It was the best of plots, it was the worst of plots

Sam Clifford

2019-10-31

Principles of graphical excellence

Many uses of visualisation in science

- Showing the data
- Showing the results of analysis
- Showing the physical phenomenon being described
- Describing an experimental setup

Tufte's principles

Tufte [1983] and Pantoliano [2012]

- Show the data
  - Induce the viewer to think about the substance of the findings rather than the methodology, the graphical design, or other aspects
  - Avoid distorting what the data have to say
  - Serve a clear purpose: description, exploration, tabulation, or decoration
- Provide clarity
  - Present many numbers in a small space, i.e., efficiently
  - Make large data sets coherent
  - Be closely integrated with the statistical and verbal descriptions of the data set

Tufte's principles

- Allow comparison where appropriate
  - Encourage the eye to compare different pieces of data
  - Reveal the data at several levels of detail, from a broad overview to the fine structure
- Visual representations of data must tell the truth
- Good graphical representations maximise data-ink and erase as much non-data-ink as possible
- Avoid chartjunk, the excessive and unnecessary use of graphical effects in graphs
- Don't map the same variable to multiple graphical elements (e.g. color and y value)
- Produce graphs with high data density

Why do we visualise?

- We don’t just make graphs because it’s fun (but it totally is!)
- We do it to communicate information
- Describing with summary statistics may not tell the whole story
- Graph must communicate what’s in your mind to reader, including key relationships
- Reader should be able to understand what the graph means and not be
  - misled into thinking something that is untrue
  - distracted from the main point

Principles of graphical excellence

Building plots

Some common geometries

Small multiples

Other plotting aesthetics

Summary

References

Building plots

Building plots

- R package `ggplot2` uses a grammar of graphics [Wickham, 2010, RStudio, 2012]
    - map variables in data frame to aesthetic options in the plot
    - choose a geometry for how to display these variables
    - adjustments to axis scales
    - adjustments to colors, themes, etc.
    - adding extra commands in a “do this, then do this” manner

Building a plot

- How do we structure a call to `ggplot` to make a plot?
  - load `ggplot2` package
  - Specify we want a `ggplot` object and which data frame we're going to use,
  - set **aesthetic options** to tell R which variables to map to the x and y axes of the plot
  - state geometry we're using to show variables

```
ggplot(data = my.data.frame,
       aes(x = my.x.variable,
           y = my.y.variable)) +
  geom_point()
```

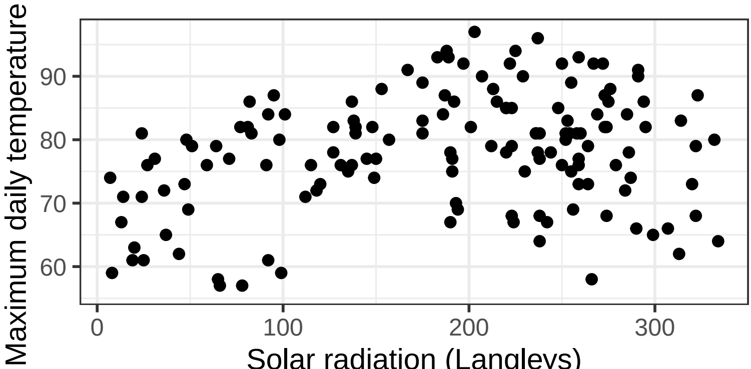
Some common geometries

Scatter plot

- For each observation in data, a pair of values (x, y) is shown as a point
- Can show more structure in the data by setting aesthetics of the geometry (mapping variables to graphical elements)
- e.g. if we want to show male and female relative bone density values with different colours

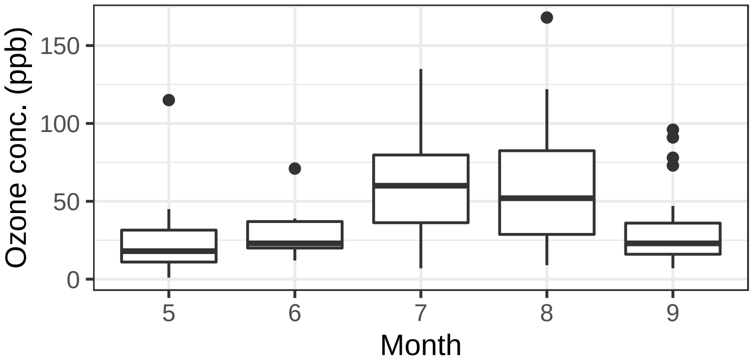
```r
data(airquality)
ggplot(data = airquality,
       aes(x = Solar.R, y = Temp)) +
  geom_point() +
  labs(x = "Solar radiation (Langleys)",
       y = "Maximum daily temperature (F)") +
  theme_bw()
```

