Class05: Data Vis with ggplot

AUTHOR
Sawyer (PID: A16335277)

Graphics systems in R

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

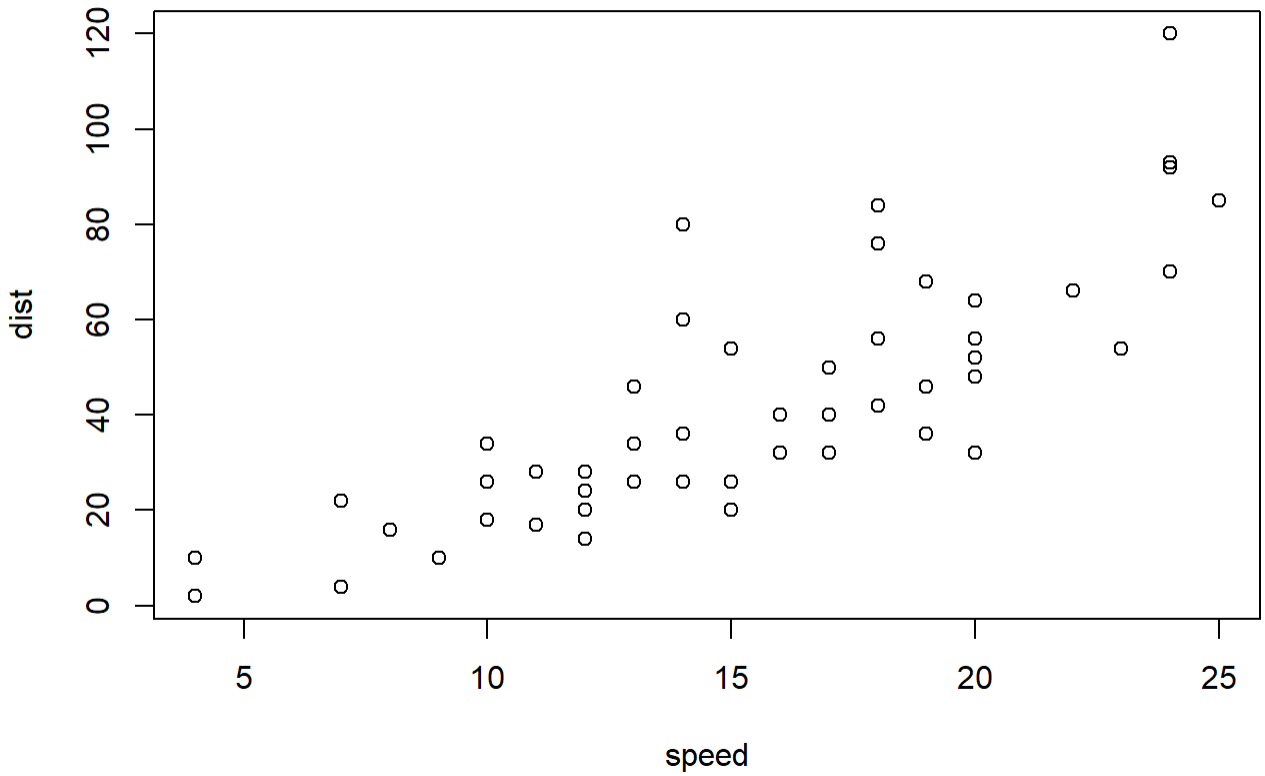
We have already played a little with **"base R"** graphics and the `plot()` function.

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. I install it (like I install any package) with the `install.packages()` function.

Creating Scatter plots

```
plot(cars)
```



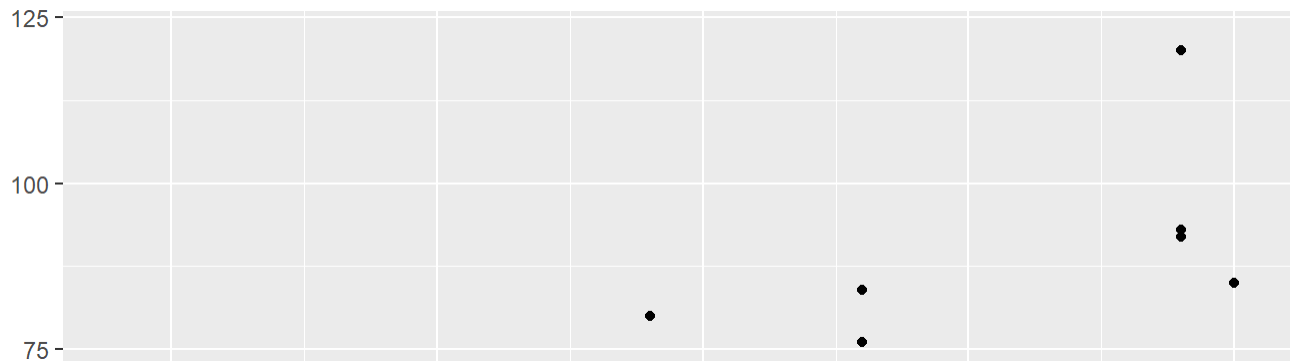
Before I can use the functions from a package I have to load up the package from my "library". We use

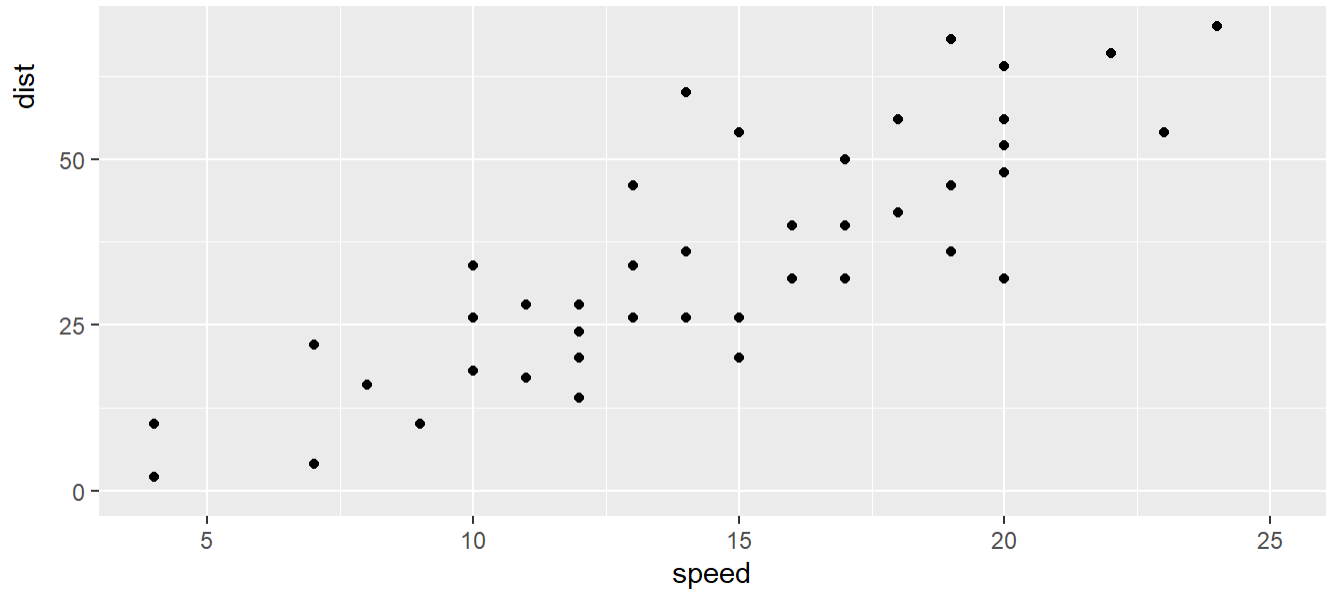
Before I can use the functions from a package I have to load up the package from my "library". We use the `library(ggplot2)` command to load it up.

```
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()
```



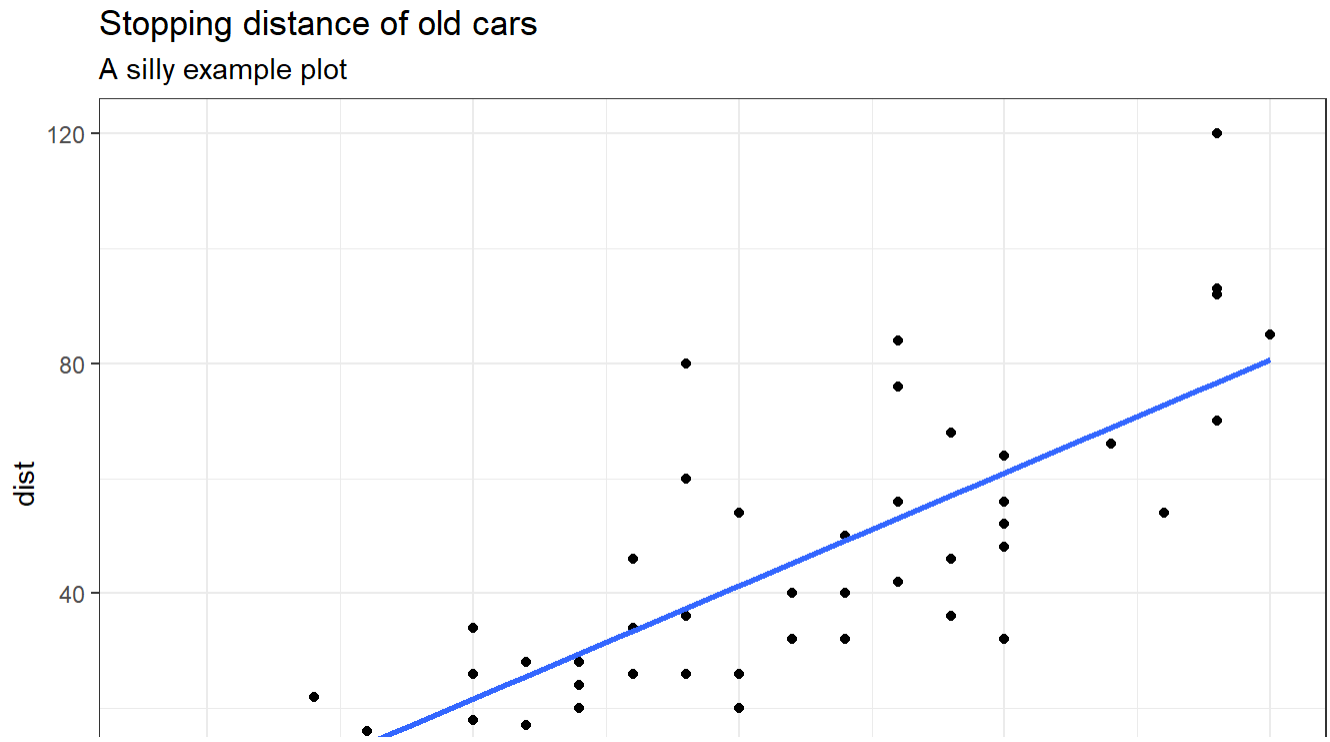


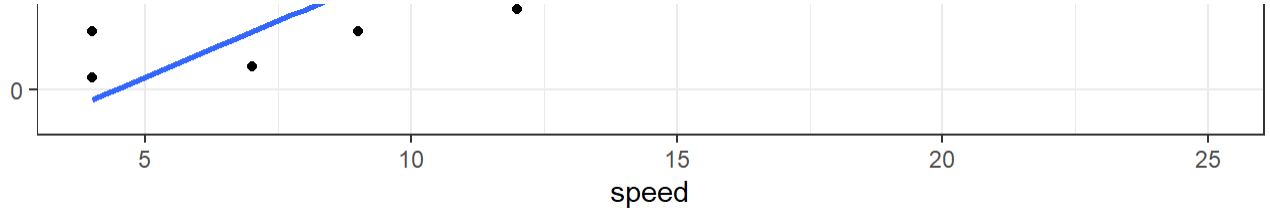
For simple plots ggplot is more verbose - it takes more code - than base R plot.

Add some more layers to our ggplot:

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

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






```
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?

