Class 5: Data Visualization with ggplot

Ellice Wang (PID: A16882742)

Intro to ggplot

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

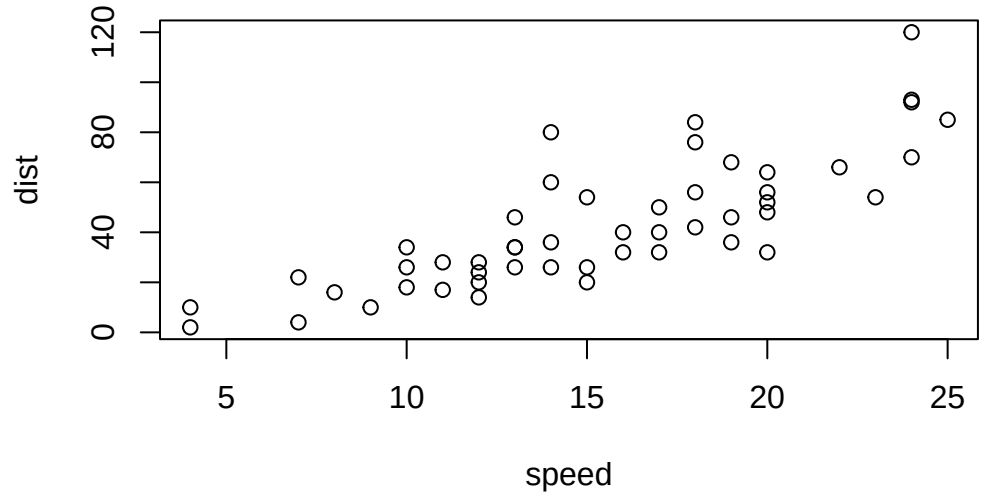
Let's start with a plot of a simple in-built dataset called 'cars'.

```
cars
```

```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
16    13   26
17    13   34
18    13   34
19    13   46
20    14   26
21    14   36
22    14   60
```

```
23    14   80
24    15   20
25    15   26
26    15   54
27    16   32
28    16   40
29    17   32
30    17   40
31    17   50
32    18   42
33    18   56
34    18   76
35    18   84
36    19   36
37    19   46
38    19   68
39    20   32
40    20   48
41    20   52
42    20   56
43    20   64
44    22   66
45    23   54
46    24   70
47    24   92
48    24   93
49    24  120
50    25   85
```

```
plot(cars)
```



Making this figure using **ggplot**. First need to install this package on my computer. To install any R package, I use function 'install.packages()'

I will run 'install.packages("ggplot2") in R console instead of this quarto document

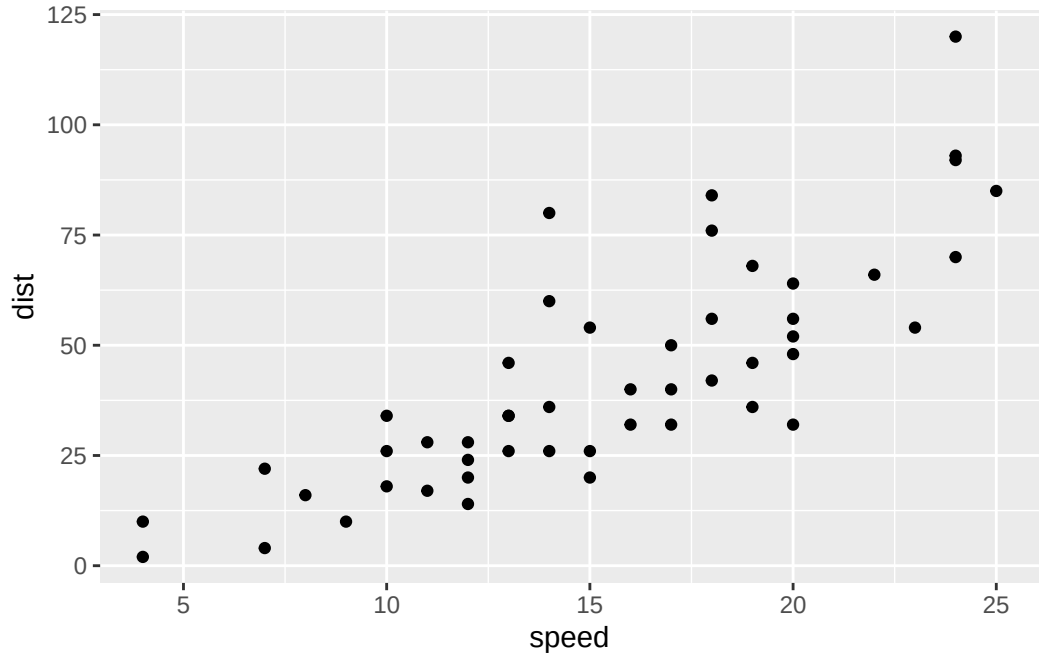
Before I can use any functions from add on packages I need to load the package from my "library()" with the 'library(ggplot2)' call

```
# load in ggplot
library(ggplot2)
```

