* + 1. **Exercises**

1. **Run ggplot(data = mpg). What do you see?**

Nothing...the plot goes blank and there is nothing to see. I assume we need to use the GEOM\_FUNCTION to actually plot the data.

1. **How many rows are in mpg? How many columns?**

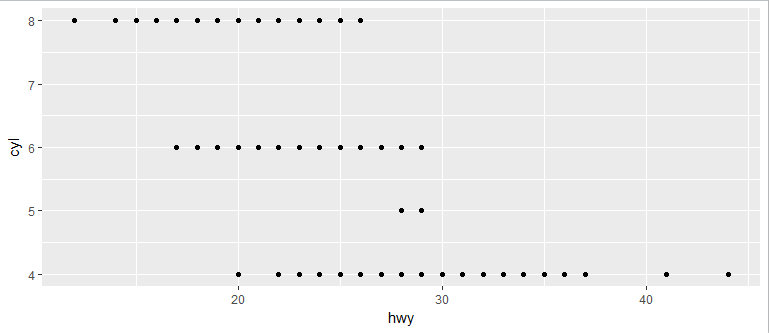
The table (or A tibble according to R) is 234 rows and 11 columns.

1. **What does the drv variable describe? Read the help for ?mpg to find out.**

It describes the drivetrain of the vehicle, whether it's front-wheel, rear-wheel, or four-wheel drive.

1. **Make a scatterplot of hwy vs cyl.**

See below screenshot



1. **What happens if you make a scatterplot of class vs drv? Why is the plot not useful?**

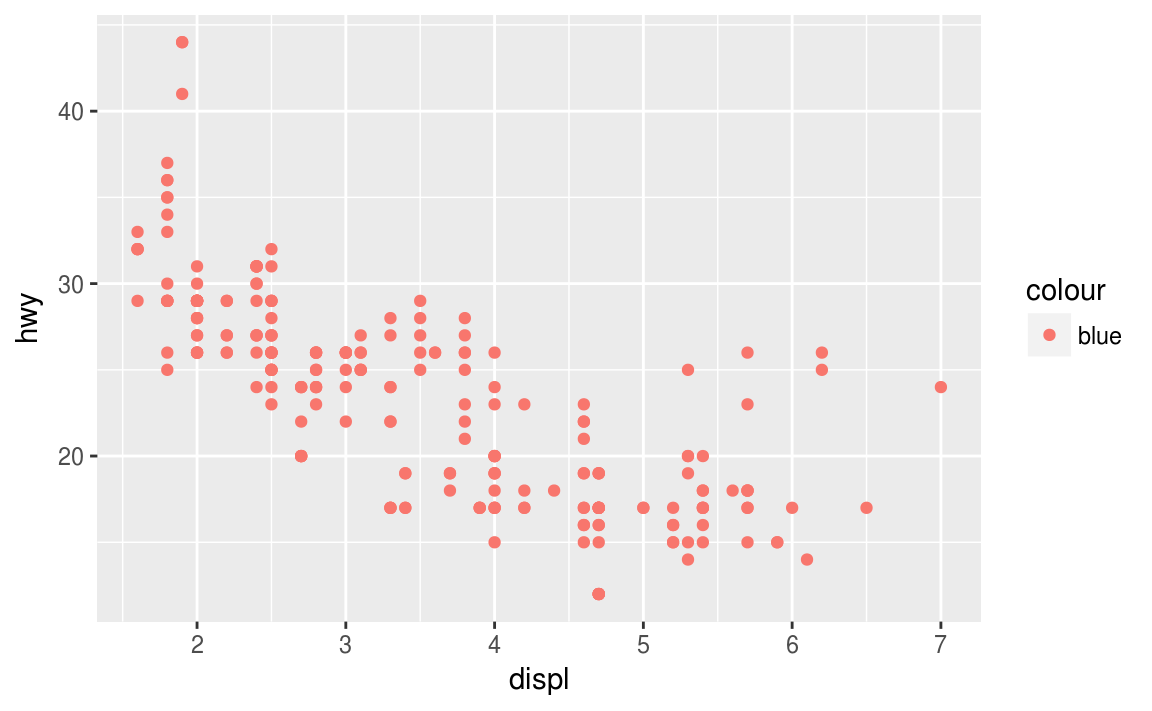
Data is not useful below since both variables don’t have a positive/linear relationship. They’re both facts about the car, but you can’t “increase/decrease” in either variable.

**3.3.1 Exercises**

1. **What’s gone wrong with this code? Why are the points not blue?**

**ggplot**(data = mpg) +

**geom\_point**(mapping = **aes**(x = displ, y = hwy, color = "blue"))



The color=”blue” needs to be outside aes(). If you instead put blue outside, like this:

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

It will turn the points blue.

1. **Which variables in mpg are categorical? Which variables are continuous? (Hint: type ?mpg to read the documentation for the dataset). How can you see this information when you run mpg?**

Categorical:

* + manufacturer
  + model
  + trans
  + drv
  + fl
  + class

Continuous:

* + displ
  + year
  + cyl
  + cty
  + hwy

I looked at the <> within the “tibble” when typing mpg. All columns that had <chr> were categorical. Things that held numbers <dbl>,<int> were continuous.

1. **Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?**

R returns “Error: a continuous variable cannot be mapped to shape” when mapping all three. For the other two (color and size) it works and does color based on a hue, and size relative from smallest to largest. So size should be used for a continuous variable, shape for a categorical variable, and color could be used by either.

