AN INTRODUCTION TO VISUALIZATION WITH R

Joseph Nathan Cohen
Department of Sociology
City University of New York, Queens College
Spring 2020

CONTENTS

Introduction 3
Data for this Lesson 3
What is Data Visualization? 3
Principles of Good Visualization 4
Comprehensibility 4
Cleanliness 4
Honesty 5
Matching the Proper Graphics with Variable Types 6
The Grammar of ggplot2 7
Basic Graph Types 7
Univariate Discrete 8
Univariate Continuous 10
Histogram 10
Area Graph 11
Bivariate Figures 11
Bivariate Discrete-Discrete 11
SINGLE Colored Bar Plot 11
Multiple grouped bars 13
Bivariate Discrete-Continuous 14
Bar Charts 14
Box Plots 15
Bivariate Continuous-Continuous 16
Scatterplot 16
Smoothed Line graph 16

Extra Functions 17
Superimposing Figures 17
Titles 17
Themes 18
More Adjustments? 19

INTRODUCTION

In this lesson, we will learn the basics of data visualization, and learn how to develop visualizations using the ***ggplot2*** package. We will discuss:

(1) the value of data visualizations,
(2) principles of good graphing,
(3) generic graph types and how to implement them on ***ggplot2***,
(4) some useful operations for refining your visualizations

Data for this Lesson

For today's lesson, we will use an extract from the World Bank's *World Development Indicators*. This data set is a massive compilation of economic and social data from across the world. We will use data from 2015. You can download an extract of the data from Blackboard or Slack: "WDI 2015 Extract.xlsx"

Be careful with this data! There are a lot of missing values. Look at the raw data file before starting your analyses.

WHAT IS DATA VISUALIZATION?

Data visualization presents data through images. It allows analysts and their audiences engage data visually. Visualization is a great way to engage visual learners.

A well-executed visualization:

- Allows you to say more with fewer words
- Can help increase audience comprehension
- Can increase the impact of important statistical findings
- Allow audiences to actively contemplate your data or findings

PRINCIPLES OF GOOD VISUALIZATION

A well-constructed visualization follows three key principles:

Comprehensibility

A reader should be able to understand the graph. The graph should have labels, and readers should be able to link data points to labels. When it is difficult to pinpoint and make sense of the values on a visualization, then it is more poorly constructed.

Check out the Figure 1 to the right. This is an example of an incomprehensible graph. It is very hard to link data points to their values. Exactly how much profit per part did parts makers get for clamps in February? Good luck with that.

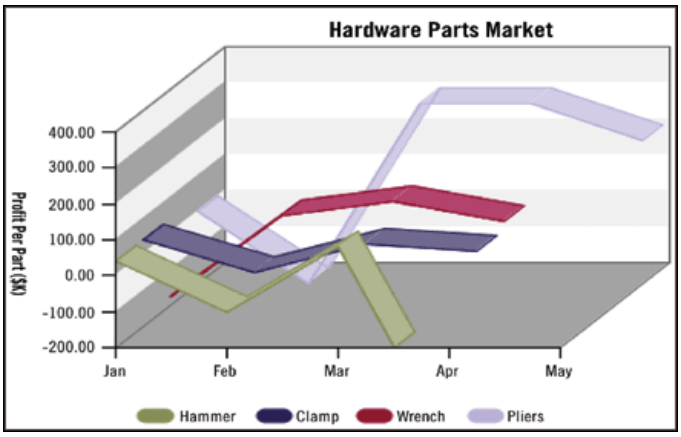
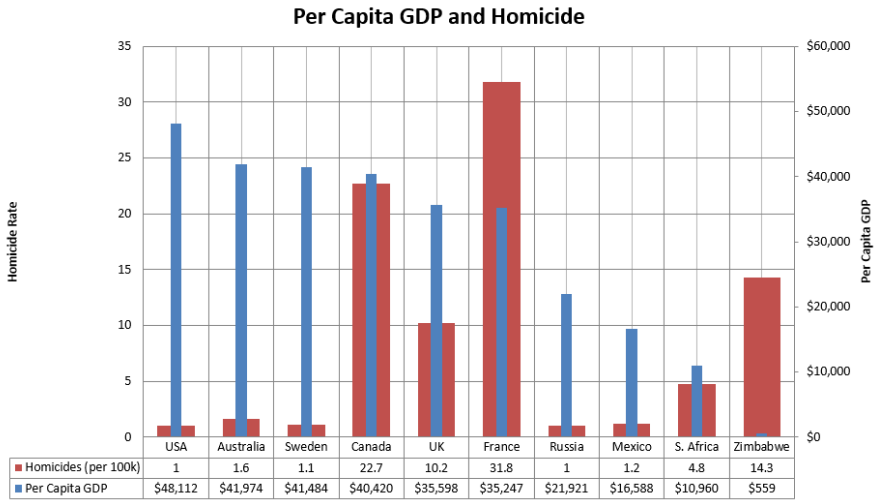


Figure 1: An Incomprehensible Graph

Cleanliness

Good visualizations should convey its information as simply as possible. It should avoid visual clutter or *chartjunk.* Remember that anyone who is reading your report is being asked to exert mental effort. If understanding you requires a lot of effort, people just won't read it. A clean graph makes understanding the information that you are trying to convey quick and easy.

