412/612 Practice Test KEY

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

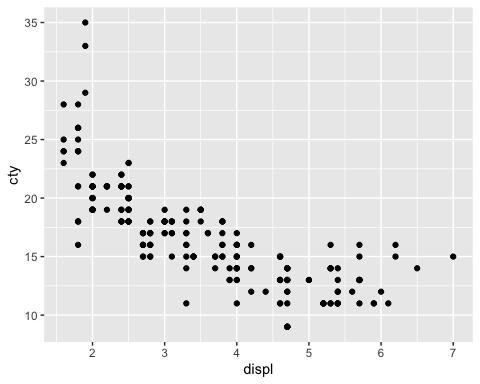
## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

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

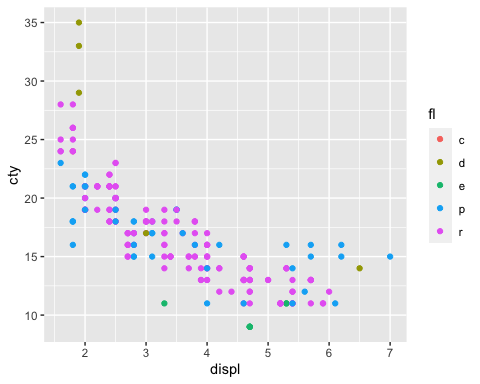
library(ggplot2)  
library(dplyr)  
  
#1) Using the mpg data frame, create a scatter plot that shows a relationship between the variables displ and cty. (displ =x and cty=y)  
mpg

## # 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

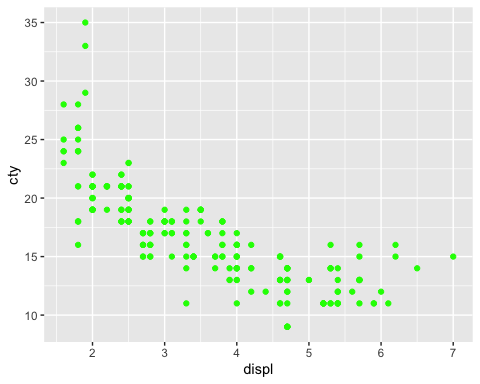
#View(mpg)  
#?mpg  
  
ggplot(data=mpg) +  
 geom\_point(mapping = aes (x=displ, y=cty))



#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)  
#Also map colors of your scatter plot to the variable fl.  
  
ggplot(data=mpg) +  
 geom\_point(mapping = aes (x=displ, y=cty, color=fl))



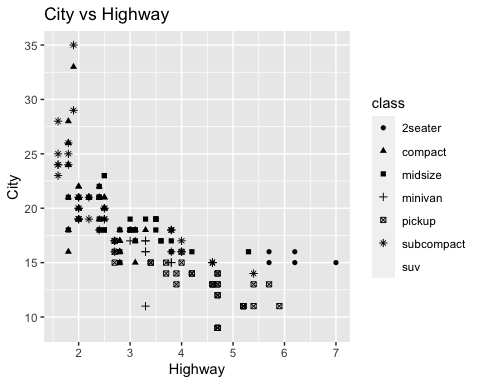
#3. Using the mpg data frame, create a scatter plot that shows a relationship between the variables displ and cty. (displ =x and cty=y)  
#also include code so that all of your scatter plot points are green.  
  
ggplot(data=mpg) +  
 geom\_point(mapping = aes (x=displ, y=cty), color = "green")



#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)  
#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 get data points with different shapes)  
  
ggplot(data=mpg) +  
 geom\_point(mapping = aes (x=displ, y=cty, shape=class)) +  
 ggtitle("City vs Highway")+  
 ylab("City") +  
 xlab("Highway")

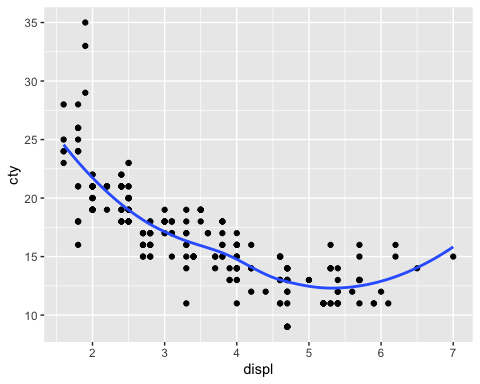
## Warning: The shape palette can deal with a maximum of 6 discrete values because  
## more than 6 becomes difficult to discriminate; you have 7. Consider  
## specifying shapes manually if you must have them.

