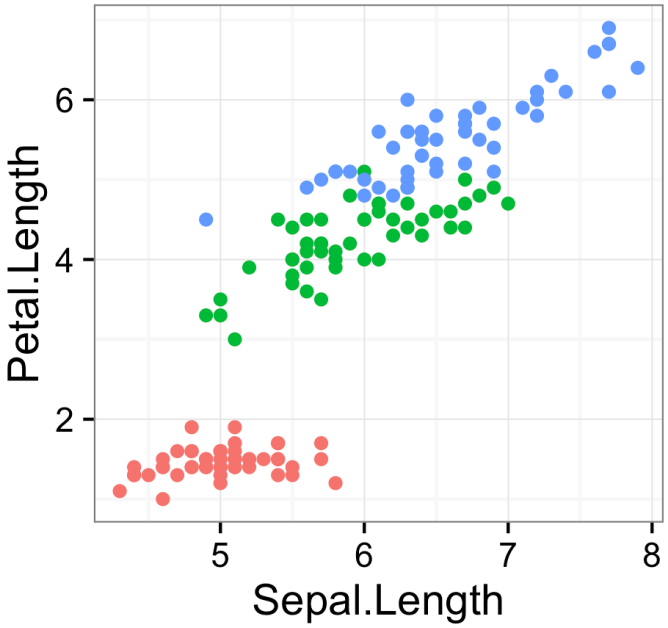
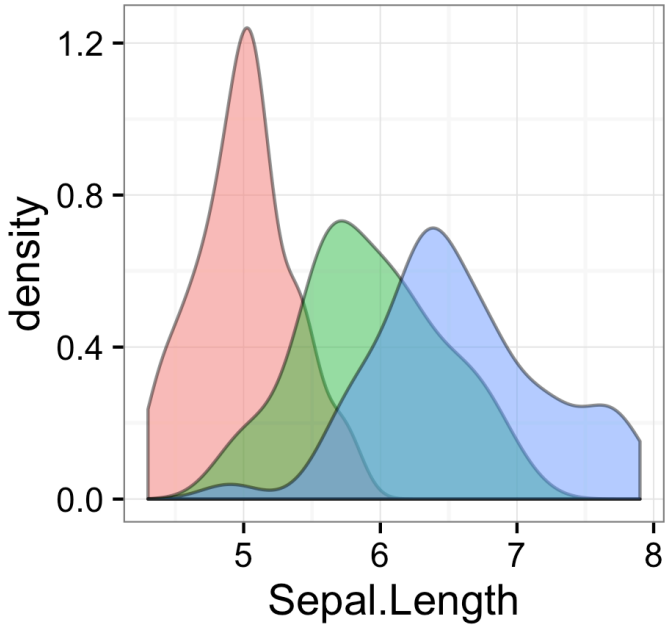
# We often need to do statistical transformations before plotting

