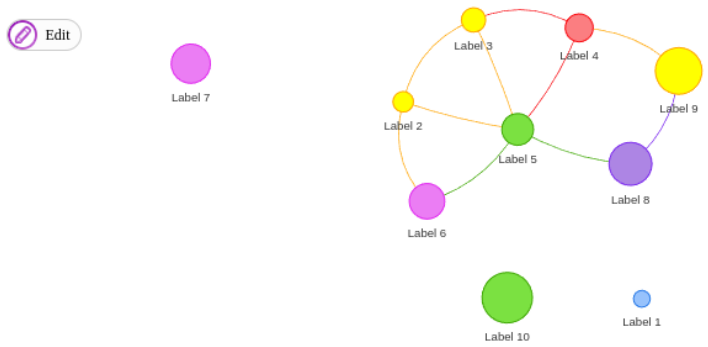
You sell the milk door to door.

4 Data visualization

- Aesthetically barren (i.e. not decorated/beautiful)
- Algorithmically drawn (i.e. easy to create/alter/check)
- Rich in data details
- Harder to interpret (depending on quality)

Example: interactive network visualization with visNetwork in R using JavaScript to visualize complex graphs (Junker, 2019).

```
visNetwork(nodes, edges, width = "100%") %>%
  visOptions(manipulation = TRUE)
```



5 Exploratory vs explanatory data visualization

How would you describe the difference? Got an example?

EXPLANATORY DATA VISUALIZATION	EXPLORATORY DATA VISUALIZATION
Explain	Explore

6 Exploratory data visualization (definition / example)

- Helps us to understand what is in a **data set**

- Alternate name: Exploratory Data Analysis (EDA)
- Quickly **identify** features, curves, lines, trends, anomalies
- Best done at a high level of **granularity**
- Difficulty: separate **signals** from **noise**

