DATA VISUALIZATION WITH GGPLOT2

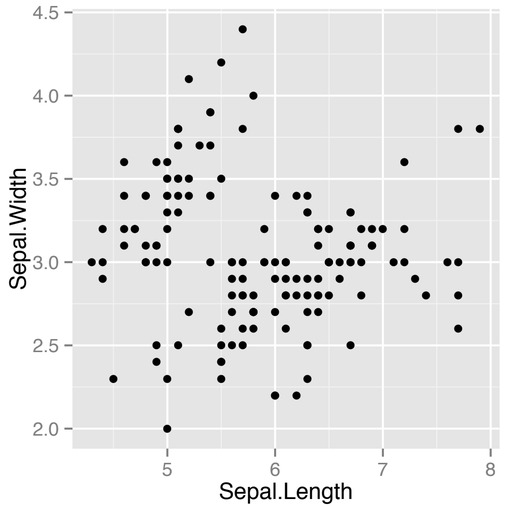
qplot

Up to now

- `ggplot()`
  - Base data layer
  - Aesthetics
  - Add geom layers
- Easy, quick & dirty: `qplot()`

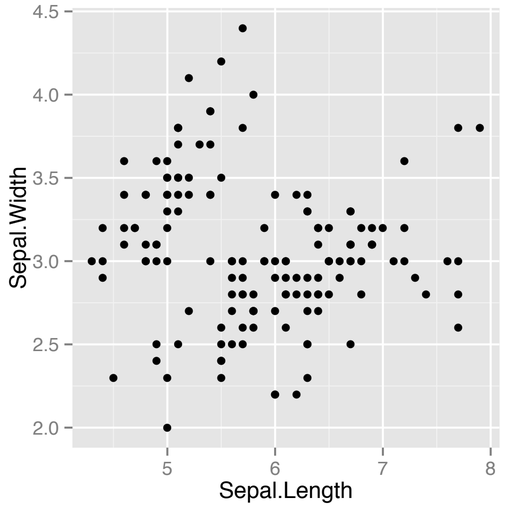
ggplot()

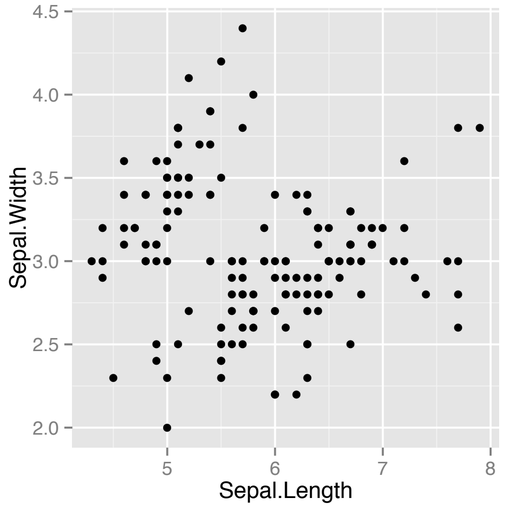
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
    geom_point()
```



qplot()

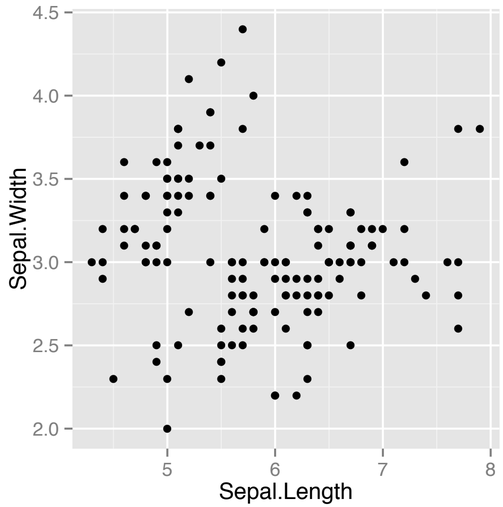
```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
    geom_point()