1. **What happens if you map the same variable to multiple aesthetics?**

R will map the data, but it is not very useful. If you use a discrete variable, it’ll say it’s not advised for shape. If you use a continuous variable, it won’t work for shape.

1. **What does the stroke aesthetic do? What shapes does it work with? (Hint: use ?geom\_point)**

It allows you to modify the width of the border. This works with shapes that have a border, for example shape 21.

1. **What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)?**

In this instance, ggplot will still work. It reads the new “variable” as true or false. So when displ < 5 it displays true, when it is greater than 5 it displays false.

### 3.5.1 Exercises

1. **What happens if you facet on a continuous variable?**

It will wrap the plot, but do it by every single distinct value in the continuous variable. It’s not very useful as it will product many different graphs in the output.

1. **What do the empty cells in plot with facet\_grid(drv ~ cyl) mean? How do they relate to this plot?**

**ggplot(data = mpg) +**

**geom\_point(mapping = aes(x = drv, y = cyl))**

The empty cells mean there are no data points for that combination of drivetrain and cylinders. For example there are no rear wheel drive cars that have 4 cylinders.

1. **What plots does the following code make? What does . do?**

**ggplot(data = mpg) +**

**geom\_point(mapping = aes(x = displ, y = hwy)) +**

**facet\_grid(drv ~ .)**

**ggplot(data = mpg) +**

**geom\_point(mapping = aes(x = displ, y = hwy)) +**

**facet\_grid(. ~ cyl)**

The first piece of code creates four scatter plats for drivetrain on the y-axis. The second piece of code creates four scatter plots for the different cylinder types on the x-axis. The period allows the user to only implement one variable and then dictate which axis it’ll go on.

1. **Take the first faceted plot in this section:**

**ggplot(data = mpg) +**

**geom\_point(mapping = aes(x = displ, y = hwy)) +**

**facet\_wrap(~ class, nrow = 2)**

**What are the advantages to using faceting instead of the colour aesthetic? What are the disadvantages? How might the balance change if you had a larger dataset?**

Allows you to quickly look at the interaction between two variables broken out by a third variable. In this example, if I wanted to see the relationship between engine size (displ) and hwy mpg by the different car types, I can quickly see subcompacts have a fairly negative linear relationship, while pickups don’t have much of a spread. Also helps the user quickly see where the different classes cluster.

1. **Read ?facet\_wrap. What does nrow do? What does ncol do? What other options control the layout of the individual panels? Why doesn’t facet\_grid() have nrow and ncol argument?**

Nrow = number of rows; ncol = number of columns

You can fix the scale (default) or let it free in both or one dimension. Can also shrink the scales to fit the output of statistics.

Facet\_grid doesn’t need nrow/ncol because it determines both of those depending on the variables within the (). The first variable ([first variable controls rows]~[second variable controls columns])

1. **When using facet\_grid() you should usually put the variable with more unique levels in the columns. Why?**

Since the x-axis labels stay at the very bottom, if you have multiple “rows” of graphs it’s harder to understand which point goes to which. Columns allow for easier reading of the plot.

### 3.6.1 Exercises

* **What geom would you use to draw a line chart? A boxplot? A histogram? An area chart?**
* Line chart = geom\_line
* Boxplot = geom\_boxplot
* Histogram = geom\_col
* Area Chart = geom\_area
* **Run this code in your head and predict what the output will look like. Then, run the code in R and check your predictions.**

**ggplot**(data = mpg, mapping = **aes**(x = displ, y = hwy, color = drv)) +

**geom\_point**() +

**geom\_smooth**(se = FALSE)

* **What does show.legend = FALSE do? What happens if you remove it?  
  Why do you think I used it earlier in the chapter?**

Does not show the “legend” of the variable. Earlier you didn’t want to show it so you wouldn’t have the drivetrain taking up unnecessary space.

* **What does the se argument to geom\_smooth() do?**

It does not show the confidence interval around the lines (aka removes the gray background)

* **Will these two graphs look different? Why/why not?**

**ggplot**(data = mpg, mapping = **aes**(x = displ, y = hwy)) +

**geom\_point**() +

**geom\_smooth**()

**ggplot**() +

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

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

These graphs will look the same, since you are not changing the x and y axis in the “local” area as you have set in the “global” area. In both cases the x-axis will be displ and the y-axis will be hwy.

* **Recreate the R code necessary to generate the following graphs.**
* ggplot(data=mpg, mapping =aes(x=displ,y=hwy))+ geom\_smooth(se = FALSE) +geom\_point()
* ggplot(data=mpg, mapping =aes(x=displ,y=hwy))+ geom\_smooth(mapping=aes(group=drv), se = FALSE) +geom\_point()
* ggplot(data=mpg, mapping =aes(x=displ,y=hwy, color=drv))+ geom\_smooth(se = FALSE) +geom\_point()
* ggplot(data=mpg, mapping =aes(x=displ,y=hwy))+ geom\_smooth(se = FALSE) +geom\_point(mapping=aes(color=drv))
* ggplot(data=mpg, mapping =aes(x=displ,y=hwy))+ geom\_smooth(mapping=aes(linetype=drv),se = FALSE) +geom\_point(mapping=aes(color=drv))
* ggplot(data=mpg, mapping =aes(x=displ,y=hwy))+ geom\_point(mapping=aes(color=drv))

### 3.7.1 Exercises

1. **What is the default geom associated with stat\_summary()? How could you rewrite the previous plot to use that geom function instead of the stat function?**

Default geom = “pointrange”

ggplot(data = diamonds) + geom\_pointrange( mapping = aes(x = cut, y = depth, ymin=depth, ymax=depth))

1. **What does geom\_col() do? How is it different to geom\_bar()?**

Geom\_col will allow you to plot values in the data on the y-axis. Geom\_bar() will use the stat\_count by default or how many times the variable on the x-axis shows up in the dataset.

