R による探索的データ分析入門

発電基盤開発課高津一誠

2018 年 10 月 5 日

データ処理は探索的プロセス

実験データ分析は、探索的なプロセスです。1 回データ処理して終了することはあまりなく、多くの場合は仮説 → 検証を繰り返して適切な結果を得ることができます。ここでは R の可視化処理を使って、その簡単な例を示します。

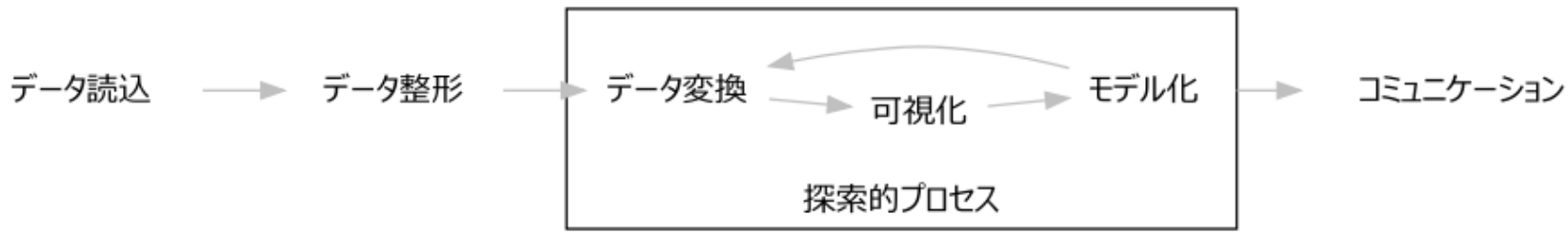

Figure 1: 探索的プロセス

実験データを読み込む

ここでは読み込みの手順は省略し、R に組み込みのテスト用データを使うことにします。使用するデータは生態学の計測データで、アヤメの花弁（petal）とガク（sepal）の長さと幅を、3つの種について 50 個体ずつ計測したデータです。

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##          <dbl>       <dbl>        <dbl>       <dbl> <fct>
## 1          5.1         3.5          1.4         0.2 setosa
## 2          4.9         3            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           3.6          1.4         0.2 setosa
## # ... with 145 more rows
```

分析しやすいよう整形する

元々のデータでは部位ごとの計測項目が変数（列）になっていましたが、部位を変数にした方が扱いやすいと思います。そこで、以下のように整形します。

```
iris_long <- iris %>%
  rownames_to_column("id") %>%
  mutate(id = as.integer(id)) %>%
  gather(key, value, matches("Length|Width")) %>%
  separate(key, into = c("Part", "amount")) %>%
  spread(amount, value)
```

```
## # A tibble: 300 x 5
##       id Species Part  Length Width
##    <int> <fct>   <chr>  <dbl> <dbl>
## 1      1 setosa  Petal    1.4   0.2
## 2      1 setosa  Sepal    5.1   3.5
## 3      2 setosa  Petal    1.4   0.2
## 4      2 setosa  Sepal    4.9   3
```

可視化する

それでは、データを確認するために可視化してみましょう。長さと幅の関係を種ごと部位ごとに確認してみることにします。

