ggplot

Anandu R

6/9/2020

Basic components of ggplot2 Plot

- A **data frame**
- **aesthetic mappings**: How data are mapped to color, size.
- **geoms**: Geometric objects like points, lines, shapes.
- **facets**: For conditional plots
- **stats**: statistical transformations like binning, quantiles, smoothing.
- **scales**: What scale an aesthetic map uses(example, male = red, female = blue)
- **coordinate system**

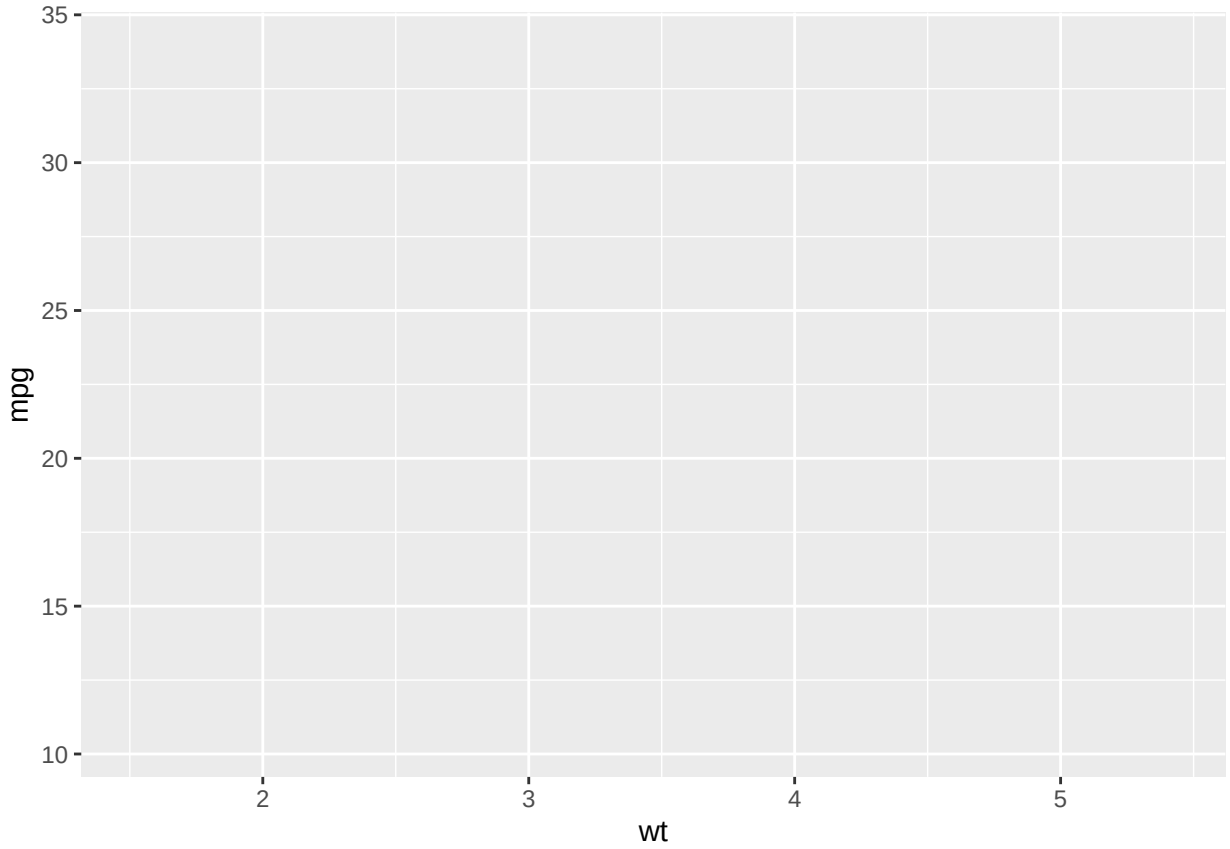
Making basic plot

Importing necessary libraries

```
invisible(library(ggplot2))
data = mtcars
```

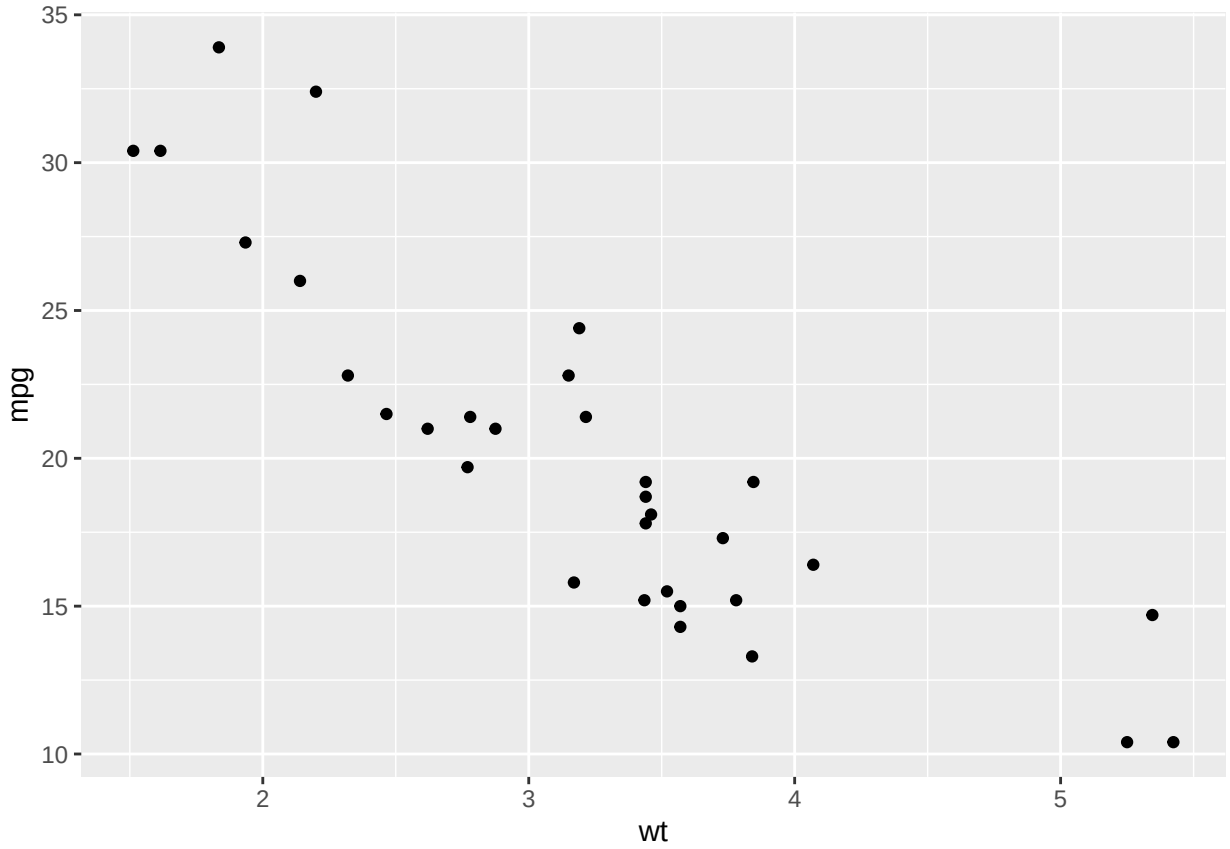
Creating plot object

```
plot = ggplot(
  data, aes(
    x = wt,
    y = mpg
  )
)
plot
```



