

Class 05: Data Visualization with GGPLOT

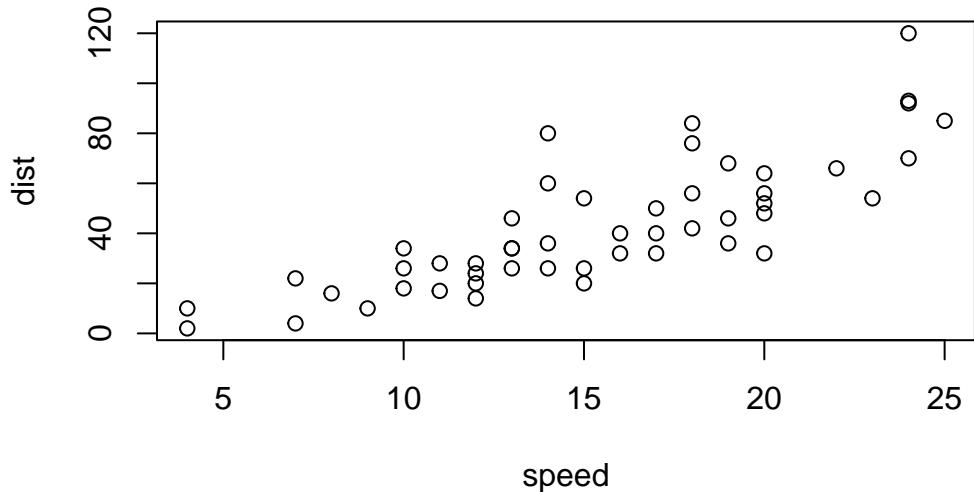
Sofia Lanaspa, PID:A17105313

Quarto

Quarto enables you to weave together content and executable code into a finished document.
To learn more about Quarto see <https://quarto.org>.

Cars Data Set

```
plot(cars)
```



```
head(cars)
```

```
speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

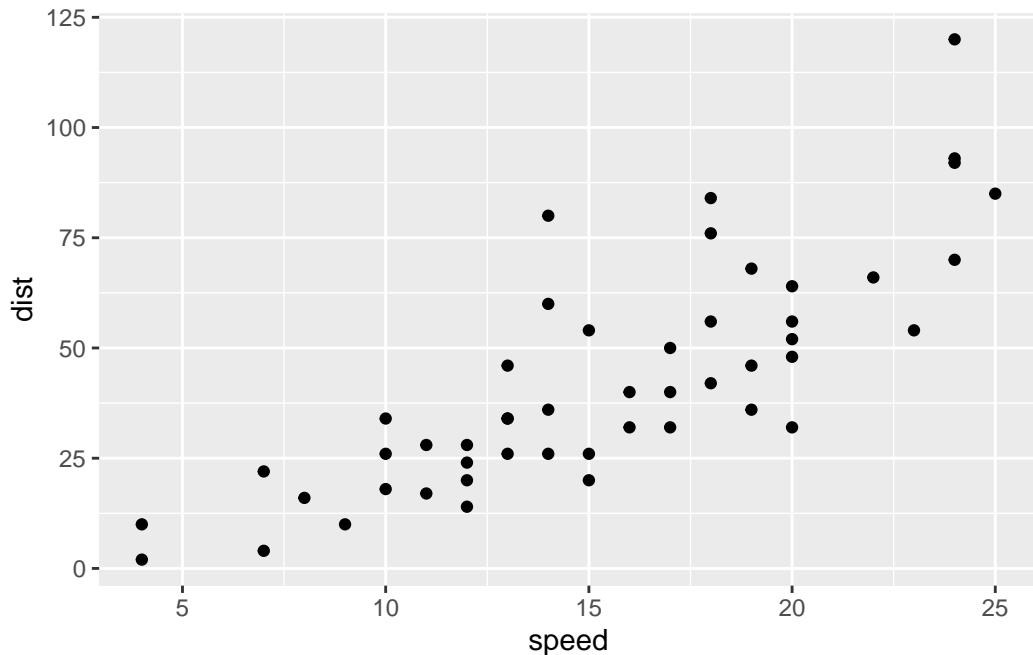
Q. How do we plot this wee data set with ggplot? All ggplot figures have at least 3 layers:

- **data**
- **aes** (how the data map to the plot)
- **geom** (how the data draw the plot, lines, points, etc)

Before using new package need to download and install it with the ‘install.packages()’ command! *We don’t want to install in quattro document because it would re-install every time

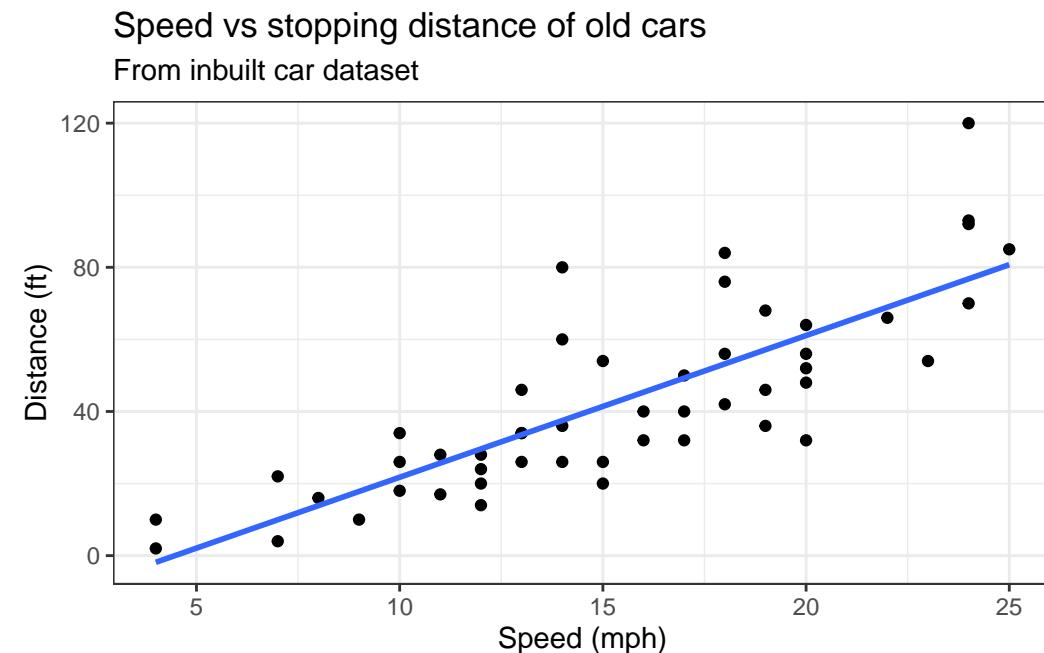
Once installed, load it with ‘library()’ function

```
# install.packages("ggplot2")
library(ggplot2)
ggplot(cars) + aes(x=speed, y=dist) + geom_point()
```



Key-point: for simple plots like the one above ggplot is more verbose (more typing), but for more difficult plots just add more ‘+’ which is easier than using base R plot()

```
myFirstPlot <- ggplot(cars) +  
  aes(speed, dist) +  
  geom_point() +  
  geom_smooth(method = "lm", se=FALSE) +  
  labs(x="Speed (mph)", y="Distance (ft)", title="Speed vs stopping distance of old cars",  
    theme_bw())  
myFirstPlot  
  
`geom_smooth()` using formula = 'y ~ x'
```



*when you need to edit a function, use AI (easier than reading the help tab)

Gene Data Set

```
url <- "https://bioboot.github.io/bimml43_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
```

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

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

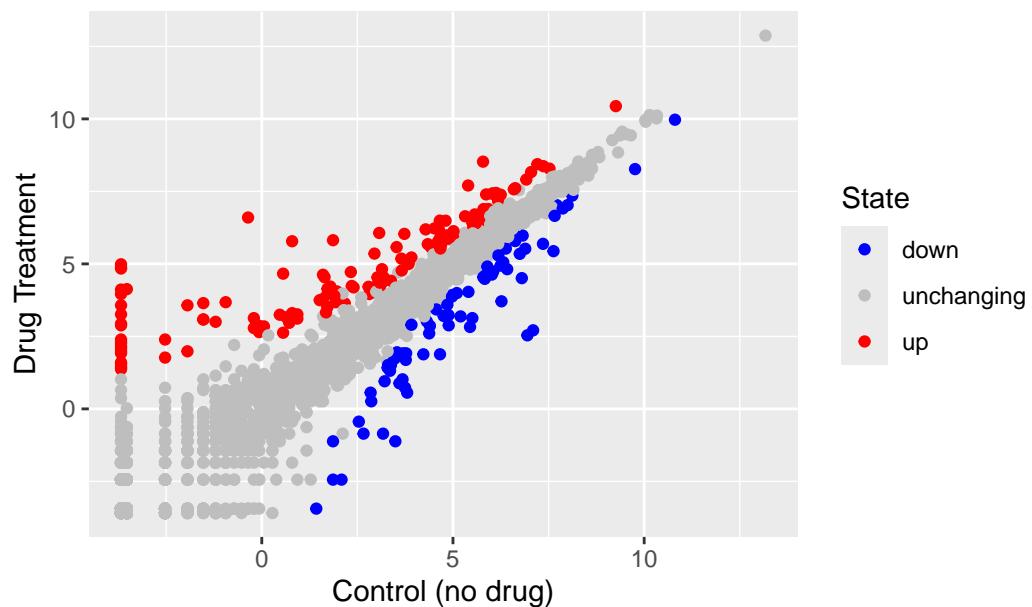
	down	unchanging	up
	72	4997	127

```
round(table(genes$State)/nrow(genes),4)
```

	down	unchanging	up
	0.0139	0.9617	0.0244

```
mySecondPlot <- ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point() +
  labs(x="Control (no drug)", y="Drug Treatment", title= "Gene expression changes upon drug +")
mySecondPlot + scale_color_manual(values=c("blue","gray","red"))
```

Gene expression changes upon drug treatment



Gapminder Data Set

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

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

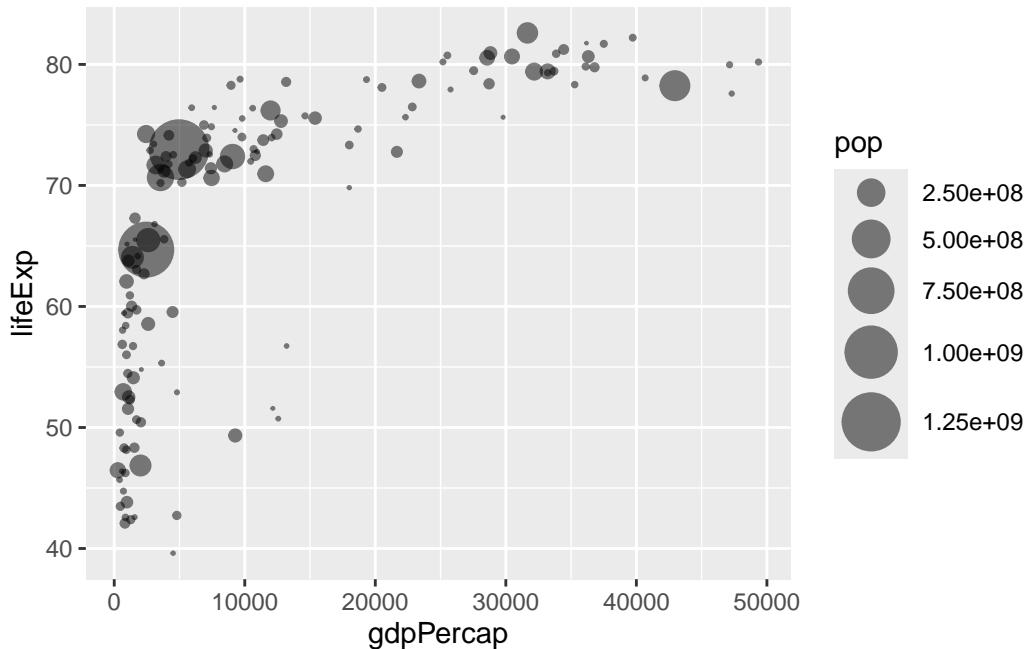
```
The following objects are masked from 'package:stats':
```

```
filter, lag
```

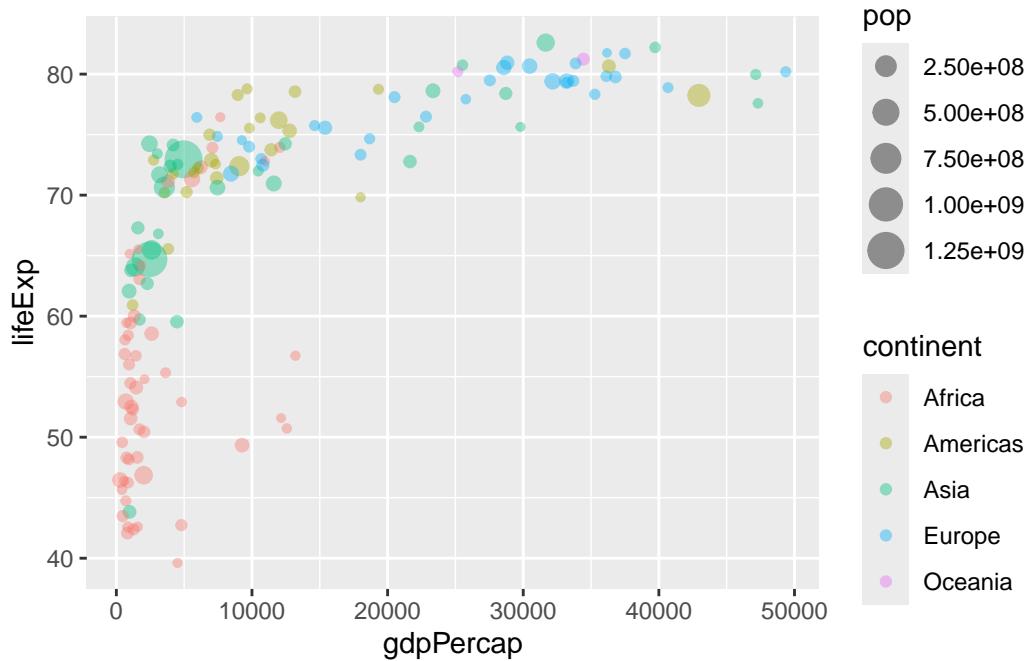
```
The following objects are masked from 'package:base':
```

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

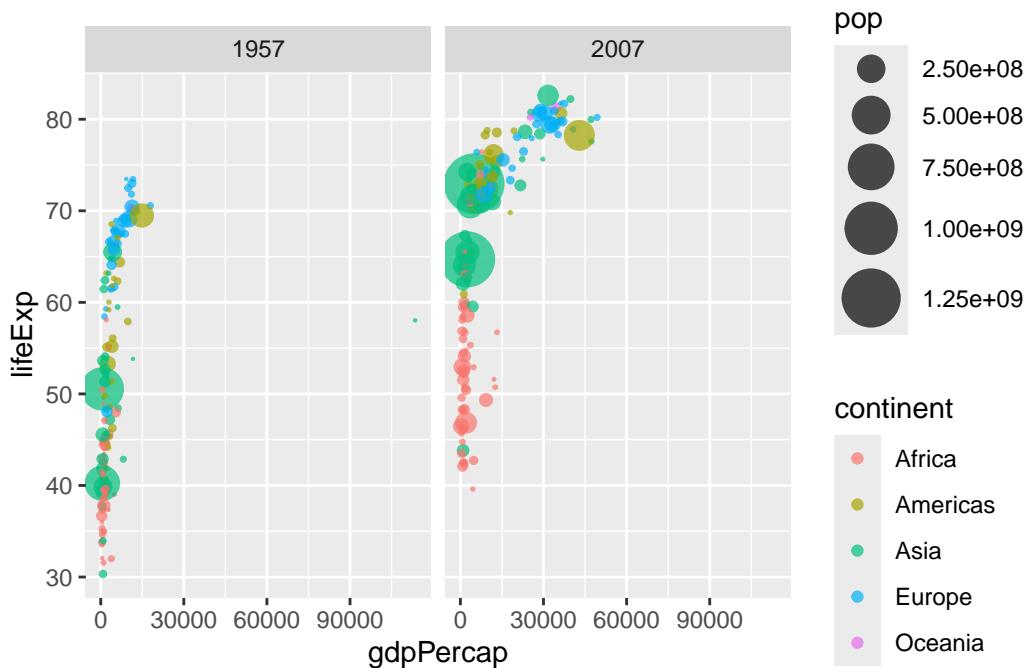
```
gapminder_2007 <- gapminder %>% filter(year==2007)
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp, size = pop), alpha=0.5) +
  scale_size_area(max_size = 10)
```



```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.4)
```



```
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)
ggplot(gapminder_1957) +
  geom_point(aes(x=gdpPercap, y=lifeExp, colour=continent, size=pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```