Example: CRAN ggplot2 package downloads 2012-2019

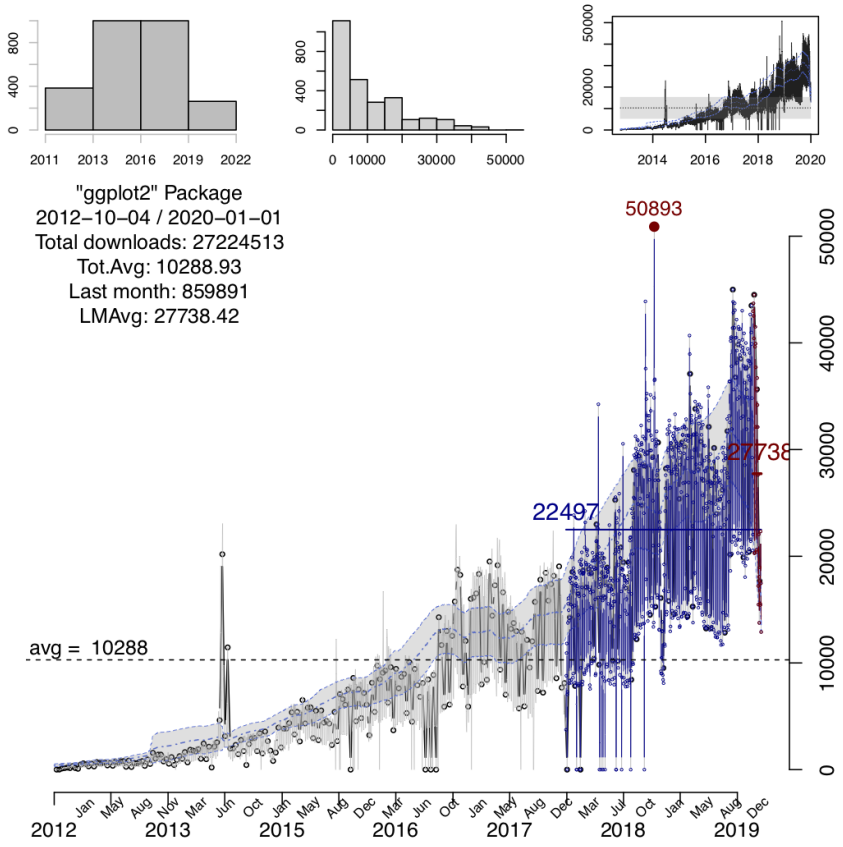


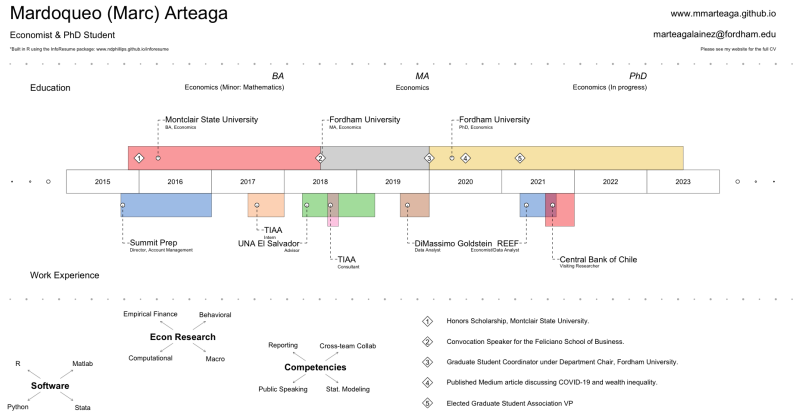
Image source: Visualize.Cran.Downloads (CRAN, 2021)
Python: You can search for downloads of specific packages at pepy.tech

7 Explanatory data visualization

- Help us **convey** findings to others

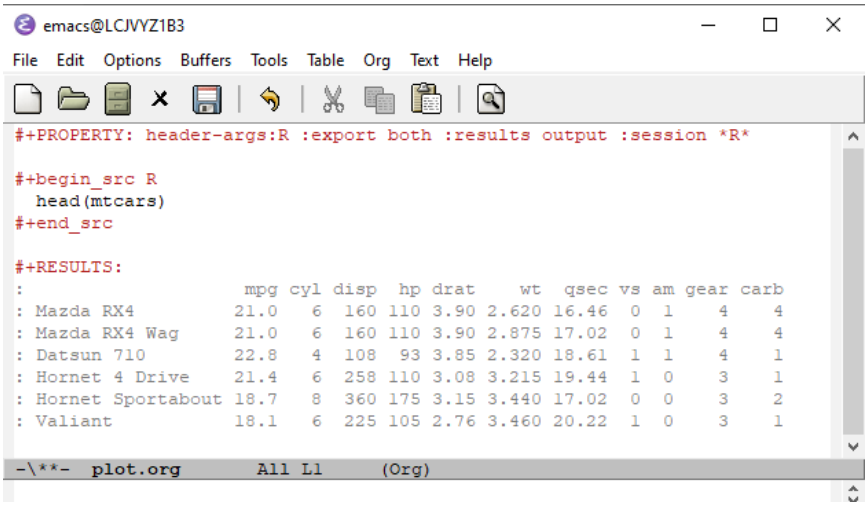
- Alternate name: "Data Storytelling"
- Known to you at the outset (apart from **feedback**)
- **Design** to accommodate a particular **audience**
- Best done after **editorial decisions** what is relevant
- Difficulty: **selecting** focused data that support your **story**

Example: Creating a visual CV using R



Source: How to create a visual CV using R! (Arteaga, 2021)

8 Preparations to code along (for those poor Windoze users)



- Open a new Org-mode file `plot.org` in Emacs
- Put this line at the top of the file `plot.org`:

  `#+PROPERTY: header-args:R :results output :session *R*`
- Activate the code by putting your cursor on the line and entering `C-c C-c`. You should see the message `Local setup has been refreshed` in the minibuffer at the bottom of the editor.
- When you execute your first R code block, you'll be asked where you want the session named `*R*` to run: enter the path to `plot.org`
- For plots, use the header `:results output graphics file :file plot.png`
- When you leave Emacs, you'll be warned that the session `*R*` is active: you can ignore this warning

9 Extended example: US cereal data

- Using: `UScereal` data frame from the `MASS` package
- 11 characteristics of 65 breakfast cereals available for sale
- Information mostly based on the package label required by US FDA
- Do:
  1. load the `MASS` package

2. load the `MASS::UScereal` dataset
3. Look at the dataset structure

```
library(MASS)  # load MASS package
data(UScereal) # load UScereal data frame
str(UScereal)  # display data frame structure
```