1. **Most geoms and stats come in pairs that are almost always used in concert. Read through the documentation and make a list of all the pairs. What do they have in common?**
2. **What variables does stat\_smooth() compute? What parameters control its behaviour?**

Y = predicted value; ymin=lower pointwise confidence interval around the mean; ymax=upper pointwise confidence interval around the mean; se = standard error

1. **In our proportion bar chart, we need to set group = 1. Why? In other words what is the problem with these two graphs?**

**ggplot**(data = diamonds) +

**geom\_bar**(mapping = **aes**(x = cut, y = ..prop..))

**ggplot**(data = diamonds) +

**geom\_bar**(mapping = **aes**(x = cut, fill = color, y = ..prop..))

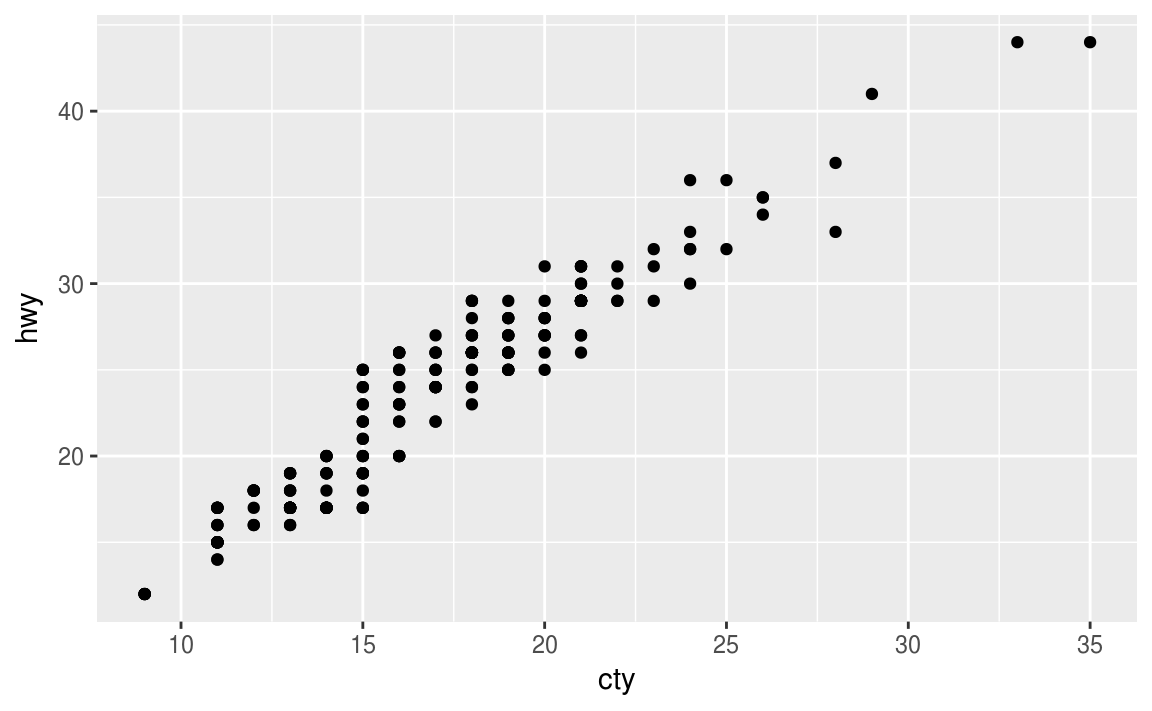
If we don’t tell it to only use one group (group =1) then it will default to find the proportion of only each x variable compared to that x-variable (aka everything will add up to 1). If you want to see the proportion of the x-values relative to the whole dataset then you must select group = 1, so the proportion property for the y-axis can compare across the whole dataset.

### 3.8.1 Exercises

1. **What is the problem with this plot? How could you improve it?**

**ggplot**(data = mpg, mapping = **aes**(x = cty, y = hwy)) +

**geom\_point**()



Points are all clustered together. If we use position = “jitter” we get a much better plot of points since they aren’t all in these “perfect” columns.

1. **What parameters to geom\_jitter() control the amount of jittering?**

“width” and “height” help keep jittering under control, otherwise the default will be 40% of the resolution of the data.

