DATA VISUALIZATION WITH GGPLOT2

Objects & Layers

Data Layer

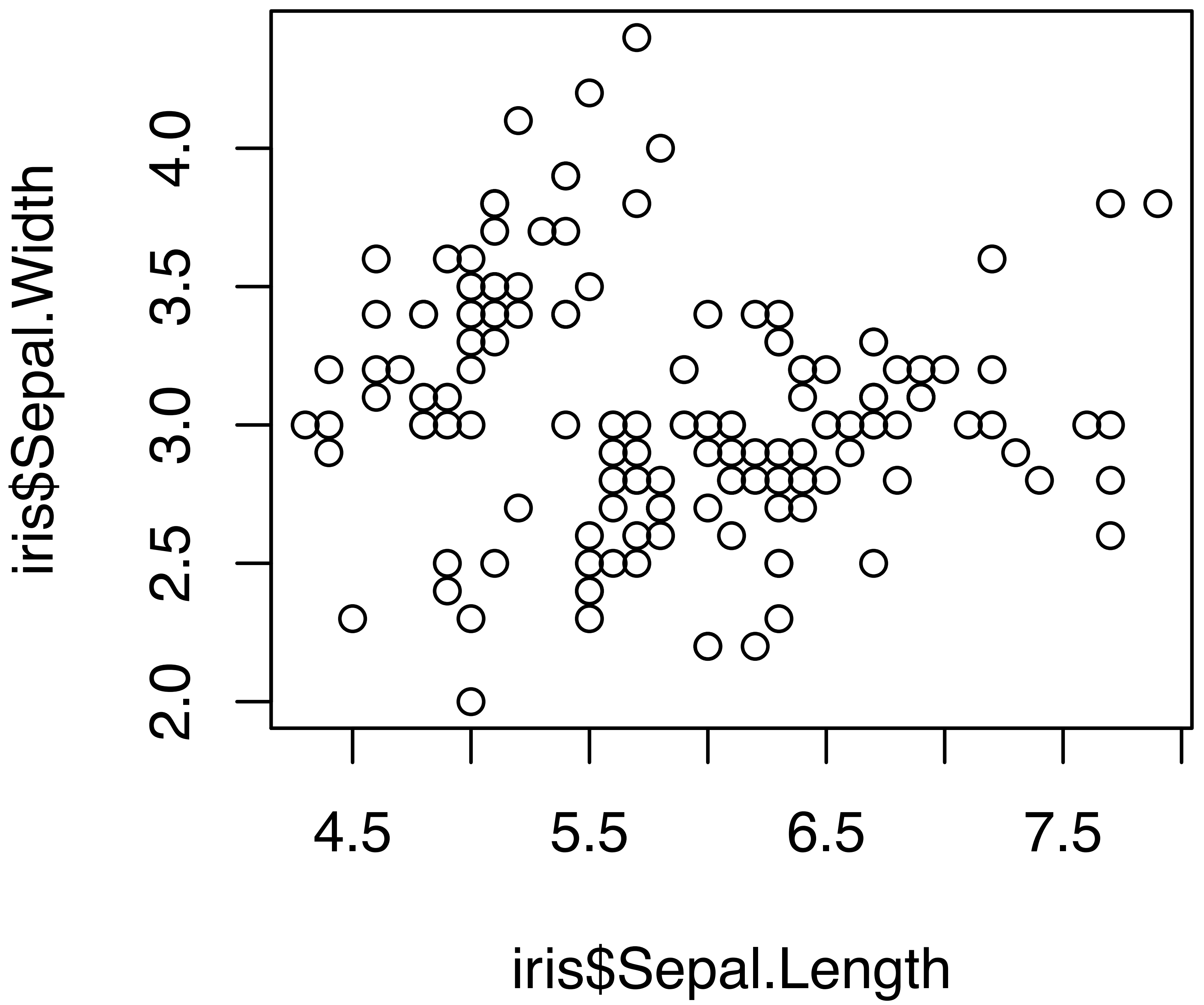
- How data structure influences plots
- How we understand data
- ggplot2 versus base

iris

```
> iris <- iris[c(5, 1:4)]
> str(iris)
'data.frame':150 obs. of  5 variables:
 $ Species     : Factor w/ 3 levels "Setosa", ..: 1 1 1 1 1 1 ...
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

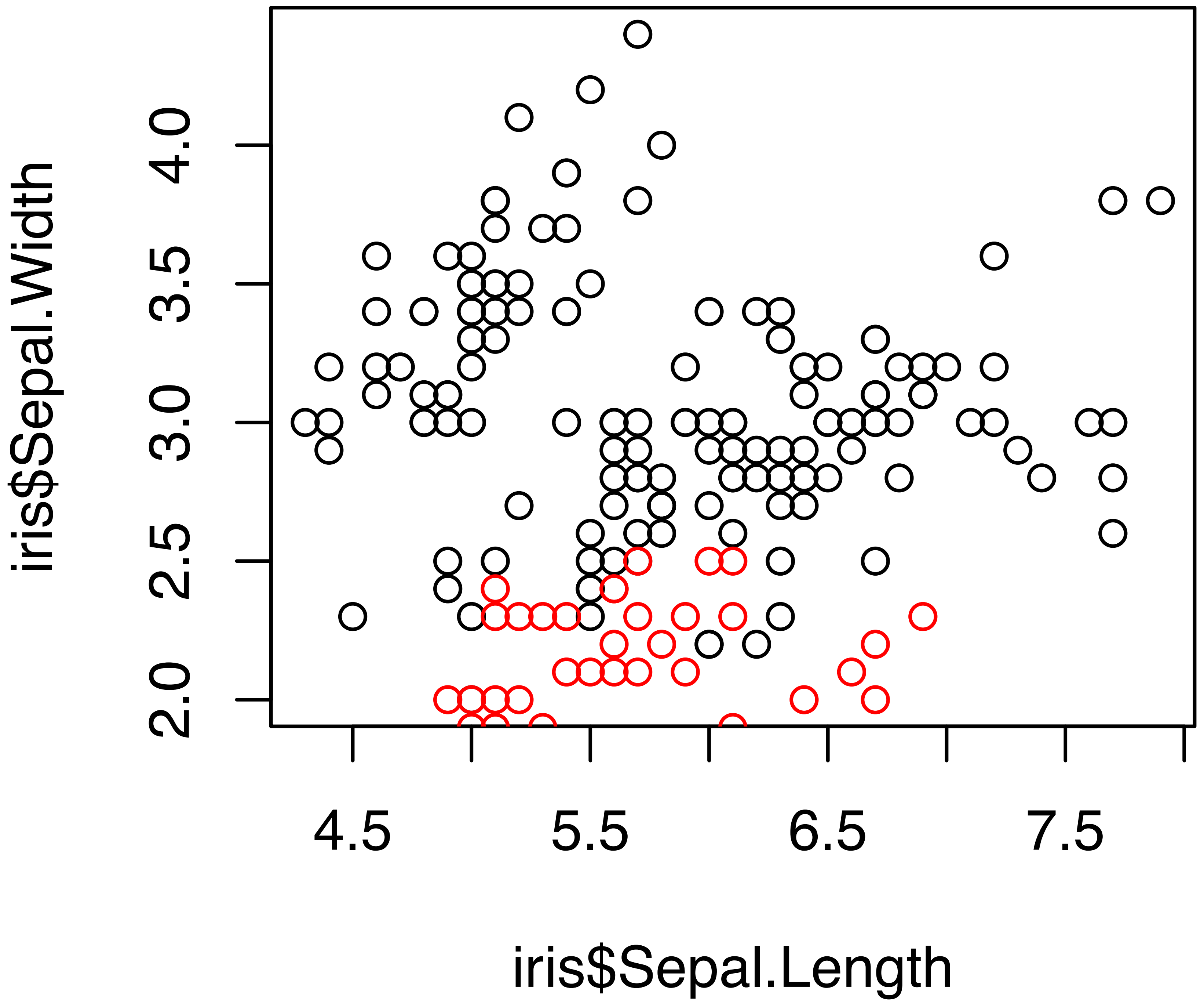
Base plot

```
> plot(iris$Sepal.Length, iris$Sepal.Width)
```



Add petal width and length

