

# 変態統計学1～ggplotを用いたグラフィカルデータ分析～

教授：shichi / 所属：cg研・midi研・プログラム研・統計班・電子煙草班・水煙草班

単位：★★★★☆ 内容：★☆☆☆☆

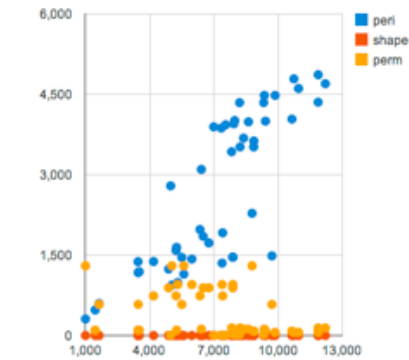
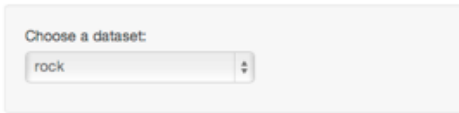
◆単位は来るが教授は変態。むしろこっちが聴講料を払ってもらいたいくらいである。出席はとるが最後の試験の日に全裸で参戦するだけでA+をくれる。ただし男子生徒の場合は容赦なくFを付けられるので注意が必要。

## はじめに

本資料はR言語を少しかじった程度の人、もしくは恐ろしく暇な変態がRを使ってグラフィカルにデータを解析してみようという試みの元制作されました。制作に当たったのはペテン師のshichiです。プログラマー歴は一年も無いです。以後お見知り置きを。

さて、ゲーム制作サークルと聞いてmis.wに入ってきた犠牲者の皆様。おめでとうございます。ついではかりに言っておきますとこのR言語、論文とかで使うことができますのでリアルな話役に立つと思います。少しだけ触ってみる事をおすすめします。それでは本編のほうに進んで参りましょう。

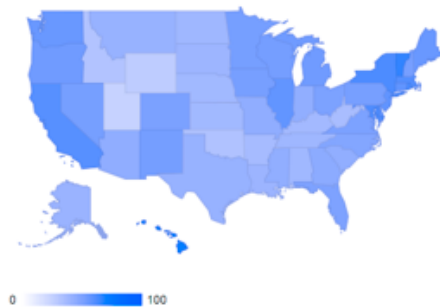
## R言語とは



Rank	Country	Population	% of World Population	Flag	Mode	Date
1	China	1339940000	19.5		✓	9 Oct 2010
2	India	1188650000	17.3		✓	9 Oct 2010
3	United States	310438000	4.52		✓	9 Oct 2010
4	Indonesia	237556363	3.46		✓	9 Oct 2010
5	Brazil	193626000	2.82		✓	9 Oct 2010
6	Pakistan	170745000	2.48		✓	9 Oct 2010
7	Bangladesh	164425000	2.39		✓	9 Oct 2010
8	Nigeria	158259000	2.3		✓	9 Oct 2010
9	Russia	141927297	2.06		✓	9 Oct 2010
10	Japan	127390000	1.85		✓	9 Oct 2010



Democratic share of the presidential vote in 2012



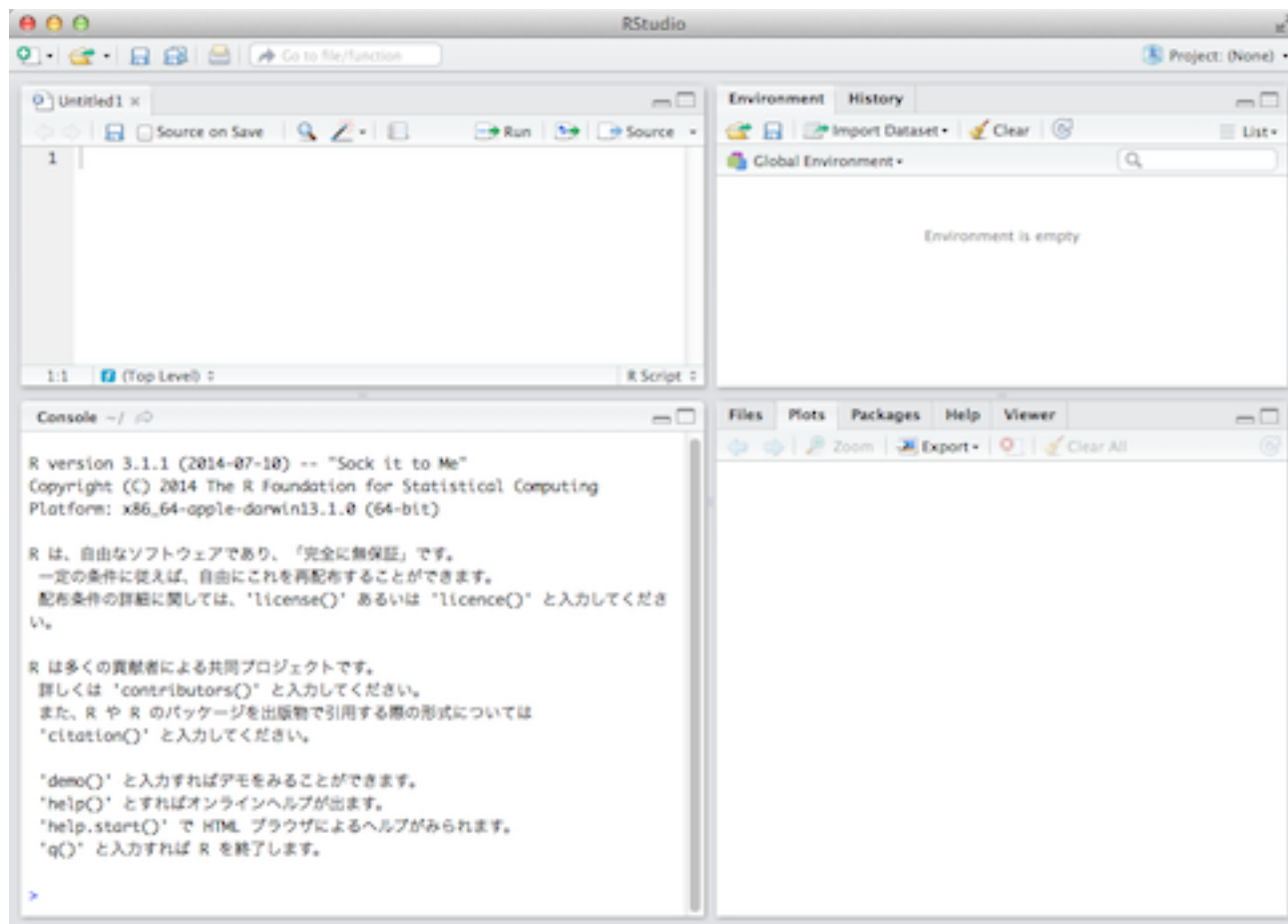
R言語は統計処理をメインとしたスクリプト言語です。似ている言語はpythonだそうです。僕はルーキーぷるぐらまなので知りません。

## 環境設定

まずはR言語 (<https://cran.rstudio.com/>)のインストールをしましょう。たぶんインストールとかは分かるので省略します。

次にRStudio (<https://www.rstudio.com/products/rstudio/download/>)という、便利にR言語を活用出来るIDEがあるので、この際インストールします。機能はFREE版で十分です。

インストールが終了したらRStudioを立ち上げて見てください。以下のような画面に成っているはずです。(初期設定をわすれたので実際どうなってるかは知りませんが適当にいじれば成るはずです)