The graph page has been made, but no data has been plotted on the graph yet, we need to specify the data

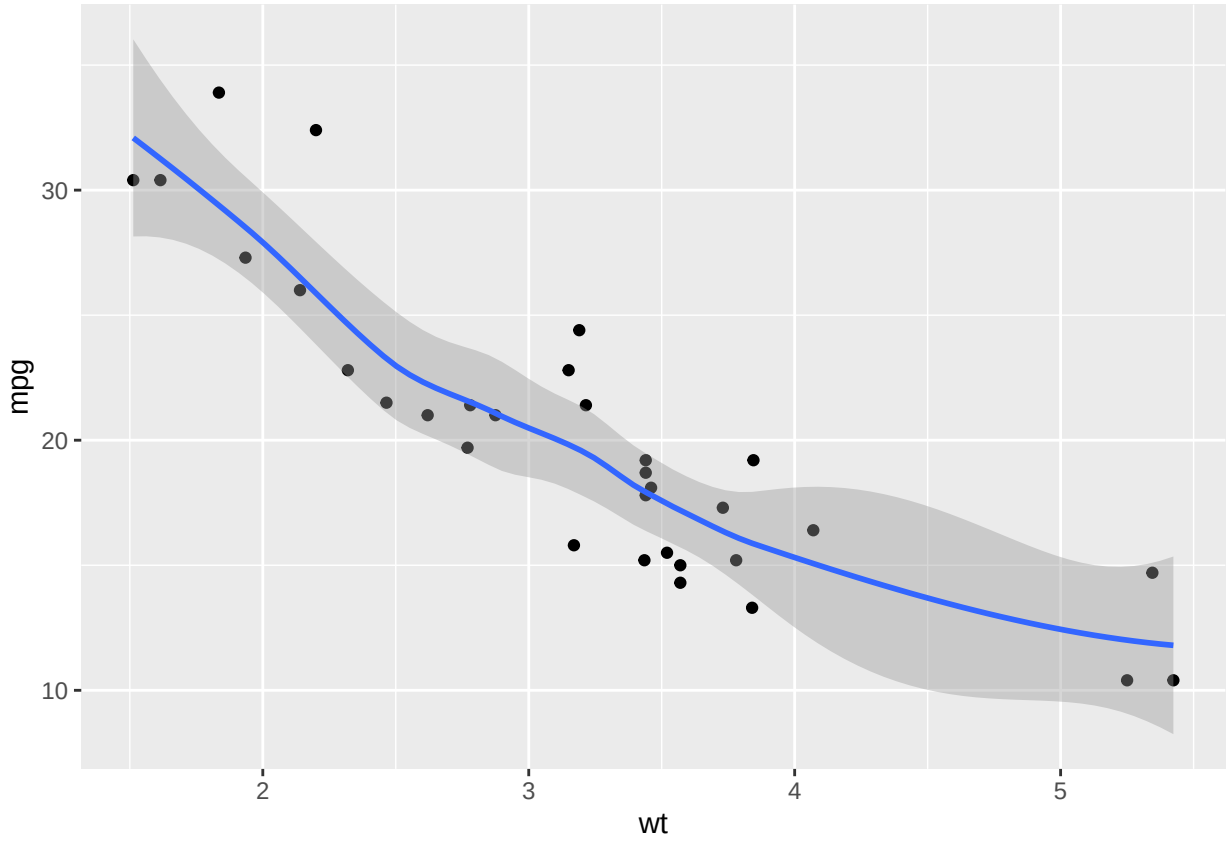
```
plot + geom_point()
```



Adding a smoother

```
plot + geom_point() + geom_smooth()
```

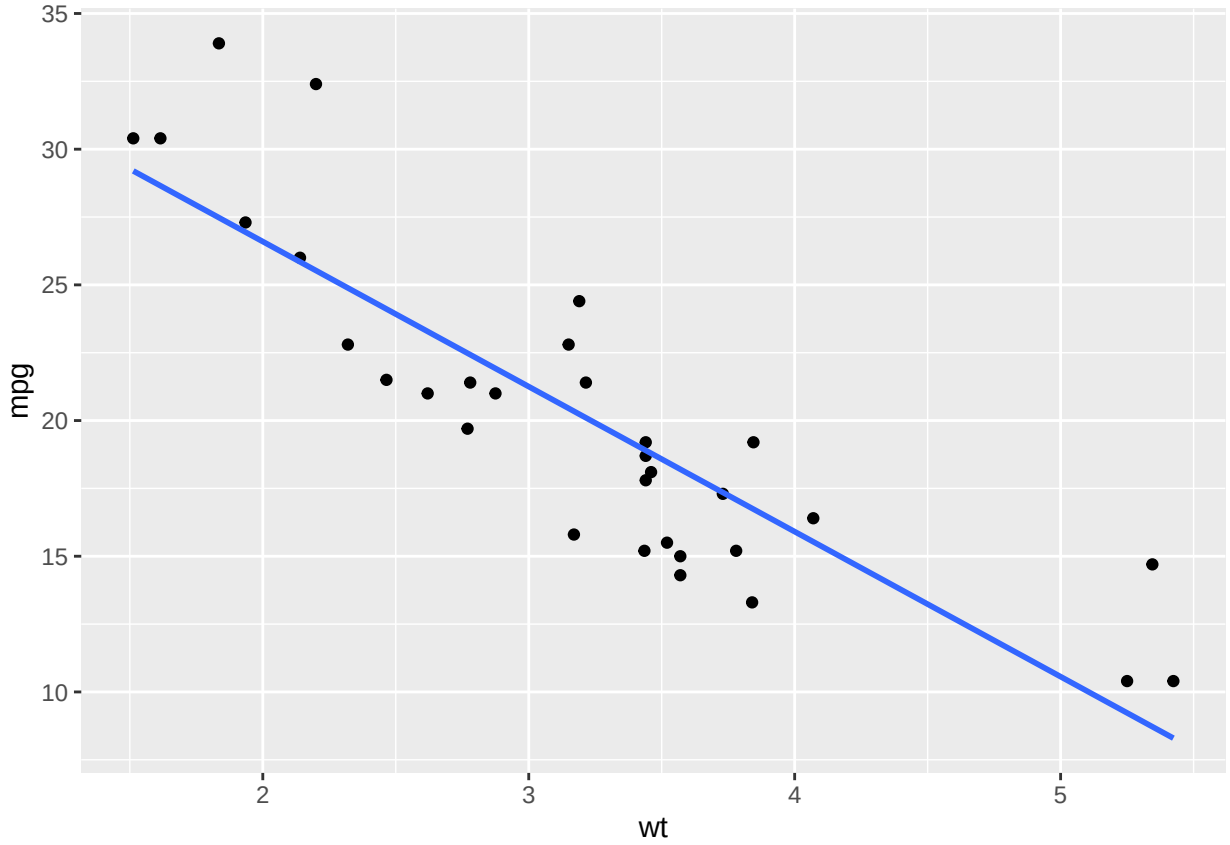
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The above smoother is by default method = "loess", using a linear model - lm smoother

```
plot + geom_point() + geom_smooth(method = "lm", se = F)
```

