Workshop.4

Sydney Russo

2025-03-19

Workshop 1

uploading libraries

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

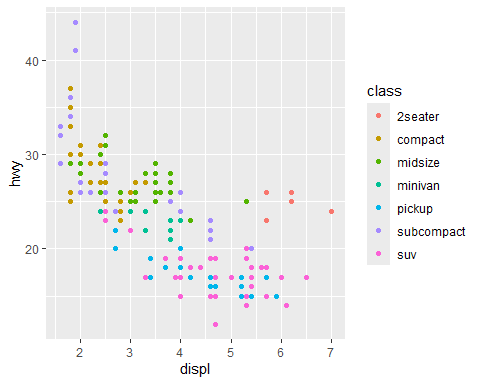
library(dplyr)  
library(ggplot2)  
library(RColorBrewer)  
library(viridis)

## Loading required package: viridisLite

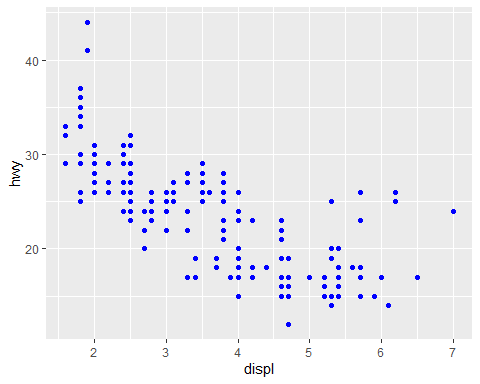
library(hexbin)  
#install.packages('viridis')

Creating the first plot

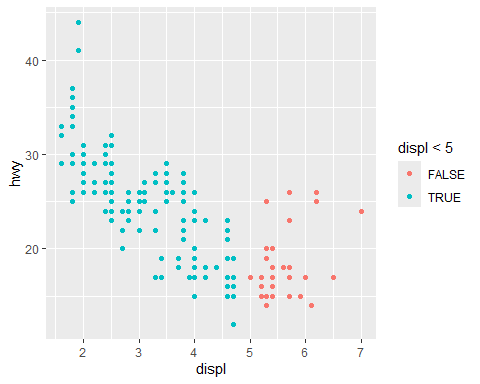
ggplot(data = mpg) +   
 geom\_point(mapping = aes(x = displ, y = hwy, colour=class))



#this graph shows a negative relationship between engine size and fuel efficiency  
  
#changing the appearance of the graph  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = hwy), color = "blue")

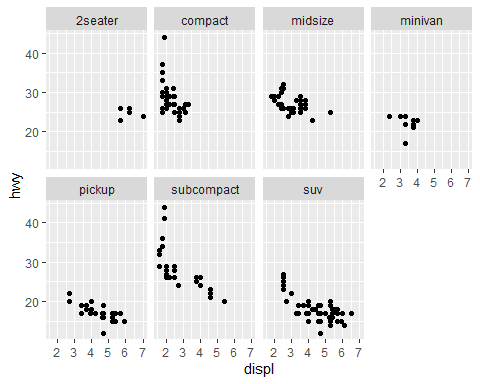


#what happens if you map an aesthetic to something other than a variable name  
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x = displ, y = hwy, colour = displ < 5))



Using Facet Wraps

ggplot(data=mpg) +   
 geom\_point(mapping =aes(x = displ, y = hwy)) +  
 facet\_wrap (~class, nrow = 2)

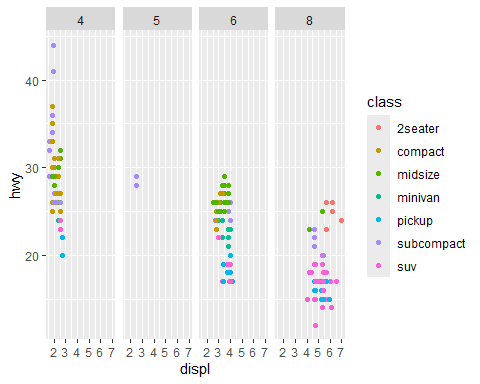


Using Facet Wrap Grids

#this allows the facet to split by more than one variable  
  
ggplot(data = mpg) +   
 geom\_point(mapping = aes(x=displ, y=hwy, colour = class))+  
 facet\_grid(drv ~ cyl)



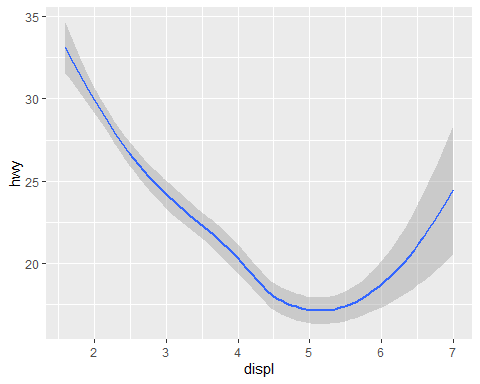
ggplot(data = mpg) +   
 geom\_point(mapping = aes(x=displ, y=hwy, colour = class))+  
 facet\_grid(.~ cyl)

 Exercise #what does nrow do? what does ncol do? what other options control the layout of the individual panels? #nrow and ncol are the number or rows and columns that are present #there are scales, shrinking (will shrink scales to fit output of statistics not raw data if true, if false it will be a range of raw data before statistical summary), switch function allows for the X and Y to switch to top, bottom, right or left. there are many more aswell.

Fitting lines

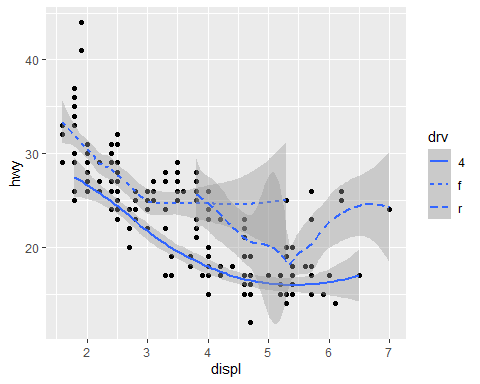
ggplot(data = mpg) +  
 geom\_smooth(mapping = aes (x=displ, y=hwy))

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



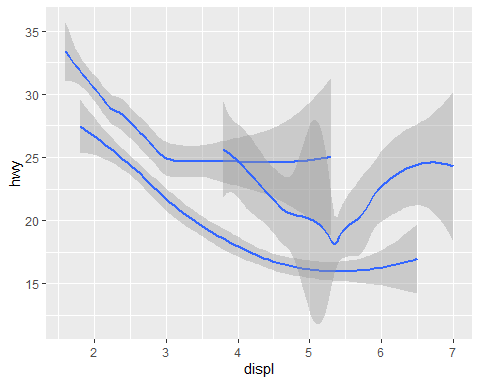
ggplot(data = mpg) +  
 geom\_point(mapping = aes(x=displ, y=hwy))+  
 geom\_smooth(mapping = aes(x=displ, y=hwy, linetype = drv))

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



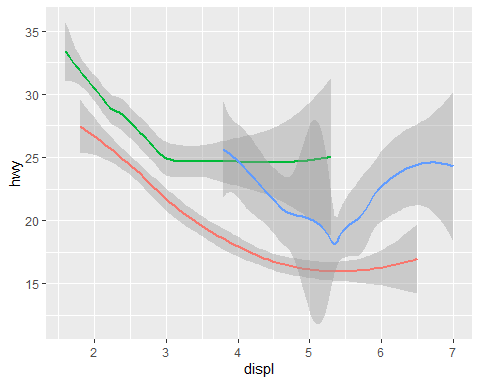
#setting the group aesthetic  
ggplot(data = mpg)+  
 geom\_smooth(mapping=aes(x=displ, y=hwy, group=drv))

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



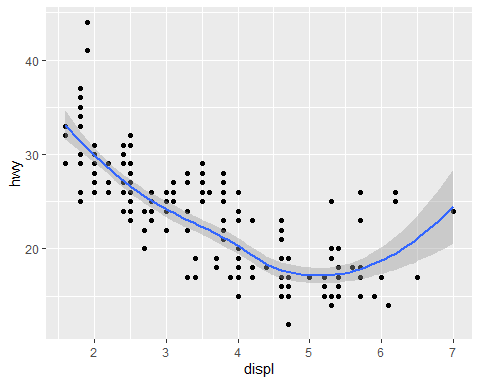
#changing the colour  
ggplot(data=mpg)+  
 geom\_smooth(  
 mapping = aes(x = displ, y = hwy, color = drv),  
 show.legend=FALSE)

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



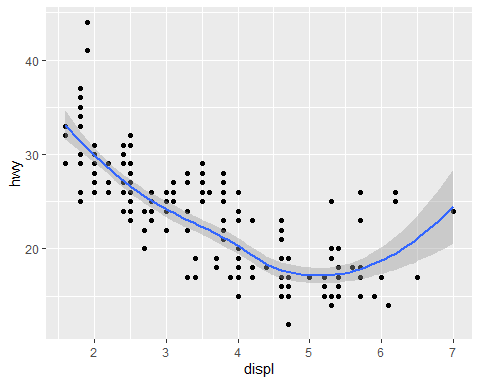
#mapping multiple geoms on one plot  
ggplot(data = mpg)+  
 geom\_point(mapping = aes(x = displ, y = hwy))+  
 geom\_smooth(mapping = aes(x = displ, y = hwy))

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



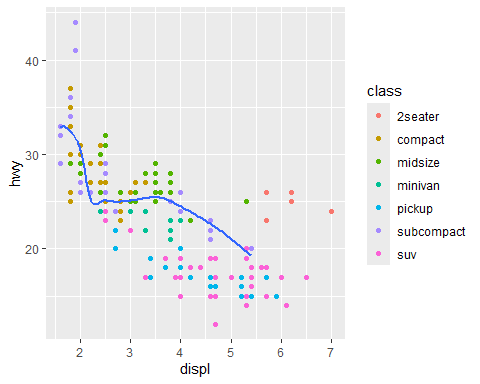
#an easier way to do above if I wanted to change anything  
ggplot(data = mpg, mapping = aes(x = displ, y = hwy))+  
 geom\_point()+  
 geom\_smooth()

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



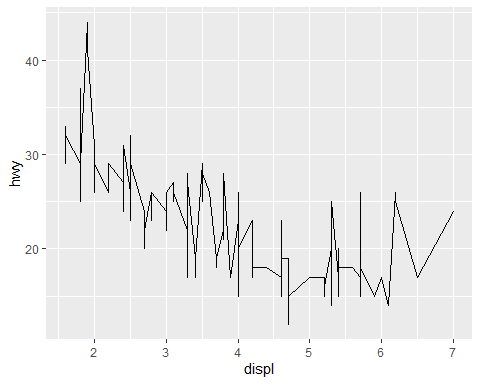
#adding colors  
ggplot(data = mpg, mapping = aes(x = displ, y = hwy))+  
 geom\_point(mapping = aes(color=class))+  
 geom\_smooth(data = filter (mpg, class =="subcompact"), se =FALSE)

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

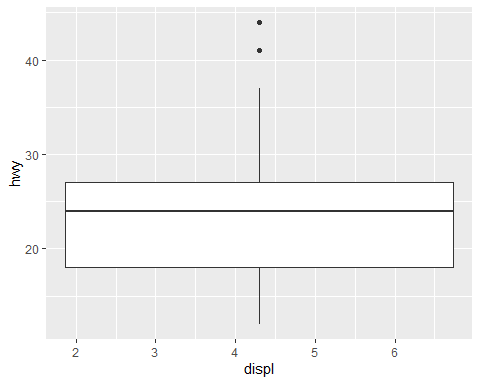


Exercise #1. line: geom\_line , Boxplot:geom\_boxplot , Histogram: geom\_histogram, an area chart: geom\_area #2. I predict that the graphs will all look different becuase they are asking for the data to be laid out in different forms.

