

1 Step 0: R and RStudio, know your tools

2 Step 1: Introduction to R

3 Bonus Step: reproducible reports

4 Step 2: Data in, data out

5 Step 3: Analyzing (tabular) data

6 Step 4: Plotting data

7 Step 5: SummarizedExperiment: your best friend for “bioinformatics datasets”

8 Useful material

Session Info

Code ▾

Introduction to R and Bioconductor - hands-on session

MSE - Module Genetic Epidemiology



([https://www.unimedizin-](https://www.unimedizin-mainz.de/imbei/startseite/mse/)

Master of Science Programme
Epidemiology

[mainz.de/imbei/startseite/mse/](https://www.unimedizin-mainz.de/imbei/startseite/mse/))

Federico Marini (<https://federicomarini.github.io>) marinif@uni-mainz.de
(<mailto:marinif@uni-mainz.de>)

IMBEI, University Medical Center Mainz (<https://www.unimedizin-mainz.de/imbei/>)

 @FedeBioinfo (<https://twitter.com/FedeBioinfo>)

Alicia Schulze (alpoplaw@uni-mainz.de (<mailto:alpoplaw@uni-mainz.de>))
IMBEI, University Medical Center Mainz (<https://www.unimedizin-mainz.de/imbei/>)

2023/07/05-06

This lecture is inspired in its structure and organisation by the “Introduction to data analysis with R and Bioconductor” - <https://carpentries-incubator.github.io/bioc-intro/> (<https://carpentries-incubator.github.io/bioc-intro/>)

1 Step 0: R and RStudio, know your tools

1.1 What is R? Why should I use R?

Available at www.r-project.org

- *free* statistical environment for interactive use
- interpreted, functional scripting/programming language - you type in, you see output
- descends from the S language, written by statisticians for statisticians

What can you do with R?

- Anything!
 - do calculations
 - write functions
 - analyse data. ALL the data. Well, almost. But really, almost.
 - apply advanced statistical techniques
 - do beautiful & publication-ready plots
 - develop interactive web-applications
 - presentations & documents (this one)

Why should I use R?

- it works! And it is quite powerful
- it is free, open-source, and available for all OS
- you can *really* do whatever you might aim to do in terms of statistics
- it offers awesome possibilities for (interactive) graphing
- it has a wide, active and competent community (ok, communities: statistics, bioinformatics, machine learning, ...)
- it can be extended with packages. More power!
- escape Point-and-Click-land, you work with syntax: you can use, re-use elements, validate & reproduce analysis

Why should I *not* use R?

- you use it or lose it
- the learning curve might be steep
- can be frustrating if you have errors
- help might be available, but it is very technical
- many packages, a blessing and a curse: how many, how good

1.2 Let's get started!

Get R - and RStudio

- go to <https://cloud.r-project.org> (<https://cloud.r-project.org>) and get the latest version
- go to <https://www.rstudio.com/products/rstudio/> (<https://www.rstudio.com/products/rstudio/>) and download RStudio
- ... or use your text editor of choice

Alternatives:

- OpenAnalytics Architect (<https://www.getarchitect.io/>) (<https://www.getarchitect.io/>)
- Microsoft R Tools for Visual Studio (www.visualstudio.com/vs/rtvs/)
- Emacs Speaks Statistics...

1.3 First ride: look around you

Open up RStudio - You'll have four panes

- the Source for your scripts and documents (top-left, in the default layout)
- your Environment/History (top-right),
- your Files/Plots/Packages/Help/Viewer (bottom-right), and
- the R Console (bottom-left).

Want to customize this?

Tools -> Global Options -> Pane Layout

Advantages of an IDE

- all in one window!
- keyboard shortcuts, autocompletion, highlighting -> type easier, do less errors

1.4 Folder structure

It is good practice to keep a set of related data, analyses, and text self-contained in a single folder, called the working directory.

Why?

- Easier to have “self-contained” research units!
- A project “does not interfere” with other projects
- Gives a structure, easier to find things, use, reuse
- Someone else (including future you) can understand what goes on

How?

RStudio projects!

Custom settings, per project.

Let's create one live now - and have the workspace NOT saved

What is the best structure?

One, used consistently - not gonna touch on naming things as it can get hot quickly :)

With R...

```
dir.create()
```

```
file.edit()
```

Where am I doing things?

The working directory is the place from where R will be looking for and saving the files.

`getwd()` / `setwd()` , but not in your scripts (fails on others' computers!)

1.5 Interacting with R

Instructions, commands.

Scripts, console - use the editor and have a complete record on what you did!

Shortcuts FTW!

Even better: Reproducible documents, with R Markdown

Nice resources on top: * RStudio cheatsheet about the RStudio IDE! * the internet/rstats community!

1.6 Seeking help

- ?
- ??
- built-in RStudio help interface - and shortcuts!

1.6.1 Where to ask for help?

- your neighbour - covid-conform, do interact within each other!
- your colleagues
- rdocumentation.org website
- the web: google, StackOverflow

The main point: describe well your problem, “help others help you”

Others need to reproduce your error to help you better: `saveRDS()` , `dput()` , `sessionInfo()`

1.7 R packages

R packages...

- are fundamental components of R ecosystem
- extend base R functionality for a specific purpose
- bundle new functions, data sets, and documentation
- are contributed by independent developers
- have dependency management

Repositories:

- CRAN: Managed official package repository network
- Bioconductor: curated bioinformatics packages (vignettes mandatory, integrated ecosystem!)
- GitHub: un-managed, bleeding edge - but also excellent ones (“just not on CRAN”)

My contributions so far:

- `flowcatchR` <https://bioconductor.org/packages/flowcatchR/>
- `pcaExplorer` <https://bioconductor.org/packages/pcaExplorer/>

- `ideal` (<https://bioconductor.org/packages/ideal/>)
- `iSEE` (<https://bioconductor.org/packages/iSEE/>)
- `GeneTonic` (<https://bioconductor.org/packages/GeneTonic>)

1.7.1 How to use packages

- Install: once (for every major R version)
- Load: in each session
- Use like any base R functionality

For Bioconductor packages...

Hide

```
install.packages("BiocManager")
```

Hide

```
library("BiocManager")
BiocManager::install()
```

Relevant commands:

- `install.packages ("packagename")` - check it online at CRAN!
- `installed.packages ()`
- `.libPaths ()`
- `update.packages ()`
- `library ("packagename")`
- `help (package="packagename") , data () , browseVignettes () , vignette () , citation ("packagename")`

Something you might have already done:

Hide

```
BiocManager::install("SummarizedExperiment")
BiocManager::install("DESeq2")
```

2 Step 1: Introduction to R

Here we will touch on the first commands in R, so that you can

- Define the following terms as they relate to R: object, assign, call, function, arguments, options.
- Assign values to objects in R.
- Learn how to name objects
- Use comments to inform script.
- Solve simple arithmetic operations in R.
- Call functions and use arguments to change their default options.

- Inspect the content of vectors and manipulate their content.
- Subset and extract values from vectors.
- Analyze vectors with missing data.

2.1 R is a powerful calculator

... but not just that.

Type the following

Hide

```
2 + 2  
# [1] 4  
log(2)  
# [1] 0.6931472  
347 * 73841  
# [1] 25622827  
7/2  
# [1] 3.5  
7%/%2  
# [1] 3  
7%%2  
# [1] 1
```

2.2 ♪♪ Help! ♪♪

Hide

```
# this calls the help for a function to plot a histogram  
?hist  
# this is just the same  
help(hist) ## what about ??
```

Hide

```
?apropos
apropos("row")
# [1] ".row"
# [3] ".rowNamesDF<="
# [5] ".rs.api.tutorialLaunchBrowser"
# [7] ".rs.formatRowNames"
# [9] ".rs.nrow"
option"
# [11] ".rs.tutorial.launchBrowser"
# [13] "add_row"
# [15] "arrow"
# [17] "as_tibble_row"
# [19] "bind_rows"
# [21] "bindROWS"
# [23] "browseEnv"
# [25] "browser"
# [27] "browserSetDebug"
# [29] "browseUCSCTrack"
# [31] "browseVignettes"
# [33] "colAvgsPerRowSet"
# [35] "combineRows"
# [37] "db_query_rows"
# [39] "elementNROWS"
# [41] "extractROWS"
# [43] "group_rows"
# [45] "indexByRow"
printing"
# [47] "mergeROWS"
# [49] "narrow"
# [51] "narrow"
# [53] "nrow"
# [55] "nrow"
# [57] "nrow"
# [59] "nrow"
# [61] "NROW"
# [63] "nrows"
# [65] "panel_rows"
# [67] "remove_rownames"
# [69] "replaceROWS"
# [71] "row_number"
# [73] "row.names.data.frame"
# [75] "row.names<="
# [77] "row.names<-.default"
# [79] "rowAlls"
# [81] "rowAnyNAs"
# [83] "rowAnyns"
# [85] "rowAvgsPerColSet"
# [87] "rowCollapse"
".rowMeans"
".rowSums"
".rs.explorer.defaultRowLimit"
".rs.isBrowserActive"
".rs.refreshShinyLaunchBrowserO
"add_row"
"add_rownames"
"arrows"
"auto_browse"
"bindROWS"
"bindROWS"
"browseKEGG"
"browserCondition"
"browserText"
"browseURL"
"colAvgsPerRowSet"
"column_to_rownames"
"cur_group_rows"
"dplyr_row_slice"
"elementNROWS"
"extractROWS"
"has_rownames"
"makeClassinfoRowForCompactPrin
ting"
"n2mfrow"
"narrow"
"new_rownwise_df"
"nrow"
"nrow"
"nrow"
"NROW"
"NROW"
"nrows"
"PlantGrowth"
"replaceROWS"
"row"
"row.names"
"row.names.default"
"row.names<-.data.frame"
"rowAlls"
"rowAnyMissings"
"rowAnyNAs"
"rowAnyns"
"rowAvgsPerColSet"
"rowCollapse"
```

```

# [89] "rowCounts"
# [91] "rowCummaxs"
# [93] "rowCummins"
# [95] "rowCumprods"
# [97] "rowCumsums"
# [99] "rowData"
# [101] "rowData<="
# [103] "rowDataColorMap<="
# [105] "RowDataTable"
# [107] "rowDiffs"
# [109] "rowIQRDiffs"
# [111] "rowIQRs"
# [113] "rowLogSumExps"
# [115] "rowMadDiffs"
# [117] "rowMads"
# [119] "rowMax"
# [121] "rowMaxs"
# [123] "rowMeans"
# [125] "rowMeans2"
# [127] "rowMedians"
# [129] "rowMin"
# [131] "rowMins"
# [133] "rownames"
# [135] "rownames"
# [137] "rownames"
# [139] "rownames_to_column"
# [141] "rownames<="
# [143] "rownames<="
# [145] "rownames<="
# [147] "rowOrderStats"
# [149] "rowPair<="
# [151] "rowPairNames<="
# [153] "rowPairs<="
# [155] "rowProds"
# [157] "rowQuantiles"
# [159] "rowRanges"
# [161] "rowRanges"
# [163] "rowRanks"
# [165] "rows_append"
# [167] "rows_insert"
# [169] "rows_update"
# [171] "rowSdDiffs"
# [173] "rowSds"
# [175] "rowSelectionColorMap"
# [177] "rowSubset<="
# [179] "rowsum.data.frame"
# [181] "rowsum.DGEList"
# [183] "rowSums"
# [185] "rowSums2"
"rowCounts"
"rowCummaxs"
"rowCummins"
"rowCumprods"
"rowCumsums"
"rowData"
"rowData"
"rowDataColorMap"
"RowDataTable"
"rowDiffs"
"rowid_to_column"
"rowIQRDiffs"
"rowIQRs"
"rowLogSumExps"
"rowMadDiffs"
"rowMads"
"rowMaxs"
"rowMeans"
"rowMeans2"
"rowMedians"
"rowMedians"
"rowMins"
"rownames"
"rownames"
"rownames"
"rownames"
"rownames"
"rownames<="
"rownames<="
"rownames<="
"rownames<="
"rowOrderStats"
"rowPair"
"rowPairNames"
"rowPairs"
"rowProds"
"rowQ"
"rowQuantiles"
"rowRanges"
"rowRanges<="
"rowRanks"
"rows_delete"
"rows_patch"
"rows_upsert"
"rowSdDiffs"
"rowSds"
"rowSubset"
"rowsum"
"rowsum.default"
"rowsum.SummarizedExperiment"
"rowSums"
"rowSums2"

```

```

# [187] "rowTabulates"
# [189] "rowVarDiffs"
# [191] "rowVars"
# [193] "rowWeightedMads"
# [195] "rowWeightedMeans"
# [197] "rowWeightedMedians"
# [199] "rowWeightedSds"
# [201] "rowWeightedVars"
# [203] "rowwise"
# [205] "separate_rows"
# [207] "tibble_row"
# [209] "validate_rowwise_df"

```

```

"rowTabulates"
"rowVarDiffs"
"rowVars"
"rowWeightedMads"
"rowWeightedMeans"
"rowWeightedMedians"
"rowWeightedSds"
"rowWeightedVars"
"sameAsPreviousROW"
"separate_rows_"
"ToothGrowth"
"xpdrows.data.frame"

```

- integrated help system, with executable examples
- (for some packages) vignettes (typical problem, commands, and workflow)
- CRAN Task Views: <https://cran.r-project.org/web/views/> (<https://cran.r-project.org/web/views/>)
- Books!
- Courses!
- Online: mailing lists, forums (StackOverflow, ...), blogs, Twitter (#rstats)

2.3 Your starting vocabulary - a.k.a. Exercise Session 0

- `getwd()` and `setwd()` - Tab is your friend
- `<-`, `=`: the assignment operator
- `ls()`, `rm()`
- `str()`
- `example()`, `help()` / `?[function]`
- `print()`
- `q()` / `quit()`
- logical operators: `TRUE`, `FALSE`, `!`, `==`, `!=`, `<`, `>`, `<=`, `>=`, `|`, `&`, `xor()`
- `c()`
- data have help items too: e.g. `cars`

Find out what these do!

2.3.1 Exercise Session 0 - Solutions

Hide

```
?getwd  
?setwd  
?`<-`  
help(ls)  
help(rm)  
?str  
?example  
help(help)  
?print  
help(quit)  
?c  
?cars
```

2.4 Make your life easier - Notes for your future self

- add comments and document your own code

[Hide](#)

```
# This is a comment
```

- write clean code - use spaces, indentation
- use an editor with syntax highlighting/some form of autocompletion

Careful here:

- R is case sensitive and has zero-tolerance with mis-spelled names
- parenthesis: open *and* close them
- special attention with missing values, factors VS strings: R is clever, but you might think differently
- do not be stingy with parentheses - if this helps you
- same goes with comments - your colleagues and your future self will thank you

2.5 Exercise session 1

Grab some mini-postit!

- find out more about the `iris` dataset. What is it about at all? How many variables are included?
How many observations?
- `rep` licate! find out a function that `rep` licates elements of a vector to produce this

[Hide](#)

```
1 1 1 1
```

BONUS: ... and this

[Hide](#)

2.5.1 Exercise Session 1 - Solutions

► Details

2.6 Data types

R can recognize different general types of data

- numbers (numeric)
- character strings (text)
- logical (e.g. `class(TRUE)`)
- factors (“integers with a set of labels”) - it is categorical data!
- special ones: dates, time, ...

When and where to use which?

2.7 I was curious about you

In a previous edition of a similar course, I wanted to know:

- How old are you?
- What is your current academic level? (PI, Postdoc, PhD, master)
- What is your current knowledge level of R? (pro-good-intermediate-poor-none)
- What is your knowledge of programming languages in general?
- What is your experience level with genomics and RNA-seq data?
- How familiar are you with mogon and parallel computing? (I am a regular user/Once in a while i used it/I know it exists/I heard we had some servers around/Is this supposed to be in the cloud?!)
- What are your expectations from the course?

3 Bonus Step: reproducible reports

Our aims:

- Understand what R Markdown is and why you should use it
- Learn how to construct an R Markdown file
- Export an R Markdown file into many file formats
- → You are all set to use Rmd to document any of your analyses!

3.1 Reproducible reports with R Markdown

R Markdown allows you to create documents that serve as a neat record of your analysis.

Why?

- we want other researchers to easily understand what we did in our analysis, otherwise nobody can be certain that you analysed your data properly (yay, reproducible research!)
- create an R markdown document as an appendix to a paper or project assignment, upload it to an online repository such as Github, or simply to keep as a personal record (*future you will thank present you for this*)

The key point is...

R Markdown documents present your code alongside its output (graphs, tables, etc.) with conventional text to explain it, a bit like a notebook. To do this, R Markdown uses **markdown** syntax.

3.2 Markdown

Markdown is a very simple *markup* language which provides methods for creating documents with headers, images, links etc. from plain text files, while keeping the original plain text file easy to read.

You can convert Markdown documents to many other file types like `.html` or `.pdf` to display the headers, images etc..

It might sound complicated. But *really isn't*,



First things first: install the required software

- R and RStudio (guess you have it already)

Hide

```
install.packages("rmarkdown")
```

Hide

```
library(rmarkdown)
```

- `knitr` comes along, `pandoc` too. You should quickly be all set!

3.3 Basics of markdown

ABC here, let's go through it:

[\(http://rmarkdown.rstudio.com/authoring_basics.html\)](http://rmarkdown.rstudio.com/authoring_basics.html)

plus... a beautiful cheat sheet is there for you!

[\(http://rmarkdown.rstudio.com/lesson-15.html\)](http://rmarkdown.rstudio.com/lesson-15.html)

[\(http://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf\)](http://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf)

Using LaTeX? No problem, you can use *LATEX* here as well!

$$(f(x) = \sum_{i=0}^n \frac{a_i}{1+x})$$

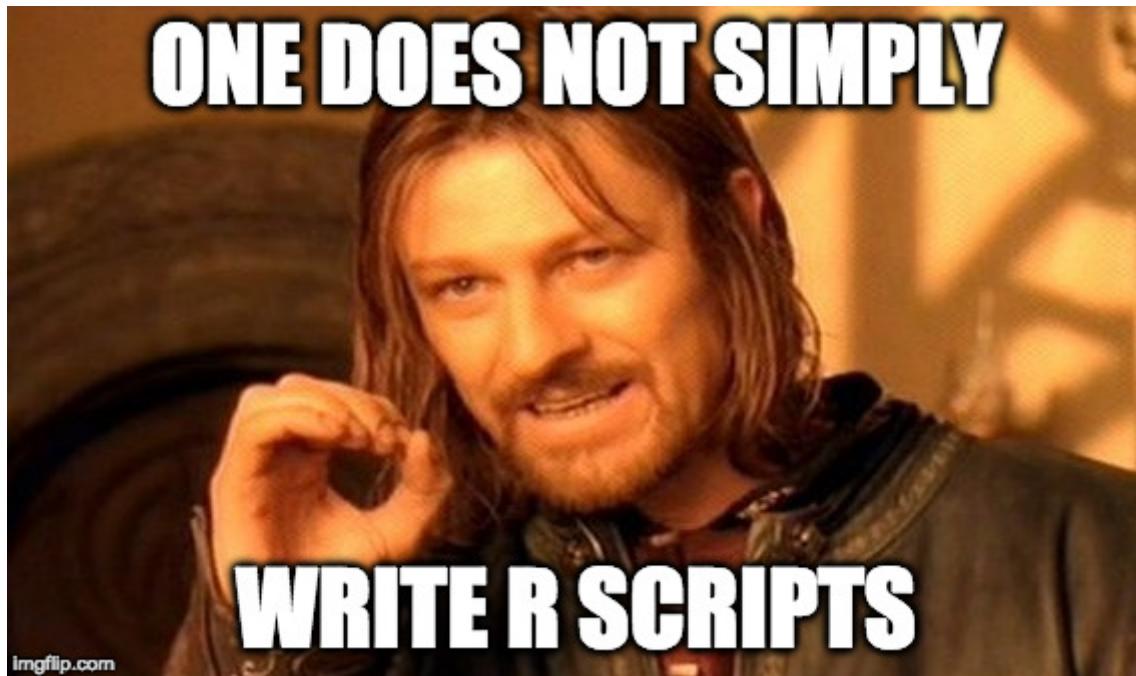
What can you do with Rmarkdown?

[\(http://rmarkdown.rstudio.com/gallery.html\)](http://rmarkdown.rstudio.com/gallery.html)

[\(http://rmarkdown.rstudio.com/formats.html\)](http://rmarkdown.rstudio.com/formats.html)

3.4 Let's create one together!

3.4.1 Why is Rmd better than R



The price to pay to have an Rmd document is sooooo small - and for that, you get

- code, text, output all together
- one file only - no need to get lost
- it even looks nice :)

3.4.2 Create an Rmarkdown file

To create a new R Markdown file (`.Rmd`), select `File -> New File -> R Markdown...` in RStudio, then choose the file type you want to create.

The newly created `.Rmd` file comes with basic instructions but we want to create our own R Markdown script, so let's get to know the different parts of an Rmd file

- An (optional) YAML header surrounded by `---`s
- R code chunks surrounded by backticks (`````)
- text mixed with simple text formatting

3.4.3 Inserting figures

Uh, you can insert figures also like this

```
! [] (images/grcat.png)
```



3.5 Insert text and code - any text, any code

```
```r
n <- 10
rnorm(n)
[1] 0.2750427 1.1800167 1.1070905 0.1732659 -0.2742060 1.5992610 -2.0
094319 -0.7177109
[9] 0.4198595 -1.1974104
```
```

Shortcut: Ctrl + Alt + I

Input code: you can use multiple languages including R, Python, and SQL, many more (specify the language in the chunk options)

Inline code can be added with `r 1+1`

3.5.1 Chunk options

Deatiled very nicely here: <https://yihui.name/knitr/options/> (<https://yihui.name/knitr/options/>)

A simple set of options which you can use for many documents:

Hide

```
set.seed(42)
knitr:::opts_chunk$set(
  comment = NA,
  fig.align = "center",
  fig.width = 7,
  fig.height = 7,
  warning = FALSE,
  eval = TRUE
)
```

3.5.2 Knit!

Use the `Knit` button in the RStudio IDE to render the file and preview the output with a single click or keyboard shortcut (`Ctrl + Shift + K`).

To generate a report from the file, run the `render` command (works also outside of RStudio):

Hide

```
library("rmarkdown")
rmarkdown:::render("yourfile.Rmd")
```

It was a deep dive, but now...

- You are familiar with the Markdown syntax and code chunk rules.
- You can include figures and tables in your Markdown reports.
- You can create R Markdown files and export them to pdf or html files.

3.6 (Much) more on Rmarkdown

- <http://rmarkdown.rstudio.com/> (<http://rmarkdown.rstudio.com/>)
- http://stat545.com/block007_first-use-rmarkdown.html (http://stat545.com/block007_first-use-rmarkdown.html)
- <http://rmarkdown.rstudio.com/lesson-1.html> (<http://rmarkdown.rstudio.com/lesson-1.html>)
- ... to <http://rmarkdown.rstudio.com/lesson-15.html> (<http://rmarkdown.rstudio.com/lesson-15.html>)
- <http://rmarkdown.rstudio.com/articles.html> (<http://rmarkdown.rstudio.com/articles.html>)

You can do much much more (presentations, websites, manuscripts,...)

3.7 Exercise session Bonus

- create a new Rmarkdown document
- can you find out how to generate a word document as output?
- insert some code you previously used for exploring the small survey data - remember, a fresh session is run when knitting, so you need the commands from the very start!

3.7.1 Exercise Session Bonus - Solutions

- File -> New File -> R Markdown... in RStudio
- add this in the yaml header

```
output:
  word_document
```

4 Step 2: Data in, data out

4.1 Importing data in R

80-20? 90-10? Import, clean, prepare, transform your data

Sources:

- Files, Clipboard, URL
- **Plain text file: Comma-separated, tab-delimited, ...**
- R format file
- SAS / Stata / SPSS file: package `haven`
- Spreadsheet (Excel): package `readxl` - highly recommended!
- Database: RSQLite, RPostgreSQL, RMySQL, ...

4.2 The vocabulary of importing

... and exporting

- `read.table()`, `write.table()` + `read.csv|delim`
- the option `stringsAsFactors=FALSE`
- `load()`, `save()` / `readRDS()`, `saveRDS()`
- via `haven`: `read_sas()`, `read_spss()` / `write_sas()`, `write_sav()`
- via `readxl`: `read_excel()`

Check out their documentation pages!

Other options: `rio`, RStudio GUI

4.3 Take a look at the data

Go to <https://github.com/federicomarini/rbioc2016> (<https://github.com/federicomarini/rbioc2016>)

-> `inst/extdata`
-> `survey_responses.csv`, in its raw format

You can load it directly like this

Hide

```
surveyrbioc <- read.csv("https://raw.githubusercontent.com/federicomarini/rbioc2016/master/inst/extdata/survey_responses.csv")
```

Or install the package and load it from there

Hide

```
library("devtools")
install_github("federicomarini/rbioc2016")
library("rbioc2016")
data(surveyrbioc)
```

4.4 Input data: Step by step, by hand?

Sometimes your data is either small and/or not in an Excel-like tabular format.

What to do? You combine the elements together!

Hide

```
Q1 <- c(28,27,33,32,29)
# should return this
Q1
[1] 28 27 33 32 29

Q2 <- c("PhD student","PhD student", "Postdoc","PhD student","PhD student")
Q2
[1] "PhD student" "PhD student" "Postdoc"      "PhD student" "PhD student"
# ... and so on
```

4.5 Combine the variables to a matrix

We have seen `c()`. We also have

- `cbind`
- `rbind`

[Hide](#)

```
firstTwo <- cbind(Q1,Q2)
firstTwo
  Q1    Q2
[1,] "28" "PhD student"
[2,] "27" "PhD student"
[3,] "33" "Postdoc"
[4,] "32" "PhD student"
[5,] "29" "PhD student"
```

[Hide](#)

```
rbind(Q1,Q2)
 [,1]      [,2]      [,3]      [,4]      [,5]
Q1 "28"     "27"     "33"     "32"     "29"
Q2 "PhD student" "PhD student" "Postdoc" "PhD student" "PhD student"
```

Is this what you wanted?

4.6 Applying the first functions

But first, what can you do on these objects?

[Hide](#)

```

sum(Q1)
[1] 149
sum(Q2)
Error in sum(Q2): invalid 'type' (character) of argument
summary(Q1)
   Min. 1st Qu. Median     Mean 3rd Qu.    Max.
   27.0    28.0    29.0    29.8    32.0    33.0
summary(Q2)
  Length     Class      Mode
  5 character character
str(Q1)
num [1:5] 28 27 33 32 29
str(Q2)
chr [1:5] "PhD student" "PhD student" "Postdoc" "PhD student" "PhD student"
mean(Q1)
[1] 29.8
dim(firstTwo)
[1] 5 2
firstTwo[,1]
[1] "28" "27" "33" "32" "29"
mean(firstTwo[,1]) # Why, damn, why? Meet coercion
[1] NA
class(firstTwo)
[1] "matrix" "array"

```

4.7 matrix, data.frame and list

- a `matrix` can contain one type of data - if numeric, you unleash all the matrix algebra power!
- a `data.frame` can store more types of data (one per column)
- a `list` is like a big box where you can put anything - but this is not always what you want

What is best?

Let's try with a `list`

[Hide](#)

```
Q3 <- c("intermediate", "poor", "good", "none", "intermediate")
mylist <- list(Q1, Q2, Q3)
mylist
[[1]]
[1] 28 27 33 32 29

[[2]]
[1] "PhD student" "PhD student" "Postdoc"      "PhD student" "PhD student"

[[3]]
[1] "intermediate" "poor"                 "good"                "none"                "intermediat
e"
```

[Hide](#)

```
## access your elements with
mylist[[1]]
[1] 28 27 33 32 29
mylist[[1]][2]
[1] 27
```

How do we create a `data.frame`?

[Hide](#)

```
mydf <- data.frame(age = Q1,
                     level = Q2,
                     rexp = Q3)

mydf
  age       level       rexp
1 28 PhD student intermediate
2 27 PhD student          poor
3 33    Postdoc           good
4 32 PhD student          none
5 29 PhD student intermediate
class(mydf$age)
[1] "numeric"
```

4.7.1 Exploring a `data.frame`

[Hide](#)

```
mydf$age    # it's all about the money :)  
[1] 28 27 33 32 29  
mydf[,1]  
[1] 28 27 33 32 29  
names(mydf)  
[1] "age"   "level" "rexp"  
rownames(mydf)  
[1] "1" "2" "3" "4" "5"  
dim(mydf)  
[1] 5 3  
nrow(mydf)  
[1] 5  
ncol(mydf)  
[1] 3
```

[Hide](#)

```

surveyrbioc <- read.csv("https://raw.githubusercontent.com/federicomarini/rbioc2016/master/inst/extdata/survey_responses.csv")
head(surveyrbioc)

      Q1        Q2        Q3        Q4        Q5
Q6
1 28 PhD student intermediate intermediate    good           I a
m a regular user
2 27 PhD student          poor intermediate   poor
I know it exists
3 33 Postdoc            good     good     good
I know it exists
4 32 PhD student          poor     poor     poor Is this supposed to b
e in the cloud?!
5 29 PhD student          none    poor     poor
I know it exists
6 33 Postdoc intermediate intermediate intermediate Once in a
while i used it

Q7
1
Learn Parallelization with R (and Bioconductor)
2
<NA>
3
use R scripts in parallel context (ex: alignments in RNA-seq)
4                                         Id like to g
ather practise to analyse count data and develope the necessary data design.
Additionally, it would be interesting to get to now how to find putative nove
l miRNAs.
5
Possible I will use R for editing RNA-seq data
6 I would describe myself as an advanced beginner, I am actively using R now
to look (mostly) at count tables from NGS data and make plots. I am especiaill
y interested in the Bioconductor and parallelization in R section, this is ne
w for me.
tail(surveyrbioc)

      Q1        Q2        Q3        Q4        Q5
17 28 Master student/else intermediate intermediate intermediate
18 39 Postdoc            good     intermediate     good
19 32 Master student/else          none    poor     genoWhat?
20 29 PhD student          poor     none     poor
21 31 PhD student          none    none     good
22 30 PhD student          none    none     good

      Q6
Q7
17                         I know it exists     better understanding of R and
to extend my knowledge
18                         I am a regular user Mainly interested in para

```

```

parallel computing options
19 Is this supposed to be in the cloud?!
<NA>
20 Is this supposed to be in the cloud?! To better understand R and
perform basic analysis
21 I heard we had some servers around working with
fastq files based on R
22 Is this supposed to be in the cloud?! I want to be able to analyze my sequence data on my own :P
names(surveyrbioc)
[1] "Q1" "Q2" "Q3" "Q4" "Q5" "Q6" "Q7"
# View(surveyrbioc)
str(surveyrbioc)
'data.frame': 22 obs. of 7 variables:
 $ Q1: int 28 27 33 32 29 33 40 23 27 23 ...
 $ Q2: chr "PhD student" "PhD student" "Postdoc" "PhD student" ...
 $ Q3: chr "intermediate" "poor" "good" "poor" ...
 $ Q4: chr "intermediate" "intermediate" "good" "poor" ...
 $ Q5: chr "good" "poor" "good" "poor" ...
 $ Q6: chr "I am a regular user" "I know it exists" "I know it exists" "Is this supposed to be in the cloud?!" ...
 $ Q7: chr "Learn Parallelization with R (and Bioconductor)" NA "use R scripts in parallel context (ex: alignments in RNA-seq)" "Id like to gather practise to analyse count data and develope the necessary data design. Additionally, it would" | truncated ...
summary(surveyrbioc)
      Q1          Q2          Q3          Q4
Q5
  Min.   :23.00   Length:22           Length:22           Length:22
  1st Qu.:27.25   Class :character   Class :character   Class :character
  Median :29.50   Mode  :character   Mode  :character   Mode  :character
  Mean   :30.14
  3rd Qu.:32.75
  Max.   :40.00
      Q6          Q7
Length:22           Length:22
Class :character   Class :character
Mode  :character   Mode  :character

surveyrbioc[, ]
      Q1          Q2          Q3          Q4          Q5
1 28      PhD student intermediate intermediate good
2 27      PhD student      poor intermediate poor
3 33      Postdoc        good       good       good

```

| | | | | | |
|----|----|-----------------------------|--------------|--------------|--------------|
| 4 | 32 | PhD student | poor | poor | poor |
| 5 | 29 | PhD student | none | poor | poor |
| 6 | 33 | Postdoc | intermediate | intermediate | intermediate |
| 7 | 40 | Postdoc | good | good | intermediate |
| 8 | 23 | Master student/ else | poor | poor | good |
| 9 | 27 | PhD student | none | poor | intermediate |
| 10 | 23 | Master student/ else | poor | poor | poor |
| 11 | 35 | Postdoc | poor | poor | intermediate |
| 12 | 34 | Master student/ else | poor | pro | poor |
| 13 | 31 | Postdoc | none | none | intermediate |
| 14 | 27 | PhD student | none | poor | poor |
| 15 | 24 | Master student/ else | poor | intermediate | intermediate |
| 16 | 28 | PhD student | none | poor | poor |
| 17 | 28 | Master student/ else | intermediate | intermediate | intermediate |
| 18 | 39 | Postdoc | good | intermediate | good |
| 19 | 32 | Master student/ else | none | poor | genoWhat? |
| 20 | 29 | PhD student | poor | none | poor |
| 21 | 31 | PhD student | none | none | good |
| 22 | 30 | PhD student | none | none | good |

Q6

1 I am a regular user
 2 I know it exists
 3 I know it exists
 4 Is this supposed to be **in** the cloud?!
 5 I know it exists
 6 Once **in** a **while** i used it
 7 I am a regular user
 8 I know it exists
 9 I know it exists
 10 Is this supposed to be **in** the cloud?!
 11 I know it exists
 12 I know it exists
 13 Is this supposed to be **in** the cloud?!
 14 I know it exists
 15 I know it exists
 16 I know it exists
 17 I know it exists
 18 I am a regular user
 19 Is this supposed to be **in** the cloud?!
 20 Is this supposed to be **in** the cloud?!
 21 I heard we had some servers around
 22 Is this supposed to be **in** the cloud?!

Q7

1
 Learn Parallelization with R (and Bioconductor)
 2
 <NA>
 3

use R scripts **in** parallel context (ex: alignments **in** RNA-seq)
4
Id like to gather practise to analyse count data and develope the necessary d
ata design. Additionally, it would be interesting to get to now how to find p
utative novel miRNAs.
5
Possible I will use R **for** editing RNA-seq data
6
I would describe myself as an advanced beginner, I am actively using R now to
look (mostly) at count tables from NGS data and make plots. I am especially i
nterested **in** the Bioconductor and parallelization **in** R section, this is new **f**
or me.
7
<NA>
8
Get to know R better, basic commands
9
To look **if** I can process my sequencing data on my own using R
10 I would like to learn more about R, especially how to use it **in** biomedical
research. Im **in** the 2nd Semester of the Rasterprogramm biomedicine, last Seme
ster I had two weeks bioinformatics and we used to work with R **for** statistic
s/ChIP-Seq/microarray-data analysis, but the time was too short, to go deep i
nto it, we just scratched the surface. So now I wish to learn some more, woul
d be great, **if** I could work with the programme by myself, **for** example **for** the
masterthesis.
11
learn more about R
12
Brush up on some R knowledge and maybe get some different perspective on Geno
me processing
13
to get a good and understandable introduction into R programming and bioinfor
matics
14
learning how to work with R
15
To learn the basics of R programming
16
<NA>
17
better understanding of R and to extend my knowledge
18
Mainly interested **in** parallel computing options
19
<NA>
20
To better understand R and perform basic analysis
21
working with fastq files based on R

4.8 Exercise session 2

Using the `surveyrbioc` object:

- Calculate the mean age of the participants
- How many participants did actually take part to the survey?
- How old was the oldest participant? (`max` can be your help)
- transpose the survey data and assign it to another variable
- Change the column names of this object and save this data set as a tab-separated ASCII file
- BONUS: what was the youngest participant expecting?

4.8.1 Exercise Session 2 - Solutions

► Details

5 Step 3: Analyzing (tabular) data

Describe, explore, transform, summarise data

5.1 Exploring, subsetting, manipulating, analysing

- `dim(x)` shows the dimensions of an object
- `str(x)` provides an overview of the structure of an object and the elements it contains
- `sum(x)`, `mean(x)`, `sd(x)` computes the sum, mean, or standard deviation of all the elements in `x`; `median(x)`, `quantile(x)`
- `length(x)` returns the number of elements in `x` (a vector)
- `sqrt(x)`, `log(x)` take the square root and the natural logarithm of a numeric - element or vector
- `hist(x, breaks=20, col="blue")` plots a histogram of variable `x` with 20 bins colored blue
- `unique(x)` returns the vector of unique elements in `x`
- `rm(x)` removes the object `x` from the environment (`rm(list=ls())` removes all objects)
- `sessionInfo()` prints information about R session and versions of all attached packages
- logical operators might often come handy!

5.2 Subsetting the data

This is the basic way it works

Hide

```
surveyrbioc [ROWS, COLUMNS]
```

You can subset with...

- integers
- blank spaces
- names
- logical vectors

Try to make a guess, given this vector.

[Hide](#)

```
vec <- c(6, 1, 3, 6, 10, 5)
```

What happens if you do this?

[Hide](#)

```
vec[2]  
[1] 1  
vec[c(5, 6)]  
[1] 10 5  
vec[-c(5, 6)]  
[1] 6 1 3 6  
vec > 5  
[1] TRUE FALSE FALSE TRUE TRUE FALSE  
vec[vec > 5]  
[1] 6 6 10
```

What happens if you do this?

[Hide](#)

```
df <- data.frame(  
  name = c("John", "Paul", "George", "Ringo"),  
  birth = c(1940, 1942, 1943, 1940),  
  instrument = c("guitar", "bass", "guitar", "drums"))  
)  
  
df  
  name birth instrument  
1  John  1940      guitar  
2  Paul  1942      bass  
3 George 1943      guitar  
4 Ringo  1940      drums  
  
df[c(2, 4), 3]  
[1] "bass"  "drums"  
df[, 1]  
[1] "John"   "Paul"   "George" "Ringo"  
df[, "instrument"]  
[1] "guitar" "bass"   "guitar" "drums"  
df$instrument  
[1] "guitar" "bass"   "guitar" "drums"
```

[Back to the survey](#)

[Hide](#)

```

# I just want the age
surveyrbioc[,1]
[1] 28 27 33 32 29 33 40 23 27 23 35 34 31 27 24 28 28 39 32 29 31 30
# or
surveyrbioc$Q1
[1] 28 27 33 32 29 33 40 23 27 23 35 34 31 27 24 28 28 39 32 29 31 30

# the first 4 columns
surveyrbioc[,c(1,2,3,4)]
   Q1           Q2           Q3           Q4
1 28    PhD student intermediate intermediate
2 27    PhD student          poor intermediate
3 33      Postdoc          good          good
4 32    PhD student          poor          poor
5 29    PhD student          none          poor
6 33      Postdoc intermediate intermediate
7 40      Postdoc          good          good
8 23 Master student/else    poor          poor
9 27      PhD student          none          poor
10 23 Master student/else    poor          poor
11 35      Postdoc          poor          poor
12 34 Master student/else    poor          pro
13 31      Postdoc          none          none
14 27      PhD student          none          poor
15 24 Master student/else    poor intermediate
16 28      PhD student          none          poor
17 28 Master student/else    intermediate intermediate
18 39      Postdoc          good intermediate
19 32 Master student/else    none          poor
20 29      PhD student          poor          none
21 31      PhD student          none          none
22 30      PhD student          none          none

surveyrbioc[,1:4]
   Q1           Q2           Q3           Q4
1 28    PhD student intermediate intermediate
2 27    PhD student          poor intermediate
3 33      Postdoc          good          good
4 32    PhD student          poor          poor
5 29    PhD student          none          poor
6 33      Postdoc intermediate intermediate
7 40      Postdoc          good          good
8 23 Master student/else    poor          poor
9 27      PhD student          none          poor
10 23 Master student/else    poor          poor
11 35      Postdoc          poor          poor
12 34 Master student/else    poor          pro
13 31      Postdoc          none          none
14 27      PhD student          none          poor

```

```

15 24 Master student/else          poor intermediate
16 28          PhD student        none      poor
17 28 Master student/else intermediate intermediate
18 39          Postdoc           good     intermediate
19 32 Master student/else          none      poor
20 29          PhD student        poor      none
21 31          PhD student        none      none
22 30          PhD student        none      none

# all but the last column
surveyrbioc[, -7]

      Q1          Q2          Q3          Q4          Q5
1 28          PhD student intermediate intermediate   good
2 27          PhD student       poor     intermediate poor
3 33          Postdoc          good     good     good
4 32          PhD student       poor     poor     poor
5 29          PhD student       none     poor     poor
6 33          Postdoc          intermediate intermediate intermediate
7 40          Postdoc          good     good     intermediate
8 23 Master student/else       poor     poor     good
9 27          PhD student       none     poor     intermediate
10 23 Master student/else      poor     poor     poor
11 35          Postdoc          poor     poor     intermediate
12 34 Master student/else      poor     pro     poor
13 31          Postdoc          none     none     intermediate
14 27          PhD student       none     poor     poor
15 24 Master student/else      poor     intermediate intermediate
16 28          PhD student       none     poor     poor
17 28 Master student/else intermediate intermediate intermediate
18 39          Postdoc          good     intermediate   good
19 32 Master student/else      none     poor     genoWhat?
20 29          PhD student       poor     none     poor
21 31          PhD student       none     none     good
22 30          PhD student       none     none     good

      Q6
1          I am a regular user
2          I know it exists
3          I know it exists
4 Is this supposed to be in the cloud?!
5          I know it exists
6          Once in a while i used it
7          I am a regular user
8          I know it exists
9          I know it exists
10 Is this supposed to be in the cloud?!
11         I know it exists
12         I know it exists
13 Is this supposed to be in the cloud?!
14         I know it exists

```

```

15           I know it exists
16           I know it exists
17           I know it exists
18           I am a regular user
19 Is this supposed to be in the cloud?!
20 Is this supposed to be in the cloud?!
21   I heard we had some servers around
22 Is this supposed to be in the cloud?!
# if you don't know we had 7 columns...
surveyrbioc[, -ncol(surveyrbioc)]
      Q1          Q2          Q3          Q4          Q5
1 28    PhD student intermediate intermediate good
2 27    PhD student           poor intermediate poor
3 33    Postdoc            good           good good
4 32    PhD student           poor           poor poor
5 29    PhD student           none           poor poor
6 33    Postdoc intermediate intermediate intermediate
7 40    Postdoc            good           good intermediate
8 23 Master student/else     poor           poor good
9 27    PhD student           none           poor intermediate
10 23 Master student/else     poor           poor poor
11 35    Postdoc            poor           poor intermediate
12 34 Master student/else     poor           pro  poor
13 31    Postdoc            none           none intermediate
14 27    PhD student           none           poor poor
15 24 Master student/else     poor intermediate intermediate
16 28    PhD student           none           poor poor
17 28 Master student/else     intermediate intermediate intermediate
18 39    Postdoc            good intermediate good
19 32 Master student/else     none           poor genoWhat?
20 29    PhD student           poor           none poor
21 31    PhD student           none           none good
22 30    PhD student           none           none good
      Q6
1           I am a regular user
2           I know it exists
3           I know it exists
4 Is this supposed to be in the cloud?!
5           I know it exists
6 Once in a while i used it
7           I am a regular user
8           I know it exists
9           I know it exists
10 Is this supposed to be in the cloud?!
11           I know it exists
12           I know it exists
13 Is this supposed to be in the cloud?!
14           I know it exists
15           I know it exists

```

```

16           I know it exists
17           I know it exists
18           I am a regular user
19 Is this supposed to be in the cloud?!
20 Is this supposed to be in the cloud?!
21     I heard we had some servers around
22 Is this supposed to be in the cloud?!

# you can subset with logical vectors, by row and by column
surveyrbioc[c(rep(TRUE,10),rep(FALSE,8)),]

      Q1          Q2          Q3          Q4          Q5
1 28    PhD student intermediate intermediate   good
2 27    PhD student            poor intermediate  poor
3 33    Postdoc             good        good   good
4 32    PhD student            poor        poor  poor
5 29    PhD student            none        poor  poor
6 33    Postdoc intermediate intermediate intermediate
7 40    Postdoc             good        good intermediate
8 23 Master student/else    poor        poor   good
9 27    PhD student            none        poor intermediate
10 23 Master student/else    poor        poor  poor
19 32 Master student/else    none        poor  genoWhat?
20 29    PhD student            poor        none  poor
21 31    PhD student            none        none   good
22 30    PhD student            none        none   good

      Q6
1           I am a regular user
2           I know it exists
3           I know it exists
4 Is this supposed to be in the cloud?!
5           I know it exists
6 Once in a while i used it
7           I am a regular user
8           I know it exists
9           I know it exists
10 Is this supposed to be in the cloud?!
19 Is this supposed to be in the cloud?!
20 Is this supposed to be in the cloud?!
21     I heard we had some servers around
22 Is this supposed to be in the cloud?!

Q7
1
Learn Parallelization with R (and Bioconductor)
2
<NA>
3
use R scripts in parallel context (ex: alignments in RNA-seq)
4

```

I'd like to gather practise to analyse count data and develope the necessary data design. Additionally, it would be interesting to get to know how to find putative novel miRNAs.

