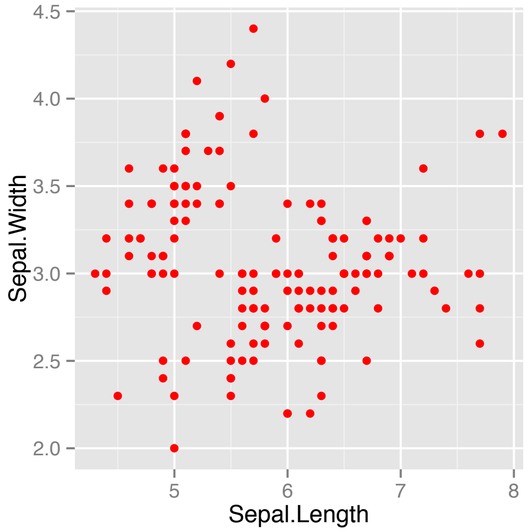
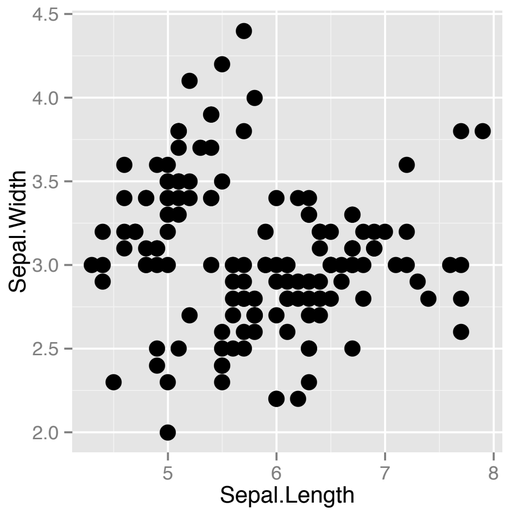
DATA VISUALIZATION WITH GGPLOT2

Visible Aesthetics

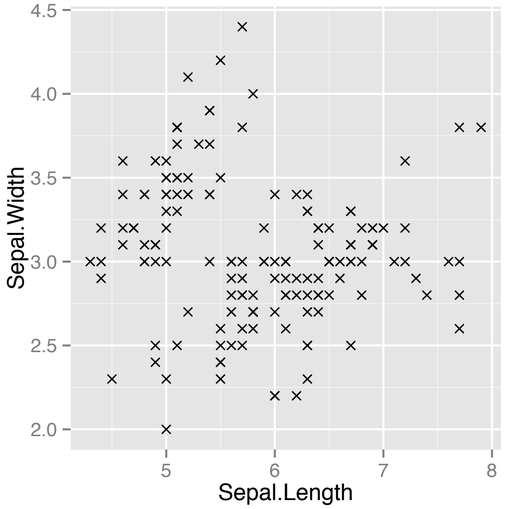
Aesthetics? Attributes!



