Class 5. Data Viz with ggplot

Ruth Barnes: A16747659

2025-01-21

Background

Q1. For which phases is data visualization important in our scientific workflows? **All of the above**

Q2. True or False: The ggplot2 package comes already installed with R? **FALSE**

Other

Q3. Which plot types are typically NOT used to compare distributions of numeric variables? **Network graphs**

Q4. Which statement about data visualization with ggplot2 is incorrect? **ggplot2 is the only way to create plots in R**

Intro to ggplot

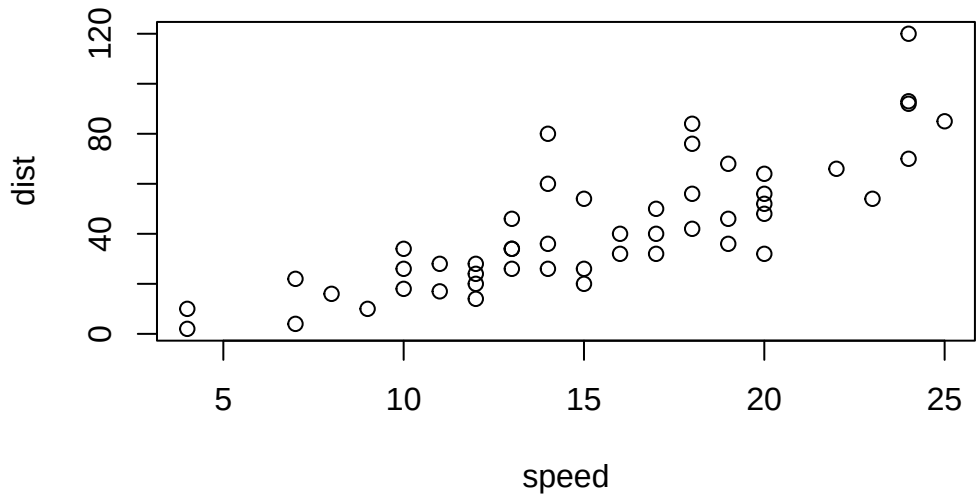
There are may graphics systems in R (ways to make plots and figures). These include "base" R plots. Today we will focus mostly on **ggplot2** package.

Let's start with a plot of a simple in-built dataset called `cars`

```
head(cars)
```

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

```
plot(cars)
```



Let's see how we can make this figure using **ggplot**. First, I need to install this package on my computer. To install any R package I use the function `install.packages()`.

> I will run `install.packages("ggplot2")` in my R console not this quarto document!

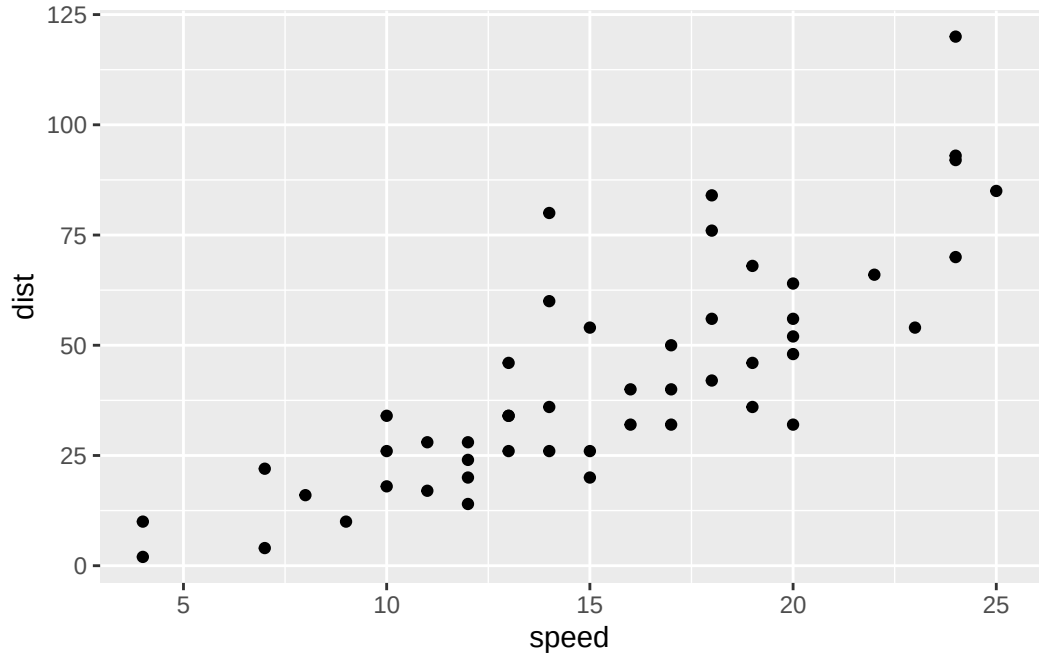
Before I can use any functions from add on packages I need to load the package from my "library()" with the `library(ggplot2)` call.

