Data Visualization in R with ggplot2

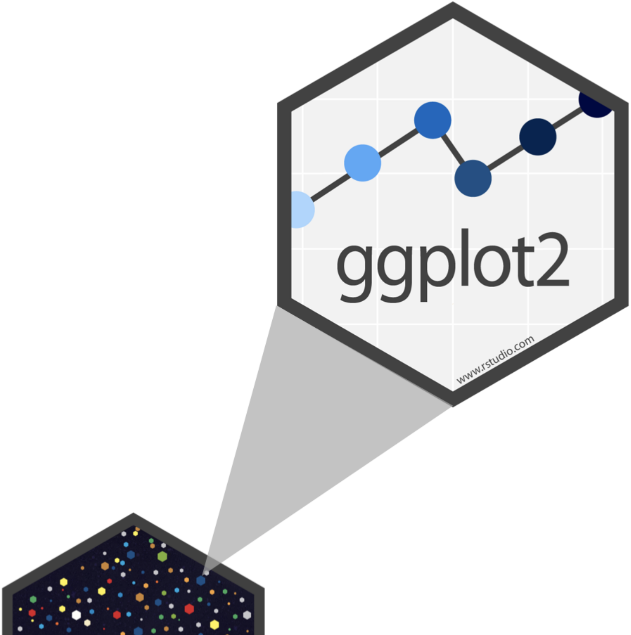
Josh Quan

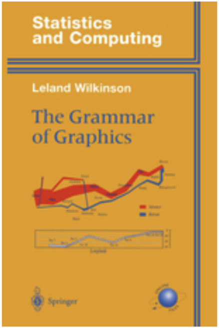
Introduction

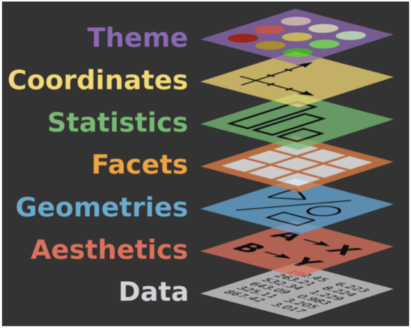
> *"The simple graph has brought more information to the data analyst's mind than any other device."*
> *— John Tukey*

- Data visualization is the creation and study of the visual representation of data.
- Many tools for visualizing data (R is one of them)
- Many approaches/systems within R for making data visualizations, **ggplot2** is one of them

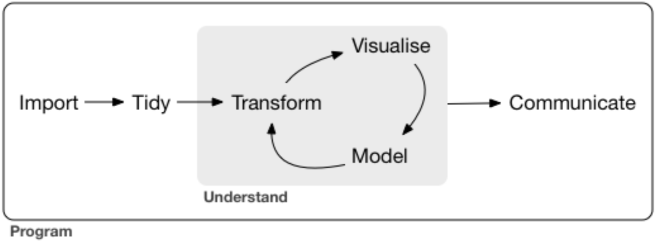
ggplot2 ∈ tidyverse



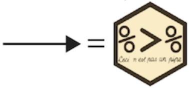

Statistics
and Computing
Leland Wilkinson
The Grammar
of Graphics
Springer

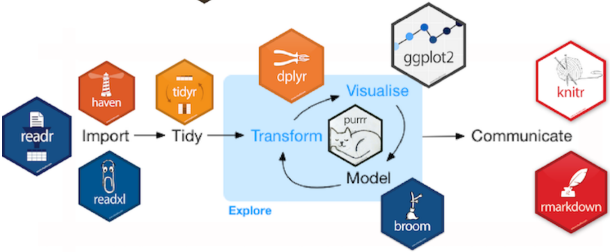

Theme
Coordinates
Statistics
Facets
Geometries
Aesthetics
A → X
B → Y
Data

ggplot2 ∈ tidyverse



ggplot2 ∈ tidyverse





data.frames

factors

strings

Dataset

Stanford Open Policing Project

Police Searches Drop Dramatically in States that Legalized Marijuana

- Police Stop Data
  - state, driver race, stop rate, marijuana legalization status

