GGPLOT AND VISUALIZATIONS

Tobias Niedermaier

GGPLOT2

WHAT IS GGPLOT2?

- **ggplot2** is a powerful data visualization package in R.
- It implements the *Grammar of Graphics*, allowing users to build complex plots from simple components.
- Part of the `tidyverse`.
- Install `ggplot2`

```
1  install.packages("ggplot2")
```

- load library in your script

```
1  library(ggplot2)
```

GENERAL SYNTAX

```
ggplot(data = ..., mapping = aes(...)) +
  geom_... +
  ... +
  ...
```

- `data`: the function expects a data frame
- `mapping(...)` Aesthetic mappings: Defines the variables that are mapped to certain visual properties with a function `aes(...)`
- `geom_...`: Geometric objects defining the type of plot

A FIRST EXAMPLE: SCATTERPLOT

- We use `iris` data set as an working example

```
1  library(ggplot2)
2  data("iris", package = "datasets")
3  head(iris)
```

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

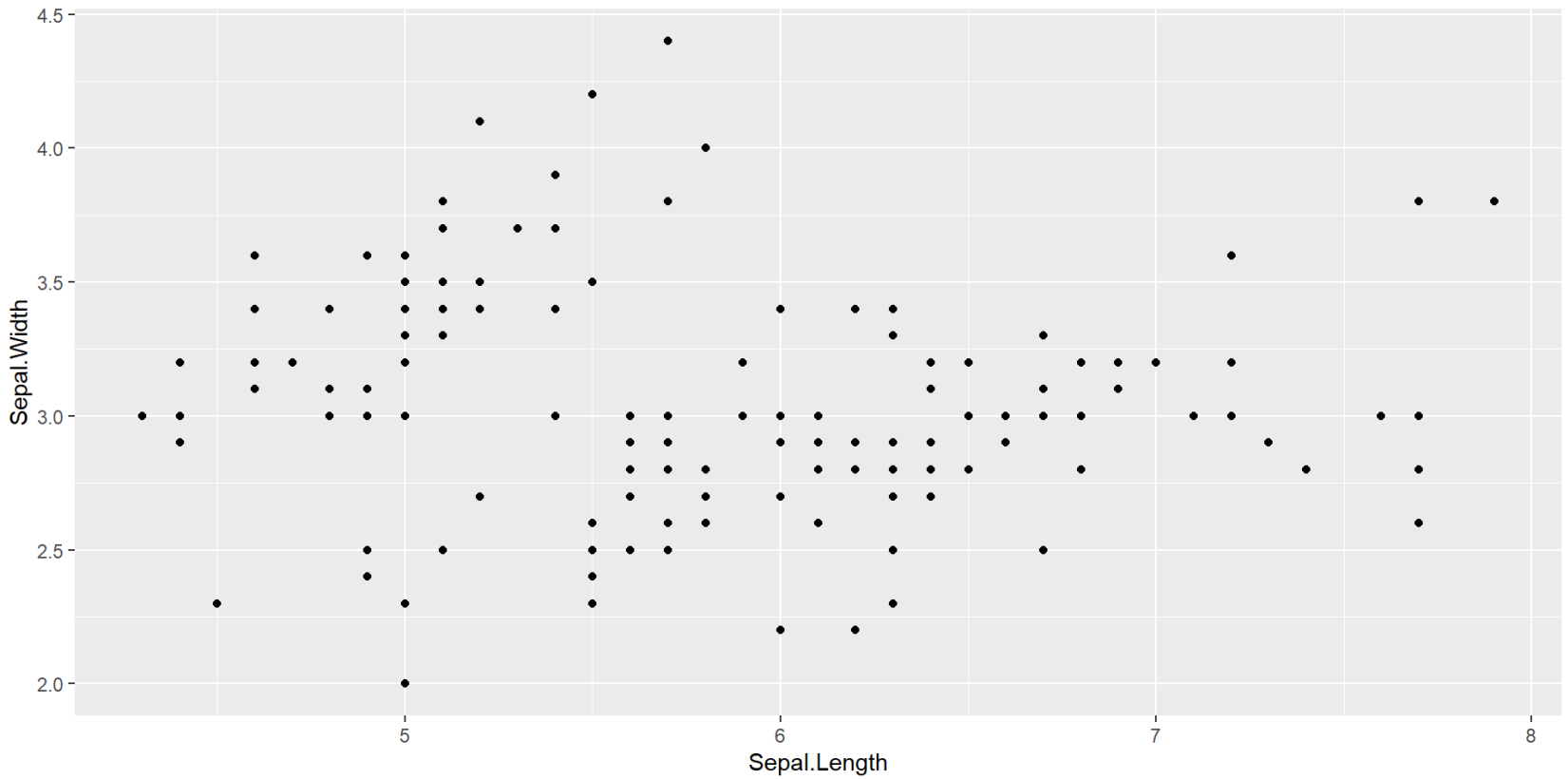
- A scatterplot can be specified using `geom_point()`

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

- Note that the variable names are *not* in quote marks. Call them as they are actual objects.

A FIRST EXAMPLE: SCATTERPLOT

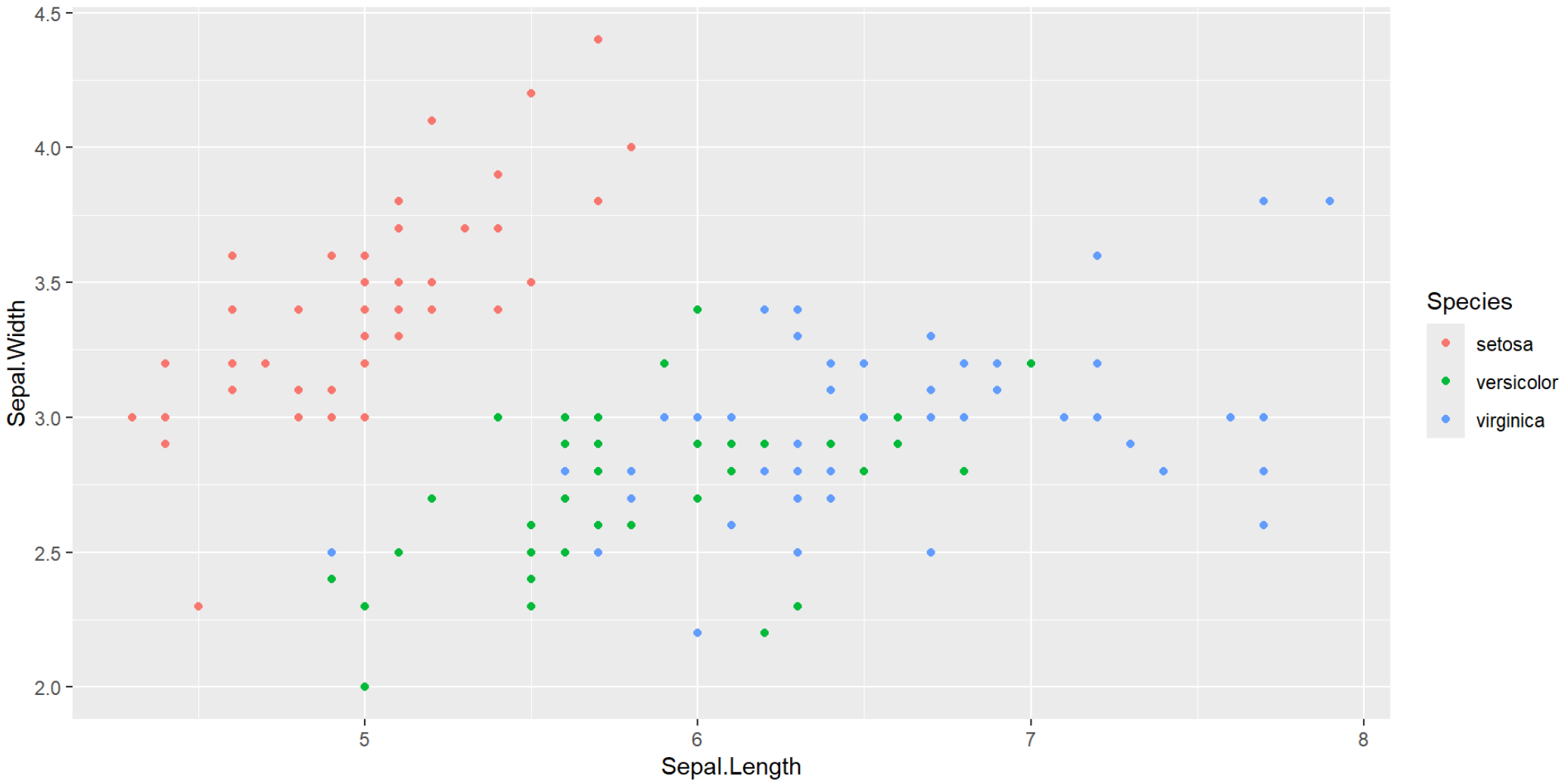
```
1  ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))+
2    geom_point()
```



- Note that the variable names are *not* in quote marks. Call them as they are actual objects.