```
> plot(iris$Sepal.Length, iris$Sepal.Width)
> points(iris$Petal.Length, iris$Petal.Width, col = "red")
```

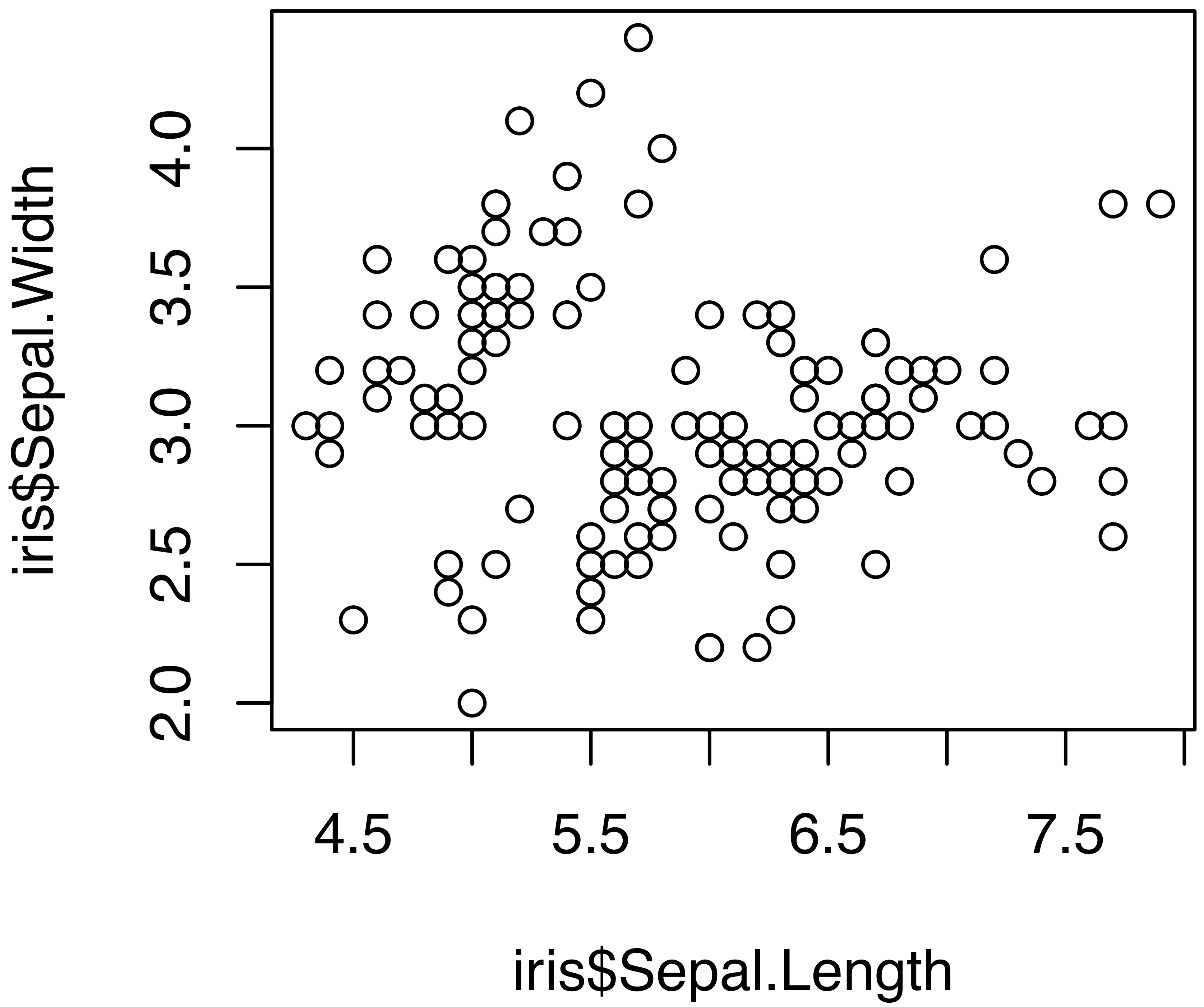


Limitations

1. Plot doesn't get redrawn
2. Plot is drawn as an image
3. Need to manually add legend
4. No unified framework for plotting

Default: point plot