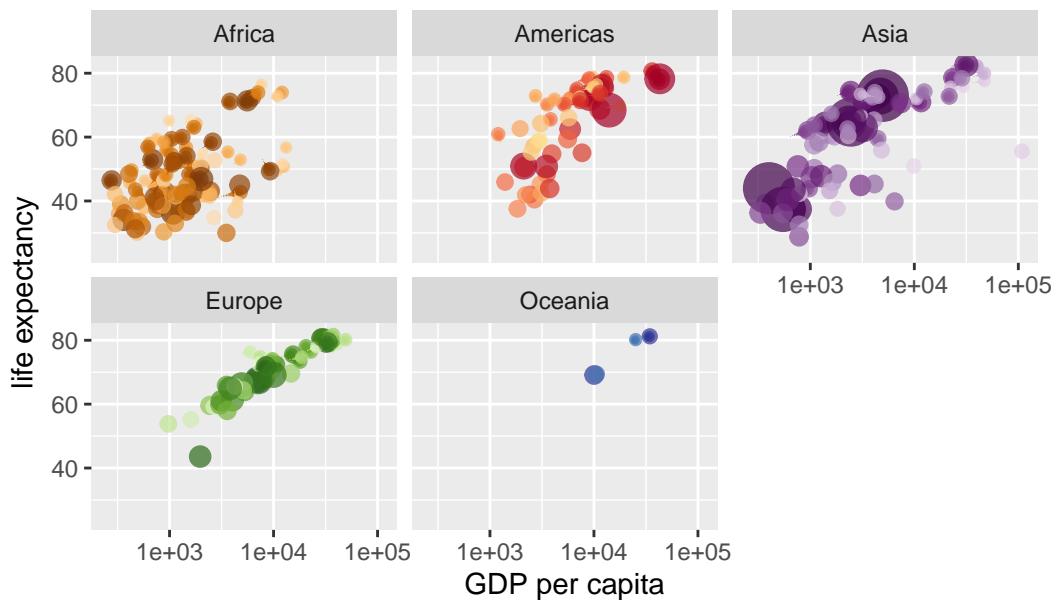


Animated Graphs

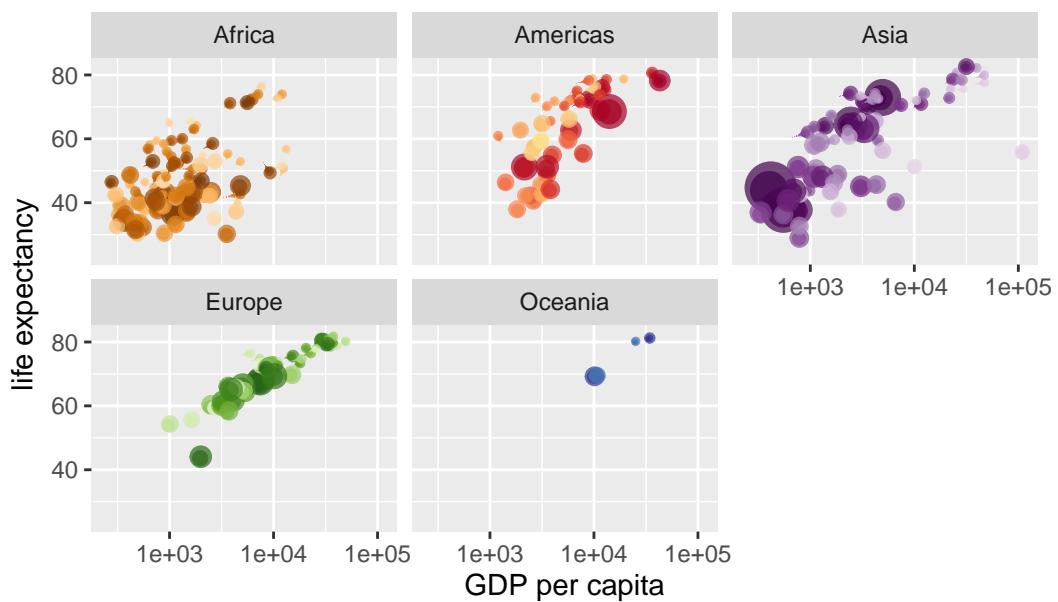
```
#install.packages("gifski")
#install.packages("gganimate")
library(gifski)
library(gganimate)
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  # Facet 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

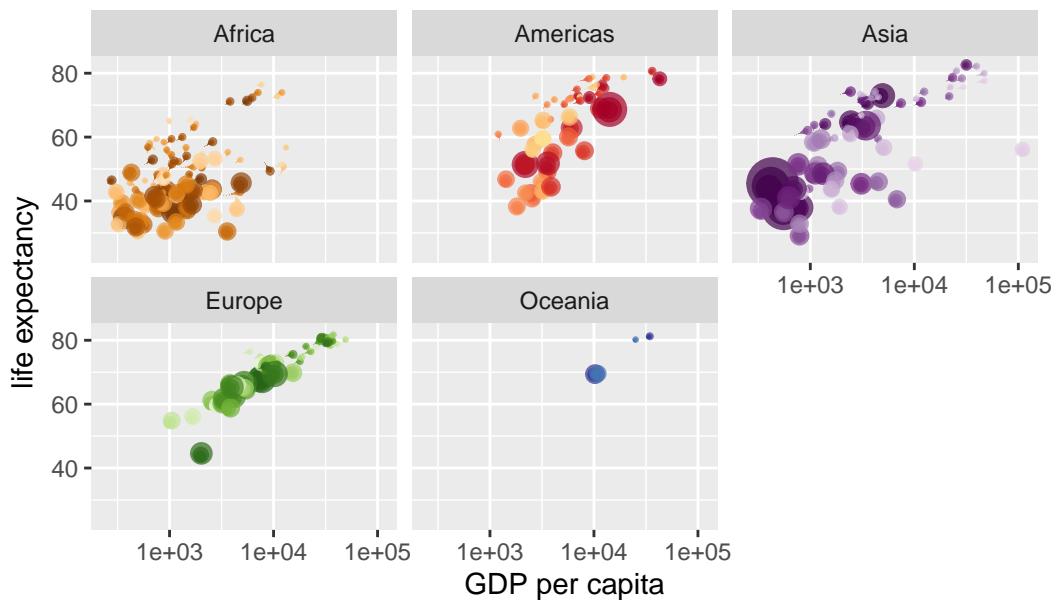
Year: 1952



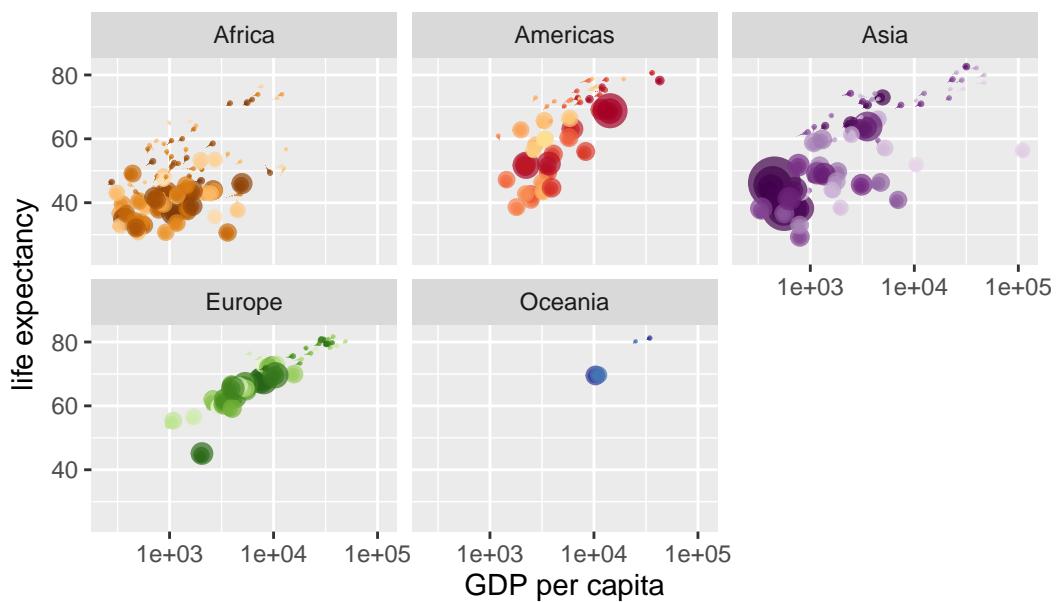
Year: 1953



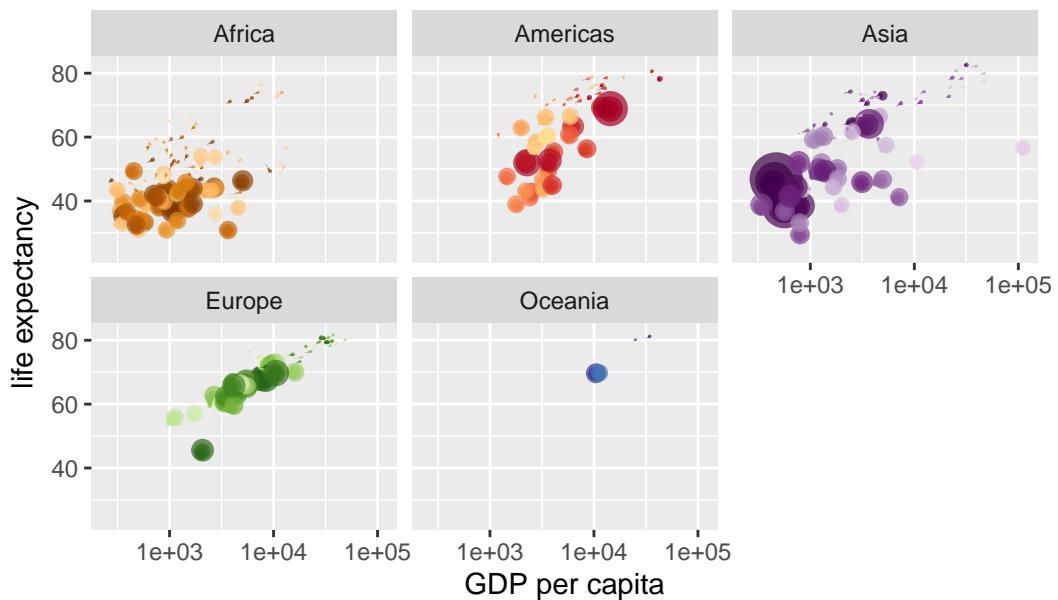
Year: 1953



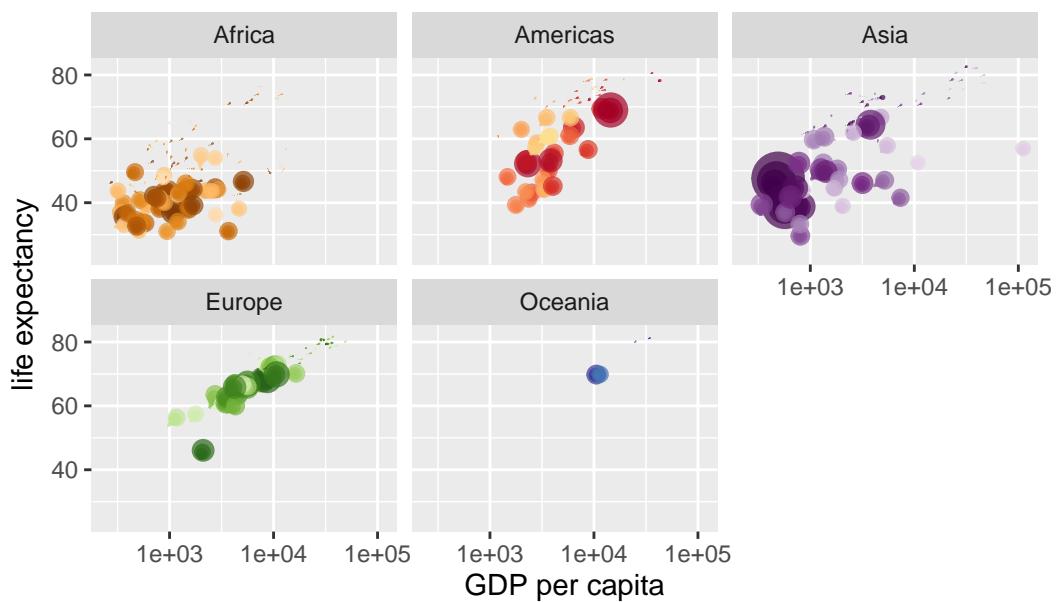
Year: 1954



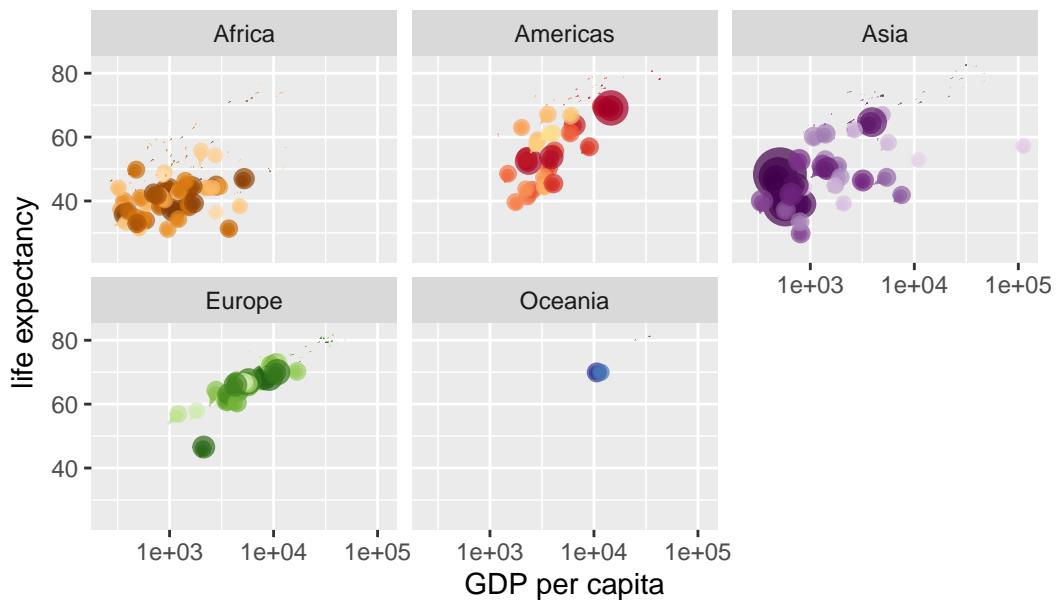
Year: 1954



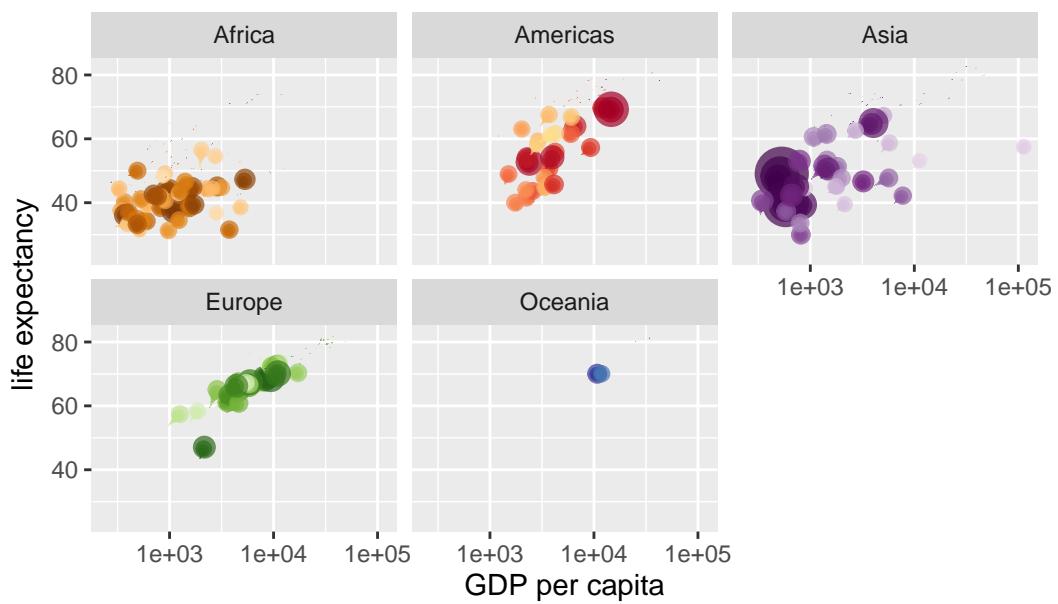
Year: 1955



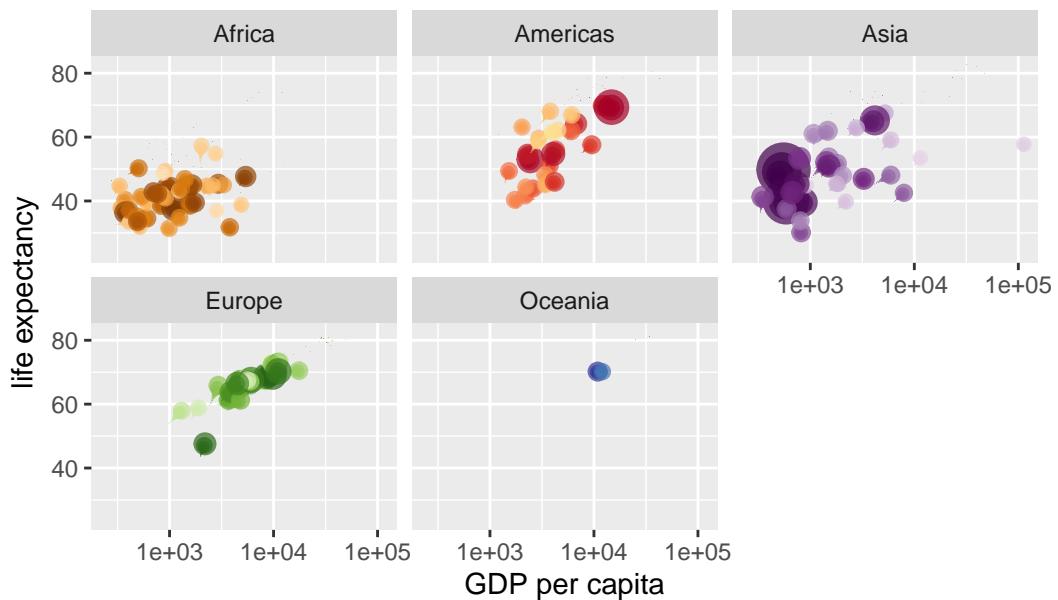
Year: 1955



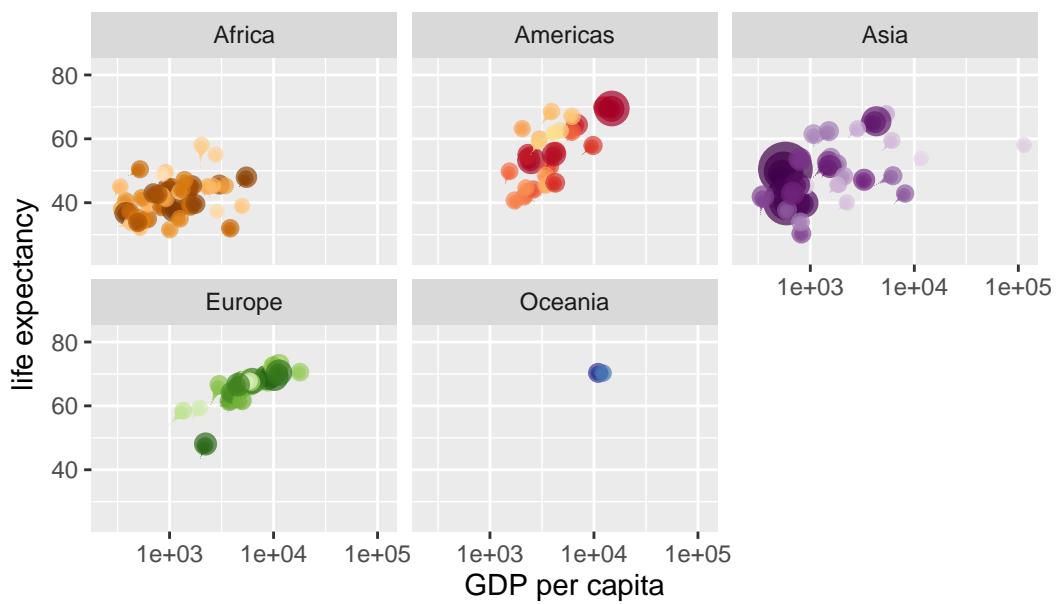
Year: 1956



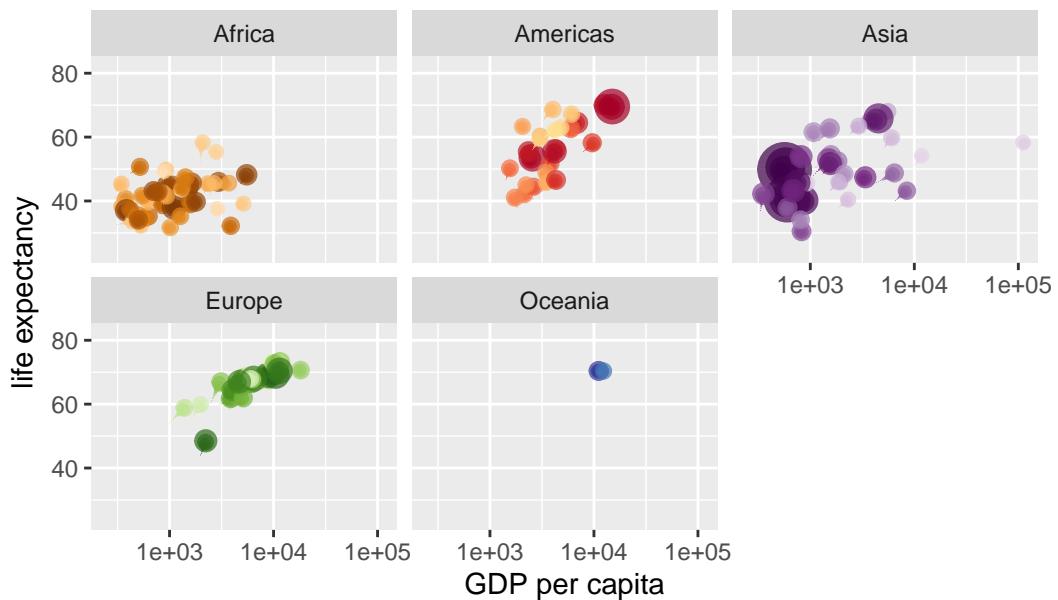
Year: 1956



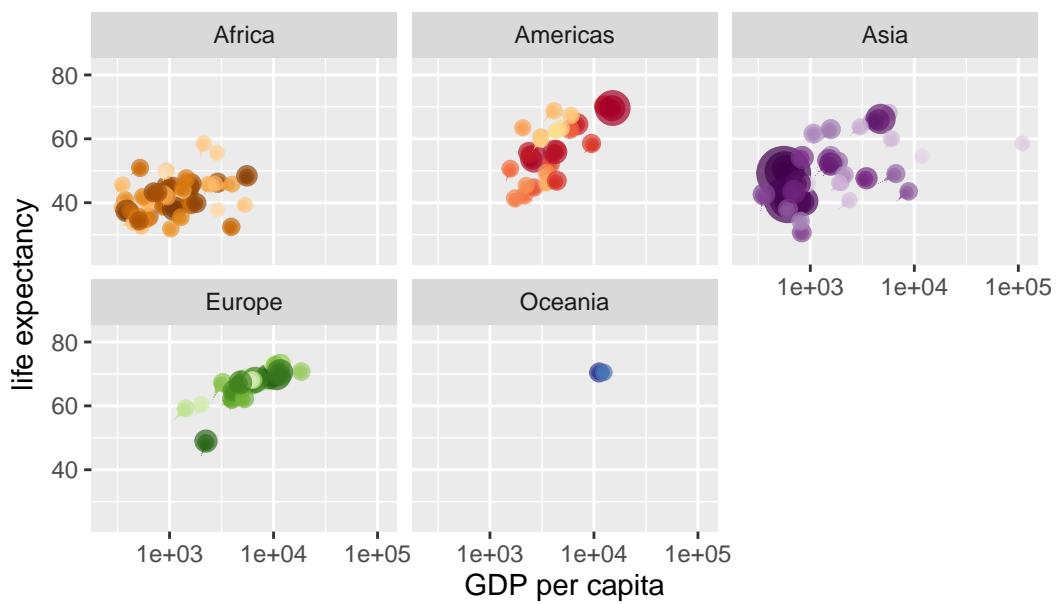
Year: 1957



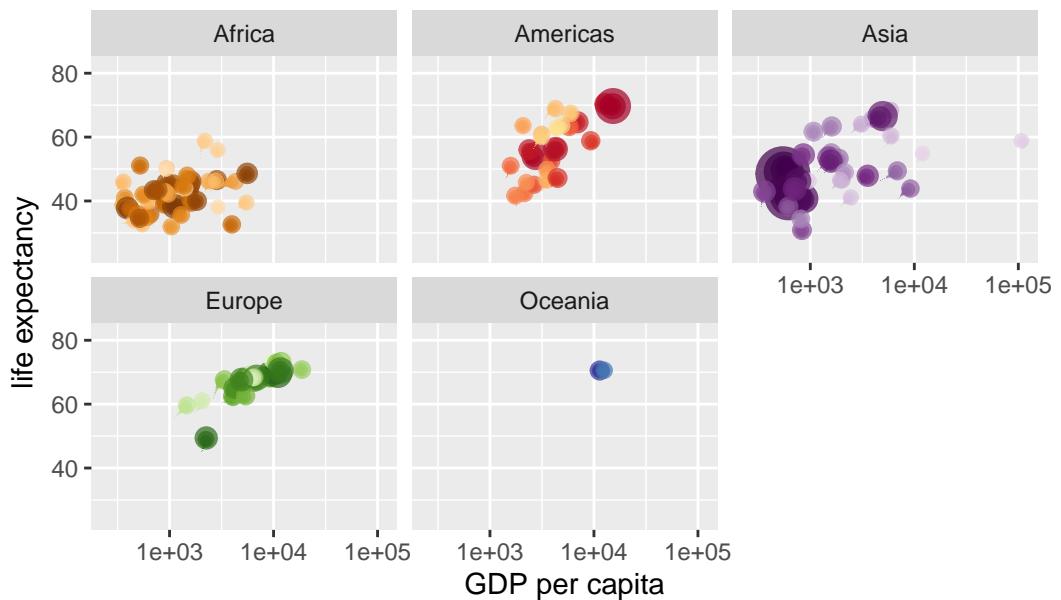
Year: 1958



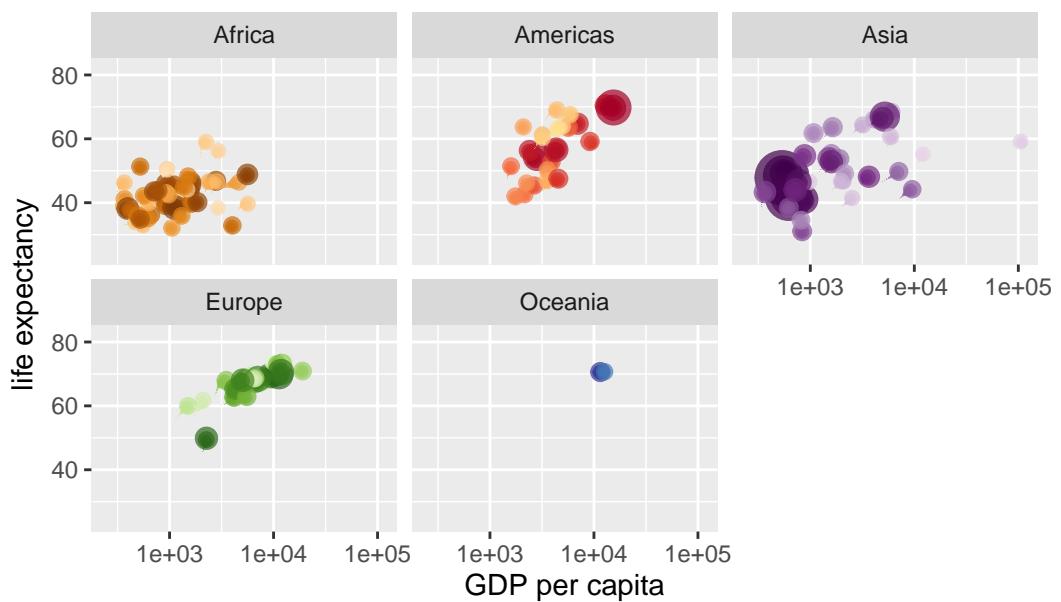