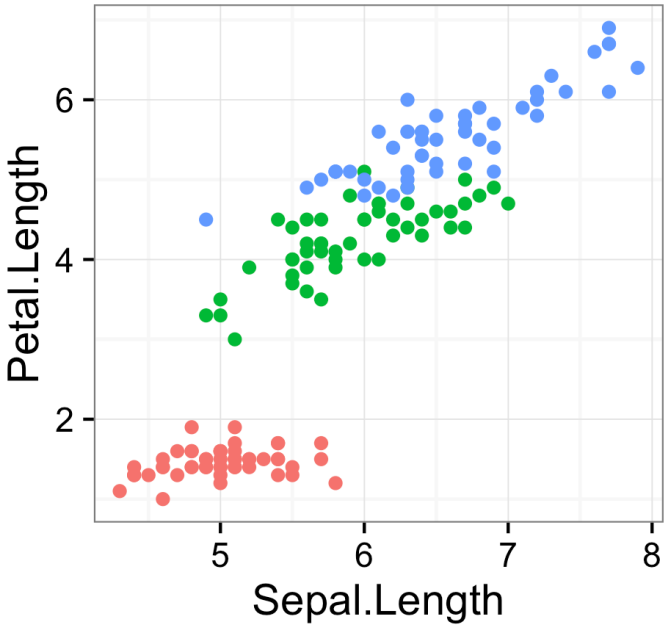




density of data points

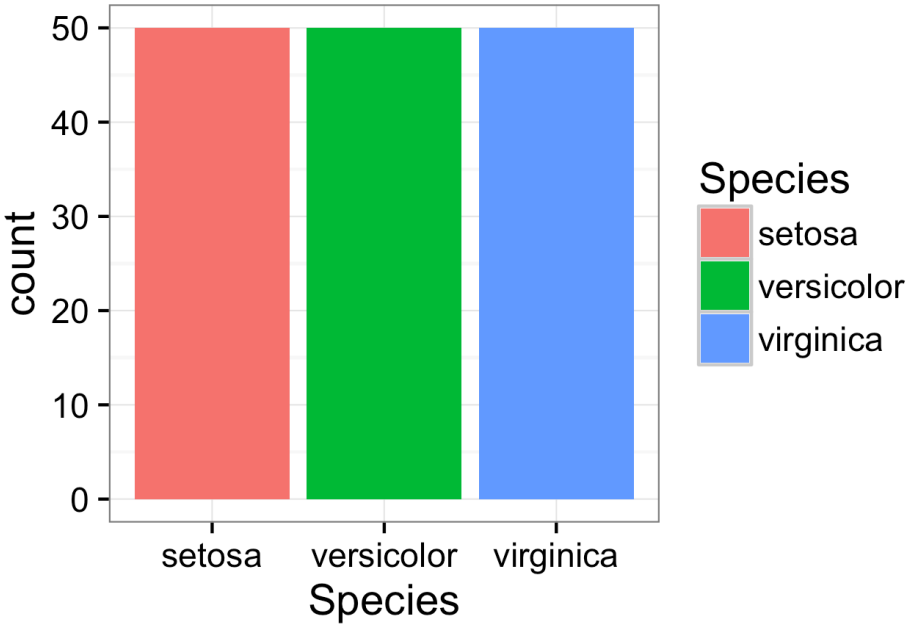



# We often need to do statistical transformations before plotting

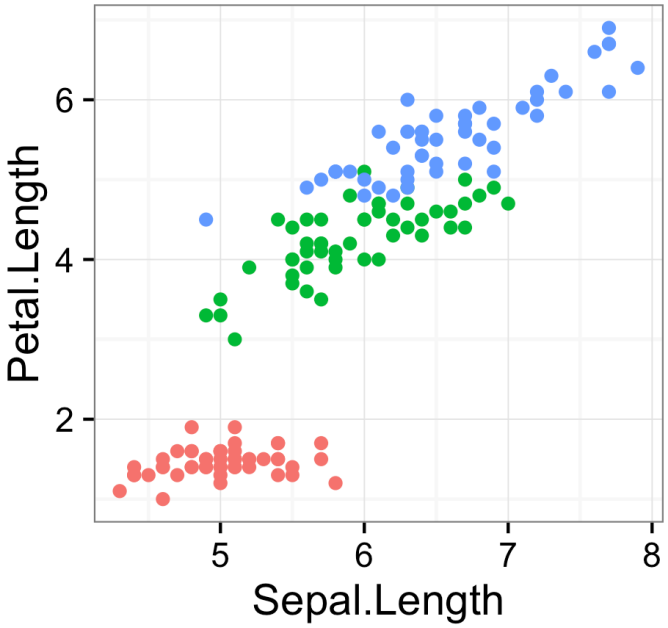




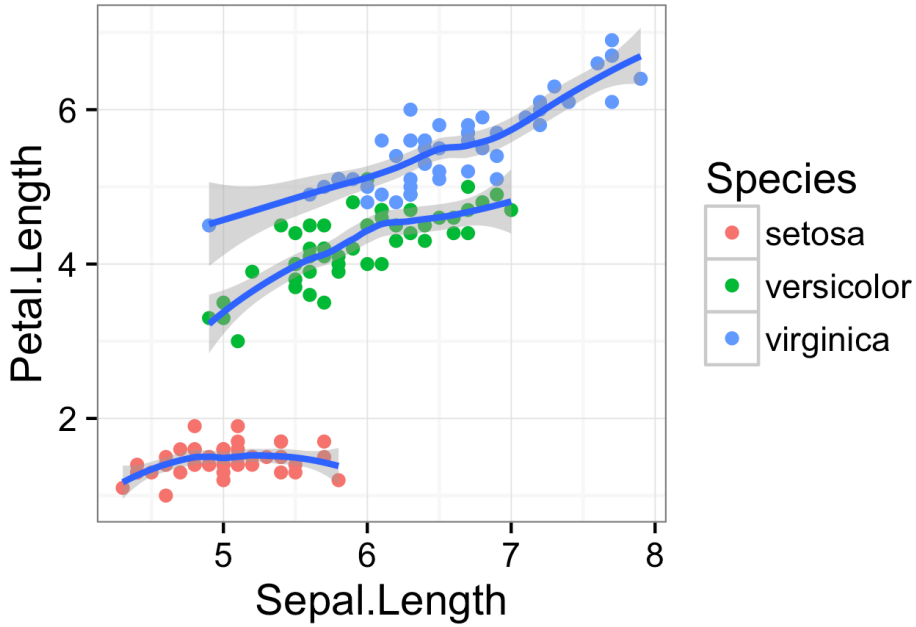
count of number of different types



# We often need to do statistical transformations before plotting

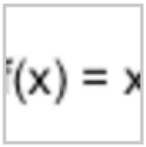



statistical smoothing/ trend lines



