Section 6

Cassie E. Skipper

September 18, 2020

## Grammar of Graphics - ggplot2 - S6, L60

# Grammar of Graphics: Components of/Layers of Information in a Plot

1. Data
2. Aesthetics
3. Geometrics
4. Statistics
5. Facets
6. Coordinates
7. Theme

## What is a Factor? - S6, L61

getwd() setwd(“~/R Programming A-Z/Section 6 - Advance Visualization/SuperDataScience files”) getwd()

# 

movies <- read.csv(“P2-Movie-Ratings.csv”) head(movies) colnames(movies) <- c(“Film”, “Genre”, “CriticRating”, “AudienceRating”, “BudgetMillions”, “Year”) head(movies) tail(movies) str(movies) check the variable (column) types (factor vs integer)

What is a Factor? Factors are categorical variables; factors are the way R works with categorical variables levels and numbers; R looks through your column finds all of the different words that exist and assigns each word a number

summary(movies) we want Year to be a categorical variable aka factor

How to convert a non-factor numeric variable –> factor factor(moviesYear <- factor(movies$Year) assign column in the dataset the new factored vector summary(movies) Year is now a factor str(movies) confirm that Year is now a factor

—————- Aesthetics - S6, L62

Aesthetics - how your data maps to what you want to see

library(ggplot2)

last time we used qplot() - q stands for quick plot ggplot is the better, more in-depth/customizable option ggplot(data=movies, aes(x=CriticRating, y=AudienceRating)) if you run just this, you will get an empty plot

add geometry ggplot(data=movies, aes(x=CriticRating, y=AudienceRating)) + geom\_point()

add color ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre)) + geom\_point()

add size ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre, size=Genre)) + geom\_point() Warning message: Using size for a discrete variable is not advised.

add size - better way ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre, size=BudgetMillions)) + geom\_point() >>> This is 1 (but we will further improve it later)

—————- Plotting with Layers - S6, L63

p <- ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre, size=BudgetMillions))

point p + geom\_point()

lines p + geom\_line()

multiple layers p + geom\_point() + geom\_line() p + geom\_line() + geom\_point() points on top of lines

—————- Overriding Aesthetics - S6, L64

q <- ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre, size=BudgetMillions))

add geom layer q + geom\_point()

overriding aesthetics - example 1 q + geom\_point(aes(size=CriticRating))

overriding aesthetics - example 2 q + geom\_point(aes(colour=BudgetMillions))

q remains the same q + geom\_point()

override x and y - example 3 q + geom\_point(aes(x=BudgetMillions)) doesn’t change the x-axis label q + geom\_point(aes(x=BudgetMillions)) + xlab(“Budget Millions $$$”)

reduce line size - example 4 q + geom\_line(size=1) + geom\_point()  
Mapping vs. Setting: aes() is mapping; assigning size=1 is just setting so you don’t need aes()

—————- Mapping vs. Setting - S6, L65

q <- ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre, size=BudgetMillions))

override x and y - example 3 q + geom\_point(aes(x=BudgetMillions)) + xlab(“Budget Millions $$$”)

r <- ggplot(data=movies, aes(x=CriticRating, y=AudienceRating)) r + geom\_point()

Add colour 1. Mapping (what we’ve done so far) r + geom\_point(aes(colour=Genre)) mapping color to Genre variable 2. Setting r + geom\_point(colour=“DarkGreen”) setting color at DarkGreen ERROR: r + geom\_point(aes(colour=“DarkGreen”))  
you’re actually mapping DarkGreen as a new variable - not as a color!!

1. Mapping r + geom\_point(aes(size=BudgetMillions))
2. Setting r + geom\_point(size=10) ERROR: r + geom\_point(aes(size=10)) r sees size=10 as a variable

—————- Histograms and Density Charts - S6, L66

Histograms s <- ggplot(data=movies, aes(x=BudgetMillions)) s + geom\_histogram(binwidth=10) count (y-axis) is generated by r; it’s a statistic

add colour s + geom\_histogram(binwidth=10, fill=“Green”) setting s + geom\_histogram(binwidth=10, aes(fill=Genre)) mapping

add a border s + geom\_histogram(binwidth=10, aes(fill=Genre), colour=“Black”) color of fill is mapped to Genre color of the borders is set to Black >>> Chart 3 (we will improve it later)

Density Charts sometimes you may need them s + geom\_density(aes(fill=Genre)) s + geom\_density(aes(fill=Genre), position=“stack”) another option: use + geom\_area(position = “stack”) if you have x and y vars: s + geom\_density(aes(fill=Genre)) + geom\_area(position=“stack”)

—————- Starting Layer Tips - S6, L67

t <- ggplot(data=movies, aes(x=AudienceRating)) t + geom\_histogram(binwidth=10, fill=“White”, colour=“Blue”)  
fill and border colors are set (not mapped)

another way to achieve this^ plot (faster to adapt) t <- ggplot(data=movies) t + geom\_histogram(binwidth=10, aes(x=AudienceRating), fill=“White”, colour=“Blue”) >>> Chart 4

t + geom\_histogram(binwidth=10, aes(x=CriticRating), fill=“White”, colour=“Blue”) >>> Chart 5

t <- ggplot() skeleton plot

—————- Statistical Transformations - S6, L68

library(ggplot2)

?geom\_smooth()

u <- ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre)) u + geom\_point() + geom\_smooth() u + geom\_point() + geom\_smooth(fill=NA)

