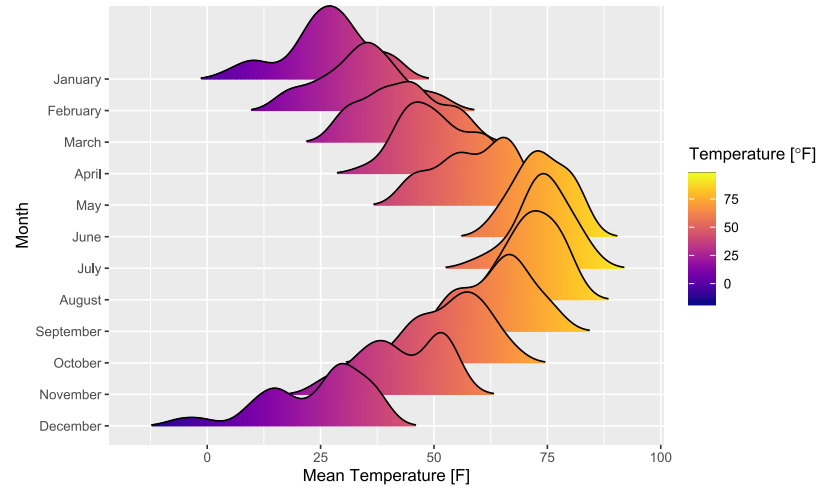




Graphics

Andrew Redd, PhD.

R Bootcamp 2020



Plots

Packages

Must have

- `ggplot2` (<https://cran.r-project.org/package=ggplot2>)
- `viridis` (<https://cran.r-project.org/package=viridis>)

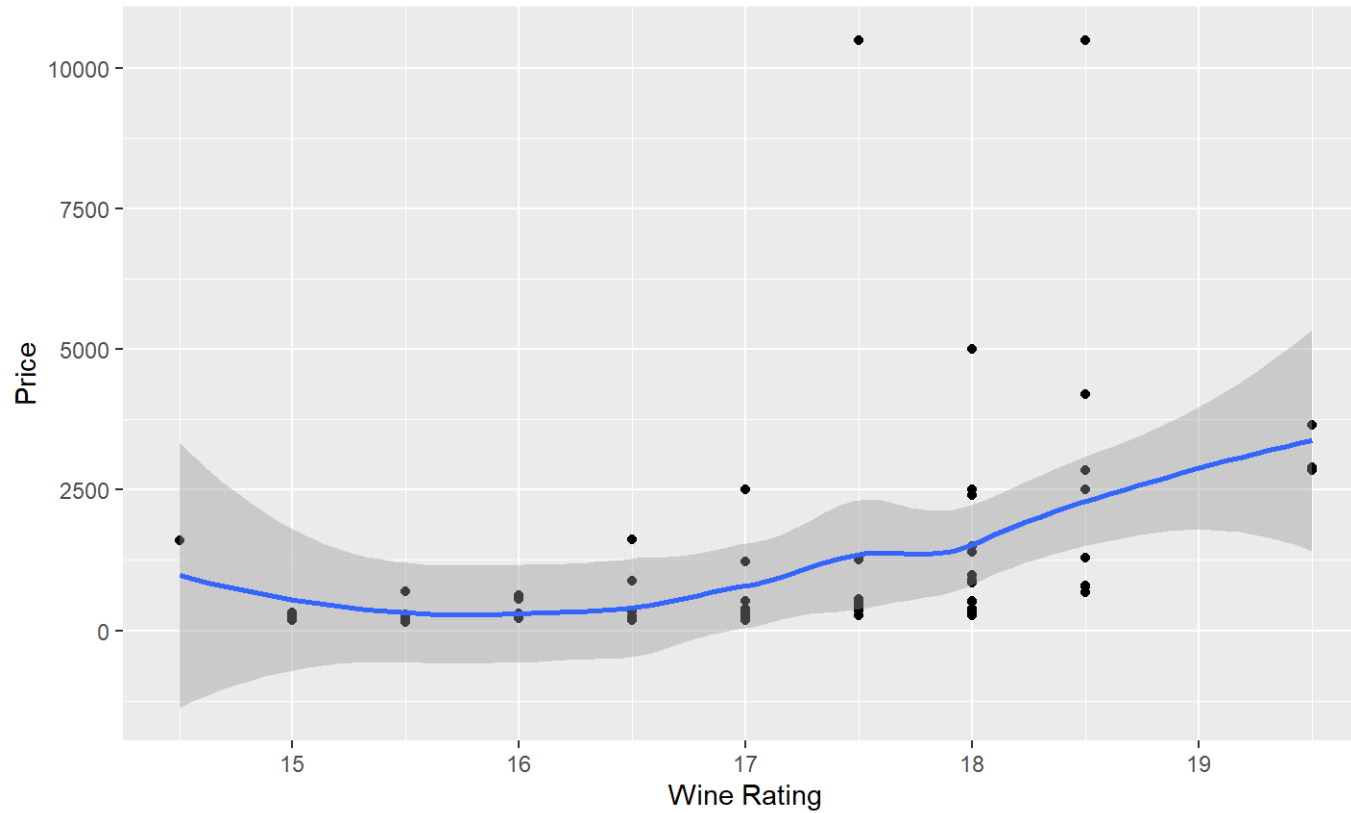
Extras to check out

- `ggribes` (<https://cran.r-project.org/package=ggplot2>)

Thought Exercise:

What are we trying to do with graphics?

Example: What are the components?



Graphics Systems in R

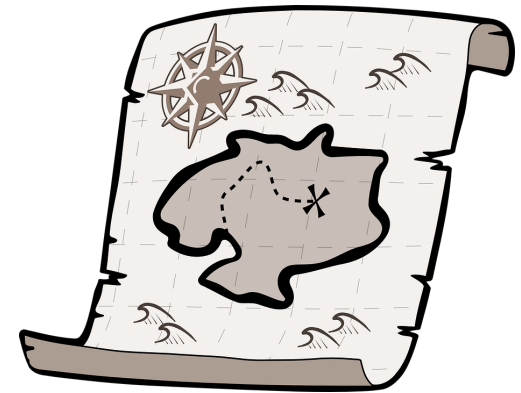
1. Base
2. Grid
3. Grammar

Base Graphics

Don't waste your time.

Grid Graphics

An even bigger waste of time.



Grammar of Graphics

The grammar of graphics is built with different components or blocks.



Building blocks: Aesthetics

- Aesthetics
 - location
 - size
 - shape
 - height
 - width
 - area
 - color
 - fill
 - border
 - transparency
 - ~~time~~/animation

Building 🧱 blocks🧱: Data

- Aesthetics...
- Data
 - map variables from data to aesthetics
 - Statistics, i.e. how to summarize the data:
 - outliers, Q1, Median Q3, outliers → Box plots
 - binning → histograms
 - density → density, violin, etc.
 - smoothing: loess, splines, etc.

Building 🧱 blocks🧱: Coordinates

- Aesthetics...
- Data
- Coordinate System
 - Cartesian
 - Geographic
 - Polar
 - Scaling
 - Logarithmic
 - Square-root

Building 🧱 blocks🧱: Facets

- Aesthetics...
- Data
- Coordinate System
- Facets
 - Grid
 - Strip
 - Wrap

Terminology

Terminology

Data

What we are visualizing

Geometry

The type of visualization.

Mapping

How variables translate variables from data to aesthetic components of the graph.

Statistic

If and how variables are to be summarized prior to being mapped to aesthetics.

Terminology

Terminology

Coordinates

How Axes are displayed.

Facets

How a visualization is to be broken into parts.

