ggplot2 の基礎

Toshiki SHIBANO

2021-02-17

目次

ggplot2 とは

Grammar of Graphics(グラフィックスの文法, Wilkinson 2005) に従って実装されたパッケージ．R には初めからグラフを描画する機能が備わっていますが，ggplot2 パッケージを使う方が多くのグラフを一貫した方法で美しく描くことができます．
今回の資料では，ggplot2 でグラフを描く概略を説明します．何故このグラフを描くのか (why)，どのようにしてこのグラフを描くのか (how) については述べません．それは別の資料の役割です．パッケージのインストールは次のように行います．

```
# ggplot2のインストール
# tidyverseをインストールすることをオススメします.
#install.packages("ggplot2")
#install.packages("tidyverse)

# パッケージの読み込み
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts --------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#library(ggplot2)
```

ggplot2 を用いたグラフの描き方

ggplot2 を用いた作図は色々な要素を足し合わせて一つのグラフを作成します．以下のコードをみれば分かりますが，データとグラフの要素を結びつけたもの (p) に + を用いてどんどん足していきます．
基本的な流れは以下のようになります．

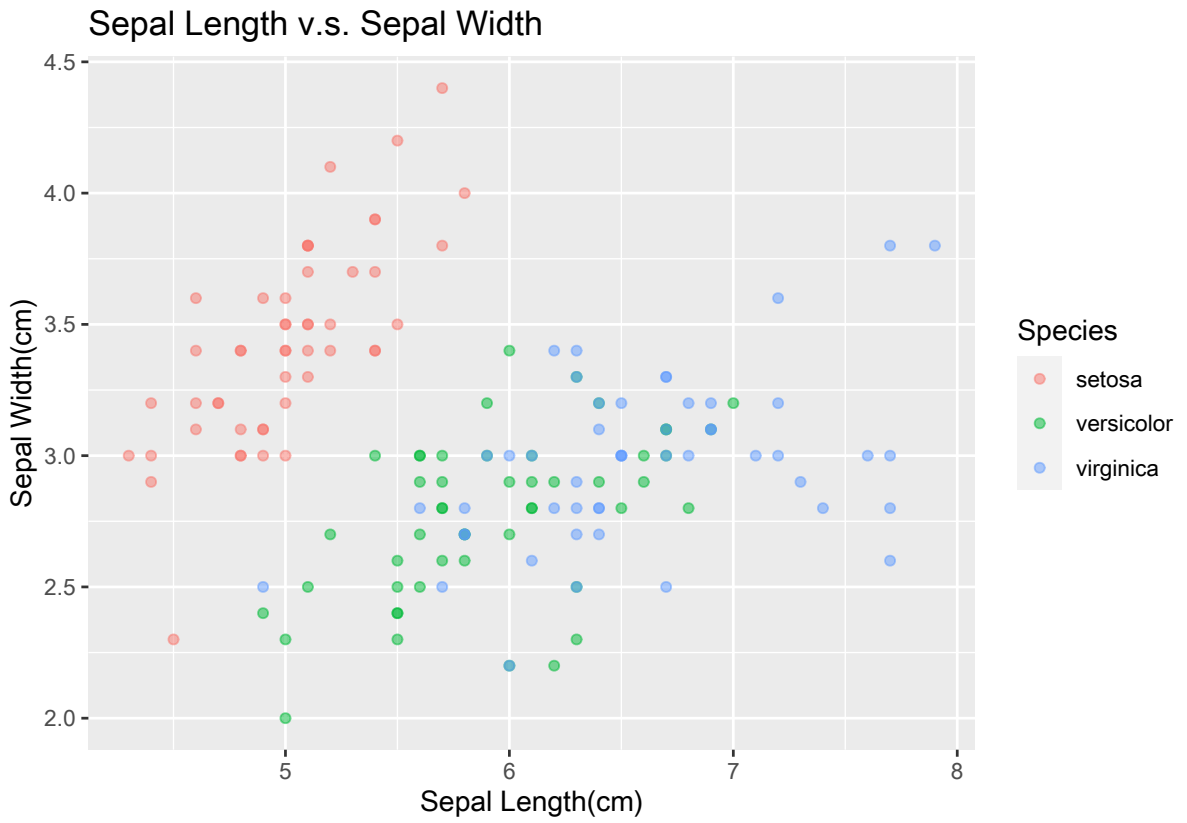
1. 作るグラフの構造を考える
2. ggplot() 関数を使ってデータを指定し，データ中の変数とグラフの要素 (点や色) を結びつける変数とグラフの要素を結びつけることを審美的要素のマッピング (aesthetic mappings) といいます
3. geom_() 関数を用いてプロットのタイプを選択する
4. 座標と目盛りを調整する
5. 凡例などを加える