Figure 2 (below) is an example of a cluttered graph. I contains a lot of data and requires that the reader exert a lot of mental energy to make sense of these numbers.



	USA	Australia	Sweden	Canada	UK	France	Russia	Mexico	S. Africa	Zimbabwe
Homicides (per 100k)	1	1.6	1.1	22.7	10.2	31.8	1	1.2	4.8	14.3
Per Capita GDP	$48,112	$41,974	$41,484	$40,420	$35,598	$35,247	$21,921	$16,588	$10,960	$559

Figure 2: A Cluttered Graph

I cleaned these figures below in Figure 3, accomplished up by dividing them into two, and sorting them. Do you find these figures easier to comprehend?

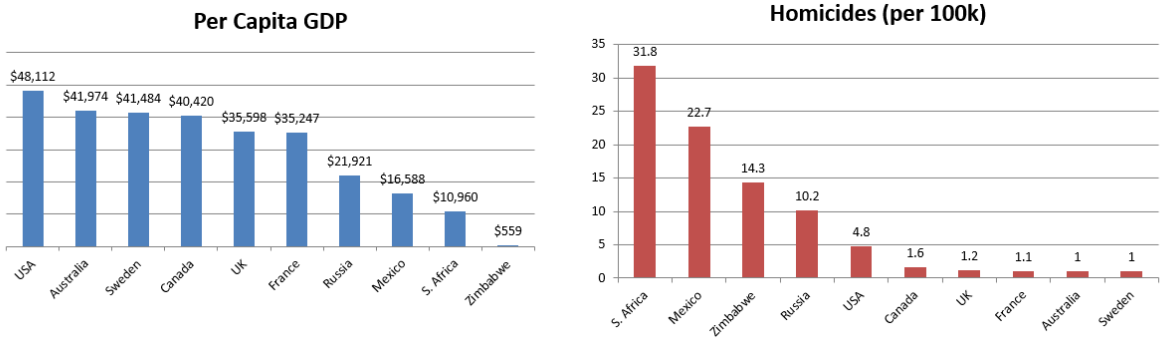


Figure 3: A Decluttered Version of the Above Figure

Honesty

By *honesty*, I mean an earnest effort to properly convey information without any intent to deceive. There are ways to create visualizations that are literally true in the sense that the data points depicted are accurate, but they are still manipulated to manage audience impressions.

Consider Figure 4 to the upper right, which depicts life expectancy by sex and race. When you read it, what is your impression about the contrast between the life expectancy of black males versus other groups?

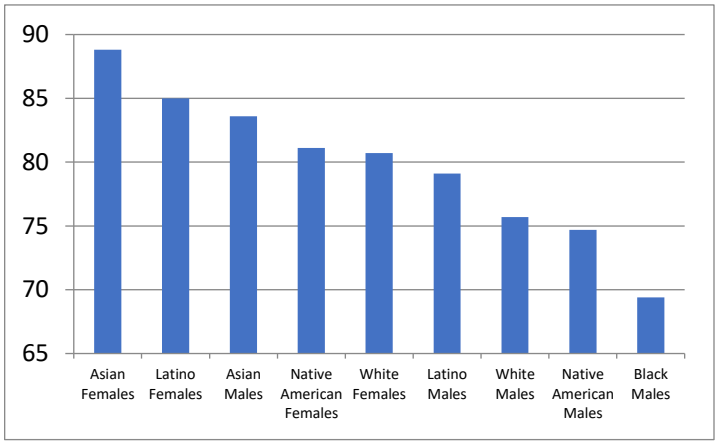


Figure 4: Life Expectancy by Sex and Race, Version 1

Now consider Figure 5 to the lower right, which depicts the same data but has a y-axis that begins at zero. Has your impression changed?

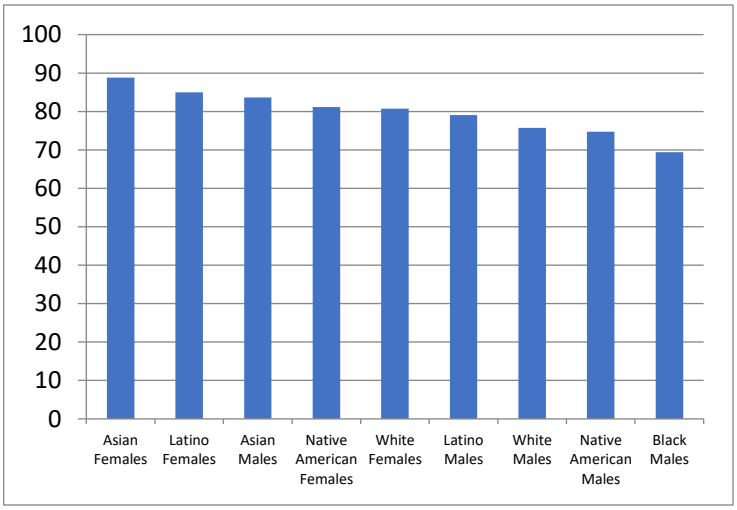


Figure 5: Life Expectancy by Race and Sex, Version 2

This is an old trick – the truncated axis. There's a bunch of them. In visualization, a lot of them involve playing with perspective and people's proclivity to make automatic judgments based on aesthetics. Here, we are influenced by the fact that we often judge differences by looking at graphical volume, and can develop impressions that are different from those communicated by the axis labels.

