STAT167 Lab #2 - Spring 2025

Ethan Choi

2025/4/11

Contents

Discussion/Lab #2 instructions

This week, we will review some base R visualization and `ggplot2` example figures from the lectures.

- First, download the `rmd` file from Canvas.
- Open this `rmd` file in RStudio and click `Knit -> Knit to PDF` to render it to PDF format. You need to have `LaTex` installed on the computer to render it to PDF format. If not, you can also render it to HTML format.
- Read this `rmd` file and the rendered `pdf/html` file side-by-side, to see how this document was generated!

- Be sure to play with this document! Change it. Break it. Fix it. The best way to learn R Markdown (or really almost anything) is to try, fail, then find out what you did wrong.
- Read over the `ggplot2` example code and check the output. If you have any questions about certain functions or parameters, it is the time to ask!
- There are some exercises through out this document. Replace **INSERT_YOUR_ANSWER** with your own answers. Knit the file, and check your results.

Please comment your R code thoroughly, and follow the R coding style guideline (https://google.github.io/styleguide/Rguide.xml). Partial credit will be deducted for insufficient commenting or poor coding styles.

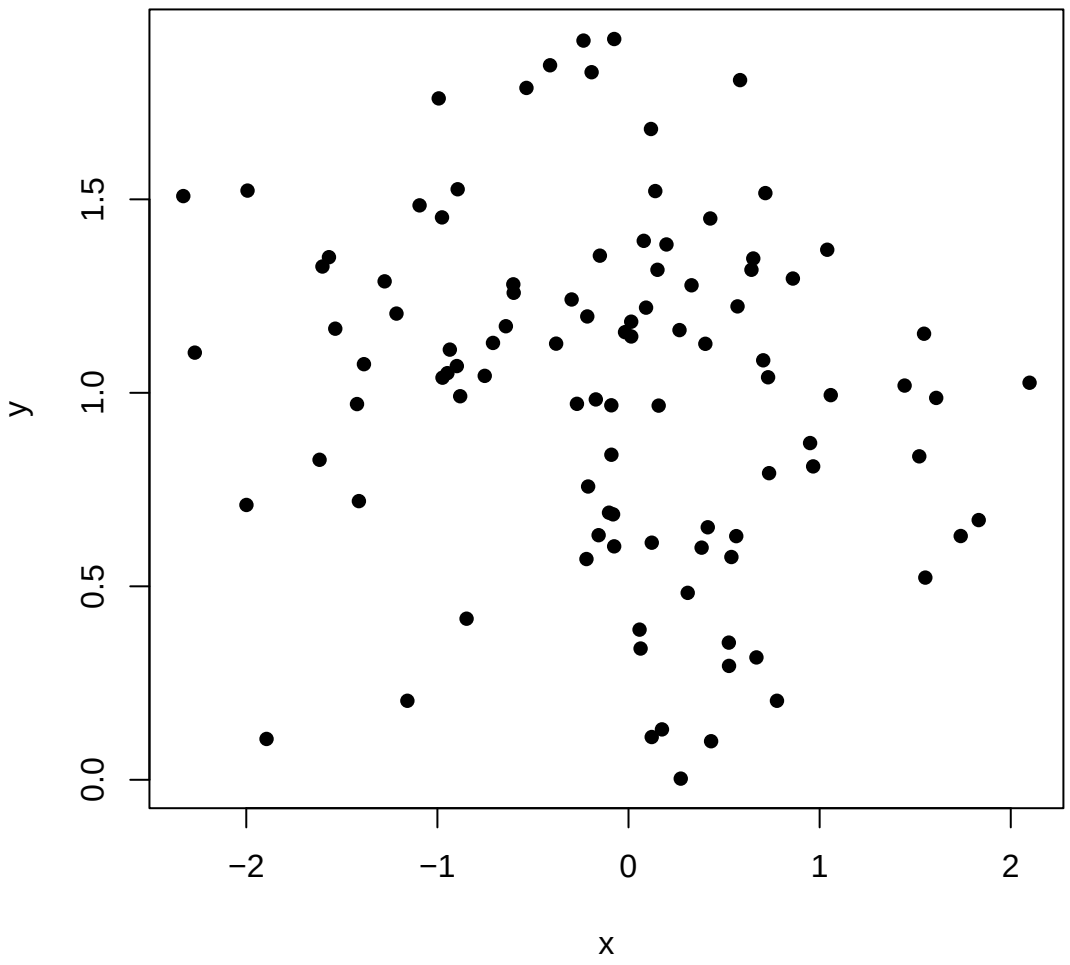
Lab submission guideline

- After you completed all exercises, save your file to `FirstnameLastname-SID-lab2.rmd` and save the rendered pdf file to `FirstnameLastname-SID-lab2.pdf`. If you can not knit it to pdf, knit it to html first and then print/save it to pdf format.
- Submit **BOTH your source `rmd` file and the knitted `pdf` file** to **GradeScope**. Do NOT create a zip file.
- You can submit multiple times, you last submission will be graded.

Lecture Review - base R graphics

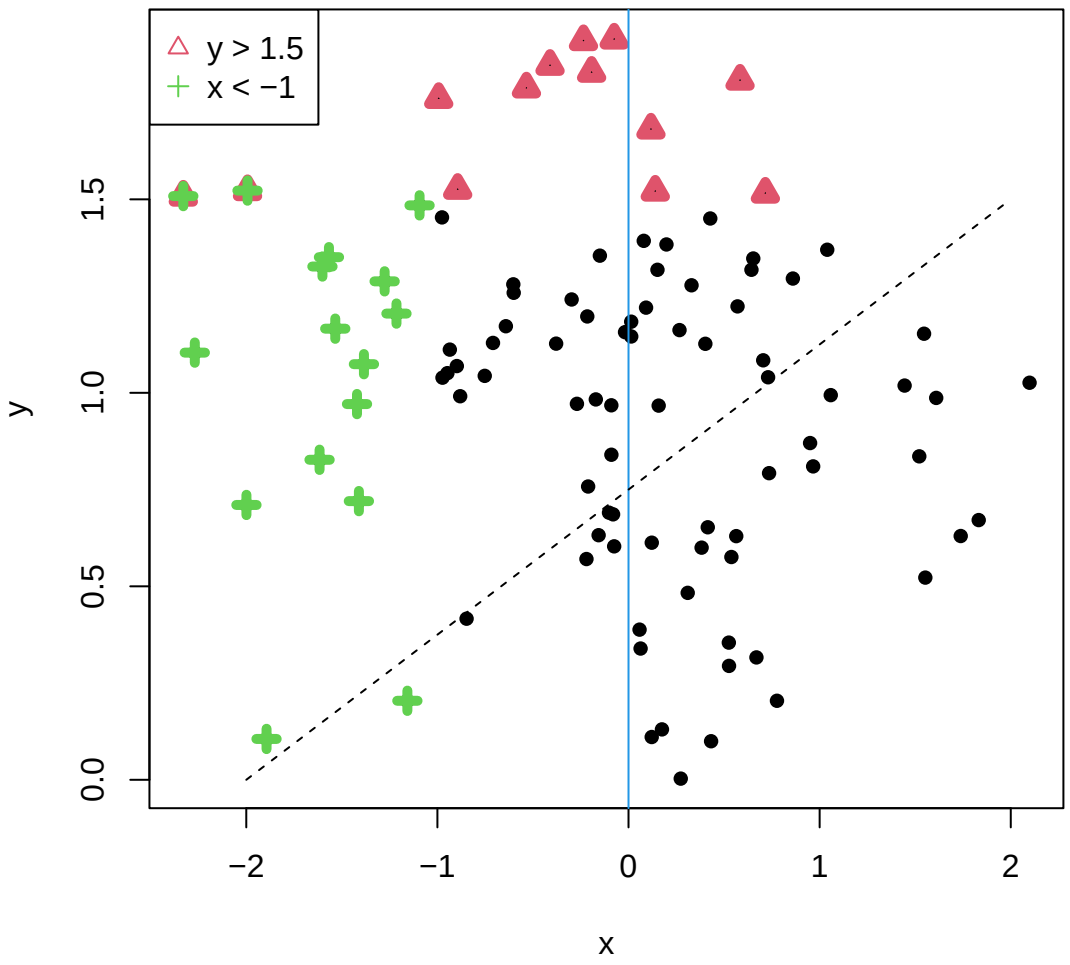
Scatter plot of two data sets

```
# generate random normals
set.seed(167)
x <- rnorm(100)
y <- rnorm(n = 100, mean = 1, sd = .5)
plot(x, y, pch = 16)
```



