





R 言語の基本

操作の基本

- 使い方はテキストベースのコマンド
- 行の最後は「Enter」キー
- 入力中の修正は   キーで修正位置に移動可能
-   キーで過去に入力したコマンドの呼び出し可能
- 無意味な文字列は「エラー」となる
- 1行で完結しない場合はプロンプトが「+」に変化→複数行に分けた入力も可能。

「変数 (オブジェクト)」の定義

- **変数 = データ表現**
- データ表現は「関数」, 「式」, 「数値」など
- = の代わりに <- (2文字) を使ってもよい。
- **データ表現** → (2文字) **変数** とすることも可能 (向きと順序に注意)

変数やデータフレームの参照

- 変数名やデータフレーム名のみ : 値・内容の表示・参照
- データフレーム名\$カラム名 : 特定のカラム (列) のみを表示・参照

「数式」の表現

- 目的変数 ~ 独立変数 : 単一の場合
- 目的変数 ~ 独立変数 1+独立変数 2 : + で複数の変数を指定

主要な関数

汎用目的

- **c** : データの連結 (ベクトル, あるいは, 数列)
- **summary** : データの要約統計や, 詳細な情報の表示
- **head** : データの先頭部分のみ表示

グラフ関係

- **plot** : データの散布図描画
- **abline** : 現在表示されているグラフへの水平線・垂直線の追加, 切片・傾きを与えた直線の追加

データ入出力

- **read.csv** : テキスト形式 (CSV 形式) のファイルからのデータの読み込み (読込結果は「データフレーム」となる)

算術関数・統計処理

- **lm** : 線形回帰分析の実行

実行結果の処理

実行結果はテキストとして「コピー」できるので、そのまま Word などへ貼り付けることで利用できる。Word の「段落」パネルで改行ピッチ（行間）を「固定値」で「12pt」などにするだけで、見やすく整形できる。フォントも等幅系（「MS ゴシック」や「MS 明朝」あるいは「Courier new」など）に変更しておくで、数字の位置も揃うので、さらに見やすくなる。

グラフの部分は、グラフ描画ウィンドウの「ファイル」メニューから「クリップボードにコピー」→「ビットマップとして」（または「メタファイルとして」）を選んでおくと「コピー」されるので、Word などにそのまま（グラフィックスとして）貼り付けできる。

Excel シートのデータを R で利用する手順

シートの整形

1. 1 行目は（R 内部で）変数名として使えるように、半角の英字と数字を基本とした名前にする。重複がないように注意する。
2. 2 行目以降に、1 行=1 サンプルとなるようにデータを埋める。
3. 桁区切り「,」は使わない（「セルの書式設定」で「標準」に戻すか、「数値」のオプションで桁区切りの表示を停めておく）。
4. Excel ブックのまままで上書き保存する（データの修正時には Excel のままが便利なので）。
5. 「ファイルの種類」を「CSV(テキスト形式)」に変えて、適当なフォルダに「名前を付けて保存」する。警告に対しては「はい」で応答する（2 回）。
6. いったん Excel を終了する（この時の警告は「いいえ」（ファイルを再保存しない）で応答する）。

R での読み方

1. 関数 `read.csv` を使用する。あらかじめ R の「ファイル」メニューで「ディレクトリの変更」を行なっておく（CSV 形式のファイルを保存したフォルダに変更）。
2. 適当な名前の変数（オブジェクト）として
変数 = read.csv("保存した CSV 形式のファイル名")
で読み込む。内容の確認は、変数名のみを入力して確認する。1 行に複数のデータが混在できるので、特に「データフレーム」と呼ばれる。
3. Excel 上で 1 行目に設定しておいた名前が、そのまま列名（カラム名）として利用できる。特定の列だけを使う時は **変数名\$カラム名** の形式で利用する。

R による分析例（1）

初回の授業で宿題としてあった `qa_01ex.pdf` の第 5 問を R で処理してみる。

R を起動後、`x` と `y` という 2 種類のデータを、以下のように定義しておく。各行の先頭の「`>`」は、プロンプトなので入力する必要はない。

```
> x=c(20, 30, 40, 50, 60, 70)
> y=c(61, 90, 95, 129, 120, 161)
```