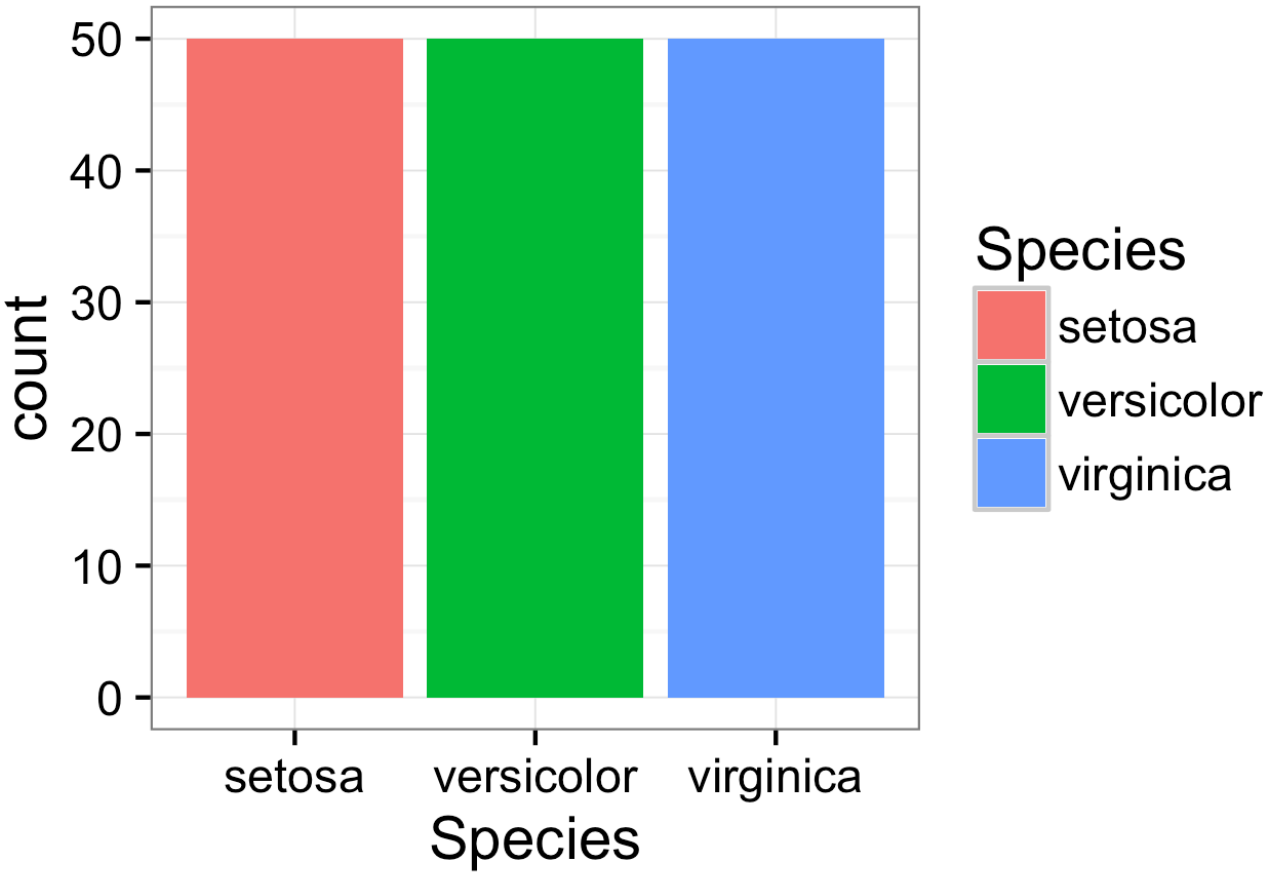
# In ggplot2, these transformations are done with stats

- `stat_ecdf`
  Empirical Cumulative Density Function
- `stat_ellipse`
  Plot data ellipses.
- `stat_function`
  Superimpose a function.
- `stat_identity`
  Identity statistic.
- `stat_qq` (geom_qq)
  Calculation for quantile-quantile plot.
- `stat_summary_2d` (stat_summary2d, stat_summary_hex)
  Bin and summarise in 2d (rectangle & hexagons)
- `stat_unique`
  Remove duplicates.



