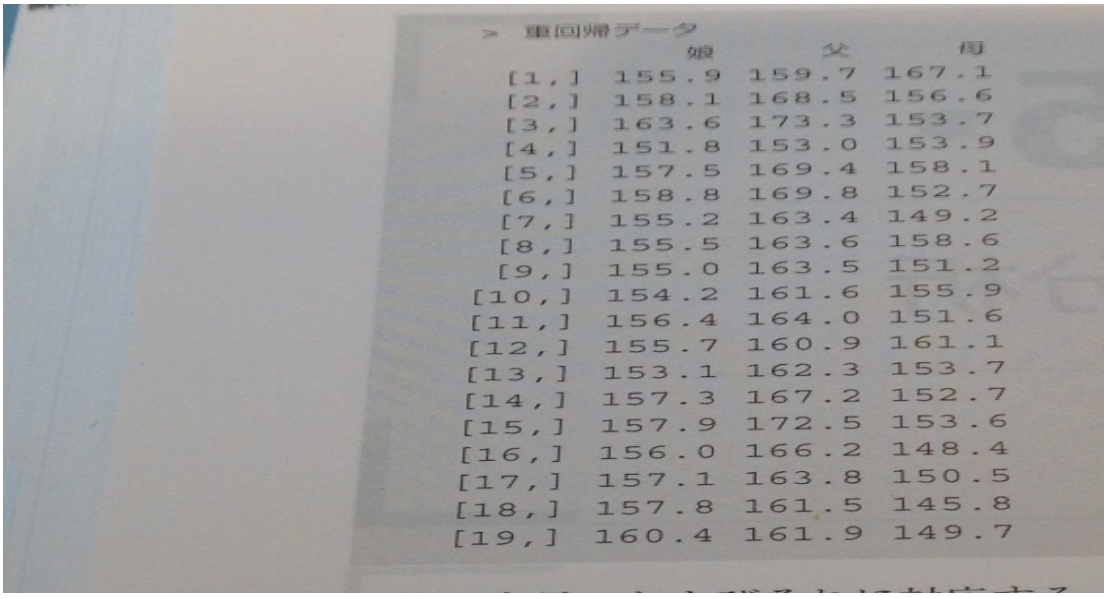
火曜４限　投野先生　英語教育学研究　金光翔一

Ⅰ知識編(『R によるやさしい統計学』より)

回帰分析

- ➢ 回帰分析とは

- ● 一つあるいは複数の変数の値を用いて、ある一つの変数の値を予測するために用いられる多変量解析の一手法
- ➢ 身長の例を今回は考える



```
> 重回帰データ
          娘     父     母
 [1,]  155.9  159.7  167.1
 [2,]  158.1  168.5  156.6
 [3,]  163.6  173.3  153.7
 [4,]  151.8  153.0  153.9
 [5,]  157.5  169.4  158.1
 [6,]  158.8  169.8  152.7
 [7,]  155.2  163.4  149.2
 [8,]  155.5  163.6  158.6
 [9,]  155.0  163.5  151.2
[10,]  154.2  161.6  155.9
[11,]  156.4  164.0  151.6
[12,]  155.7  160.9  161.1
[13,]  153.1  162.3  153.7
[14,]  157.3  167.2  152.7
[15,]  157.9  172.5  153.6
[16,]  156.0  166.2  148.4
[17,]  157.1  163.8  150.5
[18,]  157.8  161.5  145.8
[19,]  160.4  161.9  149.7
```

- ● 親の身長は一般的に遺伝すると考えられ、親の身長が高いほど子の身長も高いとされる
  →本当なのか？
  →親の身長とあるが母親と父親に何か違いがあるのか
- ✧ 母親の身長と父親の身長を使い娘の身長を予測するという回帰モデルを考える
- ● モデルの式

娘の身長=a+b1×父親の身長+b2×母親の身長+e

a→**切片**　b1、b2→**偏回帰係数**、e→**残差**　と呼ぶ

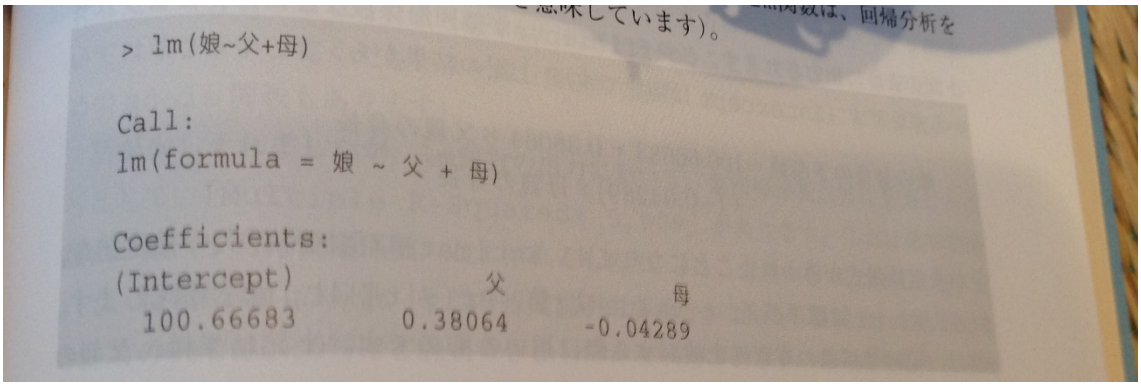
- 今回の目的は、データから最もよく娘の身長を予測すべく、２つの偏回帰係数を求めること。
- 残差とは**予測の誤差**のことで、ここでは父親の身長と母親の身長から予測した値と実際の値のずれ

➢ 用語

父親の身長と母親の身長→独立変数（予測変数、説明変数）
娘の身長→従属変数（基準変数、目的変数）

独立変数が複数あるもの→重回帰分析
独立変数が一つしかないもの→単回帰分析

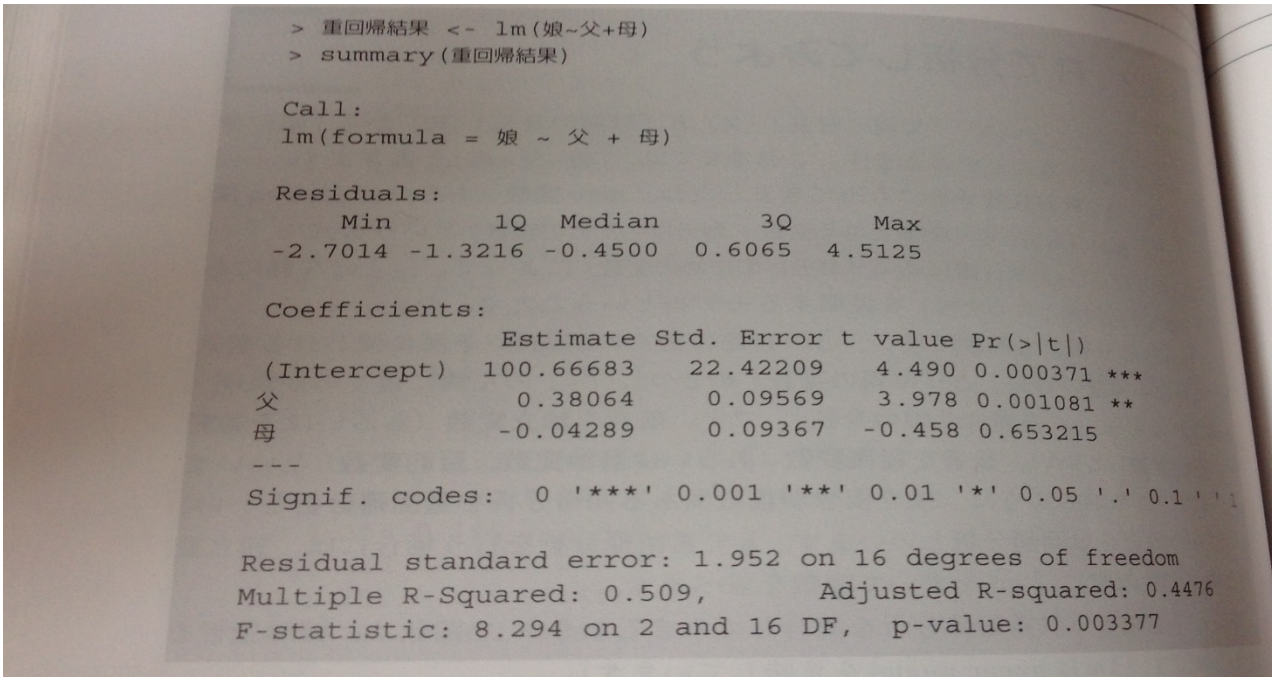
➢ 実際に重回帰分析をしてみる



- １ｍは関数(linear model)

✧ しかしこれだけでは論文など使うには不十分!

→summary 関数を用いる



```
> 重回帰結果 <- lm(娘~父+母)
> summary(重回帰結果)

Call:
lm(formula = 娘 ~ 父 + 母)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7014 -1.3216 -0.4500  0.6065  4.5125

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 100.66683   22.42209   4.490 0.000371 ***
父            0.38064    0.09569   3.978 0.001081 **
母           -0.04289    0.09367  -0.458 0.653215
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.952 on 16 degrees of freedom
Multiple R-Squared: 0.509,      Adjusted R-squared: 0.4476
F-statistic: 8.294 on 2 and 16 DF,  p-value: 0.003377
```

- Coefficients 以下は偏回帰係数などについて書かれている
- Intercept は切片
- 娘の身長の予測値=100.66683＋0.38064×父親の身長+(－0.04289)×母親の身長
- Estimate→偏回帰係数の測定値、std.Error→標準誤差、t value→t 値 Pr(>|t|)→p 値

✧ 偏回帰係数は父親は有意、母親は有意でない

- （大切！）Multiple R-squared: 0.509→<u>重相関係数の二乗</u>（または決定係数）

✧ この独立変数の説明力は 50%ほどあるということ

- F-statistics→F 値　8.294、自由度は(2,16)、p 値（p-value）→0.003377

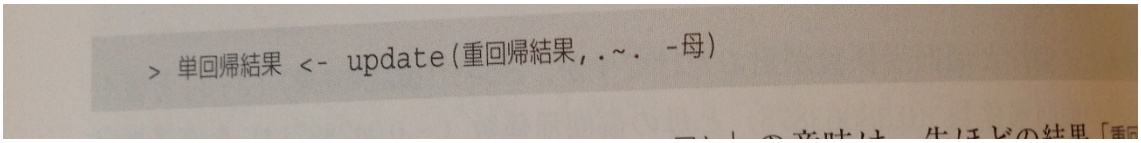
→有意水準 5%とすれば……..有意である

➢ モデル探索

- 実際には独立変数は数多くある
- 全ての変数を用いるわけではない
  →どの独立変数の組み合わせが最適か検討する

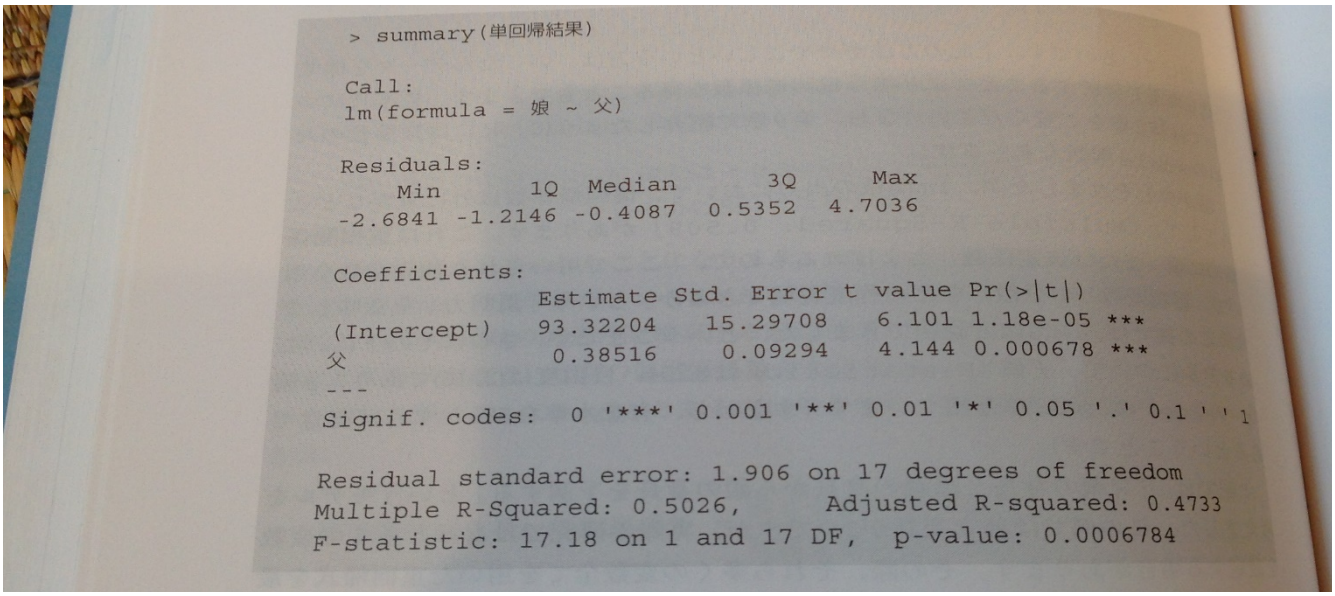
➢ 独立変数の変更

- 「父親だけ」考えてみる
- 1m 関数で指定し直す（母を外す）か update 関数でモデルの変更点を記述する→今回は update 関数



```
> 単回帰結果 <- update(重回帰結果,.~. -母)
```

- 「母」を取り去って再分析
- その結果



```
> summary(単回帰結果)

Call:
lm(formula = 娘 ~ 父)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6841 -1.2146 -0.4087  0.5352  4.7036

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  93.32204   15.29708   6.101 1.18e-05 ***
父            0.38516    0.09294   4.144 0.000678 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.906 on 17 degrees of freedom
Multiple R-Squared: 0.5026,     Adjusted R-squared: 0.4733
F-statistic: 17.18 on 1 and 17 DF,  p-value: 0.0006784
```