左上のウィンドウでコードを書きます。書いたコードは同ウィンドウ右上のSourceボタンを押すことで走らせる事が出来ます。

右上のウィンドウは基本的には作成したオブジェクトの一覧が表示される場所です。

左下はコンソールになっていて実行結果によって数字がはき出されたり、また直接コードを書くことによってテスト実行的な事が出来るように成っています。

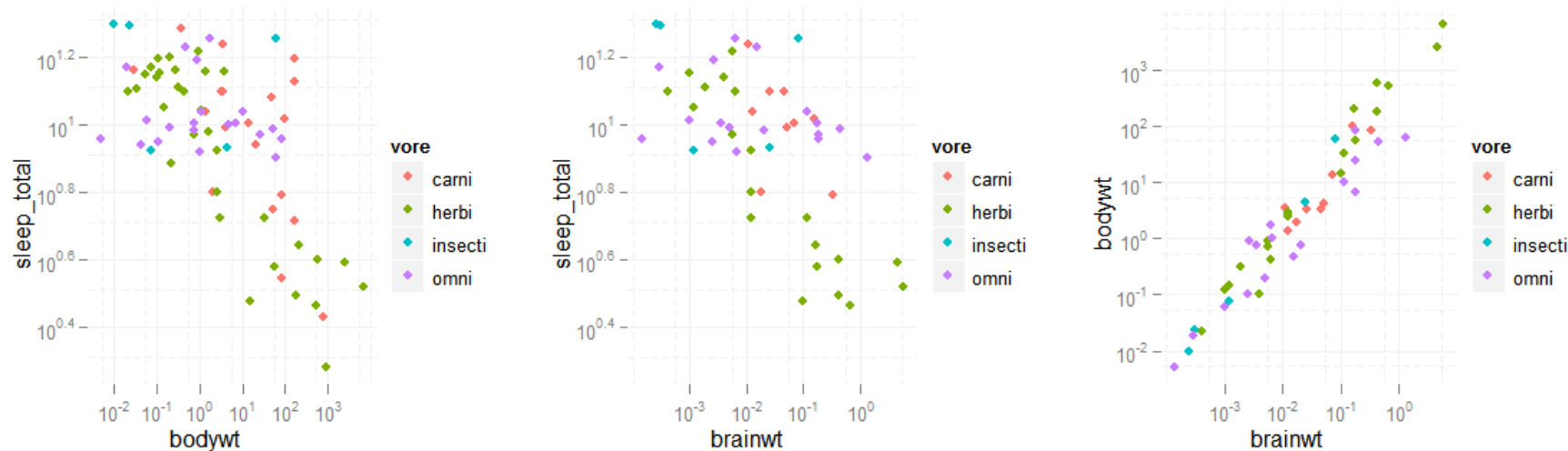
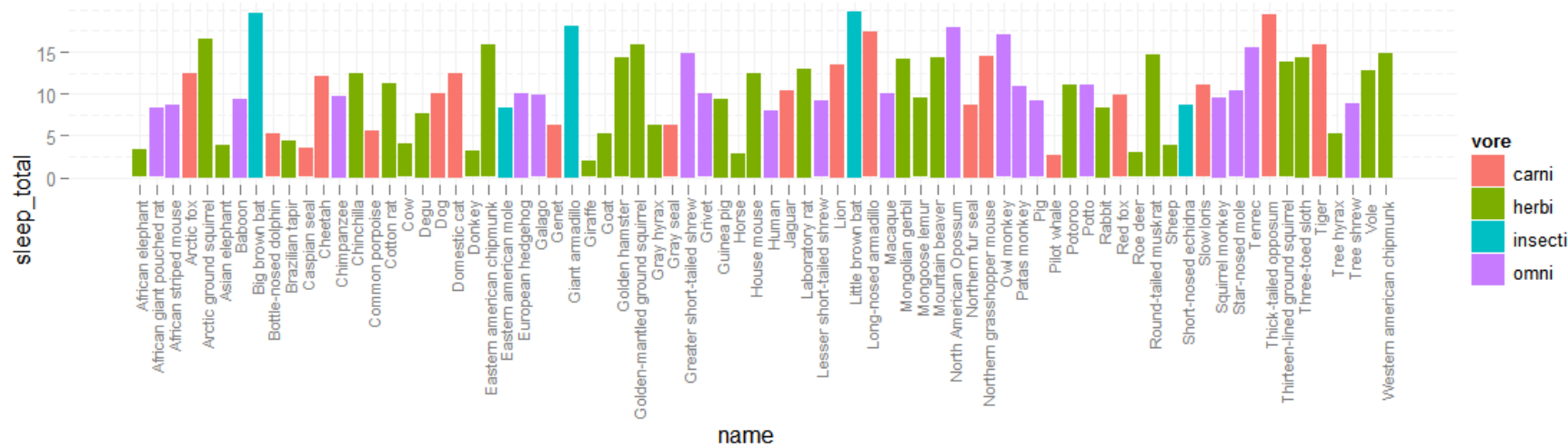
最後に右下のウィンドウではフォルダ構造や描画、リファレンス参照などを行うことが出来るように成っております。

これで準備は完了です。あなたはいつでもRのスクリプトを書くことが出来るし、お好きな統計処理をすることが出来ます。

# ggplotで遊ぼう！！！！

ggplotとは、R言語を用いて美しい図やグラフを簡単に実装できるライブラリである。

以下のような図を実装する事が出来ます



それでは早速なんかしらつくって見ましょう。

# 散布図 geom\_point()

```
#irisのサンプルデータを読み込みます
data(iris)
```

中身（一部）：

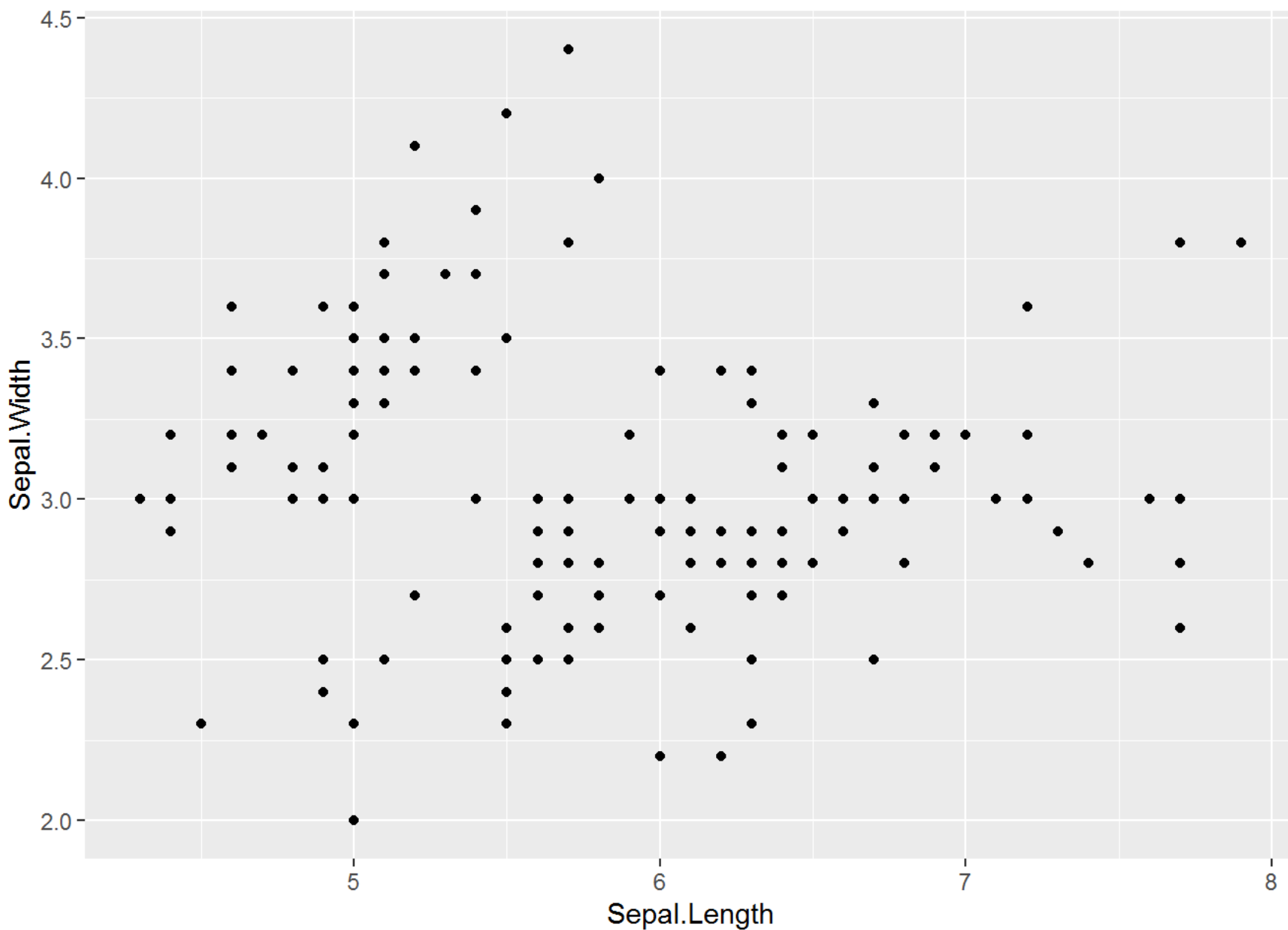
```
head(iris) #head() でデータフレームの一部を表示することができます
```

