Practically, `ggplot2` provides beautiful, hassle-free plots that take care of fiddly details like drawing legends. The plots can be built up iteratively and edited later.

— Hadley Wickham (Wickham, 2016)

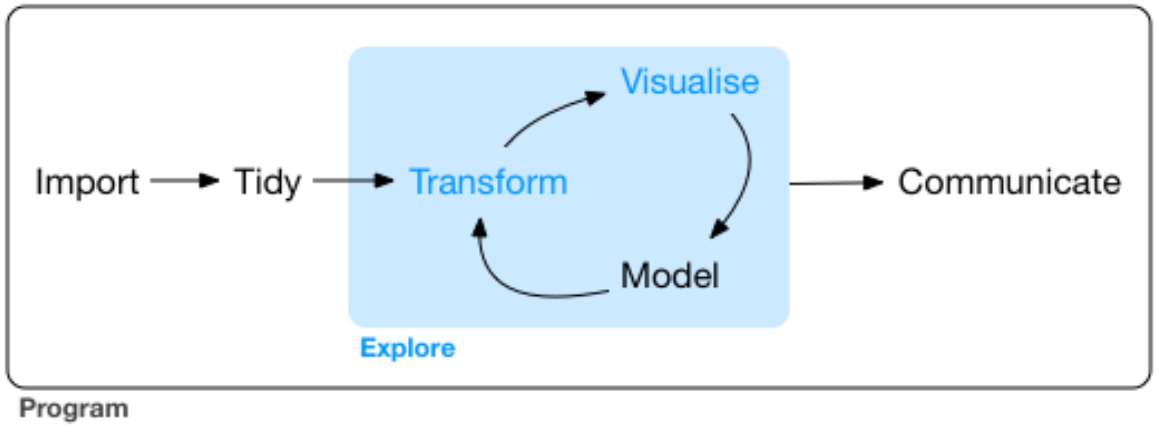
Data Science for Operational Researchers using R

05 – ggplot2

https://github.com/JimDuggan/Data-Science-for-OR

Exploratory Data Aanalysis: Goal

- Develop an understanding of your data
- Use questions as tools to guide your investigation
- A creative process “the key to asking quality questions is to generate a large quantity of questions”
- Useful to explore variation in variables



“Data exploration is the art of looking at your data, rapidly generating hypotheses, quickly testing them, then repeating again and again and again.” (Wickham and Grolemund 2017).

Defining Important Terms (Wickham and Grolemund)

Term	Explanation
Variable	A quantity, quality or property that you can measure
Value	The state of a variable when you measure it. The value of a variable may change from measurement to measurement
Observation (or case)	A set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object). An observation will contain several values, each associated with a different variable.
Tabular data	A set of values, each associated with a variable and an observation. Tabular data is tidy if each value is placed in its own "cell", each variable in its own column, and each observation in its own row. All the data in aimsir17 and ggplot2 is tidy data.

ggplot2

- Layered grammar of graphics, enables us to concisely describe the components of a graphic.
- Benefits:
  - plots can be designed in a layered manner (+ operator)
  - a wide range of plots can be generated to support decision analysis, including scatterplots, histograms and time series charts
  - charts can be developed rapidly, and this support an iterative process of decision support

ggplot2::mpg, N = 234

Variable	Description
manufacturer	Manufacturer name
model	Model name
displ	Engine displacement (litres)
year	Year of manufacture
cyl	Number of cylinders
trans	Type of transmission
drv	Type of drive train (e.g. front wheel)
cty	City miles per gallon
hwy	Highway miles per gallon
fl	Fuel type
class	"type" of car (e.g. "compact")

ggplot2::diamonds, N = 53,940

Variable	Description
carat	Weight of the diamond
cut	Quality of the cut (categorical)
color	Diamond colour (categorical)
clarity	Diamond clarity (categorical)
depth	Total depth percentage
table	Measure related to width of diamond top
price	Price in dollars
x	Length in mm
y	Width in mm
z	Depth in mm

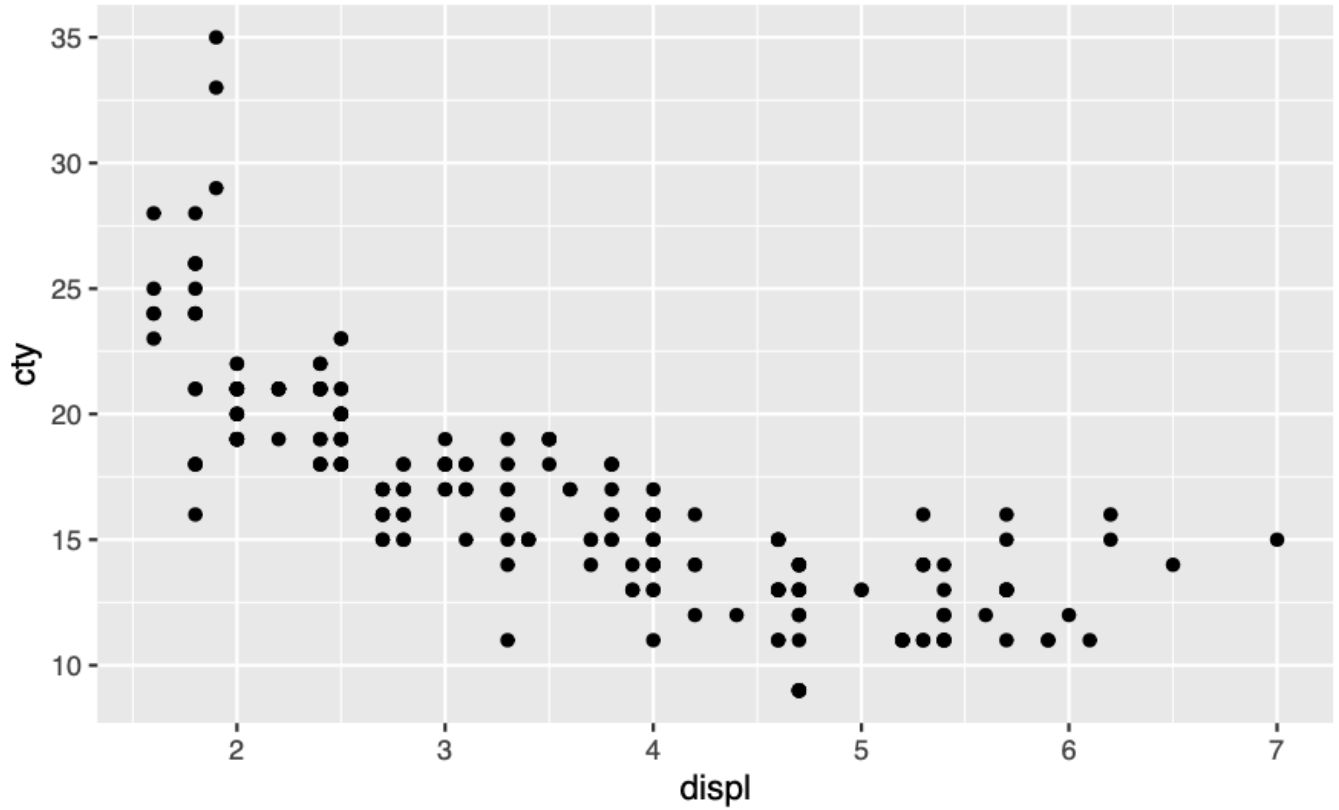
Exploring a tibble

```
library(ggplot2)
library(tibble)
```

```
mpg |> subset(select=c("displ","cty")) |> summary()
#>      displ            cty
#>  Min.   :1.60   Min.   : 9.0
#>  1st Qu.:2.40   1st Qu.:14.0
#>  Median :3.30   Median :17.0
#>  Mean   :3.47   Mean   :16.9
#>  3rd Qu.:4.60   3rd Qu.:19.0
#>  Max.   :7.00   Max.   :35.0
```

