

```

# Questions:
# Q1. All of the above
# Q2. false
# Q3. network graphs
# Q4. ggplot2 is the only way to create plots in R
# Q5. geom_point()
# Q6.
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth()
# Q7.
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)
# Q8.
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()
# Q9. 5196
# Q10. 4
# Q11. 127
# Q12. 0.0244 (2.44%)
# Q13.
ggplot(genes) +
  aes(x=Condition1, y=Condition2) +
  geom_point()
# Q14.
p + scale_colour_manual(values=c("blue","gray","red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
        x="Control (no drug) ",
        y="Drug Treatment")
# Q15.
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point()

```

Q16.

```
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)
```

Q17.

```
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)
```

Q18.

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

Q19.

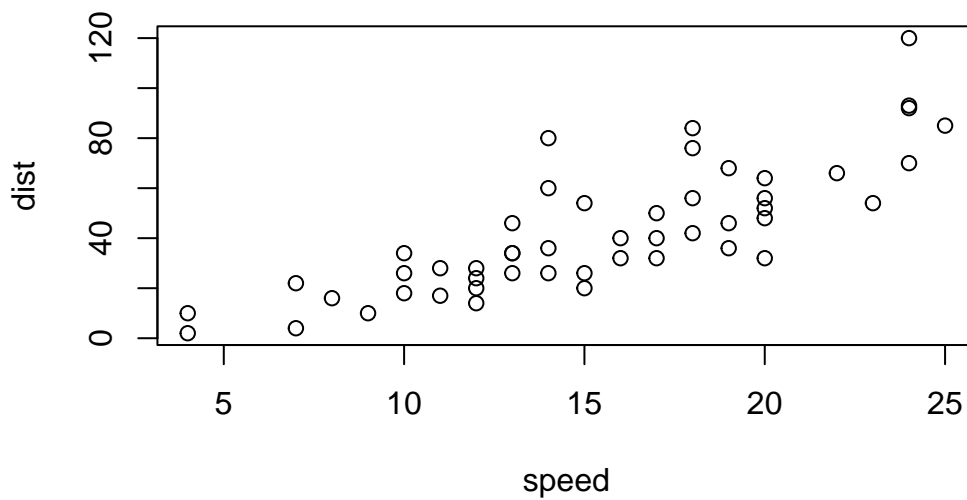
```
ggplot(gapminder_top5) +  
  aes(x=reorder(country, -pop), y=pop, fill=country) +  
  geom_col(col="gray30") +  
  guides(fill="none")
```

class05

Anjali Hamada

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

library(ggplot2)
#A quick base R plot - not ggplot2
plot(cars)
```

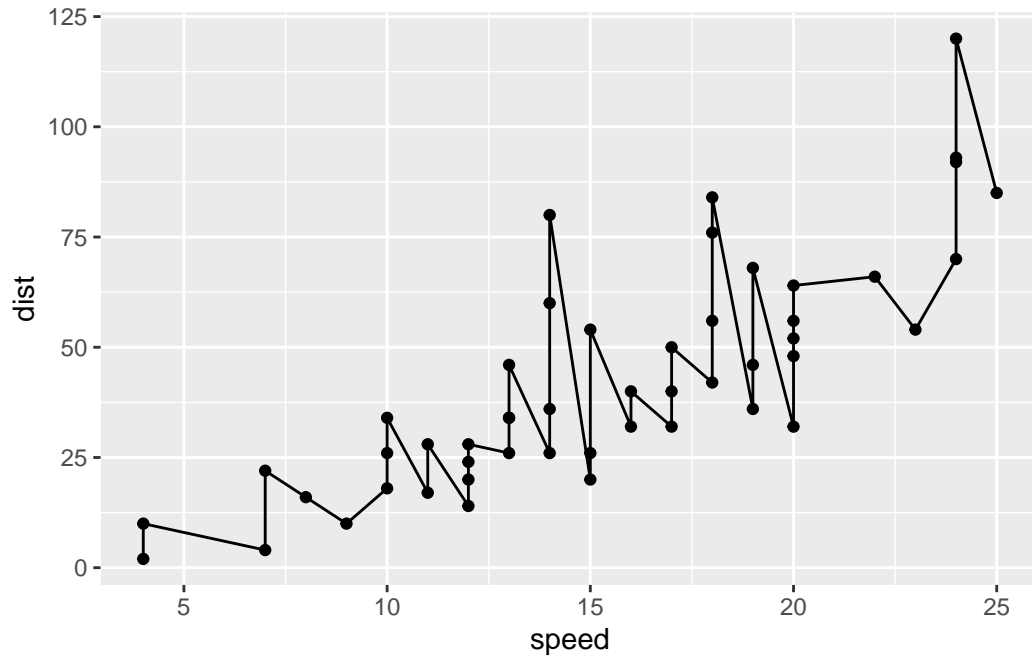


```
#Our first ggplot
ggplot(data=cars) + aes(x=speed, y=dist) + geom_point()
```



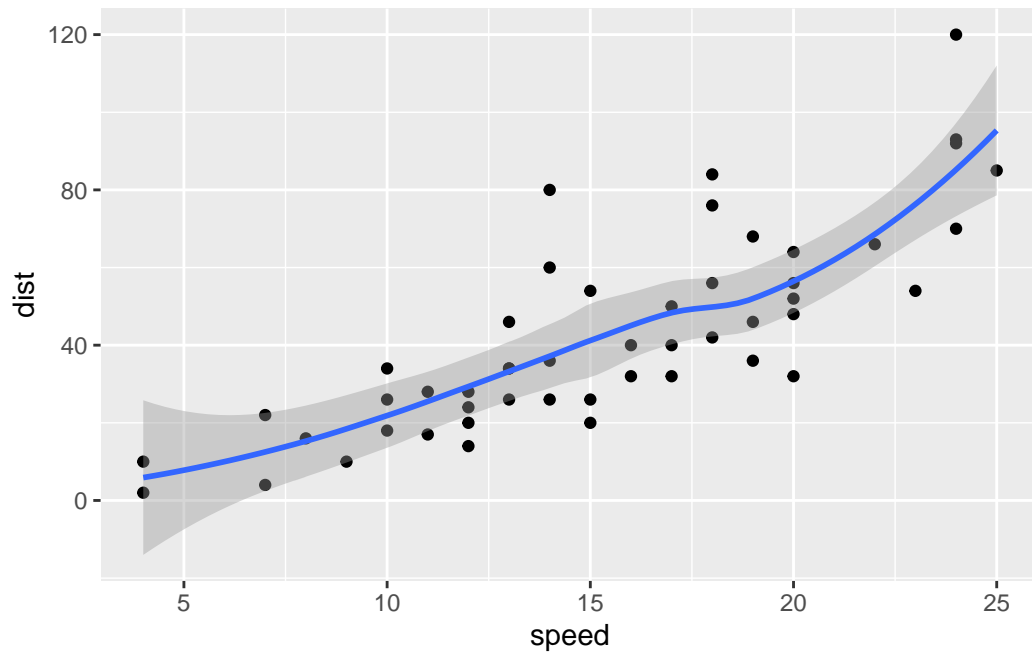
```
#Our first ggplot
p <- ggplot(data=cars) + aes(x=speed, y=dist) + geom_point()