##	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

## 通常の散布図

```
#x軸をSepal.Length、y軸をSepal.Widthとした枠を作成します。
p <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width))

#上で作成したグラフ枠内にプロットしていきます。
p <- p + geom_point()
p
```

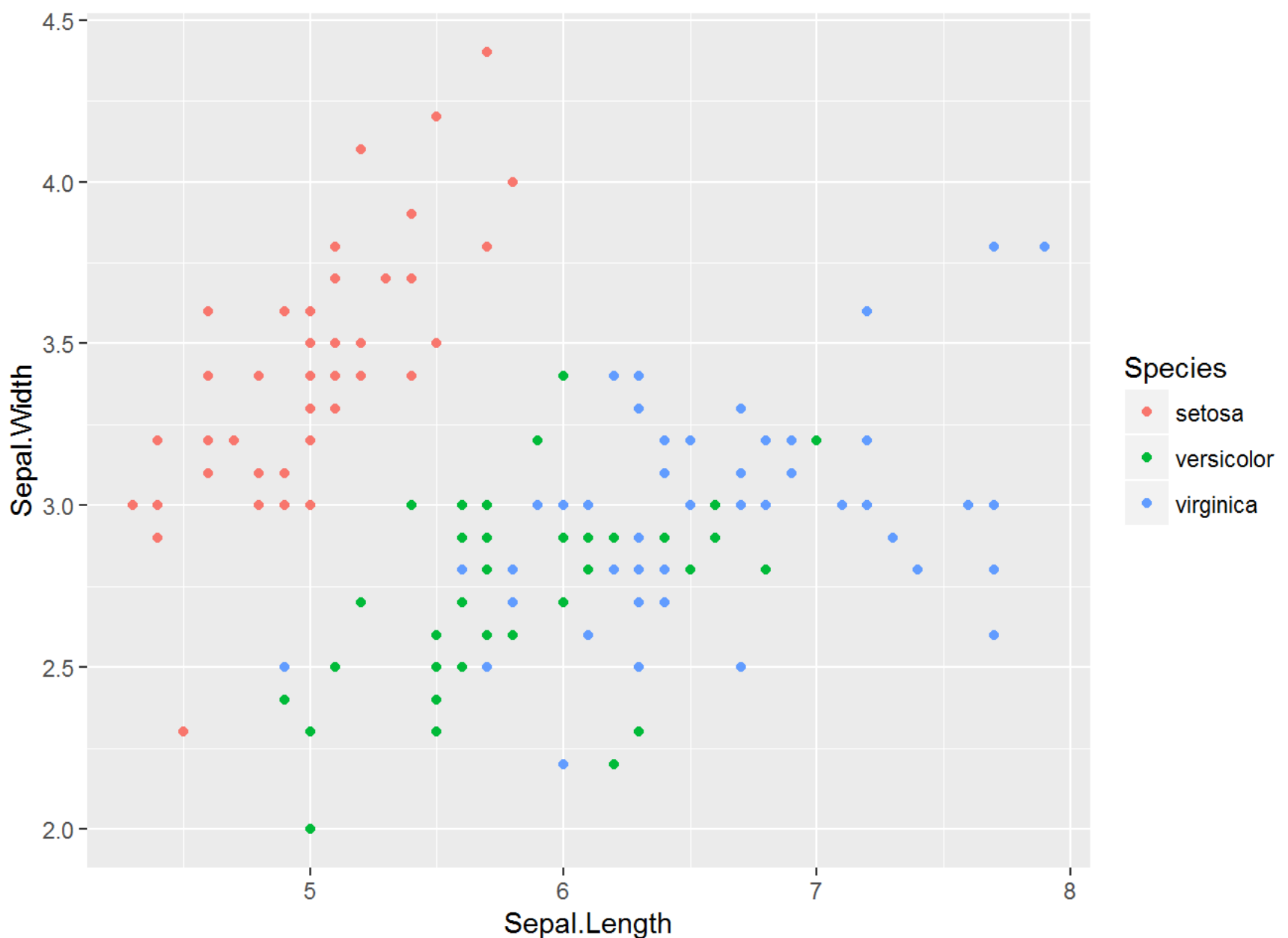


このように簡単に散布図を作成することが出来るのです。次はこの散布図をSpeciesごとに色分けをしてみることになります。

## 色分けされた散布図

```
#x軸をSepal.Length、y軸をSepal.Widthとした枠を作成します。
p <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width))

#上で作成したグラフ枠内にプロットしていきます。（Species毎にカラーリングします）
p <- p + geom_point(aes(colour=Species))
p
```