Visualising: data + mapping + geometric object

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

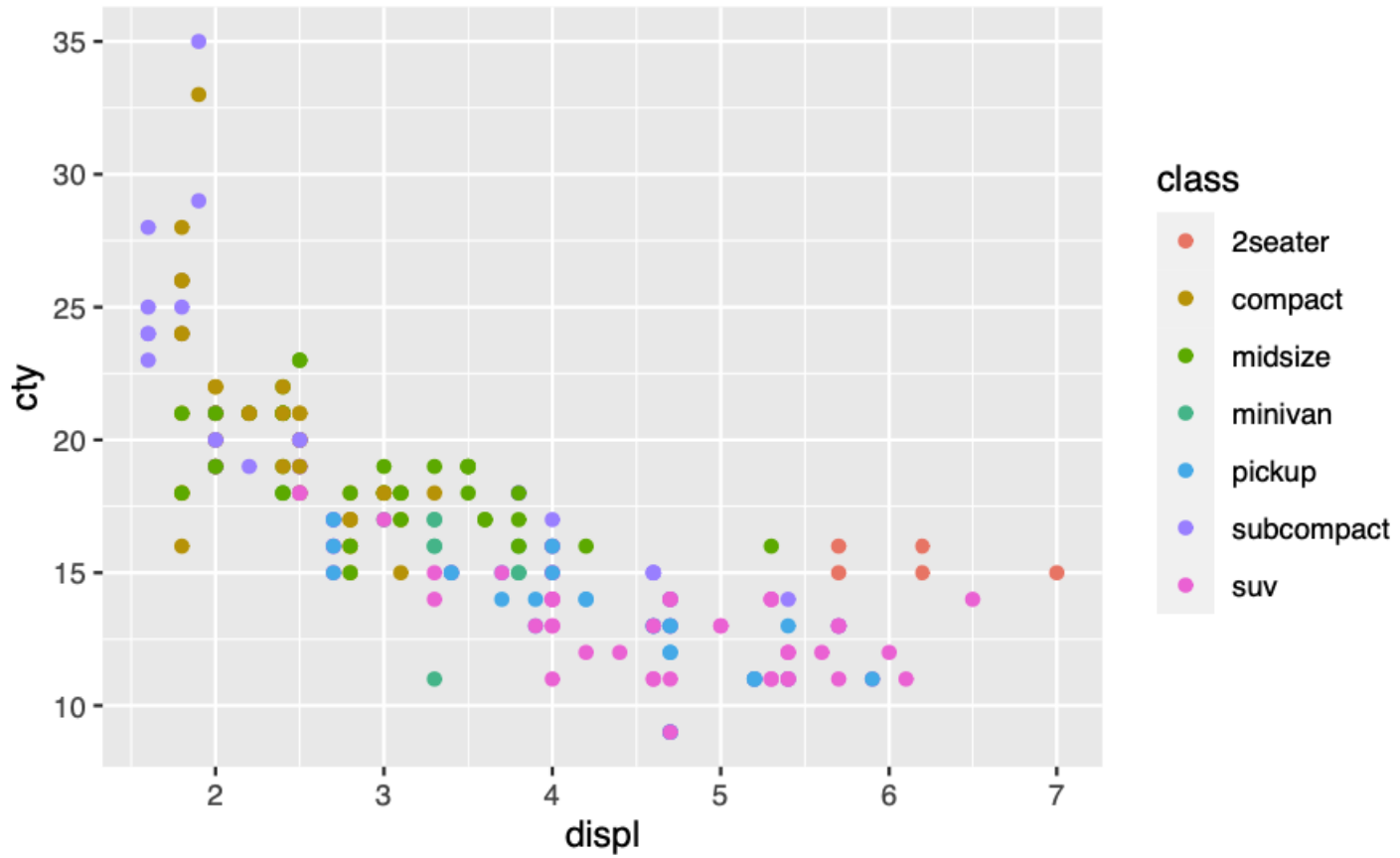


Exploring additional mappings

Seven classes of car…

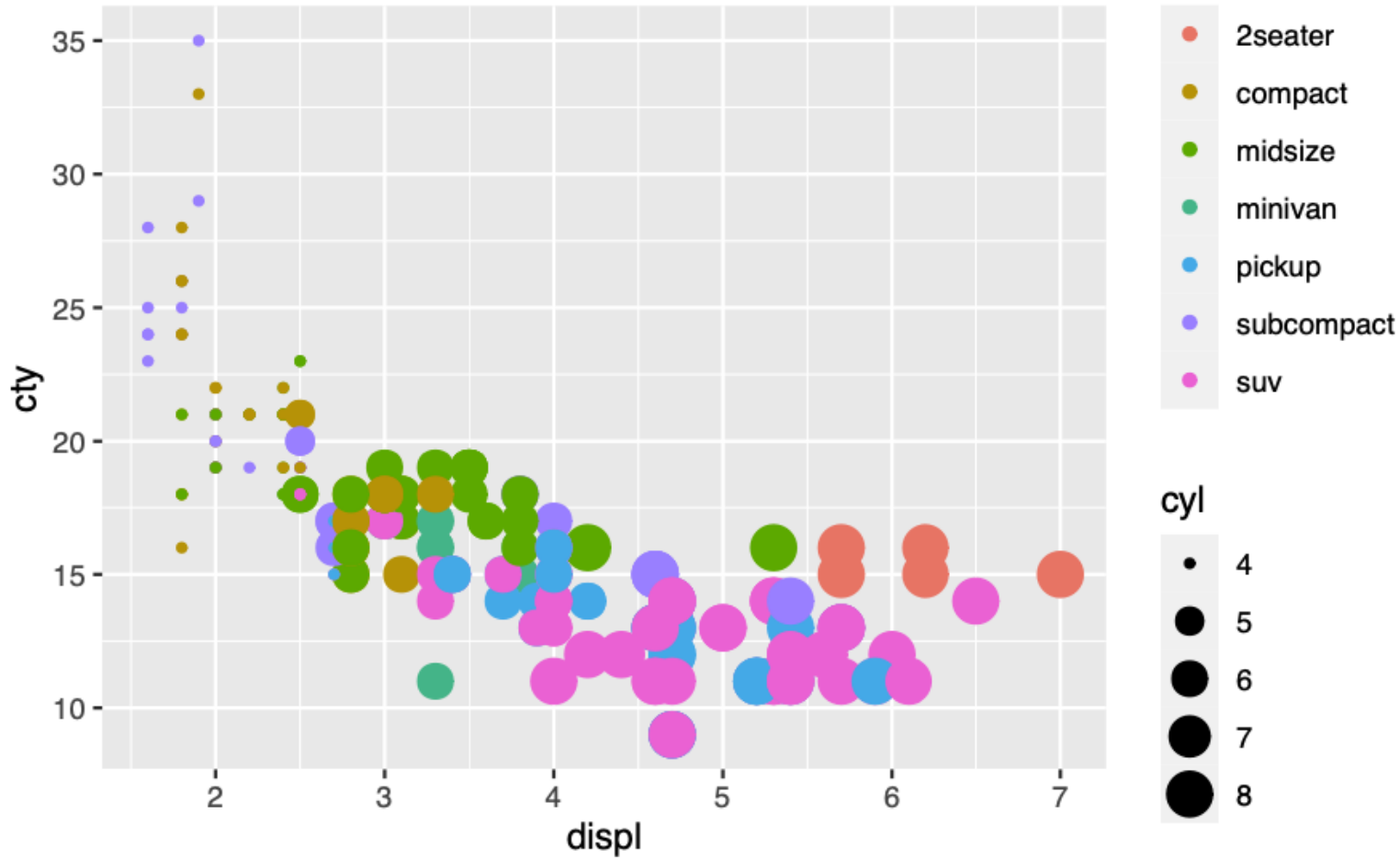
```
unique(mpg$class)
#> [1] "compact"    "midsize"    "suv"        "2seater"    "minivan"
#> [6] "pickup"     "subcompact"
```

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




```
unique(mpg$class)
#> [1] "compact"    "midsize"    "suv"        "2seater"    "minivan"
#> [6] "pickup"     "subcompact"
```

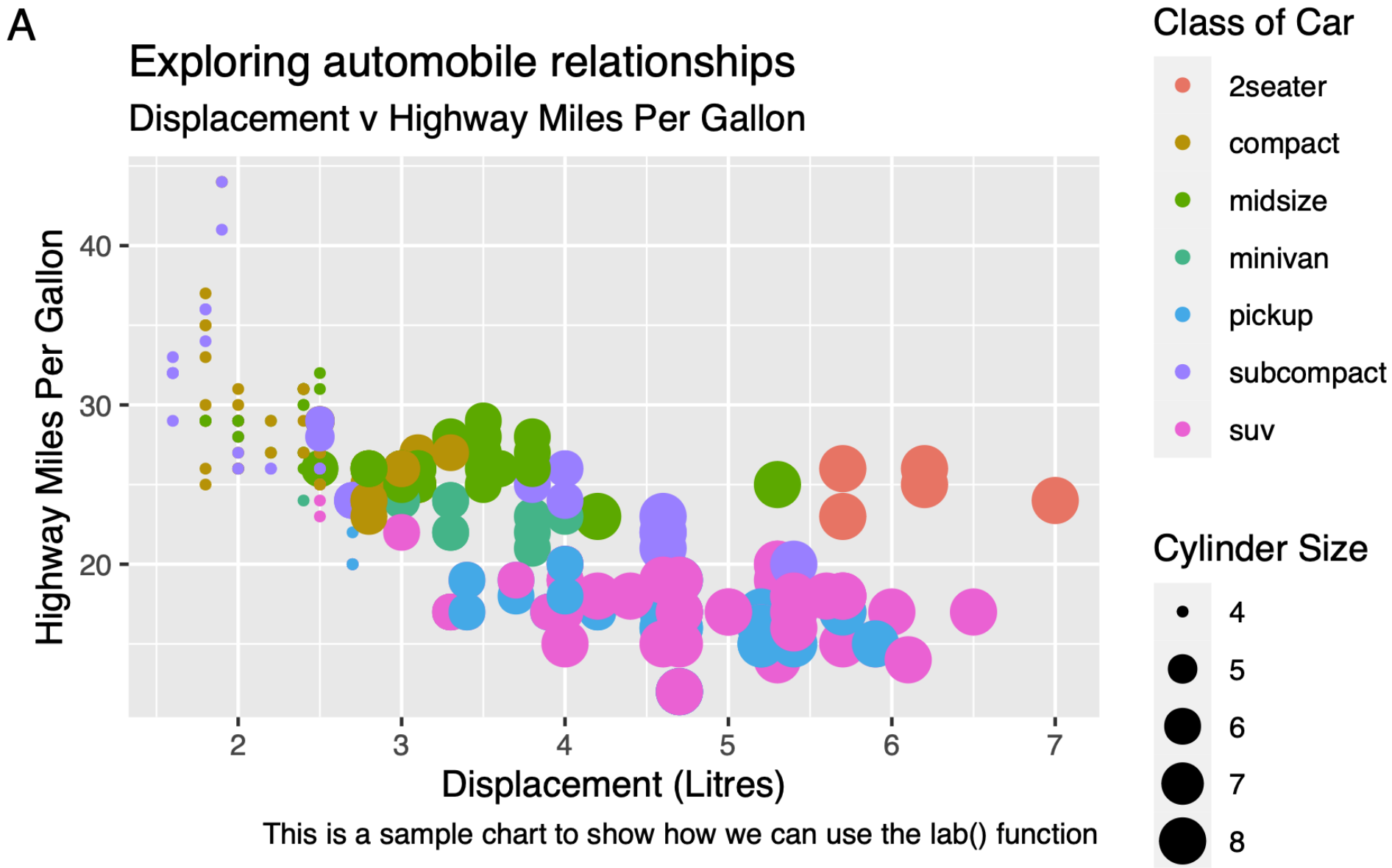
```
ggplot(data=mpg,mapping=aes(x=displ,y=cty,colour=class,size=cyl))+
  geom_point()
```



Customising Plot Appearance with lab()

Argument	Role
title	provides an overall title text for the plot
subtitle	adds a subtitle text
colour	allows you to specify the legend name for the colour attribute
caption	inserts text on the lower right hand side of your plot
size	allows you to name the size attribute
x	name the x-axis
y	name the y-axis
tag	text for tag label for top-left of plot

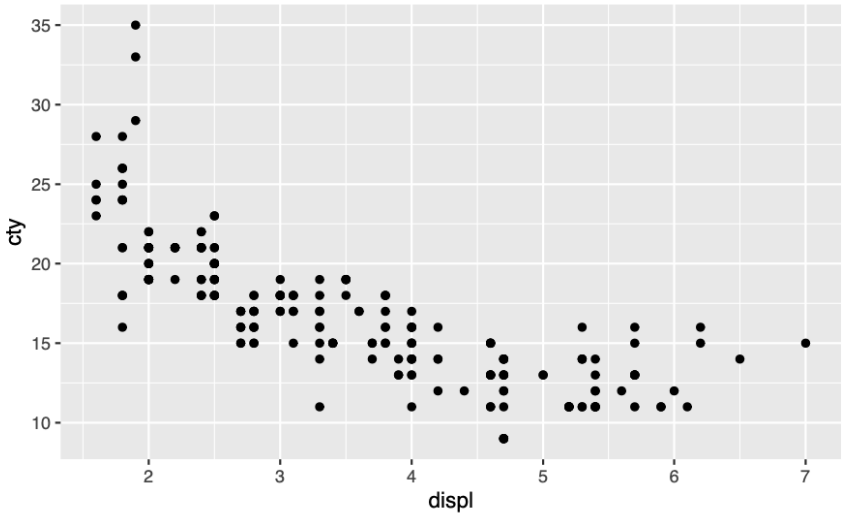
```
p1 <- ggplot(data=mpg,aes(x=displ,y=hwy,size=cyl,colour=class))+
        geom_point()

p1 <- p1 +
       labs(
         title = "Exploring automobile relationships",
         subtitle = "Displacement v Highway Miles Per Gallon",
         colour = "Class of Car",
         size = "Cylinder Size",
         caption = "This is a sample chart to show how we can use the lab() function",
         tag = "A",
         x = "Displacement (Litres)",
         y = "Highway Miles Per Gallon"
       )

p1
```


A
Exploring automobile relationships
Displacement v Highway Miles Per Gallon
Highway Miles Per Gallon
40
30
20
2
3
4
5
6
7
Displacement (Litres)
This is a sample chart to show how we can use the lab() function
Class of Car
2seater
compact
midsize
minivan
pickup
subcompact
suv
Cylinder Size
4
5
6
7
8

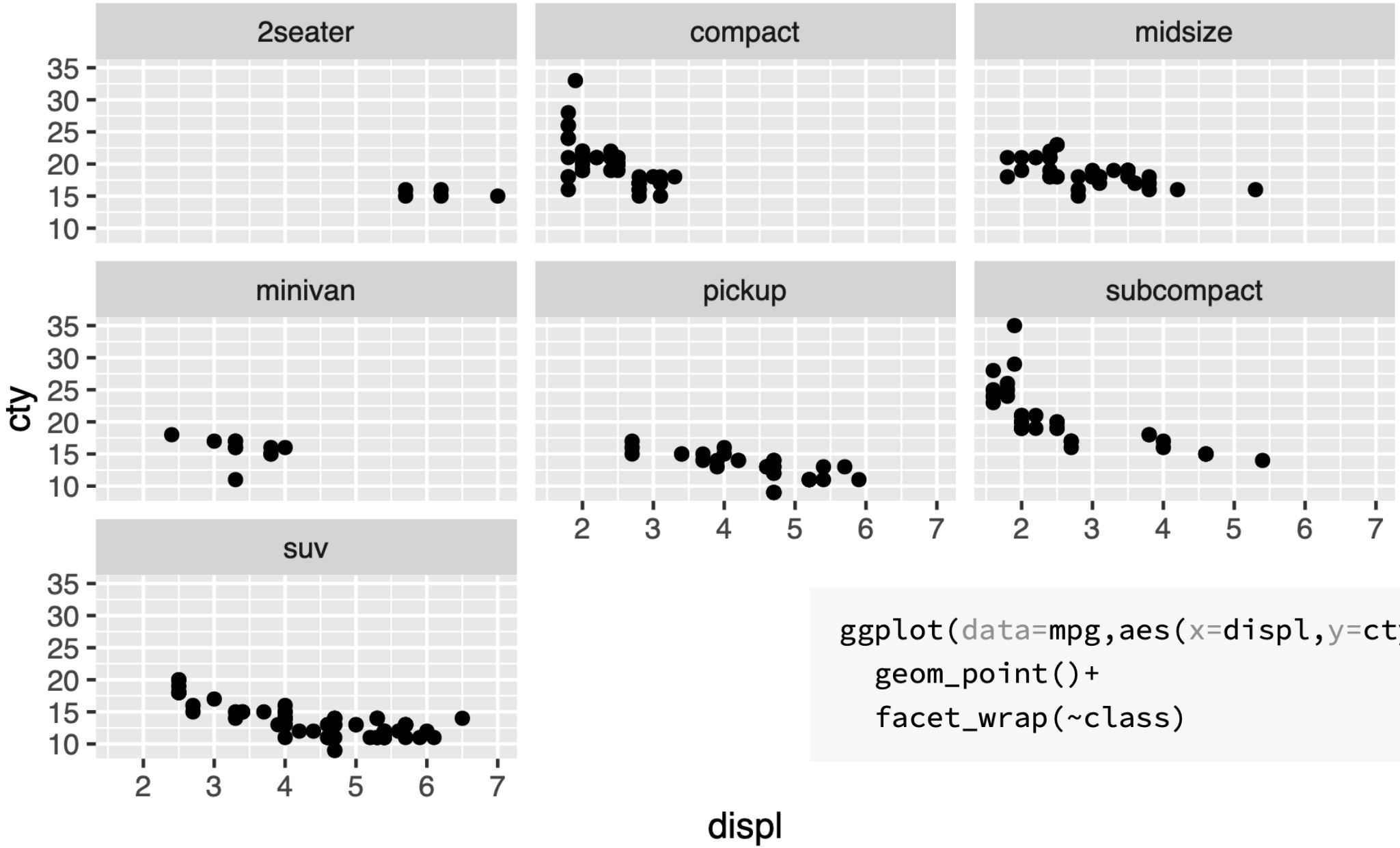
Subplots with Facets

```
unique(mpg$class)
#> [1] "compact"    "midsize"    "suv"        "2seater"    "minivan"
#> [6] "pickup"     "subcompact"
```