> qplot(Sepal.Length, Sepal.Width, data = iris)
```

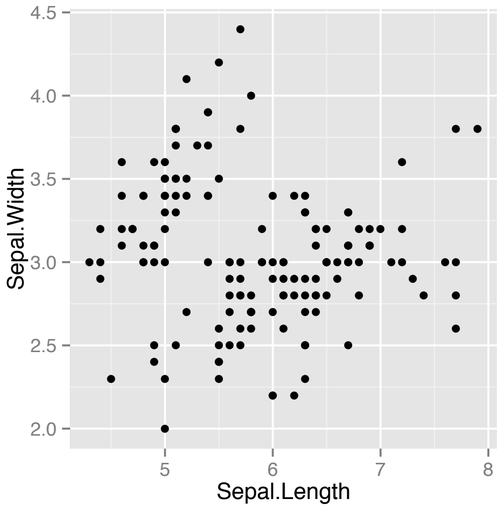


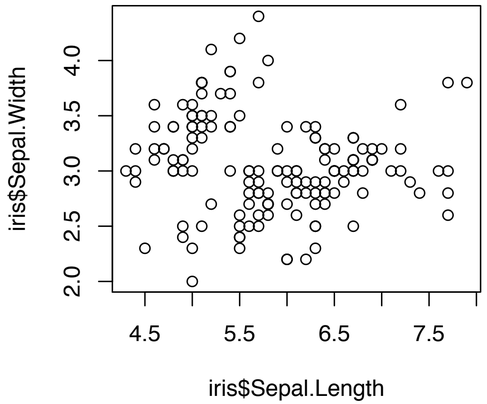


base

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

> qplot(Sepal.Length, Sepal.Width, data = iris)

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

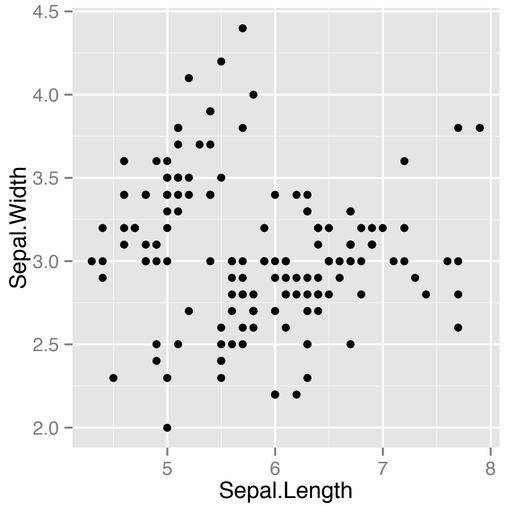


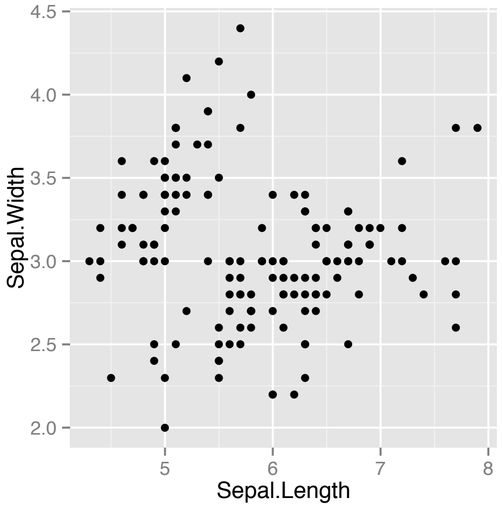


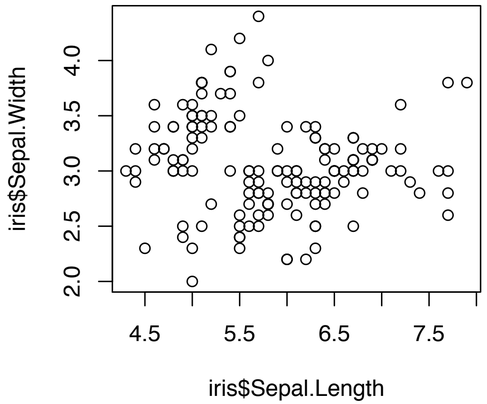


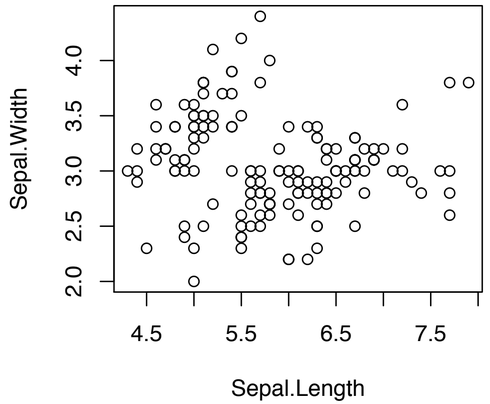
base - formula notation

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

> qplot(Sepal.Length, Sepal.Width, data = iris)

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



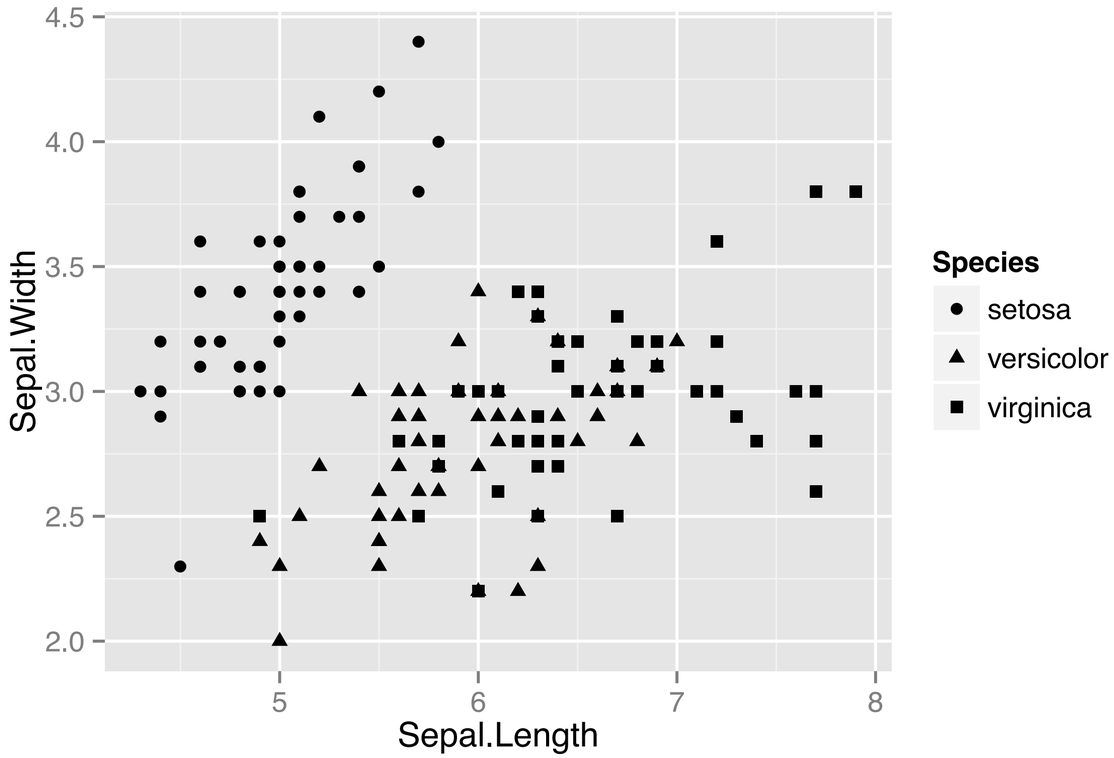






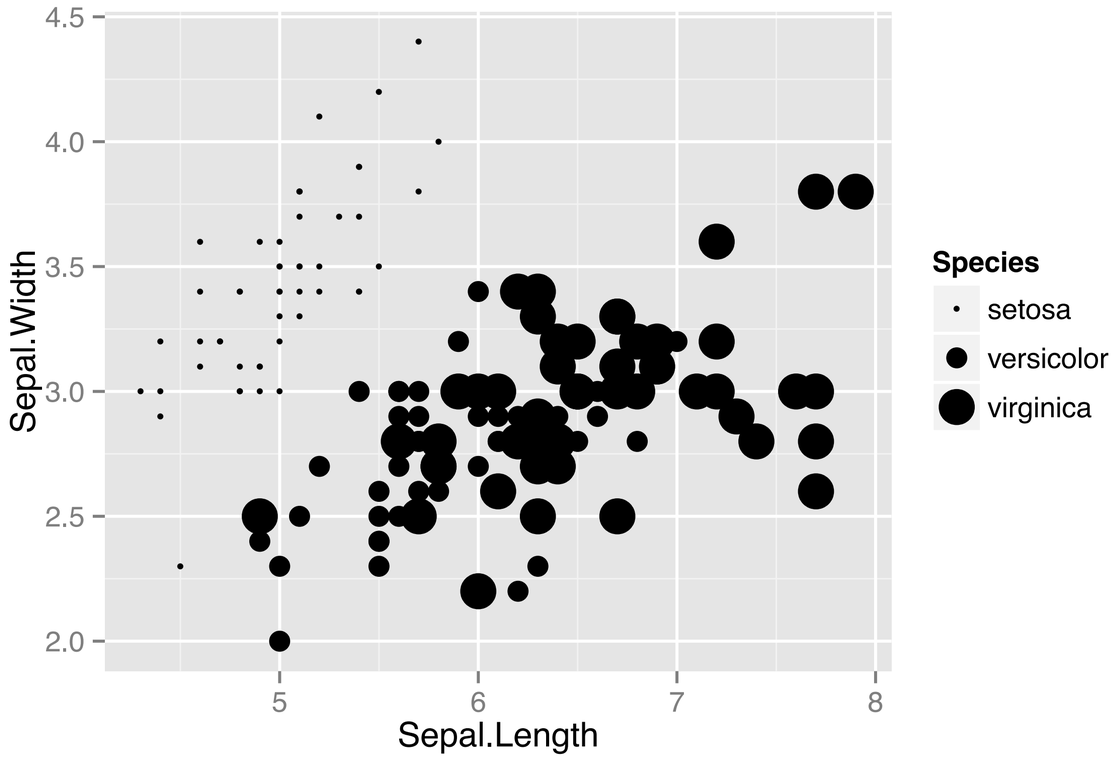
shape = Species