- 「1m(formula = 娘　~ 父)」となっており、「母」が消え、「父」のみになっている
- 説明力は約 50%→前とほぼ変わらず

✧ 重回帰分析では独立変数を取り除くと必ず取り除く前の値以下になる

- 娘の身長の予測値=93.32204+0.38516×父親の身長

➢ 回帰直線を描いてみる（上記の例）

- 横軸に「父親の身長」、縦軸に娘の身長の散布図　(plot(父、娘))

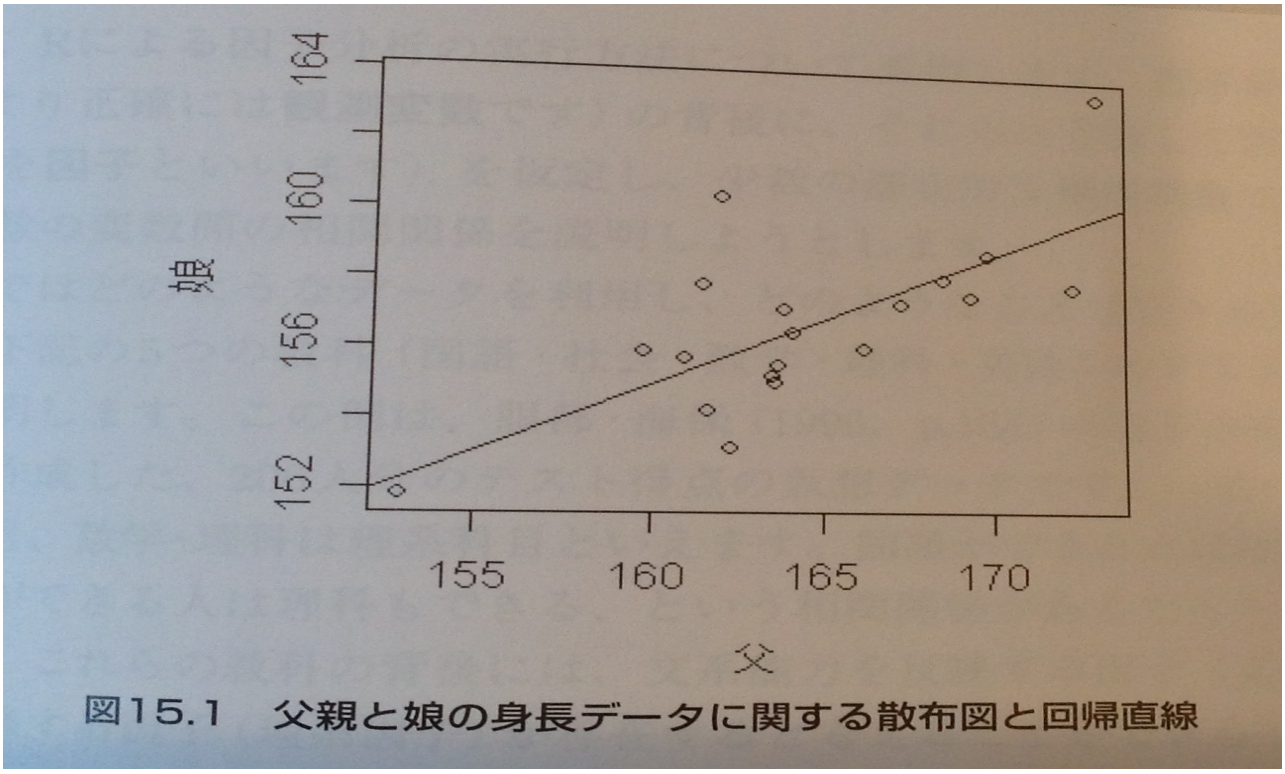


図15.1　父親と娘の身長データに関する散布図と回帰直線

Ⅱ演習編

- 始める前に…….

①授業用ページからデータセットダウンロード

②R 及び R コマンダーの立ち上げ

7.1.1 Coplots

- 使うもの

①データ: SPSS ファイル”LafranceGottardo.sav”、名前は lafrance

②コマンド：

coplot(l2phonemicawareness~grade1l1reading | nonverba

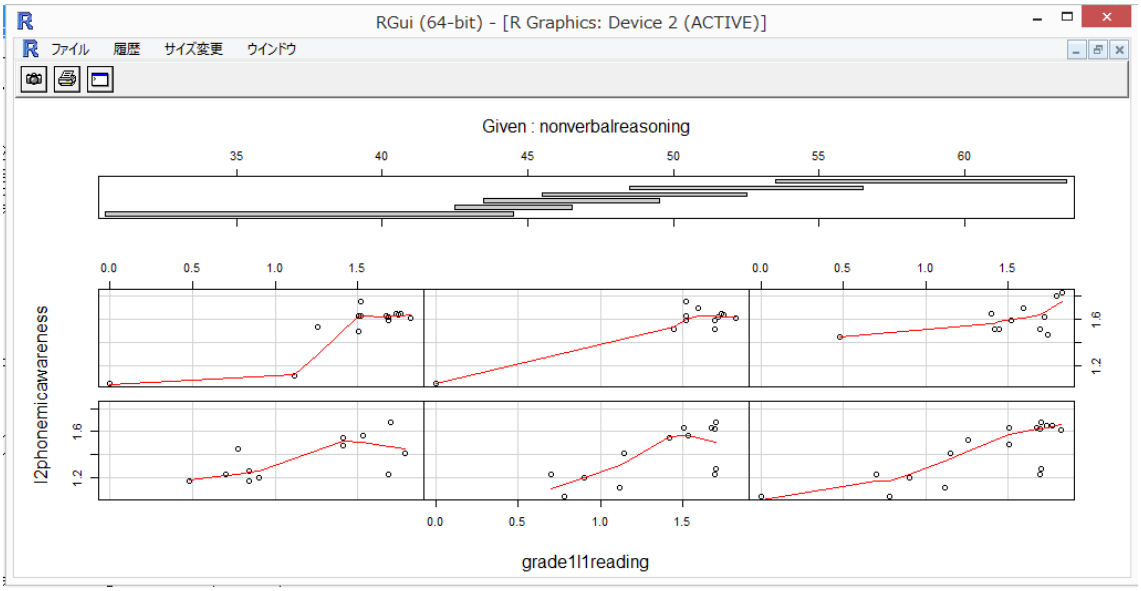
lreasoning, panel=

function(x,y,...)

panel.smooth(x,y,span=.8,...),data=lafrance)

➢ 手順

①R コンソールに上記のコードを入れる



7.1.2　3D Graph

➢ 手順

①グラフ→３次元グラフ→３次元散布図

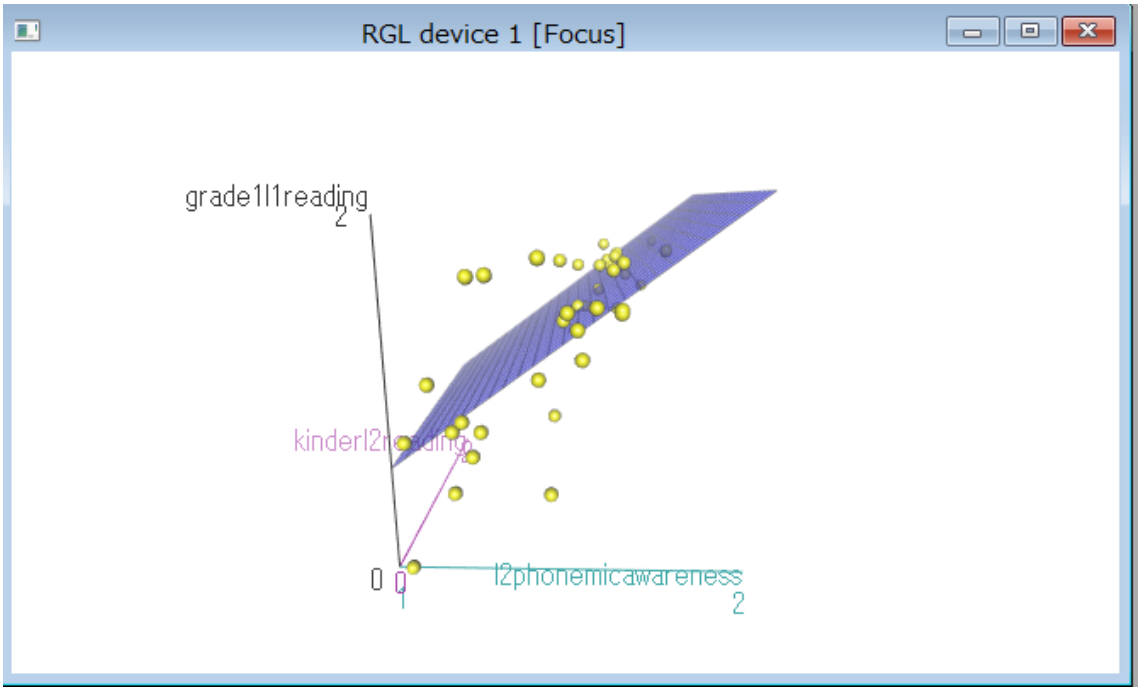
②変数をそれぞれ指定

③オプションでスムーズ回帰及び線形最小２乗に追加で✔

をつける。


3次元散布図
データ
オプション
軸のスケールを表示
表面のグリッド線を表示
残差の平方を表示する
表面の当てはめ
線形最小２乗
２次の最小２乗
スムーズ回帰
df =
<auto
加法回帰
自由度(各項) =
<auto
50%集中楕円のプロット
背景色
黒
白
点を特定
自動的に
マウスでインターラクティブに
確認しない
確認する点の数
2
ヘルプ
リセット
OK
キャンセル
適用

④OK をつける



7.1.3 Tree Models

➢ 使用するもの

①データ：SPSS ファイル”Lafrance5.sav,”、名前は lafrance5

②パッケージ： tree （ない場合はインストール）

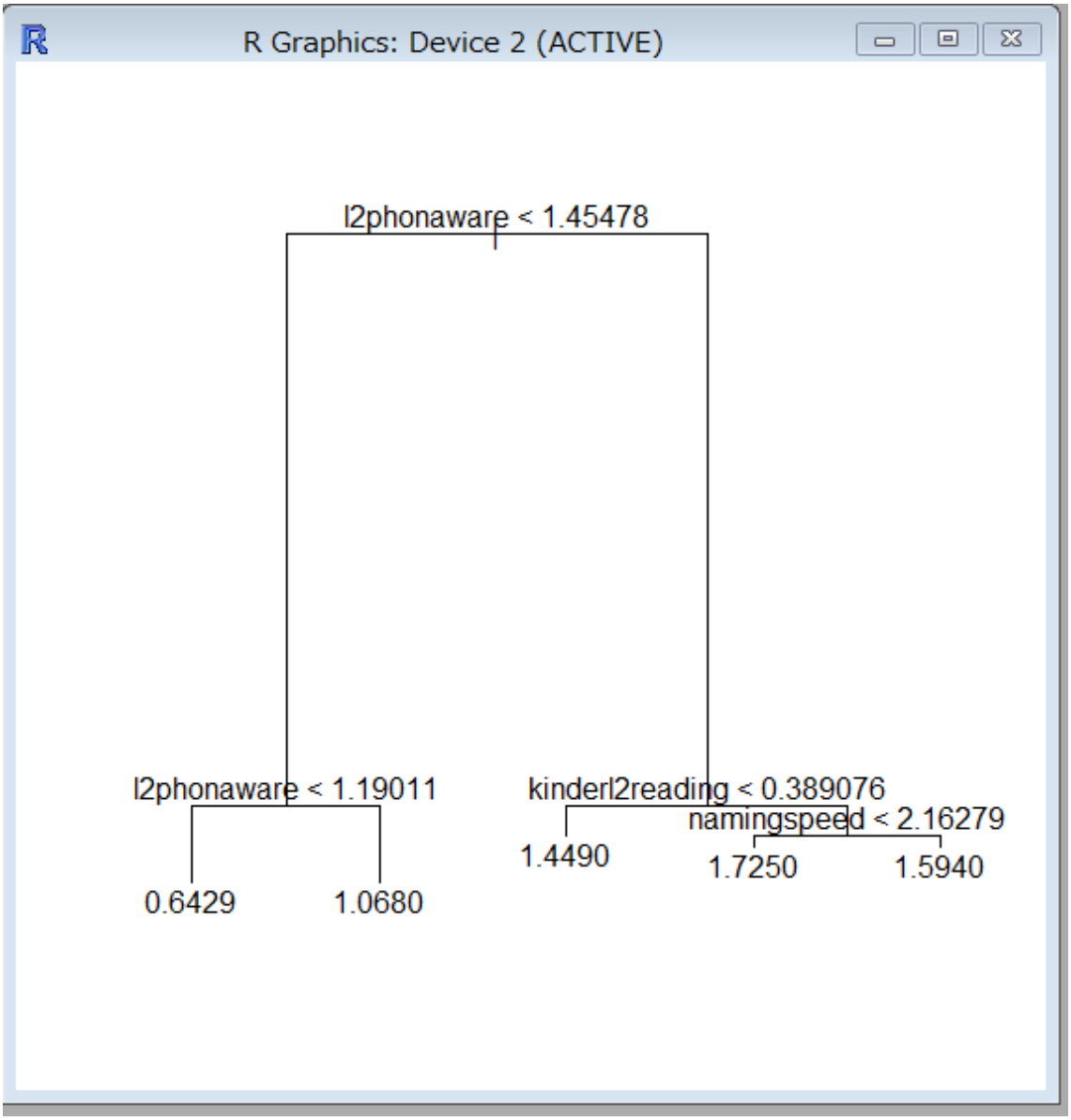
③コード:

```
library(tree)
model=tree(lafrance5)
plot(model)
text(model)
```

