RWorksheet\_Gonzales#6

Mamerto F. Gonzales Jr.

2022-11-24

#WorkSheet-6

Plotting using ggplot() and plot() Worksheet for R Programming Instructions: • Use RStudio or the RStudio Cloud accomplish this worksheet. • Save the R script as RWorksheet\_lastname#6.R. • On your own GitHub repository, push the R script, the Rmd file, as well as this pdf worksheet to the repo you have created before. • Do not forget to comment your Git repo on our VLE • Accomplish this worksheet by answering the questions being asked and writing the code manually.

Use the dataset mpg library(ggplot2) #to get the mpg dataset, load the ggplot package first data(mpg) as.data.frame(data(mpg)) #converting from list to data frame ## data(mpg) ## 1 mpg A data frame with 234 rows and 11 variables: #’ \describe{ #’

{manufacturer}{manufacturer name} #’

{model}{model name} #’

{displ}{engine displacement, in litres} #’

{year}{year of manufacture} #’

{cyl}{number of cylinders} 1 #’

{trans}{type of transmission} #’

{drv}{the type of drive train, where f = front-wheel drive, r = rear wheel dri#’

{cty}{city miles per gallon} #’

{hwy}{highway miles per gallon} #’

{fl}{fuel type} #’

{class}{“type” of car} #’ } “mpg” str(mpg) ## tibble [234 x 11] (S3: tbl\_df/tbl/data.frame) ## $ manufacturer: chr [1:234] “audi” “audi” “audi” “audi” … ## $ model : chr [1:234] “a4” “a4” “a4” “a4” … ## $ displ : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 … ## $ year : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 … ## $ cyl : int [1:234] 4 4 4 4 6 6 6 4 4 4 … ## $ trans : chr [1:234] “auto(l5)” “manual(m5)” “manual(m6)” “auto(av)” … ## $ drv : chr [1:234] “f” “f” “f” “f” … ## $ cty : int [1:234] 18 21 20 21 16 18 18 18 16 20 … ## $ hwy : int [1:234] 29 29 31 30 26 26 27 26 25 28 … ## $ fl : chr [1:234] “p” “p” “p” “p” … ## $ class : chr [1:234] “compact” “compact” “compact” “compact” … #use of glimpse() - much tidier compared to str() library(dplyr) #glimpse() is a function under dplyr package ## ## 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 2 glimpse(mpg) ## Rows: 234 ## Columns: 11 ## $ manufacturer “audi”, “audi”, “audi”, “audi”, “audi”, “audi”, “audi”, “~ ## $ model ”a4”, “a4”, “a4”, “a4”, “a4”, “a4”, “a4”, “a4 quattro”, “~ ## $ displ 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~ ## $ year 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~ ## $ cyl 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, ~ ## $ trans ”auto(l5)“,”manual(m5)“,”manual(m6)“,”auto(av)“,”auto~ ## $ drv “f”, “f”, “f”, “f”, “f”, “f”, “f”, “4”, “4”, “4”, “4”, “4~ ## $ cty 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~ ## $ hwy 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~ ## $ fl ”p”, “p”, “p”, “p”, “p”, “p”, “p”, “p”, “p”, “p”, “p”, “p~ ## $ class ”compact”, “compact”, “compact”, “compact”, “compact”, “c~ • Example. graph using ggplot() ggplot(mpg, aes(cty, hwy)) + geom\_point() 3 20 30 40 10 15 20 25 30 35 cty hwy

1. How many columns are in mpg dataset? How about the number of rows? Show the codes and its result.

mtcars

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

data(mpg)  
as.data.frame(data(mpg))

## data(mpg)  
## 1 mpg

mpg

## # A tibble: 234 × 11  
## manufacturer model displ year cyl trans drv cty hwy fl class  
## <chr> <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>  
## 1 audi a4 1.8 1999 4 auto… f 18 29 p comp…  
## 2 audi a4 1.8 1999 4 manu… f 21 29 p comp…  
## 3 audi a4 2 2008 4 manu… f 20 31 p comp…  
## 4 audi a4 2 2008 4 auto… f 21 30 p comp…  
## 5 audi a4 2.8 1999 6 auto… f 16 26 p comp…  
## 6 audi a4 2.8 1999 6 manu… f 18 26 p comp…  
## 7 audi a4 3.1 2008 6 auto… f 18 27 p comp…  
## 8 audi a4 quattro 1.8 1999 4 manu… 4 18 26 p comp…  
## 9 audi a4 quattro 1.8 1999 4 auto… 4 16 25 p comp…  
## 10 audi a4 quattro 2 2008 4 manu… 4 20 28 p comp…  
## # … with 224 more rows

ncol(mpg)

## [1] 11

nrow(mpg)

## [1] 234

* There are 11 columns and 234 rows in the mpg data frame.