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



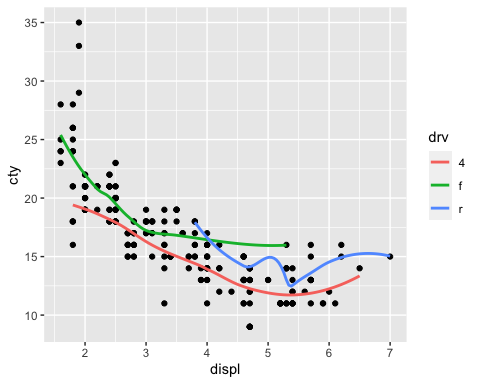
#5. Using the mpg data frame, create a smooth line fitted to the data displ and cty. (displ =x and cty=y).  
ggplot(data = mpg) +  
 geom\_point(mapping = aes (x = displ, y = cty)) +  
 geom\_smooth(mapping = aes(x = displ, y = cty), formula = y~x, se= FALSE)

## `geom\_smooth()` using method = 'loess'

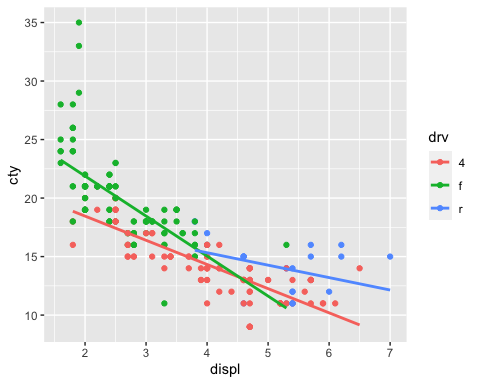


#6. Using the mpg data frame, create smooth lines over the scatter plot for the data displ and cty. (displ =x and cty=y) based on drv levels.  
ggplot(data = mpg) +  
 geom\_point(mapping = aes (x = displ, y = cty)) +  
 geom\_smooth(mapping = aes(x = displ, y = cty, color=drv), formula = y~x, se= FALSE)

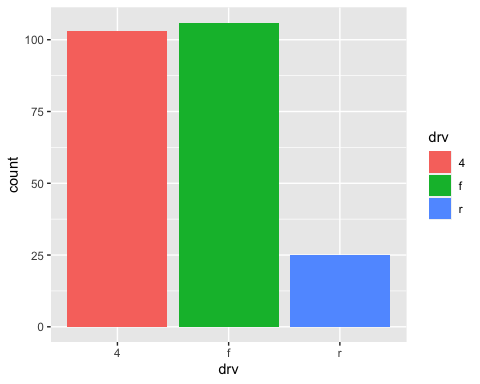
## `geom\_smooth()` using method = 'loess'



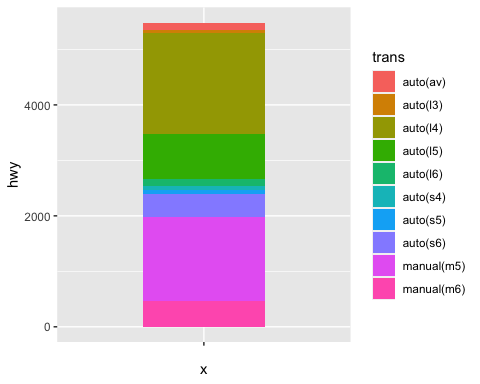
#7. Using the mpg data frame, create a scatter plot that shows a relationship between the variables displ and cty (displ =x and cty=y)  
#and also include code that produces overlayed regression lines based on drv levels.   
#The data points should also be color coded according to levels of drv.   
  