➢ 手順

①R コンソールに上記コードを入力

② R コマンダーに上記コードを入力して



- ある値を境にそれぞれの変数がどれくらい影響し合っているかを枝状の図でみるもの

➢ 今後の注意点

- **作ったモデルの名前はメモする**
- **特に数字は間違えないように**

7.3Doing the Same Type of Regression as SPSS

➢ 使用するもの

①考える変数：grade1l1reading（目的変数）

nonverbalreasoning+kinderl2reading

performance+namingspeed+workingmemory+phonologic

alawarenessInl2,

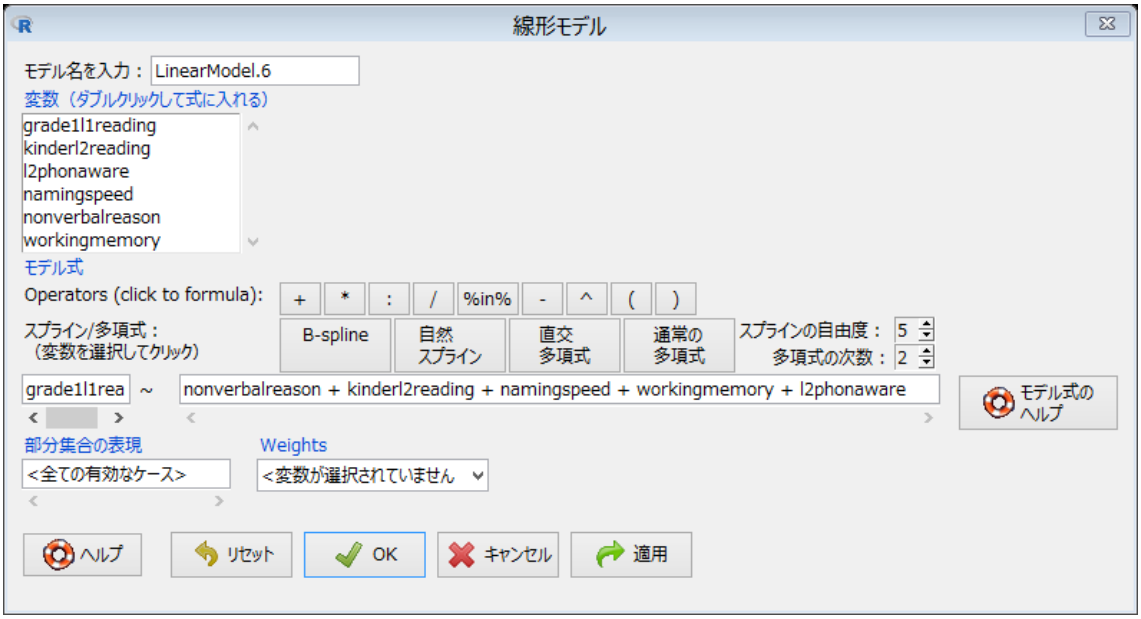
data=lafrance5（説明変数）

➢ 手順（シンプルな linear model）

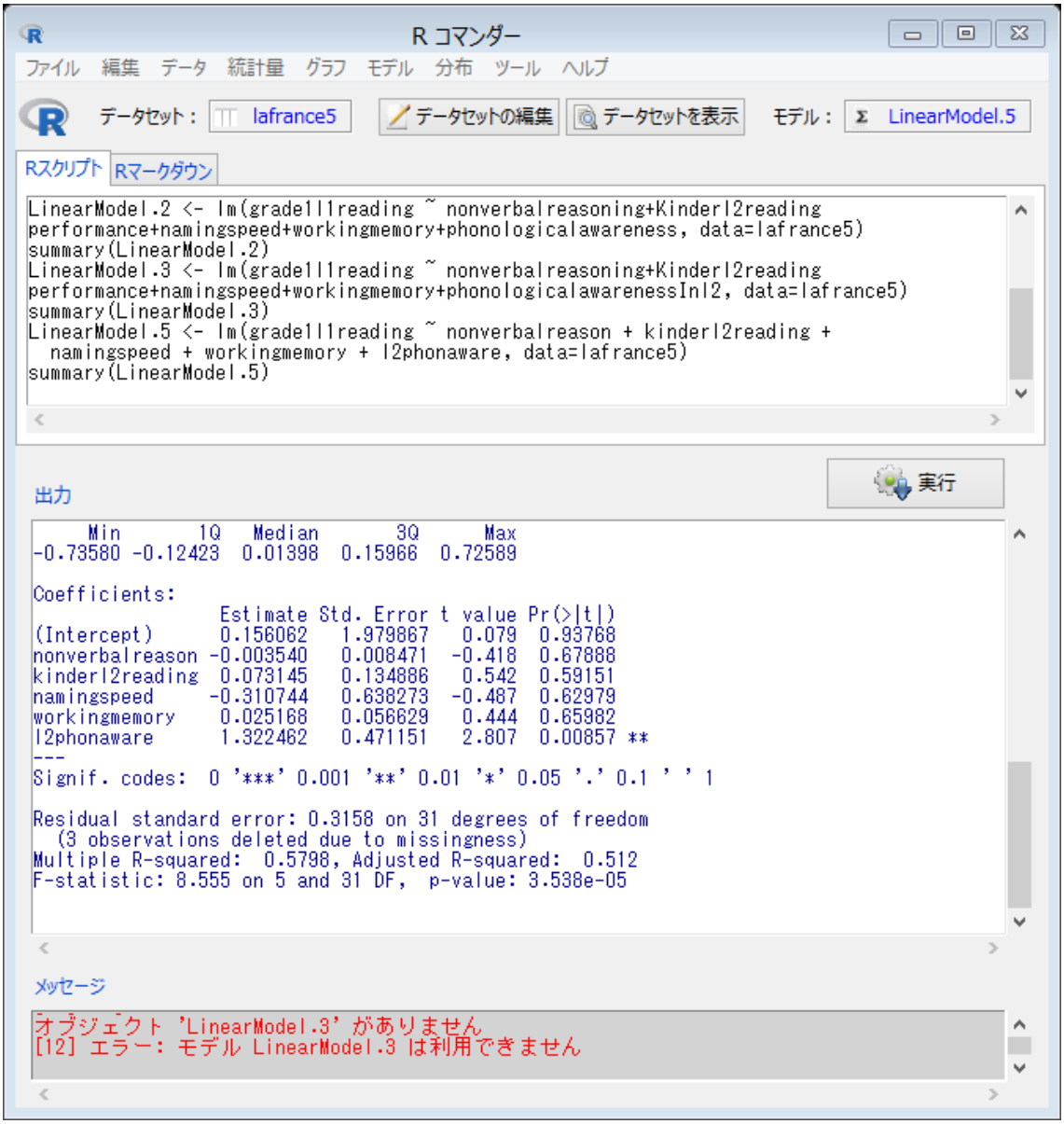
①統計量→線形モデル

②変数をそれぞれ入れる

左（目的変数）



※変数は左上のボックスから入れると良い



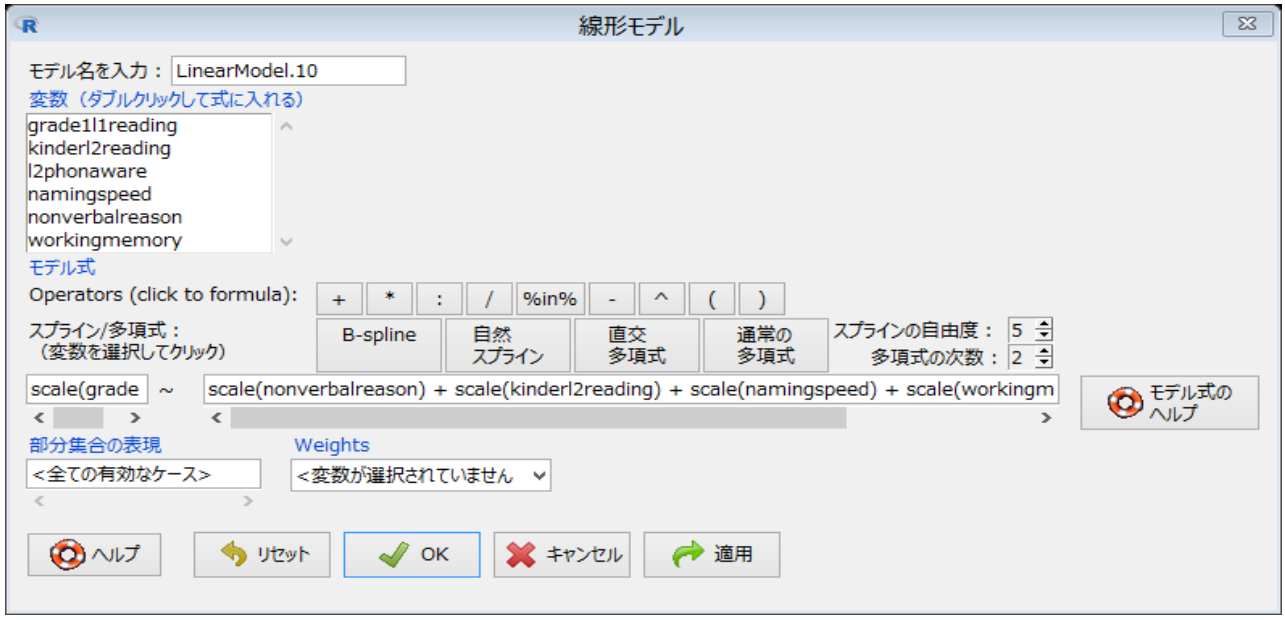
- 手順(標準化するモデル)

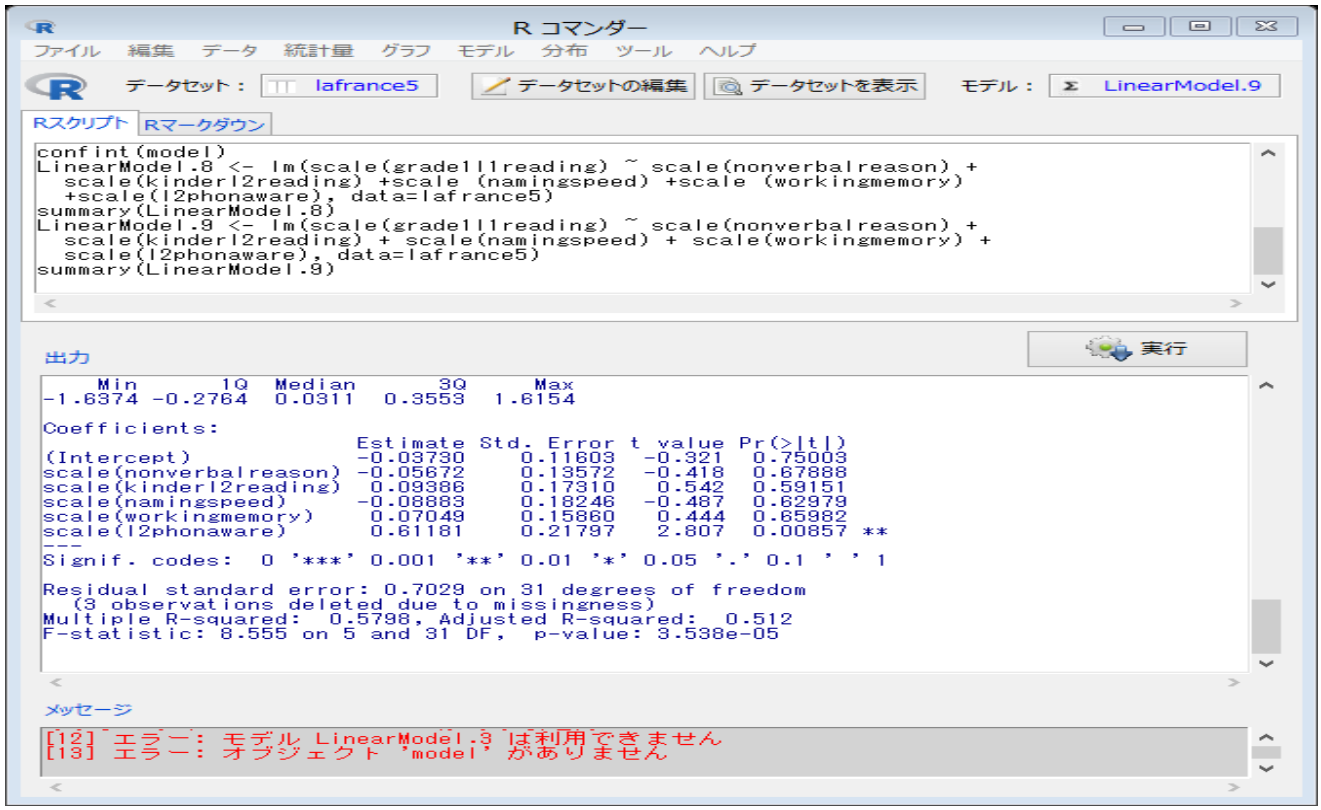
左：目的変数

scale(grade111reading)

右：目的変数

scale(nonverbalreason) + scale(kinderl2reading) + scale(namingspeed) + scale(workingmemory) + scale(l2phonaware)





- calc. relimp(model)