```
> qplot(Sepal.Length, Sepal.Width, data = iris, shape = Species)
```



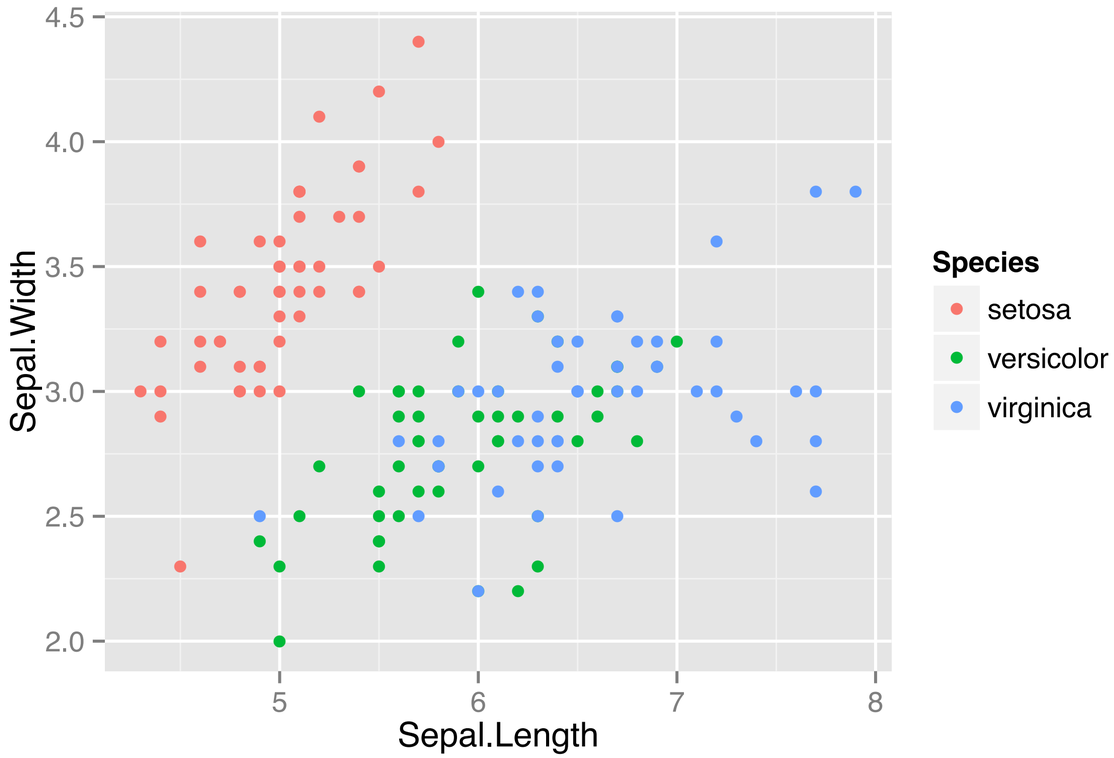
size = Species

```
> qplot(Sepal.Length, Sepal.Width, data = iris, size = Species)
```



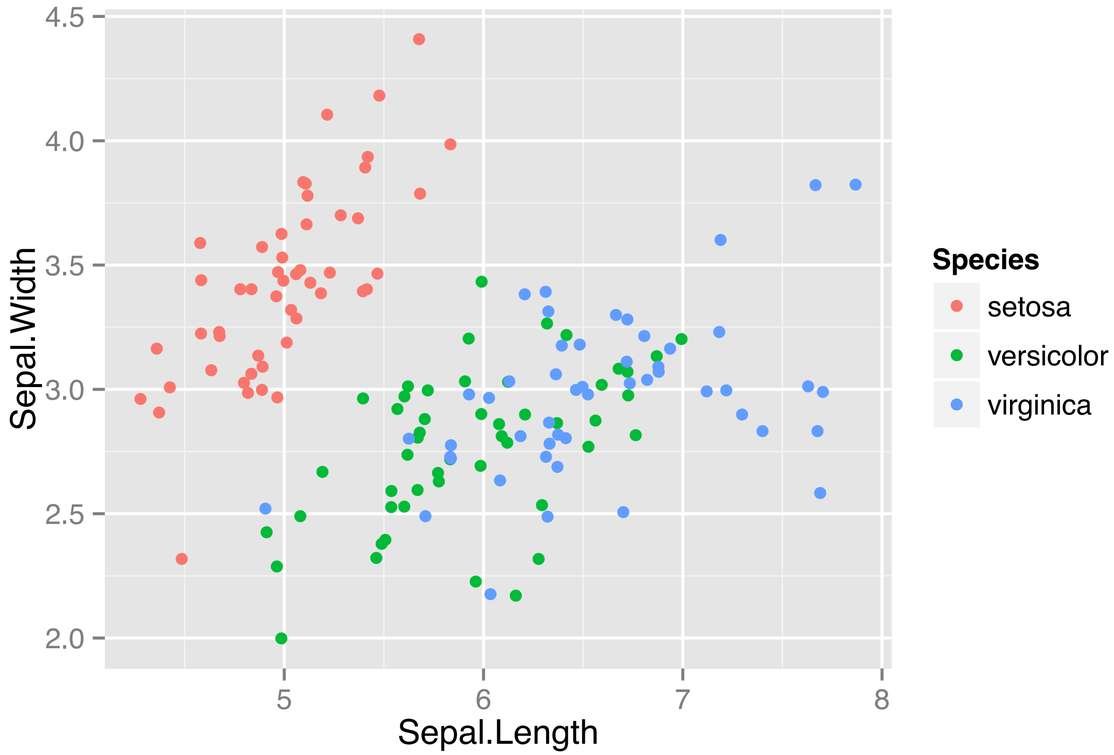
col = Species

```
> qplot(Sepal.Length, Sepal.Width, data = iris, col = Species)
```



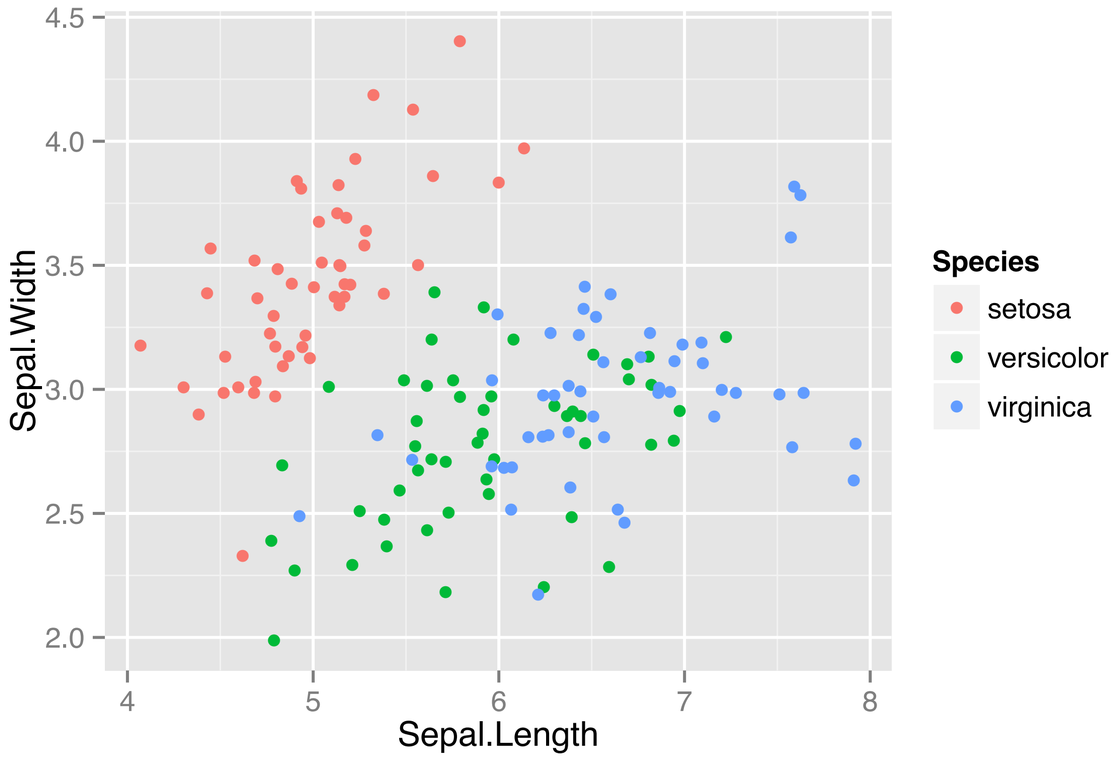
geom argument

