

Machine Learning with R: an introduction

Overview

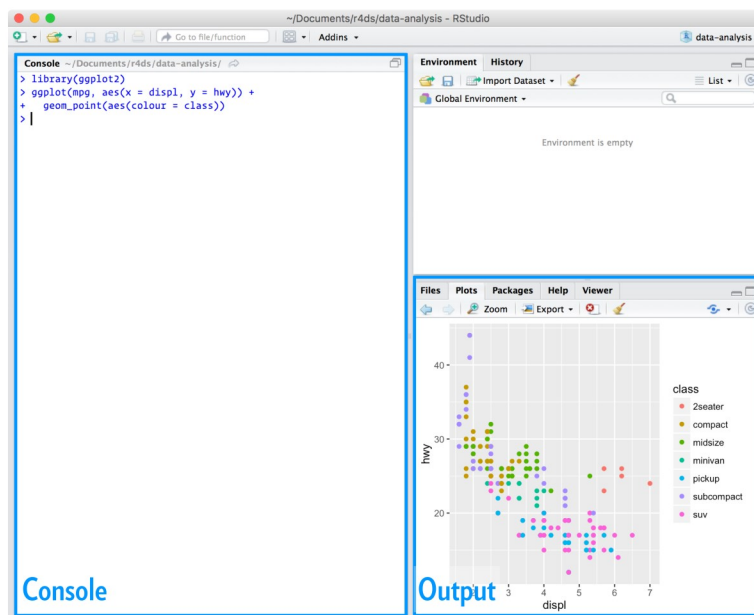
- Data preparation and feature engineering
- **Reading and importing data, data exploration** and filtering
- Cleaning and Preprocessing
- **Exploratory Data Analysis**
- Supervised learning
 - Classification: K-nearest neighbours, logistic regression, decision trees, random forest
 - Regression: K-nearest neighbours, **linear regression**, regression trees, random forest
- Model Evaluation and Model Selection

Outline

- **First day:**
 - R and R Studio
 - **Reading and importing data, exploratory analysis and visualization**

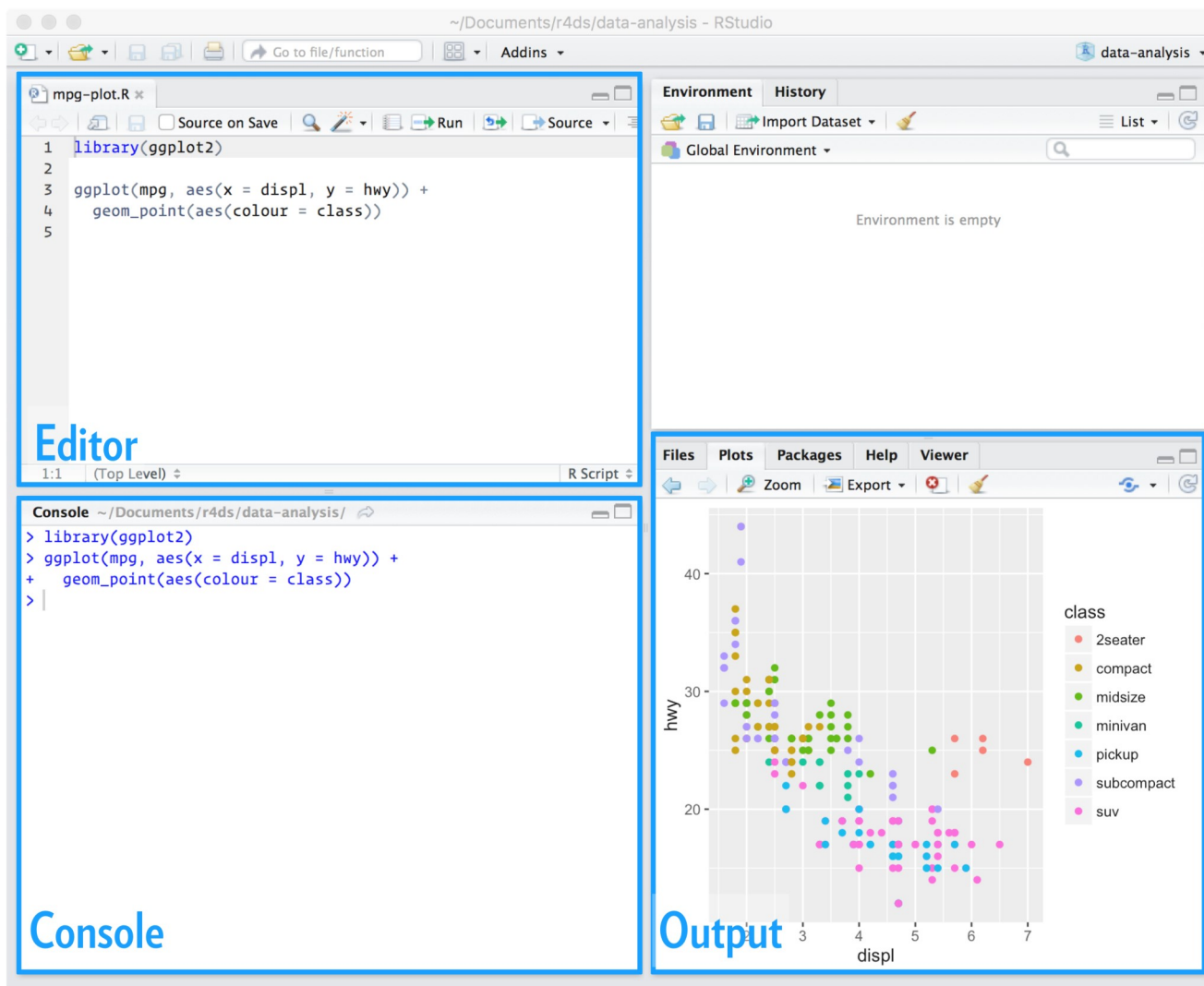
Rstudio - Overview

When you start RStudio, you'll see two key regions in the interface:



Source: R for Data Science

Scripts



Exploratory Data Analysis

- Visualization
 - Data visualization with ggplot2
- Descriptive Statistics

Data visualization - Let's start

R has several systems and functions for plotting graphs.

Package **ggplot2** (Wickham, 2016) is one of the most elegant and versatile.

- **ggplot2** implements the grammar of graphics, a coherent system for describing and making graphs.

📖 H. Wickham. **ggplot2: Elegant Graphics for Data Analysis**. Springer-Verlag New York, 2016.

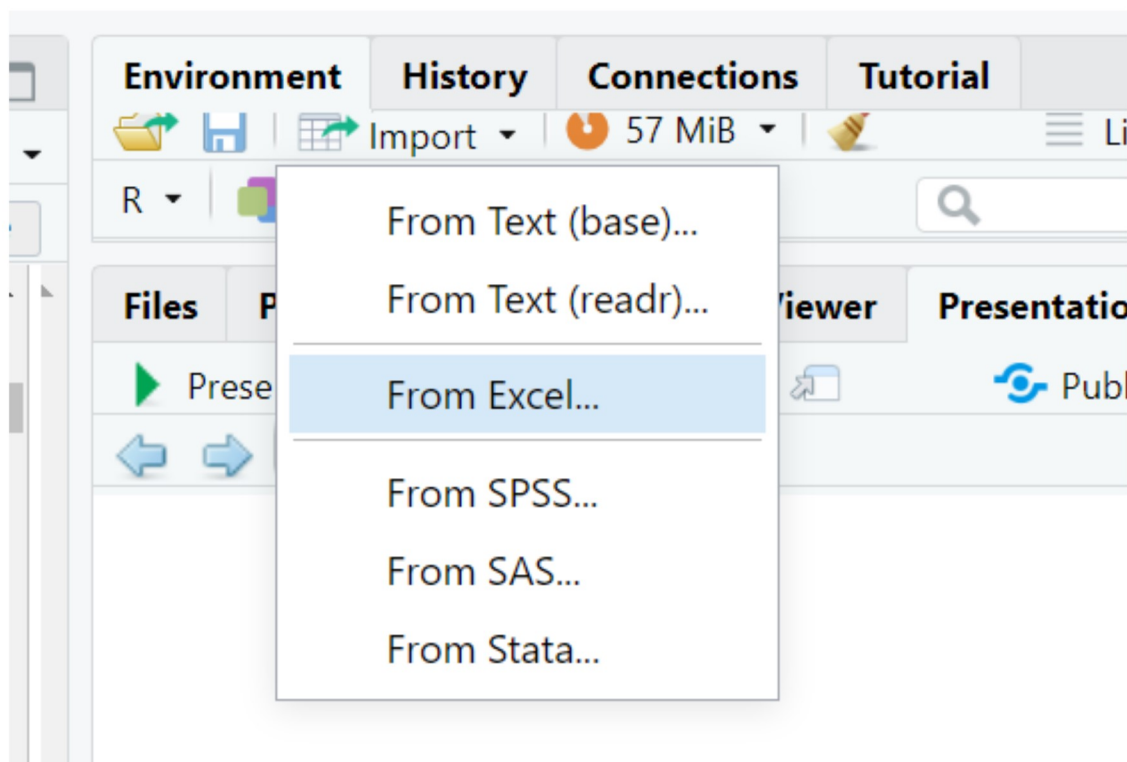
This short course focuses on ggplot2, one of the core members of the tidyverse. To access functions,

help pages (with examples), and functions, load tidyverse (that contains package **ggplot2**) or load **ggplot2**.

```
library(tidyverse)
```

The package tidyverse need to be installed before. You only install a package once, but you need to reload it every time you start a new session.

Import data



Our “first” data: The mpg data frame

This data frame is part of the ggplot2 package and contains observations collected by the US Environment Protection Agency on 38 models of cars (script_o1_mpg.R).

```
mpg
```

```
# A tibble: 234 × 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 d     comp...
```

```
10 audi a4 quattro 2 2008 4 manu... 4 20 28 p comp...
# i 224 more rows
```

More information about the data you can find in the help pages (run `help (mpg)` or `?mpg`).

Questions

- Do cars 🚗 with big engines use more fuel than cars with small engines?
- What does the relationship between engine size size and fuel efficiency look like?

Understanding mpg data set

Description: This dataset contains a subset of the fuel economy data. It contains only models which had a new release every year between 1999 and 2008. For more information: *help (mpg)* or *?mpg*. The data frame has 234 rows and 11 variables.

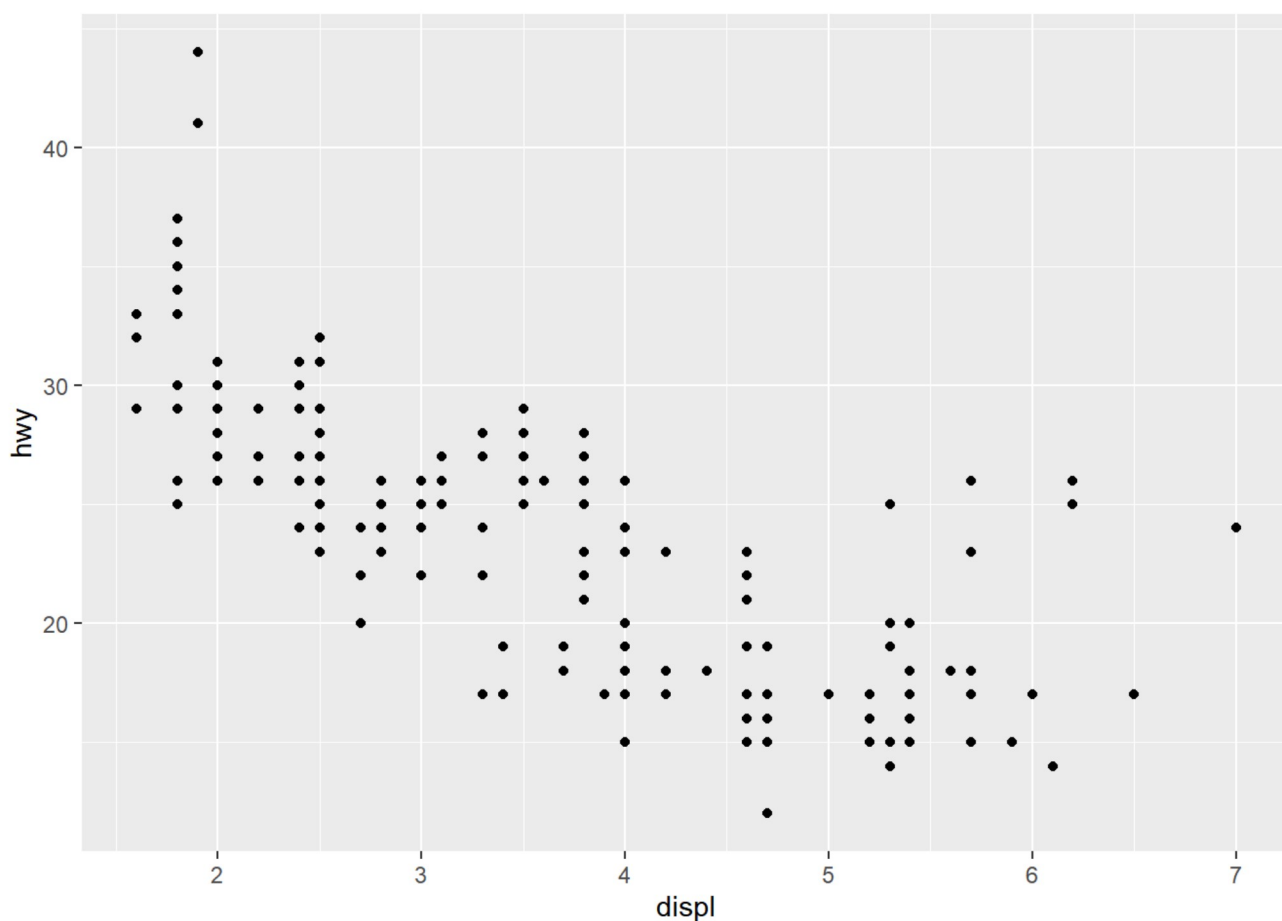
MPG data set

Variable	
manufacturer	manufacturer name
model	model name
displ	engine displacement (in liters)
year	year of manufacture
cyl	number of cilinders
trans	type of transmission
drv	the type of drive train, where f = front-wheel drive, r = rear wheel drive, 4 = 4wd
cty	fuel efficiency on the city, in miles per gallon
fl	fuel type
class	"type" of car
hwy	fuel efficiency on the highway, in miles per gallon (mpg)

Creating a scatterplot in ggplot2

To plot a scatterplot run this code to put `displ` (engines displacement) on the x-axis and `hwy` (fuel efficiency) on the y-axis.


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



ggplot () creates a coordinate systems that you can add layers to.

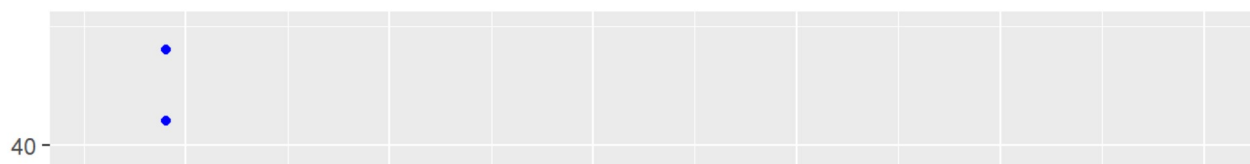
- The first argument is the data set - *ggplot(data set name)*.
- Add more layers: *geom_point ()* adds a layer of points and create a scatterplot.
- Each *geom_function* takes a mapping argument.
- This defines how variables in your data set are mapped to visual properties - *aes ()*.

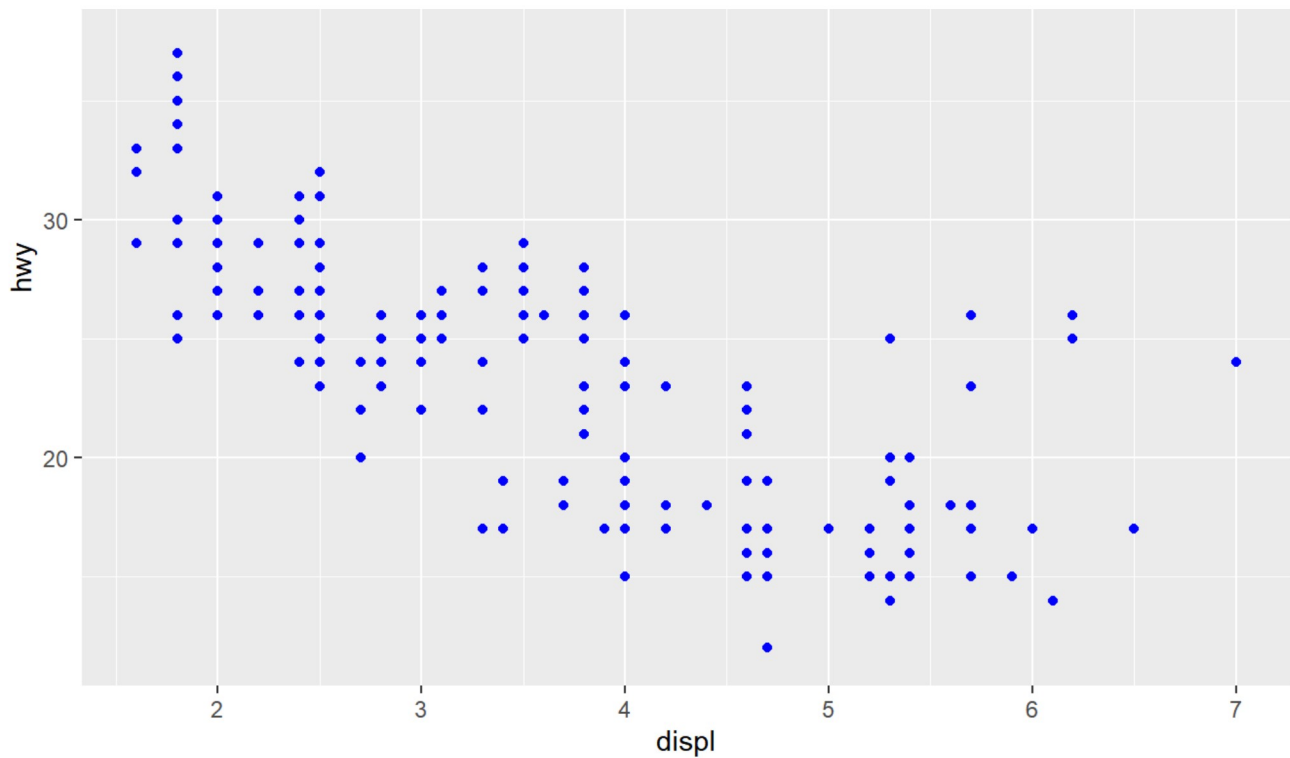
ggplot(dataset name)+geom_point(mapping = aes(x= **variable in x, y = **variable in y**))**

