Presentation of Scientific Results

Arnaud Legrand and Jean-Marc Vincent

Scientific Methodology and Performance Evaluation
M2R MOSIG, Grenoble, September-December 2015

Outline

Outline

Why do we need to visualize ? The Anscombe's Quartet

$X^{(1)}$	$Y^{(1)}$
10.00	8.04
8.00	6.95
13.00	7.58
9.00	8.81
11.00	8.33
14.00	9.96
6.00	7.24
4.00	4.26
12.00	10.24
7.00	4.82
5.00	5.68

$N = 11$ samples
Mean of $X = 9.0$
Mean of $Y = 7.5$

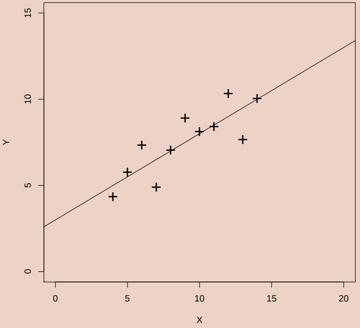
Correlation = 0.816

Why do we need to visualize ? The Anscombe's Quartet

$X^{(1)}$	$Y^{(1)}$
10.00	8.04
8.00	6.95
13.00	7.58
9.00	8.81
11.00	8.33
14.00	9.96
6.00	7.24
4.00	4.26
12.00	10.24
7.00	4.82
5.00	5.68

$N = 11$ samples
Mean of $X = 9.$
Mean of $Y = 7$
Intercept = 3
Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

Scatter plot

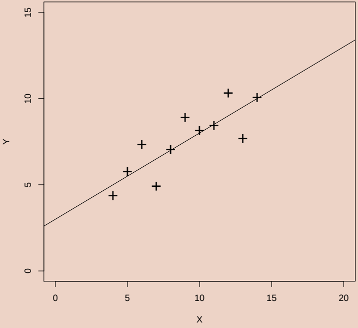


Why do we need to visualize ? The Anscombe's Quartet

$X^{(1)}$	$Y^{(1)}$
10.00	8.04
8.00	6.95
13.00	7.58
9.00	8.81
11.00	8.33
14.00	9.96
6.00	7.24
4.00	4.26
12.00	10.24
7.00	4.82
5.00	5.68

$N = 11$ samples
Mean of X = 9.
Mean of Y = 7
Intercept = 3
Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

Scatter plot



1. The data set "behaves like" a linear curve with some scatter;
2. There is no justification for a more complicated model (e.g., quadratic);
3. There are no outliers;
4. The vertical spread of the data appears to be of equal height irrespective of the X-value; this indicates that the data are equally-precise throughout and so a "regular" (that is, equi-weighted) fit is appropriate.

Why do we need to visualize ? The Anscombe's Quartet

$X^{(1)}$	$Y^{(1)}$
10.00	8.04
8.00	6.95
13.00	7.58
9.00	8.81
11.00	8.33
14.00	9.96
6.00	7.24
4.00	4.26
12.00	10.24
7.00	4.82
5.00	5.68

$N = 11$ samples
Mean of $X = 9.0$
Mean of $Y = 7.5$
Intercept $= 3$
Slope $= 0.5$
Res. stdev $= 1.237$
Correlation $= 0.816$

$X^{(2)}$	$Y^{(2)}$
10.00	9.14
8.00	8.14
13.00	8.74
9.00	8.77
11.00	9.26
14.00	8.10
6.00	6.13
4.00	3.10
12.00	9.13
7.00	7.26
5.00	4.74

$N = 11$ samples
Mean of $X = 9.0$
Mean of $Y = 7.5$
Intercept $= 3$
Slope $= 0.5$
Res. stdev $= 1.237$
Correlation $= 0.816$

$X^{(3)}$	$Y^{(3)}$
10.00	7.46
8.00	6.77
13.00	12.74
9.00	7.11
11.00	7.81
14.00	8.84
6.00	6.08
4.00	5.39
12.00	8.15
7.00	6.42
5.00	5.73

$N = 11$ samples
Mean of $X = 9.0$
Mean of $Y = 7.5$
Intercept $= 3$
Slope $= 0.5$
Res. stdev $= 1.237$
Correlation $= 0.816$

$X^{(4)}$	$Y^{(4)}$
8.00	6.58
8.00	5.76
8.00	7.71
8.00	8.84
8.00	8.47
8.00	7.04
8.00	5.25
19.00	12.50
8.00	5.56
8.00	7.91
8.00	6.89

$N = 11$ samples
Mean of $X = 9.0$
Mean of $Y = 7.5$
Intercept $= 3$
Slope $= 0.5$
Res. stdev $= 1.237$
Correlation $= 0.816$

Why do we need to visualize ? The Anscombe's Quartet

$X^{(1)}$	$Y^{(1)}$
10.00	8.04
8.00	6.95
13.00	7.58
9.00	8.81
11.00	8.33
14.00	9.96
6.00	7.24
4.00	4.26
12.00	10.24
7.00	4.82
5.00	5.68

$N = 11$ samples
Mean of $X = 9.$
Mean of $Y = 7$
Intercept = 3
Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

$X^{(2)}$	$Y^{(2)}$

Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

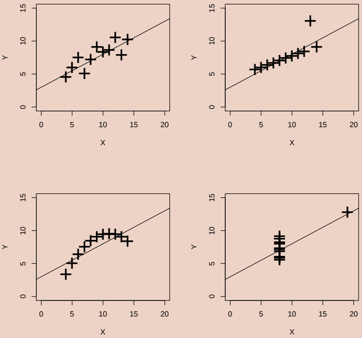
$X^{(3)}$	$Y^{(3)}$

Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

$X^{(4)}$	$Y^{(4)}$

Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

Scatter plot



Why do we need to visualize ? The Anscombe's Quartet

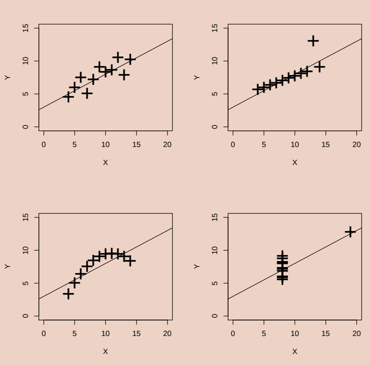
$X^{(1)}$	$Y^{(1)}$
10.00	8.04
8.00	6.95
13.00	7.58
9.00	8.81
11.00	8.33
14.00	9.96
6.00	7.24
4.00	4.26
12.00	10.24
7.00	4.82
5.00	5.68

$X^{(2)}$	$Y^{(2)}$

$X^{(3)}$	$Y^{(3)}$

$X^{(4)}$	$Y^{(4)}$

Scatter plot



1. data set 1 is clearly linear with some scatter.
2. data set 2 is clearly quadratic.
3. data set 3 clearly has an outlier.
4. data set 4 is obviously the victim of a poor experimental design with a single point far removed from the bulk of the data "wagging the dog".

$N = 11$ samples
Mean of $X = 9.$
Mean of $Y = 7$
Intercept = 3
Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

Slope = 0.5
Res. stdev = 1.237
Correlation = 0.816

Problem statement

- All analysis we perform rely on (sometimes implicit) assumptions. If these assumptions do not hold, the analysis will be a complete non-sense.
- Checking these assumptions is not always easy and sometimes, it may even be difficult to list all these assumptions and formally state them.

A visualization can help to check these assumptions.

- Visual representation resort to our cognitive faculties to check properties.
  The visualization is meant to let us detect expected and unexpected behavior with respect to a given model.

Using the "right" representations

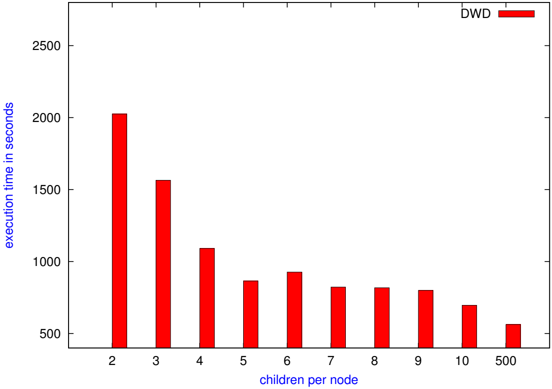
- The problem is to represent on a limited space, typically a screen with a fixed resolution, a meaningful information about the behavior of an application or system.
- $\leadsto$ need to aggregate data and be aware of what information loss this incurs.
- Every visualization emphasizes some characteristics and hides others. Being aware of the underlying models helps choosing the right representation.

Visualization and intuition

- Visualization can also be used to guide your intuition.
  Sometimes, you do not know exactly what you are looking for and looking at the data just helps.
- Some techniques (Exploratory Data Analysis) even build on this and propose to summarize main characteristics in easy-to-understand form, often with visual graphs, without using a statistical model or having formulated a hypothesis.
- **Use with care**, visualizations always have underlying models: when visualization is not adapted, what you may observe may be meaningless. Such approaches may help formulating hypothesis but these hypothesis have then to be tested upon new data-sets.

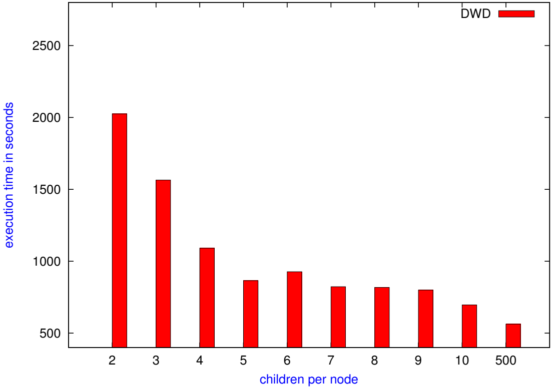
A "simple" graphical check for investigating scalability

Plotting T_p versus p.