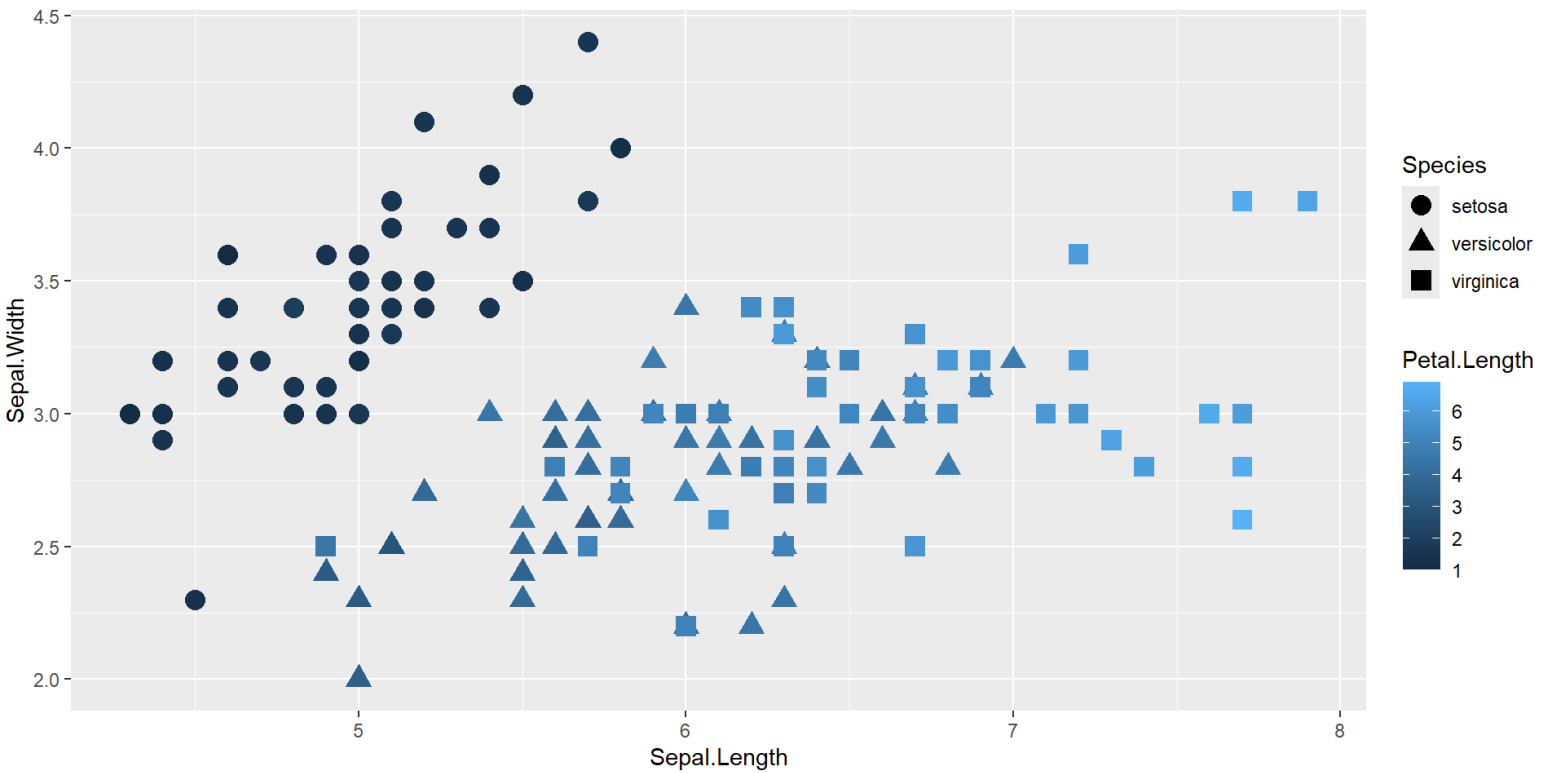
ADDING ANOTHER AESTHETIC MAPPING

```
1  ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species))+
2    geom_point()
```



ADDING COLOR FOR CONTINUOUS DATA AND SHAPE

```
1  ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width,
2                          color = Petal.Length, shape = Species))+
3    geom_point(size = 4)
```



- We added a `size` argument to the geom_point-function to make the points larger

EXERCISES 3 TASKS 1

LINES

We define a simple function

$$f(x) = sin(x) + cos(x \cdot 0.5)$$

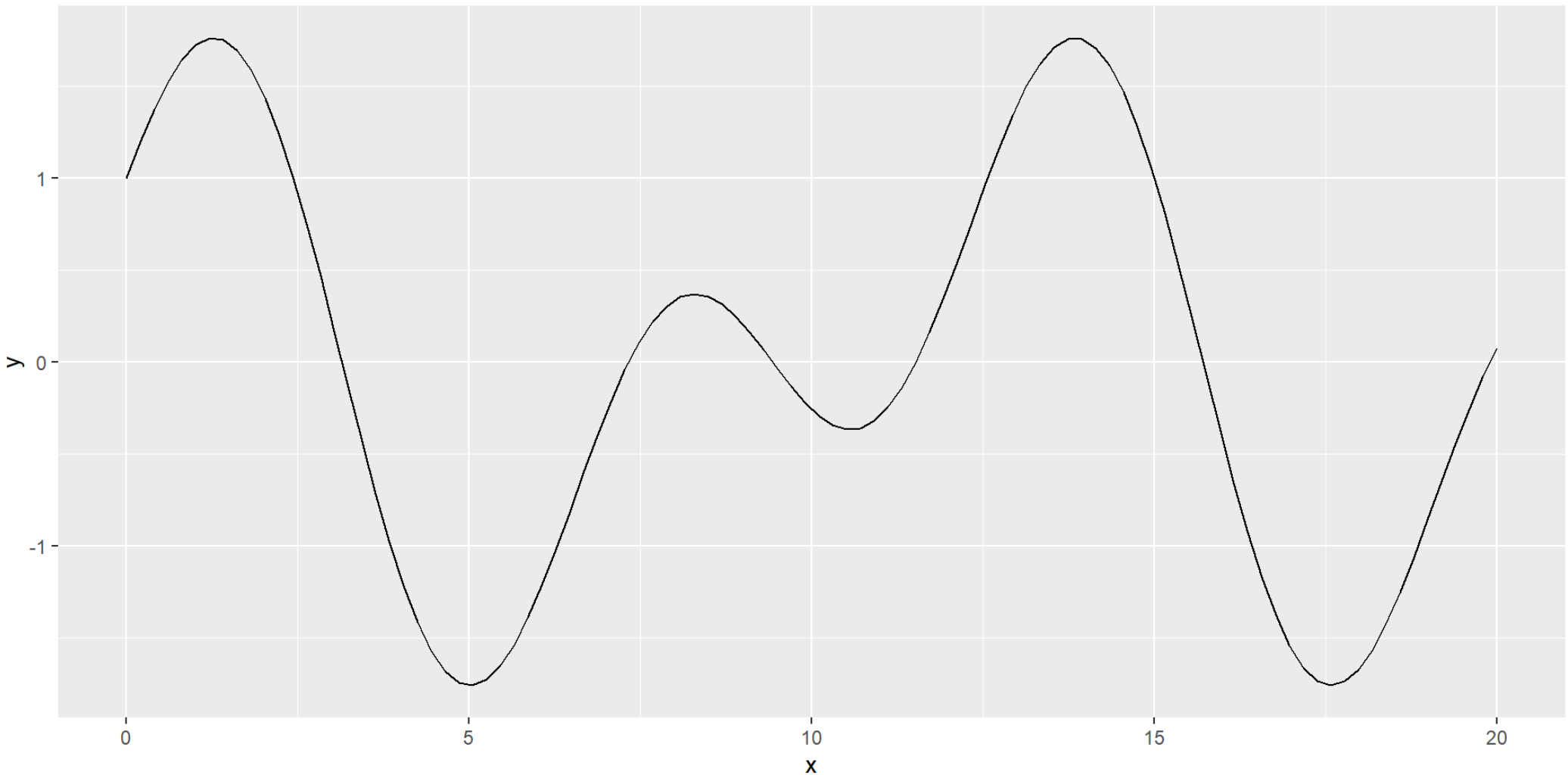
```
foo <- function(x) sin(x) + cos(x*0.5)
x <- seq(0, 20, len = 100)
y <- foo(x)
```