```
colnames(genes)
```

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

```
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
```

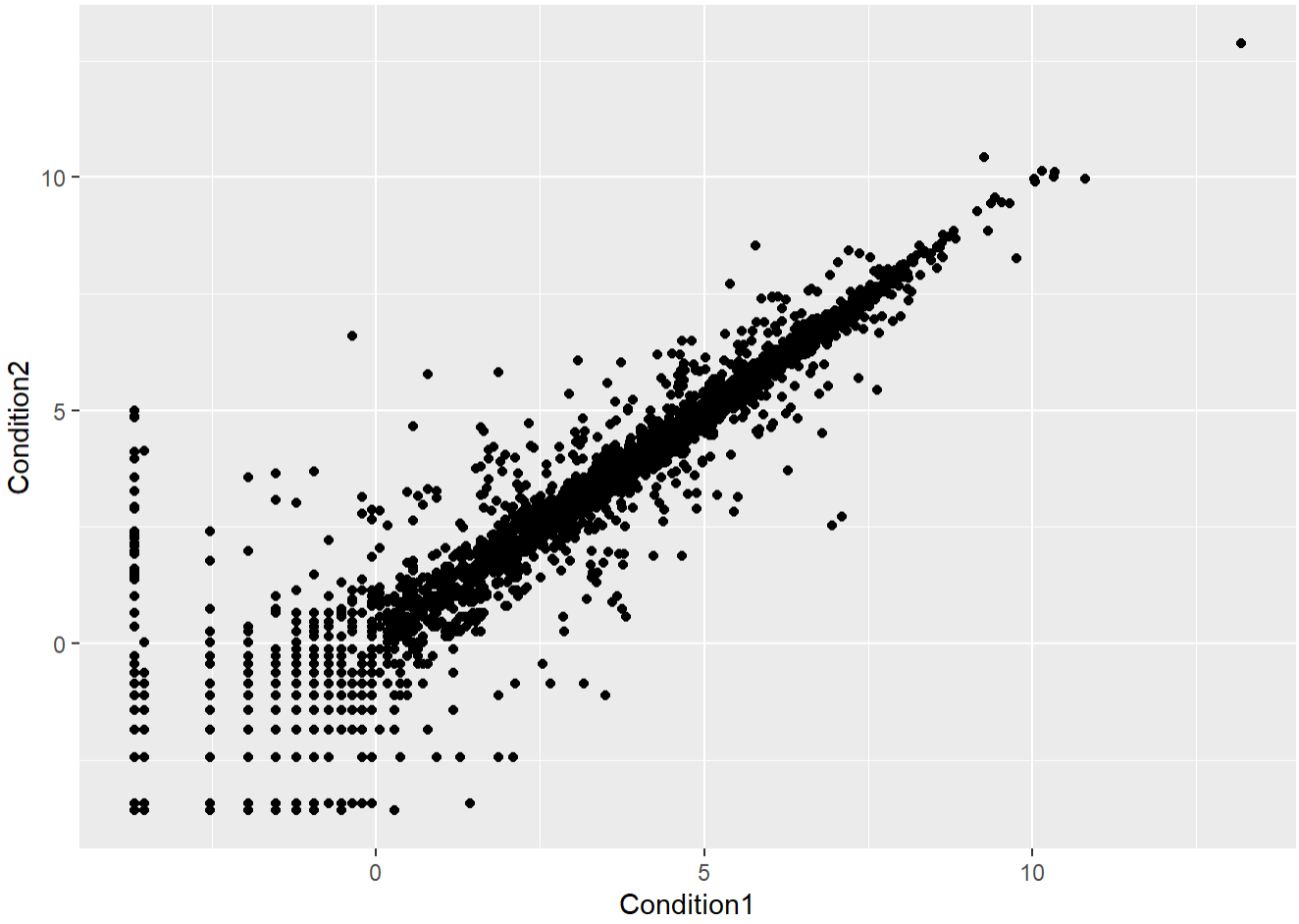
Q. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this

dataset?

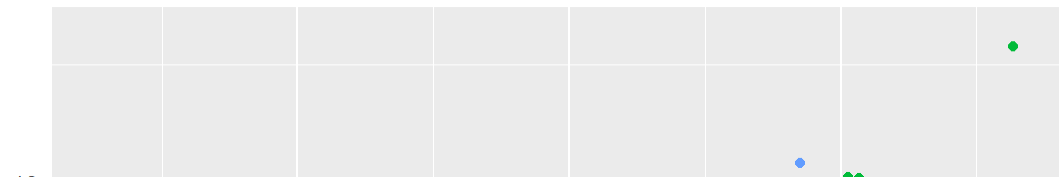
```
table(genes$State)/nrow(genes) * 100
```

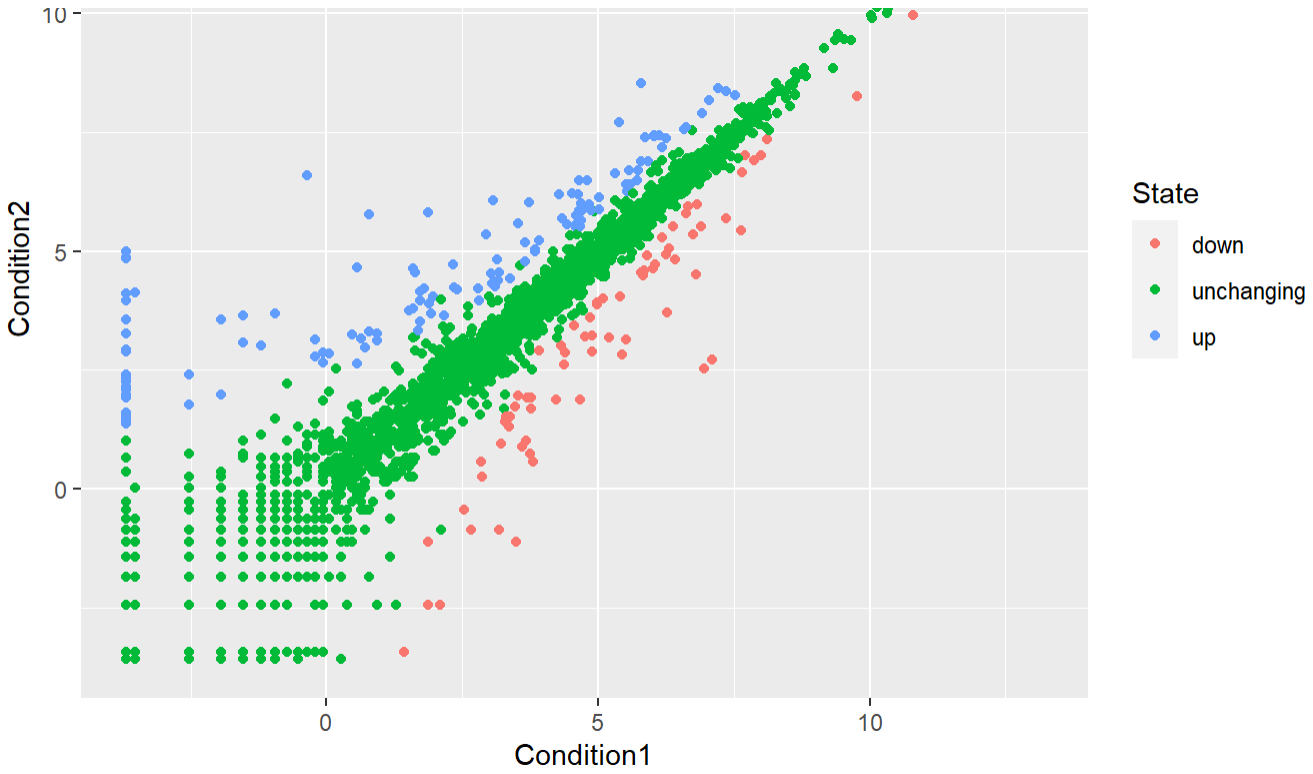
```
      down unchanging         up
 1.385681  96.170131   2.444188
```

```
ggplot(genes) +
    aes(x=Condition1, y=Condition2) +
    geom_point()
```

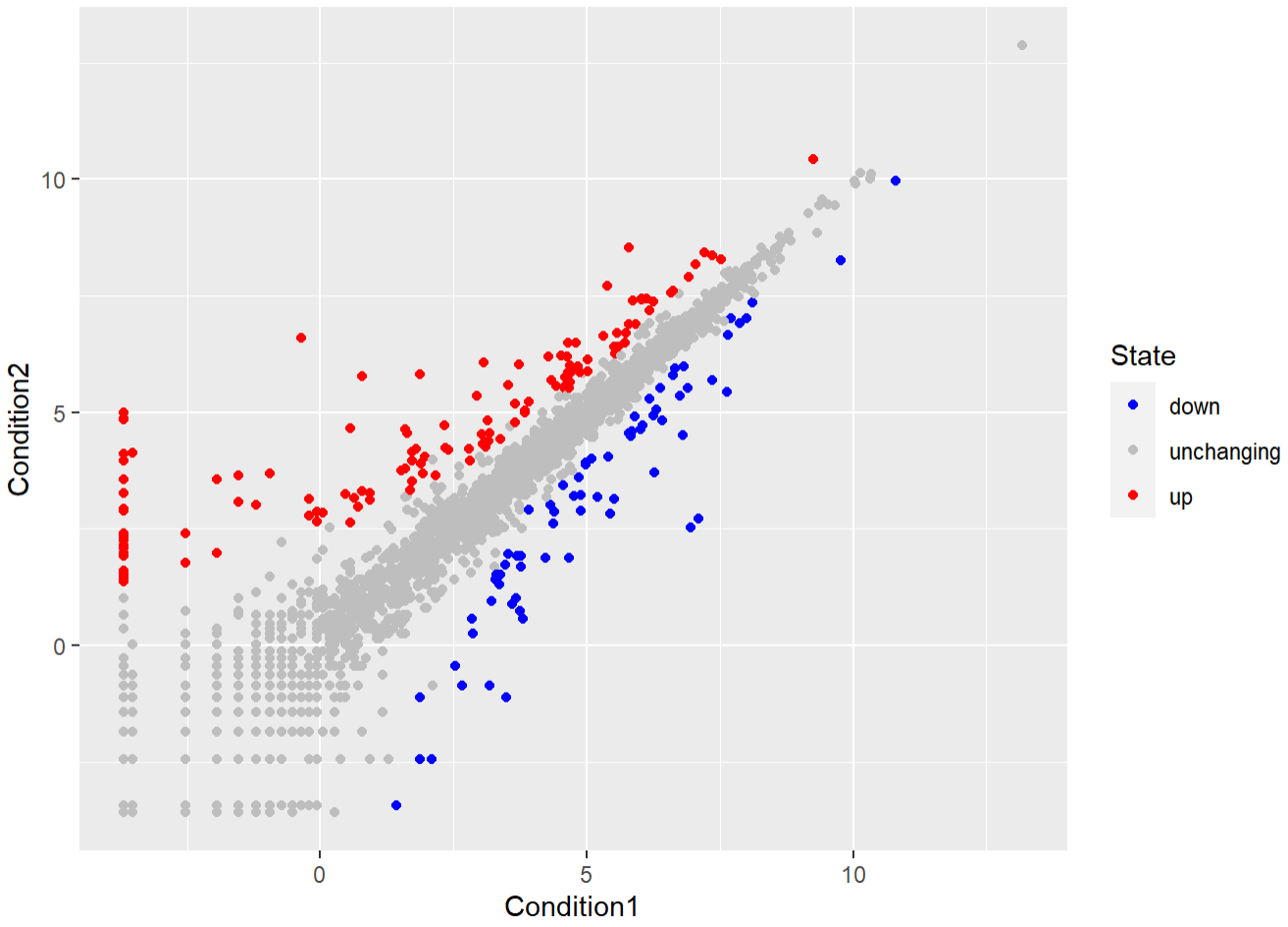