A "simple" graphical check for investigating scalability

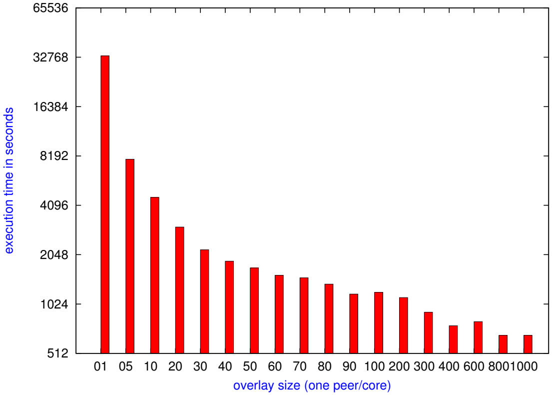
Plotting T_p versus p.



- y-axis does not start at 0, which makes speedup look more impressive
- x-axis is linear with an outlier.

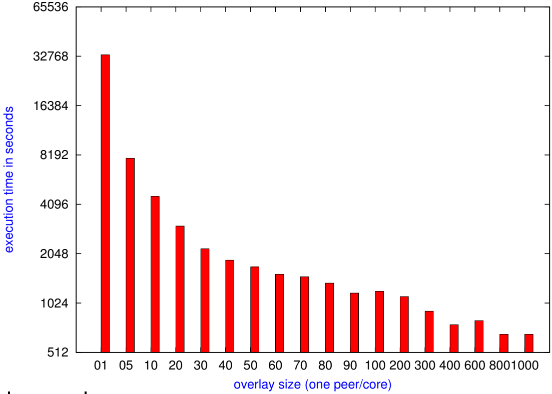
A "simple" graphical check for investigating scalability

Plotting T_p versus p.



A "simple" graphical check for investigating scalability

Plotting T_p versus p.

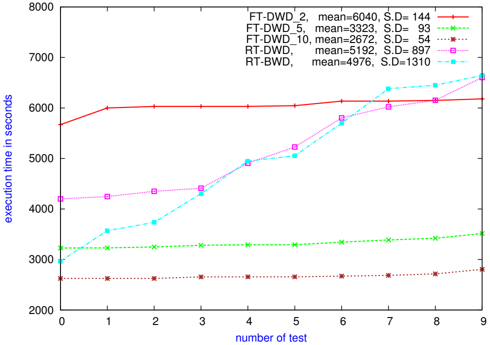


- y-axis uses log-scale
- x-axis is neither linear nor logarithmic so we cannot reason about the shape of the curve

Say, we want to test for Amhdal's law. Propose a better representation.

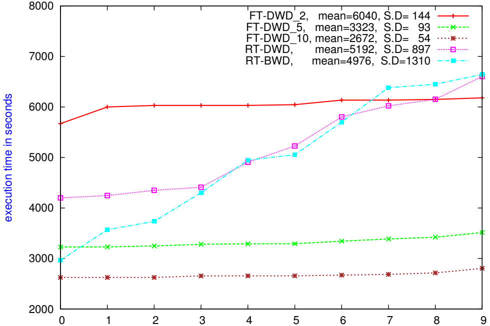
Graphically checking which alternative is better ?

5 different alternatives (`FT-DWD_2`, `FT-DWD_5`, `FT-DWD_10`, `RT-DWD`, `RT-BWD`), each tested 10 times.



Graphically checking which alternative is better ?

5 different alternatives (`FT-DWD_2`, `FT-DWD_5`, `FT-DWD_10`, `RT-DWD`, `RT-BWD`), each tested 10 times.



Outcomes have been sorted by increasing value for each alternative and are then linked together

- The shape of the lines do not make any sense. The lines group related values
- Experiment order does not make any sense and makes it look like alternatives have been evaluated in 10 different settings (, which suggests the values can be compared with each others for each setting)

Propose a better representation

Outline

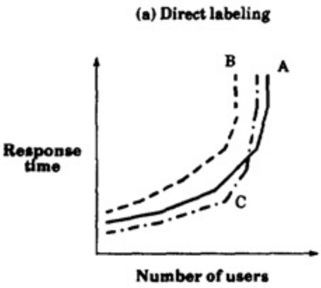
1. Data Visualization
   - Motivation
   - Jain, Chapter 10
2. Needful R Packages by Hadley Wickam
   - Plyr And Dplyr
   - Ggplot2
   - Reshape and tydiR
   - Now let's play!
   - Conclusion
3. Descriptive statistics of an univariate sample
   - Motivation
   - Initial step
   - Histograms of "Stable" samples
   - Single mode: central tendency
   - Dispersion: Variability around the central tendency
   - Going further
   - Summarizing a distribution

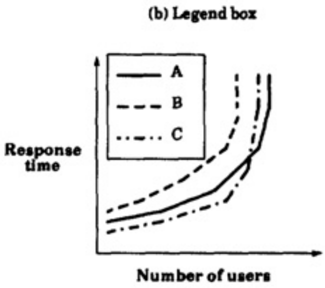
Read the basics

- For all such kind of "general" graphs where you summarize the results of several experiments, the very least you need to read is Jain's book: The Art of Computer Systems Performance Analysis. A new edition is expected in sept. 2015
- It has check lists for "Good graphics", which I made more or less available on the lecture's webpage
- It presents the most common pitfalls in data representation
- It will teach how to cheat with your figures. . .
- . . . and how to detect cheaters. ;)

Guidelines

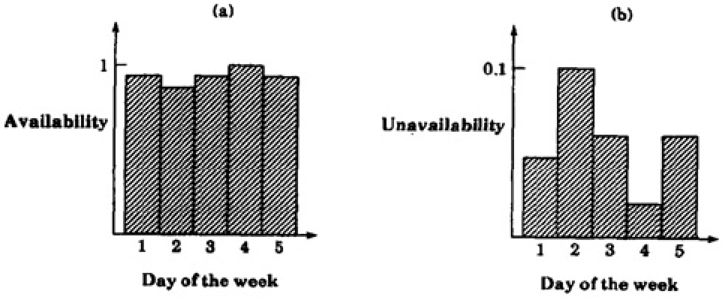
1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)





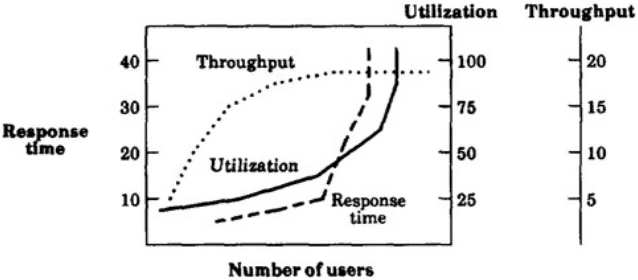
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



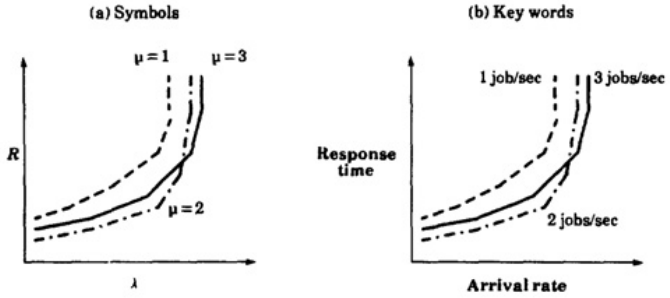
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



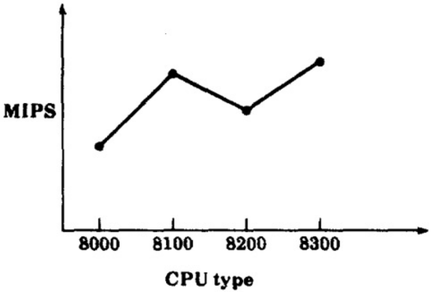
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



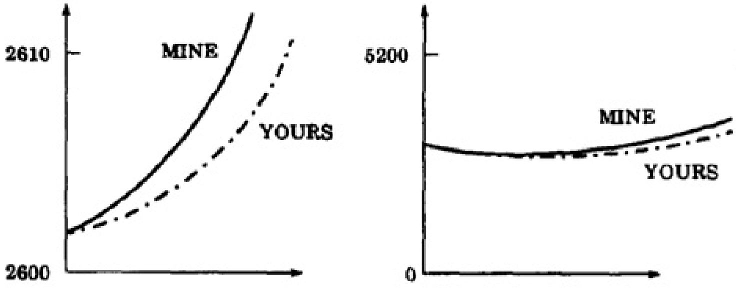
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



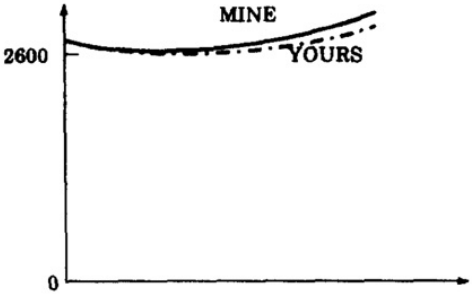
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



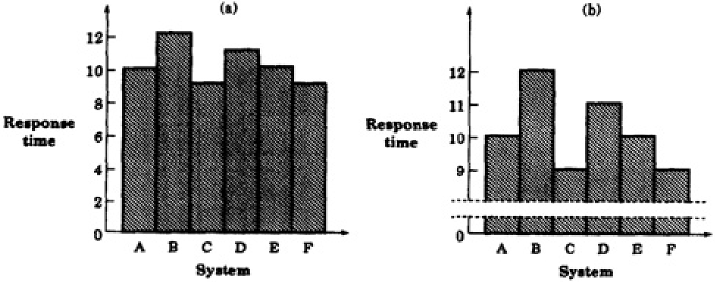
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



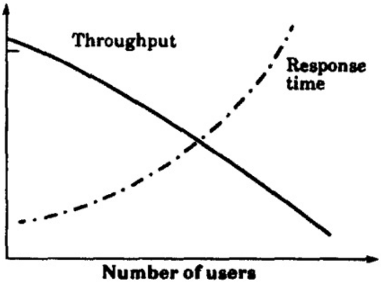
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