```
iris_long %>%
    ggplot(aes(x = Width, y = Length)) +
    geom_point(aes(color = Part)) +
    stat_smooth(method = "lm", color = "gray 40") +
    facet_grid(Part ~ Species) +
    labs(x = "幅(cm)", y = "長さ(cm)", color = "部位")
```

可視化する

比較的よい相関があるようですが、よく見てみると、どの種でもガクの方が大きいようです。

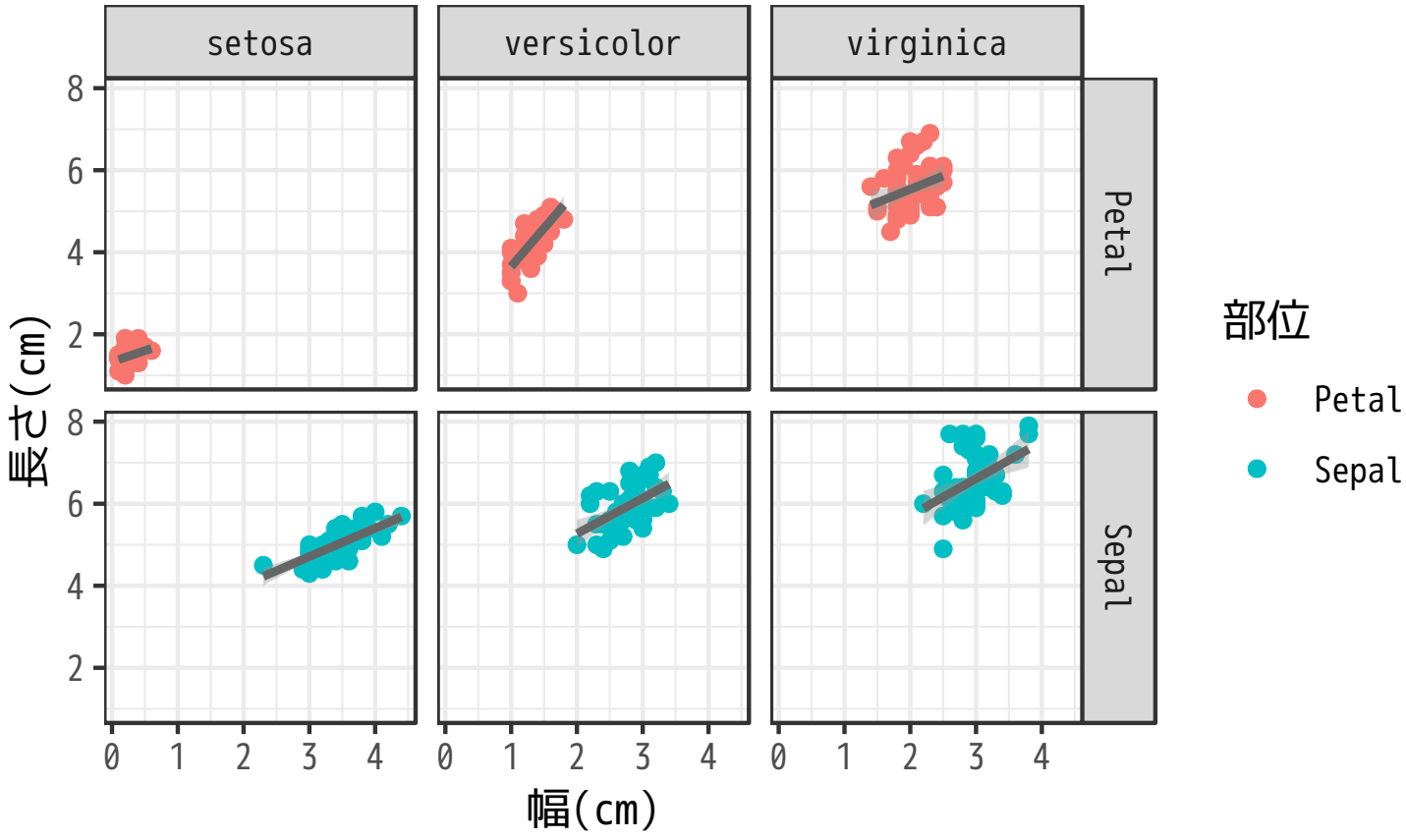

Figure 2: 最初の可視化

仮説その１

もしかすると、長さと幅の関係は花弁とガクで共通と考えたほうがいいのかもしれません。確認してみましょう。