Annotate your scatter plot

```
plot(x, y, pch = 16)
points(x[y > 1.5], y[y > 1.5], col = 2, pch = 2, lwd = 5)
points(x[x < -1], y[x < -1], col = 3, pch = 3, lwd = 5) # do not write as x[x<-1]
legend("topleft", legend = c("y > 1.5","x < -1"), col = c(2, 3), pch = c(2, 3))
abline(v = 0, col = 4)
lines(x = c(-2, 2), y = c(0, 1.5), lty = 2)
```



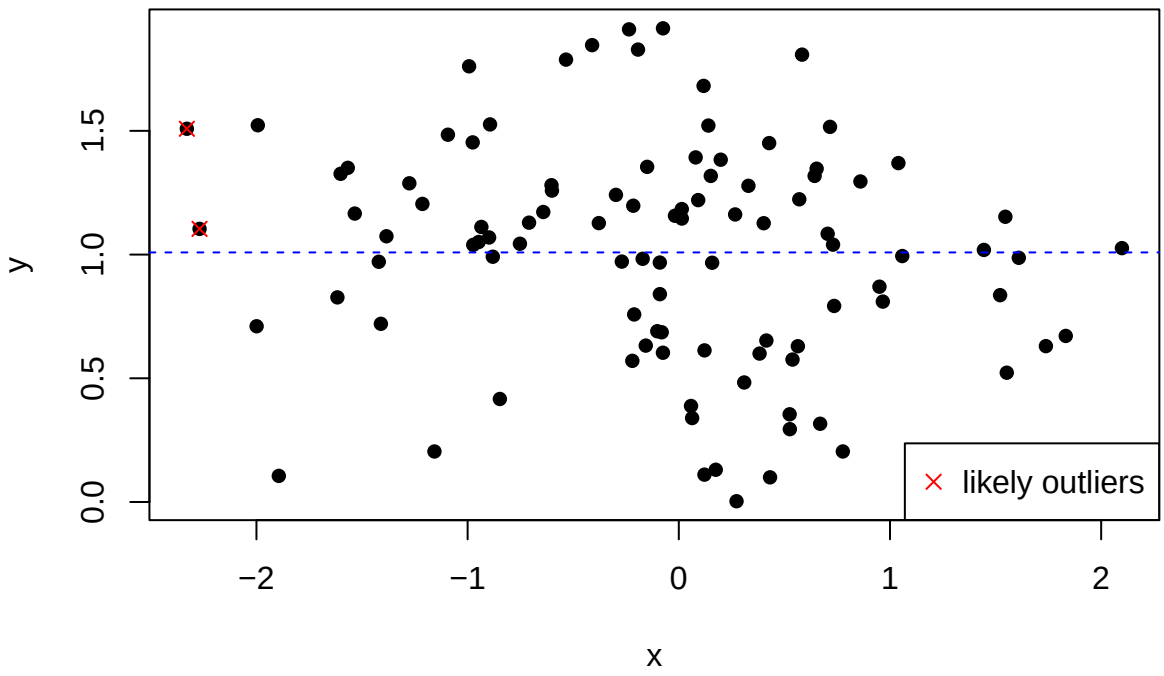
Exercise #1

Annotate your own scatter plot:

- select points with x < -2 (leave space on both sides of <), color them in red and change their point symbol (to anything other than solid points)
- add a legend at the right bottom of the figure indicating those red color points are "likely outliers"
- add a blue dashed horizontal line at $\bar{y}$

ANSWERS

```
plot(x, y, pch = 16)
y.bar <- mean(y) # it does not necessarily equal to 1.0
points(x[x < -2], y[x < -2], col = "red", pch = 4)
legend("bottomright", legend = "likely outliers", col = "red", pch = 4)
abline(h = y.bar, col = "blue", lty = 2)
```


y
0.0
0.5
1.0
1.5
−2
−1
0
1
2
x
likely outliers

Lecture Review - ggplot2

Install the tidyverse package

```
# install the tidyverse package first if you have not done it  yet.
#install.packages("tidyverse") # you can comment out this line after you have installed `tidyverse`
library(tidyverse)
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

The mpg data set

This data set contains fuel economy data 1999 - 2008 for 38 popular car models.

https://ggplot2.tidyverse.org/reference/mpg.html

