

Presentation of Scientific Results

Arnaud Legrand

Performance Evaluation Lecture
UFRGS, Porto Alegre, August 2015

① Data Visualization

Motivation

Jain, Chapter 10

② Needful R Packages by Hadley Wickam

Plyr And Dplyr

Ggplot2

Reshape and tidyR

Now let's play!

Conclusion

Outline

① Data Visualization

Motivation

Jain, Chapter 10

② Needful R Packages by Hadley Wickam

Plyr And Dplyr

Ggplot2

Reshape and tidyR

Now let's play!

Conclusion

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 $X = 9.0$

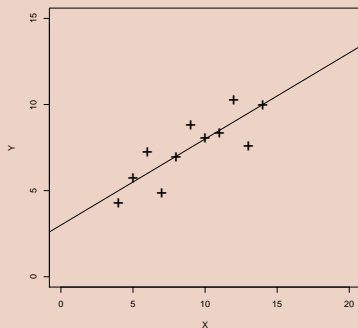
Mean of $Y = 7.5$

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

Scatter plot



$N = 11$ samples

Mean of $X = 9$

Mean of $Y = 7$

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 $Y = 7$

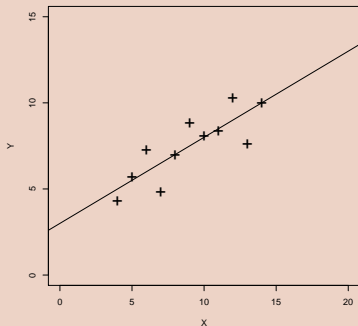
Intercept = 3

Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

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

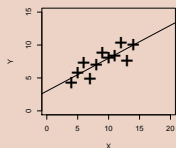
Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

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

Scatter plot

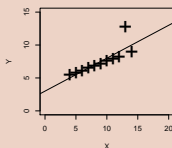


Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

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

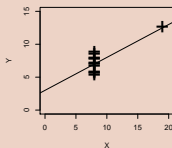
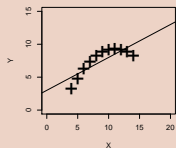


Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

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



Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

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

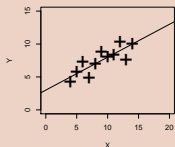
Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

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

Scatter plot

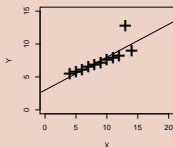


Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

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

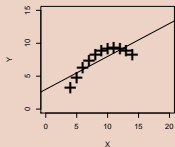


Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816

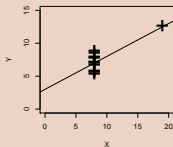
$X^{(4)}$	$Y^{(4)}$
-----------	-----------



Slope = 0.5

Res. stdev = 1.237

Correlation = 0.816



- 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".

- 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

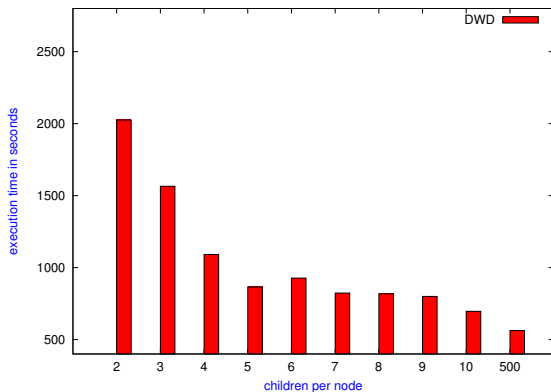
- The problem is to represent on a limited space, typically a screen with a fixed resolution, a meaningful information about the behavior of an application or system.
- \leadsto need to aggregate data and be aware of what information loss this incurs.
- Every visualization **emphasizes** some characteristics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

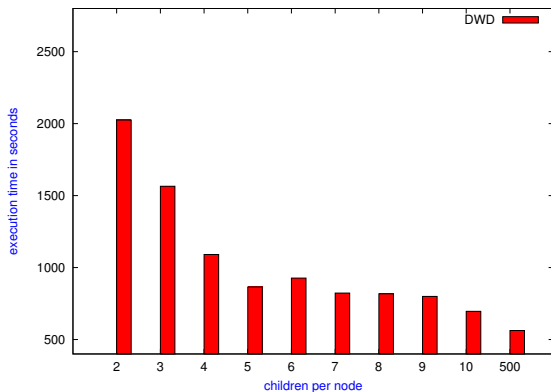
A “simple” graphical check for investigating scalability

Plotting T_p versus p .



