



DATA VISUALIZATION WITH GGPLOT2

Coordinates

Coordinates Layer

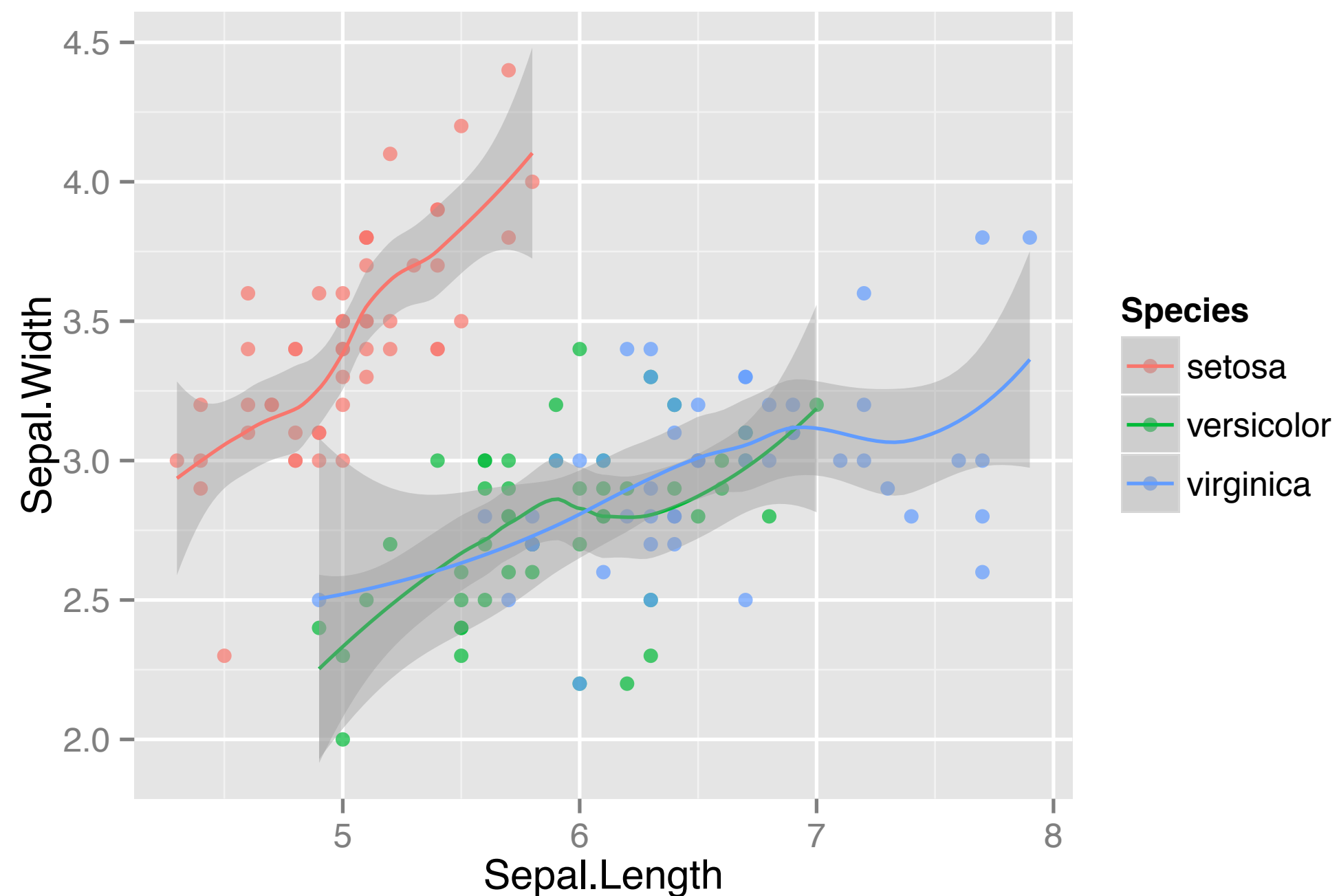
- Controls plot dimensions
- `coord_`
- `coord_cartesian()`

Zooming in

- `scale_x_continuous(limits = ...)`
- `xlim()`
- `coord_cartesian(xlim = ...)`

Original Plot

```
> iris.smooth <- ggplot(iris, aes(x = Sepal.Length,  
                                y = Sepal.Width,  
                                col = Species)) +  
  geom_point(alpha = 0.7) + geom_smooth()  
> iris.smooth
```



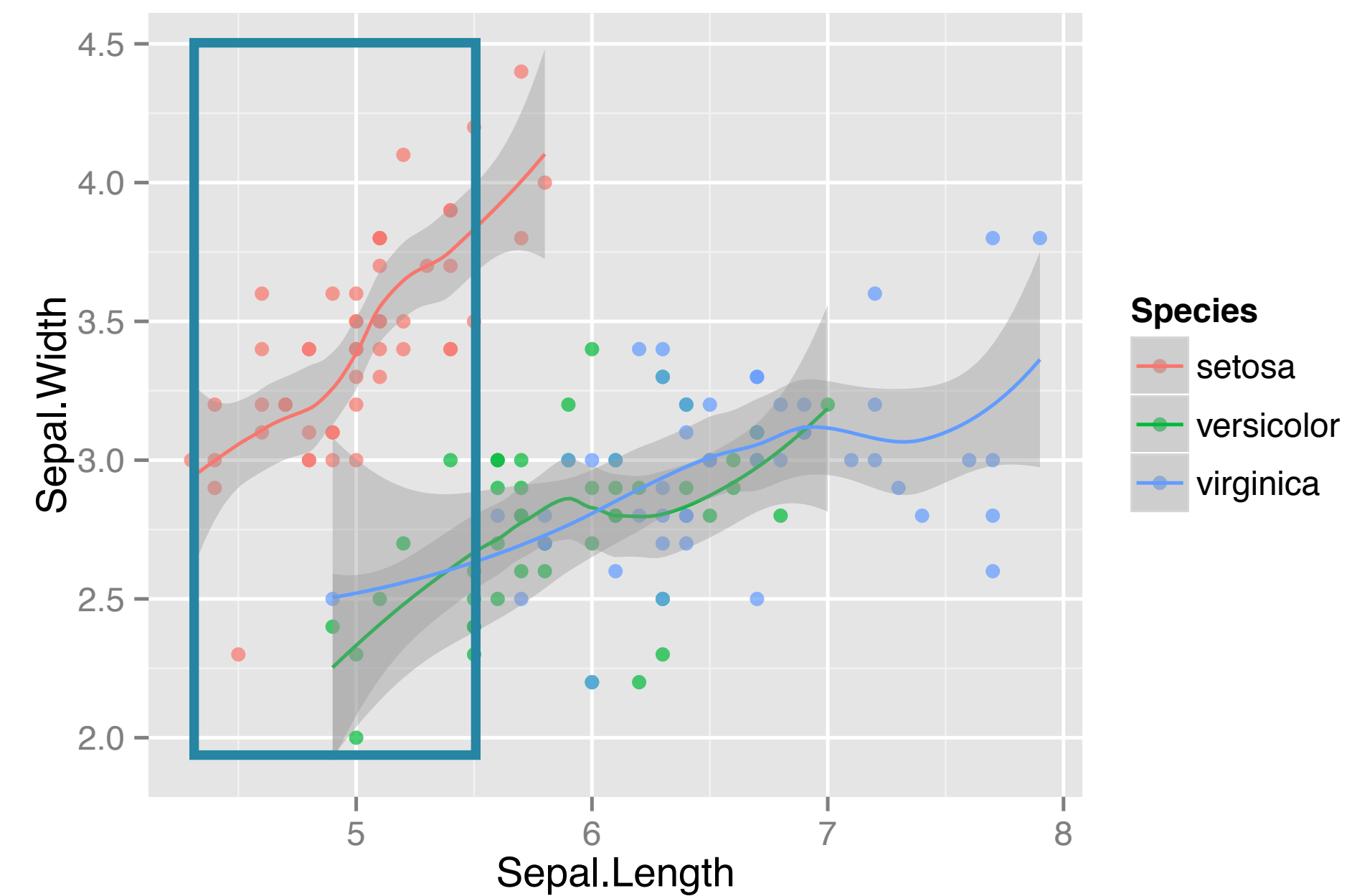
scale_x_continuous

```
> iris.smooth + scale_x_continuous(limits = c(4.5, 5.5))  
geom_smooth: method="auto" and size of largest group is <1000, so  
using loess. Use 'method = x' to change the smoothing method.  
Warning messages:  
1: Removed 7 rows containing missing values (stat_smooth).  
2: Removed 39 rows containing missing values (stat_smooth).  
3: Removed 49 rows containing missing values (stat_smooth).  
4: Removed 95 rows containing missing values (geom_point).
```

