

# R でグラフ作成

ggplot2 入門



## 本日のメニュー

---

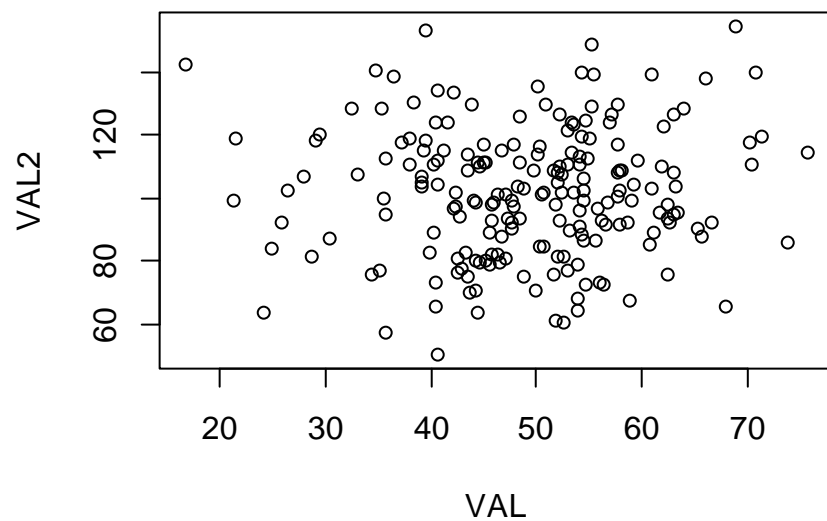
1. イントロ
2. ggplot2 事始
3. グラフの作成例
4. グラフのカスタマイズ例



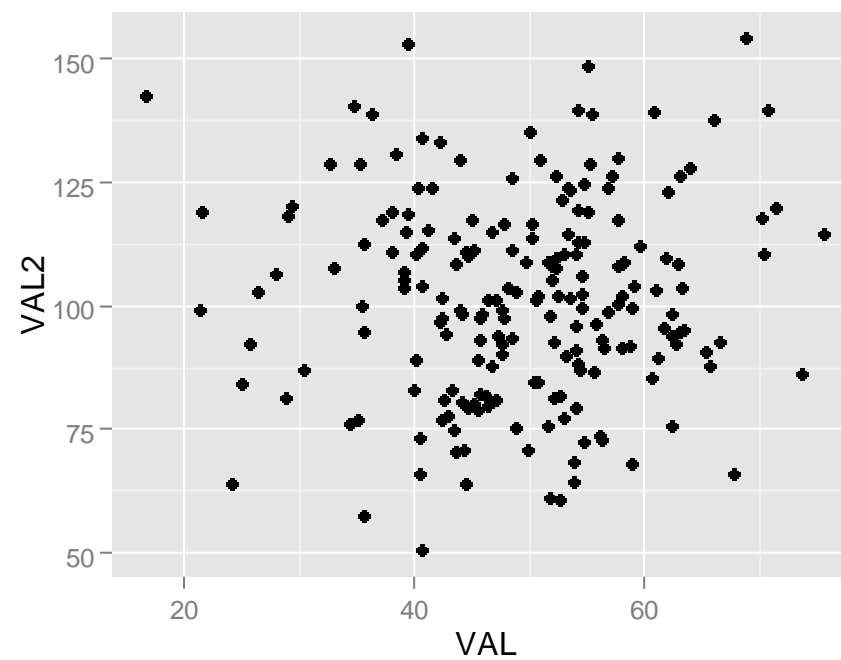
# Traditional vs. ggplot2

---

**Traditional**



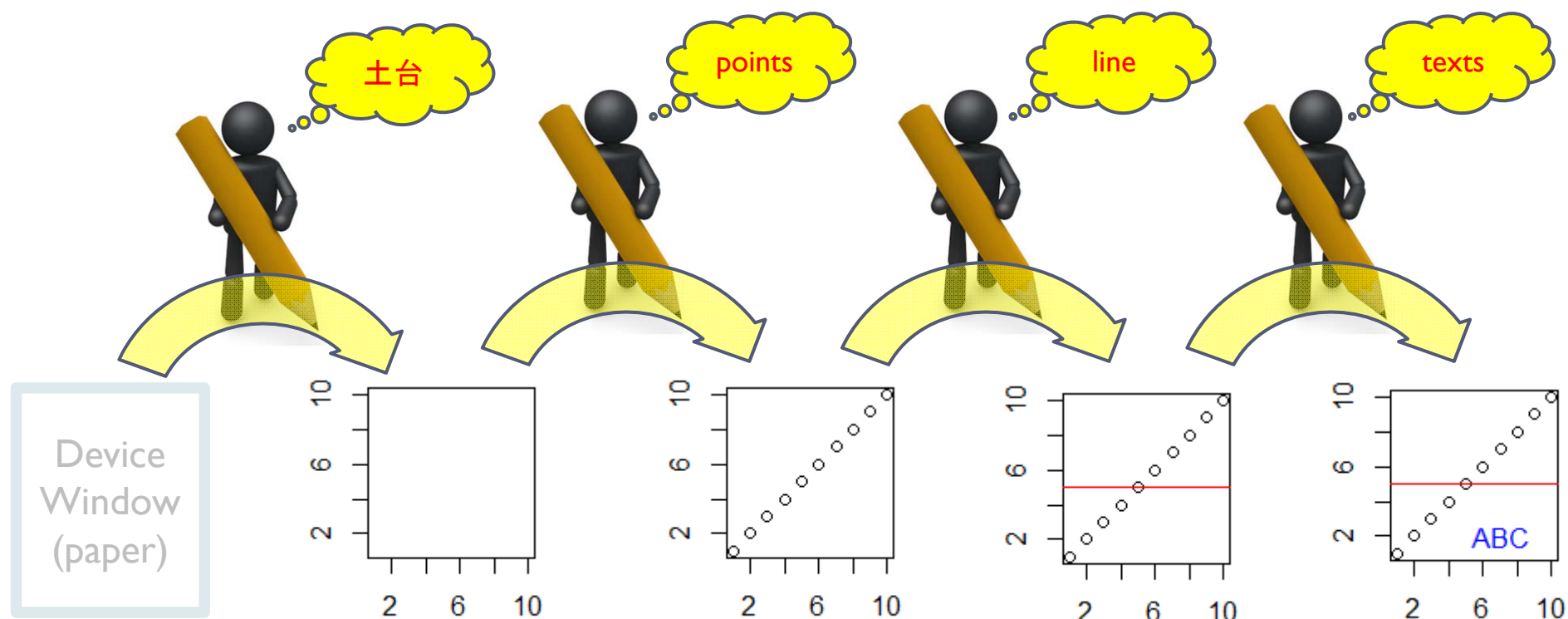
**ggplot2**





## Traditional なグラフ

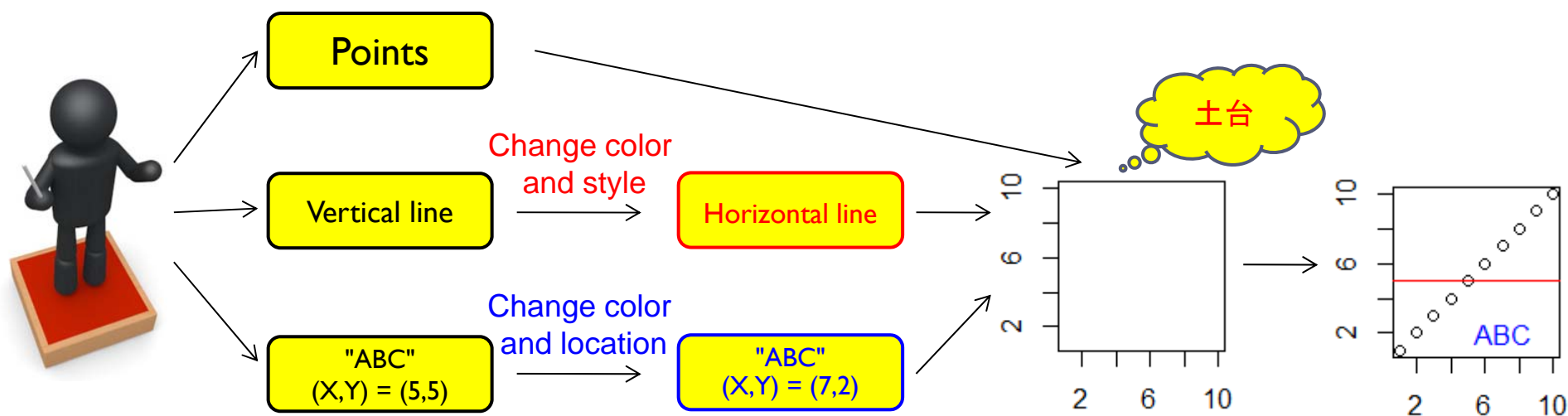
- ▶ 「ペンと紙を使って描く」スタイル
- ▶ 土台となるグラフを作った後、点や線や文字等を追記するスタイル
- ▶ 一度描いたグラフを、別のグラフを描くために再利用することは不可





## ggplot2 で作成するグラフ

- ▶ 「グラフに関するオブジェクト」を使って描くスタイル
- ▶ `ggplot()` で土台となるグラフを作った後、点や線や文字に関するオブジェクトを `geom_XXX()` 等で作成し、必要に応じてカスタマイズした後、土台に貼り付けるスタイル（オブジェクトは再利用が出来る）
- ▶ コマンド（文法）が非常に体系的で洗練されている





## 初期設定

---

- ▶ `ggplot2` パッケージ※のインストールと呼び出し

```
> install.packages("ggplot2", dep=T)
> library(ggplot2)
```

- ▶ 使用するデータ①：ToothGrowth

- ▶ 豚にビタミンC 又はオレンジジュースを与えた時の歯の長さを調べたデータ
- ▶ `len`: 長さ, `supp`: VC(ビタミンC) 又は OJ(オレンジジュース),  
`dose`: 用量(0.5, 1.0, 2.0)

```
> head(ToothGrowth, n=3)
  len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
```



## 本日のメニュー

---

1. イントロ

2. **ggplot2 事始**

3. グラフの作成例

4. グラフのカスタマイズ例



## ggplot2 事始

---

```
> base <- ggplot(ToothGrowth, aes(x=dose, y=len))
```

- ▶ 関数 `ggplot()` : プロットオブジェクト (土台) を作成する
  - ▶ `ggplot(データフレーム名, aes(x 座標の変数, y 座標の変数, 審美的属性))`
- ▶ 関数 `aes()` : `x 座標の変数`, `y 座標の変数`, [審美的属性](#)※を指定する  
(全て指定する必要は無い)
- ▶ 審美的属性 : 色, 大きさ, 線の種類, プロット点の形等
- ▶ 上記はただの土台 (変数 `base`) を作成しただけなので, これだけではグラフを作成したことにならない





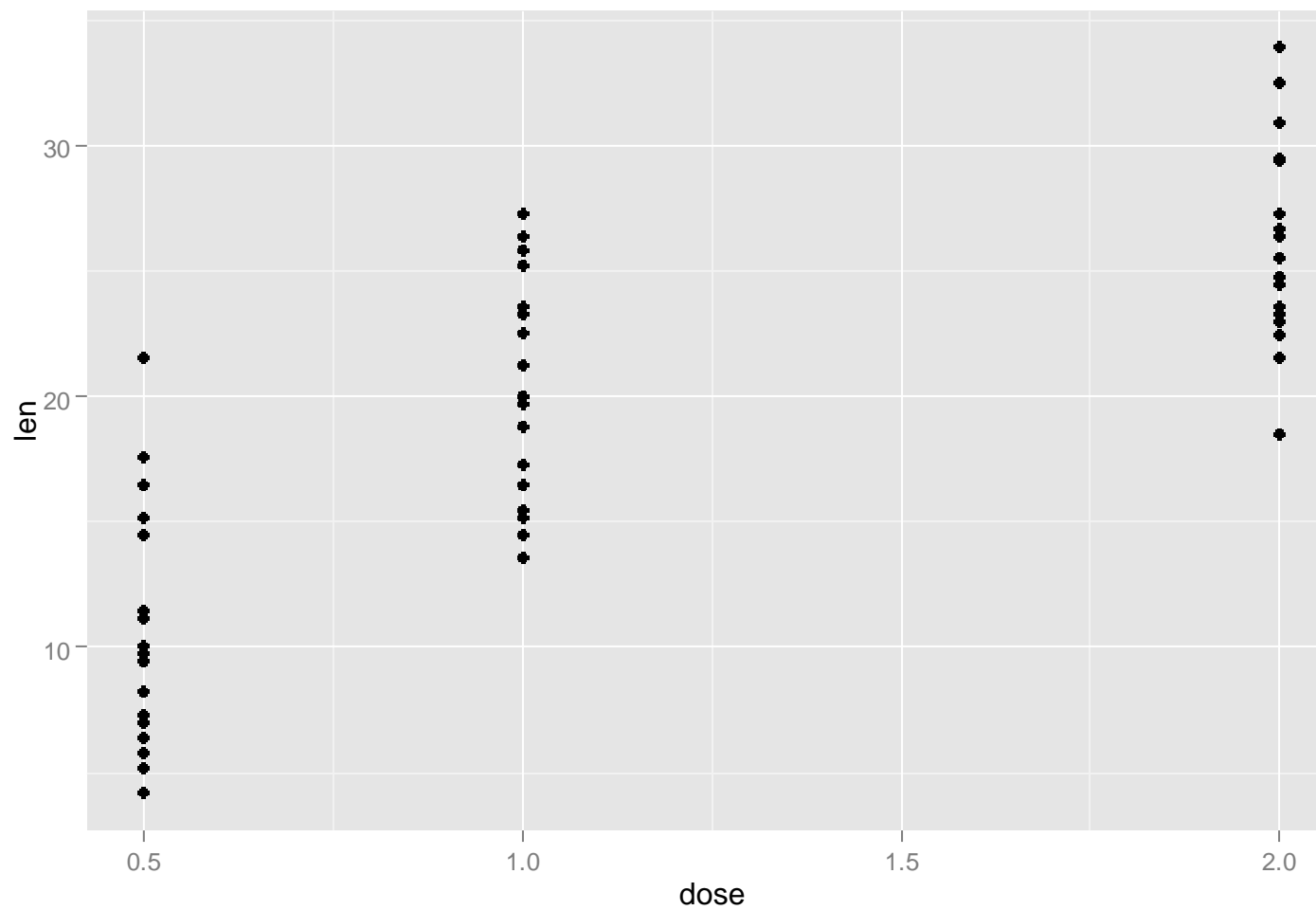
## ggplot2 事始

```
> points <- base + geom_point()  
> plot(points)  
> base + geom_point() # plot(points) と同じ働き
```