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

All ggplot figures have at least 3 things (called layers). These include:

- **data** (the input dataset I want to plot from)
- **aes** (the aesthetic mapping of the data to my plot)
- **geoms** (the geom_point(), geom_line(), etc. that I want to draw)

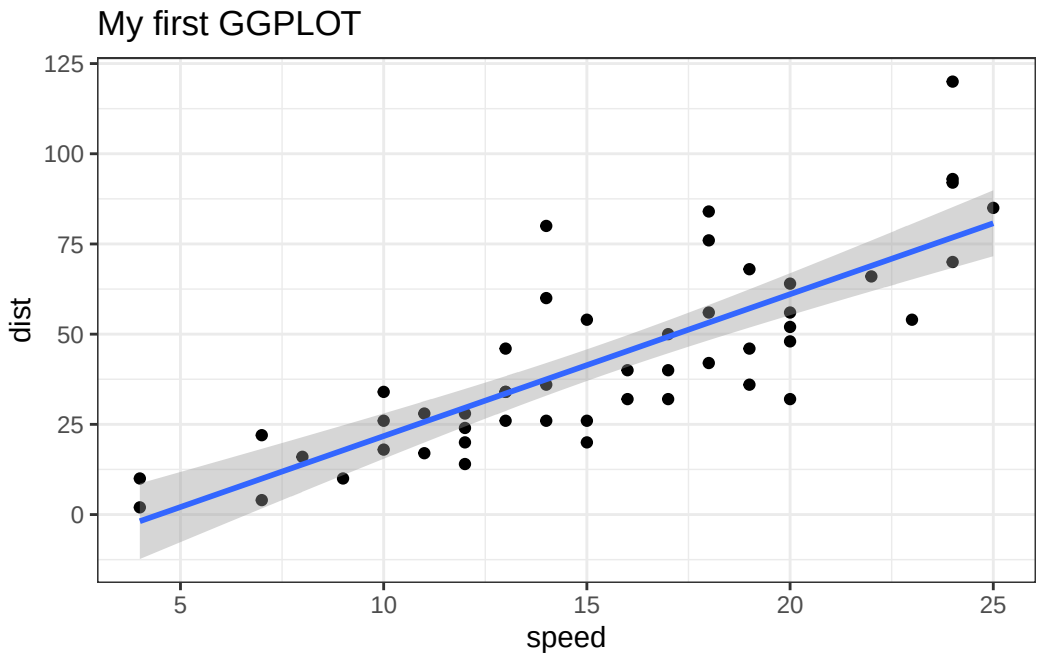
```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



Let's add a line to show the relationship here:

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
 geom_smooth(method="lm") +
  theme_bw() +
  labs(title="My first GGPLOT")
```

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



Q5. Which geometric layer should be used to create scatter plots in ggplot2? **geom_point()**

Gene Expression Figure

The code to read the dataset:

```
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
```

Q6. How many genes are in this dataset?

```
nrow(genes)
```

```
[1] 5196
```

Q7. How many columns are there and what are their names?

```
ncol(genes)
```

```
[1] 4
```

```
colnames(genes)
```

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

Q8. 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)
```

```
      down unchanging         up
        72       4997        127
```

Q9. 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), 4)
```