実際の例を通してみていきましょう．iris データセットを用います．

散布図

```
# 種別に色分けした Sepal.Length と Sepal.Width の散布図を描きたい
p <- ggplot(data = iris,
            mapping = aes(x = Sepal.Length, y = Sepal.Width, color = Species))
p + geom_point(alpha = 0.5) +
    labs(x = "Sepal Length(cm)",
```

```
        y = "Sepal Width(cm)",
        title = "Sepal Length v.s. Sepal Width")
```



散布図を書いた例を一つずつ解説していきます.

1. 作るグラフの構造を考える
   - コメントにある,「種別に色分けした Sepal.Length と Sepal.Width の散布図」が該当します
2. ggplot() 関数を使って，データを指定し，データ中の変数とグラフの要素 (点や色) を結びつける
   - ggplot(data = iris, mapping = aes(x = Sepal.Length, y = Sepal.Width, color = Species)) がこれに当たります.
   - data 引数に用いるデータを渡します．data = iris
   - mapping 引数に aes() 関数を用いて，x, y を指定します．color には各点を色分けしたい変数を指定します．この例では，x 軸に Sepal.Length，y 軸に Sepal.Width を取り，Species で色分けをしています.
3. geom_() 関数を用いてプロットのタイプを選択する
   - geom_point() で散布図を描きます
   - alpha 引数を設定して，色の濃さを調整しました．これで重なっている点が分かります.
4. 座標と目盛りを調整する
   - 今回は調整していません.

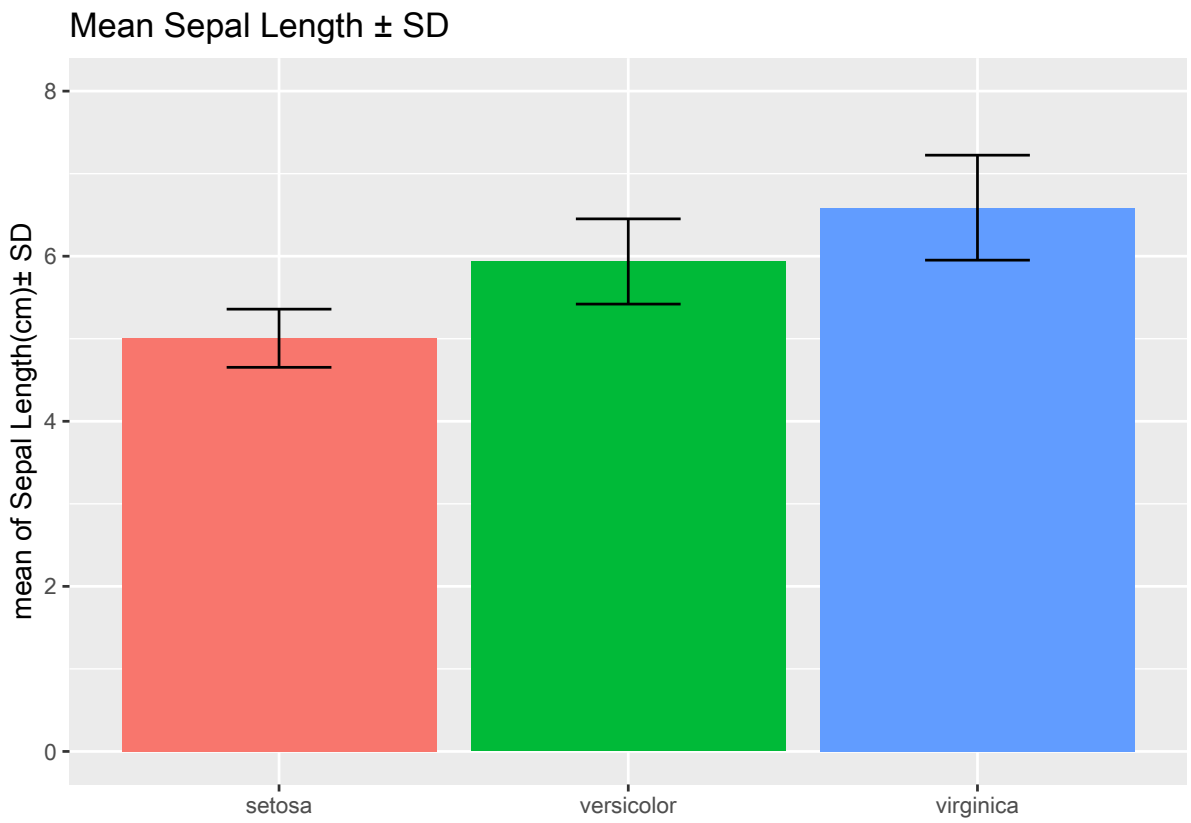
5. 凡例などを加える
    - labs() 関数を用いて，x, y 軸ラベルの変更やタイトルを加えます.

