Data Visualization Lab

Wai Lam Adele Hong A15999023

10/13/2021

Install the package ggplot2.

```
# install.packages("ggplot2")
```

Anytime I want to use this package, I need to load it.

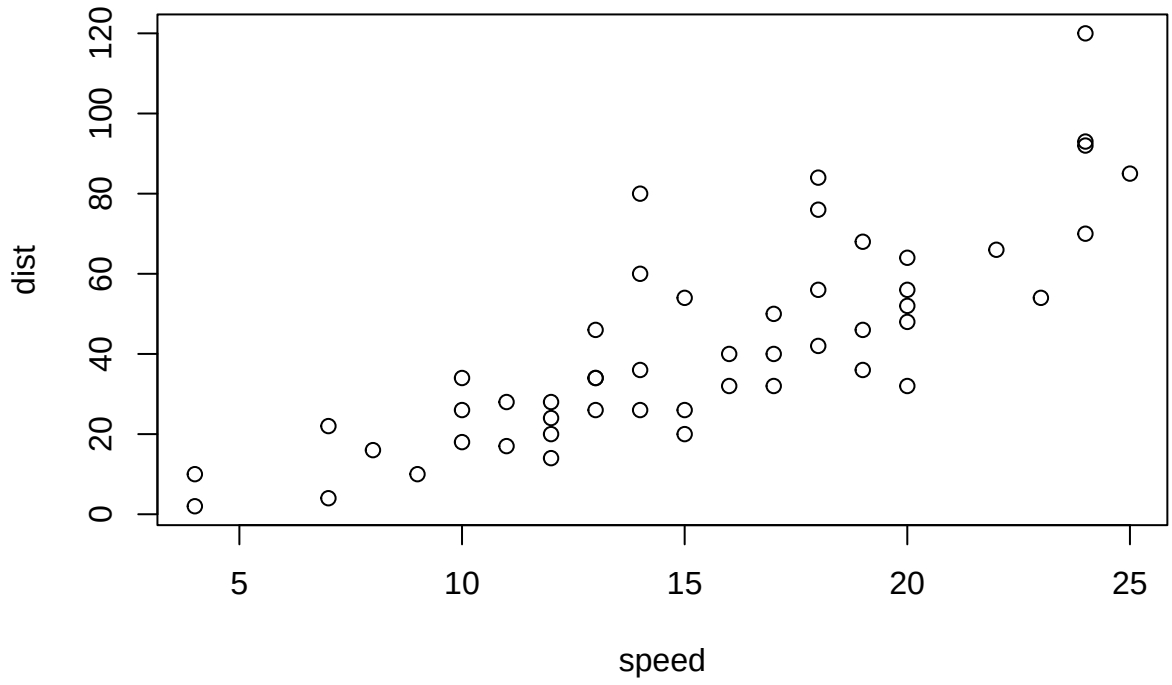
```
library(ggplot2)
```

Quick base R Plot - Cars

```
View(cars)
```

```
## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/Resources/
## modules/R_de.so'' had status 1
```