```
> qplot(Sepal.Length, Sepal.Width, data = iris, col = Species,
        geom = "jitter")
```



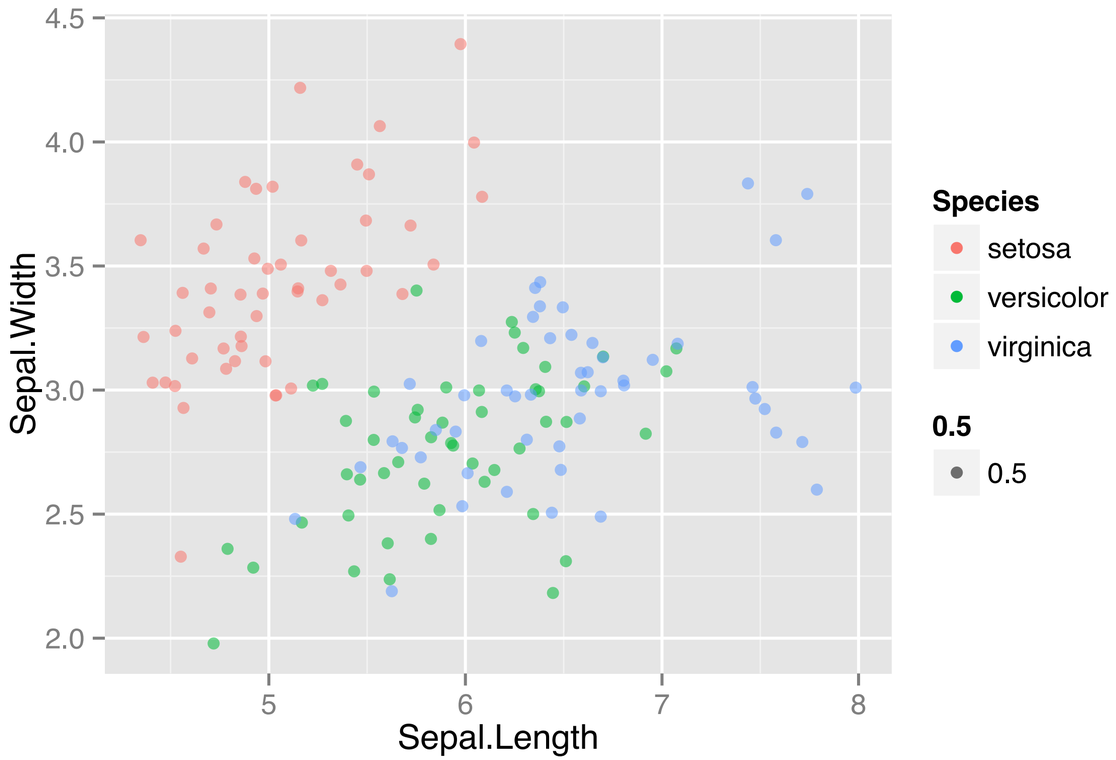
position argument

```
> qplot(Sepal.Length, Sepal.Width, data = iris, col = Species,
        position = "jitter")
```



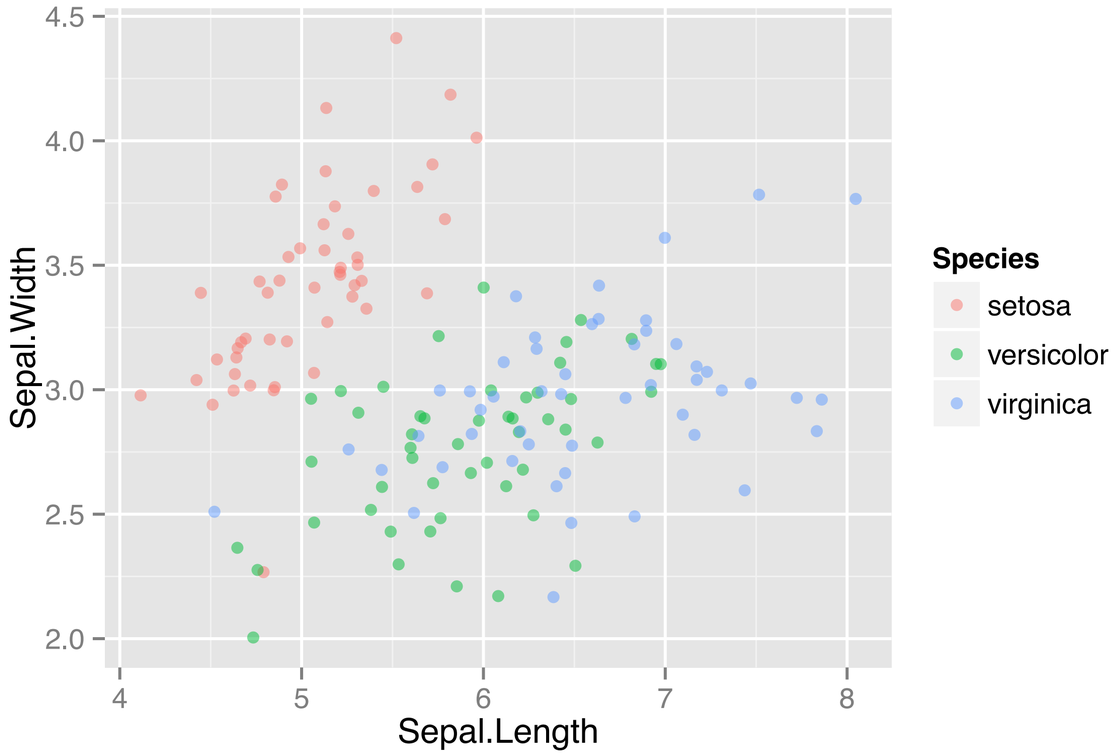
alpha

```
> qplot(Sepal.Length, Sepal.Width, data = iris, col = Species,
        position = "jitter", alpha = 0.5)
```



alpha

```
> qplot(Sepal.Length, Sepal.Width, data = iris, col = Species,
        position = "jitter", alpha = I(0.5))
```

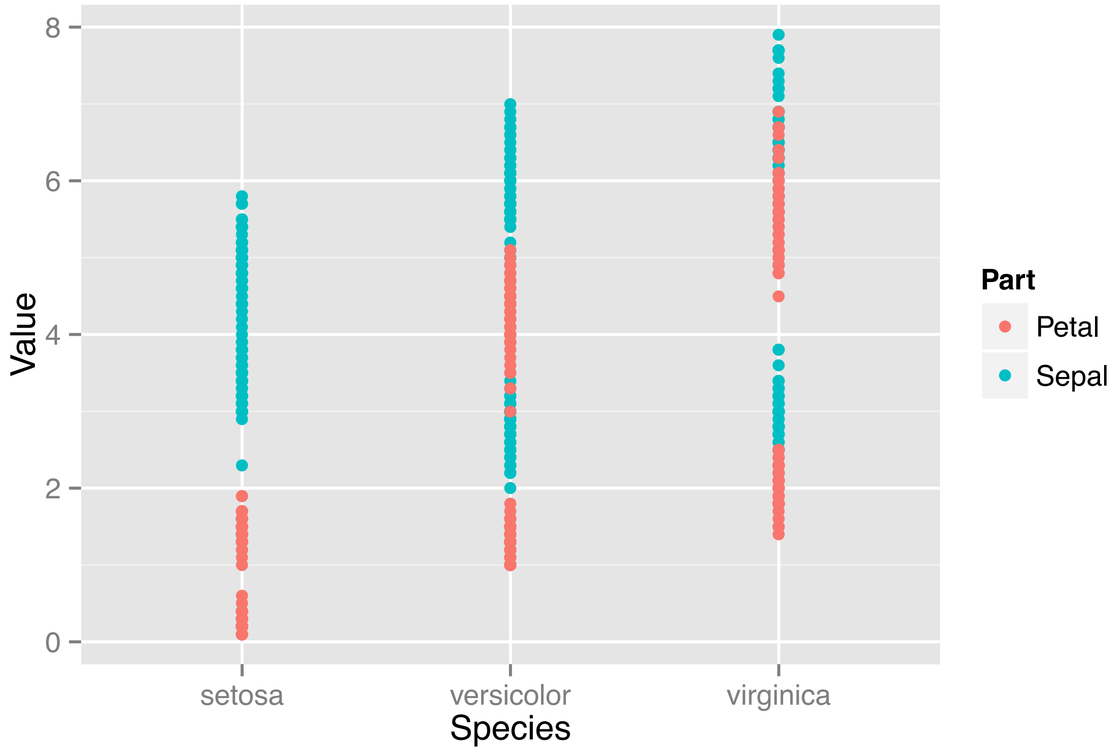


iris.tidy

