

KUDH Basics

統計ソフトウェア『R』 ワークショップ

②データの可視化

山本寛樹（北陸先端科学技術大学院大学）

2022.09.21.

パッケージのインストール

パッケージをインストールしよう

□パッケージ

●R用の関数のセット

- Rのインストール時から備わっているもの
- ウェブ上からダウンロードして使うもの
- 分析の目的にあわせて、必要なパッケージを用いる

●**install.packages()**関数を使用して、パッケージをインストール

- パッケージ名をダブルクォーテーション("")でくくって指定

●tidyverseパッケージをインストールしてみよう

- **install.packages("tidyverse")**

関数とは？

□関数 (function)

- 「入力」に対して「出力」を返すもの
 - ・ 入力する値のことを「**引数**（ひきすう）」と呼ぶ
 - ・ 引数が複数あることもあり、
左から「第1引数・第2引数・第3引数…」と呼ばれる
- Rの関数は全て、関数の名前の後にカッコ（）がついた形式
 - ・ 例：install.packages()関数



Tidyverseパッケージとは...

□データ解析のためのRパッケージ群

Components



この授業では

ggplot2 パッケージ

を用いた可視化の方法を紹介

パッケージの読み込み

□library()関数でパッケージを読み込む

- パッケージはRを起動するたびに読み込む必要があるので注意。
 - ・コードの先頭で読み込むことをオススメします。

●library(tidyverse) ...tidyverseパッケージを読み込む

```
library(tidyverse)
```

```
## -- Attaching packages -----  
----- tidyverse 1.2.1 --
```

```
## √ ggplot2 2.2.1    √ purrr  0.2.4  
## √ tibble  1.4.2    √ dplyr  0.7.4  
## √ tidyr   0.8.0    √ stringr 1.3.0  
## √ readr   1.1.1    √ forcats 0.3.0
```



tidyverseパッケージに含まれるパッケージ群

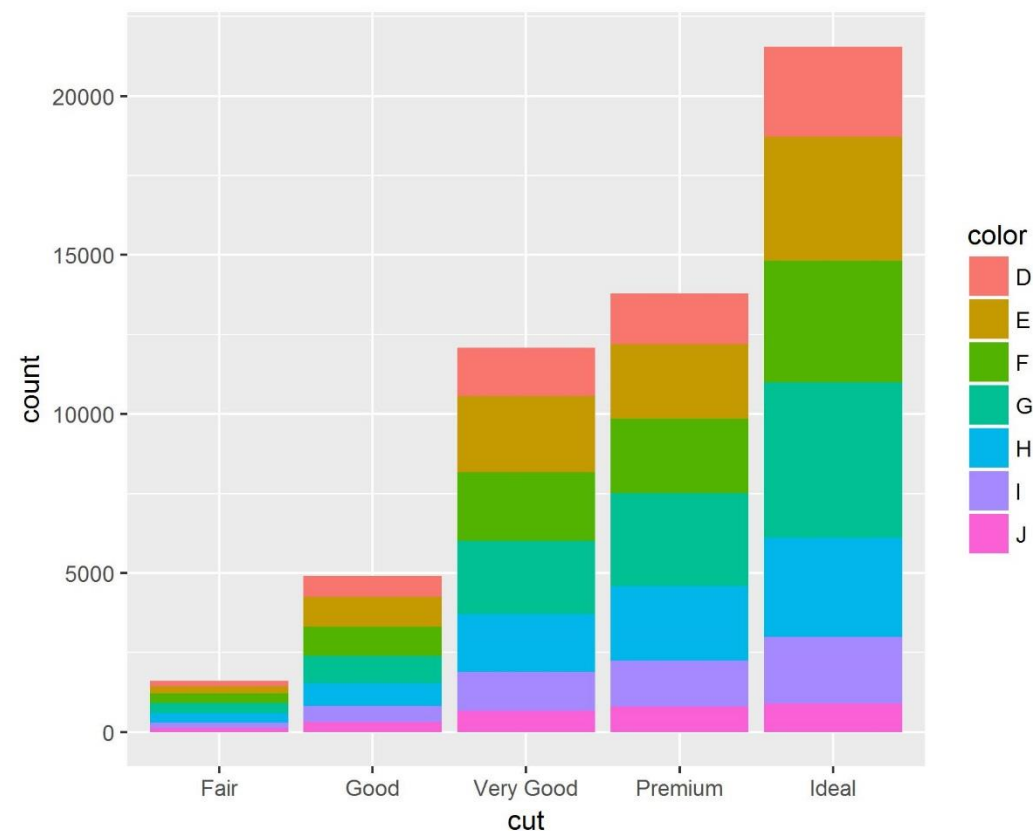
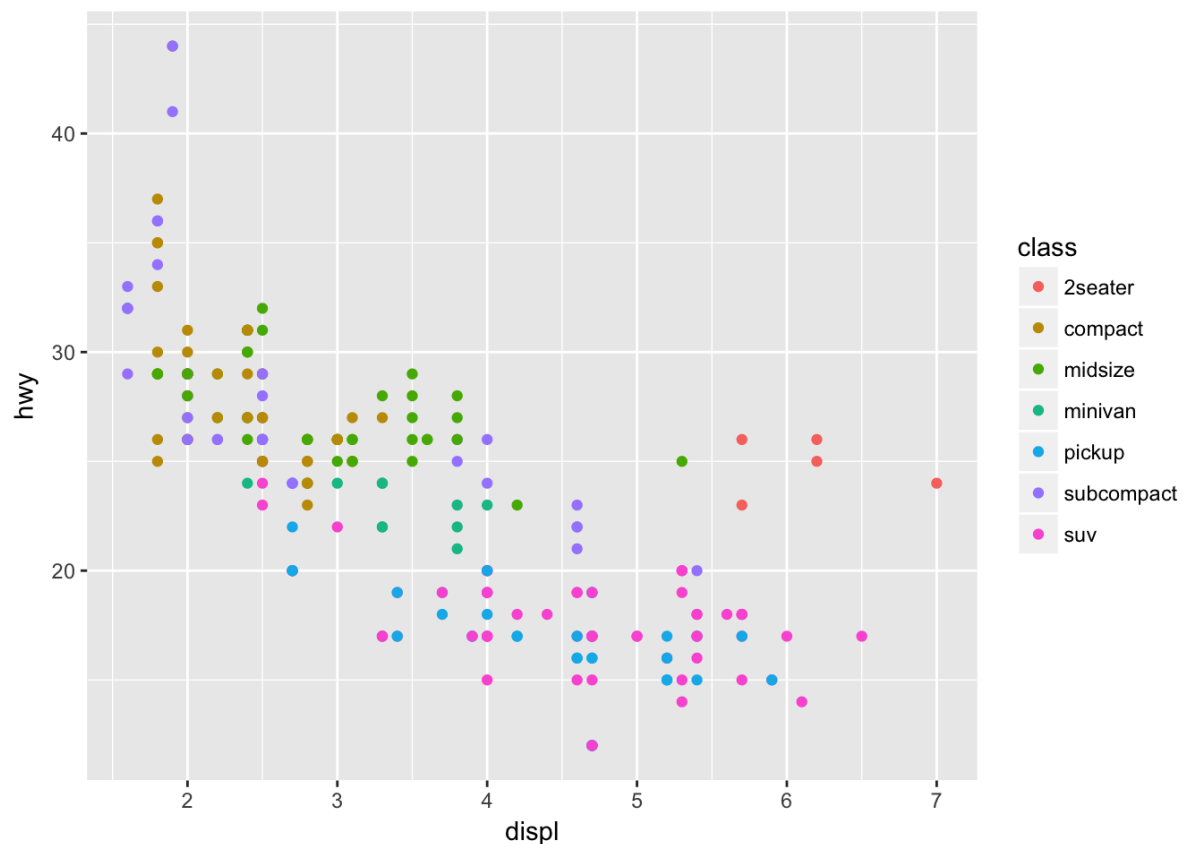
```
## -- Conflicts -----  
----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()
```

ggplot2パッケージ

ggplot2とは...



□tidyverse内の作図用パッケージ



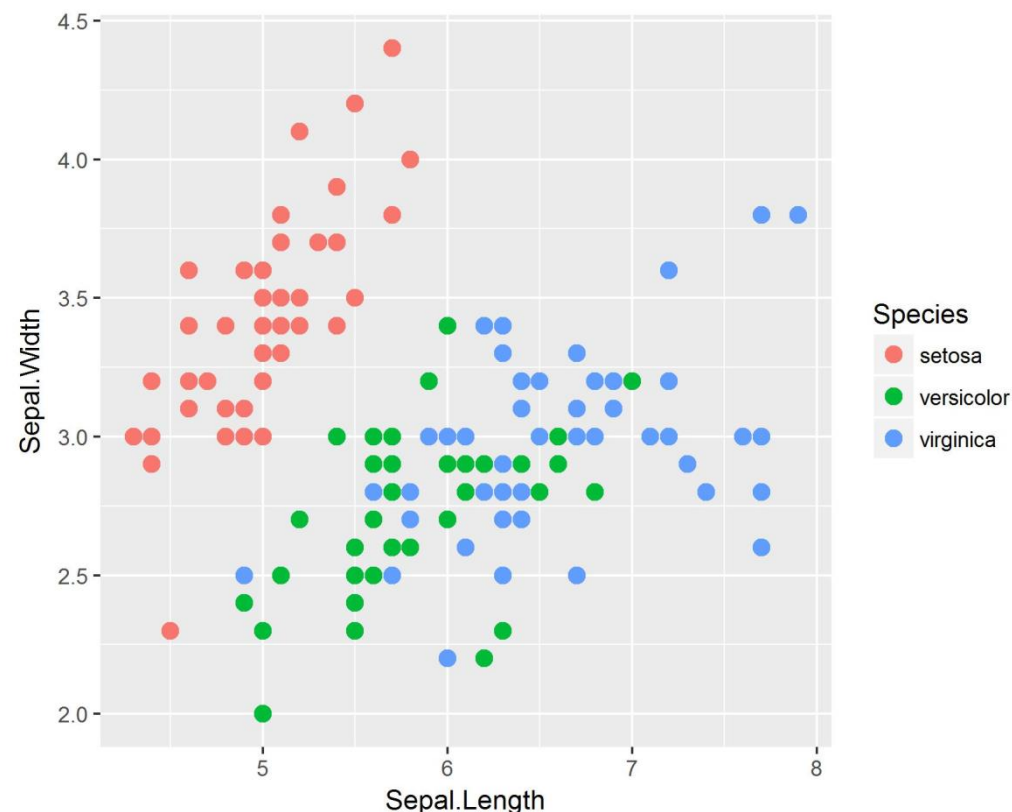
ggplot2による作図のイメージ

□レイヤーを重ねていくことで図をつくる

- ・データフレームをもとに

- ①x軸やy軸にどの変数を使うのか
- ②どの変数によってグルーピングするのか
(色分け・点の形・大きさなど...)
- ③どんな種類のグラフを描くのか
(棒グラフ・箱ヒゲ図・散布図など...)
- ④凡例・軸の名前などの全体的な体裁