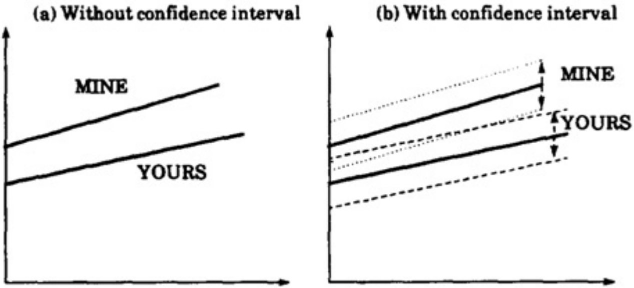
Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)

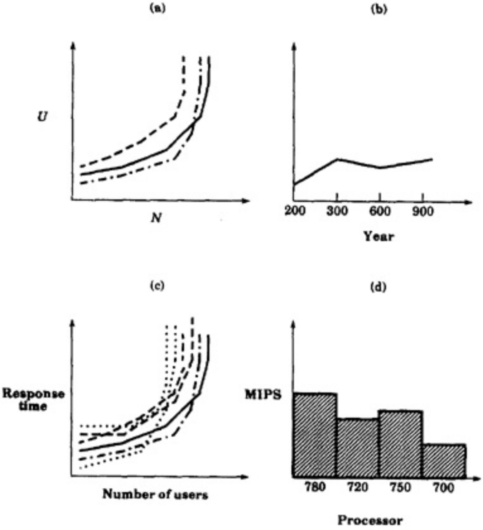


Guidelines

1. Require minimum effort to the reader: get the message (legends, labels, trends, annotations, . . .)
2. Maximize information (self-sufficient, clear labels, units, . . .)
3. Minimize Ink (avoid cluttered information. . .)
4. Use commonly accepted practices (effect along the y-axis, scales)
5. Avoid Ambiguity (coordinates, scales, colors, only one variable, . . .)



What about these ones ?



Use the right tools

R is a system for statistical computation and graphics.

- Avoid programming with R. Most things can be done with one liners.
- Excellent graphic support with **ggplot2**.
- `knitr` allows to mix R with LaTeX or Markdown. Literate programming to ease reproducible research.

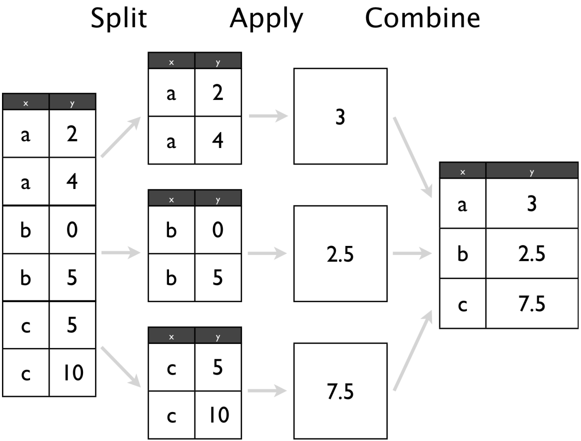
Rstudio is an IDE a system for statistical computation and graphics. It is easy to use and allows publishing on **rpubs**.

Org-mode Allows to mix sh, perl, R, ... within plain text documents and export to LaTeX, HTML, ...

Outline

plyr: the Split-Apply-Combine Strategy

Have a look at http://plyr.had.co.nz/09-user/ for a more detailed introduction.



plyr: Powerful One-liners

```r
library(plyr)
mtcars_summarized = ddply(mtcars,c("cyl","carb"), summarize,
      num = length(wt), wt_mean = mean(wt), wt_sd = sd(wt),
      qsec_mean = mean(qsec), qsec_sd = sd(qsec));
mtcars_summarized
```

```
  cyl carb num  wt_mean     wt_sd qsec_mean   qsec_sd
1   4    1   5 2.151000 0.2627118  19.37800 0.6121029
2   4    2   6 2.398000 0.7485412  18.93667 2.2924368
3   6    1   2 3.337500 0.1732412  19.83000 0.5515433
4   6    4   4 3.093750 0.4131460  17.67000 1.1249296
5   6    6   1 2.770000        NA  15.50000        NA
6   8    2   4 3.560000 0.1939502  17.06000 0.1783255
7   8    3   3 3.860000 0.1835756  17.66667 0.3055050
8   8    4   6 4.433167 1.0171431  16.49500 1.4424112
9   8    8   1 3.570000        NA  14.60000        NA
```

plyr next generation = dplyr

It's much much faster and more readable. The *tutorial* is great...

```
library(dplyr)
mtcars %>% group_by(cyl,carb) %>%
   select(wt,qsec) %>%
   summarise(num = n(),
       wt_mean = mean(wt), wt_sd = sd(wt),
       qsec_mean = mean(qsec), qsec_sd = sd(qsec)) %>%
   filter(num>=1)
```

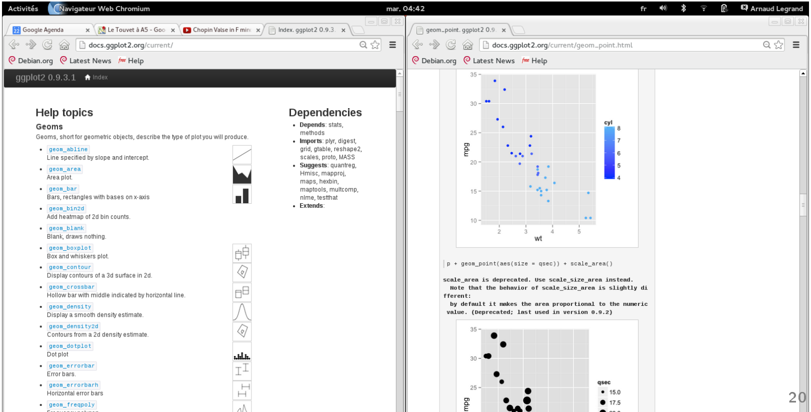
```
 Source: local data frame [9 x 7]
Groups: cyl

  cyl carb num  wt_mean     wt_sd qsec_mean   qsec_sd
1   4    1   5 2.151000 0.2627118  19.37800 0.6121029
2   4    2   6 2.398000 0.7485412  18.93667 2.2924368
3   6    1   2 3.337500 0.1732412  19.83000 0.5515433
```

Outline

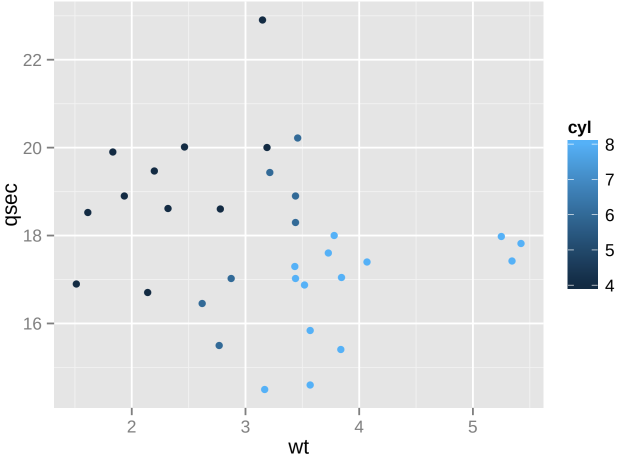
ggplot2: Modularity in Action

- ggplot2 builds on plyr and on a modular grammar of graphics
- ~~obnoxious function with dozens of arguments~~
- combine small functions using layers and transformations
- aesthetic mapping between observation characteristics (data frame column names) and graphical object variables
- an incredible documentation: http://docs.ggplot2.org/current/



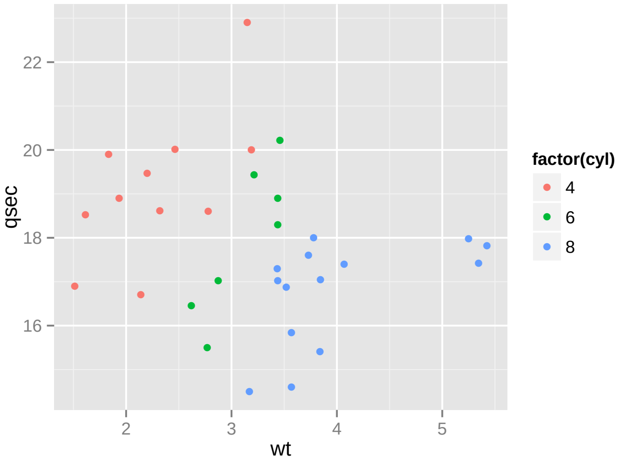
ggplot2: Illustration (1)

```
ggplot(data = mtcars, aes(x=wt, y=qsec, color=cyl)) +
        geom_point();
```



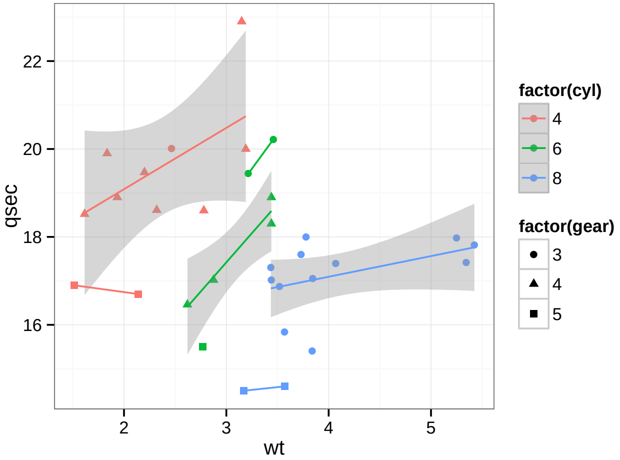
ggplot2: Illustration (2)

```
ggplot(data = mtcars, aes(x=wt, y=qsec, color=factor(cyl))) +
       geom_point();
```