A “simple” graphical check for investigating scalability

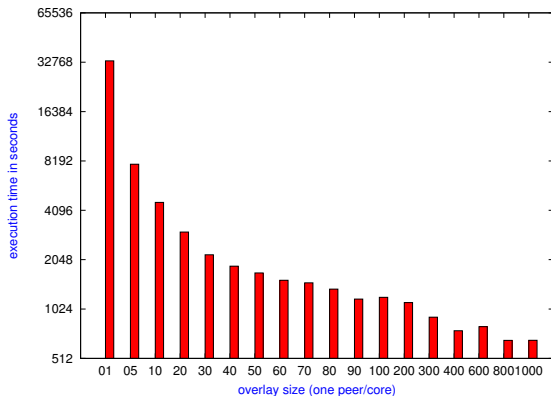
Plotting T_p versus p .



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

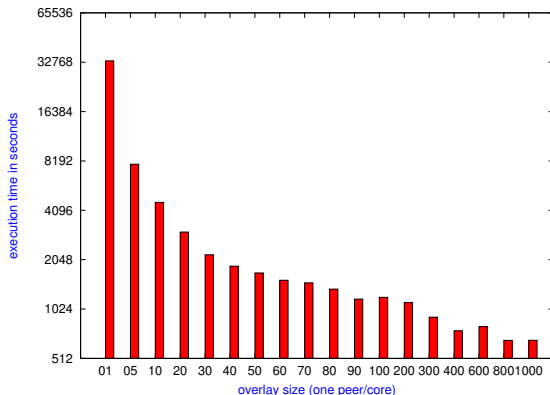
A “simple” graphical check for investigating scalability

Plotting T_p versus p .



A “simple” graphical check for investigating scalability

Plotting T_p versus p .

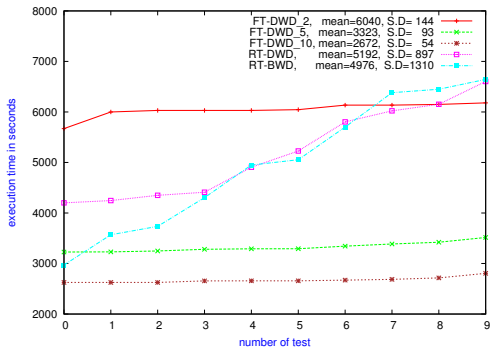


- y-axis uses log-scale
- x-axis is neither linear nor logarithmic so we cannot reason about the shape of the curve

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

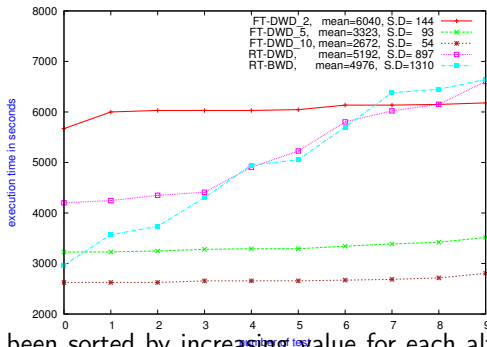
Graphically checking which alternative is better ?

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!

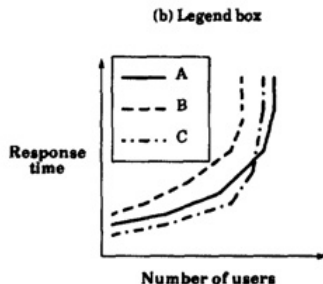
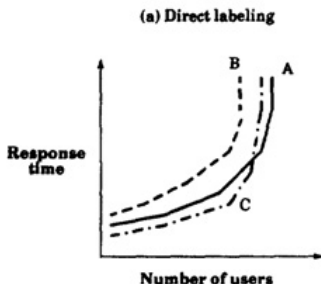
Conclusion

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*. ;)

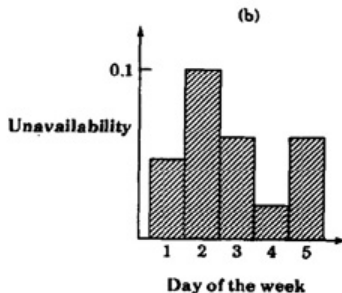
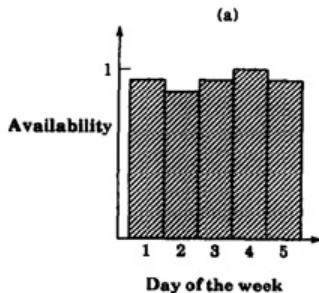
Guidelines

- 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, ...)



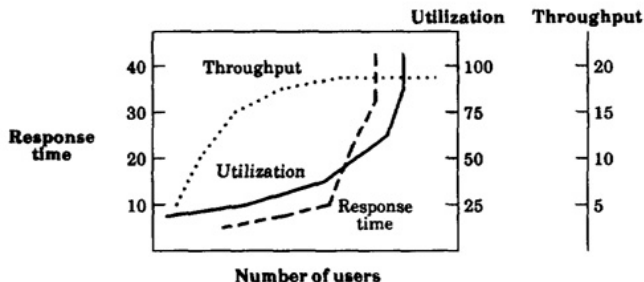
Guidelines

- 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, ...)



