Colors and themes in ggplot2

Data Wrangling and Husbandry

04/06/2020

Color basics

- For computers, colors are often described in terms of the RGB channels, but that does not match up well with how humans perceive colors

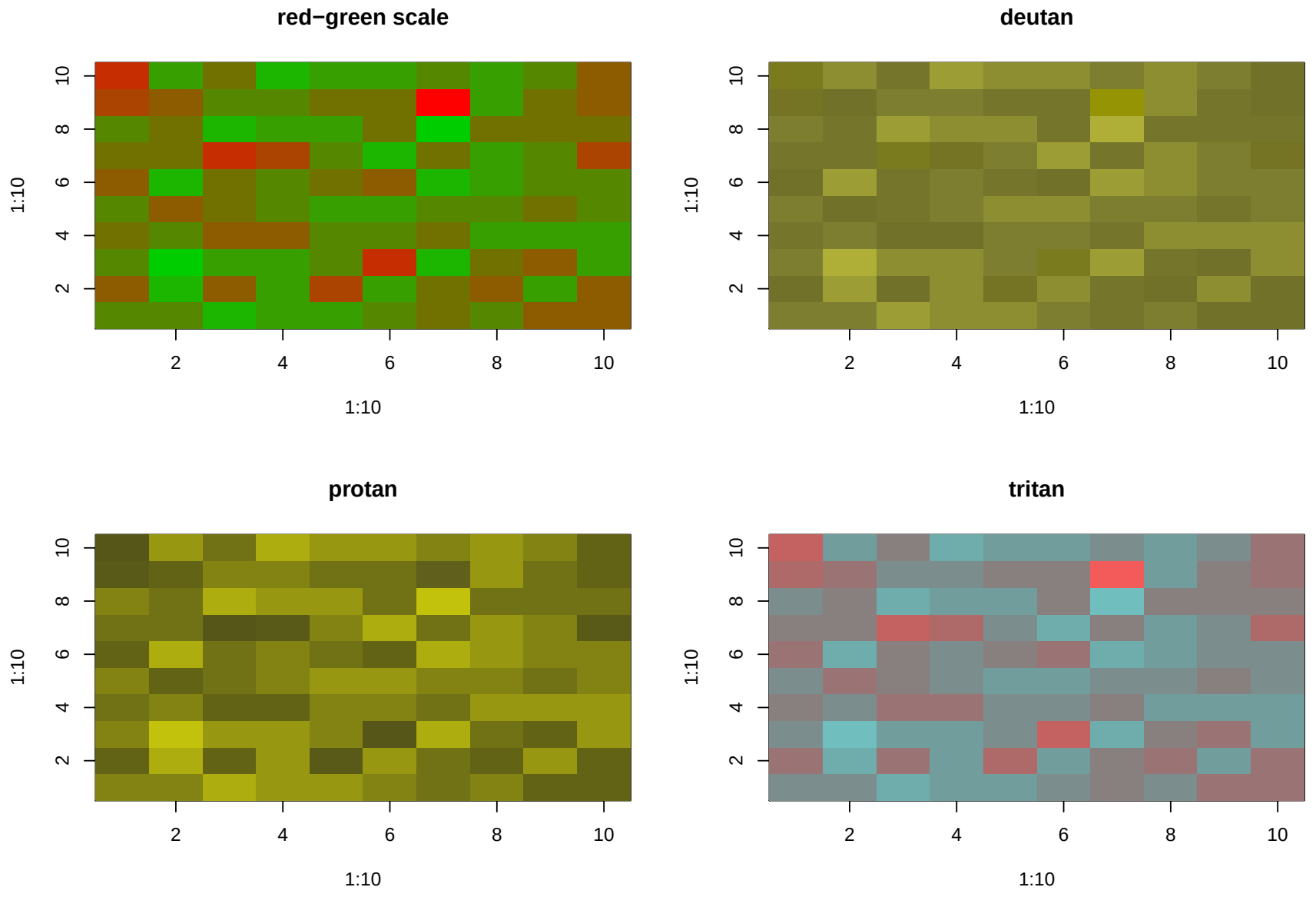
Example from http://www.hclwizard.org

- Surprisingly difficult to map colors to perceptually uniform space
- Various ways to map colors; here's a nice conceptual model of HCL space (H = hue, C = chroma, L = luminance)

There's a nice treatment of color theory at (http://www.handprint.com/HP/WCL/wcolor.html)

HCL space is quite complicated. The `colorspace` package can be useful for exploring the space if you are interested. Here are slices by constant luminance, with hue mapped to angle and chrome to radius.

- Roughly 5% of people (mostly, but not exclusively, men) have color vision deficiencies
- The site http://www.color-blindness.com/coblis/coblis.html can be used to simulate color deficiencies
- The package `dichromat` can also be used to simulate color deficiencies (and to suggest appropriate palettes).


red−green scale
deutan
protan
tritan
1:10
2
4
6
8
10

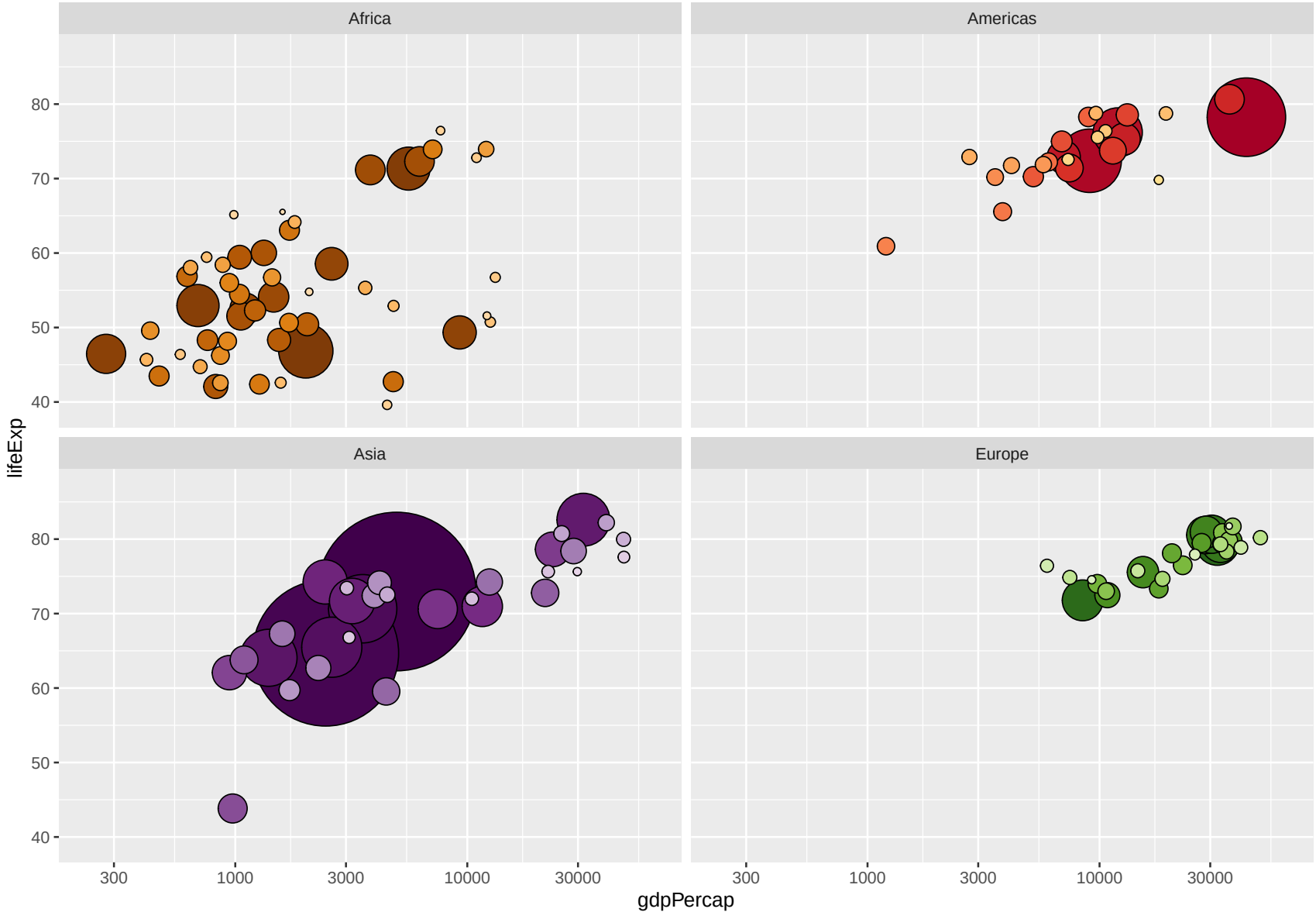
Picking colors manually in ggplot2

```
library(gapminder)
# from http://stat545.com/block019_enforce-color-scheme.htm
jdat <- gapminder %>%
  filter(continent != "Oceania") %>%
  droplevels() %>%
  mutate(country = reorder(country, desc(pop))) %>%
  arrange(year, country)

j_year <- 2007

jdat %>%
  filter(year == j_year) %>%
  ggplot(aes(x = gdpPercap, y = lifeExp, fill = country)) +
  scale_fill_manual(values = country_colors) +
  facet_wrap(~ continent) +
  geom_point(aes(size = pop), pch = 21, show.legend = FALSE
  scale_x_log10(limits = c(230, 63000)) +
  scale_size_continuous(range = c(1,40)) + ylim(c(39, 87))
```

Notce the use of `scale_fill_manual(values = country_colors)`;

```
library(gapminder)
country_colors[1:20]
```

```
##          Nigeria            Egypt         Ethiopia Congo
##        "#7F3B08"        "#833D07"        "#873F07"
##     South Africa            Sudan         Tanzania
##        "#8F4407"        "#934607"        "#974807"
##          Morocco          Algeria           Uganda
##        "#9F4D06"        "#A34F06"        "#A75106"
##       Mozambique       Madagascar    Cote d'Ivoire
##        "#AF5606"        "#B35806"        "#B75C07"
##     Burkina Faso           Malawi            Niger
##        "#BE6209"        "#C2650A"        "#C5690B"
```


Africa
Americas
Asia
Europe
lifeExp
gdpPercap
80
70
60
50
40
300
1000
3000
10000
30000