- ▶ 先程作成した土台（変数 **base**）にレイヤー※を追加した変数を作成する
- ▶ レイヤーとは「データに関連する要素」のことで，例えば上記の関数 **geom\_point()** では「点レイヤー」を追加，すなわち「グラフの種類は散布図ですよ」という属性を変数 **base** に与えていることになる
- ▶ レイヤーの種類は，例えば以下がある
  - ▶ グラフの種類（関数 **geom\_XXX()**）
  - ▶ 統計的変換（関数 **stat\_XXX()**）
- ▶ 最後に関数 **plot()** に変数 **points** を指定することでグラフが表示される



## 散布図が完成





## 関数 `geom_XXX` の種類①

関数	種類
<code>geom_abline</code>	直線（切片と傾きを指定）
<code>geom_area</code>	曲線下面積（ <b>AUC</b> ）のプロット
<code>geom_bar</code>	棒グラフ
<code>geom_bin2d</code>	ヒートマップ
<code>geom_blank</code>	空白（何も表示しない）
<code>geom_boxplot</code>	箱ひげ図
<code>geom_contour</code>	等高線プロット
<code>geom_crossbar</code>	箱ひげ図の箱だけのようなプロット
<code>geom_density</code>	密度曲線
<code>geom_density2d</code>	2次元密度推定
<code>geom_errorbar</code>	誤差に関するエラーバー
<code>geom_freqpoly</code>	頻度ポリゴン
<code>geom_hex</code>	六角形のヒートマップ（ <code>stat_binhex()</code> を参照）
<code>geom_histogram</code>	ヒストグラム
<code>geom_hline</code>	水平線を描く



## 関数 `geom_XXX` の種類②

関数	種類
<code>geom_jitter</code>	データをズラす（点等の重なりを緩和するため）
<code>geom_line</code>	線を描く
<code>geom_linerange</code>	箱ひげ図の箱を線で表したようなプロット
<code>geom_map</code>	地図にヒートマップを追記する
<code>geom_path</code>	データフレームのデータの上から順に線で繋ぐ
<code>geom_point</code>	散布図
<code>geom_pointrange</code>	平均値±標準偏差のプロット
<code>geom_polygon</code>	ポリゴンプロット
<code>geom_quantile</code>	箱ひげ図の連続変数版（ <code>stat_quantile()</code> を参照）
<code>geom_raster</code>	<code>geom_tile</code> のハイパフォーマンス版
<code>geom_rect</code>	矩形を描く
<code>geom_ribbon</code>	折れ線グラフにバンド幅を加えたプロット
<code>geom_rug</code>	ラグプロット（ <code>x / y</code> 軸にデータを表す線を追記）
<code>geom_segment</code>	線分を描く
<code>geom_smooth</code>	平滑線



## 関数 geom\_XXX の種類③

関数	種類
geom_step	階段関数
geom_text	文字列を追記
geom_tile	2次元データの密度をタイルで表示
geom_violin	バイオリンプロット
geom_vline	垂直線を追記

```
> ?geom_point
```

- ▶ 各関数で指定する審美的属性（関数 `aes()`）の引数はヘルプ参照

```
R Help on 'geom_point'
ファイル 編集

Aesthetics:

'geom_point' understands the following aesthetics (required
aesthetics are in bold):

* 'x'
* 'y'
* 'alpha'
* 'colour'
* 'fill'
* 'shape'
* 'size'
```

- ▶ 例えば、関数 `geom_point()` で指定出来る引数は以下：  
`x`, `y`, `alpha`, `color`, `fill`,  
`shape`, `size`  
(`colour` は `color` でも可)



## 関数 `stat_XXX` の種類①

関数	種類
<code>stat_bin</code>	データの bin の幅 (ヒストグラムの棒の横幅)
<code>stat_bin2d</code>	矩形 (rectangle) の中のデータ数
<code>stat_bindot</code>	ドットプロットのための bin データ
<code>stat_binhex</code>	六角形のヒートマップを描くためのデータ
<code>stat_boxplot</code>	箱ひげ図で出てくる要約統計量
<code>stat_contour</code>	等高線
<code>stat_density</code>	1次元の密度推定
<code>stat_density2d</code>	2次元の密度推定
<code>stat_ecdf</code>	経験累積分布関数
<code>stat_identity</code>	確率楕円
<code>stat_function</code>	ユーザーが指定した関数 (で計算する)
<code>stat_identity</code>	データの変換をしない (データのまま)
<code>stat_qq</code>	QQ プロット



## 関数 `stat_XXX` の種類②

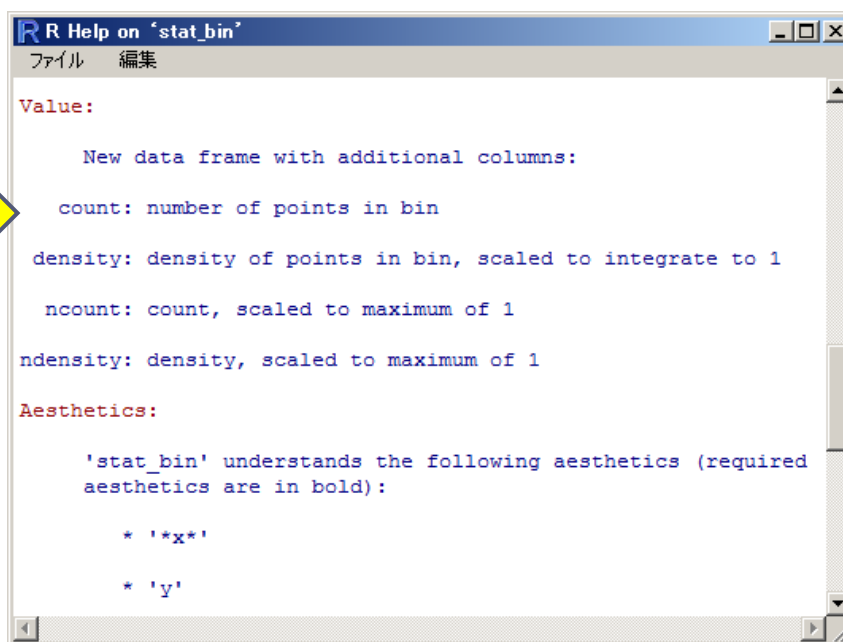
関数	種類
<code>stat_quantile</code>	分位点
<code>stat_smooth</code>	平滑化曲線
<code>stat_spoke</code>	極座標変換（ $x$ と $y$ の範囲を用いる）
<code>stat_sum</code>	同じ値のデータを合計する
<code>stat_summary</code>	データの要約統計量
<code>stat_summary2d</code>	ヒートマップの各矩形のデータ数
<code>stat_summary_hex</code>	ヒートマップの各六角形のデータ数
<code>stat_unique</code>	データの重複を除去
<code>stat_ydensity</code>	密度推定値（バイオリンプロット用）



## 関数 `stat_XXX` について

```
> ?stat_bin
```

- ▶ 各関数の返り値や「指定する審美的属性の引数」はヘルプを参照



```
R Help on 'stat_bin'
ファイル 編集

Value:

  New data frame with additional columns:

  count: number of points in bin

  density: density of points in bin, scaled to integrate to 1

  ncount: count, scaled to maximum of 1

  ndensity: density, scaled to maximum of 1

Aesthetics:

  'stat_bin' understands the following aesthetics (required
  aesthetics are in bold):

  * 'x'

  * 'y'
```

- ▶ 例えば、関数 `stat_bin()` の場合は以下：

関数 `stat_bin()` の返り値

`count`, `density`, `ncount`,  
`ndensity`

関数 `aes()` の引数

`x`, `y`





## 本日のメニュー

---

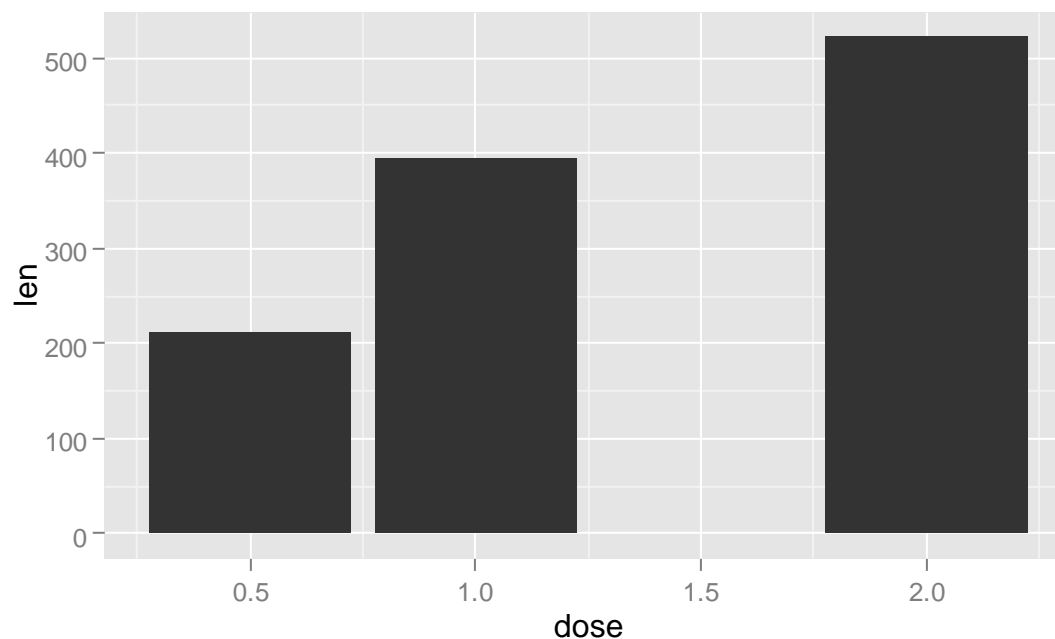
1. イントロ
2. ggplot2 事始
3. グラフの作成例
4. グラフのカスタマイズ例



## グラフの作成例：棒グラフ

```
> base <- ggplot(ToothGrowth, aes(x=dose, y=len))  
> base + geom_bar(stat="identity")
```

- ▶ **stat="identity"**：各用量の「歯の長さ」のデータを積み上げる  
→ 結果として合計をプロットしていることに相当する

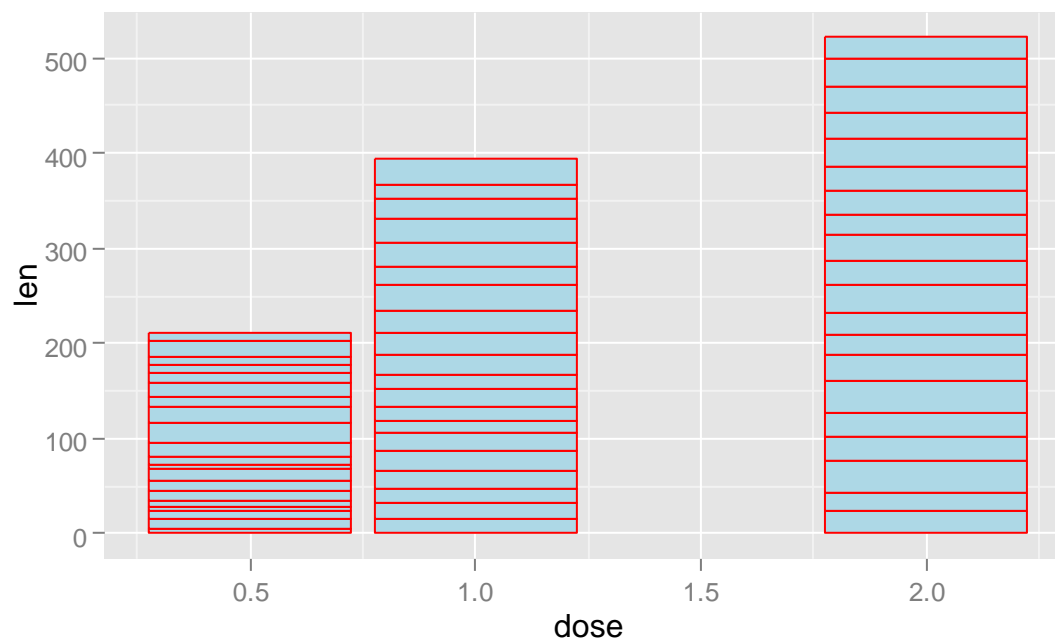




## グラフの作成例：棒グラフ

```
> base <- ggplot(ToothGrowth, aes(x=dose, y=len))  
> base + geom_bar(stat="identity", fill="lightblue",  
+   color="red")
```

- ▶ 棒の色を水色に、各データの枠を赤にする  
→ 各用量の「歯の長さ」のデータが積み上がっている様子が見て取れる





## グラフの作成例：棒グラフ

- ▶ 例えば「各用量の平均値の棒グラフ」を描きたい場合は、まず「各用量の平均値のデータフレーム」を作成する必要がある
- ▶ 方法はいくつかあるが、**doBy** パッケージの関数 **summaryBy()** で「各用量の平均値のデータフレーム」を作成することが出来る

```
> library(doBy)
> ( MEAN <- summaryBy(len ~ dose, data=ToothGrowth,
+   FUN=c(length,mean,sd)) )
```