Colors

The code below show how the change the colors of the points.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue")
```

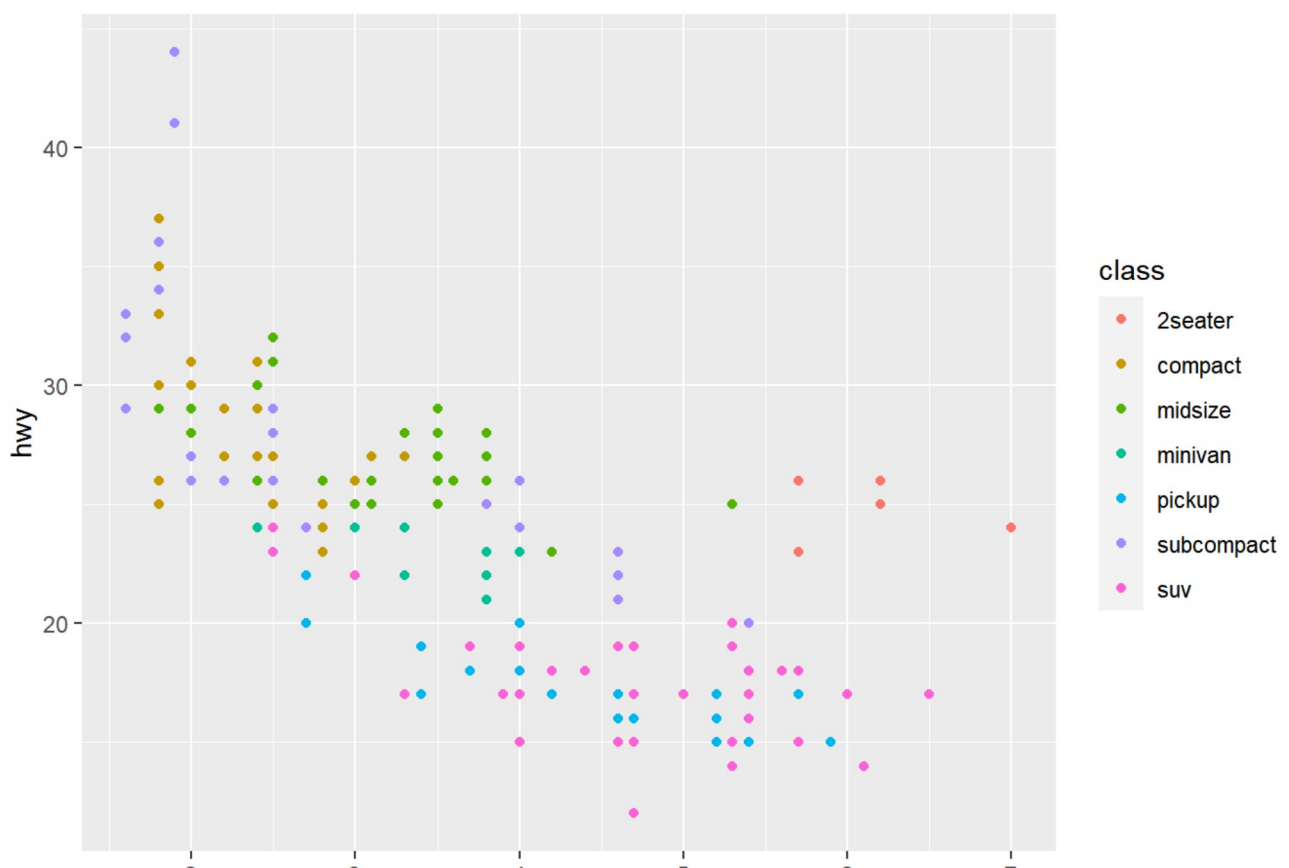




- color (or colour) = color name or number

Let map the point colors to the “class” variable to reveal the class of each car

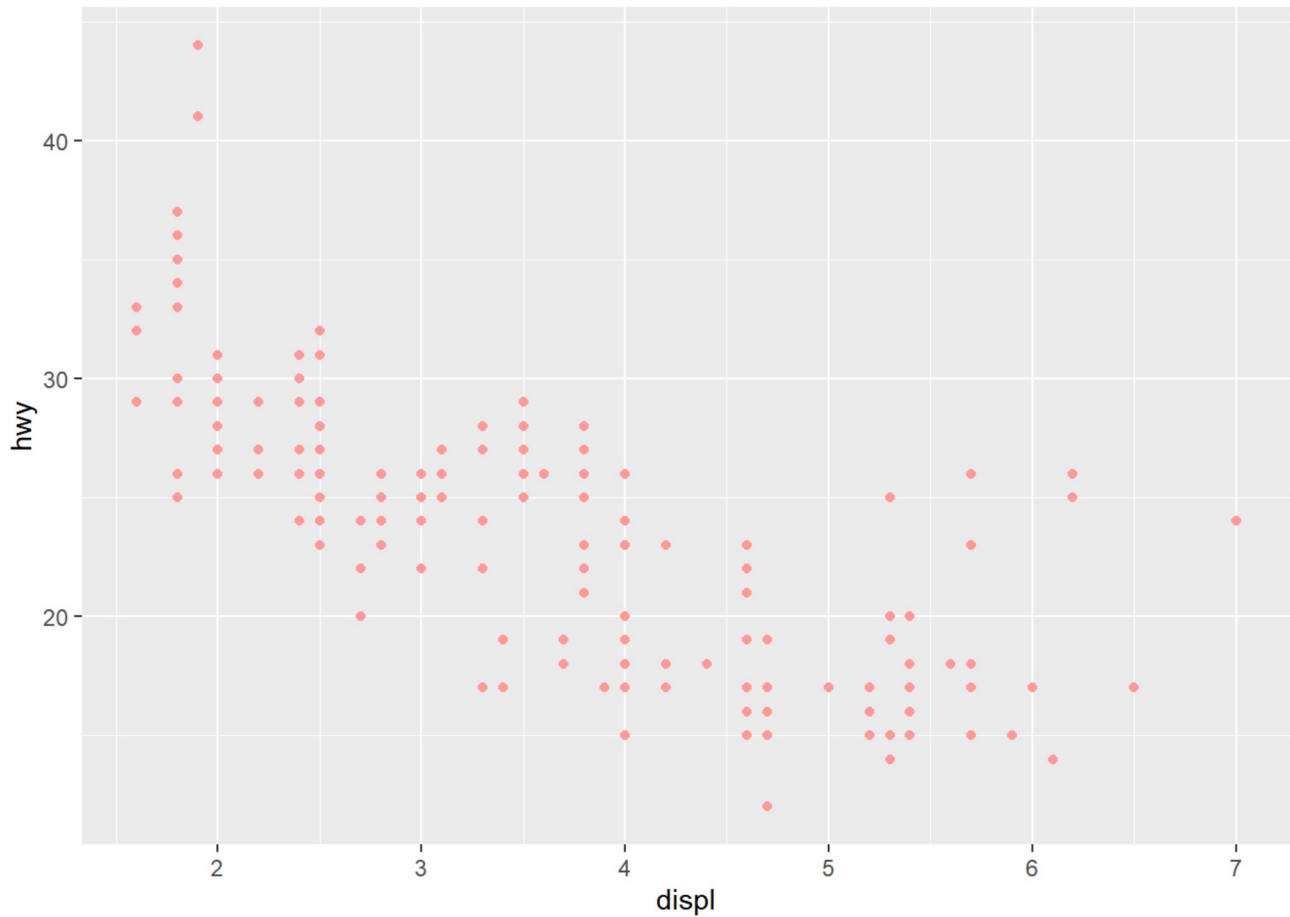
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```



2 3 4 5 6 7
displ

More colors

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy), color = "#FF9999")
```

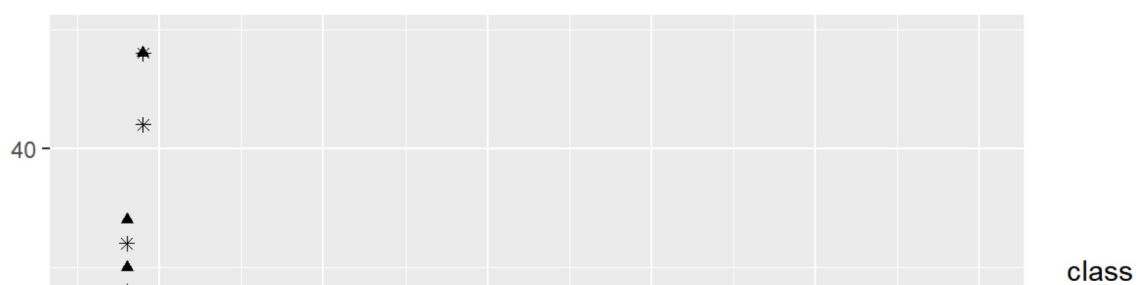


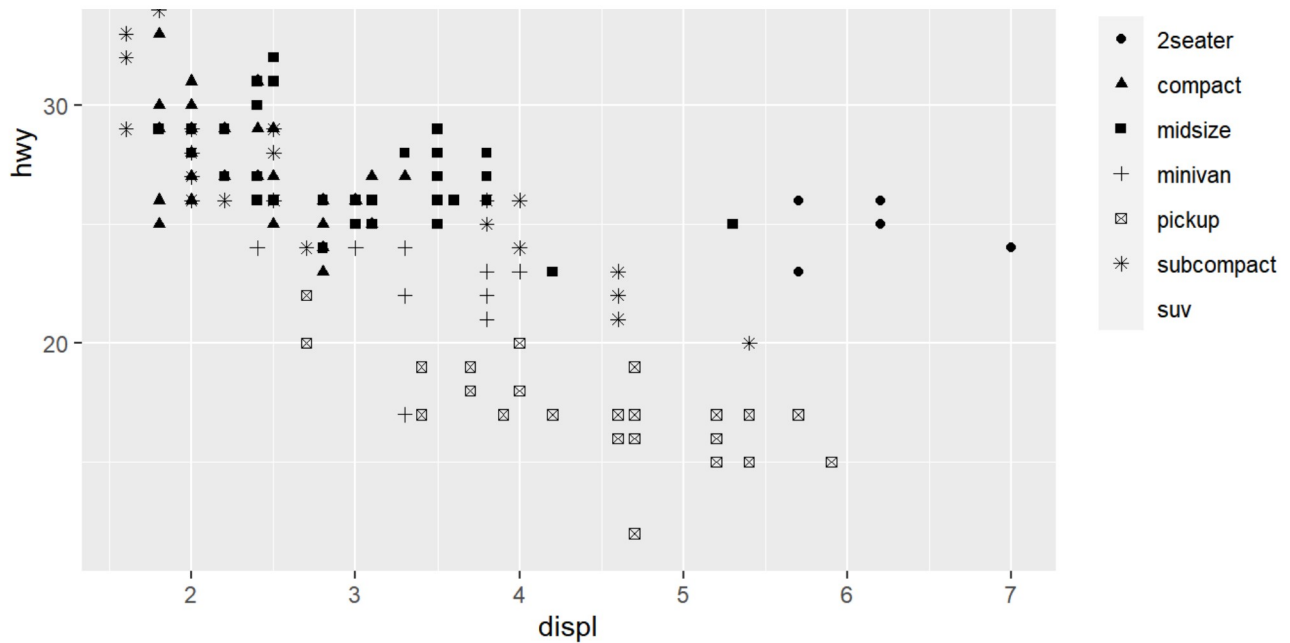
More about colors:

[http://www.cookbook-r.com/Graphs/Colors_\(ggplot2\)/#a-colorblind-friendly-palette](http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/#a-colorblind-friendly-palette)

Shape

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



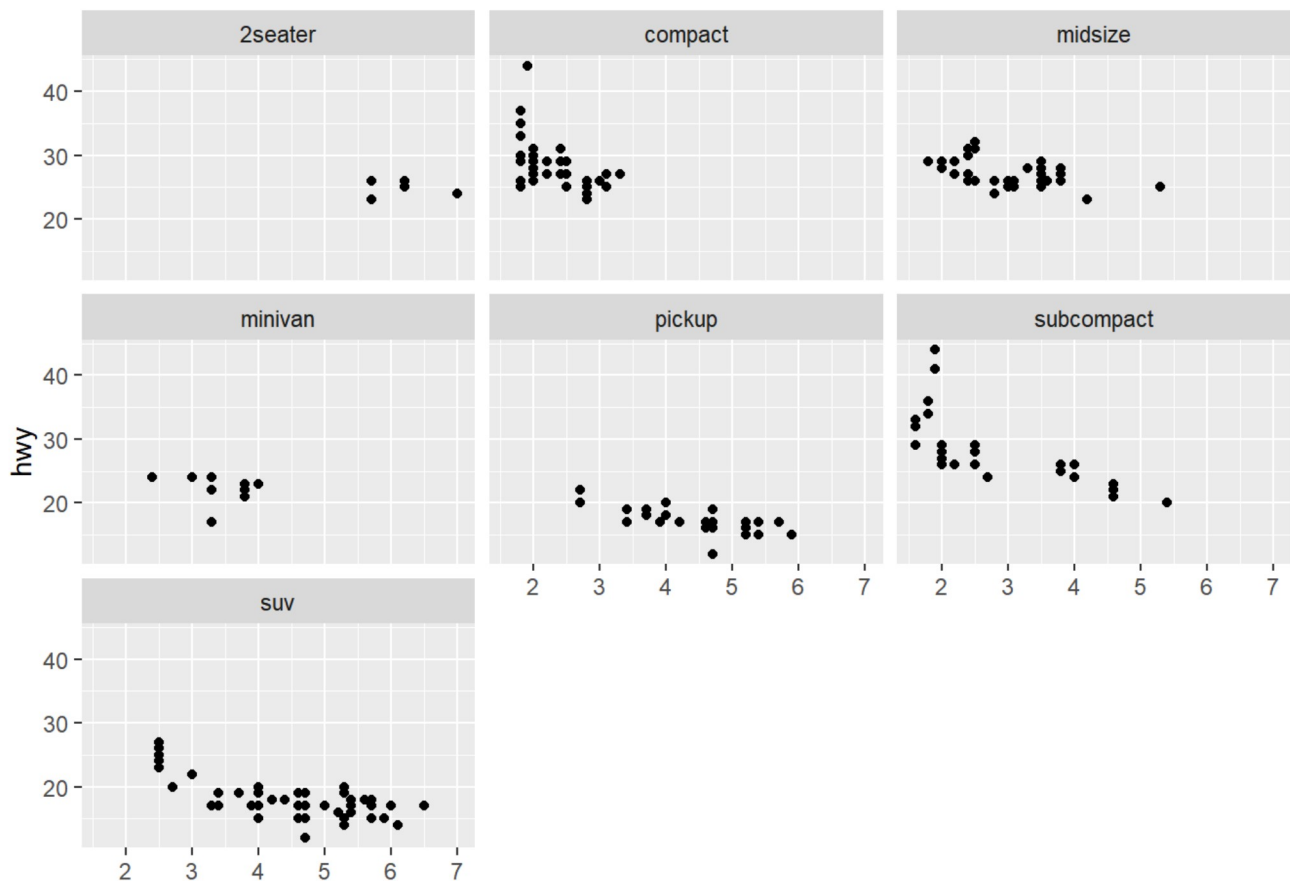


Facets

Useful to split the plot into subplots.

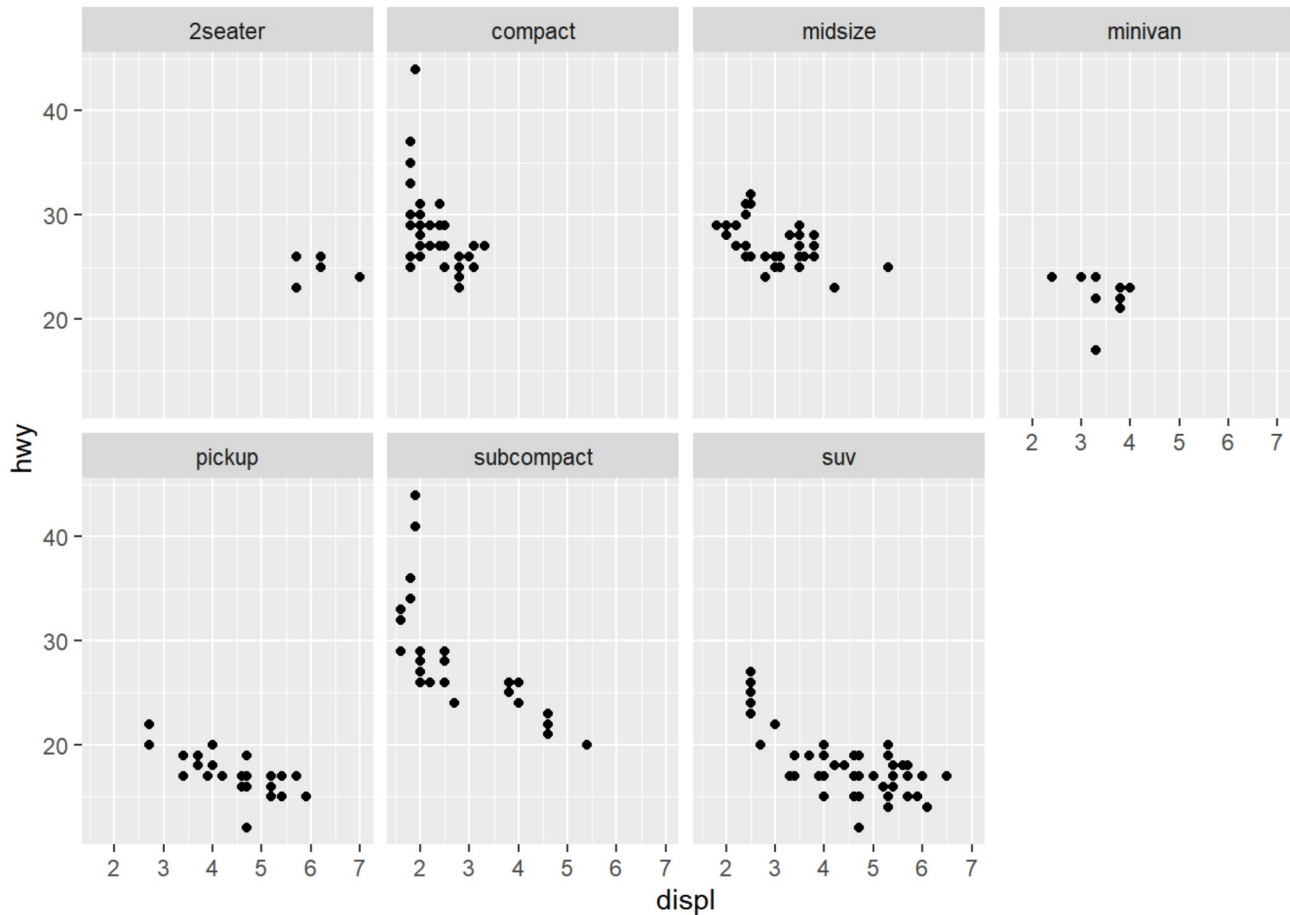
Each subplot display one subset of the data.