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), formula = y~x, se= FALSE)



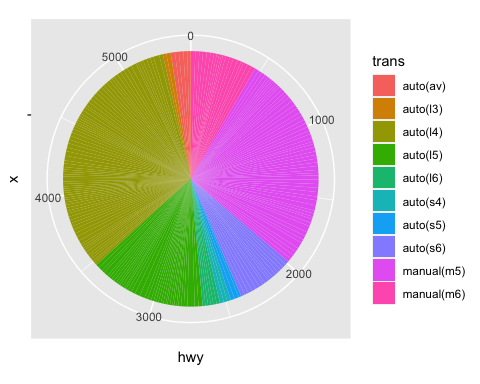
#8. Using the mpg data frame, create a bar graph for the variable drv. include code so that each bar has a different color.   
#Title your bar graph DRV Bar Graph  
  
ggplot(data=mpg) +  
 geom\_bar(mapping = aes(x = drv, fill = drv))



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



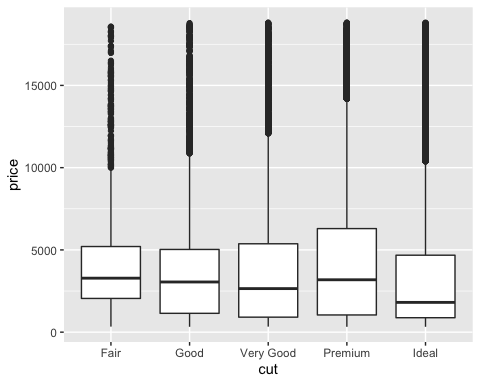
#10. Using the stacked bar graph from number 10, create a pie chart.  
pieaa <- aa + coord\_polar("y", start=0)   
pieaa



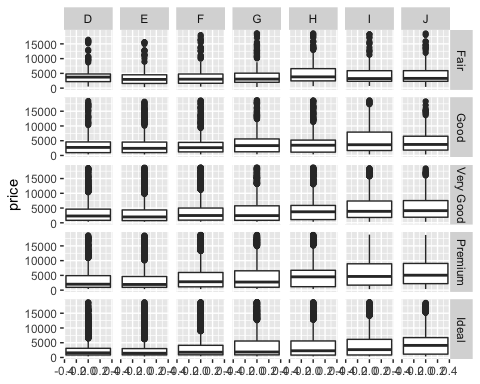
#11. Use and show R code that will produce the following side by side box plots from the diamonds data frame  
diamonds

## # 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

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

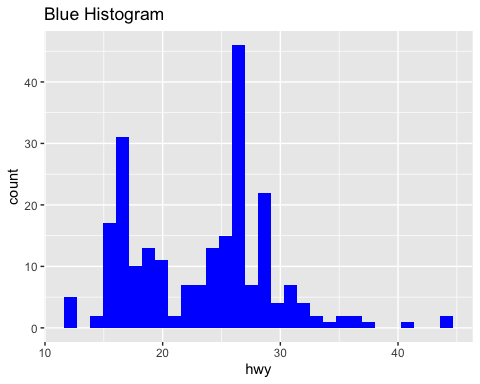


#12. Use and show R code that will produce the following faceted display of boxplots from the diamonds data frame.  
ggplot(data = diamonds) +  
 geom\_boxplot(mapping=aes(y=price)) +  
 facet\_grid(cut~color)



#13. Use and show R code that will produce a histogram that is colored blue for the hwy variable from the mpg data frame.   
#Give the histogram the title Blue Histogram.  
  
ggplot(data = mpg) +  
 geom\_histogram(mapping = aes(x=hwy), fill="blue")+  
 ggtitle("Blue Histogram")

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



#PART II  
  
#1. Using the DPLYR filter function and the mpg data frame, produce a data frame that only has output for a Dodge Durango 4wd.  
DodgeDurango4wd <- mpg%>%  
 filter(manufacturer=="dodge", model=="durango 4wd")  
DodgeDurango4wd

## # A tibble: 7 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 dodge durango … 3.9 1999 6 auto(l… 4 13 17 r suv   
## 2 dodge durango … 4.7 2008 8 auto(l… 4 13 17 r suv   
## 3 dodge durango … 4.7 2008 8 auto(l… 4 9 12 e suv   
## 4 dodge durango … 4.7 2008 8 auto(l… 4 13 17 r suv   
## 5 dodge durango … 5.2 1999 8 auto(l… 4 11 16 r suv   
## 6 dodge durango … 5.7 2008 8 auto(l… 4 13 18 r suv   
## 7 dodge durango … 5.9 1999 8 auto(l… 4 11 15 r suv