レイヤーを重ねるように指定していき、
グラフを可視化していく

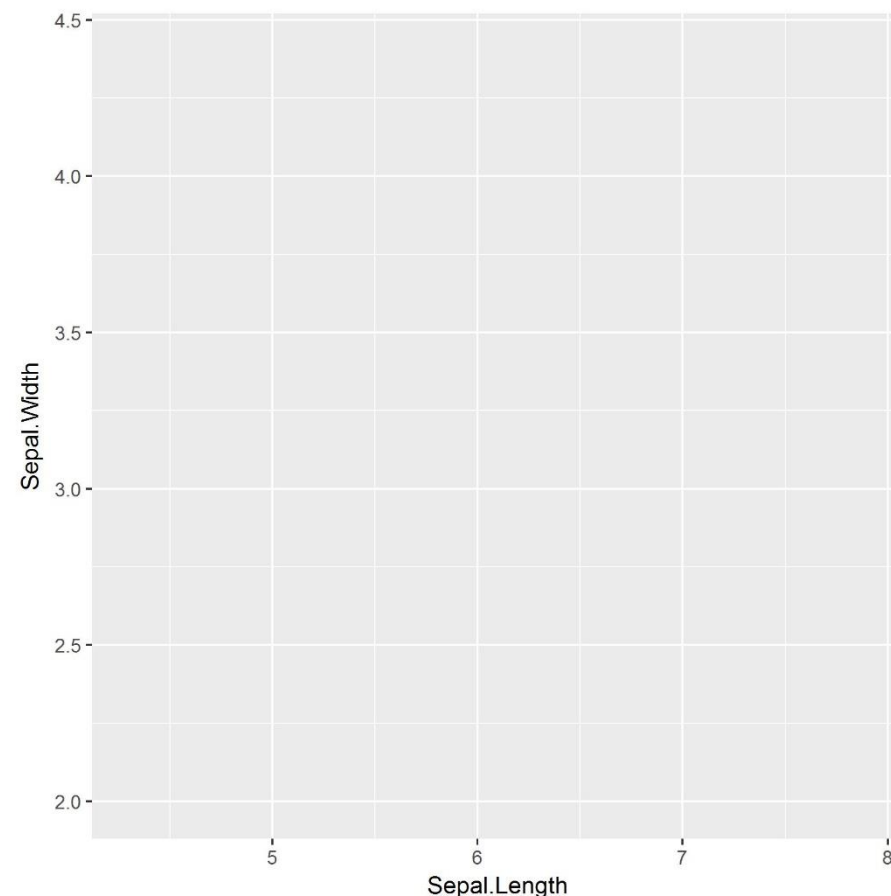


ggplot2による作図の文法

□irisでSepal.LengthとSepal.Widthの散布図を描く

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

x軸がSepal.Lengthで
y軸がSepal.Widthの座標系が描画される



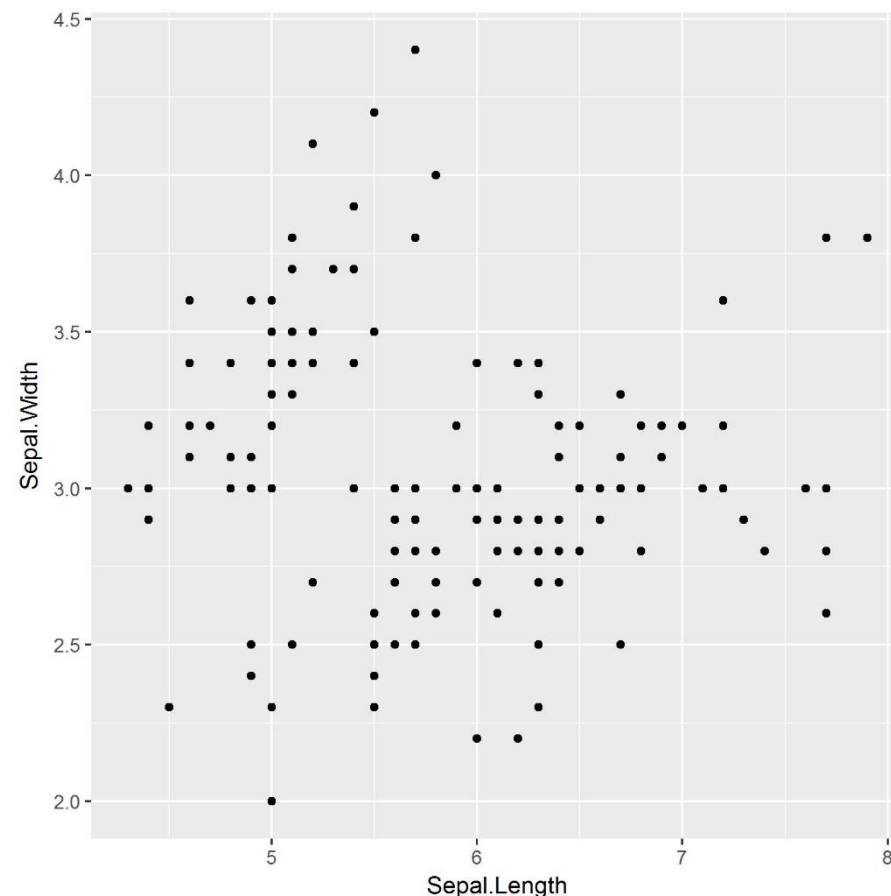
ggplot2による作図の文法

□irisでSepal.LengthとSepal.Widthの散布図を描く

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

```
geom_point()
```

座標系のレイヤーに、
散布図のレイヤーが加えられる

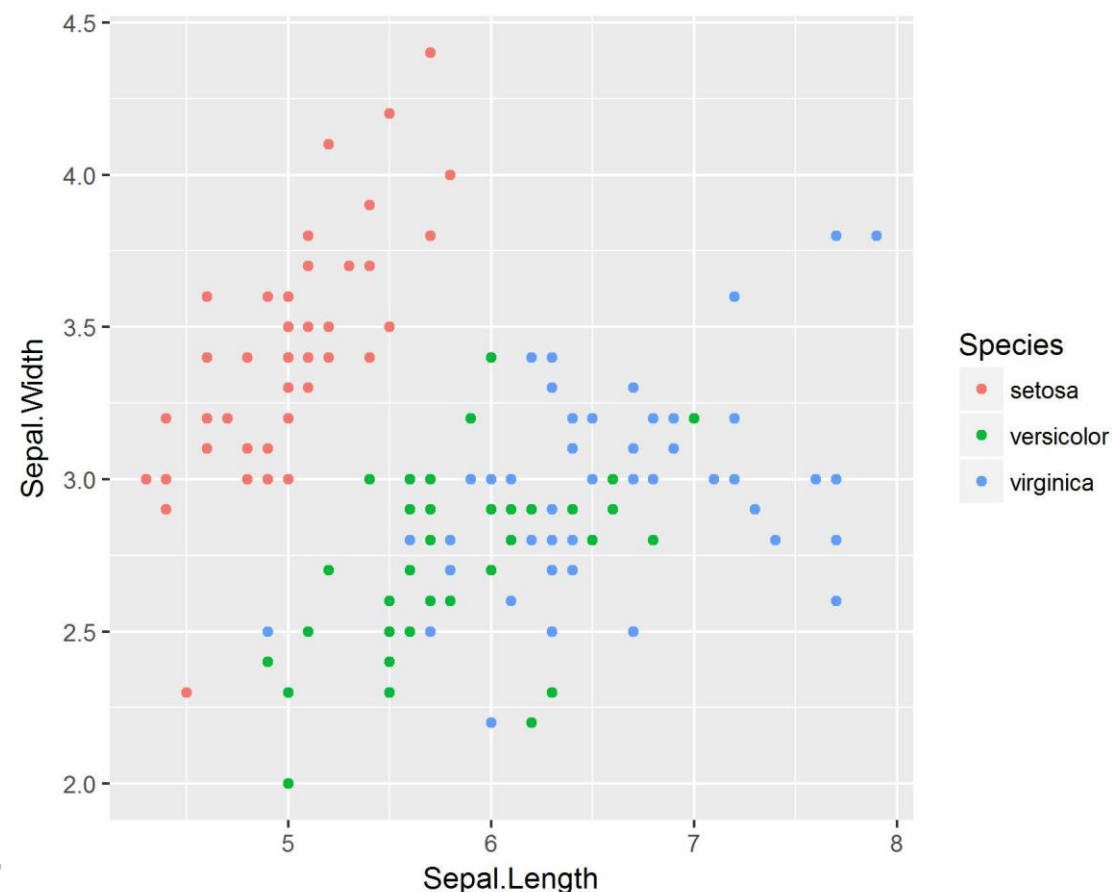


ggplot2による作図の文法

□irisでSepal.LengthとSepal.WidthのSpeciesごとの層別散布図を描く

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

座標系のレイヤーに
Speciesによって色分けがされた
層別散布図のレイヤーが加えられる

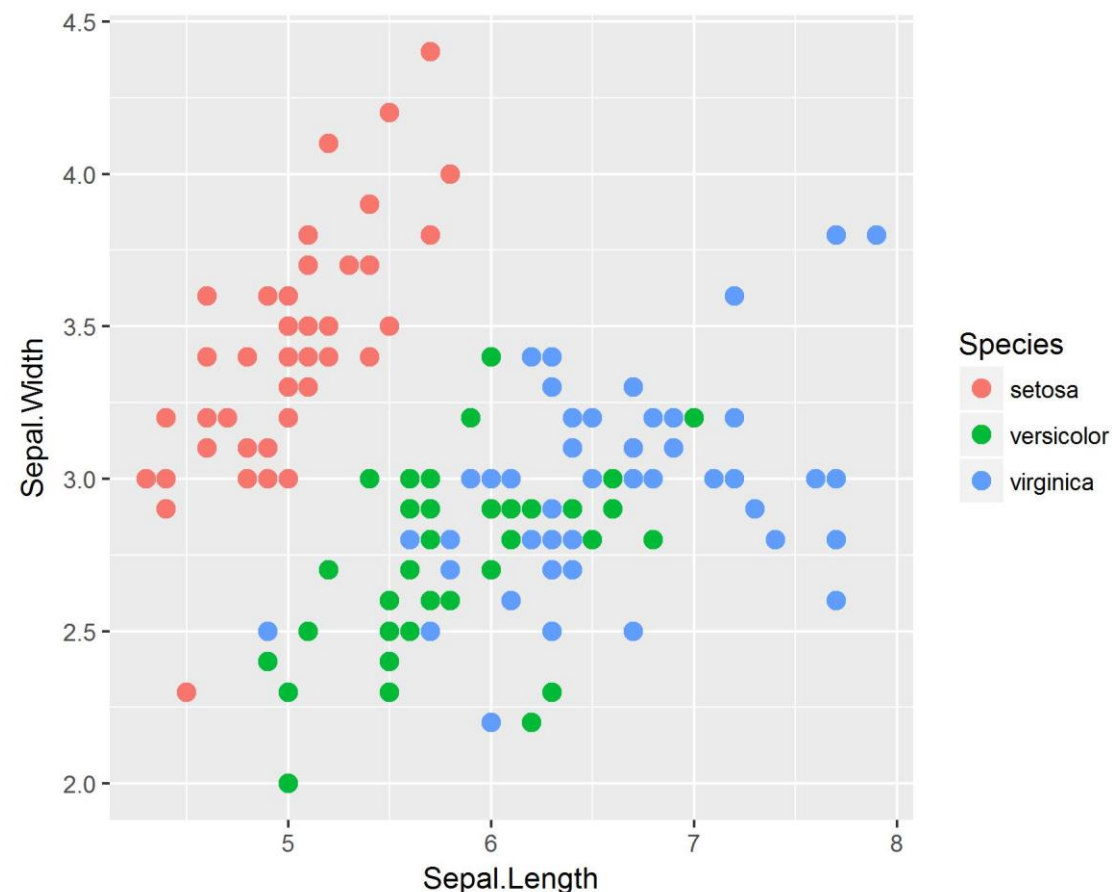


ggplot2による作図の文法

□irisでSepal.LengthとSepal.WidthのSpeciesごとの層別散布図を描く

```
ggplot(iris,  
      aes(x = Sepal.Length,  
          y = Sepal.Width,  
          colour = Species)) +  
geom_point(size = 3)
```

散布図のすべてのドットが
大きくなる



ggplot2による作図の文法

□ **ggplot()**関数で座標系・審美的属性を指定して、グラフの種類を追加