```
p <- ggplot(genes) +
    aes(x=Condition1, y=Condition2, col=State) +
    geom_point()
p
```

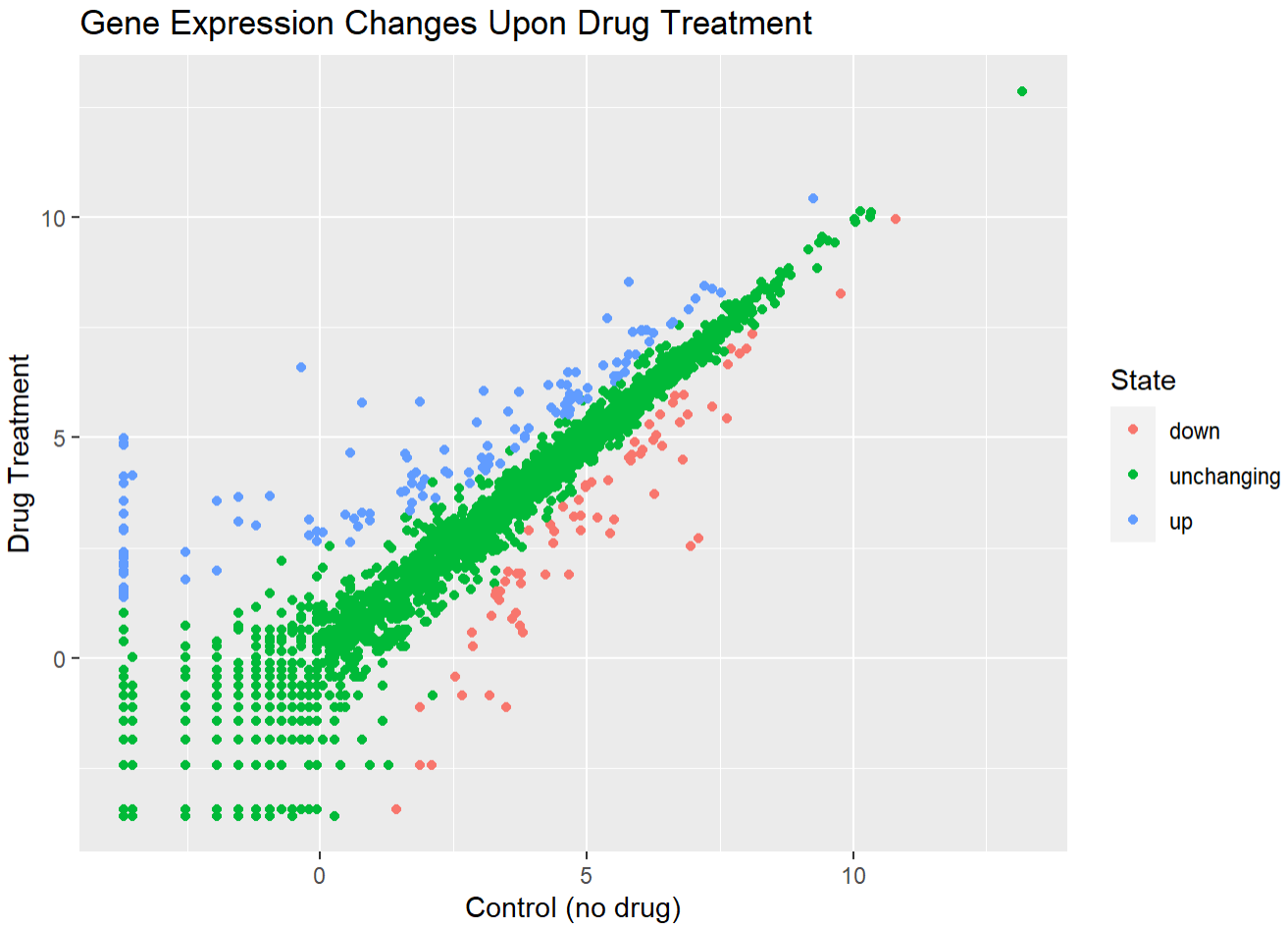





```
p + scale_colour_manual( values=c("blue","gray","red") )
```



```
p + labs(title="Gene Expression Changes Upon Drug Treatment",
       x = "Control (no drug)", y = "Drug Treatment")
```



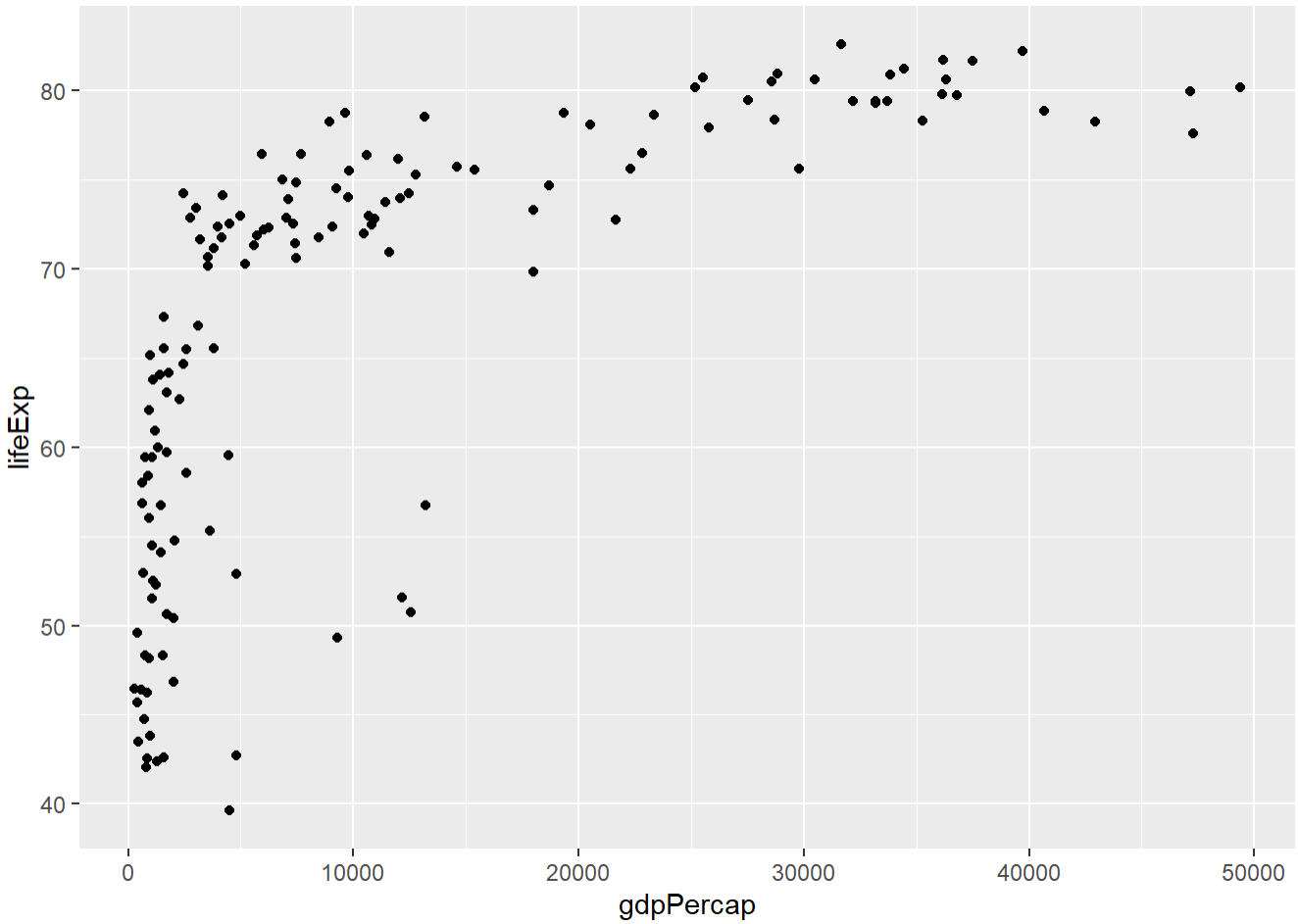

```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.tsv"

gapminder <- read.delim(url)
```

```
library(dplyr)
```

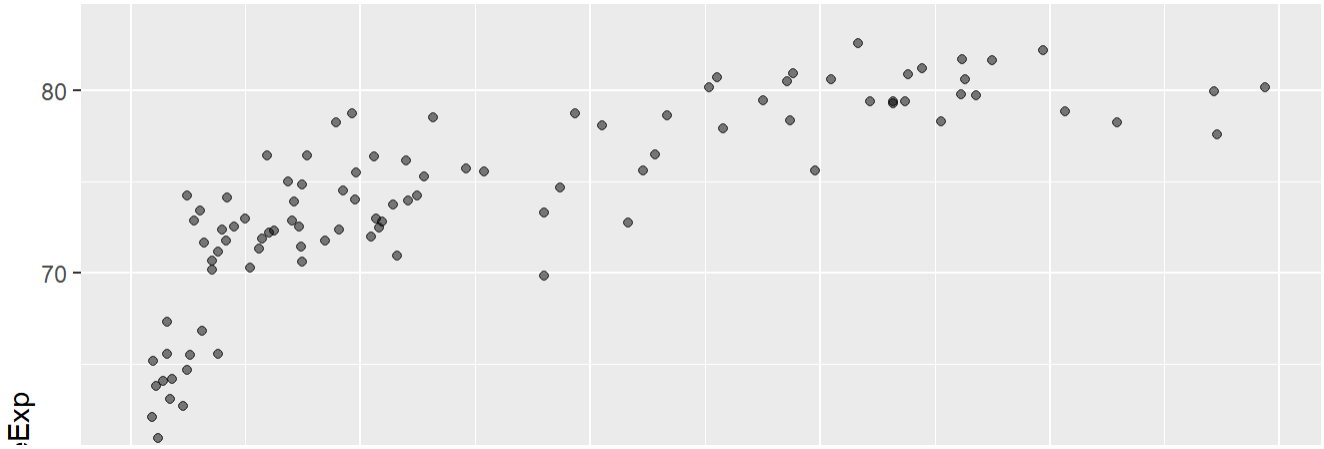
```
Attaching package: 'dplyr'

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

    filter, lag

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

    intersect, setdiff, setequal, union
```

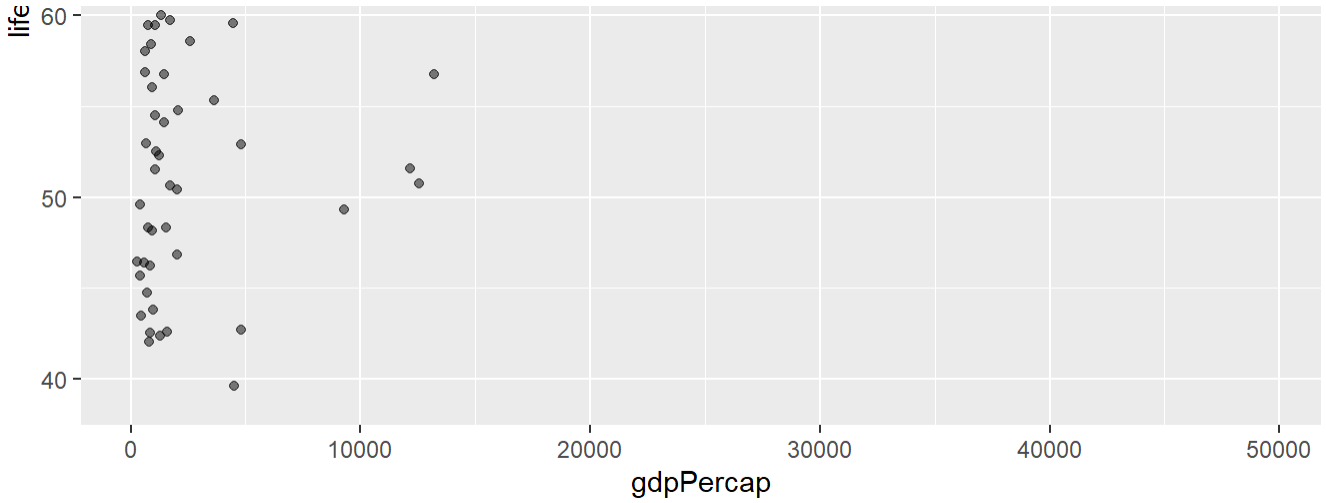
```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