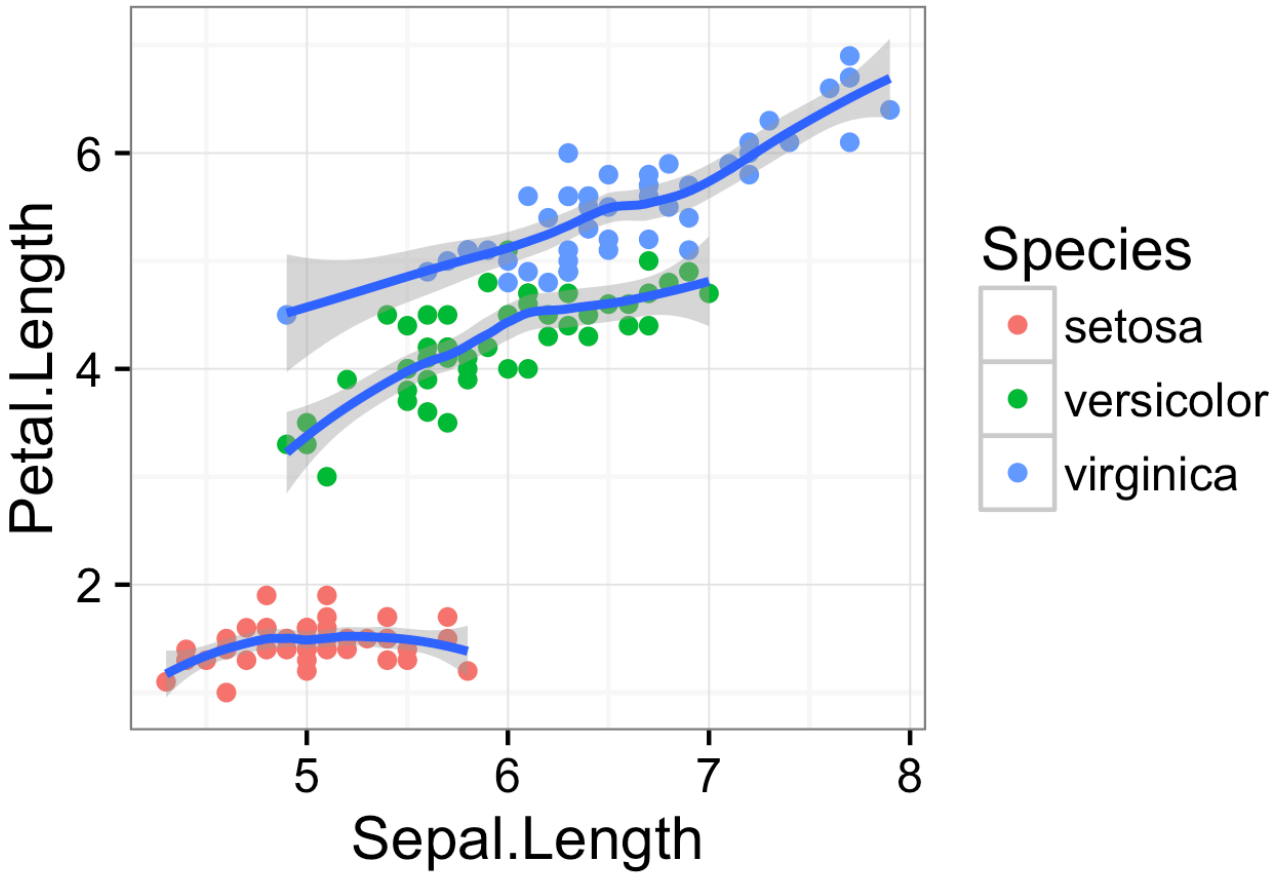
# In most cases we just need to call the appropriate geom and it calls a stat

```
ggplot(iris, aes(x=Species, fill=Species)) +
                                    geom_bar()
```