Scatter plot



Boxplot

```
ggplot(data = airquality, aes(x = factor(Month), y = Ozone)) +
  geom_boxplot() + theme_bw() +
  labs(y = "Ozone conc. (ppb)", x = "Month")
```



- **outliers** shown as dots, indicating they’re far away from typical values

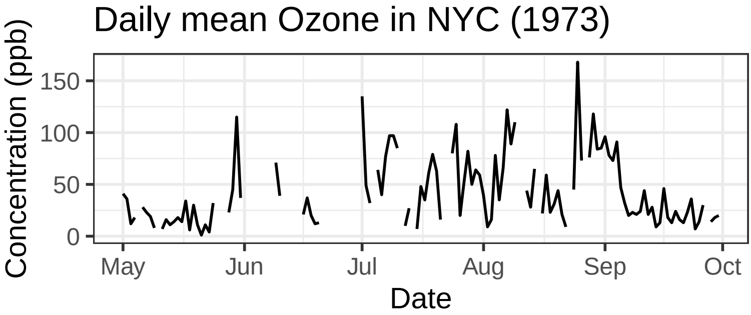
Line plot

- Similar to scatter plot, but joins pairs of values
- Useful when showing how something changes over time
- Use only when (x, y) are ordered pairs of numeric values, e.g. x is time or date
- For this reason, often referred to as **time series plot**

Line plot

- Show the Ozone concentrations over time

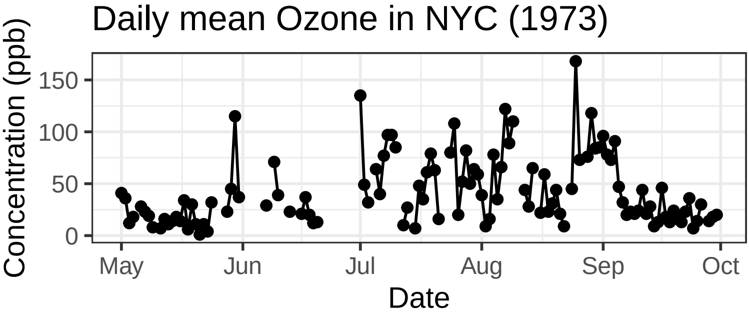
```r
# make the date column
airquality <-
  mutate(airquality,
         Date = as.Date(paste("1973", Month, Day, sep="-")))

ggplot(data=airquality, aes(x=Date, y=Ozone)) +
  geom_line() + theme_bw() +
  labs(y="Concentration (ppb)",
       title = "Daily mean Ozone in NYC (1973)")
```



Line plot

- `geom_line()` stops plotting when it hits an NA value
- If we have individual measurements in a group of NA values it won't plot that value
- Can use multiple geometries to display the same variables

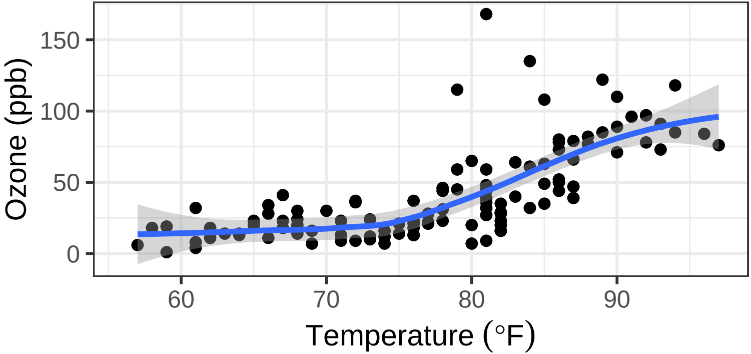
```
ggplot(data=airquality, aes(x=Date, y=Ozone)) +
  geom_line() + geom_point() + theme_bw() +
  labs(y="Concentration (ppb)",
       title = "Daily mean Ozone in NYC (1973)")
```



