PSC 505- Introduction to ggplot2

Ezgi Siir Kibris

8/29/2020

Contents

1.Installation **2**

2.Grammer of graphics **2**

2.1. Key components 2

3. Building a plot layer by layer **2**

3.1. Basic setup 3

3.2. The Layers 3

3.2.1. Scatter Plot 4

3.2.2. Line Chart 6

3.2.3. Bar Chart 8

3.2.4. Boxplot 9

3.3. The Labels 10

3.4. The Theme 11

3.5. The Facets 14

4. Time series plots **15**

5. Playing with colors **17**

5.1. RColorBrewer palettes 17

5.2. Wes Anderson movie color palettes 19

5.3. Create your own color palette 20

6. References **21**

1.Installation

install.packages(“ggplot2”)

2.Grammer of graphics

The ggplot2 package, created by Hadley Wickham, offers a powerful graphics language for creating elegant and complex plots. It is originally based on Wilkinson’s The Grammar of Graphics (2005).

2.1. Key components

1. **Data**: data set
2. **Aesthetic mapping**: x and y variables, aes
3. **Layers**: geometric elements, geoms: geom_point(), geom_smooth(), geom_bar(),…
4. **Labels**: title, x and y axis labels
5. **Theme**: Font size, background, legend
6. **Facet**: Number of graphs, groups by which we divide the data.

- put + sign between each layer. + sign has to come at the end of the line, not the start!

  Don’t do that!

  ggplot(mpg,aes(displ,hwy))

  +geom_point()

3. Building a plot layer by layer

Using fuel economy data set, we are going to plot figures layer by layer. This data set includes information about the fuel economy of popular car models in 1999 and 2008, collected by the US Environmental Protection Agency.

The variables of data set:

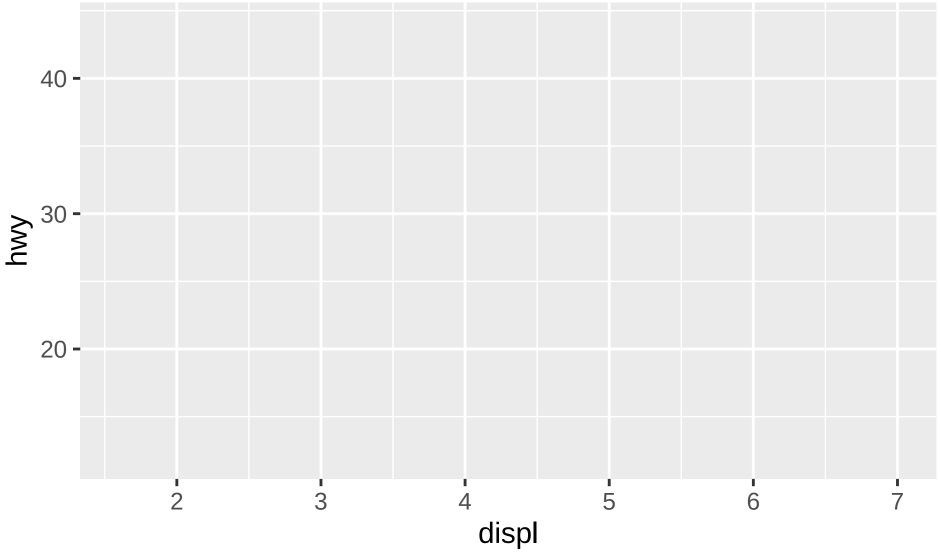
- *cty* and *hwy*: record miles per gallon (mpg) for city and highway driving.
- *displ*: the engine displacement in litres.
- *model*: model of car
- *drv*: drive train, front wheel (f), rear wheel (r) or four wheel (4).
- *class*: type of car: two seater, SUV, compact, etc.

3.1. Basic setup

```
library(ggplot2)
```

Here in this graph, data set is mpg, aesthetic mapping is defined by engine size mapped to x axis, and fuel economy to y axis.

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



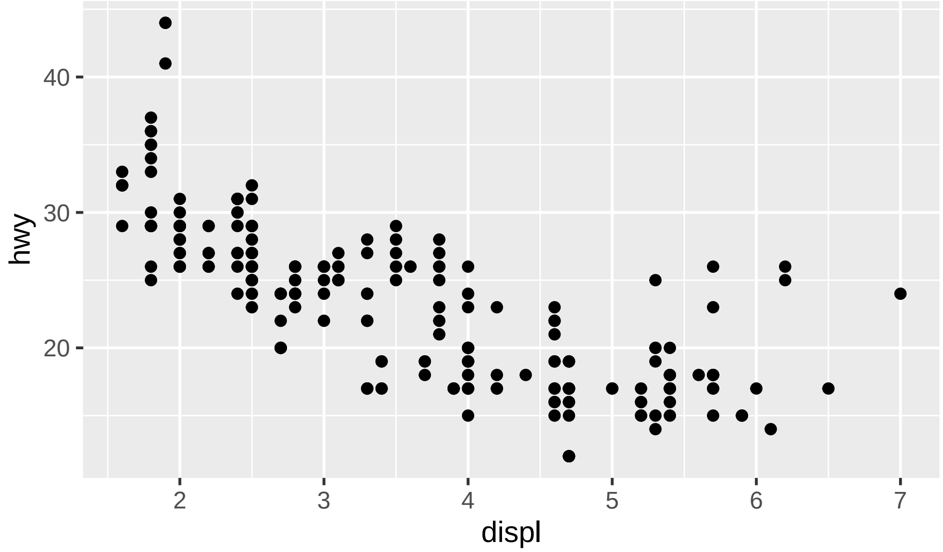
3.2. The Layers

Some useful geom functions:

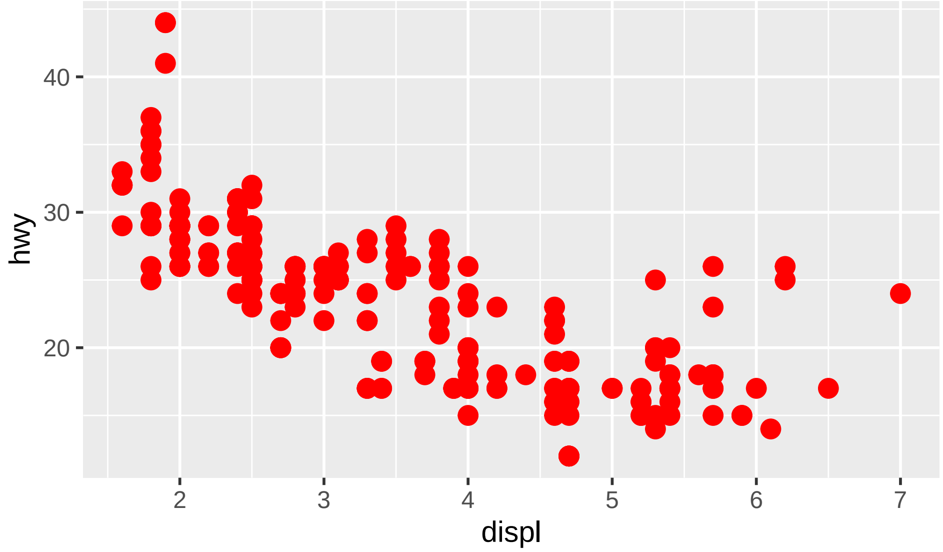
- **geom_point()**: Scatter plot
- **geom_smooth()**: Fits a smoother to the data and displays the smooth and its standard error.
- **geom_line()**: Draws lines between the data points.
- **geom_bar()**: Barchart, shows the distribution of categorical variables
- **geom_boxplot()**: Produces a box-and-whisker plot to summarize the distribution of a set of points.
- **geom_jitter()** : Adds some random noise to scatter plot to avoid overplotting.
- **geom_abline()**: Draws regression line specifying the intercept and the slope of the line.