Position

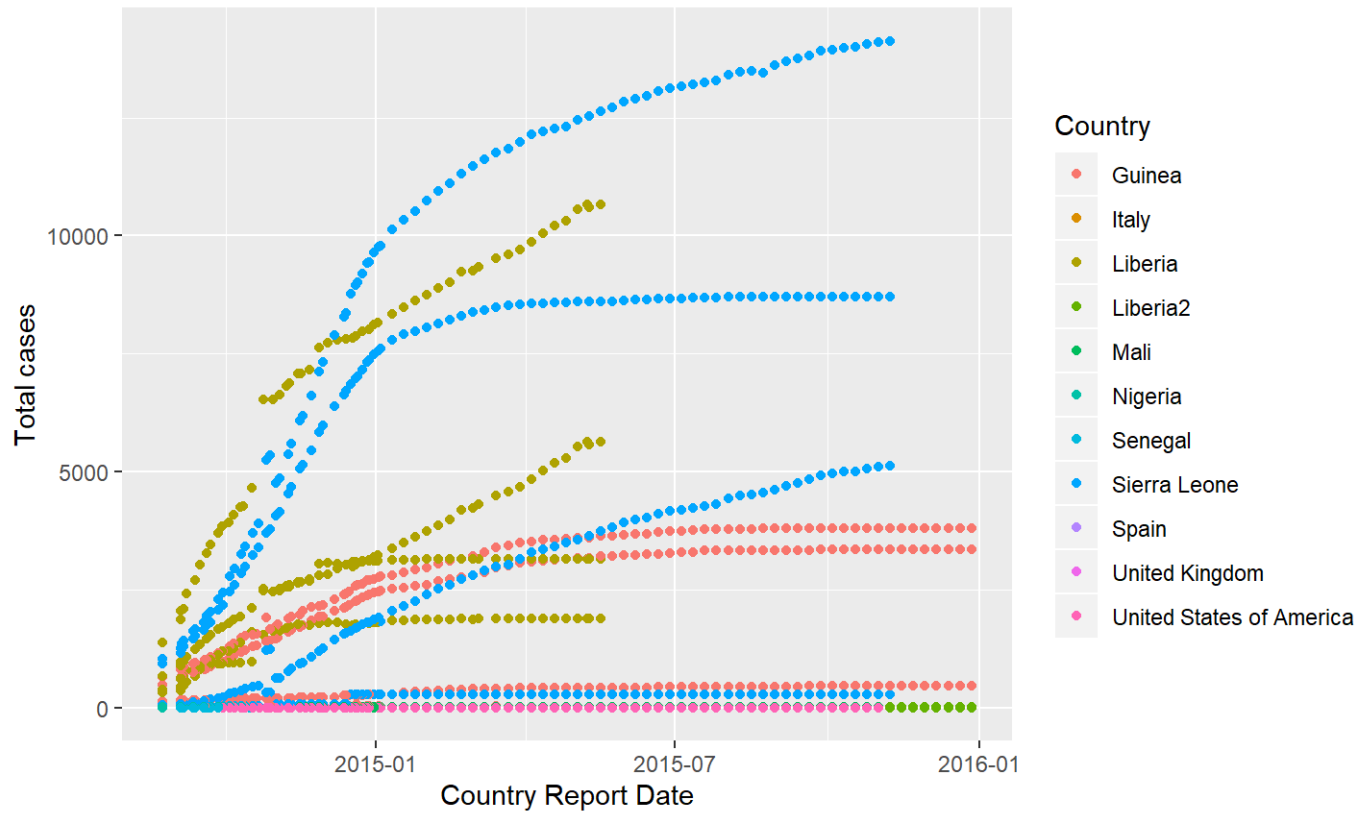
How elements should be positioned; sitter, stacked, dodge, etc.

Example

```
library(ggplot2)
load("data/ebola.data.RData")
ebola.plot1 <-
ggplot(data=ebola.data) +                #< Our 'data' component
  geom_point(                             #< a point geometry layer
    aes( x = `Country Report Date`      #< our mappings
          , y = `Total cases`
          , col = Country
        )
    , stat = 'unique'                    #< our statistic
    , position = 'identity'              #< how to position data
  )
```

Example

ebola.plot1



We can do better.

Layers

Terminology

Layers

Layers are the combination of a geometry, data, mapping, statistic, and position.

Graphs may have and often do have many layers in them.

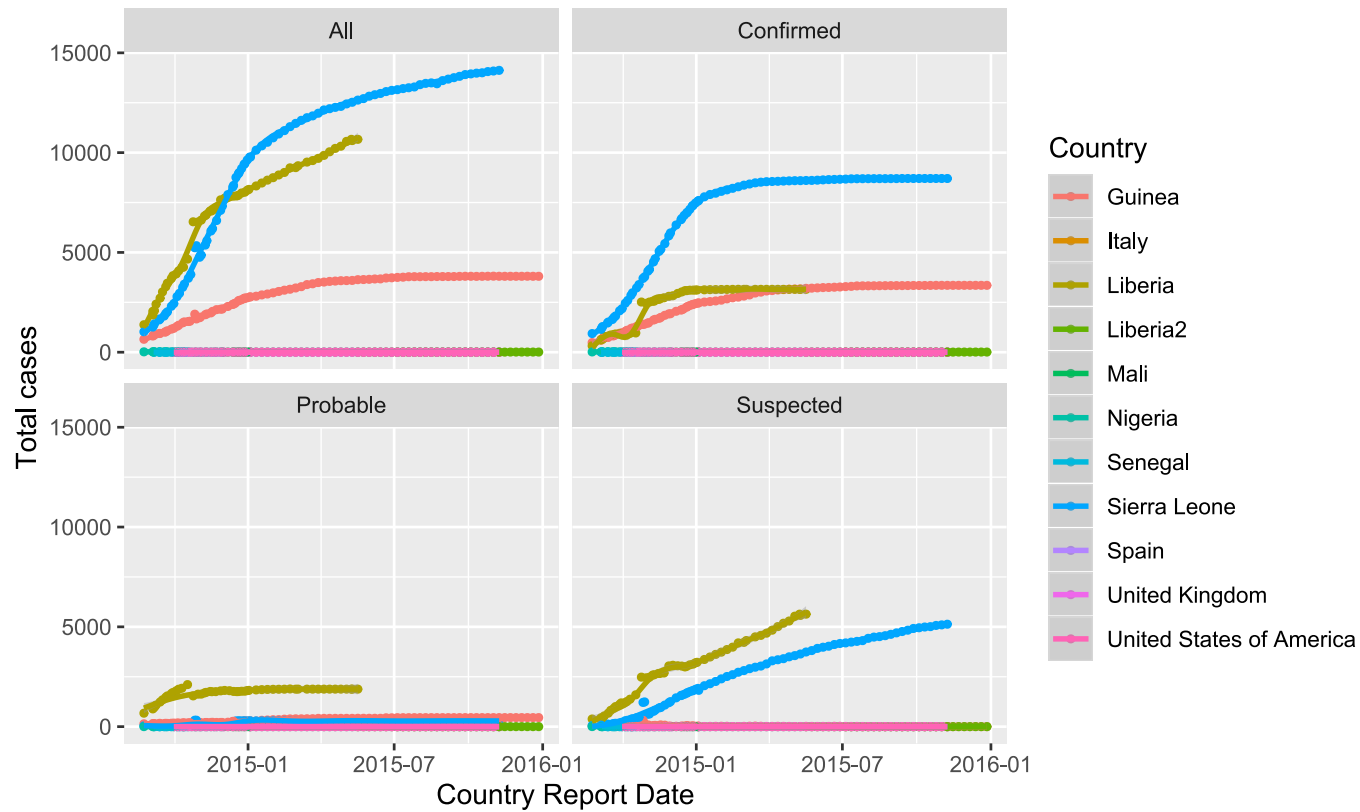


Layered Example

```
library(ggplot2)
load("data/ebola.data.RData")
ebola.plot2 <-
ggplot(data= ebola.data %>% select(2:4, 6) %>% distinct())
  #< Move mappings here because inheritance.
  , aes( x = `Country Report Date`
        , y = `Total cases`
        , col = Country
        )
  ) +
geom_point( stat = 'unique'
           , size = 1           #< An explicit aesthetic.
           ) +
geom_smooth(method='gam', formula=y ~ s(x, bs = "cs")) + #< new 'smooth' layer
facet_wrap(~`Case def.` , 2, 2) #< new 2x2 faceting for Case def.
```