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



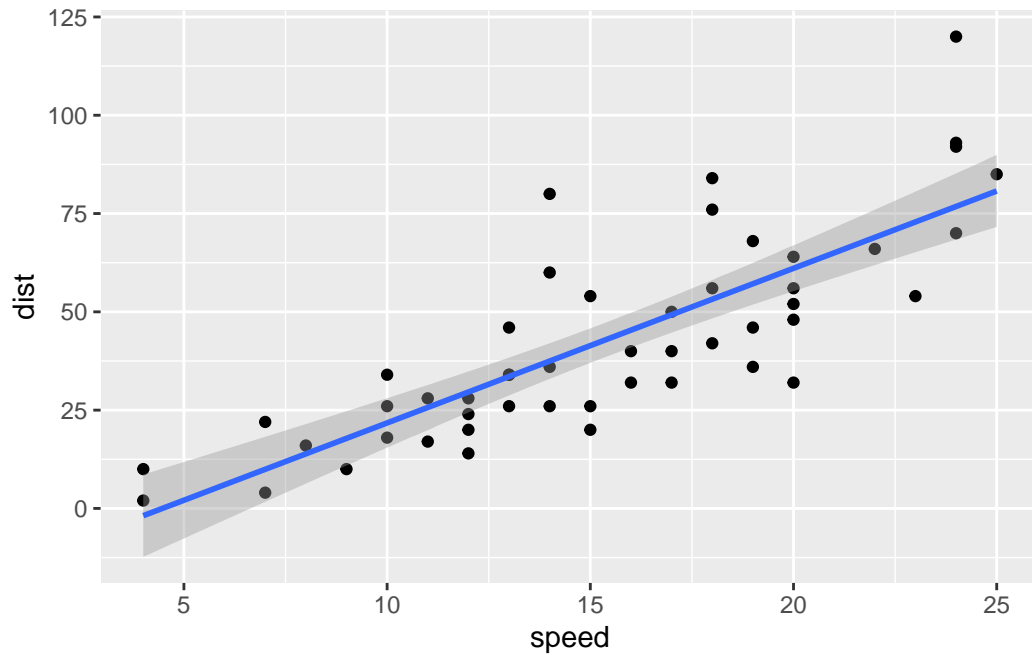
```
#Add a trend line close to the data  
p+geom_smooth()
```

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



```
p+geom_smooth(method="lm")
```

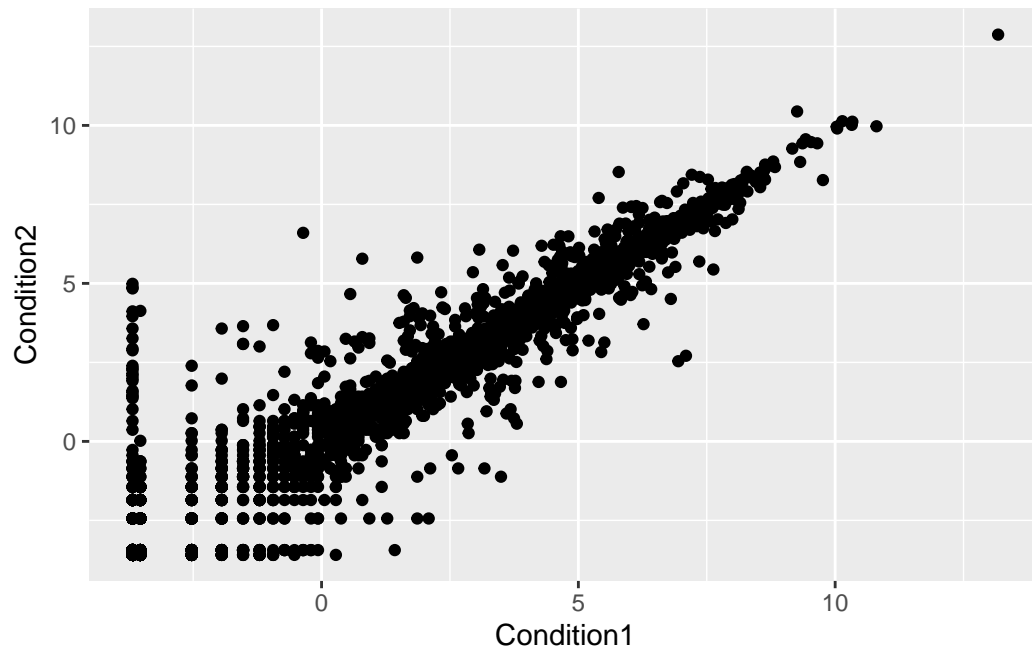
```
`geom_smooth()` using formula = 'y ~ x'
```



```
# Now let's try it with genes!
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes)
```

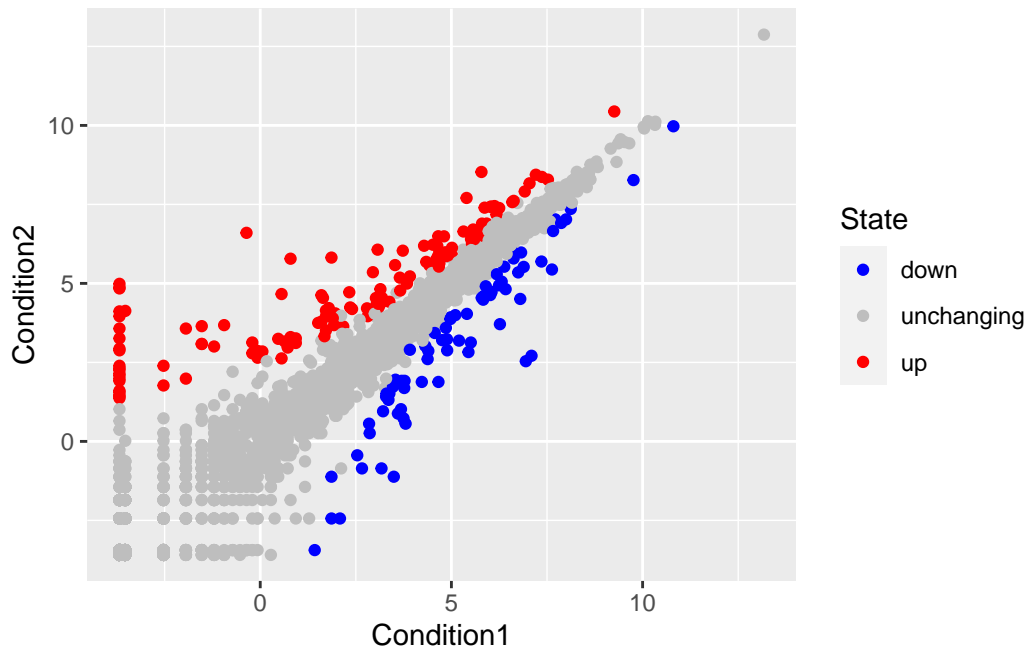
	Gene	Condition1	Condition2	State
1	A4GNT	-3.6808610	-3.4401355	unchanging
2	AAAS	4.5479580	4.3864126	unchanging
3	AASDH	3.7190695	3.4787276	unchanging
4	AATF	5.0784720	5.0151916	unchanging
5	AATK	0.4711421	0.5598642	unchanging
6	AB015752.4	-3.6808610	-3.5921390	unchanging