Interested in this topic? There's a fun little book on it by Darrell Huff called *How to Lie with Statistics.* It's an easy and fun read.

MATCHING THE PROPER GRAPHICS WITH VARIABLE TYPES

Remember from the last lesson that the first step is matching variable types with the appropriate graphs:

Variable Types	Recommended Graphs	ggplot2 Geom Type
Univariate Discrete	Histogram	geom_histogram()
Univariate Continuous	Histogram Area Graph	geom_histogram() geom_area()
Bivariate Discrete-Discrete	Bar Chart	geom_bar()
Bivariate Discrete-Continuous	Bar Chart Box Plot	geom_bar() geom_box()
Bivariate Continuous-Continuous	Scatterplot Line Graph	geom_point() geom_smooth()

The third column gives you information that will make sense once we explain how to execute visualizations using the ***ggplot2*** package. We turn to that topic next.

THE GRAMMAR OF GGPLOT2

To create an object in ***ggplot2***, your code should follow the following framework. Variables are presented in italics:

```
ggplot(data.object, aes(variables)) + geom.type + options
```

In which:

data.object is the object continaing your data table
variables are equations that identify the variables in *data.object* that will be used in creating your figure
geom.type describes the "geom" to be used in this figure. Use the table above or the *ggplot2 Cheat Sheet*
options are additional commands to manipulate various facets of the figure, like titles, colors, scales, and much more.

So, imagine that I want to show a scatterplot between the variables "weight" and "height" in a data table object called "new.data", the code would be:

```
ggplot(new.data, aes(x = height, y = weight)) + geom_scatter()
```

But sometimes manipulations need to be made. Let's do a walk through of the graph types outlined above.

BASIC GRAPH TYPES

We are working with our *World Development Indicators* data. Let's set up our session:

```
rm(list=ls())
gc()

setwd("E:/Dropbox/Teaching/DATA 712/Week 4")

library(readxl)
DAT <- read_xlsx("WDI 2015 Extract.xlsx", sheet = 1)
DAT <- data.frame(DAT)
head(DAT[1:6])
```

```
##            cname ccode electricity teen.birth hs.dropout hiv.young
## 1    Afghanistan   AFG    71.50000    73.1264         NA        NA
## 2        Albania   ALB   100.00000    20.6922    2.02836       100
## 3        Algeria   DZA    99.94307    10.7052         NA      1200
## 4 American Samoa   ASM          NA         NA         NA        NA
## 5        Andorra   AND   100.00000         NA         NA        NA
## 6         Angola   AGO    42.00000   157.3554         NA     26000
```

The data is cleaned. It's all continuous variables, but we'll manufacture discrete ones using the **cut()** command we learned last week. Let's get started!

Univariate Discrete

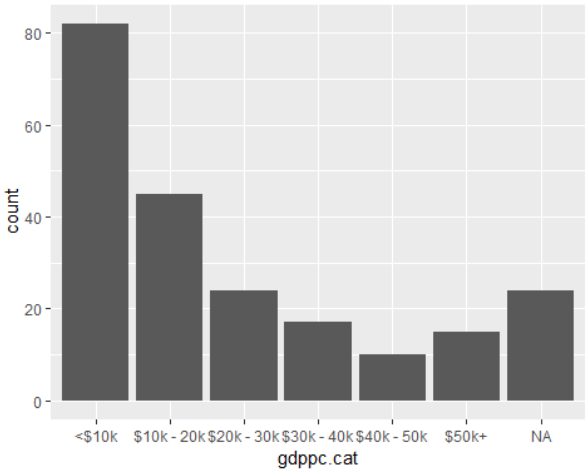
For a univariate discrete figure, try creating a histogram using the *geom_bar()* geom. To illustrate this operation, let's look at the relationship between a country's level of economic development and its maternal mortality rates.

Our data set doesn't have any discrete variables, so let's create one using the "gdp.pc" variable. This is a *per capita* GDP measure, which roughly captures how much an economy produces on a per person basis. We'll cut the variable into increments of ten thousand dollars. The resulting variable is a factor, so declare it as such:

```
#STEP 1: CREATE THE VARIABLE
DAT$gdppc.cat <- cut(DAT$gdp.pc, breaks = c(0, 10000, 20000, 30000, 40000, 50000,
9999999))
DAT$gdppc.cat <- factor(DAT$gdppc.cat, ordered = T,
                        labels = c("<$10k", "$10k - 20k", "$20k - 30k",
                                   "$30k - 40k", "$40k - 50k", "$50k+"))
```

And then we follow the above grammar to produce the figure:

```
#STEP 2: CREATE THE PLOT AND ATTACH IT TO THE VARIABLE
#Don't forget to load the "ggplot2" package
library(ggplot2)
FIG.1 <- ggplot(DAT, aes(x = gdppc.cat)) + geom_histogram(stat = 'count')
FIG.1
```