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

```
geom_point()
```

- ← データフレーム名
- ← x軸に使用する変数名
- ← y軸に使用する変数名
- ← 色分けを指定する変数名
- ← グラフの種類を指定

● **ggplot()**関数

- ・ 第1引数：データフレーム名
- ・ 第2引数：**aes()**関数内部で座標系(x軸・y軸)・審美的属性に関する変数を指定

審美的属性とは？

□グラフの外観に関する審美的属性

- 同じ種類のグラフであっても、いろんな要素によって外観は変わる
 - ・ 散布図（ドットの色・大きさ・ドットの種類・透明度）
 - ・ 折れ線グラフ（線の色・太さ・線の種類・透明度）
 - ・ 箱ヒゲ図（箱の内側の色・箱枠の色・透明度）
- グラフの外観に関する要素を**審美的属性**という
 - ・ color・fill・size・shape・linetype・alpha...etc.
 - ・ 審美的属性を指定することでグラフの外観を自由に変えることができる。

審美的属性いろいろ

審美的属性	意味	使いどころ
colour/color	線や点の色	ほとんどのグラフ
fill	内側領域の色	ほとんどのグラフ
alpha	透明度	ほとんどのグラフ
size	ドットの大きさ	散布図
shape	ドットの種類	散布図
linetype	線の種類	折れ線グラフ
lwd	線の太さ	ほとんどグラフ

aes()関数の内側か、外側か

□まず、「何によってグラフの外観を変えたいか」を考える

- データフレームdfにある変数Aの値に応じて、散布図のドットの色を変えたい
 - aes()関数の内部で、「審美的属性=変数名」を加える
 - **ggplot(df, aes(x = ..., y = ..., colour = A)) + geom_point()**
- どのドットも、その値に関係なく、全てオレンジ色にしたい
 - aes()関数の外部で、「審美的属性=...」と指定
 - **ggplot(df, aes(x = ..., y = ...)) + geom_point(colour = "orange")**