5

Possible I will use R **for** editing RNA-seq data

6

I would describe myself as an advanced beginner, I am actively using R now to look (mostly) at count tables from NGS data and make plots. I am especially interested **in** the Bioconductor and parallelization **in** R section, this is new **for** me.

7

<NA>

8

Get to know R better, basic commands

9

To look **if** I can process my sequencing data on my own using R

10 I would like to learn more about R, especially how to use it **in** biomedical research. In the 2nd Semester of the Rasterprogramm biomedicine, last Semester I had two weeks bioinformatics and we used to work with R **for** statistic s/ChIP-Seq/microarray-data analysis, but the time was too short, to go deep into it, we just scratched the surface. So now I wish to learn some more, would be great, **if** I could work with the programme by myself, **for** example **for** the masterthesis.

19

<NA>

20

To better understand R and perform basic analysis

21

working with fastq files based on R

22

I want to be able to analyze my sequence data on my own :P

surveyrbioc[c(TRUE, FALSE),] # keep in mind this behavior!

| | Q1 | Q2 | Q3 | Q4 | Q5 |
|----|----|-----------------------------|--------------|--------------|--------------|
| 1 | 28 | PhD student | intermediate | intermediate | good |
| 3 | 33 | Postdoc | good | good | good |
| 5 | 29 | PhD student | none | poor | poor |
| 7 | 40 | Postdoc | good | good | intermediate |
| 9 | 27 | PhD student | none | poor | intermediate |
| 11 | 35 | Postdoc | poor | poor | intermediate |
| 13 | 31 | Postdoc | none | none | intermediate |
| 15 | 24 | Master student/ else | poor | intermediate | intermediate |
| 17 | 28 | Master student/ else | intermediate | intermediate | intermediate |
| 19 | 32 | Master student/ else | none | poor | genoWhat? |
| 21 | 31 | PhD student | none | none | good |

Q6

1 I am a regular user
3 I know it exists
5 I know it exists
7 I am a regular user

```

9           I know it exists
11          I know it exists
13 Is this supposed to be in the cloud?!
15          I know it exists
17          I know it exists
19 Is this supposed to be in the cloud?!
21      I heard we had some servers around

Q7
1                               Learn Parallelization with R (and Bioc
onductor)
3           use R scripts in parallel context (ex: alignments in
RNA-seq)
5           Possible I will use R for editing RNA
-seq data
7
<NA>
9           To look if I can process my sequencing data on my ow
n using R
11          learn mor
e about R
13 to get a good and understandable introduction into R programming and bioin
formatics
15          To learn the basics of R pr
ogramming
17          better understanding of R and to extend my
knowledge
19
<NA>
21          working with fastq files b
ased on R

# guess what this does?
surveyrbioc$Q2=="PhD student"
[1] TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
FALSE  TRUE FALSE  TRUE
[17] FALSE FALSE FALSE  TRUE  TRUE  TRUE

```

5.3 Exercise session 3

- How many PhD students did reply?
- What is the proportion of PhD students to all other participants?
- How old are they on average?
- How many of the participants are older than 30?
- How many postdocs are younger than 35?
- How many of the participants did not reply to the last question?

5.3.1 Exercise Session 3 - Solutions

► Details

5.4 Manipulating and analysing your data

You can

- sort the data (see `sort` and `order`)
- transform your data: apply rules (formulas, logics, insight altogether)
- combine two datasets or more (if you `merge` them)
- do some statistics on your data

5.5 Sorting the data

Hide

```
myord <- order(surveyrbioc$Q1)
myord
[1]  8 10 15  2  9 14  1 16 17  5 20 22 13 21  4 19  3  6 12 11 18  7

head(surveyrbioc[myord,1:5],4)
      Q1          Q2    Q3          Q4          Q5
8 23 Master student/else poor       poor       good
10 23 Master student/else poor       poor       poor
15 24 Master student/else poor intermediate intermediate
2 27           PhD student poor intermediate       poor
sorted_surv <- surveyrbioc[myord,1:6]
```

`sort()` returns you the sorted data, `order()` the indices only

5.6 Transforming the data

Hide

```

# transforming a variable
newsurvey <- surveyrbioc[,1:5]
newsurvey$ageroot <- sqrt(newsurvey$Q1)
head(newsurvey)

      Q1        Q2        Q3        Q4        Q5    ageroot
1 28 PhD student intermediate intermediate good 5.291503
2 27 PhD student           poor intermediate poor 5.196152
3 33 Postdoc             good       good good 5.744563
4 32 PhD student           poor       poor poor 5.656854
5 29 PhD student           none       poor poor 5.385165
6 33 Postdoc intermediate intermediate intermediate 5.744563

# creating groups out of a continuous variable
newsurvey$agegroup <- cut(newsurvey$Q1,breaks = c(20,30,40))
head(newsurvey)

      Q1        Q2        Q3        Q4        Q5    ageroot agegroup
1 28 PhD student intermediate intermediate good 5.291503 (20,30]
2 27 PhD student           poor intermediate poor 5.196152 (20,30]
3 33 Postdoc             good       good good 5.744563 (30,40]
4 32 PhD student           poor       poor poor 5.656854 (30,40]
5 29 PhD student           none       poor poor 5.385165 (20,30]
6 33 Postdoc intermediate intermediate intermediate 5.744563 (30,40]

```

Use case for `merge`: you have *two* sets you are playing with! Think in advance what you need for that purpose...

5.7 We want statistics!

Are PhD students *significantly* younger than postdocs? Are there any differences in the age of the three groups?

[Hide](#)

```

phds <- surveyrbioc[surveyrbioc$Q2=="PhD student",]
postdocs <- surveyrbioc[surveyrbioc$Q2=="Postdoc",]
t.test(phds$Q1,postdocs$Q1)

Welch Two Sample t-test

data: phds$Q1 and postdocs$Q1
t = -4.0528, df = 6.4476, p-value = 0.005767
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-10.146796 -2.586537
sample estimates:
mean of x mean of y
28.80000 35.16667
aov(data=surveyrbioc,Q1~Q2) # What is missing here?
Call:
aov(formula = Q1 ~ Q2, data = surveyrbioc)

Terms:
          Q2 Residuals
Sum of Squares 216.8242 207.7667
Deg. of Freedom      2         19

Residual standard error: 3.306824
Estimated effects may be unbalanced

```

Much more on this: in the next courses!

5.8 Simple yet powerful functions

`tapply`

You want to calculate the median age of each academic group in here

[Hide](#)

```

md <- median(surveyrbioc$Q1)
md_master <- median(surveyrbioc$Q1[surveyrbioc$Q2=="Master student/else"])
md_phd <- median(surveyrbioc$Q1[surveyrbioc$Q2=="PhD student"])
md_postdocs <- median(surveyrbioc$Q1[surveyrbioc$Q2=="Postdoc"])
c(md_master,md_phd,md_postdocs)
[1] 26.0 28.5 34.0

```

`tapply` splits the data of the first variable on the levels of the second variable, and applies the function (*any* function)

[Hide](#)

```
tapply(X = surveyrbioc$Q1, INDEX = surveyrbioc$Q2, FUN = median)
Master student/else          PhD student           Postdoc
                26.0              28.5              34.0
```

lapply and sapply

Back to our iris dataset

[Hide](#)

```
names(iris)
[1] "Sepal.Length" "Sepal.Width"   "Petal.Length" "Petal.Width"   "Species"
```

We want the average sepal length and width, and the same for the petals. Uh, and we want the standard deviation too.

[Hide](#)

```
# the inefficient way:
sepplen_m <- mean(iris$Sepal.Length)
sepwid_m <- mean(iris$Sepal.Width)
petlen_m <- mean(iris$Petal.Length)
petwid_m <- mean(iris$Petal.Width)

sepplen_m <- sd(iris$Sepal.Length)
# ... and so on
```

-> Apply a Function over a List or Vector

[Hide](#)

```
# we will use just the first four columns
lapply(iris[,1:4],mean)
$Sepal.Length
[1] 5.843333

$Sepal.Width
[1] 3.057333

$Petal.Length
[1] 3.758

$Petal.Width
[1] 1.199333
sapply(iris[,1:4],mean)
Sepal.Length Sepal.Width Petal.Length Petal.Width
      5.843333      3.057333      3.758000      1.199333
lapply(iris[,1:4],sd)
$Sepal.Length
[1] 0.8280661

$Sepal.Width
[1] 0.4358663

$Petal.Length
[1] 1.765298

$Petal.Width
[1] 0.7622377
# ...
```

The major difference is in the presentation of the output

```
summary
```

Try out `summary` on a `data.frame`

[Hide](#)

```

summary(iris)
   Sepal.Length      Sepal.Width       Petal.Length      Petal.Width      Specie
   Min.    :4.300      Min.    :2.000      Min.    :1.000      Min.    :0.100      setosa    :50
   1st Qu.:5.100      1st Qu.:2.800      1st Qu.:1.600      1st Qu.:0.300      versicolor:50
   Median  :5.800      Median  :3.000      Median  :4.350      Median  :1.300      virginica :50
   Mean    :5.843      Mean    :3.057      Mean    :3.758      Mean    :1.199
   3rd Qu.:6.400      3rd Qu.:3.300      3rd Qu.:5.100      3rd Qu.:1.800
   Max.    :7.900      Max.    :4.400      Max.    :6.900      Max.    :2.500

```

Alternatives in other packages:

- `describe()` in the `Hmisc` package
- `skim()` from `skimr`
- `create_report()` from Data Explorer

`table`

Hide

```
table(surveyrbioc$Q3)
```

| good | intermediate | none | poor |
|------|--------------|------|------|
| 3 | 3 | 8 | 8 |

```
table(surveyrbioc$Q4)
```

| good | intermediate | none | poor | pro |
|------|--------------|------|------|-----|
| 2 | 6 | 4 | 9 | 1 |

```
table(surveyrbioc$Q2, surveyrbioc$Q3)
```

| | good | intermediate | none | poor |
|---------------------|------|--------------|------|------|
| Master student/else | 0 | 1 | 1 | 4 |
| PhD student | 0 | 1 | 6 | 3 |
| Postdoc | 3 | 1 | 1 | 1 |

- want the sums? Try `addmargins()`
- looking for the percentage values? `prop.table()`
- somewhat nicer output: `ftable()`

Hide

```

addmargins(table(surveyrbioc$Q2,surveyrbioc$Q3))

      good intermediate none poor Sum
Master student/else    0          1    1    4   6
PhD student           0          1    6    3  10
Postdoc               3          1    1    1   6
Sum                   3          3    8    8  22

prop.table(table(surveyrbioc$Q2,surveyrbioc$Q3))

      good intermediate     none     poor
Master student/else 0.00000000 0.04545455 0.04545455 0.18181818
PhD student         0.00000000 0.04545455 0.27272727 0.13636364
Postdoc             0.13636364 0.04545455 0.04545455 0.04545455

```

Please always do check the docs!

5.9 Exercise session 4

The `MASS` package contains the dataset `Cars93`, which stores the data on 93 makes of car sold in US

- you'll need the package *and* the data
- Type `Market` specifies the type of market the car is aimed at. Find the cheapest car in each type, and the one with the greatest fuel efficiency
- compute the mean horsepower for each type
- create two `data.frames`, one for US cars, the other one with non-US cars
- export the US cars to a text file
- save the non-US cars data to a binary file (`.RData`)

5.9.1 Exercise Session 4 - Solutions

► Details

6 Step 4: Plotting data

6.1 Graphics in R

- powerful environment for visualizing scientific data
- integrated graphics **AND** statistics
- publication-ready quality
- fully programmable, highly reproducible

Many ways for the same task:

- `base graphics (plot)`
- `ggplot2`
- `lattice`

- interactive visualizations such as `plotly`, `ggvis` or other libraries

Why bother plotting at all?

- facilitate comparisons
- identify trends
- generate hypotheses

6.2 You can do all this with R

Look for some inspiration here, it is an excellent place to start and learn!

<https://r-graph-gallery.com/> (<https://r-graph-gallery.com/>)

6.3 The `plot` function

First thing: take a look at the overview documentation of `plot`

[Hide](#)

```
?plot
```

We will see

- scatter plots
- boxplots
- barplots
- histograms

6.4 `plot` parameters

Required:

- x variable
- y variable

Other options

- title with `main`
- axes labels with `xlab` and `ylab`
- axes limits with `xlim` and `ylim`
- symbols, colors and sizes: `pch`, `col` and `cex` - as atomic elements or as vectors

6.5 Get to know the data: `mpg`

[Hide](#)

```
library(ggplot2) # this is useful per se, and contains the dataset we will be  
using  
?mpg
```

This dataset contains a subset of the fuel economy data that the EPA makes available on <http://fueleconomy.gov>

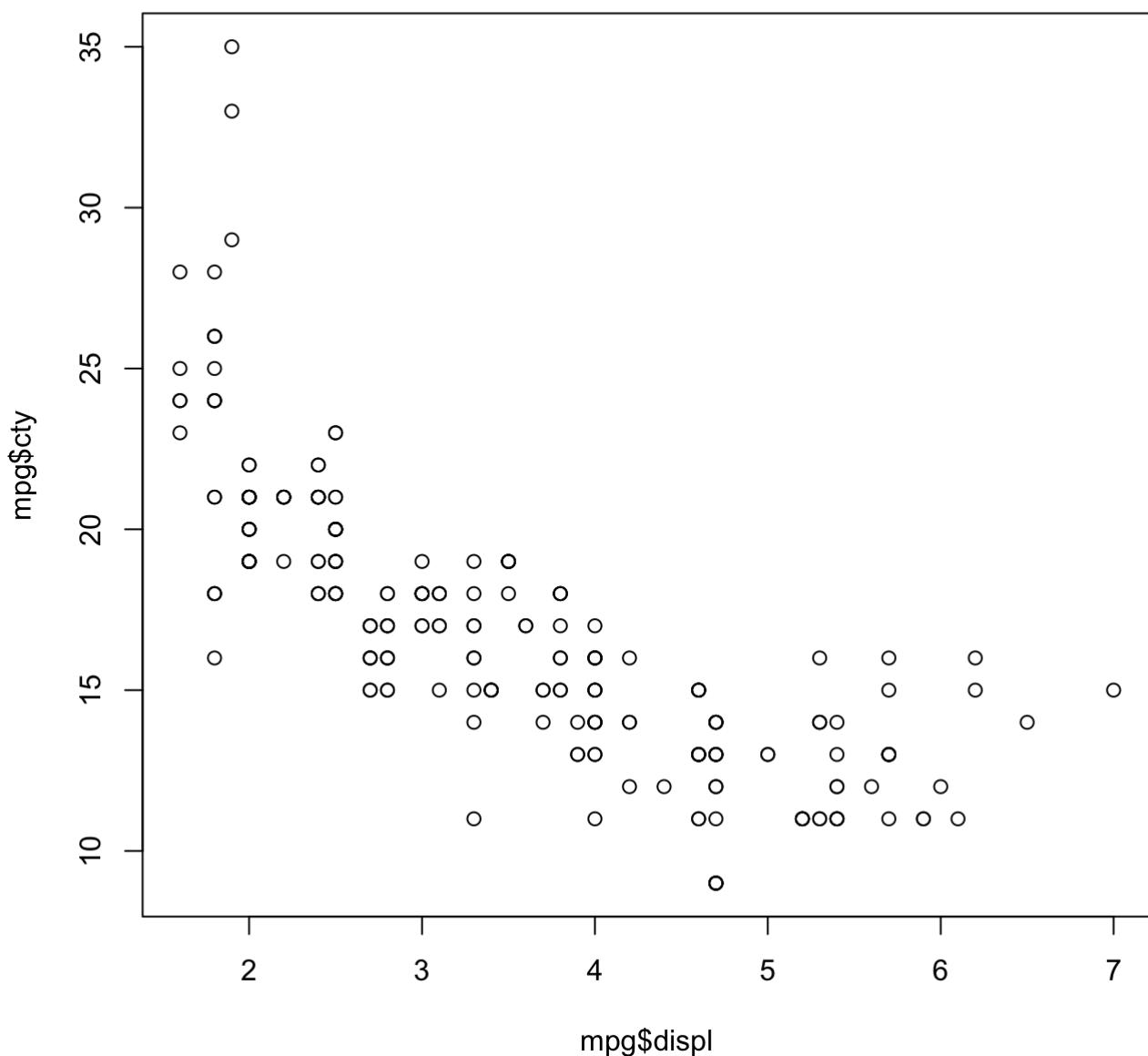
Hide

Make a guess: what do you expect to see between fuel consumption and engine size?

6.6 Scatter plots

Hide

```
plot(mpg$displ, mpg$cty)
```



Bonus: what is the `cor` relation?

[Hide](#)

```
cor(mpg$displ,mpg$cty)
[1] -0.798524
cor(mpg$displ,mpg$cty,method="spearman")
[1] -0.8809049
```

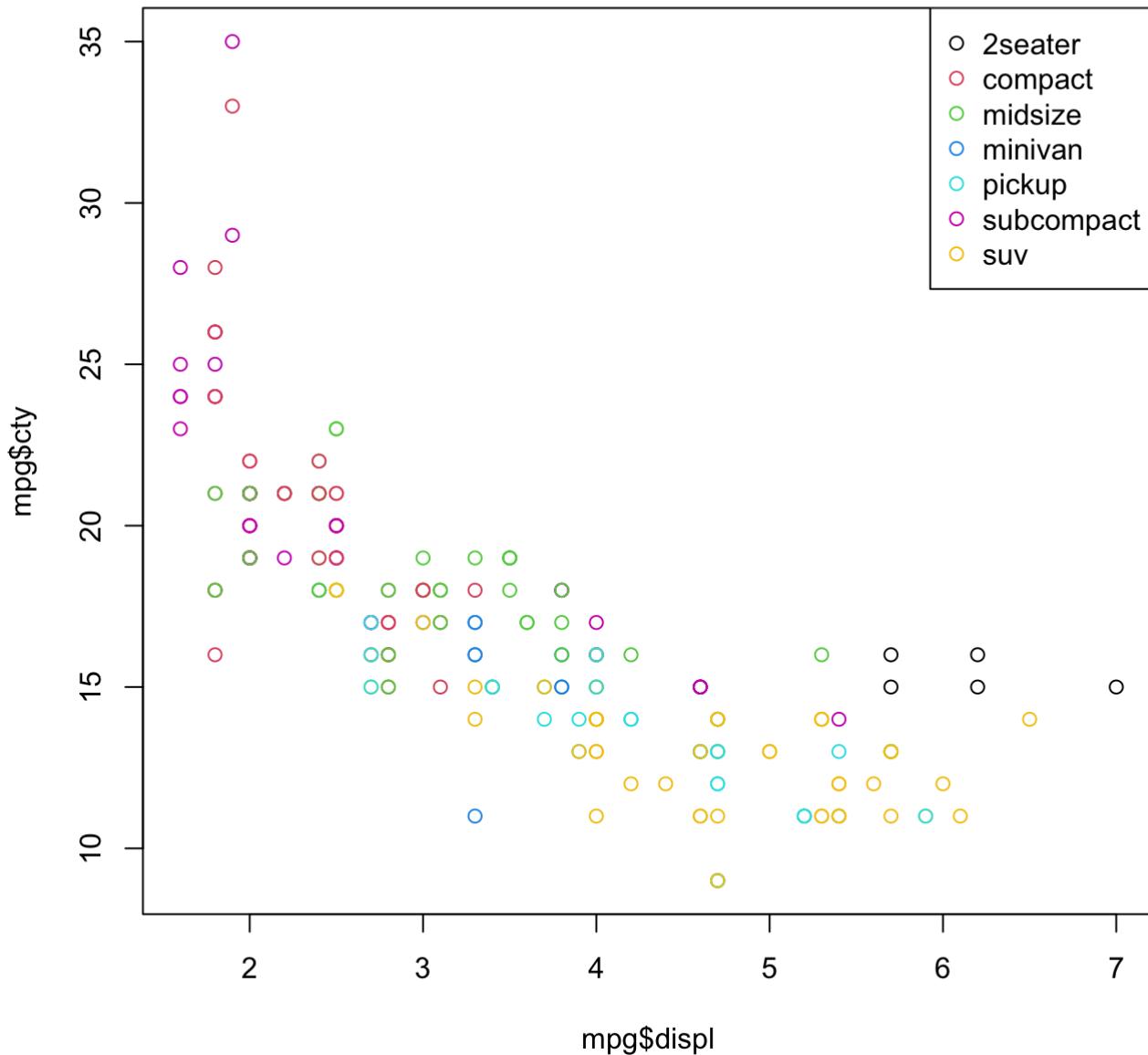
6.6.1 Can we do more?

[Hide](#)

```

mpg$mygroup <- as.numeric(factor(mpg$class))
plot(mpg$displ,mpg$cty,
      col = mpg$mygroup)
legend("topright",legend = levels(factor(mpg$class)),col=levels(factor(mpg$my
group)),pch=1)

```

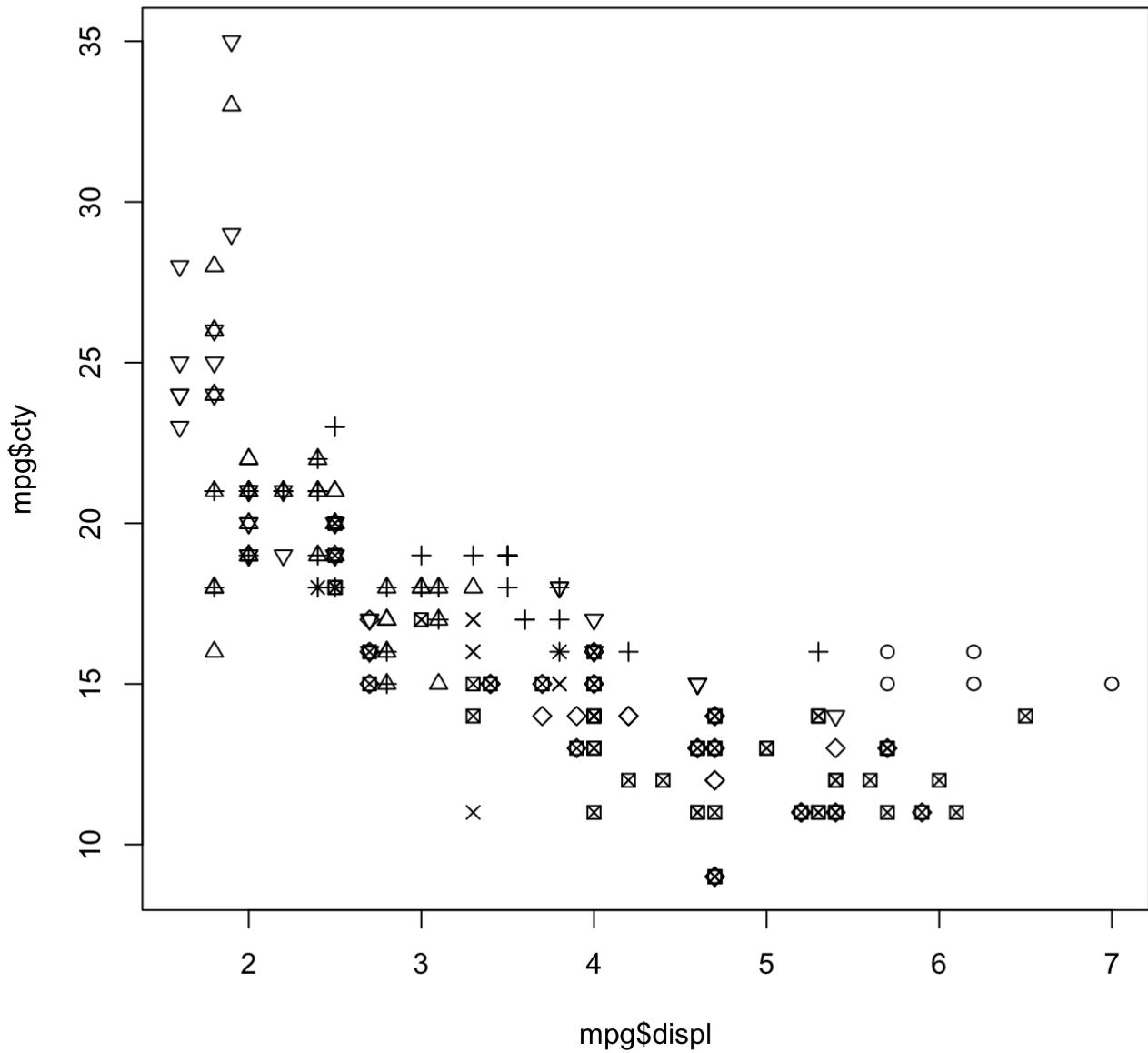


[Hide](#)

```

plot(mpg$displ,mpg$cty,
      pch = as.numeric(factor((mpg$class))))

```



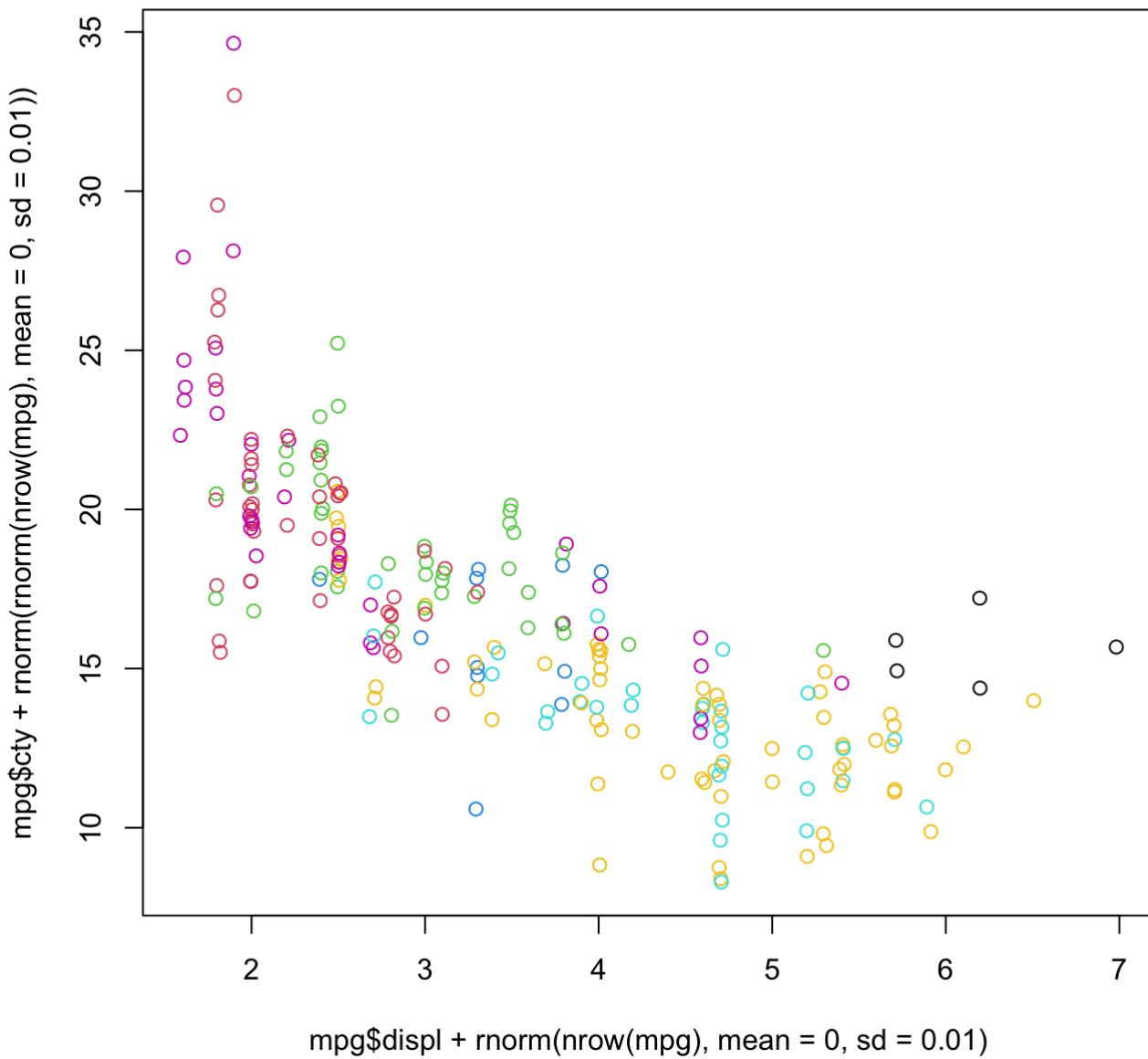
This shows we have quite some overlap of points. What can we do?

Adding some jitter...

[Hide](#)

```
plot(x = mpg$displ + rnorm(nrow(mpg), mean = 0, sd = 0.01),
      y = mpg$cty + rnorm(rnorm(nrow(mpg), mean = 0, sd = 0.01)),
      col = mpg$mygroup,
      main = "now with jitter!")
```

now with jitter!



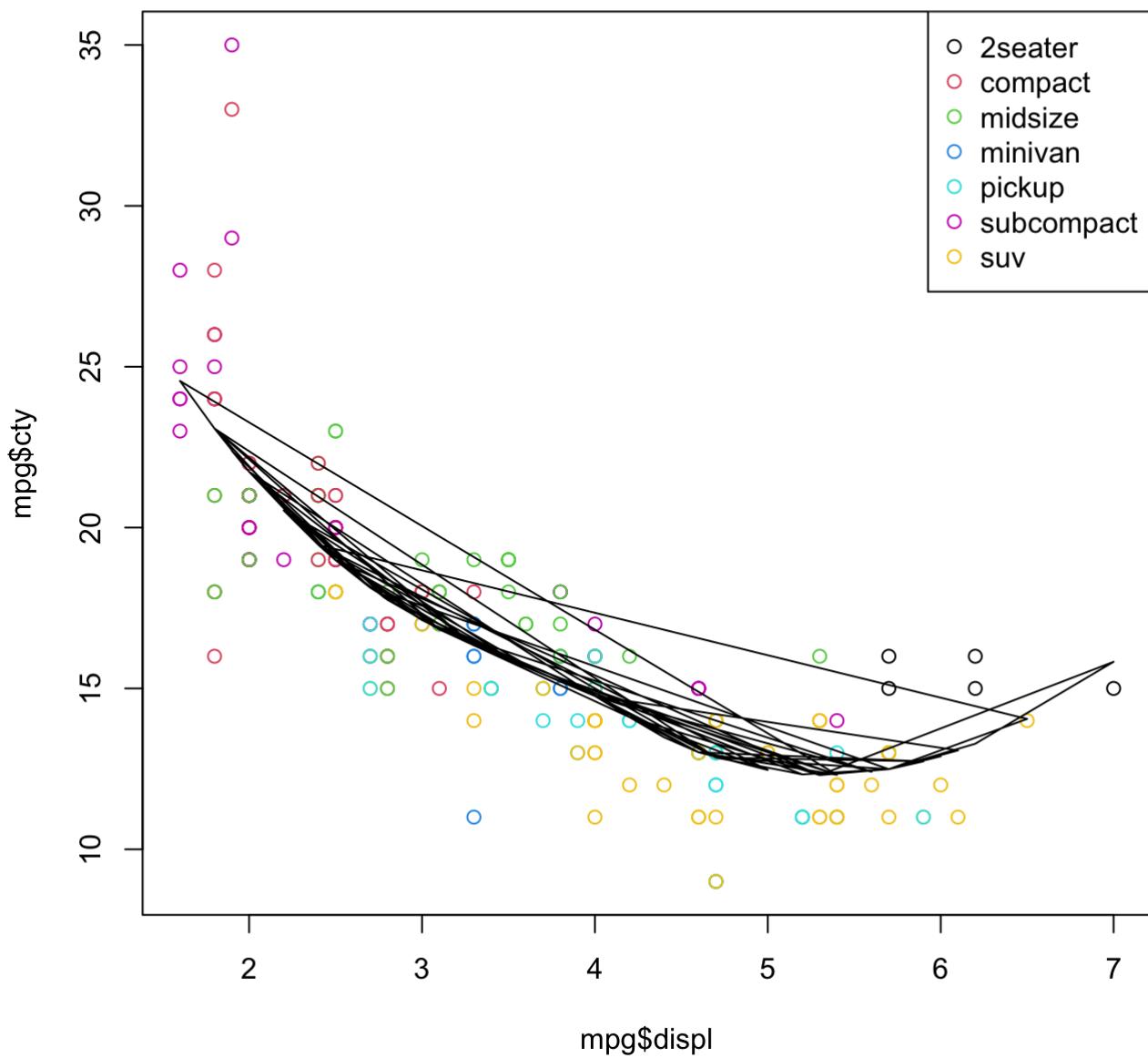
Adding a smoothing line

Trying to see a pattern? Add a smoothing curve.

This one is wrong - missing the reordering of points

[Hide](#)

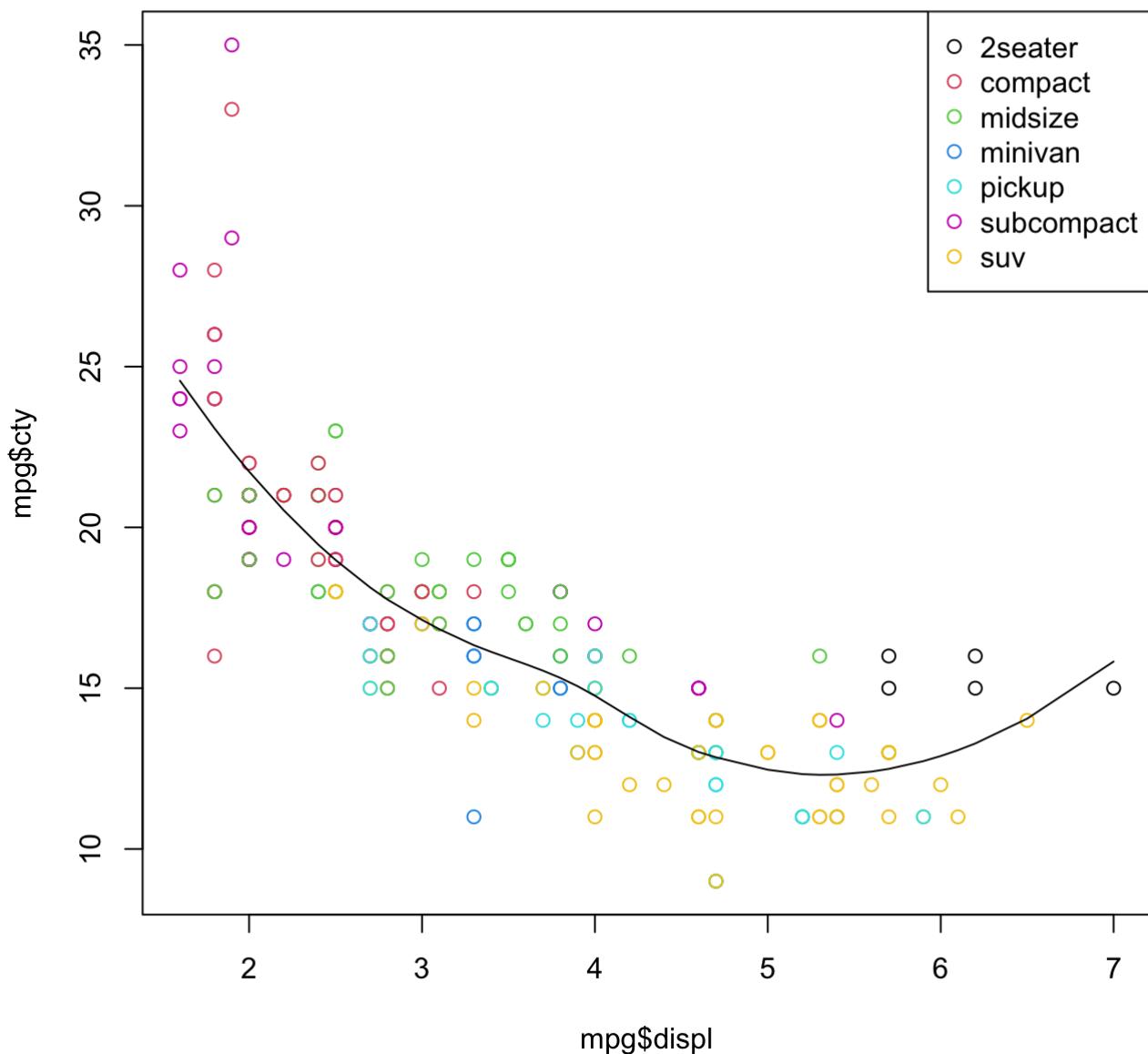
```
plot(mpg$displ,mpg$cty, col = mpg$mygroup)
myloess <- loess(cty~displ, data=mpg)
myfit <- fitted(myloess)
lines(mpg$displ,myfit)
legend("topright",legend = levels(factor(mpg$class)),col=levels(factor(mpg$my
group)),pch=1)
```



This one is correct!

[Hide](#)

```
plot(mpg$displ, mpg$cty, col = mpg$mygroup)
myloess <- loess(cty~displ, data=mpg)
myfit <- fitted(myloess)
myord <- order(mpg$displ)
lines(mpg$displ[myord], myfit[myord])
legend("topright", legend = levels(factor(mpg$class)), col=levels(factor(mpg$mygroup)), pch=1)
```



`lines` can add (almost) anything (any line).