Type	Property
Colour	Red



Type	Property
Size	10



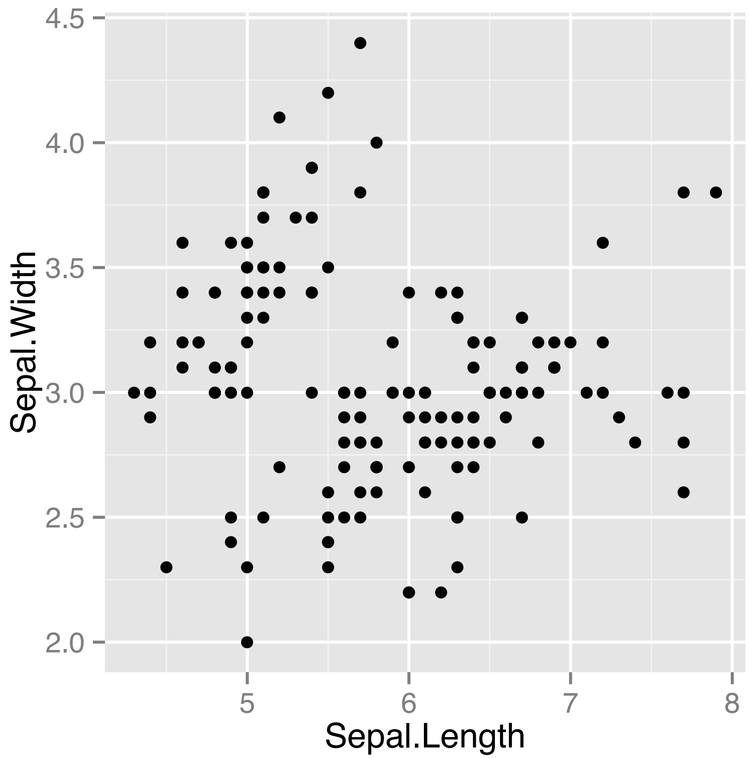
Type	Property
Shape	4

Type	Variable
Colour	Species

mapping Species on colour

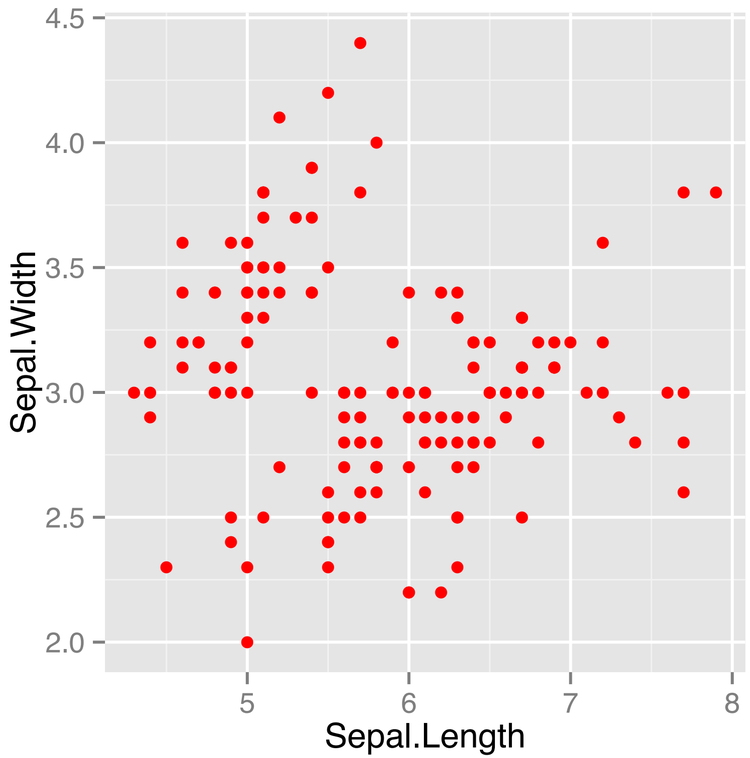
Mapping

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



Attribute

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

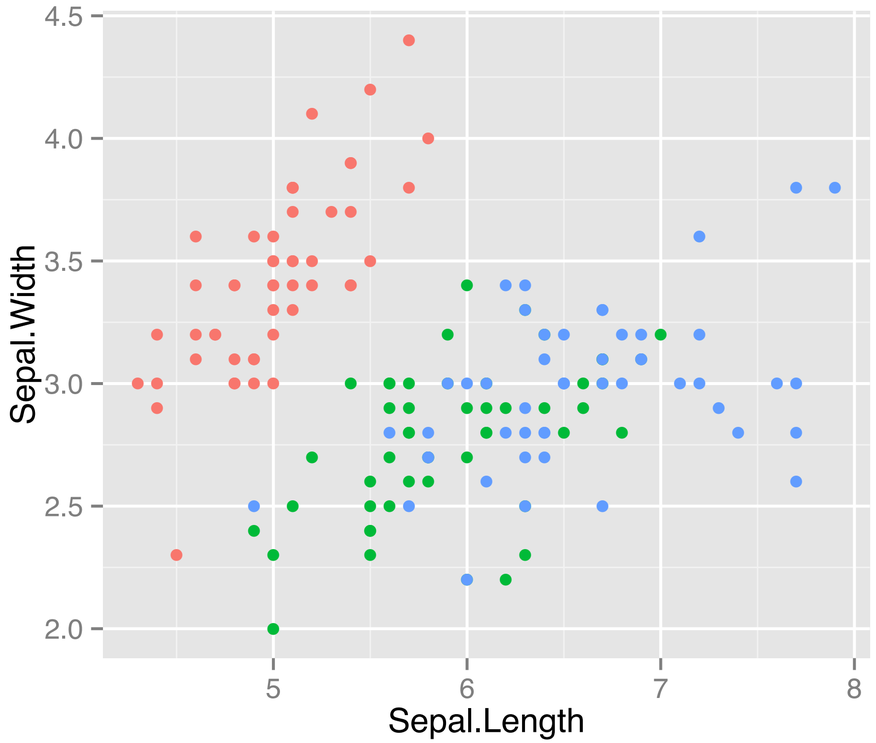


so mapping occurs in aes()
but you can specify attributes
just in that geom. Interestingly
you can also specify aes() within
any geom - that will do the mapping

That is more commonly done if you
wish to use different data set
for the other geom

Mapping onto color

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

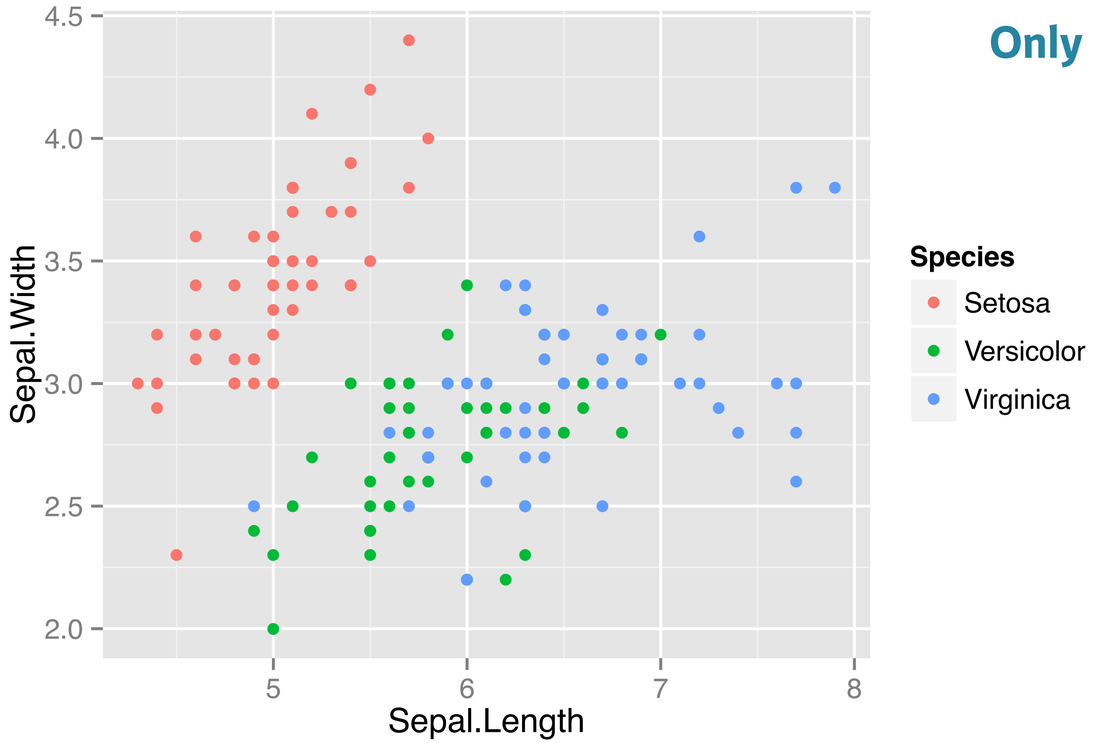


Data frame column mapped onto visible aesthetic

Aesthetics in aes(), attributes in geom_()

Mapping onto color (2)

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



Only if different data sources

Typical Aesthetics

Aesthetic	Description
x	X axis position
y	Y axis position
colour	Colour of dots, outlines of other shapes
fill	Fill colour
size	Diameter of points, thickness of lines
alpha	Transparency
linetype	Line dash pattern
labels	Text on a plot or axes
shape	Shape

DATA VISUALIZATION WITH GGPLOT2

Aesthetics Best Practices

Which Aesthetic?

- Be creative
- Clear guidelines
- Jacques Bertin (cartographer)
  - *The Semiology of Graphics*, 1967
- William Cleveland
  - Perception of visual elements (90s)

Form follows Function

there is a function to the plot and it depends on who your audience is

for other specialists

for general public

if data is not effectively presented - it is junk :-]

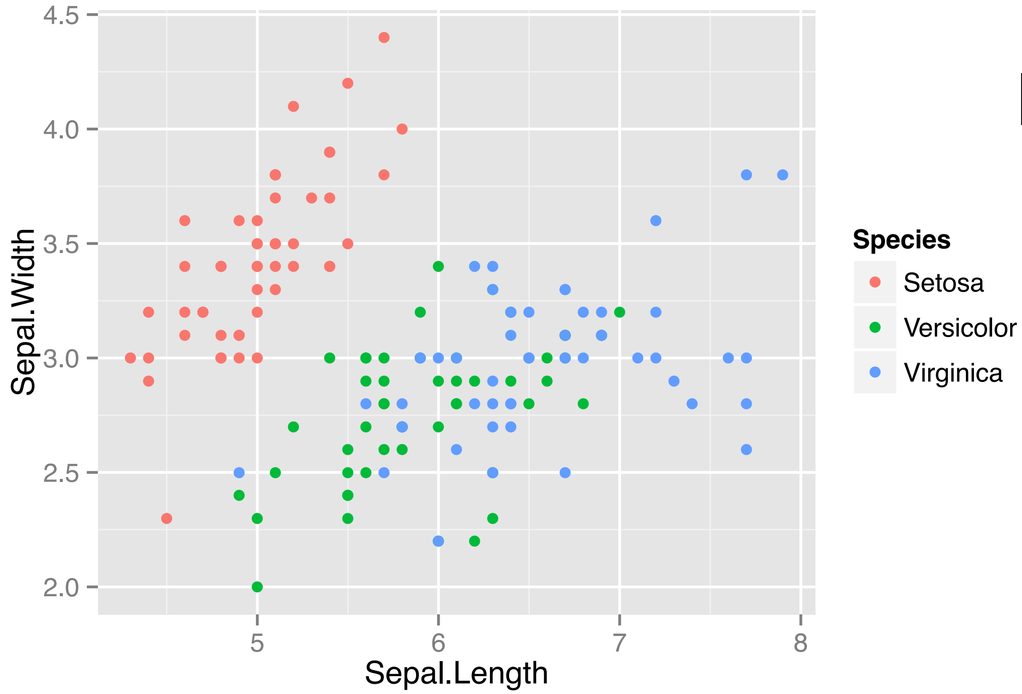
Aesthetics

never misrepresent your data
do not confuse the reader (overly complex figures)
encode the data (numbers, tags) into visual language

Aesthetic	Description
x	X axis position
y	Y axis position
colour	Colour of dots, outlines of other shapes
fill	Fill colour
size	Diameter of points, thickness of lines
alpha	Transparency
linetype	Line dash pattern
labels	Text on a plot or axes
shape	Shape