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



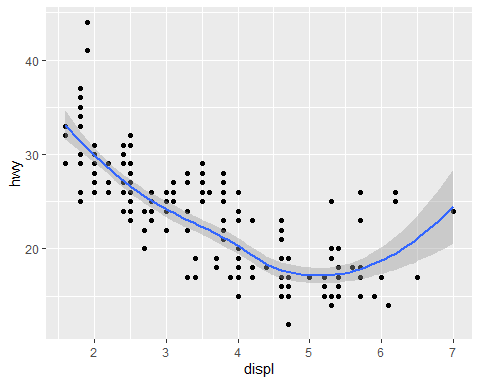
ggplot(data = mpg, mapping = aes(x=displ, y=hwy))+  
 geom\_boxplot()



#3. I would assume that these two graphs are saying the same thing just in 2 different ways. and yes they are the same.

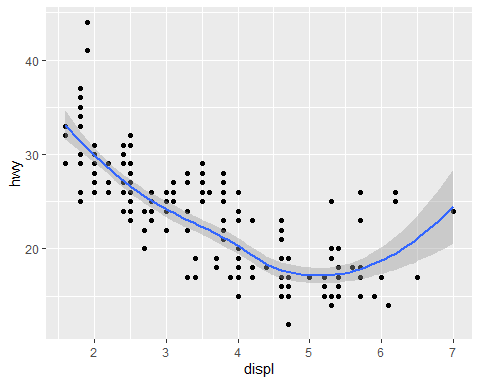
ggplot(data = mpg, mapping = aes(x = displ, y = hwy))+  
 geom\_point()+  
 geom\_smooth()

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



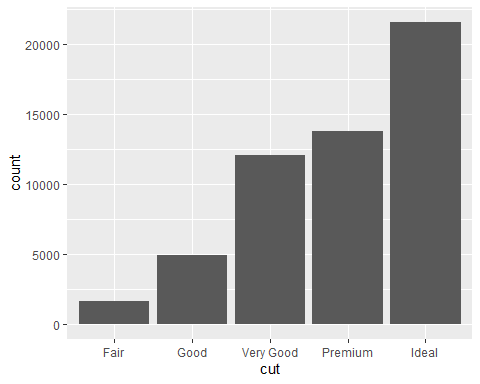
ggplot(data = mpg)+  
 geom\_point(mapping = aes(x = displ, y = hwy))+  
 geom\_smooth(mapping = aes(x = displ, y = hwy))

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



Plotting Statistics

ggplot(data = diamonds) +  
 geom\_bar(mapping = aes(x =cut))



Overriding defaults in ggplot

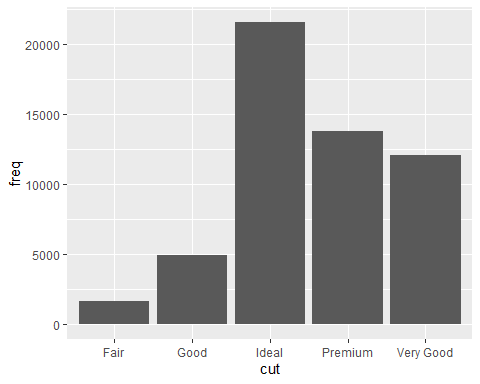
demo <- tribble(  
 ~cut, ~freq,  
 "Fair", 1610,  
 "Good", 4906,  
 "Very Good", 12082,  
 "Premium", 13791,  
 "Ideal", 21551  
)  
demo

## # A tibble: 5 × 2  
## cut freq  
## <chr> <dbl>  
## 1 Fair 1610  
## 2 Good 4906  
## 3 Very Good 12082  
## 4 Premium 13791  
## 5 Ideal 21551

diamonds

## # A tibble: 53,940 × 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 0.29 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  
## # ℹ 53,930 more rows

ggplot(data = demo) +  
 geom\_bar(mapping = aes(x = cut, y = freq), stat = "identity")

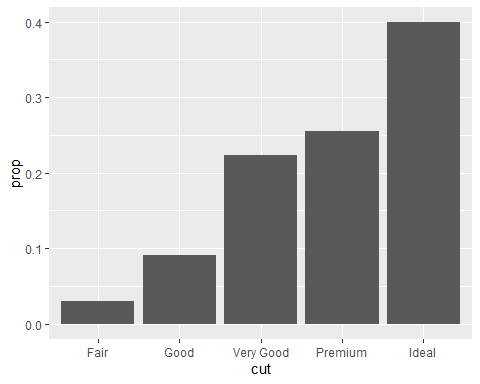


diamonds

## # A tibble: 53,940 × 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 0.29 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  
## # ℹ 53,930 more rows

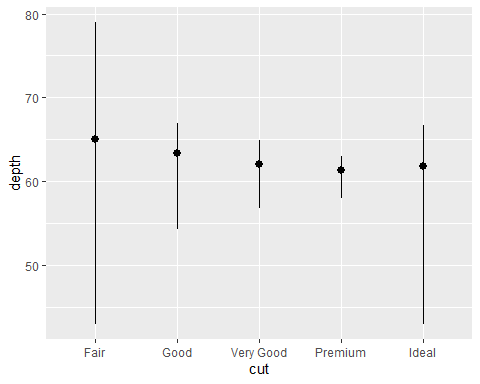
ggplot(data=diamonds)+  
 geom\_bar(mapping = aes(x=cut, y = stat(prop), group = 1))

## Warning: `stat(prop)` was deprecated in ggplot2 3.4.0.  
## ℹ Please use `after\_stat(prop)` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

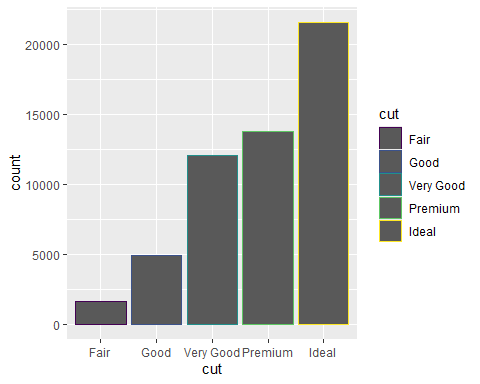


Plotting Statistical details

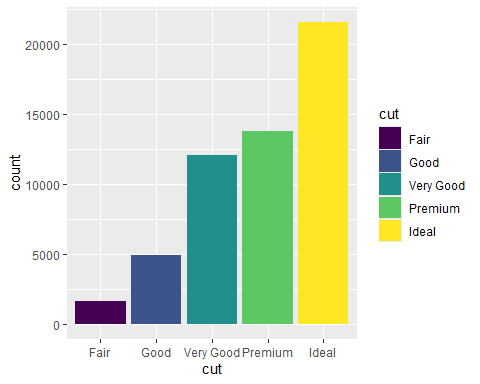
ggplot(data=diamonds)+  
 stat\_summary(  
 mapping =aes(x = cut, y = depth),  
 fun.min = min,  
 fun.max = max,  
 fun = median  
 )



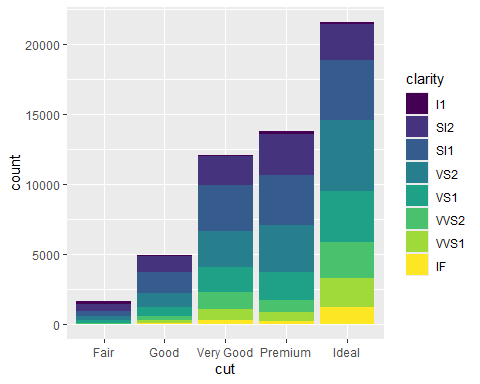
#making it pretty  
ggplot(data = diamonds) +  
 geom\_bar(mapping = aes(x = cut, colour = cut))



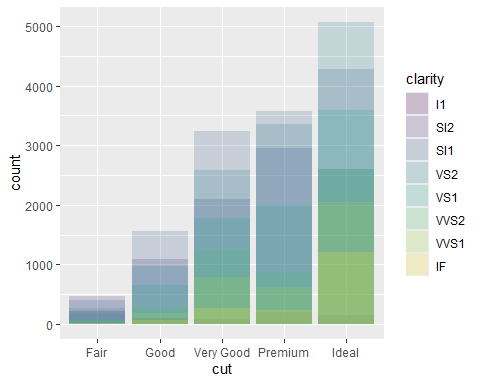
ggplot(data = diamonds)+  
 geom\_bar(mapping = aes (x = cut, fill = cut))



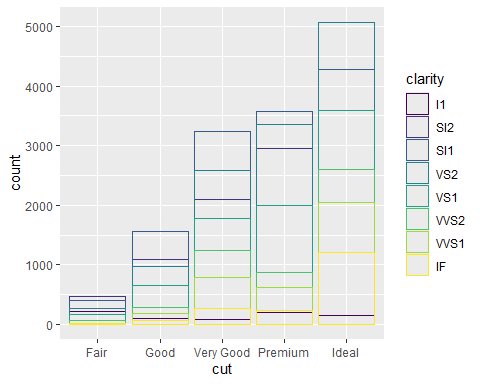
ggplot(data=diamonds)+  
 geom\_bar(mapping = aes(x=cut, fill = clarity))



