

Class 5: Data visualization

Jaewon Kim (PID: A16675101)

2024-01-23

Graphic systems in R

There are many graphic systems in R for making plots and figures.

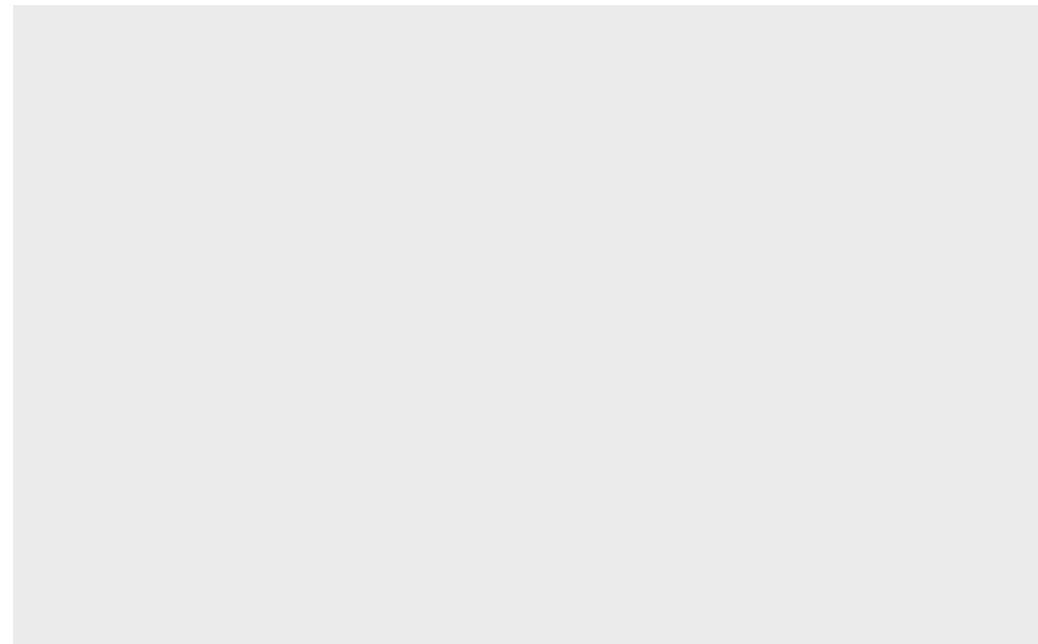
We have already played a little with “**base R**” graphics and the `plot()` function.

Today we will start learning about a popular graphics package called `ggplot2()`.

This is an add on package - i.e. we need to install it. I install it (like I install any package) with the `install.packages()` function.

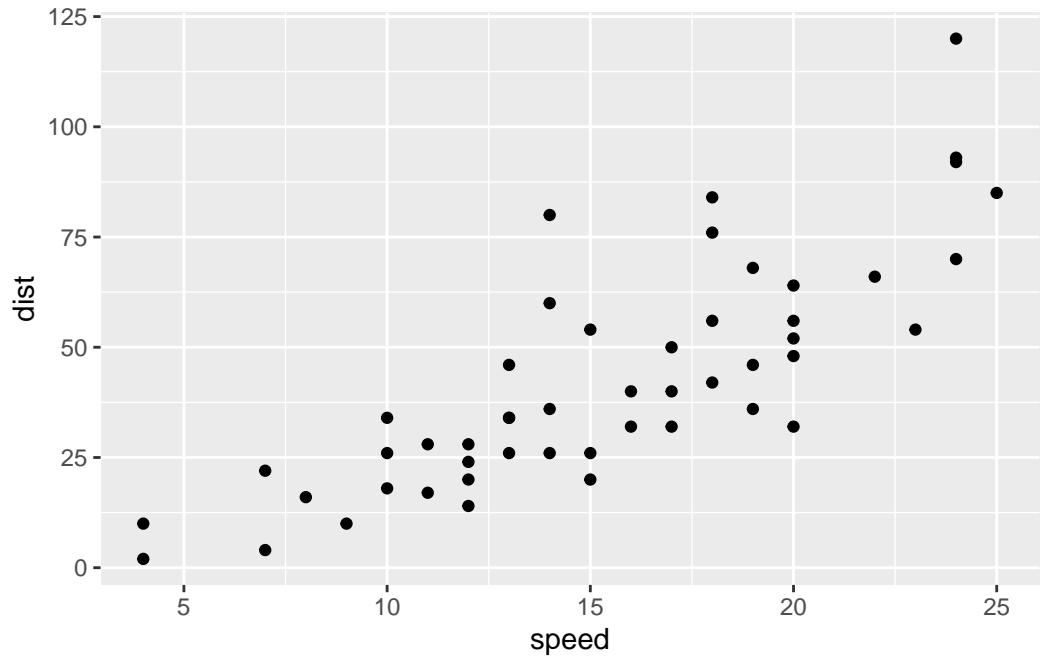
Before I can use the functions from a package I have to load up the package from my “library”. We use `library(ggplot2)` command to load it up.

```
library(ggplot2)
ggplot(cars)
```



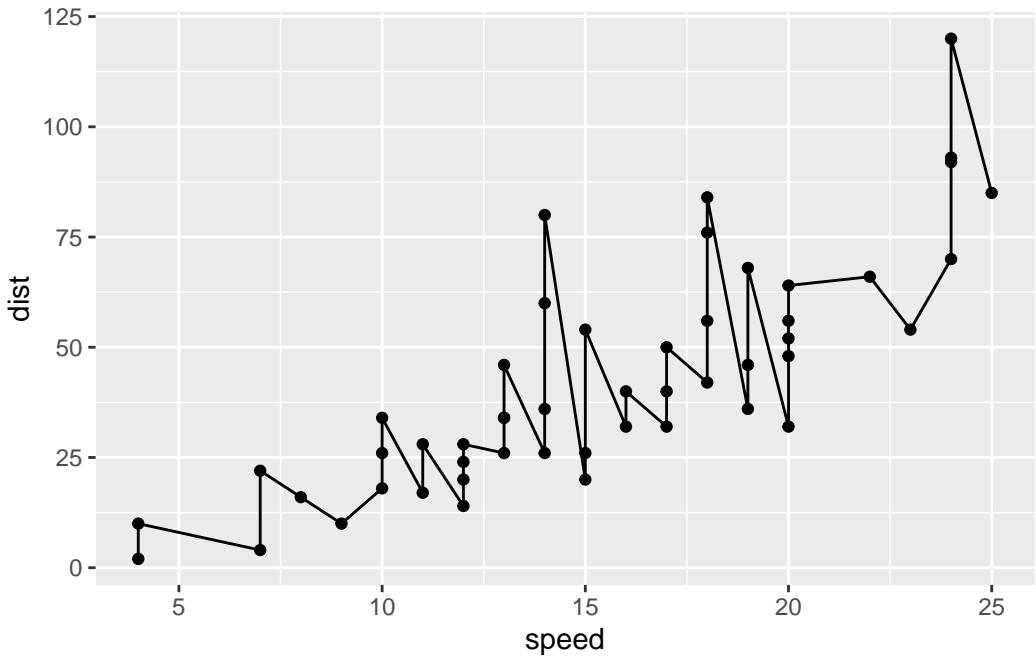
Every ggplot is made up of at least 3 things: - data (the numbers etc. that will go into your plot) - aes (how the columns of data map to the plot aesthetics) geoms (how the plot actually looks, points, bars, lines, etc.)

```
ggplot(cars) +  
  aes(x=speed, y=dist) + geom_point()
```

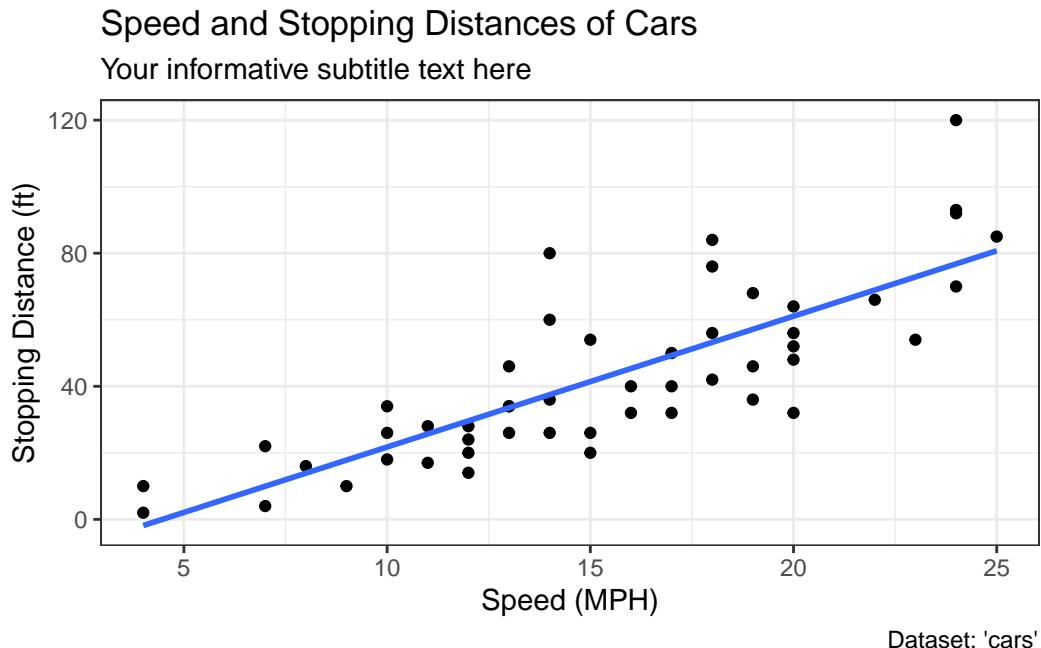


For simple plots ggplot is more verbose - it takes more code - than base R plot.

```
ggplot(cars) +  
  aes(x=speed, y=dist) + geom_point() + geom_line()
```



```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  labs(title="Speed and Stopping Distances of Cars",  
       x="Speed (MPH)", y="Stopping Distance (ft)",  
       subtitle = "Your informative subtitle text here",  
       caption="Dataset: 'cars'") +  
  geom_smooth(method="lm", se=FALSE) +  
  theme_bw()  
  
`geom_smooth()` using formula = 'y ~ x'
```



Adding more plot aes through aes()

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes)
```

	Gene	Condition1	Condition2	State
1	A4GNT	-3.6808610	-3.4401355	unchanging
2	AAAS	4.5479580	4.3864126	unchanging
3	AASDH	3.7190695	3.4787276	unchanging
4	AATF	5.0784720	5.0151916	unchanging
5	AATK	0.4711421	0.5598642	unchanging
6	AB015752.4	-3.6808610	-3.5921390	unchanging

Q. Use the nrow() function to find out how many genes are in this dataset. What is your answer?

```
nrow(genes)
```

```
[1] 5196
```

Q. How many columns did you find?

```
colnames(genes)
```

```
[1] "Gene"      "Condition1" "Condition2" "State"
```

```
ncol(genes)
```

```
[1] 4
```

Q. Use the table() function on the State column of this data.frame to find out how many ‘up’ regulated genes there are. What is your answer?

```
table(genes$State)
```

```
< table of extent 0 >
```

Q. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

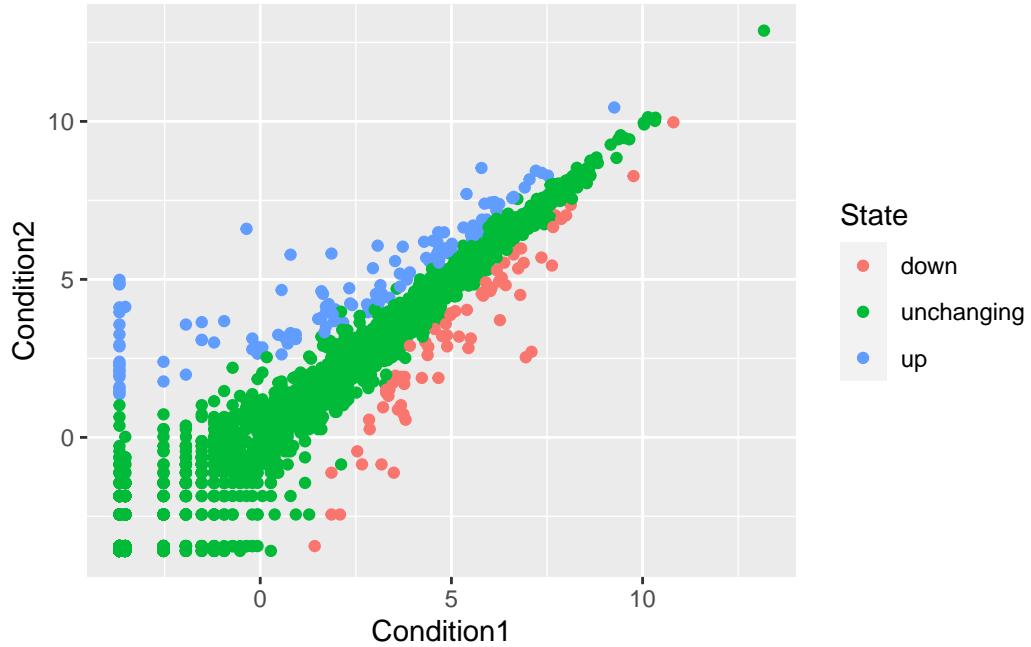
```
round(table(genes$State)/nrow(genes), 3)
```

	down	unchanging	up
	0.014	0.962	0.024

Q. Complete the code below to produce the following plot. There is extra information in this dataset, namely the State column, which tells us whether the difference in expression values between conditions is statistically significant.

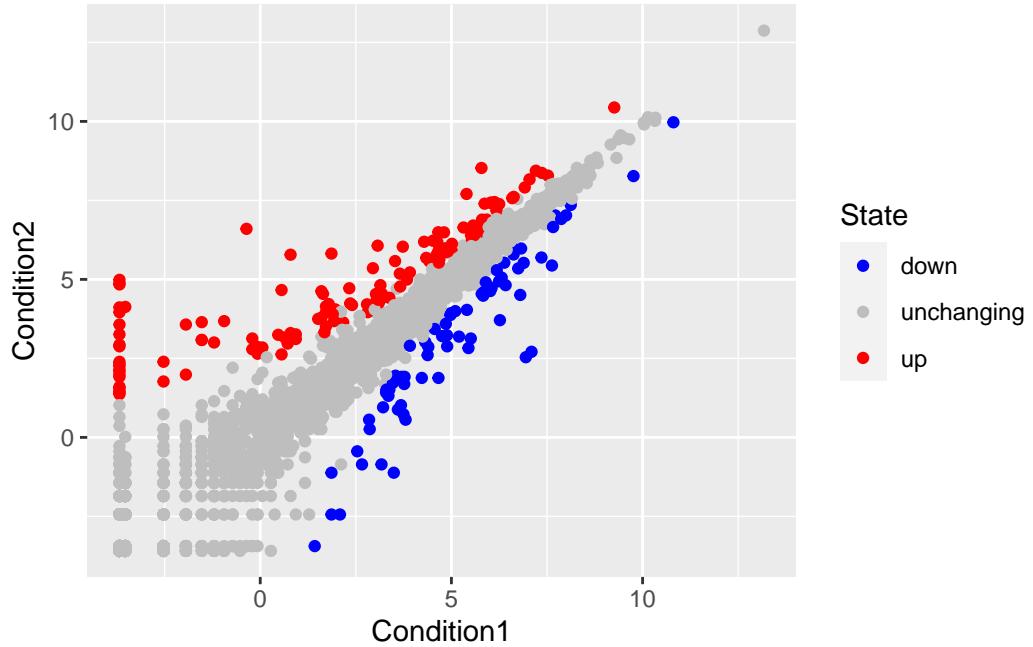
```
p <- ggplot(genes) + aes(x = Condition1, y = Condition2, col = State) +  
  geom_point()
```

```
p
```



Q. I am not a big fan of these default colors so let's change them up by adding another layer to explicitly specify our color scale.

```
p2 <- p +
  scale_colour_manual(values = c("blue", "grey", "red"))
p2
```

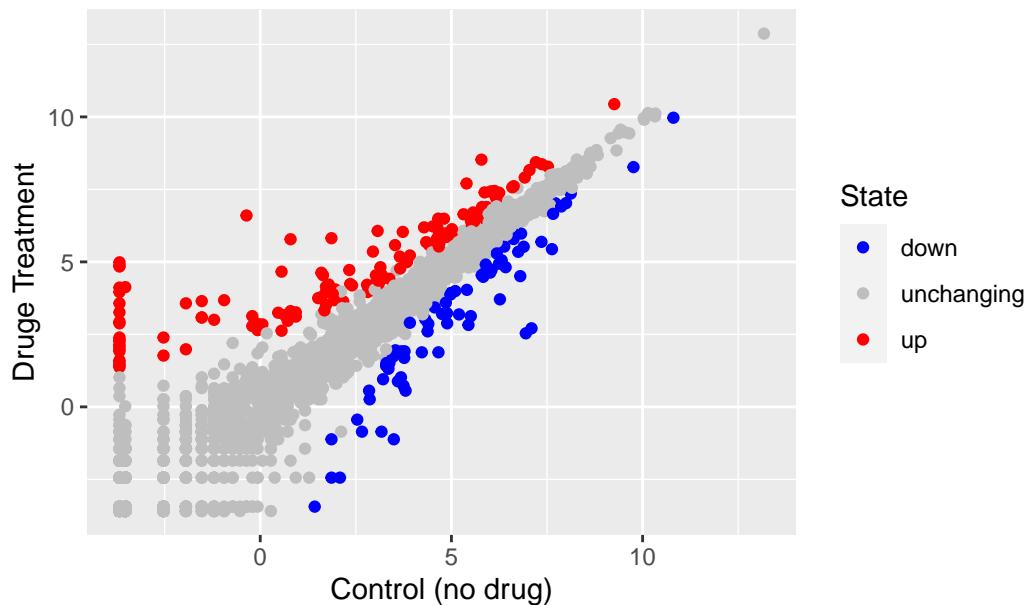


Q. Nice, now add some plot annotations to the p object with the labs() function so your plot looks like the following:

```
p <- p + scale_colour_manual(values = c("blue","grey","red")) +
  labs(title = "Gene Expression Changes Upon Drug Treatment", x = "Control (no drug)", y = "Drug Treatment")
```

p

Gene Expression Changes Upon Drug Treatment



Going further

Q. Install gapminder and filter 2007 data

```
#install.packages("gapminder")
library(gapminder)

#install.packages("dplyr")
library(dplyr)

: 'dplyr'

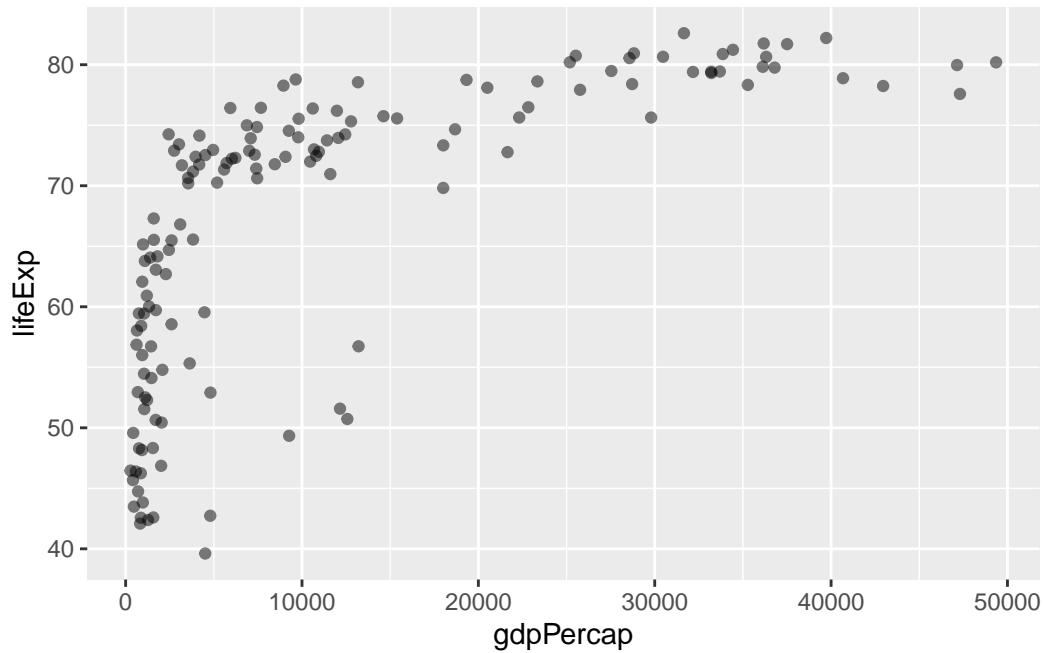
The following objects are masked from 'package:stats':
  filter, lag

The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union
```

```
gapminder_2007 <- gapminder %>% filter(year == 2007)
```

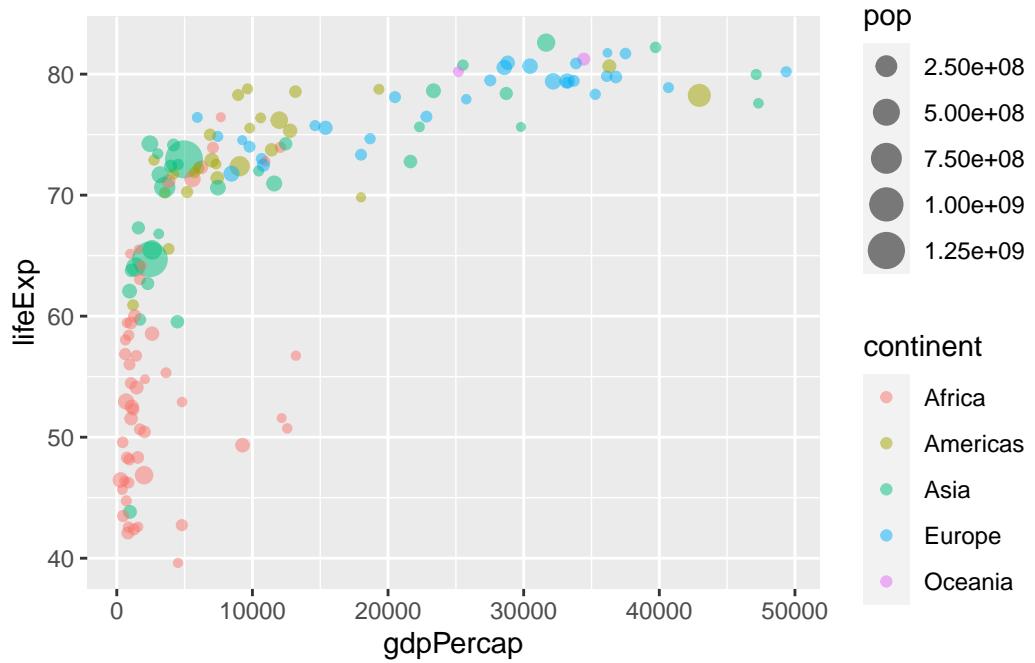
Q. Complete the code below to produce a first basic scatter plot of this gapminder_2007 dataset. Make points slightly transparent

```
ggplot(gapminder_2007) +  
  aes(x = gdpPercap, y = lifeExp) +  
  geom_point(alpha = 0.5)
```



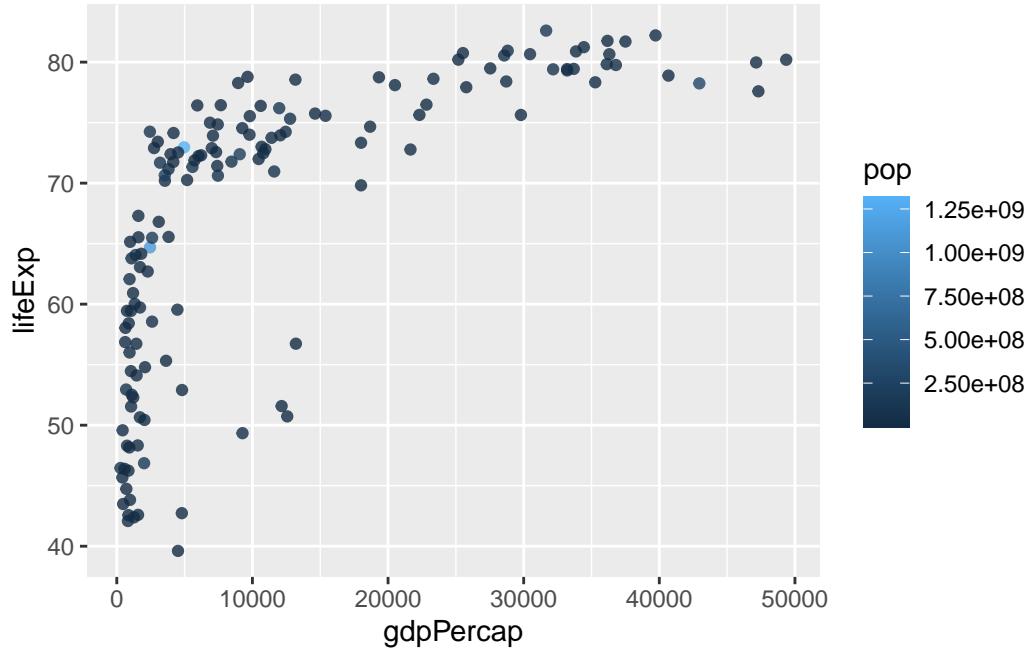
Q. Add size and continent aes

```
ggplot(gapminder_2007) +  
  aes(x = gdpPercap, y = lifeExp, color = continent, size = pop) +  
  geom_point(alpha = 0.5)
```



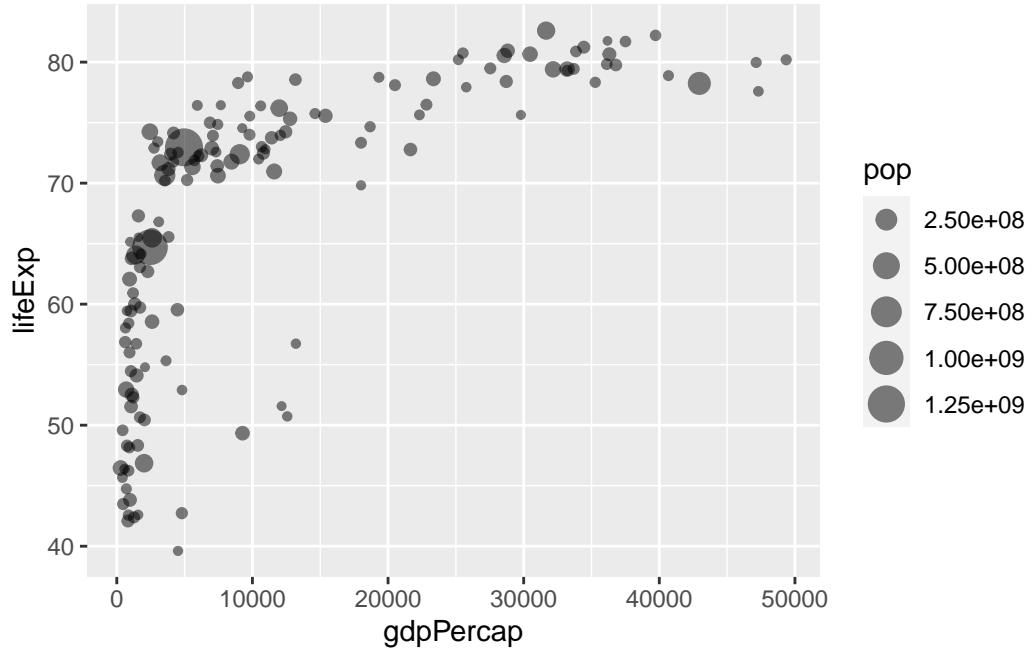
Q. coloring scheme based on the categorical data type of the variable continent.

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



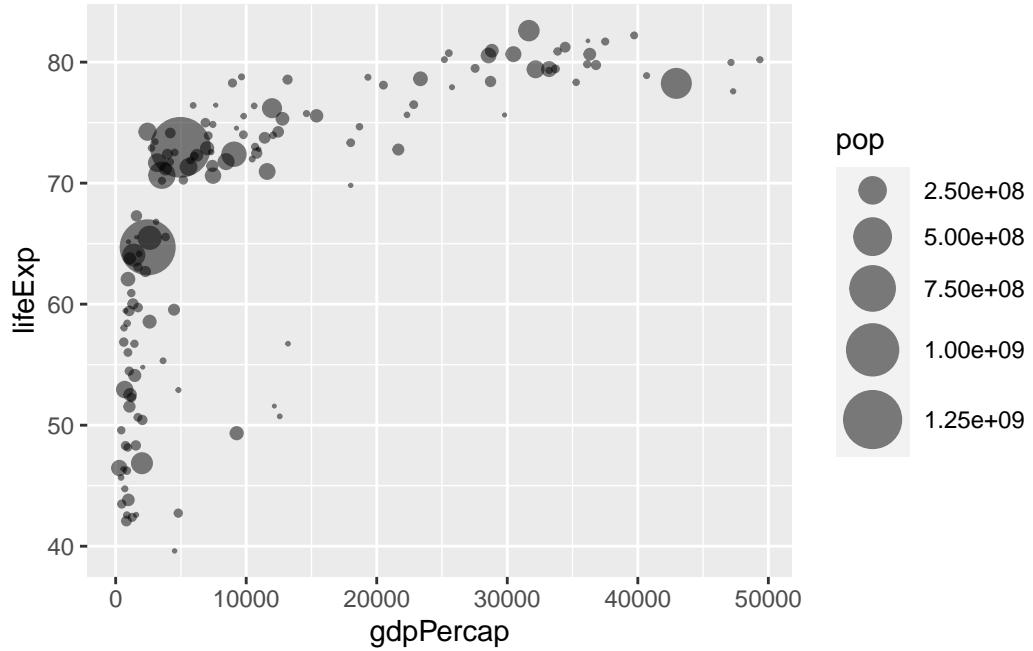
Q. For the gapminder_2007 dataset we can plot the GDP per capita (x=gdpPercap) vs. the life expectancy (y=lifeExp) and set the point size based on the population (size=pop) of each country we can use:

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```



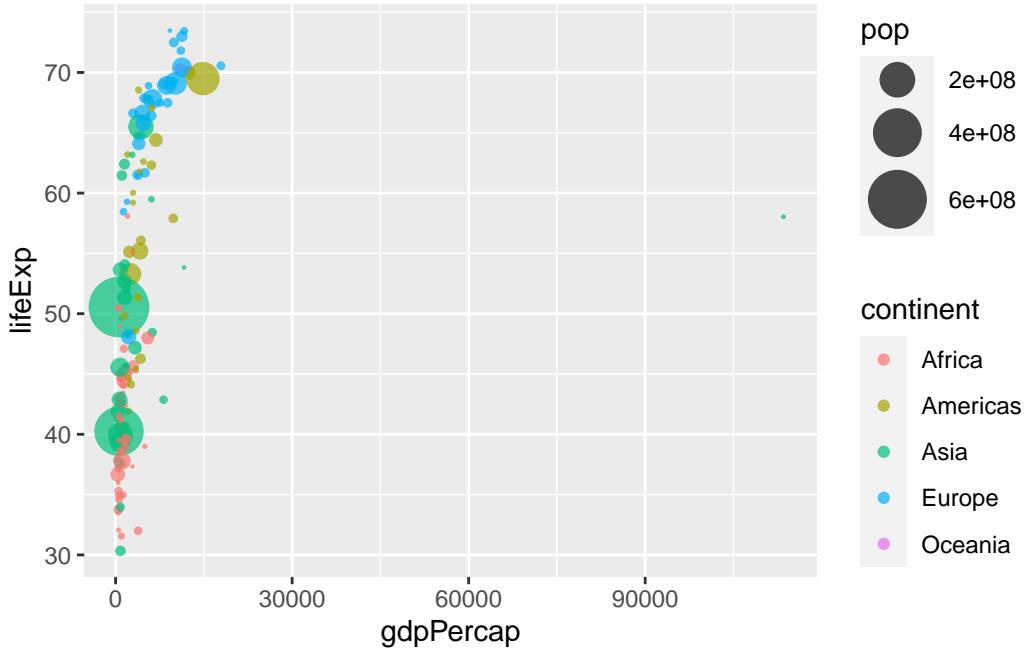
Q. To reflect the actual population differences by the point size we can use the `scale_size_area()` function instead.