Year: 1958



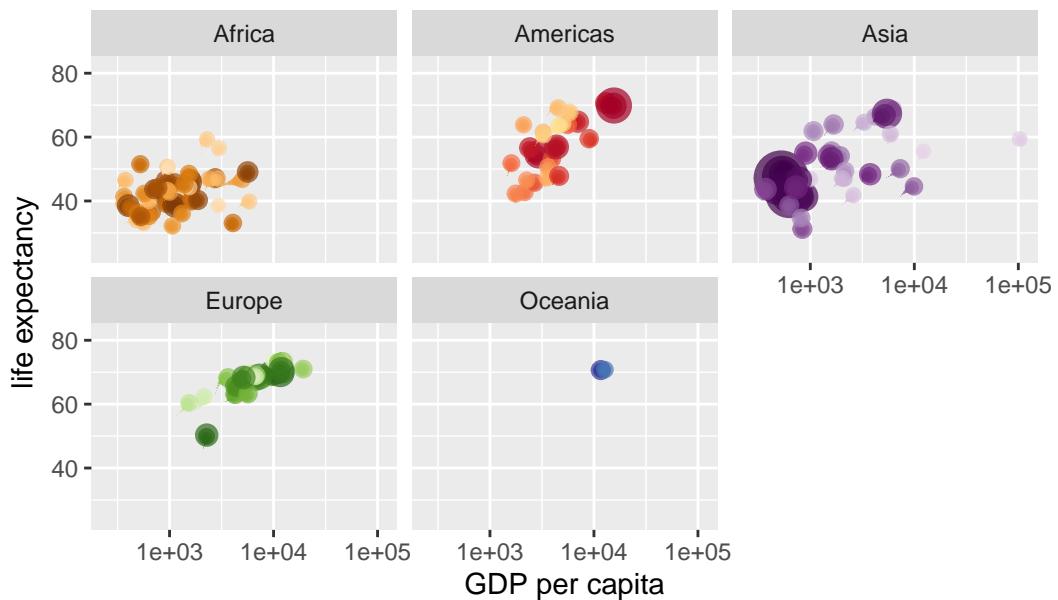
Year: 1959



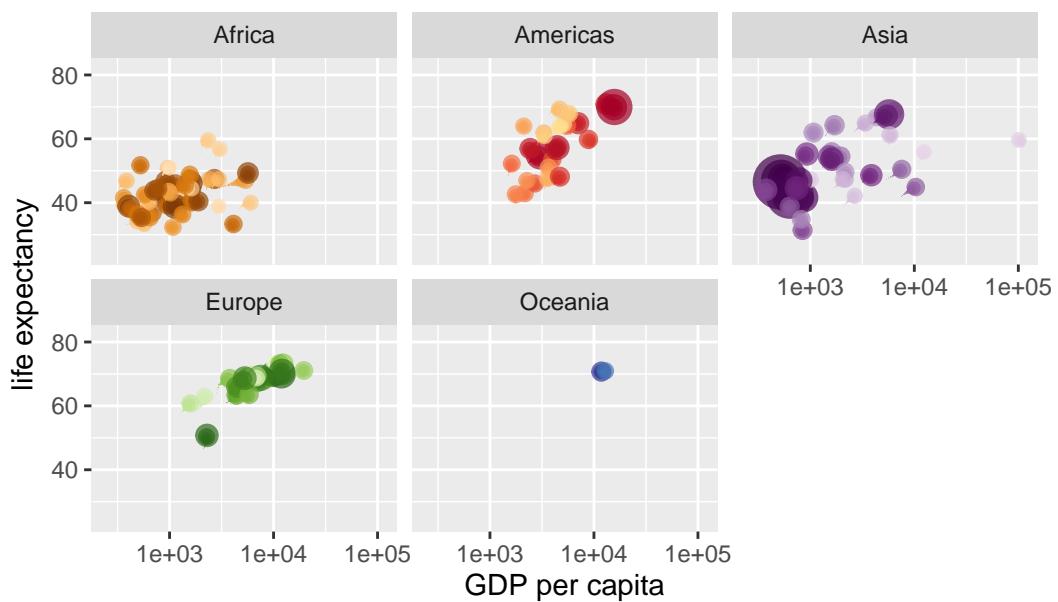
Year: 1959



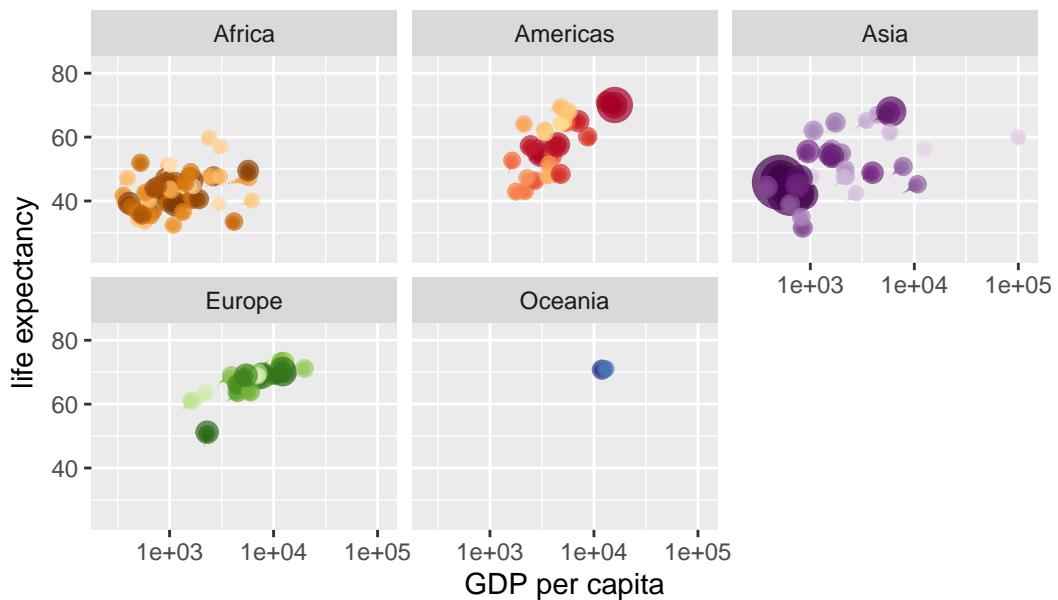
Year: 1960



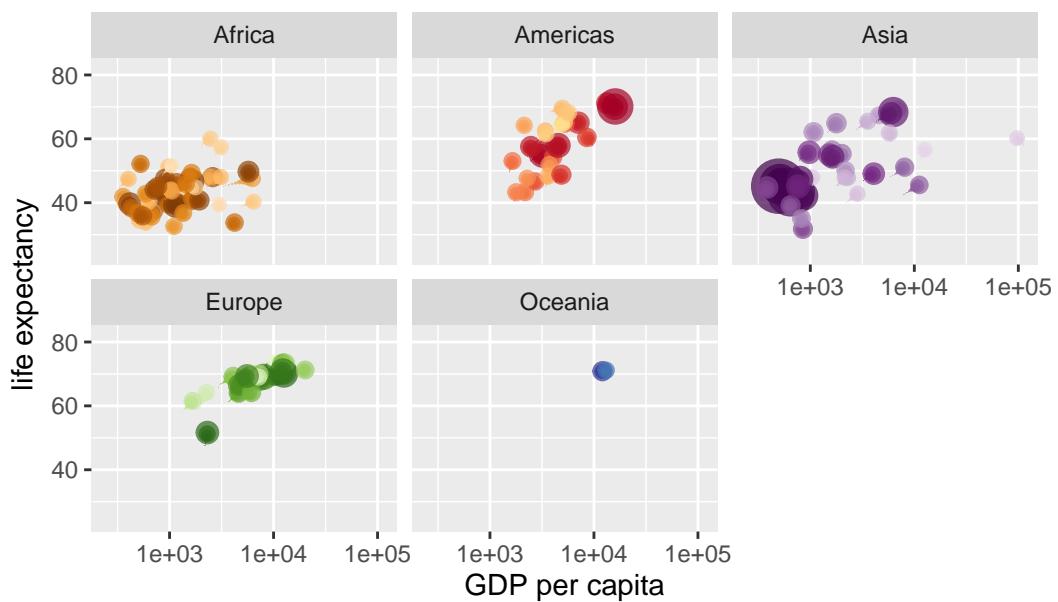
Year: 1960



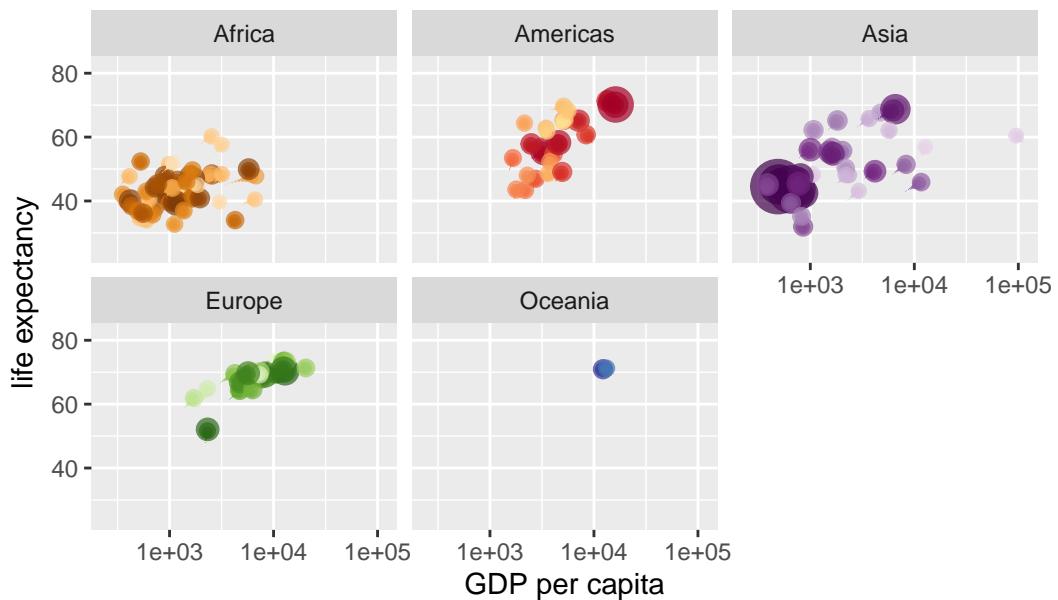
Year: 1961



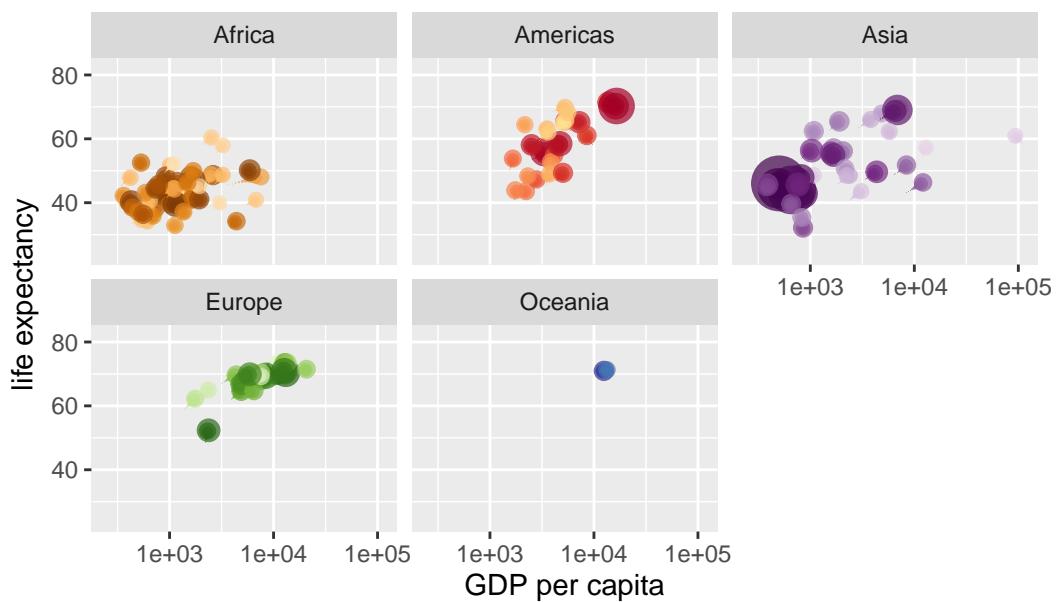
Year: 1961



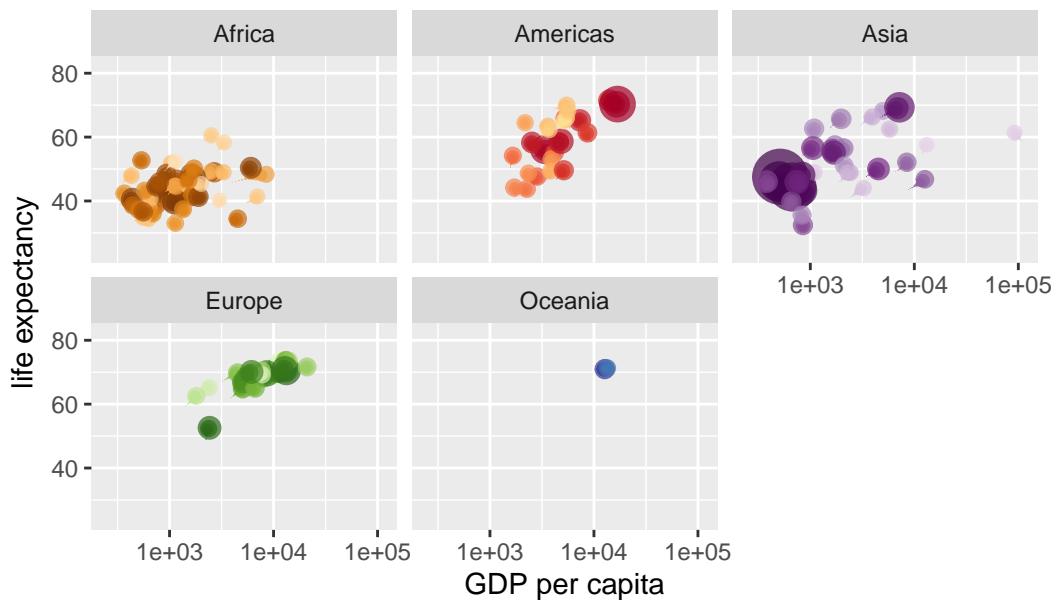
Year: 1962



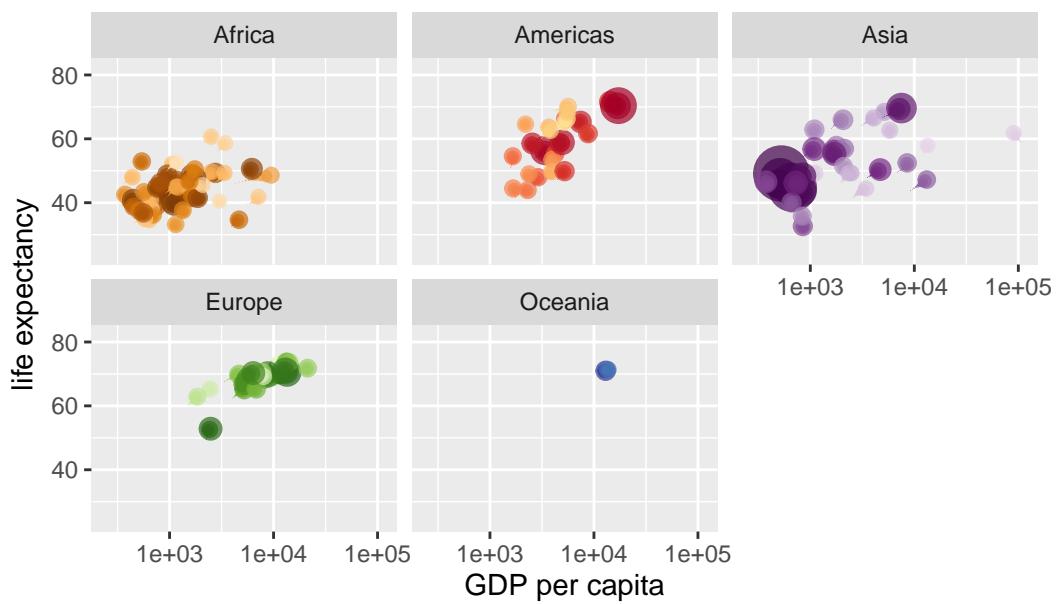
Year: 1963



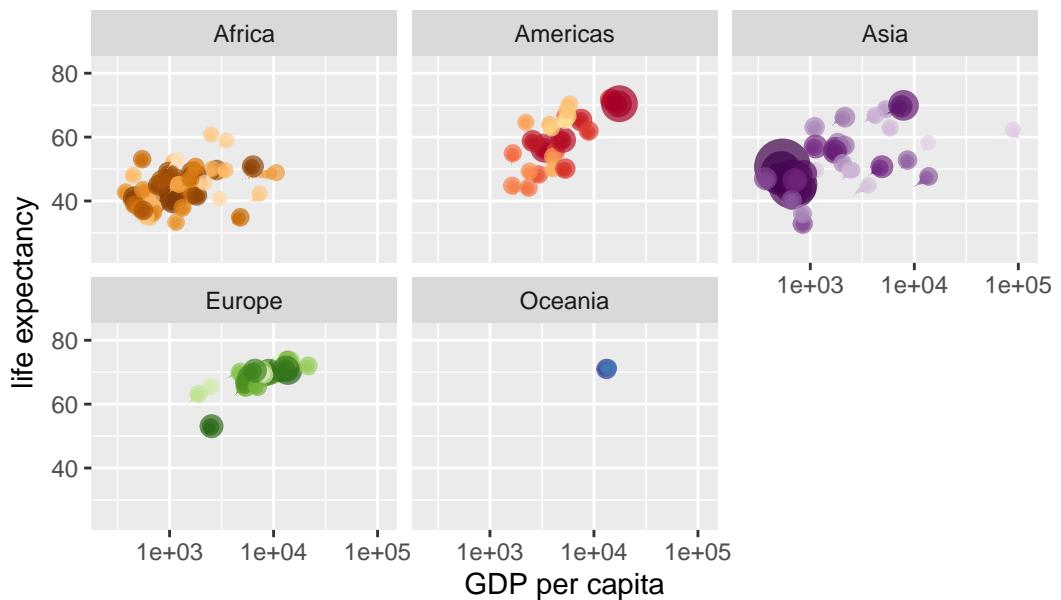
Year: 1963



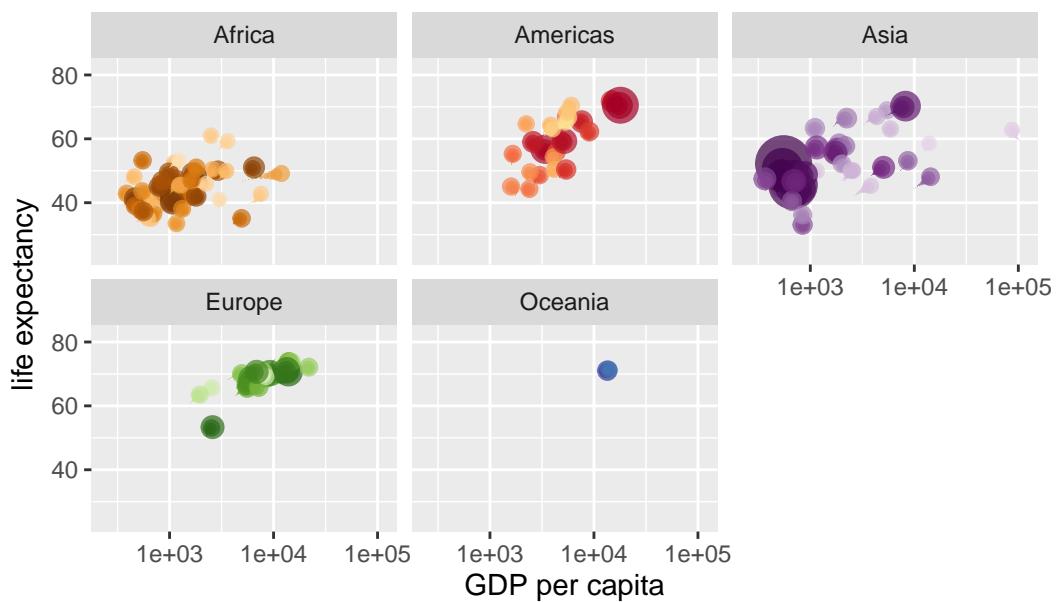
Year: 1964



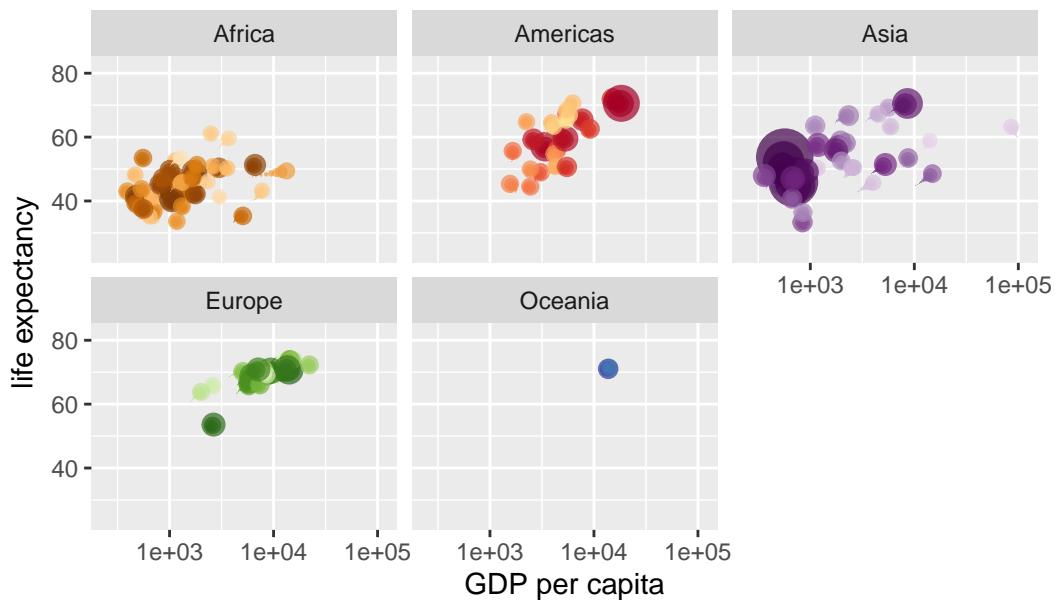
Year: 1964



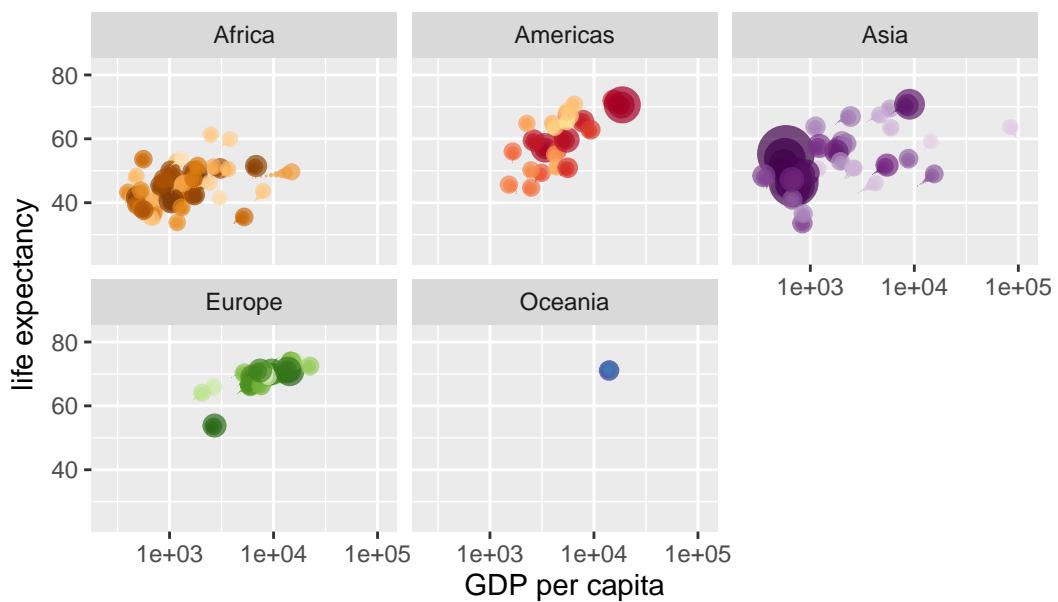
Year: 1965



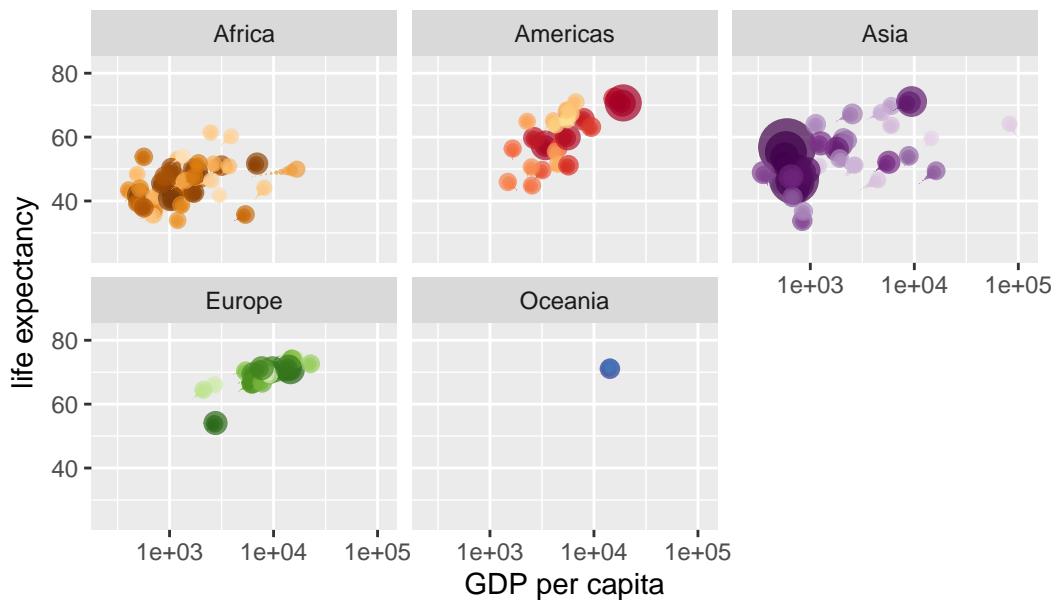
Year: 1965



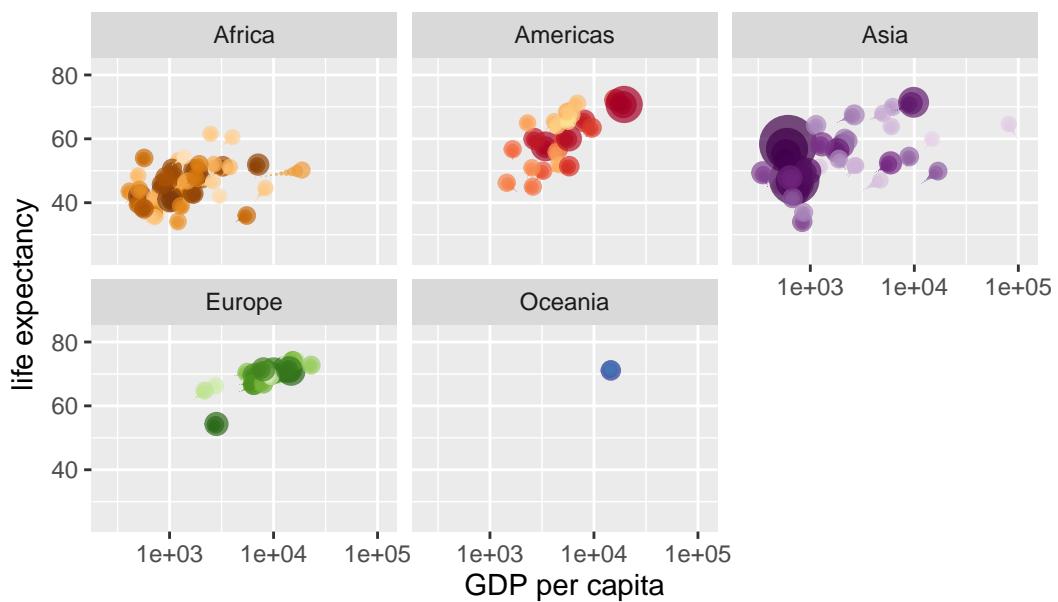
Year: 1966



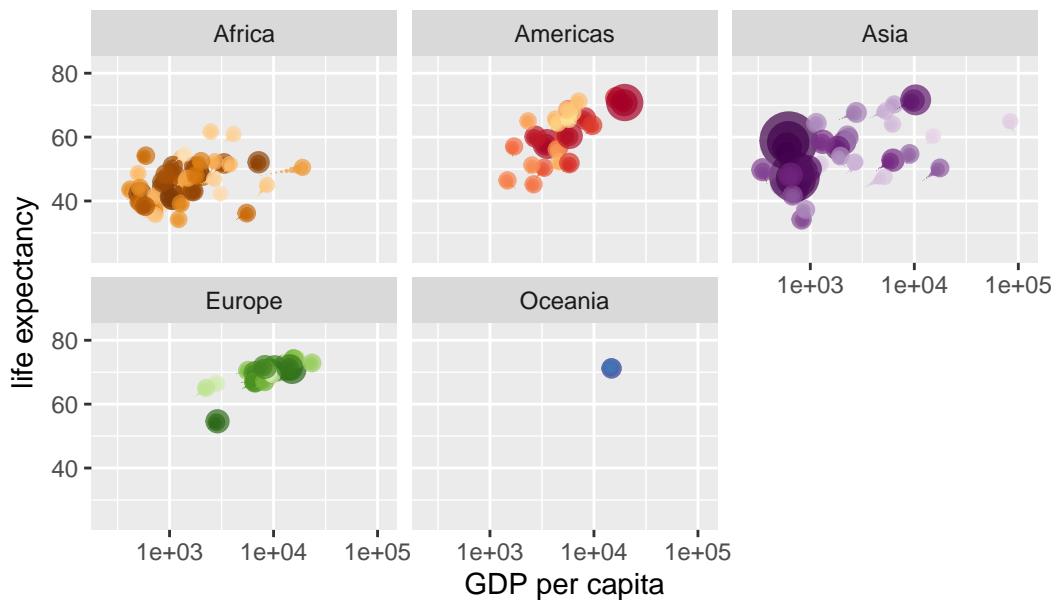
Year: 1966



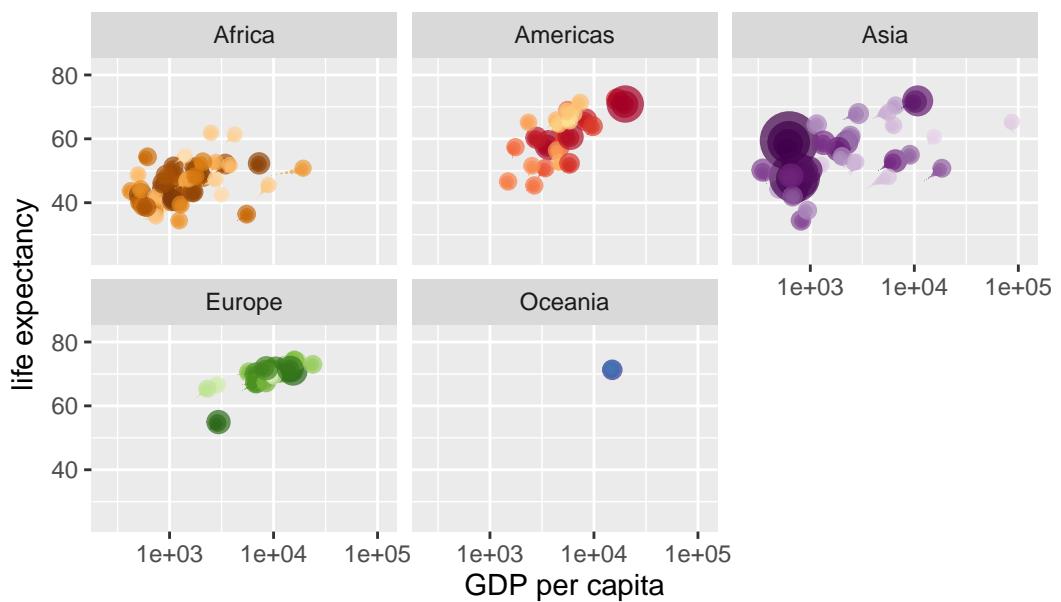
Year: 1967



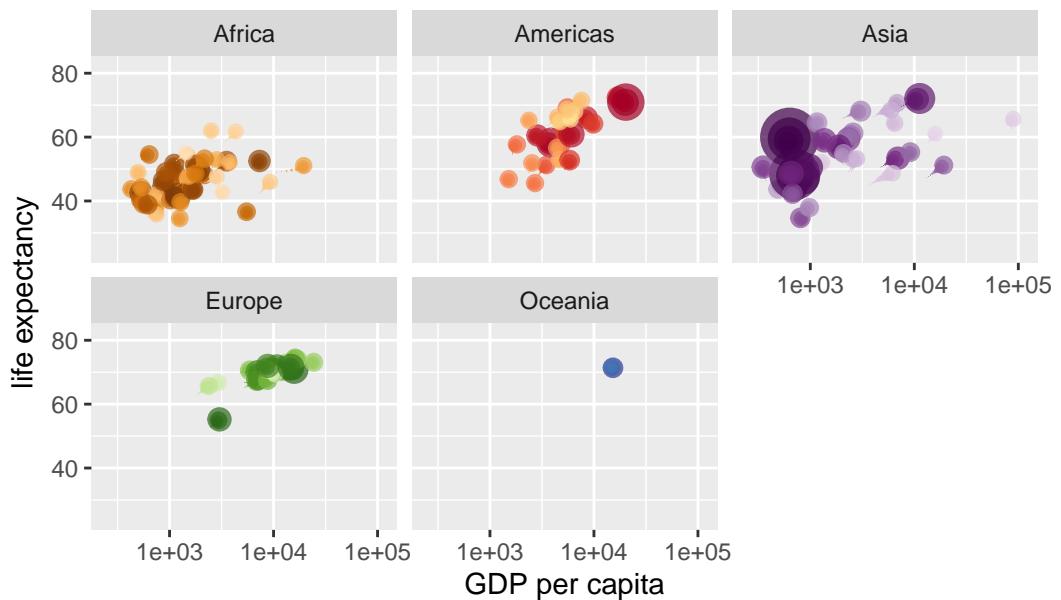
Year: 1968



