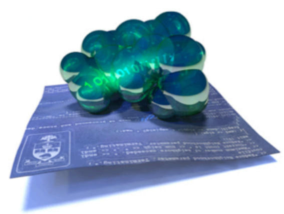
A
Bioinformatics
Course

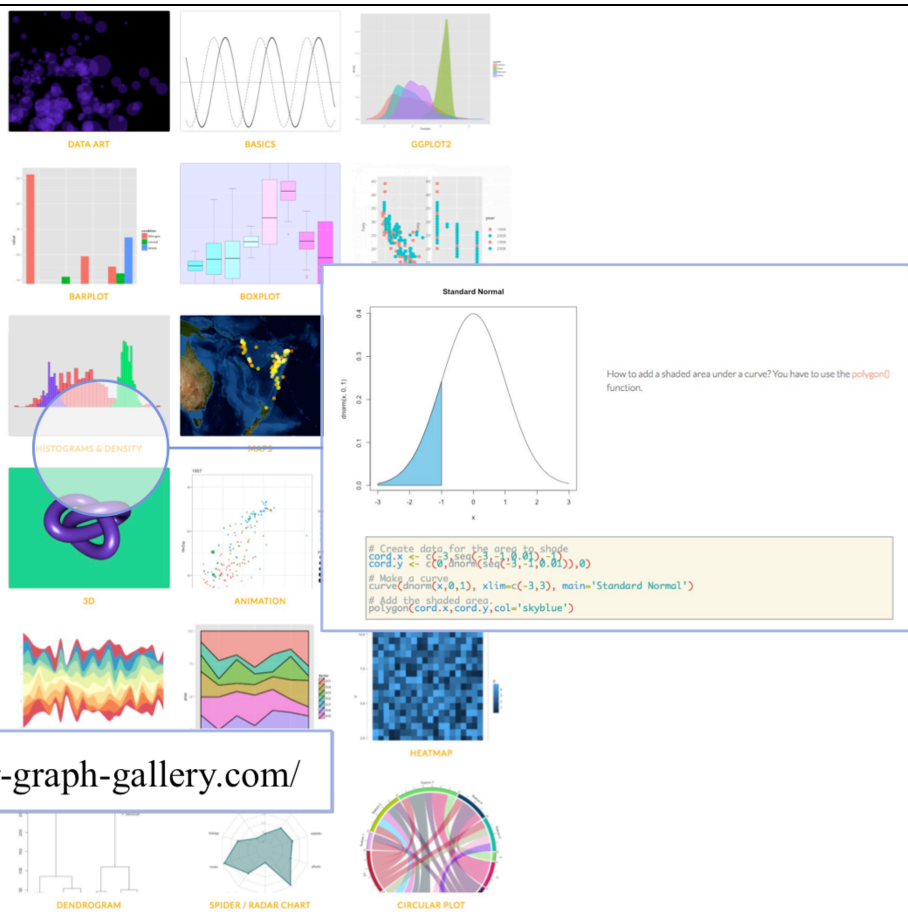
Plotting



Boris Steipe

department of biochemistry – department of molecular genetics
University of Toronto

Graphics



http://www.r-graph-gallery.com/

Easy access to expressive graphics is one of the greatest benefits of **R**. The R graph gallery has many examples, complete with source-code. **Very** useful site.

Good graphics are immensely valuable. Poor graphics are worse than none.

If you want to learn more about good graphics and information design, find a copy of Edward Tufte's **The Visual Display of Quantitative Information**. You can also visit his Web site to get a sense of the field (www.edwardtufte.com).

Fundamentally, there is one simple rule.

Use less ink.

The rule has many corollaries.

Being able to create graphs does not automatically mean being able to create **good** graphs.

Use Less Ink

To “use less ink” ...

Make sure that all elements on your graphics are necessary.

