

Class 5: Data Viz with ggplot

Leah Kim A16973745

Table of contents

Background	1
Gene Expression Plot	6
Custom Color Plot	7
Using Different geoms	8
File location Online	13
Facetting	14
OPTIONAL CONTENT	15
Going Further	15
Bar Charts	19
Flipping Bar Charts	22
Extentions: Animation	24

Background

There are many graphics systems available in R. These include “base” R and tones of add on packages like **ggplot2**.

Let’s compare “base” and **ggplot2** briefly. We can use some example data that is built-in with R called **cars**:

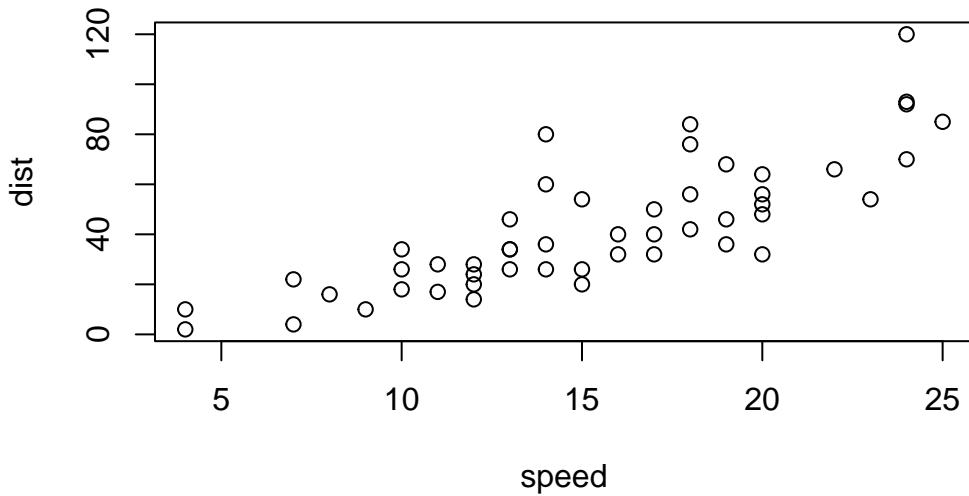
```
head(cars)
```

```
speed dist
1      4     2
2      4    10
```

```
3      7      4
4      7     22
5      8     16
6      9     10
```

In base R, I can just call `plot()`.

```
plot(cars)
```



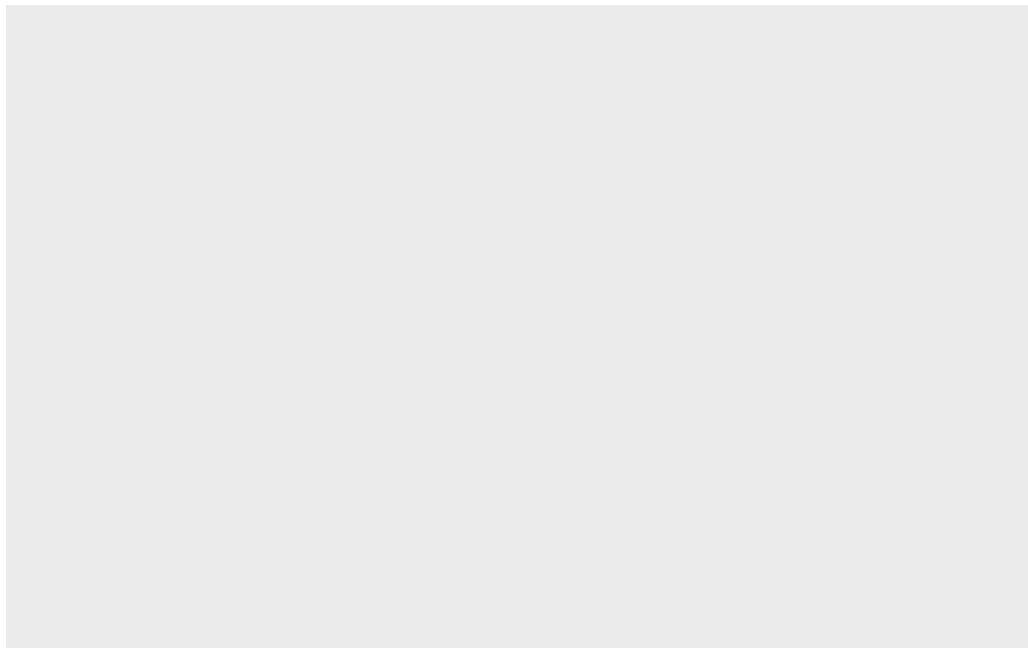
How can we do this with **ggplot2**.

First we need to install the package. We do this `install.packages("ggplot2")`. I only need to do this once and then it will be available on my computer from then on.

Key point: I only install packages in the R console, not within quarto docs or R scripts.

Before I use any add-on package, I must load it up with a call to `library()`.

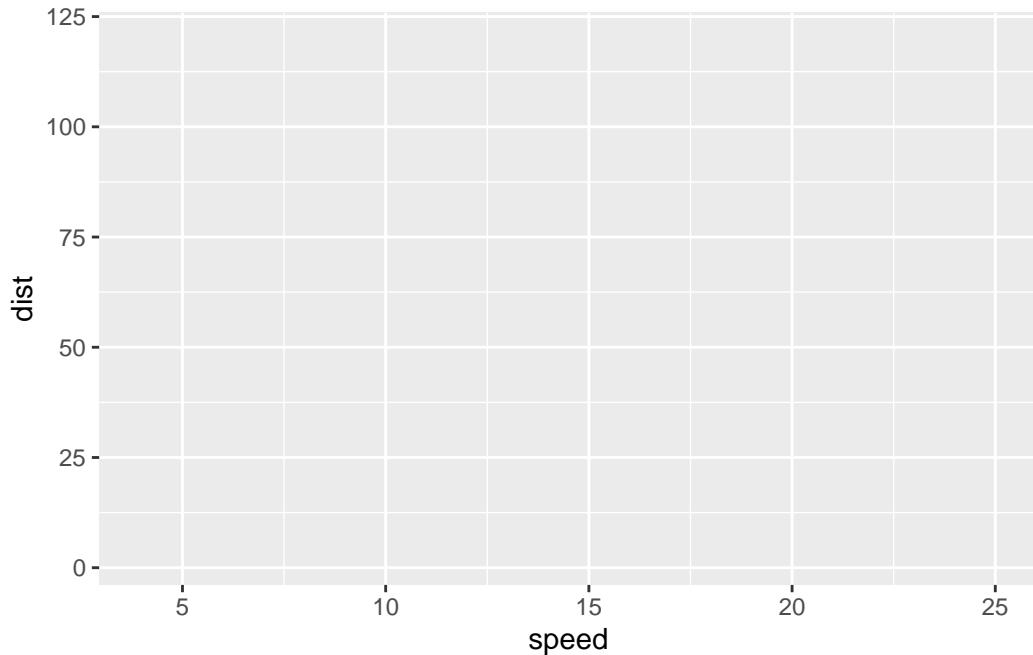
```
#install.packages("ggplot2") ##uncomment to install if needed
library(ggplot2)
ggplot(cars)
```



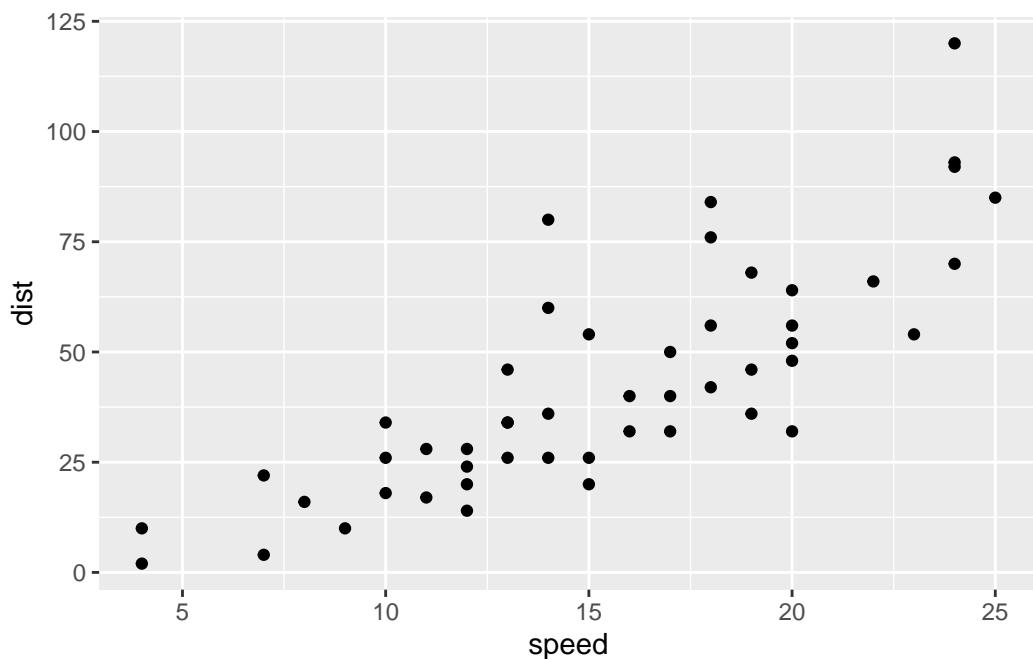
Every ggplot has at least 3 things:

- The **data** (in our case, `cars`)
- the **aesthetics** (how the data maps to the plot)
- the **geoms** that determine how the plot is drawn (lines, points, columns, etc)

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



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



For “simple” plots, ggplot is much more verbose than base R but the defaults are nicer and for complicated plots, it becomes much more efficient and structured.

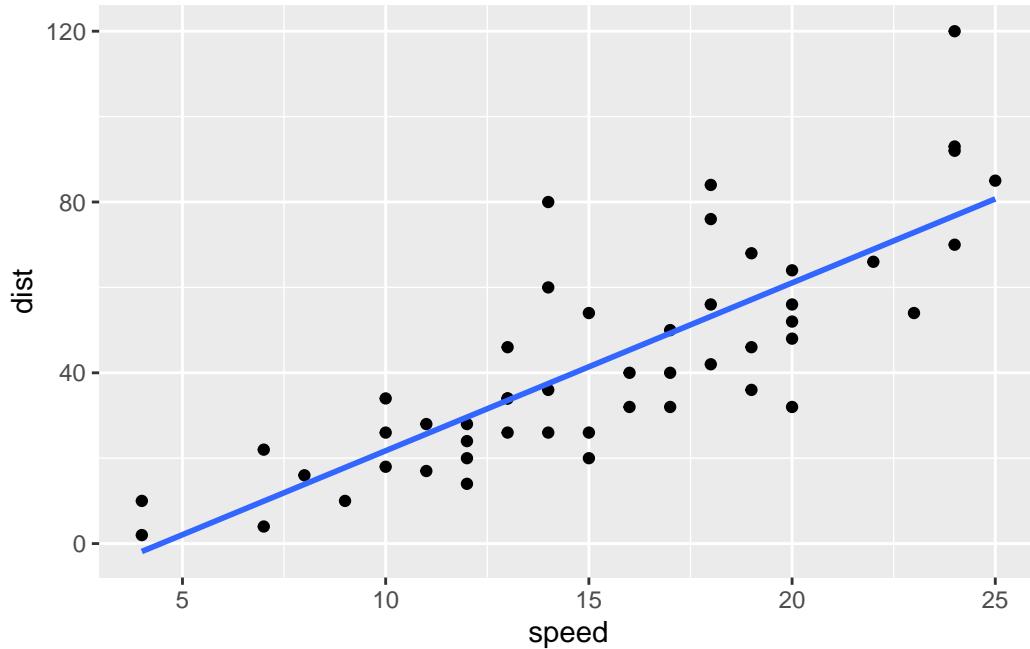
Q. Add a line to show the relationship of speed to stopping distance (i.e. add another “layer”)

```
p <- ggplot(cars) +  
  aes(x=speed, y = dist) +  
  geom_point() +  
  geom_smooth(method="lm", se=FALSE)
```

I can always save any ggplot object (i.e. plot) and then use it later for adding more layers.

```
p
```

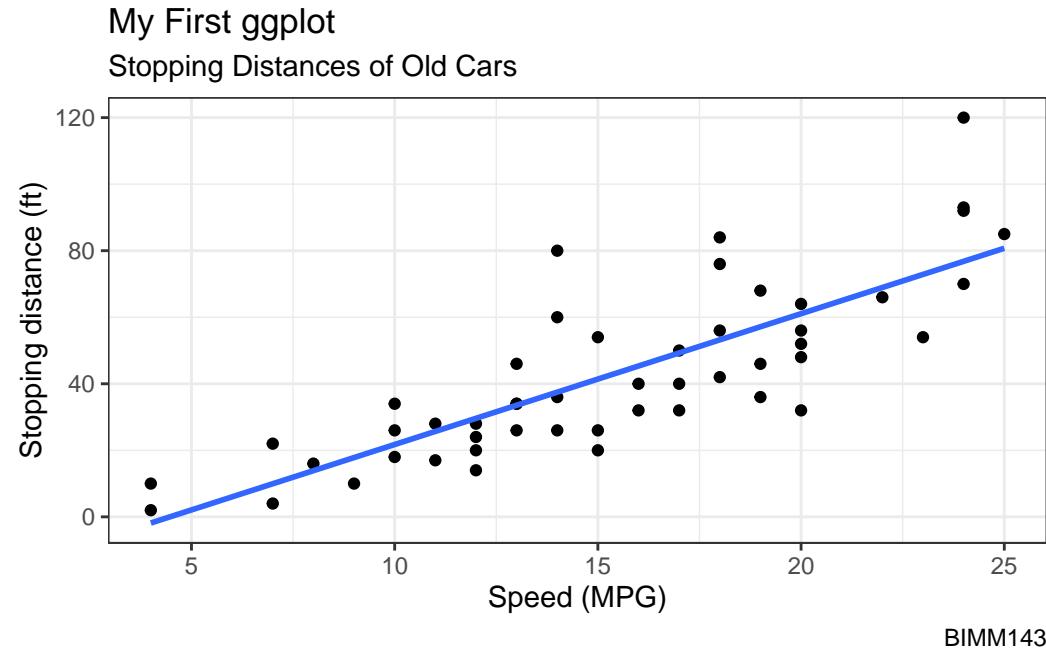
```
`geom_smooth()` using formula = 'y ~ x'
```



Q. Add a title and subtitle to the plot

```
p + labs(title="My First ggplot",  
        subtitle = "Stopping Distances of Old Cars", caption = "BIMM143", x="Speed (MPG)", y = "Distancia de frenado (pies)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Gene Expression Plot

Read input data into R

```
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. How many genes are in this dataset?

```
nrow(genes)
```

[1] 5196

Q. How many columns are there?

```
ncol(genes)
```

[1] 4

Q. How many ‘up’ and ‘down’ regulated genes are there?

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

	down	unchanging	up
	72	4997	127

Q. Using your values above, what fraction of total genes is up-regulated in this dataset?

```
round(table(genes$State)/nrow(genes) * 100, 2)
```

	down	unchanging	up
	1.39	96.17	2.44

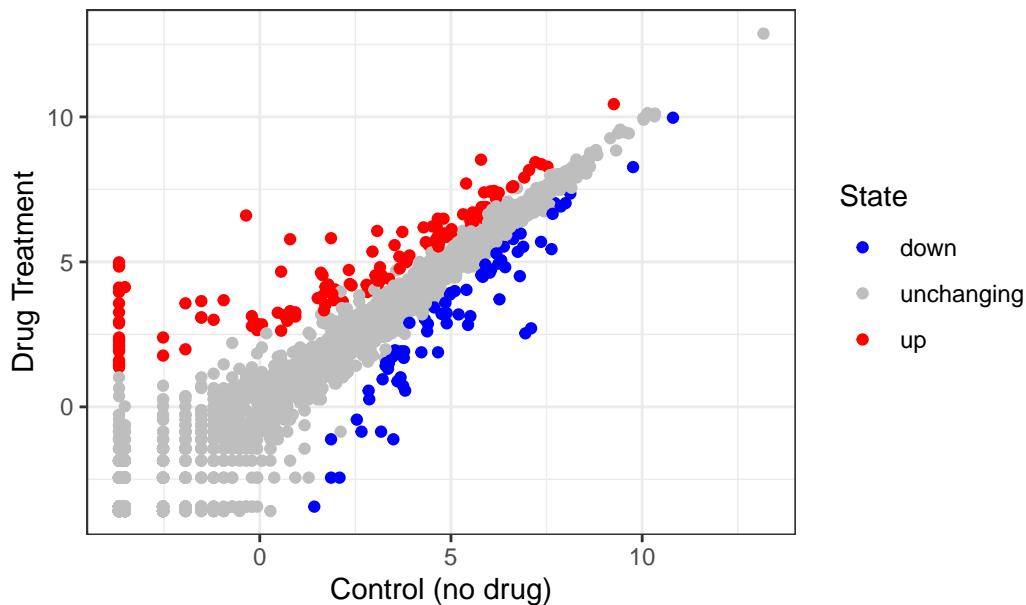
Custom Color Plot

Q. Make a first plot of this data

```
genePlot <- ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) + geom_point() +  
  scale_colour_manual( values=c("blue", "grey", "red")) +  
  labs(title="Gene Expression Changes Upon Drug Treatment", x="Control (no drug)", y ="Drug")  
  theme_bw()
```

```
genePlot
```

Gene Expression Changes Upon Drug Treatment



Using Different geoms

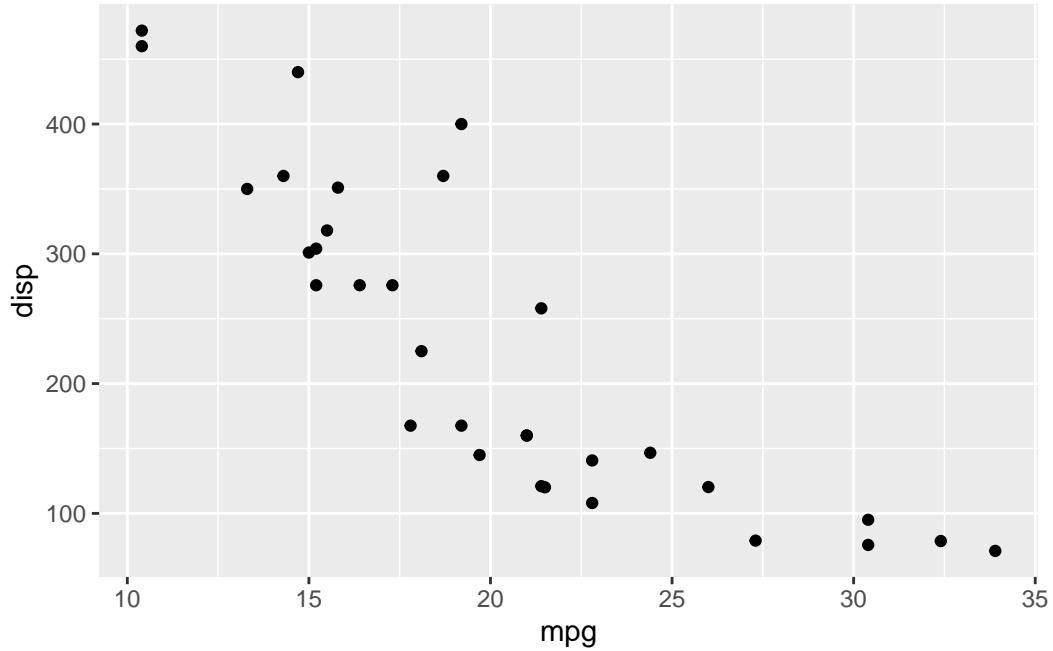
Let's plot some aspects of the inbuilt mtcars dataset.

```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

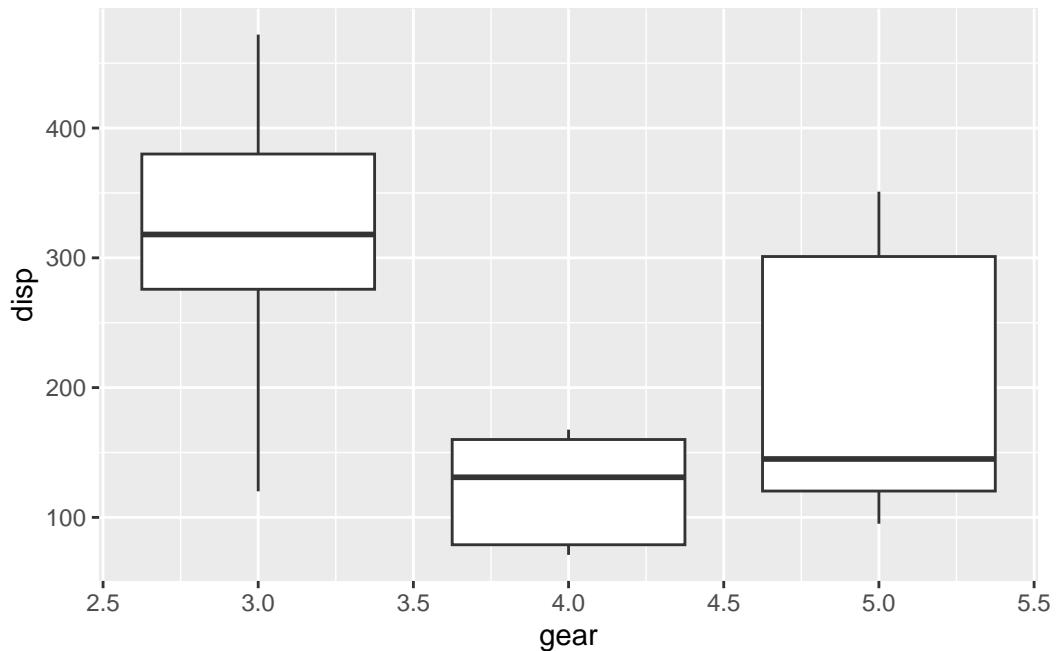
Q. Scatter of mpg vs disp

```
p1 <- ggplot(mtcars) +  
  aes(mpg, disp) +  
  geom_point()  
p1
```



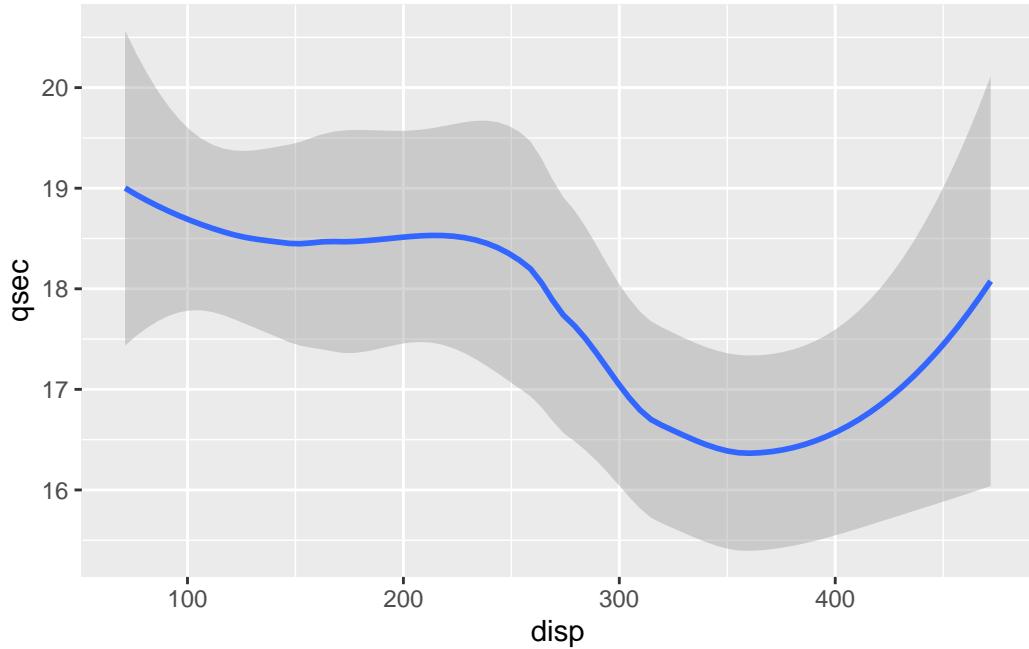
Q. Boxplot of gear vs disp

```
p2 <- ggplot(mtcars) +  
  aes(gear, disp, group = gear) +  
  geom_boxplot()  
p2
```



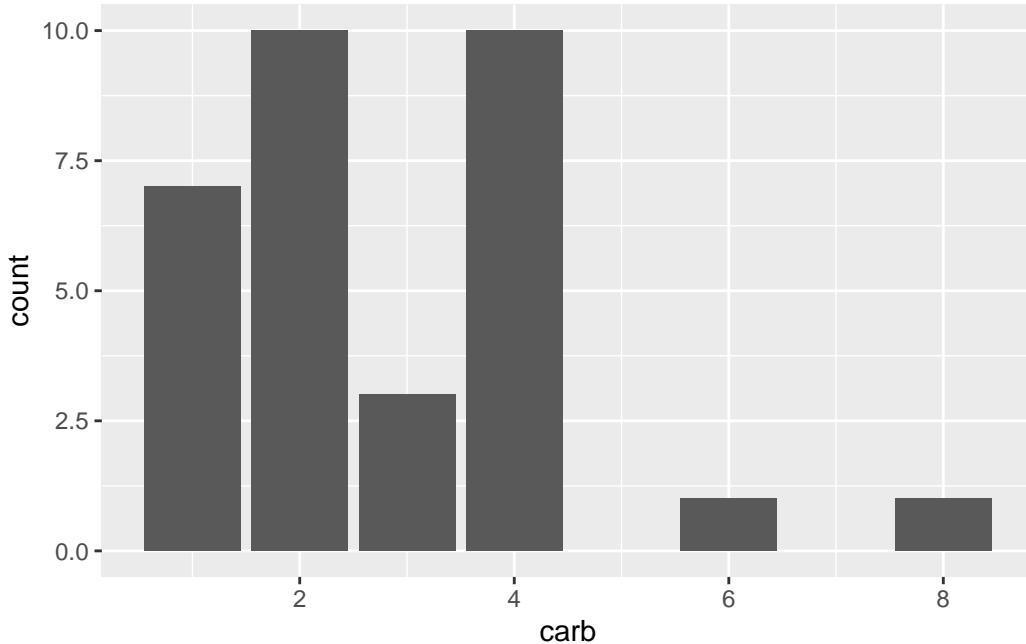
Q. Barplot of carb

```
p3 <- ggplot(mtcars) +  
  aes(disp, qsec) +  
  geom_smooth()  
p3  
  
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Q. Smooth of disp vs qsec

```
p4 <- ggplot(mtcars) +  
  aes(carb) +  
  geom_bar()  
p4
```

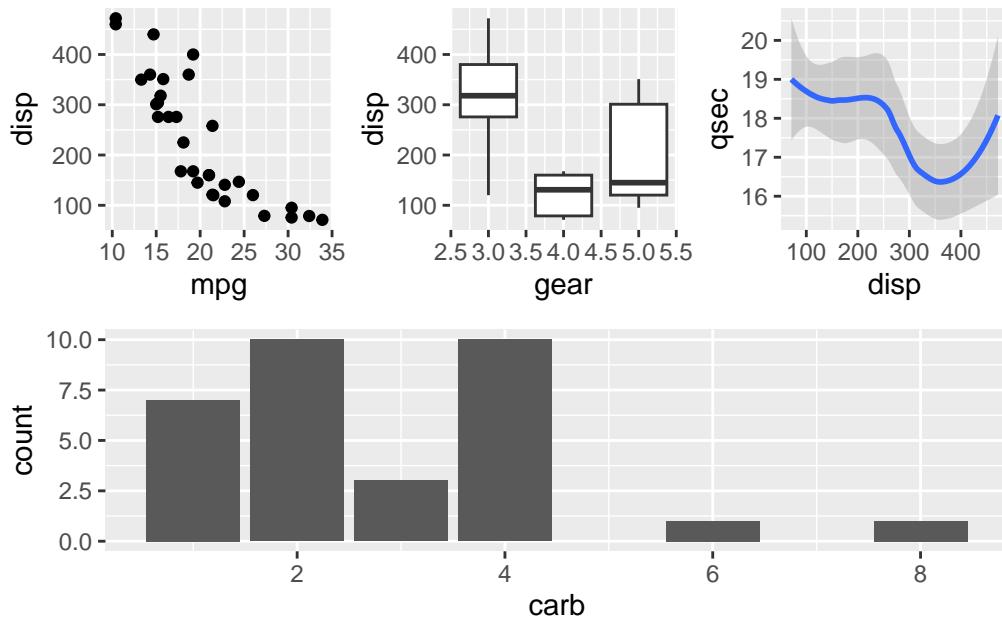


I want to combine all these plots into one figure with multiple panels.

