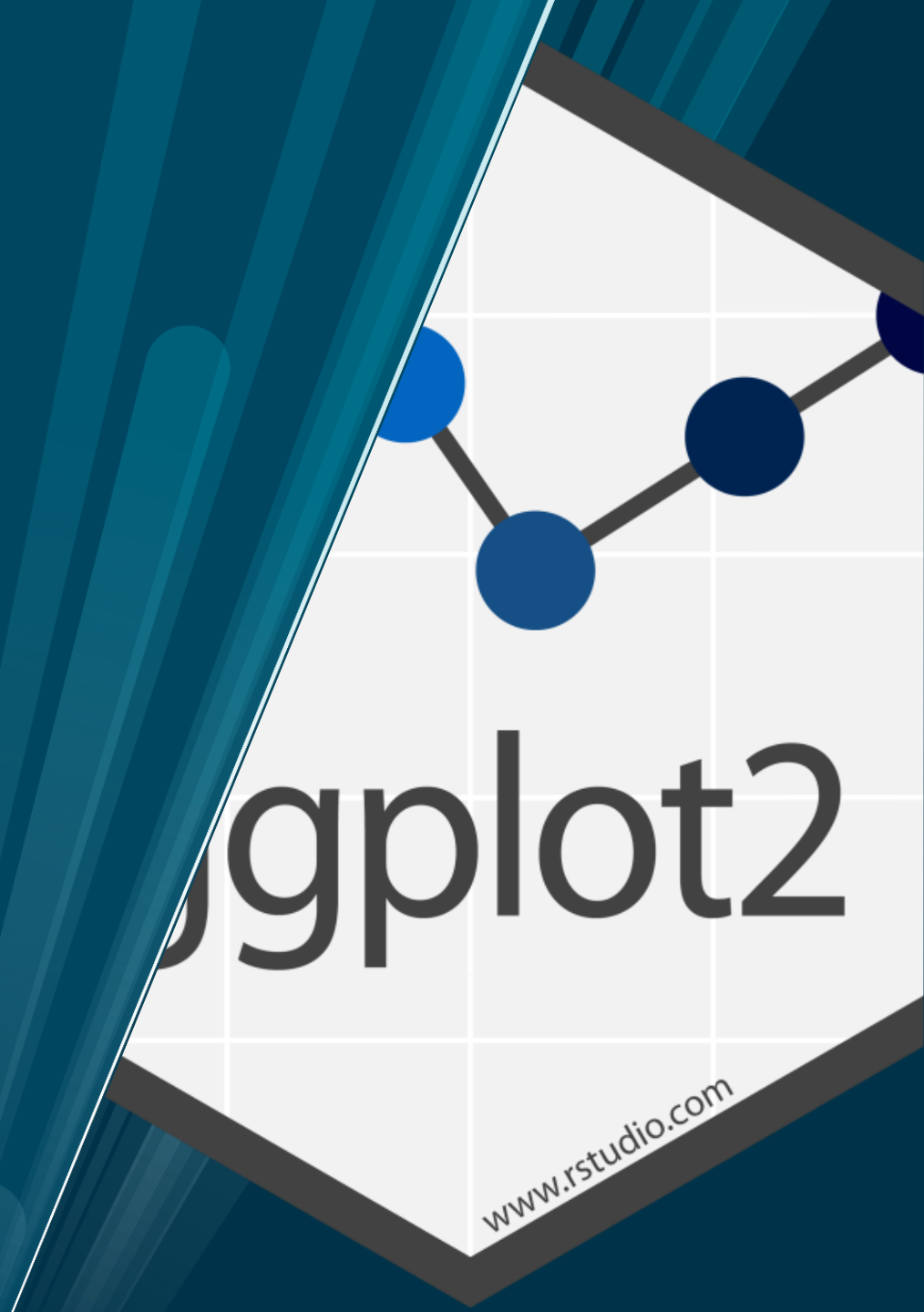
UiT The Arctic University of Norway

FSK-2053 Data science & bioinformatics for fisheries and aquaculture

Lecture 3: *Data Visualization – ggplot2*



Daniel Kumazawa Morais
daniel.morais@uit.no

Learning objectives: Session 3

3.1 Principles of data visualization

- Understand the best practices for designing representative data visualizations.

- Know how to decide the most suitable visualization for each type of data.

3.2. Introduction to ggplot2

- Understand the basics of the grammar of graphics underlying the ggplot2 workflow.

- Be able to add different layers to a graphic in ggplot2 by combining data, an aesthetic mapping, a geom, and other optional elements.

- Be able to change the aesthetics of different graphic elements in ggplot2.

- Know the basic types of visualizations in ggplot2 and how to generate them:

    scatter plots (two quantitative variables)

    line plots (quantitative variables and an ordered factor)

    bar plots (summarising values and a factor)

    jitter plots (one quantitative variable and a factor)

    box plots / violin plots (one quantitative variable, several groups and a factor)

    histograms, distributions and density plots (one quantitative variable)

    heatmaps / bubble plots (one quantitative variable and two factors)

    contour plots (three quantitative variables)

Principles of data visualization

We will show some examples of plot styles we should avoid, explain how to improve them, and use these as motivation for a list of principles. We compare and contrast plots that follow these principles to those that don't.

The principles are mostly based on research related to how humans detect patterns and make visual comparisons. The preferred approaches are those that best fit the way our brains process visual information.

When deciding on a visualization approach it is also important to keep our goal in mind. We may be comparing a viewable number of quantities, describing a distribution for categories or numeric values, comparing the data from two groups, or describing the relationship between two numerical variables.

For a data scientist it is important to adapt and optimize graphs to the audience. For example, an exploratory plot made for ourselves will be different than a chart intended to communicate a finding to a general audience.

More information:

Karl Broman's talk "Creating effective figures and tables"
https://www.biostat.wisc.edu/~kbroman/presentations/graphs2017.pdf

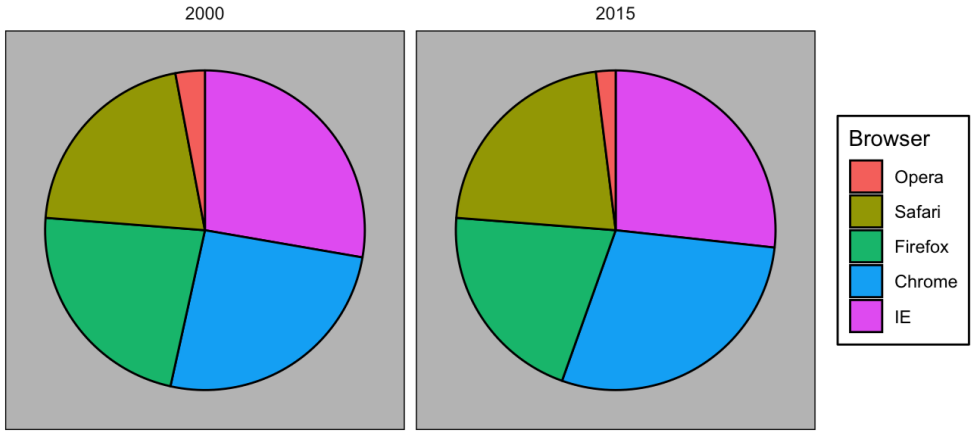
Peter Aldhous' Introduction to Data Visualization course:
http://paldhous.github.io/ucb/2016/dataviz/index.html

Encoding data using visual cues

There are several approaches at our disposal to encode data, including: position, aligned lengths, angles, area, brightness, and color hue.

A widely used graphical representation of percentages, popularized by Microsoft Excel, is the pie chart.

Avoid using pie charts! The human brain is not good at precisely quantifying angles and are even worse when only area is available. We are much better in processing information from positions and lengths.

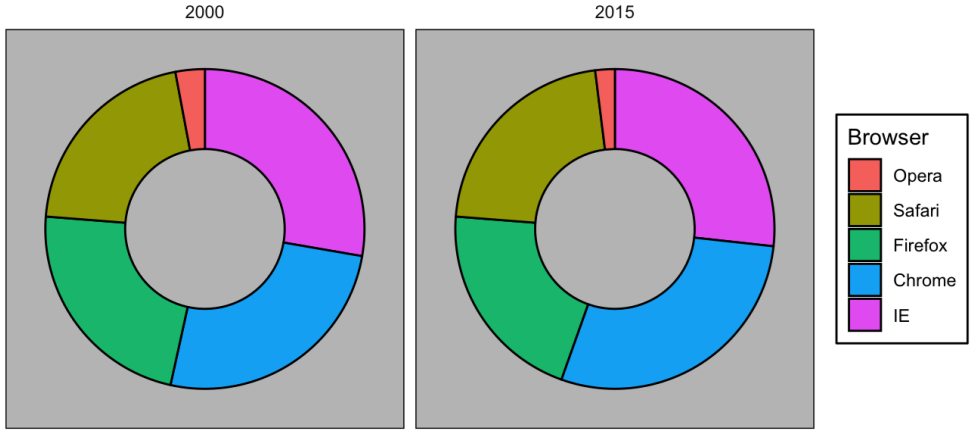


Pie chart

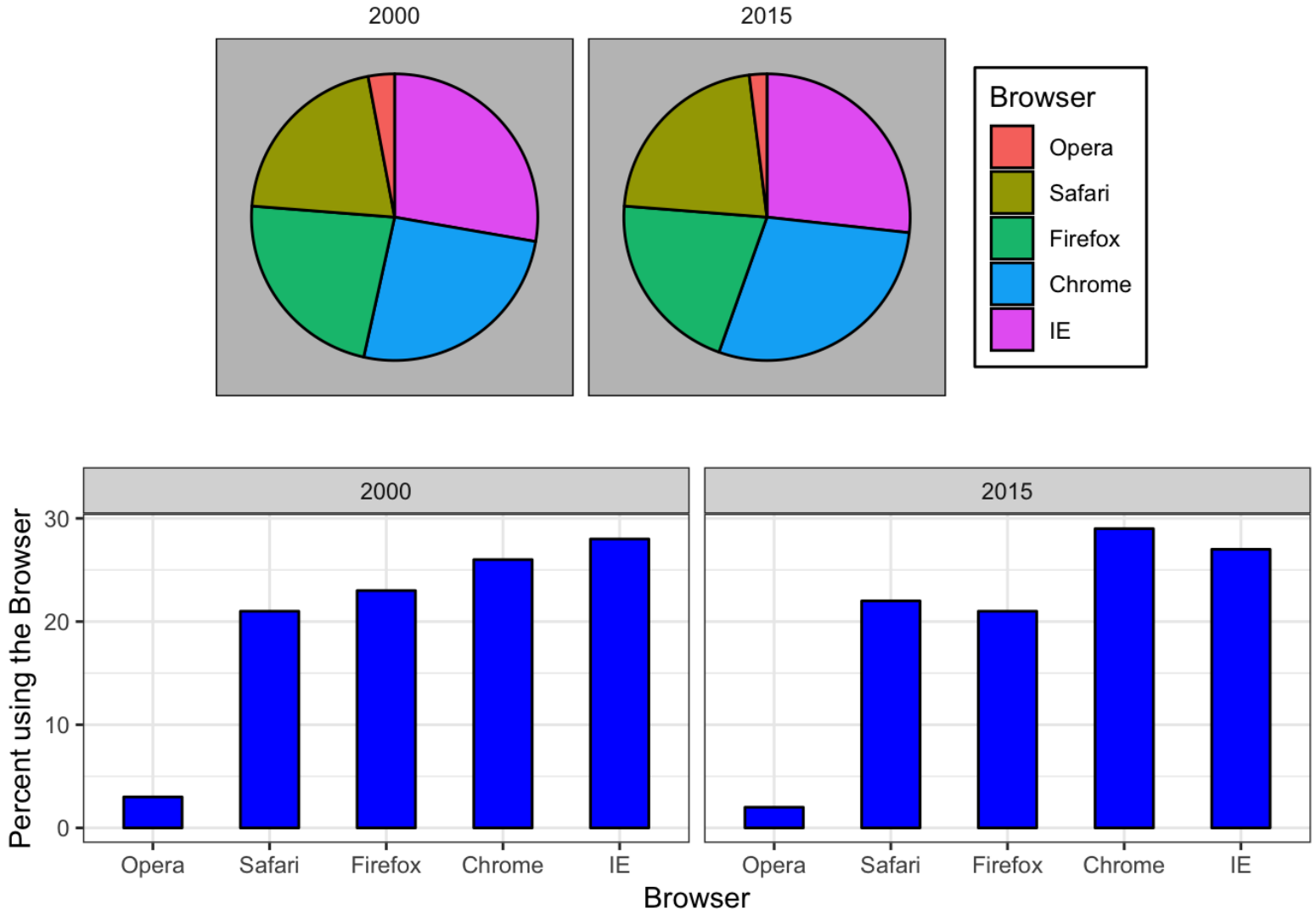
information coded in:
- angles
- areas

Donut chart

information coded in:
- areas

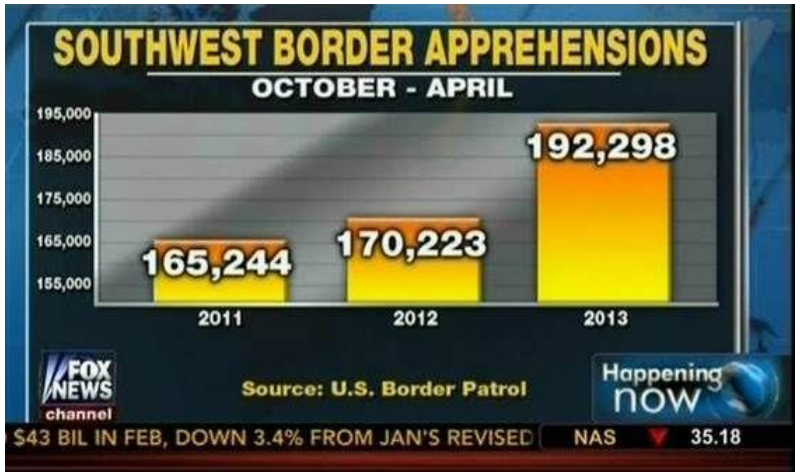


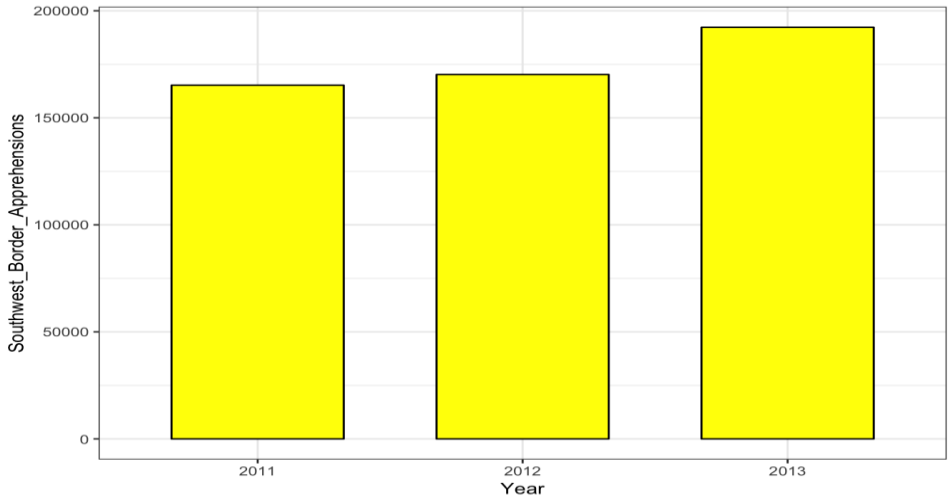
A series of **barplots** allow our brains to easily compare the represented values.
Information coded in **aligned lengths**.