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

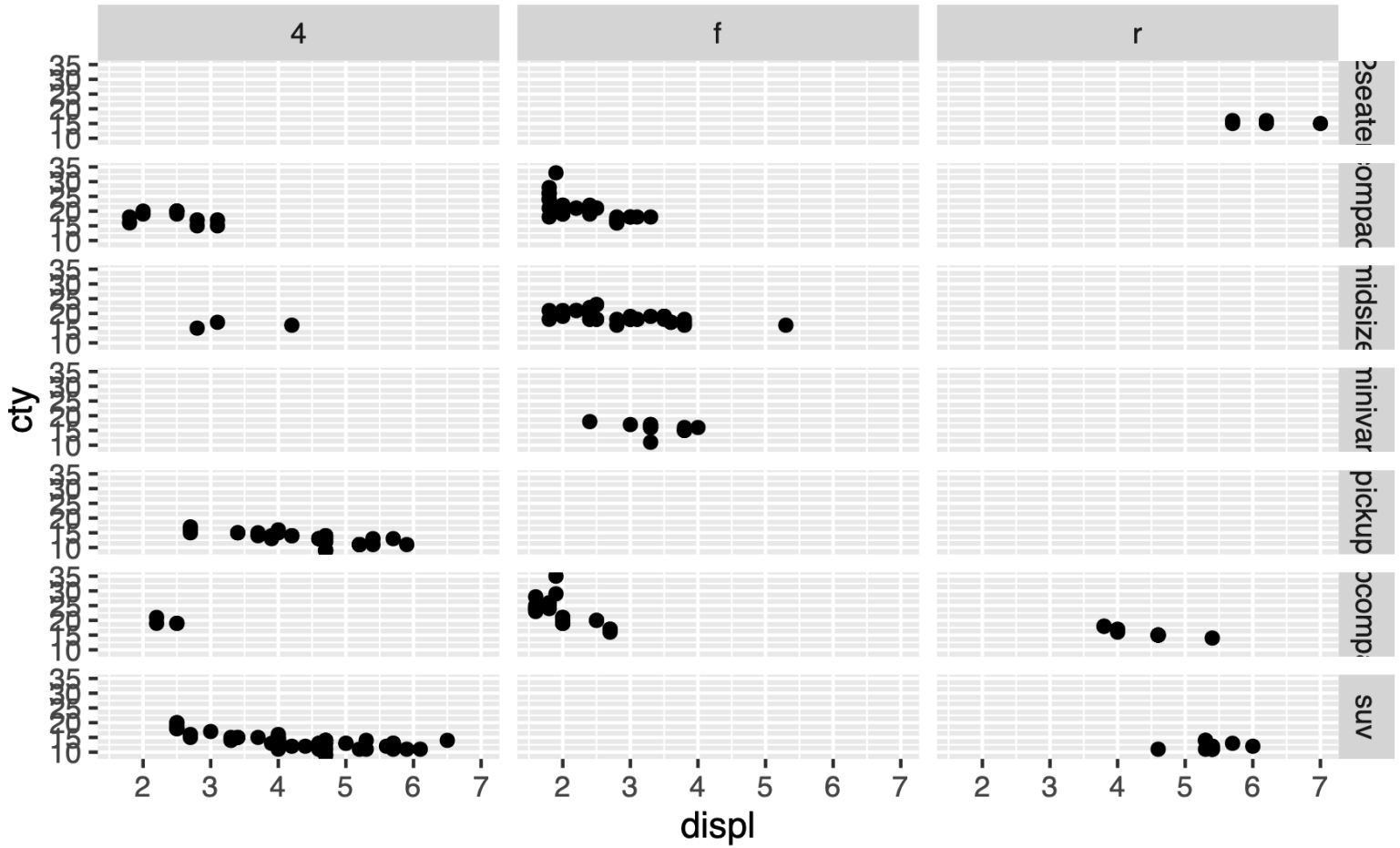



```
ggplot(data=mpg,aes(x=displ,y=cty))+
  geom_point()+
  facet_wrap(~class)
```



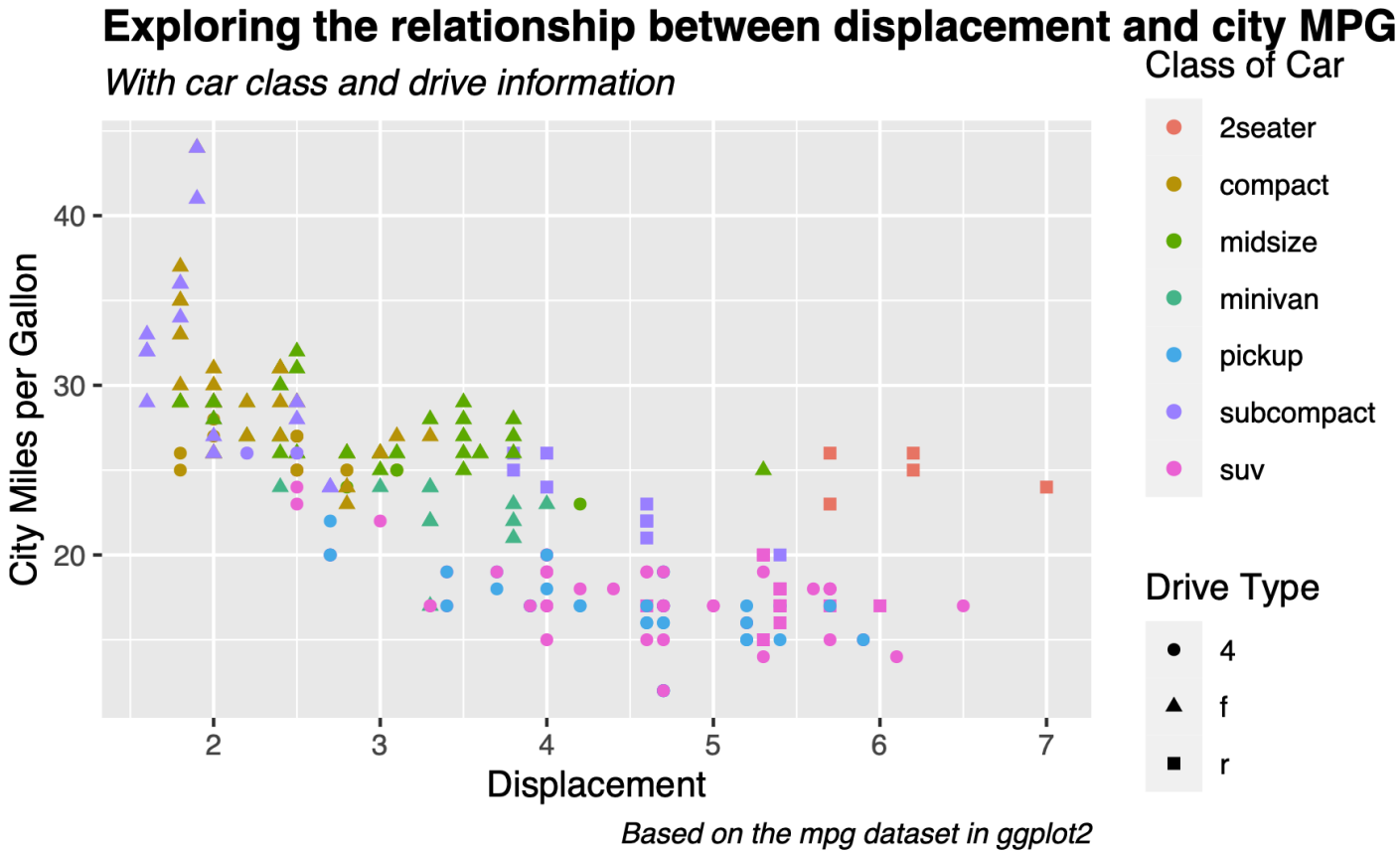

```
ggplot(data=mpg,aes(x=displ,y=cty))+
  geom_point()+
  facet_wrap(~class)
```

```
ggplot(data=mpg,mapping = aes(x=displ,y=cty))+
  geom_point()+
  facet_grid(class~drv)
```



Challenge 5.1

Generate the following plot from the `mpg` tibble in `ggplot2`. The x-variable is `displ` and the y-variable `cty`. Make use of the `lab()` and `theme()` functions.

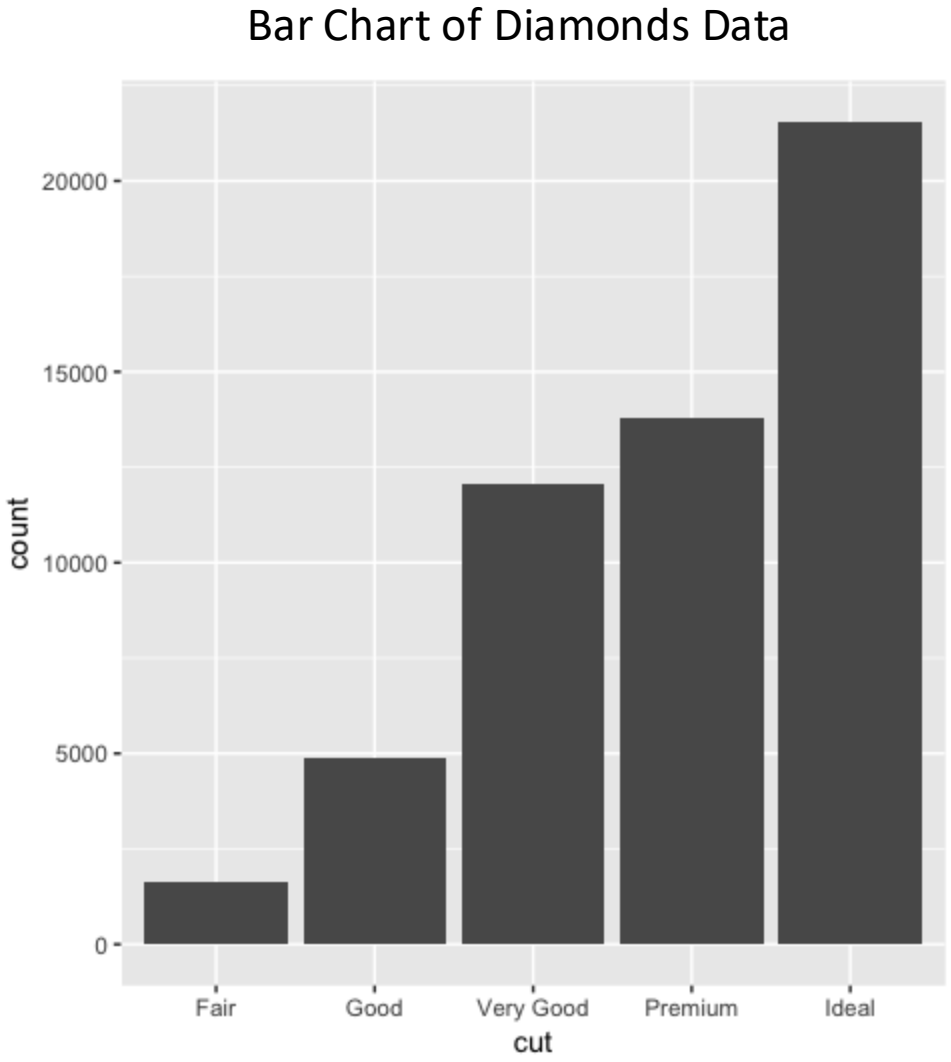


Statistical Transformations

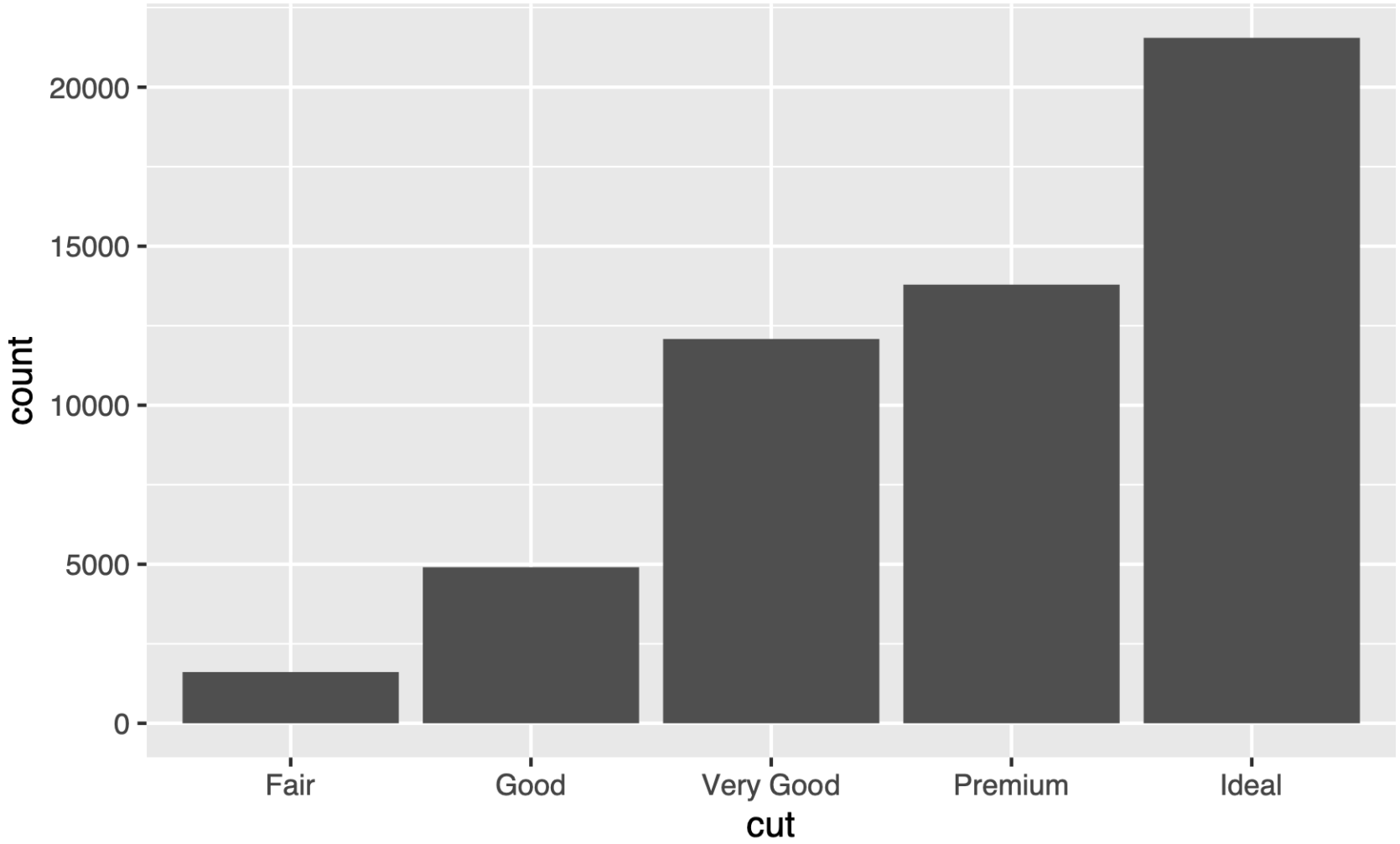
- Many graphs, like scatterplots, plot the raw values of the dataset
- However, other graphs (e.g. bar charts) *calculate new values to plot*
  - **Bar charts, histograms and frequency polygons** bin your data and plot bin counts, the number of points that fall in each bin
  - **Smoothers** fit a model to your data and the plot predictions from the model
  - **Boxplots** compute a robust summary of the distribution and display a specially formatted box

