

# Rによる非線形最小二乗法<sup>†</sup>

## 1. 非線形回帰モデル

回帰モデルにおいて被説明変数が**未知パラメータについて線形である**場合は「線形回帰モデル」とよばれる。例えば以下のようなモデルはすべて線形回帰モデルの例である。

$$\text{(例 1)} \quad y_i = \alpha + \beta x_i + \varepsilon_i, \quad \text{(例 2)} \quad y_i = \alpha + \beta \log(x_i) + \varepsilon_i, \quad \text{(例 3)} \quad \log y_i = \alpha + \beta \frac{1}{x_i} + \varepsilon_i$$

このようなモデルの場合は通常の最小二乗法によって未知パラメータ  $\alpha, \beta$  を推定することができる。このような線形回帰モデルに対して、被説明変数が**未知パラメータについて非線形である**場合は「**非線形回帰モデル**」と呼ばれる。例えば以下のようなモデルである。

$$\text{(例 4)} \quad y_i = \exp(\alpha + \beta x_i) + \varepsilon_i, \quad \text{(例 5)} \quad y_i = \frac{1}{1 + \exp(\alpha + \beta x_i)} + \varepsilon_i$$

$$\text{(例 6)} \quad y_i = \sin(\alpha + \beta x_i) + \varepsilon_i, \quad \text{(例 7)} \quad y_i = \alpha + \alpha\beta x_i + \beta z_i + \varepsilon_i$$

このようなモデルはより一般的に

$$y_i = m(\mathbf{x}_i, \boldsymbol{\theta}) + \varepsilon_i$$

と表せる。ここで  $\mathbf{x}_i$  は説明変数のベクトル、 $\boldsymbol{\theta}$  は未知パラメータのベクトル、 $\varepsilon_i$  は直接観測できない誤差項である。説明変数  $\mathbf{x}_i$  と誤差項  $\varepsilon_i$  は独立であると仮定する。例えば上記の例 5 では

$$\mathbf{x}_i = x_i, \quad \boldsymbol{\theta} = (\alpha, \beta), \quad m(x_i, \boldsymbol{\theta}) = \frac{1}{1 + \exp(\alpha + \beta x_i)}$$

となる。このような非線形回帰モデルに含まれる未知パラメータを推定するには**非線形最小二乗法**と呼ばれる方法を用いる。非線形最小二乗法の具体的なやり方はここでは述べていない(線形の最小二乗法と原理的にはまったく同じである)。以下では R によって上記のようなモデルを推定する方法のみを述べる。

## 2. R 関数 “nlm” による非線形最小二乗法

### 2.1 人口データの非線形モデル推定

R によって非線形最小二乗推定法を行うやり方を簡単に説明する。説明のために用いるデータとして、パッケージの “car” の中に入っているアメリカの 10 年ごとの人口の推移のデータ(単位:百万人)を使用する。まずパッケージ car をダウンロード(インストール)する(パッケージのダウンロードおよびインストールの仕方はすでに説明したので省略)。次に

```
> library(car)
```

としてパッケージを R に読み込ませて使用できるようにする。パッケージを読み込んだので、ここに入っている “USPop” というデータ(アメリカの人口推移)が使用できる。最初の 5 行を見てみると

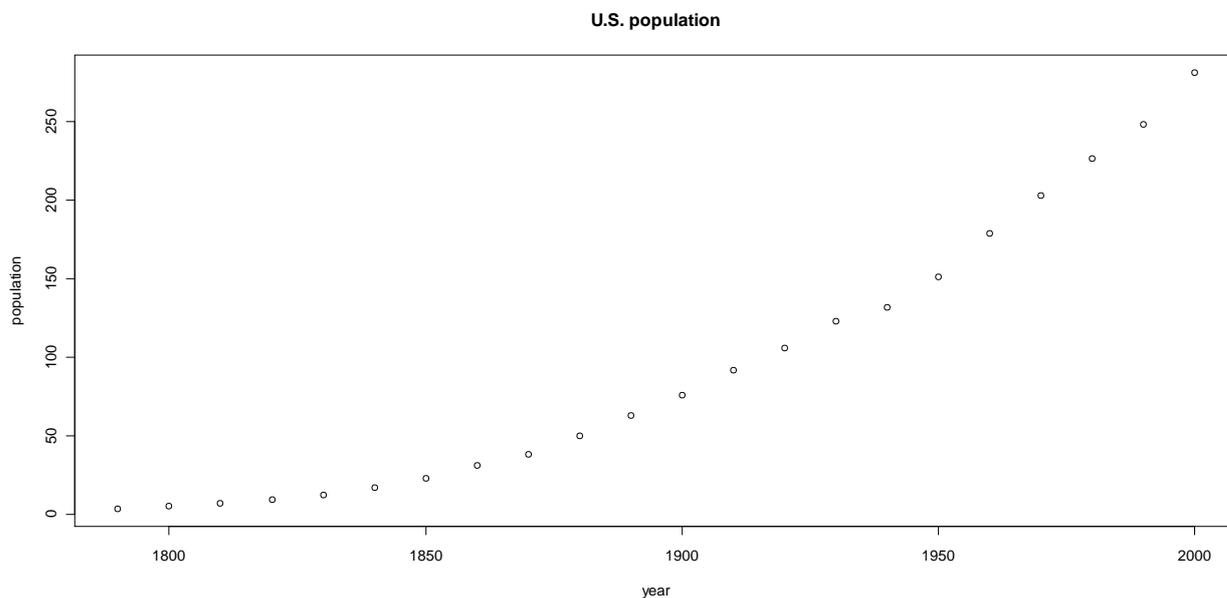
```
> head(USPop, 5)
  year population
1 1790   3.929214
2 1800   5.308483
3 1810   7.239881
4 1820   9.638453
5 1830  12.860702
```