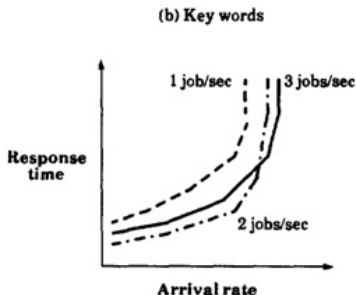
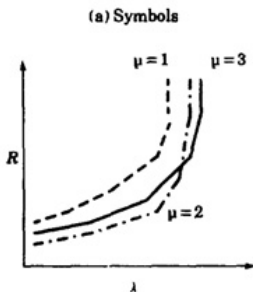
Guidelines

- 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, ...)



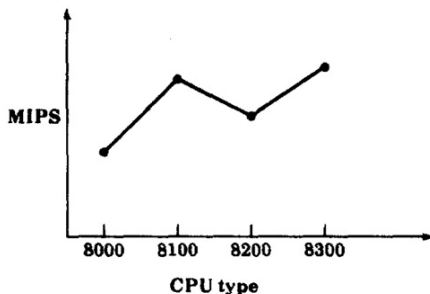
Guidelines

- 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, ...)



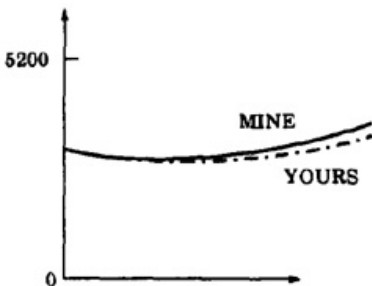
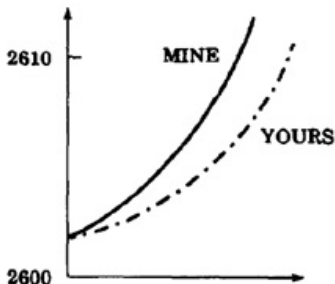
Guidelines

- 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, ...)



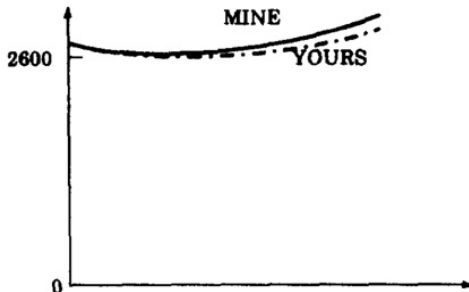
Guidelines

- 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, ...)



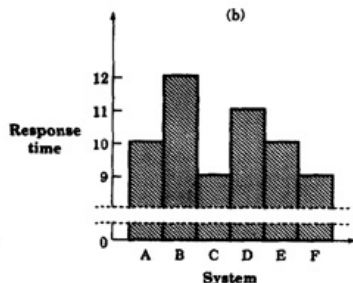
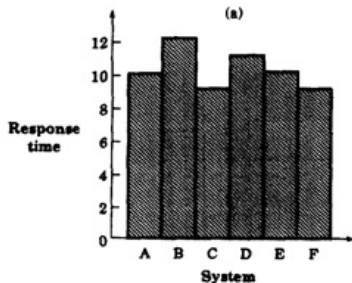
Guidelines

- 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, ...)



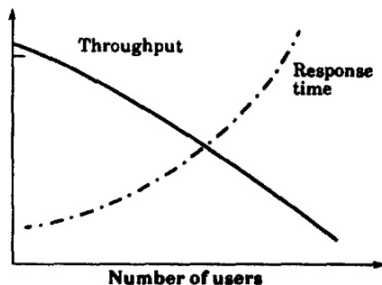
Guidelines

- 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, ...)



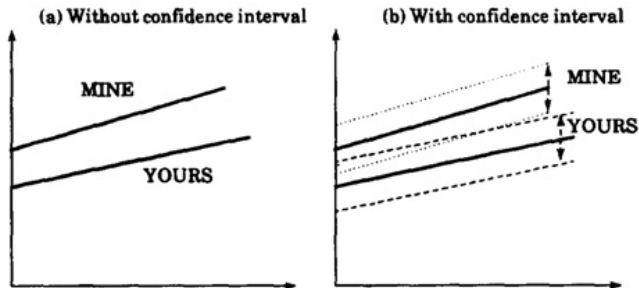
Guidelines

- 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, ...)