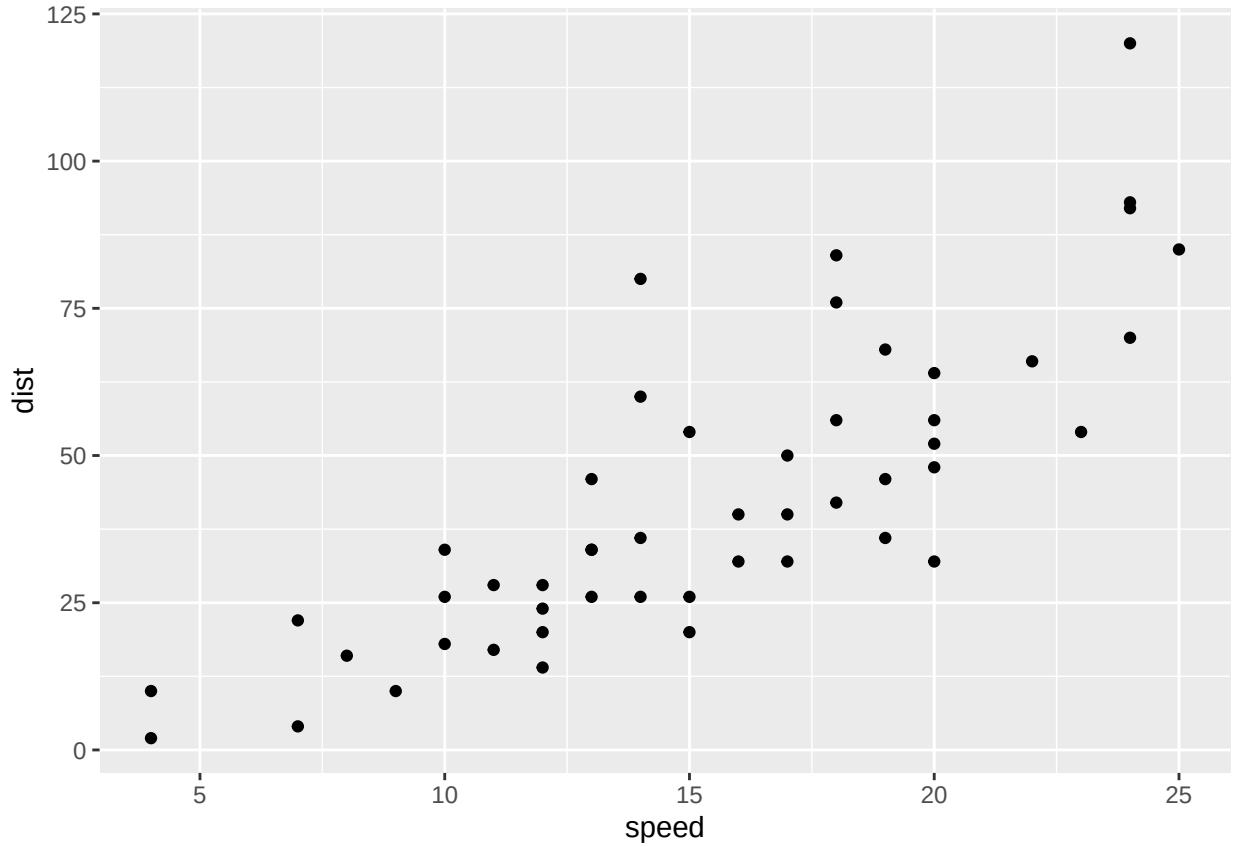
```
plot(cars)
```



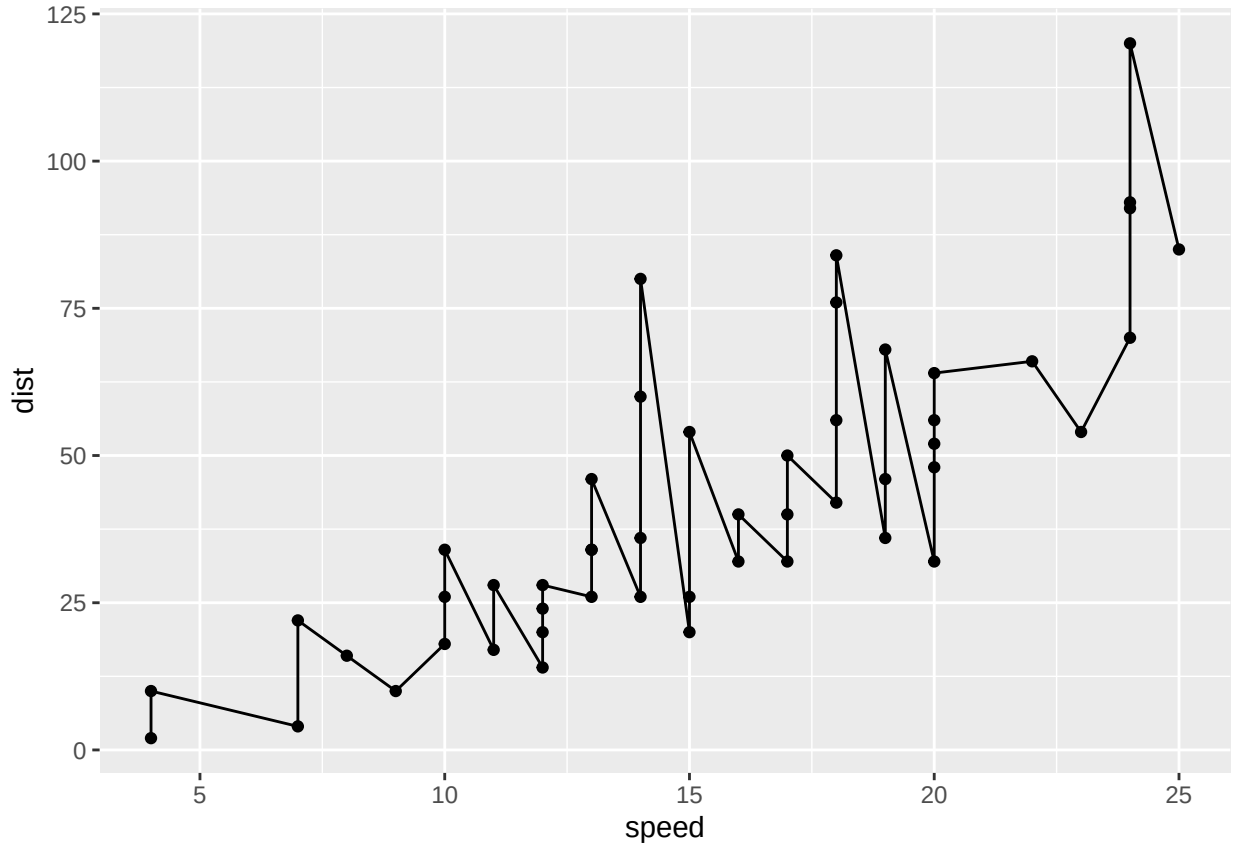

```
# This is NOT ggplot. It's only a simple plot.
```

Let's make our first ggplot. We need data + aes + geoms (data, aesthetic, geometry).

