

# Class05: Data Vis with ggplot

Irene (PID: A16197563)

## Graphics systems in R

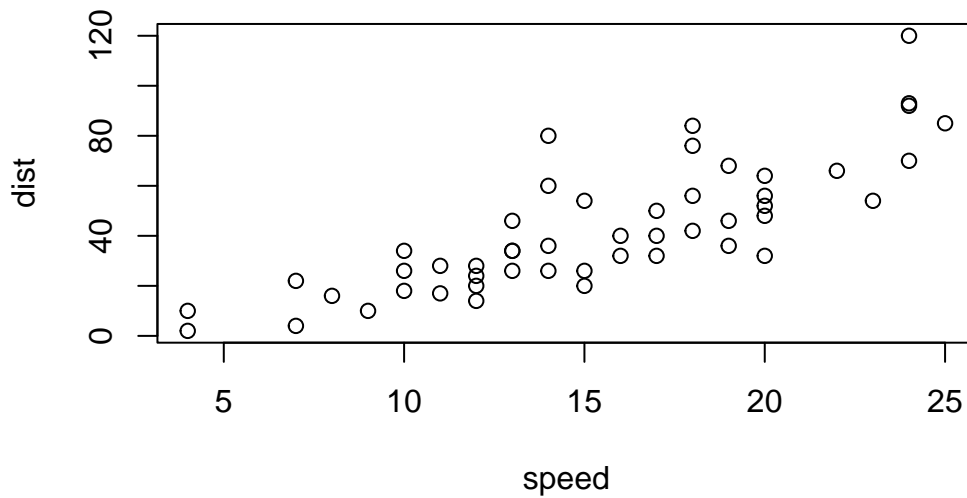
There are many graphic systems in R for making plots and figures.

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()`.

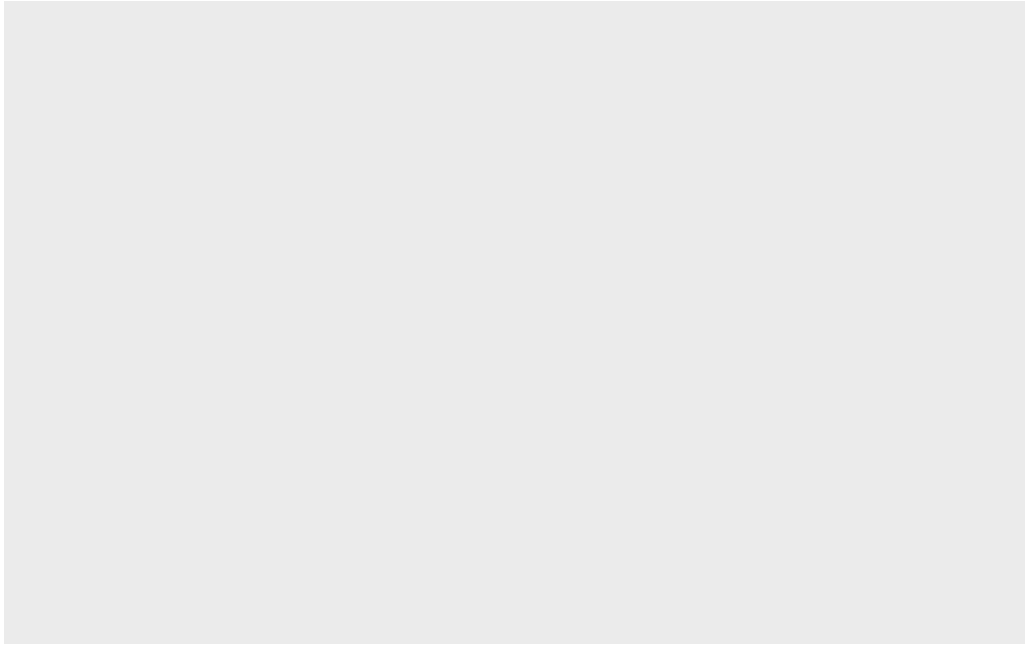
This is an add on package - i.e. we need to install it. (like I install any package) with the `install.packages()` function.

```
plot(cars)
```



