PracticeTestKeyTest1

James Dickens

9/20/2020

#412/612 PRACTICE TEST  
library(ggplot2)  
library(tidyverse)

## -- Attaching packages ------------------------------------------------------------------ tidyverse 1.3.0 --

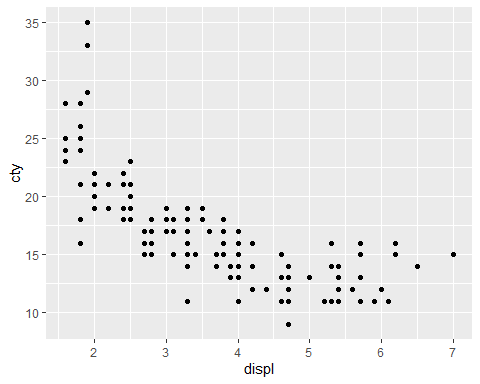
## v tibble 3.0.3 v dplyr 1.0.2  
## v tidyr 1.1.1 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0  
## v purrr 0.3.4

## -- Conflicts --------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

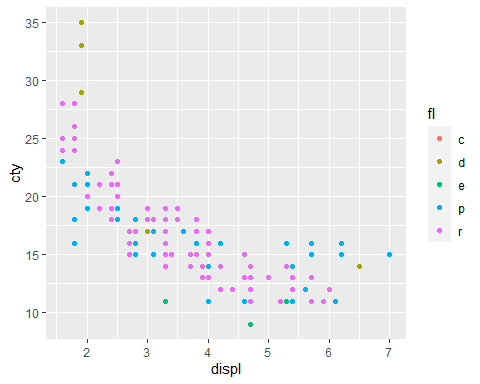
library(dplyr)  
mpg

## # A tibble: 234 x 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(l~ f 18 29 p comp~  
## 2 audi a4 1.8 1999 4 manual~ f 21 29 p comp~  
## 3 audi a4 2 2008 4 manual~ f 20 31 p comp~  
## 4 audi a4 2 2008 4 auto(a~ f 21 30 p comp~  
## 5 audi a4 2.8 1999 6 auto(l~ f 16 26 p comp~  
## 6 audi a4 2.8 1999 6 manual~ f 18 26 p comp~  
## 7 audi a4 3.1 2008 6 auto(a~ f 18 27 p comp~  
## 8 audi a4 quat~ 1.8 1999 4 manual~ 4 18 26 p comp~  
## 9 audi a4 quat~ 1.8 1999 4 auto(l~ 4 16 25 p comp~  
## 10 audi a4 quat~ 2 2008 4 manual~ 4 20 28 p comp~  
## # ... with 224 more rows

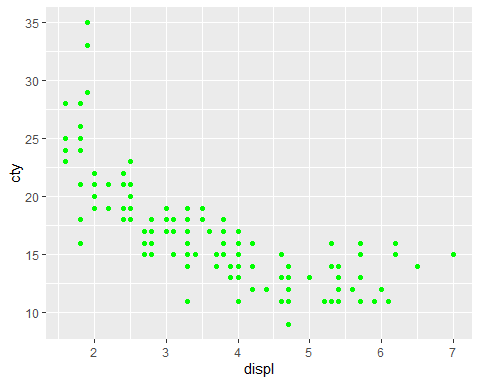
#PART 1 GGPLOT PRACTICE (Use tidyverse coding for all problems)  
#1) Using the mpg data frame, create a scatter plot that shows a relationship between the   
# variables displ and cty. (displ =x and cty=y)  
  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = cty))



#2) Using the mpg data frame, create a scatter plot that shows a relationship between the   
# variables displ and cty. (displ =x and cty=y), and also map colors of your scatter plot to the   
# variable fl.  
  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = cty, color = fl))



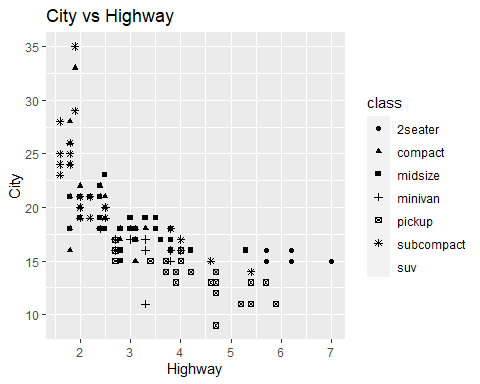
#3) Using the mpg data frame, create a scatter plot that shows a relationship between the variables   
# displ and cty. (displ =x and cty=y), and also include code so that all of your scatter plot   
# data points are green.  
  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = cty), color = "green")