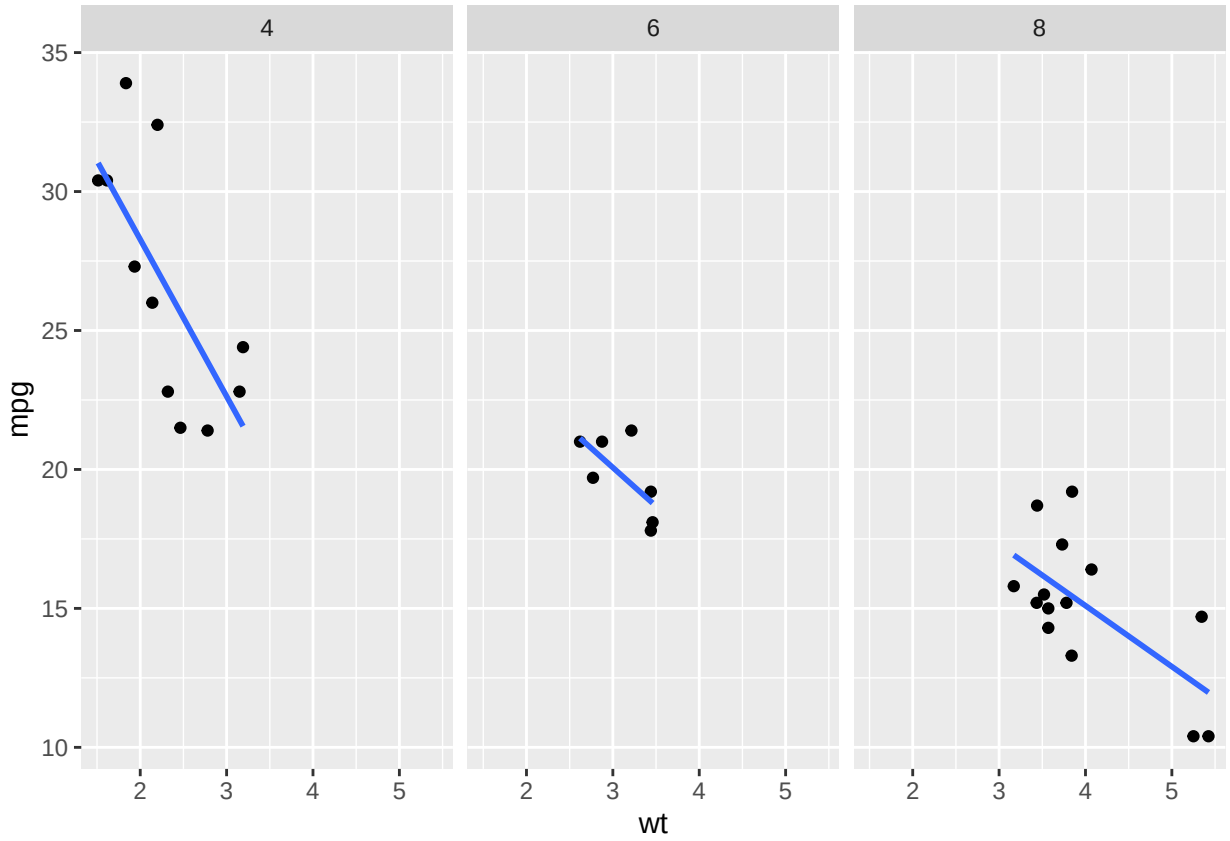
```
## `geom_smooth()` using formula 'y ~ x'
```



Adding facets

```
plot + geom_point() + facet_grid(.~cyl) + geom_smooth(method = "lm",se = F)
```

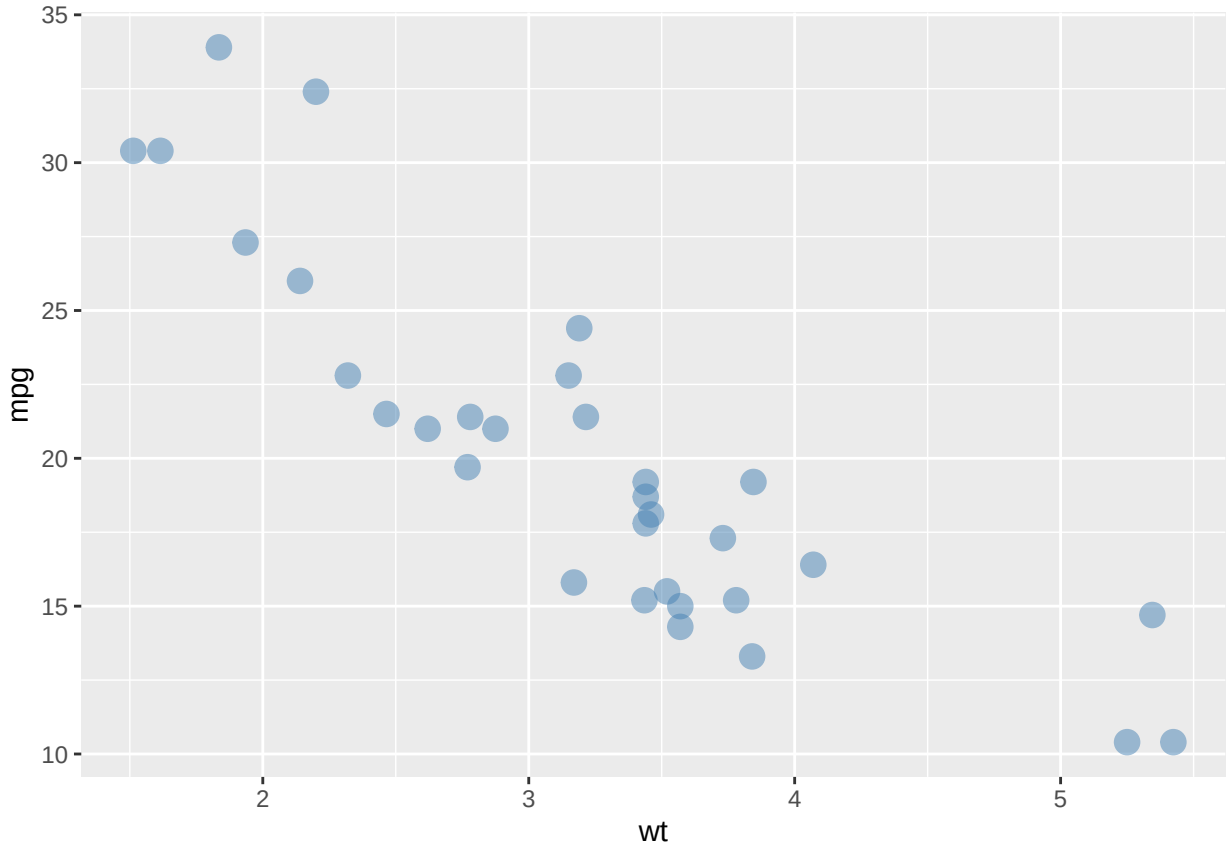
```
## `geom_smooth()` using formula 'y ~ x'
```



