Introduction to ggplot2

Andy Grogan-Kaylor

March 17, 2021

Table of Contents

# Background

[R](https://www.r-project.org/foundation/) has a number of graphing libraries, including *base* graphics that are installed whenever you install R.

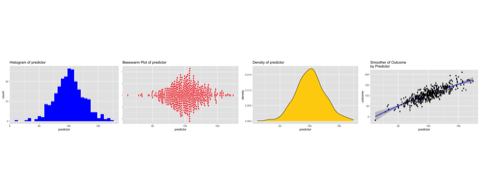
[ggplot2](http://ggplot2.org/), is a graphing library in R that makes beautiful graphs. ggplot2 graph syntax can be formidably complex, with a somewhat steep learning curve.

That being said, learning ggplot2 is worth the effort for a couple of reasons. First, the graphs are beautiful. Second, ggplot2’s syntax, though seemingly arcane at times, forces you to think about the nature of your data, and the ideas that you are graphing. Lastly, a little bit of knowledge about ggplot2 can go a long way, and can build a powerful foundation for future learning.

# ggplot in 3 easy steps (maybe 2 easy steps)

## **aesthetic**: *what* you want to graph (e.g. x, y, z).

## **geom**: *how* you want to graph it.



## **options**: *optional* titles, themes, etc.

# A Simple Quick Example

The intent of this tutorial is to build the foundation of this idea that:

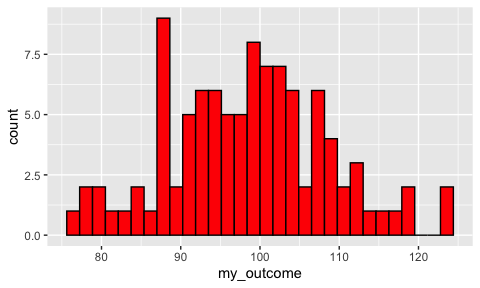
A little bit of ggplot can go a long way

and to give you a simple introduction to the idea that any ggplot graph is composed of:

an aesthetic + a geom or two + other optional elements like titles and themes.

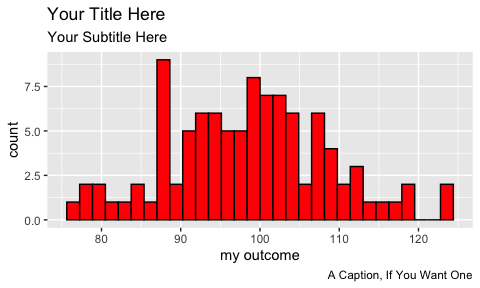
So, as a quick and simple example…

library(ggplot2)  
  
ggplot(my\_demo\_data, # the data that I am using  
 aes(x = my\_outcome)) + # aesthetic: what I am graphing  
 geom\_histogram(fill = "red", # geom: how I am graphing it  
 color = "black") # options: fill = "red"; color = "black"



And now, with labels…

ggplot(my\_demo\_data, # the data that I am using  
 aes(x = my\_outcome)) + # aesthetic: what I am graphing  
 geom\_histogram(fill = "red", # geom: how I am graphing it  
 color = "black") +  
 labs(title = "Your Title Here",  
 subtitle = "Your Subtitle Here",  
 caption = "A Caption, If You Want One",  
 x = "my outcome",  
 y = "count")



This document is a *very brief* introduction to the *basic* ideas of ggplot2. More information about ggplot can be found [here](http://ggplot2.org/). More ggplot2 examples can be found [here](http://www-personal.umich.edu/~agrogan/how_to_choose_a_chart/how_to_choose_a_chart_v3.html).

# Call The Relevant Libraries

You will need a few [R libraries](http://www-personal.umich.edu/~agrogan/R/introduction-to-R.html#2_base_r_and_libraries) to work in ggplot. You may *only* need library(ggplot2), but some of these other libraries may also be helpful.

library(ggplot2) # beautiful graphs  
  
library(ggthemes) # nice themes for ggplot2  
  
library(ggbeeswarm) # "beeswarm" plots  
  
library(cowplot) # arrrange graphs  
  
library(pander) # nice tables  
  
library(psych) # nice table of descriptive statistics

# Simulated Data

In this example, we simulate some data. But your own learning of ggplot will progress more quickly if you use data that you have access to, on an issue that you care about.