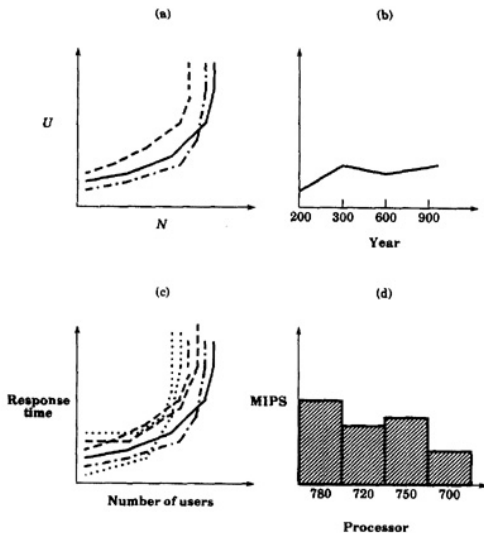


Guidelines

- 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

① Data Visualization

Motivation

Jain, Chapter 10

② Needful R Packages by Hadley Wickam

Plyr And Dplyr

Ggplot2

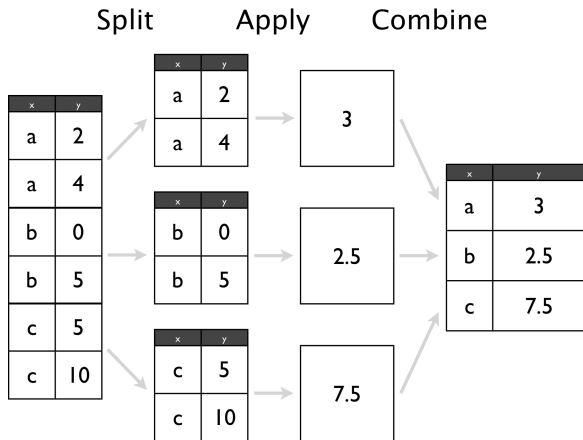
Reshape and tidyR

Now let's play!

Conclusion

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

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

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

```

1 library(dplyr)
2 mtcars %>% group_by(cyl,carb) %>%
3   select(wt,qsec) %>%
4   summarise(num = n(),
5             wt_mean = mean(wt), wt_sd = sd(wt),
6             qsec_mean = mean(qsec), qsec_sd = sd(qsec)) %>%
7   filter(num>=1)

```

```

1 Source: local data frame [9 x 7]
2 Groups: cyl
3
4   cyl carb num  wt_mean    wt_sd qsec_mean  qsec_sd
5 1     4   1   5 2.151000 0.2627118  19.37800 0.6121029
6 2     4   2   6 2.398000 0.7485412  18.93667 2.2924368
7 3     6   1   2 3.337500 0.1732412  19.83000 0.5515433

```

Outline

① Data Visualization

Motivation

Jain, Chapter 10

② Needful R Packages by Hadley Wickam

Plyr And Dplyr

Ggplot2

Reshape and tidyR

Now let's play!

Conclusion

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/>

Activities | Navigateur Web Chromium | mar. 04:42 | fr | Arnaud Legrand

Google Agenda | Le Touvet à A5 - Geo | Chopin Valse in F min: x | Index: ggplot2 0.9.3 | x

docs.ggplot2.org/current/ | Debian.org | Latest News | Help

ggplot2 0.9.3.1 | Index

Help topics

Geoms

Geoms, short for geometric objects, describe the type of plot you will produce.

- [geom_line](#)
Line specified by slope and intercept.
- [geom_area](#)
Area plot.
- [geom_bar](#)
Bars, rectangles with bases on x-axis.
- [geom_histogram](#)
Add histogram of 2d bin counts.
- [geom_blank](#)
Blank, draws nothing.
- [geom_boxplot](#)
Box and whiskers plot.
- [geom_contour](#)
Display contours of a 3d surface in 2d.
- [geom_crossbar](#)
Hollow bar with middle indicated by horizontal line.
- [geom_density](#)
Display a smooth density estimate.
- [geom_density2d](#)
Contours from a 2d density estimate.
- [geom_dotplot](#)
Dotplot.
- [geom_errorbar](#)
Error bars.
- [geom_errorbarh](#)
Horizontal error bars.
- [geom_freqpoly](#)

Dependencies

- **Depends:** stats, methods
- **Imports:** plyr, digest, grid, reshape2, scales, proto, MASS
- **Suggests:** quantreg, Hmisc, maptools, maps, heatmap, maptools, multcomp, rma, lme4
- **Extends:**

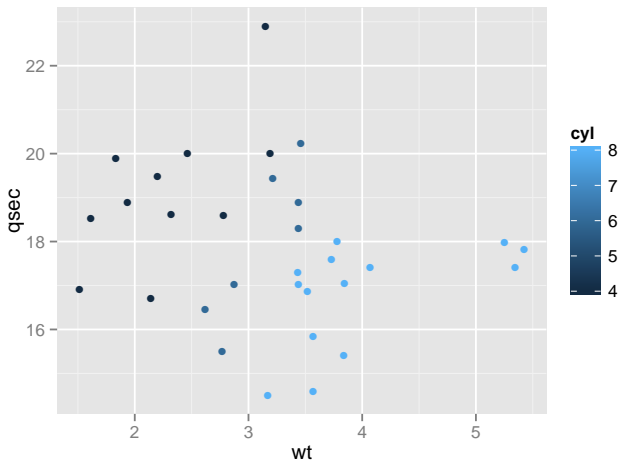
`p = geom_point(aes(size = qsec)) + scale_area()`

