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

1　データ分析プロセスは探索的

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。[*1] ここでは下図で示されるデータ分析プロセスの流れを、簡単な例を元に確認していきましょう。[*2]

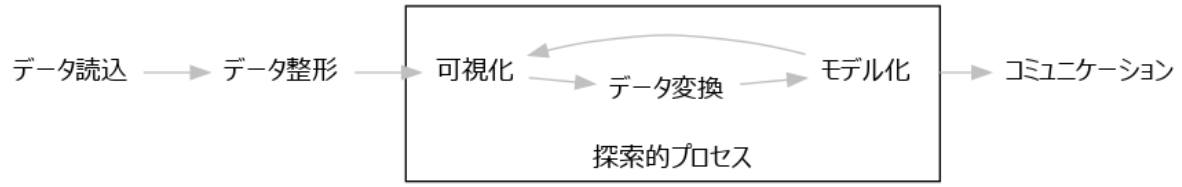

図1　探索的データ分析

2　実験データを読み込む

まず行うのは、実験データを格納したデータファイルを R に読み込むことです。R では、CSV（カンマ区切り）などのテキストファイルや Excel ファイルを簡単に読み込むことができます。[*3]
ただし、ここでは読み込む手順は省略し、R に組み込みのテスト用データ `iris` 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
```

[*1] 多角的な視点でデータ分析することは重要ですが、ツールの支援がなければ大きな負担になってしまいます。R はデータ分析の全てのプロセスを支援できるツールです。

[*2] ここでは R の書き方の説明は行いません。データ分析の考え方を大まかに理解してください。この内容をすべて使えるようになったらこのコースは終了です。また、この資料は『R ではじめるデータサイエンス』(2017) や R コンソーシアム、R Studio カンファレンスの発表を元にしています。

[*3] 計測器には独自形式のバイナリファイルでデータを記録するものがありますが、その場合でもデータの分析を他のツールでも行えるように、データ変換機能が用意されていることがほとんどです。

```
## 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
```

3　データを整形する

次に行うのは、読み込んだデータを分析に適した形に整形することです。例えば、複数の実験データを 1 つの Excel シートに混在させている場合、それぞれが区別できるように整える必要があります。[*4]
ここで使用するデータは整った形をしていますが、部位ごとの寸法が変数（列）になっているのを、部位と寸法を別変数にした方が、この後の分析がやりやすくなります。

そこで、以下のように整形します。なお、id は個体番号です。

```
iris_long <- iris %>%
  rowid_to_column("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
## 5      3 setosa  Petal    1.3   0.2
## # ... with 295 more rows
```

4　可視化する

計測データには、測定系の不具合や人為的なミス、環境の影響などによる不正なデータが含まれていることがあります。まず生データを可視化し、正常なデータか確認することが重要です。
ここでは長さと幅の関係を、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)) +
```

[*4] 余分なデータが含まれている場合など、データ読み込みの際に工夫する必要がある場合もあります。

```
    geom_point(aes(color = Part)) +
    facet_grid(Part ~ Species) +
    labs(x = "幅 (cm)", y = "長さ (cm)", color = "部位")
```