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

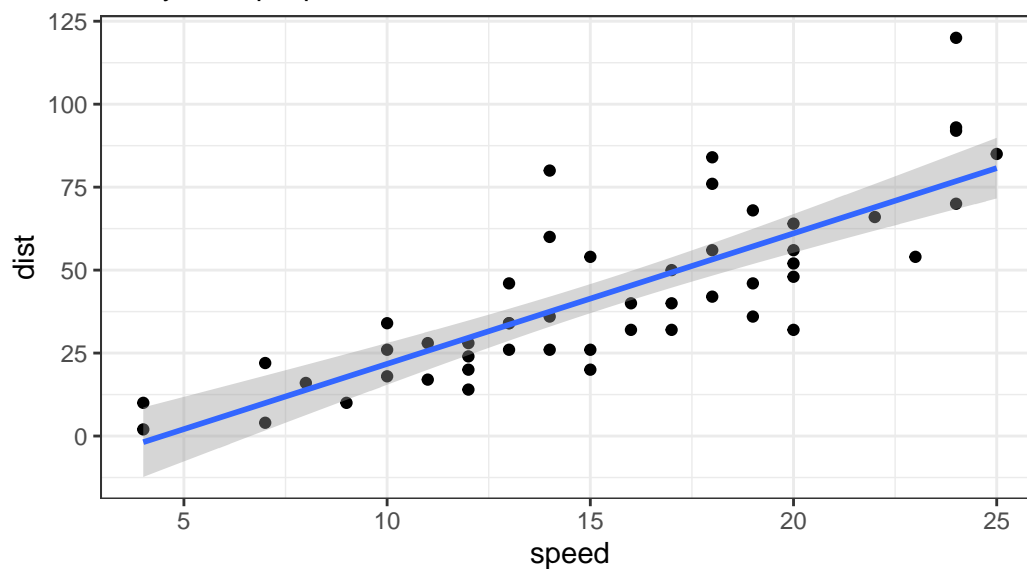
Add some more layers to our ggplot:

```
ggplot(cars) +  
  aes(x=speed, y=dist)+  
  geom_point()+  
  geom_smooth(method = "lm")+  
  labs(title = "stopping distance of old car", subtitle = "A silly example plot")+  
  theme_bw()
```

`geom\_smooth()` using formula = 'y ~ x'

## stopping distance of old car

A silly example plot



```
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. Use the `colnames()` function and the `ncol()` function on the `genes` data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find?

```
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)
```

down	unchanging	up
72	4997	127

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) * 100, 2 )
```

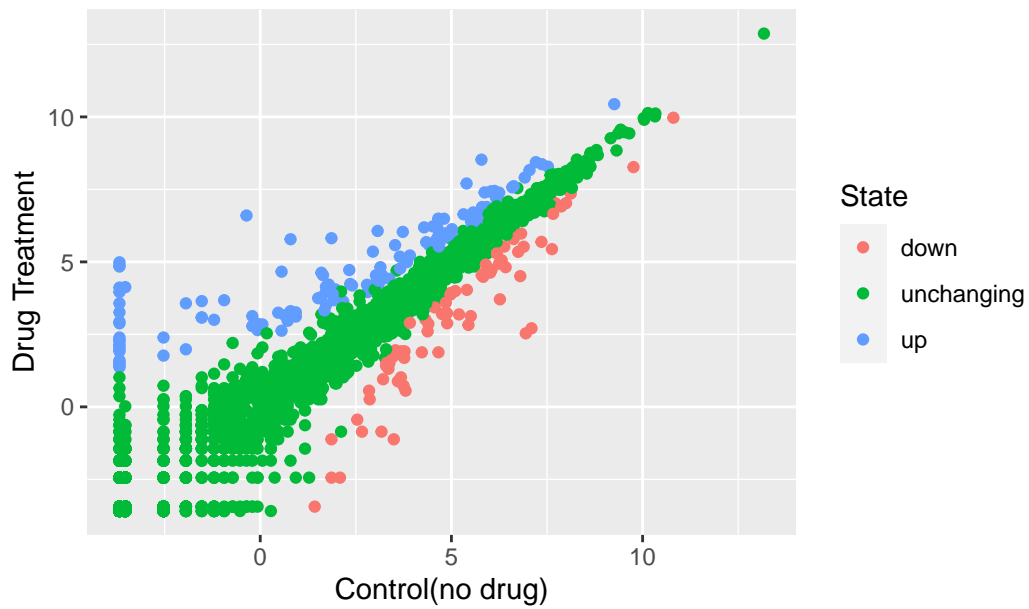
down	unchanging	up
1.39	96.17	2.44

plot:

```
p <- ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col = State) +  
  geom_point()+  
  labs(title = "Gene Expression Changes Upon Drug Treatment",  
        x= "Control(no drug)",  
        y = "Drug Treatment")
```