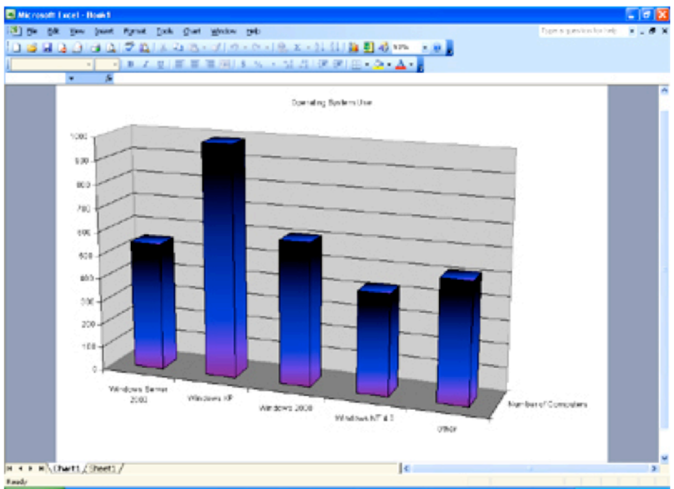
Make sure that all elements on your graphics are informative.

Make sure that all information in your data is displayed.

Not all of **R**’s plotting defaults adhere to this golden rule.

EXAMPLE

“... What if you could gussy up a report or pretty up a chart without much additional work? What if, using just one extra line of code, you could create a Microsoft Excel column chart that included a cool gradient fill like this one:”



(From Microsoft TechNet)

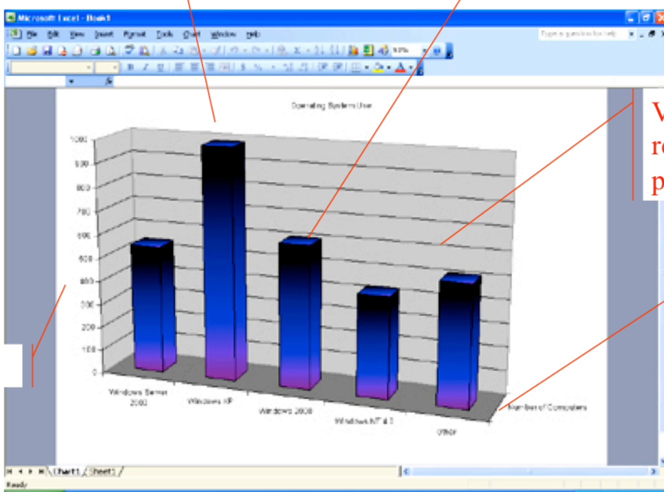
But here is an example from *another* world. I didn’t make this up, seriously. What's wrong with this "graph"? Spot the problems!

Example

“... What if you could gussy up a report or pretty up a chart without much additional work? What if ... a line ... could create a Microsoft E... hat ... gradient fill like this one.”

Only five numbers actually

Meaningless colors, no connection to actual scale. Note that the range differs for each stack!



Values can't be easily retrieved from graph due to parallax.

What does the third dimension even mean?