We can use the **patchwork** package to do this.

```
#install.packages('patchwork')
library(patchwork)
((p1 | p2 | p3) / p4)

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



To save a plot as a file:

```
#paste under plot that was made, width and height are adjustable
#can also export plot if in viewer
ggsave(filename="myplot.png", width=10, height=10)
```

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

File location Online

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

```
# A tibble: 6 x 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.

```

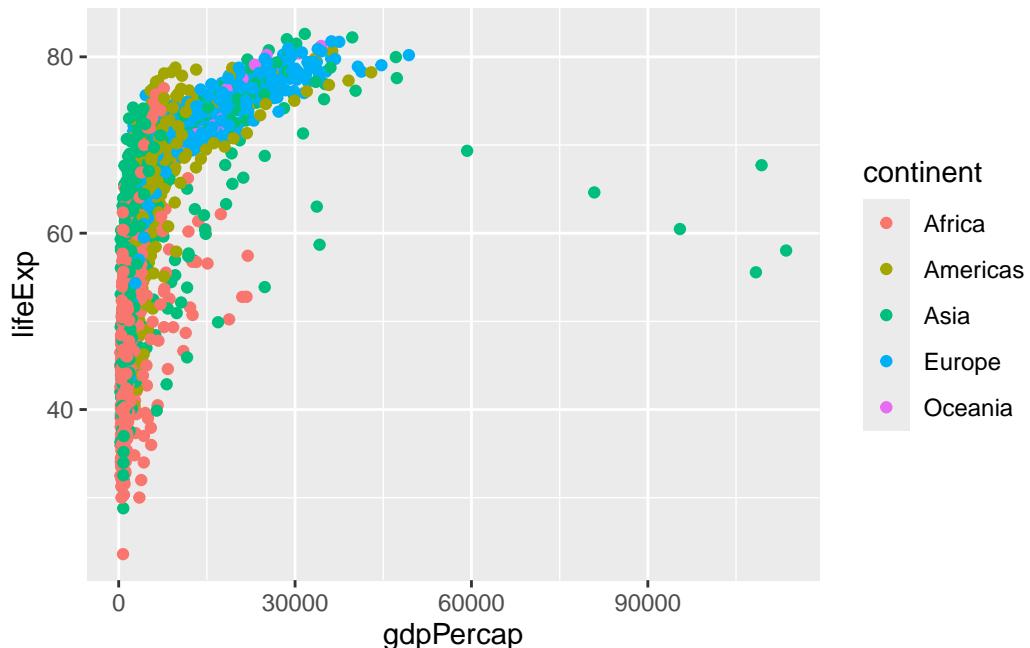
Q. How many countries are in this dataset

```
length(table(gapminder$country))
```

```
[1] 142
```

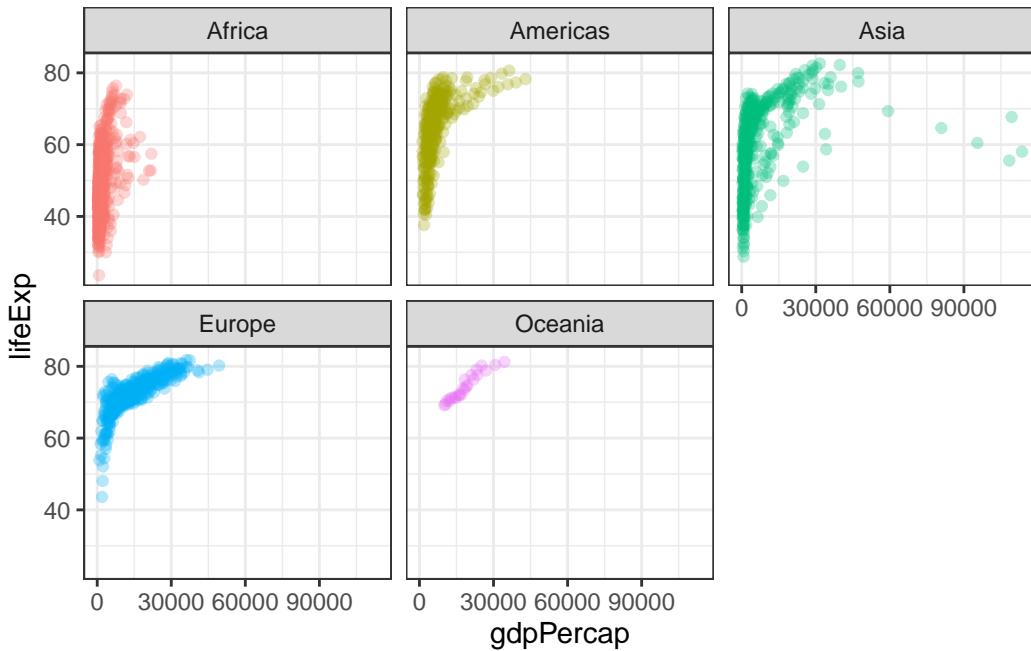
Q. Plot GDP vs Life Expectency, colored by continent

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



Facetting

```
ggplot(gapminder) +
  aes(x = gdpPercap, y = lifeExp, col = continent) +
  geom_point(alpha = 0.3, show.legend = FALSE) +
  facet_wrap(~continent) +
  theme_bw()
```



OPTIONAL CONTENT

Going Further

Q. Complete the code below to produce a first basic scatter plot of this gapminder_2007 dataset:

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

```
Attaching package: 'dplyr'
```

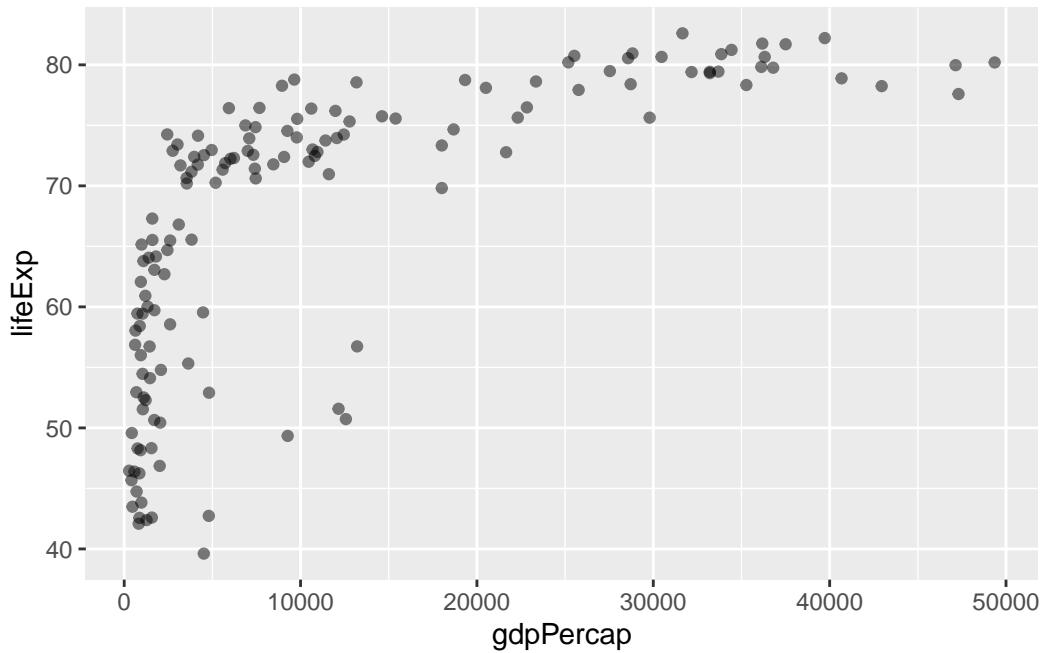
```
The following objects are masked from 'package:stats':
```

```
filter, lag
```

```
The following objects are masked from 'package:base':
```

```
intersect, setdiff, setequal, union
```

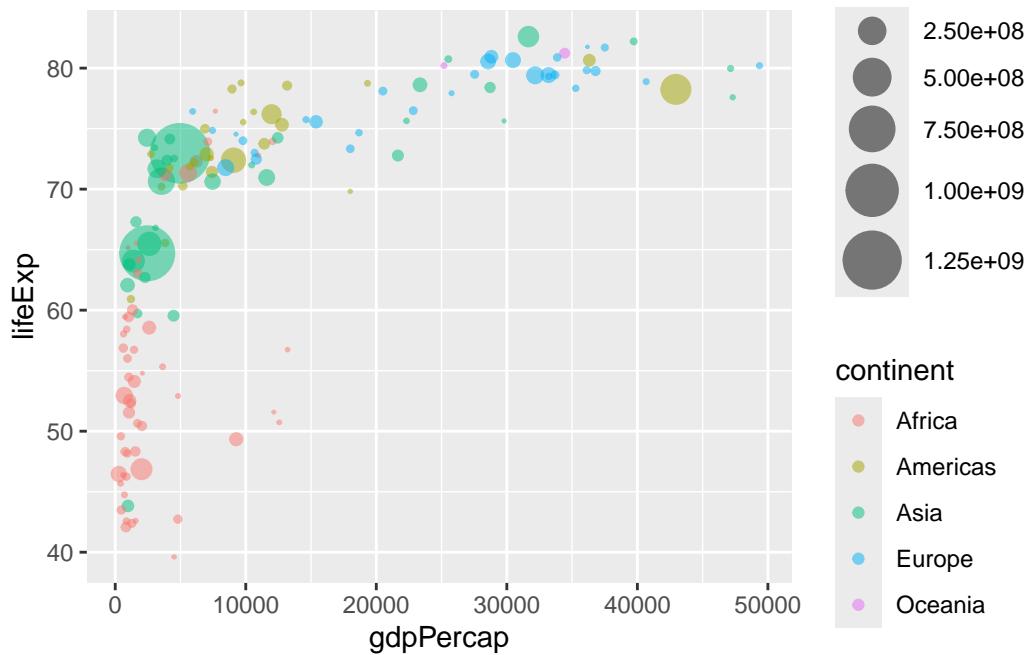
```
gapminder_2007 <- gapminder %>% filter(year ==2007)
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```



Customized plot

```
gapminder_2007 <- gapminder %>% filter(year ==2007)
gap_2007 <- ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5) +
  scale_size_area(max_size = 10)

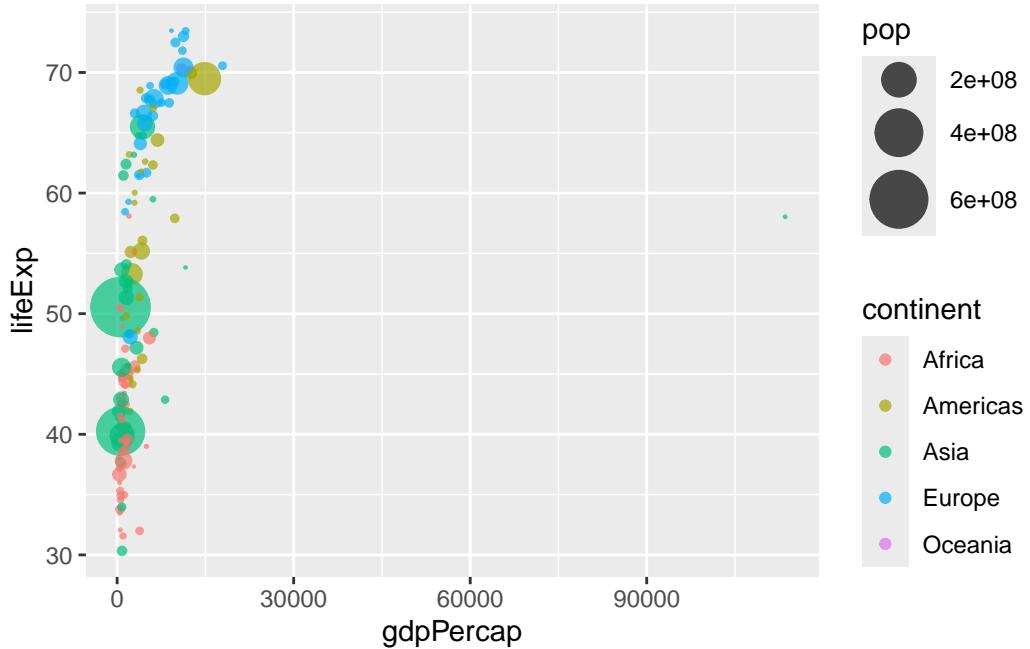
gap_2007
```



Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957?

```
gapminder_1957 <- gapminder %>% filter(year == 1957)
gap_1957 <- ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color = continent, size = pop) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 10)

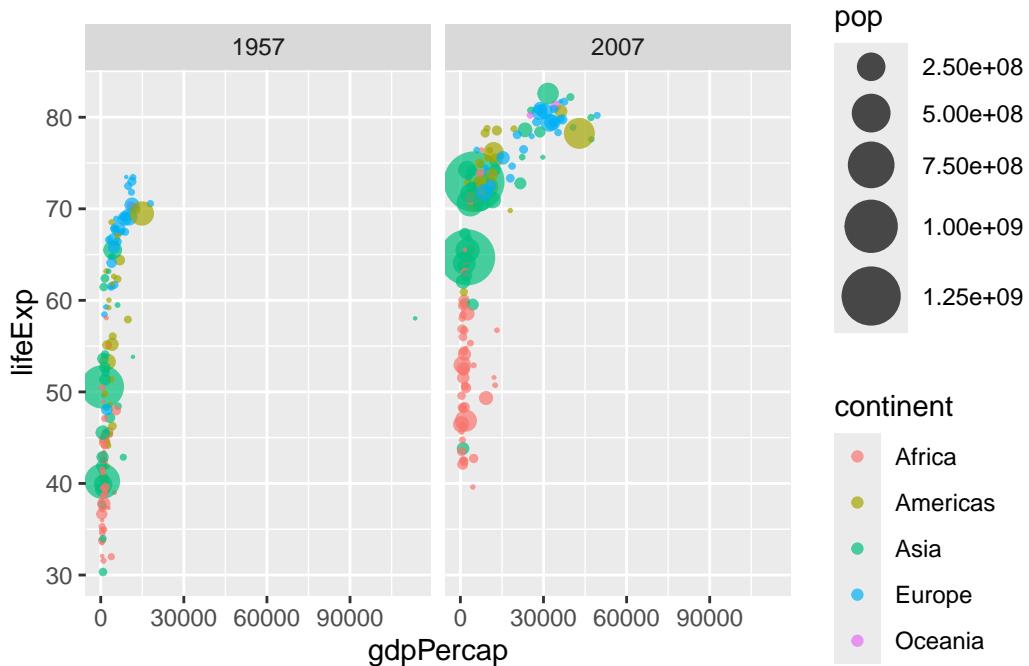
gap_1957
```



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_compare <- gapminder %>% filter(year == 1957 | year == 2007)
gap_compare <- ggplot(gapminder_compare) +
  aes(x = gdpPercap, y = lifeExp, color = continent, size = pop) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)

gap_compare
```



Bar Charts

Creating a simple bar chart

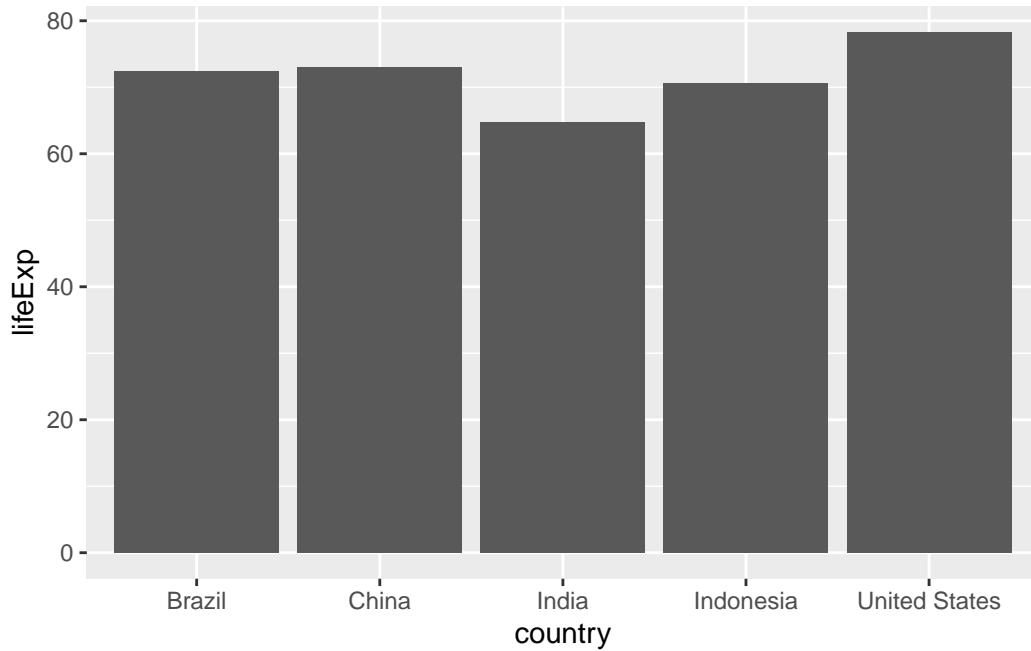
```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)
```

```
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.
```

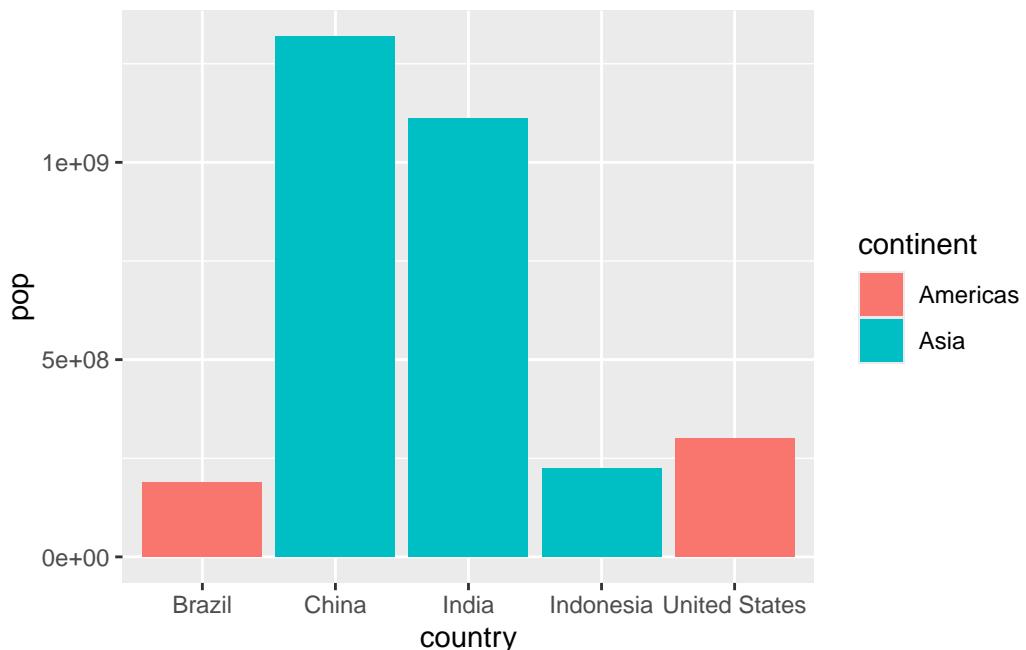
Create a bar chart showing the life expectancy of the five biggest countries by population in 2007

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

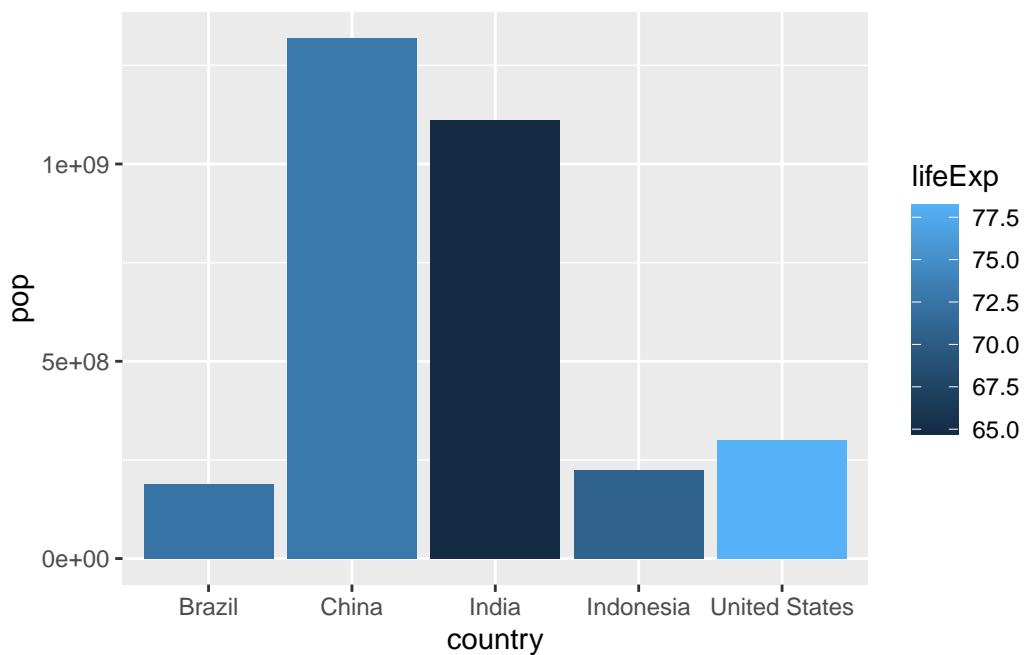


Coloring barcharts

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

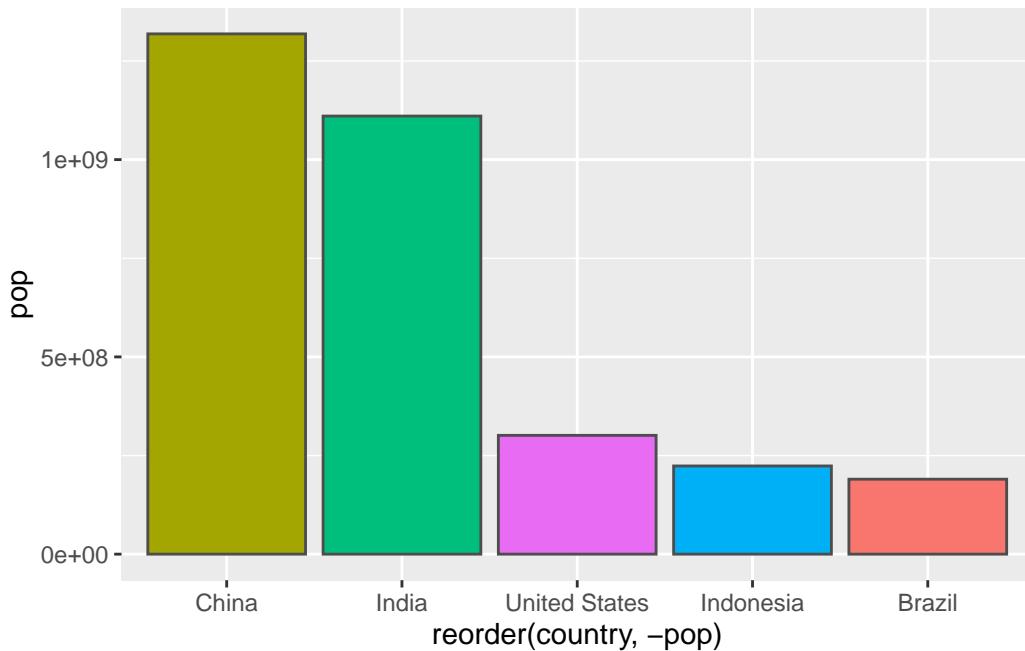


```
ggplot(gapminder_top5) +  
  geom_col(aes(x = country, y = pop, fill = lifeExp))
```



Plot Population size by country. Create a bar chart showing the population (in millions) of the five biggest countries by population in 2007.

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



Flipping Bar Charts

First plot

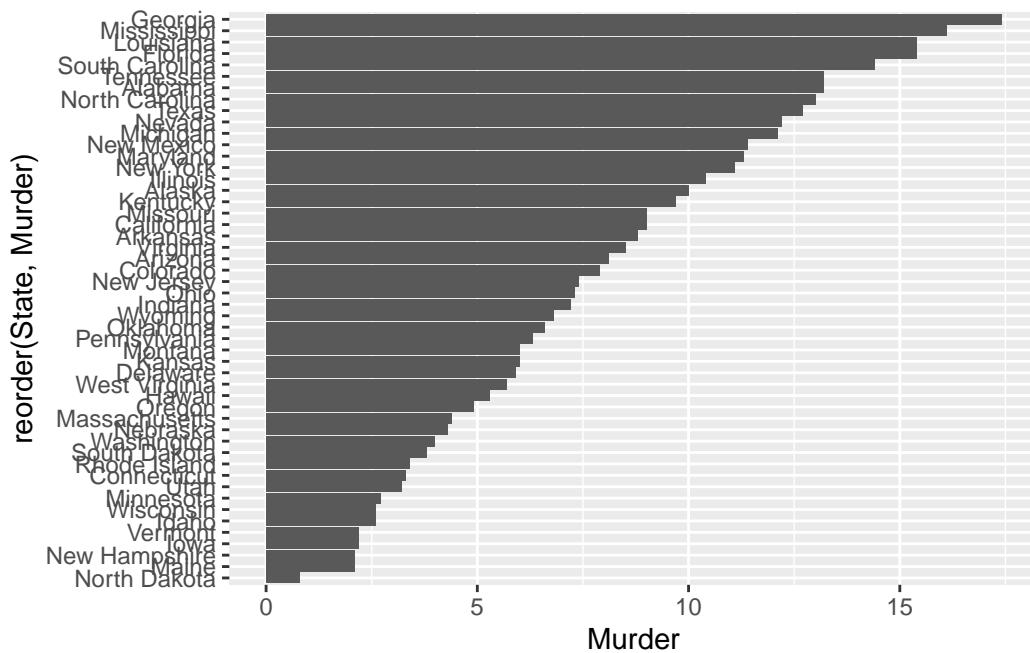
```
head(USArrests)
```

	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

```

USArrests$State <- rownames(USArrests)
ggplot(USArrests) + aes(x=reorder(State, Murder), y = Murder) +
  geom_col() +
  coord_flip()

```

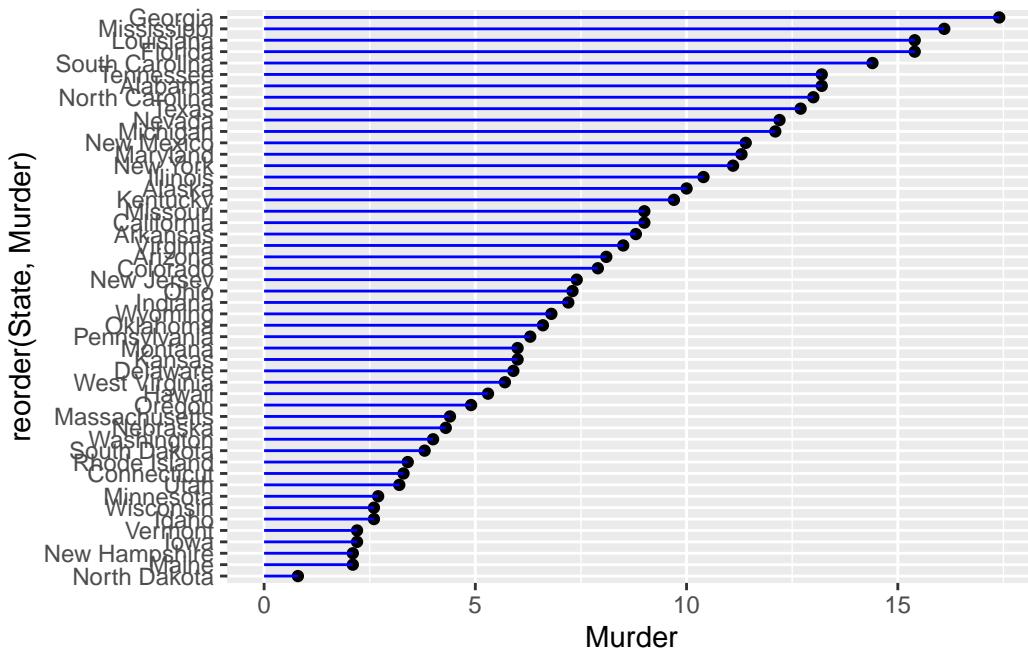


Custom Plot

```

ggplot(USArrests) +
  aes(x = reorder(State, Murder), y = Murder) +
  geom_point() +
  geom_segment(aes(x = State,
                    xend = State,
                    y = 0,
                    yend = Murder), color = "blue") +
  coord_flip()

```



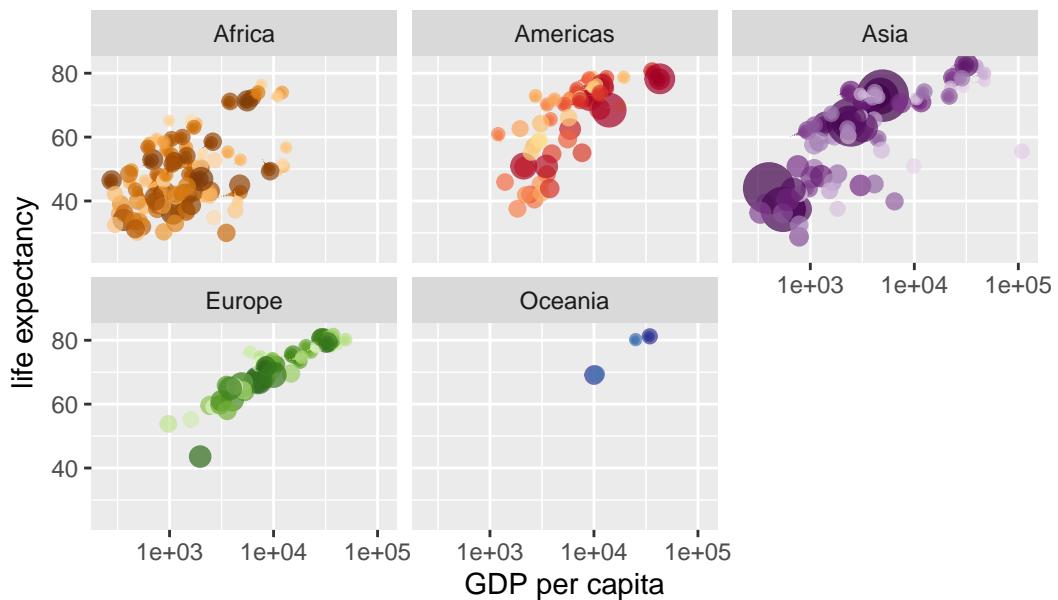
Extentions: Animation

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

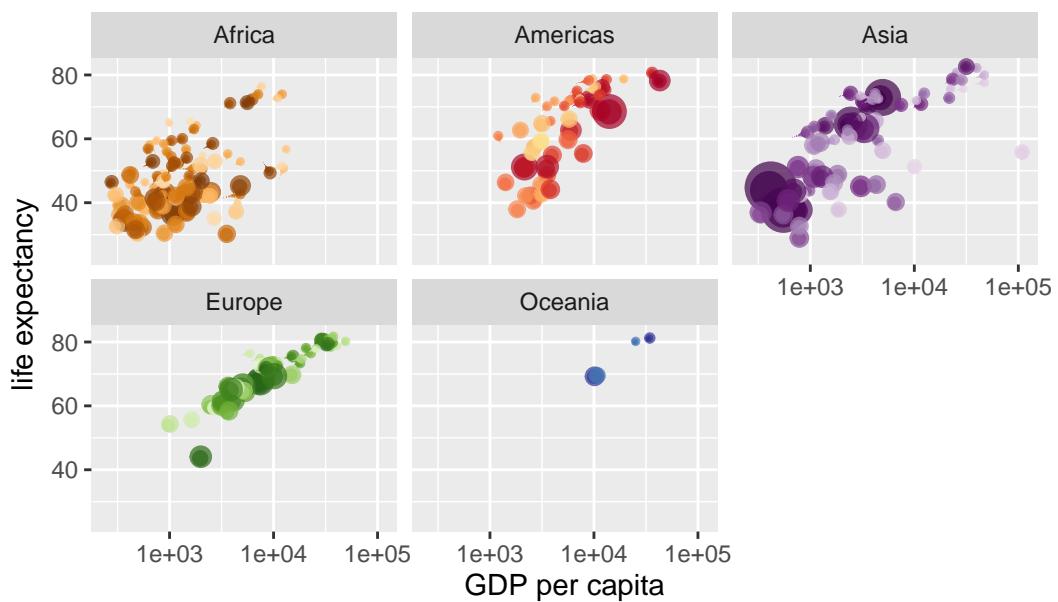
# Setup nice regular ggplot of the gapminder data
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  # Facet by continent
  facet_wrap(~continent) +
  # Here comes the gganimate specific bits
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
  transition_time(year) +
  shadow_wake(wake_length = 0.1, alpha = FALSE)
```