Other color palettes

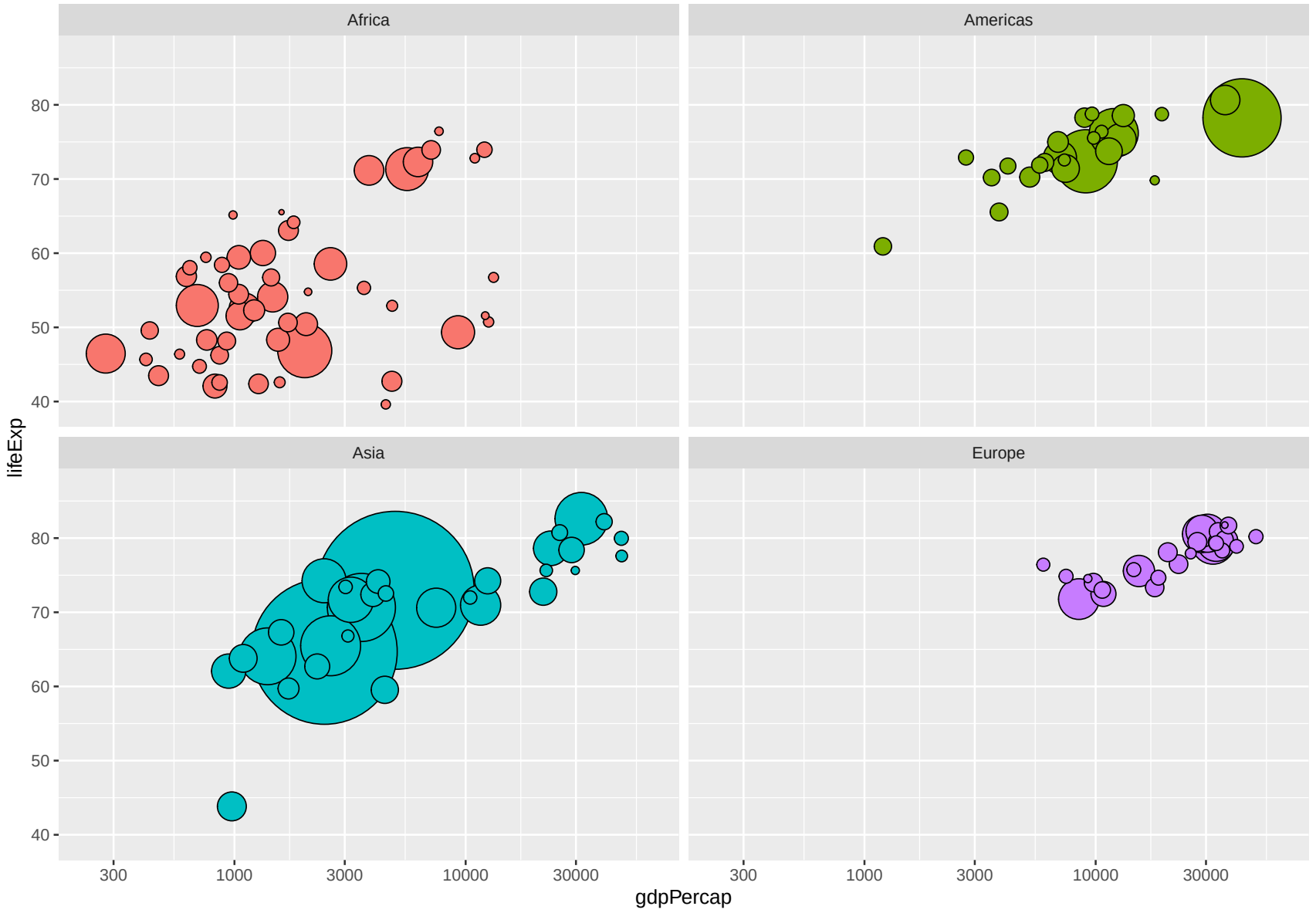
Except in unusual circumstances, I do not recommend picking specific colors, but instead to take advantage of existing color palettes. To illustrate, we'll fix the plot other than the colors (and color by continent, not country).

```
reference_plot <-
jdat %>%
  filter(year == j_year) %>%
  ggplot(aes(x = gdpPercap, y = lifeExp, fill = continent))
#  scale_fill_manual(values = continent_colors) +
  facet_wrap(~ continent) +
  geom_point(aes(size = pop), pch = 21, show.legend = FALSE
  scale_x_log10(limits = c(230, 63000)) +
  scale_size_continuous(range = c(1,40)) + ylim(c(39, 87))
```

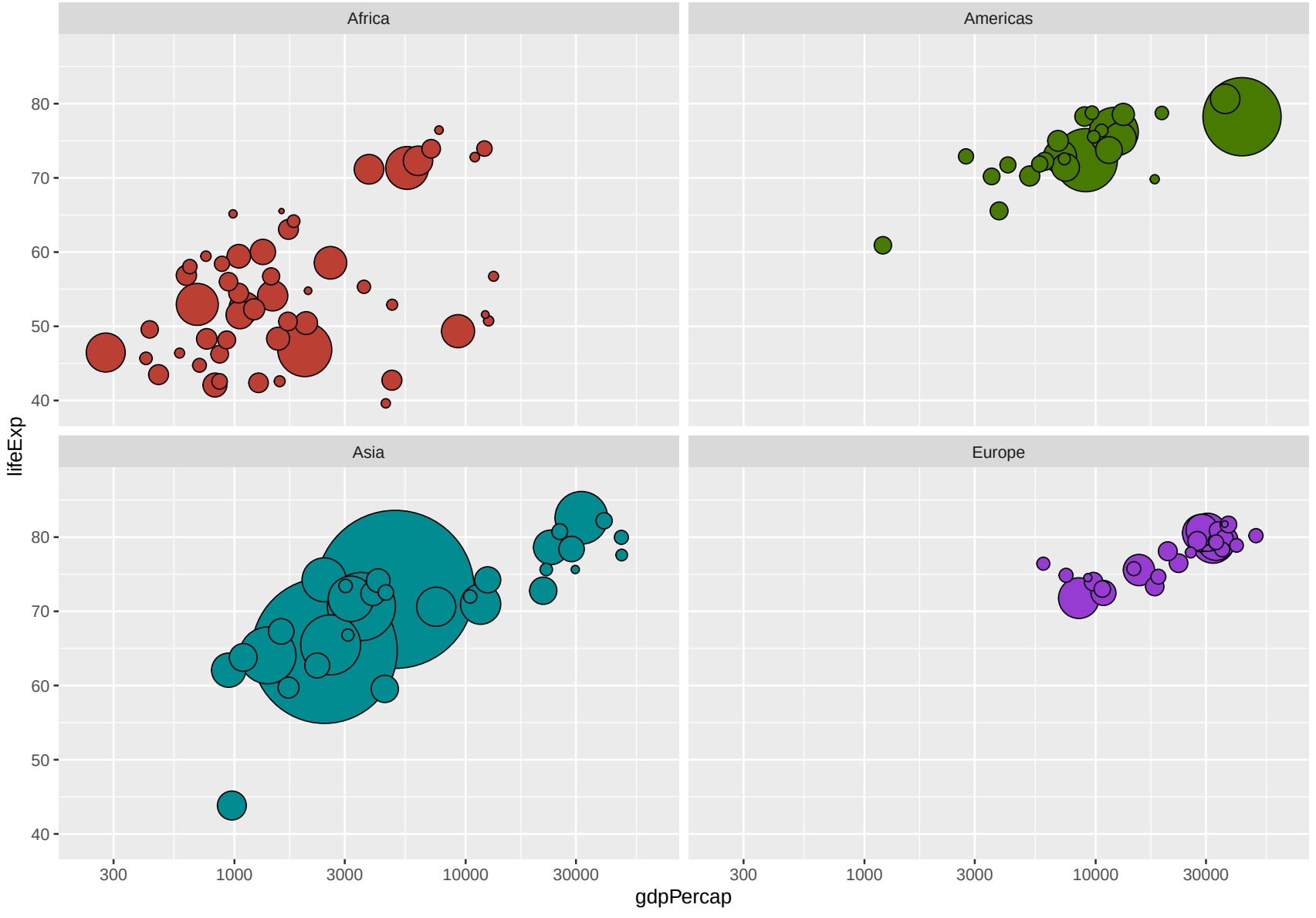
ggplot's default

- The default for ggplot is the function `scale_color_hue()`, which picks hues around the HCL wheel (with chroma and luminance fixed)

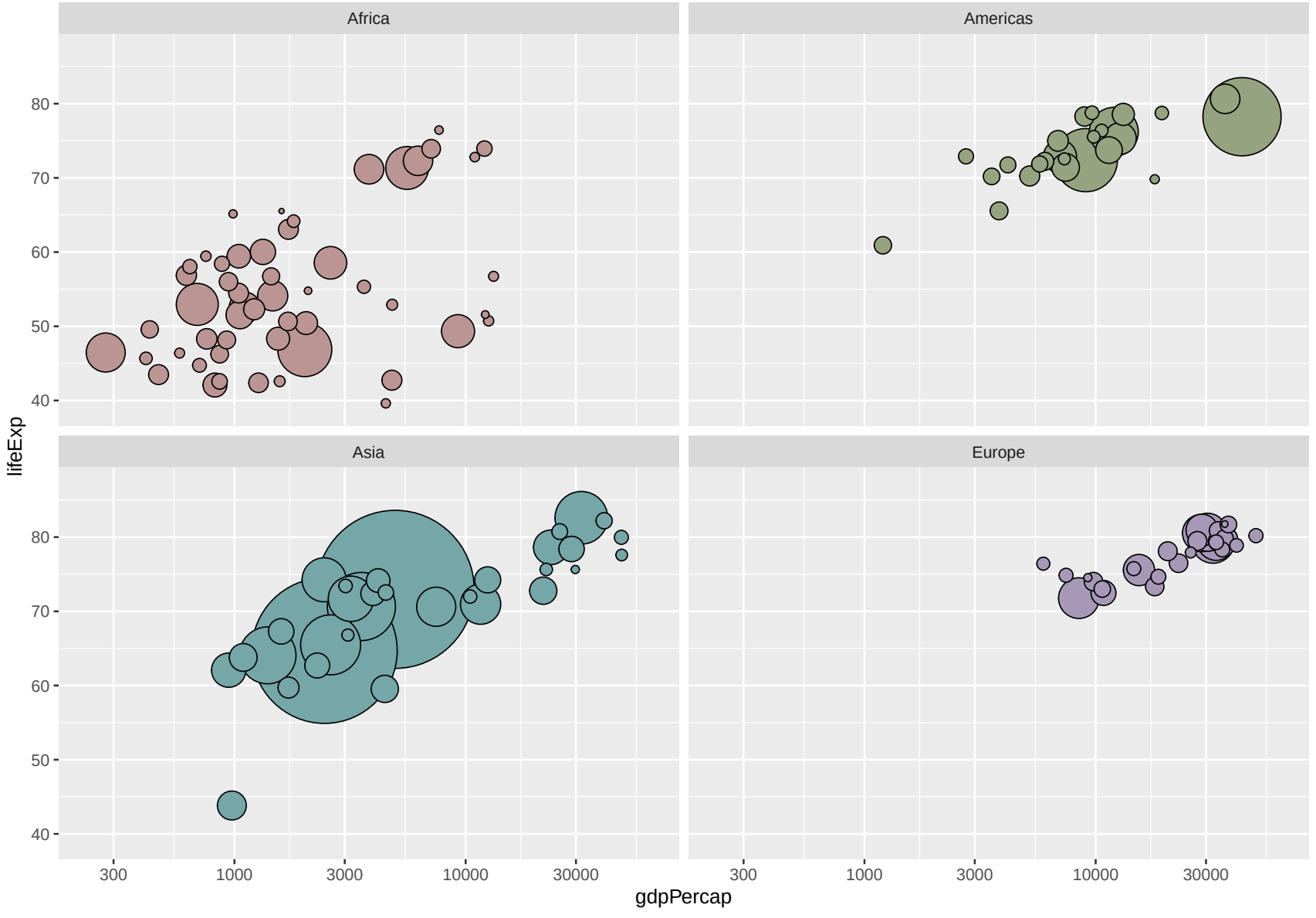
reference_plot



```
reference_plot + scale_fill_hue(l = 45)
```