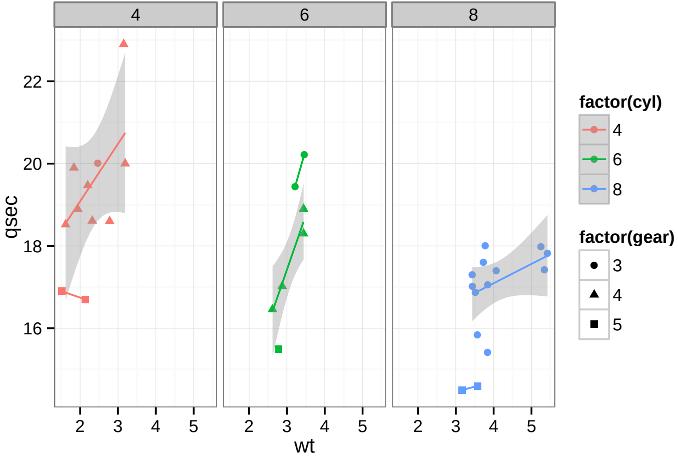
ggplot2: Illustration (3)

```
ggplot(data = mtcars, aes(x=wt, y=qsec, color=factor(cyl),
        shape = factor(gear))) +  geom_point() + theme_bw() +
        geom_smooth(method="lm");
```



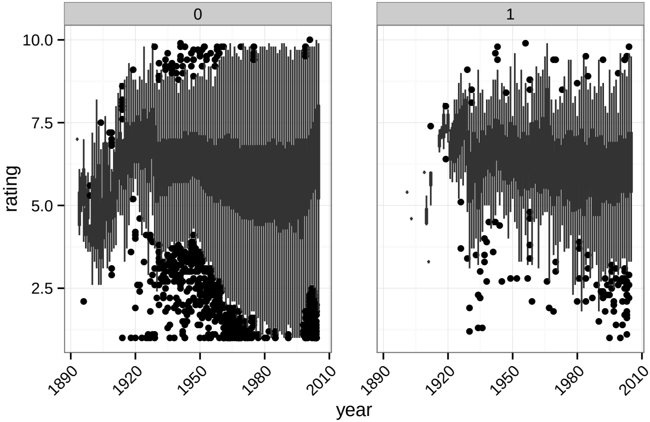
ggplot2: Illustration (4)

```
ggplot(data = mtcars, aes(x=wt, y=qsec, color=factor(cyl),
        shape = factor(gear))) + geom_point() + theme_bw() +
        geom_smooth(method="lm") + facet_wrap(~ cyl);
```



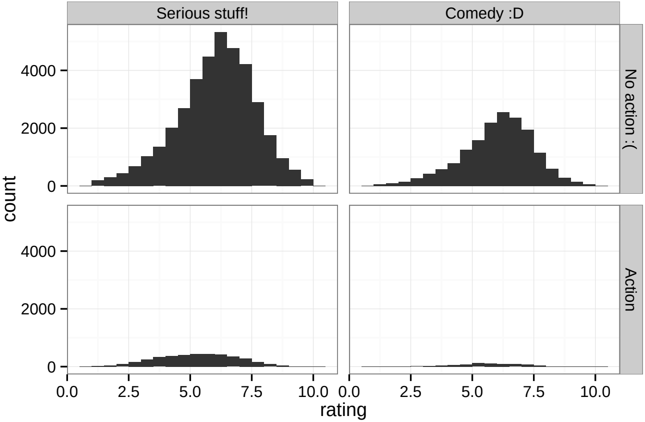
ggplot2: Illustration (5)

```r
ggplot(data = movies, aes(x=year,y=rating,group=factor(year))) +
        geom_boxplot() + facet_wrap(~Romance) + theme_bw() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1),
      panel.margin = unit(2, "lines"));
```



ggplot2: Illustration (6)

```
ggplot(movies, aes(x = rating)) + geom_histogram(binwidth = 0.5)+
        facet_grid(Action ~ Comedy, labeller=mf_labeller) +
        theme_bw() + theme(panel.margin = unit(.5, "lines"));
```



Outline

"Messy" data

When using ggplot or plyr, your data may not in the right shape, in which case you should give a try to reshape/melt

```
messy <- data.frame(
  name = c("Wilbur", "Petunia", "Gregory"),
  a = c(67, 80, 64),
  b = c(56, 90, 50)
)
messy
```

```
     name  a  b
1  Wilbur 67 56
2 Petunia 80 90
3 Gregory 64 50
```

- a and b are two different types of drugs and the values correspond to heart rate
- ggplot faceting or coloring based on the drug type is a pain
- we need a way to make "wide" data longer

Reshape

```
library(reshape)
cleaner = melt(messy,c("name"))
names(cleaner)=c("name","drug","heartrate")
cleaner
```

```
     name drug heartrate
1  Wilbur    a        67
2 Petunia    a        80
3 Gregory    a        64
4  Wilbur    b        56
5 Petunia    b        90
6 Gregory    b        50
```

Tidyr

Just like plyr, reshape is a little magical. tidyr is the new generation (faster, more coherent). Again, the *tutorial* is great.

```
library(tidyr)
library(dplyr)
messy %>% gather(drug, heartrate, -name)
```

```
     name drug heartrate
1  Wilbur    a        67
2 Petunia    a        80
3 Gregory    a        64
4  Wilbur    b        56
5 Petunia    b        90
6 Gregory    b        50
```

Hint: Avoid mixing old-generation with new-generation as it overrides some function names and leads to weird behaviors

Outline

Summarizing information

You may like these cheat sheets:

https://www.rstudio.com/resources/cheatsheets/

```
df = read.csv("data/set1.csv",header=T)
# Alternatively: read.csv("https://raw.githubusercontent.com/
#            alegrand/SMPE/master/lectures/data/set1.csv")
head(df,n=2)
```

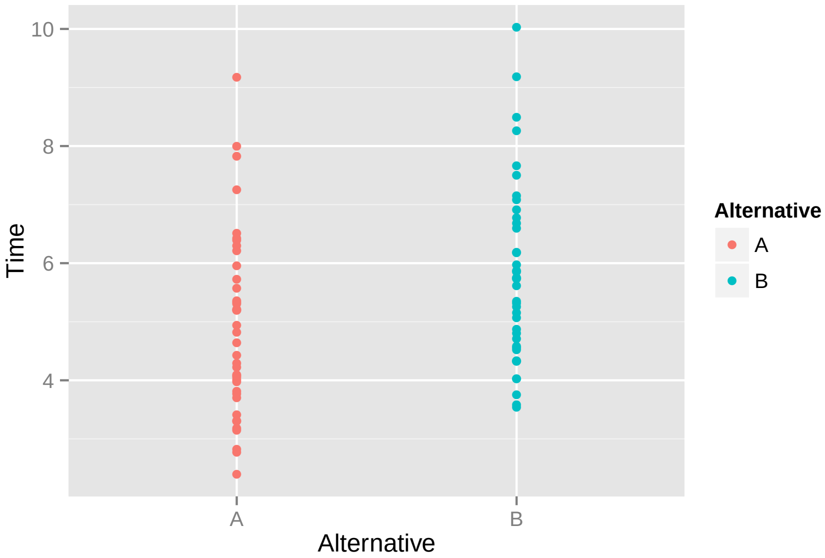
```
         A        B
1 7.256717 8.261171
2 3.813100 4.335301
```

Get the following summary using plyr/reshape or dplyr/tydir:

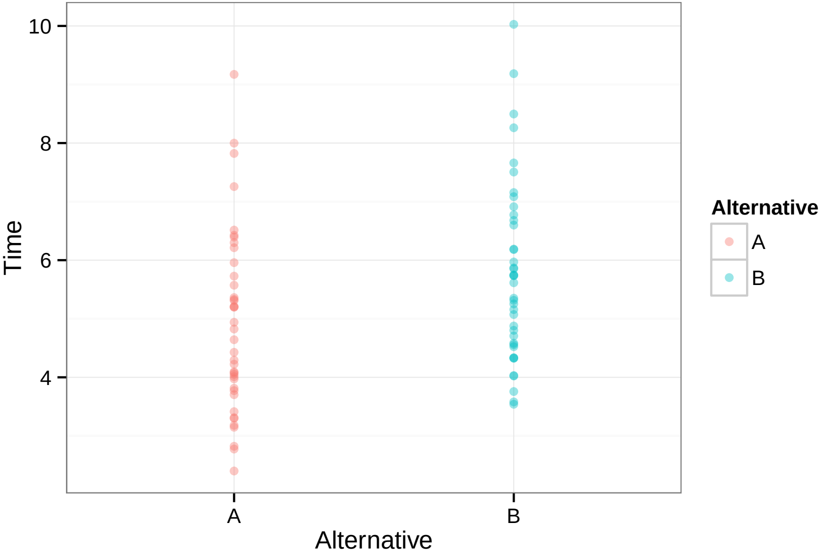
```
Source: local data frame [2 x 6]

  Alternative num     mean       sd      min       max
1           A  40 4.903817 1.544423 2.400016  9.172525
2           B  40 5.783643 1.542987 3.539874 10.027147
```

