Class 05: Data Visualization with GGPLOT

Changcheng Li(A69027828)

##Using ggplot

To use ggplot2 we first need to install it on our computers. To do this we will use the function 'install.packages()'

Before I use any package functions I have to load them up with library()

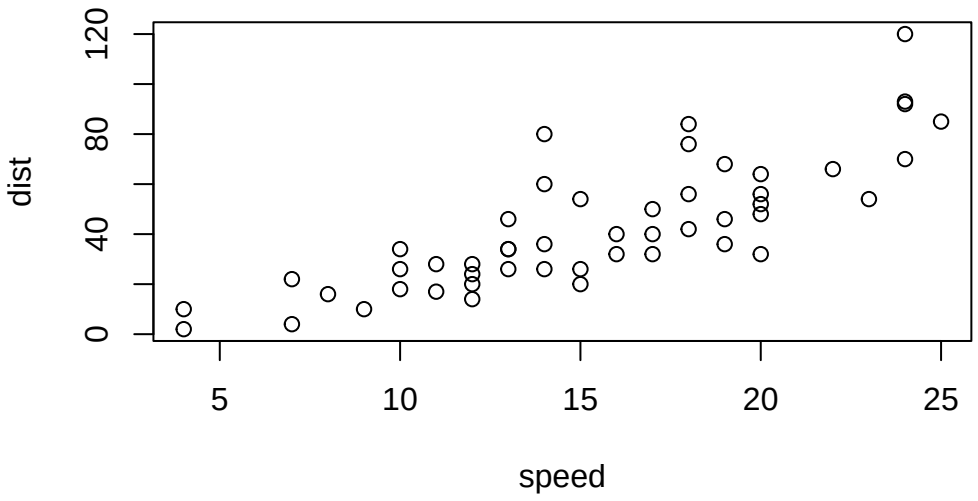
```
library(ggplot2)
ggplot(cars)
```



```
head(cars)
```

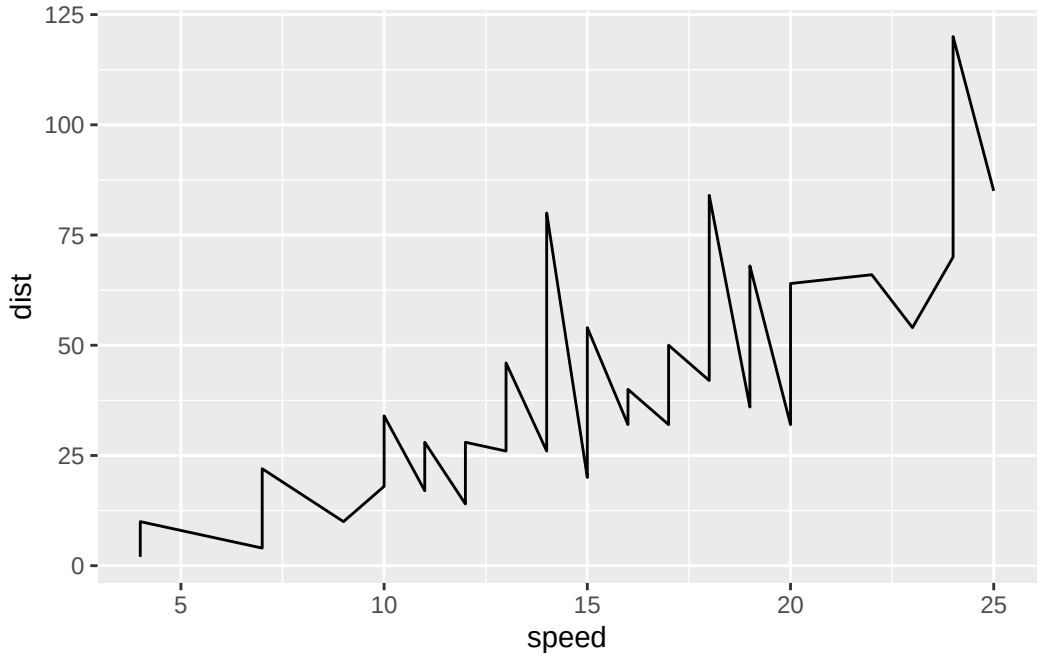
```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

```
plot(cars)
```



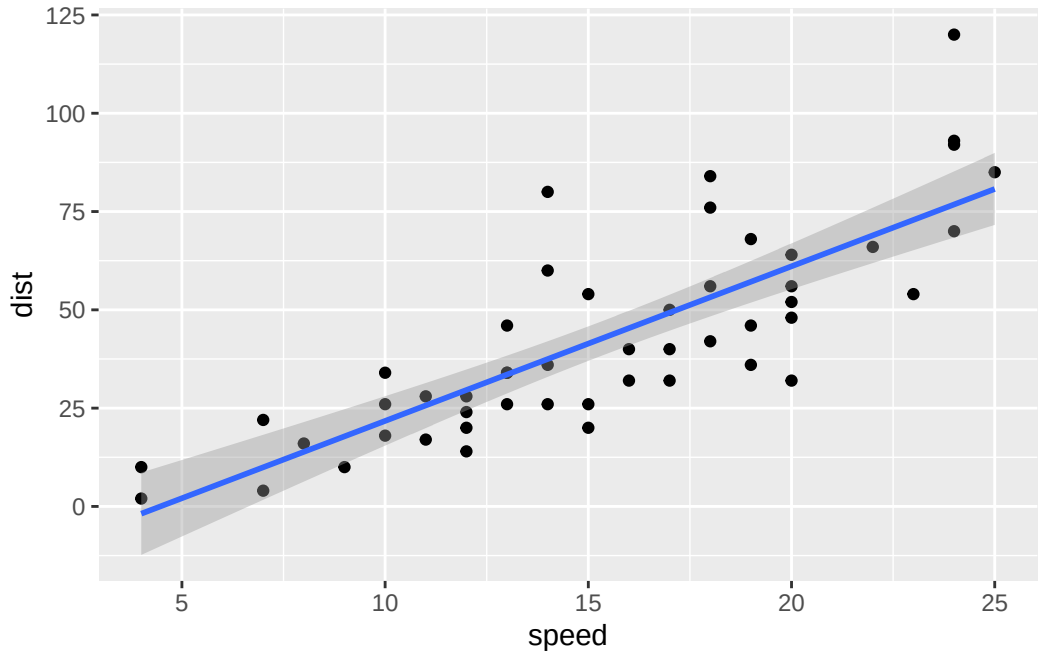
To use ggplot I need to spell out at least 3 things: - data(the stuff I want to plot as a data.frame) - aesthetics (aes() values - how the data map to the plot) - geoms (how I want things drown)

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



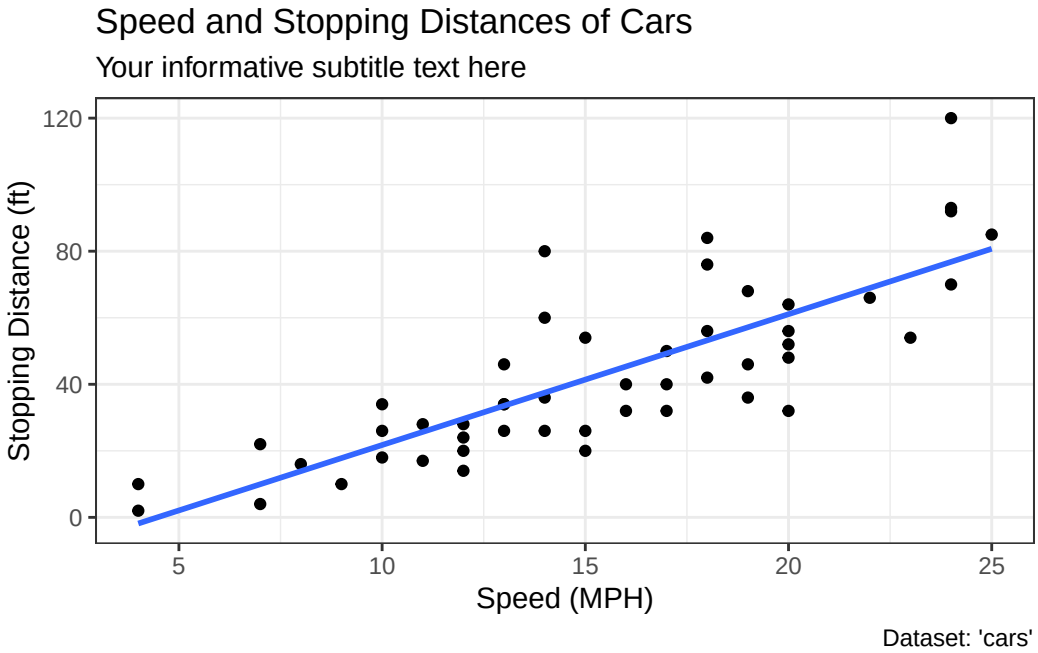

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm")
```