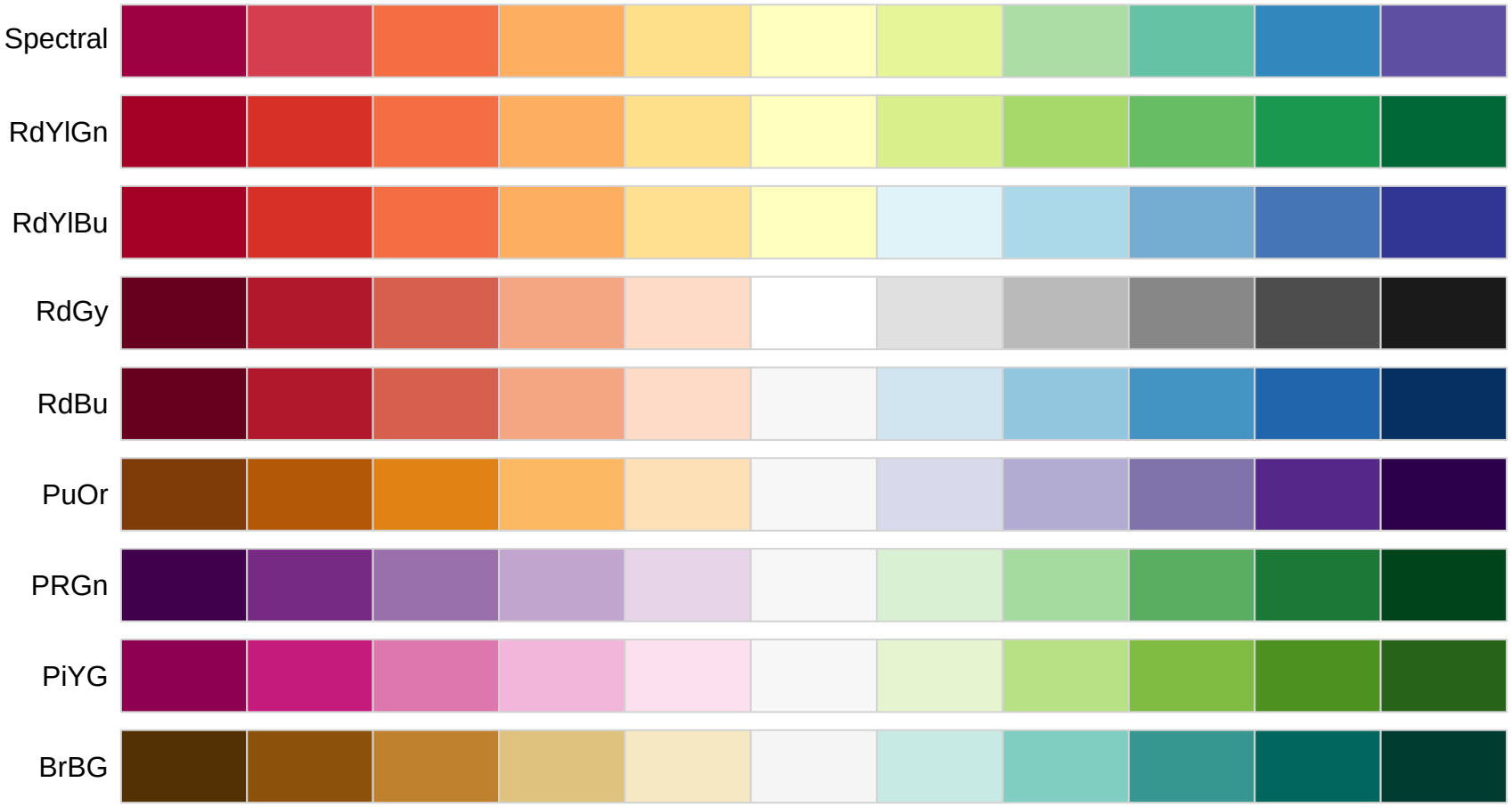
```
reference_plot + scale_fill_hue(c = 25)
```



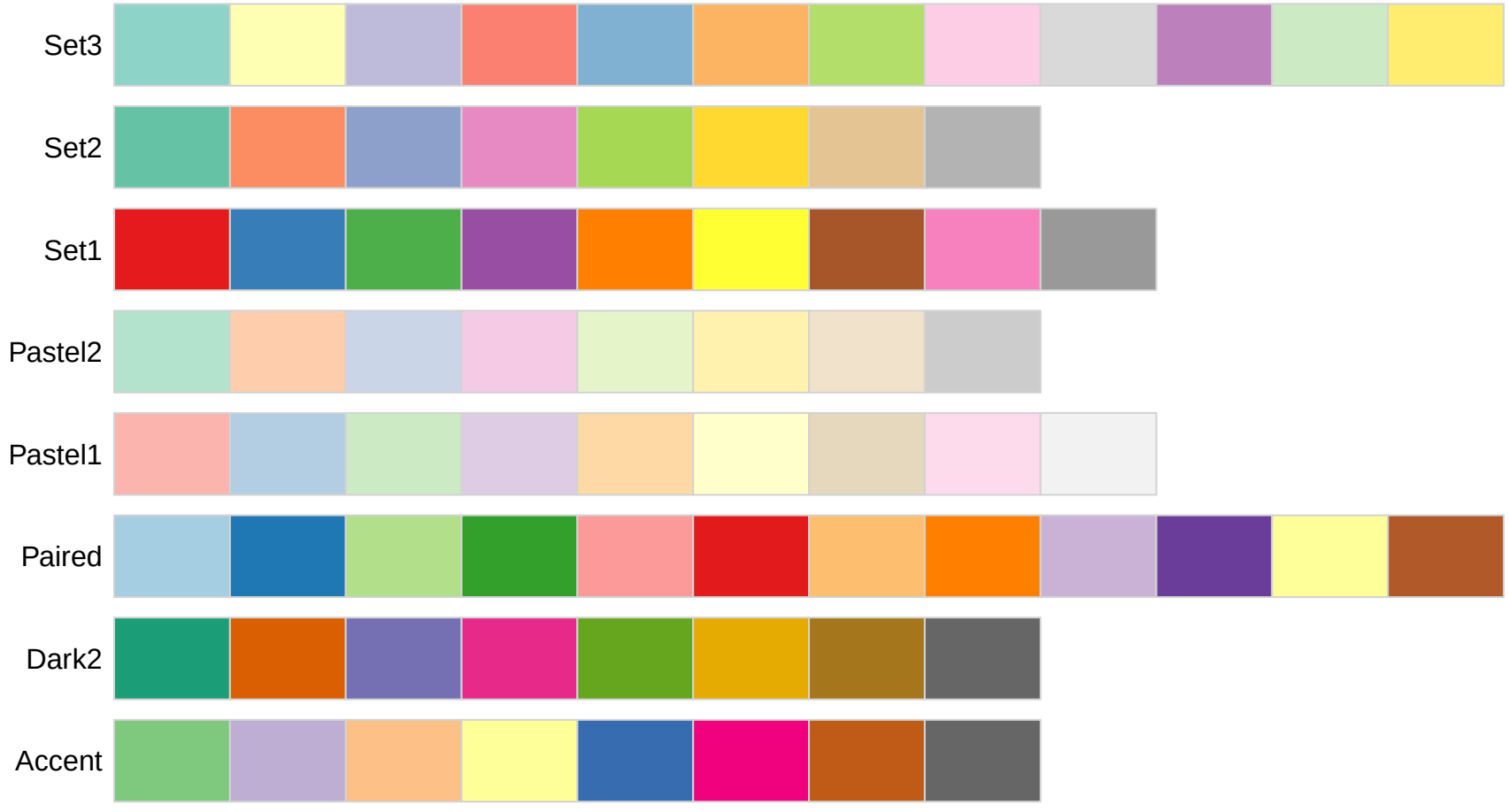
Color brewer

Colorbrewer, http://colorbrewer2.org, is an influential set of palettes originally selected for maps.

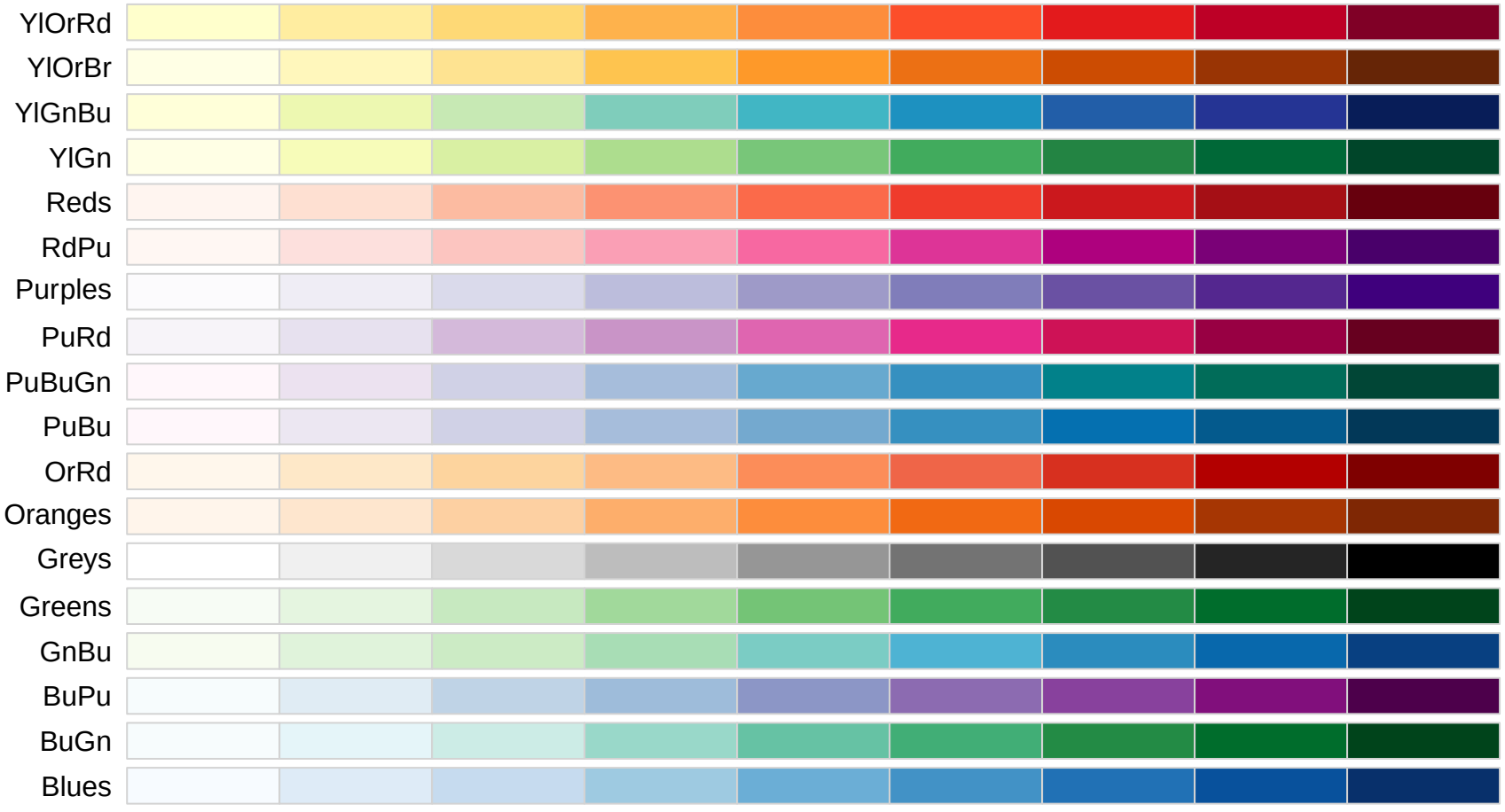
```
library(RColorBrewer)
display.brewer.all(type = "div")
```