- We now deliberately ignore the `data` argument in ggplot and just define x and y.

```
ggplot(mapping = aes(x = x, y = y))+
  geom_line()
```

LINES

```
1  ggplot(mapping = aes(x = x, y = y))+
2    geom_line()
```



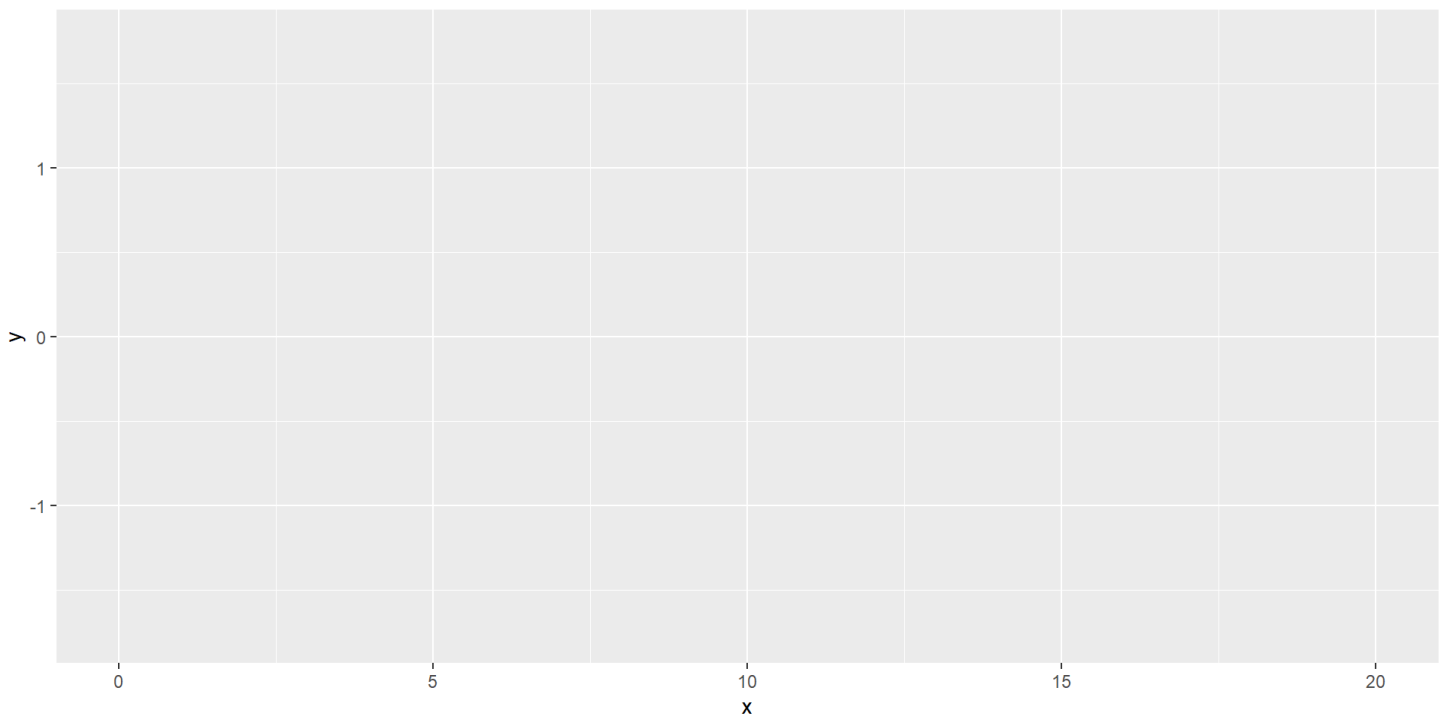
GGPLOT OBJECTS

- We can assign the ggplot as an object…

```
1  g <- ggplot(mapping = aes(x = x, y = y))
```

- and look at it:

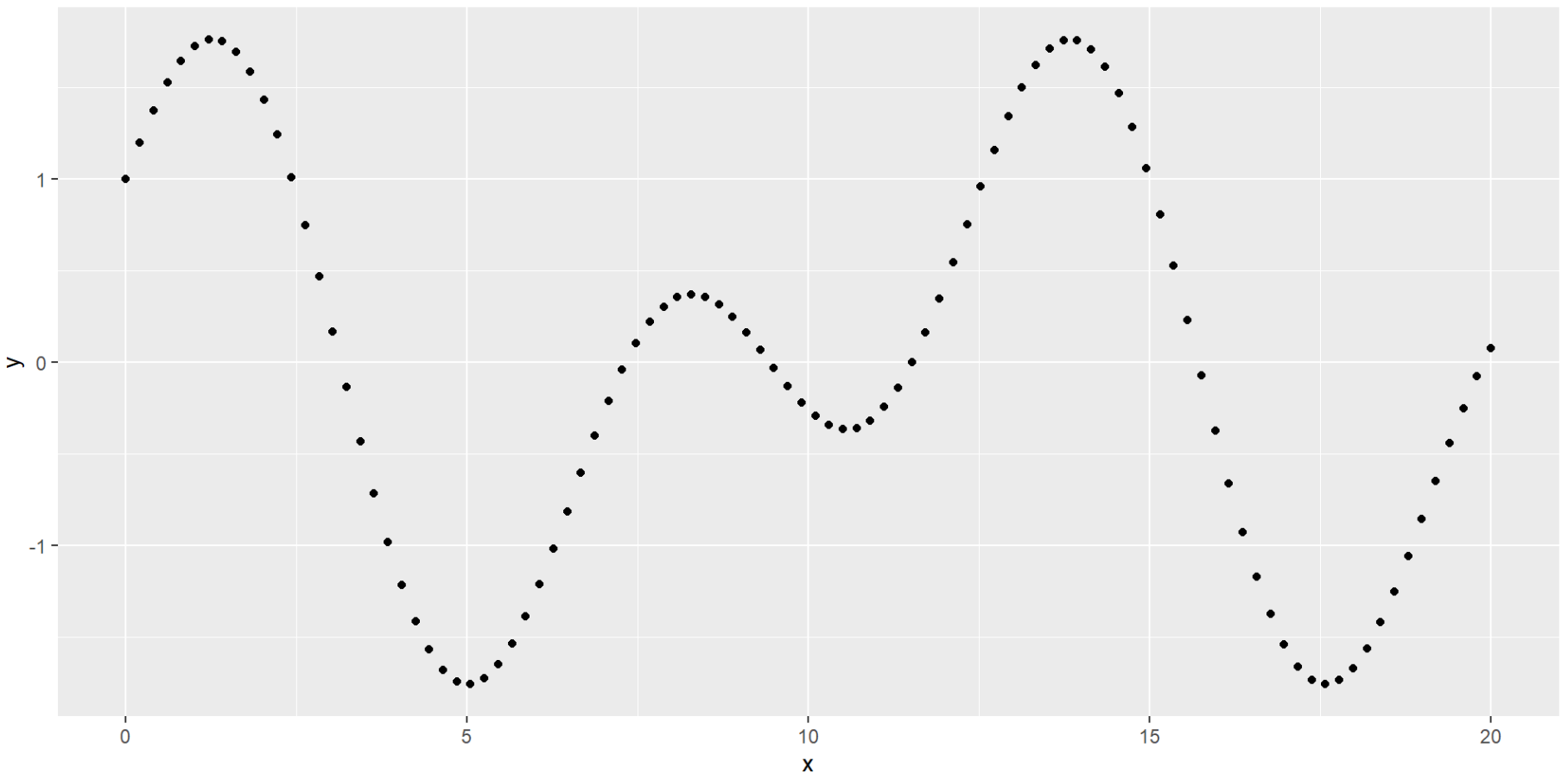
```
1  g
```



GGPLOT OBJECTS

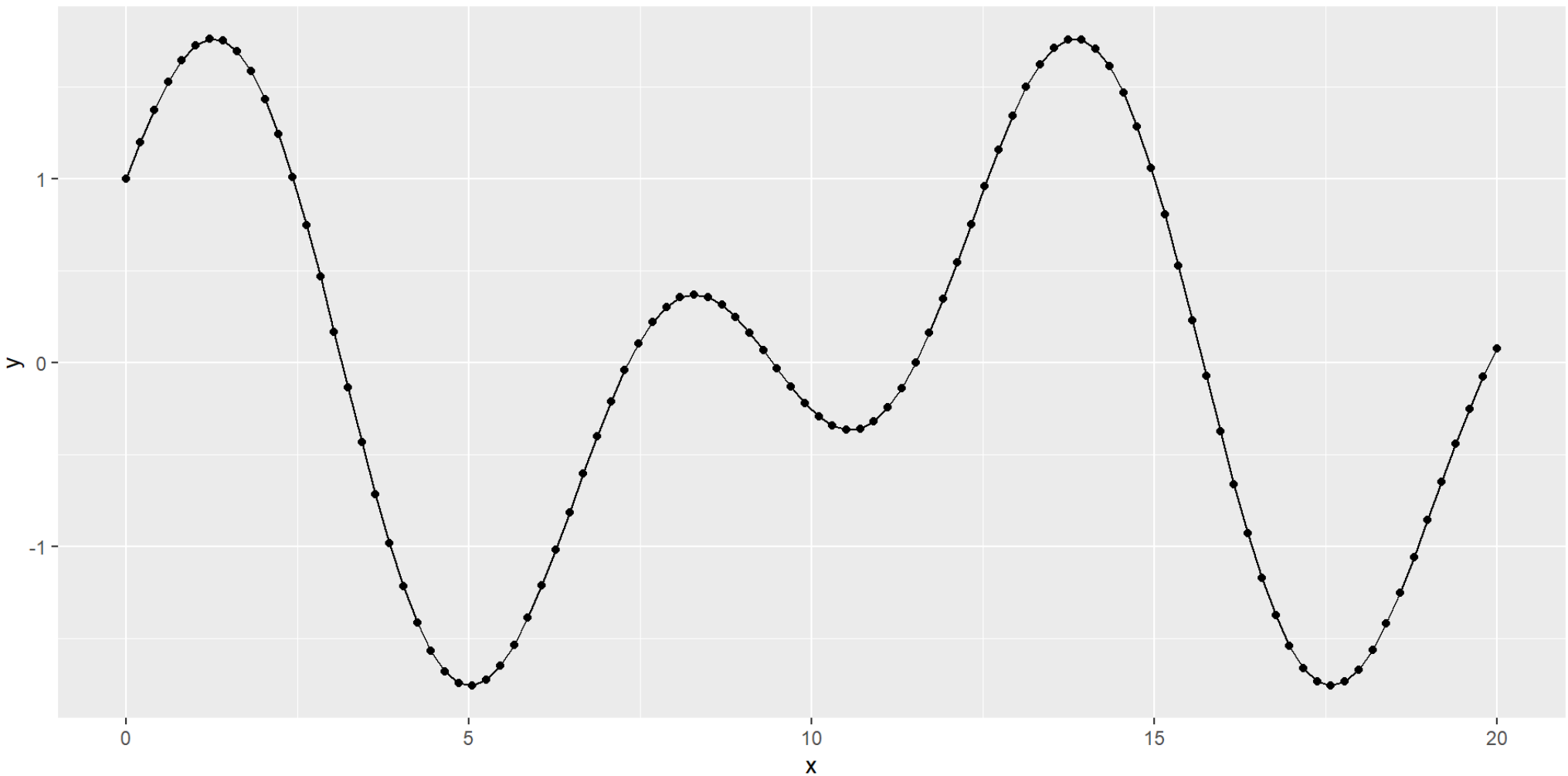
- adding layers later to an object:

```
1  g +
2    geom_point()
```



ADD MULTIPLE LAYERS

```
1  g +
2    geom_point()+
3    geom_line()
```



SUBPLOTS

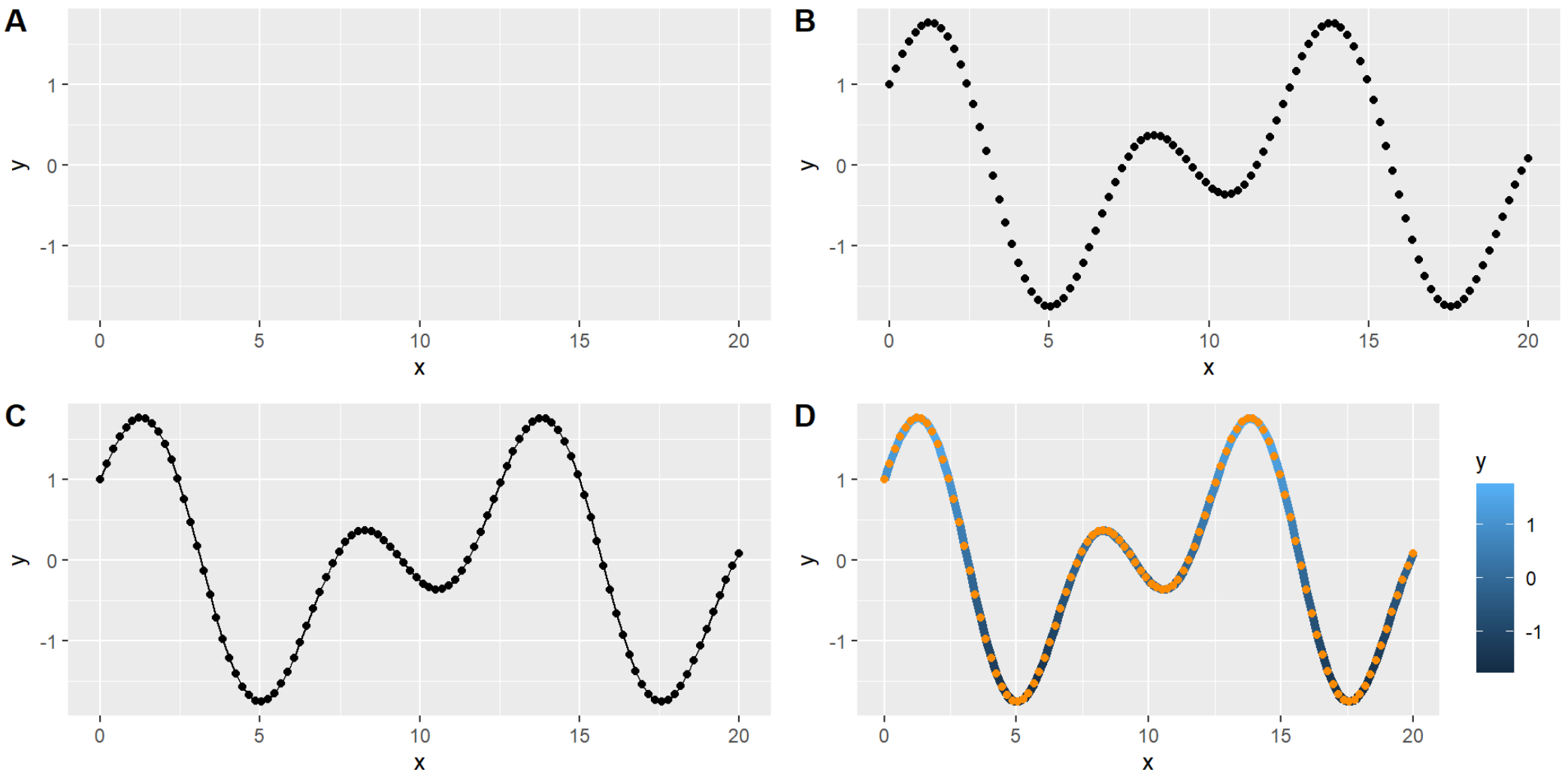
Multiple plots can be designed using external packages. Here, we use `cowplot`.

```
 1  library(cowplot)
 2
 3  # assign two objects
 4  g_point <- g +
 5    geom_point()
 6
 7  g_point_line <- g +
 8    geom_point()+
 9    geom_line()
10
11  g_point_line_color <- g +
12    geom_line(aes(color = y), linewidth=2)+
13    geom_point(color = "darkorange")
14
15  plot_grid(g, g_point, g_point_line, g_point_line_color,
16            nrow = 2, ncol = 2,
17            labels="AUTO")
```