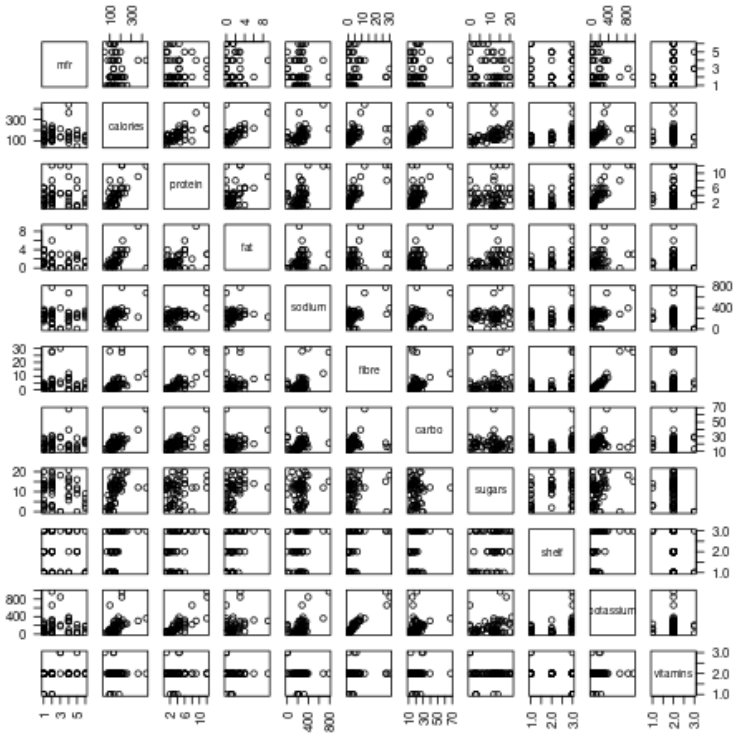
```
'data.frame': 65 obs. of  11 variables:
 $ mfr      : Factor w/ 6 levels "G","K","N","P",..: 3 2 2 1 2 1 6 4 5 1 ...
 $ calories : num  212 212 100 147 110 ...
 $ protein  : num  12.12 12.12 8 2.67 2 ...
 $ fat      : num  3.03 3.03 0 2.67 0 ...
 $ sodium   : num  394 788 280 240 125 ...
 $ fibre    : num  30.3 27.3 28 2 1 ...
 $ carbo    : num  15.2 21.2 16 14 11 ...
 $ sugars   : num  18.2 15.2 0 13.3 14 ...
 $ shelf    : int  3 3 3 1 2 3 1 3 2 1 ...
 $ potassium: num  848.5 969.7 660 93.3 30 ...
 $ vitamins : Factor w/ 3 levels "100%","enriched",..: 2 2 2 2 2 2 2 2 2 2 ...
```

10 Exploratory plot array

A useful, and common, exploratory plot is a panel of pairwise scatterplots to summarize the data frame.

- Make a pairplot of `UScereal` with the parameter `las = 2`.

  ```
  plot(UScereal, las = 2)
  ```



- The `las` parameter sets the position of the tick labels - `las=2` means always perpendicular to the axis.
- Diagonal elements of the array list the name of the variable in the x-axis of all plots in that column, and the y-axis of all plots in that row.
- For 11 variables, 110 plots are shown. Some indicate strong relationships, e.g. `fat` and `calories`.
- Some variables, like `vitamins`, show only few values

- Question: how can you find out more about the `las` parameter?

  1. Look up `help(plot)` - you will find it (`C-s las`) in the examples as an `axis` parameter.
  2. Look up `help(axis)` - you will find it (`C-s las`) as an argument of `par`, the list that contains all graphical parameters (like `options` contains display parameters).

3. Look up the `las` parameter definition in `help(par)`.

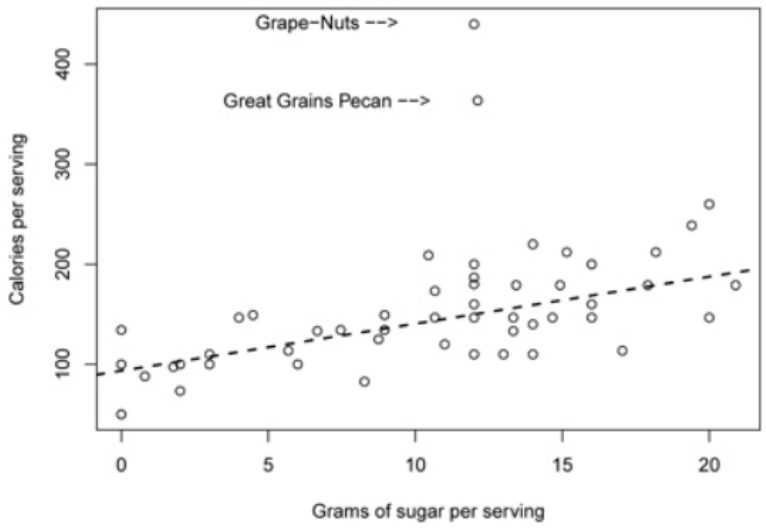
– Check the structure of `par()`, list `par()`, check the value of `las`:

```
##str(par())
par()$las
```

```
[1] 0
```

11 Explanatory scatterplot

- What is this plot trying to say?