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



Controlling the rows

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

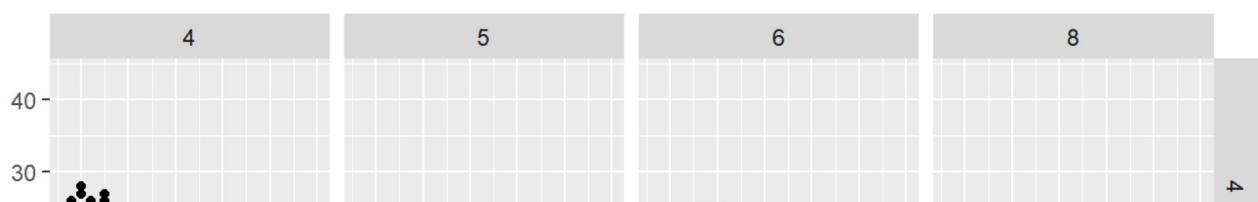


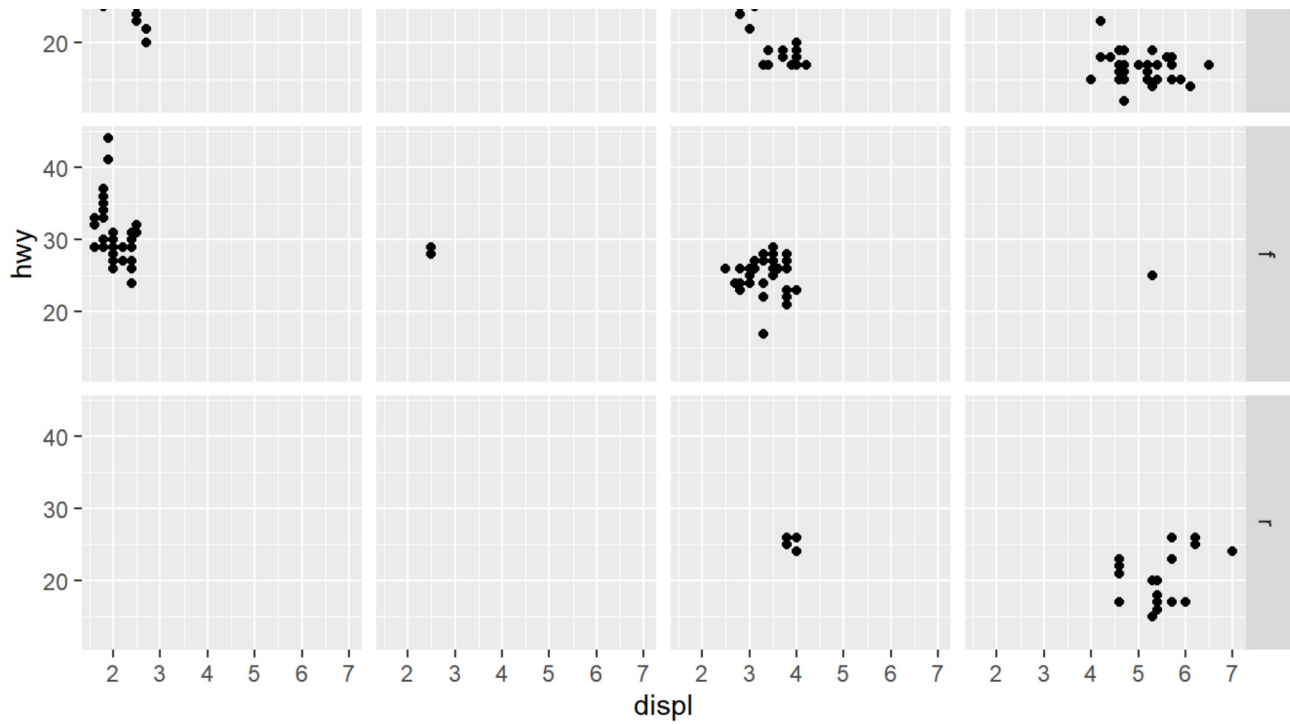
Combination of two variables

How the plot look with adding drv ?

drv = the type of drive train, where f = front-wheel drive, r = rear wheel drive, 4 = 4

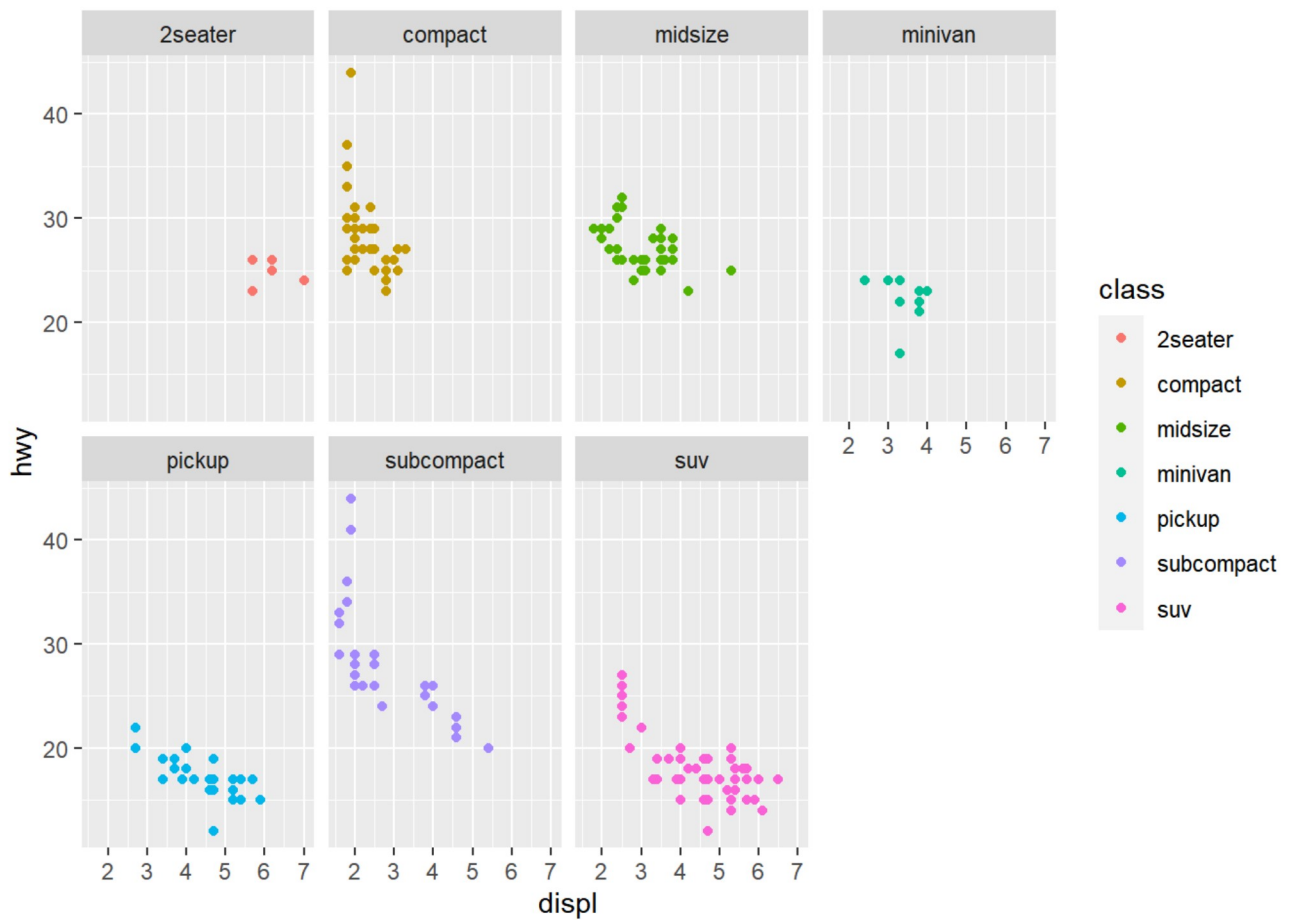
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ cyl)
```





Some colors

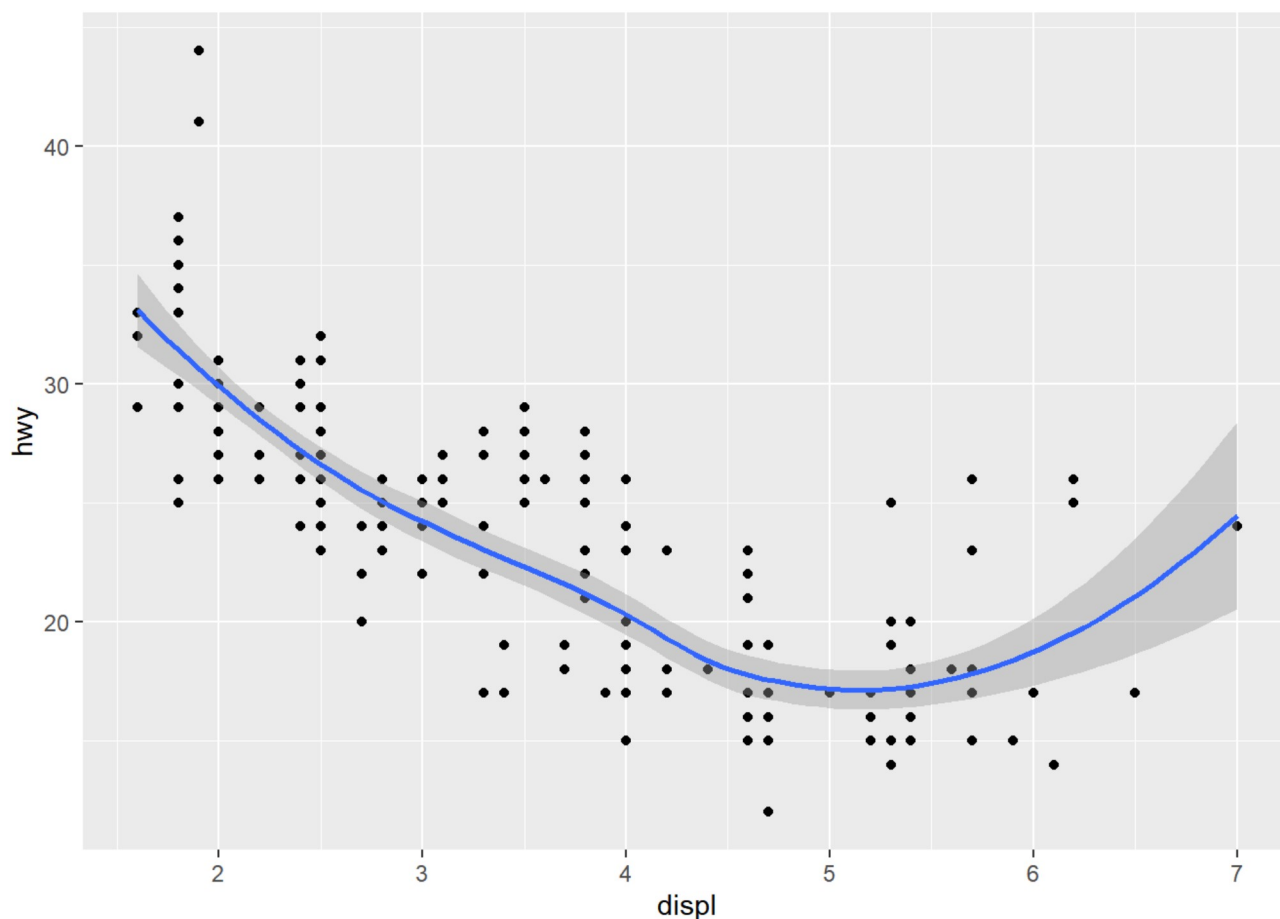
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class)) +
  facet_wrap(~class, nrow = 2)
```



Adding lines

`geom_smooth()` - Creates smoothed conditional means

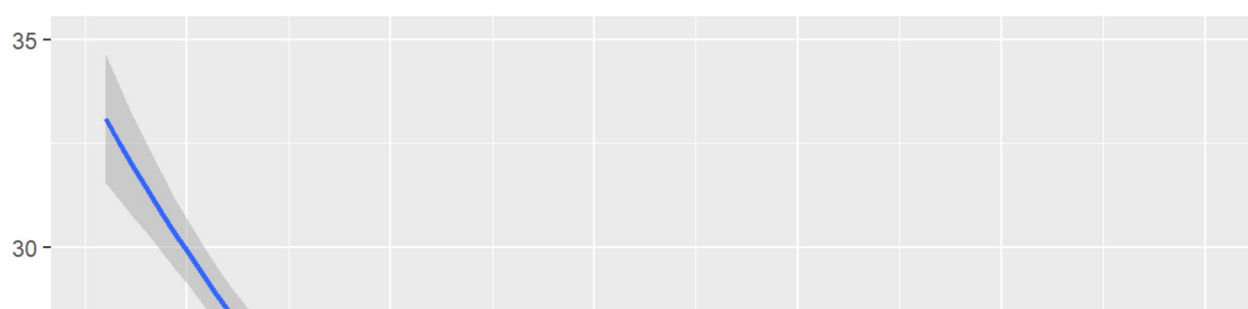
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

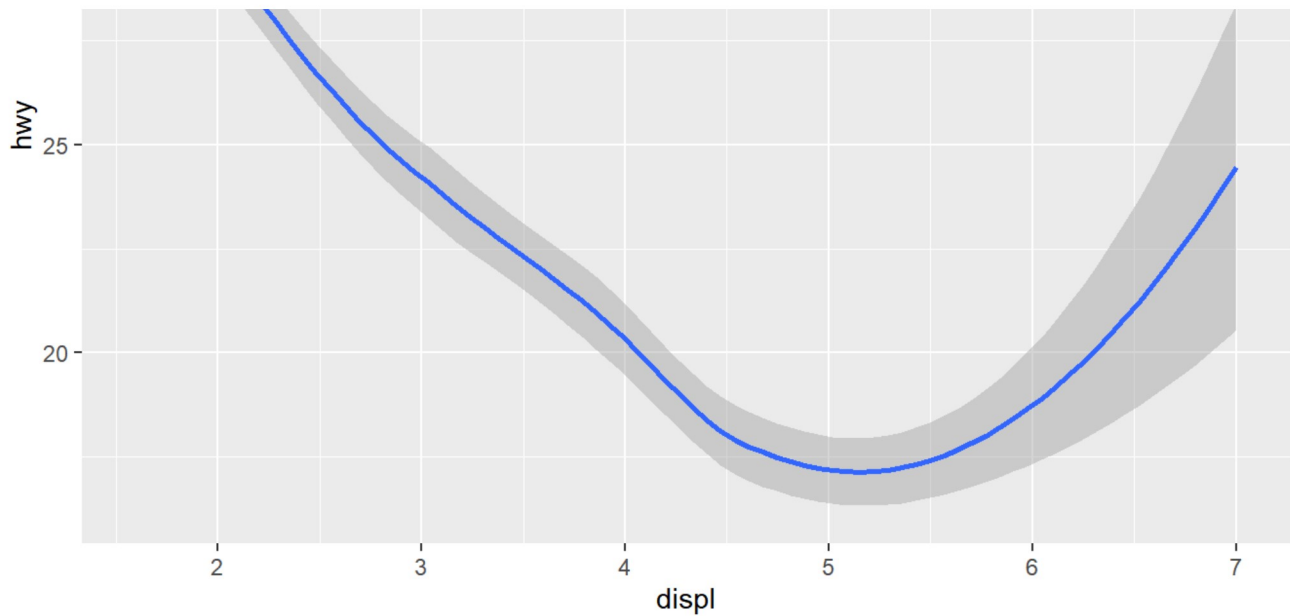


Some variations

Without the points

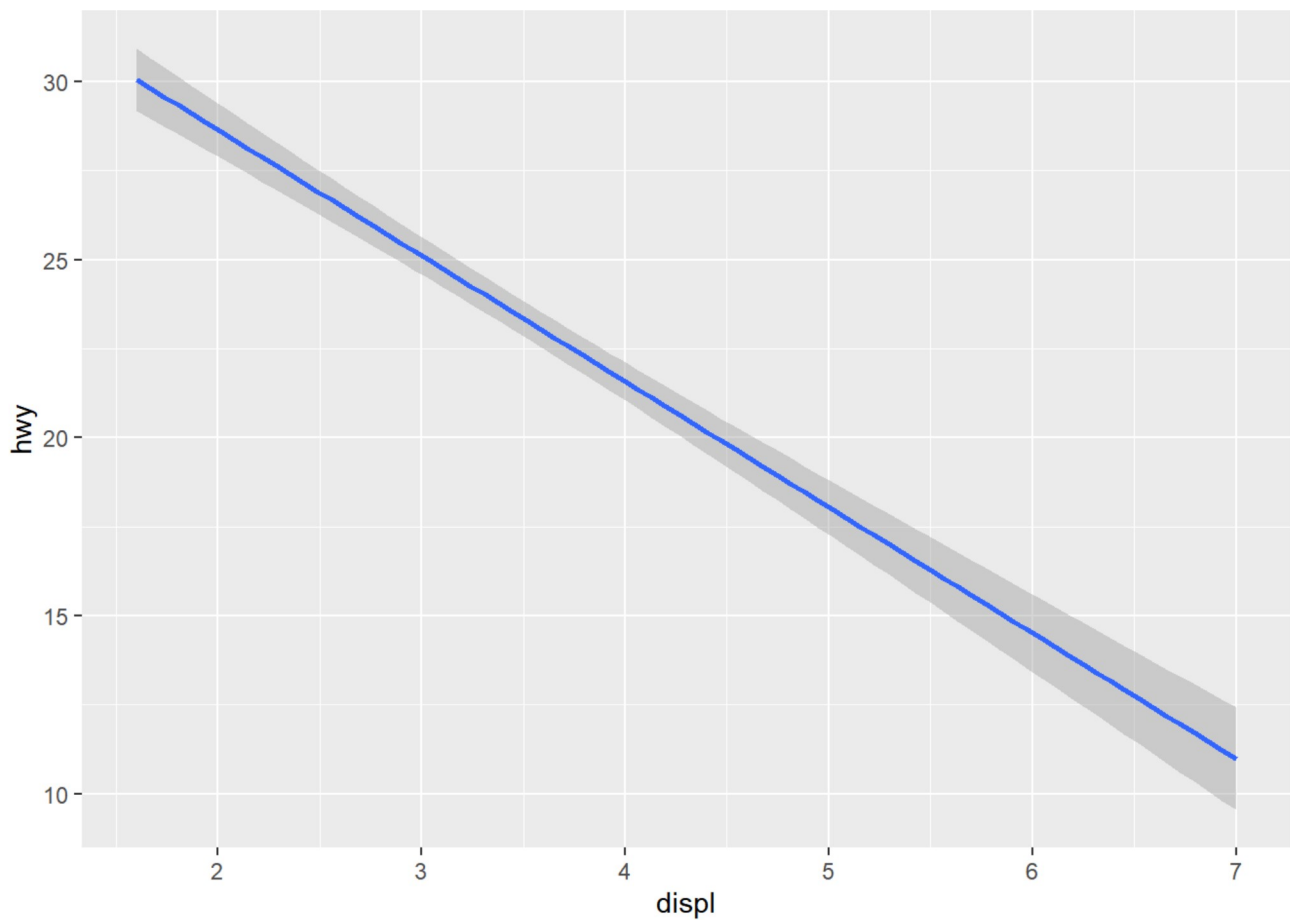
```
ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```





Using linear regression (lm)