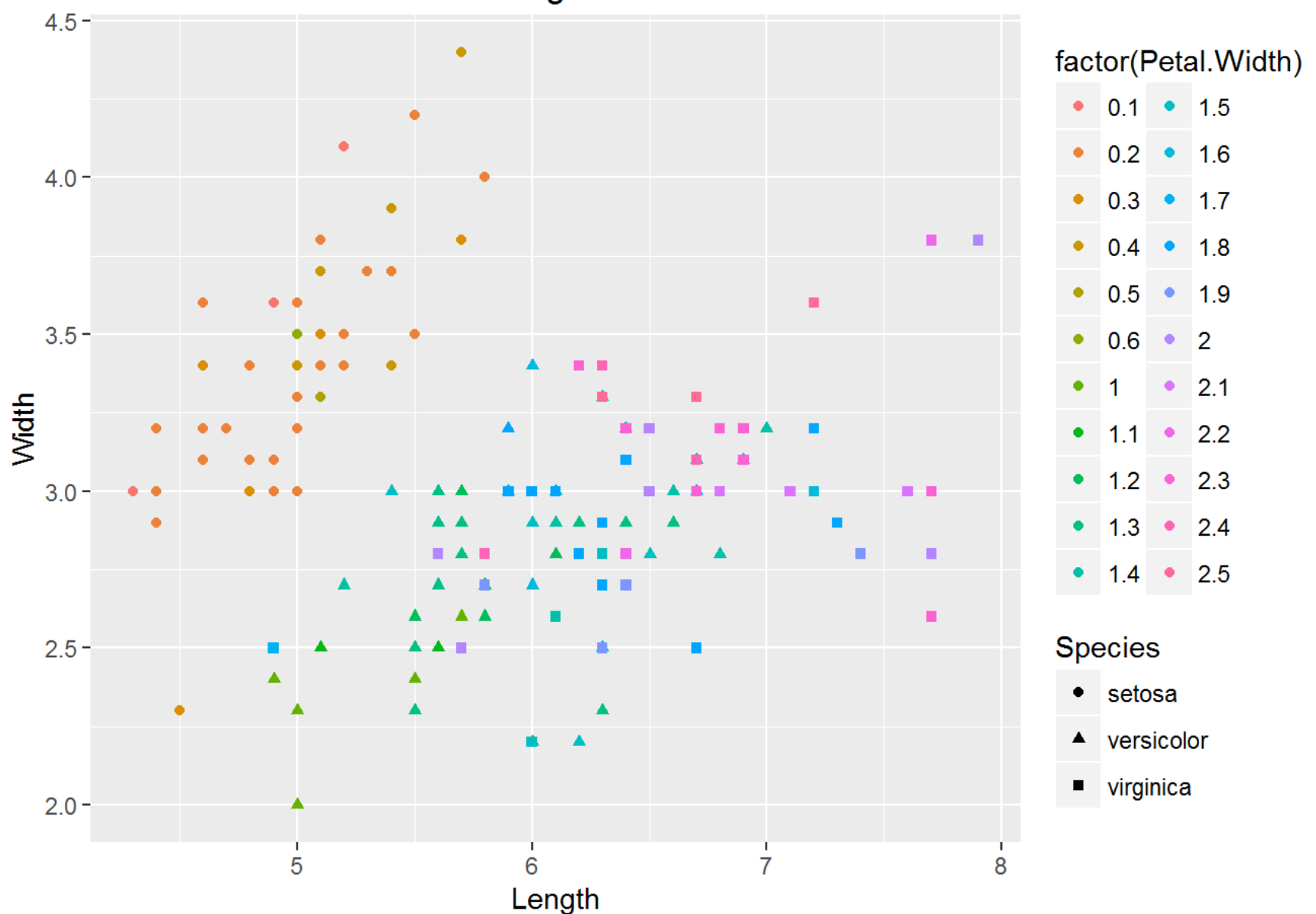
`geom_point()`のカッコのなかにカラー対応させたい属性を指定するだけで簡単に色分けまでできることがわかりました。このようにggplotにおけるグラフへの加筆、属性追加は基本的にはレイヤーの足し算のような雰囲気理解していただくと理解がスムーズになるような気がします。それではそれっぽい感じのグラフになるように一気に実装して見ましょう。

## 良い感じの散布図

```
#x軸をSepal.Length、y軸をSepal.Widthとした枠を作成します。
p <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width))

#上で作成したグラフ枠内にプロットしていきます。
p <- p + geom_point(aes(colour=factor(Petal.Width), shape=Species))+ ggtitle("IRIS
Length/Width") + xlab("Length") + ylab("Width")
p
```

## IRIS Length/Width



ggtitle:タイトルを挿入します      xlab:x軸タイトルを挿入します      ylab:y軸タイトルを挿入します 図と見比べれば何となくわかると思います。このほかにもプロットの大きさや近似線などいろいろなものを指定出来ますので試してみてください。

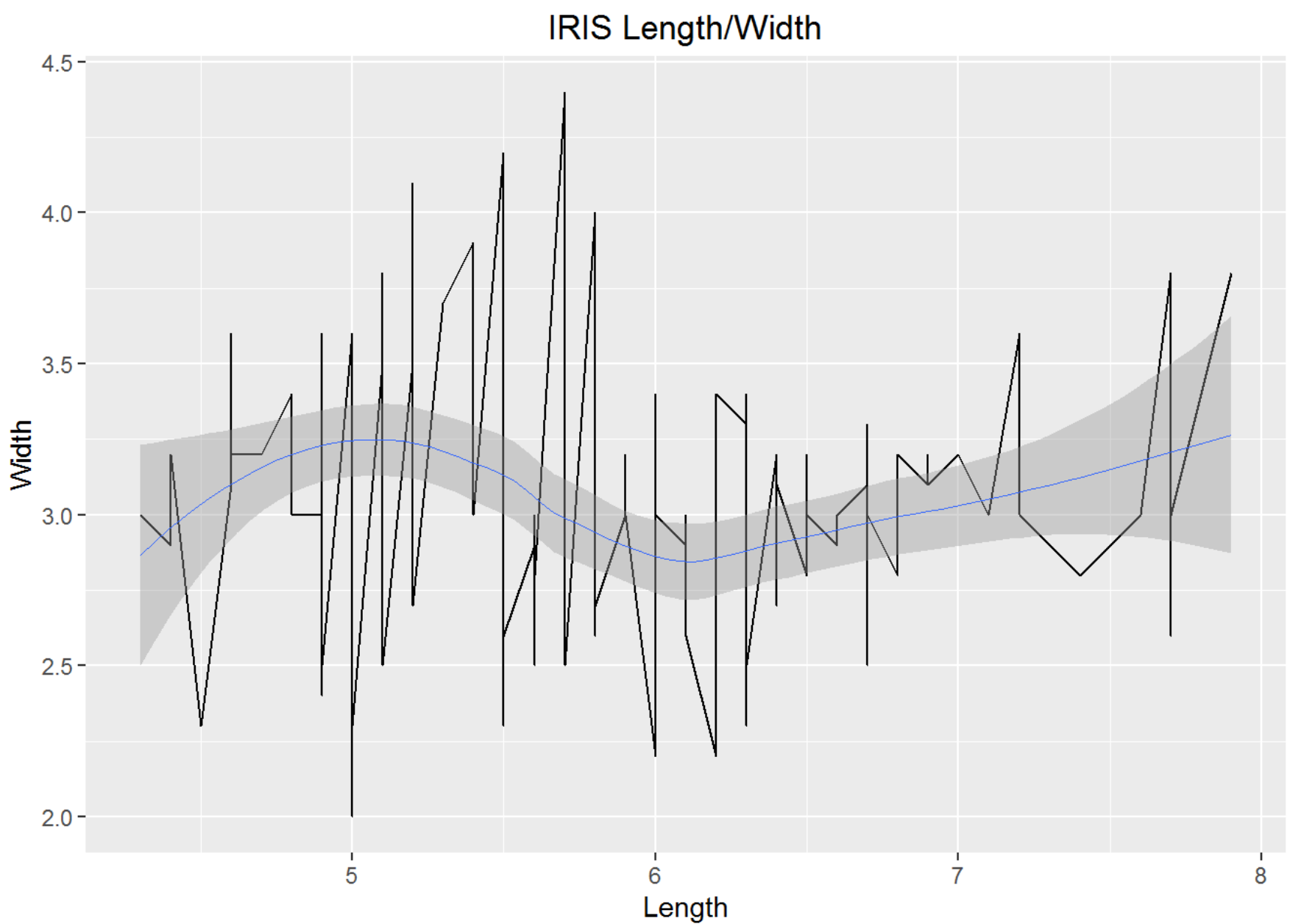
## 折れ線グラフ geom\_line()

### 近似線つき折れ線グラフ

引き続きirisのデータを使用します。前回のgeom\_point()をgeom\_line()に置き換えるだけで折れ線グラフにすることが出来ます。

```
#x軸をSepal.Length、y軸をSepal.Widthとした枠を作成します。
p <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width))

#上で作成したグラフ枠内にプロットしていきます。
p <- p + geom_line()+ ggtitle("IRIS Length/Width") + xlab("Length") + ylab("Width"
)
p <- p + geom_smooth (size = 0.1) #近似線を描く(sizeで線の太さの指定)
p
```