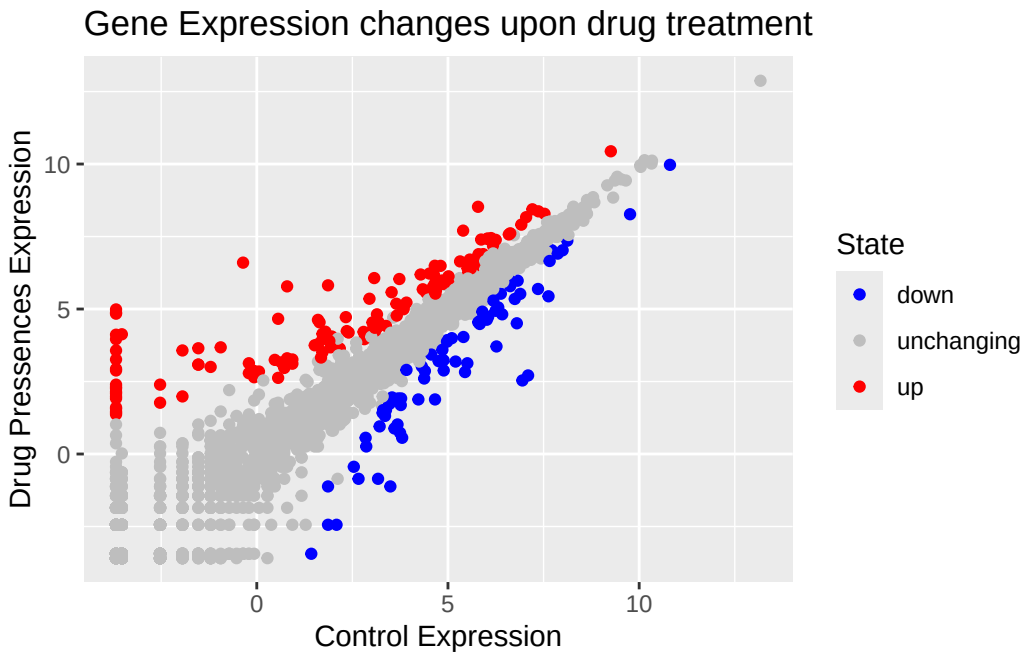
```
      down unchanging         up
    0.0139     0.9617     0.0244
```

```
n.tot <- nrow(genes)
vals <- table(genes$State)

vals.percent <- vals/n.tot * 100
round(vals.percent, 2)
```

```
      down unchanging         up
      1.39      96.17       2.44
```

A First Plot of this Dataset

```
ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
   geom_point() +
  labs(title= "Gene Expression changes upon drug treatment", x="Control Expression", y="Drug
scale_color_manual(values=c("blue", "gray", "red"))
```



Going Further

Exploring the gapmider dataset: The gapminder dataset contains economic and demographic data about various countries since 1952.

```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.tsv

gapminder <- read.delim(url)

head(gapminder)
```

```
      country continent year lifeExp      pop gdpPercap
```

```
1 Afghanistan      Asia 1952  28.801  8425333  779.4453
2 Afghanistan      Asia 1957  30.332  9240934  820.8530
3 Afghanistan      Asia 1962  31.997 10267083  853.1007
4 Afghanistan      Asia 1967  34.020 11537966  836.1971
5 Afghanistan      Asia 1972  36.088 13079460  739.9811
6 Afghanistan      Asia 1977  38.438 14880372  786.1134
```

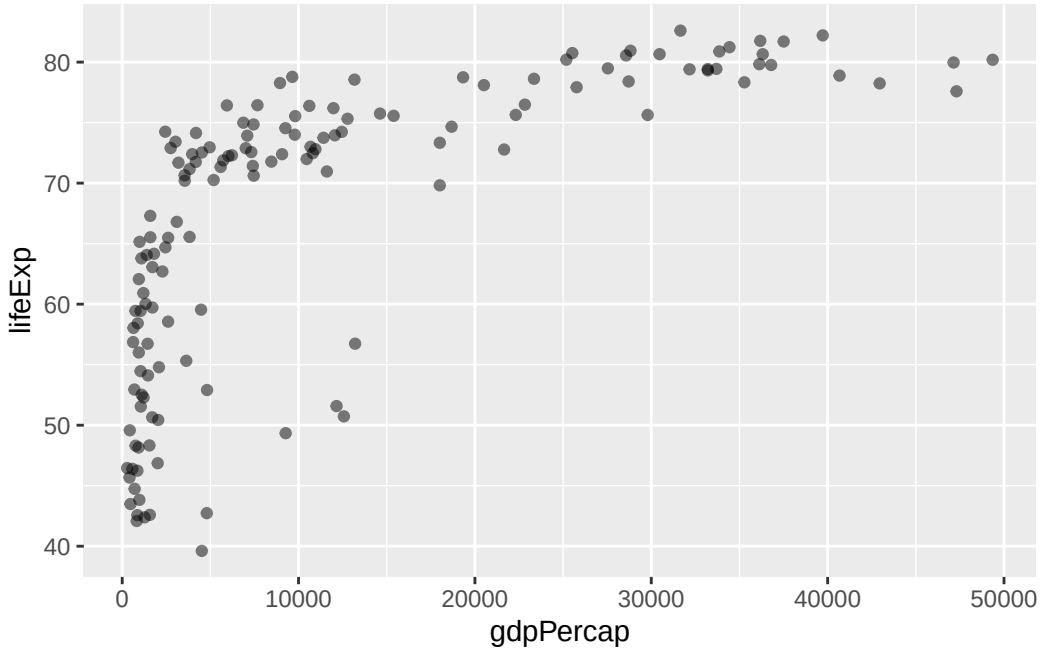
Install.packages("dplyr")

```
library(dplyr)
```