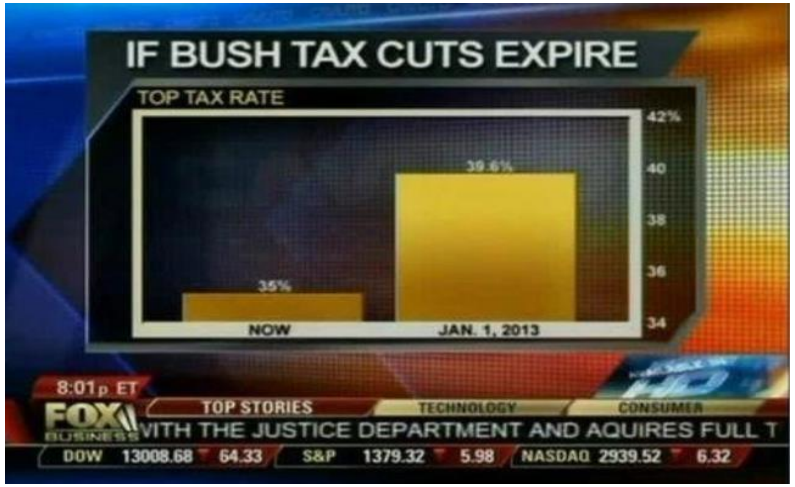


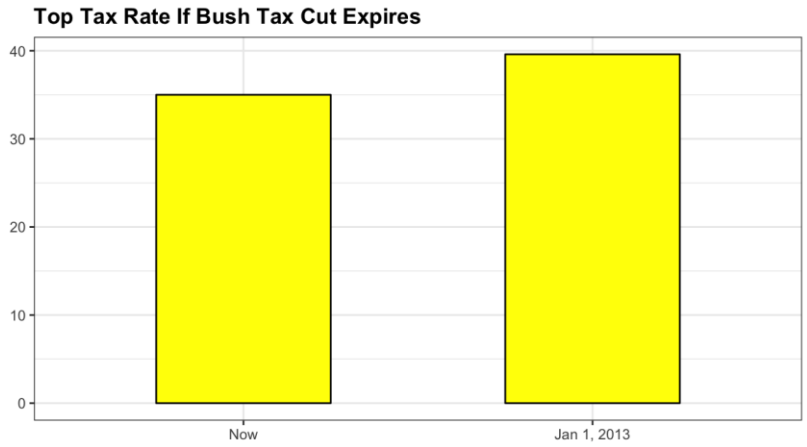
Know when to include zero

When using barplots it is dishonest not to start the bars at 0. This is because, by using a barplot, we are implying **the length is proportional to the quantities being displayed**. By avoiding 0, relatively small differences can be made to look much bigger than they actually are. This approach is often used by politicians or media organizations trying to exaggerate a difference.

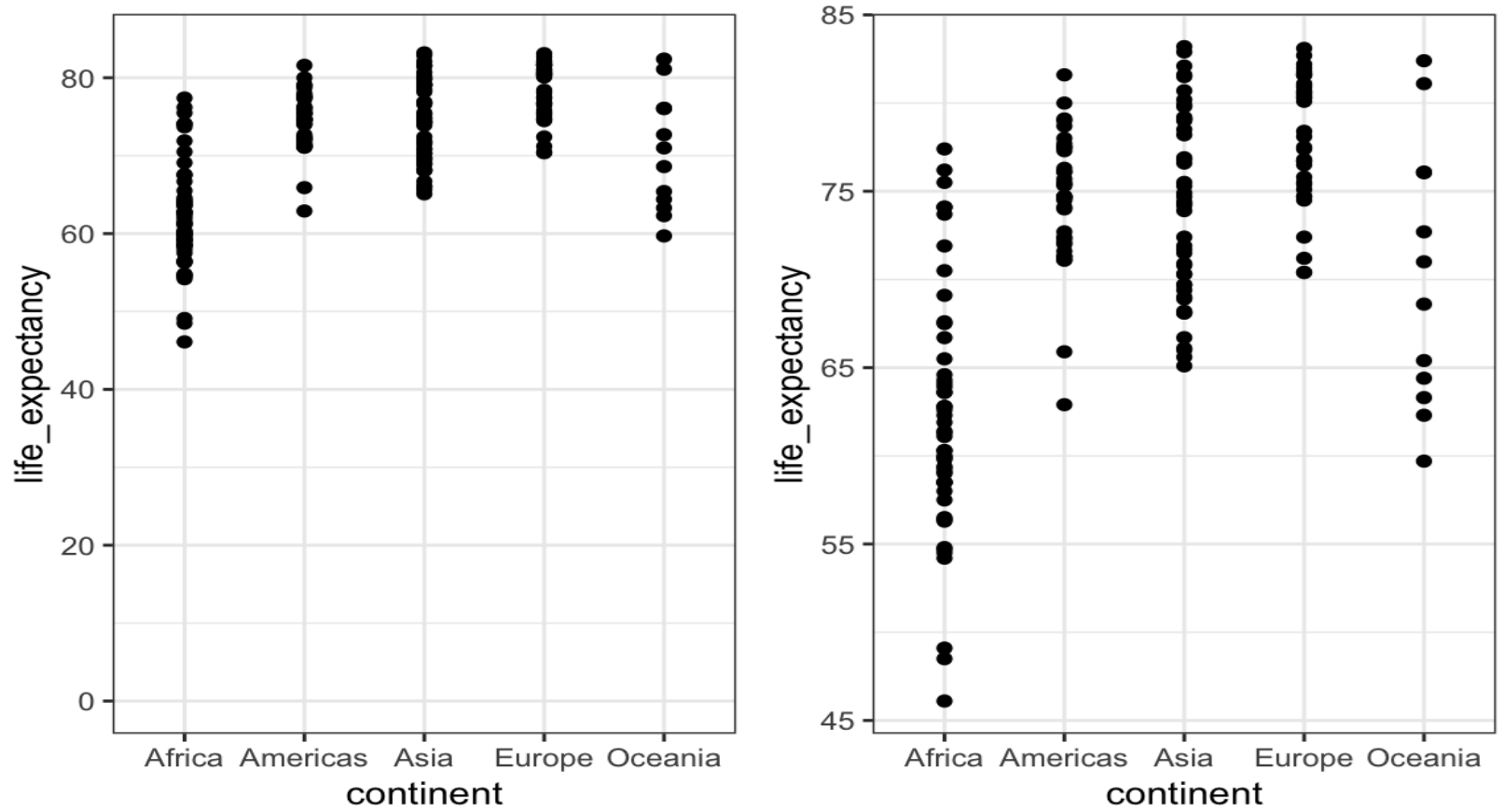








When using position rather than length, it is not necessary to include 0. This is particularly the case when we want to compare differences between groups relative to the variability seen within the groups.

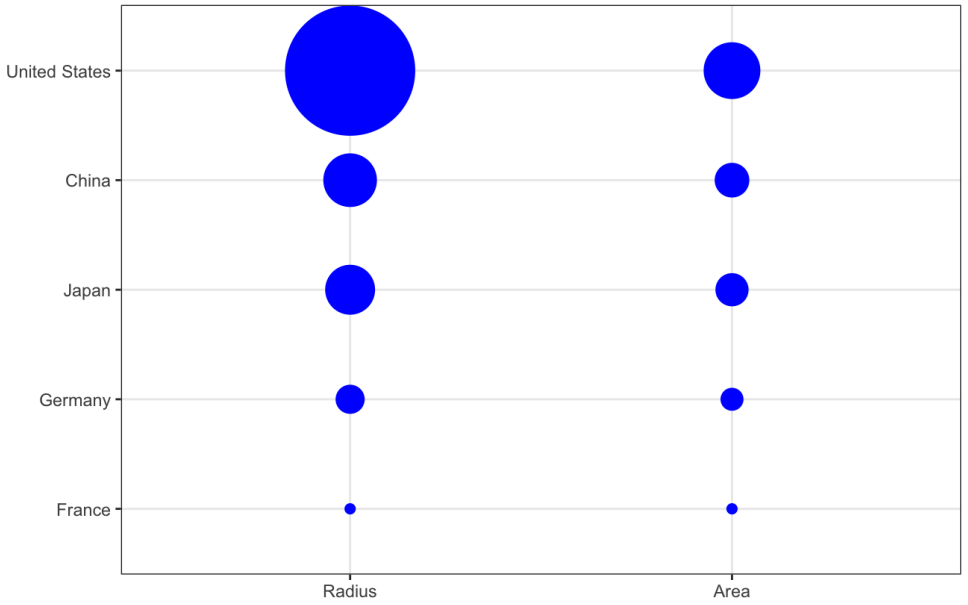


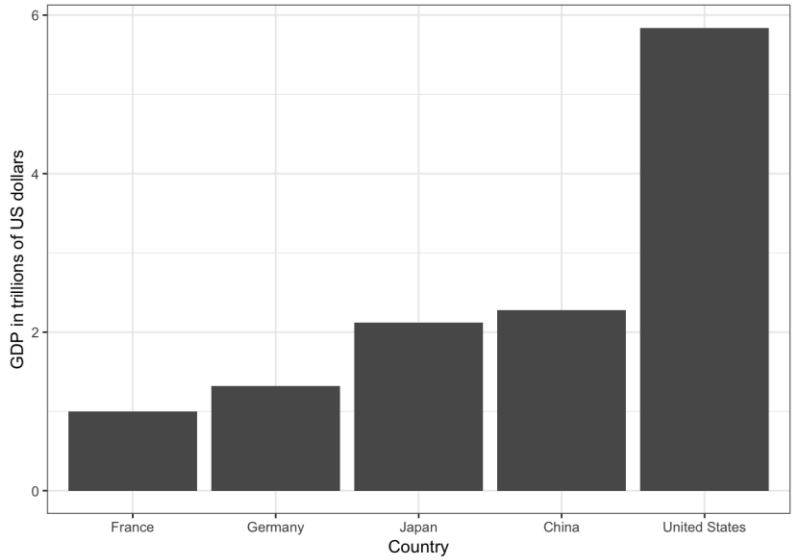
The space between 0 and 43 in the plot on the left adds no information and makes it harder to appreciate the between and within variability. Here, 0 should not be included.

Do not distort quantities

The US appears to have an economy over five times larger than China and over 30 times larger than France. However, when looking at the actual numbers one sees that this is not the case. The actual ratios are 2.6 and 5.8 times bigger than China and France, respectively. Using areas is more honest than using radii.

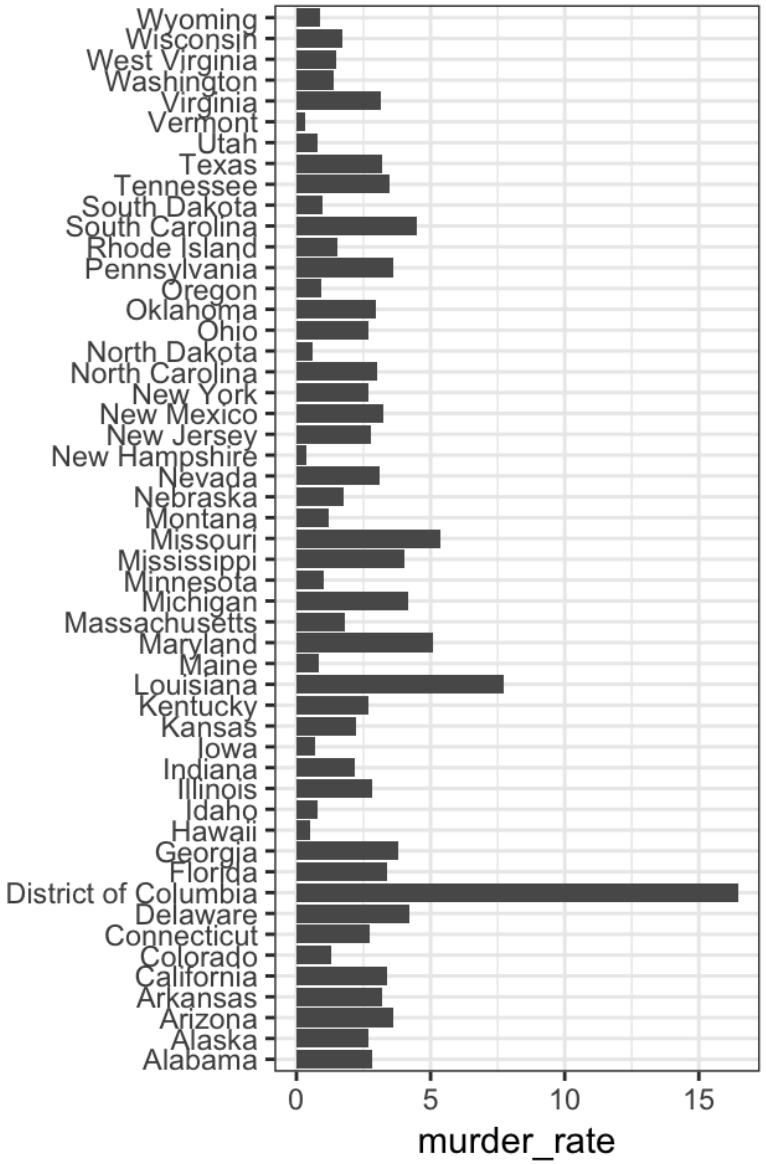
Anyway, circle areas should not be used here. Using lengths is more honest.