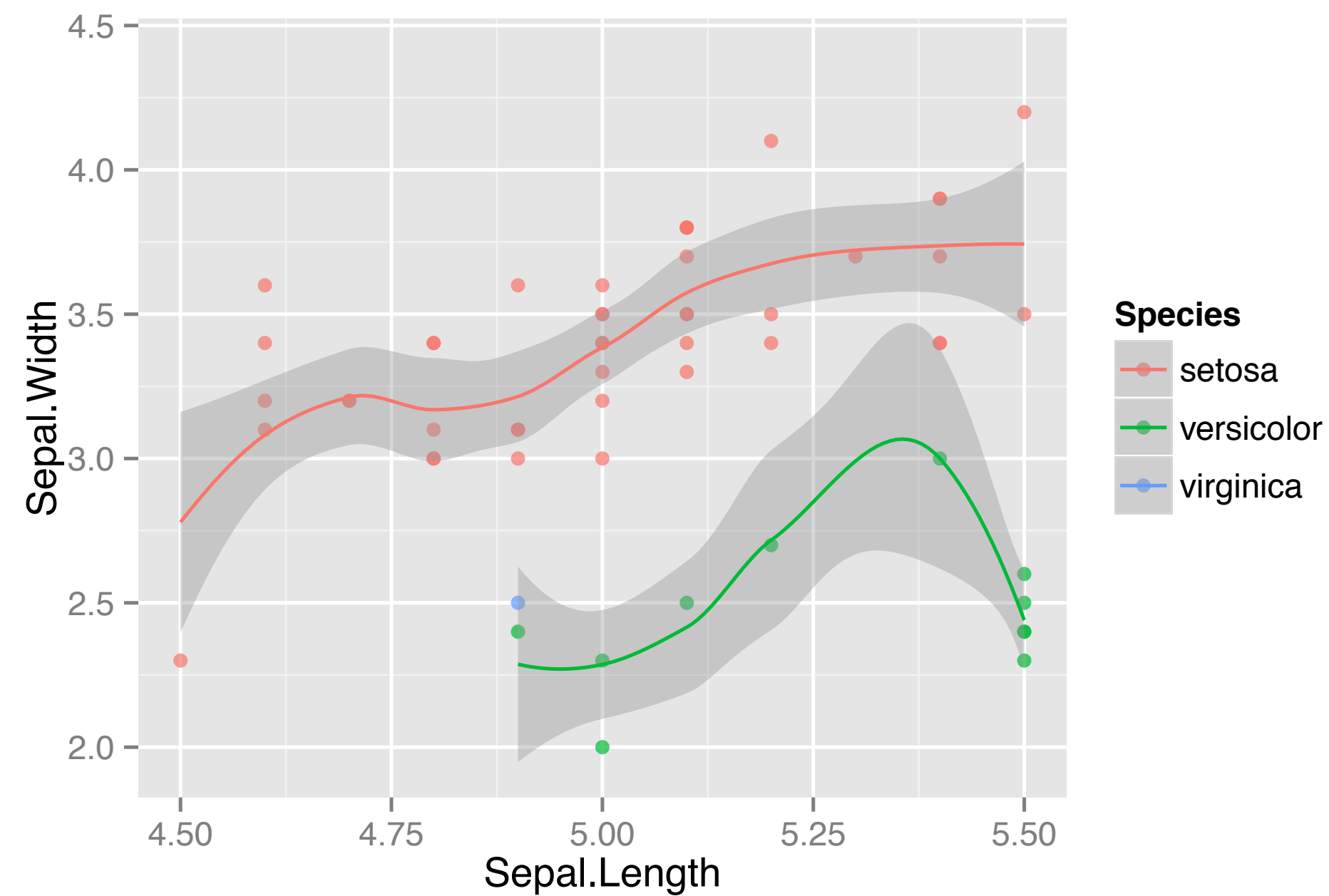
This filters out rows not fitting to the limits; basically this is due to using `scale_x_continuous` or even just `xlim(c(3.5,4.5))` to set limits
-> in contrast using `coords_` achieves true zooming in and LOESS curves continue as if other data were present and we simply put `xlim` inside `coords_` layer

scale_x_continuous

```
> iris.smooth + scale_x_continuous(limits = c(4.5, 5.5))
```



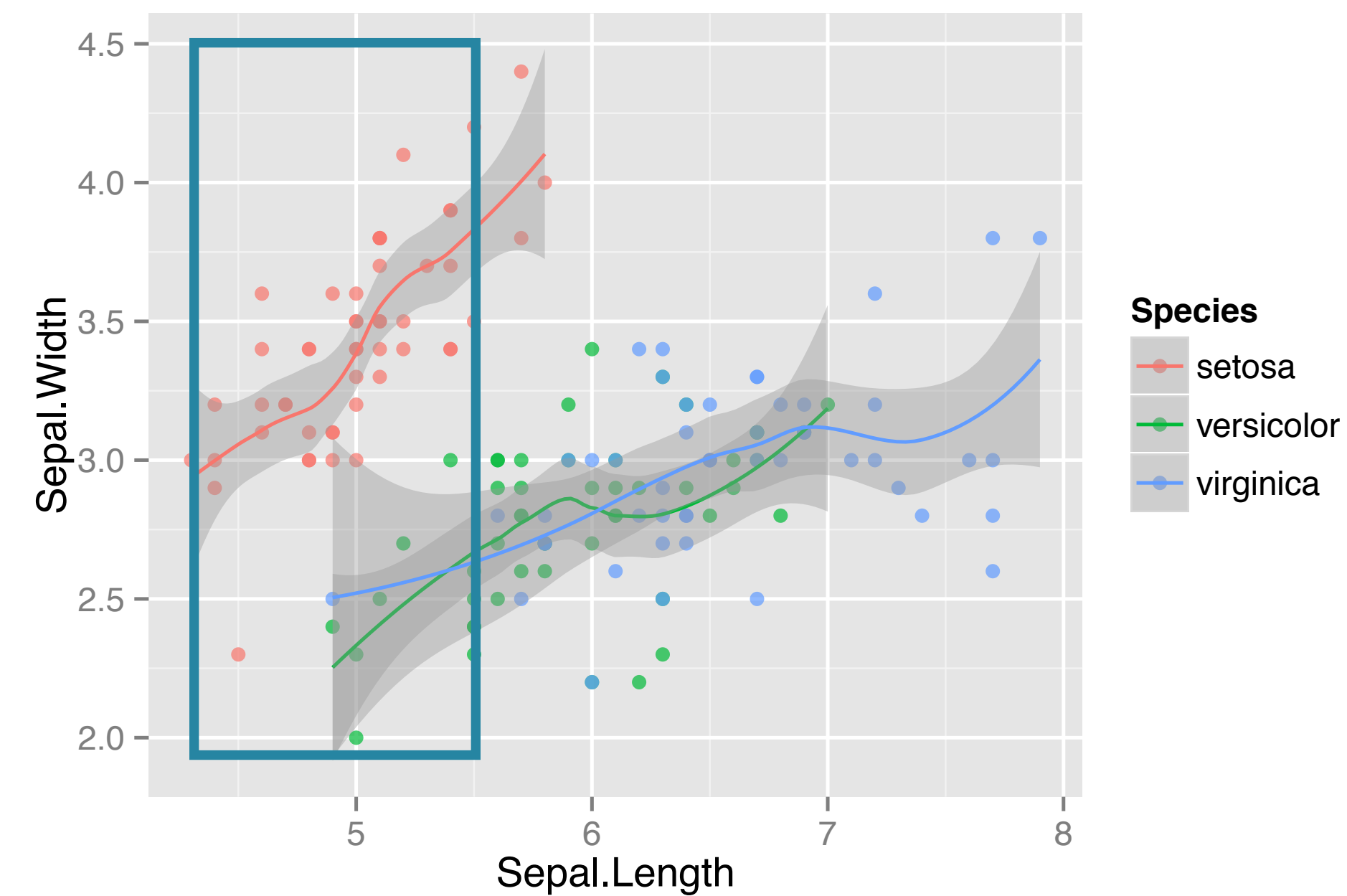
Original



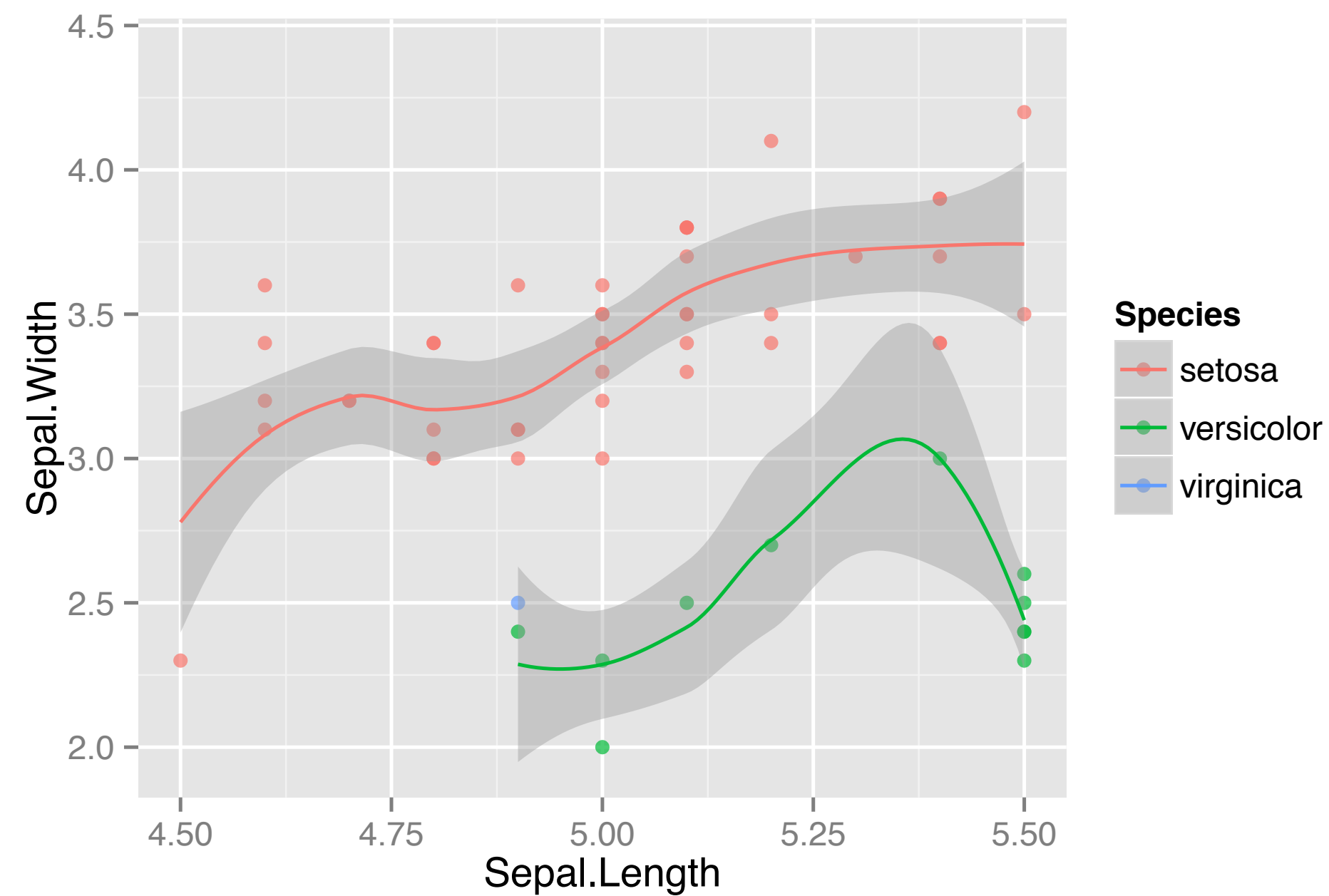
zoomed in, `scale_x_continuous()`
Parts of original data set filtered!