```
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop), alpha = 0.5) +
  scale_size_area(max_size = 10)
```



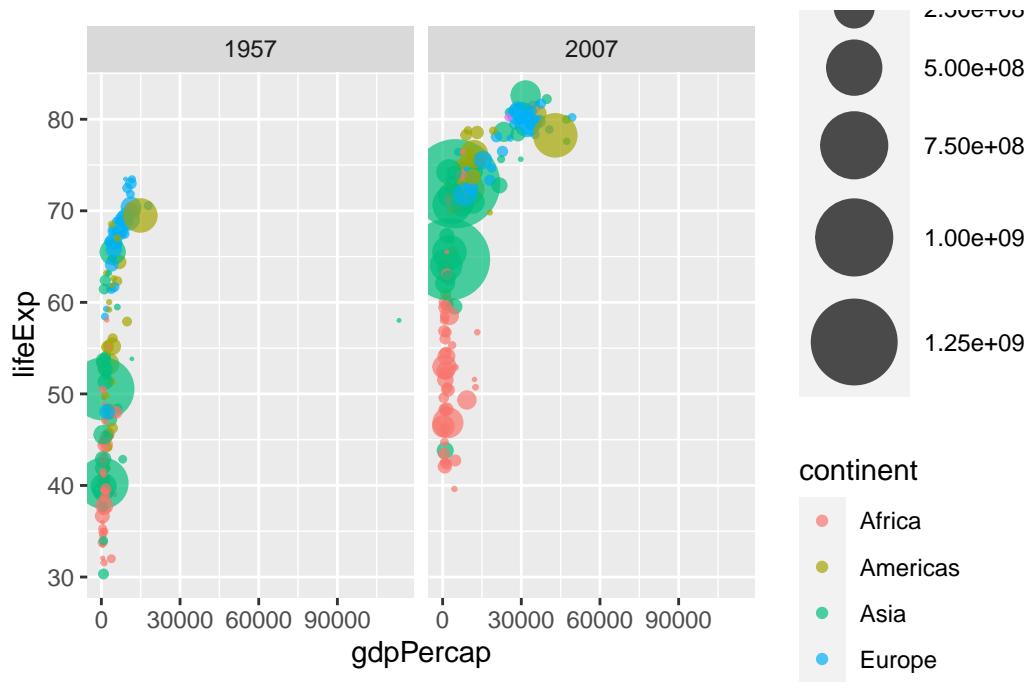
Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

```
gapminder_1957 <- gapminder %>% filter(year == 1957)
ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color=continent,
      size = pop) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 10)
```



Q. Do the same steps above but include 1957 and 2007 in your input dataset for ggplot(). You should now include the layer facet_wrap(~year) to produce the following plot:

```
gapminder_1957 <- gapminder %>% filter(year==1957 | year == 2007)
ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color=continent,
      size = pop) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 15) +
  facet_wrap(~year)
```



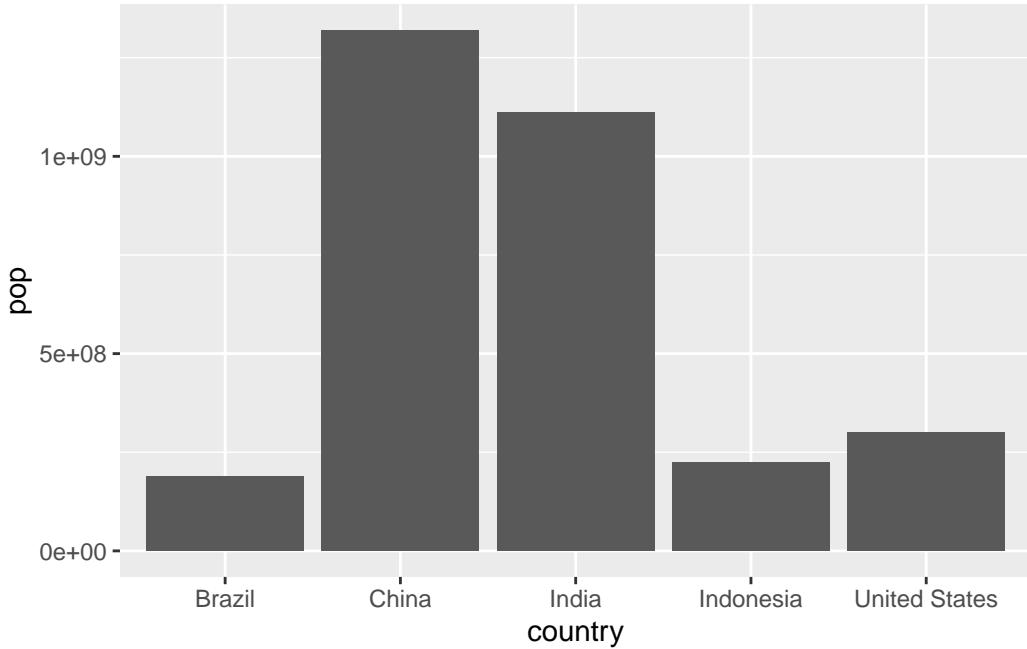
Q. Create bar chart for the biggest countries by population in 2007

```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

gapminder_top5
```

```
# A tibble: 5 x 6
  country   continent year lifeExp      pop gdpPerCap
  <fct>     <fct>    <int>   <dbl>     <int>     <dbl>
1 China      Asia      2007    73.0 1318683096     4959.
2 India      Asia      2007    64.7 1110396331     2452.
3 United States Americas 2007    78.2 301139947     42952.
4 Indonesia  Asia      2007    70.6 223547000     3541.
5 Brazil     Americas  2007    72.4 190010647     9066.
```

```
ggplot(gapminder_top5) + geom_col(aes(x = country, y = pop))
```



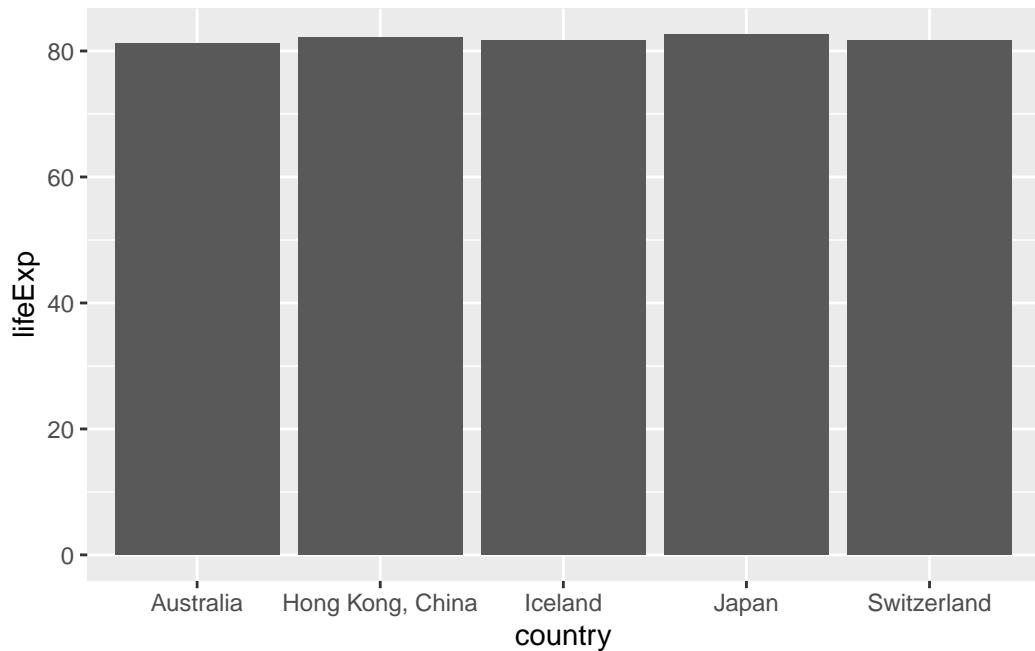
Q. Create a bar chart showing the life expectancy of the five biggest countries by population in 2007.

```
gapminder_top5exp <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(lifeExp)) %>%
  top_n(5, lifeExp)
```

```
gapminder_top5
```

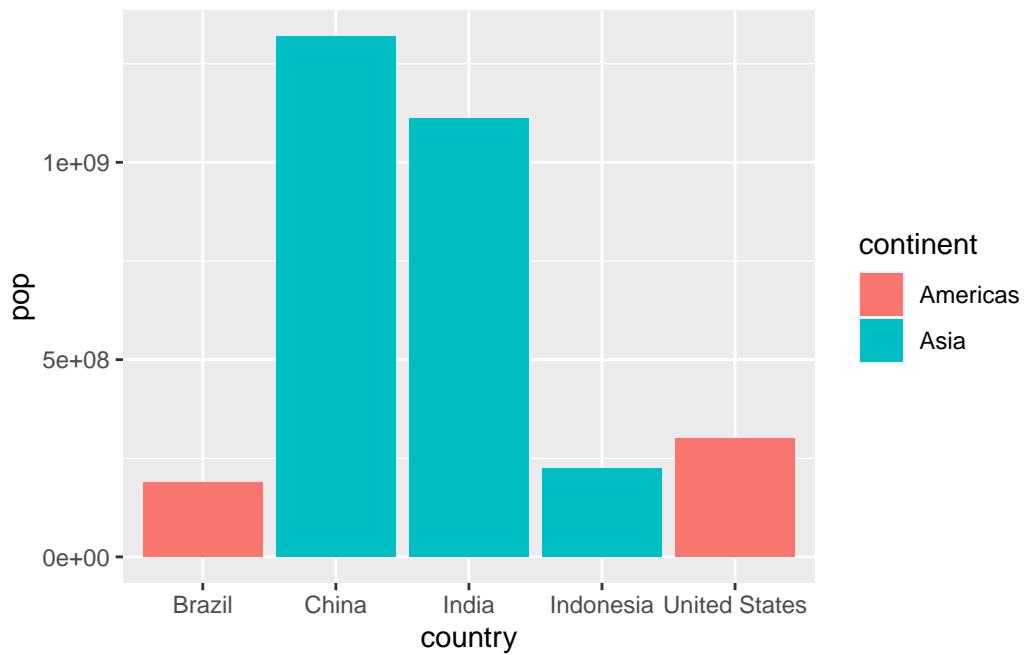
```
# A tibble: 5 x 6
  country      continent year lifeExp      pop gdpPercap
  <fct>        <fct>    <int>   <dbl>     <int>     <dbl>
1 China         Asia      2007    73.0 1318683096     4959.
2 India         Asia      2007    64.7 1110396331     2452.
3 United States Americas  2007    78.2 301139947      42952.
4 Indonesia     Asia      2007    70.6 223547000      3541.
5 Brazil        Americas  2007    72.4 190010647      9066.
```

```
ggplot(gapminder_top5exp) + geom_col(aes(x = country, y = lifeExp))
```



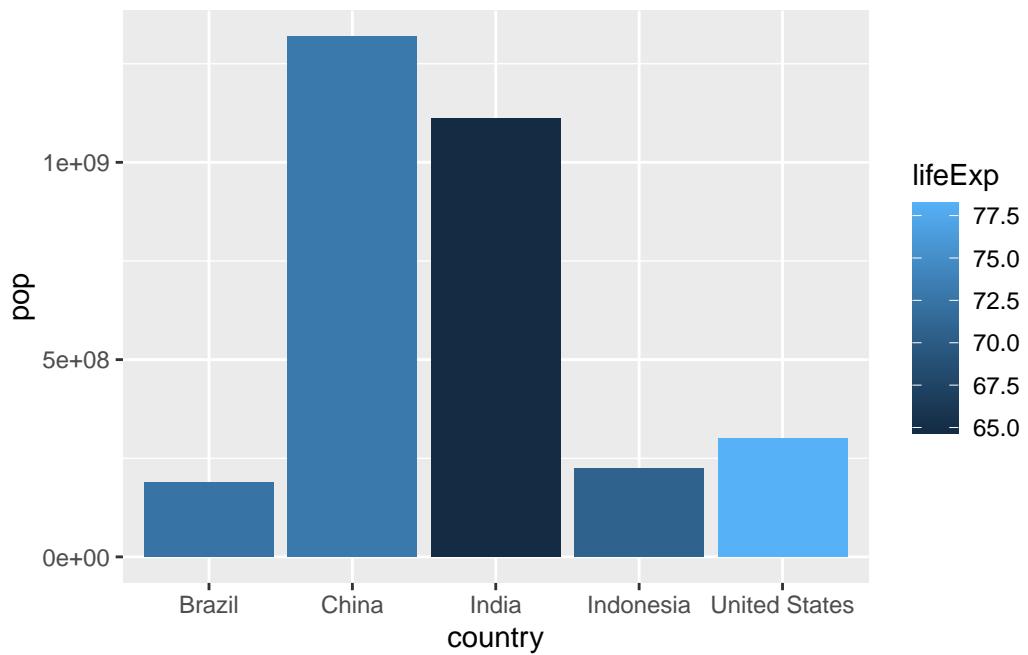
Q. Use color bars for each population.

```
ggplot(gapminder_top5) +  
  geom_col(aes(x = country, y = pop, fill = continent))
```



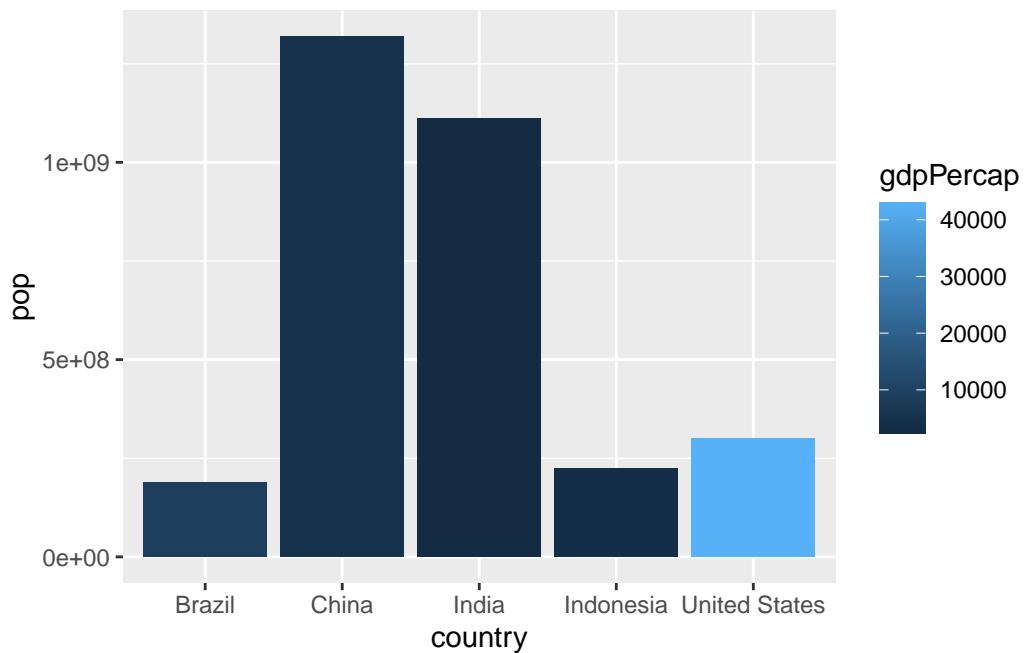
Q. Do the same but using life expectancy as filler

```
ggplot(gapminder_top5) +  
  geom_col(aes(x = country, y = pop, fill = lifeExp))
```

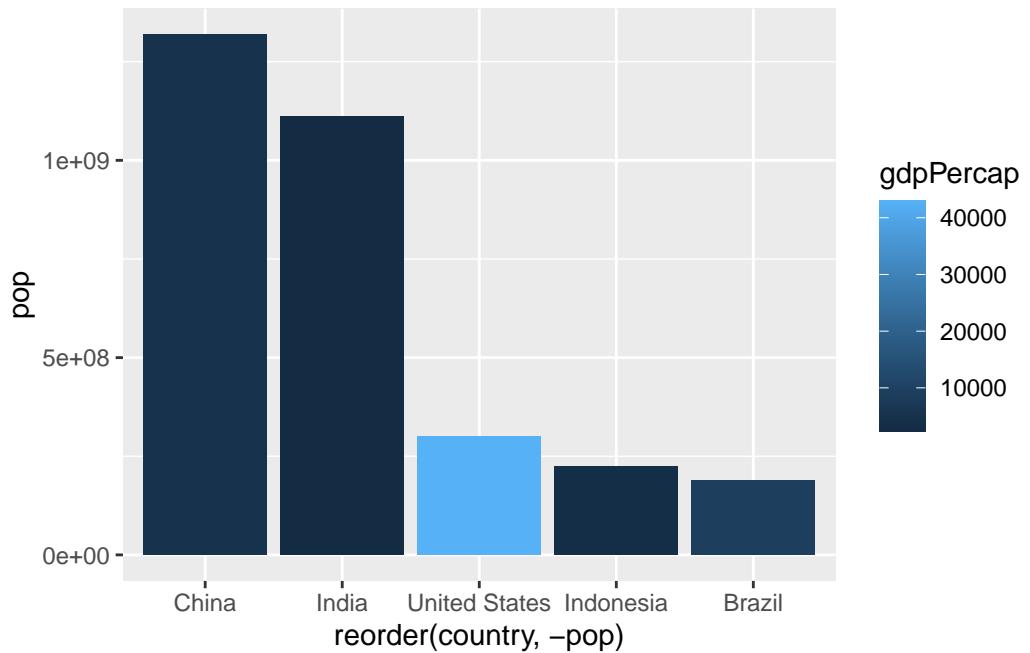


Q. Fill graph based on gdpPercap, change orders of bars, and fill by country.

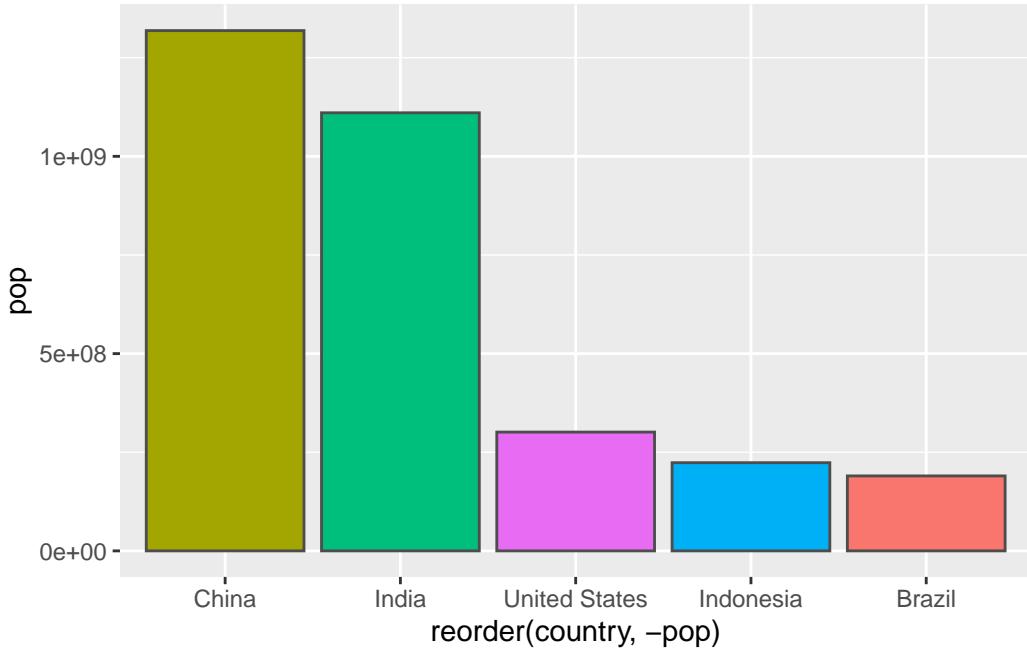
```
#Plot bargraph  
ggplot(gapminder_top5) +  
  geom_col(aes(x = country, y = pop, fill = gdpPercap))
```



```
#Reorder by population
ggplot(gapminder_top5) +
  geom_col(aes(x = reorder(country, -pop), y = pop, fill = gdpPerCap))
```



```
#Fill by country
ggplot(gapminder_top5) +
  aes(x = reorder(country, -pop), y = pop, fill = country) +
  geom_col(col = "gray30") +
  guides(fill = "none")
```



Q. Flip the bars

```
#Look at structure of USArrests
head(USArrests)
```

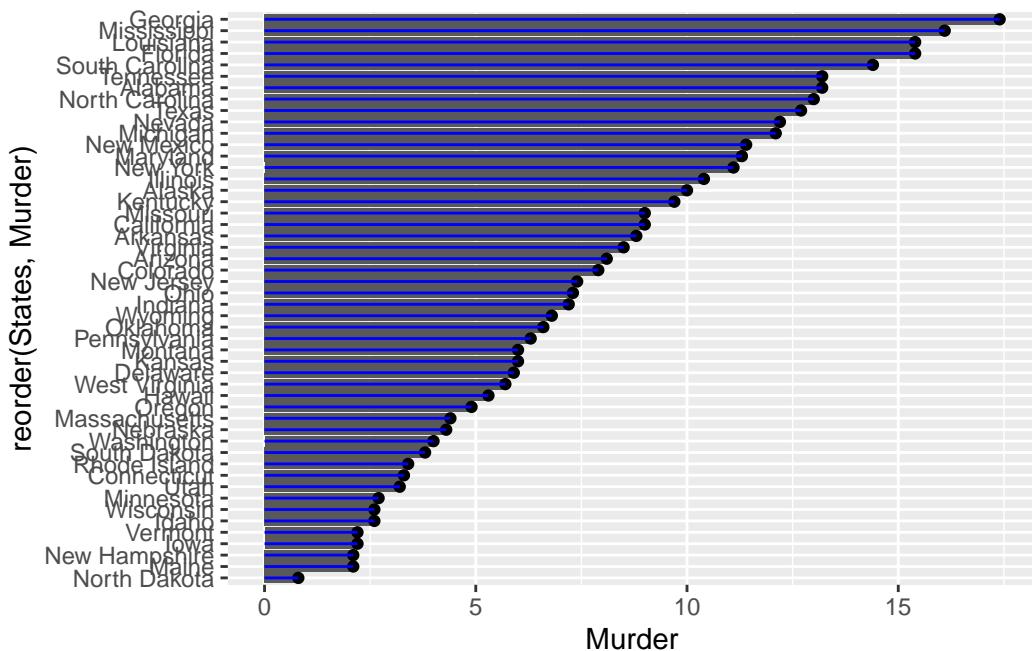
	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

```
#Create flipped bar chart
USArrests$States <- rownames(USArrests)
Murderplot <- ggplot(USArrests, aes(x = reorder(States, Murder), y = Murder)) + geom_col()
```

Q. Combine geom_point() and geom_segment()

```
Murderplot +
  geom_point() +
```

```
geom_segment(aes(x = States, xend = States,
                  y = 0, yend = Murder), color = "blue")
```



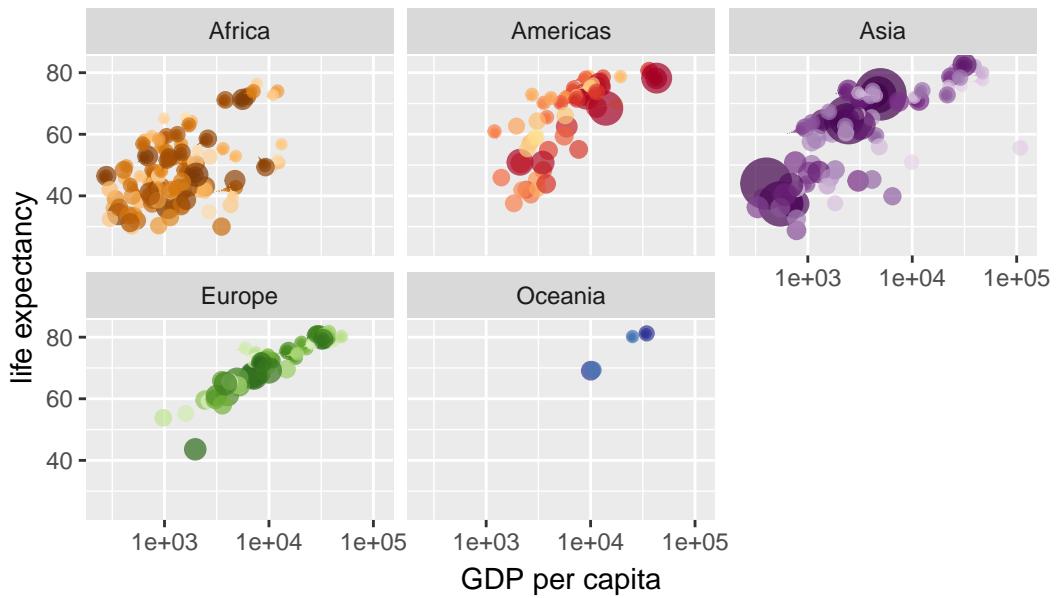
Q. Extension - Animation

```
# Install install.packages("gifski")
# Install install.packages("gganimate")

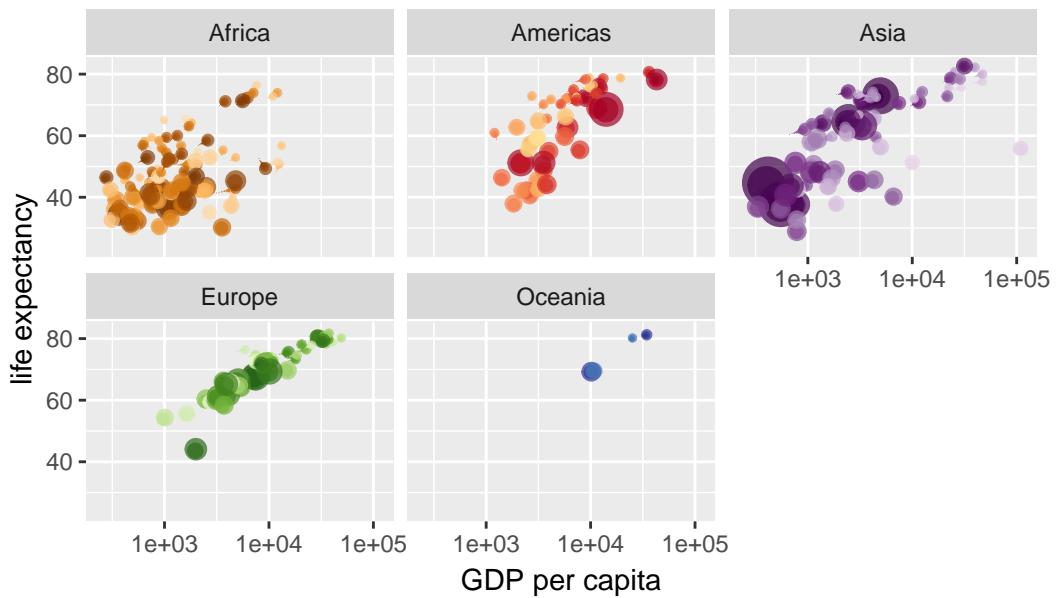
library(gapminder)
library(gganimate)

ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  facet_wrap(~continent) +
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
  transition_time(year) +
  shadow_wake(wake_length = 0.1, alpha = FALSE)
```

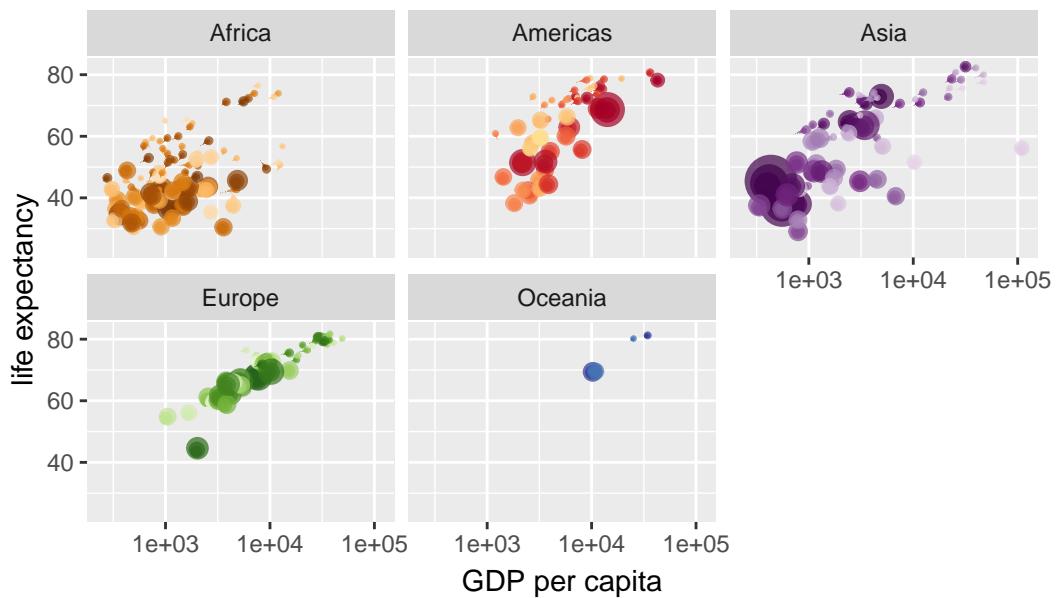
Year: 1952



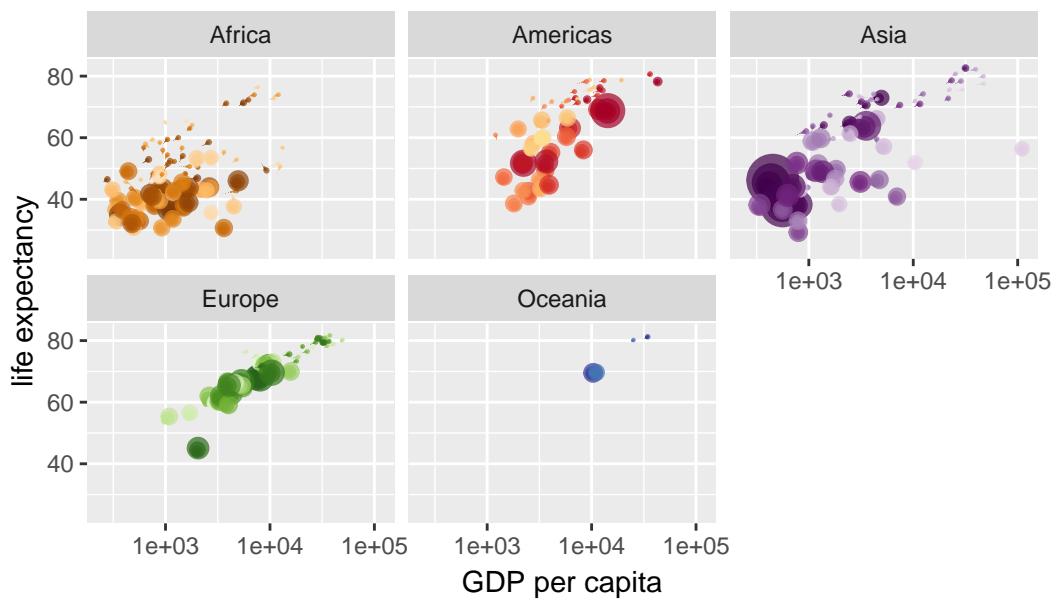
Year: 1953



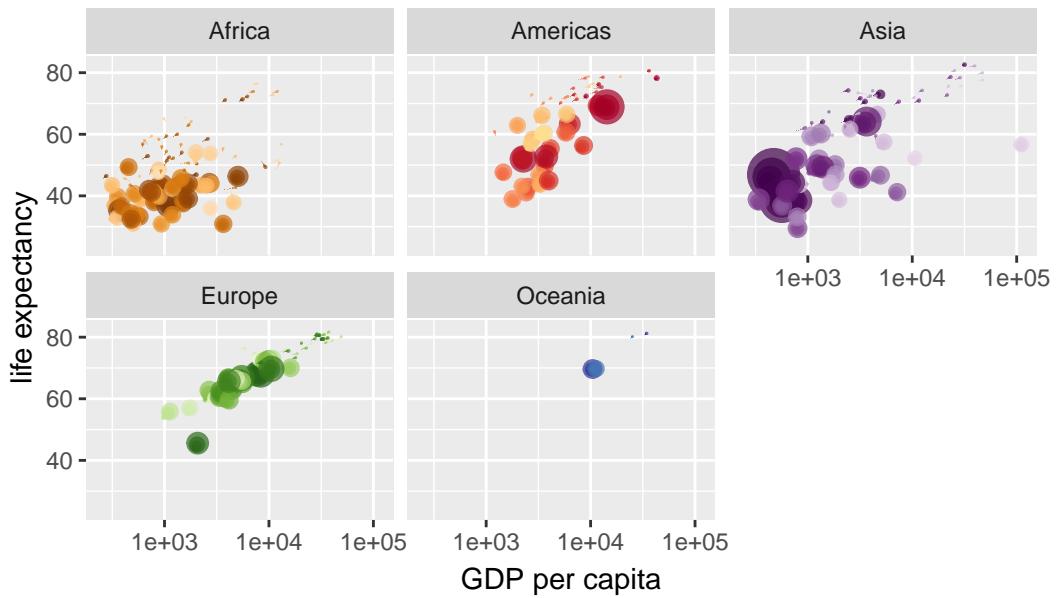
Year: 1953



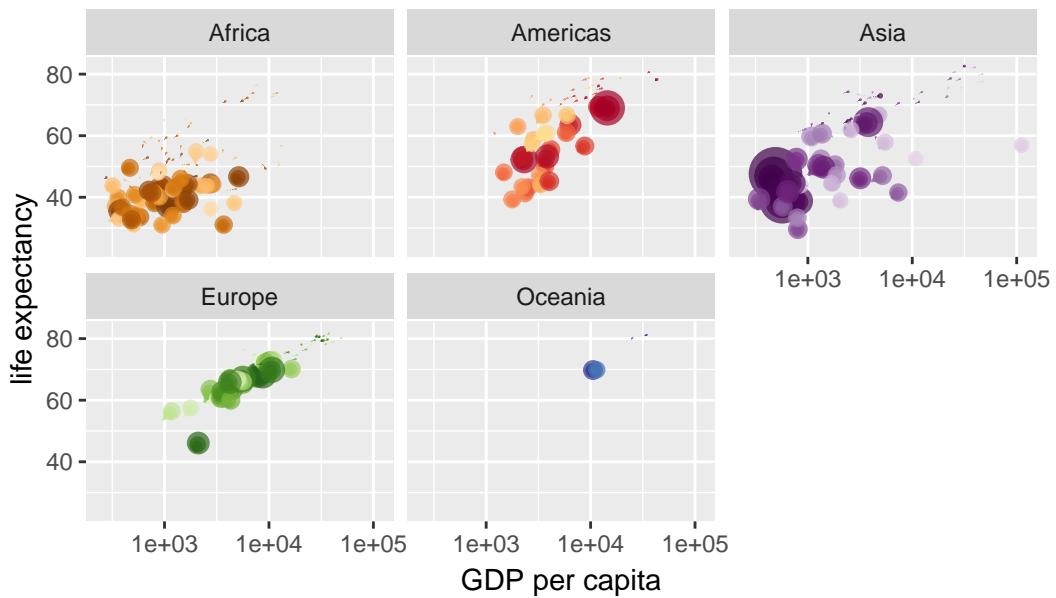
Year: 1954



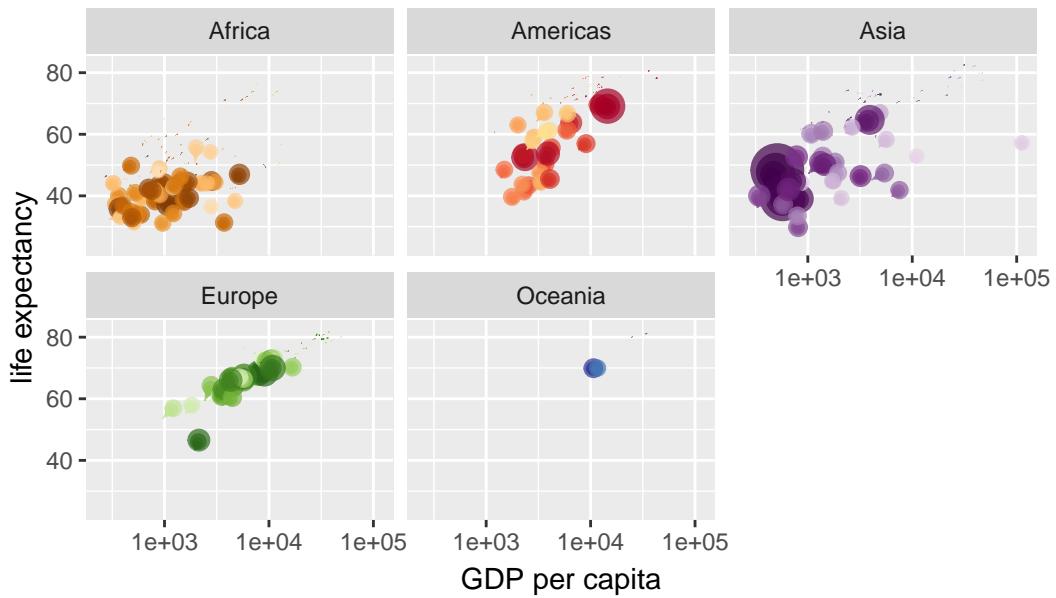
Year: 1954



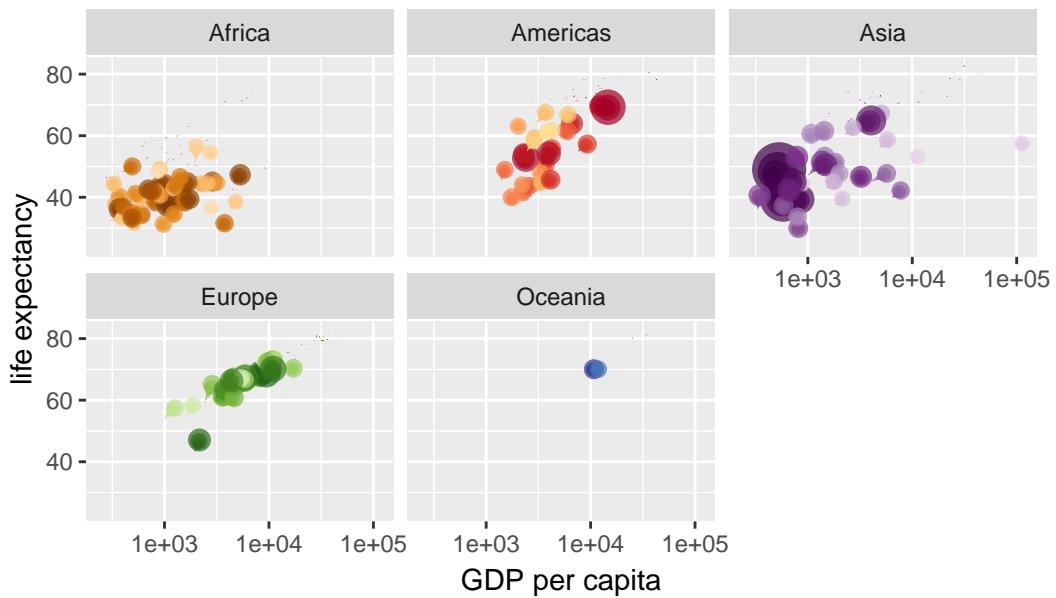
Year: 1955



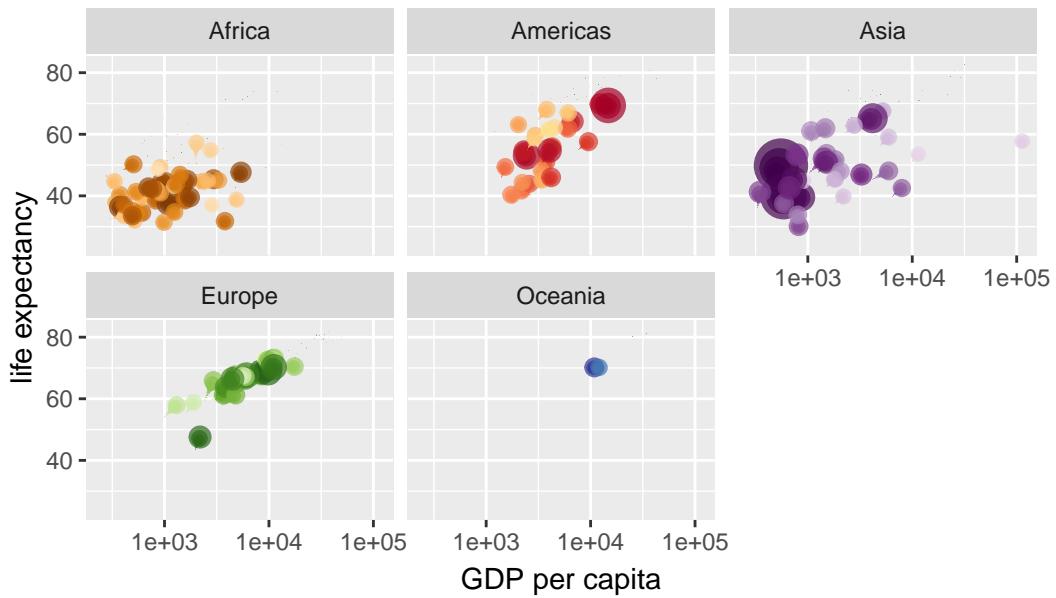
Year: 1955



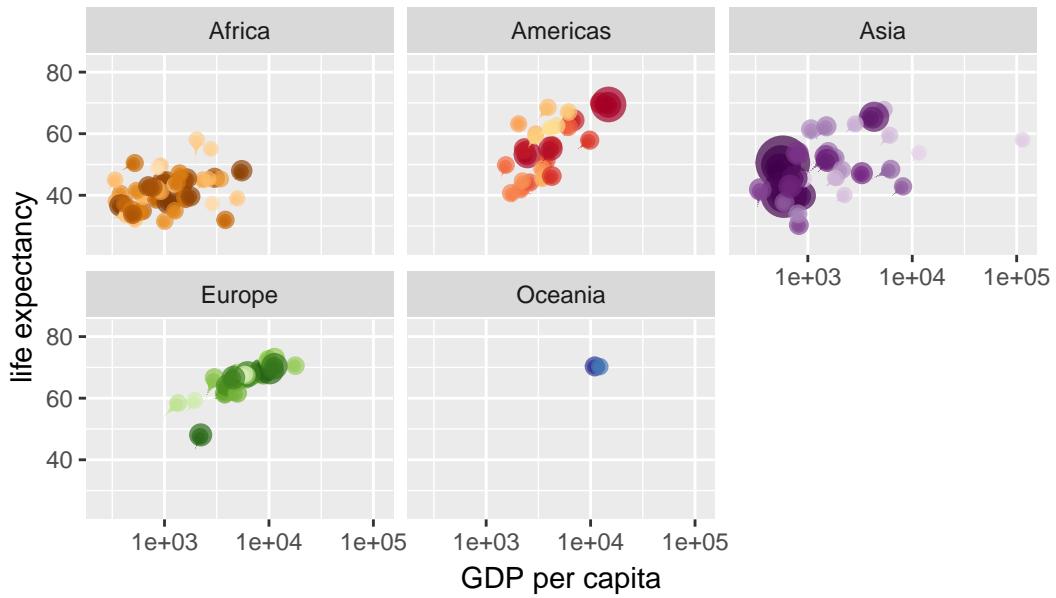
Year: 1956



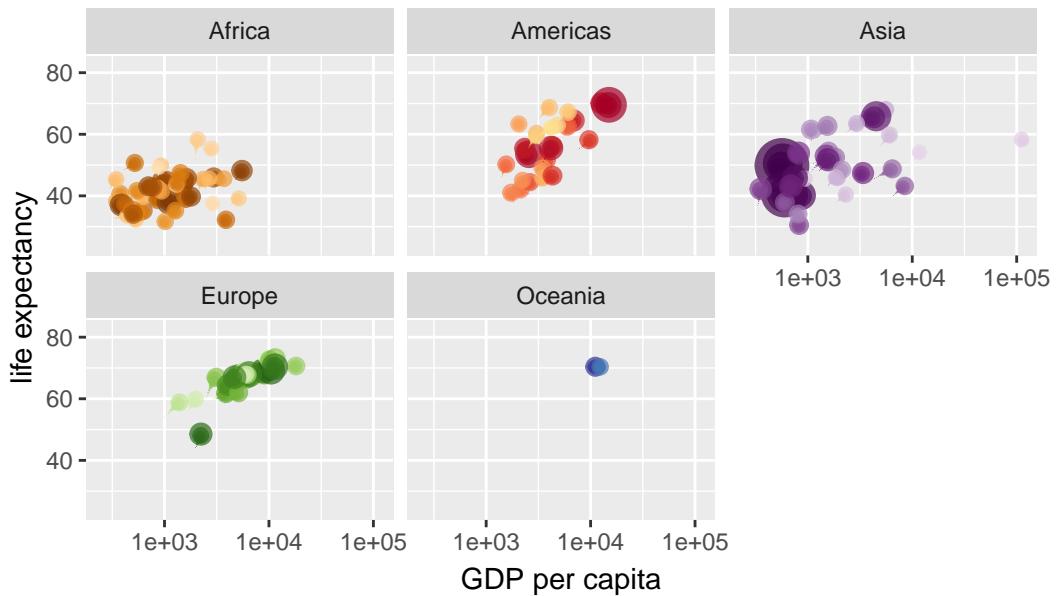
Year: 1956



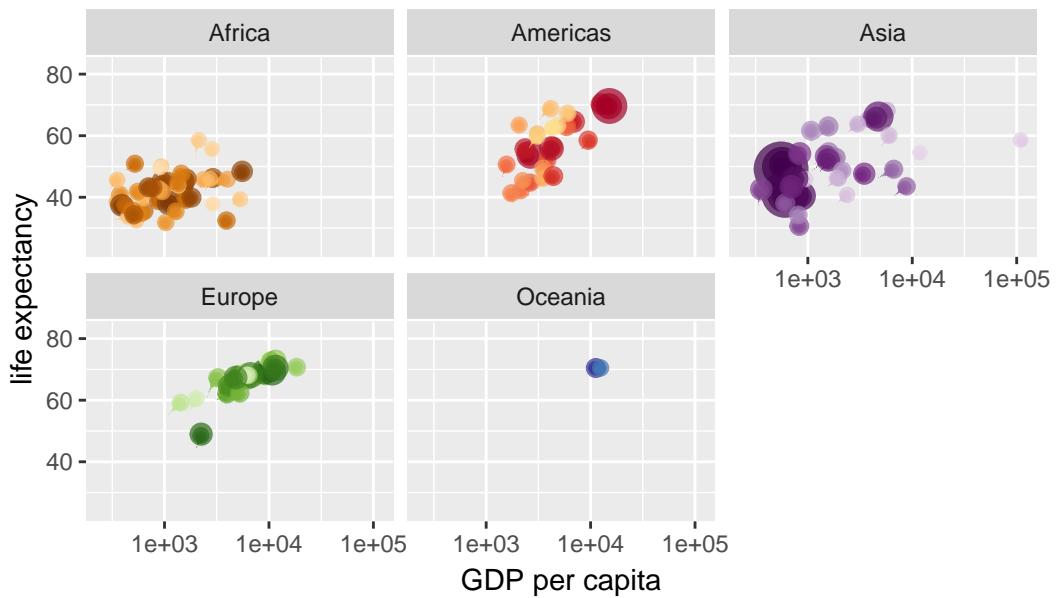
Year: 1957

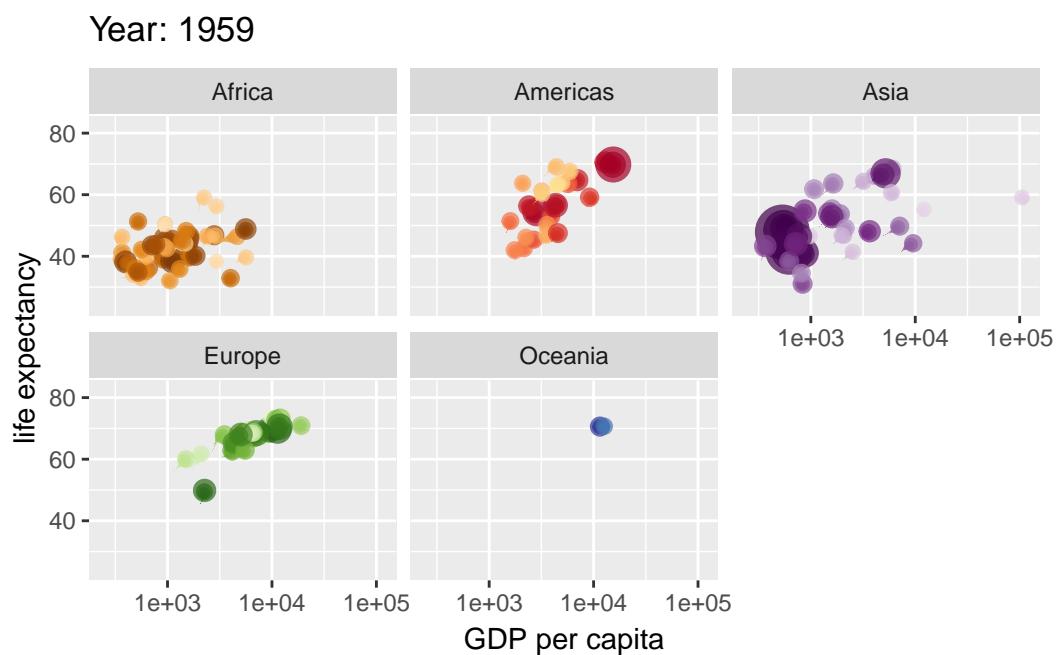
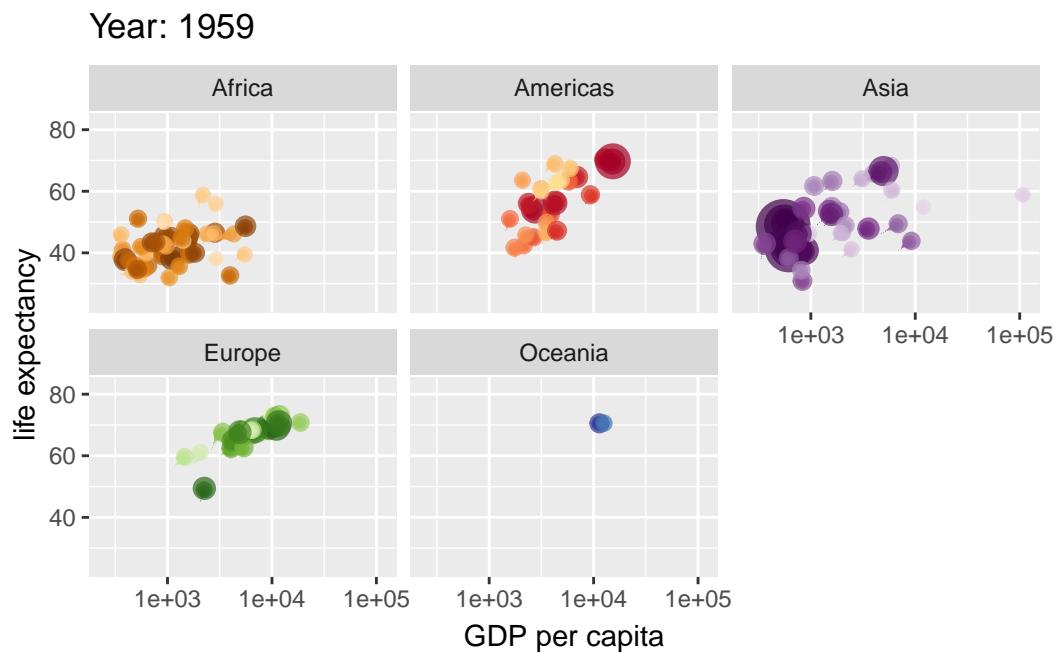


Year: 1958

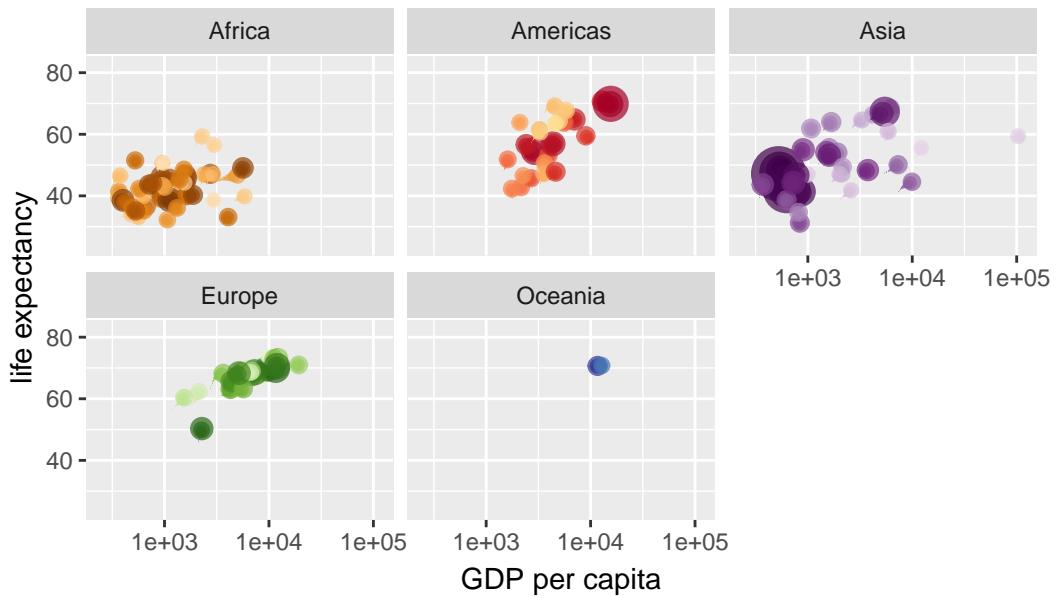


Year: 1958

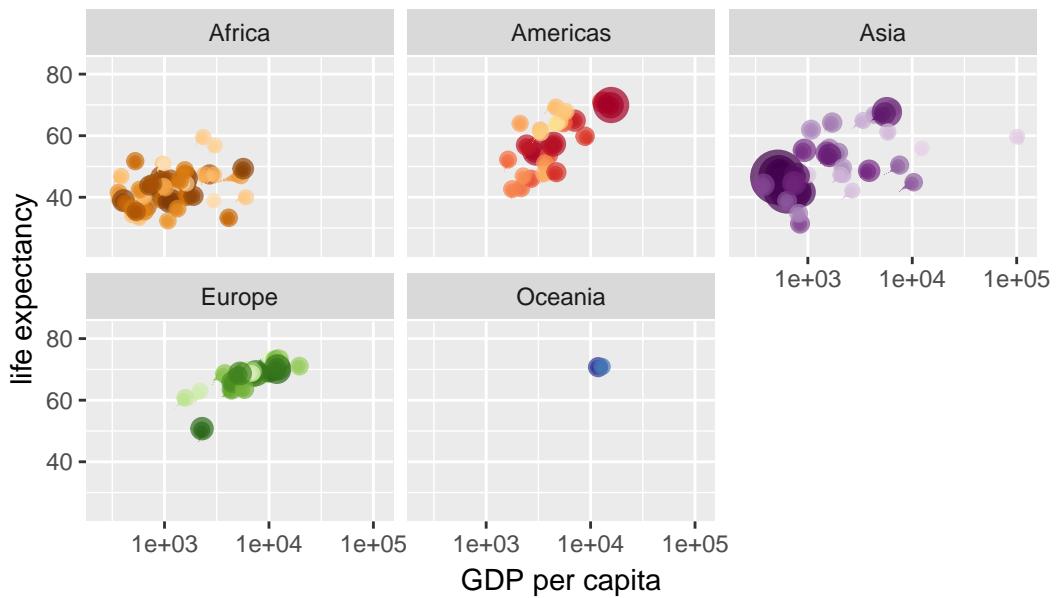




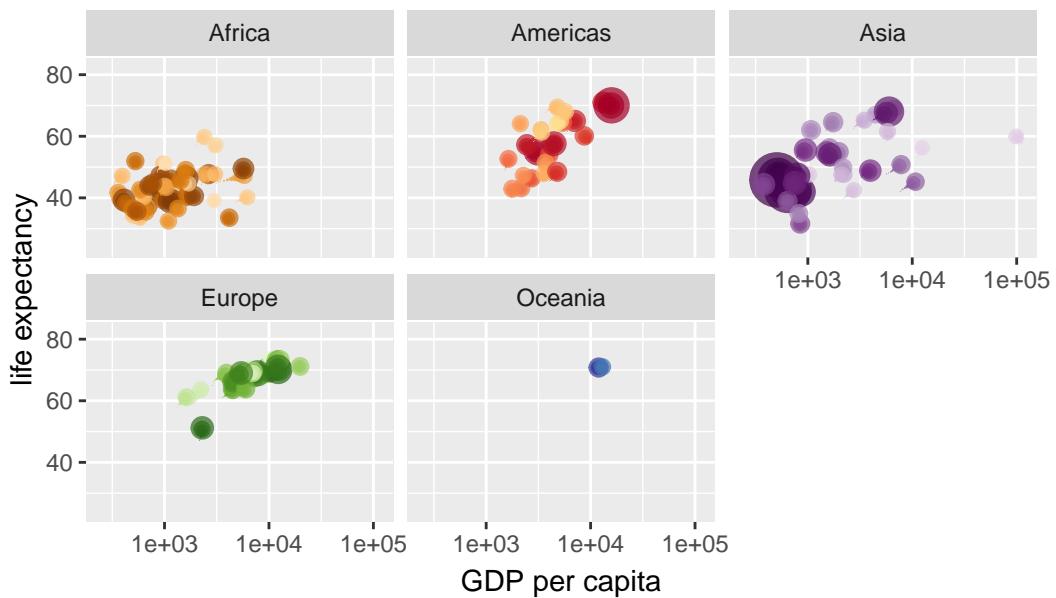
Year: 1960



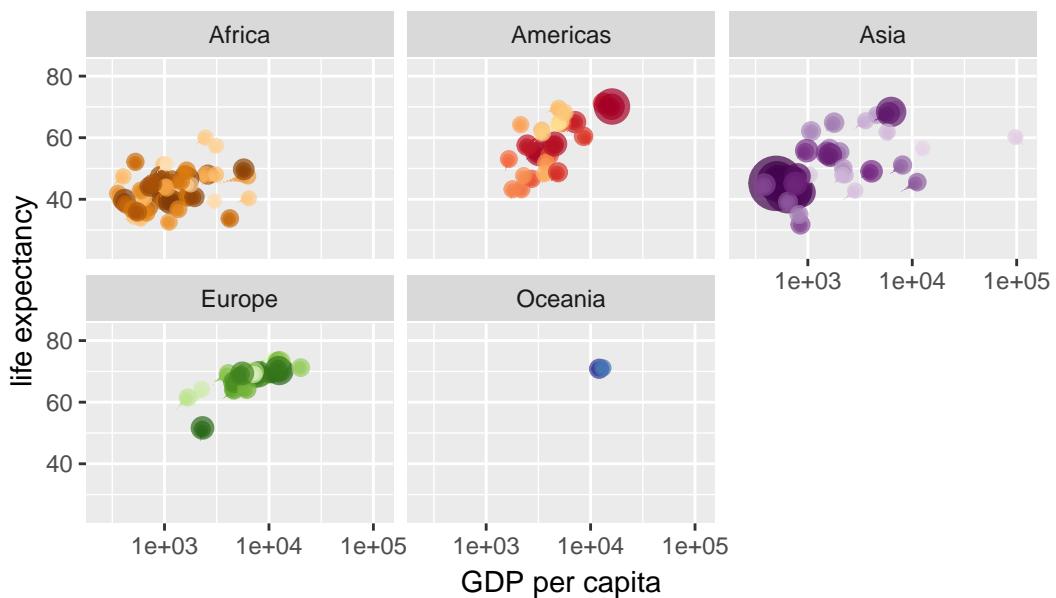
Year: 1960



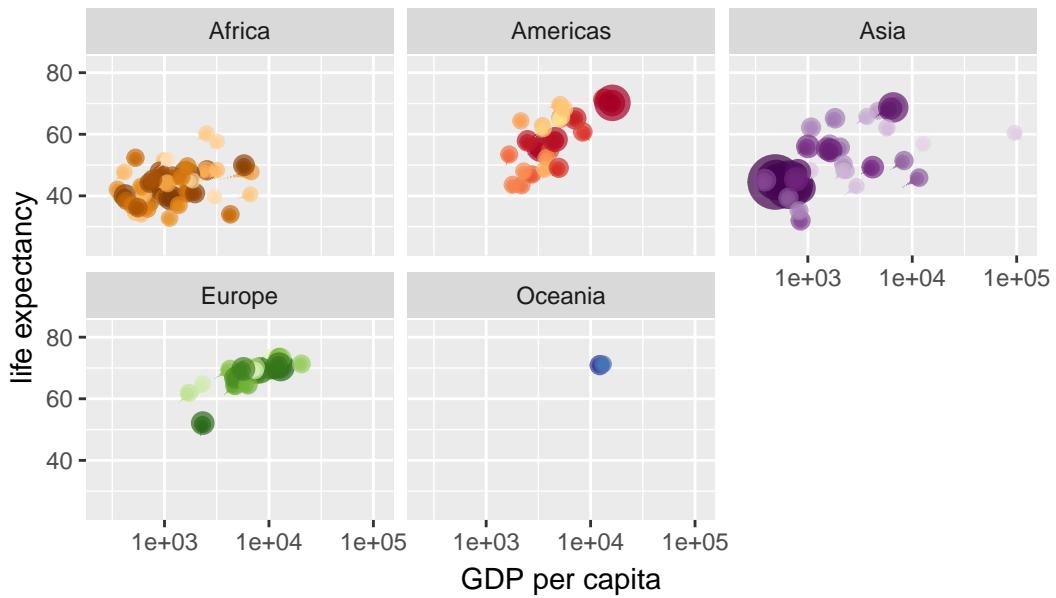
Year: 1961