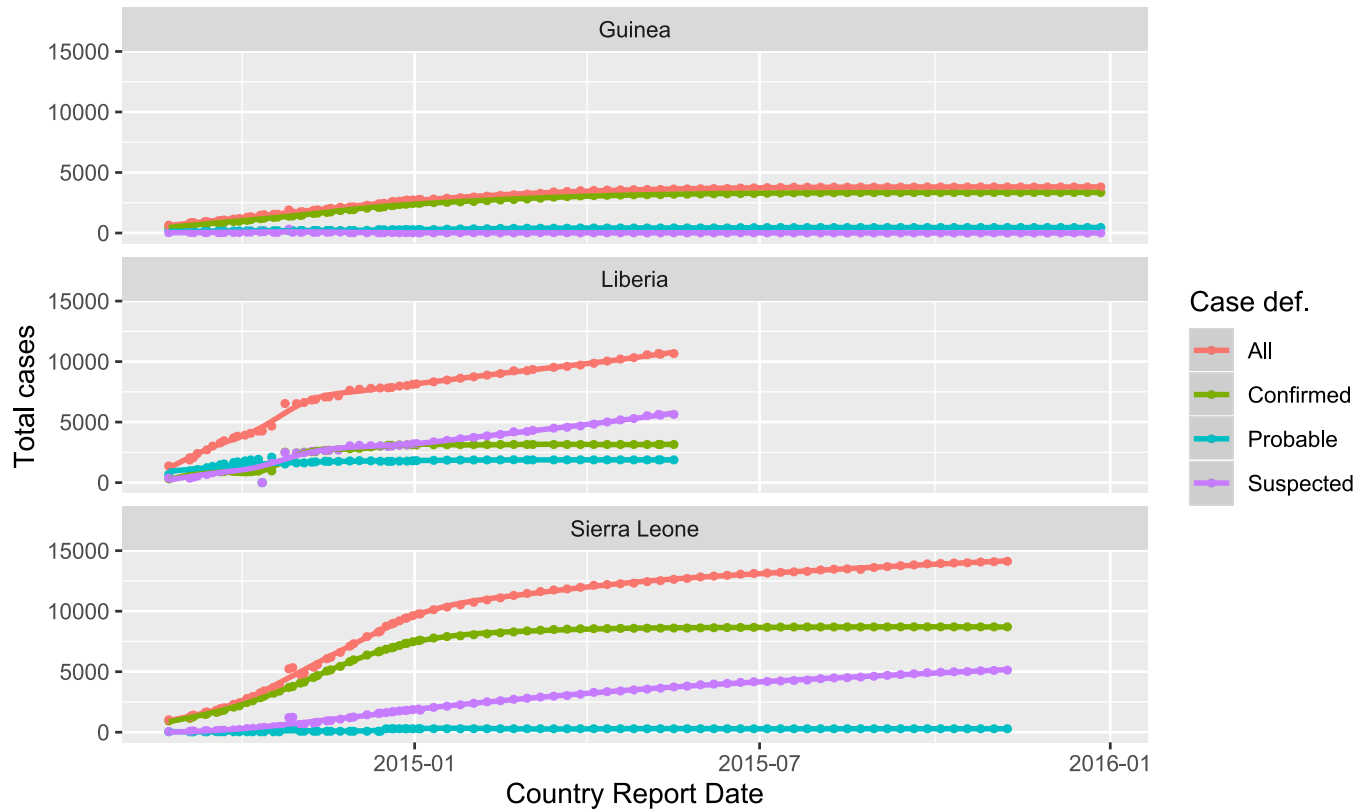
Layered Example

ebola.plot2



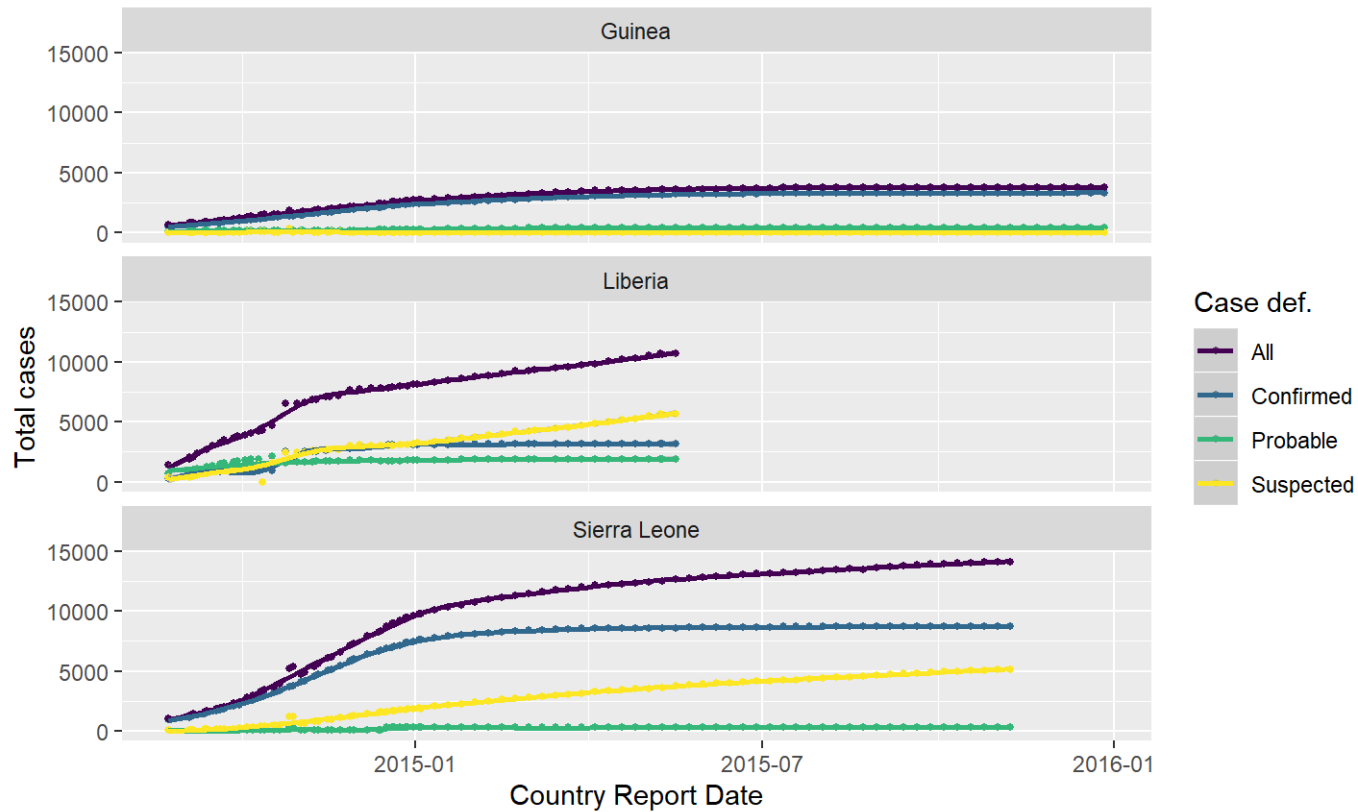
Changing Aesthetics

ebola.plot3



A comment about color

```
ebola.plot3 + scale_color_viridis_d()
```



Themes

Themes are what control the non-data elements of a visualization

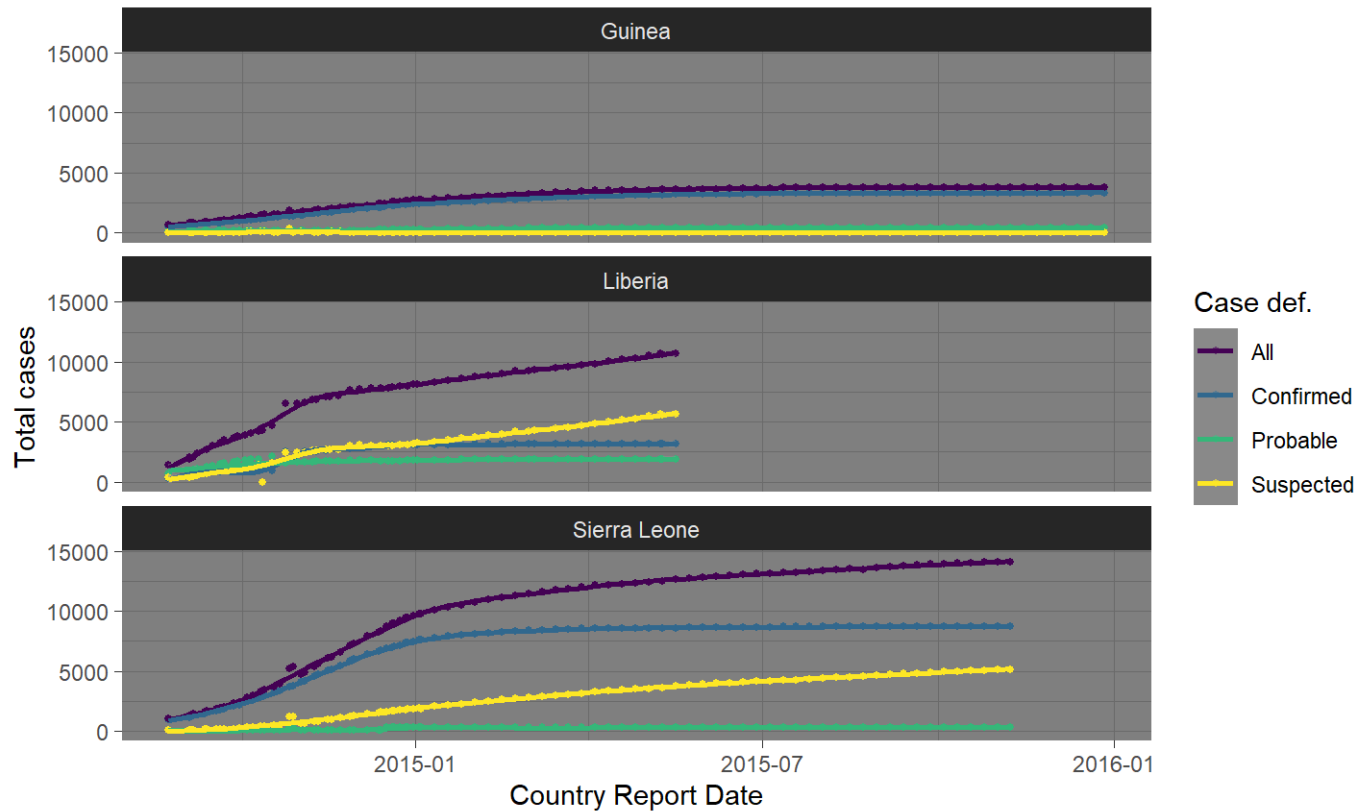
- Axes
- Grid
- Background
- Legend positioning, etc.

Built-in themes

- | | |
|-------------------------------|------------------------------|