xlim

```
> iris.smooth + xlim(c(4.5, 5.5))  
Warning messages: ...
```



Original

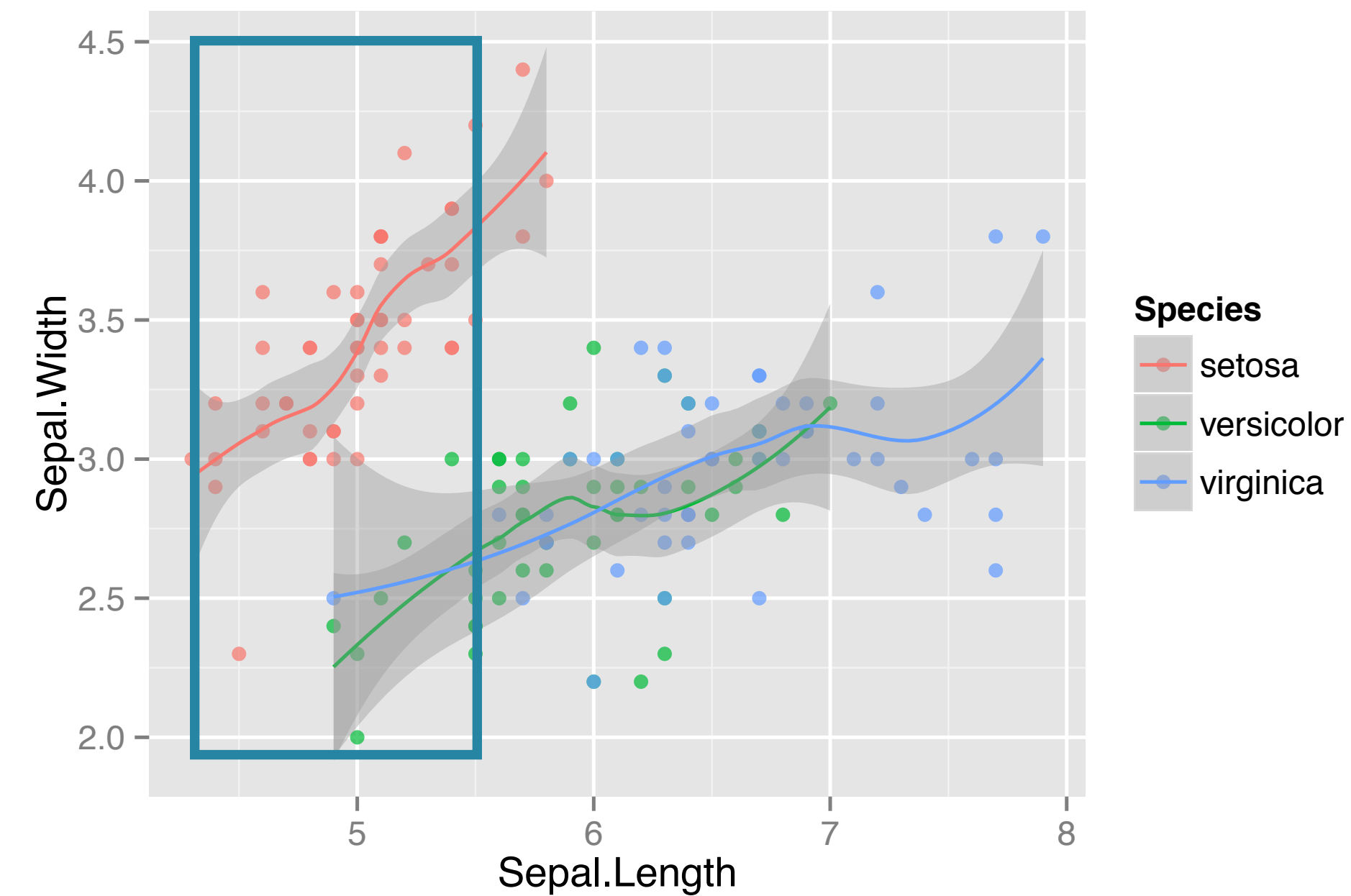


zoomed in, xlim()

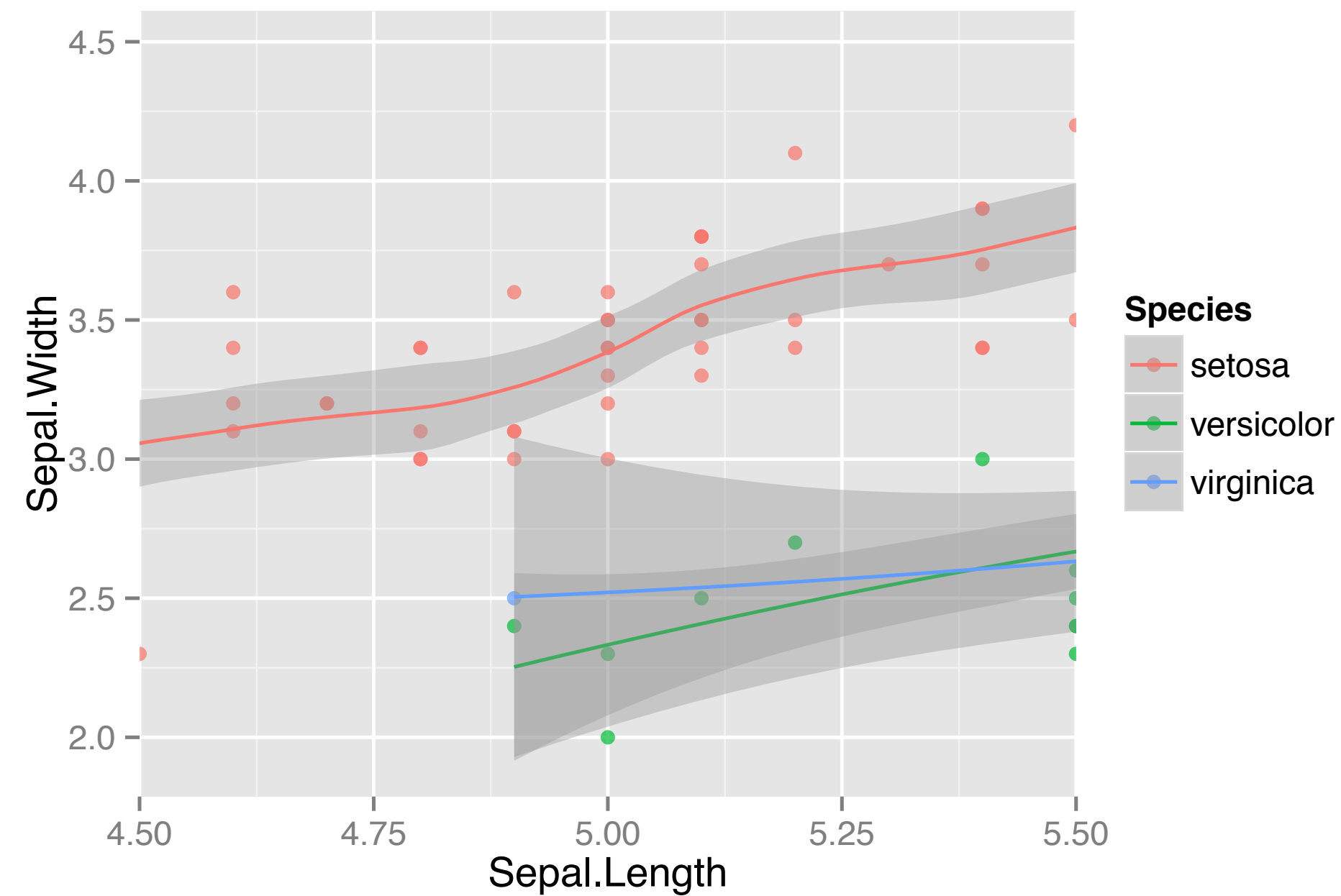
Same effect as scale_x_continuous()