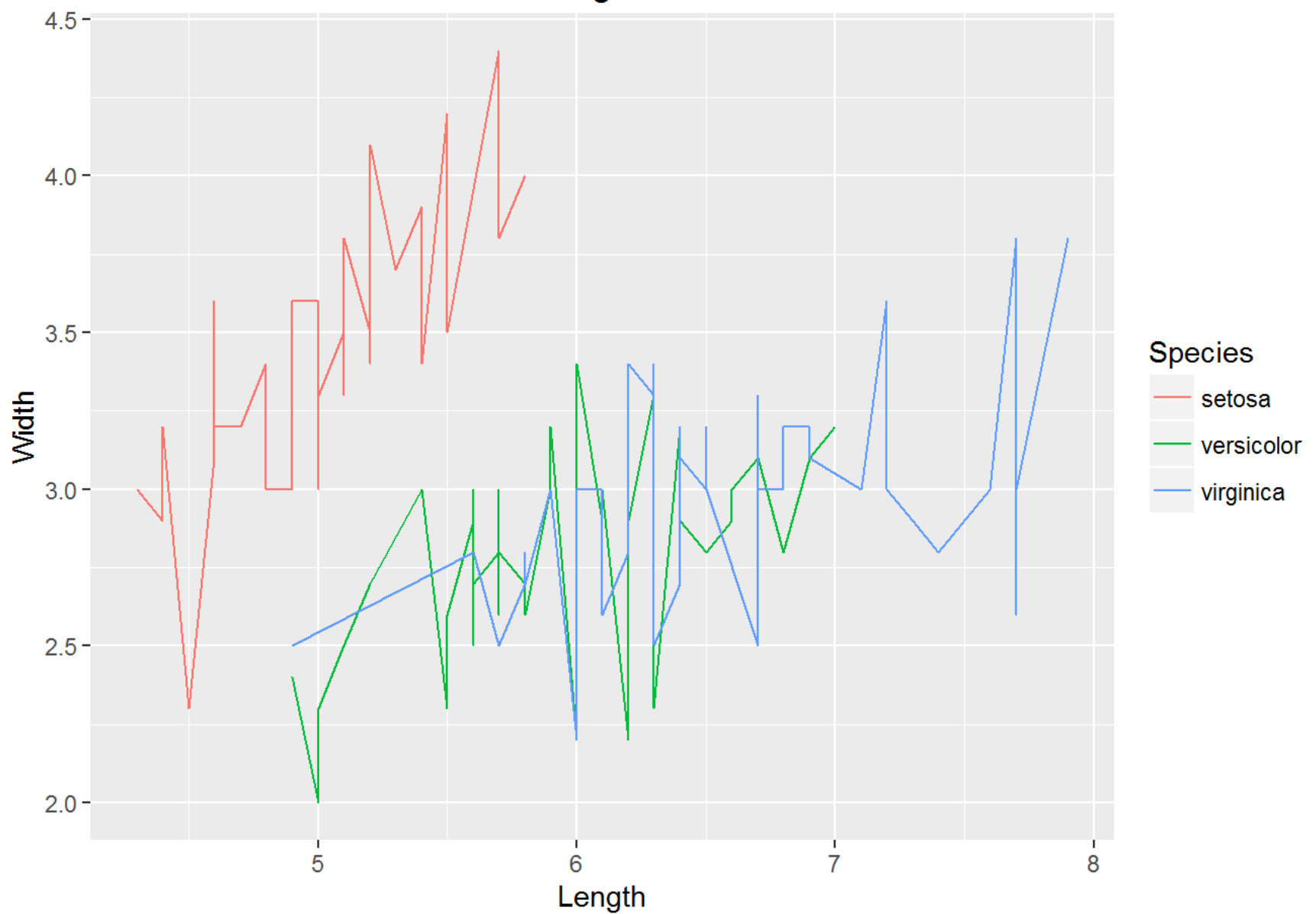
## 複数の折れ線グラフ

折れ線グラフを作成する際に系統ごとに折れ線を作成し比較したいことがあると思います。今回はそのようなグラフを作成します。

```
#x軸をSepal.Length、y軸をSepal.Widthとした枠を作成します。
p <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, group=Species, colour=Species
))

#上で作成したグラフ枠内にプロットしていきます。
p <- p + geom_line()+ ggtitle("IRIS Length/Width") + xlab("Length") + ylab("Width"
)
p
```

IRIS Length/Width



## 棒グラフ geom\_bar()

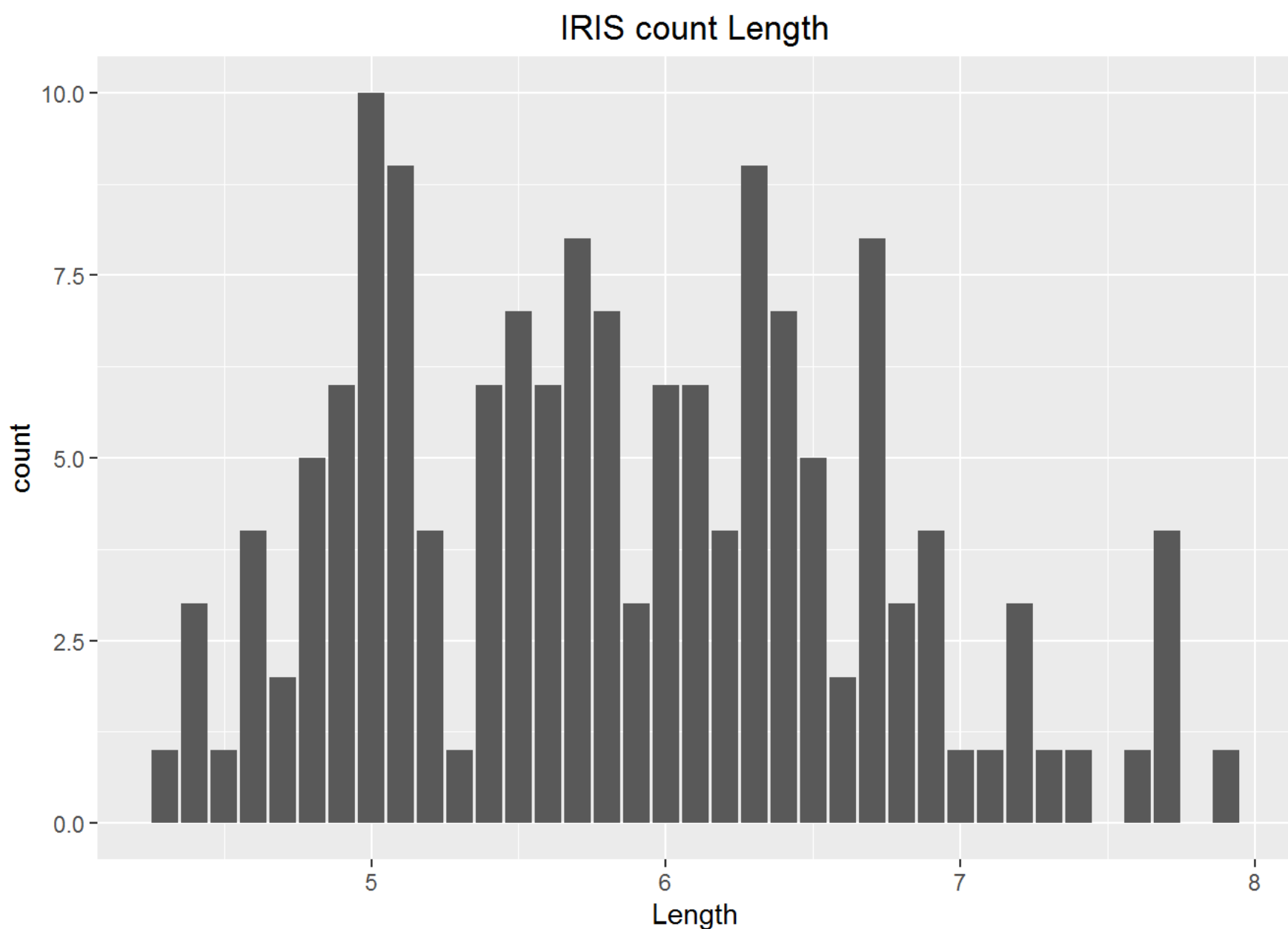
### 通常の棒グラフ

通常の棒グラフを作ります。今回は長さ毎に区切ったときの該当数をカウントし、それを棒グラフにしました。 `geom_bar()`はyを指定しないとこのような仕様になります。

```
#x軸をSepal.Length、y軸をSepal.Widthとした棒を作成します。
p <- ggplot(iris, aes(x=Sepal.Length))

#上で作成したグラフ枠内にプロットしていきます。
p <- p + geom_bar()+ ggtitle("IRIS count Length")+ xlab("Length")
p
```