Aesthetics - Continuous Variables

```
> ggplot(iris.1, aes(x = Sepal.Length,
                     y = Sepal.Width,
                     col = Species)) +
    geom_point()
```

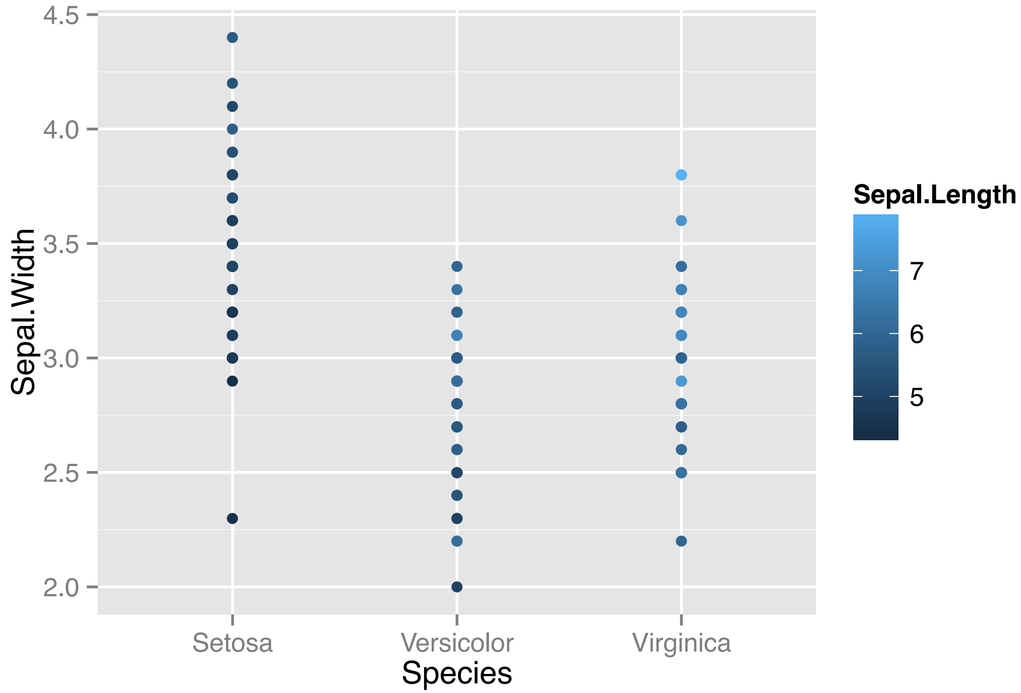


position on a common scale

color assigned to map to factor variable

Aesthetics - Continuous Variables

```
> ggplot(iris.1, aes(col = Sepal.Length,
                     y = Sepal.Width,
                     x = Species)) +
    geom_point()
```



color assigned to continuous variable

Aesthetics - Continuous Variables

Aesthetic	Description
x	X axis position
y	Y axis position
size	Diameter of points, thickness of lines
alpha	Transparency
colour	Colour of dots, outlines of other shapes
fill	Fill colour

less useful -> lower in the table

one data column can be assigned to more than one aesthetic e.g. size AND alpha

Guide - Continuous Variables

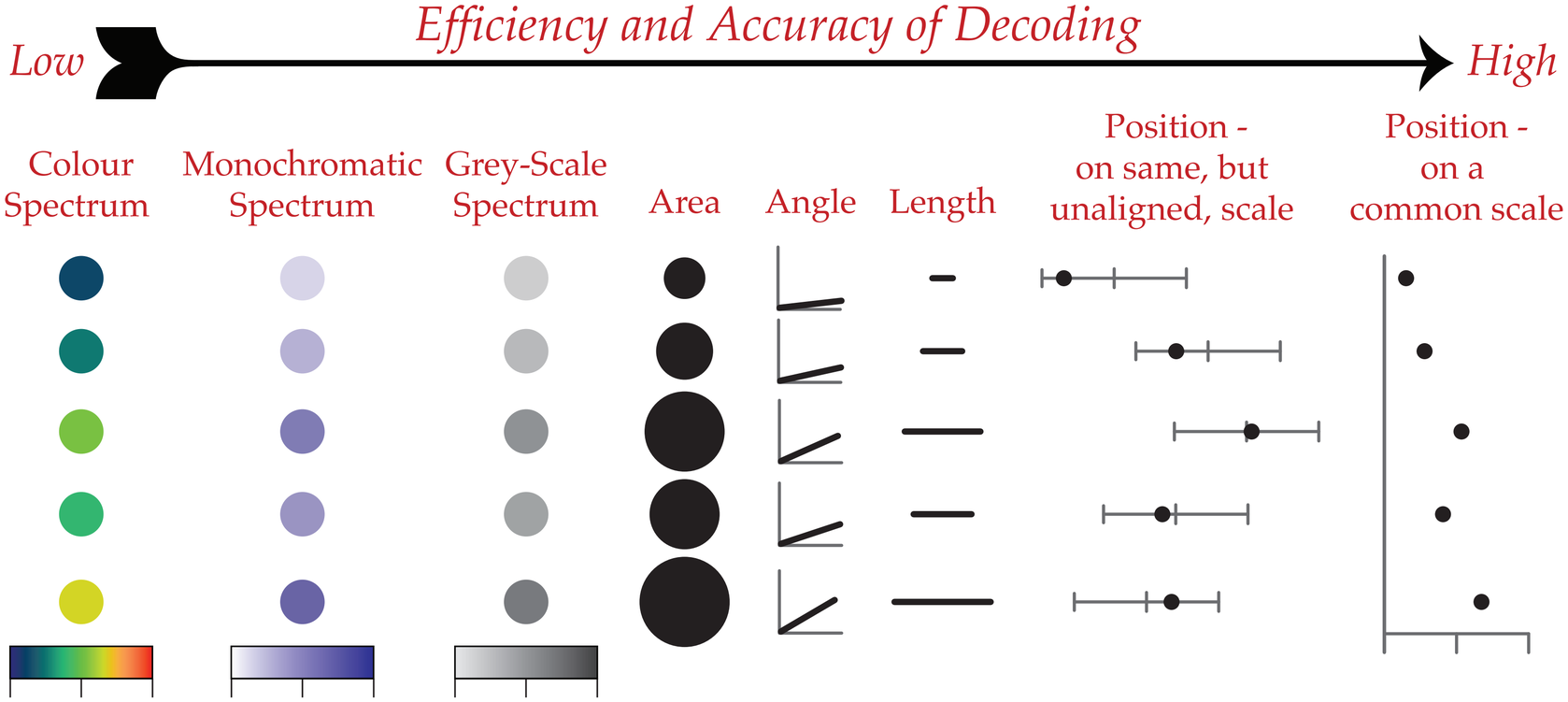
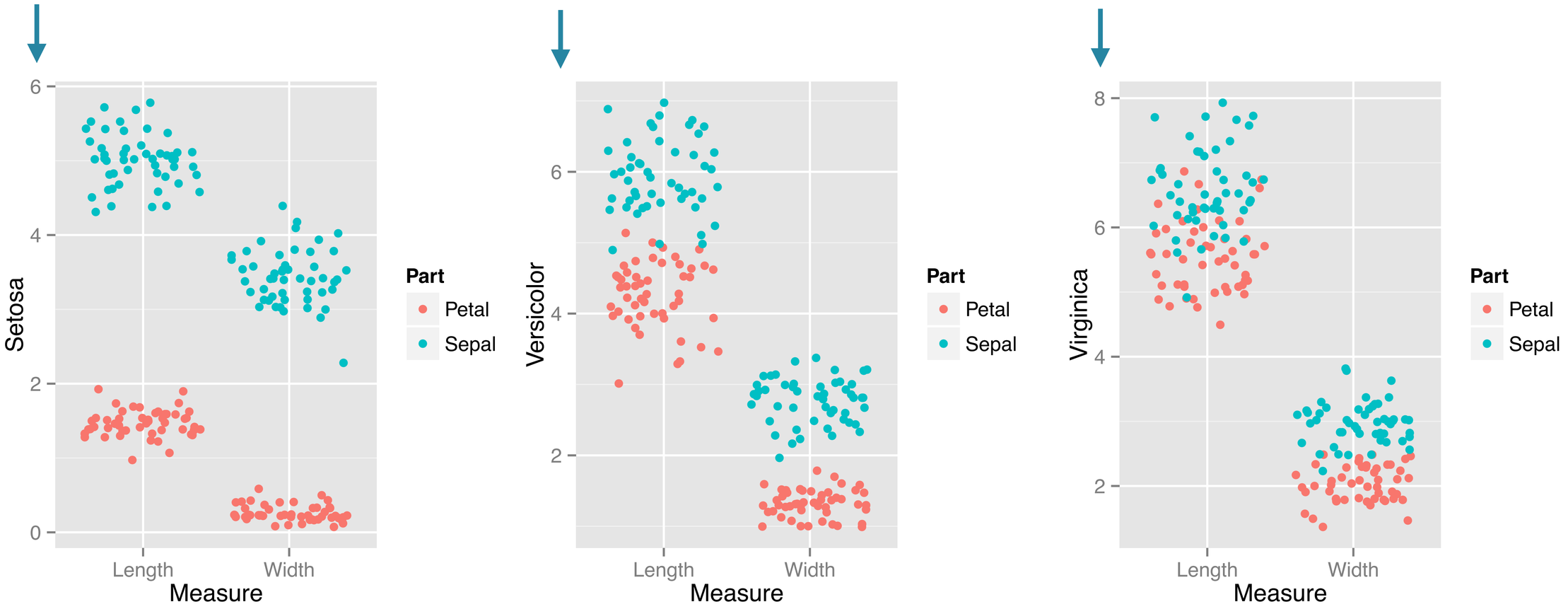