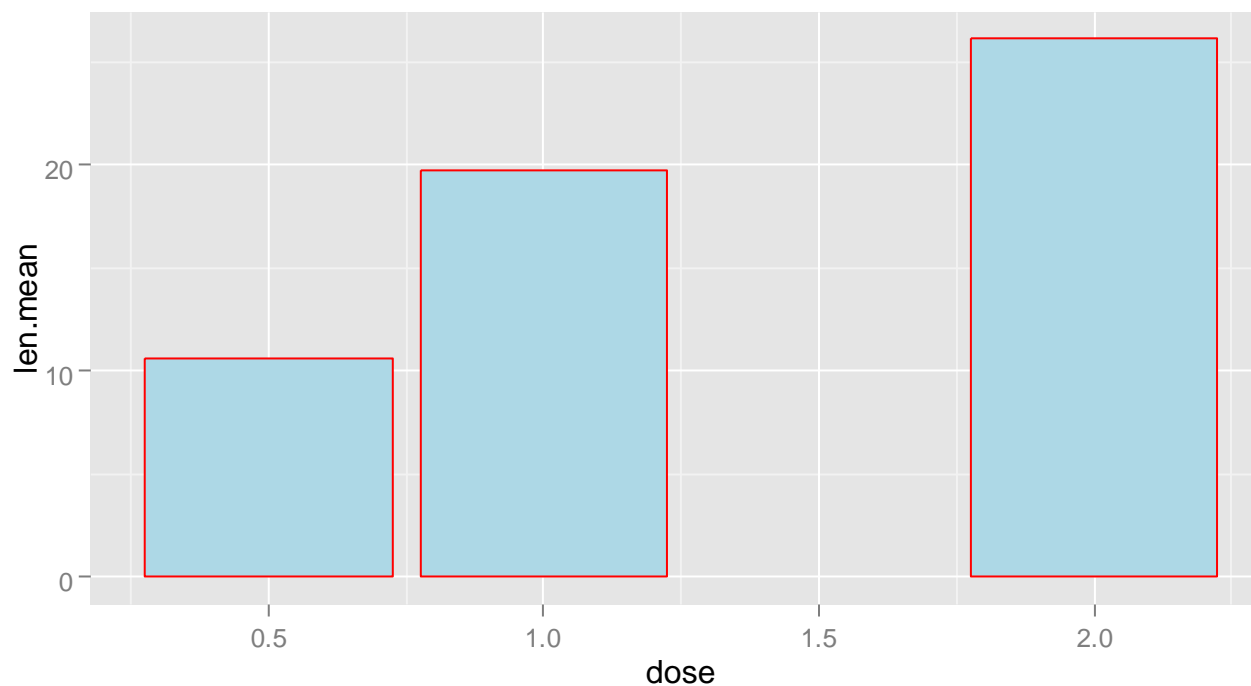
```
  dose len.length len.mean  len.sd
1  0.5          20  10.605  4.499763
2  1.0          20  19.735  4.415436
3  2.0          20  26.100  3.774150
```



## グラフの作成例：棒グラフ

```
> base2 <- ggplot(MEAN, aes(x=dose, y=len.mean))  
> base2 + geom_bar(stat="identity", fill="lightblue",  
+   color="red")
```

- ▶ 各用量の「歯の長さの平均値」に関する棒グラフ

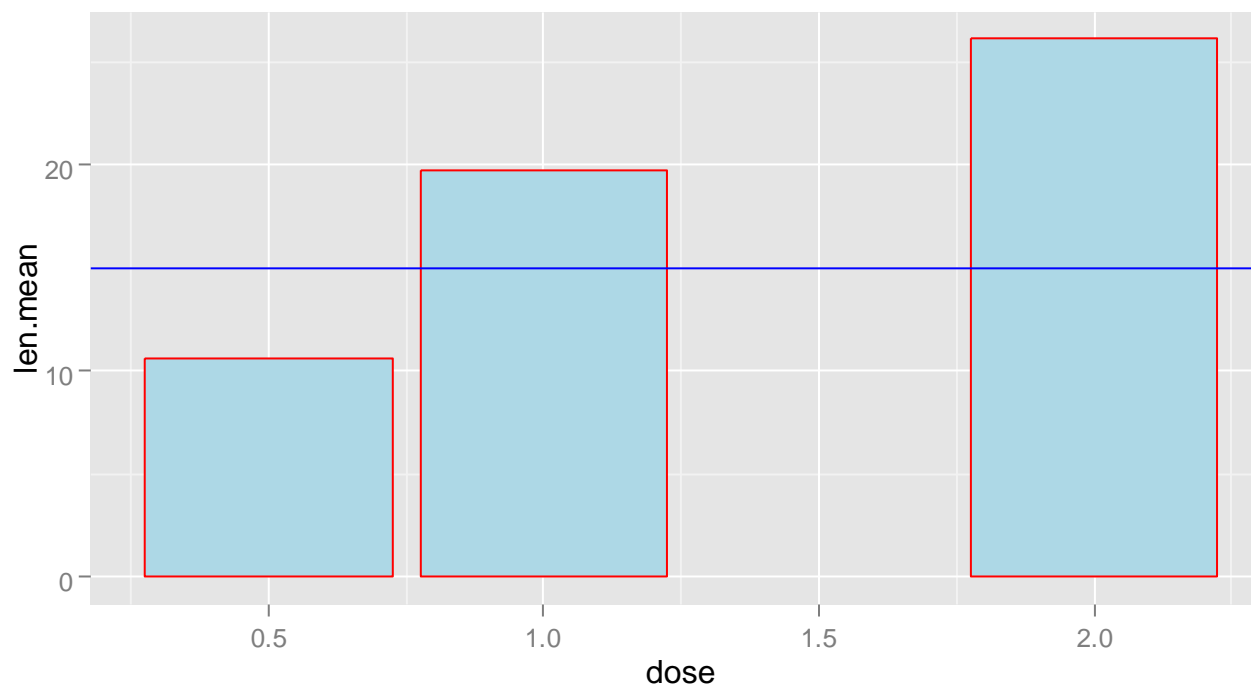




## グラフの作成例：棒グラフ+参照線

```
> base2 <- ggplot(MEAN, aes(x=dose, y=len.mean))  
> base2 + geom_bar(stat="identity", fill="lightblue", color="red") +  
+   geom_hline(yintercept=15, color="blue")
```

- ▶ 各用量の「歯の長さの平均値」に関する棒グラフに参照線を追記
- ▶ 関数 `geom_hline()` や `geom_vline()` で参照線が描ける

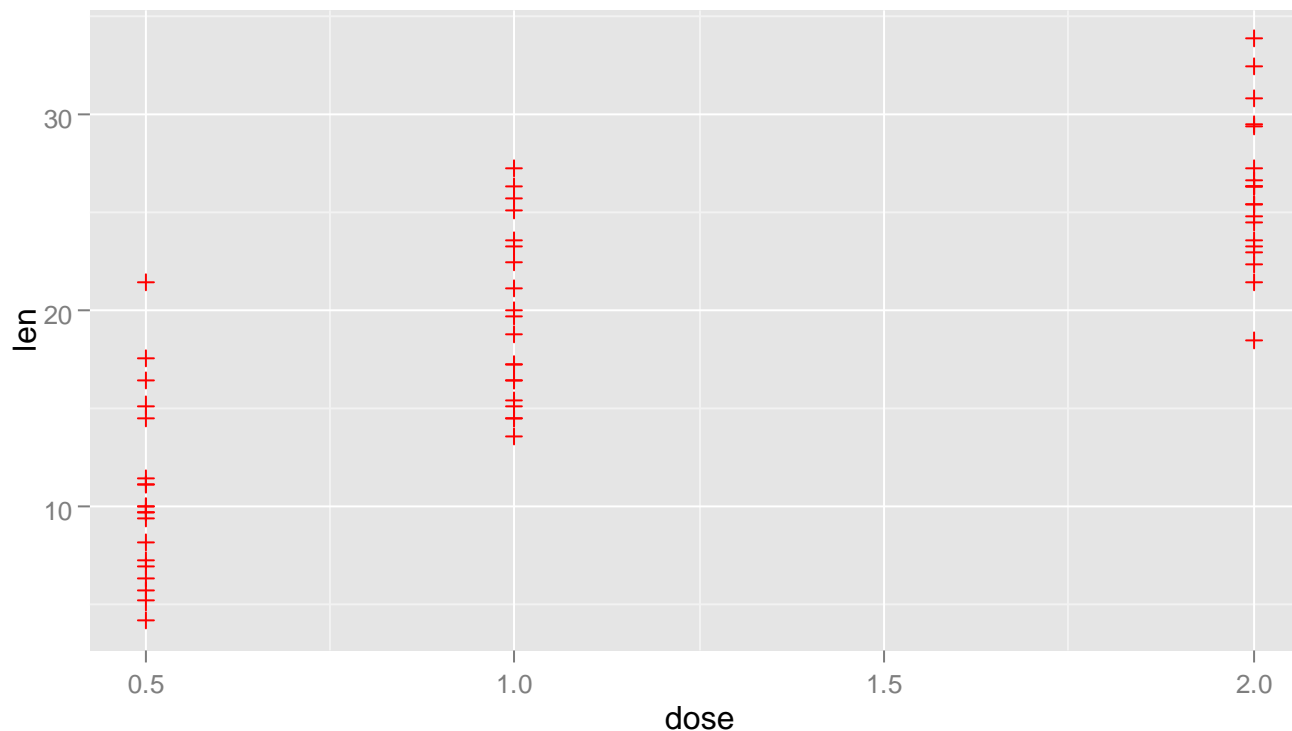




## グラフの作成例：散布図

```
> base <- ggplot(ToothGrowth, aes(x=dose, y=len))  
> base + geom_point(shape=3, size=2, color="red")
```

- ▶ 関数 `geom_point()` で散布図を作成することが出来る
- ▶ 引数 `shape` や `size` や `color` で形や大きさや色を変更できる

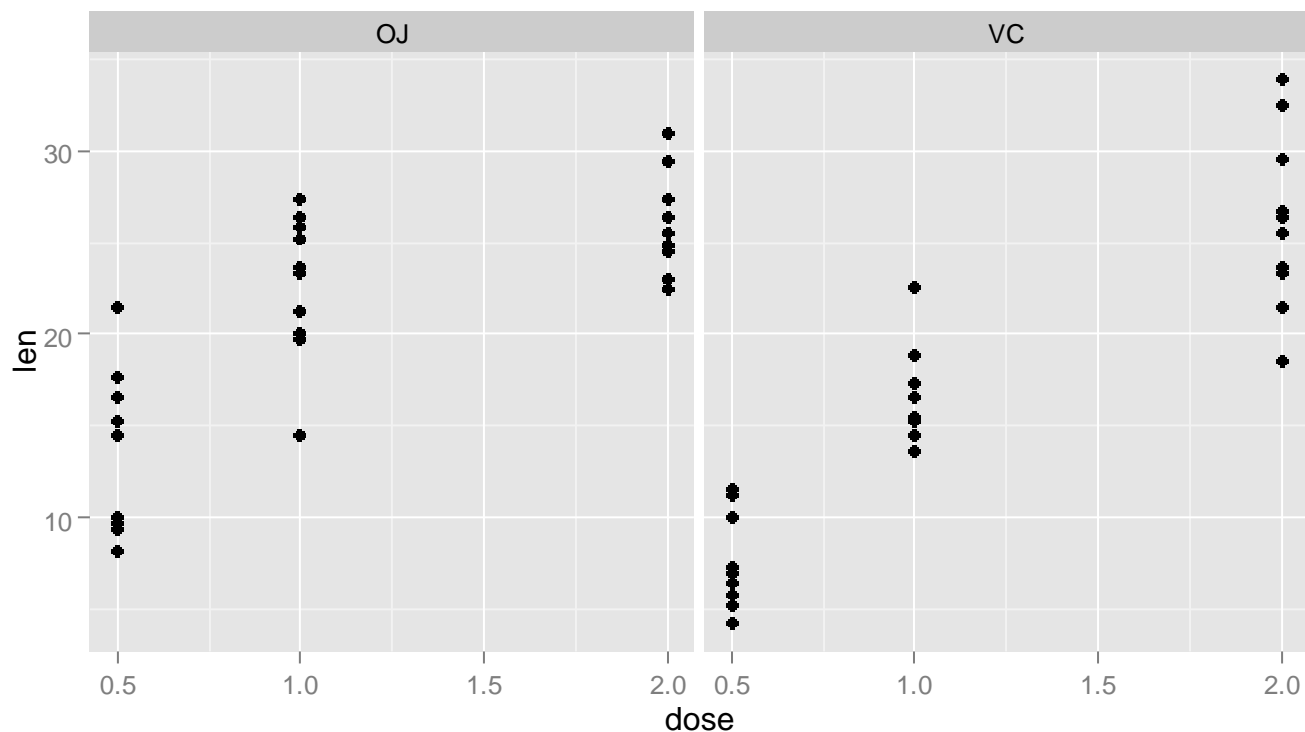




## グラフの作成例：散布図

```
> base <- ggplot(ToothGrowth, aes(x=dose, y=len))  
> base + geom_point() + facet_wrap(~ supp, nrow=1, ncol=2)  
> base + geom_point() + facet_grid(. ~ supp) # 上記と同じ
```

- ▶ 層別したグラフを作成する場合は関数 `facet_wrap()` や `facet_grid()` を用いる



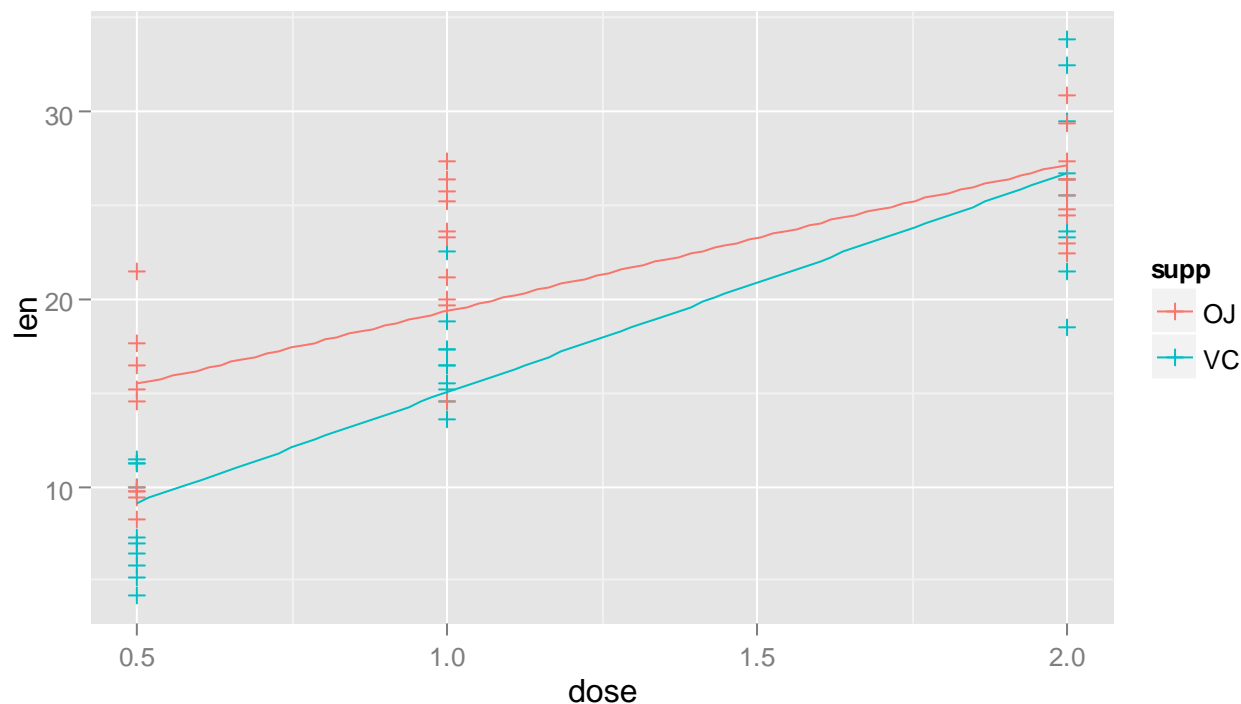




## グラフの作成例：散布図＋回帰直線（**supp** 別）

```
> base <- ggplot(ToothGrowth, aes(x=dose, y=len, color=supp))  
> base + geom_point(shape=3, size=2) +  
+   stat_smooth(method=lm, se=FALSE) # デフォルト：formula=y~x
```