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

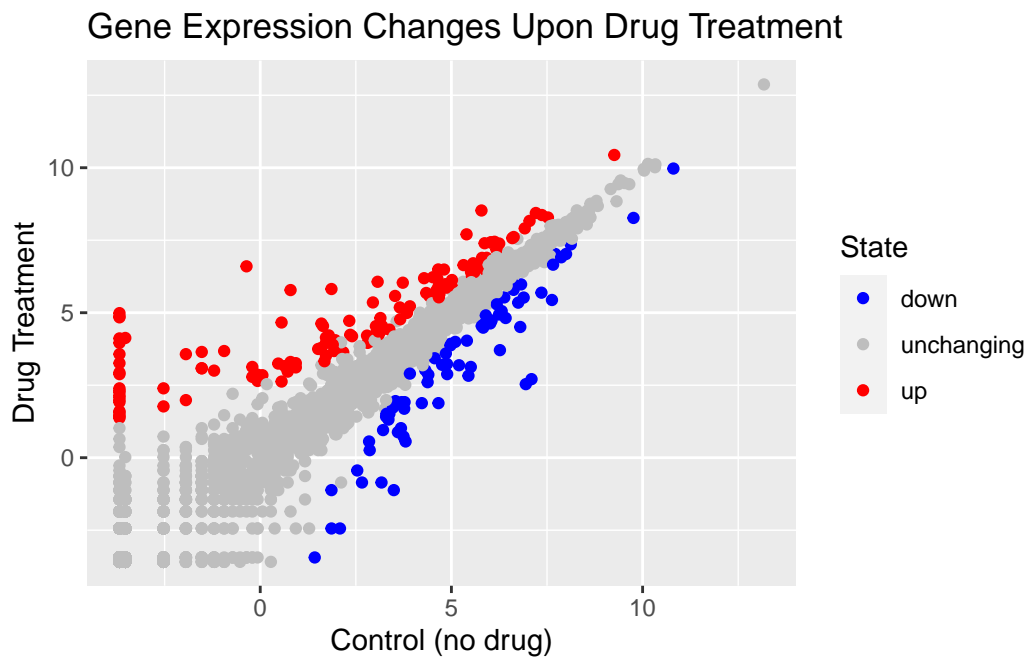


```
#map State to point color
p <- ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point()

#change up colors
p + scale_colour_manual( values=c("blue","gray","red") )
```

```
#change axis/title labels
p + scale_colour_manual( values=c("blue","gray","red") ) + labs(title="Gene Expression Cha
```



```
#going further for gapminder
library(gapminder)
library(dplyr)
```

Attaching package: 'dplyr'