Here are the first few rows of simulated data:

|  |  |  |
| --- | --- | --- |
| predictor | outcome | group |
| 130.5 | 128 | 0 |
| 115.3 | 119 | 0 |
| 65.64 | 69.22 | 0 |
| 88.34 | 75.46 | 0 |
| 88.65 | 88.63 | 1 |
| 77.41 | 104.7 | 0 |
| 104.8 | 110.8 | 1 |
| 62.25 | 68.51 | 0 |
| 158.7 | 146.3 | 0 |
| 116.9 | 173.8 | 0 |

# The Essential Idea Of ggplot2 Is Simple.

There are 3 essential elements to any ggplot call:

1. An *aesthetic* that tells ggplot which variables are being mapped to the *x axis*, *y axis*, (and often other attributes of the graph, such as the *color fill*). Intuitively, the aesthetic can be thought of as *what you are graphing*.
2. A *geom* or *geometry* that tells ggplot about the basic structure of the graph. Intuitively, the geom can be thought of as *how you are graphing it*.
3. Other options, such as a *graph title*, *axis labels* and *overall theme* for the graph.

## ggplot2 Starts By Calling The aesthetic

For one variable:

p <- ggplot(mydata, aes(x = ...)) This says there is only one variable running along the horizontal *x* axis in the aesthetic.

The p <-... means that we are *assigning* this graph aesthetic to plot *p*. We can then add other features to plot *p* as we continue our work. This *iterative* nature of ggplot2 is one of the things that makes it so powerful. As your workflow and your documents become more complex, you can build a simple consistent foundation[[1]](#footnote-37) for your graphs, then add something simple to make a first graph, and a different something simple to make a second graph.

For two variables:

p <- ggplot(mydata, aes(x = ..., y = ...)) This says there are two variables: one for the horizontal *x* axis; and another for the vertical *y* axis, in the aesthetic.

## We Then Call The geometry

We can then add different geometries to our plot:

For one variable:

+ geom\_density() This says add a density geometry to the graph.

+ geom\_histogram() This says add a histogram geometry to the graph.

+ geom\_violin() This says add a *violin plot* geometry to the graph.

+ geom\_beeswarm() This says add a *beeswarm* geometry to the graph.

A *beeswarm* is a creative layout of points that intuitively lets you understand the distribution of a quantity. The *beeswarm* geometry requires separate installation of the ggbeeswarm package. You also need to call library(ggbeeswarm) to use this geometry.

For two variables:

+ geom\_point() This says add a point (scatterplot) geometry to the graph.

+ geom\_smooth() This says add a smoother to the graph.

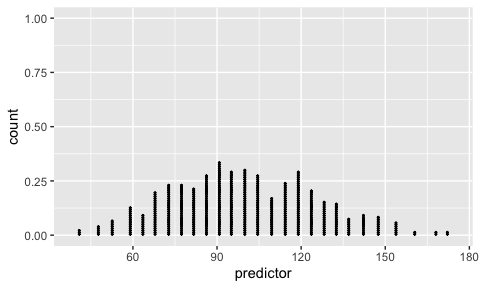
# Examples

## One Continuous Variable At A Time

### Dotplot

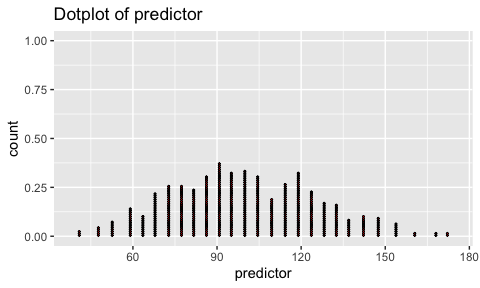
# call ggplot2 where aesthetic is: x uses our predictor variable  
  
p1 <- ggplot(mydata,   
 aes(x = predictor))

p1 +   
 geom\_dotplot(dotsize = .15) # add dotplot geom



### Add Some Options

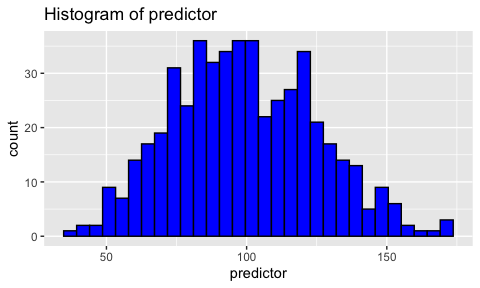
p1 +   
 geom\_dotplot(dotsize = .15,   
 fill="red") + # add dotplot geom in red  
 labs(title ="Dotplot of predictor") # Add title