- ▶ 回帰直線を追記する場合は関数 `stat_smooth()` を用いる
- ▶ 関数 `ggplot()` の引数 `color` にグループ (`supp`) を指定する

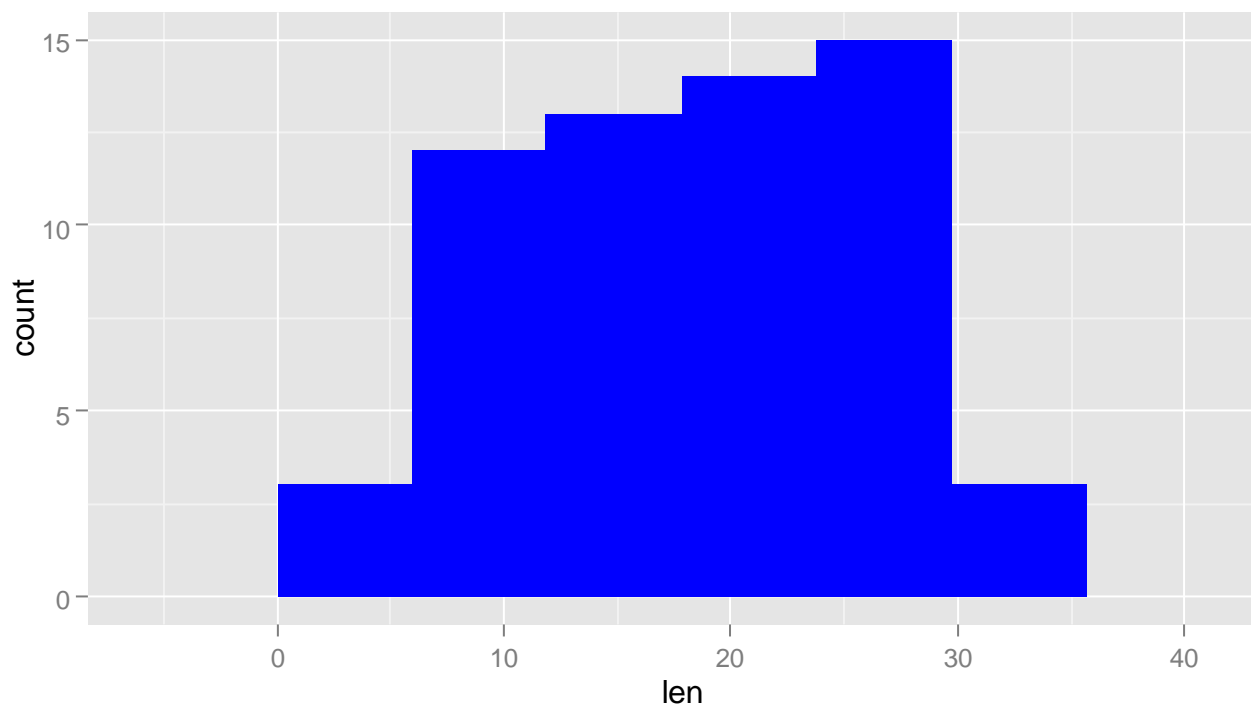




## グラフの作成例：ヒストグラム

```
> nc <- diff(range(ToothGrowth$len)/nclass.FD(ToothGrowth$len))
> base <- ggplot(ToothGrowth, aes(x=len))
> base + geom_histogram(binwidth=nc, fill="blue")
> base + geom_histogram(binwidth=nc, fill="blue", aes(y=..count..))
```

- ▶ ヒストグラムを作成する場合は関数 `geom_histogram()` を用いる
- ▶ 明示的に「y 軸は頻度」と明示する場合は `..count..` を用いる

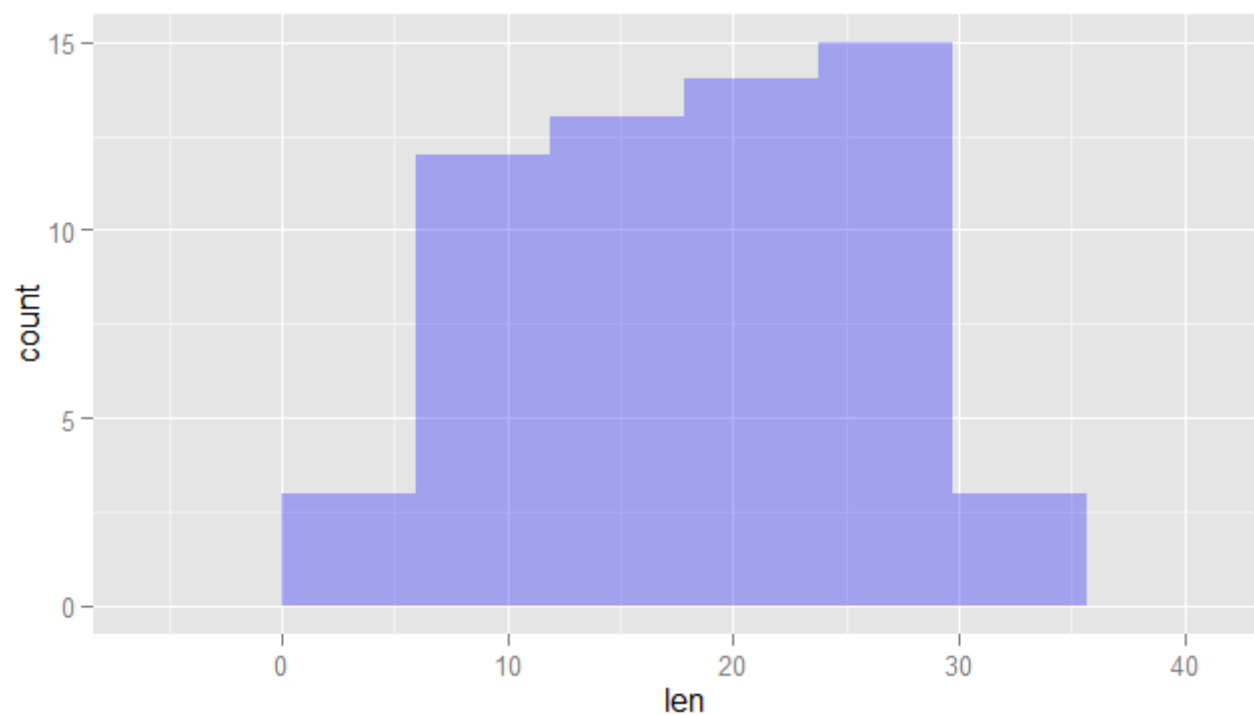




## グラフの作成例：ヒストグラム

```
> nc <- diff(range(ToothGrowth$len)/nclass.FD(ToothGrowth$len))  
> base <- ggplot(ToothGrowth, aes(x=len))  
> base + geom_histogram(binwidth=nc, fill="blue", alpha=0.3)
```

- ▶ ヒストグラムを作成する場合は関数 `geom_histogram()` を用いる
- ▶ 透過にする場合は引数 `alpha` を用いる

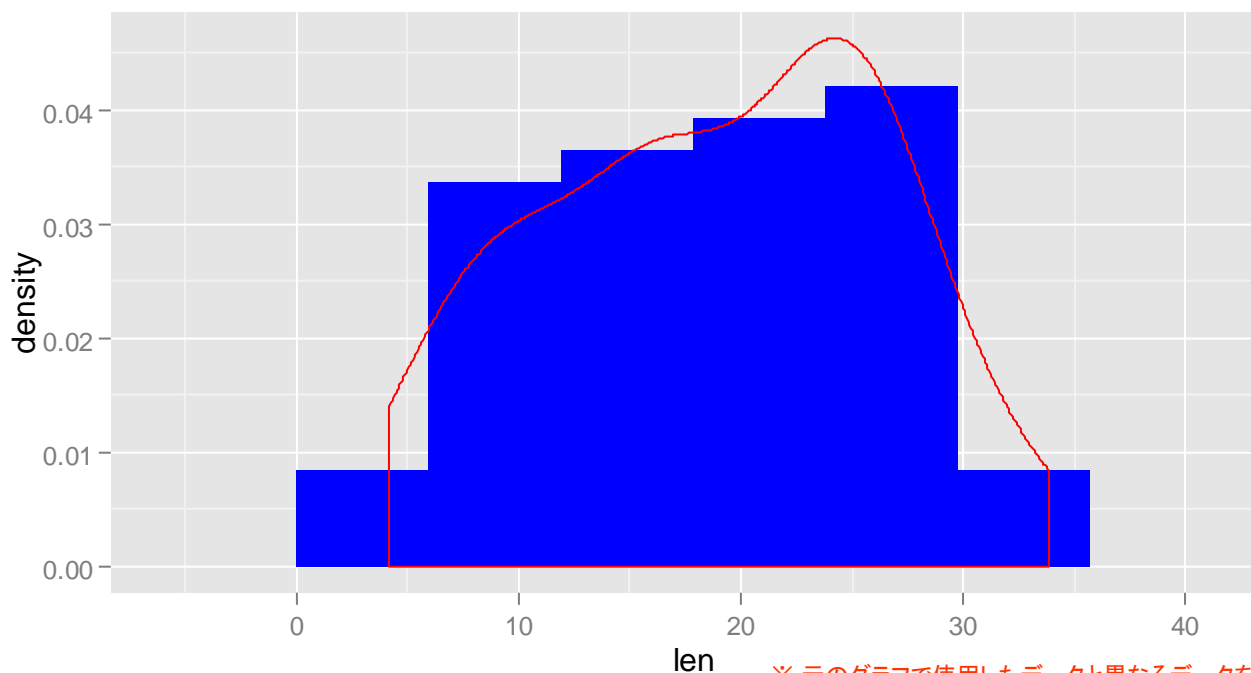




## グラフの作成例：ヒストグラムと密度曲線

```
> nc <- diff(range(ToothGrowth$len)/nclass.FD(ToothGrowth$len))
> base <- ggplot(ToothGrowth, aes(x=len))
> base + geom_histogram(binwidth=nc, fill="blue", aes(y=..density..)) +
+   geom_density(col="red", fill=NA)
```

- ▶ ヒストグラムに密度曲線を上書きする場合は上記のようにする
- ▶ 明示的に「y 軸は密度」と明示するため `..density..` を用いる

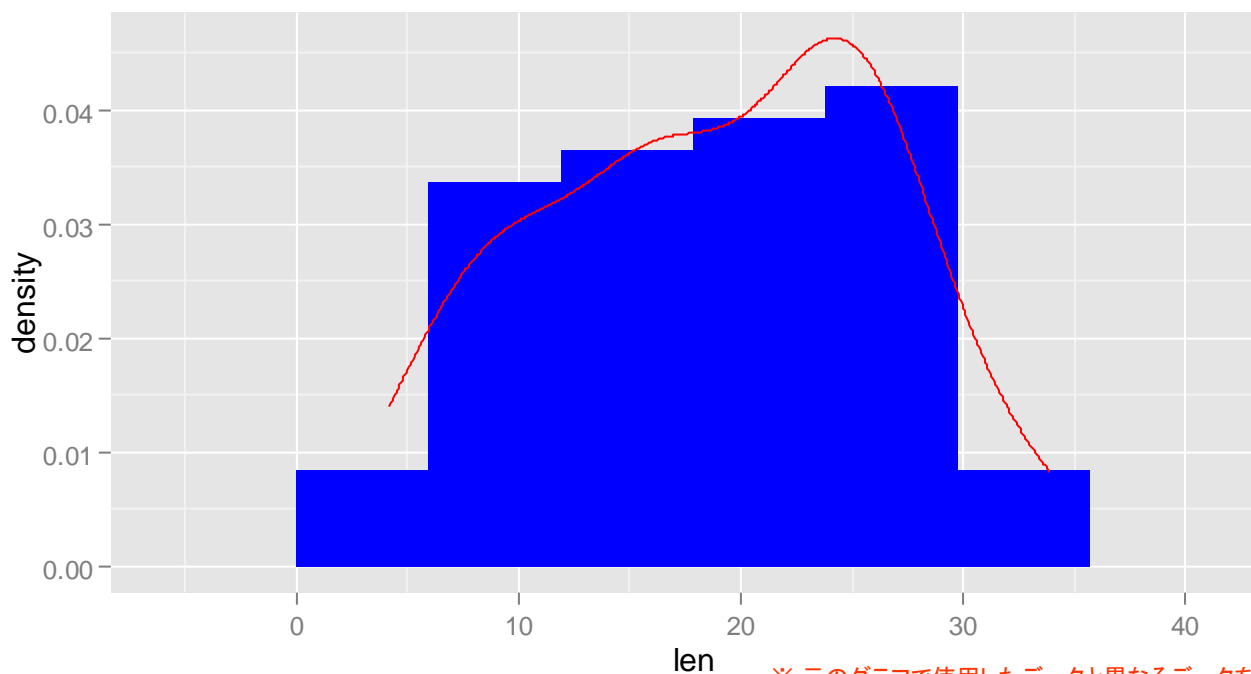




## グラフの作成例：ヒストグラムと密度曲線

```
> nc <- diff(range(ToothGrowth$len)/nclass.FD(ToothGrowth$len))
> base <- ggplot(ToothGrowth, aes(x=len))
> base + geom_histogram(binwidth=nc, fill="blue", aes(y = ..density..)) +
+   geom_line(stat="density", col="red") + expand_limits(y=0)
```