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

- **data** (input datset 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)

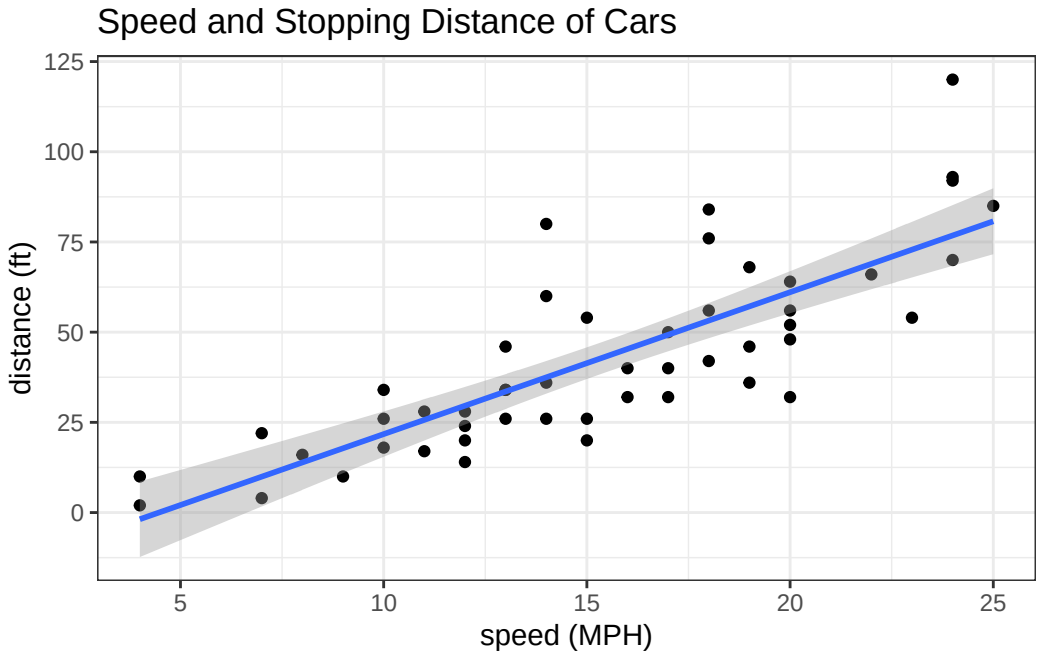
```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



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="Speed and Stopping Distance of Cars", x="speed (MPH)", y="distance (ft)")
```

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



Q1: For which phases is data visualization important in our scientific workflows? Communication of results, exploratory data analysis, and detection of outliars

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

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

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

Genes 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
```

How many genes are in this dataset? 5196 genes

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? 4 columns

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? 127 up regulated genes

Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset? 2.44