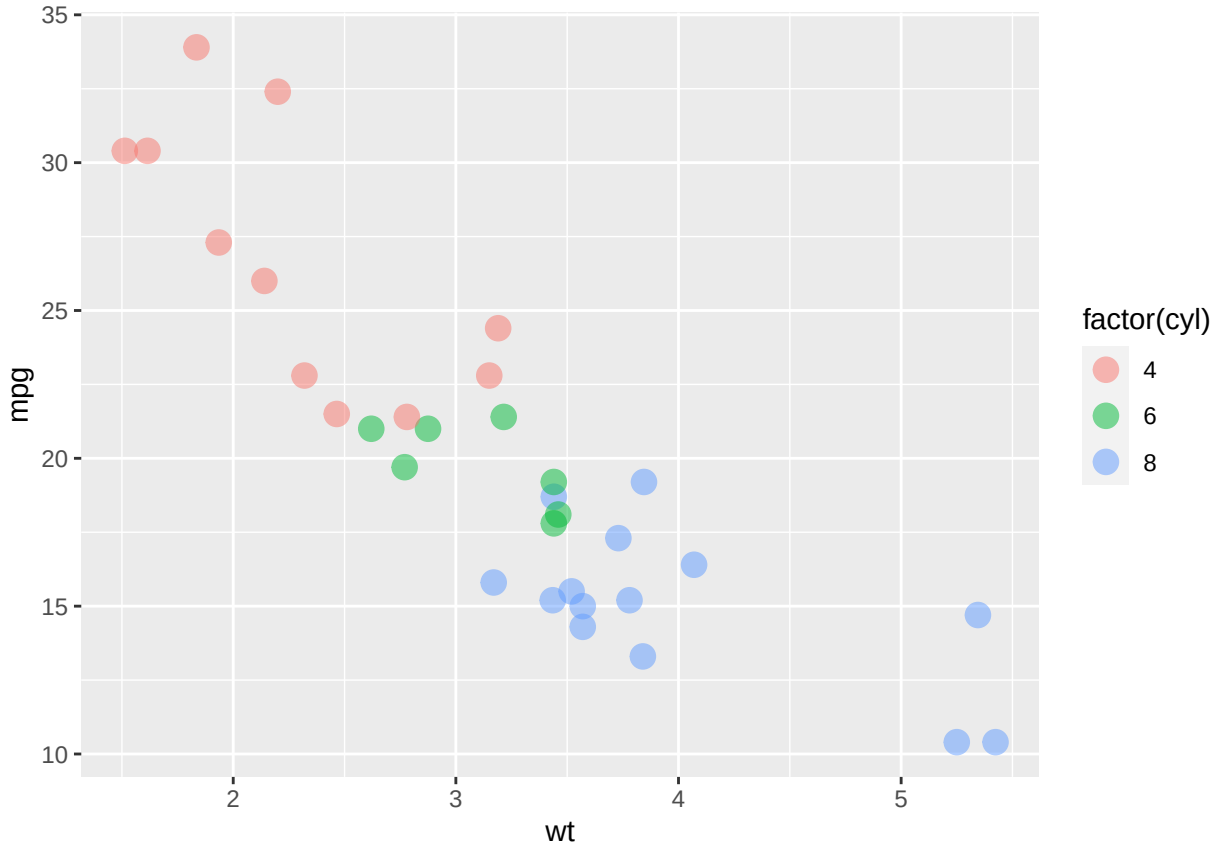
Annotations

- Labels: xlab(), ylab(), ggtitle()
- Two standard themes
  - theme_gray() - default
  - theme_bw()

```
plot + geom_point(size = 4,color = "steelblue", alpha = 0.5)
```