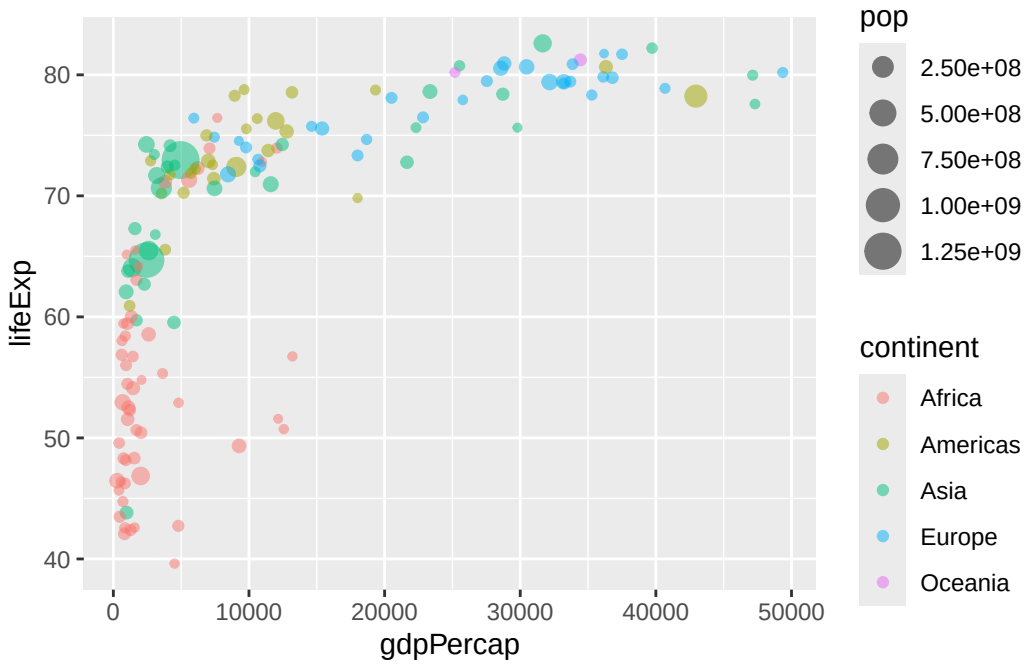
```
Attaching package: 'dplyr'

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

    filter, lag

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

    intersect, setdiff, setequal, union
```

```
gapminder_2007 <- filter(gapminder, year==2007)
```