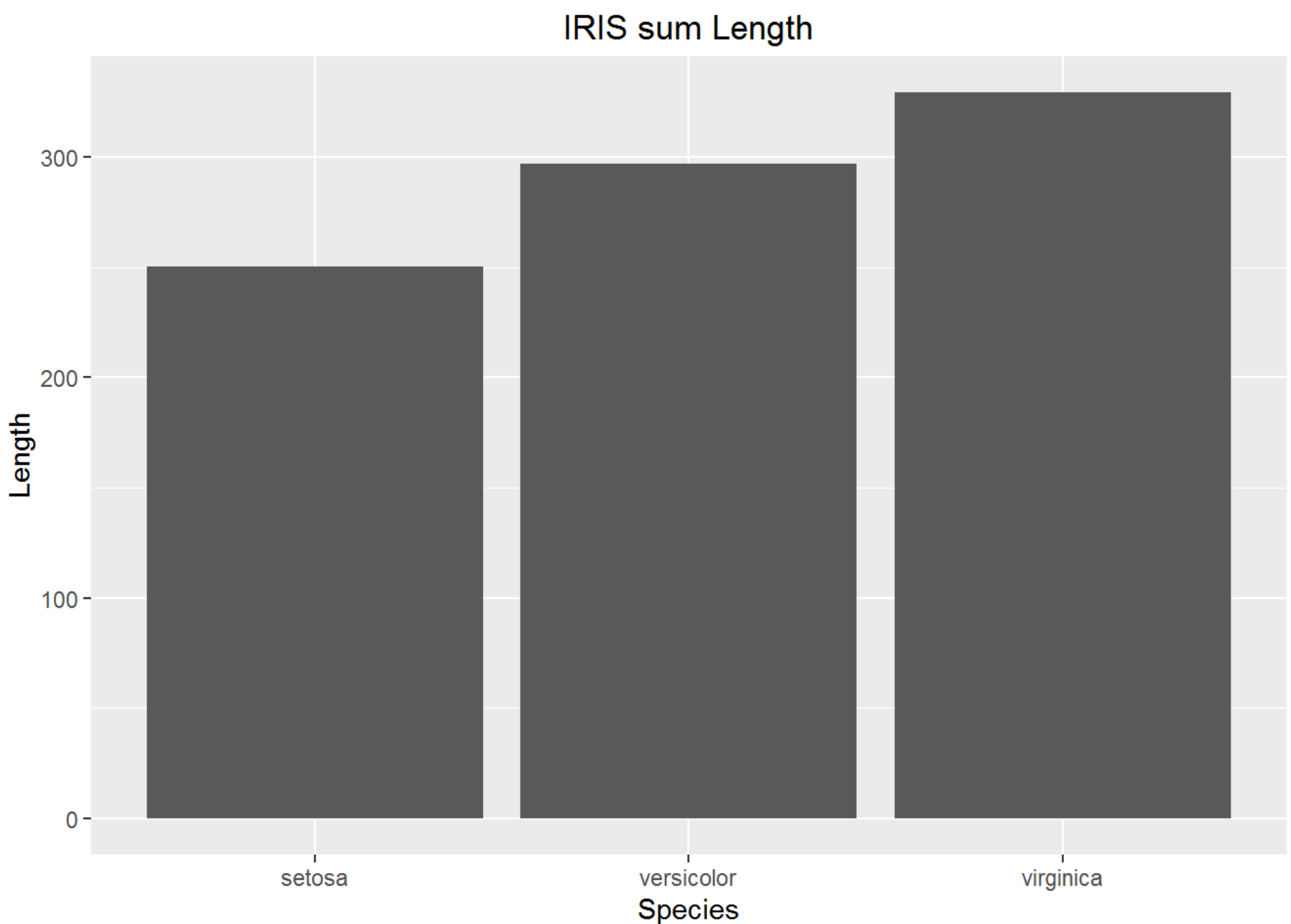


## Species毎Length総計の棒グラフ

一番地道にlengthの総計の棒グラフを描かせるには、一度irisのデータを集計・加工しSpeciesとその集計のみからなるデータフレームをつくると良いでしょう。しかし加工なしでggplotのみによって実装しようとするとな下のようになります。

```
#x軸をSpecies、y軸をSepal.Lengthとした棒を作成します
p <- ggplot(iris, aes(y=Sepal.Length, x=Species))

#上で作成したグラフ枠内にプロットしていきます。
p <- p + stat_summary(fun.y=sum,geom="bar") + ggtitle("IRIS sum Length") + xlab(
"Species") + ylab("Length")
p
```



`stat_summary(fun.y=sum,geom="bar")`の`fun.y`に`sum`を指定する事によって合計を集計した棒グラフを作成する事が出来ます。また代わりに`mean`を指定すると平均値の集計グラフが出来るなど、様々な活用が出来ます。

