Lab 6. Advanced Graphics and Data Manipulation

# Introduction

Today is your first day working with an **R Package** - aka, the expansion packs that make R so fun\* and versatile. They are also sometimes called **libraries**. You will be working with two packages you have seen me working with in lecture - ggplot2, a graphics package, and dplyr which helps to manipulate data. They can be tricky to learn at first, but make your life a lot simpler in the long run.

This lab does not have any statistics outcomes. The goal is to help you learn some new graphics types so that in the next few weeks as our data gets more and more complicated, you’ll have the coding skills to look at it and figure it out. This lab does come with a script, so that you can modify that and not deal with any copy-paste errors.

(\*well, for a given definition of fun anyway)

# Learning Outcomes

By the end of today’s class you should be able to do the following in R:

* Install packages using the install.packages() function
* Load packages using the library() function
* Use geom\_histogram() to make a basic histogram
* Use geom\_bar() to make a basic bar plot, staggered and stacked
* Use geom\_density() to make a basic density plot
* Use the col, fill, alpha arguments to change your aesthetics
* Determine if your arguments should go in the aes() code or not
* Use facet\_wrap() to facet plots
* Visually evaluate multivariate plots to make recommendations
* Choose an appropriate plot to look at data

# Part 1: Importing Data and Packages

## Part 1.1 Data

Today you will be using a dataset on video game sales for games that have more than 100,000 copies sold. The data is from here: <https://www.kaggle.com/gregorut/videogamesales> though I have also posted it on D2L. This data set contains information on the games, and some sales information (in millions) in North America, Europe, Japan, Other Countries, and the total Global Sales.

It’s a lot of data - too much. So for today’s scenario, you have been employed by a company named “Idea Factory.” They’ve made a lot of games and done fairly well for themselves, but want to expand their North American sales. They’ve asked you to work for them and help them come up with some strategies for how they can be a little bit more like Nintendo, which has done very very well in North America.

You’ll need to use the subset() function to create three different datasets - one called *IFa* for Idea Factory, one called *Nin* for Nintendo, and one called *VG* for both companies.

## Part 1.2 Packages

Most packages are pretty easy to install. You use the install.packages() function, and make sure to spell the package name correctly AND put it in quotation marks. If you don’t, it’ll give you a warning - something like it not being found, or needing RTools to install it.

install.packages("dplyr")  
install.packages("ggplot2")

You only need to install packages once on your computer, but you need to load them every time you open R. This helps keep R a little faster - I have hundreds of packages, and R would take forever to load if all those packages opened every time I opened up R. So I select the ones I want each time, and use the library() function to remind R where they are.

library("dplyr")  
library("ggplot2")

You see the red warning from dplyr? Should look something like this:

##   
## 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

That just means that there are other functions in R that have that same name - and right now, R will assume you want the dplyr versions instead. Don’t worry about this message.

# Part 2: Graphics in ggplot2

## Part 2.1 Introduction

**NOTE: There is a good section of powerpoint on ggplot’s weird grammar. If you are working on your own, please stop and peruse the powerpoint before going any further.**

ggplot2 is a graphical package for R. You can consider it a sort of ‘regional accent’: sort of like how folks from the south say y’all, there are some differences in the way that ggplot2 code is written. It can be a little confusing, but ggplot is a lot more powerful than the hist() function you learned before and once your code gets more complicated it becomes MUCH easier to use ggplot2.

So to start with, here is a very basic ‘sentence’ in ggplot2 (**note** that the function is ggplot(), but the package is ggplot2):

ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram()

## Part 2.2 Customize Histograms

Not very exciting, but it gets better! Here are a few example lines of code to customize your plots. Play around with them, and then answer the question below to show you understand what each chunk of the language does.

ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram(col="blue")   
ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram(fill="green")   
ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram(alpha=.5)  
ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram(binwidth=.02)

### QUESTION 1: To make a histogram that has green bars with blue outline, and all of it is translucent, which of the following do you use:

A. ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram(fill="green", col = "blue", alpha = .5)   
B. ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram(fill="green", col = "blue", alpha = 1)   
C. ggplot(IFa, aes(x=NA\_Sales)) + geom\_histogram(col="green", border = "blue", alpha = .5)   
D. ggplot(IFa, aes(x=NA\_Sales, fill="green", col = "blue")) + geom\_histogram(alpha = .5)

### QUESTION 2: Try these lines of code. What happens if you put the ‘fill’ argument inside the aes() function?

ggplot(IFa, aes(x = NA\_Sales)) + geom\_histogram(fill="blue")  
ggplot(IFa, aes(x = NA\_Sales)) + geom\_histogram(aes(fill="blue"))  
ggplot(IFa, aes(x = NA\_Sales), fill = "blue") + geom\_histogram()  
ggplot(IFa, aes(x = NA\_Sales, fill = "blue")) + geom\_histogram()

1. It changes the border of the histogram
2. It colors it as a variable called “blue” but doesn’t make it blue
3. It colors it as a variable called “blue” and makes it blue
4. Nothing!

## Part 2.3 Different Shapes

Histograms are all well and good but honestly, you know how to make them already. What about a few more exciting options, ones we haven’t done before? In ggplot2, these are known as **geometries**, because they change the plot shape. They all start with geom\_ and then have a different name to describe what they do. Two easy ones are *density diagrams* and *dotplots*:

ggplot(IFa, aes(x=NA\_Sales)) + geom\_density()

ggplot(IFa, aes(x=NA\_Sales)) + geom\_dotplot()

But that’s not all! Other geometries like boxplots, violin plots and scatterplots and so many others are just as easy to do! The only difference is that these plots all require two variables - in some cases that variable is continuous, and in some it is categorical.

ggplot(IFa, aes(x=NA\_Sales, y = Year)) + geom\_point()

ggplot(IFa, aes(x=NA\_Sales, y = Genre)) + geom\_boxplot()

ggplot(IFa, aes(x=NA\_Sales, y = Genre)) + geom\_violin()

### QUESTION 3: Which of the above geometries requires two continuous numeric variables?

1. geom\_point()
2. geom\_boxplot()
3. geom\_violin()

## Part 2.4: Faceting, adding colors

While these graphics are interesting, they don’t tell us very much about our data. We want to know if Idea Factory needs to change the type of game they’re making. There are two ways of making a plot that does that - first, you can use color by group, and secondly you can split out your data by some sort of categorical data into many different panels, or **facets**.

To start with color, put your variable you want to color and split things out by in your aes code:

ggplot(IFa, aes(x=NA\_Sales, fill = Genre)) + geom\_density()

To make things easier to see, add the alpha argument to make things see through:

ggplot(IFa, aes(x=NA\_Sales, fill = Genre)) + geom\_density(alpha = .5)

Still kind of rough, so let’s add in the second option - splitting things out quickly by your category. This code is getting a little unwieldy, so I’m going to start putting things on different lines. Just make sure to have the + at the end of each line!

ggplot(IFa, aes(x=NA\_Sales, fill = Genre)) +   
 geom\_density(alpha = .5) +   
 facet\_wrap(~Genre)

### QUESTION 4: According to the above graphic, which of the following is true about Idea Factory:

1. They make way more Role-Playing and Adventure Games than other types of games
2. Their Adventure games on average make a lot of money
3. Their Action games typically make a lot of money
4. Most of these games make more than 2 million each

## Part 2.5 Cross-Compare

You can also make plots that only take a single categorical variable, like a barplot:

ggplot(IFa, aes(x=Genre)) + geom\_bar()

### QUESTION 5: What does the y axis represent in this geom\_bar() geometry?

1. Maximums of North American Sales
2. Averages of North American Sales
3. Number of North American Sales
4. Number of entries for that genre

We want to know how Nintendo is different, so we’re going to use the fill argument to color by publisher:

ggplot(data = VG, aes(x = Genre, fill = Publisher))+  
 geom\_bar()

This creates what is called a **stacked** bar chart, where the y axis no longer corresponds perfectly to each category (Idea Factory did not make 100 some Action games, they made like… 30). Use position to change that:

ggplot(data = VG, aes(x = Genre, fill = Publisher))+  
 geom\_bar(position = position\_dodge(preserve = "single"))

HINT: Can’t read the x labels? Add this code: + theme(axis.text.x = element\_text(angle=45, hjust=1))