```
n.tot <- nrow(genes)
colnames(genes)
```

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

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

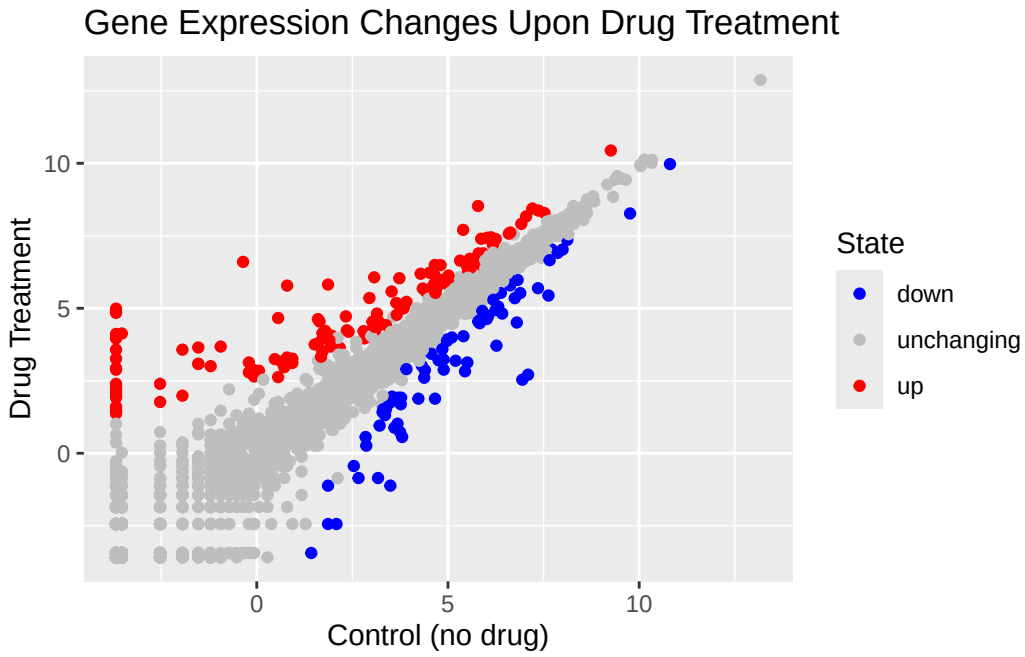
```
      down unchanging         up
      1.39      96.17       2.44
```

```
round(vals/n.tot*100, 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 (no drug)",
       y="Drug Treatment") +
  scale_color_manual(values=c("blue", "grey", "red"))
```



Gapminder dataset

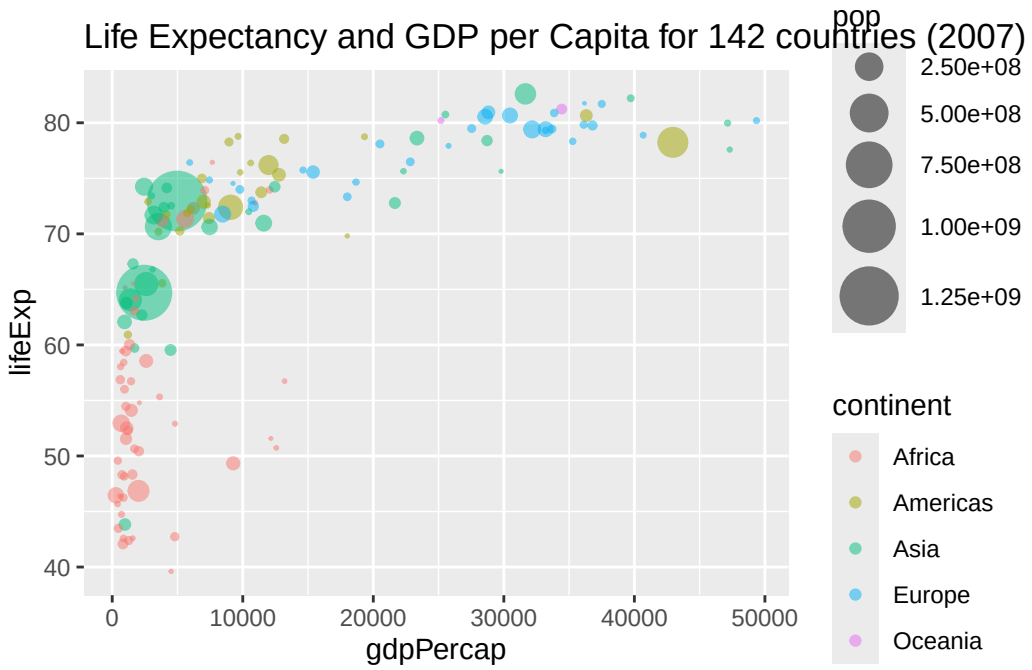
```
# load in dataset from url
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.tsv
gapminder <- read.delim(url)