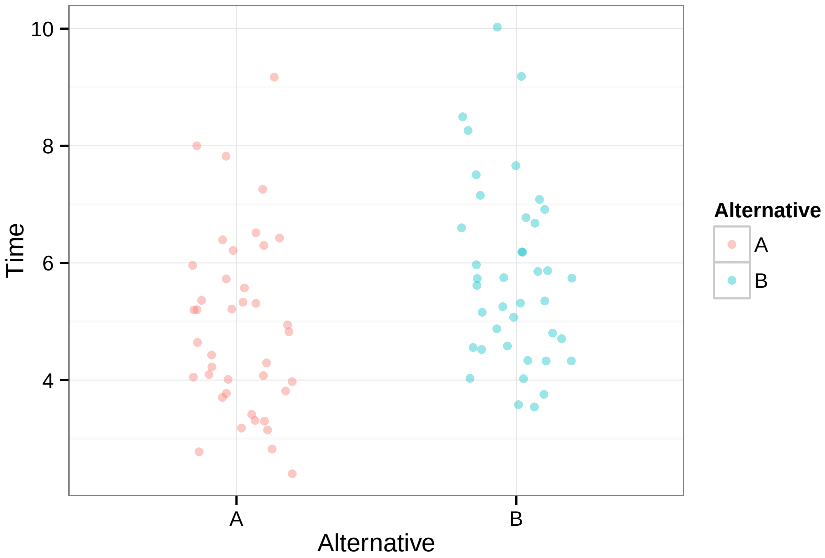
Plot the data



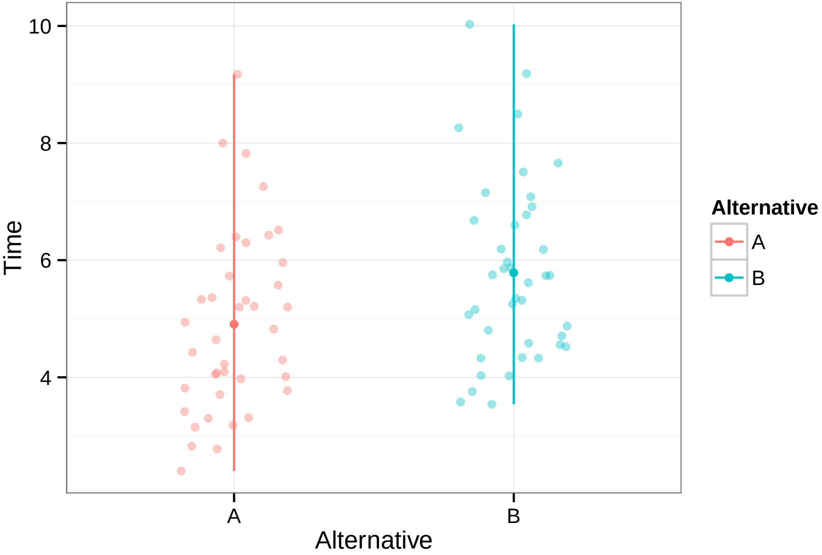
Alleviate over-plotting



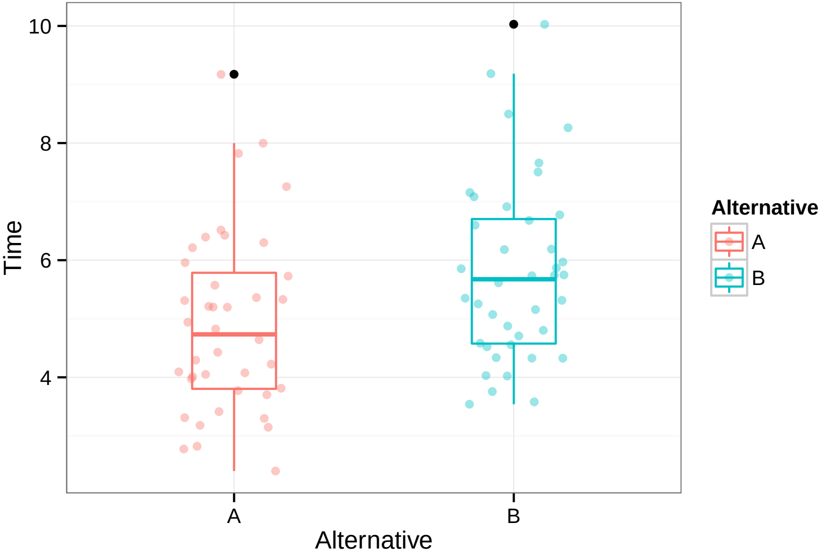
Avoid over-plotting



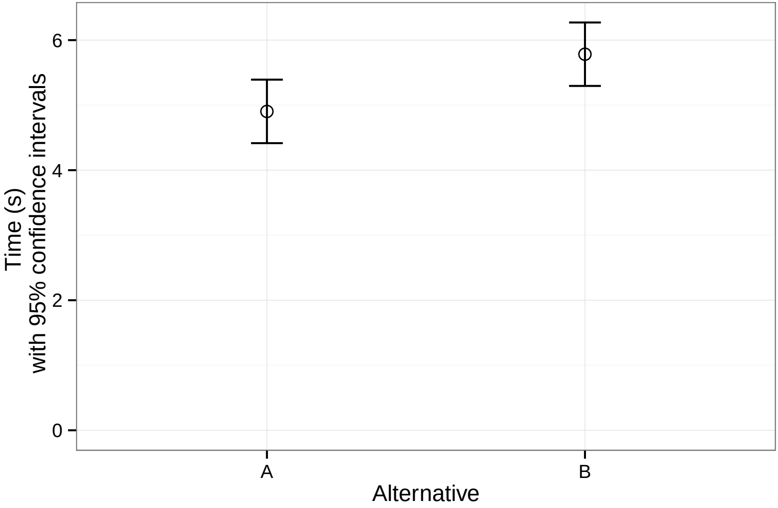
Add summary information



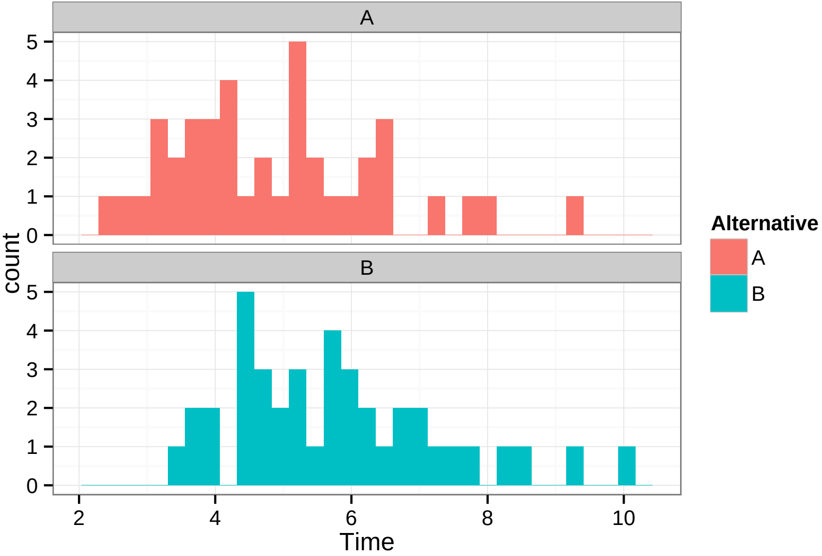
Add more standard summaries



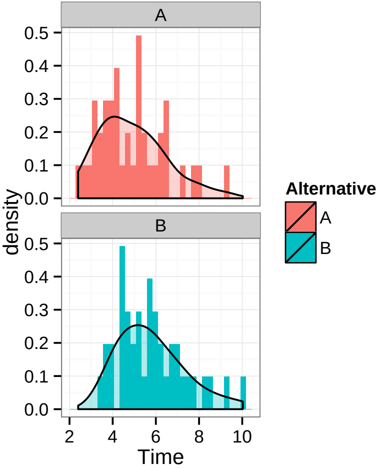
Or depict confidence intervals

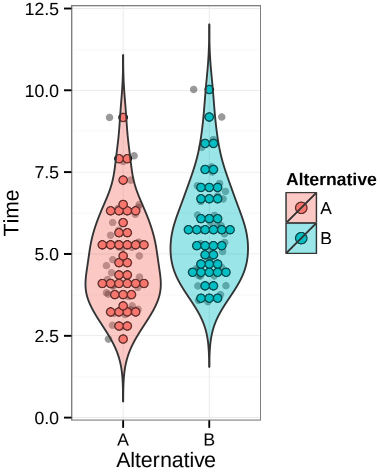


Or use histograms. . .



Be careful with fancy plots you do not fully understand!





Outline

Take away Message

- R, ggplot and other such tools are incredibly powerful for presenting data. They are much more high level than any other tools I have seen so far.
- Mastering it will save you a lot of time as it will allow to look at your data through different angles and thus check many hypothesis and present them in the best possible way
- Read at least Jain's book: The Art of Computer Systems Performance Analysis
- However, you may have started understanding that a visualization is meant to check or to illustrate one particular aspect and that this "aspect" relies on a mathematical model. I will thus explain you in the next lecture what this model is.

To do for the Next Time: Use what you just learned to improve your data analysis, the article you're currently writing, ...

Outline

Motivation

We have set up a world where we keep collecting data, huge amount of data...
Sweet, what knowledge can we exctract from such data? How do we summarize a data set?
With a few numbers, some graphics? How? Why is this difficult?

> *There are three kinds of lies: lies, damned lies and statistics*
>
> – Mark Twain's Autobiography
>
> *Statistical thinking will one day be as necessary for efficient citizenship as the ability to read or write*
>
> – Attributed to H. G. Wells
>
> *The only statistics you can trust are those you falsified yourself*
>
> – Winston Churchill

Outline

I just got new Tees!

