Data Visualization with ggplot2
& Publication Ready Graphics

2491 - Data Challenge
Sarah Whitmee

Introduction and about this practical session

This practical will put into practice skills in using ggplot2 for visualising data and test your ability to critically evaluate existing data visualisations and hopefully improve them, in order to help you generate publication quality graphics.

• Assumed skills
– Writing R code into a script
– Identifying things that are visually pleasing

• Learning objectives
– Identifying the link between code and the graph it produces
– Being able to critique a graph
– Understanding why and how data is encoded and decoded visually
– Understanding the subjectivity of what is aesthetically pleasing

• Professional skills
– Creating high quality graphics

Make sure the packages tidyverse and here are installed.

Task 1 - Visualisation of the FEV1 data

Download the data file (fev1.csv) you need for the practical by browsing to the git repository at https://github.com/samclifford/2491_eda - you will be working with the dataset **fev1.csv.**

- On one person's computer, using the dataset **fev1.csv** build a plot that shows the relationship between FEV1 and age.
- For some questions we will only work on a sample of 20 individuals from the dataset. The code to make this sample is in task 3.

Question: Given the strength of the linear association between these two variables, do you think a linear trend would be an appropriate model?

Task 2 - Improving the plot

You may wish to save the plot object in Activity 1 and add to it, continuing to do so from here on, or copy-paste the code and modify it.

- Add meaningful labels for the x and y axes, including units, and change the plot's colour theme from the default. You may want to consult the suggested reading or search online for the included ggplot2 theme choices.

- Add a smooth line of best fit to the plot. You may wish to change its colour, turn off the standard error ribbon, or make other changes to it to help show the data and improve contrast with the background colour of your plot. The default behaviour is to use a LOESS smoother (Cleveland, Grosse, and Shyu 1992) which can be set with method = 'loess' as an argument to geom_smooth(). You could also use a generalised additive model (Wood 2017) with method = 'mgcv'.

Task 3 - Collaborative activities

Between you and your group, divide up the following activities so that each of you is working on one (you may need to double up). Can you do this using git?

BEFORE YOU START CREATE THE REPEATED MEASUREMENTS DATA

```
set.seed(10)

fev1_sampled <- fev1 %>%
        count(id) %>%
        filter(n > 6) %>%
        slice_sample(n = 20) %>%
        select(id) %>%
        inner_join(fev1)
```

Activity 3a - Showing structure.
We have repeat measurements on 20 individuals ***fev1_sampled***. Through either geometry grouping or other aesthetic options, determine a way to highlight which observations belong to the same individual.

Activity 3b - How many observations per individual?
Going back to the full data, many of the 300 individuals in the downloaded data set have been measured multiple times over the years? Count the number of times that each id is measured and make a bar plot to summarise the proportion of individuals who have 1, 2, etc. measurements.

Activity 3c - Incorporating height
Make a plot that shows both FEV1 and age but also includes height. There are several ways to do this.

Tidy up

Make sure you have saved your R script. If you have made changes, you can commit and push all your changes up to your shared remote repository if you are using one and then group members can pull those changes.

##

Activity 4 - Building an attempt at a plot

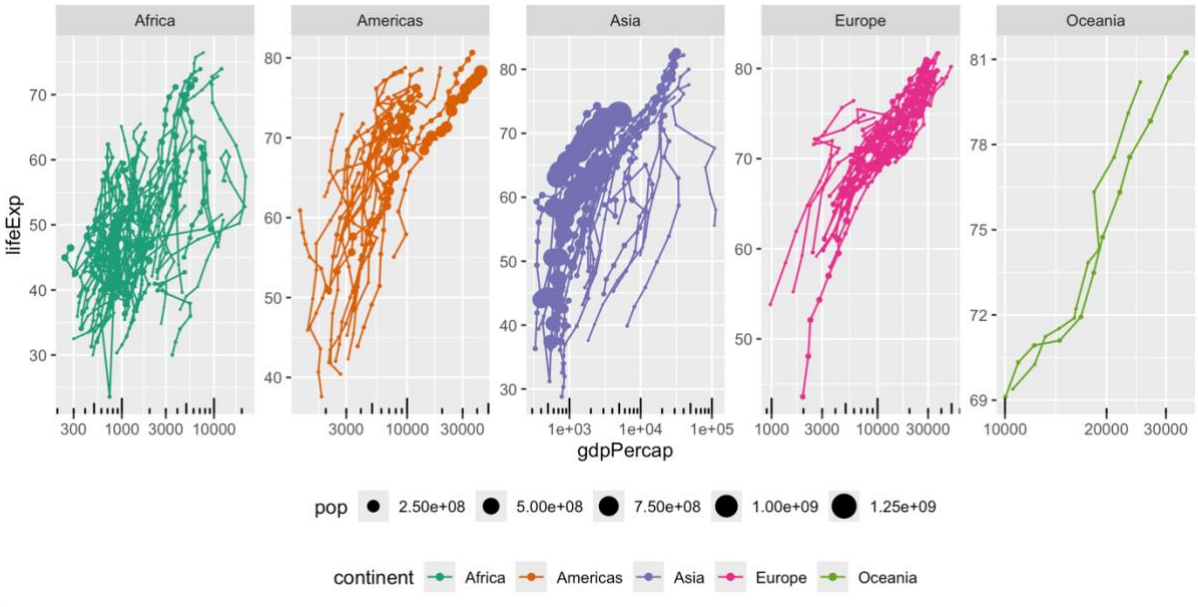
We will be looking at the gapminder data set as found in the gapminder package (Bryan 2017). This data has been collected from countries around the world and contains data on life expectancy, population and GDP per capita for 142 countries from 1952 to 2007.