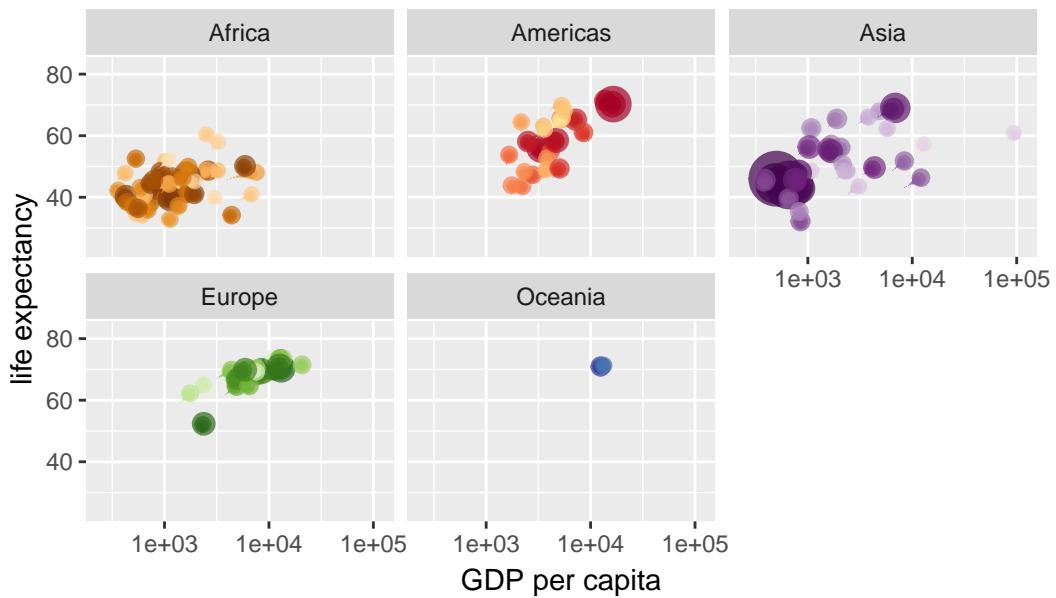
Year: 1961



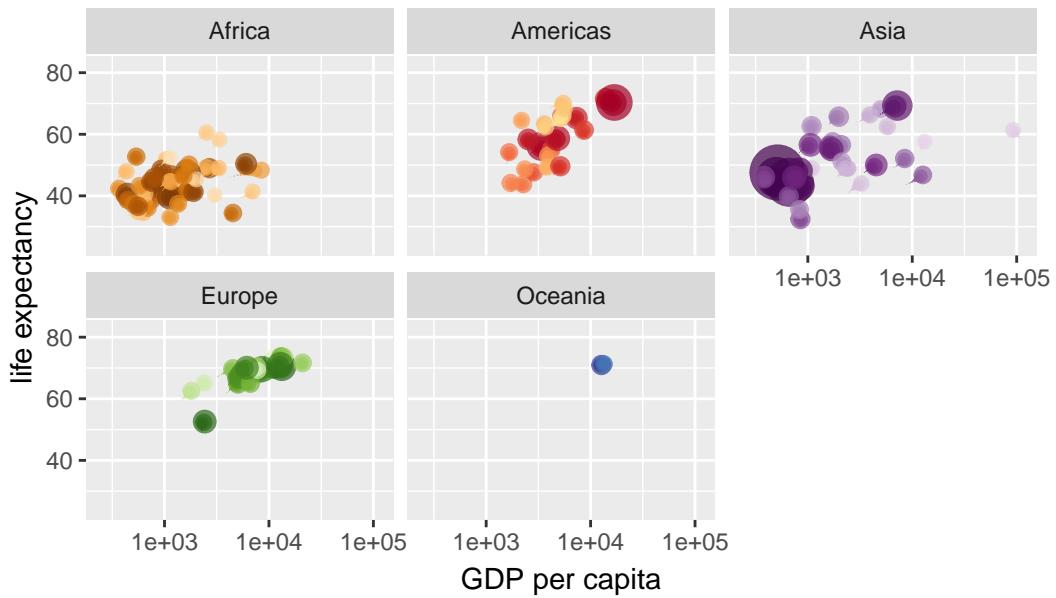
Year: 1962



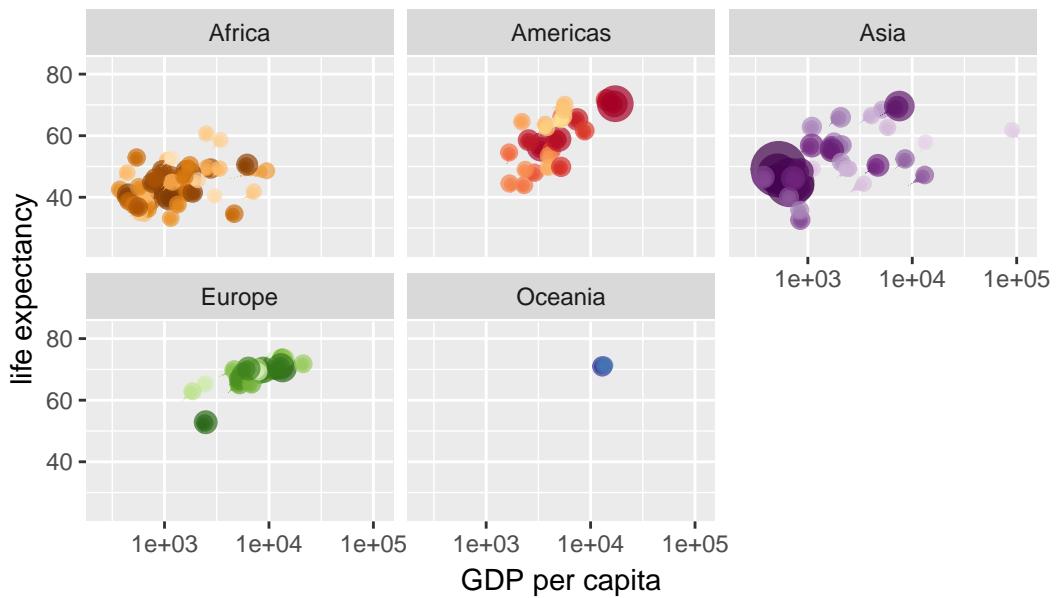
Year: 1963



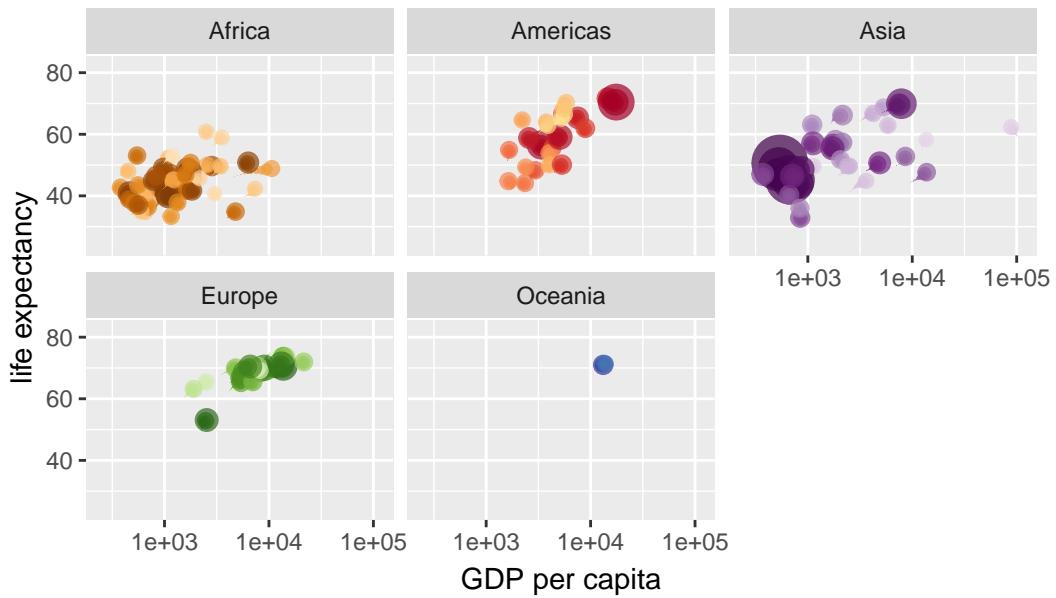
Year: 1963



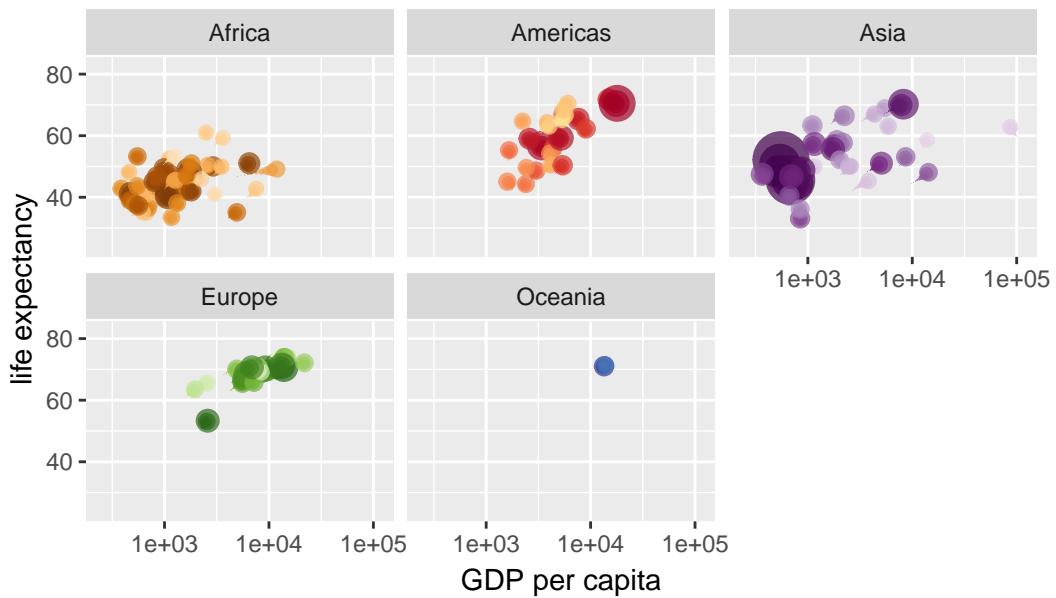
Year: 1964



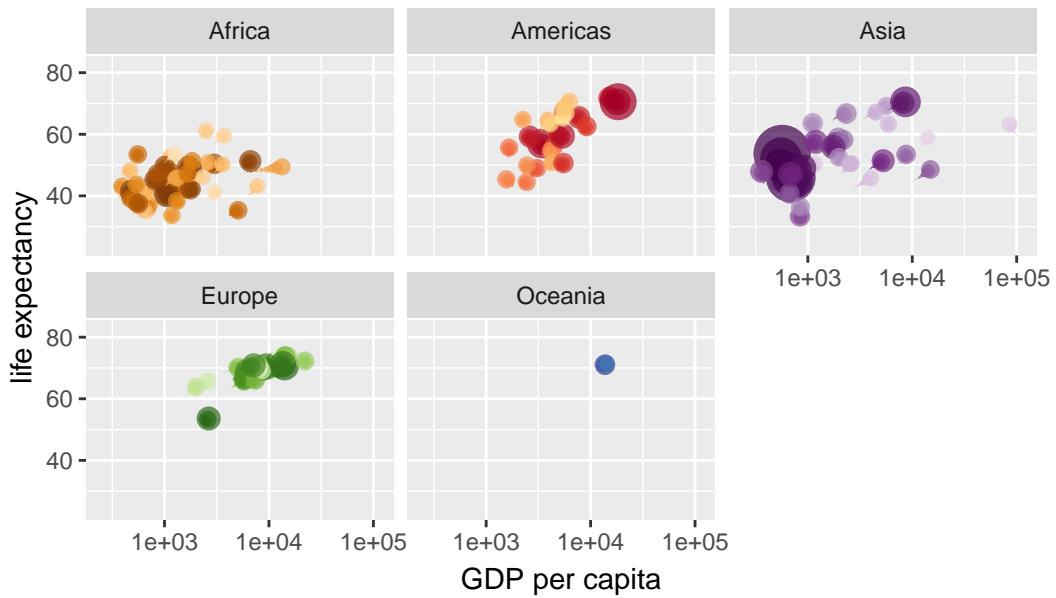
Year: 1964



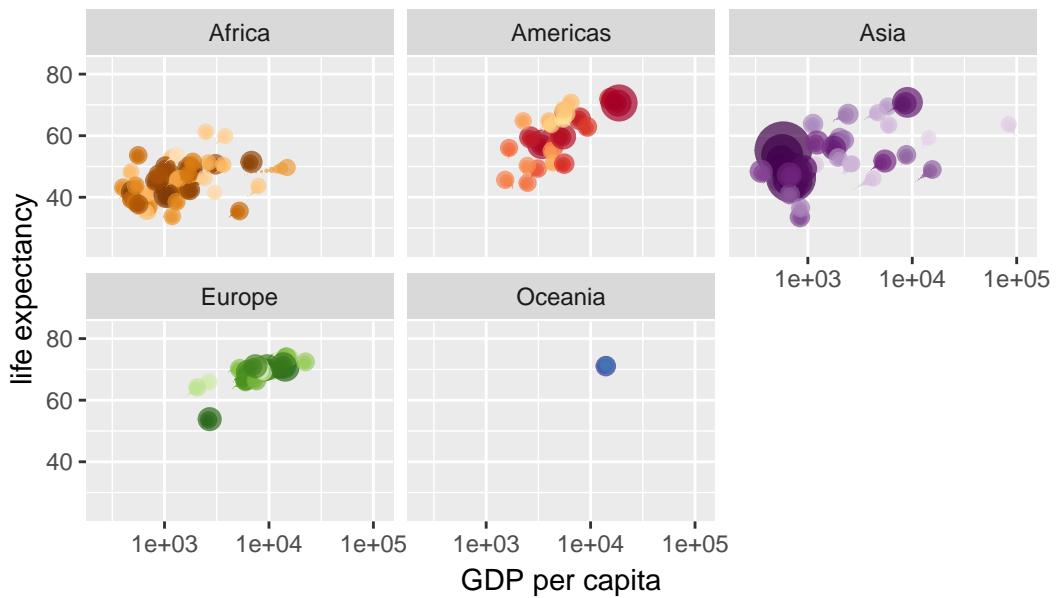
Year: 1965



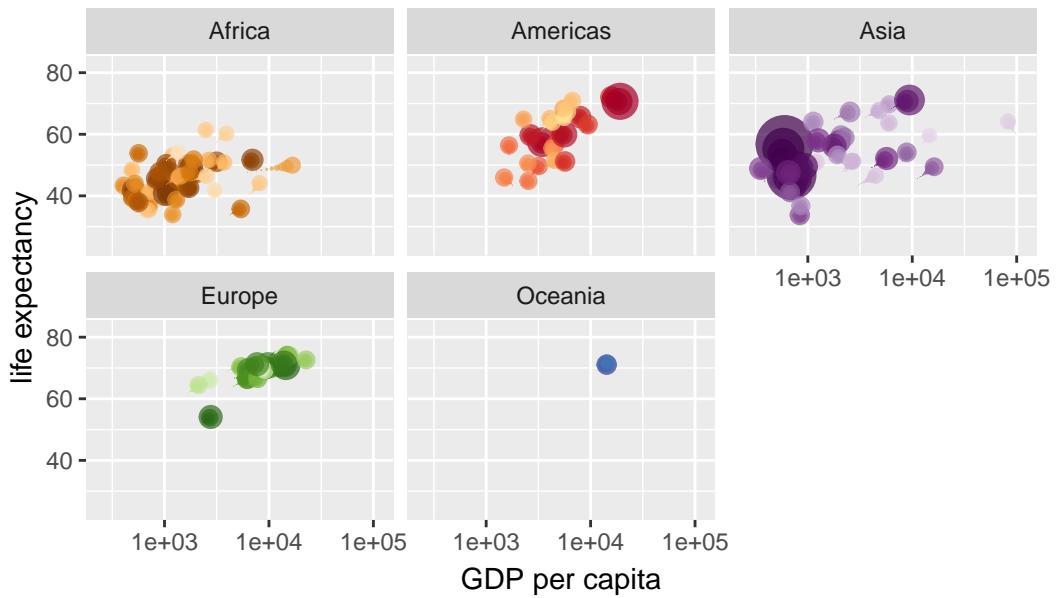
Year: 1965



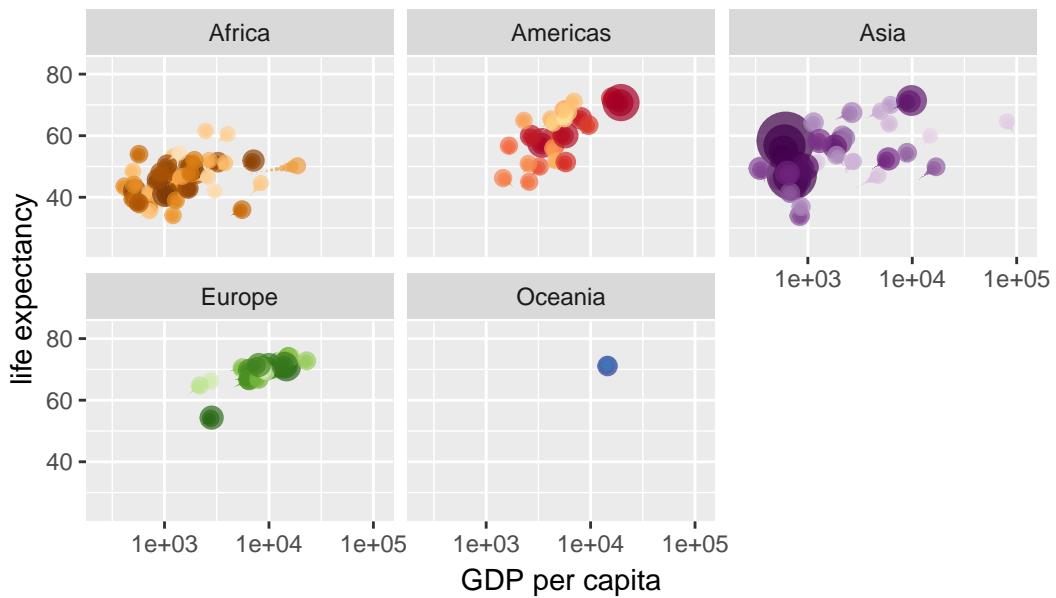
Year: 1966



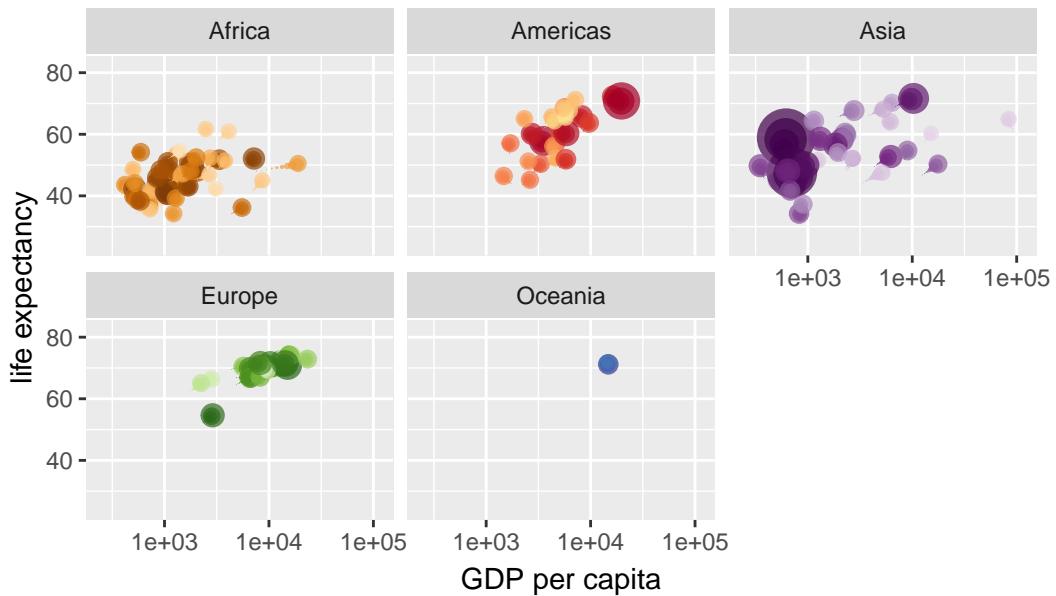
Year: 1966



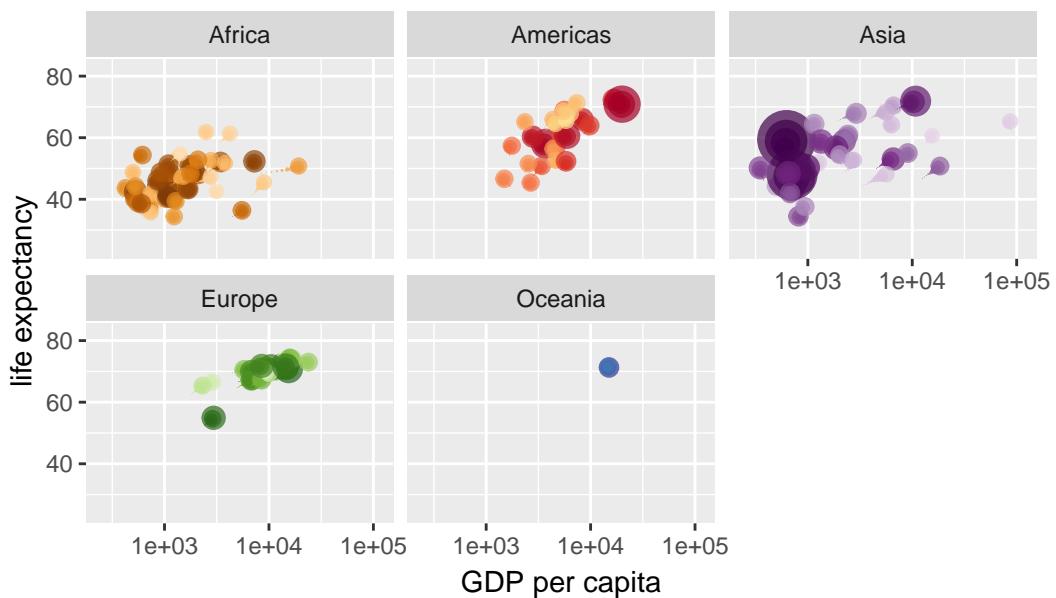
Year: 1967



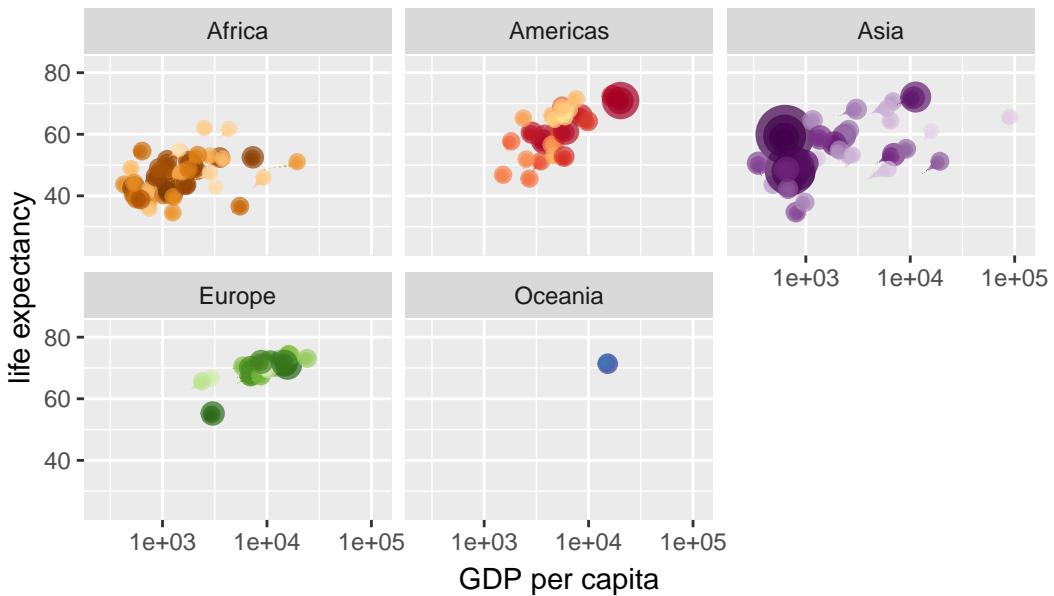
Year: 1968



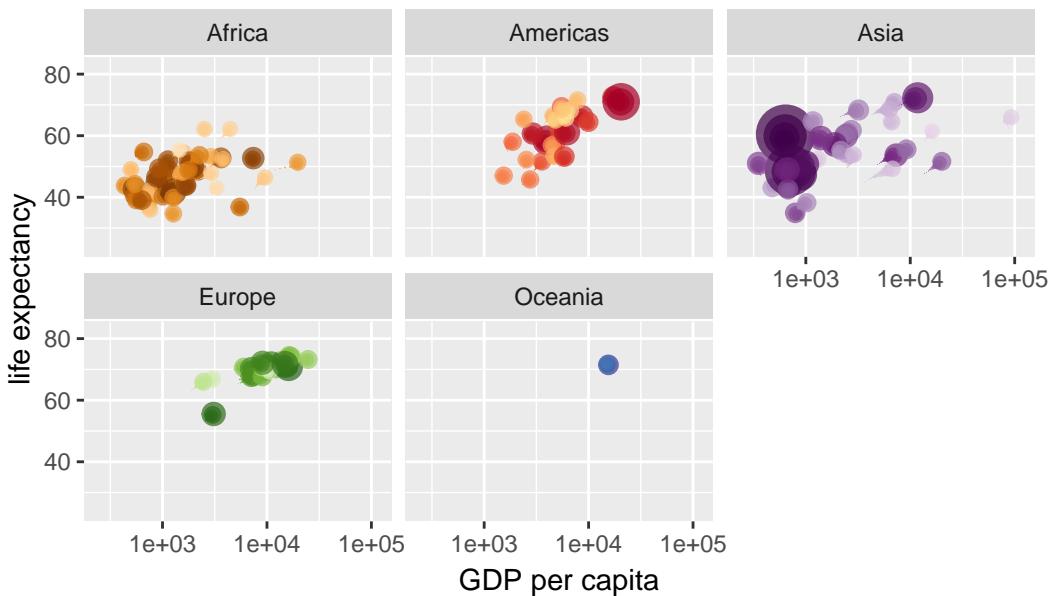
Year: 1968



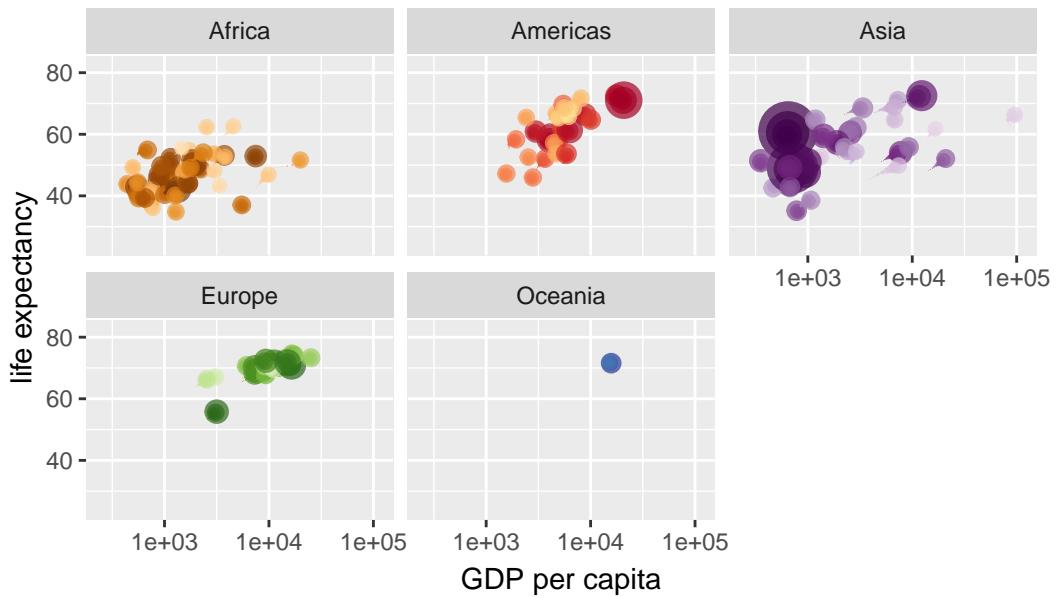
Year: 1969



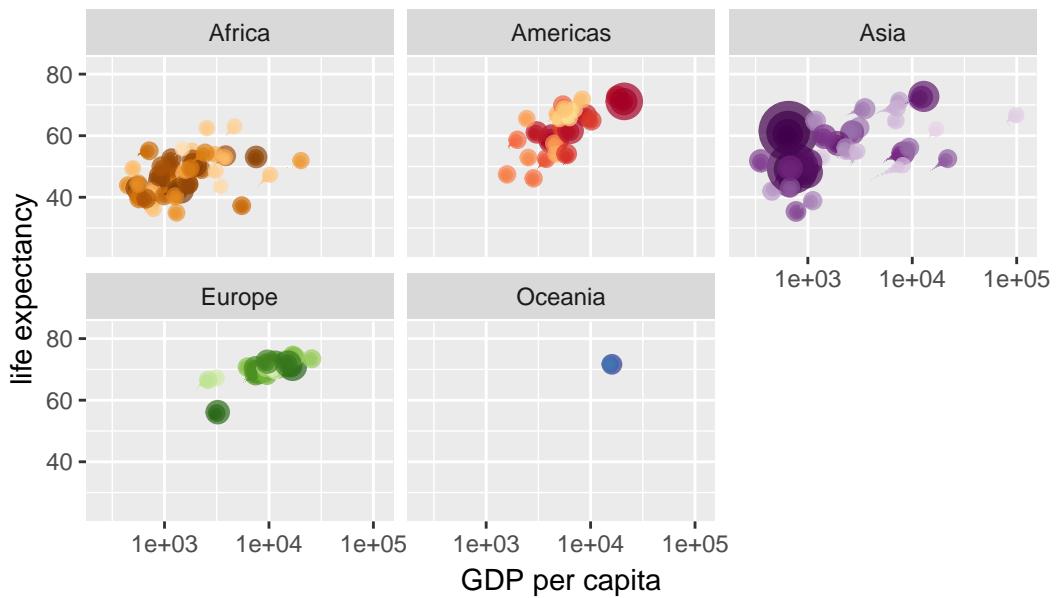
Year: 1969



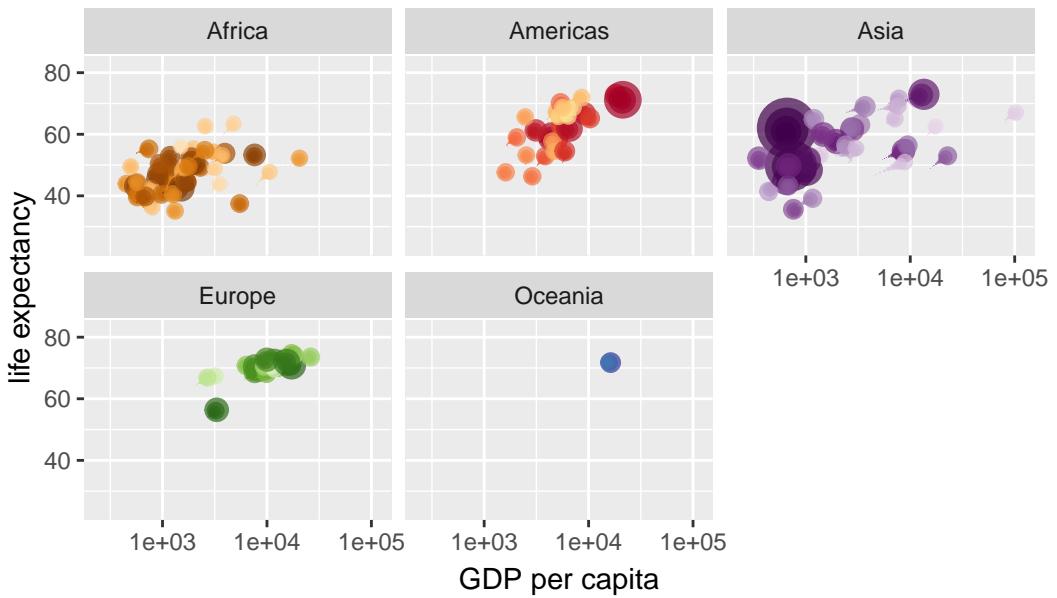
Year: 1970



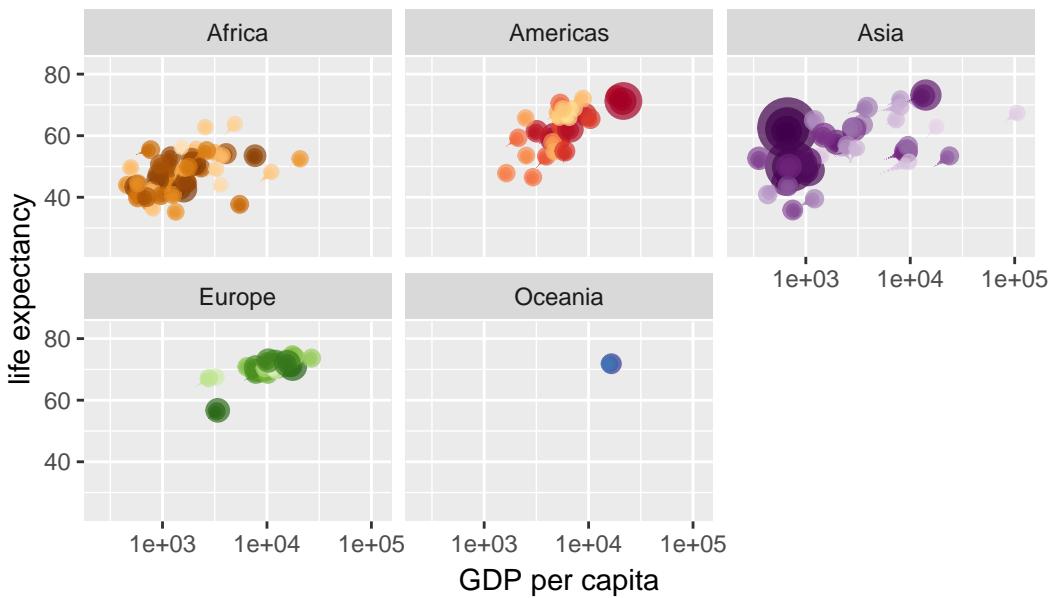
Year: 1970



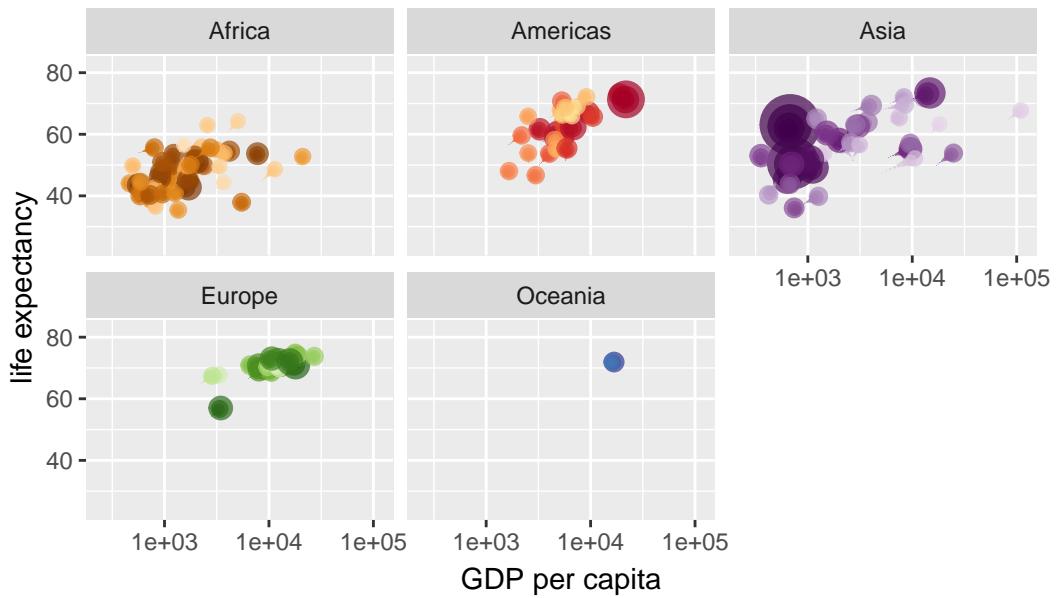
Year: 1971



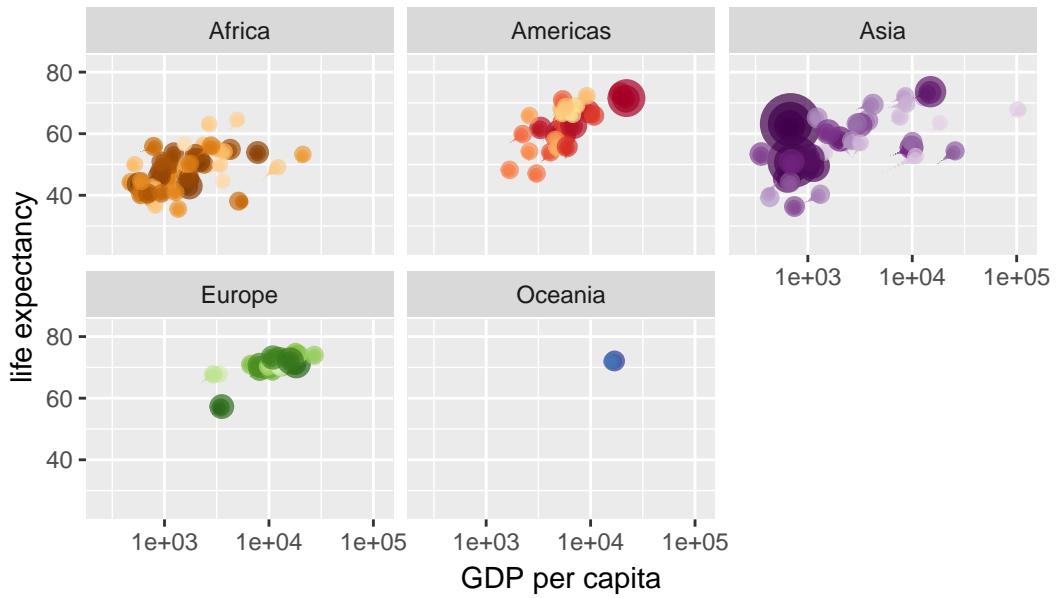
Year: 1971



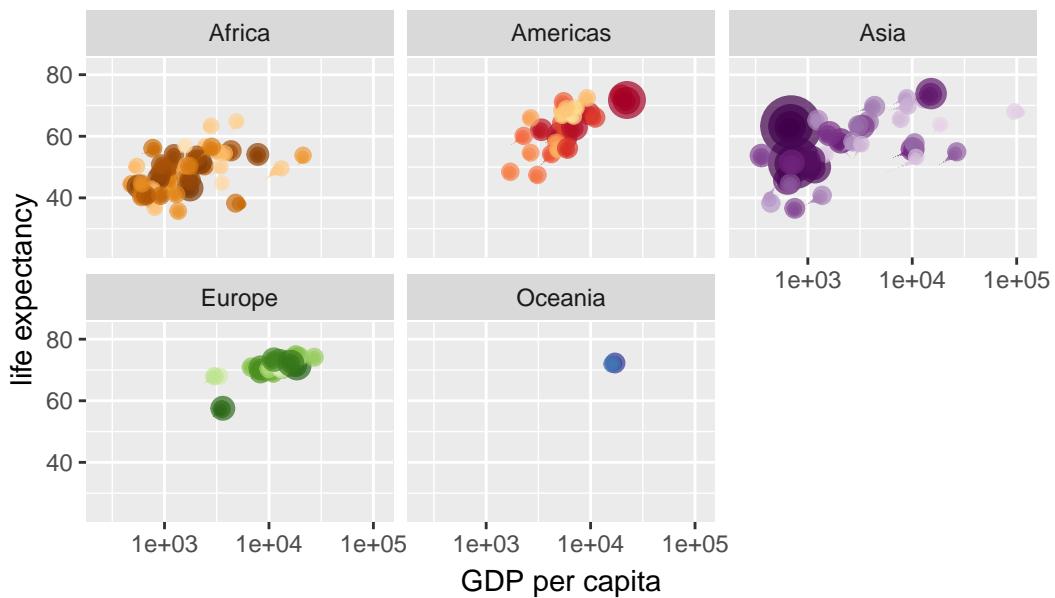
Year: 1972