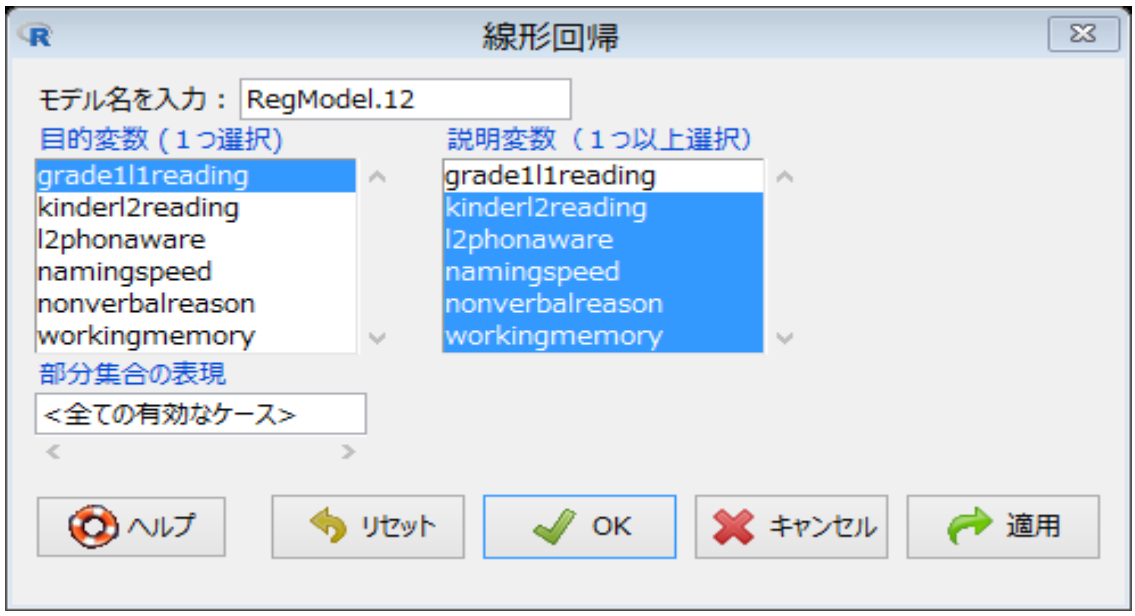
説明力のある変数を分かりやすく表示してくれるモデル

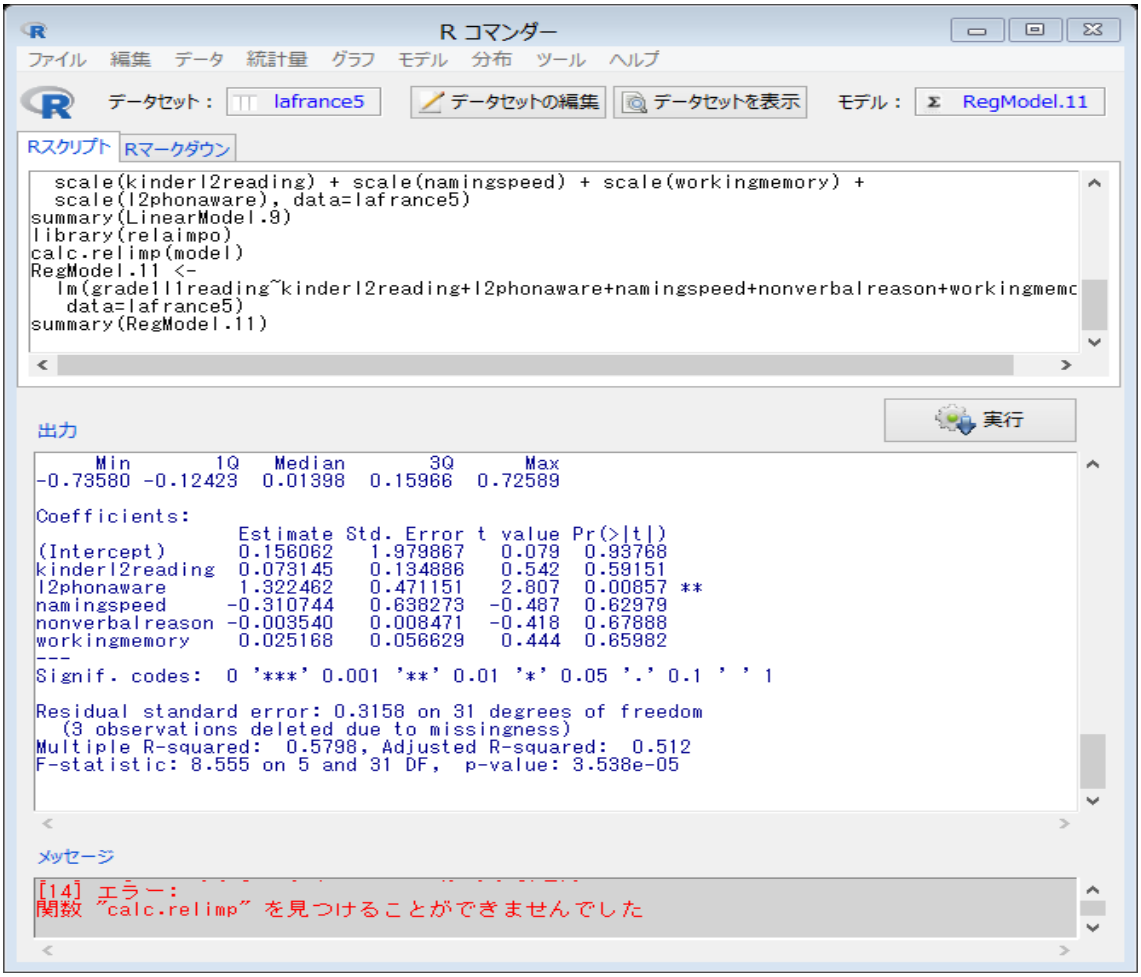
一つずつの変数のおよその説明力が記される

➢ モデルを作ってみよう！

①統計量→モデルの適合→線形回帰

②調べたい目的変数と従属変数を設定





7.3.1 Reporting the Results of a Regression Analysis

- 報告の際には変数同士の相関、非標準化係数を伝える必要があるがとりわけ R^2 値は説明力を端的にあらわすのでこれは特に伝えること

7.3.2 Reporting the Results of a Standard Regression

- 変数同士の内的な相関と変数単体の説明力の高さを示す

7.3.3 Reporting the Results of a Sequential Regression

- 予測変数を説明する最適な目的変数の組み合わせ、その説明力の高さを示す。ある一つの説明力の高い目的変数を基準として考える場合がある

7.5 Finding the Best Fit

- 様々な変数及びその組み合わせ方で最適なモデルを探す
- R コマンドの「線形回帰」で様々なモデルを試せる

→単に変数同士を足す、相互関係があるかたまりでみる、ある変数同士のかたまりから引くなど

7.5.1 First Steps to Finding the Minimal Adequate Model in R

➢ 使用するもの

①ファイル：SPSS ファイル、Lafrance5.sav、名前は lafrance5

②パッケージ： mice(なかったらダウンロード)

③コマンド ： ※（各行空けないと読み込んでくれない）

```
library(mice)
imp<-mice(lafrance5)
complete(imp)
```

lafrance<-complete(imp)

- 手順

①パッケージの mice を読み込む

②R コンソールに上記コマンドを打ち込む

③missing value を補ったデータが出る

	grade111reading	workingmemory	namingspeed
1	1.7634280	2	2.406489
2	0.7781513	1	2.568917
3	1.7323938	2	2.354953
4	1.5051500	2	2.177161
5	1.5051500	4	2.010342
6	1.6901961	4	2.201861
7	0.4771213	2	2.247556
8	0.6989700	2	2.319169
9	0.8450980	2	2.443310
10	1.7160033	3	2.119850
11	1.1461280	0	2.451495
12	1.4149733	4	2.239650
13	0.4771213	4	2.279325
14	1.7853298	4	2.123067
15	1.6901961	3	2.377215
16	1.6720979	4	2.161937
17	1.7481880	2	2.316159
18	1.6989700	2	2.377215
19	1.4471580	2	2.200440
20	0.8450980	1	2.319169

	g1l1reading	workingmemory	namingspeed
1	1.7634280	2	2.406489
2	0.7781513	1	2.568917
3	1.7323938	2	NA
4	1.5051500	2	2.177161
5	1.5051500	4	2.010342
6	1.6901961	4	2.201861
7	0.4771213	2	2.247556
8	0.6989700	2	2.319169
9	0.8450980	2	2.443310
10	1.7160033	3	2.119850
11	1.1461280	0	2.451495
12	1.4149733	4	2.239650
13	0.4771213	4	2.279325
14	1.7853298	4	2.123067
15	1.6901961	3	2.377215
16	1.6720979	4	2.161937
17	1.7481880	2	2.316159
18	1.6989700	2	NA
19	1.4471580	2	2.200440
20	0.8450980	1	NA

- 考えるモデル：model1=lm(g1l1wr~pal2*kl2wr*ns, na.action=na.exclude, data=lafrance)

- 考える変数：　grade1l1reading（目的変数）

　l2phonaware * kinderl2reading * namingspeed, na.action=na.exclude（説明変数）

- 手順（最大のモデルを作る）【名前を model1 とする】

①統計量→モデルの適合→線形モデル

②上記の変数をそれぞれ左（目的変数）、右（説明変数）に入れる

```
Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                               -1.6932     8.8055  -0.192    0.849
l2phonaware                                2.6557     6.0993   0.435    0.666
kinderl2reading                           30.2125    42.3673   0.713    0.481
namingspeed                                0.3982     3.7303   0.107    0.916
l2phonaware:kinderl2reading              -18.7523    25.8400  -0.726    0.474
kinderl2reading:namingspeed              -13.3703    19.5694  -0.683    0.500
l2phonaware:namingspeed                   -0.5534     2.6394  -0.210    0.835
l2phonaware:kinderl2reading:namingspeed    8.3334    11.9367   0.698    0.491

Residual standard error: 0.3197 on 29 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.5974, Adjusted R-squared:  0.5002
F-statistic: 6.147 on 7 and 29 DF,  p-value: 0.0001835
```

（このモデルはメモ）

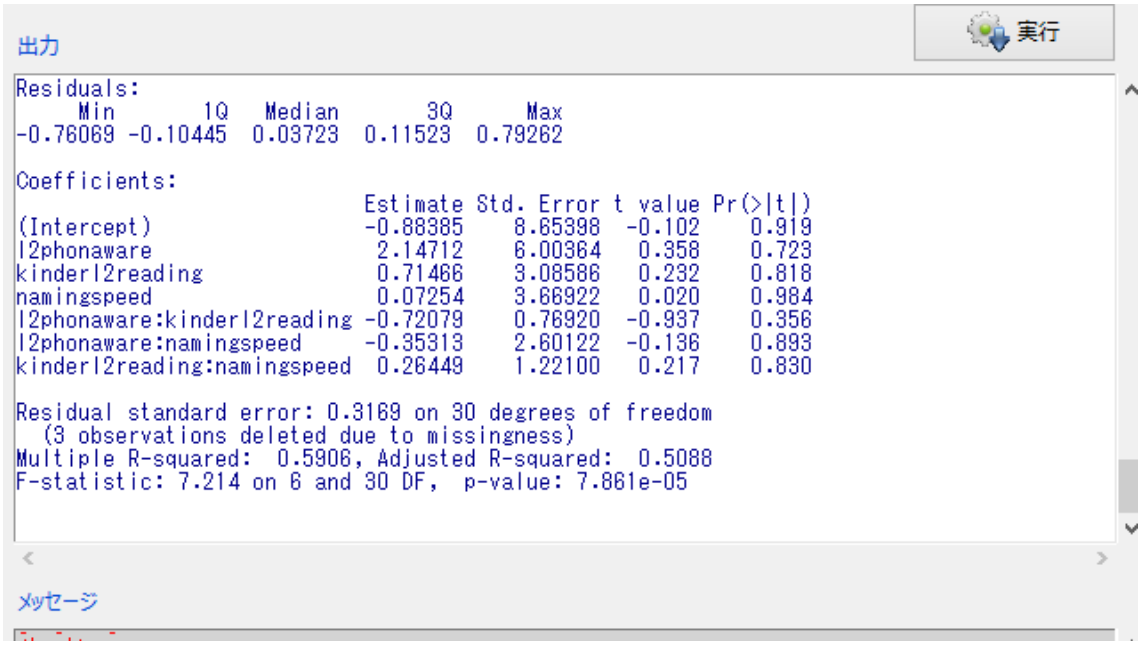
（追加）

- 先ほどのモデルから変数を引いたものを作ってみよう
- 考えるモデル：model2=update(model1,~.-pal2:kl2wr:ns, data=lafrance)
- 考える変数：

grade1l1reading（目的変数）

l2phonaware * kinderl2reading * namingspeed -l2phonaware:kinderl2reading:namingspeed（説明変数）

手順①線形モデルで上記の変数を入れる[名前を model2 とする]



（このモデルはメモ）

➢ 手順（anova 分析をしてみる）

①anova(model1,model2)※を R コマンダーに入力

※ここでの model1 と model2 は先ほど出した２つの線形モデルの名称を入れる

```
> anova(LinearModel.36,LinearModel.37)
Analysis of Variance Table

Model 1: grade111reading ~ l2phonaware * kinderl2reading * namingspeed
Model 2: grade111reading ~ l2phonaware * kinderl2reading * namingspeed -
    l2phonaware:kinderl2reading:namingspeed
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     29 2.9632
2     30 3.0130 -1   -0.0498 0.4874 0.4907
```

残差は model1 が低い

➢ 手順（AIC 分析）

①先ほどと同じ要領で AIC（model1,model2）を R コマン

ダーに入力

```
> AIC(LinearModel.36,LinearModel.37)
                df       AIC
LinearModel.36  9 29.58919
LinearModel.37  8 28.20586
```

変数の数と説明力の高さの適切さを見る

AIC が低い方が好ましいモデル

● 教科書の model2~7 は

モデルを作る→anova 分析→前のモデルからあまり影響が大きくなさそうな変数を引く→anova 分析.....の繰り返し→残差も少なく、最少の変数、高い説明力のある最適なモデルをつくる

✧ モデルの例

■

model2=update(model1,~.- pal2:kl2wr:ns, data=lafrance)

３つの変数の組み合わせを引いた例

model3=update(model2,~.- pal2:ns, data=lafrance)

影響があまりなさそうな２変数の組み合わせを引いた例

- テキスト中の model7 と Null model と比べる

今まで作ってた中で最少のモデル(model7)と null モデルを比べる

→作ったモデルが本当に適切（変数が少なく、残差も小さく、説明力が高い）なものか検証する為

- bootStepAIC

✧ 上記とは別の最適なモデルを探す手法。

✧ 上記より比較的簡単に出せる

✧ ただし、missing value を含むデータセットは機能しない

✧ Miwed-effects models(12 章)と呼ばれるもの向け一般化されている

7.5.2 Reporting the Results of Regression in R

➢ 報告の際の注意点

- データが完全か(missing を含むのか)
- 説明変数同士の相関
- 少ない変数の最適なモデル

7.6 Further Steps to Finding the Best Fit:

Overparameterization and Polynomial Regression

- 変数同士の全ての組み合わせと効果を含んだ最初のモデルを作ってみよう

①grade1l1reading　（目的変数）

nonverbalreason + I(nonverbalreason^2) + workingmemory + I(workingmemory^2) + namingspeed + I(namingspeed^2) + l2phonaware + I(l2phonaware^2) + kinderl2reading + I(kinderl2reading^2)　（説明変数）