– Scatterplot of calories vs. sugar variables, augmented with a robust regression line ("robustness" refers to assumptions about the data)

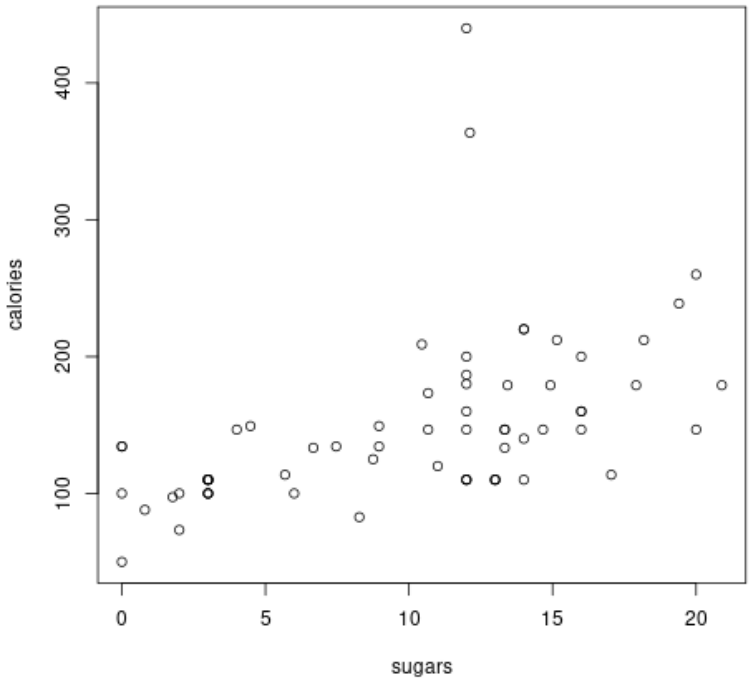
– Dashed line highlights the trend our eye sees in the data if we ignore the two outlying points.

– The outliers correspond to cereals that have much higher calories than any of the others.

– The annotation of labels and text in the plot further aids the interpretation.

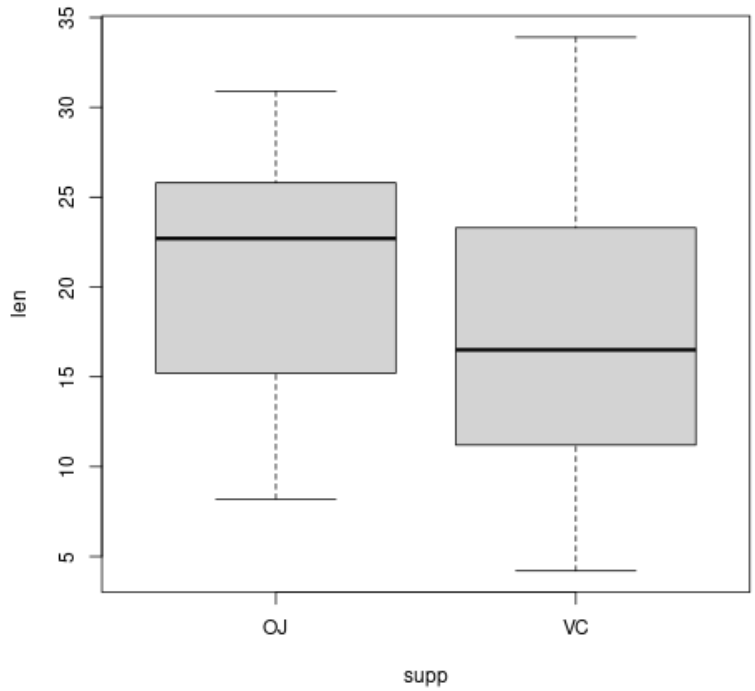
- Let's plot it.

```
library(MASS)
plot(data=UScereal, calories ~ sugars)
```



- You must be careful which variables you're trying to plot:

```
plot(data=ToothGrowth, len ~ supp)
```



12 Base graphics plot functions

- Base graphics is the system originally built into the R language
- It's most common generic function is `plot`
- Base graphics are controlled by 72 graphics *parameters*
- Displays can be customized by *low-level* plotting functions
- Examples: `abline`, `lines`, `points`, `text`, `legend` etc.

[1] Plot types not seen in this lecture yet: sunflower plots (scatterplots that reduce overplotting by turning multiple points into petals); mosaic plots (mosaic of rectangles whose height represents the proportional value); bubbleplots (scatterplot with a third dimension represented with the size of the dots).