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

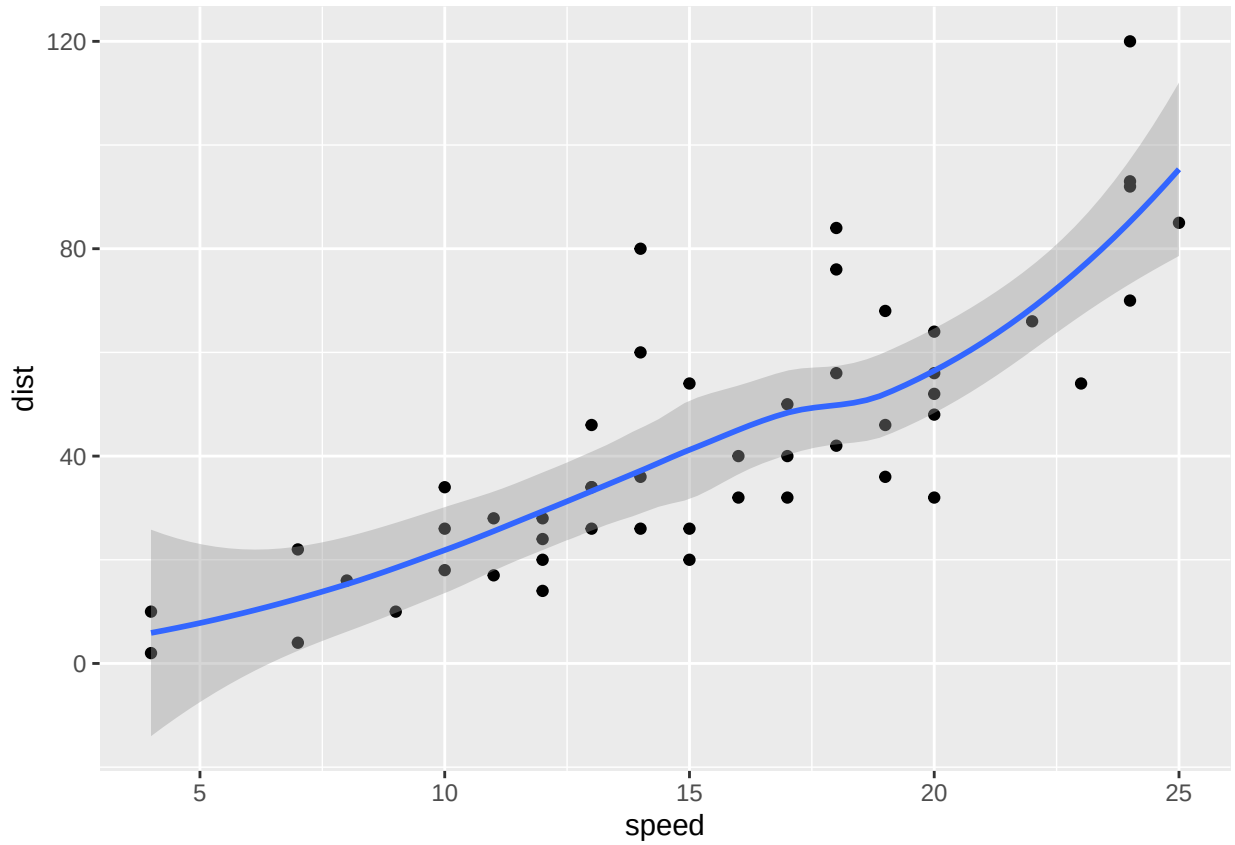



```
# "aes" is x and y axes. & visuals. "geom" is type of plot (how it's plotted, e.g point, line, bar, etc
p <- ggplot(data=cars) + aes(x=speed, y=dist) + geom_point()

# Add a line with geom_line
p + geom_line()
```



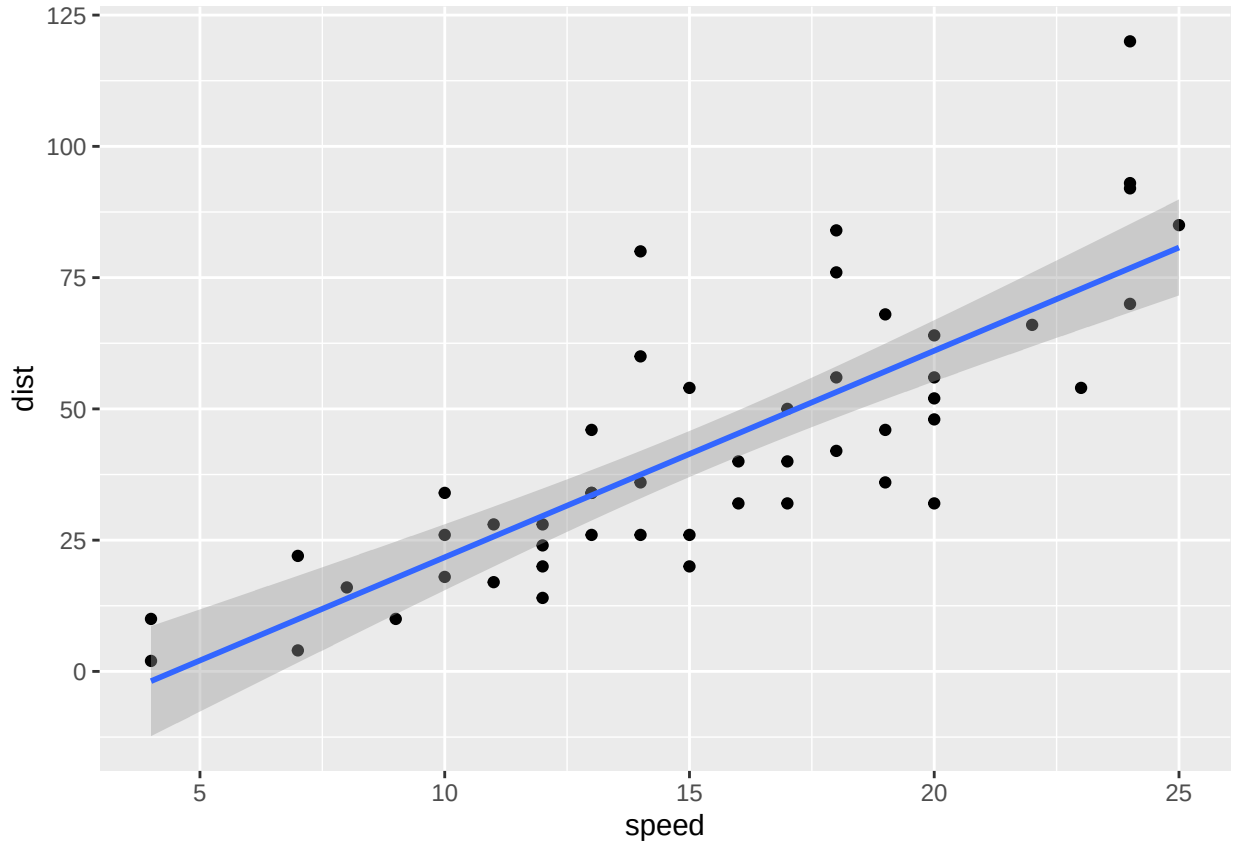

```
p + geom_smooth()
```

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




```
# Code below shows trendline ("lm" means linear model)
p + geom_smooth(method="lm")
```

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



Read in our drug expression data

```
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 there?

```
r nrow(genes)
## [1] 5196 Q. How many columns and what are the names?
r ncol(genes)
## [1] 4
r colnames(genes)
## [1] "Gene"       "Condition1" "Condition2" "State"
```

Q. How many “up” regulated genes?

