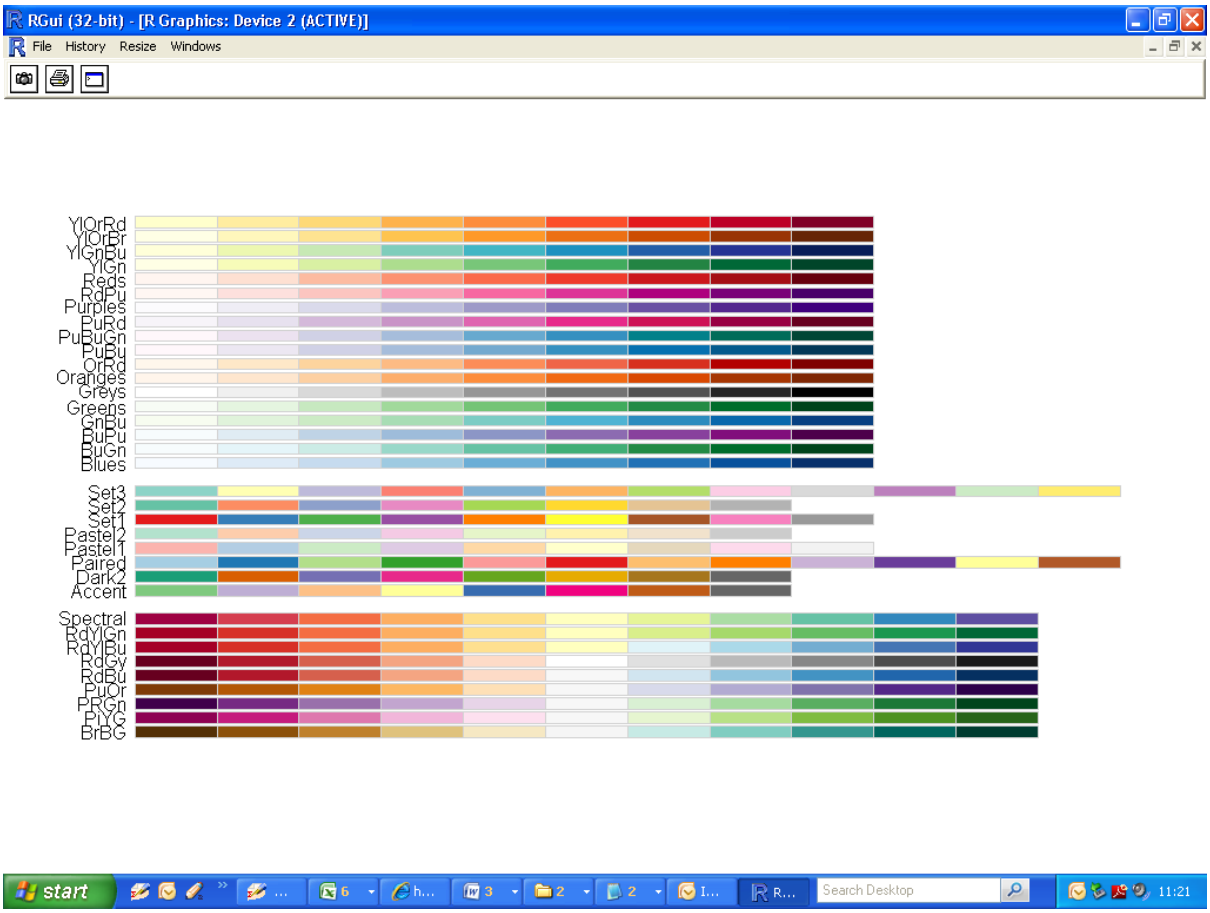
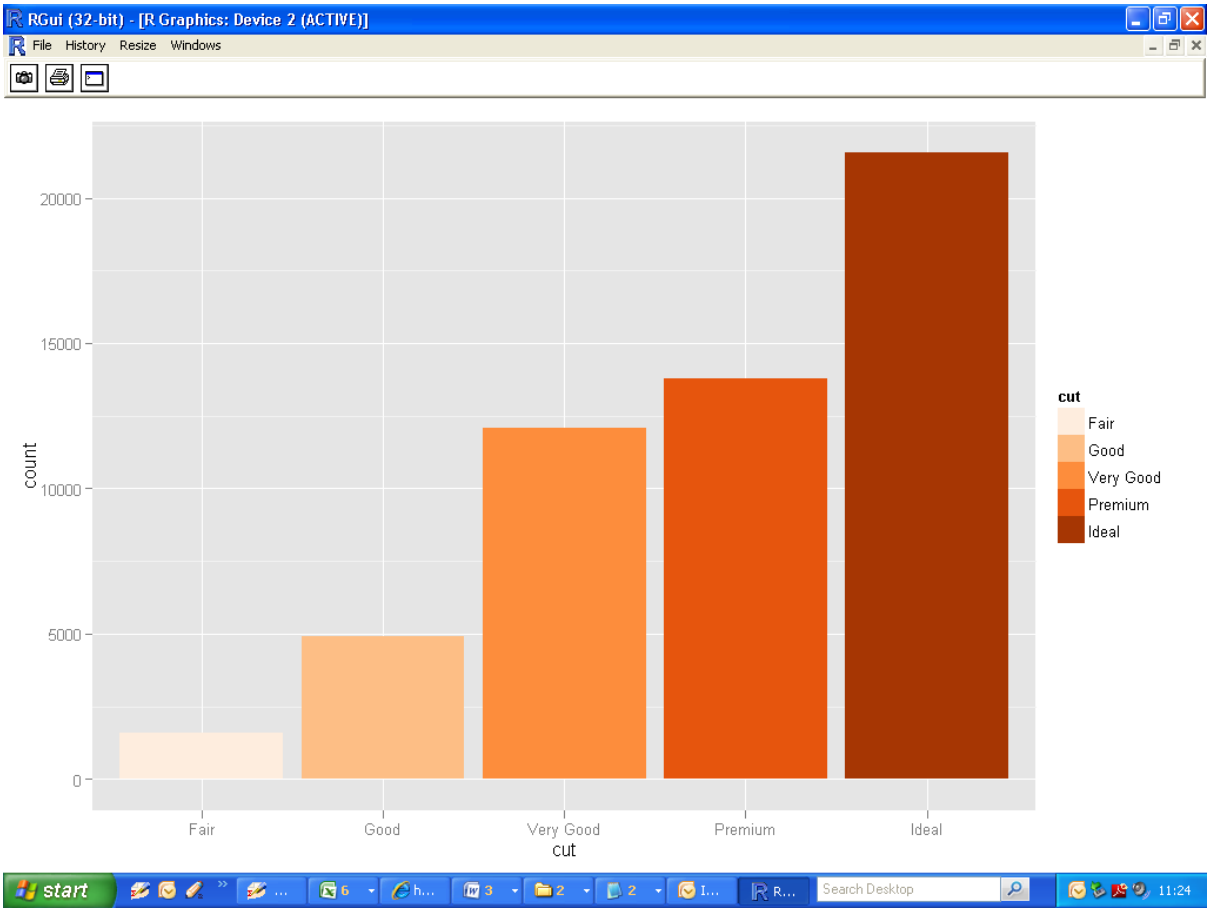
Palettes

```
>RColorBrewer::display.brewer.all()
```



```
>QPlot + scale_fill_brewer(palette="Oranges")
```