```
?mpg
## starting httpd help server ... done
dim(mpg)  # dimension of the table
## [1] 234  11
mpg  # print/view mpg (we will introduce tibble later in this class)
## # 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~ f        18    29 p     comp~
##  2 audi         a4           1.8  1999     4 manu~ f        21    29 p     comp~
##  3 audi         a4           2    2008     4 manu~ f        20    31 p     comp~
##  4 audi         a4           2    2008     4 auto~ f        21    30 p     comp~
##  5 audi         a4           2.8  1999     6 auto~ f        16    26 p     comp~
##  6 audi         a4           2.8  1999     6 manu~ f        18    26 p     comp~
##  7 audi         a4           3.1  2008     6 auto~ f        18    27 p     comp~
##  8 audi         a4 quattro   1.8  1999     4 manu~ 4        18    26 p     comp~
##  9 audi         a4 quattro   1.8  1999     4 auto~ 4        16    25 p     comp~
## 10 audi         a4 quattro   2    2008     4 manu~ 4        20    28 p     comp~
## # i 224 more rows
str(mpg)  # list the structures in mpg
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
```

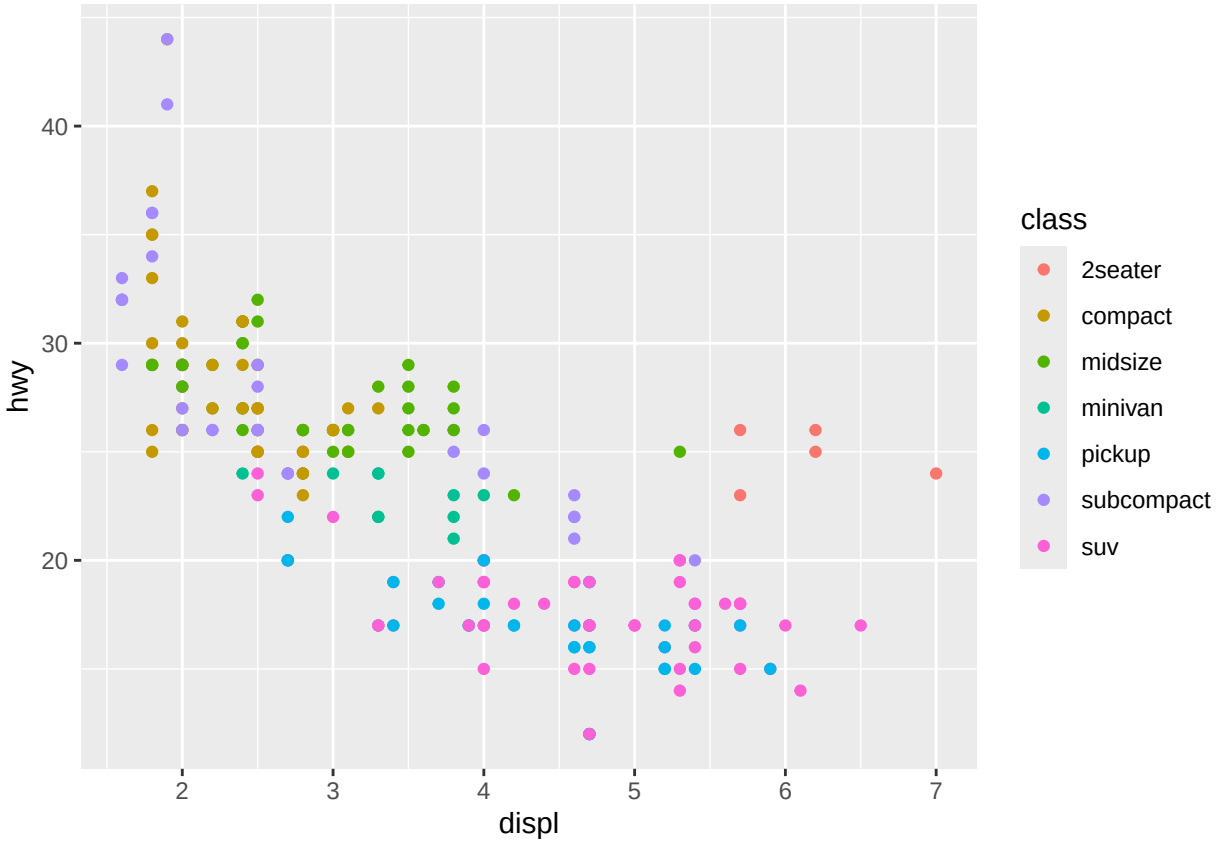
```
##  $ cty          : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy          : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl           : chr [1:234] "p" "p" "p" "p" ...
##  $ class        : chr [1:234] "compact" "compact" "compact" "compact" ...
glimpse(mpg) # get a glimpse of the mpg data
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "~
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, ~
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto~
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4~
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p~
## $ class        <chr> "compact", "compact", "compact", "compact", "compact", "c~
```

The complete graphing template in ggplot2

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,         # optional
     position = <POSITION> # optional
  ) +
  <COORDINATE_FUNCTION> +  # optional
  <FACET_FUNCTION> +       # optional
  <SCALE_FUNCTION> +       # optional
  <THEME_FUNCTION>         # optional
```