coord_cartesian

```
> iris.smooth + coord_cartesian(xlim = c(4.5, 5.5))
```



Original



zoomed in, xlim()
just zoom in

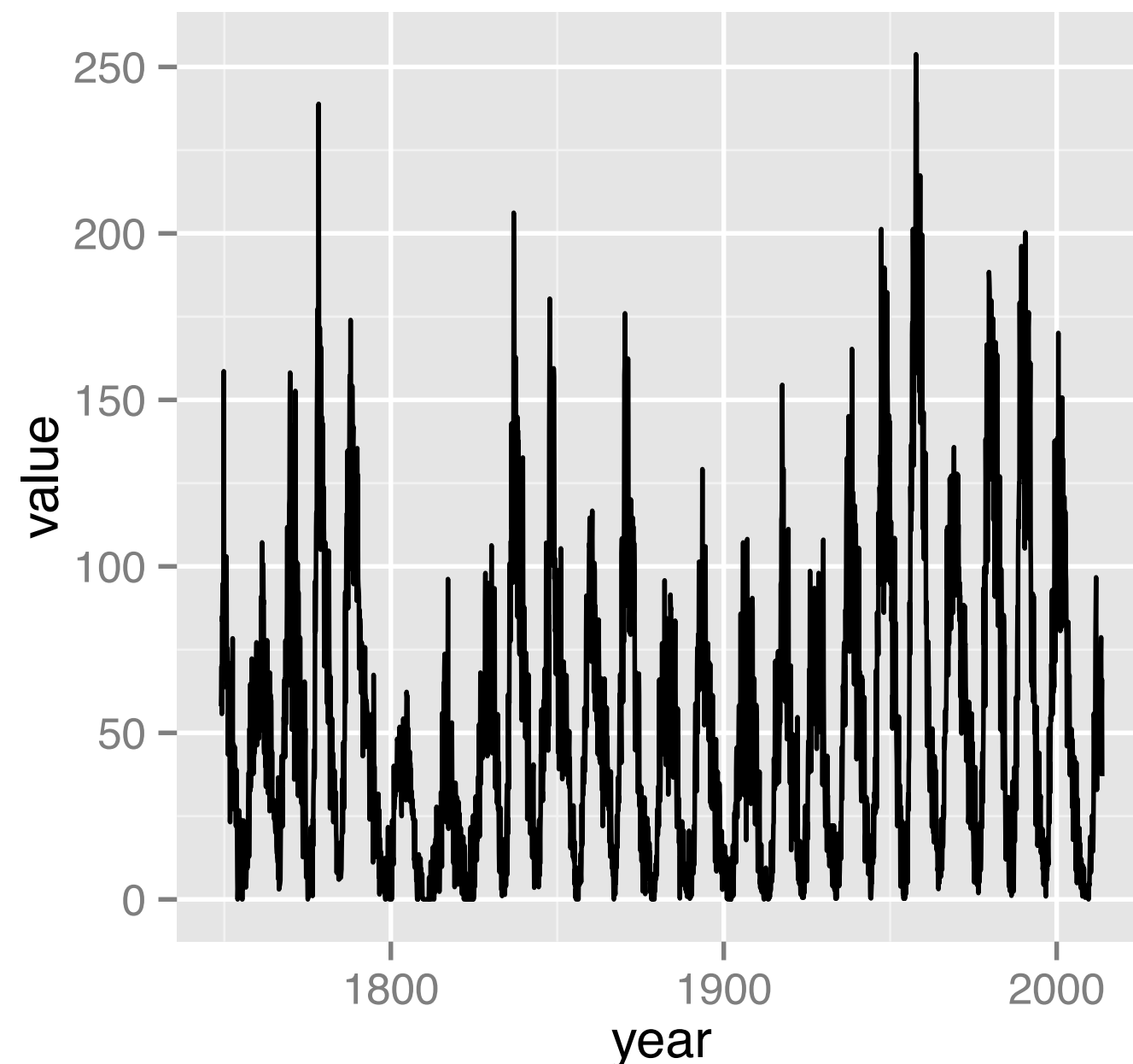
Aspect Ratio

- Height-to-width ratio
- Deception!

Data often seem to convey a different message with different aspect ratio to the plot
- Standardization attempts
- Typically 1:1

Sunspots

```
> library(reshape2); library(zoo)
> sunspots.m <- data.frame(year = index(sunspot.month),
                           value = melt(sunspot.month)$value)
> ggplot(sunspots.m, aes(x = year, y = value)) +
  geom_line() +
  coord_equal() # a 1:1 aspect ratio
```

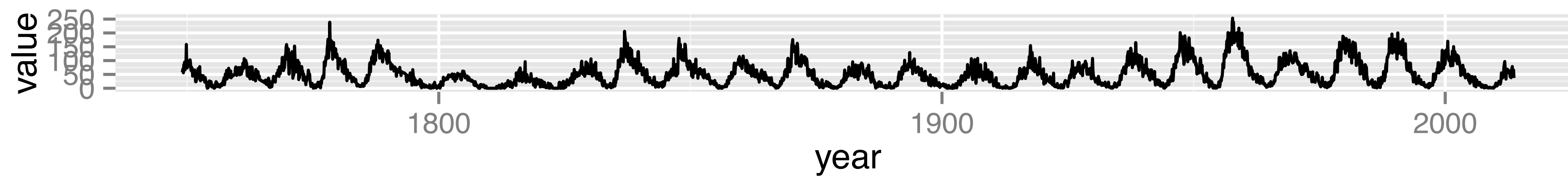


1. Oscillating period of 11 years
2. Sunspot numbers change over long periods

Sunspots

```
> ggplot(sunspots.m, aes(x = year, y = value)) +  
  geom_line() +  
  coord_fixed(0.055)
```

1. Oscillating period of 11 years
2. Sunspot numbers change over long periods
3. Sunspots arise more quickly than they disappear