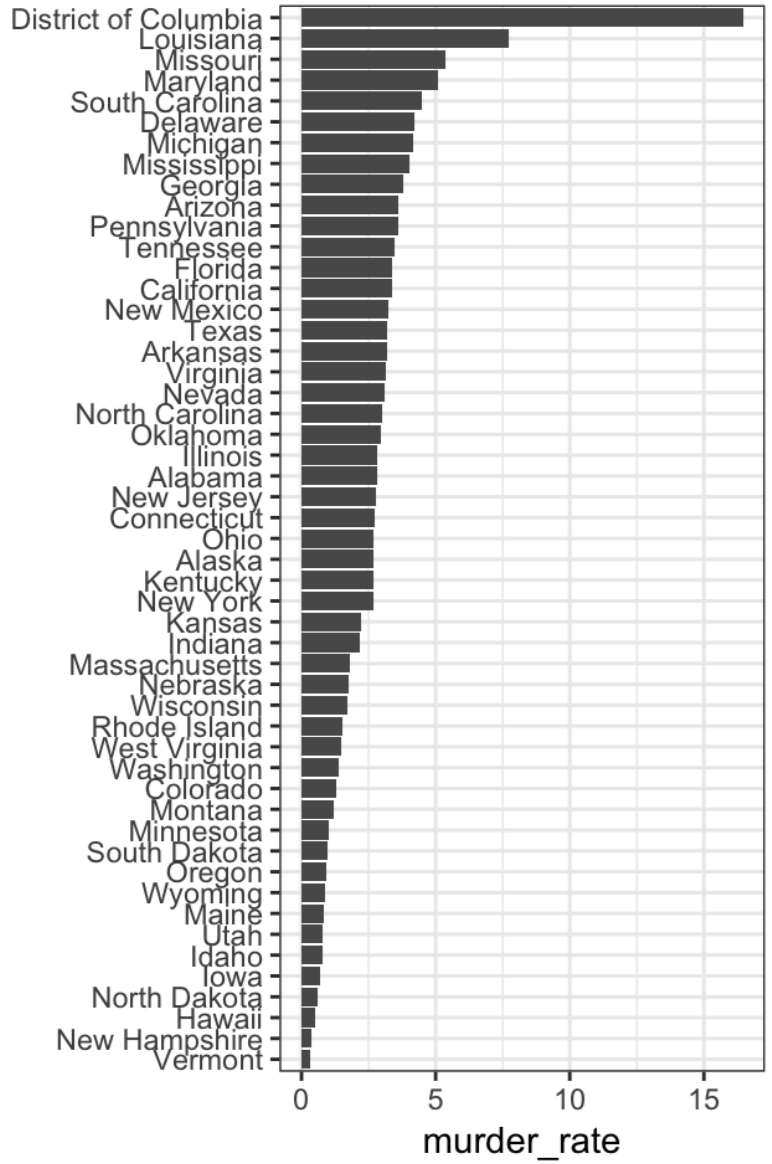


Order unordered factors by a meaningful value

Alphabetical

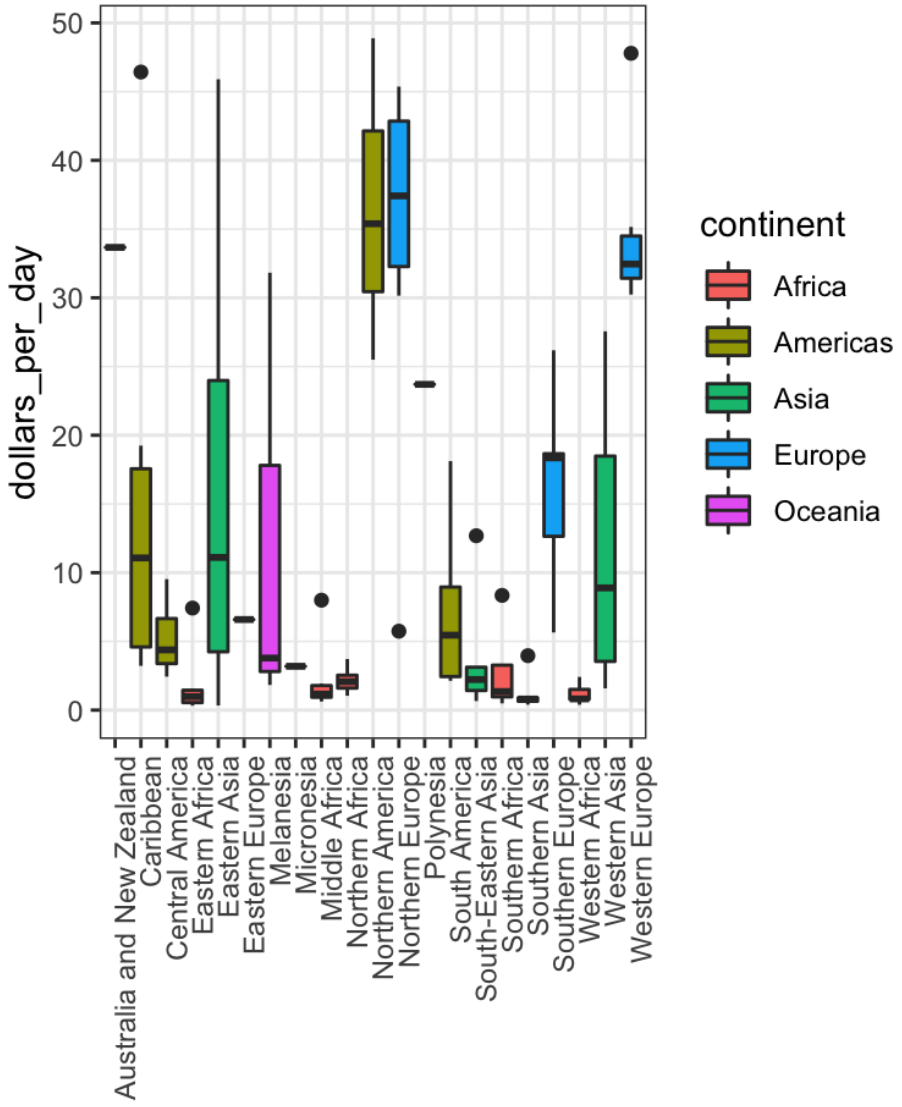


Ordered by murder_rate

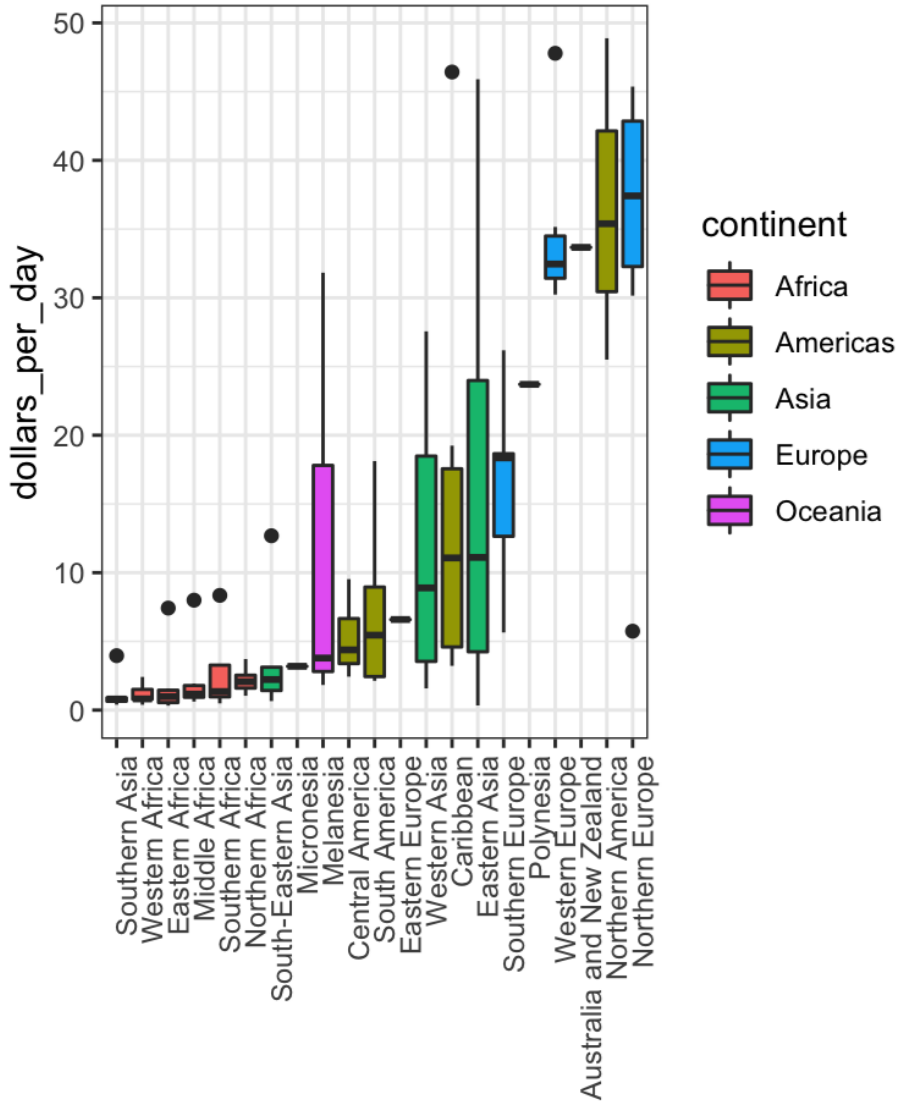


Order unordered factors by a meaningful value

Alphabetical



Ordered by group median



Show the data

These barplots show the average human height and the standard error whiskers.

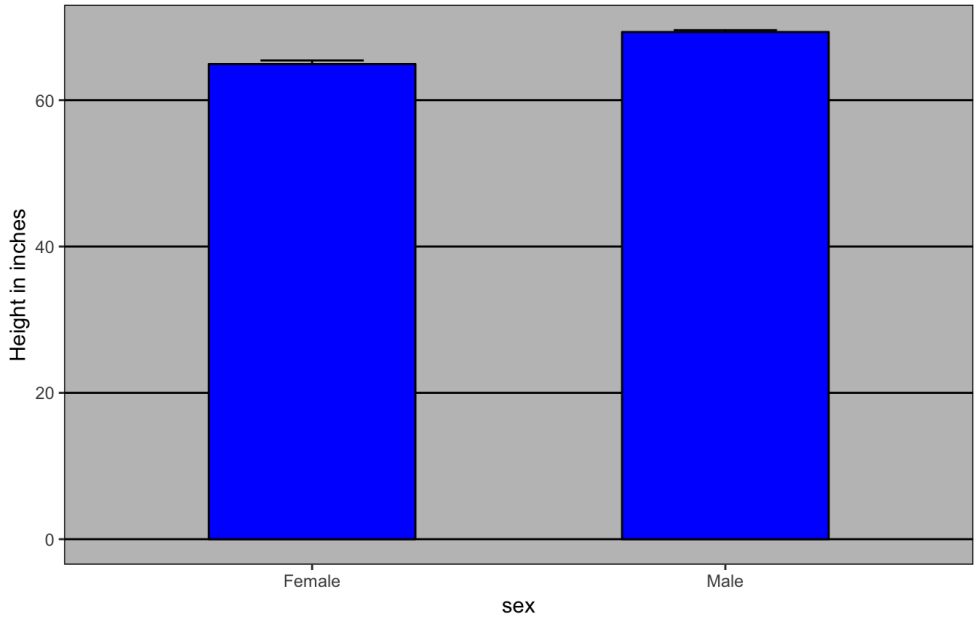
However, they do not show the distribution of the values at all.

The bars go to 0, does this mean there are tiny humans measuring less than one foot?

Are all males taller than the tallest females?

Is there a range of heights?

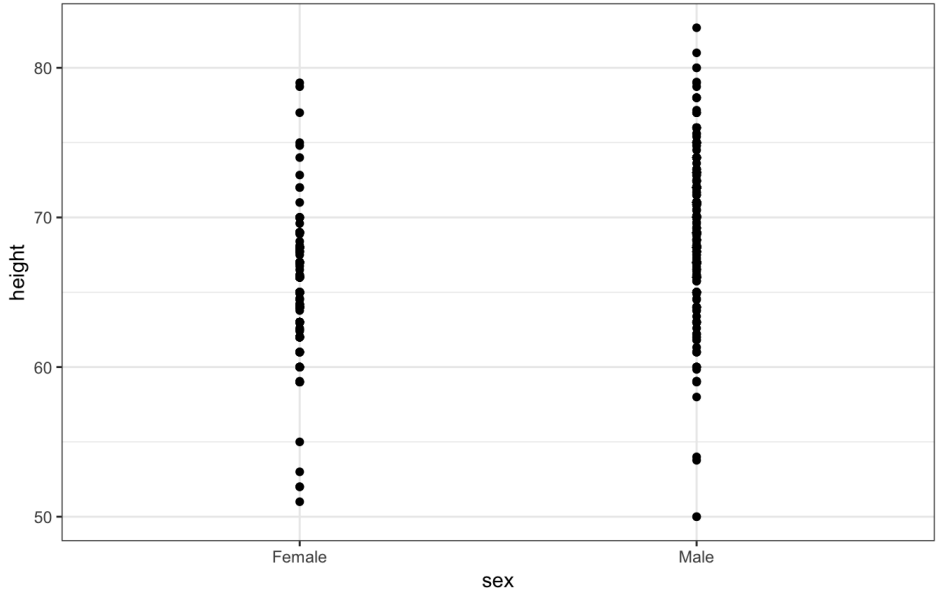
Someone can't answer these questions since we have provided almost no information on the height distribution.



Barplots are useful for showing one number, but not very useful to describe distributions.