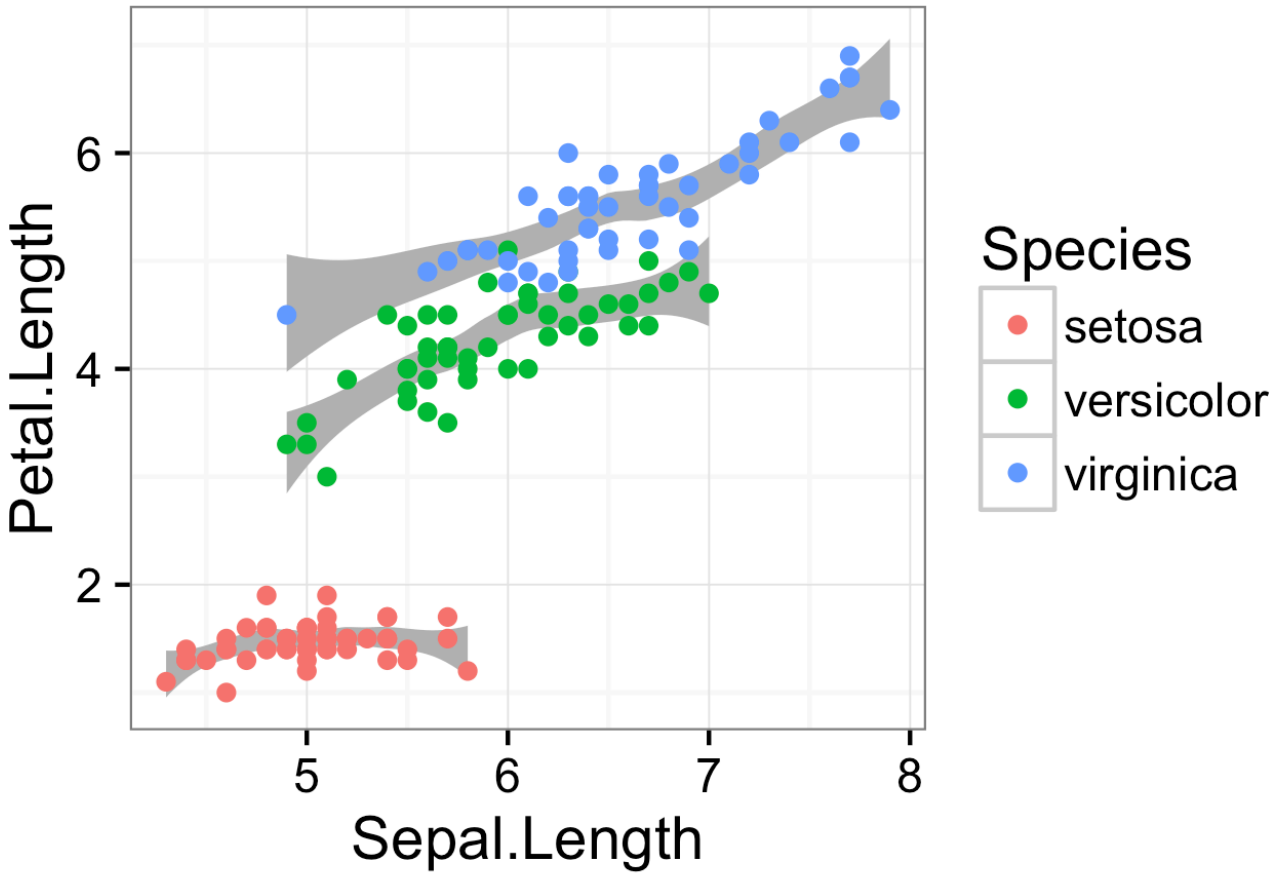
# In most cases we just need to call the appropriate geom and it calls a stat

```
ggplot(iris, aes(x=Sepal.Length, y=Petal.Length)) +
                    geom_point(aes(color=Species)) +
                    geom_smooth(aes(group=Species))
```



# However, sometimes it can be helpful to call the stat directly

```
ggplot(iris, aes(x=Sepal.Length, y=Petal.Length)) +
stat_smooth(aes(group=Species), geom="ribbon", fill='gray70') +
                                        geom_point(aes(color=Species))
```

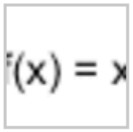


# Scales define how to map data onto aesthetics

- `scale_colour_grey` (scale_color_grey, scale_fill_grey)
  Sequential grey colour scale.

- `scale_colour_hue` (scale_color_discrete, scale_color_hue, scale_colour_discrete, scale_fill_discrete, scale_fill_hue)
  Qualitative colour scale with evenly spaced hues.

- `scale_identity` (scale_alpha_identity, scale_color_identity, scale_colour_identity, scale_fill_identity, scale_linetype_identity, scale_shape_identity, scale_size_identity)
  Use values without scaling.