Themes

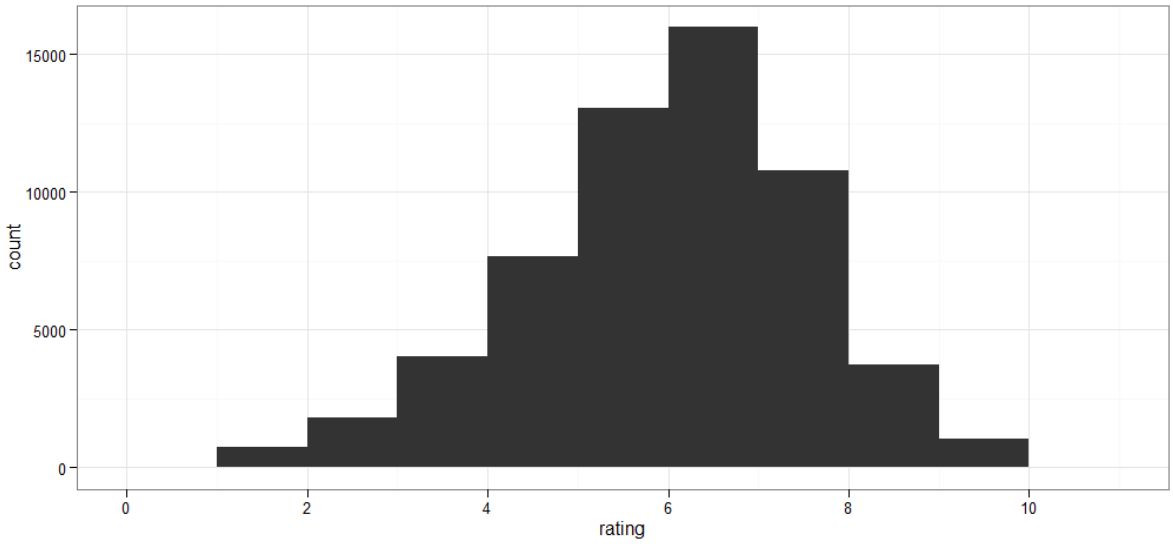
```
> hgram <- qplot(rating, data = movies, binwidth = 1)
> hgram
```