基本的な作図

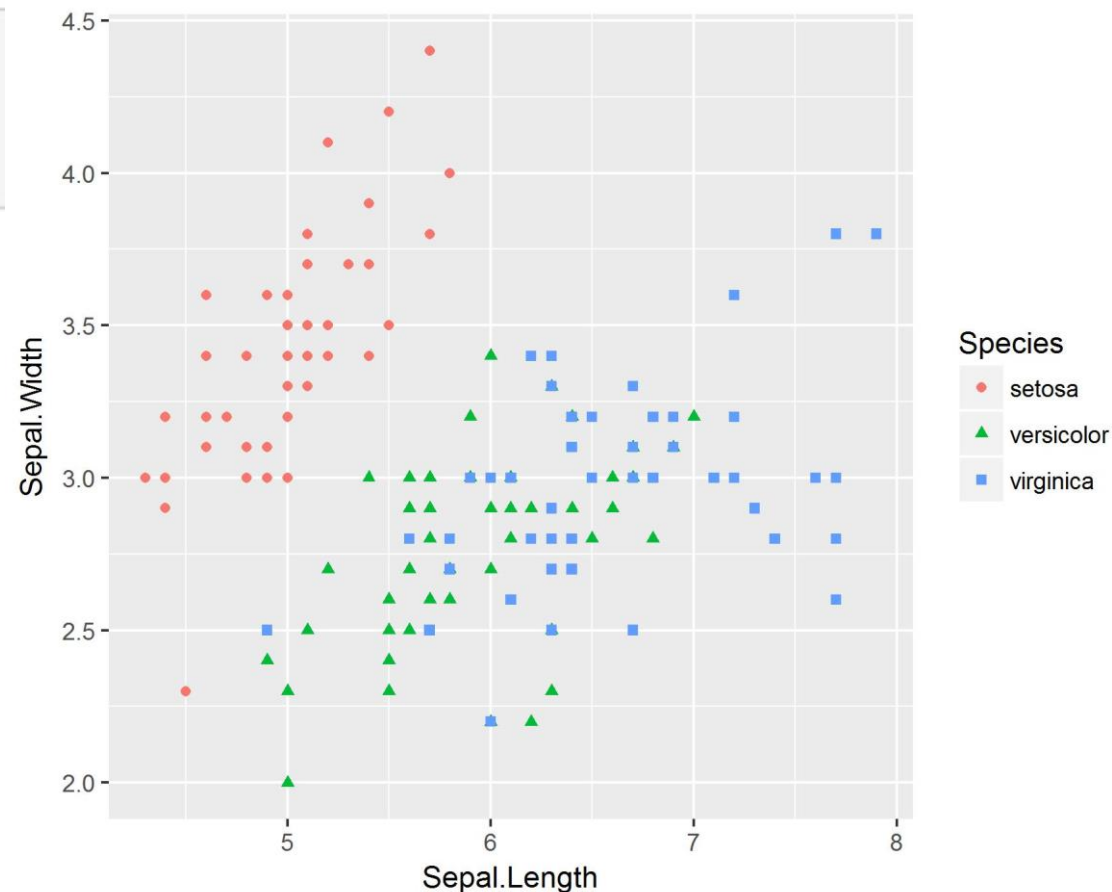
散布図

□ggplot()関数+geom_point()関数

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

●審美的属性

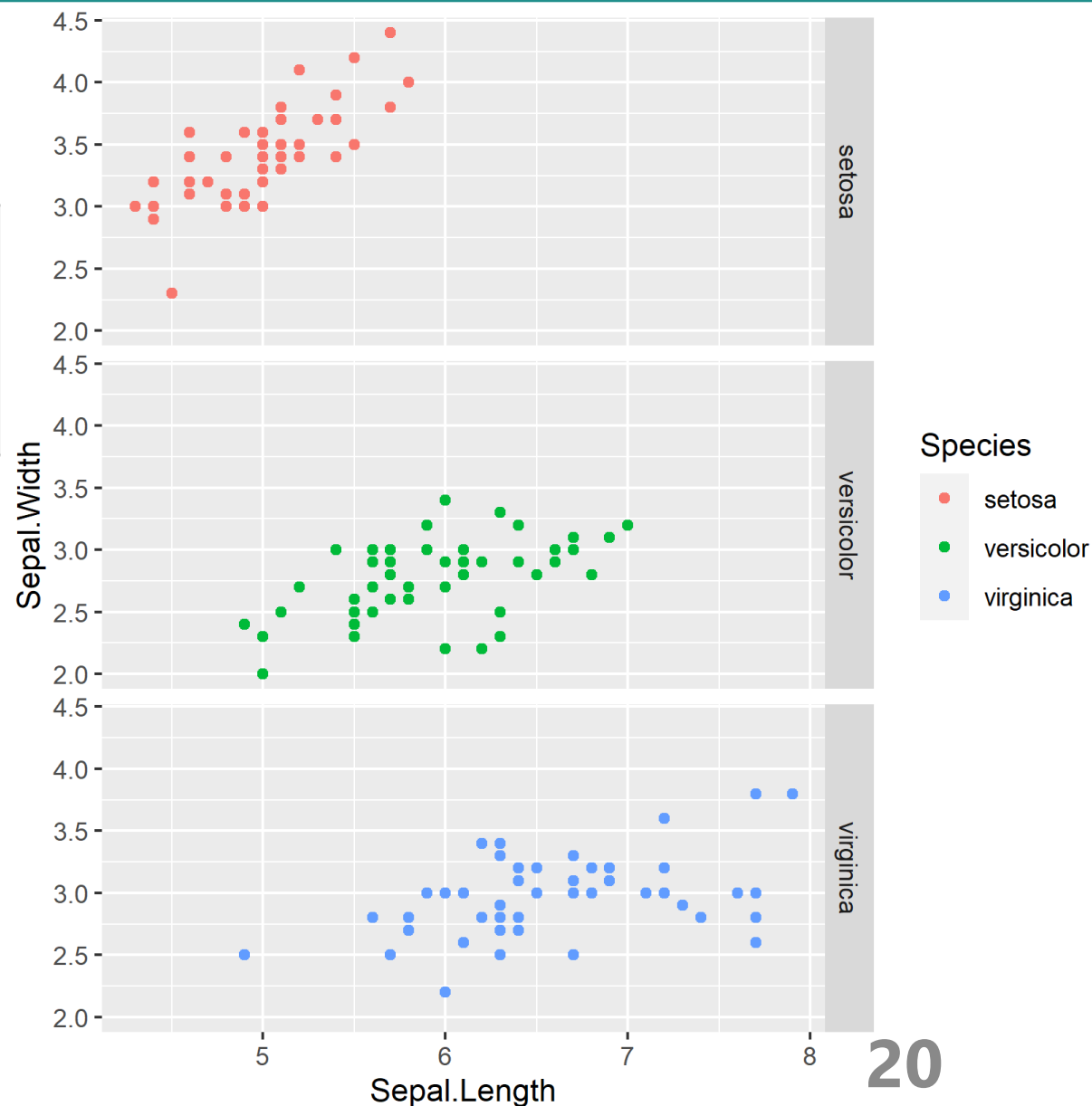
- **alpha** (ドットの透明度 ; 0~1)
- **color** (ドットの境界線の色)
- **fill** (ドットの内側の色)
- **shape** (ドットの種類)
- **size** (ドットの大きさ)



サブプロット（垂直方向）

□facet_grid(分割したい変数~.)関数

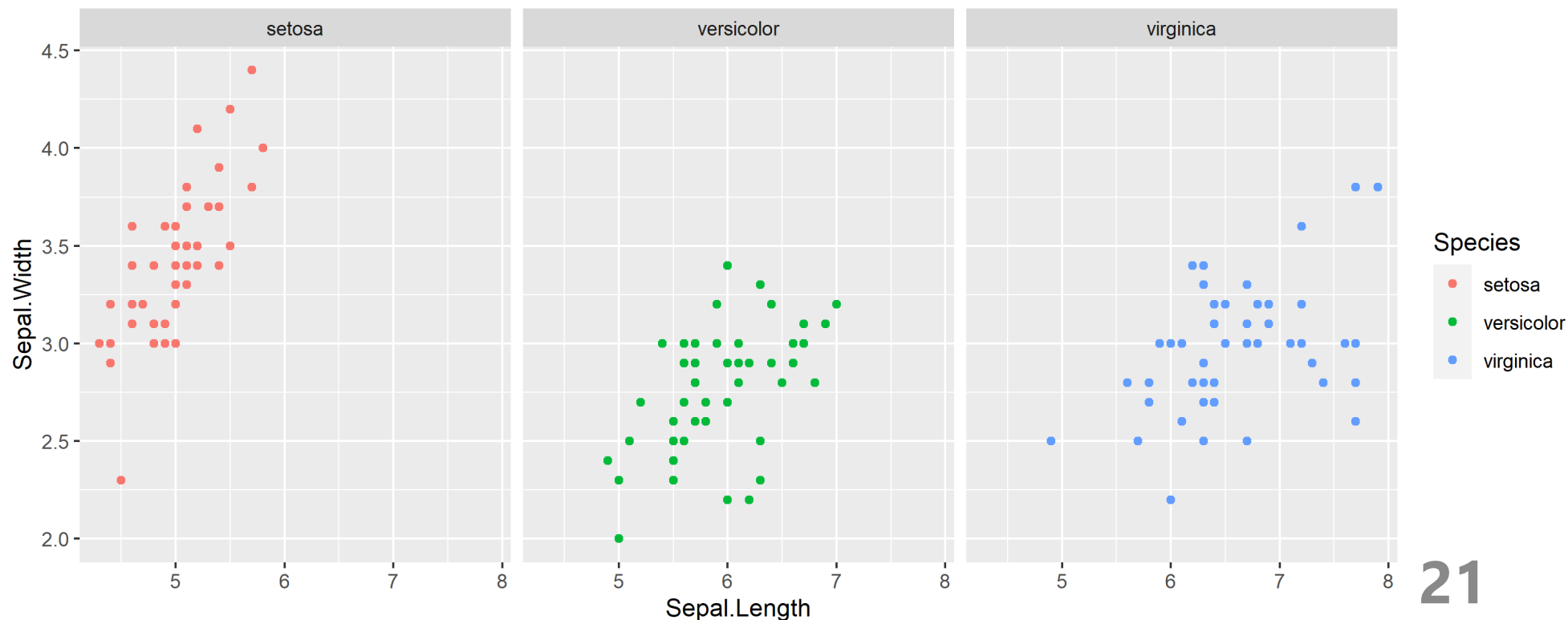
```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,  
                 color = Species)) +  
  geom_point() +  
  facet_grid(Species~.)
```



サブプロット (水平方向)

□ `facet_grid(.~分割したい変数)関数`

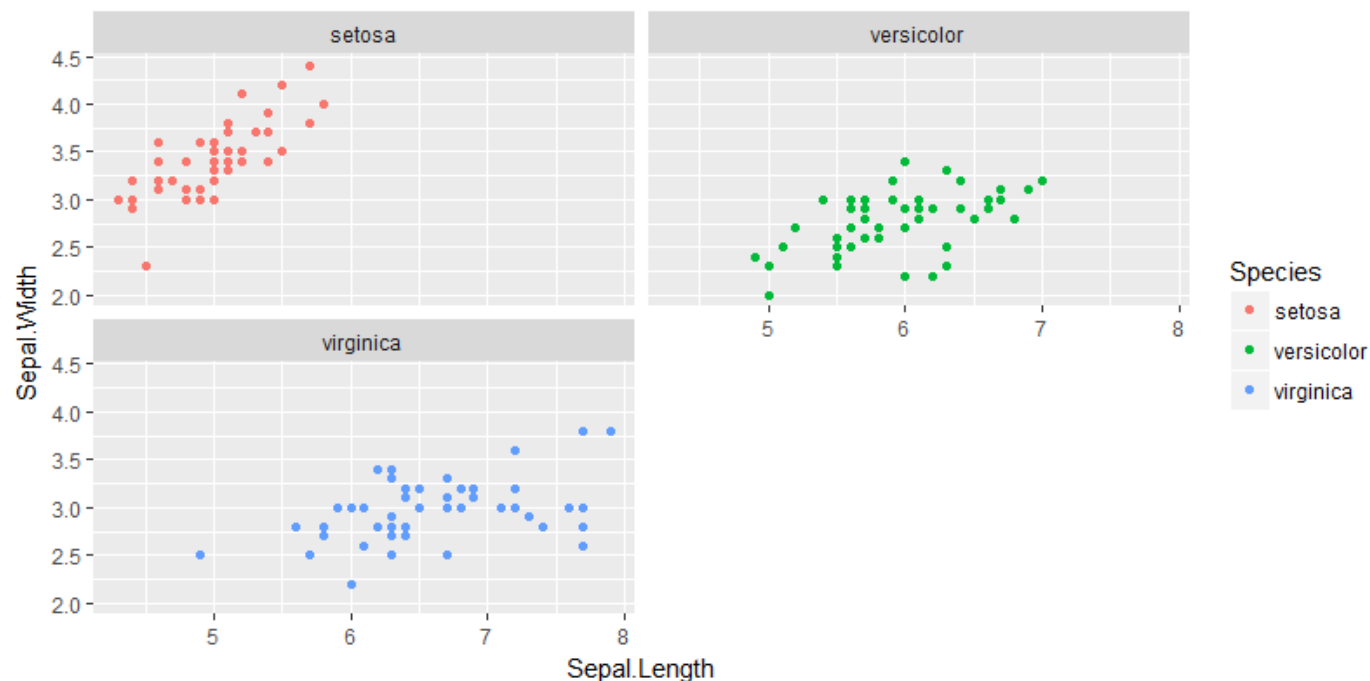
```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +  
  geom_point() +  
  facet_grid(.~Species)
```