`points` works in a similar way to superimpose, well, points

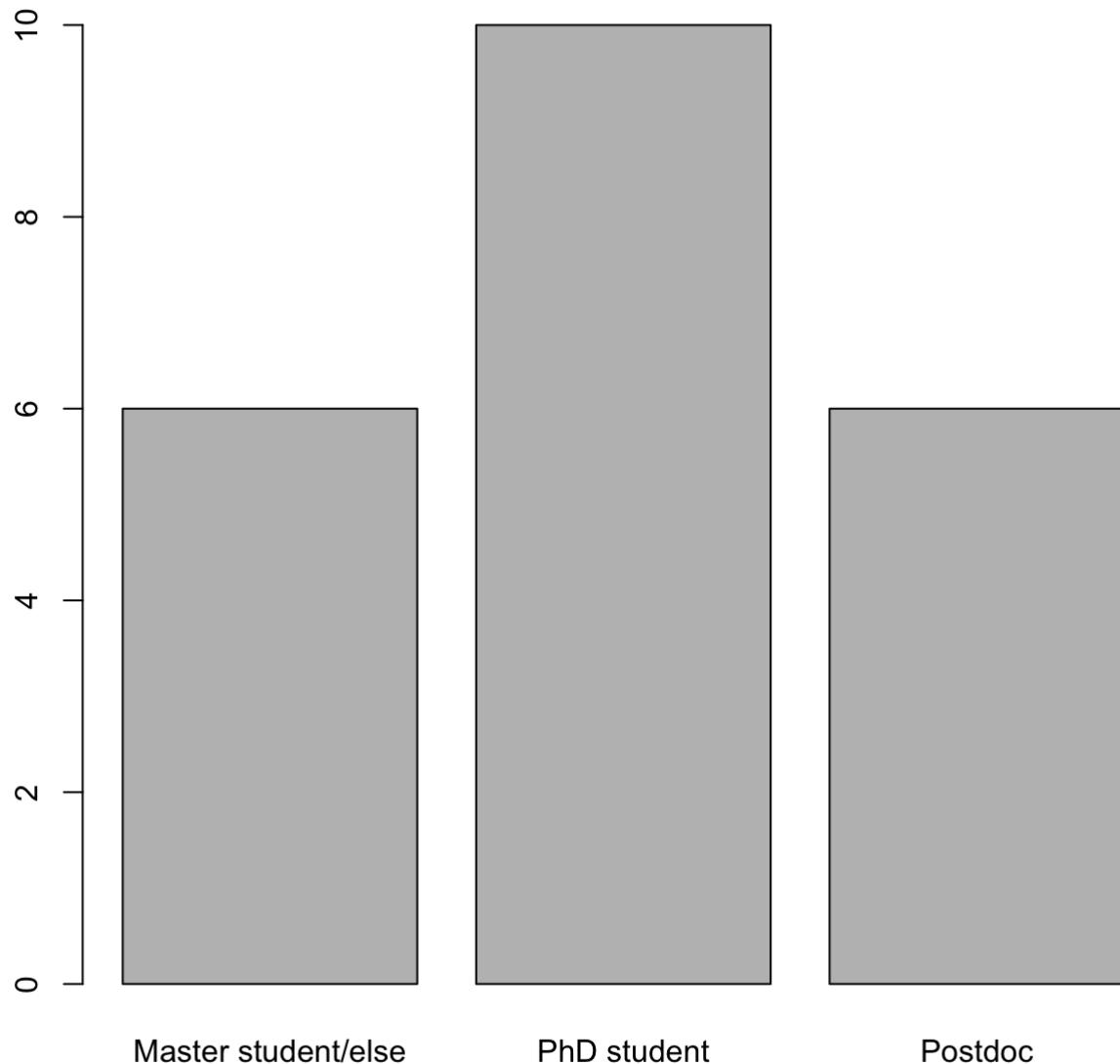
6.7 Bar charts

[Hide](#)

```
?barplot
```

[Hide](#)

```
academia_levels <- table(surveyrbioc$Q2)
barplot(academia_levels)
```

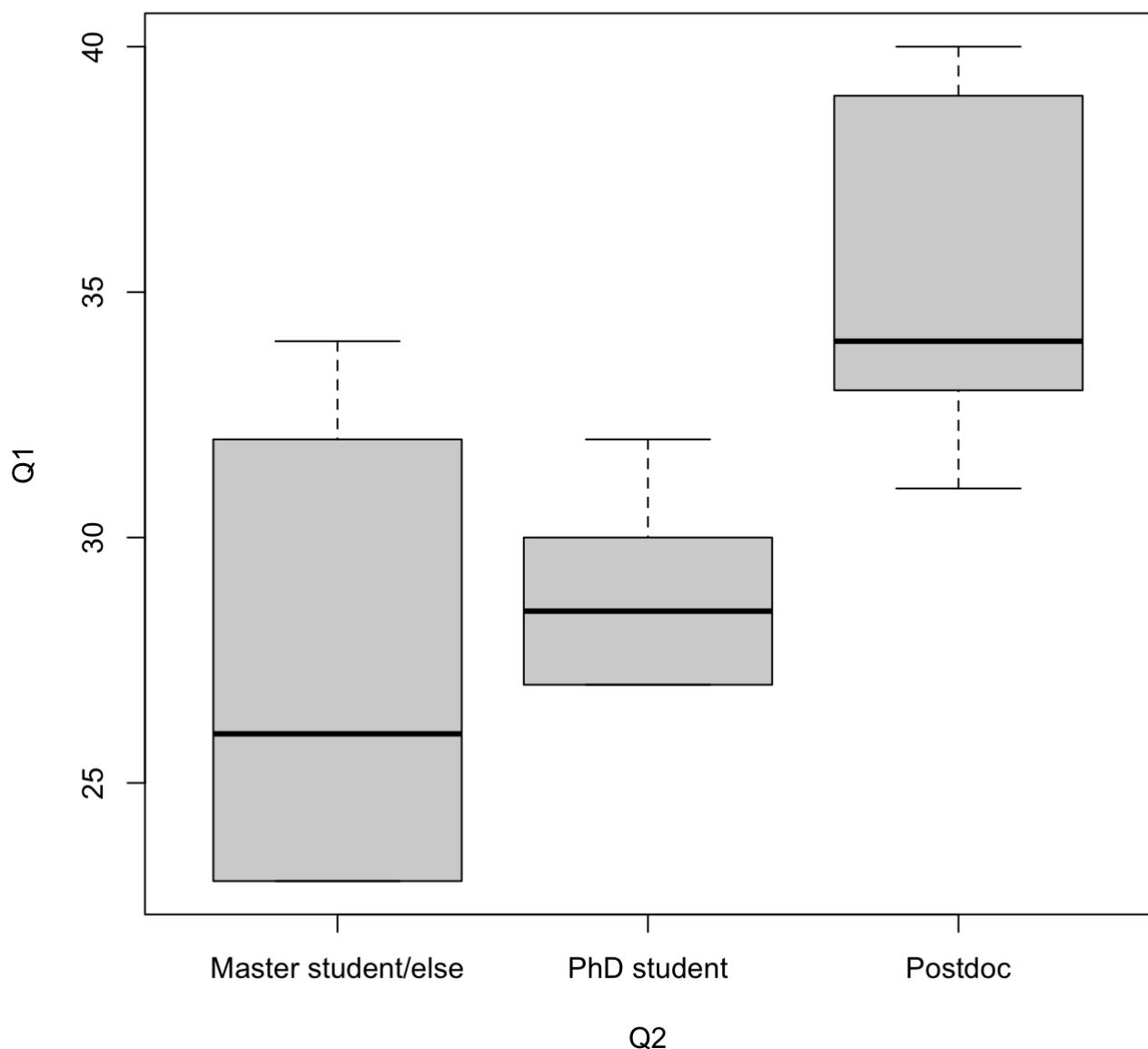


6.8 Boxplots

How is the age distributed across academic levels? Check the help of `boxplot`

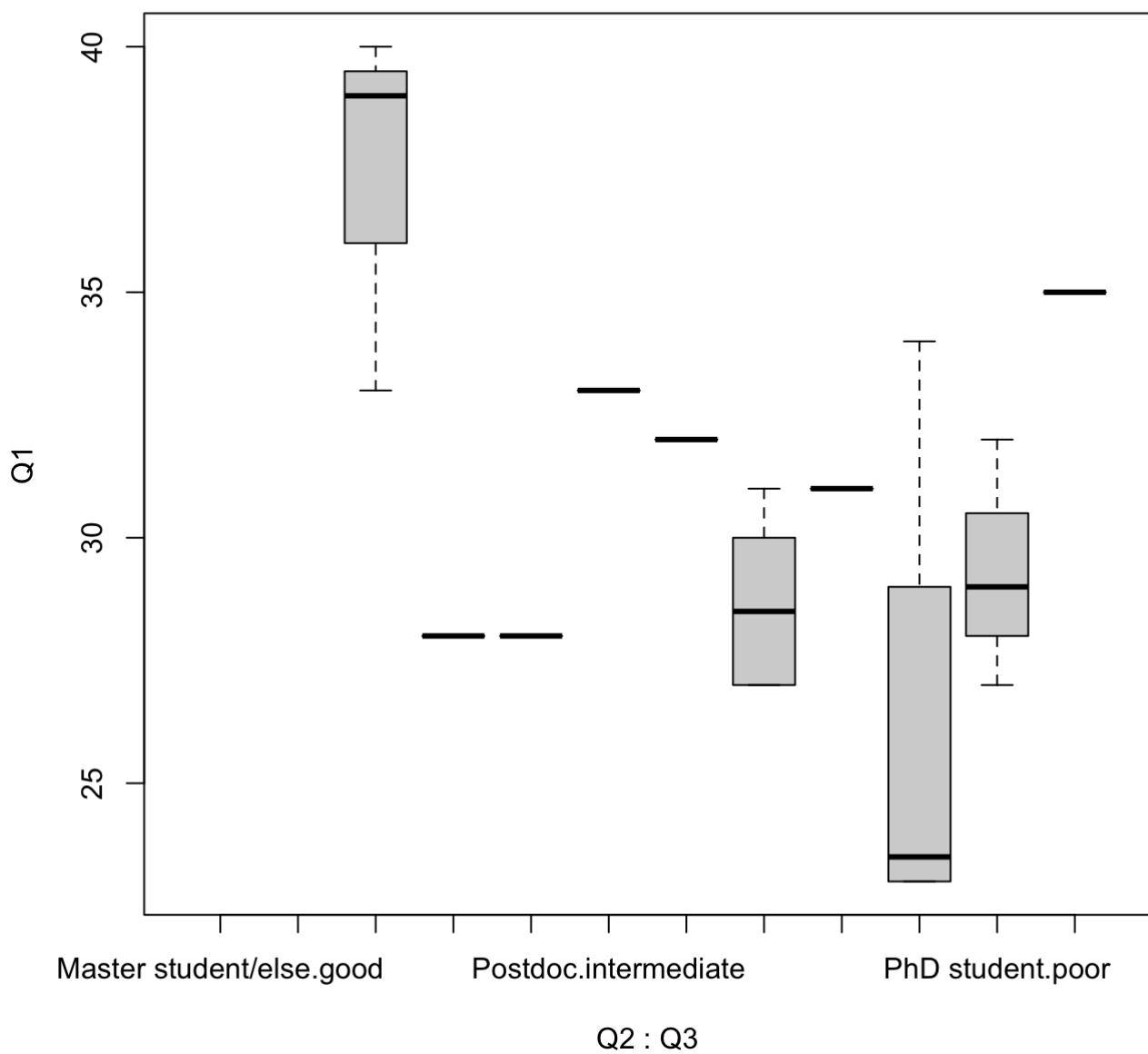
- A formula is required!
- Don't worry, it's nothing but your `y~x` variables - ok, it can get more complicated

```
boxplot(Q1~Q2,  
        data = surveyrbioc)
```



Splitting on more factors

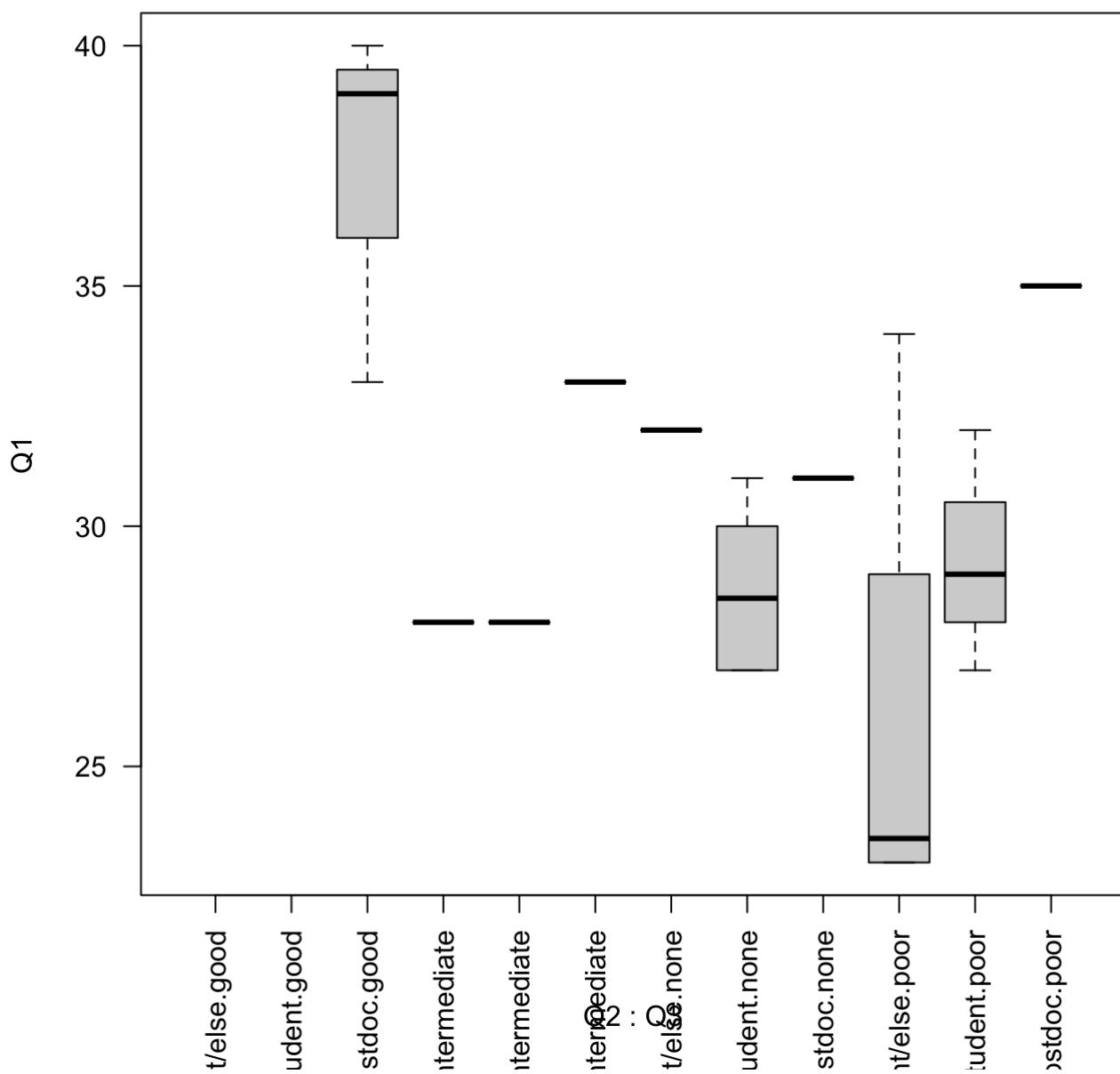
```
boxplot(Q1~Q2+Q3,  
        data = surveyrbio)
```



Making it more readable...

[Hide](#)

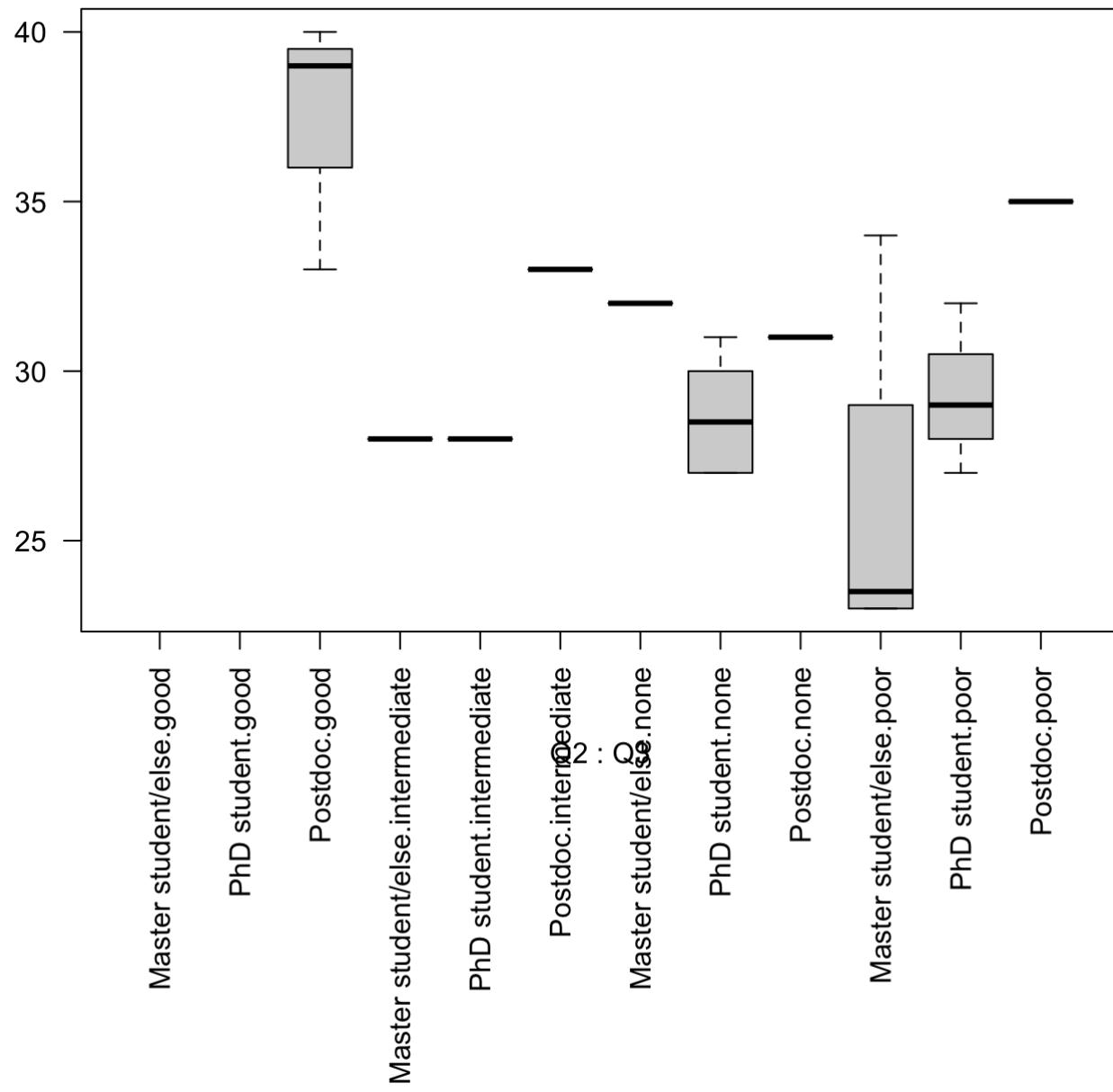
```
boxplot(Q1~Q2+Q3,  
        data = surveyrbio,  
        las = 2)
```



Changing the `par` ameters allows you to control many aspects on plot appearance `par` is your best friend - and enemy (see `?par`)

Hide

```
par(mar=c(15, 3, 2, 2))
boxplot(Q1~Q2+Q3, data = surveyrbioc, las = 2)
```



`par(...)` has many arguments; here, the useful/most used ones

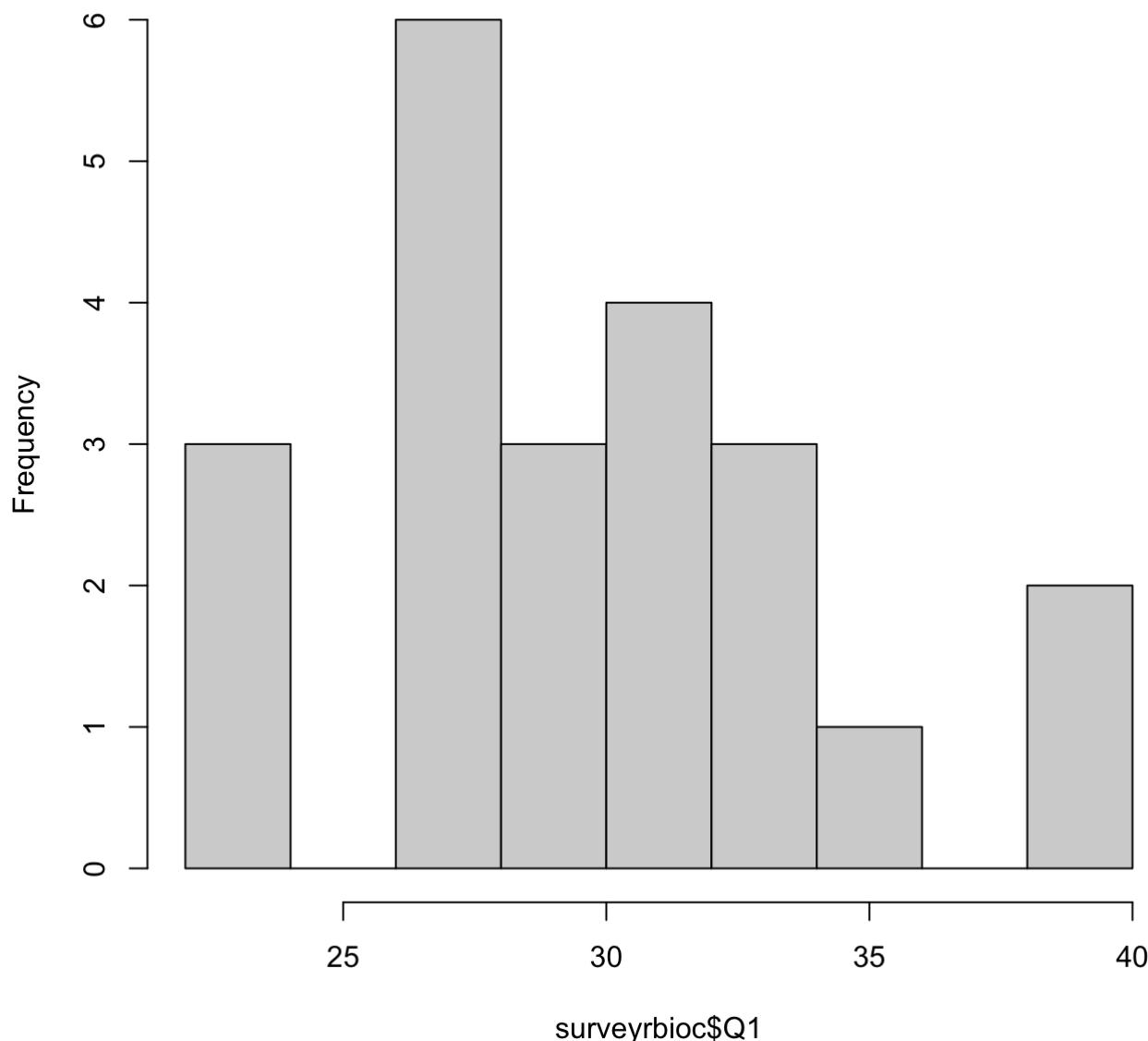
- `mar` for handling the margins
- `cex`, `col`, `pch` and co. are all parameters of `par`
- `las` to change the style of the axis labels
- `mfrow` to draw an array of figures

6.9 Histograms

Hide

```
hist(surveyrbioc$Q1, breaks = 8)
```

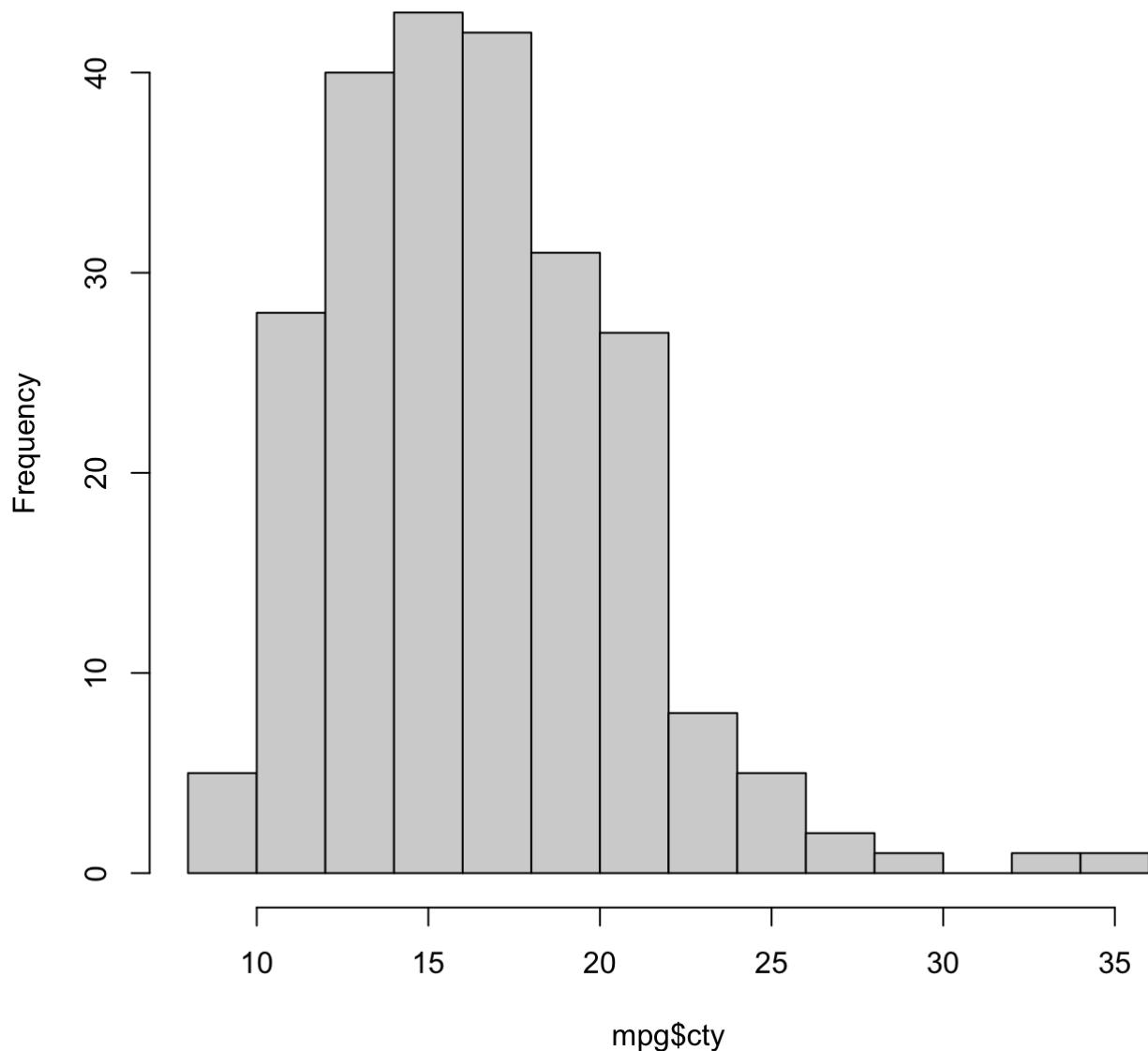
Histogram of surveyrbioc\$Q1



6.10 More histograms!

```
hist(mpg$cty, breaks = 10)
```

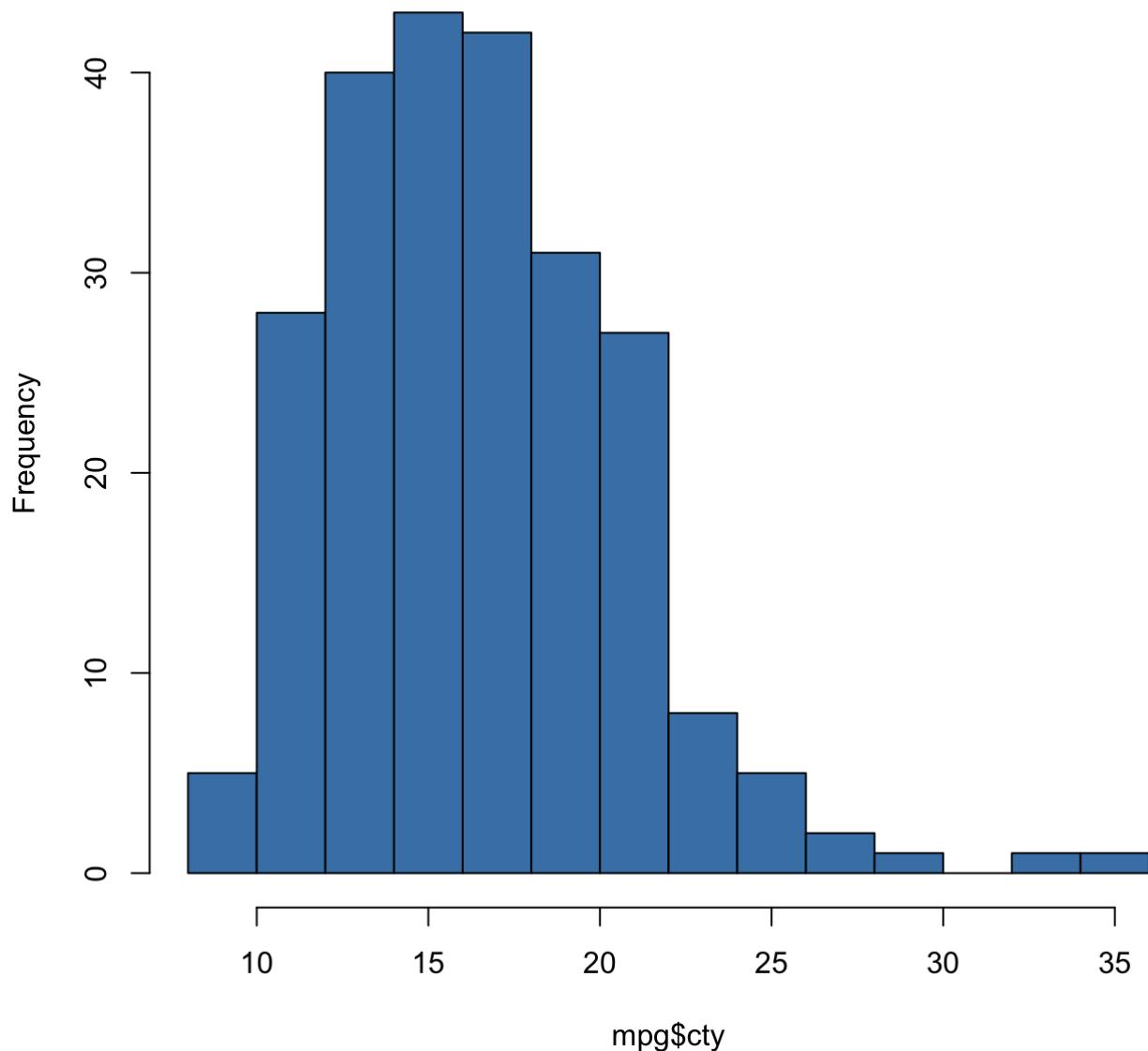
Histogram of mpg\$cty



[Hide](#)

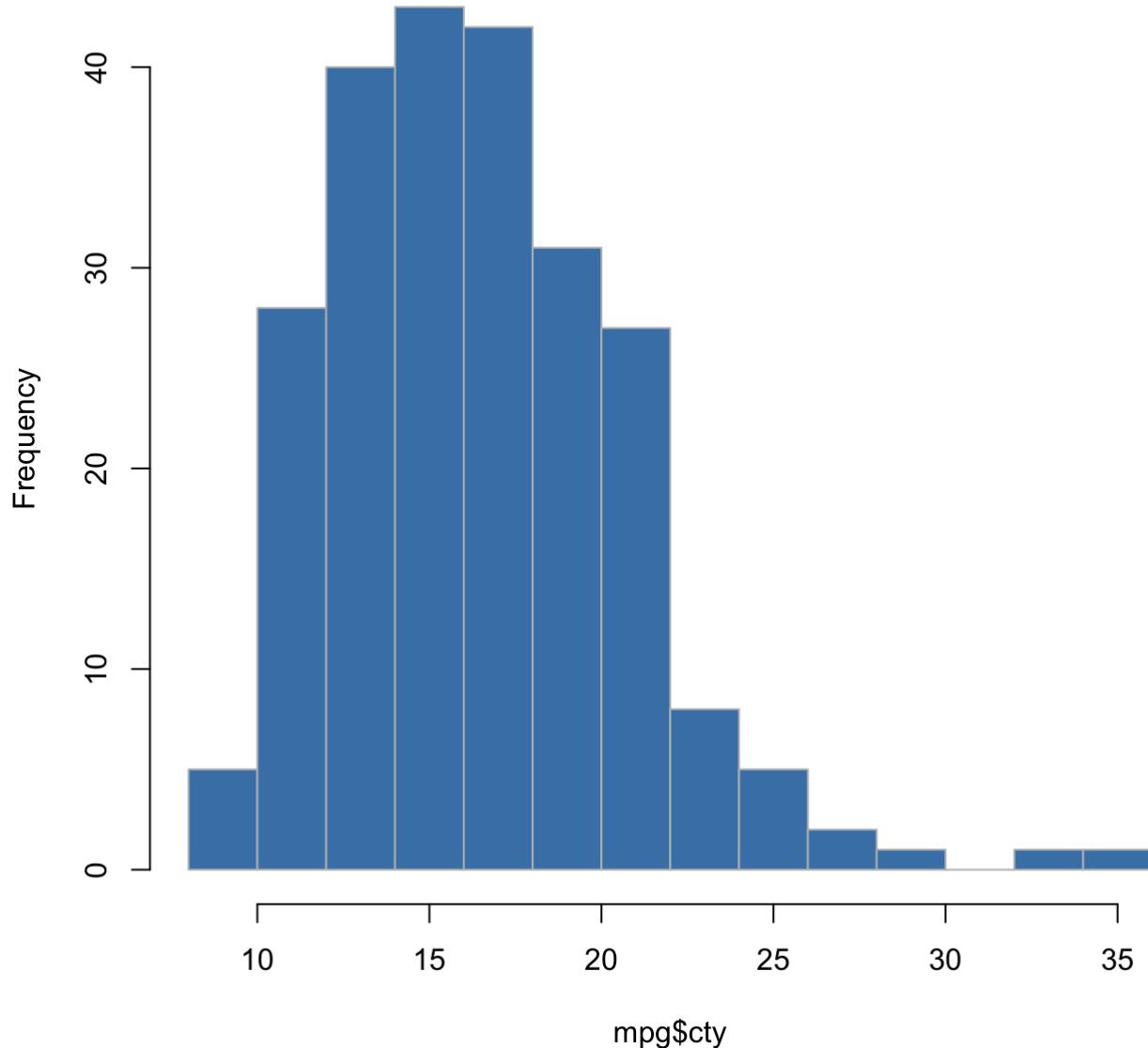
```
hist(mpg$cty, breaks = 10, col = "steelblue")
```

Histogram of mpg\$cty



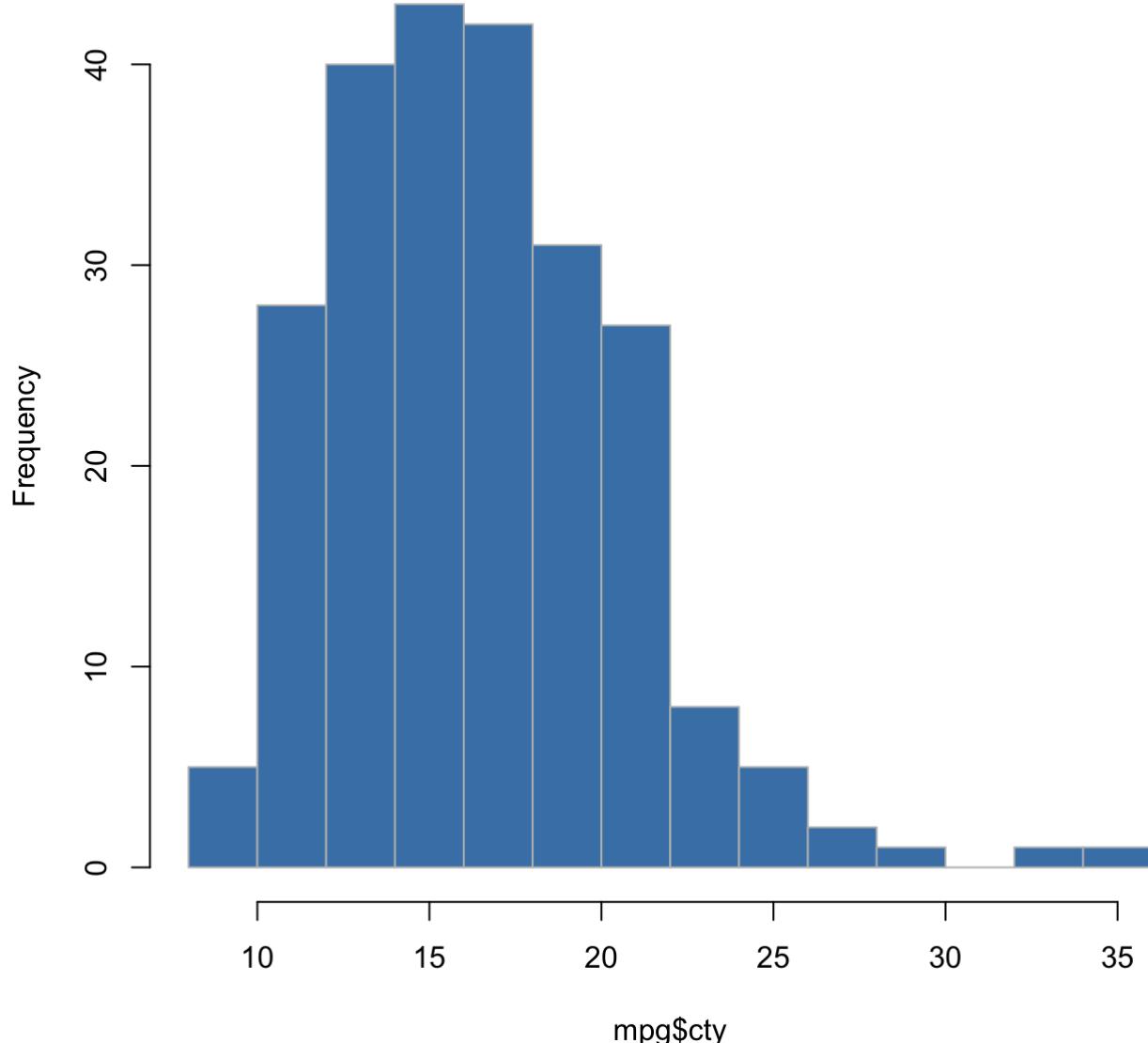
```
hist(mpg$cty, breaks = 10, col = "steelblue", border = "gray")
```

Histogram of mpg\$cty



```
hist(mpg$cty, breaks = 10, col = "steelblue", border = "gray", main = "Distribution of miles/gallon consumption in city traffic")
```

Distribution of miles/gallon consumption in city traffic



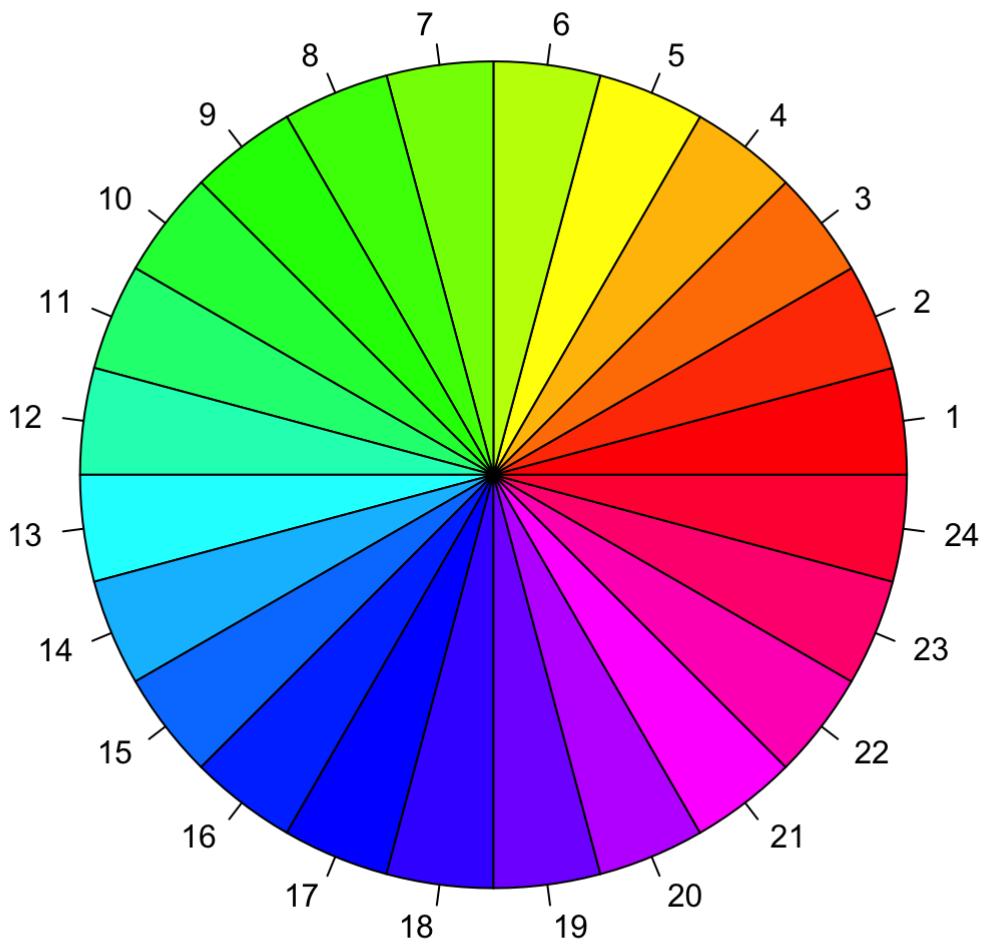
6.11 How to do nice pie charts

DON'T.

If you **really** need to do it...