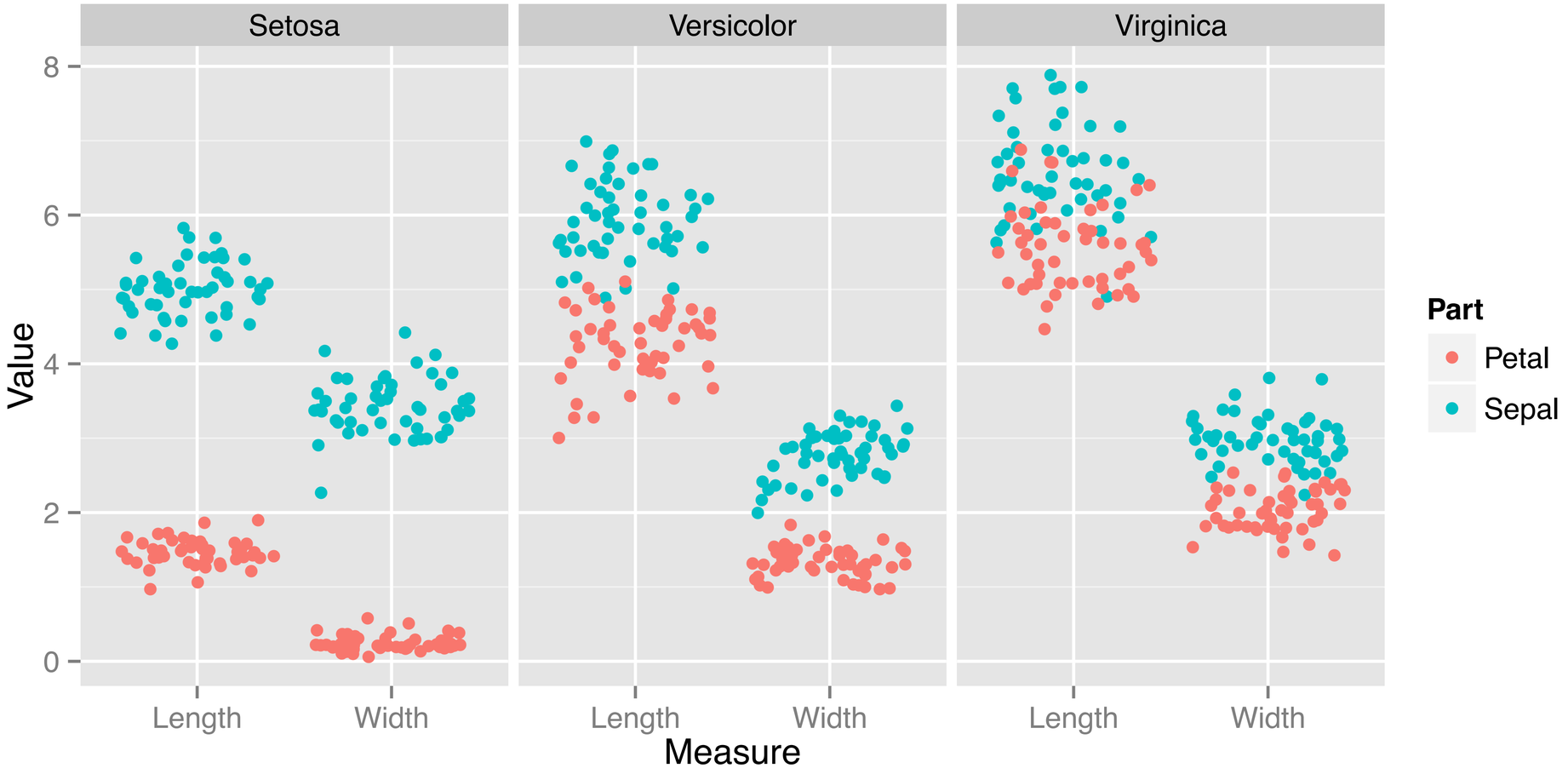
Image adapted from Wong, B, *Nat Met*, 7 (9), 2010, p665