```
> str(iris.tidy)
'data.frame':600 obs. of  4 variables:
 $ Species: Factor w/ 3 levels "setosa","versicolor",..: 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 ...
```

Continuous vs Categorical

```
> qplot(Species, Value, data = iris.tidy, col = Part)
```

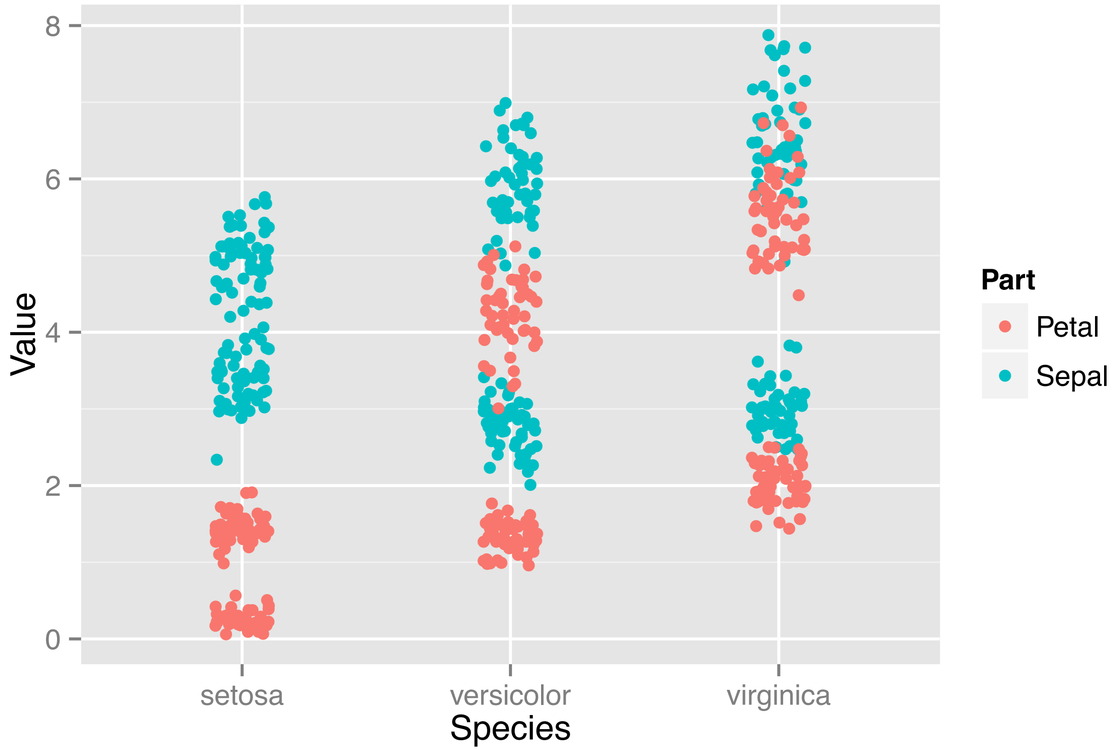


position = "jitter"

```
> qplot(Species, Value, data = iris.tidy, col = Part,
        position = "jitter")
```

Jitter manually

```
> posn.j <- position_jitter(0.1)
> qplot(Species, Value, data = iris.tidy, col = Part,
        position = posn.j)
```



Comparison

```
> posn.j <- position_jitter(0.1)

> # qplot
> qplot(Species, Value, data = iris.tidy, col = Part,
        position = posn.j)