| • <code>theme_grey</code> | • <code>theme_dark</code> |
| • <code>theme_bw</code> | • <code>theme_minimal</code> |
| • <code>theme_linedraw</code> | • <code>theme_classic</code> |
| • <code>theme_light</code> | |

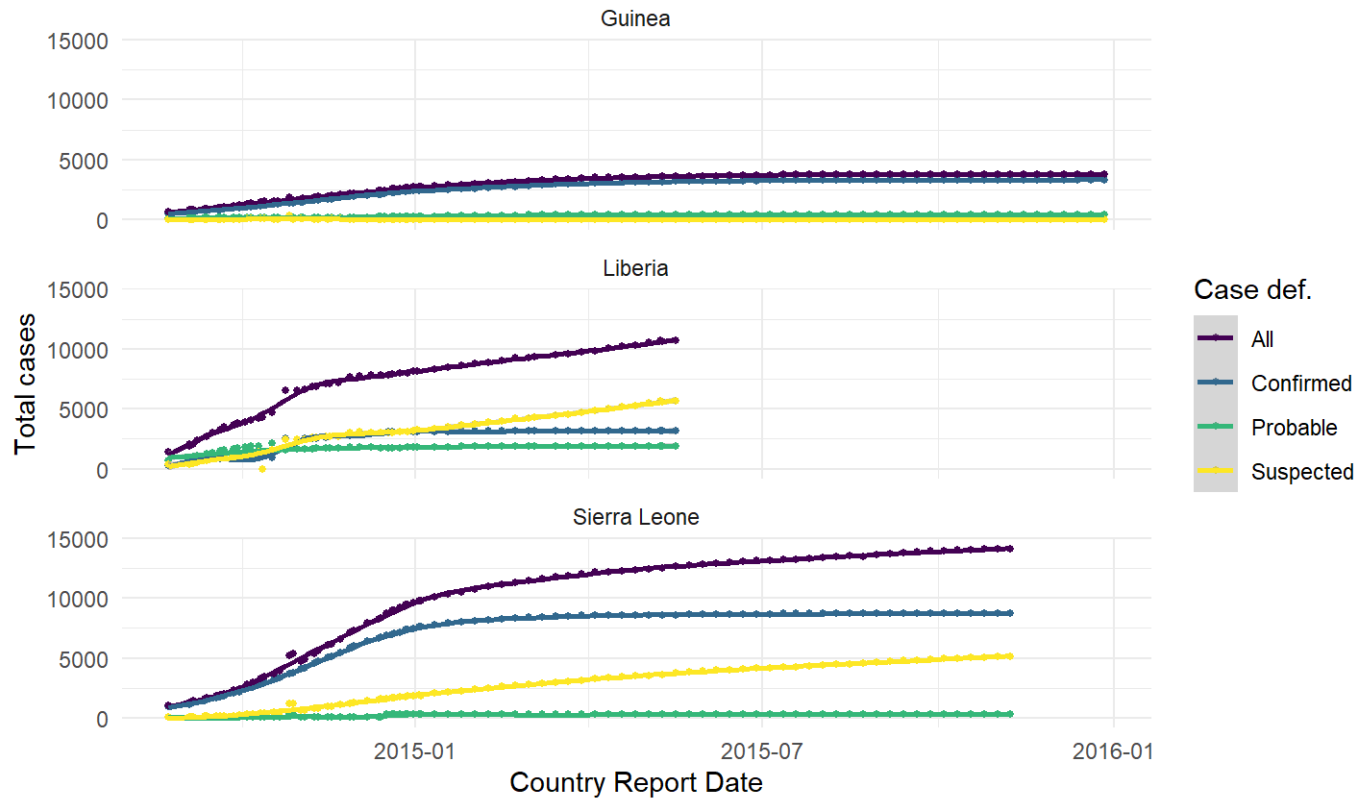
Theme Example, Dark

```
ebola.plot3 + scale_color_viridis_d() + theme_dark()
```



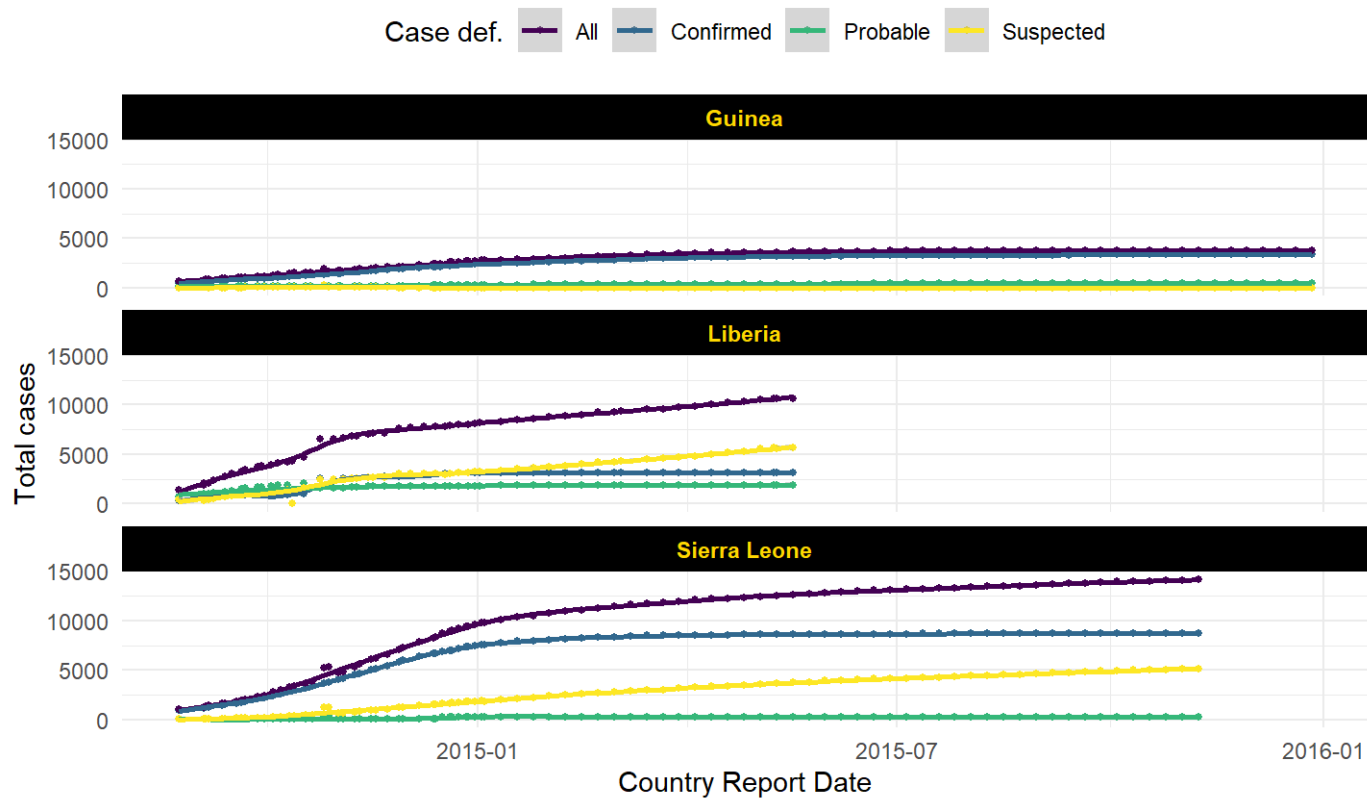
Theme Example, Minimal

```
ebola.plot3 + scale_color_viridis_d() + theme_minimal()
```