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

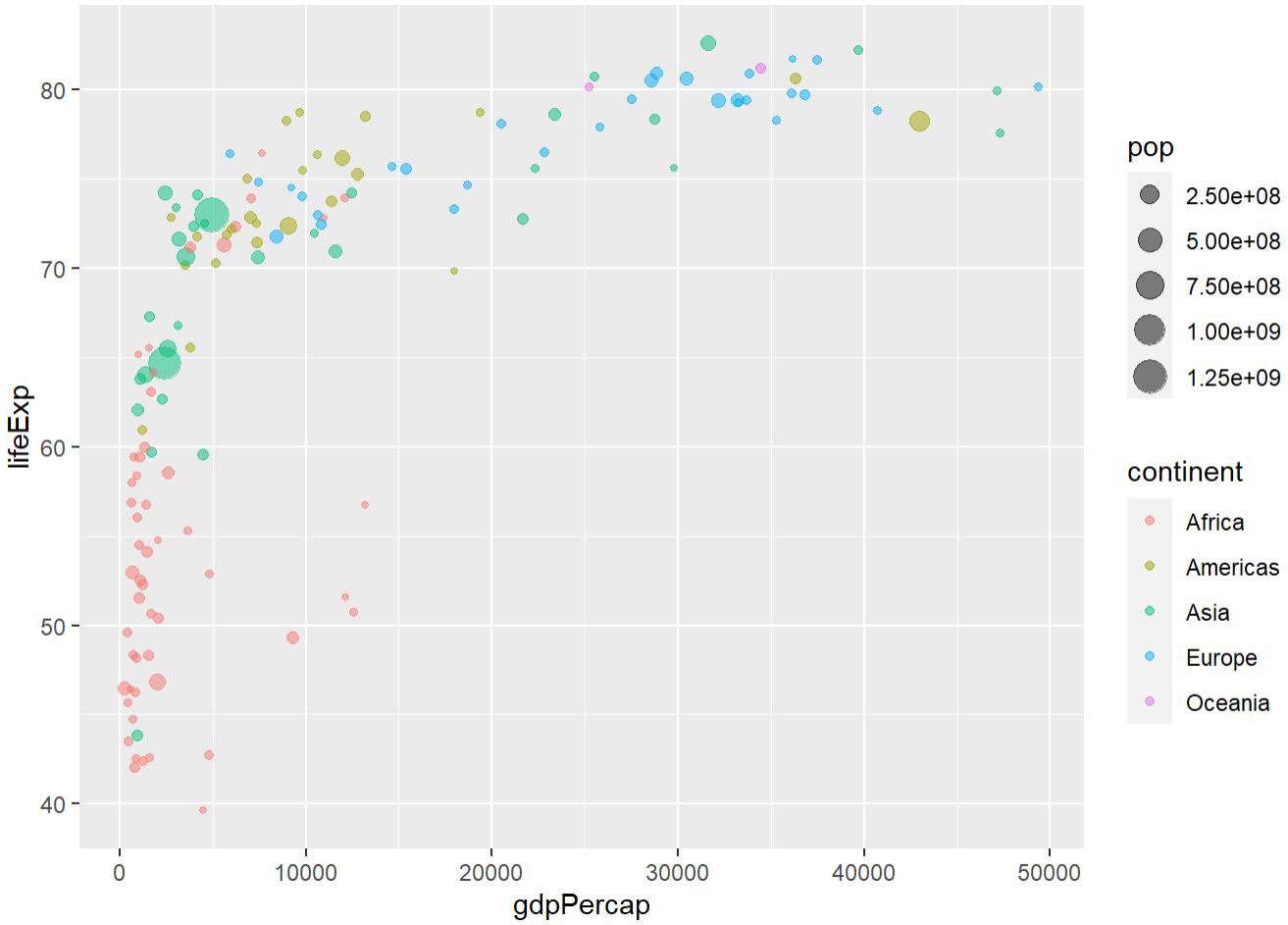



```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```

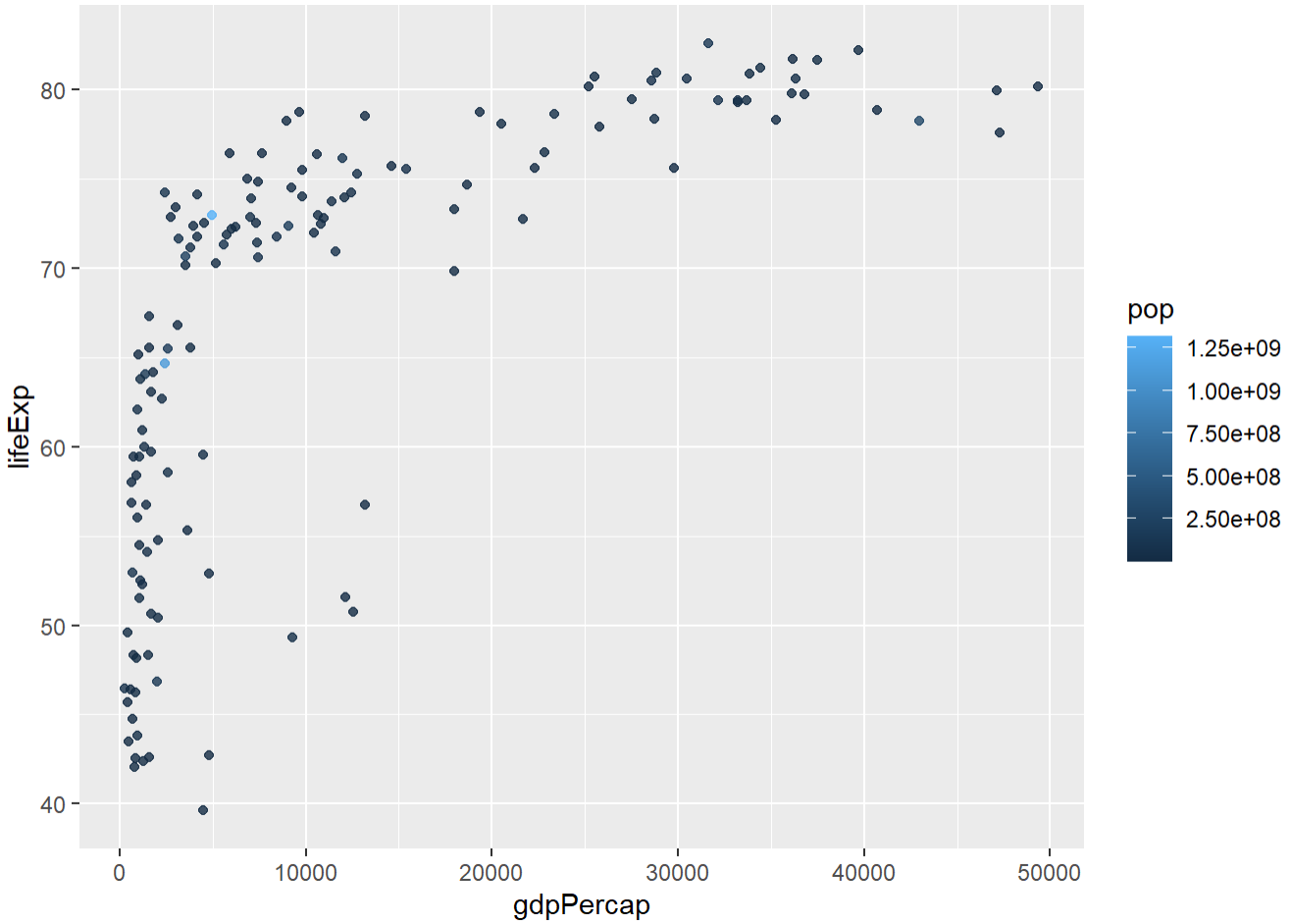




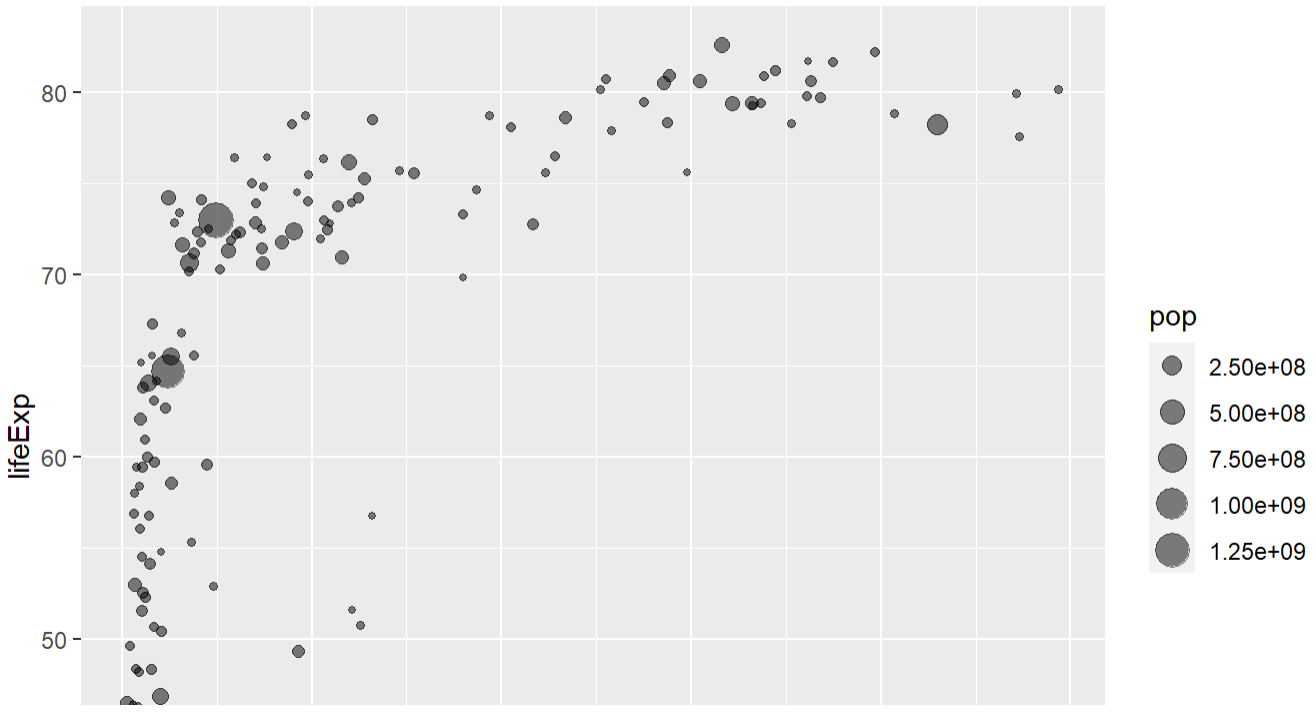

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```

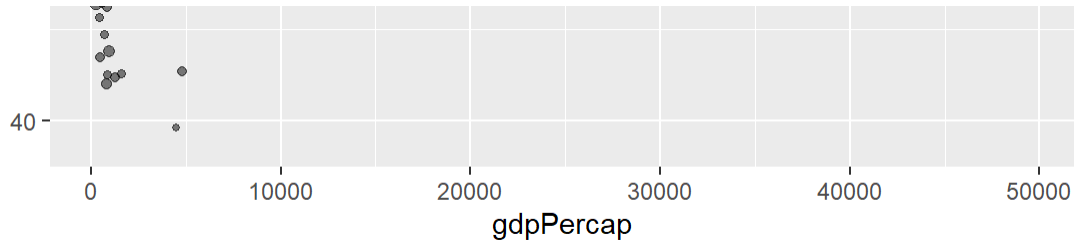