[Hide](#)

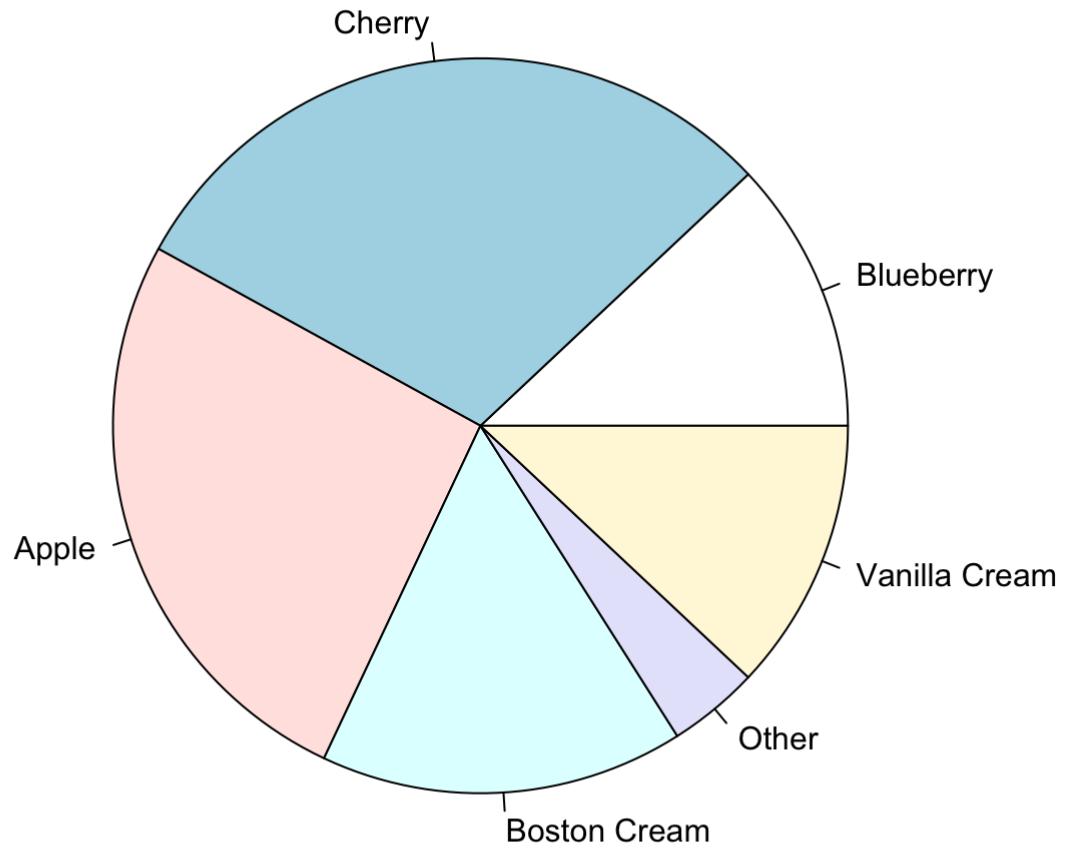
```
?pie  
example(pie) # especially the last one  
  
pie> require(grDevices)  
  
pie> pie(rep(1, 24), col = rainbow(24), radius = 0.9)
```

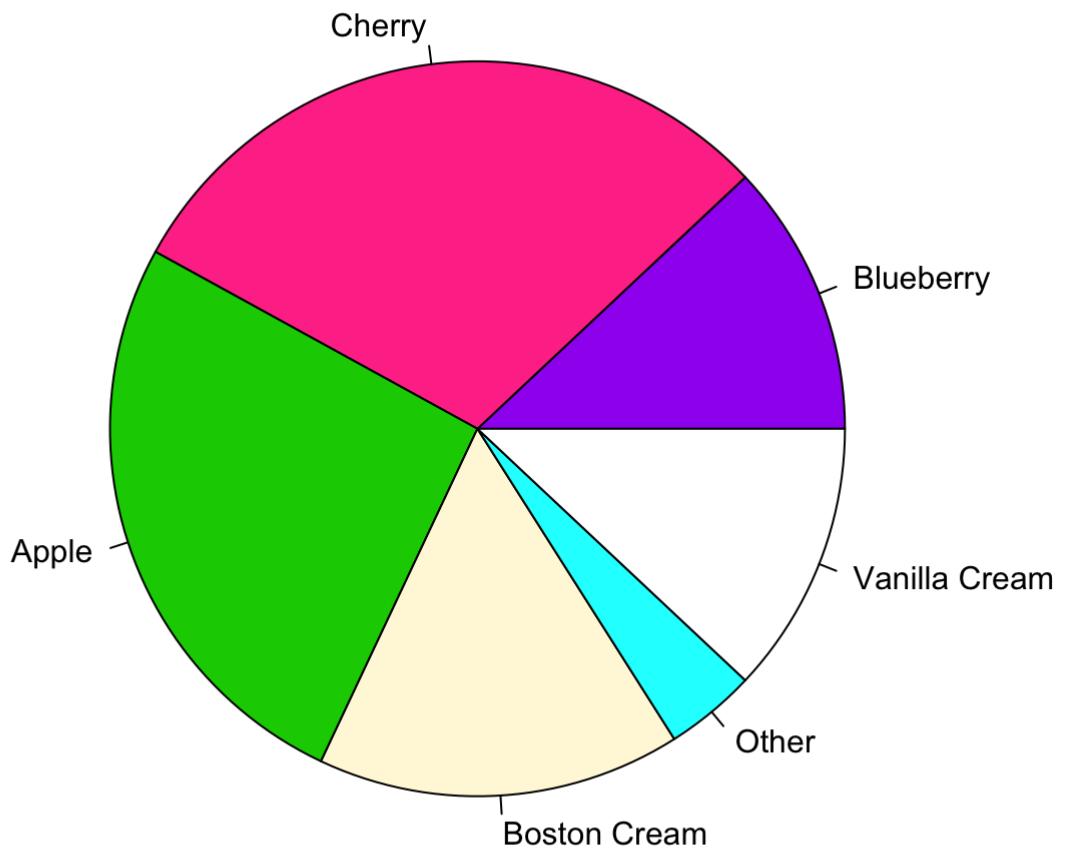


```
pie> pie.sales <- c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)

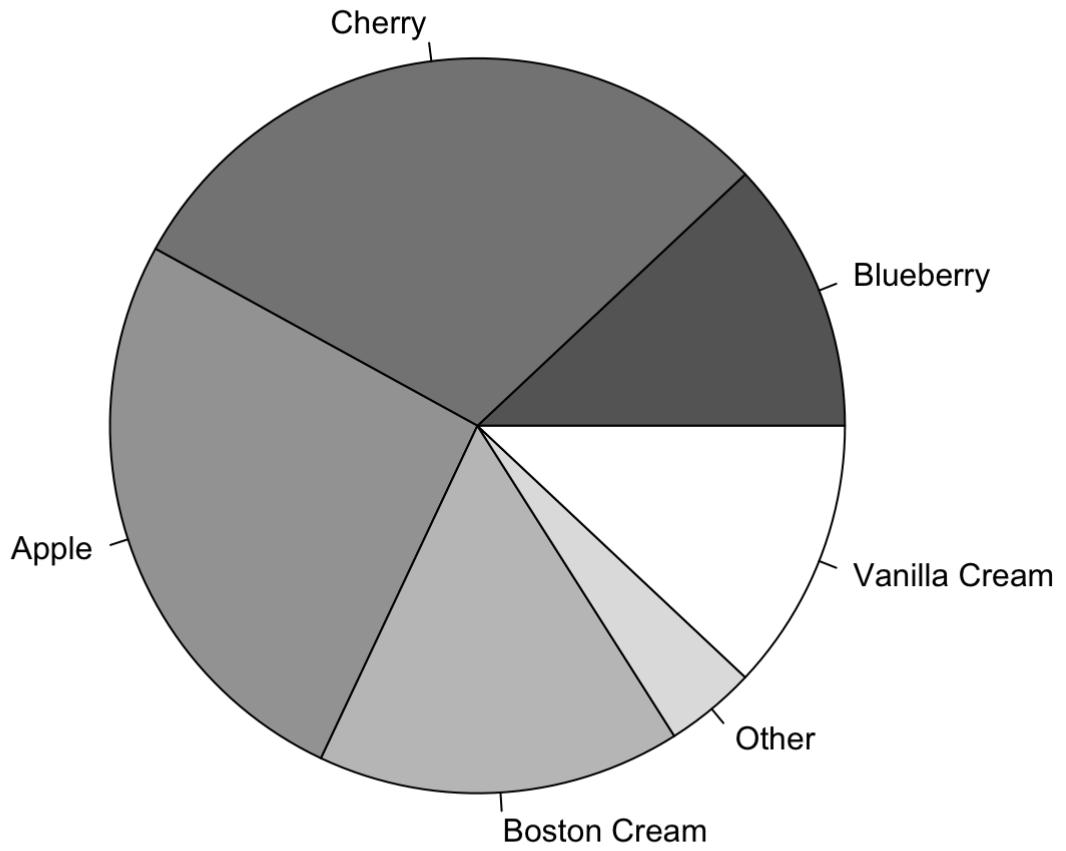
pie> names(pie.sales) <- c("Blueberry", "Cherry",
pie+     "Apple", "Boston Cream", "Other", "Vanilla Cream")

pie> pie(pie.sales) # default colours
```

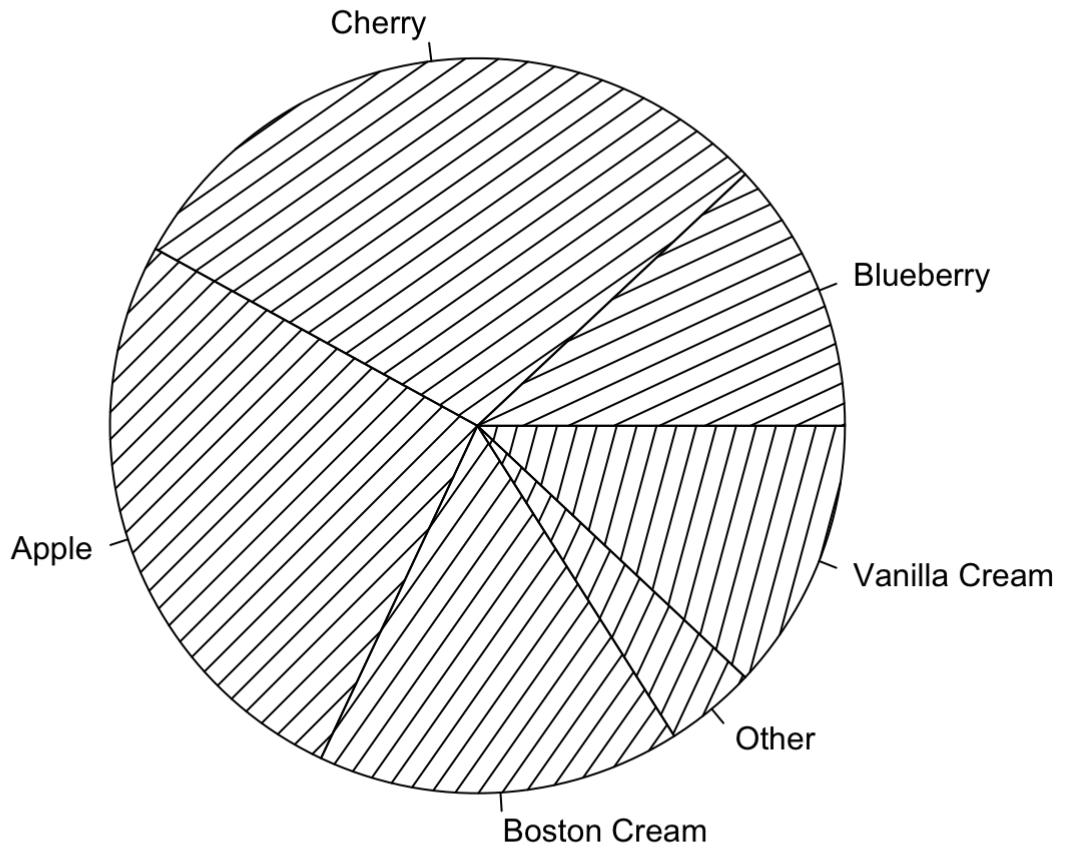




```
pie> pie(pie.sales, col = gray(seq(0.4, 1.0, length.out = 6)))
```

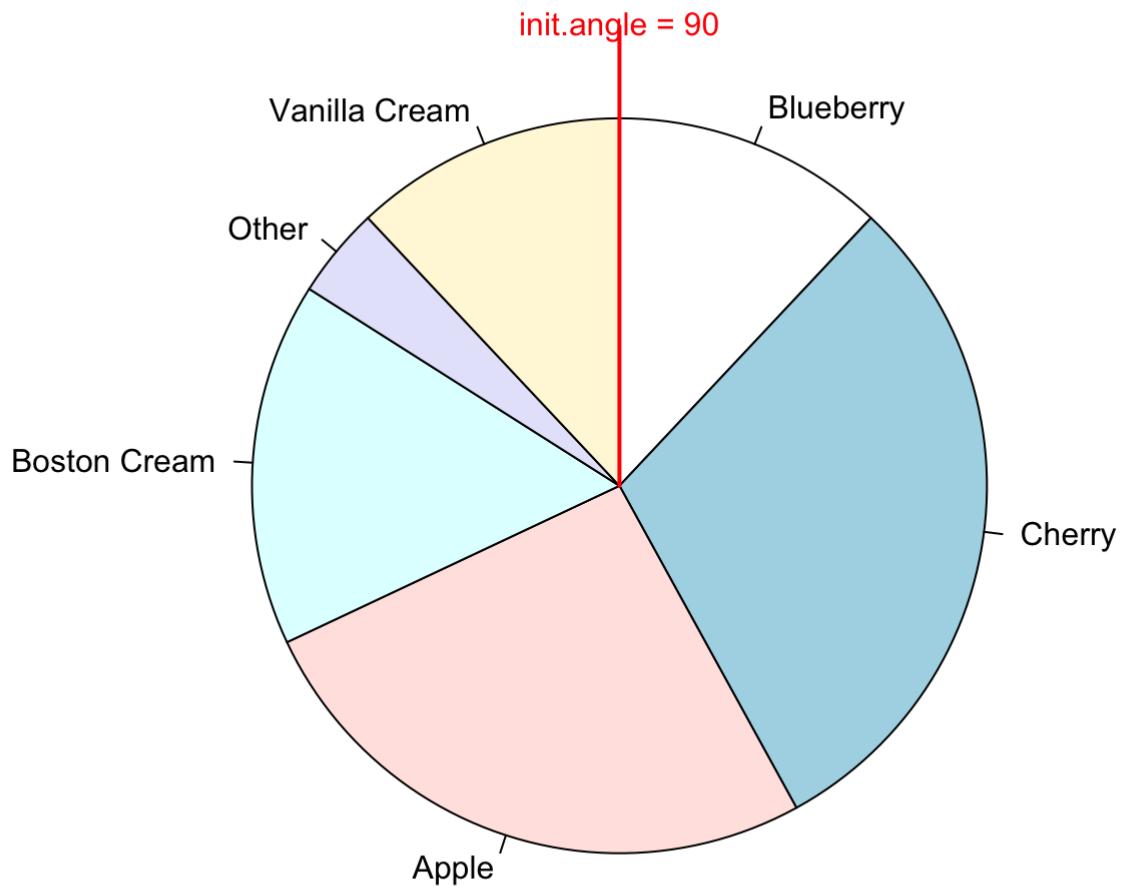


```
pie> pie(pie.sales, density = 10, angle = 15 + 10 * 1:6)
```



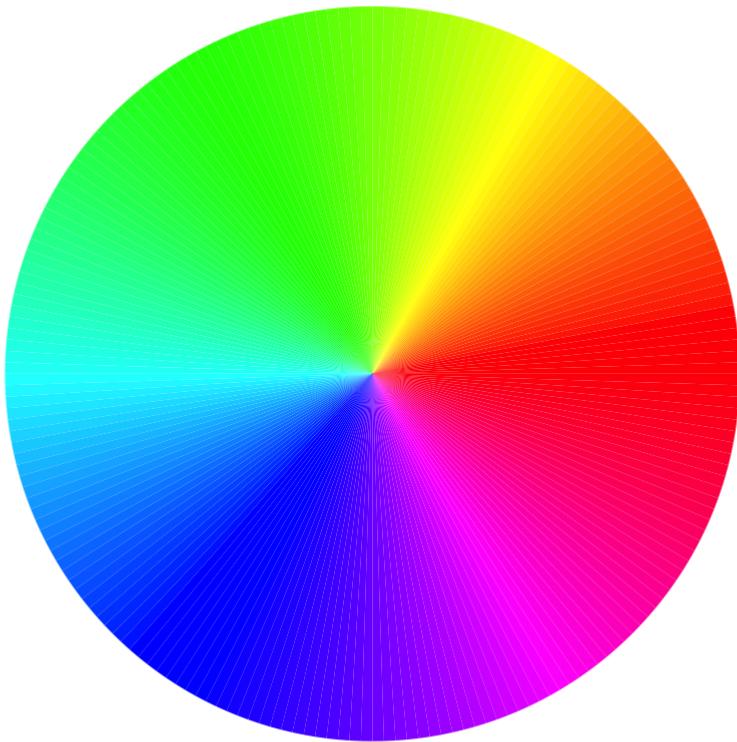
```
pie> pie(pie.sales, clockwise = TRUE, main = "pie(*, clockwise = TRUE)")
```

pie(*, clockwise = TRUE)

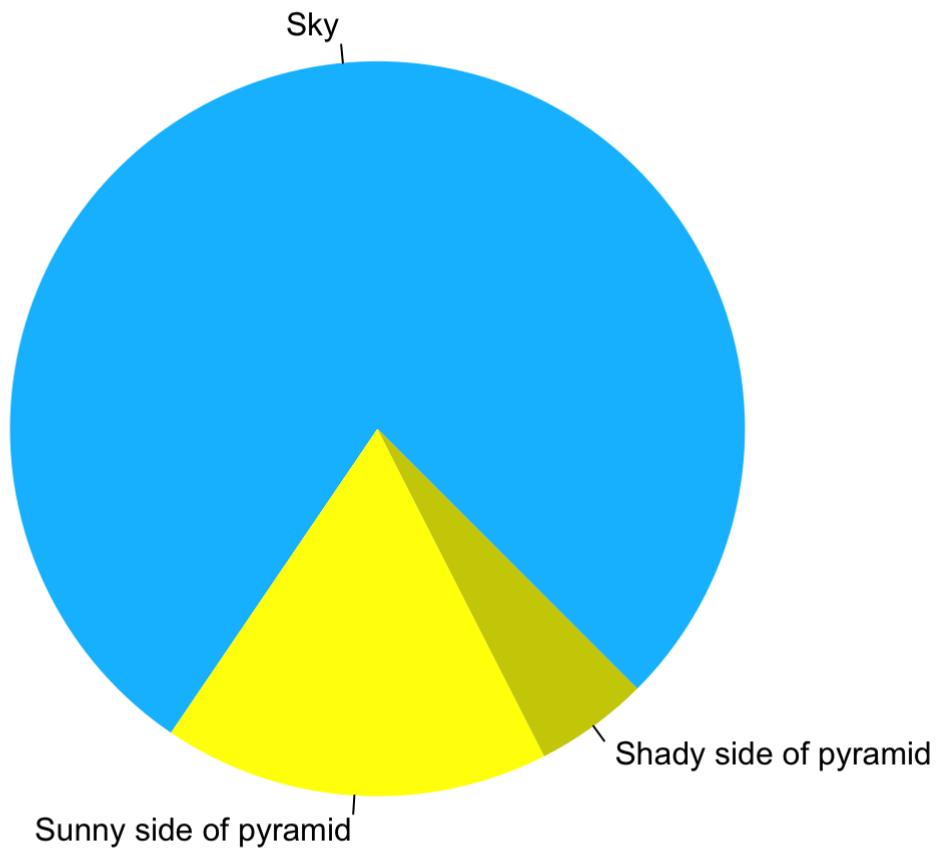


```
pie> segments(0, 0, 0, 1, col = "red", lwd = 2)
pie> text(0, 1, "init.angle = 90", col = "red")
pie> n <- 200
pie> pie(rep(1, n), labels = "", col = rainbow(n), border = NA,
pie+     main = "pie(*, labels=\\\"\\\", col=rainbow(n), border=NA, . .")
```

```
pie(*, labels="", col=rainbow(n), border=NA,..
```

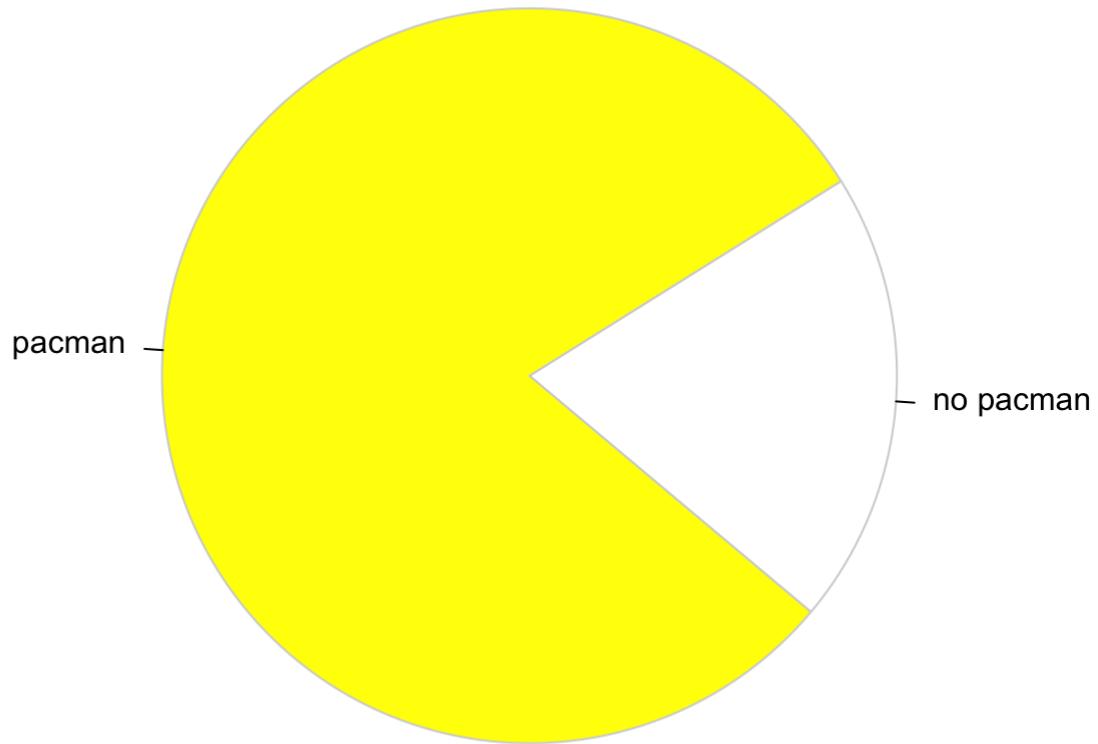


```
pie> ## Another case showing pie() is rather fun than science:  
pie> ## (original by FinalBackwardsGlance on http://imgur.com/gallery/wWrpu4  
X)  
pie> pie(c(Sky = 78, "Sunny side of pyramid" = 17, "Shady side of pyramid" =  
5),  
pie+     init.angle = 315, col = c("deepskyblue", "yellow", "yellow3"), borde  
r = FALSE)
```



[Hide](#)

```
pie(c(20, 80), init.angle=-40,  
    col=c("white", "yellow"),  
    label=c("no pacman", "pacman"),  
    border = "lightgrey")
```



... or switch from pie to waffle (seriously)

6.12 How to do 3D exploded pie charts

DON'T. And this time I mean it

sadly enough there would be packages for this, too

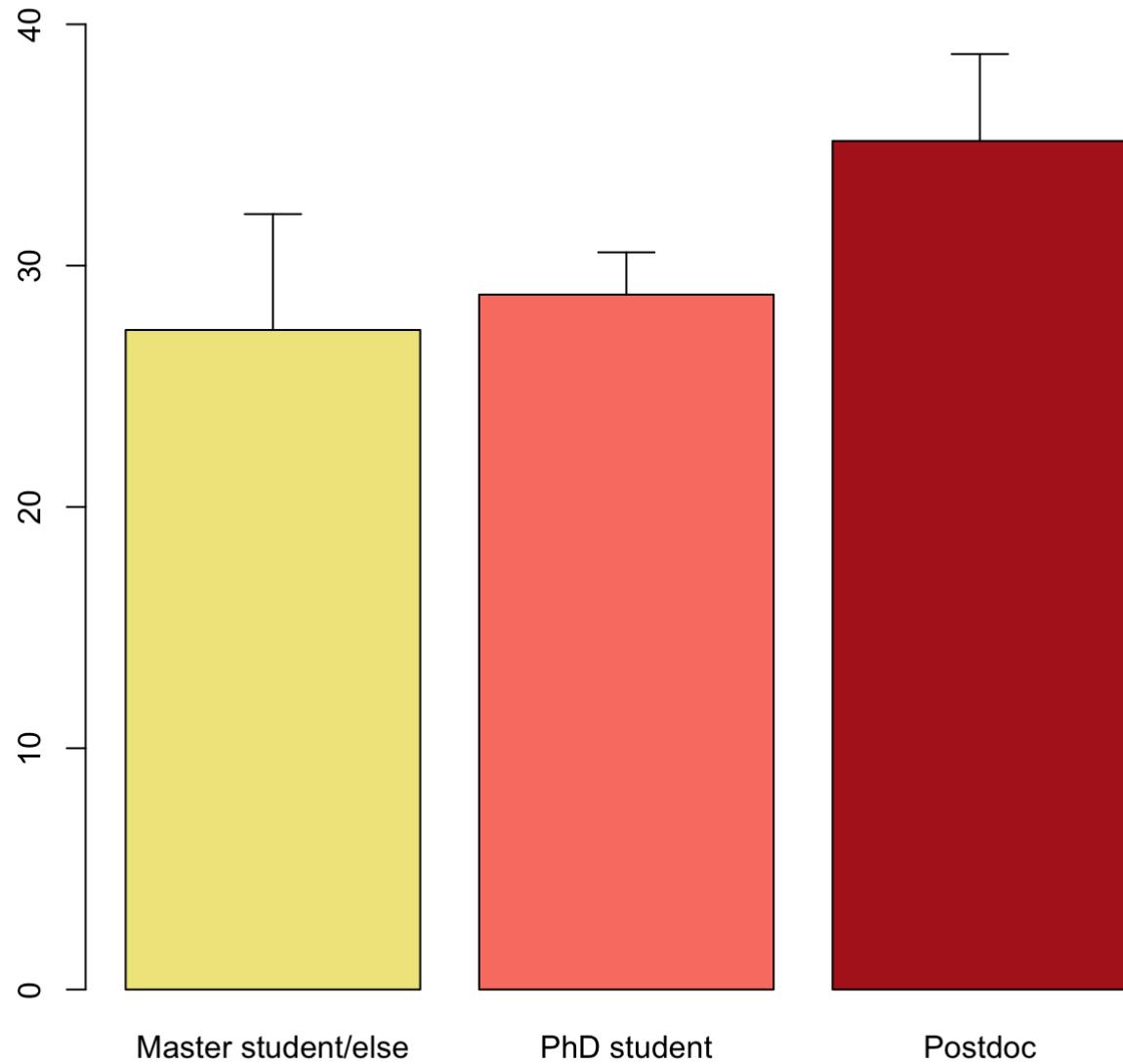
6.13 Extra: dynamite plots

a.k.a. Why is this bad?

```

age_by_group <- tapply(surveyrbioc$Q1,surveyrbioc$Q2,mean)
sd_by_group <- tapply(surveyrbioc$Q1,surveyrbioc$Q2,sd)
mybar <- barplot(age_by_group,col=c("khaki","salmon","firebrick"), ylim=c(0,max(age_by_group) + 5))
# mybar, inspect it
arrows(mybar, age_by_group,mybar, (age_by_group + sd_by_group), length = 0.15,angle= 90)

```



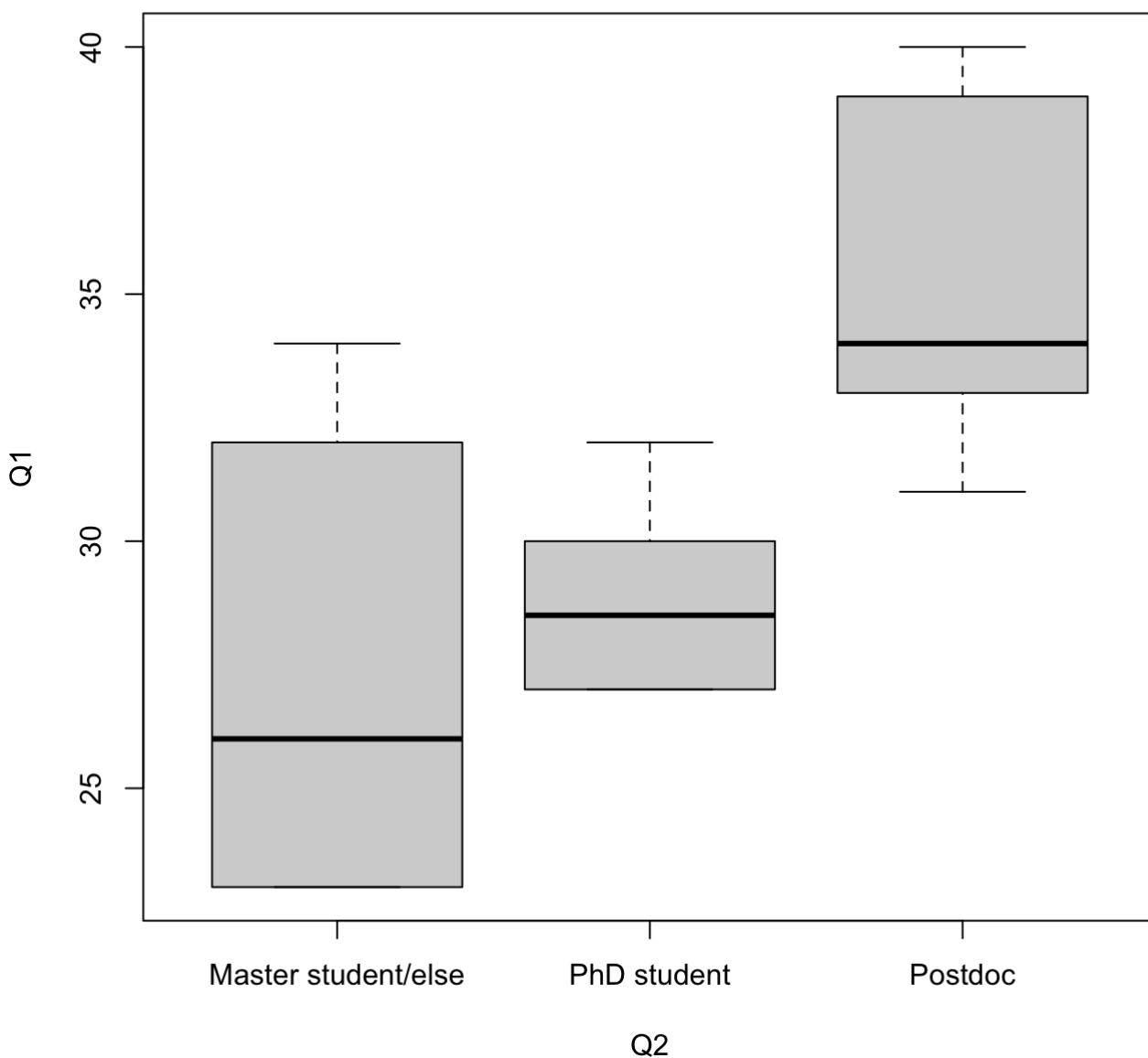
Dynamite plots VS boxplots

[Hide](#)

```

boxplot(Q1~Q2,
        data = surveyrbioc)

```



Median VS distribution VS actual points... What do you really want to show?

6.14 What can you do more with your plot?

- change the points type - see `type` in `?plot`
- use log scales - see `log`
- annotate (some of) the points - with `text`
- change font sizes, styles and so on
- use special characters with `expression`
- save the plot
 - use the point-and-click interface in RStudio
 - code it

6.14.1 Saving your plots

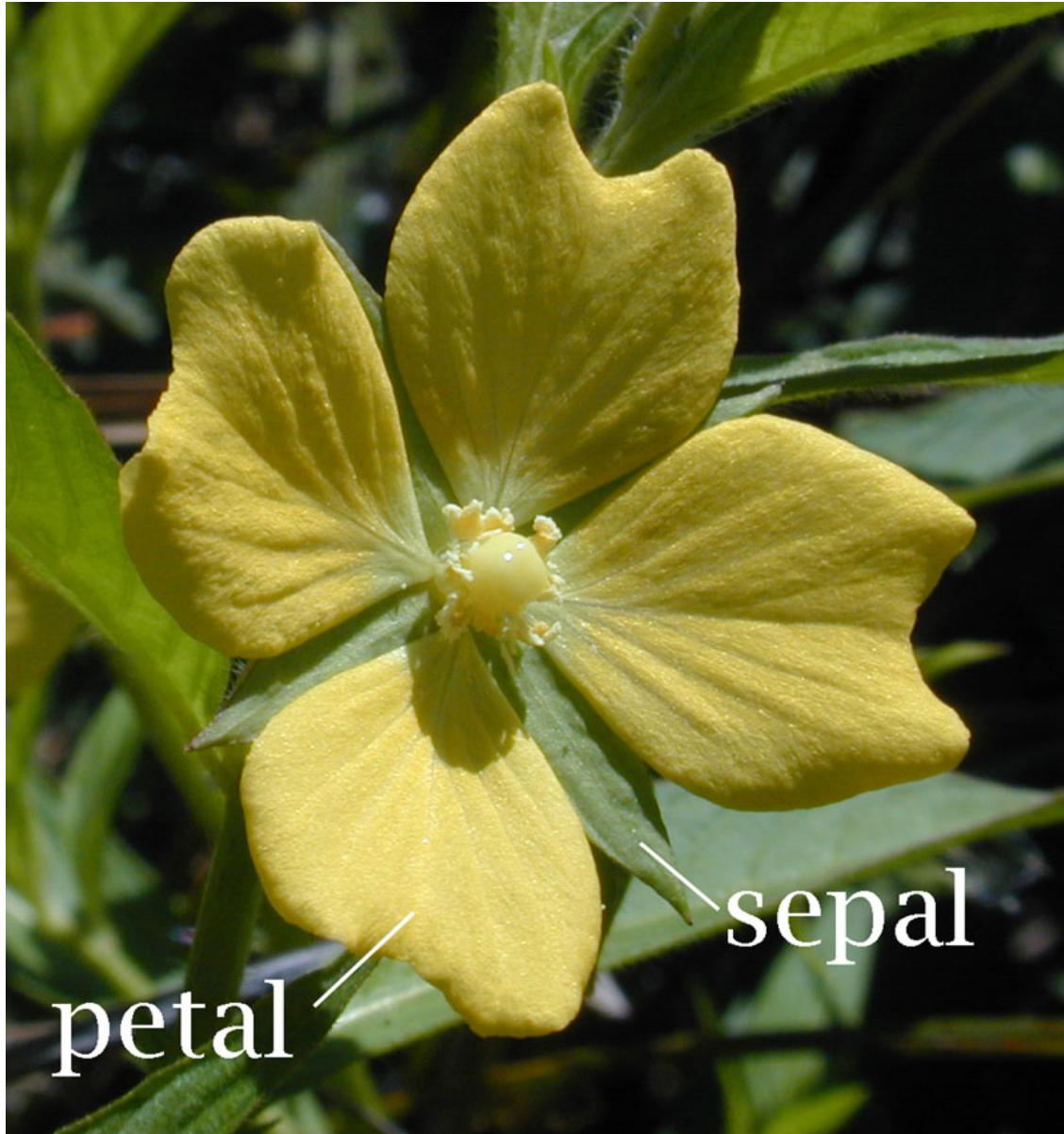
General code structure for this

```
opendevice()  
...  
code for the plot  
...  
closedevice()
```

[Hide](#)

```
pdf("myfilename.pdf")  
# see also alternatives:  
## png()  
## jpeg()  
plot(mpg$displ,mpg$cty,  
     col = mpg$mygroup)  
dev.off()
```

6.15 Petals and sepals



6.16 Exercise session 5

Back to the `iris`. Three species are there. Explore the dataset in the following ways:

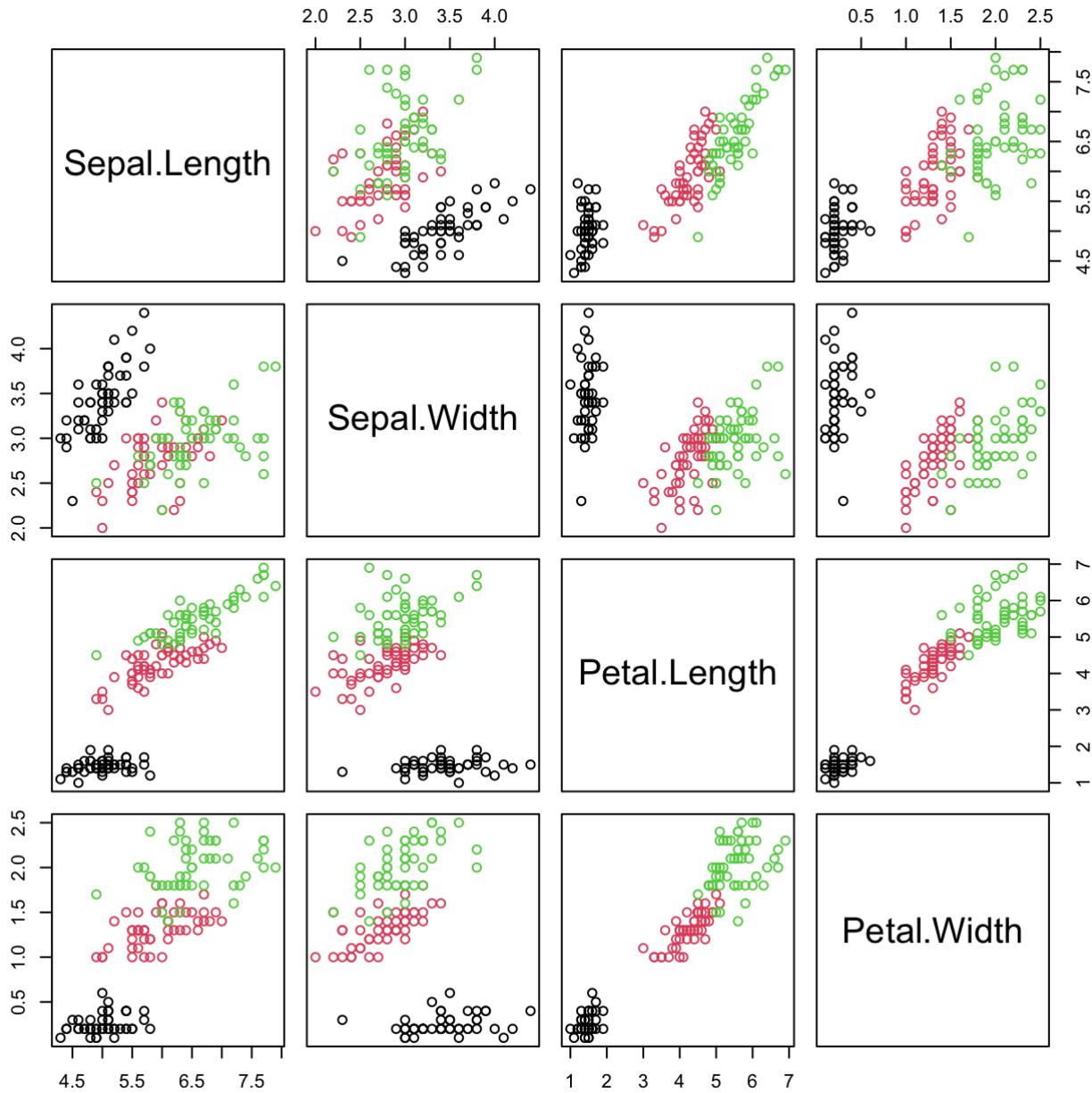
- draw a histogram of the petal length. What do you see?
- plot sepal length versus petal length. Add different colors to highlight the species
- do the same for sepal width and sepal length, and this time use a different symbol for the species. Add a legend and a title if you want
- (harder) calculate the mean values of each feature for each species, organizing it in a matrix where the rows are the species names. Generate a stacked bar plot with it, and another one where the bars are arranged horizontally
- feel free to go back to the survey data and explore it further!

6.16.1 Exercise Session 5 - Solutions

► Details

6.17 Something cool to have an overview...

```
pairs(iris[,1:4], col=iris$Species)
```



You can use the panels even more cleverly, check the help of `pairs!` !

This is a collection on graphs in R - with the underlying code too.

<http://shiny.stat.ubc.ca/r-graph-catalog/> (<http://shiny.stat.ubc.ca/r-graph-catalog/>)

6.18 The gapminder project

<https://www.youtube.com/watch?v=hVimVzgtD6w> (<https://www.youtube.com/watch?v=hVimVzgtD6w>)

6.19 Meet ggplot2

But first, meet the `gapminder` data

[Hide](#)

```
library(gapminder)
head(gapminder)
# A tibble: 6 × 6
  country     continent   year lifeExp      pop gdpPercap
  <fct>       <fct>     <int>    <dbl>    <int>      <dbl>
1 Afghanistan Asia        1952     28.8  8425333     779.
2 Afghanistan Asia        1957     30.3  9240934     821.
3 Afghanistan Asia        1962     32.0 10267083     853.
4 Afghanistan Asia        1967     34.0 11537966     836.
5 Afghanistan Asia        1972     36.1 13079460     740.
6 Afghanistan Asia        1977     38.4 14880372     786.

head(country_colors)
  Nigeria           Egypt           Ethiopia Congo, Dem. Rep.       South
Africa
  "#7F3B08"        "#833D07"        "#873F07"        "#8B4107"        "#
8F4407"
  Sudan
  "#934607"

head(continent_colors)
  Africa   Americas      Asia    Europe   Oceania
  "#7F3B08" "#A50026" "#40004B" "#276419" "#313695"
```

Variables:

- `country`
- `continent`
- `year`
- `lifeExp`, life expectancy at birth
- `pop`, total population
- `gdpPercap`, per-capita GDP

6.20 The ggplot2 philosophy

`gg` stands for the grammar of graphics

- you provide the `data`
- you map the data to `aes` thetistics (shape, size, colour)
- you add `geom`s to specify how you want to have the data plotted

- you can have statistical transformations
- facets allow you to do quick elegant multi plots

It can come across somewhat harder since

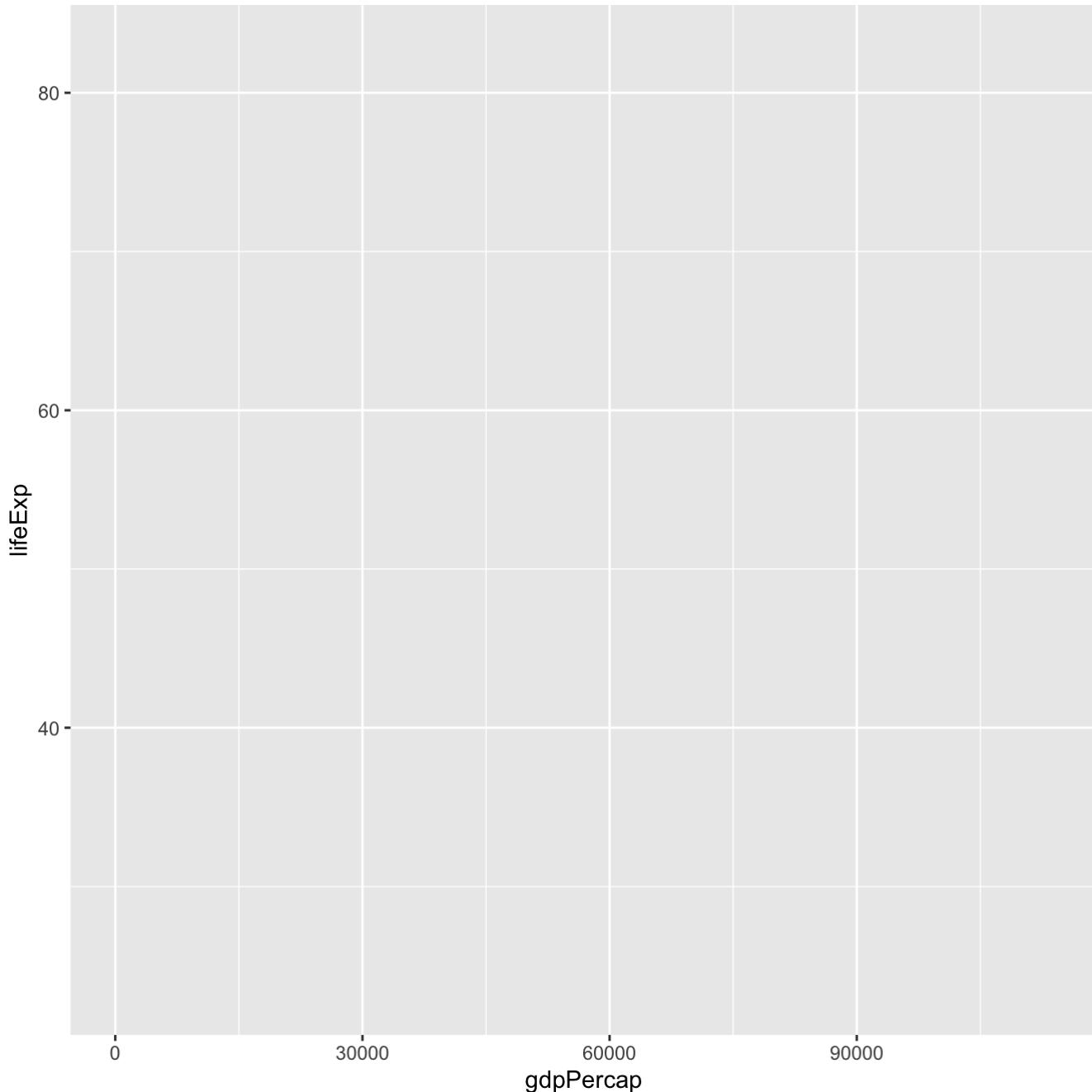
- data need to be tidy - one observation per row
- requires an extra step for abstraction

yet, it makes the whole process of “thinking data” more natural.