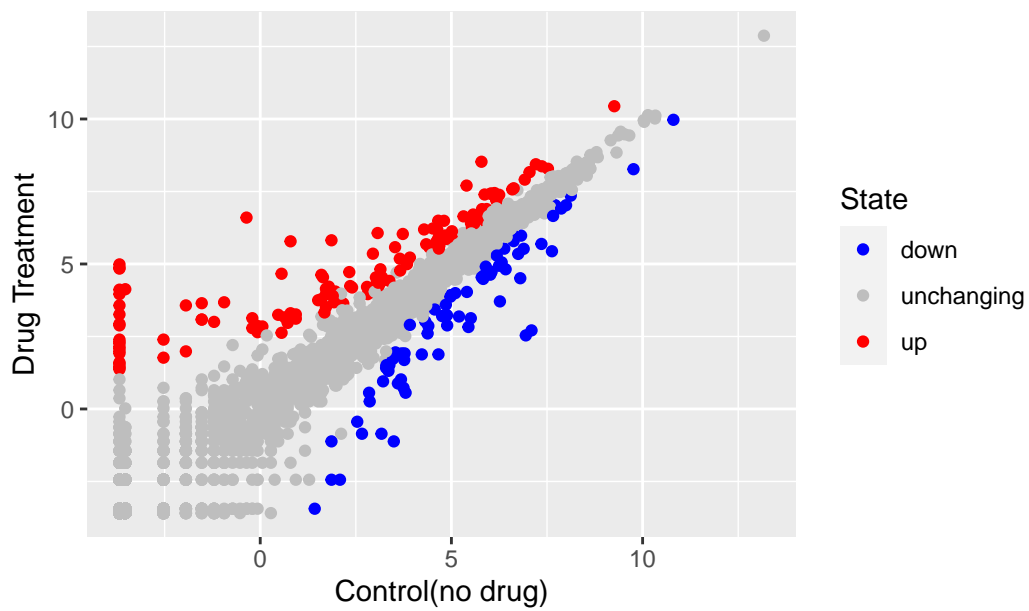
```
p
```

Gene Expression Changes Upon Drug Treatment



```
p + scale_colour_manual(values=c("blue","gray","red"))
```

Gene Expression Changes Upon Drug Treatment



```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder."

gapminder <- read.delim(url)

library(dplyr)
```

Attaching package: 'dplyr'