### Different Geoms

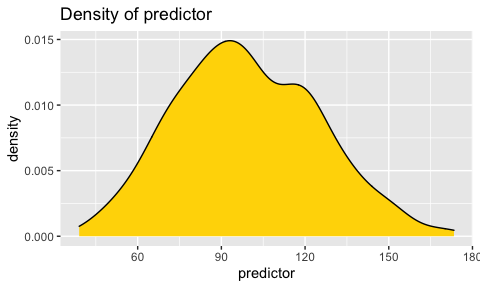
#### Histogram

p1 + geom\_histogram(fill = "blue",   
 color="black") + # add histogram geom in blue  
 labs(title ="Histogram of predictor") # Add title



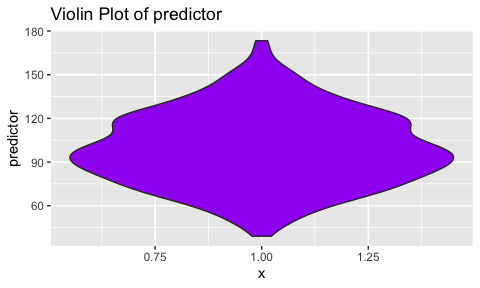
#### Density

p1 + geom\_density(fill = "gold") + # add density geom in gold  
 labs(title ="Density of predictor") # Add title



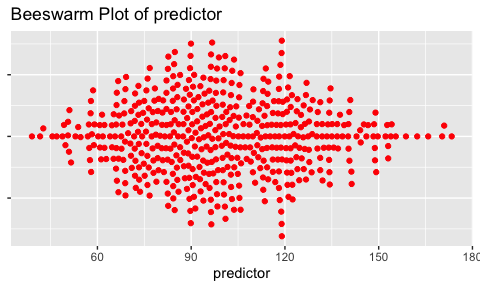
#### Violin Plot

p2 <- ggplot(mydata,   
 aes(x = 1, # we need an aesthetic with \_x\_  
 y = predictor)) # & \_y\_  
  
p2 + geom\_violin(fill = "purple") +  
 labs(title ="Violin Plot of predictor") # Add title



#### Beeswarm

p3 <- ggplot(mydata,   
 aes(x = predictor, # we need an aesthetic with \_x\_  
 y = 1)) # & \_y\_  
  
p3 + geom\_beeswarm(color = "red",   
 groupOnX = FALSE) +  
 labs(title = "Beeswarm Plot of predictor") + # Add title  
 theme(axis.title.y = element\_blank(),   
 axis.text.y = element\_blank()) # tweak y axis

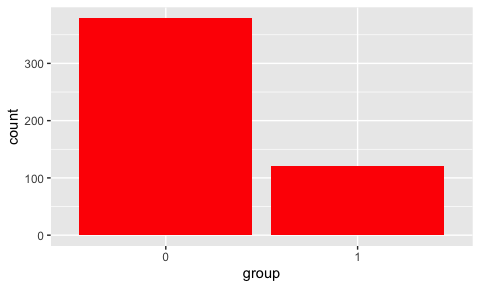


## One Categorical Variable at a Time

The easiest way to represent a single categorical variable is likely a bar graph.

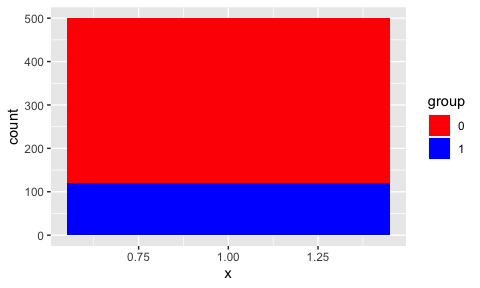
Here bars represent the **count** of observations in each group.

p\_barchart <- ggplot(mydata,  
 aes(x = group)) +  
 geom\_bar(fill = "red")   
  
p\_barchart



Changing the *aesthetic* slightly results in a *stacked* bar chart. Since all groups are stacked in 1 bar, we have to add information about the colors that we want to use to distinguish the groups.