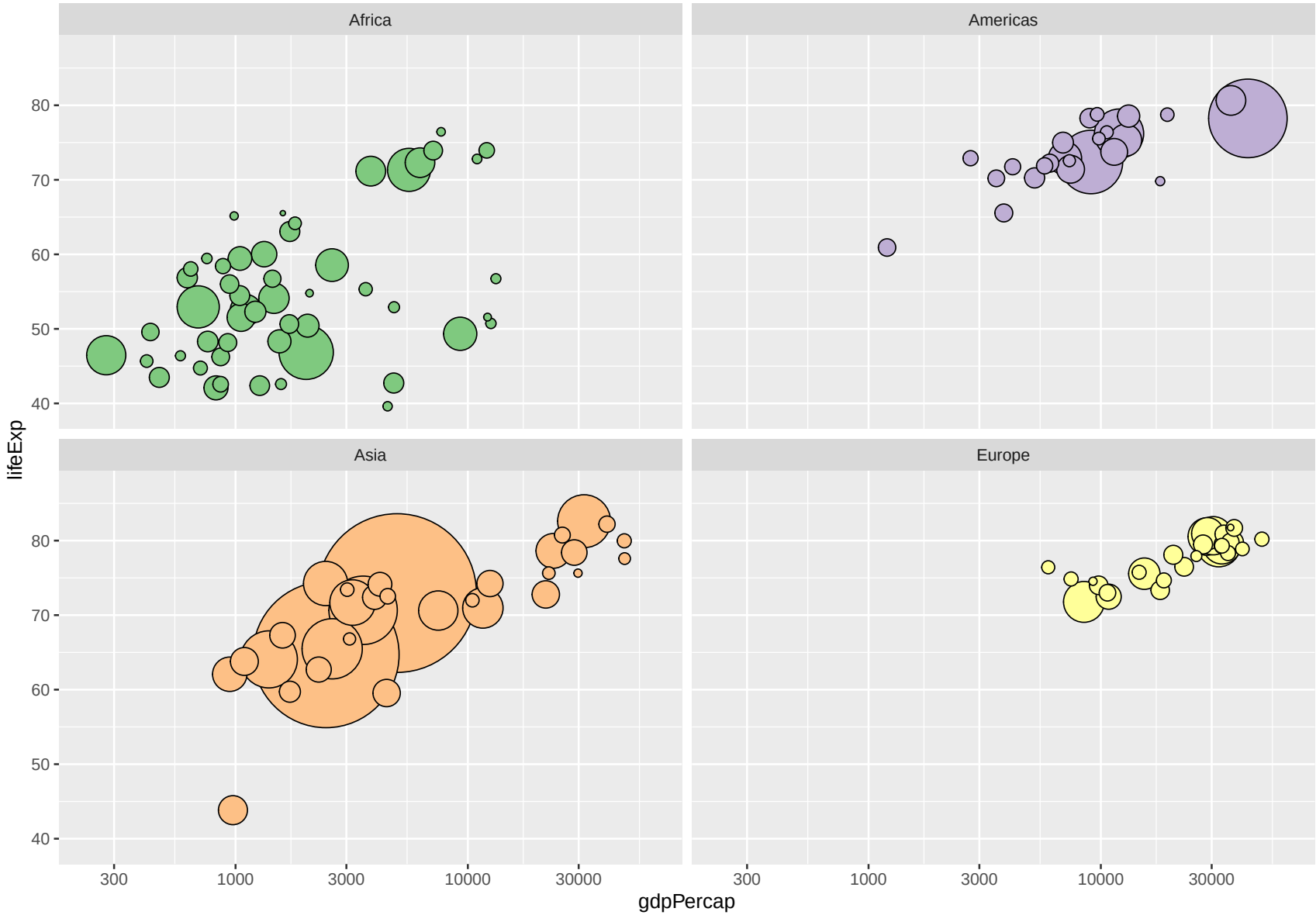
```
display.brewer.all(type = "qual")
```



```
display.brewer.all(type = "seq")
```



```
reference_plot + scale_fill_brewer(type = "qual", palette =
```



`scale_color_distiller()` applies the ColorBrewer color scales to continuous data

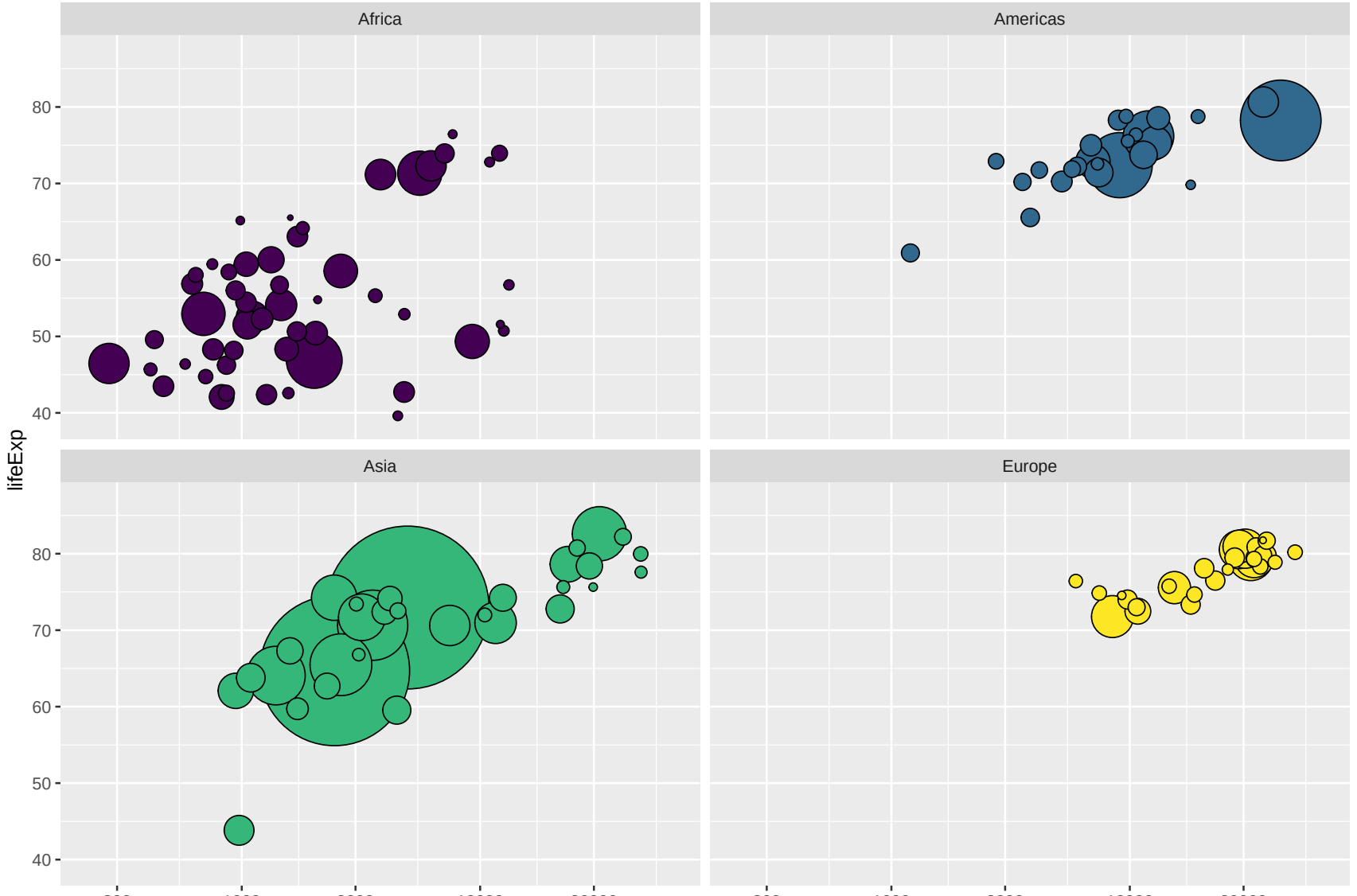
A newly popular palette comes via the `viridis` package, which takes an approach from MatLab.

> *"These color maps are designed in such a way that they will analytically be perfectly perceptually-uniform, both in regular form and also when converted to black-and-white. They are also designed to be perceived by readers with the most common form of color blindness."*"

```
library(viridis)
```

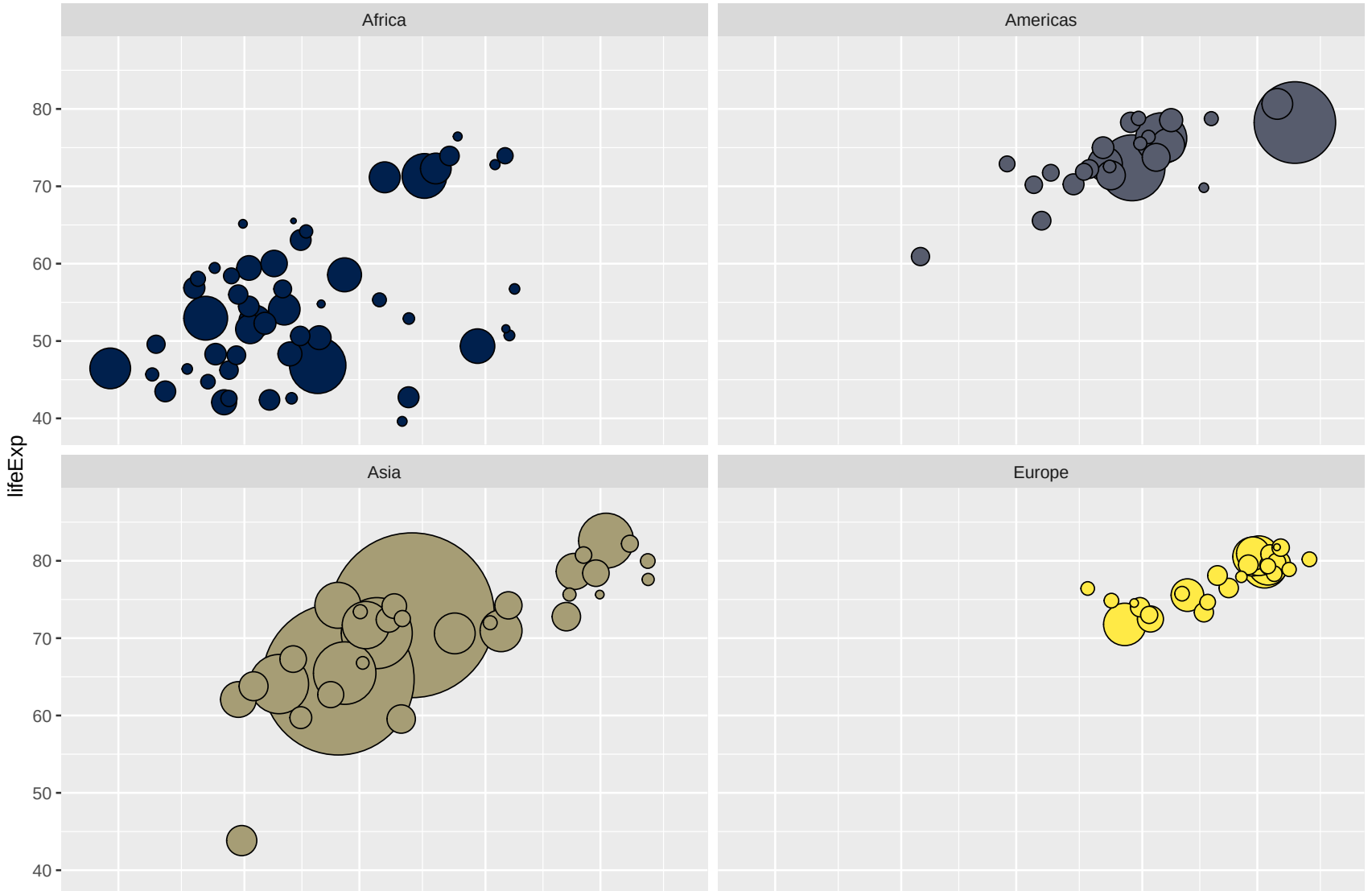
```
## Loading required package: viridisLite
```

```
reference_plot + scale_fill_viridis(discrete = TRUE)
```

