ggplot - by no means an exhaustive guide

Nicole

4/9/2019

## Set-up

Load library - you can either load ggplot specifically or load ‘tidyverse’

library(ggplot2)  
library(tidyverse)

## ── Attaching packages ────────────

## ✔ tibble 2.0.1 ✔ purrr 0.3.0   
## ✔ tidyr 0.8.0 ✔ dplyr 0.8.0.1  
## ✔ readr 1.1.1 ✔ stringr 1.3.1   
## ✔ tibble 2.0.1 ✔ forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## Warning: package 'dplyr' was built under R version 3.5.2

## ── Conflicts ─────────────────────  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

Import data: I have included dummy data on the github repo for an example.

dummydata <- read.csv("ggplot dummy data.csv")  
# View data   
str(dummydata)

## 'data.frame': 80 obs. of 4 variables:  
## $ Subject : Factor w/ 40 levels "sub1","sub10",..: 1 12 23 34 36 37 38 39 40 2 ...  
## $ PT.Score : num 0.25 0.25 0.25 0.25 0.25 0.39 0.39 0.39 0.39 0.39 ...  
## $ TrialType : Factor w/ 2 levels "CanSee","DoesNotSee": 1 1 1 1 1 1 1 1 1 1 ...  
## $ NetworkSize: int 1 2 3 4 5 6 7 8 9 10 ...

# How many subjects?  
length(unique(dummydata$Subject))

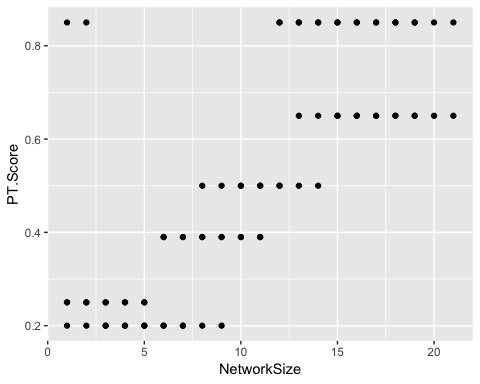
## [1] 40

## Using ggplot!

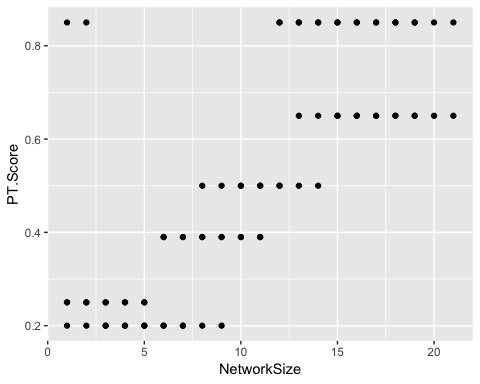
Here is the code for basic ggplot. ggplot follows the ‘grammar of graphics’. Google that phrase if you want to know more about it - it takes about how ggplots are set up.

You start with a base:

plot <- ggplot(data = dummydata, aes(x = NetworkSize, y = PT.Score)) +  
 geom\_point()  
plot



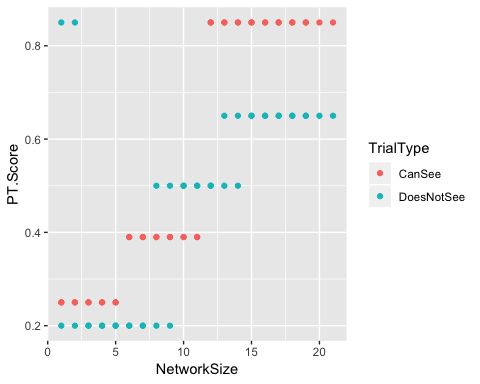
# You could also write it like this and it returns the same thing  
plot <- ggplot(dummydata, aes(NetworkSize, PT.Score)) +  
 geom\_point()  
plot



### Coloring points by a third variable

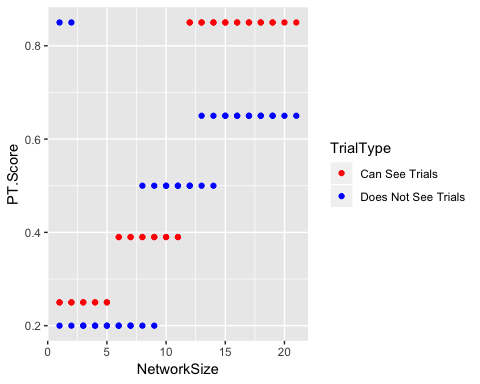
Then you can color those points by a third variable. For example, what if we want to color those points by TrialType?