1. Which manufacturer has the most models in this data set? Which model has the most variations? Ans:
2. Group the manufacturers and find the unique models. Copy the codes and result.

ManufacturerModels <- mpg %>%   
group\_by(manufacturer) %>%   
tally(sort = TRUE)  
ManufacturerModels

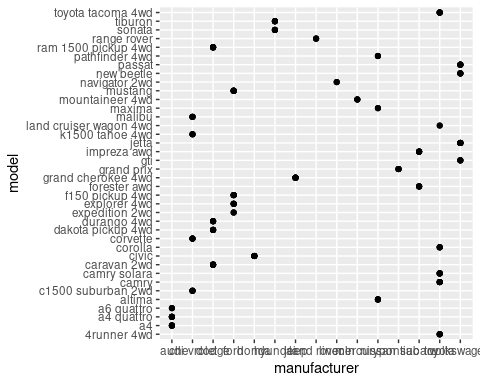
## # A tibble: 15 × 2  
## manufacturer n  
## <chr> <int>  
## 1 dodge 37  
## 2 toyota 34  
## 3 volkswagen 27  
## 4 ford 25  
## 5 chevrolet 19  
## 6 audi 18  
## 7 hyundai 14  
## 8 subaru 14  
## 9 nissan 13  
## 10 honda 9  
## 11 jeep 8  
## 12 pontiac 5  
## 13 land rover 4  
## 14 mercury 4  
## 15 lincoln 3

unique(mpg$model)

## [1] "a4" "a4 quattro" "a6 quattro"   
## [4] "c1500 suburban 2wd" "corvette" "k1500 tahoe 4wd"   
## [7] "malibu" "caravan 2wd" "dakota pickup 4wd"   
## [10] "durango 4wd" "ram 1500 pickup 4wd" "expedition 2wd"   
## [13] "explorer 4wd" "f150 pickup 4wd" "mustang"   
## [16] "civic" "sonata" "tiburon"   
## [19] "grand cherokee 4wd" "range rover" "navigator 2wd"   
## [22] "mountaineer 4wd" "altima" "maxima"   
## [25] "pathfinder 4wd" "grand prix" "forester awd"   
## [28] "impreza awd" "4runner 4wd" "camry"   
## [31] "camry solara" "corolla" "land cruiser wagon 4wd"  
## [34] "toyota tacoma 4wd" "gti" "jetta"   
## [37] "new beetle" "passat"

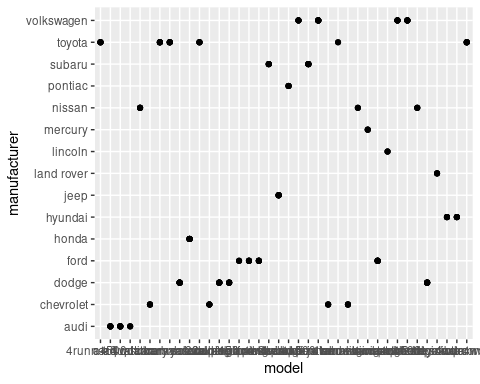
1. Graph the result by using plot() and ggplot(). Write the codes and its result.

ggplot(mpg, aes(manufacturer, model)) +  
geom\_point()



1. Same dataset will be used. You are going to show the relationship of the modeland the manufacturer.
2. What does ggplot(mpg, aes(model, manufacturer)) + geom\_point() show?

ggplot(mpg, aes(model, manufacturer)) + geom\_point()



1. For you, is it useful? If not, how could you modify the data to make it more informative?

* Yes it is very useful because it is very easy to get information from.

1. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result.