となっている。year は西暦である。データの特徴を視覚的にとらえるために散布図を描いてみる。

```
> plot(population~year, data=USPop, main="U.S. population")
```

<sup>†</sup> この資料は私のゼミおよび講義で R の使用法を説明するために作成した資料です。ホームページ上で公開しており、自由に参照して頂いて構いません。ただし、内容について、一応検証してありますが、間違いがあるかもしれません。間違いがあった場合でもそれによって生じるいかなる損害、不利益について責任は負いかねますのでご了承ください。

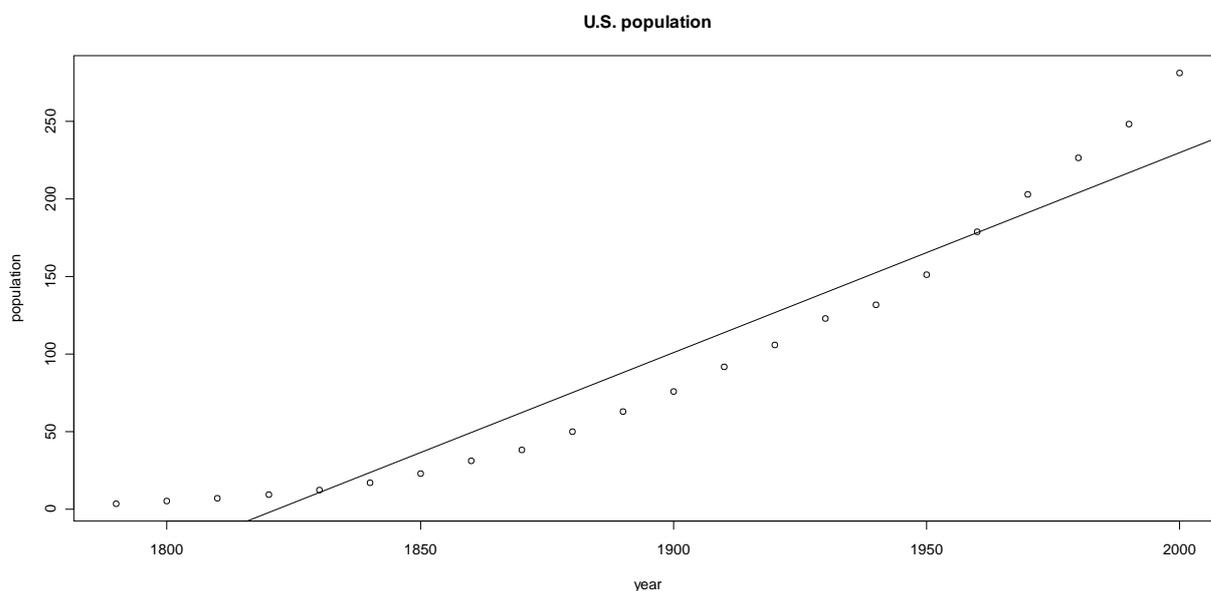
すると以下のような散布図が描かれる。



この図より明らかなように人口は西暦に比例して増えていくもののその関係は直線的ではない。例えばここに最小二乗法によって推定した直線を書き込んでみよう

```
> abline(lm(population~year, data=USPop))
```

すると、以下のように直線が書き込まれる。この直線はあまりデータにフィットしていないことがわかる。



上記から考えられるのは人口の推移は西暦と非線形の関係にあるであろうという事である。そこで以下のような非線形回帰モデルを考える。

$$y_i = m(x_i, \theta) + \varepsilon_i = \frac{\alpha}{1 + \exp[-(\beta_0 + \beta_1 x_i)]} + \varepsilon_i, i = 1, \dots, N.$$