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

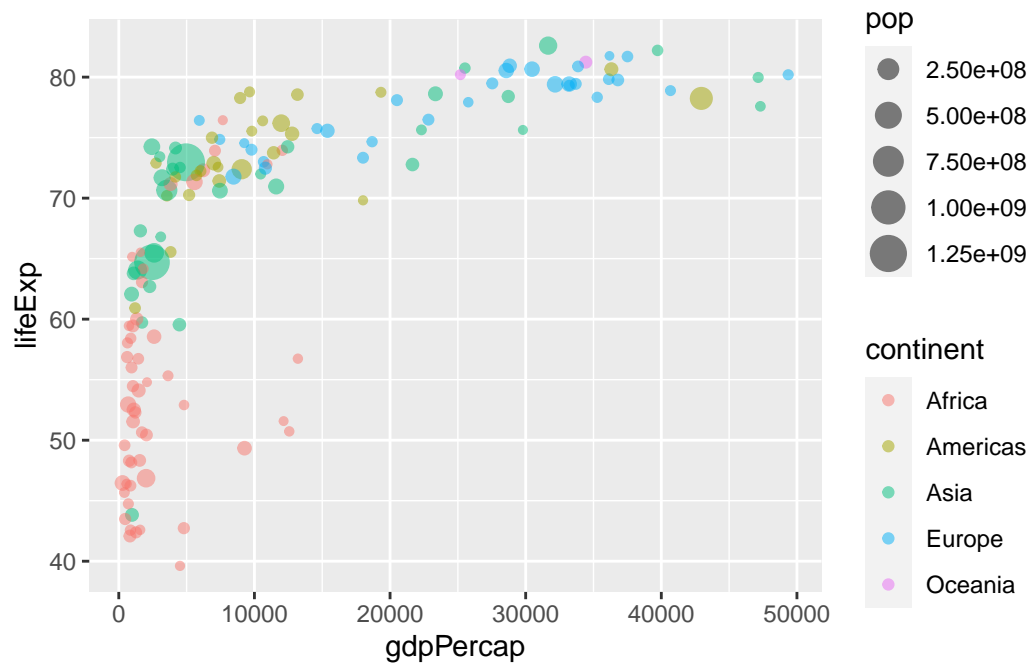
filter, lag

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

intersect, setdiff, setequal, union

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

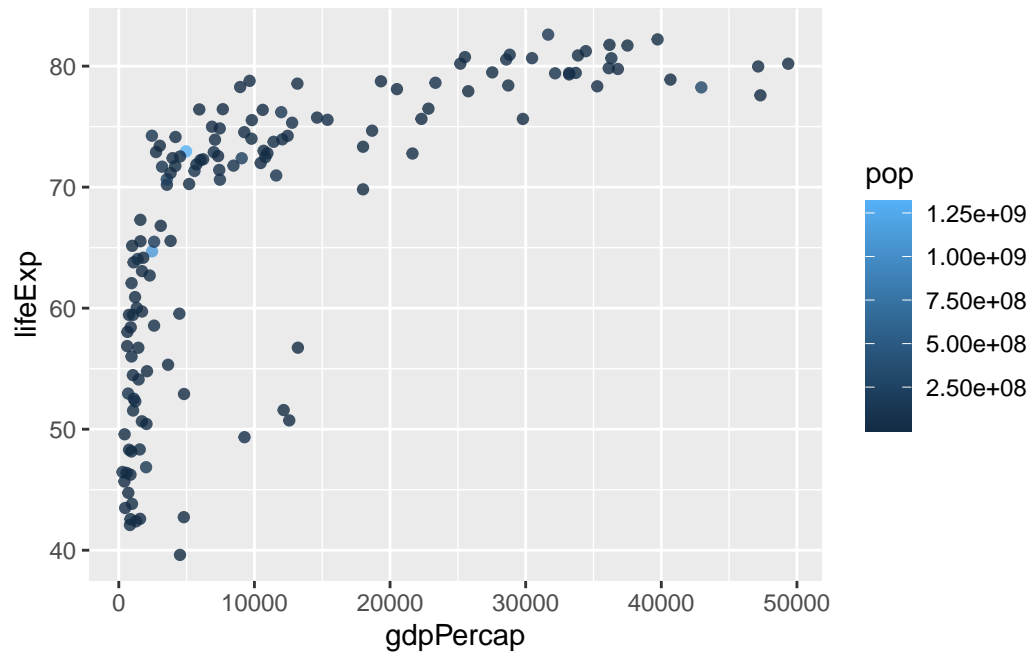
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color= continent, size = pop) +
  geom_point(alpha=0.5)
```



color the points by the numeric variable population pop:

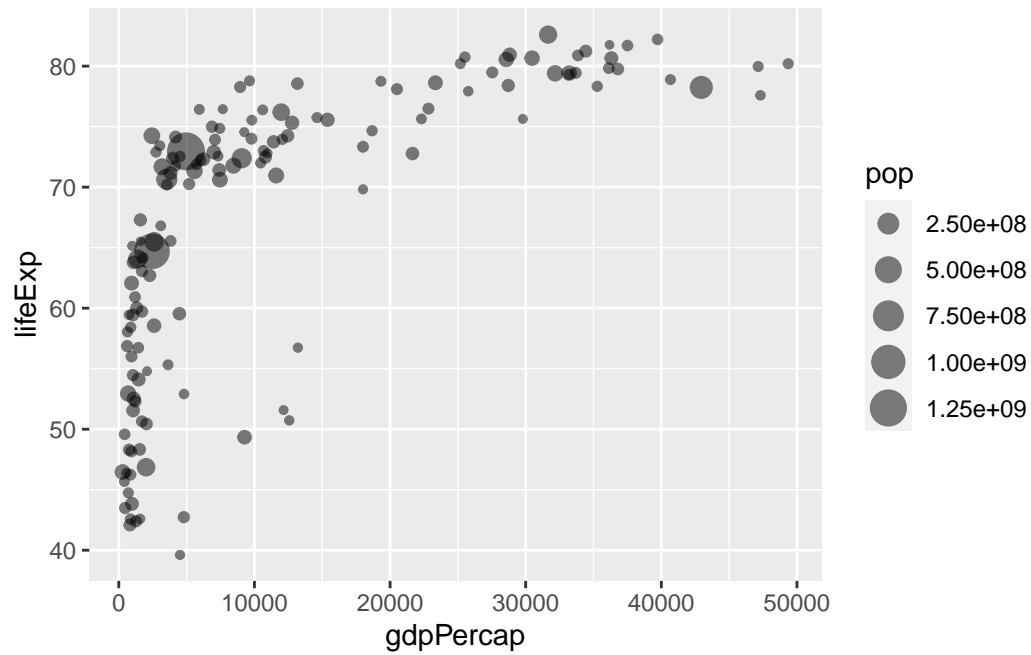
```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```