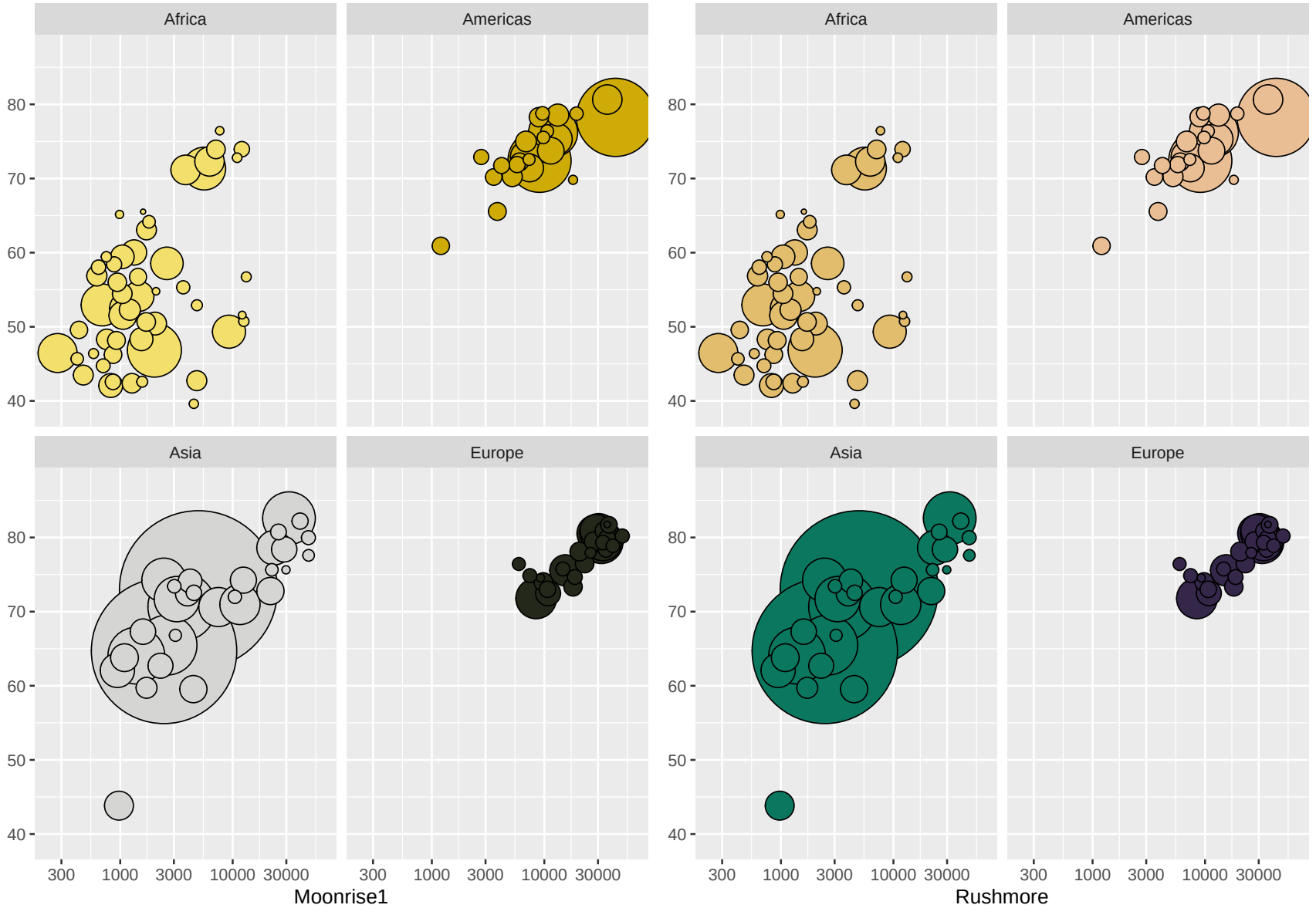


If you use `option = "E"` in `scale_fill_viridis()`, you'll get a palette suitable for those with color blindness.

```
library(viridis)
reference_plot + scale_fill_viridis(discrete = TRUE, optio
```



The Wes Anderson Palettes


Africa
Americas
Asia
Europe
80
70
60
50
40
300
1000
3000
10000
30000
Moonrise1
Rushmore

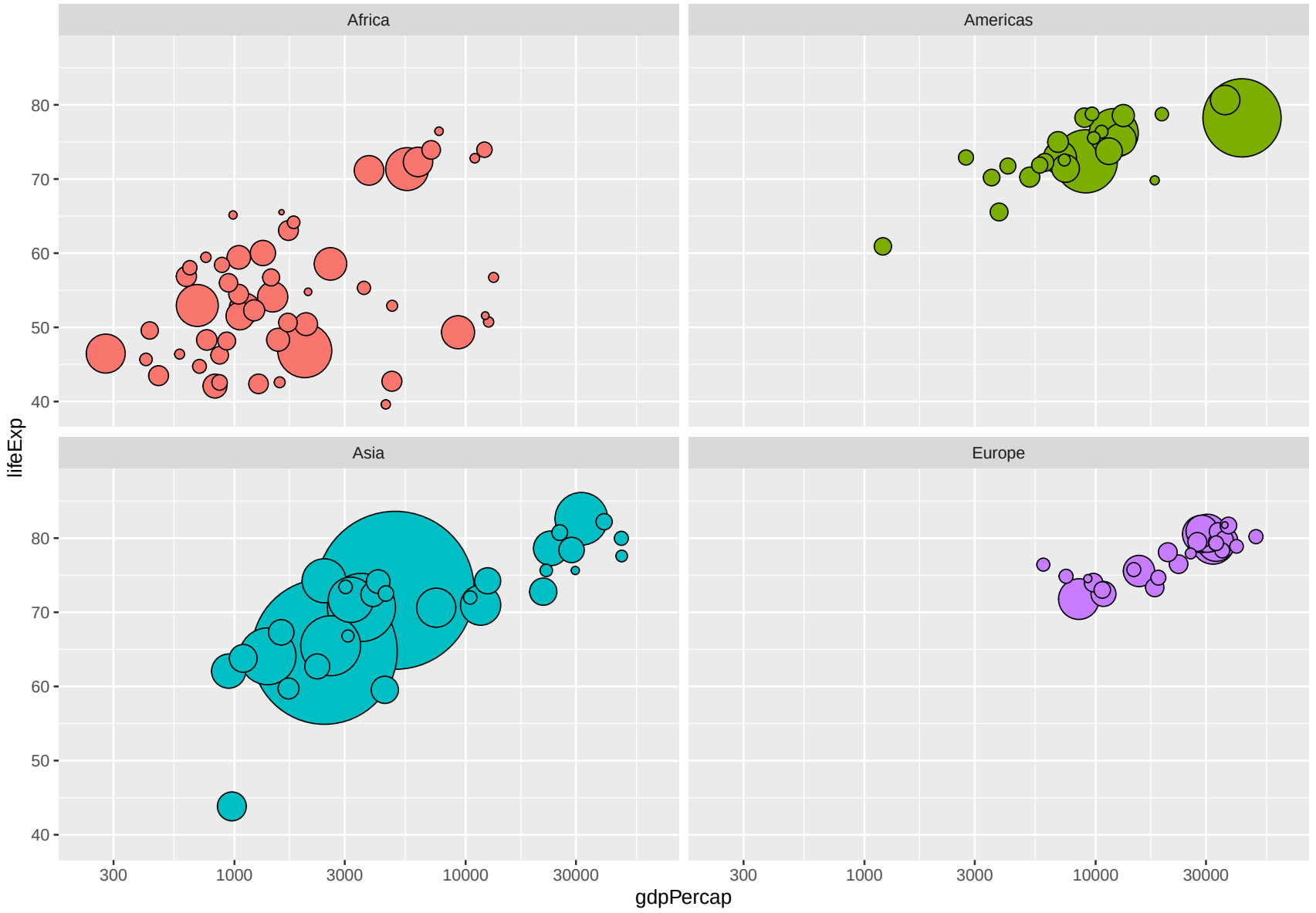
Note

For all of these, remember that to use `scale_color_*()`, not `scale_fill_*()` if you want to affect the color of lines and points.

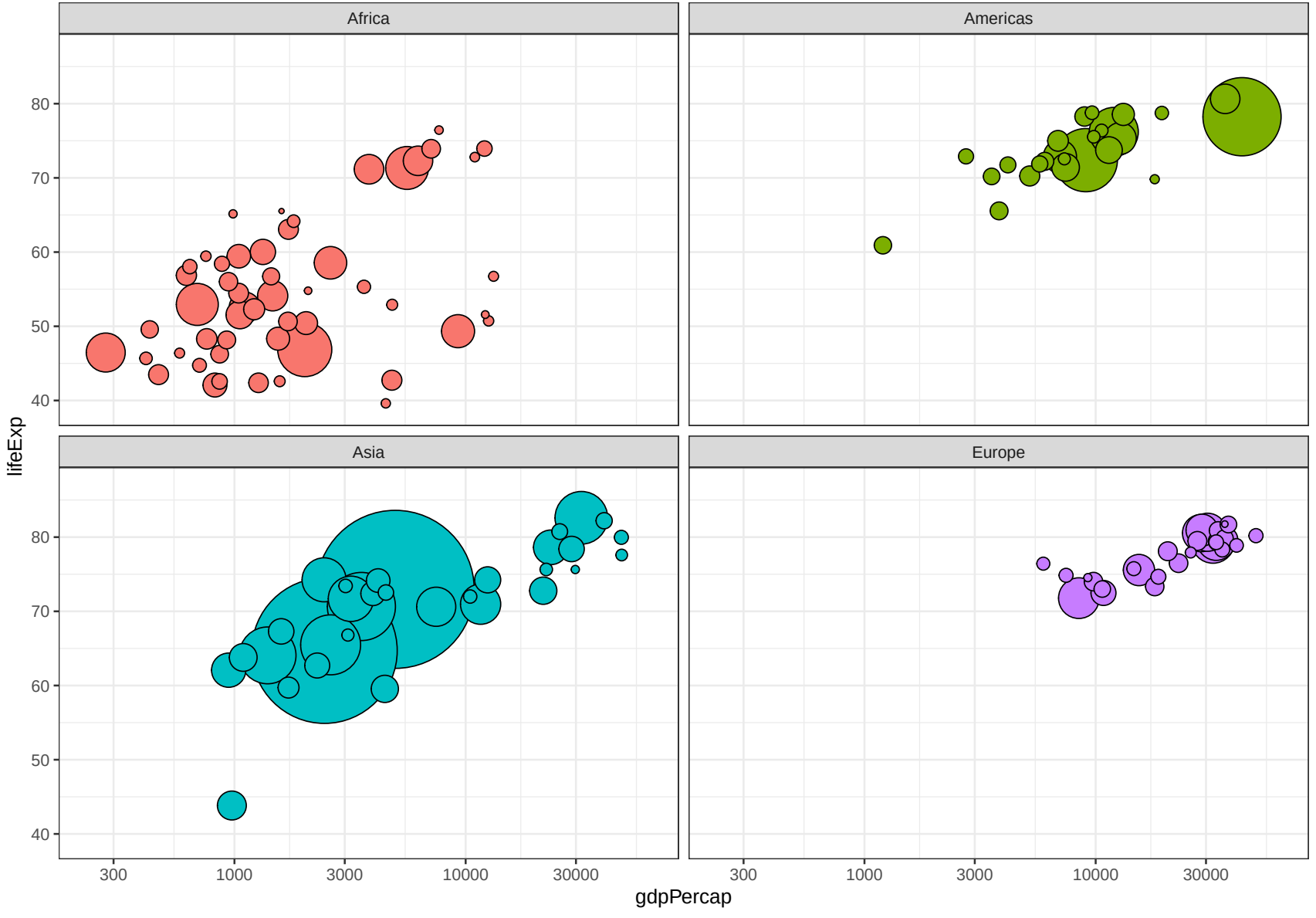
Themes

- Every plot has many non-data elements, such as background color, font, font size, and so on. You can adjust these individually with `theme()`, but I don't usually bother unless I'm tweaking something for publication. (See `ggThemeAssist` under the RStudio Addins menu for a really useful tool, however.)
- However, there are a number of complete themes, some part of the `ggplot2` package and some in the `ggthemes` package. The `ggthemes` package also include color palettes.
- You can examine the what the theme does with, for example `theme_linedraw`.