ここで  $y_i$  は人口、 $x_i$  は西暦である。未知パラメータは  $\theta = (\alpha, \beta_0, \beta_1)$  の3つである。このような非線形のモデルを推定するRの関数として“nls”関数がある。上記のモデルを nls 関数を用いて推定してみよう。以下のように入力する。

```
> result=nls(population~a/(1+exp(-(b0+b1*year))),
+ start=list(a=400,b0=-49,b1=0.025),data= USPop, trace=TRUE)
```

ここで population は USPop のデータにある population が被説明変数である事を表している。「~」の後の  $a/(1+\exp(-(b_0+b_1 \cdot \text{year})))$  はそれに対して具体的にどのような非線形の関数をあてはめるかを指定している。year は USPop にある year という変数である。コマンドを打ち込んでいる途中で Enter キーを押せば次の行に移り、その際 + と表示される。ここでは上記の部分を入力した後 Enter キーによって次の行に移っている。そこから次の

```
start=list(a=400,b0=-49,b1=0.025)
```

を打ち込んでいるが、これは**初期値**と呼ばれるものである。非線形最小二乗法を始めるにはまず未知パラメータの値として初期値を指定する必要がある。そこから収束計算によって、未知パラメータの真の値の推定値を計算する。初期値は基本的には適当な値でよいが、通常、この初期値が真の値に近ければ近いほど収束計算が早く終わる。逆に初期値が真の値から離れているほど計算に時間がかかる。また場合によっては初期値が異なると計算結果が異なるという事もある。いくつか異なった初期値を試すのが普通である。ここでは上記のようにした。最後の trace=TRUE は途中の収束計算の経過を表示するかどうかを指定するものである。trace=TRUE であれば途中の収束計算の経過を表示し、trace=FALSE であれば表示しない。何も入力しなければ自動的に trace=FALSE となる。

Enter キーを押して上記のコマンドを実行すると、計算の途中結果が表示される。左からそれぞれ「非線形最小二乗法の残差平方和の値」、「a の値」「b0 の値」「b1 の値」の収束計算の途中経過が表示される。

推定結果は summary 関数によって見ることができる。

```
> summary(result)

Formula: population ~ a/(1 + exp(-(b0 + b1 * year)))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
a  440.833328   35.000136   12.60 1.14e-10 ***
b0 -42.706978    1.839138  -23.22 2.08e-15 ***
b1   0.021606    0.001007   21.45 8.87e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.909 on 19 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.481e-06
```

パラメータの推定値は Parameters: の下の Estimate の列に並んでいる。Std. Error, t value, Pr(>|t|) は標準誤差、それぞれ t 値 (推定値/標準誤差: 真の値が 0 という帰無仮説の検定用)、その P 値である。Residual standard error は誤差項  $\varepsilon_i$  の標準偏差の推定値である。

推定値の下での、この非線形回帰モデルの残差は

```
> residuals(result)
```

によって見ることができる(確認してください)。また、推定値の下でのあてはめ値は

```
> predict(result)
```

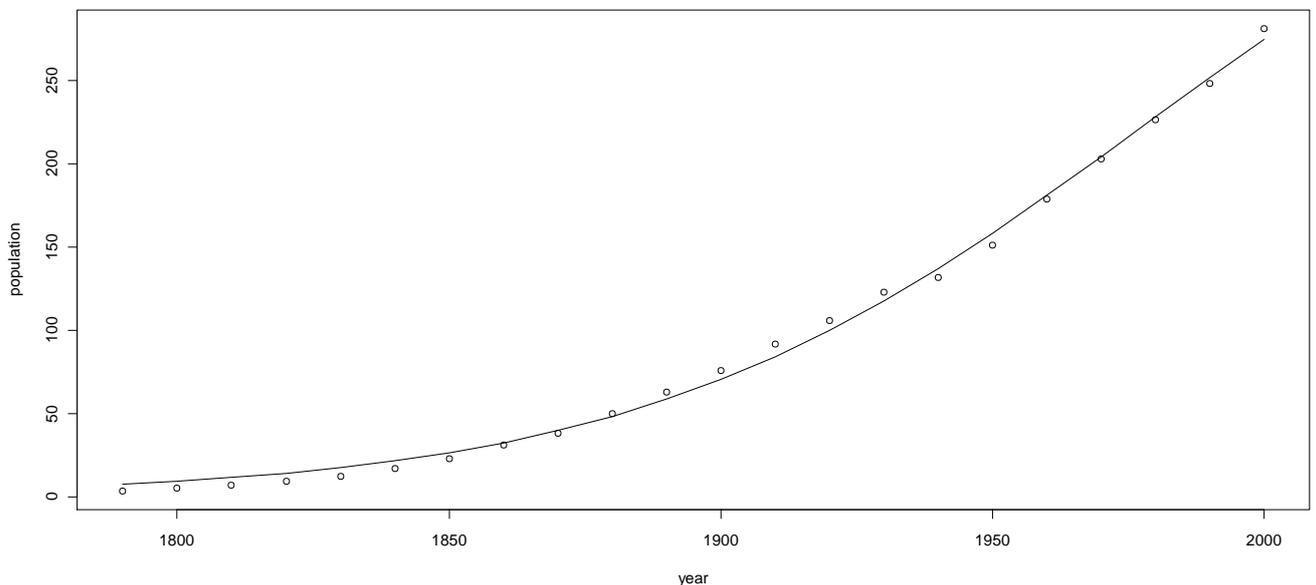
によって見る事ができる(確認してください)。実際の西暦と人口の値の散布図に非線形の当てはめ値の曲線を書き込むこともできる。(以前の図を閉じて)改めて、実際の西暦と人口の散布図を書いておこう。

```
> plot(population~year,USPop)
```