Notice the default grey theme with grey background and white gridlines.

The alternative to the default them is the black and white theme (`theme_bw()`)with white background and grey gridlines.

Change the theme to this theme and draw the plot again

```
> theme_set(theme_bw())
> hgram
>
> # to switch back
> #  theme_set(theme_grey())
> # hgram
```



Each theme is made up of multiple elements. The theme system comes with a number of built-in element rendering functions with a limited set of parameters. By adjusting these parameters you can control things like text size and colour, background and grid line colours and text orientation.

By combining multiple elements you can create your own theme.

You can see the settings for the current theme with **`theme_get()`.** There is quite a lot out output, and only a small portion is included below.

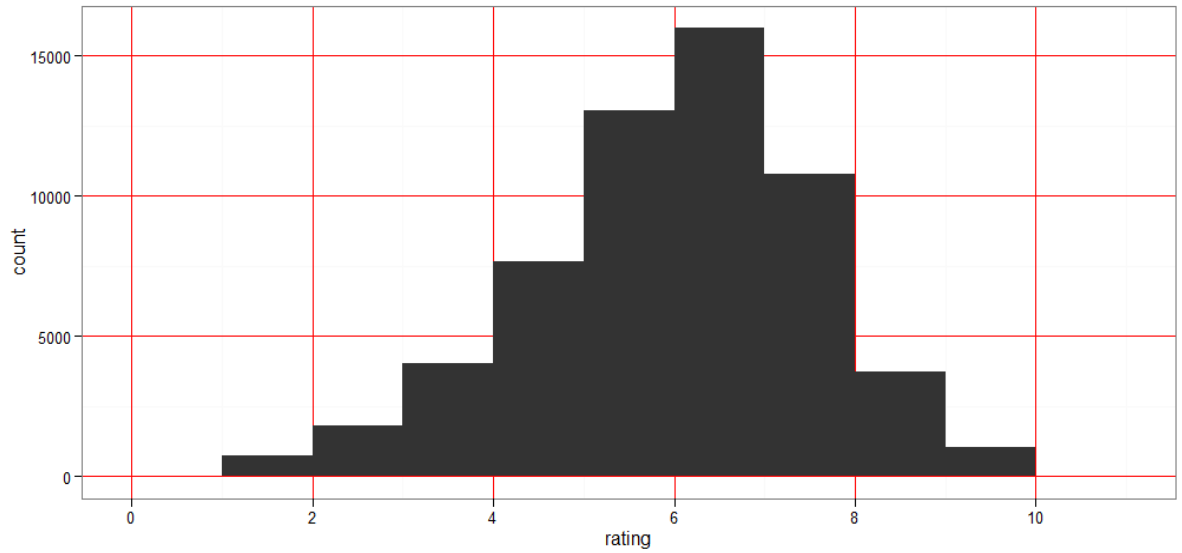
```
> theme_get()
$axis.line
theme_blank()

$axis.text.x
theme_text(family = base_family, size = base_size * 0.8, vjust = 1,
    lineheight = 0.9)

$axis.text.y
theme_text(family = base_family, size = base_size * 0.8, hjust = 1,
    lineheight = 0.9)
…
…
```