```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy), method = "lm")
```

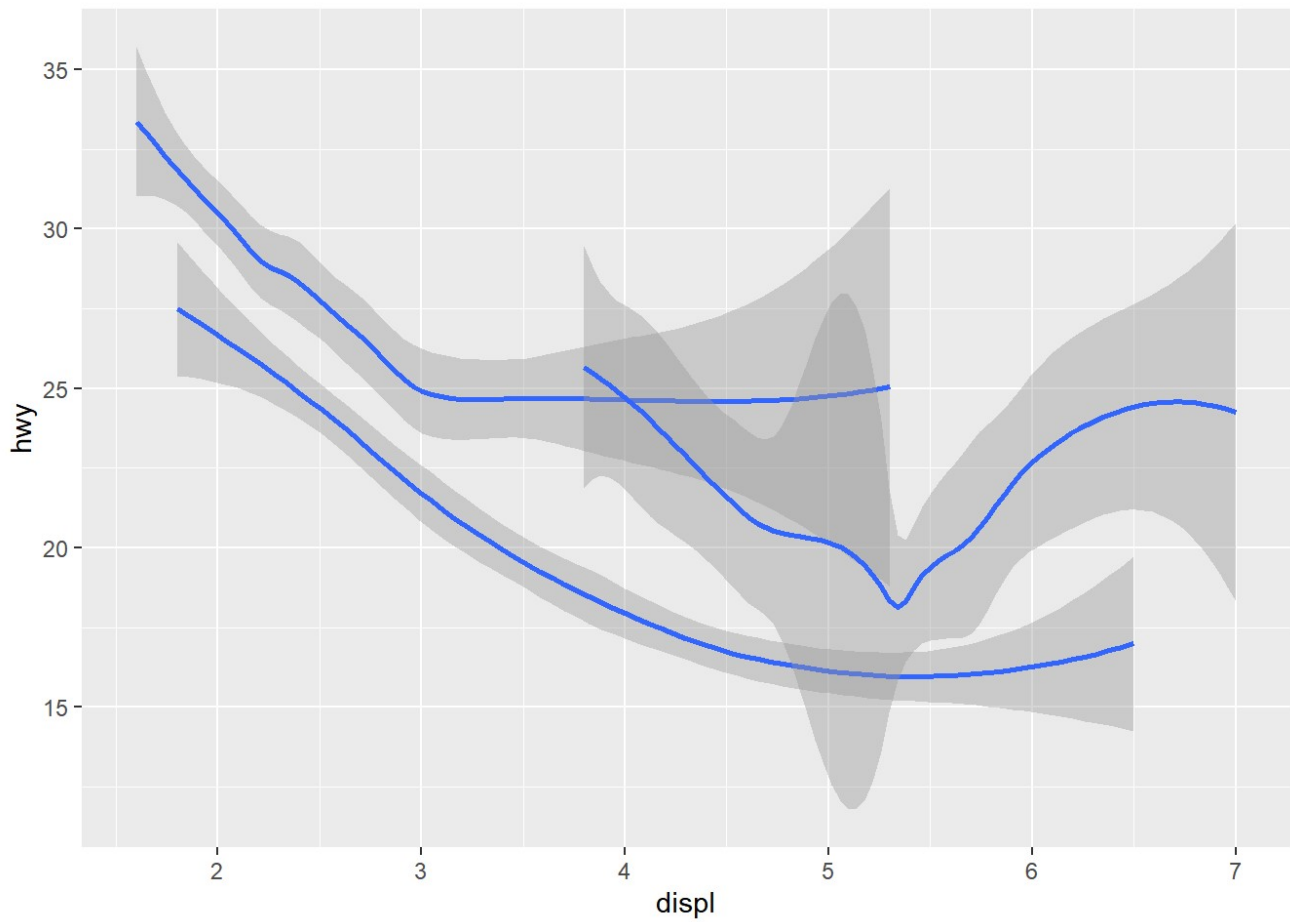


Some variations

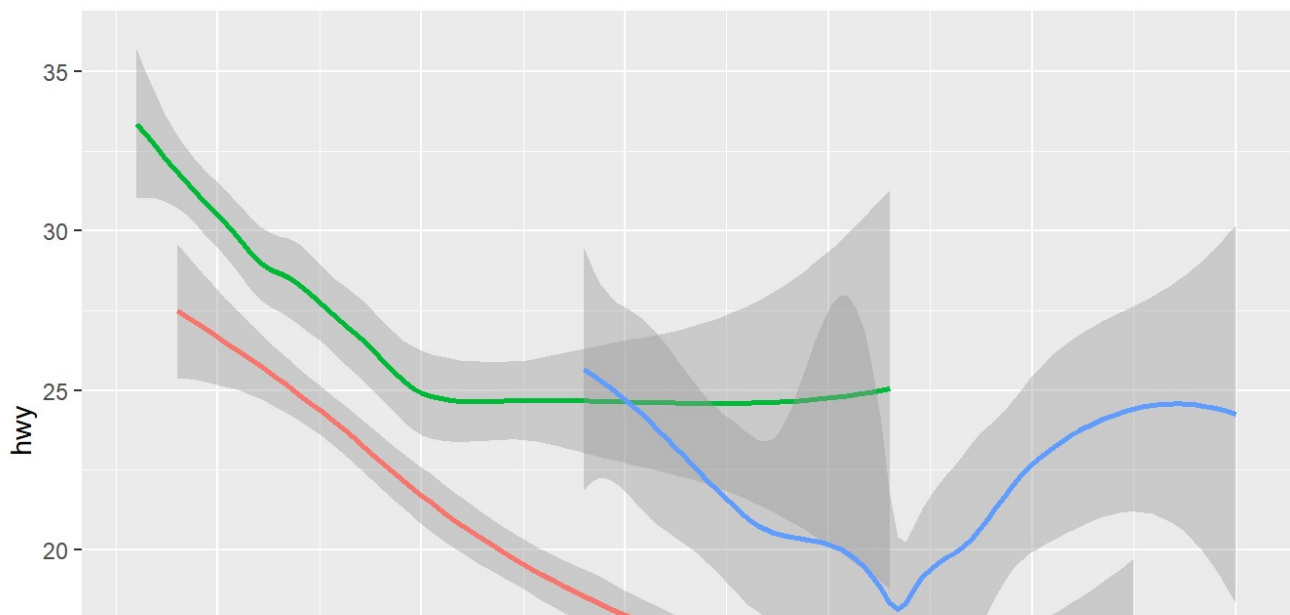
- `geom_smooth()` + `group` - separates the cars into different lines (colors and shapes) based on groups (drv value)

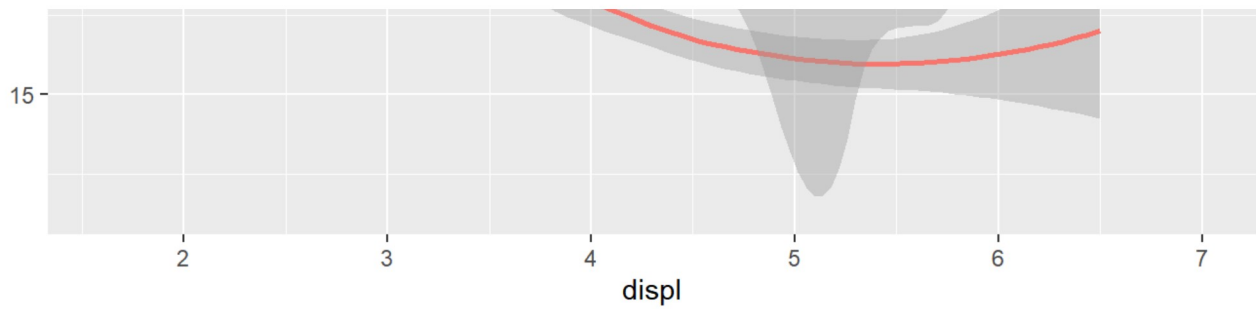
groups (drv value,

```
ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy, group = drv))
```

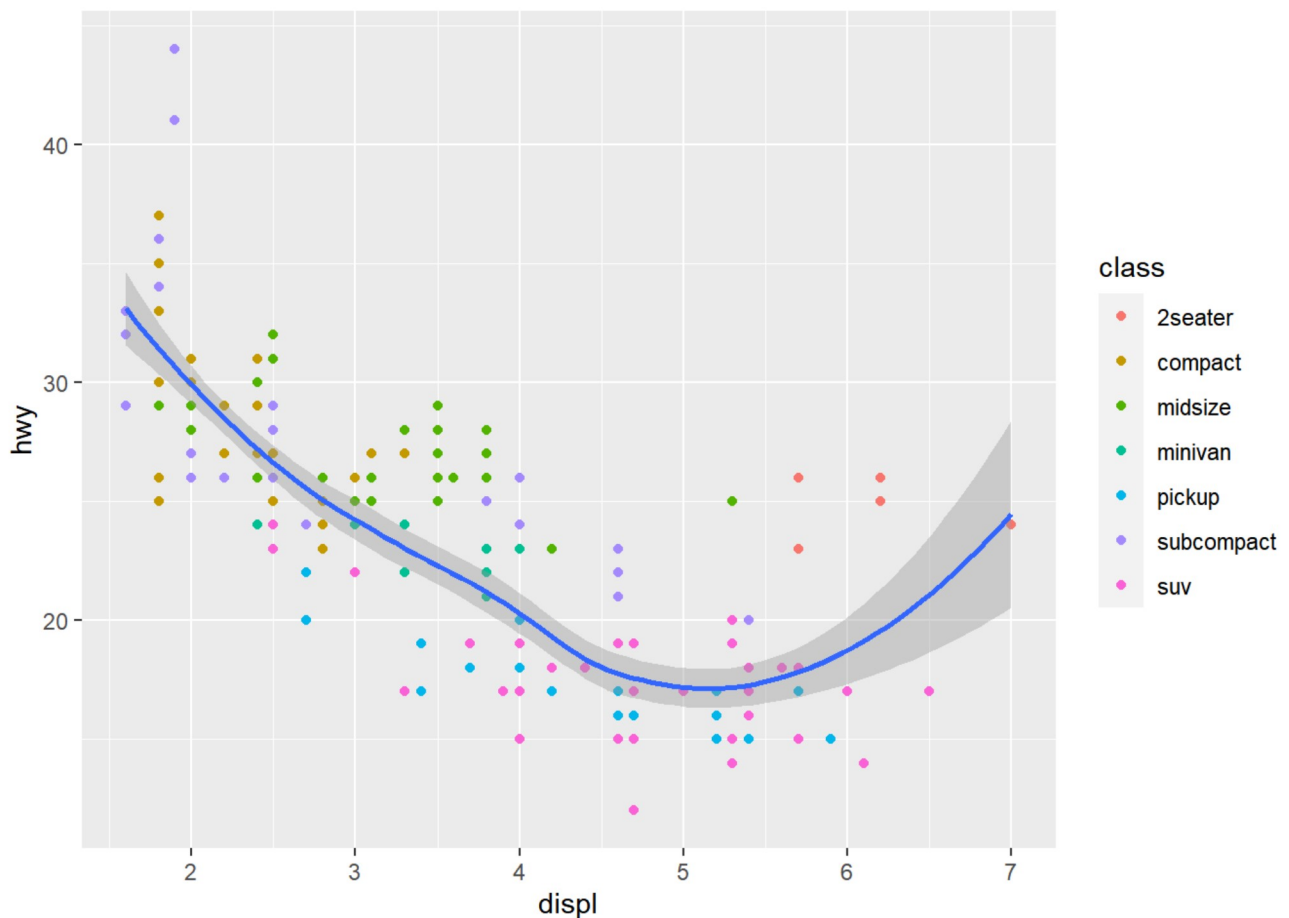


```
ggplot(data = mpg) +  
  geom_smooth(  
    mapping = aes(x = displ, y = hwy, color = drv),  
    show.legend = FALSE  
  )
```





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



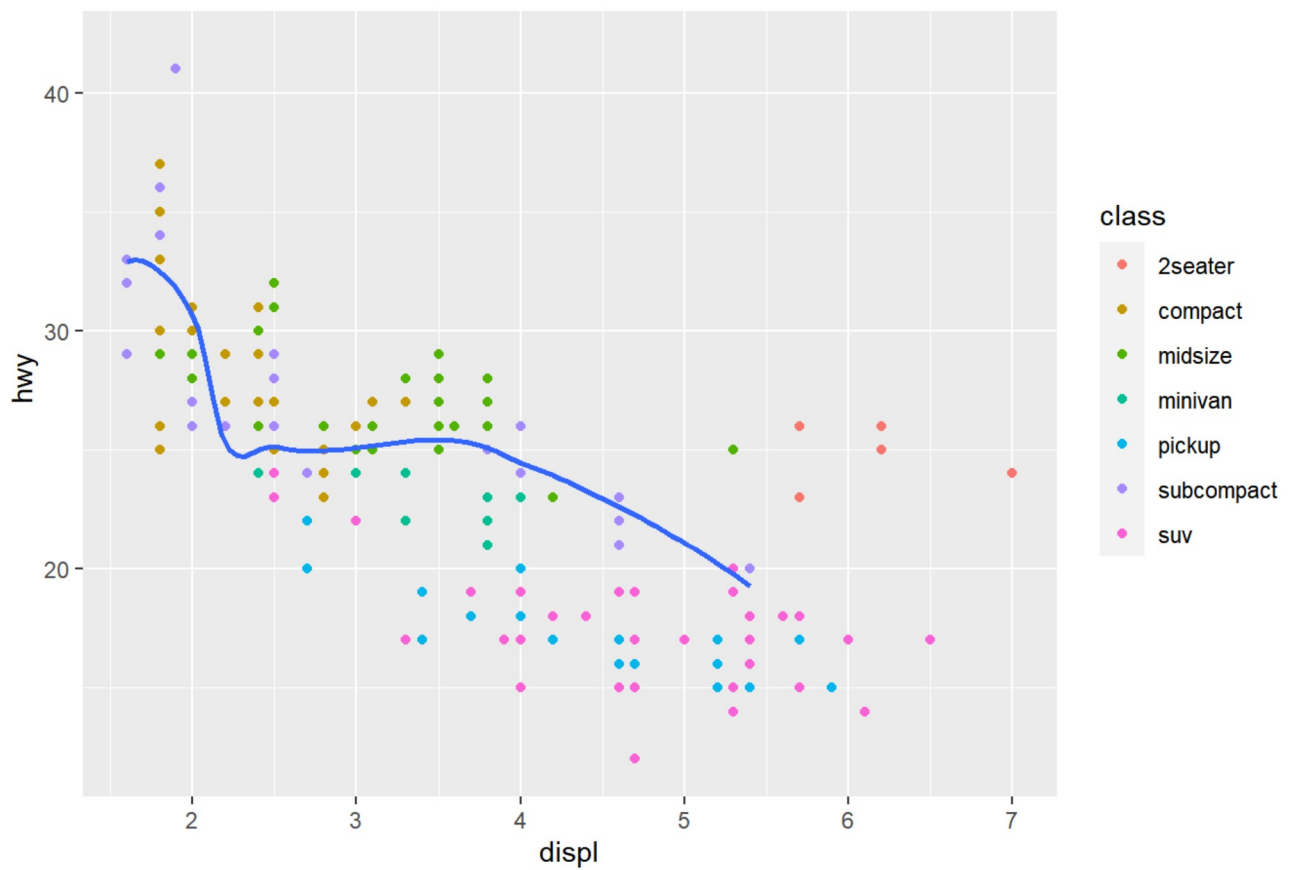
Filtering

Suppose we are interested only in the results of a specific group: subcompact car.

Smooth line for subcompact car: `filter(mpg, class == "subcompact")`

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)
```



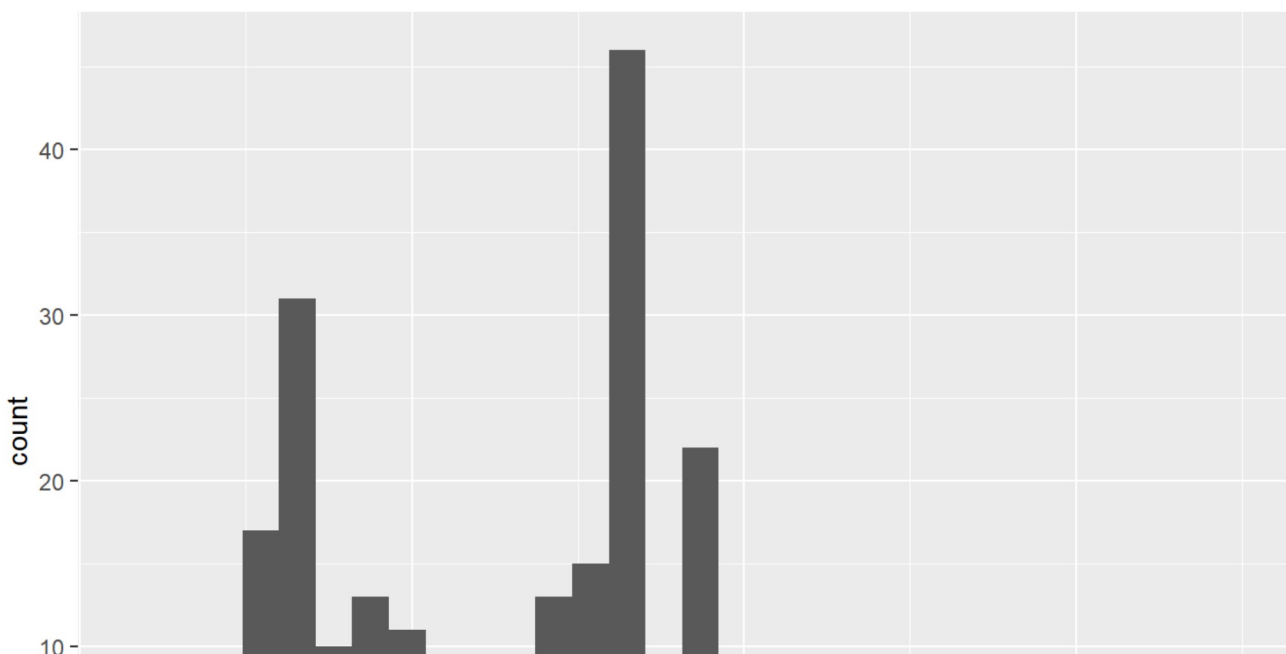


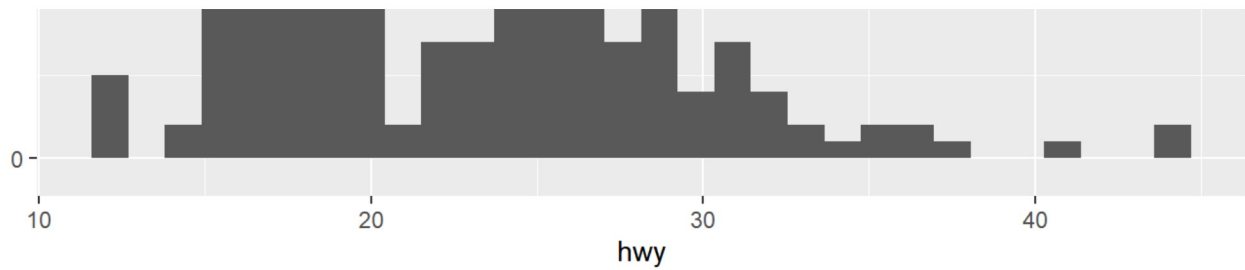
Histograms

The most common graph of the distribution of one quantitative variable is a histogram. Histogram can be used for quantitative continuous variables to find the shape or the distribution of the variable of interest.