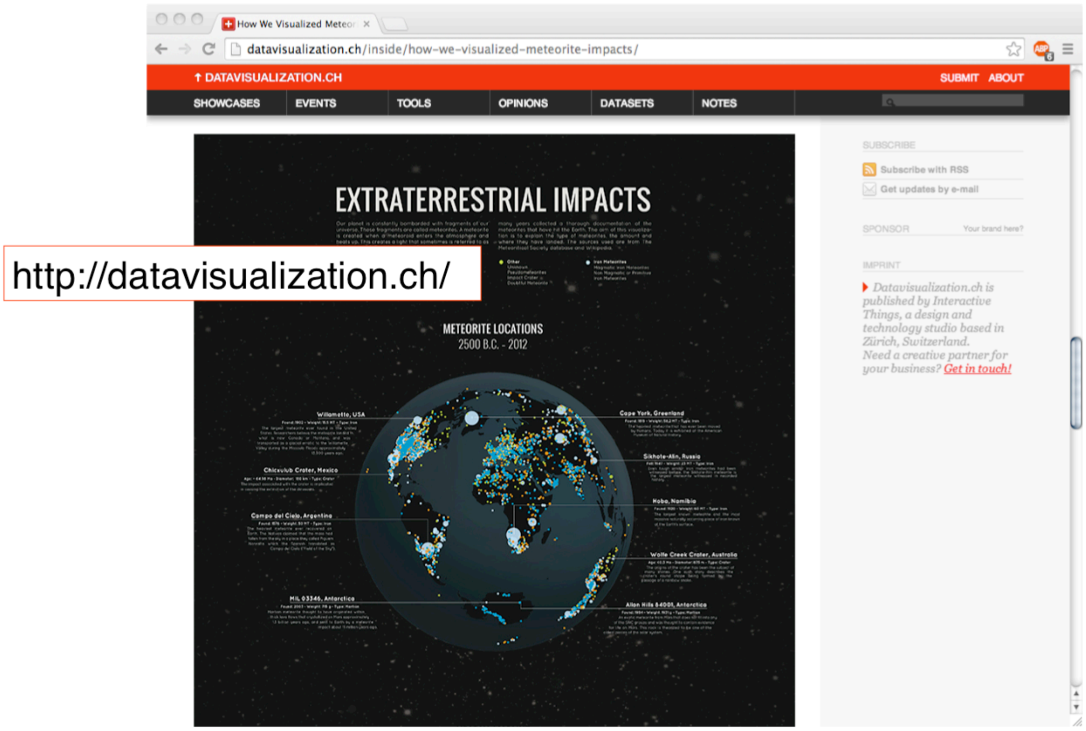
No units

(From Microsoft TechNet)

You will find again and again and again: just because you **can** do something does not automatically mean it’s a good idea.

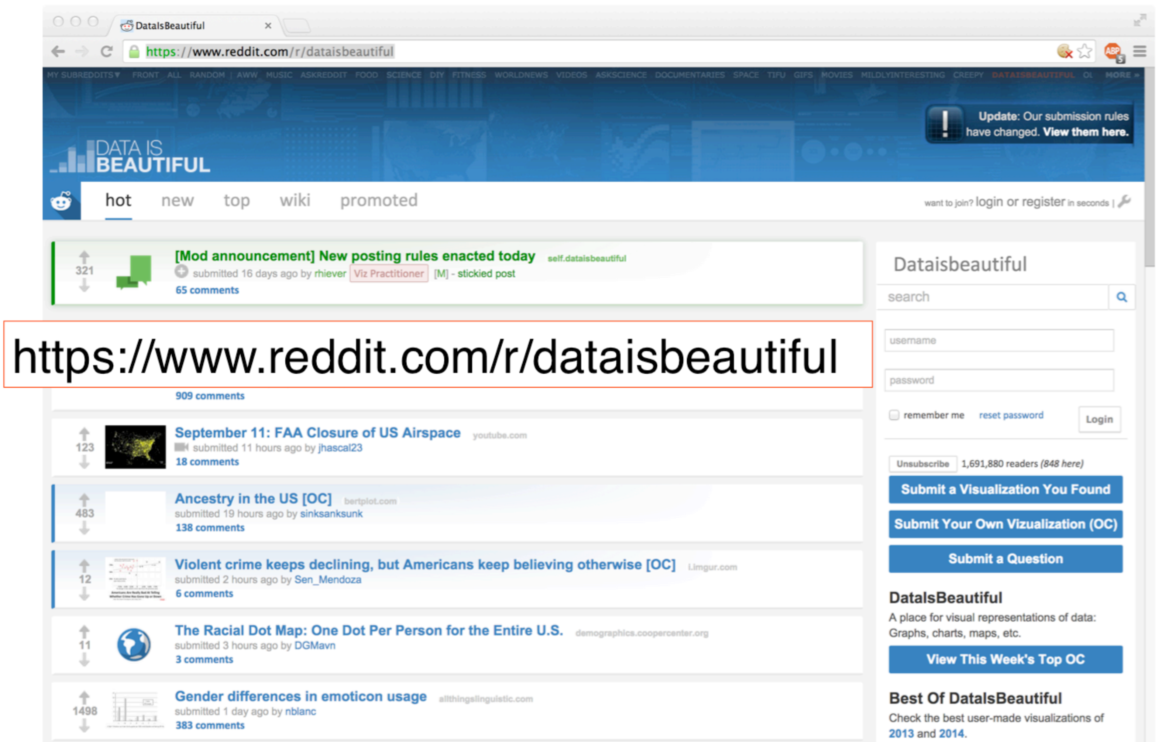
(Hold my beer.)

Better examples



Good graphics examples can be found on the web. These are examples of "information design" – not "gussying up your data". The goal is not to hide the insufficiency of your data in a pretty picture, but to make the relationships and significance as obvious as possible. The goal is "storytelling".

Better examples



This "sub-reddit" does not only post interesting examples of data visualization, but the comments often explore **why** a visualization is good (or bad) and what could be improved. Good source for learning new approaches.