This simple ggplot2 code generates more information:

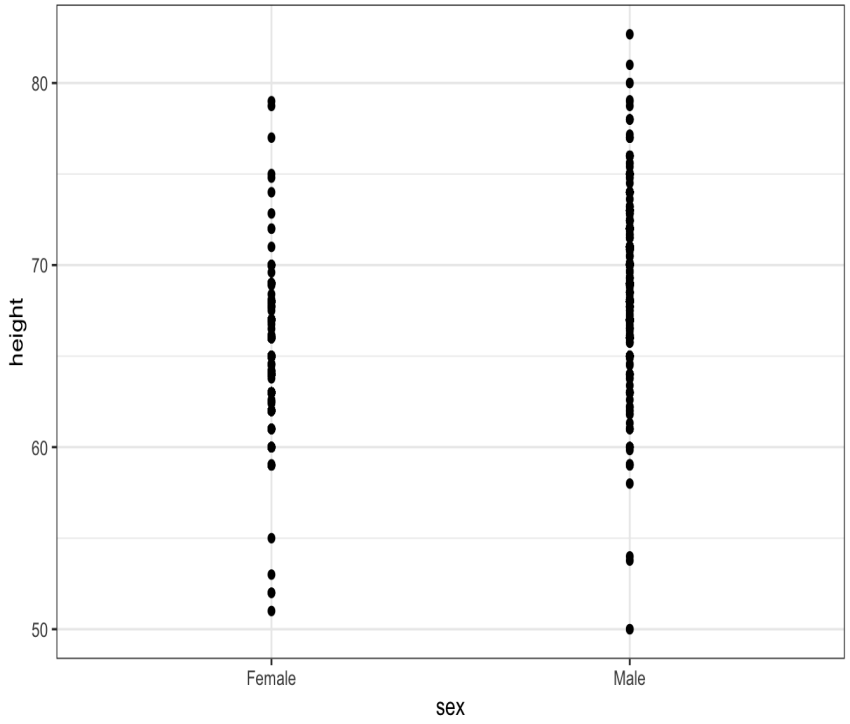
```
heights %>%
  ggplot(aes(sex, height)) +
  geom_point()
```



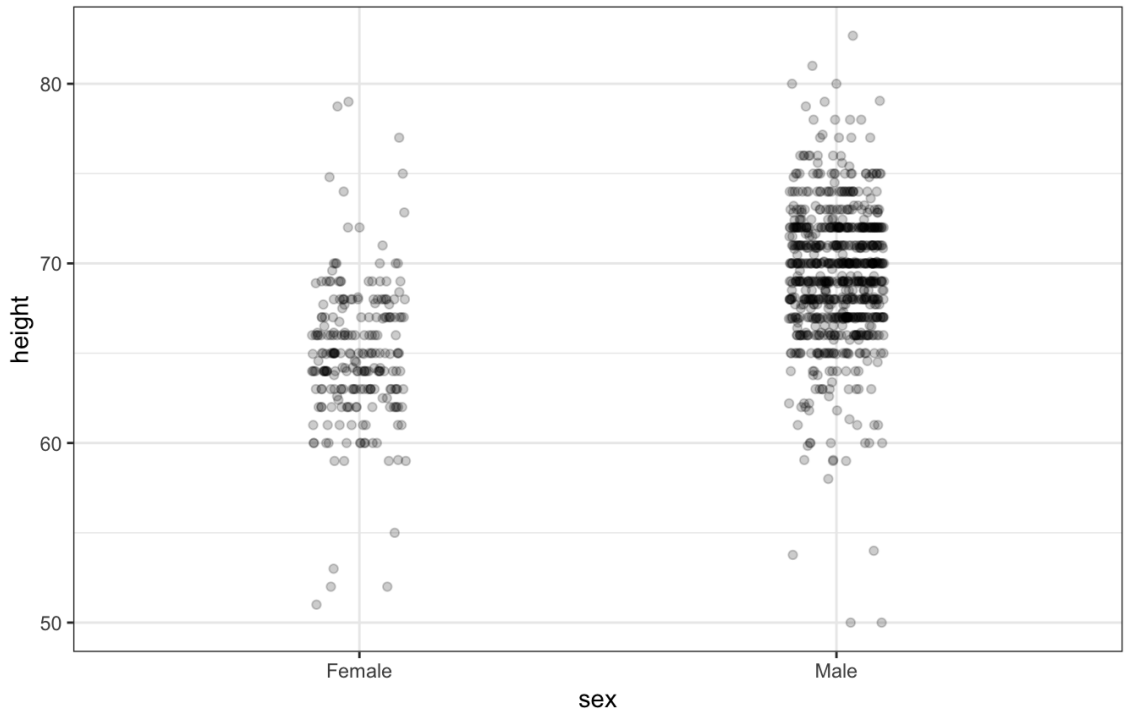
Show the data

Here is the same plot with jitter and alpha blending:

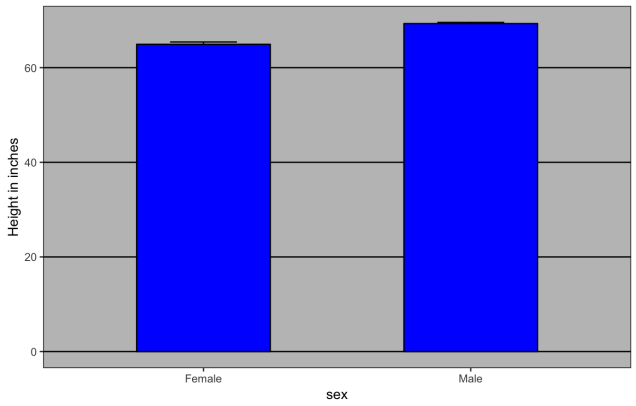
```
heights %>%
  ggplot(aes(sex, height))
```

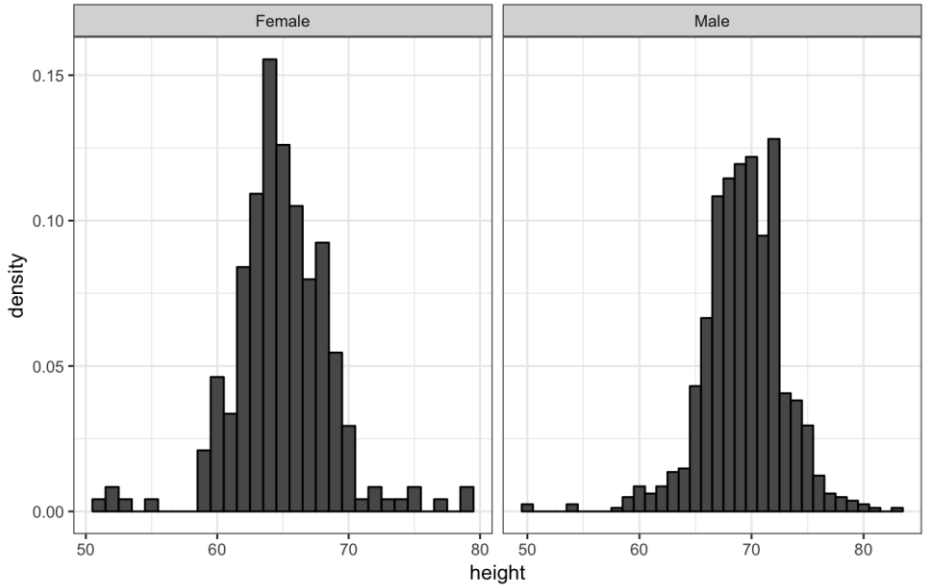


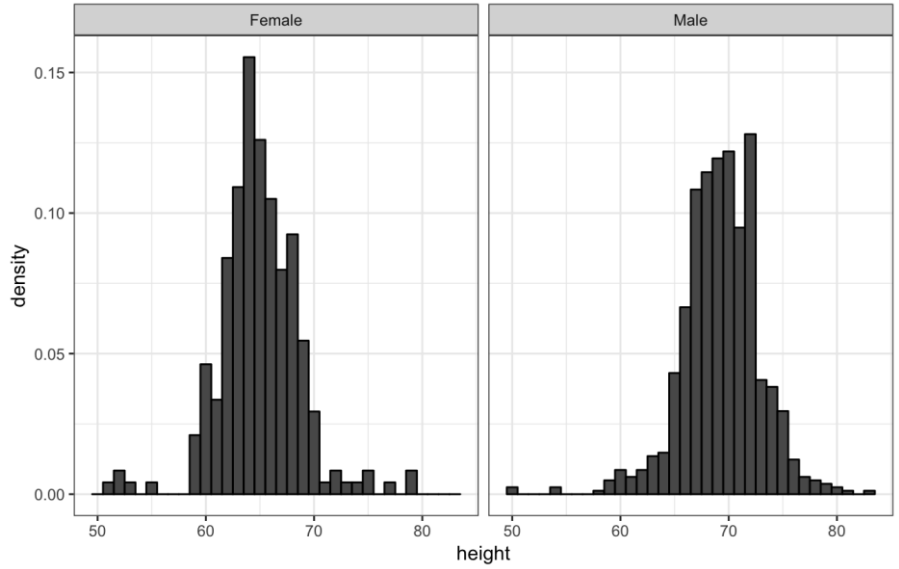

```
heights %>%
  ggplot(aes(sex, height)) +
  geom_jitter(width = 0.1, alpha = 0.2)
```