```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```

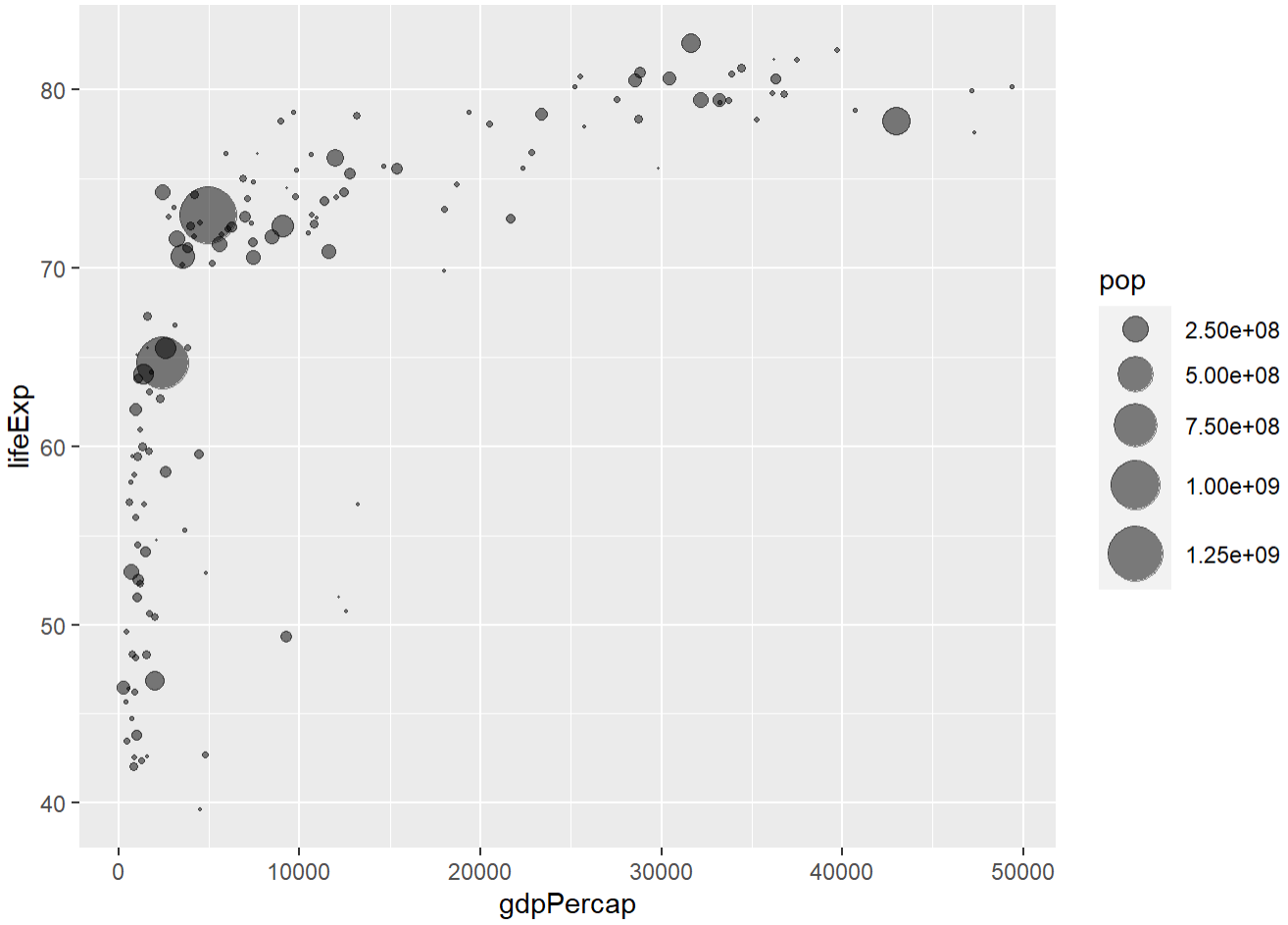



```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```



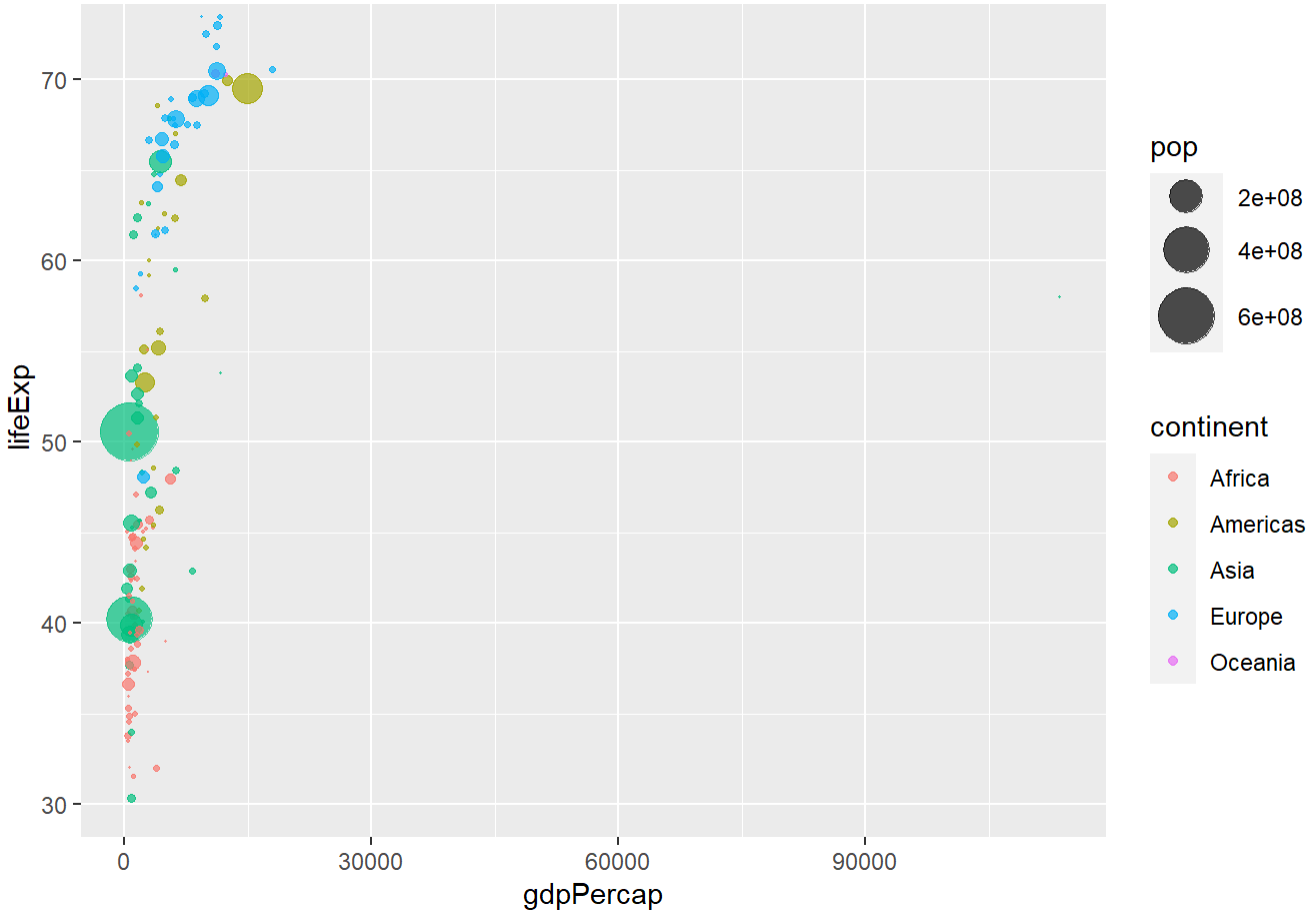



```
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop), alpha=0.5) +
  scale_size_area(max_size = 10)
```



-Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

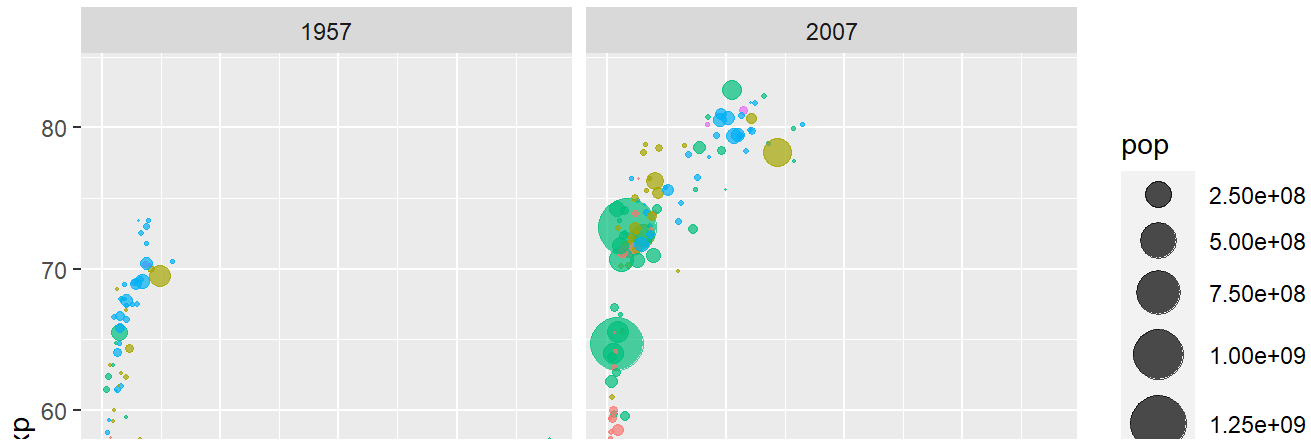
```
gapminder_1957 <- gapminder %>% filter(year==1957)
ggplot(gapminder_1957) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.7) +
scale_size_area(max_size = 10)
```

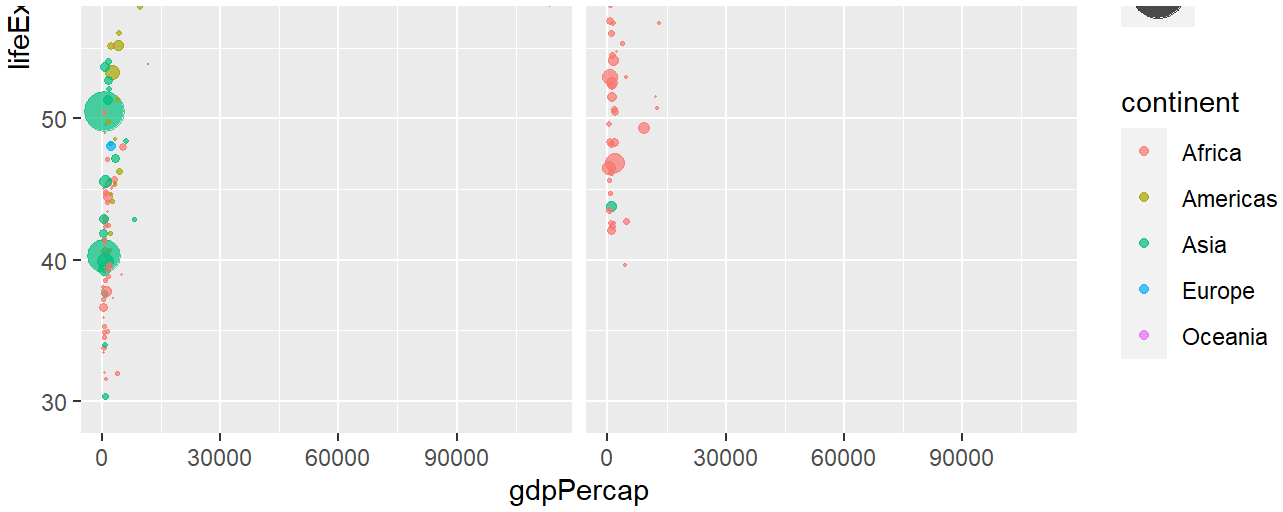


In this plot there is one clear outlier on the far right, slightly distorting the visual. It is not easy to compare to this data to that of 2007 because the data are not side by side and do not share axes.

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



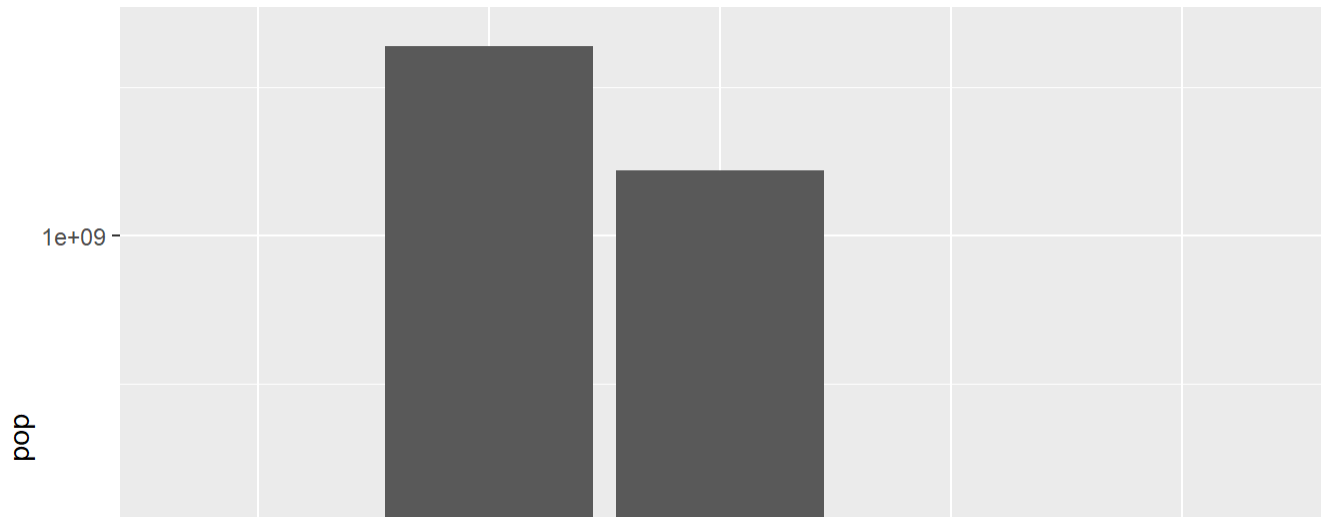


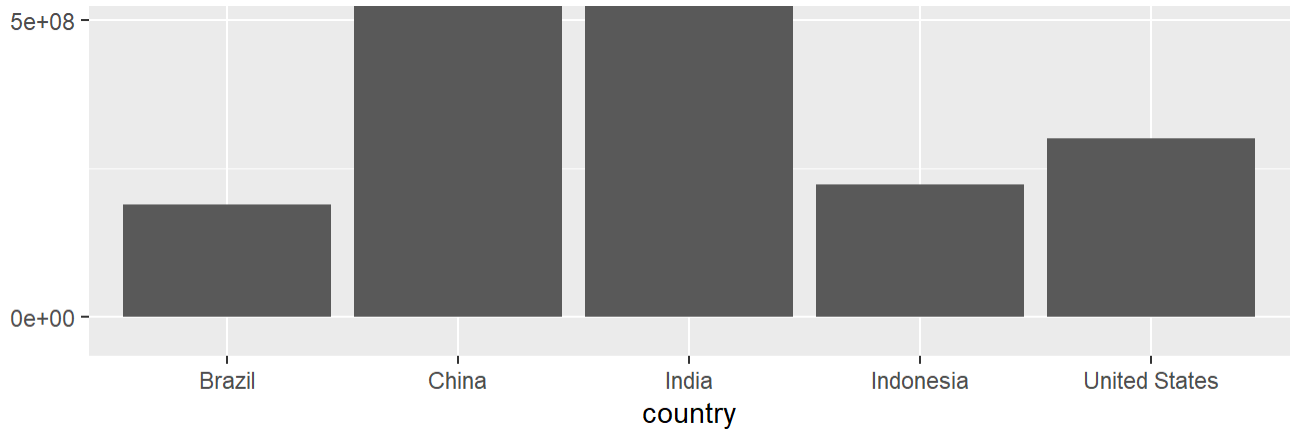
Bar charts

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

gapminder_top5
```

```
       country continent year lifeExp        pop gdpPercap
1        China      Asia 2007  72.961 1318683096  4959.115
2        India      Asia 2007  64.698 1110396331  2452.210
3 United States  Americas 2007  78.242  301139947 42951.653
4    Indonesia      Asia 2007  70.650  223547000  3540.652
5       Brazil  Americas 2007  72.390  190010647  9065.801
```