#4) Using the mpg data frame, create a scatter plot that shows a relationship between the variables  
# displ and cty. (displ =x and cty=y), and also include code so that the scatter plot has different  
# shapes or characters according to class. Code so that your scatter plot has the title   
# City vs Highway the y axis is labled City and the x axis is labeld Highway.  
# (Check out the ggplot graphing example towards the end of Chapter one in your book to find out how  
# to code a scatter plot for different shapes.  
  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = cty, shape = class)) +  
 ggtitle("City vs Highway") +  
 xlab("Highway")+  
 ylab("City")

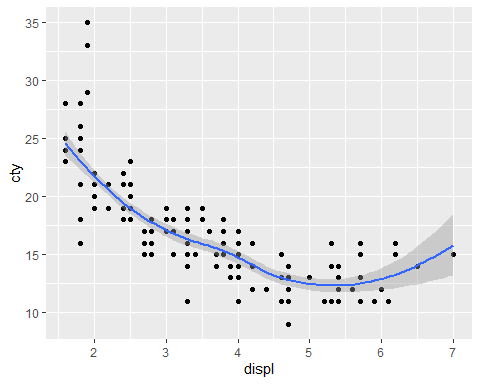
## Warning: The shape palette can deal with a maximum of 6 discrete values because  
## more than 6 becomes difficult to discriminate; you have 7. Consider  
## specifying shapes manually if you must have them.

## Warning: Removed 62 rows containing missing values (geom\_point).



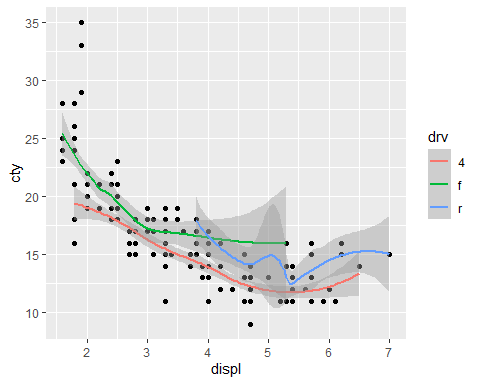
#5) Using the mpg data frame, create a smooth line fitted to the data displ and cty.   
# (displ =x and cty=y).  
  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = cty)) +  
 geom\_smooth(mapping = aes(x = displ, y = cty))

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



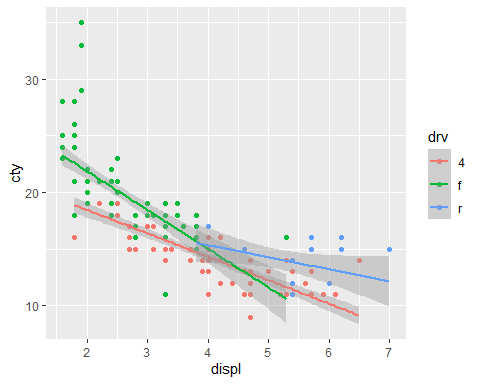
#6) Using the mpg data frame, create smooth lines over the scatter plot for the data displ and cty.   
# (displ =x and cty=y) based on drv levels.  
  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = cty)) +  
 geom\_smooth(mapping = aes(x = displ, y = cty, color = drv))

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

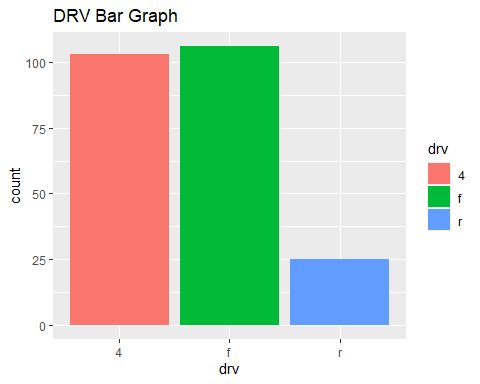


#7) Using the mpg data frame, create a scatter plot that shows a relationship between the  
# variables displ and cty. (displ =x and cty=y), and also include code that produces overlayed   
# regression lines based on drv levels. The data points should also be color coded according to  
# levels of drv  
  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = cty, color =drv)) +  
 geom\_smooth(method =lm, mapping = aes(x = displ, y = cty, color = drv))

