

Rによる線形回帰分析

1 最小二乗法による推定

1.1 データ cars の読み込み

まず使用するデータを読み込む。R のパッケージ `datasets` にあるデータ `cars` を使用する。

R のパッケージを読み込むには、一番上の段にある

「パッケージ」→「パッケージのインストール」

とクリックしていくと、ミラーサイトを選ぶ画面になる(どこのサイトからインストールをするかを選ぶ画面)。これは適当に近そうなところ(日本のどこか)を選んでおけば問題ない(「Japan(Tokyo)」とか「Japan(Yonezawa)」とか)、出てきたパッケージの一覧から「`datasets`」を選んで「OK」をクリックする。これでパッケージ `datasets` をインストールできた。

パッケージ `datasets` のデータを実際に使うには、さらに以下の操作が必要となる。

「パッケージ」→「パッケージの読み込み」をクリックする。するとインストールしたパッケージの一覧が表示されるのでそこから `datasets` を選んで「OK」をクリックする。もしくは R console に

```
> library(datasets)
```

と入力する。これでパッケージ `datasets` にあるデータを使用できるようになる。以下では `cars` というデータを使用する。

データ「`cars`」を確認するために

```
> cars
```

と入力すると

```
  speed dist
1     4     2
2     4    10
3     7     4
4     7    22
      ⋮
```

と(すべての)データが表示される。これはブレーキをかけた時の車のスピード(`speed`)とその場合の静止までに走行した距離(`dist`)のデータである。最初の3行だけ表示するには

```
> head(cars,3)
```

と入力する。

さきほどのデータ `cars` に対して、 speed_i を i 番目の車の `speed`、 dist_i を i 番目の車の `dist` とし、 x_i と y_i をそれぞれの自然対数を取ったもの、すなわち、 $x_i = \log(\text{speed}_i)$ 、 $y_i = \log(\text{dist}_i)$ とする。これらのデータに対して、 y_i を x_i に回帰した単回帰モデル：

$$y_i = \alpha + \beta x_i + e_i$$

を推定する。ここで e_i は線形回帰モデルの誤差項である。

1.2 最小二乗法による単回帰モデルの推定

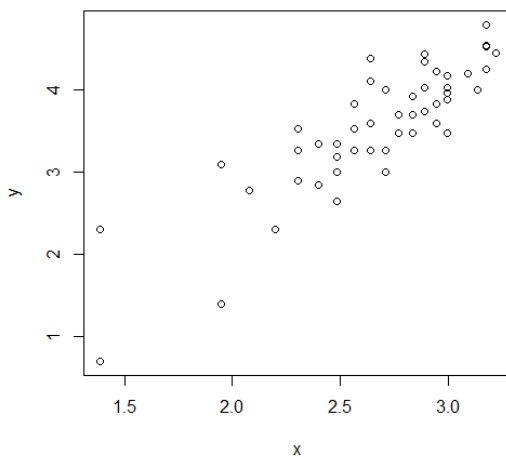
R の `lm` 関数を用いて推定する。まず、`speed` と `dist` のデータの対数変換する。以下のように入力する(R で対数変換は関数 `log` を用いる)。

```
> x=log(cars$speed)
> y=log(cars$dist)
```

データ `x` と `y` の散布図を描くには

```
> plot(x,y)
```

と入力する。以下の図が表示される。



これらのデータに対して、先ほどの線形回帰モデルを最小二乗法で推定するには、以下のように入力する。

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

推定結果を見るには、`summary` 関数を用いる。以下のように入力する。

```
> summary(result)
```

すると結果として以下のように表示される。

```
Call:
lm(formula = y ~ x)

Residuals:
    Min       1Q   Median       3Q      Max
-1.00215 -0.24578 -0.02898  0.20717  0.88289

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.7297      0.3758  -1.941   0.0581 .
x             1.6024      0.1395  11.484 2.26e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4053 on 48 degrees of freedom
Multiple R-squared:  0.7331,    Adjusted R-squared:  0.7276
F-statistic: 131.9 on 1 and 48 DF,  p-value: 2.259e-15
```