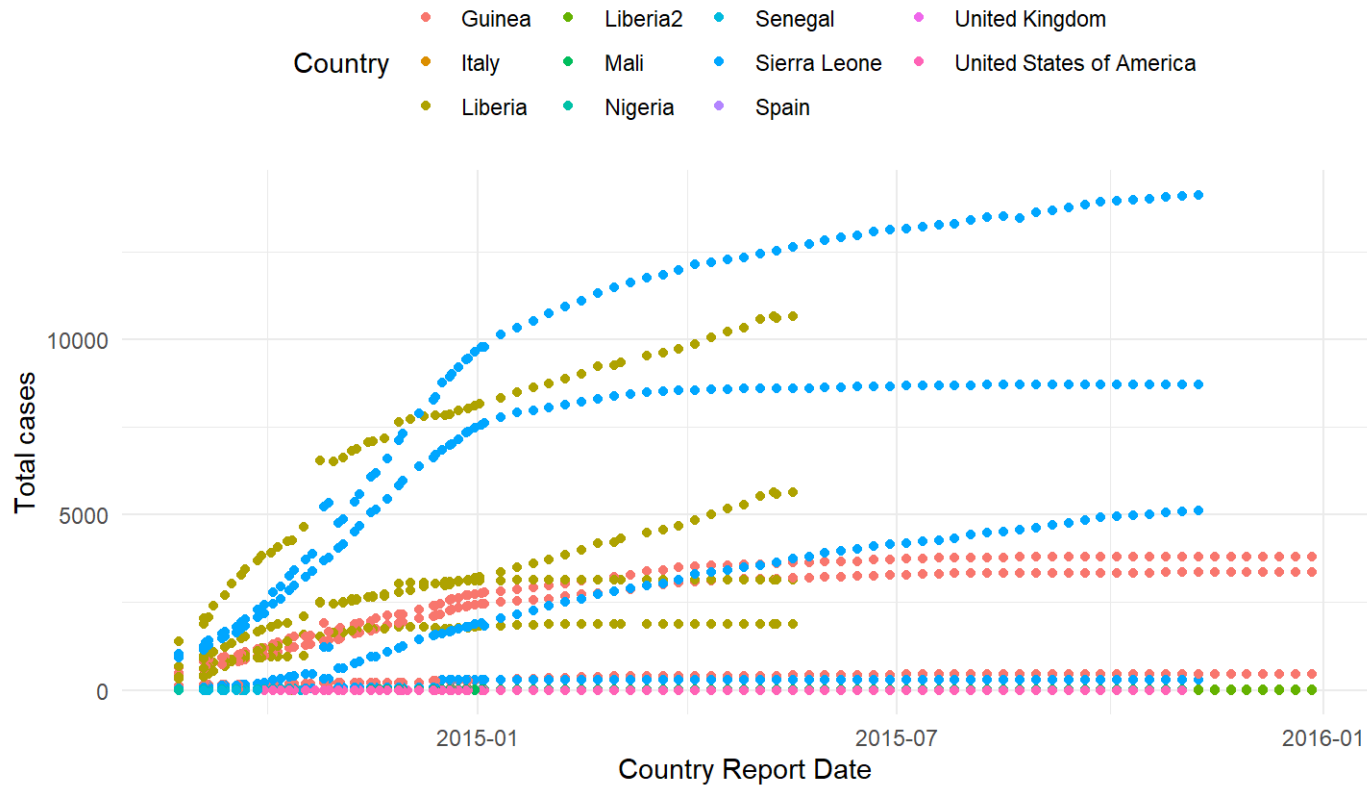
Customizing themes

```
ebola.plot3 + scale_color_viridis_d() + theme_minimal() +  
  theme(legend.position="top"  
        , strip.background=element_rect(fill='black')  
        , strip.text= element_text(color='gold', face='bold'))
```



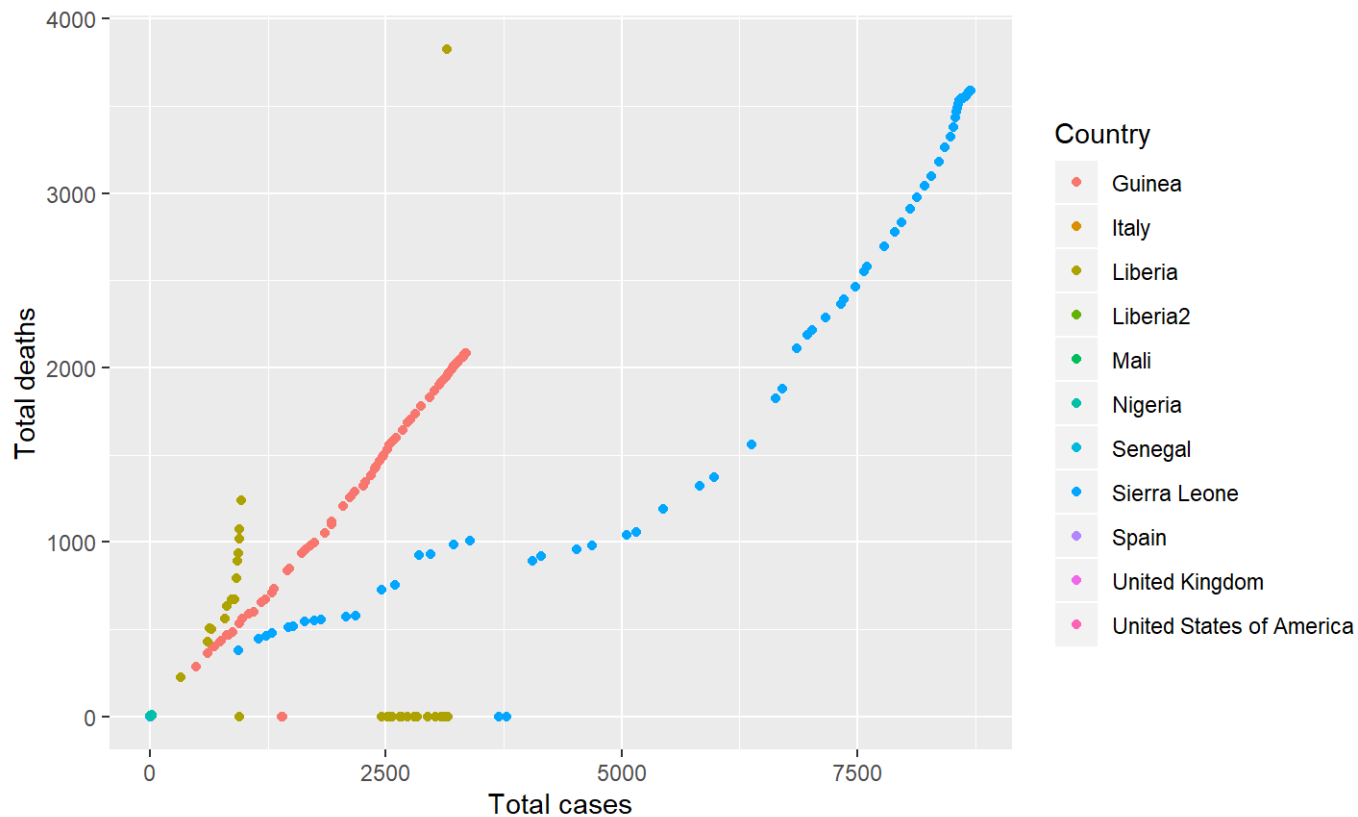
Reuse your theme

```
my.theme <- theme_minimal() + theme(legend.position="top",  
  strip.background=element_rect(fill='black'),  
  strip.text= element_text(color='gold', face='bold'))  
ebola.plot1+my.theme
```