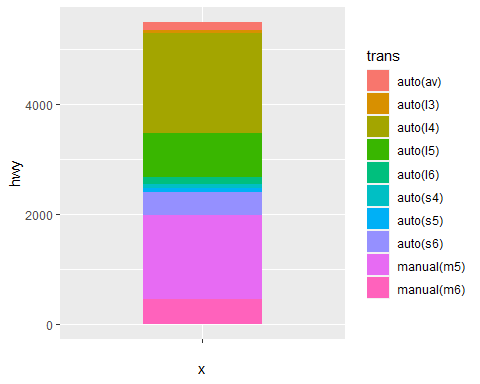
## `geom\_smooth()` using formula 'y ~ x'



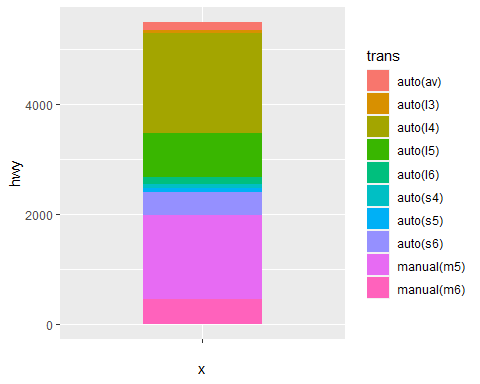
#8) Using the mpg data frame, create a bar graph for the variable drv. Include code so that  
# each bar has a different color. Title your bar graph DRV Bar Graph  
  
ggplot(data = mpg) +  
 geom\_bar(mapping = aes(x = drv, fill = drv)) +  
 ggtitle("DRV Bar Graph")



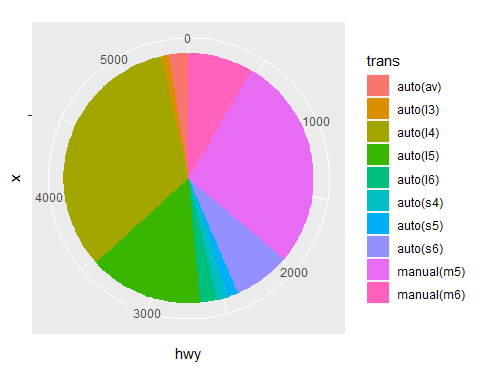
#9. Using the mpg data frame, for the variable trans, create a bar graph that shows color  
#stacked bars over the variable trans for the variable hwy  
  
ggplot(data = mpg, aes(x="", y = hwy, fill=trans))+  
 geom\_bar(width = .5, stat = "identity")



#10. Using the stacked bar graph from number 10, create a pie chart.  
  
ggplot(data = mpg, aes(x="", y = hwy, fill=trans))+  
 geom\_bar(width = .5, stat = "identity") -> sbg  
sbg



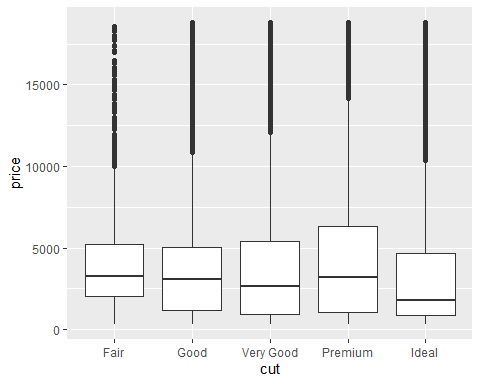
pie1 <- sbg + coord\_polar("y", start=0)   
pie1



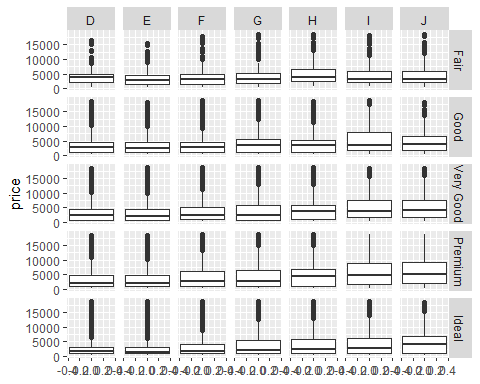
# 11.  
  
diamonds

## # A tibble: 53,940 x 10  
## carat cut color clarity depth table price x y z  
## <dbl> <ord> <ord> <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43  
## 2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31  
## 3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31  
## 4 0.290 Premium I VS2 62.4 58 334 4.2 4.23 2.63  
## 5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75  
## 6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48  
## 7 0.24 Very Good I VVS1 62.3 57 336 3.95 3.98 2.47  
## 8 0.26 Very Good H SI1 61.9 55 337 4.07 4.11 2.53  
## 9 0.22 Fair E VS2 65.1 61 337 3.87 3.78 2.49  
## 10 0.23 Very Good H VS1 59.4 61 338 4 4.05 2.39  
## # ... with 53,930 more rows

ggplot(data = diamonds) +  
 geom\_boxplot(mapping = aes(x = cut, y = price))