CarsModel <- mpg %>%   
group\_by(model) %>%   
tally(sort = TRUE)  
CarsModel

## # A tibble: 38 × 2  
## model n  
## <chr> <int>  
## 1 caravan 2wd 11  
## 2 ram 1500 pickup 4wd 10  
## 3 civic 9  
## 4 dakota pickup 4wd 9  
## 5 jetta 9  
## 6 mustang 9  
## 7 a4 quattro 8  
## 8 grand cherokee 4wd 8  
## 9 impreza awd 8  
## 10 a4 7  
## # … with 28 more rows

1. Plot using the geom\_bar() + coord\_flip() just like what is shown below. Show codes and its result.

ggplot(CarsModel, aes(x = model, y = n, fill = "rainbow")) +  
geom\_bar(stat = "identity") + coord\_flip()

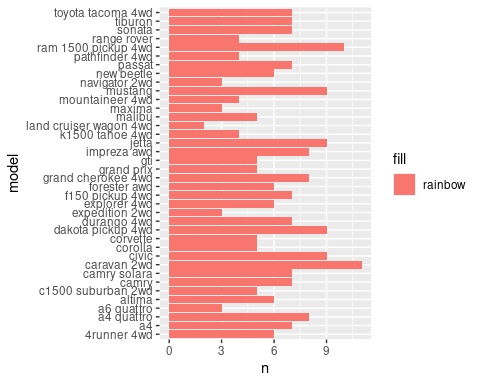
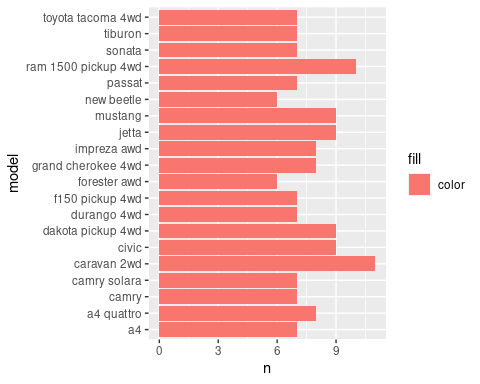


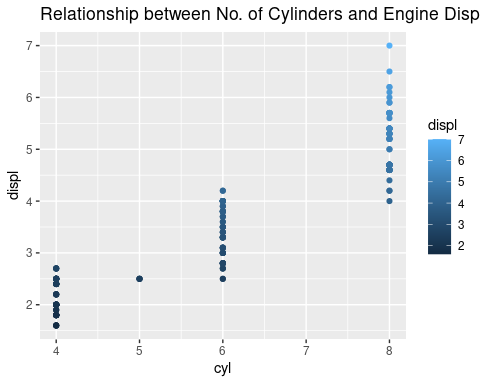
Figure 1: Car Models b. Use only the top 20 observations. Show code and results.

ggplot(CarsModel[tail(order(CarsModel$n), 20), ], ) +   
 aes(model, fill = "color", n) + geom\_bar(stat = "identity") + coord\_flip()



1. Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom\_point with aesthetic colour = engine displacement. Title should be “Relationship between No. of Cylinders and Engine Displacement”.
2. Show the codes and its result.

CylVsDispl <- ggplot(mpg, aes(x = cyl, y = displ, color = displ)) +  
geom\_point()  
print(CylVsDispl + ggtitle("Relationship between No. of Cylinders and Engine Displacement"))

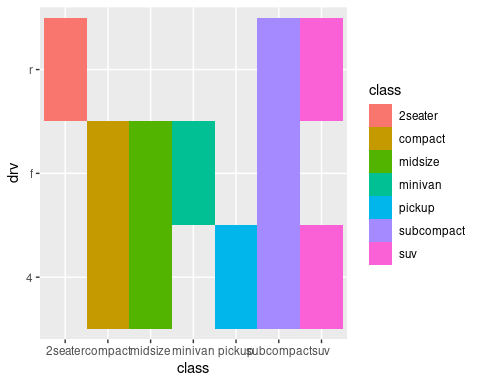


1. How would you describe its relationship?

* The higher the number of Cylinders, the engine displacement also rises.