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




```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  labs(title="Speed and Stopping Distances of Cars",
       x="Speed (MPH)",
       y="Stopping Distance (ft)",
       subtitle = "Your informative subtitle text here",
       caption="Dataset: 'cars'") +
  geom_smooth(method="lm", se=FALSE) +
  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. How many genes are there?

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

Q. How many are "up"

```
sum(genes$State == "up")
```

```
[1] 127
```

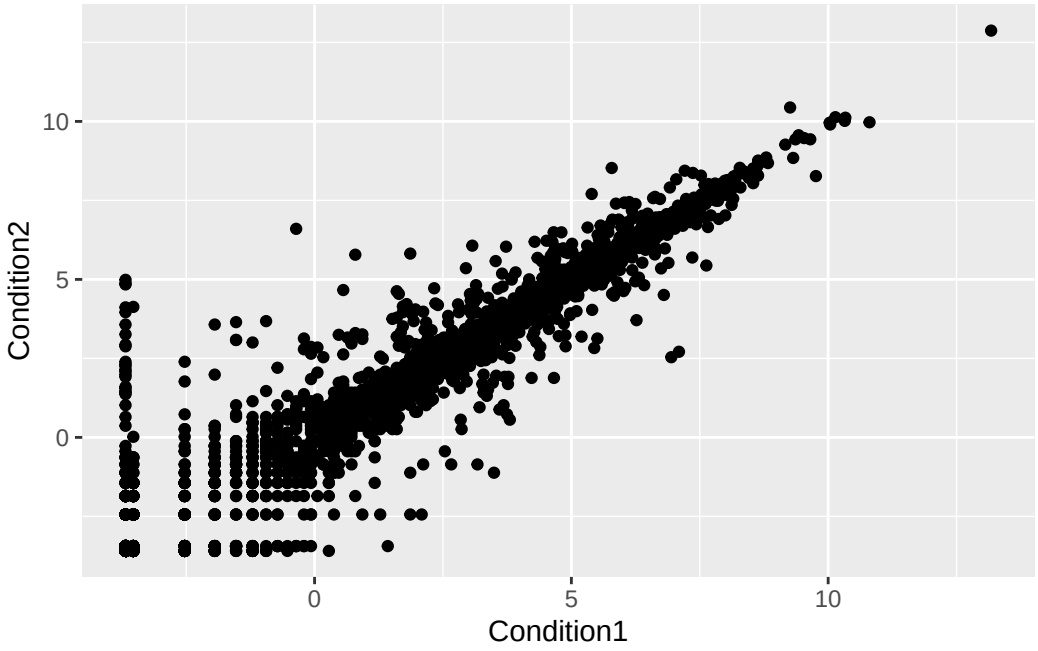
```
table(genes$State)
```

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

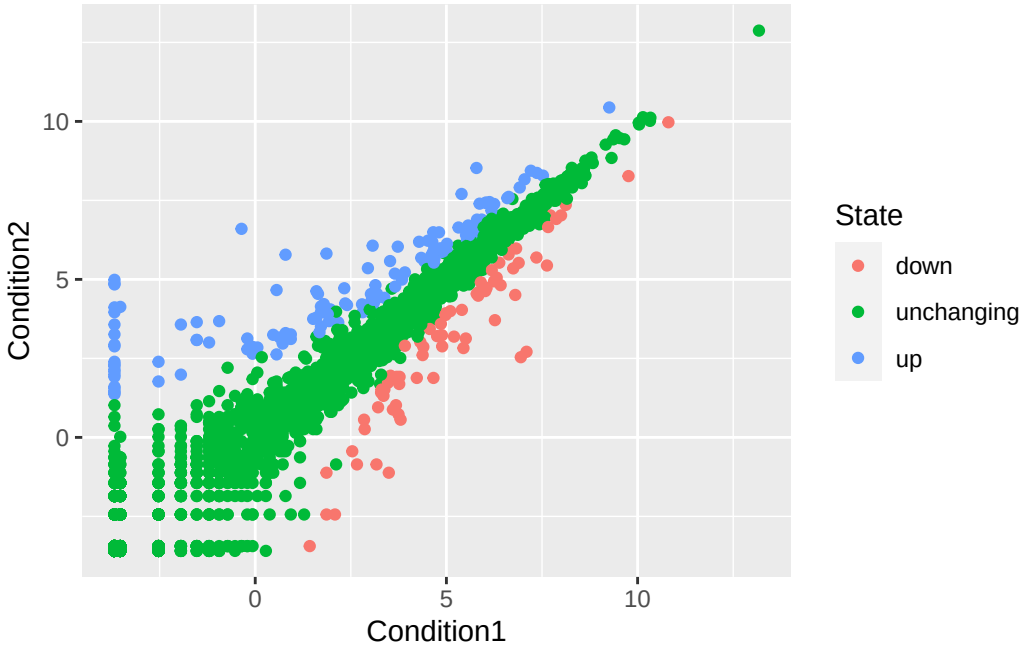
```
round( table(genes$State)/nrow(genes) * 100, 2 )
```

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

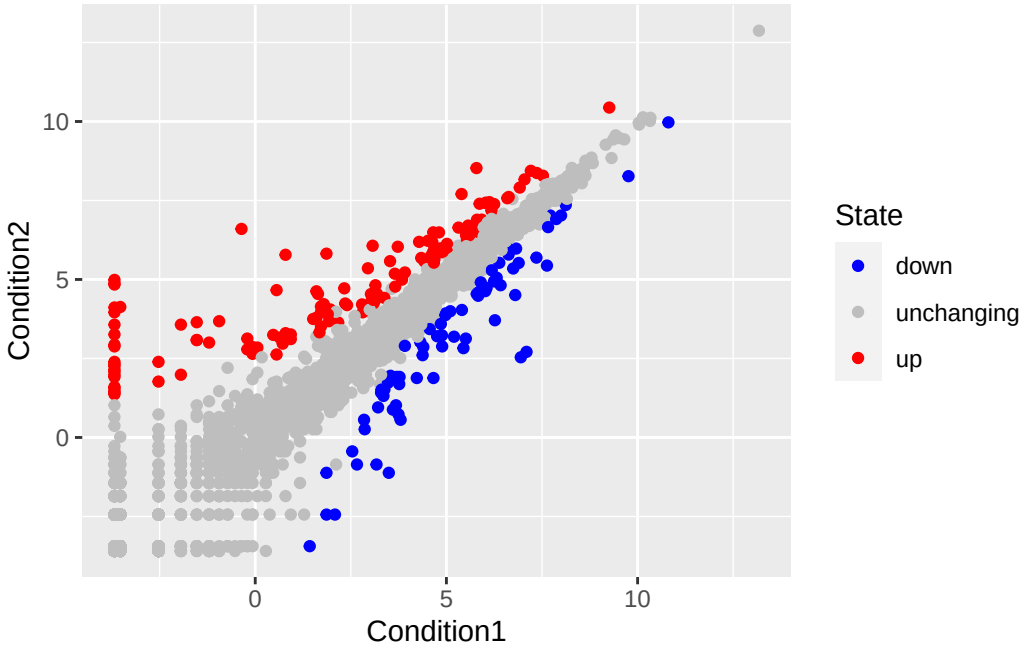
```
ggplot(genes) +
    aes(x=Condition1, y=Condition2) +
    geom_point()
```



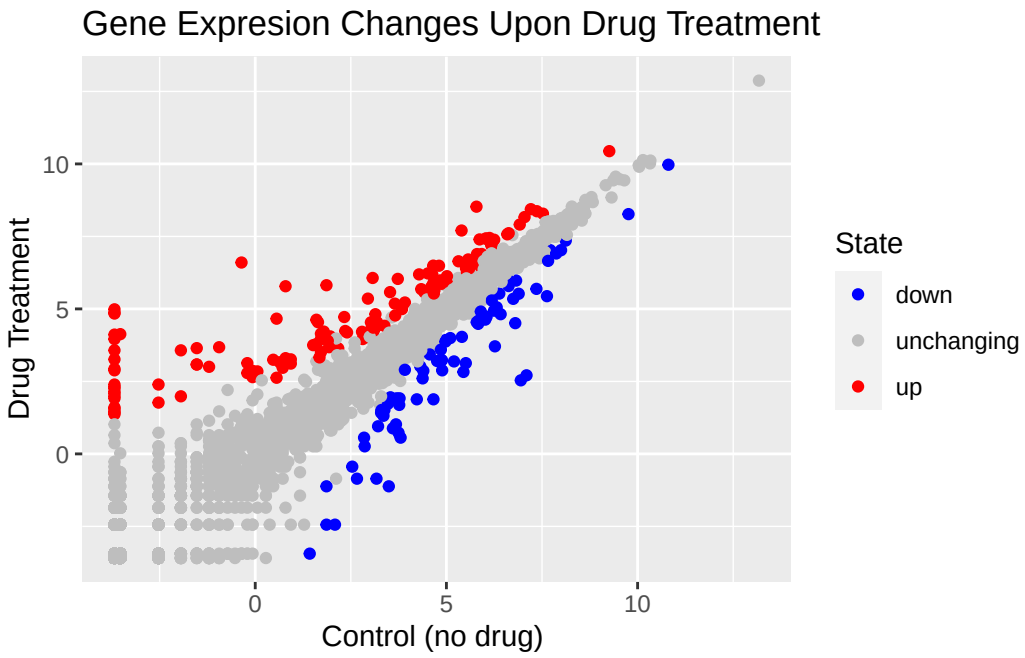