②線形モデルに上の変数を入れる【model.3 と名前を付ける】

出力

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.493e+01  1.469e+01   1.017    0.319
nonverbalreason      6.166e-02  8.096e-02   0.762    0.453
I(nonverbalreason^2) -6.226e-04  8.179e-04  -0.761    0.453
workingmemory        1.177e-01  1.633e-01   0.721    0.478
I(workingmemory^2)   -1.180e-02  3.182e-02  -0.371    0.714
namingspeed          -1.907e+01  1.394e+01  -1.368    0.183
I(namingspeed^2)      4.225e+00  3.098e+00   1.364    0.184
l2phonaware           7.420e+00  4.882e+00   1.520    0.141
I(l2phonaware^2)     -2.196e+00  1.730e+00  -1.269    0.216
kinderl2reading       1.851e-01  3.537e-01   0.523    0.605
I(kinderl2reading^2) -2.265e-02  2.588e-01  -0.087    0.931

Residual standard error: 0.3191 on 26 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.6403, Adjusted R-squared:  0.5019
F-statistic: 4.627 on 10 and 26 DF,  p-value: 0.0007977
```

（このモデルはメモ）

- boot.stepAIC を試してみる

➢ 手順

①library(bootStepAIC)

boot.stepAIC(model3, data=lafrance)　を入力

※先ほどの線形モデルを入れること

出力

```
Summary of Bootstrapping the 'stepAIC()' procedure for

Call:
lm(formula = grade111reading ~ nonverbalreason + I(nonverbalreason^2) +
    workingmemory + I(workingmemory^2) + namingspeed + I(namingspeed^2) +
    l2phonaware + I(l2phonaware^2) + kinderl2reading + I(kinderl2reading^2),
    data = lafrance5)

Bootstrap samples: 100
Direction: backward
Penalty: 2 * df

Covariates selected
                     (%)
l2phonaware           88
I(l2phonaware^2)      72
nonverbalreason       70
I(nonverbalreason^2)  68
namingspeed           65
I(namingspeed^2)      64
workingmemory         57
```

出力

```
I(kinderl2reading^2)   45
I(workingmemory^2)     42

Coefficients Sign
                      +  (%)  -  (%)
I(namingspeed^2)      96.88    3.12
l2phonaware           95.45    4.55
workingmemory         92.98    7.02
nonverbalreason       90.00   10.00
kinderl2reading       76.36   23.64
I(kinderl2reading^2)  39.53   60.47
I(workingmemory^2)    21.43   78.57
I(l2phonaware^2)      16.67   83.33
I(nonverbalreason^2)  10.29   89.71
namingspeed            6.15   93.85

Stat Significance
                        (%)
l2phonaware           81.82
I(nonverbalreason^2)  69.12
workingmemory         68.42
nonverbalreason       67.14
```

※このあと少し長いので割愛

- 最適なモデルが分かる

➢ 10two-way interactions をしてみる

- ２つの変数の組み合わせを含む１０この目的変数

➢ 使用するもの

①用いる変数

grade1l1reading（目的変数）

nonverbalreason + workingmemory + namingspeed + l2phonaware + kinderl2reading + l2phonaware:kinderl2reading + nonverbalreason:namingspeed + namingspeed:l2phonaware + nonverbalreason:workingmemory + namingspeed:kinderl2reading （説明変数）

②boot.stepAIC のコマンド

➢ 手順

➢ ①線形モデルに上記の変数を入力

```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     9.658051  11.585398   0.834   0.4121
nonverbalreason                -0.273631   0.212813  -1.286   0.2099
workingmemory                  -0.872738   0.471316  -1.852   0.0755 .
namingspeed                    -3.527696   4.834418  -0.730   0.4721
l2phonaware                     3.584681   6.161256   0.582   0.5657
kinderl2reading                 2.542555   3.265657   0.779   0.4433
l2phonaware:kinderl2reading    -2.019912   0.987599  -2.045   0.0511 .
nonverbalreason:namingspeed     0.096623   0.088331   1.094   0.2840
namingspeed:l2phonaware        -0.950086   2.655631  -0.358   0.7234
nonverbalreason:workingmemory   0.019205   0.009795   1.961   0.0607 .
namingspeed:kinderl2reading     0.384269   1.287310   0.299   0.7677
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3141 on 26 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.6516, Adjusted R-squared:  0.5175
F-statistic: 4.862 on 10 and 26 DF,  p-value: 0.0005602
```

②library(bootStepAIC)

boot.stepAIC(model1, data=lafrance)

※図は長いので割愛

➢ このあとの流れ model3~model6 まで

今用いたモデルの目的変数を変えたり、引いたりする→bootStepAIC で出力して分析→前のモデルの目的変数を変えたり、引いたり……の繰り返し

そして最後に出たものを保持

➢ Model7,8

Three-way(NR:WM:NS のような形)でやって bootstep.で分析

model7= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+

NS:PAL2:KL2WR+NR:WM:PAL2+NR:NS:KL2WR+NR:

WM:NS+

WM:PAL2:KL2WR, data=lafrance)

- Model9

5 four-way と 1five-way を含んだ 11 の目的変数で構成されたもの

model9= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+

NR:WM:NS:PAL2 + NR:WM:NS:KL2WR +

NR:WM:PAL2:KL2WR +

NR:NS:PAL2:KL2WR + WM:NS:PAL2:KL2WR +

NR:WM:NS:PAL2:KL2WR,

data=lafrance)

- Model10,11

２乗変数 1(NS^2)を含むモデル

model10=lm(G1L1WR~ NR + WM + NS + I(NS^2) +

PAL2 + I(PAL2^2) + KL2WR+

WM:KL2WR +NR:KL2WR +NR:PAL2+ #two-way

interactions

NR:WM:NS +WM:PAL2:KL2WR + #three-way interactions NR:WM:NS:KL2WR + WM:NS:PAL2:KL2WR, #four-way interactions

data=lafrance)

- 最も説明力のある説明変数一つで構成されたモデルを作る(model12 と名付ける)

（追加）

①grade1l1reading(目的変数)

②l2phonaware（説明変数）

- 手順

①線形モデルに上記の変数を入力

（このモデルの名前もメモ）

✧ Model11 と model12 を比べて model11 は説明力もあり残差も少ない

7.7 Examing Regression Assumptions

- 回帰を分析するプロットを作ろう！

- 使用するもの

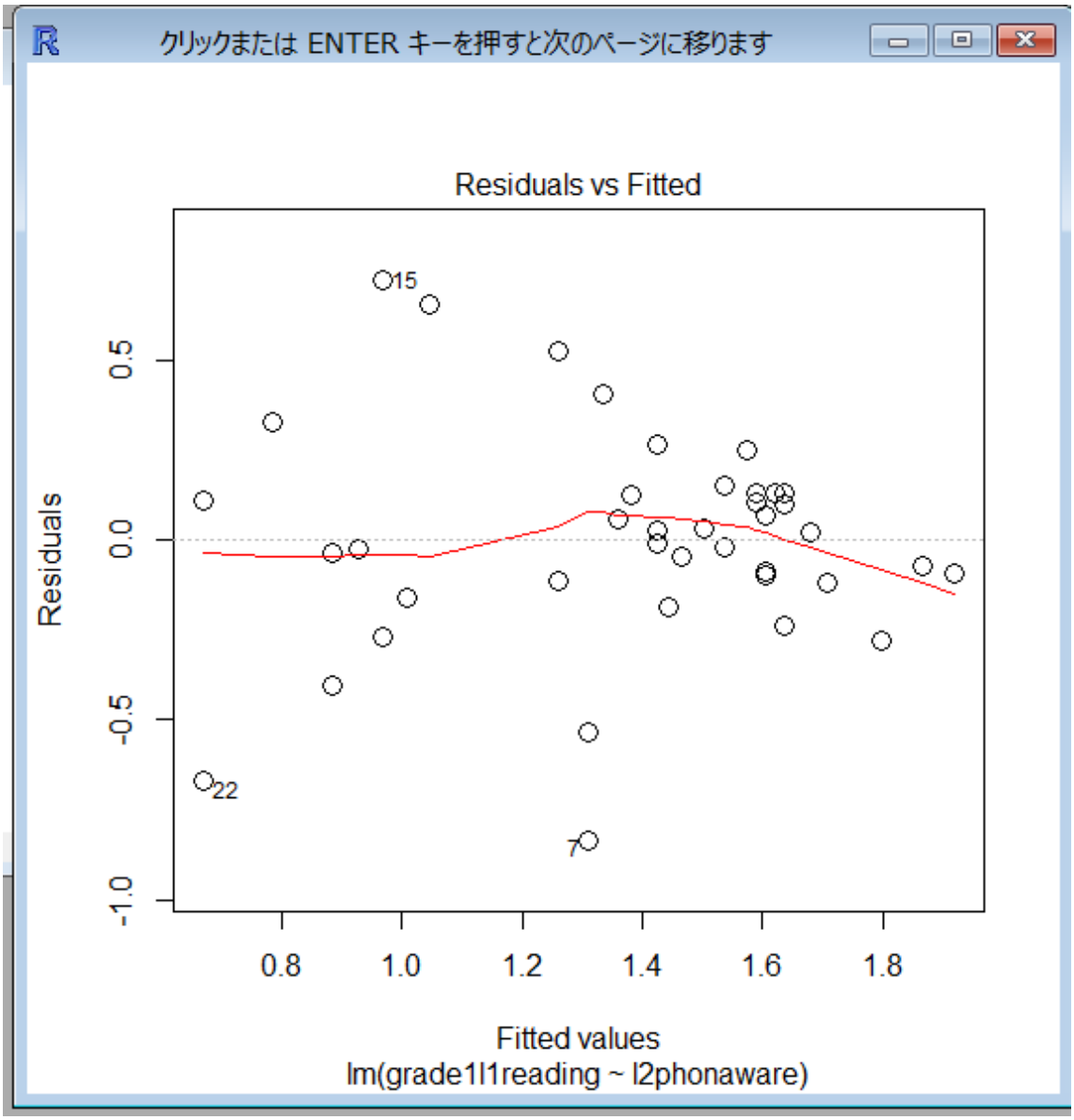
①ファイル：SPSS ファイル LafranceGottardo.sav、名前は lafrance

②コマンド：plot(model12,cex=1.5)

※先ほど使ったモデル

- 手順

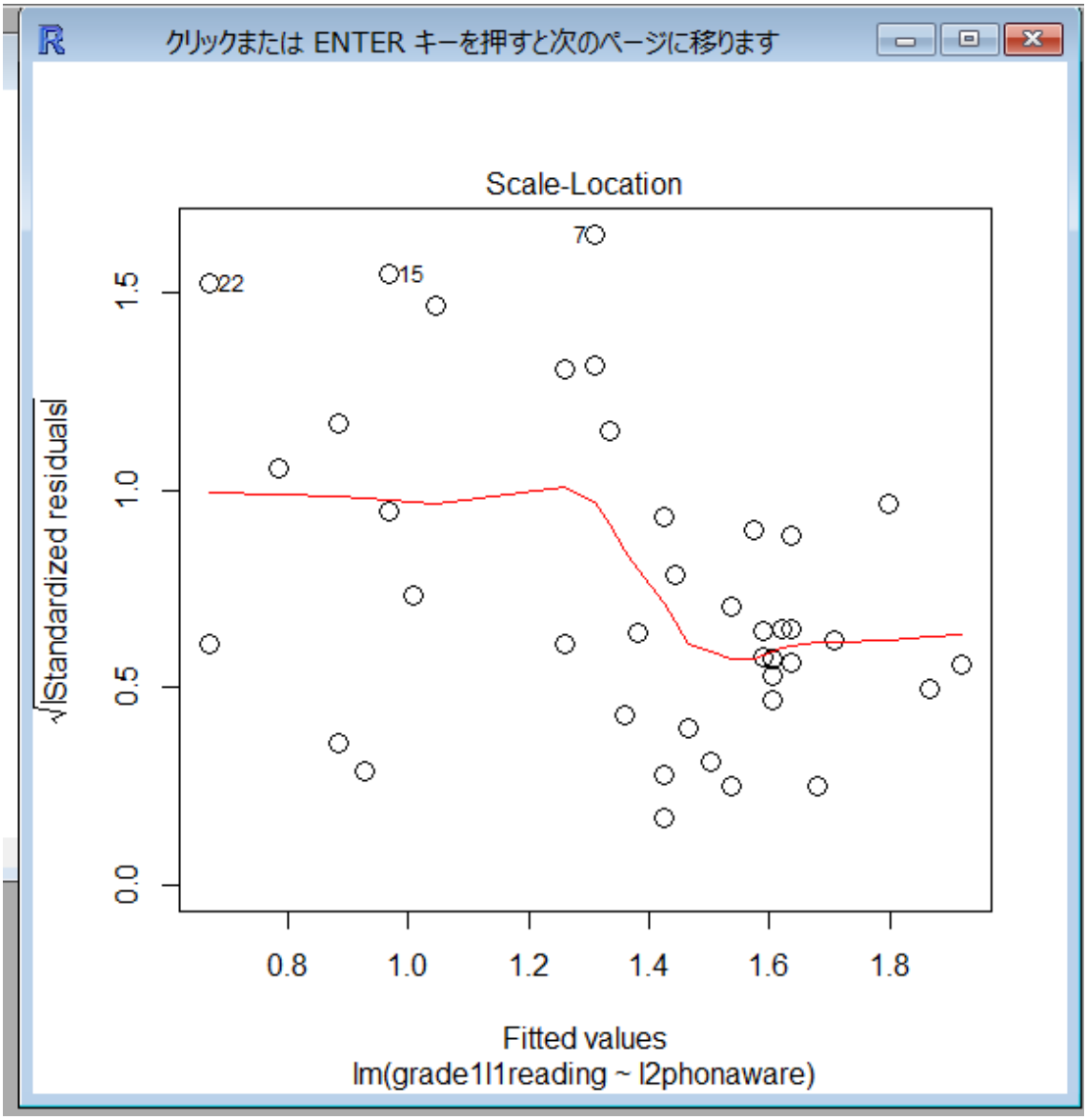
①R コマンドに上記のコマンドを入れる

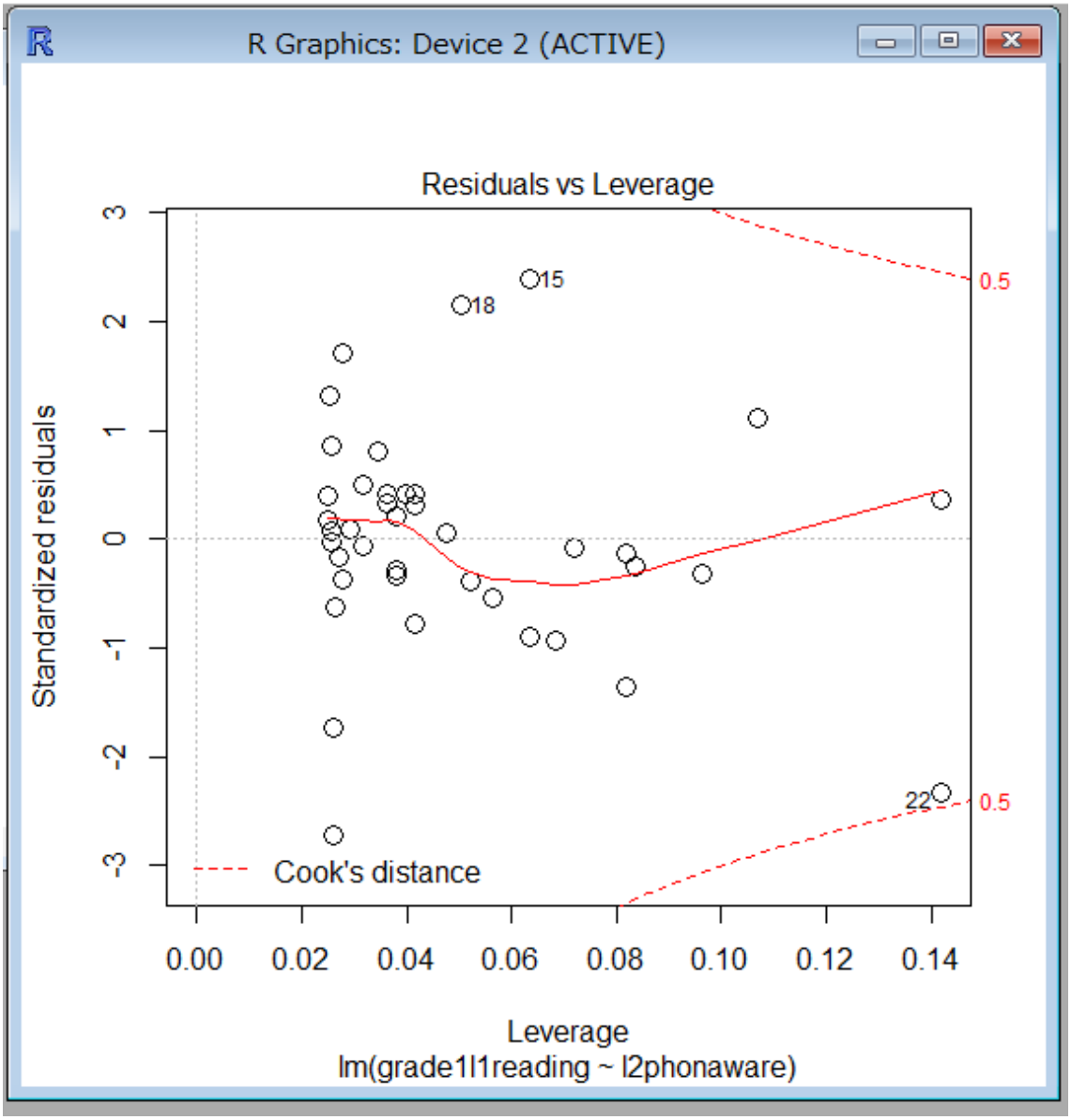


- 変数があるところでは集中している
- 不等分散性を示している

→変数の等質性が失われている

✧ こういった問題はいくつかの変数を移行することで解決できる


クリックまたは ENTER キーを押すと次のページに移ります
Scale-Location
7
22
15
1.5
1.0
0.5
0.0
√|Standardized residuals|
0.8
1.0
1.2
1.4
1.6
1.8
Fitted values
lm(grade1l1reading ~ l2phonaware)



- はずれ値を除外してみる
- 手順

①線形モデルの目的変数に subset=(lafrance !=22)を入力

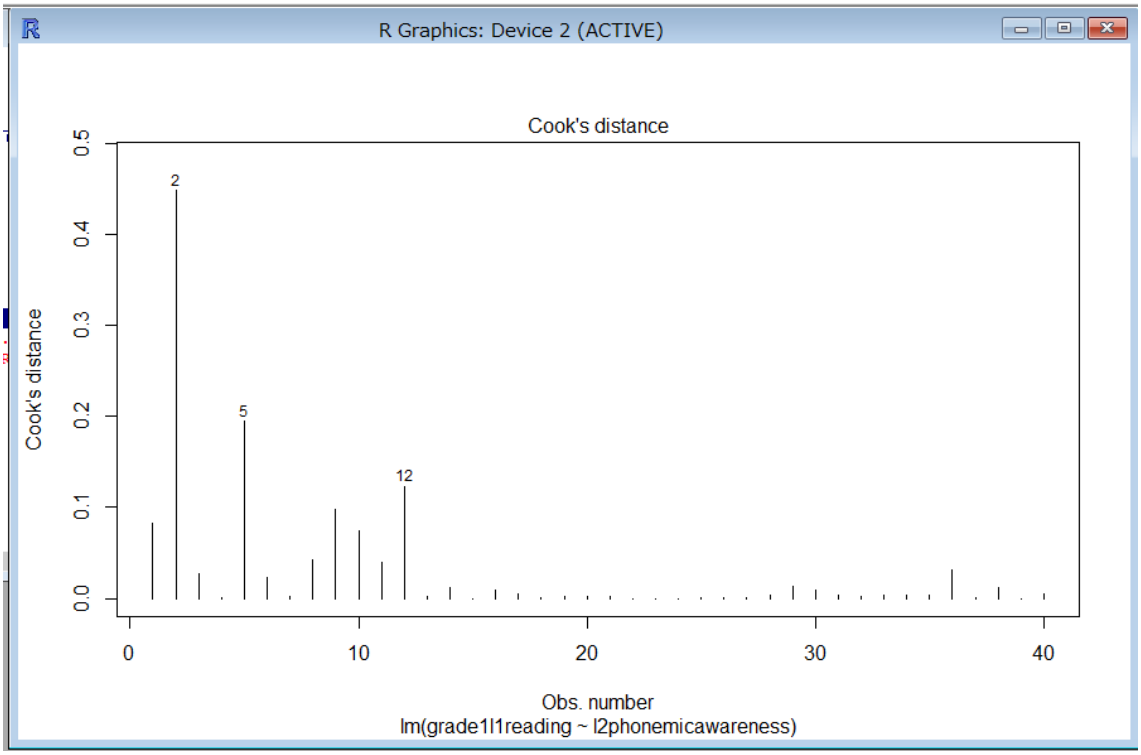
出力