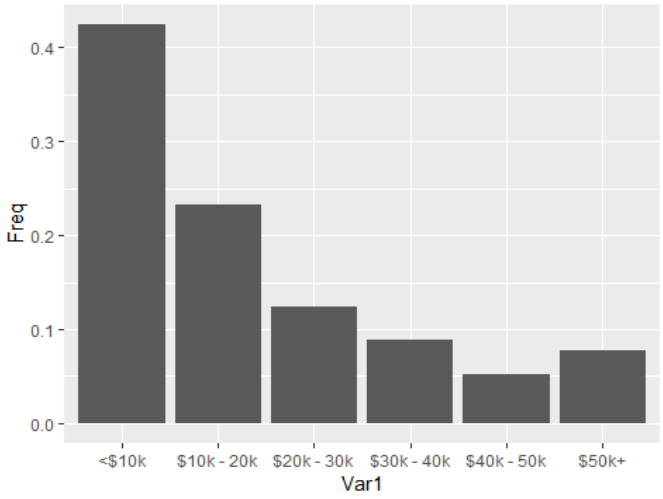
This is a basic plot. We'll discuss how to tweak it momentarily. Right now, I just want to focus on producing the figure. Note that the result will be a y-axis expressed as counts.

Want a Y-Axis Denominated in Percentages? What if you want to generate a histogram with a y-axis denominated in proportions (i.e., percent of sample)? My strategy is to produce a frequency table, and then use that as the data table from which the graph is generated to plot a bar chart:

```
GDP.FREQ <- prop.table(table(DAT$gdppc.cat))
GDP.FREQ <- data.frame(GDP.FREQ)
GDP.FREQ

##         Var1       Freq
## 1      <$10k 0.42487047
## 2 $10k - 20k 0.23316062
## 3 $20k - 30k 0.12435233
## 4 $30k - 40k 0.08808290
## 5 $40k - 50k 0.05181347
## 6      $50k+ 0.07772021

FIG.2 <- ggplot(GDP.FREQ, aes(x = Var1, y = Freq)) +
  geom_bar(stat = 'identity')
FIG.2
```



Again, these are crude graphs. We'll work on refining them momentarily. First, let's get through the basic graph types.

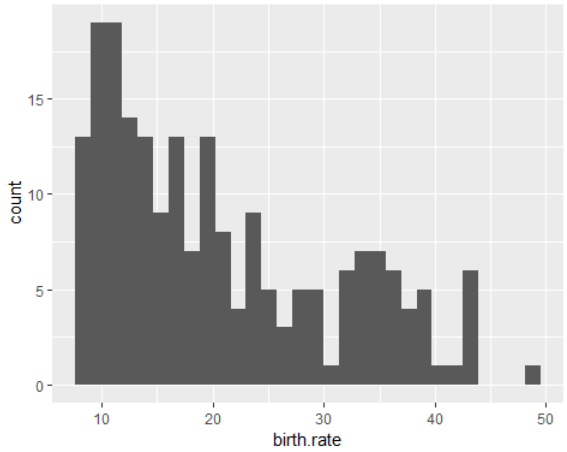
Univariate Continuous

To produce univariate continuous figures, I generally use histograms or area graphs. They have slightly different aesthetics, and you have to choose the one that best suits the narrative in which the graph is embedded.

HISTOGRAM

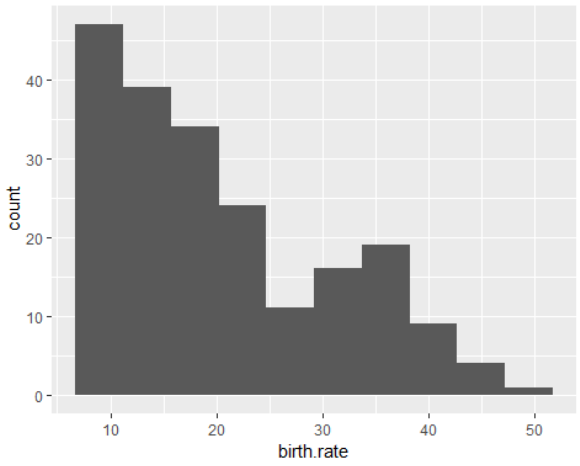
A histogram will produce a bar-like chart. Let's graph birth rates using the above syntax:

```
FIG.3 <- ggplot(DAT, aes(x = birth.rate)) + geom_histogram()
FIG.3
```



The default is to create 30 "bins" (or bars representing different parts of the different distribution. You can adjust these settings as option to the *geom_histogram()* subcommand:

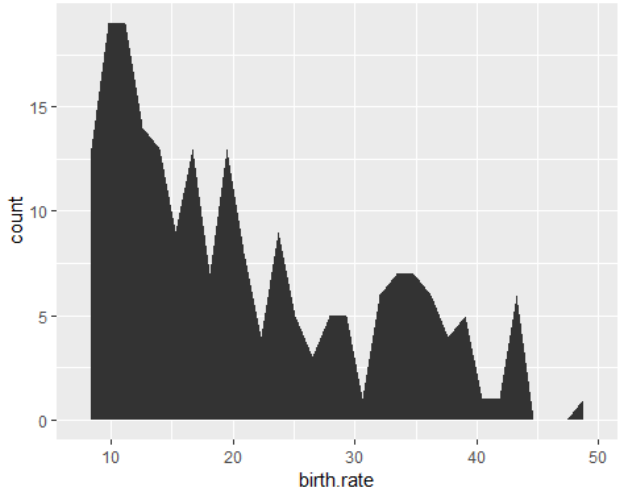
```
FIG.4 <- ggplot(DAT, aes(x = birth.rate)) + geom_histogram(bins = 10)
FIG.4
```