これより、 α の推定値は -0.7297 、 β の推定値は 1.6024 であることがわかる。P 値を見ると、 β は 1%有意水準で有意だが(0 と有意に異なる)、 α については、10%では有意だが 5%有意水準では有意になっていない(0 と有意に異なっていない)ことがわかる。

なお、先ほどは speed と dist のデータを対数変換して x と y という変数を新たに作り、それらに対して最小二乗推定を行ったが、これらを別々ではなく一度にやるには

```
> result2=lm(log(dist)~log(speed), cars)
```

と入力すればよい、結果は

```
> summary(result2)
```

によって確認できる。同じ推定結果になっていることが確認できる。

また、先ほどは切片を含めたモデルを推定したが、切片は有意でなかったので、ためしに切片を含めないモデル:

$$y_i = \beta x_i + e_i$$

を推定を推定してみる。この切片を含めないモデルを推定するには

```
> result3 = lm(y~x-1)
```

と入力する(-1 が追加で入力してあることに注意)。結果は

```
> summary(result3)
Call:
lm(formula = y ~ x - 1)

Residuals:
    Min       1Q   Median       3Q      Max
-1.21083 -0.22501  0.01129  0.25636  0.85978

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
x  1.33466    0.02187    61.02  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4166 on 49 degrees of freedom
Multiple R-squared:  0.987,    Adjusted R-squared:  0.9867
F-statistic: 3724 on 1 and 49 DF, p-value: < 2.2e-16
```

となる。今回の β の推定は 1.33466 である。若干値が変わっているが、この場合も 1% 有意水準で有意である。

1.3 データ LifeCycleSavings の読み込み

パッケージ `dataset` にある `LifeCycleSavings` というデータを用いる。このデータの最初の 5 行をみると(すべてのデータを見るには単に `LifeCylceSavings` と入力すればよい)

```
> head(LifeCycleSavings, 5)
      sr  pop15 pop75  dpi ddpi
Australia 11.43 29.35  2.87 2329.68 2.87
Austria   12.07 23.32  4.41 1507.99 3.93
Belgium    13.17 23.80  4.43 2108.47 3.82
Bolivia     5.75 41.89  1.67  189.13 0.22
Brazil     12.88 42.19  0.83  728.47 4.56
```

となる。これはそれぞれの国についての

```
sr:      個人所得
pop15:   15 歳以下の人口の割合
pop75:   75 歳以上の人口の割合
dpi:     一人当たり可処分所得
ddpi:    dpi の成長率
```

という変数の 1960 年から 1970 年の平均のデータである(1 列目は国名)。

1.4 最小二乗法による重回帰モデルの推定

このデータに対して、個人所得を他のデータに線形回帰させた

$$sr_i = \alpha + \beta_1 \text{pop15}_i + \beta_2 \text{pop75}_i + \beta_3 \text{dpi}_i + \beta_4 \text{ddpi}_i + e_i$$

という線形回帰モデルを考える。ここで sr_i は i 番目の国の sr である。他の変数についても同様に定義する。また e_i は誤差項である。まずはこのモデルを最小二乗法で推定してみる。

上記の線形回帰モデル(重回帰モデル)を推定するには以下のように入力する

```
> sr_result = lm(sr~pop15+pop75+dpi+ddpi,LifeCycleSavings)
```