サブプロット（指定した列数で）

□`facet_wrap(~分割したい変数, ncol=...)`関数

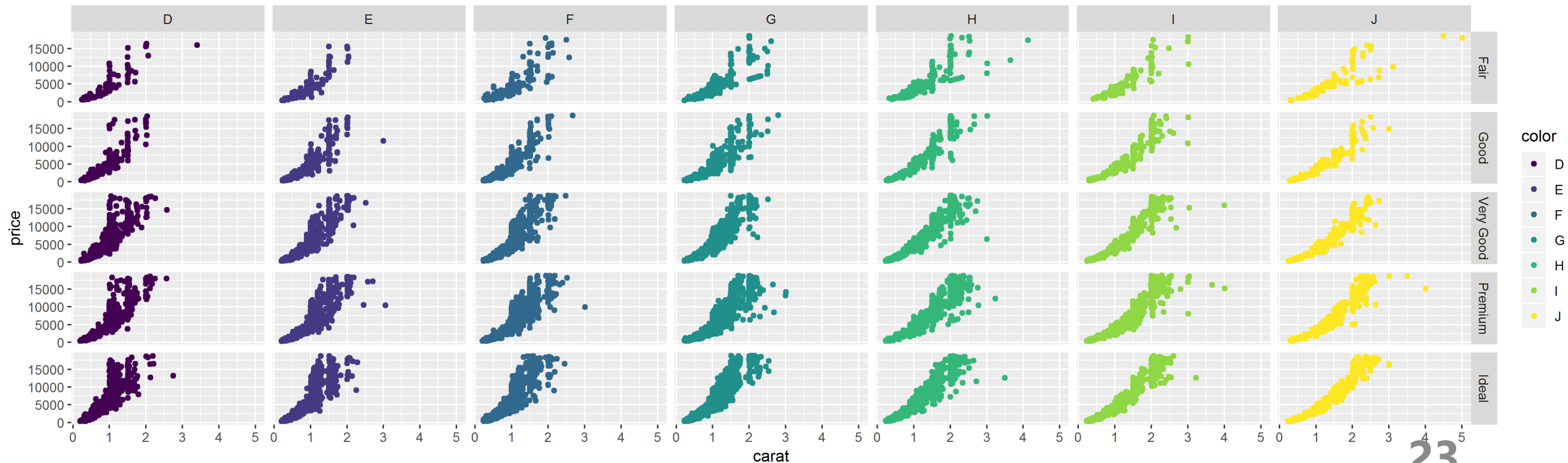
```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +  
  geom_point() +  
  facet_wrap(~Species, ncol = 2)
```



サブプロット (二要因の組み合わせ)

□facet_grid(変数1 ~ 変数2)関数

```
ggplot(diamonds, aes(x = carat, y = price, color = cut)) +  
  geom_point() +  
  facet_grid(cut~color)
```



sleep : 睡眠薬と睡眠時間変化のデータ

□参加者に2種類の薬(睡眠薬/偽薬)を投与して、睡眠時間の変化を検討

- ・ 10名の参加者に 2 種類の薬両方を投与 (対応ありのデータ)

睡眠時間の増加 (extra)	投与された薬 (group)	参加者のID (ID)
0.7	1 (偽薬)	1
-1.6	1 (偽薬)	2
...	...	...
0.0	1 (偽薬)	9
2.0	1 (偽薬)	10
1.9	2 (睡眠薬)	1
0.8	2 (睡眠薬)	2
...	...	...
4.6	2 (睡眠薬)	9
3.4	2 (睡眠薬)	10



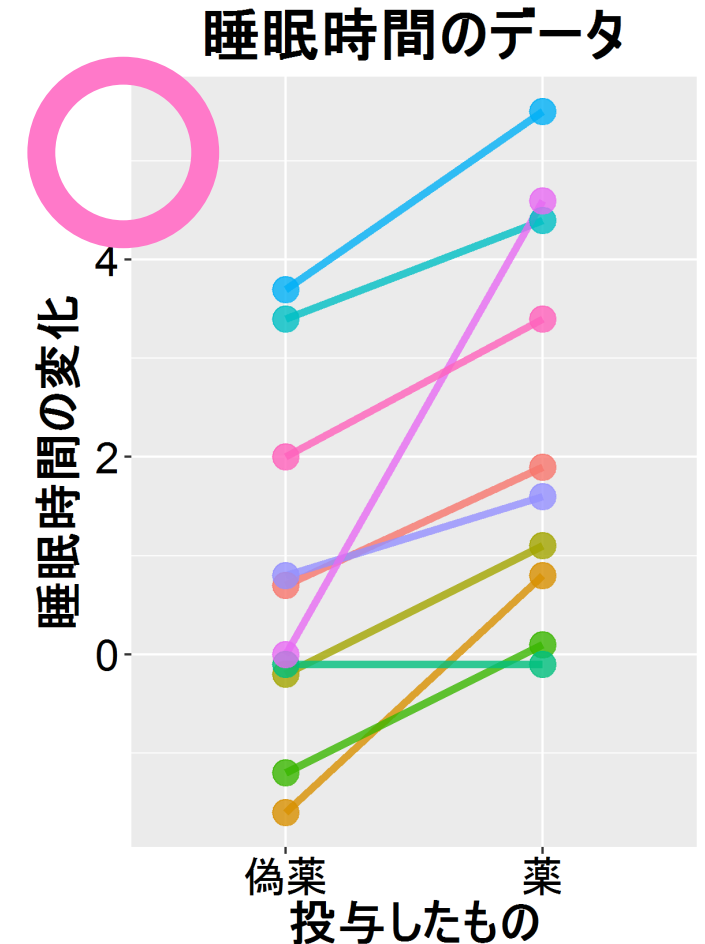
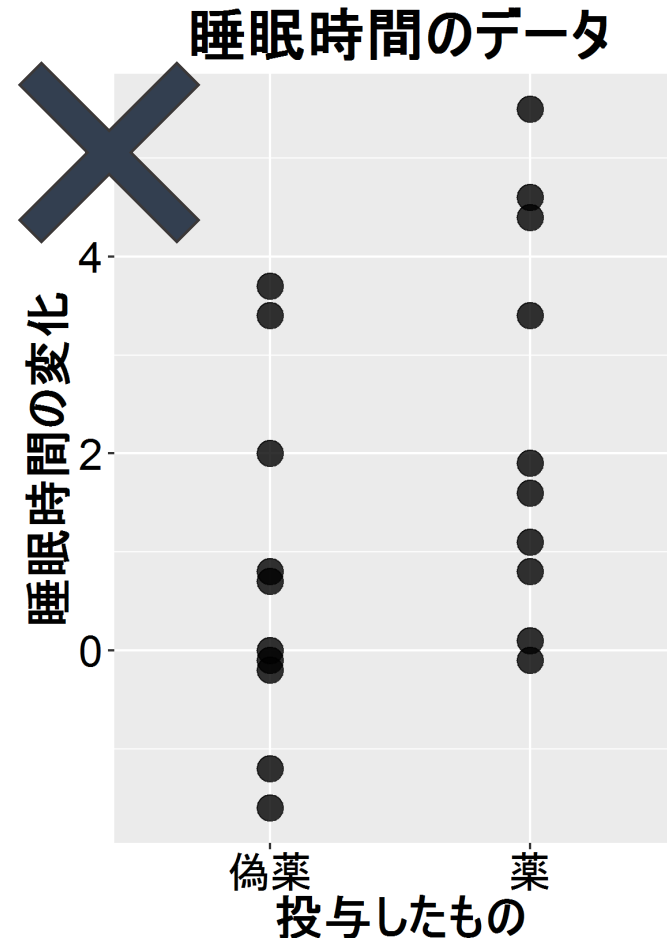
対応のあるデータの可視化：線グラフ

□対応のある観測値同士を線で結ぶ

- ・「対応」を表現

睡眠障害患者への薬投与後の
睡眠時間の変化

ID	偽薬	薬
1	0.7	1.9
2	-1.6	0.8
3	-0.2	1.6
⋮	⋮	⋮
9	0	4.6
10	2	3.4

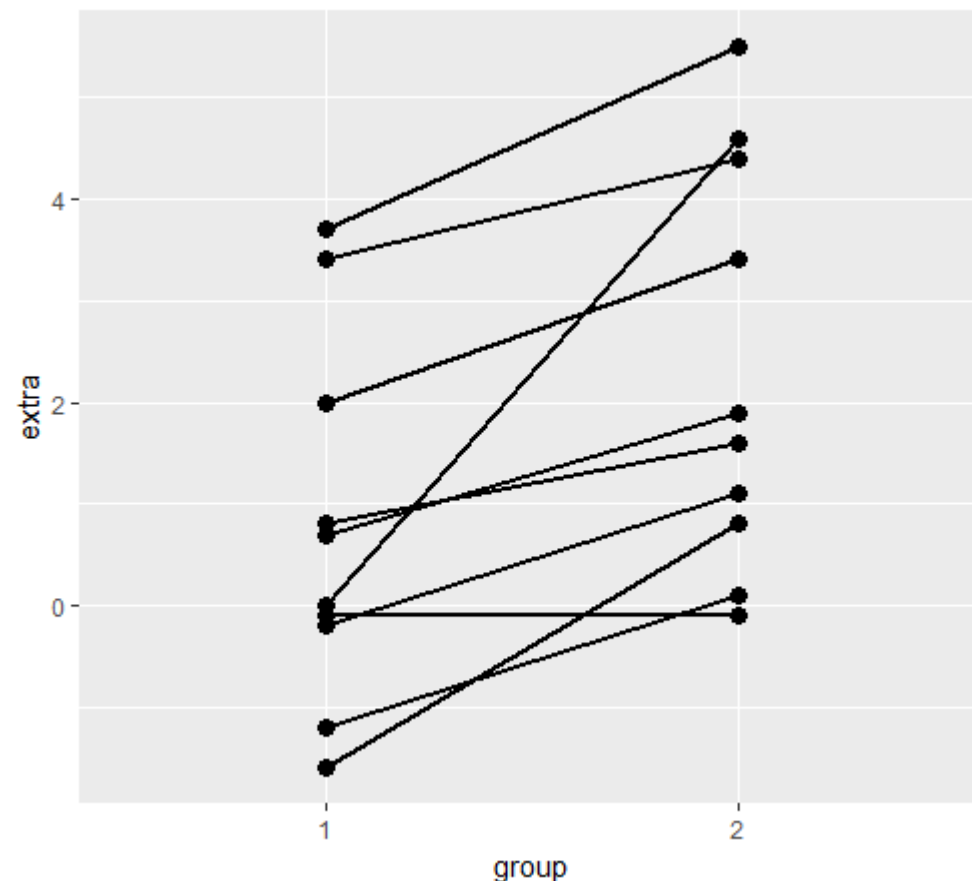


対応のあるデータの可視化

□ 散布図 + 折れ線グラフで、観測値と対応関係を可視化

- aes()内部でgroup="対応関係を表す変数"

```
ggplot(sleep, aes(x = group, y = extra, group = ID)) +  
  geom_line(lwd = 1) +  
  geom_point(size = 3)
```