```
Call:
lm(formula = grade111reading ~ l2phonaware, data = lafrance5,
    subset = (lafrance != 22))

Residuals:
     Min       1Q   Median       3Q      Max
-0.83387 -0.11485  0.00512  0.12734  0.72097

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.9713     0.3596  -2.701   0.0103 *
l2phonaware   1.5771     0.2398   6.577 9.24e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3113 on 38 degrees of freedom
  (320 observations deleted due to missingness)
Multiple R-squared:  0.5323, Adjusted R-squared:   0.52
F-statistic: 43.26 on 1 and 38 DF,  p-value: 9.24e-08
```

- 外れ値の影響が見つかるプロットを作ろう
- 使用するもの
- コマンド

plot(model12,which=4)

①上のコマンドを R コマンダーに入力



- VIF で見てみる
- 使用するもの

①パッケージ:VIF パッケージをインストールして読み込み

②コマンド

model.vif=lm(grade1l1reading ~ nonverbalreasoning + kinderl2reading + namingspeed + workingmemory + l2phonemicawareness+ ,data=lafrance)

vif(model.vif)

➢ 手順

①上記のコマンドを R コンソールに入力

```
> model.vif=lm(grade111reading ~ nonverbalreasoning + kinder12reading + namingspeed + workingmemory + l2phonemicawareness
+ ,data=lafrance)
> vif(model.vif)
 nonverbalreasoning    kinder12reading        namingspeed       workingmemory
           1.400075           2.260397           2.151198           1.863077
l2phonemicawareness
           3.246957
>
```

➢ MASS でプロットしてみよう

➢ 使用するもの

①パッケージ： MASS、 インストールして読み込み

②コマンド

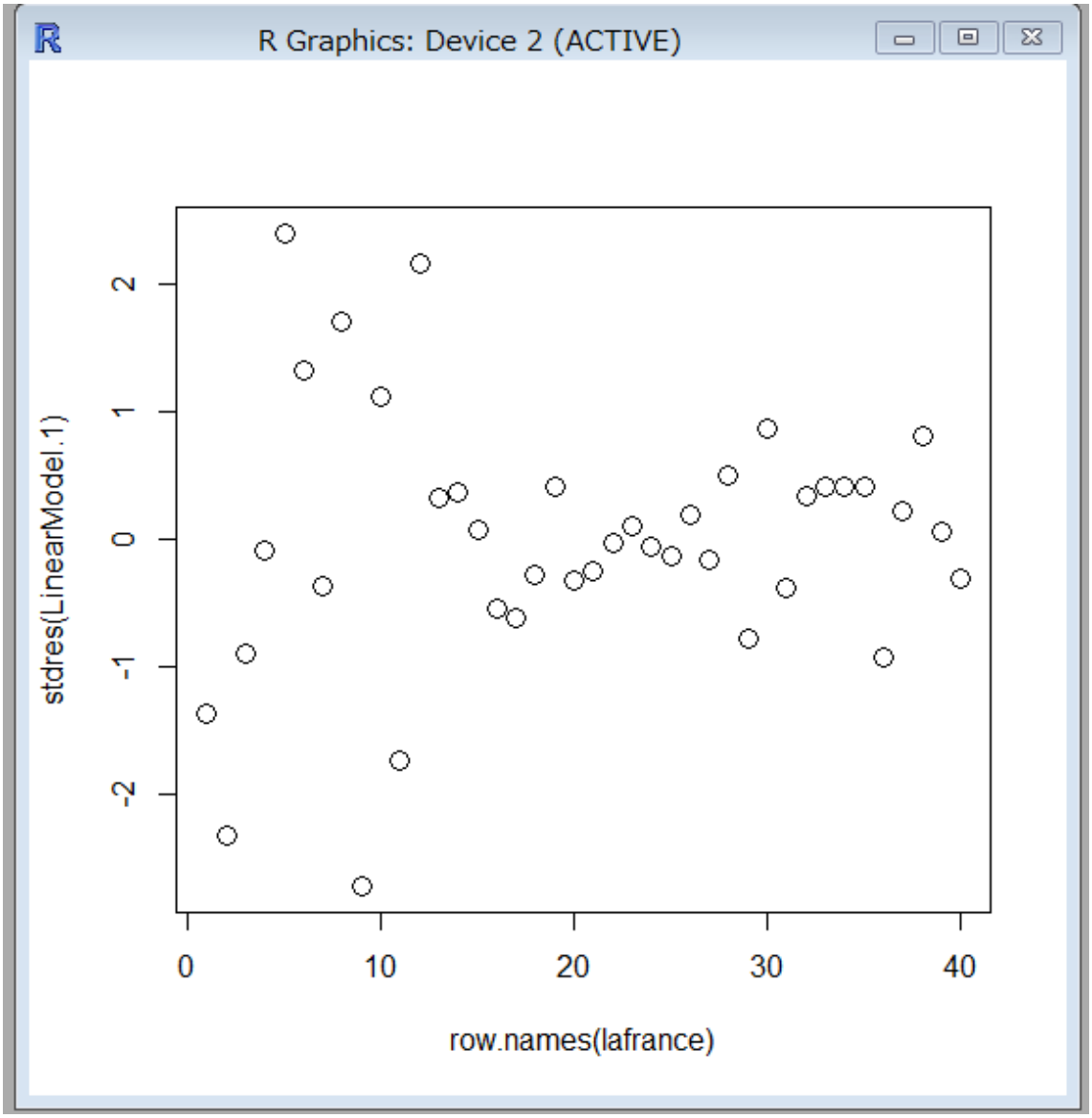
library(MASS)

plot(row.names(lafrance),stdres(model12),cex=1.5)

※model12 は自分の作ったモデルの名前を入れる

➢ 手順

①上記のコマンドを入れる



7.9 Robust Regression

- 回帰分析にはしばしば外れ値が存在する
- 外れ値により第１種の誤り（有意ではないのに有意としてしまう)、第２種の誤り（有意であるのに有意と判断しない)

✧ これらを避けるべく、外れ値を考慮した robust model をつくる必要がある

7.9.1 Visualizing Robust Regression

➢ Robustbase を使ってみよう

➢ 使用するもの

①ファイル：Lafrance3.csv、名前は Lafrance3

※データ→データをインポートする→テキストファイルまたはクリップボードから、を選ぶこと

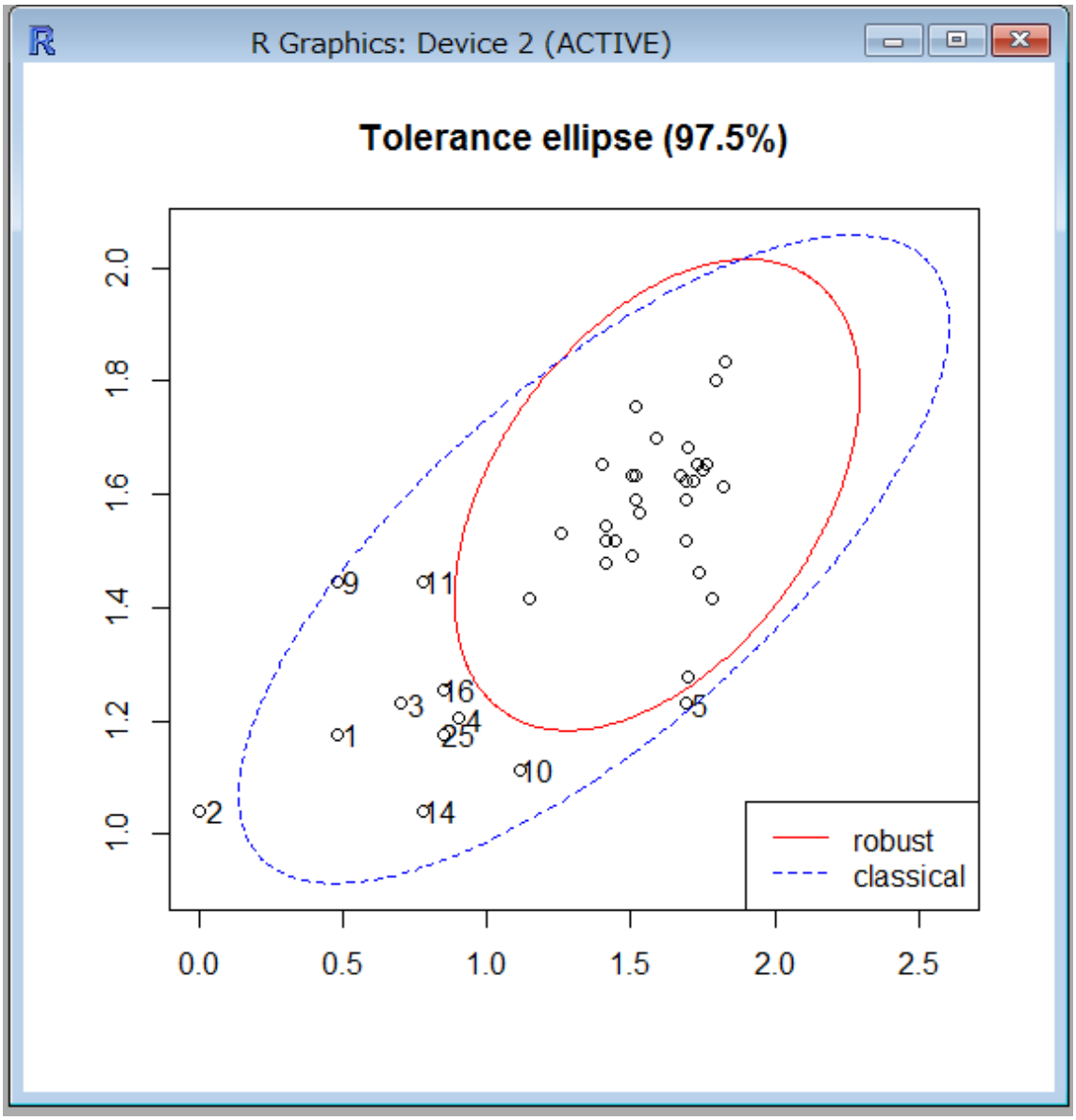
②パッケージ: robustbase、インストールして読み込み

③コマンド