推定結果は以下のようになる

```
> summary(sr_result)
```

Call:

```
lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.2422	-2.6857	-0.2488	2.4280	9.7509

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	28.5660865	7.3545161	3.884	0.000334	***
pop15	-0.4611931	0.1446422	-3.189	0.002603	**
pop75	-1.6914977	1.0835989	-1.561	0.125530	
dpi	-0.0003369	0.0009311	-0.362	0.719173	
ddpi	0.4096949	0.1961971	2.088	0.042471	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.803 on 45 degrees of freedom

Multiple R-squared: 0.3385, Adjusted R-squared: 0.2797

F-statistic: 5.756 on 4 and 45 DF, p-value: 0.0007904

切片と pop15 が 1%有意水準で有意、ddpi が 5%有意水準で有意となっている。

2 リッジおよびラッソ回帰による推定

2.1 リッジ回帰による推定

リッジ回帰やラッソ回帰の推定においては、通常、被説明変数として、もとの被説明変数 y_i からその標本平均を引いて基準化した変数 $\tilde{y}_i = y_i - \bar{y}$ が用いられ、また説明変数としては、もとの説明変数を標本平均が 0 全標本分散が 1 になるように基準化した $\tilde{x}_{i,k} = (x_{i,k} - \bar{x}_k) / \hat{\sigma}_k$, $k=1, \dots, K$ が用いられる。ここで K は説明変数の数であり、

$$\bar{y} = n^{-1} \sum_{i=1}^n y_i, \quad \bar{x}_k = n^{-1} \sum_{i=1}^n x_{i,k}, \quad \text{および} \quad \hat{\sigma}_k = \sqrt{n^{-1} \sum_{i=1}^n (x_{i,k} - \bar{x}_k)^2},$$

である。この場合、もとの重回帰モデルが

$$y_i = \alpha + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_K x_{i,K} + e_i$$

である場合は、リッジやラッソ回帰では

$$y_i - \bar{y} + \bar{y} = \alpha + \beta_1 \left(\hat{\sigma}_1 \frac{x_{i,1} - \bar{x}_1}{\hat{\sigma}_1} + \bar{x}_1 \right) + \beta_2 \left(\frac{x_{i,2} - \bar{x}_2}{\hat{\sigma}_2} \right) + \cdots + \beta_K \left(\frac{x_{i,K} - \bar{x}_K}{\hat{\sigma}_K} \right) + e_i$$

$$\Leftrightarrow \tilde{y}_i = \tilde{\alpha} + \tilde{\beta}_1 x_{i,1} + \tilde{\beta}_2 x_{i,2} + \cdots + \tilde{\beta}_K x_{i,K} + e_i$$

というモデルを推定していることになることに注意、ここで

$$\tilde{\alpha} = \alpha + \beta_1 \bar{x}_1 + \beta_2 \bar{x}_2 + \cdots + \beta_K \bar{x}_K - \bar{y}, \quad \tilde{\beta}_k = \hat{\sigma}_k \beta_k, \quad k=1, \dots, K$$

である。ただし、 $E(e_i)=0$ とすると $E(y_i) = \alpha + \beta_1 E(x_{i,1}) + \beta_2 E(x_{i,2}) + \cdots + \beta_K E(x_{i,K})$ なので、通常、 n が十分大きければ、 $\bar{y} \approx E(y)$, $\bar{x}_k \approx E(x_{i,k})$ なので、 $\tilde{\alpha}$ は

$$\begin{aligned} \tilde{\alpha} &\approx \alpha - E(y) + \beta_1 E(x_1) + \beta_2 E(x_1) + \cdots + \beta_2 E(x_1) \\ &= \alpha - [\alpha + \beta_1 E(x_1) + \beta_2 E(x_1) + \cdots + \beta_2 E(x_1)] + \beta_1 E(x_1) + \beta_2 E(x_1) + \cdots + \beta_2 E(x_1) \\ &= 0 \end{aligned}$$

となり、 $\tilde{\alpha}$ は 0 とみなして(ほぼ)問題ない。よって実際には

$$\tilde{y}_i = \tilde{\beta}_1 x_{i,1} + \tilde{\beta}_2 x_{i,2} + \cdots + \tilde{\beta}_K x_{i,K} + e_i$$

という回帰モデルを推定しても、 n が十分大きければ(ほぼ)問題ないので切片を入れないのが慣習となっている(ただし下記の R による推定では切片も推定しているよう)。また、 n が十分大きければ $\hat{\sigma}_k \approx \sigma_k = \sqrt{\text{var}(x_{i,k})}$ であるので、この回帰モデルの係数 $\tilde{\beta}_k$ ともとの回帰モデルの係数 β_k の間には、 $\tilde{\beta}_k \approx \sigma_k \beta_k$ という関係があることにも注意。

リッジ回帰推定は R のパッケージ `glmnet` を用いて行う。

さきほどと同じ要領でパッケージ `glmnet` をインストールし、さらに読み込んで使えるようにする。

ここでは以下のモデルを推定する。

$$sr_i = \tilde{\beta}_1 \text{pop15}_i + \tilde{\beta}_2 \text{pop75}_i + \tilde{\beta}_3 \text{dpi}_i + \tilde{\beta}_4 \text{ddpi}_i + e_i, \quad i=1, \dots, n$$

ここで sr_i は先ほどの sr_i からその標本平均を引いたものであり、 pop15_i は pop15_i を標本平均が 0、全標本分散が 1 になるように基準化したものである。

まず、データを上記のように基準化する。以下のように入力する。