Boxplots u <- ggplot(data=movies, aes(x=Genre, y=AudienceRating, colour=Genre)) u + geom\_boxplot() u + geom\_boxplot(size=1.2) see bonus tutorial for boxplots u + geom\_boxplot(size=1.2) + geom\_point() tip / hack: u + geom\_boxplot(size=1.2) + geom\_jitter() geom\_jitter() puts the dots on the boxplots to help visualize the data another way: u + geom\_jitter() + geom\_boxplot(size=1.2) dots behind the boxplots another example: u + geom\_jitter() + geom\_boxplot(size=1.2, alpha=0.5) alpha controls transparency (0-1) boxplots are 50% transparent >>> Chart 6

Challenge t <- ggplot(data=movies, aes(x=Genre, y=CriticRating, colour=Genre)) t + geom\_jitter() + geom\_boxplot(size=1.2, alpha=0.5)

—————- Using Facets - S6, L69

v <- ggplot(data=movies, aes(x=BudgetMillions)) v + geom\_histogram(binwdith=10) v + geom\_histogram(binwdith=10, aes(fill=Genre), colour=“Black”)

Facets: allow you to create lots of charts histogram for each genre

uniform scales: v + geom\_histogram(binwdith=10, aes(fill=Genre), colour=“Black”) + facet\_grid(Genre~.) rows=Genre facet\_grid(row~column) notice that the Comedy genre takes up the entire scale all scales are uniform, unless specified

different scales: v + geom\_histogram(binwdith=10, aes(fill=Genre), colour=“Black”) + facet\_grid(Genre~., scales=“free”) scales=free removes scale uniformity

Scatterplots: w <- ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, colour=Genre)) w + geom\_point(size=3)

scatterplot facets: w + geom\_point(size=3) + facet\_grid(Genre~.)

w + geom\_point(size=3) + facet\_grid(.~Year)

w + geom\_point(size=3) + facet\_grid(Genre~Year)

w + geom\_point(size=3) + geom\_smooth() + facet\_grid(Genre~Year)

w + geom\_point(aes(size=BudgetMillions)) + geom\_smooth() + facet\_grid(Genre~Year) >>> Chart 1 (but stil will improve) the geom\_smooth is stretching the axes; we need to zoom in (see L70)

—————- Coordinates - S6, L70

limits zooming in & out

w + geom\_point(aes(size=BudgetMillions)) + geom\_smooth() + facet\_grid(Genre~Year)

m <- ggplot(data=movies, aes(x=CriticRating, y=AudienceRating, size=BudgetMillions, colour=Genre)) m + geom\_point()

zoom in on top right quadrant m + geom\_point() + xlim(50,100) + ylim(50,100)

^this won’t always work well n <- ggplot(data=movies, aes(x=BudgetMillions)) n + geom\_histogram(binwidth=10, aes(fill=Genre), colour=“Black”) cut if off up to 50 n + geom\_histogram(binwidth=10, aes(fill=Genre), colour=“Black”) + ylim(0,50) this cuts off any data that goes to 50 or beyond

instead of this^, Zoom in using coord\_cartesian(): n + geom\_histogram(binwidth=10, aes(fill=Genre), colour=“Black”) + coord\_cartesian(ylim=c(0,50))

back to improving Chart 1; zoom in w + geom\_point(aes(size=BudgetMillions)) + geom\_smooth() + facet\_grid(Genre~Year) + coord\_cartesian(ylim=c(0,100)) >>> Chart $1

—————- Perfecting by Adding Themes - S6, L71

Themes: includes all non-data ink ex: position of labels and legends, color of background, text

improving s (Chart 3) from L66 Histograms & Density Charts o <- ggplot(data=movies, aes(x=BudgetMillions)) h <- o + geom\_histogram(binwidth=10, aes(fill=Genre), colour=“Black”) h

axes labels h + xlab(“Money Axis”) + ylab(“Number of Movies”)

label formatting h + xlab(“Money Axis”) + ylab(“Number of Movies”) + theme(axis.title.x = element\_text(colour=“DarkGreen”, size=15), axis.title.y = element\_text(colour=“Red”, size=15))

tick mark formatting h + xlab(“Money Axis”) + ylab(“Number of Movies”) + theme(axis.title.x = element\_text(colour=“DarkGreen”, size=15), axis.title.y = element\_text(colour=“Red”, size=15), axis.text.x = element\_text(size=10), axis.text.y = element\_text(size=10))

?theme()

legend formatting h + xlab(“Money Axis”) + ylab(“Number of Movies”) + theme(axis.title.x = element\_text(colour=“DarkGreen”, size=15), axis.title.y = element\_text(colour=“Red”, size=15), axis.text.x = element\_text(size=10), axis.text.y = element\_text(size=10),

legend.title = element\_text(size=15),  
 legend.text = element\_text(size=10),  
 legend.position = c(1,1), can pass 0 (start of axis) or 1 (end of axis)  
 legend.justification = c(1.05,1.05)) anchors the legend

import fonts install.packages(“extrafont”) library(extrafont) font\_import(“C:/Windows/Fonts”) loadfonts(device = “win”)

plot title h + xlab(“Money Axis”) + ylab(“Number of Movies”) + ggtitle(“Movie Budget Distribution”) + theme(axis.title.x = element\_text(colour=“DarkGreen”, size=15), axis.title.y = element\_text(colour=“Red”, size=15), axis.text.x = element\_text(size=10), axis.text.y = element\_text(size=10),

legend.title = element\_text(size=15),  
 legend.text = element\_text(size=10),  
 legend.position = c(1,1), can pass 0 (start of axis) or 1 (end of axis)  
 legend.justification = c(1.05,1.05), anchors the legend  
   
 plot.title = element\_text(colour="DarkBlue",   
 size=20,   
 family="Courier New",  
 hjust=0.5))