```
library(robustbase)
covPlot(cbind(Lafrance3$G1L1WR,Lafrance3$PAL2),whi
ch="tolEllipsePlot",
classic=T, cor=T)
```

➢ 手順

①上記コマンドを R コンソールに入力（今回自分はウィンドウのみしか出ず）(lafrance だと出た)



- robust 回帰では通常の回帰と robust 回帰とで比べた値や図が出てくる

7.9.2 Robust Regression Methods

➢ Robust の回帰モデルをやってみる

※教科書では「Lafrance3」ファイルを使っているが、自分は出力できなかったので今回は「lafrance」ファイルを使用

➢ 用いるもの

● コマンド：

```
library(robust)
lafrance.fit=fit.models(list(Robust="lmRob",      LS="lm"),
formula=grade1l1reading~l2phonemicawareness*kinderl
2reading*namingspeed ,data=lafrance)
summary(lafrance.fit)
```

※このコマンドは後の plot の図を出すためにも必要なので必ず入力して読み込ませる

➢ 手順

①上記コマンドを R コマンダーもしくは R コンソールに入力

```
Calls:
Robust: lmRob(formula = grade111reading ~ l2phonemicawareness * kinderl2reading *
    namingspeed, data = lafrance)
    LS: lm(formula = grade111reading ~ l2phonemicawareness * kinderl2reading *
    namingspeed, data = lafrance)

Residual Statistics:
            Min       1Q  Median      3Q     Max
Robust: -0.7475 -0.08554 0.03470 0.11374 0.7788
    LS: -0.8093 -0.10880 0.02042 0.08399 0.7348

Coefficients:
                                                        Estimate Std. Error t value Pr(>|t|)
                                     (Intercept): Robust:  -9.0600     6.7777  -1.337   0.1907
                                                      LS:  -2.2490     8.6498  -0.260   0.7965

                                l2phonemicawareness: Robust:   6.9767     4.7005   1.484   0.1475
                                                      LS:   2.1775     6.0338   0.361   0.7206

                                    kinderl2reading: Robust:  78.1462    35.2734   2.215   0.0340 *
                                                      LS:  36.6445    43.3275   0.846   0.4040

                                       namingspeed: Robust:   3.4204     2.8953   1.181   0.2462
                                                      LS:   0.5044     3.6800   0.137   0.8918

            l2phonemicawareness:kinderl2reading: Robust: -47.7295    21.5170  -2.218   0.0338 *
                                                      LS: -21.9682    26.4649  -0.830   0.4126

                l2phonemicawareness:namingspeed: Robust:  -2.3056     2.0464  -1.127   0.2683
                                                      LS:  -0.2304     2.6147  -0.088   0.9303

                    kinderl2reading:namingspeed: Robust: -35.5772    16.3354  -2.178   0.0369 *
                                                      LS: -16.3227    20.0155  -0.816   0.4208

l2phonemicawareness:kinderl2reading:namingspeed: Robust:  21.7474     9.9649   2.182   0.0365 *
                                                      LS:   9.7982    12.2278   0.801   0.4289
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- コマンドを入力すると通常の lm と robust の比較をそれぞれ数値で見ることができる
- R^2値は基本 robust の方が低いが残差が少ない

➢ Robust のみのモデルでやる

先ほどの普通の回帰モデルと同じように様々な組み合わせを考えてみる

➢ 使用するコマンド

m1.robust=lmRob(grade1l1reading~l2phonemicawarenes

s*kinderl2reading*namingspeed,data=lafrance)

summary(m1.robust)