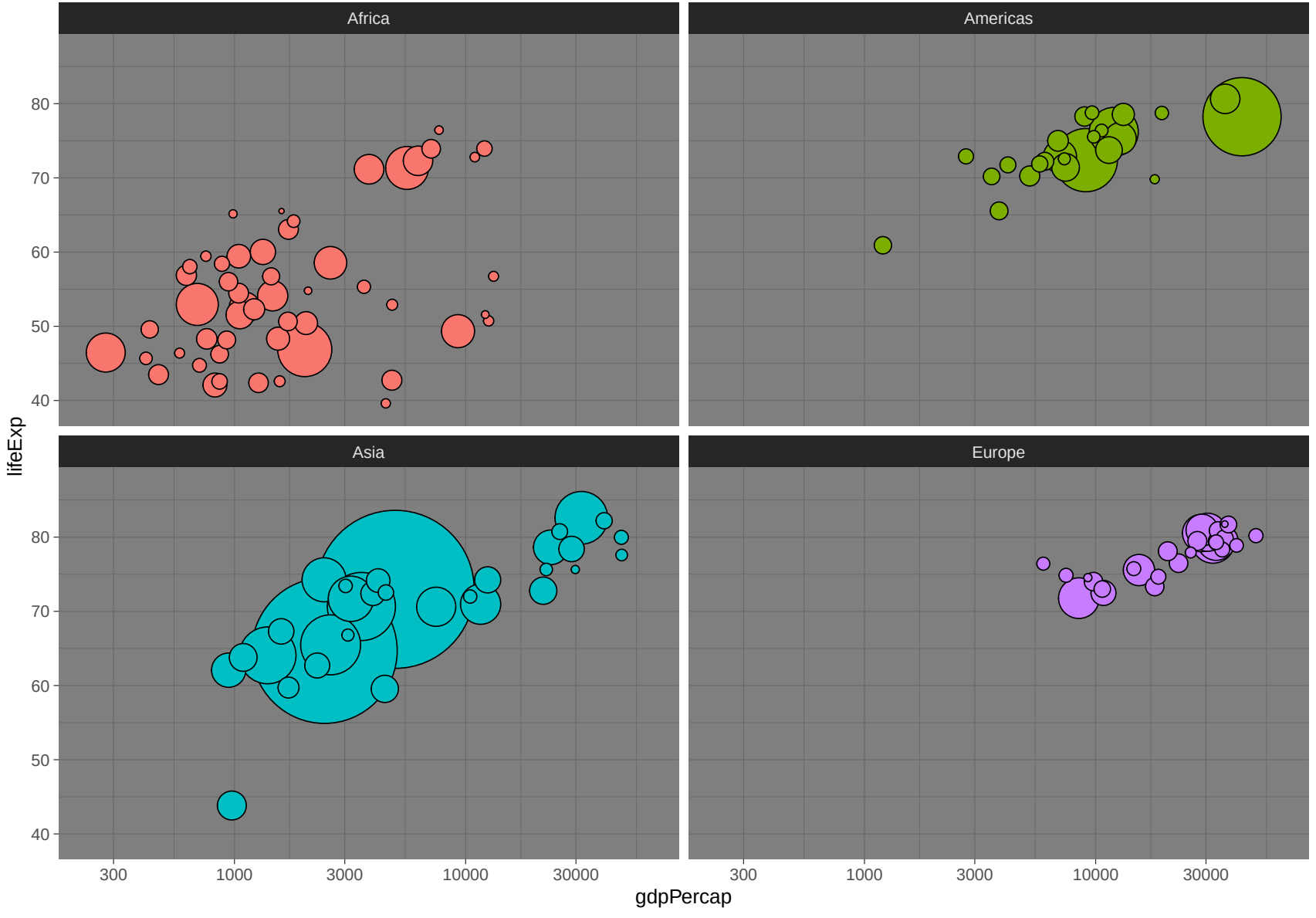
reference_plot



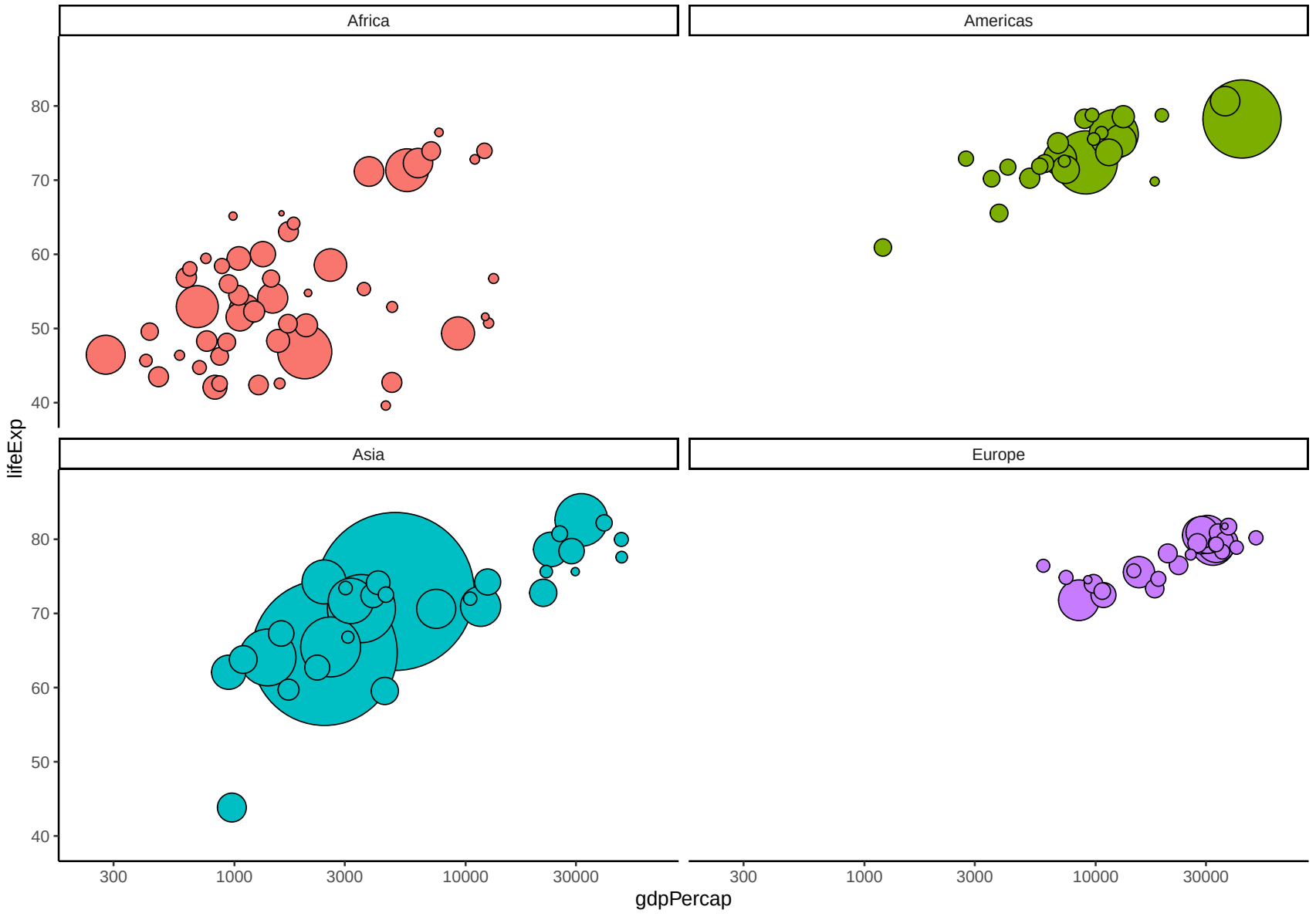
```
reference_plot + theme_bw()
```



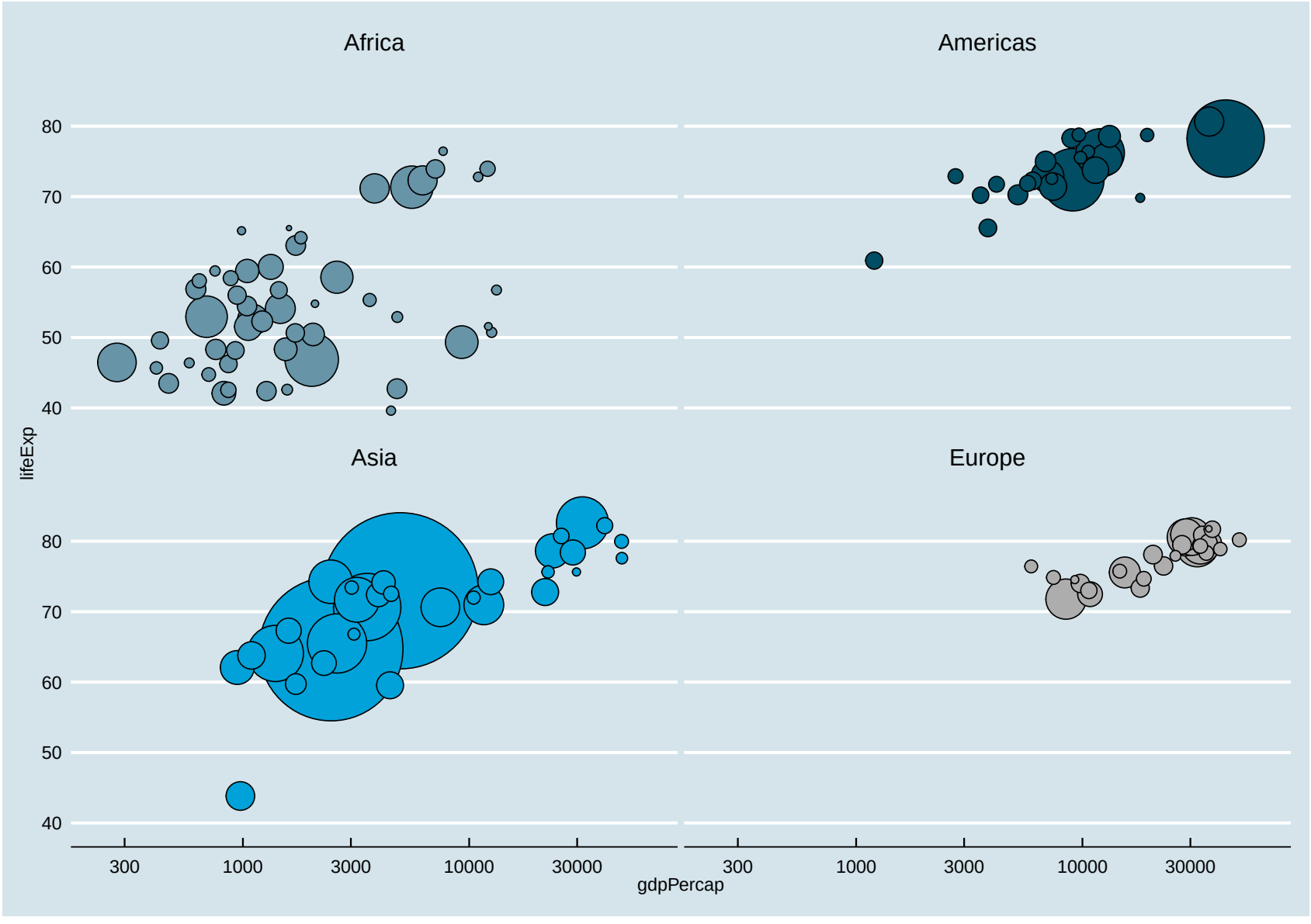
```
reference_plot + theme_dark()
```



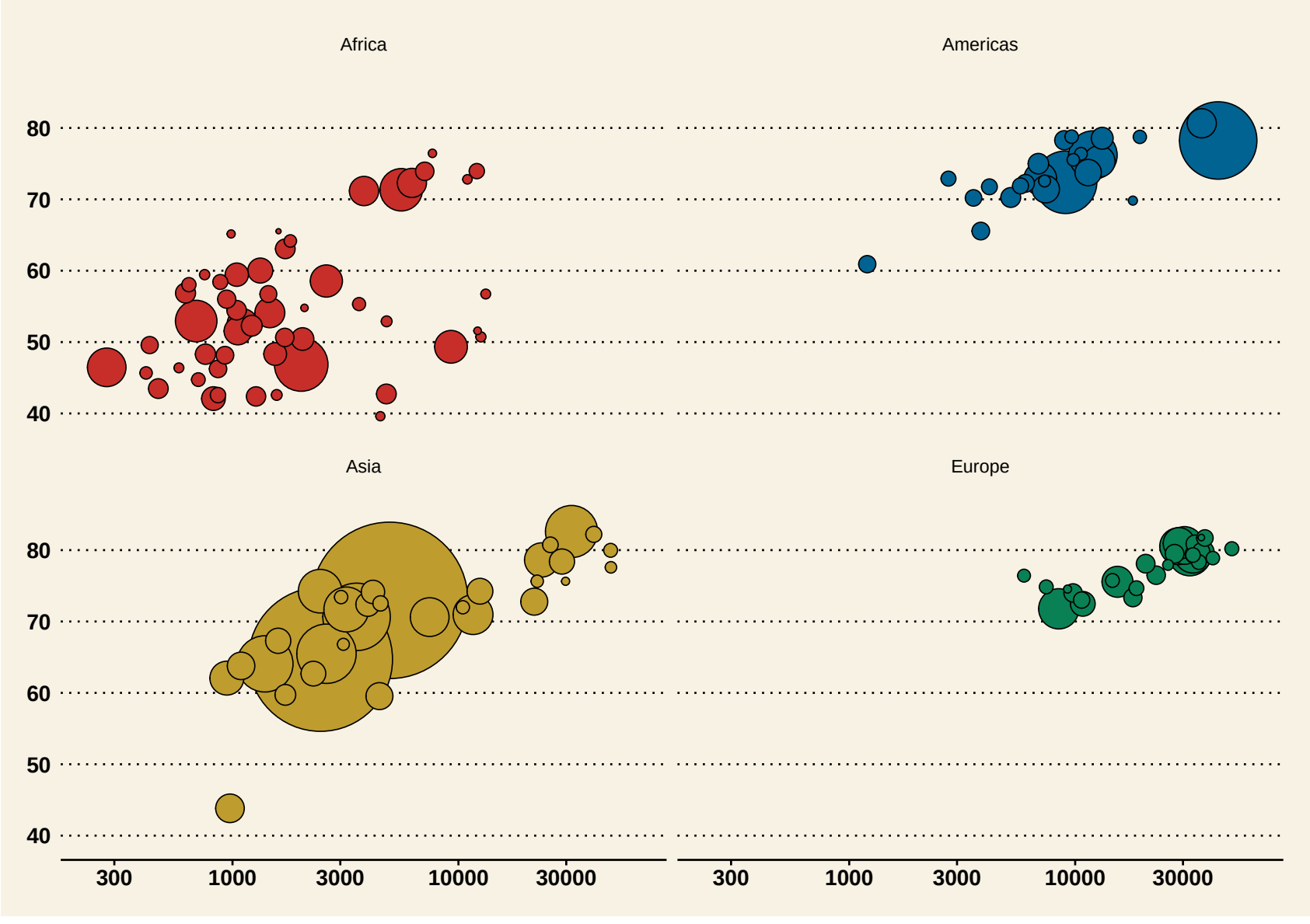
```
reference_plot + theme_classic()