AREA GRAPH

An area graph gives a different representation of the data:

```
FIG.5 <- ggplot(DAT, aes(x = birth.rate)) + geom_area(stat='bin')
FIG.5
```



BIVARIATE FIGURES

Bivariate Discrete-Discrete

SINGLE COLORED BAR PLOT

One option is to create a colored bar plot, in which the proportions are denoted by color in single-group categories.

We will create two discrete variables using the **cut()** function on the variables "gdp.pc" and "mat.mortality". We will create three categories: low (below the 25th percentile), middle (between the 25th and 75th) or high (above the 75th). Our first step is to create an object with a frequency table.

```
#STEP 1: CUT FIRST VARIABLE (NOT NECESSARY IF VARIABLE ALREADY DISCRETE)
#(not necessary if variable is already discrete in your set)
#Find the cut points to construct our variable
quantile(DAT$gdp.pc, probs = c(0, 0.25, 0.75, 1), na.rm = T)

##          0%         25%         75%        100%
##    702.9853   4247.3817  27041.9805 115940.0272

#Cut the variable to create new discrete variable
DAT$gdppc.cat <- cut(DAT$gdp.pc, breaks = c(0, 4247, 27041, 9999999))

#Specify variable as ordinal factor
DAT$gdppc.cat <- factor(DAT$gdppc.cat, ordered=T, labels = c("Low", "Medium", "Hi
gh"))
```

```r
#STEP 2: CUT SECOND VARIABLE (NOT NECESSARY IF VARIABLE ALREADY DISCRETE)
#Do the same wiht the other variable
quantile(DAT$mat.mortality, probs = c(0, 0.25, 0.75, 1), na.rm = T)
```

```
##   0%  25%  75% 100%
##    3   14  229 1360
```

```r
DAT$matmort.cat <- cut(DAT$mat.mortality, breaks = c(0, 14, 229, 9999))
DAT$matmort.cat <- factor(DAT$matmort.cat, ordered = T, labels = c("Low", "Medium
", "High"))

#STEP 3: CREATE TABLE
#Here, we are going to describe maternal mortality as a percentage of economic de
velopment levels:
TAB.1 <- prop.table(table(DAT$gdppc.cat, DAT$matmort.cat), 1)
TAB.1
```

```
##
##                 Low     Medium       High
##   Low    0.00000000 0.23913043 0.76086957
##   Medium 0.12500000 0.79545455 0.07954545
##   High   0.85714286 0.11904762 0.02380952
```

```r
#STEP 4: TRANSFORM INTO DATA FRAME
#Recall that the first variable is GDP per capita, and the second is maternal mor
tality
TAB.1 <- data.frame(TAB.1)
head(TAB.1)
```

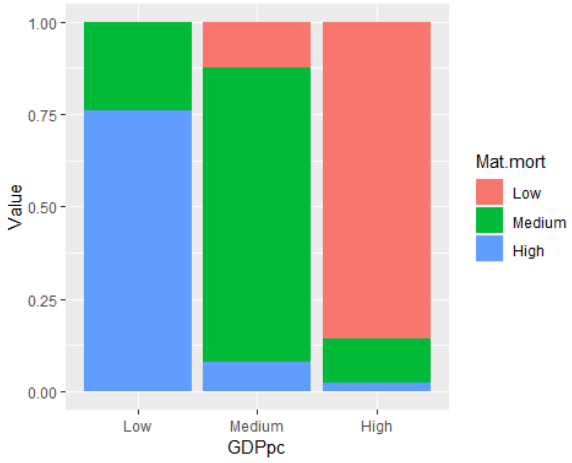
```
##     Var1   Var2      Freq
## 1    Low    Low 0.0000000
## 2 Medium    Low 0.1250000
## 3   High    Low 0.8571429
## 4    Low Medium 0.2391304
## 5 Medium Medium 0.7954545
## 6   High Medium 0.1190476
```

```r
#STEP 5: RENAME VARIBLES
#Let's rename the variables to make it easier to read when the graph is produced
names(TAB.1) <- paste(c("GDPpc", "Mat.mort", "Value"))
head(TAB.1)
```

```
##    GDPpc Mat.mort     Value
## 1    Low      Low 0.0000000
## 2 Medium      Low 0.1250000
## 3   High      Low 0.8571429
## 4    Low   Medium 0.2391304
## 5 Medium   Medium 0.7954545
## 6   High   Medium 0.1190476
```

Then we generate the figure using the table we just generated. We will attach the plot to an object, and then evoke that object to see the results. Following the basic grammar above (don't forget to load the ***ggplot2*** package). Note that there is a sub-command -- stat='identity' – option evoked in the geom term:

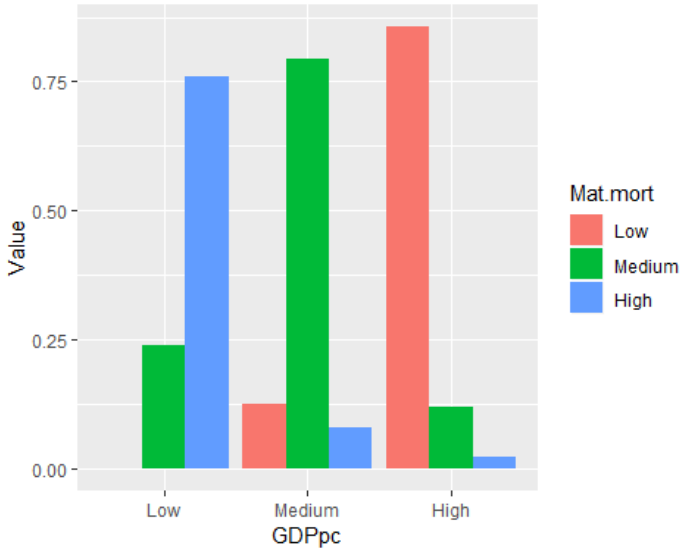
```
FIG.6 <- ggplot(TAB.1, aes(x = GDPpc, y = Value, fill = Mat.mort))
         + geom_bar(stat='identity')
FIG.6
```



MULTIPLE GROUPED BARS

An alternative is grouped bars. Here you are adding terms to the ***aes()*** and ***geom_bar()*** terms.

```
FIG.7 <- ggplot(TAB.1, aes(x = GDPpc, y = Value, fill = Mat.mort)) + geom_bar(sta
t='identity', position=position_dodge())
FIG.7
```



Bivariate Discrete-Continuous

We will consider two figures for these types of variable combinations. One is simpler and the other is more descriptive but harder for people to understand. Let's start with the simpler one.

BAR CHARTS

The trick here is to generate a table of summary statistics, and then graph the table you generate. The process is similar to the one we used for bivariate discrete comparisons. Let's use our discrete GDP variable, and compare the prevalence of smoking in these groups.

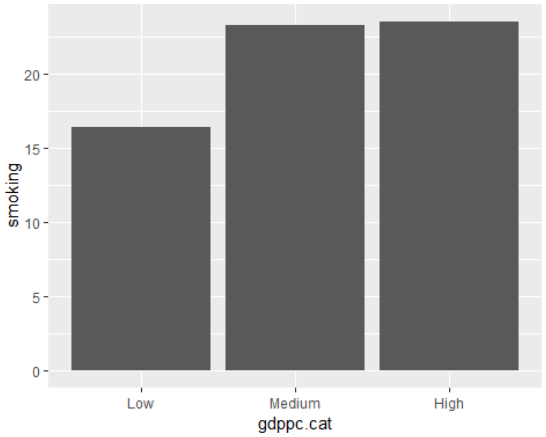
My strategy is to calculate group averages using the **aggregate()** command from last week:

```
TAB.2 <- aggregate(smoking ~ gdppc.cat, DAT, mean)
TAB.2

##   gdppc.cat  smoking
## 1       Low 16.40667
## 2    Medium 23.30274
## 3      High 23.56410
```

Then use that object as the data table to produce the figure:

```
FIG.8 <- ggplot(TAB.2, aes(x = gdppc.cat, y = smoking)) + geom_bar(stat = 'identi
ty')
FIG.8
```



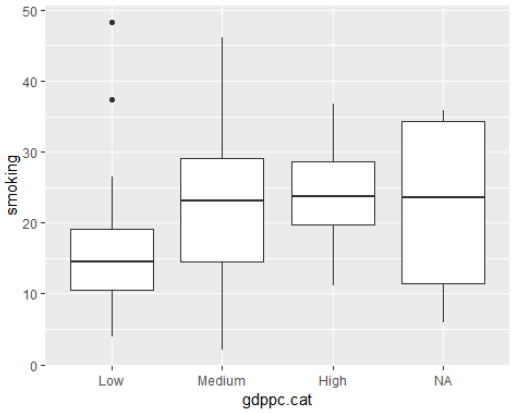
You can use this method with just about any descriptive statistic you can produce using **aggregate()** or any other method.

BOX PLOTS

Box plots have the advantage of describing a distribution in more detail, but people find them harder to read.