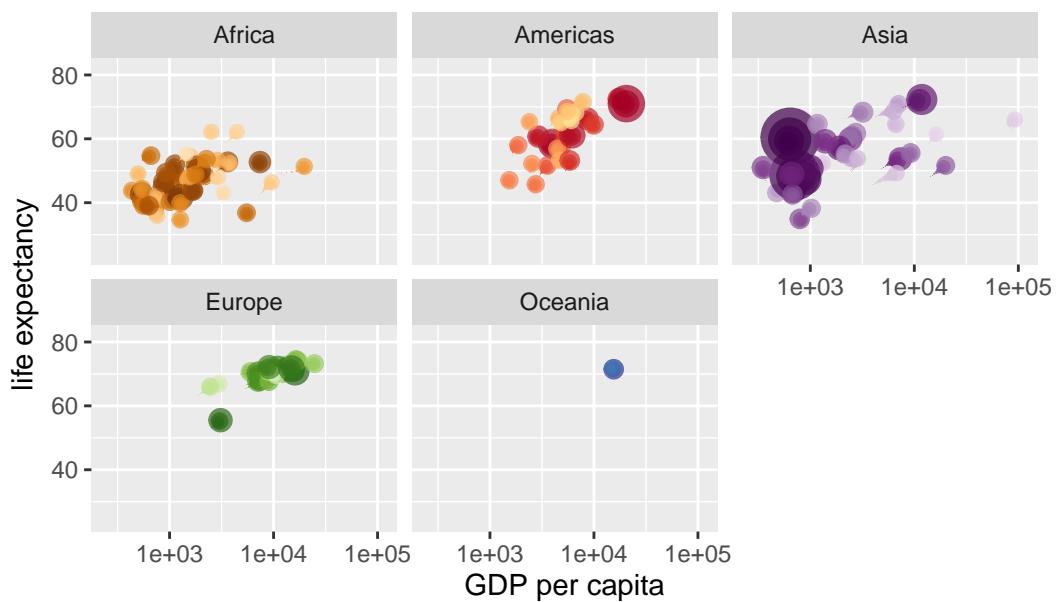
Year: 1968



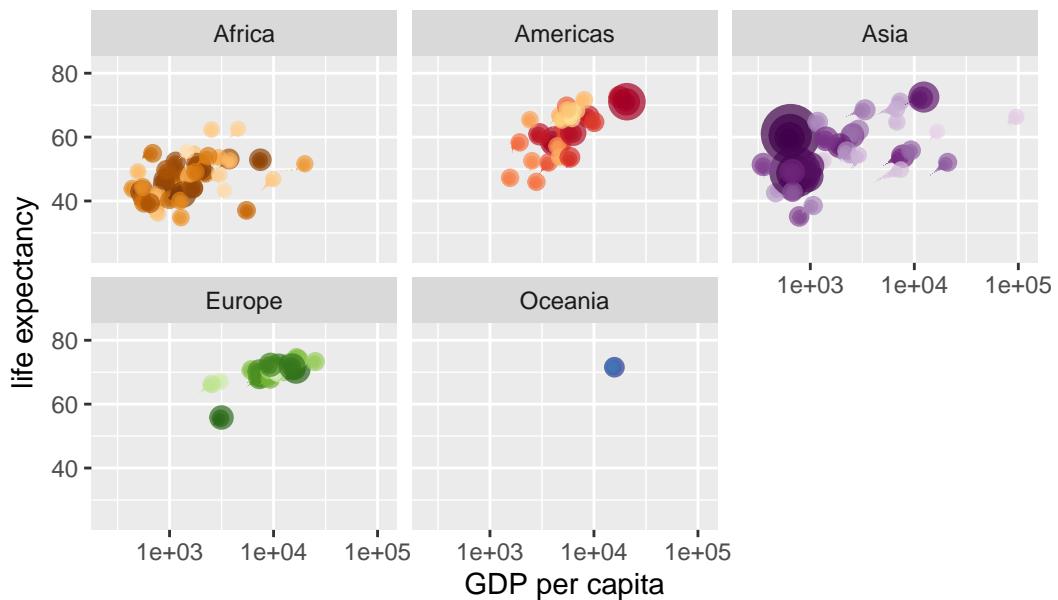
Year: 1969



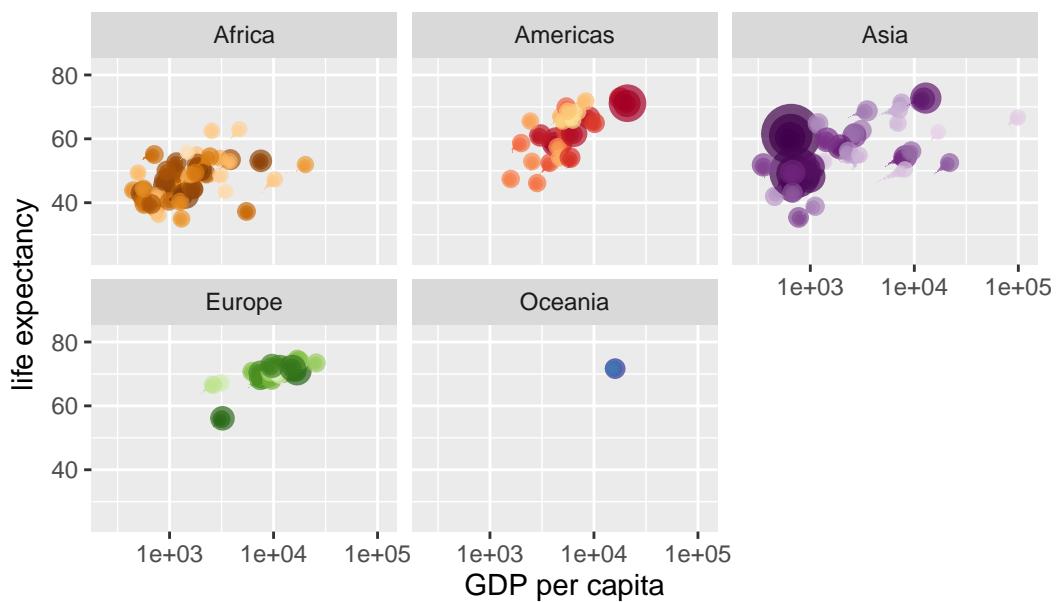
Year: 1969



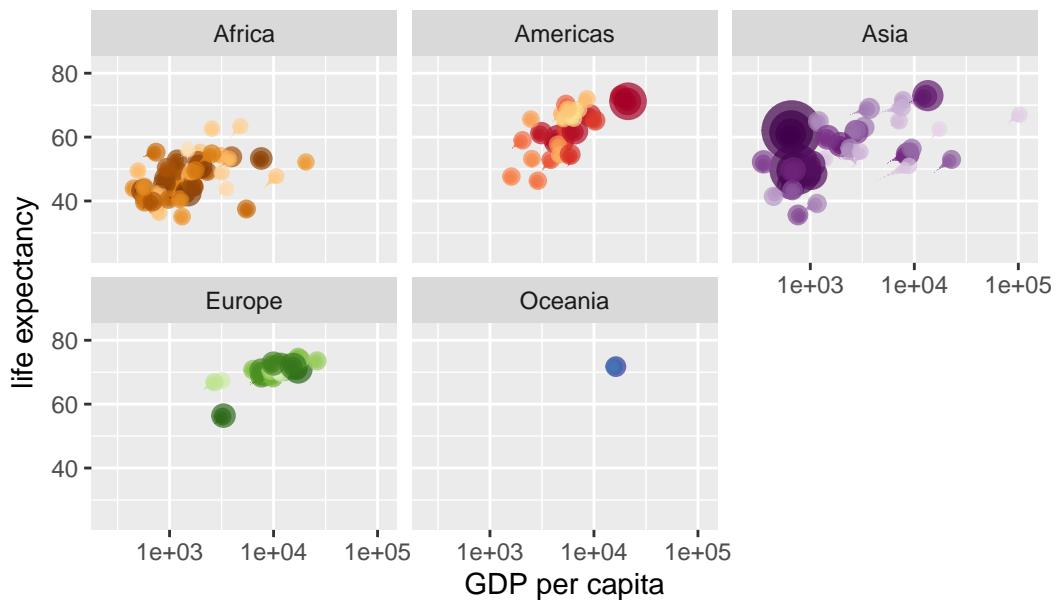
Year: 1970



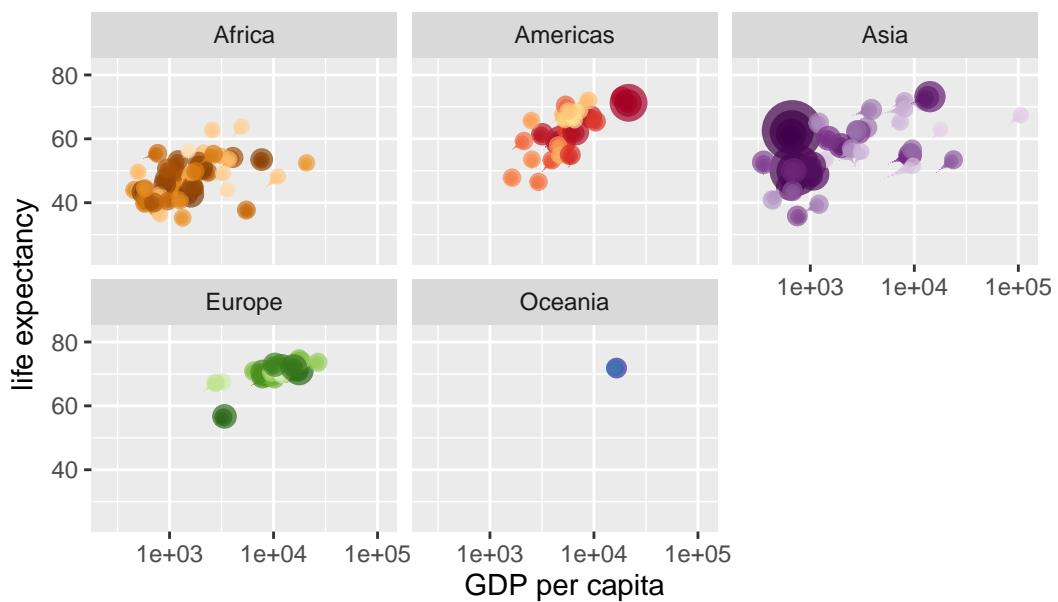
Year: 1970



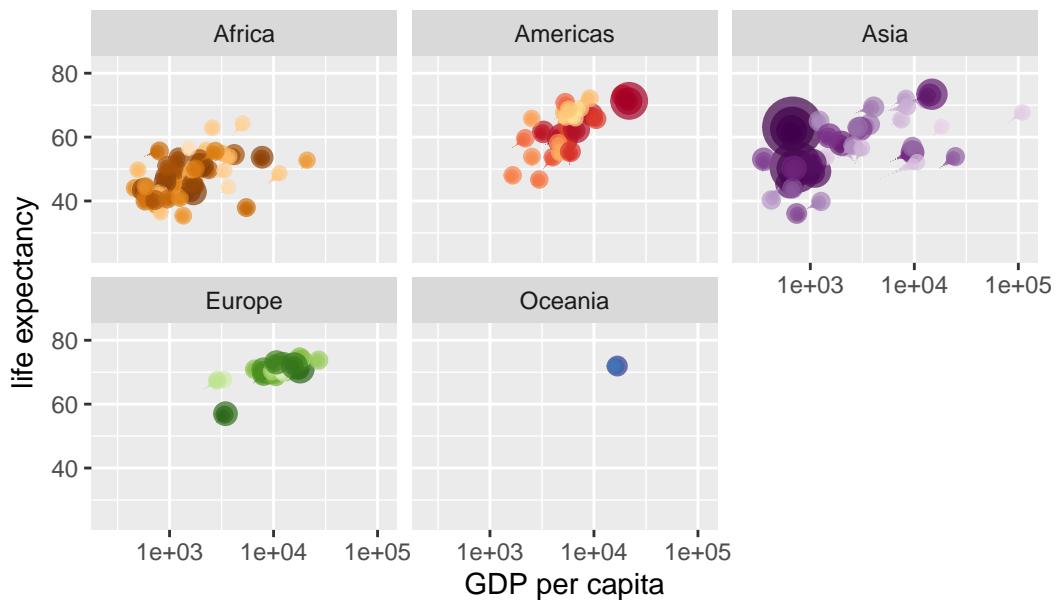
Year: 1971



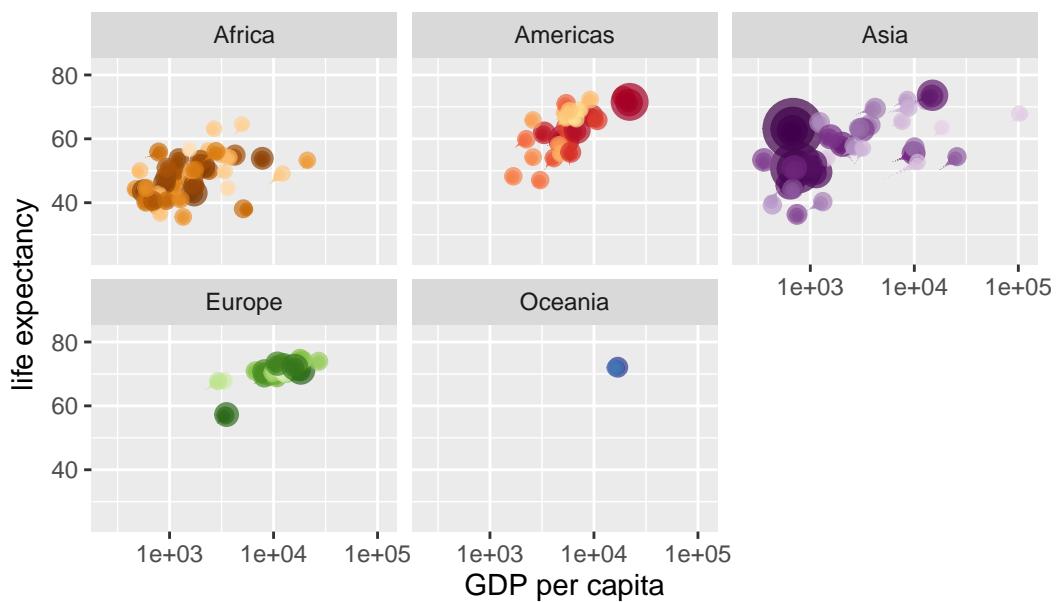
Year: 1971



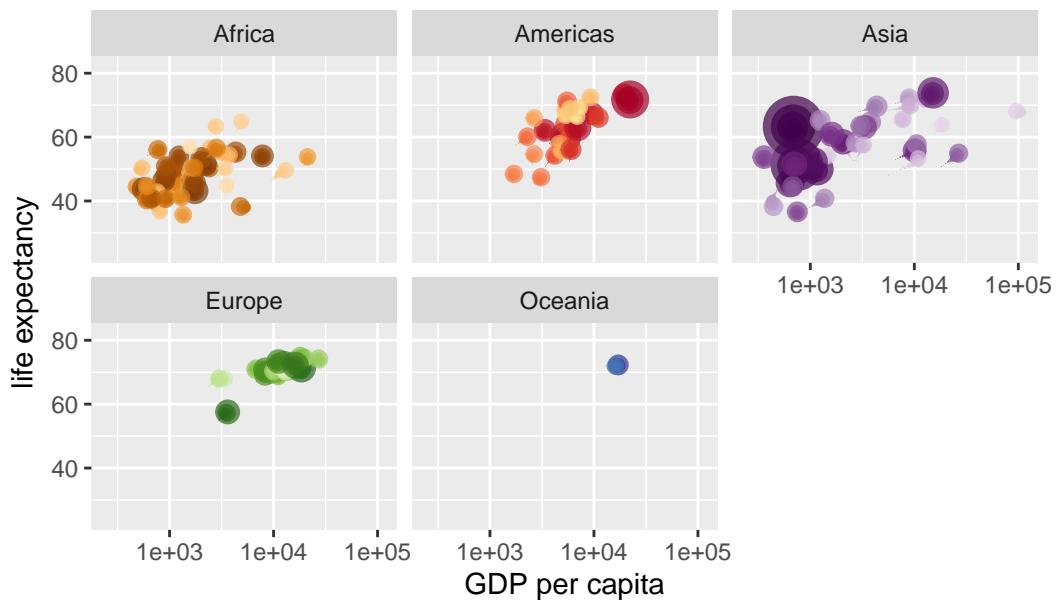
Year: 1972



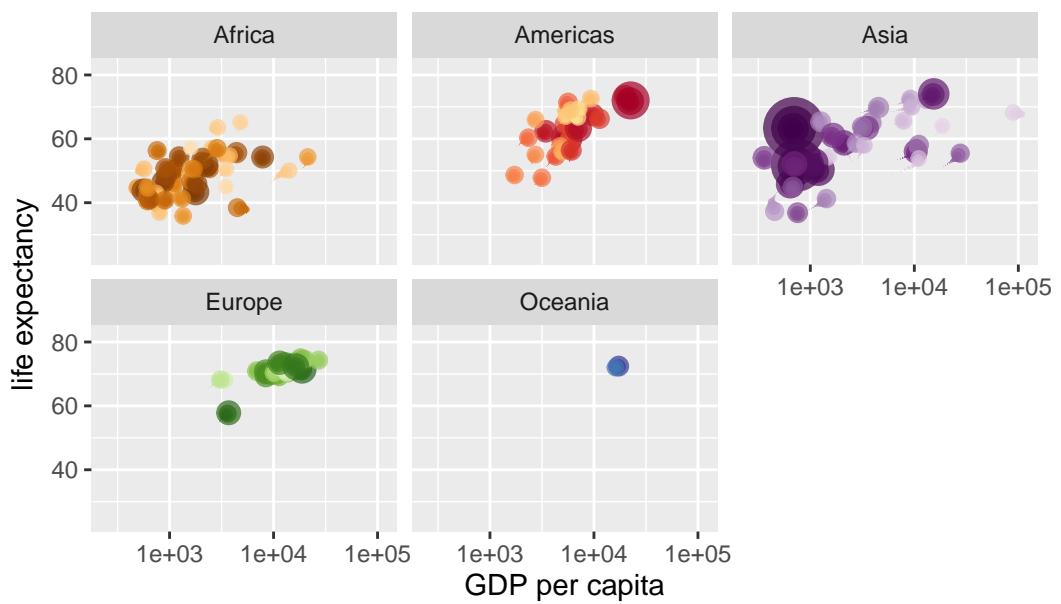
Year: 1973



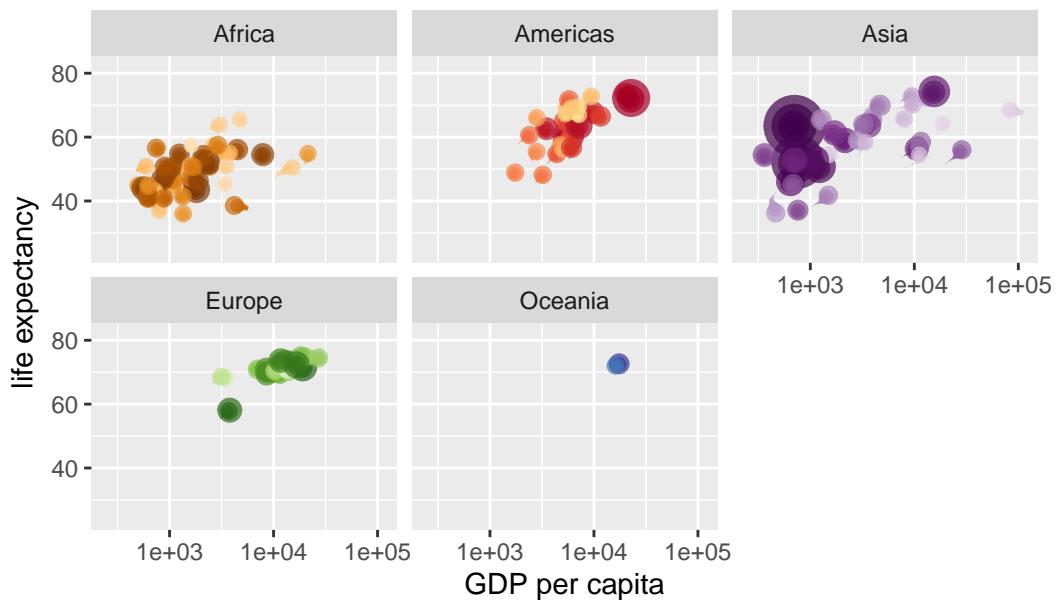
Year: 1973



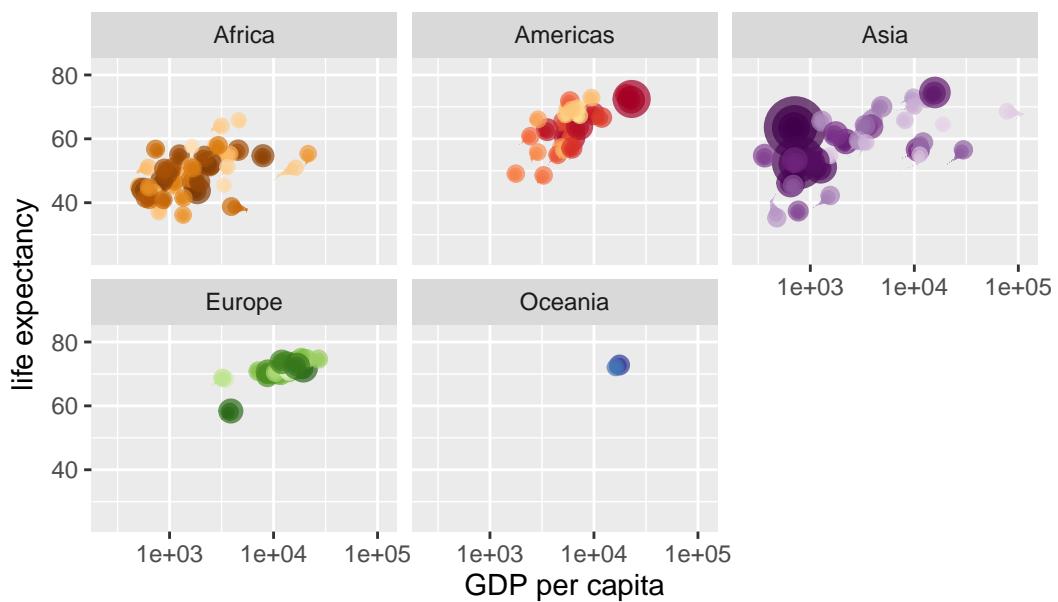
Year: 1974



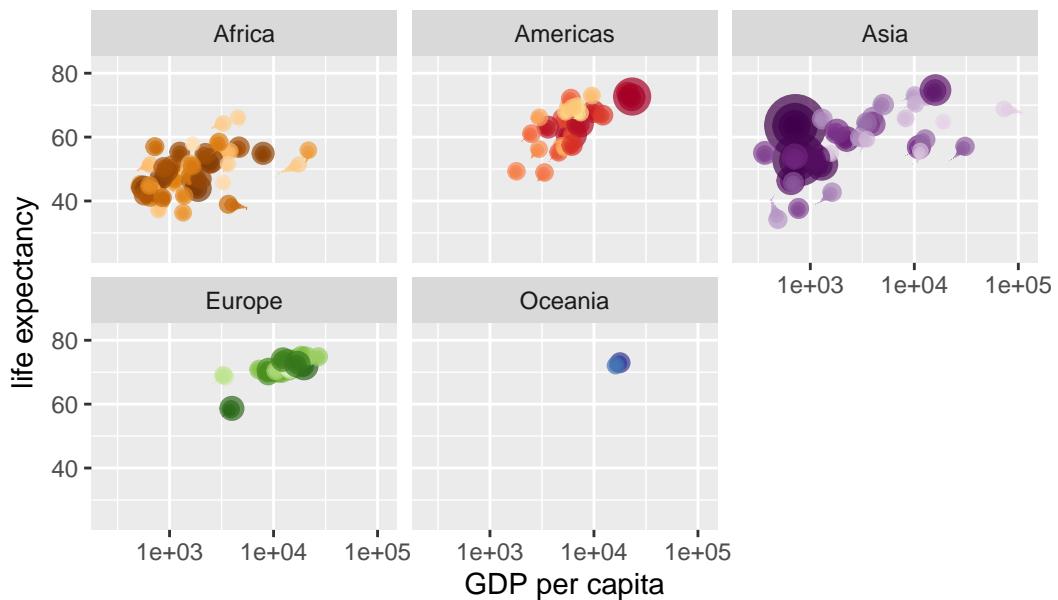
Year: 1974



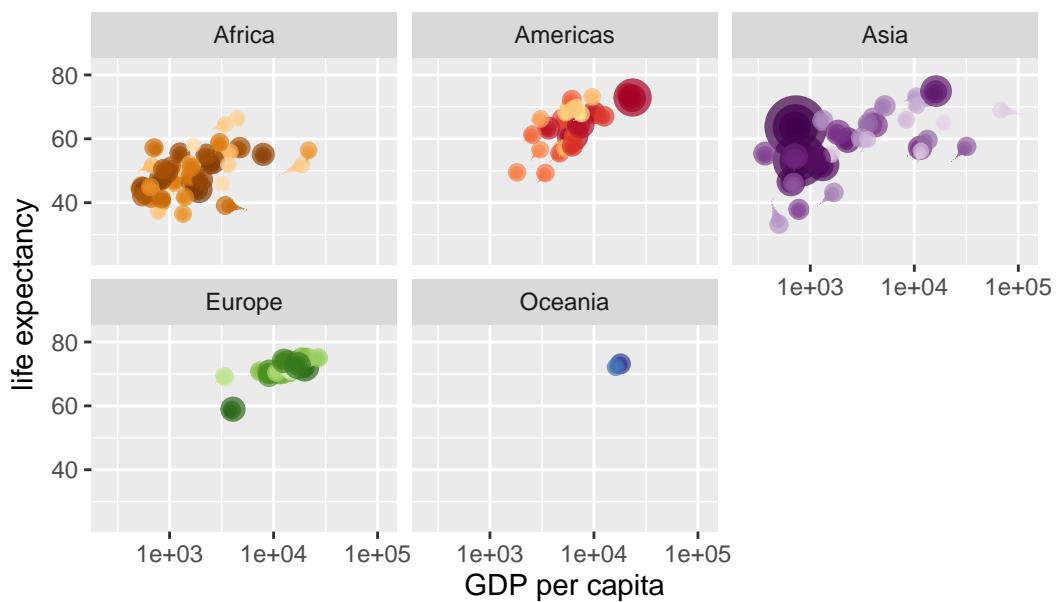
Year: 1975



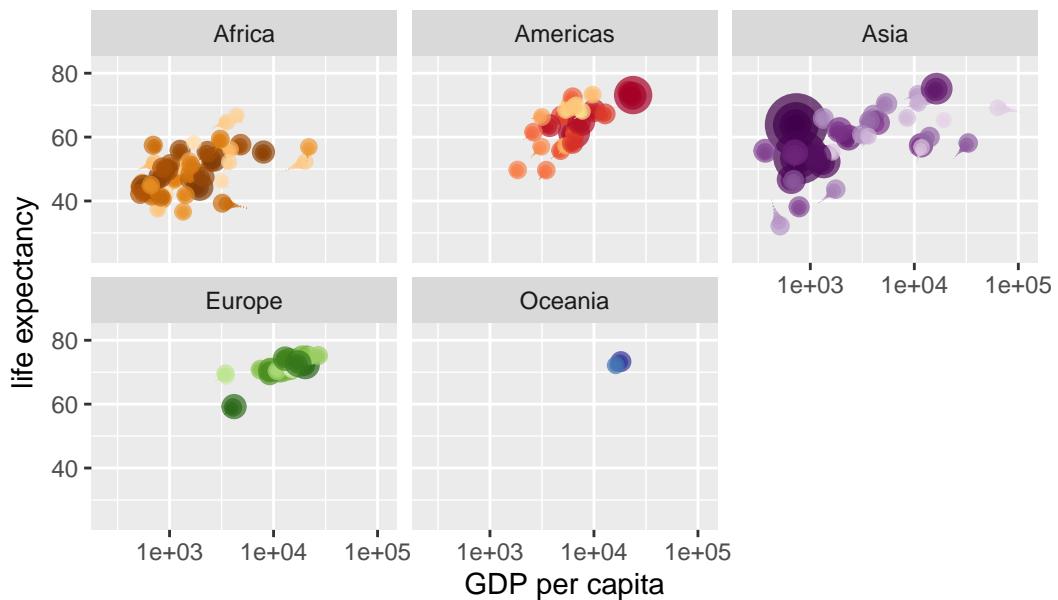
Year: 1975



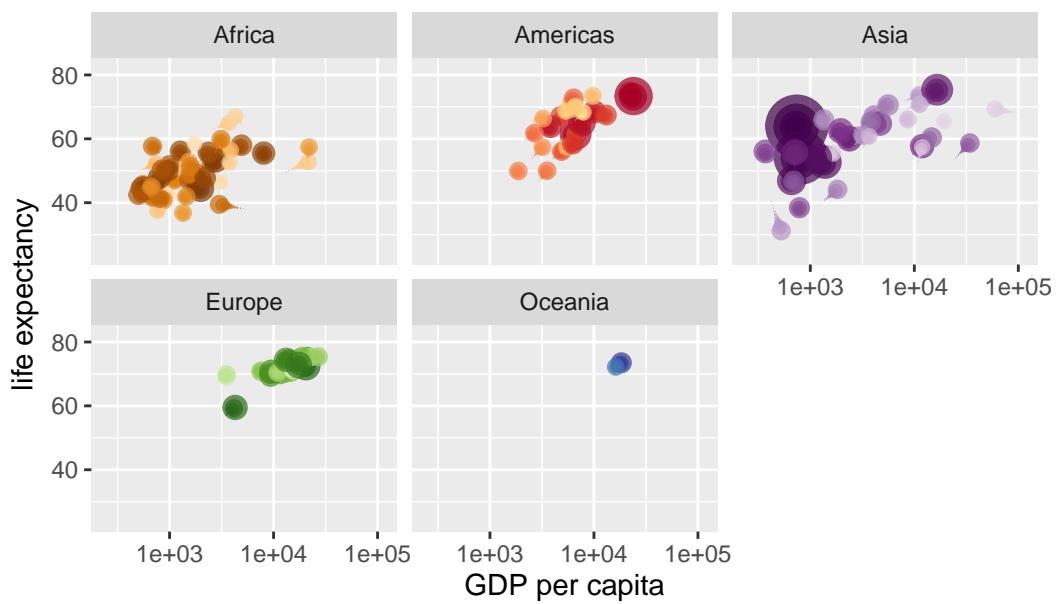
Year: 1976



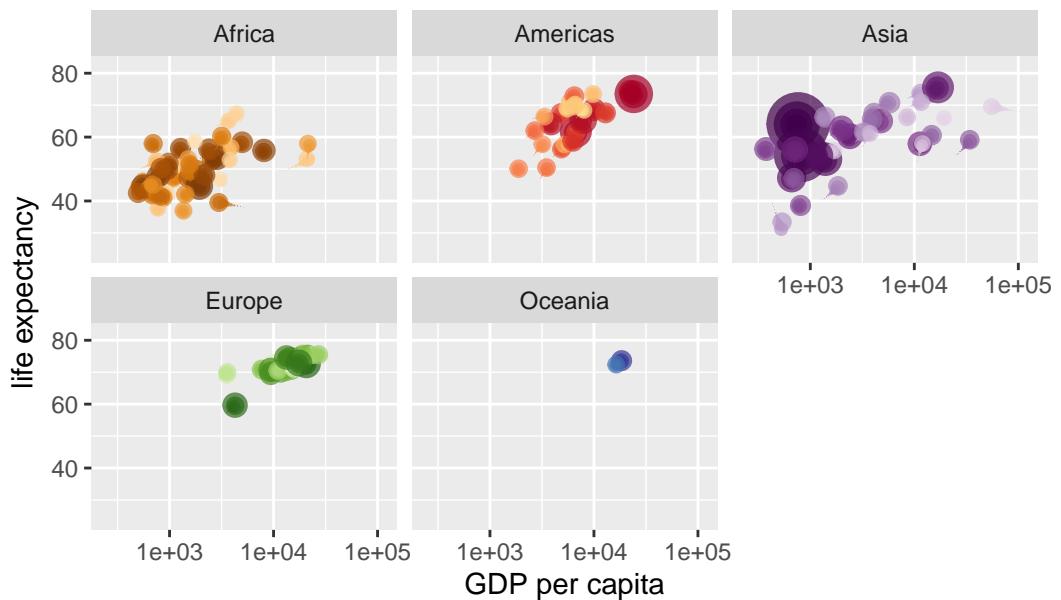
Year: 1976



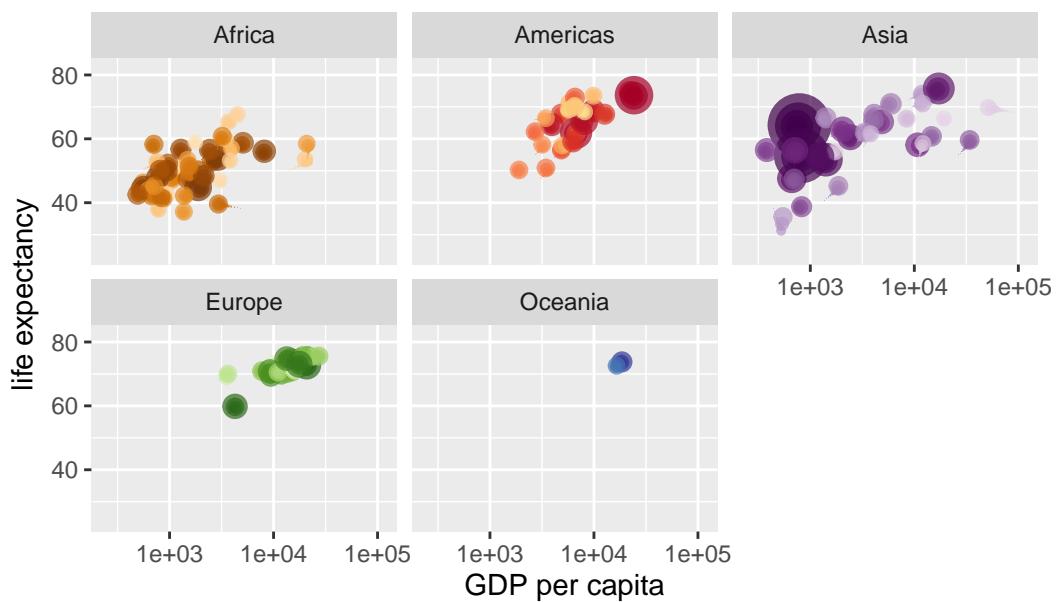
Year: 1977



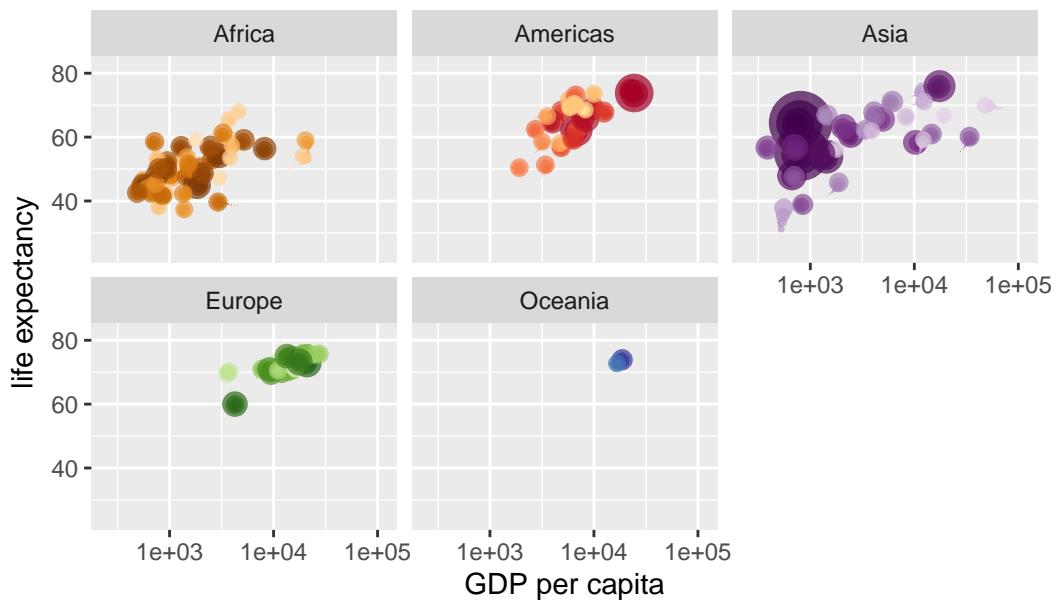
Year: 1978