```



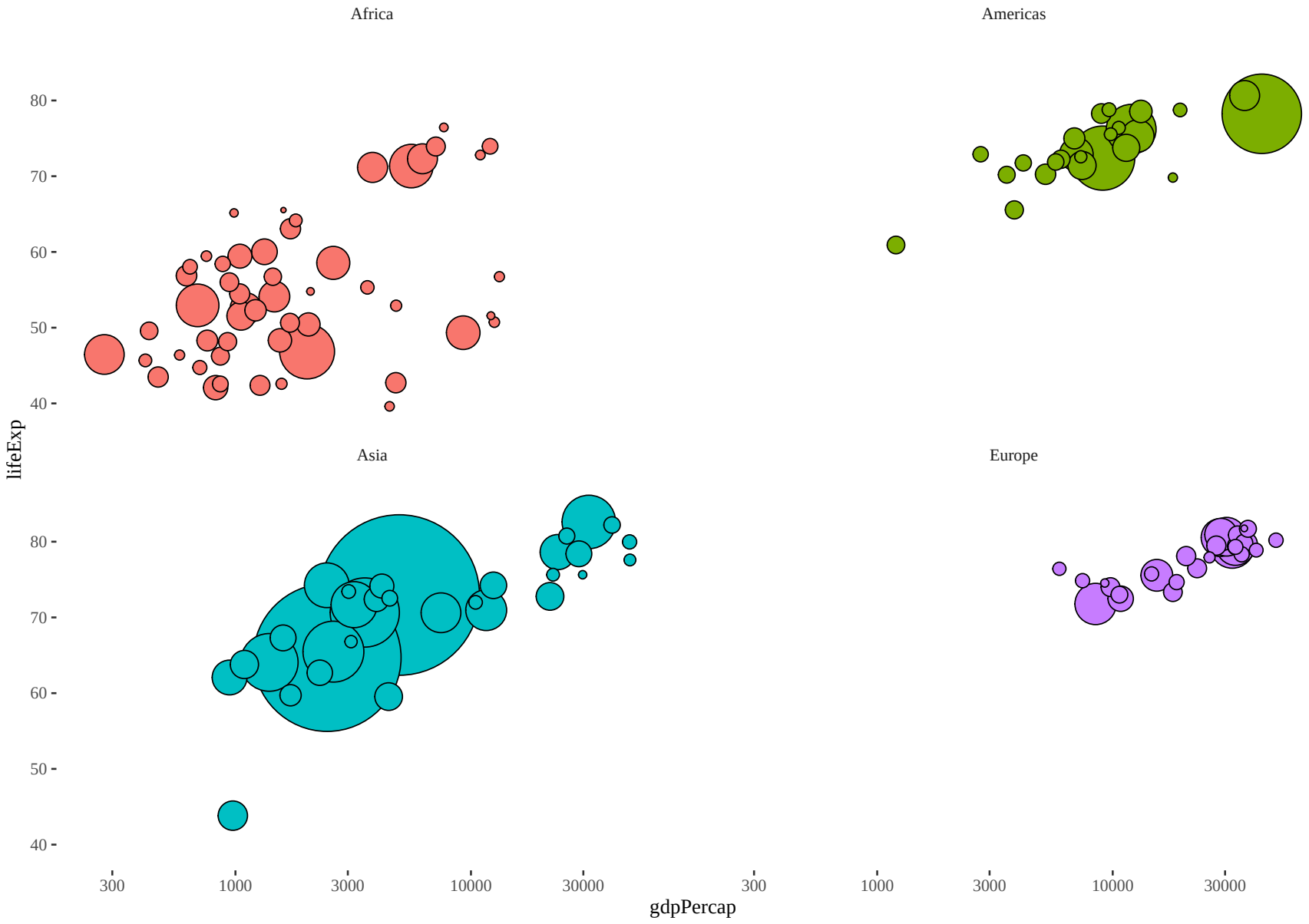
```
library(ggthemes)
reference_plot + theme_economist() + scale_fill_economist()
```



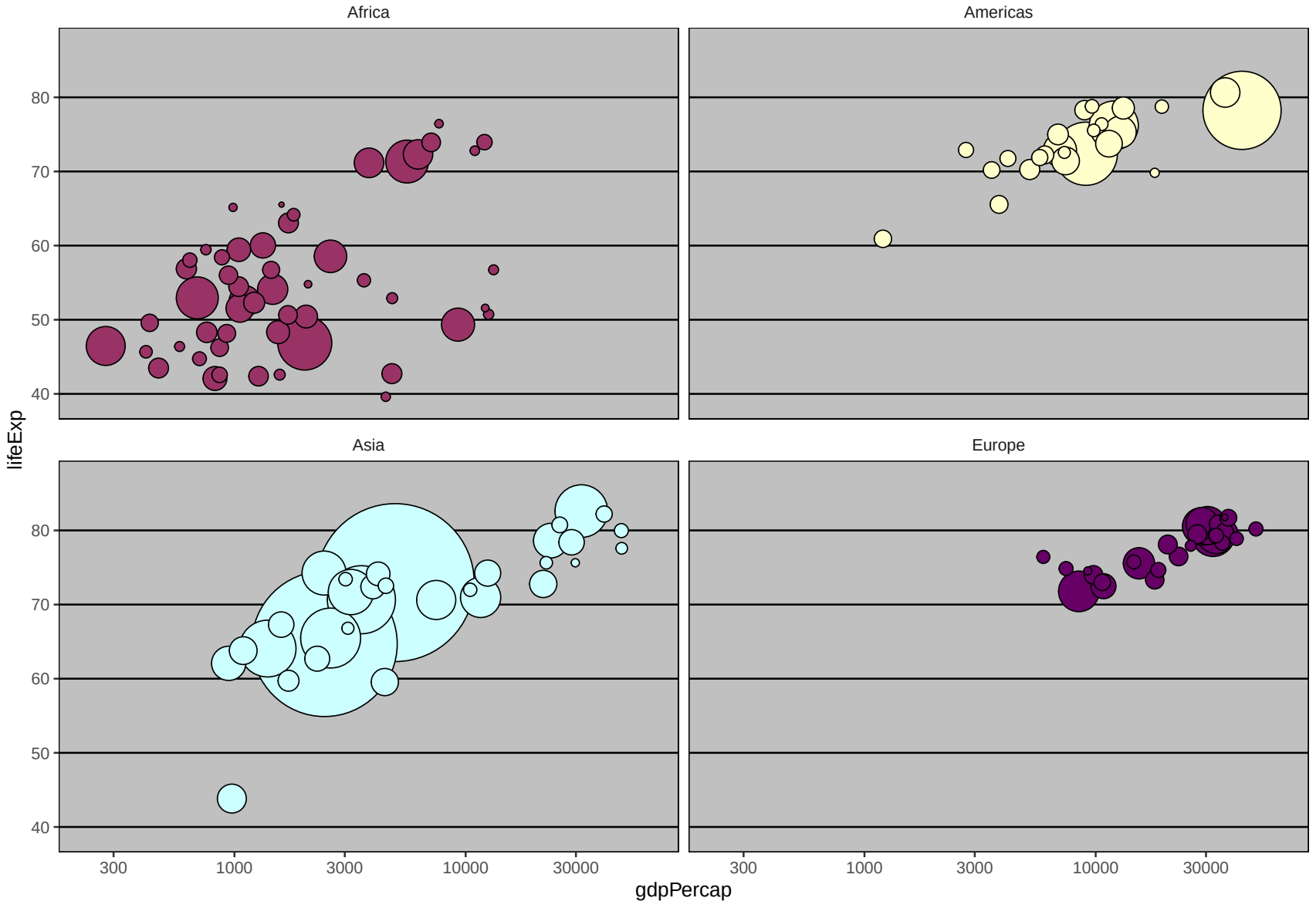
```
reference_plot + theme_wsj()+ scale_fill_wsj()
```



```
reference_plot + theme_tufte()
```