```
> plot(iris$Sepal.Length, iris$Sepal.Width, type = "p")
```



ggplot2

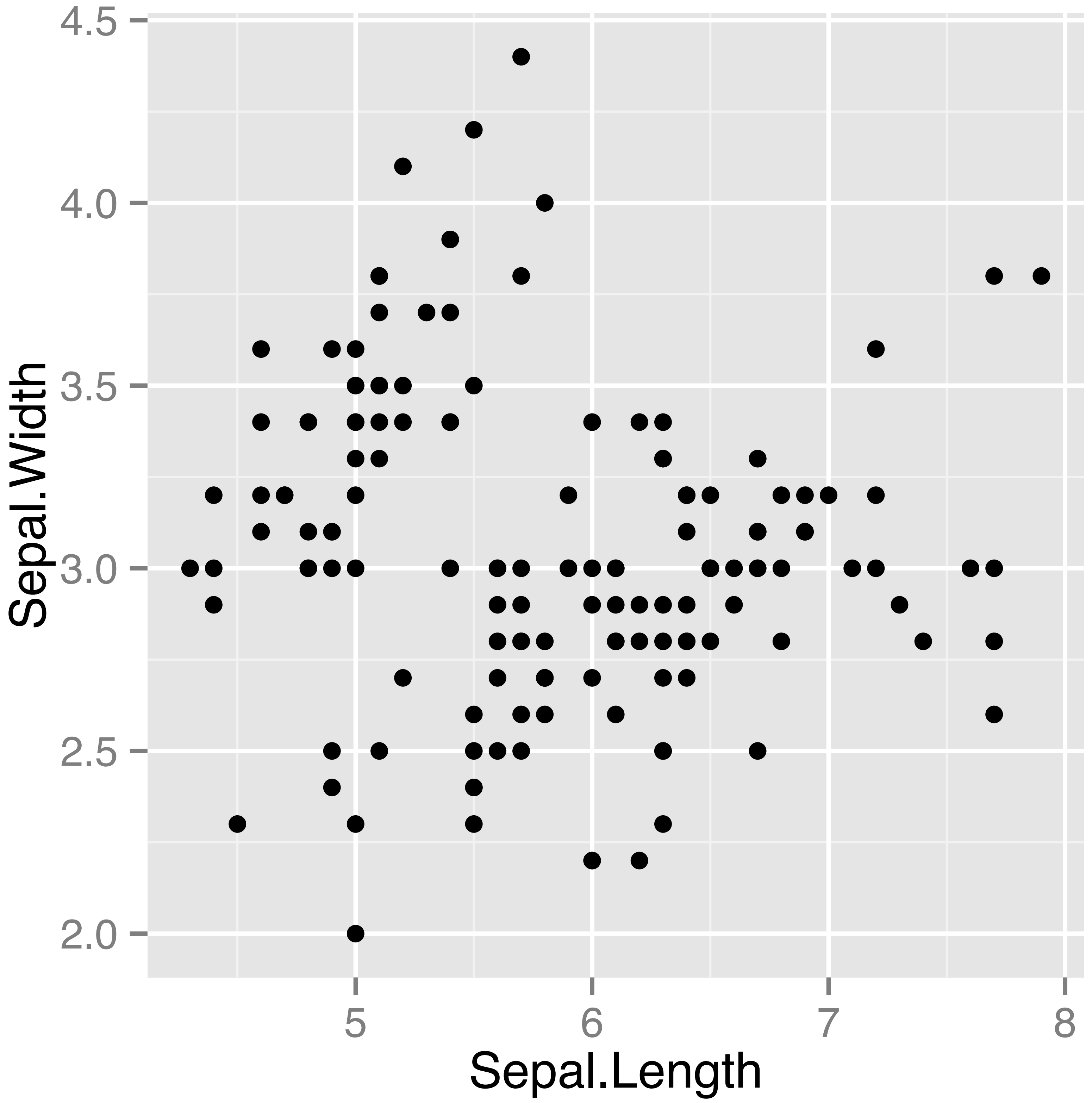
Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
	X	Y		

data

aesthetics

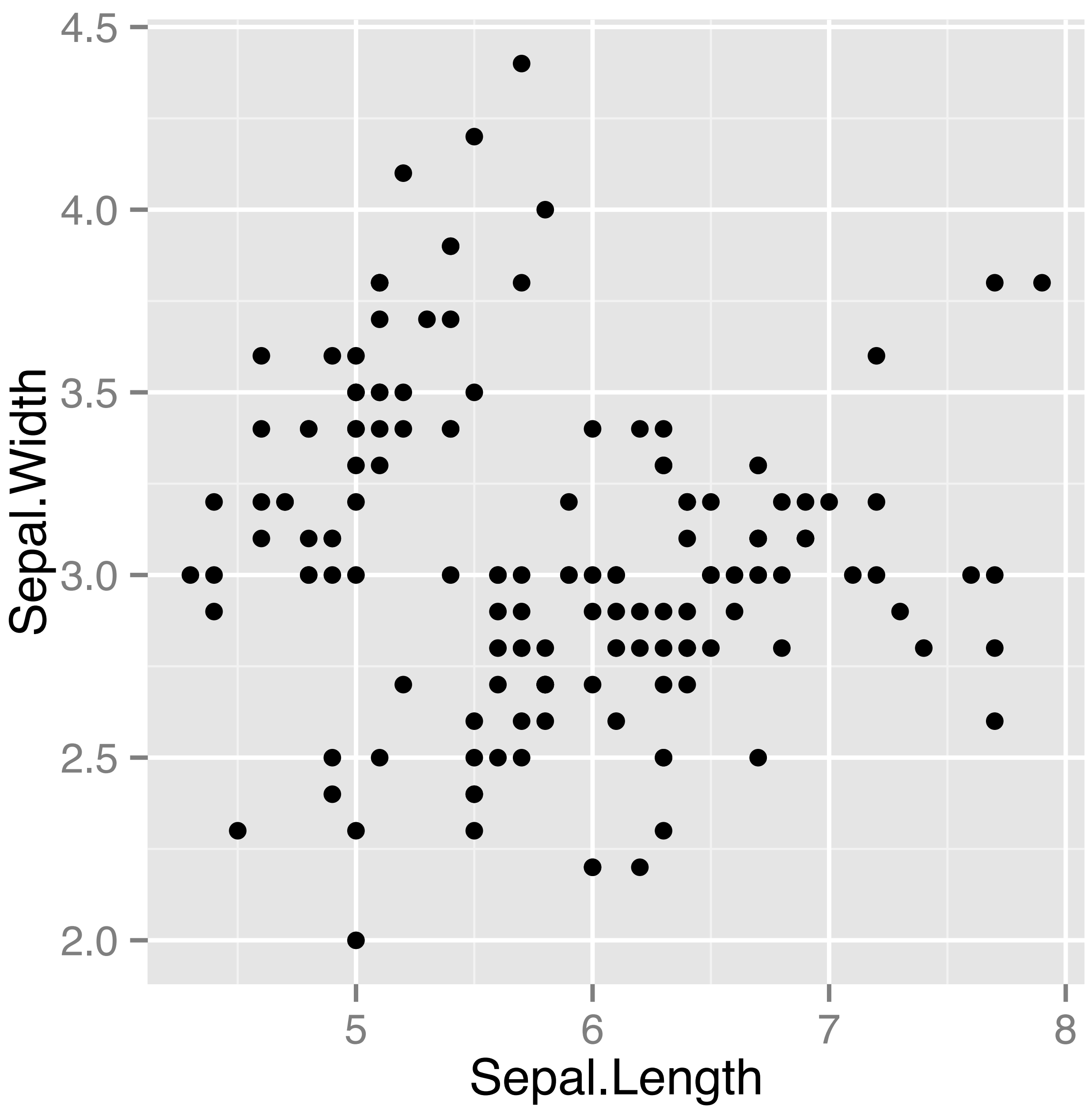
```
> ggplot(iris, aes(x = Sepal.Length,
                   y = Sepal.Width)) +
    geom_point()
```

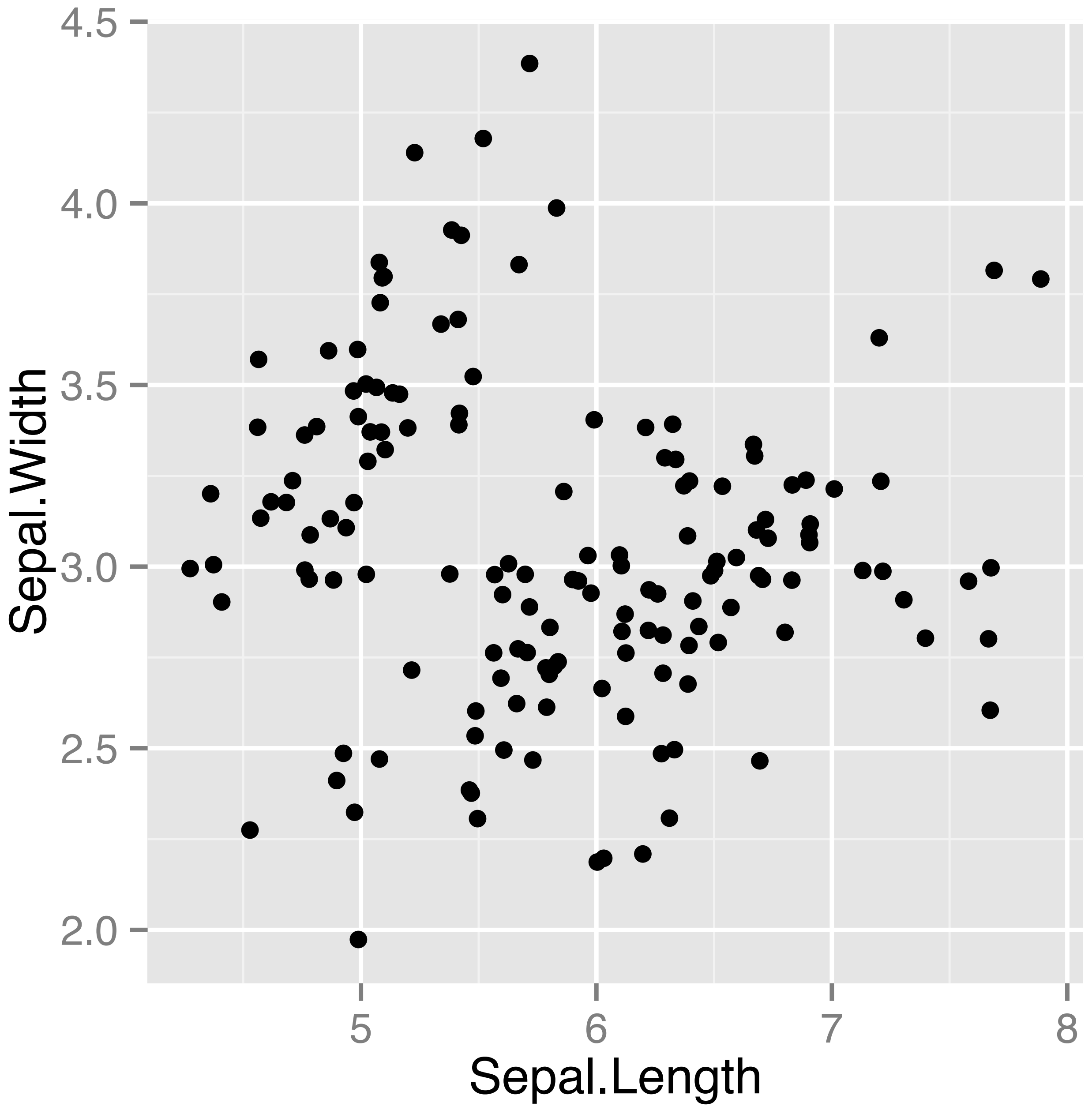
geometry

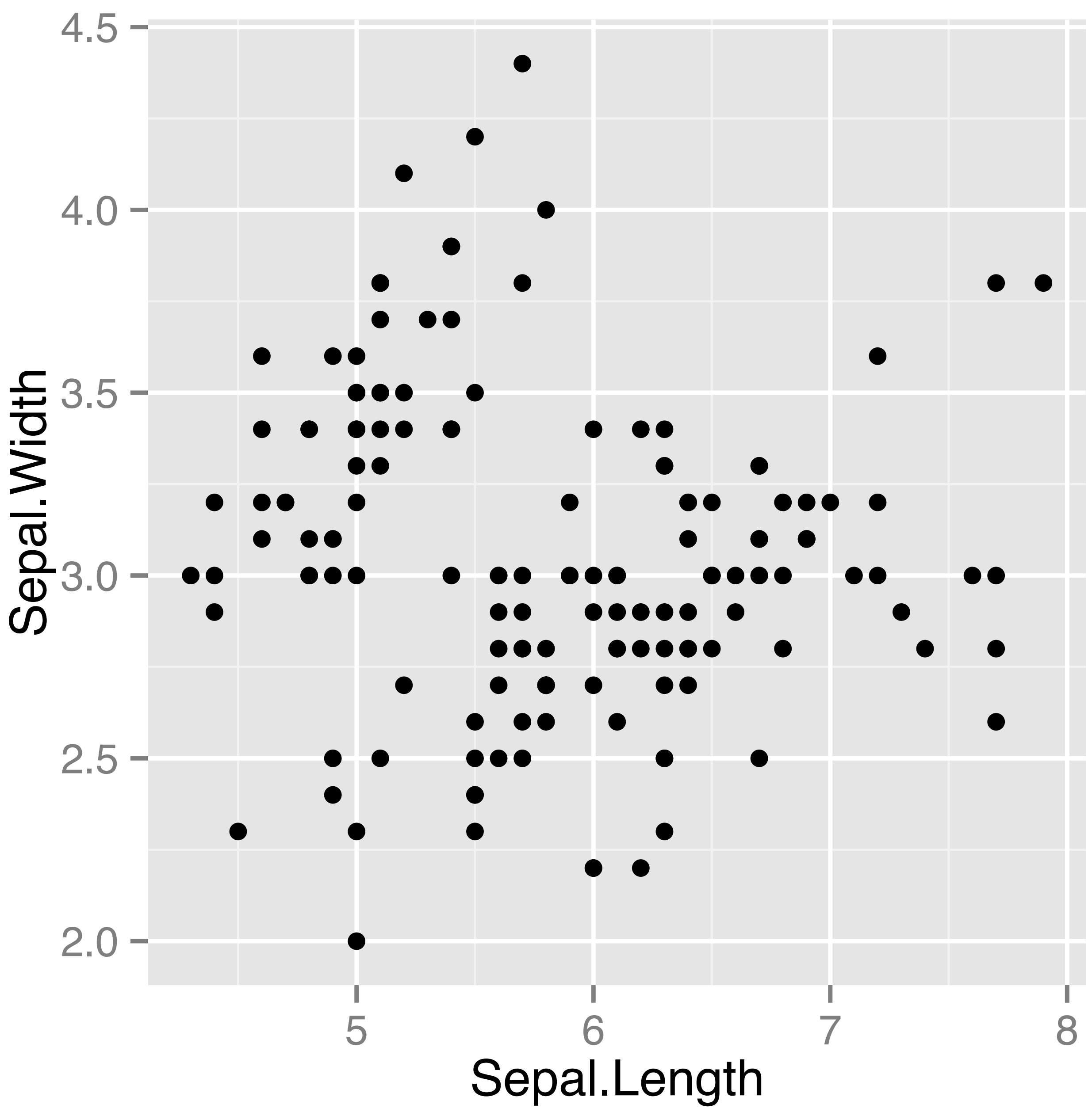


ggplot object