6.20.1 A quick dive into the many options

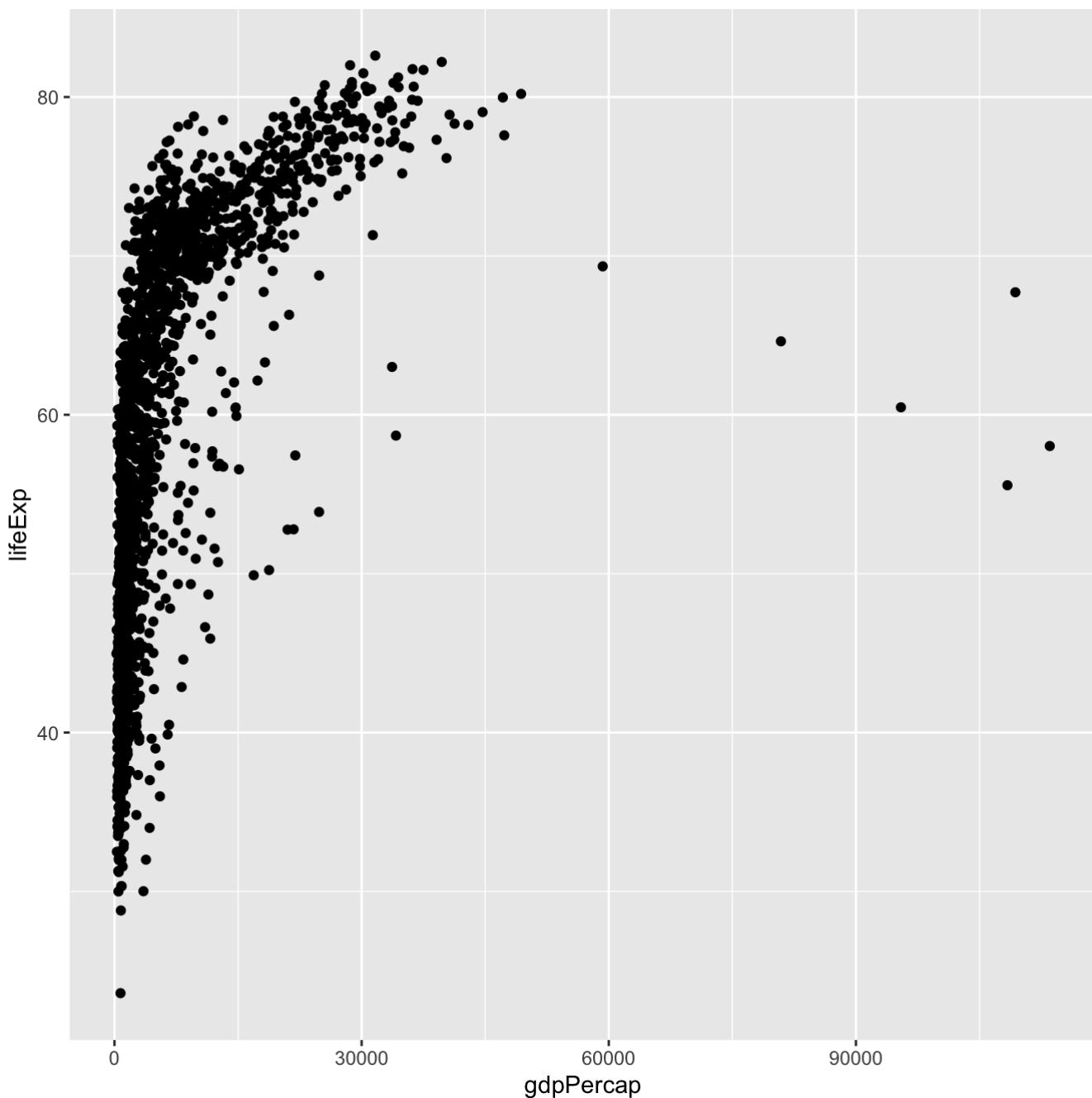
[Hide](#)

```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp))
```



[Hide](#)

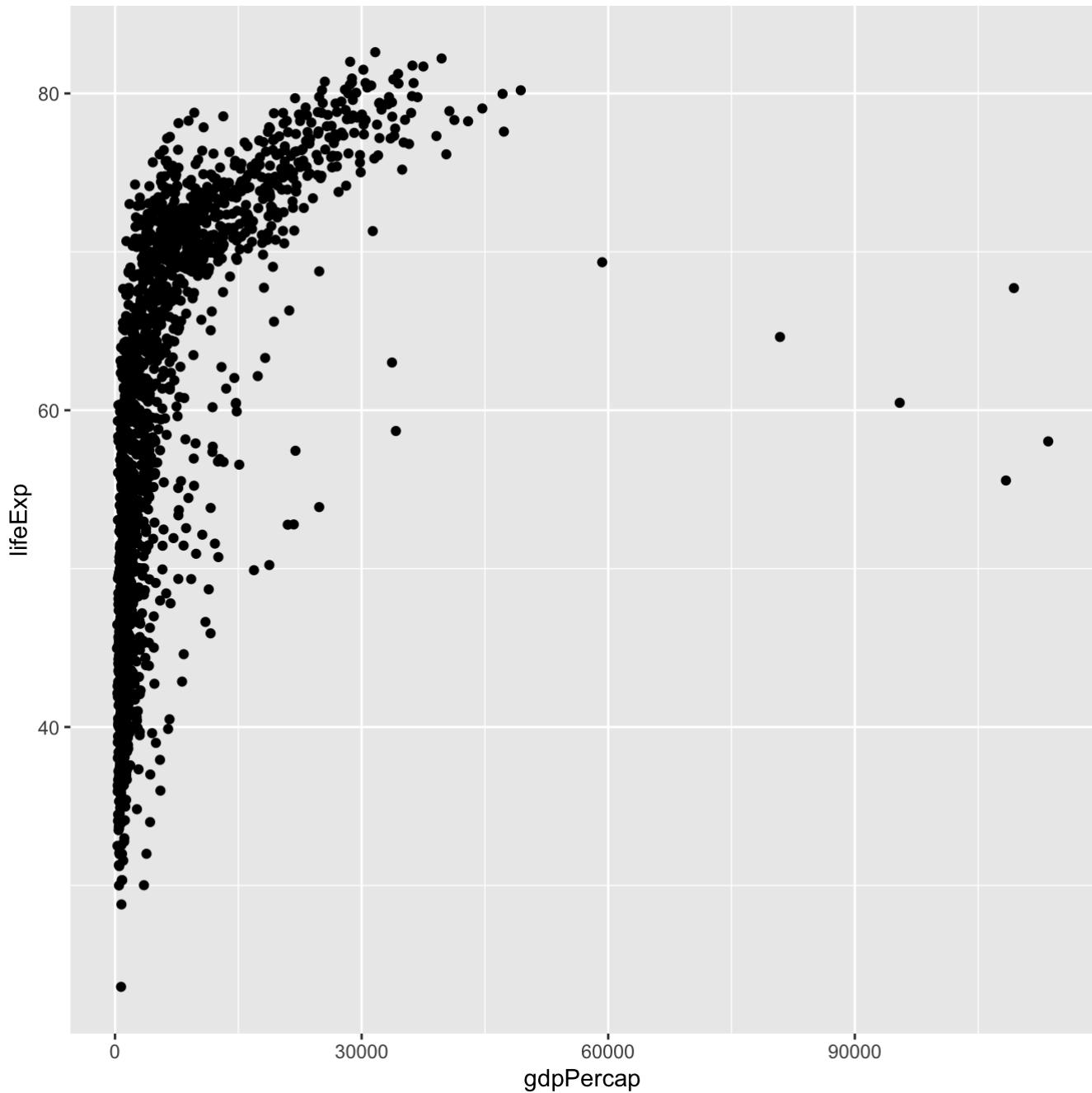
```
ggplot(gapminder,aes(x = gdpPercap, y = lifeExp)) + geom_point()
```



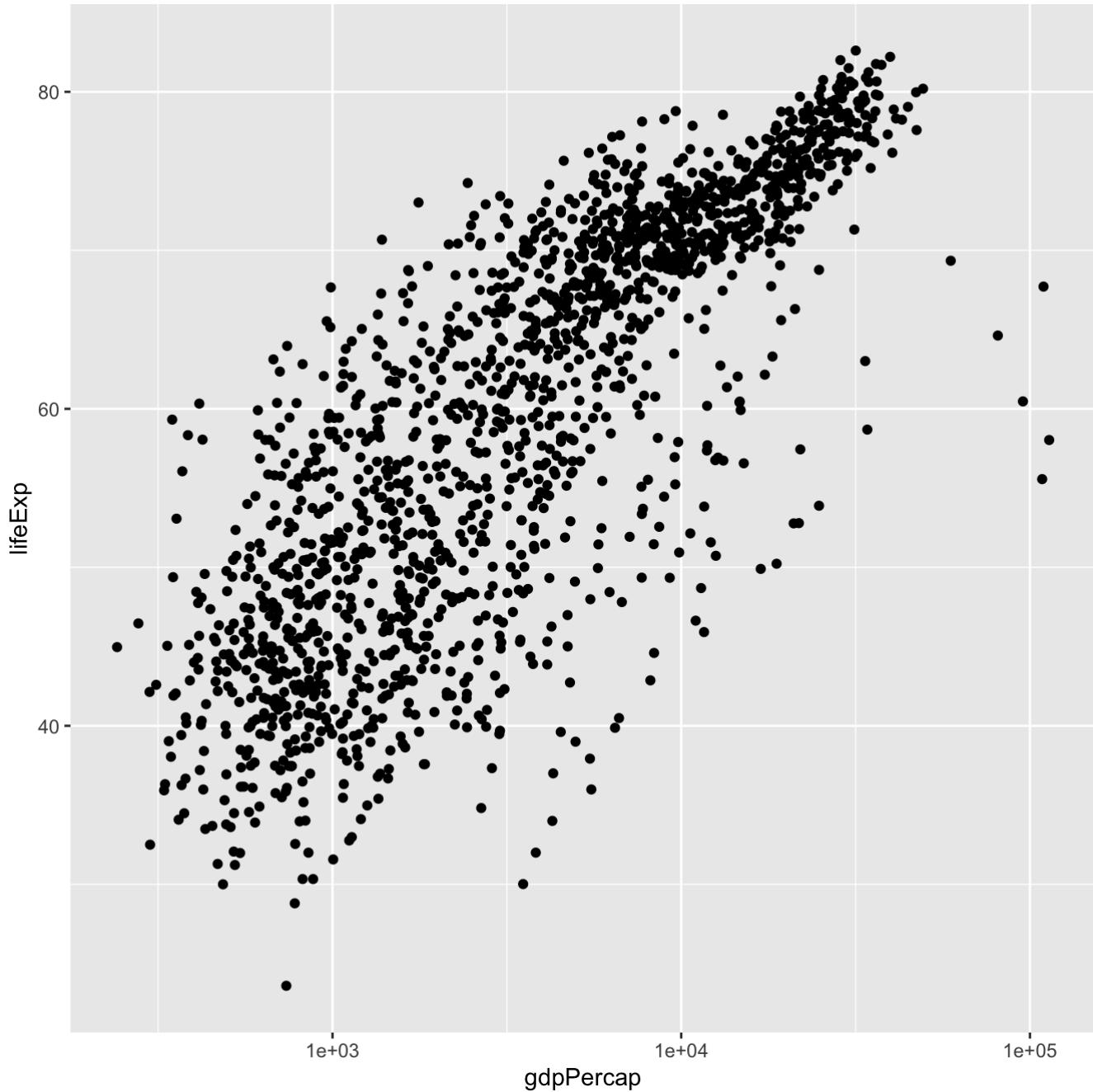
We can store `ggplot` plot objects into a variable - and build upon that later

[Hide](#)

```
p <- ggplot(gapminder,aes(x = gdpPercap, y = lifeExp))  
p + geom_point()
```



```
p + geom_point() + scale_x_log10()
```

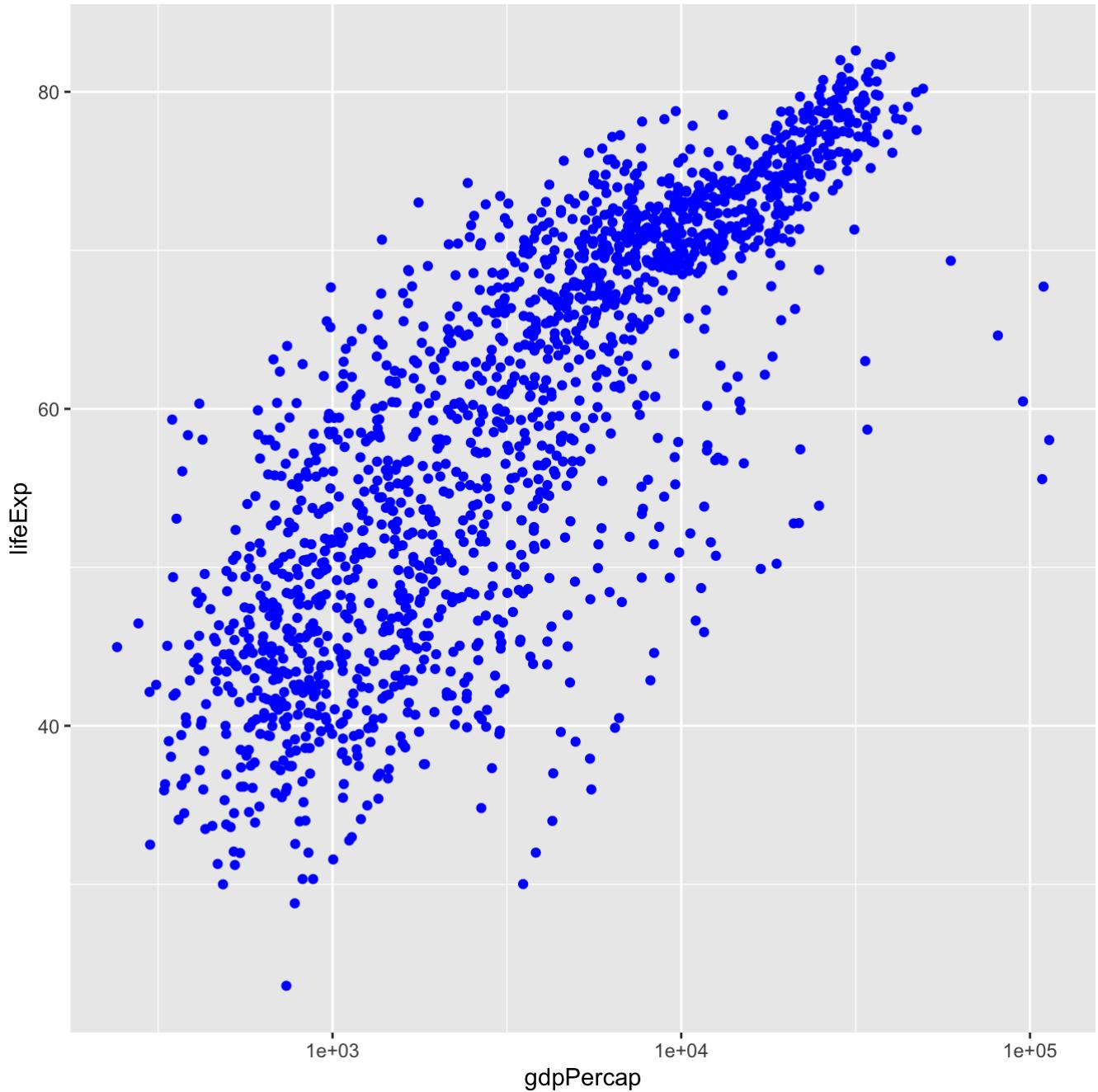


```
p <- p + scale_x_log10()
```

[Hide](#)

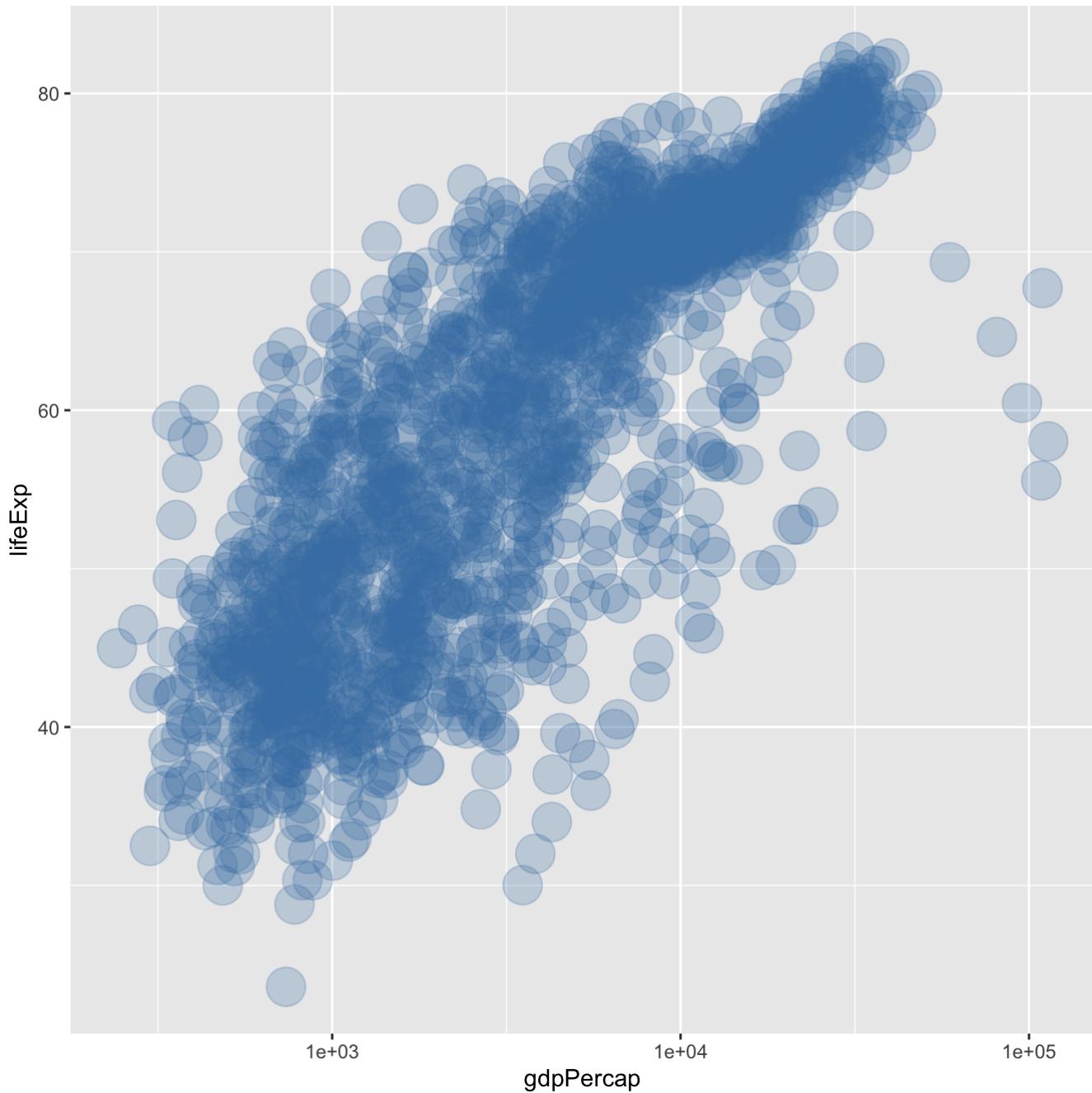
```
p + geom_point(color="blue")
```

[Hide](#)

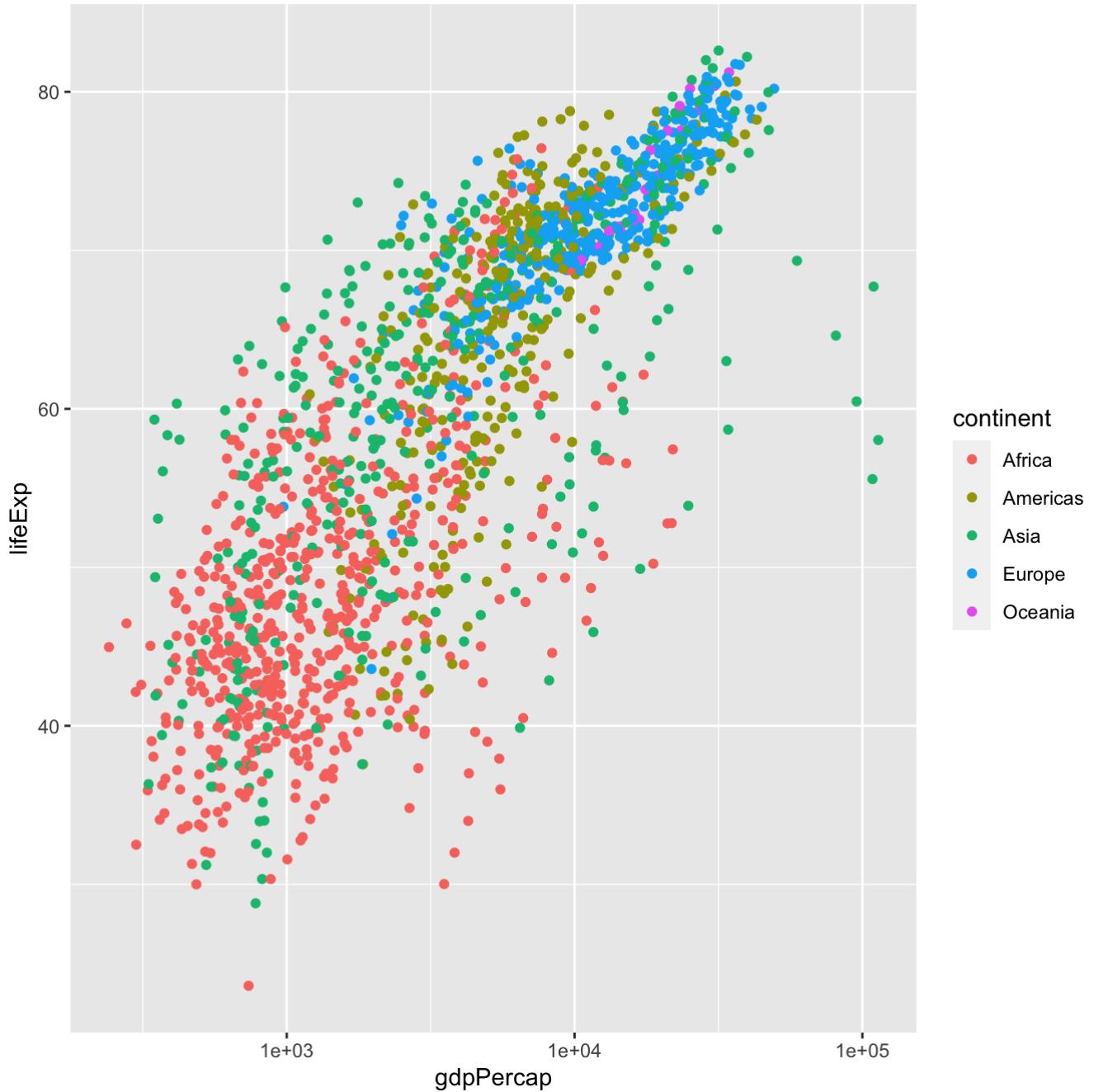


[Hide](#)

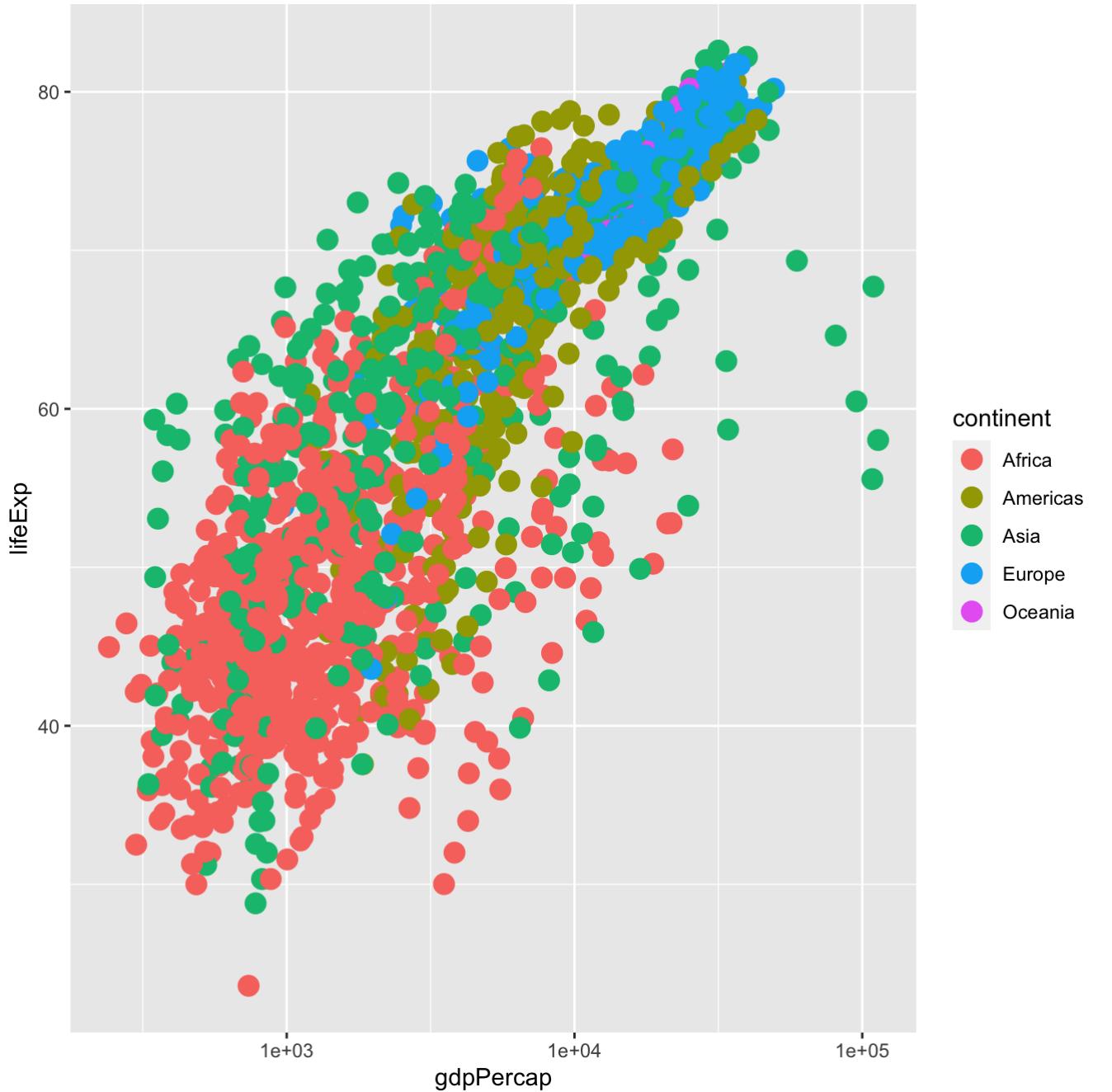
```
p + geom_point(color="steelblue", pch=19, size=8, alpha=1/4)
```



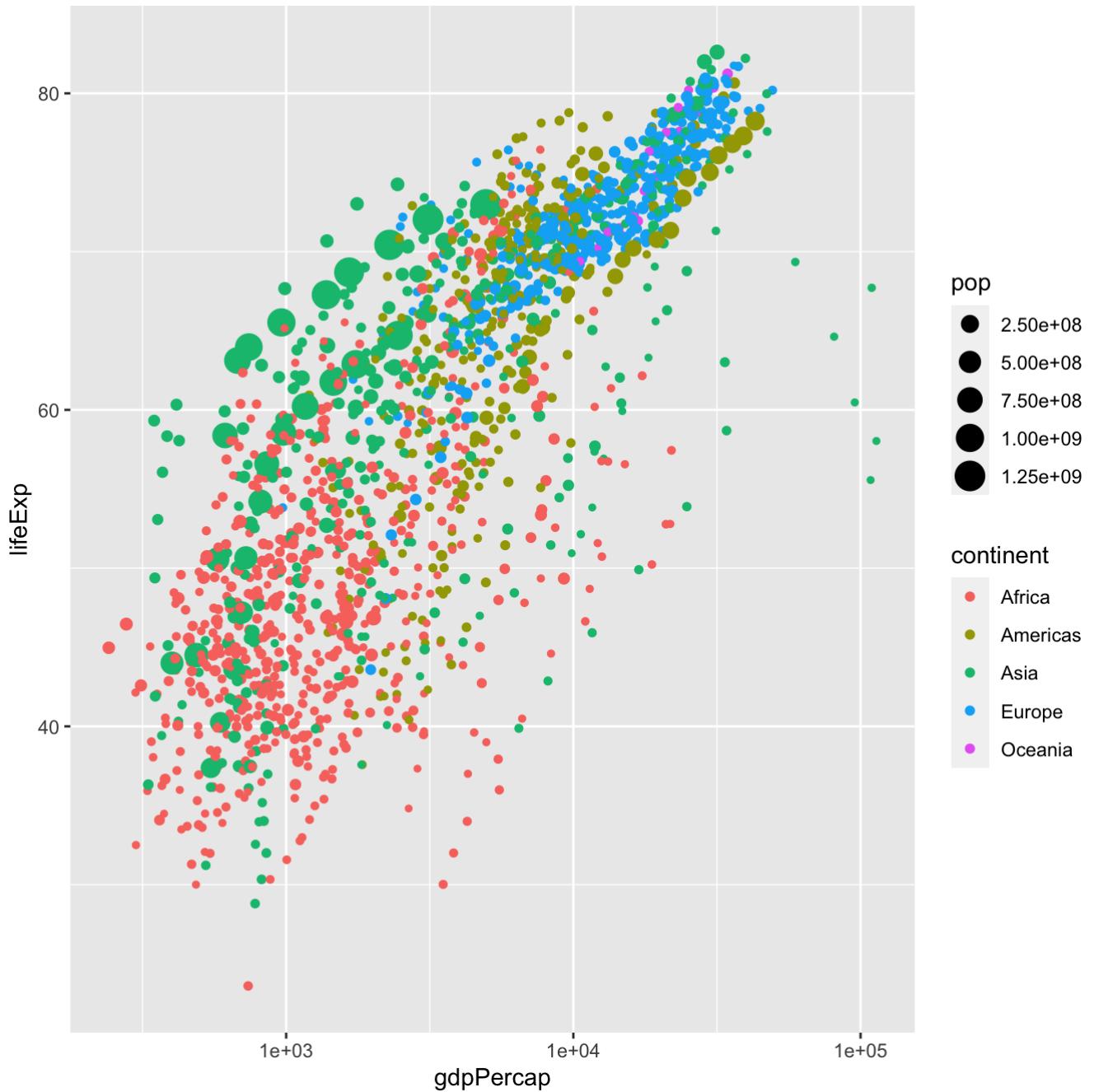
```
p + geom_point(aes(color=continent))
```



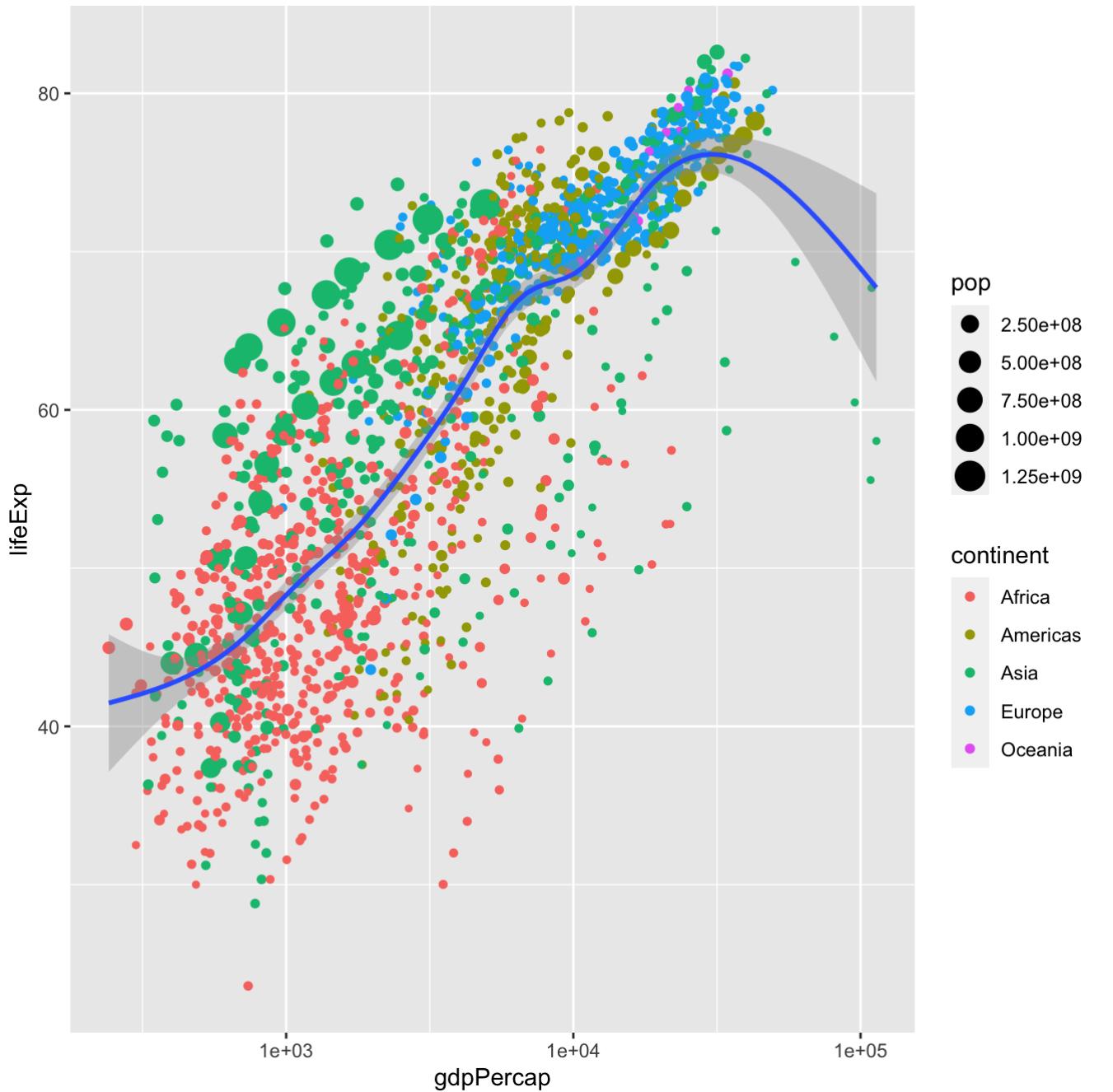
```
p + geom_point(aes(col=continent), size=4)
```



```
p + geom_point(aes(col=continent, size=pop))
```

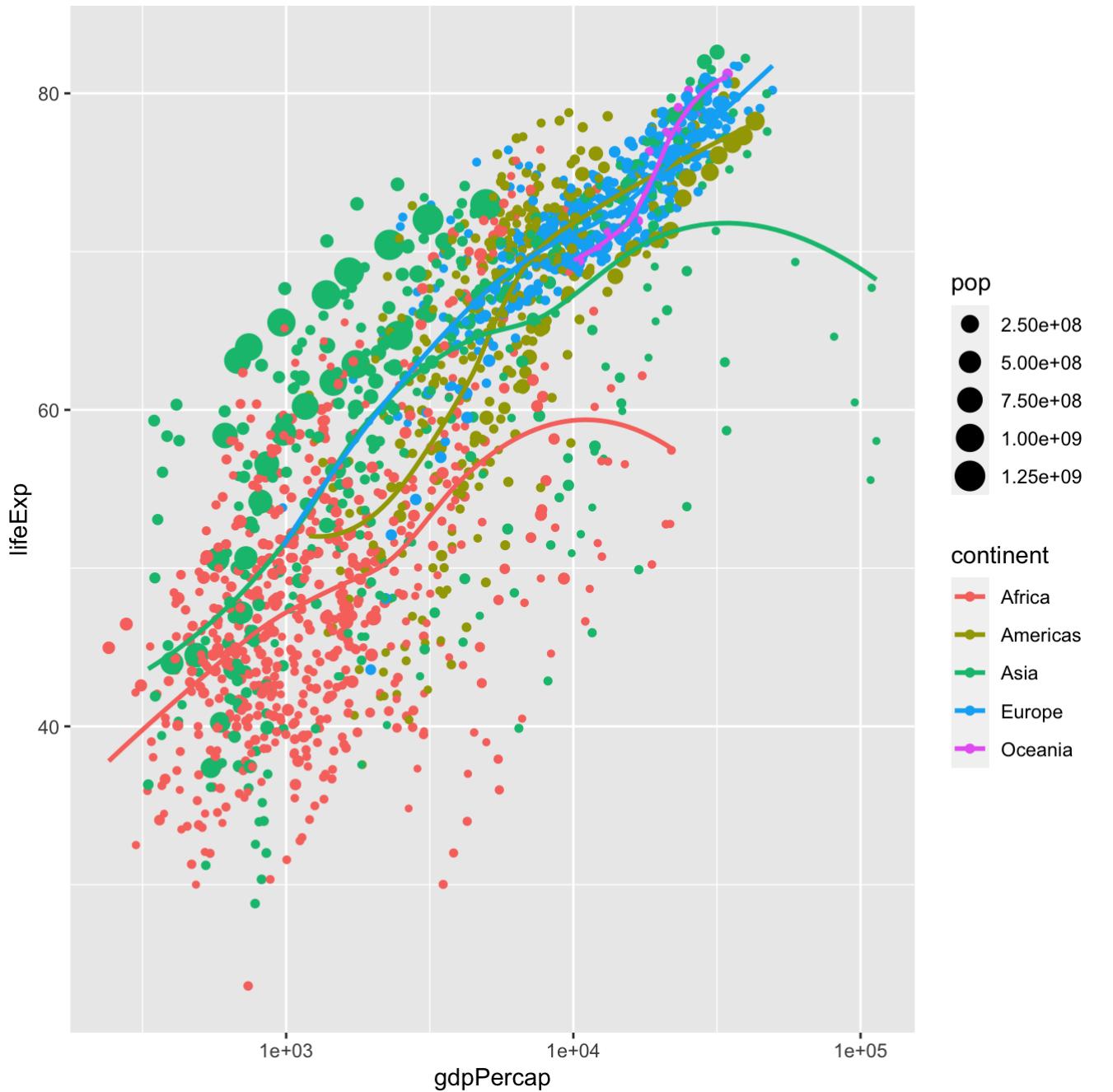


```
p + geom_point(aes(col=continent, size=pop)) + geom_smooth()
```

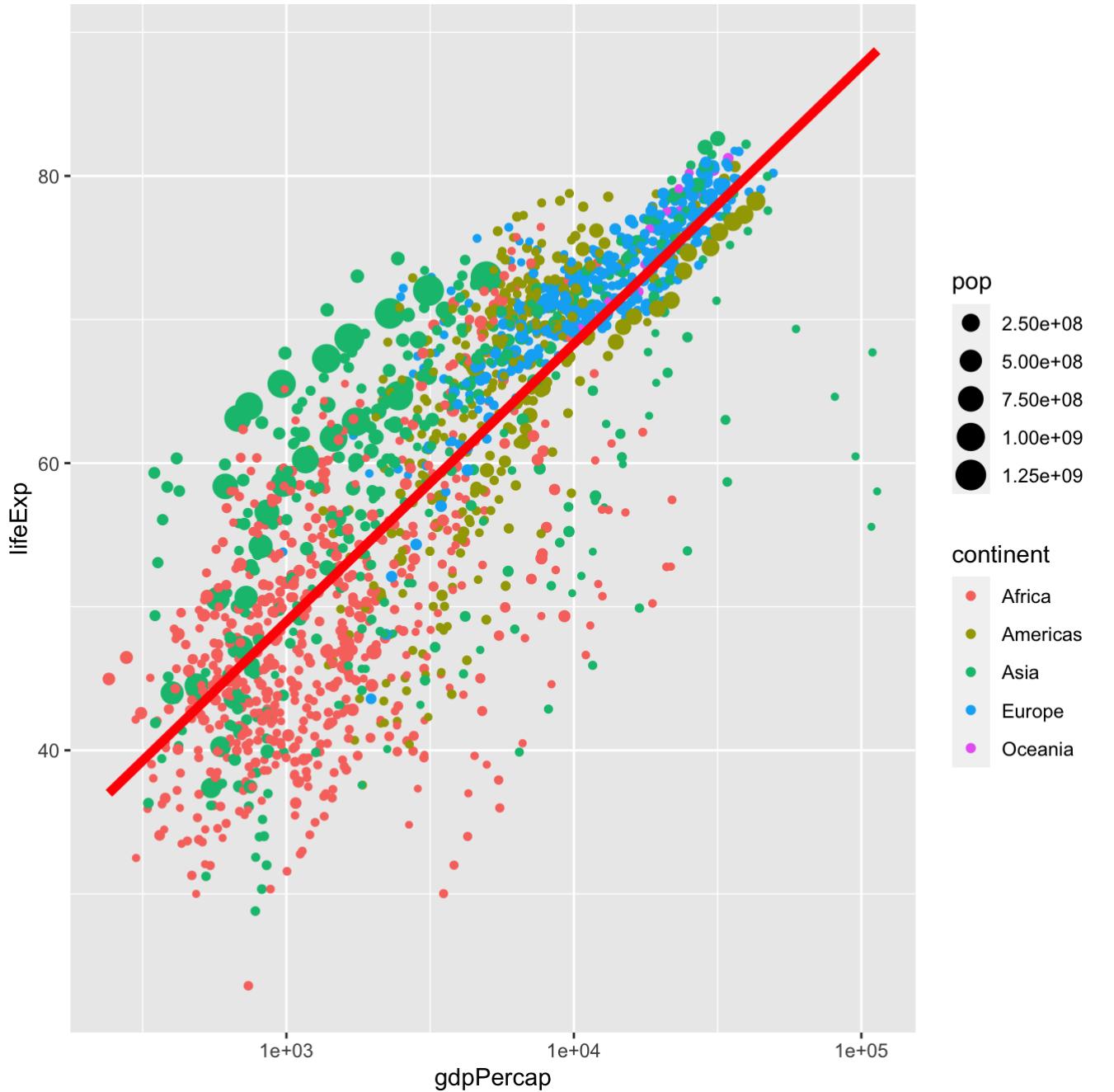


Hide

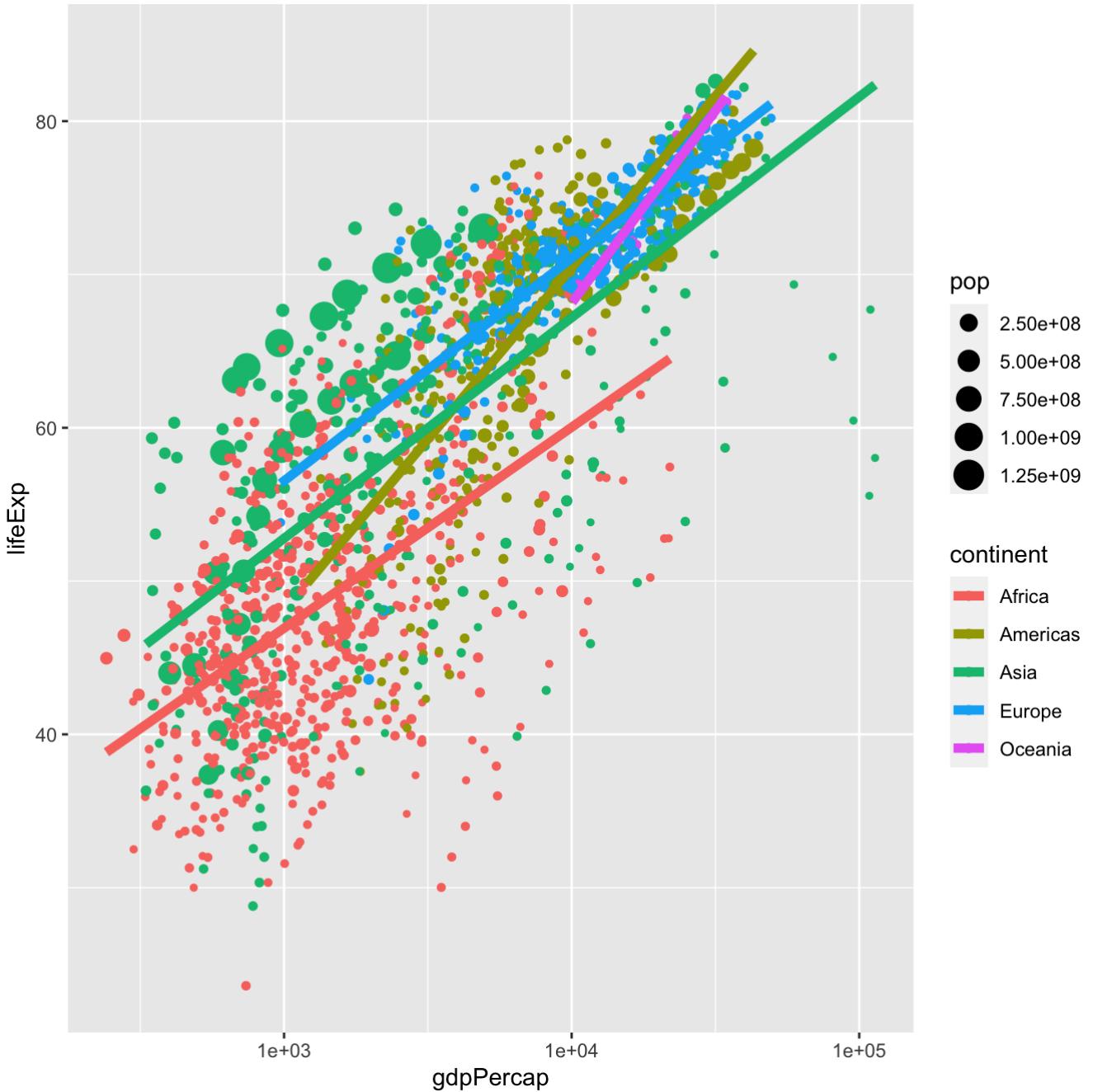
```
niceone <- p + geom_point(aes(col=continent, size=pop)) +
  geom_smooth(aes(col=continent), se=FALSE)
niceone
```