Variation

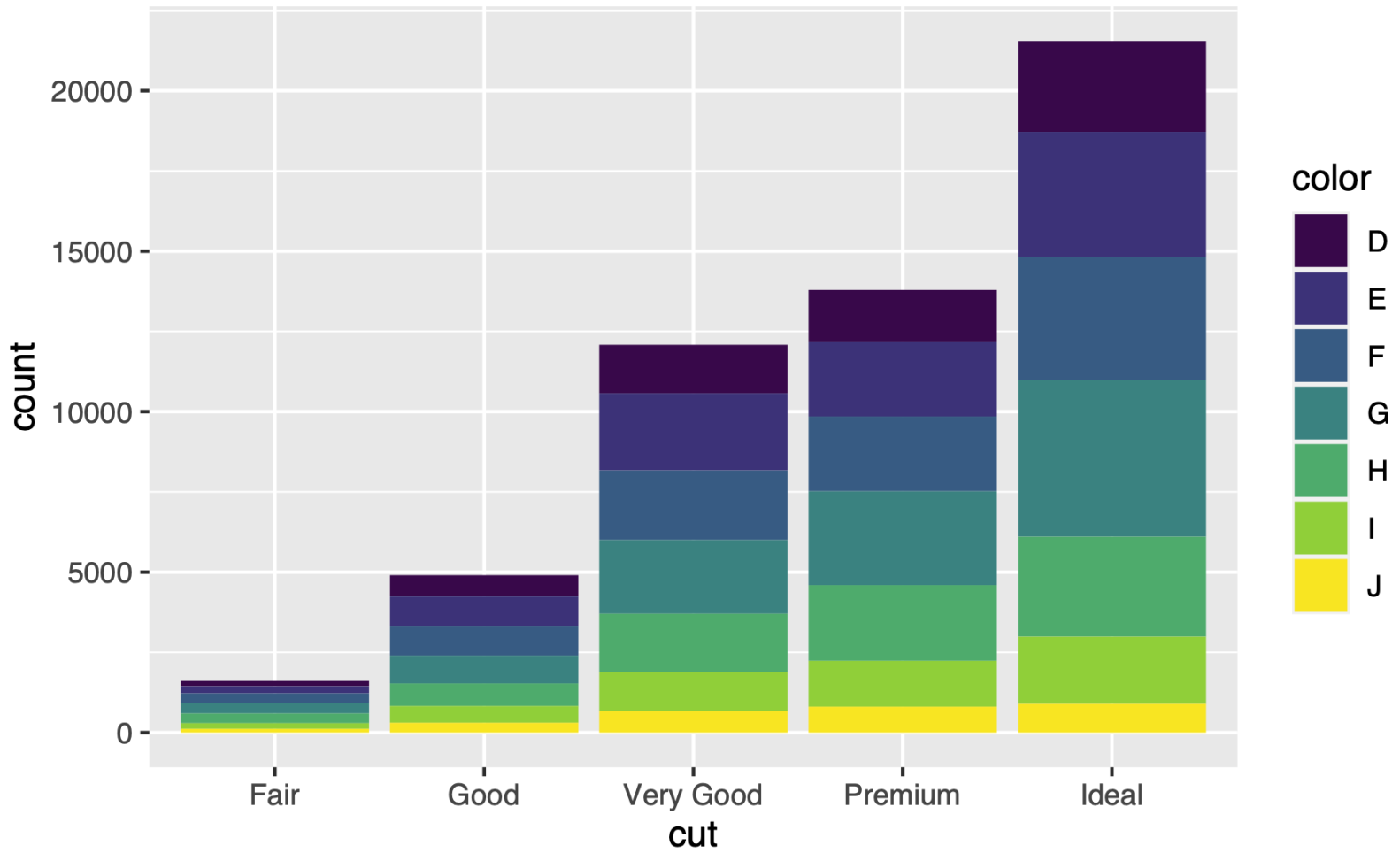
- ***Variation*** is the tendency of the values of a variable to change from measurement to measurement
- If you measure a continuous variable twice, you can get two different results
  - For example, temperature at 12.00 and temperature at 13.00
- Very variable has its own pattern of variation
- The best way to understand the pattern is to *visualize the distribution* of a variable's values.



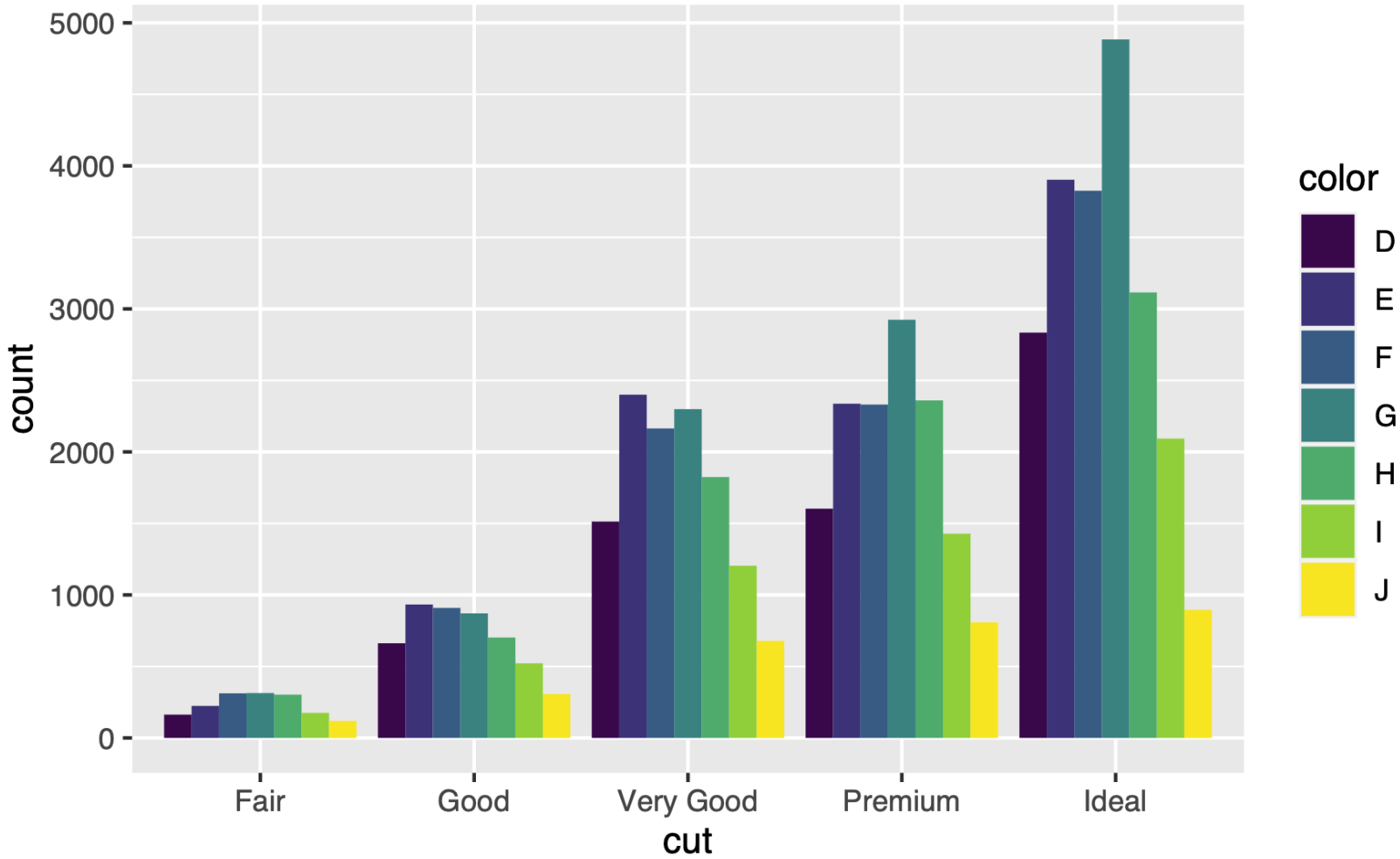
```
ggplot(data=diamonds,mapping=aes(x=cut))+
  geom_bar()
```



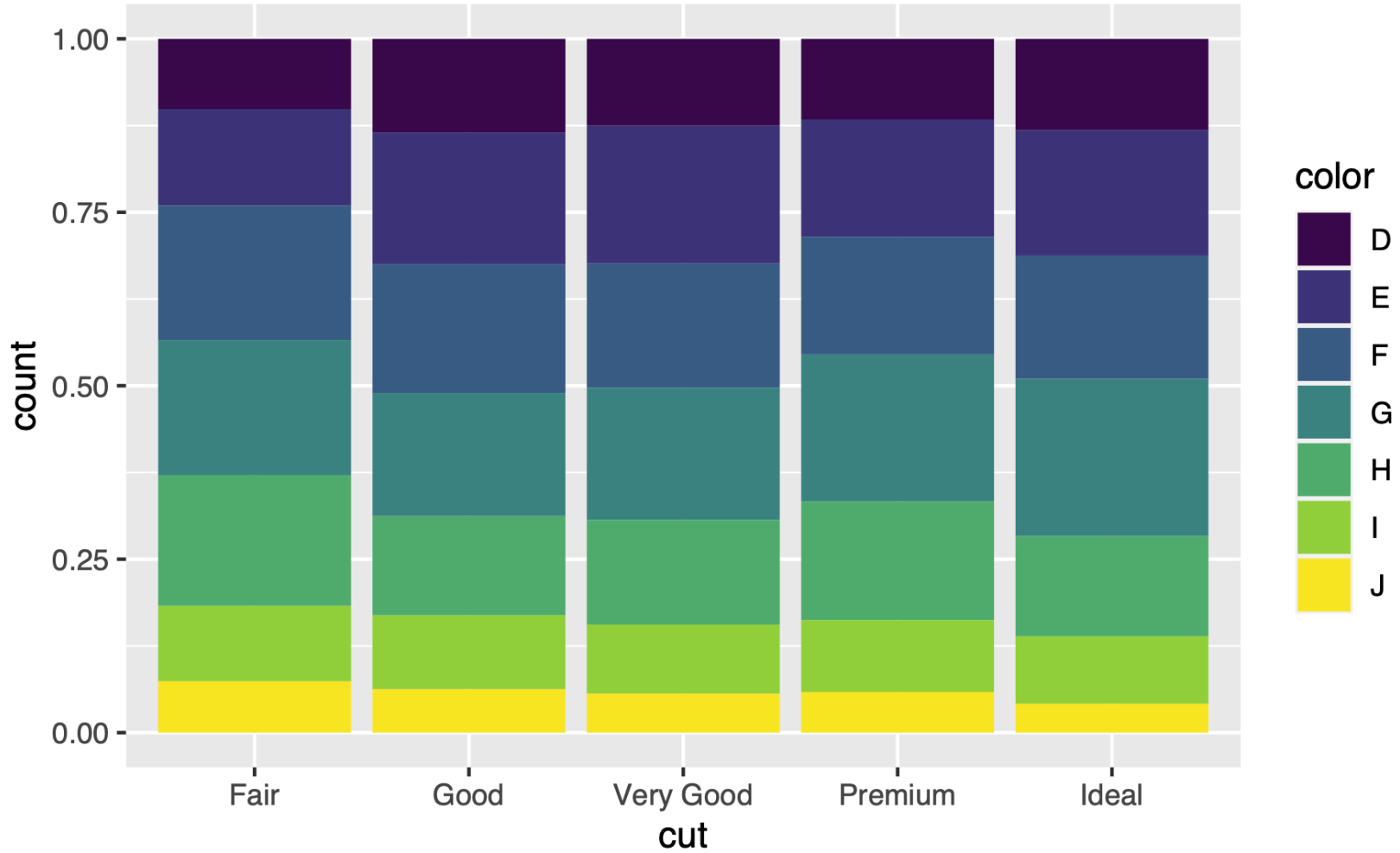
```
ggplot(data=diamonds,mapping=aes(x=cut,fill=color))+
  geom_bar()
```



```
ggplot(data=diamonds,mapping=aes(x=cut,fill=color))+
  geom_bar(position="dodge")
```