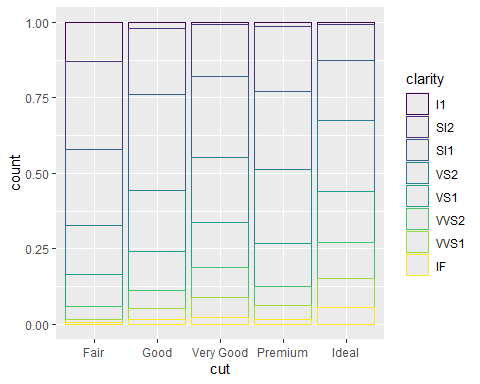
#altering transparency  
ggplot(data=diamonds, mapping = aes(x=cut, fill=clarity))+  
 geom\_bar(alpha=1/5, position = "identity")



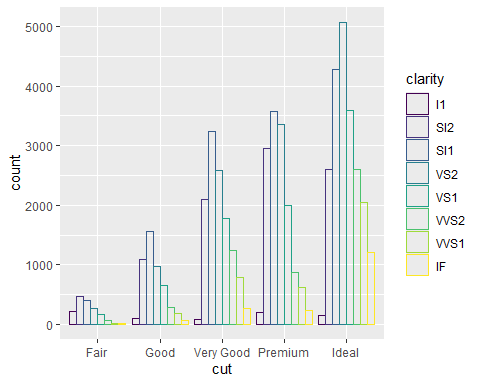
#to color the bar outlines with no fill color  
ggplot(data=diamonds, mapping = aes(x=cut, color=clarity))+  
 geom\_bar(fill=NA, position = "identity")



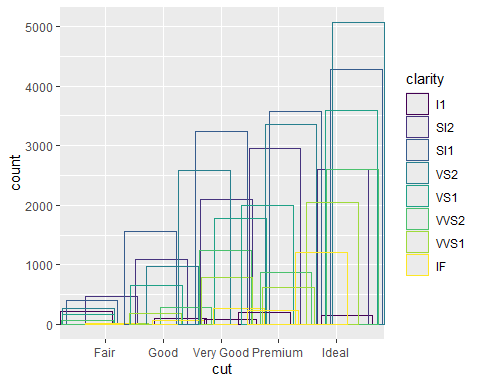
ggplot(data=diamonds, mapping = aes(x=cut, color=clarity))+  
 geom\_bar(fill=NA, position = "fill")



ggplot(data=diamonds, mapping = aes(x=cut, color=clarity))+  
 geom\_bar(fill=NA, position = "dodge")



ggplot(data=diamonds, mapping = aes(x=cut, color=clarity))+  
 geom\_bar(fill=NA, position = "jitter")

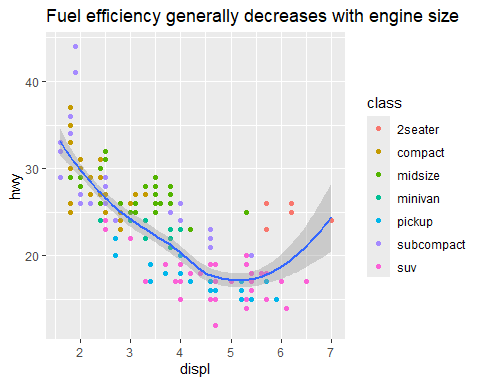


WORKSHOP 2

Labels

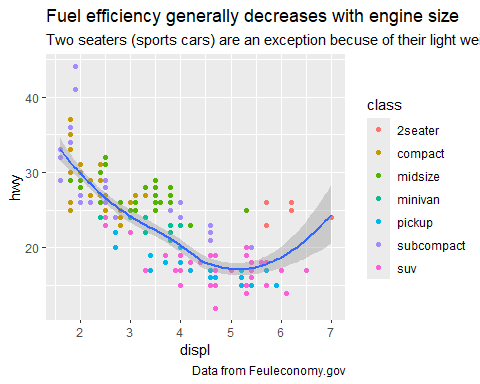
ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color=class))+  
 geom\_smooth(se.e = FALSE)+  
 labs(title = "Fuel efficiency generally decreases with engine size")

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



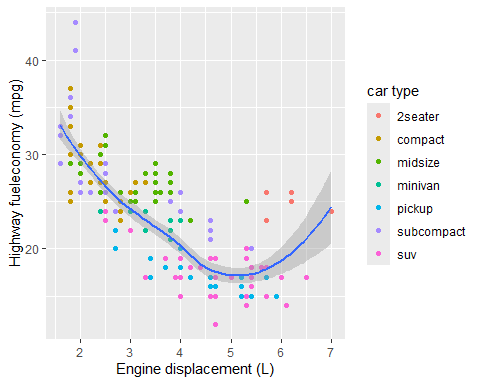
#subtitle adds additional detail in a smaller font beneath the title and caption adds text at the bottom right of the plot  
#caption adds text at the bottom right of the plot, often used to describe the source of the data  
  
ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color=class))+  
 geom\_smooth(se.e = FALSE)+  
 labs(title = "Fuel efficiency generally decreases with engine size",  
 subtitle = "Two seaters (sports cars) are an exception becuse of their light weight",  
 caption = "Data from Feuleconomy.gov")

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



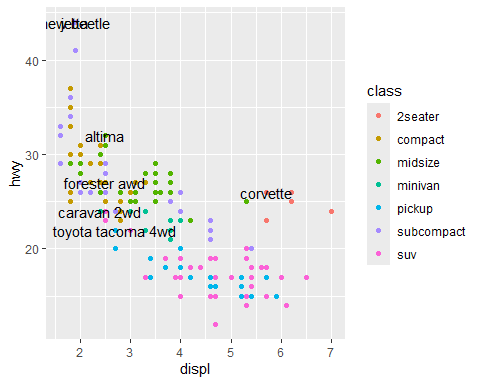
#you can also use labs to change the labels on the axis and legend titles   
ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color=class))+  
 geom\_smooth(se.e = FALSE)+  
 labs(  
 x = "Engine displacement (L)",  
 y = "Highway fueleconomy (mpg)",  
 colour = "car type"  
 )

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



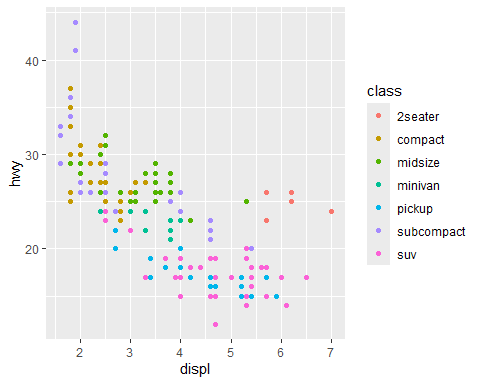
Annotations

best\_in\_class <- mpg |>  
 group\_by(class) |>  
 filter(row\_number(desc(hwy))==1)  
  
ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color=class))+  
 geom\_text(aes(label = model), data = best\_in\_class)



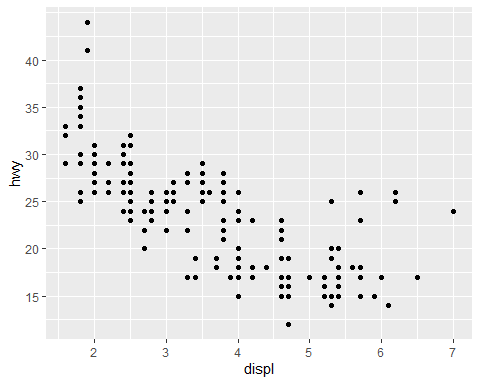
Scales

ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color = class))+  
 scale\_x\_continuous()+  
 scale\_y\_continuous()+  
 scale\_color\_discrete()

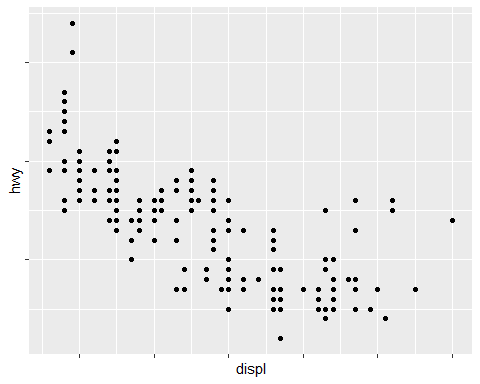


Axis Ticks

ggplot(mpg, aes(displ, hwy))+  
 geom\_point()+  
 scale\_y\_continuous(breaks = seq(15,40, by = 5))

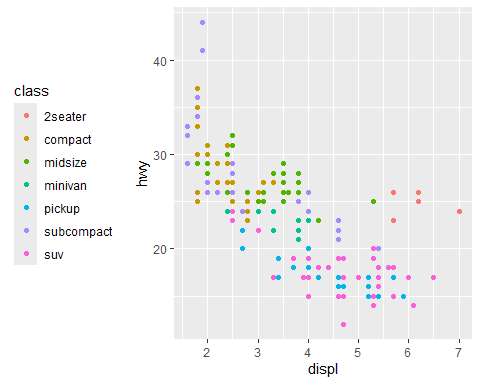


ggplot(mpg, aes(displ, hwy))+  
 geom\_point()+  
 scale\_x\_continuous(labels = NULL)+  
 scale\_y\_continuous(labels = NULL)

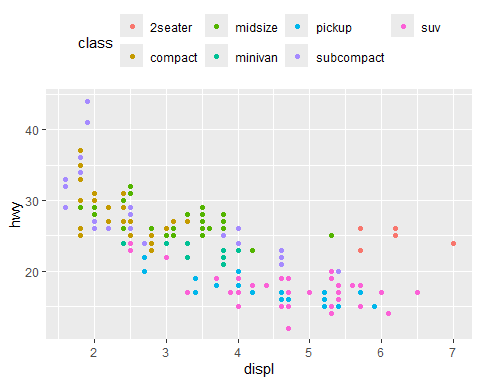


Legends and color schemes

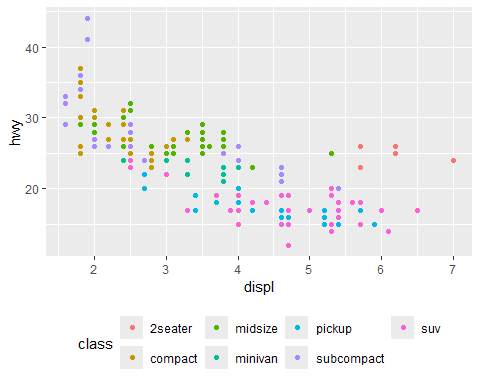
base <- ggplot (mpg, aes(displ, hwy))+  
 geom\_point(aes(color=class))  
base + theme(legend.position ="left")