Warning in formals(fun): argument is not a function

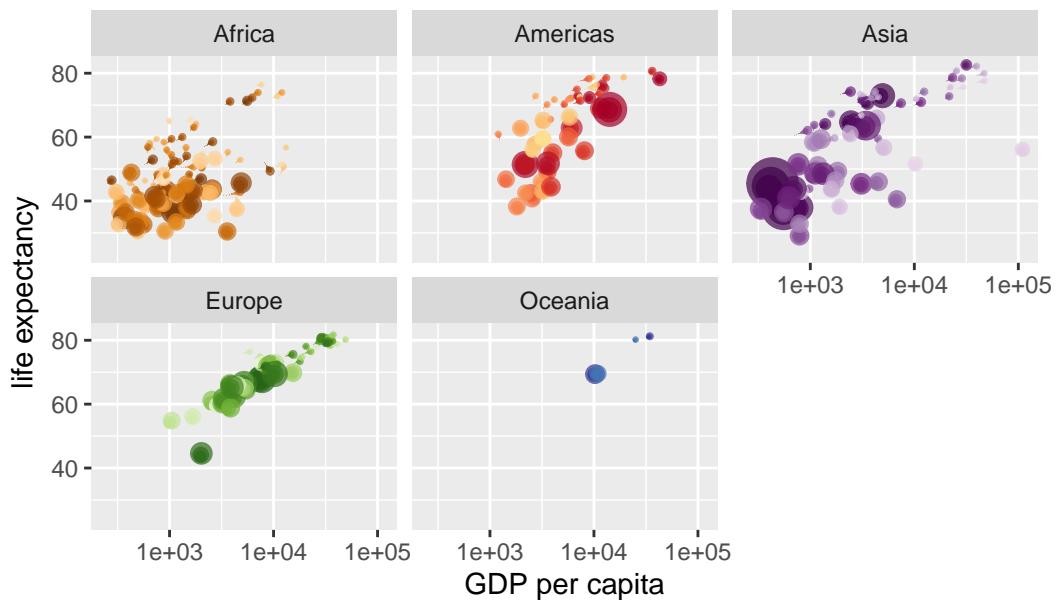
Year: 1952



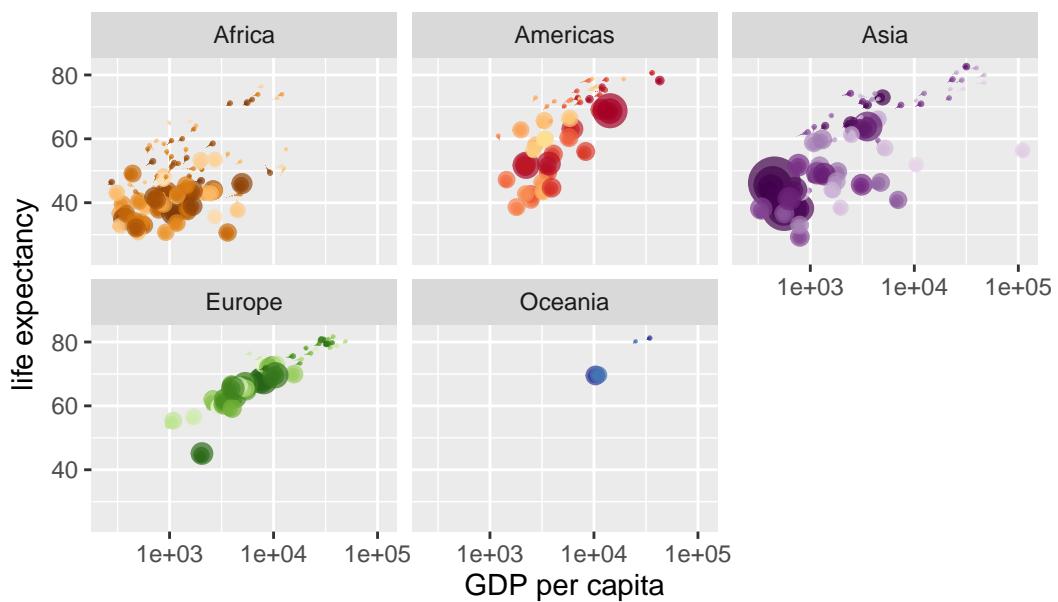
Year: 1953



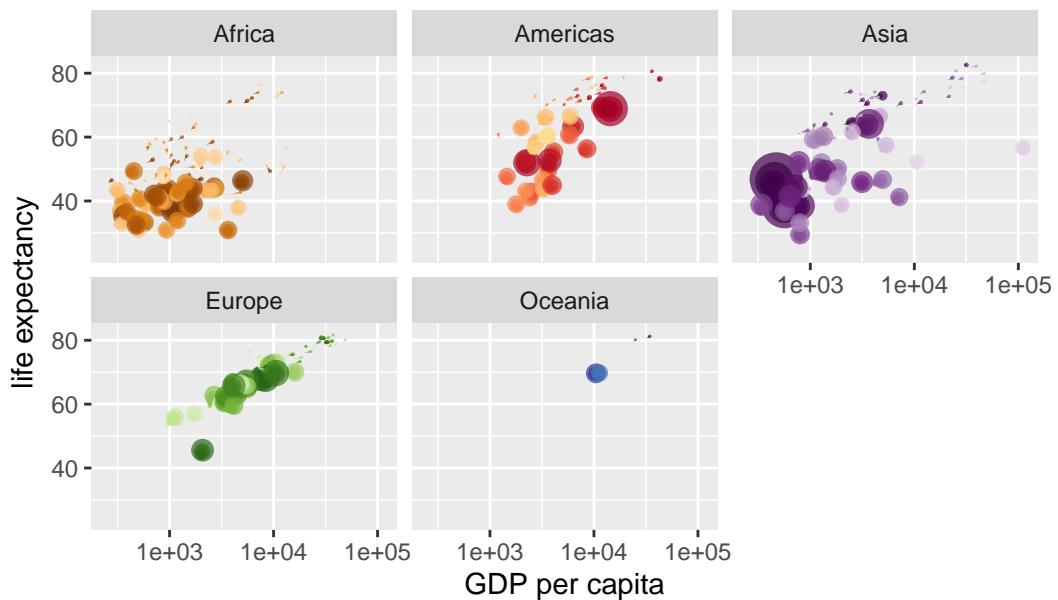
Year: 1953



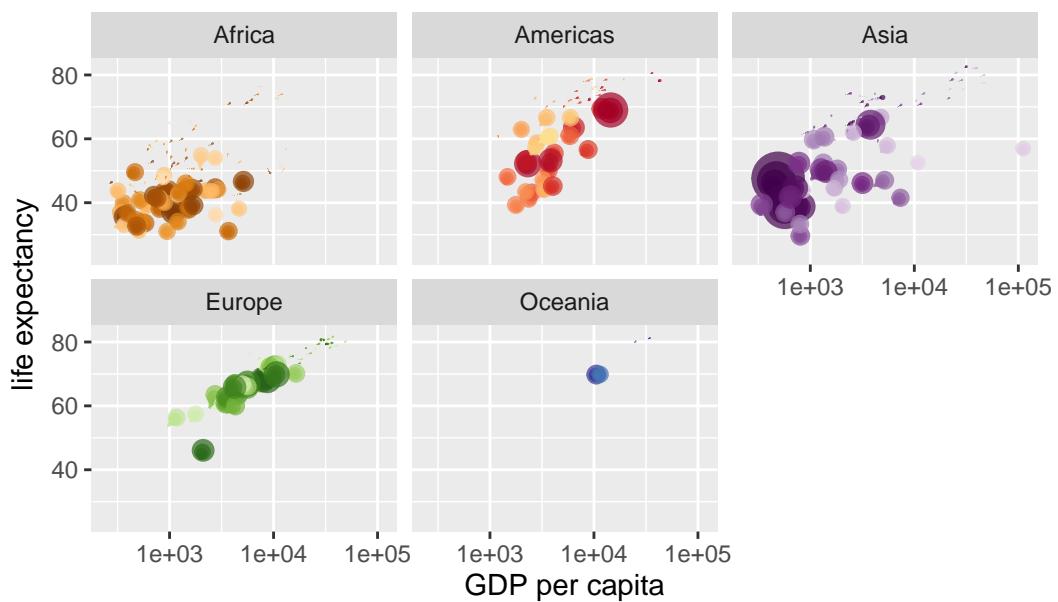
Year: 1954



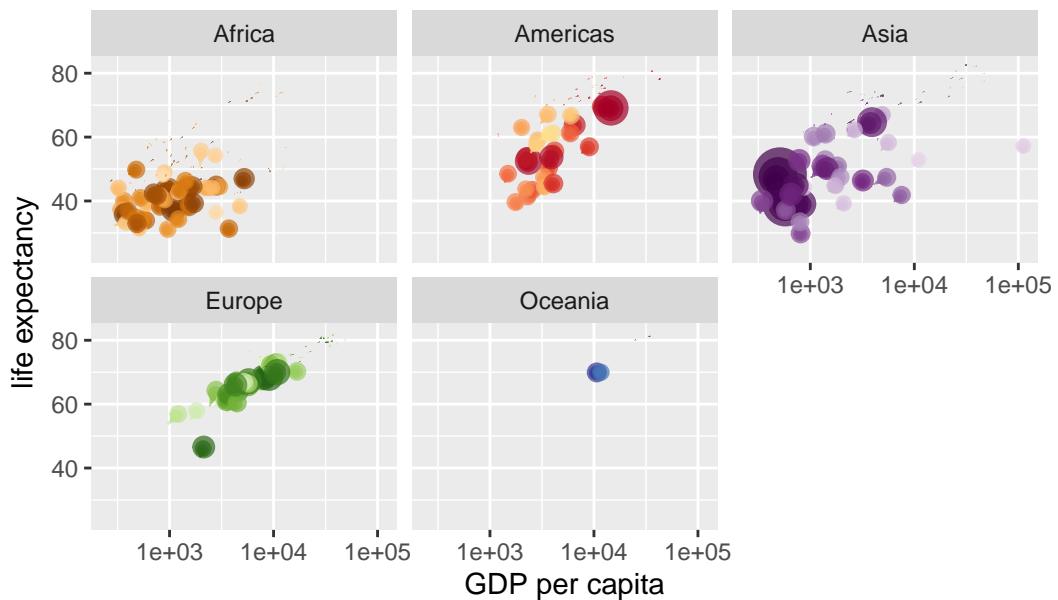
Year: 1954



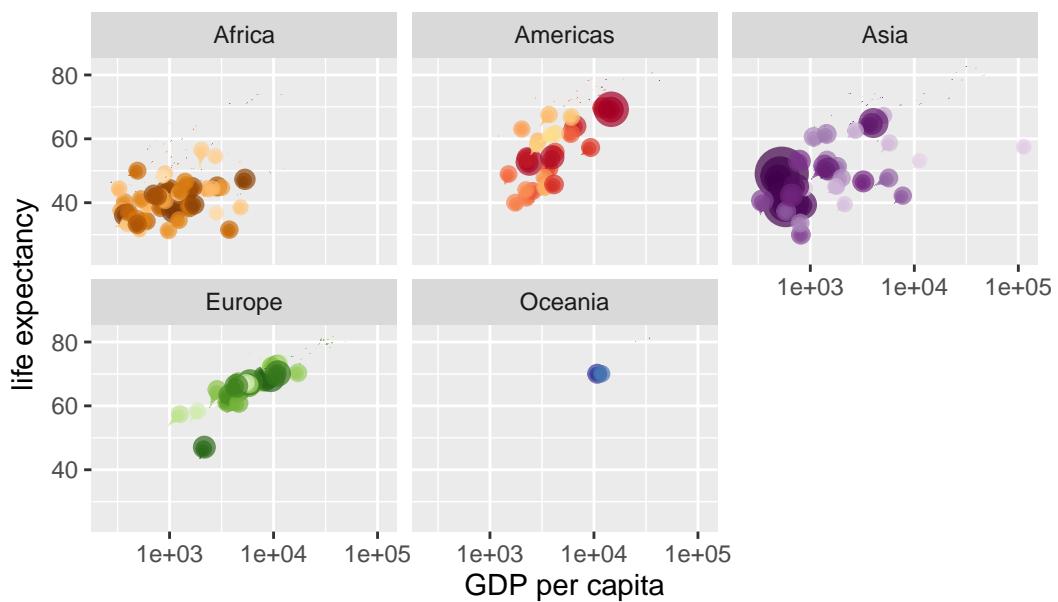
Year: 1955



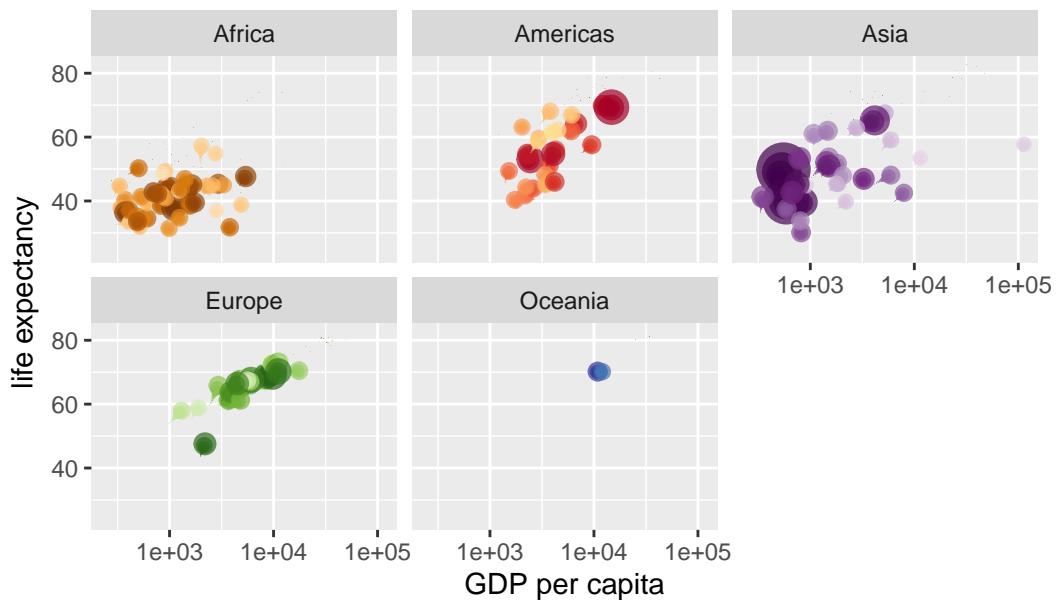
Year: 1955



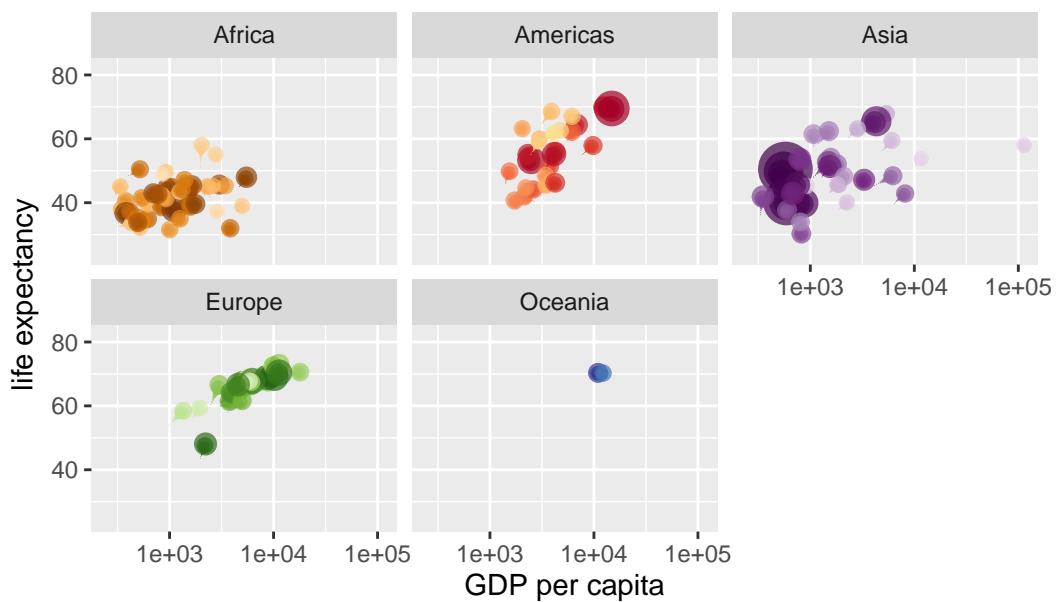
Year: 1956



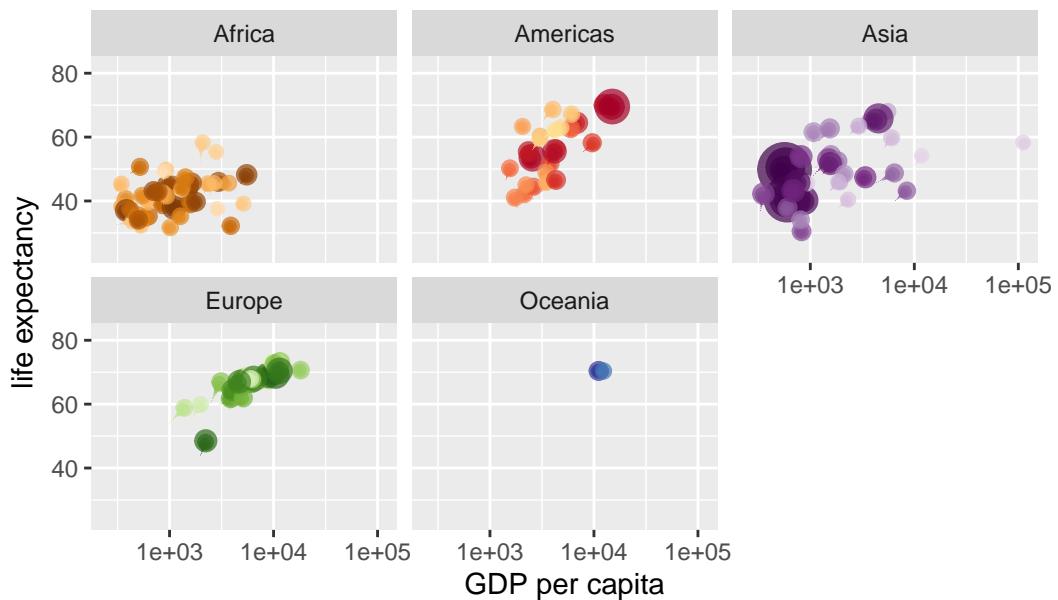
Year: 1956



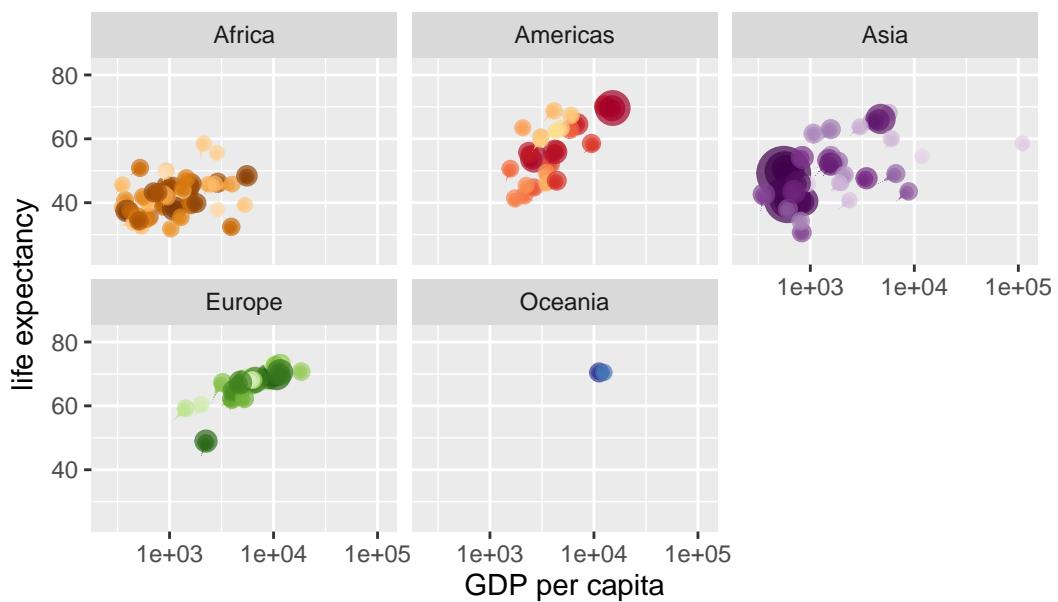
Year: 1957



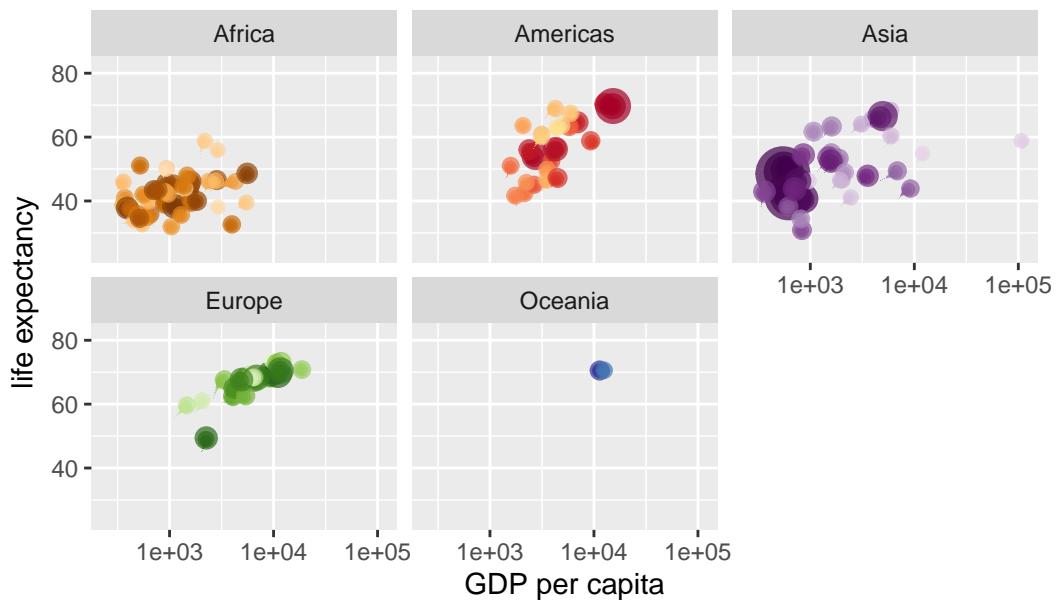
Year: 1958



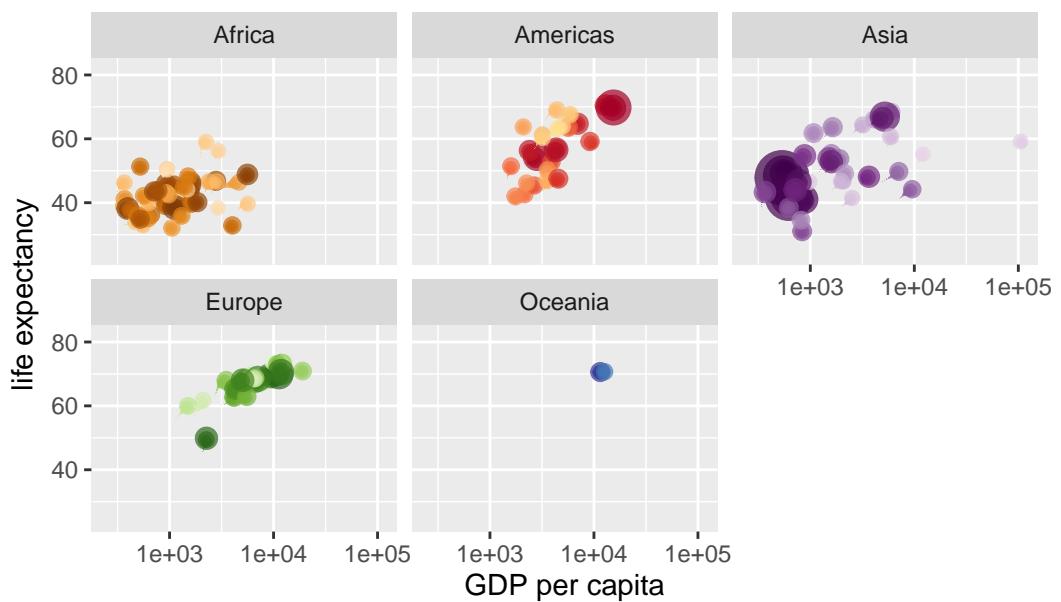
Year: 1958



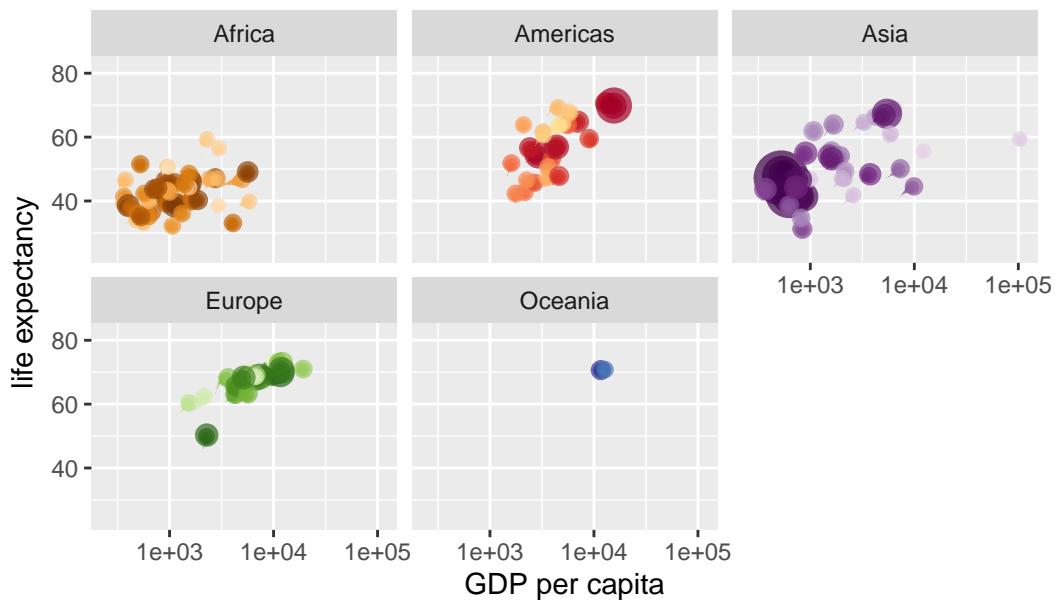
Year: 1959



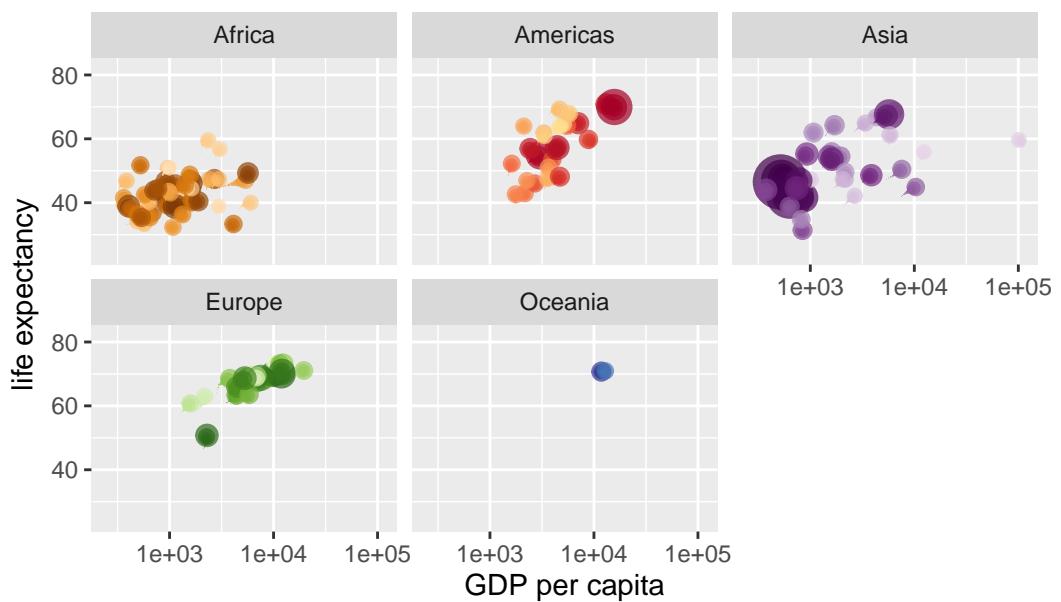
Year: 1959



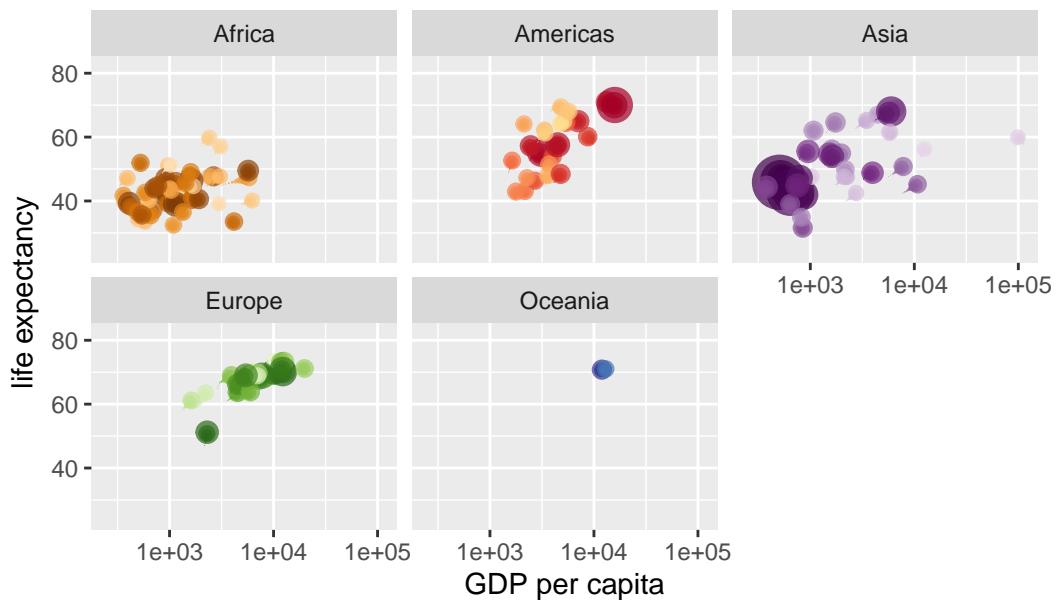
Year: 1960



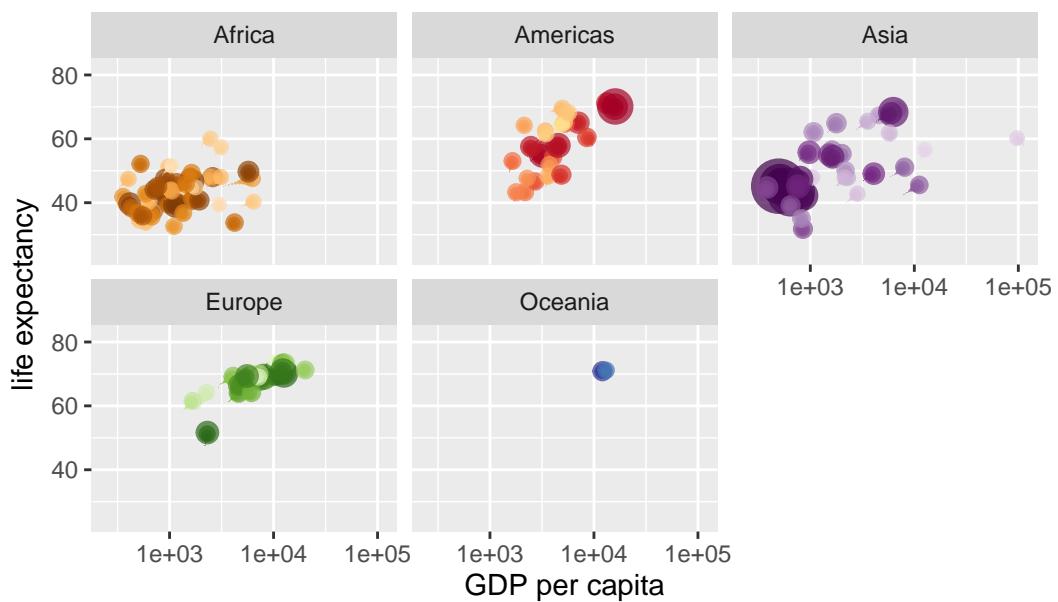