# load in dplyr package
library(dplyr)
```

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

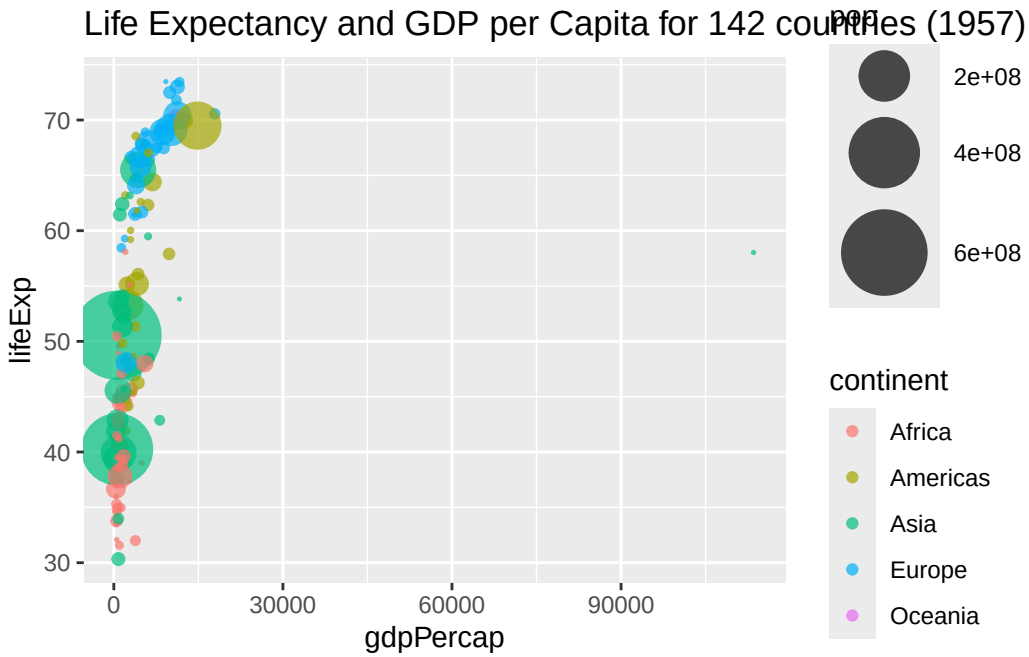
```
The following objects are masked from 'package:stats':

    filter, lag

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

    intersect, setdiff, setequal, union
```

plot Gapminder dataset from 2007 based on GDP per capita and life expectancy

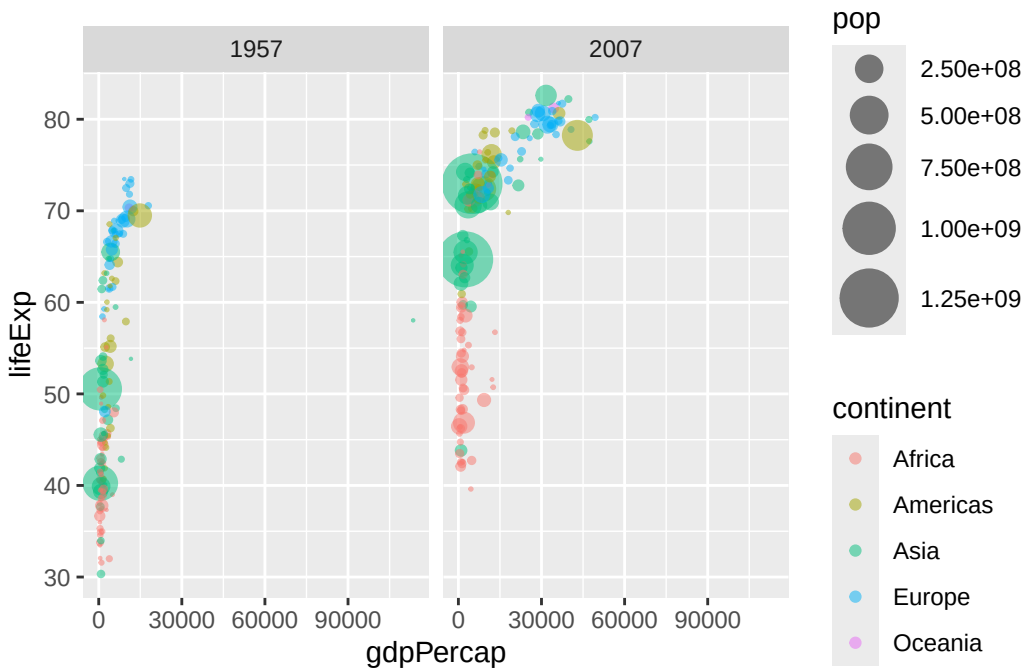
```
gapminder_2007 <- gapminder %>% filter(year==2007)
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5) +
  labs(title="Life Expectancy and GDP per Capita for 142 countries (2007)") +
  scale_size_area(max_size=10)
```