Note that we have different color arguments:

- In line 12 *inside* `aes(...)` with a variable name
- In line 13 *outside* of `aes(...)`
- Control line width accordingly using `linewidth` (here: outside `aes(...)`)

SUBPLOTS



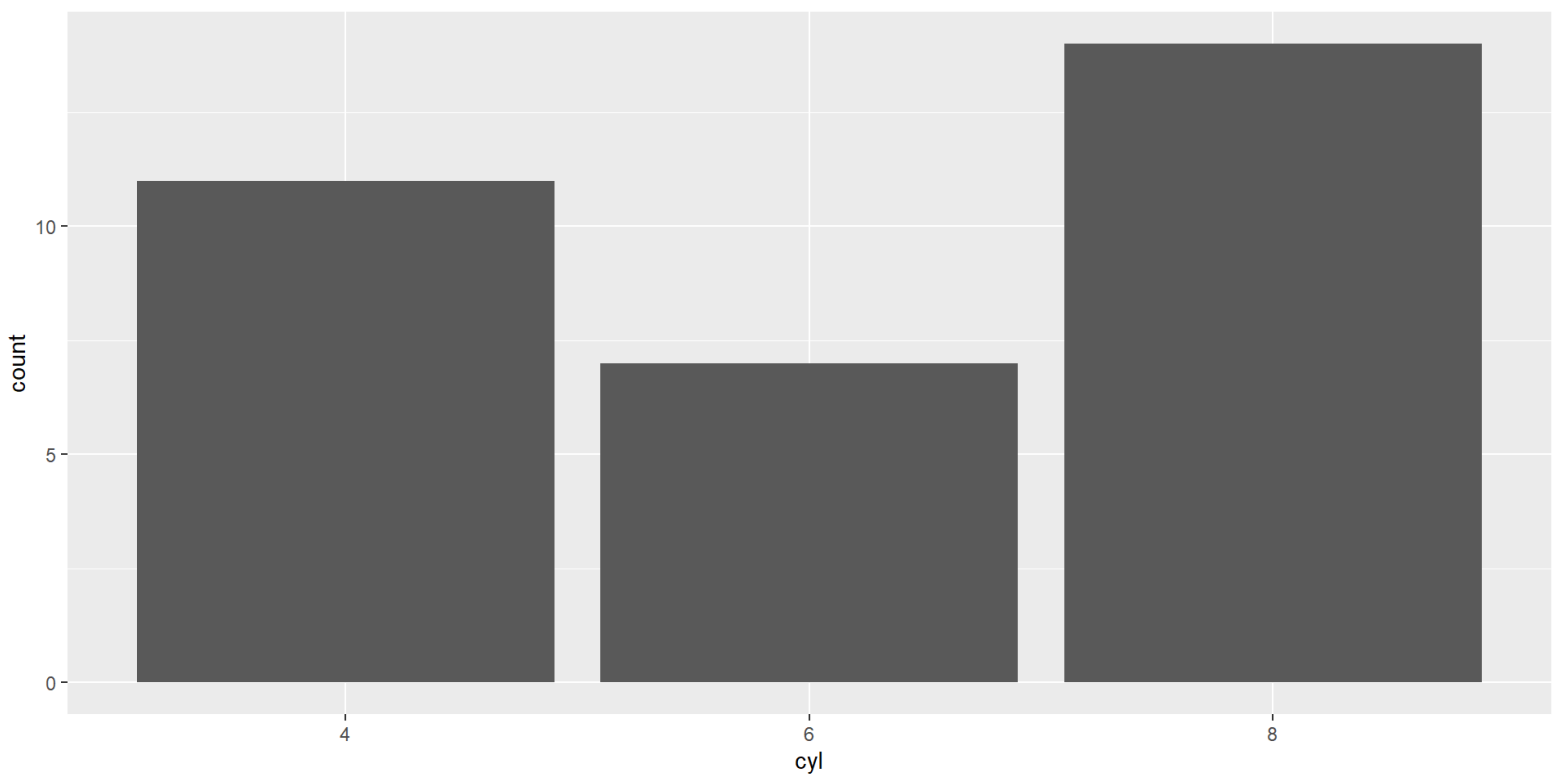
EXERCISES 3 TASKS 2

OTHER TYPES OF PLOT

BARPLOT

The syntax stays the same for a type of plots. - A barplot only requires aesthetics for x. - We use the `mtcars` data set as an example

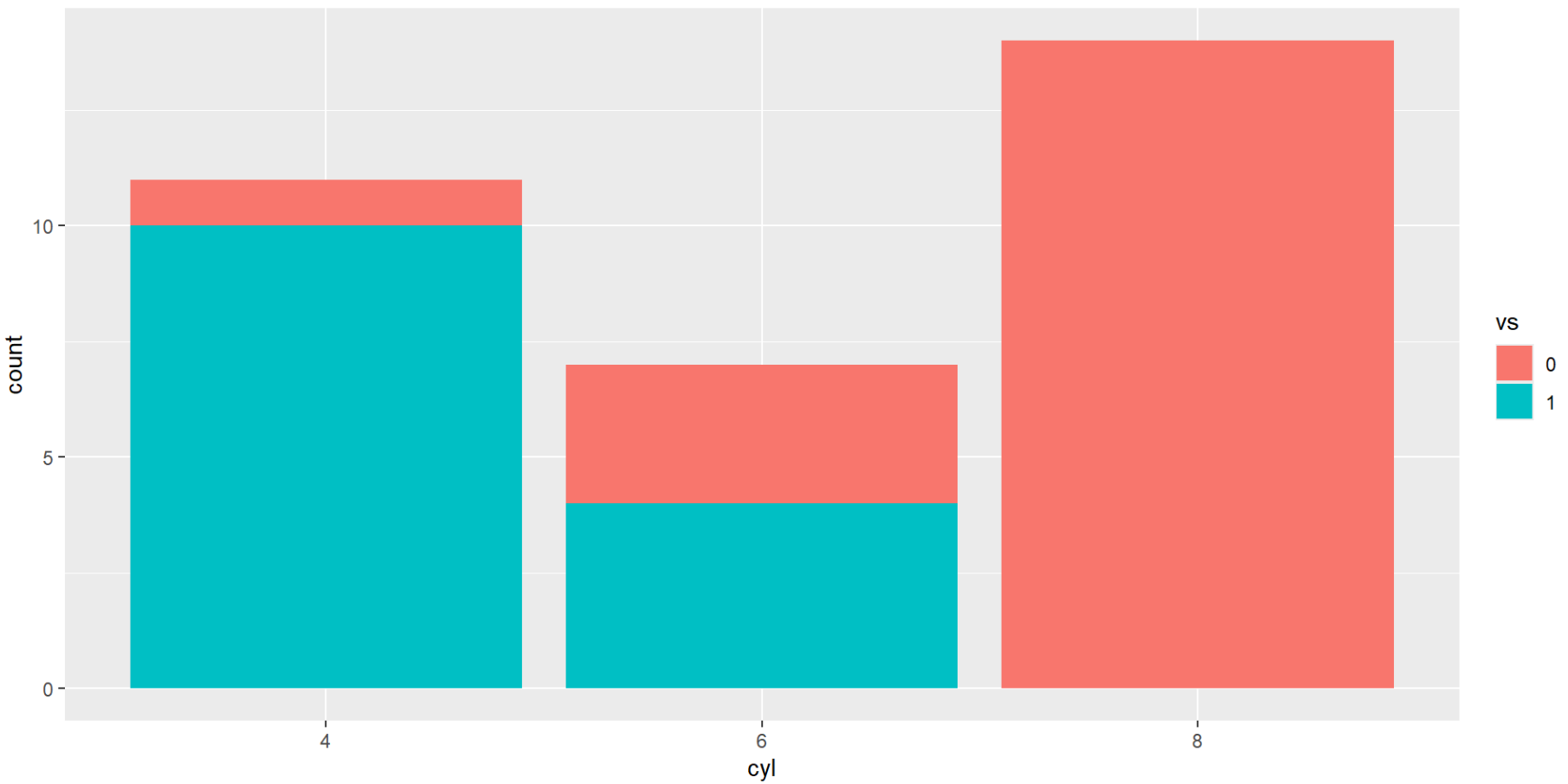
```
1  data <- mtcars
2  data$cyl <- as.factor(data$cyl)
3  ggplot(data, aes(cyl))+
4    geom_bar()
```



ADD COLOR

Use `fill` instead of color here.