1. **Compare and contrast geom\_jitter() with geom\_count().**

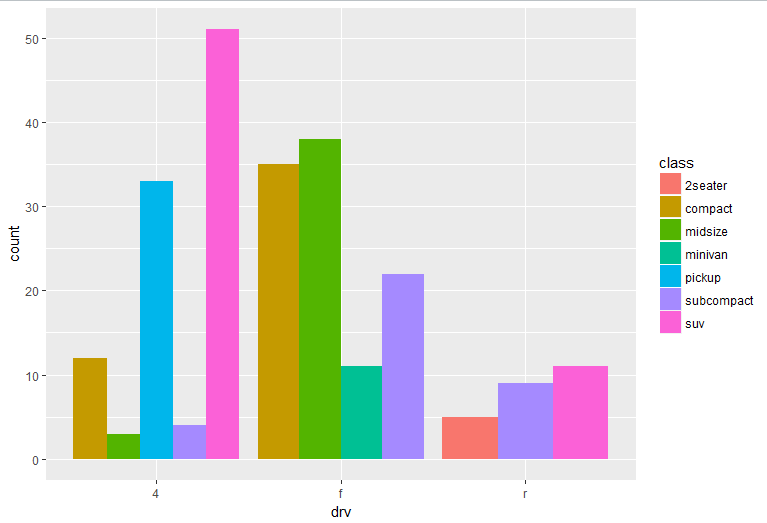
Geom\_jitter() uses the stat of identity and the position = “jitter”. While geom\_count() uses the stat of sum and the position = “identity”. Both are different ways to handle “overplotting”, jitter does it by adding random noise to the points while count does it by counting observations at different positions.

1. **What’s the default position adjustment for geom\_boxplot()? Create a visualisation of the mpgdataset that demonstrates it.**

Default position = “dodge” and this code/visualization helps demonstrate it:

ggplot(data = mpg) +

geom\_bar(mapping = aes(x = drv, fill = class), position = "dodge")



### 3.9.1 Exercises

1. **Turn a stacked bar chart into a pie chart using coord\_polar().**
2. **What does labs() do? Read the documentation.**

Labs helps with labelling in your graph. It can add a title, subtitle, label your axes and caption the data source. In essence it’s very useful to provide more data to the person reading the graph.

1. **What’s the difference between coord\_quickmap() and coord\_map()?**

Coord\_map() projects portions of the earth (which is spherical) onto the 2d plane of a graph. Since the maps are usually not in straight lines, this can require A LOT of computation power. If you need a quick map with straight lines, use coord\_quickmap()

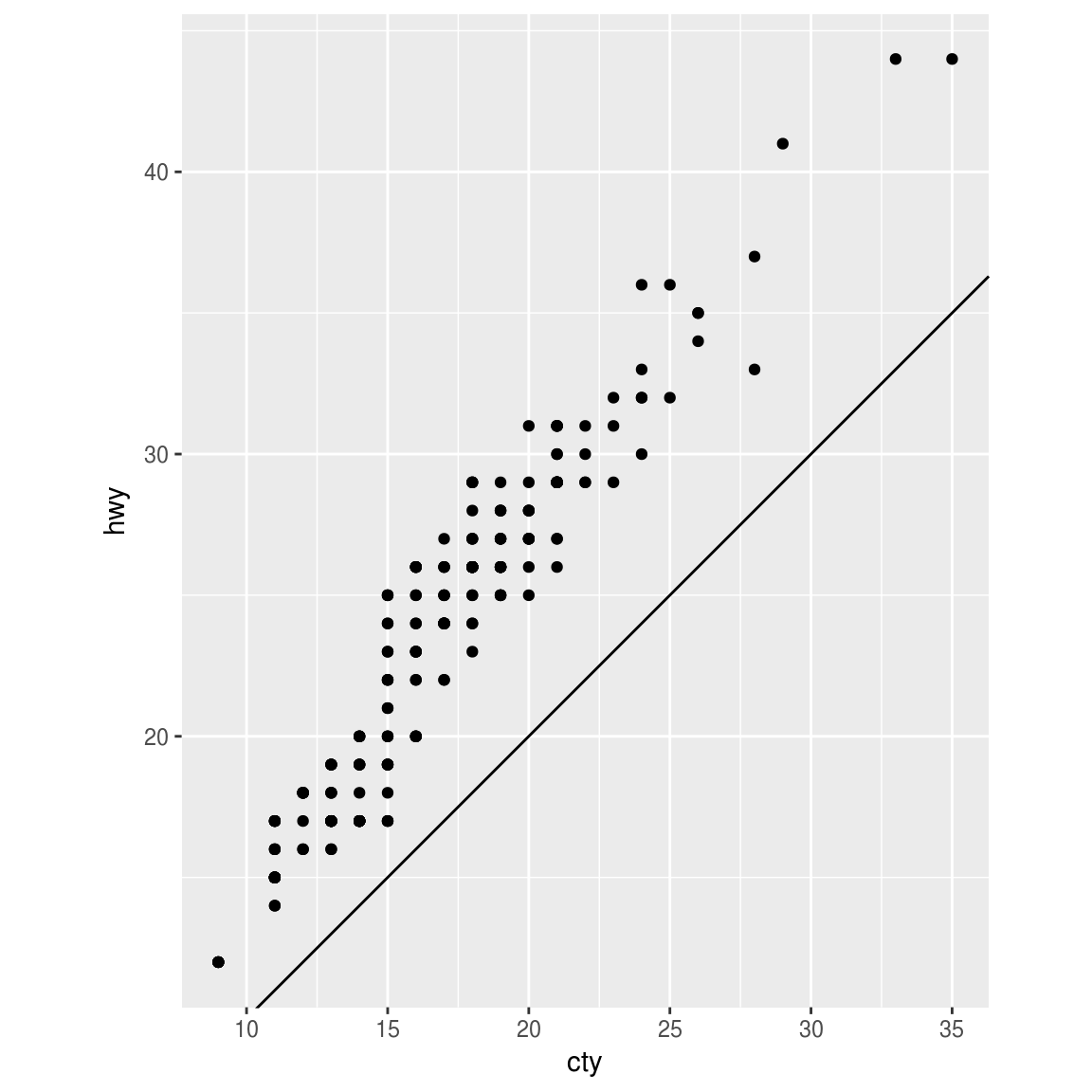
1. **What does the plot below tell you about the relationship between city and highway mpg? Why is coord\_fixed() important? What does geom\_abline() do?**