3.2.1. Scatter Plot

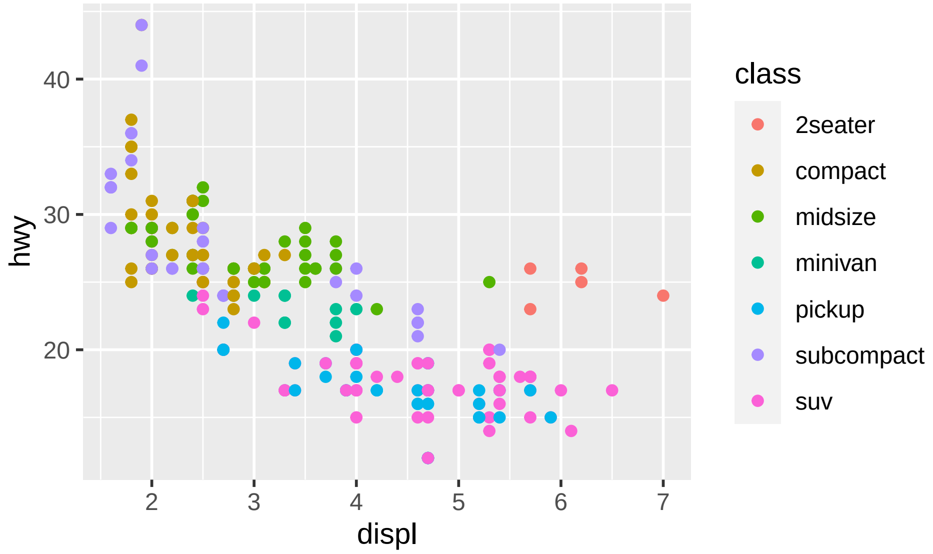
```
ggplot(mpg, aes(displ,hwy)) + geom_point()
```



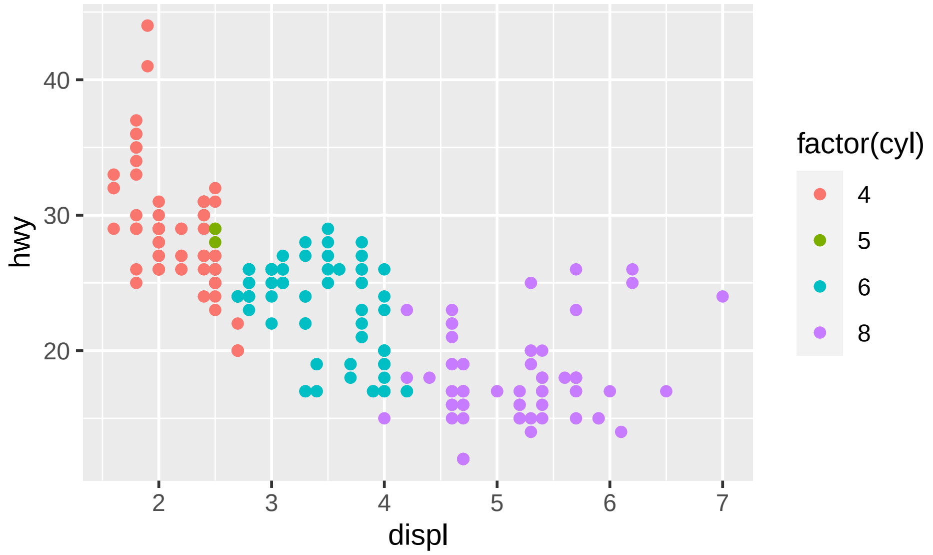
```
ggplot(mpg, aes(displ,hwy)) + geom_point(size=3,color="red")
```



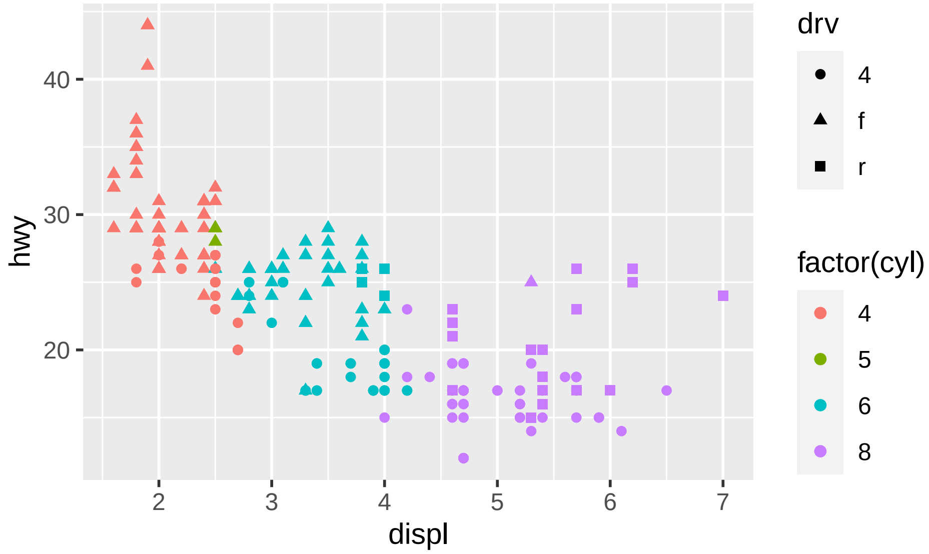
```
ggplot(mpg, aes(displ,hwy, color=class)) +
  geom_point()
```



```
ggplot(mpg, aes(displ,hwy, color=factor(cyl))) +
  geom_point()
```

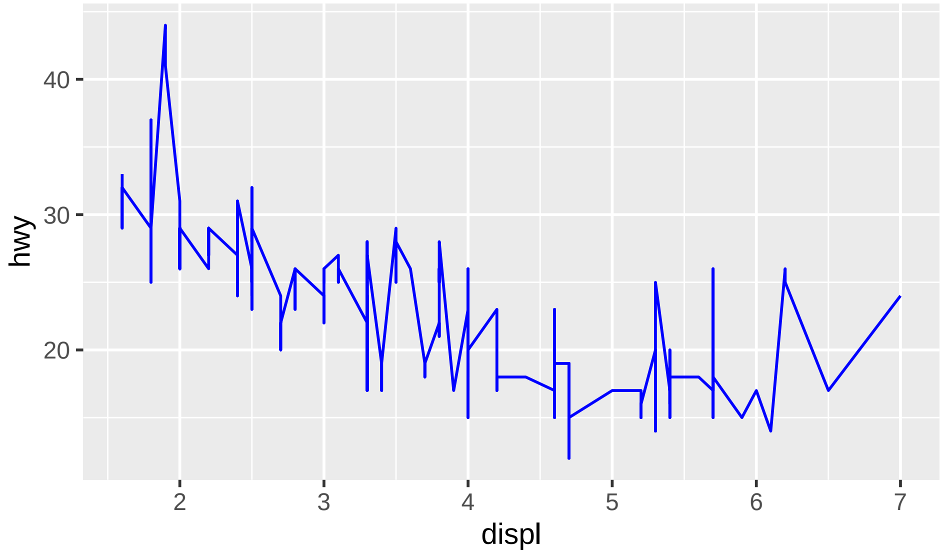


```
ggplot(mpg, aes(displ,hwy, color=factor(cyl))) +
  geom_point(aes(shape=drv))
```