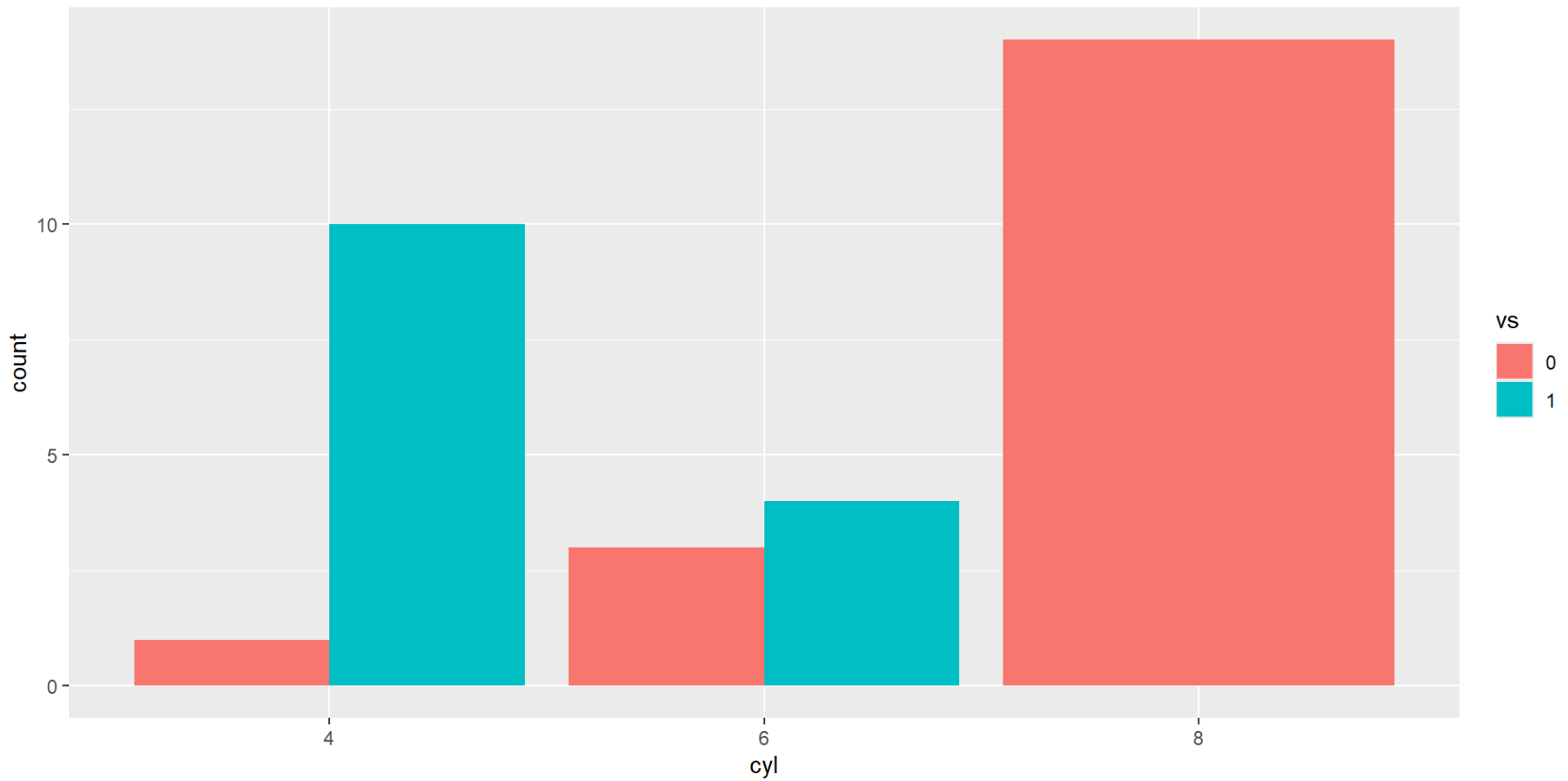
```
1  data$vs <- as.factor(data$vs)
2  ggplot(data, aes(cyl, fill = vs))+
3    geom_bar()
```



ADD COLOR

Side by side:

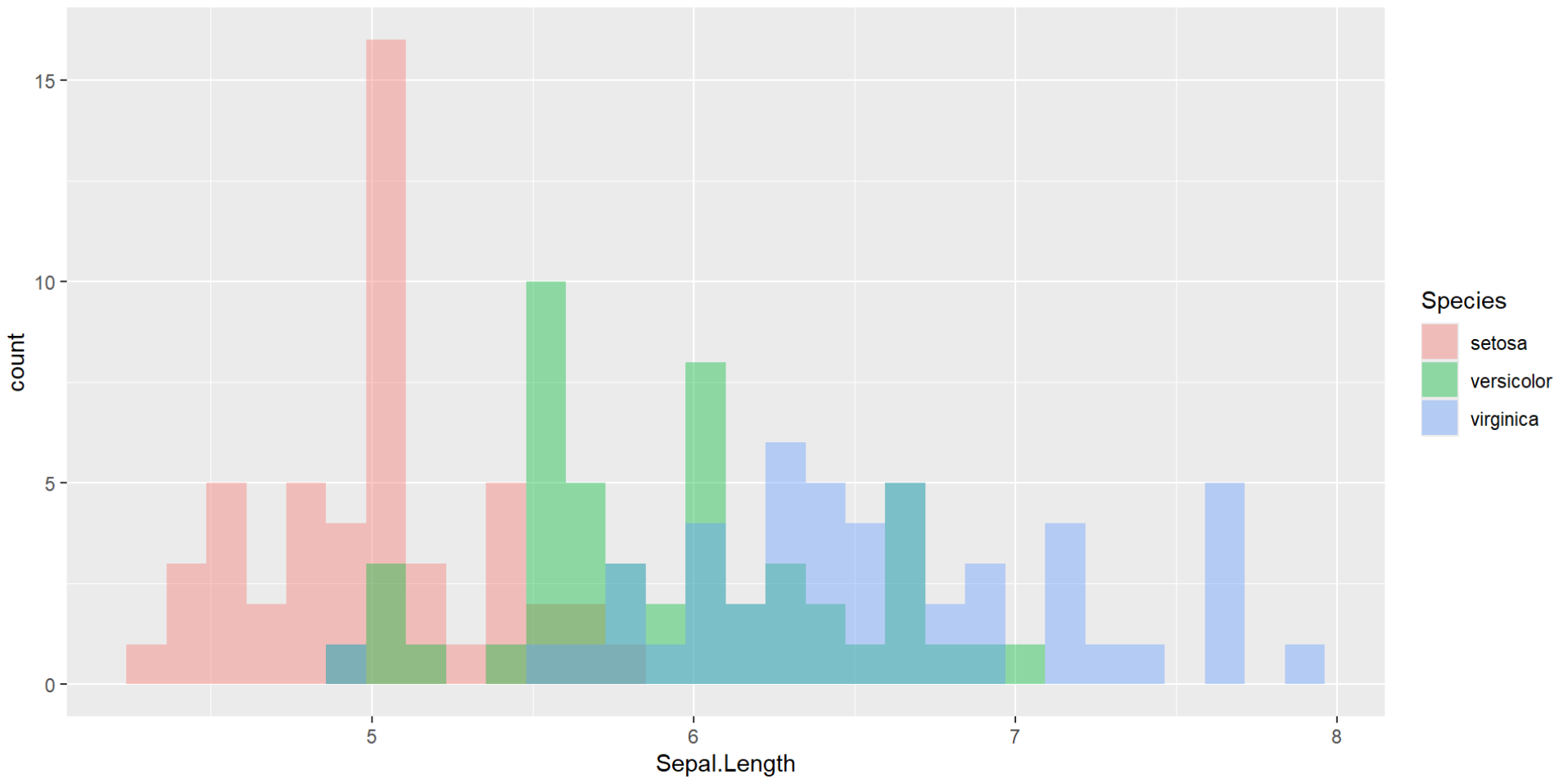
```
1  data$vs <- as.factor(data$vs)
2  ggplot(data, aes(cyl, fill = vs))+
3    geom_bar(position = "dodge")
```