Year: 1960



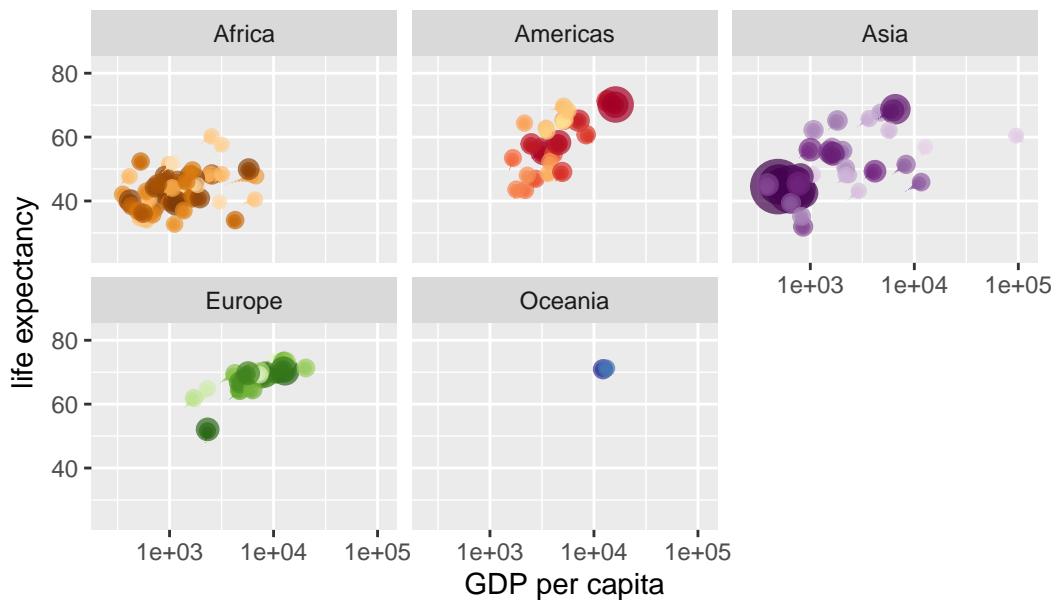
Year: 1961



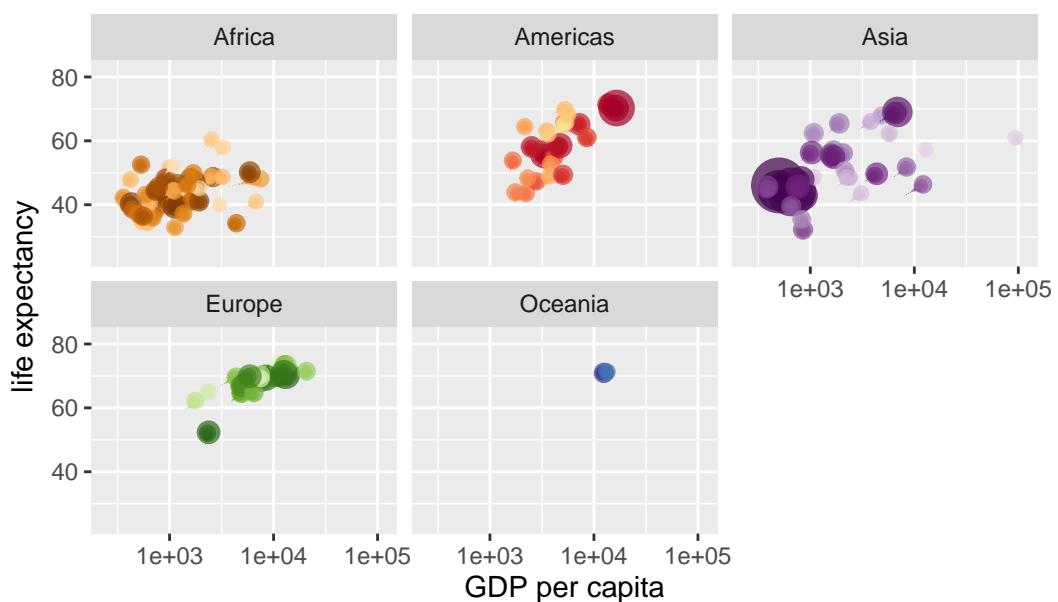
Year: 1961



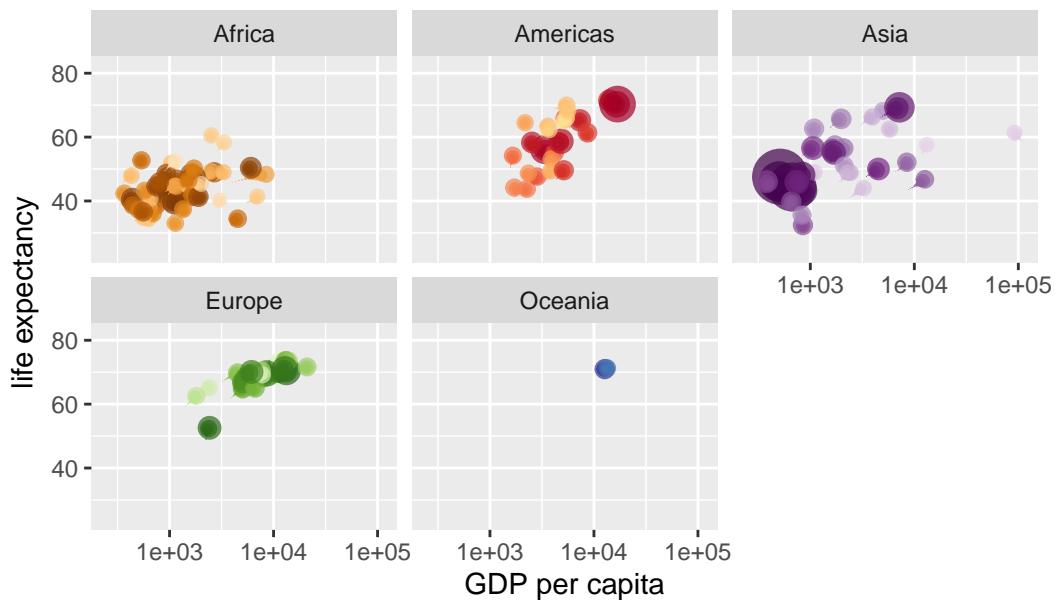
Year: 1962



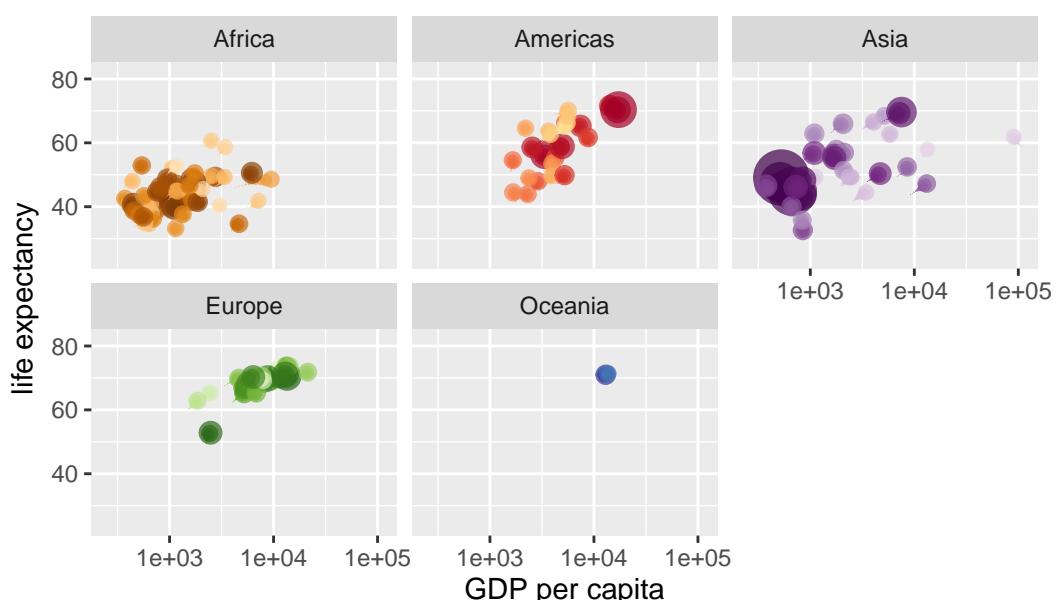
Year: 1963



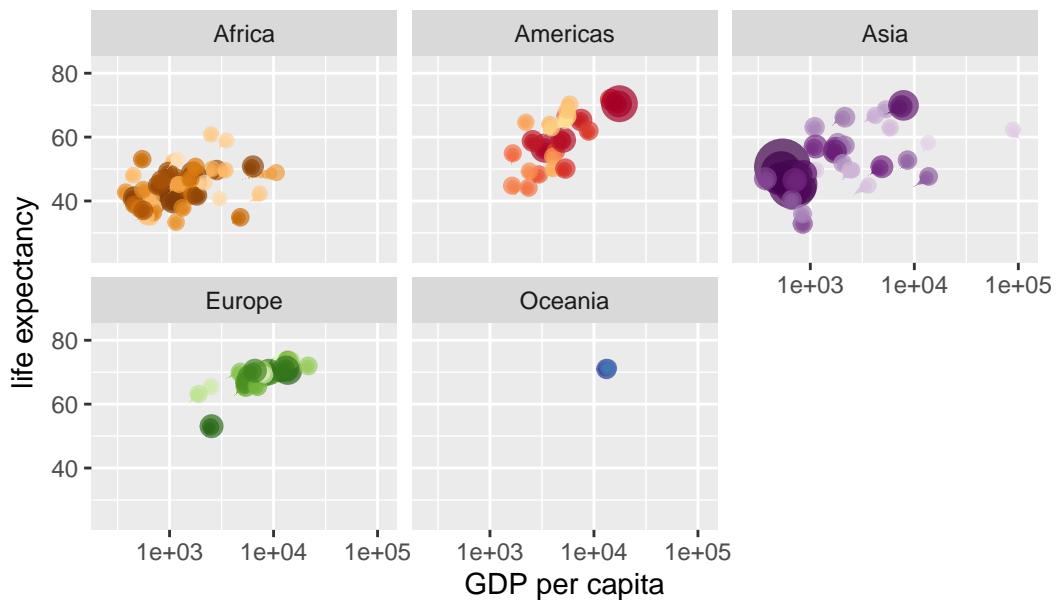
Year: 1963



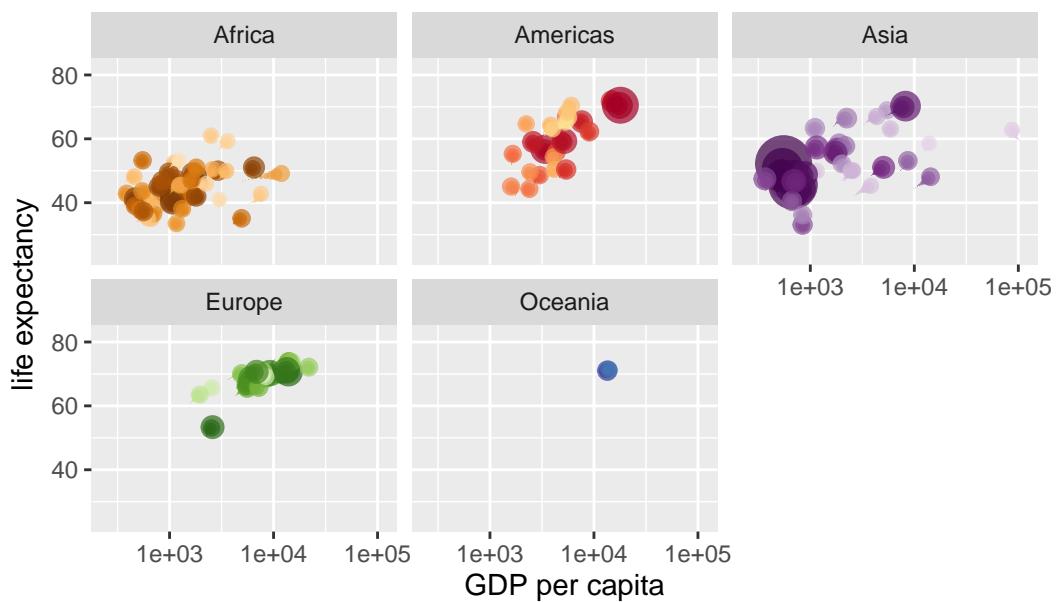
Year: 1964



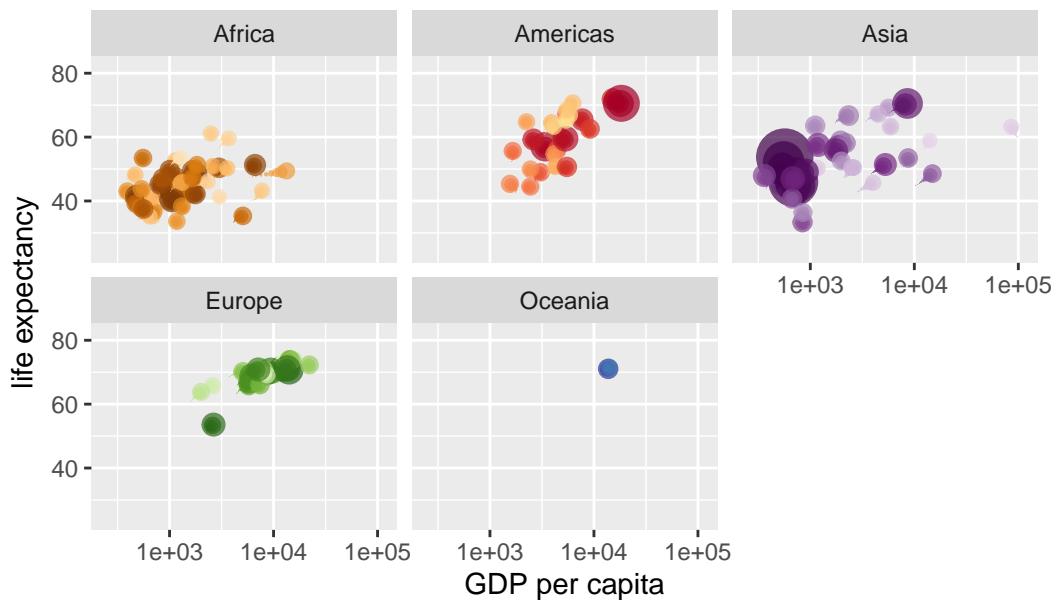
Year: 1964



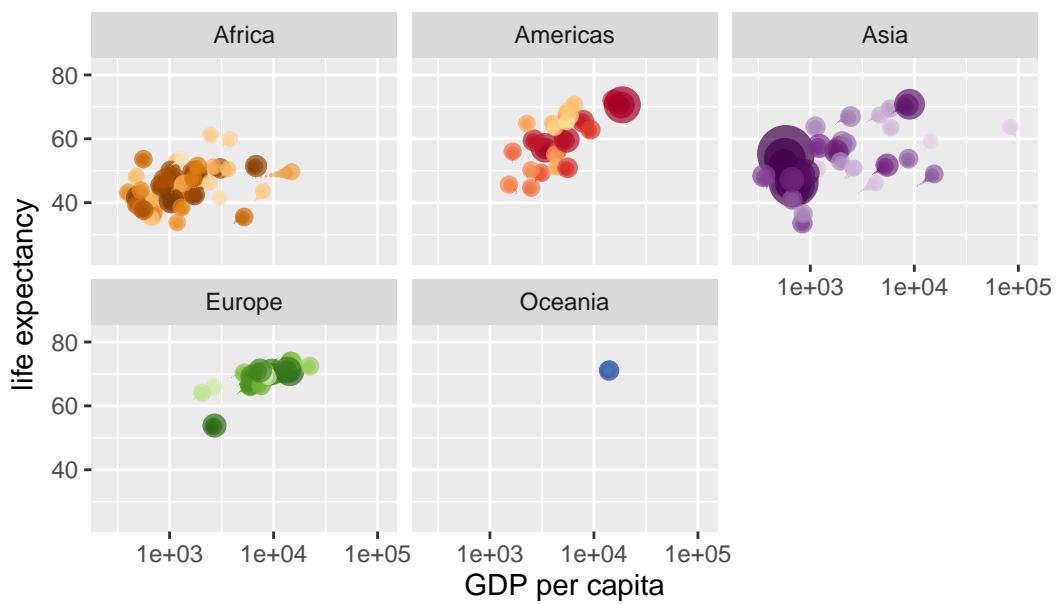
Year: 1965



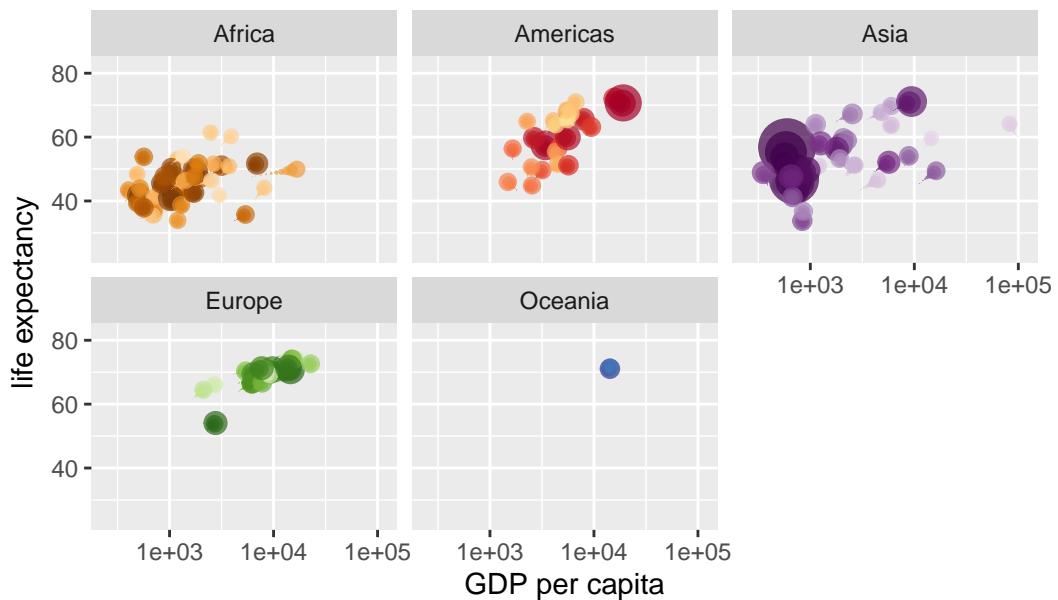
Year: 1965



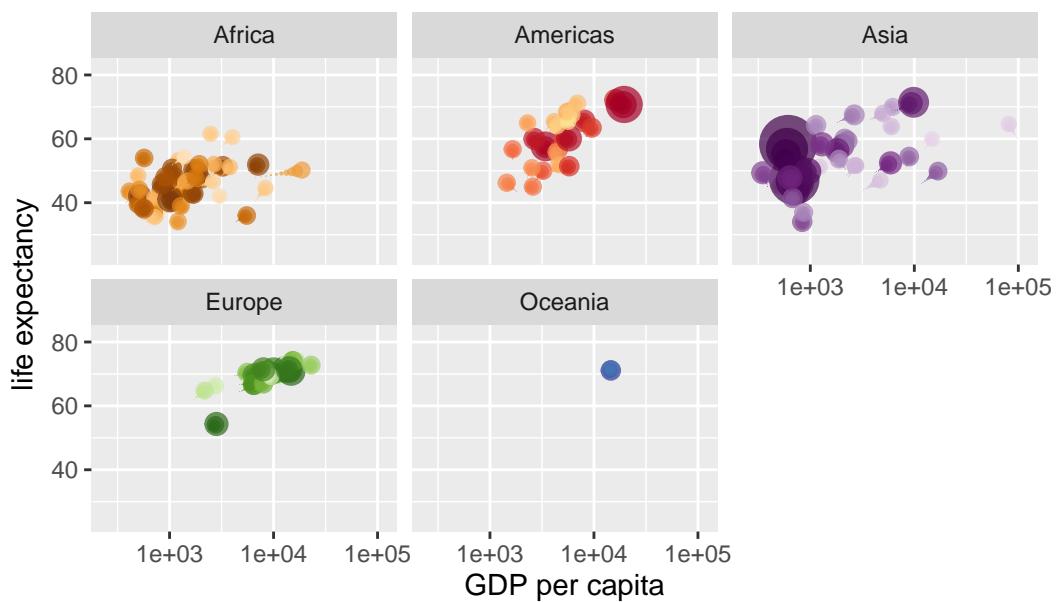
Year: 1966



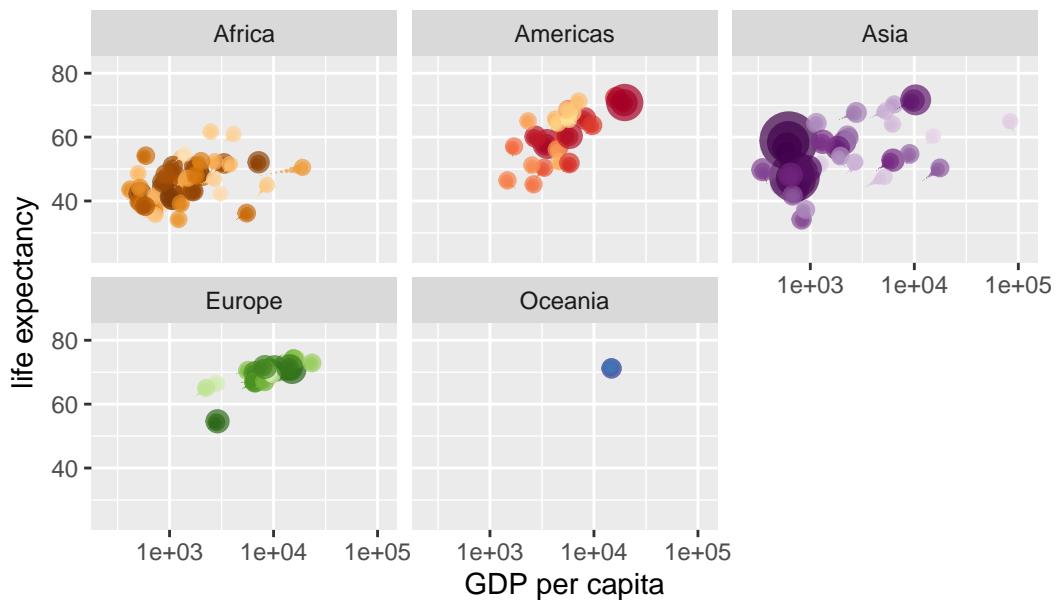
Year: 1966



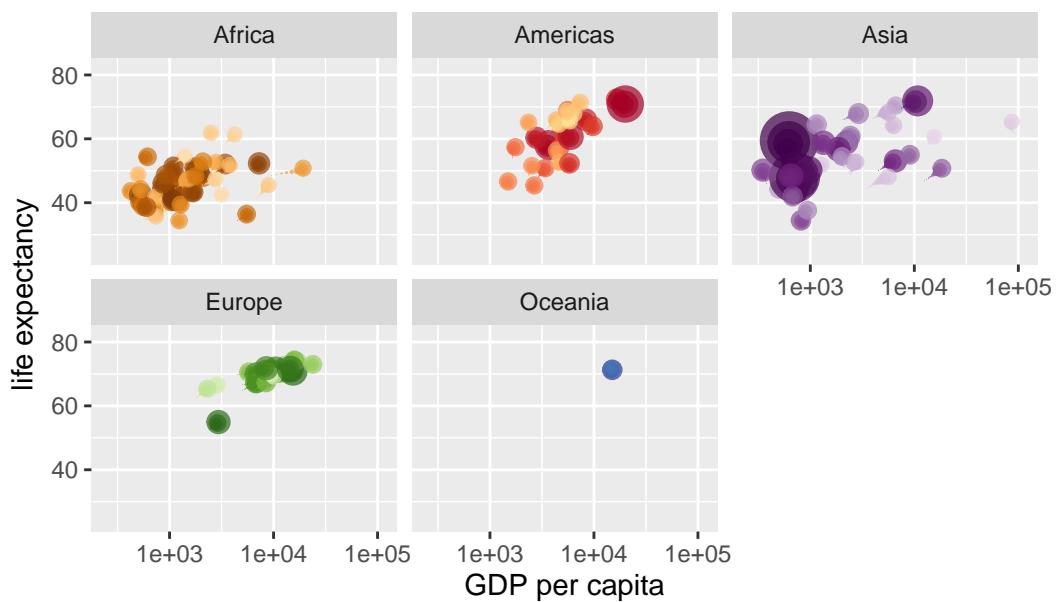
Year: 1967



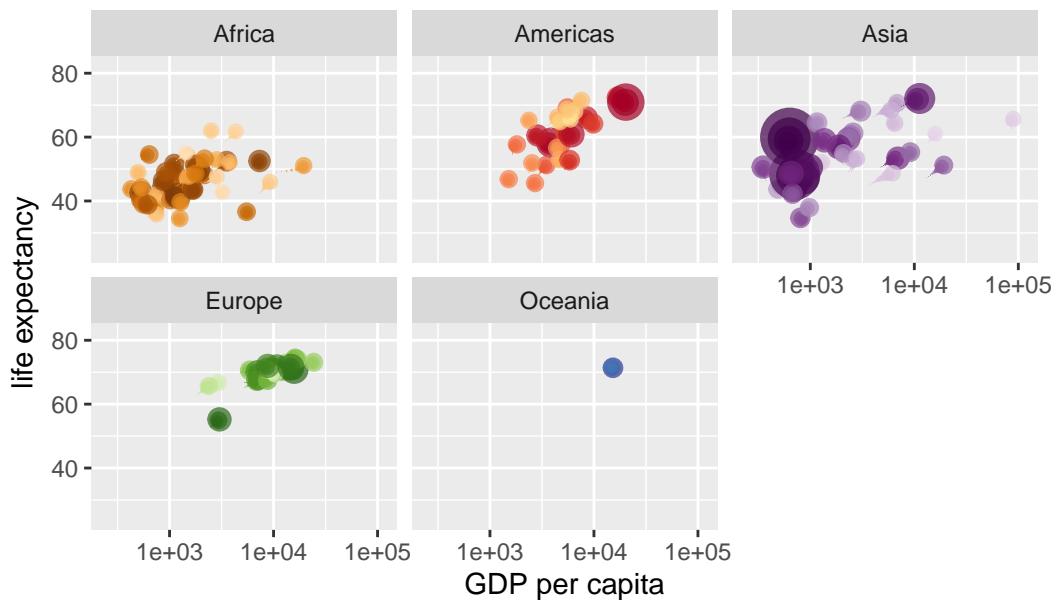
Year: 1968



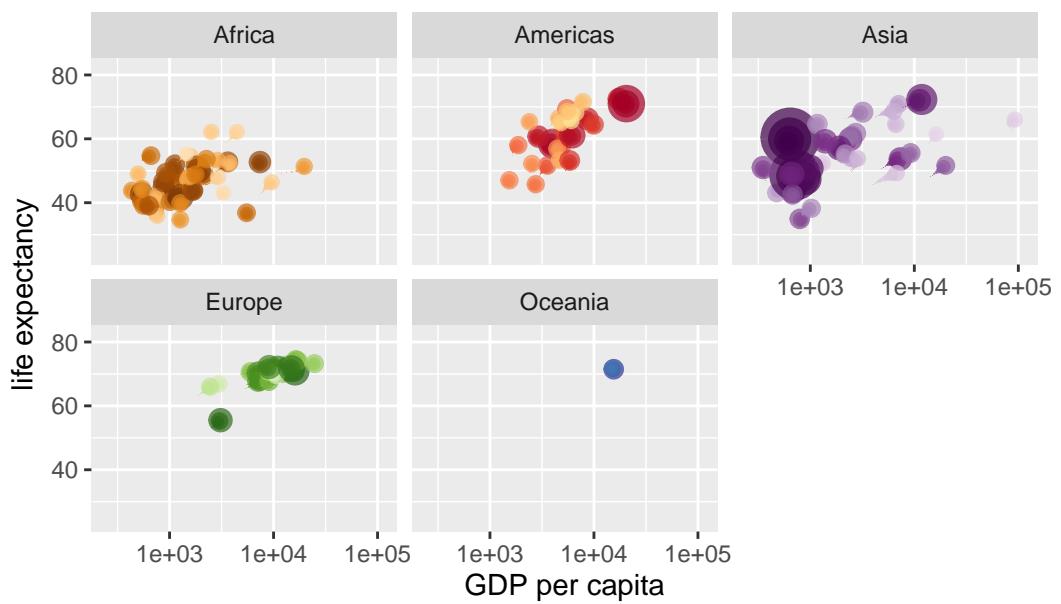
Year: 1968



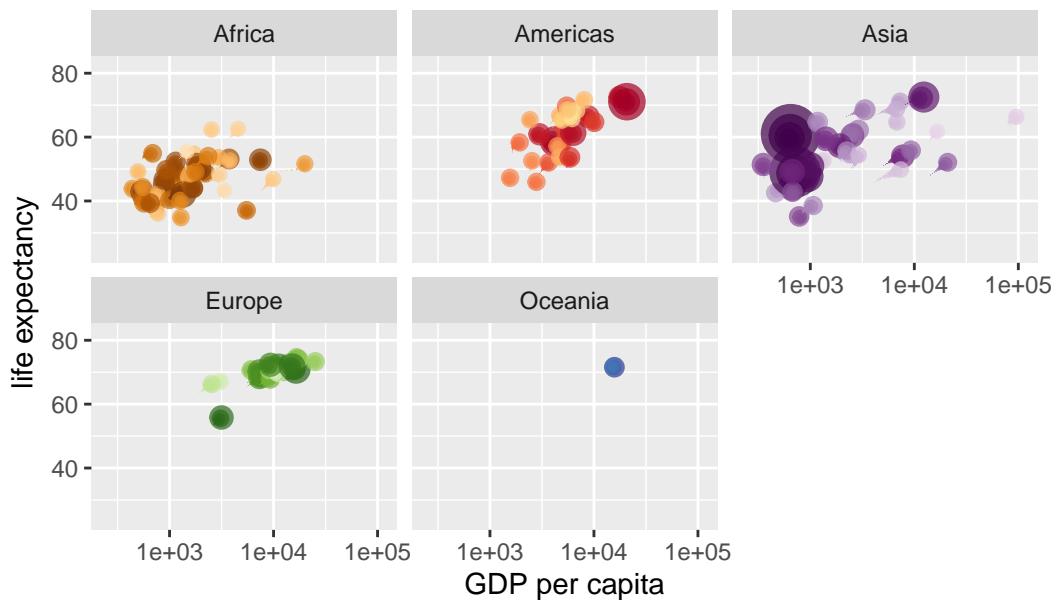
Year: 1969



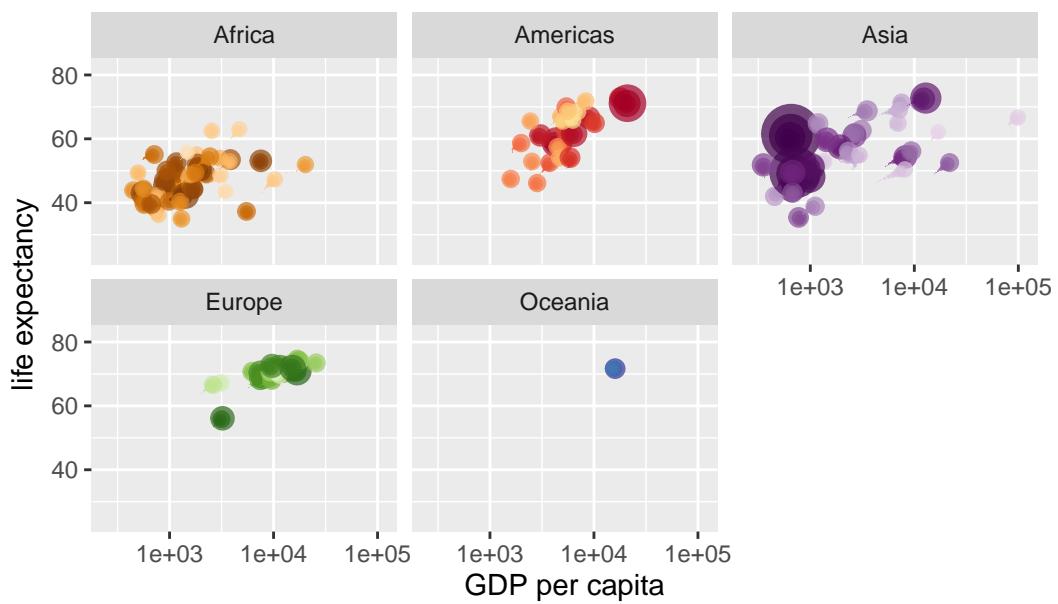
Year: 1969