> # ggplot
> ggplot(iris.tidy, aes(x = Species, y = Value, col = Part)) +
    geom_point(position = posn.j)
```

Fine for easy plots

Very flexible

DATA VISUALIZATION WITH GGPLOT2

Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

Wrap-up

Data
Explore
Confirm
and
Analyse
Scientist
Explain
Inform
and
Persuade
Reader

Two main things

- Grammatical Elements
- Aesthetic Mappings

Grammatical Elements

Article	*The*	*A*	*The*
Adjective	*quick brown*	*rabid red*	
Noun	*fox*	*fox*	*Hunter*
Verb	*jumps*	*bit*	*shot*
Preposition	*over*		
Article	*the*	*the*	*the*
Adjective	*lazy*	*friendly*	*rabid red*
Noun	*dog.*	*dog.*	*fox.*

Grammatical Elements

Grammatical Elements

Aesthetic Mappings

- How to map variables onto aesthetics
- Aesthetics = scales = encoding elements

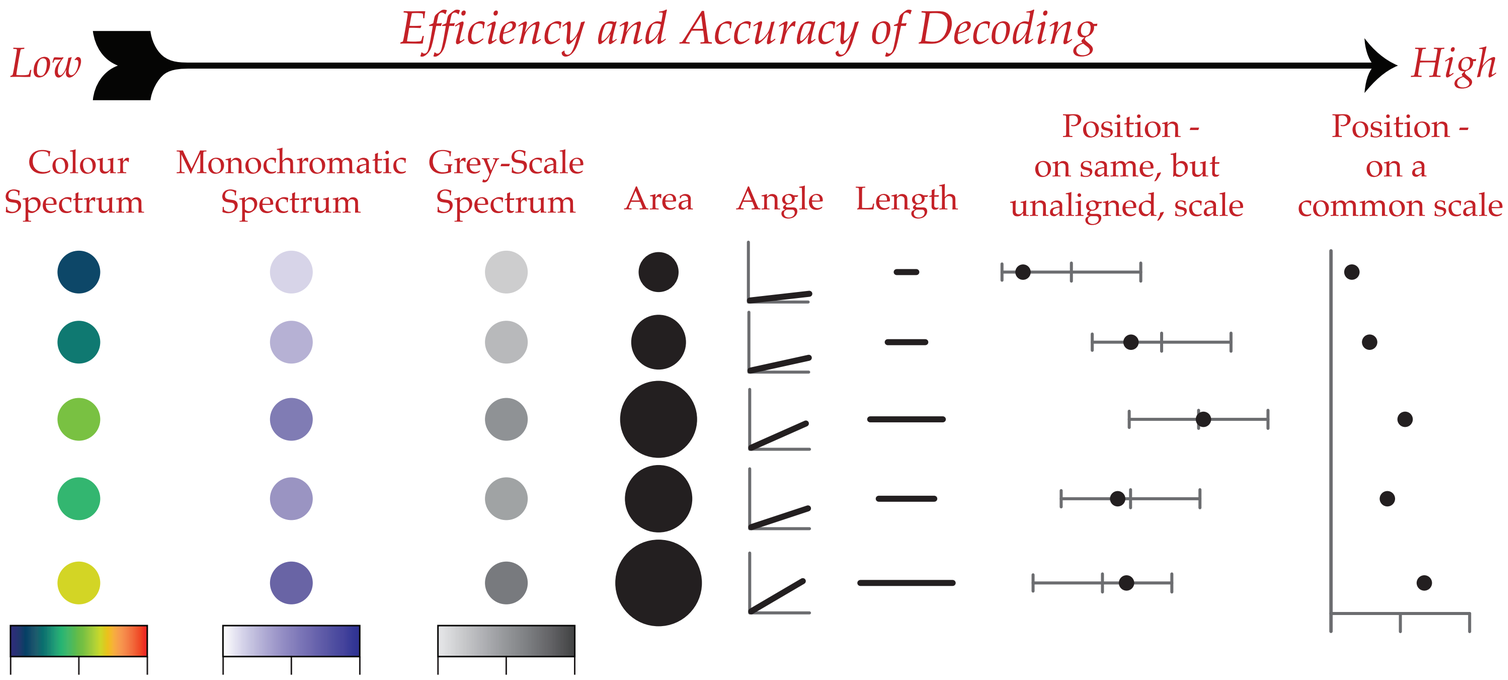
- Which variable?
- Which variable for which aesthetic?

iris