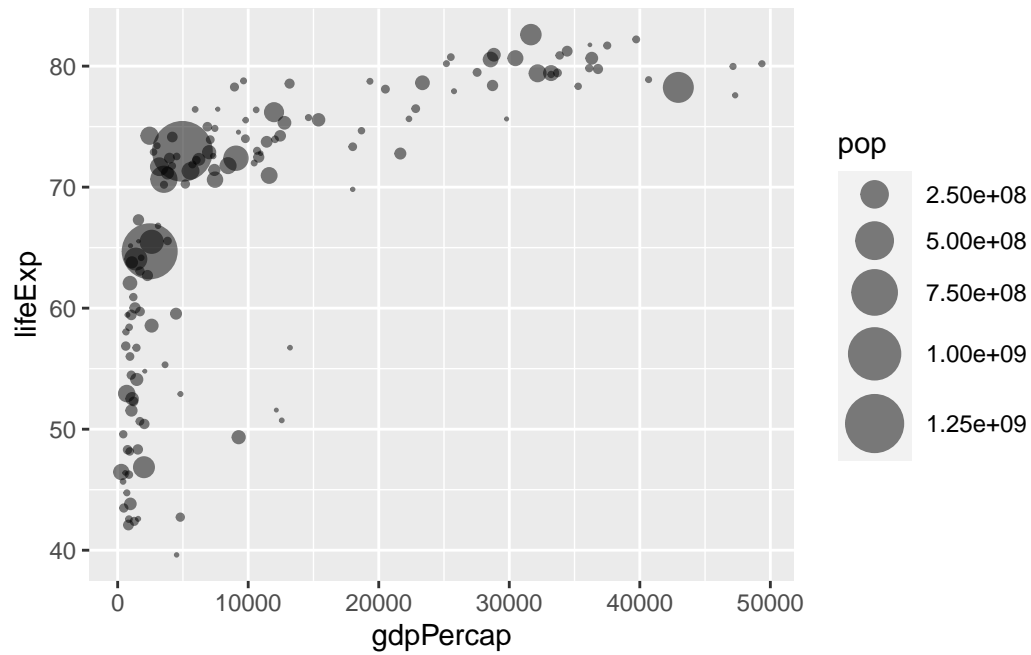
Adjusting point size:

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



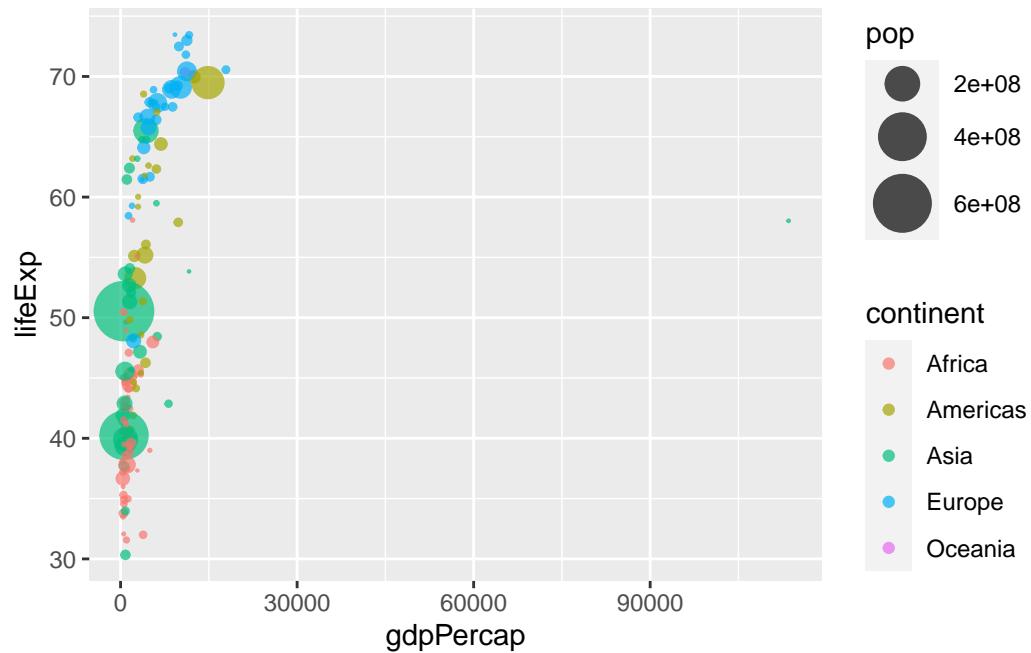
To reflect the actual population differences by the point size:

```
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)

ggplot(gapminder_1957)+aes(x = gdpPercap, y = lifeExp, color=continent,size = pop) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 10)
```



Comparison:

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