```
> p <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width))
> p + geom_point()
> p + geom_jitter()
> p <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
         geom_point()
> p
```







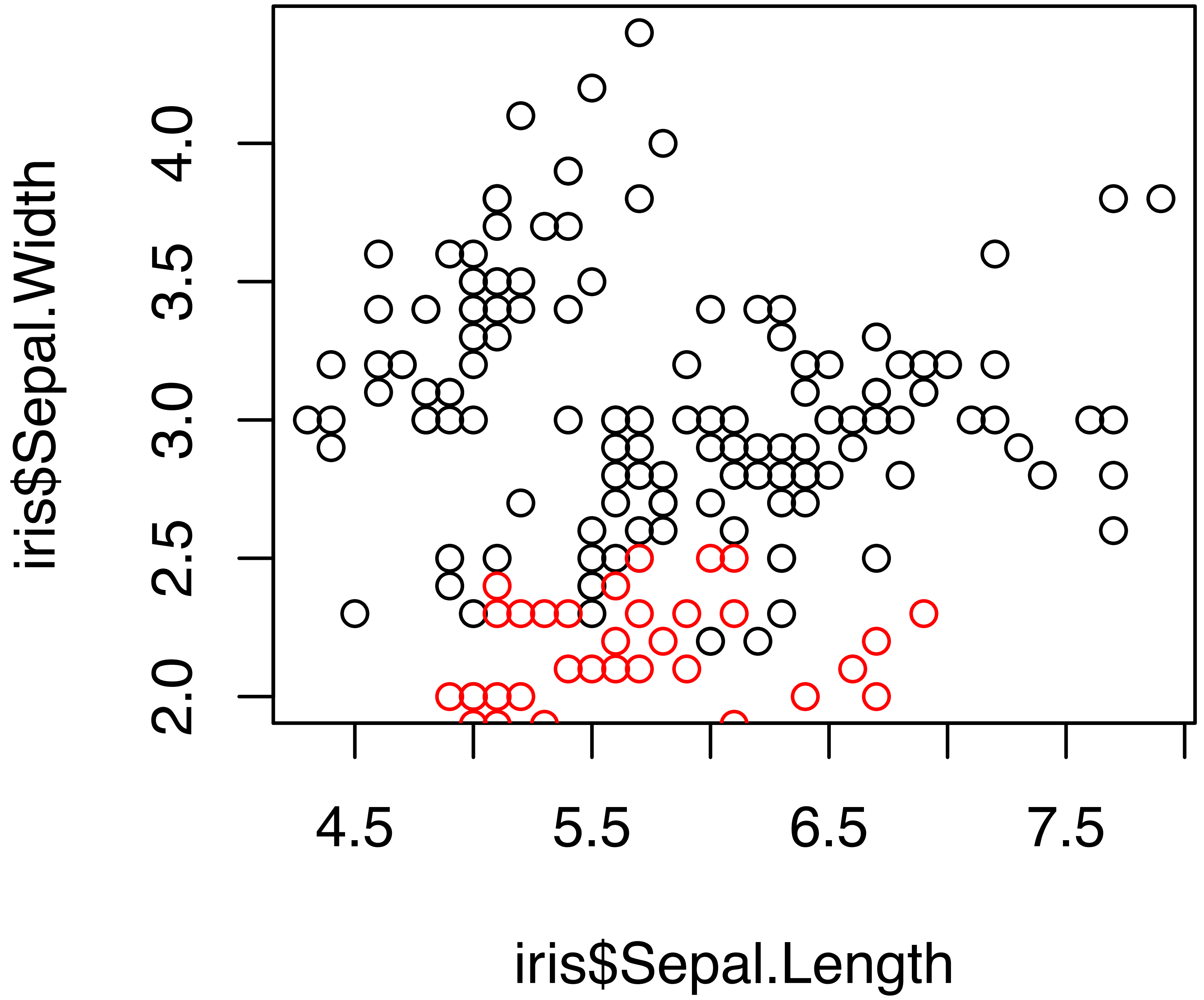
Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

Proper Data Format

Base

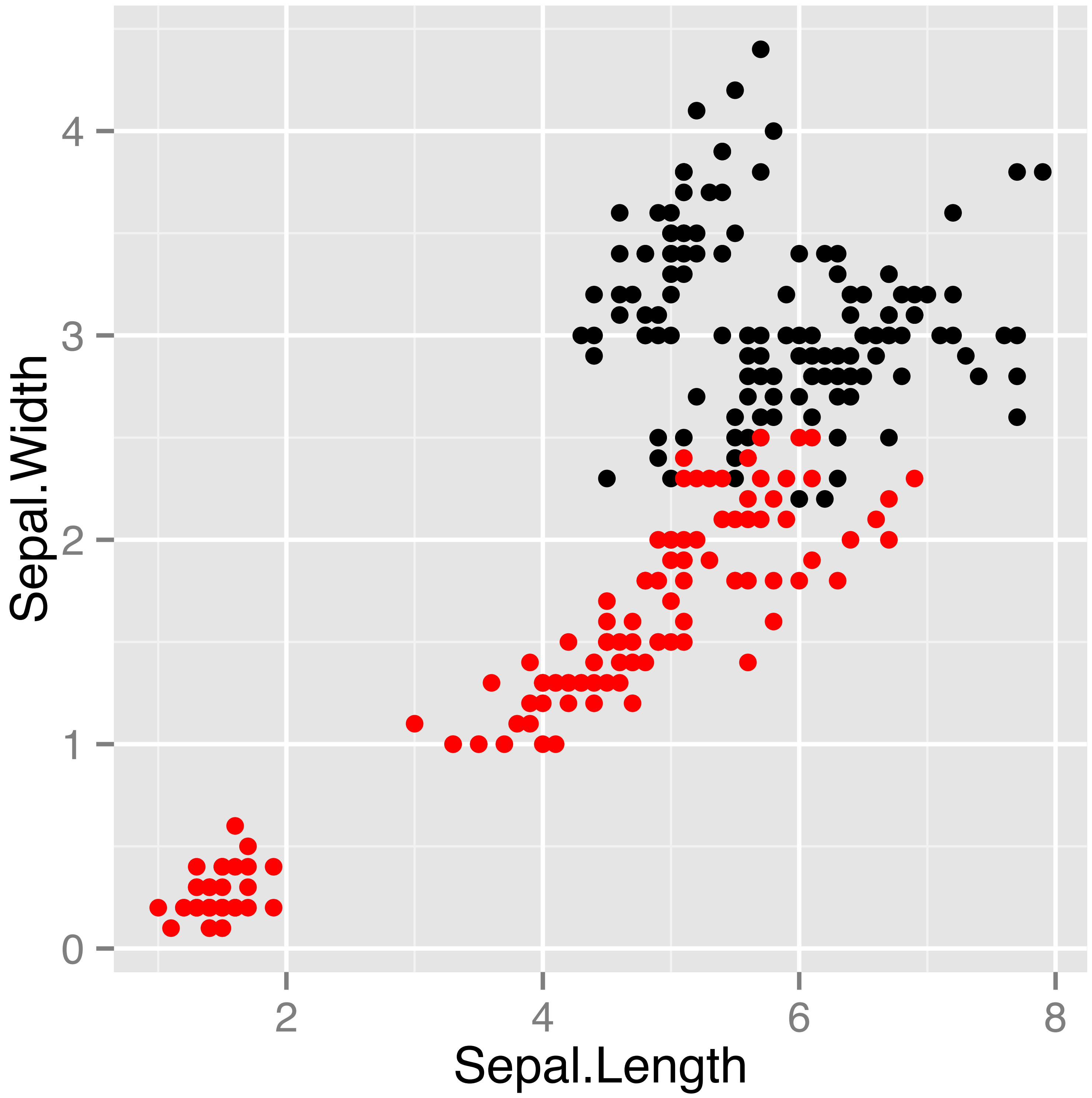
```
> plot(iris$Sepal.Length, iris$Sepal.Width)
> points(iris$Petal.Length, iris$Petal.Width, col = "red")
```



ggplot2

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
    geom_point() +
    geom_point(aes(x = Petal.Length, y = Petal.Width), col = "red")
```