Example: aesthetic mappings for `geom_point()`

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, col = class))
```

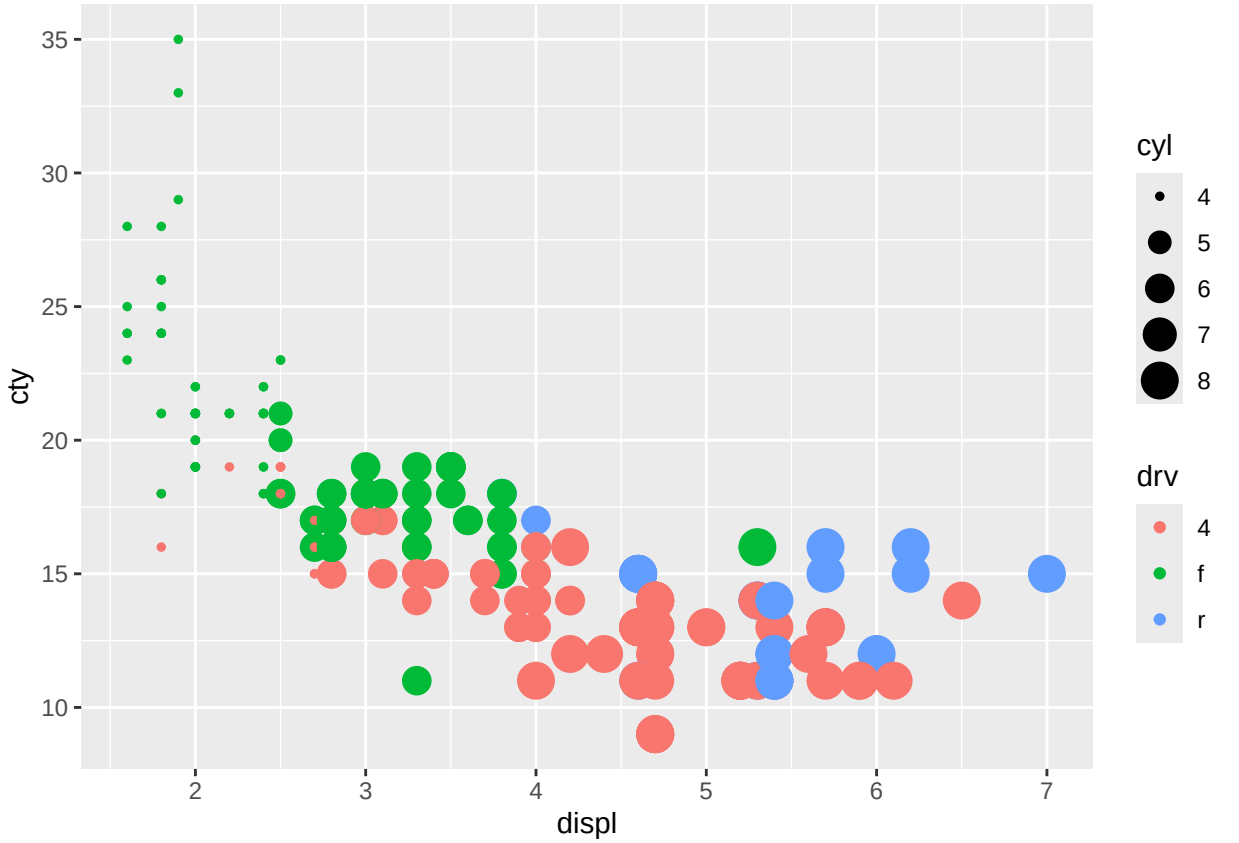


Exercise #2

Write your own `ggplot2` code to make a scatterplot of `cty` (y-axis) against `displ` (x-axis); map `drv` to the color aesthetic; and map `cyl` to the `size` aesthetic.

INSERT_YOUR_ANSWER