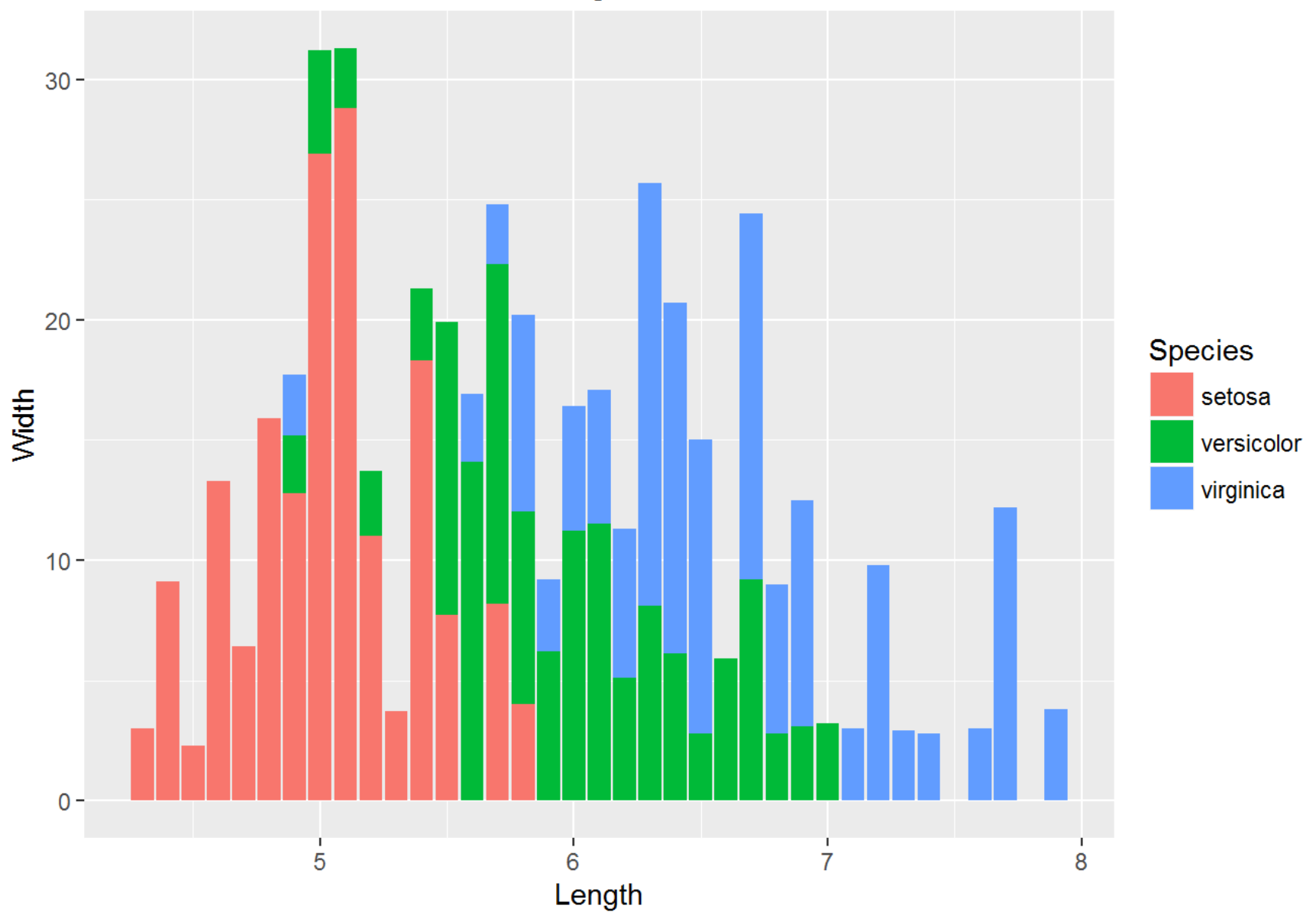
## 積み上げ棒グラフ

`fill`に値を指定する事で下図のような積み上げ棒グラフを作成することが出来ます。

```
#x軸をSepal.Length、y軸をSepal.Widthとした棒を作成します。
p <- ggplot(iris, aes(x=Sepal.Length,y=Sepal.Width,fill=Species))

#上で作成したグラフ枠内にプロットしていきます。
p <- p + geom_bar(stat = "identity")+ ggtitle("IRIS Length/Width") + xlab("Length"
) + ylab("Width")
p
```

IRIS Length/Width



## ヒストグラム geom\_histogram()

### Speciesごとのヒストグラム

*#x軸をSepal.Length、y軸をSepal.Widthとした枠を作成します。*

```
p <- ggplot(iris, aes(x=Sepal.Length,fill=Species))
```

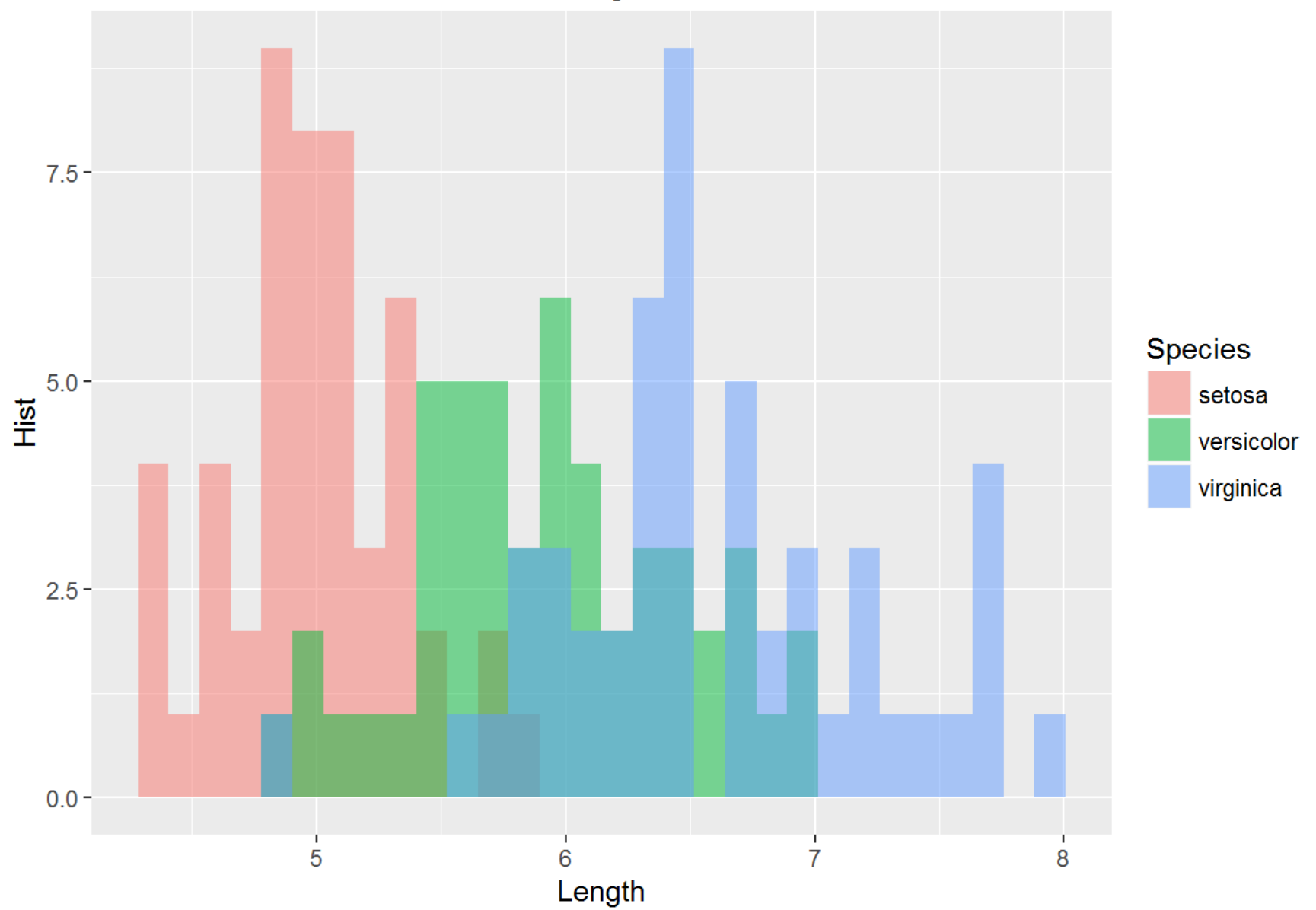
*#上で作成したグラフ枠内にプロットしていきます。*

```
p <- p + geom_histogram(position = "identity",alpha = 0.5)+ ggtitle("IRIS Length H  
ist") + xlab("Length") + ylab("Hist")
```