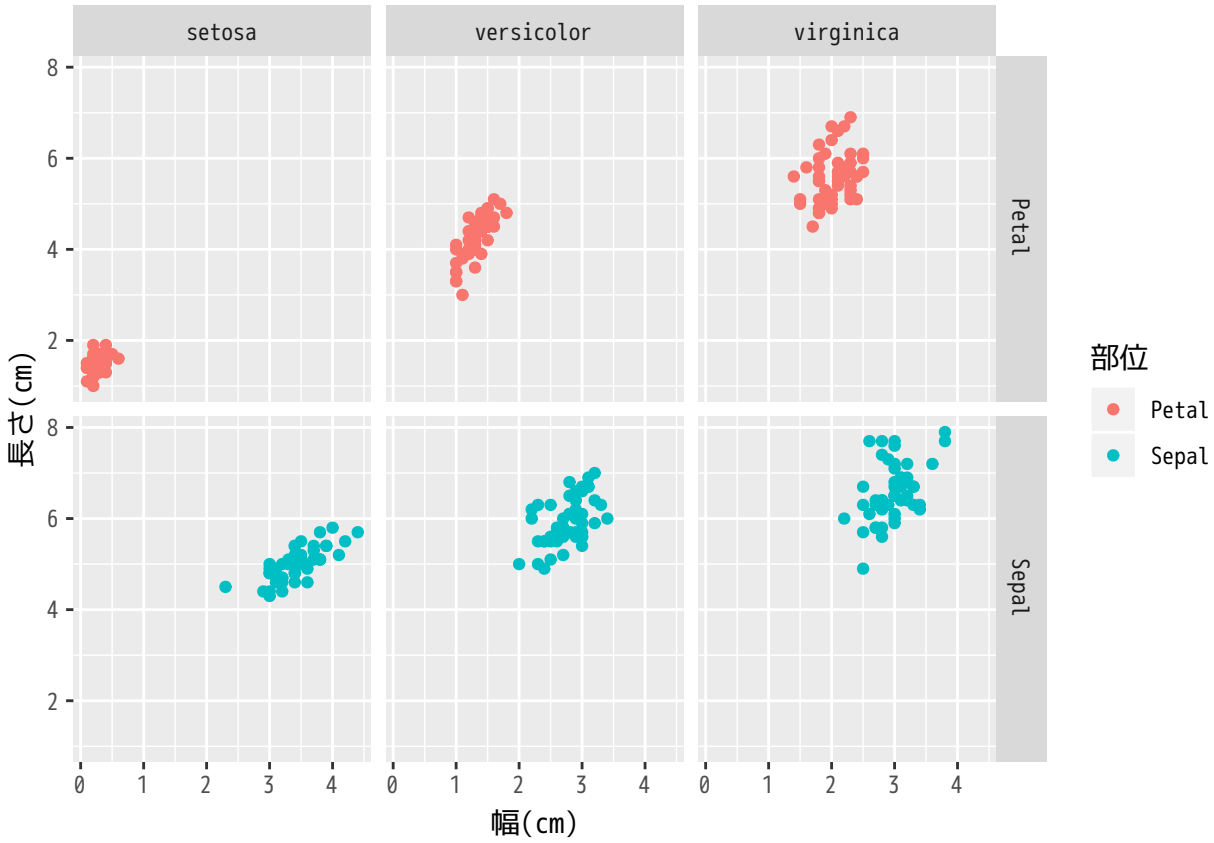

図2　種ごと部位ごとに可視化する

比較的バラツキの大きいデータなので問題ない範囲だと思いますが、setosa のガク（sepal）の幅が最小となるデータが他から少し外れているようです。

5　データを変換する

データ分析の対象が、大量の計測データの一部だったり、計測データから求めた演算結果だったりする場合に行うデータ処理のことを、変換と呼びます。
ここでは先ほどチェックしたときに注目したデータが計測ミスによる外れ値であると仮定して、除外してみましょう。

```
iris_long <- iris_long %>%
  filter(!(Species == "setosa" & Part == "Sepal" & Width < 2.5))
```

6　モデル化する（仮説 1）

長さと幅に相関があるか、95% 信頼区間つきの近似直線[*5]を描いて確認してみましょう。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。

```
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")
```