次にこの散布図に当てはめ値の曲線を加えるために

```
> lines(seq(1790,2000,by=10),predict(result))
```

と入力し、エンターキーを押す。ここで `seq(1790,2000,by=10)` は 1790 から 2000 まで 10 ずつ増加させた数列(英語で sequence)のデータを作るという意味である。すると散布図に当てはめ値の曲線が書き込まれた以下の図が出力される。



上記の手順は例えば、曲線を書き込む手順をいくつか段階わけして

```
> pre1=predict(result)
```

```
> pyear1=seq(1790,2000,by=10)
```

(ここで p は予測-predict-用の西暦という事で p を付けているが、名前は何でもよい)

```
> lines(pyear1,pre1)
```

としても同じ図が出力される(確認してください)。

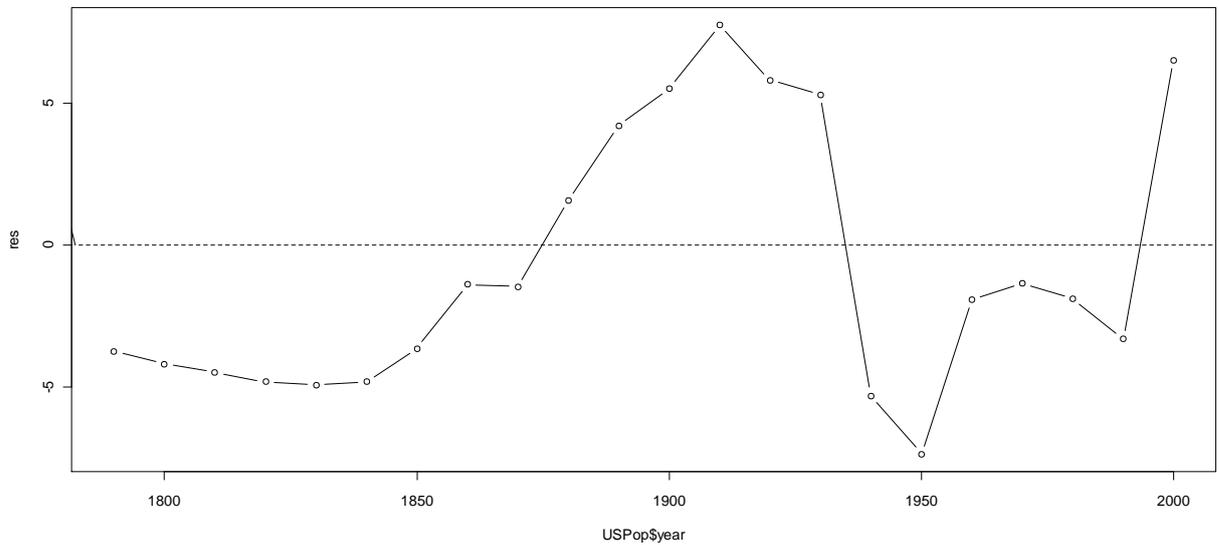
さらに残差をプロットするには

```
> res=residuals(result)
```

```
> plot(USPop$year,res,type='b')
```

```
> abline(h=0,lty=2)
```

というコマンドを実行する。2行目の `abline` は線を加えるという事で `h=0` は y 軸の 0 の部分に水平な (horizontal) 線を加えるという事である。`lty=2` はその線が点線となるようにするという事である。以下のような図が出力される。



また、もとのデータは 2000 年までしかないが、それ以降の年の人口の予測値を推定値の下でのこの非線形モデルを使用して予測したものを計算することもできる。まず

```
> pyear2 = data.frame(year=seq(1790,2100,by=10))
```