棒グラフ

```
# 種別に Sepal.Length の平均値および標準偏差の棒グラフを描きたい
# まずは dplyr を使って Sepal.Length の平均値および標準偏差を求める
iris_by_species <-
  iris %>%
  group_by(Species) %>%
  summarise(N = n(),
            mean_Sepal_Length = mean(Sepal.Length, na.rm = TRUE),
            sd_Sepal_Length = sd(Sepal.Length, na.rm = TRUE),
            .groups = "drop"
            )

# データの確認
iris_by_species
```

Species	N	mean_Sepal_Length	sd_Sepal_Length
setosa	50	5.006	0.3524897
versicolor	50	5.936	0.5161711
virginica	50	6.588	0.6358796

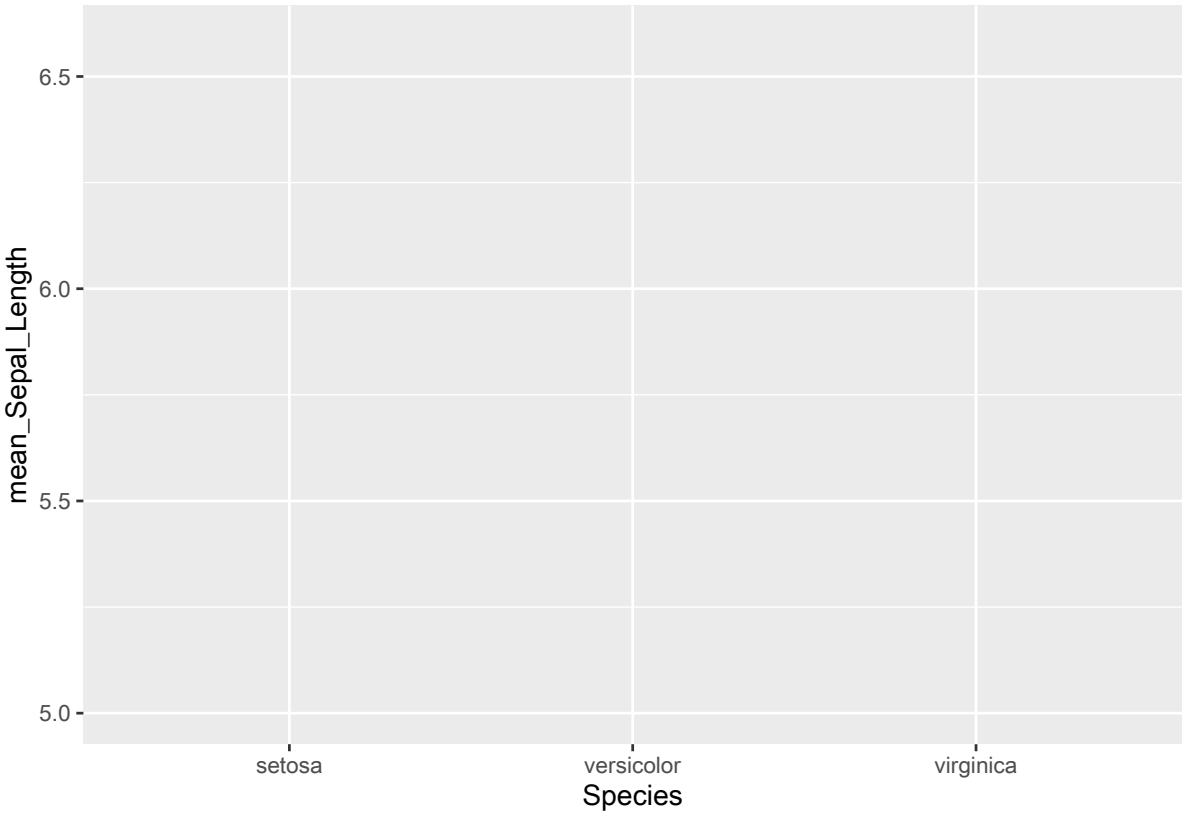
```
# 棒グラフの描画
p <- ggplot(data = iris_by_species,
            mapping = aes(x = Species, y = mean_Sepal_Length, fill = Species))
p + geom_col() +
    geom_errorbar(mapping = aes(ymin = mean_Sepal_Length - sd_Sepal_Length,
                                ymax = mean_Sepal_Length + sd_Sepal_Length),
                  width = 0.3) +
    scale_y_continuous(limits = c(0, 8)) +
    labs(x = NULL, y = "mean of Sepal Length(cm)± SD",
         title = "Mean Sepal Length ± SD") +
    guides(fill = "none")
```