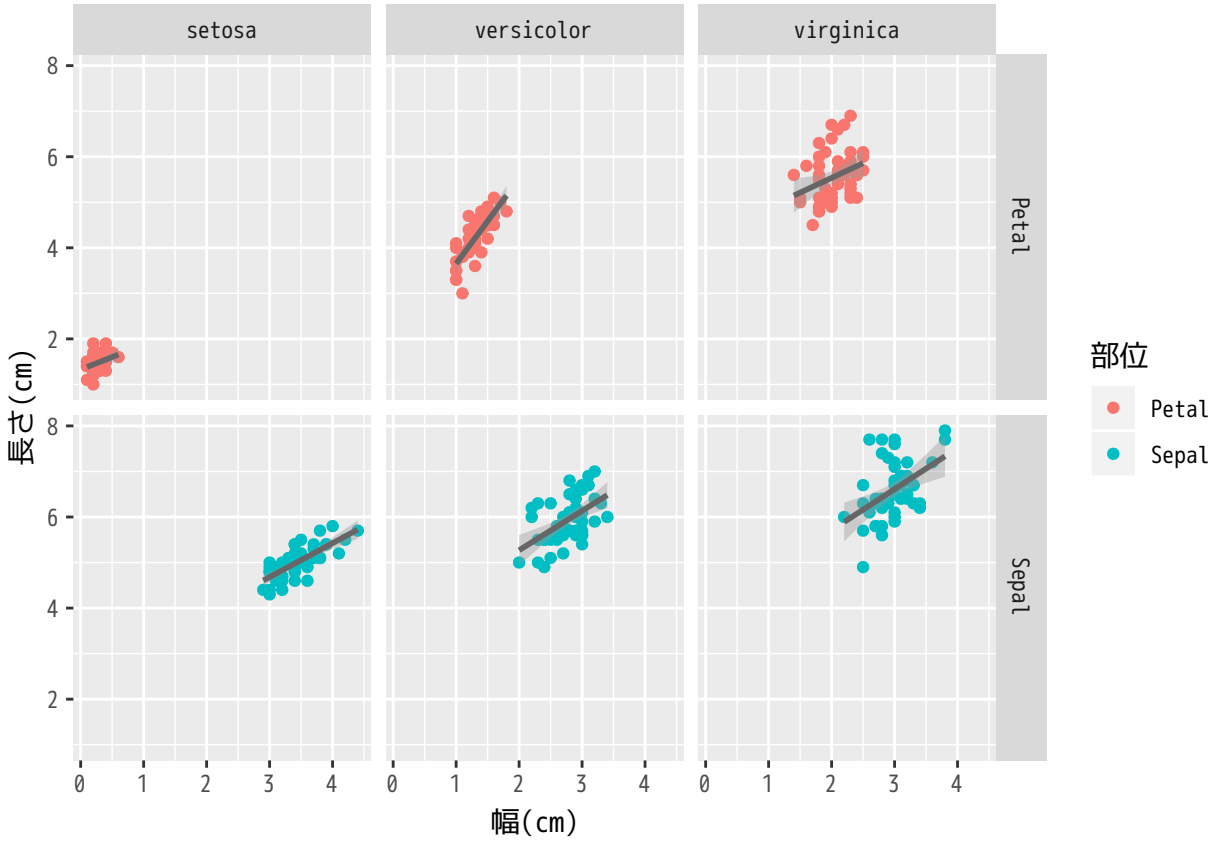

図3　仮説 1：長さと幅に相関がある

[*5] ここでいうモデルとは、仮説を確認できる数学的モデルのことです。この場合は、信頼区間つきの近似直線を描くことがモデル化です。

7　モデル化する（仮説 2）

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。花弁とガクをまとめたグラフを作って、確認してみましょう。