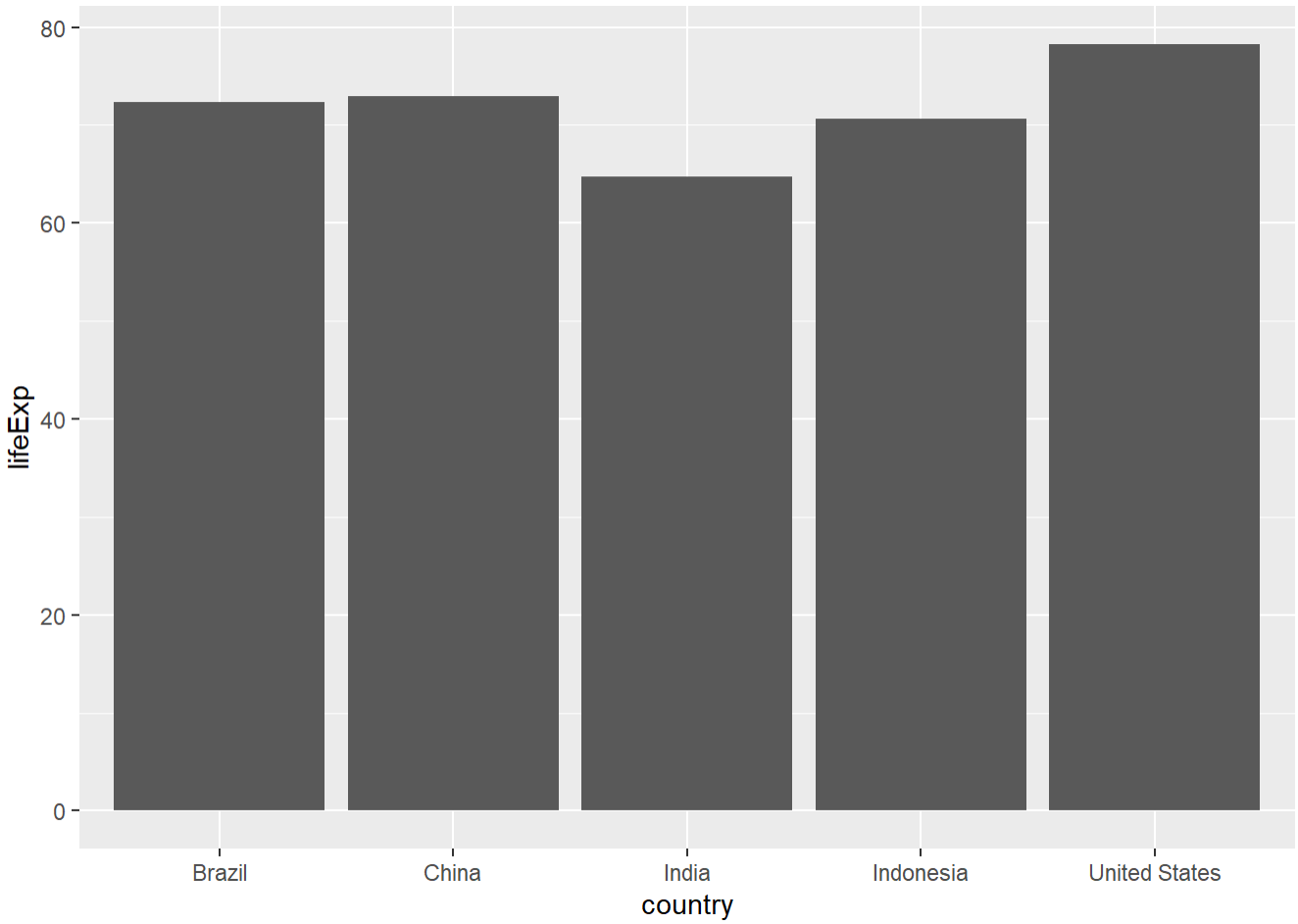
```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop))
```



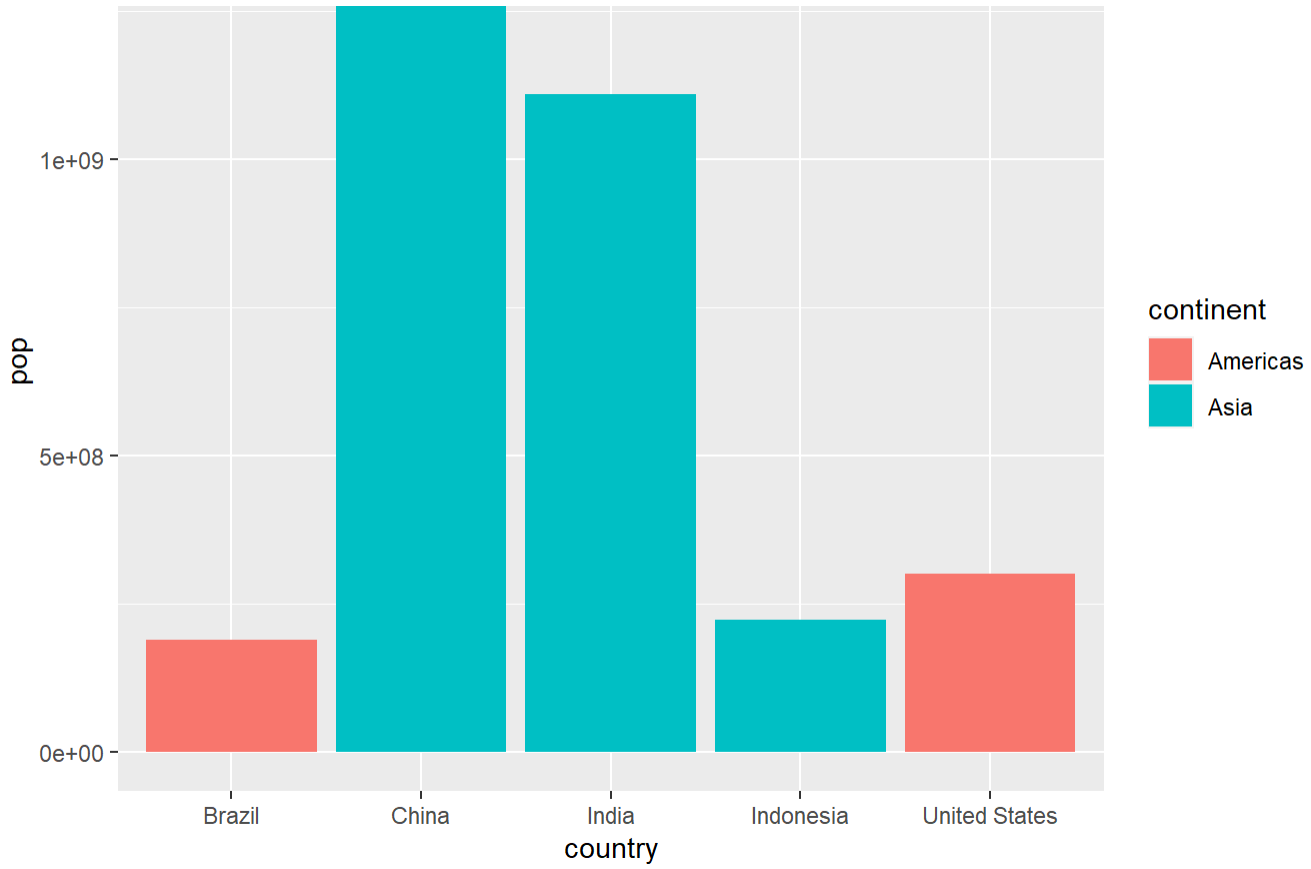


Q Create a bar chart showing the life expectancy of the five biggest countries by population in 2007.

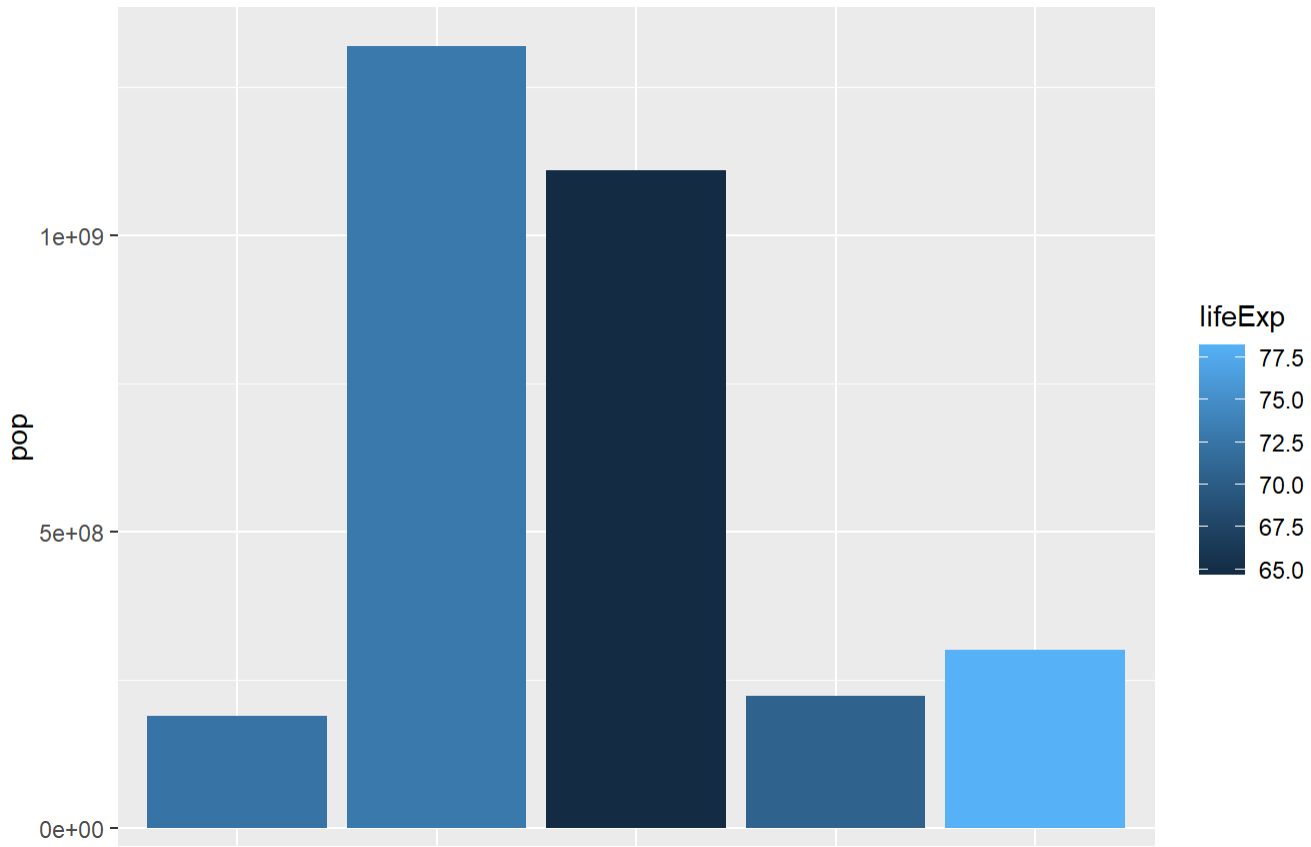
```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = lifeExp))
```