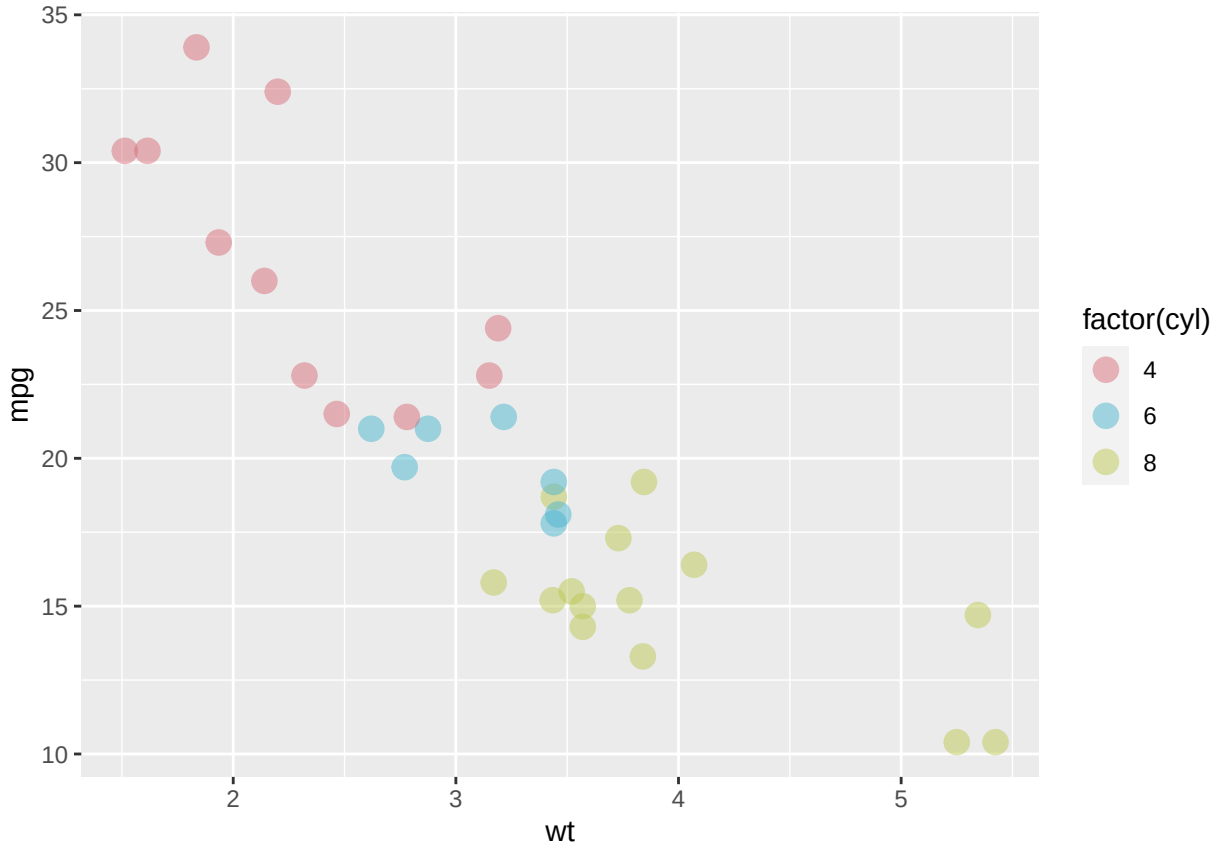



```
plot + geom_point(aes(color = factor(cyl)),size = 4, alpha = 0.5)
```



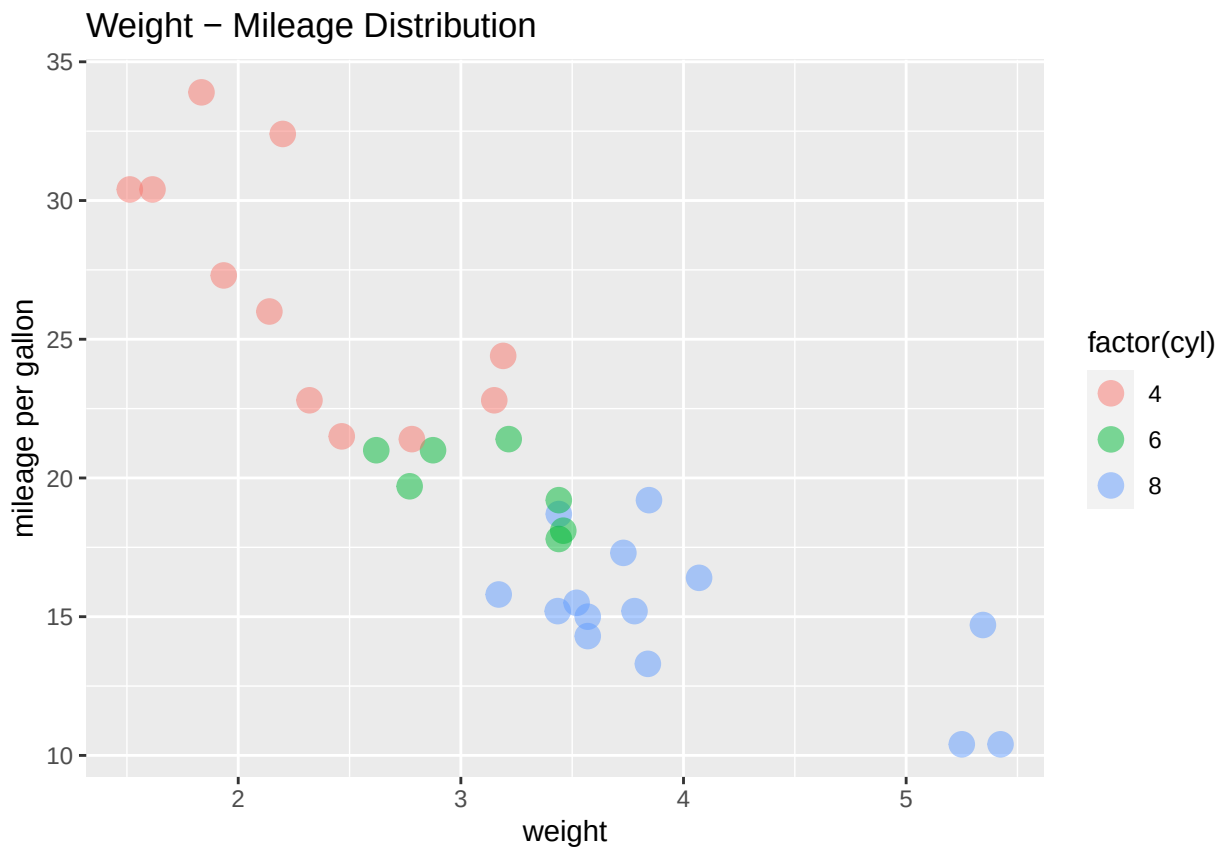
Custom colors

```
colors <- c("4" = "#D9717D", "6" = "#4DB6D0", "8" = "#BECA55")
plot + geom_point(aes(color = factor(cyl)),size = 4, alpha = 0.5) + scale_color_manual(values = colors)
```