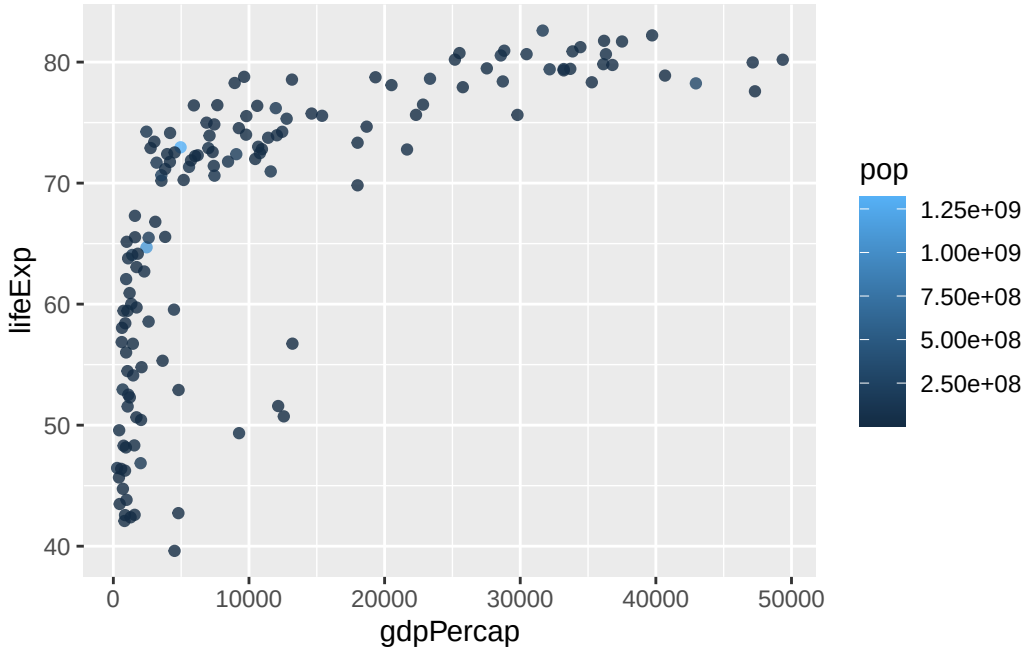
```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```



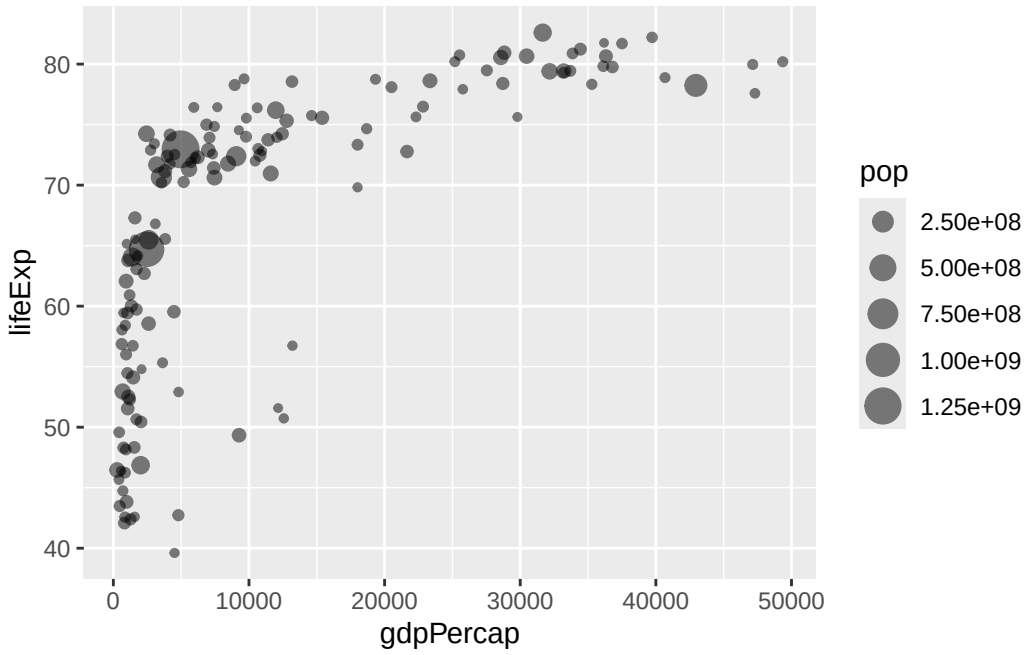

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, size=pop, color=continent) +
  geom_point(alpha=0.5)
```