```
p <- ggplot(genes) +
    aes(x=Condition1, y=Condition2, col=State) +
    geom_point()
p
```




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



```
p + scale_colour_manual(values=c("blue","gray","red")) +
    labs(title="Gene Expresion Changes Upon Drug Treatment",
         x="Control (no drug) ",
         y="Drug Treatment")
```




```
# File location online
library(gapminder)
```

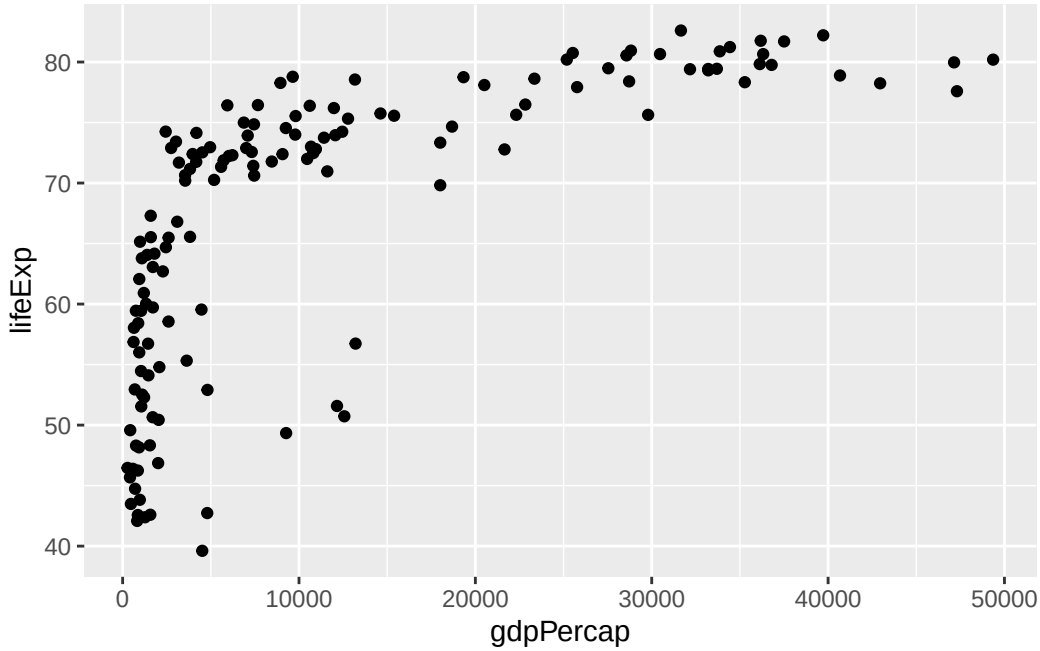
```
## un-comment to install if needed
library(dplyr)
```

```
Attaching package: 'dplyr'

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

    filter, lag

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

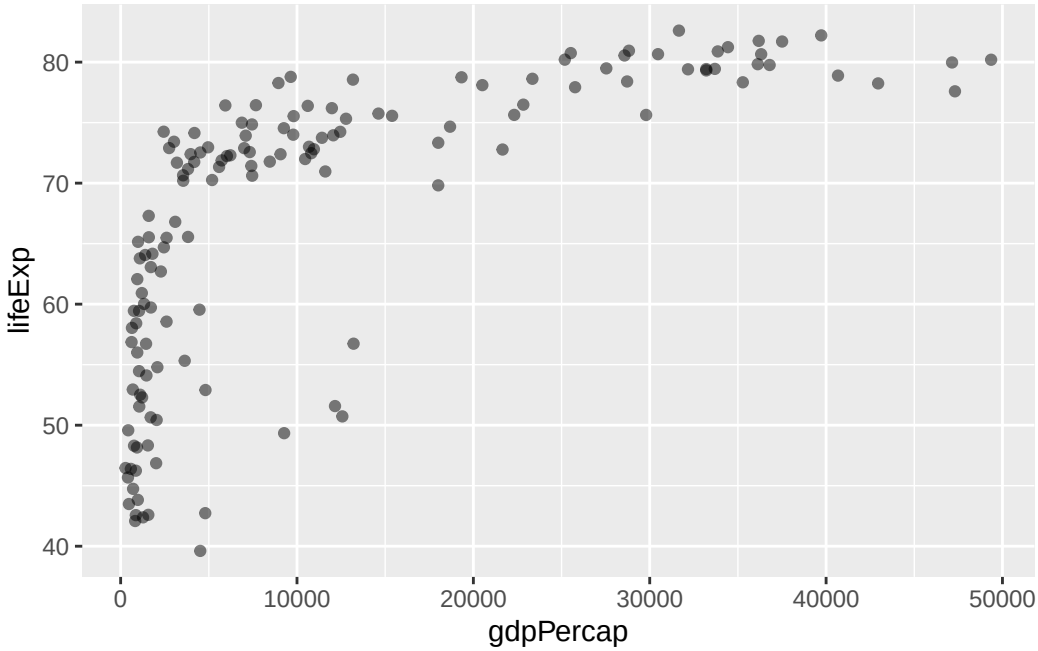
```
  intersect, setdiff, setequal, union
```