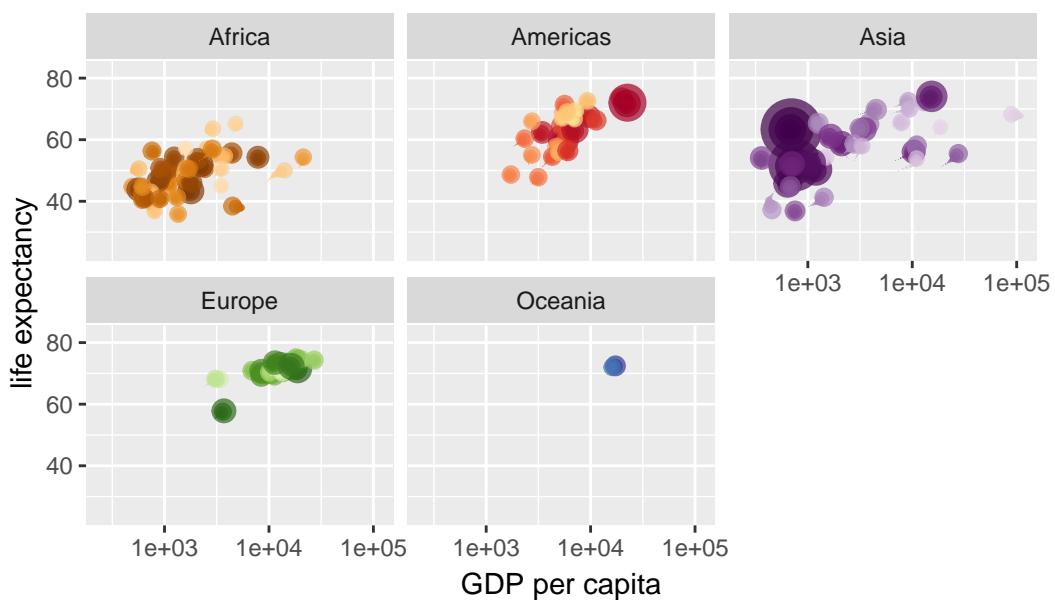
Year: 1973



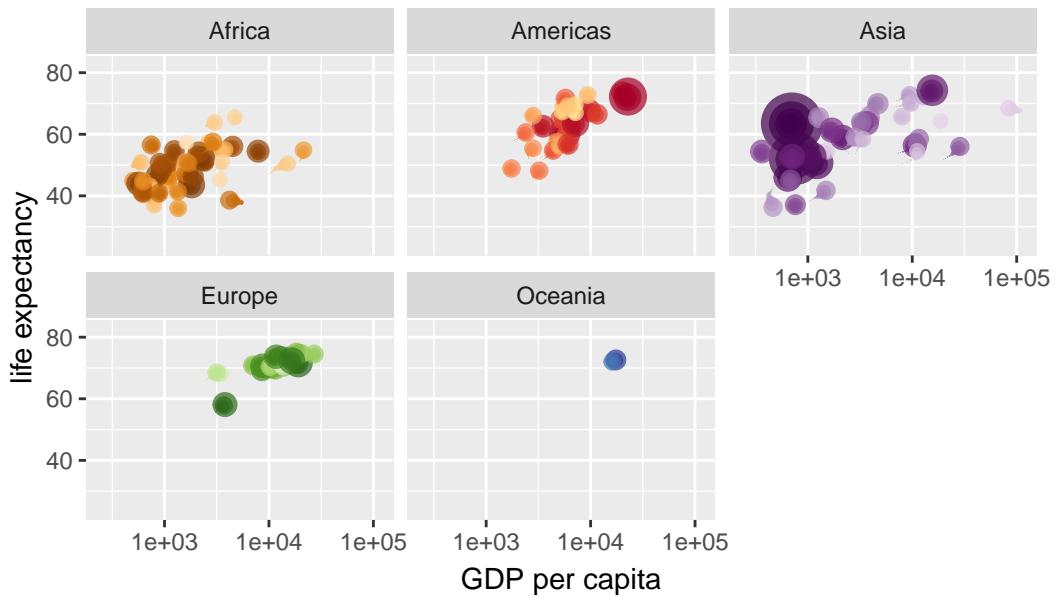
Year: 1973



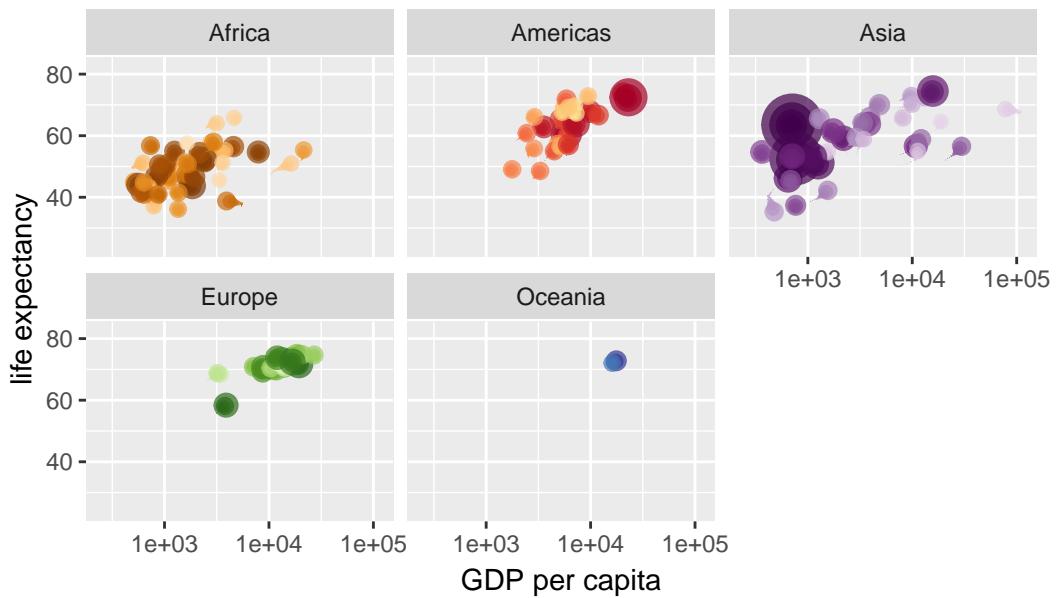
Year: 1974



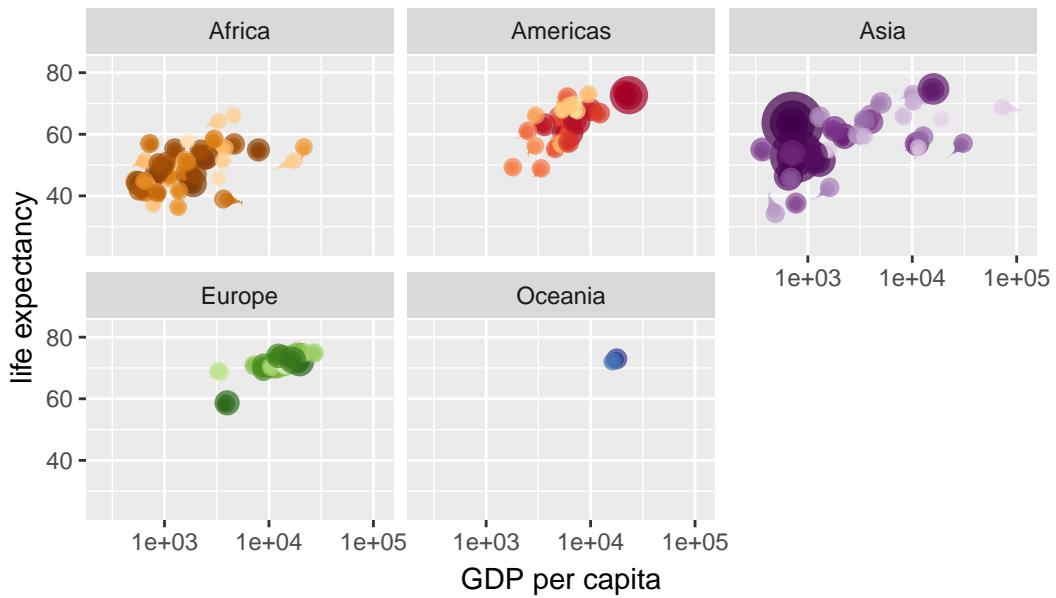
Year: 1974



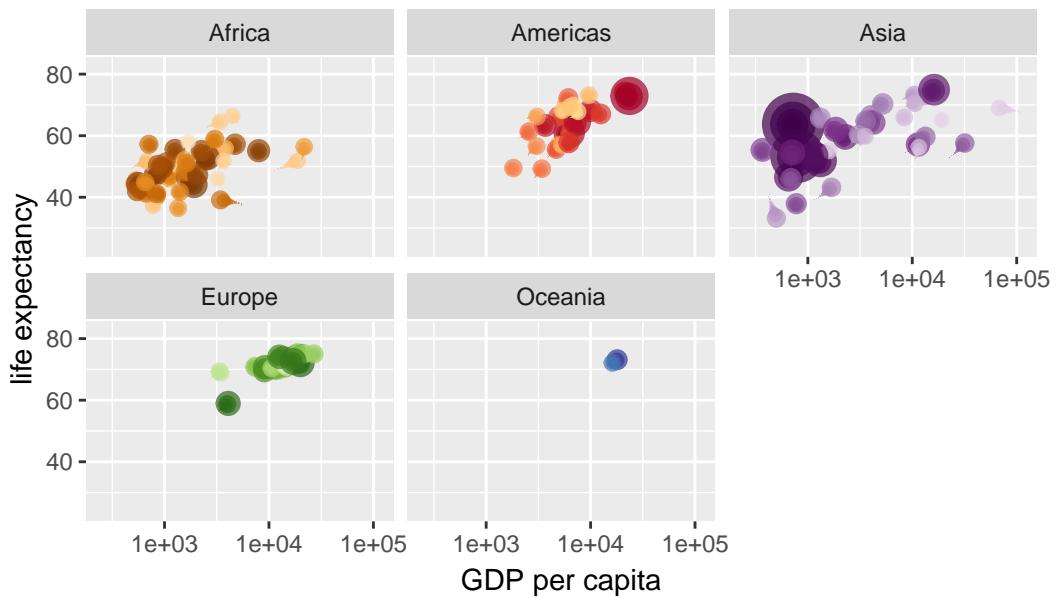
Year: 1975



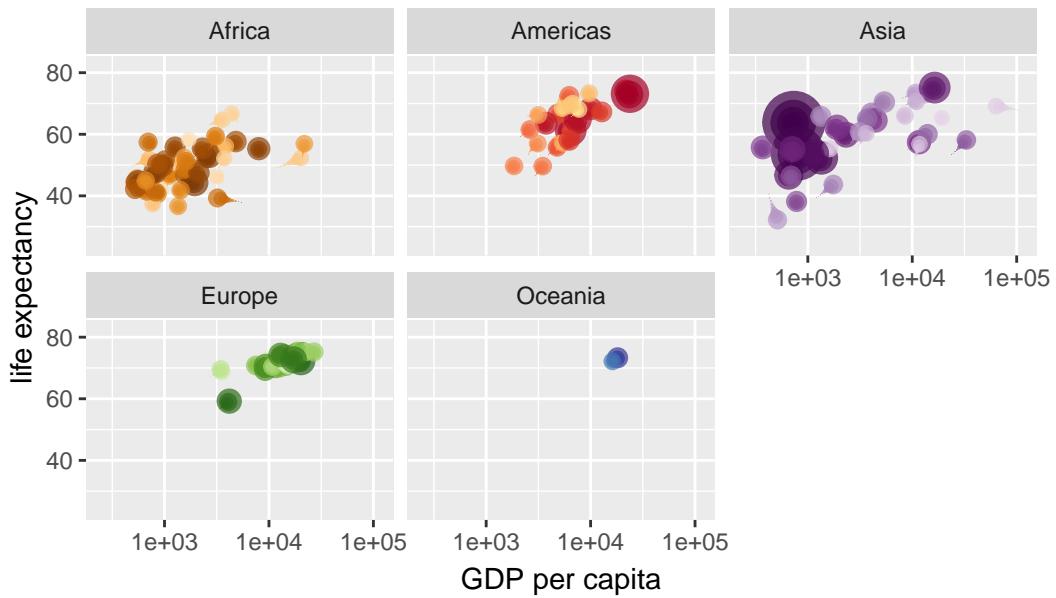
Year: 1975



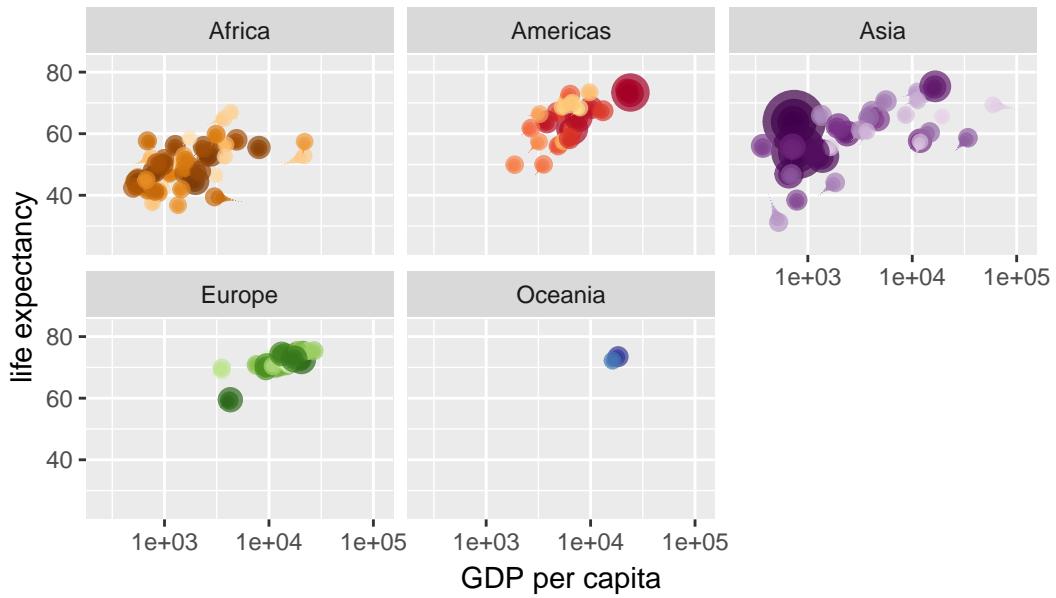
Year: 1976



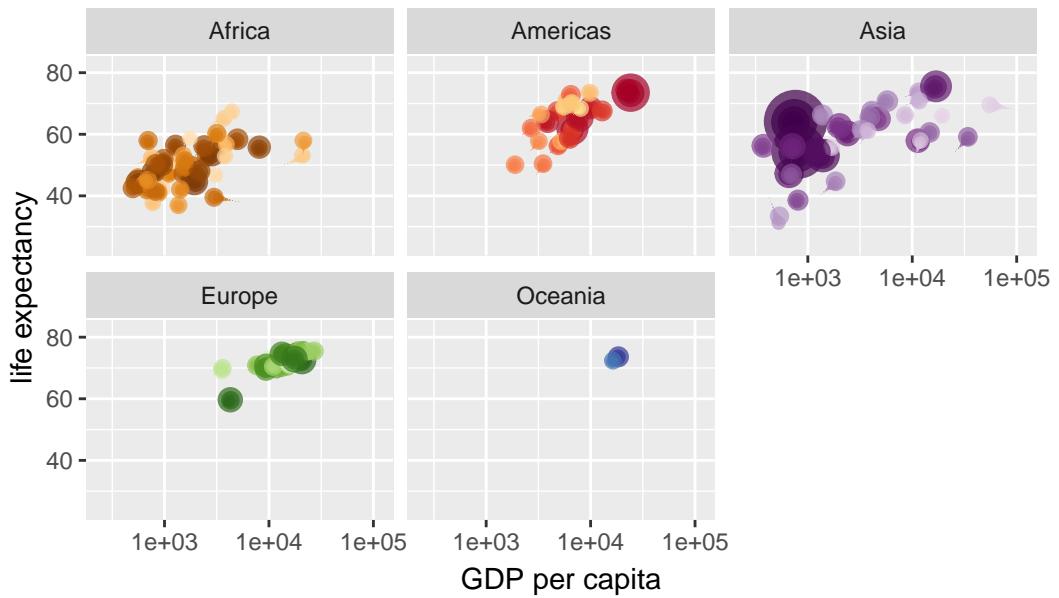
Year: 1976



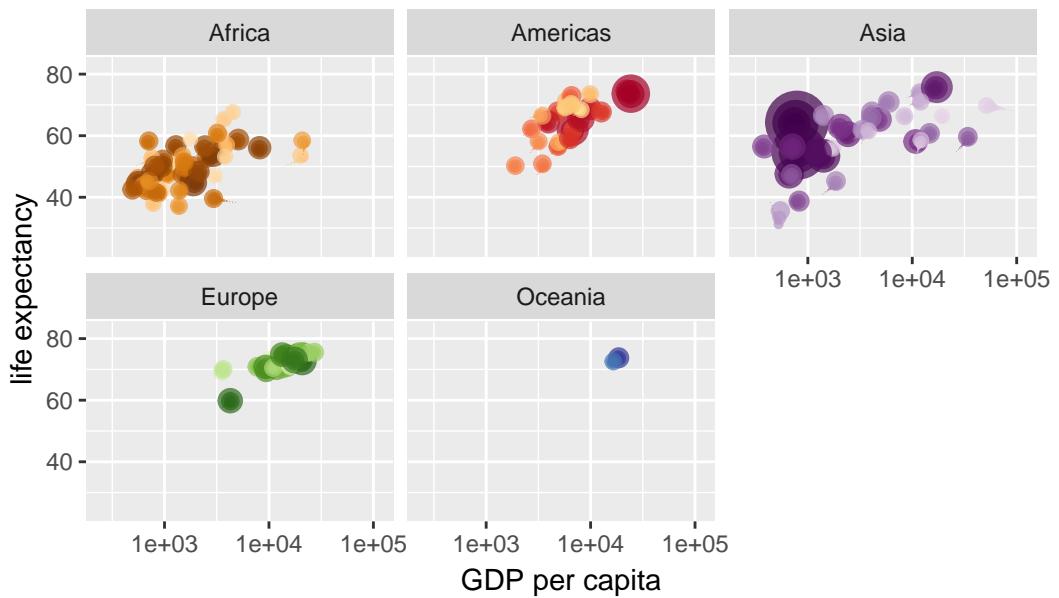
Year: 1977



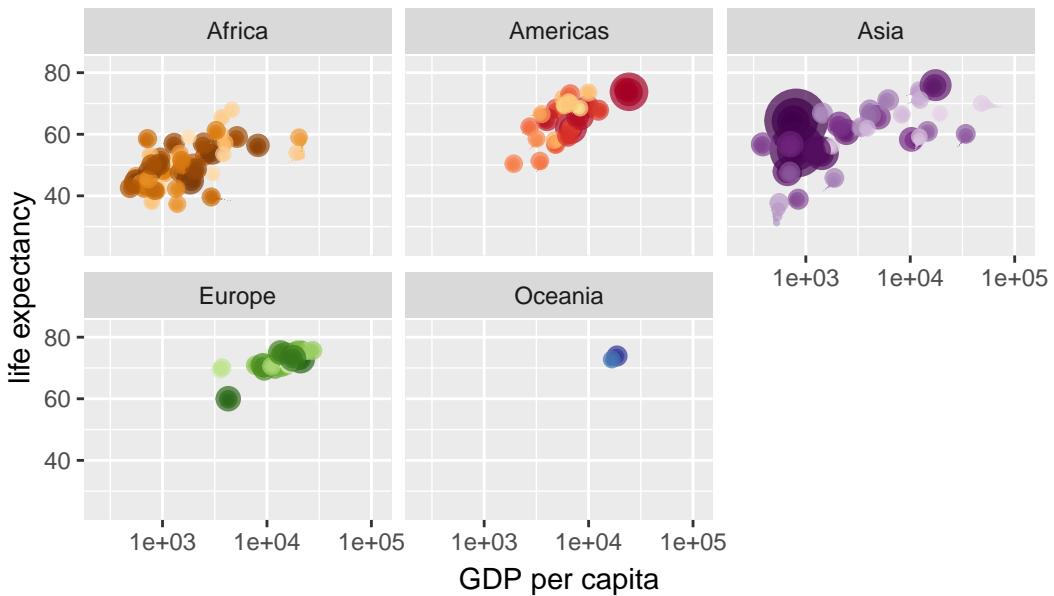
Year: 1978



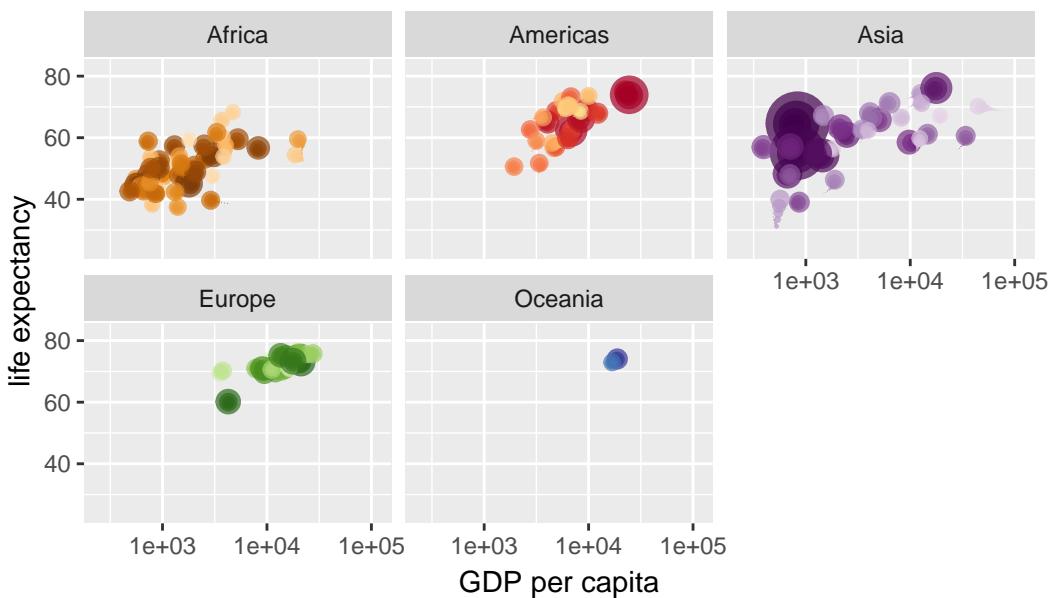
Year: 1978



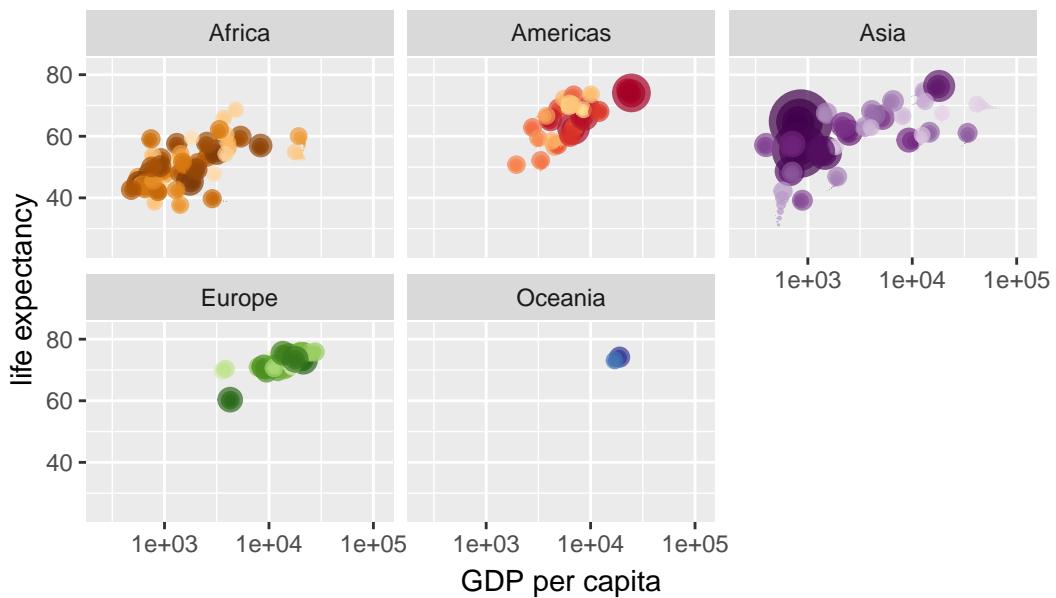
Year: 1979



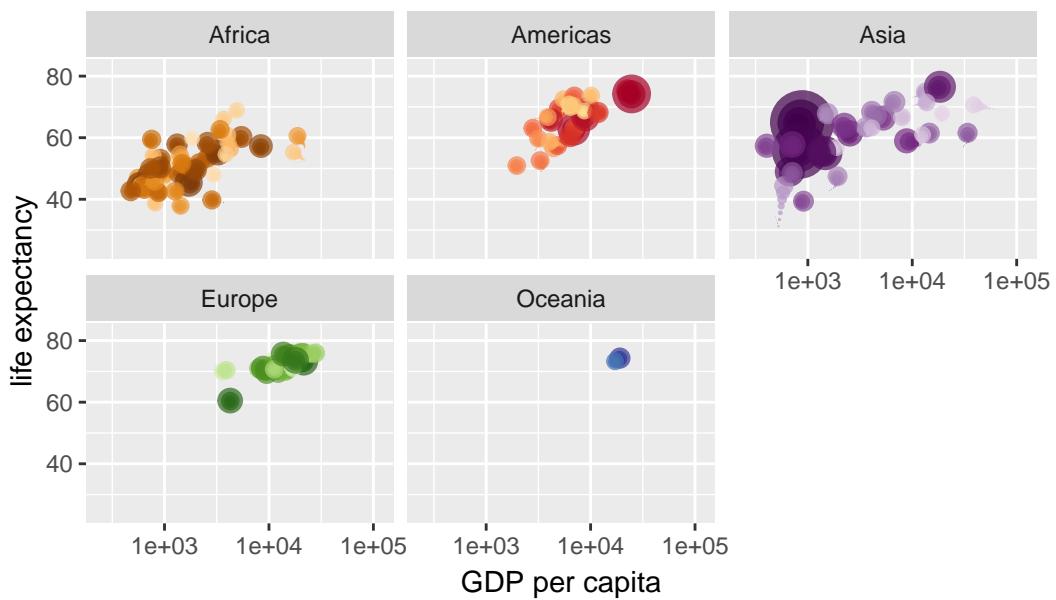
Year: 1979



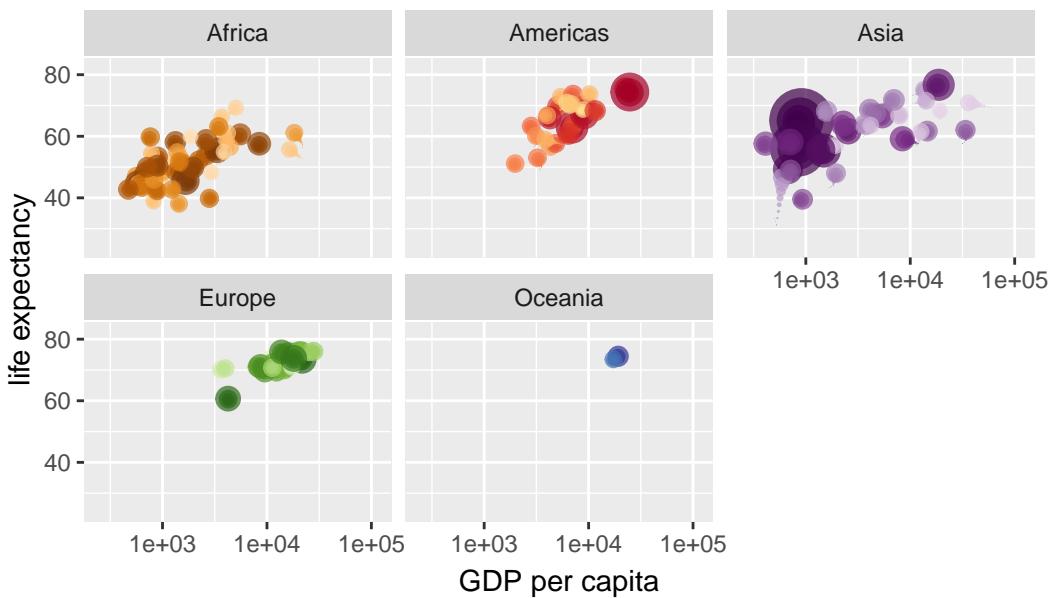
Year: 1980



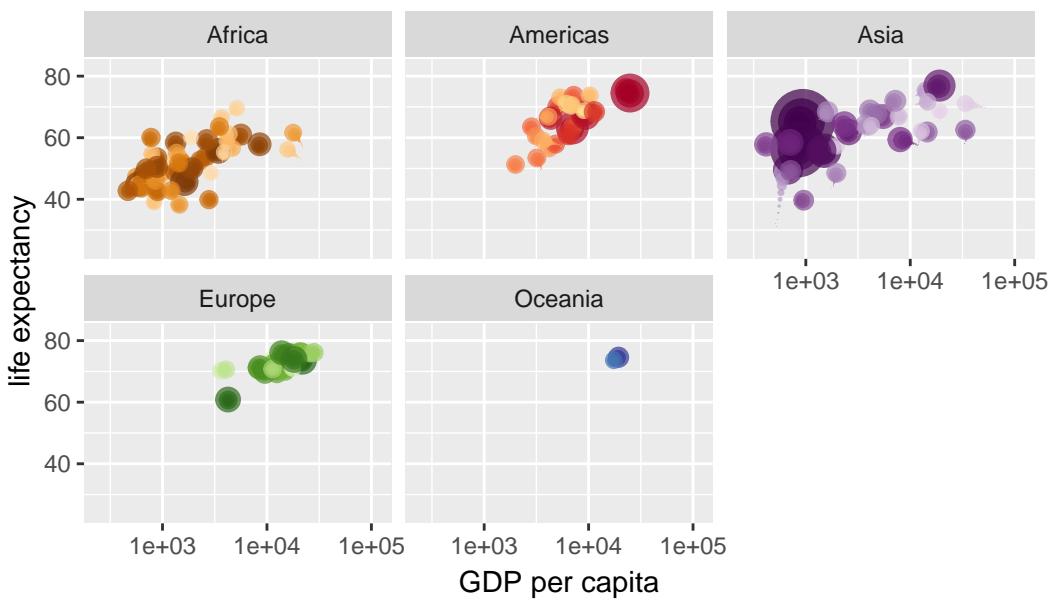
Year: 1980



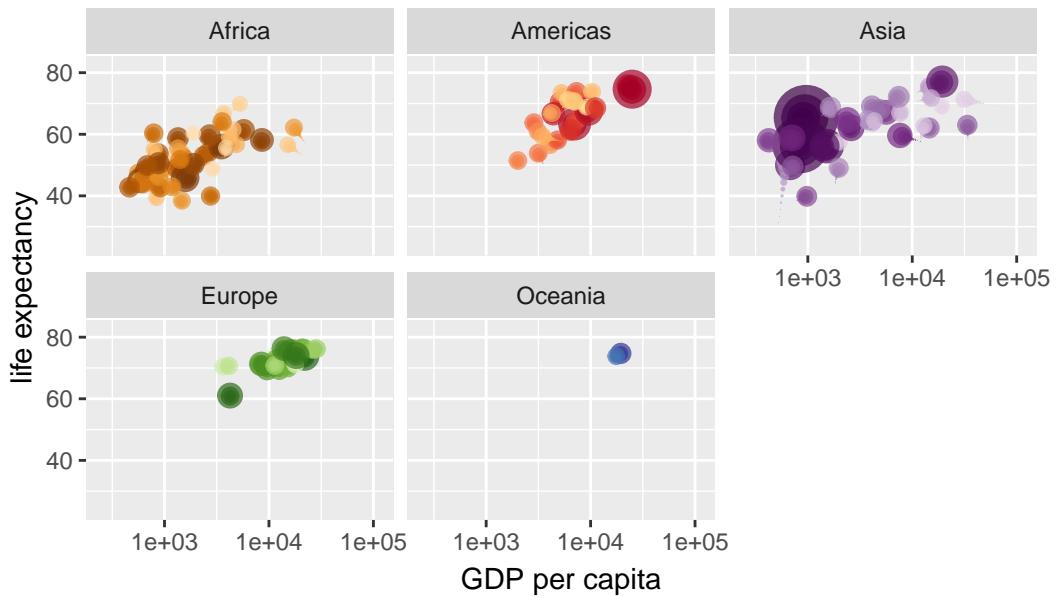
Year: 1981



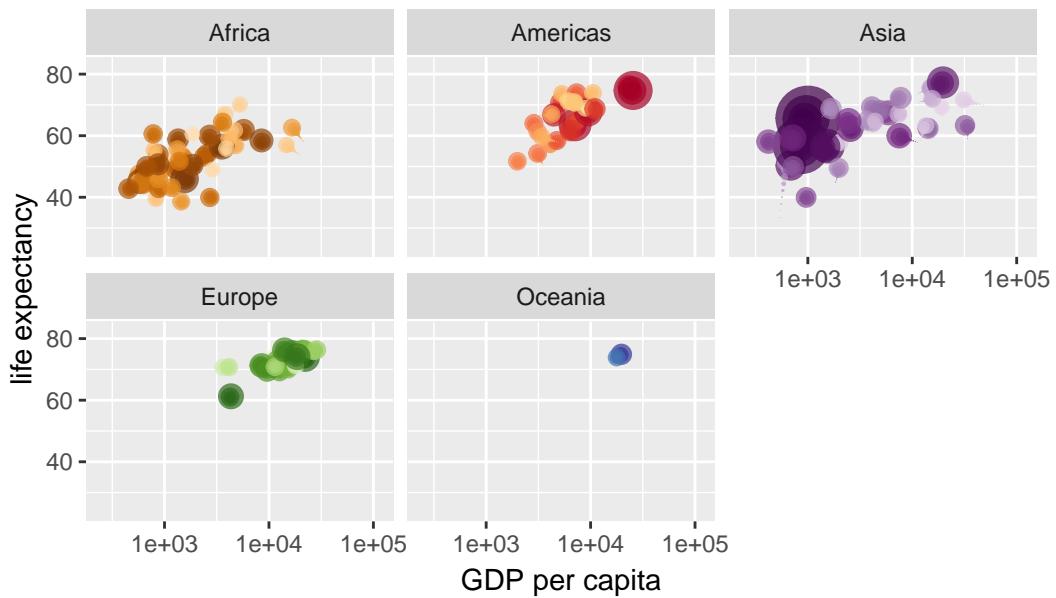
Year: 1981



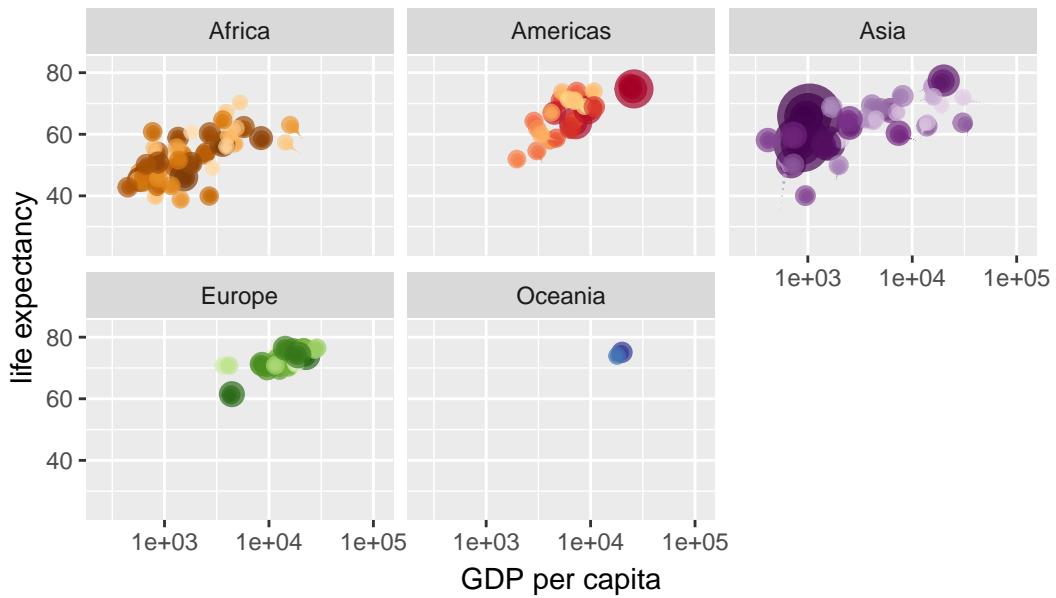
Year: 1982



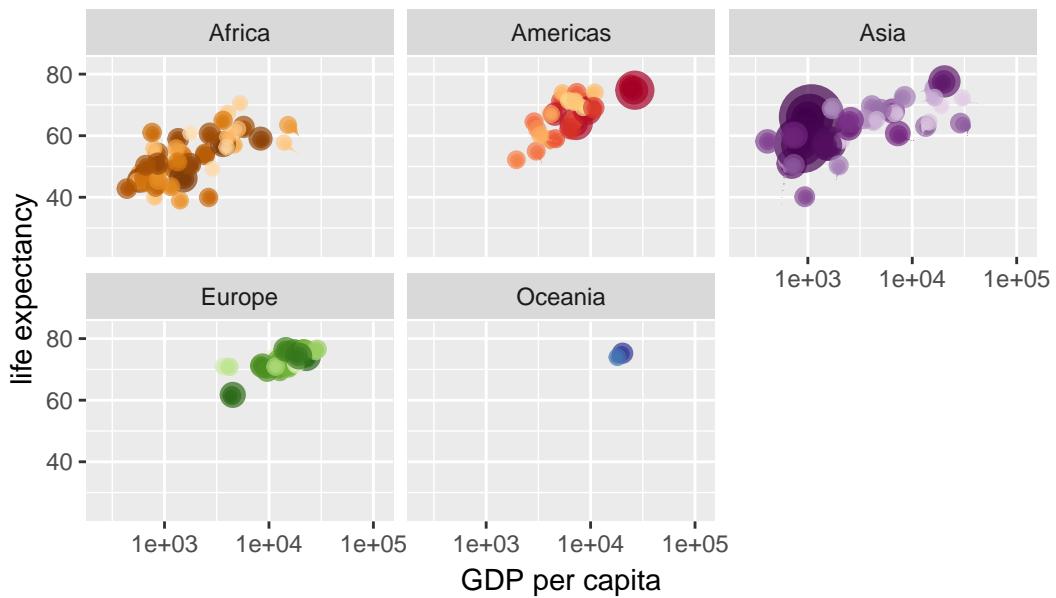
Year: 1983



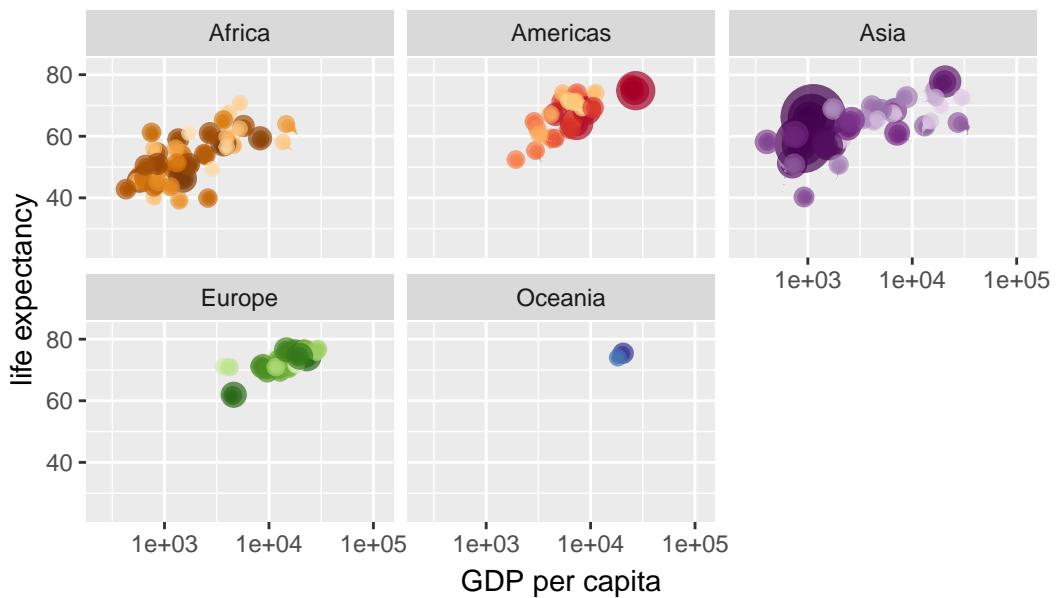
Year: 1983



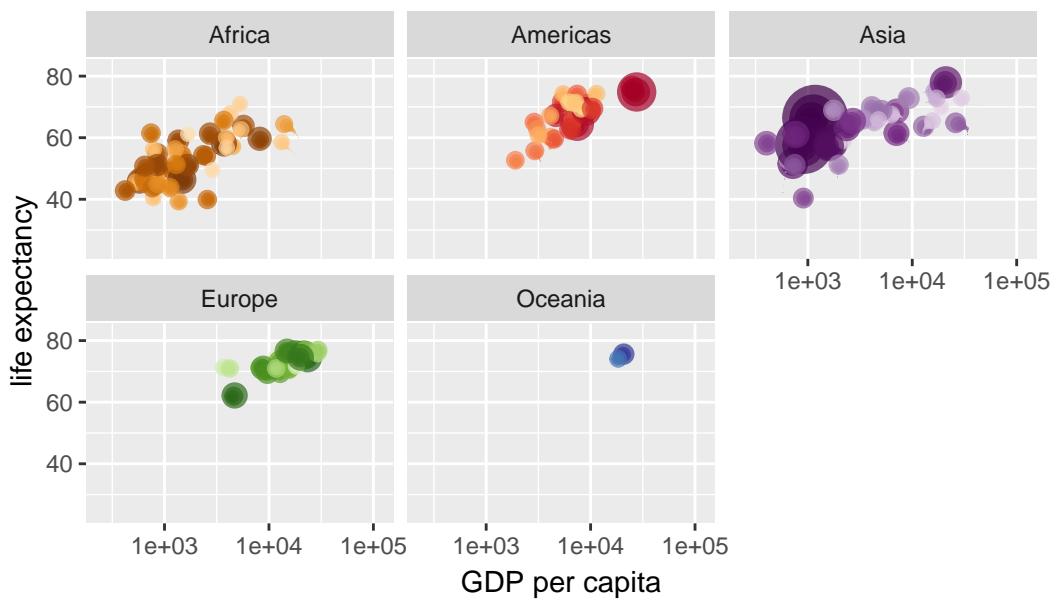