```
iris_long %>%
    ggplot(aes(x = Width, y = Length)) +
    geom_point(aes(color = Part)) +
    stat_smooth(method = "lm", color = "gray 40") +
    facet_wrap(~ Species) +
    labs(x = "Width(cm)", y = "Length(cm)")
```

仮説その１

よい相関があるので、仮説は妥当だったようです。更に見てみると、近似直線の傾きがどれも似ているようです。

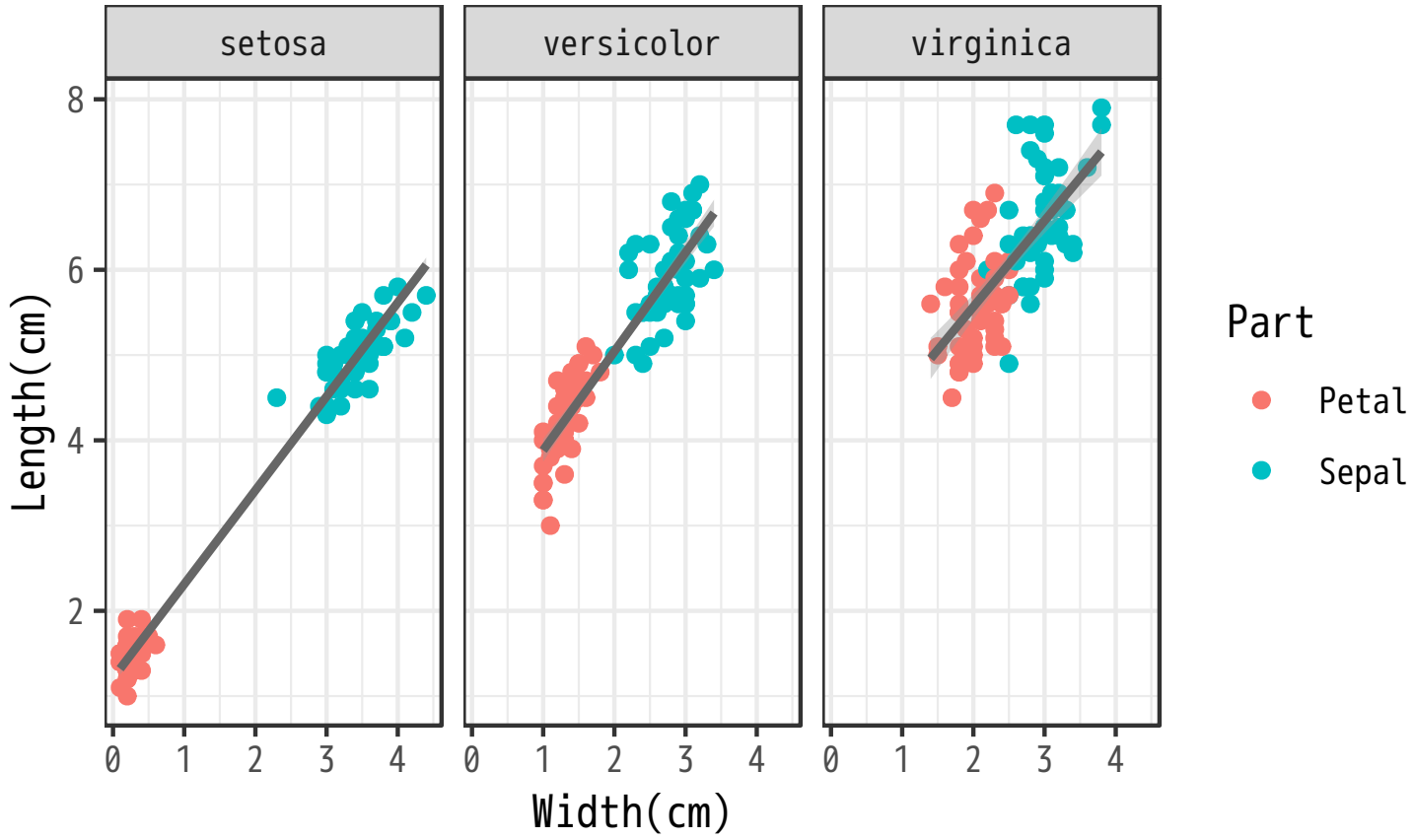

Figure 3: 長さと幅の関係

仮説その２

長さと幅の傾きは種が異なっても共通かもしれません。確認してみましょう。