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

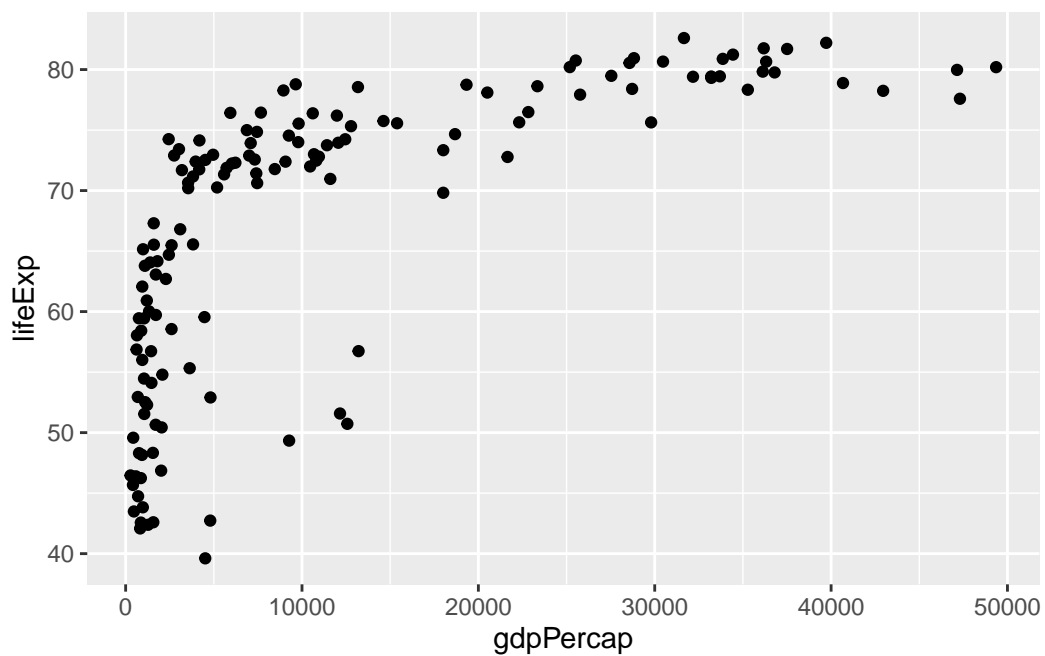
filter, lag

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

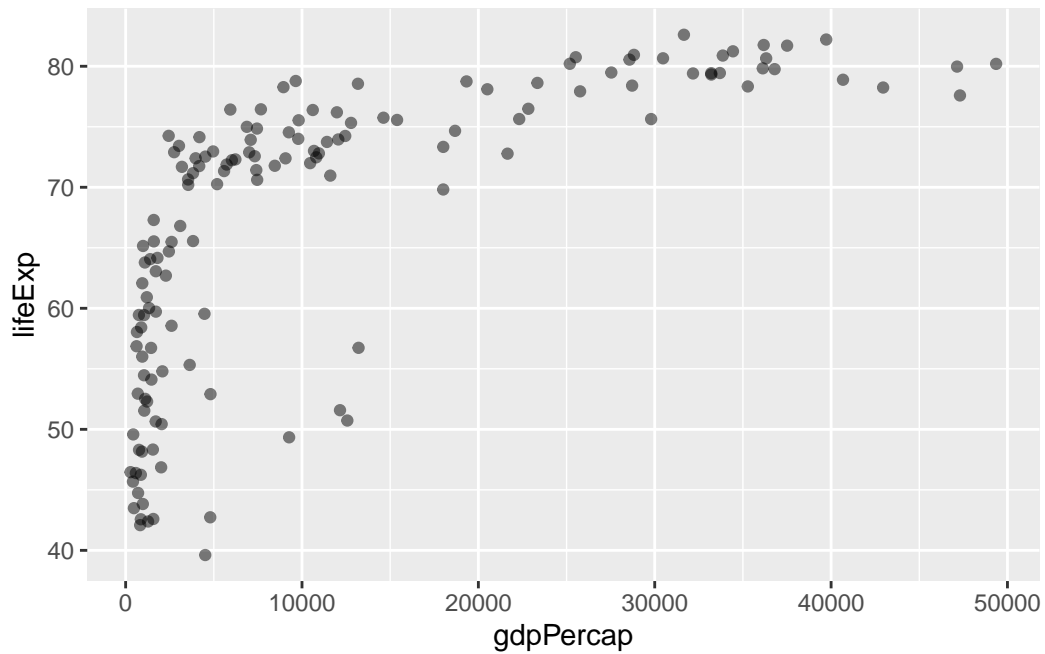
intersect, setdiff, setequal, union

```
#filters to only contain rows with year 2007
gapminder_2007 <- gapminder %>% filter(year==2007)

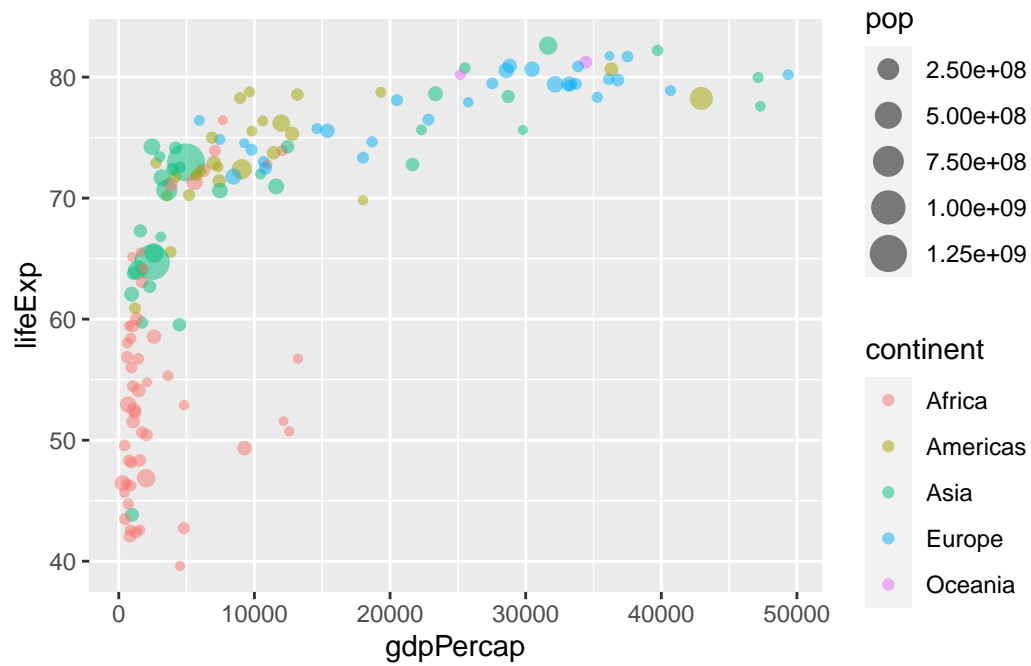
#scatterplot
ggplot(gapminder_2007) + aes(x=gdpPercap, y=lifeExp) + geom_point()
```



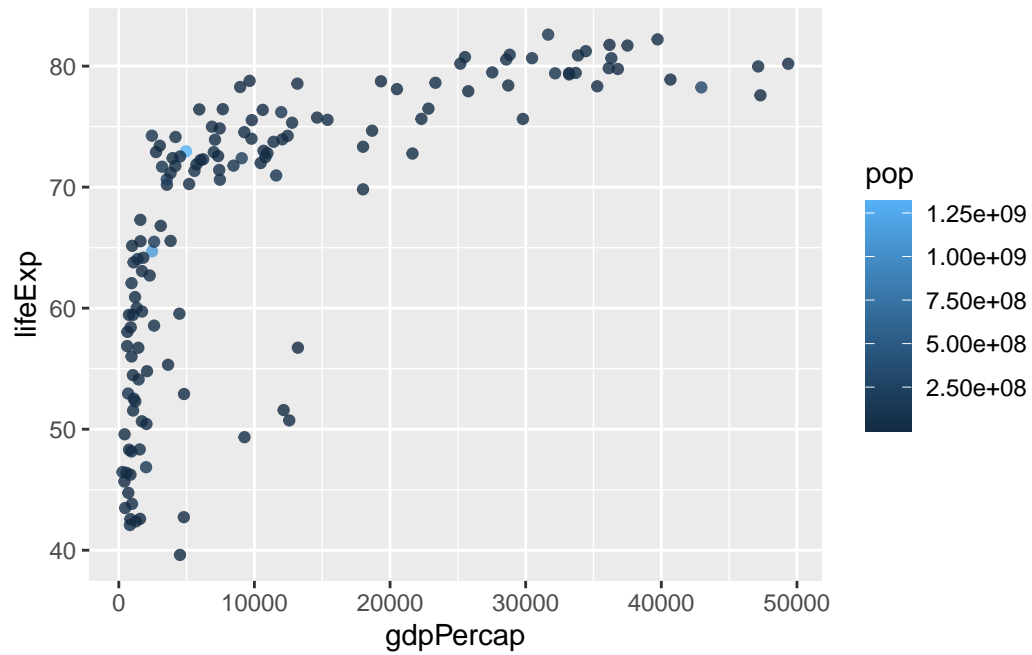
```
#change alpha (transparency)
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```



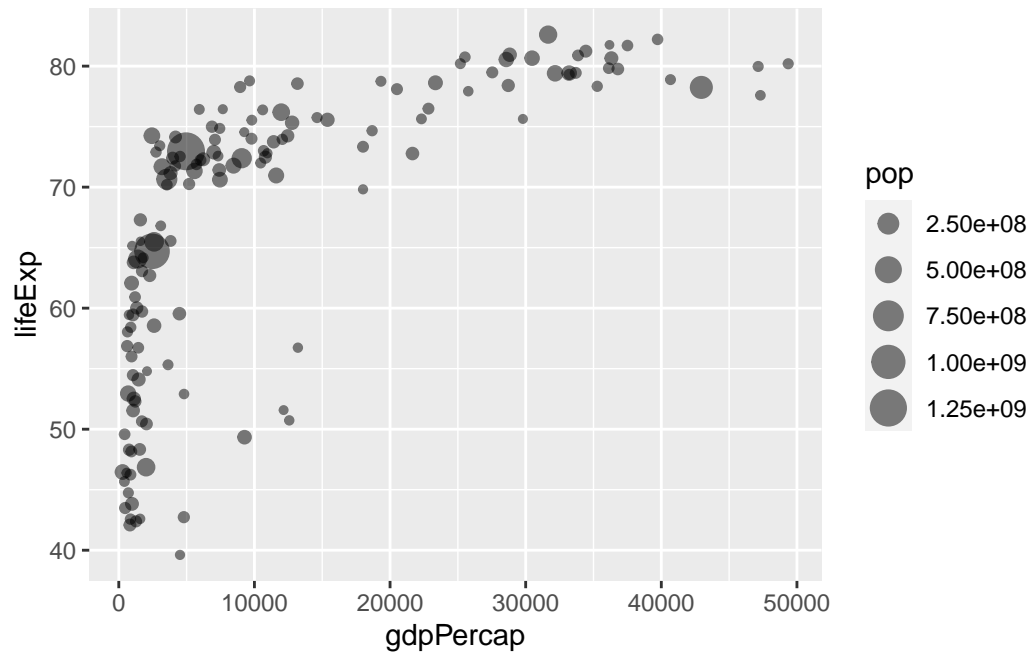
```
#map continent to color and population (pop) to size
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```