- A series of measurements (one value per measurement)
- Nature of the measurements
  - Factors (nominal data)

```
 [1] Red   Red   Black Green Blue  Black White Black Blue
[10] White Black White Red   Black Black Red   Red   Black
[19] Black Black
Levels: Black Blue Green Red White
```

  - Ordered factors (ordinal data)

```
 [1] XL M  S  XL M  M  M  XL M  L  M  L  M  M  M  L  M
[18] M  XL M
Levels: S < M < L < XL
```

  - Numbers (e.g., price, duration, . . .) (numerical data)

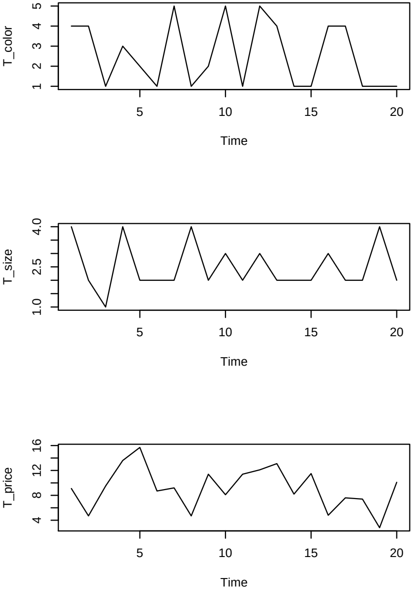
```
 [1]  9.1  4.7  9.5 13.6 15.7  8.7  9.2  4.7 11.4  8.1
[11] 11.4 12.1 13.1  8.2 11.5  4.8  7.6  7.4  2.8 10.1
```

```
str(T_size); # May want to use the str function
```

```
Ord.factor w/ 4 levels "S"<"M"<"L"<"XL": 4 2 1 4 2 2 2 4 2 3
```

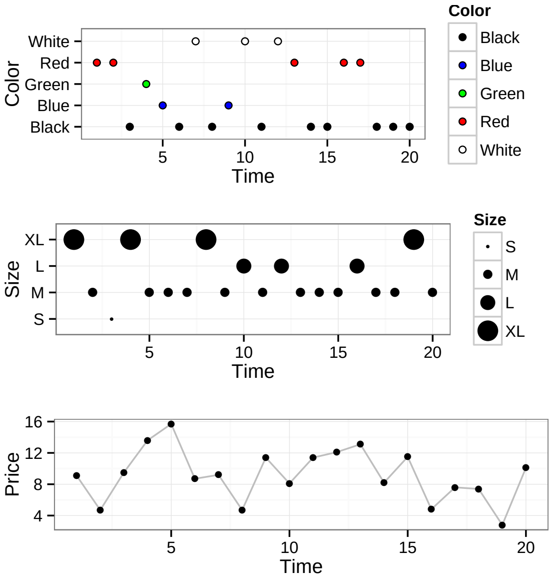
Use `plot.ts` (for time series)

```
par(mfrow=c(3,1));
plot.ts(T_color,xy.lines=F);
plot.ts(T_size,xy.lines=F);
plot.ts(T_price,xy.lines=F);
```

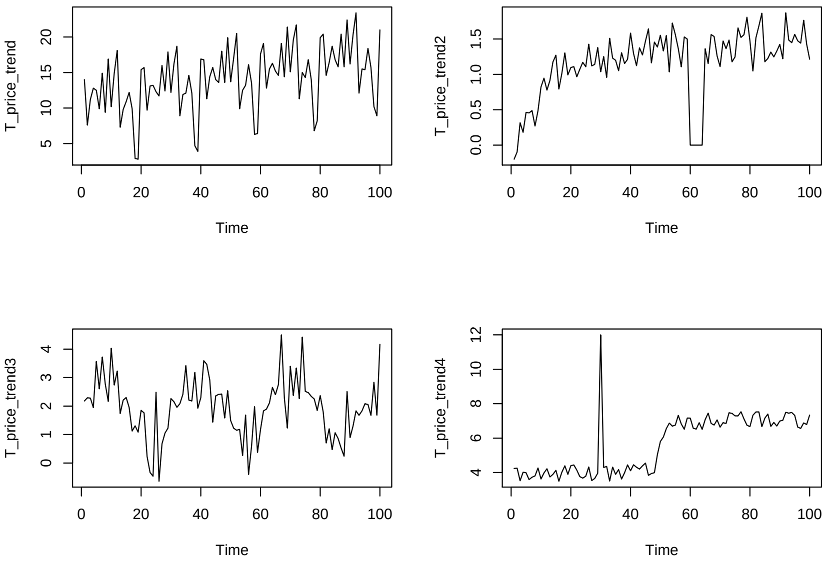


Are these sample "structured"?

Fancier output can be built using ggplot2



There could indeed be "trends"



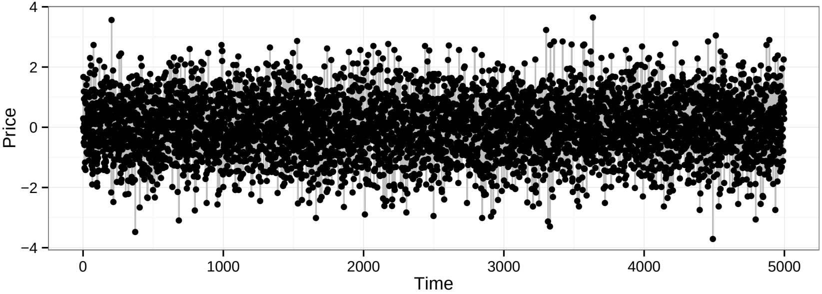
What should we look for?

- Structured/unstructured
- Trend, evolution
- Localization/order of magnitude
- Outliers, aberrant values

This preliminary study will:

- guide your analysis
- provide feedback on your experimental setup

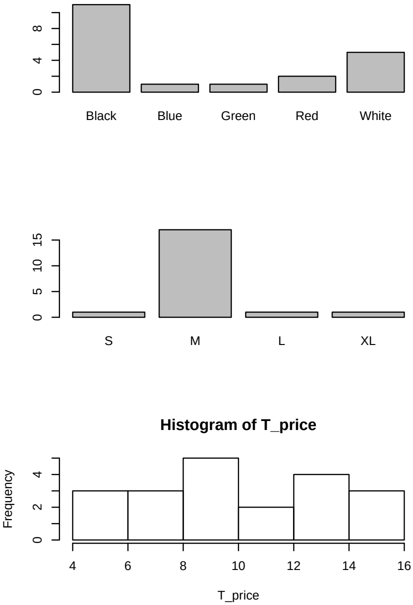
This may be harder to do than it looks...



Outline

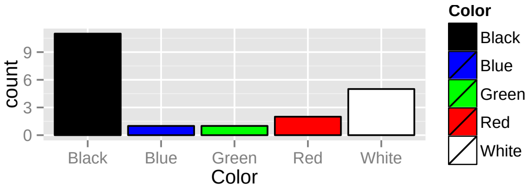
Bar charts vs. Histograms

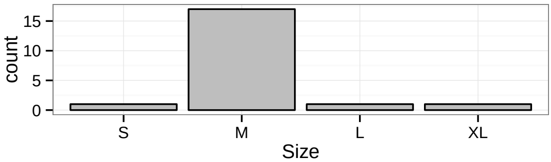
```
par(mfrow=c(3,1));
plot(T_color,xy.lines=F);
plot(T_size,xy.lines=F);
hist(T_price,xy.lines=F);
```

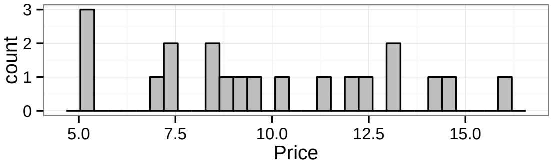


Bar charts vs. Histograms

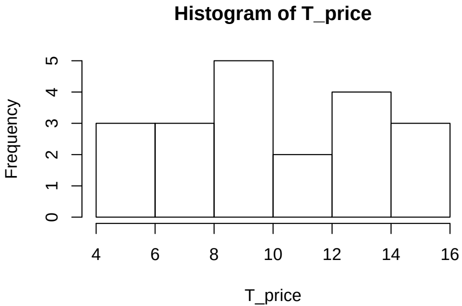
Again, fancier output can be built using ggplot2

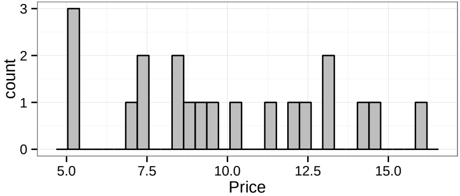


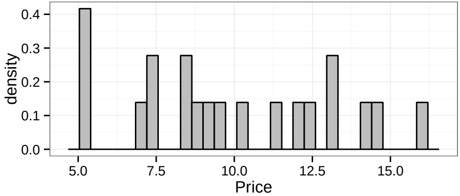


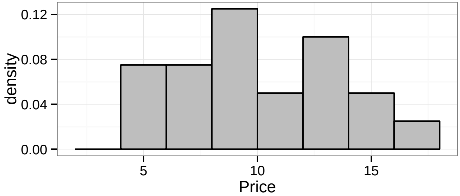


Wait, why are these histograms so different?









Beware of histograms

Rather indicate density than count

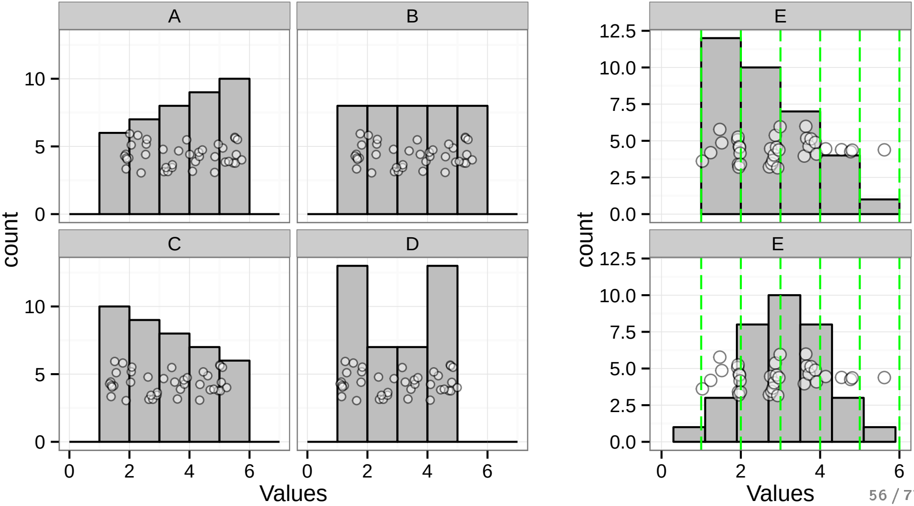
How many bins? Which binwidth?