Year: 1984



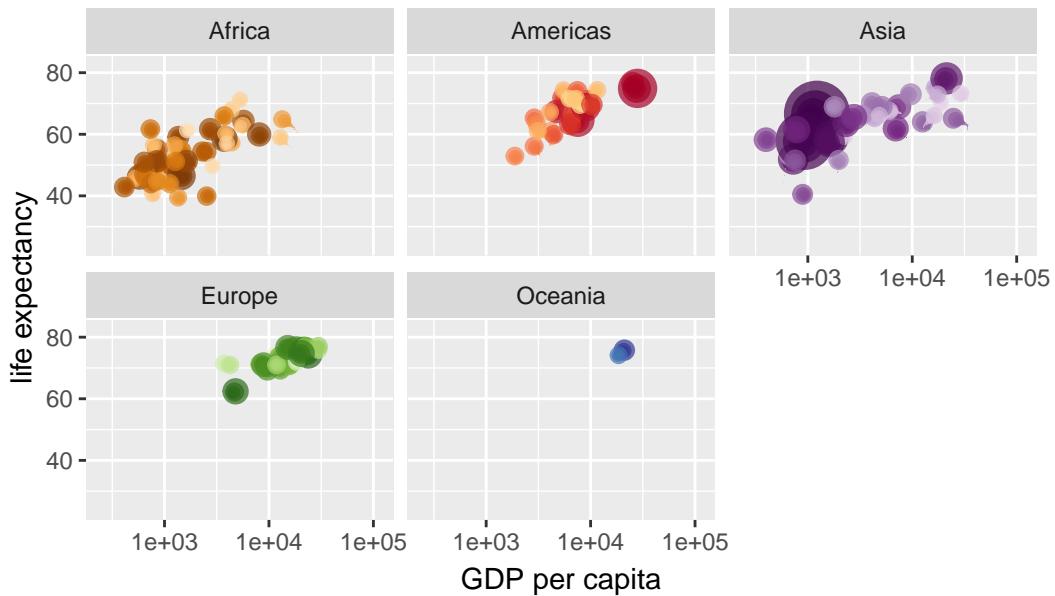
Year: 1984



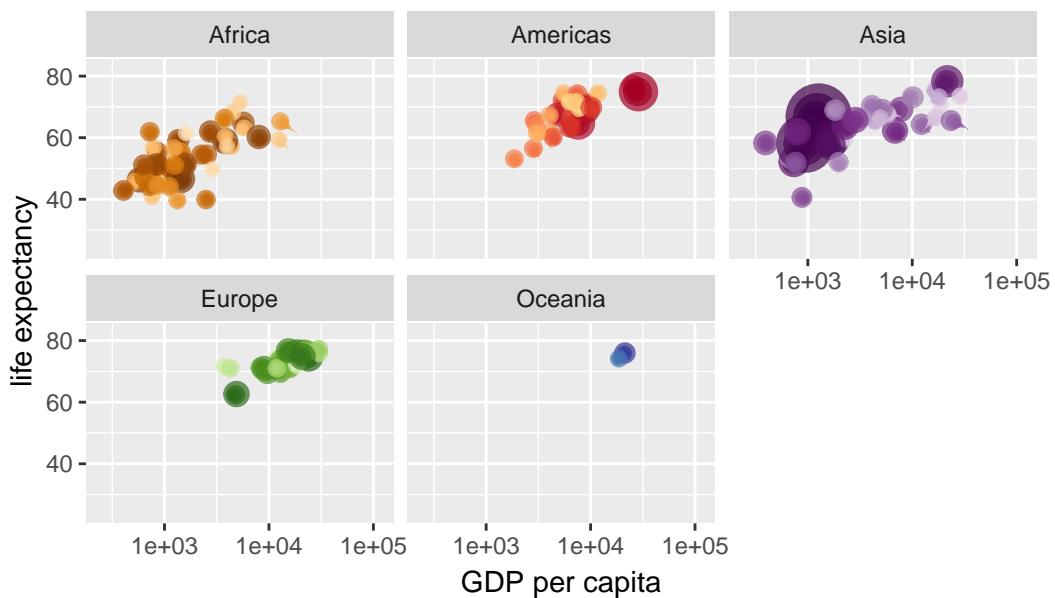
Year: 1985



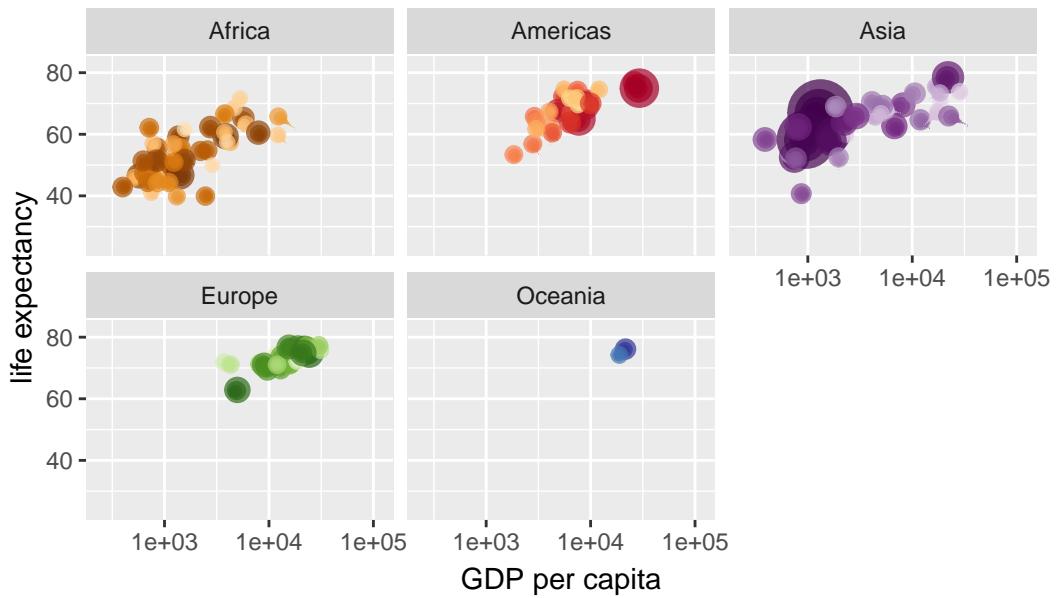
Year: 1985



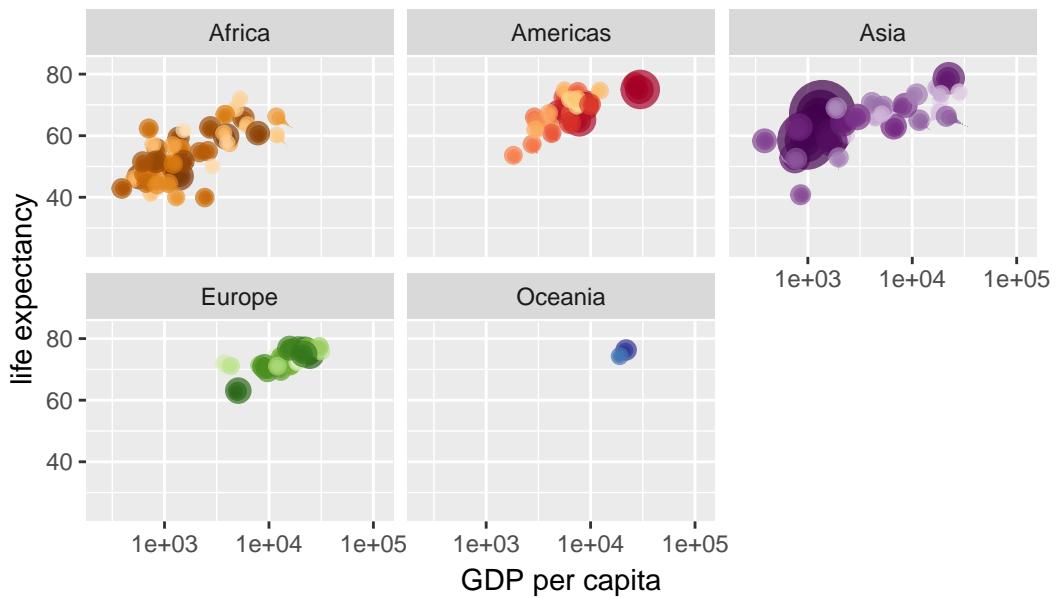
Year: 1986



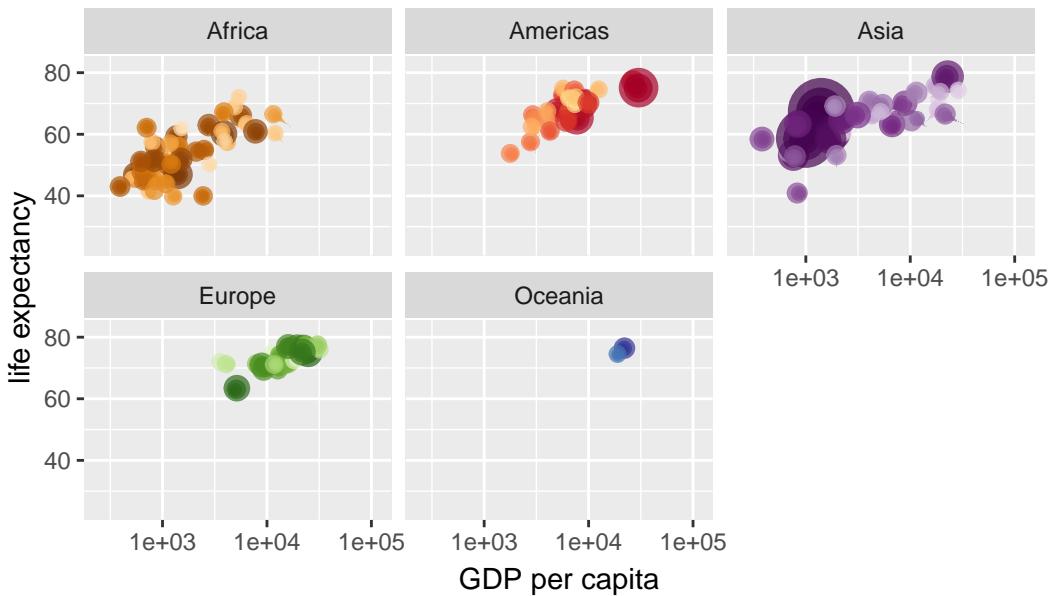
Year: 1986



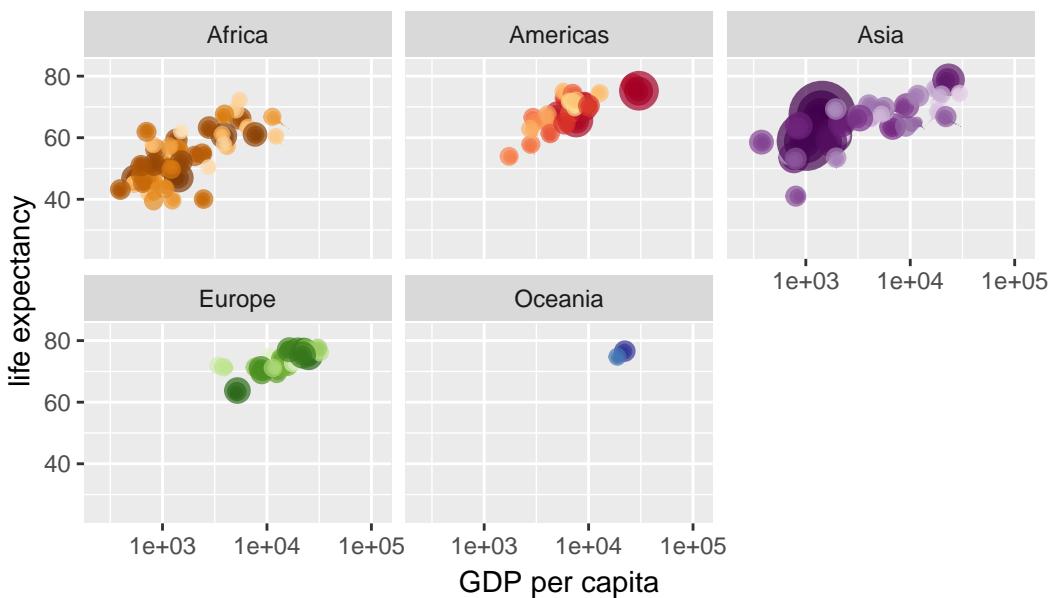
Year: 1987



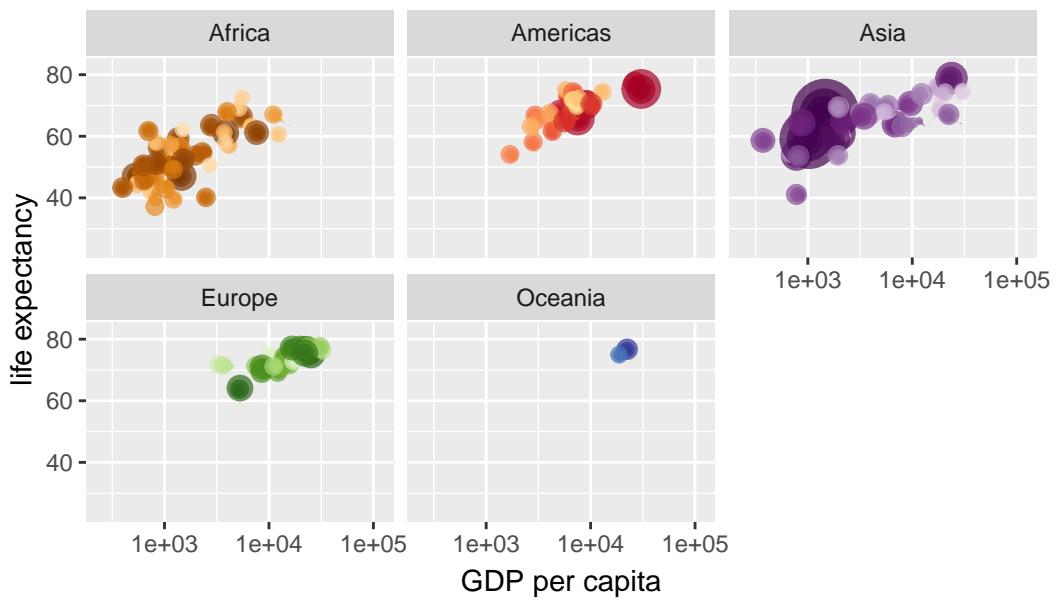
Year: 1988



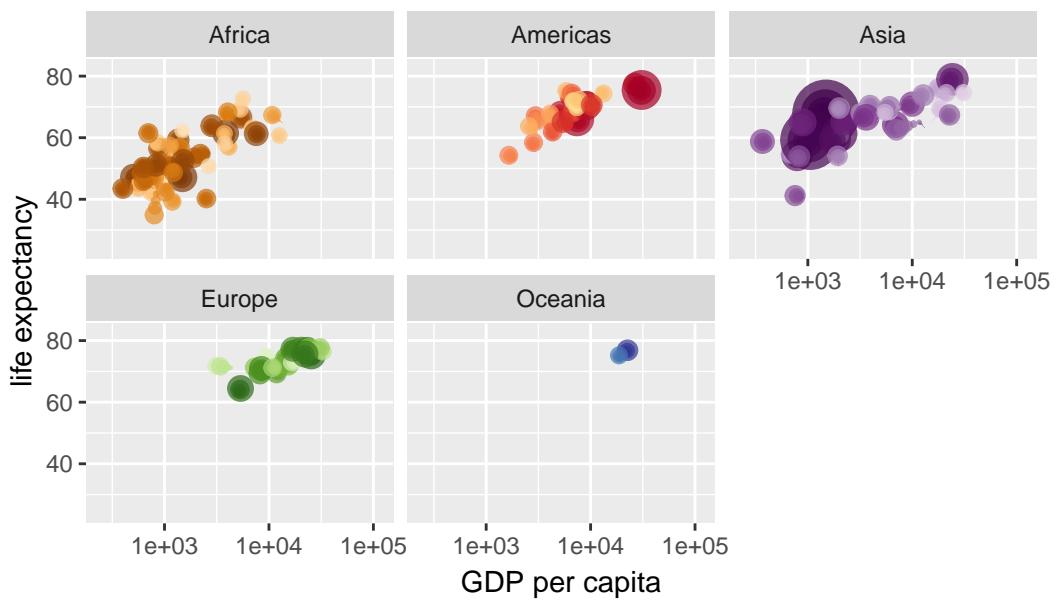
Year: 1988



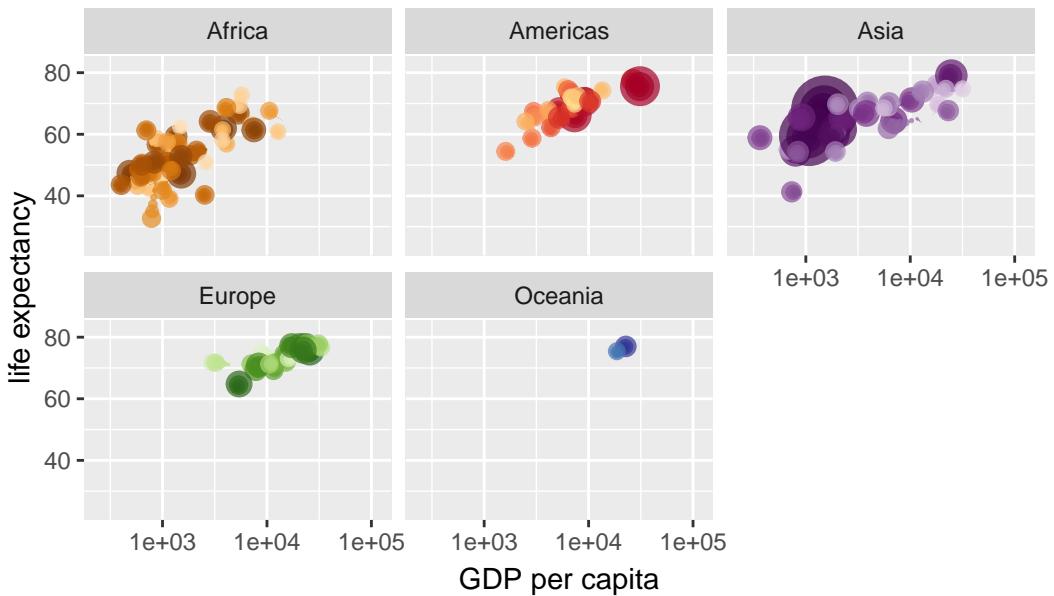
Year: 1989



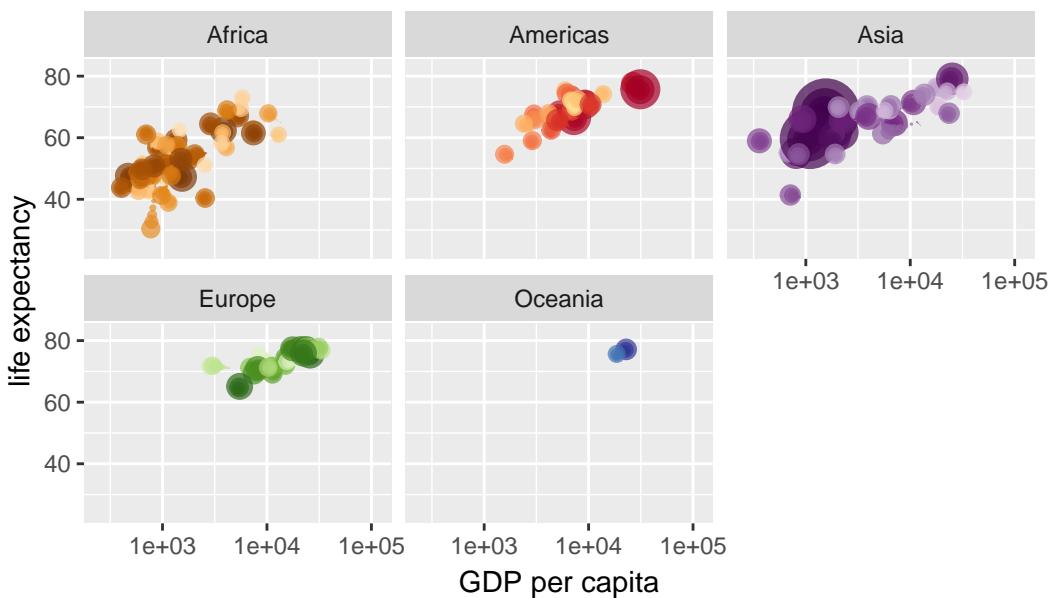
Year: 1989



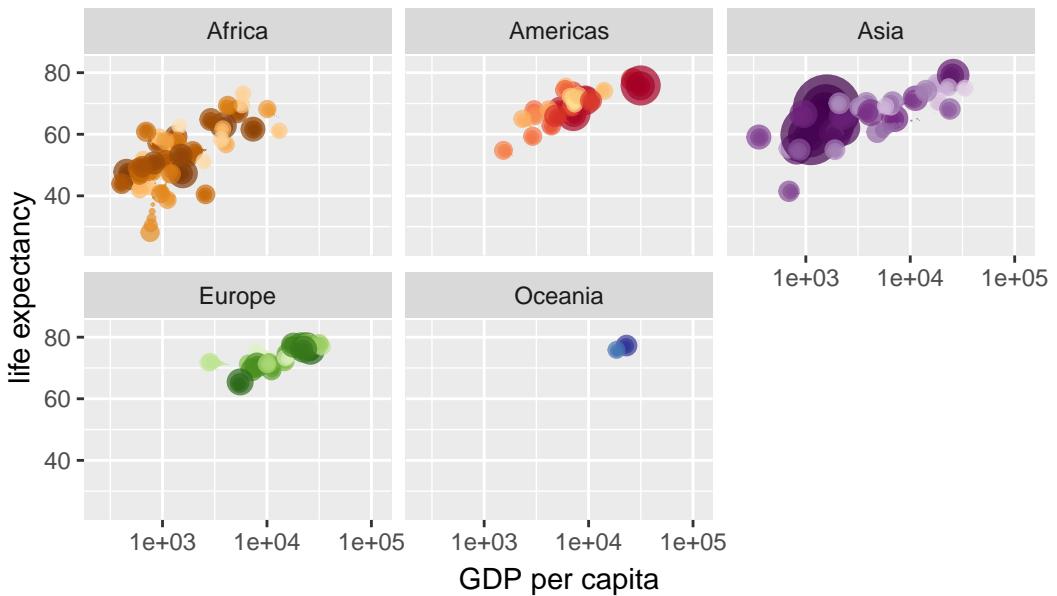
Year: 1990



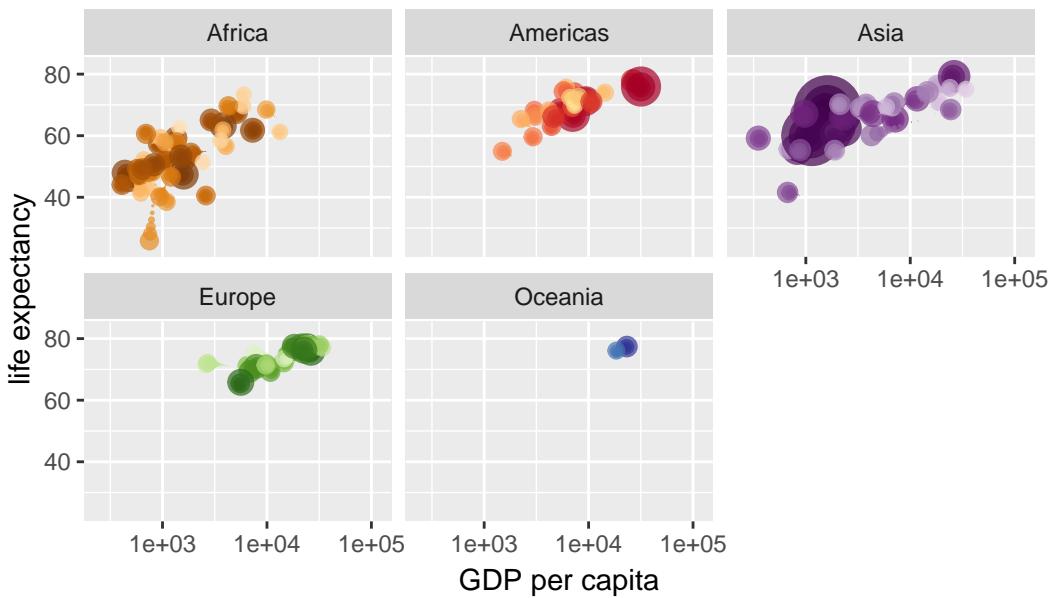
Year: 1990



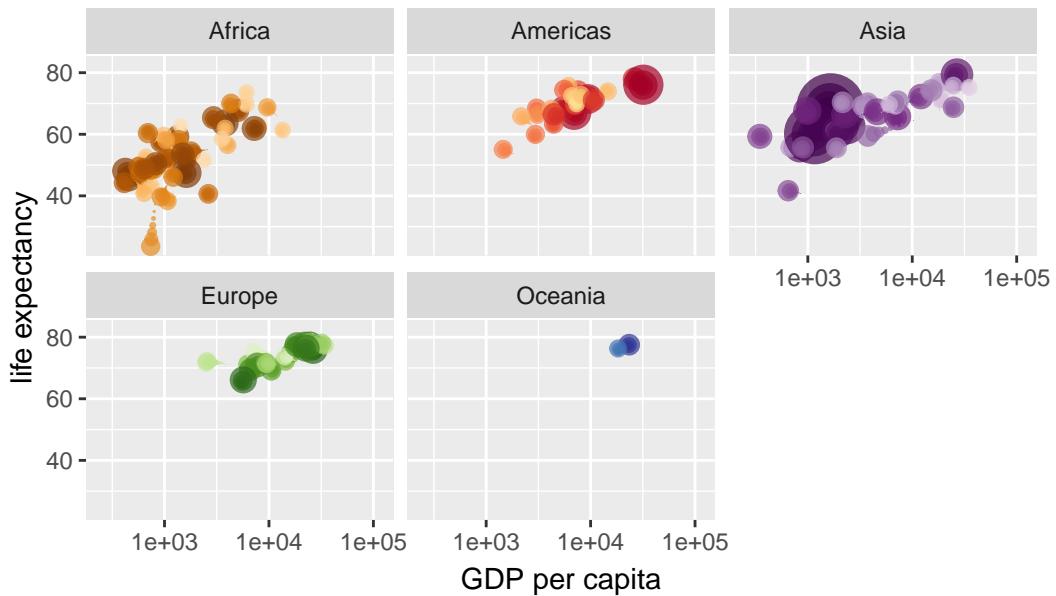
Year: 1991



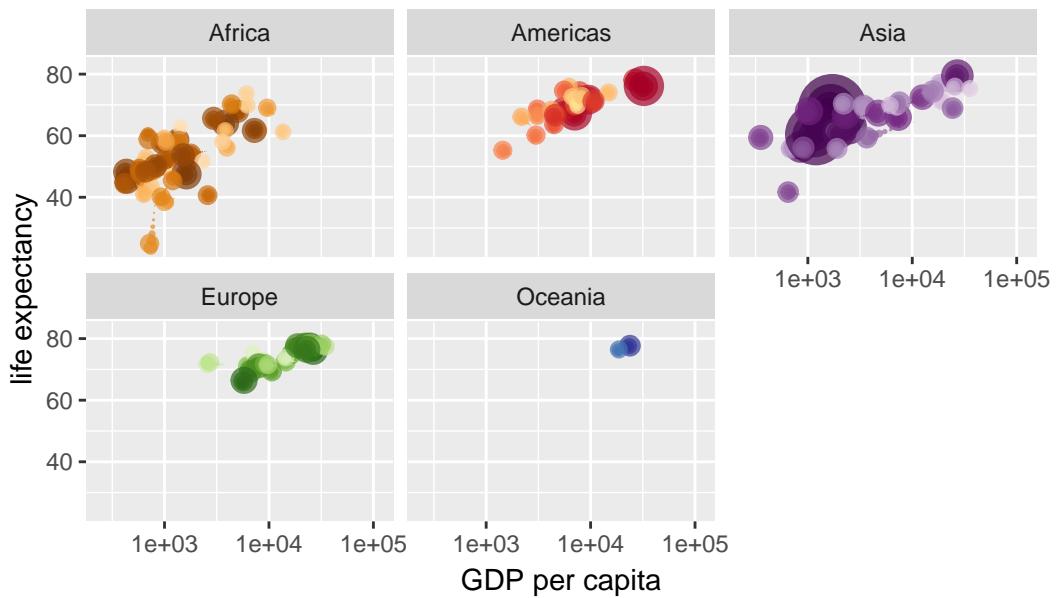
Year: 1991



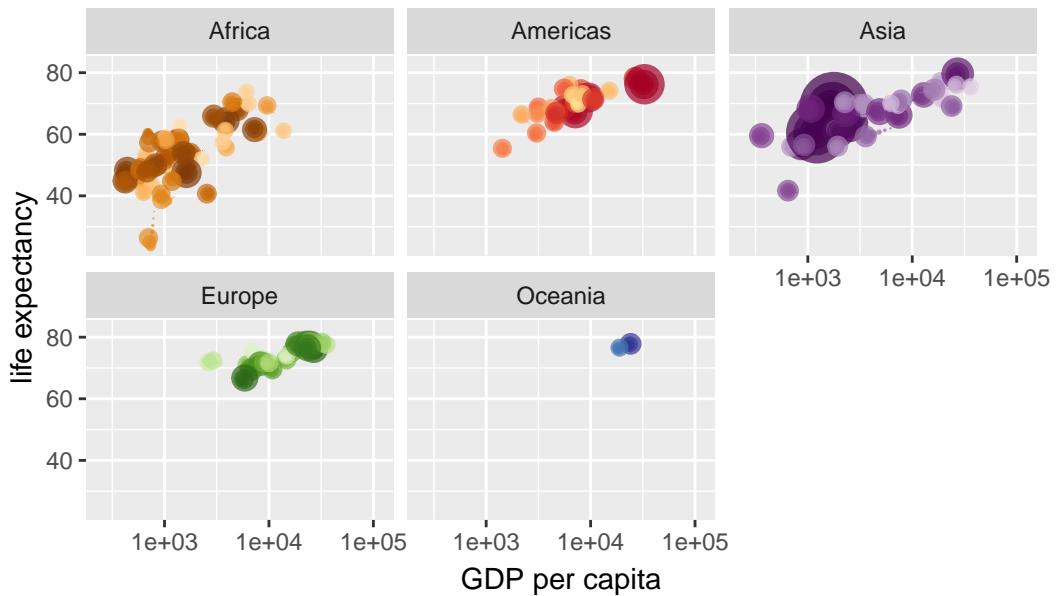
Year: 1992



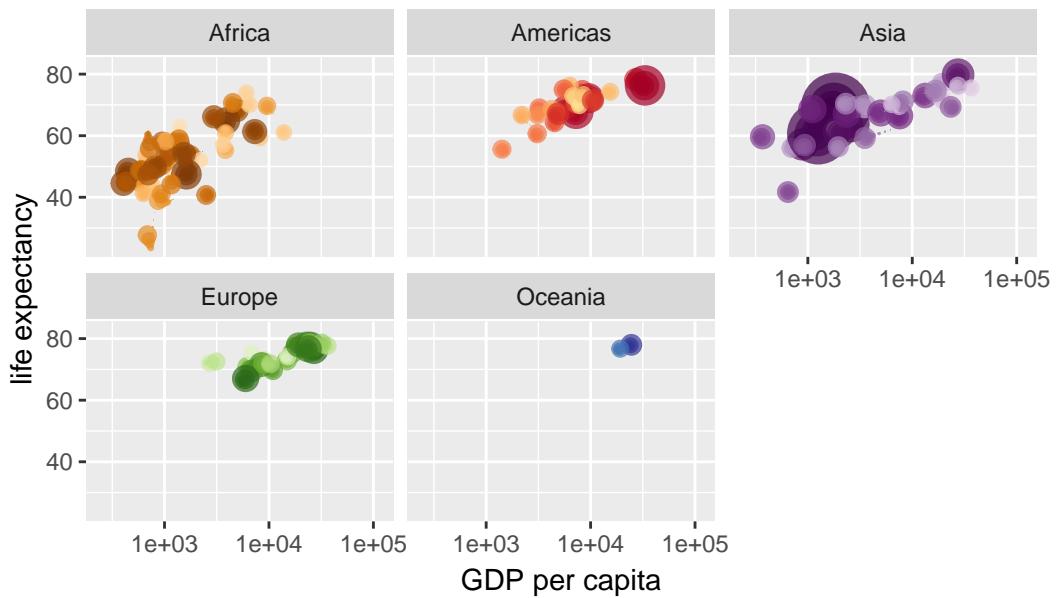
Year: 1993



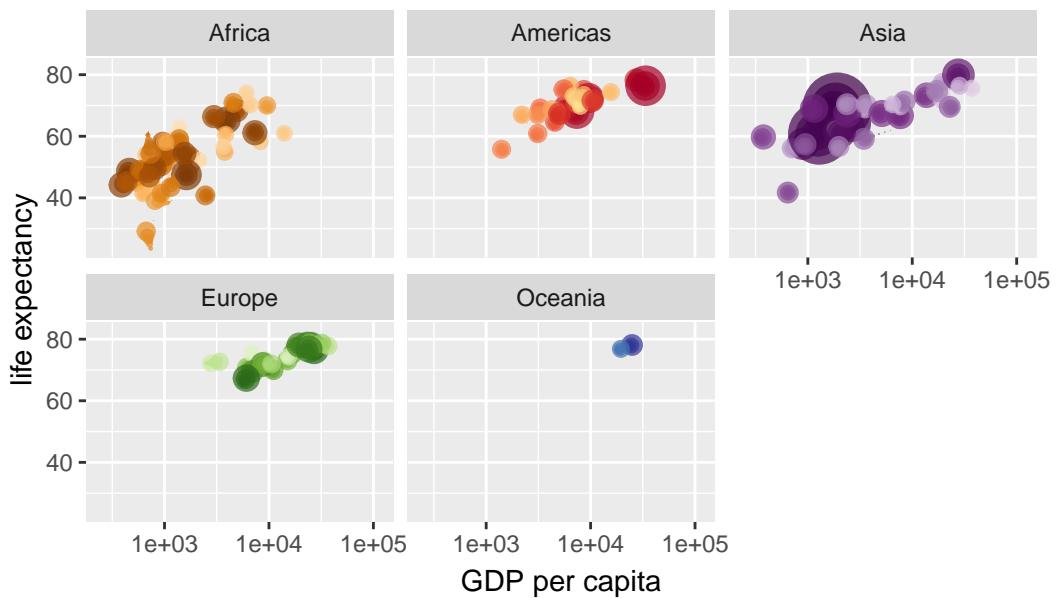
Year: 1993



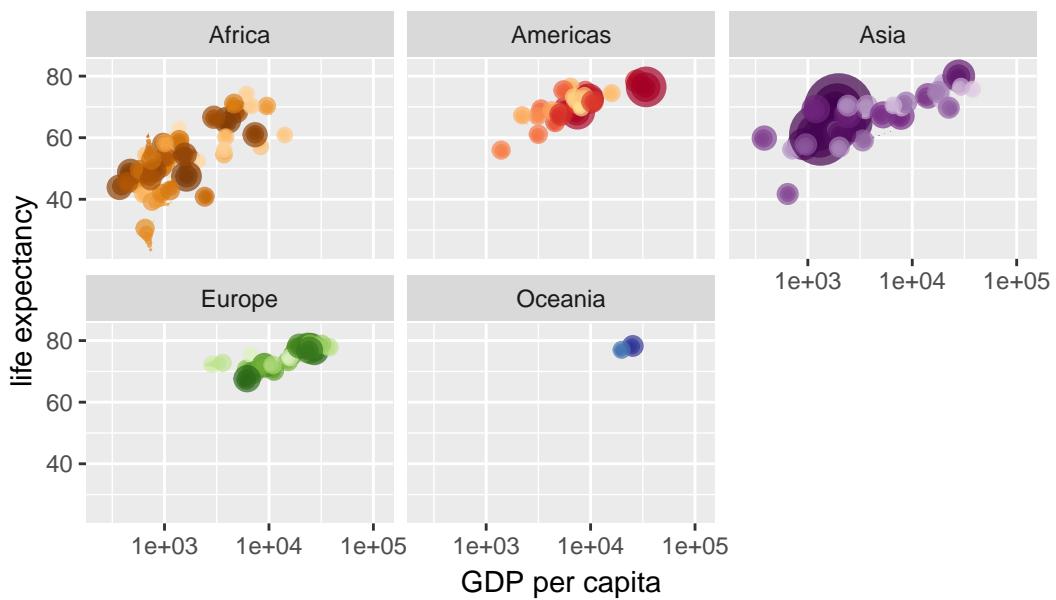
Year: 1994



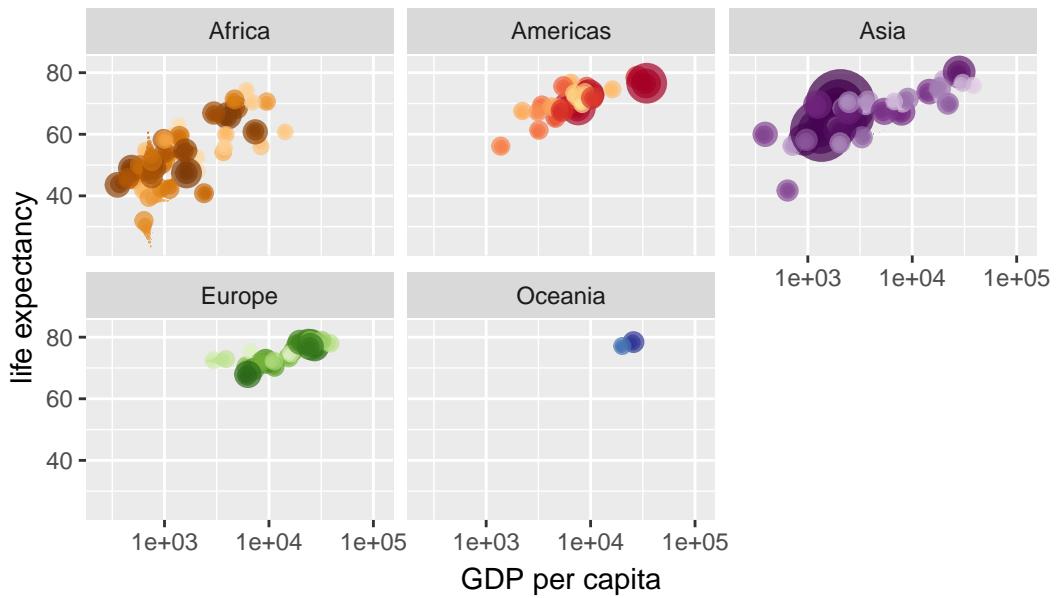
Year: 1994



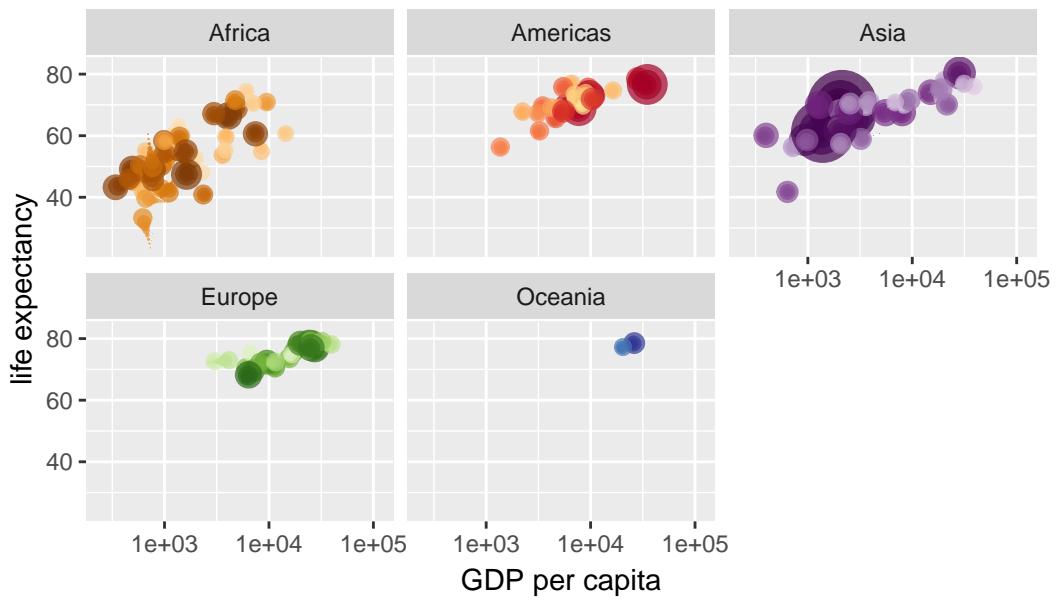
Year: 1995