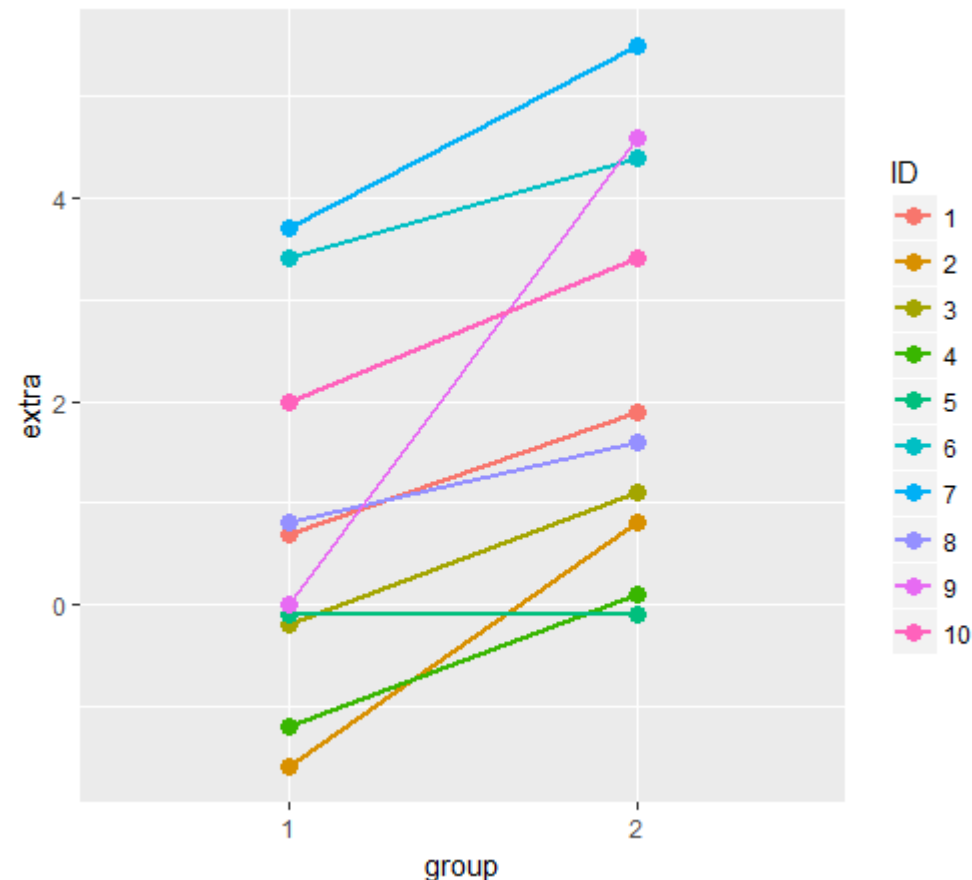


対応のあるデータの可視化

□ 散布図 + 折れ線グラフで、観測値と対応関係を可視化

- aes()内部でgroup="対応関係を表す変数"
 - colorも指定することで色分けもできる。

```
ggplot(sleep, aes(x = group, y = extra, group = ID,  
                  color = ID)) +  
  geom_line(lwd = 1) +  
  geom_point(size = 3)
```



**可視化を通して
データの傾向を把握する**

ダイヤモンドの価格はどうか決まる？

□ダイヤモンドのデータ

- ダイヤモンド53940個のカラット・形状・色・価格のデータ
 - ・「価格に影響を与える要因」に関する仮説を考え、それを可視化してみよう

	カラット	形状	色	透明度	価格
1	0.23	Ideal	E	SI2	326
2	0.21	Premium	E	SI1	326
3	0.23	Good	E	VS1	327
4	0.29	Premium	I	VS2	334
⋮	⋮	⋮	⋮	⋮	⋮
53938	0.70	Very Good	D	SI1	2757
53939	0.86	Premium	H	SI2	2757
53940	0.75	Ideal	D	SI2	2757



ダイヤモンドの価格はどう決まる？

□もともとのデータサイズが大きすぎるので、サンプリング

- もともとのデータの10%分のデータを使用しましょう
 - PCのスペックによっては、処理に時間がかかる場合があるため

●sample_frac()関数

- データの行数を指定した割合にランダムに削減

●以下のコードを入力

- set.seed(1234)
sample_frac(diamonds, 0.1) -> df

```
set.seed(1234) # 乱数の種  
sample_frac(diamonds, 0.1) -> df
```



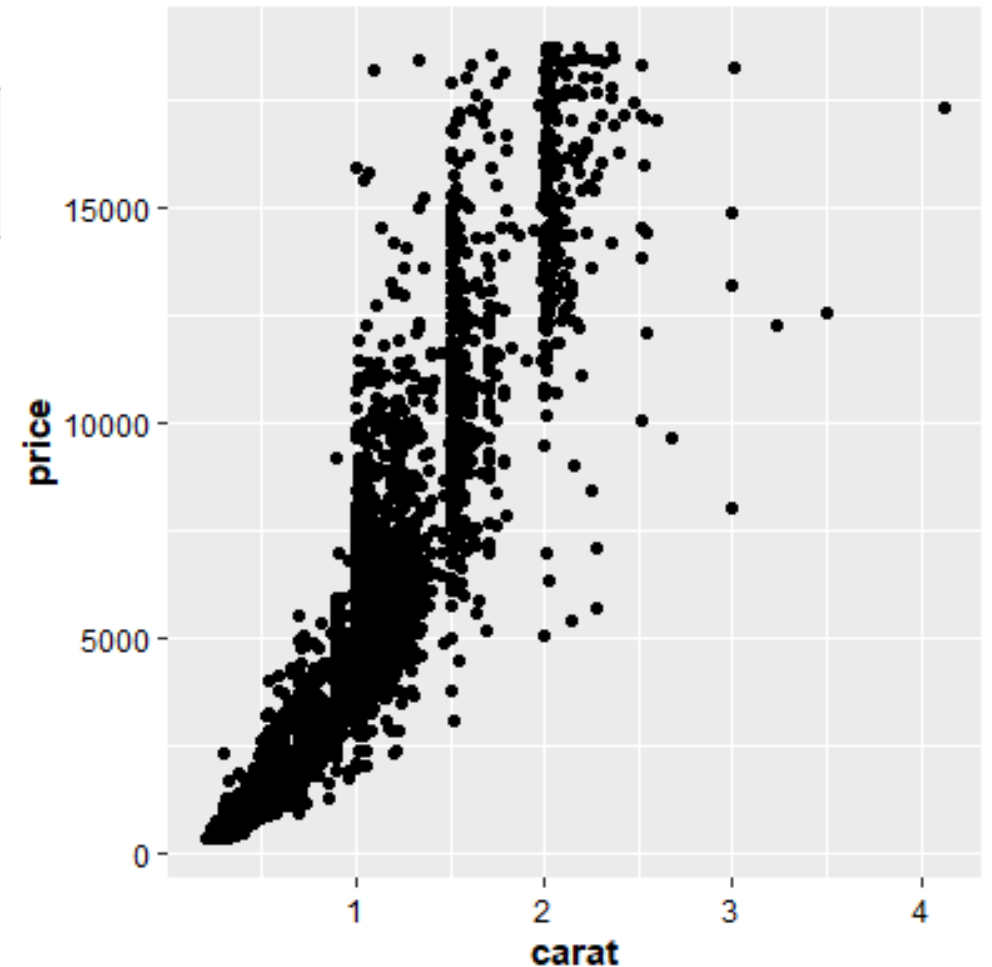
ダイヤモンドの価格はどう決まる？

□仮説①「重いダイヤは高いだろう」

```
ggplot(df, aes(x = carat, y = price)) +  
  geom_point()
```