p\_stacked\_barchart <- ggplot(mydata,   
 aes(x = 1,  
 fill = group)) +  
 geom\_bar() +  
 scale\_fill\_manual(values = c("red", "blue"))  
  
p\_stacked\_barchart

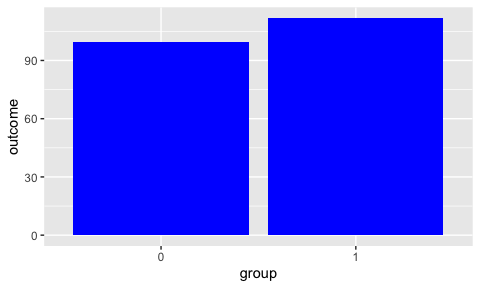


## A Categorical Variable and A Continuous Variable

### Barchart

Here bars represent the **average value of our outcome variable** for members of each group.

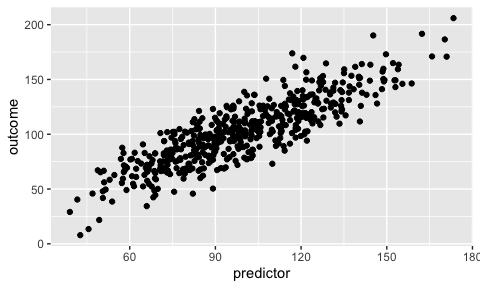
p\_barchart\_of\_mean <- ggplot(mydata,   
 aes(x = group, # slightly different aesthetic  
 y = outcome)) +   
 stat\_summary(fun = mean, # take the mean of the data  
 fill = "blue", # fill color  
 geom = "bar") # we want to summarize data with bars  
  
p\_barchart\_of\_mean



## Two Continuous Variables At A Time

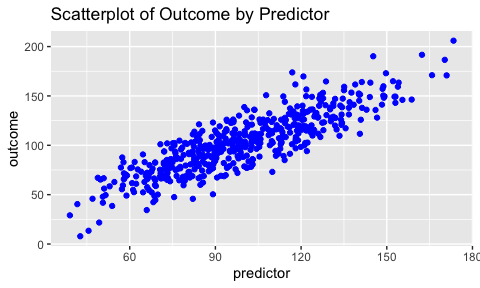
### Basic Scatterplot

# call ggplot2 where aesthetic uses both predictor and outcome  
  
p4 <- ggplot(mydata,   
 aes(x = predictor,   
 y = outcome)) # set up aesthetic  
  
p4 + geom\_point() # add point geom (scatterplot)



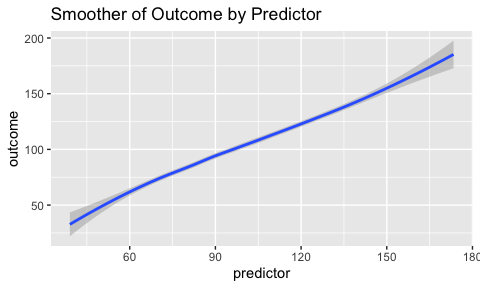
### Add Some Options

p4 + # start with basic plot that has only an aesthetic  
 geom\_point(color = "blue") + # add point geom in blue  
 labs(title ="Scatterplot of Outcome by Predictor") # add title



### Try A Smoother

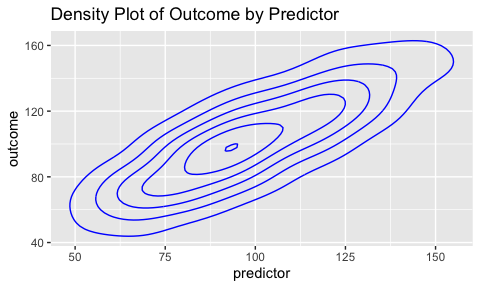
p4 +   
 geom\_smooth() + # add smooth geom  
 labs(title ="Smoother of Outcome by Predictor") # add title



### Try A Density Plot

#### Simple Density

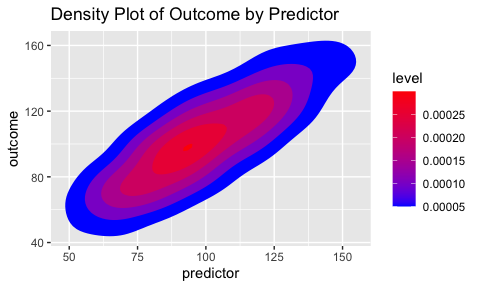
p4 +   
 geom\_density2d(color = "blue") + # add density geom   
 labs(title ="Density Plot of Outcome by Predictor") # add title