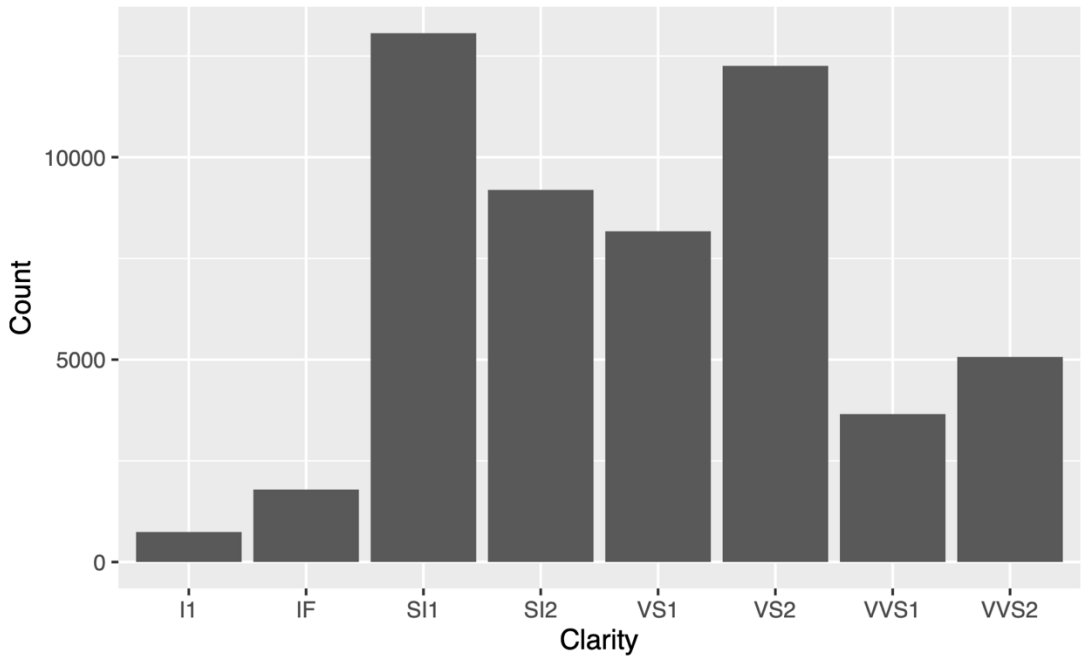
```
ggplot(data=diamonds,mapping=aes(x=cut,fill=color))+
  geom_bar(position="fill")
```



Bar chart without transformations

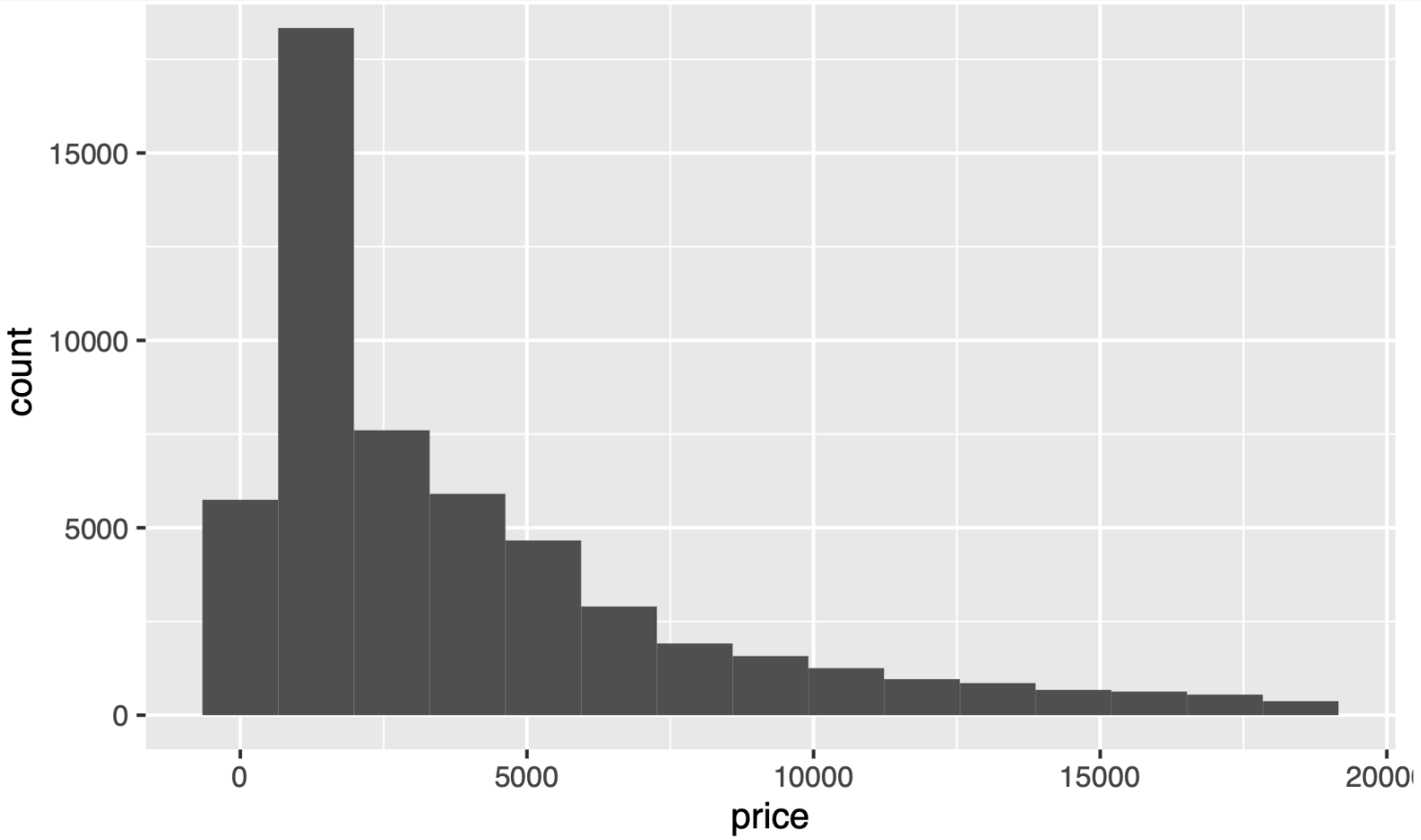
```
cl_cat   <- unique(as.character(diamonds$clarity))
cl_count <- unlist(lapply(cl_cat,function(x)sum(x==diamonds$clarity)))
cl_res   <- tibble(Clarity=cl_cat, Count=cl_count)
cl_res
#> # A tibble: 8 x 2
#>   Clarity Count
#>   <chr>   <int>
#> 1 SI2      9194
#> 2 SI1     13065
#> 3 VS1      8171
#> 4 VS2     12258
#> 5 VVS2     5066
#> 6 VVS1     3655
#> 7 I1        741
#> 8 IF       1790
```

```
ggplot(data=cl_res,aes(x=Clarity,y=Count))+
  geom_bar(stat="identity")
```

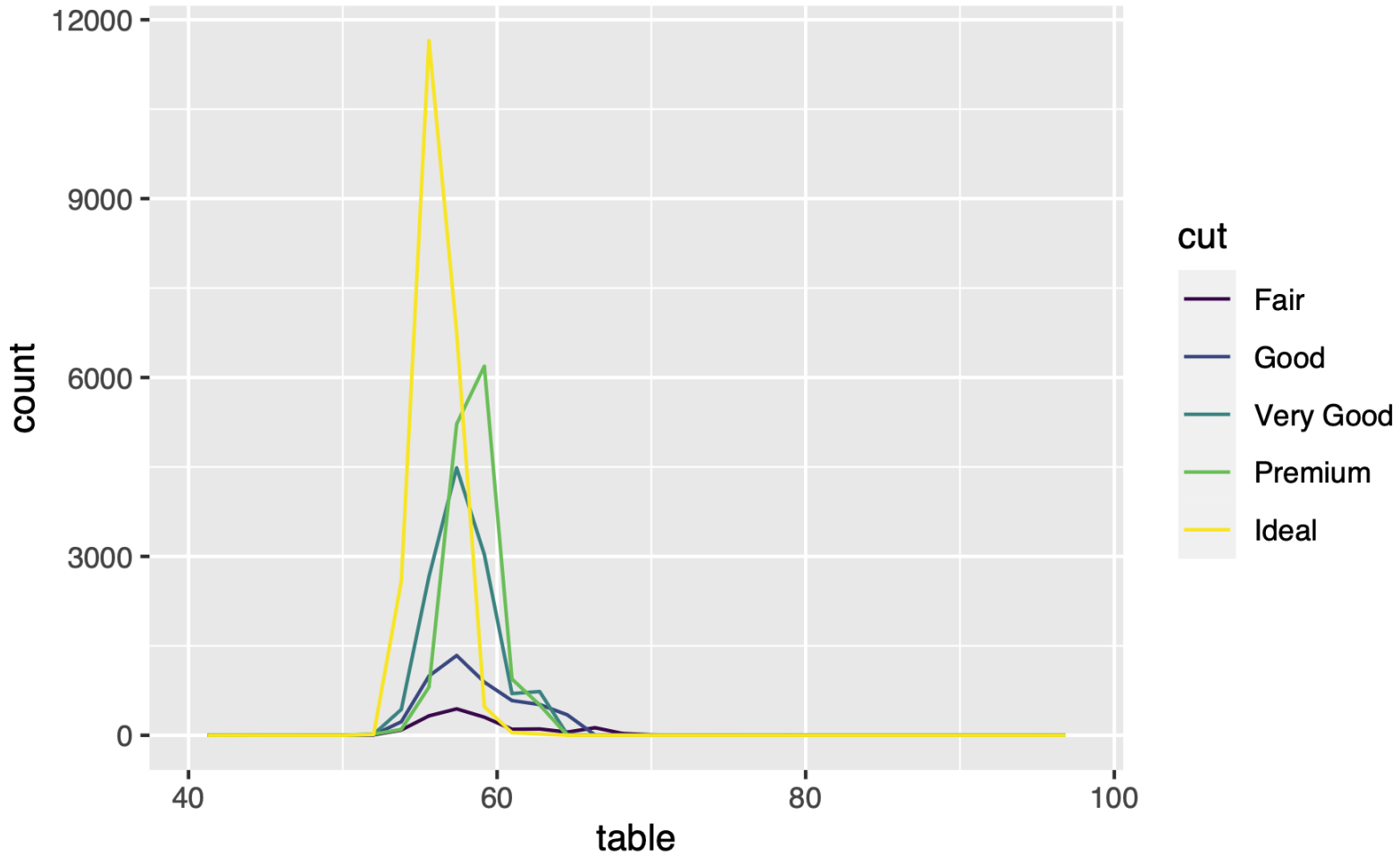


Visualising distributions (continuous variable)

```
ggplot(data=diamonds,mapping=aes(x=price))+
  geom_histogram(bins = 15)
```