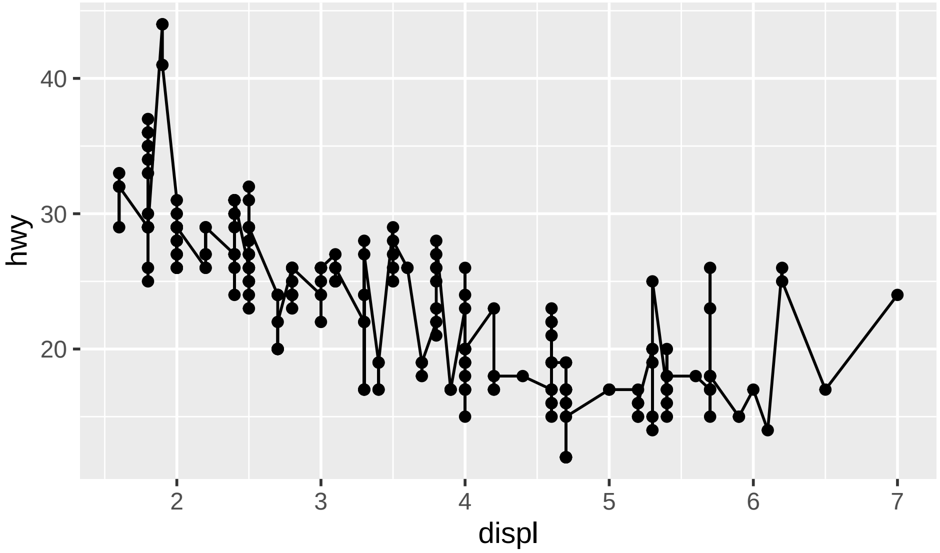


3.2.2. Line Chart

```
ggplot(mpg, aes(displ,hwy)) + geom_line(color="blue")
```



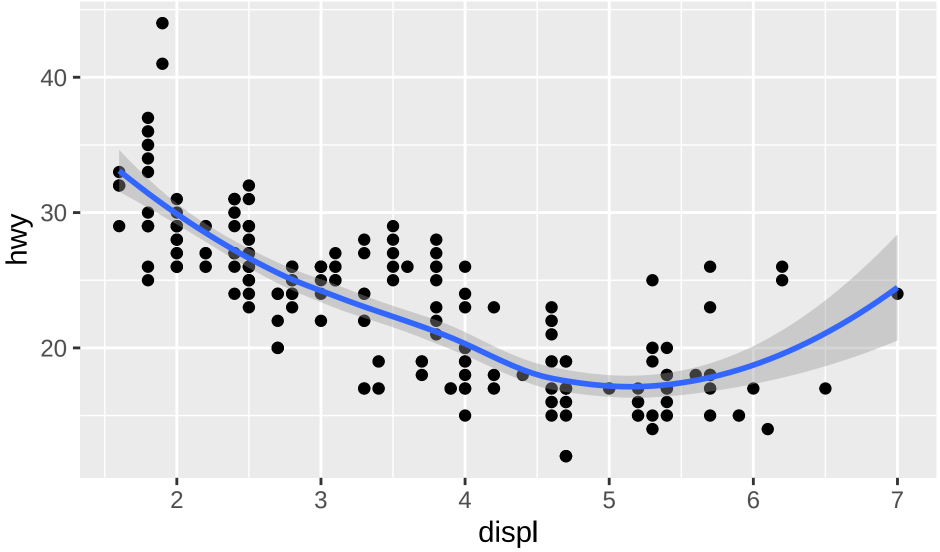
```
ggplot(mpg, aes(displ,hwy))+geom_point()+geom_line()
```



Adding a smoothered line

```
ggplot(mpg, aes(displ,hwy))+geom_point()+geom_smooth()
```

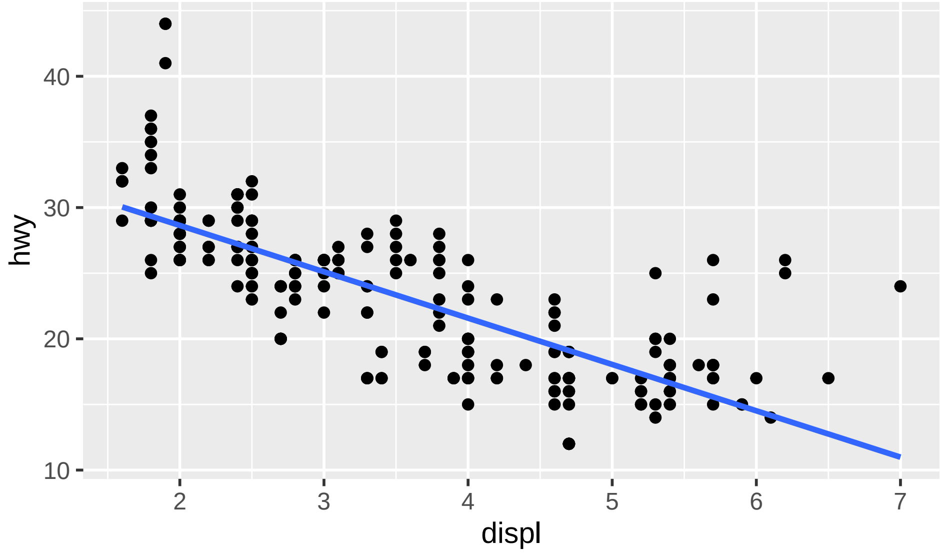
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Fitting linear model and adding the line of best fit

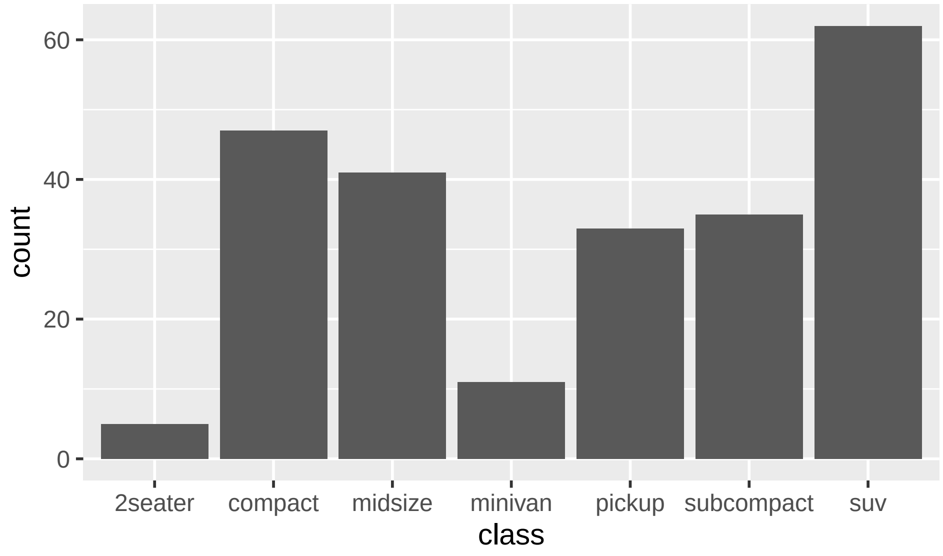
```
ggplot(mpg, aes(displ,hwy))+geom_point()+
  geom_smooth(method="lm",se=FALSE) # no confidence intervals
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

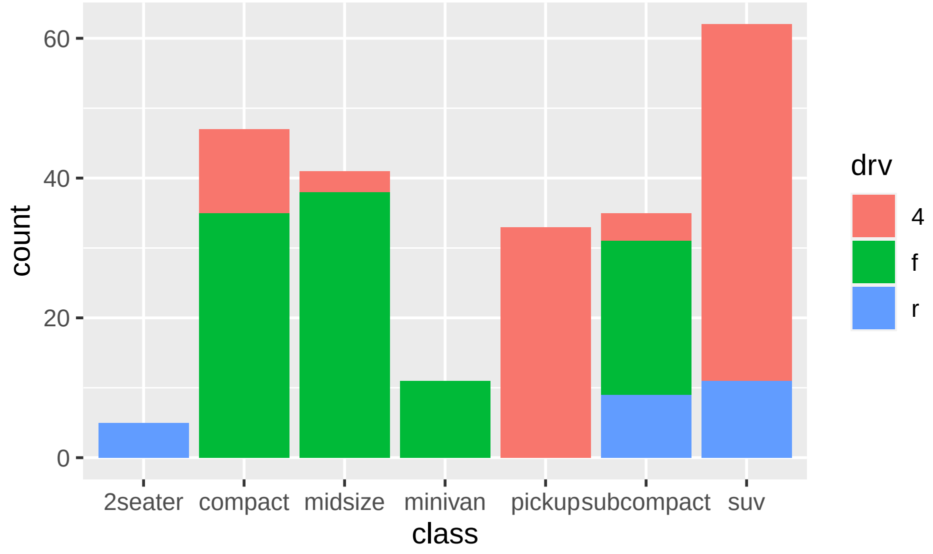


3.2.3. Bar Chart