and this we now can see
because we have squashed
the plot aspect ratio



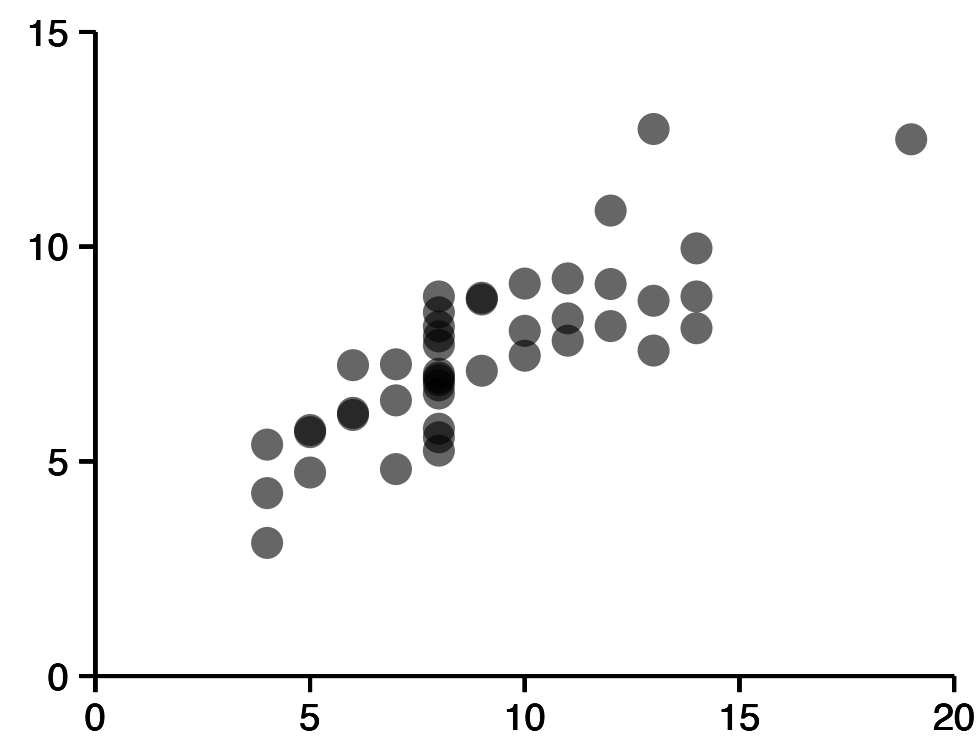
DATA VISUALIZATION WITH GGPLOT2

Facets

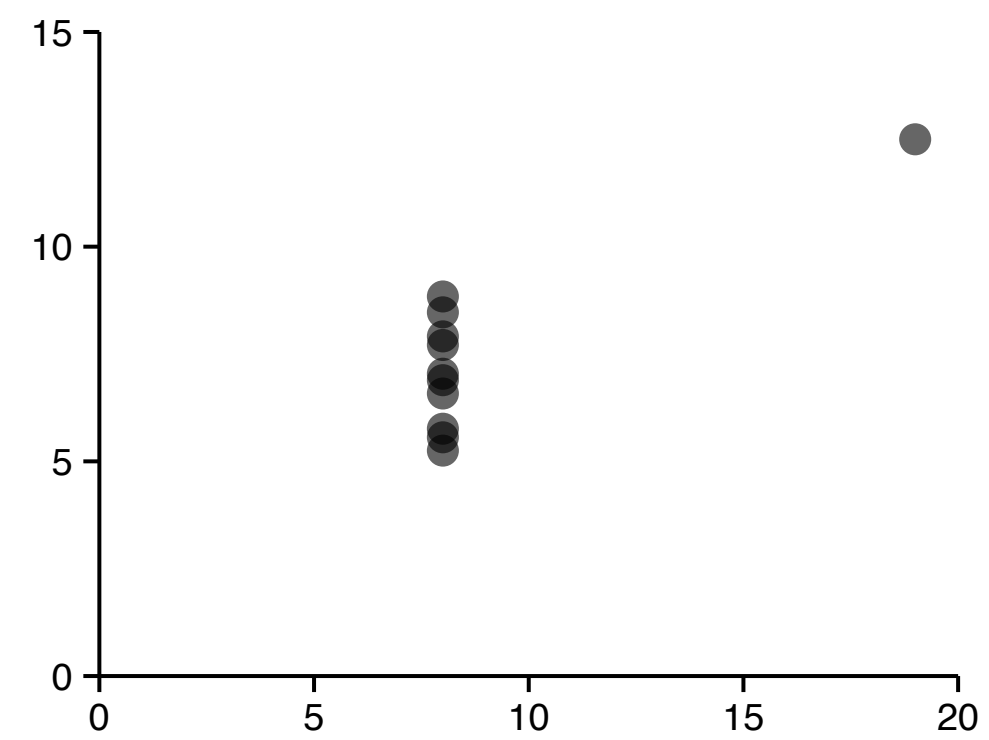
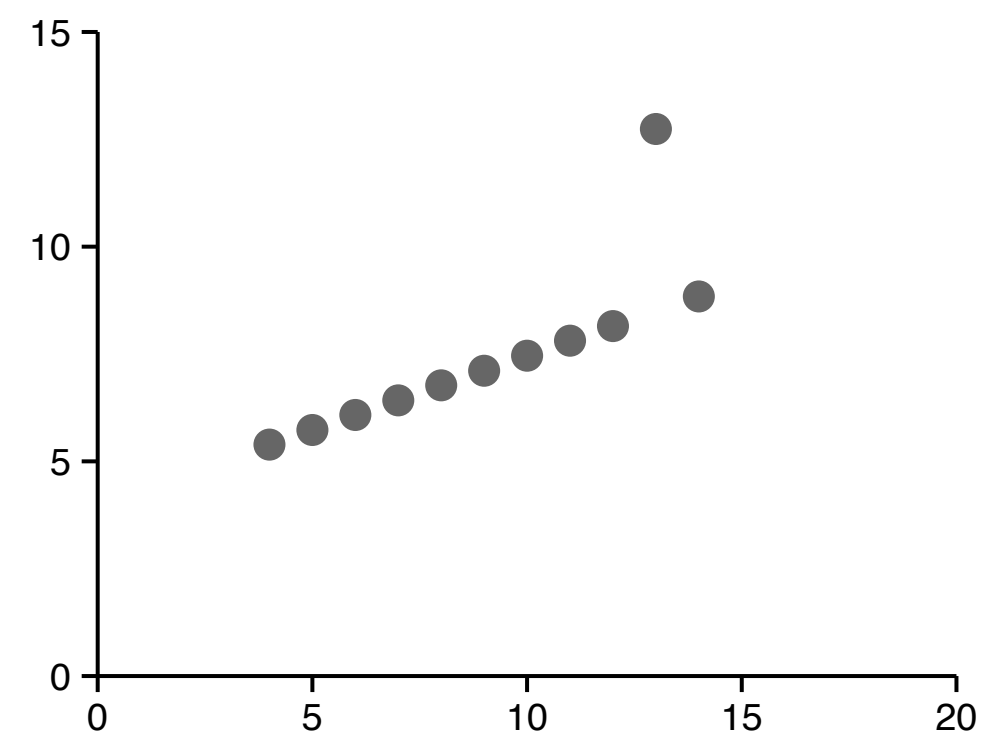
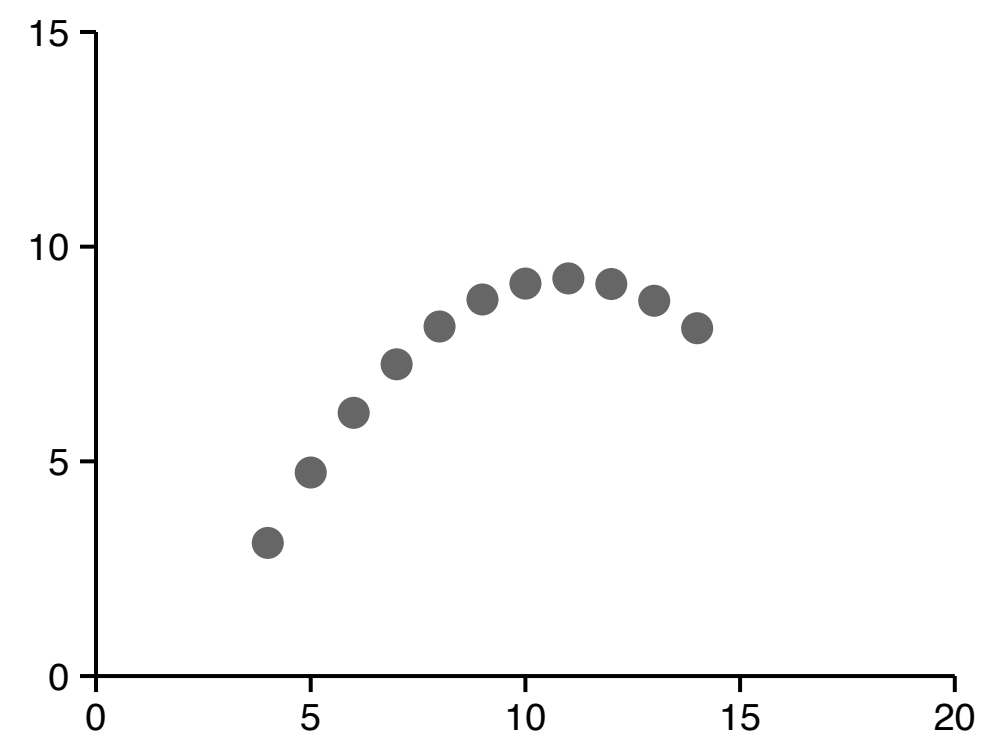
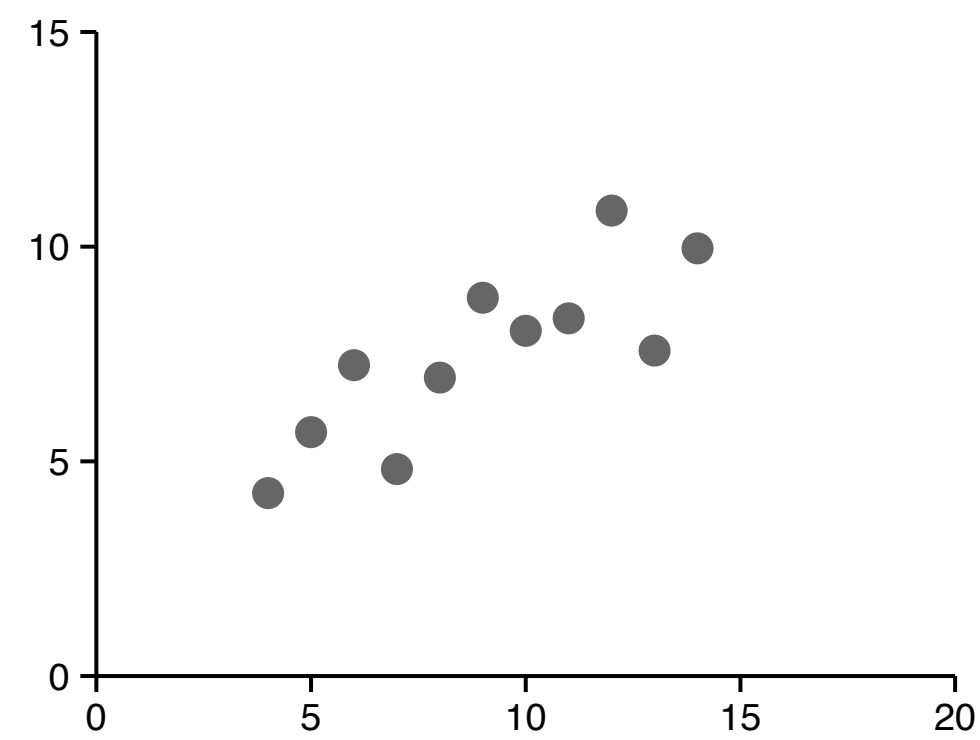
Facets

- Straight-forward yet useful
- Concept of Small Multiples
 - Edward Tufte
 - *Visualization of Quantitative Information, 1983*

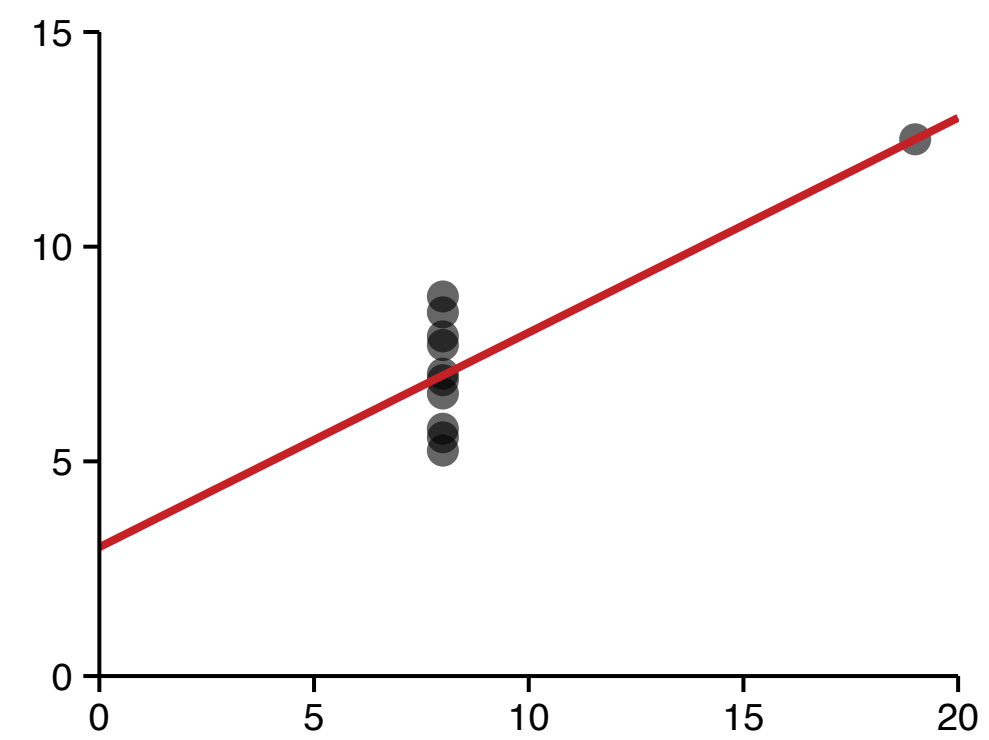
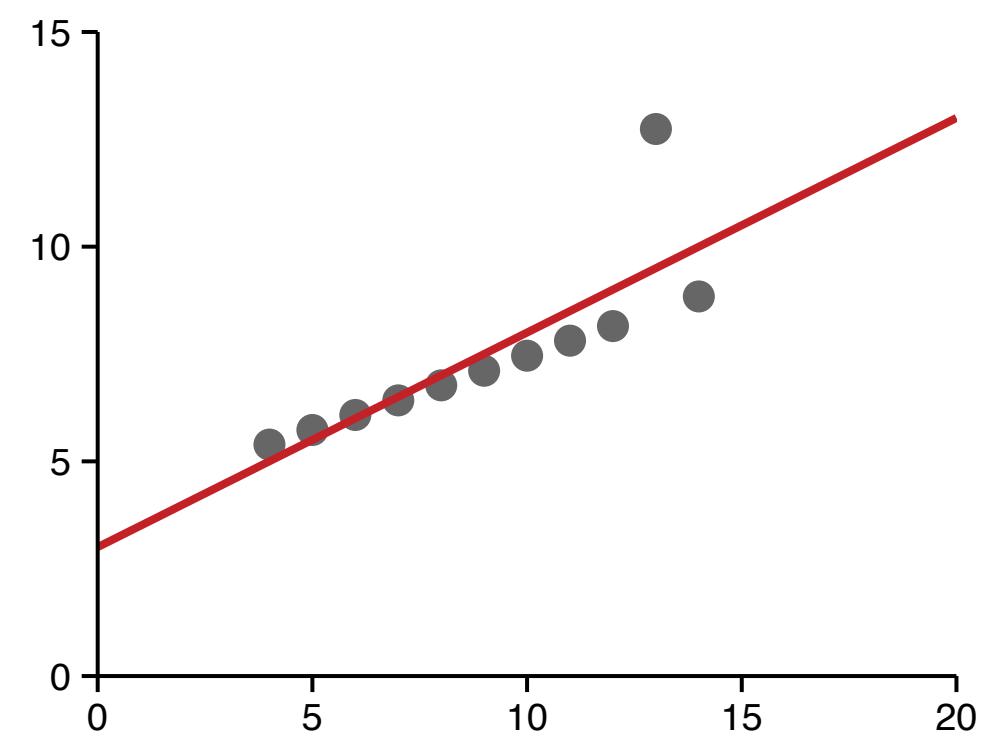
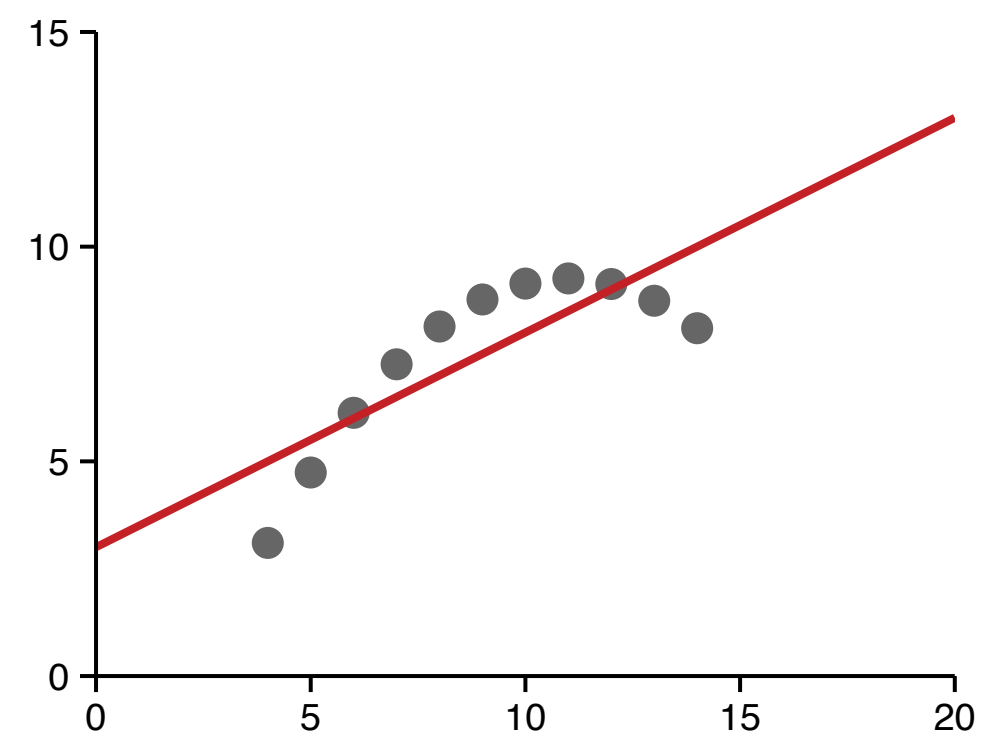
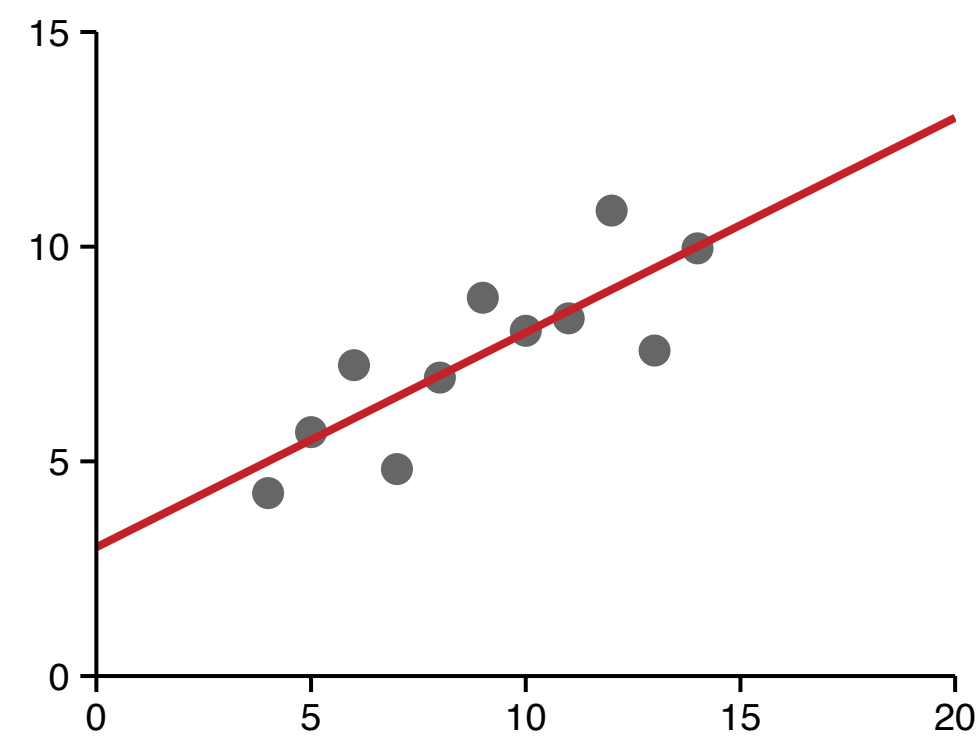
Facets