```
> x=scale(LifeCycleSavings[,2:5])
> y=LifeCycleSavings[,1]-mean(LifeCycleSavings[,1])
```

ここで $A[:,a:b]$ はデータ A の a 列目から b 列目までのデータを取り出すことを意味し、関数 `scale` はデータを標本平均が 0、標本分散が 1 になるように基準化する (関数 `scale` は全標本分散ではなく標本分散が 1 になるように基準化しているが、観測数 n が大きければ違いはほぼないのでここでは無視する)。

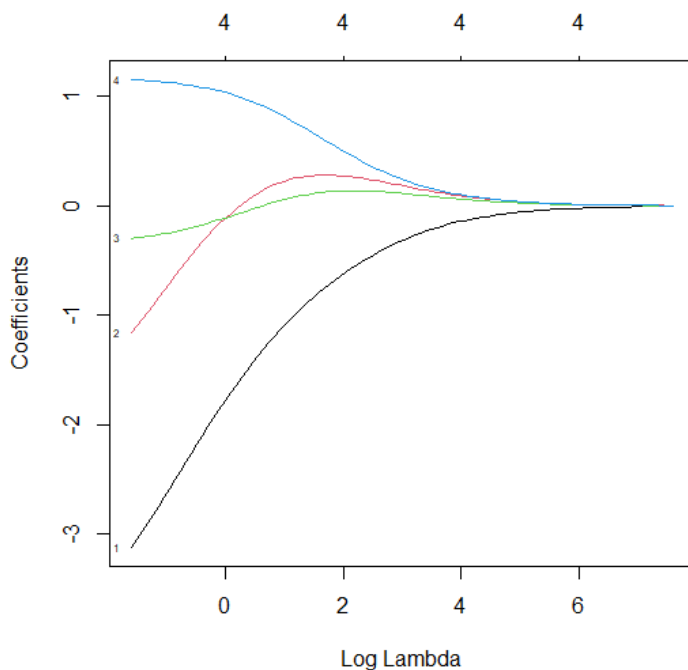
リッジ推定を行うには以下のように入力する。

```
> result_ridge=glmnet(x,y,alpha=0)
(x と y の順序に注意)
```

リッジ推定は調整パラメーター $\lambda > 0$ (ここでは説明しないがわかっているとする) の値によって推定結果が変わるのが、その λ と推定結果の関係 (解パスと呼ばれる) を図示するために、以下のように入力する

```
> plot(result_ridge,xvar="lambda",label=TRUE)
```

これによって以下の図が得られる。X 軸は実際には $\log \lambda$ になっている (よって $\lambda = 1$ は $\log \lambda = 0$ に対応する)。



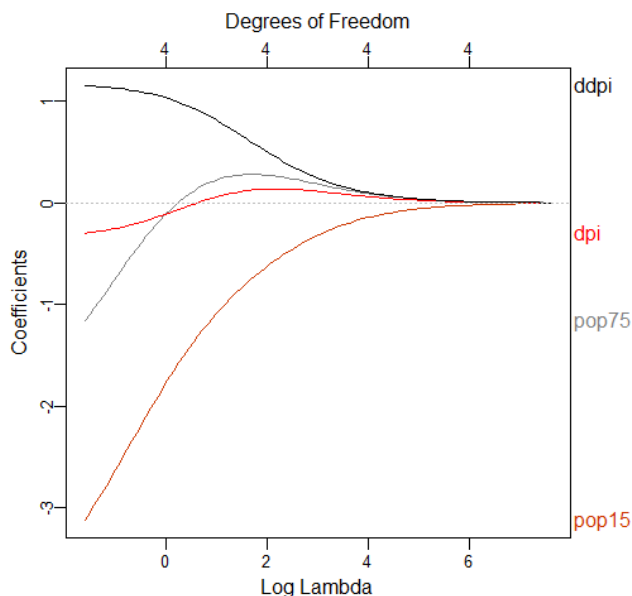
ここで、図の上の数字 (4, 4, 4, 4 のこと) はそれぞれの λ ($\log \lambda$) の値に対して、0 でない係数の

数である。リッジ回帰においては λ が大きくなると、係数の推定値は0に収束していくが、完全に0になることはないので、 λ が大きい場合にも係数の推定値は4つとも0にはなっていない。

上記の図は係数が番号で示されているのでどの線がどの係数に対応するのかわかりづらい。どの線がどの係数の解パスかわかりやすくするために、パッケージ `plotmo` の `plot_glmnet` 関数を用いる。パッケージ `plotmo` をインストールし、読み込んで、以下のように入力する。