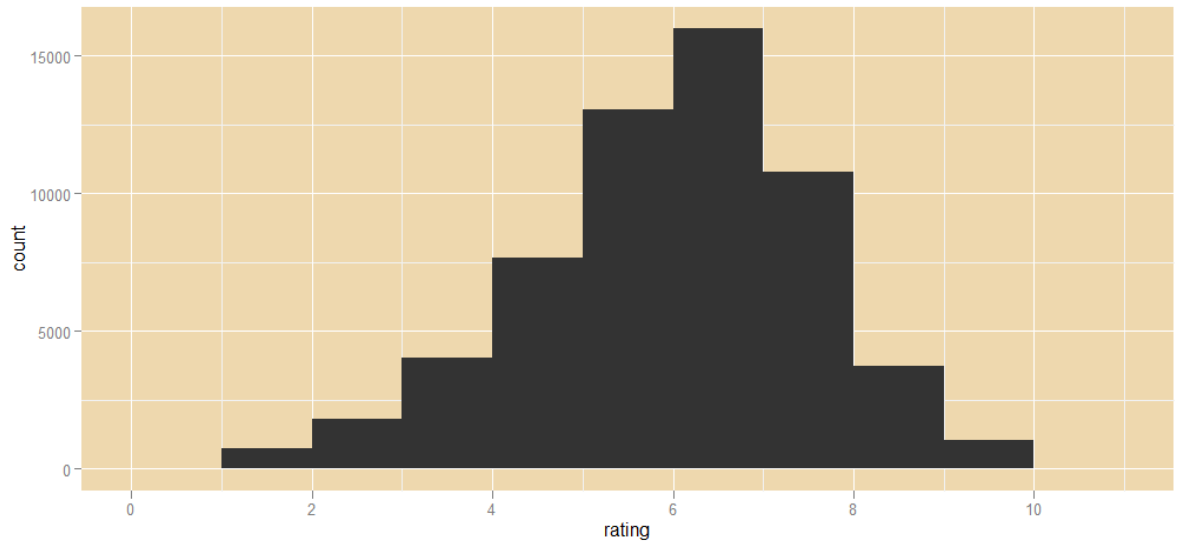
We can make a variation on the current plot theme by using the opts() command.

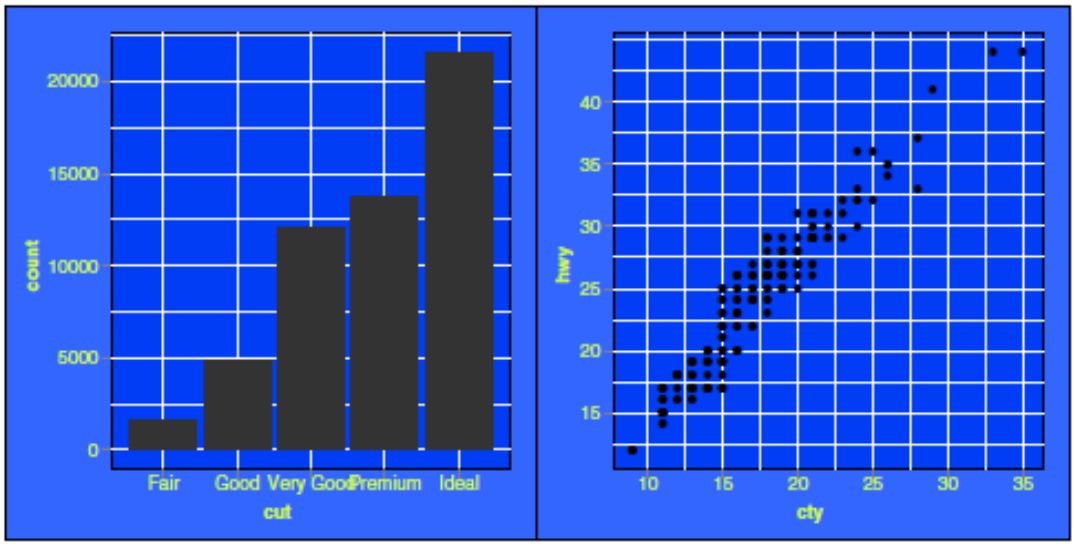
```
hgram + opts(panel.grid.major = theme_line(colour = "red"))
```



Let's try change the back ground colour.

```
hgram + opts(panel.background=theme_rect(fill = "wheat2", colour =
NA))
```





Create your own theme

- The process of creating your own theme is not a simple process for beginners.
- It is useful to know that theme_bw and theme_grey are functions.
- The source code for these functions can beaccessed by writing the name of the theme – with no brackets. Copy (i.e. ctrl V) the source code.

```
>theme_bw
```

- Create a new theme by defining the new theme name and pasting the source code.

```
>KevTheme  = function (base_size = 12, base_family = "")
{
    structure(list(axis.line = theme_blank(),
        axis.text.x = theme_text(family = base_family,
        size = base_size * 0.8, lineheight = 0.9, vjust = 1),
        …
        …
```

- This is just the same as the old theme. However, you can edit the source code using the edit command. A Dialogue box opens that allows the user to re-specify the settings.

```
>edit(KevTheme)
```

- From here on – trial and error is the best approach – each user would have their own preferences.