Adding labels

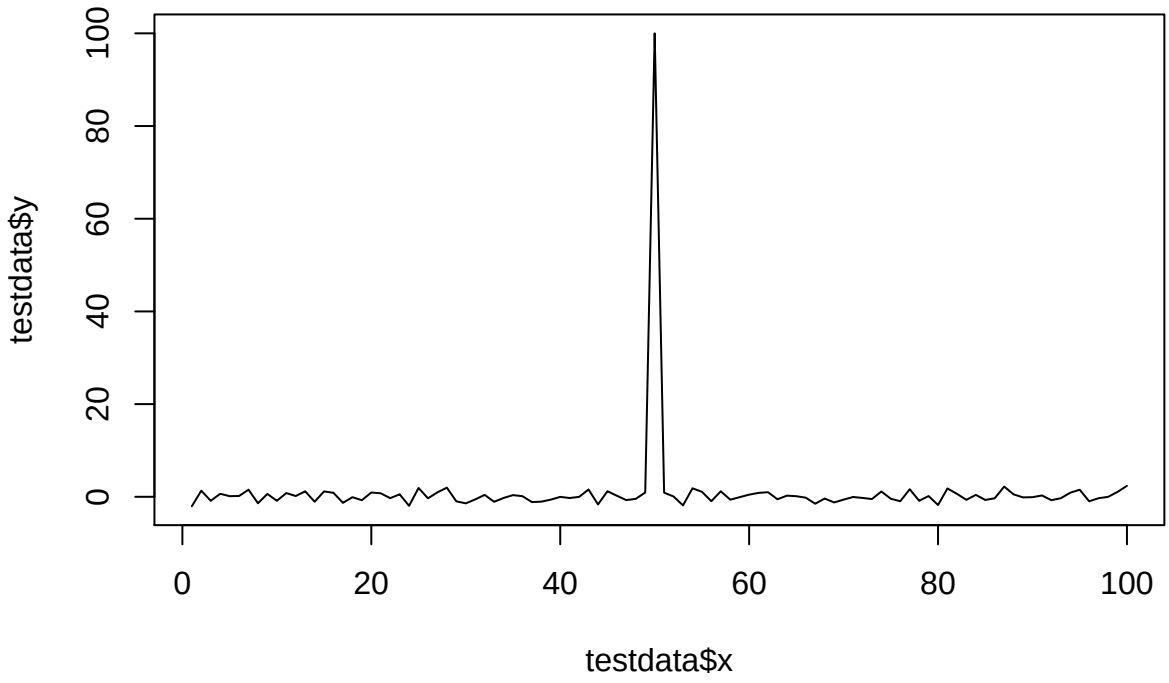
```
plot + geom_point(aes(color = factor(cyl)),size = 4, alpha = 0.5) + labs(title = "Weight - Mileage Distr
```



Dealing with scaling and cartesian coordinates

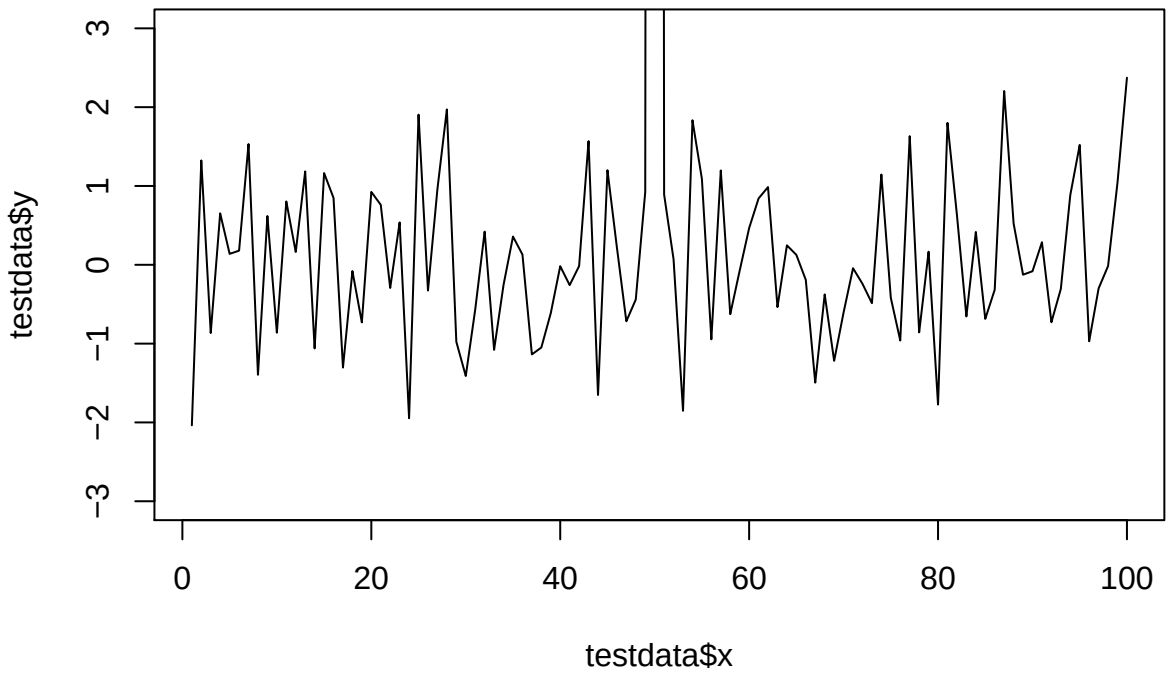
To create a line plot to analyse outlier within data

```
testdata = data.frame(x = 1:100, y = rnorm(100))
testdata[50,2] = 100
plot(testdata$x,testdata$y, type = "l")
```



To only look at the regular data using the base plot function

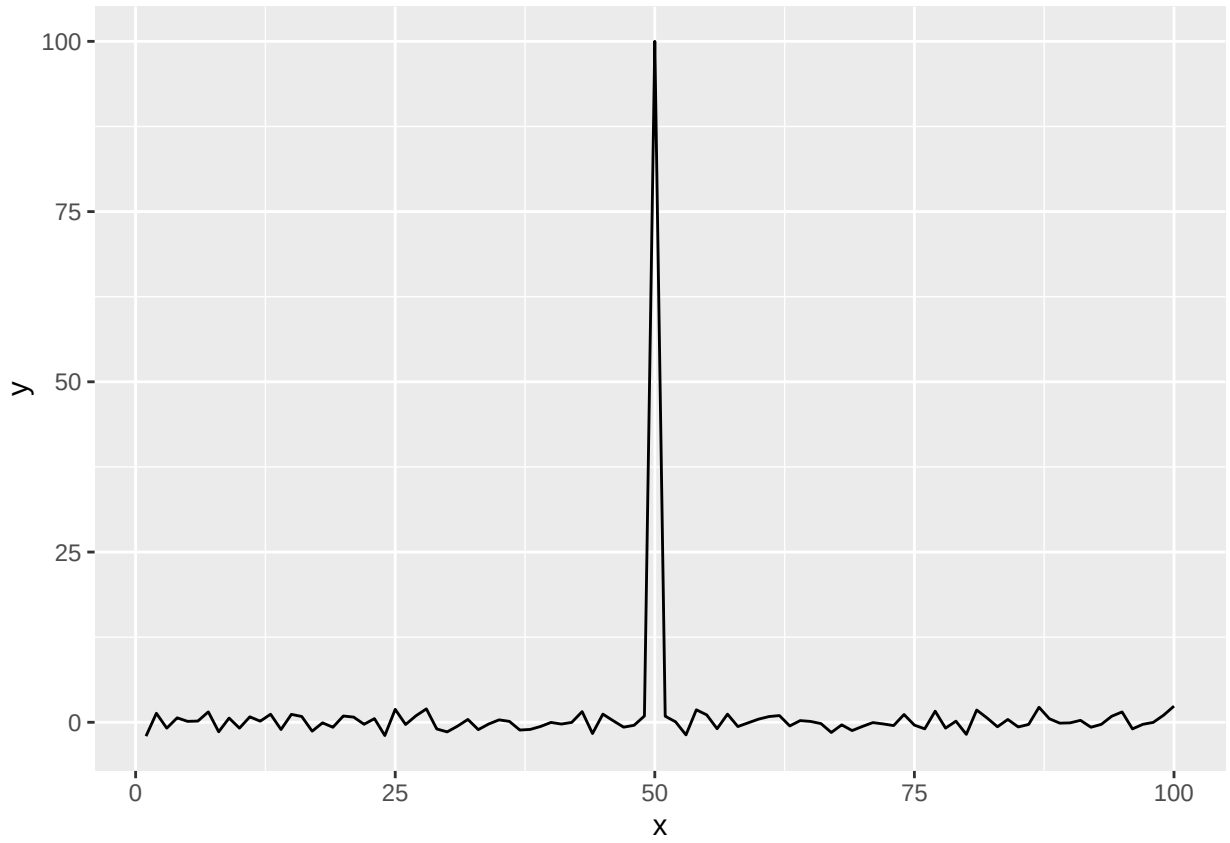
```
plot(testdata$x,testdata$y, type = "l", ylim = c(-3,3))
```