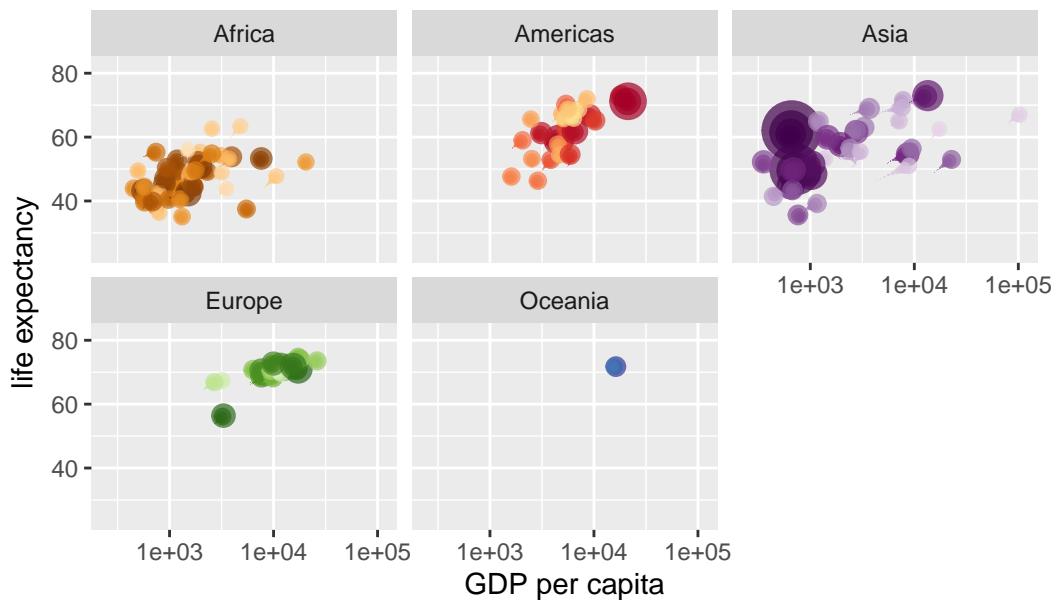
Year: 1970



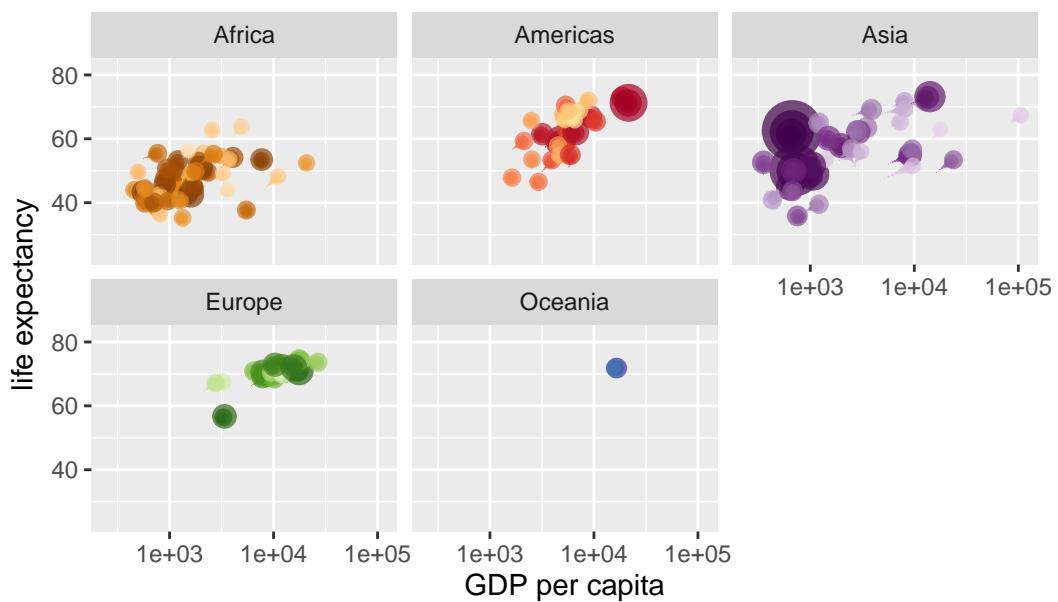
Year: 1970



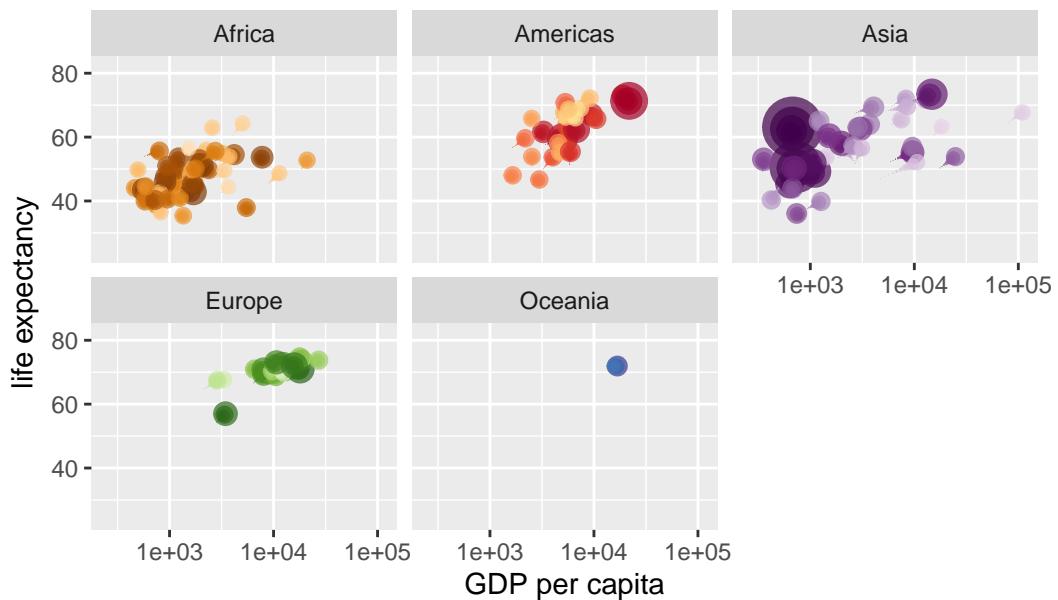
Year: 1971



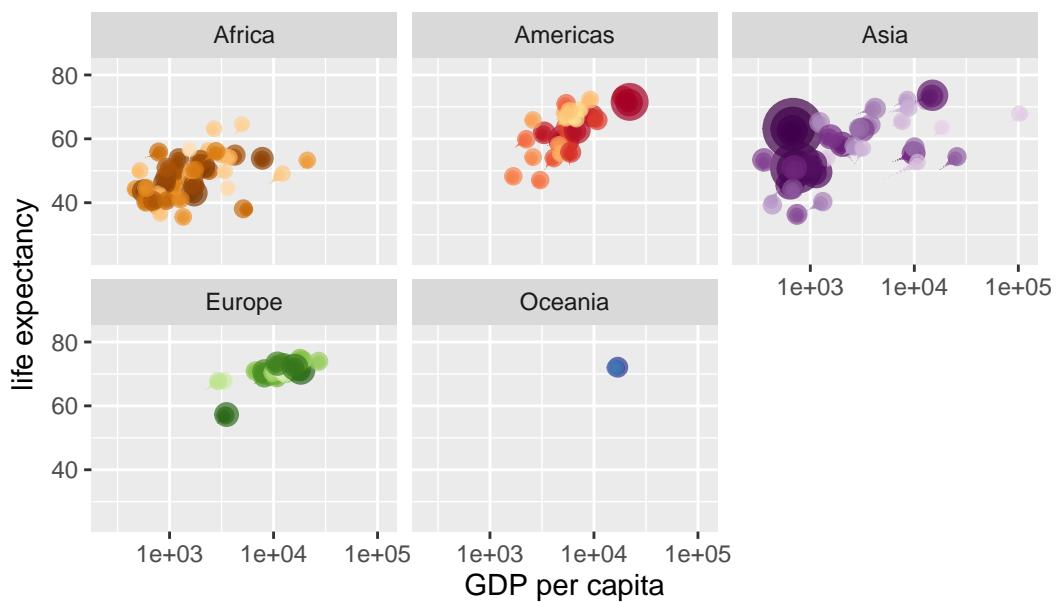
Year: 1971



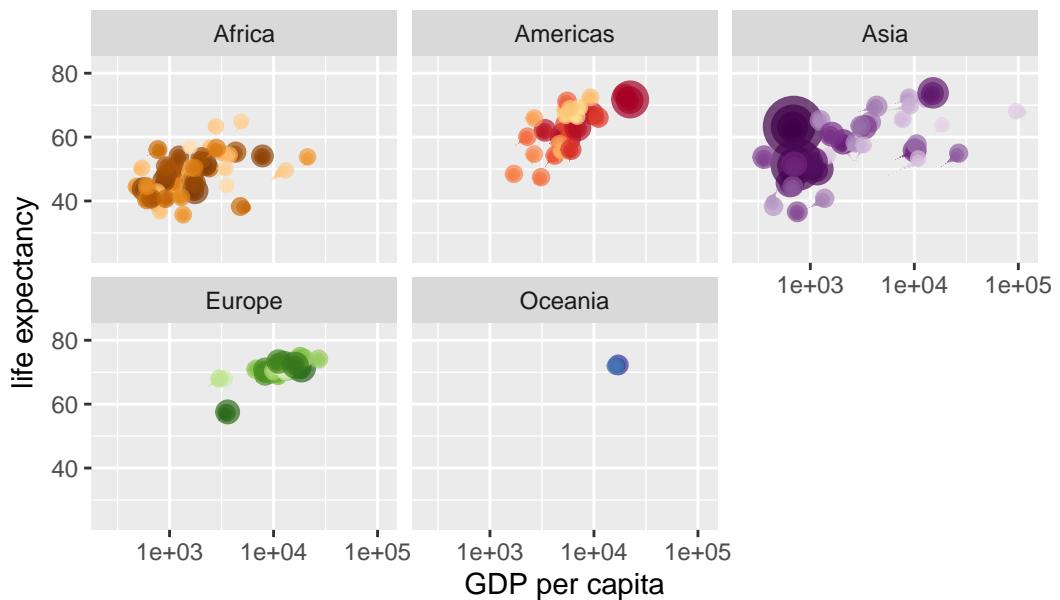
Year: 1972



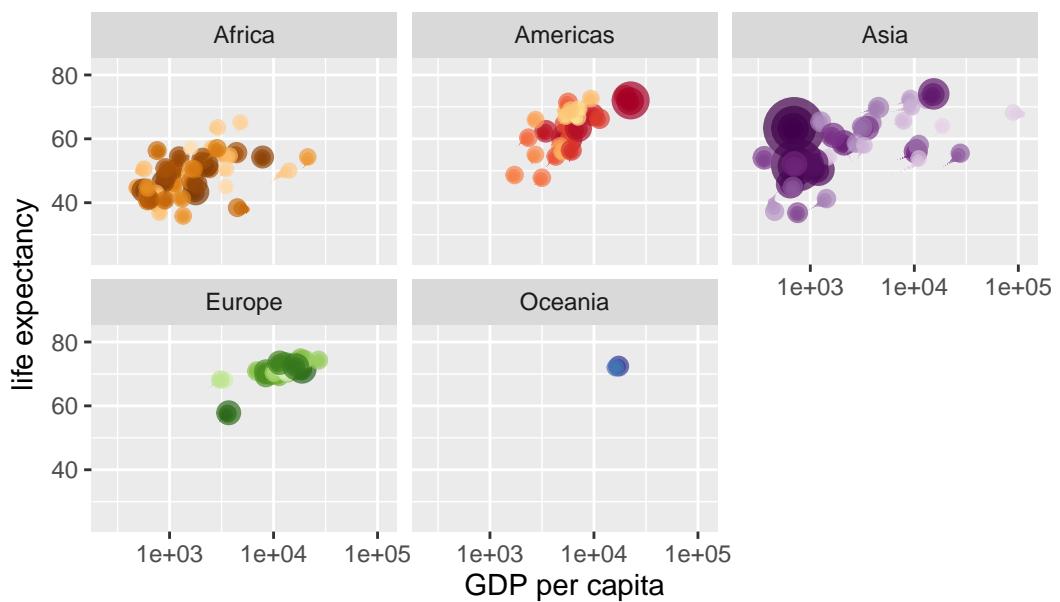
Year: 1973



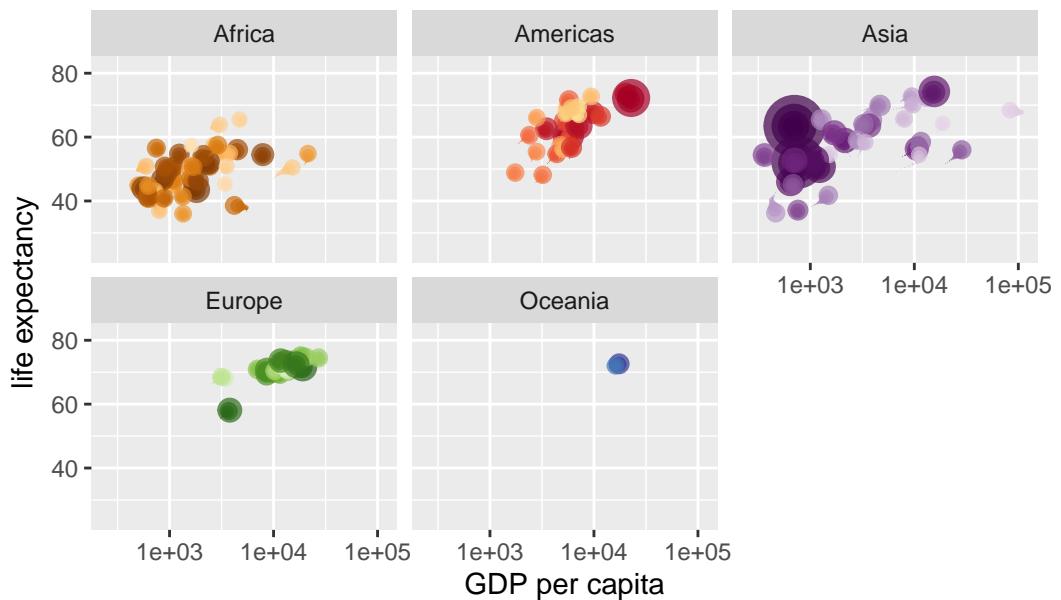
Year: 1973



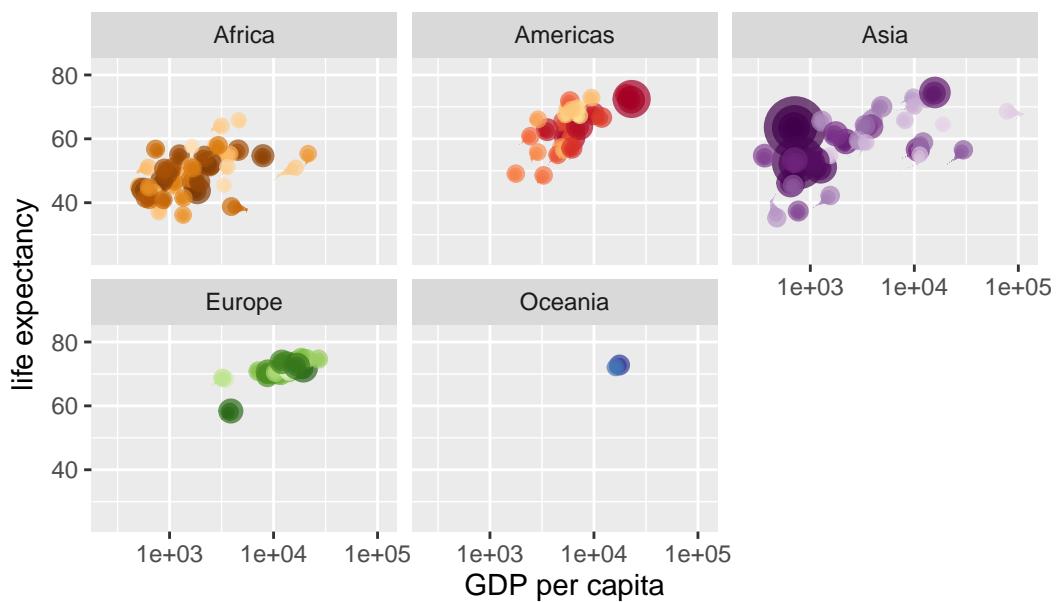
Year: 1974



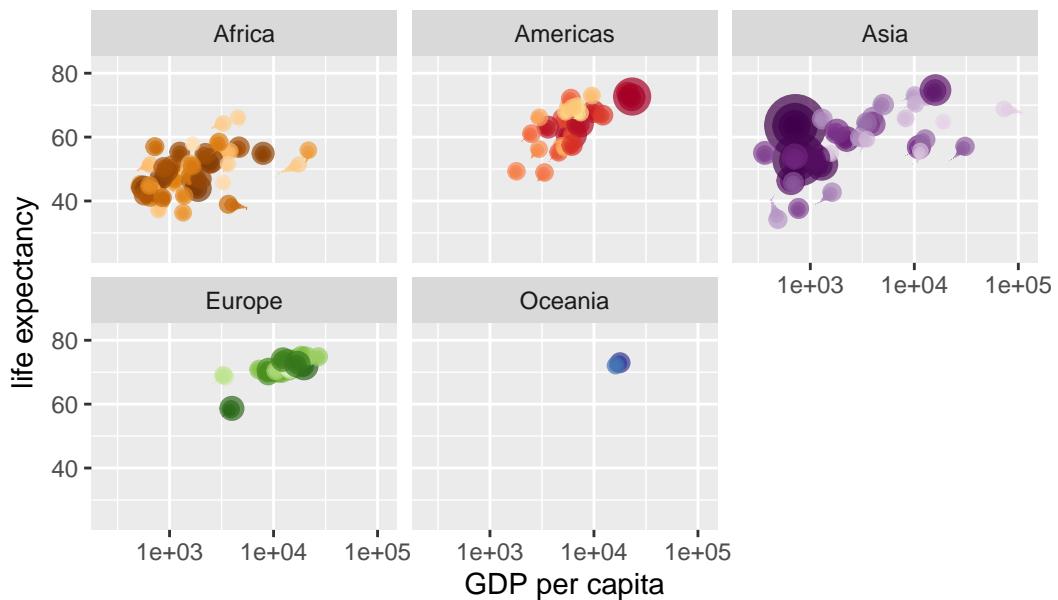
Year: 1974



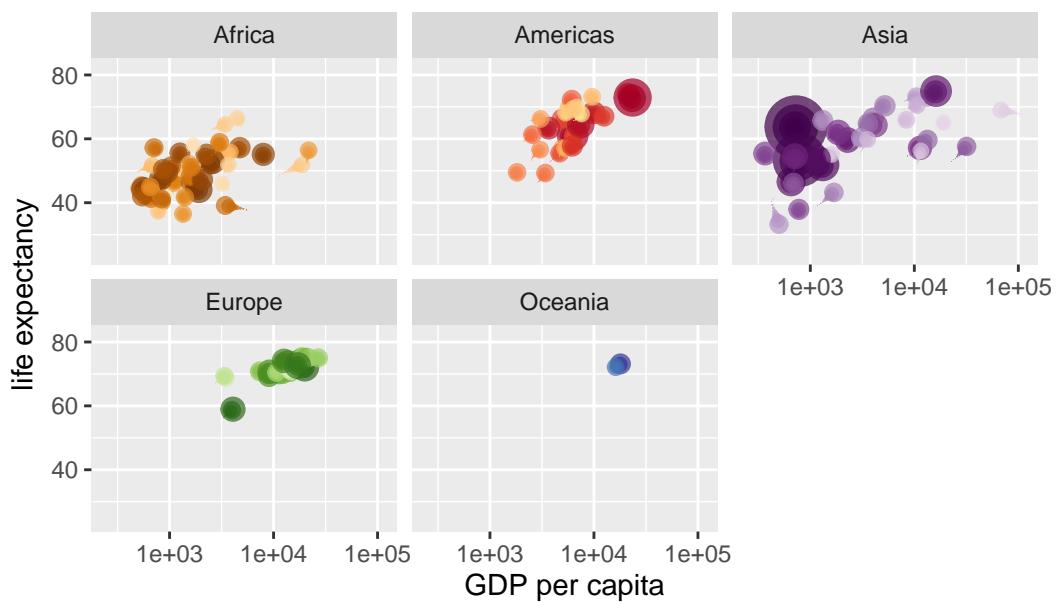
Year: 1975



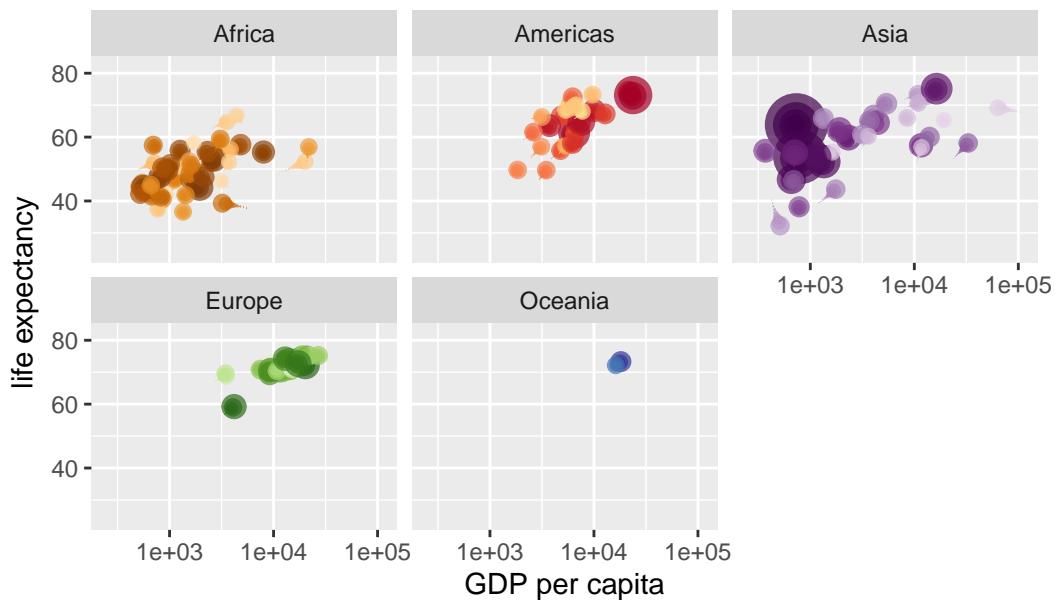
Year: 1975



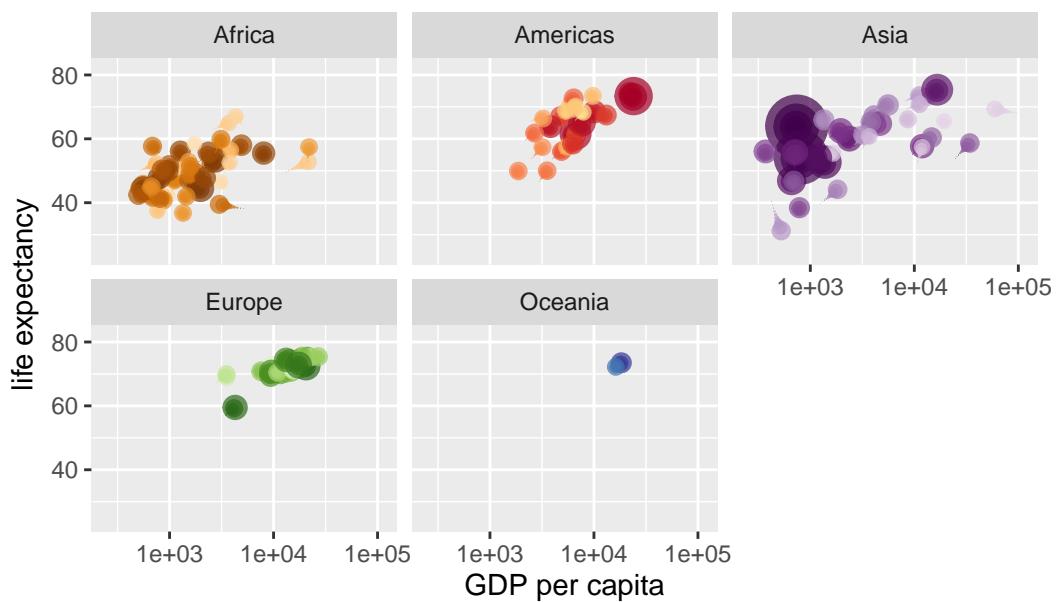
Year: 1976



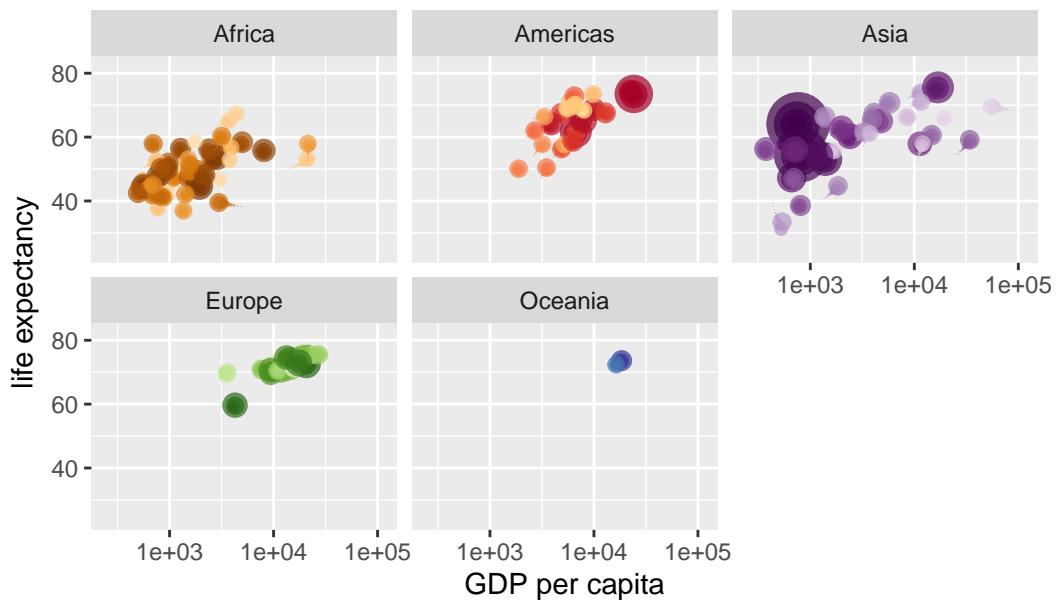
Year: 1976



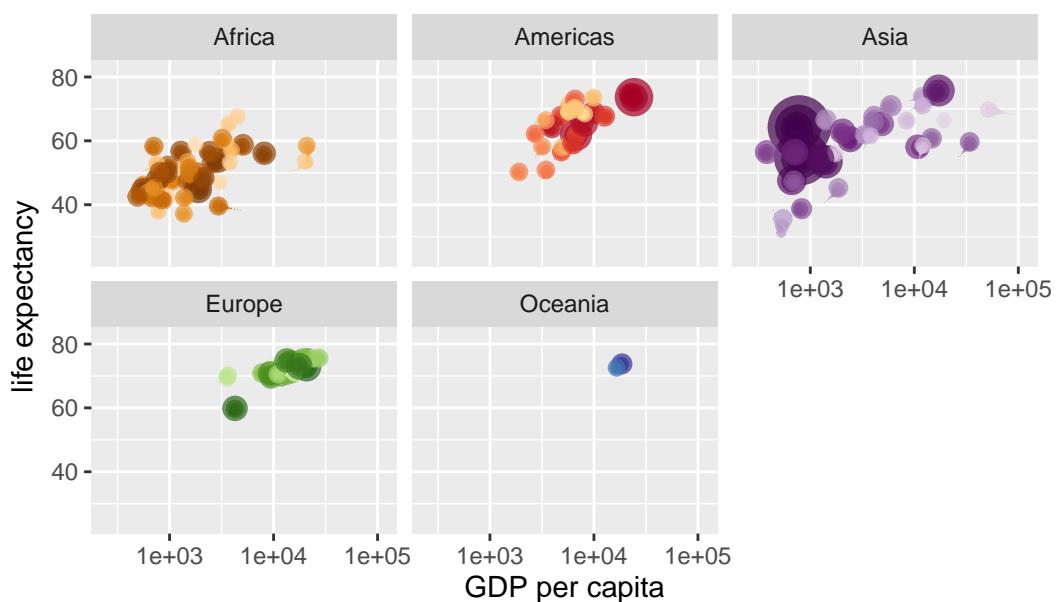
Year: 1977



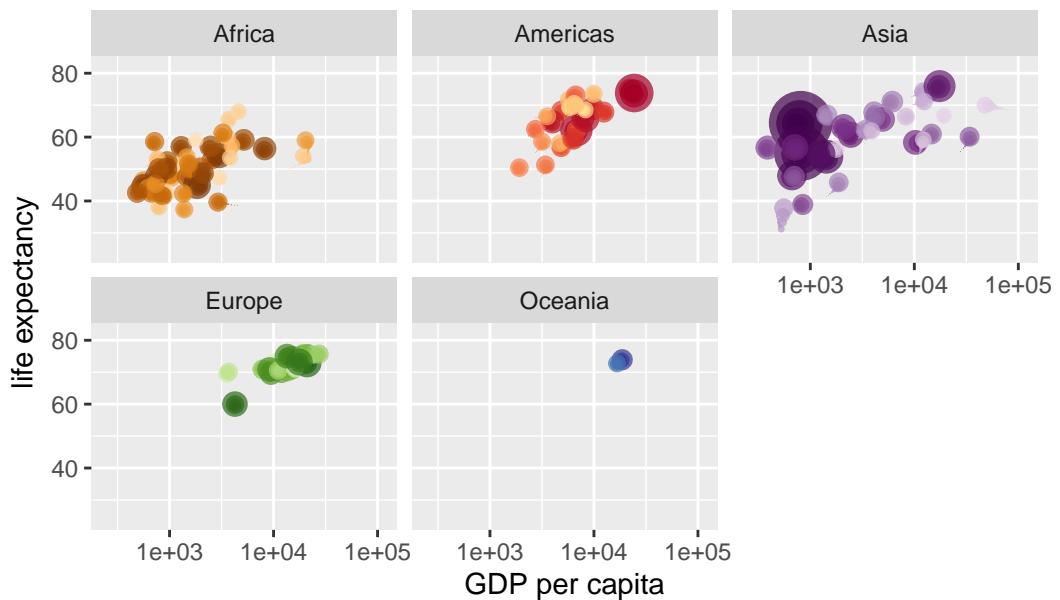
Year: 1978



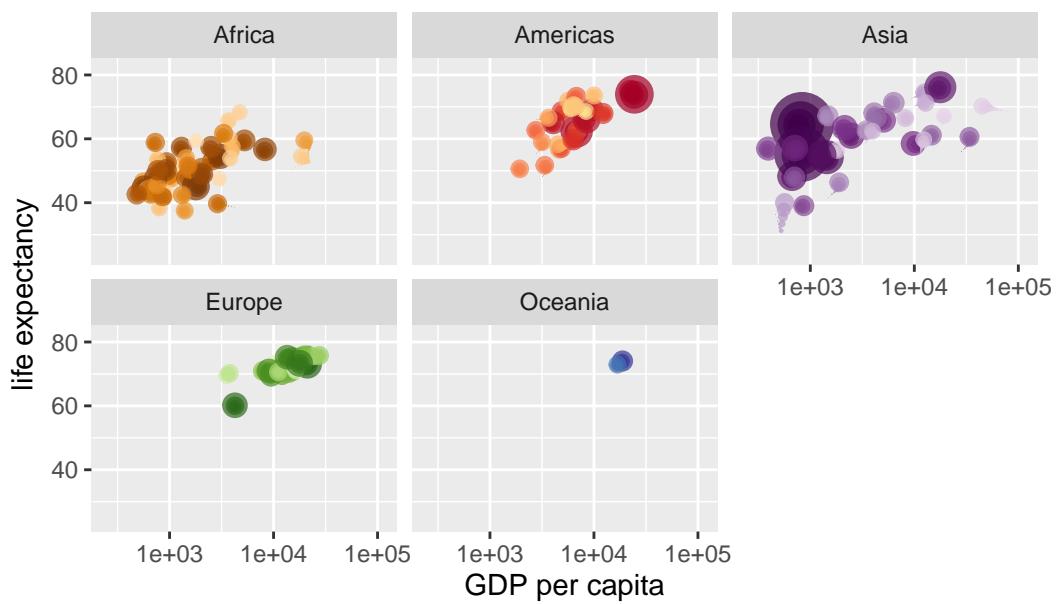
Year: 1978



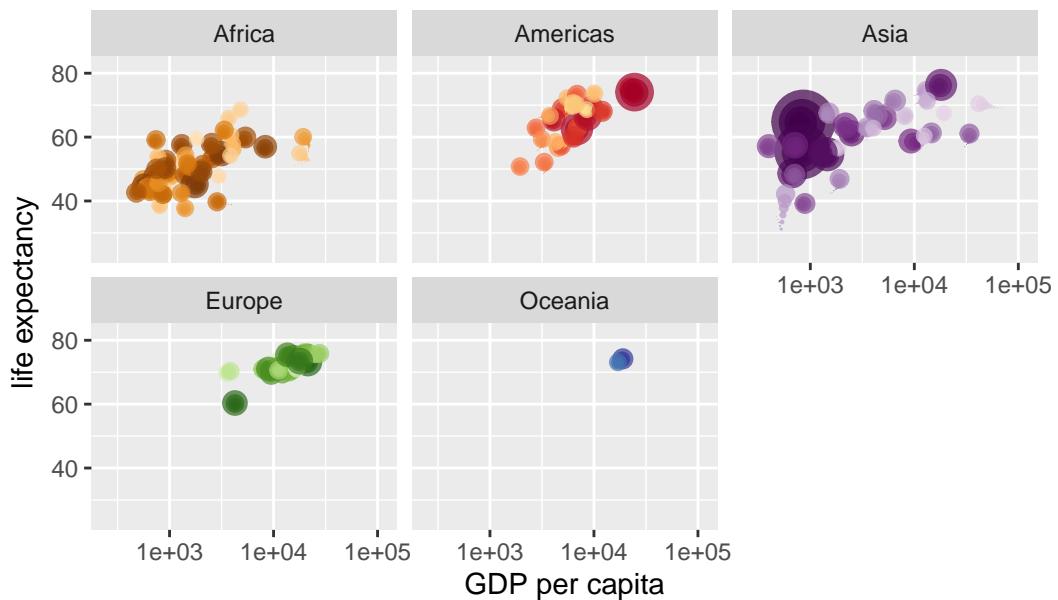