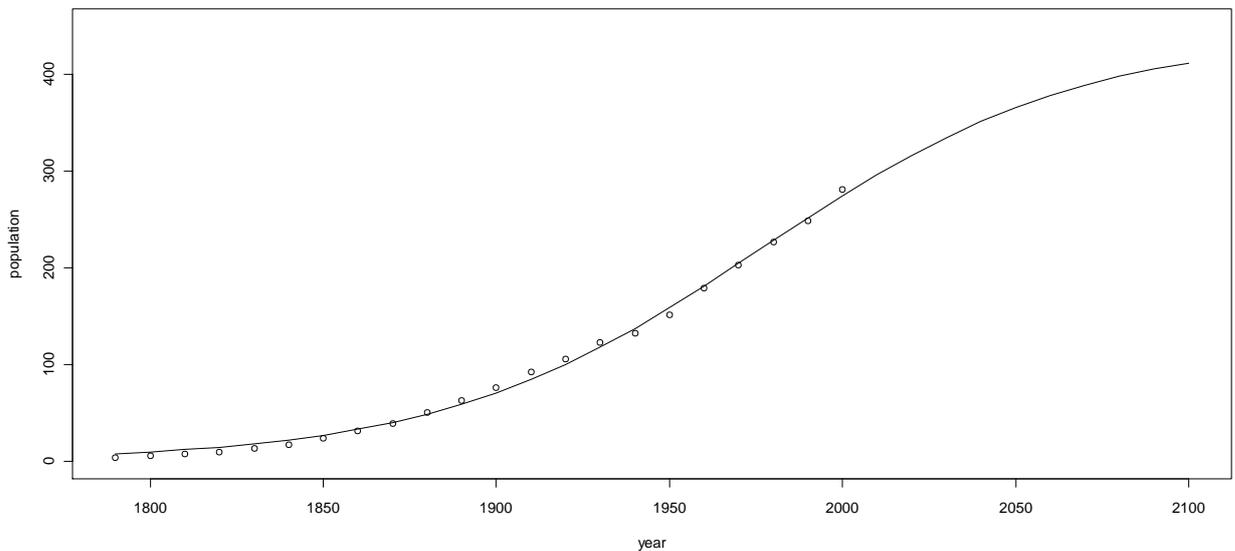
によって 1790 年から 2100 年までの 10 年ごとのデータを作る。次に

```
> pre2=predict(result,pyear2)
```

とすれば 2001 年以降 (23 個めから) の予測値も出力され、それに pre2 という名前がつけられる。最後に予測図と実際の値との図は

```
> plot(population~year,USPop,xlim=c(1790,2100),ylim=c(0,450))
> lines(pyear2$year,pre2)
```

によって描くことができる。以下のような図が出力される。



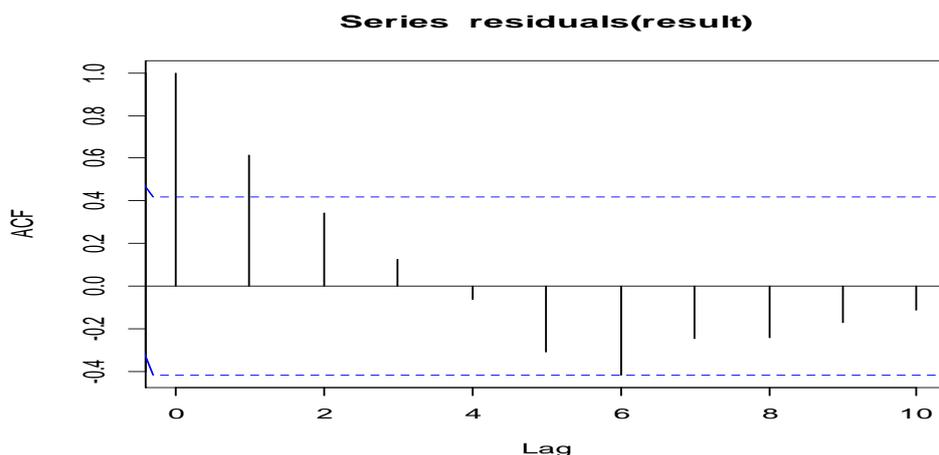
2000年以降の予測値が図示されているのがわかる。

## 2.2 誤差項をAR(1)モデルにしてみる

上の残差のプロットは残差に正の系列相関があることを示唆している。実際に残差の自己相関を

```
> acf(residuals(result), lag.max=10)
```

で計算してみると以下のようなになる。



これより、1 次の自己相関が高いことがわかる。よって(この図ではむしろ MA(1)のように見えるが、ひとまず)誤差項が AR(1)モデルに従っているとしてモデル化してみよう。以下のようなモデルを考える。

$$y_i = \frac{\alpha}{1 + \exp[-(\beta_0 + \beta_1 x_i)]} + \varepsilon_i, \quad \varepsilon_i = \phi \varepsilon_{i-1} + u_i, \quad u_i \sim \text{i.i.d.}(0, \sigma^2)$$