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

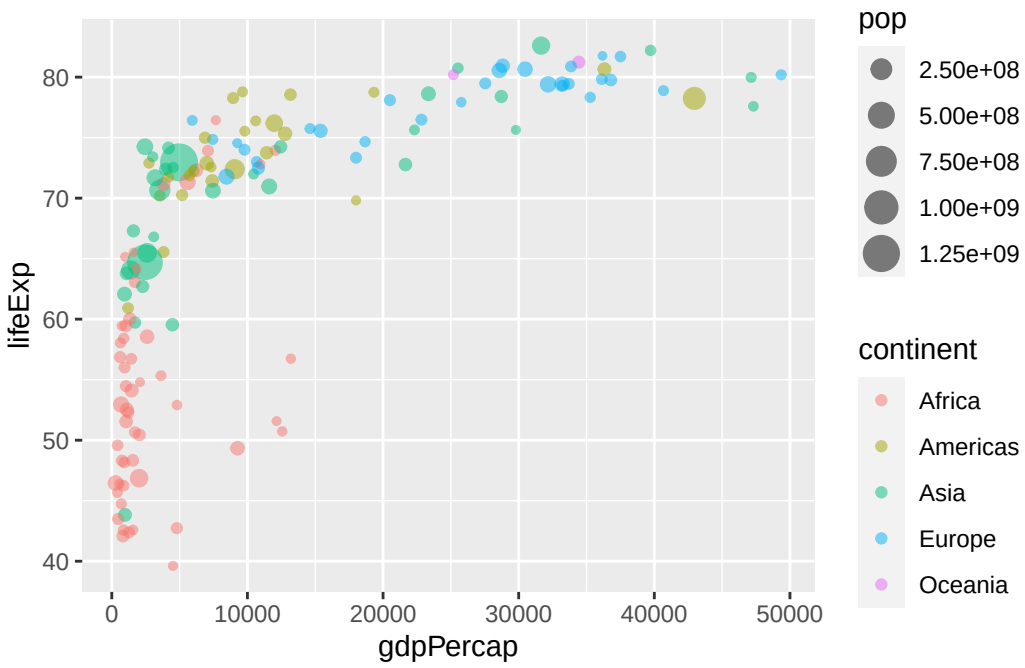
```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point()
```