FUNCTION	OBJECT TYPE	NATURE OF PLOT[1]
`plot`	Many	Depends on object type
`barplot`	Numeric	Bar plot
`boxplot`	Formula, numeric, list	Boxplot summary
`hist`	Numeric	Histogram
`sunflowerplot`	Numeric + Numeric	Sunflower plot
`mosaicplot`	Formula or table	Mosaic plot
`symbols`	Multiple numeric	Bubbleplots etc.

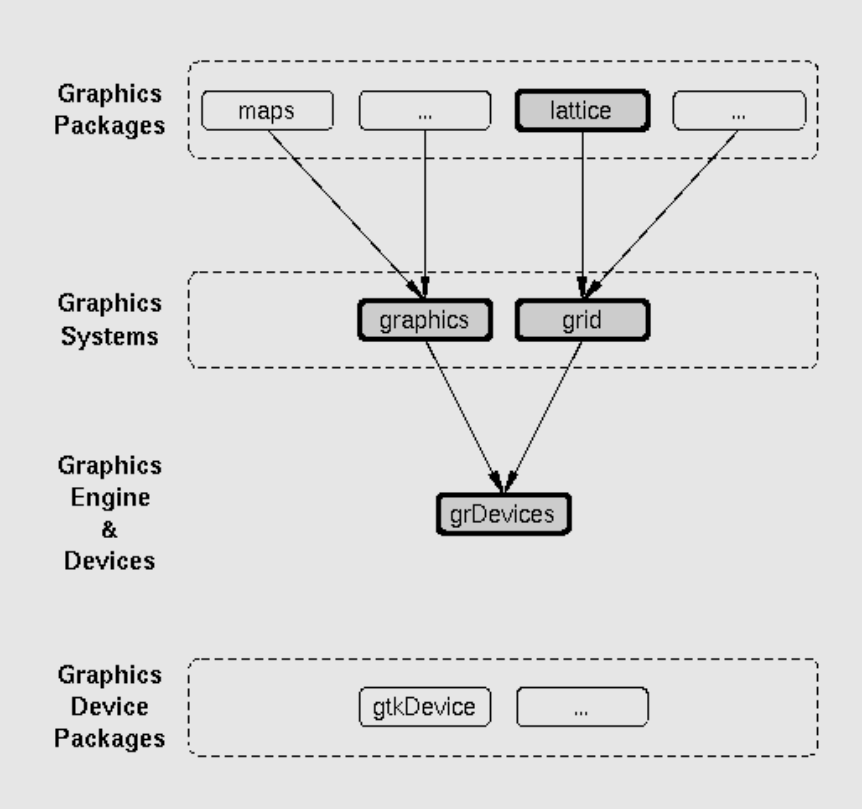
13 Many many types of built-in diagrams

- There are more than 40 types of useful diagrams in R

14 `grid` graphics

- The `grid` package uses the grDevices graphics engine
- Some packages use it, e.g. vcd (for graphing categorical variables)
- Nothing to do with the `grid` function of the base R package (which draws a grid over the plot, see `?grid`)
- More info: Paul Murrell's documents