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

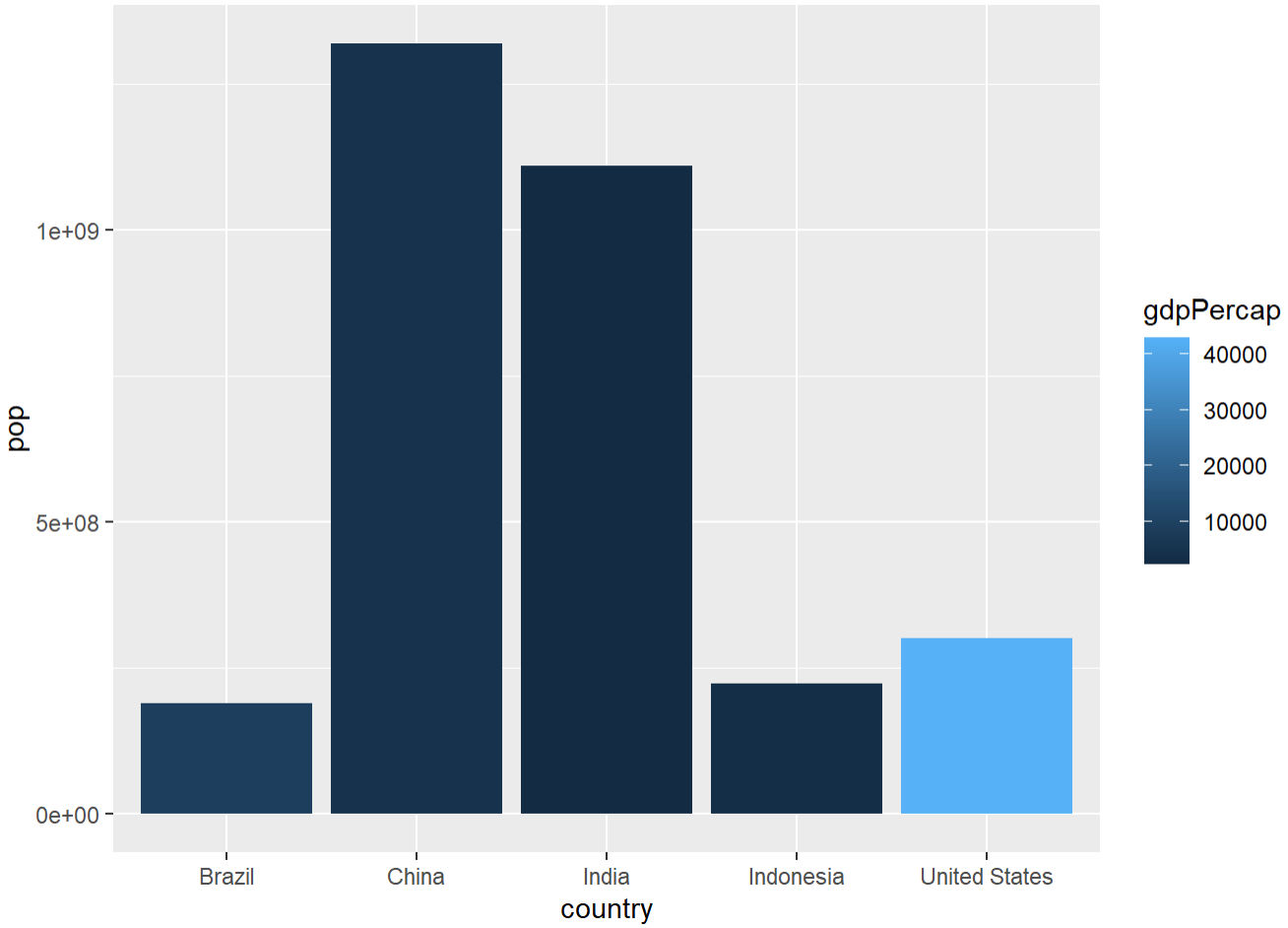



```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop, fill = lifeExp))
```



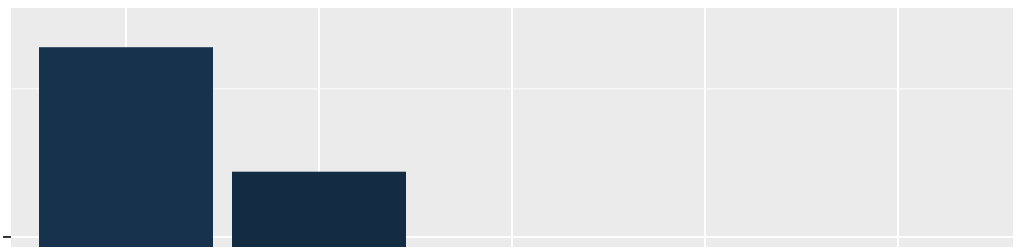
Q. Plot population size by country. Create a bar chart showing the population (in millions) of the five biggest countries by population in 2007.

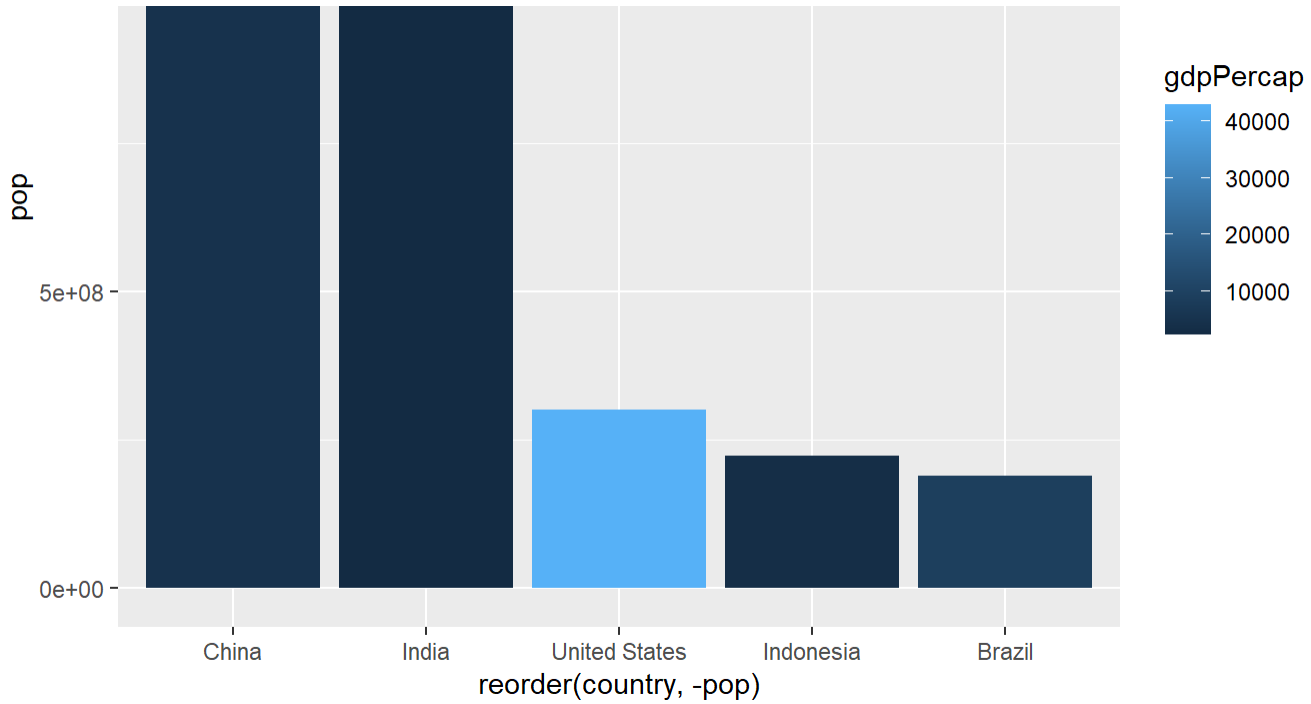
```
ggplot(gapminder_top5) +
  aes(x=country, y=pop, fill=gdpPercap) +
  geom_col()
```



and change the order of the bars

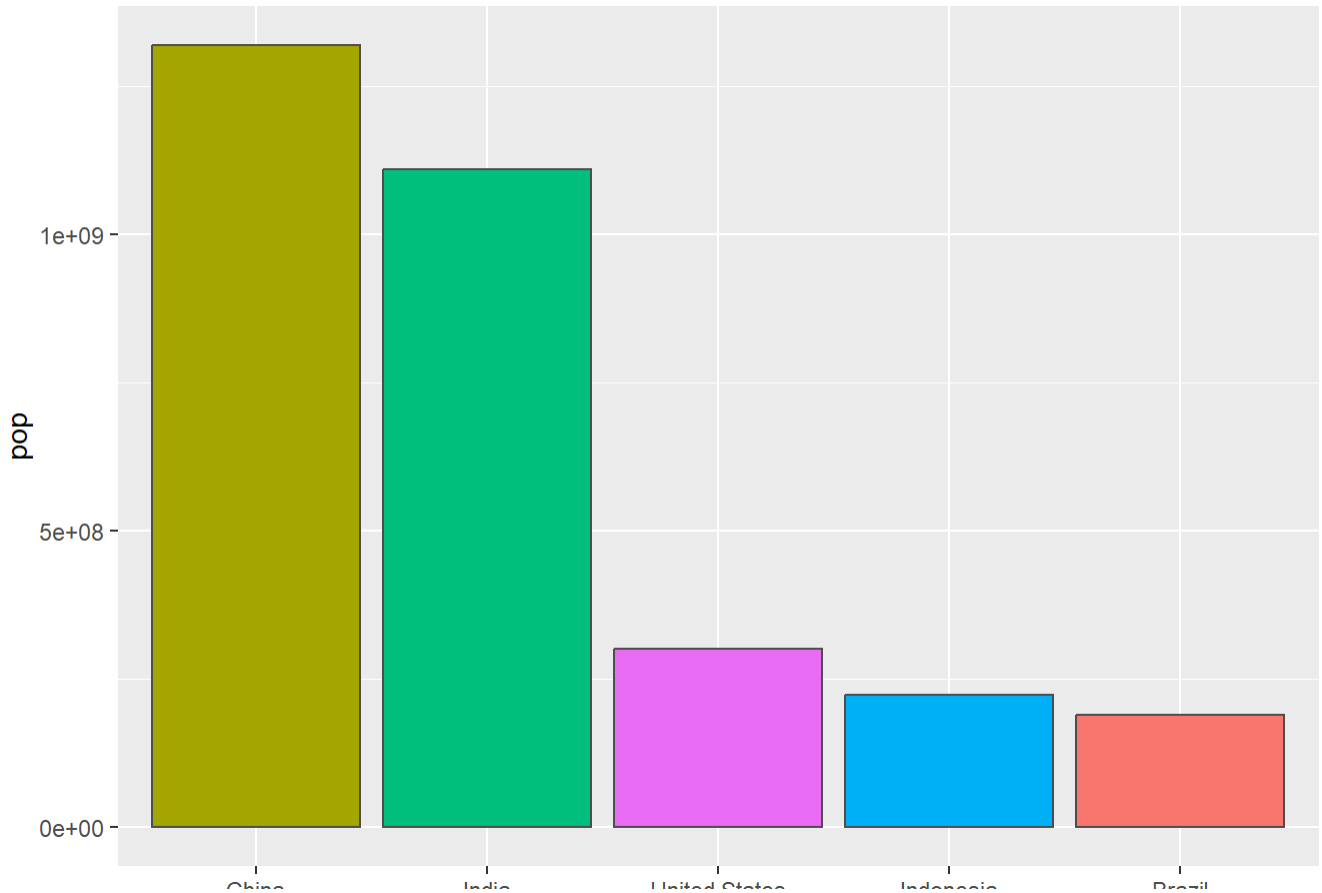
```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=gdpPercap) +
  geom_col()
```