```
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 = 15) +
  labs(title="Life Expectancy and GDP per Capita for 142 countries (1957)")
```




```
gapminder_combo <- gapminder %>% filter(year == 1957 | year == 2007)
ggplot(gapminder_combo) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```

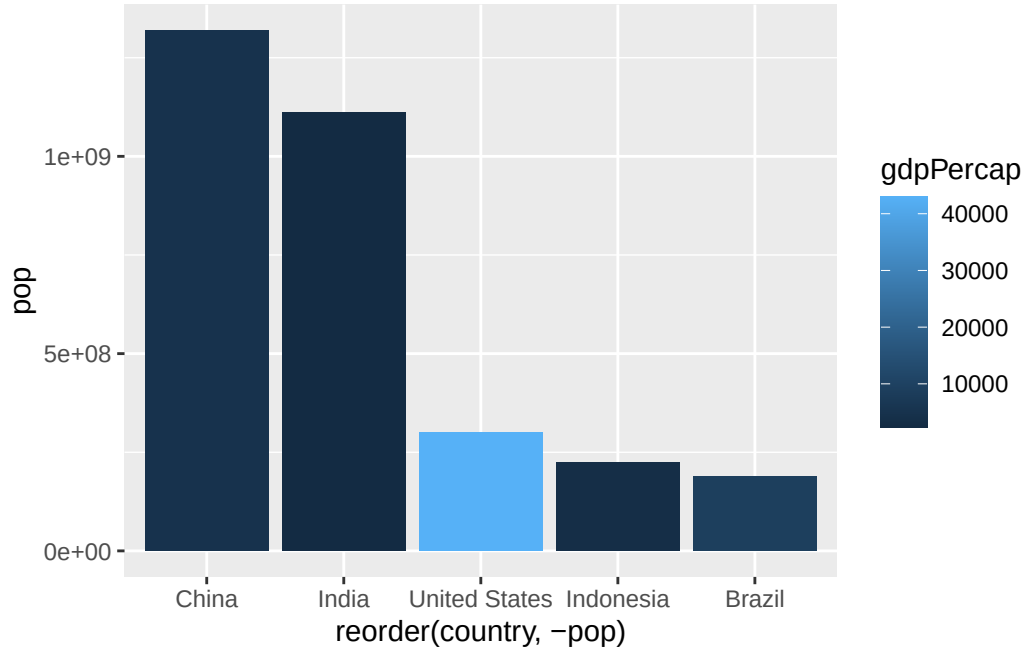


Bar charts

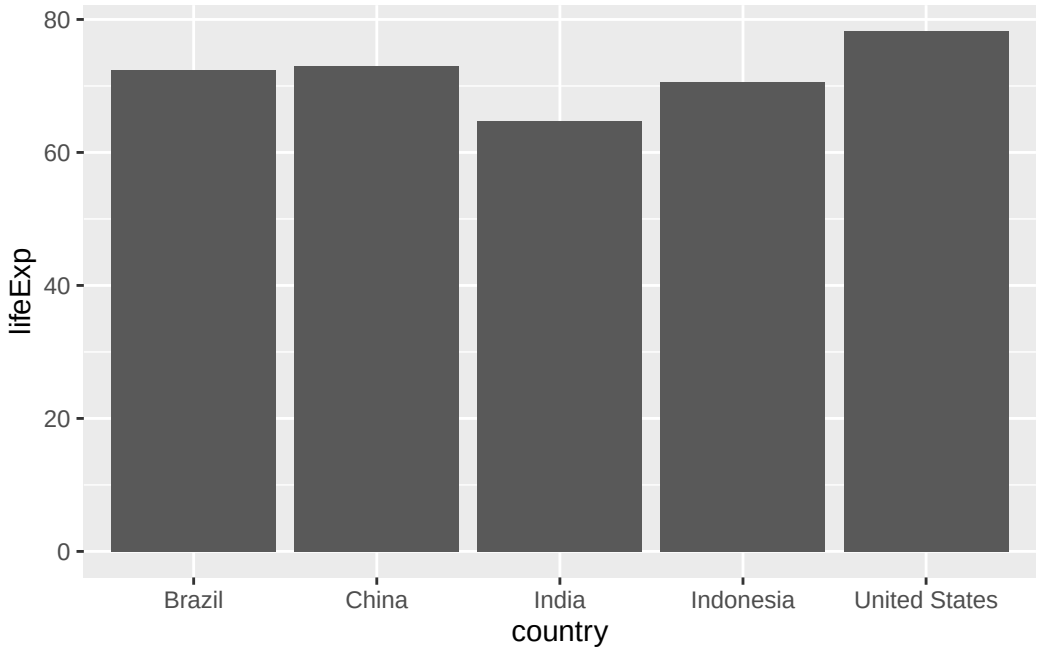
```
gapminder_top5 <- gapminder %>% filter(year==2007) %>% arrange(desc(pop)) %>%