Facets

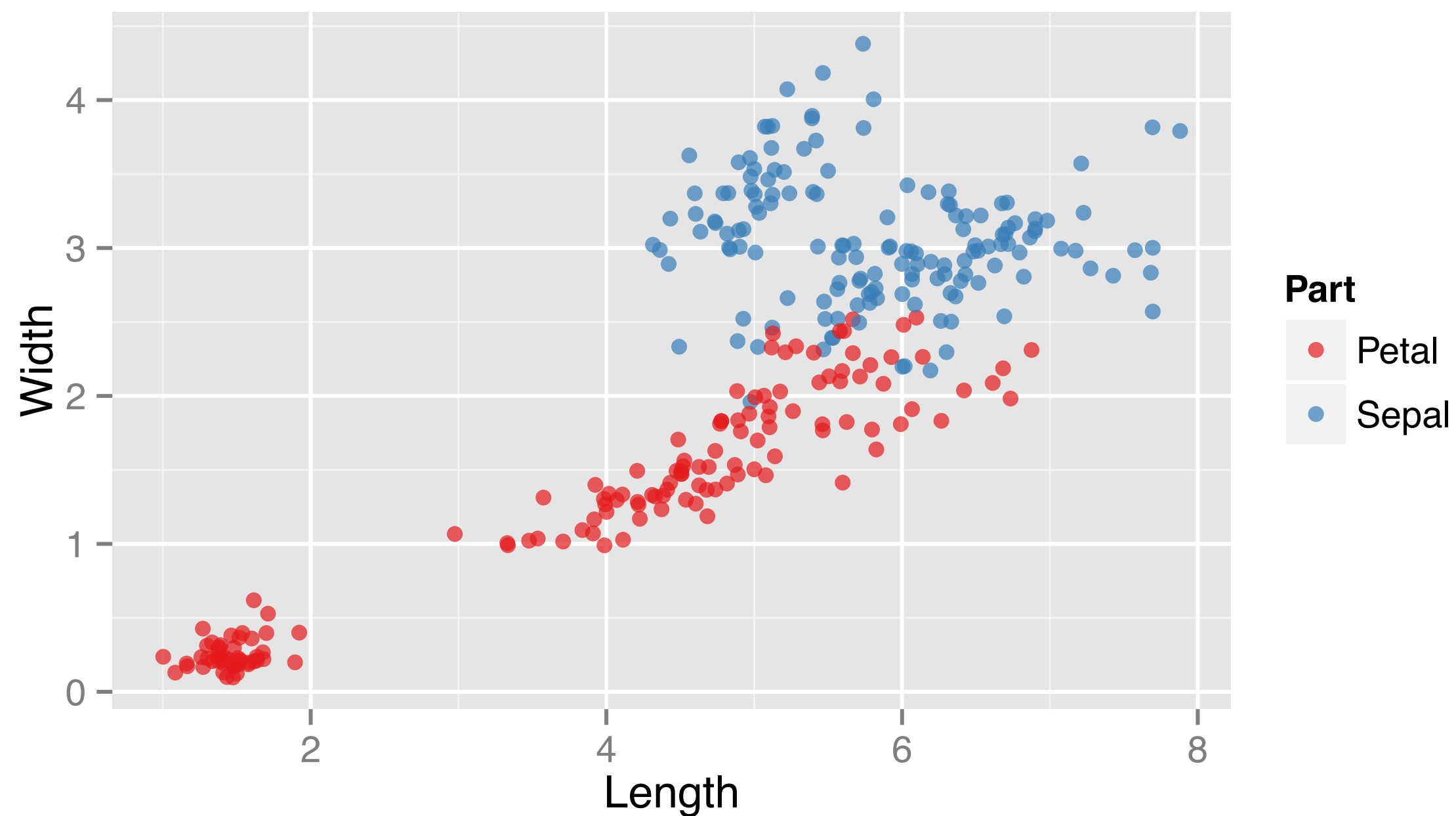


Facets



iris.wide

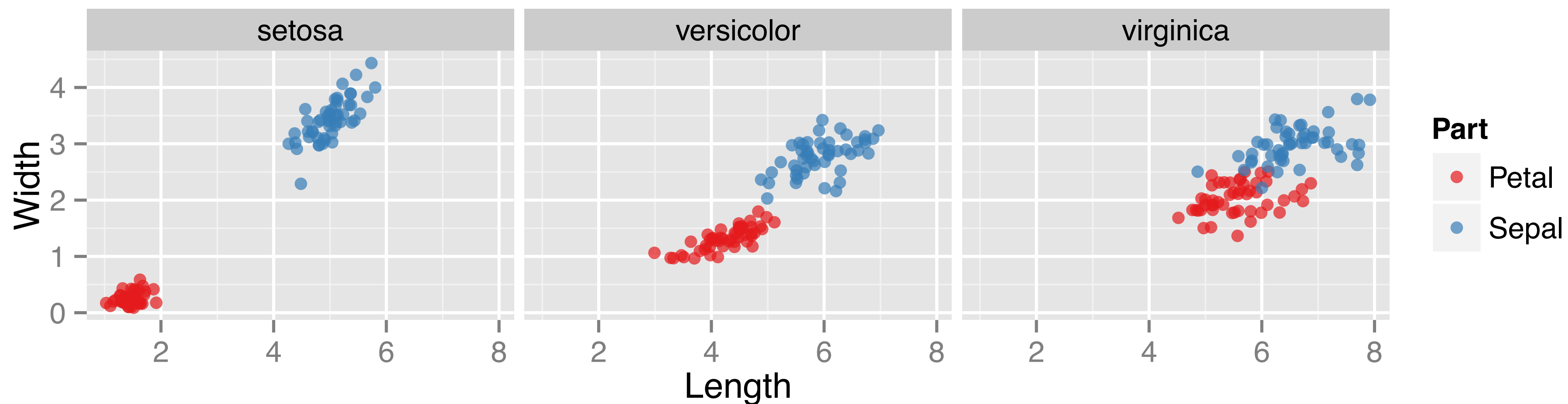
```
> p <- ggplot(iris.wide, aes(x = Length, y = Width, col = Part)) +  
  geom_point(position = position_jitter(), alpha = 0.7) +  
  scale_color_brewer(palette = "Set1") +  
  coord_fixed()  
  
> p
```