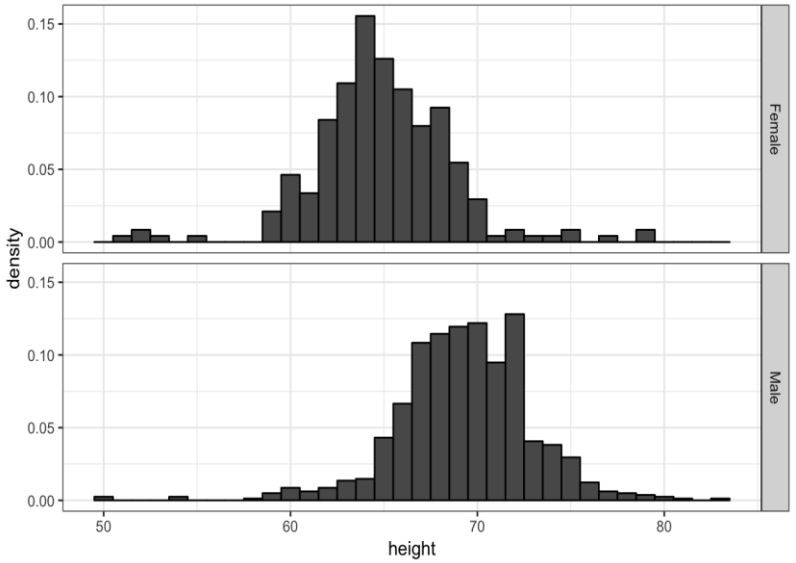


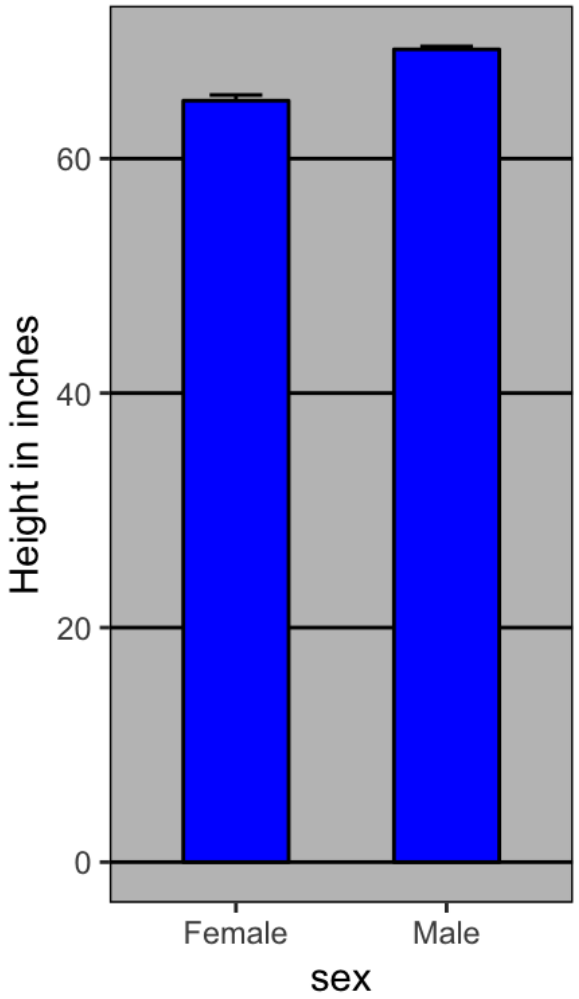
Ease comparisons: Use common axes

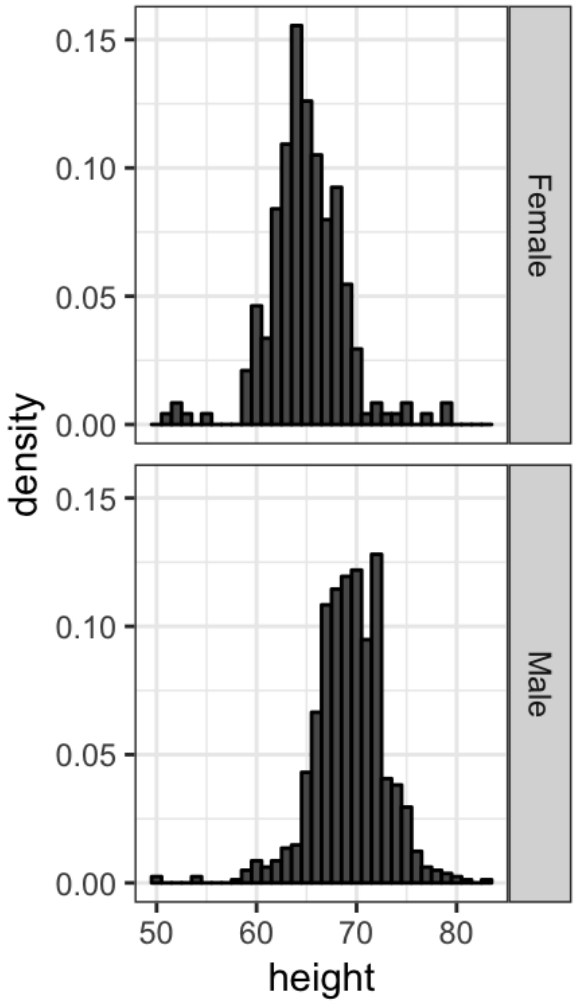


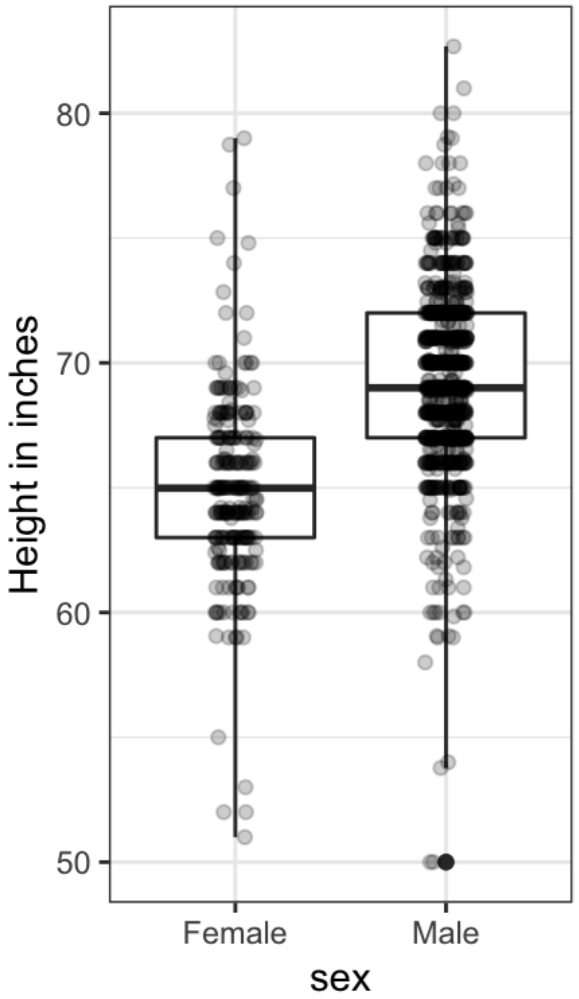




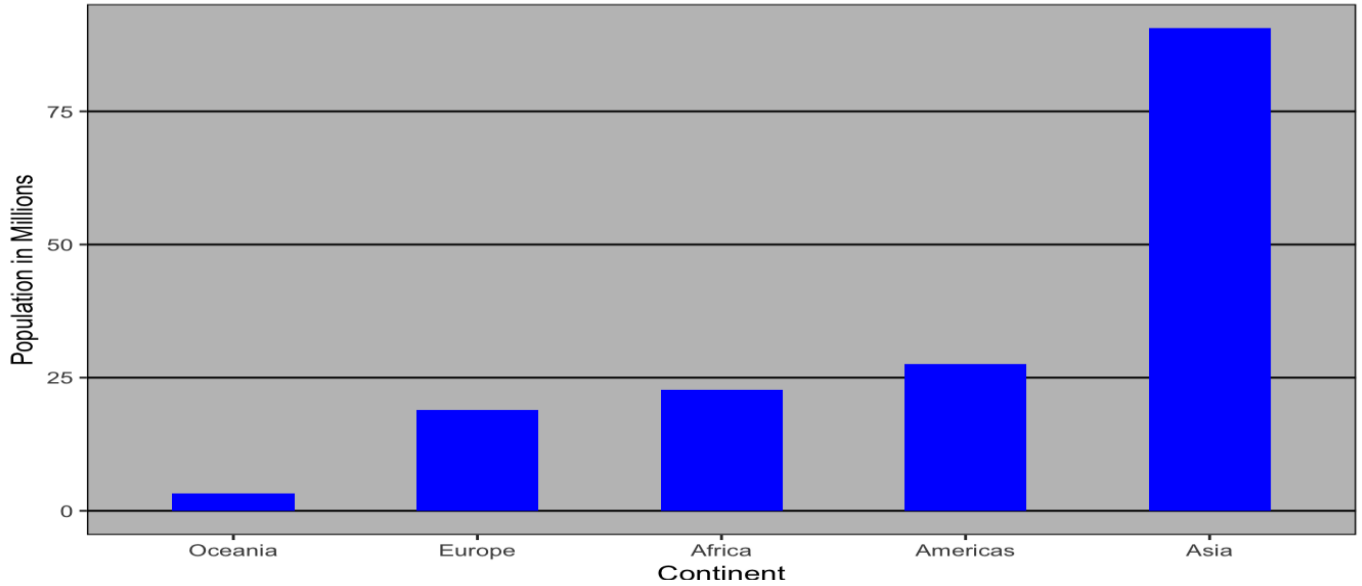


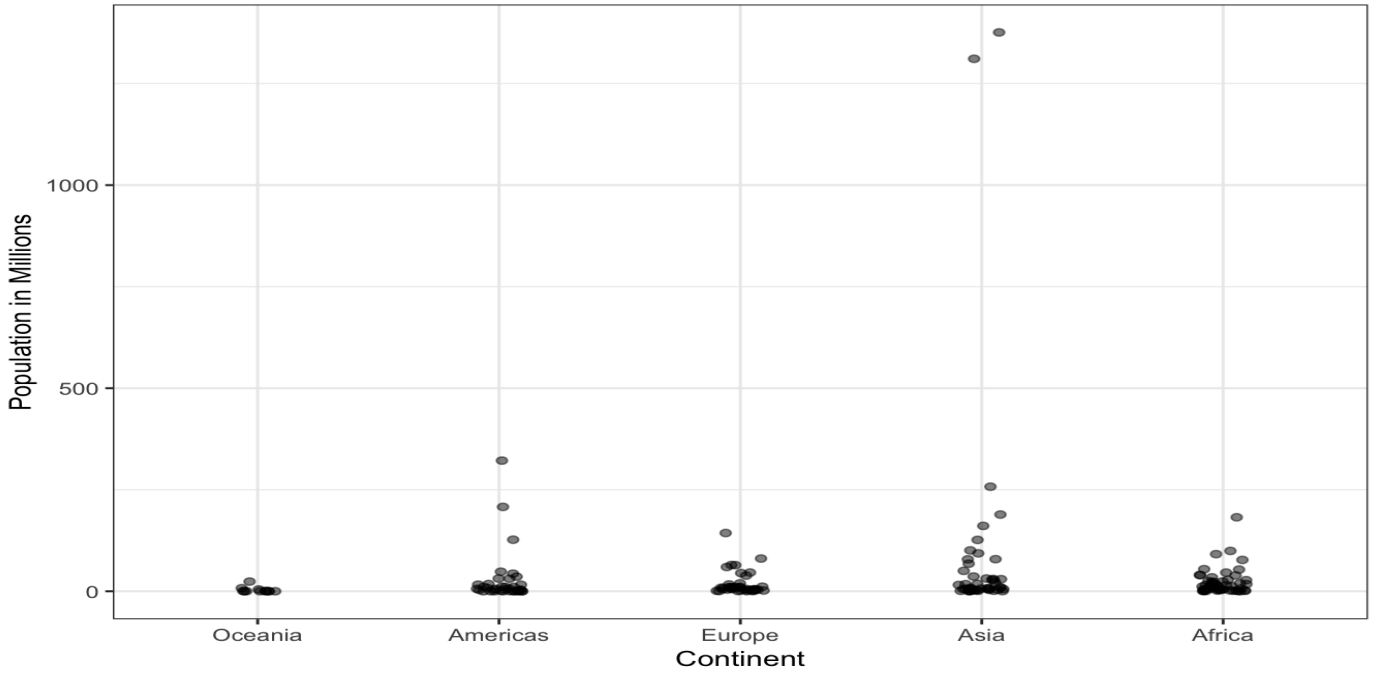

Height in inches
60
40
20
0
Female
Male
sex


density
0.15
0.10
0.05
0.00
Female
Male
50
60
70
80
height

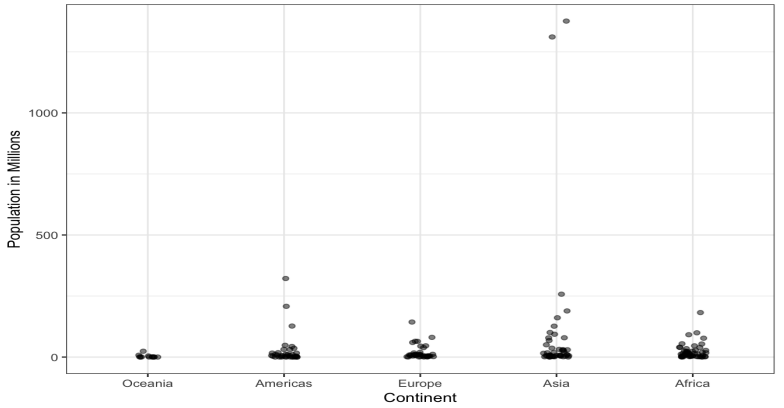

Height in inches
80
70
60
50
Female
Male
sex

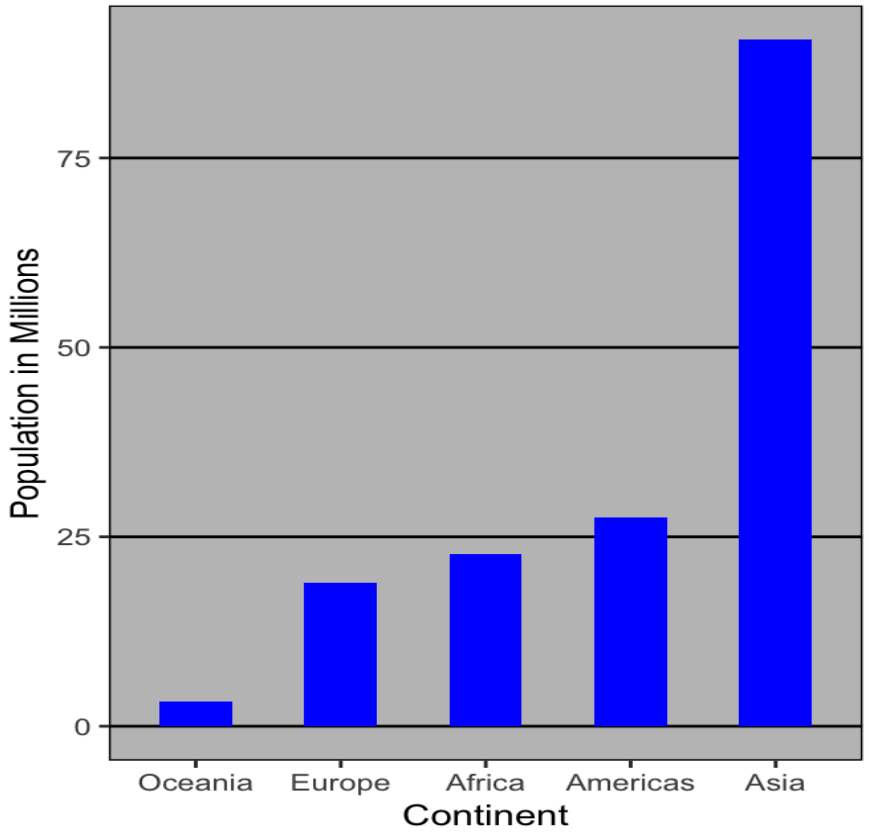
Consider transformations

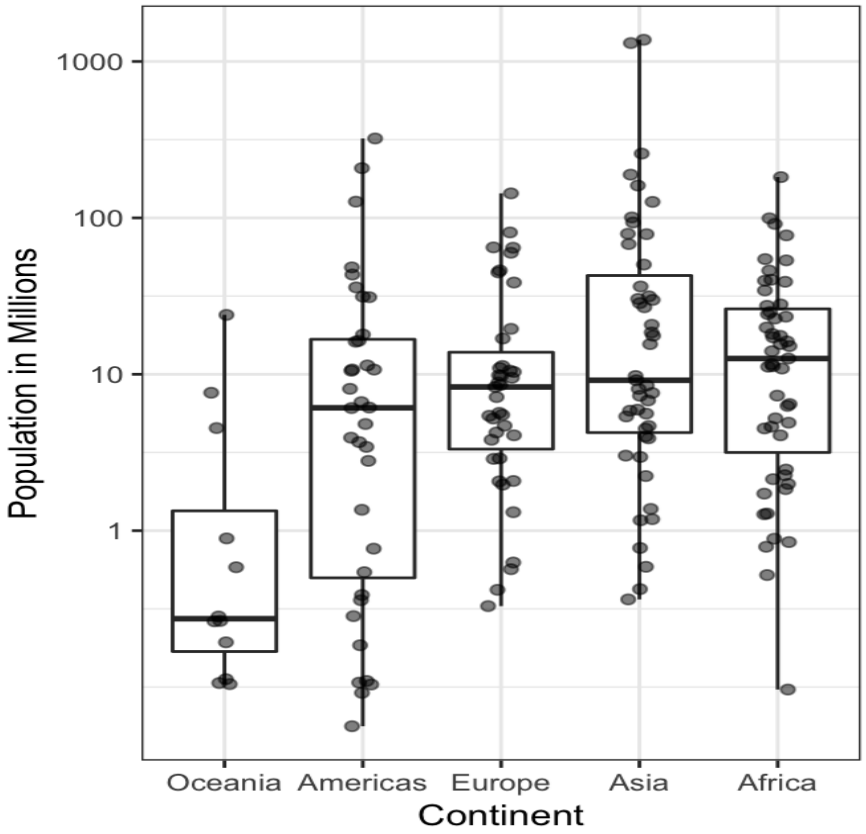




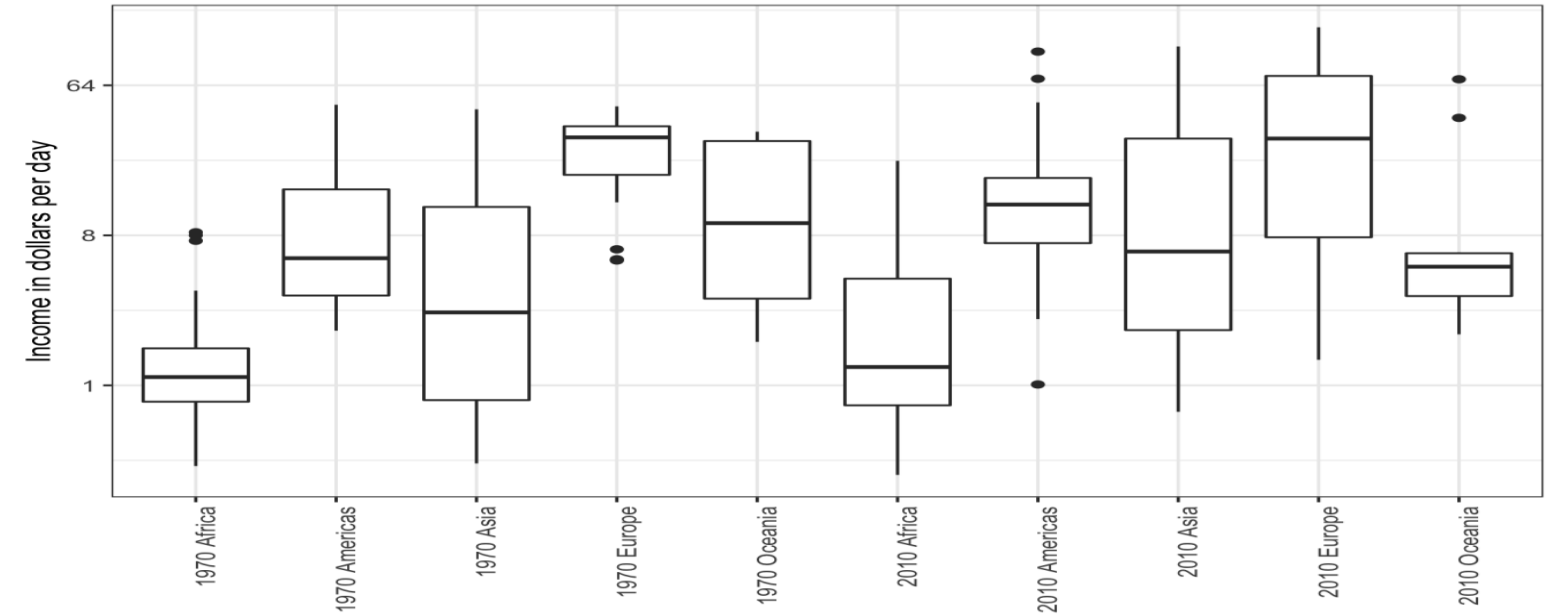
Consider transformations

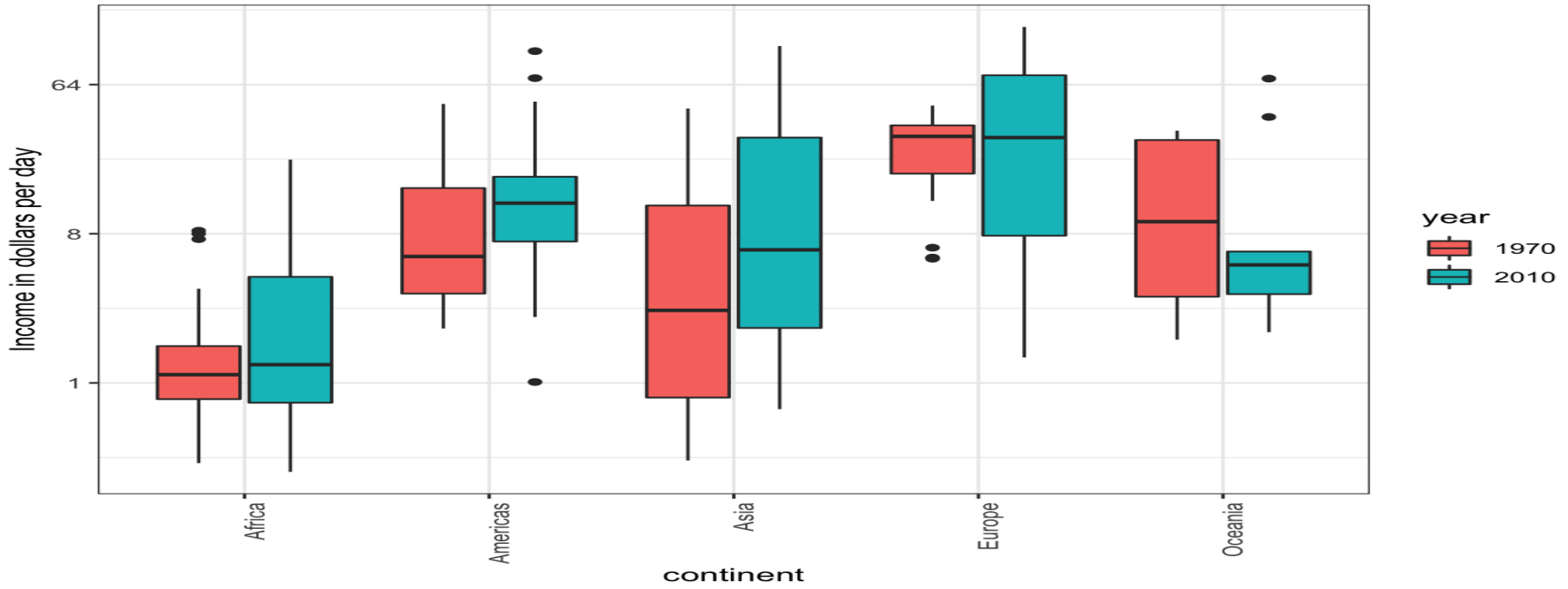






Adjacent visual cues





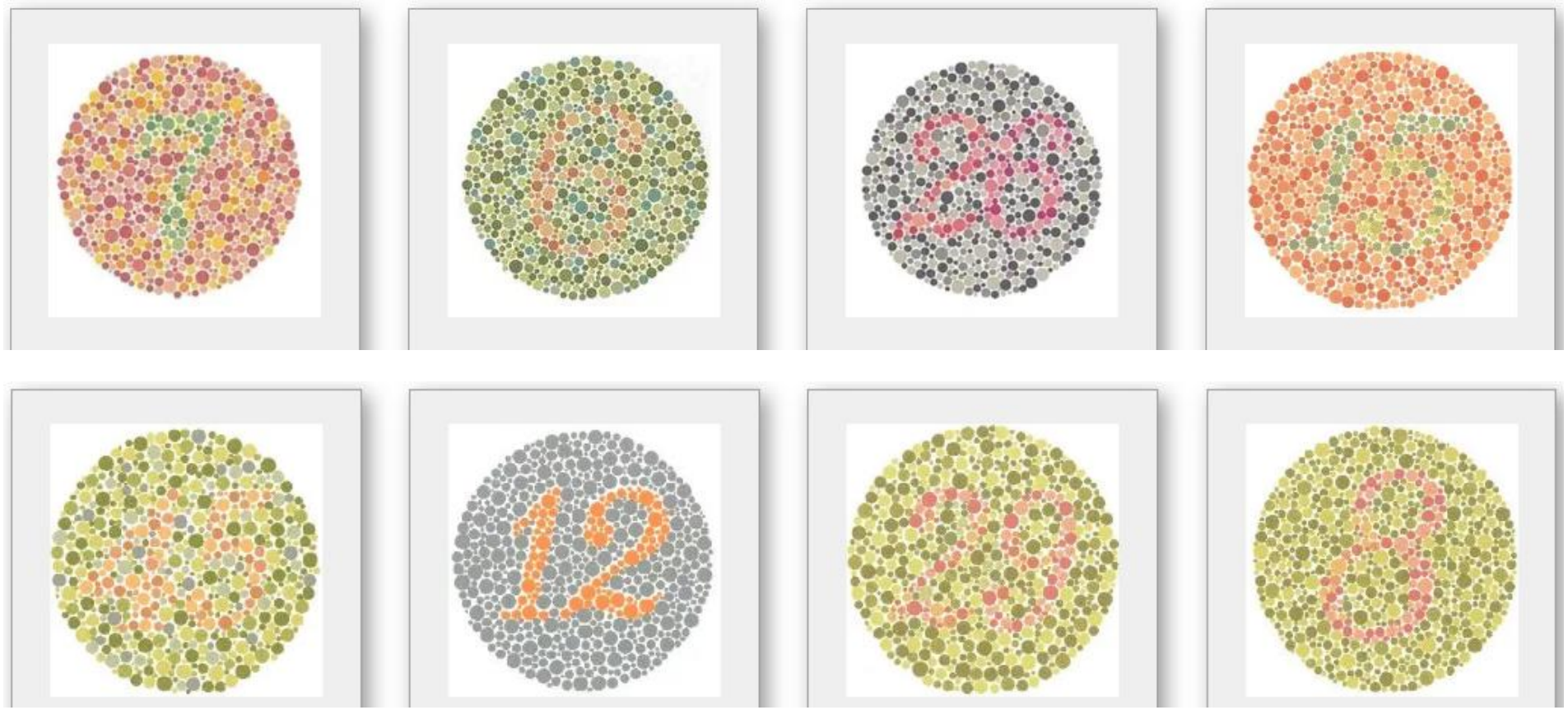
The power and importance of choosing the colour palette.

- The big Lebowski, 1998
- The shinning, 1980
- Pulp Fiction, 1994
- Midsommar, 2019

https://www.esquire.com/

The power and importance of choosing the colour palette.

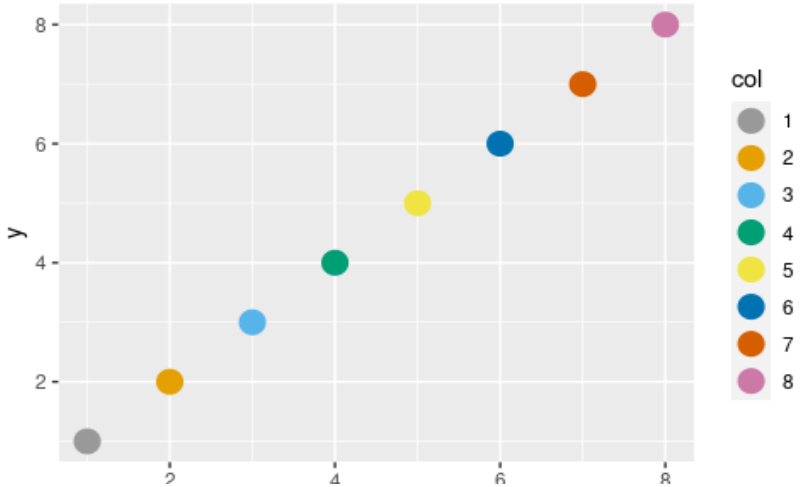
Colour blindness affects 8% of men and ~0.5% of women, approximately.