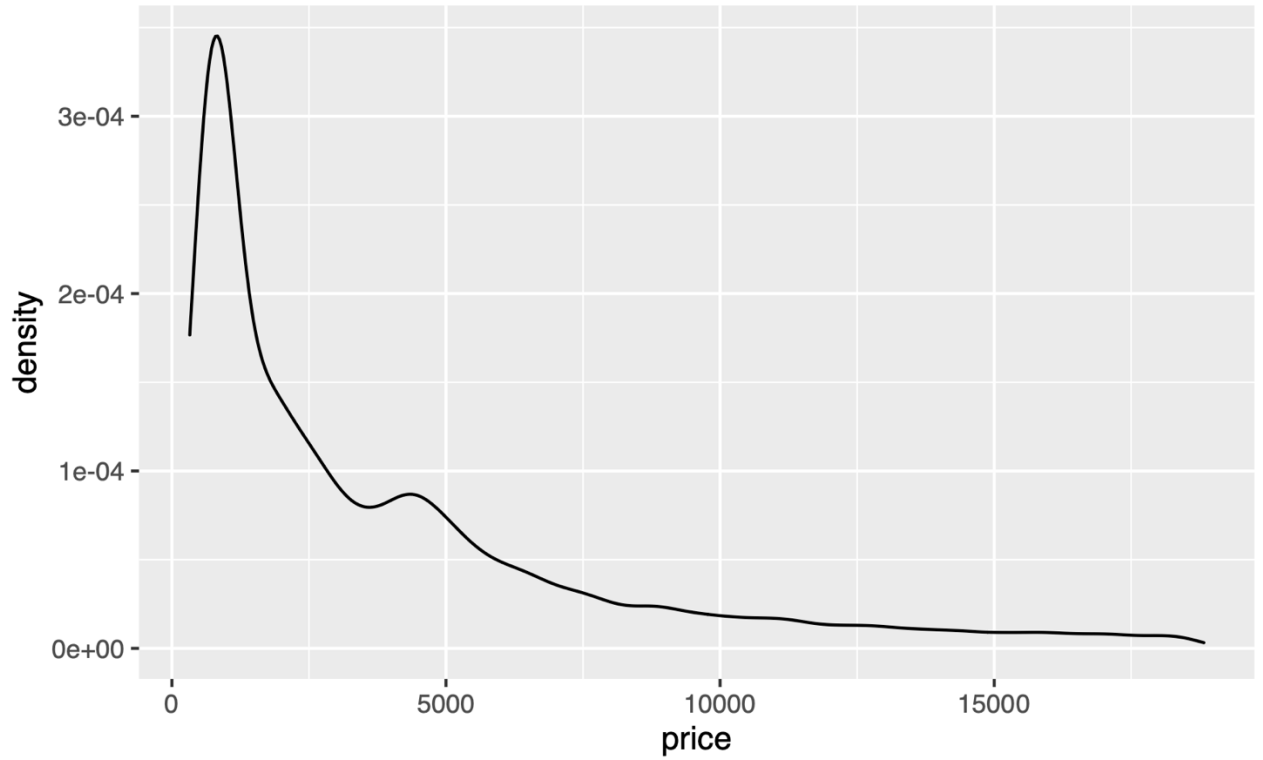


```
ggplot(data=diamonds,mapping=aes(x=table,colour=cut))+
  geom_freqpoly()
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



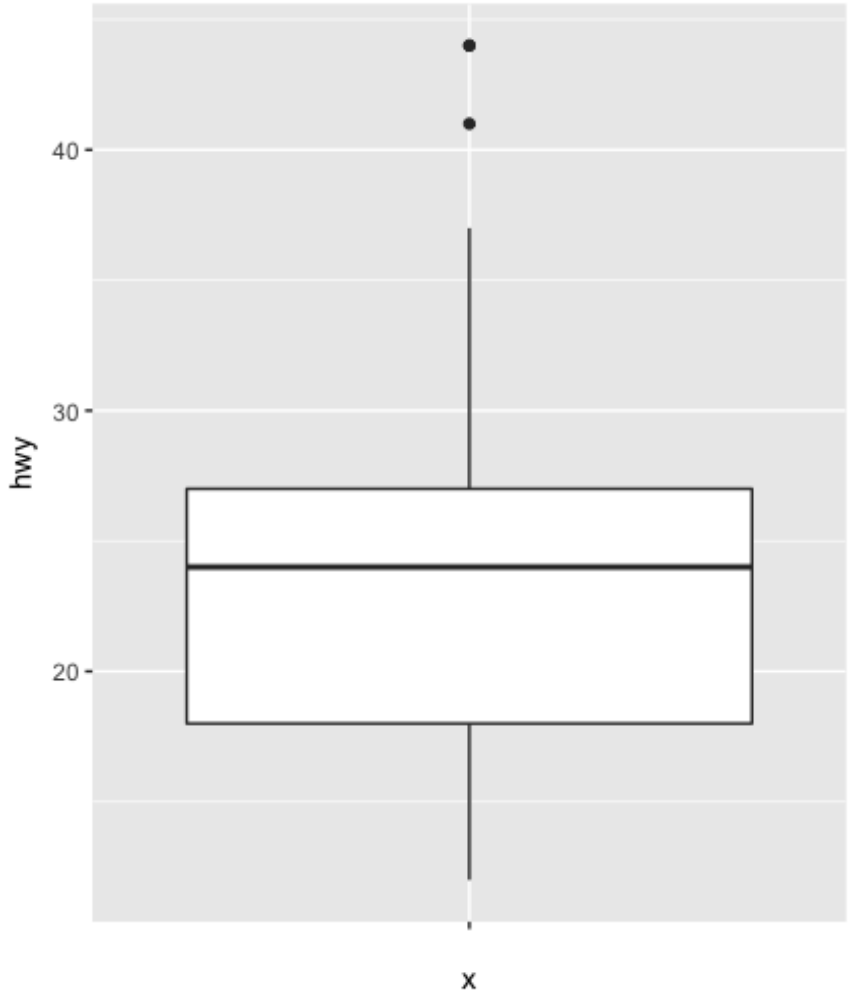
Kernel density estimates

```
ggplot(data=diamonds,mapping=aes(x=price))+
  geom_density()
```

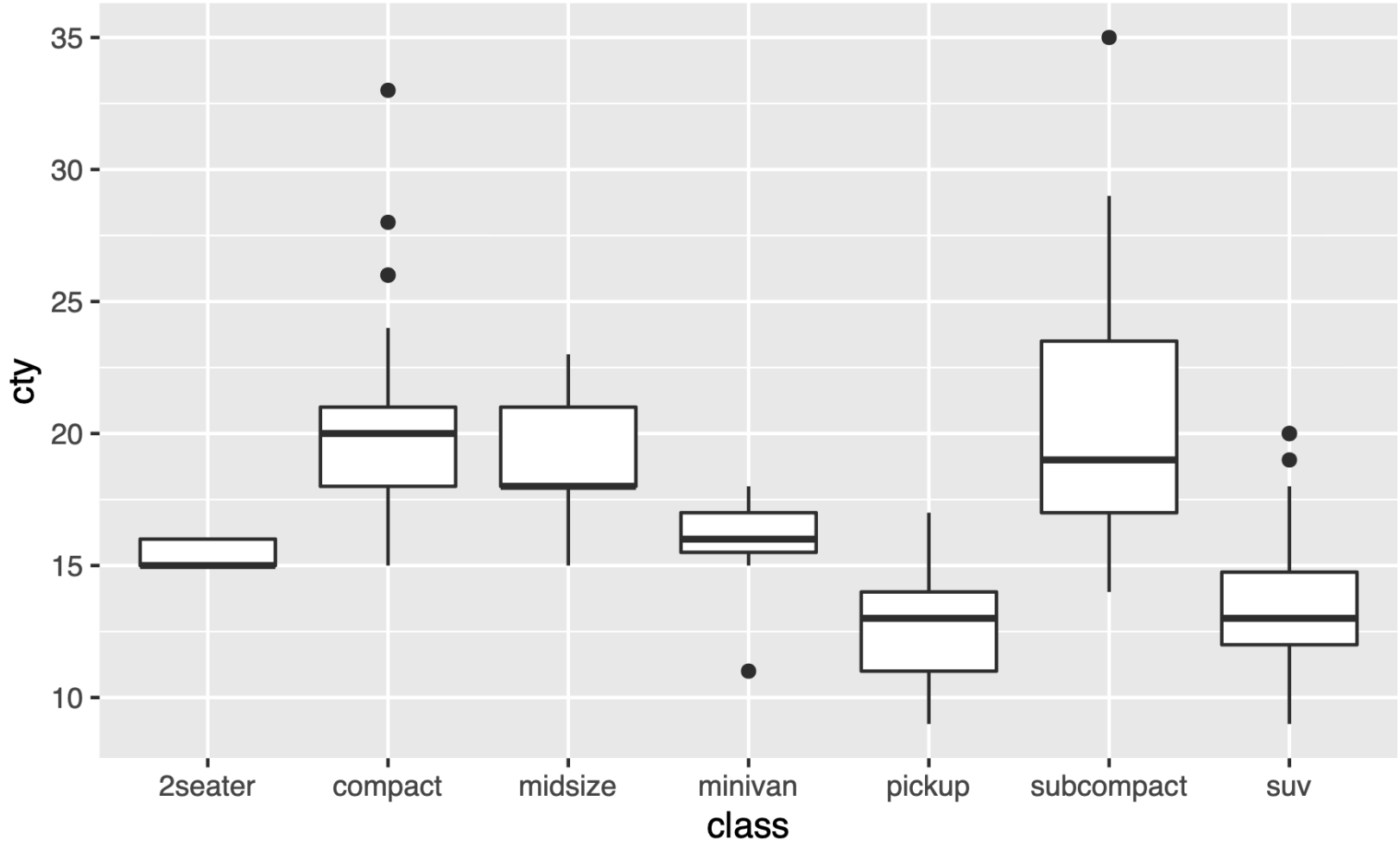


Boxplot

- Display the distribution of a continuous variable broken down by a categorical variable
- Box that stretches from the 25th to 75th percentile a distance known as the interquartile range (IRQ)
- Median in the middle of box
- Points outside more that 1.5 times the IQR from either edge of the box are displayed (outliers)
- Whisker extends to the farthest non-outlier point in the distribution

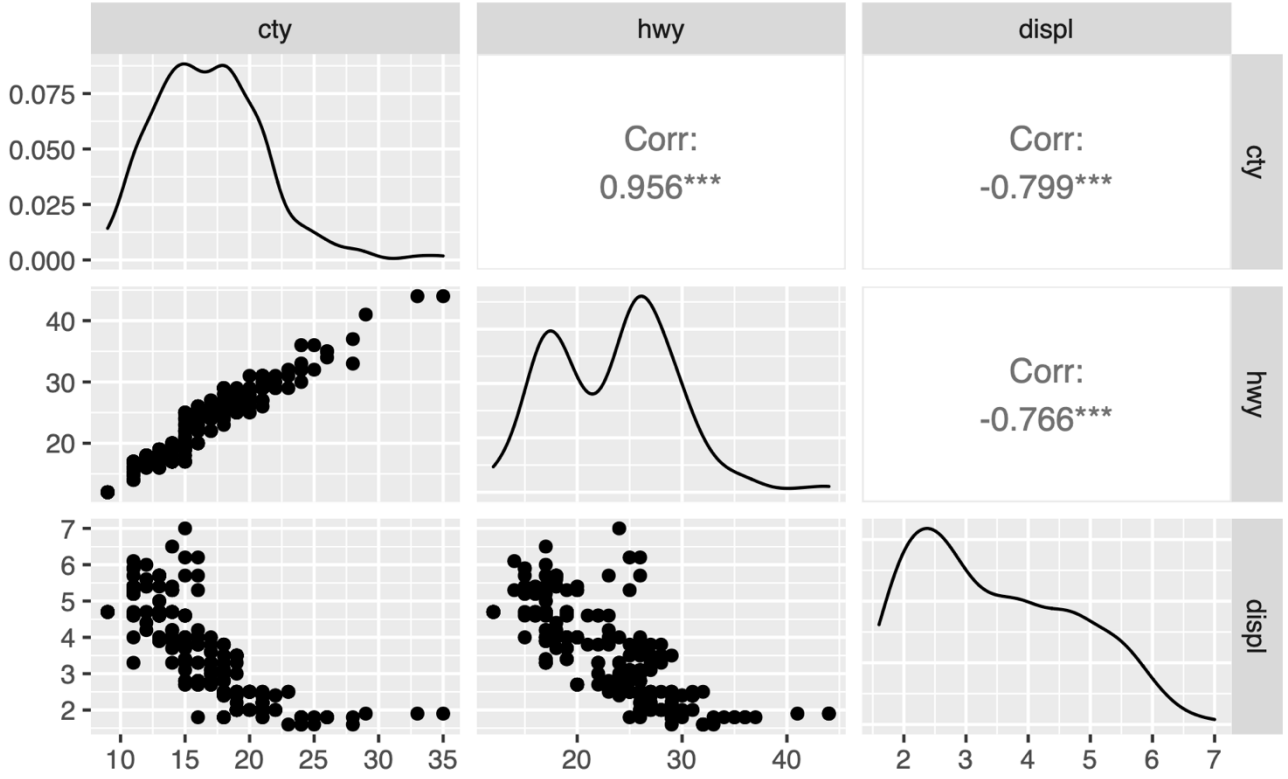