```
r table(genes$State)
## ##       down unchanging         up ##         72
4997        127
r # For a summary of data frame, use table() function
```

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

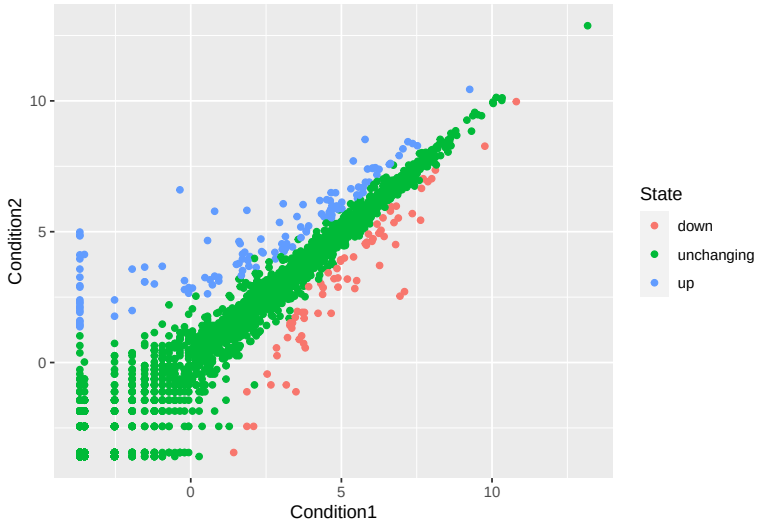
```
r round((table(genes$State) / nrow(genes)) * 100, 2)
```

```
## ##       down unchanging         up ##       1.39
96.17       2.44
r # Each value for State divided by the total # of
genes. The "2" at the end indicates how many decimals
to round it to.
```

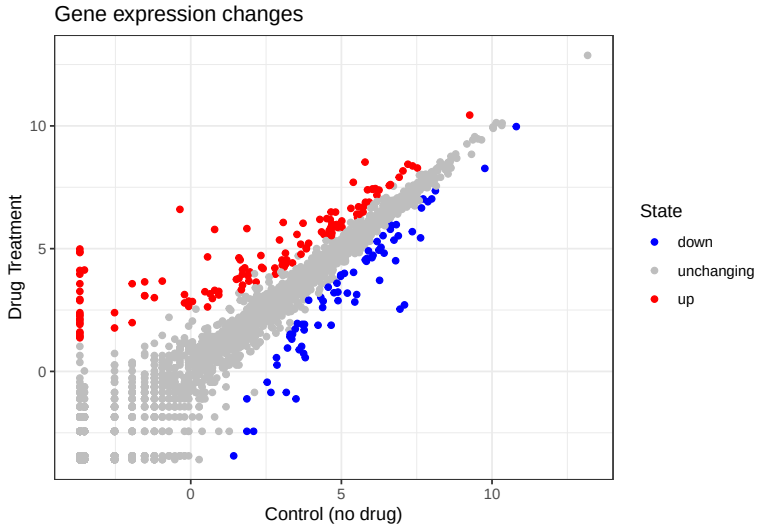
Let's make our first plot attempt. NOTE: if we used the variable "p", it would override previous command regarding "p" (storing obj in "p")

"'r g <- ggplot(data=genes) + aes(x=Condition1, y=Condition2, col=State) +geom_point() # col=State color-codes based on types of State.

g "'




```
r # Change colors g +
scale_color_manual(values=c("blue", "gray", "red")) +
labs(title="Gene expression changes", x="Control (no
drug)", y="Drug Treatment") + theme_bw()
```



```
r # With "labs() function, we can add title, axis
titles, legends, etc. # theme_bw function changes theme
of graph
```

Optional Section

Install gapminder package, which is dataset for economic and demographic info of various countries throughout the yrs.

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

```
## # A tibble: 1,704 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.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

Install **dplyr code** to focus in on a single year.