価格

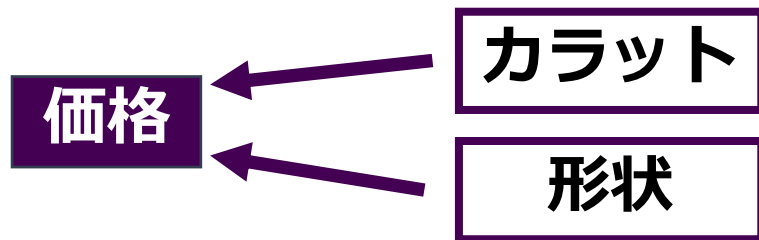
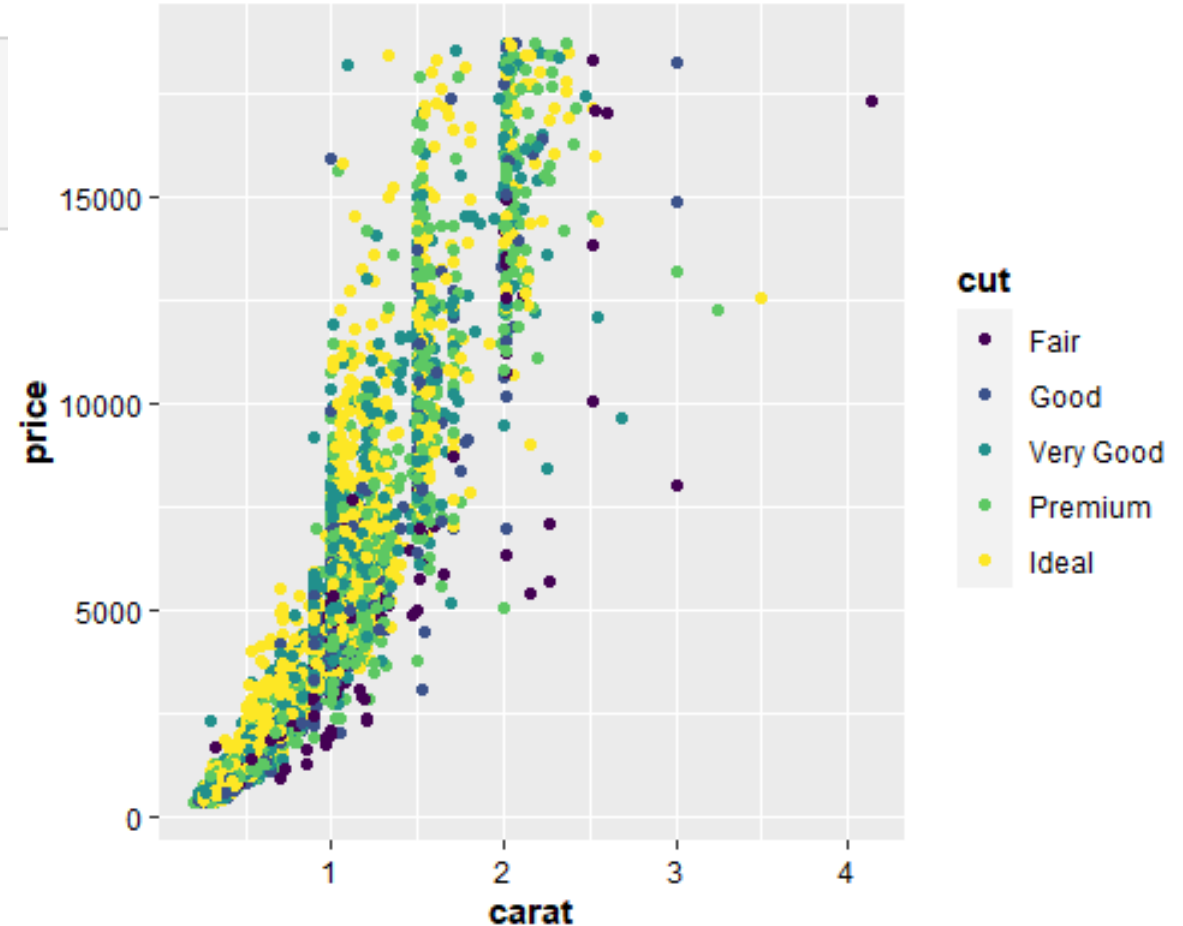
カラット



ダイヤモンドの価格はどう決まる？

□仮説② 「よい形状で重いダイヤは高いだろう」

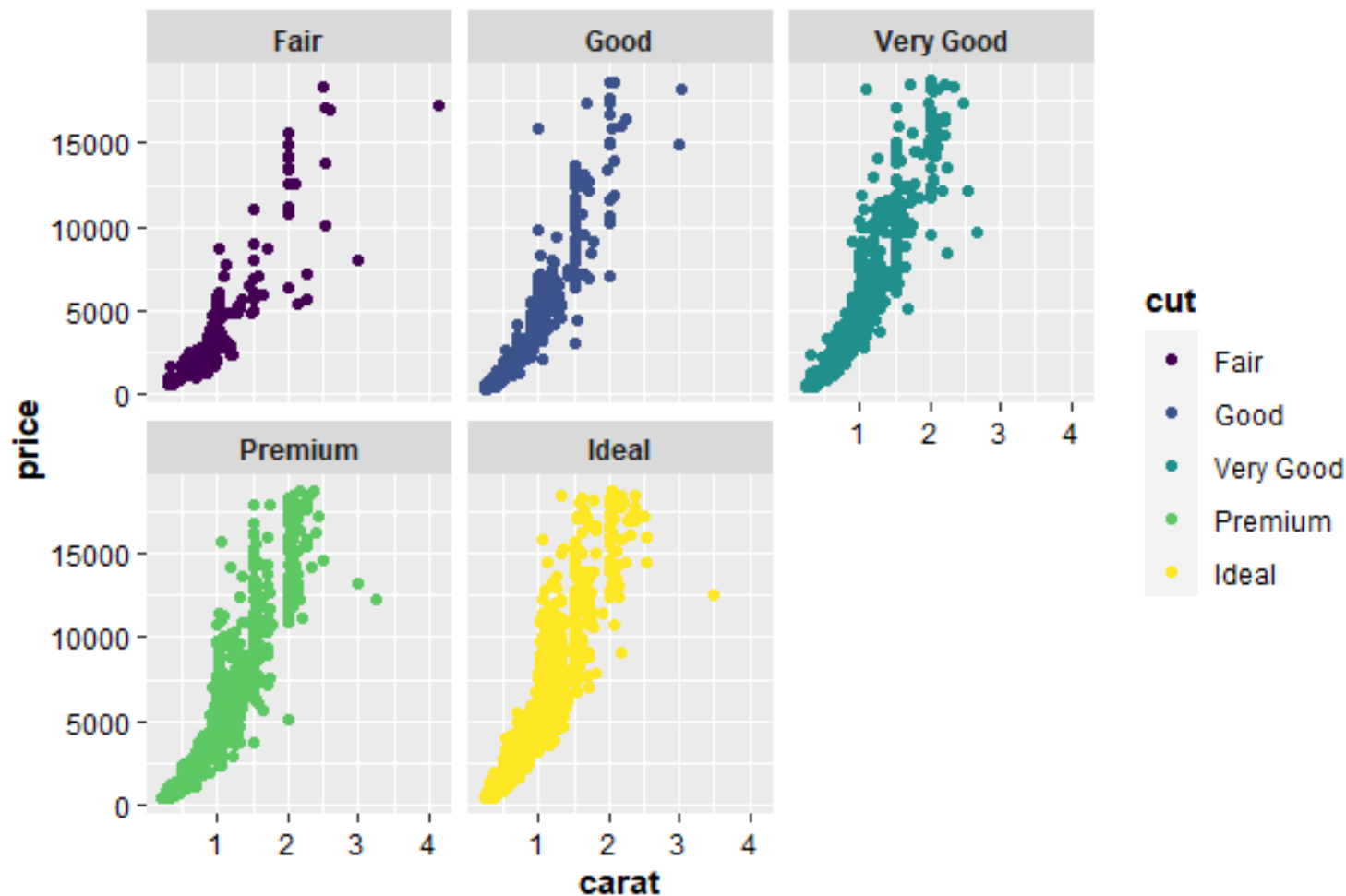
```
ggplot(df, aes(x = carat, y = price,  
               color = cut)) +  
  geom_point()
```



ダイヤモンドの価格はどうか決まる？

□仮説②「よい形状で重いダイヤは高いだろう」

```
ggplot(df, aes(x = carat, y = price,  
               color = cut)) +  
  geom_point() +  
  facet_wrap(~cut, ncol = 3)
```



価格

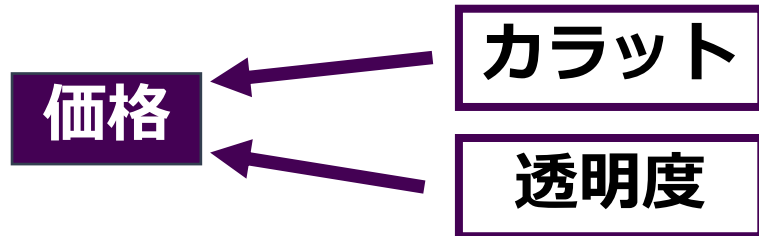
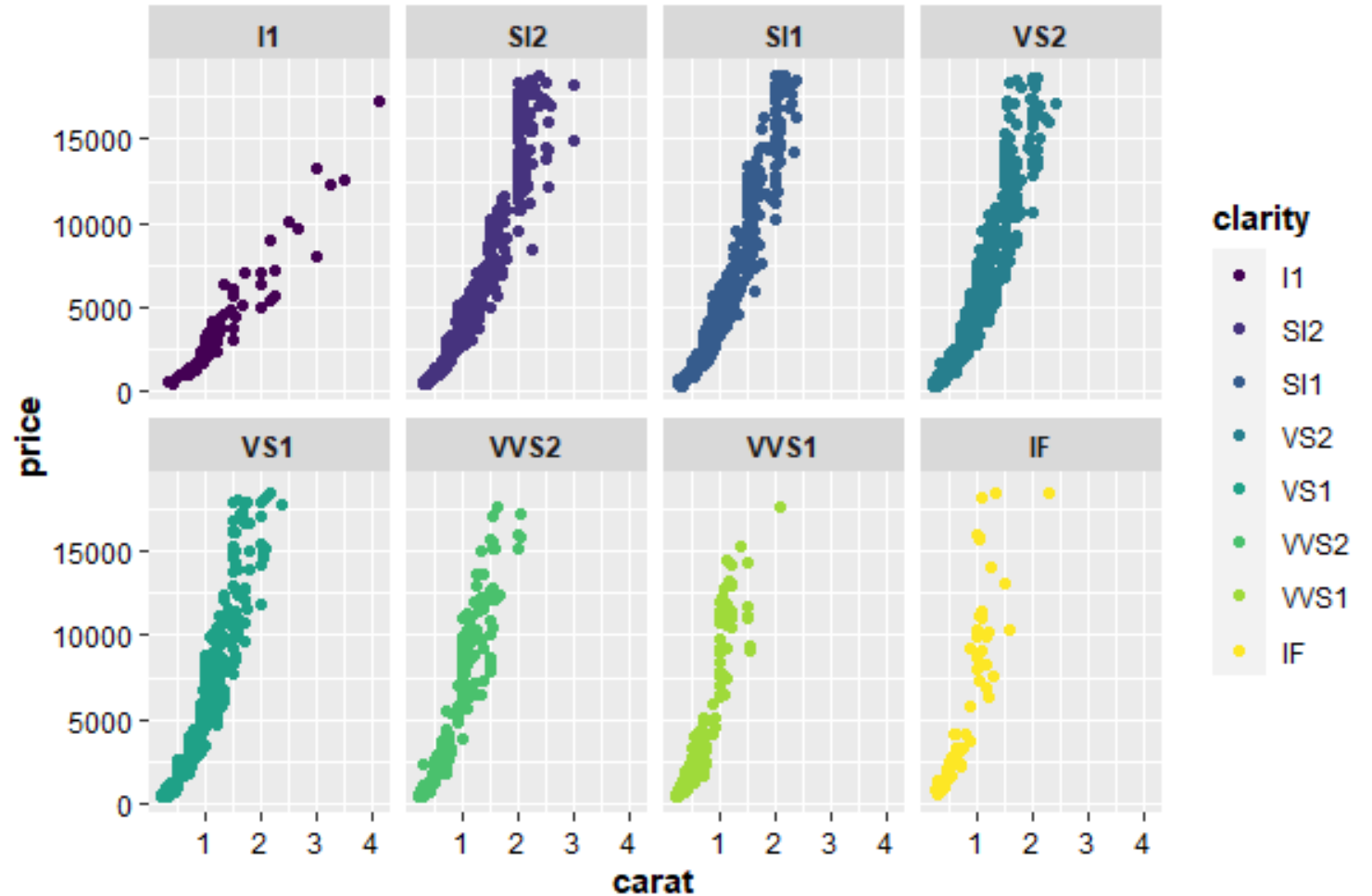
カラット