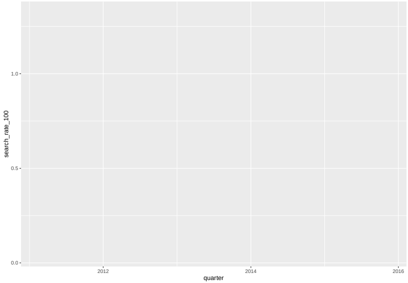
```r
stops <- read_csv("./data/opp-search-marijuana_state.csv")
  filter(state %in% c("WA", "CO")) %>%
  mutate(legalization_status = ifelse(quarter <= "2013-01-0
         search_rate_100 = search_rate * 100)
```

Basic ggplot2 syntax

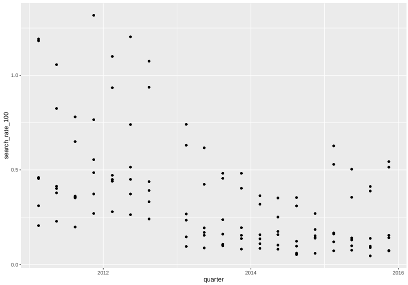
- DATA
- MAPPING
- GEOM

Step-by-step

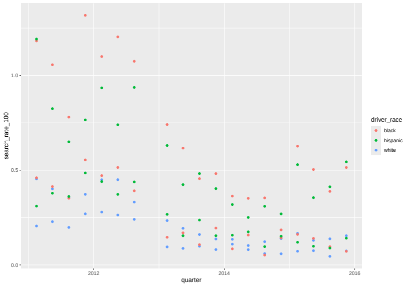
```r
ggplot(data = stops)
ggplot(data = stops, mapping = aes(x = quarter, y = search_
```



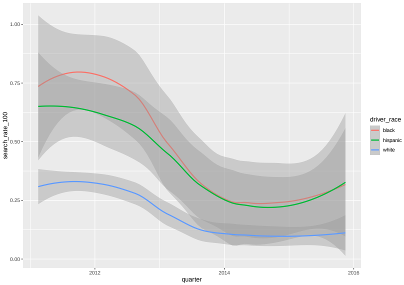
```
ggplot(data = stops, mapping = aes(x = quarter, y = search_
  geom_point()
```



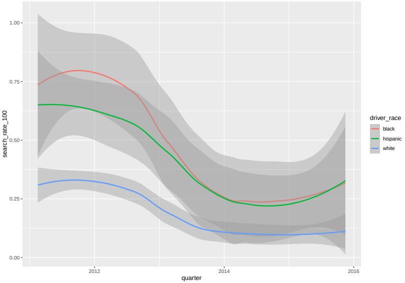
```r
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point()
```



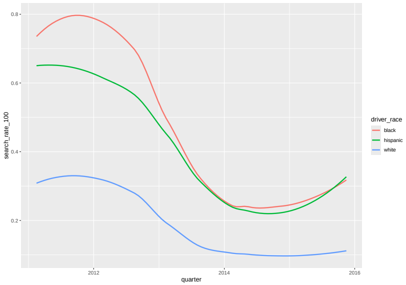
```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_smooth()
```



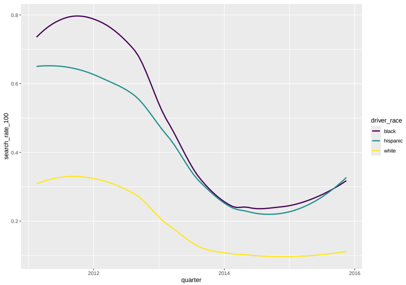
```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_smooth(method = "loess")
```



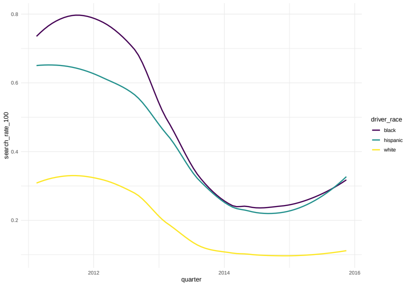
```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_smooth(method = "loess", se = FALSE)
```