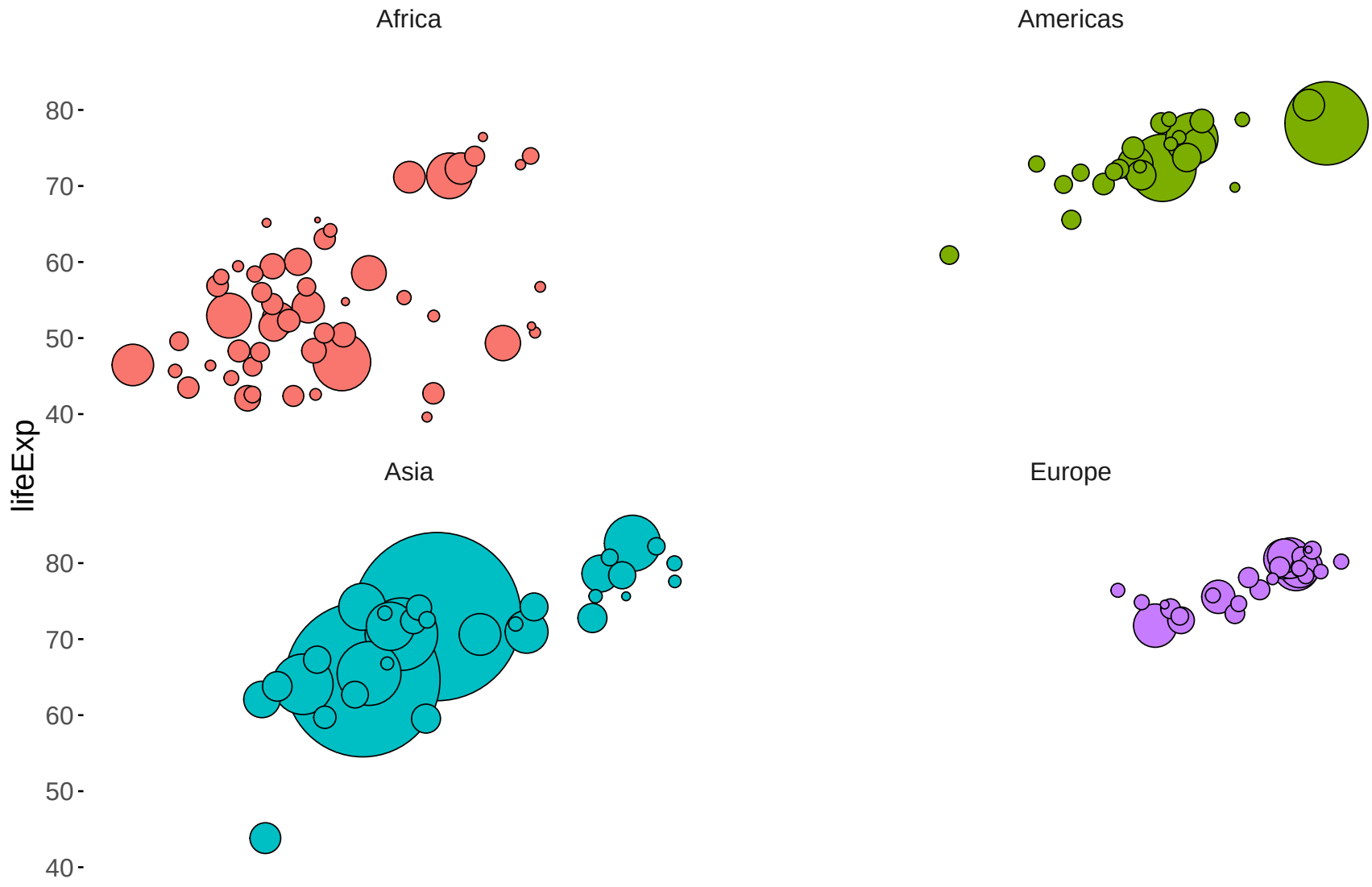


```
reference_plot + theme_excel() + scale_fill_excel()
```



```
library(extrafont)
library(xkcd)
reference_plot + theme_xkcd()
```

```
## Warning in theme_xkcd(): Not xkcd fonts installed! See v
```



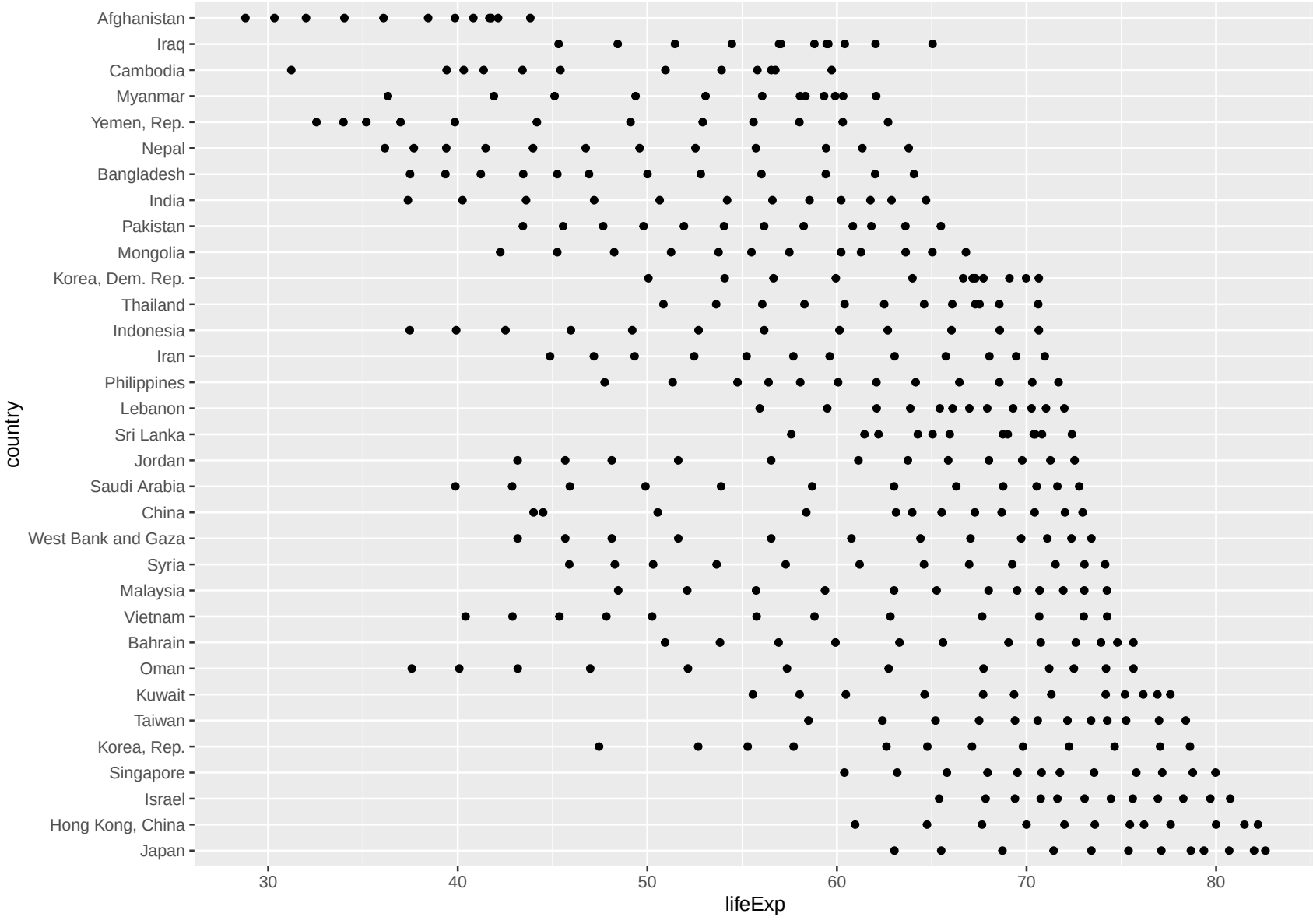
- Remember that we discussed other aspects of controlling the look of plots in Week 2.
- If you need to place several plots on the page, there are several packages you can use
  - `patchwork`
  - `gridExtra`
  - `cowplots`

Taking advantage of `fct_reorder()` or `fct_reorder2()` when plotting

(because this doesn't fit anywhere else)

- If you have a factor variable involved in plotting (or a character variable that will be converted to a factor), `ggplot` will plot it in the order of the factor—alphabetical if nothing else
- If that's not what you want, reordering the factor will do the trick
  - `fct_reorder()` to reorder by a single value
  - `fct_reorder2()` to reorder by two values

```r
library(gapminder)

p <- gapminder %>%
  filter(continent == "Asia") %>%
  mutate(country = fct_reorder2(country, .x = year, .y = li
  ggplot(aes(x = lifeExp, y = country)) +
  geom_point()
```


Afghanistan
Iraq
Cambodia
Myanmar
Yemen, Rep.
Nepal
Bangladesh
India
Pakistan
Mongolia
Korea, Dem. Rep.
Thailand
Indonesia
Iran
Philippines
Lebanon
Sri Lanka
Jordan
Saudi Arabia
China
West Bank and Gaza
Syria
Malaysia
Vietnam
Bahrain
Oman
Kuwait
Taiwan
Korea, Rep.
Singapore
Israel
Hong Kong, China
Japan
country
30
40
50
60
70
80
lifeExp

In class exercise

Make a plot of the use of the name Taylor from 1986 on by sex, using points connected by lines. Experiment with different colors and themes—impress your classmates. You can find a description of the themes in the `ggthemes` package at https://cran.r-project.org/web/packages/ggthemes/