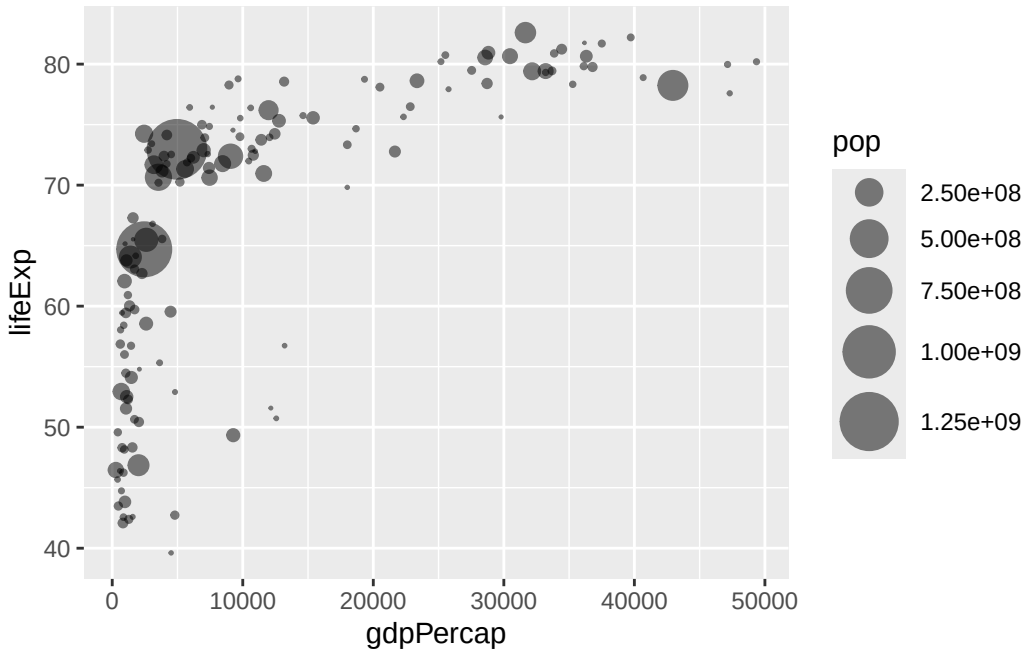
```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



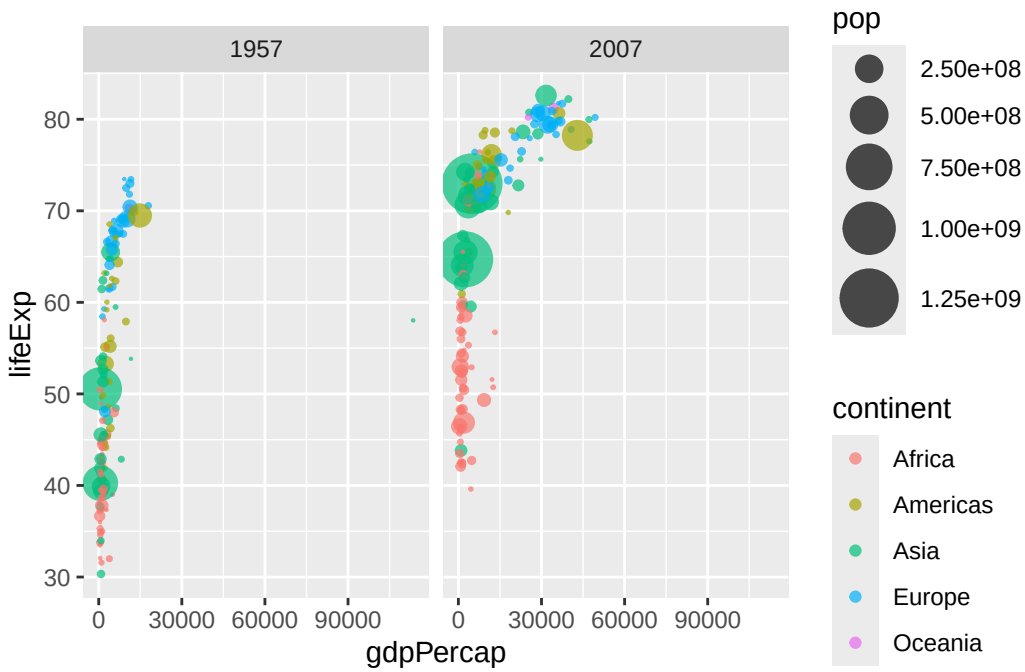

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```




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




```
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 = 10) +
  facet_wrap(~year)
```



Extensions: Animation

Combining Plots