Unaligned y axes

also 3 separate plots v.s. 3 facets

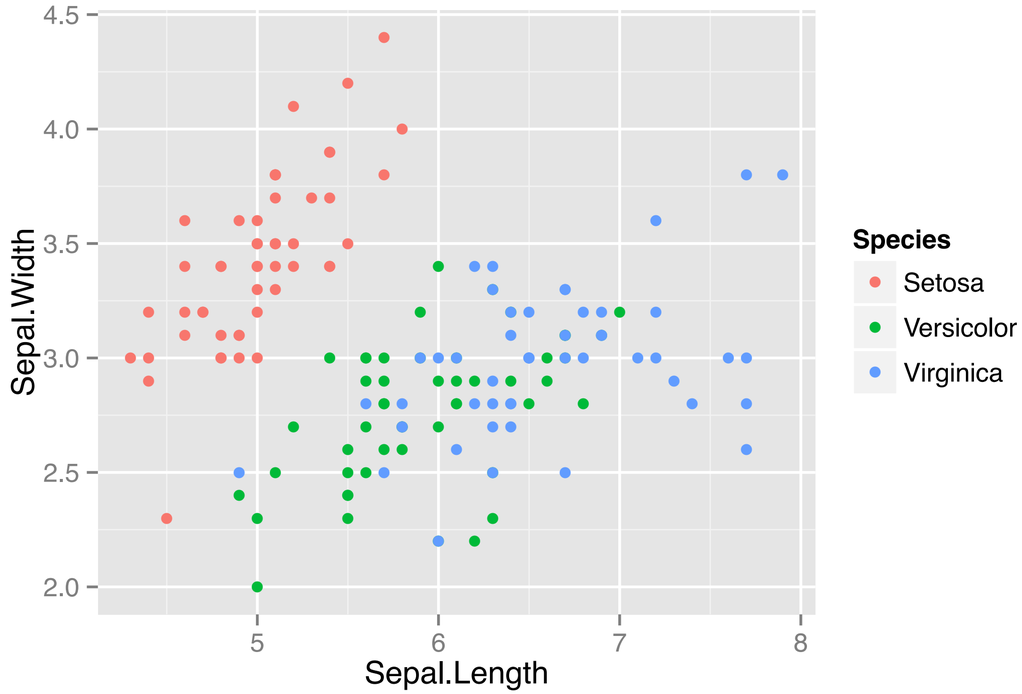


Common y axis



Aesthetics - Categorical Variables

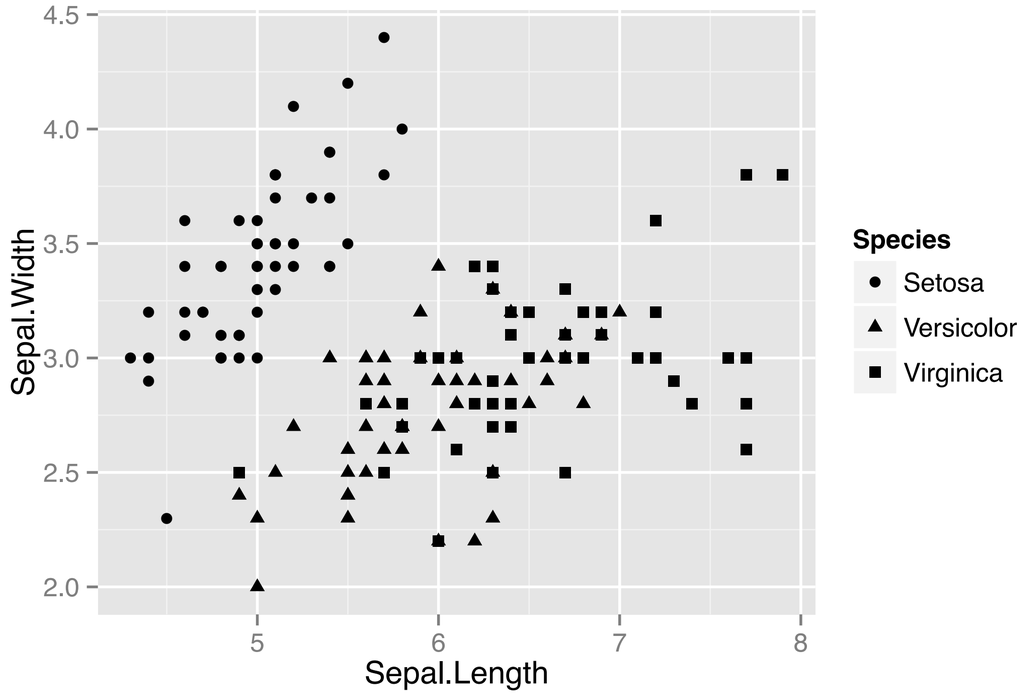
```
> ggplot(iris.1, aes(x = Sepal.Length,
                     y = Sepal.Width,
                     col = Species)) +
    geom_point()
```



categorical variables have more diverse choices
- since they represent small and finite groups
however it is good to choose something that
is easy to interpret.
Aim to remove unnecessary visual elements - not data

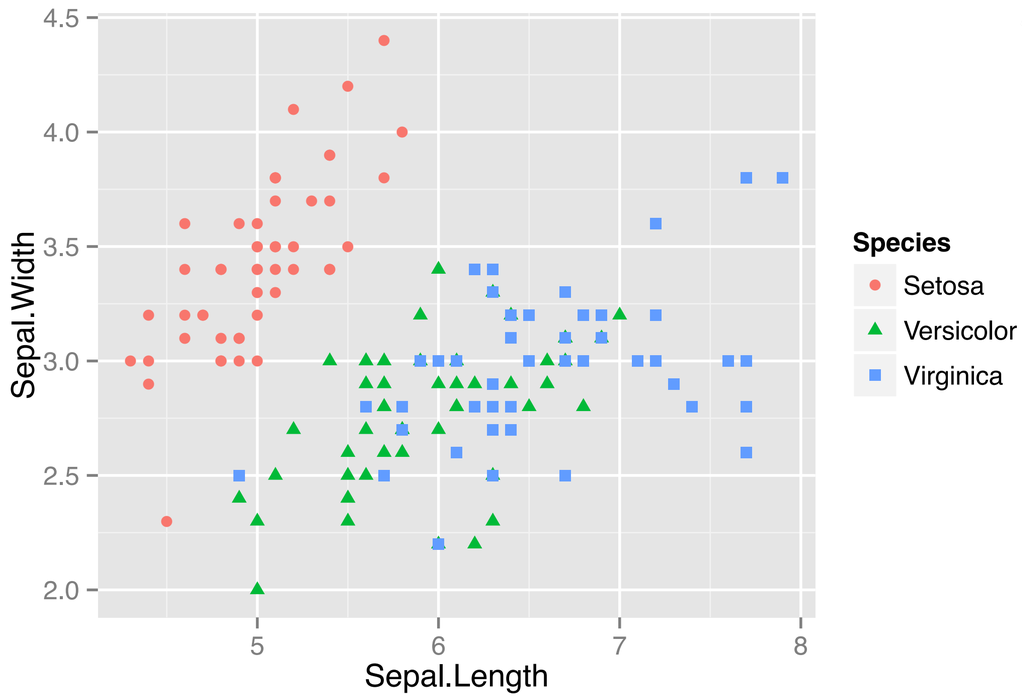
Aesthetics - Categorical Variables

```
> ggplot(iris.1, aes(x = Sepal.Length,
                     y = Sepal.Width,
                     shape = Species)) +
    geom_point()
```



Aesthetics - Categorical Variables

```
> ggplot(iris.1, aes(x = Sepal.Length, y = Sepal.Width,
                     shape = Species, col = Species)) +
  geom_point()
```