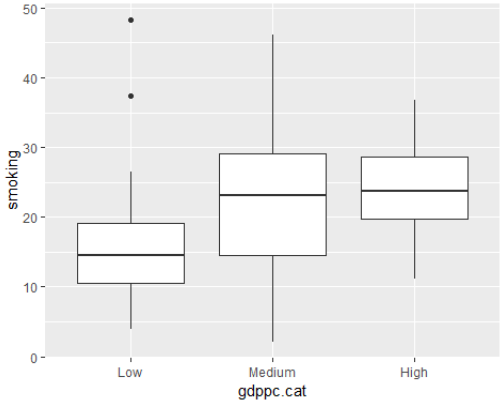
The implementation here is more straightforward:

```
ggplot(DAT, aes(gdppc.cat, y = smoking)) + geom_boxplot()
```



However, we might not want that "NA" category. We can get rid of it by subsetting the data set.

```
FIG.8 <- ggplot(subset(DAT, !is.na(gdppc.cat)), aes(x = gdppc.cat, y = smoking))
+ geom_boxplot()
FIG.8
```

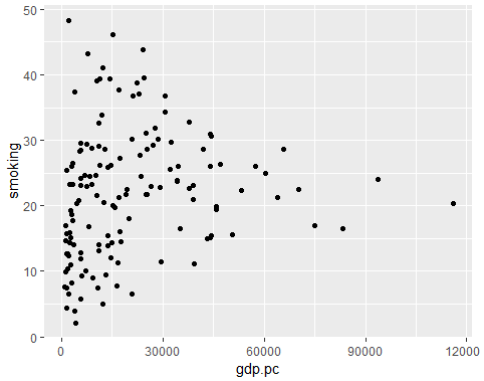


Bivariate Continuous-Continuous

Here, the recommended graphic is a scatterplot or a non-linear line plot. Implementation is straightforward for both. Below, we compare *per capita* GDP and smoking rates.

SCATTERPLOT

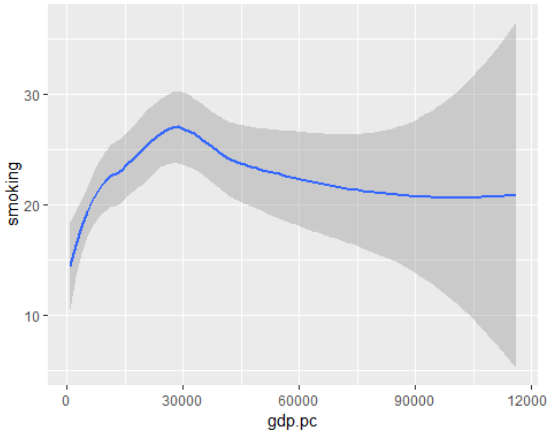
```
FIG.9 <- ggplot(DAT, aes(x = gdp.pc, y = smoking)) + geom_point()
FIG.9
```



SMOOTHED LINE GRAPH

Be careful wth this one. It's appealing, but it can also mislead

```
FIG.10 <- ggplot(DAT, aes(x = gdp.pc, y = smoking)) + geom_smooth()
FIG.10
```

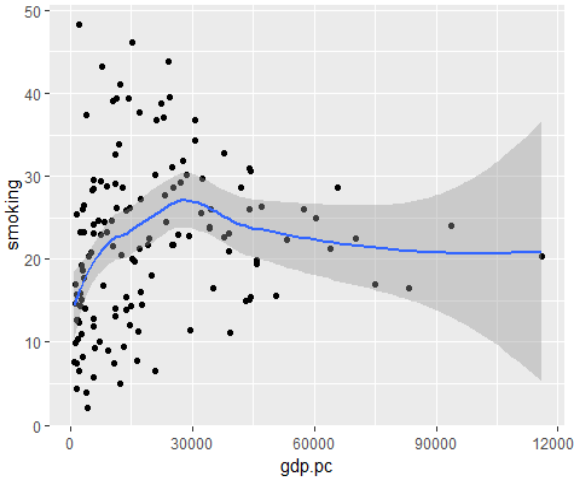


EXTRA FUNCTIONS

Superimposing Figures

You can superimpose figures by adding geoms. For example, let's say we wanted to blend Figures 9 and 10 above. You can just add a *geom_smooth()* to FIG.9

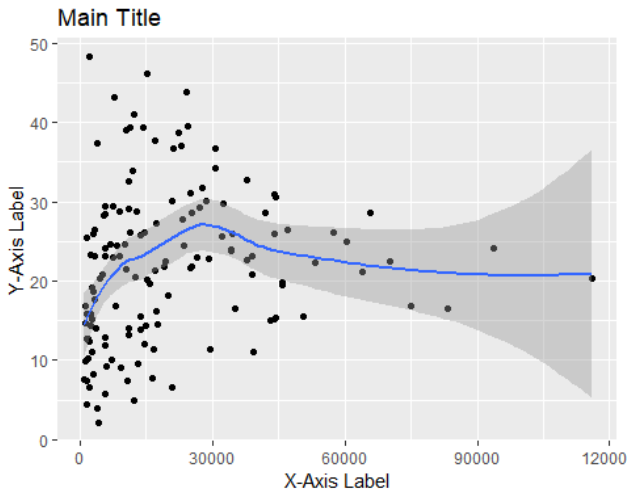
```
FIG.9 + geom_smooth()
```



Titles

You can add titles to graphs or axes using the ***ggtitle(), xlab()*** and ***ylab()*** options:

```
FIG.11 <- FIG.9 + geom_smooth() + ggtitle("Main Title") +
  xlab("X-Axis Label") + ylab("Y-Axis Label")
FIG.11
```