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

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

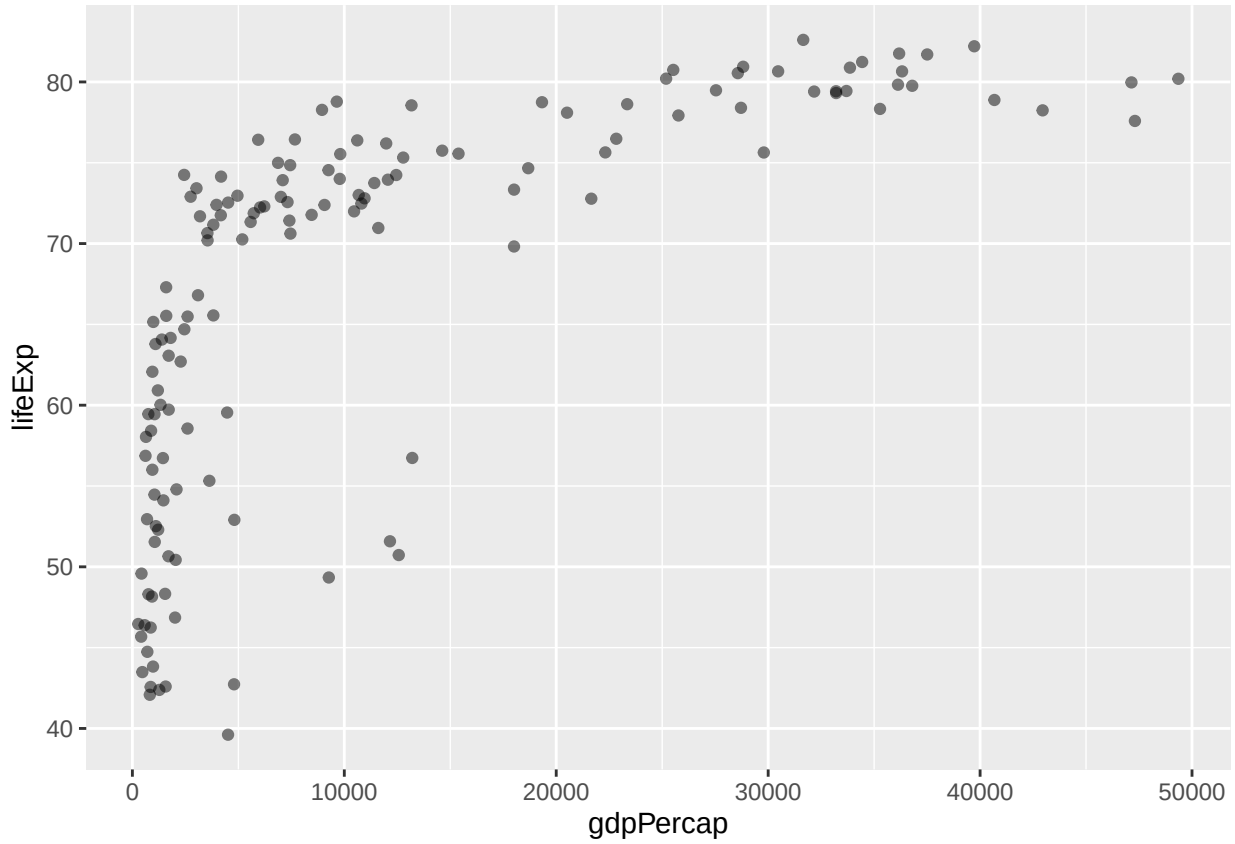
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
gapminder_2007 <- gapminder %>% filter(year==2007)
# Filters to contain only the rows with a year value of 2007
```

Let's make a scatterplot for 2007 data subset.

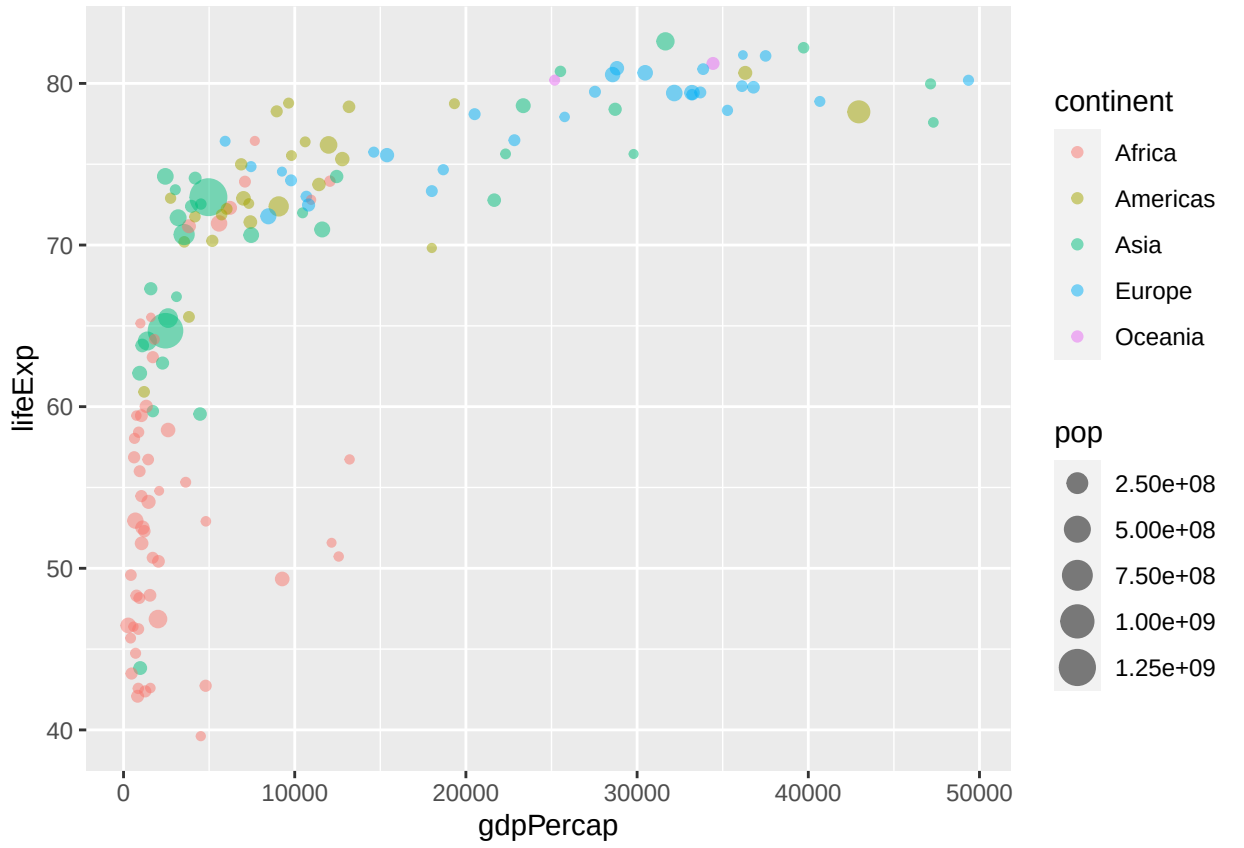
```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```




```
# "alpha=0.5" helps make data points a little more transparent to see overlap more clearly
```

Add variables to aes, using additional arguments.