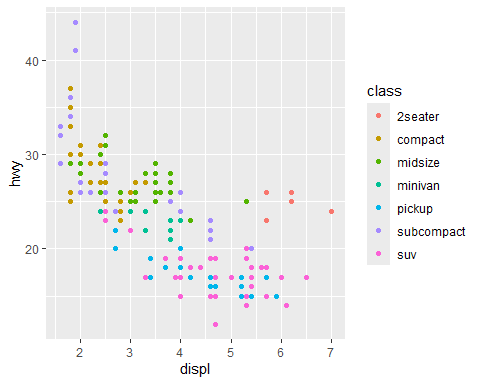
base+theme(legend.position = "top")



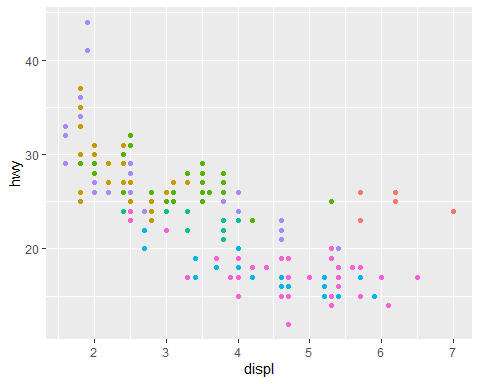
base+theme(legend.position = "bottom")



base+theme(legend.position = "right") #this is the default legend position

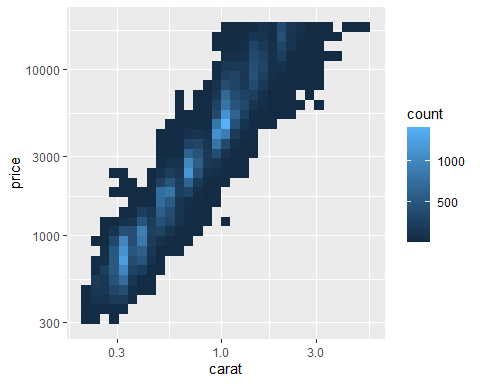


base+theme(legend.position = "NONE")

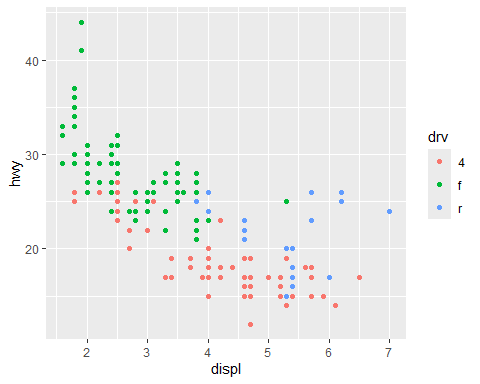


Replacing a scale

ggplot (diamonds, aes(carat, price)) +  
 geom\_bin2d()+  
 scale\_x\_log10()+  
 scale\_y\_log10()

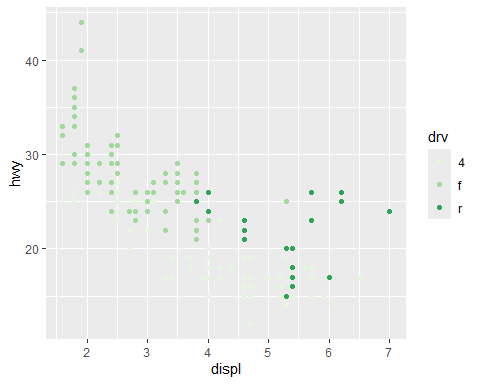


ggplot(mpg,aes(displ,hwy))+  
 geom\_point(aes(color = drv))



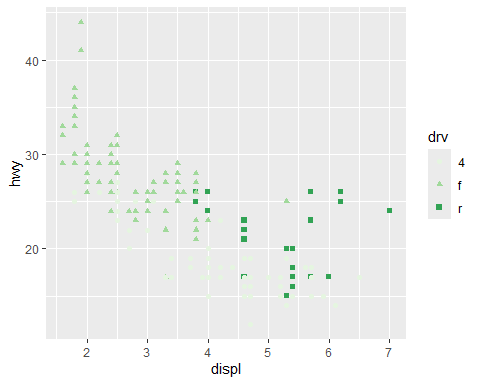
ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color=drv))+  
 scale\_color\_brewer(palette = "set1")

## Warning: Unknown palette: "set1"

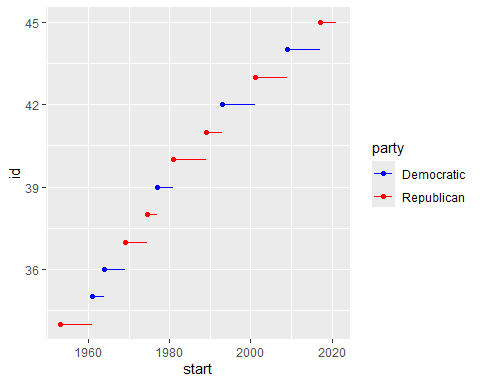


ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color=drv, shape = drv))+  
 scale\_color\_brewer(palette = "set1")

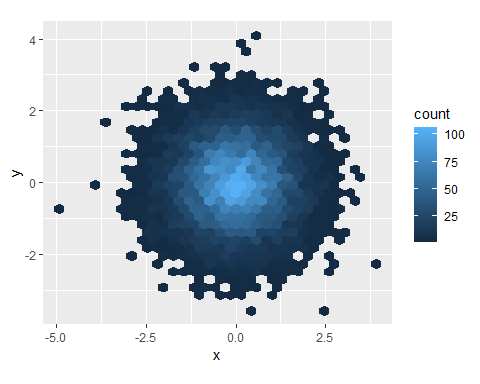
## Warning: Unknown palette: "set1"



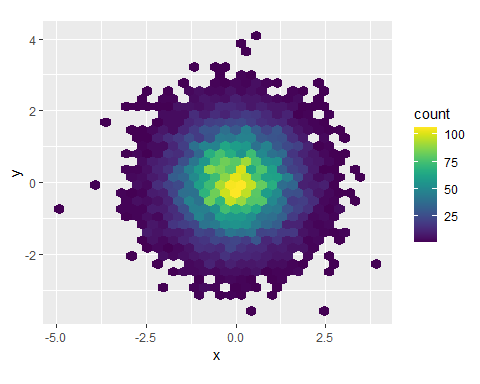
presidential|>  
 mutate(id =33 + row\_number()) |>  
 ggplot(aes(start, id, color=party))+  
 geom\_point()+  
 geom\_segment(aes(xend = end, yend = id))+  
 scale\_color\_manual(values=c(Republican = "red", Democratic = "blue"))



df <- tibble( #not we are just making a fake dataset so we can plot it  
 x= rnorm(10000),  
 y = rnorm(10000)  
)  
ggplot(df, aes(x, y)) +  
 geom\_hex() +   
 coord\_fixed()

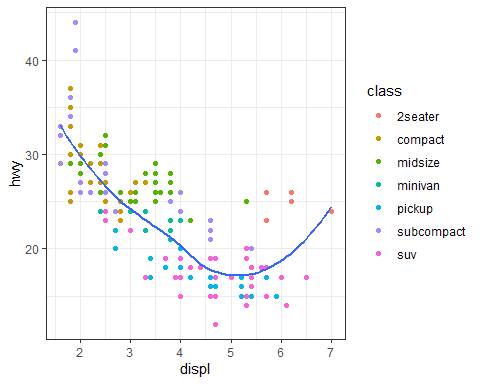


ggplot(df, aes(x,y))+  
 geom\_hex()+  
 viridis::scale\_fill\_viridis()+  
 coord\_fixed()

 Themes

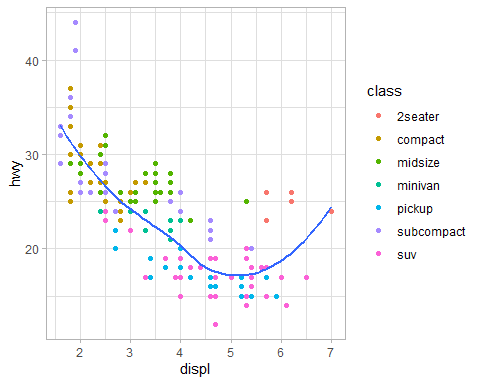
ggplot(mpg, aes(displ, hwy))+  
 geom\_point(aes(color=class))+  
 geom\_smooth(se = FALSE)+  
 theme\_bw()

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



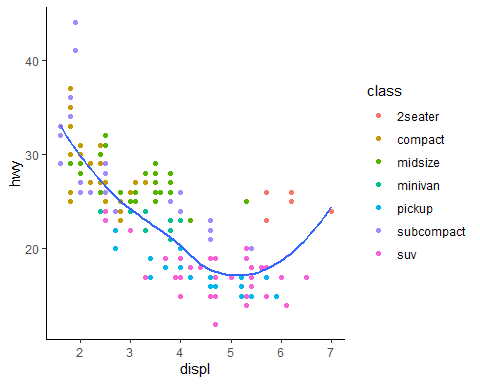
ggplot(mpg,aes(displ,hwy))+  
 geom\_point(aes(color=class))+  
 geom\_smooth(se = FALSE)+  
 theme\_light()

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



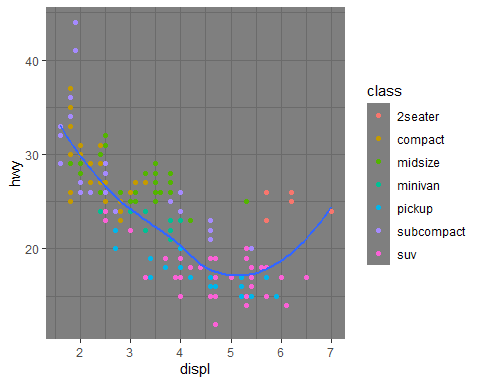
ggplot(mpg,aes(displ,hwy))+  
 geom\_point(aes(color=class))+  
 geom\_smooth(se = FALSE)+  
 theme\_classic()

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



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

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

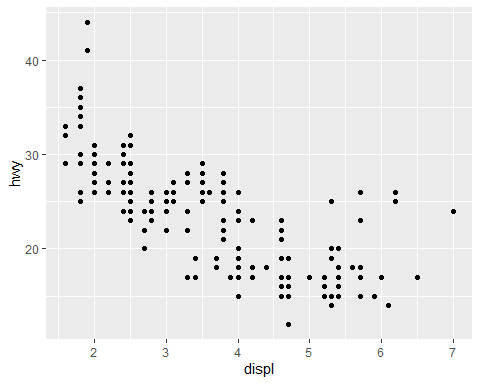


theme(panel.border = element\_blank(),  
 panel.grid.minor.x = element\_blank(),  
 panel.grid.minor.y = element\_blank(),  
 legend.position = "bottom",  
 legend.title = element\_blank(),  
 legend.text = element\_text(size=8),  
 panel.grid.major = element\_blank(),  
 axis.text.y=element\_text(color="black"),  
 axis.text.x=element\_text(color="black"),  
 text=element\_text(family="Arial"))