`scale_area` is deprecated. Use `scale_size_area` instead.
Note that the behavior of `scale_size_area` is slightly different:
by default it makes the area proportional to the numeric value. (deprecated; last used in version 0.9.2)

20 / 41

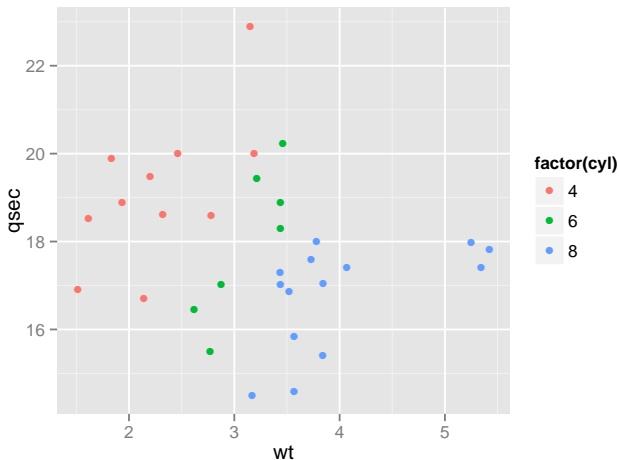
ggplot2: Illustration (1)

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



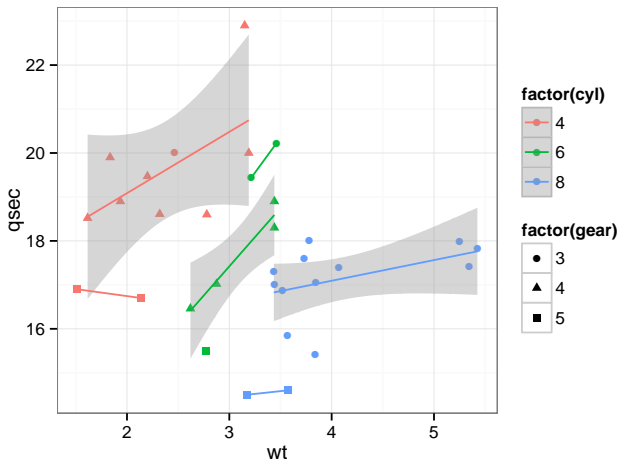
ggplot2: Illustration (2)

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



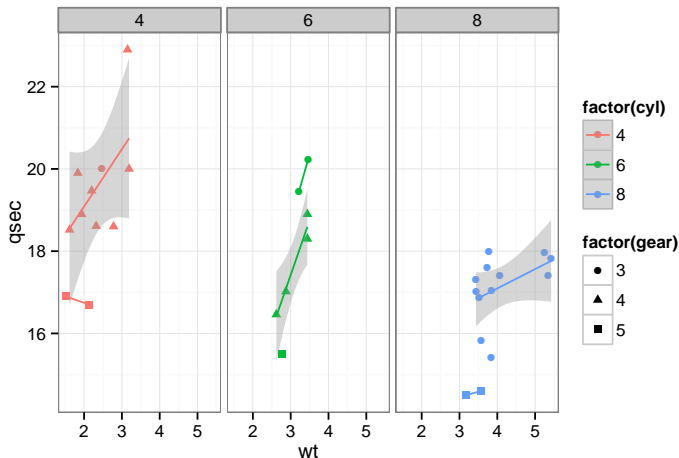
ggplot2: Illustration (3)

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



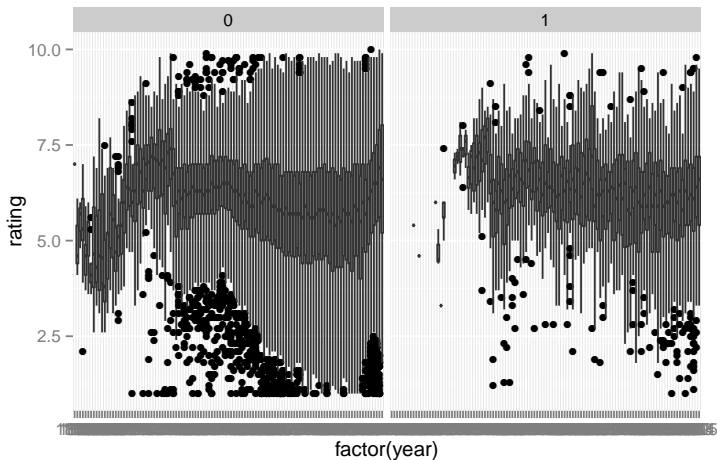
ggplot2: Illustration (4)