**ggplot**(data = mpg, mapping = **aes**(x = cty, y = hwy)) +

**geom\_point**() +

**geom\_abline**() +

**coord\_fixed**()



Coord\_fixed will set the aspect ratio of the graph between the two axes, in the default it’s 1. Geom\_abline creates a fixed line

## 4.4 Practice

1. Why does this code not work?

my\_variable <- 10

my\_varıable

*#> Error in eval(expr, envir, enclos): object 'my\_varıable' not found*

Look carefully! (This may seem like an exercise in pointlessness, but training your brain to notice even the tiniest difference will pay off when programming.)

This doesn’t work because in the second version my\_var1able has a “1” instead of an “i”.

1. Tweak each of the following R commands so that they run correctly:

**library**(tidyverse)

**ggplot**(dota = mpg) +

**geom\_point**(mapping = **aes**(x = displ, y = hwy))

**fliter**(mpg, cyl = 8)

**filter**(diamond, carat > 3)

First off, change “dota” to “data”

Then for the first filter, change “fliter” to “filter” and “cyl=8” to “cyl==8”

For the last filter, change “diamond” to “diamonds”

1. Press Alt + Shift + K. What happens? How can you get to the same place using the menus?

It opens up the keyboard shortcut menu. Through the menus you could select “help” and then in the drop down, select “keyboard shortcut menu”

**5.2.4 Exercises**

1. Find all flights that
2. Had an arrival delay of two or more hours
   1. Question1 <- filter(flights, arr\_delay >= 120)
3. Flew to Houston (IAH or HOU)
   1. Question2 <- filter(flights, dest %in% c('IAH','HOU'))
4. Were operated by United, American, or Delta
   1. Question3 <- filter(flights, carrier %in% c('UA','DL','AA'))
5. Departed in summer (July, August, and September)
   1. Question4 <- filter(flights,month %in% c(7,8,9))
6. Arrived more than two hours late, but didn’t leave late
   1. Question5 <- filter(flights,arr\_delay > 120 & dep\_delay <= 0)
7. Were delayed by at least an hour, but made up over 30 minutes in flight
   1. Question6 <- filter(flights, dep\_delay >= 60 & (dep\_delay - arr\_delay)>30)
8. Departed between midnight and 6am (inclusive)
   1. Question7 <- filter(flights, dep\_time >= 0001 & dep\_time <= 0600)
9. Another useful dplyr filtering helper is between(). What does it do? Can you use it to simplify the code needed to answer the previous challenges?

Yep. Could rewrite question 7 this way: Question7 <- filter(flights, between(dep\_time, 0001,0600))

1. How many flights have a missing dep\_time? What other variables are missing? What might these rows represent?

8,225. I’m assuming this means the flight was cancelled. The dep\_delay, arr\_time, and arr\_delay variables are all missing values as well in these rows.

1. Why is NA ^ 0 not missing? Why is NA | TRUE not missing? Why is FALSE & NA not missing? Can you figure out the general rule? (NA \* 0 is a tricky counterexample!)

These first three arguments aren’t missing because in the first one, anything ^ 0 will be 1. Second one, the OR statement means TRUE will override the missing value. Same thing for FALSE & NA, the FALSE component will return FALSE.

### 5.3.1 Exercises

1. How could you use arrange() to sort all missing values to the start? (Hint: use is.na()).

Question1 <- arrange(flights, desc(is.na(dep\_time)))

1. Sort flights to find the most delayed flights. Find the flights that left earliest.

Question2a <- arrange(flights, desc(dep\_delay))

Question2b <- arrange(flights, dep\_delay)

1. Sort flights to find the fastest flights.

Question3 <- arrange(flights, air\_time)

1. Which flights travelled the longest? Which travelled the shortest?

From Newark to LGA is 17 miles. JFK to Honolulu is the longest at 4,983 miles

### 5.4.1 Exercises

1. Brainstorm as many ways as possible to select dep\_time, dep\_delay, arr\_time, and arr\_delayfrom flights.

Select each individually

Select one and then everything()

Rename one of these columns

1. What happens if you include the name of a variable multiple times in a select() call?

It will still show that variable once.

1. What does the one\_of() function do? Why might it be helpful in conjunction with this vector?

vars <- **c**("year", "month", "day", "dep\_delay", "arr\_delay")

It allows you to take in a vector (such as “vars” above) and then pass that through to select. Allows you to create a vector and then not have to retype everytime.

1. Does the result of running the following code surprise you? How do the select helpers deal with case by default? How can you change that default?