Smooth plot

- Often too much data in a scatter plot to see pattern
- Maybe we want to show the reader the trend in the data
- `geom_smooth()` generates a **scatterplot smoother** that shows the overall relationship between y and x

```r
ggplot(data=airquality, aes(x=Temp, y=Ozone)) +
  geom_point() + geom_smooth() + theme_bw() +
  labs(x=expression(Temperature~(degree*F)), y="Ozone (ppb)")
```



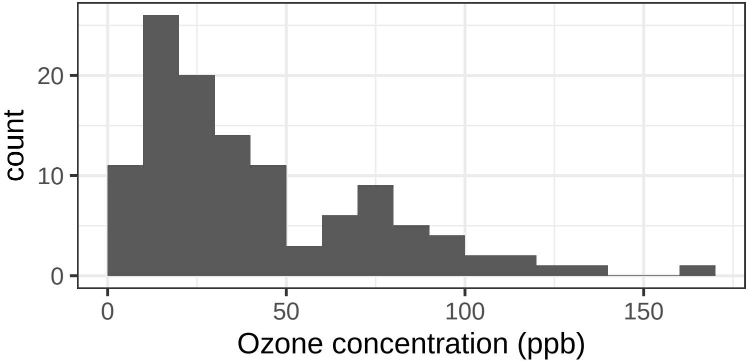
Bar/column plots

- Good for showing the amount of something (requires meaningful zero)
- `geom_col()` - xy plot given some variables `x`, `y`
- `geom_bar()` - counts number of of times categorical x occurs
- `geom_histogram()` - counts number of times x in bin

Bar/column plots

- e.g. a histogram of Ozone concentrations

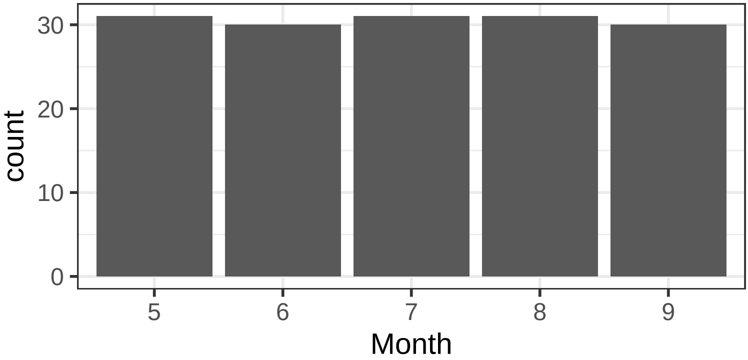
```
ozone_hist <-
  ggplot(data = airquality, aes(x = Ozone)) +
  geom_histogram(binwidth = 10, boundary = 0) +
  labs(x = "Ozone concentration (ppb)") +
  theme_bw()

ozone_hist
```



Bar/column plots

- e.g. a bar plot of month

```
ggplot(data = airquality, aes(x = factor(Month))) +
  geom_bar() + labs(x = "Month") + theme_bw()
```



Small multiples

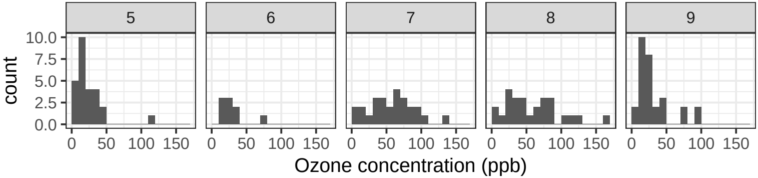
Small multiples

- Group a plot by some categorical variable
- Repeat a basic graph for groups in the data
    - air quality data has information about, e.g. months
- Can view 3-5 dimensions in the data on a 2D page
    - Often a better alternative to 3D, since it doesn't distort comparisons
    - Inner axes relate to the smallest X-Y plots
    - Outer axes relate to the grouping variables
- Avoids using loops

Small multiples

- By adding one extra command we can tell R to repeat the histogram plot of Ozone concentration for each value of `Month`

```
ozone_hist + facet_wrap( ~ Month, nrow = 1)
```