このようにして目的のグラフを描くことが出来ます．今回は一つ一つ何が起きているかを確認していきましょう．

データの宣言および審美的要素のマッピング

```
p <- ggplot(data = iris_by_species,
            mapping = aes(x = Species, y = mean_Sepal_Length, fill = Species))
p
```

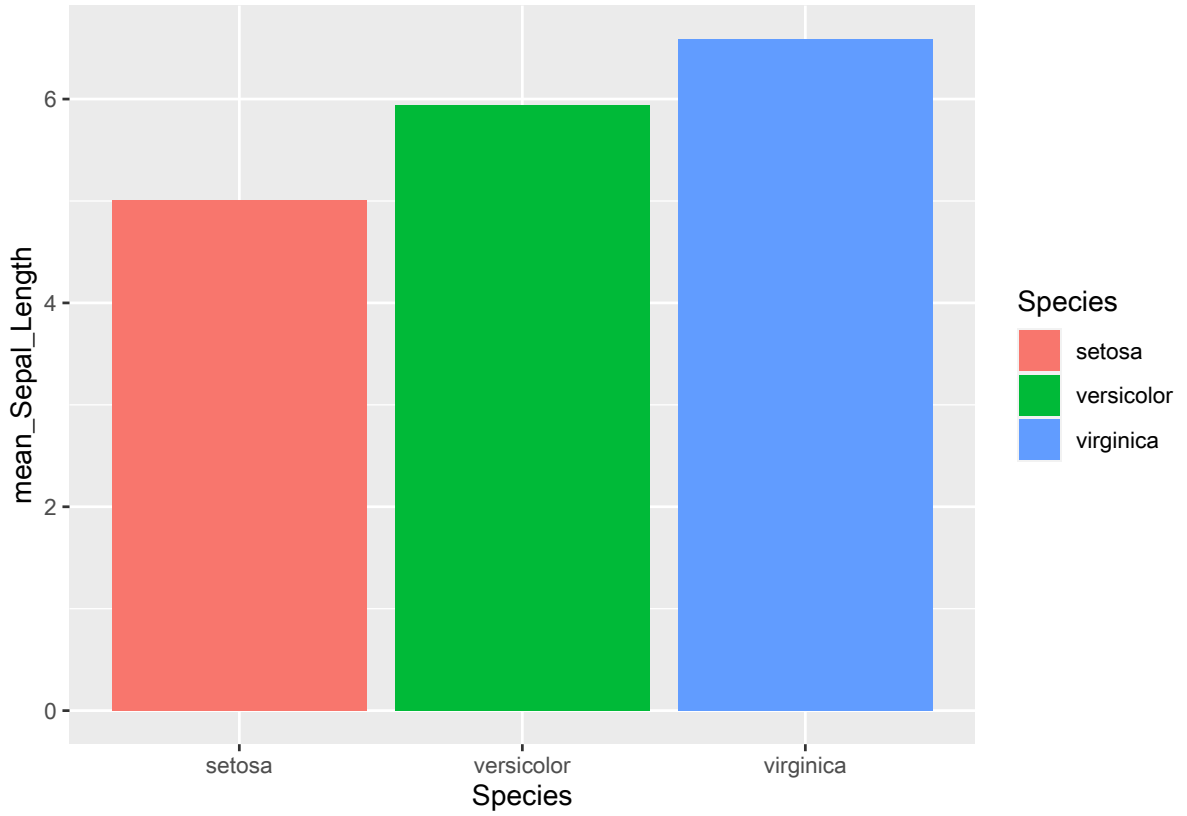


これはデータとして iris_by_species を使うと宣言し，x に Species，y に mean_Sepal_Length をマッピングします．`fill = Species` で Species で色分けすると伝えます．fill の代わりに color にするとどうなるか確認してみてください．まだこの時点では，使うデータと x, y や fill が与えられただけなので，描画されません．