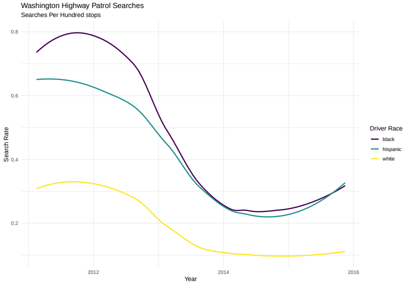
```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_smooth(method = "loess", se = FALSE) +
  scale_color_viridis_d()
```



```r
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_smooth(method = "loess", se = FALSE) +
  scale_color_viridis_d() +
  theme_minimal()
```



```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_smooth(method = "loess", se = FALSE) +
  scale_color_viridis_d() +
  theme_minimal() +
  labs(x = "Year", y = "Search Rate", color = "Driver Race'
       title = "Washington Highway Patrol Searches", subtit
```



ggplot, the making of

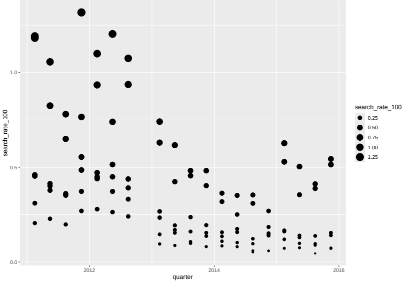
1. “Initialize” a plot with ggplot()
2. Add layers with geom_ functions

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))+
  geom_point(mapping = aes(x = displ, y = hwy))
```

Mapping

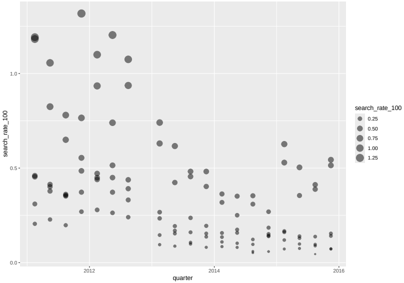
Size data points by a numerical variable

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point()
```



Set alpha value

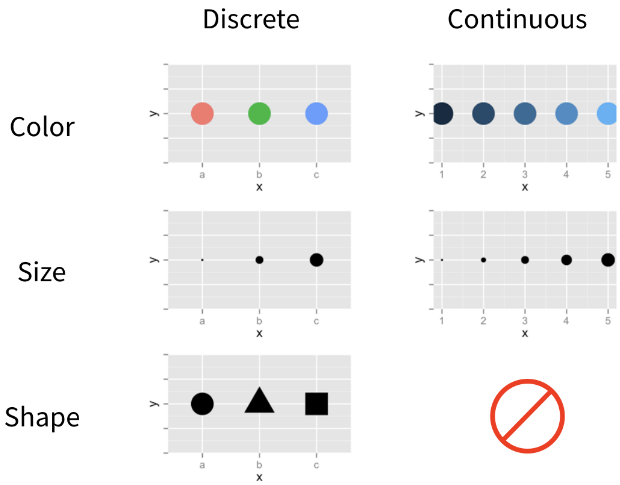
```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point(alpha = 0.5)
```



Your turn!

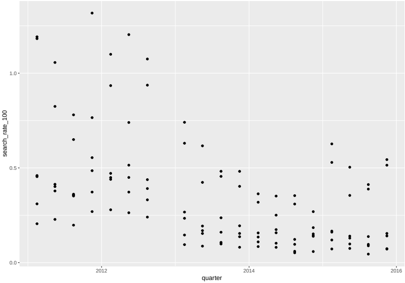
Exercise: Using information from https://ggplot2.tidyverse.org/articles/ggplot2-specs.html add color, size, alpha, and shape aesthetics to your graph. Experiment. Do different things happen when you map aesthetics to discrete and continuous variables? What happens when you use more than one aesthetic?

```r
stops %>% ggplot(aes(x = quarter , y = search_rate_100, col
  geom_point() +
  theme_minimal(base_size = 12)
```