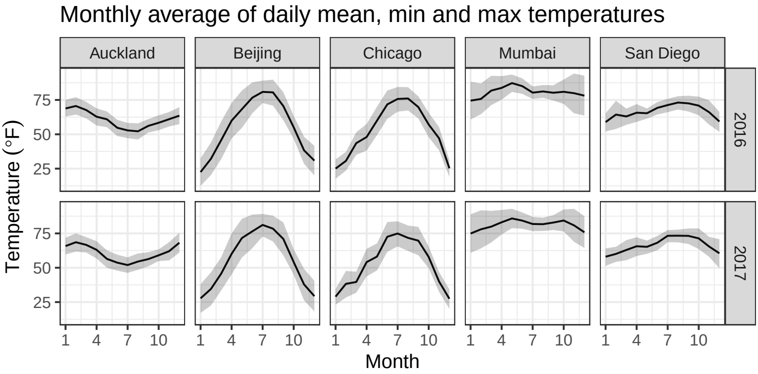


Small multiples

- If we have two (or more) grouping variables we can use `facet_grid(V1 ~ V2)` to tell R to repeat the plotting geometries for each value of `V1` and `V2` as rows and columns of a grid

```r
library(mosaicData)

data(Weather)

annual_5_cities <- Weather %>%
  group_by(city, month, year) %>%
  summarise_at(.vars = vars(contains("temp")), .funs = list(mean))  %>%
  ggplot(data = ., aes(x = month, y = avg_temp)) +
  geom_line() +
  geom_ribbon(aes(ymin = low_temp, ymax = high_temp),
              alpha = 0.25) + theme_bw() +
  xlab("Month") + ylab(expression(Temperature~(degree*F))) +
  ggtitle("Monthly average of daily mean, min and max temperatures") +
  facet_grid(year ~ city) +
  scale_x_continuous(breaks = seq(1, 12, by = 3))
```

Small multiples



Other plotting aesthetics

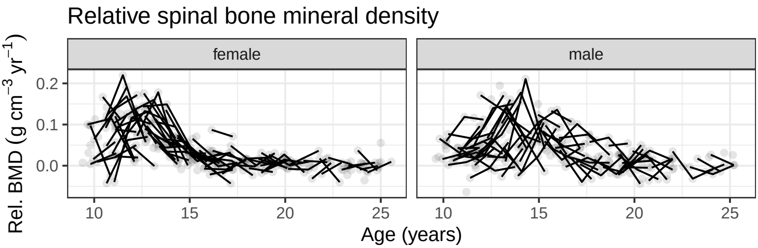
#aesthetic

- `ggplot2` allows the passing of arguments to a plotting geometry
- Map things other than x and y coordinates
- **Aesthetics** map variables to graphical elements such as
    - **group** (repeat geometry for a grouping variable in same axes)
    - **size**
    - **shape**
    - **colour**
    - **alpha** transparency
    - **fill** colour
- We need to put these inside `aes()` brackets if we wish to map a variable
- Optionally, putting them outside (but still inside the geometry) allows us to apply one value across the whole geometry

Group

- Instead of splitting all data up with small multiples, we could use grouping to show each each group on a common set of axes

```r
library(ElemStatLearn)
data(bone)

ggplot(data = bone, aes(x = age, y = spnbmd)) +
  geom_point(alpha=0.1) + facet_wrap(~ gender) +
  geom_line(aes(group = idnum)) +  theme_bw() +
  labs(x = "Age (years)", y = expression(Rel.~BMD~(g~cm^{-3}~yr^{-1})),
       title = "Relative spinal bone mineral density")
```

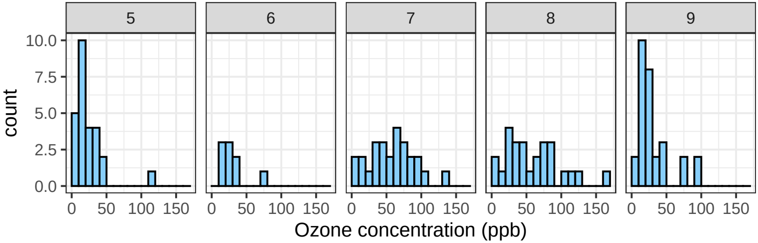


- Especially useful when many, many groups

Colour and fill

- We can change the colour for the geometry as a whole by putting it outside the `aes()` brackets
  - `colour` is the external part of geometry (e.g. bar boundary)
  - `fill` is the internal part of geometry