内容が正しいか確認するには、それぞれの名前のみを入力する。青字は、Rからの出力を示している。

```
> x
[1] 20 30 40 50 60 70
> y
[1] 61 90 95 129 120 161
```

データの散布図を plot で描いてみる。

```
> plot(y ~ x)
```

新たにグラフ用のウィンドウが開かれ、図 1 のようなグラフが表示される。

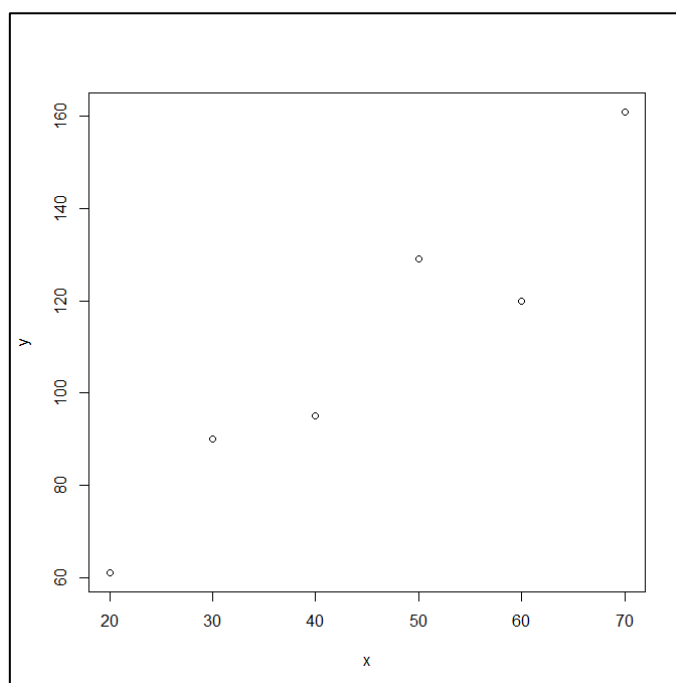


図 1 コマンド plot による散布図の描画結果

y を被説明変数、x を説明変数とする回帰分析を行なってみる。

```
> lm(y ~ x)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)          x
    29.105         1.783
```

処理結果は、与えられた数式の形と回帰係数 (coefficient) のみが表示される。より、

詳しい情報を得るために、処理結果を新たな変数¹に保存して、`summary`を使用することで得られる。

```
> lm.result = lm(y ~ x)
> summary(lm.result)

Call:
lm(formula = y ~ x)

Residuals:
    1     2     3     4     5     6 
-3.762  7.410 -5.419 10.752 -16.076  7.095 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  29.1048    13.1540   2.213  0.09136 .
x              1.7829     0.2733   6.524  0.00285 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.43 on 4 degrees of freedom
Multiple R-squared:  0.9141,    Adjusted R-squared:  0.8926 
F-statistic: 42.56 on 1 and 4 DF,  p-value: 0.002851
```

オブジェクト `lm.result` には、その他の情報も含まれているが、`plot` 直後のグラフに `abline` を使って回帰直線を引くこともできる²。図 1 の散布図が表示されたままであることを確認したら、`abline` を用いて直線を引く。

```
> abline(lm.result, col="red")
```