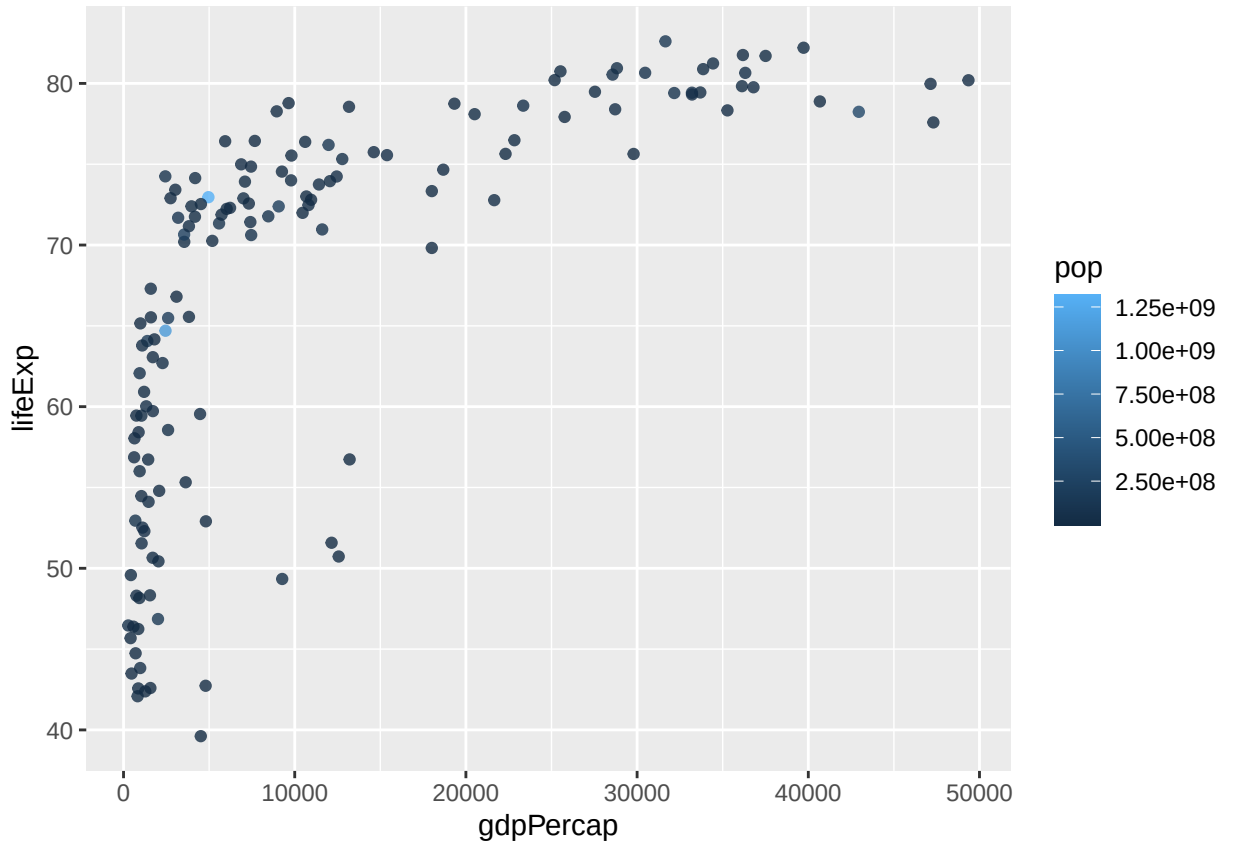
```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```




```
# Each dimension adds a new layer to the plot! Cool!
```

What happens if we display "pop" using color?

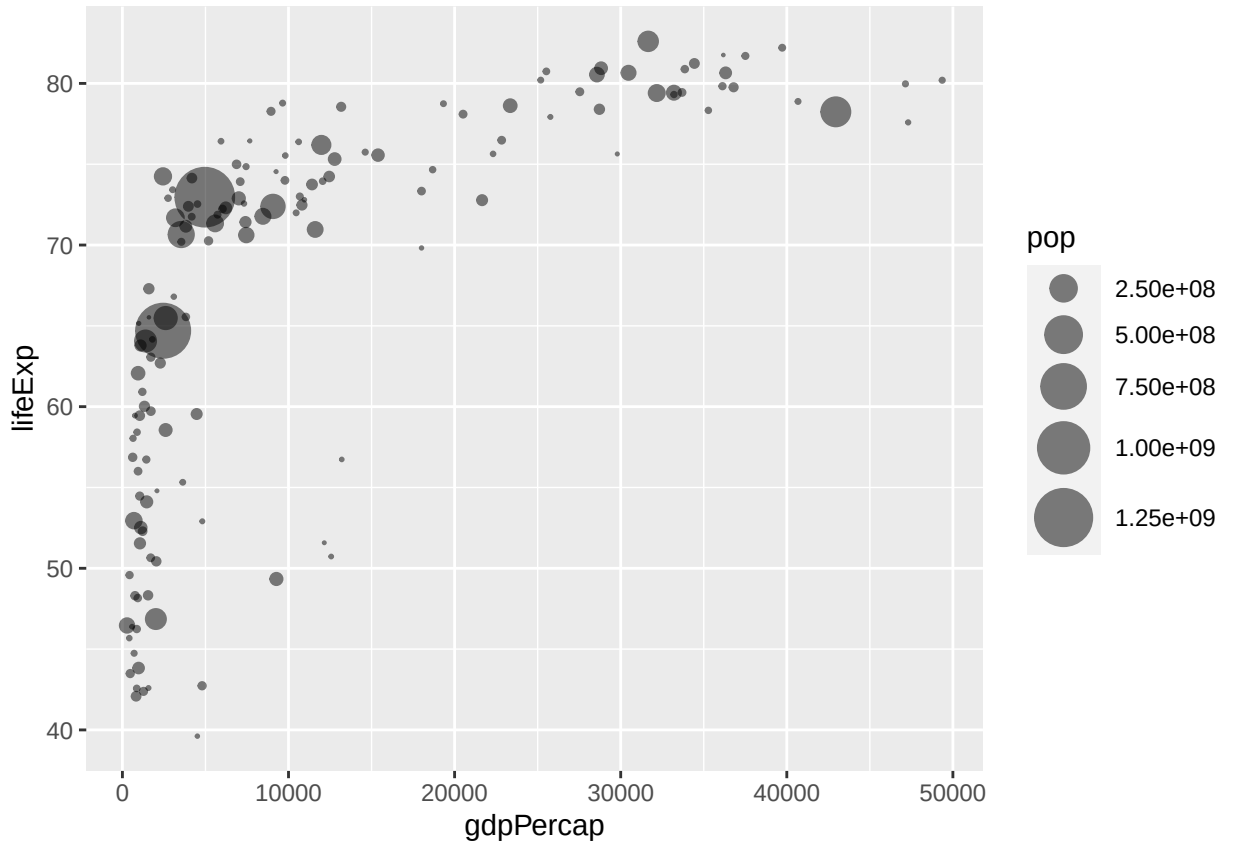
```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```




```
# Since the "pop" data is continuous (as opposed to discrete), it is displayed as a color *spectrum*
```