ここで、 $\varepsilon_i = y_i - \alpha / (1 + \exp[-(\beta_0 + \beta_1 x_i)])$  を  $\varepsilon_i = \phi \varepsilon_{i-1} + u_i$  に代入すると、このモデルは

$$y_i = \frac{\alpha}{1 + \exp[-(\beta_0 + \beta_1 x_i)]} + \phi \left[ y_{i-1} - \frac{\alpha}{1 + \exp[-(\beta_0 + \beta_1 x_{i-1})]} \right] + u_i \quad (1)$$

と表すことができる。このモデルを推定してみよう。

このモデルの推定において注意すべき点の1つが  $\phi$  の推定である。定常性を仮定するなら  $\phi$  の値の取りうる範囲は  $-1 < \phi < 1$  となる。よって推定値もこの範囲の値にならないといけないが、通常統計ソフトの最適化アルゴリズムでは未知パラメータについて最適化する際、パラメータの取りうる範囲に制約を課さずに最適化するものがほとんどである。つまりこの場合、そのようなアルゴリズムを用いて上記の  $\phi$  を推定すると、計算の途中で  $\phi$  が上記の範囲の外側の値を取ってしまう可能性があり、その場合は目的関数がおかしな値をとリアルゴリズムがストップしてしまう(ただし、後に見るように実は `nl` 関数にはパラメータの範囲を指定することができるアルゴリズムが存在する。が、ひとまずそれは使用しないものとして話を進める)。このような問題を避けるためには、例えば  $\phi$  を直接推定するのではなく、 $\phi$  はある無制約パラメータ  $\kappa$  の関数で、その関数は  $-1$  から  $1$  の間の値をとる、すなわち  $\phi = \phi(\kappa)$ 、 $-1 < \phi(\kappa) < 1$ 、 $-\infty < \kappa < \infty$  であるして  $\kappa$  を推定し、 $\kappa$  の推定値  $\hat{\kappa}$  を用いて  $\phi$  を  $\hat{\phi} = \phi(\hat{\kappa})$  と推定する方法が考えられる。このような変換としては、例えば

$$\phi(\kappa) = 2\lambda(\kappa) - 1, \quad \lambda(\kappa) = \frac{1}{1 + \exp(-\kappa)}$$

が考えられる。 $\lambda(\kappa)$  はロジスティック分布の分布関数であり(0, 1)の範囲の値を取るが、それを2倍して1を引いているので  $\phi(\kappa)$  は(-1, 1)の範囲の値を取ることになる。

それでは(1)式のモデルを推定する。ただし、今回は自己相関は正の値なので、 $\phi$  の取りうる値としては(0, 1) とすれば十分であろう。よって  $\phi(\kappa) = \lambda(\kappa)$  とする。この時、(1)式をパラメータ  $\kappa$  について書き直

すと

$$y_i = \frac{\alpha}{1 + \exp[-(\beta_0 + \beta_1 x_i)]} + \frac{1}{1 + \exp(-\kappa)} \left[ y_{i-1} - \frac{\alpha}{1 + \exp[-(\beta_0 + \beta_1 x_{i-1})]} \right] + u_i \quad (2)$$

となる。このモデルの推定において必要な系列は  $y_i, y_{i-1}, x_i, x_{i-1}$  であるので、それらを作っておこう(データの数一つ減ることに注意)。元のデータを見てみると

```
> USPop$population
 [1]  3.929214  5.308483  7.239881  9.638453 12.860702
 [6] 17.063353 23.191876 31.443321 38.558371 50.189209
[11] 62.979766 76.212168 92.228496 106.021537 123.202624
[16] 132.164569 151.325798 179.323175 203.302031 226.542199
[21] 248.709873 281.421906
> USPop$year
 [1] 1790 1800 1810 1820 1830 1840 1850 1860 1870 1880 1890 1900
[13] 1910 1920 1930 1940 1950 1960 1970 1980 1990 2000
```

なので、元のデータはそれぞれ 22 個ある。ここからまず  $y_i$  の系列を作る(これを pop とする)

```
> pop=USPop$population[2:22]
> pop
 [1]  5.308483  7.239881  9.638453 12.860702 17.063353
 [6] 23.191876 31.443321 38.558371 50.189209 62.979766
[11] 76.212168 92.228496 106.021537 123.202624 132.164569
[16] 151.325798 179.323175 203.302031 226.542199 248.709873
[21] 281.421906
```