In case of ggplot adding the ylim() function to the plot first subsets the data to only include points within limit and then plot the data

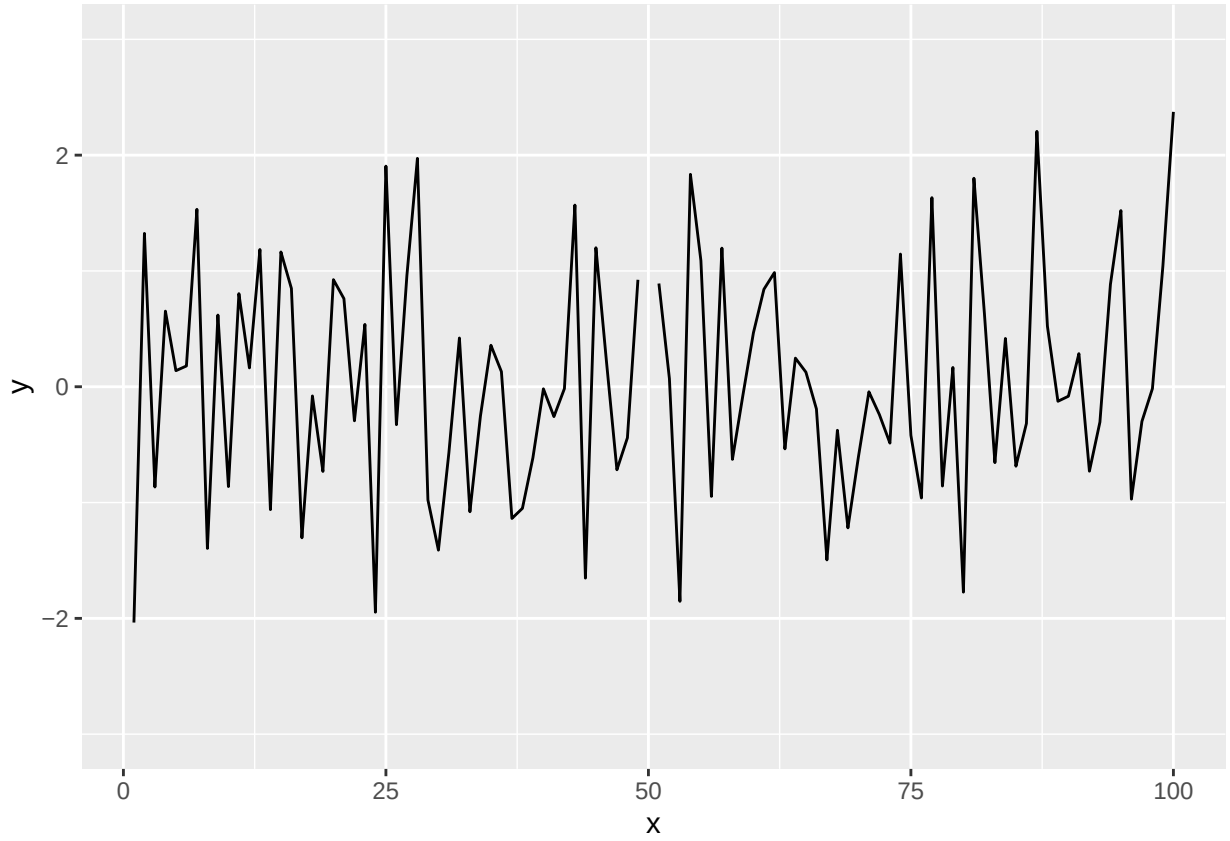
- The original data:

```
testplot = ggplot(
  testdata,
  aes(x = x,
      y = y)
)
testplot + geom_line()
```