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



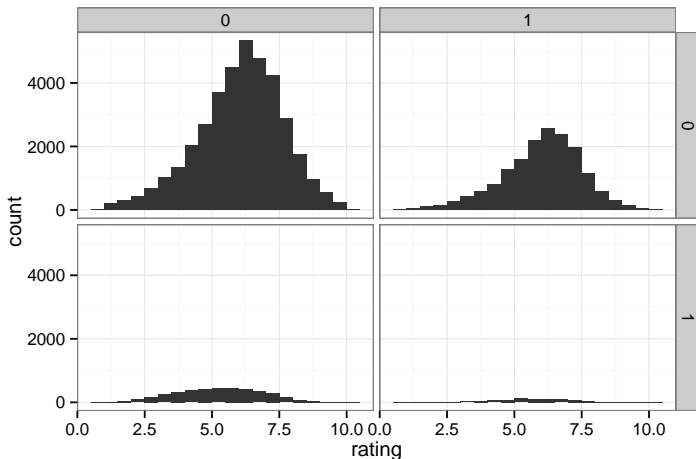
ggplot2: Illustration (5)

```
1 ggplot(data = movies, aes(x=factor(year),y=rating)) +  
2   geom_boxplot() + facet_wrap(~Romance)
```



ggplot2: Illustration (6)

```
1 ggplot(movies, aes(x = rating)) + geom_histogram(binwidth = 0.5) +  
2   facet_grid(Action ~ Comedy) + theme_bw();
```



Outline

① Data Visualization

Motivation

Jain, Chapter 10

② Needful R Packages by Hadley Wickam

Plyr And Dplyr

Ggplot2

Reshape and tidyR

Now let's play!

Conclusion

"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**

```
1 messy <- data.frame(  
2   name = c("Wilbur", "Petunia", "Gregory"),  
3   a = c(67, 80, 64),  
4   b = c(56, 90, 50)  
5 )  
6 messy
```

```
1      name  a  b  
2 1 Wilbur 67 56  
3 2 Petunia 80 90  
4 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

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

```
1      name drug heartrate
2 1 Wilbur    a         67
3 2 Petunia   a         80
4 3 Gregory   a         64
5 4 Wilbur    b         56
6 5 Petunia   b         90
7 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.

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

```
1      name drug heartrate
2 1 Wilbur    a         67
3 2 Petunia   a         80
4 3 Gregory   a         64
5 4 Wilbur    b         56
6 5 Petunia   b         90
7 6 Gregory   b         50
```

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

Outline

① Data Visualization

Motivation

Jain, Chapter 10

② Needful R Packages by Hadley Wickam

Plyr And Dplyr

Ggplot2

Reshape and tidyR

Now let's play!

Conclusion

Summarizing information

You may like these **cheat sheets**:

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

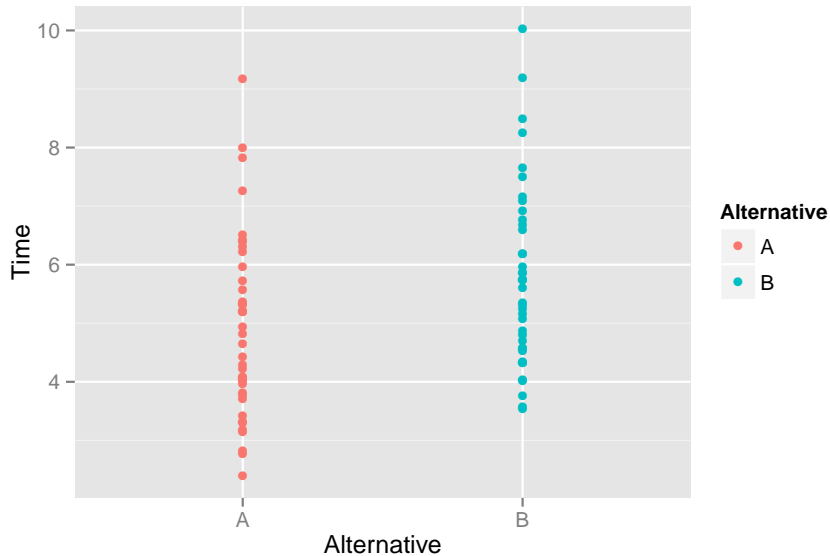
```
1 df = read.csv("data/set1.csv",header=T)
2 # Alternatively: read.csv("https://raw.githubusercontent.com/
3 #                       alegrand/SMPE/master/lectures/data/set1.csv")
4 head(df,n=2)
```

```
1           A           B
2 1 7.256717 8.261171
3 2 3.813100 4.335301
```