同様に  $y_{i-1}$  の系列は (これを popL とする。L は lag の L)

```
> popL=USPop$population[1:21]
> popL
 [1]  3.929214  5.308483  7.239881  9.638453 12.860702
 [6] 17.063353 23.191876 31.443321 38.558371 50.189209
[11] 62.979766 76.212168 92.228496 106.021537 123.202624
[16] 132.164569 151.325798 179.323175 203.302031 226.542199
[21] 248.709873
```

同様に  $x_i$  と  $x_{i-1}$  の系列は

```
> year=USPop$year[2:22]
> yearL=USPop$year[1:21]
```

と作ることができる。それでは (2) 式を推定してみよう。以下のコマンドを打ち込む (少し長いが)。

```
> result2=nls(pop~a/(1+exp(-(b0+b1*year)))
+ +(1/(1+exp(-k)))*(popL-(a/(1+exp(-(b0+b1*yearL))))),
+ start=list(a=400,b0=-49,b1=0.025,k=0),trace=TRUE)
```

(適当なところで改行していることに注意)。結果は

```
> summary(result2)
```

```
Formula: pop ~ a/(1 + exp(-(b0 + b1 * year))) + (1/(1 + exp(-k))) * (popL -
(a/(1 + exp(-(b0 + b1 * yearL)))))
```

Parameters:

```
Estimate Std. Error t value Pr(>|t|)
a 633.931269 253.142655 2.504 0.022749 *
b0 -36.091054 6.823476 -5.289 6.01e-05 ***
```

```
b1 0.017933 0.003754 4.777 0.000175 ***
k 1.666962 1.522341 1.095 0.288789
```

```
---
```

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

```
Residual standard error: 3.412 on 17 degrees of freedom
```

```
Number of iterations to convergence: 12
```

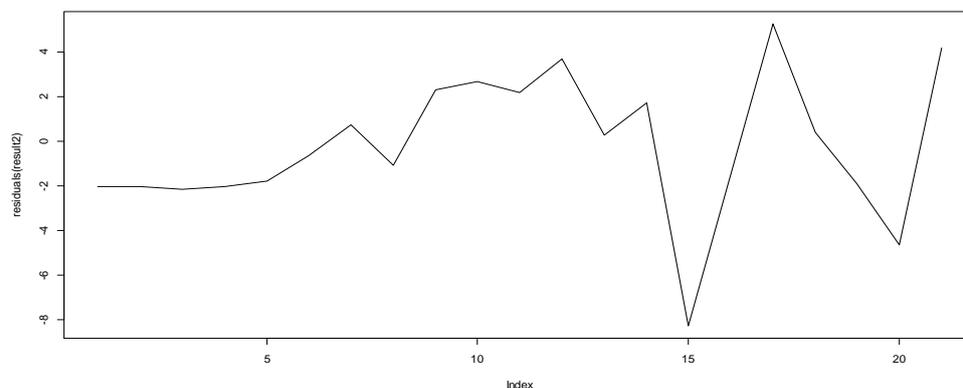
```
Achieved convergence tolerance: 6.218e-06
```

となる(ここで  $\kappa$  の推定値が有意になっていないが、これは直観的には  $\phi$  が 0.5 と有意に異なるということ、 $\phi$  が 0 と異なるということではないことに注意)。  $\kappa$  の推定値から  $\phi$  の推定値を計算してみると

```
> 1/(1+exp(-1.666962))
[1] 0.8411704
```

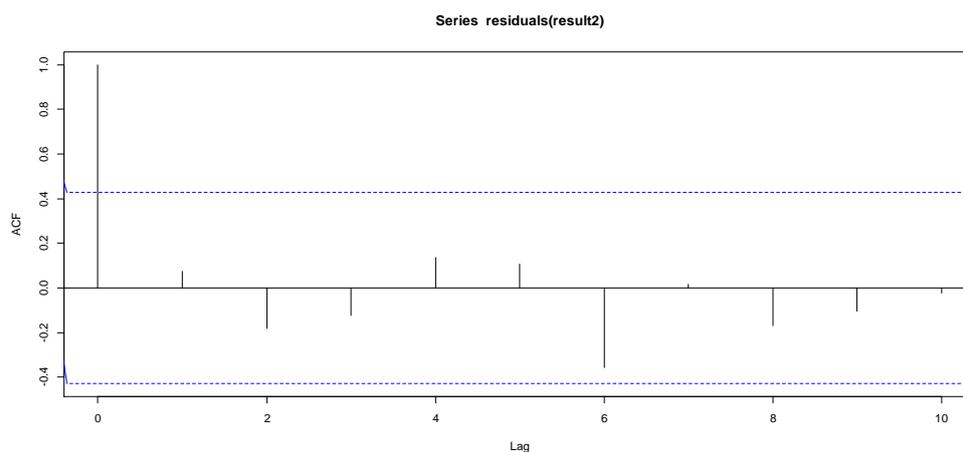
である。だいたい  $\hat{\phi} \approx 0.84$  であり、かなり高い値になっている。残差をプロットしてみると

```
> plot(residuals(result2), type="l")
```