```
ggplot(mpg, aes(class)) + geom_bar()
```

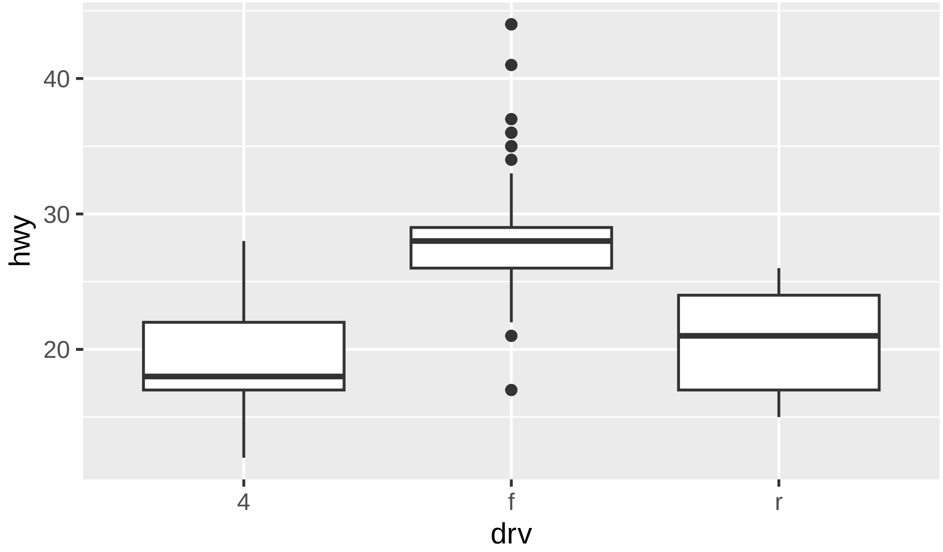


```
ggplot(mpg, aes(class,fill=drv)) + geom_bar()
```



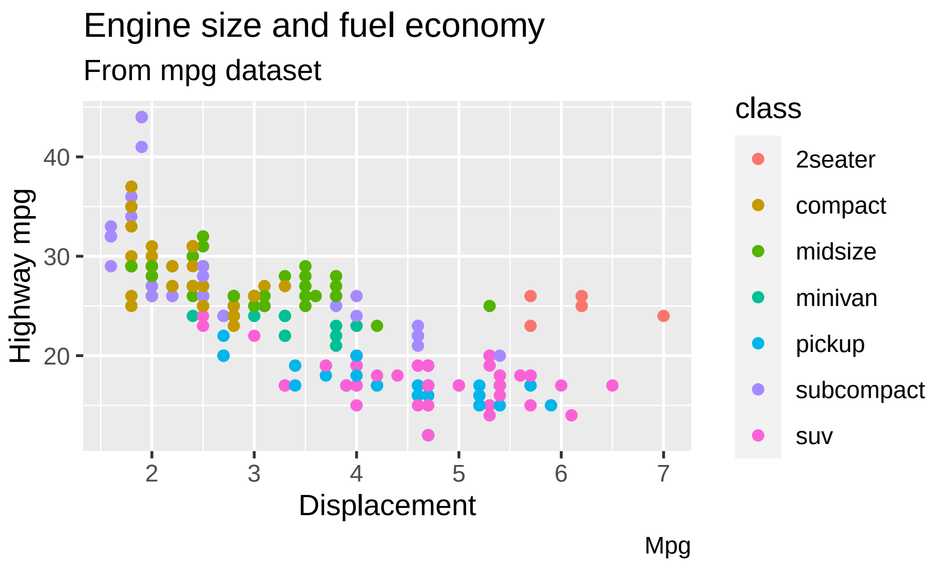
3.2.4. Boxplot

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

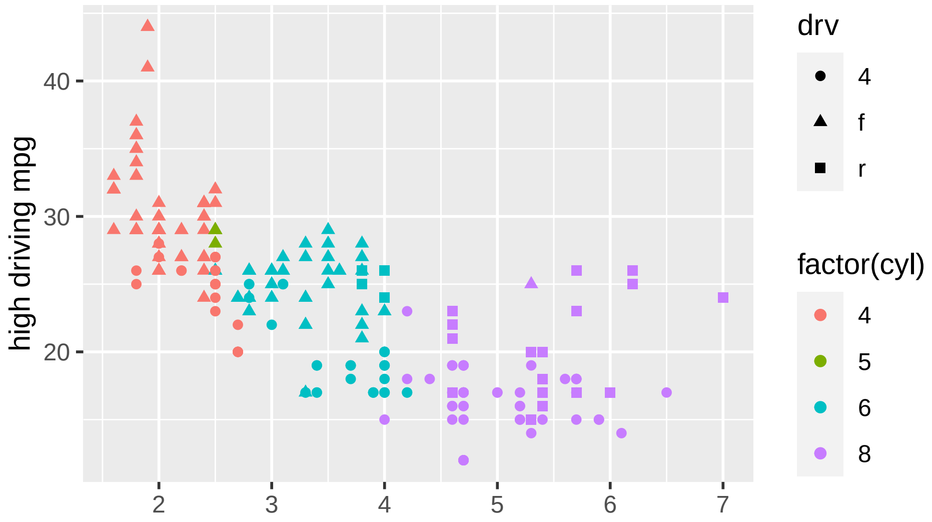


3.3. The Labels

```
ggplot(mpg, aes(displ,hwy, color=class)) +
  geom_point()+labs(title="Engine size and fuel economy", subtitle="From mpg dataset",
                    y="Highway mpg", x="Displacement", caption="Mpg")
```

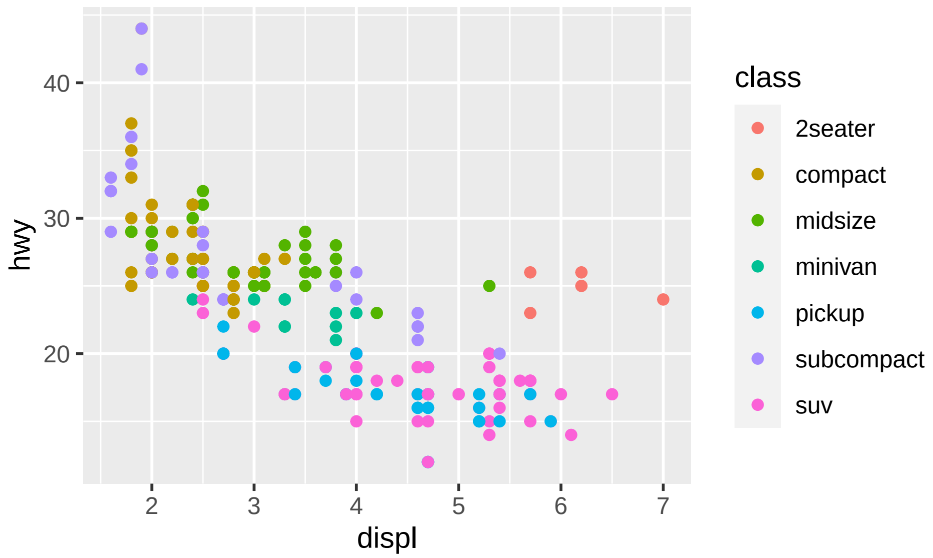