#### Filled Density

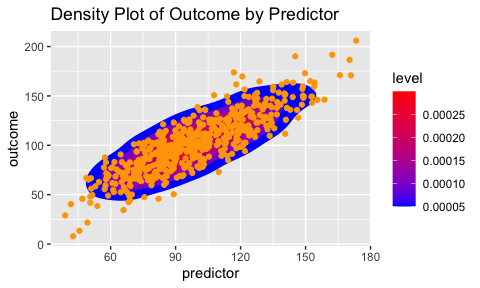
While not strictly necessary, the use of scale\_fill\_gradient seems to improve the presentation. You can choose your own colors.

p4 +   
 stat\_density\_2d(aes(fill = ..level..),   
 geom = "polygon") + # add filled density geom   
 scale\_fill\_gradient(low = "blue",  
 high = "red") +  
 labs(title ="Density Plot of Outcome by Predictor") # add title



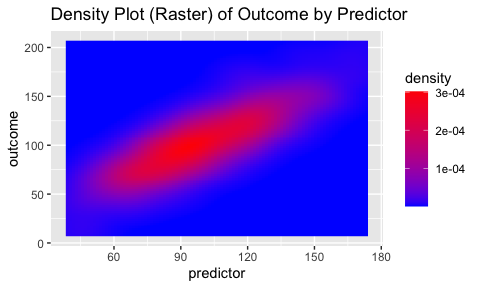
#### Add Points

p4 +   
 stat\_density\_2d(aes(fill = ..level..),   
 geom = "polygon") + # add filled density geom  
 geom\_point(color = "orange") +  
 scale\_fill\_gradient(low = "blue",  
 high = "red") +  
 labs(title ="Density Plot of Outcome by Predictor") # add title



#### Use a Raster Geom Instead

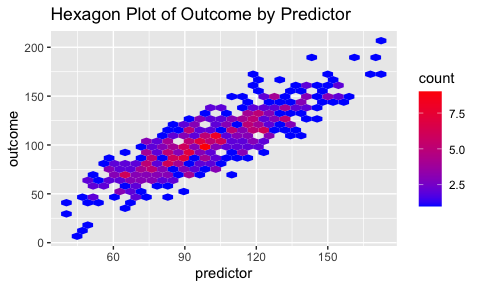
p4 +   
 stat\_density\_2d(geom = "raster",   
 aes(fill = ..density..),  
 contour = FALSE) +  
 scale\_fill\_gradient(low = "blue",  
 high = "red") +  
 labs(title ="Density Plot (Raster) of Outcome by Predictor") # add title



### Try a Hexagon Geom

geom\_hex may be a useful visualization, especially when there is the possiblity of *over-plotting* due to many many points.

p4 +   
 geom\_hex() +  
 scale\_fill\_gradient(low = "blue",  
 high = "red") +  
 labs(title ="Hexagon Plot of Outcome by Predictor") # add title

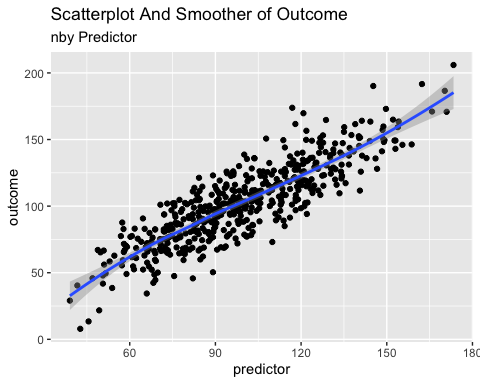


### Combine Points and Smoother And Add Some Themes

#### Themes Included With ggplot2

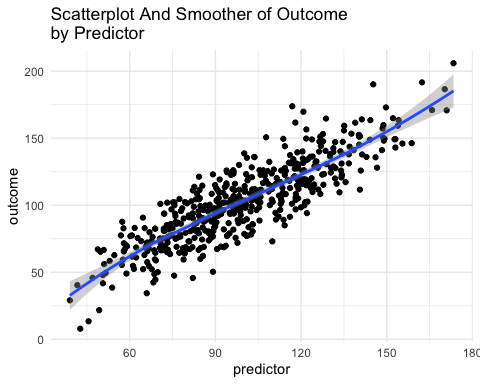
##### Default ggplot2 Theme

p4 +   
 geom\_point() + # point geom  
 geom\_smooth() + # add smooth geom  
 labs(title ="Scatterplot And Smoother of Outcome",   
 subtitle = "nby Predictor") + # add title  
 theme\_grey() # default theme