Think of the colour blind

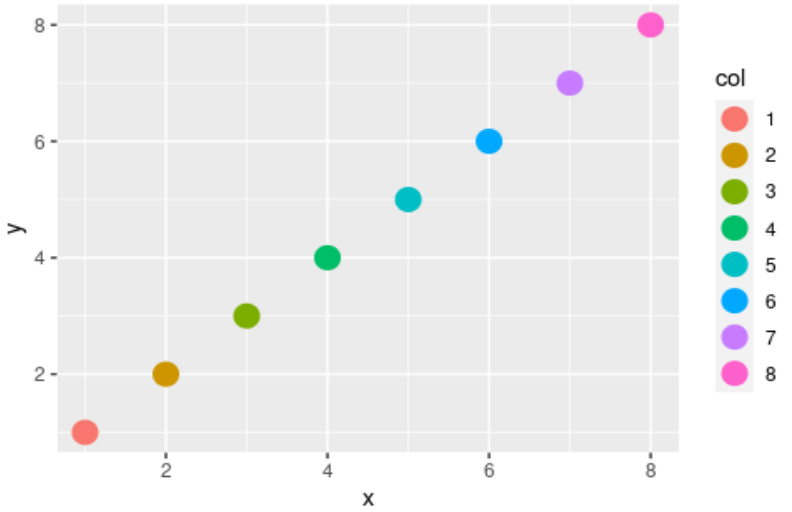
About 10% of the population is colour blind. Unfortunately, the default colours used in ggplot are not optimal for this group. However, ggplot does make it easy to change the colour palette used in plots. Here is an example of how we can use a colour blind friendly palette. There are several resources that help you select colours, for example: https://bconnelly.net/posts/creating_colorblind-friendly_figures/

```
color_blind_friendly_cols <- c("#999999", "#E69F00",
"#56B4E9","#009E73", "#F0E442", "#0072B2","#D55E00",
"#CC79A7")

p1 <- data.frame(x=1:8, y=1:8, col = as.character(1:8)) %>%
  ggplot(aes(x, y, color = col)) +
  geom_point(size=5)

p1 + scale_color_manual(values=color_blind_friendly_cols)
```



This is using the default colour palette of ggplot2

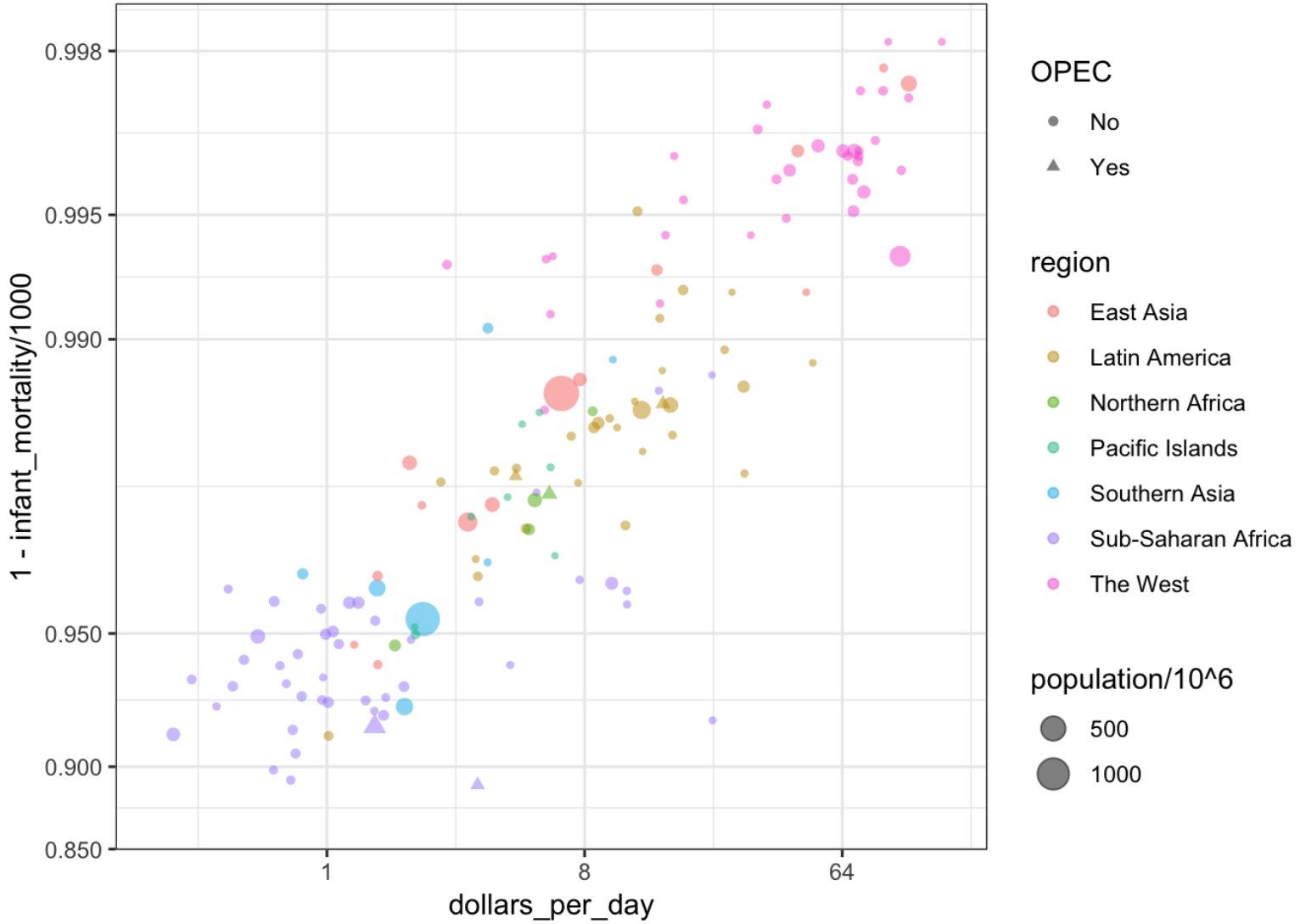
```
data.frame(x=1:8, y=1:8, col = as.character(1:8)) %>%
  ggplot(aes(x, y, color = col)) +
  geom_point(size=5)
```



Encoding a third variable

We can encode categorical variables with colour hues, and shapes.