## List of 10  
## $ text :List of 11  
## ..$ family : chr "Arial"  
## ..$ face : NULL  
## ..$ colour : NULL  
## ..$ size : NULL  
## ..$ hjust : NULL  
## ..$ vjust : NULL  
## ..$ angle : NULL  
## ..$ lineheight : NULL  
## ..$ margin : NULL  
## ..$ debug : NULL  
## ..$ inherit.blank: logi FALSE  
## ..- attr(\*, "class")= chr [1:2] "element\_text" "element"  
## $ axis.text.x :List of 11  
## ..$ family : NULL  
## ..$ face : NULL  
## ..$ colour : chr "black"  
## ..$ size : NULL  
## ..$ hjust : NULL  
## ..$ vjust : NULL  
## ..$ angle : NULL  
## ..$ lineheight : NULL  
## ..$ margin : NULL  
## ..$ debug : NULL  
## ..$ inherit.blank: logi FALSE  
## ..- attr(\*, "class")= chr [1:2] "element\_text" "element"  
## $ axis.text.y :List of 11  
## ..$ family : NULL  
## ..$ face : NULL  
## ..$ colour : chr "black"  
## ..$ size : NULL  
## ..$ hjust : NULL  
## ..$ vjust : NULL  
## ..$ angle : NULL  
## ..$ lineheight : NULL  
## ..$ margin : NULL  
## ..$ debug : NULL  
## ..$ inherit.blank: logi FALSE  
## ..- attr(\*, "class")= chr [1:2] "element\_text" "element"  
## $ legend.text :List of 11  
## ..$ family : NULL  
## ..$ face : NULL  
## ..$ colour : NULL  
## ..$ size : num 8  
## ..$ hjust : NULL  
## ..$ vjust : NULL  
## ..$ angle : NULL  
## ..$ lineheight : NULL  
## ..$ margin : NULL  
## ..$ debug : NULL  
## ..$ inherit.blank: logi FALSE  
## ..- attr(\*, "class")= chr [1:2] "element\_text" "element"  
## $ legend.title : list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## $ legend.position : chr "bottom"  
## $ panel.border : list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## $ panel.grid.major : list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## $ panel.grid.minor.x: list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## $ panel.grid.minor.y: list()  
## ..- attr(\*, "class")= chr [1:2] "element\_blank" "element"  
## - attr(\*, "class")= chr [1:2] "theme" "gg"  
## - attr(\*, "complete")= logi FALSE  
## - attr(\*, "validate")= logi TRUE

Saving and Exporting your Plots

ggplot(mpg, aes(displ, hwy)) + geom\_point()



ggsave("my-plot.pdf")

## Saving 5 x 4 in image

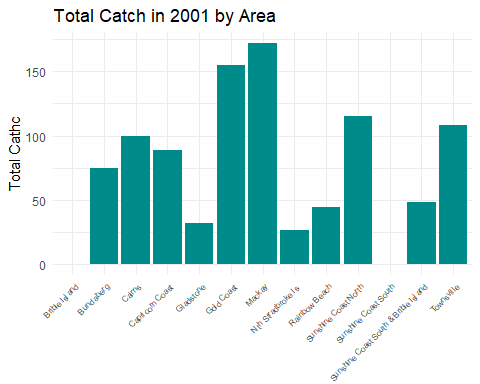
Workshop 2 Assignment

Qfish <-read.csv("../data/QFish\_Data.csv")  
print(Qfish)

## Area X2001.Total  
## 1 Bribie Island 0  
## 2 Bundaberg 75  
## 3 Cairns 100  
## 4 Capricorn Coast 89  
## 5 Gladstone 32  
## 6 Gold Coast 155  
## 7 Mackay 172  
## 8 Nth Stradbroke Is. 26  
## 9 Rainbow Beach 44  
## 10 Sunshine Coast North 115  
## 11 Sunshine Coast South 0  
## 12 Sunshine Coast South & Bribie Island 48  
## 13 Townsville 108

Plotting the data, it is mainly cleaned, there are no NAs or missing values

library(ggplot2)  
ggplot(Qfish, aes (x = Area, y = X2001.Total))+  
 geom\_bar(stat = "identity", fill = "darkcyan") +  
 theme\_minimal()+  
 labs(title = "Total Catch in 2001 by Area",  
 x = NULL,  
 y = "Total Cathc") +  
 theme(axis.text.x =element\_text(angle = 45, hjust = 1, size = 6))



To Import the dataset: My first step for importing the dataset was to export a .csv file from Qfish. Once I have it in my correct file I then used the read.csv and the path to import the file into Rstudio.To make what I wanted easier to work with I created a new .csv file with only total 2001 data and the area. Cleaning the data: I fixed the columns with the proper headers and created a new dataset with the data I needed Creating my plot: I decided to make a bar graph as that was the best way to show the total number of species and animals caught in the year 2001. I created the boxplot and played around with color, phone sizes and axis labels.

Workshop 3

library(tidyverse)  
table1

## # A tibble: 6 × 4  
## country year cases population  
## <chr> <dbl> <dbl> <dbl>  
## 1 Afghanistan 1999 745 19987071  
## 2 Afghanistan 2000 2666 20595360  
## 3 Brazil 1999 37737 172006362  
## 4 Brazil 2000 80488 174504898  
## 5 China 1999 212258 1272915272  
## 6 China 2000 213766 1280428583

library(ggplot2)  
ggplot(table1, aes(year, cases))+  
 geom\_line(aes(group = country), colour = "grey50") +  
 geom\_point(aes(colour = country))

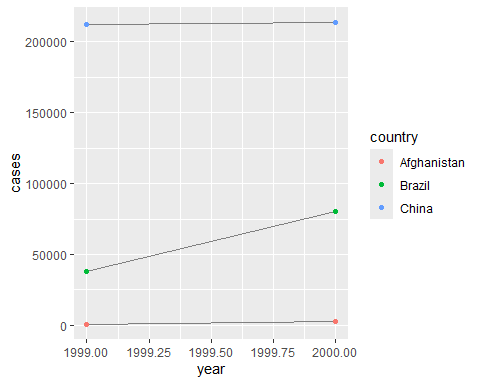


table1

## # A tibble: 6 × 4  
## country year cases population  
## <chr> <dbl> <dbl> <dbl>  
## 1 Afghanistan 1999 745 19987071  
## 2 Afghanistan 2000 2666 20595360  
## 3 Brazil 1999 37737 172006362  
## 4 Brazil 2000 80488 174504898  
## 5 China 1999 212258 1272915272  
## 6 China 2000 213766 1280428583

table2

## # A tibble: 12 × 4  
## country year type count  
## <chr> <dbl> <chr> <dbl>  
## 1 Afghanistan 1999 cases 745  
## 2 Afghanistan 1999 population 19987071  
## 3 Afghanistan 2000 cases 2666  
## 4 Afghanistan 2000 population 20595360  
## 5 Brazil 1999 cases 37737  
## 6 Brazil 1999 population 172006362  
## 7 Brazil 2000 cases 80488  
## 8 Brazil 2000 population 174504898  
## 9 China 1999 cases 212258  
## 10 China 1999 population 1272915272  
## 11 China 2000 cases 213766  
## 12 China 2000 population 1280428583

table3

## # A tibble: 6 × 3  
## country year rate   
## <chr> <dbl> <chr>   
## 1 Afghanistan 1999 745/19987071   
## 2 Afghanistan 2000 2666/20595360   
## 3 Brazil 1999 37737/172006362   
## 4 Brazil 2000 80488/174504898   
## 5 China 1999 212258/1272915272  
## 6 China 2000 213766/1280428583

1. Table 1 is set up well with country, year cases and population. This shows the population per year per country with the number of cases. table2 shows the count of cases and population in each country separated by year. table3 has the rate in there but as the growth rate per year in each country but it has not been calculated yet.
   1. using table 1 we would use a combination of the pipe function and the pivot longer to call the cases with country and year because there are 2 years for each country. and use the mutate function to put it into a new row in the table
2. and then to do the matching population per country per year, we would do the same as above but instead of case it would be population and then would mutate it so that it created a new row in the chart
3. using those new columns and creating a new column I can then calculate the the case by population and multiply it by 10,000 and use the mutate function to put that back into the dataset.

billboard |>  
 pivot\_longer(  
 cols = starts\_with("wk"),  
 names\_to = "week",  
 values\_to = "rank",  
 values\_drop\_na = TRUE  
 )

## # A tibble: 5,307 × 5  
## artist track date.entered week rank  
## <chr> <chr> <date> <chr> <dbl>  
## 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk1 87  
## 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk2 82  
## 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk3 72  
## 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk4 77  
## 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk5 87  
## 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk6 94  
## 7 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk7 99  
## 8 2Ge+her The Hardest Part Of ... 2000-09-02 wk1 91  
## 9 2Ge+her The Hardest Part Of ... 2000-09-02 wk2 87  
## 10 2Ge+her The Hardest Part Of ... 2000-09-02 wk3 92  
## # ℹ 5,297 more rows

df <- tribble (  
 ~id, ~bp1, ~bp2,  
 "A", 100, 120,  
 "B", 140, 115,  
 "C", 120, 125  
)  
#this is creating out own dataset called df with 3 variables  
  
df|>  
 pivot\_longer(  
 cols = bp1:bp2,  
 names\_to = "measurement",  
 values\_to ="value"  
 )

## # A tibble: 6 × 3  
## id measurement value  
## <chr> <chr> <dbl>  
## 1 A bp1 100  
## 2 A bp2 120  
## 3 B bp1 140  
## 4 B bp2 115  
## 5 C bp1 120  
## 6 C bp2 125

#the above took the values for both bp1 and bp2 into one column and then another column was its matched bp1 or bp2

cms\_patient\_experience |>  
 pivot\_wider(  
 id\_cols = starts\_with("org"),  
 names\_from = measure\_cd,  
 values\_from = prf\_rate  
 )