plot\_color <- ggplot(data = dummydata, aes(x = NetworkSize, y = PT.Score,  
 colour = TrialType)) +  
 geom\_point()  
plot\_color



How to change those **colors**. The colors it defaults too suck. You can change the colors by adding this line of code. You can also change what you call the 3rd variable by using “labels”.

plot\_color2 <- ggplot(data = dummydata, aes(x = NetworkSize, y = PT.Score,  
 colour = TrialType)) +  
 geom\_point() + scale\_colour\_manual(values = c("red", "blue"), labels = c("Can See Trials", "Does Not See Trials"))  
plot\_color2

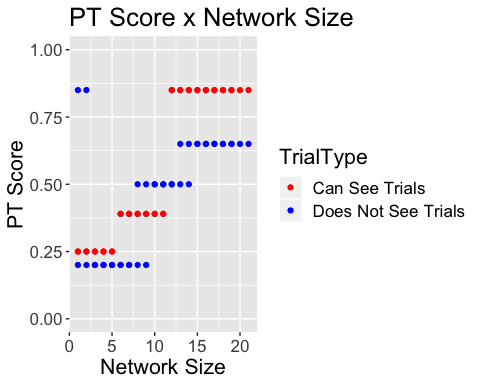


**If/when scale\_colour\_manual doesn’t work**. Try scale\_colour\_discrete or scale\_colour\_fill. scale\_colour\_manual works for categorical variables. Those other functions works for continious variables.

### Change axis, axis titles, and size

How to change the axis titles and sizes! I will do that all at once here, but the code is pretty self-explanatory:

plot\_titlesandsize <- ggplot(dummydata, aes(NetworkSize, PT.Score, colour = TrialType)) +  
 geom\_point() + ggtitle("PT Score x Network Size") + ylab("PT Score") + xlab("Network Size") +  
 theme(text = element\_text(size = 16)) + ylim(0,1) +   
 scale\_colour\_manual(values = c("red", "blue"), labels = c("Can See Trials", "Does Not See Trials"))  
plot\_titlesandsize



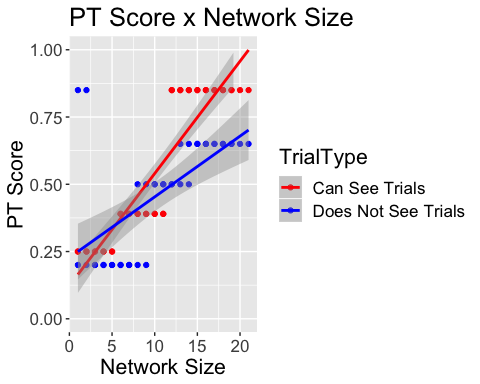
You can change the x-axis by adding ‘xlim()’

### Add regression lines or other lines

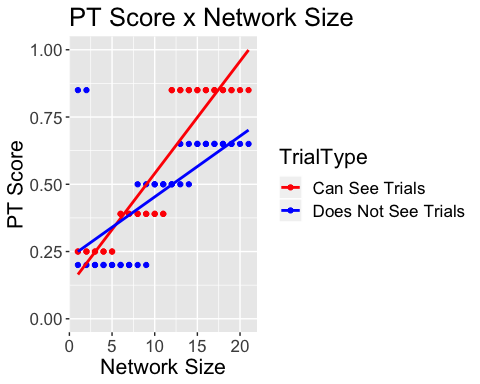
To add a regression line, you should use geom\_smooth. Depending on what line you want will change where you put the code: ‘colour = TrialType’.

If you want separate regression lines for each of your values (i.e., separate lines for CanSee and CannotSee), then you do the following:

plot\_lines <- ggplot(dummydata, aes(NetworkSize, PT.Score, colour = TrialType)) +  
 geom\_point() + ggtitle("PT Score x Network Size") + ylab("PT Score") + xlab("Network Size") +  
 theme(text = element\_text(size = 16)) + ylim(0,1) +   
 scale\_colour\_manual(values = c("red", "blue"), labels = c("Can See Trials", "Does Not See Trials")) +  
 geom\_smooth(method = "lm")  
plot\_lines