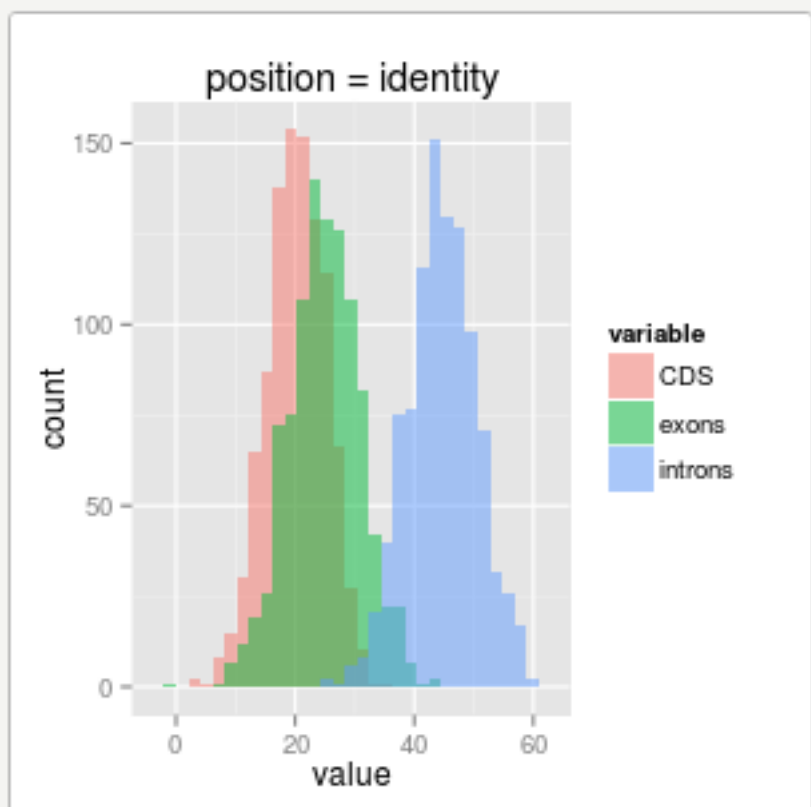
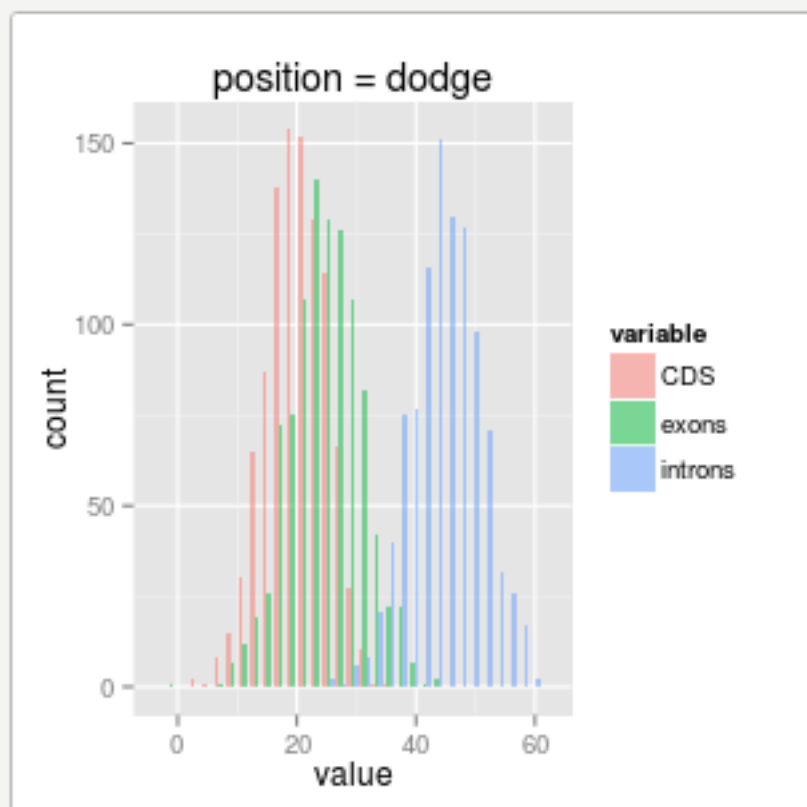
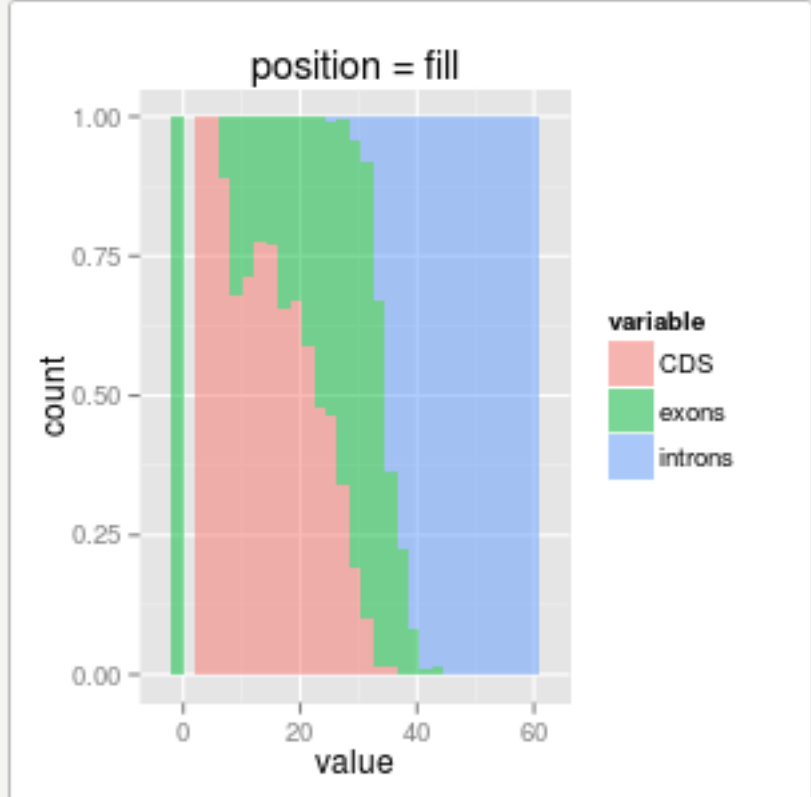
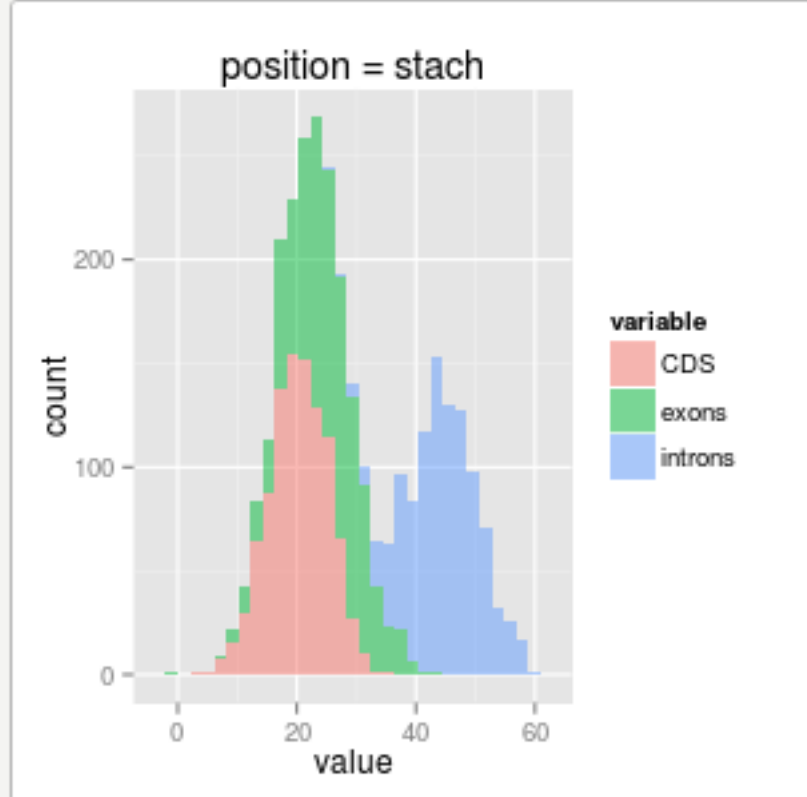
```
p
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

IRIS Length Hist



alphaの値を指定する事によって透明度をいじる事が出来ます。 またpositionにはいろいろな値を指定する事が出来ます。



## 終わりに

このほかにもggplotにはたくさんの機能が有ります。利用できるものは利用していき、これからの学生生活に役立てていきましょう！