#2. Using the DPLYR filter function and the mpg data frame, produce a data frame that only has output for  
#vehicles whose city mileage is less than 10 miles per gallon and whose highway mileage is less than 16 miles per gallon.  
mpg%>%  
 filter(cty<10, hwy<16)

## # A tibble: 5 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 dodge dakota pi… 4.7 2008 8 auto(… 4 9 12 e pick…  
## 2 dodge durango 4… 4.7 2008 8 auto(… 4 9 12 e suv   
## 3 dodge ram 1500 … 4.7 2008 8 auto(… 4 9 12 e pick…  
## 4 dodge ram 1500 … 4.7 2008 8 manua… 4 9 12 e pick…  
## 5 jeep grand che… 4.7 2008 8 auto(… 4 9 12 e suv

#3. Using the DPLYR arrange function and the mpg data frame, produce a data frame that displays displ in descending order  
mpg%>%  
 arrange(desc(displ))

## # 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 chevrolet corvette 7 2008 8 manua… r 15 24 p 2sea…  
## 2 chevrolet k1500 ta… 6.5 1999 8 auto(… 4 14 17 d suv   
## 3 chevrolet corvette 6.2 2008 8 manua… r 16 26 p 2sea…  
## 4 chevrolet corvette 6.2 2008 8 auto(… r 15 25 p 2sea…  
## 5 jeep grand ch… 6.1 2008 8 auto(… 4 11 14 p suv   
## 6 chevrolet c1500 su… 6 2008 8 auto(… r 12 17 r suv   
## 7 dodge durango … 5.9 1999 8 auto(… 4 11 15 r suv   
## 8 dodge ram 1500… 5.9 1999 8 auto(… 4 11 15 r pick…  
## 9 chevrolet c1500 su… 5.7 1999 8 auto(… r 13 17 r suv   
## 10 chevrolet corvette 5.7 1999 8 manua… r 16 26 p 2sea…  
## # … with 224 more rows

#4. Using the DPLYR arrange function and the mpg data frame, produce a data frame of 30 observations that display  
#city miles per gallon in ascending order. Which vehicle has the lowest city miles per gallon ?  
mpg%>%  
 arrange(cty)%>%  
 print(n=30)

## # 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 dodge dakota p… 4.7 2008 8 auto(… 4 9 12 e pick…  
## 2 dodge durango … 4.7 2008 8 auto(… 4 9 12 e suv   
## 3 dodge ram 1500… 4.7 2008 8 auto(… 4 9 12 e pick…  
## 4 dodge ram 1500… 4.7 2008 8 manua… 4 9 12 e pick…  
## 5 jeep grand ch… 4.7 2008 8 auto(… 4 9 12 e suv   
## 6 chevrolet c1500 su… 5.3 2008 8 auto(… r 11 15 e suv   
## 7 chevrolet k1500 ta… 5.3 2008 8 auto(… 4 11 14 e suv   
## 8 chevrolet k1500 ta… 5.7 1999 8 auto(… 4 11 15 r suv   
## 9 dodge caravan … 3.3 2008 6 auto(… f 11 17 e mini…  
## 10 dodge dakota p… 5.2 1999 8 manua… 4 11 17 r pick…  
## 11 dodge dakota p… 5.2 1999 8 auto(… 4 11 15 r pick…  
## 12 dodge durango … 5.2 1999 8 auto(… 4 11 16 r suv   
## 13 dodge durango … 5.9 1999 8 auto(… 4 11 15 r suv   
## 14 dodge ram 1500… 5.2 1999 8 auto(… 4 11 15 r pick…  
## 15 dodge ram 1500… 5.2 1999 8 manua… 4 11 16 r pick…  
## 16 dodge ram 1500… 5.9 1999 8 auto(… 4 11 15 r pick…  
## 17 ford expediti… 4.6 1999 8 auto(… r 11 17 r suv   
## 18 ford expediti… 5.4 1999 8 auto(… r 11 17 r suv   
## 19 ford f150 pic… 5.4 1999 8 auto(… 4 11 15 r pick…  
## 20 jeep grand ch… 6.1 2008 8 auto(… 4 11 14 p suv   
## 21 land rover range ro… 4 1999 8 auto(… 4 11 15 p suv   
## 22 land rover range ro… 4.6 1999 8 auto(… 4 11 15 p suv   
## 23 lincoln navigato… 5.4 1999 8 auto(… r 11 17 r suv   
## 24 lincoln navigato… 5.4 1999 8 auto(… r 11 16 p suv   
## 25 toyota land cru… 4.7 1999 8 auto(… 4 11 15 r suv   
## 26 chevrolet c1500 su… 6 2008 8 auto(… r 12 17 r suv   
## 27 dodge ram 1500… 4.7 2008 8 manua… 4 12 16 r pick…  
## 28 dodge ram 1500… 4.7 2008 8 manua… 4 12 16 r pick…  
## 29 ford expediti… 5.4 2008 8 auto(… r 12 18 r suv   
## 30 land rover range ro… 4.2 2008 8 auto(… 4 12 18 r suv   
## # … with 204 more rows