**select**(flights, **contains**("TIME"))

I would have assumed, case sensitivity would have meant nothing matches. But you need to add in ignore.case, which you can do like this

select(flights, contains("TIME", ignore.case = FALSE))

### 5.5.2 Exercises

1. Currently dep\_time and sched\_dep\_time are convenient to look at, but hard to compute with because they’re not really continuous numbers. Convert them to a more convenient representation of number of minutes since midnight.

mutate(flights,hour = dep\_time %/% 100, minute = dep\_time %%100, NiceTime = hour\*60 + minute)

1. Compare air\_time with arr\_time - dep\_time. What do you expect to see? What do you see? What do you need to do to fix it?

Would expect to see the same thing. But the numbers are different, because arrival time are in the 24hour clock, while arrival time is in a continuous variable. Like above, need to convert arr\_time and dep\_time to be continuous numbers as minutes since midnight.

1. Compare dep\_time, sched\_dep\_time, and dep\_delay. How would you expect those three numbers to be related?

Well dep\_delay is going to be dep\_time minus sched\_dep\_time assuming both have been turned into continuous variables.

1. Find the 10 most delayed flights using a ranking function. How do you want to handle ties? Carefully read the documentation for min\_rank().

View(arrange(test2, desc(min\_rank(dep\_delay))))

It will treat ties as the same “rank”. If we wanted to take the “first” tie then we should use row\_number()

1. What does 1:3 + 1:10 return? Why?

An array of ascending numbers. You are summing two arrays in this example, starting with 1 + 1, then 2 + 2, then 3 + 3 at this pion the first array resets while the second keeps going, aka 1 + 4, 2 +5, 3 +6, etc.

1. What trigonometric functions does R provide?

?Trig reveals R will provide the basic Trig functions of cos, sin, tan, acos, asin, atan, etc.

### 5.6.7 Exercises

1. Brainstorm at least 5 different ways to assess the typical delay characteristics of a group of flights. Consider the following scenarios:
   * A flight is 15 minutes early 50% of the time, and 15 minutes late 50% of the time.
2. Using this summary function, the median arrival delay for a flight delayed is 21 minutes and the median arrival time for a flight being early is 15 minutes.
   * A flight is always 10 minutes late.
   * A flight is 30 minutes early 50% of the time, and 30 minutes late 50% of the time.
   * 99% of the time a flight is on time. 1% of the time it’s 2 hours late.

flightDelayInfo <-

not\_cancelled %>%

group\_by(flight) %>%

summarise(test = sum(arr\_delay == -15),

flights15Late = sum(arr\_delay >= 15)/n() ,

flights15Early = sum(arr\_delay <= - 15)/n(),

flights10Late = sum(arr\_delay == 10)/n(),

flights30Late = sum(arr\_delay >= 30)/n() ,

flights30Early = sum(arr\_delay <= - 30)/n() ,

flightsOnTime = sum(arr\_delay == 0)/n(),

flights2Hour = sum(arr\_delay >= 120)/n()) %>%

summarise(total\_15 = sum(flights15Early == .5 & flights15Late == .5)

, total\_30 = sum(flights30Early == .5 & flights30Late == .5)

, total\_10 = sum(flights10Late == 1)

, total\_99Perc = sum(flightsOnTime >= 0.99 & flights2Hour ==.01))

Which is more important: arrival delay or departure delay?

Definitely arrival delay. That’s the one that will cause passengers the most headache. You can make up departure delay in the air.

1. Come up with another approach that will give you the same output as not\_cancelled %>% count(dest) and not\_cancelled %>% count(tailnum, wt = distance) (without using count()).

not\_cancelled %>%

group\_by(dest) %>%

summarise(

n = n()

)

1. Our definition of cancelled flights (is.na(dep\_delay) | is.na(arr\_delay) ) is slightly suboptimal. Why? Which is the most important column?

It’s checking to see if both variables contain missing values for the same flight (row). In theory if the dep\_delay is null (aka the flight never took off) then the arr\_delay should also be null (it didn’t land). But this does not seem to be the case in the data set. There are at least 1200 rows where the departure delay was populated but arrival delay was not. In this instance, I would argue dep\_delay is more important since it populates more data.

1. Look at the number of cancelled flights per day. Is there a pattern? Is the proportion of cancelled flights related to the average delay?
2. Which carrier has the worst delays? Challenge: can you disentangle the effects of bad airports vs. bad carriers? Why/why not? (Hint: think about flights %>% group\_by(carrier, dest) %>% summarise(n()))
3. What does the sort argument to count() do. When might you use it?

### 7.3.4 Exercises

1. Explore the distribution of each of the x, y, and z variables in diamonds. What do you learn? Think about a diamond and how you might decide which dimension is the length, width, and depth.

All three variables have a positive linear relationship (as the lenght increases, so does the width and depth).

X = length, y = width, z= depth

1. Explore the distribution of price. Do you discover anything unusual or surprising? (Hint: Carefully think about the binwidth and make sure you try a wide range of values.)

Majority of the diamonds fall between $500-$1500. Also, there are NO diamonds priced between $1460-1540. That’s suprising considering there are 100’s of diamonds in the other bin’s of size 10.

1. How many diamonds are 0.99 carat? How many are 1 carat? What do you think is the cause of the difference?

23 diamonds are 0.99 carat and 1558 are 1 carat. Not knowing much about diamonds, one guess would be it’s purely for commercial reasons. The weight at that point is so close to 1 carat, and the diamond seller would be able to sell the diamond for more if it was marketed as 1 carat.

1. Compare and contrast coord\_cartesian() vs xlim() or ylim() when zooming in on a histogram. What happens if you leave binwidth unset? What happens if you try and zoom so only half a bar shows?