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

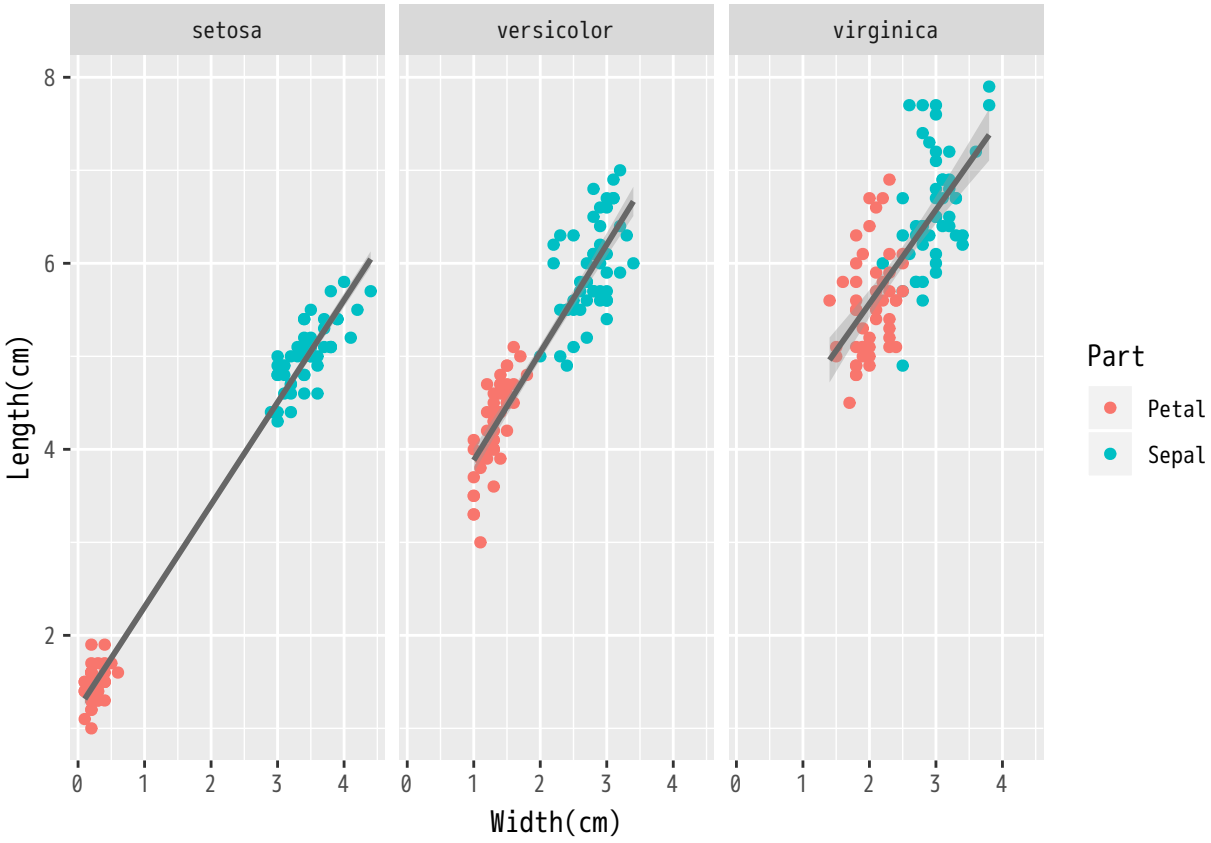

図4　仮説 2：長さと幅の関係は部位で共通

個別に描いていた近似直線と傾きが同じ程度なので、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の種でも似ているようです。