#lowest city mpg: dodge dakota pickup, dodge durango, dodge ram 1500 pickup, dodge ram 1500 pickup, jeep grand cherokee   
  
  
#5.Using the DPLYR filter and select functions and the mpg data frame, produce a data frame that displays all   
#ford vehicles for 1999 whose city miles per gallon is less than 16 and whose highway miles per gallon is also less than 16.  
mpg%>%  
 filter(manufacturer=="ford", year==1999, cty<16, hwy<16)%>%  
 select(manufacturer, year, cty, hwy)

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

#6. Using the DPLYR filter, select and mutate functions and the mpg data frame, produce a data frame that displays   
#the difference between highway mileage and city mileage for the ford mustang.  
mpg%>%  
 filter(manufacturer=="ford", model=="mustang")%>%  
 mutate(difference = hwy-cty)%>%  
 select(manufacturer, model, cty, hwy, difference)

## # 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)  
flights

## # A tibble: 336,776 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int> <int>  
## 1 2013 1 1 517 515 2 830 819  
## 2 2013 1 1 533 529 4 850 830  
## 3 2013 1 1 542 540 2 923 850  
## 4 2013 1 1 544 545 -1 1004 1022  
## 5 2013 1 1 554 600 -6 812 837  
## 6 2013 1 1 554 558 -4 740 728  
## 7 2013 1 1 555 600 -5 913 854  
## 8 2013 1 1 557 600 -3 709 723  
## 9 2013 1 1 557 600 -3 838 846  
## 10 2013 1 1 558 600 -2 753 745  
## # … with 336,766 more rows, and 11 more variables: arr\_delay <dbl>,  
## # carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,  
## # air\_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time\_hour <dttm>

#?flights  
#View(flights)  
  
#Use DPLYR functions and the piping operator to produce a data frame that shows   
#arrival delay times in descending order for American Airlines on March 17, 2013.   
flights%>%  
 arrange(desc(arr\_delay))%>%  
 filter(carrier=="AA", month==3, day==17, year==2013)

## # A tibble: 88 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int> <int>  
## 1 2013 3 17 1724 1550 94 2057 1950  
## 2 2013 3 17 1635 1545 50 1949 1910  
## 3 2013 3 17 1730 1700 30 2029 1950  
## 4 2013 3 17 1808 1745 23 2156 2120  
## 5 2013 3 17 1734 1710 24 2048 2015  
## 6 2013 3 17 1037 1030 7 1417 1355  
## 7 2013 3 17 1529 1530 -1 1932 1910  
## 8 2013 3 17 1629 1610 19 1821 1800  
## 9 2013 3 17 853 900 -7 1239 1220  
## 10 2013 3 17 1901 1905 -4 2244 2225  
## # … with 78 more rows, and 11 more variables: arr\_delay <dbl>, carrier <chr>,  
## # flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>,  
## # distance <dbl>, hour <dbl>, minute <dbl>, time\_hour <dttm>

#8. Using the mpg data frame , dplyr functions and the pipe operator, produce a data frame that displays   
#the mean mpg for city driving for manufacturers in the year 1999 only in descending order.   
#Which manufacturer got the best average gas mileage in 1999?  
mpg%>%  
 filter(year==1999)%>%  
 group\_by(manufacturer)%>%   
 summarise(AverageCityMileage = mean(cty))%>%   
 arrange(desc(AverageCityMileage))

## `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

#Honda got the best average city gas mileage in for cars made in 1999