```
iris_long %>%
  ggplot(aes(x = Width, y = Length, color = Species)) +
    geom_point(aes(shape = Part), size = 3) +
    stat_smooth(method = "lm") +
    labs(x = "Width(cm)", y = "Length(cm)")
```

仮説その２

95% 信頼区間（グレーの領域）を考慮すると、同じ傾きである可能性は高そうです。

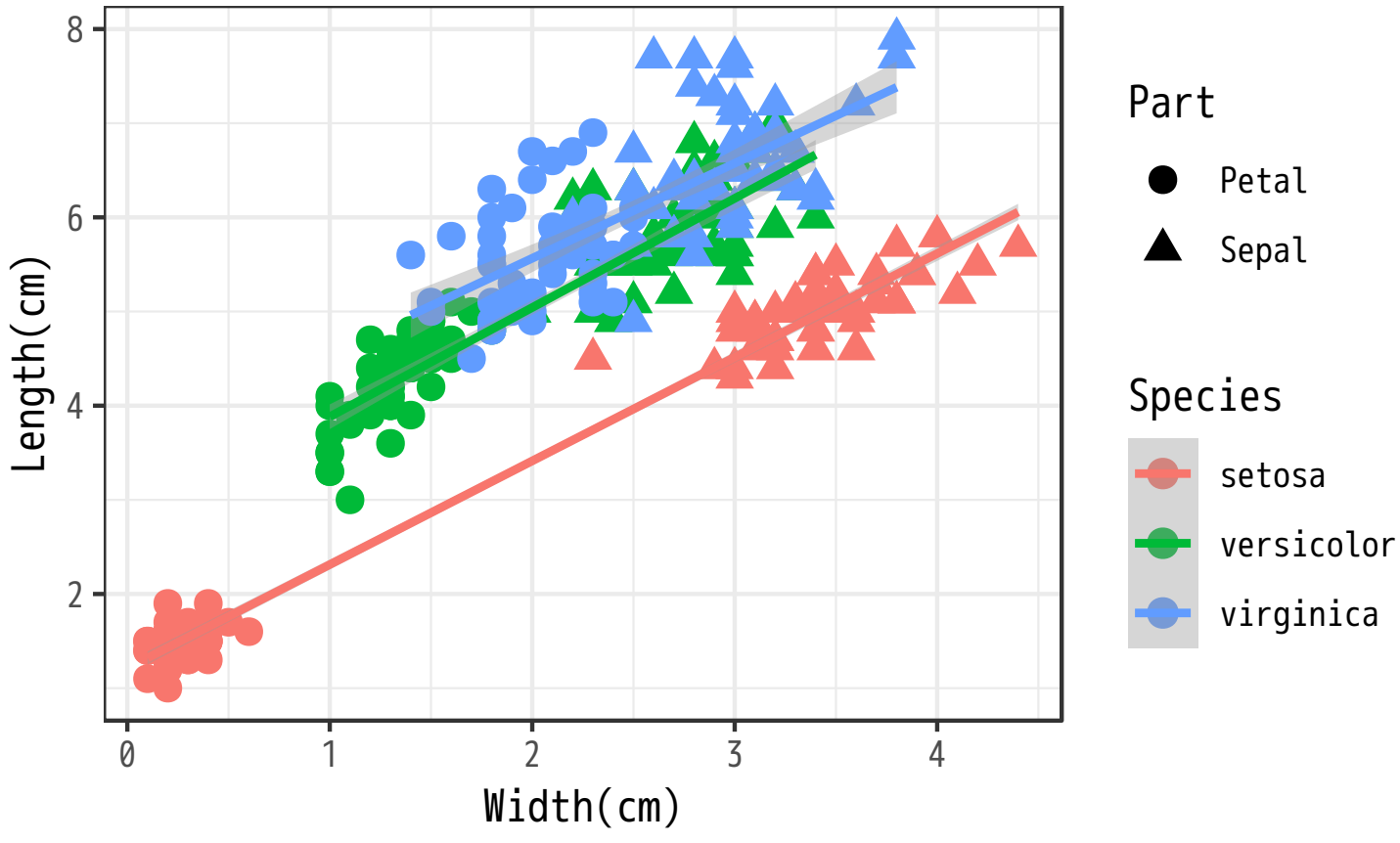

Figure 4: 傾きの比較

線形モデルの結果を数値で取得する

いままではグラフで確認していましたが、モデル化の結果を数値で取得することもできます。係数は推定値と標準誤差が estimate と std.err に、無相関の t 検定の結果は static と p.value に示されています。

```
lm_coef <- iris_long %>%
  group_by(Species) %>%
  summarise(list(lm(Length ~ Width) %>% tidy())) %>%
  unnest()
```

Species	term	estimate	std.error	statistic	p.value
setosa	(Intercept)	1.2112	0.035693	33.933	5.6621e-56
setosa	Width	1.1011	0.014594	75.453	1.2694e-88
versicolor	(Intercept)	2.7233	0.111338	24.460	1.6248e-43
versicolor	Width	1.1595	0.050909	22.776	6.2632e-41
virginica	(Intercept)	3.5476	0.252439	14.053	3.1885e-25
virginica	Width	1.0090	0.098541	10.239	3.6690e-17

Figure 5: 線形モデル係数

線形モデルの結果を数値で取得する

ここから 95% 信頼区間を求めるには、以下のように直接計算することもできますし、

```
ci_lwr <- function(coef, err, n, p = 0.95) {
  coef -1 * err * qt(df = n - 2, p = 1 - (1 - p)/2)
}

ci_upr <- function(coef, err, n, p = 0.95) {
  coef +1 * err * qt(df = n - 2, p = 1 - (1 - p)/2)
}

lm_coef_ci <- lm_coef %>%
  filter(term == "Width") %>%
  group_by(Species) %>%
  summarise(ci_lwr = ci_lwr(estimate, std.error, 100),
            ci_upr = ci_upr(estimate, std.error, 100))
```

線形モデルの結果を数値で取得する

以下のように信頼区間を求める関数を使うこともできます。