Histogram of variable *mpg*.

```
ggplot(data=mpg)+geom_histogram(mapping=aes(x=hwy))
```

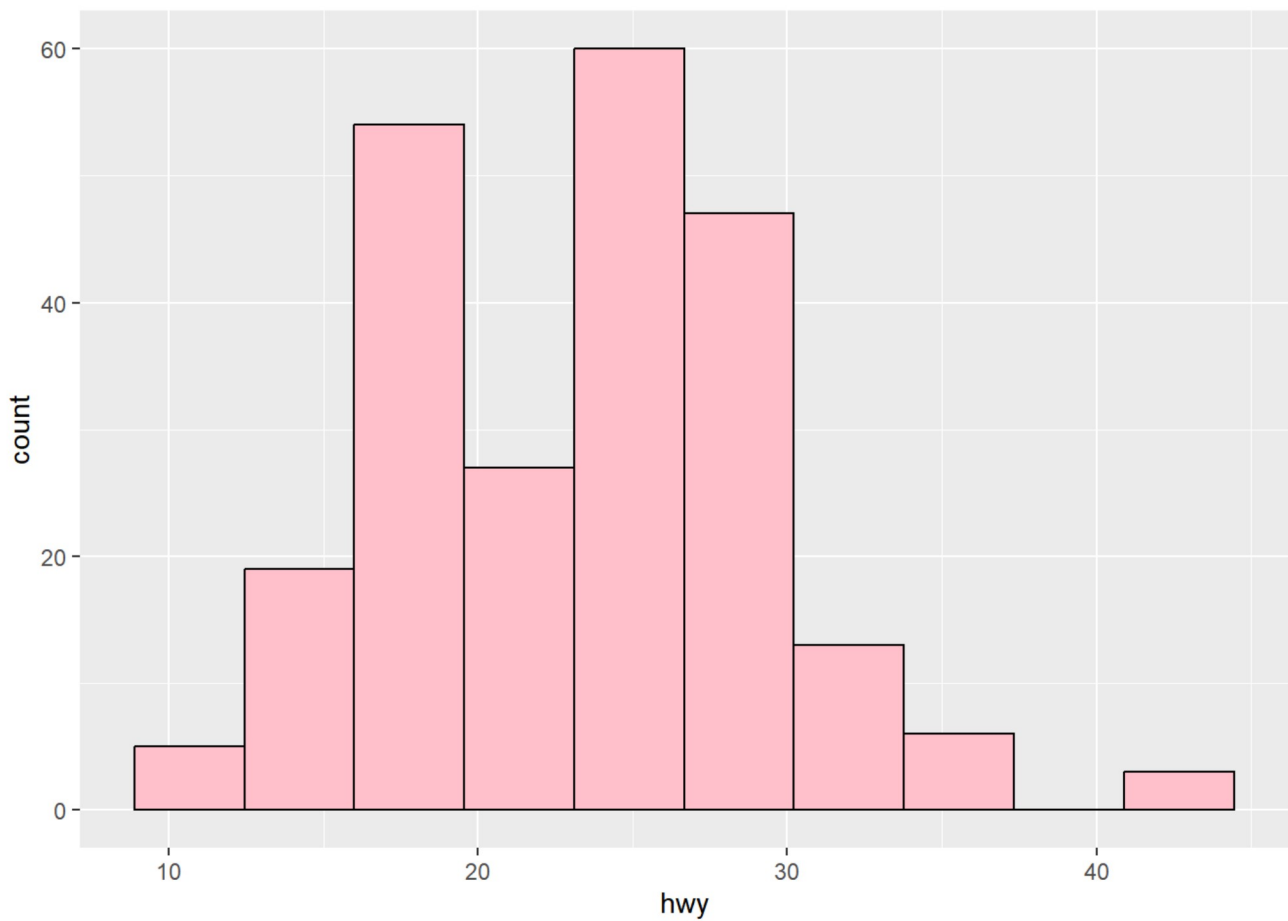




Bins, binwidth and colors

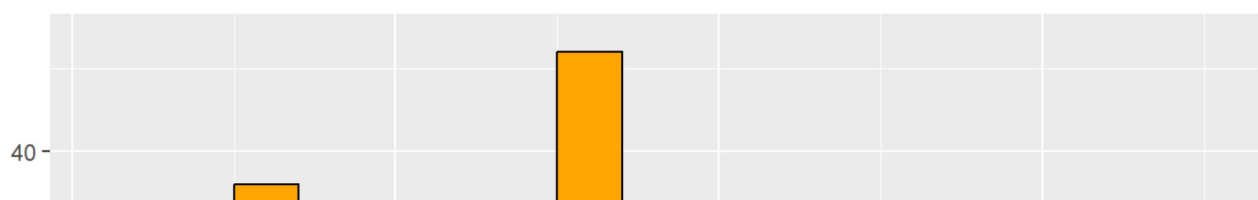
Number of bins and colors

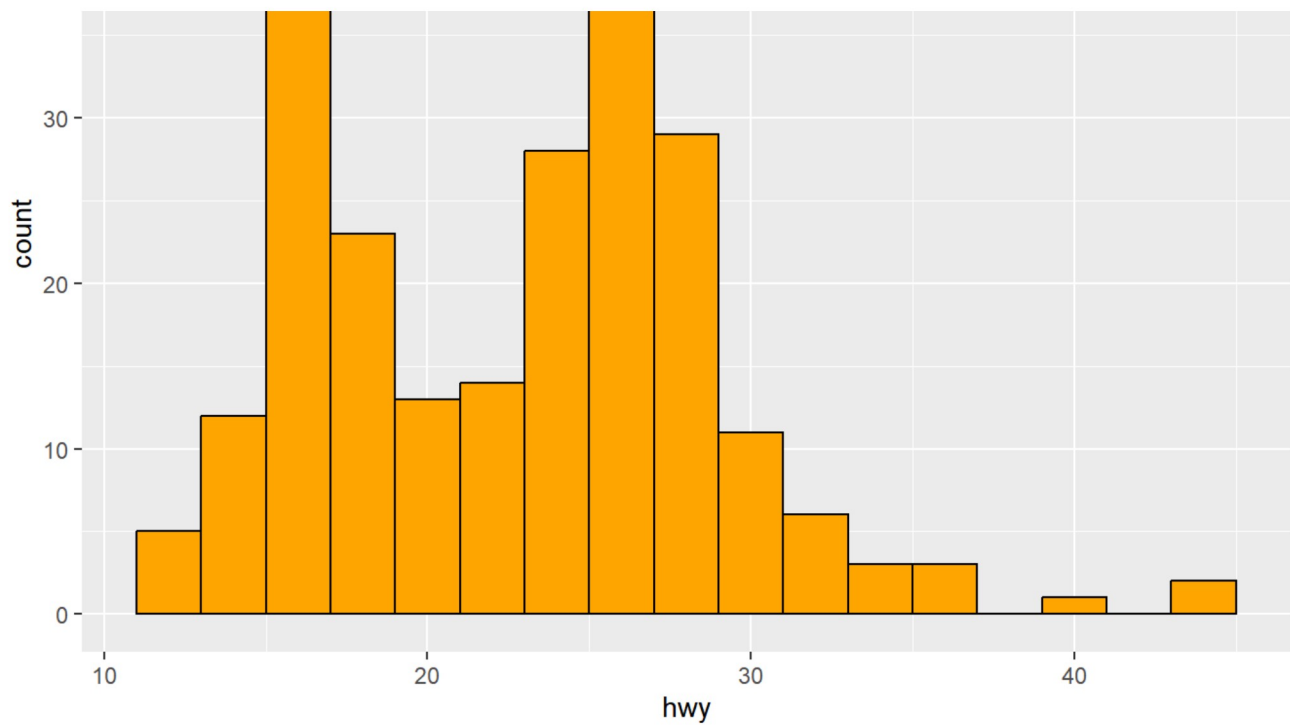
```
ggplot(data = mpg) +  
  geom_histogram(mapping = aes(x = hwy), bins = 10, color = "black", fill = "pink")
```



Binwidth

```
ggplot(data = mpg) +  
  geom_histogram(mapping = aes(x = hwy), binwidth = 2, color = "black", fill = "orange")
```

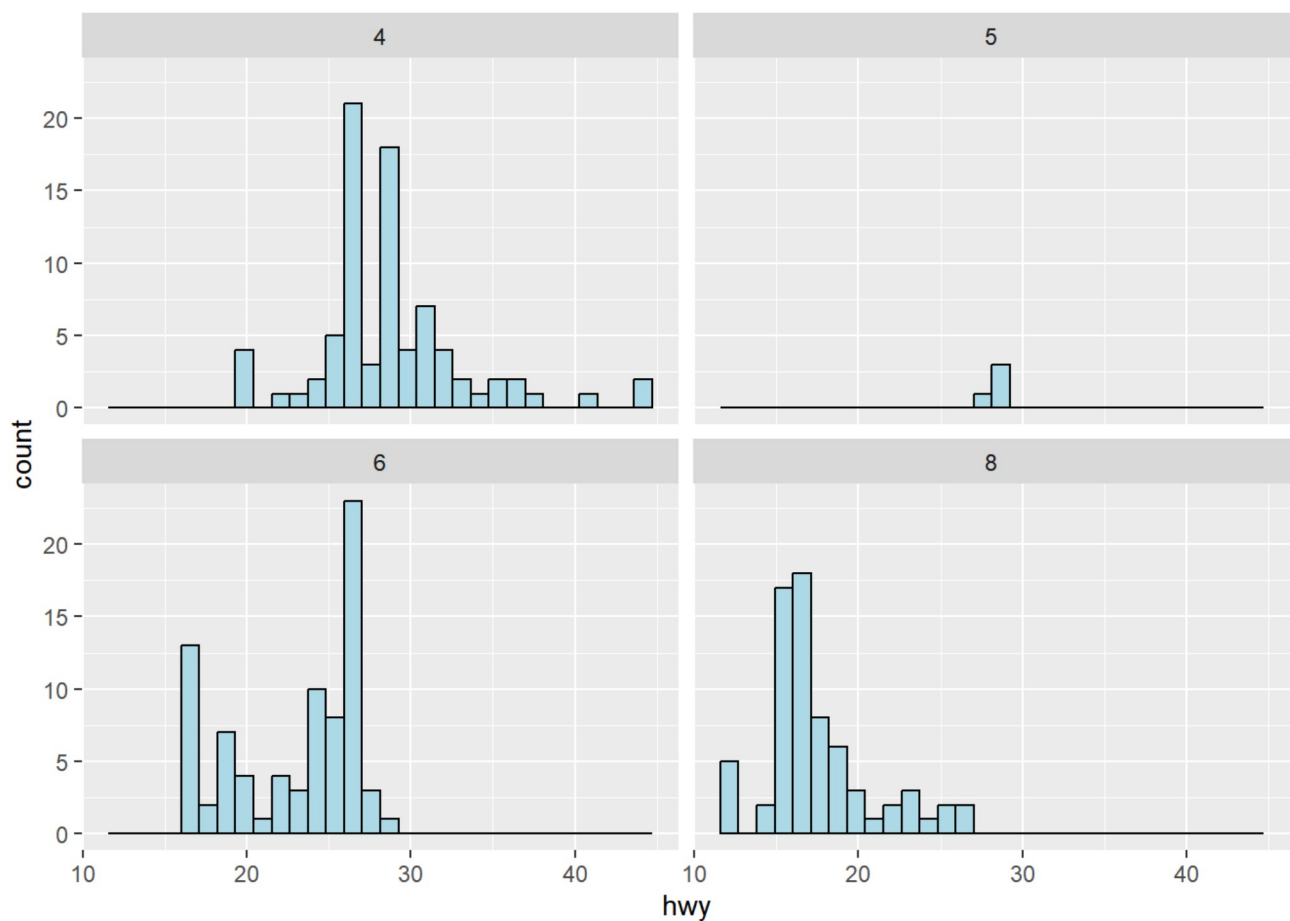




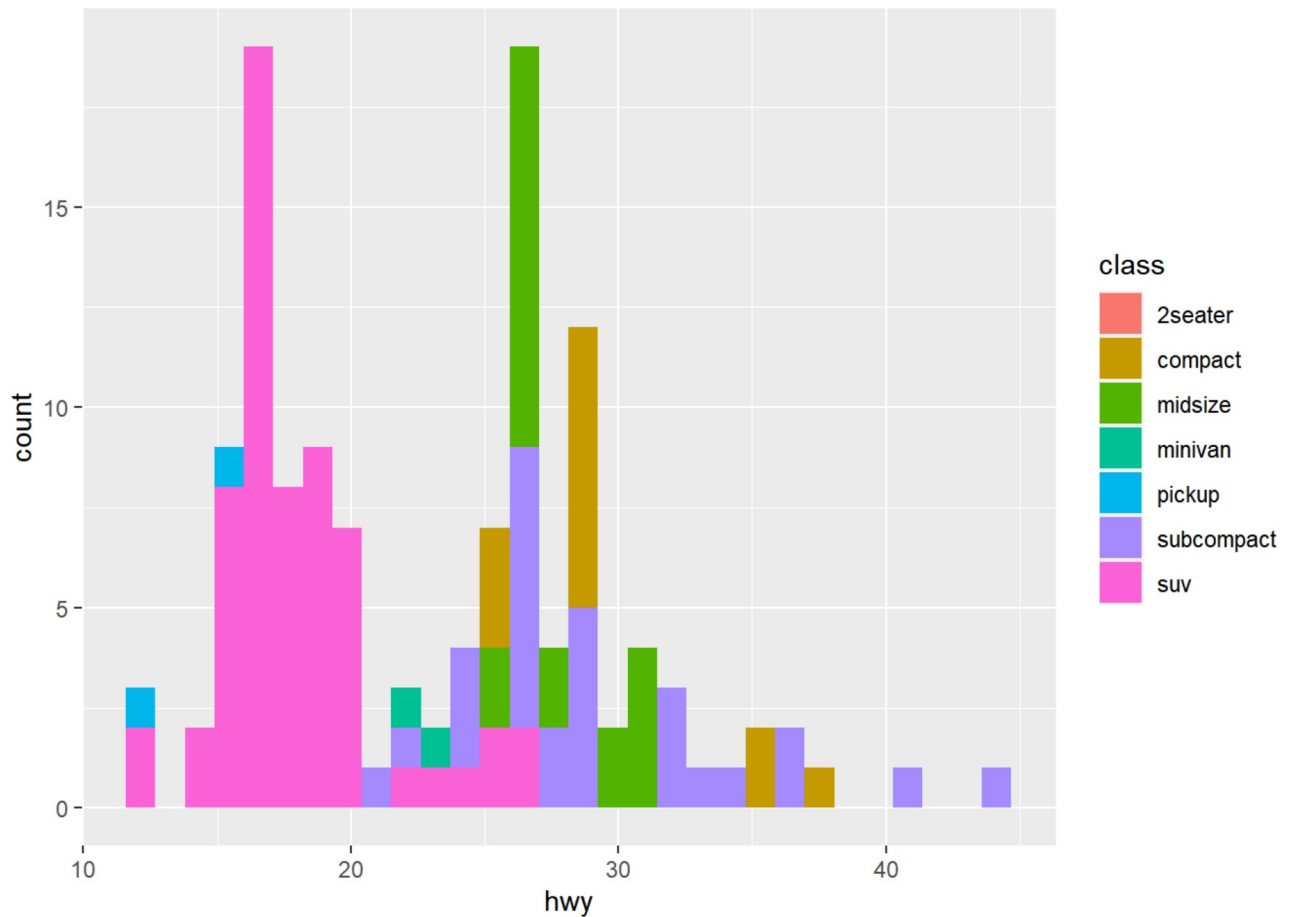
Histograms per group

Distribution of *hwy* (highway miles per gallon) grouping *cyl* (number of cylinders)

```
ggplot(mpg)+geom_histogram(mapping = aes(x=hwy),color="black",fill="lightblue")+facet_wrap('cyl')
```



```
ggplot(mpg)+geom_histogram(aes(x=hwy,fill=class),position="identity")
```

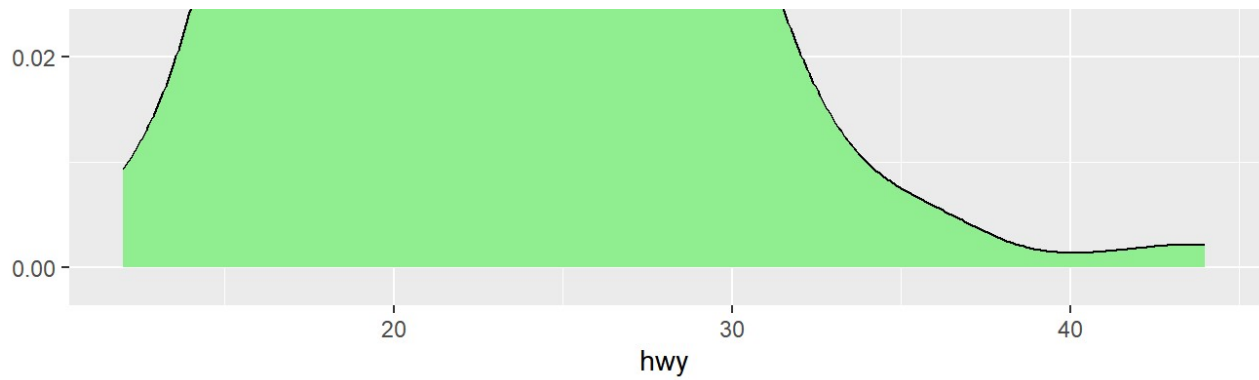


Density plots

Density Plot is also used to visualize the distribution of a numeric variable at continuous intervals. This chart is a variation of the Histogram that uses Kernel Smoother.