The Shortcut `qp1ot` (quick plot)

```
plotting.data <- ebola.data %>% filter(`Case def.`=='Confirmed')
qplot(`Total cases`, `Total deaths`, color = Country, data=plotting.data)
```



Exercises with Bordeaux Wines

1. Parker Points vs. Coates Points
2. Parker Points or Coates Points vs. Price
3. Price by First Growth or Cult Wine
4. A plot that includes at least
 1. Price
 2. Parker Points or Coates Points
 3. two of First Growth, Cult Wine, Pomerol, or Vintage Superstar

10:00

Solution: Preliminary

Fix the wine data so that the labels for indicators are a little nicer.

```
library(magrittr)
```

```
Message:##  
## Attaching package: 'magrittr'
```

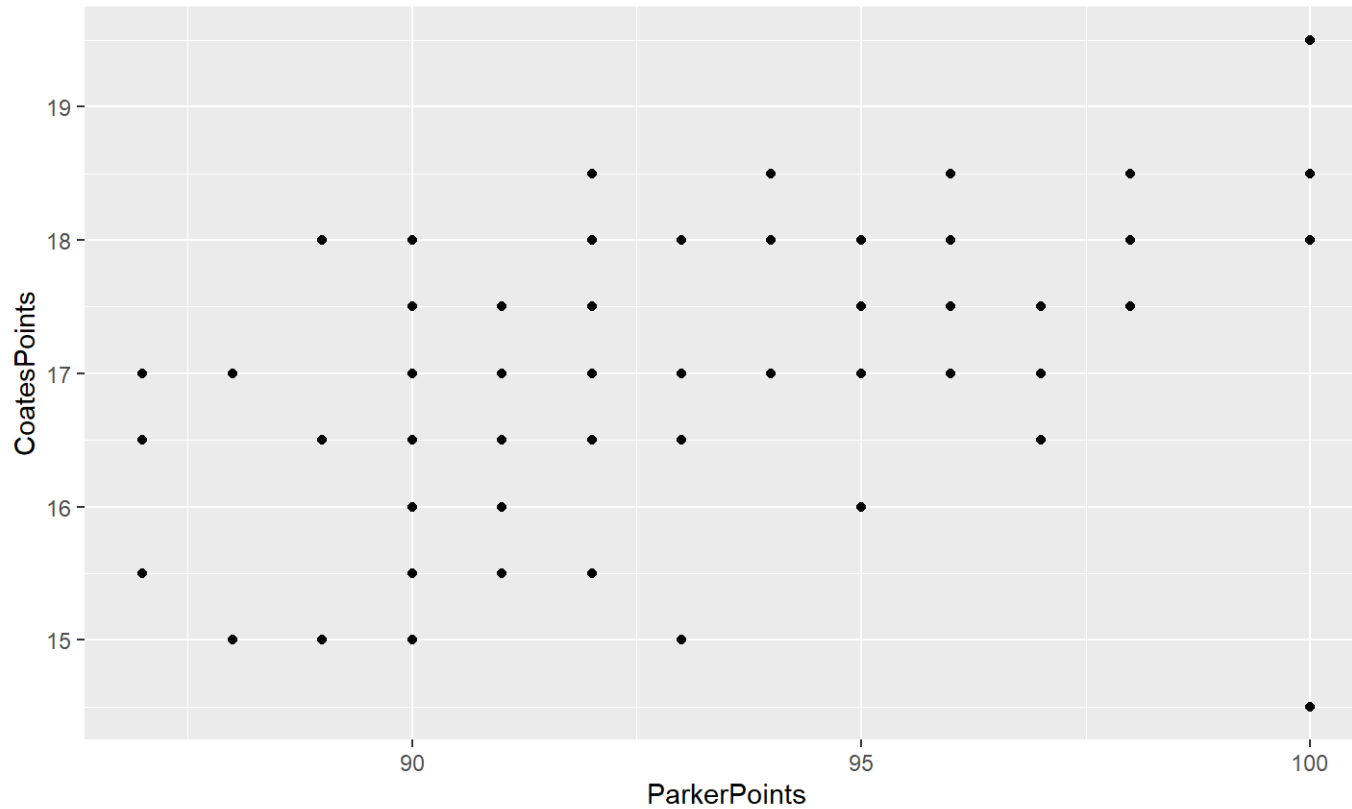
```
Message:## The following object is masked from 'package:purrr':  
##  
##      set_names
```

```
Message:## The following object is masked from 'package:tidyr':  
##  
##      extract
```

```
#assign back pipe  
wine <- read.csv("data/Bordeaux.csv") %>%  
  mutate_at( vars(FirstGrowth:VintageSuperstar)  
             , ~forcats::fct_recode(factor(.), No='0', Yes='1')  
             )
```

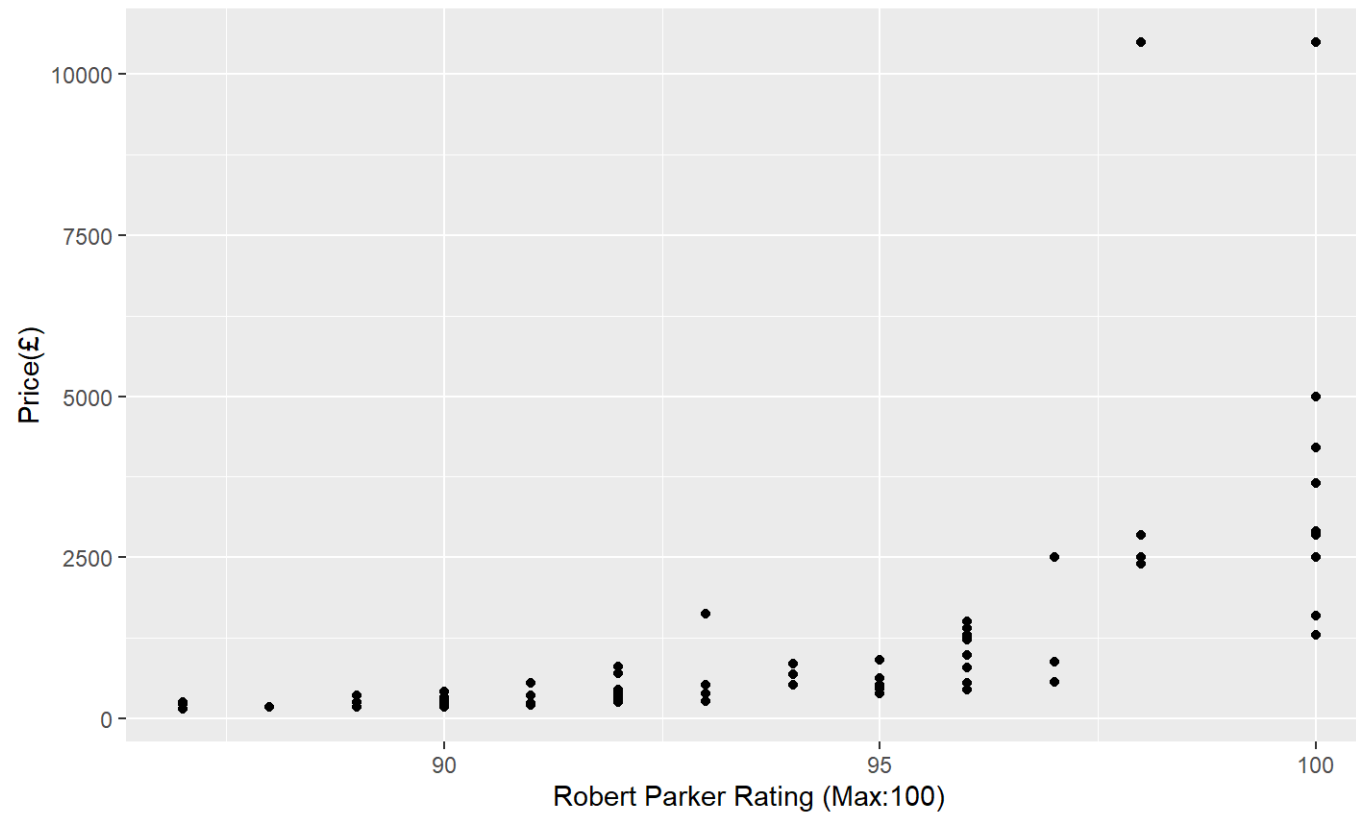
Solution: Parker Points vs. Coates Points

```
qplot(data=wine, ParkerPoints, CoatesPoints, geom='point')
```