Discrete
Continuous
Color
Size
Shape
y
x
a
b
c
1
2
3
4
5

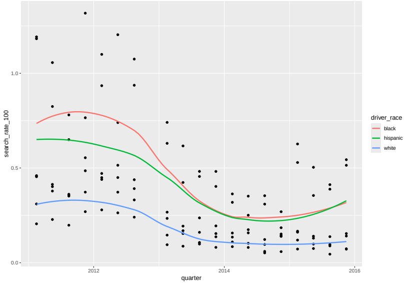
Mappings can be at the geom level

```
ggplot(data = stops) +
  geom_point(mapping = aes(x = quarter, y = search_rate_100
```



Different mappings for different geoms

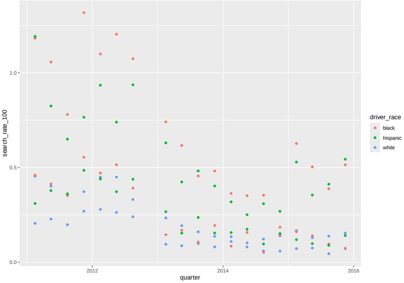
```
ggplot(data = stops, mapping = aes(x = quarter, y = search_
  geom_point() +
  geom_smooth(aes(color = driver_race), method = "loess", s
```



Set vs. map

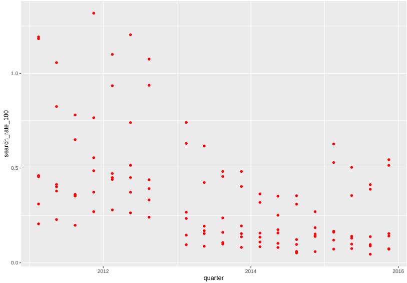
- To **map** an aesthetic to a variable, place it inside aes()

```
ggplot(data = stops,
  mapping = aes(x = quarter,
                y = search_rate_100,
            color = driver_race)) +
  geom_point()
```



- To **set** an aesthetic to a value, place it outside aes()

```r
ggplot(data = stops,
  mapping = aes(x = quarter,
                y = search_rate_100)) +
  geom_point(color = "red")
```

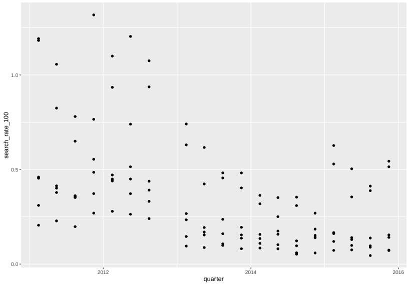


- Can specify HTML color codes

```r
ggplot(data = stops,
  mapping = aes(x = quarter,
                y = search_rate_100)) +
  geom_point(color = "#63B3E8")
```

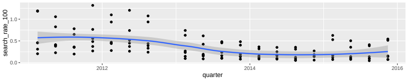
Data can be passed in

```
stops %>%
  ggplot(aes(x = quarter, y = search_rate_100)) +
    geom_point()
```



Assign ggplot() to objects for layering

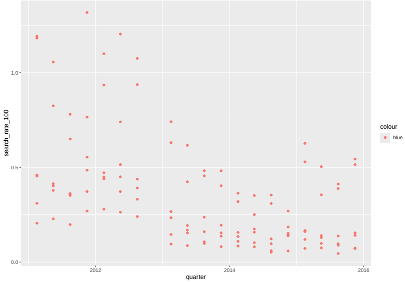
```r
p <- ggplot(stops, aes(x = quarter, y = search_rate_100)) +
  geom_point()

p + geom_smooth()
```



Common early pitfalls

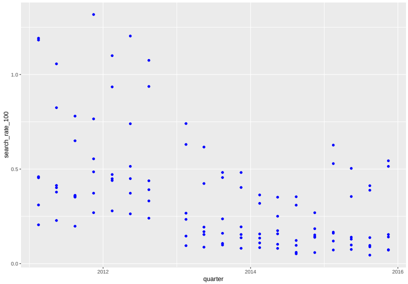
Mappings that aren't

```
ggplot(data = stops) +
  geom_point(aes(x = quarter, y = search_rate_100, color =
```



Mappings that aren't

```
ggplot(data = stops) +
  geom_point(aes(x = quarter, y = search_rate_100), color =
```



Your turn!