  top_n(5, pop)

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=reorder(country,-pop), y=pop,
                                       fill=gdpPercap))
```



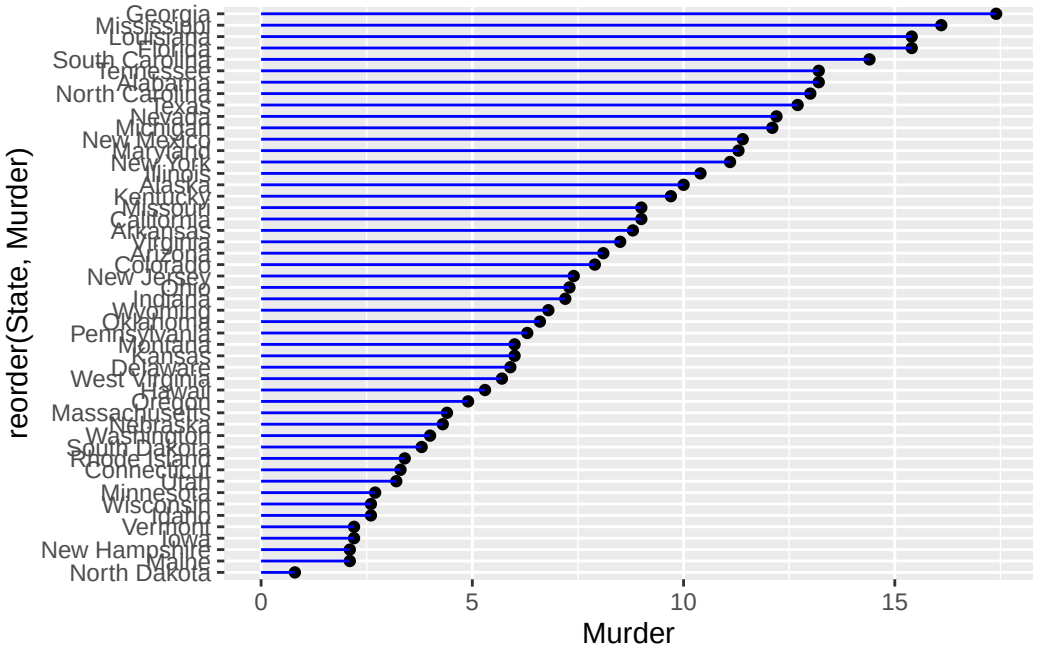

```
ggplot(gapminder_top5) + geom_col(aes(x=country, y=lifeExp))
```



```
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_point() + geom_segment(aes(x=State, xend=State, y=0, yend=Murder),
                              color="blue") + coord_flip()
```