1. **Plotting space is adjusted**
2. **ggplot2 produces an object**

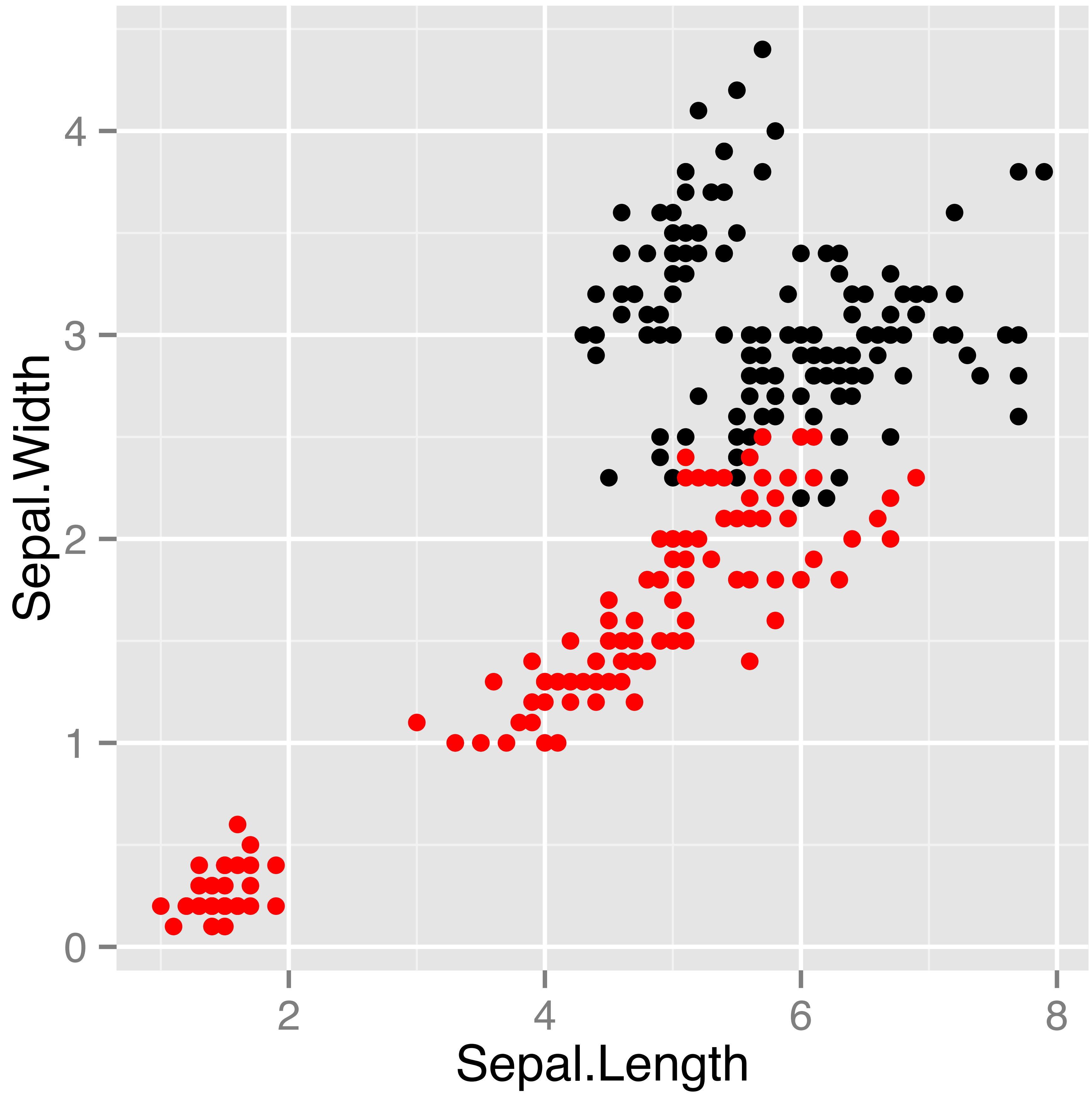


iris

```
> str(iris)
'data.frame':150 obs. of  5 variables:
 $ Species     : Factor w/ 3 levels "Setosa", ..: 1 1 1 1 1 1 ...
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

iris

Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
	X	Y	X	Y



width

length

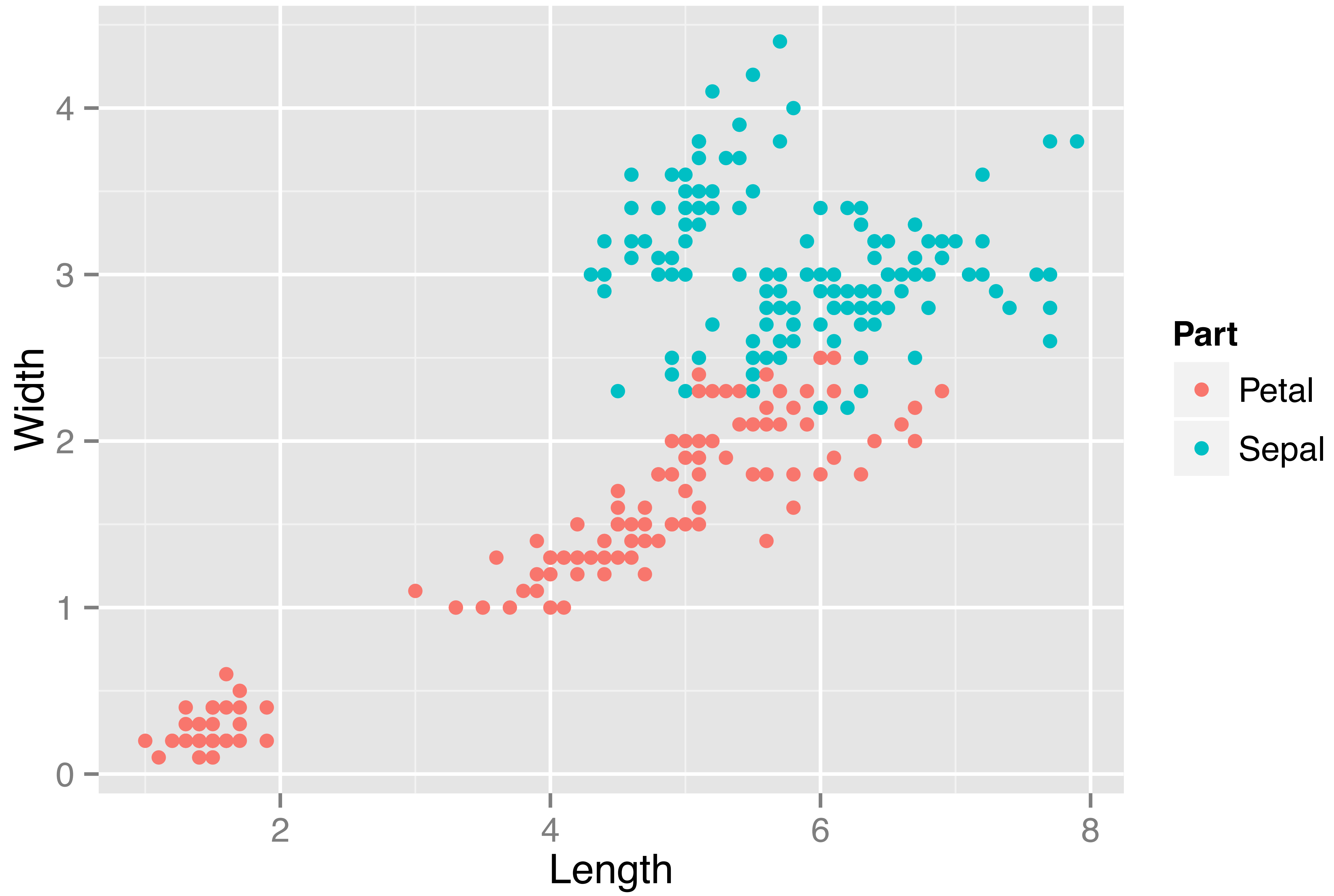
iris.wide

```
> str(iris.wide)
'data.frame':300 obs. of  4 variables:
 $ Species: Factor w/ 3 levels "Setosa",..: 1 1 1 1 1 1 1 1 1 ...
 $ Part   : chr  "Petal" "Petal" "Petal" "Petal" ...
 $ Length : num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Width  : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

iris.wide

Species	Part	Length	Width
	Colour	X	Y

```
> ggplot(iris.wide, aes(x = Length, y = Width, col = Part)) +
    geom_point()
```



Wide format

```
> head(iris)
  Species Sepal.Length Sepal.Width Petal.Length Petal.Width
1  Setosa          5.1         3.5          1.4         0.2
2  Setosa          4.9         3.0          1.4         0.2
3  Setosa          4.7         3.2          1.3         0.2
4  Setosa          4.6         3.1          1.5         0.2
5  Setosa          5.0         3.6          1.4         0.2
6  Setosa          5.4         3.9          1.7         0.4

> head(iris.wide)
  Species  Part Length Width
1  Setosa Petal    1.4   0.2
2  Setosa Petal    1.4   0.2
3  Setosa Petal    1.3   0.2
4  Setosa Petal    1.5   0.2
5  Setosa Petal    1.4   0.2
6  Setosa Petal    1.7   0.4
```

Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

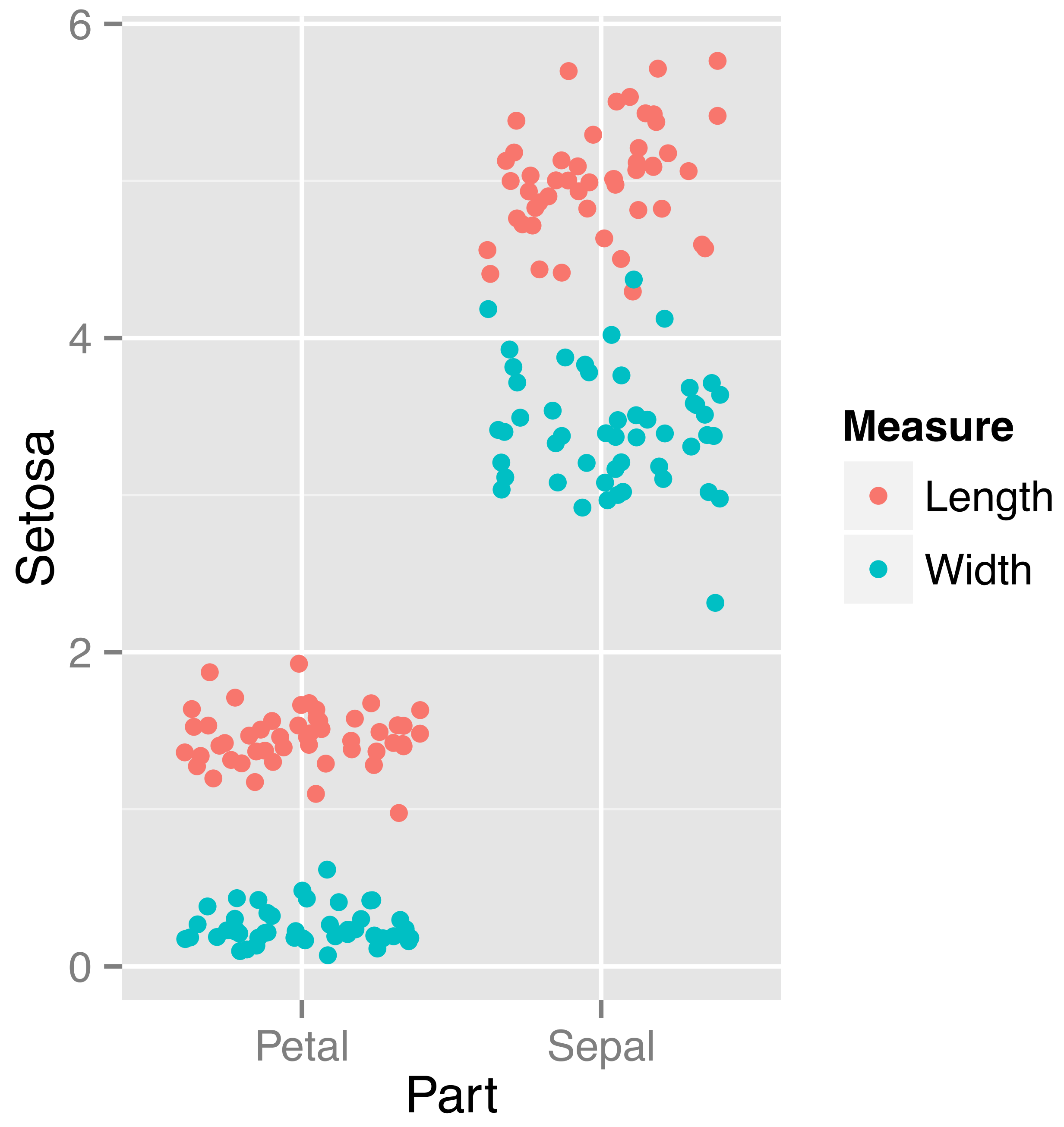
Tidy data

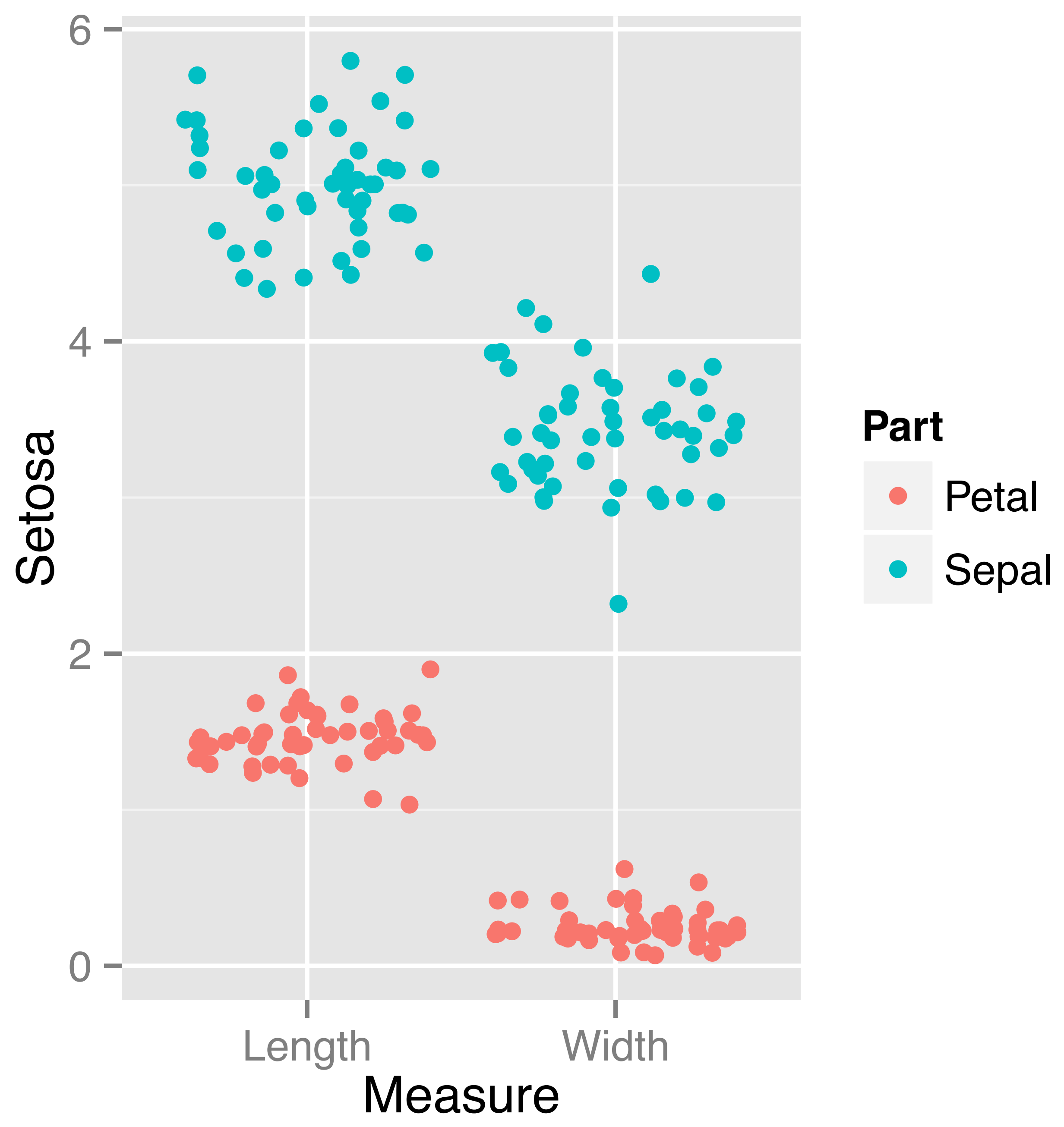
iris.wide2

```
> str(iris.wide2)
'data.frame':200 obs. of  5 variables:
 $ Measure   : chr  "Length" "Length" "Length" "Length" ...
 $ Part      : chr  "Petal" "Petal" "Petal" "Petal" ...
 $ Setosa    : num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Versicolor: num  4.7 4.5 4.9 4 4.6 4.5 4.7 3.3 4.6 3.9 ...
 $ Virginica : num  6 5.1 5.9 5.6 5.8 6.6 4.5 6.3 5.8 6.1 ...
```