##### The “minimal” theme

p4 +   
 geom\_point() + # point geom  
 geom\_smooth() + # add smooth geom  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor") + # add title  
 theme\_minimal() # default theme

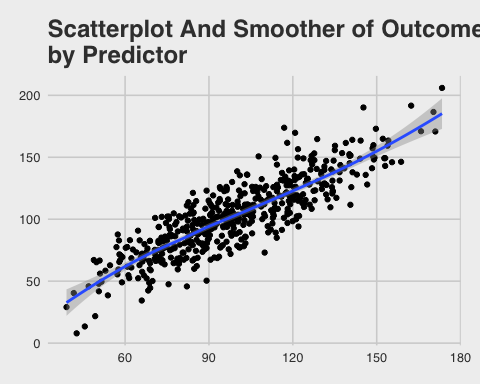


#### Themes requiring ggthemes()

The themes below make use of library(ggthemes) which you will need to install.

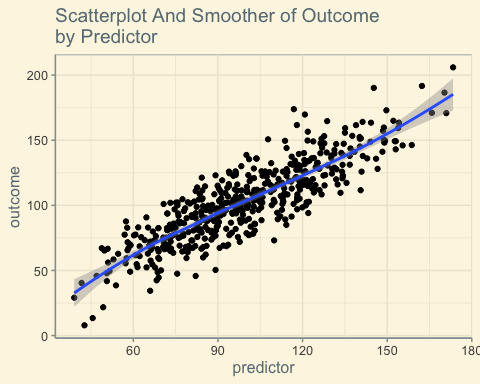
##### “538” Theme

p4 +   
 geom\_point() + # point geom  
 geom\_smooth() + # add smooth geom  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor") + # add title  
 theme\_fivethirtyeight() + # "538"-like theme  
 scale\_color\_fivethirtyeight() # "538"-like colors



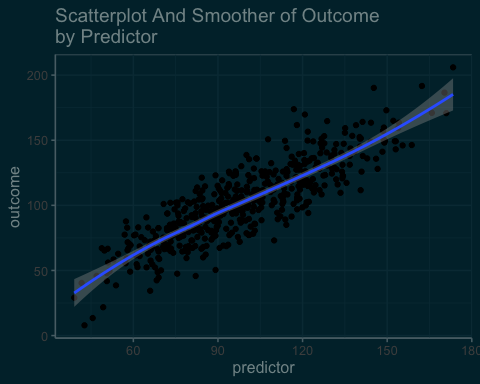
##### “Solarized Theme”

p4 +   
 geom\_point() + # point geom  
 geom\_smooth() + # add smooth geom  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor") + # add title  
 theme\_solarized() + # Google Docs theme  
 scale\_colour\_solarized() # Google Docs colors



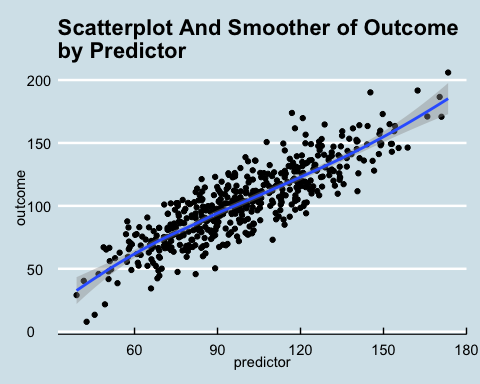
##### “Solarized Dark” Theme

p4 +   
 geom\_point() + # point geom  
 geom\_smooth() + # add smooth geom  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor") + # add title  
 theme\_solarized(light = FALSE) + # solarized dark theme  
 scale\_colour\_solarized("blue") # solarized dark color palette



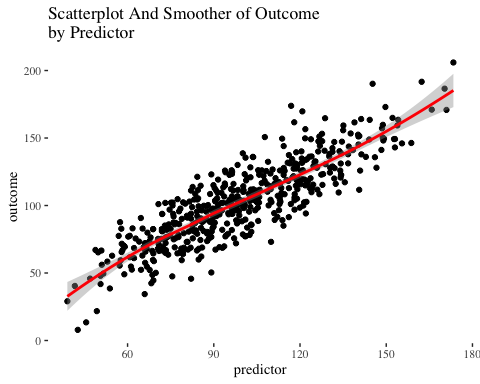
##### “Economist” Theme

p4 +   
 geom\_point() + # point geom  
 geom\_smooth() + # add smooth geom  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor") + # add title  
 theme\_economist() + # Economist magazine theme  
 scale\_colour\_economist() # Economist magazine colors