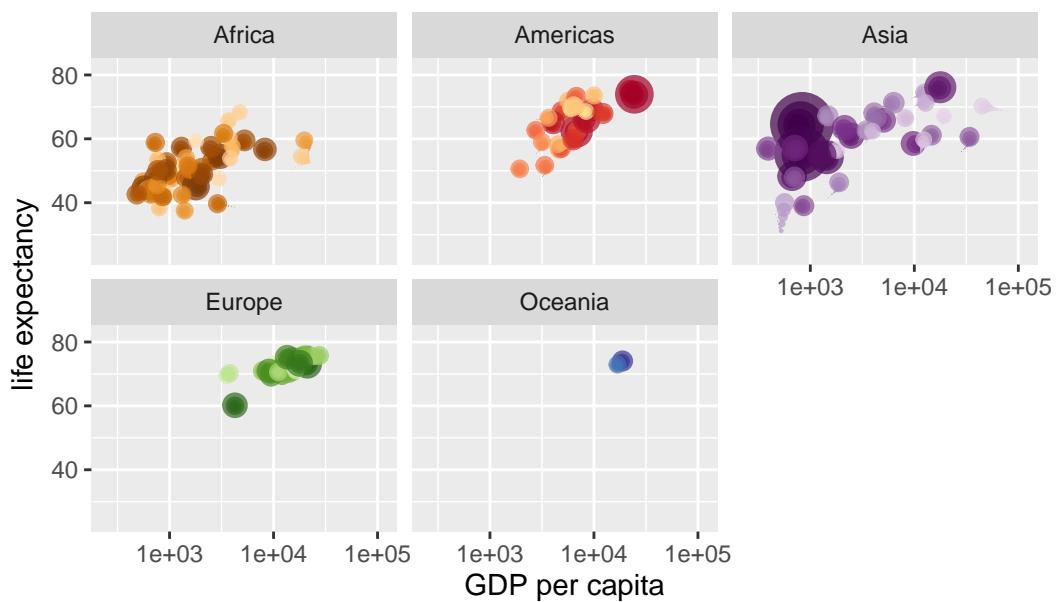
Year: 1978



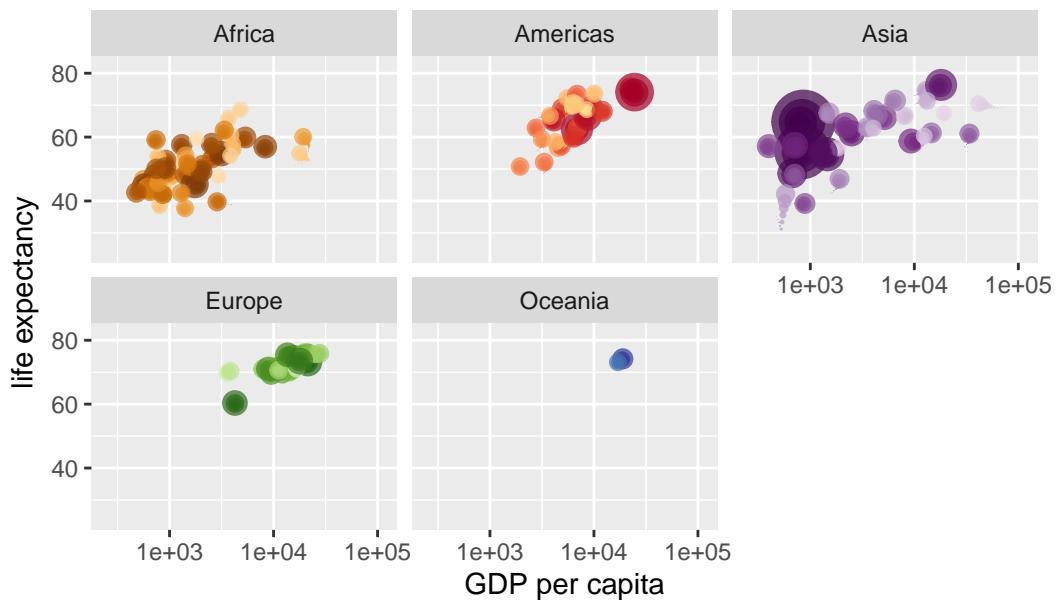
Year: 1979



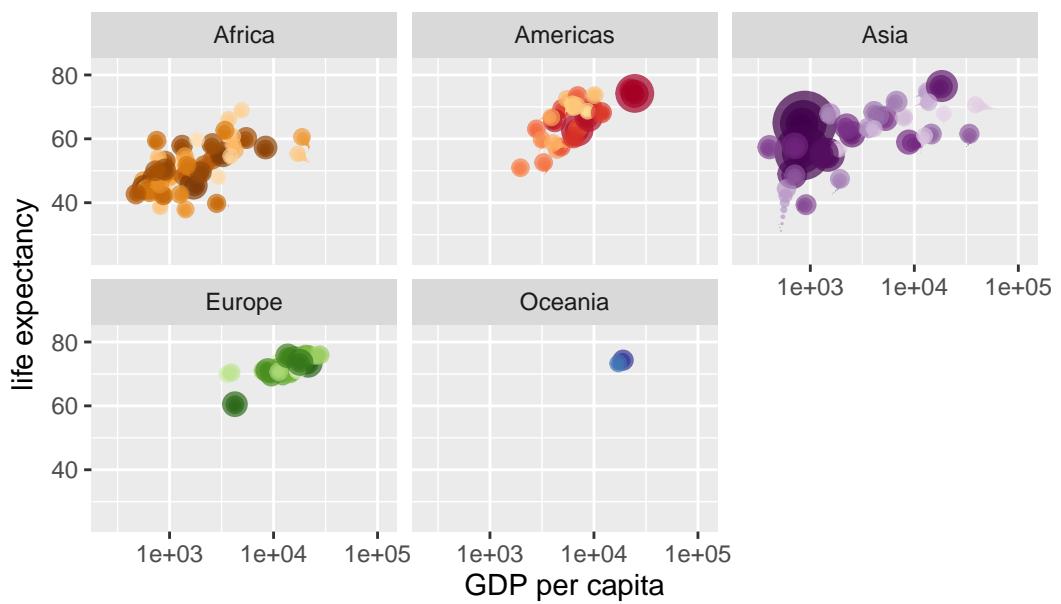
Year: 1979



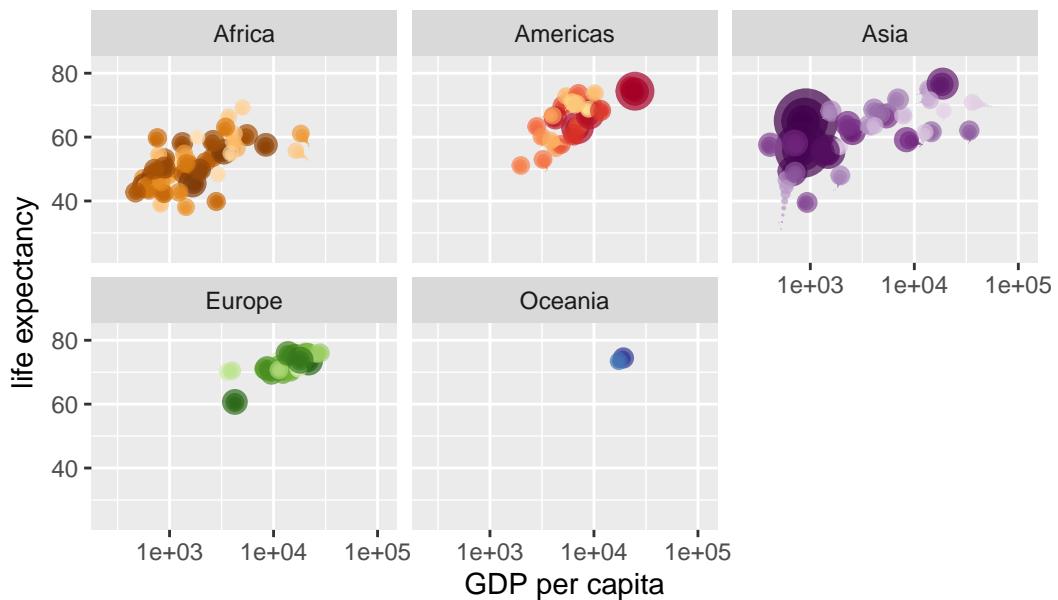
Year: 1980



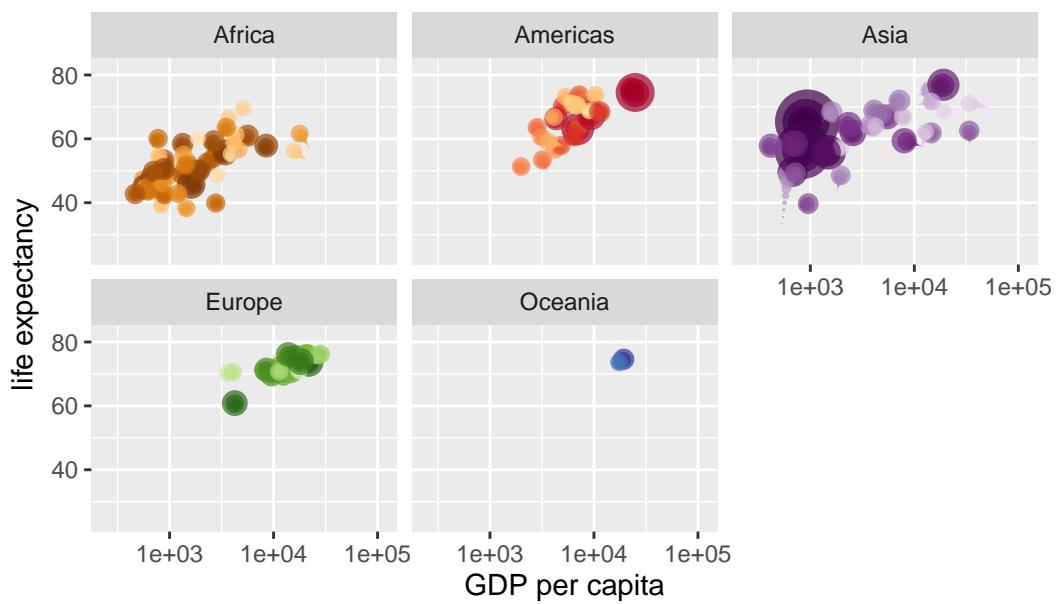
Year: 1980



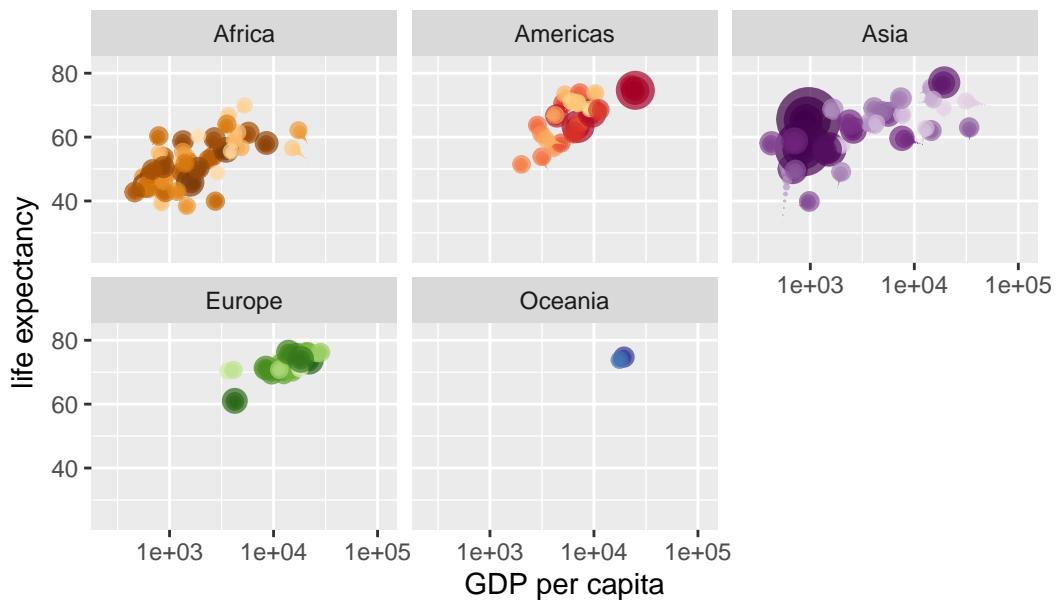
Year: 1981



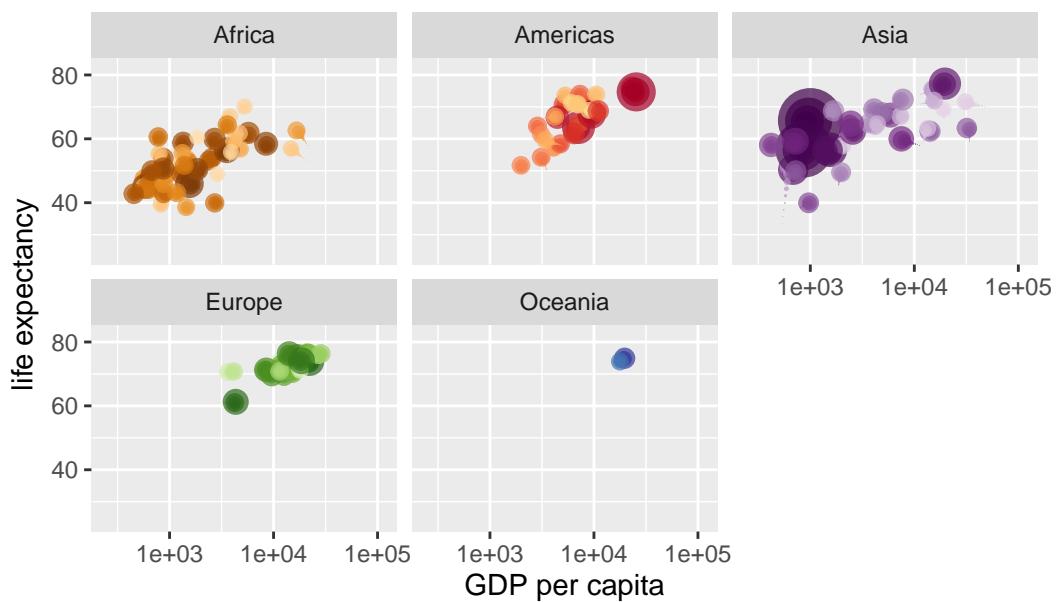
Year: 1981



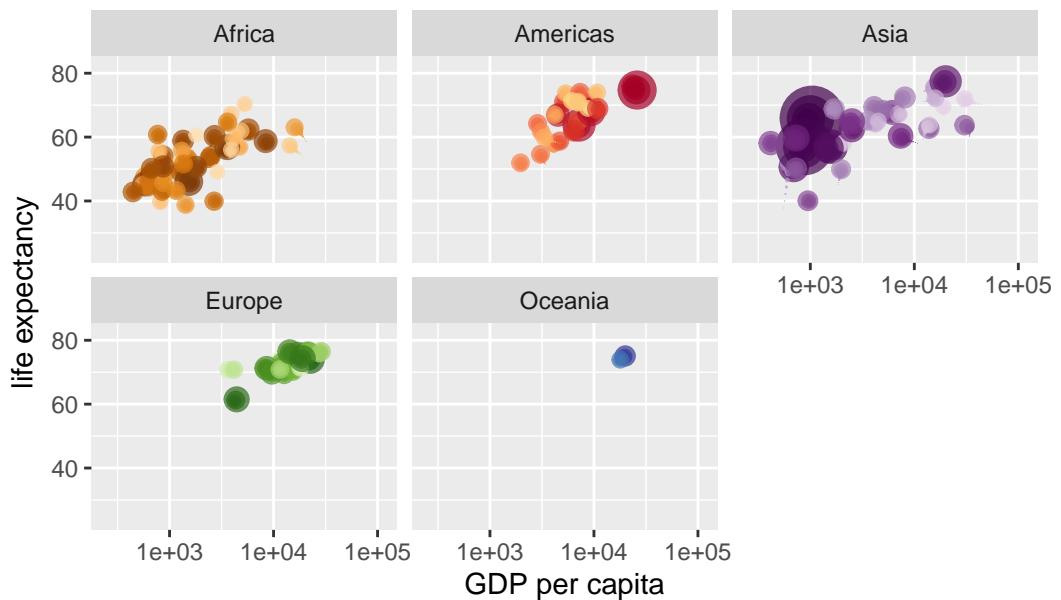
Year: 1982



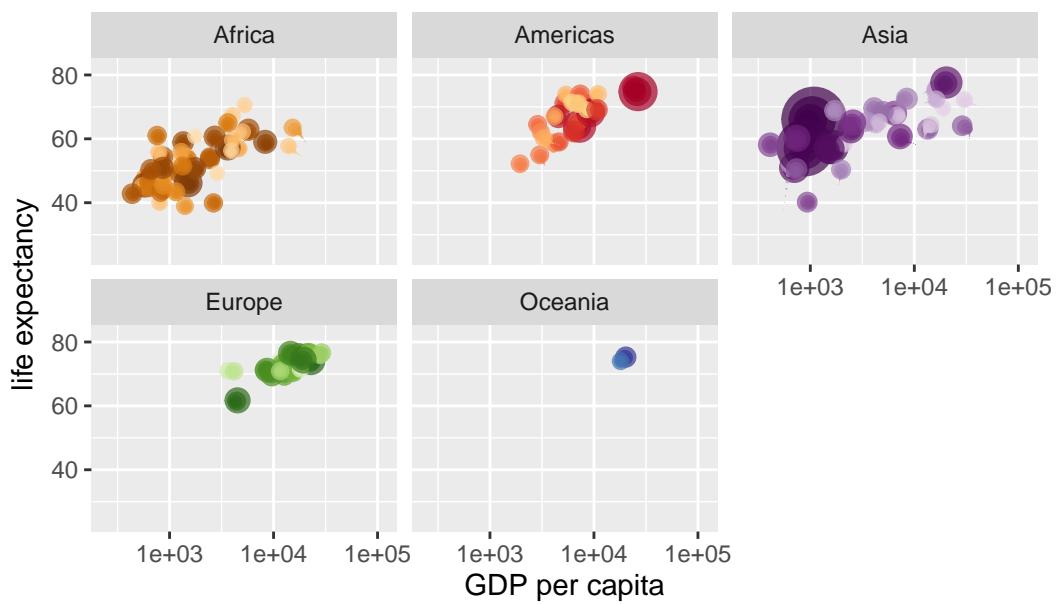
Year: 1983



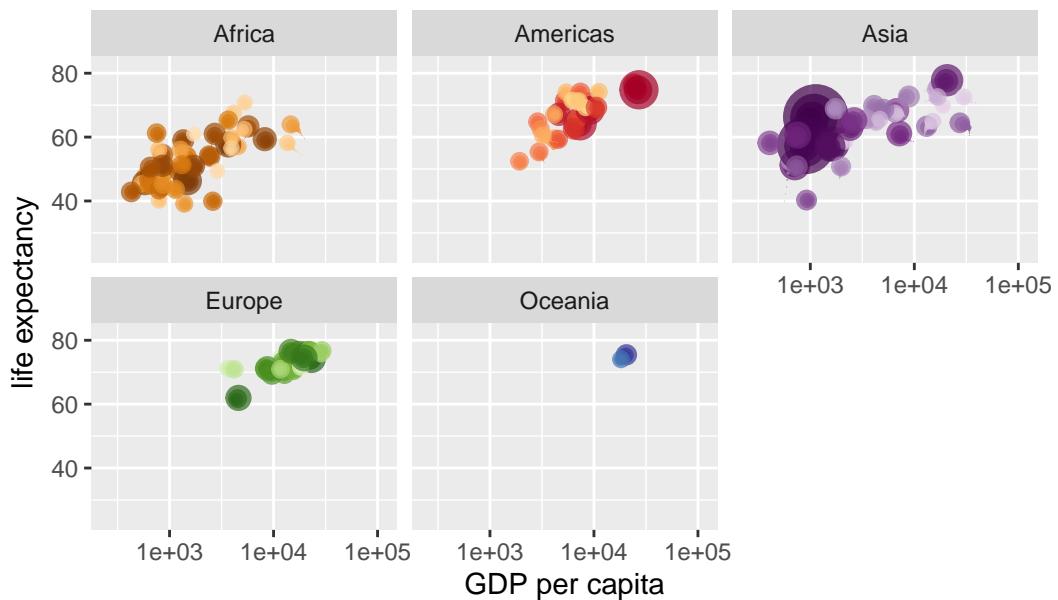
Year: 1983



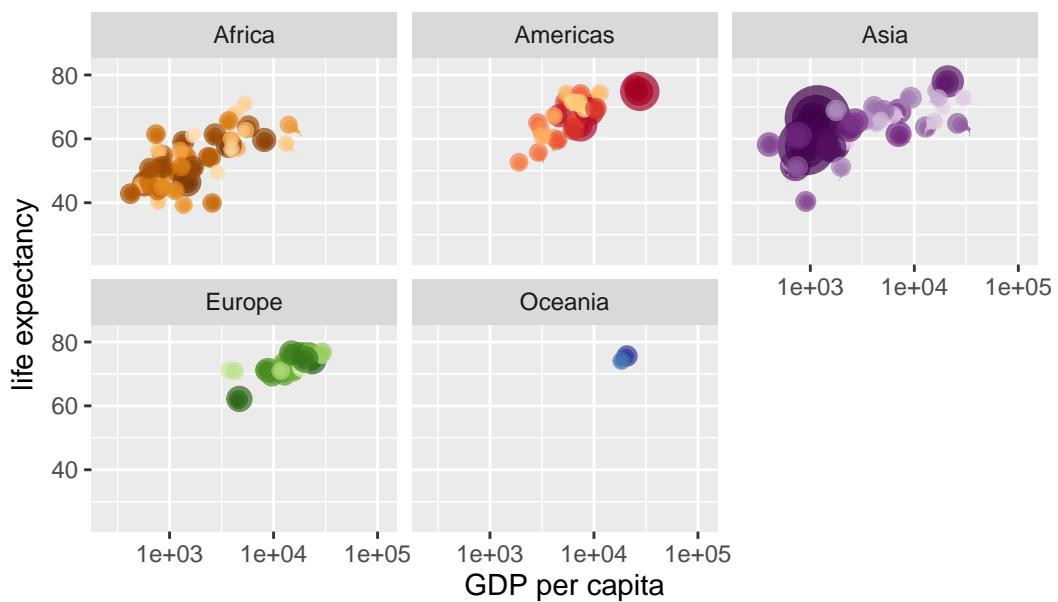
Year: 1984



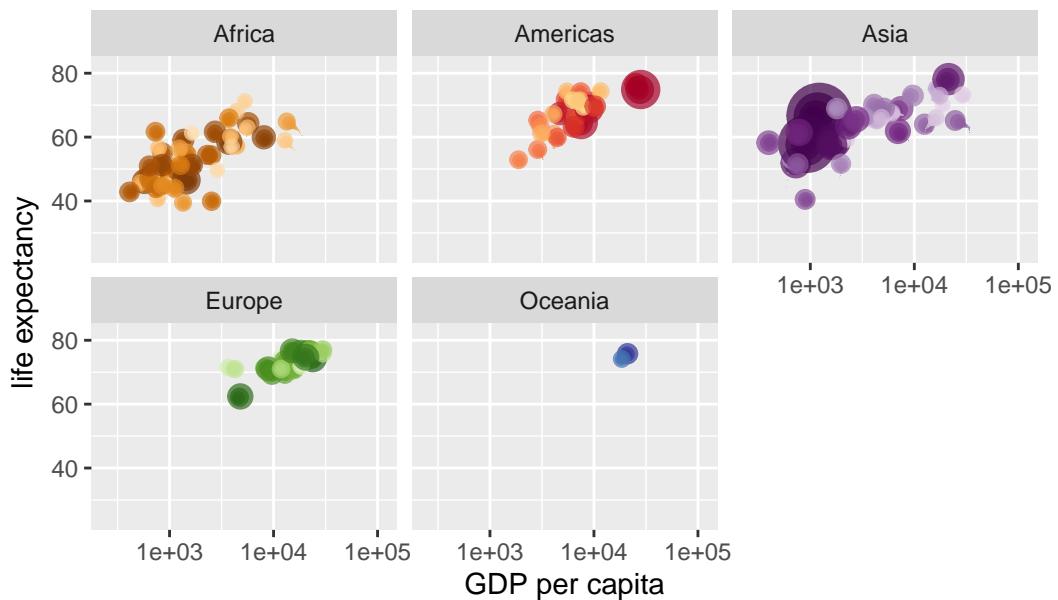
Year: 1984



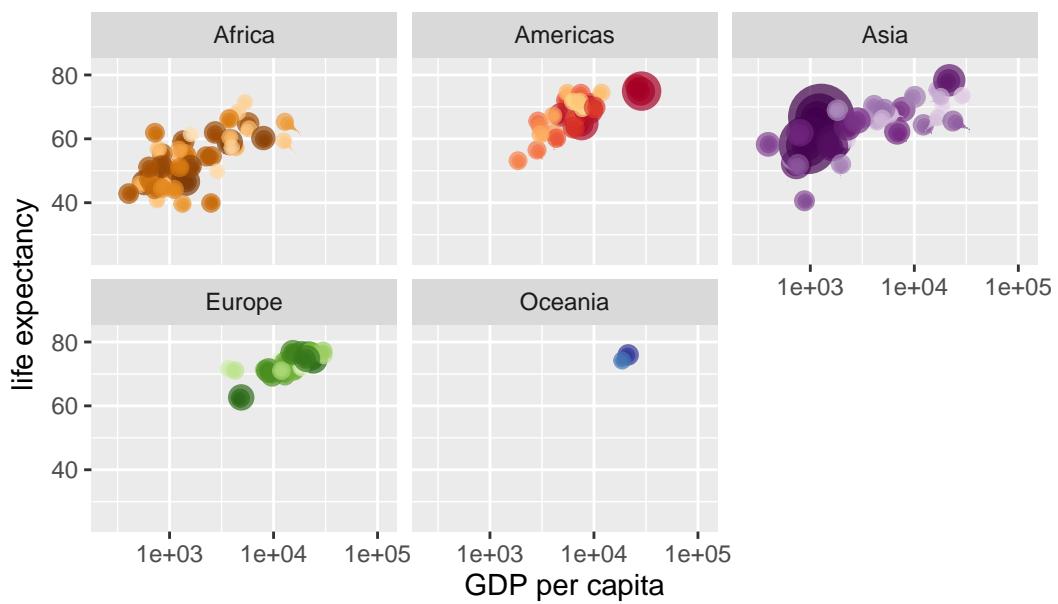
Year: 1985



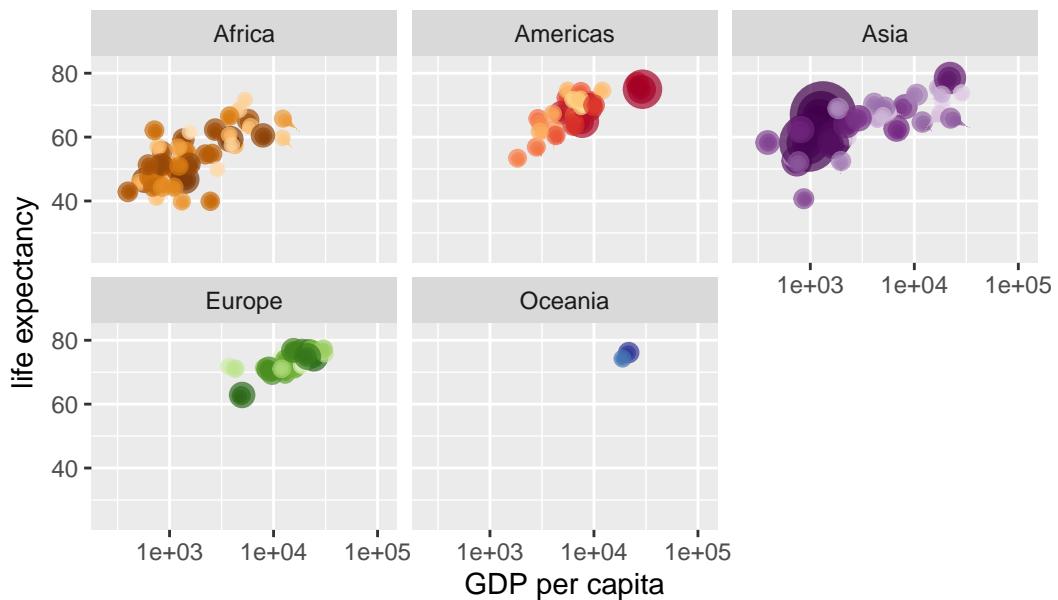
Year: 1985



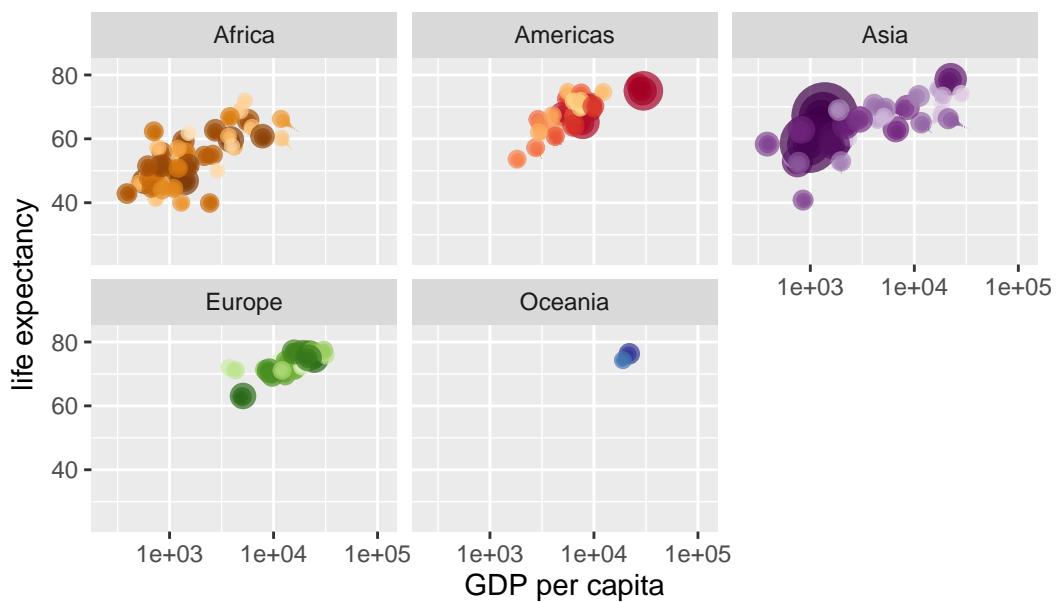
Year: 1986



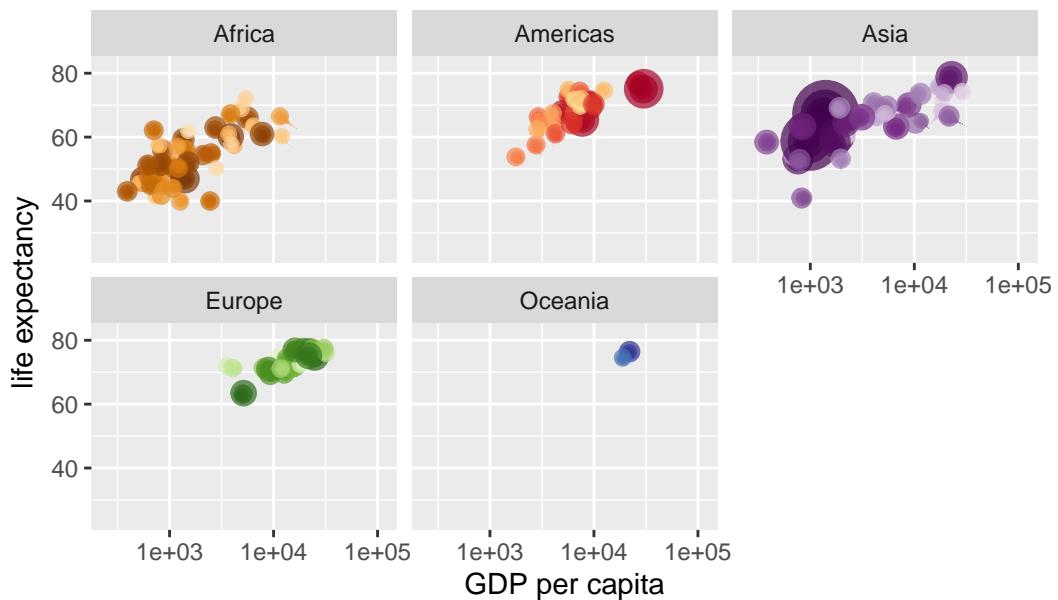
Year: 1986



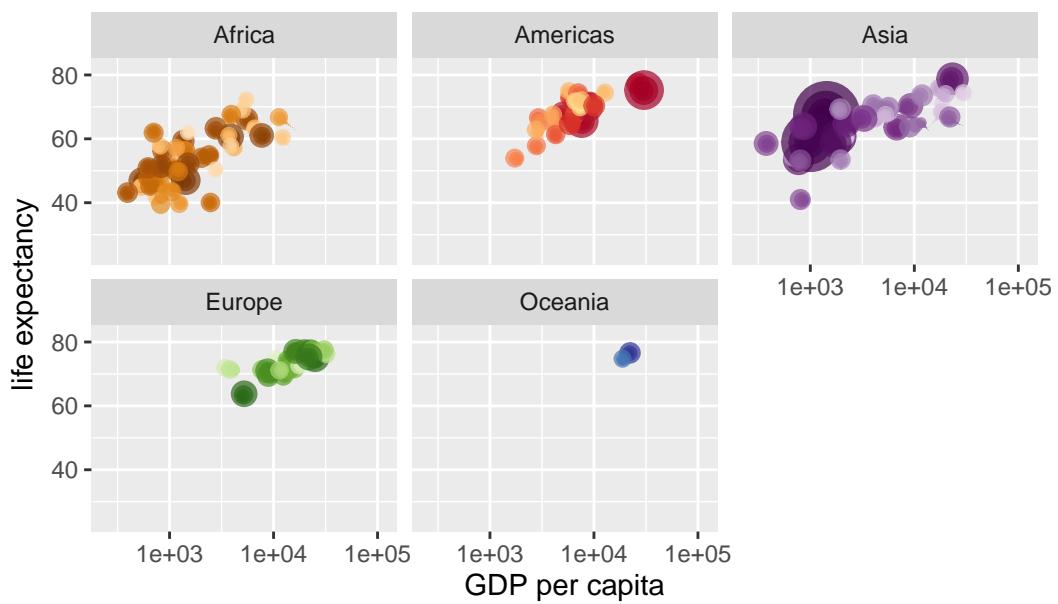
Year: 1987



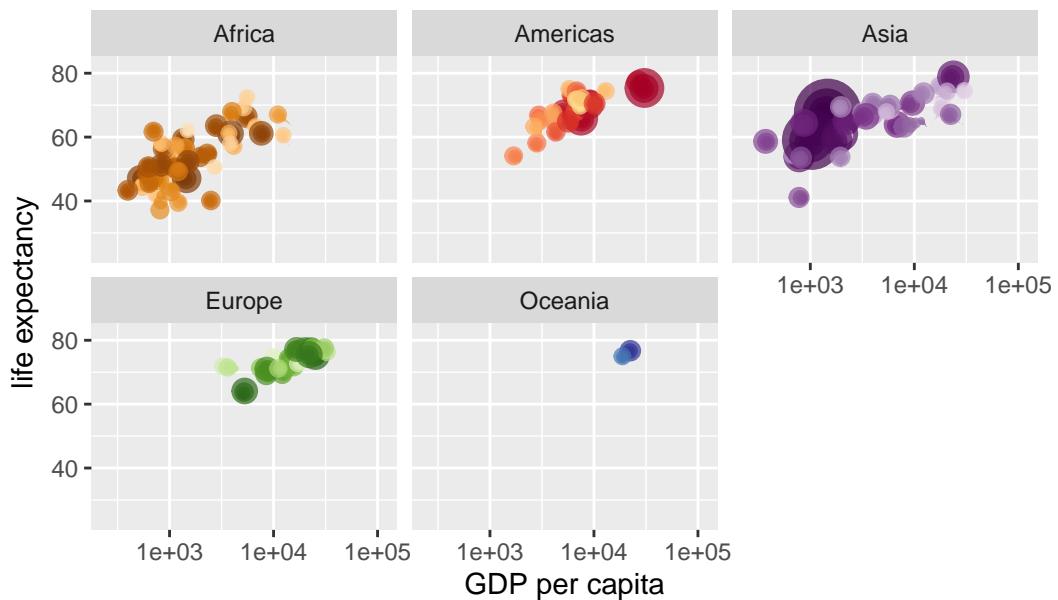