補足
今回は説明するためにあえて棒グラフに色をつけましたが，このグラフなら色は必要ないと思います．

棒グラフを描く

```
p <- p + geom_col()
p
```



geom_col() で棒グラフを描きます.

補足

棒グラフは geom_bar() 関数でも描くことができます (むしろこっちがメインな気がします). geom_bar() 関数は，デフォルトでは個数を出力するようになっており，上の例で

```
p + geom_bar()
```

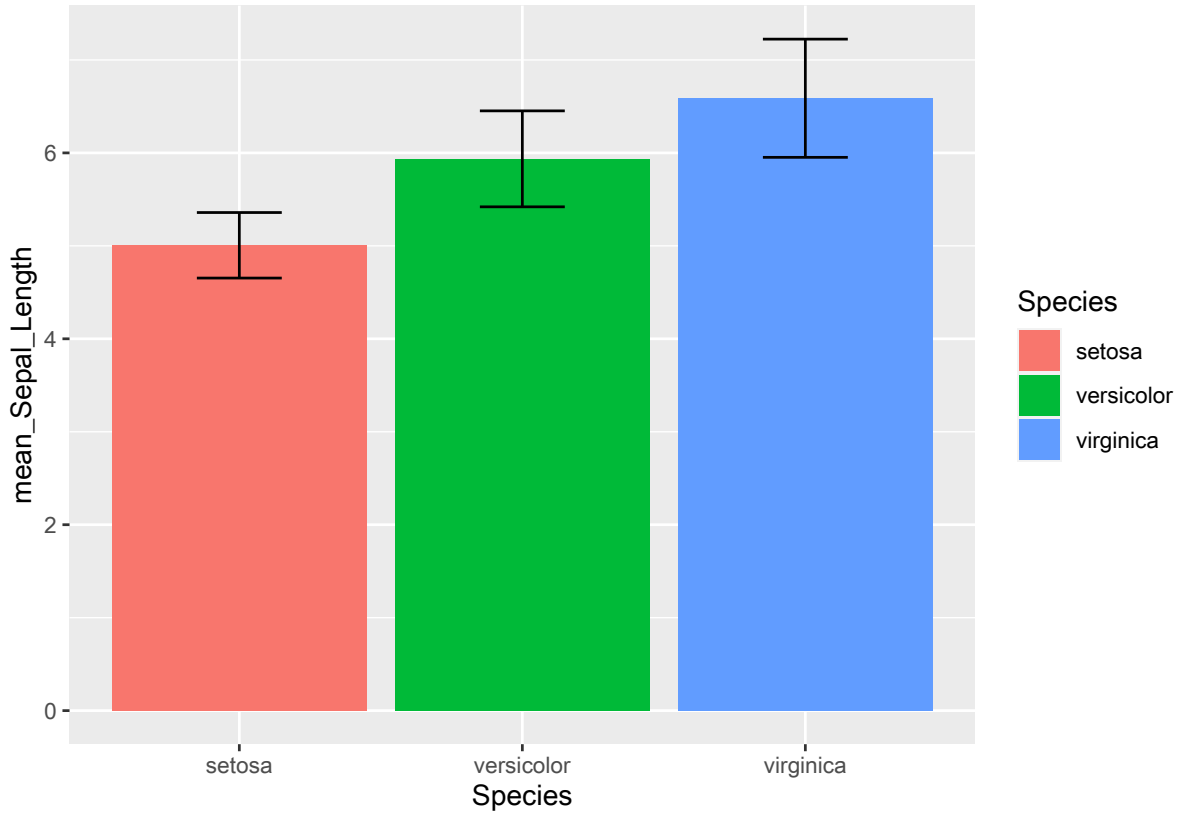
とすると，エラーが出ます．ですので，

```
p + geom_bar(stat = "identity")
```

とすることで，描くことができます.