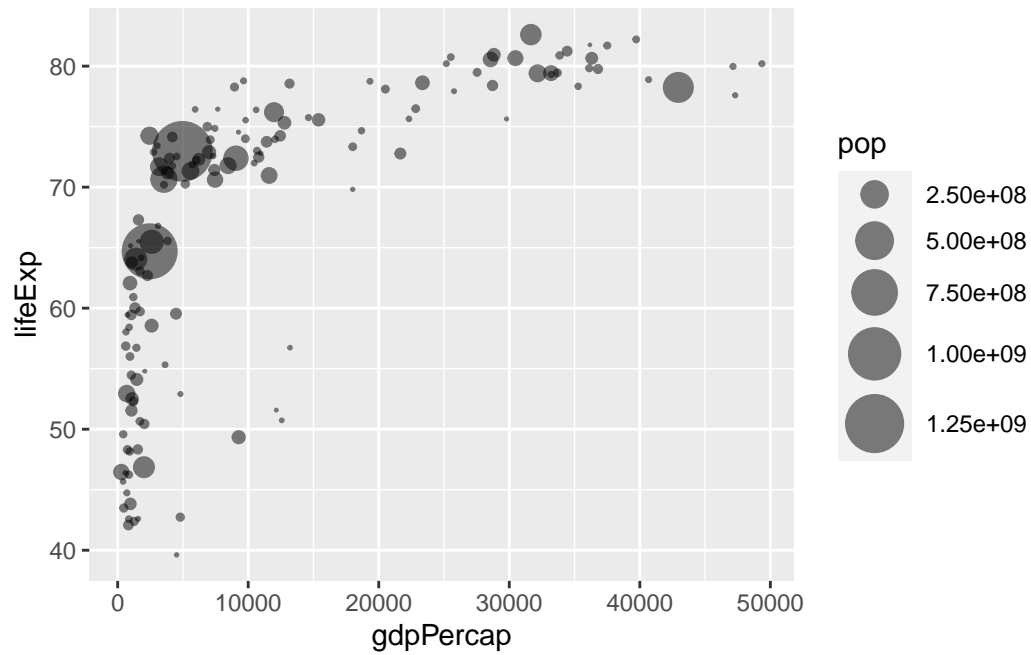
```
#we can also color by pop
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



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

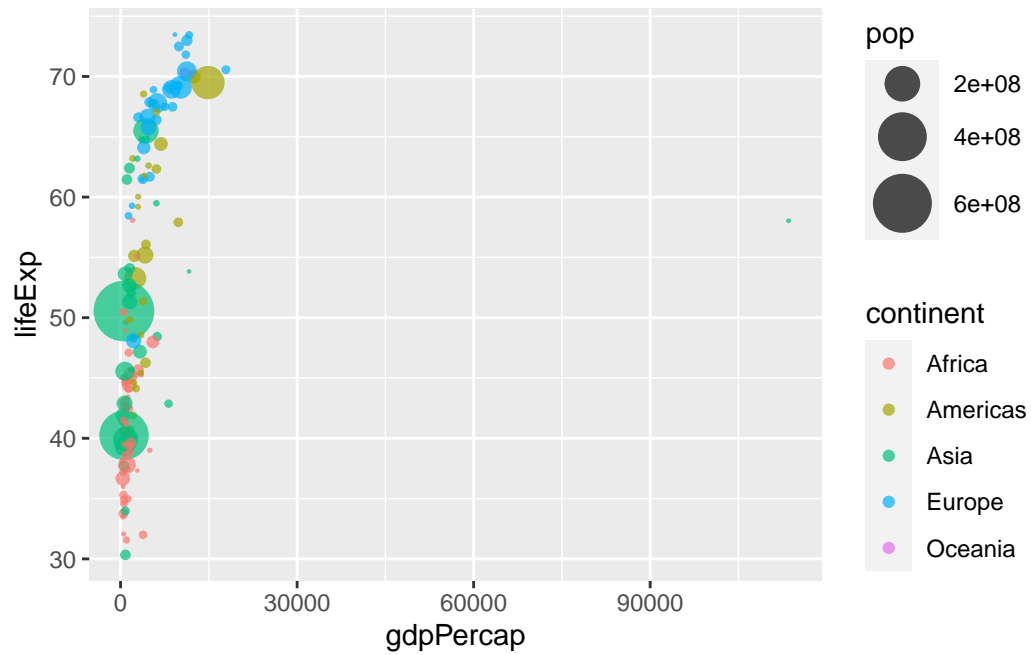


```
#we can make the actual differences be based on point size
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPerCap, y = lifeExp, size = pop), alpha=0.5) +
  scale_size_area(max_size = 10)
```



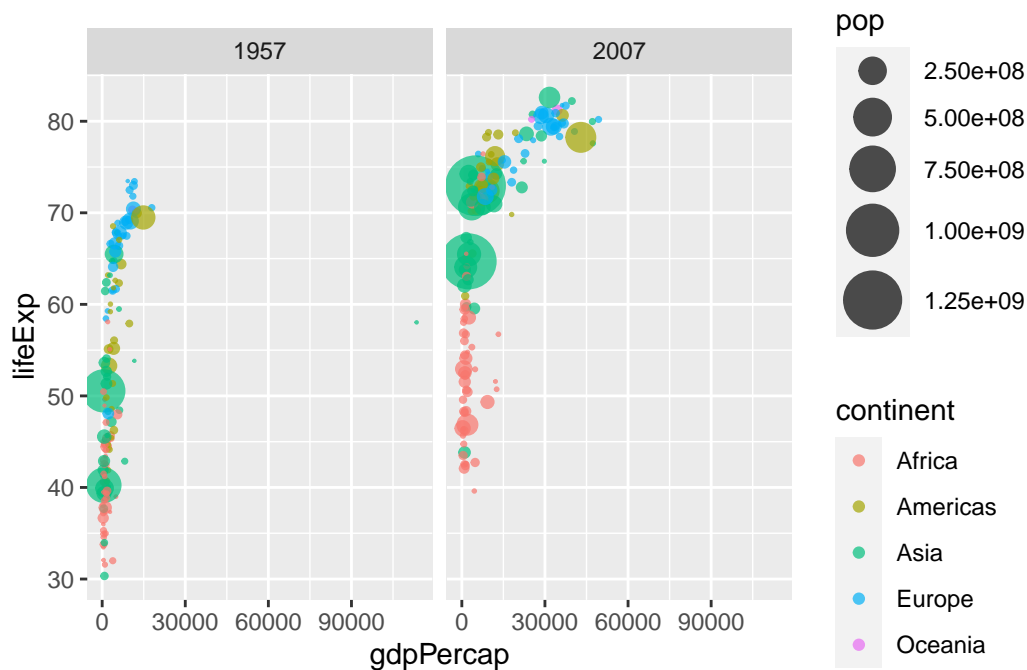
```
#doing the same for 1957
gapminder_1957 <- gapminder %>% filter(year==1957)