iris.wide

Add another variable - primary use of facets
Aid in visual perception

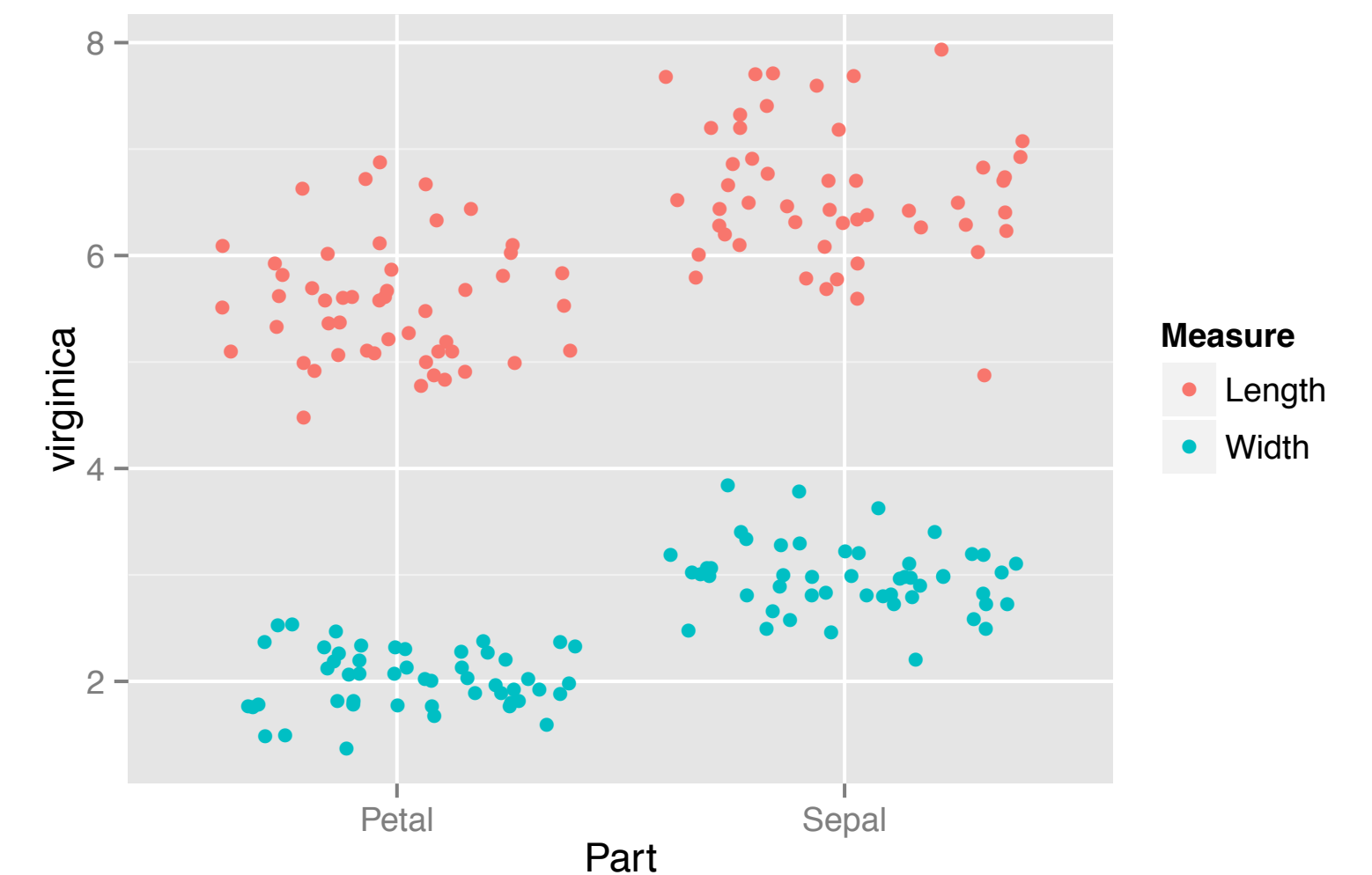
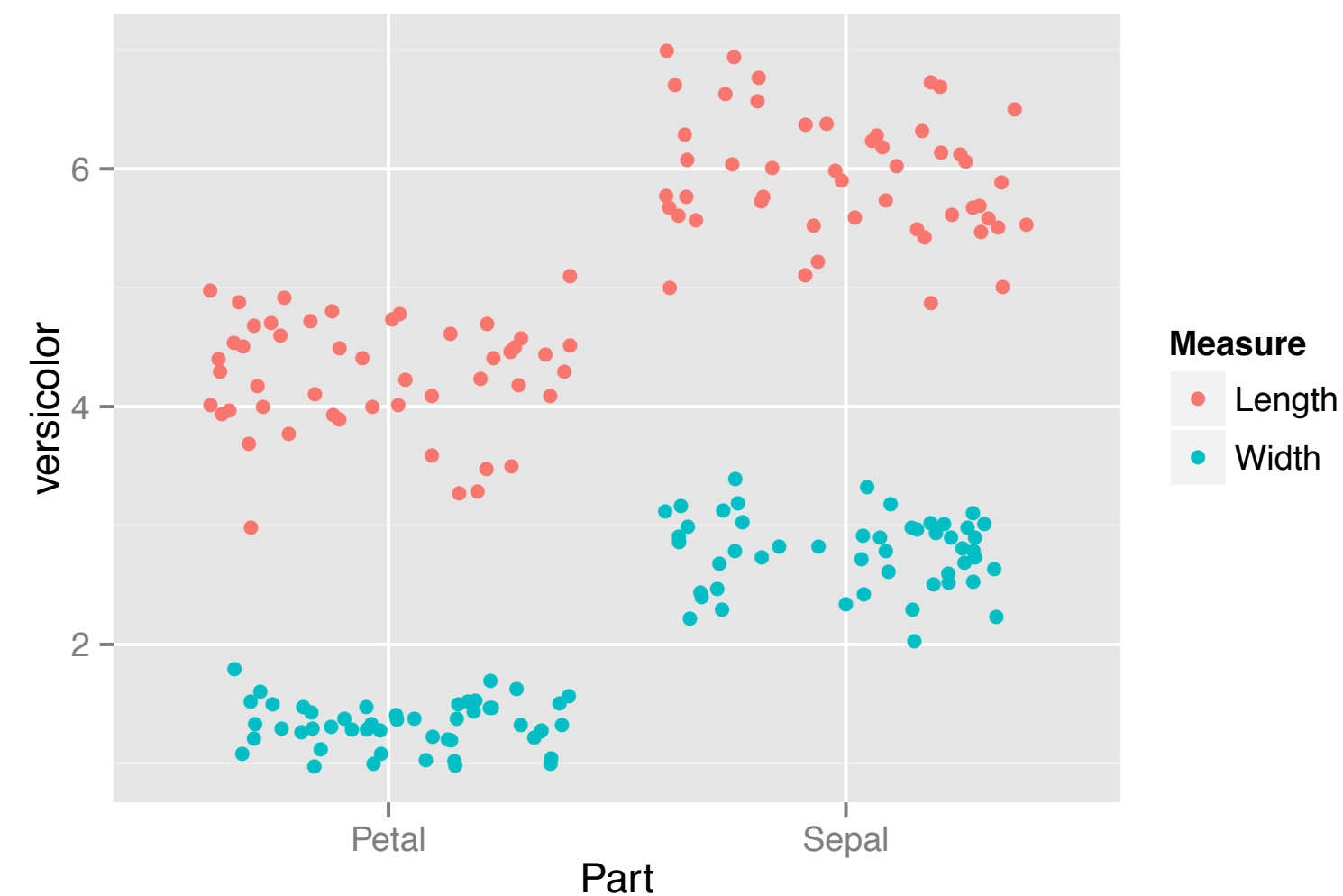
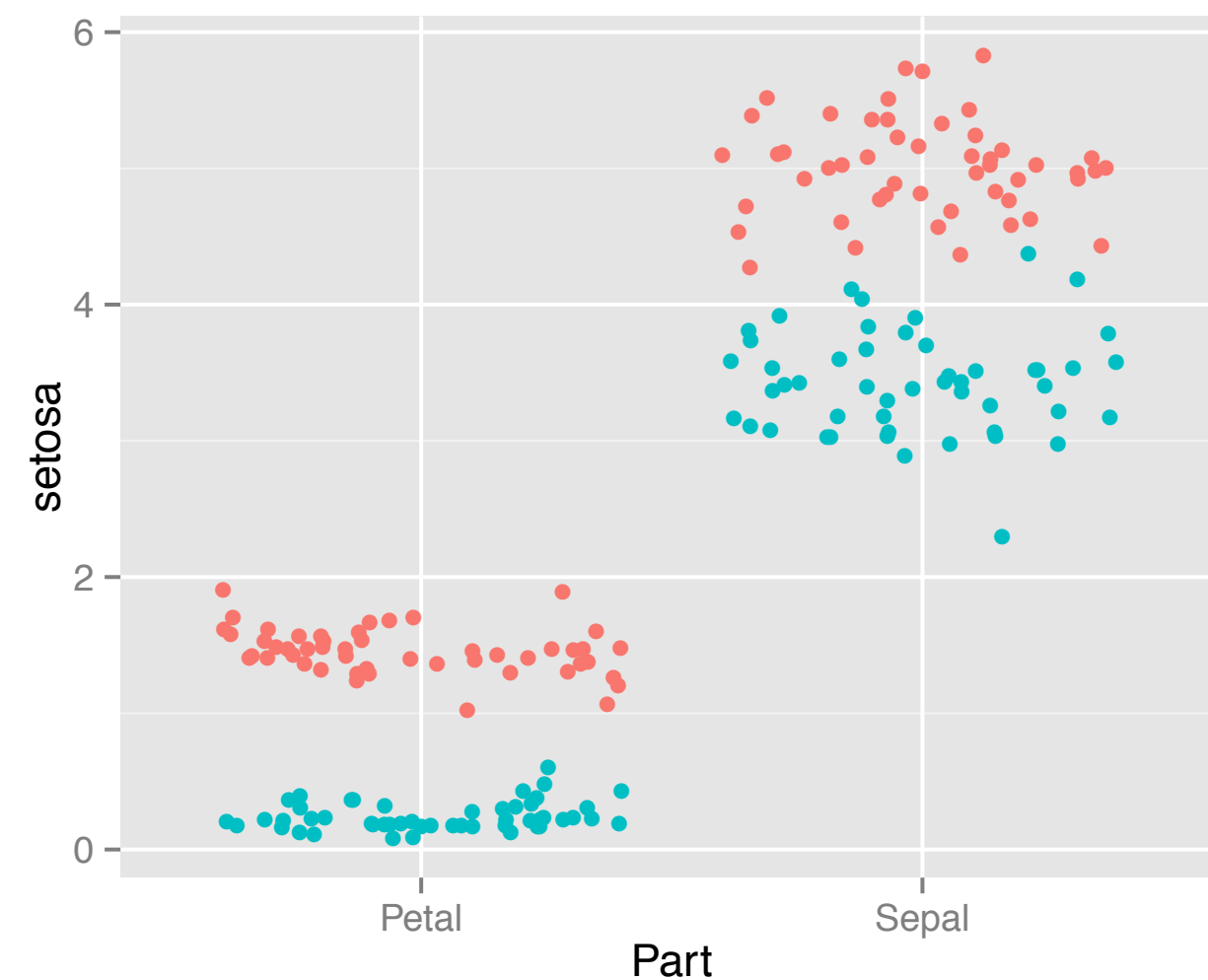
```
> p <- ggplot(iris.wide, aes(x = Length, y = Width, col = Part)) +  
  geom_point(position = position_jitter(), alpha = 0.7) +  
  scale_color_brewer(palette = "Set1") +  
  coord_fixed()  
> p + facet_grid(. ~ Species)      rows ~ columns
```