```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = cty, color = drv, size = cyl))
```



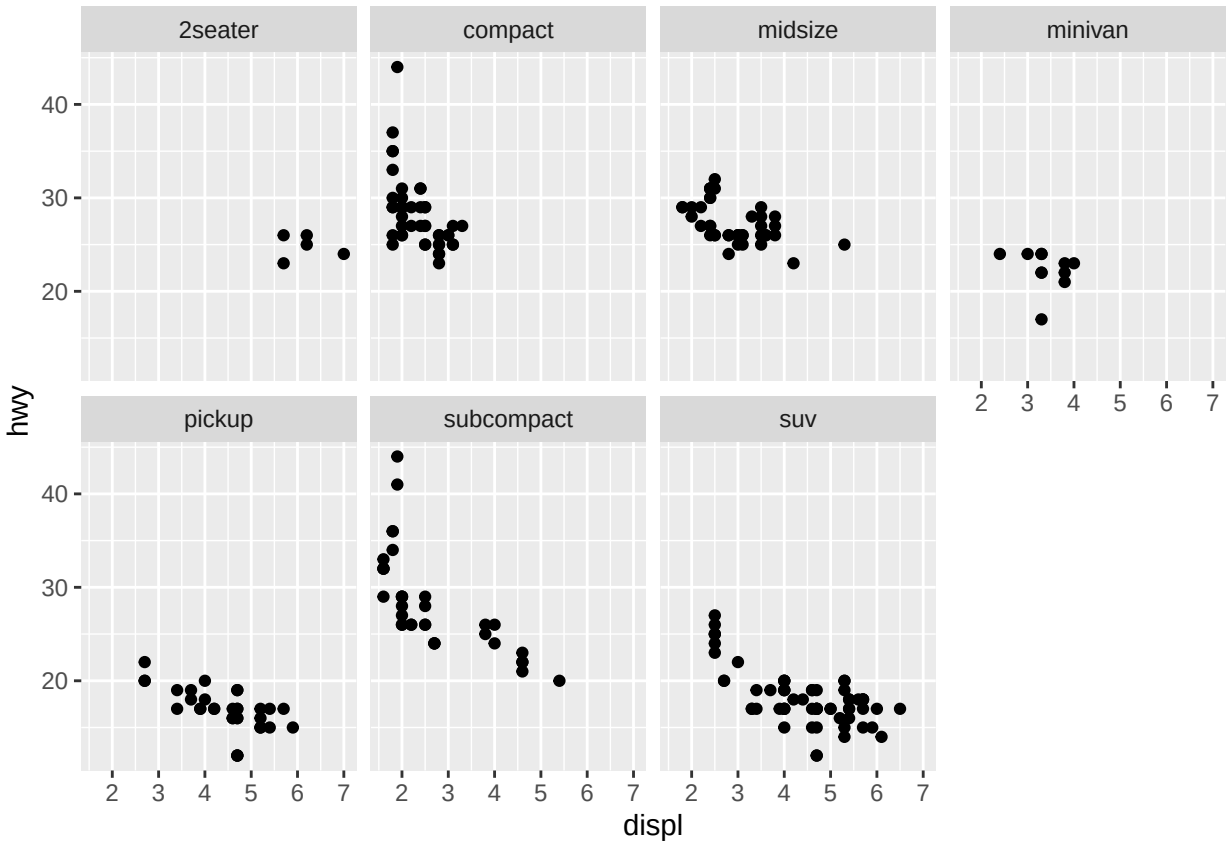
What do you observe from your plot? Briefly describe any patterns or relationships you notice.

INSERT_YOUR_ANSWER As `displ` increases, `cty` tends to decreases. Cars with different `drv` types are clustered with variation, and vehicles with more cyls seem to have a lower `cty' value. Thesizeaesthetic displays that high`cyl‘ cars typically have a larger engine displacement in addition to lower fuel efficiency.

Facets - making subplots that each display one subset of the data.