ggplot(gapminder_1957) + aes(x = gdpPercap, y = lifeExp, color=continent, size = pop) + ge
```



```
#including both '57 and '07
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) + sc
```

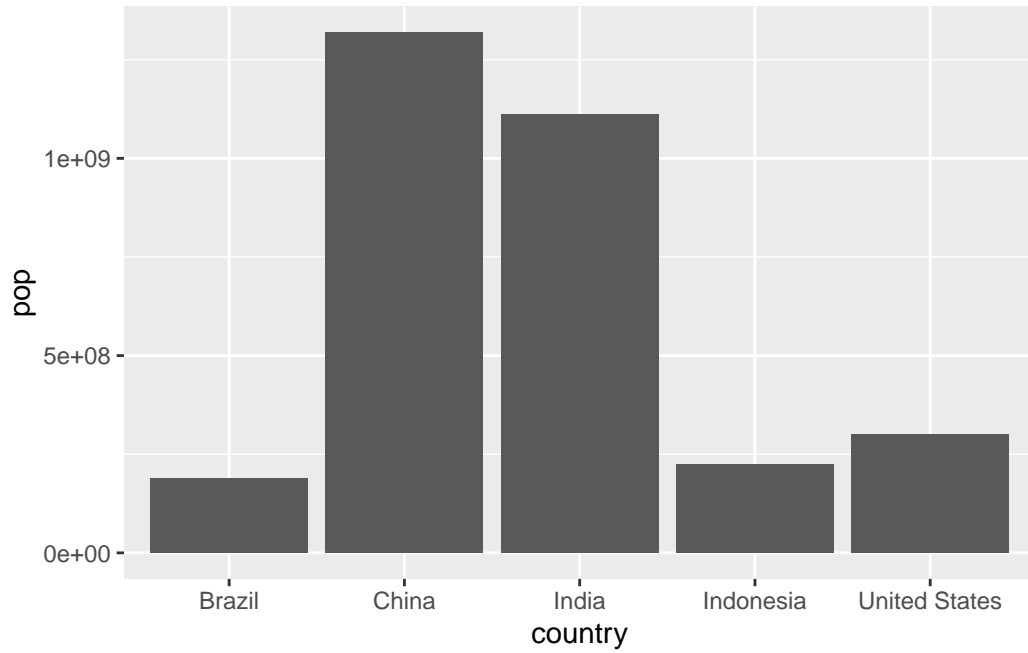