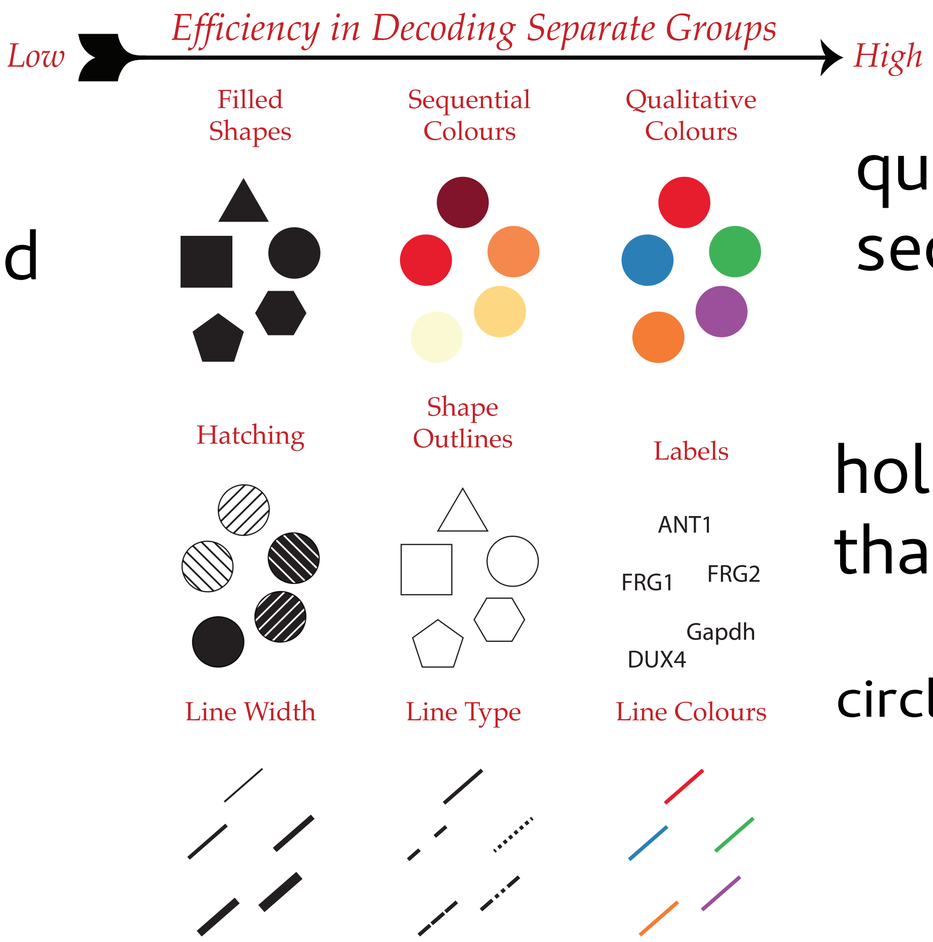


you can use two mappings for one variable (categorical variable)

Aesthetics - Categorical Variables

Aesthetic	Description
labels	Text on a plot or axes
fill	Fill colour
shape	Shape of point
alpha	Transparency
linetype	Line dash pattern
size	Diameter of points, thickness of lines

Aesthetics - Categorical Variables



better avoid left side elements in graphs

qual cols - nominal variables
seq col - for ordinal variables

hollow shapes easier to distinguish than solid shapes

circles preferable to shapes with straight lines

Image adapted from Wong, B, *Nat Met*, 7 (9), 2010, p665

DATA VISUALIZATION WITH GGPLOT2

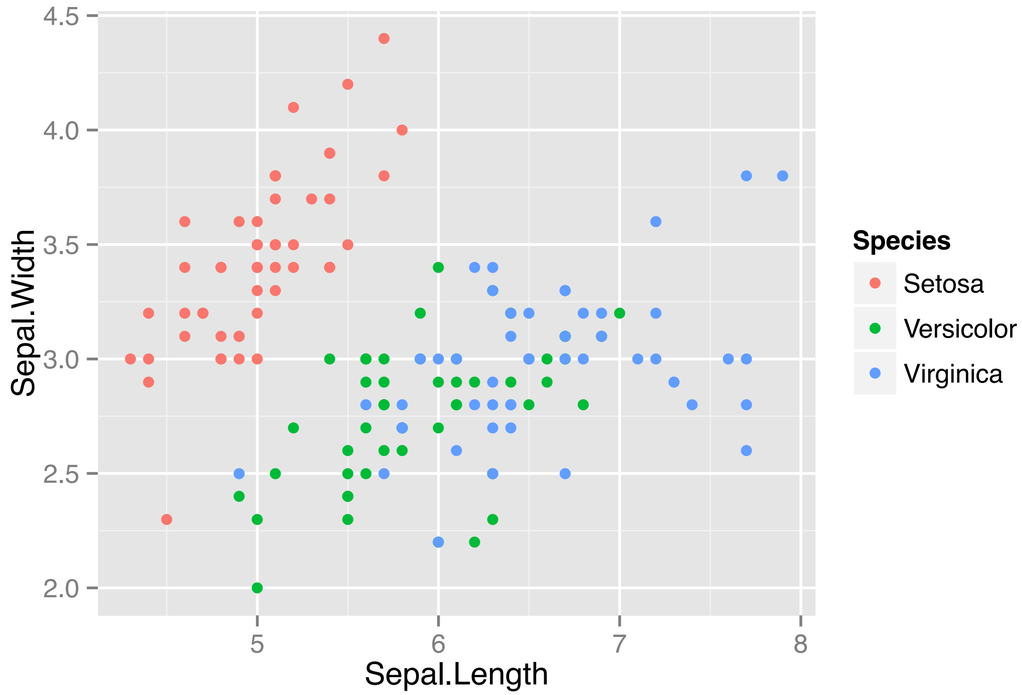
Modifying Aesthetics

Positions

- identity — value in the dataframe is exactly where the value (geom) will be placed on the plot
- dodge
- stack
- fill
- jitter — when there is too much overplotting and we need to add some random noise to x and y. We specify the amount of jitter such as o.1 or 0.7
- jitterdodge

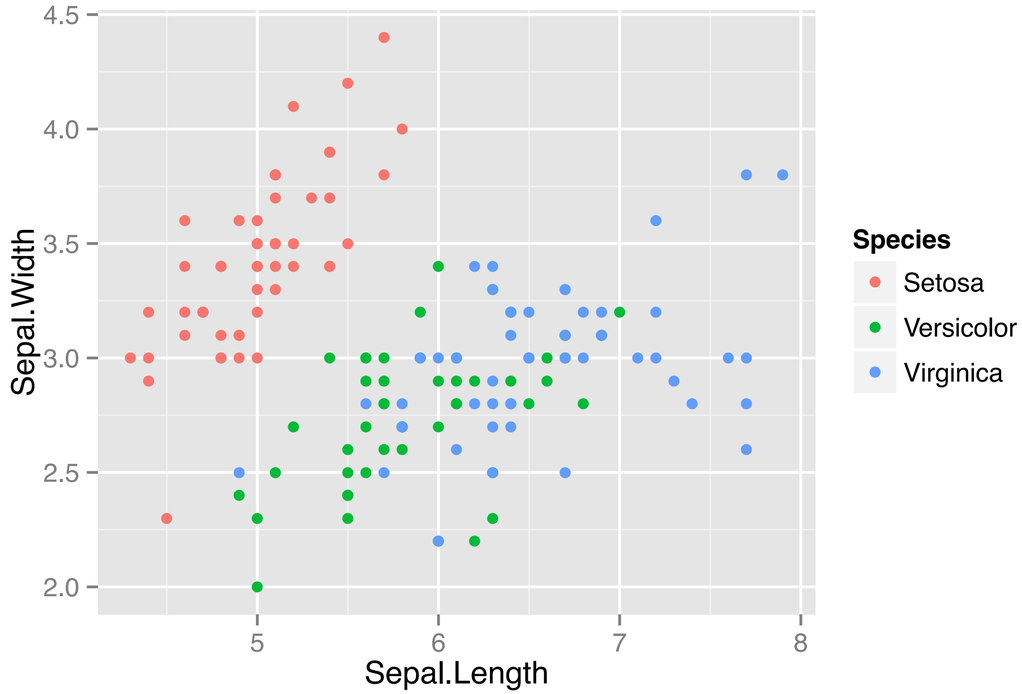
position identity (default)