iris.wide2

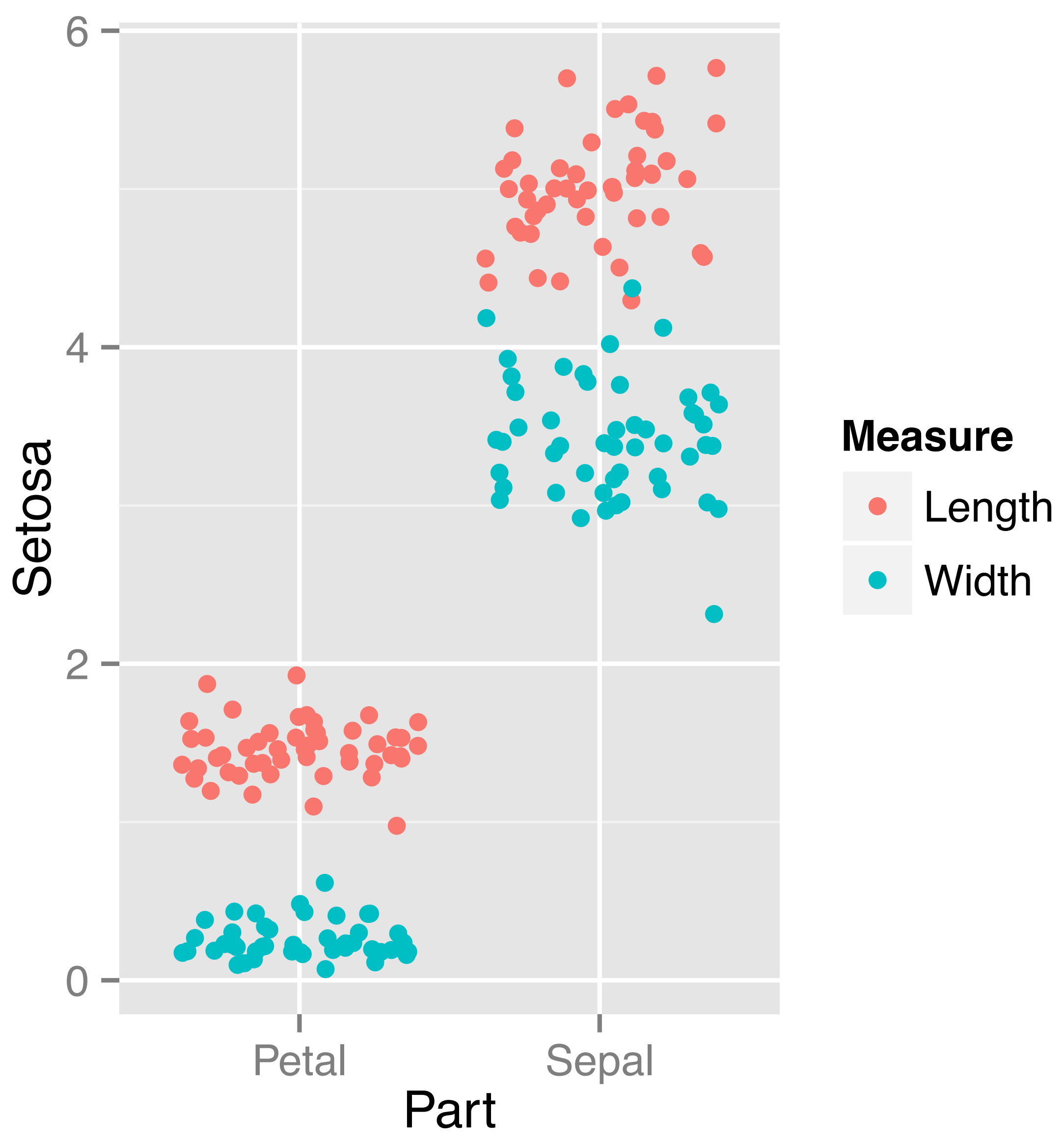
```
> ggplot(iris.wide2, aes(x = Part, y = Setosa, col = Measure)) +
    geom_jitter()

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

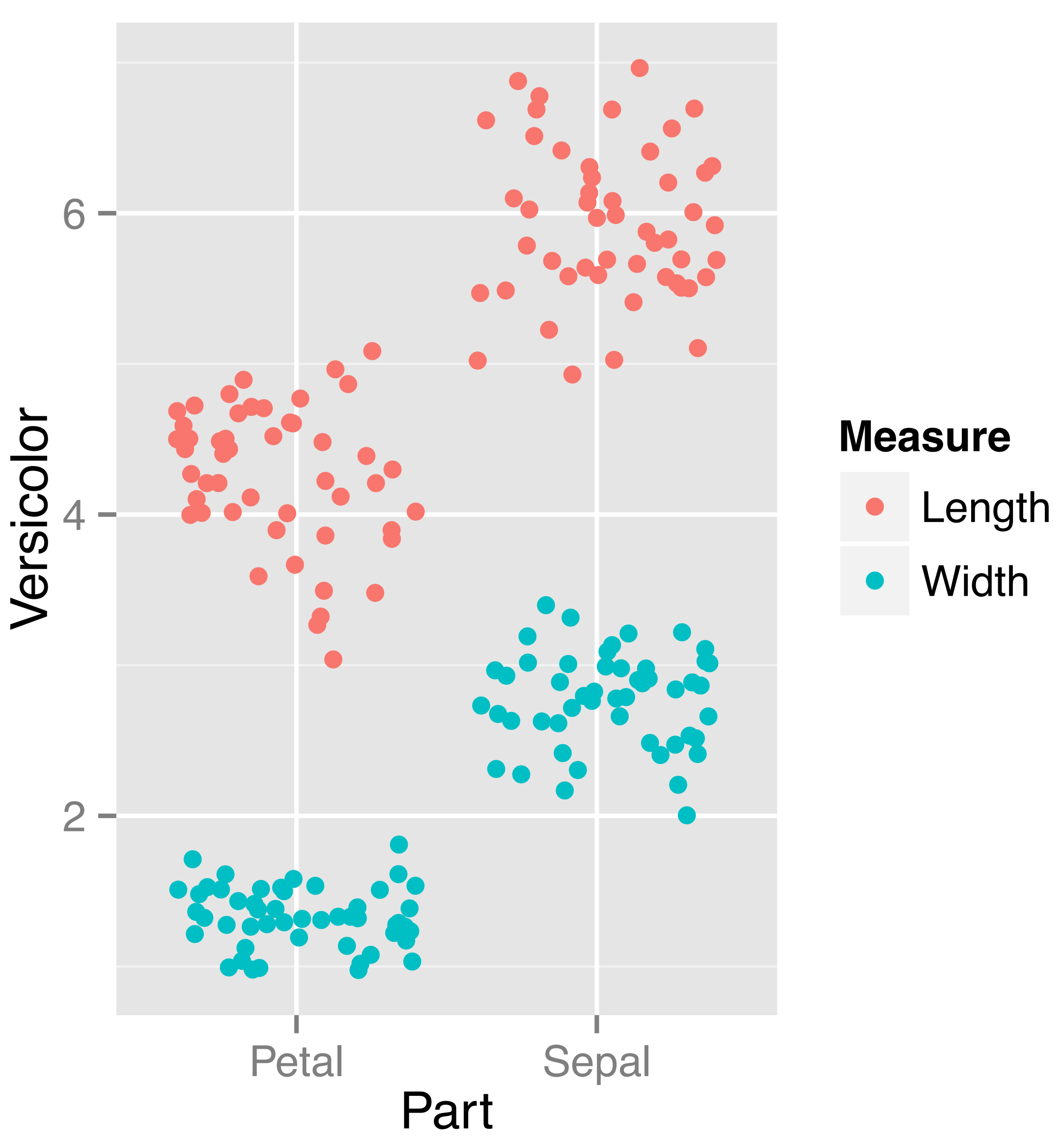


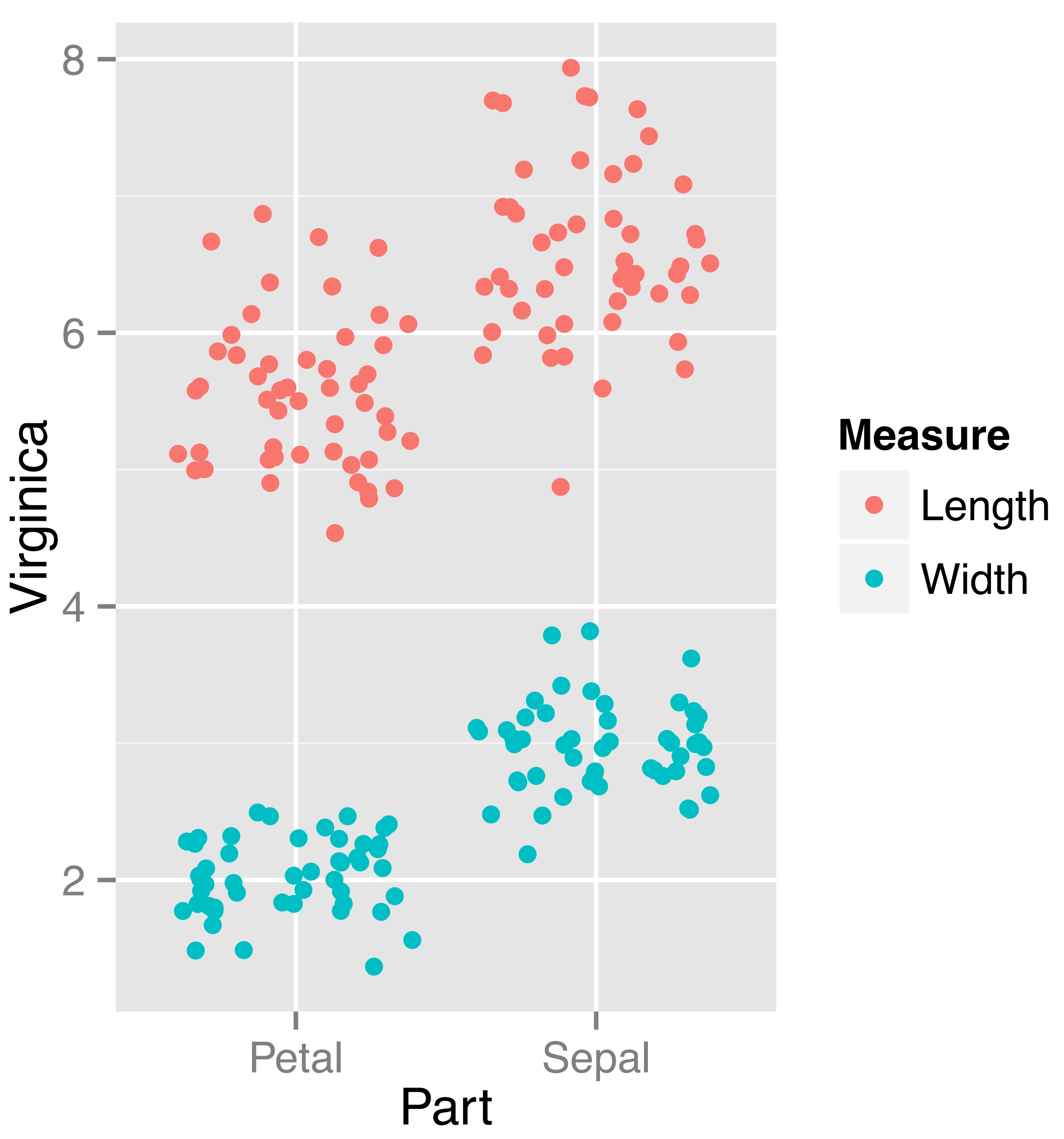


All three classes