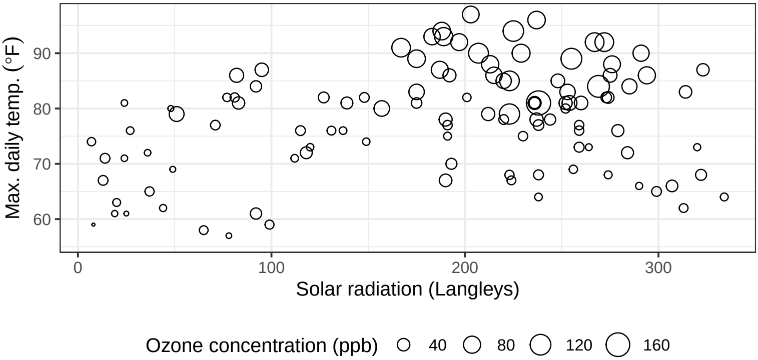
```
ggplot(data = airquality, aes(x = Ozone)) +
  geom_histogram(binwidth = 10, boundary = 0,
                 fill = "lightskyblue", color = "black") +
  labs(x = "Ozone concentration (ppb)") +
  theme_bw() + facet_wrap(~Month, nrow = 1)
```



Size

- Size refers to the elements of the plotting geometry
  - radius of points
  - thickness of lines

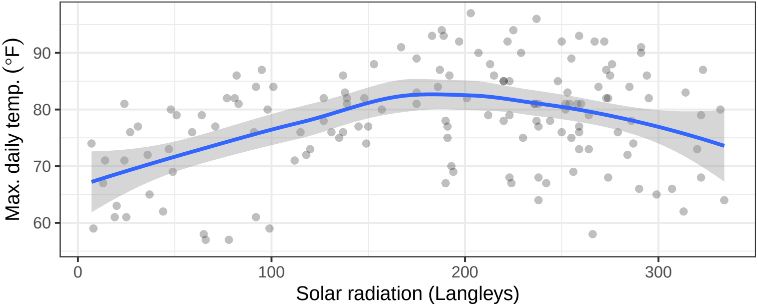
```r
ggplot(data = airquality, aes(x = Solar.R, y = Temp)) + theme_bw() +
  geom_point(aes(size = Ozone), pch = 1) + theme(legend.position = "bottom") +
  scale_size_area(name = "Ozone concentration (ppb)") +
  labs(x = "Solar radiation (Langleys)",
       y = expression(Max.~daily~temp.~(degree*F)))
```



Alpha transparency

- Alpha refers to the transparency (1 = solid, 0 = fully transparent)
- Useful when you've got lots of things stacked on top of each other in a plot

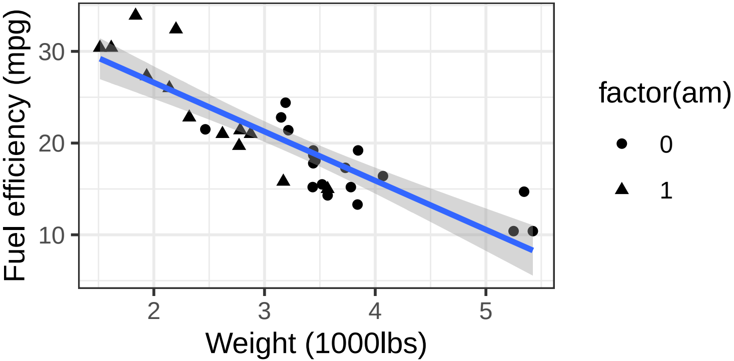
```
ggplot(data = airquality, aes(x = Solar.R, y = Temp)) + theme_bw() +
  geom_point(alpha = 0.25) + theme(legend.position = "bottom") +
  geom_smooth() +
  labs(x = "Solar radiation (Langleys)",
       y = expression(Max.~daily~temp.~(degree*F)))
```