- `ggplot` defaults to $k = 30$ bins of width $h =$ `range/30` ☹
- Square-root choice: $k = \sqrt{n}$ (Excel, ☹)
- Sturges: $k = \lceil \log_2 n + 1 \rceil$ (default for `hist` in R)
- Rice: $k = \lceil 2n^{1/3} \rceil$
- Scott: $k = \left\lceil \frac{\max x - \min x}{h} \right\rceil$, where: $h = \frac{3.5\hat{\sigma}}{n^{1/3}}$ (equivalent to Rice under some conditions)
- ...

Beware of Histograms

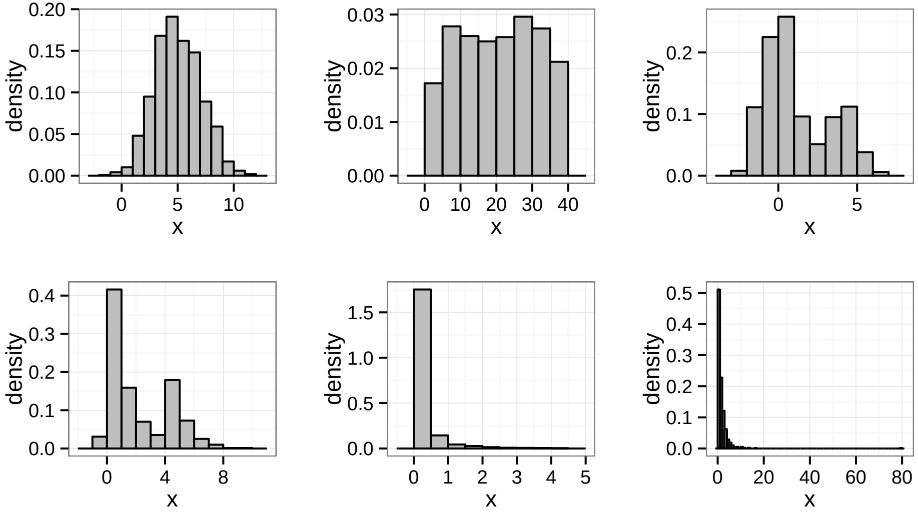
At which value should the bin start?

- In most cases, the binning is aligned on human readable values, which can create nasty artifacts (nice illustration from *stackexchange*)



What should we look for?

Shape: flat? symmetrical? multi-modal? Play with `binwidth` (and `origin` if you have few samples) to uncover the full story behind your data...

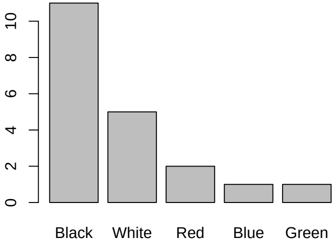


Outline

- What is the mode (most frequent value)?
- Sort values according to their frequency. . .

```
summary(T_color)
```

```
Black  Blue Green   Red White
   11     1     1     2     5
```



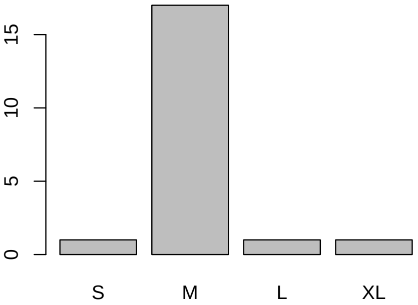

```
col_freq=table(T_color);
T_color <- factor(T_color,
    levels = names(col_freq[order(col_freq, decreasing = TRUE)]));
plot(T_color);
```

Ordinal Values

- What is the mode (most frequent value)?

```
summary(T_size)
```

```
S   M   L  XL
1  17   1   1
```



- May still want to sort values according to their frequency...
- Median: not implemented in standard R for ordinal values, as it's not well defined

```
median(T_size)
library(DescTools)
median(T_size) # :(
```

```
Error in median.default(T_size) : requires numerical data
[1] NA
```

Numerical Values

```
str(T_price);
```

```
num [1:20] 14.5 13.1 9.3 6.9 8.6 7.2 7.3 12.4 13.1 16 ...
```

```
summary(T_price);
```

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
5.200   7.275   9.500   9.960  12.580  16.000
```

- min, max, median in R
- Median: 50% of values are smaller than 9.5
  (a possible measure of central tendency)

Numerical Values

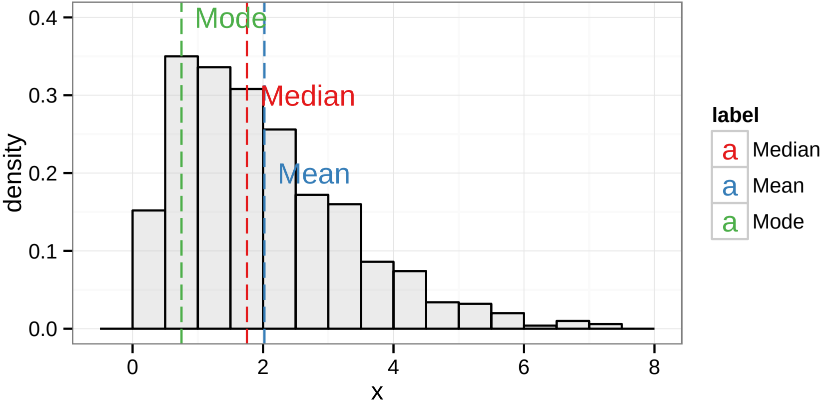
The mode and the median are measures of central tendency (typical value)

- Note: There may be several modes and it depends on binning...

There is also the (arithmetic) mean: $A = \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

```
mean(T_price)
```

```
[1] 9.96
```



Things to know about the mean

- This measure is sensitive to "outliers".
  - One aberrant (say very large) value will drag the mean to the right while it would not change the median
- The key question is what makes sense?
  - Your favorite pair has been added a +20% mark-up in August but you have a -20% discount as a regular customer. Is the price the same?
    - No, you actually saved 4% of the original price ($1.2 \times .8 = .96$).
  - You drove half the way at 50mph and half of the way at 100mph. Did you drive on average at 75mph?
    - Obviously not...
  - Although you can compute the average of gains/loss, it is not at all what you would consider as the average gain.
  - May want to consider the geometric or the harmonic mean...

$$G = \sqrt[n]{\prod_{i=1}^{N} x_i} \text{ or } H = \frac{1}{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{x_i}}$$

What should I look for?

- If the distribution is unimodal and symmetrical, then

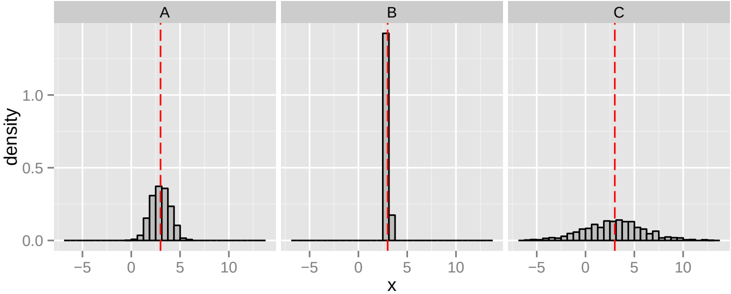
$$\text{mean} = \text{mode} = \text{median}$$

- Depending on the problem, one or the other may be more relevant
- Anyway, reporting such measure with no indication about variability is generally useless

Outline

Variance

We expect most values to be "around" the mean



Departure from the mean:

- Mean absolute deviation: $\frac{1}{N}\sum_{i=1}^{N}|x_i - A|$
  - Rarely used
- Variance: $V = \frac{1}{N}\sum_{i=1}^{N}(x_i - A)^2$
  - only positive values and gives more importance to large deviations ☺
  - not homogeneous to the mean (units) ☹
- Standard deviation: $SD = \sqrt{V}$

Quantile

```r
quantile(T_price,c(.05,.25,.5,.75,.95))
```

```
   5%    25%    50%    75%    95%
4.605  7.550  9.150 11.425 13.705
```

Inter-Quantile Range:

- Inter-quartile range: $IQR = Q_{75} - Q_{25}$
- But other values are possible, e.g., $Q_{95} - Q_5$
- Range: max − min (may grow unbounded)
  - $\rightsquigarrow$ quite difficult to use

What about nominal or ordinal values?

There is for example the notion of Entropy: how many bits are required to encode the sample?
Say there is a fraction f_v of items with value v.

$$H = -\sum_{v \in V} f_v \log_2(f_v)$$

$-(x+y)\log_2(x+y) < -x\log_2(x) - y\log_2(y)$ so the smaller the entropy, the more condensed/predictable the sample distribution

- $H([0, 1, 0, 0]) = 0$
- $H([.25, .25, .25, .25]) = 2$
- $H([1/n, \ldots, 1/n]) = \log_2(n)$ so you generally normalize H by $\log_2(n)$

This notion can be extended to numerical values (but the computation is complex as it depends on the binning...)

Outline

Skewness

Remember the mean and the variance:

- $A = \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
- $V = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2$

Could we measure the asymmetry of the samples around the mean?

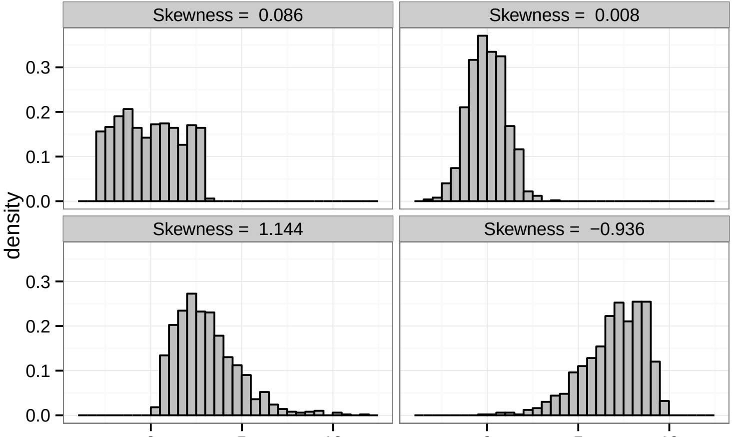
- Proposal 1: $\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})$ (always 0... ☹)
- Proposal 2: $\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^3$ (not well normalized... ☹)

$$S = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{\underbrace{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right]}_{\text{variance}}^{3/2}}$$

Could we illustrate this a bit?