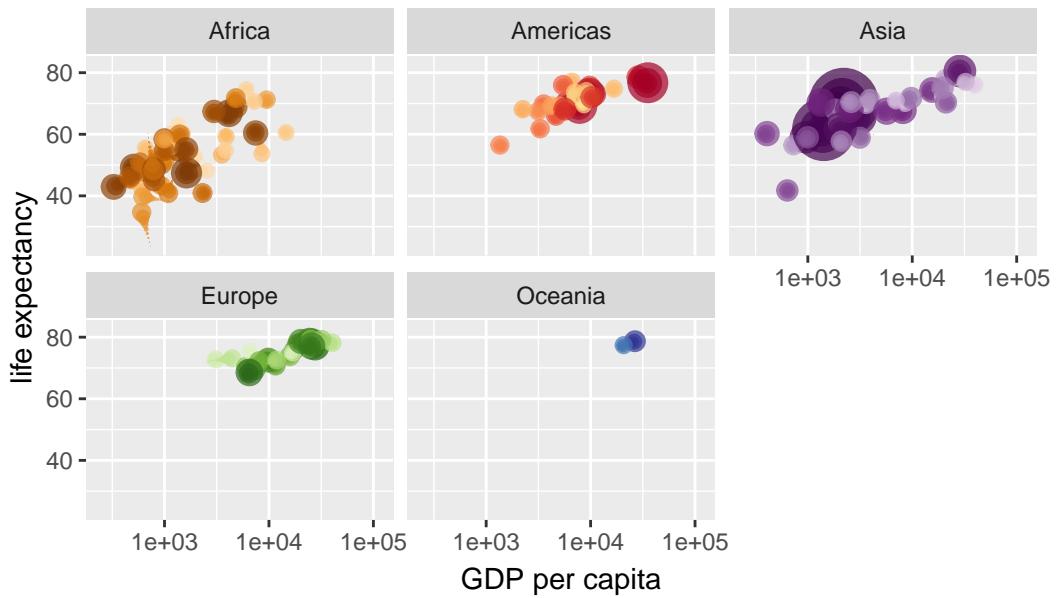
Year: 1995



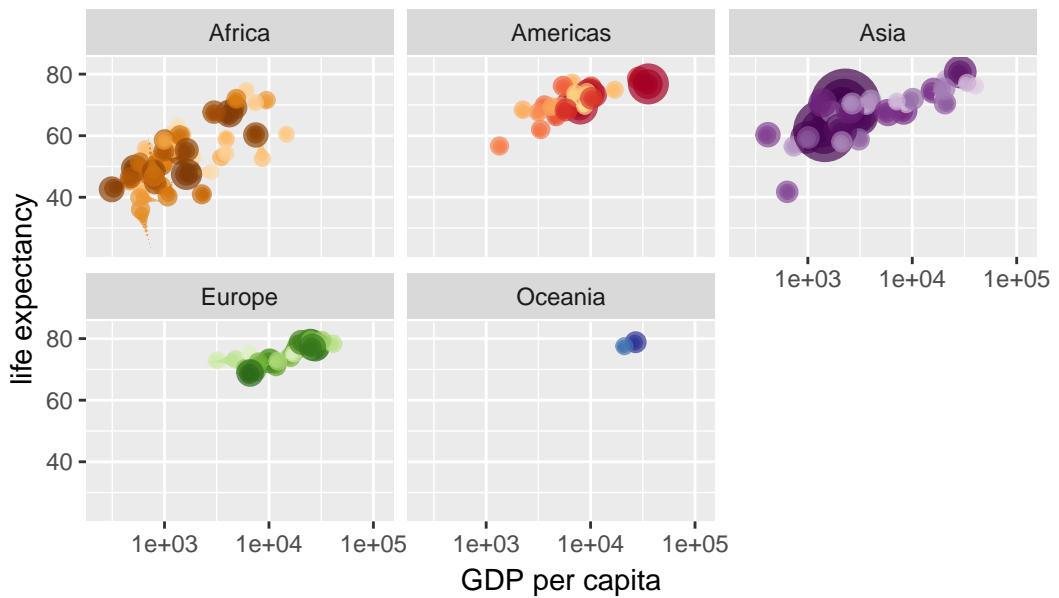
Year: 1996



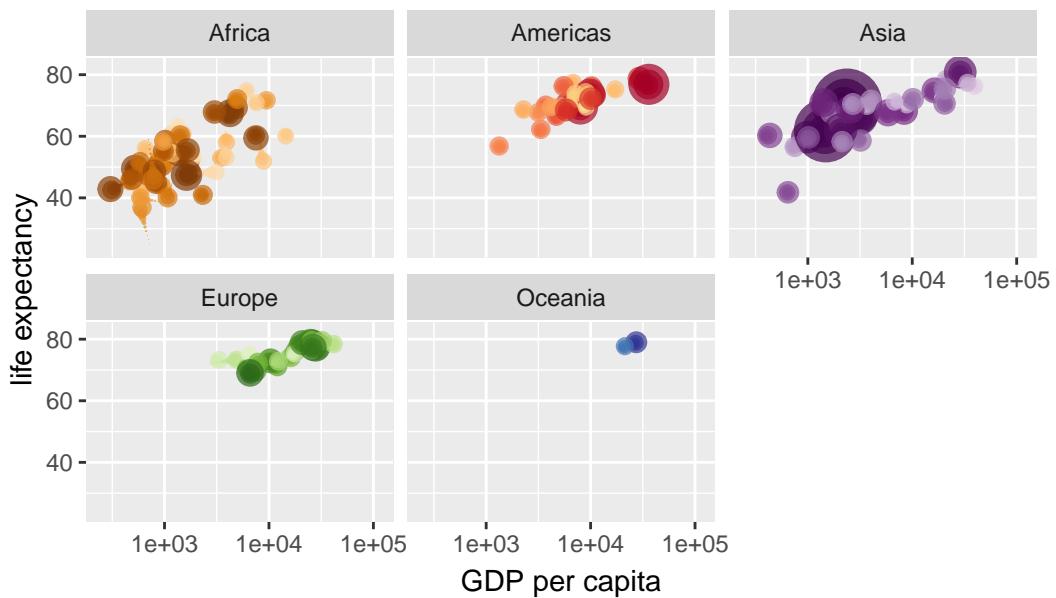
Year: 1996



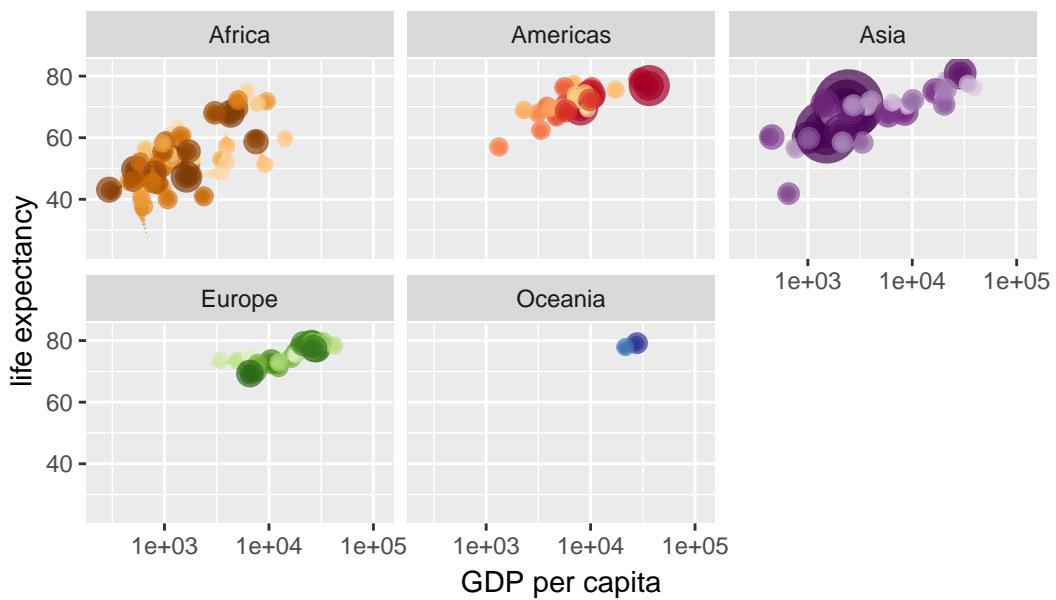
Year: 1997

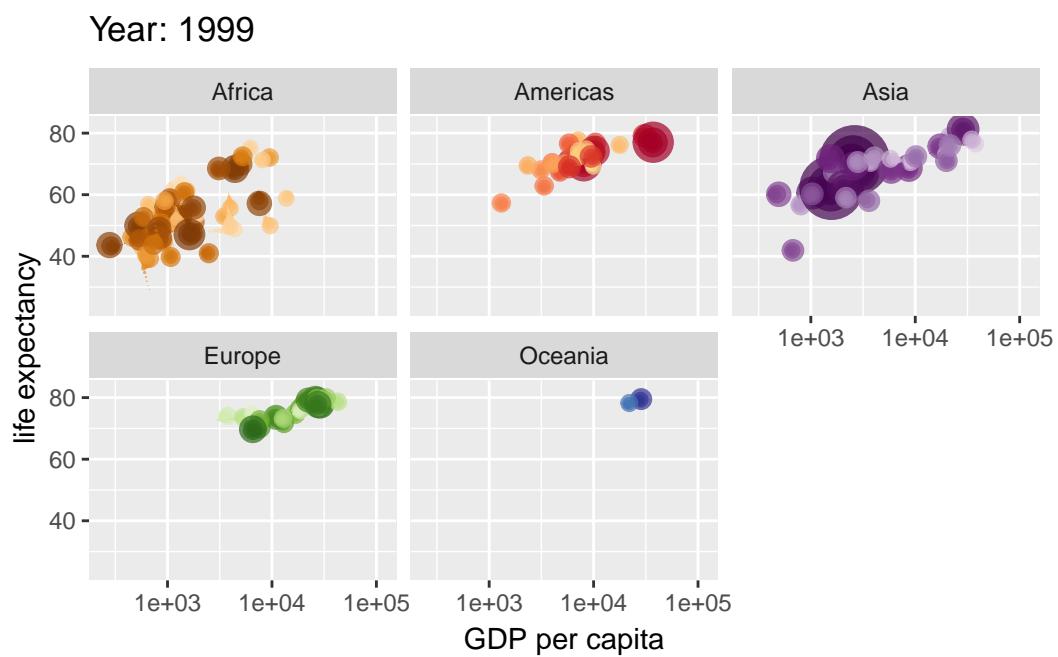
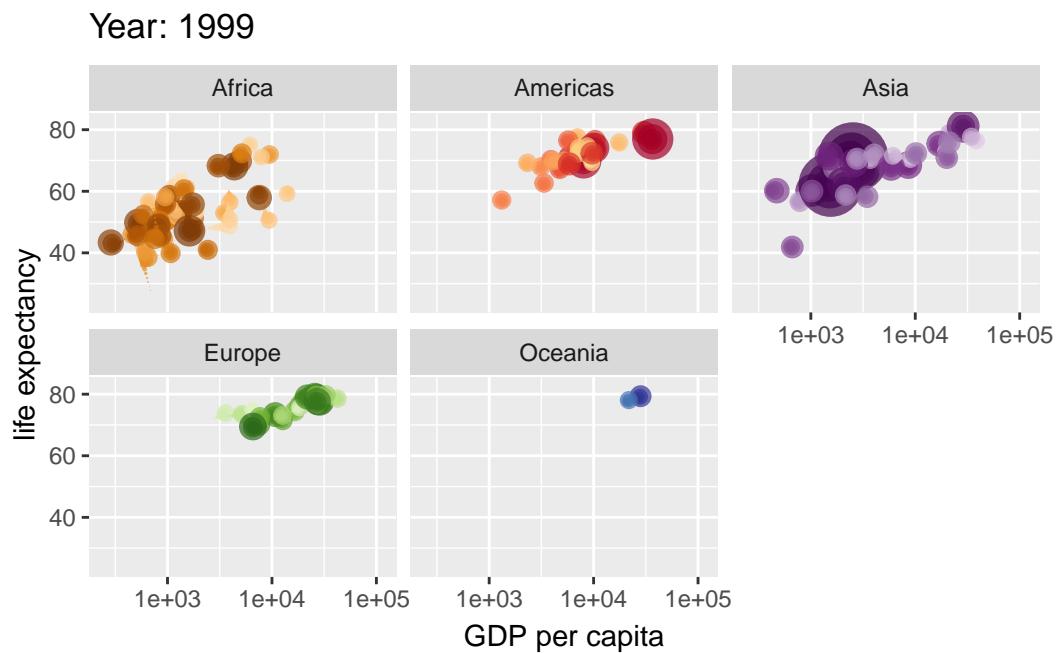


Year: 1998

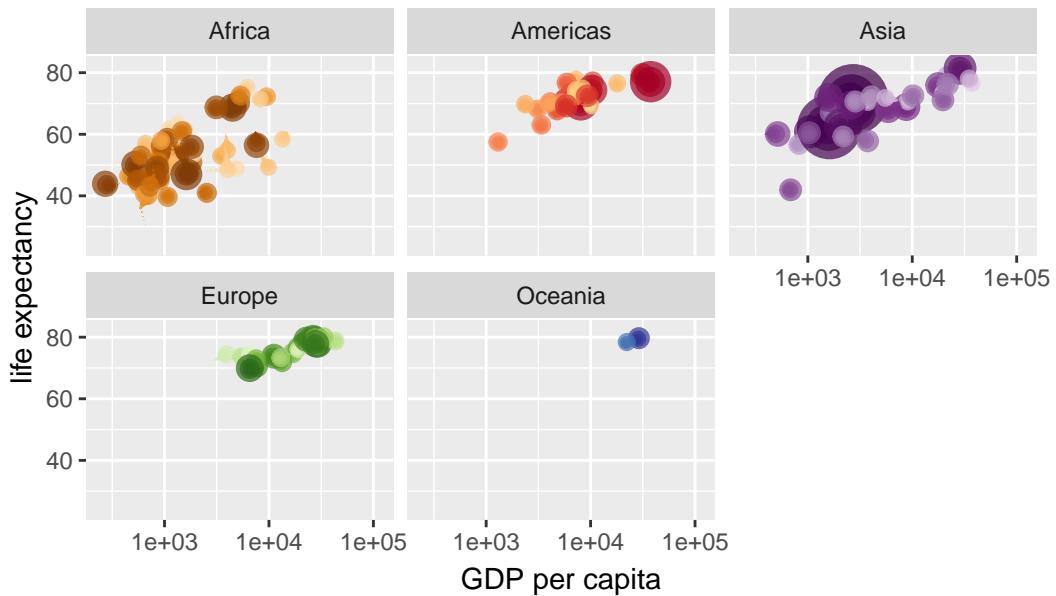


Year: 1998

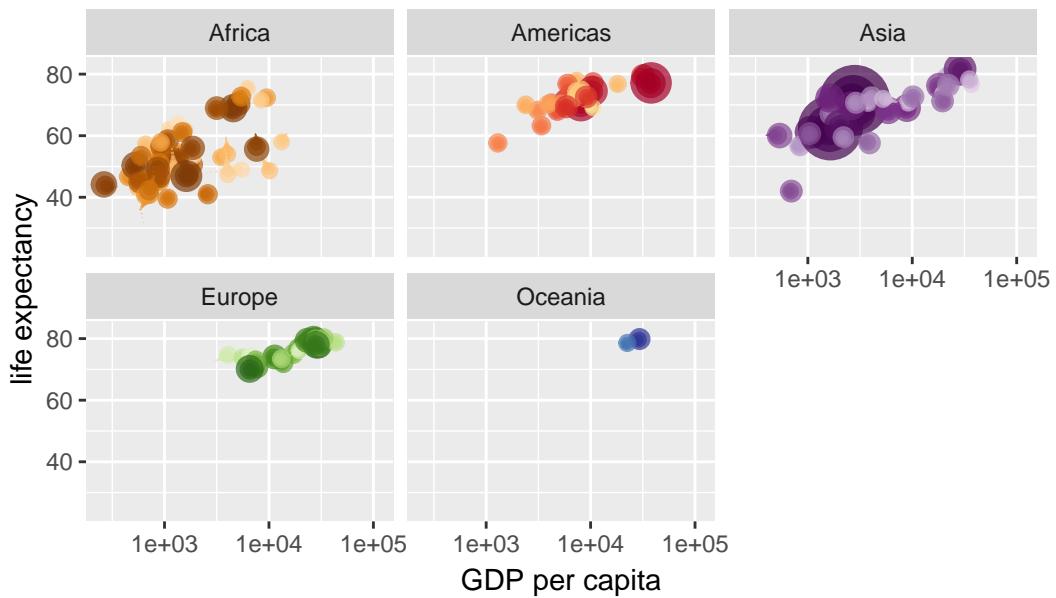




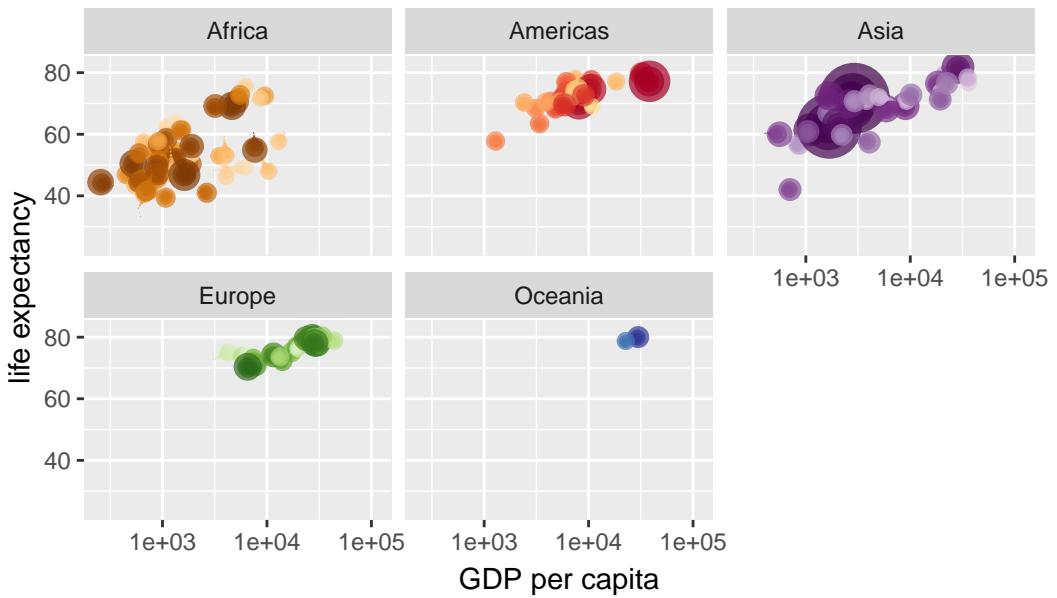
Year: 2000



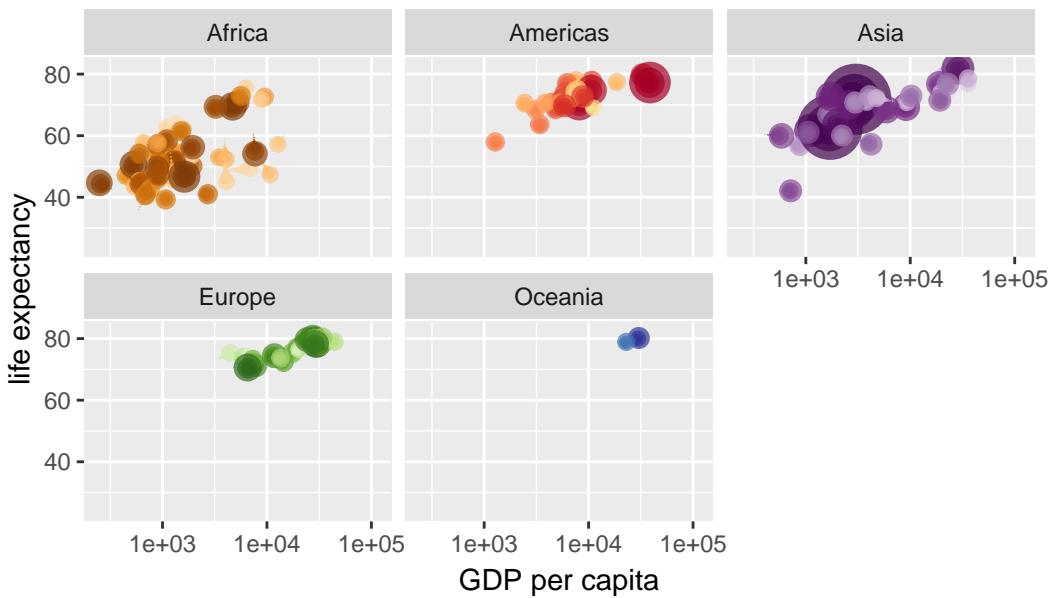
Year: 2000



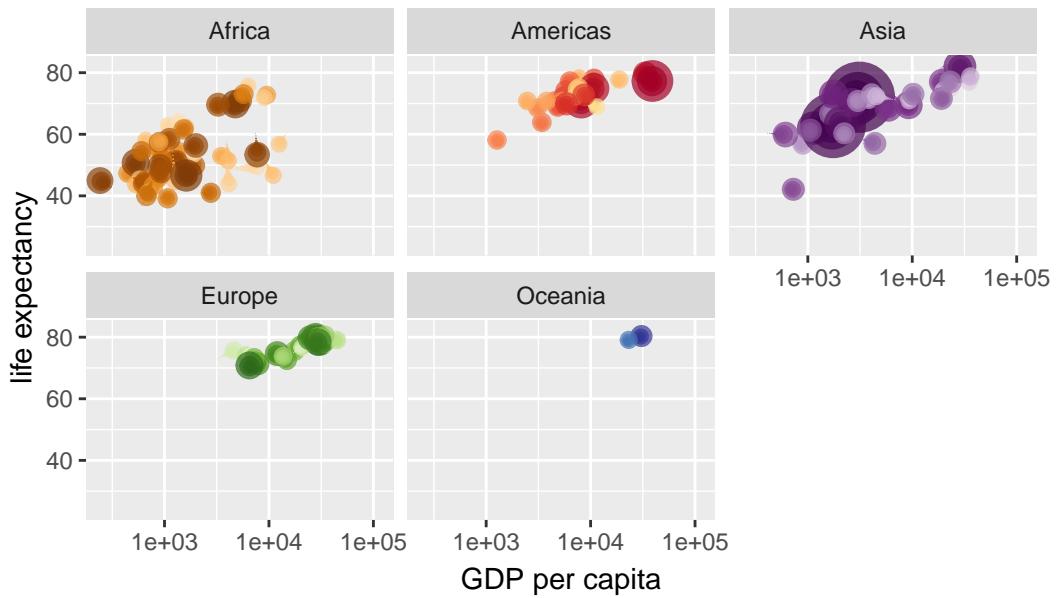
Year: 2001



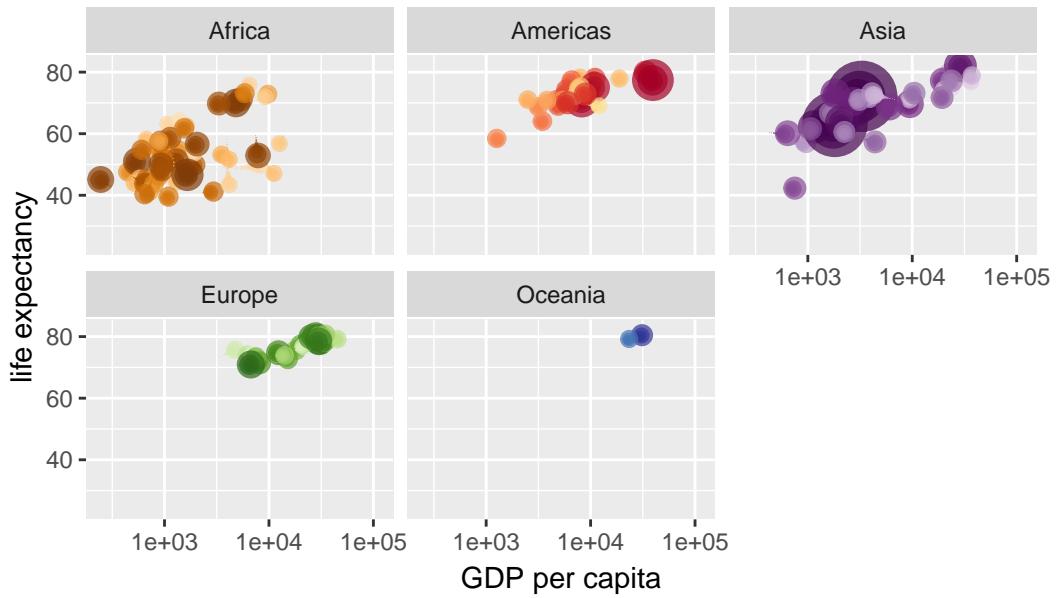
Year: 2001



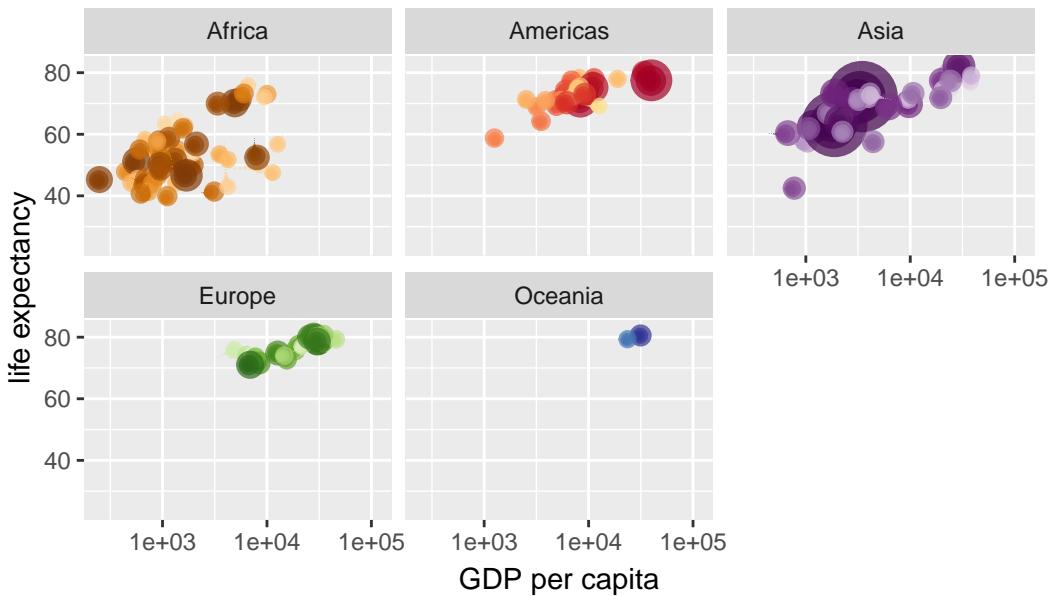
Year: 2002



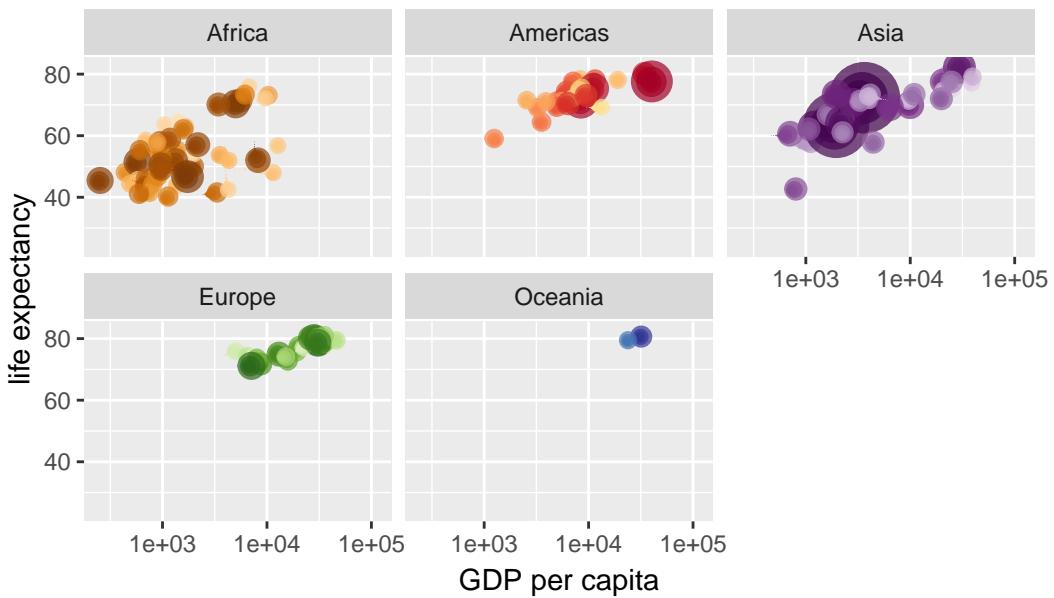
Year: 2003



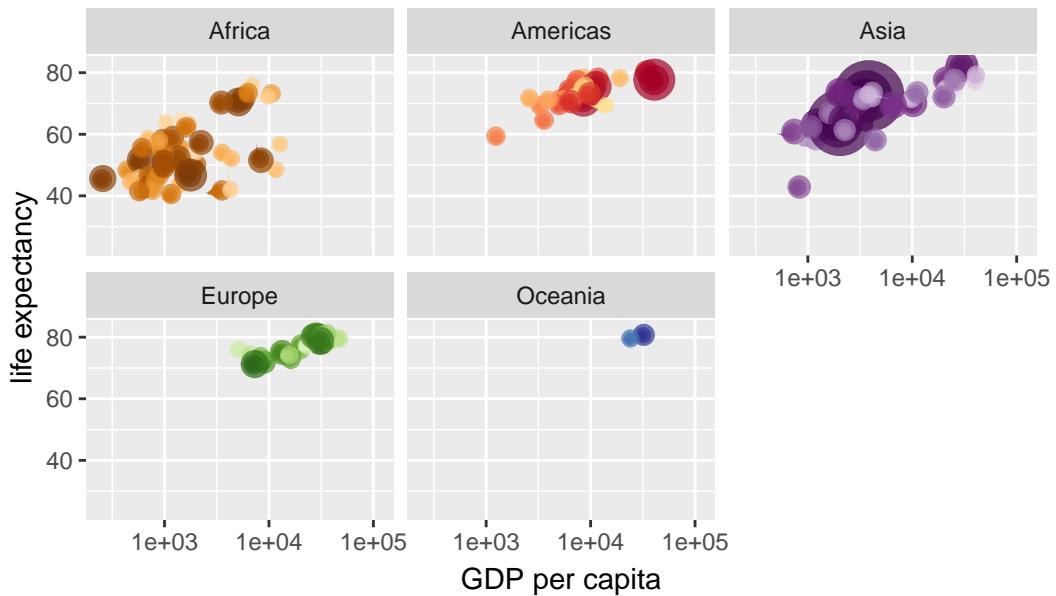
Year: 2003



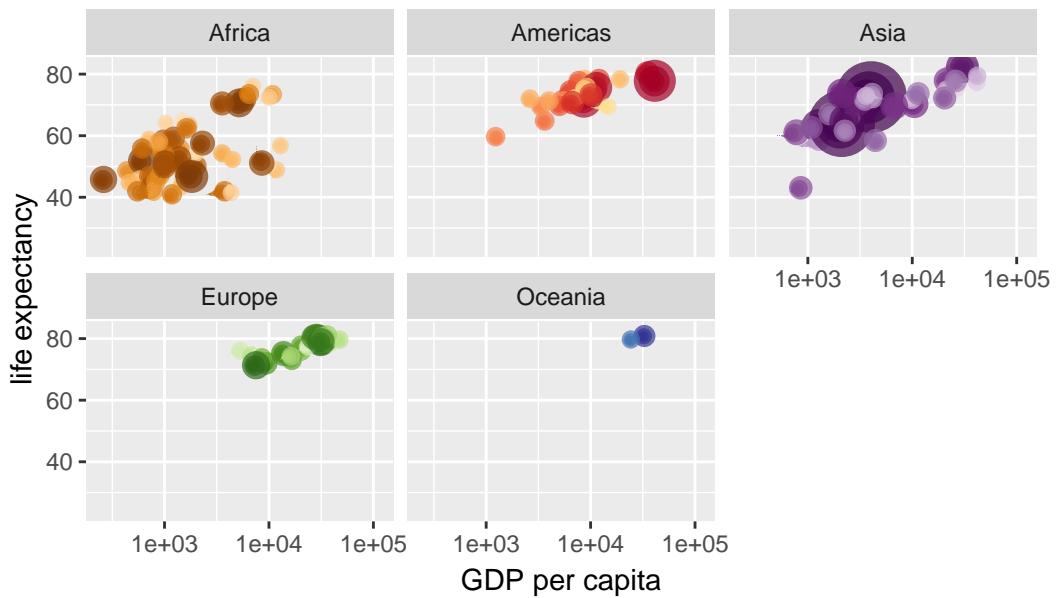
Year: 2004



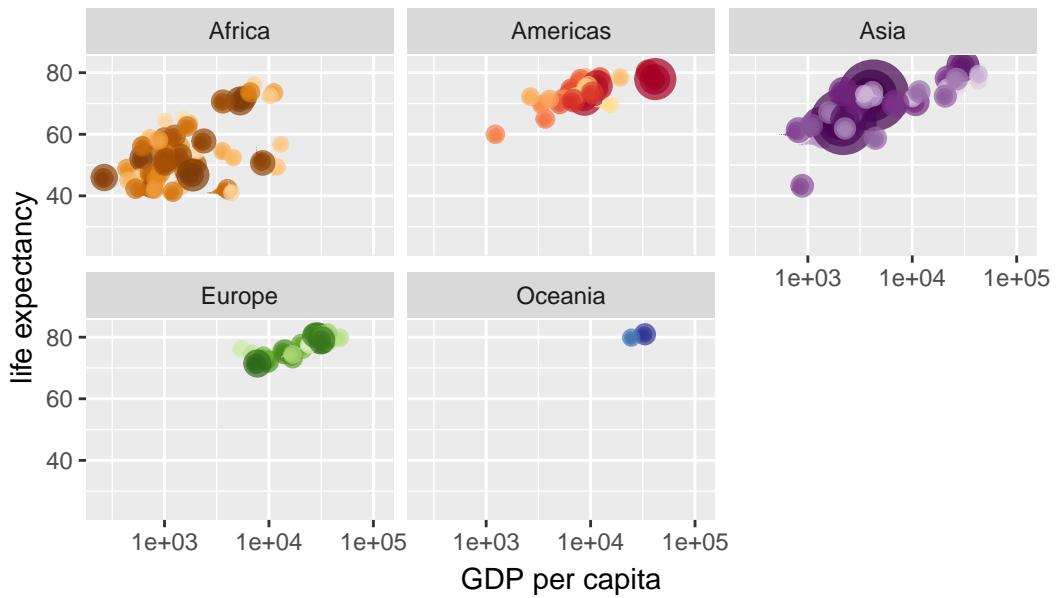
Year: 2004



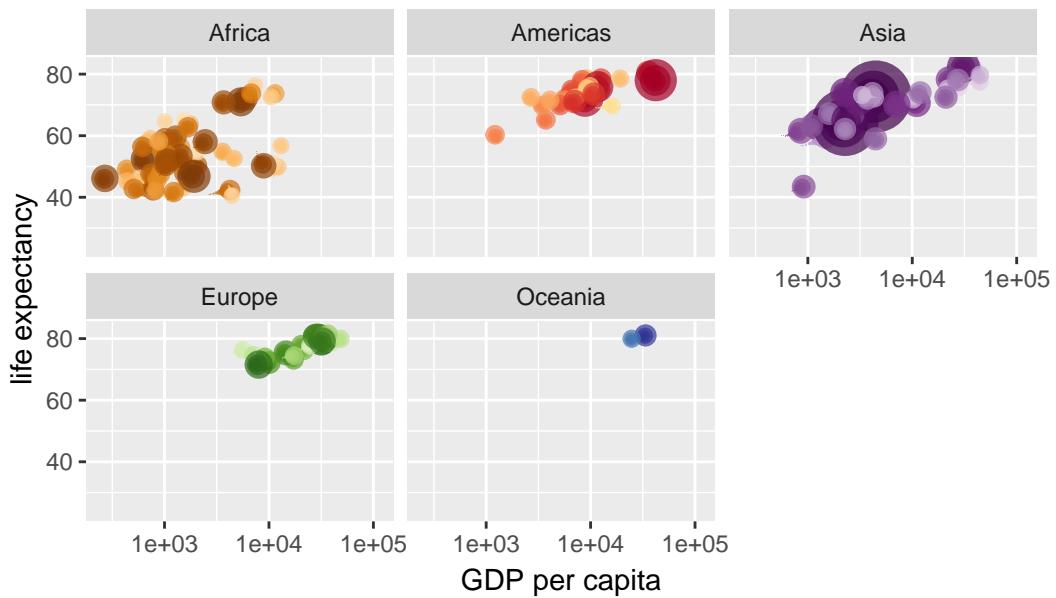
Year: 2005



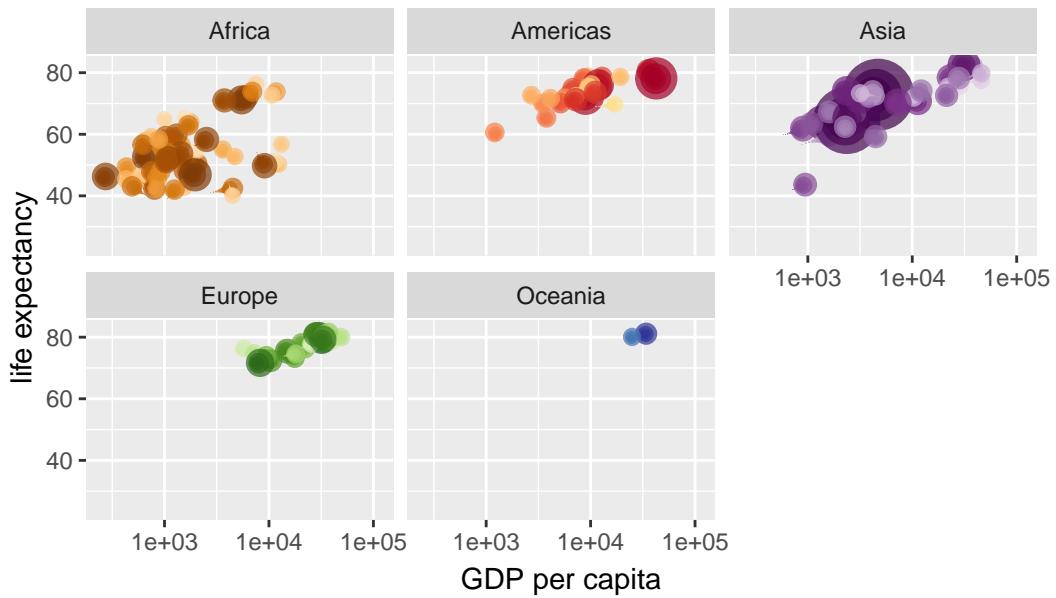
Year: 2005



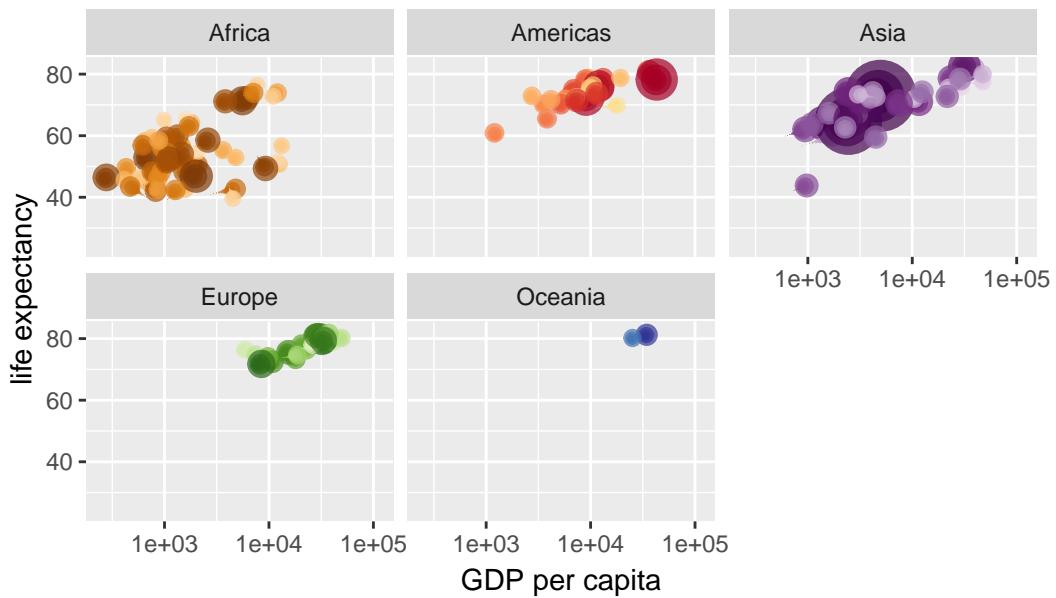
Year: 2006



Year: 2006



Year: 2007



Q. Combine plots