```
ggplot(data=mpg,mapping=aes(y=cty,x=class))+
  geom_boxplot()
```



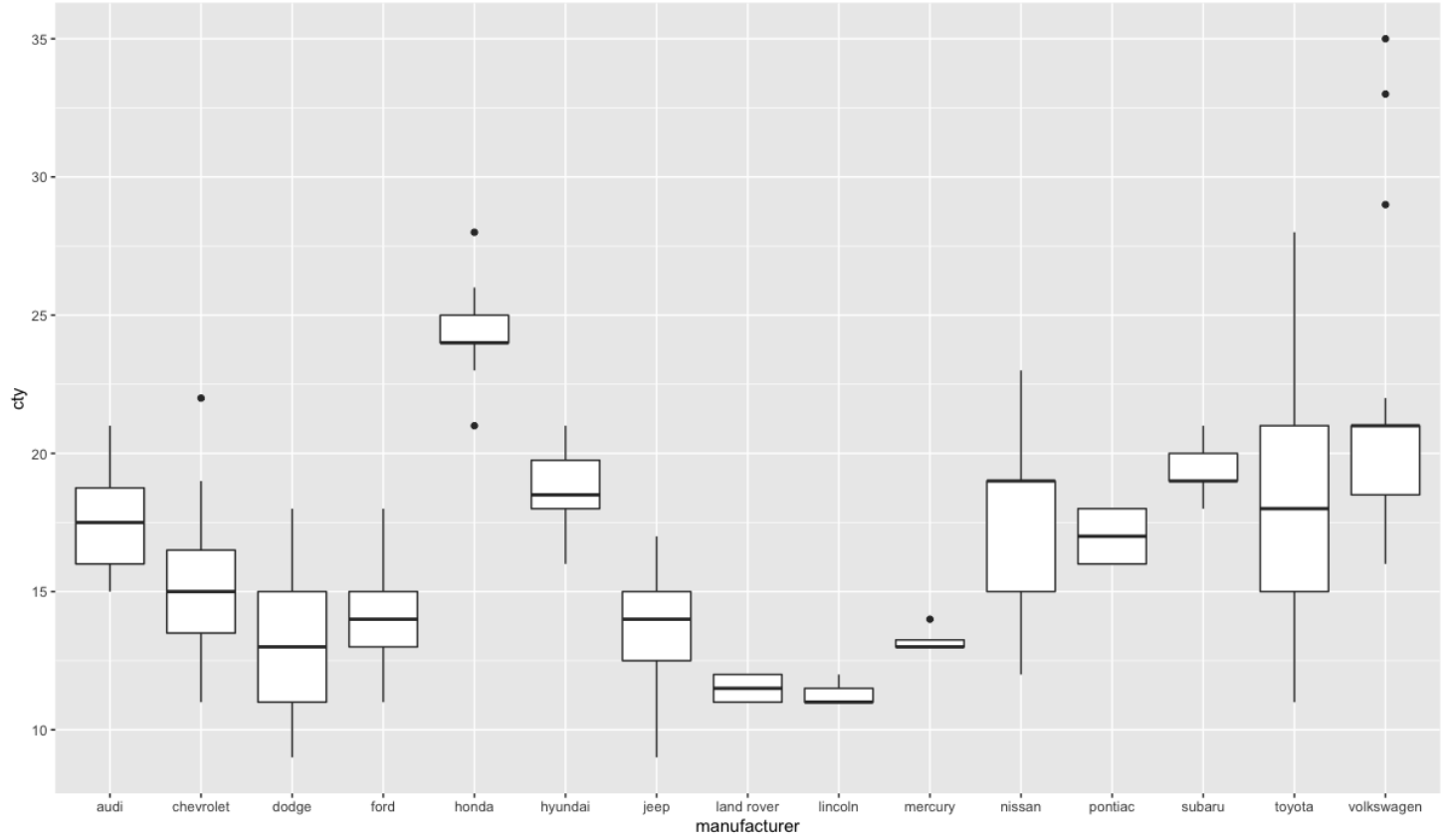
Covariation with ggpairs()

```
library(GGally)
my_vars <- subset(mpg,select=c(cty,hwy,displ))
ggpairs(my_vars)
```



Challenge 5.2

Generate the following boxplot from mpg



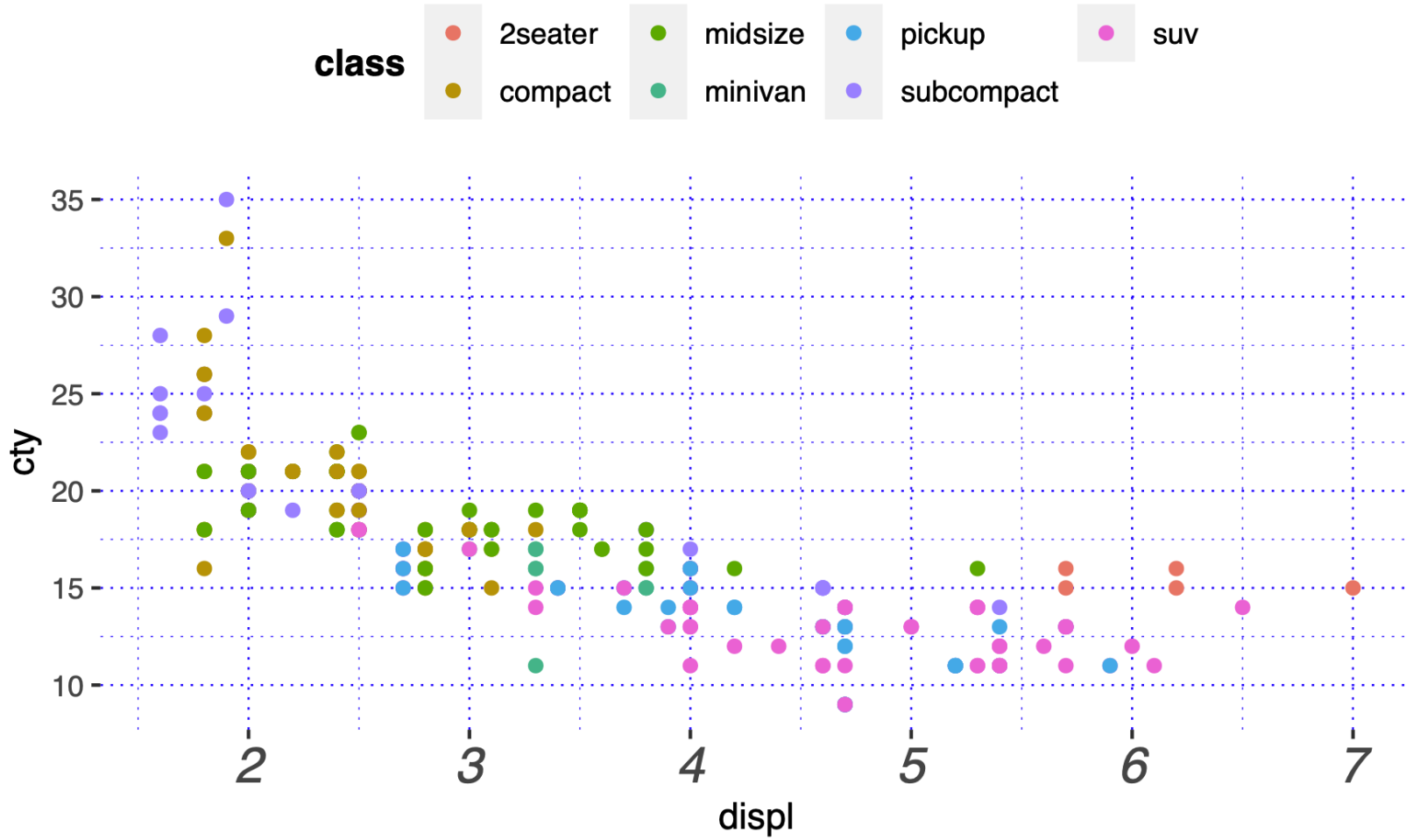
Themes in ggplot2

The theme system allows you to control non-data plot elements (fonts, ticks, legends etc)

- Theme elements specify non-data elements legend.title
- Element functions to describe the visual properties of an element element_text()
- The theme() function to change default theme elements
- Ready to use themes theme_bw()

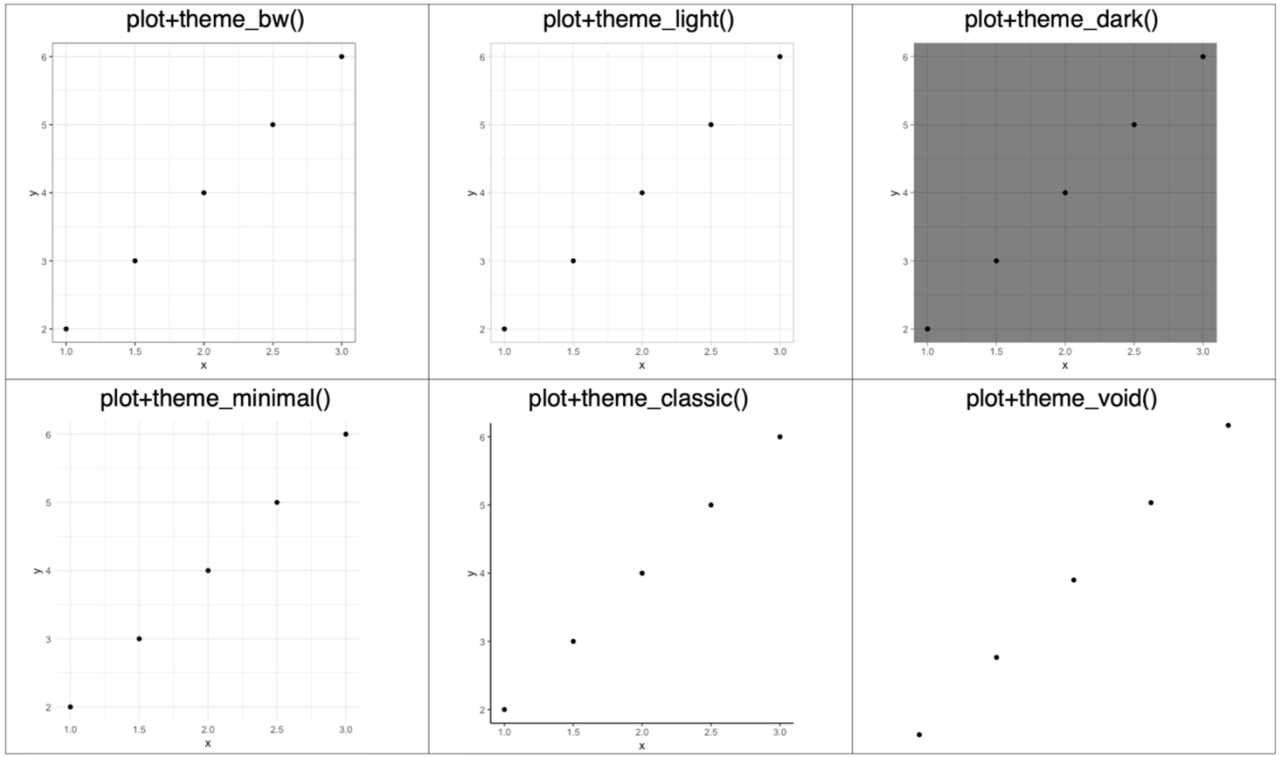
```
ggplot(data=mpg,aes(x=displ,y=cty,colour=class))+
  geom_point()+
  theme(legend.title = element_text(face="bold"),
        legend.position = "top",
        axis.text.x = element_text(size=15,face="italic"),
        panel.background = element_rect(fill="white"),
        panel.grid = element_line(color = "blue", size = 0.3, linetype = 3))
```

```
ggplot(data=mpg,aes(x=displ,y=cty,colour=class))+
  geom_point()+
  theme(legend.title = element_text(face="bold"),
        legend.position = "top",
        axis.text.x = element_text(size=15,face="italic"),
        panel.background = element_rect(fill="white"),
        panel.grid = element_line(color = "blue", size = 0.3, linetype = 3))
```

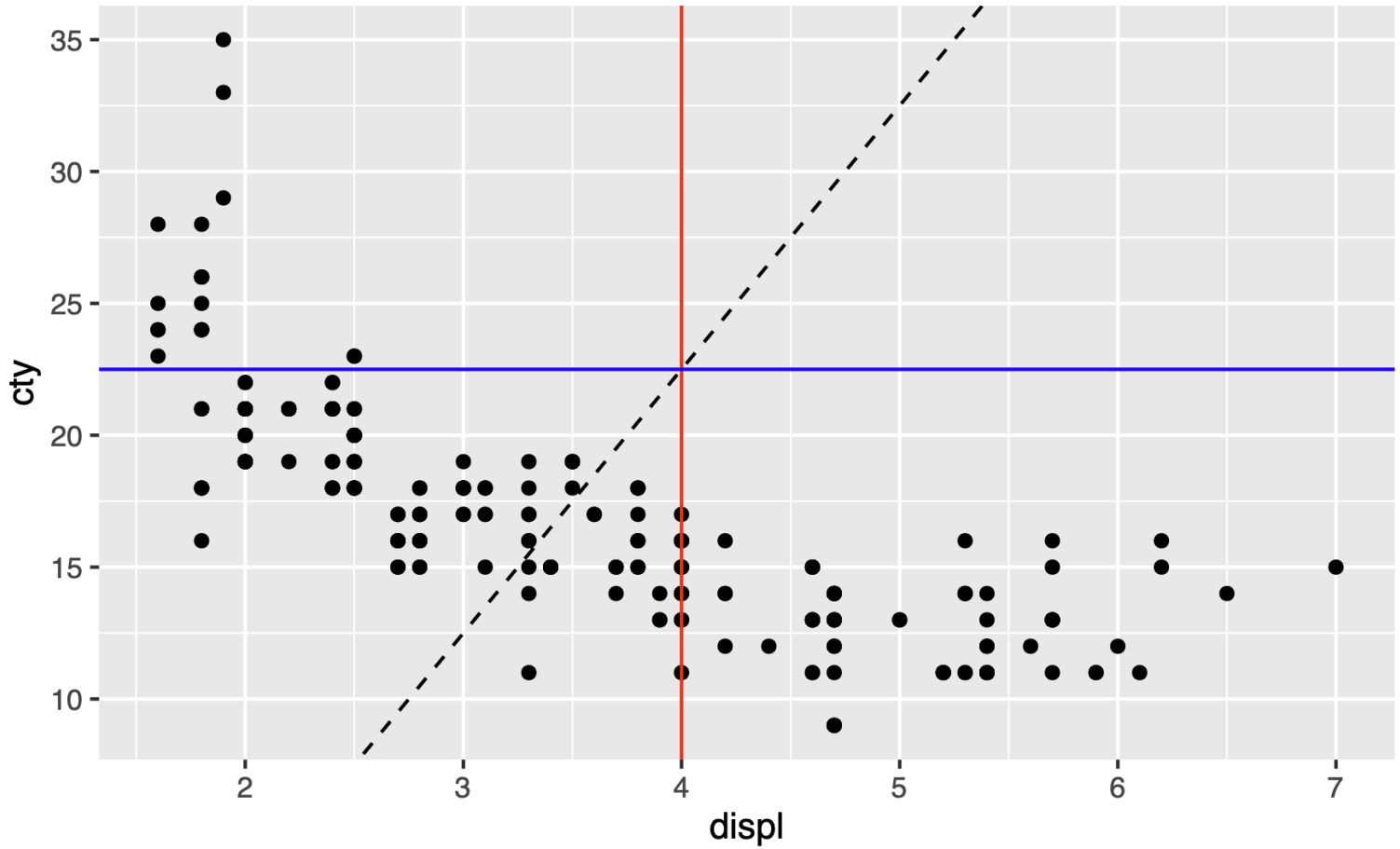


Ready to use themes