If you leave binwidth unset as you zoom in, then you’re just getting a fatter bar. For example with the diamonds data set, if you leave the binwidth at a 1000 and zoom in on the $1000 to $2000 area (so you can see why there’s a gap) then it just becomes a fat blob. Same thing, if you try to only have “half” the bar show then the bar will just take up the entire plot. The plot/graph doesn’t “resize” the axes themselves.

### 7.4.1 Exercises

1. What happens to missing values in a histogram? What happens to missing values in a bar chart? Why is there a difference?

In a histogram the value is dropped since the x aesthetic needs to be a continuous variable. In a bar chart it would be plotted as a group since the x aesthetic is a categorical variable.

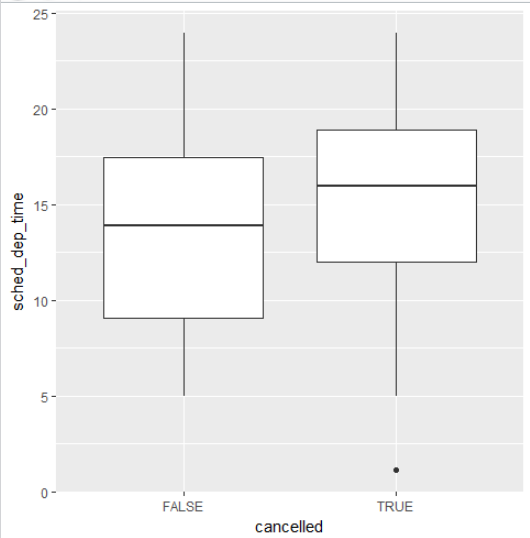
1. What does na.rm = TRUE do in mean() and sum()?

It removes the NA values from any vector BEFORE calculating the mean and the sum.

#### 7.5.1.1 Exercises

1. Use what you’ve learned to improve the visualisation of the departure times of cancelled vs. non-cancelled flights.

Create a boxplot

. 

1. What variable in the diamonds dataset is most important for predicting the price of a diamond? How is that variable correlated with cut? Why does the combination of those two relationships lead to lower quality diamonds being more expensive?

Carat seems to be the most important variable. There does not seem to be much of a correlation between carat (weight) and quality. Meaning lower quality diamonds could be more expensive solely because they weigh more (are bigger).

1. Install the ggstance package, and create a horizontal boxplot. How does this compare to using coord\_flip()?

install.packages("ggstance")

library(ggstance)

ggplot(data = diamonds) +

geom\_boxploth(mapping = aes(x = carat, y = cut))

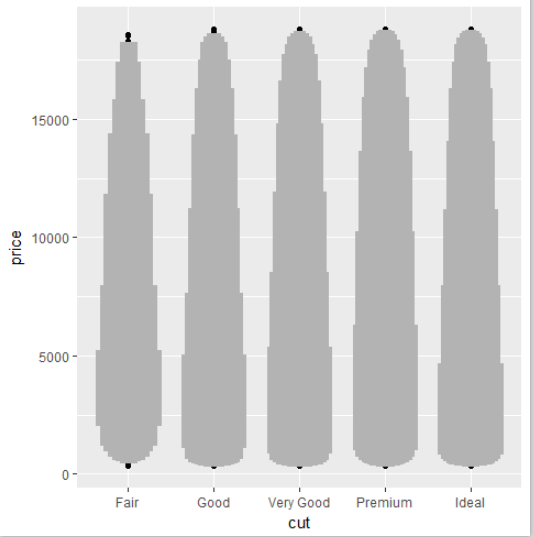
ggplot(data = diamonds) +

geom\_boxplot(mapping = aes(x=cut, y = carat)) +

coord\_flip()

The output is the same for both, but you have to flip x and y between the two.

1. One problem with boxplots is that they were developed in an era of much smaller datasets and tend to display a prohibitively large number of “outlying values”. One approach to remedy this problem is the letter value plot. Install the lvplot package, and try using geom\_lv() to display the distribution of price vs cut. What do you learn? How do you interpret the plots?



While all cut’s have “outliers” in prices, the difference in “spread” between Very Good and Ideal doesn’t too different.

1. Compare and contrast geom\_violin() with a facetted geom\_histogram(), or a coloured geom\_freqpoly(). What are the pros and cons of each method?

The geom\_violin() means you don’t have to select “bins”, it is more of a “sloping” histogram. While the geom\_histogram with a facet is easier to initially read since you can just see the bars and distribution.

1. If you have a small dataset, it’s sometimes useful to use geom\_jitter() to see the relationship between a continuous and categorical variable. The ggbeeswarm package provides a number of methods similar to geom\_jitter(). List them and briefly describe what each one does.

There are two methods:

**geom\_quasirandom** that produces plots that resemble something between jitter and violin. There are several different methods that determine exactly how the random location of the points is generated.

**geom\_beeswarm** creates a shape similar to a violin plot, but by offsetting the points.

#### 7.5.2.1 Exercises

1. How could you rescale the count dataset above to more clearly show the distribution of cut within colour, or colour within cut?

You would want to create a variable that takes the percentage of cut within each color.

1. Use geom\_tile() together with dplyr to explore how average flight delays vary by destination and month of year. What makes the plot difficult to read? How could you improve it?