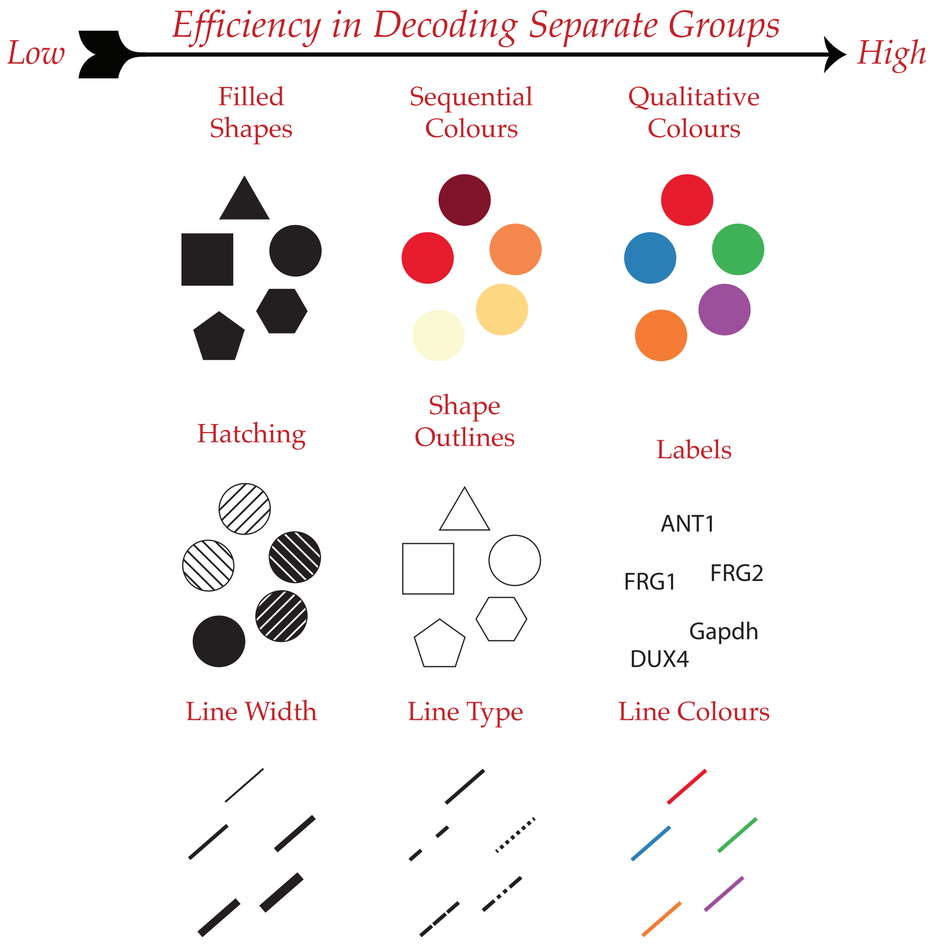
- iris `Species` `Sepal.Length` `Sepal.Width` `Petal.Length` `Petal.Width`
- iris.wide `Species` `Part` `Length` `Width`
- iris.mixed `Species.Part` `Length` `Width`
- iris.tidy `Species` `Part` `Measure` `Value`

- Choice of data format depends on desired plot!

Aesthetics - Continuous Variables



Aesthetics - Categorical Variables

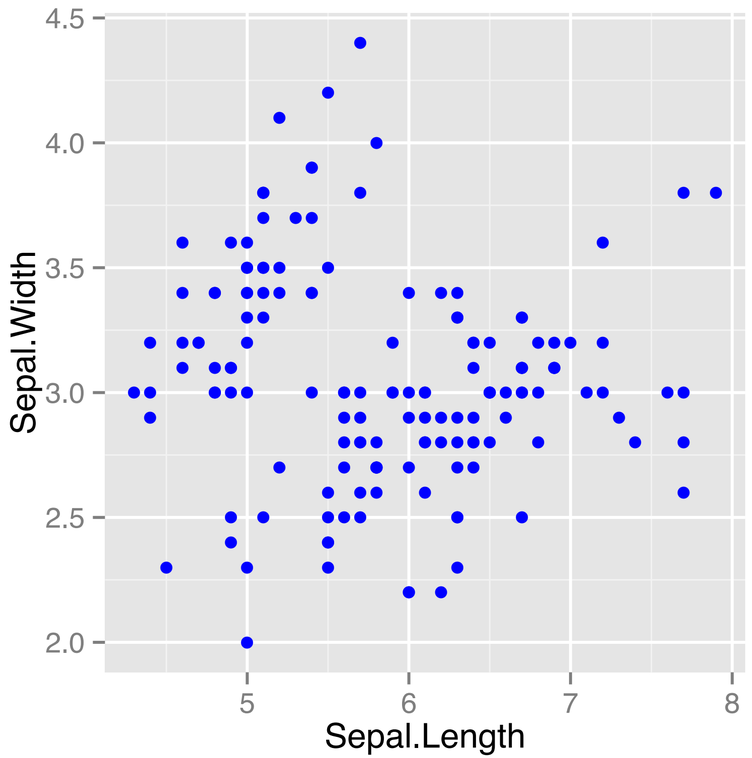


Common plot types

- Scatter plots
  - points, jitter, abline
- Line plots
  - line
- Bar plots
  - histogram, bar, errorbar

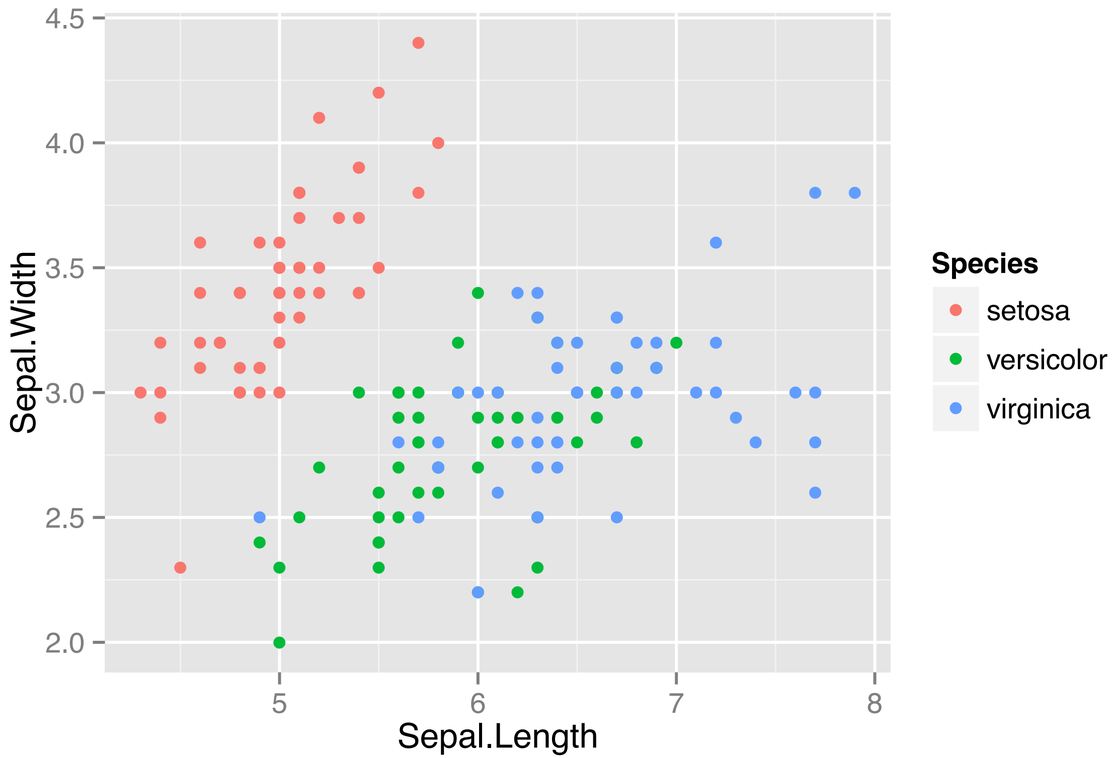
Attribute

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



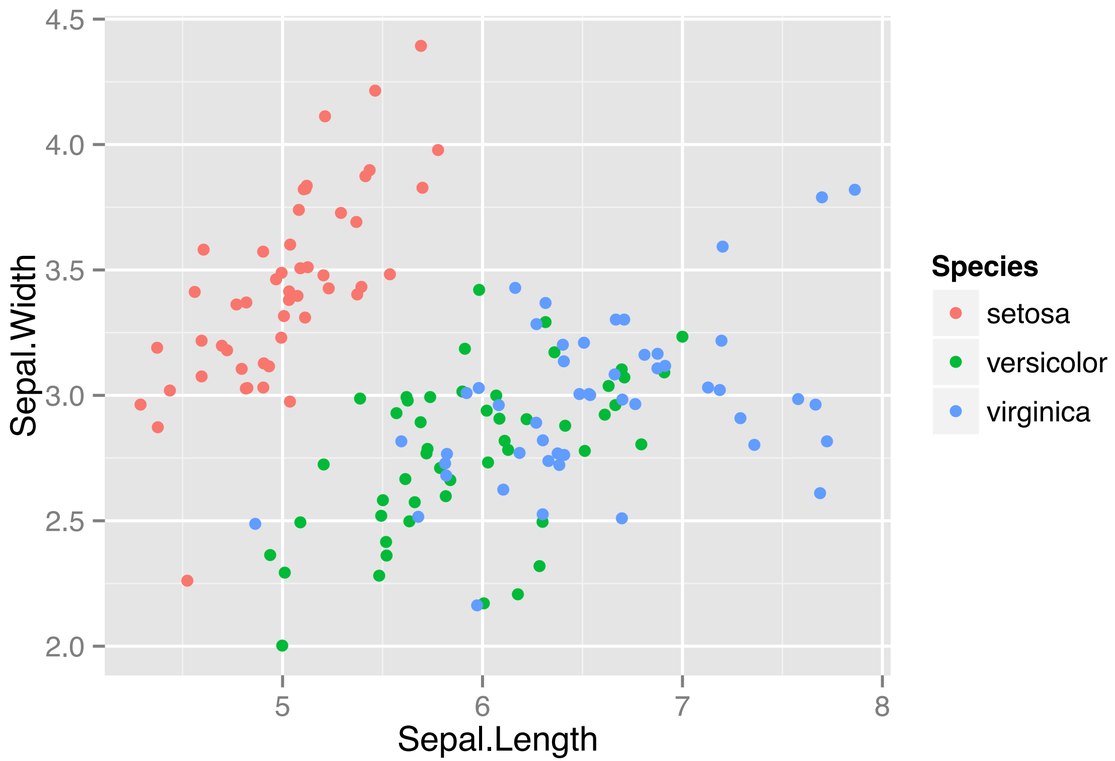
Aesthetic

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



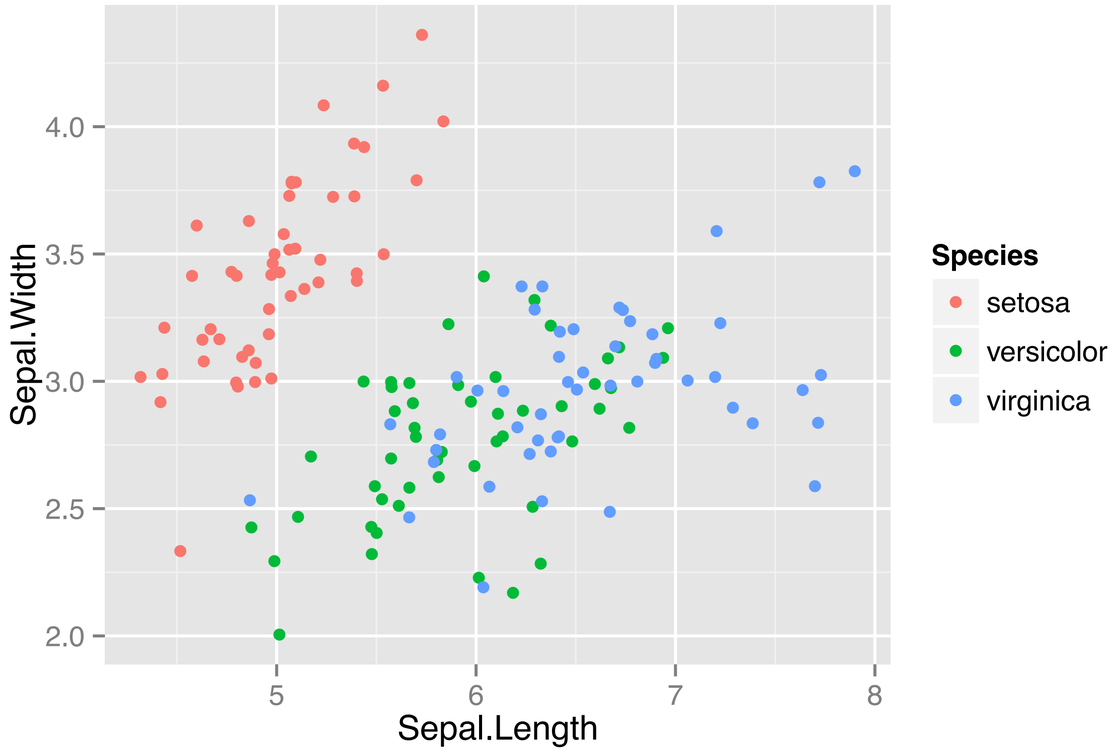
position = "jitter"

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point(position = "jitter")
```