Year: 1979



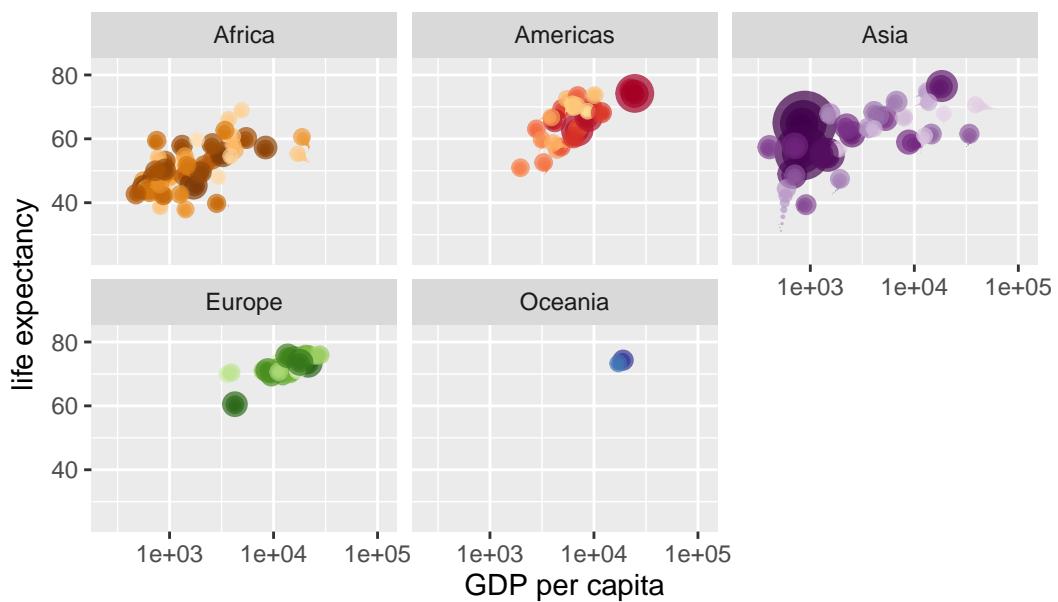
Year: 1979



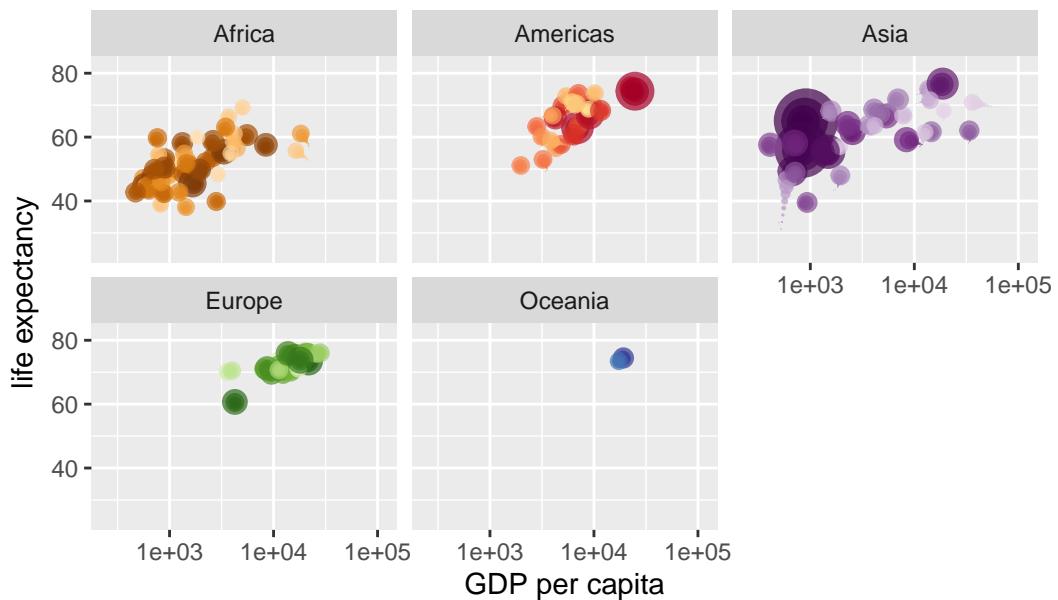
Year: 1980



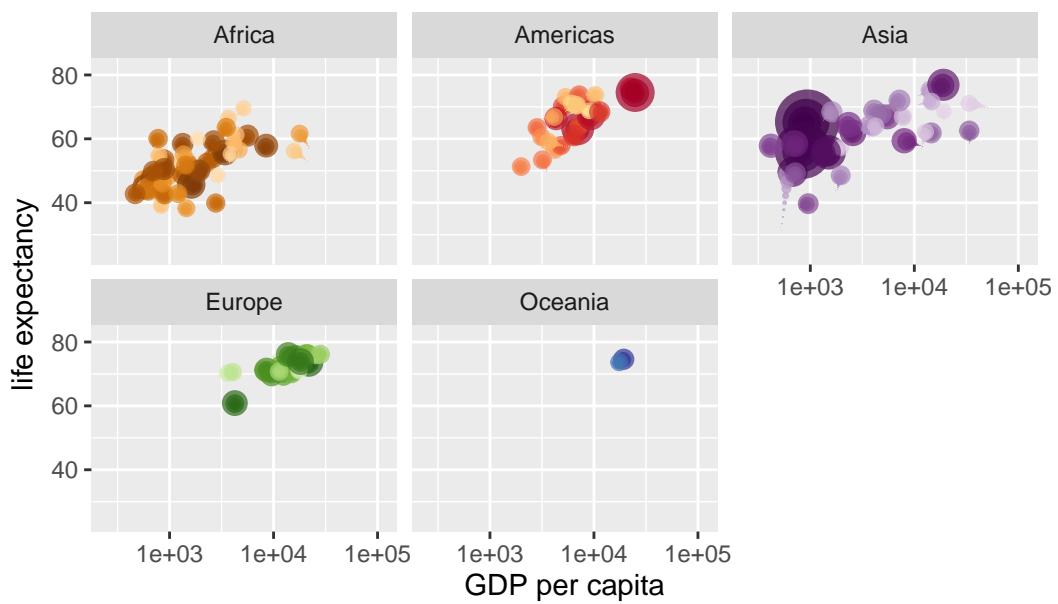
Year: 1980



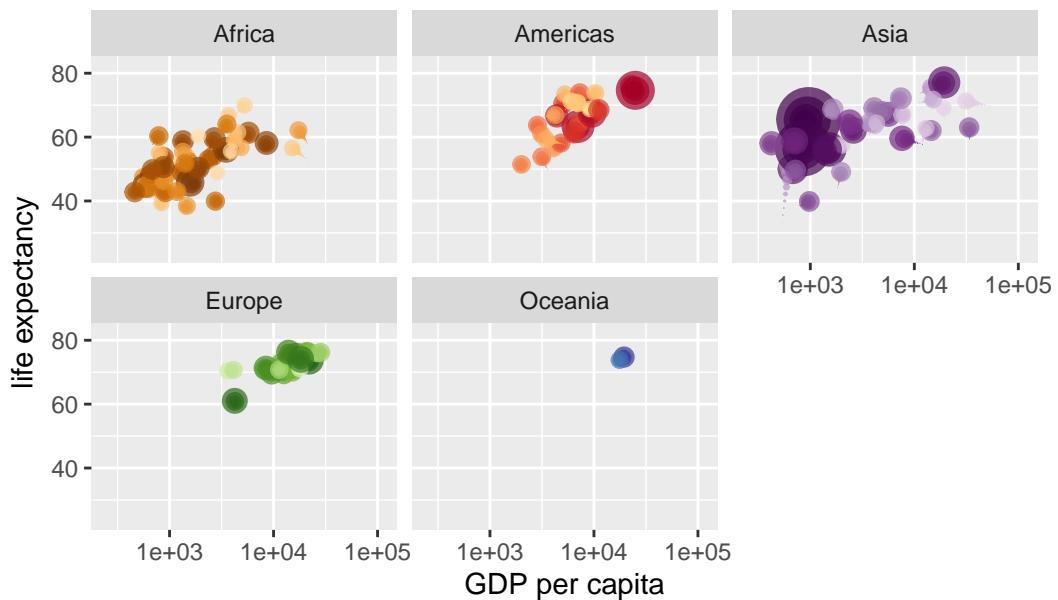
Year: 1981



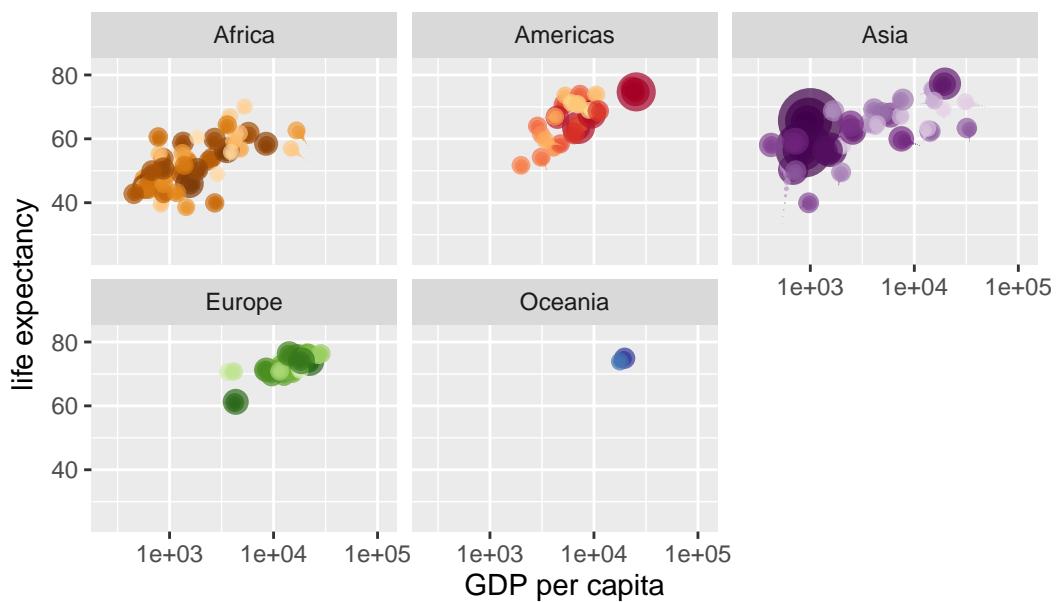
Year: 1981



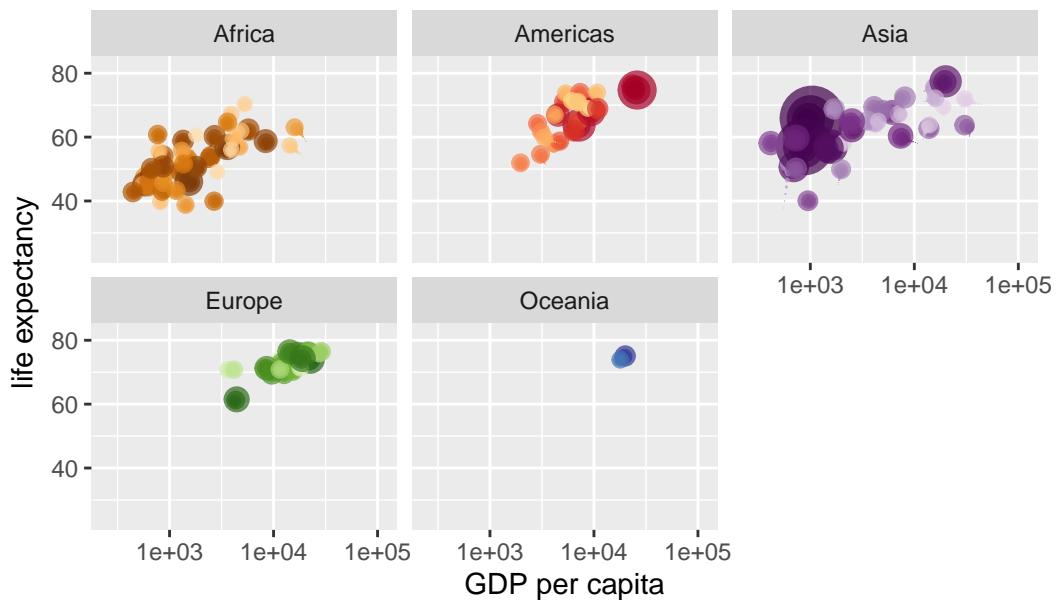
Year: 1982



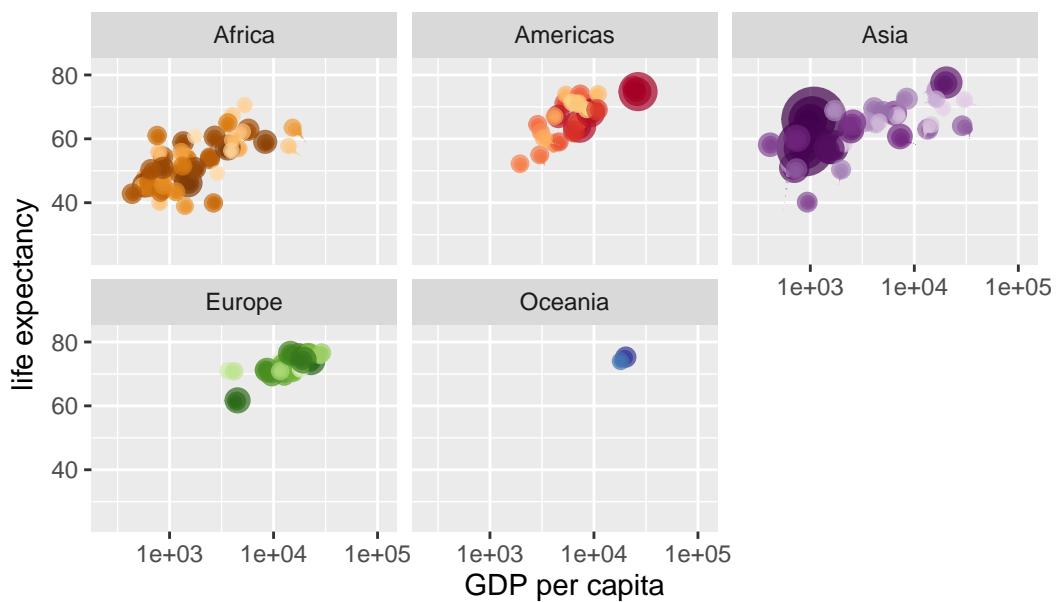
Year: 1983



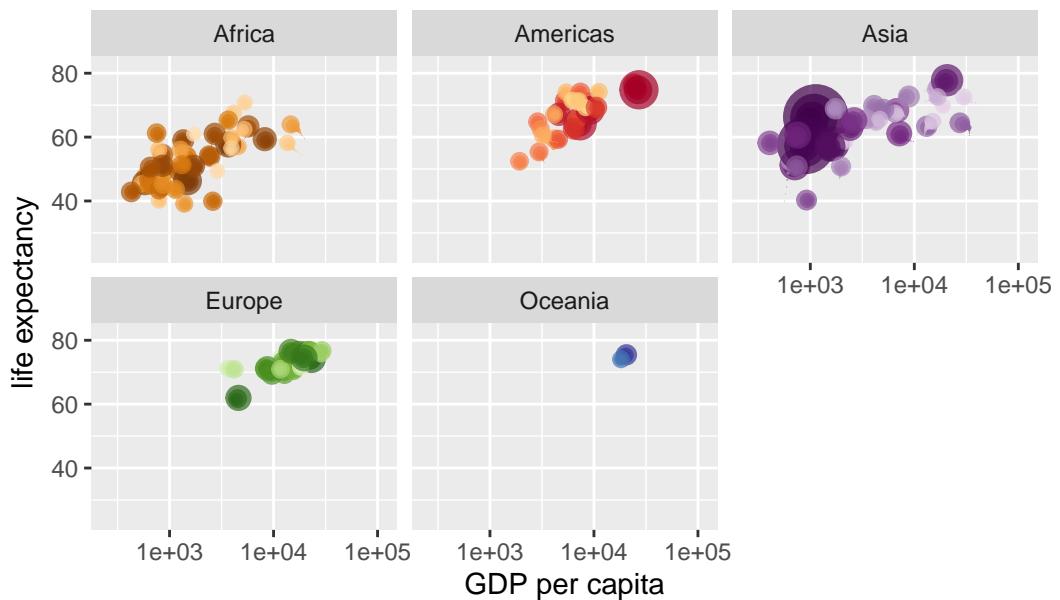
Year: 1983



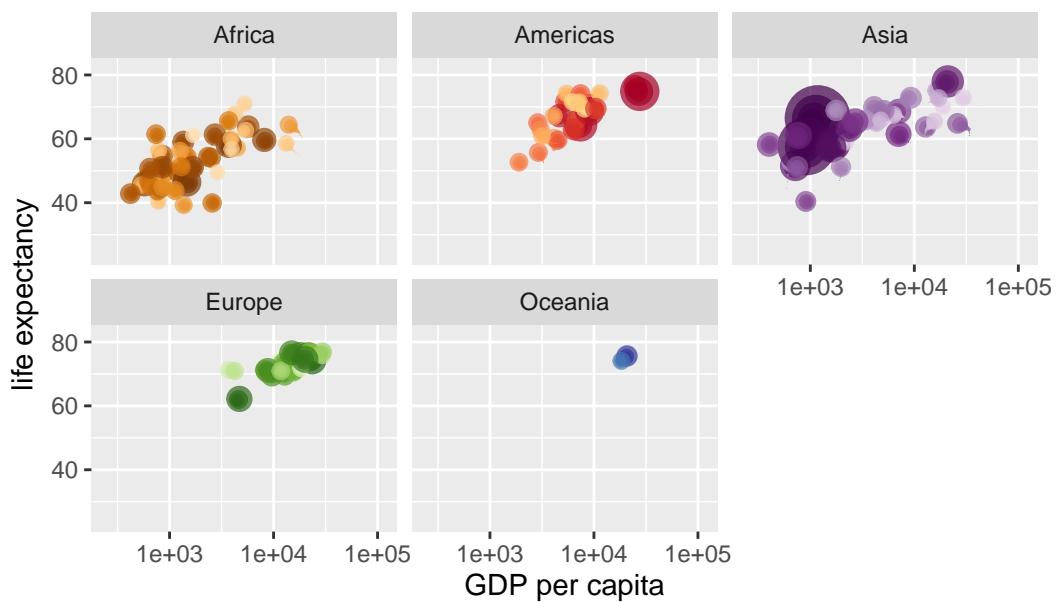
Year: 1984



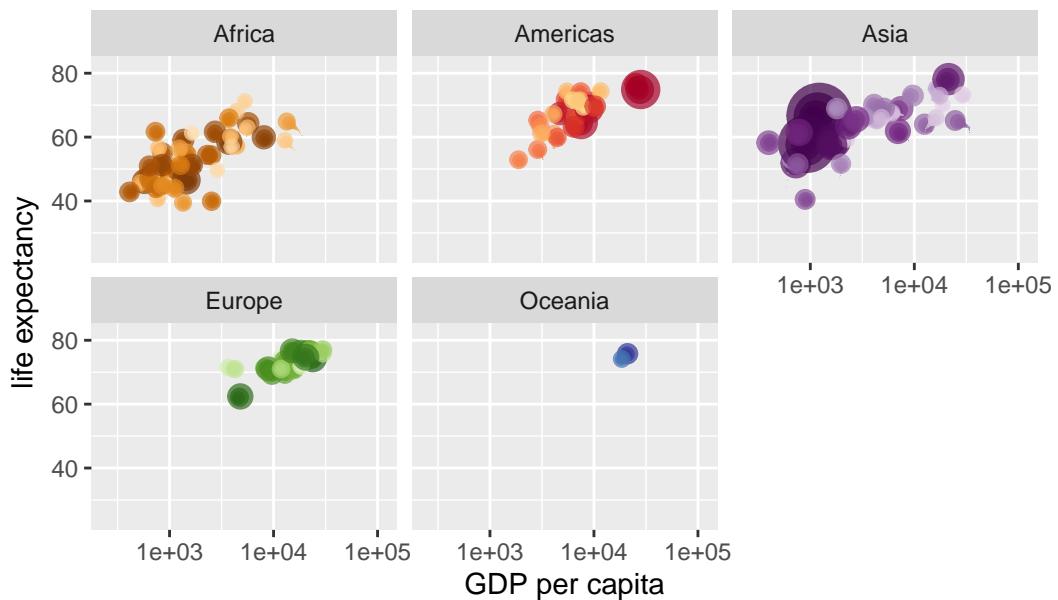
Year: 1984



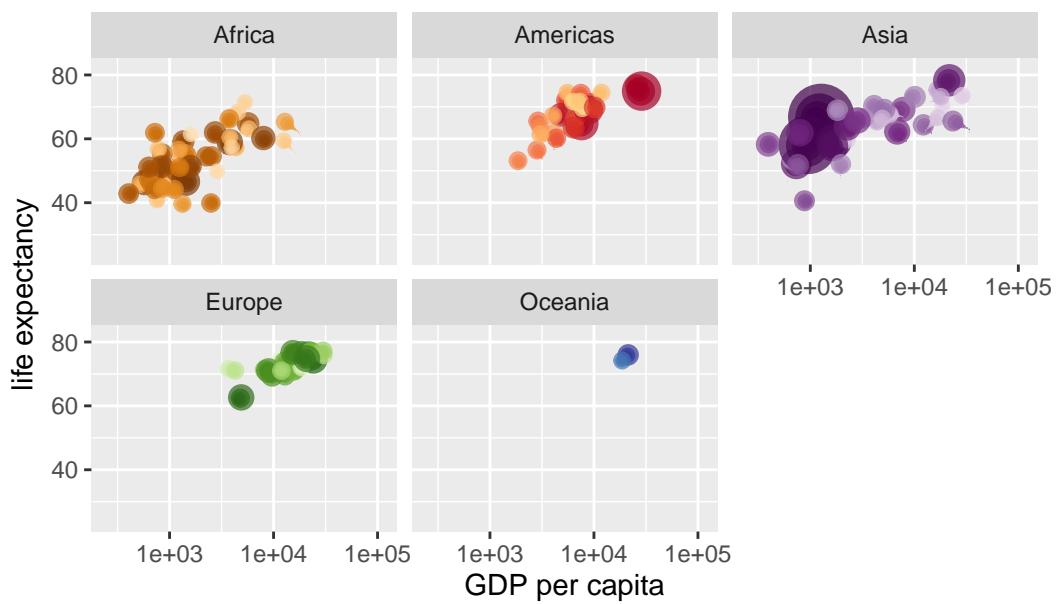
Year: 1985



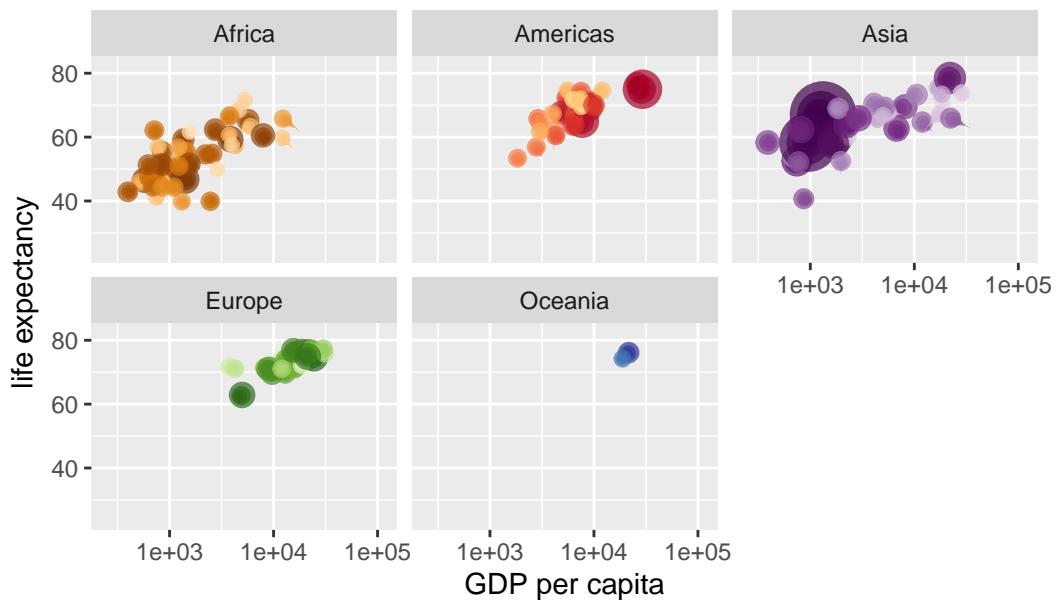
Year: 1985



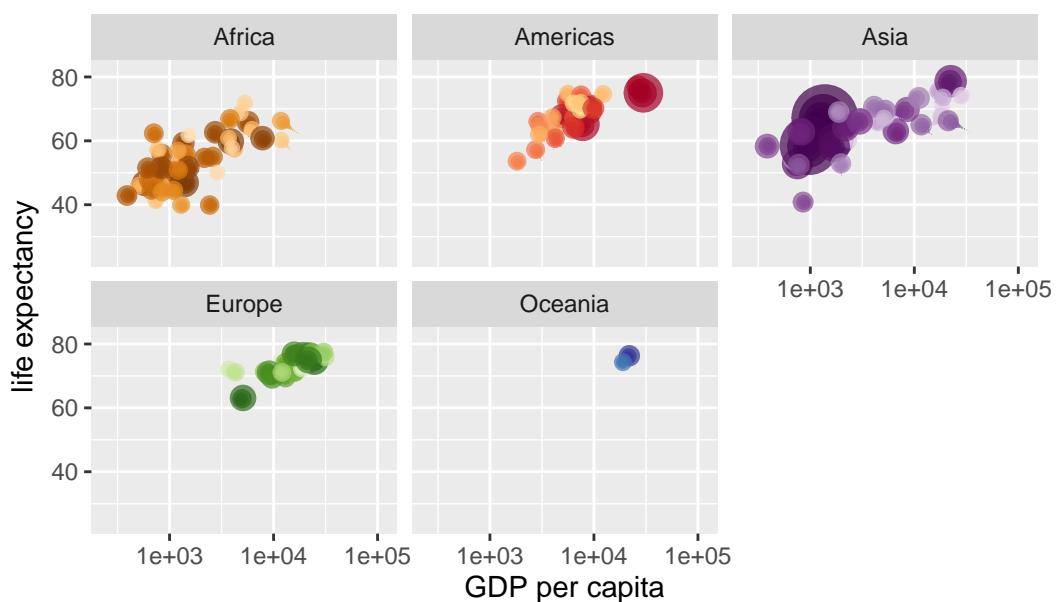
Year: 1986



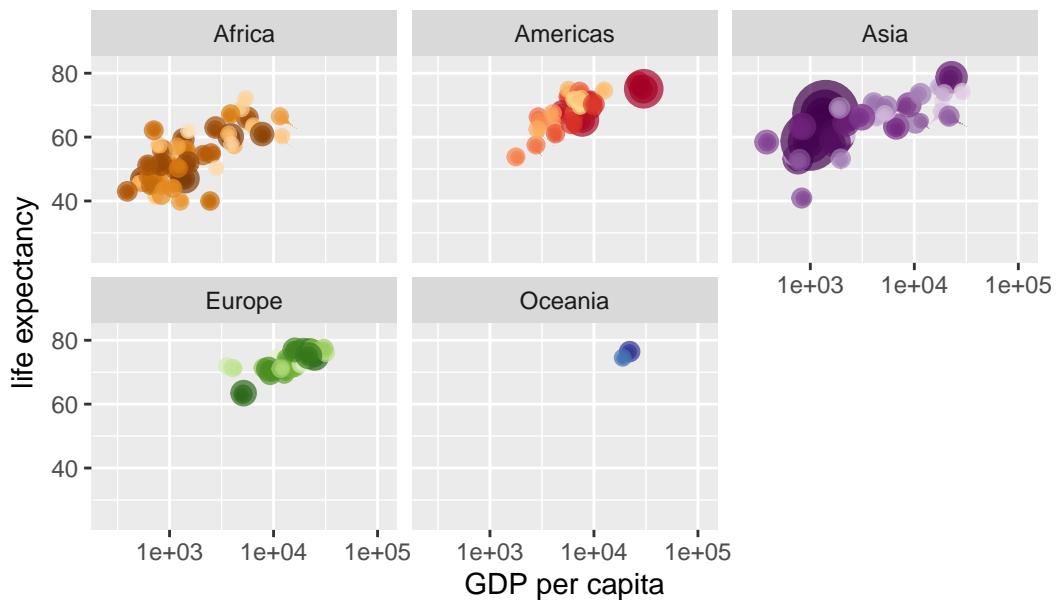
Year: 1986



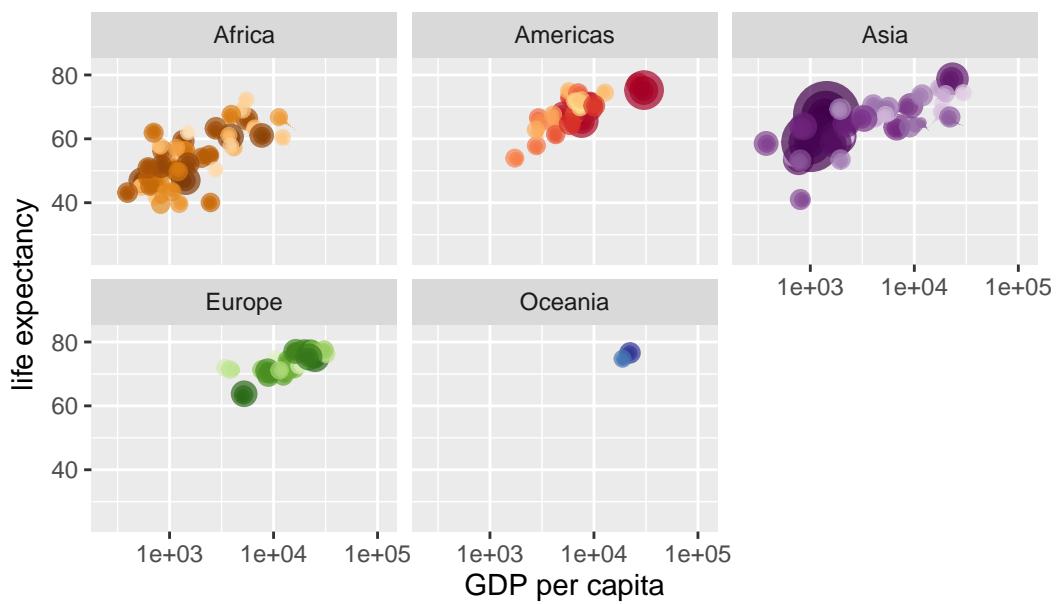
Year: 1987



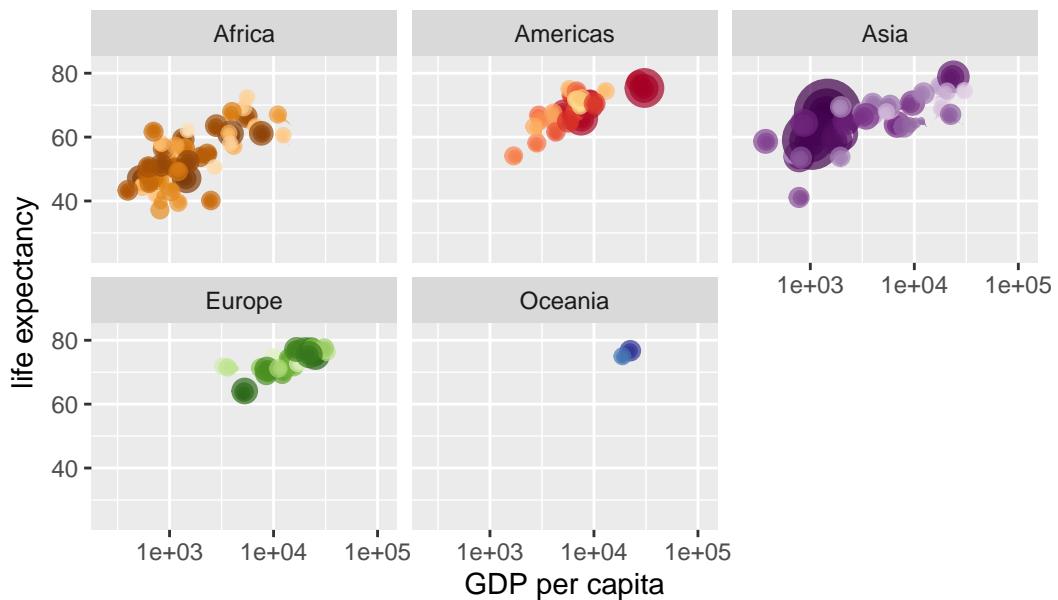