HISTOGRAM

Here, we use `iris` again. - `position = "identity"` to overplot histograms

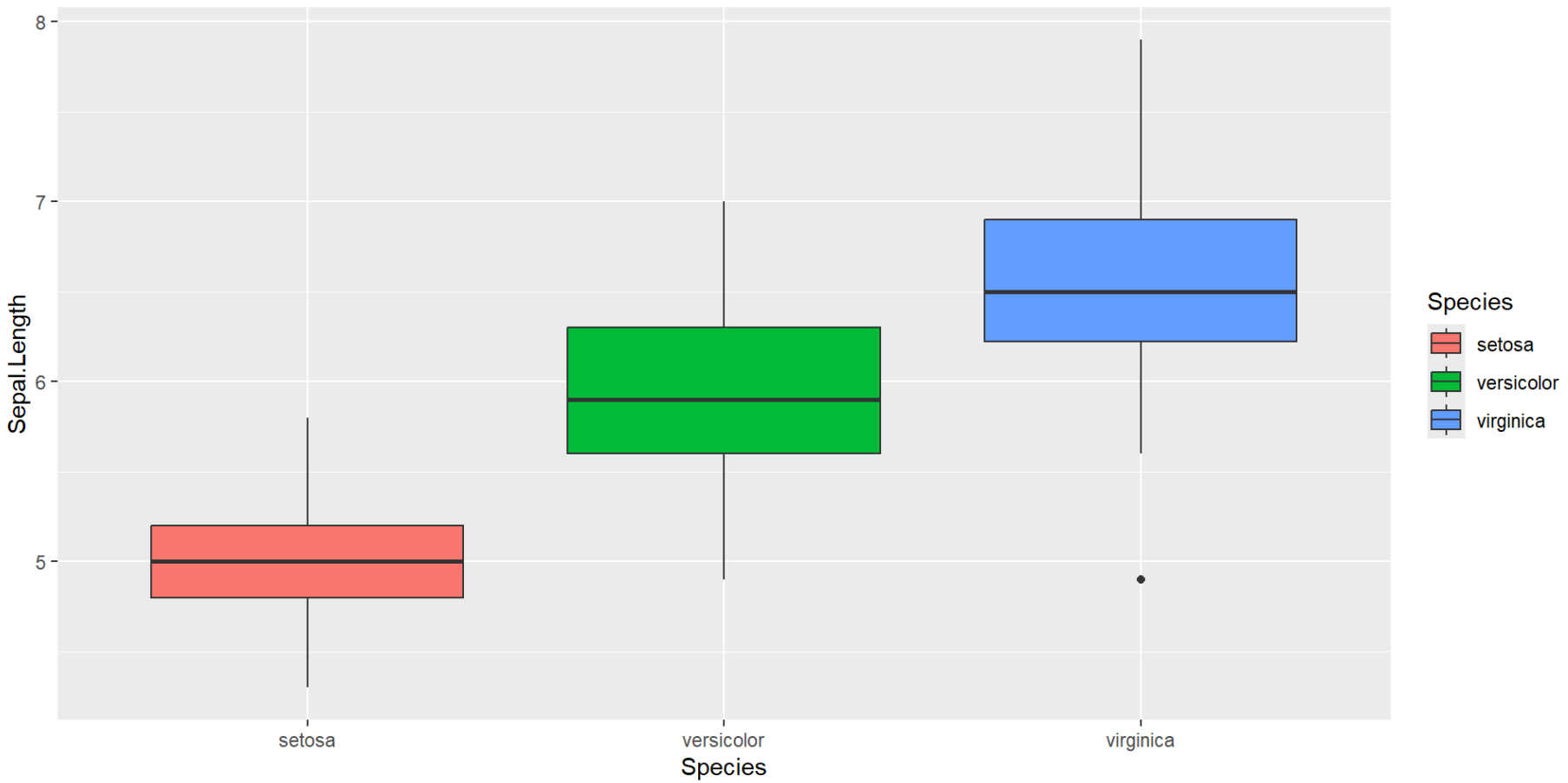
```
1  ggplot(iris, aes(Sepal.Length, fill = Species))+
2    geom_histogram(bins = 30, alpha = 0.4, position = "identity") # alpha for transparency
```



BOXPLOT

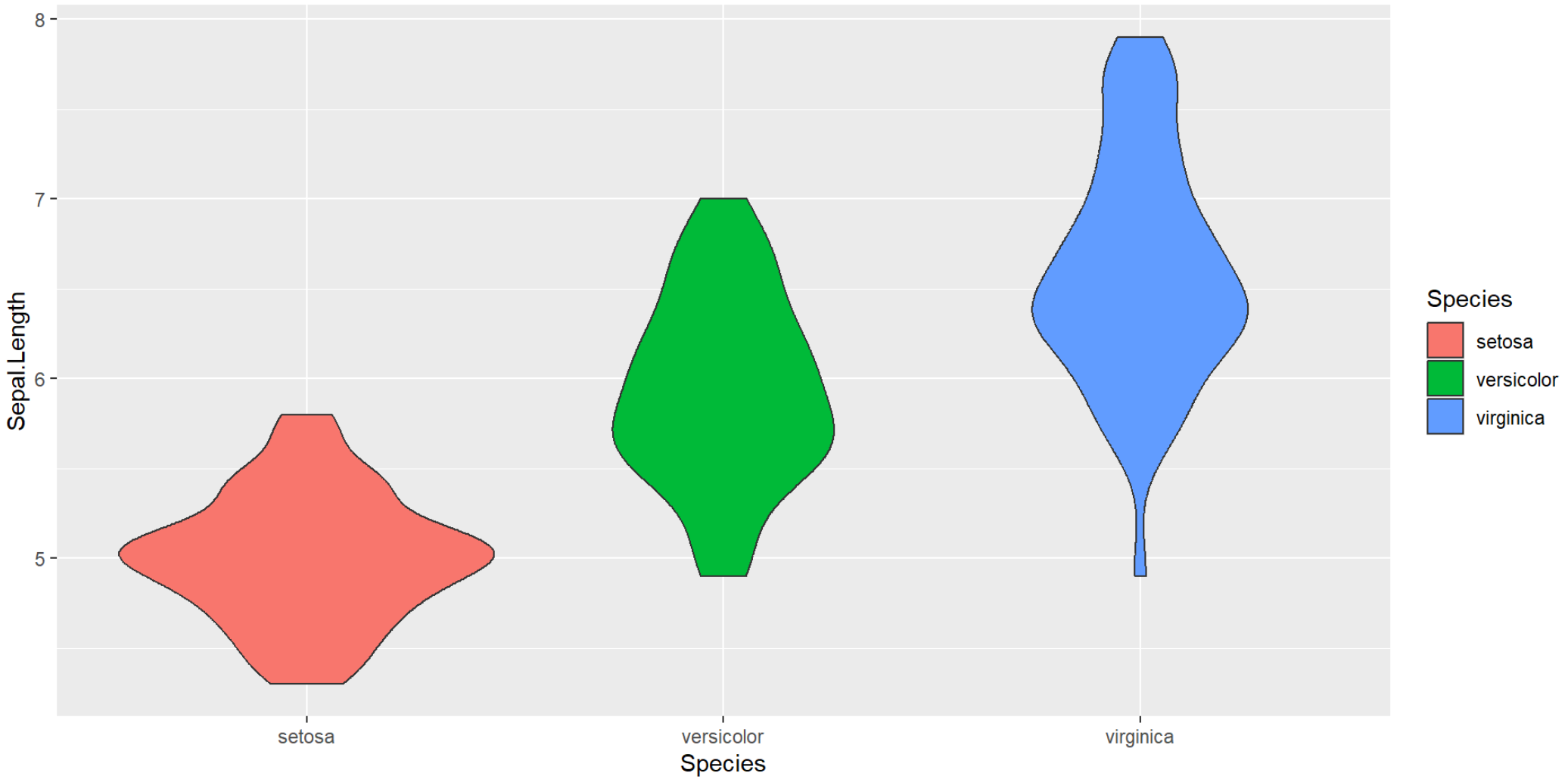
- Note that we have `Species` on the x-axis *and* as fill color

```
1  ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species))+
2    geom_boxplot()
```