Exercise: What is wrong with the following?

```
stops %>%
  ggplot(aes(x = quarter, y = search_rate_100, color = lega
    geom_point()
```

+ and %>%

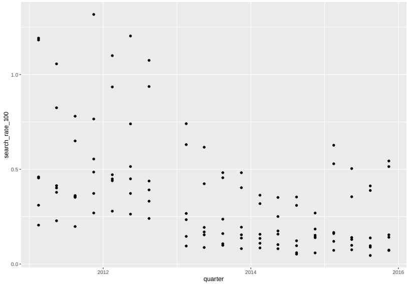
What is wrong with the following?

```
stops %>%
  ggplot(aes(x = quarter, y = search_rate_100, color = lega
    geom_point()
```

```
## Error in `geom_point()`:
## ! `mapping` must be created by `aes()`.
## i Did you use `%>%` or `|>` instead of `+`?
```

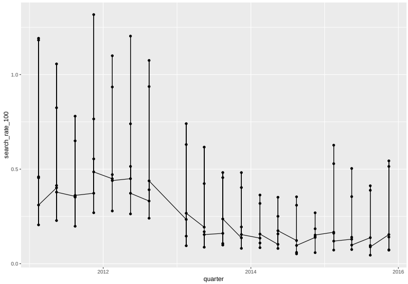
Basic plot

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100))
  geom_point()
```



Two layers

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100))
  geom_point()  +
  geom_line()
```

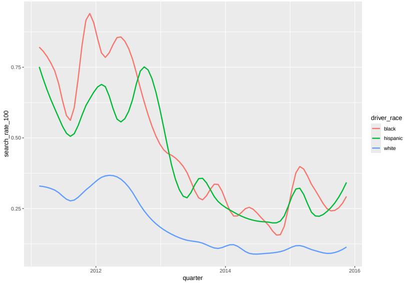


The power of groups

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point() +
  geom_line()
```

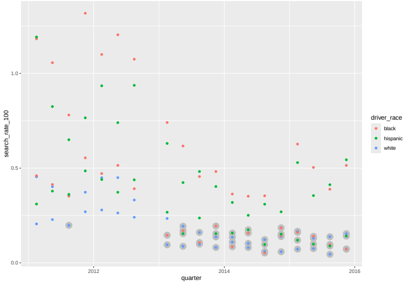
Now we've got it

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_smooth(span = 0.2, se = FALSE)
```



Control data by layer

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point(data = filter(stops, search_rate_100 < .2),
             size = 5, color = "gray") +
  geom_point()
```

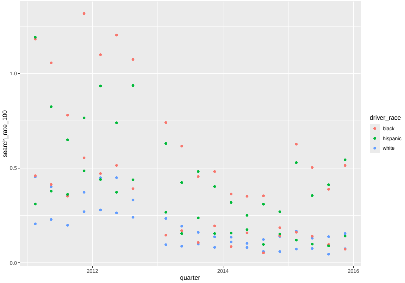


Your turn!

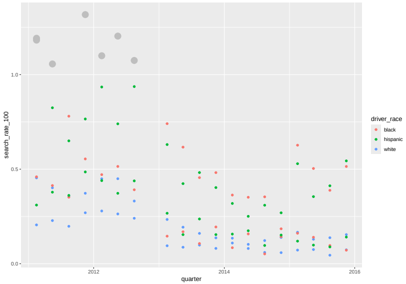
Exercise: Work with your neighbor to sketch what the following plots will look like. No cheating! Do not run the code, just think through the code for the time being.