Facets by a single variable - `facet_wrap()`

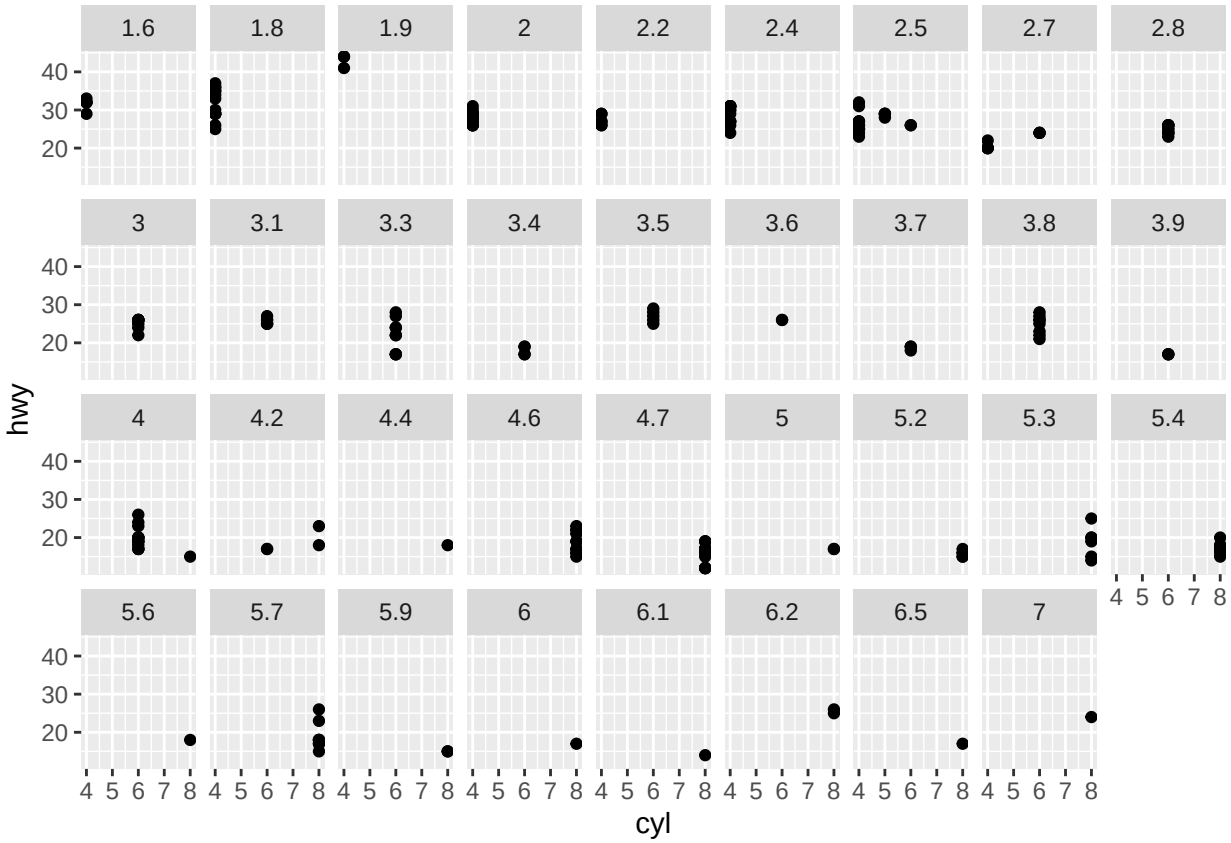
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



Exercise #3

Look at the following code and output figure. What happens if you facet on a continuous variable?

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cyl, y = hwy)) +
  facet_wrap(~ displ, nrow = 4)
```



INSERT_YOUR_ANSWER Faceting by continuous variables creates a singular panel per unique value of the given variable. This can result in a large number of subplots with very few information, or points, in each. This is not ideal, and may cause a report to lack visual sufficiency.