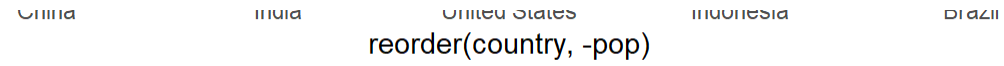




and just fill by country

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



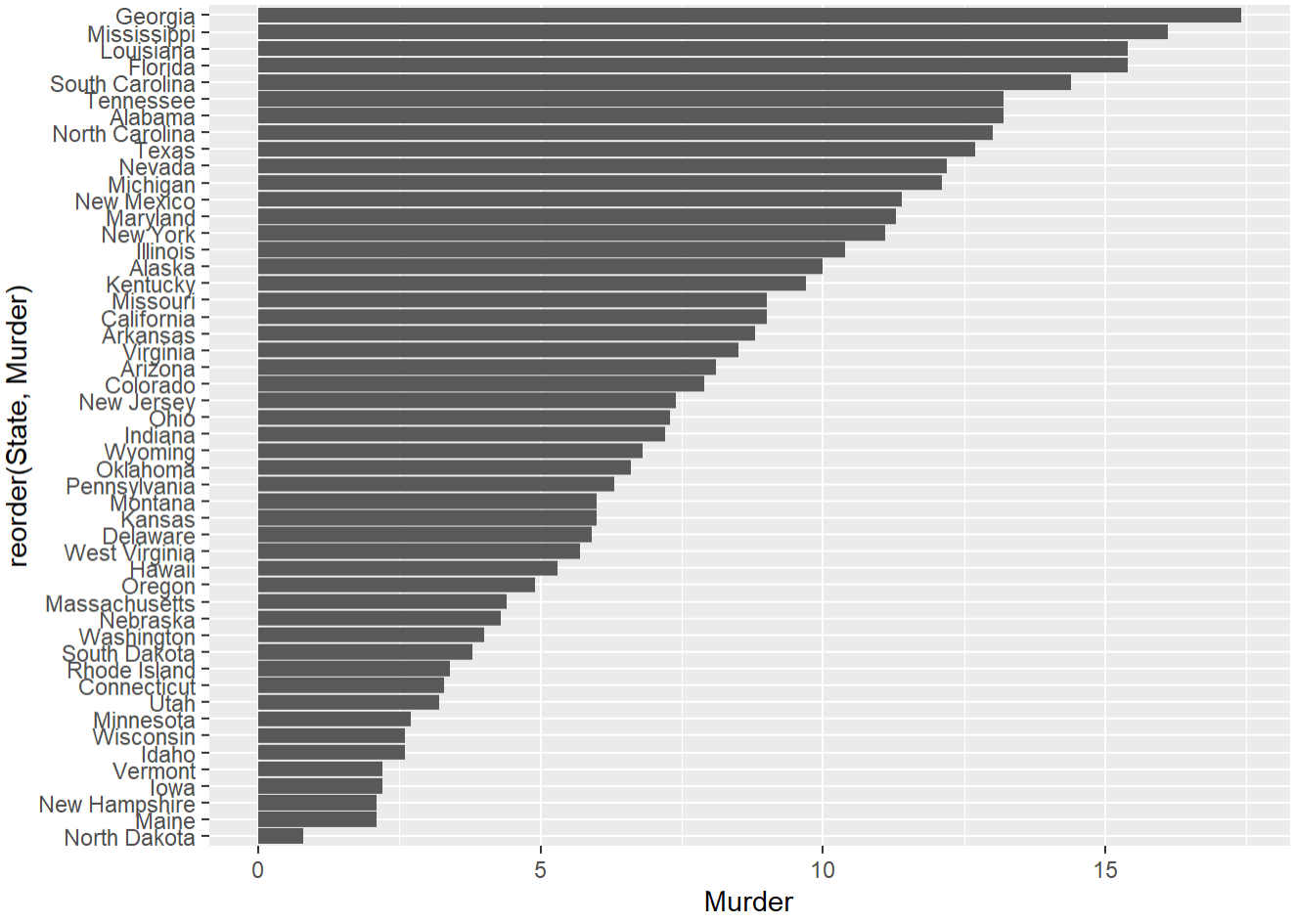


Flipping bar charts

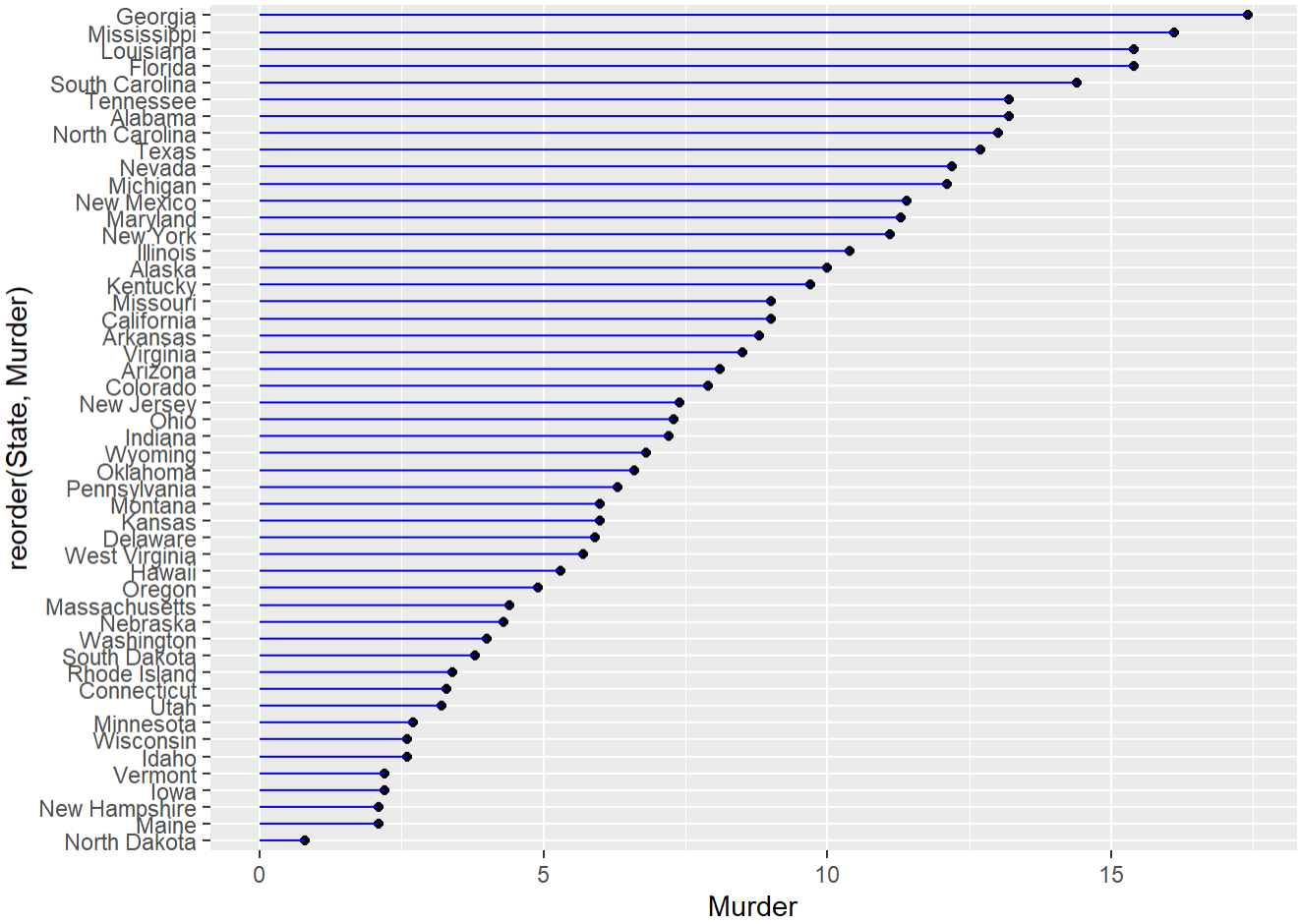
```
head(USArrests)
```

	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()
```



```
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()
```



Extensions: Animation

```
library(gapminder)
```

```
Attaching package: 'gapminder'

The following object is masked _by_ '.GlobalEnv':

    gapminder
```

```
library(gganimate)
```

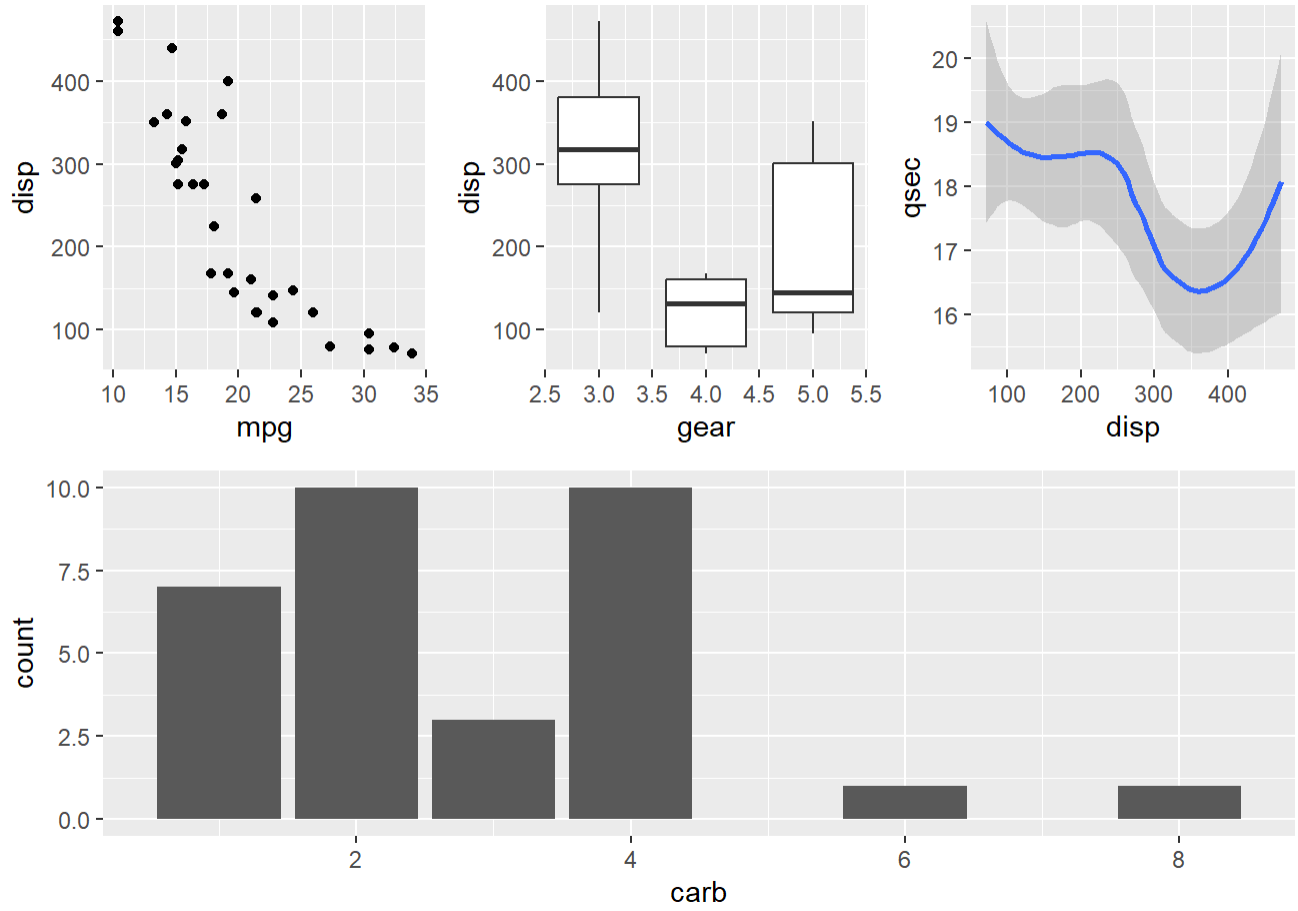
```
# Setup nice regular ggplot of the gapminder data
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  # Facet by continent
  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)
```

Combining plots

```
library(patchwork)

# Setup some example plots
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(disp, qsec))
```

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

# Use patchwork to combine them here:
(p1 | p2 | p3) /
      p4
```

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



About this document

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default


locale:
[1] LC_COLLATE=English_United States.utf8
```

```
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] patchwork_1.2.0 gganimate_1.0.8 gapminder_1.0.0 dplyr_1.1.4
[5] ggplot2_3.4.4

loaded via a namespace (and not attached):
 [1] Matrix_1.6-1.1    gtable_0.3.4      jsonlite_1.8.8    crayon_1.5.2
 [5] compiler_4.3.2    tidyselect_1.2.0  progress_1.2.3    splines_4.3.2
 [9] scales_1.3.0      yaml_2.3.8        fastmap_1.1.1     lattice_0.21-9
[13] R6_2.5.1          labeling_0.4.3    generics_0.1.3    knitr_1.45
[17] tibble_3.2.1      munsell_0.5.0     pillar_1.9.0      rlang_1.1.3
[21] utf8_1.2.4        stringi_1.8.3     xfun_0.41         cli_3.6.2
[25] tweenr_2.0.2      withr_3.0.0       magrittr_2.0.3    mgcv_1.9-0
[29] digest_0.6.34     grid_4.3.2        hms_1.1.3         lifecycle_1.0.4
[33] nlme_3.1-163      prettyunits_1.2.0 vctrs_0.6.5       evaluate_0.23
[37] glue_1.7.0        farver_2.1.1      gifski_1.12.0-2   fansi_1.0.6
[41] colorspace_2.1-0  rmarkdown_2.25    tools_4.3.2       pkgconfig_2.0.3
[45] htmltools_0.5.7
```