```
pre_legalization_high <- stops %>%
  filter((quarter < "2013-01-01" & search_rate_100 > 1.0))
```

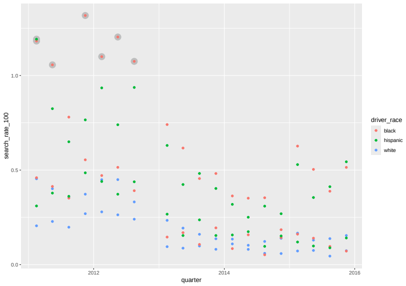
```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_point()
```



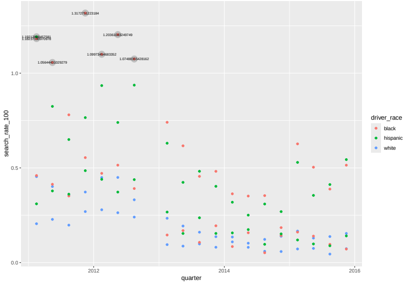
```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_point() +
  geom_point(data = pre_legalization_high, size = 5, color
```



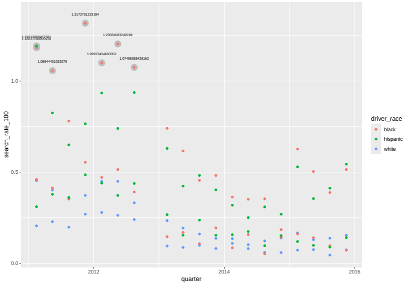
```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_point(data = pre_legalization_high, size = 5, color
  geom_point()
```



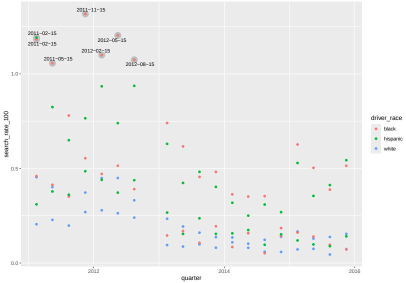
```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_point(data = pre_legalization_high, size = 5, color
  geom_point() +
  geom_text(data = pre_legalization_high, aes(y = search_ra
             size = 2, color = "black")
```



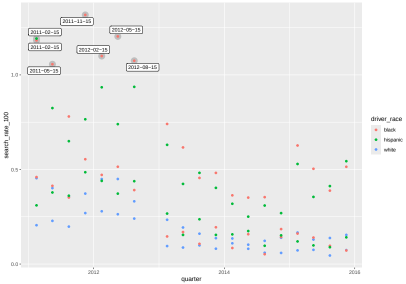
```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_point(data = pre_legalization_high, size = 5, color
  geom_point() +
  geom_text(data = pre_legalization_high, aes(y = search_ra
            size = 2, color = "black")
```



```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_point(data = pre_legalization_high, size = 5, color
  geom_point() +
  geom_text_repel(data = pre_legalization_high,
                  aes(x = quarter, y = search_rate_100,
                      label = as.character(quarter)),
                  size = 3, color = "black")
```



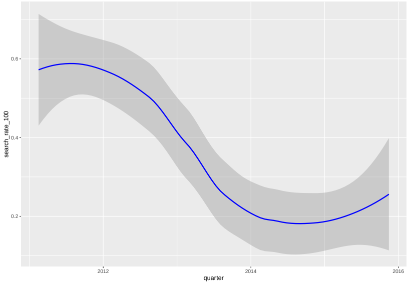
```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_point(data = pre_legalization_high, size = 5, color
  geom_point() +
  geom_label_repel(data = pre_legalization_high,
                   aes(x = quarter, y = search_rate_100,
                       label = as.character(quarter)),
                   size = 3, color = "black")
```



Your turn!

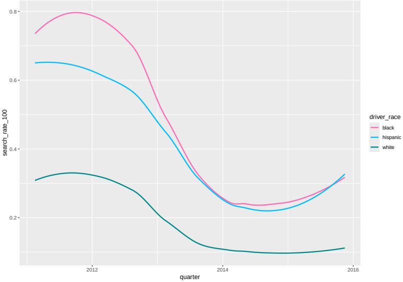
Exercise: How would you fix the following plot?