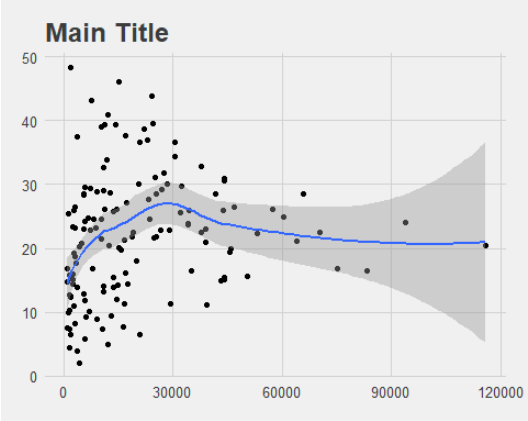


Themes

You can completely change the figure aesthetics using themes. You can get a bunch of them by using the ***ggthemes*** package. Check it out. Make your graph look like they do it on *FiveThirtyEight:*

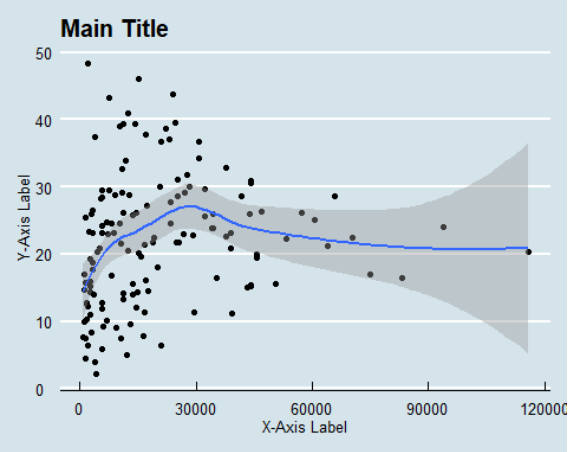
```
library(ggthemes)

FIG.11 + theme_fivethirtyeight()
```



Or like *The Economist.* There's many more. I just want you to know about the concept of themes.

```
FIG.11 + theme_economist()
```

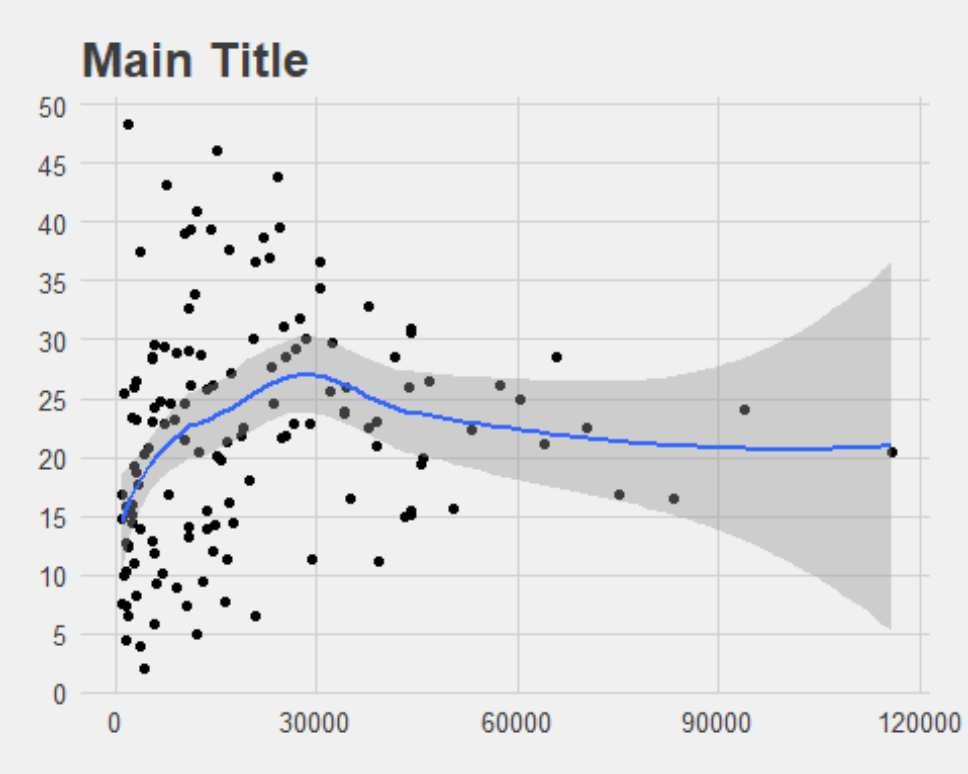


Custom Tick Marks on Continuous Variables' Axes

Setting the axis settings for discrete variables happens according to the settings you used when specifying a variable as a factor. For more on how to arrange discrete axes, check out this page: https://stackoverflow.com/questions/3253641/change-the-order-of-a-discrete-x-scale

For a continuous scale, many adjustments are possible Check out Stack Overflow, which can give you instructions on whatever you need. However, to change tick marks, you manipulate the **scale_x_continuous()** or **scale_y_continuous()** option, and set the "breaks=" option. You will notice that I use the **seq()** command, which delivers a number sequence:

```
FIG.11 + theme_fivethirtyeight() +
  scale_y_continuous(breaks = seq(0,50,5))
```



More Adjustments?

There are tons of Internet resources. Just enter what you want to do, and be sure to add "ggplot" to your query.