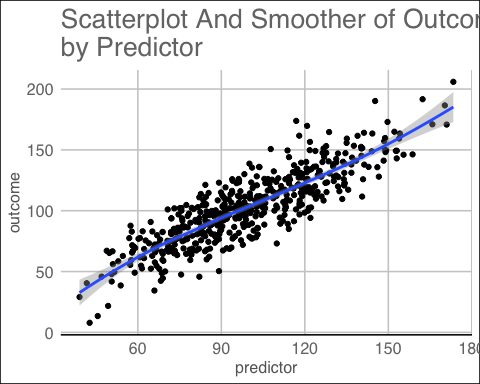
##### “Tufte” Theme

# same plot with theme and geom based on the work of Edward Tufte  
  
p4 +   
 geom\_point() +   
 geom\_smooth(color = "red") +   
 theme\_tufte() +  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor")



##### “Google Docs Theme”

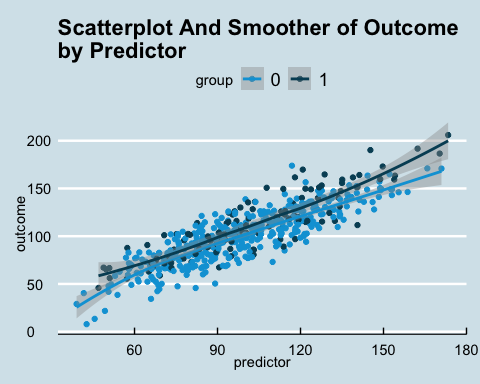
p4 +   
 geom\_point() + # point geom  
 geom\_smooth() + # add smooth geom  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor") + # add title  
 theme\_gdocs() + # Google Docs theme  
 scale\_colour\_gdocs() # Google Docs colors



## Two Continous Variables And A Third Categorical Variable

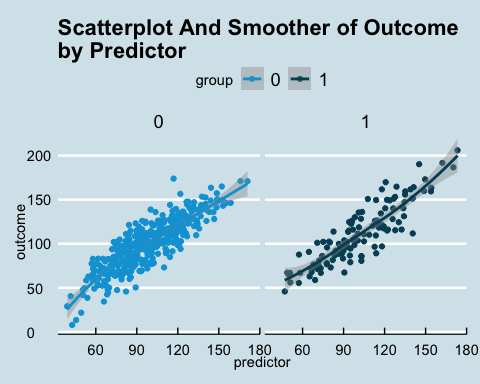
### Modify the aesthetic to include *group*.

p5 <- ggplot(mydata,   
 aes(x = predictor,   
 y = outcome,   
 color = group)) # aesthetic includes color by group  
  
p5 + geom\_point() +   
 geom\_smooth() +   
 theme\_economist() +  
 scale\_color\_economist() +  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor")



### Add facets or “small multiples” by group

p5 +   
 geom\_point() +   
 geom\_smooth() +   
 facet\_wrap(~group) + # facets or "small multiples" by group  
 theme\_economist() +  
 scale\_color\_economist() +  
 labs(title ="Scatterplot And Smoother of Outcome \nby Predictor")



# There Is A Lot More That Can Be Done With ggplot2

More information can be found at [ggplot2](https://ggplot2.tidyverse.org/).

More ggplot2 examples can be found [here](https://agrogan1.github.io/dataviz/how-to-choose-a-chart/how-to-choose-a-chart-v3.html).

Graphics made with the [ggplot2](https://ggplot2.tidyverse.org/) graphing library created by Hadley Wickham.

Available online at <https://www.umich.edu/~agrogan>

*Quick Introduction to ggplot2* by Andrew Grogan-Kaylor is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-sa/4.0/).

Last updated: March 17 2021 at 11:34

1. By way of illustration, this foundation could be just an aesthetic (e.g. aes(...)) alone, or possibly an aesthetic plus a theme (e.g. theme\_tufte()), plus axis labels to create a consistent look and feel for your graphs across a report. [↑](#footnote-ref-37)