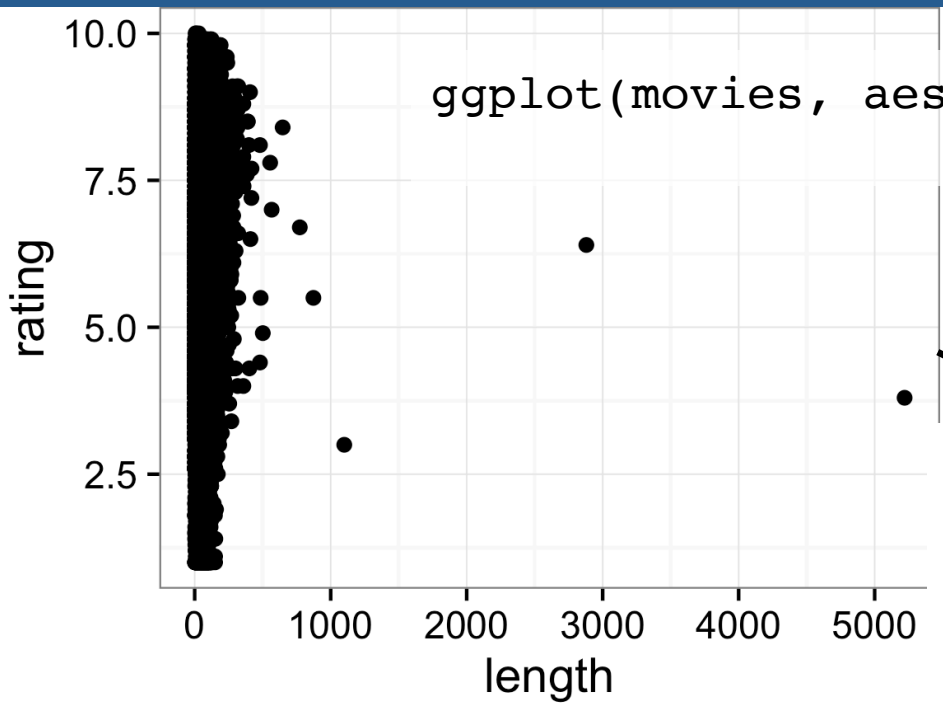

- `scale_manual` (scale_alpha_manual, scale_color_manual, scale_colour_manual, scale_fill_manual, scale_linetype_manual, scale_shape_manual, scale_size_manual)
  Create your own discrete scale.


- `scale_linetype` (scale_linetype_continuous, scale_linetype_discrete)
  Scale for line patterns.

- `scale_shape` (scale_shape_continuous, scale_shape_discrete)
  Scale for shapes, aka glyphs.

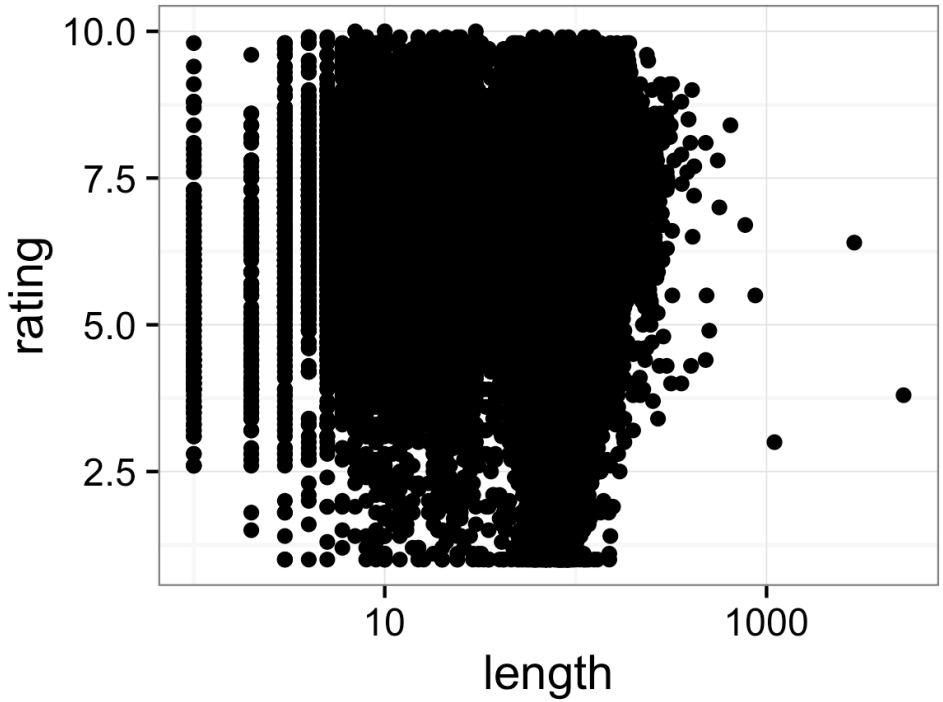
- `scale_size` (scale_radius, scale_size_area, scale_size_continuous, scale_size_discrete)