```
ggplot(mpg, aes(displ,hwy, color=factor(cyl))) +
  geom_point(aes(shape=drv))+ xlab ("") + ylab("high driving mpg")
```

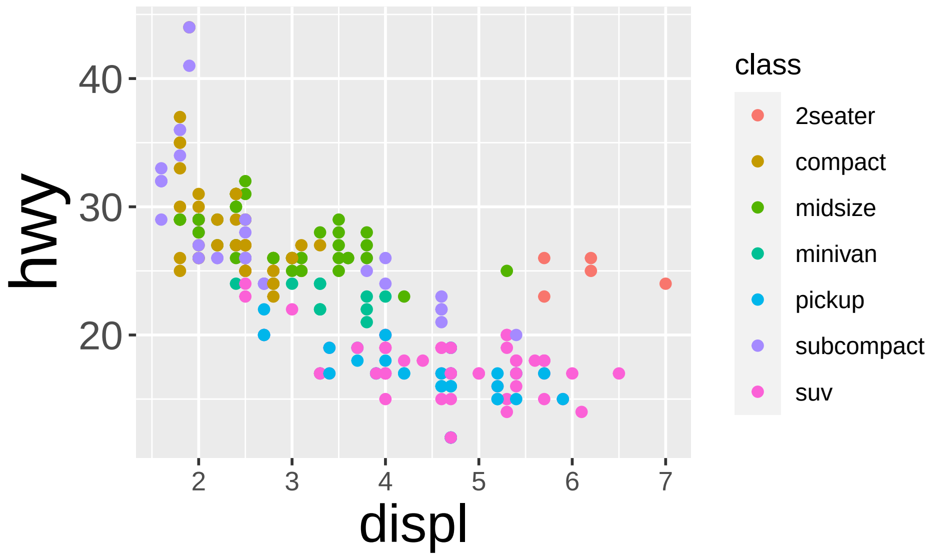


3.4. The Theme

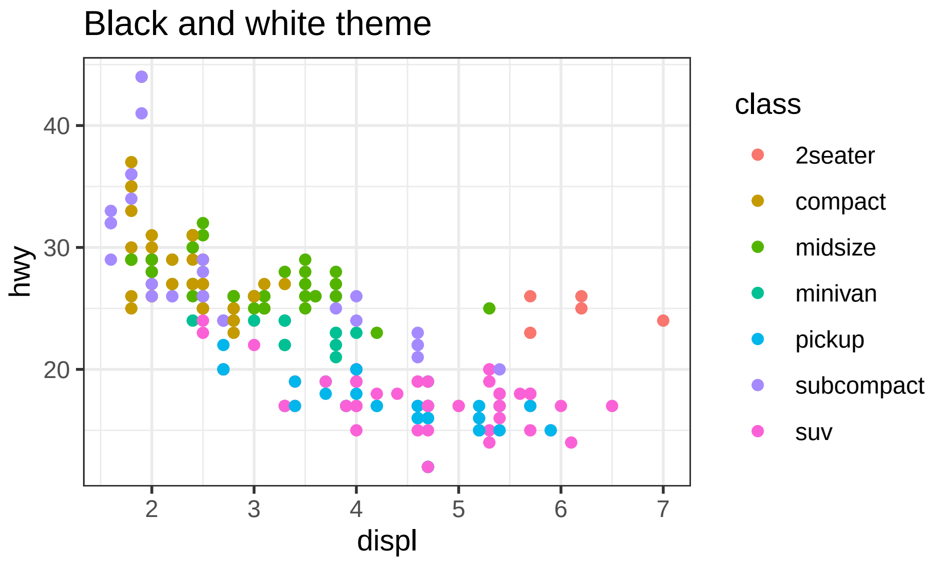
```
p=ggplot(mpg, aes(displ,hwy, color=class)) +
  geom_point()
p
```



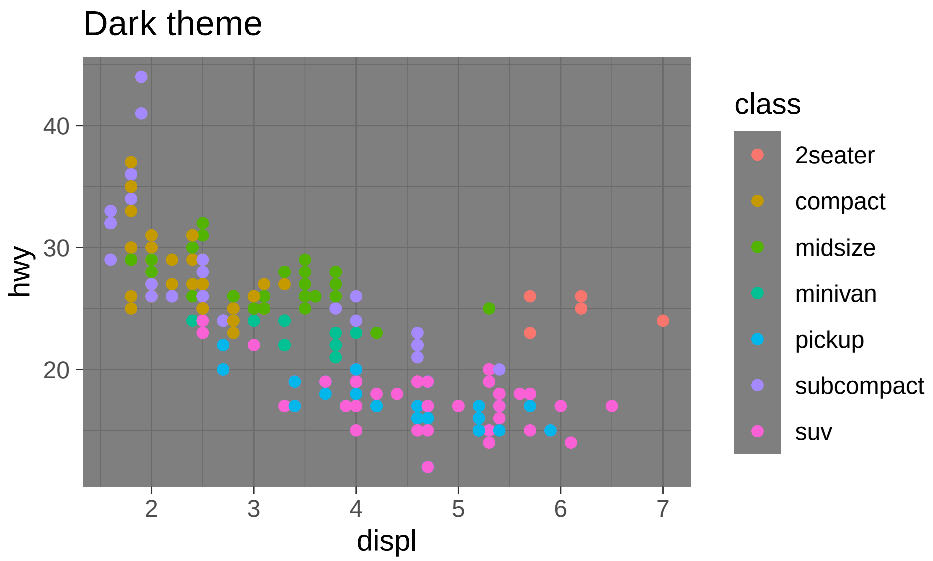
```
p+theme(plot.title=element_text(size=30, face="bold"),
                  axis.text.x=element_text(size=10),
                  axis.text.y=element_text(size=15),
                  axis.title.x=element_text(size=20),
                  axis.title.y=element_text(size=25))
```



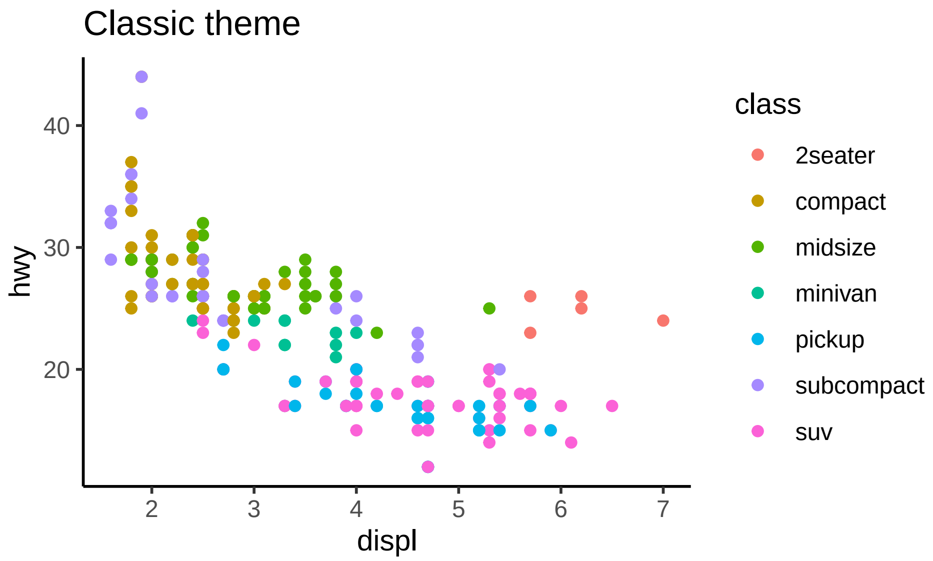
```
p+labs(title="Black and white theme")+theme_bw()
```



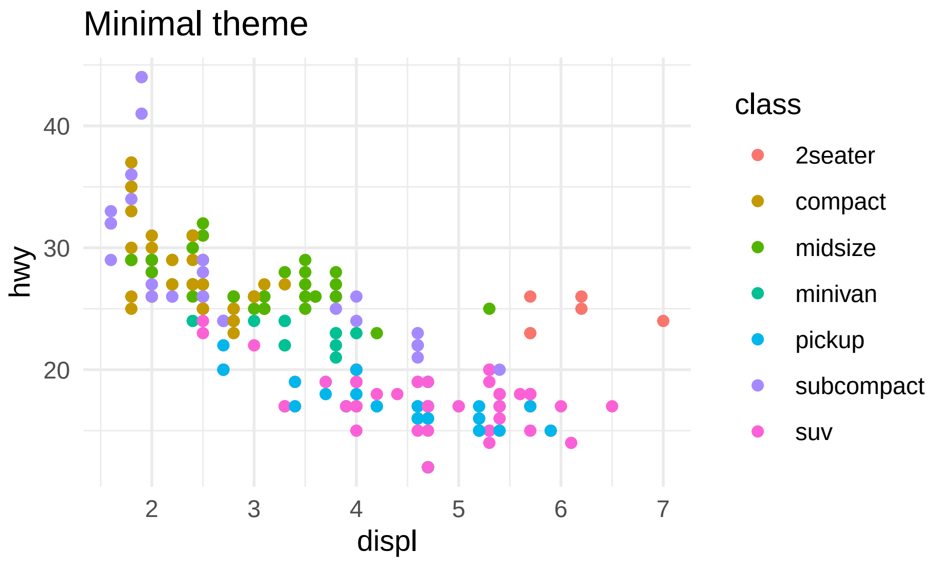
```
p+labs(title="Dark theme")+theme_dark()
```