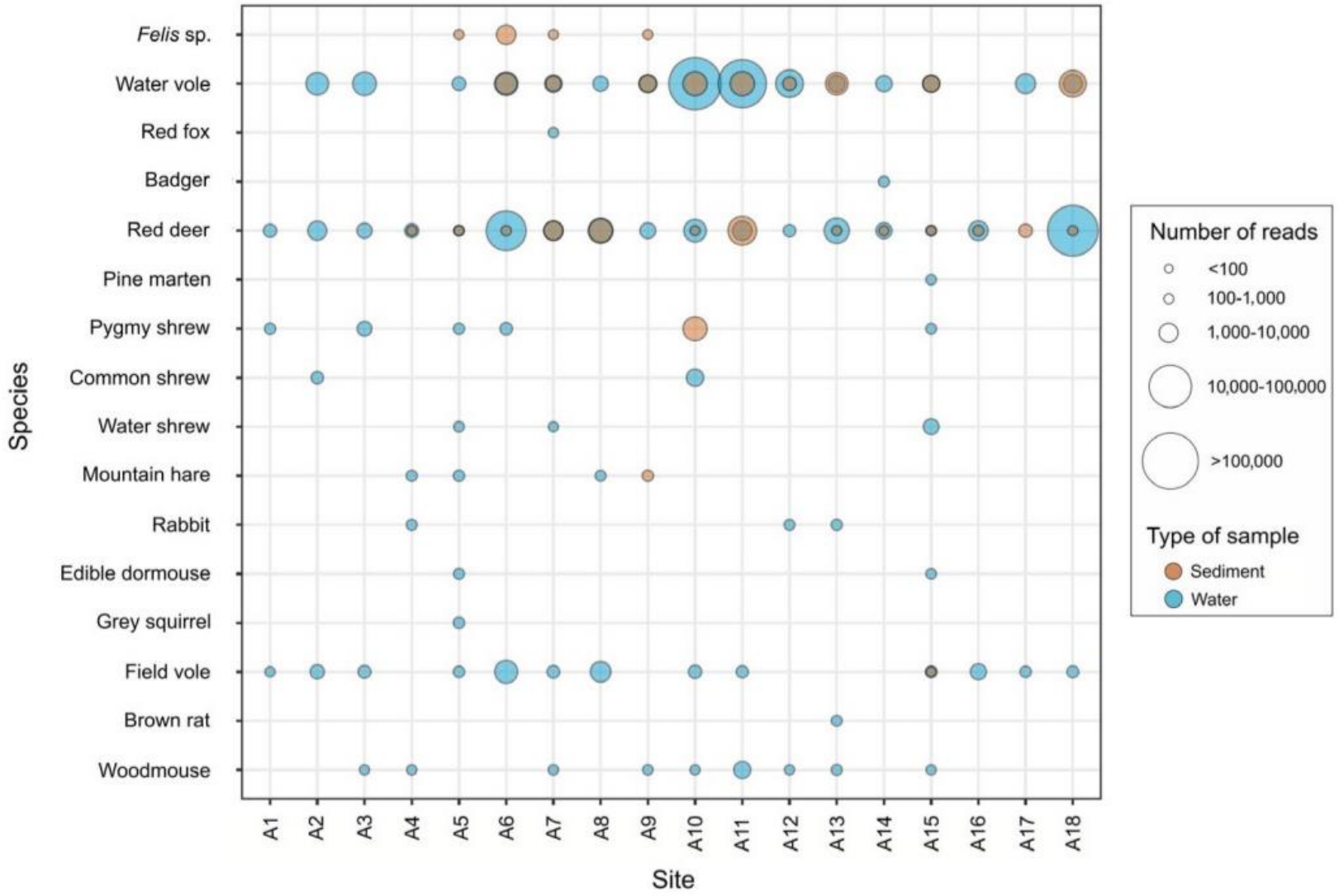
Quantitative variables (numerical or ordered factors) can be encoded by point sizes.



Encoding a third variable

Bubble plots use point sizes to encode quantitative variables

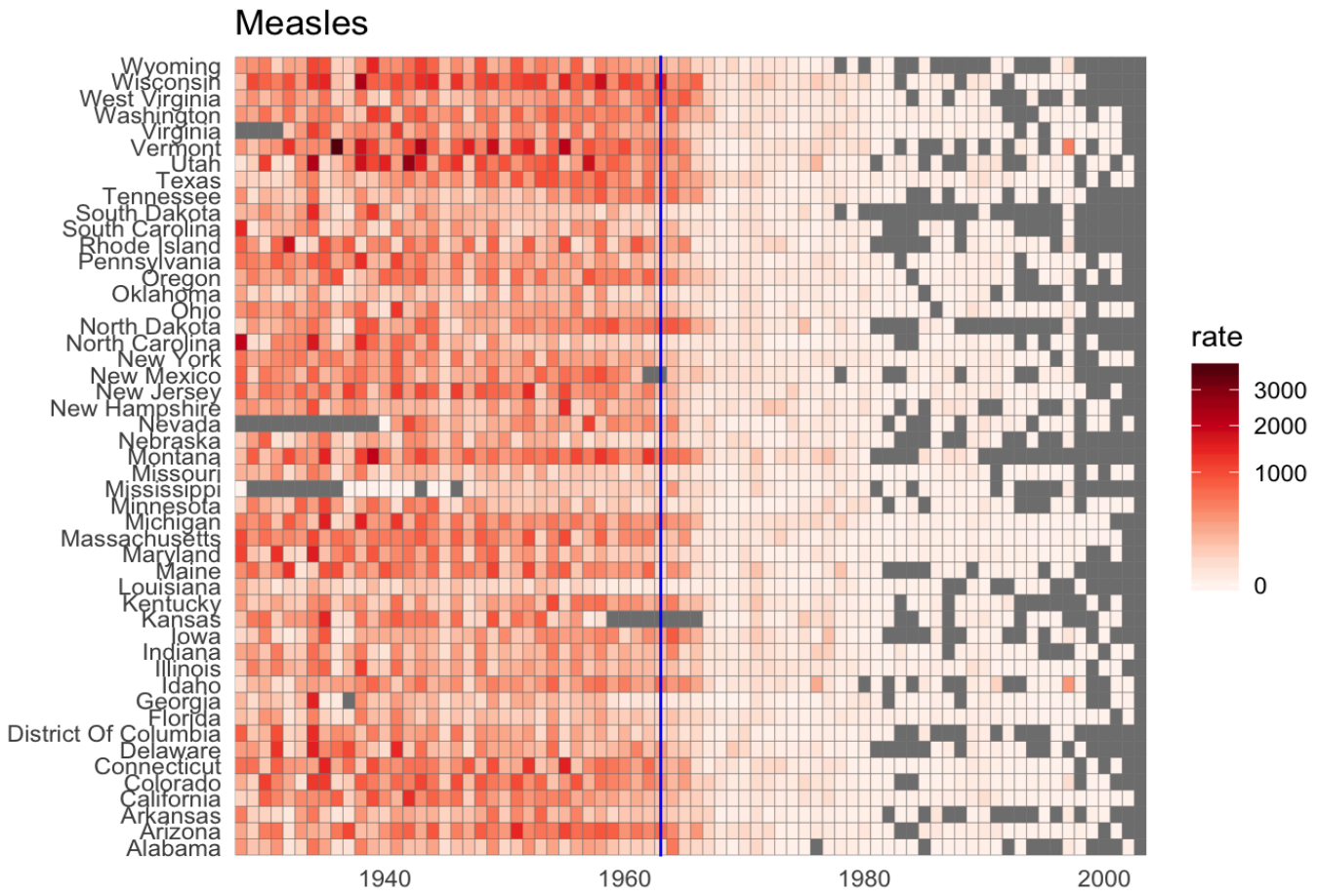
The X-Y canvas is made up by a grid of two unordered factors in this case.
But you can have quantitative axes (as in the previous example)



Encoding a third variable

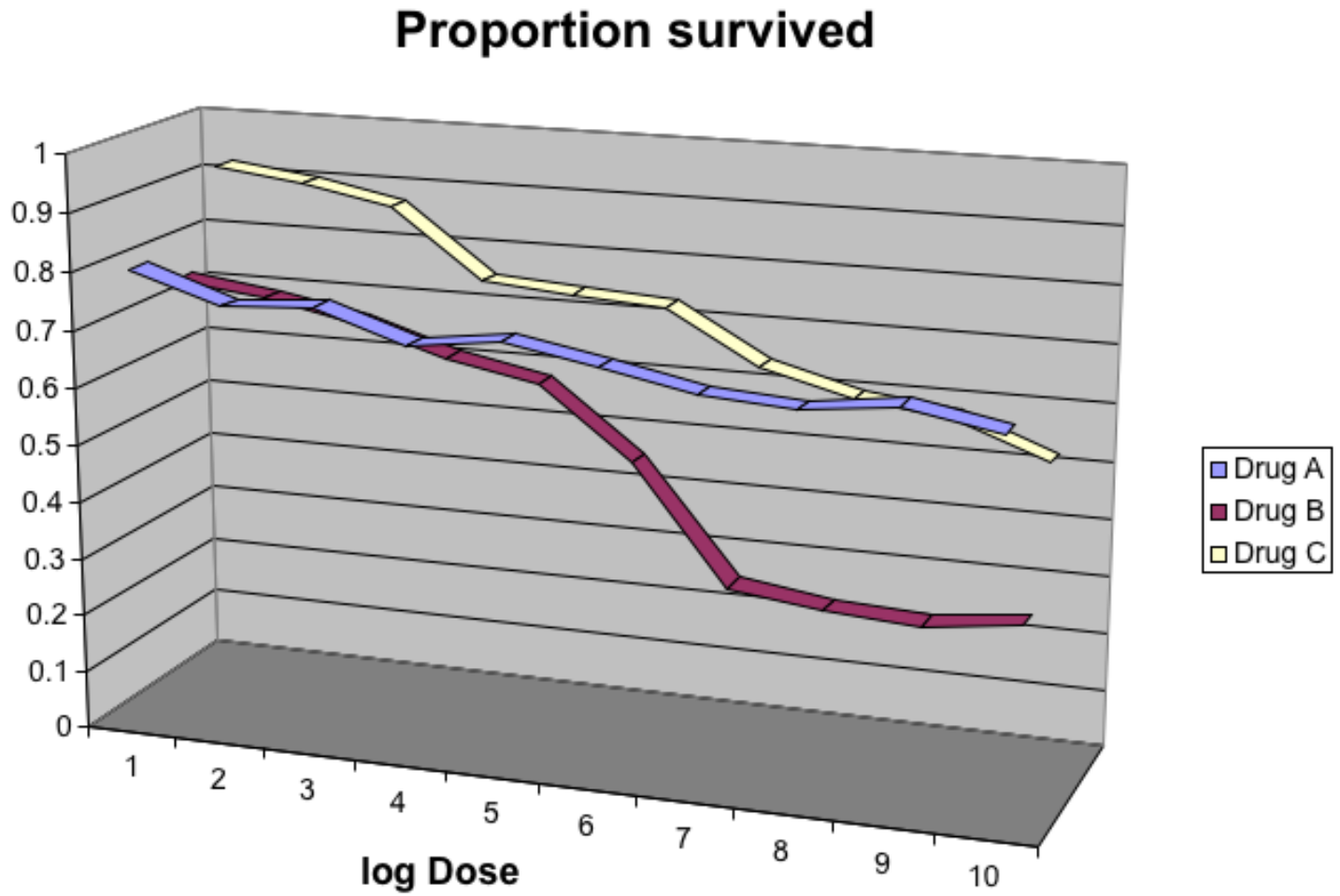
For quantitative variables we can use colour, intensity or size. A **heatmap** uses colours or colour intensities.