```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  geom_smooth(color = "blue")
```



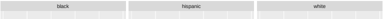
Specifying colors

```
ggplot(stops, aes(x = quarter, y = search_rate_100, color =
  scale_color_manual(values = c("#FF6EB4", "#00BFFF", "#008
  geom_smooth(se = FALSE)
```



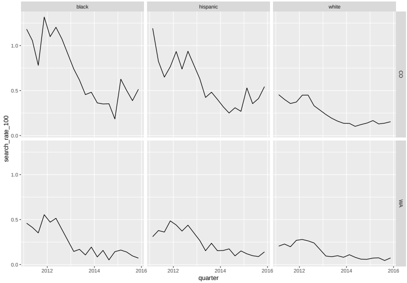
Splitting over facets

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100))
  geom_smooth() +
  facet_wrap( ~ driver_race)
```



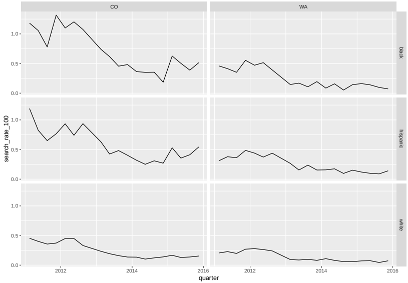
facet_grid

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100))
  geom_line() +
  facet_grid(state ~ driver_race)
```



facet_grid

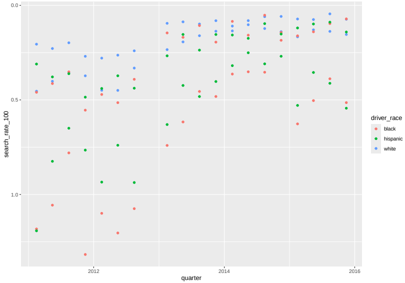
```
ggplot(data = stops, aes(x = quarter, y = search_rate_100))
  geom_line() +
  facet_grid(driver_race ~ state)
```



Scales and legends

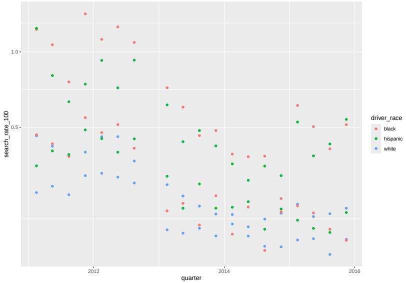
Scale transformation

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point() +
  scale_y_reverse()
```



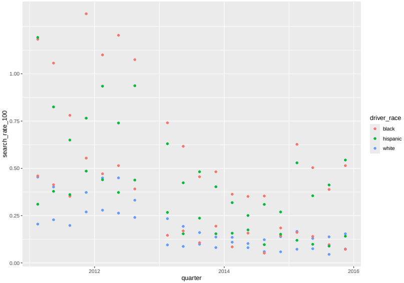
Scale transformation

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point() +
  scale_y_sqrt()
```



Scale details

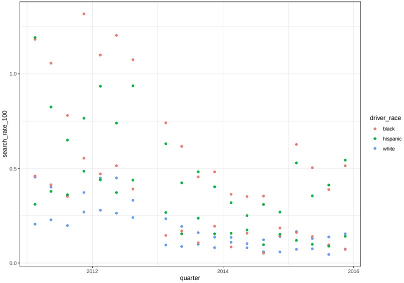
```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point() +
  scale_y_continuous(breaks = c(0, 0.25, 0.5, .75, 1.0))
```



Themes

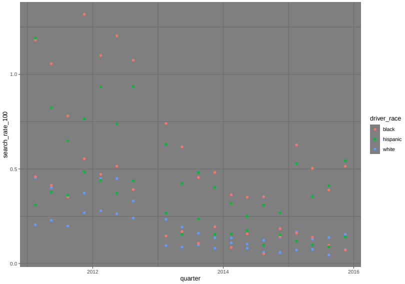
Overall themes

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point() +
  theme_bw()
```



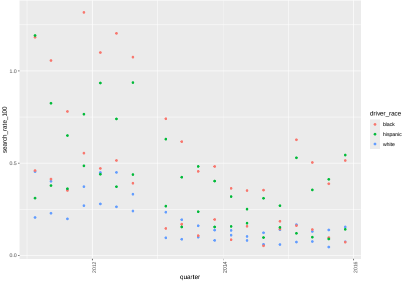
Overall themes

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point() +
  theme_dark()
```



Customizing theme elements