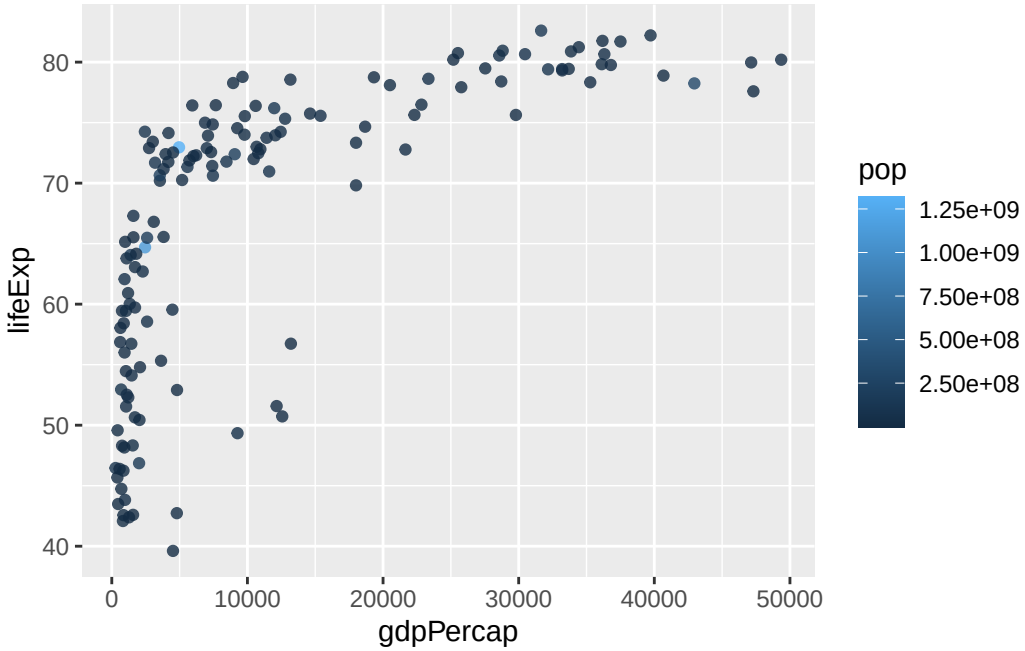

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```



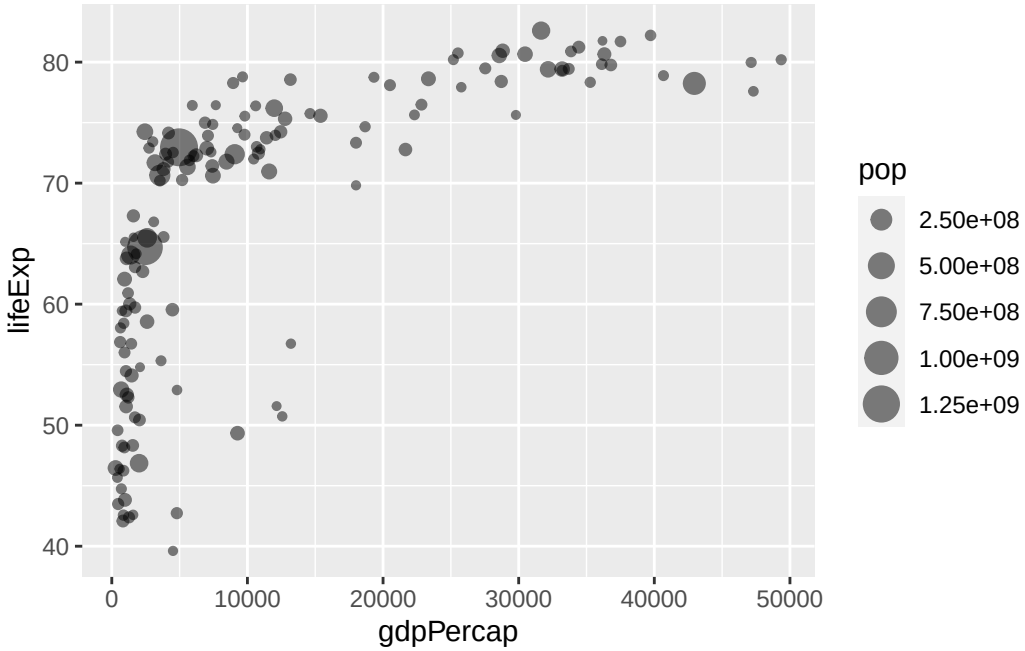

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```



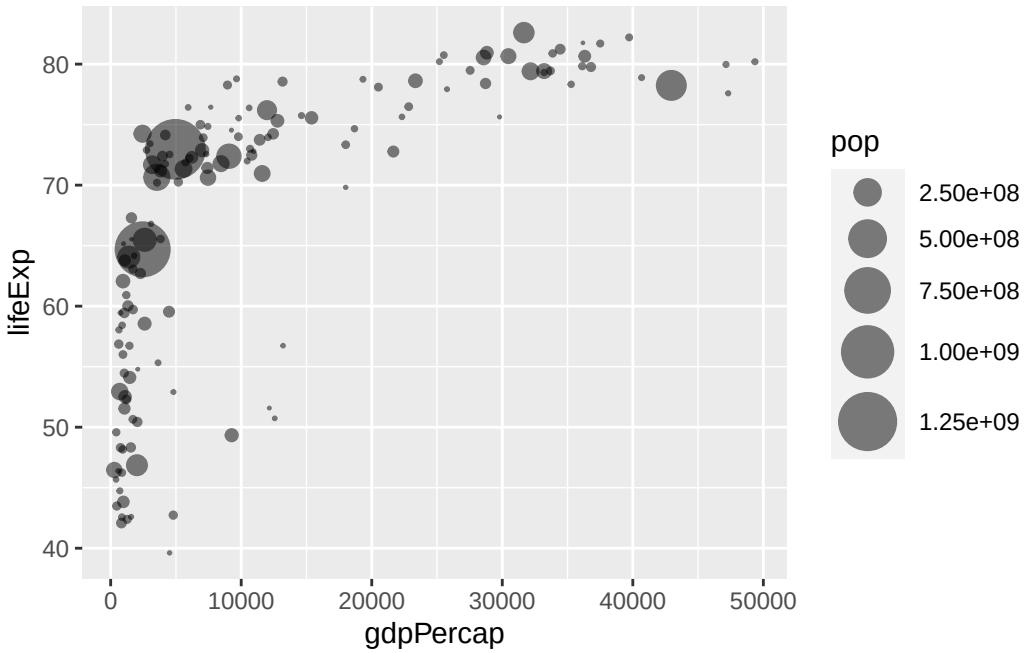

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



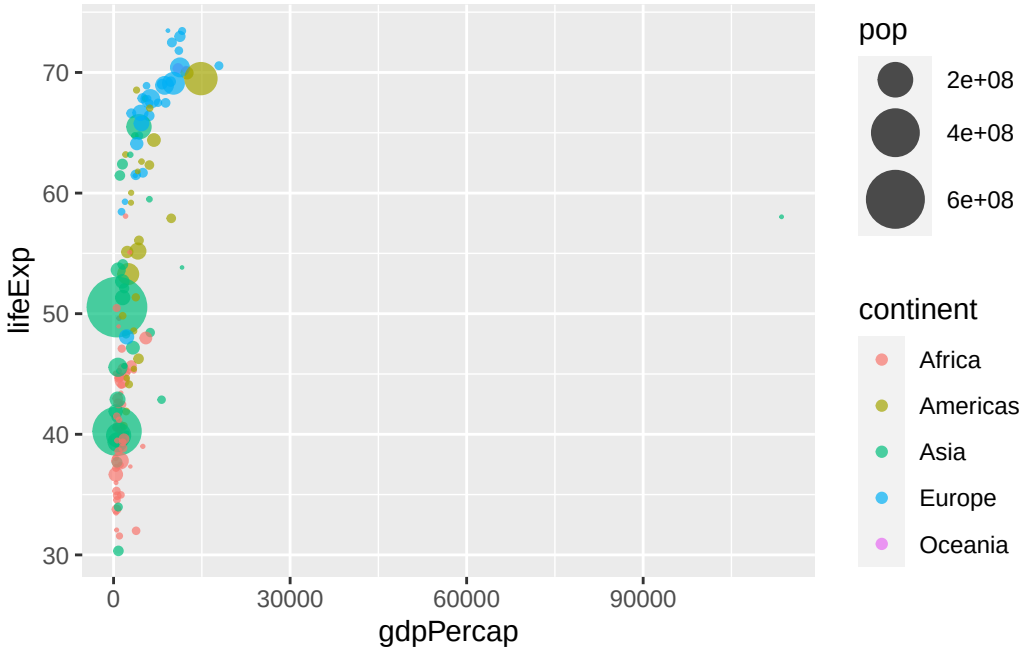

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```



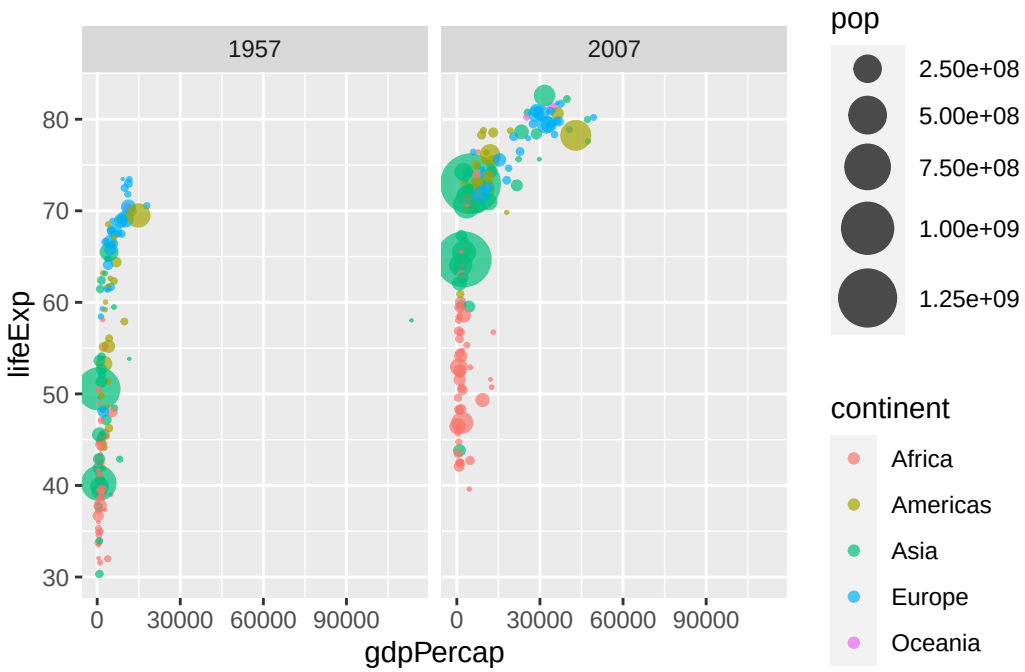

```
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop), alpha=0.5) +
  scale_size_area(max_size = 10)
```




```
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)
```



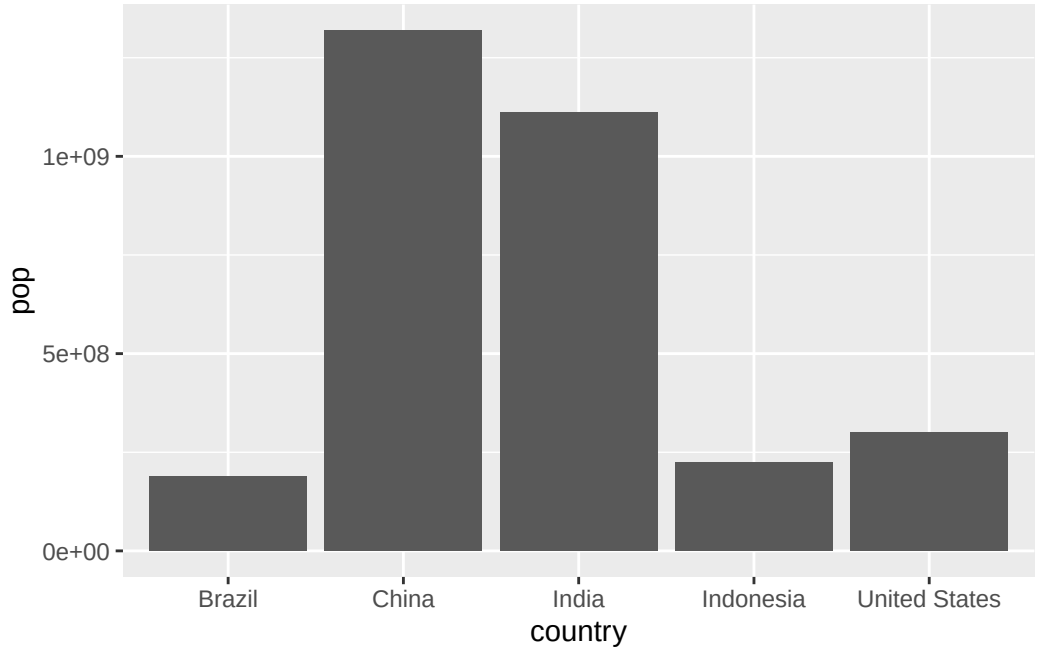

```
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)

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)
```