Shape

- Can change point shape to help identify grouping
- Most useful when there’s only a few groups

```
data(mtcars)
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(shape=factor(am))) + theme_bw() +
  xlab("Weight (1000lbs)") + ylab("Fuel efficiency (mpg)") +
  geom_smooth(method = "lm")
```



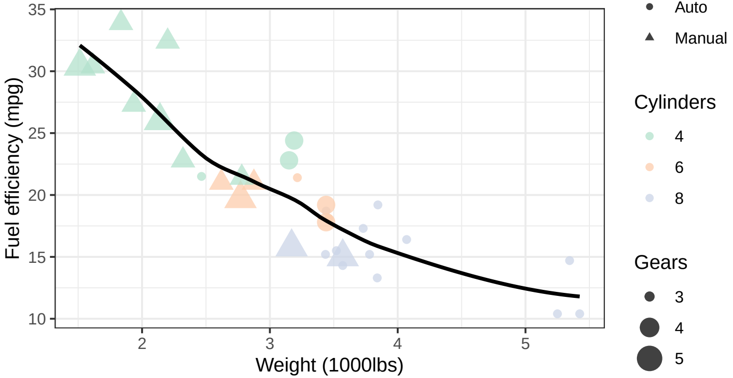
Changing the default options

Many `scale_*` functions allow us to set options for the relevant aesthetic and corresponding legend name, e.g.

- `scale_color_gradient()` makes a color gradient for when we use `aes(color=...)`
- `scale_fill_brewer()` sets a color palette for `aes(fill=...)` using colour schemes at http://colorbrewer2.org/
- `scale_shape(name = "Transmission", ...)` changes the title from "factor(am)" to "Transmission" for `aes(shape=factor(am))` in the previous slide
- `scale_x_log10()` changes the x axis to have a logarithmic scale in increasing powers of 10.
- Find more at the `ggplot2` documentation page

Changing the default options

```
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(alpha=0.75, aes(shape=factor(am), color = factor(cyl),
                             size=factor(gear))) + theme_bw() +
  scale_shape(name="Transmission", breaks=c("0", "1"),
              labels=c("Auto", "Manual")) +
  scale_size_discrete(name="Gears") +
  scale_color_brewer(name="Cylinders", palette = "Pastel2") +
  xlab("Weight (1000lbs)") + ylab("Fuel efficiency (mpg)") +
  geom_smooth(method = "loess", se=FALSE, color = "black")
```



Summary

Summary

- We make graphs to tell a story with data
- Graphs should draw the reader in and explain what they're seeing
- Plots are built from
  - geometric objects
  - axis scales
  - coordinate systems (linear scale, logarithmic scale, 2D, 3D, etc.)
  - annotations (e.g. heading in small multiples)

Summary

- Successively building a plot with a grammar of graphics allows development of complex plots from simple elements and small changes
- Choose a plotting geometry that helps tell the story
- Meaningful labels remove ambiguity and confusion

Further reading

- Extra notes on Tufte's principles
- History of visualisation
  - Friendly [2005]
  - Friendly [2006]
- Visualisation to help decision making
  - Tufte [1997]
- ggplot2 resources
  - Wickham [2010]
  - RStudio [2012]

M. Friendly. Milestones in the history of data visualization: A case study in statistical historiography. In C. Weihs and W. Gaul, editors, *Classification: The Ubiquitous Challenge*, pages 34–52. Springer, New York, 2005. URL http://www.math.yorku.ca/SCS/Papers/gfkl.pdf.

M. Friendly. A brief history of data visualization. In C. Chen, W. Härdle, and A Unwin, editors, *Handbook of Computational Statistics: Data Visualization*, volume III. Springer-Verlag, Heidelberg, 2006. URL http://www.datavis.ca/papers/hbook.pdf.

Mike Pantoliano. Data visualization principles: Lessons from tufte, 2012. URL https://moz.com/blog/data-visualization-principles-lessons-from-tufte.

RStudio. Rstudio cheat sheets, 2012. URL https://www.rstudio.com/resources/cheatsheets/.

Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

E.R. Tufte. *Visual and statistical thinking*. Graphics Press, 1997. ISBN 9781930824157. URL https://www.edwardtufte.com/tufte/books_textb.

Hadley Wickham. A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28, 2010. doi: 10.1198/jcgs.2009.07098.