- ▶ ヒストグラムに密度曲線を上書きする場合は上記のようにする
- ▶ 明示的に「y 軸は密度」と明示するため `..density..` を用いる

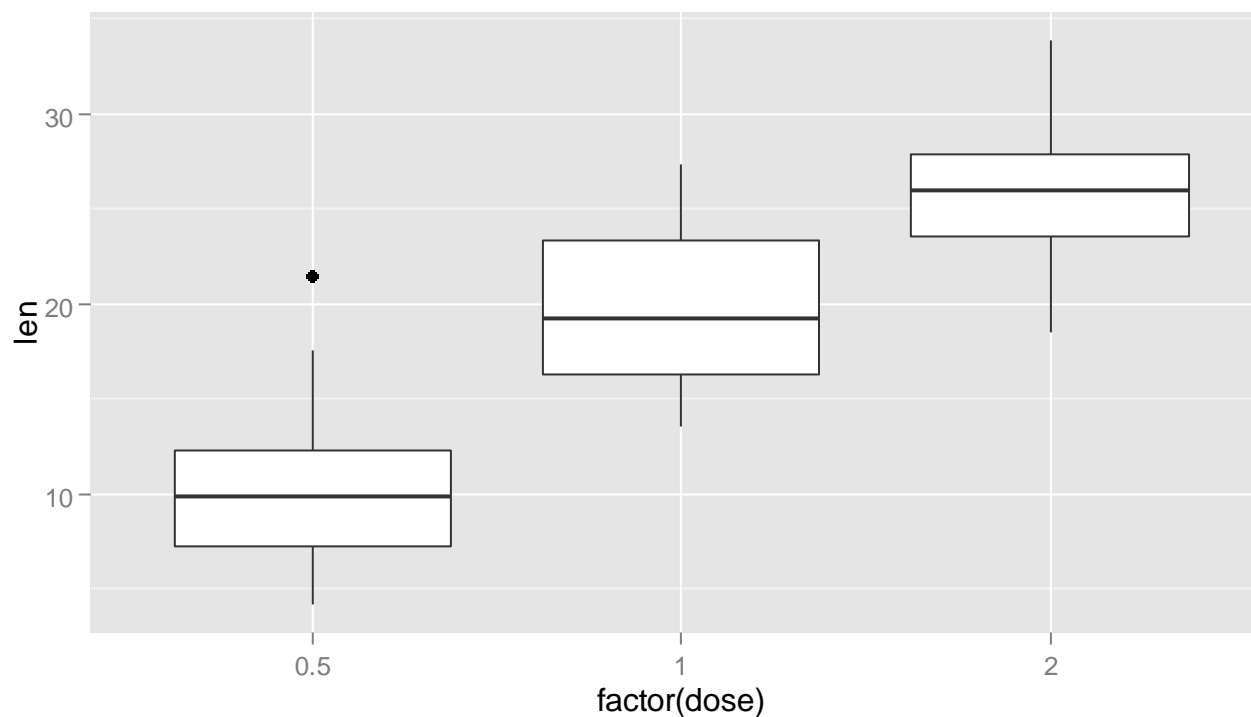




## グラフの作成例：箱ひげ図

```
> base <- ggplot(ToothGrowth, aes(x=factor(dose), y=len))  
> base + geom_boxplot()
```

- ▶ 箱ひげ図を作成する場合は関数 `geom_boxplot()` を用いる

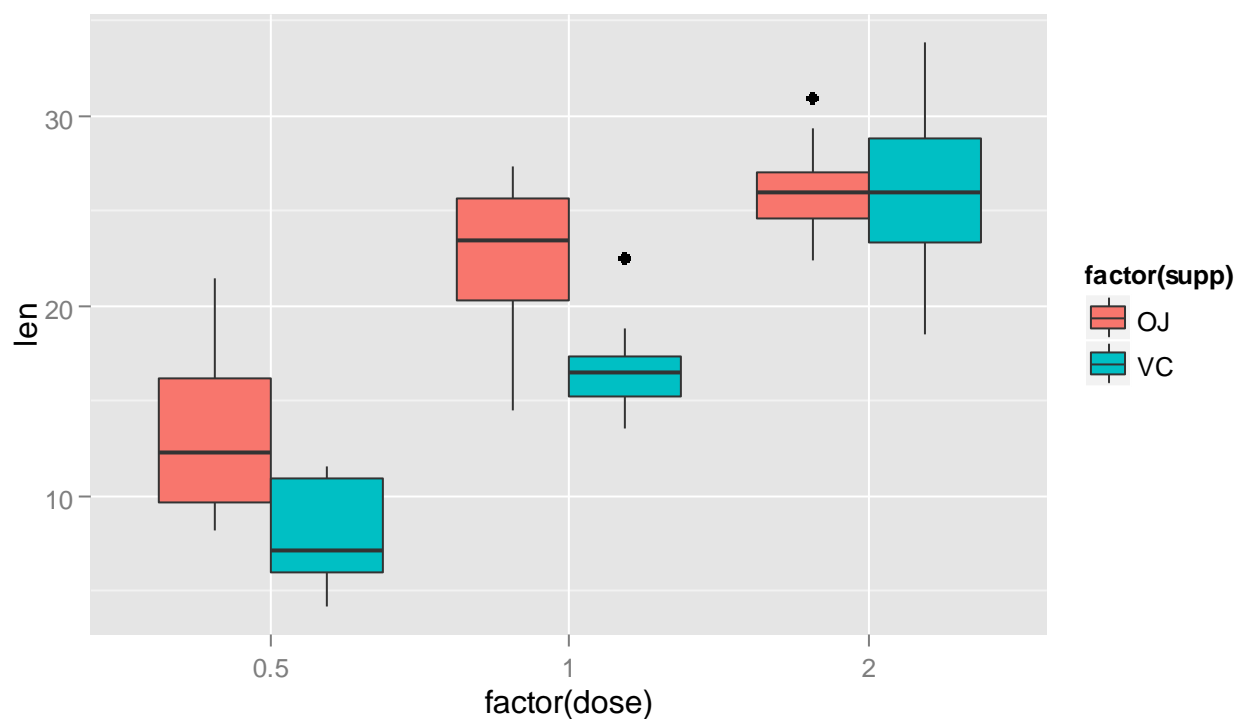




## グラフの作成例：箱ひげ図

```
> base <- ggplot(ToothGrowth, aes(x=factor(dose), y=len))  
> base + geom_boxplot(aes(fill=factor(supp)), outlier.size=2)
```

- ▶ 箱ひげ図を作成する場合は関数 `geom_boxplot()` を用いる
- ▶ 層別を行う場合は関数 `aes()` の引数 `fill` に指定する

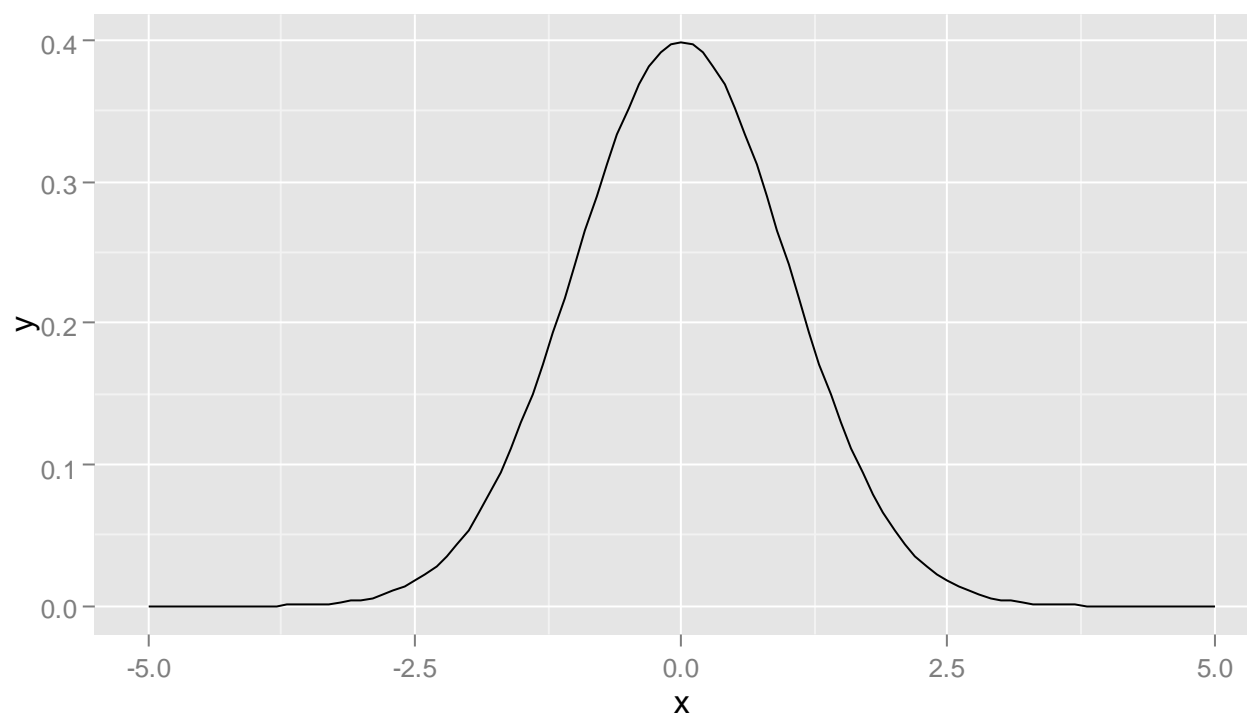




## グラフの作成例：数学関数の描画

```
> base <- ggplot(data.frame(x = c(-5, 5)), aes(x))  
> base + stat_function(fun=dnorm)
```

- ▶ 数学関数を描く場合は関数 `stat_function()` を用いる



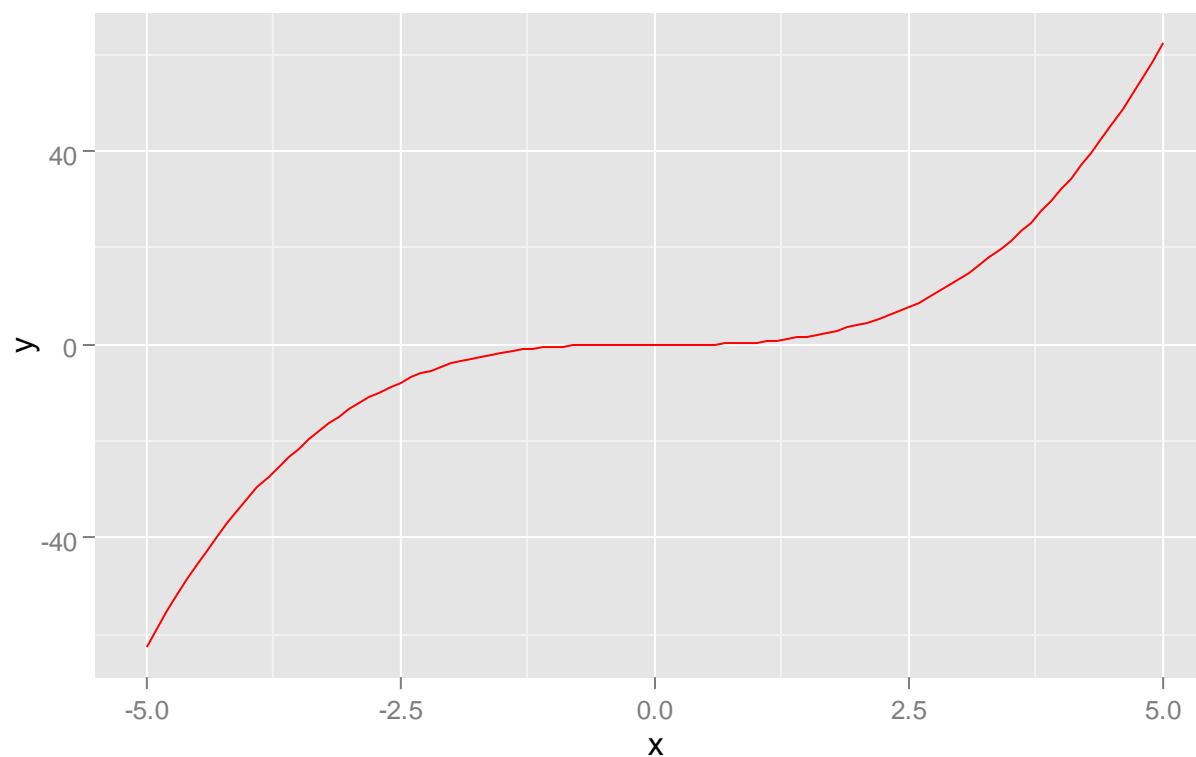




## グラフの作成例：数学関数の描画

```
> f <- function(x) x^3/2  
> base <- ggplot(data.frame(x = c(-5, 5)), aes(x))  
> base + stat_function(fun=f, color="red")
```

- ▶ 数学関数を描く場合は関数 `stat_function()` を用いる

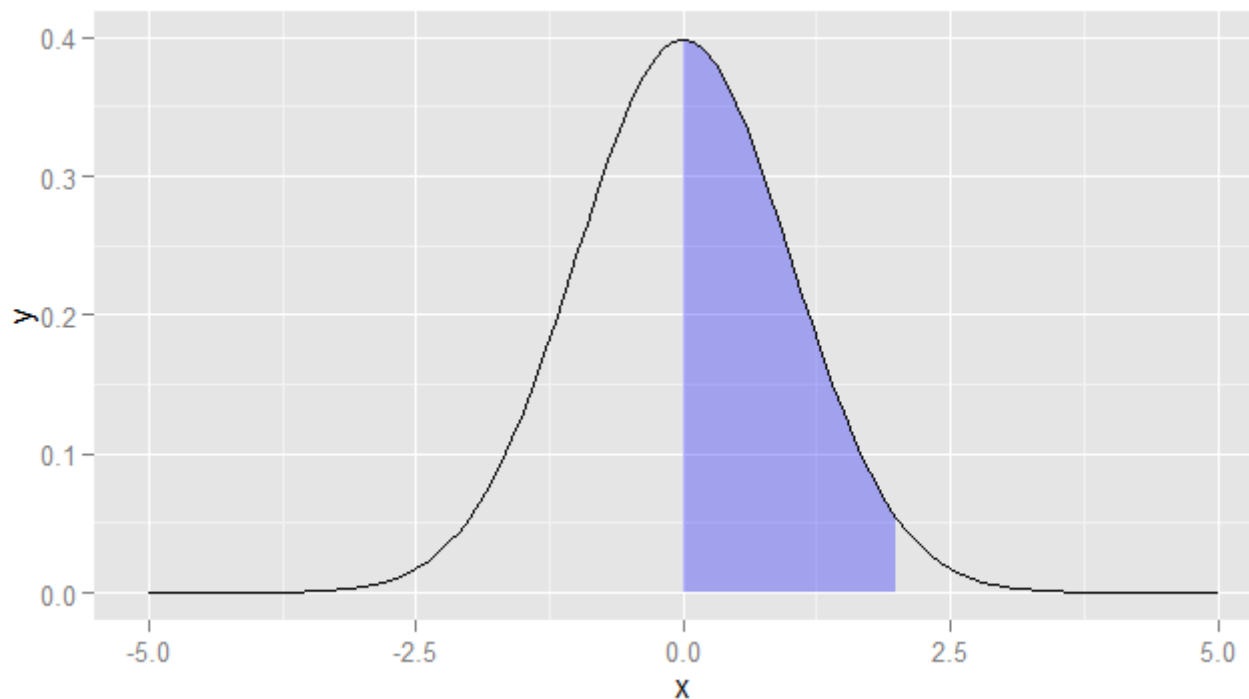




## グラフの作成例：数学関数の描画

```
> xlimit <- function(x) { y <- dnorm(x); y[x<0 | x>2] <- NA; y }  
> base <- ggplot(data.frame(x = c(-5, 5)), aes(x))  
> base + stat_function(fun=xlimit, geom="area", fill="blue", alpha=0.3) +  
+ stat_function(fun=dnorm)
```