```
# bar charts!
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)
gapminder_top5
```

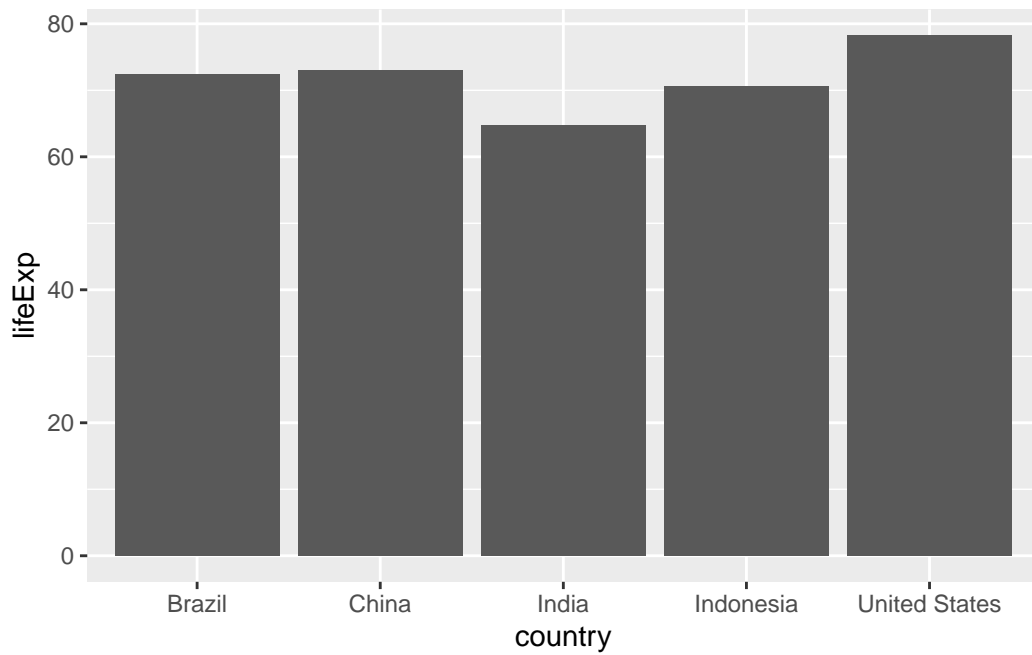
A tibble: 5 x 6

	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.

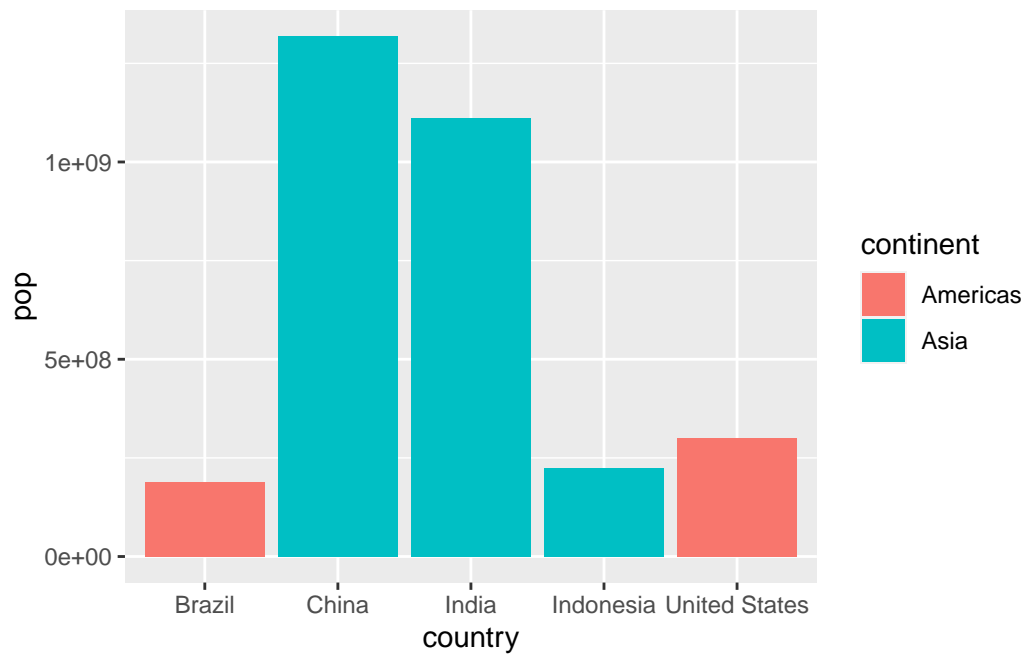
```
# a simple bar chart
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = pop))
```



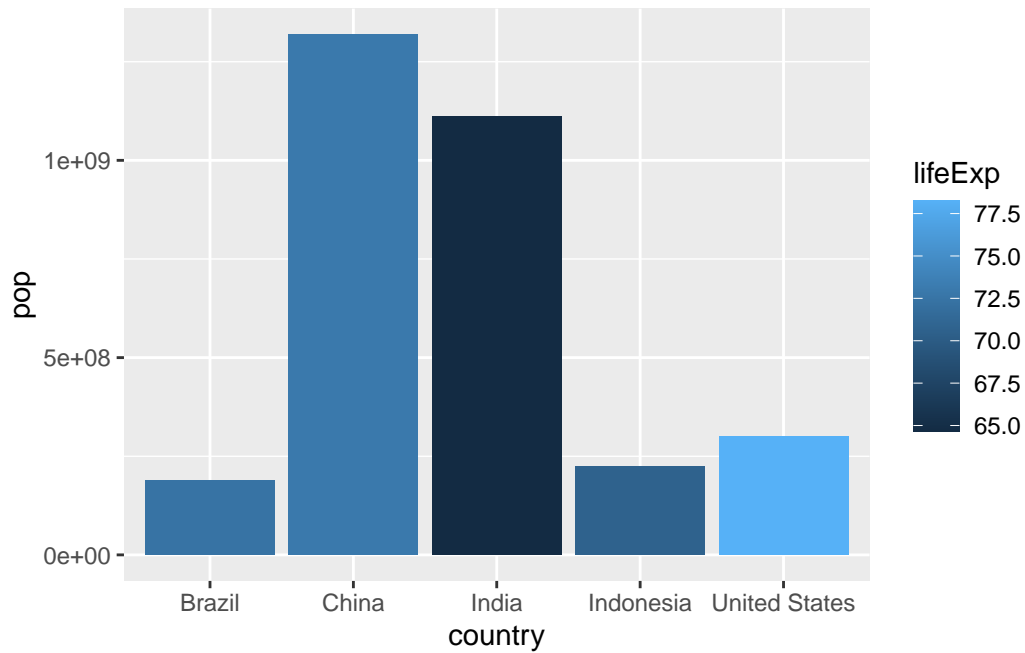
```
# top 5 life expectancy in 2007  
ggplot(gapminder_top5) + geom_col(aes(x = country, y = lifeExp))
```



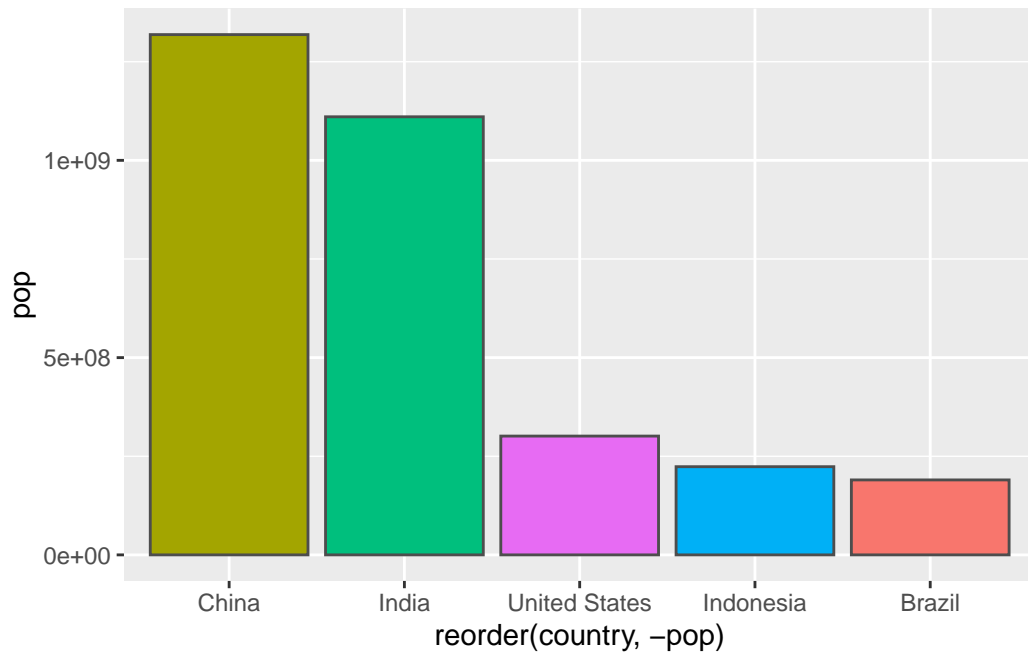
```
#fill bars with color
ggplot(gapminder_top5) + geom_col(aes(x = country, y = pop, fill = continent))
```



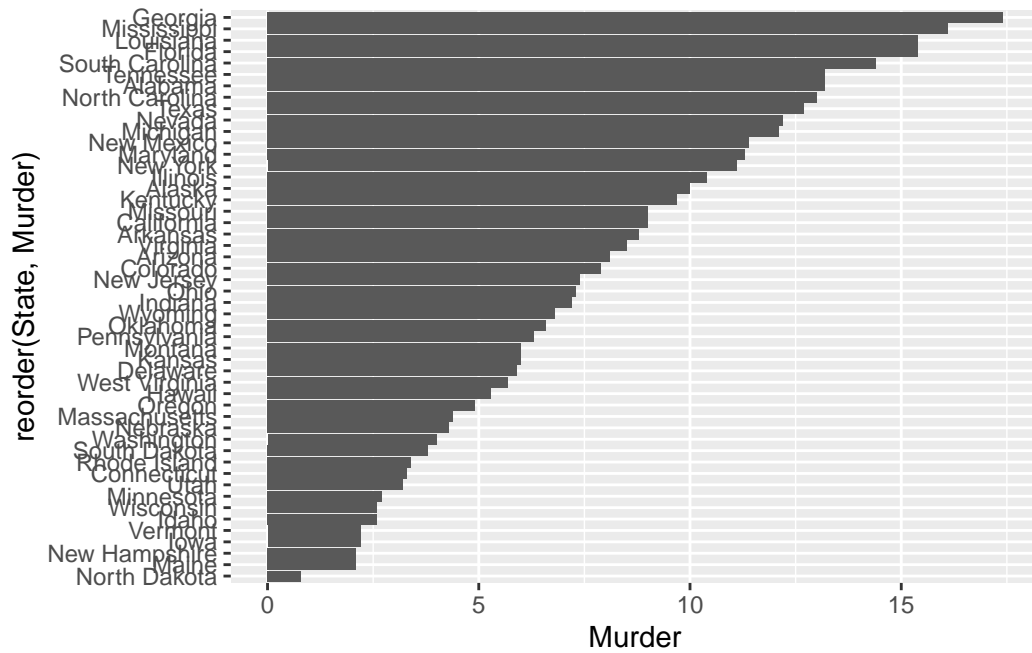
```
#can we fill with a numeric variable like lifeExp?
ggplot(gapminder_top5) + geom_col(aes(x = country, y = pop, fill = lifeExp))
```



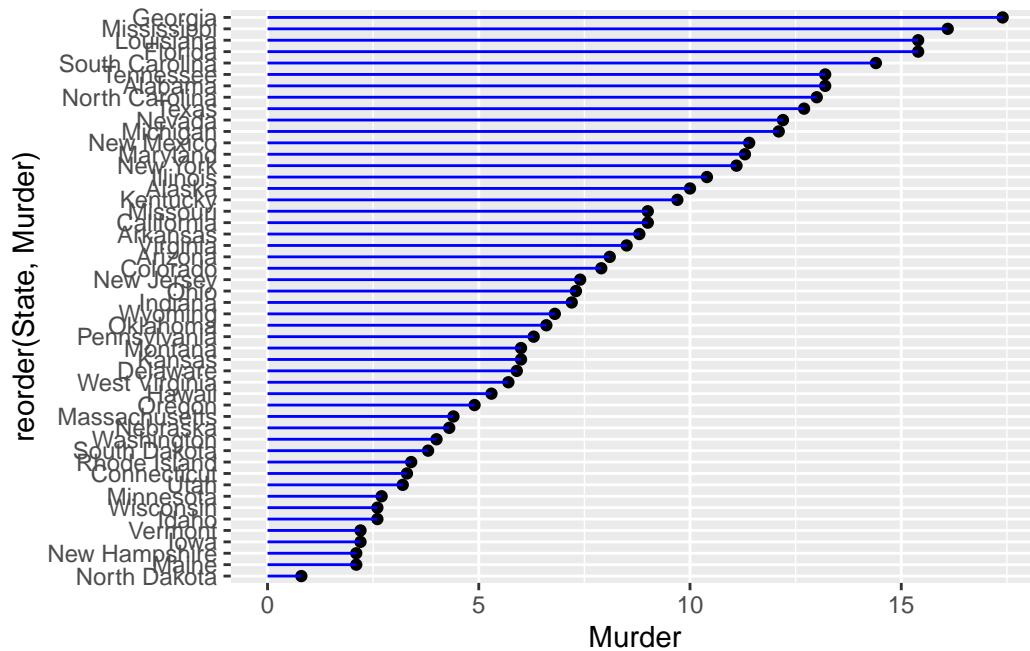
```
# Plot population size by country. Create a bar chart showing the population (in millions)
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=country) +
  geom_col(col="gray30") +
  guides(fill="none")
```