# Example 1: Change scaling of x axis

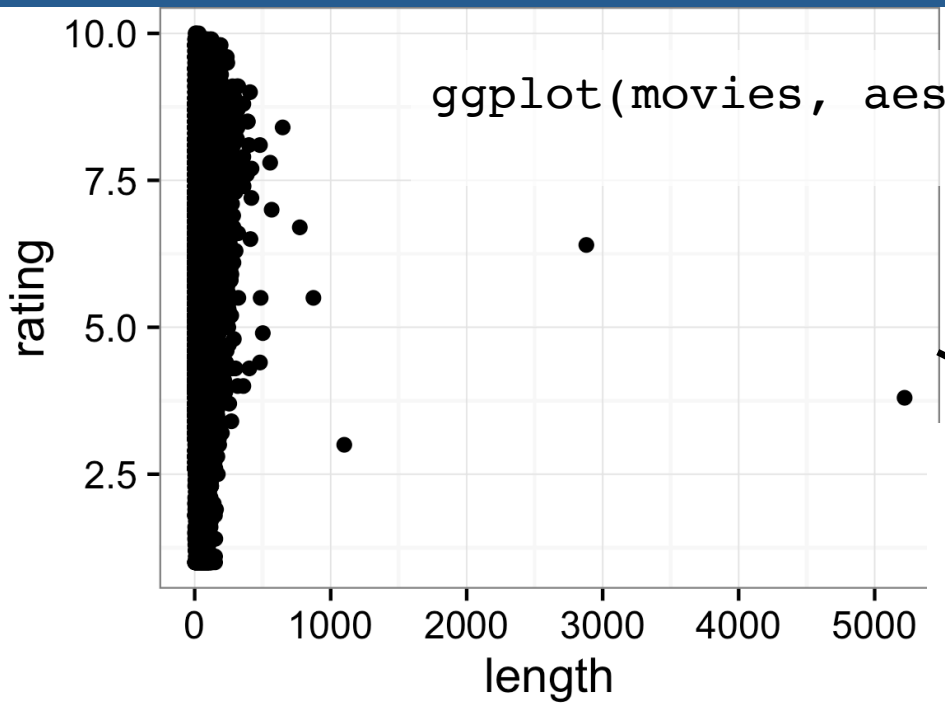



```
ggplot(movies, aes(x=length, y=rating)) +
                                    geom_point()
```

```
+ scale_x_log10()
```