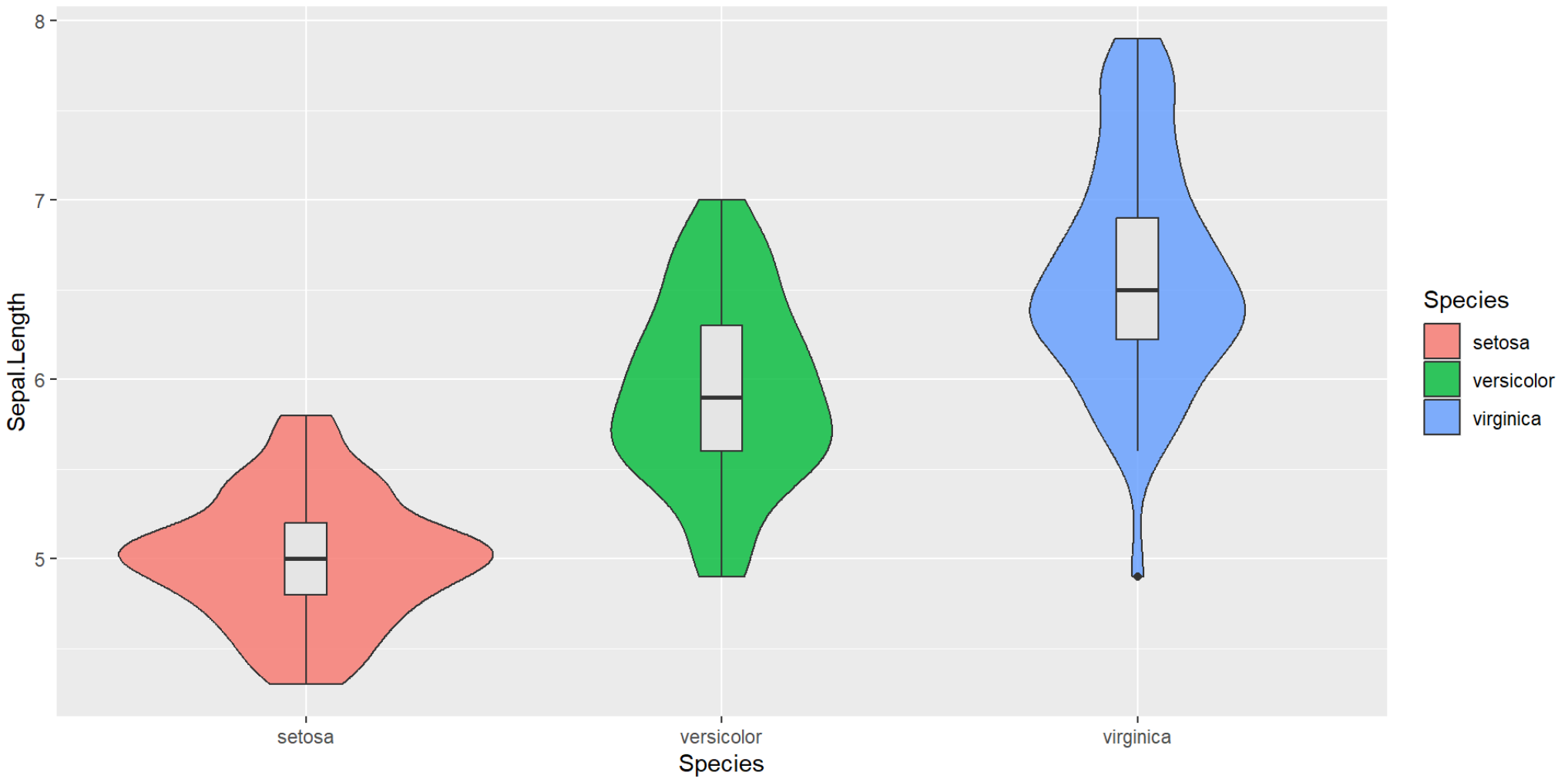
VIOLIN

```
1  ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species))+
2    geom_violin()
```



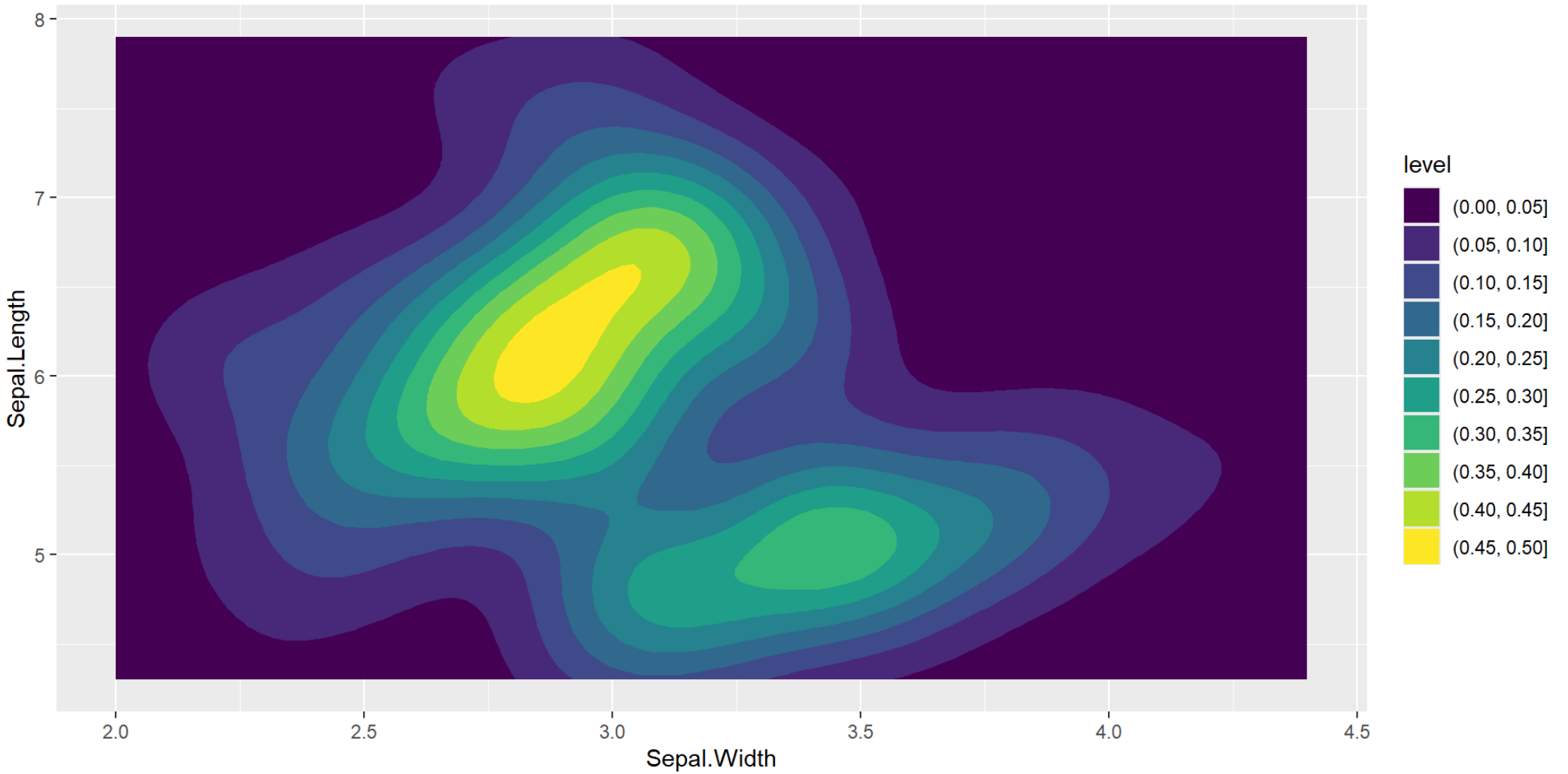
COMBINATION

```
1  ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species))+
2    geom_violin(alpha = 0.8)+
3    geom_boxplot(width=0.1, fill="grey90")
```



2-DIM DENSITY

```
1  ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length))+
2    geom_density2d_filled()
```



EXERCISES 3 TASKS 3