見分けがつくように、描画色を赤 (`col="red"`) に指定した。結果は、図 2 のようになる。

¹ 正しくは「オブジェクト」と呼ぶ。

² 単回帰（説明変数の数が1つ）の場合のみ。

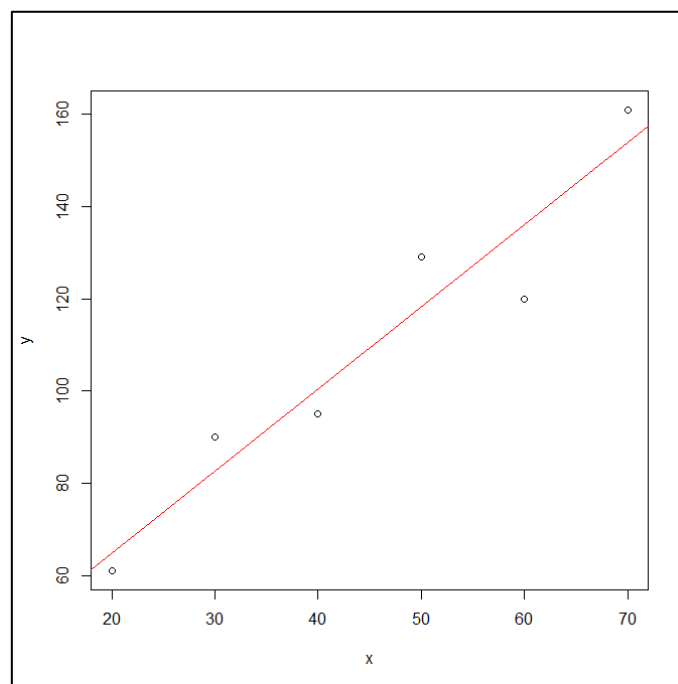


図 2 「図 1」の散布図に `abline` で回帰直線を加えた例

Rによる分析例（2）

データ定義を、`x=c(...)` のような形で行うと、入力ミス元になるので、あらかじめ外部ファイルとして保存してあるデータを読み取るのが一般的である。例題として用意してある `data0.csv` を適当な場所に保存したら、保存場所に応じて、R の「ファイル」メニューで「ディレクトリの変更」を行なっておく³。

```
> data0=read.csv("data0.csv", header=TRUE)
```

内容が正しく読めているか、`head` で最初の数行分を確認してみる。

```
> head(data0)
  No Height Sex Father Mother
1  1   162   1   161   147
2  2   165   1   156   152
3  3   167   1   168   158
```

(次ページに続く)

³ R の `setwd` コマンドで、

```
> setwd("保存場所のパス")
```

としてもよい。Windows の場合、パスの区切り記号 `¥` は `/` に置き換えて表記する。

例) 保存場所が `C:¥Users¥sakkun¥Desktop¥数量分析データ` の場合

```
setwd("C:/Users/sakkun/Desktop/数量分析データ")
```

と入力する。

(前頁からの続き)

4	4	167	1	167	158
5	5	168	1	166	158
6	6	168	1	160	155

変数どうしの散布図を作成してみる。data0は「データフレーム」と呼ばれる構造を持つデータとなっているので、その一部を用いるには、\$を使った表記で、

```
> plot(data0$Height ~ data0$Father)
> plot(data0$Height ~ data0$Mother)
```

とするか、data=でデータフレーム全体を指示して

```
> plot(Height ~ Father, data=data0)
> plot(Height ~ Mother, data=data0)
```

とする。また、データフレーム全体を指示すると、行列散布図を描いてくれる。

```
> plot(data0)
```

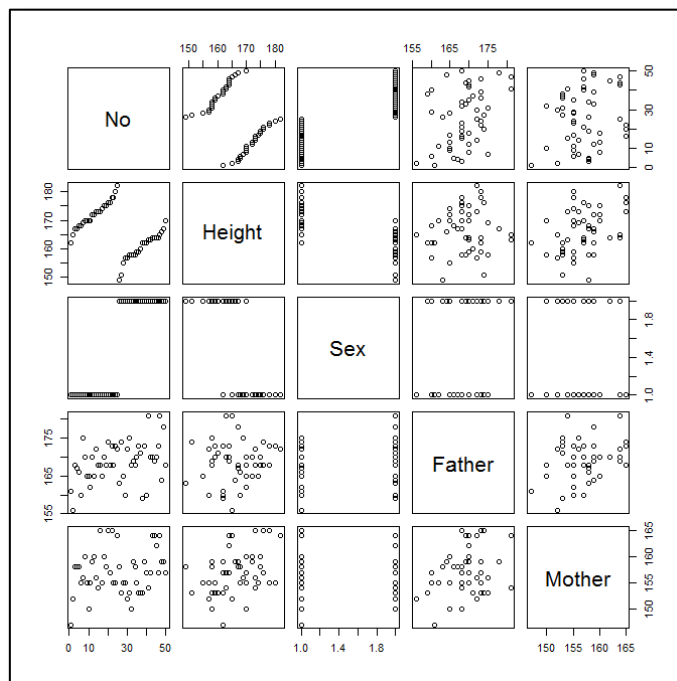


図 3 データフレーム全体を使った行列散布図の例

lmでの式の示し方も同様であるので、復習として、回帰分析を Height ~ Fatherあるいは Height ~ Motherの単回帰モデルで行ってみよ。