### QUESTION 6: What genre of game is Idea Factory producing way more of than Nintendo?

1. Misc
2. Puzzle
3. Adventure
4. Action

Let’s take a look at the Nintendo data now - specifically, what are their best-selling games? You’ll notice that their data is very skewed, which makes it hard to see - so I’ve added scale\_x\_continuous to take a closer look:

ggplot(Nin, aes(x=NA\_Sales, fill = Genre)) +   
 geom\_density(alpha = .5)+  
 scale\_x\_continuous(limits = c(1, 55))

### QUESTION 7: scale\_x\_continuous() is a function most similar to which arguement in the hist() function?

1. ylim
2. xlim
3. xlab
4. xintercept

## Part 2.6 Colors and Fills

Don’t like the colors that R automatically gives you? The scale\_color\_manual() and scale\_fill\_manual() functions will be your new bestest friends. They help you give R a list of values to color by, in order. IE:

ggplot(IFa, aes(x=NA\_Sales, fill = Genre, col = Platform)) +   
 geom\_density(alpha = .5)+  
 scale\_fill\_manual(values = c("red", "blue", "goldenrod", "black", "purple"))+  
 scale\_color\_manual(values = c("mediumseagreen", "violet", "yellowgreen", "darkorchid1", "pink", "yellow", "dodgerblue"))

# Part 3 - Data Manipulation with dplyr

The dplyr package is intended to help sort and arrange data by groups. It means ‘d’ as in data, and ‘plyr’ as in a set of pliers. Instead of a +, it uses a %>% to tell R to keep looking. It is a very easy way to do summary statistics by group! Here is some example code:

Table1 <- VG %>%  
 group\_by(Genre) %>%  
 summarize(Mean = mean(NA\_Sales))

The group\_by() function tells it what your categorical grouping is, the summarize() function (which you may also spell as summarise()) tells it that you want to do something according to those groups - in this case, take the mean of the North American sales column. But that’s not all! You can add extra groups, like Publisher:

Table2 <- VG %>%  
 group\_by(Genre, Publisher) %>%  
 summarize(Mean = mean(NA\_Sales))

You can also do more than the mean - you just have to name each new summary, and separate them with a comma:

Table3 <- VG %>%  
 group\_by(Genre, Publisher) %>%  
 summarize(Mean = mean(NA\_Sales), Maximum = max(NA\_Sales))

### QUESTION 8: Which of the following codes gives you the median sales by publisher for each genre?

A. VG %>% group\_by(Genre, Publisher) %>% summarize(Mean = median(NA\_Sales))  
B. VG %>% group\_by(Genre) %>% summarize(Median = median(NA\_Sales, Publisher))  
C. VG %>% summarize(Median = median(NA\_Sales, Publisher, Genre))  
D. VG %>% group\_by(Genre, Publisher) %>% summarize(Median = mean(NA\_Sales))

You can also use other functions, like the shapiro.test() function, by groups (but only if there are enough samples per group!)

Table4 <- VG %>%  
 group\_by(Publisher) %>%  
 summarize(p = shapiro.test(NA\_Sales)$p.value)  
  
Table5 <- VG %>%  
 group\_by(Genre) %>%  
 summarize(p = shapiro.test(NA\_Sales)$p.value)

### QUESTION 9: According to the shapiro test results above, which is true:

1. The central tendency of all of your data is best described by a mean
2. The central tendency of all of your data is normally distributed
3. The central tendency of all of your data is best described by a median

# Part 4 Recommendations and Interpretations

Remember that the lab for today’s question was: Idea Factory wants to expand their North American sales. They’ve asked you to work for them and help them come up with some strategies for how they can be more successful in that area, like Nintendo. For the rest of this lab, you will use the things you learned above to present your argument to them - **what should they do to expand their North American sales like Nintendo, according to this data**?

### QUESTION 10: First, write what it is you think they should do. IE, abandon one genre for another, move to another platform like the Wii, etc. You do not need to use a t.test or anything, just make a recomendation based on what you’ve seen (and feel free to look into the data more).

### QUESTION 11: Second, create a graphic that illustrates your point using ggplot2. Copy-paste your code on D2L.

### QUESTION 12: Third, create a table that illustrates your point using dplyr. Copy-paste your code on D2L. (you do not need to modify your table, just provide the dplyr grouped code)

### EXTRA CREDIT: Look at the plot below. It uses only code from this lab and those previously. Try to replicate it as best you can - close codes get an extra credit point.