```
lm_coef_ci <- iris_long %>%
  group_by(Species) %>%
  summarise(list(lm(Length ~ Width) %>%
                   confint(., "Width") %>% as_tibble())) %>%
  unnest()
```

その結果は以下のようになり、傾きが同一である可能性があるといえます。

Species	2.5 %	97.5 %
setosa	1.07219	1.1301
versicolor	1.05848	1.2605
virginica	0.81341	1.2045

Figure 6: 傾きの信頼区間

線形モデルの妥当性確認

最後に、直線回帰が適切に行えているのか、残差の分布を確認してみましょう。

```
lm_resid <- iris_long %>%
  group_by(Species) %>%
  summarise(list(lm(Length ~ Width) %>% augment())) %>%
  unnest()

p1 <- lm_resid %>%
  ggplot(aes(x = Width, y = .resid)) +
  geom_point(alpha = 0.4) +
  facet_wrap(~ Species, scales = "free_x")

p2 <- lm_resid %>%
  ggplot(aes(x = .resid)) +
  geom_histogram(bins = 8) +
  facet_wrap(~ Species, scales = "free_x")
```

線形モデルの妥当性確認

説明変数（Width）に対して残差のバラツキはほぼ均等のようですし、残差の分布も偏っていないようなので、問題ないでしょう。

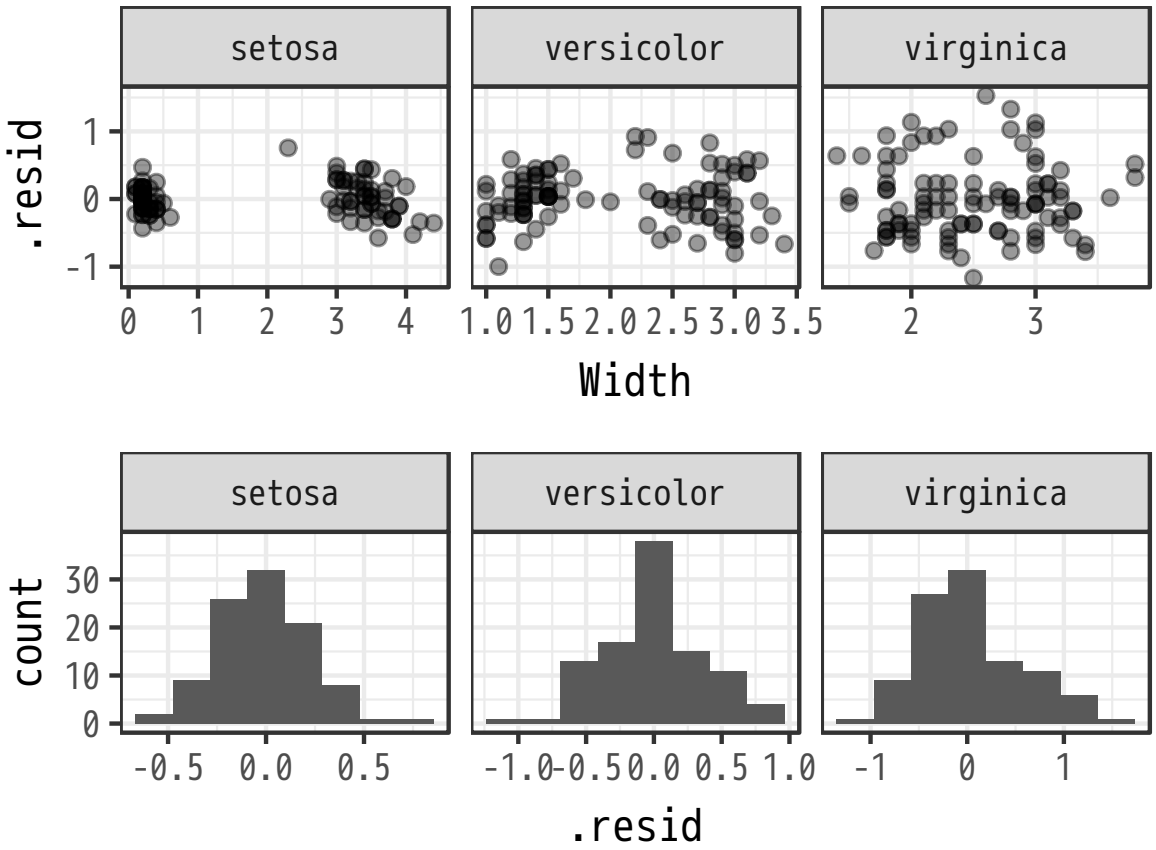

Figure 7: 残差プロット