iris.wide2

Each plot has separate y axis - hard to compare
Three different plot functions - difficult n messy

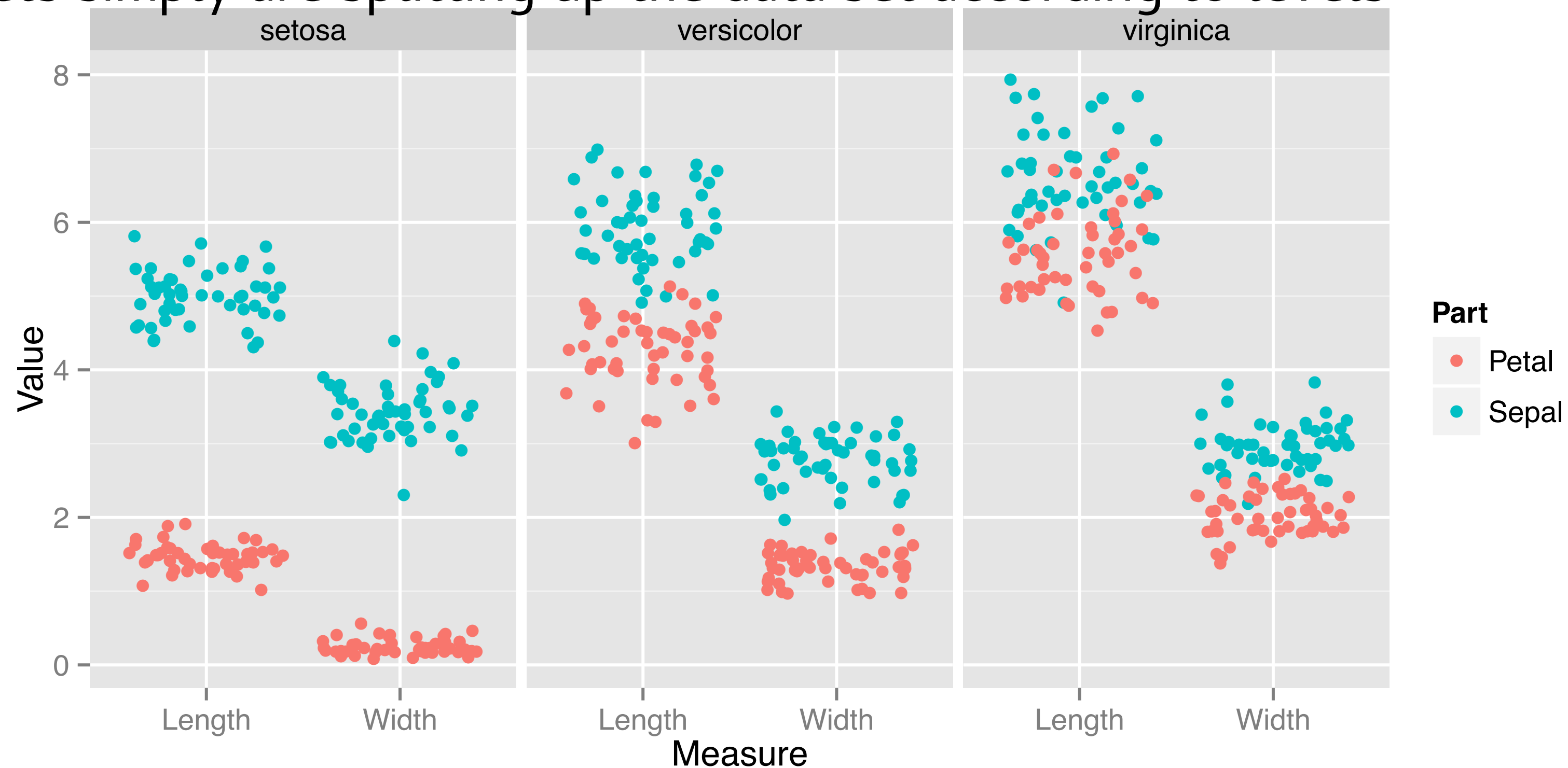
```
> ggplot(iris.wide2, aes(x = Part, y = setosa, col = Measure)) +  
  geom_jitter()  
> ggplot(iris.wide2, aes(x = Part, y = versicolor, col = Measure)) +  
  geom_jitter()  
> ggplot(iris.wide2, aes(x = Part, y = virginica, col = Measure)) +  
  geom_jitter()
```



iris.tidy

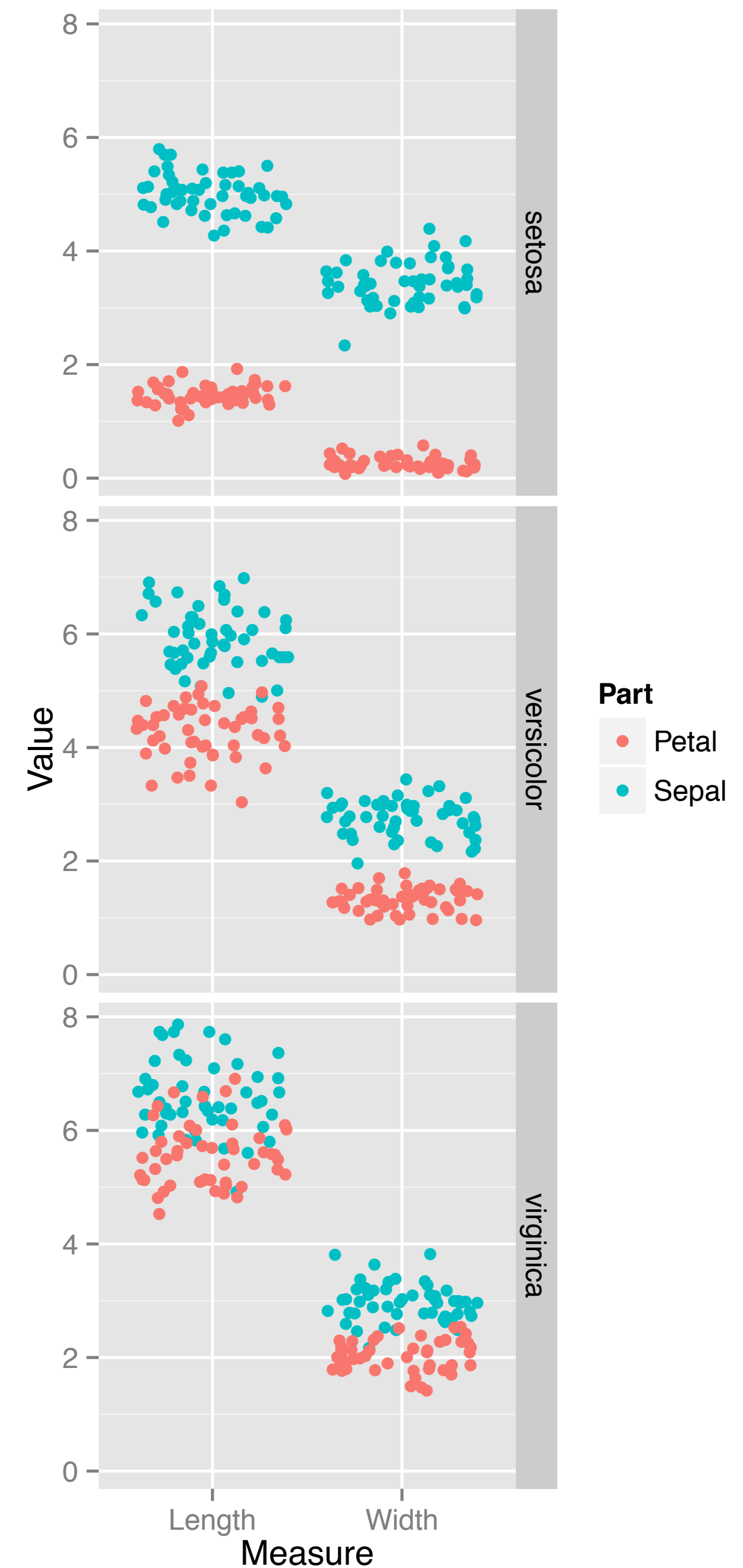
```
> ggplot(iris.tidy, aes(x = Measure, y = Value, col = Part)) +  
  geom_jitter() +  
  facet_grid(. ~ Species)
```

facets simply are splitting up the data set according to levels



iris.tidy - wrong

```
> ggplot(iris.tidy, aes(x = Measure, y = Value, col = Part)) +  
  geom_jitter() +  
  facet_grid(Species ~ .)
```



Other options

- Split according to rows and columns
 - Wrap subplots into columns
- ~ separates rows and columns
. stands for no variable for either row or column position