Year: 1988



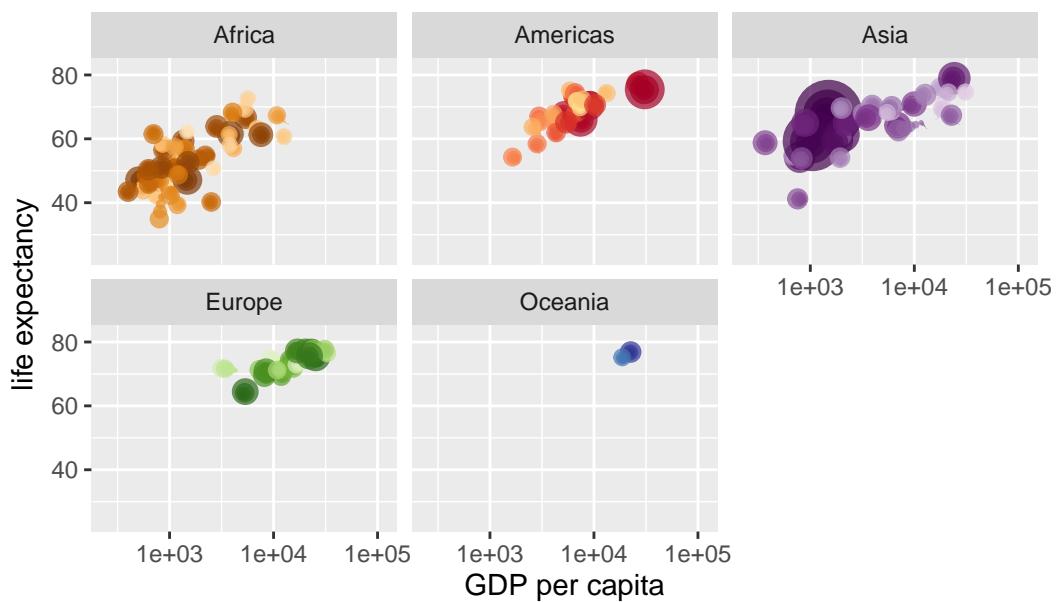
Year: 1988



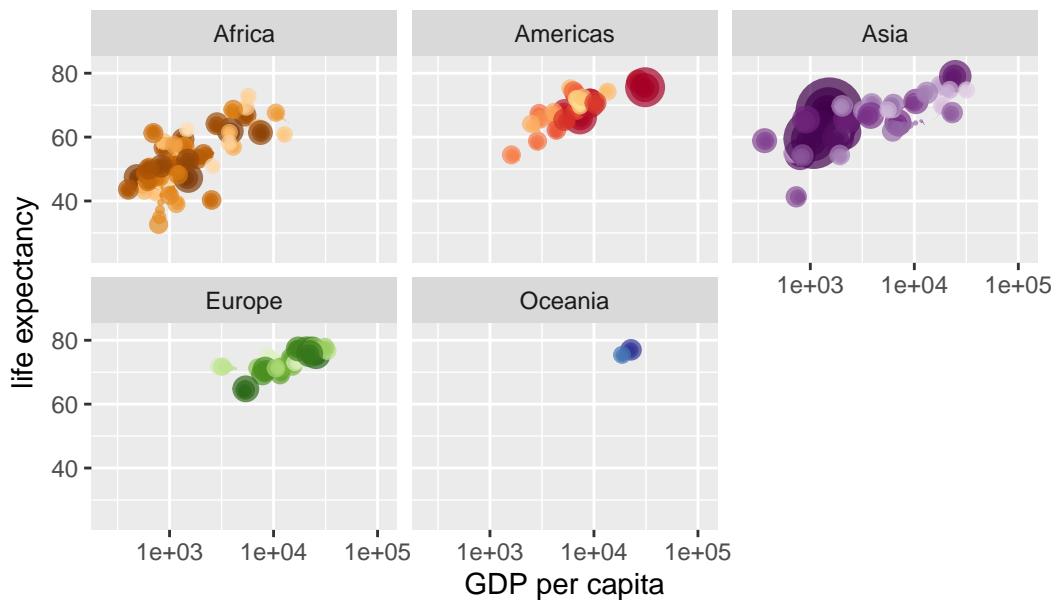
Year: 1989



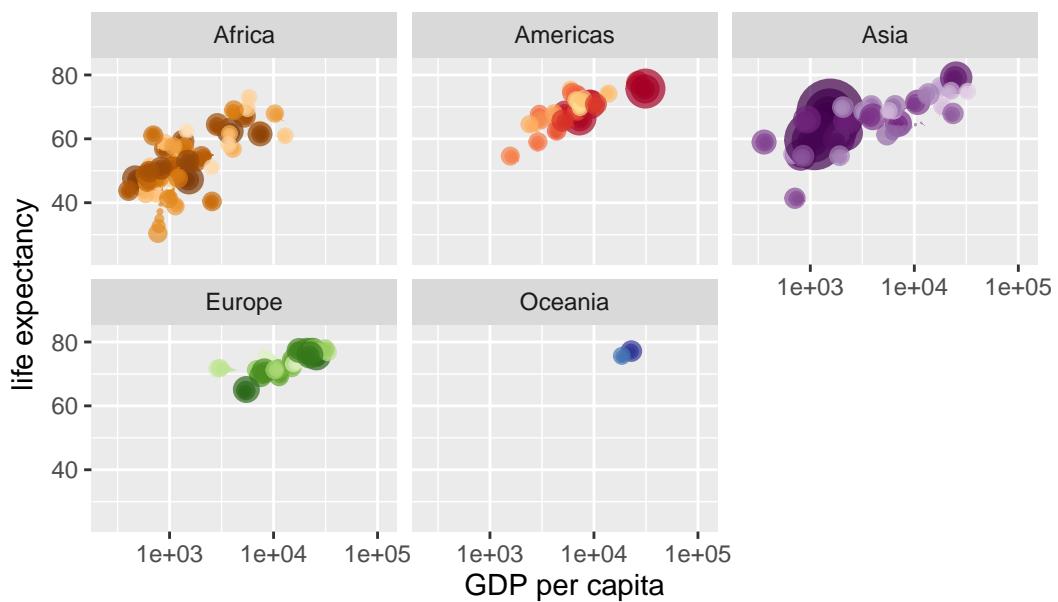
Year: 1989



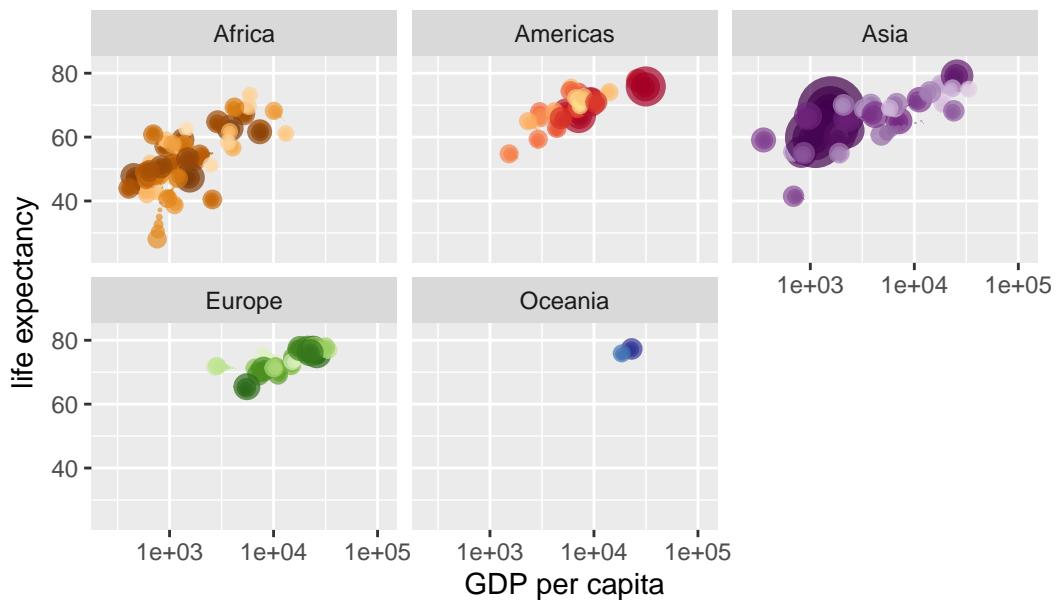
Year: 1990



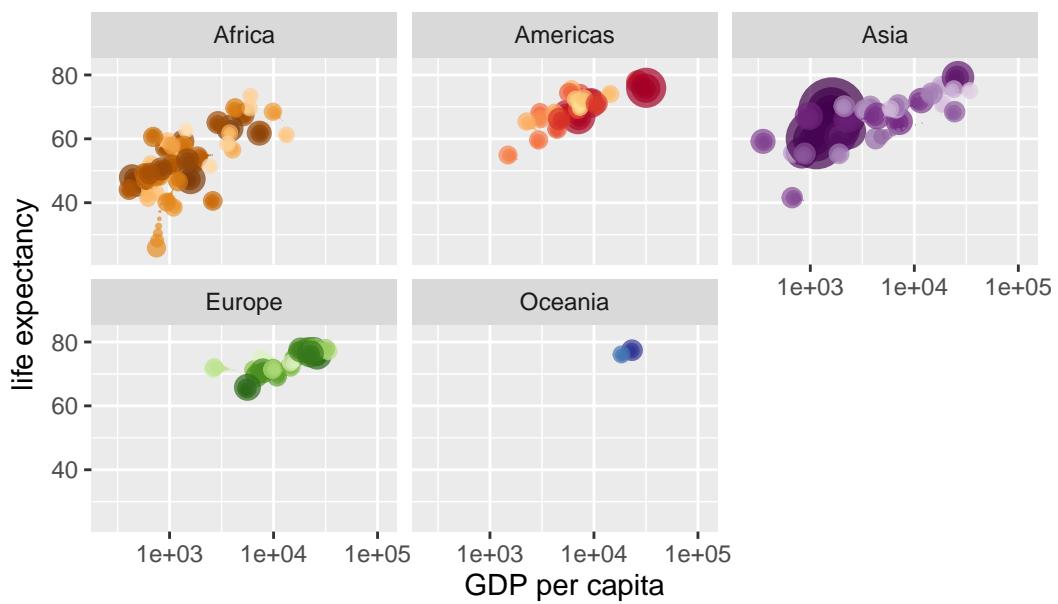
Year: 1990



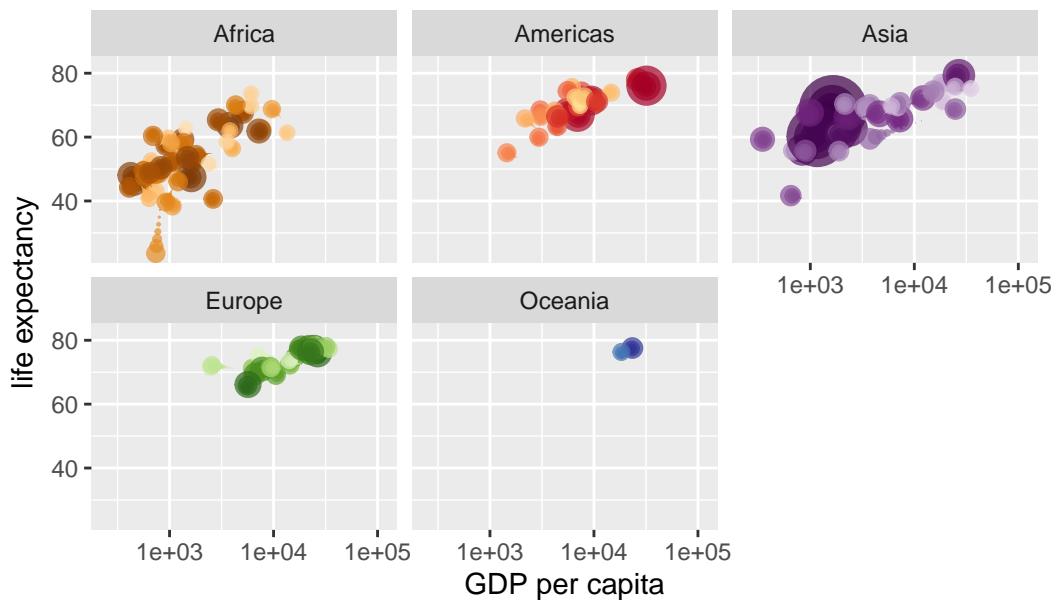
Year: 1991



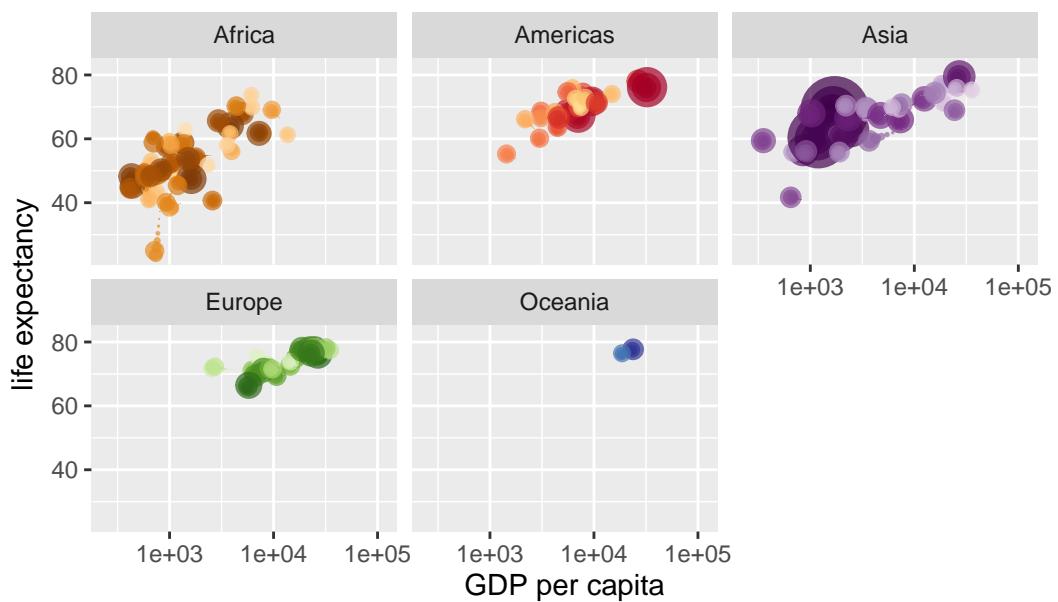
Year: 1991



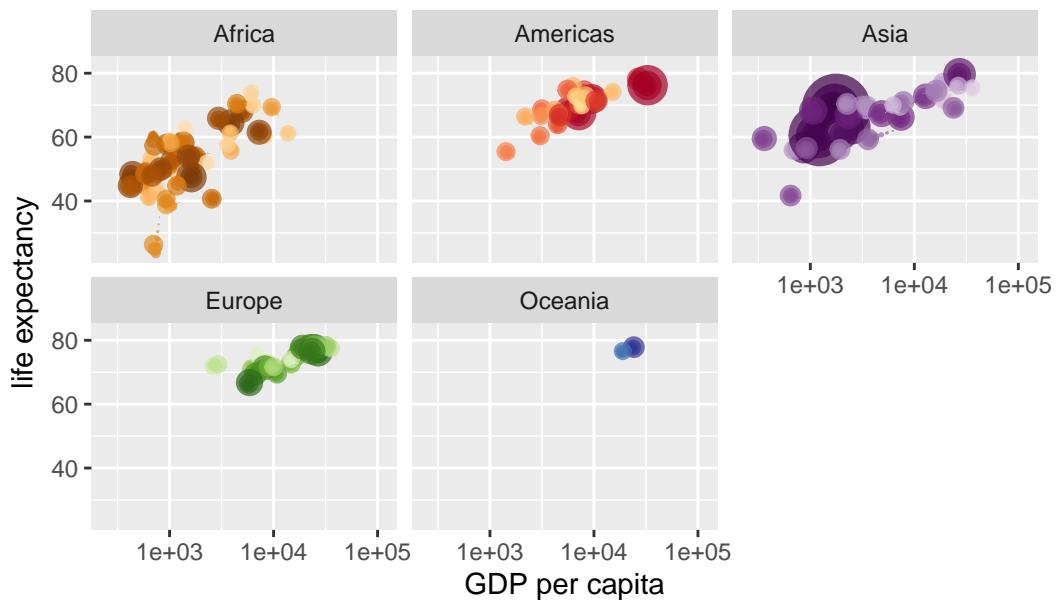
Year: 1992



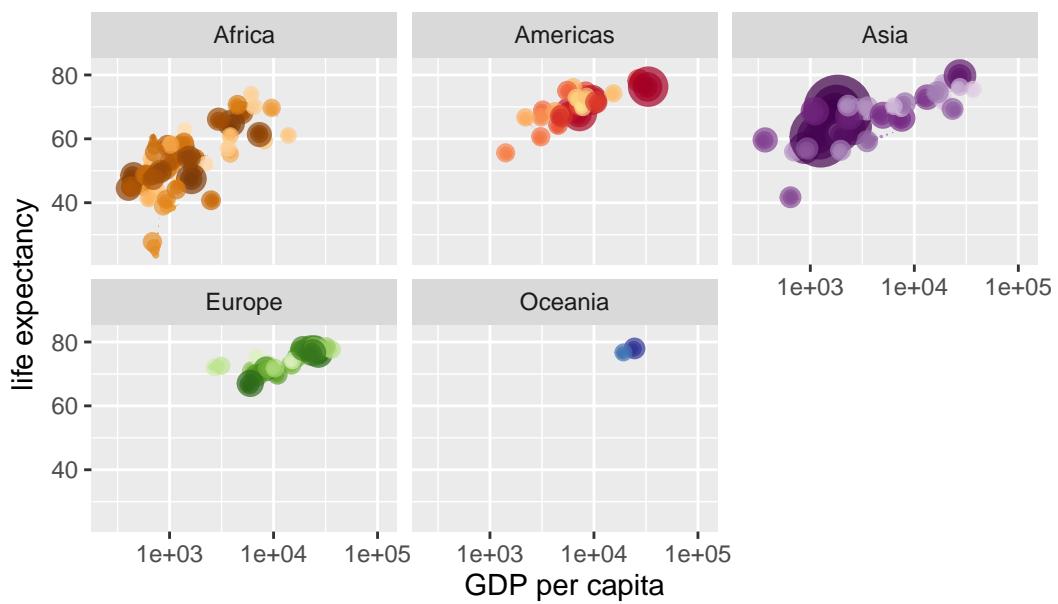
Year: 1993



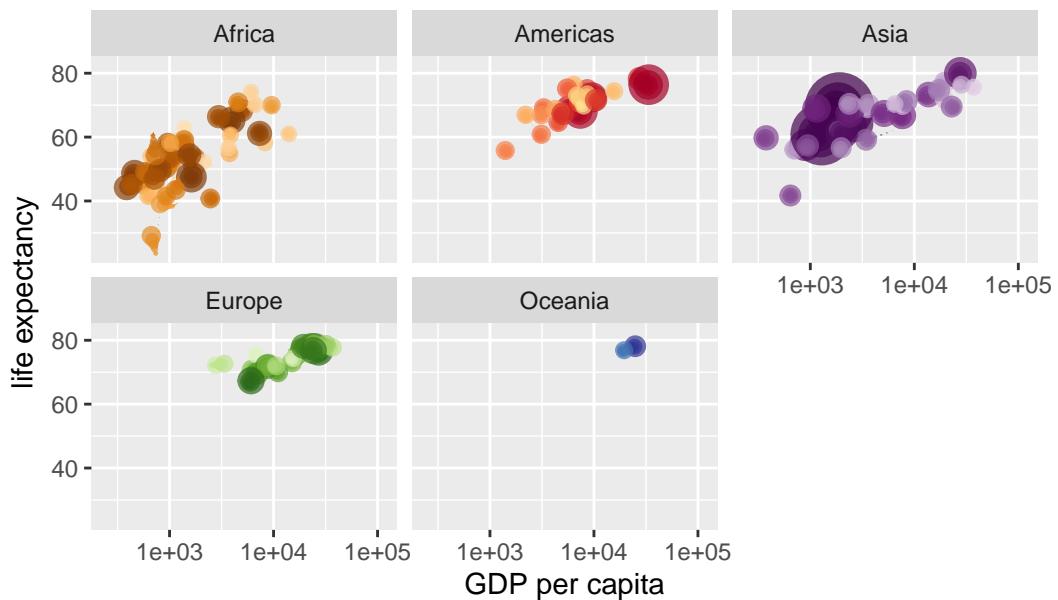
Year: 1993



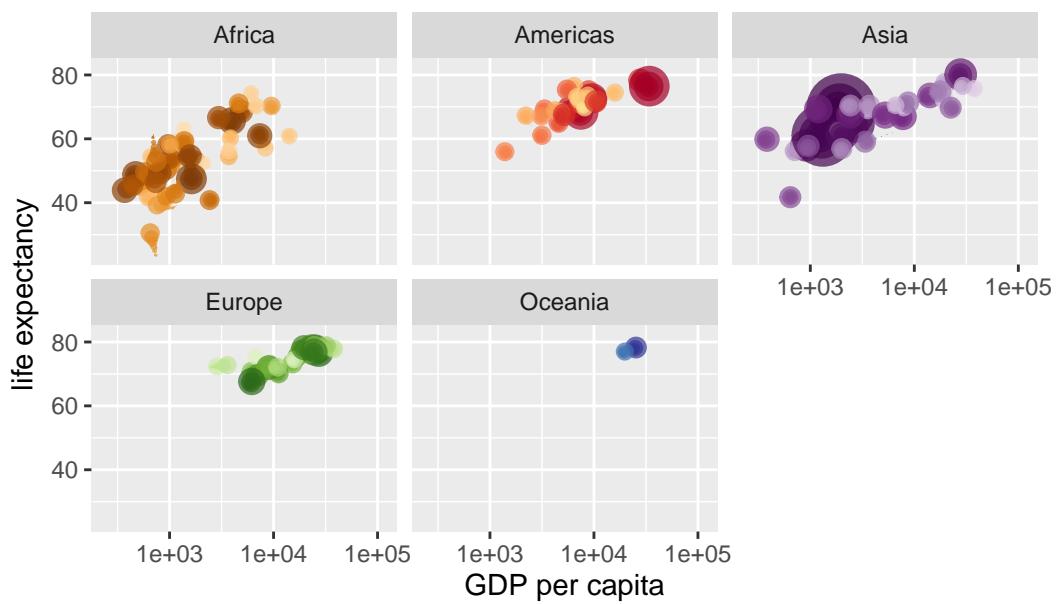
Year: 1994



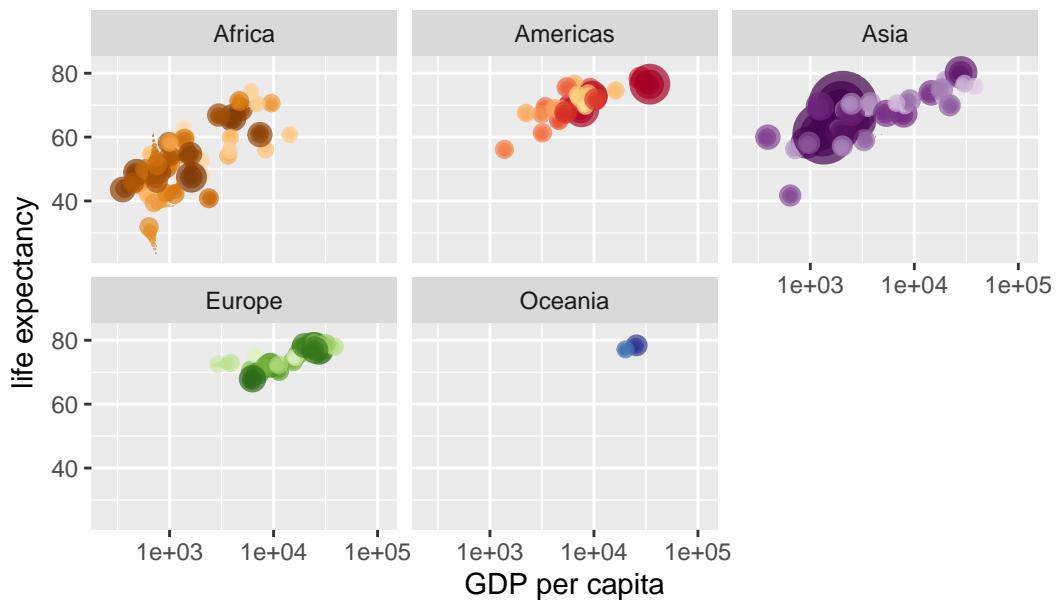
Year: 1994



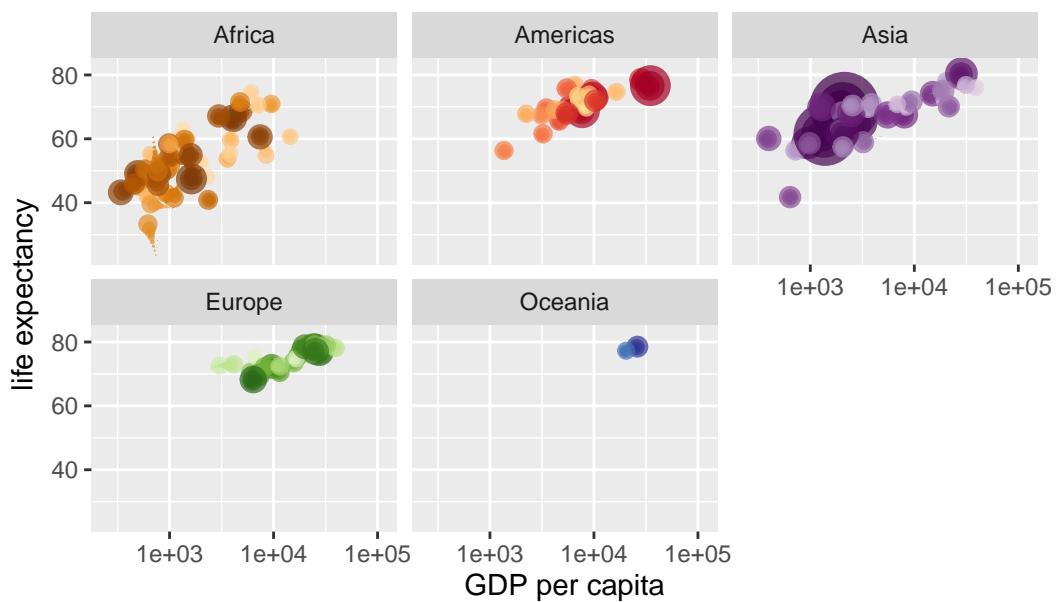
Year: 1995



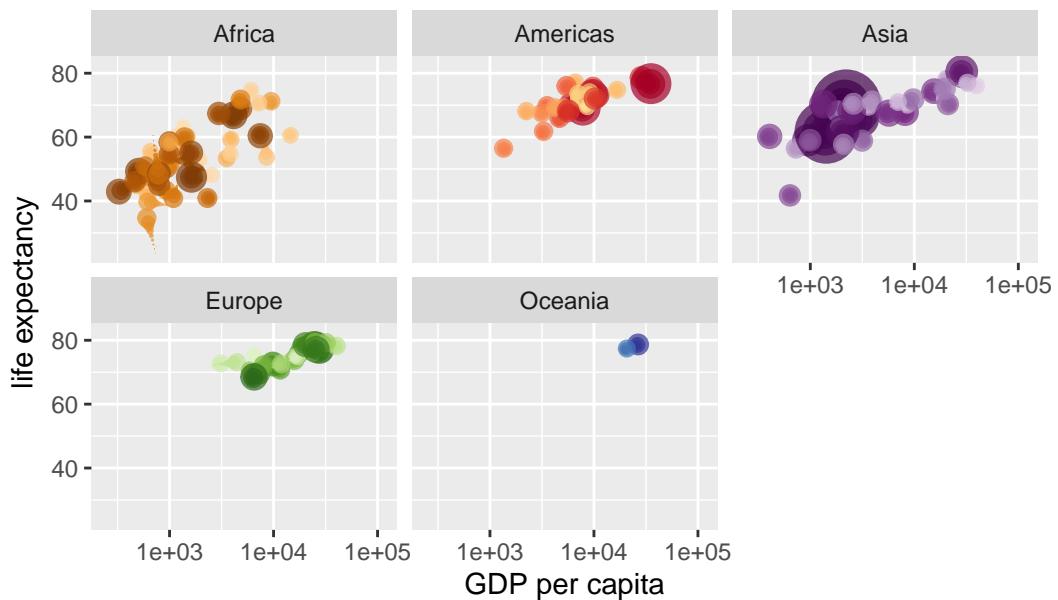
Year: 1995



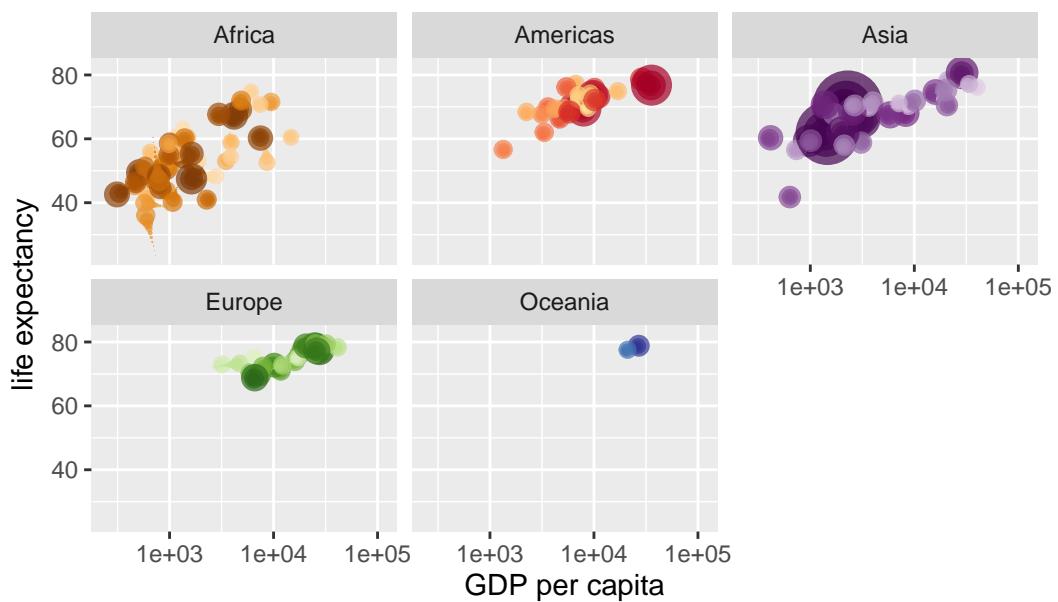
Year: 1996



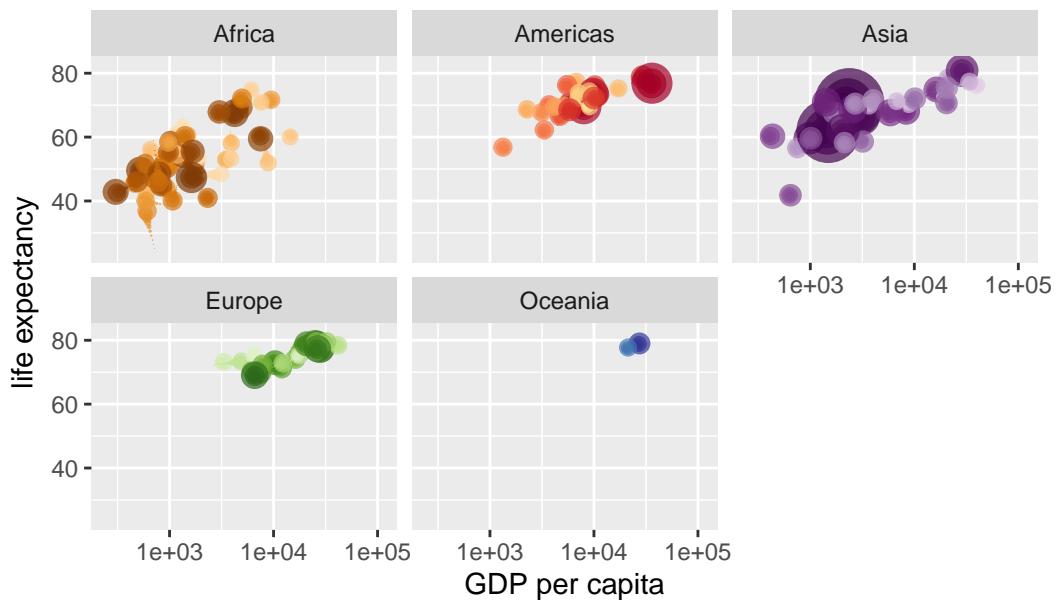
Year: 1996