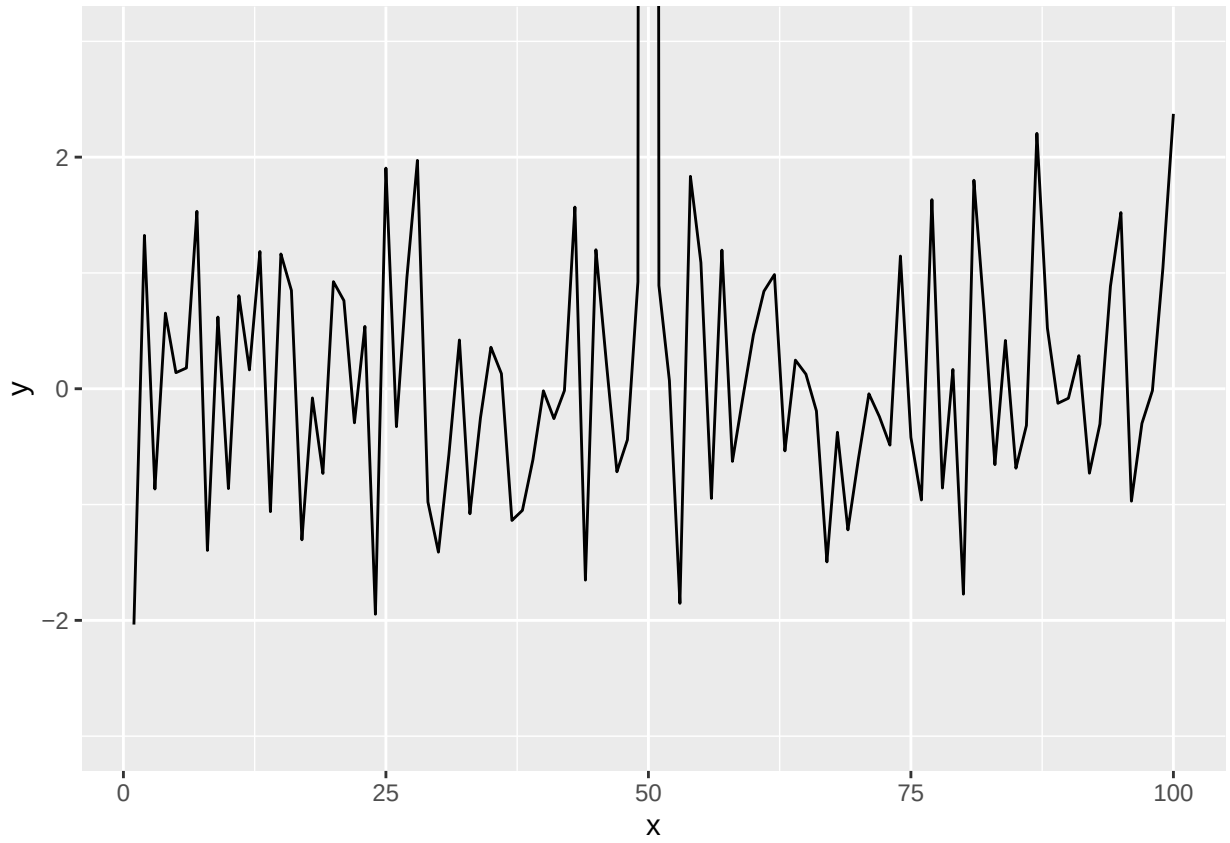
- adding ylim()

```
testplot + geom_line() + ylim(-3,3)
```



We can clearly see the data at x = 50 is missing, to solve this we add coord_cartesian() instead of ylim() directly

```
testplot + geom_line() + coord_cartesian(ylim = c(-3,3))
```



Dealing with continous variables during plot

In mnay cases we'd be required to plot by conditioning continous variables, there are several ways to go about this, one method is to categorise the varaible into reasonable series of ranges, we use the cut() function for this purpose

```
cutpoints = quantile(data$hp, seq(0,1,length = 4), na.rm = T)
data$hp_cutlevels = cut(data$hp, cutpoints)
levels(data$hp_cutlevels)
```

```
## [1] "(52,109]"  "(109,175]" "(175,335]"
```

Cleaning up the variable

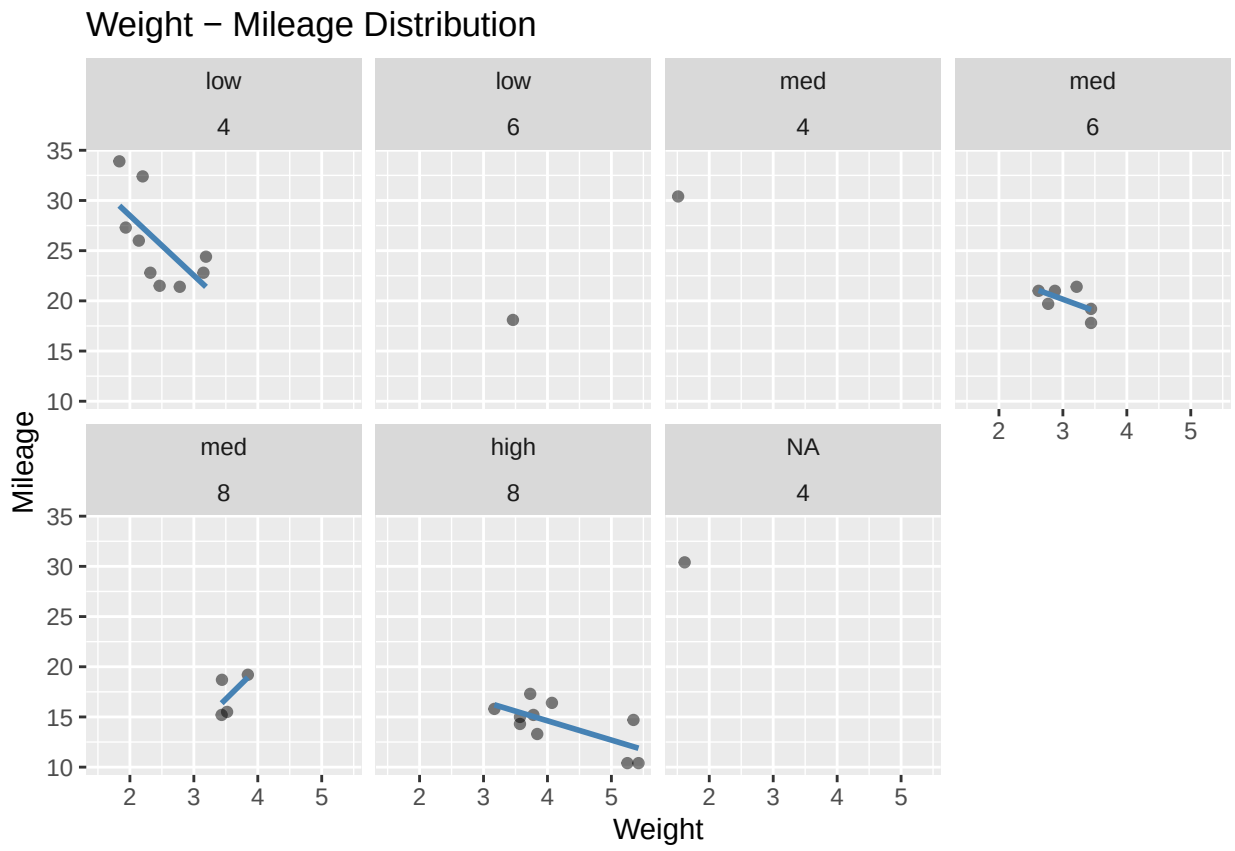
```
lev_hp = levels(data$hp_cutlevels)
data$hp_cutlevels = factor(data$hp_cutlevels, levels = lev_hp, labels = c("low","med","high"))
levels(data$hp_cutlevels)
```

```
## [1] "low"  "med"  "high"
```

Plotting the data

```
plot + geom_point(
  alpha = 0.5
) + facet_wrap(
  data$hp_cutlevels~data$cyl,
  nrow = 3,
  ncol = 4
) + geom_smooth(
  method = "lm",
  se = F,
  col = "steelblue"
) + labs(
  title = "Weight - Mileage Distribution",
  x = "Weight",
  y = "Mileage"
)
```

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



Since there is so few data, not all facets were populated