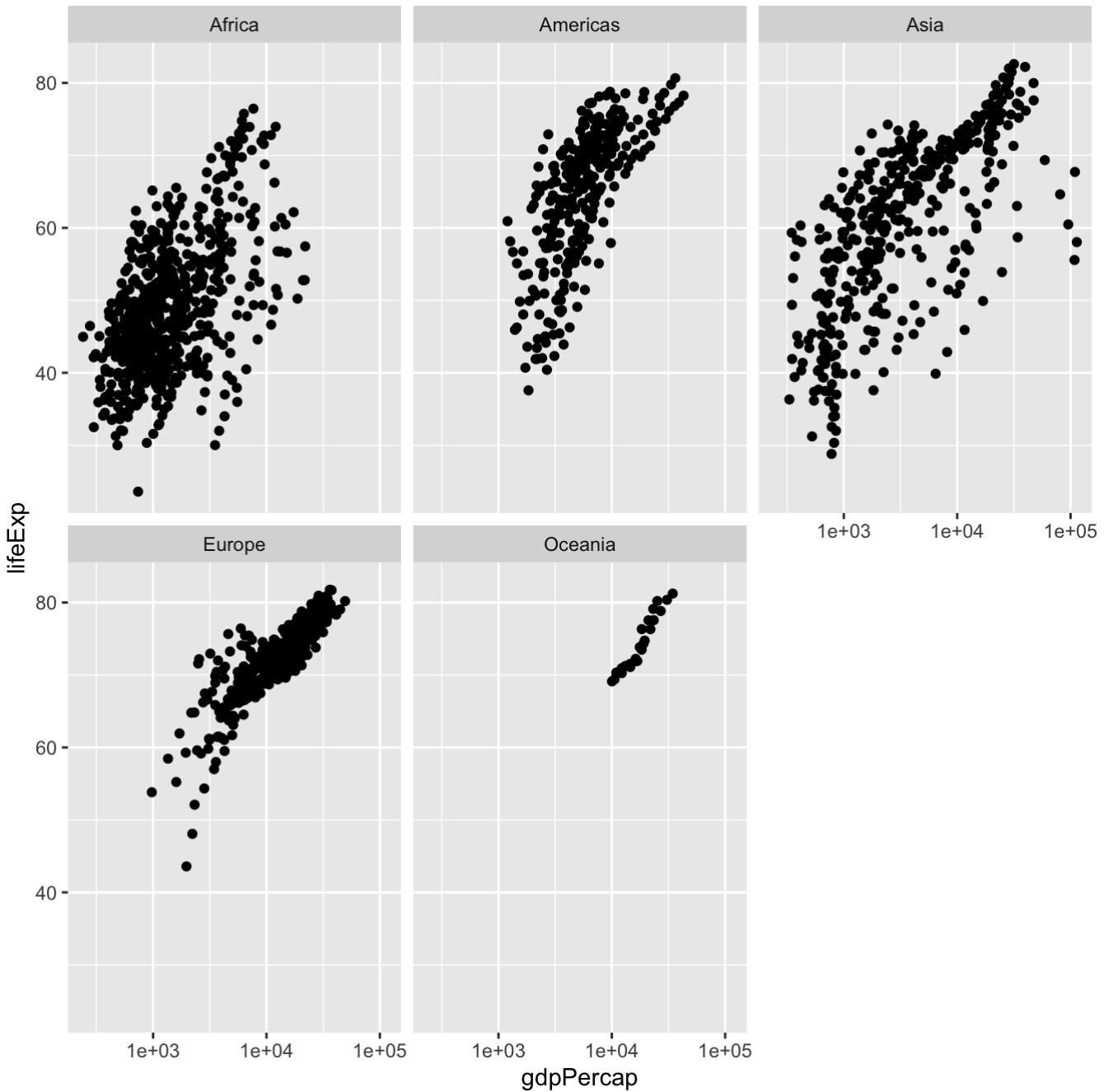
```
p + geom_point(aes(col=continent, size=pop)) +  
  geom_smooth(lwd=2, se=FALSE, method="lm", col="red")
```



```
p + geom_point(aes(col=continent, size=pop)) +  
  geom_smooth(aes(col=continent), lwd=2, se=FALSE, method="lm")
```

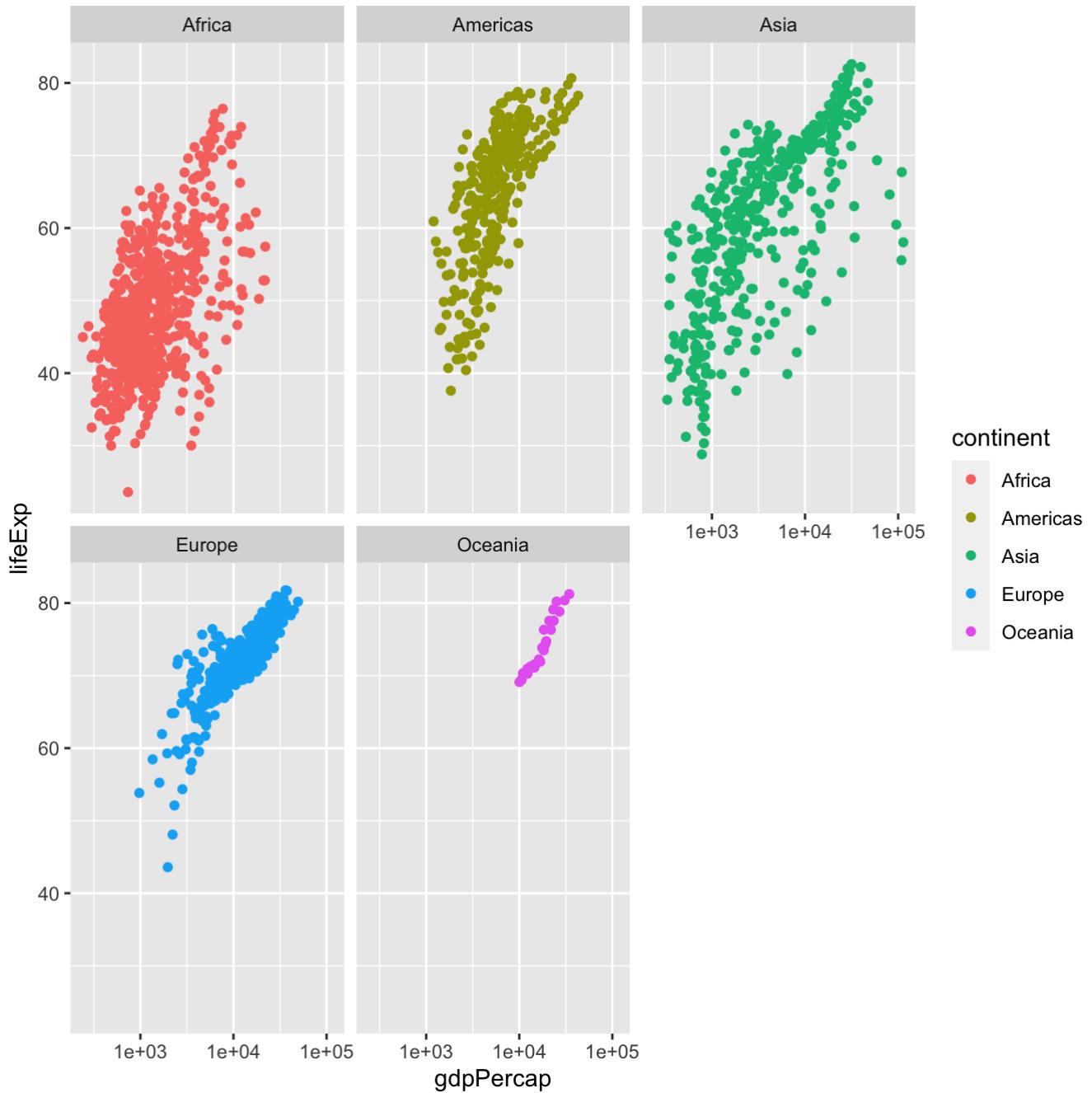


```
p + geom_point() + facet_wrap(~continent)
```

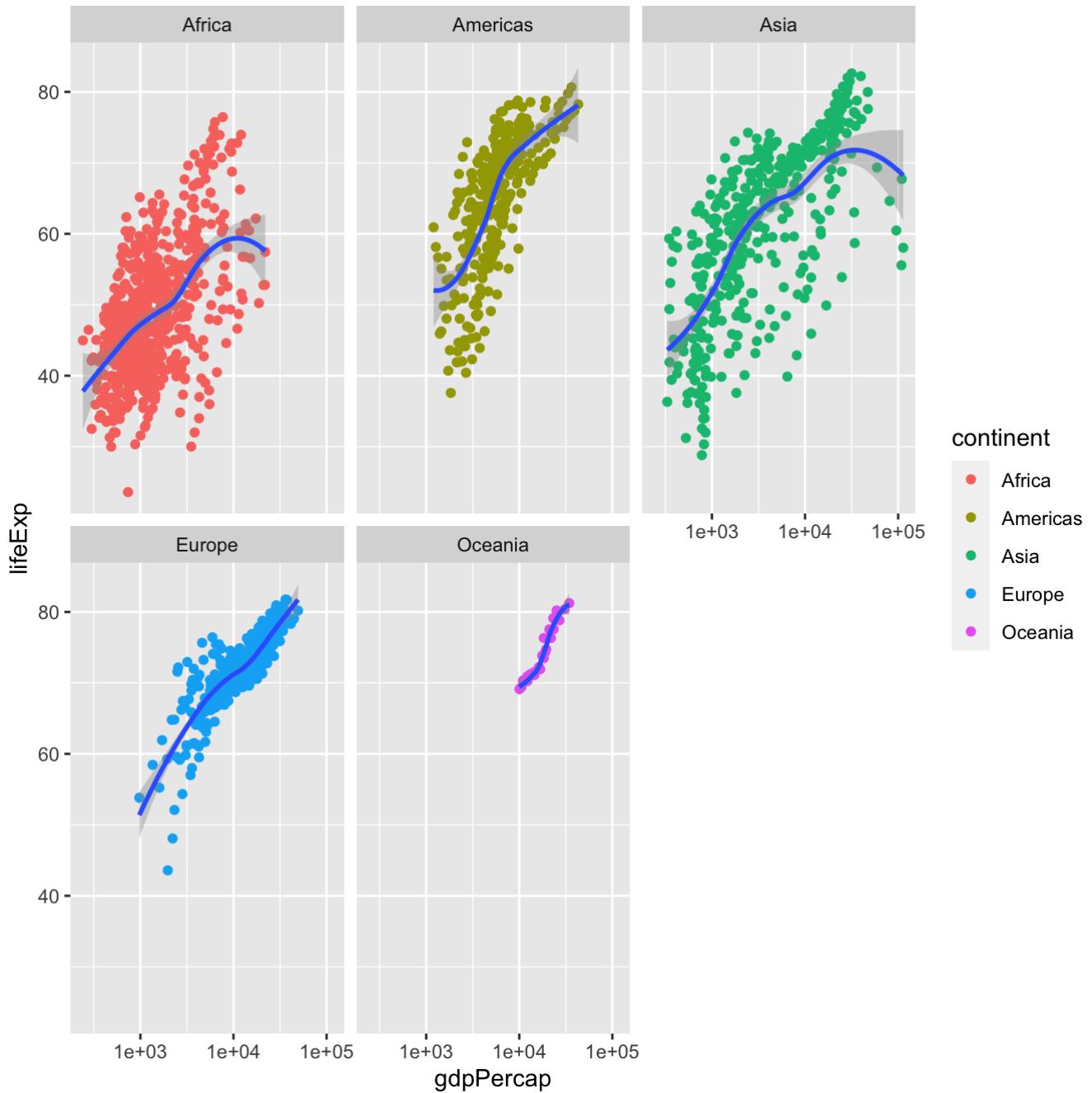


[Hide](#)

```
p + geom_point(aes(col=continent)) + facet_wrap(~continent)
```



```
p + geom_point(aes(col=continent)) + geom_smooth() + facet_wrap(~continent)
```



6.20.2 Saving the plots

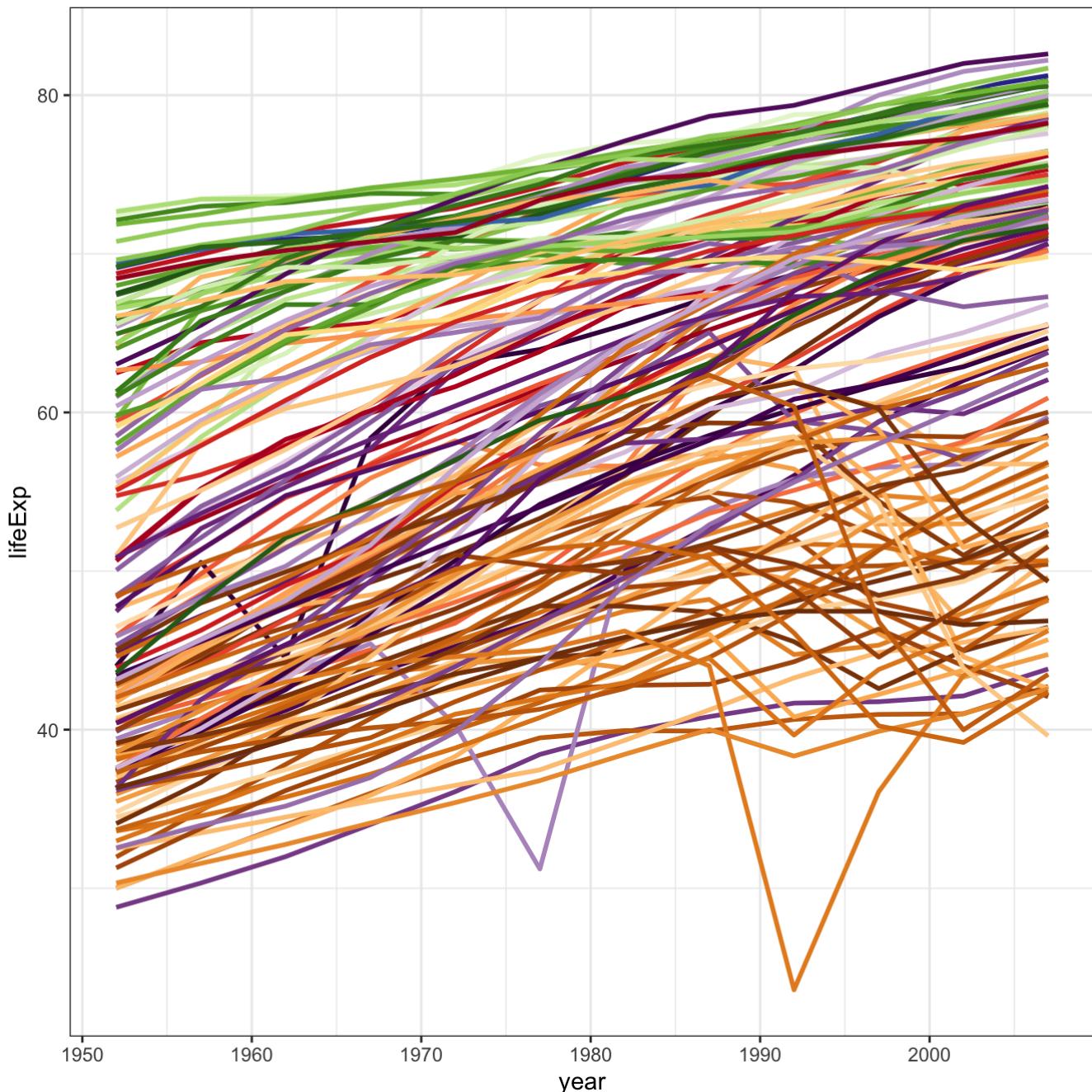
Hide

```
ggsave(file="myplot.png")
```

6.20.3 Line plots

Hide

```
ggplot(gapminder,  
       aes(x = year, y = lifeExp, group = country, color = country))  
  +  
  geom_line(lwd = 1, show.legend = FALSE) +  
  scale_color_manual(values = country_colors) +  
  theme_bw() + theme(strip.text = element_text(size = rel(1.1)))
```

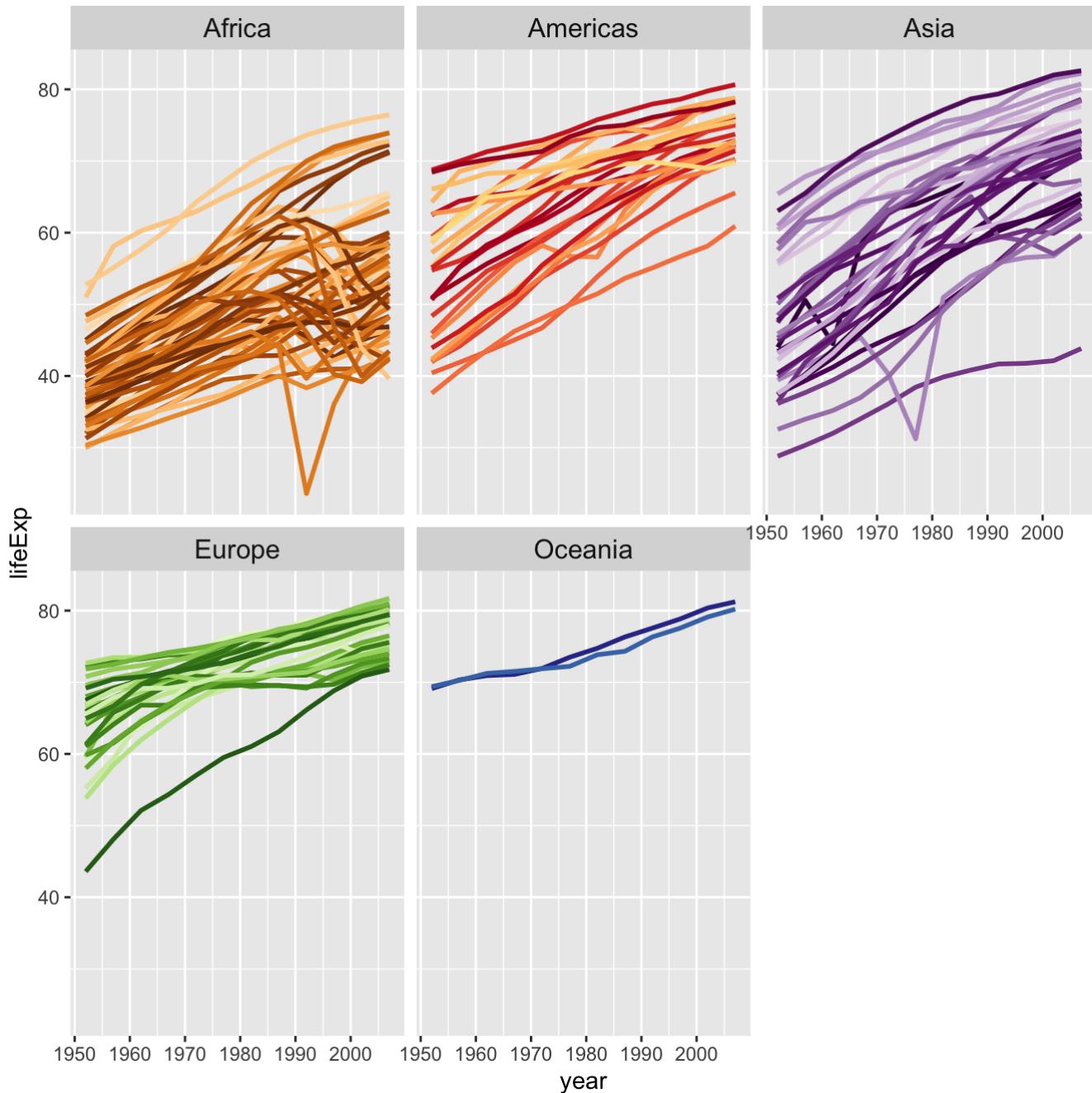


[Hide](#)

```

bp <- ggplot(gapminder,
  aes(x = year, y = lifeExp, group = country, color = country)
) +
  geom_line(lwd = 1, show.legend = FALSE) + facet_wrap(~ continent) +
  scale_color_manual(values = country_colors) + theme(strip.text = element_text(size = rel(1.1)))
bp

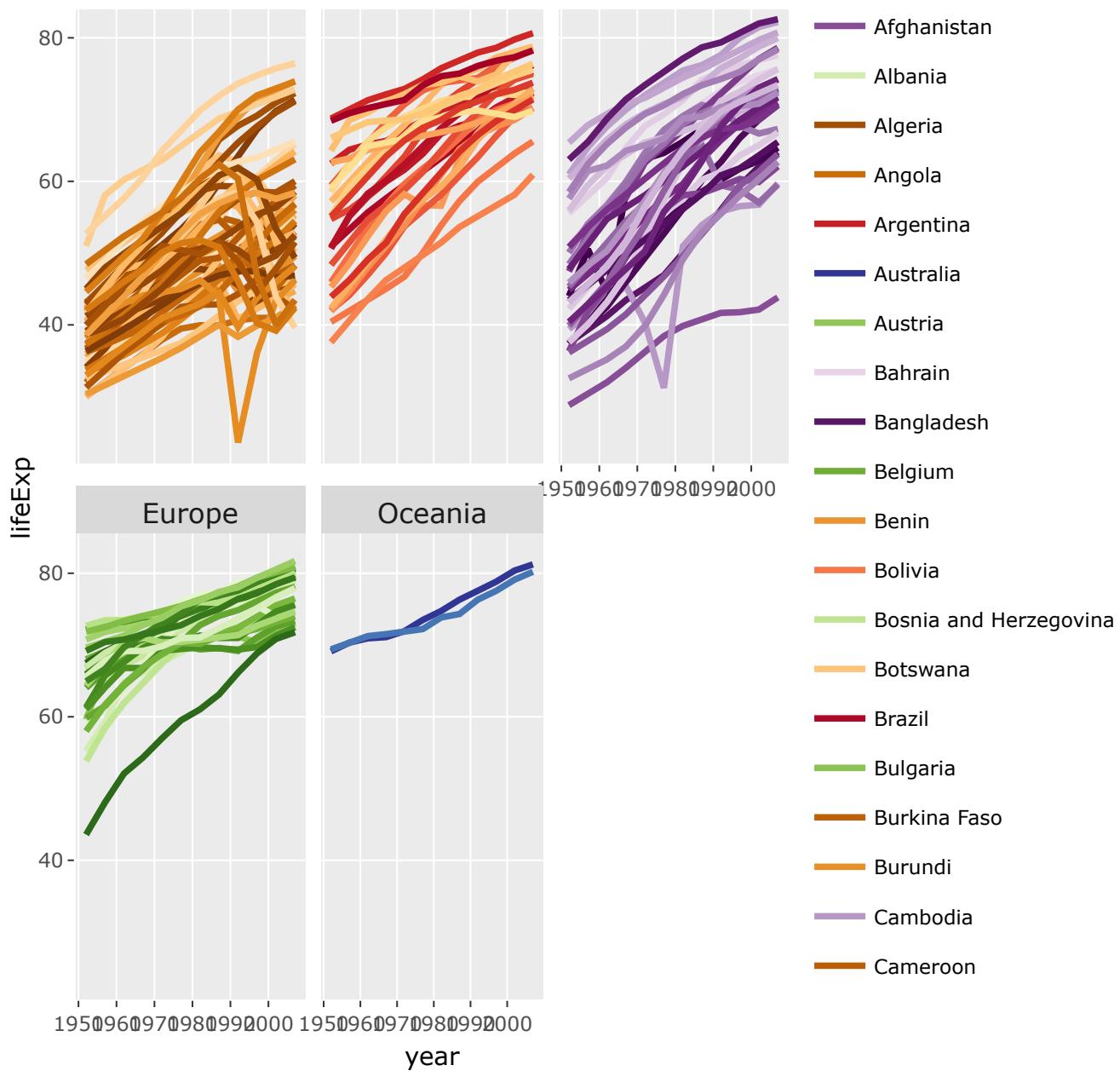
```



[Hide](#)

```
plotly::ggplotly(bp)
```





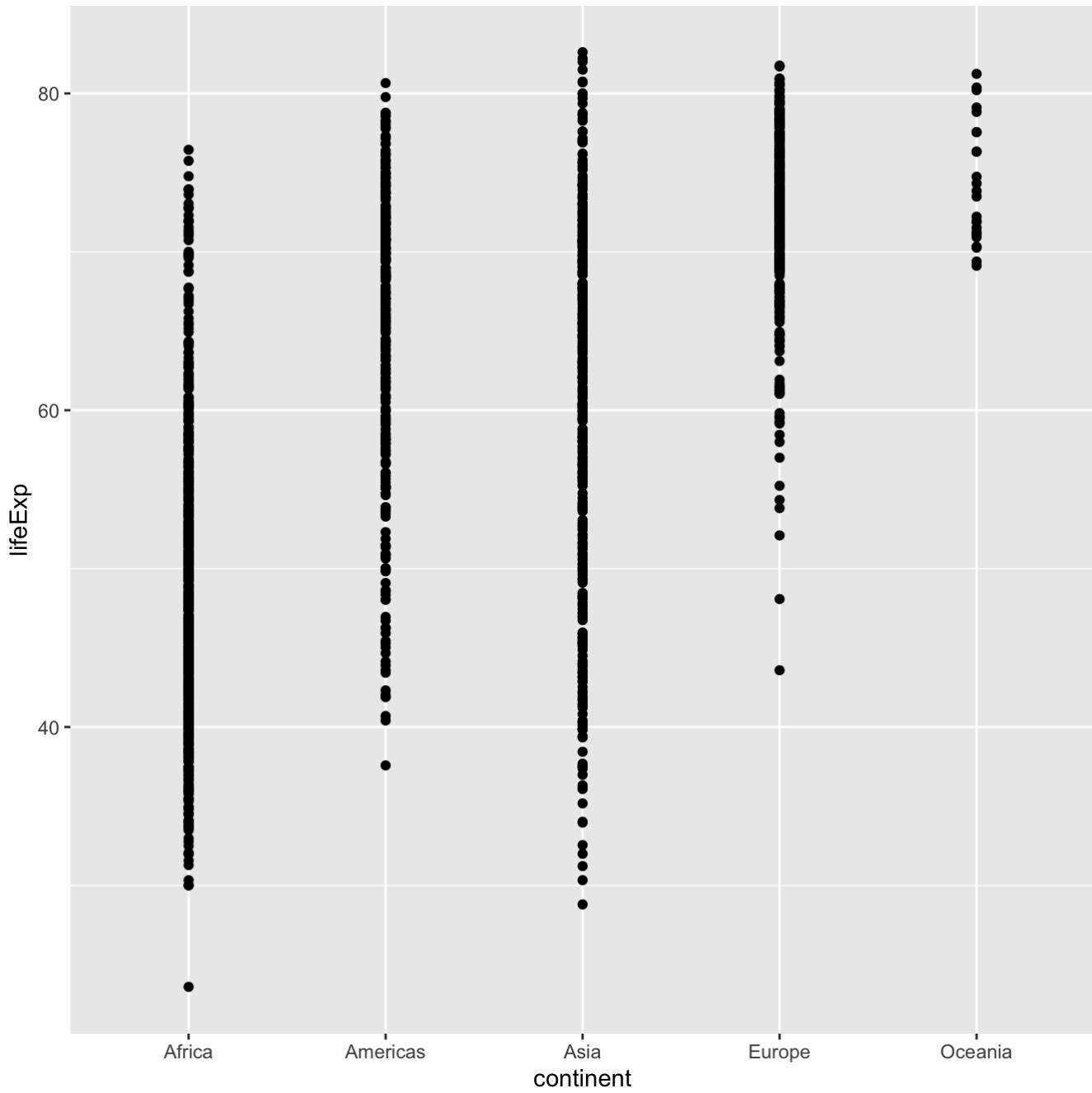
6.20.4 Boxplots

Hide

```
# now it is a categorical x VS continuous y
p <- ggplot(gapminder, aes(x = continent, y = lifeExp))
```

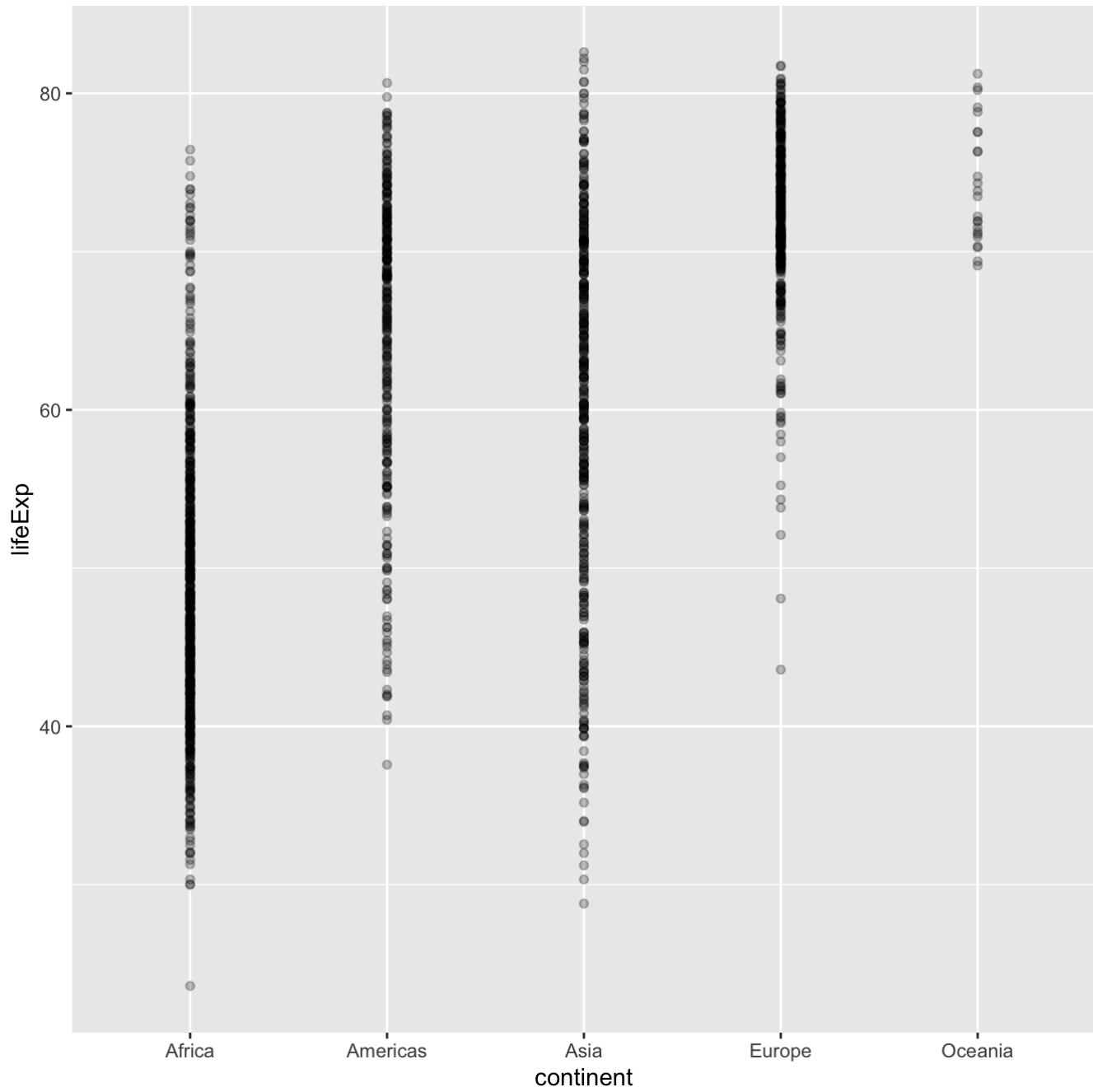
Hide

```
p + geom_point()
```



[Hide](#)

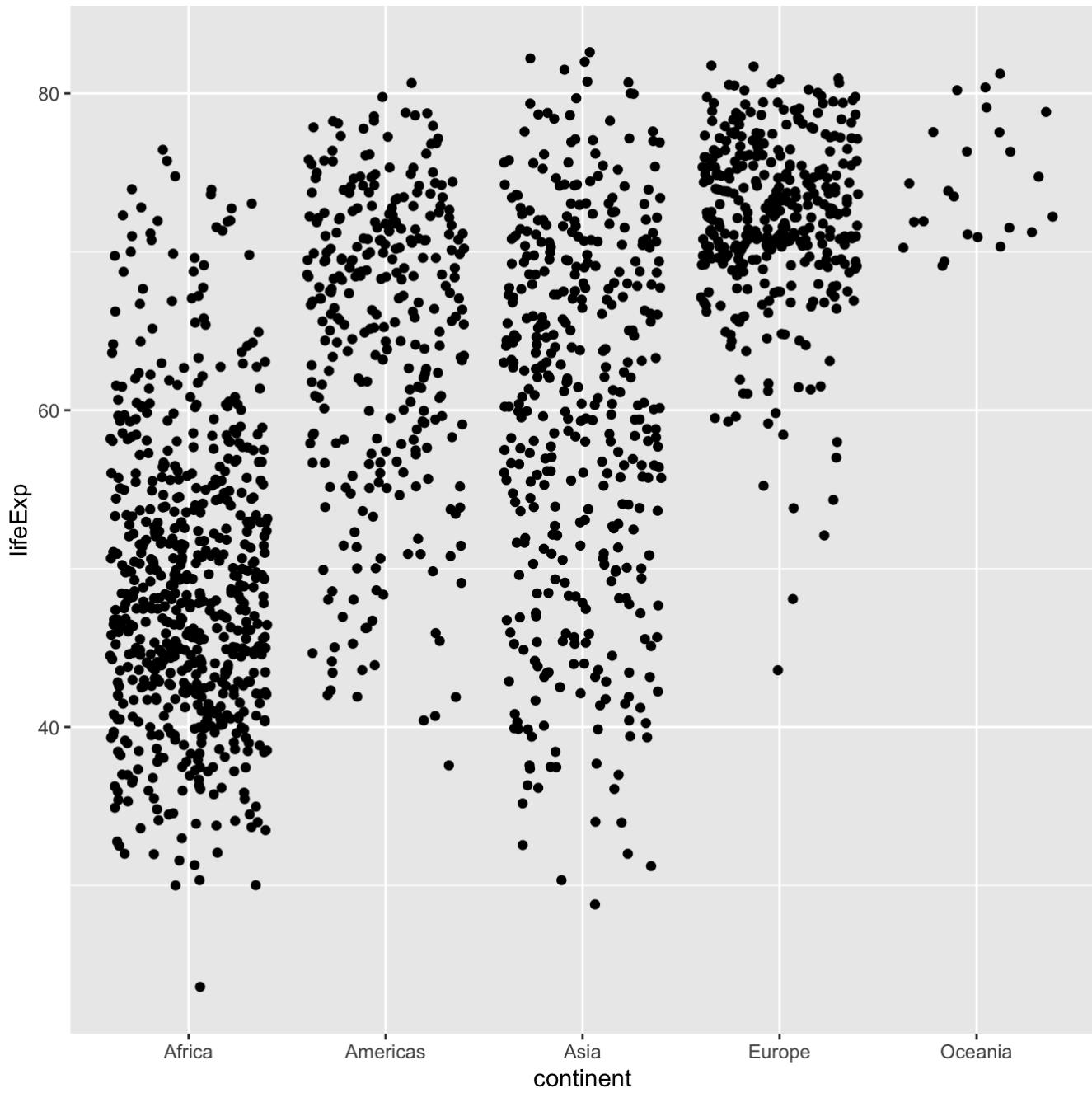
```
p + geom_point(alpha=1/4)
```



It is so easy to escape *dynamite* plots!