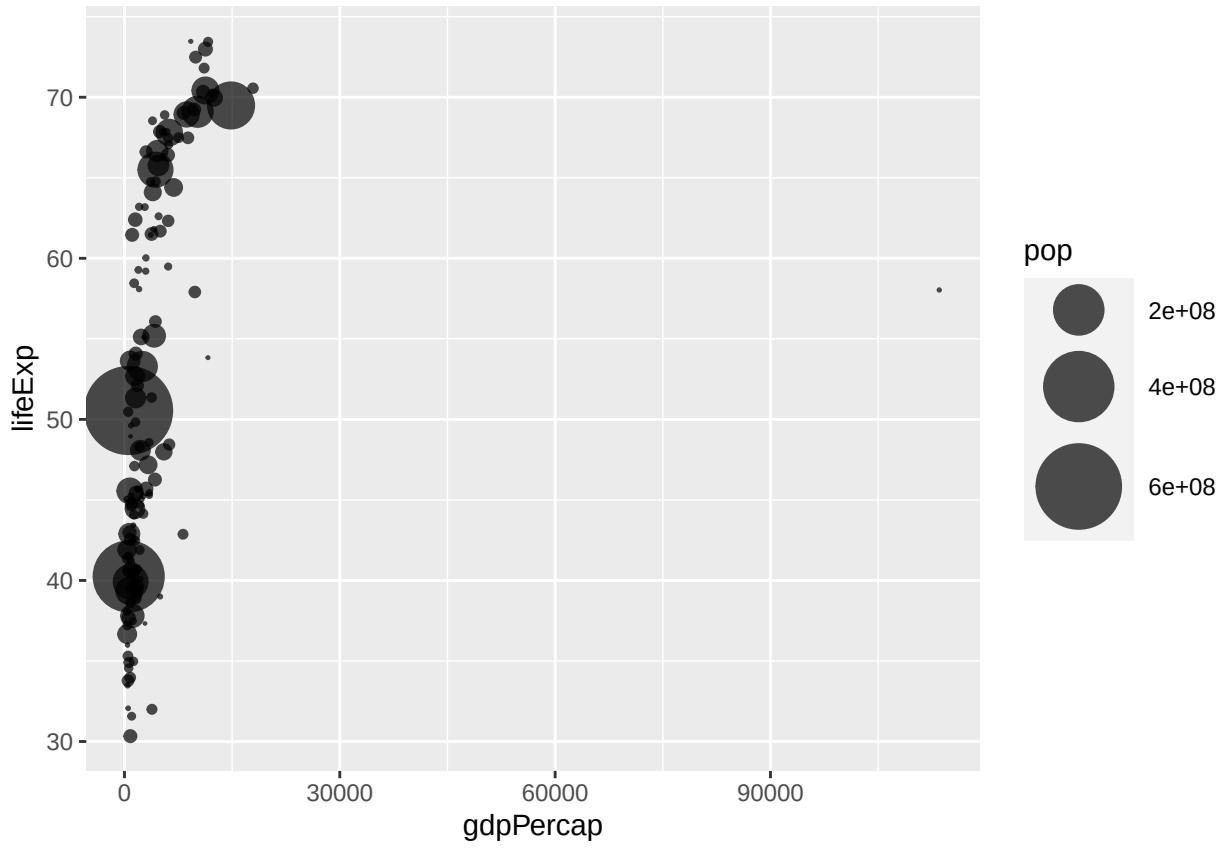
When we use size to display "pop", we realize the sizes don't proportionally reflect the population (i.e the sizes are binned by default). To fix this, we use **scale_size_area()** function.

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



Q. Can you adapt your plot for the year 1957? A. Yes. It's relatively difficult to compare the 2007 plot to the 1957 because they have different scales (i.e x and y axes have different ranges).

```
# Don't forget to filter yr 1957 and save to "gapminder_1957"
gapminder_1957 <- gapminder %>% filter(year==1957)
ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp,
                size = pop) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 15)
```