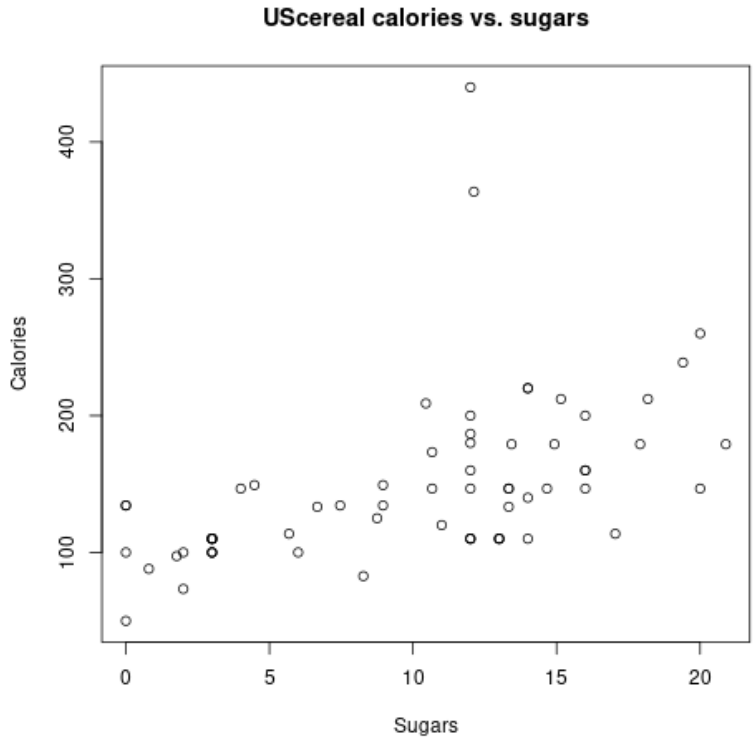
Figure 1: Source: R graph gallery



15 `grid` graphics example

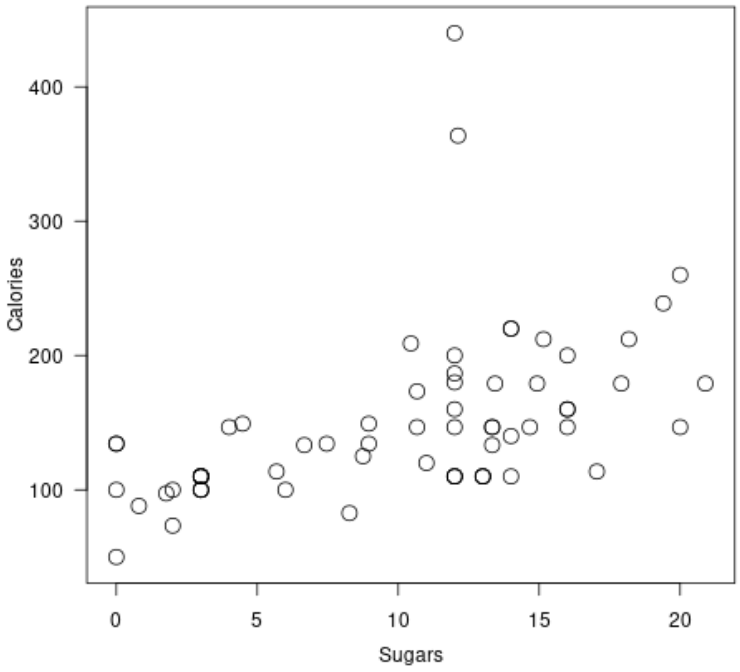
- The following plot is generated with the base R (built-in) package. Below you find the code to create this plot using the `grid` package, demonstrating the greater flexibility but also steeper learning curve.
- Scatterplot with base R `plot`

```
library(MASS)
plot(UScereal$sugars,UScereal$calories,
     xlab="Sugars",
     ylab="Calories")
title("UScereal calories vs. sugars")
```



- Scatterplot with `grid` package:

```
library(MASS)
x <- UScereal$sugars
y <- UScereal$calories
library(grid)
pushViewport(plotViewport())
pushViewport(dataViewport(x,y))
grid.rect()
grid.xaxis()
grid.yaxis()
grid.points(x,y)
grid.text("Calories",x=unit(-3,"lines"),rot=90)
grid.text("Sugars",y=unit(-3,"lines"),rot=0)
popViewport(2)
```



16 `lattice` graphics

- Based on grid graphics, shipped with base R (needs to be loaded)
- Alternative implementation to many standard plotting functions, including scatterplots, bar charts, boxplots, histograms, QQ-plots
- lattice has different default options for plot customization and some additional features, like the *multipanel conditioning plot*
- An example from `UScereal`

```
library(MASS)
str(UScereal)
```

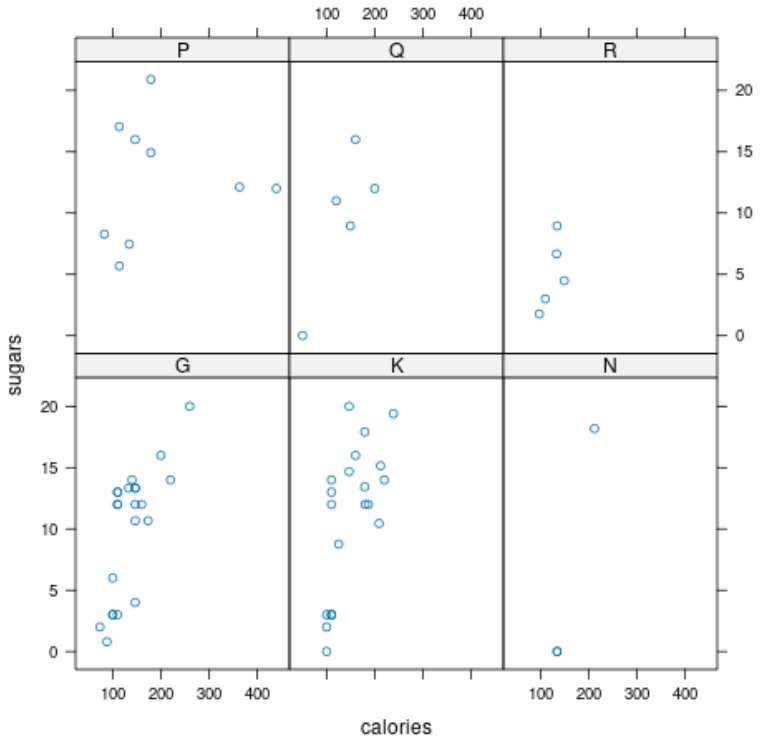
```
'data.frame': 65 obs. of  11 variables:
 $ mfr      : Factor w/ 6 levels "G","K","N","P",..: 3 2 2 1 2 1 6 4 5 1 ...
```

```
 $ calories : num  212 212 100 147 110 ...
 $ protein  : num  12.12 12.12 8 2.67 2 ...
 $ fat      : num  3.03 3.03 0 2.67 0 ...
 $ sodium   : num  394 788 280 240 125 ...
 $ fibre    : num  30.3 27.3 28 2 1 ...
 $ carbo    : num  15.2 21.2 16 14 11 ...
 $ sugars   : num  18.2 15.2 0 13.3 14 ...
 $ shelf    : int  3 3 3 1 2 3 1 3 2 1 ...
 $ potassium: num  848.5 969.7 660 93.3 30 ...
 $ vitamins : Factor w/ 3 levels "100%","enriched",..: 2 2 2 2 2 2 2 2 2 2 ...
```