## # A tibble: 95 × 8  
## org\_pac\_id org\_nm CAHPS\_GRP\_1 CAHPS\_GRP\_2 CAHPS\_GRP\_3 CAHPS\_GRP\_5 CAHPS\_GRP\_8  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0446157747 USC C… 63 87 86 57 85  
## 2 0446162697 ASSOC… 59 85 83 63 88  
## 3 0547164295 BEAVE… 49 NA 75 44 73  
## 4 0749333730 CAPE … 67 84 85 65 82  
## 5 0840104360 ALLIA… 66 87 87 64 87  
## 6 0840109864 REX H… 73 87 84 67 91  
## 7 0840513552 SCL H… 58 83 76 58 78  
## 8 0941545784 GRITM… 46 86 81 54 NA  
## 9 1052612785 COMMU… 65 84 80 58 87  
## 10 1254237779 OUR L… 61 NA NA 65 NA  
## # ℹ 85 more rows  
## # ℹ 1 more variable: CAHPS\_GRP\_12 <dbl>

df <- tribble (  
 ~id, ~measurement, ~value,  
 "A", "bp1", 100,  
 "B", "bp2", 115,  
 "B", "bp1", 140,  
 "A", "bp2", 120,  
 "A", "bp3", 105  
)  
print(df)

## # A tibble: 5 × 3  
## id measurement value  
## <chr> <chr> <dbl>  
## 1 A bp1 100  
## 2 B bp2 115  
## 3 B bp1 140  
## 4 A bp2 120  
## 5 A bp3 105

df |>  
 pivot\_wider(  
 names\_from = measurement,  
 values\_from = value  
 )

## # A tibble: 2 × 4  
## id bp1 bp2 bp3  
## <chr> <dbl> <dbl> <dbl>  
## 1 A 100 120 105  
## 2 B 140 115 NA

df |>  
 distinct (measurement) |>  
 pull()

## [1] "bp1" "bp2" "bp3"

df |>  
 select(-measurement, -value) |>  
 distinct()|>  
 mutate(x = NA, y = NA, z = NA)

## # A tibble: 2 × 4  
## id x y z   
## <chr> <lgl> <lgl> <lgl>  
## 1 A NA NA NA   
## 2 B NA NA NA

Excercise 4.5.5

stocks <- tibble(  
 year = c(2015, 2015, 2016, 2016),  
 half = c(1, 2, 1, 2),  
 return = c(1.88, 0.59, 0.92, 0.17)  
)  
  
stocks %>%  
 pivot\_wider(names\_from = year, values\_from = return) %>%  
 pivot\_longer('2015':'2016',names\_to = "year", values\_to = "return")

## # A tibble: 4 × 3  
## half year return  
## <dbl> <chr> <dbl>  
## 1 1 2015 1.88  
## 2 1 2016 0.92  
## 3 2 2015 0.59  
## 4 2 2016 0.17

1. it is becuase they sort the data differently. One organizes by 1/2 1 na dthe other by 1/2 2
2. the code fails because the years need the quotations around them
3. this tibble would need to be longer. the variables are pregnant, male and female

table3 |>  
 separate(rate, into = c("cases", "population"), convert = TRUE)

## # A tibble: 6 × 4  
## country year cases population  
## <chr> <dbl> <int> <int>  
## 1 Afghanistan 1999 745 19987071  
## 2 Afghanistan 2000 2666 20595360  
## 3 Brazil 1999 37737 172006362  
## 4 Brazil 2000 80488 174504898  
## 5 China 1999 212258 1272915272  
## 6 China 2000 213766 1280428583

table3 |>  
 separate(year, into = c("century", "year"), sep = 2)

## # A tibble: 6 × 4  
## country century year rate   
## <chr> <chr> <chr> <chr>   
## 1 Afghanistan 19 99 745/19987071   
## 2 Afghanistan 20 00 2666/20595360   
## 3 Brazil 19 99 37737/172006362   
## 4 Brazil 20 00 80488/174504898   
## 5 China 19 99 212258/1272915272  
## 6 China 20 00 213766/1280428583

table5|>  
 unite(new,century,year,sep = "")

## # A tibble: 6 × 3  
## country new rate   
## <chr> <chr> <chr>   
## 1 Afghanistan 1999 745/19987071   
## 2 Afghanistan 2000 2666/20595360   
## 3 Brazil 1999 37737/172006362   
## 4 Brazil 2000 80488/174504898   
## 5 China 1999 212258/1272915272  
## 6 China 2000 213766/1280428583

treatment <- tribble(  
 ~person, ~treatment, ~response,  
 "Derrick", 1, 7,   
 NA, 2, 10,  
 NA, 3, NA,  
 "Katherine", 1, 4  
)  
  
#you can use fill in these missing values with tidyr::fill(), it works like select()  
treatment |>  
 fill(everything())

## # A tibble: 4 × 3  
## person treatment response  
## <chr> <dbl> <dbl>  
## 1 Derrick 1 7  
## 2 Derrick 2 10  
## 3 Derrick 3 10  
## 4 Katherine 1 4

#this treatment is sometimes calles "last observation carried forward"

x <- c(1,4,5,7,NA)  
coalesce(x,0)

## [1] 1 4 5 7 0

x <- c(1,4,5,7,-99)  
na\_if(x,-99)

## [1] 1 4 5 7 NA

x<- c (NA, NaN)  
x\*10

## [1] NA NaN

x==1

## [1] NA NA

is.na(x)

## [1] TRUE TRUE

stocks <- tibble(  
 year = c(2020,2020,2020,2020,2021,2021,2021),  
 qrt = c(1,2,3,4,2,3,4),  
 price = c(1.88,0.59,0.35,NA,0.92,0.17,2.66)  
)  
  
stocks |>  
 pivot\_wider(  
 names\_from = qrt,  
 values\_from = price  
 )

## # A tibble: 2 × 5  
## year `1` `2` `3` `4`  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2020 1.88 0.59 0.35 NA   
## 2 2021 NA 0.92 0.17 2.66

students <- read\_csv("https://pos.it/r4ds-students-csv", na=c("N/A", "")) #this brings in a csv file from the internet

## Rows: 6 Columns: 5  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (4): Full Name, favourite.food, mealPlan, AGE  
## dbl (1): Student ID  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

#The N/A is not recognised as an actual NA so we converted it so that it does  
students

## # A tibble: 6 × 5  
## `Student ID` `Full Name` favourite.food mealPlan AGE   
## <dbl> <chr> <chr> <chr> <chr>  
## 1 1 Sunil Huffmann Strawberry yoghurt Lunch only 4   
## 2 2 Barclay Lynn French fries Lunch only 5   
## 3 3 Jayendra Lyne <NA> Breakfast and lunch 7   
## 4 4 Leon Rossini Anchovies Lunch only <NA>   
## 5 5 Chidiegwu Dunkel Pizza Breakfast and lunch five   
## 6 6 Güvenç Attila Ice cream Lunch only 6

students |>  
 rename(  
 student\_id = 'Student ID',  
 full\_name = 'Full Name'  
 )

## # A tibble: 6 × 5  
## student\_id full\_name favourite.food mealPlan AGE   
## <dbl> <chr> <chr> <chr> <chr>  
## 1 1 Sunil Huffmann Strawberry yoghurt Lunch only 4   
## 2 2 Barclay Lynn French fries Lunch only 5   
## 3 3 Jayendra Lyne <NA> Breakfast and lunch 7   
## 4 4 Leon Rossini Anchovies Lunch only <NA>   
## 5 5 Chidiegwu Dunkel Pizza Breakfast and lunch five   
## 6 6 Güvenç Attila Ice cream Lunch only 6

Exercise 4.7.3

1. each of these are creating a file.
2. there are not enough column names
3. there are too many rows for the amount of columns present
4. there is a quotation issue with this one
5. there is an issue with number and character
6. the last one needs a comma not a semicolon

library(tidyverse)  
#install.packages("nycflights13")  
library(nycflights13)  
  
airlines

## # A tibble: 16 × 2  
## carrier name   
## <chr> <chr>   
## 1 9E Endeavor Air Inc.   
## 2 AA American Airlines Inc.   
## 3 AS Alaska Airlines Inc.   
## 4 B6 JetBlue Airways   
## 5 DL Delta Air Lines Inc.   
## 6 EV ExpressJet Airlines Inc.   
## 7 F9 Frontier Airlines Inc.   
## 8 FL AirTran Airways Corporation  
## 9 HA Hawaiian Airlines Inc.   
## 10 MQ Envoy Air   
## 11 OO SkyWest Airlines Inc.   
## 12 UA United Air Lines Inc.   
## 13 US US Airways Inc.   
## 14 VX Virgin America   
## 15 WN Southwest Airlines Co.   
## 16 YV Mesa Airlines Inc.

airports

## # A tibble: 1,458 × 8  
## faa name lat lon alt tz dst tzone   
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr>   
## 1 04G Lansdowne Airport 41.1 -80.6 1044 -5 A America/…  
## 2 06A Moton Field Municipal Airport 32.5 -85.7 264 -6 A America/…  
## 3 06C Schaumburg Regional 42.0 -88.1 801 -6 A America/…  
## 4 06N Randall Airport 41.4 -74.4 523 -5 A America/…  
## 5 09J Jekyll Island Airport 31.1 -81.4 11 -5 A America/…  
## 6 0A9 Elizabethton Municipal Airport 36.4 -82.2 1593 -5 A America/…  
## 7 0G6 Williams County Airport 41.5 -84.5 730 -5 A America/…  
## 8 0G7 Finger Lakes Regional Airport 42.9 -76.8 492 -5 A America/…  
## 9 0P2 Shoestring Aviation Airfield 39.8 -76.6 1000 -5 U America/…  
## 10 0S9 Jefferson County Intl 48.1 -123. 108 -8 A America/…  
## # ℹ 1,448 more rows

planes

## # A tibble: 3,322 × 9  
## tailnum year type manufacturer model engines seats speed engine  
## <chr> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
## 1 N10156 2004 Fixed wing multi… EMBRAER EMB-… 2 55 NA Turbo…  
## 2 N102UW 1998 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## 3 N103US 1999 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## 4 N104UW 1999 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## 5 N10575 2002 Fixed wing multi… EMBRAER EMB-… 2 55 NA Turbo…  
## 6 N105UW 1999 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## 7 N107US 1999 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## 8 N108UW 1999 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## 9 N109UW 1999 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## 10 N110UW 1999 Fixed wing multi… AIRBUS INDU… A320… 2 182 NA Turbo…  
## # ℹ 3,312 more rows

weather

## # A tibble: 26,115 × 15  
## origin year month day hour temp dewp humid wind\_dir wind\_speed  
## <chr> <int> <int> <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 EWR 2013 1 1 1 39.0 26.1 59.4 270 10.4   
## 2 EWR 2013 1 1 2 39.0 27.0 61.6 250 8.06  
## 3 EWR 2013 1 1 3 39.0 28.0 64.4 240 11.5   
## 4 EWR 2013 1 1 4 39.9 28.0 62.2 250 12.7   
## 5 EWR 2013 1 1 5 39.0 28.0 64.4 260 12.7   
## 6 EWR 2013 1 1 6 37.9 28.0 67.2 240 11.5   
## 7 EWR 2013 1 1 7 39.0 28.0 64.4 240 15.0   
## 8 EWR 2013 1 1 8 39.9 28.0 62.2 250 10.4   
## 9 EWR 2013 1 1 9 39.9 28.0 62.2 260 15.0   
## 10 EWR 2013 1 1 10 41 28.0 59.6 260 13.8   
## # ℹ 26,105 more rows  
## # ℹ 5 more variables: wind\_gust <dbl>, precip <dbl>, pressure <dbl>,  
## # visib <dbl>, time\_hour <dttm>