➢ 手順

①上記コマンドを R コマンダーに入力

出力

```
Call:
lmRob(formula = grade1l1reading ~ l2phonemicawareness * kinderl2reading *
    namingspeed, data = lafrance)

Residuals:
      Min        1Q    Median        3Q       Max
-0.74750 -0.08554  0.03470  0.11374  0.77884

Coefficients:
                                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                                     -9.060      6.778  -1.337   0.1907
l2phonemicawareness                              6.977      4.701   1.484   0.1475
kinderl2reading                                 78.146     35.273   2.215   0.0340 *
namingspeed                                      3.420      2.895   1.181   0.2462
l2phonemicawareness:kinderl2reading            -47.730     21.517  -2.218   0.0338 *
l2phonemicawareness:namingspeed                 -2.306      2.046  -1.127   0.2683
kinderl2reading:namingspeed                    -35.577     16.335  -2.178   0.0369 *
l2phonemicawareness:kinderl2reading:namingspeed  21.747      9.965   2.182   0.0365 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

このような形で 2way なども試していく

➢ bootstepAIC は使えないので、robust final prediction error(RFPE)という値を見てみる。これが低いほど良い

➢ 使用するもの

● コマンド：

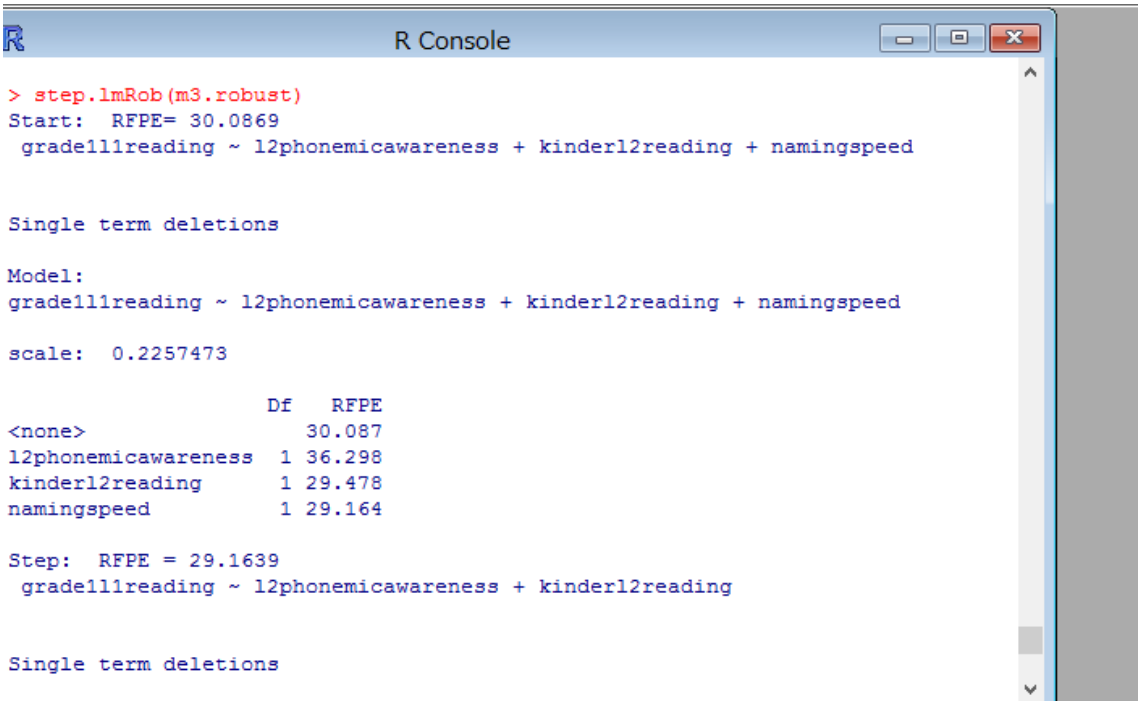
m3.robust=lmRob(grade1l1reading~l2phonemicawarenes

s+kinderl2reading+namingspeed,data=lafrance)

step.lmRob(m3.robust)

- 手順

①上記コマンドを R コンソールに入力



- 4 つのプロット図(LS と Robust の比較)

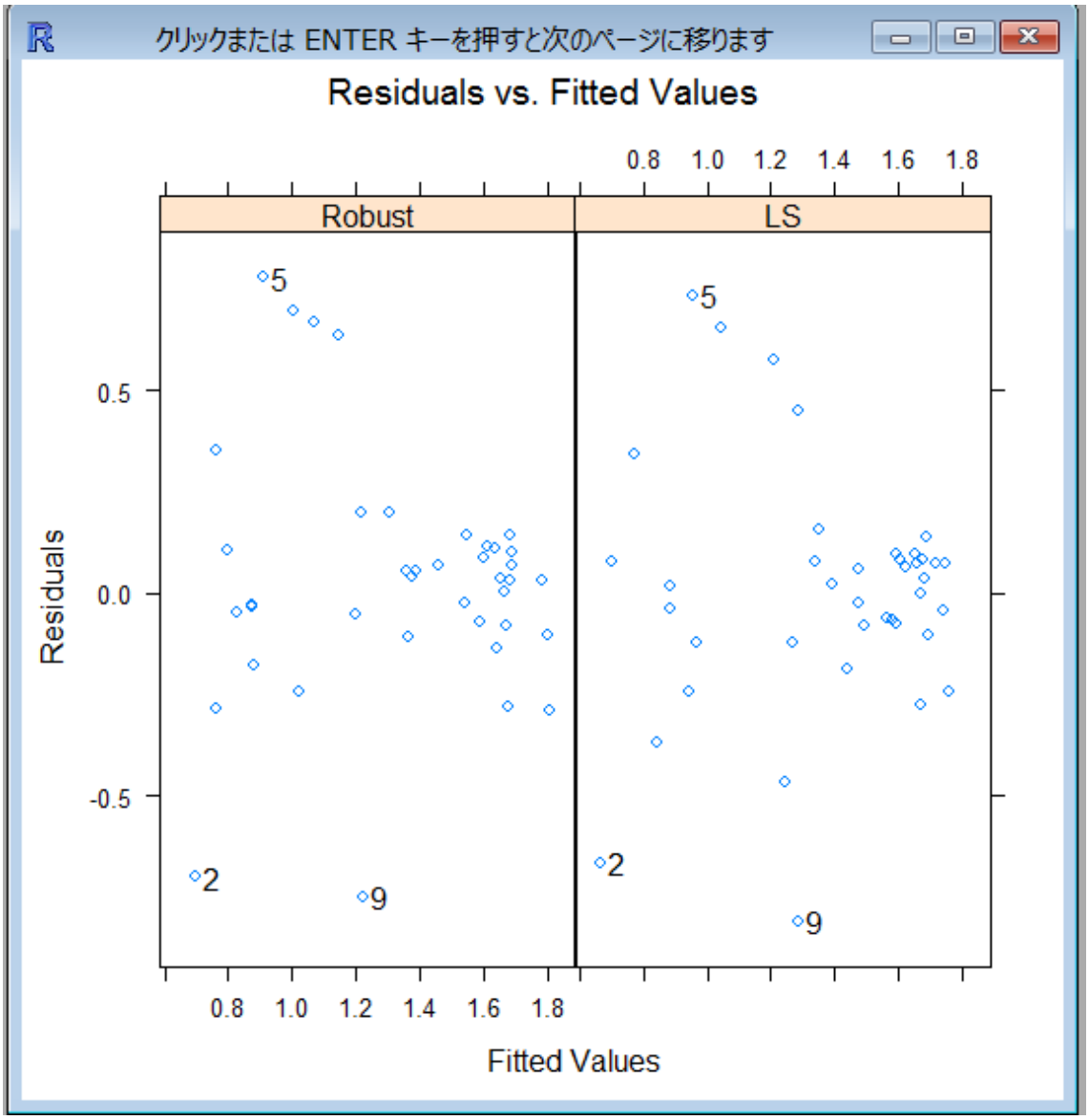
- 使用するもの

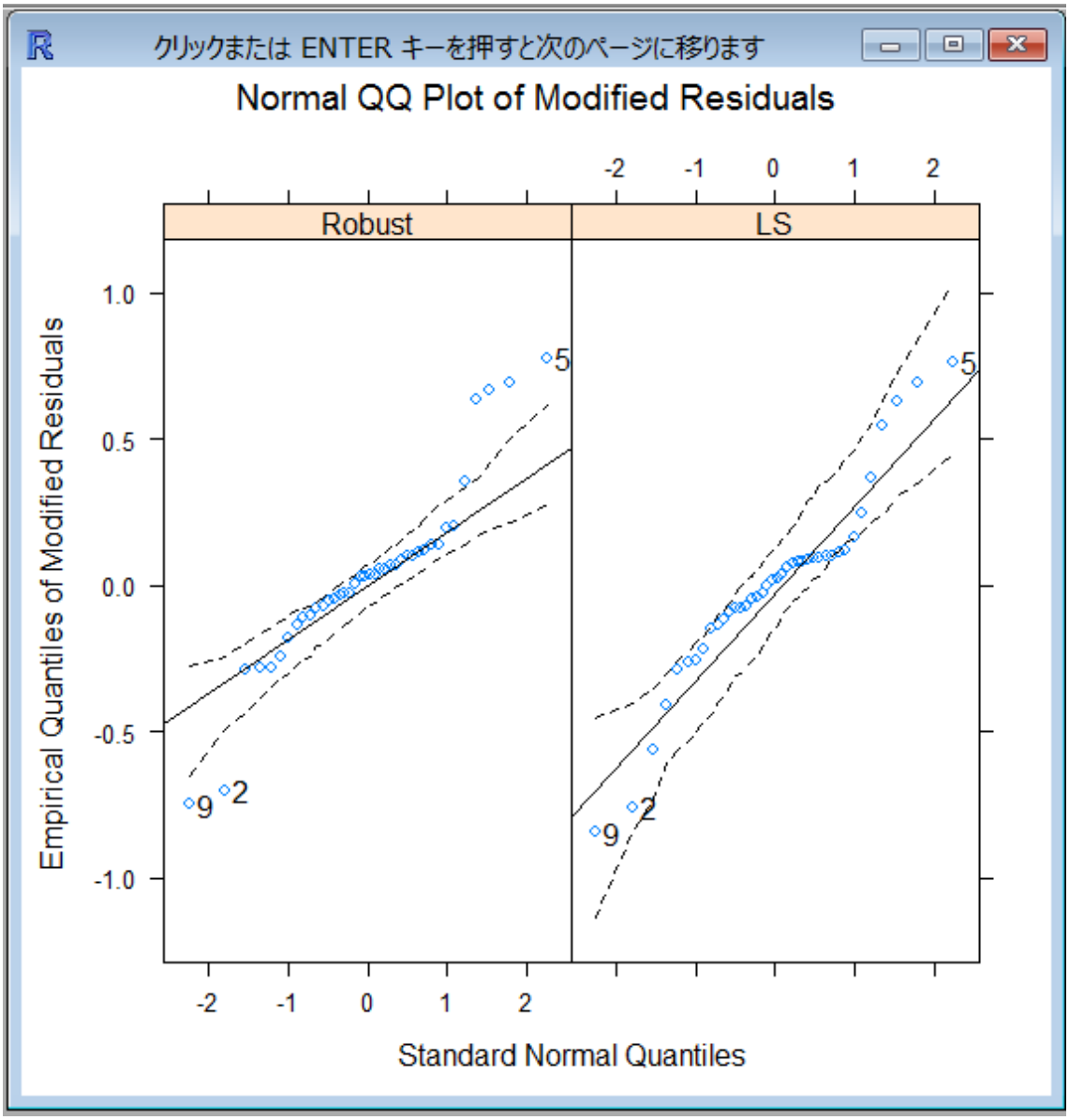
コマンド

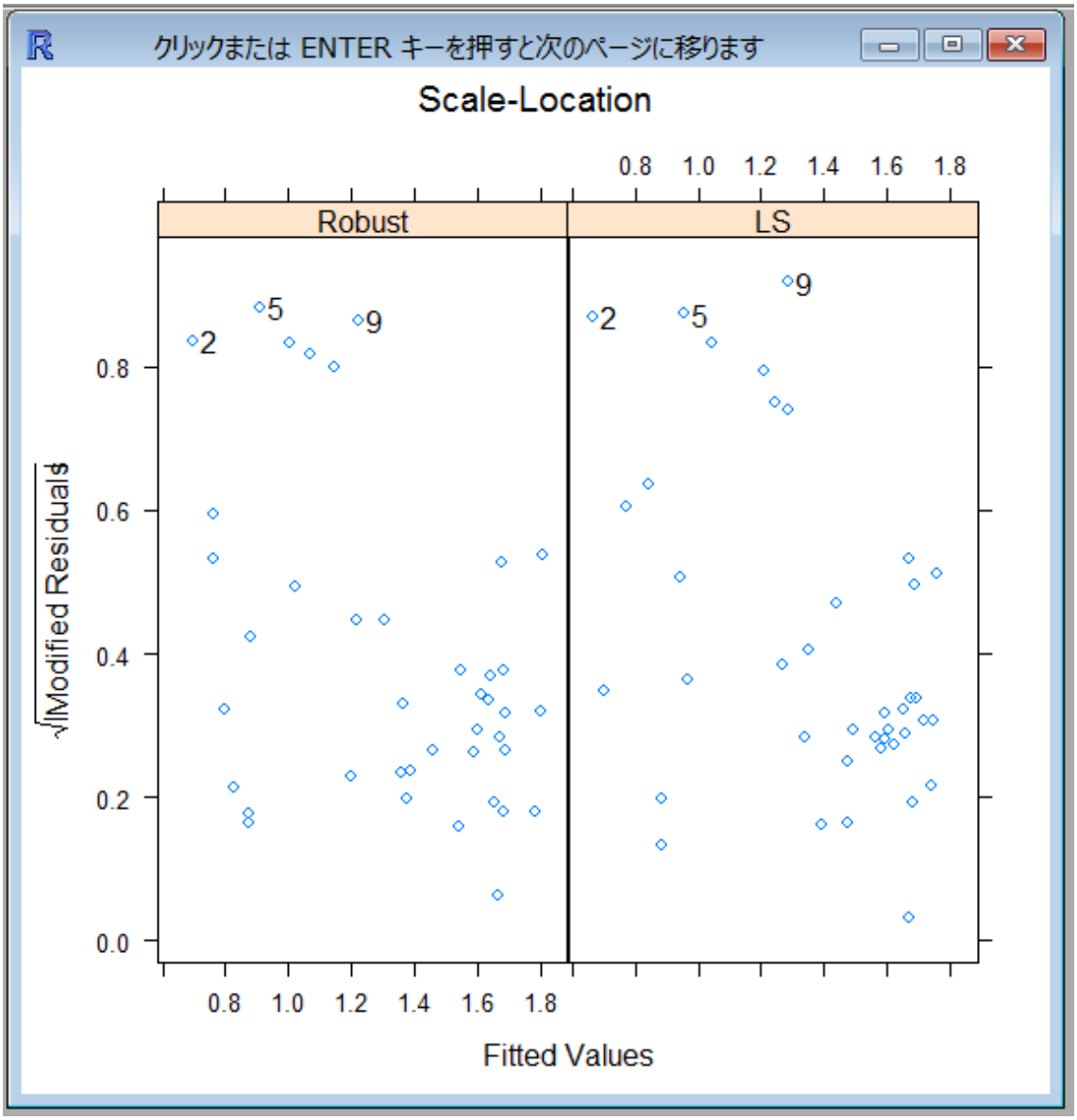
plot(lafrance.fit)

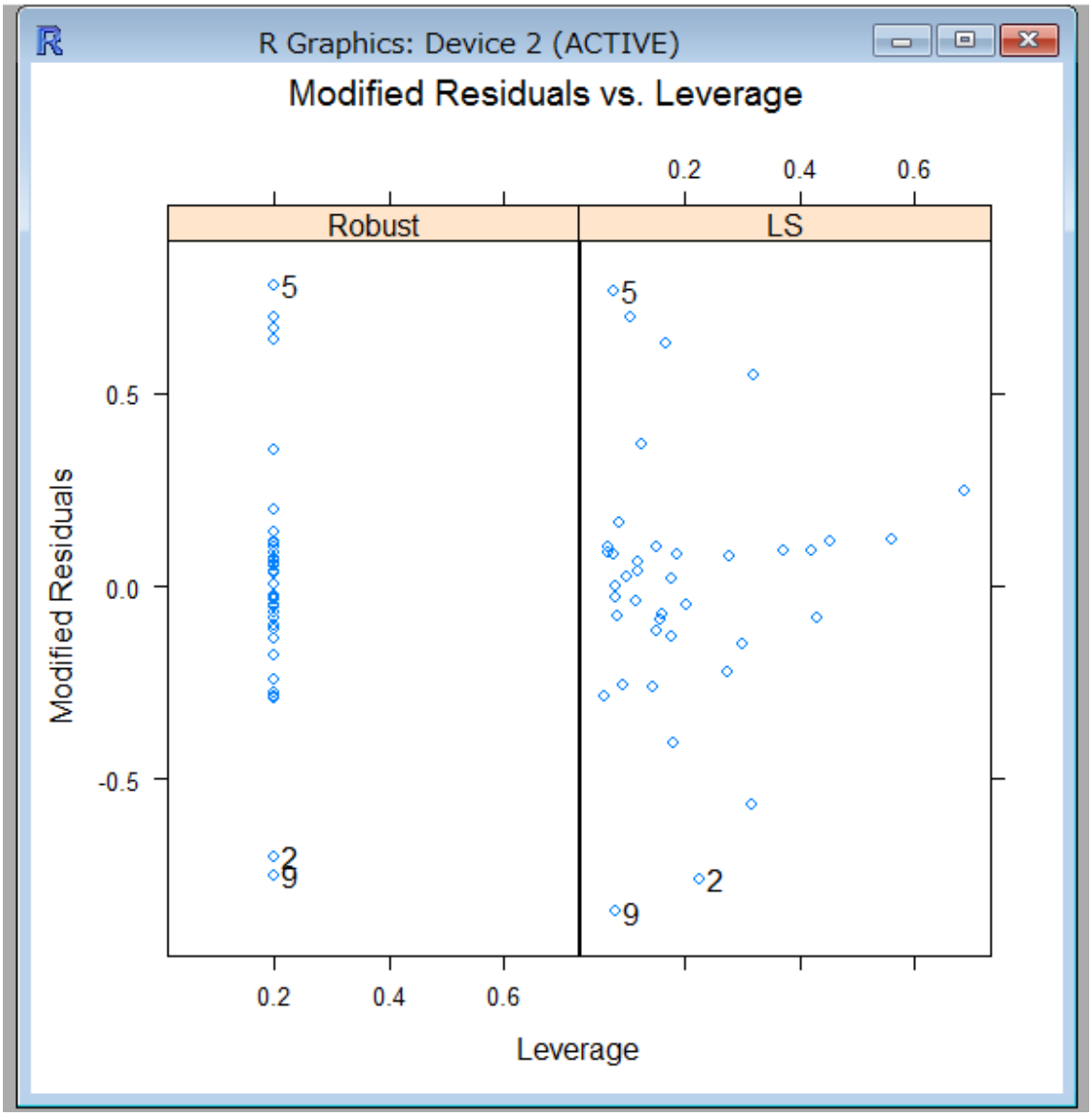
- 手順

①上記コマンドを R コマンダーに入力


クリックまたは ENTER キーを押すと次のページに移ります
Residuals vs. Fitted Values
Robust
LS
Residuals
Fitted Values
0.8
1.0
1.2
1.4
1.6
1.8
0.5
0.0
-0.5
5
2
9


クリックまたは ENTER キーを押すと次のページに移ります
Normal QQ Plot of Modified Residuals
Robust
LS
Empirical Quantiles of Modified Residuals
Standard Normal Quantiles
-2
-1
0
1
2
1.0
0.5
0.0
-0.5
-1.0
5
9
2


クリックまたは ENTER キーを押すと次のページに移ります
Scale-Location
Robust
LS
√|Modified Residuals|
Fitted Values
0.0
0.2
0.4
0.6
0.8
0.8
1.0
1.2
1.4
1.6
1.8
2
5
9



※ただし、これらの図は一部教科書のものと異なる

- Normal QQ Plot of Residuals

✧ 両方とほぼ線に沿っているのであまり差がない

✧ ただ問題は LS は点を厳密に制限していること

Robust は上記よりも多くの値を含めている

- Kernel　Density of Residuals

✧ ドットの点は影響点と x 外れ値を考えられる

✧ 山の頂上の点は外れ値

- Standardized Residuals vs Robust Distances

✧ X 外れ値が 3 つほどある

✧ しかし、robust 分析には影響しない

- Residuals vs Fitted Values

✧ 不等分散性が robust にもまだ見られる

✧ 一部の点は大きく異なっている