```
ggplot(data = stops, aes(x = quarter, y = search_rate_100,
  geom_point() +
  theme(axis.text.x = element_text(angle = 90))
```

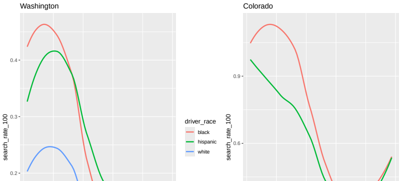


Combining several plots to a grid

```
wa_stops <- stops %>% filter(state == "WA") %>%
  ggplot(aes(x = quarter, y = search_rate_100, color = driv
  geom_smooth(se = FALSE) +
  labs(title = "Washington")

co_stops <- stops %>% filter(state == "CO") %>%
  ggplot(aes(x = quarter, y = search_rate_100, color = driv
  geom_smooth(se = FALSE) +
  labs(title = "Colorado") +
  theme(legend.position = "none")
```

Combining several plots to a grid

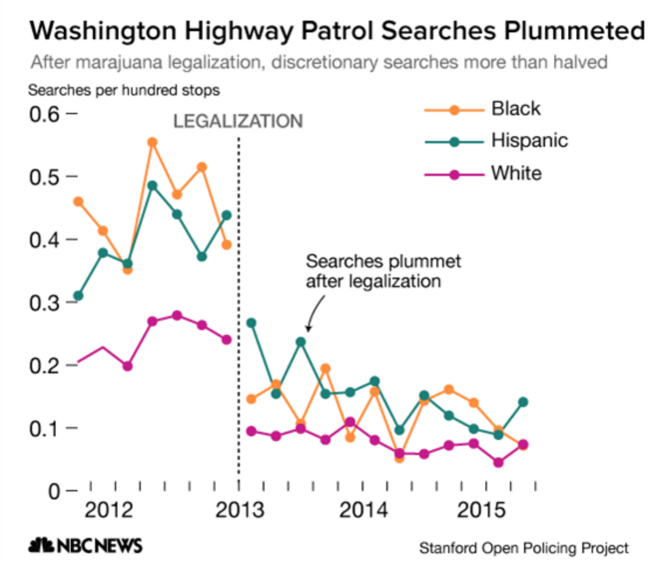
```
wa_stops + co_stops
```



Your turn!

Final Exercise:

Recreate this chart



Starter code:

Recap

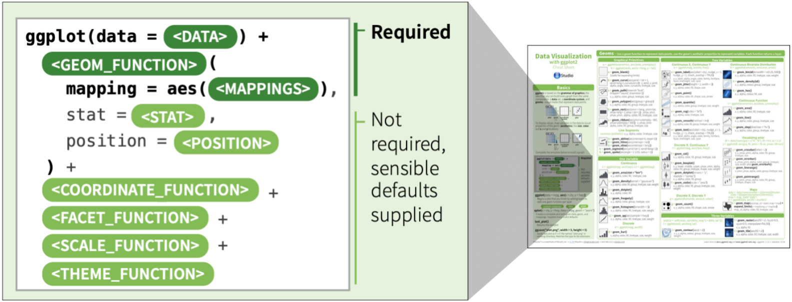
The basics

- map variables to aethestics
- add “geoms” for visual representation layers
- scales can be independently managed
- legends are automatically created
- statistics are sometimes calculated by geoms

ggplot2 template

Make any plot by filling in the parameters of this template

```r
knitr::include_graphics("./img/ggplot2-template.png")
```



```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

Required
Not required, sensible defaults supplied
Data Visualization with ggplot2
Cheat Sheet

Learn more

- Books:
  - R for Data Science by Grolemund and Wickham
  - R Graphics Cookbook by Chang
  - Data Visualization: A Practical Introduction by Healy
- ggplot2.tidyverse.org
- ggplot2 Cheat sheet