1. Get the total number of observations for drv - type of drive train (f = front-wheel drive, r = rear wheel drive, 4 = 4wd) and class - type of class (Example: suv, 2seater, etc.). Plot using the geom\_tile() where the number of observations for class be used as a fill for aesthetics.
2. Show the codes and its result for the narrative in #6.

mpg %>%  
count(class, drv) %>%  
ggplot(aes(x = class, y = drv)) +  
geom\_tile(mapping = aes(fill = class))

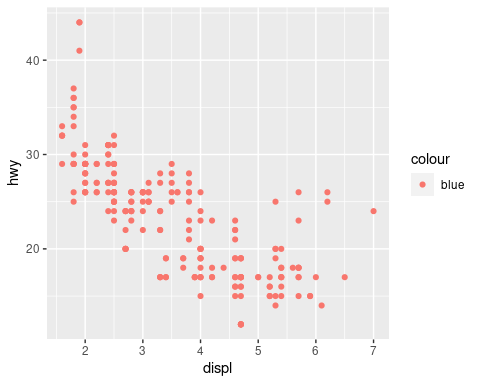


1. Interpret the result.

* Different types of cars have different types of Driving.

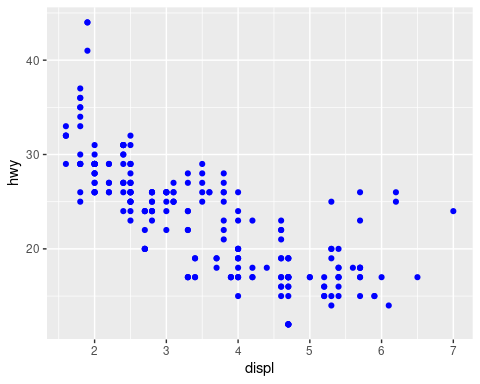
1. Discuss the difference between these codes. Its outputs for each are shown below. • Code #1

ggplot(data = mpg) +  
geom\_point(mapping = aes(x = displ, y = hwy, colour = "blue"))



* Code #2

ggplot(data = mpg) +  
geom\_point(mapping = aes(x = displ, y = hwy), colour = "blue")



* Putting the values inside the aes generates a legend and makes the color with that while putting it outside aes ggplot2 did not make the legend automatically and inputted your value.

1. Try to run the command ?mpg. What is the result of this command?

?mpg

* It scours the internet and shows its description and usage

1. Which variables from mpg dataset are categorical?

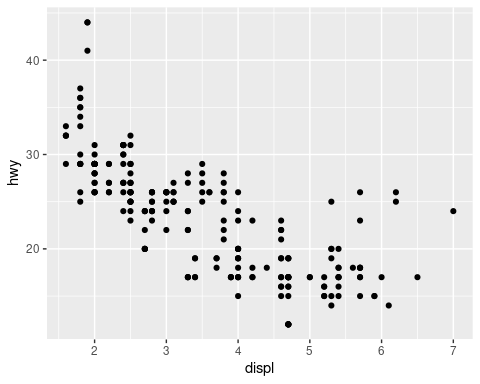
* The variables that are categorical in mpg dataset are manufacturer, model, trans, drv, fl, and class.

1. Which are continuous variables?

* The continuous variables in mpg dataset are displ, year, cyl, cty, and hwy

1. Plot the relationship between displ (engine displacement) and hwy(highway miles per gallon). Mapped it with a continuous variable you have identified in #5-b. What is its result? Why it produced such output?

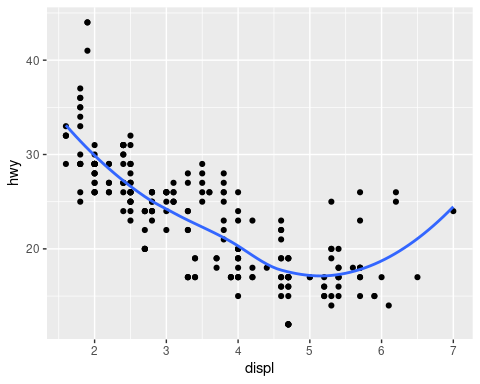
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
geom\_point()



1. Plot the relationship between displ (engine displacement) and hwy(highway miles per gallon) using geom\_point(). Add a trend line over the existing plot using geom\_smooth() with se = FALSE. Default method is “loess”.

ggplot(mpg, aes(displ, hwy)) +   
geom\_point() + geom\_smooth(se = FALSE)

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



1. Using the relationship of displ and hwy, add a trend line over existing plot. Set the se = FALSE to remove the confidence interval and method = lm to check for linear modeling.

ggplot(mpg, aes(displ, hwy)) +   
geom\_point() + geom\_smooth(method = "lm", se = FALSE)

## `geom\_smooth()` using formula = 'y ~ x'