#we are going to join the datasets, first we have to make tibbles of all of the data that we want from each dataset

planes |>  
 count(tailnum) |>  
 filter(n>1)

## # A tibble: 0 × 2  
## # ℹ 2 variables: tailnum <chr>, n <int>

weather |>  
 count(year,month,day,hour,origin)|>  
 filter(n>1)

## # A tibble: 3 × 6  
## year month day hour origin n  
## <int> <int> <int> <int> <chr> <int>  
## 1 2013 11 3 1 EWR 2  
## 2 2013 11 3 1 JFK 2  
## 3 2013 11 3 1 LGA 2

flights |>  
 count(year,month,day,flight) |>  
 filter(n>1)

## # A tibble: 29,768 × 5  
## year month day flight n  
## <int> <int> <int> <int> <int>  
## 1 2013 1 1 1 2  
## 2 2013 1 1 3 2  
## 3 2013 1 1 4 2  
## 4 2013 1 1 11 3  
## 5 2013 1 1 15 2  
## 6 2013 1 1 21 2  
## 7 2013 1 1 27 4  
## 8 2013 1 1 31 2  
## 9 2013 1 1 32 2  
## 10 2013 1 1 35 2  
## # ℹ 29,758 more rows

flights |>  
 count(year,month,day,tailnum) |>  
 filter(n>1)

## # A tibble: 64,928 × 5  
## year month day tailnum n  
## <int> <int> <int> <chr> <int>  
## 1 2013 1 1 N0EGMQ 2  
## 2 2013 1 1 N11189 2  
## 3 2013 1 1 N11536 2  
## 4 2013 1 1 N11544 3  
## 5 2013 1 1 N11551 2  
## 6 2013 1 1 N12540 2  
## 7 2013 1 1 N12567 2  
## 8 2013 1 1 N13123 2  
## 9 2013 1 1 N13538 3  
## 10 2013 1 1 N13566 3  
## # ℹ 64,918 more rows

then we are going to use mutating joins which adds variables to the right side of the sata table

flights2 <- flights |>  
 select(year:day,hour,origin,dest,tailnum,carrier)  
flights2

## # A tibble: 336,776 × 8  
## year month day hour origin dest tailnum carrier  
## <int> <int> <int> <dbl> <chr> <chr> <chr> <chr>   
## 1 2013 1 1 5 EWR IAH N14228 UA   
## 2 2013 1 1 5 LGA IAH N24211 UA   
## 3 2013 1 1 5 JFK MIA N619AA AA   
## 4 2013 1 1 5 JFK BQN N804JB B6   
## 5 2013 1 1 6 LGA ATL N668DN DL   
## 6 2013 1 1 5 EWR ORD N39463 UA   
## 7 2013 1 1 6 EWR FLL N516JB B6   
## 8 2013 1 1 6 LGA IAD N829AS EV   
## 9 2013 1 1 6 JFK MCO N593JB B6   
## 10 2013 1 1 6 LGA ORD N3ALAA AA   
## # ℹ 336,766 more rows

flights2 |>  
 select(-origin, -dest) |>  
 left\_join(airlines, by = "carrier")

## # A tibble: 336,776 × 7  
## year month day hour tailnum carrier name   
## <int> <int> <int> <dbl> <chr> <chr> <chr>   
## 1 2013 1 1 5 N14228 UA United Air Lines Inc.   
## 2 2013 1 1 5 N24211 UA United Air Lines Inc.   
## 3 2013 1 1 5 N619AA AA American Airlines Inc.   
## 4 2013 1 1 5 N804JB B6 JetBlue Airways   
## 5 2013 1 1 6 N668DN DL Delta Air Lines Inc.   
## 6 2013 1 1 5 N39463 UA United Air Lines Inc.   
## 7 2013 1 1 6 N516JB B6 JetBlue Airways   
## 8 2013 1 1 6 N829AS EV ExpressJet Airlines Inc.  
## 9 2013 1 1 6 N593JB B6 JetBlue Airways   
## 10 2013 1 1 6 N3ALAA AA American Airlines Inc.   
## # ℹ 336,766 more rows

flights2 |>  
 select(-origin, -dest) |>  
 mutate(name=airlines$name[match(carrier, airlines$carrier)])

## # A tibble: 336,776 × 7  
## year month day hour tailnum carrier name   
## <int> <int> <int> <dbl> <chr> <chr> <chr>   
## 1 2013 1 1 5 N14228 UA United Air Lines Inc.   
## 2 2013 1 1 5 N24211 UA United Air Lines Inc.   
## 3 2013 1 1 5 N619AA AA American Airlines Inc.   
## 4 2013 1 1 5 N804JB B6 JetBlue Airways   
## 5 2013 1 1 6 N668DN DL Delta Air Lines Inc.   
## 6 2013 1 1 5 N39463 UA United Air Lines Inc.   
## 7 2013 1 1 6 N516JB B6 JetBlue Airways   
## 8 2013 1 1 6 N829AS EV ExpressJet Airlines Inc.  
## 9 2013 1 1 6 N593JB B6 JetBlue Airways   
## 10 2013 1 1 6 N3ALAA AA American Airlines Inc.   
## # ℹ 336,766 more rows

x <- tribble(  
 ~key, ~val\_x,  
 1, "x1",  
 2, "x2",  
 3, "x3"  
)  
  
y <- tribble(  
 ~key, ~val\_y,  
 1, "y1",  
 2, "2",  
 3, "y3"  
)  
  
  
x |>  
 inner\_join(y, by = "key")

## # A tibble: 3 × 3  
## key val\_x val\_y  
## <dbl> <chr> <chr>  
## 1 1 x1 y1   
## 2 2 x2 2   
## 3 3 x3 y3

#left join: keeps all observations in x  
#right join: keeps all observations in y  
# full join: keeps all observations in x and y

flights2 |>  
 left\_join(weather)

## Joining with `by = join\_by(year, month, day, hour, origin)`

## # A tibble: 336,776 × 18  
## year month day hour origin dest tailnum carrier temp dewp humid  
## <int> <int> <int> <dbl> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 2013 1 1 5 EWR IAH N14228 UA 39.0 28.0 64.4  
## 2 2013 1 1 5 LGA IAH N24211 UA 39.9 25.0 54.8  
## 3 2013 1 1 5 JFK MIA N619AA AA 39.0 27.0 61.6  
## 4 2013 1 1 5 JFK BQN N804JB B6 39.0 27.0 61.6  
## 5 2013 1 1 6 LGA ATL N668DN DL 39.9 25.0 54.8  
## 6 2013 1 1 5 EWR ORD N39463 UA 39.0 28.0 64.4  
## 7 2013 1 1 6 EWR FLL N516JB B6 37.9 28.0 67.2  
## 8 2013 1 1 6 LGA IAD N829AS EV 39.9 25.0 54.8  
## 9 2013 1 1 6 JFK MCO N593JB B6 37.9 27.0 64.3  
## 10 2013 1 1 6 LGA ORD N3ALAA AA 39.9 25.0 54.8  
## # ℹ 336,766 more rows  
## # ℹ 7 more variables: wind\_dir <dbl>, wind\_speed <dbl>, wind\_gust <dbl>,  
## # precip <dbl>, pressure <dbl>, visib <dbl>, time\_hour <dttm>

flights2 |>  
 left\_join(planes, by = "tailnum")

## # A tibble: 336,776 × 16  
## year.x month day hour origin dest tailnum carrier year.y type   
## <int> <int> <int> <dbl> <chr> <chr> <chr> <chr> <int> <chr>   
## 1 2013 1 1 5 EWR IAH N14228 UA 1999 Fixed wing mult…  
## 2 2013 1 1 5 LGA IAH N24211 UA 1998 Fixed wing mult…  
## 3 2013 1 1 5 JFK MIA N619AA AA 1990 Fixed wing mult…  
## 4 2013 1 1 5 JFK BQN N804JB B6 2012 Fixed wing mult…  
## 5 2013 1 1 6 LGA ATL N668DN DL 1991 Fixed wing mult…  
## 6 2013 1 1 5 EWR ORD N39463 UA 2012 Fixed wing mult…  
## 7 2013 1 1 6 EWR FLL N516JB B6 2000 Fixed wing mult…  
## 8 2013 1 1 6 LGA IAD N829AS EV 1998 Fixed wing mult…  
## 9 2013 1 1 6 JFK MCO N593JB B6 2004 Fixed wing mult…  
## 10 2013 1 1 6 LGA ORD N3ALAA AA NA <NA>   
## # ℹ 336,766 more rows  
## # ℹ 6 more variables: manufacturer <chr>, model <chr>, engines <int>,  
## # seats <int>, speed <int>, engine <chr>

flights2 |>  
 left\_join(airports, c("dest"= "faa"))

## # A tibble: 336,776 × 15  
## year month day hour origin dest tailnum carrier name lat lon alt  
## <int> <int> <int> <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 2013 1 1 5 EWR IAH N14228 UA Georg… 30.0 -95.3 97  
## 2 2013 1 1 5 LGA IAH N24211 UA Georg… 30.0 -95.3 97  
## 3 2013 1 1 5 JFK MIA N619AA AA Miami… 25.8 -80.3 8  
## 4 2013 1 1 5 JFK BQN N804JB B6 <NA> NA NA NA  
## 5 2013 1 1 6 LGA ATL N668DN DL Harts… 33.6 -84.4 1026  
## 6 2013 1 1 5 EWR ORD N39463 UA Chica… 42.0 -87.9 668  
## 7 2013 1 1 6 EWR FLL N516JB B6 Fort … 26.1 -80.2 9  
## 8 2013 1 1 6 LGA IAD N829AS EV Washi… 38.9 -77.5 313  
## 9 2013 1 1 6 JFK MCO N593JB B6 Orlan… 28.4 -81.3 96  
## 10 2013 1 1 6 LGA ORD N3ALAA AA Chica… 42.0 -87.9 668  
## # ℹ 336,766 more rows  
## # ℹ 3 more variables: tz <dbl>, dst <chr>, tzone <chr>