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



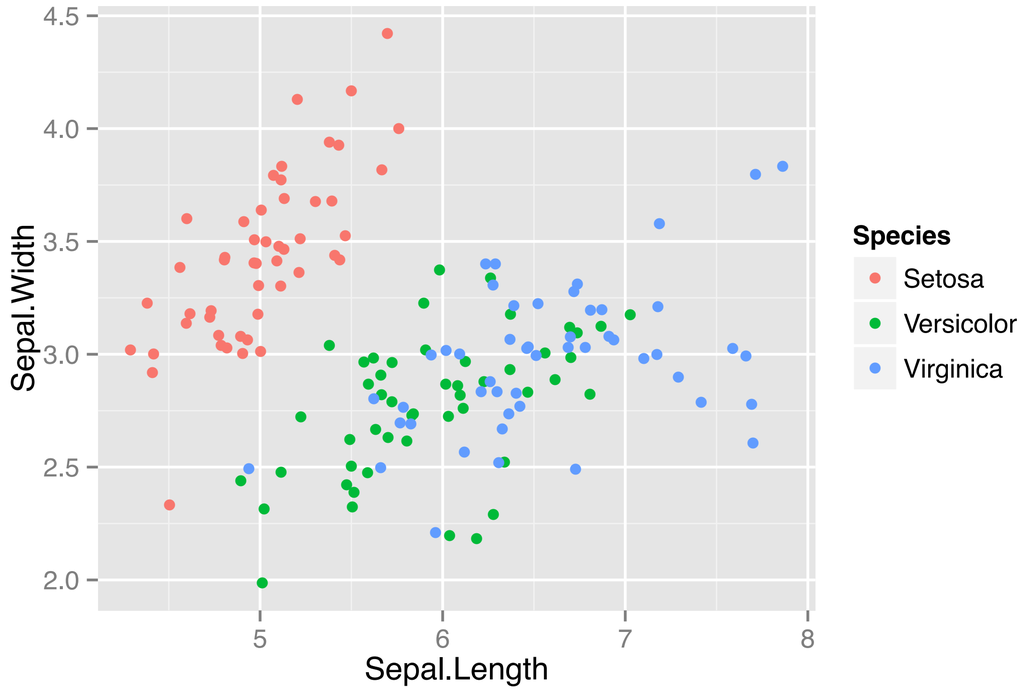
position identity (default)

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



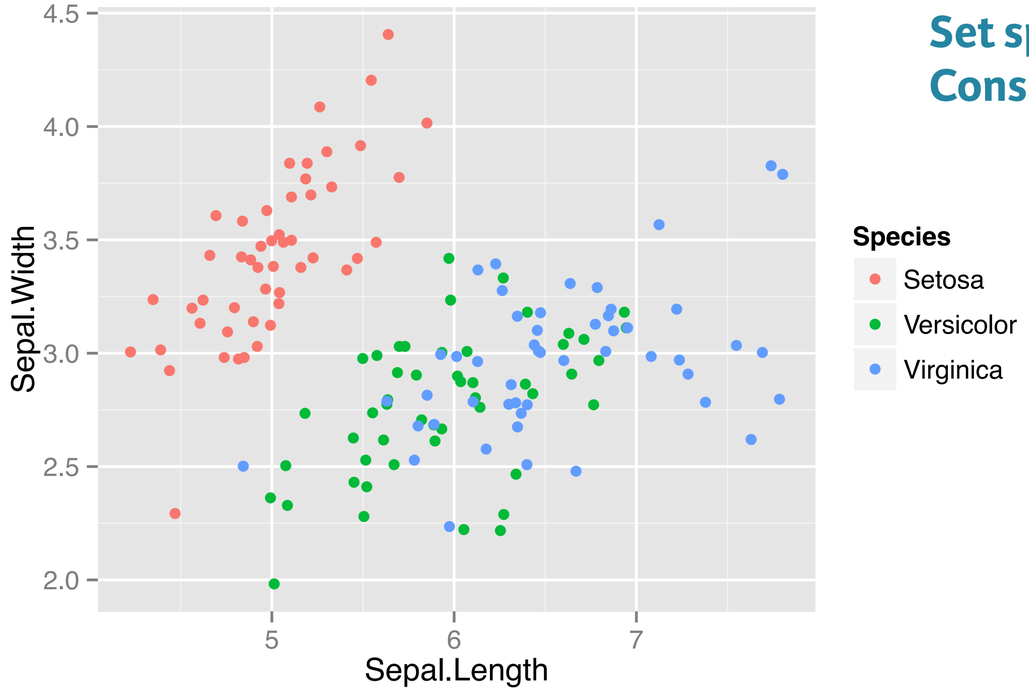
position jitter

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



position jitter (2)

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



Set specific arguments for the position
Consistency in jitter across plots

because we can re-use that position in all our plots

Scale Functions

- scale_x...
- scale_y...
- scale_color...
- scale_fill...
- scale_color...
- scale_shape...
- scale_linetype...

each of the aesthetics is a scale which we map data onto. Color is a scale and x and y is a scale therefore we can access them with scale underscore functions

factor - discrete - categorical - qualitative (depending on context but they all are the same)

first argument is the name of the scale. then the most common are:

limits - the limits of the scale

breaks - controls the breaks in the guide

expand - numeric vector of length 2 that controls space between axes and data

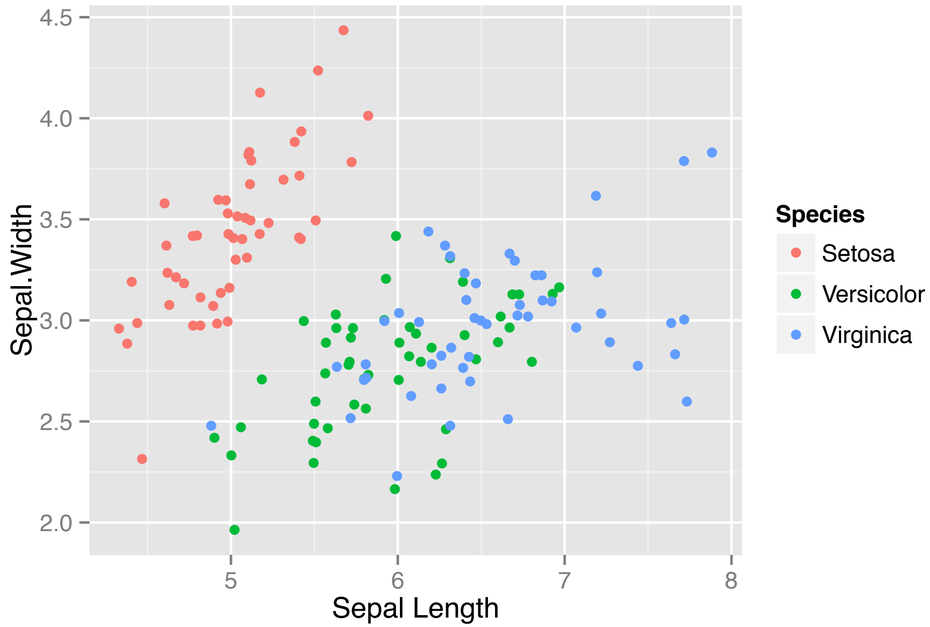
labels -are just the category names (such as in legend)

Scale Functions

- scale_x_continuous
- scale_y...
- scale_color_discrete
- scale_fill...
- scale_color...
- scale_shape...
- scale_linetype...