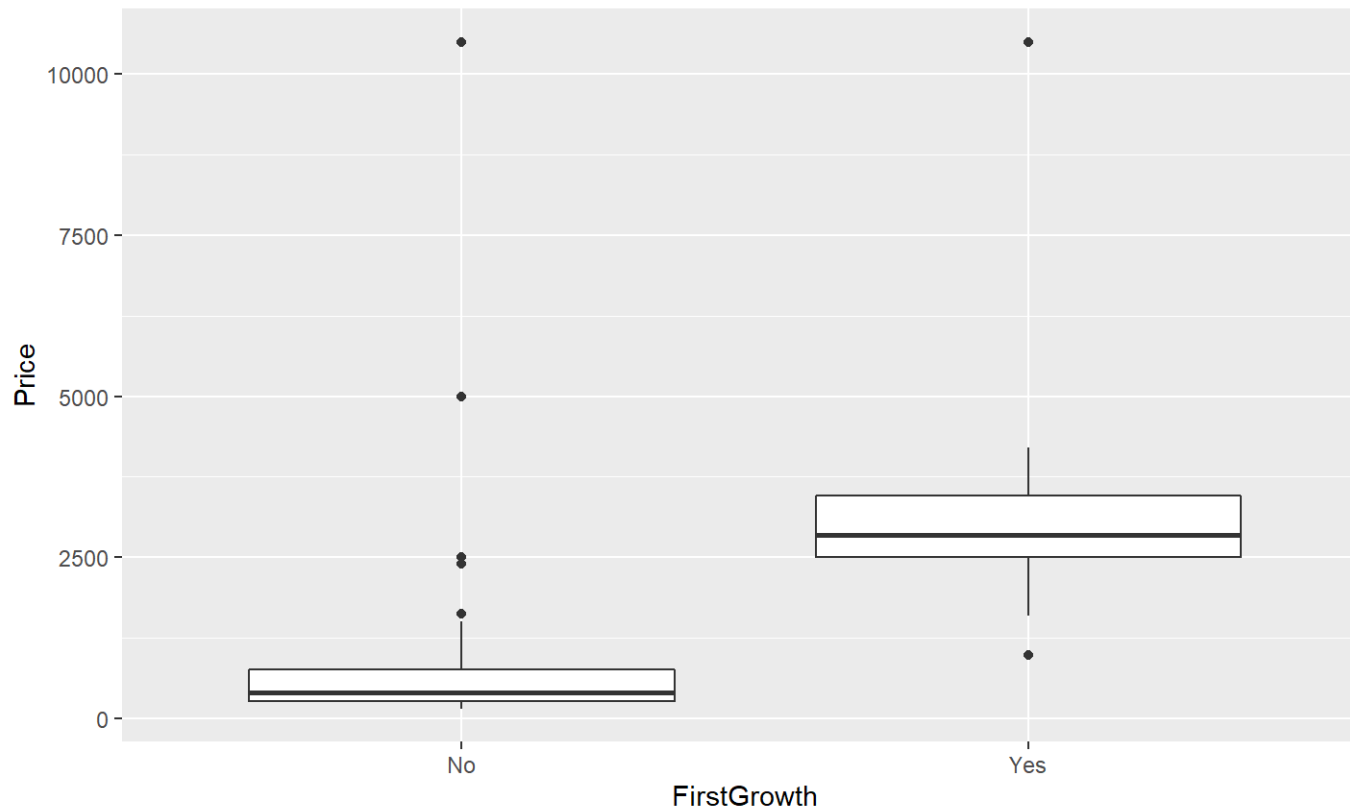
Solution: Parker Points vs. Price

```
qplot(data=wine, ParkerPoints, Price, geom='point') +  
  labs(y = 'Price(\uA3)', x="Robert Parker Rating (Max:100)")
```



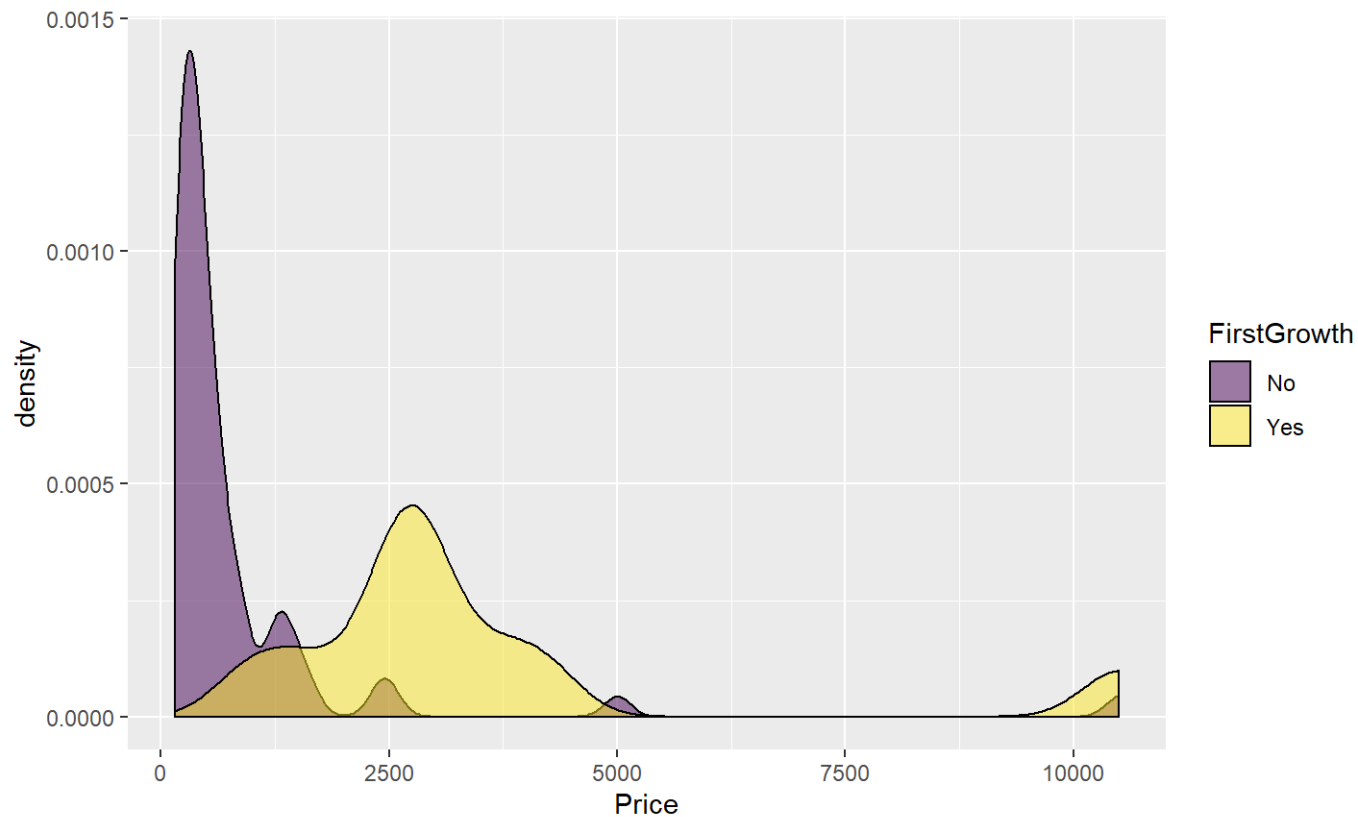
Solution: Price by First Growth Box plot

```
qplot(data=wine, x=FirstGrowth, y=Price  
      , geom='boxplot', group=FirstGrowth) #< must include a group.
```