- ▶ 数学関数を描く場合は関数 `stat_function()` を用いる
- ▶ 一部の領域を色で塗ることも出来る

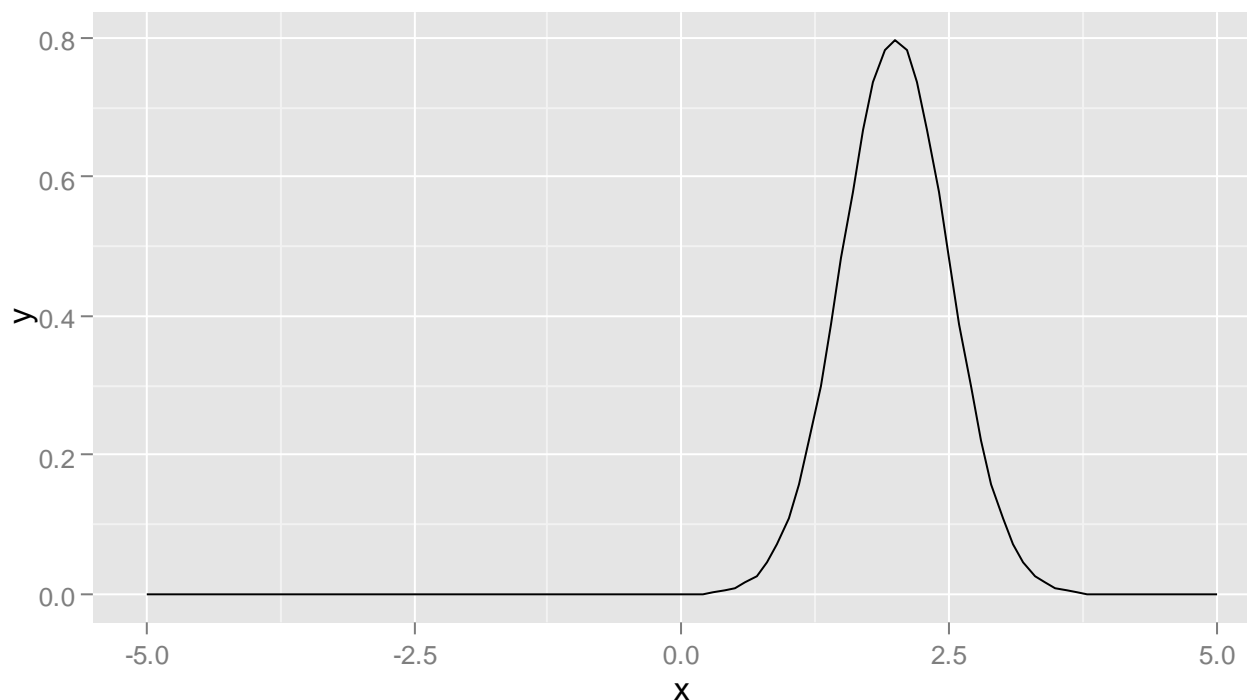




## グラフの作成例：数学関数の描画

```
> base <- ggplot(data.frame(x = c(-5, 5)), aes(x))  
> base + stat_function(fun=dnorm, args=list(mean=2, sd=0.5))
```

- ▶ 数学関数を描く場合は関数 `stat_function()` を用いる
- ▶ `x` 以外の変数の指定は引数 `args` にリストを指定する

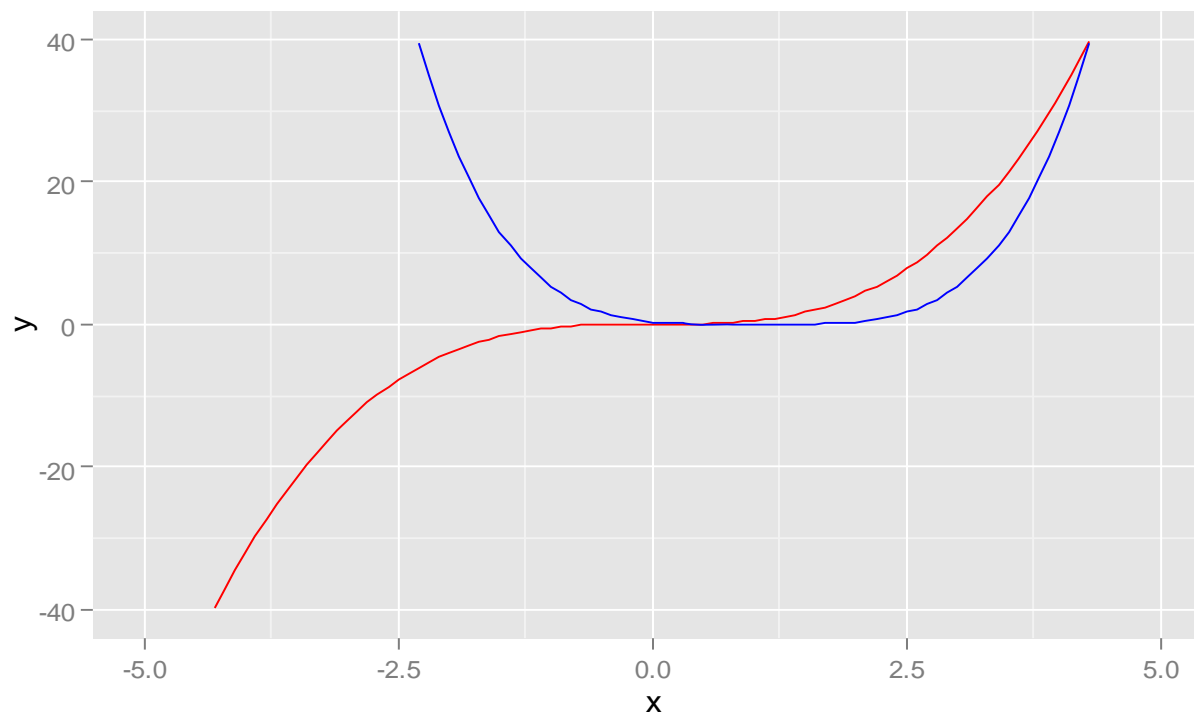




## グラフの作成例：数学関数の描画

```
> f <- function(x) x^3/2 ; g <- function(x,y) (x-y)^4/3  
> base <- ggplot(data.frame(x = c(-5, 5)), aes(x))  
> base + stat_function(fun=f, color="red") +  
+   stat_function(fun=g, color="blue", args=list(y=1)) + ylim(c(-40,40))
```

- ▶ 数学関数を描く場合は関数 `stat_function()` を用いる
- ▶ `x` 以外の変数の指定は引数 `args` にリストを指定する





# 頻度集計

len >= 15 のデータを対象に, Dose と supp の頻度集計と割合を算出

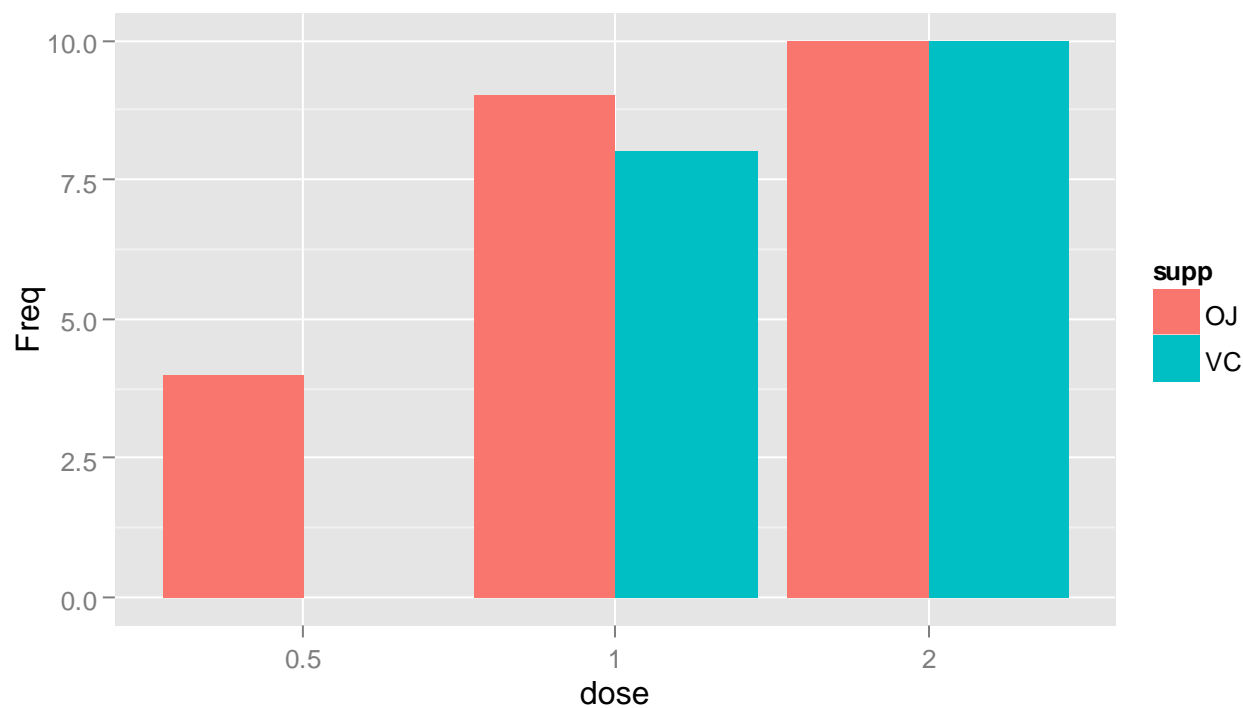
```
> TMP1 <- xtabs( ~ dose + supp, data=subset(ToothGrowth,len>=15))
> TMP2 <- prop.table(FREQ, 1)
> ( TAB1 <- reshape(as.data.frame(TMP1), v.names="Freq", idvar="supp",
+   varying=list(3), direction="long") )[, -3]
  dose supp Freq TREAT
1  0.5  OJ    4     1
2   1   OJ    9     2
3   2   OJ   10     3
4  0.5  VC    0     4
5   1   VC    8     5
6   2   VC   10     6

> ( TAB2 <- reshape(as.data.frame(TMP2), v.names="Freq", idvar="supp",
+   varying=list(3), direction="long") )[, -3]
  dose supp      Freq TREAT
1  0.5  OJ 1.0000000     1
2   1   OJ 0.5294118     2
3   2   OJ 0.5000000     3
4  0.5  VC 0.0000000     4
5   1   VC 0.4705882     5
6   2   VC 0.5000000     6
```



## グラフの作成例：頻度に関する棒グラフ

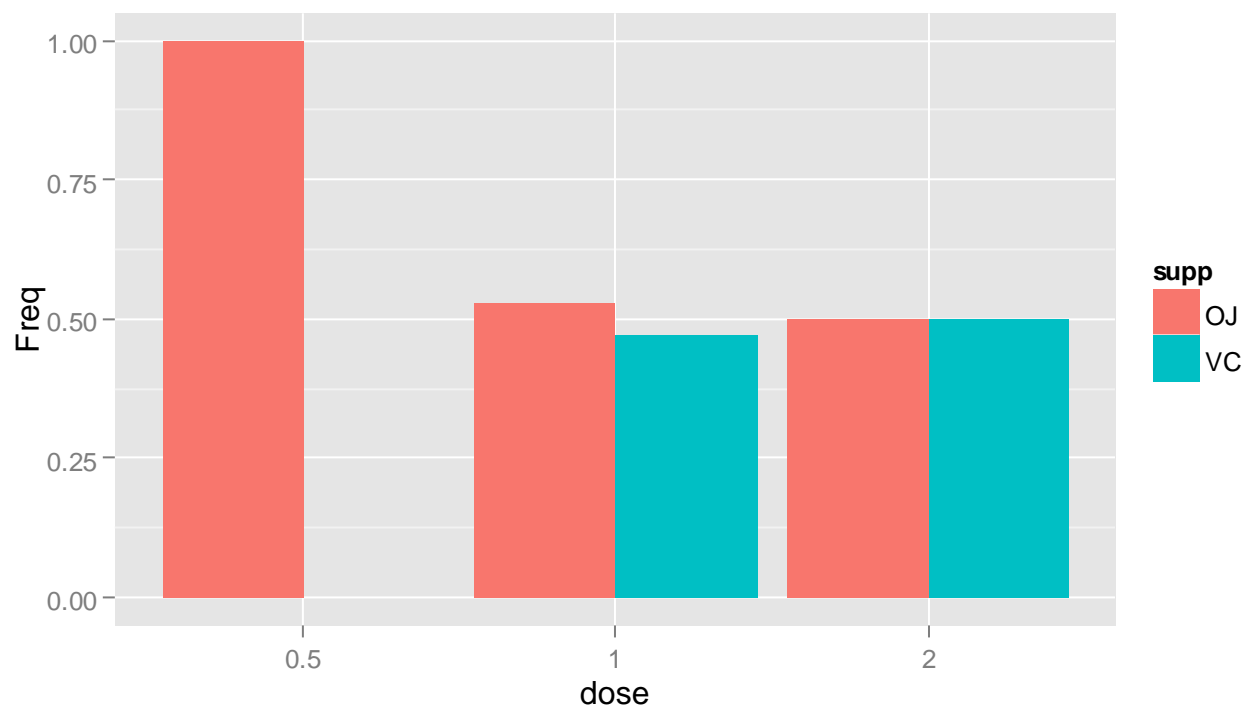
```
> ggplot(TAB1, aes(x=dose, y=Freq, fill=supp)) +  
+   geom_bar(position="dodge", stat="identity")
```