Workshop 4: Spatial data

#install.packages("sf")  
#install.packages("terra")  
#install.packages("tmap")  
  
library(sf) #simple features

## Linking to GEOS 3.13.0, GDAL 3.10.1, PROJ 9.5.1; sf\_use\_s2() is TRUE

library(terra) #for raster

## terra 1.8.42

##   
## Attaching package: 'terra'

## The following object is masked from 'package:tidyr':  
##   
## extract

library(tmap) #thematic maps are geographical maps in whcih spatial data distributions are visualized  
library(tidyverse)  
library(readr)

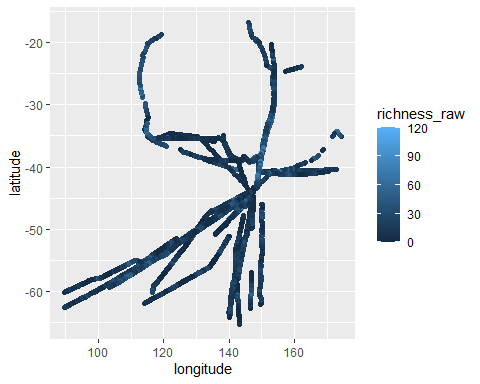
dat <- read\_csv("C:/Users/sydne/OneDrive - James Cook University/Documents/JCU/MB5370/github/SR-Code/Module04\_SR/data/data-for-course/copepods\_raw.csv")

## Rows: 5313 Columns: 11  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): sample\_time\_utc, project, route, vessel, region  
## dbl (6): silk\_id, segment\_no, latitude, longitude, meanlong, richness\_raw  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

dat

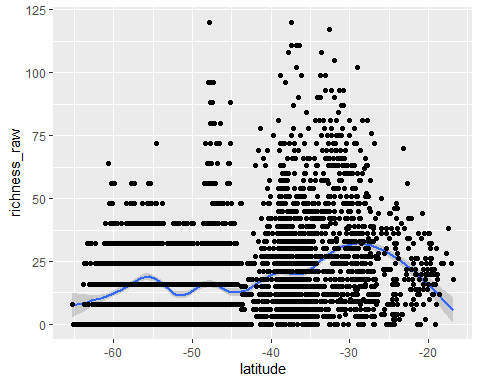
## # A tibble: 5,313 × 11  
## silk\_id segment\_no latitude longitude sample\_time\_utc project route vessel   
## <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr>   
## 1 1 1 -28.3 154. 26/06/2009 22:08 AusCPR BRSY ANL Win…  
## 2 1 5 -28.7 154. 26/06/2009 23:12 AusCPR BRSY ANL Win…  
## 3 1 9 -29.0 154. 27/06/2009 0:17 AusCPR BRSY ANL Win…  
## 4 1 13 -29.3 154. 27/06/2009 1:22 AusCPR BRSY ANL Win…  
## 5 1 17 -29.7 154. 27/06/2009 2:26 AusCPR BRSY ANL Win…  
## 6 1 18 -29.8 154. 27/06/2009 2:43 AusCPR BRSY ANL Win…  
## 7 1 26 -30.4 153. 27/06/2009 4:52 AusCPR BRSY ANL Win…  
## 8 1 30 -30.7 153. 27/06/2009 5:57 AusCPR BRSY ANL Win…  
## 9 1 33 -31.0 153. 27/06/2009 6:45 AusCPR BRSY ANL Win…  
## 10 1 37 -31.3 153. 27/06/2009 7:50 AusCPR BRSY ANL Win…  
## # ℹ 5,303 more rows  
## # ℹ 3 more variables: meanlong <dbl>, region <chr>, richness\_raw <dbl>

library(ggplot2)  
  
ggplot(dat)+  
 aes(x=longitude,y=latitude, color=richness\_raw)+  
 geom\_point()



ggplot(dat, aes(x = latitude, y = richness\_raw))+  
 stat\_smooth()+  
 geom\_point()

## `geom\_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'



sdat <-st\_as\_sf(dat, coords = c("longitude", "latitude"),  
 crs = 4326)  
#st\_as\_sf converts different data types to simple feature  
#coords gives the name of the columns that relate to the spatial coordinates   
#crs stands for coordinate reference system  
  
crs4326 <-st\_crs(4326)  
crs4326$Name

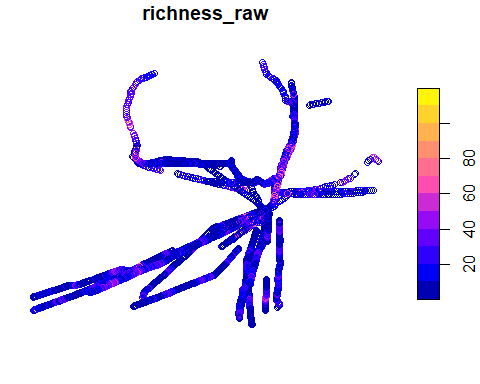
## [1] "WGS 84"

sdat

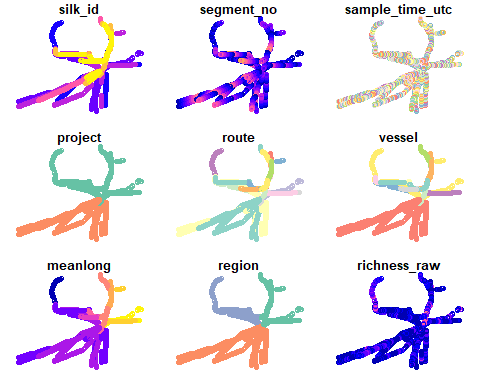
## Simple feature collection with 5313 features and 9 fields  
## Geometry type: POINT  
## Dimension: XY  
## Bounding box: xmin: 89.6107 ymin: -65.2428 xmax: 174.335 ymax: -16.80253  
## Geodetic CRS: WGS 84  
## # A tibble: 5,313 × 10  
## silk\_id segment\_no sample\_time\_utc project route vessel meanlong region  
## \* <dbl> <dbl> <chr> <chr> <chr> <chr> <dbl> <chr>   
## 1 1 1 26/06/2009 22:08 AusCPR BRSY ANL Windar… 153. East   
## 2 1 5 26/06/2009 23:12 AusCPR BRSY ANL Windar… 153. East   
## 3 1 9 27/06/2009 0:17 AusCPR BRSY ANL Windar… 153. East   
## 4 1 13 27/06/2009 1:22 AusCPR BRSY ANL Windar… 153. East   
## 5 1 17 27/06/2009 2:26 AusCPR BRSY ANL Windar… 153. East   
## 6 1 18 27/06/2009 2:43 AusCPR BRSY ANL Windar… 153. East   
## 7 1 26 27/06/2009 4:52 AusCPR BRSY ANL Windar… 153. East   
## 8 1 30 27/06/2009 5:57 AusCPR BRSY ANL Windar… 153. East   
## 9 1 33 27/06/2009 6:45 AusCPR BRSY ANL Windar… 153. East   
## 10 1 37 27/06/2009 7:50 AusCPR BRSY ANL Windar… 153. East   
## # ℹ 5,303 more rows  
## # ℹ 2 more variables: richness\_raw <dbl>, geometry <POINT [°]>

Cartography

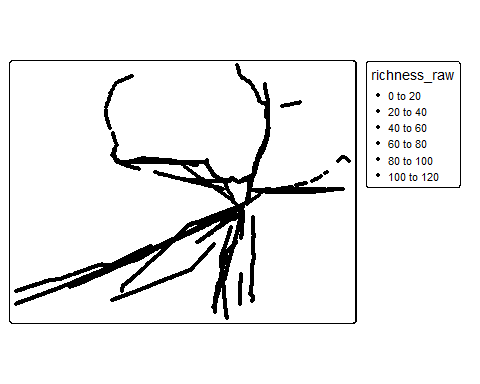
plot(sdat["richness\_raw"])



plot(sdat)



tm\_shape(sdat)+  
 tm\_dots(col = "richness\_raw")



#tmap\_save(tm1,filename = "richness-map.png",  
 #width = 600, heights = 600)  
# cant get the above tm1 to work, I do not uderstand where it is coming from COME BACK

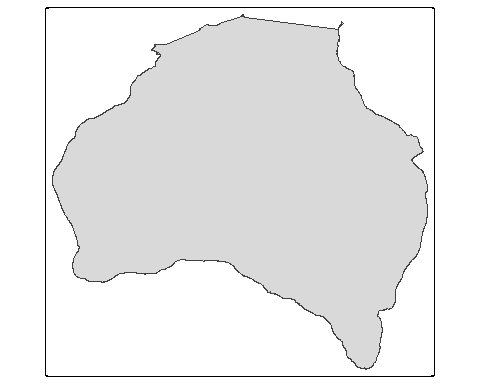
aus <- st\_read("C:/Users/sydne/OneDrive - James Cook University/Documents/JCU/MB5370/github/SR-Code/Module04\_SR/data/data-for-course/spatial-data/Aussie/Aussie.shp")

## Reading layer `Aussie' from data source   
## `C:\Users\sydne\OneDrive - James Cook University\Documents\JCU\MB5370\github\SR-Code\Module04\_SR\data\data-for-course\spatial-data\Aussie\Aussie.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 8 features and 1 field  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: 112.9211 ymin: -43.63192 xmax: 153.6389 ymax: -9.229614  
## Geodetic CRS: WGS 84

shelf <- st\_read("C:/Users/sydne/OneDrive - James Cook University/Documents/JCU/MB5370/github/SR-Code/Module04\_SR/data/data-for-course/spatial-data/aus\_shelf/aus\_shelf.shp")

## Reading layer `aus\_shelf' from data source   
## `C:\Users\sydne\OneDrive - James Cook University\Documents\JCU\MB5370\github\SR-Code\Module04\_SR\data\data-for-course\spatial-data\aus\_shelf\aus\_shelf.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 1 feature and 1 field  
## Geometry type: MULTIPOLYGON  
## Dimension: XY  
## Bounding box: xmin: 112.2242 ymin: -44.1284 xmax: 153.8942 ymax: -8.8798  
## Geodetic CRS: GRS 1980(IUGG, 1980)

tm\_shape(shelf)+  
 tm\_polygons()



tm\_shape(shelf, box = sdat)+  
 tm\_polygons()+  
 tm\_shape(aus)+  
 tm\_polygons()+  
 tm\_shape(sdat)+  
 tm\_dots()