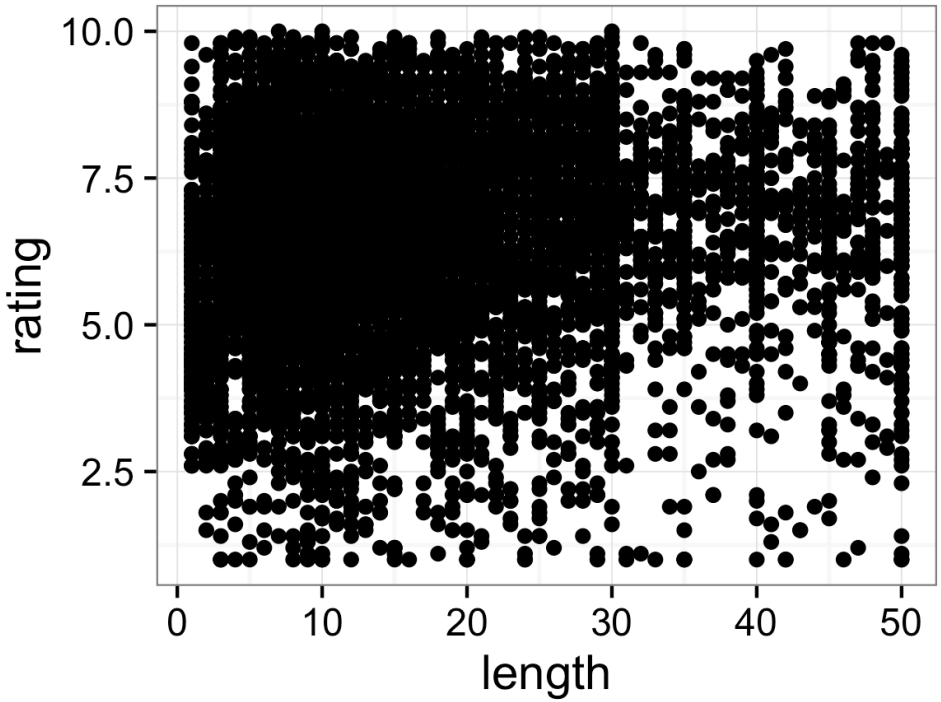


# Example 1: Change scaling of x axis

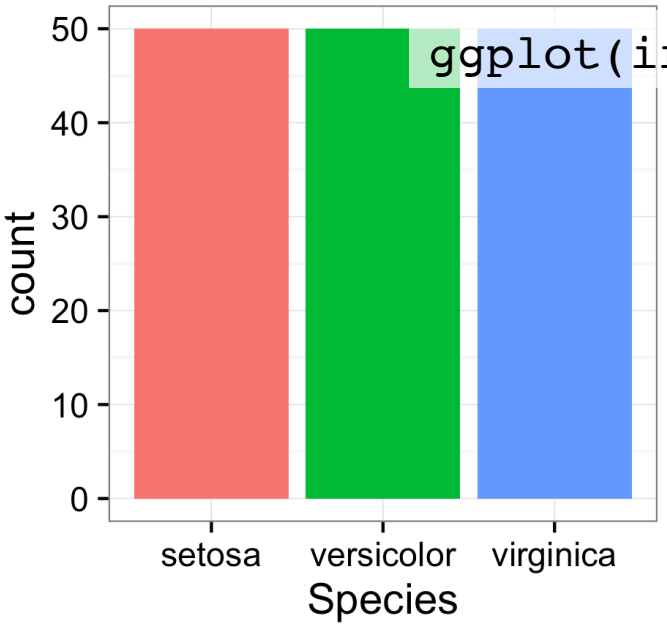



```
ggplot(movies, aes(x=length, y=rating)) +
                                    geom_point()
```

```
+ xlim(1, 50)
```

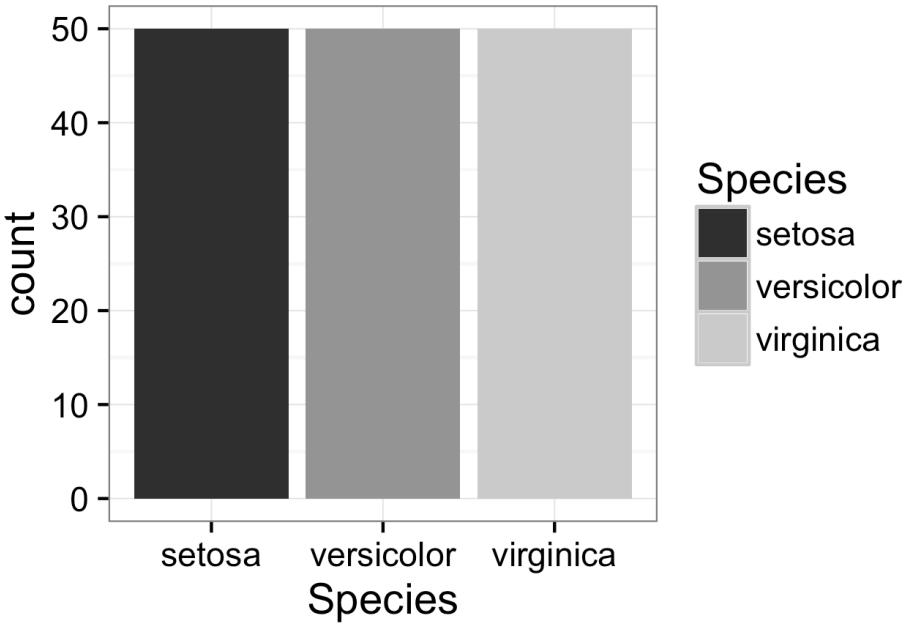


# Example 2: Change color scaling

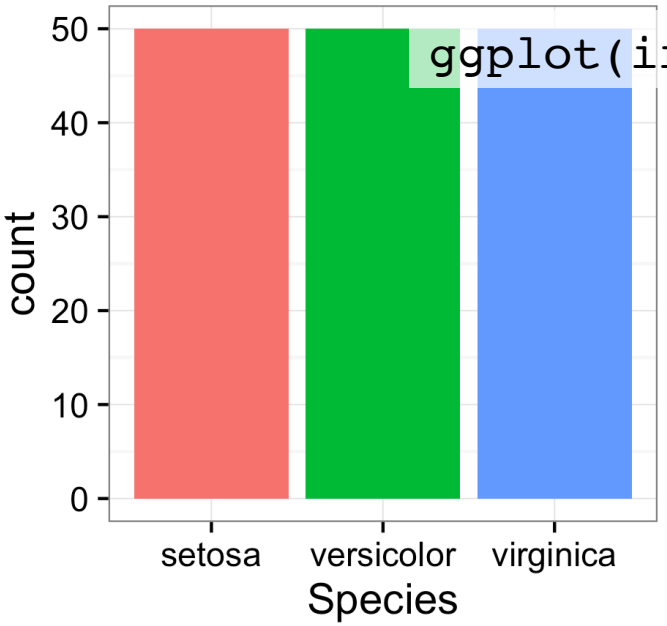



```
ggplot(iris, aes(x=Species, fill=Species)) +
                                        geom_bar()
```


```
+ scale_fill_grey()
```



# Example 2: Change color scaling




```
ggplot(iris, aes(x=Species, fill=Species)) +
                                        geom_bar()
```


```
+ scale_fill_brewer()
```