```
library(moments)
skewness(runif(1000))
```

```
[1] 0.04626483
```



Kurtosis

- peakedness (width of peak), tail weight, lack of shoulders...
- measure infrequent extreme deviations, as opposed to frequent modestly sized deviations

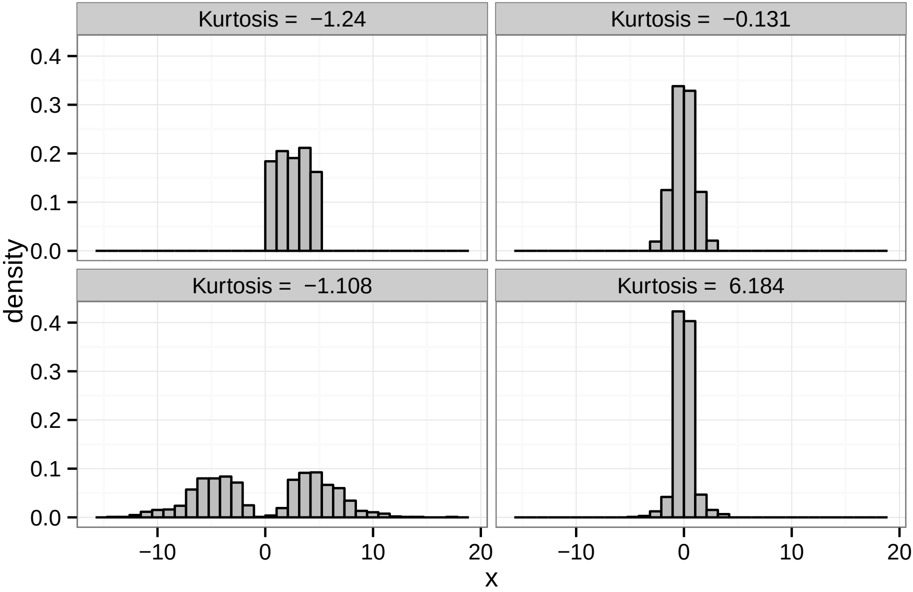
$$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^4}{\underbrace{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right]}_{\text{variance}}^2} - 3$$

The -3 is here so that normal distribution have a Kurtosis of 0

```r
library(moments)
x = rnorm(1000) ; var(x);
kurtosis(x)-3
```

```
[1] 1.039743
[1] 0.01825114
```

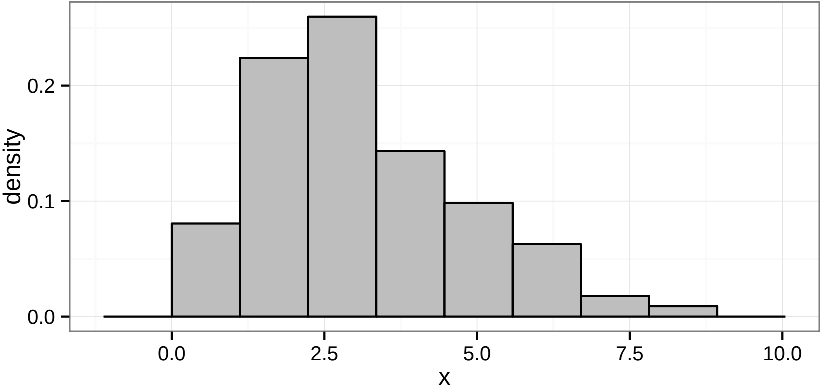
Kurtosis



Kurtosis = −1.24
Kurtosis = −0.131
Kurtosis = −1.108
Kurtosis = 6.184
density
0.0
0.1
0.2
0.3
0.4
−10
0
10
20
x

Outline

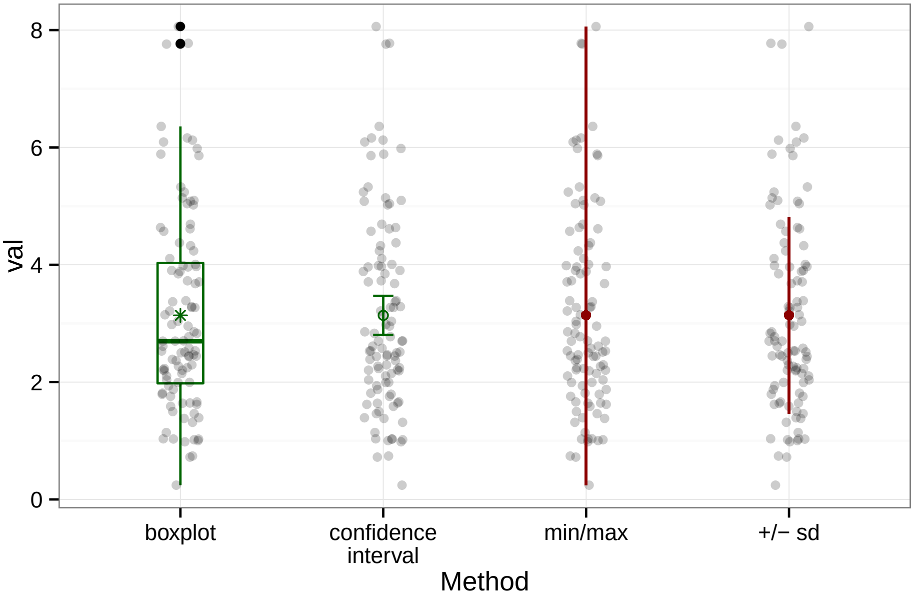
Classical information



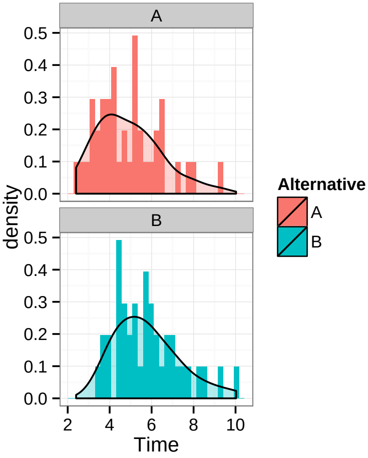

```
summary(x)
var(x)
```

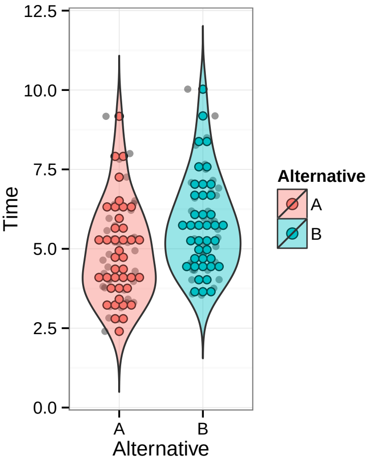
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4065  1.8430  2.5020  2.8660  3.6310  7.0220
[1] 2.117541
```

Good and bad summaries



Be careful with fancy plots you do not fully understand!

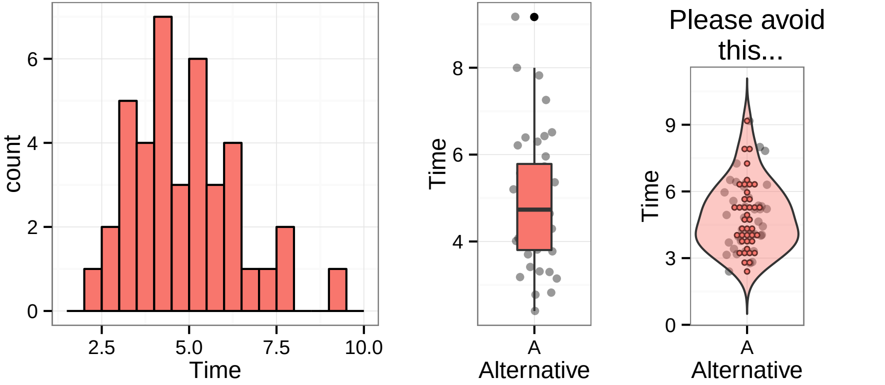




The average human has one breast and one testicle

– Des McHale

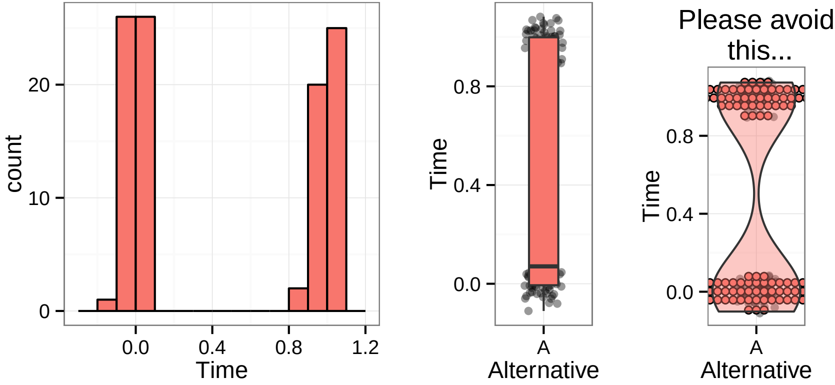
Be careful with fancy plots you do not fully understand!



The average human has one breast and one testicle

– Des McHale

Be careful with fancy plots you do not fully understand!



The average human has one breast and one testicle

– Des McHale