We often need to quickly 'quantify' a data set, and this can be done using a set of *summary statistics* (mean, median, variance, standard deviation).

```
x <- rnorm(100, mean=0, sd=1)
> mean(x)
[1] 0.002912563
> median(x)
[1] -0.0594199
> IQR(x)
[1] 1.264738
> var(x)
[1] 1.04185
> summary(x)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.272000 -0.608800 -0.059420  0.002913  0.655900  2.582000
```

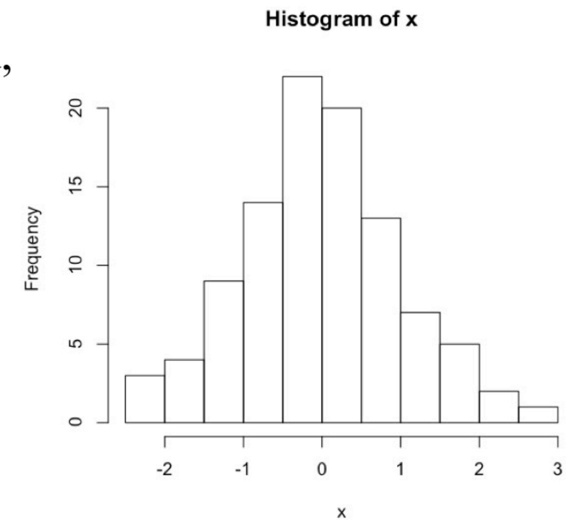
Think: what are the *units* of variance and standard deviation?

Let's talk about about simple graphics for descriptive statistics.

Histograms are often a good, first view of the data.

Random sampling: Generate 100 observations from a N(0, 1).

Histograms can be used to estimate densities!



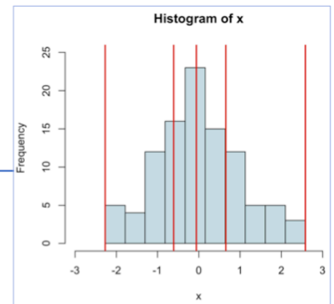

```
x <- rnorm(100, mean=0, sd=1)
hist(x)
```

GRAPHICS FOR DESCRIPTIVE STATISTICS

Empirical Quantiles can be thought of as summing over the stacks of a histogram:

The *p*-quantile has the property that *p*% of the observations are less than or equal to it.

Empirical quantiles can be easily obtained in R.




```
> set.seed(100)
> x <- rnorm(100, mean=0, sd=1)
> quantile(x)
        0%        25%        50%        75%       100%
-2.2719255 -0.6088466 -0.0594199  0.6558911  2.5819589
> quantile(x, probs=c(0.1, 0.2, 0.9))
       10%        20%        90%
-1.1744996 -0.8267067  1.3834892
```

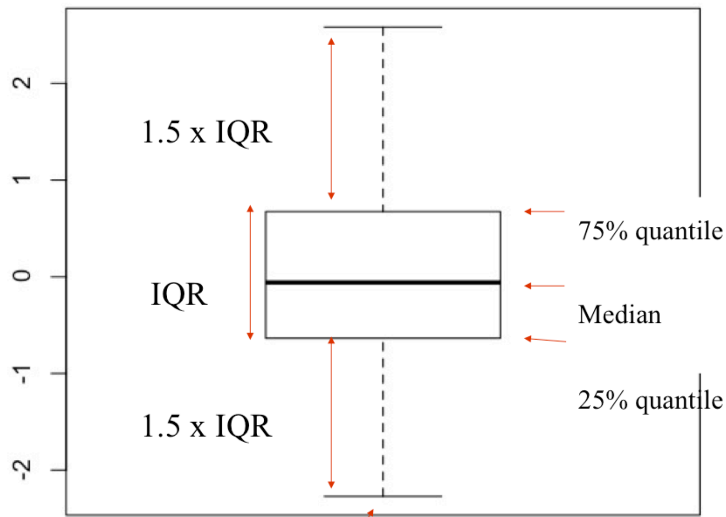
The R function `rnorm()` returns random deviates – i.e. random numbers drawn from a probability distribution. I use it here to illustrate the calculation of quantiles. The inset picture shows a histogram of the values, and five vertical lines corresponding to the quantile boundaries. You can estimate that the area under the curve (area in the histograms) for each quantile is the same.