エラーバーを加える

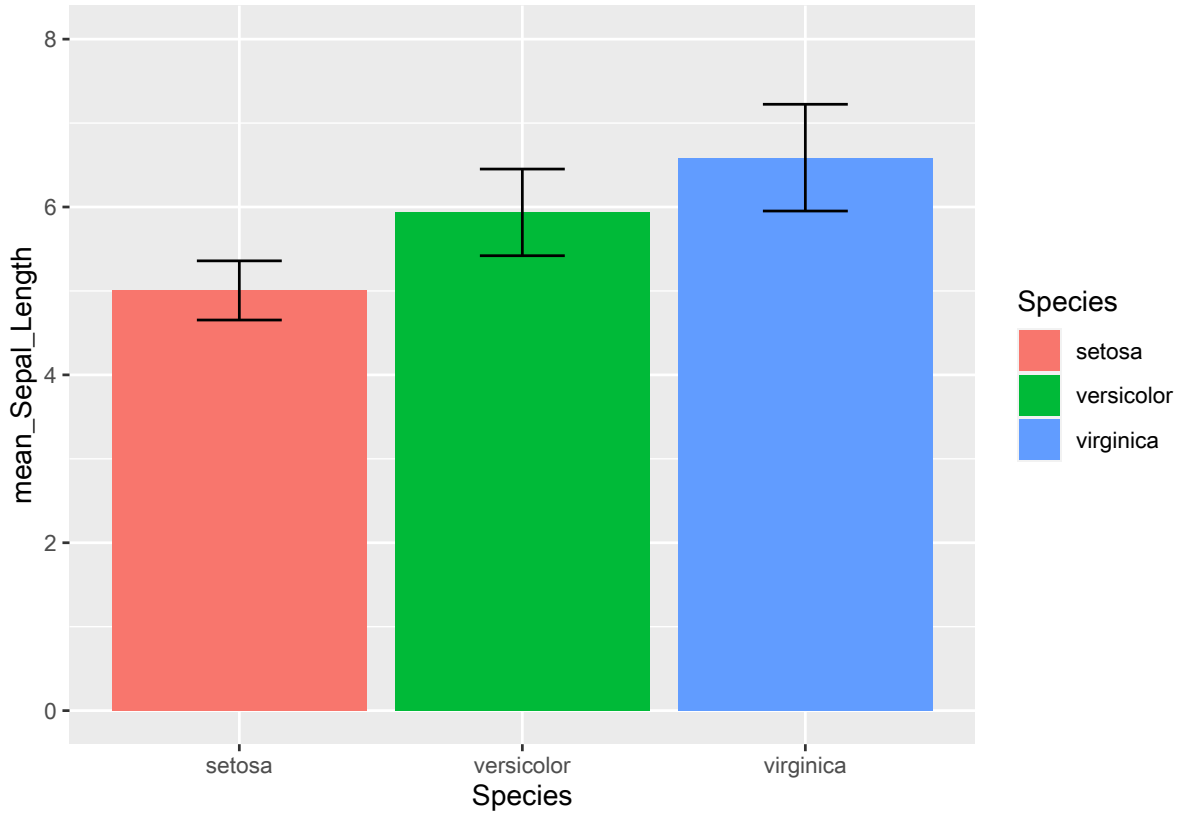
```
p <- p + geom_errorbar(mapping = aes(ymin = mean_Sepal_Length - sd_Sepal_Length,
                                     ymax = mean_Sepal_Length + sd_Sepal_Length),
                       width = 0.3)
p
```



エラーバーを描き加えます．geom_errorbar() 関数を用いて，mapping に ymin と ymax を指定することで，エラーバーを描くことがでます．ymin は mean から sd を引いたもの，ymax は足したものになってます．width はエラーバーの幅を示しています．