- Plot calories vs sugars and group by manufacturer ("G": General Mills,"K": Kellogg's,"N": Nabisco,"P": Post,"Q": Quaker Oats,"R": Ralston Purina):

```
library(MASS) # load MASS package for Cars93 data set
library(lattice) # load lattice package

## plot MPG.city vs. Horsepower, conditioned by Cylinders
xyplot(sugars ~ calories | mfr, data = UScereal)
```



- Another example for the `Cars93`

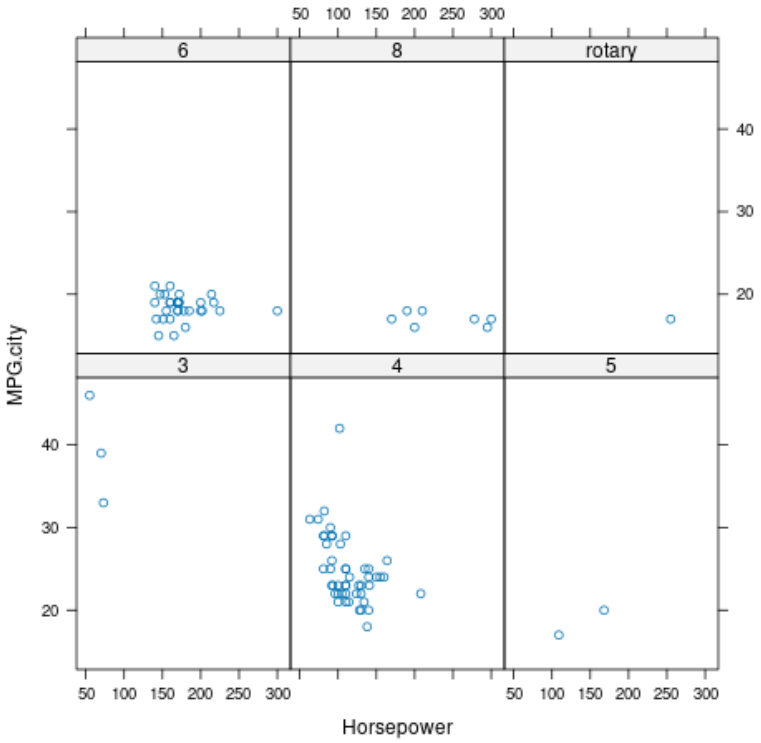
```
library(MASS)
str(Cars93)
```

```
'data.frame': 93 obs. of  27 variables:
 $ Manufacturer      : Factor w/ 32 levels "Acura","Audi",..: 1 1 2 2 3 4 4 4 4 5
 $ Model             : Factor w/ 93 levels "100","190E","240",..: 49 56 9 1 6 24 5
 $ Type              : Factor w/ 6 levels "Compact","Large",..: 4 3 1 3 3 3 2 2 3
 $ Min.Price         : num  12.9 29.2 25.9 30.8 23.7 14.2 19.9 22.6 26.3 33 ...
 $ Price             : num  15.9 33.9 29.1 37.7 30 15.7 20.8 23.7 26.3 34.7 ...
 $ Max.Price         : num  18.8 38.7 32.3 44.6 36.2 17.3 21.7 24.9 26.3 36.3 ...
 $ MPG.city          : int  25 18 20 19 22 22 19 16 19 16 ...
 $ MPG.highway       : int  31 25 26 26 30 31 28 25 27 25 ...
 $ AirBags           : Factor w/ 3 levels "Driver & Passenger",..: 3 1 2 1 2 2 2 2
 $ DriveTrain        : Factor w/ 3 levels "4WD","Front",..: 2 2 2 2 3 2 2 3 2 2 .
```

```
$ Cylinders         : Factor w/ 6 levels "3","4","5","6",..: 2 4 4 4 2 2 4 4 4 5
$ EngineSize        : num  1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...
$ Horsepower        : int  140 200 172 172 208 110 170 180 170 200 ...
$ RPM               : int  6300 5500 5500 5500 5700 5200 4800 4000 4800 4100 ...
$ Rev.per.mile      : int  2890 2335 2280 2535 2545 2565 1570 1320 1690 1510 ...
$ Man.trans.avail   : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1 1 1 ...
$ Fuel.tank.capacity: num  13.2 18 16.9 21.1 21.1 16.4 18 23 18.8 18 ...
$ Passengers        : int  5 5 5 6 4 6 6 6 5 6 ...
$ Length            : int  177 195 180 193 186 189 200 216 198 206 ...
$ Wheelbase         : int  102 115 102 106 109 105 111 116 108 114 ...
$ Width             : int  68 71 67 70 69 69 74 78 73 73 ...
$ Turn.circle       : int  37 38 37 37 39 41 42 45 41 43 ...
$ Rear.seat.room    : num  26.5 30 28 31 27 28 30.5 30.5 26.5 35 ...
$ Luggage.room      : int  11 15 14 17 13 16 17 21 14 18 ...
$ Weight            : int  2705 3560 3375 3405 3640 2880 3470 4105 3495 3620 ...
$ Origin            : Factor w/ 2 levels "USA","non-USA": 2 2 2 2 2 1 1 1 1 1 ..
$ Make              : Factor w/ 93 levels "Acura Integra",..: 1 2 4 3 5 6 7 9 8 1
```