The plot:

```
qBounds <- quantile(x)
hist(x,
     breaks=seq(min(x), max(x), by=((max(x) - min(x)) / 10) ),
     xlim=c(floor(min(x)), ceiling(max(x))),
     ylim=c(0,25),
     col="#BBD5DD")
abline(v=qBounds, col="#CC0000", lwd=2)
```

Descriptive statistics can also be summarized in a Box plot.

```
x <- rnorm(100,
   mean=0, sd=1)
boxplot(x)
```



Everything above and below 1.5 x IQR is considered an "outlier".

IQR = Inter Quantile Range = 75% quantile – 25% quantile

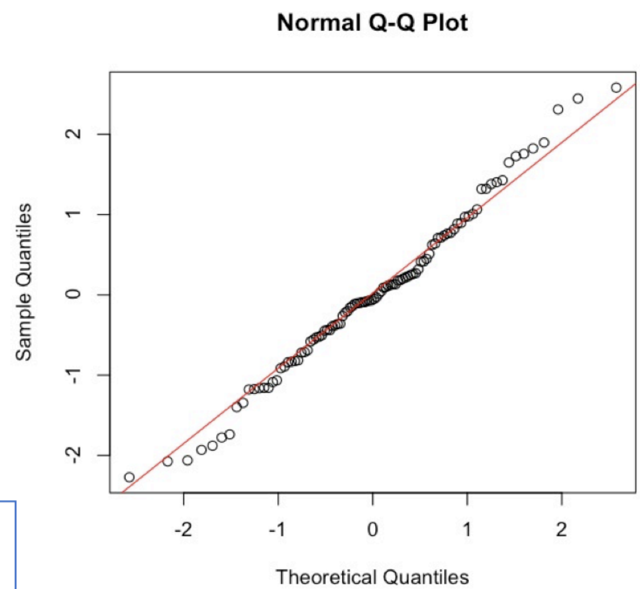
Many statistical methods make some assumption about the distribution of the data (e.g. Normal distribution).

A quantile-quantile plot is a graphical method to visually verify such assumptions.

A QQ-plot shows the theoretical quantiles versus the empirical quantiles – i.e. the quantiles we expect, vs. the quantiles we actually observe. If the distribution assumed (theoretical one) is indeed the correct one, we should observe a straight line: theoretical and empirical quantiles match.

R provides `qqnorm()` and `qqplot()` to evaluate whether data is normally distributed.

qqnorm() is only valid for the normal distribution!

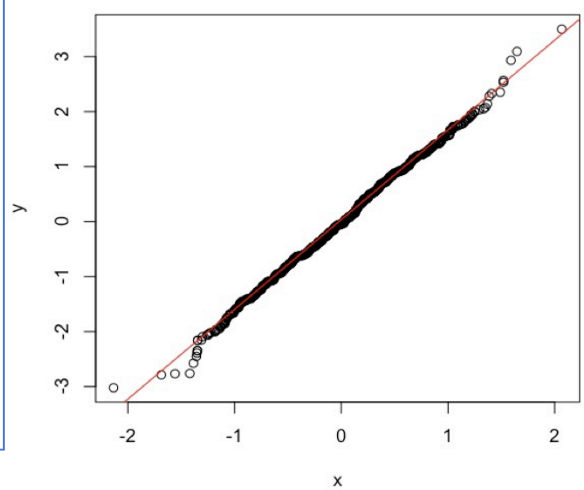



```
x <- rnorm(100, mean=0, sd=1)
qqnorm(x)
qqline(x, col=2)
```

qqplot() can plot one distribution against another. For example, we can use it to verify the *Central Limit Theorem* by simulating small perturbations of data, and comparing the result against .normally distributed values.

```
generateVariates <- function(n) {
  Nvar <- 10000
  Vout <- c()
  for (i in 1:n) {
    x <- runif(Nvar, -0.01, 0.01)
    Vout <- c(Vout, sum(x) )
  }
  return(Vout)
}

x <- generateVariates(1000)
y <- rnorm(1000, mean=0, sd=1)
qqplot(x, y)
qqline(x, y, col=2)
```



http://steipe.biochemistry.utoronto.ca/abc

B O R I S . S T E I P E @ U T O R O N T O . C A

DEPARTMENT OF BIOCHEMISTRY & DEPARTMENT OF MOLECULAR GENETICS
UNIVERSITY OF TORONTO, CANADA