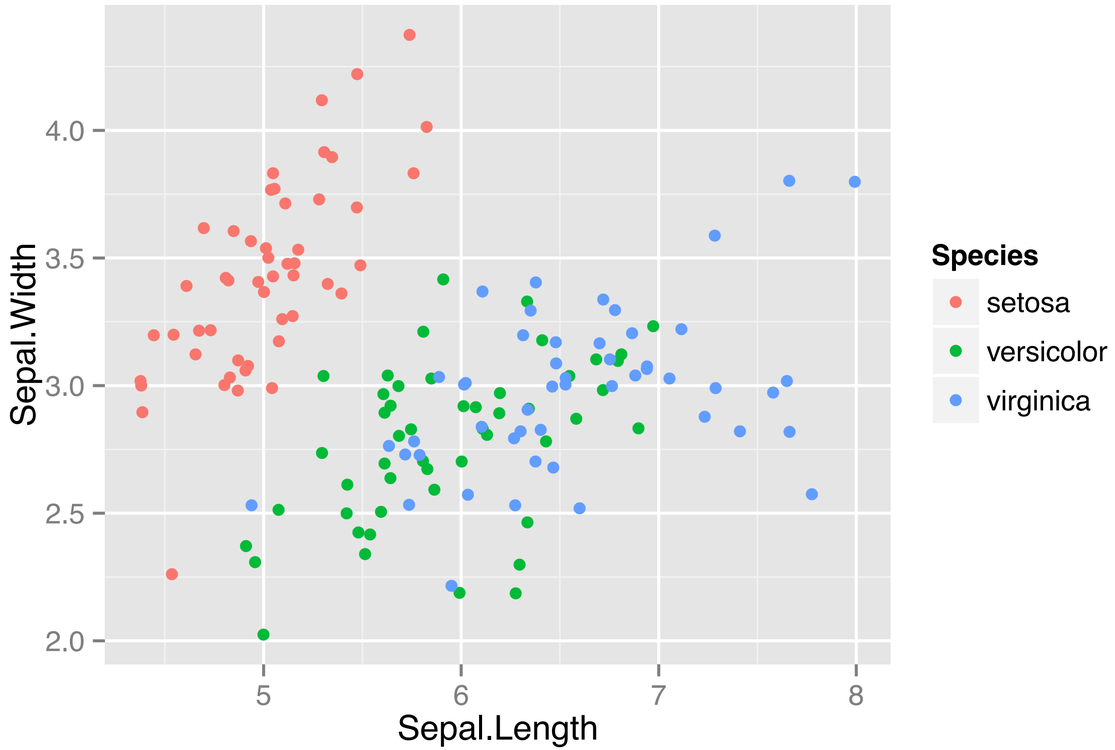
geom_jitter

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



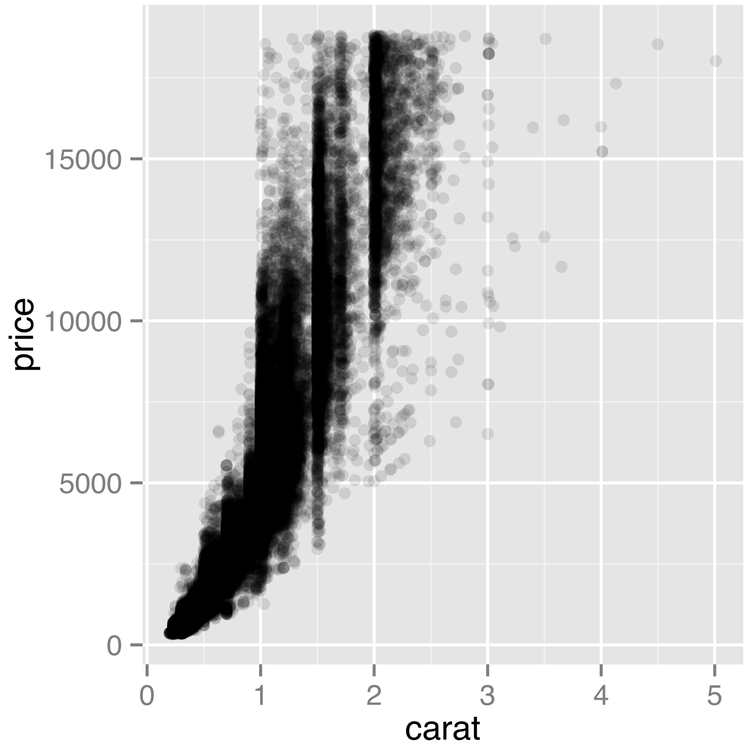
Manual jitter

```
> posn.j <- position_jitter(0.1)
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point(position = posn.j)
```



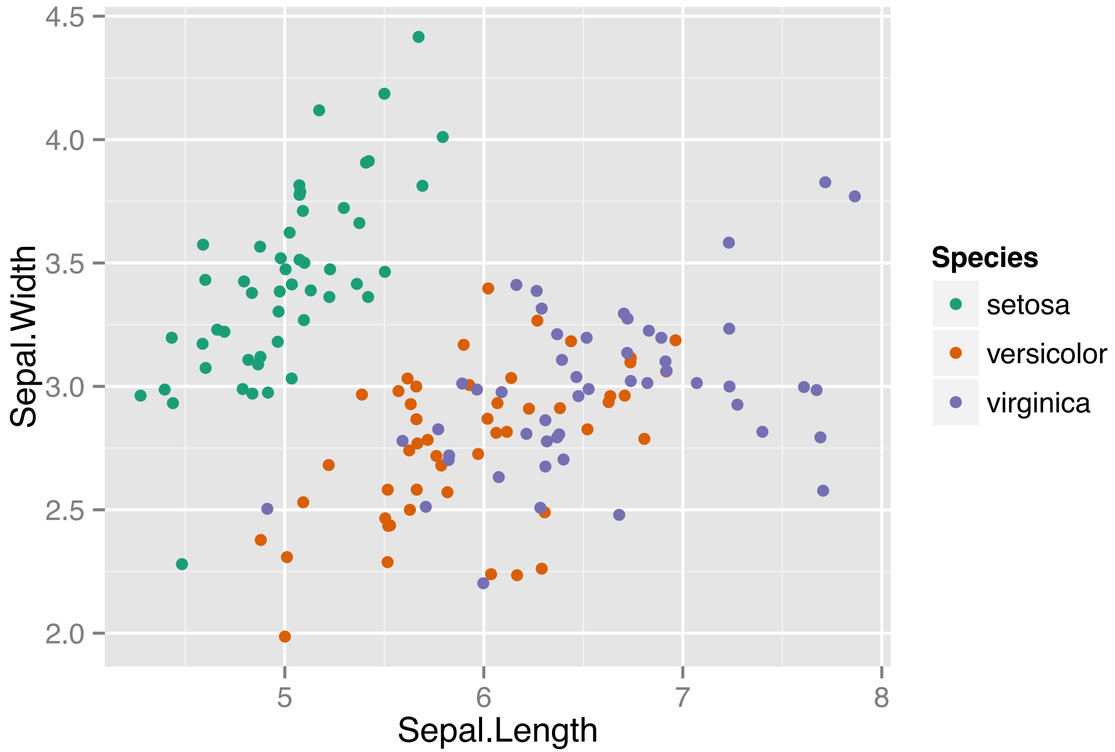
Diamonds

```
> ggplot(diamonds, aes(carat, price)) +
    geom_point(alpha = 0.1)
```



Scales

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_jitter() +
    scale_colour_brewer(palette = "Dark2")
```



DATA VISUALIZATION WITH GGPLOT2

Let’s practice!