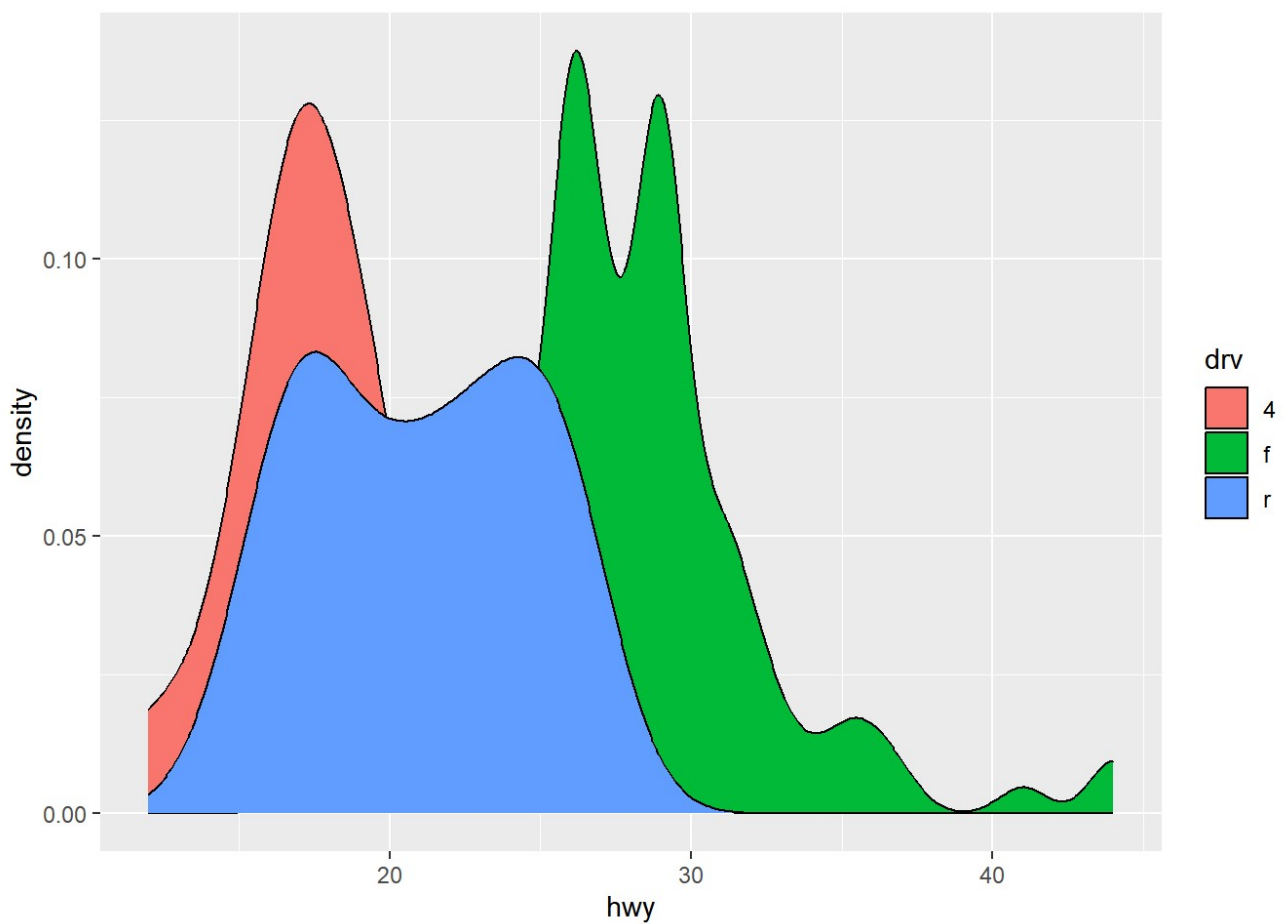
```
ggplot(data = mpg) +  
  geom_density(mapping = aes(x = hwy),fill="lightgreen")
```



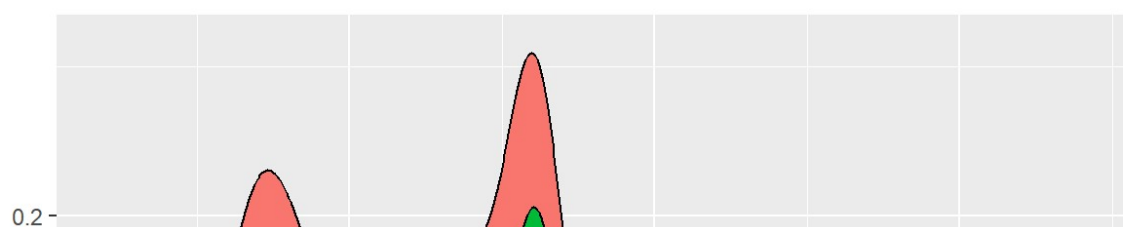


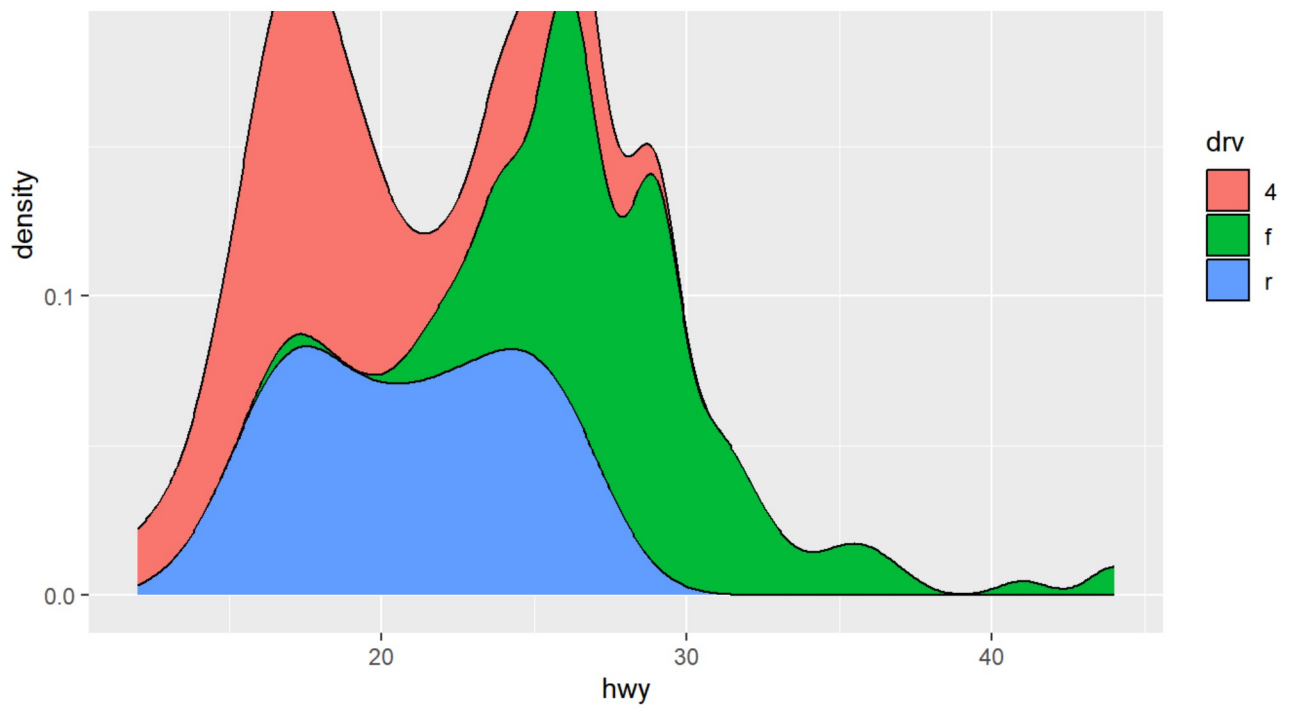
- Density plot grouping by *drv*.
- Positions= identity, stack and fill

```
ggplot(data = mpg) +
  geom_density(mapping = aes(x = hwy, fill=drv), position="identity")
```



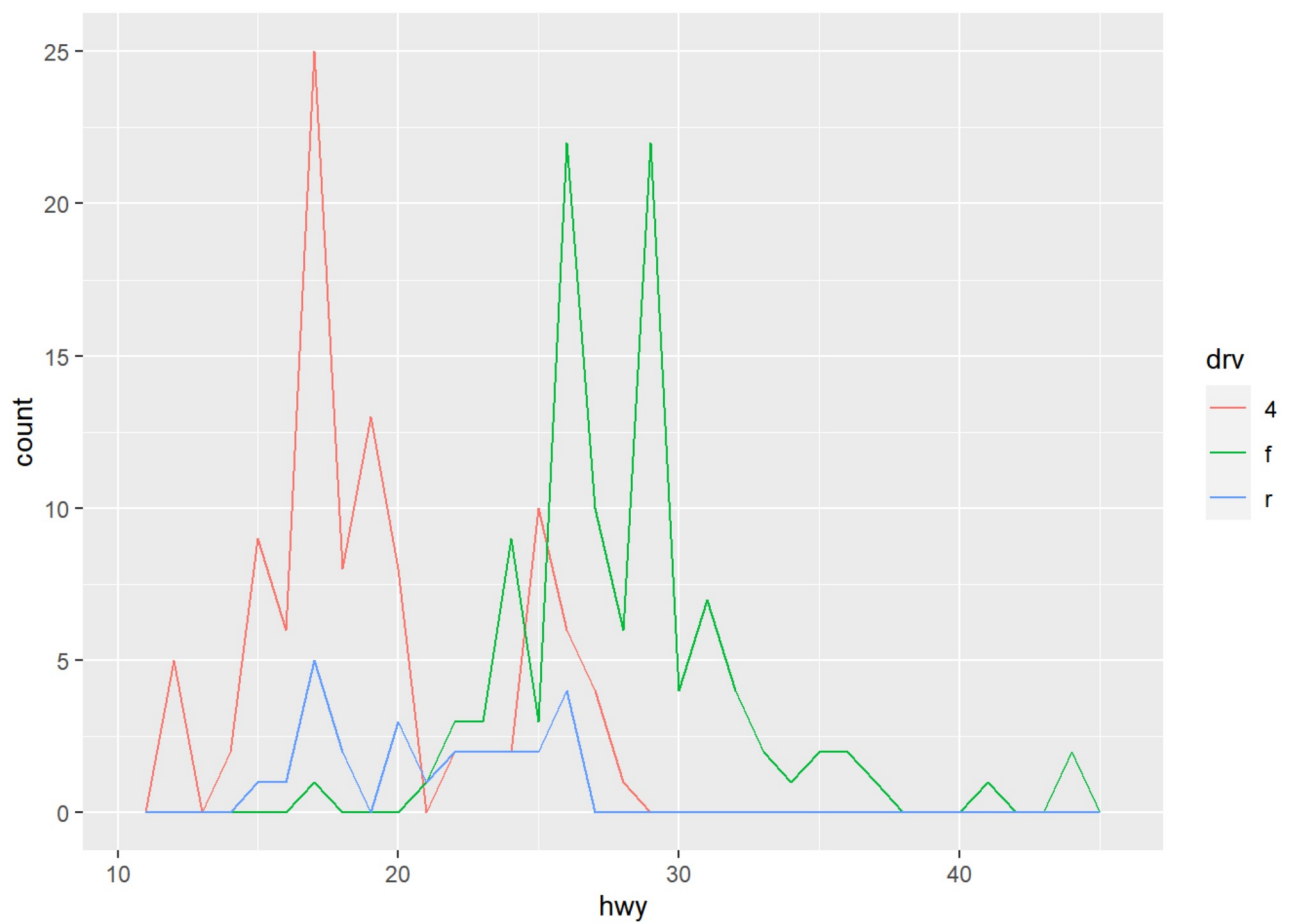
```
ggplot(data = mpg) +
  geom_density(mapping = aes(x = hwy, fill=drv), position="stack")
```





Frequency Polygons

```
ggplot(data = mpg, mapping = aes(x = hwy, colour = drv)) +  
  geom_freqpoly(binwidth = 1)
```

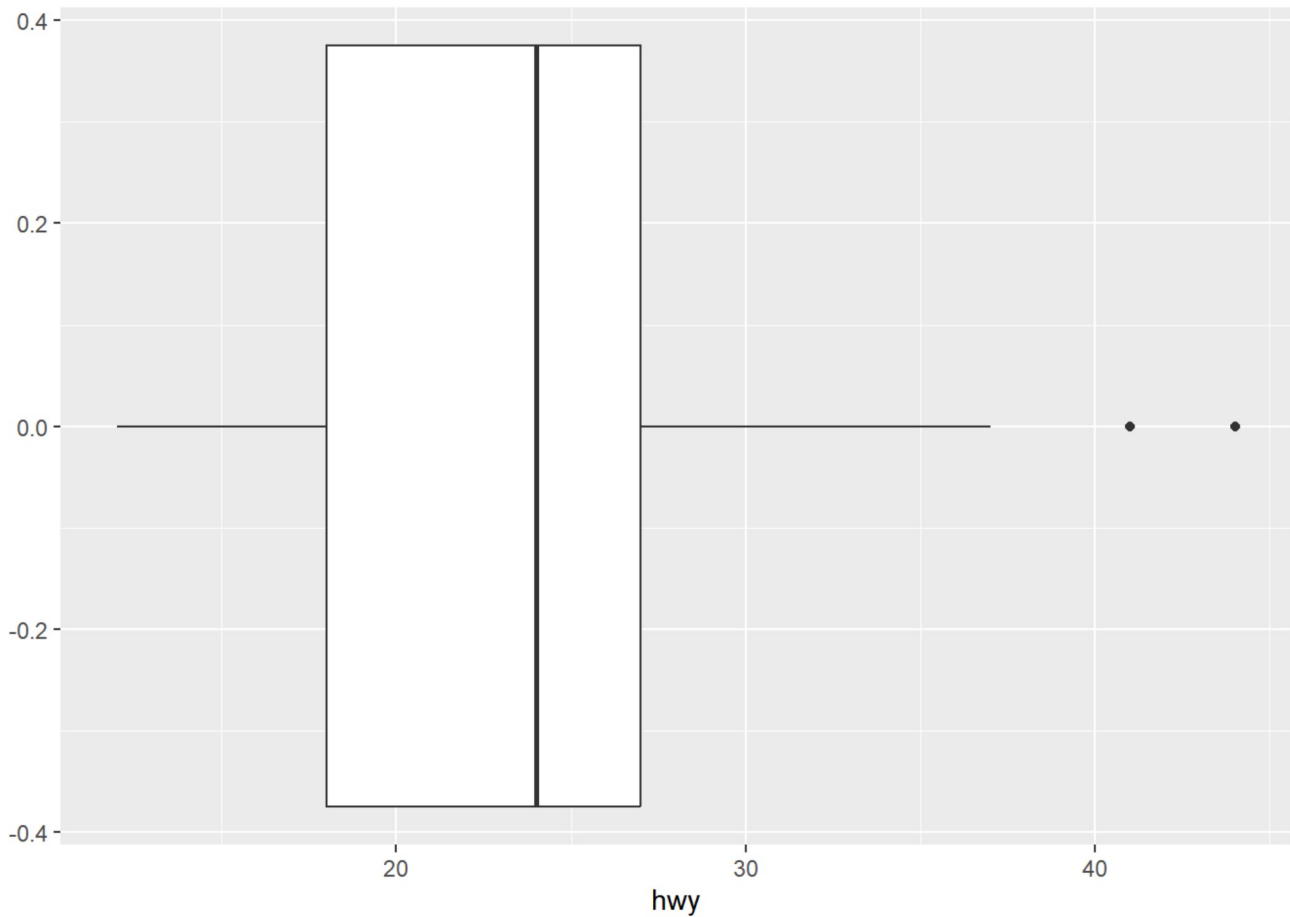


Box plots and vioplots

A boxplot chart give information about distribution and spread of the data.

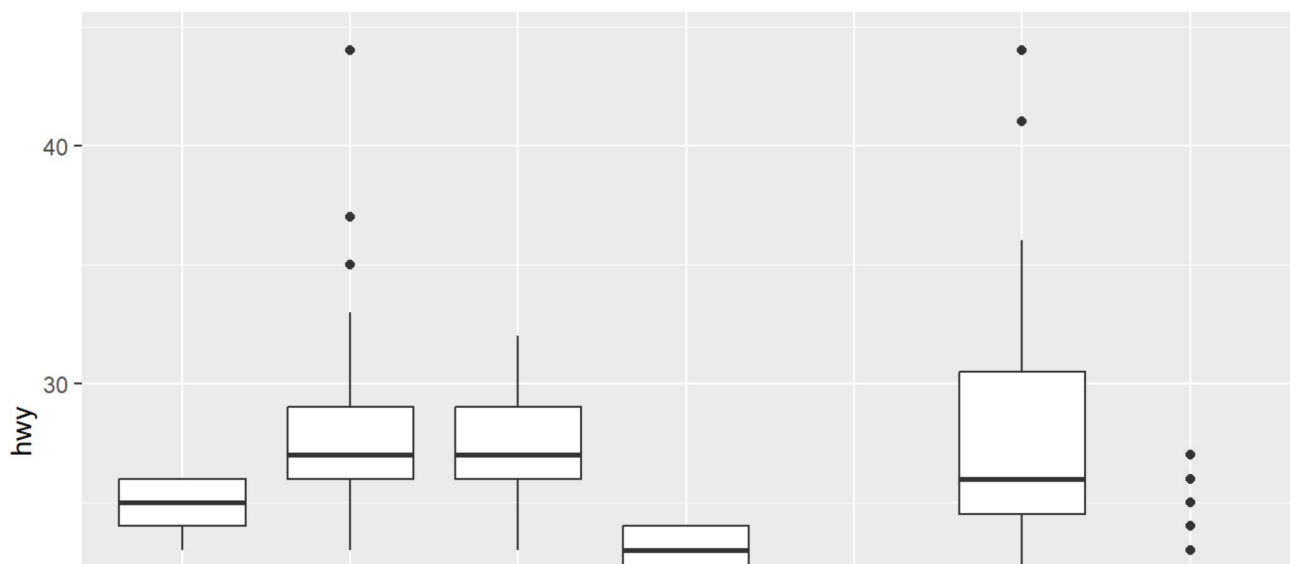
Box plot of hwy

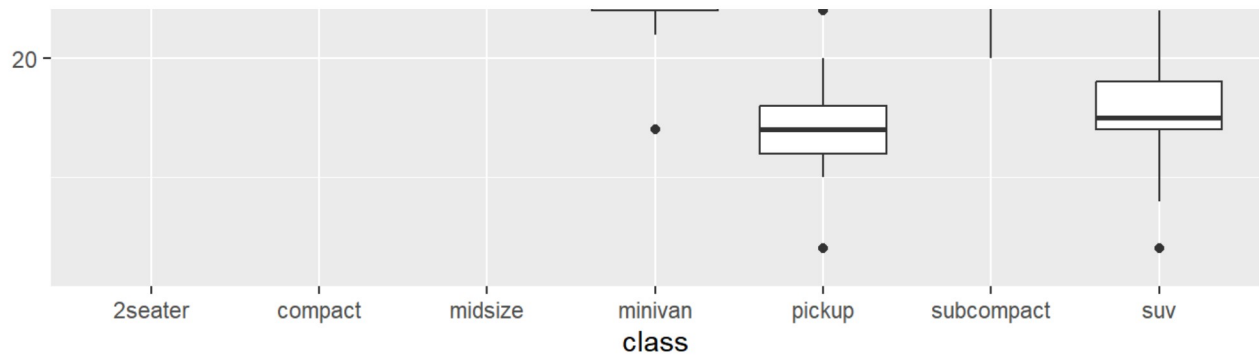
```
ggplot(data = mpg, aes(hwy)) + geom_boxplot()
```