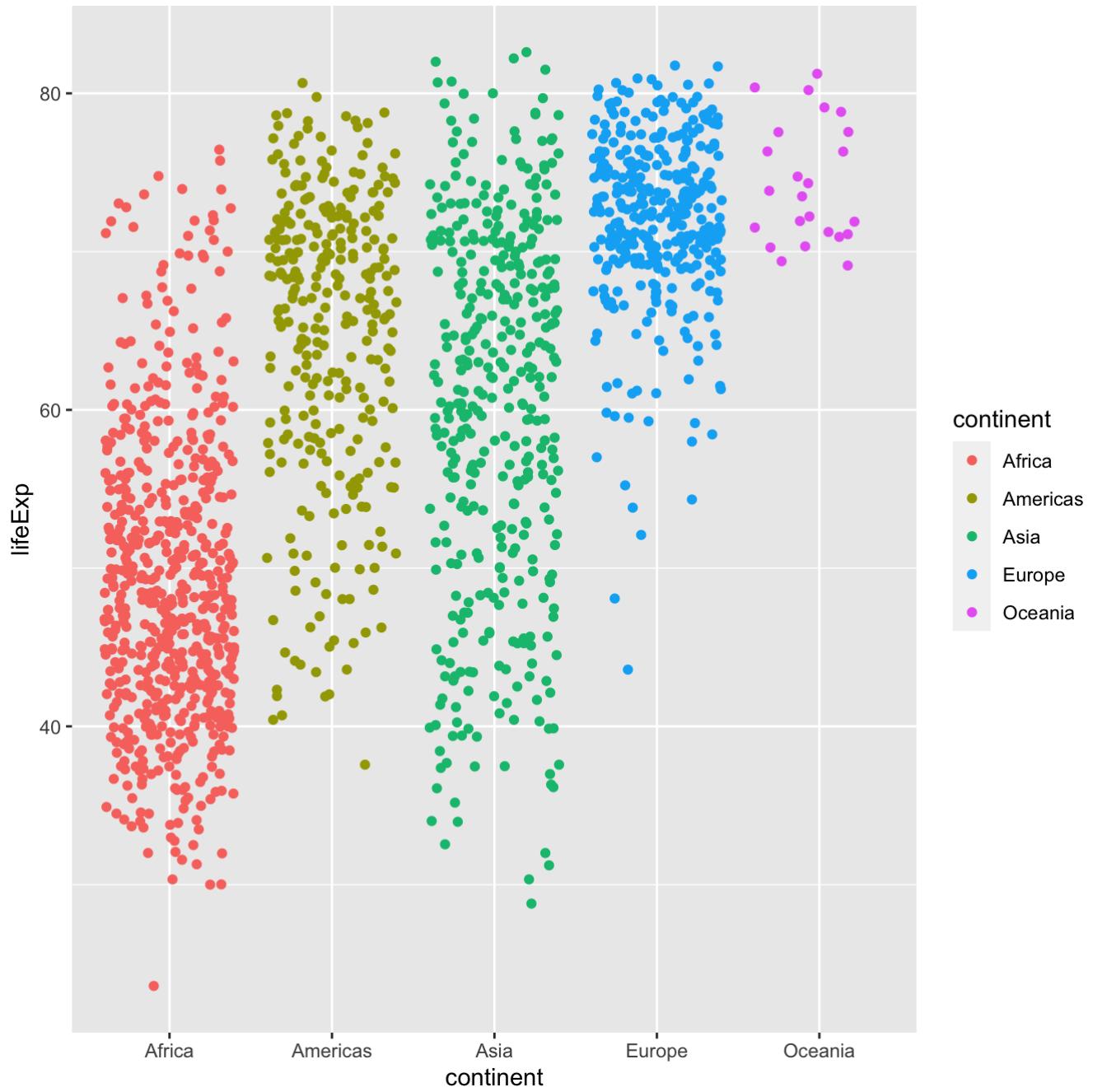
[Hide](#)

```
p + geom_jitter()
```



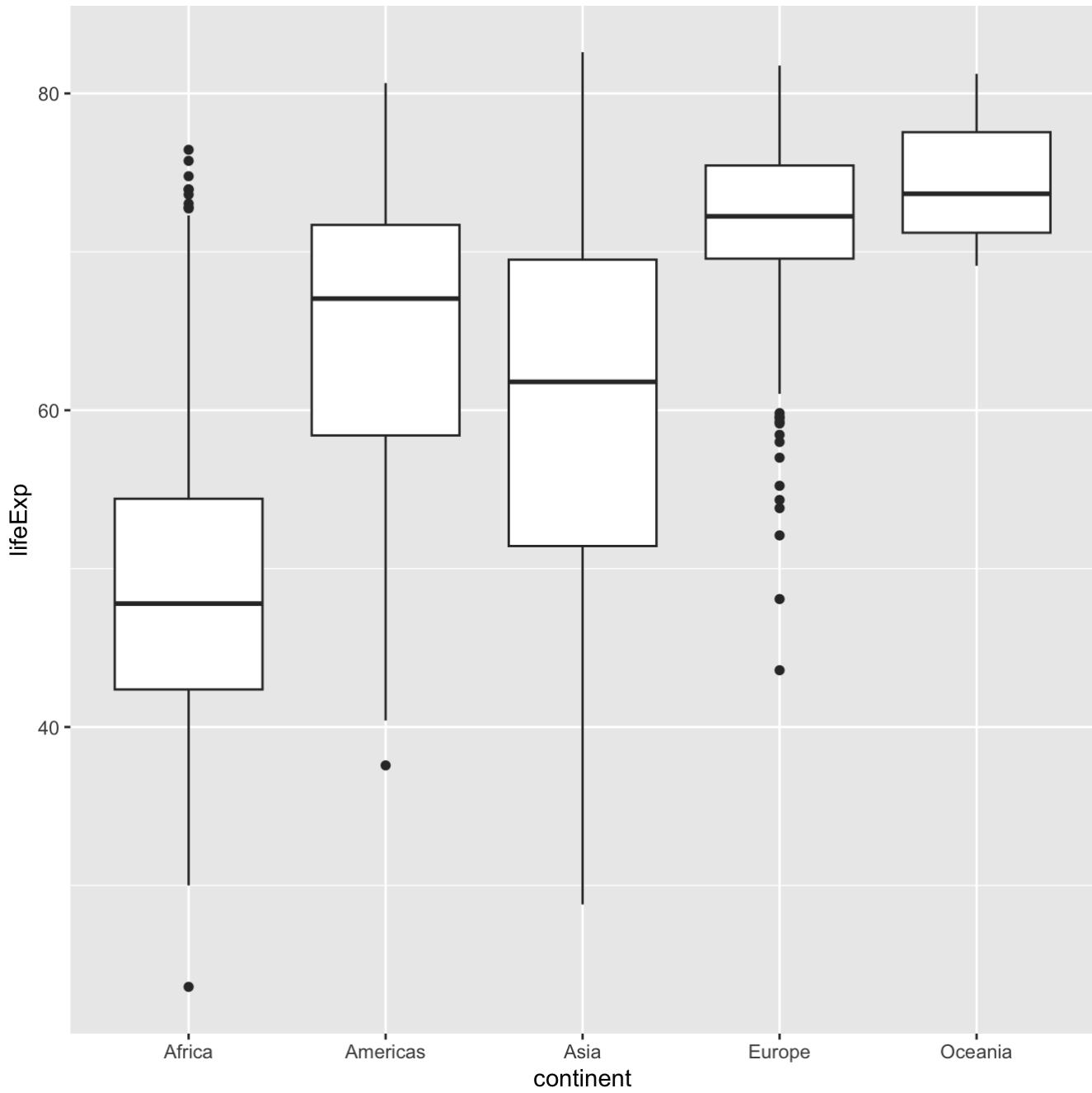
[Hide](#)

```
p + geom_jitter(aes(col=continent))
```



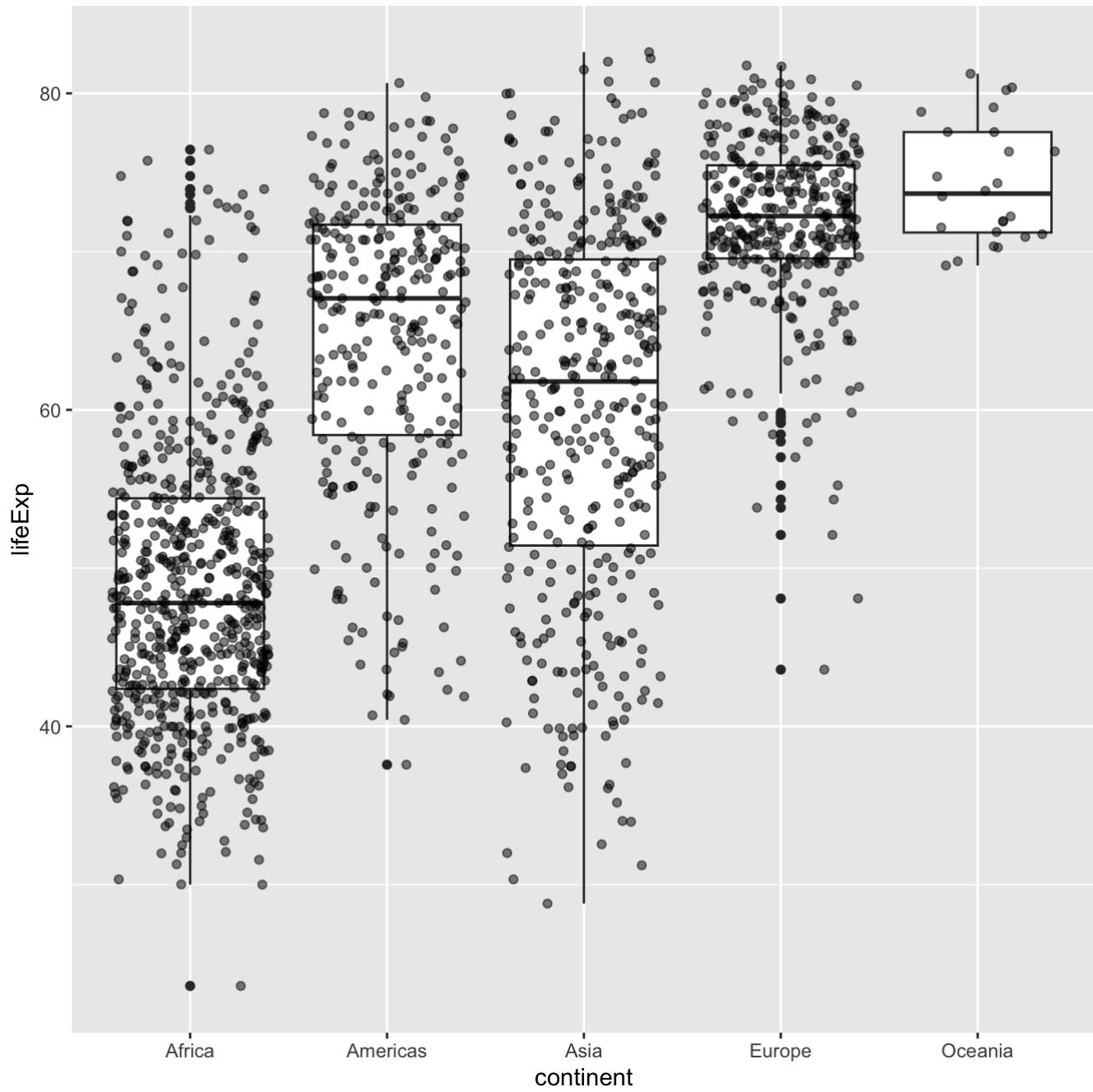
Hide

```
p + geom_boxplot()
```



Hide

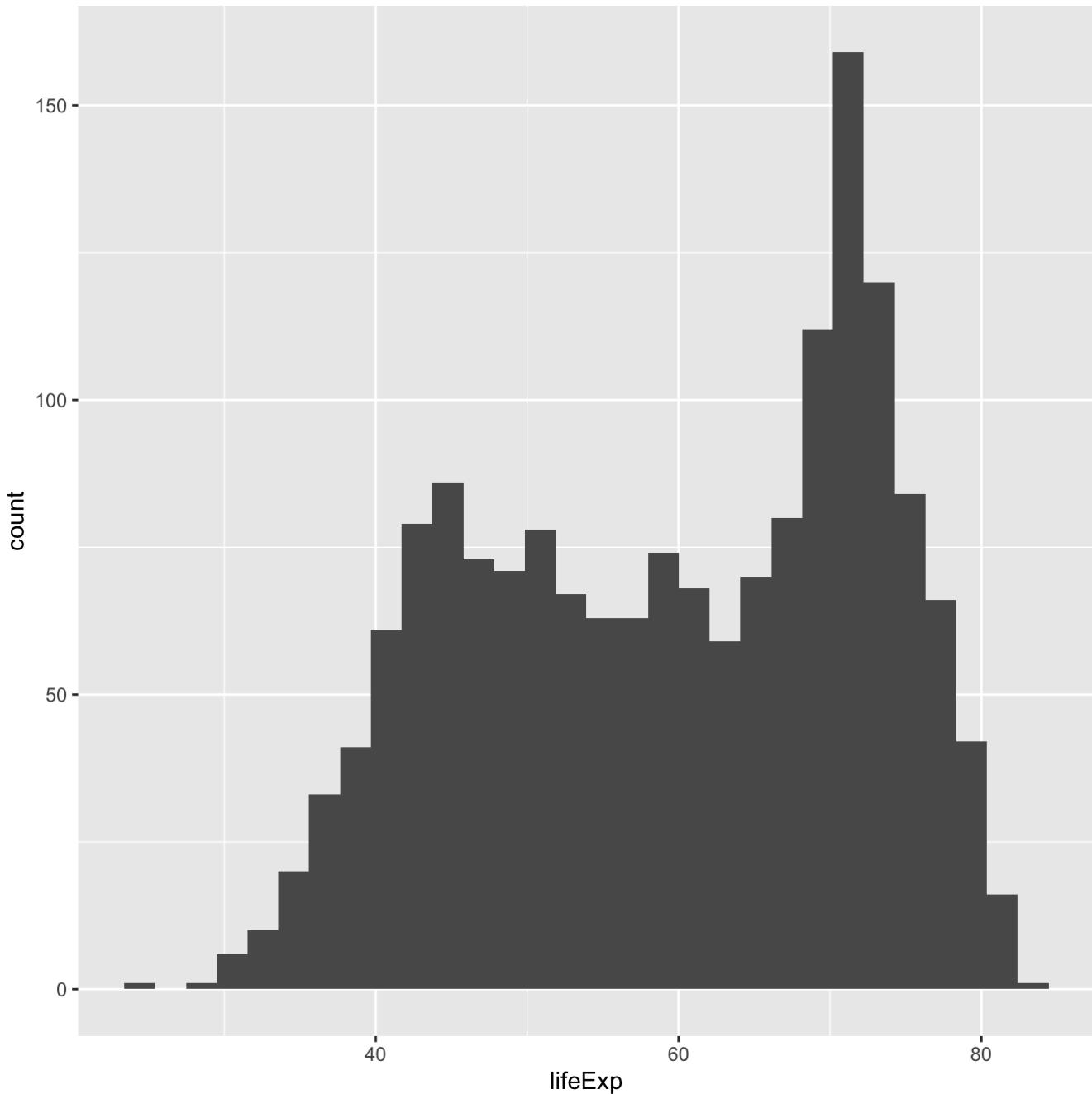
```
p + geom_boxplot() + geom_jitter(alpha=1/2)
```



6.20.5 Histograms

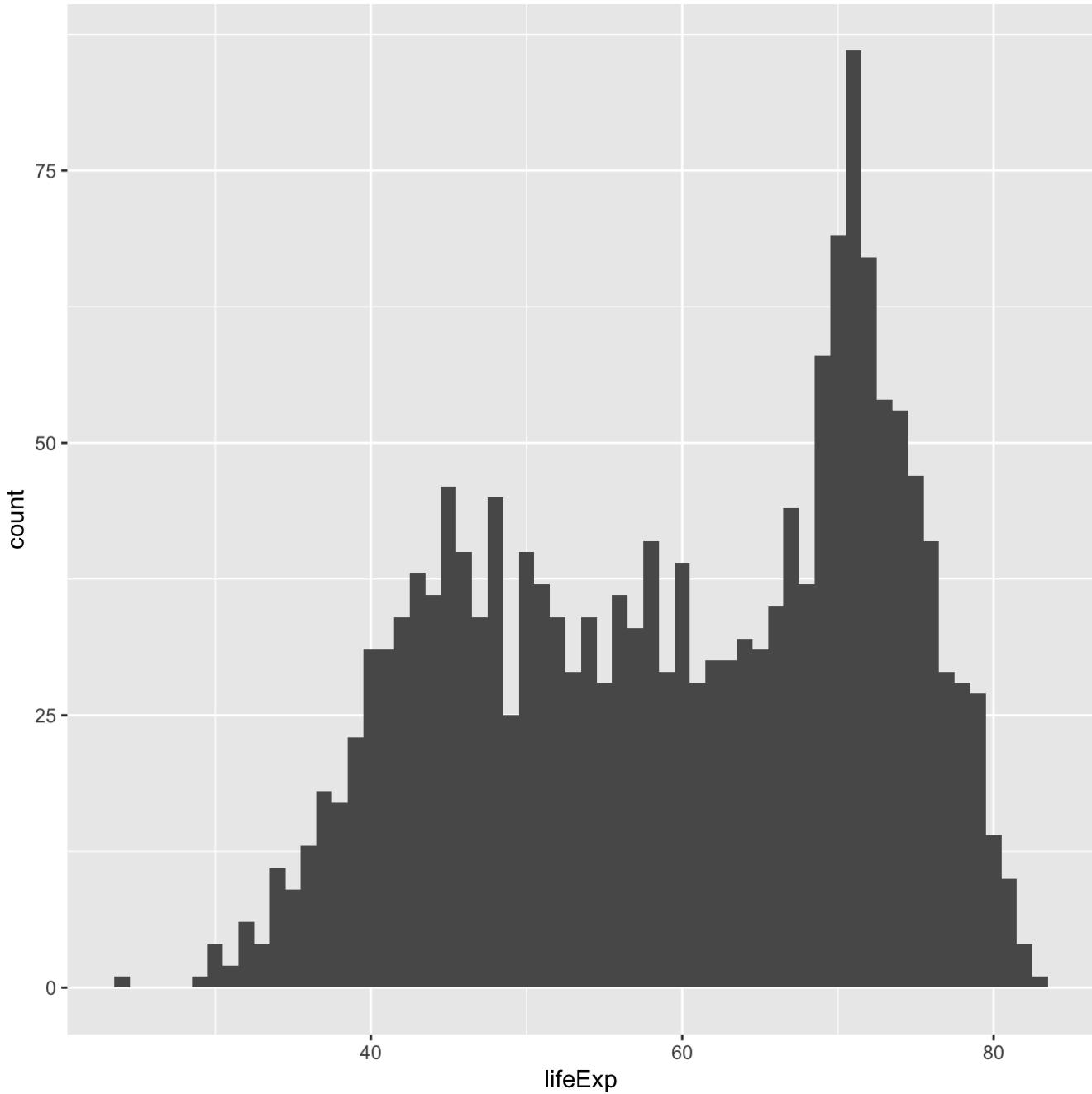
[Hide](#)

```
p <- ggplot(gapminder, aes(lifeExp))  
p + geom_histogram()
```



[Hide](#)

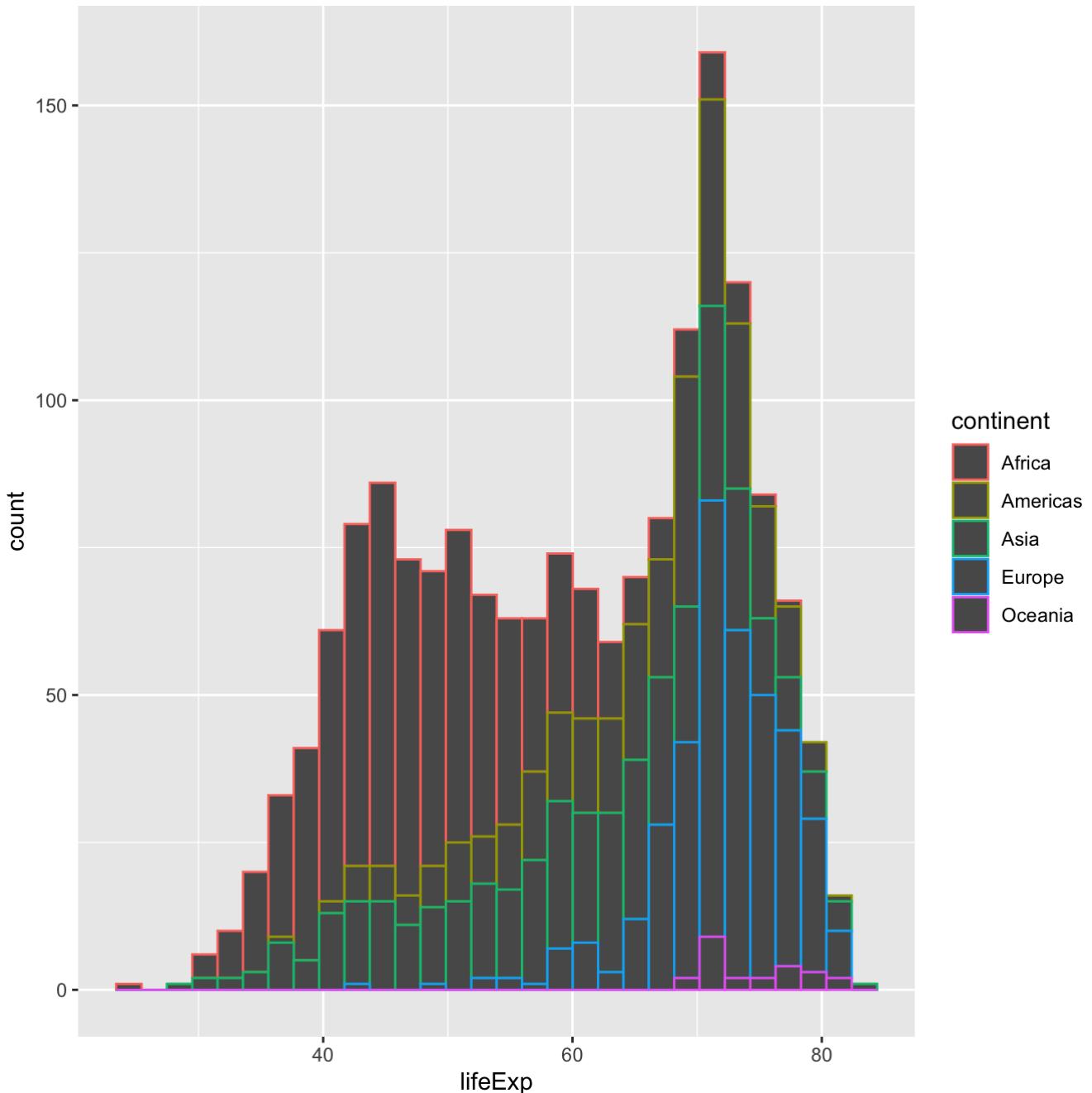
```
p + geom_histogram(binwidth=1)
```



Stacked histogram are much easier in this framework

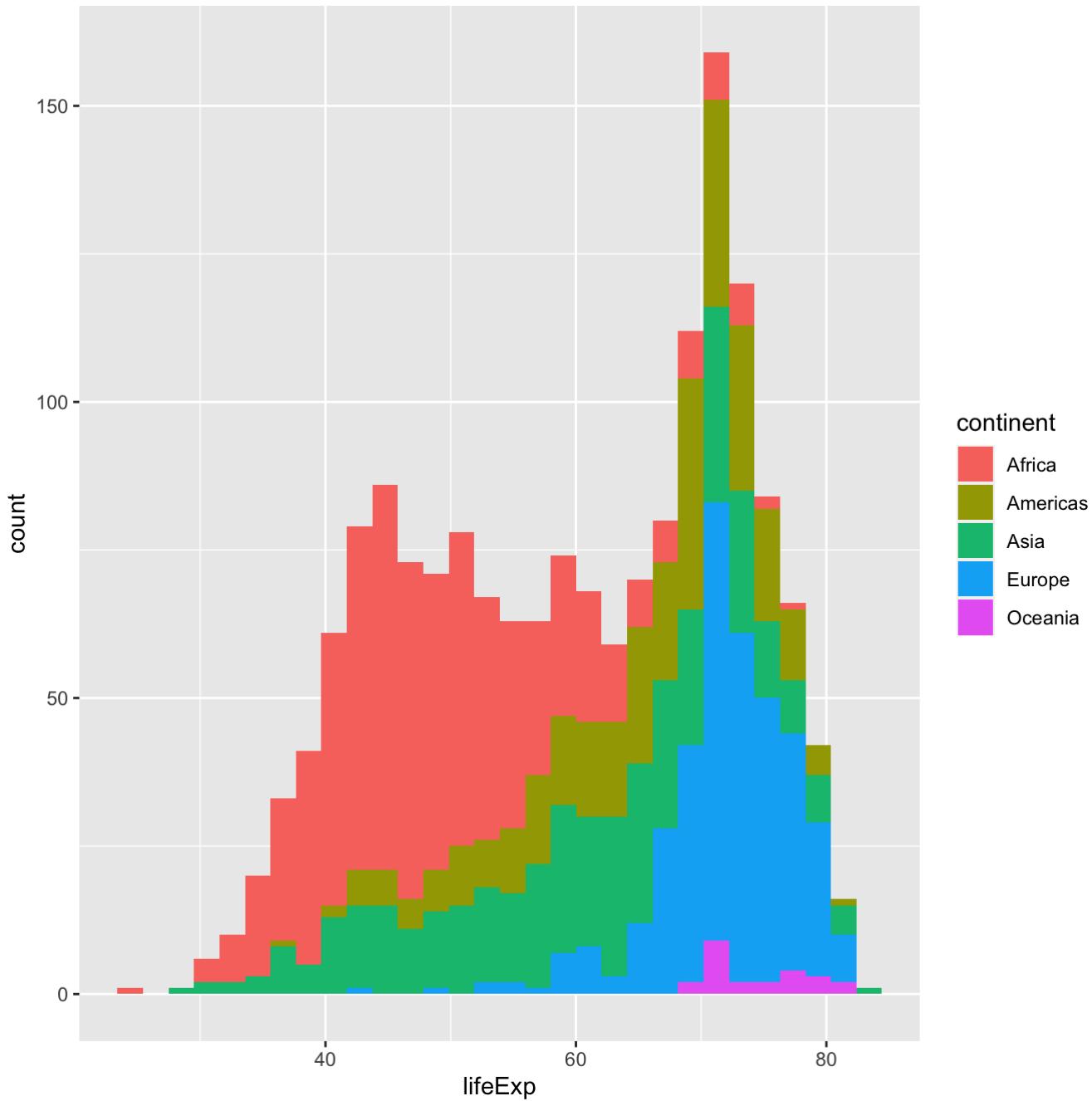
[Hide](#)

```
p + geom_histogram(aes(color=continent))
```

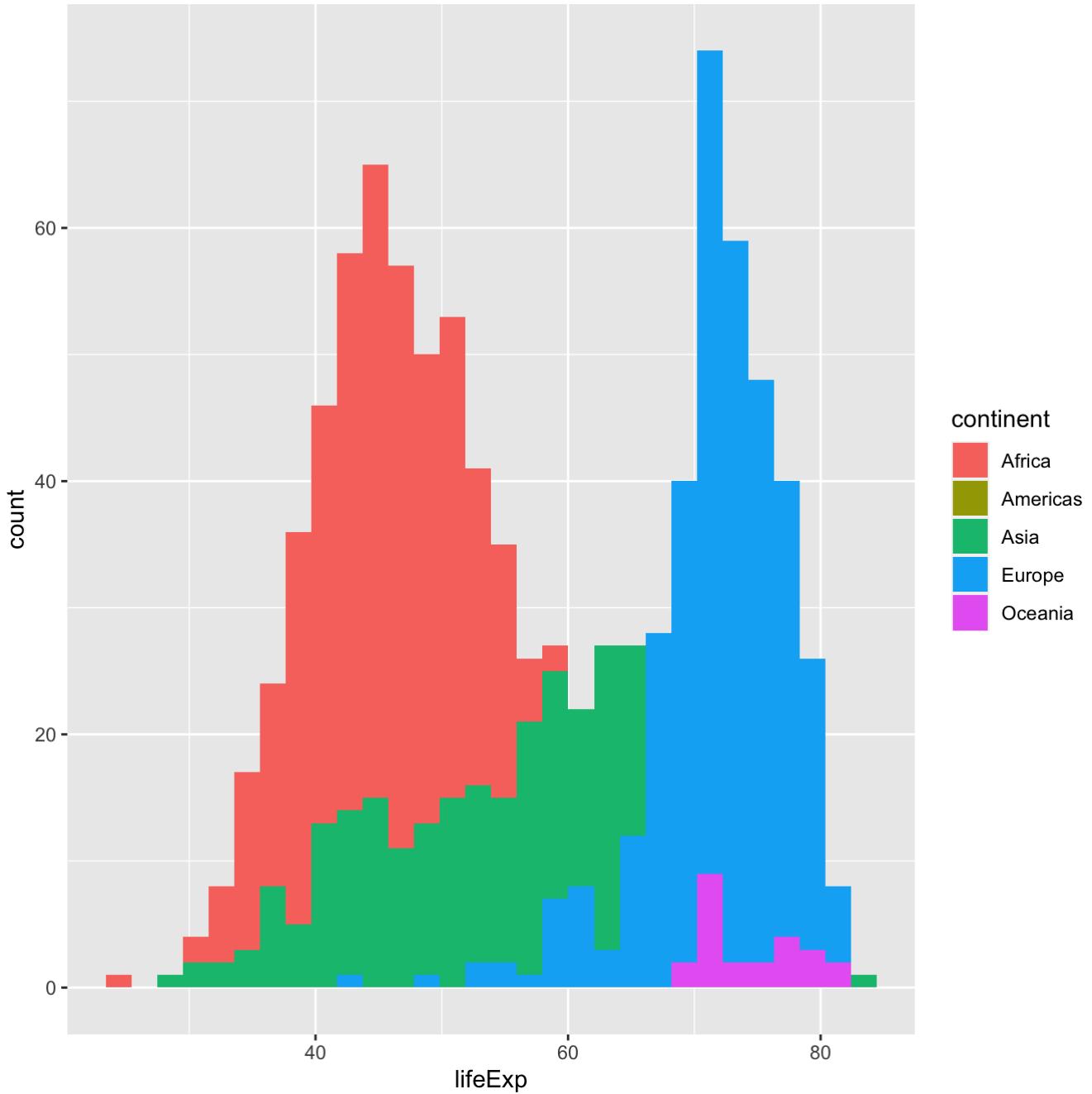


Hide

```
p + geom_histogram(aes(fill=continent))
```



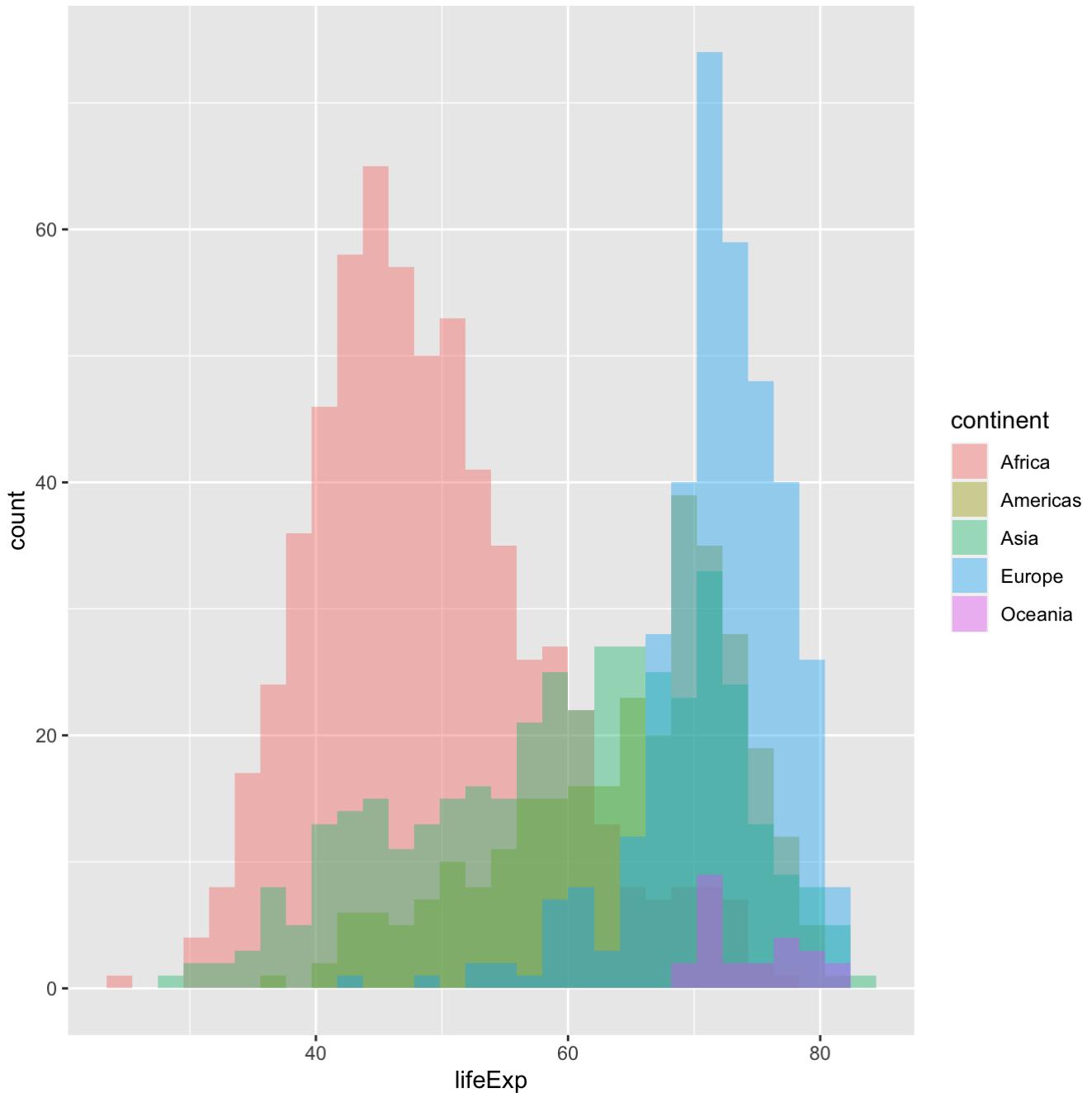
```
p + geom_histogram(aes(fill=continent), position="identity")
```



... and so is the superimposing of more than one distribution

[Hide](#)

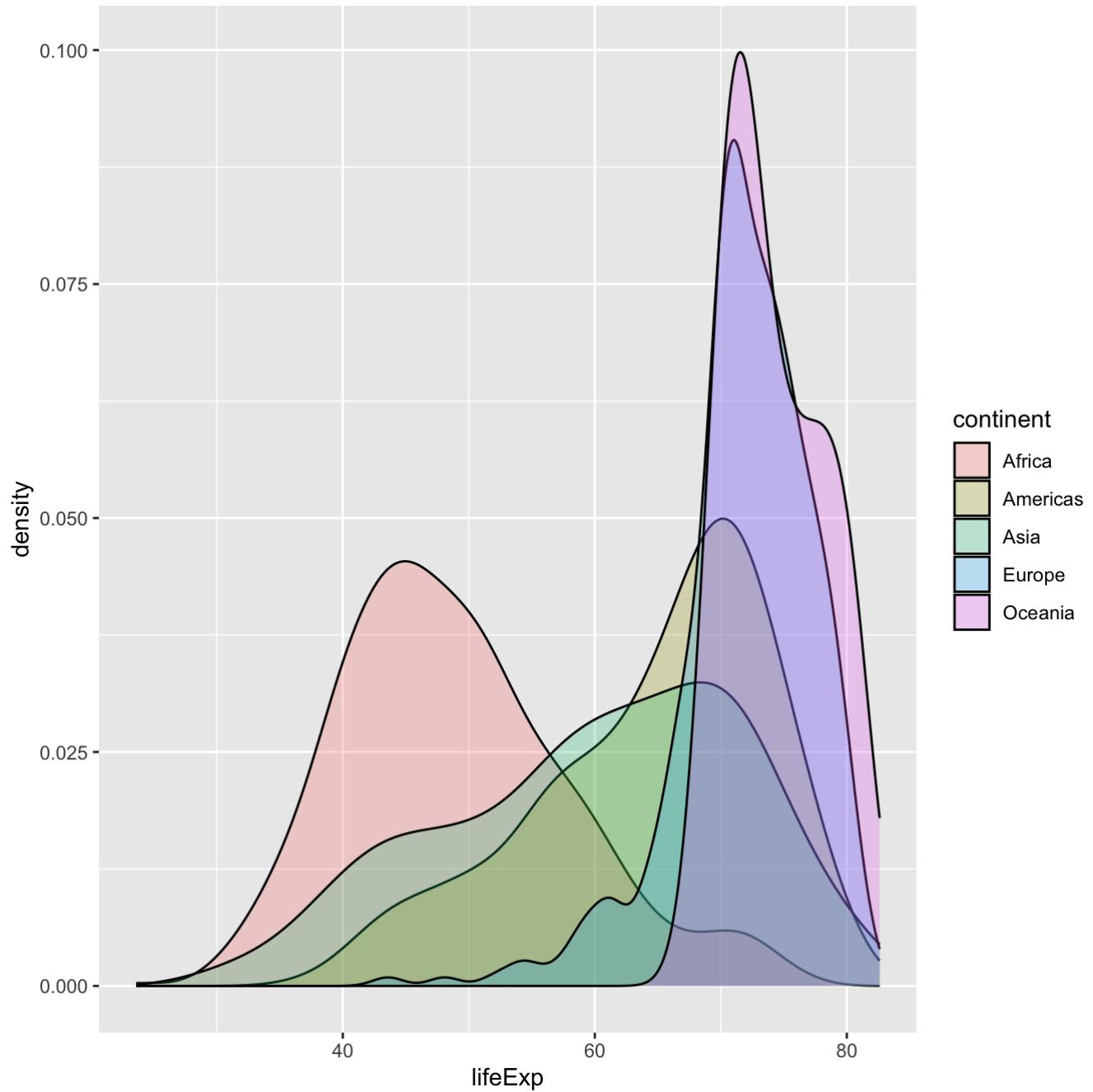
```
p + geom_histogram(aes(fill=continent), position="identity", alpha = 0.4)
```



Similar to histogram, you can use also density plots

[Hide](#)

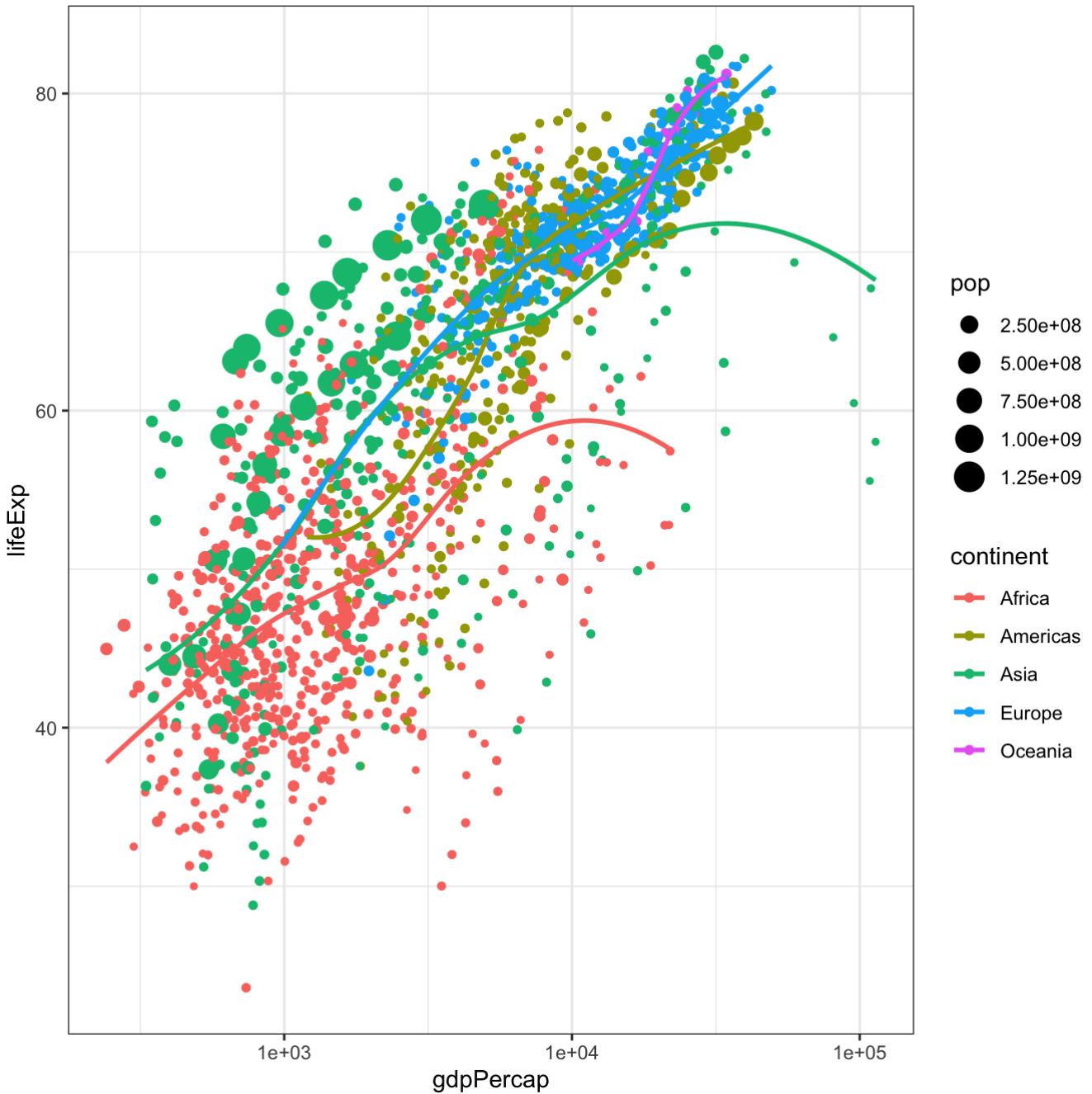
```
p + geom_density(aes(fill=continent), alpha=1/4)
```



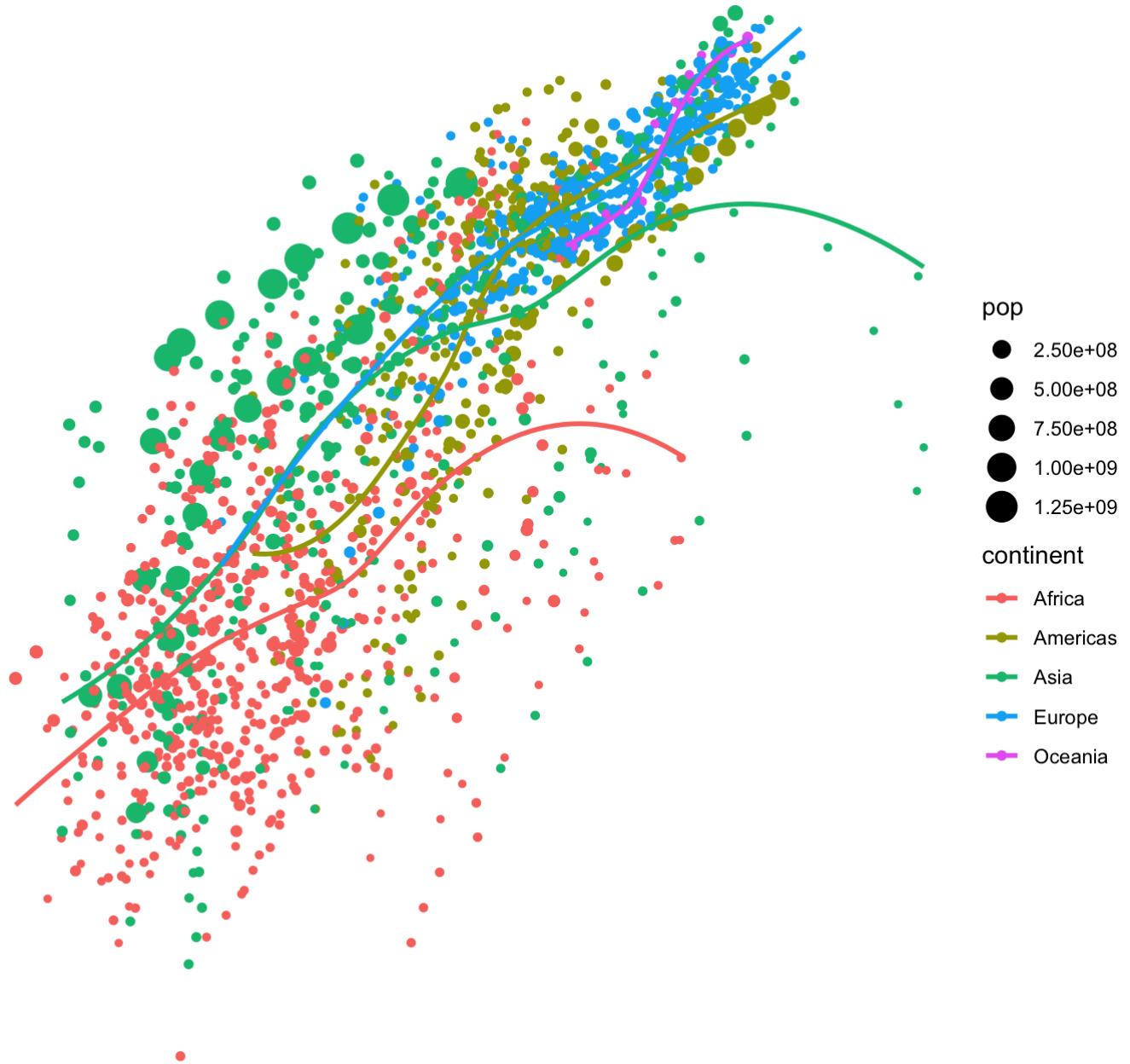
6.20.6 Themes: a quick way to put a new shirt on

[Hide](#)

```
nicerone + theme_bw()
```

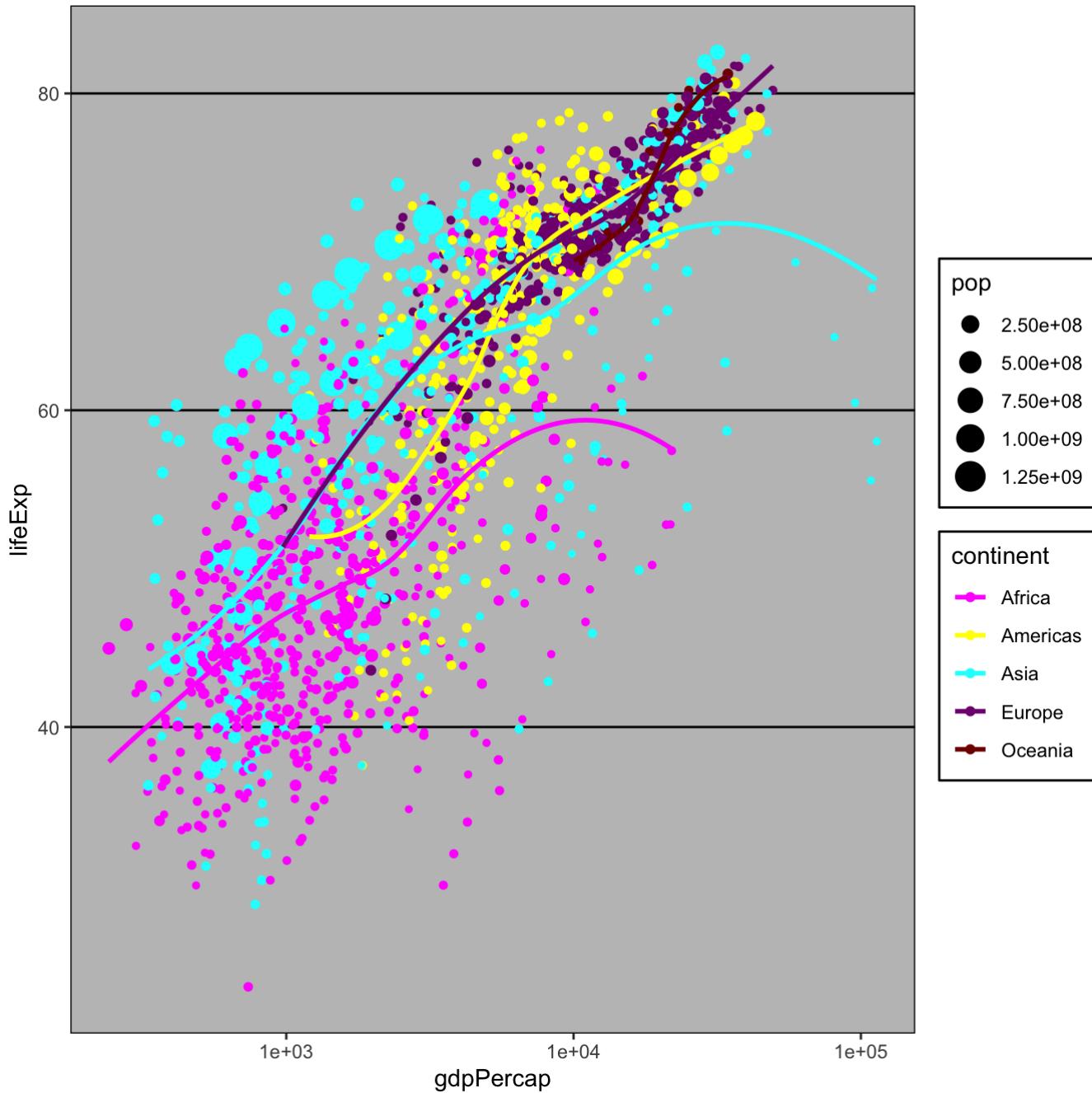


```
niceone + theme_void()
```



If you really really really have to...

```
library("ggthemes")
niceone + theme_excel() + scale_color_excel()
```



6.21 Exercise session 6 - Homework if you want

- try to recreate the plots you did with base graphics, this time using `ggplot2`
- pick a nice plot you would like to have in your next manuscript: can you think of what you need to do it? I am talking of
 - what data type?
 - what transformations?
 - what plot type/layer?