```
p+labs(title="Classic theme")+theme_classic()
```



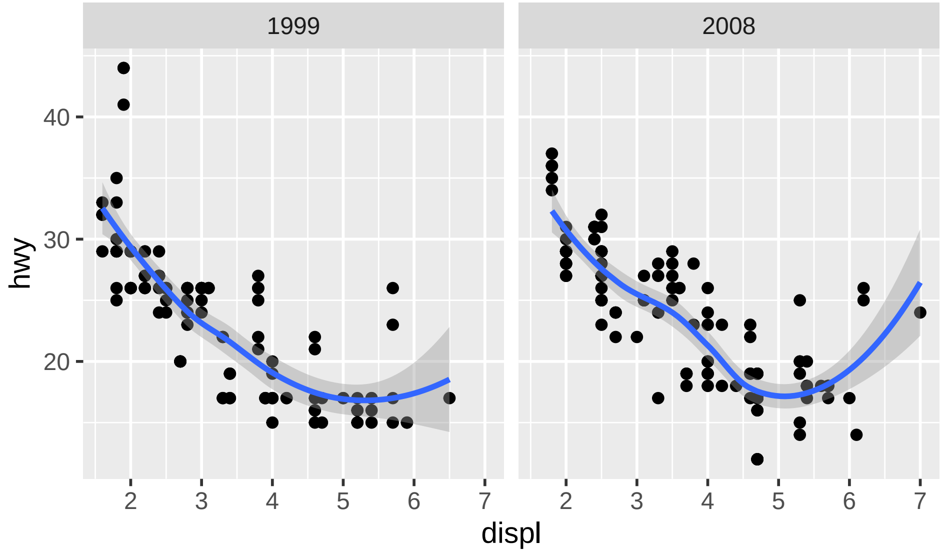
```
p+labs(title="Minimal theme")+theme_minimal()
```



3.5. The Facets

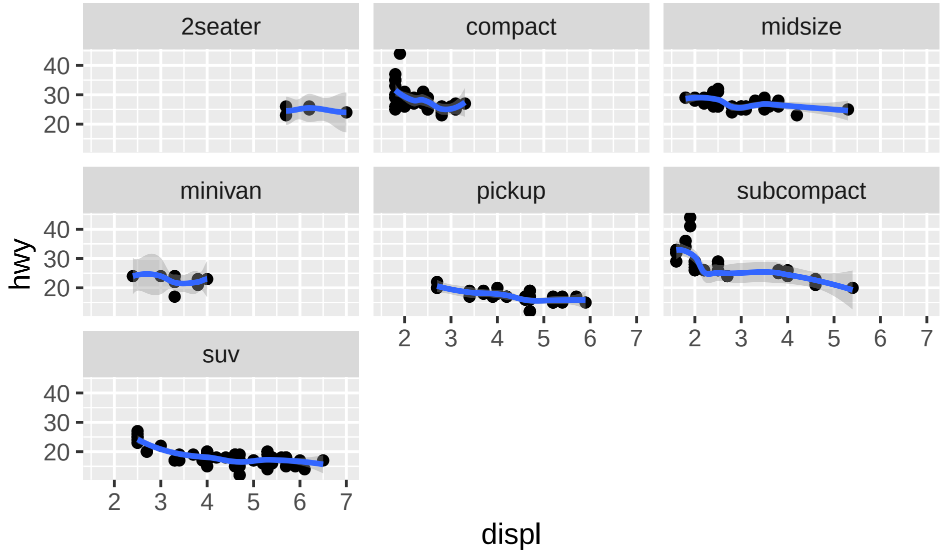
```
ggplot(data=mpg, aes(displ,hwy)) +
  geom_point()+
  geom_smooth()+
  facet_wrap(~year)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

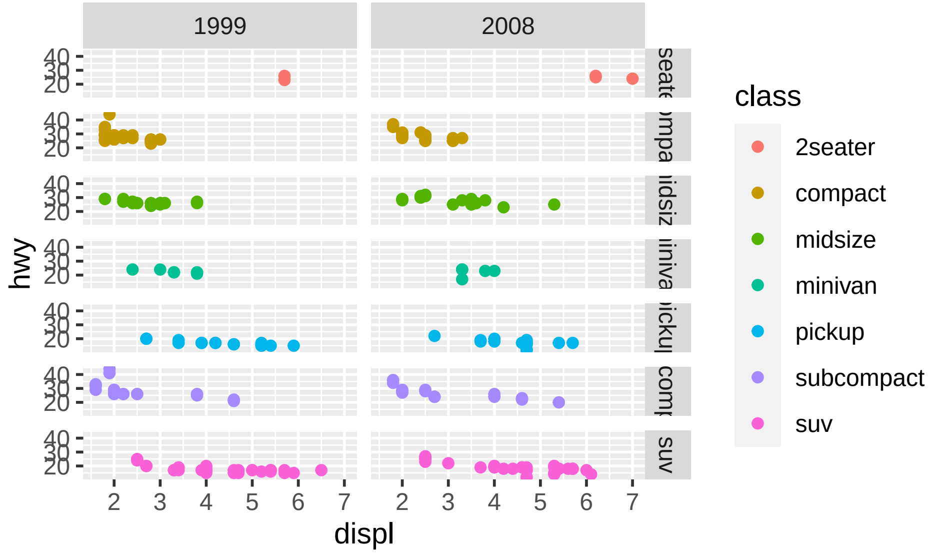


```
ggplot(data=mpg, aes(displ,hwy)) +
  geom_point()+
  geom_smooth()+
  facet_wrap(~class)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



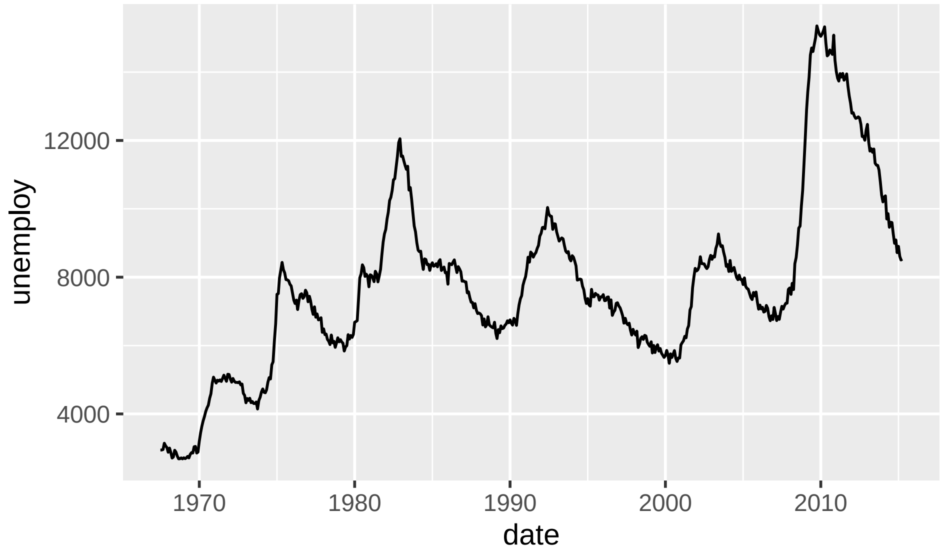
```
p+facet_grid(class~year)
```



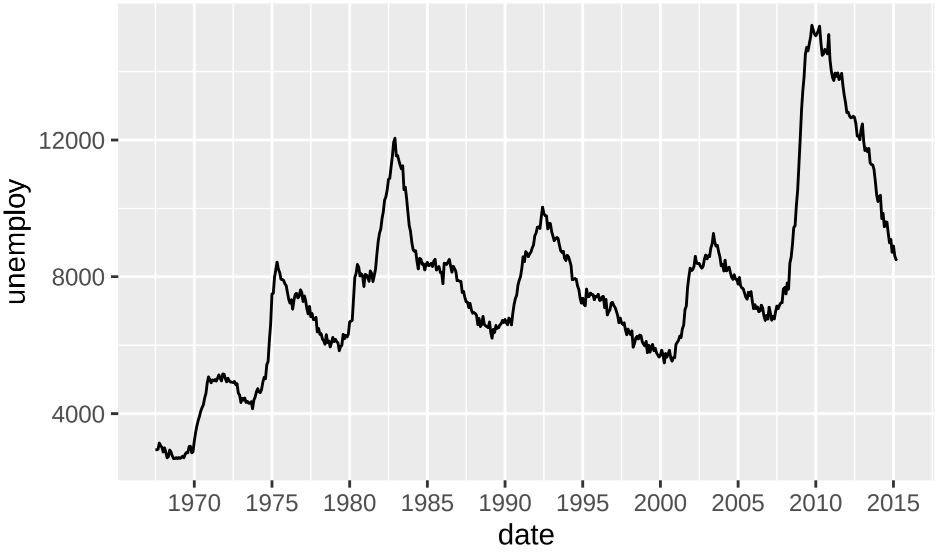
4. Time series plots

Economics data set contains economic data on the US measure over last 40 years. The figure below shows unemployment over time.

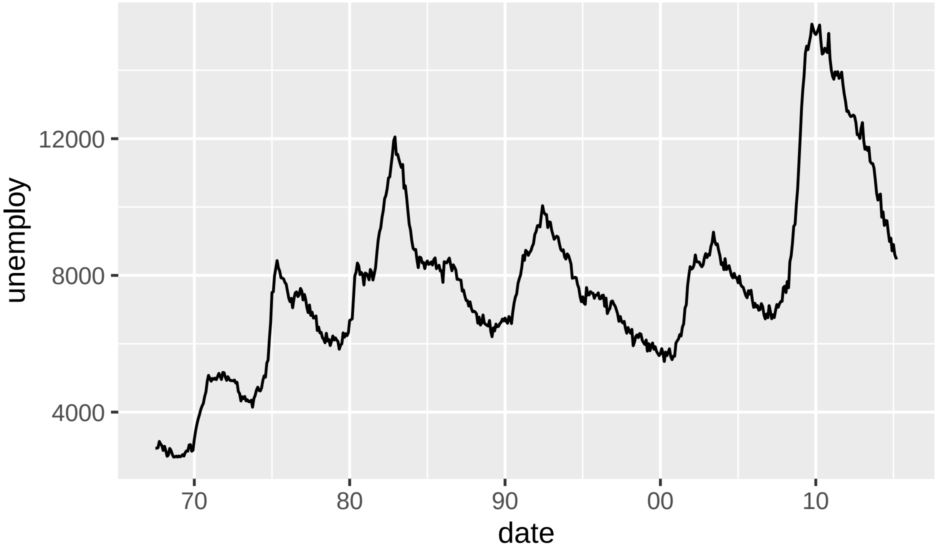
```
ts=ggplot(economics, aes (date,unemploy))+
  geom_line()
ts
```



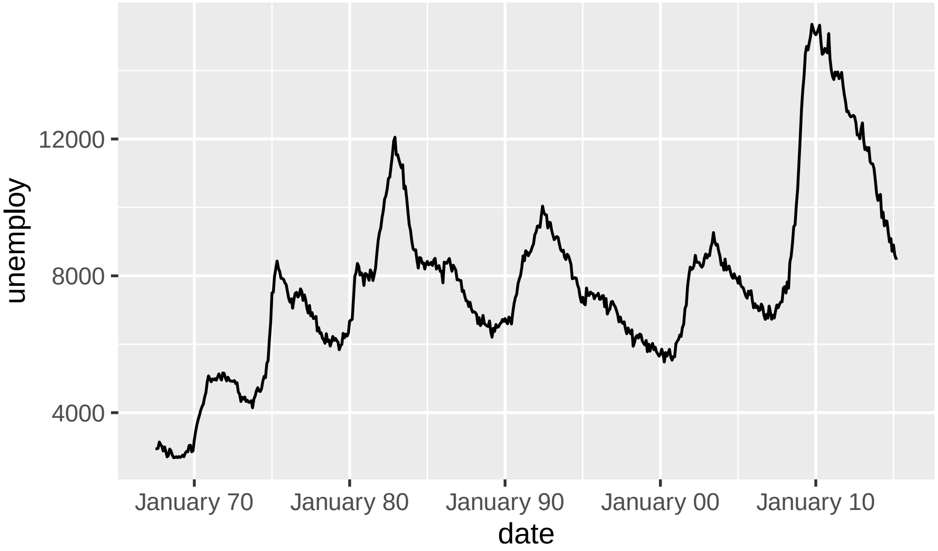
```
ts+scale_x_date(date_labels="%Y",date_breaks = "5 years")
```