```
d <- tibble(x=seq(1,3,by=0.5),y=2*x)
plot <- ggplot(data=d,mapping=aes(x=x,y=y))+
          geom_point()
```



```
ggplot(data=mpg,mapping=aes(x=displ,y=cty))+
  geom_point()+
  geom_vline(xintercept = 4,colour="red")+
  geom_hline(yintercept = 22.5,colour="blue")+
  geom_abline(slope=10,intercept=-17.5,linetype=2)
```



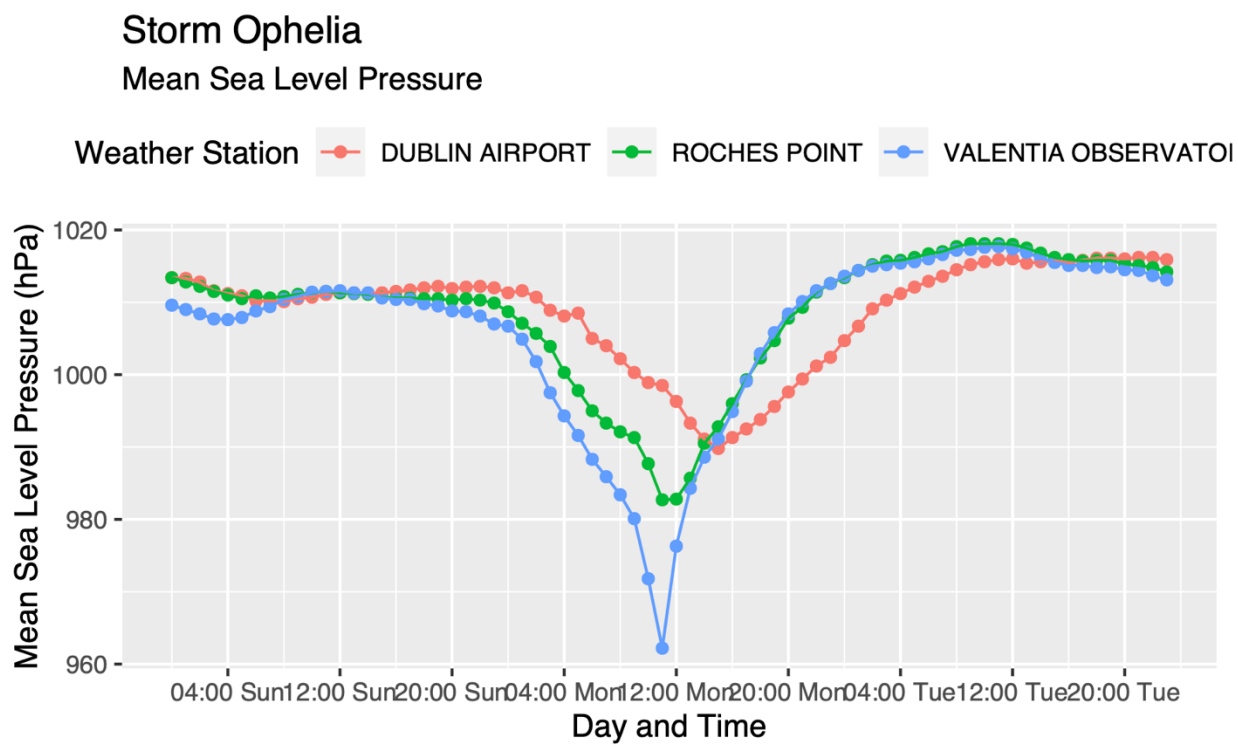
Mini case – Storm Ophelia 2017

- aimsir17 package (CRAN)
- 219K records
  - Hourly observations
  - 25 weather stations
- Variables
  - rain
  - temp
  - rhum
  - msl
  - wdsp
  - wddir

```
library(aimsir17)

observations
#> # A tibble: 219,000 x 12
#>    station  year month   day  hour date                rain  temp
#>    <chr>   <dbl> <dbl> <int> <int> <dttm>              <dbl> <dbl>
#>  1 ATHENRY  2017     1     1     0 2017-01-01 00:00:00   0     5.2
#>  2 ATHENRY  2017     1     1     1 2017-01-01 01:00:00   0     4.7
#>  3 ATHENRY  2017     1     1     2 2017-01-01 02:00:00   0     4.2
#>  4 ATHENRY  2017     1     1     3 2017-01-01 03:00:00   0.1   3.5
#>  5 ATHENRY  2017     1     1     4 2017-01-01 04:00:00   0.1   3.2
#>  6 ATHENRY  2017     1     1     5 2017-01-01 05:00:00   0     2.1
#>  7 ATHENRY  2017     1     1     6 2017-01-01 06:00:00   0     2
#>  8 ATHENRY  2017     1     1     7 2017-01-01 07:00:00   0     1.7
#>  9 ATHENRY  2017     1     1     8 2017-01-01 08:00:00   0     1
#> 10 ATHENRY  2017     1     1     9 2017-01-01 09:00:00   0     1.1
#> # ... with 218,990 more rows, and 4 more variables: rhum <dbl>,
#> #   msl <dbl>, wdsp <dbl>, wddir <dbl>
#> # i Use `print(n = ...)` to see more rows, and `colnames()` to see al
```

```r
storm <- observations |>
           subset(month==10 &
                  day %in% 15:17 &
                  station %in% c("VALENTIA OBSERVATORY",
                                 "ROCHES POINT",
                                 "DUBLIN AIRPORT"),
                  select=c(station,date,temp,msl,wdsp))
storm
#> # A tibble: 216 x 5
#>    station         date                temp   msl  wdsp
#>    <chr>           <dttm>             <dbl> <dbl> <dbl>
#>  1 DUBLIN AIRPORT 2017-10-15 00:00:00  14.7 1013.    13
#>  2 DUBLIN AIRPORT 2017-10-15 01:00:00  14.5 1013.    15
#>  3 DUBLIN AIRPORT 2017-10-15 02:00:00  14.5 1013.    13
#>  4 DUBLIN AIRPORT 2017-10-15 03:00:00  14.5 1012.    13
#>  5 DUBLIN AIRPORT 2017-10-15 04:00:00  14.4 1011.    16
#>  6 DUBLIN AIRPORT 2017-10-15 05:00:00  14.7 1011.    16
#>  7 DUBLIN AIRPORT 2017-10-15 06:00:00  14.9 1010.    13
#>  8 DUBLIN AIRPORT 2017-10-15 07:00:00  15   1010.    11
#>  9 DUBLIN AIRPORT 2017-10-15 08:00:00  15.3 1010.    12
#> 10 DUBLIN AIRPORT 2017-10-15 09:00:00  15.9 1010.    12
#> # ... with 206 more rows
#> # i Use `print(n = ...)` to see more rows
```

```
ggplot(data=storm,aes(x=date,y=msl,colour=station))+
  geom_point()+
  geom_line() +
  theme(legend.position = "top")+
  scale_x_datetime(date_breaks = "8 hour",date_labels = "%H:%M %a")+
  labs(title="Storm Ophelia",
       subtitle = "Mean Sea Level Pressure",
       x="Day and Time",
       y="Mean Sea Level Pressure (hPa)",
       colour="Weather Station")
```



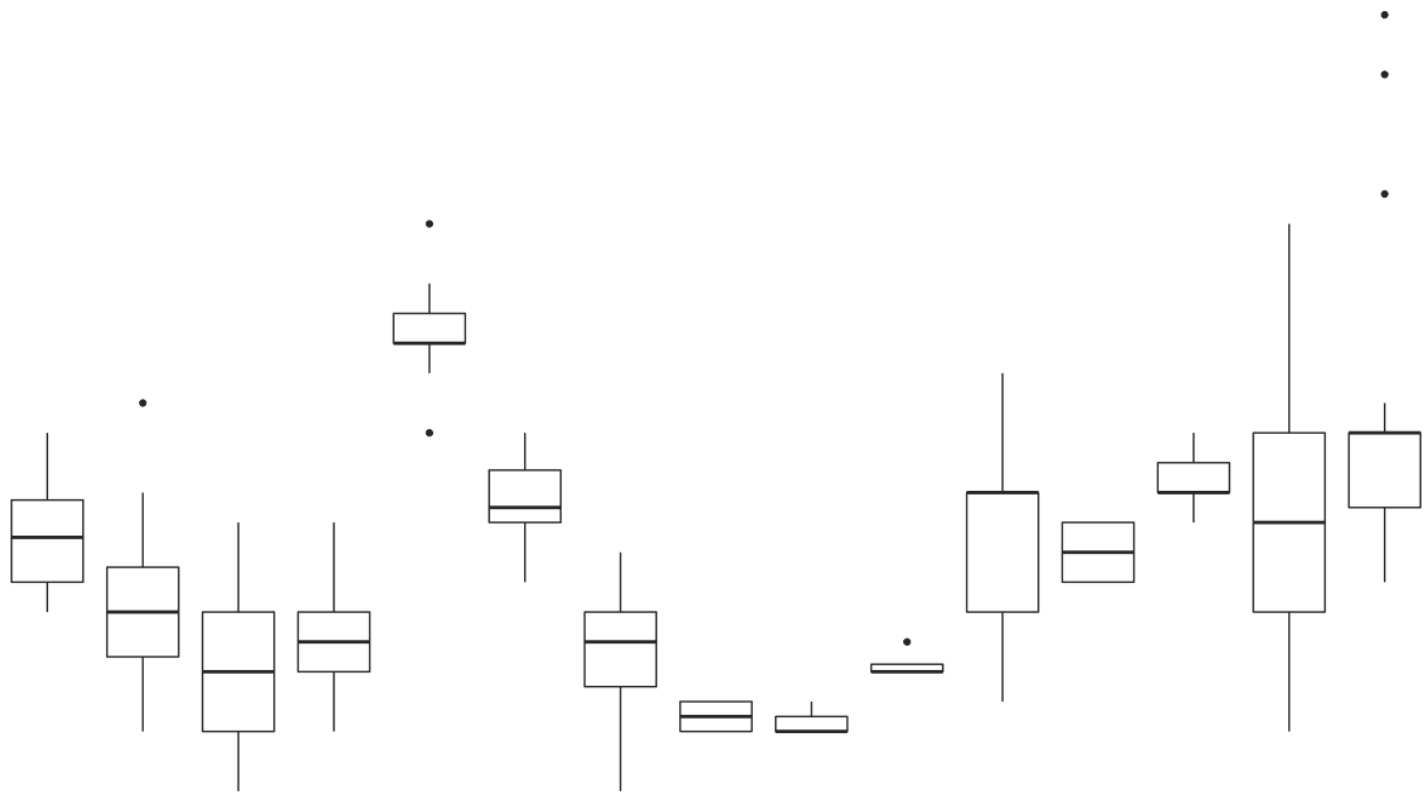
scale_x_datetime()

String	Meaning for `date_labels`
`%S`	Second (00-59)
`%M`	Minute (00-59)
`%l`	Hour, 12 hour clock (1-12)
`%I`	Hour, 12 hour clock (01-12)
`%p`	am/pm
`%H`	Hour, in 24-hour clock (00-23)
`%a`	Day of the week, abbreviated (Mon-Sun)
`%A`	Day of week, full (Monday-Sunday)
`%e`	Day of month (1-31)
`%d`	Day of month (00-31)

String	Meaning for `date_labels`
`%m`	Month, numeric (01-12)
`%b`	Month, abbreviated (Jan-Dec)
`%B`	Month, full (January-December)
`%y`	Year, without scentrut (00-99)
`%Y`	Year, with century (0000-9999)

Challenge 5.3

Generate the following boxplot from mpg. The data is the same as 5.2



Useful geoms

Geom	Purpose
geom_smooth()	Fits a smoother to data and displays the smooth and its standard error
geom_boxplot()	Produces a box-and-whisker plot to summarise the distribution of a set of points
geom_histogram() geom_freqpoly()	Shows the distribution of continuous variables
geom_bar()	Shows the distribution of categorical variables
geom_path() geom_line()	Draws lines between data points
geom_area()	Draws an area plot, which is a line plot filled to the y-axis. Multiple groups will be stacked upon each other
geom_rect() geom_tile() geom_raster()	Draw rectangles
geom_polygon()	Draws polygons, which are filled paths.

Summary

- Layered grammar of graphics, enables us to concisely describe the components of a graphic.
- Benefits:
  - plots can be designed in a layered manner (+ operator)
  - a wide range of plots can be generated to support decision analysis, including scatterplots, histograms and time series charts
  - charts can be developed rapidly, and this support an iterative process of decision support

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