## グラフの作成例：割合に関する棒グラフ

```
> ggplot(TAB2, aes(x=dose, y=Freq, fill=supp)) +  
+   geom_bar(position="dodge", stat="identity")
```





## 本日のメニュー

---

1. イントロ
2. ggplot2 事始
3. グラフの作成例
4. **グラフのカスタマイズ例**

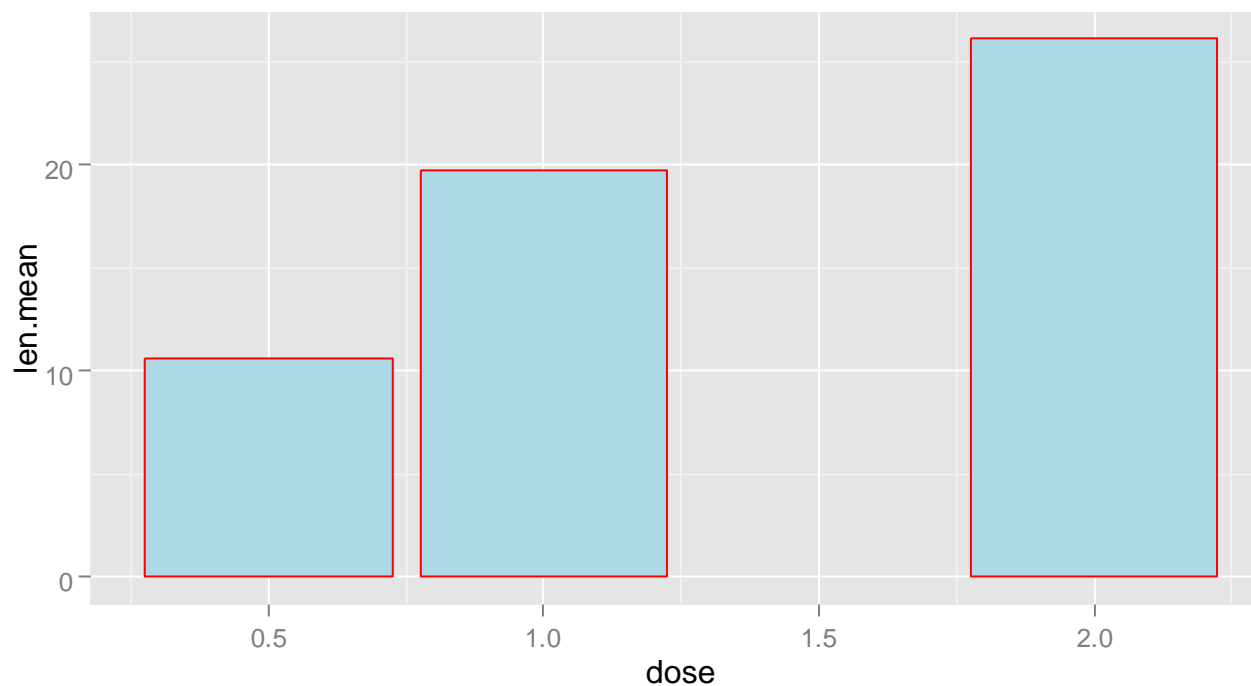




## グラフの作成例：棒グラフ

```
> MEAN <- summaryBy(len ~ dose, data=ToothGrowth, FUN=c(length,mean,sd))  
> base3 <- ggplot(MEAN, aes(x=dose, y=len.mean)) +  
+   geom_bar(stat="identity", fill="lightblue", color="red")  
> base3
```

- ▶ 各用量の「歯の長さの平均値」に関する棒グラフ

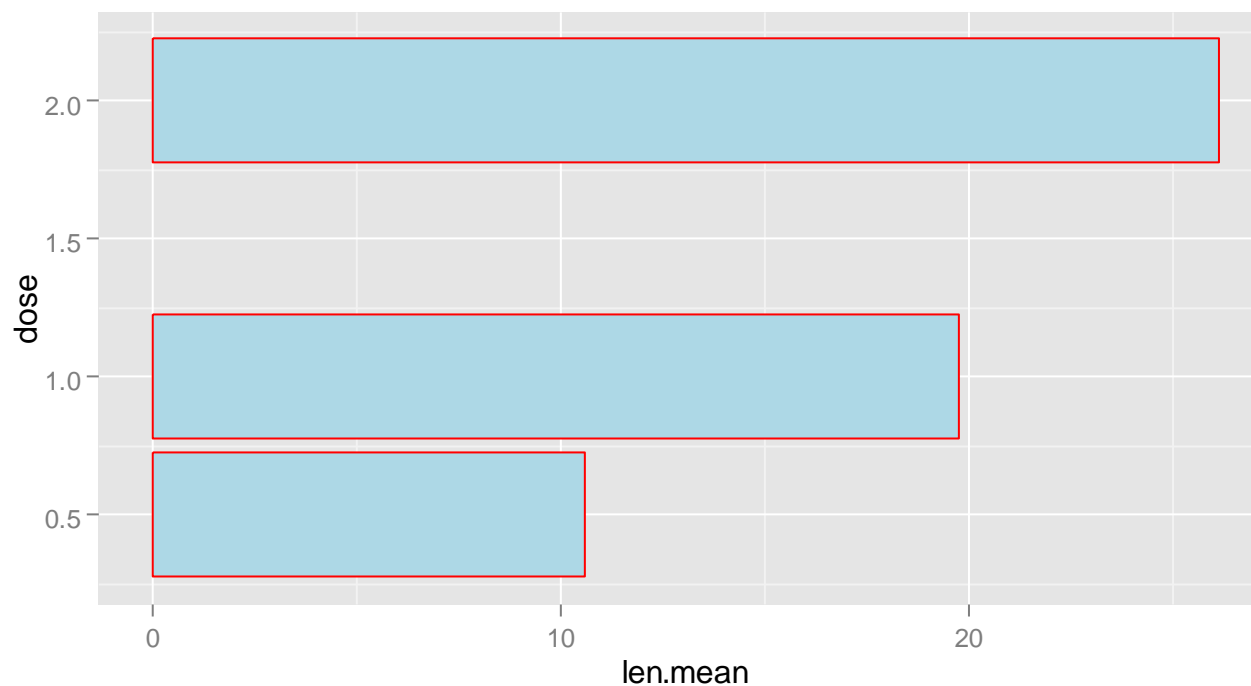




## グラフのカスタマイズ例：棒グラフの転置

```
> base3 + coord_flip()
```

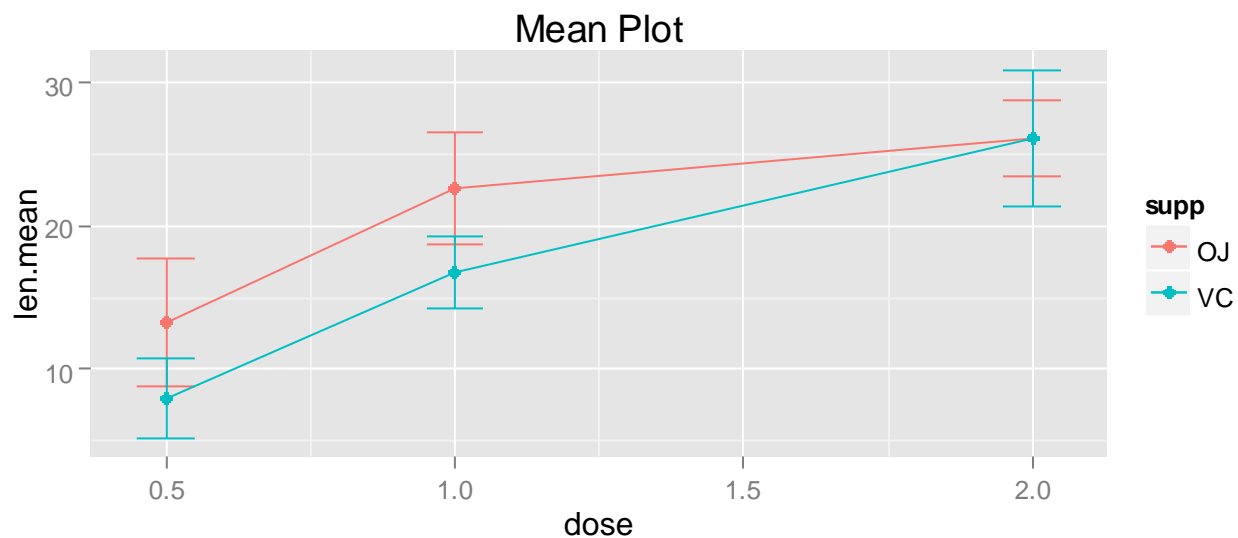
- ▶ 各用量の「歯の長さの平均値」に関する棒グラフを
- ▶ 関数 `coord_flip()` でグラフを転置する





## グラフの作成例：平均値の推移図

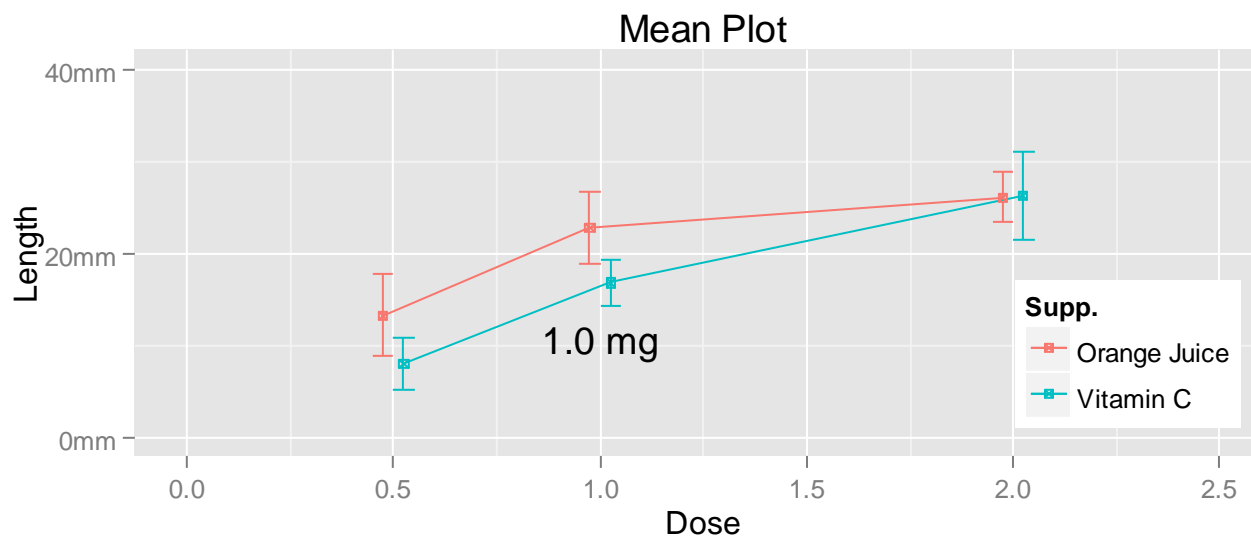
```
> ( MEAN <- summaryBy(len ~ dose + supp, data=ToothGrowth, FUN=c(length,mean,sd)) )
  dose supp len.length len.mean  len.sd
1  0.5   OJ         10    13.23 4.459709
2  0.5   VC         10     7.98 2.746634
3  1.0   OJ         10    22.70 3.910953
4  1.0   VC         10    16.77 2.515309
5  2.0   OJ         10    26.06 2.655058
6  2.0   VC         10    26.14 4.797731
> ggplot(MEAN, aes(x=dose, y=len.mean, color=supp)) +
+   geom_errorbar(aes(ymin=len.mean-len.sd, ymax=len.mean+len.sd), width=.1) +
+   geom_line() + geom_point() + ggtitle("Mean Plot")
```





## グラフのカスタマイズ例：平均値の推移図

```
> pd <- position_dodge(0.1) # 曲線同士が重ならないように少しズラす
> ggplot(MEAN, aes(x=dose, y=len.mean, color=supp)) +
+   geom_errorbar(aes(ymin=len.mean-len.sd, ymax=len.mean+len.sd), width=.1, position=pd) +
+   geom_line(position=pd) + geom_point(position=pd, size=1.5, shape=7) +
+   scale_color_manual(values=c("OJ"="red", "VC"="blue")) +
+   scale_color_discrete(limits=c("OJ", "VC"), labels=c("OJ"="Orange Juice", "VC"="Vitamin C")) +
+   theme(legend.position=c(0.75,0.0), legend.justification=c(0,0)) +
+   xlab("Dose") + ylab("Length") + ggtitle("Mean Plot") + labs(color="Supp.") +
+   xlim(c(0,2.5)) +
+   scale_y_continuous(limits=c(0,40), breaks=seq(0,40,20), labels=c("0mm", "20mm", "40mm")) +
+   annotate("text", x=1, y=11, label="1.0 mg")
```





## グラフのカスタマイズ用関数

---

- ▶ グラフの色
  - ▶ `scale_color_manual(values=c("OJ"="red","VC"="blue"))`
- ▶ 凡例の中身（`color` の部分は適宜修正）
  - ▶ `scale_color_discrete(limits=c("OJ","VC"), labels=c("OJ"="Orange Juice","VC"="Vitamin C"))` :  
凡例で出力するカテゴリ（`limits`）とそのラベル（`labels`）を指定
- ▶ 凡例の位置
  - ▶ `theme(legend.position=c(0.75,0), legend.justification=c(1,0))` : 特定の位置
  - ▶ `theme(legend.position = "top" or "right" or "bottom" or "left" or "none")` :  
ざっくり位置を指定する場合