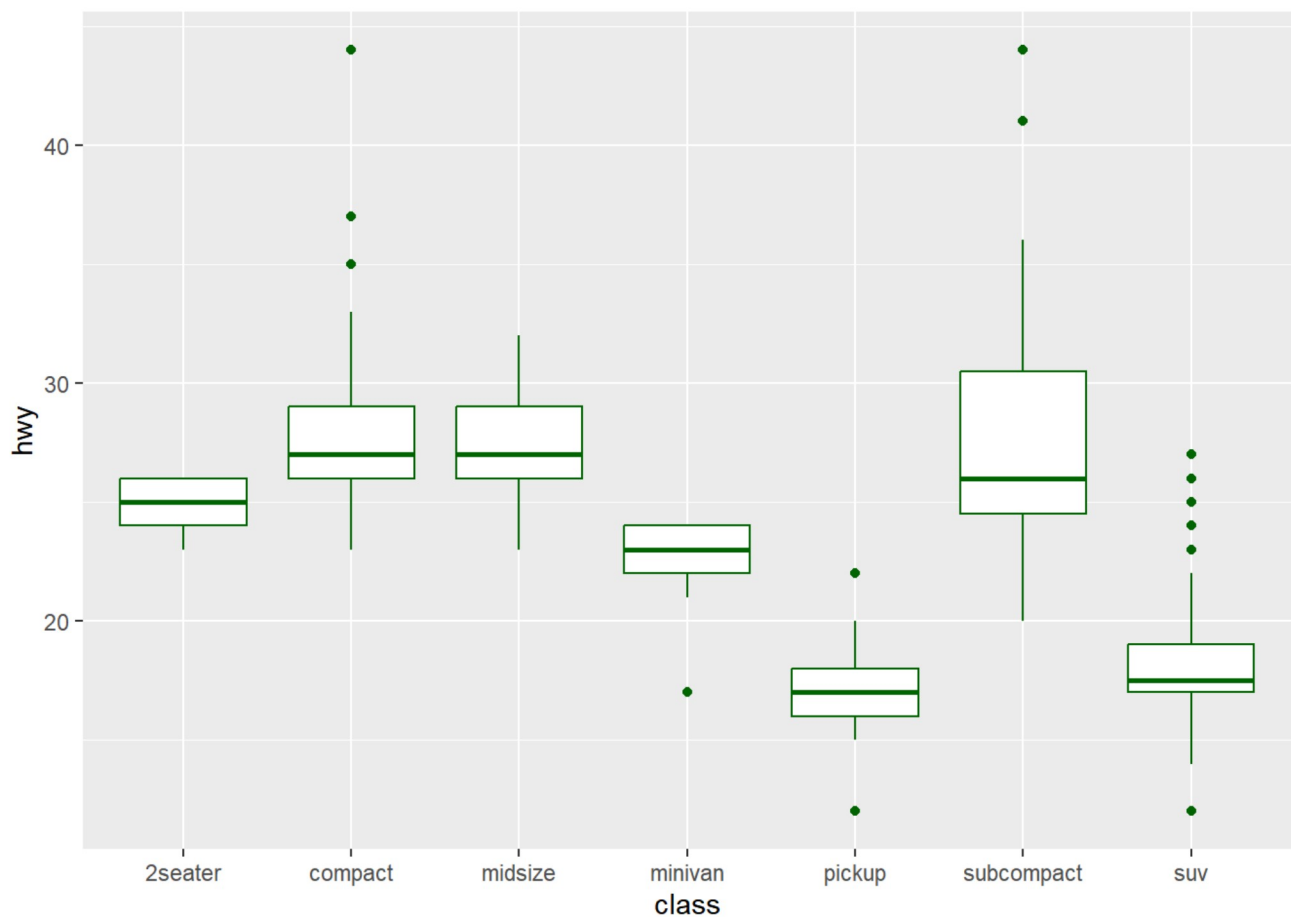
Box plots hwy versus class

```
ggplot(data = mpg, aes(class,hwy)) + geom_boxplot()
```





```
ggplot(data = mpg) + geom_boxplot(aes(class,hwy),fill = "white", colour = "darkgreen")
```



Vioplots

The violin plot gives a look at where the majority of the points are aggregated.

```
ggplot(data = mpg) + geom_violin(aes(class,hwy))
```

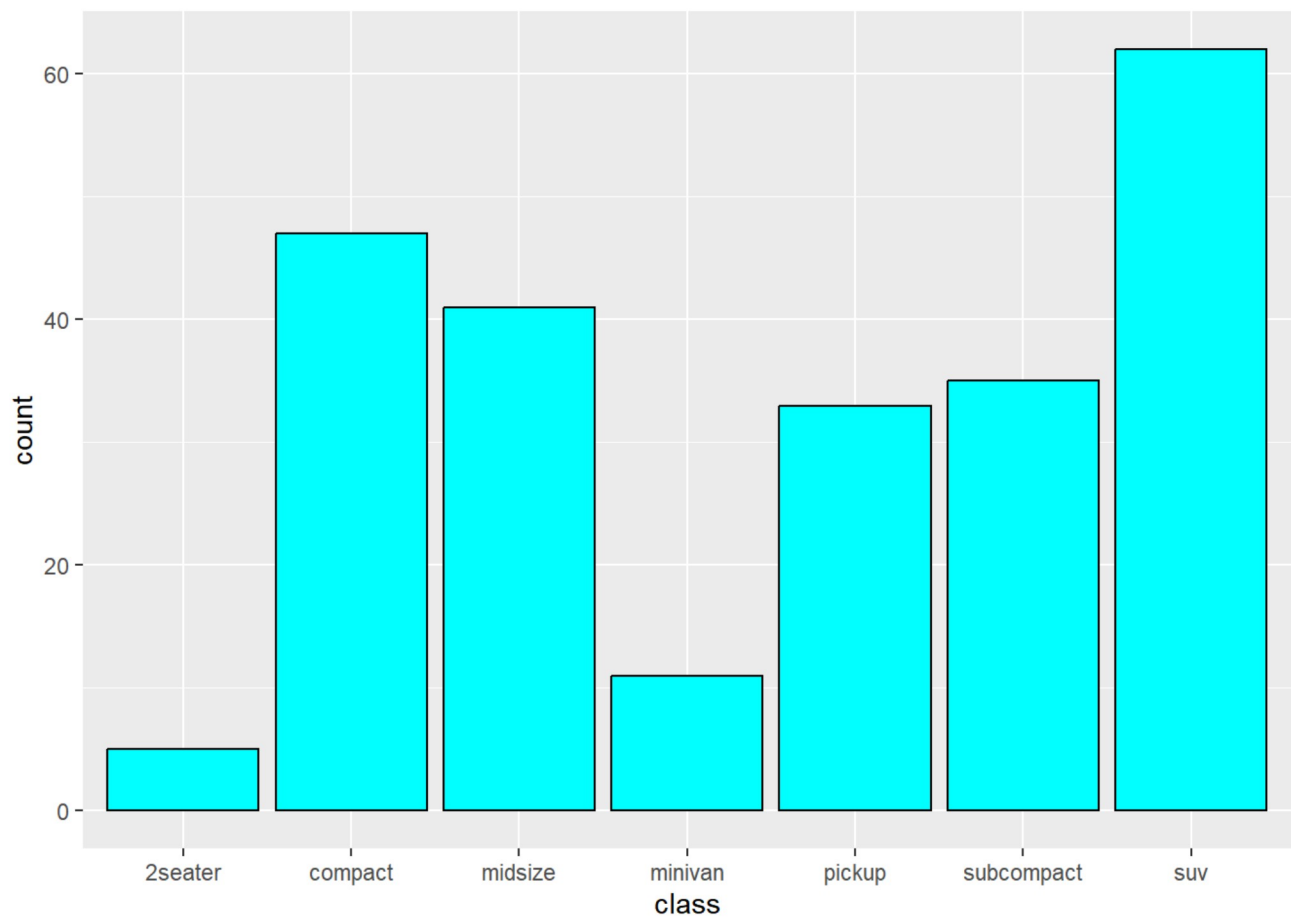




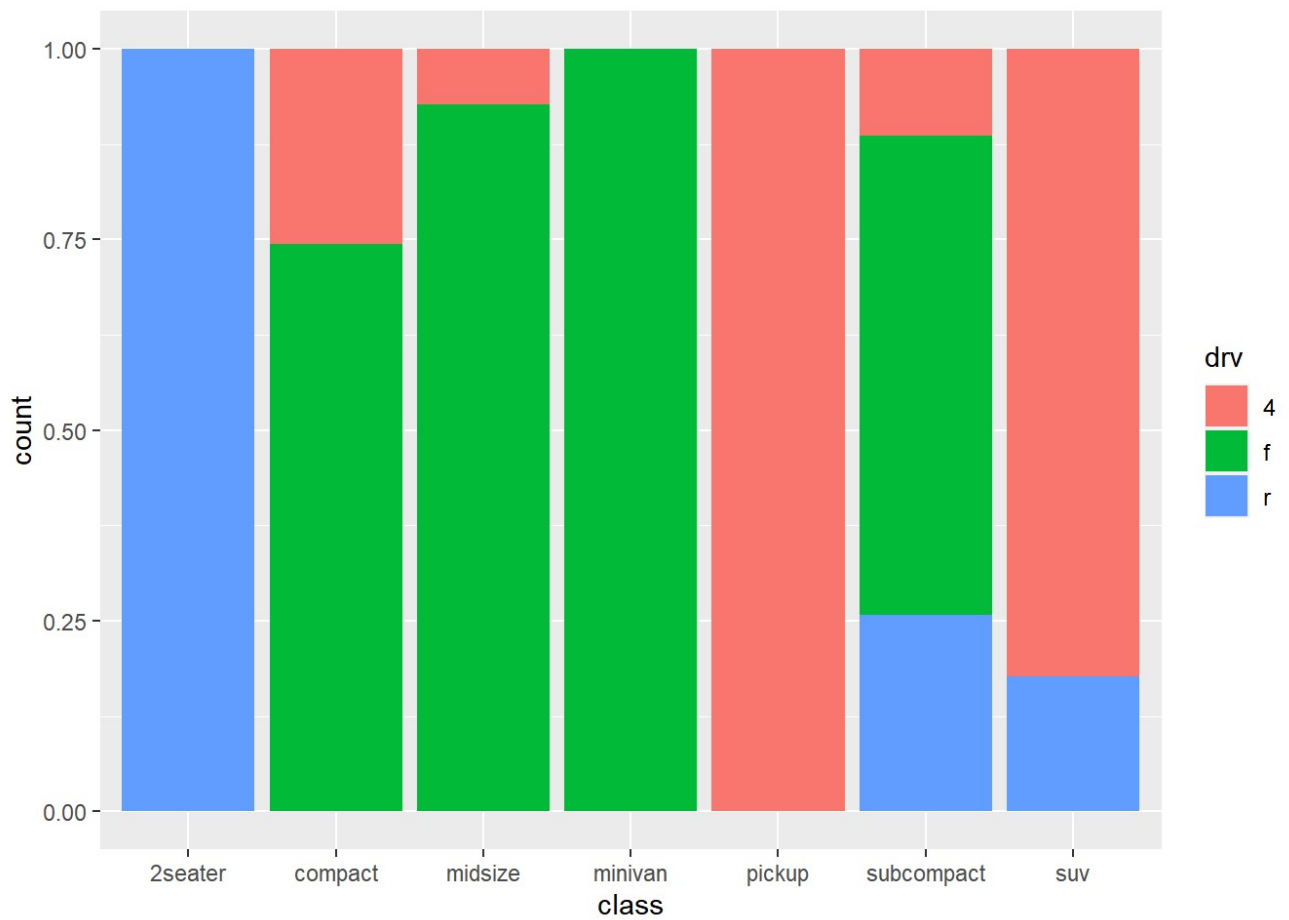
Graphics for qualitative data

Bar charts

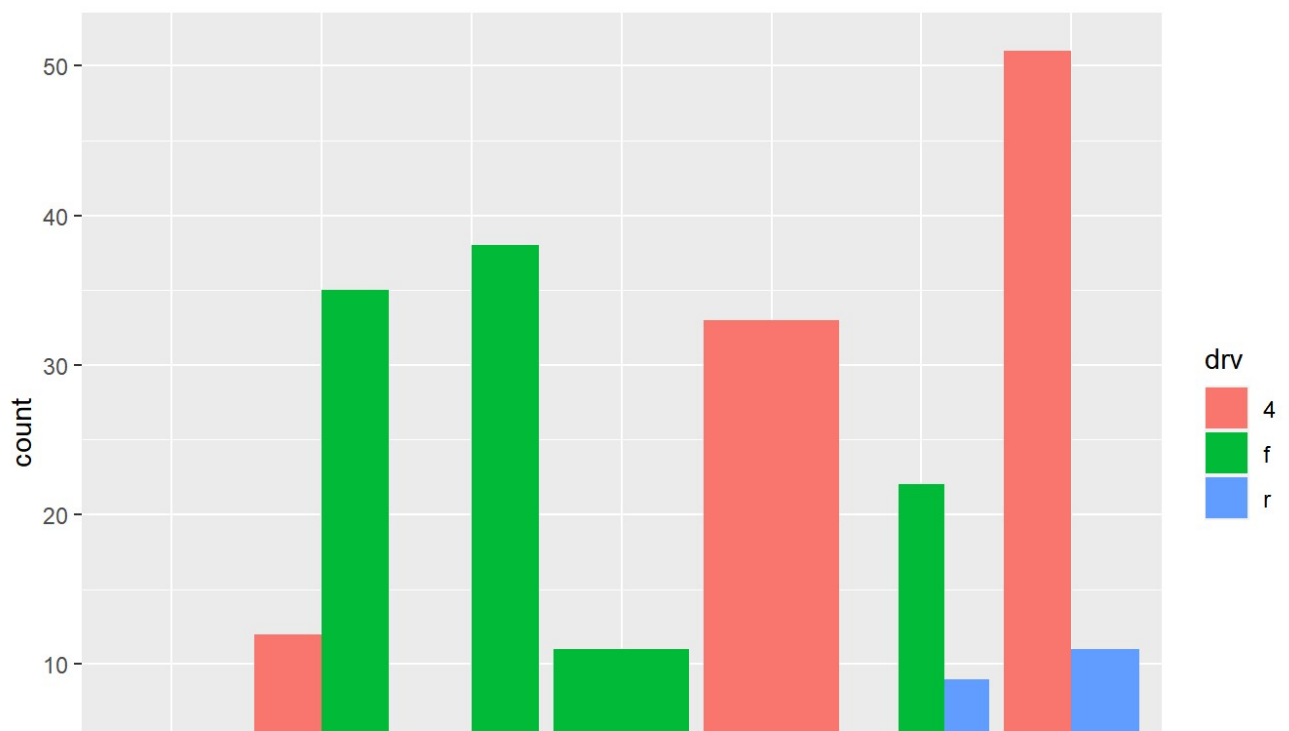
```
ggplot(data = mpg) +  
  geom_bar(mapping = aes(x = class), colour="black", fill="cyan")
```



```
ggplot(data = mpg) +  
  geom_bar(mapping = aes(x = class, fill = drv), position = "fill")
```



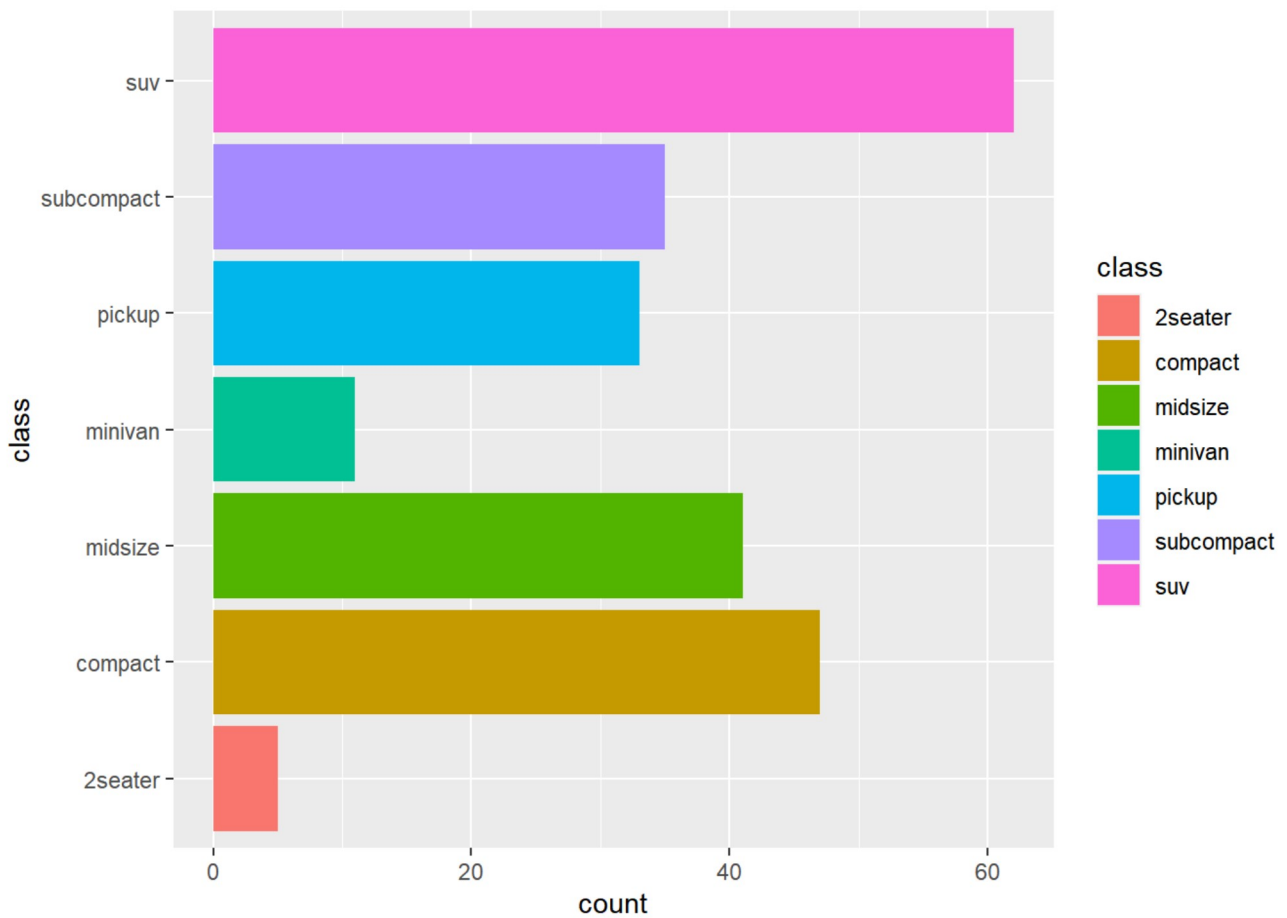
```
ggplot(data = mpg) +  
  geom_bar(mapping = aes(x = class, fill = drv), position = "dodge")
```