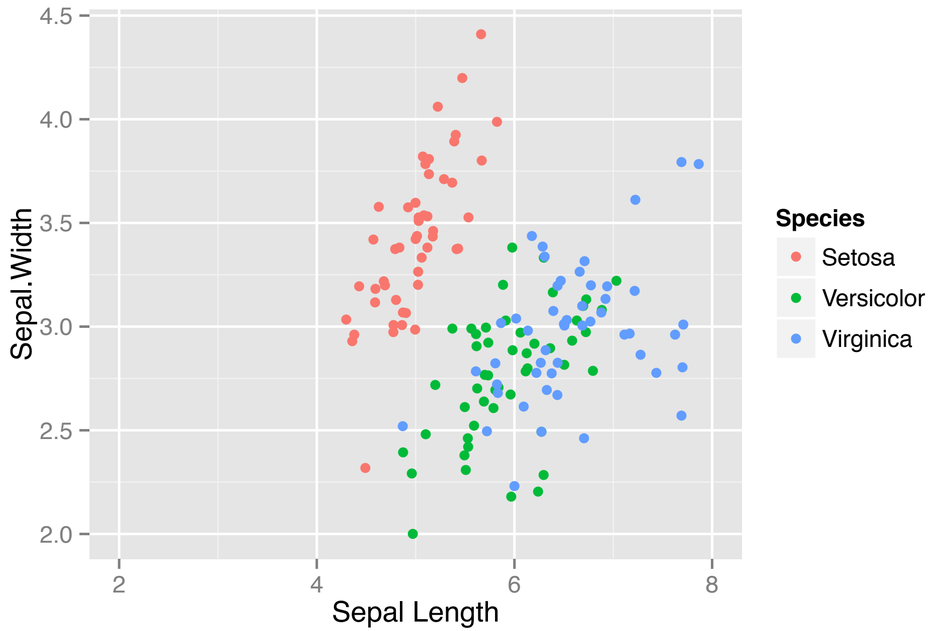
scale_

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



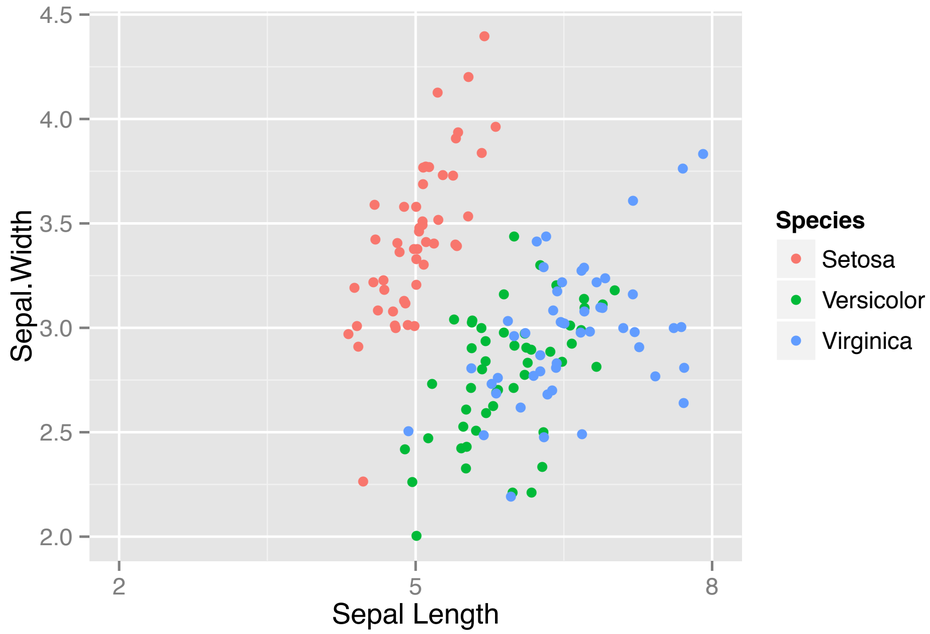
limit

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point(position = "jitter") +
    scale_x_continuous("Sepal Length", limits = c(2, 8)) +
    scale_color_discrete("Species")
```



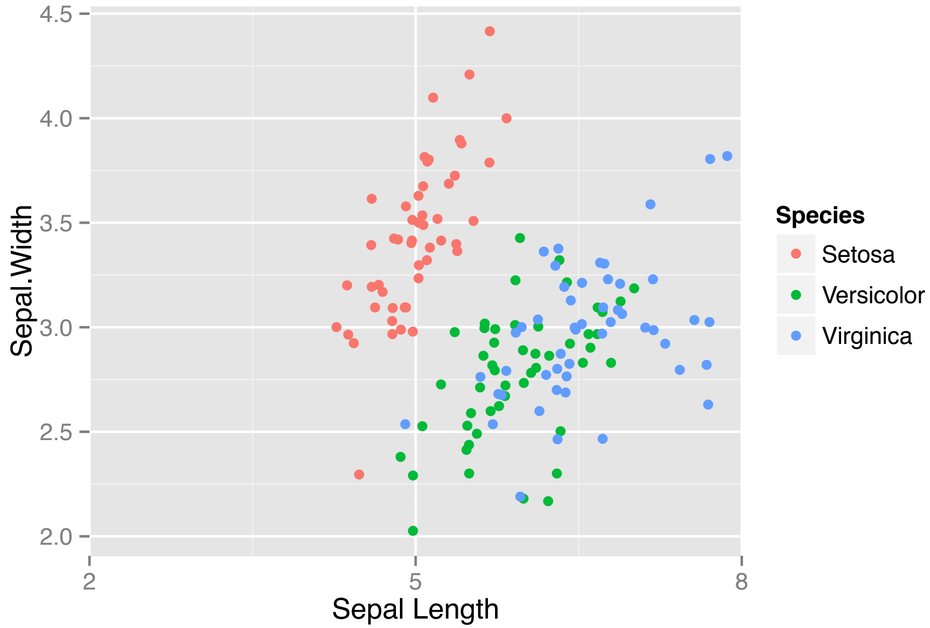
breaks

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point(position = "jitter") +
    scale_x_continuous("Sepal Length", limits = c(2, 8),
                       breaks = seq(2, 8, 3)) +
    scale_color_discrete("Species")
```



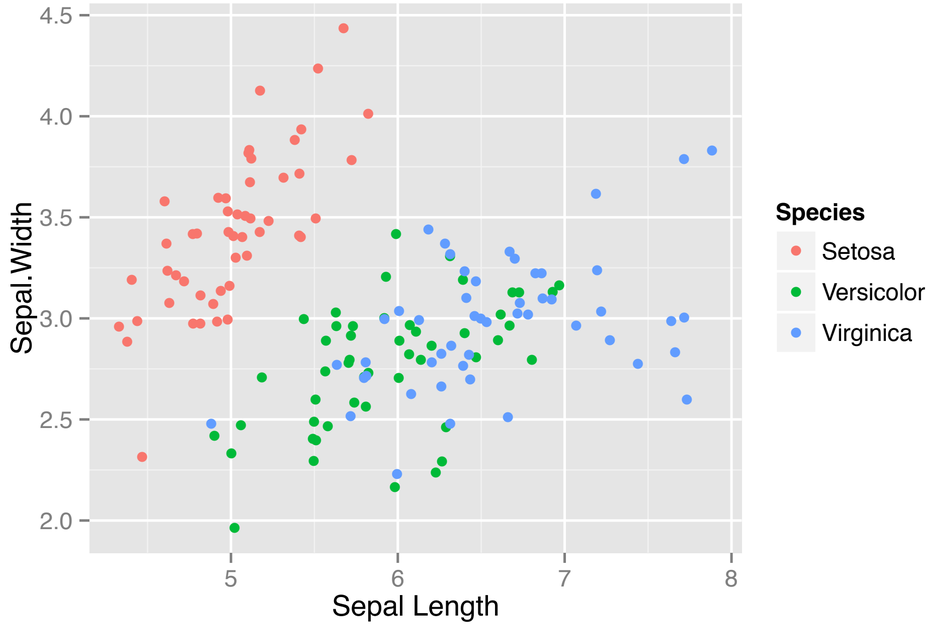
expand

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point(position = "jitter") +
    scale_x_continuous("Sepal Length", limits = c(2, 8),
                       breaks = seq(2, 8, 3), expand = c(0, 0)) +
    scale_color_discrete("Species")
```



labels

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +
    geom_point(position = "jitter") +
    scale_x_continuous("Sepal Length", limits = c(2, 8),
                       breaks = seq(2, 8, 3), expand = c(0, 0)) +
    scale_color_discrete("Species",
                         labels = c("Setosa", "Versicolour", "Virginica")))
```



labs

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