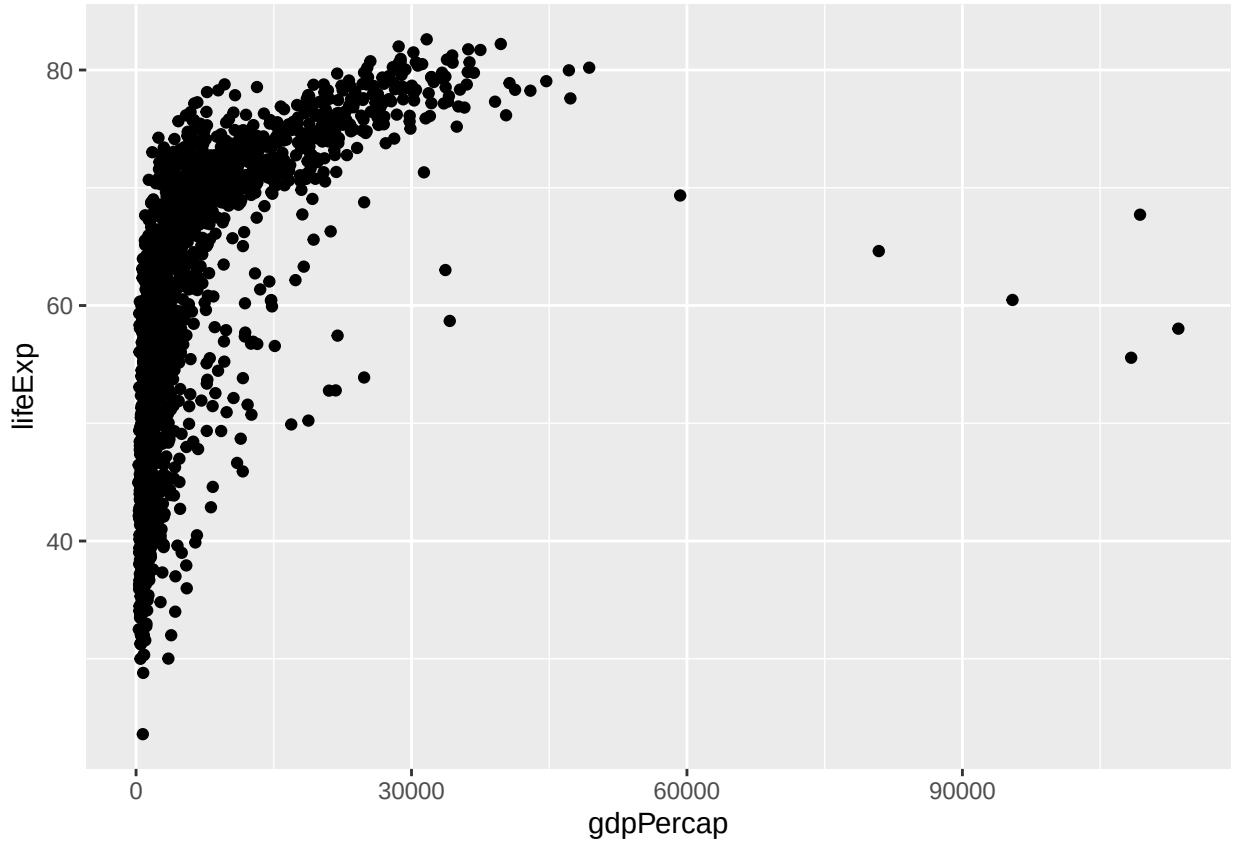


Q. Let's compare them in an easier way; put both years as an input for ggplot. Use **facet_wrap()** function.

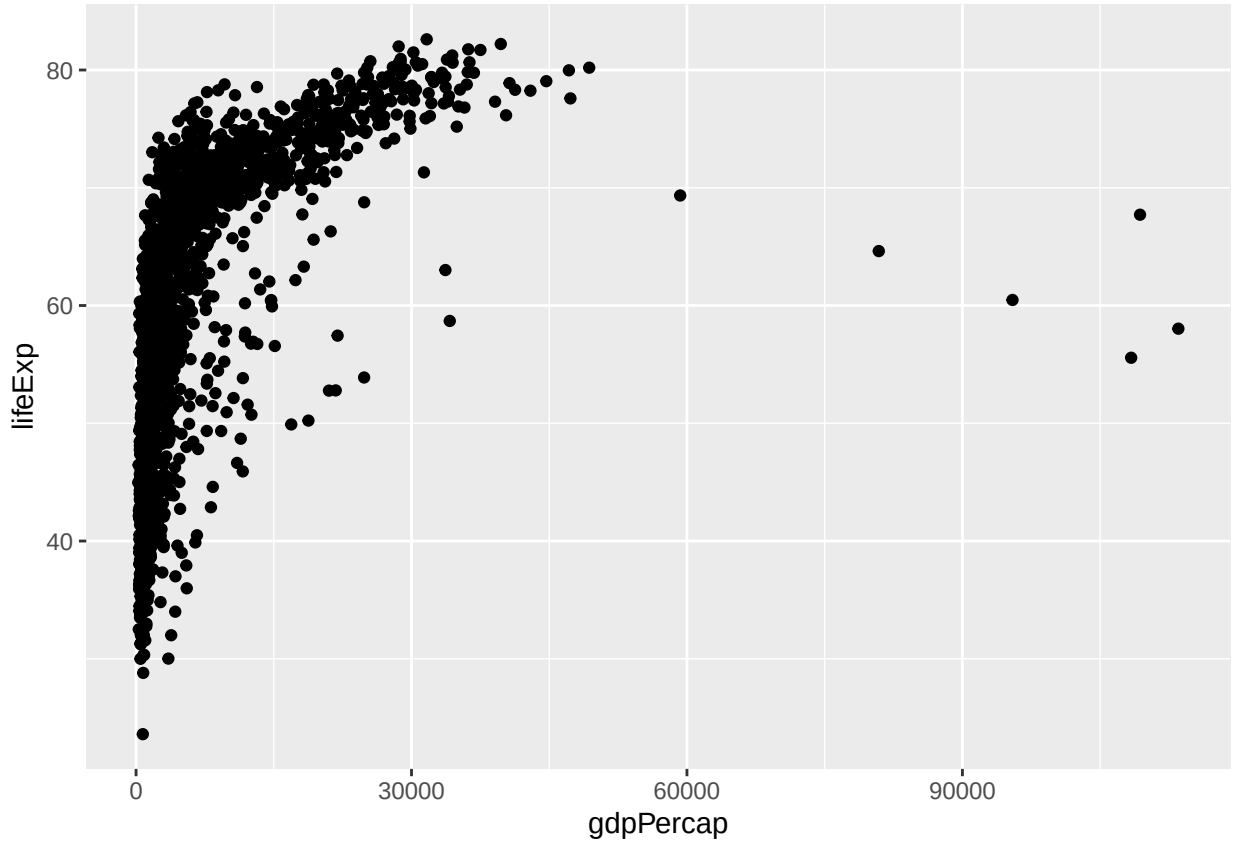
```
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)

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)
```


```
# Notice geom_point could also be written like this ^
```

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




```
# Note: this is also a viable way of writing the code above:
ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point()
```


80
60
40
lifeExp
0
30000
60000
90000
gdpPercap