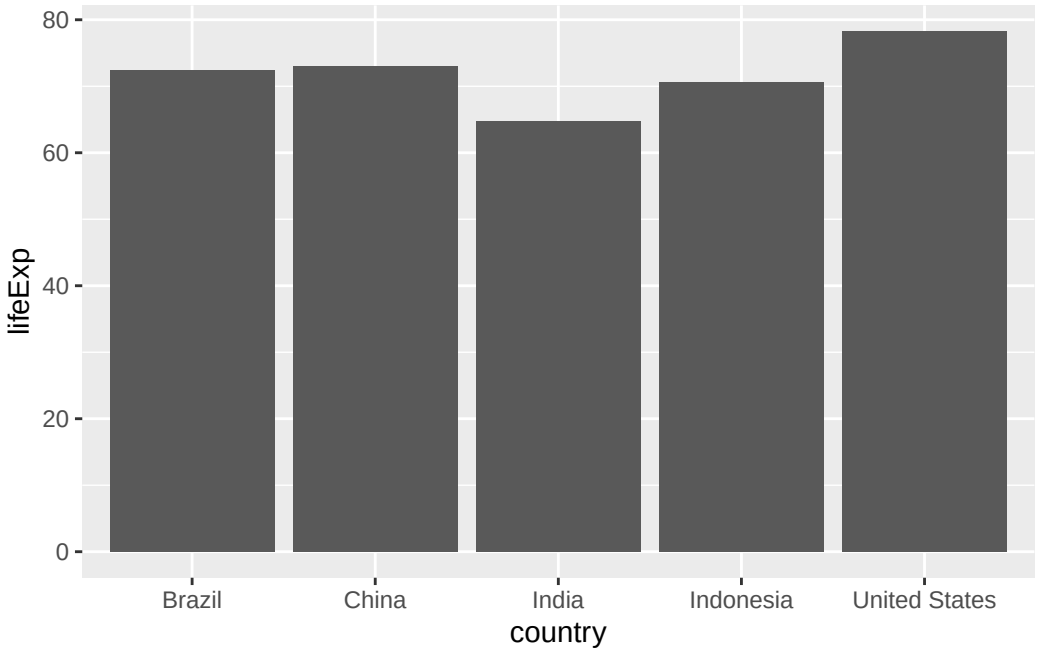

```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

gapminder_top5
```

```
# A tibble: 5 x 6
  country       continent  year lifeExp        pop gdpPercap
  <fct>         <fct>     <int>   <dbl>      <int>     <dbl>
1 China         Asia       2007    73.0 1318683096     4959.
2 India         Asia       2007    64.7 1110396331     2452.
3 United States Americas   2007    78.2  301139947    42952.
4 Indonesia     Asia       2007    70.6  223547000     3541.
5 Brazil        Americas   2007    72.4  190010647     9066.
```

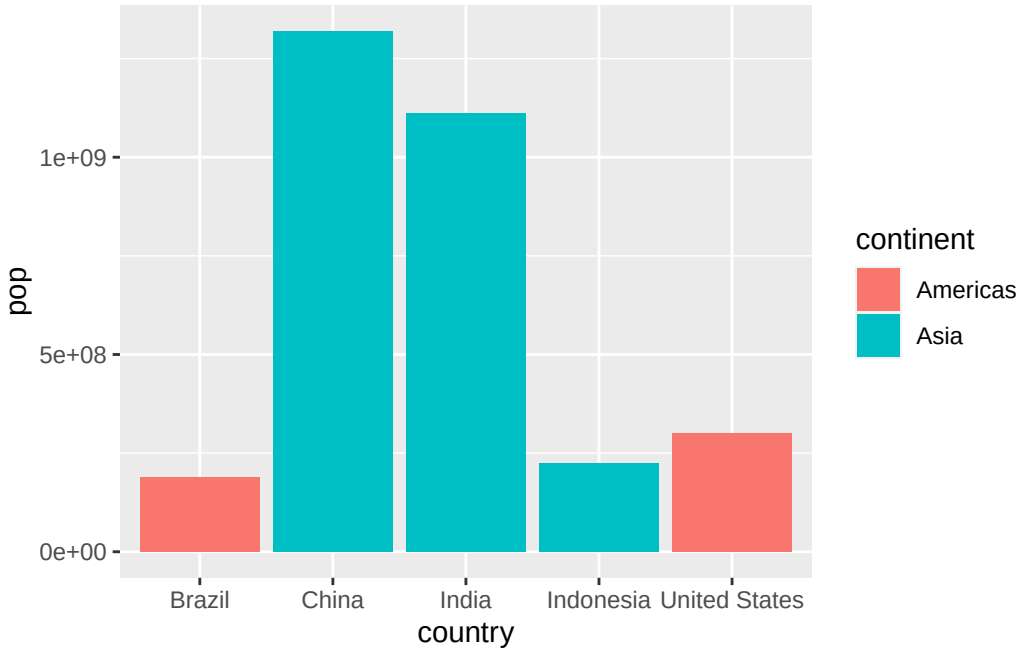
```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop))
```



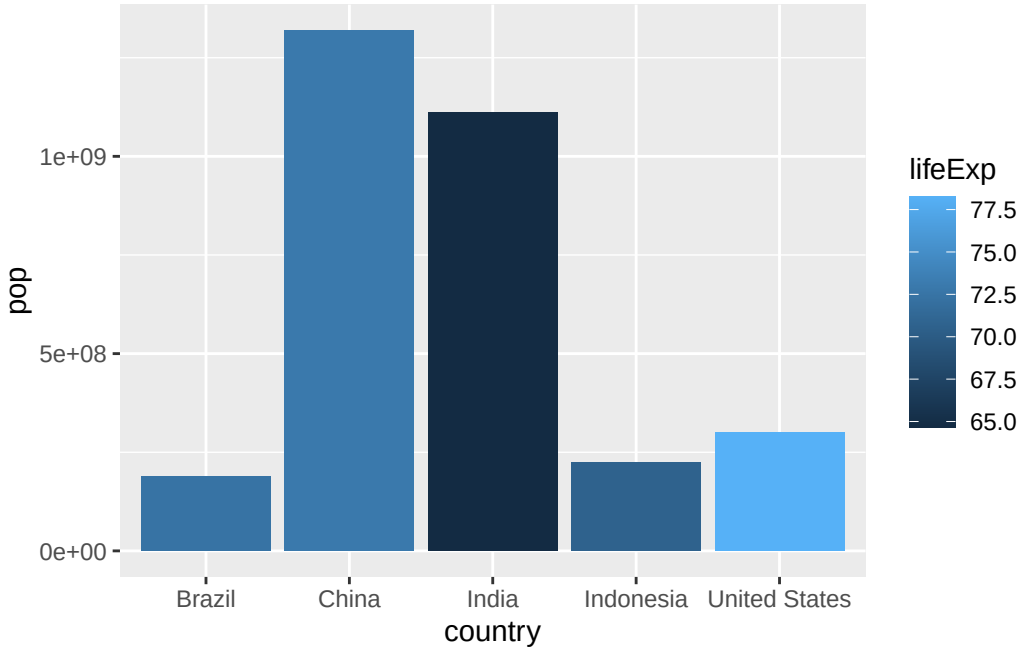

```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = lifeExp))
```



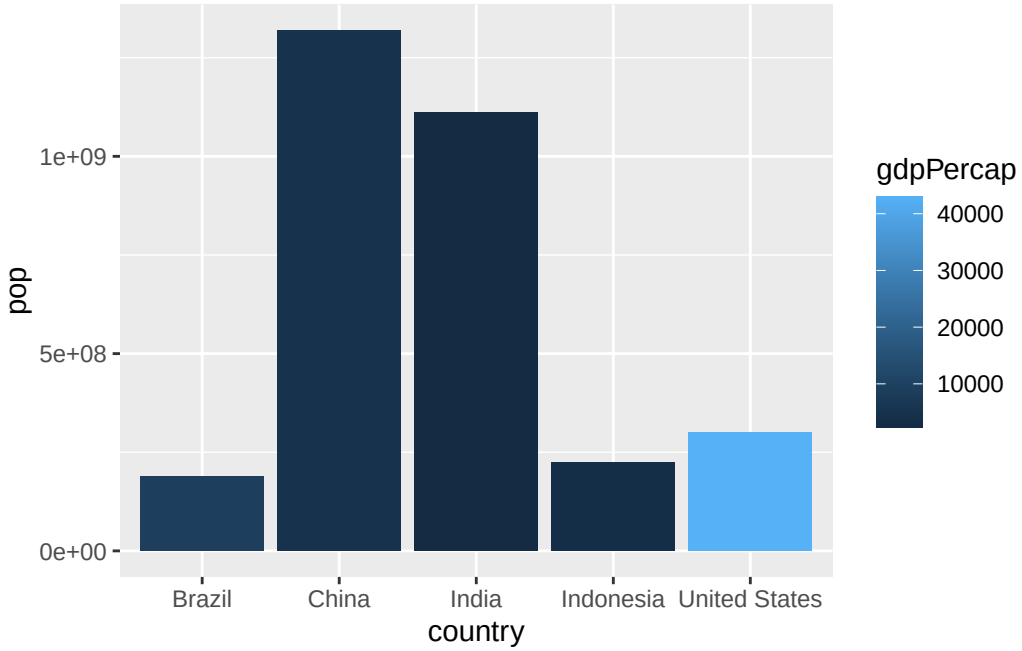
```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop, fill = continent))
```