スケールの調整

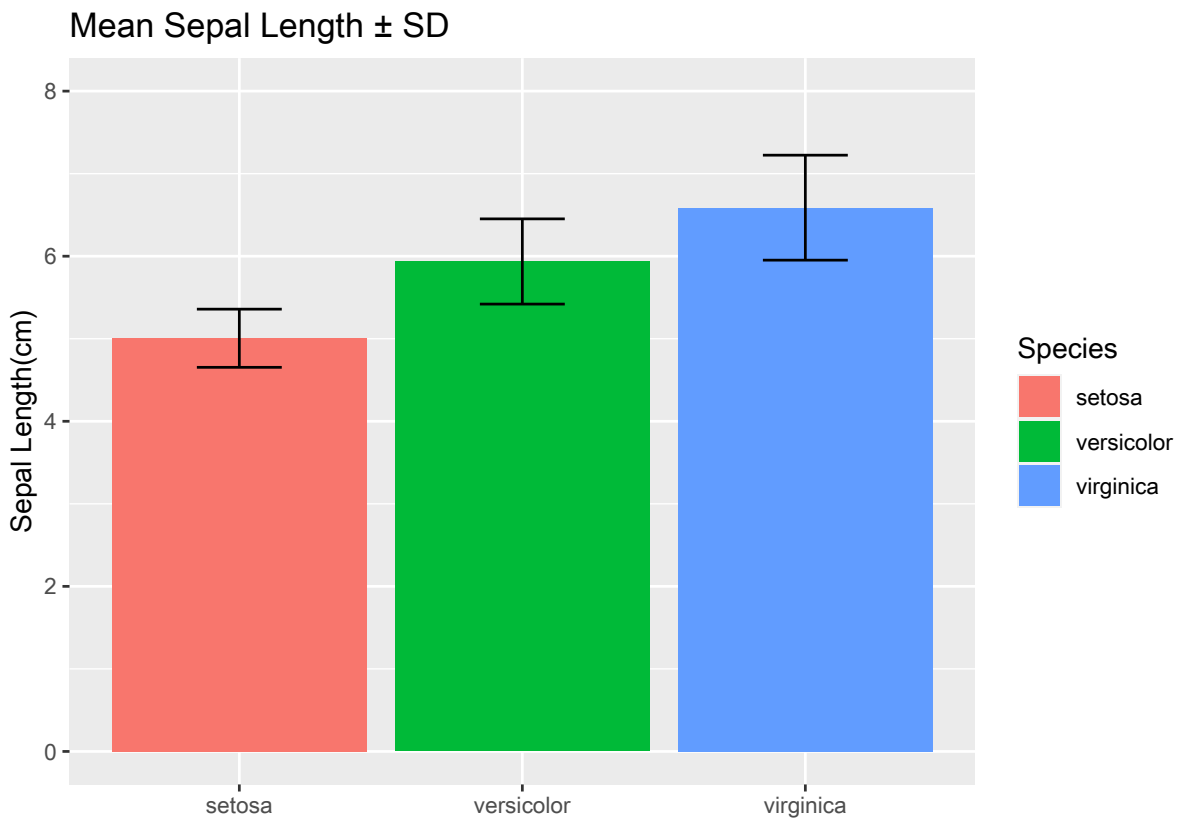
```
p <- p + scale_y_continuous(limits = c(0, 8))

p
```



`scale_y_continuous(limits = c(0, 8))`
で y 軸の範囲を設定します．scale_x_continuous を用いれば x 軸の範囲を設定することができます．また，breaks 引数を設定することで，軸のラベルの表示名を自由に調整できます．