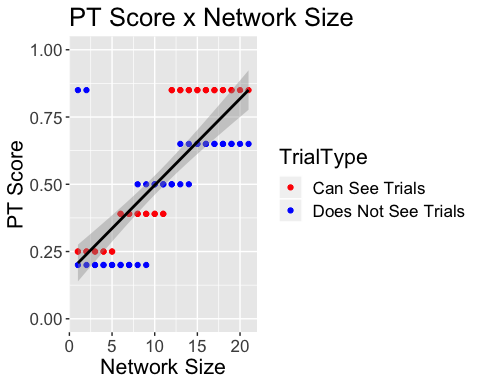


# indicating se = FALSE in geom\_smooth will get rid of se bars  
plot\_lines <- ggplot(dummydata, aes(NetworkSize, PT.Score, colour = TrialType)) +  
 geom\_point() + ggtitle("PT Score x Network Size") + ylab("PT Score") + xlab("Network Size") +  
 theme(text = element\_text(size = 16)) + ylim(0,1) +   
 scale\_colour\_manual(values = c("red", "blue"), labels = c("Can See Trials", "Does Not See Trials")) +  
 geom\_smooth(method = "lm", se = FALSE)  
plot\_lines



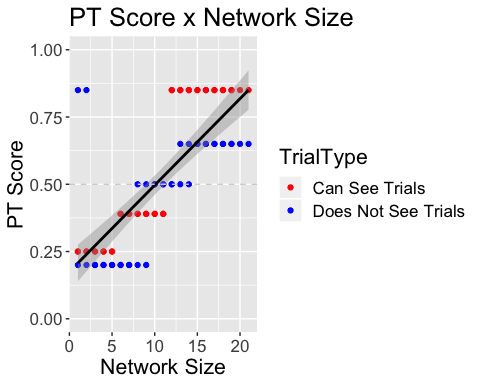
If you just want the **main effect** (i.e., the regression line for the effect of Networks Size on PT score), then the code should look like this. You need to move ‘colour = TrialType’ into geom\_point. This has to do with the ‘grammar of graphics’ and how ggplot is understanding your data.

plot\_lines <- ggplot(dummydata, aes(NetworkSize, PT.Score)) +  
 geom\_point(aes(colour = TrialType)) + ggtitle("PT Score x Network Size") + ylab("PT Score") + xlab("Network Size") +  
 theme(text = element\_text(size = 16)) + ylim(0,1) +   
 scale\_colour\_manual(values = c("red", "blue"), labels = c("Can See Trials", "Does Not See Trials")) +  
 geom\_smooth(method = "lm", col = "black")  
plot\_lines



Adding a line to indicate change performance. geom\_hline(yintercept=20, linetype=“dashed”, color = “red”, size=2)

plot\_lines2 <- ggplot(dummydata, aes(NetworkSize, PT.Score)) +  
 geom\_point(aes(colour = TrialType)) + ggtitle("PT Score x Network Size") + ylab("PT Score") + xlab("Network Size") +  
 theme(text = element\_text(size = 16)) + ylim(0,1) +   
 scale\_colour\_manual(values = c("red", "blue"), labels = c("Can See Trials", "Does Not See Trials")) +  
 geom\_smooth(method = "lm", col = "black") + geom\_hline(yintercept = .5, linetype = "dashed",  
 color = "light grey")  
plot\_lines2



### A plot with lots of things that you can explore.

I am including a final plot of other things I have done to ggplot. Feel free to work through this code to see how you can change the end result.

If you want to change the font of ggplots, you have to run the following code:

library(extrafont)

## Registering fonts with R

font\_import()

## Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system.  
## Continue? [y/n]

## Exiting.

A beautiful, finished product!

ggplot(dummydata, aes(NetworkSize, PT.Score)) +  
 # this line changes properties of the points  
 geom\_point(size = 2.5, aes(colour = TrialType)) +  
 # this line adds titles   
 ylab("PT Score") + xlab("Network Size") + ggtitle("PT Score x Network Size") +  
 # this line changes the axis and "jitters" pts - it spreads them out  
 ylim(0,1) + xlim(0,20) + geom\_jitter(size = 2.5, aes(colour = TrialType)) +  
 # this line changes the color  
 scale\_colour\_manual(values = c("blue", "red"), labels = c("Can See", "Does Not See")) +  
 # this line adds a horizontal line  
 geom\_hline(yintercept = .50, linetype = "dashed", col = "black") +   
 # this line changes the font and font size   
 theme(text = element\_text(size = 16, family = "Georgia")) +  
 # this gets rid of the grey background  
 theme(panel.background = element\_blank()) +  
 # this adds back axis lines   
 theme(axis.line = element\_line(colour = "black")) +  
 # this moves the legend to the bottom  
 theme(legend.position = "bottom") +  
 # this makes the numbers on the axis bigger   
 theme(axis.text.x = element\_text(size = 14)) +  
 theme(axis.text.y = element\_text(size = 14)) +  
 # this adds the regression line  
 geom\_smooth(method = "lm", col = "black")