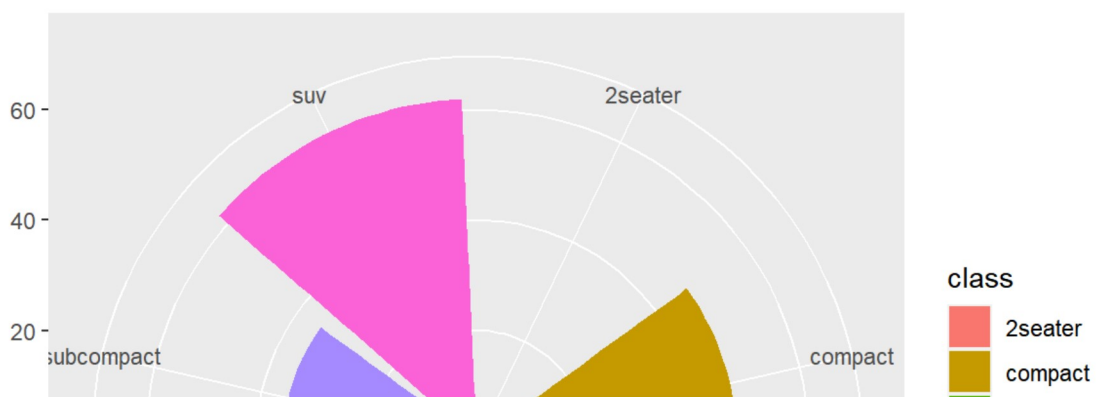


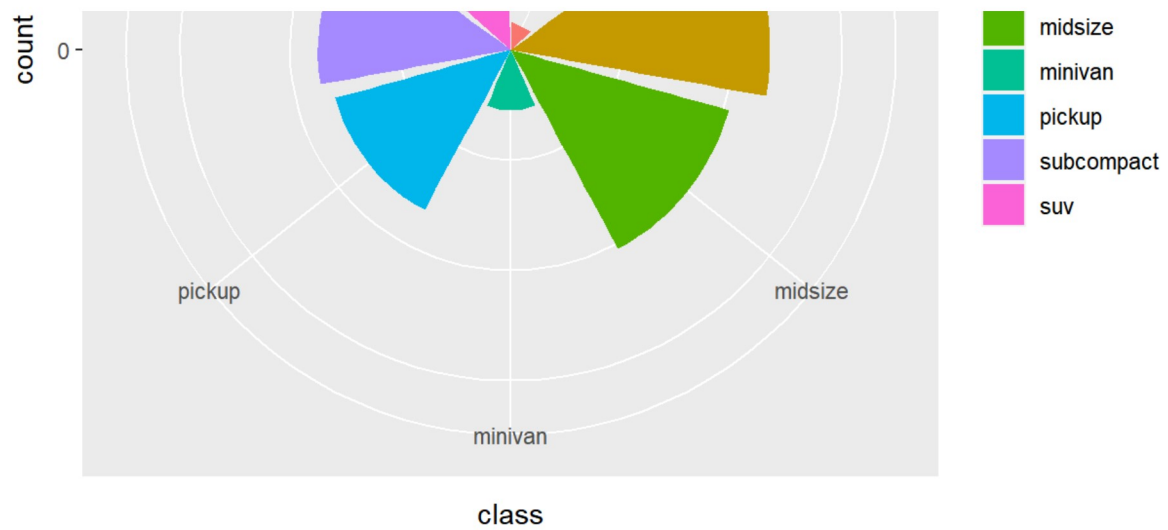
Some variations: flip an polar

```
ggplot(data = mpg) + geom_bar( mapping = aes(x = class, fill = class)) +
  coord_flip()
```



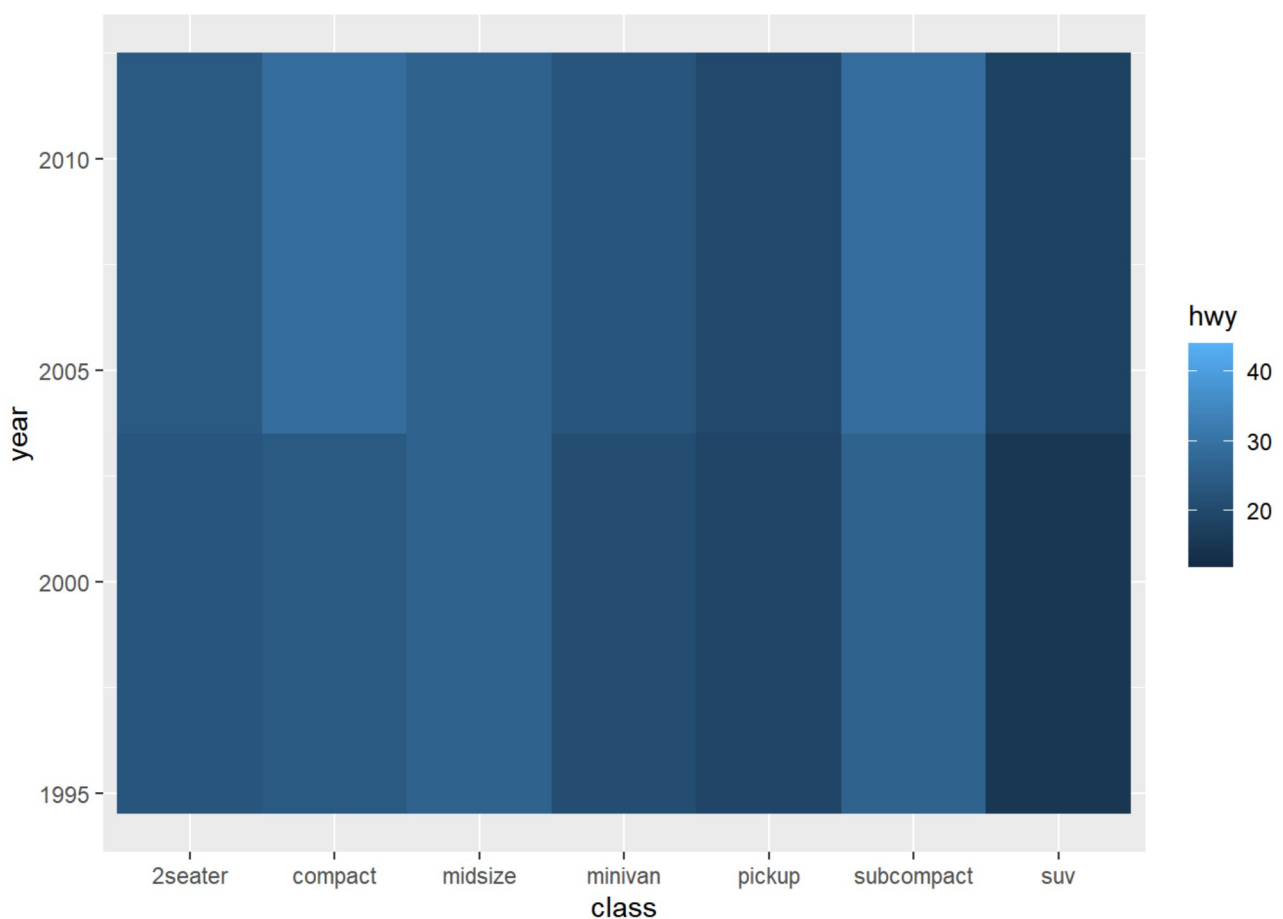
```
ggplot(data = mpg) + geom_bar( mapping = aes(x = class, fill = class)) +
  coord_polar()
```





Tile plot - heat map

```
ggplot(data = mpg) + geom_tile(mapping=aes(x=class,y=year,fill=hwy))
```



Descriptive statistics

Summary of all variables in a data set. R has several functions and packages for that.

Function `summary()` of basic R

```
summary(mpg)
```

```
manufacturer      model      displ      year
Length:234      Length:234      Min.   :1.600      Min.   :1999
Class :character  Class :character  1st Qu.:2.400      1st Qu.:1999
Mode  :character  Mode  :character  Median :3.300      Median :2004
                        Mean  :3.472      Mean  :2004
                        3rd Qu.:4.600      3rd Qu.:2008
                        Max.   :7.000      Max.   :2008

      cyl      trans      drv      cty
Min.   :4.000      Length:234      Length:234      Min.   : 9.00
1st Qu.:4.000      Class :character  Class :character  1st Qu.:14.00
Median :6.000      Mode  :character  Mode  :character  Median :17.00
Mean   :5.889                                     Mean  :16.86
3rd Qu.:8.000                                     3rd Qu.:19.00
Max.   :8.000                                     Max.   :35.00

      hwy      fl      class
Min.   :12.00      Length:234      Length:234
1st Qu.:18.00      Class :character  Class :character
Median :24.00      Mode  :character  Mode  :character
Mean   :23.44
3rd Qu.:27.00
Max.   :44.00
```

Another way

Using %>% (pipe operator)

```
mpg %>%
  summarise(mean = mean(hwy), sd = sd(hwy), n = n())
```

```
# A tibble: 1 × 3
  mean    sd    n
<dbl> <dbl> <int>
1  23.4  5.95  234
```

```
mpg %>%
  group_by(class) %>%
  summarise(mean = mean(hwy), sd = sd(hwy), median = median(hwy), n = n())
```

```
# A tibble: 7 × 5
  class      mean    sd median    n
<chr>    <dbl> <dbl> <dbl> <int>
1 2seater    24.8  1.30   25     5
2 compact    28.3  3.78   27    47
3 midsize    27.3  2.14   27    41
4 minivan    22.4  2.06   23    11
5 pickup     16.9  2.27   17    33
6 subcompact 28.1  5.38   26    35
7 suv        18.1  2.98   17    62
```

```
7 suv      18.1  2.98  17.5    62
```

Table formatting

R has many functions and packages to format table (stargaze, etc). We are using *knitr* package and function *kable()*. We stored the results in a object named *descriptives*.

```
descriptives<-mpg %>%  
  group_by(class) %>%  
  summarise(mean = mean(hwy), sd = sd(hwy), median = median (hwy), n = n())
```

```
knitr::kable(descriptives)
```

class	mean	sd	median	n
2seater	24.80000	1.303840	25.0	5
compact	28.29787	3.781620	27.0	47
midsize	27.29268	2.135930	27.0	41
minivan	22.36364	2.062655	23.0	11
pickup	16.87879	2.274280	17.0	33
subcompact	28.14286	5.375012	26.0	35
suv	18.12903	2.977973	17.5	62

Package skimr

skimr is designed to provide summary statistics about variables in data frames, tibbles, data tables and vectors. It is opinionated in its defaults, but easy to modify.

In base R, the most similar functions are *summary()* for vectors and data frames and *fivenum()* for numeric vectors

```
library(skimr)  
skim(mpg)
```

Data summary






Name	mpg
Number of rows	234
Number of columns	11
Column type frequency:	
character	6
numeric	5

Group variables	None
-----------------	------

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
manufacturer	0	1	4	10	0	15	0
model	0	1	2	22	0	38	0
trans	0	1	8	10	0	10	0
drv	0	1	1	1	0	3	0
fl	0	1	1	1	0	5	0
class	0	1	3	10	0	7	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
displ	0	1	3.47	1.29	1.6	2.4	3.3	4.6	7	
year	0	1	2003.50	4.51	1999.0	1999.0	2003.5	2008.0	2008	
cyl	0	1	5.89	1.61	4.0	4.0	6.0	8.0	8	
cty	0	1	16.86	4.26	9.0	14.0	17.0	19.0	35	
hwy	0	1	23.44	5.95	12.0	18.0	24.0	27.0	44	

Calling functions

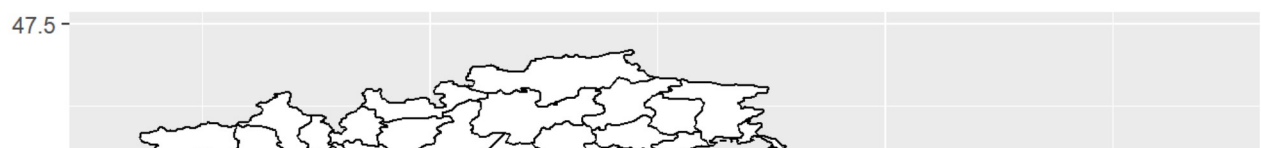
```
{r} elisa_henning <- 16 elisa_henning}
```

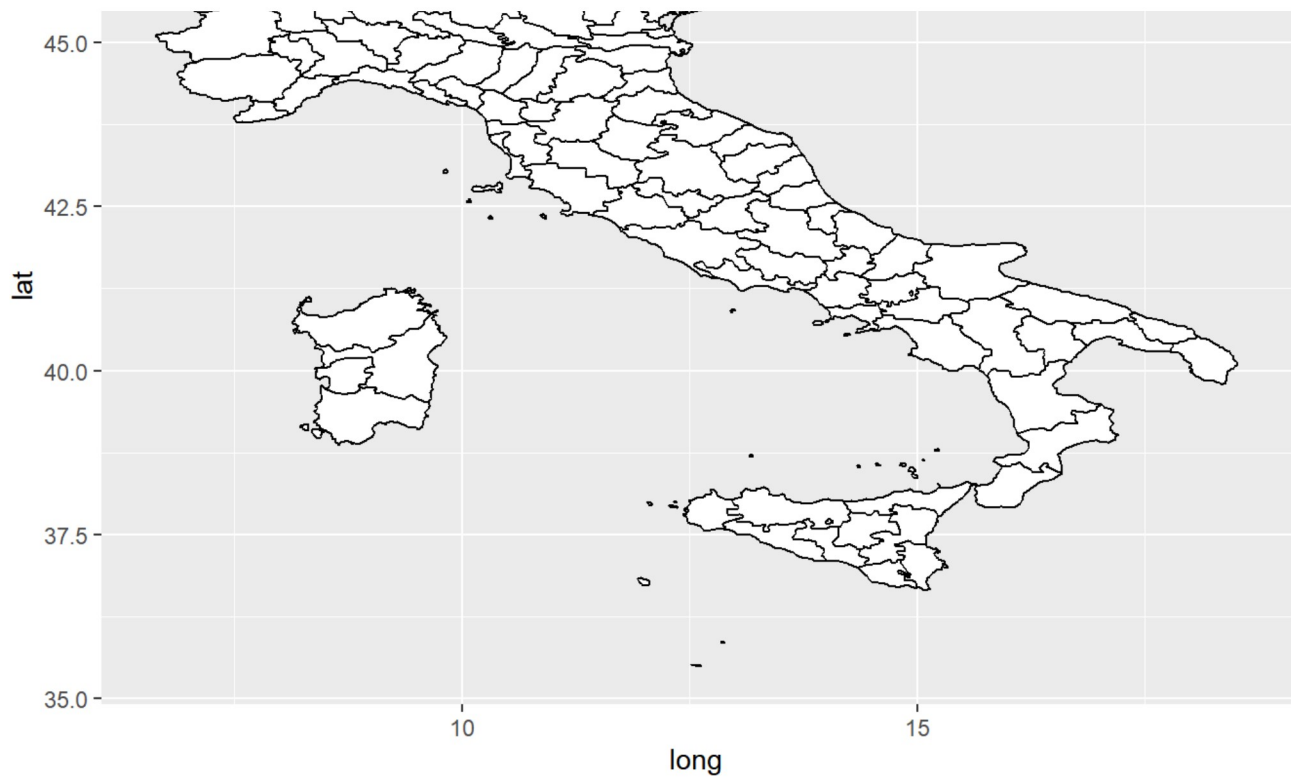
```
{r} descriptives<-mpg %>%   group_by(class) %>%   summarise(mean = mean(hwy), sd = sd(hwy),
median = median (hwy), n = n())}
```

Maps

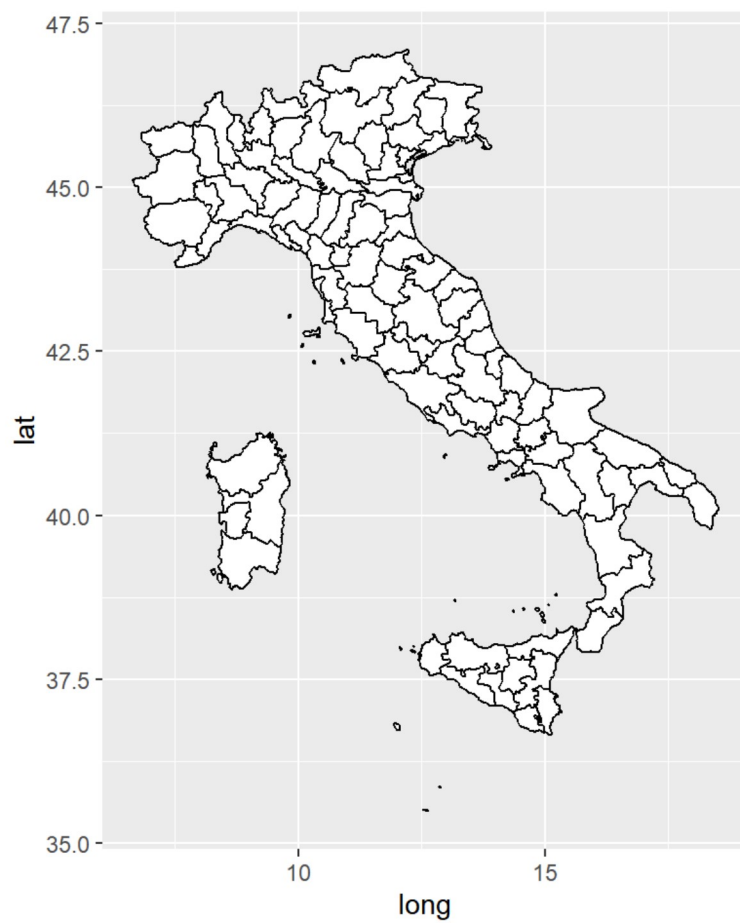
```
it <- map_data("italy")

ggplot(it, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", colour = "black")
```





```
ggplot(it, aes(long, lat, group = group)) +  
  geom_polygon(fill = "white", colour = "black") +  
  coord_quickmap()
```



Reports and Figures

- Compile a report in html, Microsoft word or pdf (LaTeX): [video](#)
 - It is necessary to install some additional packages (markdown, knitr,...).
- Export and save the plots: [video](#)

References

The references used in this document:

[R for Data Science \(2e\) \(hadley.nz\)](#)

[An Introduction to ggplot2](#)

[ggplot2 cheatsheets](#)