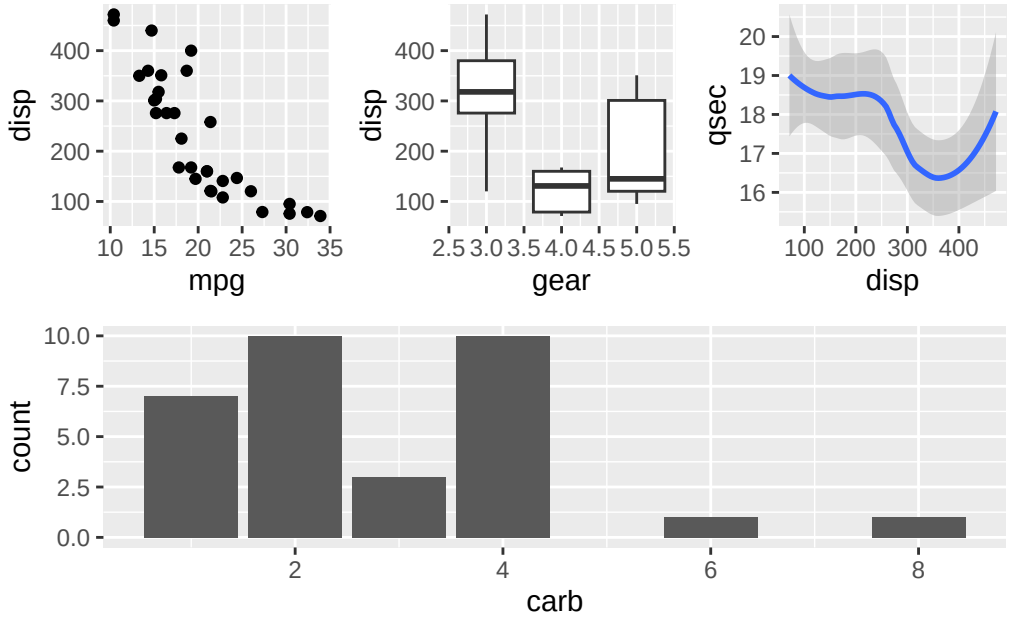



```
library(patchwork)
```

```
Warning: package 'patchwork' was built under R version 4.4.1
```

```
# 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 4.4.0 (2024-04-24)
Platform: x86_64-apple-darwin20
Running under: macOS Monterey 12.7.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRlapack.dylib;
```

```
locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Los_Angeles
tzcode source: internal

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

other attached packages:
[1] patchwork_1.3.0 dplyr_1.1.4     ggplot2_3.5.1

loaded via a namespace (and not attached):
 [1] vctrs_0.6.5       nlme_3.1-166      cli_3.6.3         knitr_1.49
 [5] rlang_1.1.4       xfun_0.50         generics_0.1.3    jsonlite_1.8.9
 [9] labeling_0.4.3    glue_1.8.0        colorspace_2.1-1  htmltools_0.5.8.1
[13] scales_1.3.0      rmarkdown_2.29    grid_4.4.0        evaluate_1.0.3
[17] munsell_0.5.1     tibble_3.2.1      fastmap_1.2.0     yaml_2.3.10
[21] lifecycle_1.0.4   compiler_4.4.0    pkgconfig_2.0.3   mgcv_1.9-1
[25] rstudioapi_0.17.1 lattice_0.22-6    farver_2.1.2      digest_0.6.37
[29] R6_2.5.1          tidyselect_1.2.1  splines_4.4.0     pillar_1.10.1
[33] magrittr_2.0.3    Matrix_1.7-1      withr_3.0.2       tools_4.4.0
[37] gtable_0.3.6
```