8　モデル化する（仮説 3）

長さと幅の傾きは種が異なっても共通かもしれません。傾きが確認しやすいようにグラフを作り変えて、確認してみましょう。

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

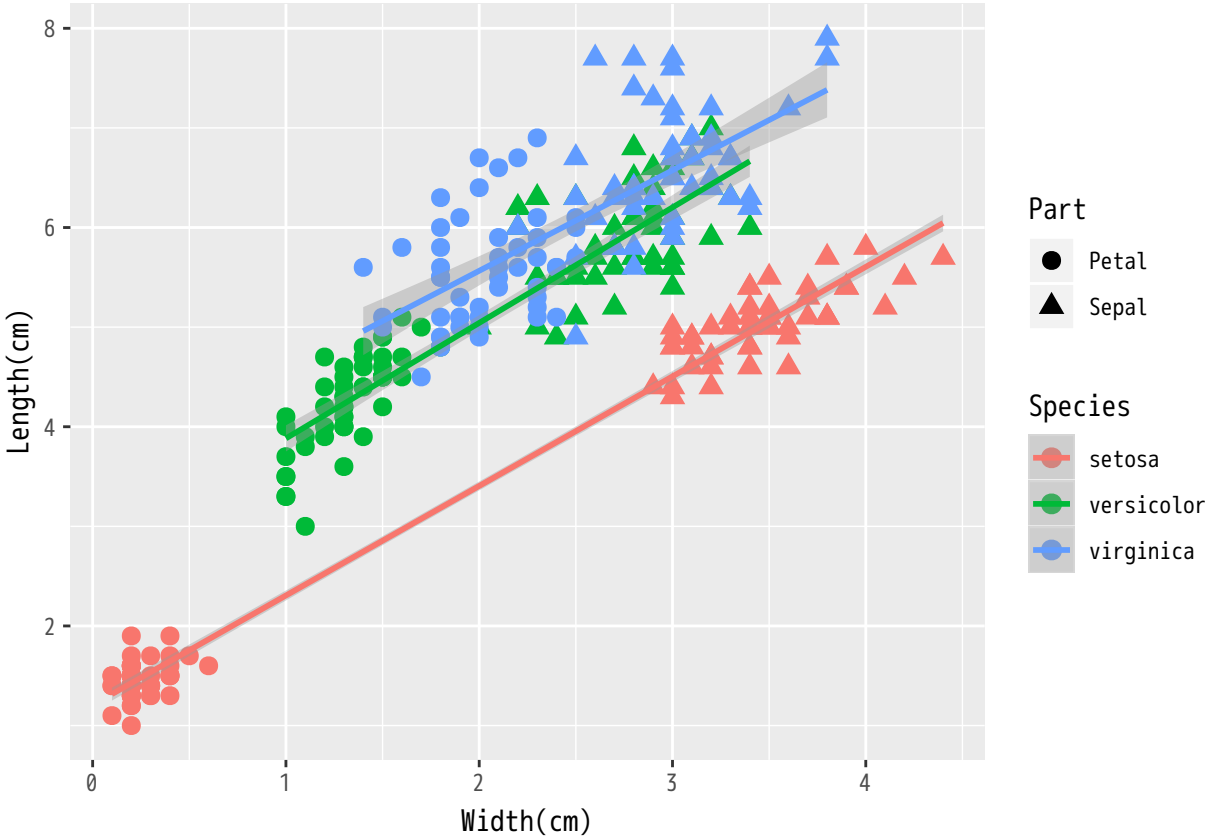

図5　仮説 3：傾きは同一に見える

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。

9　モデル化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。ここでは仮説 3 のモデル化の結果を見てみましょう。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)	1.2060	0.033951	35.522	2.1036e-57
setosa	Width	1.0998	0.013874	79.272	5.3958e-90
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

図6　線形モデルの結果を確認する

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))
```

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。

```r
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	1.07226	1.1273
versicolor	1.05848	1.2605
virginica	0.81341	1.2045

図7　仮説 2：傾きは同一といえる

10　モデル化の妥当性を確認する

モデル化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。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。

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

lm_resid %>%
  ggplot(aes(Width, .resid)) +
  geom_point(alpha = 0.4) +
  facet_wrap(~ Species, scales = "free_x")
```

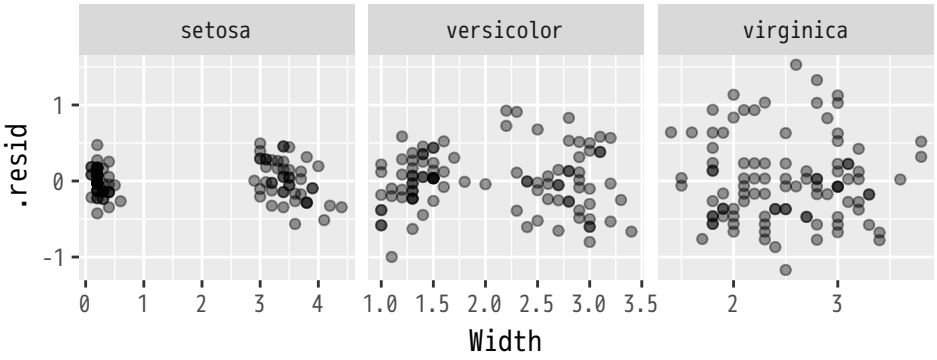

図8　残差プロットを確認してモデルをチェックする

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

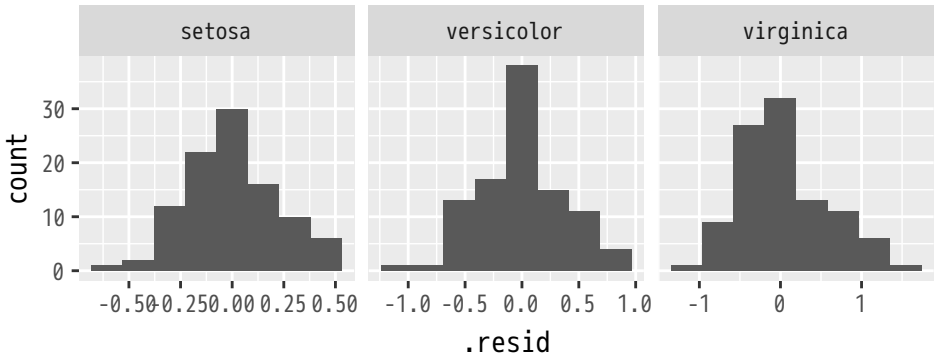

図9　残差プロットを確認してモデルをチェックする

11　分析結果を伝える（コミュニケーション）

データ分析が終了しても、レポートを作成して結果を報告するプロセスが残っています。R で作成したグラフや表をコピー＋ペーストする方法では、量が多くなると効率が悪くなりますしミスが混入しやすくなります。そこで、R にはレポートを自動作成する機能も備わっています。[*6]

[*6] この文書自体、R で作成しています。また、R のコードと結果を一緒にまとめることでデータ分析の再現可能性を高めることができます。