形状

ダイヤモンドの価格はどう決まる？

□仮説③「透明度が高く重いダイヤは高いだろう」

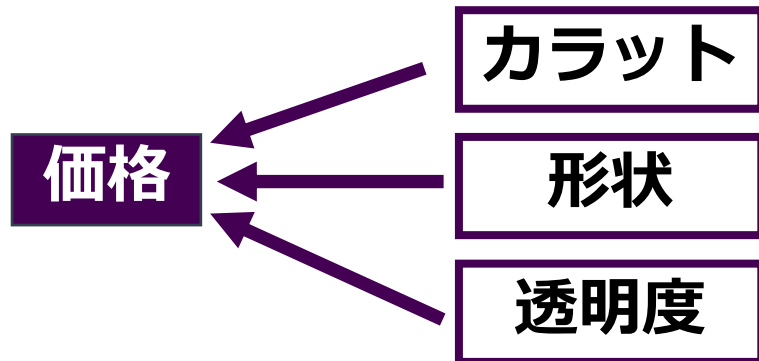
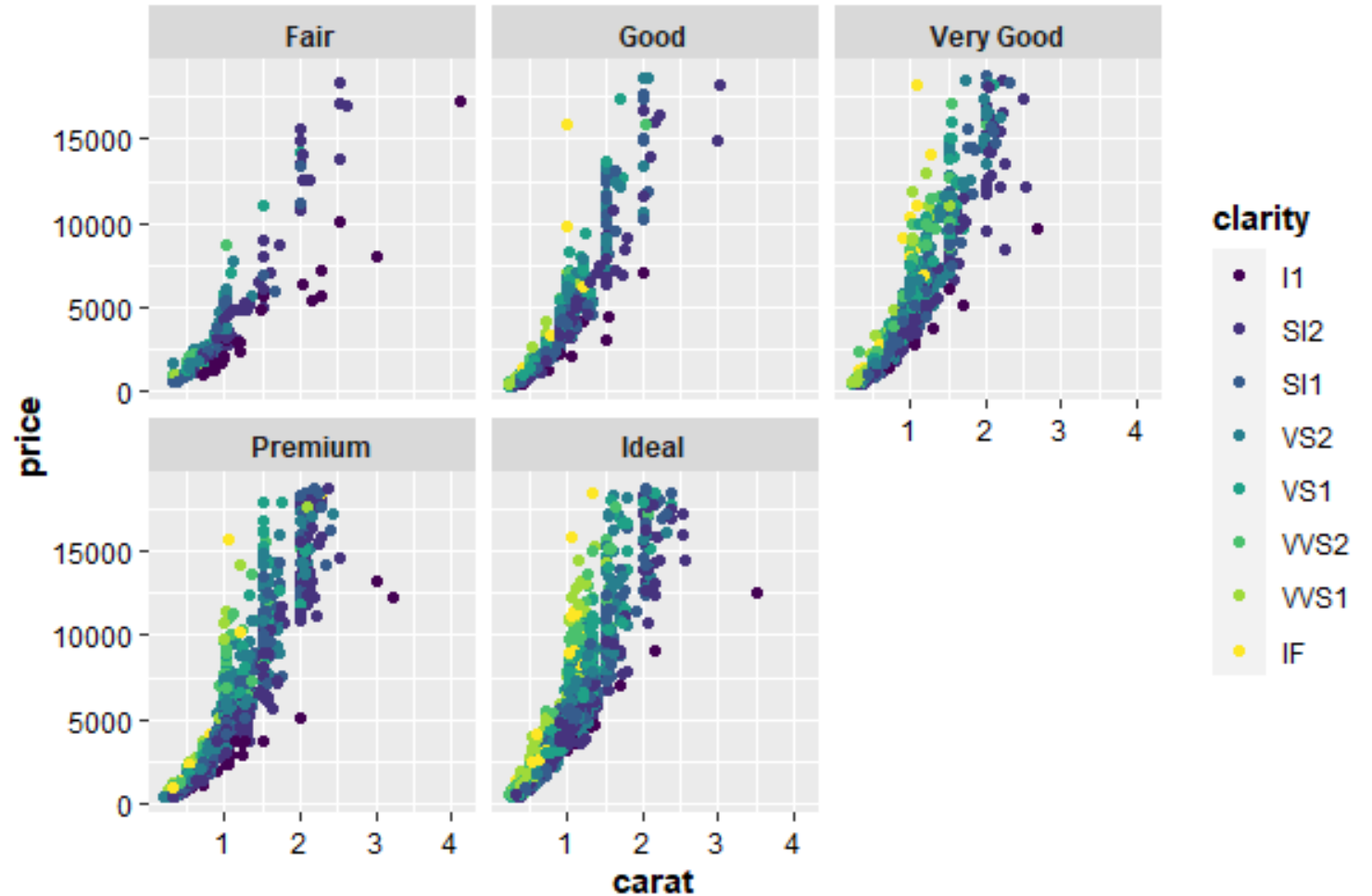
```
ggplot(df, aes(x = carat, y = price,  
               color = clarity)) +  
  geom_point() +  
  facet_wrap(~clarity, ncol = 4)
```



ダイヤモンドの価格はどう決まる？

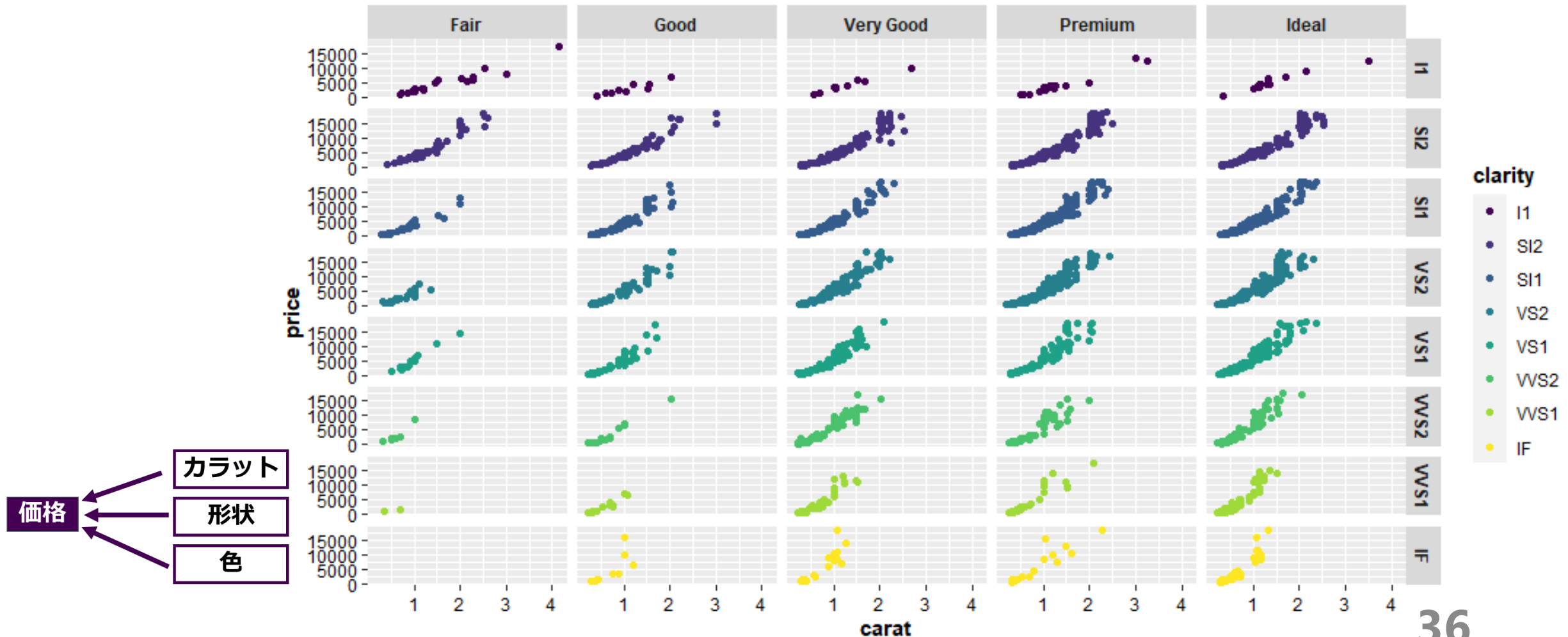
□仮説④「形状が良く，透明度が高く，重いダイヤは高いだろう」

```
ggplot(df, aes(x = carat, y = price,  
               color = clarity)) +  
  geom_point() +  
  facet_wrap(~cut, ncol = 3)
```



ダイヤモンドの価格はどう決まる？

□仮説④「形状が良く，透明度が高く，重いダイヤは高いだろう」



質問など