```

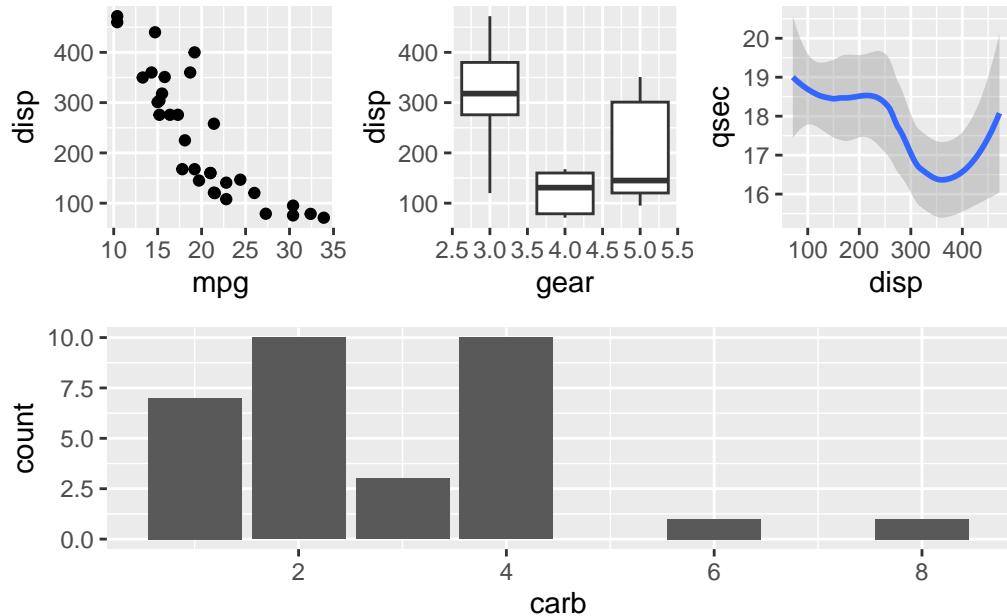
#Install install.packages("patchwork")
library(patchwork)

#Create some plots
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(disp, qsec))
p4 <- ggplot(mtcars) + geom_bar(aes(carb))

#Combine them
(p1 | p2 | p3) / p4

```

`geom_smooth()` using method = 'loess' and formula = 'y ~ x'



Q. Report setup of document

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
Running under: Windows 10 x64 (build 19045)
```

```
Matrix products: default
```

```
locale:
```

```
[1] LC_COLLATE=Korean_Korea.utf8  LC_CTYPE=Korean_Korea.utf8  
[3] LC_MONETARY=Korean_Korea.utf8 LC_NUMERIC=C  
[5] LC_TIME=Korean_Korea.utf8
```

```
time zone: America/Los_Angeles
```

```
tzcode source: internal
```

```
attached base packages:
```

```
[1] stats      graphics   grDevices utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] patchwork_1.2.0 gganimate_1.0.8 dplyr_1.1.4      gapminder_1.0.0  
[5] ggplot2_3.4.4
```

```
loaded via a namespace (and not attached):
```

```
[1] Matrix_1.6-1.1     gtable_0.3.4       jsonlite_1.8.8     crayon_1.5.2  
[5] compiler_4.3.2    tidyselect_1.2.0   progress_1.2.3     splines_4.3.2  
[9] scales_1.3.0      yaml_2.3.8        fastmap_1.1.1     lattice_0.21-9  
[13] R6_2.5.1         labeling_0.4.3    generics_0.1.3     knitr_1.45  
[17] tibble_3.2.1      munsell_0.5.0    pillar_1.9.0      rlang_1.1.3  
[21] utf8_1.2.4        stringi_1.8.3    xfun_0.41        cli_3.6.2  
[25] tweenr_2.0.2      withr_3.0.0      magrittr_2.0.3    mgcv_1.9-0  
[29] digest_0.6.34     grid_4.3.2       hms_1.1.3        lifecycle_1.0.4  
[33] nlme_3.1-163     prettyunits_1.2.0 vctrs_0.6.5     evaluate_0.23  
[37] glue_1.7.0        farver_2.1.1     gifski_1.12.0-2  fansi_1.0.6  
[41] colorspace_2.1-0 rmarkdown_2.25   tools_4.3.2      pkgconfig_2.0.3  
[45] htmltools_0.5.7
```