Year: 1988



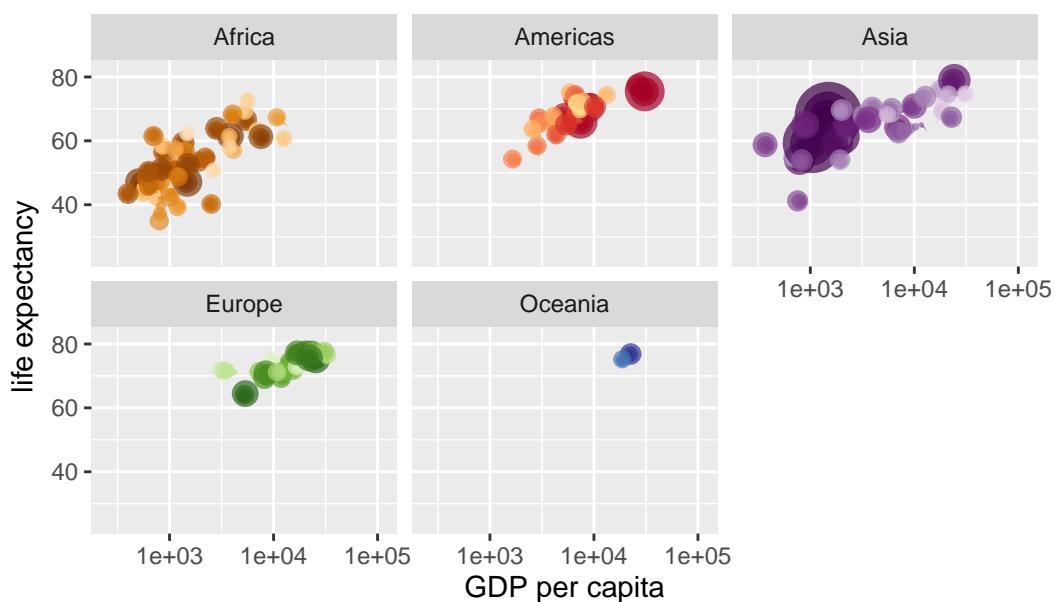
Year: 1988



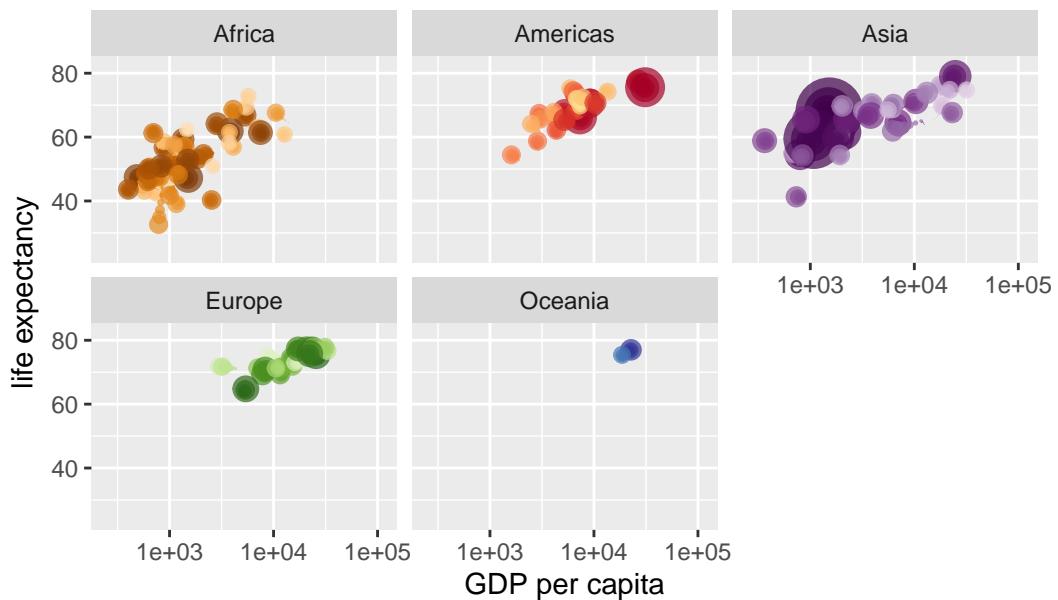
Year: 1989



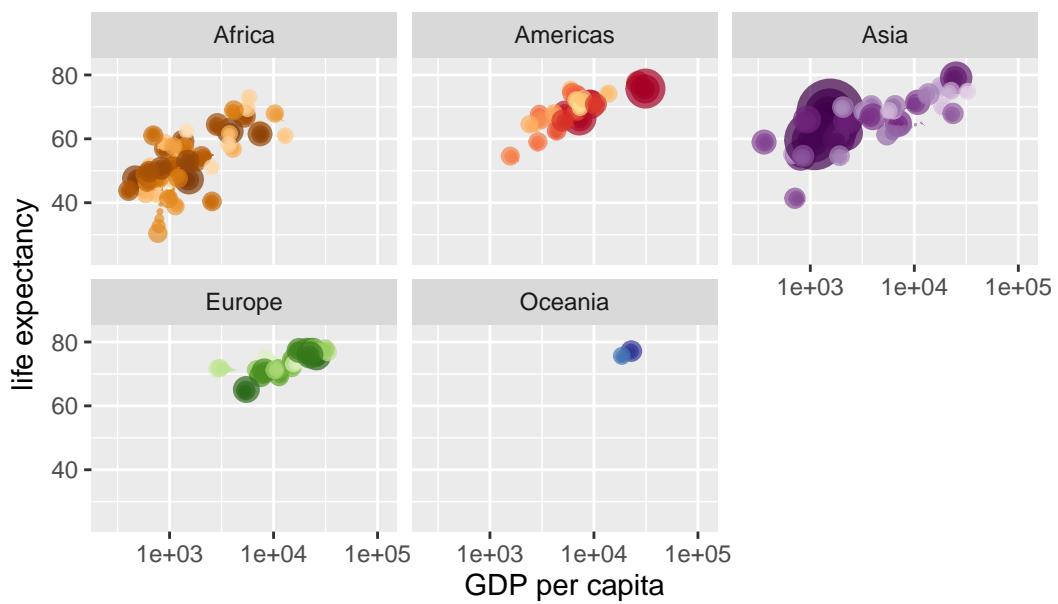
Year: 1989



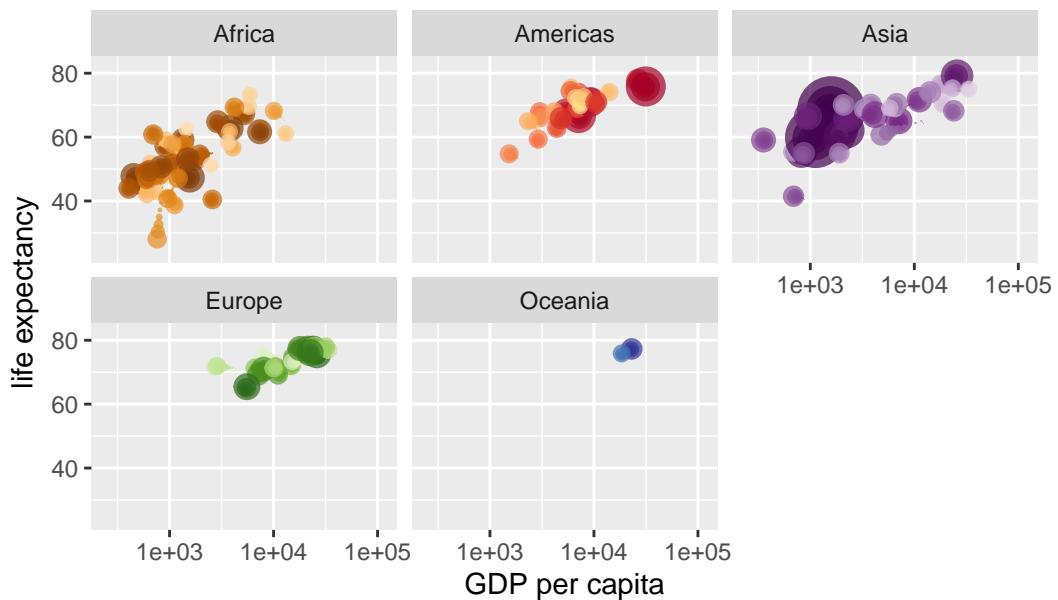
Year: 1990



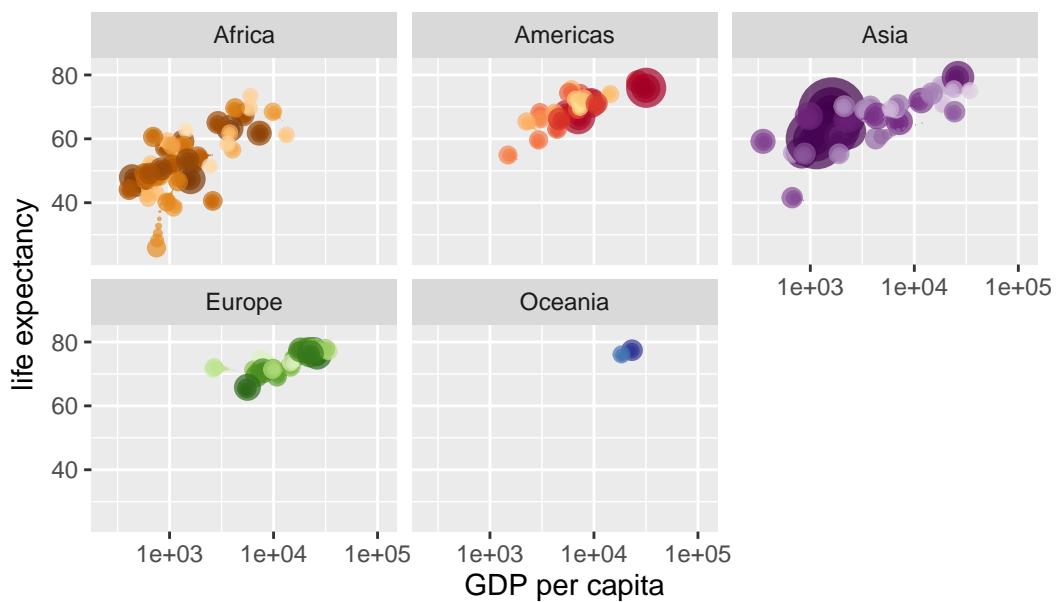
Year: 1990



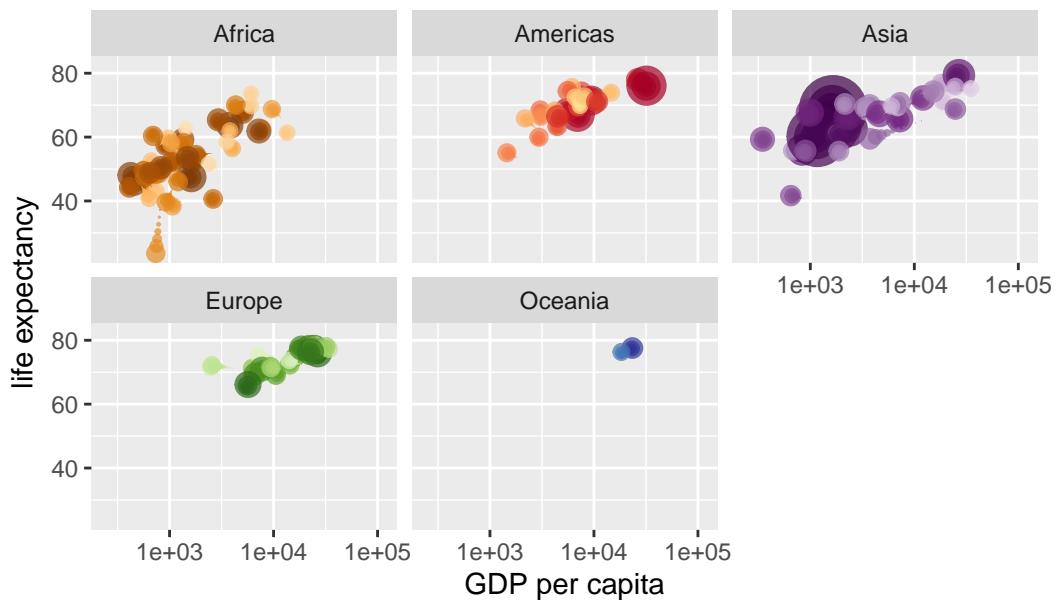
Year: 1991



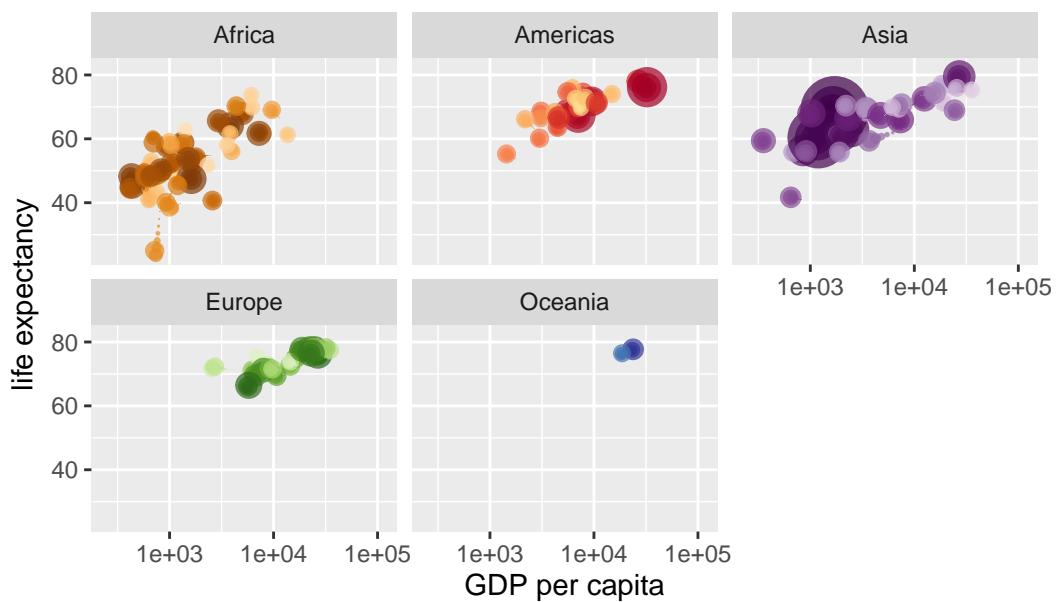
Year: 1991



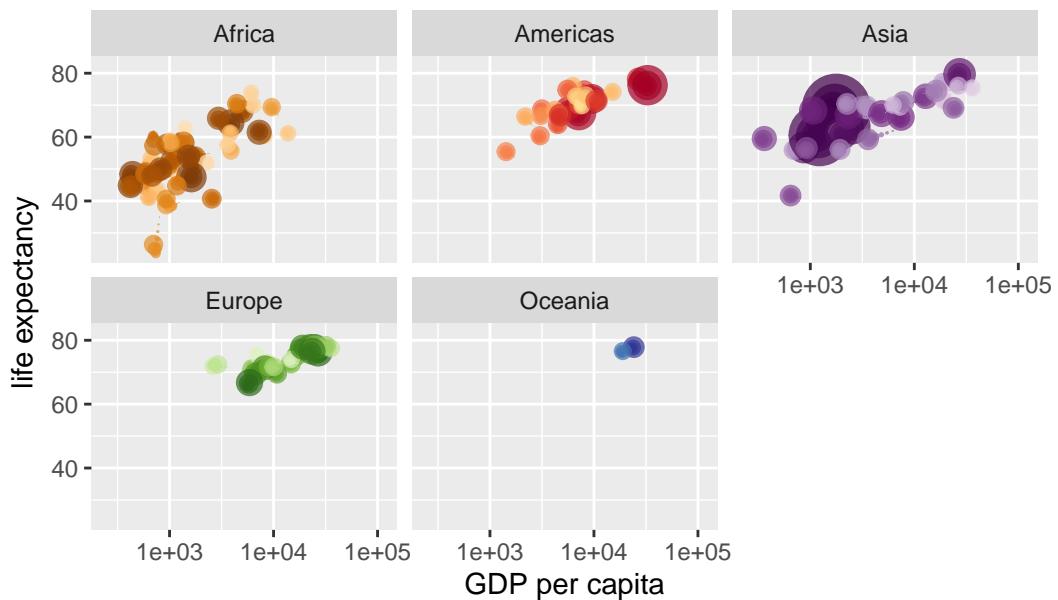
Year: 1992



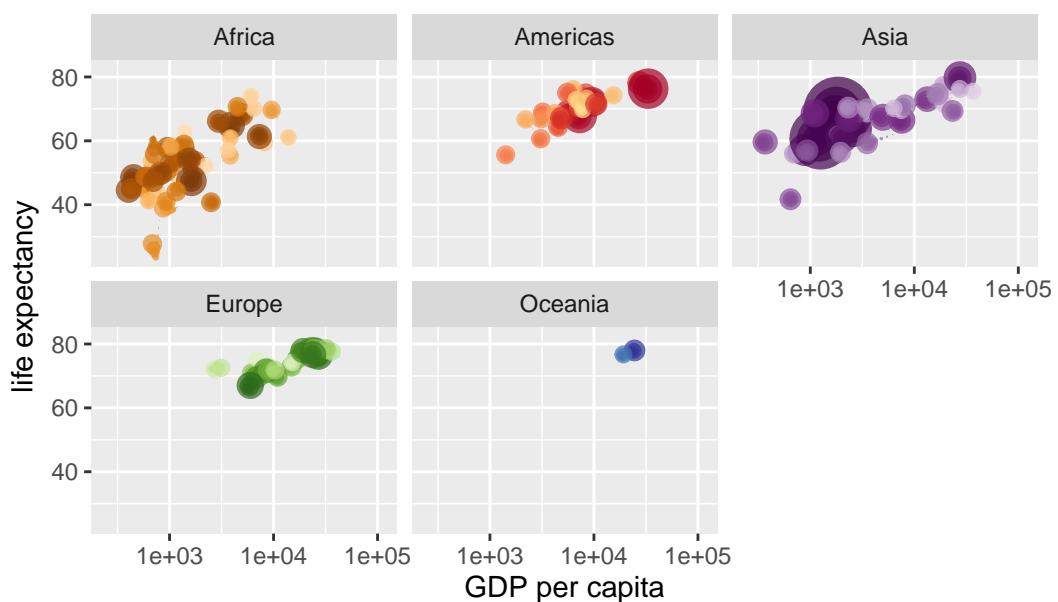
Year: 1993



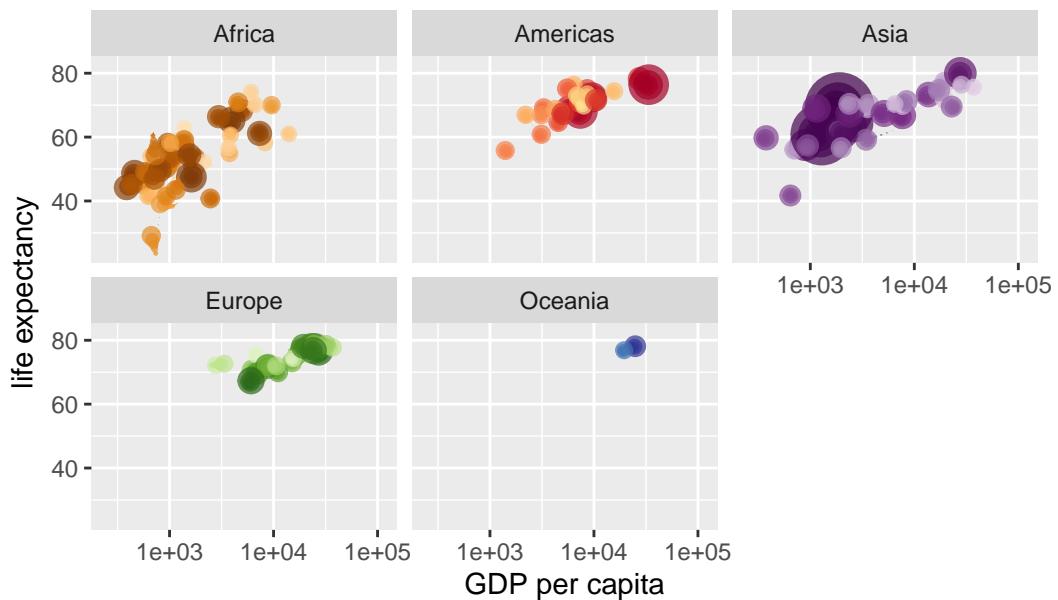
Year: 1993



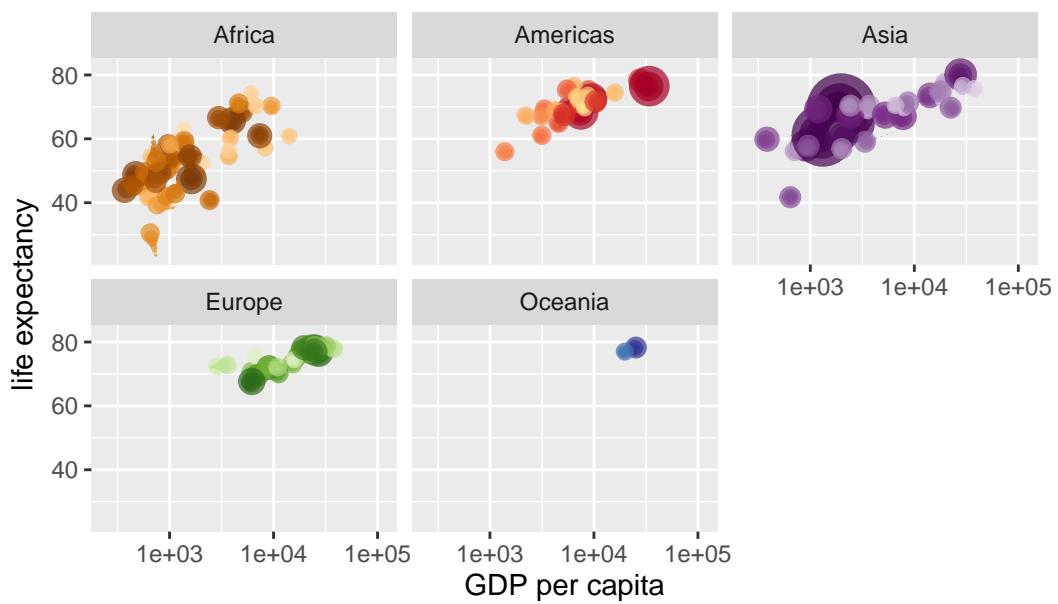
Year: 1994



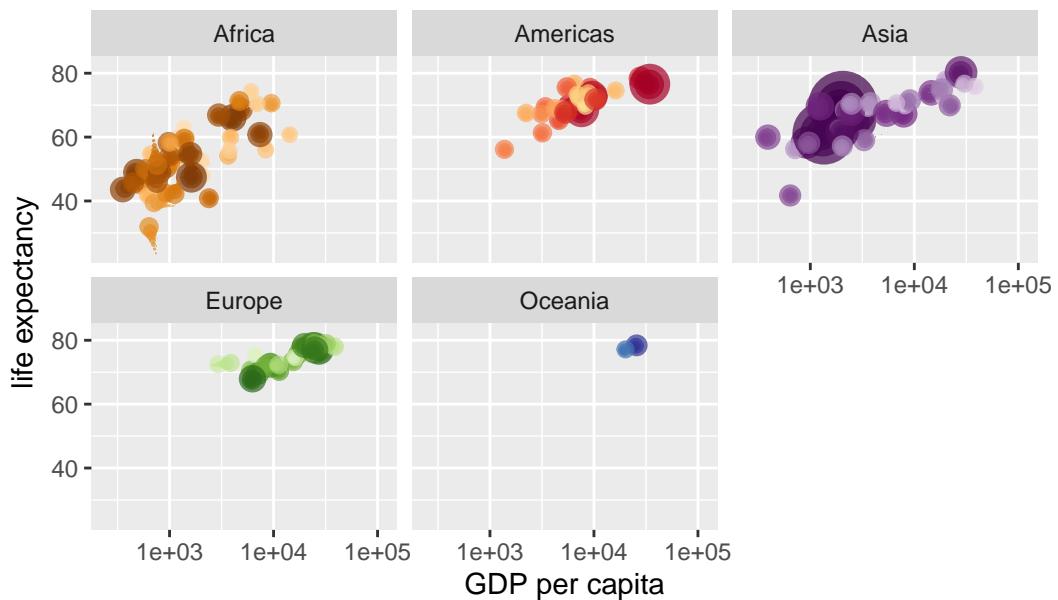
Year: 1994



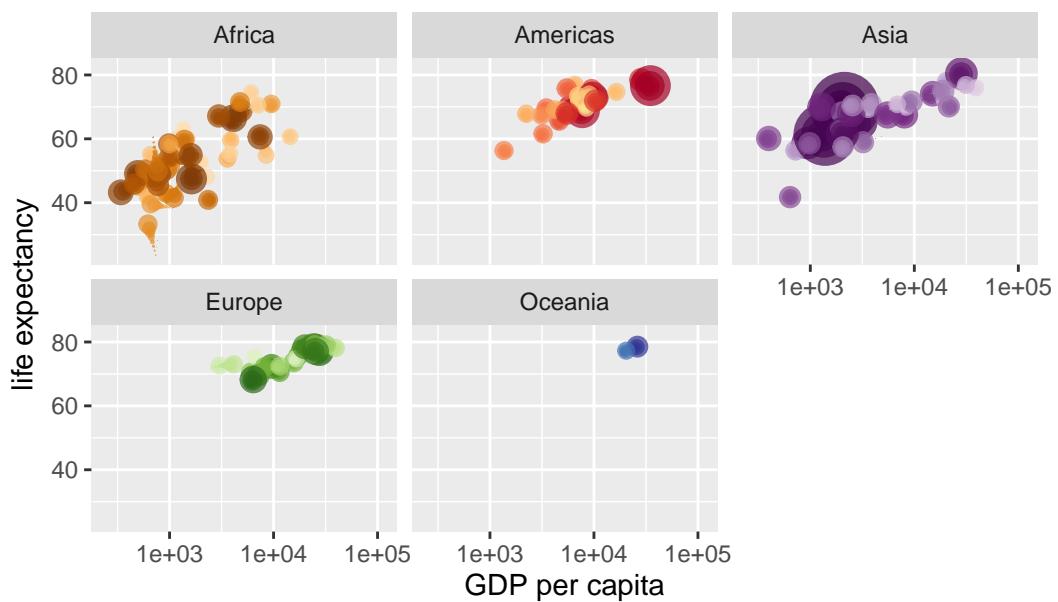
Year: 1995



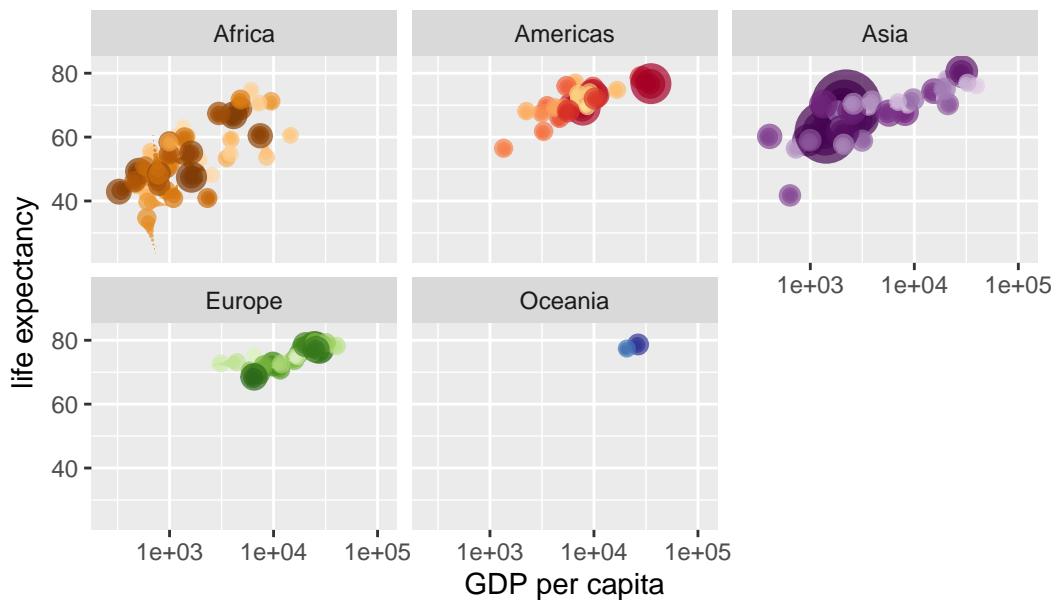
Year: 1995



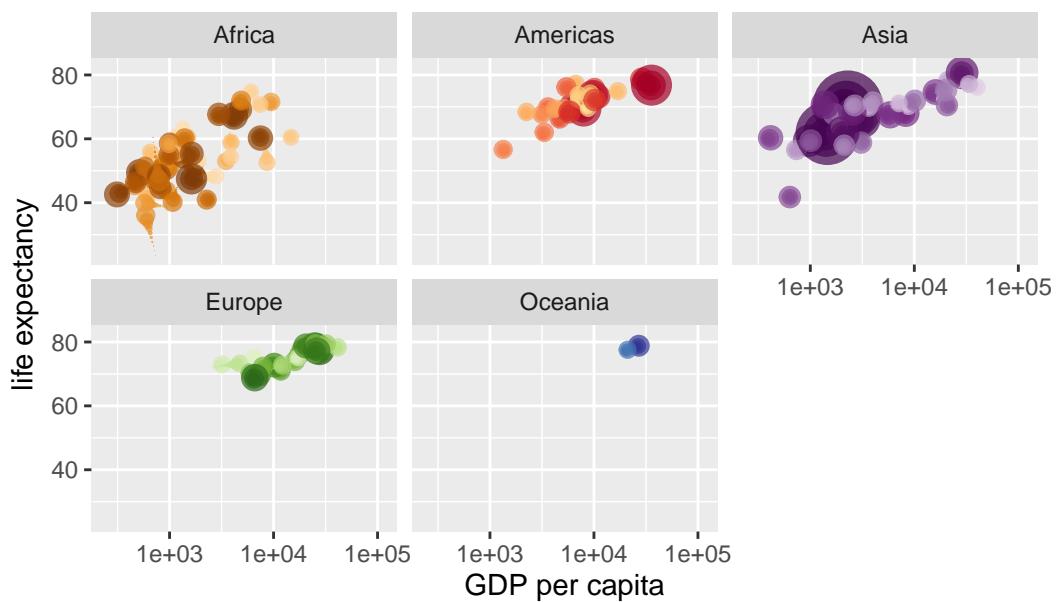
Year: 1996



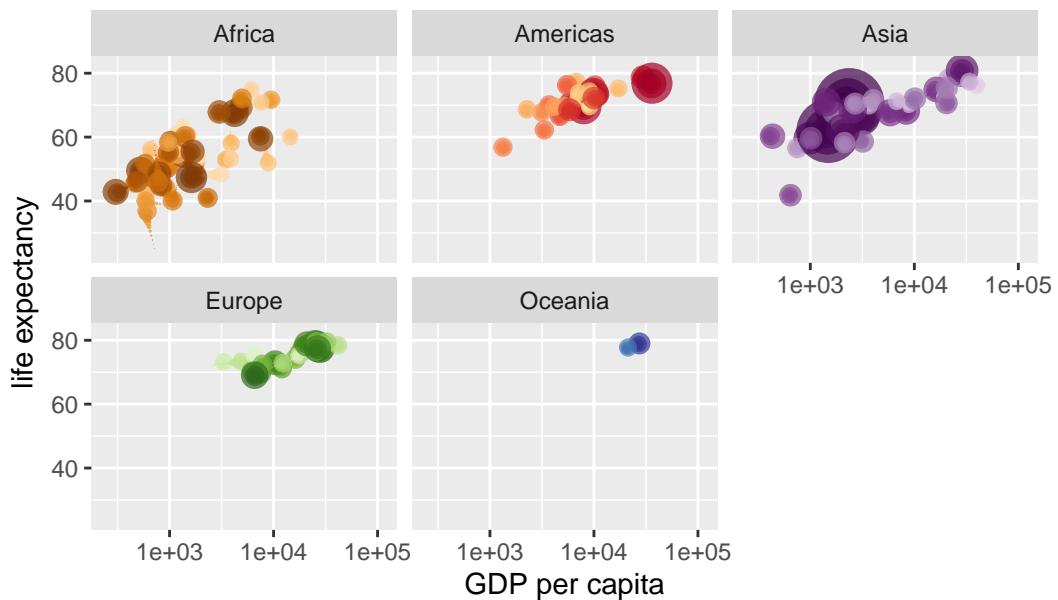
Year: 1996



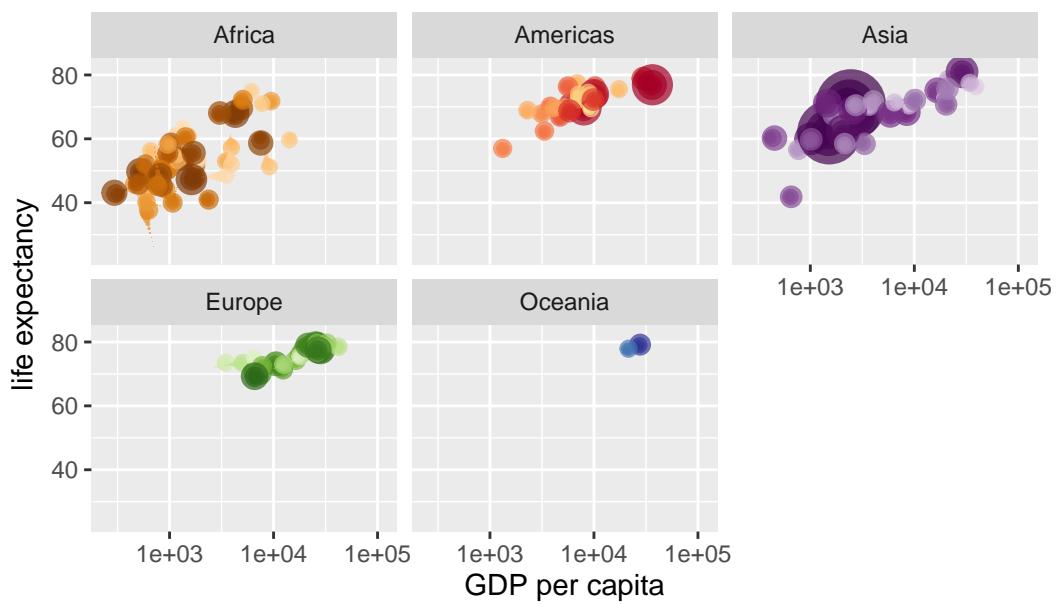
Year: 1997