- You can plot two `numeric` variables and group them by a `factor`:

```
## plot MPG.city vs. Horsepower, conditioned by Cylinders
xyplot(MPG.city ~ Horsepower | Cylinders, data = Cars93)
```



- You can `group` variables and get an automatic legend per group.
- Price to be paid: simple annotations are harder to do than base R.

17 Grammar of graphics with `ggplot2`

- Grammar of graphics construction based on human perception
- Better support for multipanel conditioning plots
- Highly extensible, complex, steep learning curve (see here)

18 Concept summary

- Infographics are design-rich and built to inform, data visualizations (and dashboards) are data-rich and built to be flexible and alterable

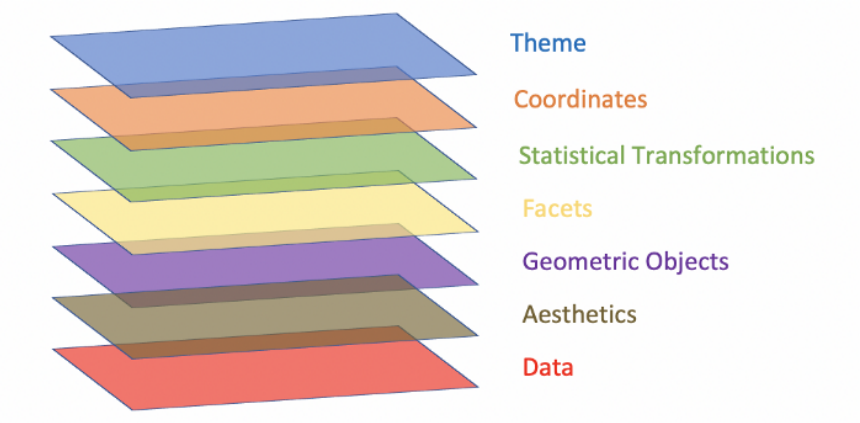


Figure 2: Grammar of Graphics (gg) philosophy

- Exploratory/explanatory graphics have different challenges. EDA: separate signal from noise; storytelling: tell a good story!

19 Code summary

COMMAND	MEANING
`method`	Available methods for generic functions

20 Glossary

TERM	MEANING
Infographics	
Data visualization	
Exploratory graphics	
Explanatory graphics	

21 References

- Arteaga M (20 January, 2021). How to create a visual CV using R!. Online: mmarteaga.github.io.
- CRAN (27 April 2021). Visualize downloads from CRAN Packages. Online: cran.r-project.org.