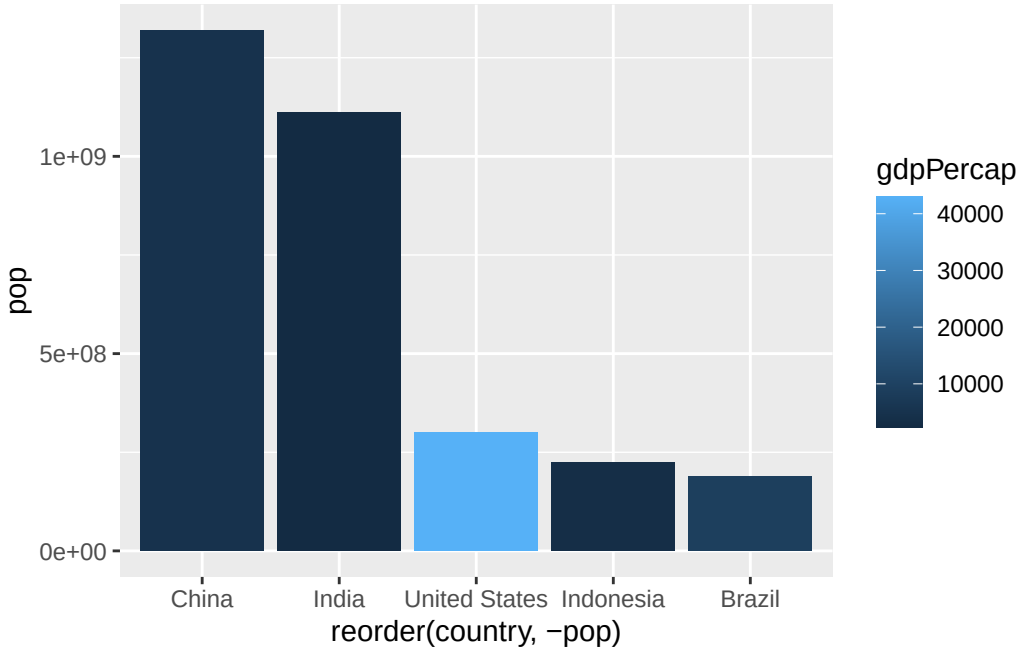

```
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop, fill = lifeExp))
```



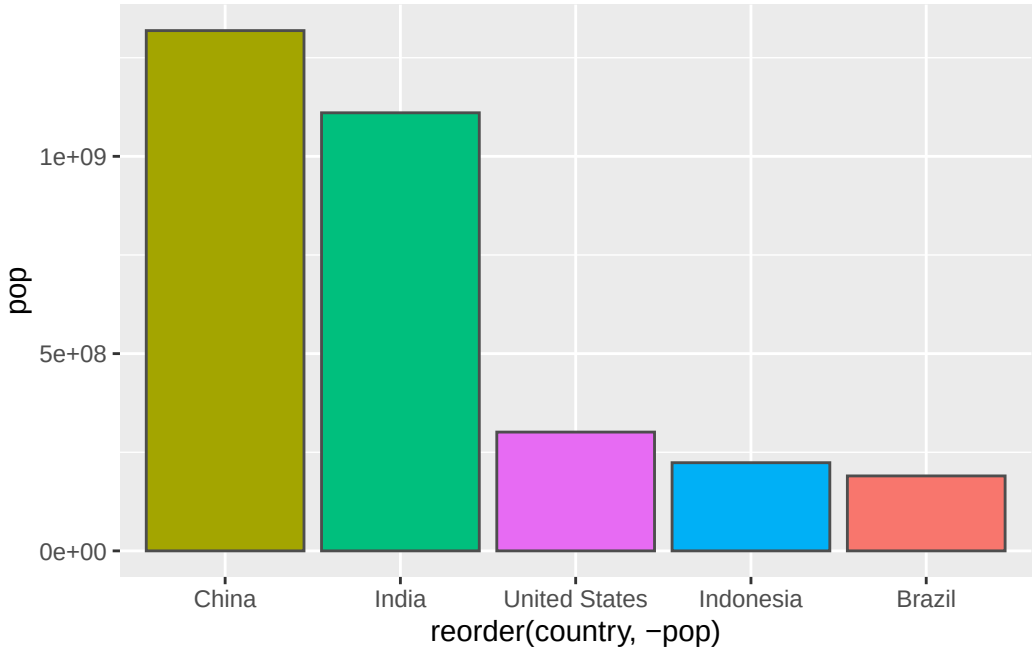

```
ggplot(gapminder_top5) +
  aes(x=country, y=pop, fill=gdpPercap) +
  geom_col()
```




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



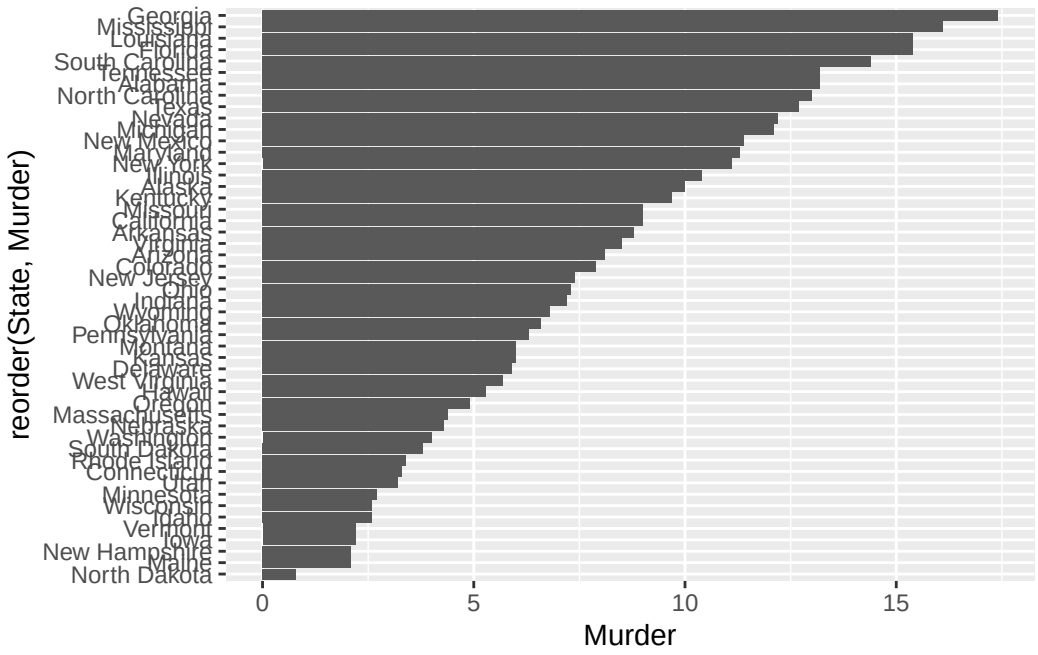

```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=country) +
  geom_col(col="gray30") +
  guides(fill="none")
```