► Details

7 Step 5: SummarizedExperiment: your best friend for “bioinformatics datasets”

7.1 Next steps

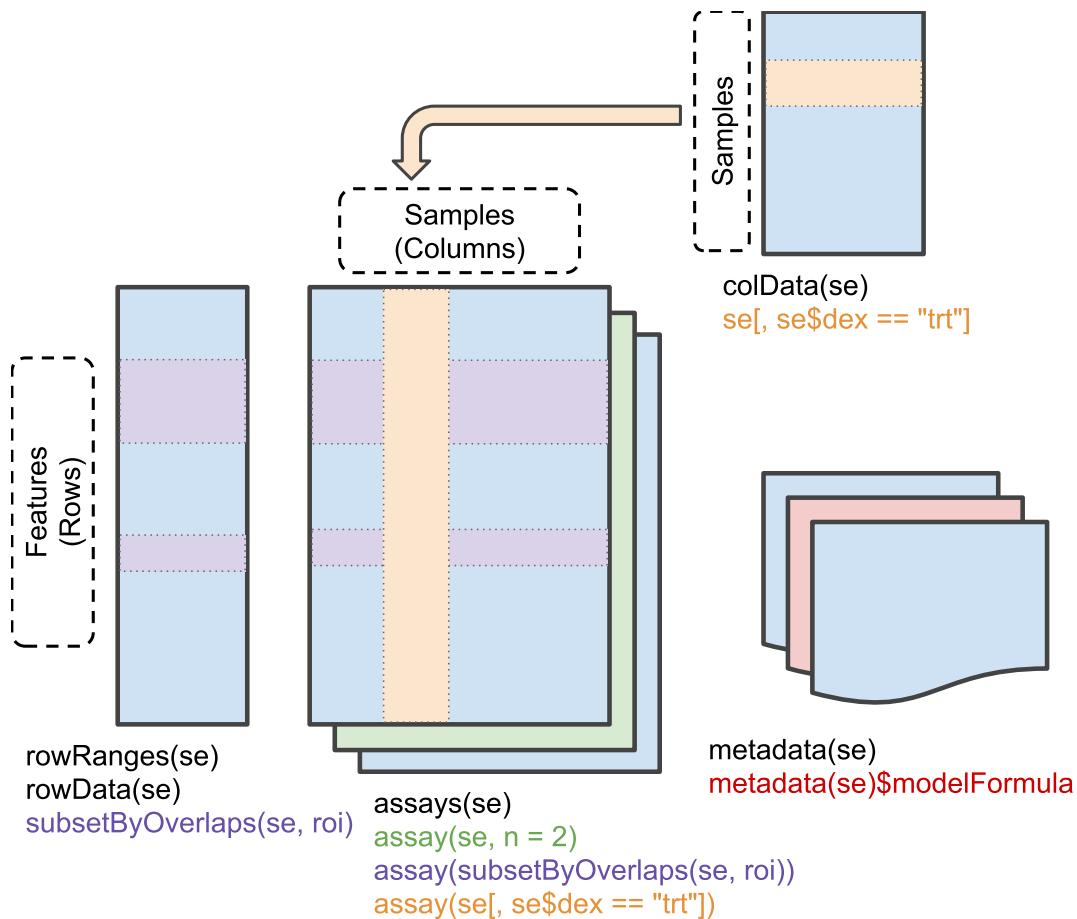
Data in bioinformatics is often complex. To deal with this, developers define specialised data containers (termed classes) that match the properties of the data they need to handle.

This aspect is central to the **Bioconductor**(<https://www.bioconductor.org>) project which uses the same **core data infrastructure** across packages. This certainly contributed to Bioconductor’s success. Bioconductor package developers are advised to make use of existing infrastructure to provide coherence, interoperability and stability to the project as a whole.

To illustrate such an omics data container, we’ll present the `SummarizedExperiment` class.

7.2 SummarizedExperiment

The figure below represents the anatomy of `SummarizedExperiment`.



Objects of the class `SummarizedExperiment` contain :

- **One (or more) assay(s)** containing the quantitative omics data (expression data), stored as a matrix-like object. Features (genes, transcripts, proteins, ...) are defined along the rows and samples along the columns.

- A **sample metadata** slot containing sample co-variates, stored as a data frame. Rows from this table represent samples (rows match exactly the columns of the expression data).
- A **feature metadata** slot containing feature co-variates, stored as data frame. The rows of this dataframe's match exactly the rows of the expression data.

The coordinated nature of the SummarizedExperiment guarantees that during data manipulation, the dimensions of the different slots will always match (i.e the columns in the expression data and then rows in the sample metadata, as well as the rows in the expression data and feature metadata) during data manipulation. For example, if we had to exclude one sample from the assay, it would be automatically removed from the sample metadata in the same operation.

The metadata slots can grow additional co-variates (columns) without affecting the other structures.

Questions

Q1 - Can you think of data examples what can fit into this container?

Q2 - What if the data has some “specific” peculiarities on top of this tabular-like representation?

7.2.1 Creating a SummarizedExperiment

Hide

```

rna <- read_csv("data/rnaseq.csv")
head(rna)
# A tibble: 6 × 19
  gene   sample expression organism    age sex   infection strain time tissue
  <chr>  <chr>      <dbl> <chr>     <dbl> <chr> <chr>     <chr> <dbl> <chr>
  <dbl>  <dbl> <chr>
1 Asl    GSM25...       1170 Mus mus...     8 Fema... Influenz... C57BL...     8 Cereb...
14 109900 argini...
2 Apod   GSM25...       36194 Mus mus...    8 Fema... Influenz... C57BL...     8 Cereb...
14 11815 apolip...
3 Cyp2d... GSM25...      4060 Mus mus...    8 Fema... Influenz... C57BL...     8 Cereb...
14 56448 cytoch...
4 Klk6   GSM25...       287  Mus mus...    8 Fema... Influenz... C57BL...     8 Cereb...
14 19144 kallik...
5 Fcrls  GSM25...       85   Mus mus...    8 Fema... Influenz... C57BL...     8 Cereb...
14 80891 Fc rec...
6 Slc2a4 GSM25...       782  Mus mus...    8 Fema... Influenz... C57BL...     8 Cereb...
14 20528 solute...
# i 6 more variables: ensembl_gene_id <chr>, external_synonym <chr>, chromosome_name <chr>,
# gene_biotype <chr>, phenotype_description <chr>, hsapiens_homolog_associated_gene_name <chr>
head(as.data.frame(rna))
  gene   sample expression    organism age   sex   infection strain ti
  me   tissue mouse
1 Asl   GSM2545336       1170 Mus musculus  8 Female InfluenzaA C57BL/6
8 Cerebellum 14
2 Apod   GSM2545336       36194 Mus musculus  8 Female InfluenzaA C57BL/6
8 Cerebellum 14
3 Cyp2d22 GSM2545336      4060 Mus musculus  8 Female InfluenzaA C57BL/6
8 Cerebellum 14
4 Klk6   GSM2545336       287  Mus musulus  8 Female InfluenzaA C57BL/6
8 Cerebellum 14
5 Fcrls  GSM2545336       85   Mus musulus  8 Female InfluenzaA C57BL/6
8 Cerebellum 14
6 Slc2a4 GSM2545336       782  Mus musulus  8 Female InfluenzaA C57BL/6
8 Cerebellum 14
  ENTREZID
product
1 109900                           argininosuccinate lyase, transcript
variant X1
2 11815                            apolipoprotein D, transcript
variant 3
3 56448 cytochrome P450, family 2, subfamily d, polypeptide 22, transcript
variant 2
4 19144                            kallikrein related-peptidase 6, transcript
variant 2

```

```

5     80891          Fc receptor-like S, scavenger receptor, transcript
variant X1
6     20528          solute carrier family 2 (facilitated glucose transport
r), member 4
    ensembl_gene_id external_synonym chromosome_name gene_biotype
1 ENSMUSG00000025533 2510006M18Rik           5 protein_coding
2 ENSMUSG00000022548 <NA>                  16 protein_coding
3 ENSMUSG00000061740 2D22                  15 protein_coding
4 ENSMUSG00000050063 Bssp                  7 protein_coding
5 ENSMUSG00000015852 2810439C17Rik         3 protein_coding
6 ENSMUSG00000018566 Glut-4                 11 protein_coding
phenotype_description hsapiens_homolog_associated
_gene_name
1      abnormal circulating amino acid level
ASL
2      abnormal lipid homeostasis
APOD
3      abnormal skin morphology
CYP2D6
4      abnormal cytokine level
KLK6
5 decreased CD8-positive alpha-beta T cell number
FCRL2
6      abnormal circulating glucose level
SLC2A4

```

Remember the `rna` dataset that we have used previously.

From this table we have already created 3 different tables - we read them in as serialized r objects.

[Hide](#)

```

count_matrix <- readRDS("data/count_matrix.RDS")
sample_metadata <- readRDS("data/sample_metadata.RDS")
gene_metadata <- readRDS("data/gene_metadata.RDS")

```

- **An expression matrix**

[Hide](#)

```

count_matrix[1:5, ]
  GSM2545336 GSM2545337 GSM2545338 GSM2545339 GSM2545340 GSM2545341 GSM
2545342 GSM2545343
Asl       1170      361      400      586      626      988
836       535
Apod     36194     10347     9173     10620     13021     29594
24959     13668
Cyp2d22   4060      1616     1603     1901      2171     3349
3122      2008
Klk6       287       629      641      578      448      195
186       1101
Fcrls     85        233      244      237      180       38
68        375
  GSM2545344 GSM2545345 GSM2545346 GSM2545347 GSM2545348 GSM2545349 GSM
2545350 GSM2545351
Asl       586       597      938      1035      494      481
666       937
Apod     13230     15868     27769     34301     11258     11812
15816     29242
Cyp2d22   2254      2277     2985      3452      1883     2014
2417      3678
Klk6       537       567      327      233      742      881
828       250
Fcrls     199       177      89       67      300      233
231       81
  GSM2545352 GSM2545353 GSM2545354 GSM2545362 GSM2545363 GSM2545380
Asl       803       541      473      748      576      1192
Apod     20415     13682     11088     15916     11166     38148
Cyp2d22   2920      2216     1821      2842      2011      4019
Klk6       798       710      894      501      598      259
Fcrls     303       285      248      179      184       68
dim(count_matrix)
[1] 1474    22

```

- **A table describing the samples**

Hide

```

sample_metadata
# A tibble: 22 × 9
  sample      organism    age sex infection   strain  time tissue
  <chr>     <chr>     <dbl> <chr>    <chr>    <chr>    <dbl> <chr>
mouse
<dbl>
  1 GSM2545336 Mus musculus     8 Female InfluenzaA C57BL/6     8 Cerebellum
14
  2 GSM2545337 Mus musculus     8 Female NonInfected C57BL/6     0 Cerebellum
9
  3 GSM2545338 Mus musculus     8 Female NonInfected C57BL/6     0 Cerebellum
10
  4 GSM2545339 Mus musculus     8 Female InfluenzaA C57BL/6     4 Cerebellum
15
  5 GSM2545340 Mus musculus     8 Male   InfluenzaA C57BL/6     4 Cerebellum
18
  6 GSM2545341 Mus musculus     8 Male   InfluenzaA C57BL/6     8 Cerebellum
6
  7 GSM2545342 Mus musculus     8 Female InfluenzaA C57BL/6     8 Cerebellum
5
  8 GSM2545343 Mus musculus     8 Male   NonInfected C57BL/6     0 Cerebellum
11
  9 GSM2545344 Mus musculus     8 Female InfluenzaA C57BL/6     4 Cerebellum
22
10 GSM2545345 Mus musculus     8 Male   InfluenzaA C57BL/6     4 Cerebellum
13
# i 12 more rows

```

- **A table describing the genes**

[Hide](#)

```

gene_metadata
# A tibble: 1,474 × 9
  gene      ENTREZID product      ensembl_gene_id external_synonym chrom
  <chr>     <dbl>   <chr>      <chr>        <chr>        <chr>
  osome_name gene_biotype
  <chr>      <dbl>   <chr>      <chr>        <chr>        <chr>
<chr>
  1 Asl       109900 argininosuccinate... ENSMUSG0000002... 2510006M18Rik 5
protein_cod...
  2 Apod      11815 apolipoprotein D,... ENSMUSG0000002... <NA>        16
protein_cod...
  3 Cyp2d22   56448 cytochrome P450, ... ENSMUSG0000006... 2D22        15
protein_cod...
  4 Klk6      19144 kallikrein relate... ENSMUSG0000005... Bssp        7
protein_cod...
  5 Fcrls     80891 Fc receptor-like ... ENSMUSG0000001... 2810439C17Rik 3
protein_cod...
  6 Slc2a4    20528 solute carrier fa... ENSMUSG0000001... Glut-4       11
protein_cod...
  7 Exd2      97827 exonuclease 3'-5'... ENSMUSG0000003... 4930539P14Rik 12
protein_cod...
  8 Gjc2      118454 gap junction prot... ENSMUSG0000004... B230382L12Rik 11
protein_cod...
  9 Plp1      18823 proteolipid prote... ENSMUSG0000003... DM20        X
protein_cod...
  10 Gnb4     14696 guanine nucleotid... ENSMUSG0000002... 6720453A21Rik 3
protein_cod...
# i 1,464 more rows
# i 2 more variables: phenotype_description <chr>, hsapiens_homolog_associated_gene_name <chr>

```

We will create a `SummarizedExperiment` from these tables:

- The count matrix that will be used as the `assay`
- The table describing the samples will be used as the `sample metadata` slot
- The table describing the genes will be used as the `features metadata` slot

To do this we can put the different parts together using the `SummarizedExperiment` constructor:

Hide

```

#BiocManager::install("SummarizedExperiment")
library("SummarizedExperiment")

```

Hide

```
se <- SummarizedExperiment(assays = count_matrix,
                           colData = sample_metadata,
                           rowData = gene_metadata)
se
class: SummarizedExperiment
dim: 1474 22
metadata(0):
assays(1): ''
rownames(1474): Asl Apod ... Lmx1a Pbx1
rowData names(9): gene ENTREZID ... phenotype_description
  hsapiens_homolog_associated_gene_name
colnames(22): GSM2545336 GSM2545337 ... GSM2545363 GSM2545380
colData names(9): sample organism ... tissue mouse
```

Using this data structure, we can access the expression matrix with the `assay` function:

[Hide](#)

```

head(assay(se))
      GSM2545336 GSM2545337 GSM2545338 GSM2545339 GSM2545340 GSM2545341 GSM
2545342 GSM2545343
Asl       1170        361        400        586        626        988
836       535
Apod      36194       10347       9173       10620       13021       29594
24959     13668
Cyp2d22   4060        1616       1603       1901       2171       3349
3122      2008
Klk6       287         629        641        578        448        195
186       1101
Fcrls     85          233        244        237        180         38
68        375
Slc2a4    782         231        248        265        313       786
528       249
      GSM2545344 GSM2545345 GSM2545346 GSM2545347 GSM2545348 GSM2545349 GSM
2545350 GSM2545351
Asl       586         597        938       1035        494        481
666       937
Apod      13230       15868       27769       34301       11258       11812
15816     29242
Cyp2d22   2254        2277       2985       3452        1883       2014
2417      3678
Klk6       537         567        327        233        742        881
828       250
Fcrls     199         177        89          67        300        233
231       81
Slc2a4    266         357        654        693        271        304
349       715
      GSM2545352 GSM2545353 GSM2545354 GSM2545362 GSM2545363 GSM2545380
Asl       803         541        473        748        576        1192
Apod      20415       13682       11088       15916       11166       38148
Cyp2d22   2920        2216       1821        2842       2011        4019
Klk6       798         710        894        501        598        259
Fcrls     303         285        248        179        184         68
Slc2a4    513         320        248        350        317        796
dim(assay(se))
[1] 1474    22

```

We can access the sample metadata using the `colData` function:

[Hide](#)

```

colData(se)
DataFrame with 22 rows and 9 columns
      sample   organism    age     sex infection st
rain    time
<character> <character> <numeric> <character> <character> <charac-
ter> <numeric>
GSM2545336  GSM2545336 Mus musculus      8 Female InfluenzaA C57
BL/6          8
GSM2545337  GSM2545337 Mus musculus      8 Female NonInfected C57
BL/6          0
GSM2545338  GSM2545338 Mus musculus      8 Female NonInfected C57
BL/6          0
GSM2545339  GSM2545339 Mus musculus      8 Female InfluenzaA C57
BL/6          4
GSM2545340  GSM2545340 Mus musculus      8 Male  InfluenzaA C57
BL/6          4
...
...
GSM2545353  GSM2545353 Mus musculus      8 Female NonInfected C57
BL/6          0
GSM2545354  GSM2545354 Mus musculus      8 Male  NonInfected C57
BL/6          0
GSM2545362  GSM2545362 Mus musculus      8 Female InfluenzaA C57
BL/6          4
GSM2545363  GSM2545363 Mus musculus      8 Male  InfluenzaA C57
BL/6          4
GSM2545380  GSM2545380 Mus musculus      8 Female InfluenzaA C57
BL/6          8
      tissue   mouse
<character> <numeric>
GSM2545336  Cerebellum      14
GSM2545337  Cerebellum      9
GSM2545338  Cerebellum      10
GSM2545339  Cerebellum      15
GSM2545340  Cerebellum      18
...
...
GSM2545353  Cerebellum      4
GSM2545354  Cerebellum      2
GSM2545362  Cerebellum      20
GSM2545363  Cerebellum      12
GSM2545380  Cerebellum      19
dim(colData(se))
[1] 22   9

```

We can also access the feature metadata using the `rowData` function:

[Hide](#)

```

head(rowData(se))
DataFrame with 6 rows and 9 columns
      gene ENTREZID          product ensembl_gene_id exter-
nal_synonym
<character> <numeric> <character> <character>
Asl           Asl     109900 argininosuccinate ly.. ENSMUSG00000025533 25
10006M18Rik
Apod          Apod    11815 apolipoprotein D, tr.. ENSMUSG00000022548
NA
Cyp2d22       Cyp2d22  56448 cytochrome P450, fam.. ENSMUSG00000061740
2D22
Klk6          Klk6    19144 kallikrein related-p.. ENSMUSG00000050063
Bssp
Fcrls         Fcrls   80891 Fc receptor-like S, .. ENSMUSG00000015852 28
10439C17Rik
Slc2a4         Slc2a4  20528 solute carrier famil.. ENSMUSG00000018566
Glut-4

      chromosome_name gene_biotype phenotype_description hsapiens_homolo-
g_associated_gene_name
<character> <character> <character>
<character>
Asl           5 protein_coding abnormal circulating..
ASL
Apod          16 protein_coding abnormal lipid homeo..
APOD
Cyp2d22       15 protein_coding abnormal skin morpho..
CYP2D6
Klk6          7 protein_coding abnormal cytokine le..
KLK6
Fcrls         3 protein_coding decreased CD8-positi..
FCRL2
Slc2a4         11 protein_coding abnormal circulating..
SLC2A4
dim(rowData(se))
[1] 1474      9

```

7.2.2 Subsetting a SummarizedExperiment

`SummarizedExperiment` can be subset just like with data frames, with numerics or with characters or logicals.

Below, we create a new instance of class `SummarizedExperiment` that contains only the 5 first features for the 3 first samples.

Hide

```
sel <- se[1:5, 1:3]
sel
class: SummarizedExperiment
dim: 5 3
metadata(0):
assays(1): ''
rownames(5): Asl Apod Cyp2d22 Klk6 Fcrls
rowData names(9): gene ENTREZID ... phenotype_description
  hsapiens_homolog_associated_gene_name
colnames(3): GSM2545336 GSM2545337 GSM2545338
colData names(9): sample organism ... tissue mouse
```

[Hide](#)

```

colData(sel)
DataFrame with 3 rows and 9 columns
      sample   organism     age     sex infection    st
rain     time
<character> <character> <numeric> <character> <character> <character>
<numeric>
GSM2545336  GSM2545336 Mus musculus      8 Female InfluenzaA C57
BL/6          8
GSM2545337  GSM2545337 Mus musculus      8 Female NonInfected C57
BL/6          0
GSM2545338  GSM2545338 Mus musculus      8 Female NonInfected C57
BL/6          0
      tissue   mouse
<character> <numeric>
GSM2545336  Cerebellum      14
GSM2545337  Cerebellum      9
GSM2545338  Cerebellum      10
rowData(sel)
DataFrame with 5 rows and 9 columns
      gene ENTREZID      product ensemble_gene_id external_synonym
<character> <numeric>      <character>      <character>
Asl          Asl      109900 argininosuccinate ly.. ENSMUSG00000025533 25
10006M18Rik
Apod         Apod     11815 apolipoprotein D, tr.. ENSMUSG00000022548
NA
Cyp2d22     Cyp2d22    56448 cytochrome P450, fam.. ENSMUSG00000061740
2D22
Klk6         Klk6     19144 kallikrein related-p.. ENSMUSG00000050063
Bssp
Fcrls        Fcrls    80891 Fc receptor-like S, .. ENSMUSG00000015852 28
10439C17Rik
      chromosome_name  gene_biotype phenotype_description hsapiens_homolog_associated_gene_name
<character> <character> <character>
<character>
Asl          5 protein_coding abnormal circulating..
ASL
Apod         16 protein_coding abnormal lipid homeo..
APOD
Cyp2d22     15 protein_coding abnormal skin morpho..
CYP2D6
Klk6         7 protein_coding abnormal cytokine le..
KLK6
Fcrls        3 protein_coding decreased CD8-positi..
FCRL2

```

We can also use the `colData()` function to subset on something from the sample metadata, or the `rowData()` to subset on something from the feature metadata. For example, here we keep only miRNAs and the non infected samples:

[Hide](#)

```

sel <- se[rowData(se)$gene_biotype == "miRNA",
           colData(se)$infection == "NonInfected"]
sel
class: SummarizedExperiment
dim: 7 7
metadata(0):
assays(1): ''
rownames(7): Mir1901 Mir378a ... Mir128-1 Mir7682
rowData names(9): gene ENTREZID ... phenotype_description
  hsapiens_homolog_associated_gene_name
colnames(7): GSM2545337 GSM2545338 ... GSM2545353 GSM2545354
colData names(9): sample organism ... tissue mouse
assay(sel)
  GSM2545337 GSM2545338 GSM2545343 GSM2545348 GSM2545349 GSM2545353 GS
M2545354
Mir1901      45       44      74      55      68      33
60
Mir378a      11        7       9       4      12       4
8
Mir133b      4         6       5       4       6       7
3
Mir30c-2     10        6      16      12       8      17
15
Mir149       1         2       0       0       0       0
2
Mir128-1     4         1       2       2       1       2
1
Mir7682      2         0       4       1       3       5
5
colData(sel)
DataFrame with 7 rows and 9 columns
  sample   organism   age   sex   infection   st
  train    time
  <character> <character> <numeric> <character> <character> <charac-
ter> <numeric>
GSM2545337  GSM2545337 Mus musculus      8   Female NonInfected   C57
BL/6          0
GSM2545338  GSM2545338 Mus musculus      8   Female NonInfected   C57
BL/6          0
GSM2545343  GSM2545343 Mus musculus      8   Male   NonInfected   C57
BL/6          0
GSM2545348  GSM2545348 Mus musculus      8   Female NonInfected   C57
BL/6          0
GSM2545349  GSM2545349 Mus musculus      8   Male   NonInfected   C57
BL/6          0
GSM2545353  GSM2545353 Mus musculus      8   Female NonInfected   C57
BL/6          0
GSM2545354  GSM2545354 Mus musculus      8   Male   NonInfected   C57

```

```

BL/6      0
          tissue     mouse
          <character> <numeric>
GSM2545337 Cerebellum      9
GSM2545338 Cerebellum     10
GSM2545343 Cerebellum     11
GSM2545348 Cerebellum      8
GSM2545349 Cerebellum      7
GSM2545353 Cerebellum      4
GSM2545354 Cerebellum      2
rowData(sel)
DataFrame with 7 rows and 9 columns
          gene  ENTREZID       product  ensembl_gene_id external_syn
onym chromosome_name
          <character> <numeric>       <character>       <character>       <charac
ter>       <character>
Mir1901    Mir1901  100316686  microRNA  1901  ENSMUSG00000084565      Mirn
1901        18
Mir378a    Mir378a   723889   microRNA  378a  ENSMUSG00000105200      Mir
n378        18
Mir133b    Mir133b   723817   microRNA  133b  ENSMUSG00000065480      mir
133b        1
Mir30c-2   Mir30c-2   723964   microRNA  30c-2  ENSMUSG00000065567      mir 3
0c-2         1
Mir149     Mir149    387167   microRNA  149   ENSMUSG00000065470      Mir
n149        1
Mir128-1   Mir128-1   387147   microRNA  128-1 ENSMUSG00000065520      Mir
n128        1
Mir7682    Mir7682   102466847  microRNA  7682  ENSMUSG00000106406      mmu-mir-
7682        1
          gene_biotype  phenotype_description hsapiens_homolog_associated_gene
_name
          <character>       <character>       <chara
cter>
Mir1901    miRNA           NA
NA
Mir378a    miRNA  abnormal mitochondri..      MI
R378A
Mir133b    miRNA  no abnormal phenotyp..
R133B
Mir30c-2   miRNA           NA
R30C2
Mir149     miRNA  increased circulatin..
NA
Mir128-1   miRNA  no abnormal phenotyp..
128-1
Mir7682    miRNA           NA
NA

```

For the following exercise, you should download the SE.rda object (that contains the `se` object), and open the file using the ‘load()’ function.

[Hide](#)

```
download.file(url = "https://raw.githubusercontent.com/UCLouvain-CBIO/bioinfo-training-01-intro-r/master/data/SE.rda",
              destfile = "data/SE.rda")
load(file = "data/SE.rda")
```

7.3 Exercise session 6

Extract the gene expression levels of the 3 first genes in samples at time 0 and at time 8.

7.3.1 Exercise session 6 - Solutions

► Details

7.3.1.1 Adding variables to metadata

We can also add information to the metadata. Suppose that you want to add the center where the samples were collected...

[Hide](#)

```

colData(se)$center <- rep("University of Illinois", nrow(colData(se)))
colData(se)
DataFrame with 22 rows and 10 columns
      sample organism   age   sex infection st
rain    time
<character> <character> <numeric> <character> <character> <character> <numeric>
GSM2545336  GSM2545336 Mus musculus     8 Female InfluenzaA C57
BL/6       8
GSM2545337  GSM2545337 Mus musculus     8 Female NonInfected C57
BL/6       0
GSM2545338  GSM2545338 Mus musculus     8 Female NonInfected C57
BL/6       0
GSM2545339  GSM2545339 Mus musculus     8 Female InfluenzaA C57
BL/6       4
GSM2545340  GSM2545340 Mus musculus     8 Male  InfluenzaA C57
BL/6       4
...
...
GSM2545353  GSM2545353 Mus musculus     8 Female NonInfected C57
BL/6       0
GSM2545354  GSM2545354 Mus musculus     8 Male  NonInfected C57
BL/6       0
GSM2545362  GSM2545362 Mus musculus     8 Female InfluenzaA C57
BL/6       4
GSM2545363  GSM2545363 Mus musculus     8 Male  InfluenzaA C57
BL/6       4
GSM2545380  GSM2545380 Mus musculus     8 Female InfluenzaA C57
BL/6       8
      tissue mouse          center
<character> <numeric> <character>
GSM2545336  Cerebellum    14 University of Illinois
GSM2545337  Cerebellum    9 University of Illinois
GSM2545338  Cerebellum    10 University of Illinois
GSM2545339  Cerebellum    15 University of Illinois
GSM2545340  Cerebellum    18 University of Illinois
...
...
GSM2545353  Cerebellum    4 University of Illinois
GSM2545354  Cerebellum    2 University of Illinois
GSM2545362  Cerebellum    20 University of Illinois
GSM2545363  Cerebellum   12 University of Illinois
GSM2545380  Cerebellum   19 University of Illinois

```

This illustrates that the metadata slots can grow indefinitely without affecting the other structures!

Take-home message

- `SummarizedExperiment` represent an efficient way to store and to handle omics data.

- They are used in many Bioconductor packages.

If you follow next training focused on RNA sequencing analysis, you will learn to use the Bioconductor `DESeq2` package to do some differential expression analyses. `DESeq2`'s whole analysis is handled in a `SummarizedExperiment`.

8 Useful material

Books

- R in a nutshell, R cookbook, R graphics cookbook (@O'Reilly media)
- A Beginner's Guide to R (Zuur, Ieno, Meesters, @Springer)
- R Programming for Data Science (Peng, @Leanpub)
- R Programming for Bioinformatics (Gentleman, @CRC)
- Bioconductor Case Studies (@Springer)
- Data Analysis for the Life Sciences (Irizarry, Love, @Leanpub)
- Bioconductor - An Introduction to Core Technologies (Hansen, @Leanpub)
- R for Data Science (Wickham, Grolemund, @O'Reilly)
- the whole `Use R!` book series: <https://link.springer.com/bookseries/6991>
(<https://link.springer.com/bookseries/6991>)
- this one from CRC: <https://www.crcpress.com/Chapman--HallCRC-The-R-Series/book-series/CRCTHERSER> (<https://www.crcpress.com/Chapman--HallCRC-The-R-Series/book-series/CRCTHERSER>)
- <https://link.springer.com/book/10.1007/978-3-662-53670-4>
(<https://link.springer.com/book/10.1007/978-3-662-53670-4>)
- <https://link.springer.com/book/10.1007/978-3-662-49102-7>
(<https://link.springer.com/book/10.1007/978-3-662-49102-7>)

Courses

- <https://www.datacamp.com/courses/free-introduction-to-r>
(<https://www.datacamp.com/courses/free-introduction-to-r>)
- <https://www.coursera.org/specializations/jhu-data-science>
(<https://www.coursera.org/specializations/jhu-data-science>)
- <https://www.edx.org/course/introduction-r-data-science-microsoft-dat204x-7>
(<https://www.edx.org/course/introduction-r-data-science-microsoft-dat204x-7>)

Misc

- <http://r4stats.com/articles/why-r-is-hard-to-learn/> (<http://r4stats.com/articles/why-r-is-hard-to-learn/>)
- <https://www.rstudio.com/resources/cheatsheets/> (<https://www.rstudio.com/resources/cheatsheets/>)
- swirl - learn R in R: <http://swirlstats.com/> (<http://swirlstats.com/>)

Session Info

[Hide](#)

```

sessionInfo()
R version 4.3.0 (2023-04-21)
Platform: x86_64-apple-darwin20 (64-bit)
Running under: macOS Monterey 12.6.4

Matrix products: default
BLAS:    /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/libBLAS.dylib
LAPACK:  /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib; LAPACK version 3.11.0

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats4      stats       graphics   grDevices  utils       datasets   methods    base

other attached packages:
[1] clusterProfiler_4.8.1      topGO_2.52.0          SparseM_1.81
[4] GO.db_3.17.0              graph_1.78.0           GeneTonic_2.4.0
[7] iSEEU_1.12.0              iSEEHex_1.2.0          iSEE_2.12.0
[10] SingleCellExperiment_1.22.0 pheatmap_1.0.12        apeglm_1.22.1
[13] pcaExplorer_2.26.1        bigmemory_4.6.1         edgeR_3.42.4
[16] limma_3.56.2              ExploreModelMatrix_1.12.0 GenomicFeatures_
1.52.1
[19] org.Hs.eg.db_3.17.0       AnnotationDbi_1.62.1  DESeq2_1.40.2
[22] tximeta_1.18.0            macrophage_1.16.0      knitr_1.43
[25] BiocStyle_2.28.0          SummarizedExperiment_1.30.2 Biobase_2.60.0
[28] GenomicRanges_1.52.0       GenomeInfoDb_1.36.1    IRanges_2.34.1
[31] S4Vectors_0.38.1          BiocGenerics_0.46.0    MatrixGenerics_
1.12.2
[34] matrixStats_1.0.0          lubridate_1.9.2        forcats_1.0.0
[37] stringr_1.5.0              dplyr_1.1.2             purrr_1.0.1
[40] readr_2.1.4                tidyverse_2.0.0          tibble_3.2.1
[43] tidyverse_2.0.0            ggthemes_4.2.4          gapminder_1.0.0
[46] ggplot2_3.4.2              MASS_7.3-60             rmarkdown_2.22

loaded via a namespace (and not attached):
[1] GSEABase_1.62.0          vroom_1.6.3            progress_1.
2.2
[4] DT_0.28                  Biostrings_2.68.1       pagedown_0.
20
[7] vctrs_0.6.3              digest_0.6.32          png_0.1-8
[10] shape_1.4.6              shinyBS_0.61.1         registry_0.

```

```

5-1
[13] ggrepel_0.9.3           reshape2_1.4.4        qvalue_2.3
2.0
[16] httpuv_1.6.11          foreach_1.5.2       withr_2.5.0
[19] ggfun_0.1.1            xfun_0.39          ellipsis_0.
3.2
[22] survival_3.5-5        memoise_2.0.1      xaringanExt
ra_0.7.0
[25] hexbin_1.28.3         gson_0.1.0        tidytree_0.
4.2
[28] GlobalOptions_0.1.2    prettyunits_1.1.1   KEGGREST_1.
40.0
[31] promises_1.2.0.1       httr_1.4.6        downloader_
0.4
[34] restfulr_0.0.15       ps_1.7.5          rstudioapi_
0.14
[37] shinyAce_0.4.2         DOSE_3.26.1       miniUI_0.1.
1.1
[40] generics_0.1.3         archive_1.1.5     base64enc_
0.1-3
[43] processx_3.8.1        curl_5.0.1        zlibbioc_1.
46.0
[46] ggraph_2.1.0           polyclip_1.10-4   ca_0.71.1
[49] GenomeInfoDbData_1.2.10 RBGL_1.76.0      threejs_0.
3.3
[52] interactiveDisplayBase_1.38.0 xtable_1.8-4      doParallel_
1.0.17
[55] evaluate_0.21          S4Arrays_1.0.4     BiocFileCac
he_2.8.0
[58] hms_1.1.3              bookdown_0.34     colorspace_
2.1-0
[61] filelock_1.0.2         visNetwork_2.1.2   shinyWidget
s_0.7.6
[64] magrittr_2.0.3         Rgraphviz_2.44.0  ggtree_3.8.
0
[67] later_1.3.1            viridis_0.6.3     lattice_0.2
1-8
[70] NMF_0.26                genefilter_1.82.1 shadowtext_
0.1.2
[73] XML_3.99-0.14          cowplot_1.1.1     pillar_1.9.
0
[76] nlme_3.1-162            iterators_1.0.14  gridBase_0.
4-7
[79] compiler_4.3.0          stringi_1.7.12    shinycssloa
ders_1.0.0
[82] Category_2.66.0         TSP_1.2-4        dendextend_
1.17.1
[85] GenomicAlignments_1.36.0 plyr_1.8.8       crayon_1.5.
2

```

| | | |
|------------------------------|-------------------------|-------------|
| [88] BiocIO_1.10.0 | gridGraphics_0.5-1 | emdbook_1. |
| 3.12 | | |
| [91] locfit_1.5-9.8 | graphlayouts_1.0.0 | bit_4.0.5 |
| [94] chromote_0.1.1 | fastmatch_1.1-3 | codetools_ |
| 0.2-19 | | |
| [97] crosstalk_1.2.0 | bslib_0.5.0 | GetoptLong_ |
| 1.0.5 | | |
| [100] plotly_4.10.2 | mime_0.12 | splines_4. |
| 3.0 | | |
| [103] circlize_0.4.15 | Rcpp_1.0.10 | servr_0.27 |
| [106] dbplyr_2.3.2 | tippy_0.1.0 | HDO.db_0.9 |
| 9.1 | | |
| [109] blob_1.2.4 | utf8_1.2.3 | clue_0.3-64 |
| [112] BiocVersion_3.17.1 | AnnotationFilter_1.24.0 | fs_1.6.2 |
| [115] backbone_2.1.2 | expm_0.999-7 | ggplotify_ |
| 0.1.1 | | |
| [118] Matrix_1.5-4.1 | tzdb_0.4.0 | tweenr_2.0. |
| 2 | | |
| [121] pkgconfig_2.0.3 | tools_4.3.0 | cachem_1.0. |
| 8 | | |
| [124] RSQLite_2.3.1 | viridisLite_0.4.2 | DBI_1.1.3 |
| [127] numDeriv_2016.8-1.1 | fastmap_1.1.1 | scales_1.2. |
| 1 | | |
| [130] grid_4.3.0 | shinydashboard_0.7.2 | Rsamtools_ |
| 2.16.0 | | |
| [133] AnnotationHub_3.8.0 | sass_0.4.6 | patchwork_ |
| 1.1.2 | | |
| [136] coda_0.19-4 | BiocManager_1.30.21 | fontawesome |
| _0.5.1 | | |
| [139] farver_2.1.1 | scatterpie_0.2.1 | tidygraph_ |
| 1.2.3 | | |
| [142] mgcv_1.8-42 | yaml_2.3.7 | renderthis_ |
| 0.2.0 | | |
| [145] AnnotationForge_1.42.2 | rtracklayer_1.60.0 | cli_3.6.1 |
| [148] webshot_0.5.5 | lifecycle_1.0.3 | rsconnect_ |
| 0.8.29 | | |
| [151] mvtnorm_1.2-2 | xaringan_0.28 | tximport_1. |
| 28.0 | | |
| [154] rintrojs_0.3.2 | BiocParallel_1.34.2 | annotate_1. |
| 78.0 | | |
| [157] timechange_0.2.0 | gttable_0.3.3 | rjson_0.2.2 |
| 1 | | |
| [160] ggridges_0.5.4 | ape_5.7-1 | parallel_4. |
| 3.0 | | |
| [163] jsonlite_1.8.5 | colourpicker_1.2.0 | seriation_ |
| 1.4.2 | | |
| [166] bitops_1.0-7 | bigmemory.sri_0.1.6 | bit64_4.0.5 |
| [169] assertthat_0.2.1 | yulab.utils_0.0.6 | heatmaply_ |
| 1.4.2 | | |

```
[172] bs4Dash_2.3.0                      bdsmatrix_1.3-6          icons_0.2.0
[175] jquerylib_0.1.4                       highr_0.10              GOSemSim_2.
26.0
[178] shinyjs_2.1.0                         lazyeval_0.2.2           shiny_1.7.4
[181] dynamicTreeCut_1.63-1                 enrichplot_1.20.0        htmltools_
0.5.5
[184] rappdirs_0.3.3                        formatR_1.14             ensemblDb_
2.24.0
[187] glue_1.6.2                            XVector_0.40.0           RCurl_1.98-
1.12
[190] treeio_1.24.1                          ComplexUpset_1.3.3       gridExtra_
2.3
[193] igraph_1.5.0                           R6_2.5.1                labeling_0.
4.2
[196] cluster_2.1.4                          rngtools_1.5.2           bbmle_1.0.2
5
[199] aplot_0.1.10                          DelayedArray_0.26.3      tidyselect_
1.2.0
[202] vipor_0.4.5                           ProtGenerics_1.32.0      GOstats_2.6
6.0
[205] ggforce_0.4.1                         xml2_1.3.4               emo_0.0.0.9
000
[208] munsell_0.5.0                          data.table_1.14.8        websocket_
1.4.1
[211] fgsea_1.26.0                           htmlwidgets_1.6.2         ComplexHeat
map_2.16.0
[214] RColorBrewer_1.1-3                     biomaRt_2.56.1            rlang_1.1.1
[217] uuid_1.1-0                            fansi_1.0.4
```