```
#flipping a bar chart
USArrests$State <- rownames(USArrests)
ggplot(USArrests) +
  aes(x=reorder(State,Murder), y=Murder) +
  geom_col() +
  coord_flip()
```



```
#changing format
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()
```



```
#animations
library(gapminder)
library(gganimate)
```

No renderer backend detected. gganimate will default to writing frames to separate files
Consider installing:

- the ``gifski`` package for gif output
- the ``av`` package for video output

and restarting the R session

```
# 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_color_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)
```

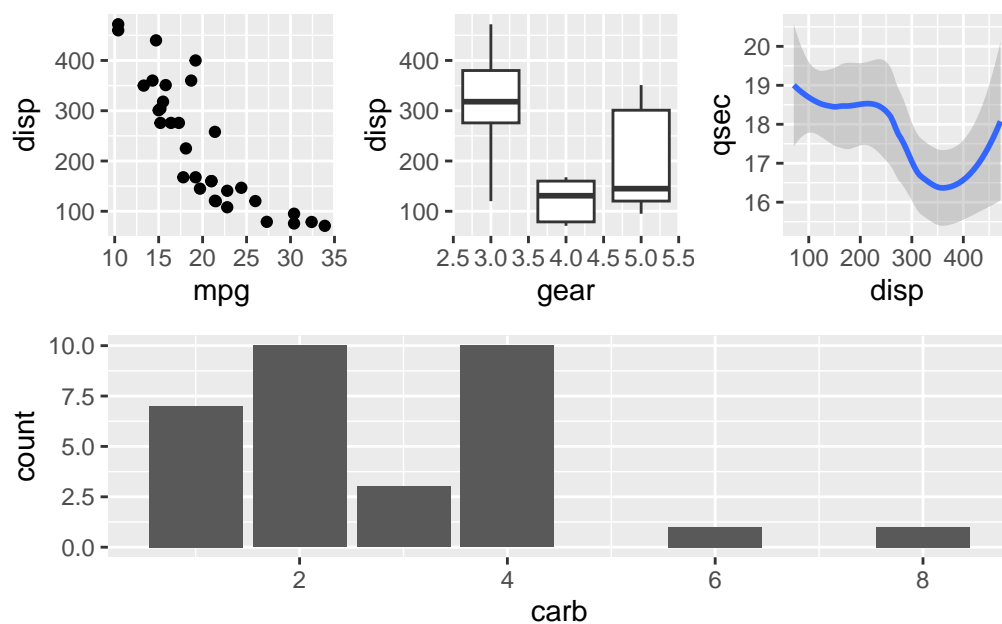
Warning: No renderer available. Please install the gifski, av, or magick package to create animated output

```
# 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(dis, 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 4.3.2 (2023-10-31)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.0
```

```
Matrix products: default
```

```
BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/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_1.2.0 gganimate_1.0.8 dplyr_1.1.4      gapminder_1.0.0
[5] ggplot2_3.4.4
```

```
loaded via a namespace (and not attached):
```

```
[1] Matrix_1.6-5      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_0.22-5
[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_1.9-1
[29] digest_0.6.34     grid_4.3.2        hms_1.1.3         lifecycle_1.0.4
[33] nlme_3.1-164      prettyunits_1.2.0 vctrs_0.6.5       evaluate_0.23
[37] glue_1.7.0        farver_2.1.1      fansi_1.0.6       colorspace_2.1-0
[41] rmarkdown_2.25    tools_4.3.2       pkgconfig_2.0.3   htmltools_0.5.7
```