```
ts+scale_x_date(date_labels="%y")
```



```
ts+scale_x_date(date_labels = "%B %y")
```



For more about date_labels, see: https://www.r-bloggers.com/customizing-time-and-date-scales-in-ggplot2

5. Playing with colors

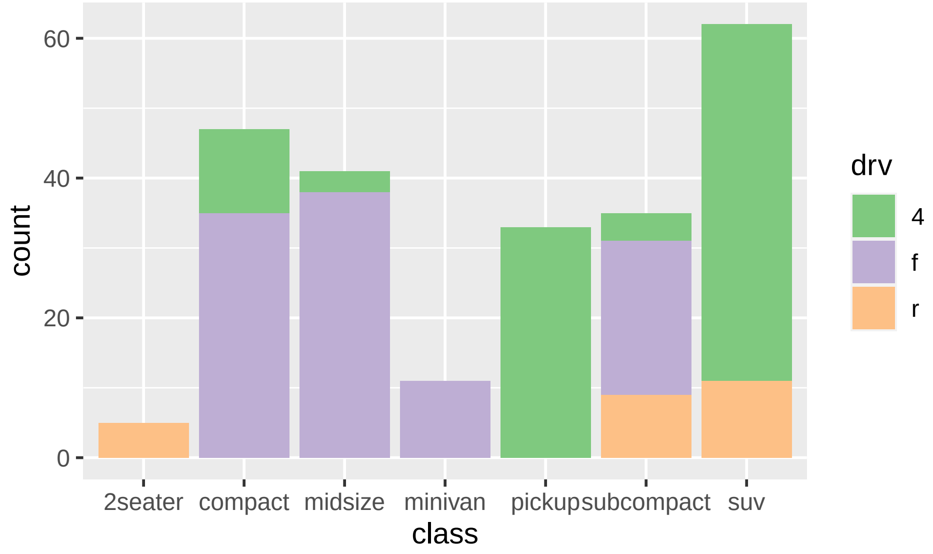
5.1. RColorBrewer palettes

The RColorBrewer package creates a nice looking color palettes. You should first install it as follow: install.packages("RColorBrewer").

To display all the color palettes in the package, type this:

```
library(RColorBrewer)
display.brewer.all()
display.brewer.all(colorblindFriendly = TRUE) # display only colorblind-friendly brewer palettes
```

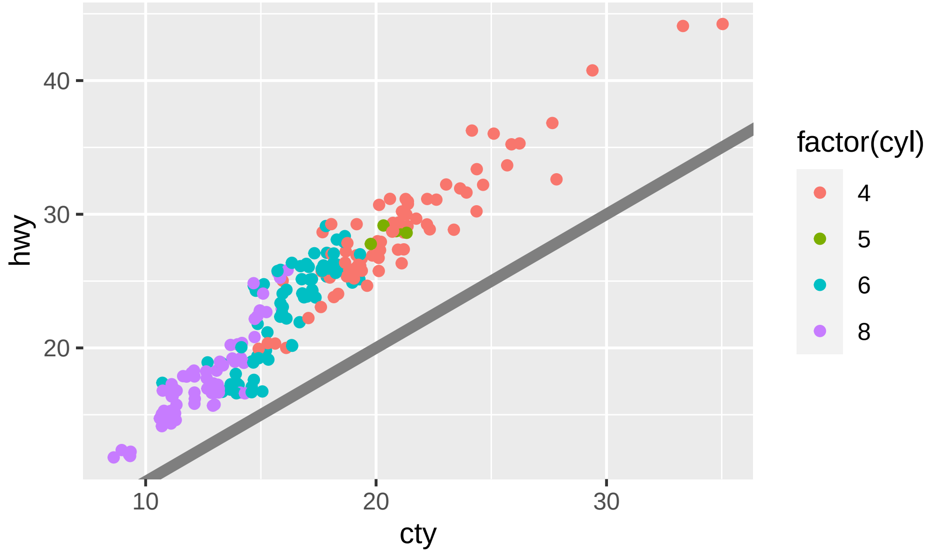
```
ggplot(mpg, aes(class,fill=drv)) + geom_bar()+scale_fill_brewer(palette="Accent")
```



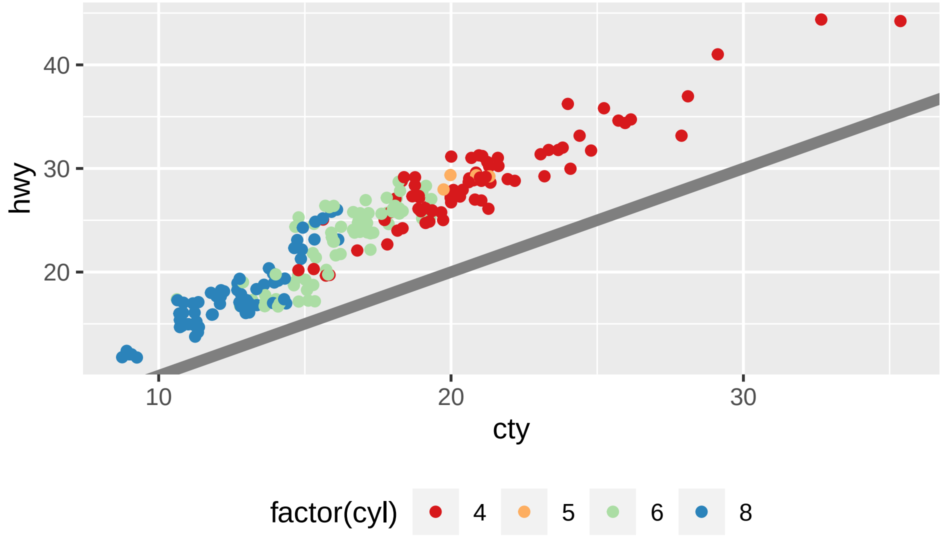
```
base=ggplot(mpg, aes (cty,hwy,color=factor(cyl)))+
  geom_jitter() +
  geom_abline(color="grey50",size=2)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
base
```



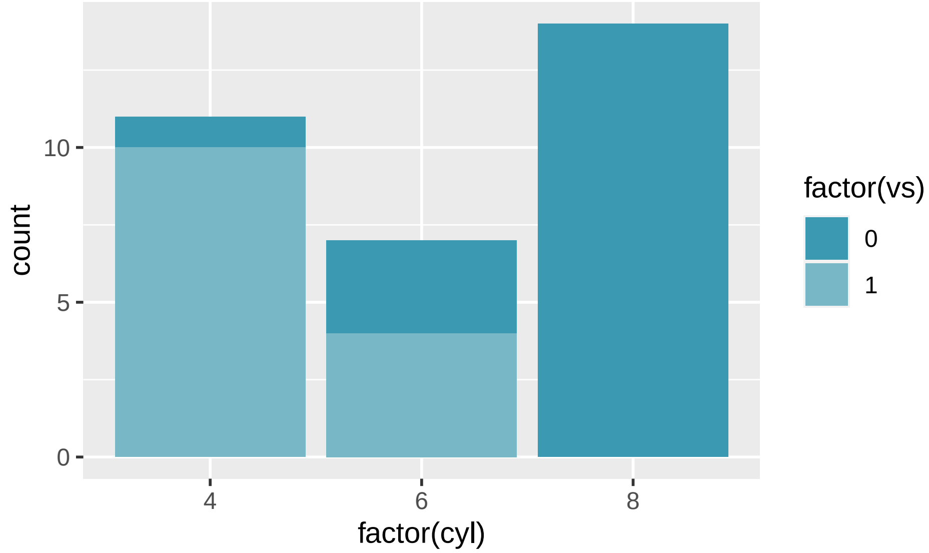
```
base+scale_color_brewer(palette="Spectral")+theme(legend.position = "bottom")
```



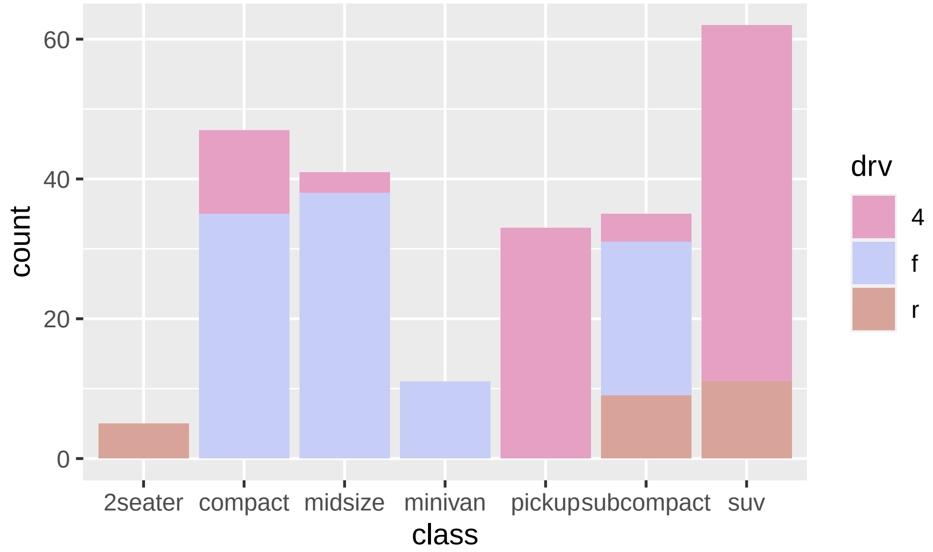
5.2. Wes Anderson movie color palettes

It contains 16 color palettes from Wes Anderson movies:

```
library("wesanderson")
ggplot(mtcars, aes(factor(cyl), fill=factor(vs))) +  geom_bar() +
  scale_fill_manual(values = wes_palette("Zissou1"))
```

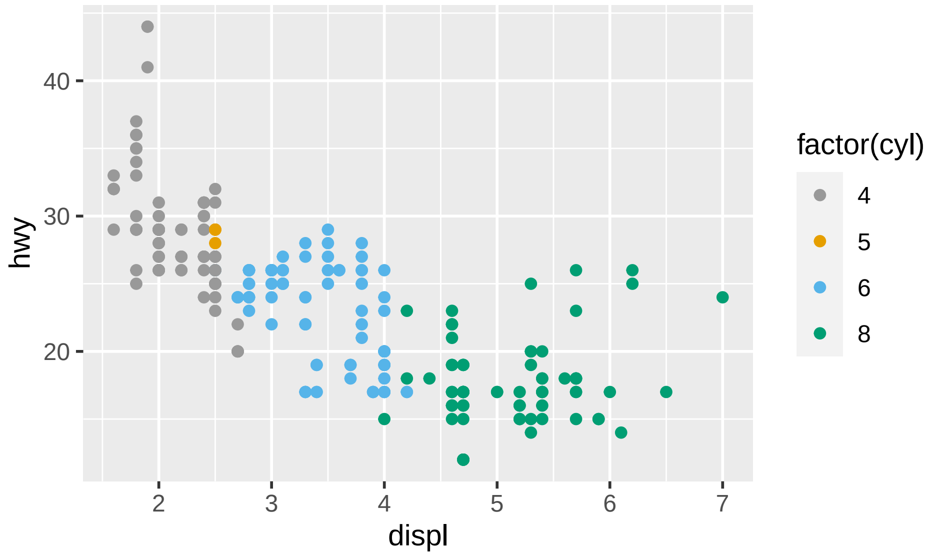


```
ggplot(mpg, aes(class,fill=drv)) + geom_bar()+
  scale_fill_manual(values = wes_palette("GrandBudapest2"))
```



5.3. Create your own color palette

```
cbp1 <- c("#999999", "#E69F00", "#56B4E9", "#009E73",
          "#F0E442", "#0072B2", "#D55E00", "#CC79A7") #colorblind-friendly palette
ggplot(data=mpg, aes(displ,hwy, color=factor(cyl))) +
  geom_point()+scale_color_manual(values = cbp1)
```



6. References

Wickham, H., & Grolemund, G. (2017). R for data science: Import, tidy, transform, visualize, and model data (First ed.). Beijing, [China]: O'Reilly.

Wickham, H.(2016). ggplot2 elegant graphics for data analysis (Second ed.). Houston, TX: Springer.

http://r-statistics.co/ggplot2-Tutorial-With-R.html

https://www.datanovia.com/en/blog/top-r-color-palettes-to-know-for-great-data-visualization

https://www.r-bloggers.com/customizing-time-and-date-scales-in-ggplot2/