```
> plot_glmnet(result_ridge, xvar="lambda", label=TRUE)
```

すると以下の図が得られる。横に説明変数の名前が出てくるため、どの線がどの係数の解パスかわかりやすくなっている。



任意の λ の値に対する、推定結果を見たい時には `coef` 関数を使うことができる。例えば $\lambda=1$ の場合 ($\log \lambda=0$ の場合) の推定結果を見たい場合、以下のように入力する。

```
> coef(result_ridge, s=1)
5 x 1 sparse Matrix of class "dgCMatrix"
      s1
(Intercept) 1.201565e-15
pop15      -1.773291e+00
pop75      -1.089266e-01
dpi        -1.092122e-01
ddpi       1.035876e+00
```

$\lambda=1$ の値に対応する係数の推定結果が出力される。

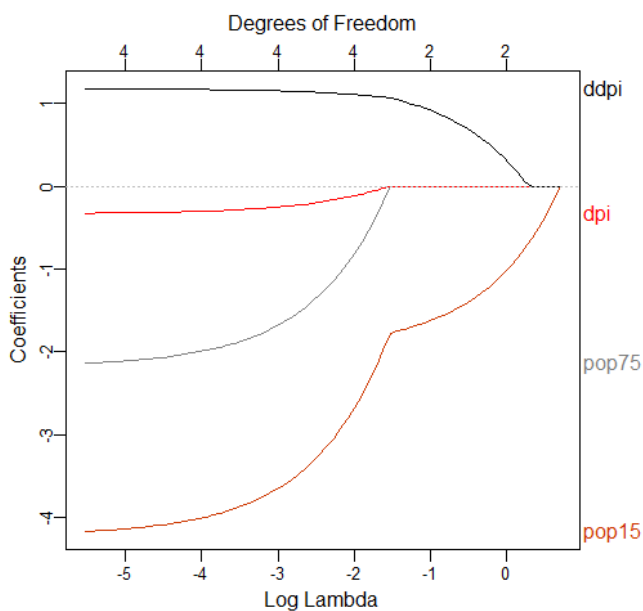
2.1 ラッソ回帰による推定

ラッソ回帰による推定もパッケージ `glmnet` の `glmnet` 関数によって行える。先ほどと同じデータに対して、ラッソ回帰をするには以下のように入力する。

```
> result_lasso=glmnet(x,y,alpha=1)
```

ラッソ回帰も調整パラメーター λ の値によって推定値が変わるが、その解パスはパッケージ `plotmo` の `plot_glmnet` 関数を用いることによって以下のように図示できる。

```
> plot_glmnet(result_lasso,xvar="lambda",label=TRUE)
```



ラッソ回帰の場合は、 λ が大きくなると、推定値が正確に 0 になる係数がある。上記でも、 λ がある値以上になると `dpi` と `pop75` の係数の推定値が 0 になっていることがわかる。例えば $\lambda=1$ の場合の推定結果を見てみると、

```
> coef(result_lasso,s=1)
5 x 1 sparse Matrix of class "dgCMatrix"
      s1
(Intercept)  9.632926e-16
pop15        -1.016170e+00
pop75         .
dpi           .
ddpi          3.068189e-01
```

であるから、`pop75` と `dpi` の係数が 0 になっていることがわかる。