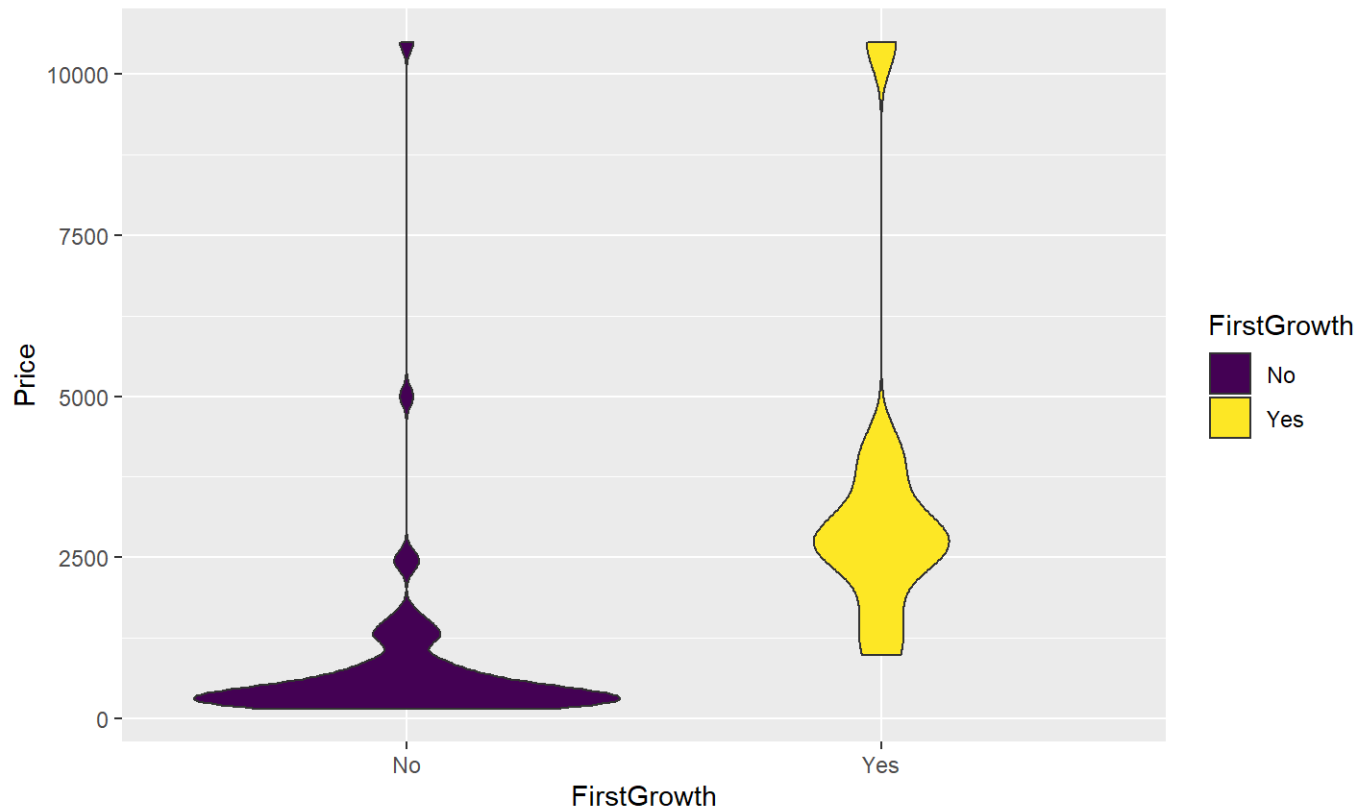
Solution: Price by First Growth Density

```
ggplot(data=wine) + geom_density(aes(x=Price, fill=FirstGrowth), alpha=0.5)+  
  scale_fill_viridis_d()
```



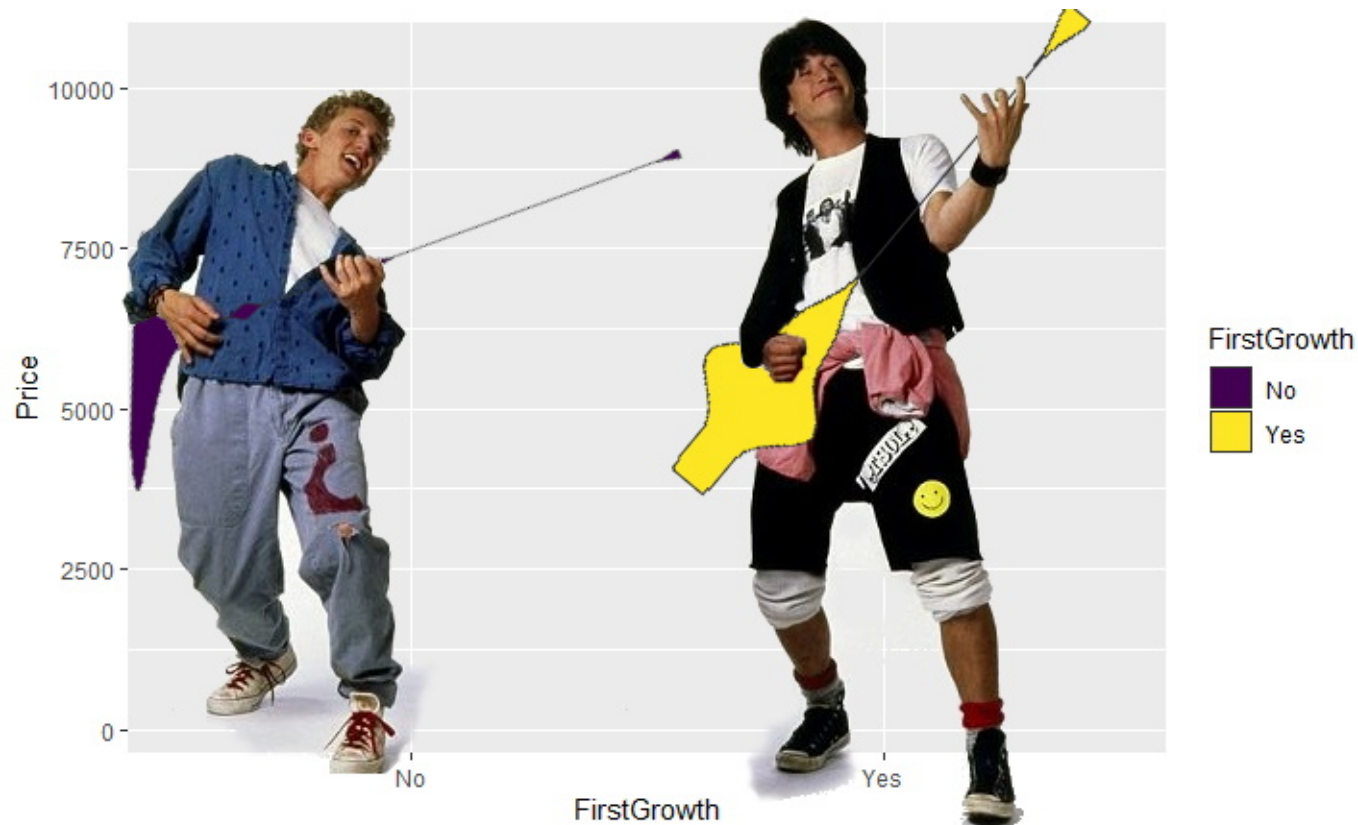
Solution: Price by First Growth Violins

```
ggplot(data=wine) + geom_violin(aes(y=Price, x=FirstGrowth, fill=FirstGrowth))+  
  scale_fill_viridis_d()
```



Solution: Price by First Growth Guitar Plot

```
ggplot(data=wine) + geom_violin(aes(y=Price, x=FirstGrowth, fill=FirstGrowth))+  
  scale_fill_viridis_d()
```



Solution: all the variables.

```
all.the.variables <- ggplot(data=wine) +  
  # Five variables here.  
  geom_point(aes( x    = ParkerPoints, y    = CoatesPoints  
                 , col = Price, shape = Pomerol:VintageSuperstar  
                 )  
             # Jitter the position to avoid over-plotting  
             , position=position_jitter(.4, .4))+  
  # grid for two more variables  
  facet_grid( FirstGrowth~CultWine  
             , labeller=label_both  
             ) + #^ Labels have both variable name and level.  
  # Make colors pop for high value wines.  
  scale_color_viridis_c(trans = 'log10', option="inferno", end=0.75) +  
  labs( x = "Robert Parker Rating (Max:100)"  
        , y = "Clive Coates Rating (Max:20)"  
        , color = 'Price(\uA3)')
```

Solution: all the variables.

all.the.variables