```
> 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()
```





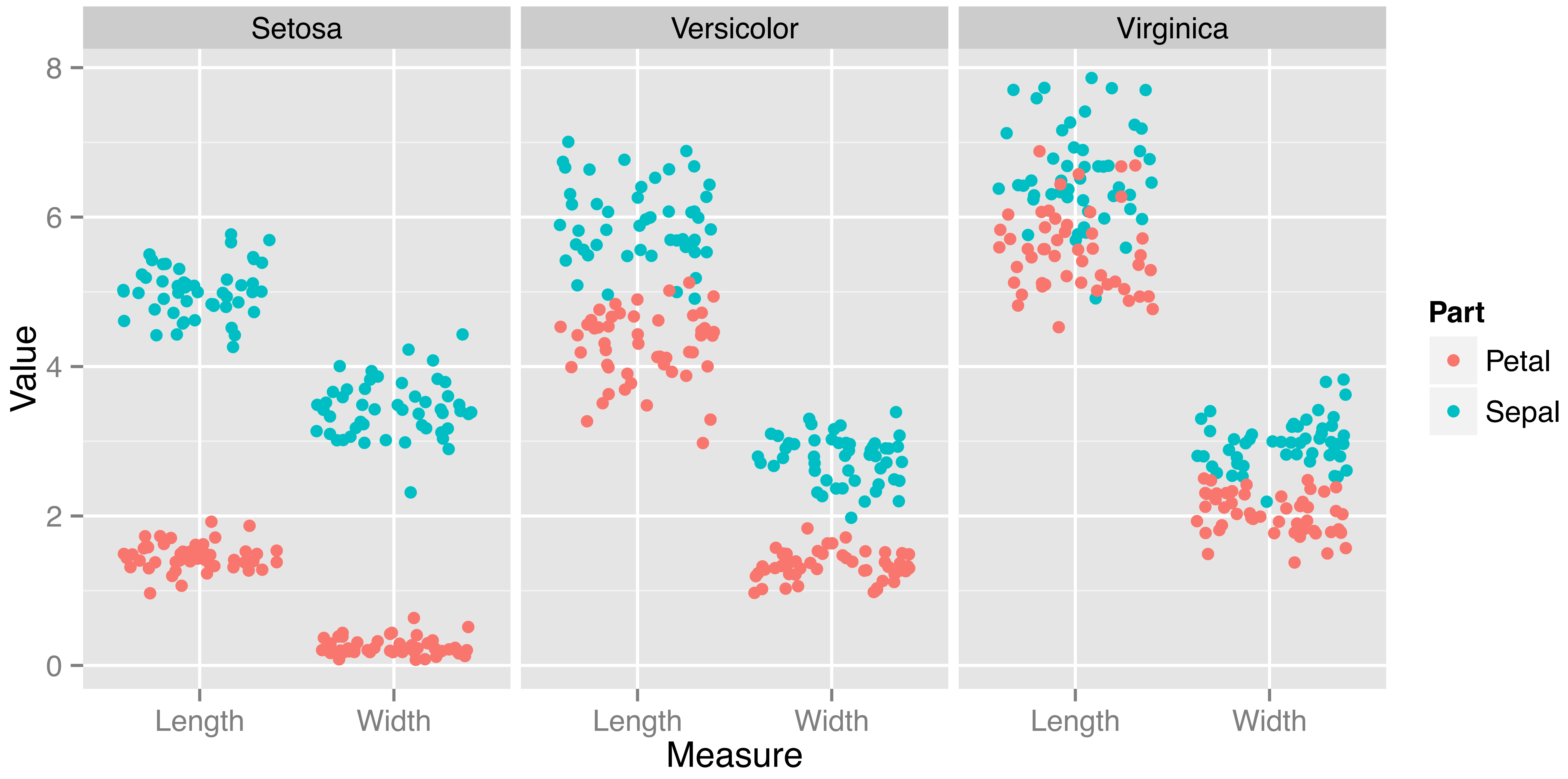


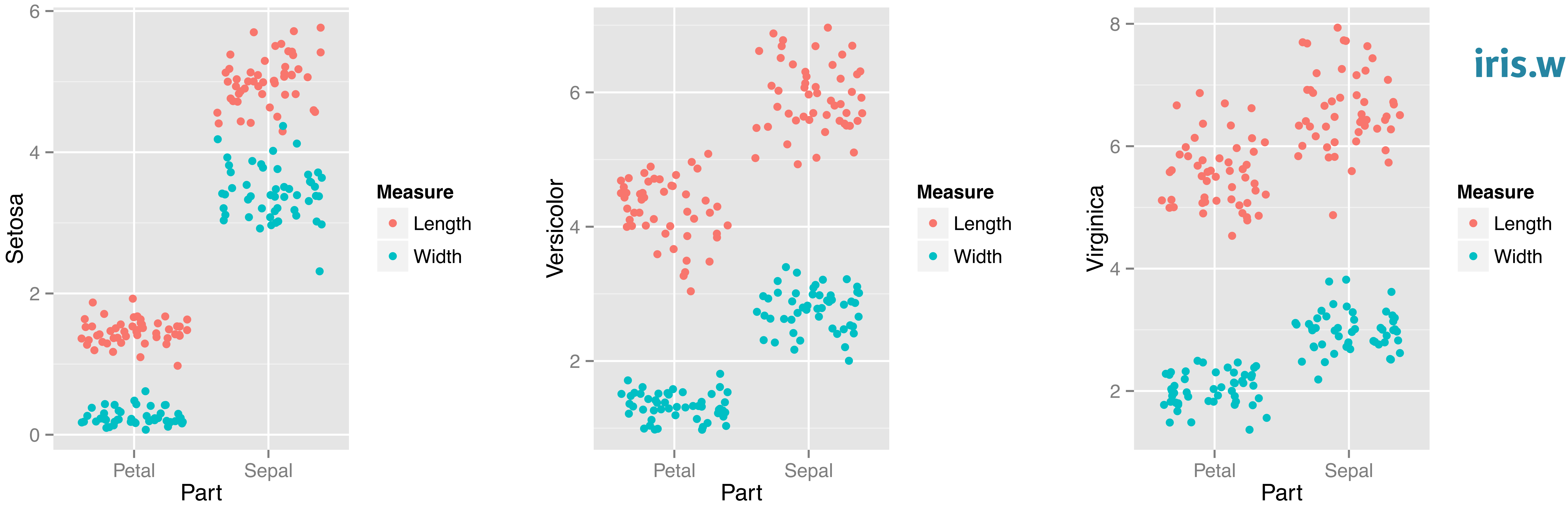
iris.tidy

```
> str(iris.tidy)
'data.frame':600 obs. of  4 variables:
 $ Species: Factor w/ 3 levels "Setosa",..: 1 1 1 1 1 1 1 1 1 ...
 $ Part   : chr  "Sepal" "Sepal" "Sepal" "Sepal" ...
 $ Measure: chr  "Length" "Length" "Length" "Length" ...
 $ Value  : num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
```

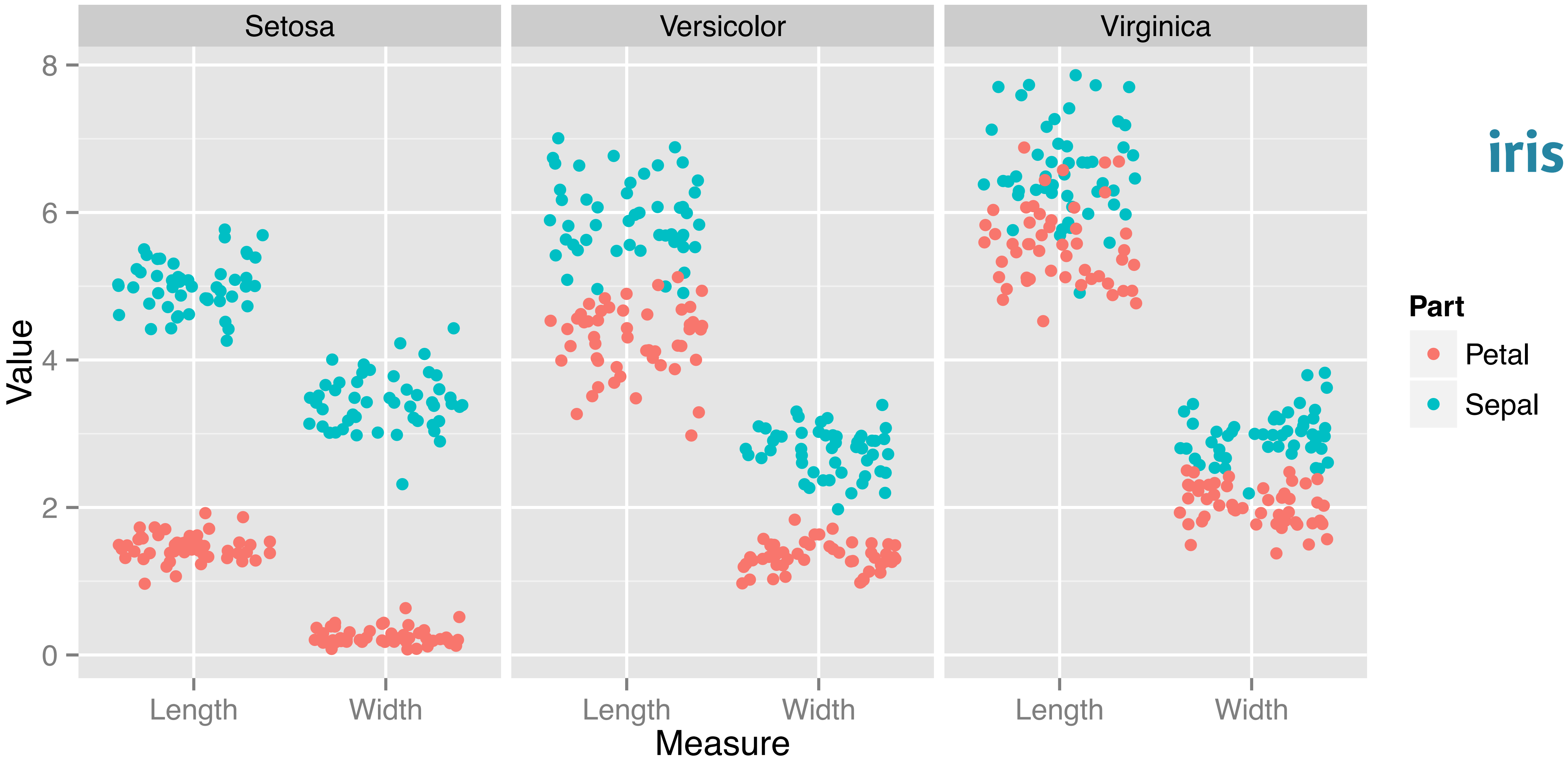
iris.tidy

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





iris.wide2



iris.tidy

Let’s practice!