Exercise 4.1: Copy and paste the code below to produce a plot showing how the relationship between GDP, life expectancy and population vary over time and continent.

******Ask if you cannot load gapminder***

```
library(gapminder)
library(tidyverse)
data(gapminder)

ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
        geom_path(aes(group = country, color = continent)) +
        geom_point(aes(color = continent, size = pop)) +
        scale_color_brewer(palette = 'Dark2') + scale_x_log10() +
        annotation_logticks(sides = 'b') + facet_wrap( ~ continent, scales = 'free', nrow = 1) +
        scale_size_area() +        theme(legend.position = 'bottom', legend.box = 'vertical')
```



Exercise 4.1: Discuss, within your group, what you think is good and bad about this plot. Does it conform to Tufte's principles of graphical excellence? Is it easy to interpret? Does it show the relationship we are interested in? List three important improvements that are needed for this graph to be useful.

Exercise 4.2: As a group, discuss what you think each line of code in the above block does.

Activity 5 – Making a better graph

Based on the ideas discussed, build a graph which your group believes better shows the relationship between life expectancy and GDP. First think about what story you want your plot to tell; are you interested in trends over space and/or time? Are you interested in a particular continent or even just one country?

You may choose to either modify the code given above or create your own graph from scratch. Make sure your code is written in your script i.e. with appropriate comments.

Some things you may wish to consider:

- fixing up the axis labels
- a relevant title
- a different theme
- different plotting geometries
- different aesthetic options for colour, shape, etc.

You may wish to sketch the graph by hand before attempting to write the R code to generate it. This will help you and your group come to an agreement about the plot you want to make and will help the tutors understand what you're aiming for when you ask them for help.

If you get stuck, look at the ggplot2 documentation or ask a tutor.

Exercise 5.1: Make a plot, save it to your computer and write comments in your code or standalone document that outline what the changes you made were and why.

Activity 5.2 – Making a worse graph

Make a new graph as in the previous activity but make **it as bad as possible** while still attempting to honestly show the information (i.e. don't add things to the plot which can't be derived from the variables in the plot).

Consider the principles of graphical excellence and how can we go against them to make a truly terrible plot. Think about what was bad about the plot provided earlier.

Consider abusing the ability to map graphical options (e.g. colour, fill, line type, point size) to our variables of interest.

Exercise 5.2: Make a plot, save it to your computer and write comments in your code or standalone document that outline what the changes you made were and why.

##

Answers

Answer 4.1:
The plot shows each country's data connected by a line so it's clear that individual country's GDP and life expectancy changes over time in a sequence, although we don't know which direction along the line we are travelling forward in time. By separating each continent out, we can see the trends geographically, such as Europe being clustered quite closely and Africa being very scattered (lots of variation across space and time, rather than a nice orderly procession from bottom left to top right). Logarithmic axis helps put the lines at approximately 45 degree angles rather than as difficult to read logarithmic curves. The large points obscure the data. Continent is mapped to both colour and facet. It's hard to make comparisons across continents because the x and y scales are different, so we must think very hard about whether Asia is richer or poorer than Europe, on average, and we may miss that there is very little overlap in the Oceania and Africa values.

Three things which might be worth changing:
1. Same axis scaling

2. Show less data (either fewer variables or fewer continents/years)
3. Put time on the x axis

Answer 4.2:

```
library(gapminder) # load package with data
library(tidyverse) # load package to manipulate and visualise data
data(gapminder) # load the gapminder data

# make a plot with the gapminder data
ggplot(data = gapminder, # put GDP as X variable and LE as Y
aes(x = gdpPercap, y = lifeExp)) +

# draw the X-Y pairs as a line, ordered by appearance in data frame
# group the lines by country and colour them by the continent
geom_path(aes(group = country, color = continent)) +

# draw points at each X-Y pair, coloured by continent and
# with their size based on the POP variable
geom_point(aes(color = continent, size = pop)) +

# change the colour scheme from the default for any colour aesthetics
# make a logarithmic scale on the X axis
scale_color_brewer(palette = 'Dark2') + scale_x_log10() +

# put logarithmic ticks on the bottom X axis showing that we don't have
# a uniform scaling
annotation_logticks(sides = 'b') +

# put each continent on its own set of axes, with all plots in one row
facet_wrap( ~ continent, scales = 'free', nrow = 1) +

# the size of the points, based on the POP variable, should have an area
# proportional to value of POP, rather than their radius
scale_size_area() +

# put the plot legend at the bottom and stack the variable keys vertically
theme(legend.position = 'bottom',
legend.box = 'vertical')
```

Answer 4.3:

There are many ways to do this, but a scatter plot which is static in time and shows the variability across space can help us avoid showing too much. Essentially we run out of things to reasonably change to show all dimensions of the data.

1. Reduce number of variables shown
2. Remove faceting, relying on colour to show difference
3. Clarity around units of variables
4. Human friendly axis labels

Example code

```
p <- ggplot(data = filter(gapminder, year == 2007),
aes(x = gdpPercap/1000, y = lifeExp)) +
geom_point(aes(group = country,#
size = pop*1e-6,
color = continent),
alpha = 0.75) +
scale_color_brewer(palette = "Dark2", name = "Continent") +
scale_x_log10() +
scale_size_area(name = "Population (millions)") +
theme_bw() + theme(legend.position = "bottom", legend.box = "vertical") +
xlab("GDP per capita, adjusted for inflation (1000 USD)") +
ylab("Life expectancy at birth (years)") +
theme(panel.grid.minor.x = element_blank()) +
annotation_logticks(sides = "bt")
p
```