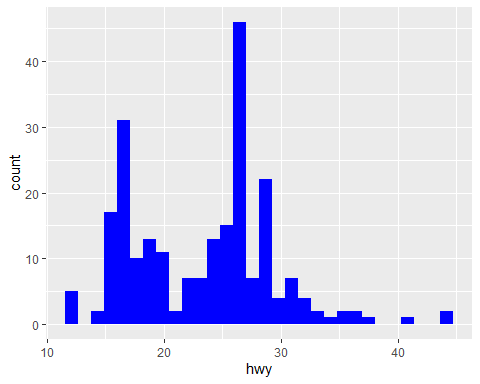


# 12  
  
ggplot(data = diamonds) +  
 geom\_boxplot(mapping = aes(y = price)) +  
 facet\_grid(cut~color)



# 13  
ggplot(data = mpg) +  
 geom\_histogram(mapping = aes( x = hwy ), fill = "blue")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# DPLYR Problems  
# 1.   
mpg%>%   
 select(manufacturer,model)%>%  
 filter(manufacturer == "dodge", model == "durango 4wd")

## # A tibble: 7 x 2  
## manufacturer model   
## <chr> <chr>   
## 1 dodge durango 4wd  
## 2 dodge durango 4wd  
## 3 dodge durango 4wd  
## 4 dodge durango 4wd  
## 5 dodge durango 4wd  
## 6 dodge durango 4wd  
## 7 dodge durango 4wd

# 2.  
mpg%>%  
 select(manufacturer,model,cty,hwy)%>%  
 filter(cty<10, hwy<16)

## # A tibble: 5 x 4  
## manufacturer model cty hwy  
## <chr> <chr> <int> <int>  
## 1 dodge dakota pickup 4wd 9 12  
## 2 dodge durango 4wd 9 12  
## 3 dodge ram 1500 pickup 4wd 9 12  
## 4 dodge ram 1500 pickup 4wd 9 12  
## 5 jeep grand cherokee 4wd 9 12

# 3.  
mpg%>%  
 select(manufacturer,model,displ)%>%  
 arrange(desc(displ))

## # A tibble: 234 x 3  
## manufacturer model displ  
## <chr> <chr> <dbl>  
## 1 chevrolet corvette 7   
## 2 chevrolet k1500 tahoe 4wd 6.5  
## 3 chevrolet corvette 6.2  
## 4 chevrolet corvette 6.2  
## 5 jeep grand cherokee 4wd 6.1  
## 6 chevrolet c1500 suburban 2wd 6   
## 7 dodge durango 4wd 5.9  
## 8 dodge ram 1500 pickup 4wd 5.9  
## 9 chevrolet c1500 suburban 2wd 5.7  
## 10 chevrolet corvette 5.7  
## # ... with 224 more rows

#4.  
mpg%>%  
 select(manufacturer,model,trans,displ,cty,hwy)%>%  
 arrange(cty)%>%  
 print(n=30)