It’s a mess due to how many destinations there are. It could be improved by either stripping out certain destinations (say the bottom 50%) or sorting them by a meaningful quantity (distance, number of flights, etc.)

1. Why is it slightly better to use aes(x = color, y = cut) rather than aes(x = cut, y = color) in the example above?

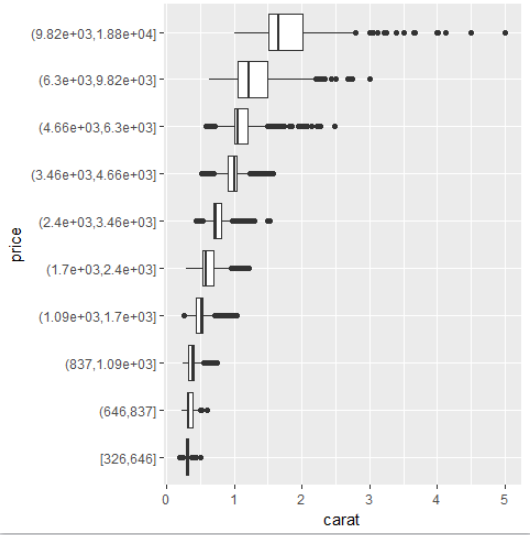
It’s better to use the categorical variable on the y-axis because it’s easer to read the variables, since if they were on the x-axis they might be overlapping.

#### 7.5.3.1 Exercises

1. Instead of summarising the conditional distribution with a boxplot, you could use a frequency polygon. What do you need to consider when using cut\_width() vs cut\_number()? How does that impact a visualisation of the 2d distribution of carat and price?

Cut\_width may have different total numbers within each bin, while cut\_number will have the same number of observations in each bin.

1. Visualise the distribution of carat, partitioned by price.



1. How does the price distribution of very large diamonds compare to small diamonds. Is it as you expect, or does it surprise you?

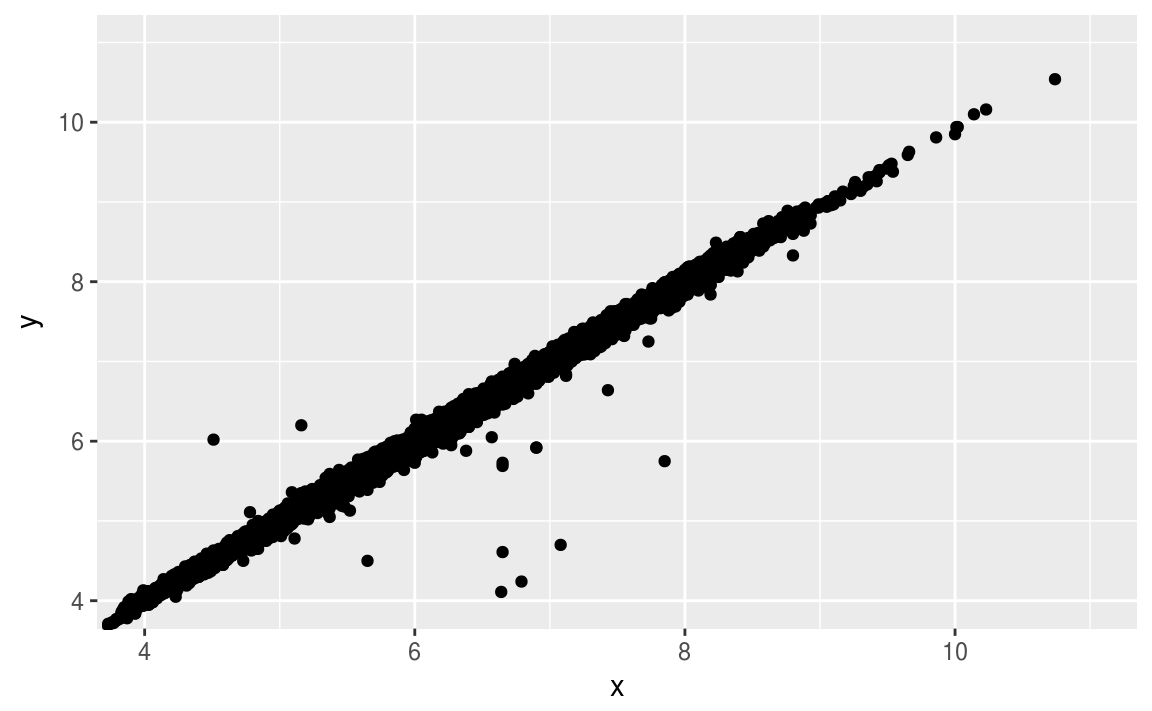
Yes it makes sense. Larger diamonds have a wider spread in price than small diamonds, most likely because at that size the cut, style, and color start playing a much more important role.

1. Combine two of the techniques you’ve learned to visualise the combined distribution of cut, carat, and price.
2. Two dimensional plots reveal outliers that are not visible in one dimensional plots. For example, some points in the plot below have an unusual combination of x and y values, which makes the points outliers even though their x and y values appear normal when examined separately.

**ggplot**(data = diamonds) +

**geom\_point**(mapping = **aes**(x = x, y = y)) +

**coord\_cartesian**(xlim = **c**(4, 11), ylim = **c**(4, 11))



Why is a scatterplot a better display than a binned plot for this case?

The scatter plot helps to display the high correlation between x and y, but also shows the outliers clearly.