ラベルの設定

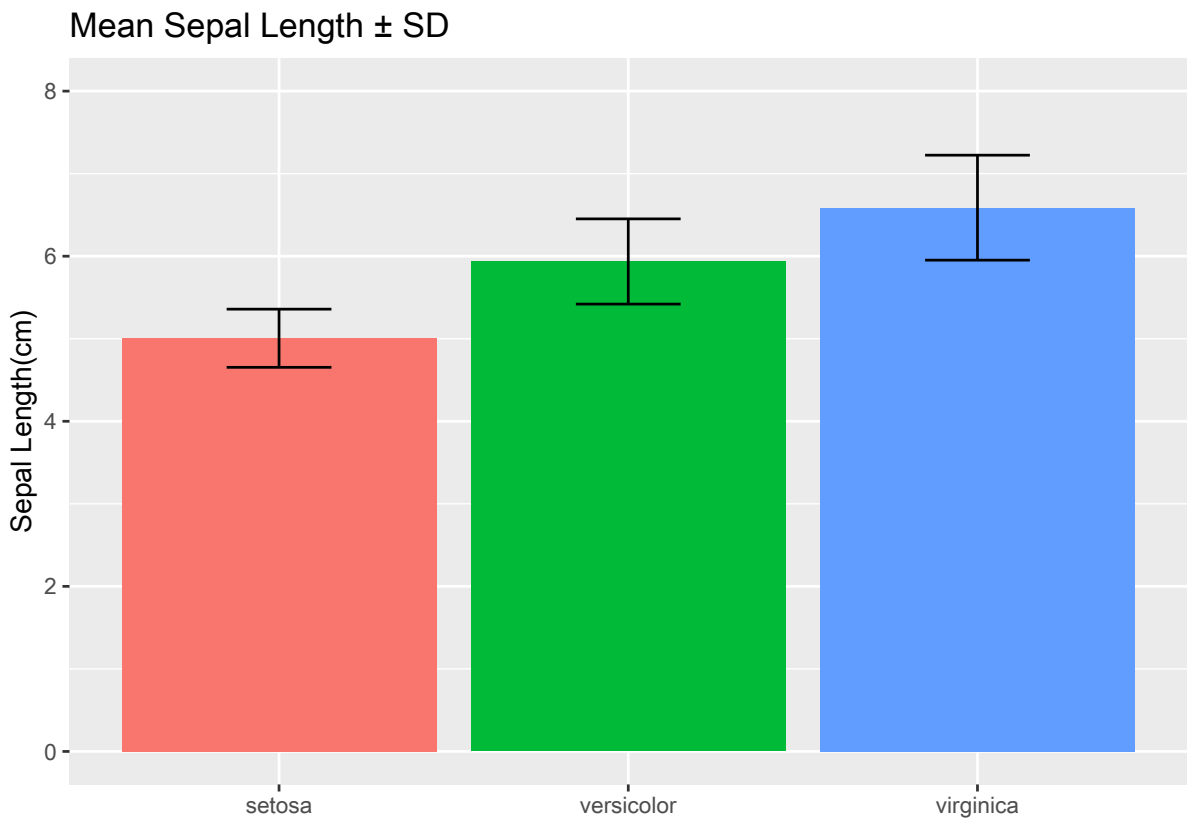
```
p <- p + labs(x = NULL, y = "Sepal Length(cm)",
              title = "Mean Sepal Length ± SD")
p
```



x 軸名を消し，y 軸名を変えます.

凡例の設定

```
p <- p + guides(fill = "none")
p
```



凡例は特に必要ではないので，消します．

まとめると

1. 作るグラフの構造を考える
   - 平均値 ± 標準偏差の棒グラフを描く
2. ggplot() 関数を使ってデータを指定し，データ中の変数とグラフの要素 (点や色) を結びつける
   - ggplot() 関数の文
3. geom_() 関数を用いてプロットのタイプを選択する
   - geom_col() を用いて棒グラフを描画
   - geom_errorbar() を用いてエラーバーを描画
4. 座標と目盛りを調整する
   - scale_y_continuous() で目盛りを調整
5. 凡例などを加える
   - labs() で x 軸，y 軸，タイトルを調整
   - guides() で判例を消去

さいごに

初めは何をしてるか分からないと思うので，とりあえず写経しましょう．そして棒グラフの時のように一つずつ確認していきましょう．私も少しずつ資料を作っていきますが，しっかり学びたい方は以下に挙げる参考文献を見ることをオススメします．特に Data Visualization がオススメです (邦訳版を読み途中です)．なぜこのグラフを描くのか (why)，どうやってこのグラフを描くのか (how) を両方とも学ぶことが出来ます．私のグラフに関する資料は主にこの本を参考にすると思います．

参考文献

- Data Visualization by Kieran Healy
  - 邦訳本『データ分析のためのデータ可視化入門』
- R for Data Science by Hadley Wickham & Garrett Grolemund
  - R for Data Science の chapter 3
- ggplot2 by Hadley Wickham
  - ggplot2 の本．かなり読み応えがありそう．読んでないです．
- https://ggplot2.tidyverse.org/reference/index.html
  - ggplot2 の公式サイト
- https://www.jaysong.net/ggplot_intro1/
  - 日本語解説サイト．
- https://github.com/rstudio/cheatsheets/blob/master/translations/japanese/ggplot2-cheatsheet-2.0-ja.pdf
  - ggplot2 のチートシート．手元に置いておきましょう．