## グラフのカスタマイズ関数

---

- ▶ ラベル関係
  - ▶ `xlab("Dose")` : x 軸のラベルを指定
  - ▶ `ylab("Length")` : y 軸のラベルを指定
  - ▶ `ggtitle("Mean Plot")` : グラフのタイトルを指定
  - ▶ `labs(color="Supp.")` : 凡例のタイトルを指定
- ▶ 座標の範囲
  - ▶ `xlim(c(0,2.5))` : x 軸の範囲
  - ▶ `ylim(c(0,40))` : y 軸の範囲
  - ▶ `scale_x_continuous(limits=c(0,2.5), breaks=seq(0,2.5,0.5))` : x 軸の範囲
  - ▶ `scale_y_continuous(limits=c(0,40), breaks=seq(0,40,10), labels=c("0mm","20mm","40mm"))` : y 軸の範囲 + 刻み幅に関する情報



## グラフのカスタマイズ用関数

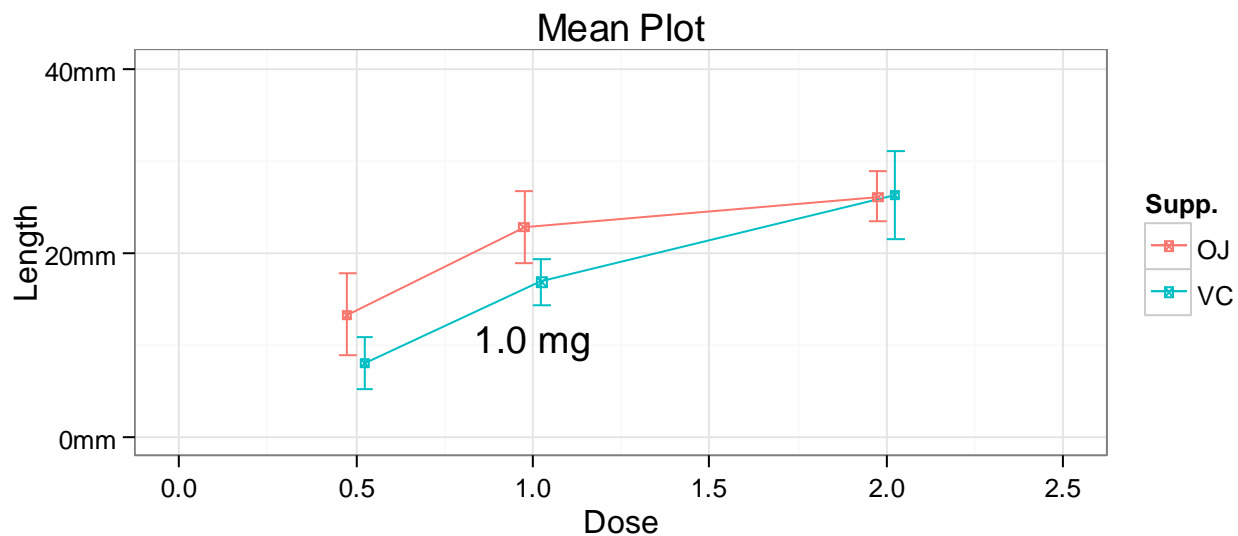
---

- ▶ 文字の追記
  - ▶ `annotate("text", x=1, y=11, label="1.0 mg")`
- ▶ テーマ → 見栄えをパッと変えたいとき
  - ▶ `theme_gray()`
  - ▶ `theme_bw()`
  - ▶ `theme_linedraw()`
  - ▶ `theme_light()`
  - ▶ `theme_minimal()`
  - ▶ `theme_classic()`



## テーマの使用例：theme\_bw()

```
> pd <- position_dodge(0.1) # 曲線同士が重ならないように少しズラす
> ggplot(MEAN, aes(x=dose, y=len.mean, color=supp)) +
+   geom_errorbar(aes(ymin=len.mean-len.sd, ymax=len.mean+len.sd), width=.1, position=pd) +
+   geom_line(position=pd) + geom_point(position=pd, size=1.5, shape=7) +
+   xlab("Dose") + ylab("Length") + ggtitle("Mean Plot") + labs(color="Supp.") +
+   xlim(c(0,2.5)) +
+   scale_y_continuous(limits=c(0,40), breaks=seq(0,40,20), labels=c("0mm", "20mm", "40mm")) +
+   annotate("text", x=1, y=11, label="1.0 mg") +
+   theme_bw()
```







## 関数 `element_xx()` と関数 `theme()`

---

▶ さらなるカスタマイズを行う場合は以下の関数を使用する

- ▶ `element_blank()`
- ▶ `element_line()`
- ▶ `element_rect()`
- ▶ `element_text()`
- ▶ `element_limits()`

▶ `theme()`

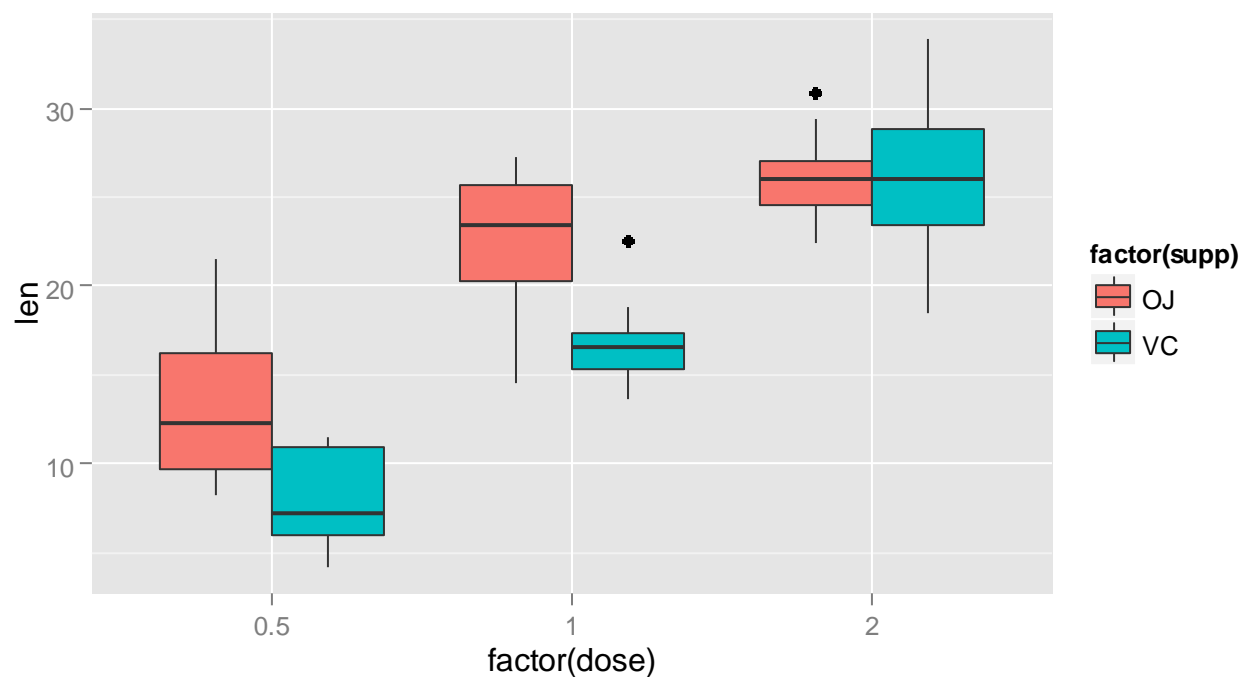
<http://docs.ggplot2.org/current/theme.html>



## グラフの作成例：箱ひげ図

```
> base <- ggplot(ToothGrowth, aes(x=factor(dose), y=len)) +  
+   geom_boxplot(aes(fill=factor(supp)), outlier.size=2)  
> base
```

- ▶ 箱ひげ図を作成する場合は関数 `geom_boxplot()` を用いる
- ▶ 層別を行う場合は関数 `aes()` の引数 `fill` に指定する

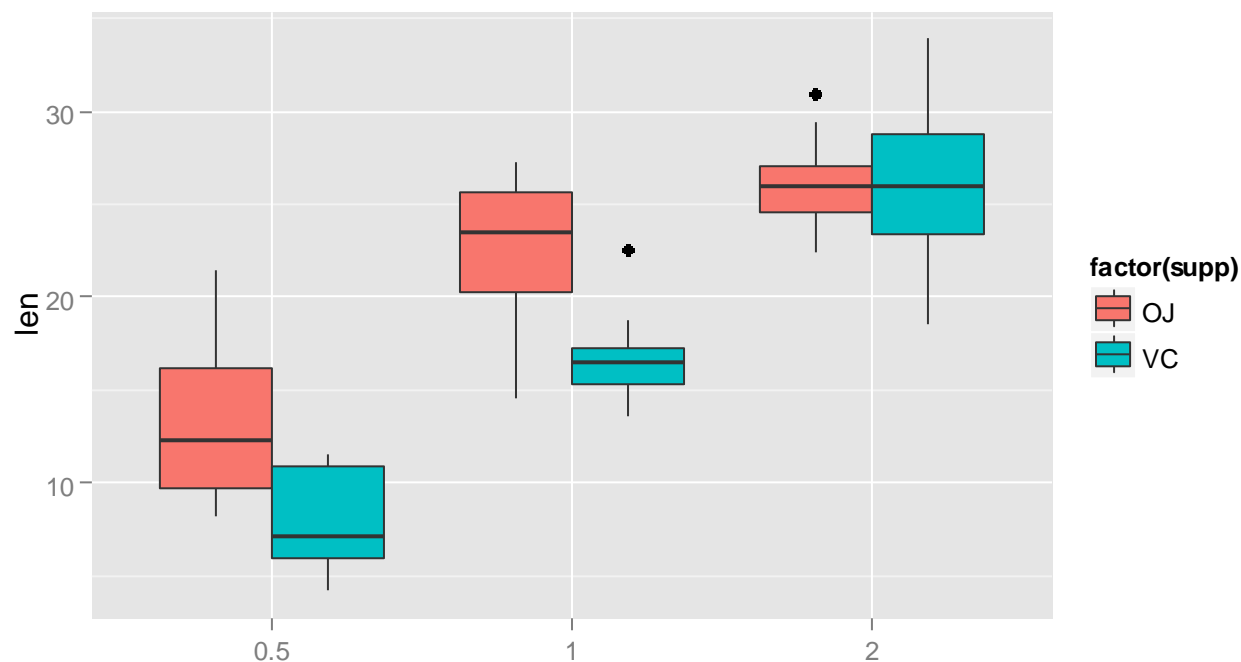




## グラフのカスタマイズ例：箱ひげ図

```
> base + theme(axis.title.x=element_blank())
```

- ▶ x 軸のラベルを消す場合は関数 `theme()` の引数 `axis.title.x` に関数 `element_blank()` を指定する

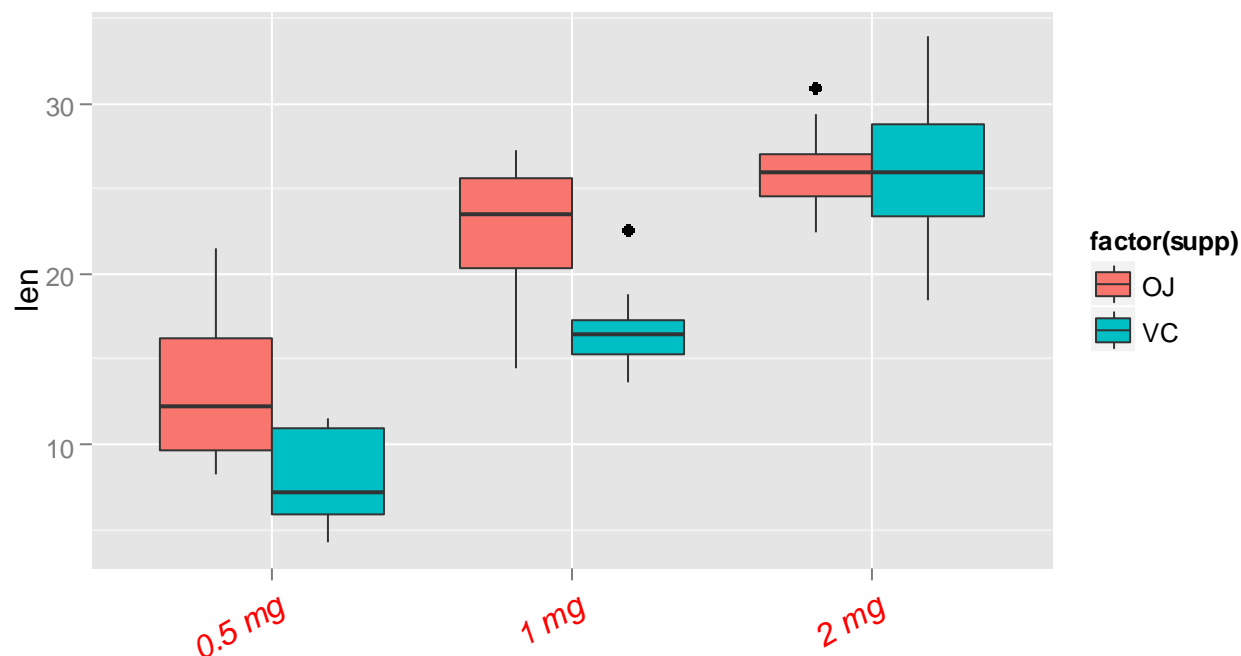




## グラフのカスタマイズ例：箱ひげ図

```
> base + scale_x_discrete(labels=c("0.5 mg", "1 mg", "2 mg")) +  
+   theme(axis.title.x=element_blank(),  
+         axis.text.x=element_text(angle=30, hjust=1, vjust=1,  
+                                   face="italic", colour="red", size=12))
```

- ▶ 関数 `theme` の引数 `axis.text.x` にいろいろ指定することで、  
x 軸のメモリのラベルの書式を変更することが出来る





## 本日のメニュー

---

1. イントロ
2. ggplot2 事始
3. グラフの作成例
4. グラフのカスタマイズ例



## 参考文献

---

- ▶ Wickham, Hadley (2009)  
「ggplot2 (Springer)」
  - ▶ 【邦訳】石田 基広 他訳 (2012)  
「グラフィックスのための R プログラミング (Springer Japan)」
- ▶ Winston Chang (2013)  
「R Graphics Cookbook (Oreilly & Associates Inc.)」
  - ▶ 【邦訳】石井 弓美子 他訳 (2013)  
「R グラフィックスクックブック (Oreilly japan)」
- ▶ 上記で分からない場合はすぐに Google で検索  
( ggplot2 のバージョンアップが結構行われているため)

# R でグラフ作成

終