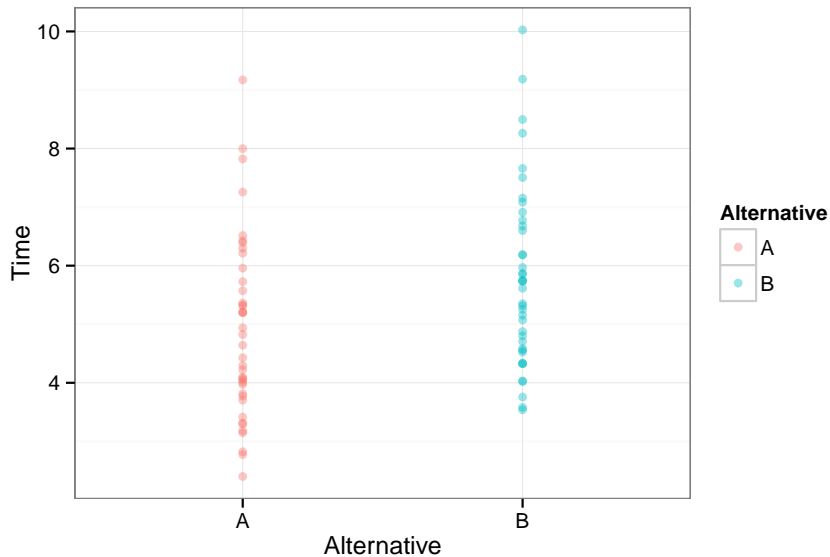
Get the following summary using `plyr/reshape` or `dplyr/tydir`:

```
1 Source: local data frame [2 x 6]
2
3   Alternative num    mean      sd      min      max
4 1           A  40 4.903817 1.544423 2.400016 9.172525
5 2           B  40 5.783643 1.542987 3.539874 10.027147
```