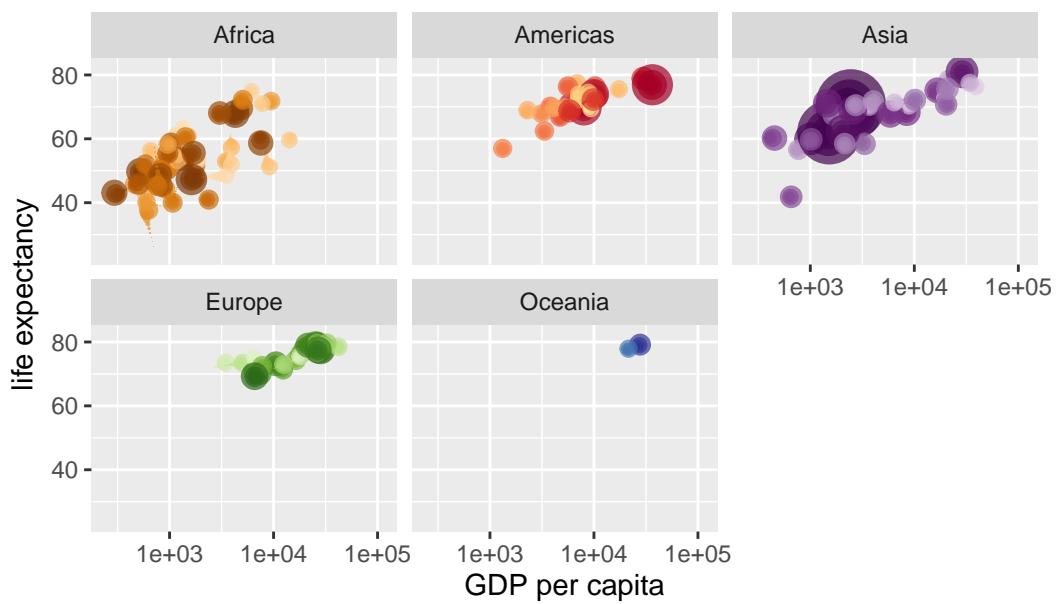
Year: 1997



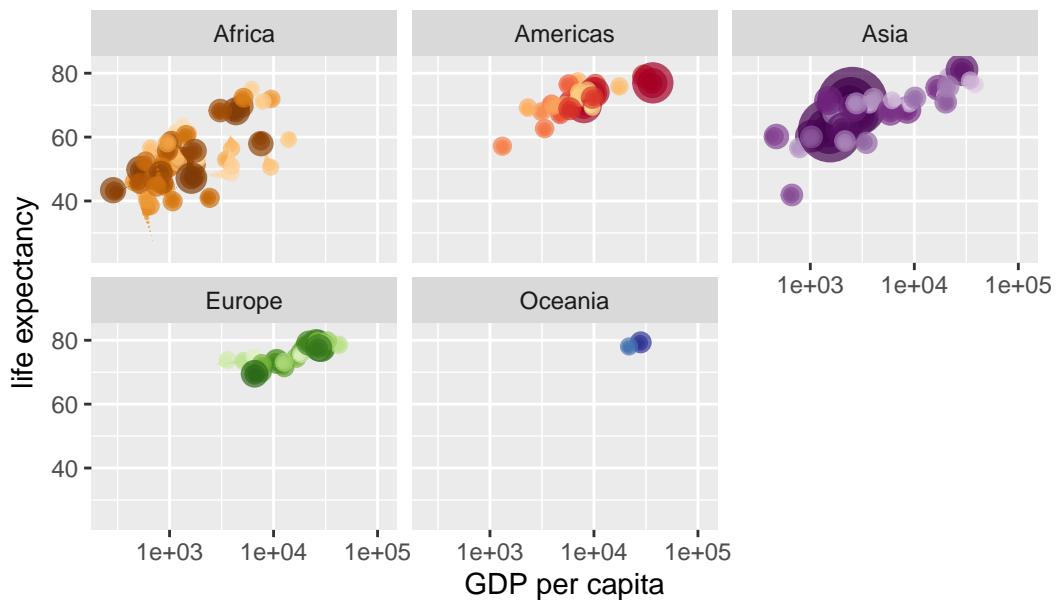
Year: 1998



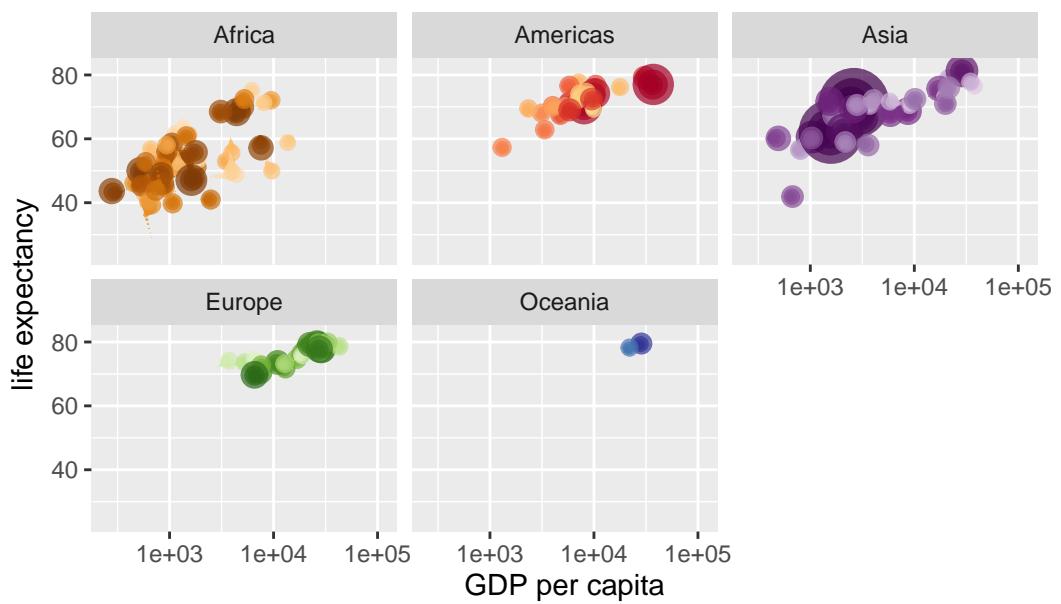
Year: 1998



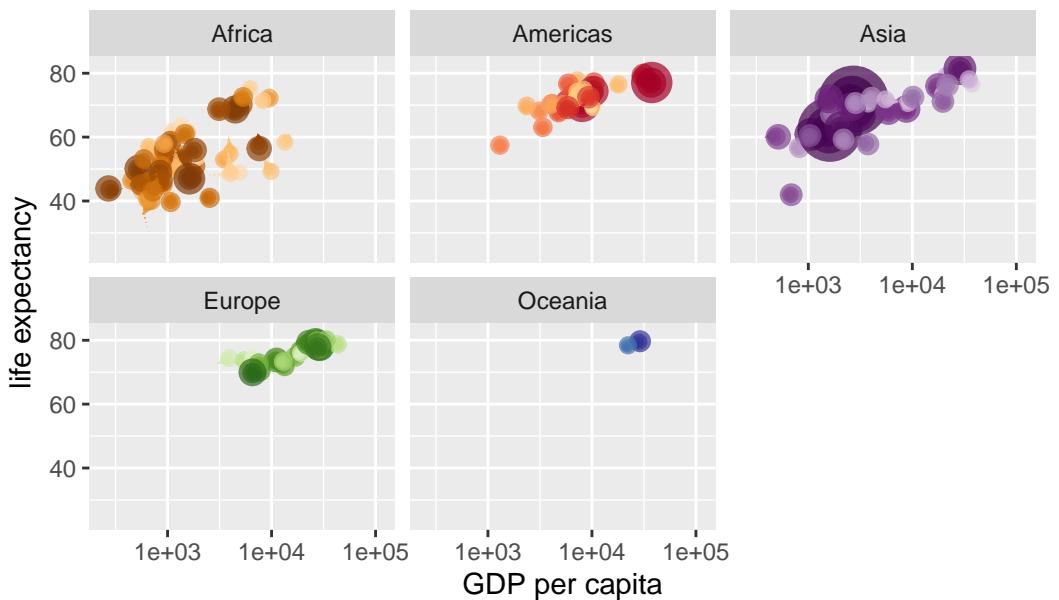
Year: 1999



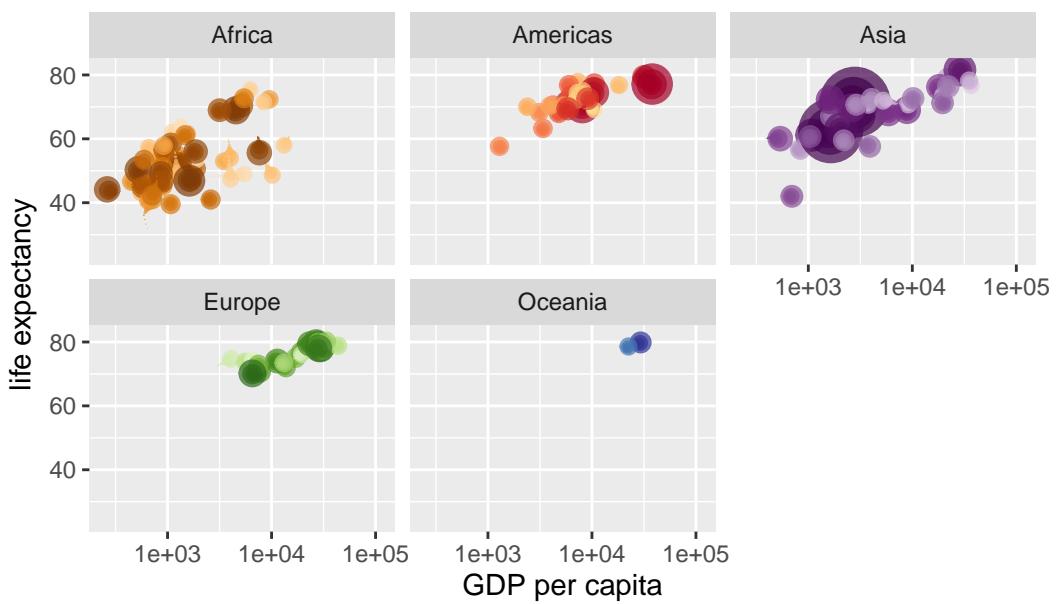
Year: 1999



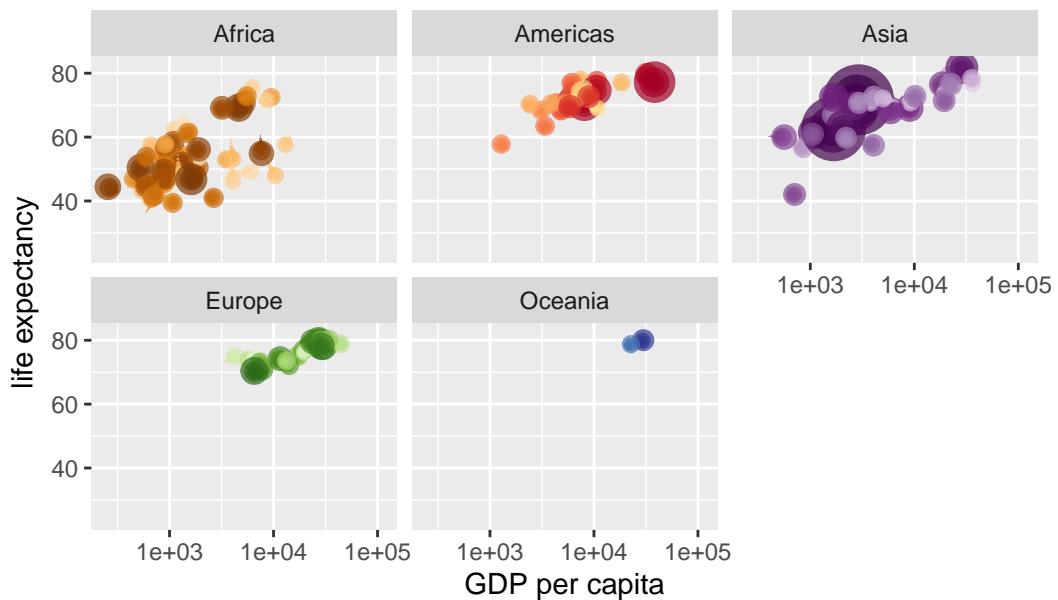
Year: 2000



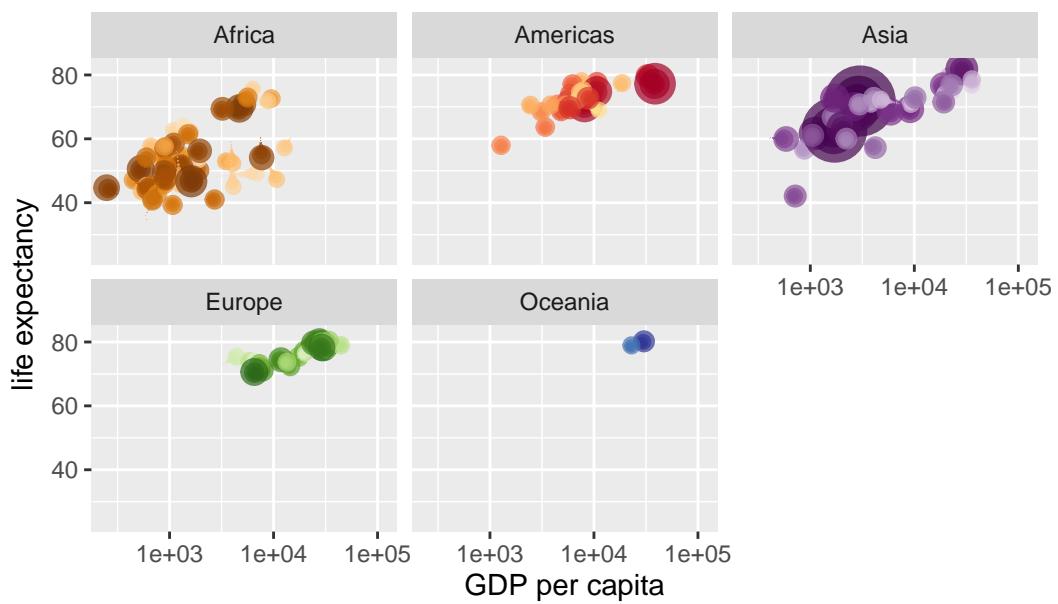
Year: 2000



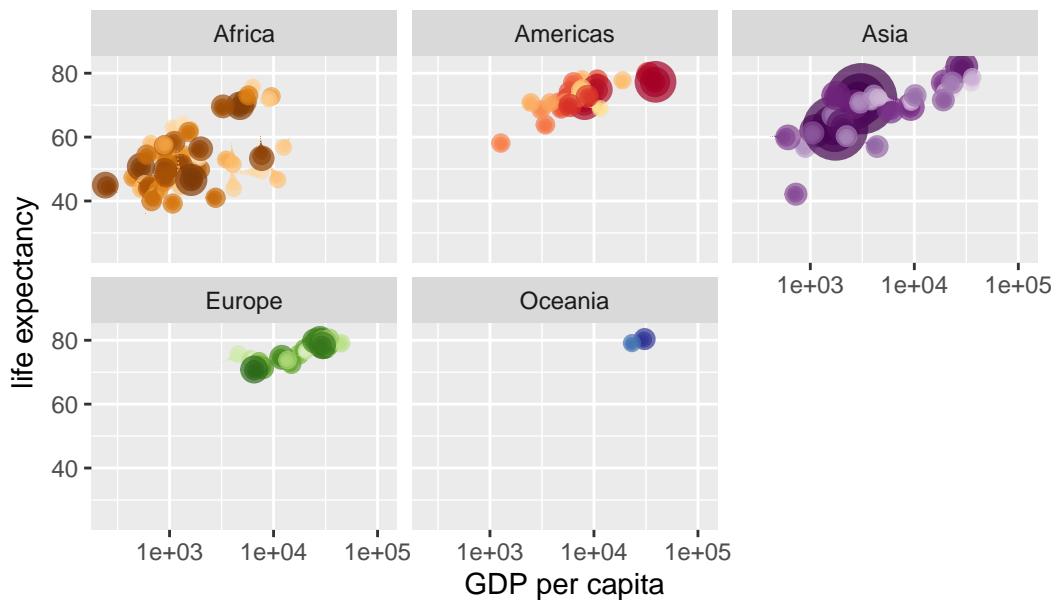
Year: 2001



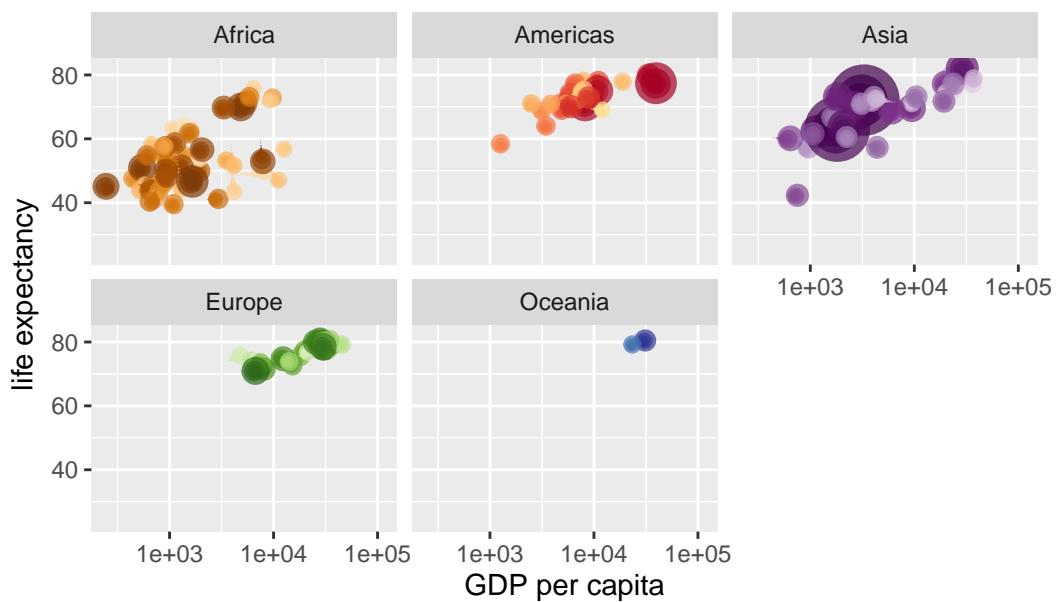
Year: 2001



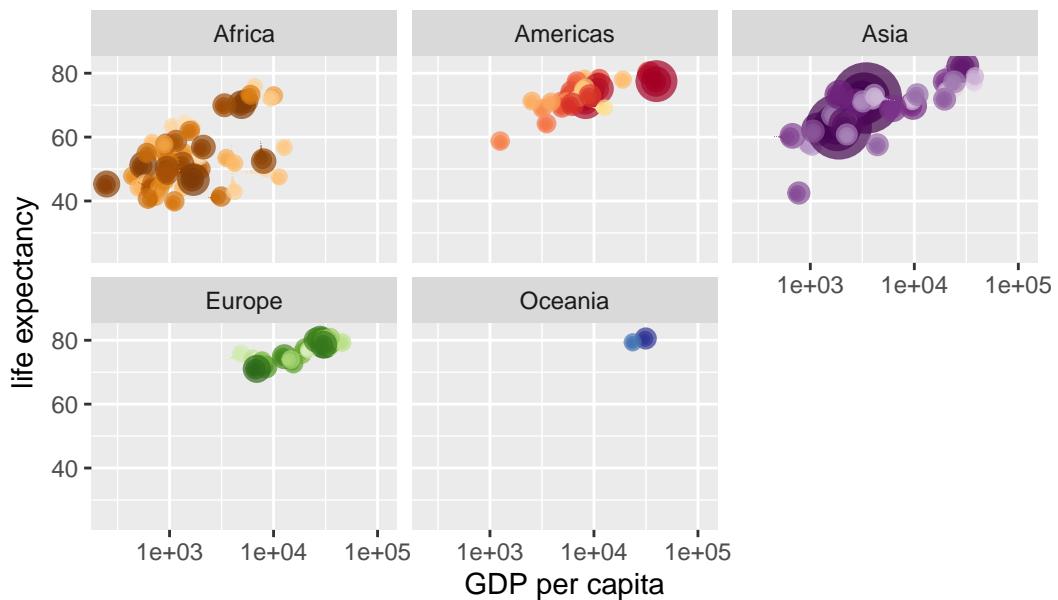
Year: 2002



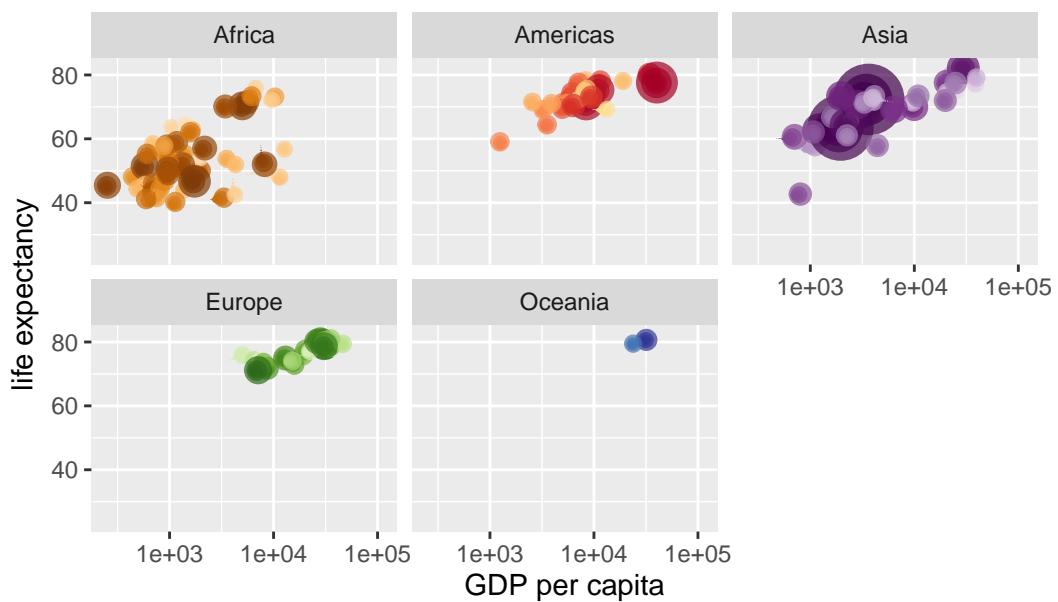
Year: 2003



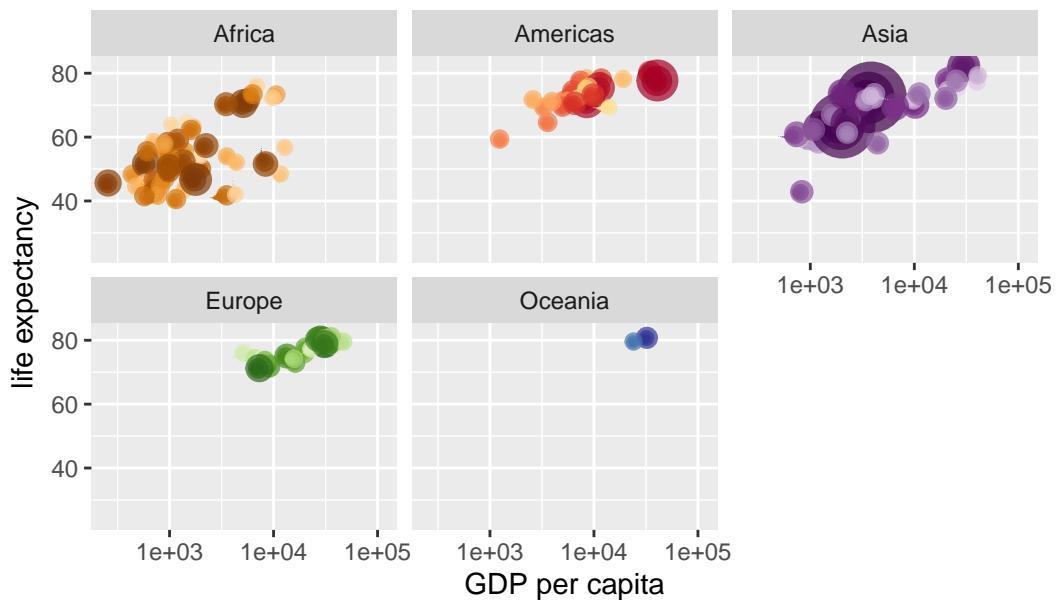
Year: 2003



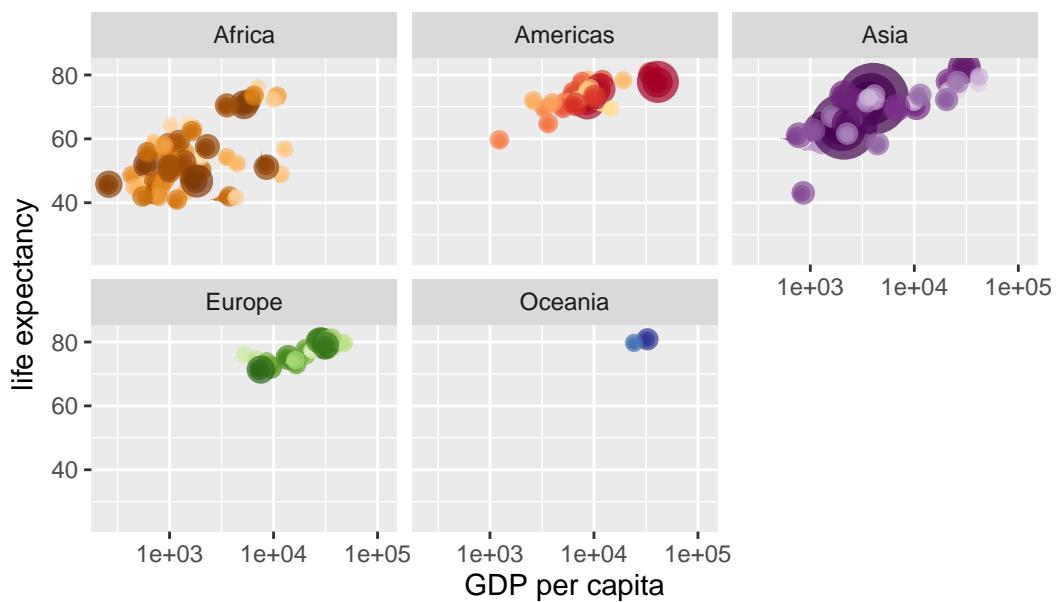
Year: 2004



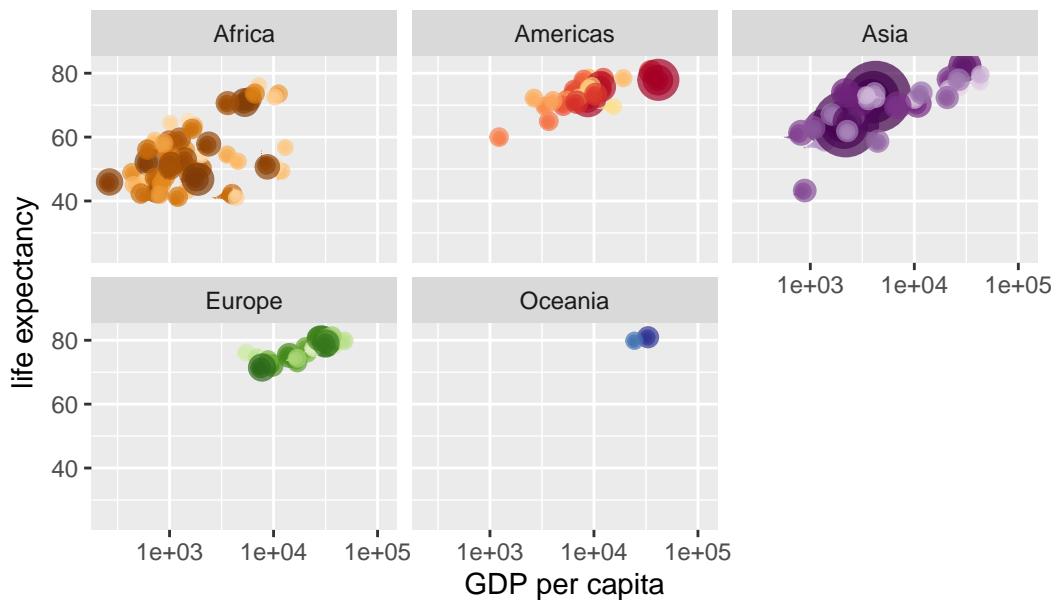
Year: 2004



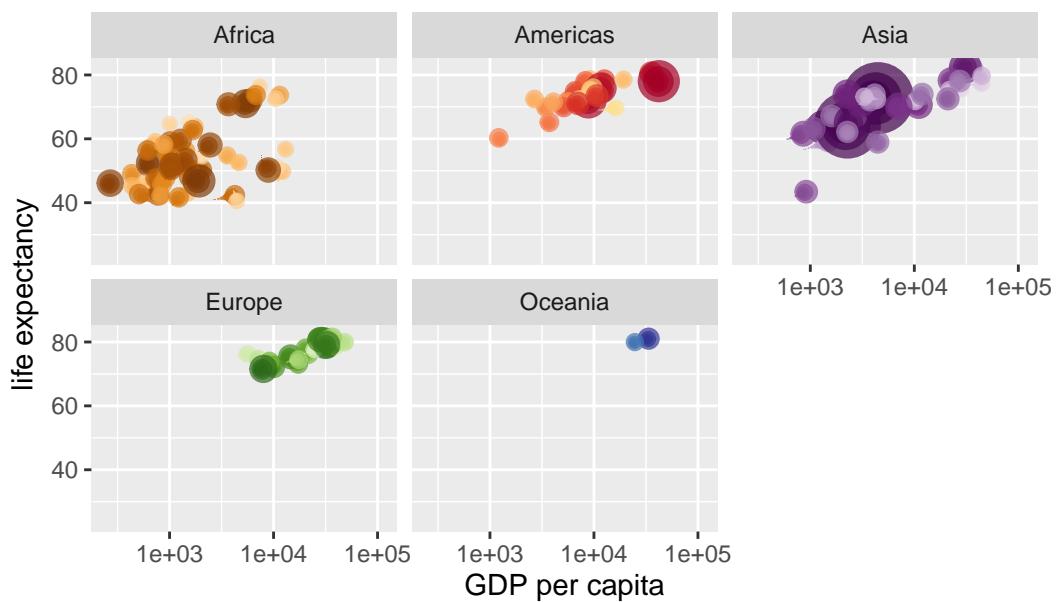
Year: 2005



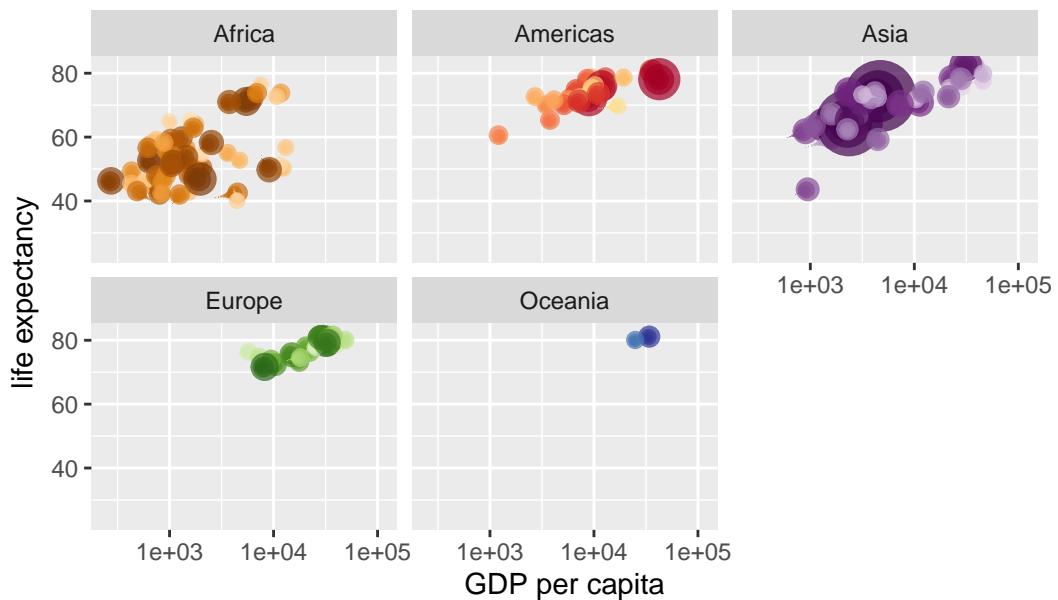
Year: 2005



Year: 2006



Year: 2006



Year: 2007