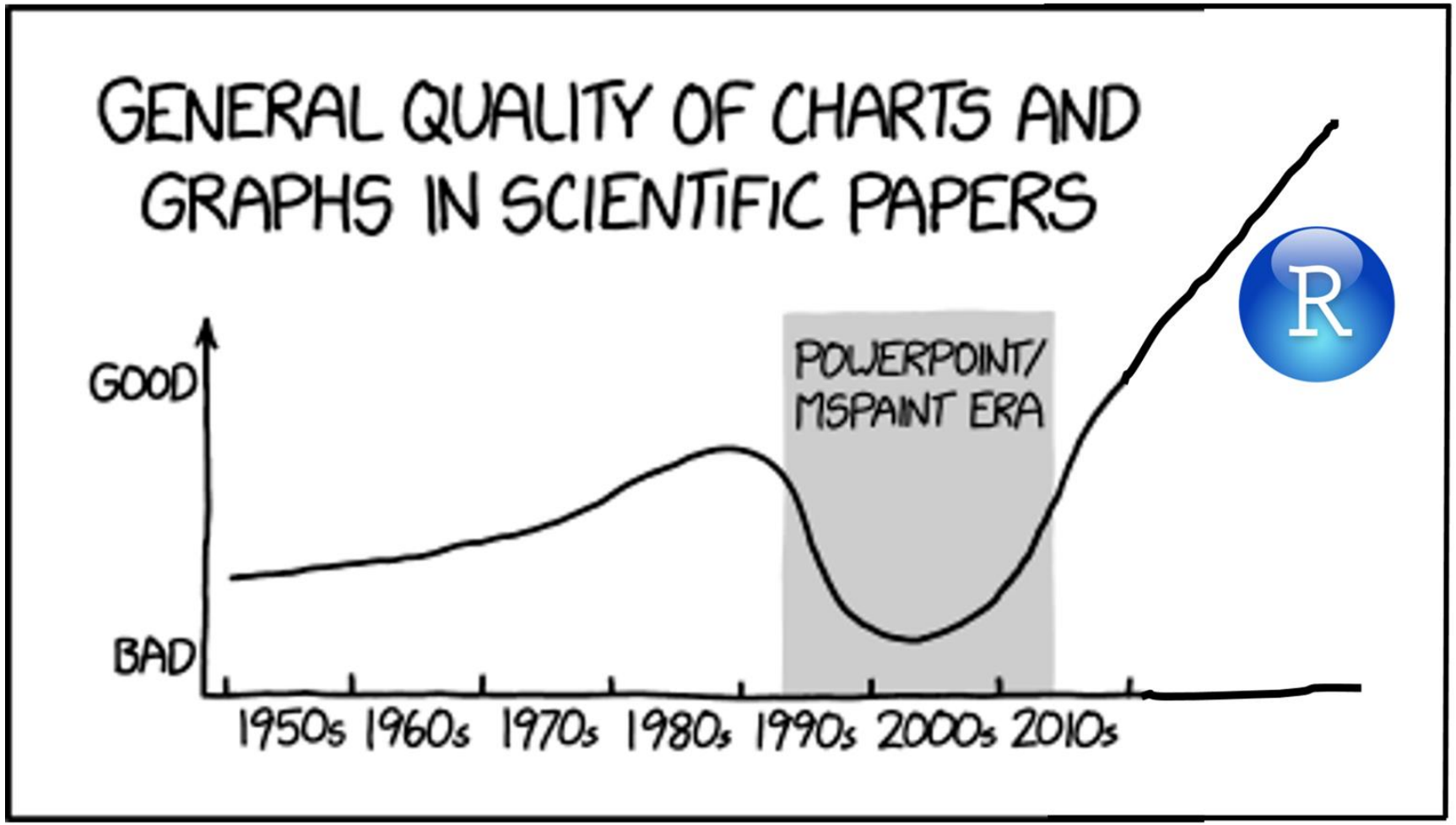
However, using colour to represent a quantity is not optimal. Position and length would be better cues. If we are willing to lose state information, we could make a version of this plot showing the values with position.



Avoid pseudo-3D plots

This plot shows three variables: dose, drug type and survival. Although your screen is flat and two dimensional, the plot tries to imitate three dimensions and assigns a dimension to each variable. The extra dimension is not only unnecessary, it makes what should be an easy plot to decipher nearly impossible to draw conclusions from.




GENERAL QUALITY OF CHARTS AND GRAPHS IN SCIENTIFIC PAPERS
GOOD
BAD
POWERPOINT/ MSPAINT ERA
R
1950s 1960s 1970s 1980s 1990s 2000s 2010s

The philosophy in ggplot2

All **plots** are composed of two components: the **data**, and the **mapping**, the description of how the data's variables are mapped to aesthetic attributes.

There are five mapping components:

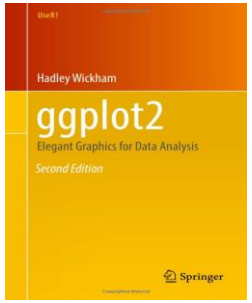
A **layer** is a collection of **geometric elements** and **statistical transformations**. Geometric elements, is what you actually see in the plot: points, lines, polygons, etc. Statistical transformations summarise the data: e.g. binning and counting observations to create a histogram, or fitting a linear model.

Scales map values in the data space to values in the aesthetic space. This includes the use of colour, shape or size. Scales also draw the legend and axes.

A **coordinate system**, describes how data coordinates are mapped to the plane of the graphic. It also provides axes and gridlines to help read the graph.

A **facet** specifies how to break up and display subsets of data as small multiples.

A **theme** controls the finer points of display, like the font size and background colour.



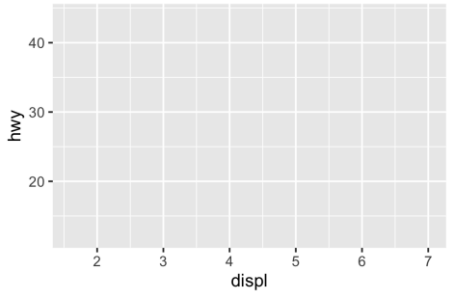
Everything you need to know about ggplot2:
https://ggplot2-book.org/index.html

The philosophy in ggplot2

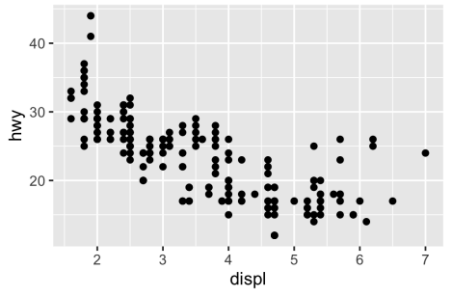
ggplot2 is designed to work iteratively. You start with a layer that shows the raw data. Then you add layers of annotations and/or statistical summaries.

Build a plot layer by layer

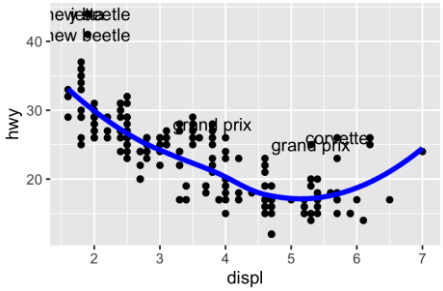
```
p <- ggplot(mpg, aes(displ, hwy))
```




```
p <- p + geom_point()
```



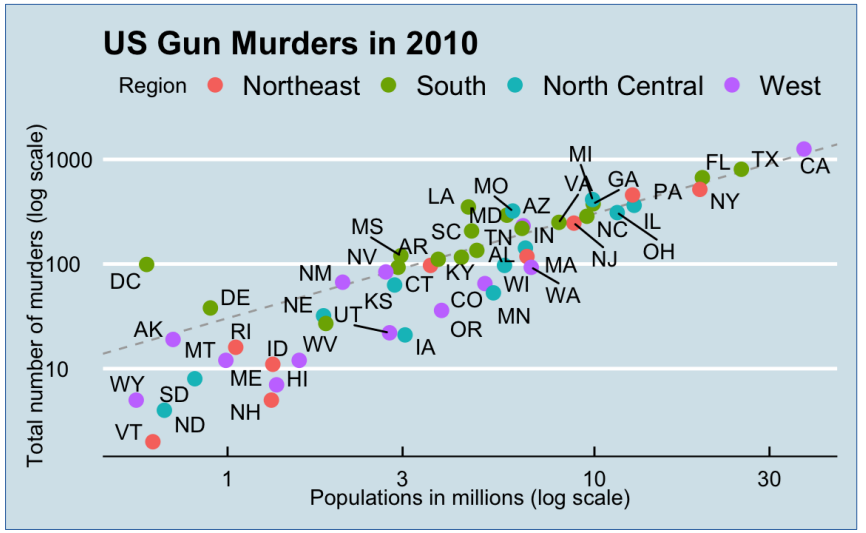

```
p <- p + geom_point() +
  geom_line(data = grid, colour = "blue", size = 1.5) +
  geom_text(data = outlier, aes(label = model))
```



https://ggplot2-book.org/layers.html

Components of a ggplot

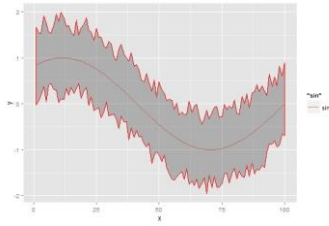
```
ggplot(data = DATA,
       mapping = aes(MAPPING)) +
       geom_1() +
       geom_2() +
  … +
       stats() +
       labs() +
       scales() +
       theme()
       ...
```

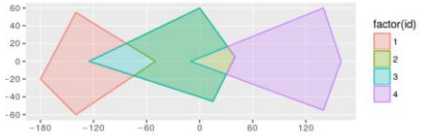


Geoms (1)

Graphical primitives:

geom_blank(): display nothing.
geom_point(): points.
geom_path(): paths.
geom_ribbon(): ribbons, a path with vertical thickness.
geom_segment(): a line segment, specified by start and end position.
geom_rect(): rectangles.
geom_polygon(): filled polygons.
geom_text(): text.

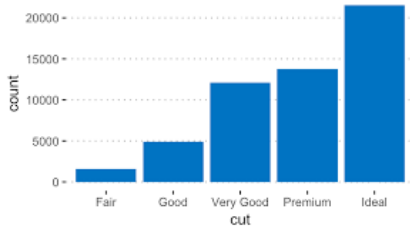




One variable:

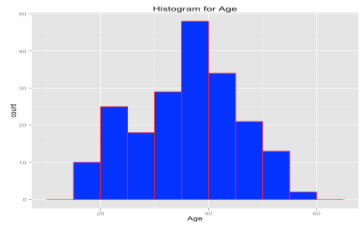
Discrete:

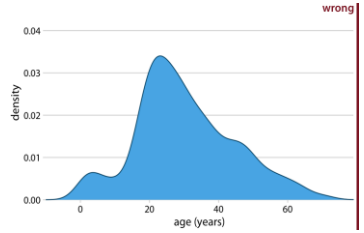
geom_bar(): display distribution of discrete variable.

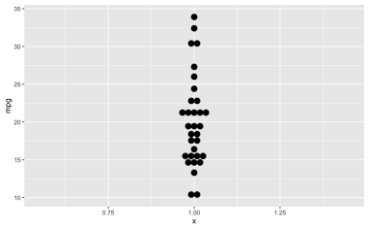


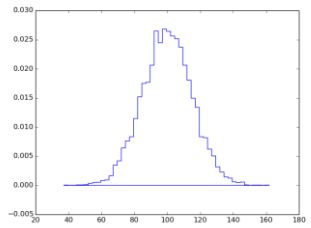
Continuous:

geom_histogram(): bin and count continuous variable, display with bars.
geom_density(): smoothed density estimate.
geom_dotplot(): stack individual points into a dot plot.
geom_freqpoly(): bin and count continuous variable, display with lines.







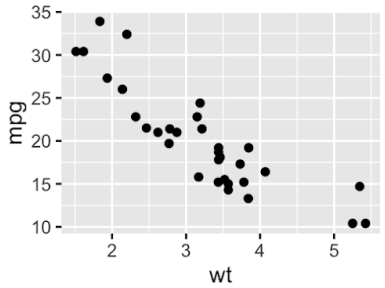


Geoms (2)

Two variables:

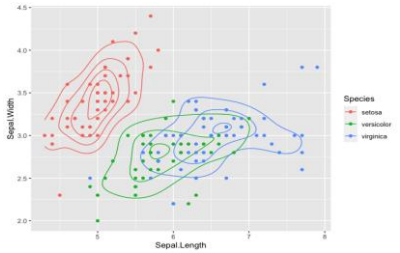
Both continuous:

geom_point(): scatterplot.
geom_quantile(): smoothed quantile regression.
geom_rug(): marginal rug plots.
geom_smooth(): smoothed line of best fit.
geom_text(): text labels.



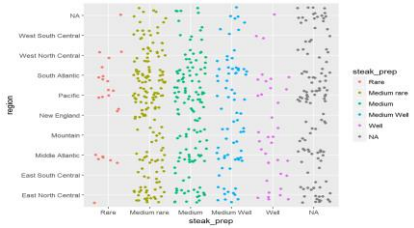
Show distribution:

geom_bin2d(): bin into rectangles and count.
geom_density2d(): smoothed 2d density estimate.
geom_hex(): bin into hexagons and count.



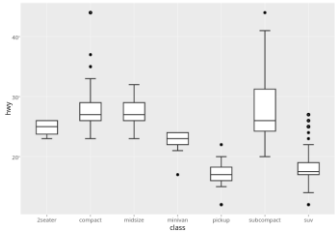
At least one discrete:

geom_count(): count number of point at distinct locations
geom_jitter(): randomly jitter overlapping points.



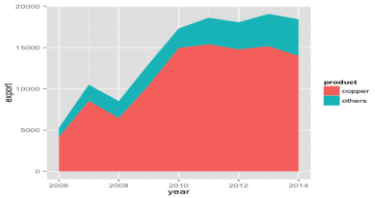
One continuous, one discrete:

geom_bar(stat = "identity"): a bar chart of precomputed summaries.
geom_boxplot(): boxplots.
geom_violin(): show density of values in each group.



One time, one continuous:

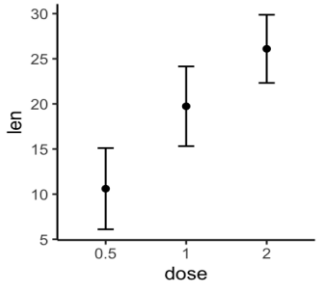
geom_area(): area plot.
geom_line(): line plot.
geom_step(): step plot.



Geoms (3)

Display uncertainty:

geom_crossbar(): vertical bar with center.
geom_errorbar(): error bars.
geom_linerange(): vertical line.
geom_pointrange(): vertical line with center.

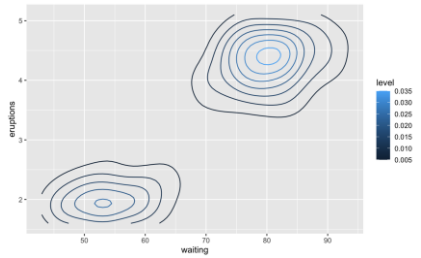


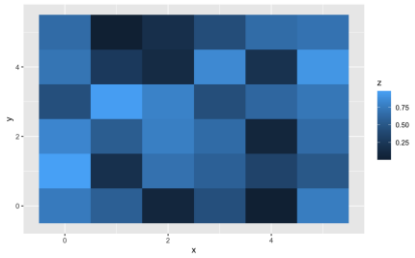
Spatial:

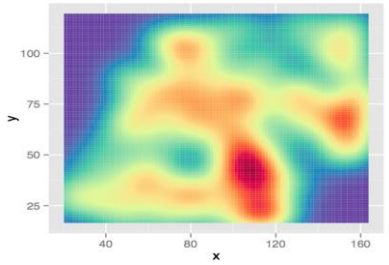
geom_map(): fast version of geom_polygon() for map data.

Three variables:

geom_contour(): contours.
geom_tile(): tile the plane with rectangles.
geom_raster(): fast version of geom_tile() for equal sized tiles.







Next class Hands-on Practice 2

Practising with ggplot2