## # A tibble: 234 x 6  
## manufacturer model trans displ cty hwy  
## <chr> <chr> <chr> <dbl> <int> <int>  
## 1 dodge dakota pickup 4wd auto(l5) 4.7 9 12  
## 2 dodge durango 4wd auto(l5) 4.7 9 12  
## 3 dodge ram 1500 pickup 4wd auto(l5) 4.7 9 12  
## 4 dodge ram 1500 pickup 4wd manual(m6) 4.7 9 12  
## 5 jeep grand cherokee 4wd auto(l5) 4.7 9 12  
## 6 chevrolet c1500 suburban 2wd auto(l4) 5.3 11 15  
## 7 chevrolet k1500 tahoe 4wd auto(l4) 5.3 11 14  
## 8 chevrolet k1500 tahoe 4wd auto(l4) 5.7 11 15  
## 9 dodge caravan 2wd auto(l4) 3.3 11 17  
## 10 dodge dakota pickup 4wd manual(m5) 5.2 11 17  
## 11 dodge dakota pickup 4wd auto(l4) 5.2 11 15  
## 12 dodge durango 4wd auto(l4) 5.2 11 16  
## 13 dodge durango 4wd auto(l4) 5.9 11 15  
## 14 dodge ram 1500 pickup 4wd auto(l4) 5.2 11 15  
## 15 dodge ram 1500 pickup 4wd manual(m5) 5.2 11 16  
## 16 dodge ram 1500 pickup 4wd auto(l4) 5.9 11 15  
## 17 ford expedition 2wd auto(l4) 4.6 11 17  
## 18 ford expedition 2wd auto(l4) 5.4 11 17  
## 19 ford f150 pickup 4wd auto(l4) 5.4 11 15  
## 20 jeep grand cherokee 4wd auto(l5) 6.1 11 14  
## 21 land rover range rover auto(l4) 4 11 15  
## 22 land rover range rover auto(l4) 4.6 11 15  
## 23 lincoln navigator 2wd auto(l4) 5.4 11 17  
## 24 lincoln navigator 2wd auto(l4) 5.4 11 16  
## 25 toyota land cruiser wagon 4wd auto(l4) 4.7 11 15  
## 26 chevrolet c1500 suburban 2wd auto(l4) 6 12 17  
## 27 dodge ram 1500 pickup 4wd manual(m6) 4.7 12 16  
## 28 dodge ram 1500 pickup 4wd manual(m6) 4.7 12 16  
## 29 ford expedition 2wd auto(l6) 5.4 12 18  
## 30 land rover range rover auto(s6) 4.2 12 18  
## # ... with 204 more rows

#5.  
mpg%>%  
 select(manufacturer,year,cty,hwy)%>%  
 filter(manufacturer=="ford",year == 1999,cty<16,hwy<16)

## # A tibble: 1 x 4  
## manufacturer year cty hwy  
## <chr> <int> <int> <int>  
## 1 ford 1999 11 15

#6.  
mpg%>%  
 select(manufacturer,model,cty,hwy)%>%  
 filter(manufacturer=="ford",model=="mustang")%>%  
 mutate(difference = (hwy - cty))

## # A tibble: 9 x 5  
## manufacturer model cty hwy difference  
## <chr> <chr> <int> <int> <int>  
## 1 ford mustang 18 26 8  
## 2 ford mustang 18 25 7  
## 3 ford mustang 17 26 9  
## 4 ford mustang 16 24 8  
## 5 ford mustang 15 21 6  
## 6 ford mustang 15 22 7  
## 7 ford mustang 15 23 8  
## 8 ford mustang 15 22 7  
## 9 ford mustang 14 20 6

#7.  
library(nycflights13)  
library(dplyr)  
library(tidyverse)  
  
  
flights%>%  
 select(carrier,year,month, day, arr\_delay)%>%  
 filter(carrier =="AA")%>%  
 filter(year == 2013, month == 3,day ==17, arr\_delay > 10)%>%  
 arrange(desc(arr\_delay))

## # A tibble: 16 x 5  
## carrier year month day arr\_delay  
## <chr> <int> <int> <int> <dbl>  
## 1 AA 2013 3 17 67  
## 2 AA 2013 3 17 39  
## 3 AA 2013 3 17 39  
## 4 AA 2013 3 17 36  
## 5 AA 2013 3 17 33  
## 6 AA 2013 3 17 22  
## 7 AA 2013 3 17 22  
## 8 AA 2013 3 17 21  
## 9 AA 2013 3 17 19  
## 10 AA 2013 3 17 19  
## 11 AA 2013 3 17 16  
## 12 AA 2013 3 17 15  
## 13 AA 2013 3 17 14  
## 14 AA 2013 3 17 13  
## 15 AA 2013 3 17 12  
## 16 AA 2013 3 17 12

#8.   
mpg%>%  
 select(manufacturer , year,cty, model)%>%  
 filter(year == 1999)%>%  
 group\_by(manufacturer) %>%  
 summarise(AverageCityMileage = mean(cty))%>%  
 arrange(desc(AverageCityMileage))%>%  
 print(n = 30)

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 15 x 2  
## manufacturer AverageCityMileage  
## <chr> <dbl>  
## 1 honda 24.8  
## 2 volkswagen 21.2  
## 3 subaru 19   
## 4 hyundai 18.3  
## 5 toyota 18.2  
## 6 nissan 17.7  
## 7 audi 17.1  
## 8 pontiac 17   
## 9 chevrolet 15.1  
## 10 jeep 14.5  
## 11 ford 13.9  
## 12 mercury 13.5  
## 13 dodge 13.4  
## 14 land rover 11   
## 15 lincoln 11