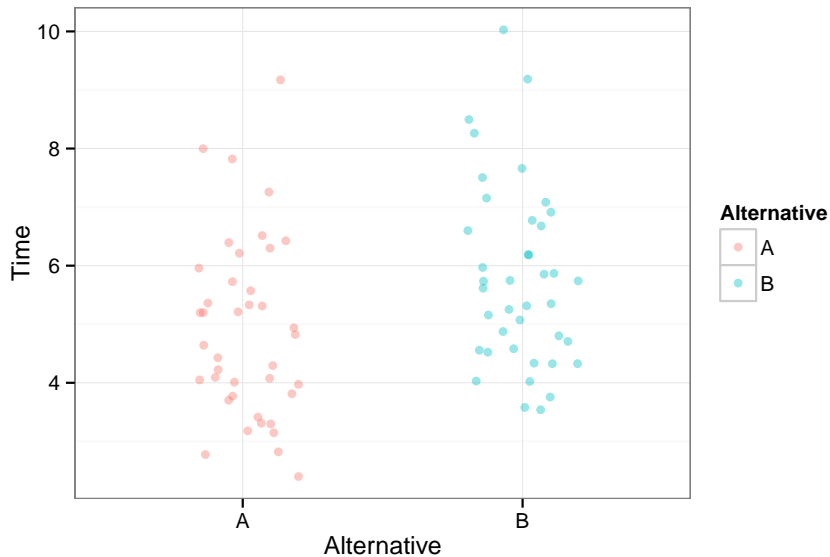
Plot the data



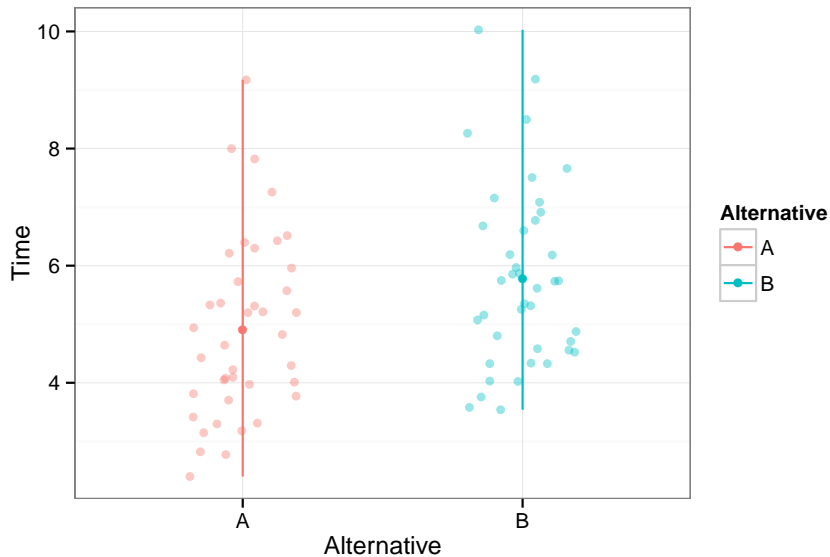
Alleviate over-plotting



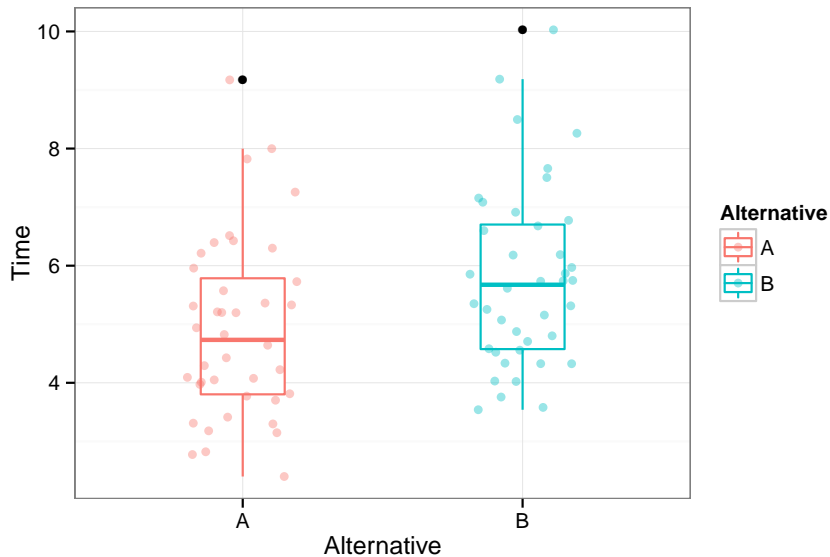
Avoid over-plotting



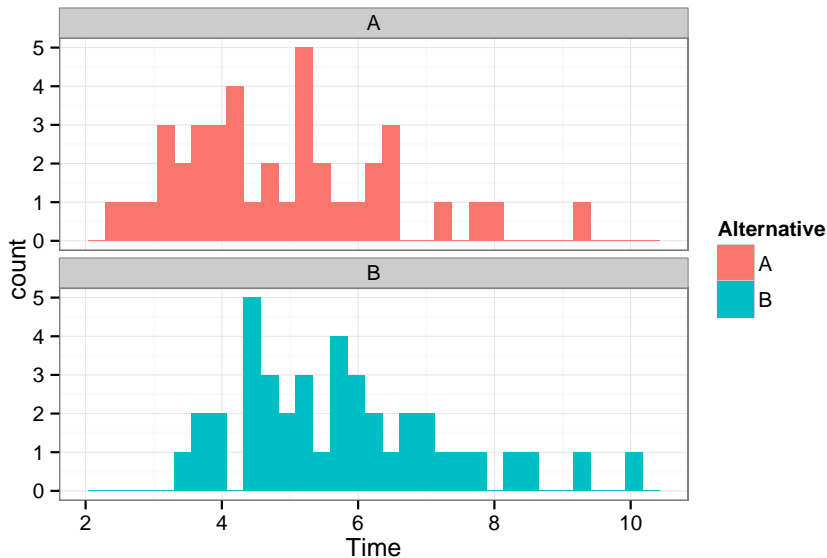
Add summary information



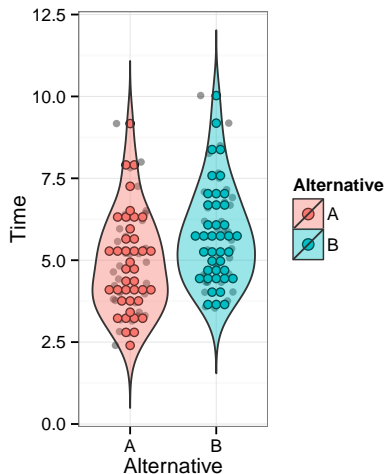
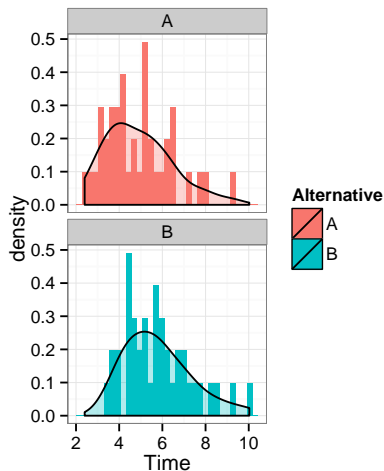
Add more standard summaries



Or use histograms. . .



Be careful with fancy plots you do not fully understand!



Outline

① Data Visualization

Motivation

Jain, Chapter 10

② Needful R Packages by Hadley Wickam

Plyr And Dplyr

Ggplot2

Reshape and tidyR

Now let's play!

Conclusion

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, ...