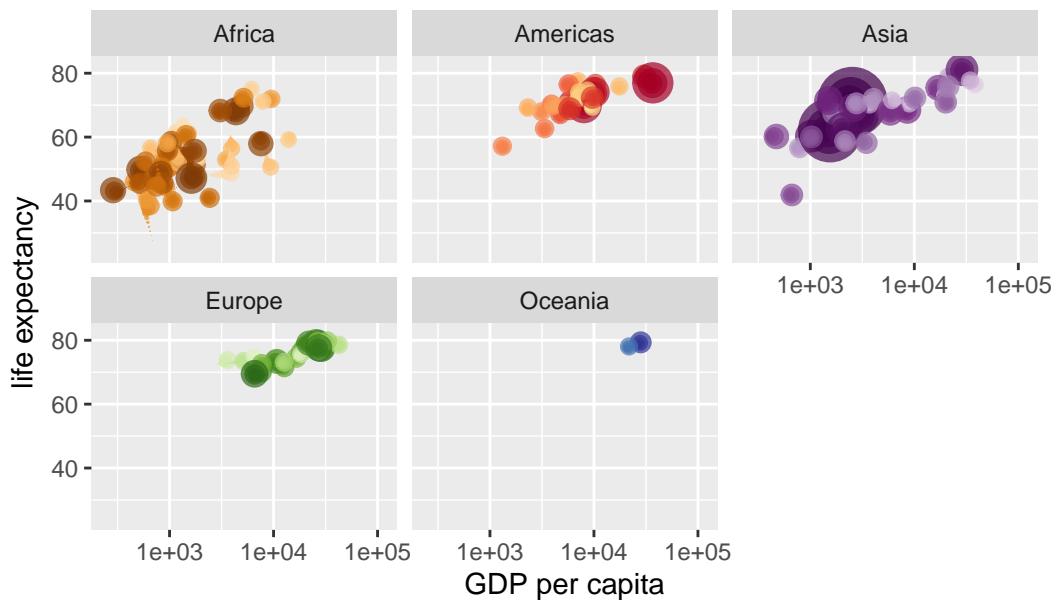
Year: 1998



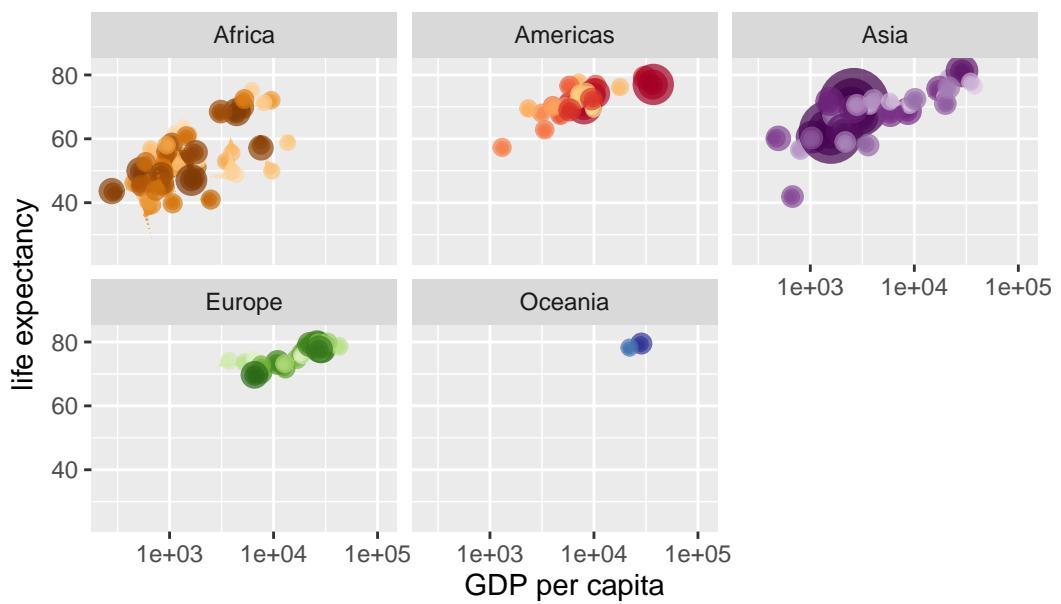
Year: 1998



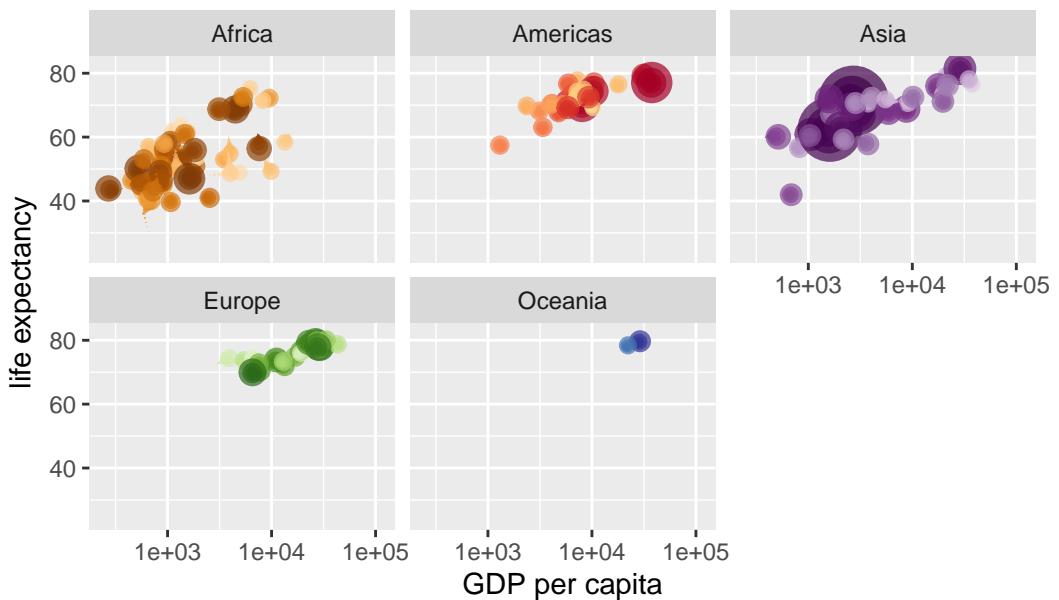
Year: 1999



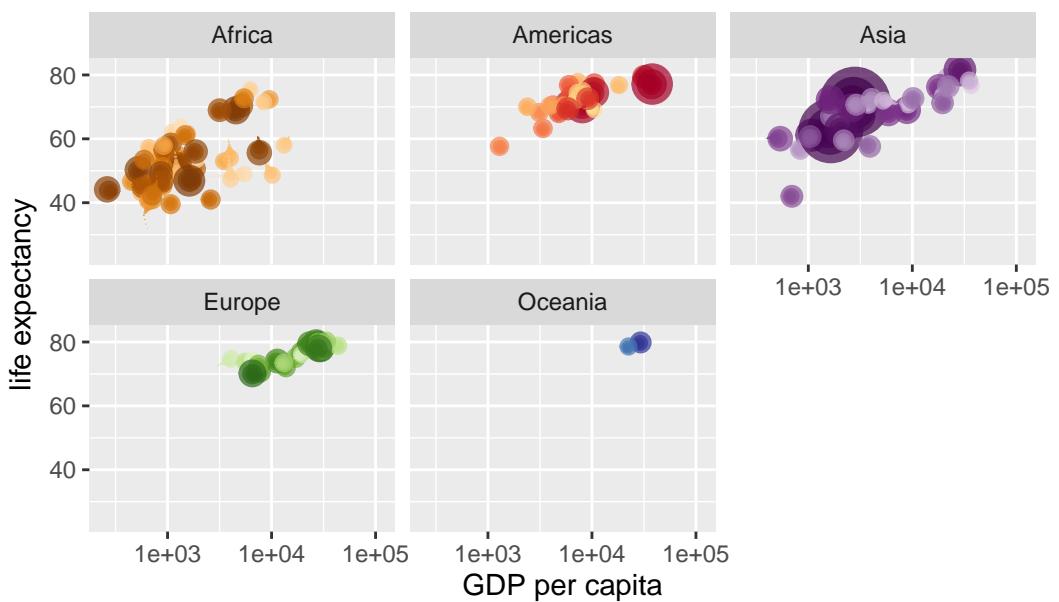
Year: 1999



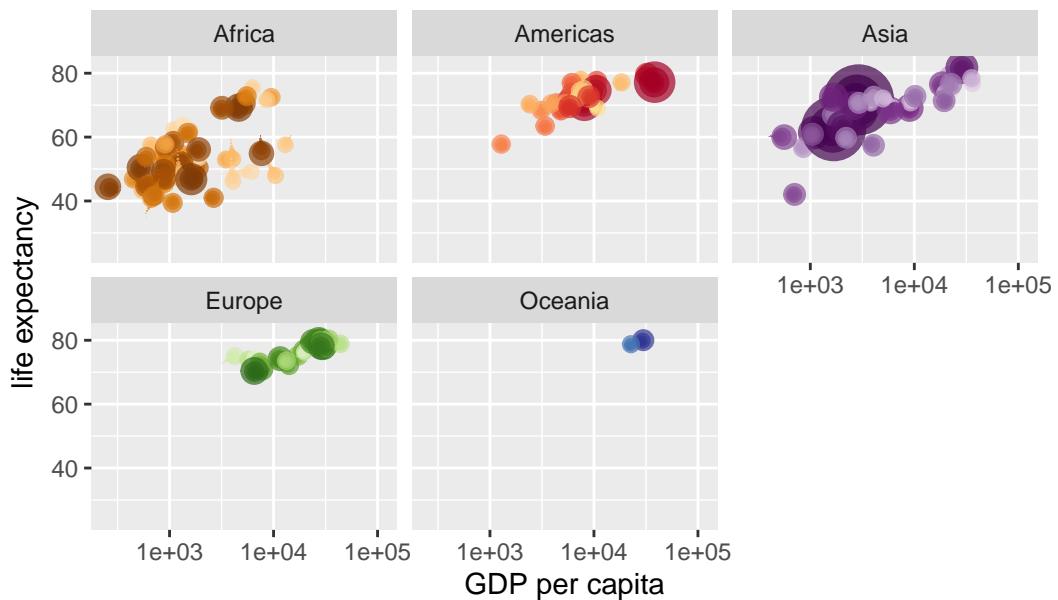
Year: 2000



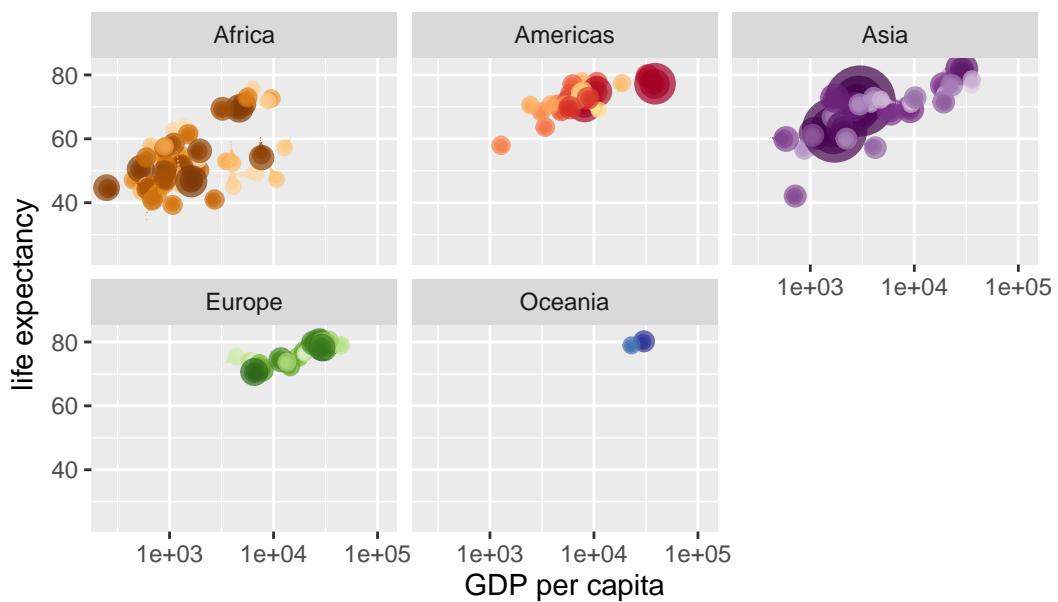
Year: 2000



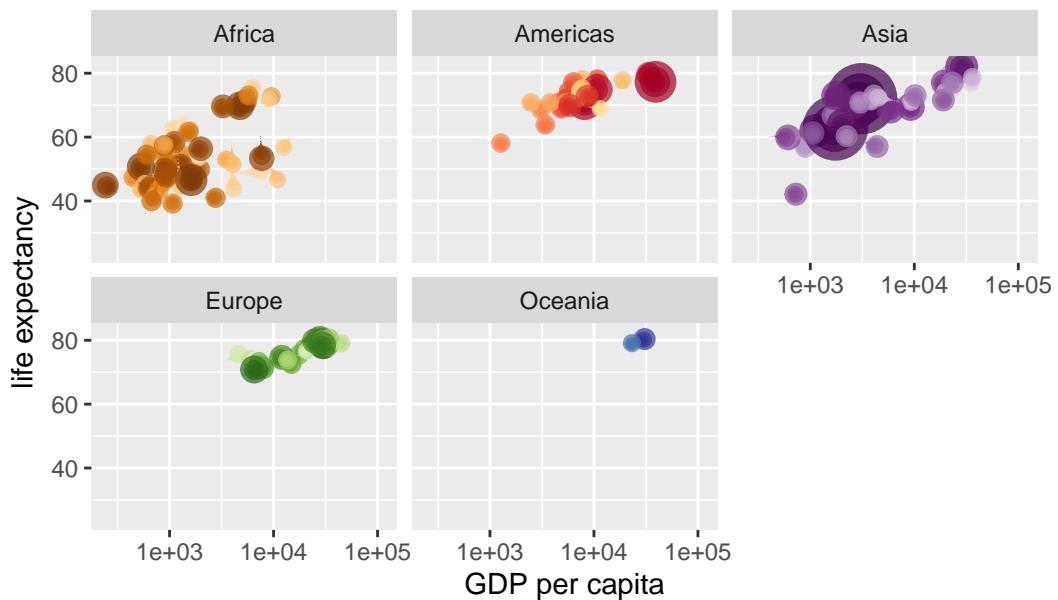
Year: 2001



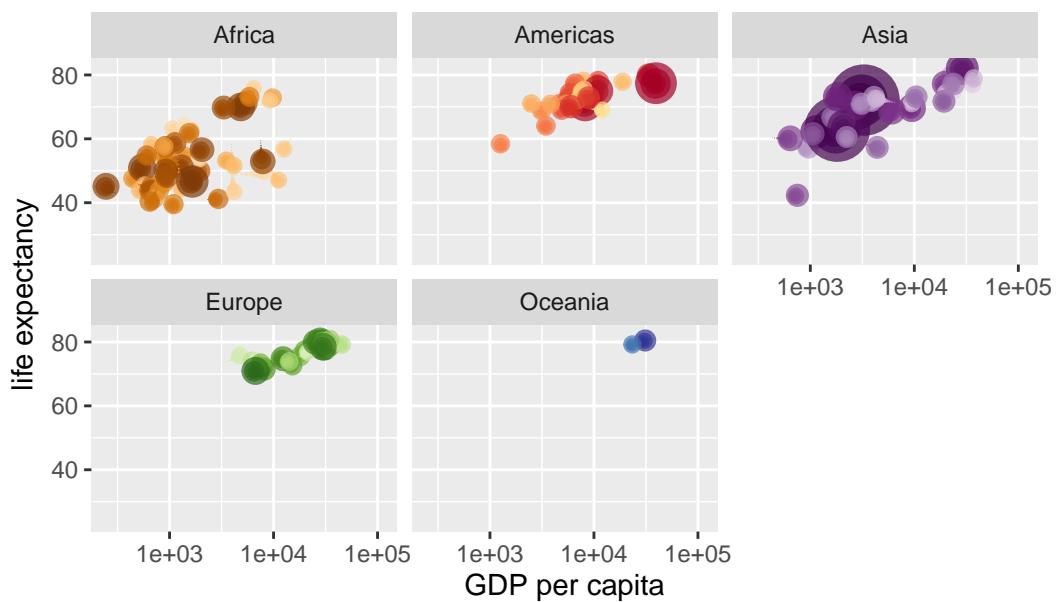
Year: 2001



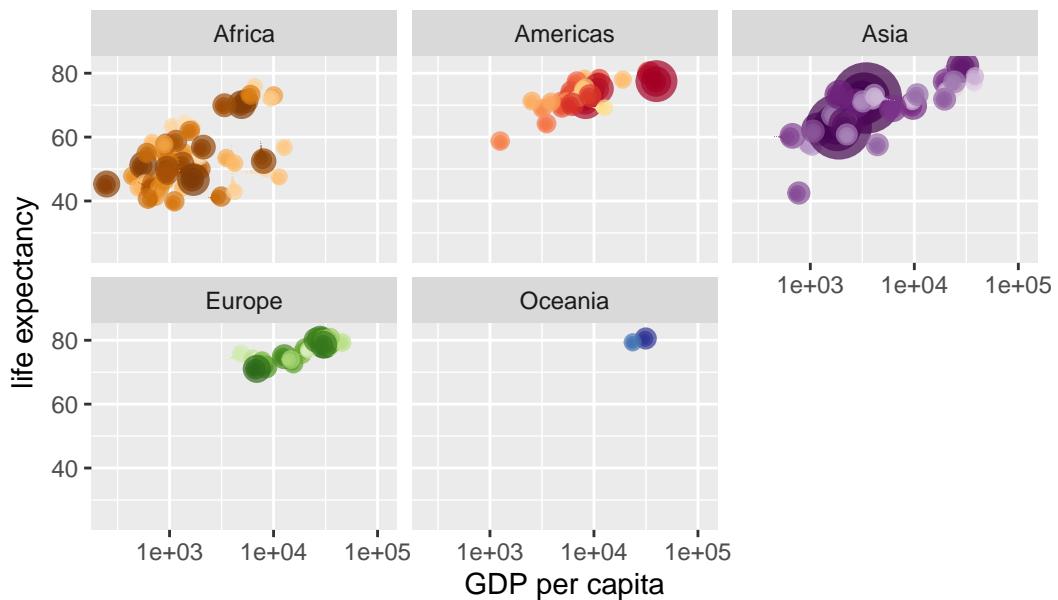
Year: 2002



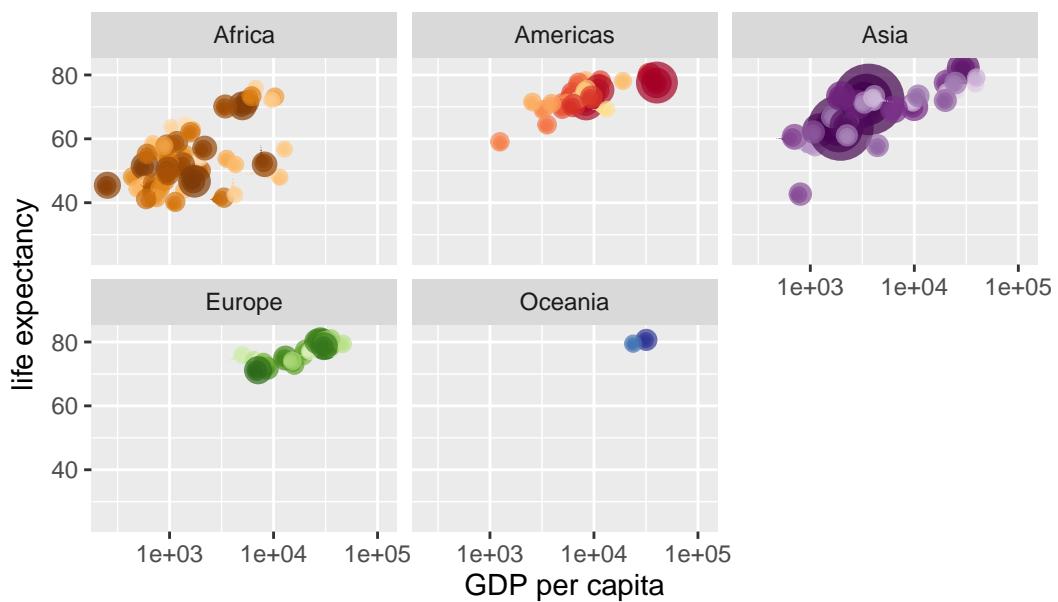
Year: 2003



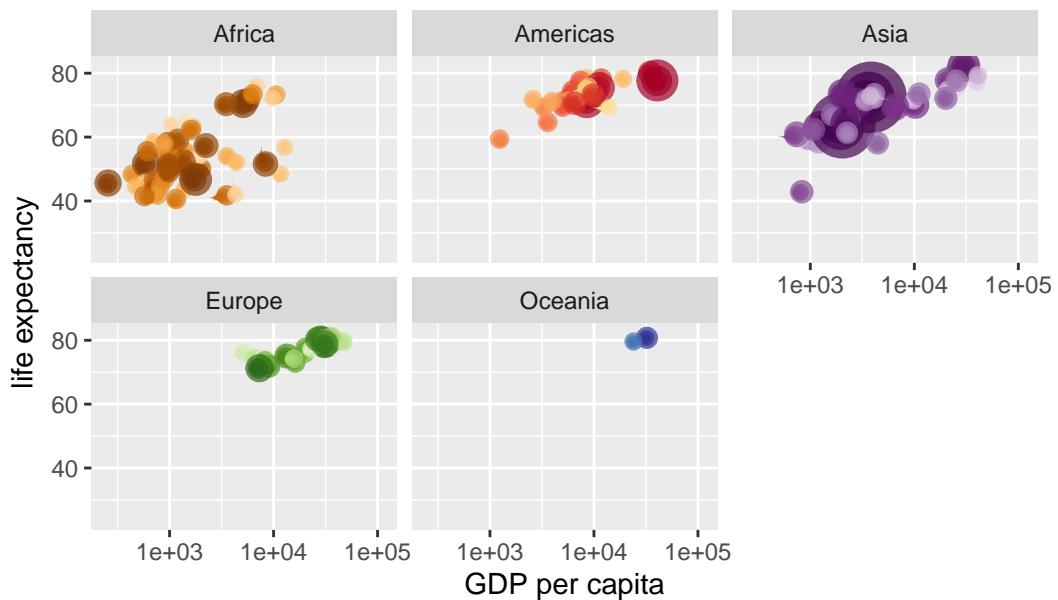
Year: 2003



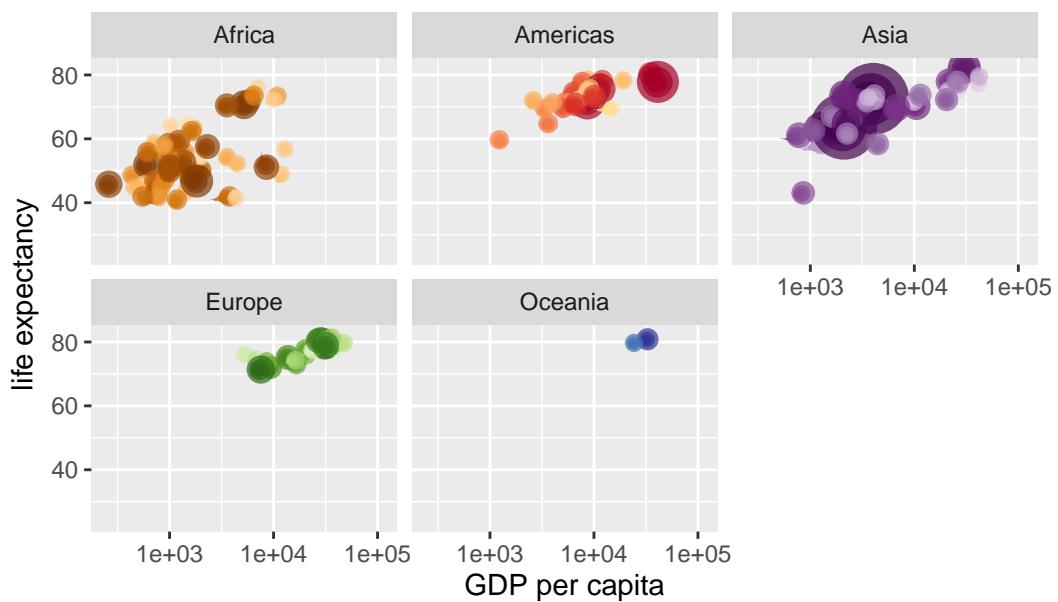
Year: 2004



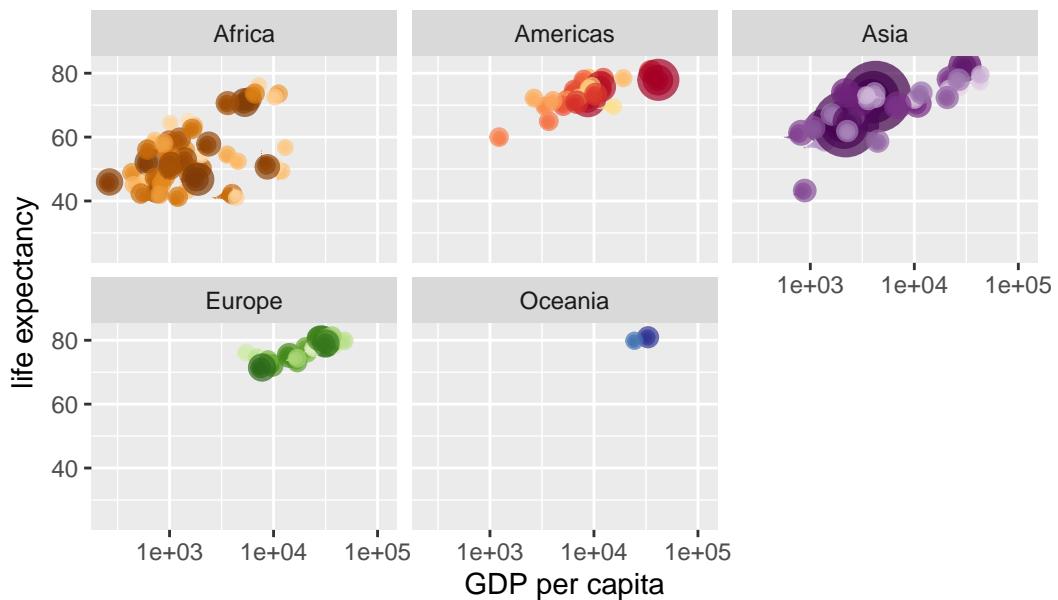
Year: 2004



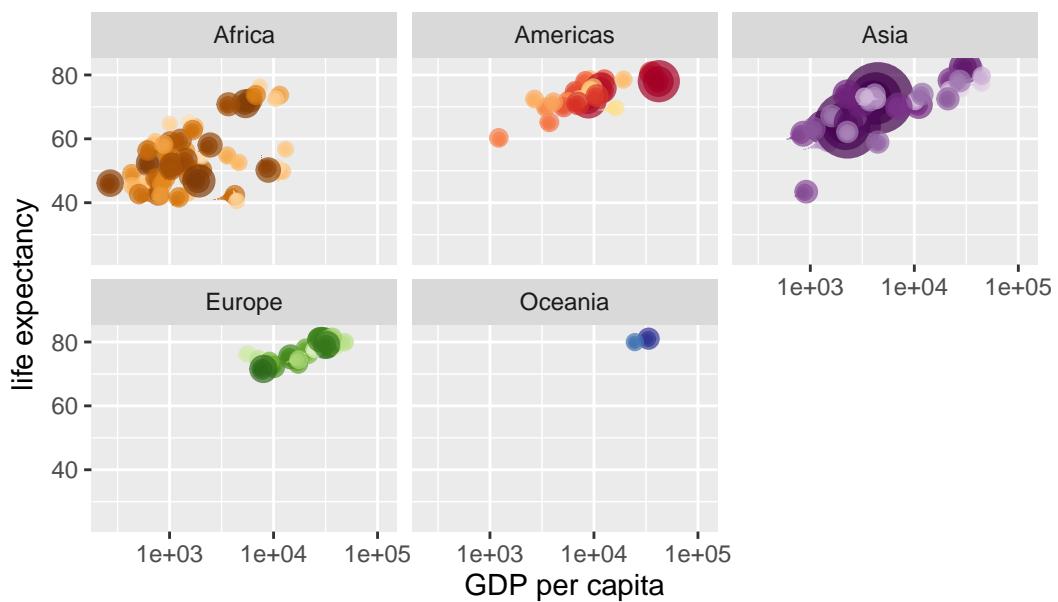
Year: 2005



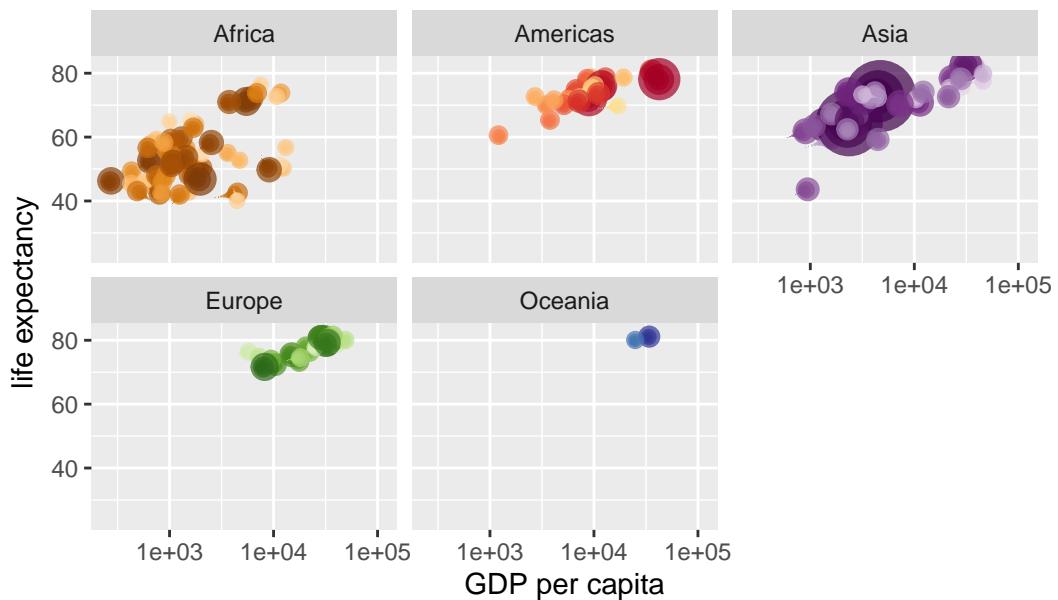
Year: 2005



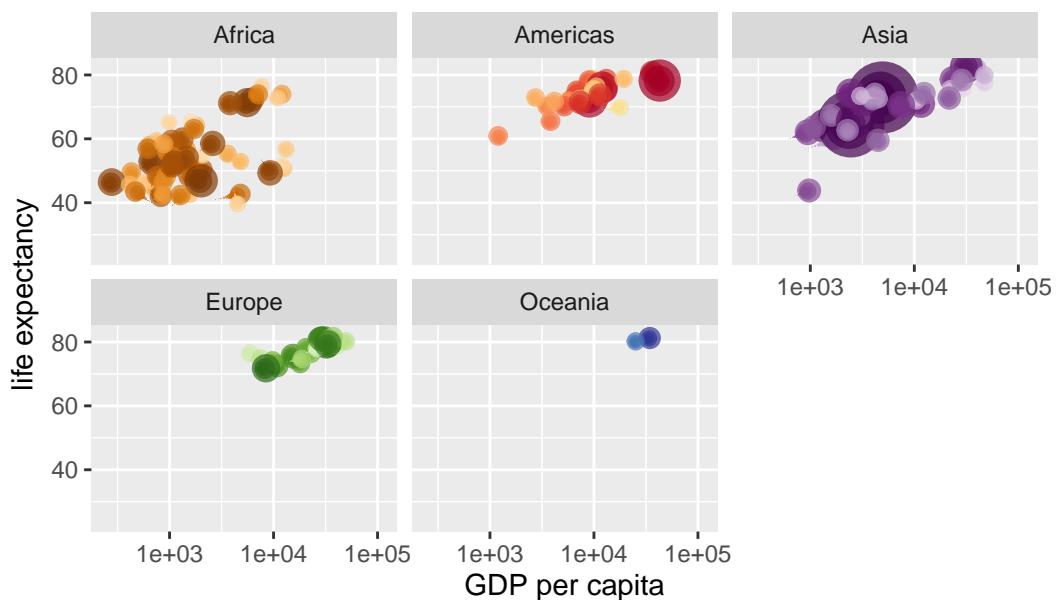
Year: 2006



Year: 2006



Year: 2007



Bar Charts

Find how many years there are in the data set:

```
length(unique(gapminder$year))
```

```
[1] 12
```

Q. Extract data for US in 1992

```
filter(gapminder, country == "United States", year == 1992)
```

```
# A tibble: 1 x 6
  country      continent  year lifeExp      pop gdpPercap
  <fct>        <fct>     <int>   <dbl>     <int>     <dbl>
1 United States Americas  1992     76.1 256894189     32004.
```

Q. Population of Ireland in the last year?

```
filter(gapminder, country == "Ireland", year == 2007)
```

```
# A tibble: 1 x 6
  country continent  year lifeExp      pop gdpPercap
  <fct>    <fct>     <int>   <dbl>     <int>     <dbl>
1 Ireland Europe     2007     78.9 4109086     40676.
```

Q. What countries have a pop smaller than ireland in 2007?

```
y2007 <- filter(gapminder, year == 2007)
irePop <- filter(y2007, country == "Ireland") ["pop"]
irePop
```

```
# A tibble: 1 x 1
  pop
  <int>
1 4109086
```

```
filter(y2007, pop < 4109086)
```

```
# A tibble: 31 x 6
  country      continent year lifeExp     pop gdpPercap
  <fct>        <fct>    <int>   <dbl>   <int>     <dbl>
1 Albania      Europe     2007    76.4 3600523     5937.
2 Bahrain      Asia       2007    75.6 708573      29796.
3 Botswana     Africa     2007    50.7 1639131     12570.
4 Comoros      Africa     2007    65.2 710960      986.
5 Congo, Rep.  Africa     2007    55.3 3800610     3633.
6 Djibouti     Africa     2007    54.8 496374      2082.
7 Equatorial Guinea Africa     2007    51.6 551201     12154.
8 Gabon        Africa     2007    56.7 1454867     13206.
9 Gambia       Africa     2007    59.4 1688359      753.
10 Guinea-Bissau Africa    2007    46.4 1472041      579.
# i 21 more rows
```