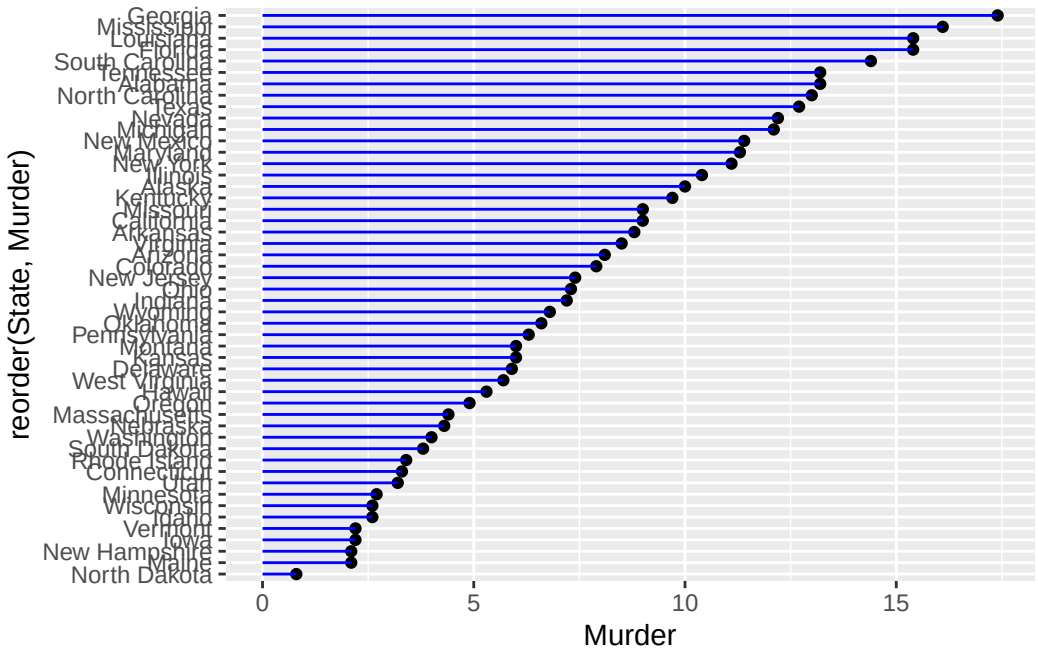

```
head(USArrests)
```

```
           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()
```

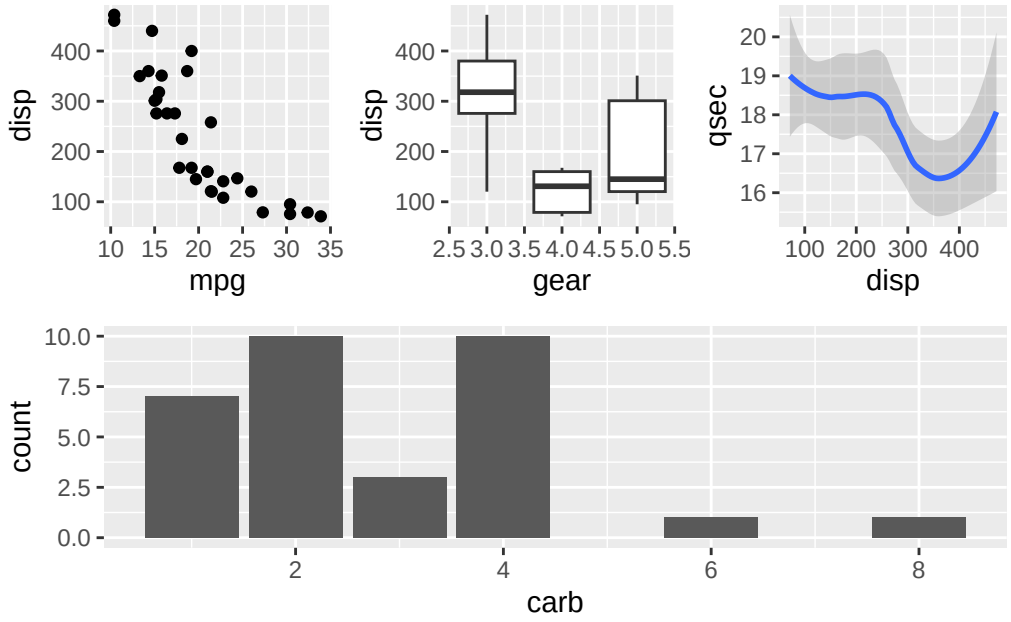



```
library(gapminder)
library(gganimate)
# Setup nice regular ggplot of the gapminder data
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10()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

```
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'
```




```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default


locale:
[1] LC_COLLATE=Chinese (Simplified)_China.utf8
[2] LC_CTYPE=Chinese (Simplified)_China.utf8
```

```
[3] LC_MONETARY=Chinese (Simplified)_China.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_China.utf8

time zone: America/Tijuana
tzcode source: internal

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

other attached packages:
[1] patchwork_1.1.3 dplyr_1.1.3     gapminder_1.0.0 ggplot2_3.4.4

loaded via a namespace (and not attached):
 [1] Matrix_1.5-4.1    gtable_0.3.4      jsonlite_1.8.7    compiler_4.3.1
 [5] tidyselect_1.2.0  splines_4.3.1     scales_1.2.1      yaml_2.3.7
 [9] fastmap_1.1.1     lattice_0.21-8    R6_2.5.1          labeling_0.4.3
[13] generics_0.1.3    knitr_1.44        tibble_3.2.1      munsell_0.5.0
[17] pillar_1.9.0      rlang_1.1.1       utf8_1.2.3        xfun_0.40
[21] cli_3.6.1         withr_2.5.1       magrittr_2.0.3    mgcv_1.8-42
[25] digest_0.6.33     grid_4.3.1        lifecycle_1.0.3   nlme_3.1-162
[29] vctrs_0.6.4       evaluate_0.22     glue_1.6.2        farver_2.1.1
[33] fansi_1.0.5       colorspace_2.1-0  rmarkdown_2.25    tools_4.3.1
[37] pkgconfig_2.0.3   htmltools_0.5.6.1
```

Quarto

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

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

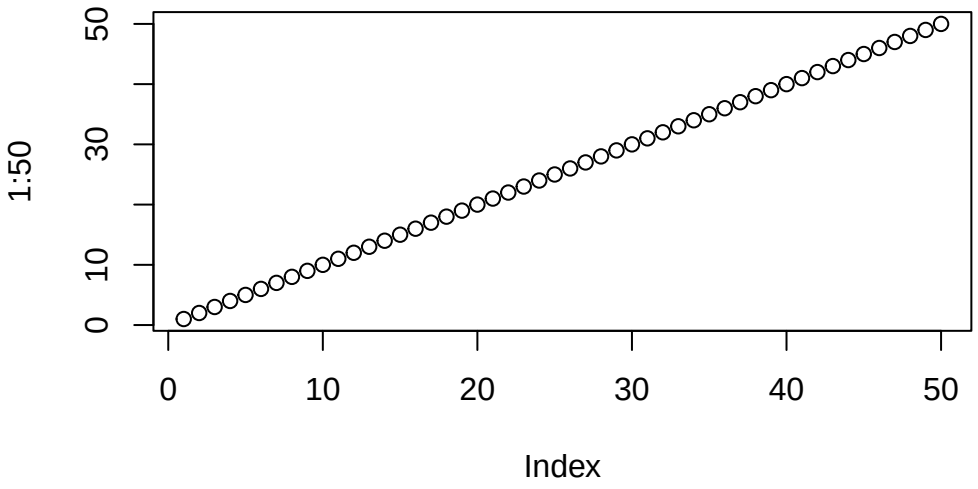
```
1 + 1
```

```
[1] 2
```

```
#This is text
```

You can add options to executable code like this

```
plot(1:50)
```




```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).