のようになる。その自己相関は

```
> acf(residuals(result2), lag.max=10)
```

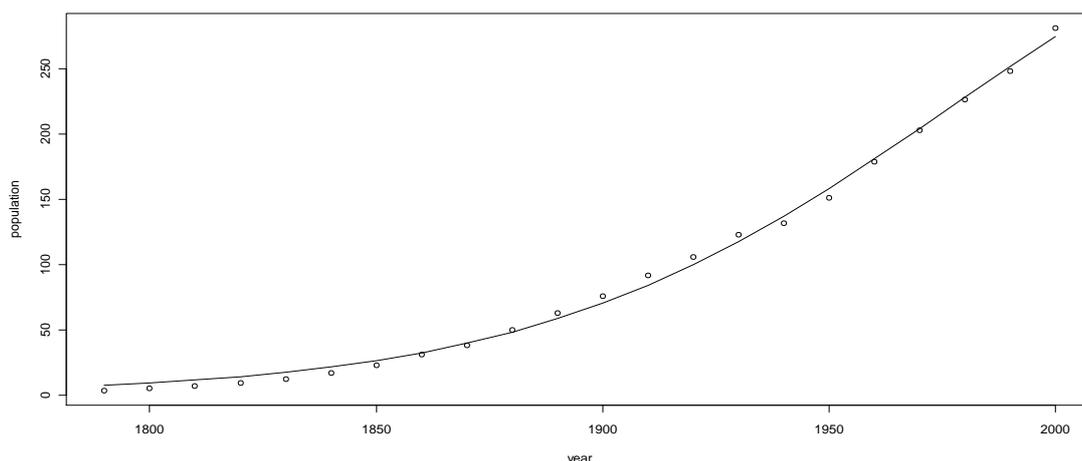
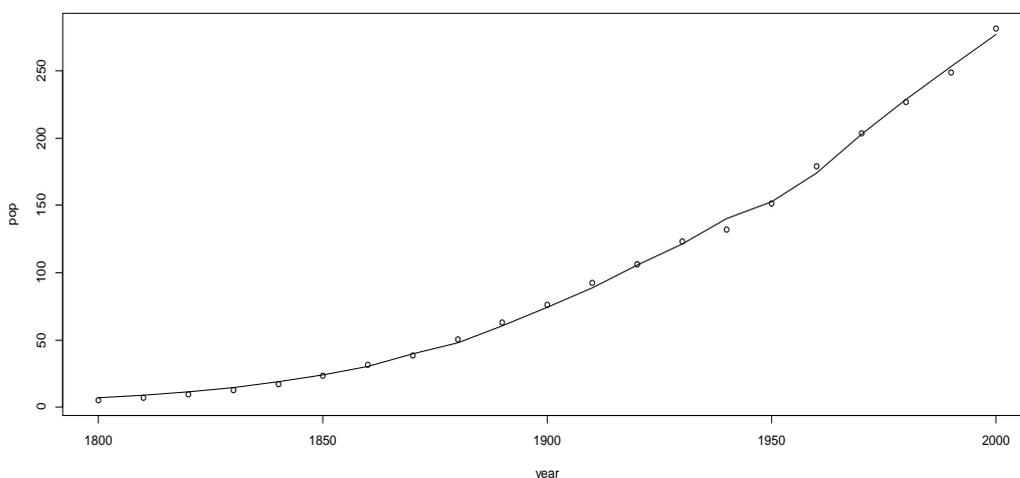


となる。一応自己相関が消えているのがわかる。またあてはめ値のグラフは

```
> plot(pop~year)
> lines(seq(1800,2000,by=10), predict(result2))
```

より下の図のようになる (pop と year は 1800 から 2000 までの値であることに注意) 下の図のさらに

下の図はもとのあてはめ値のグラフ。



若干あてはまりがよくなっているのが見て取れる。これは推定結果のところの誤差項の分散が小さくなっていることからわかる。このモデルを用いて予測を行うこともできるが、若干の注意が必要である。このモデルでは説明変数に被説明変数のラグが入っているため、1期先予測であれば(被説明変数の)現在の値、2期先予測であれば1期先の値が必要になる。1期先の値はもちろんまだ観測されていないので、その予測値で置き換えて2期先予測を行い、またその2期先予測値を用いて3期先予測を行うという逐次的に予測値を計算する必要がある。

#### 4. アルゴリズム“port”を使用してパラメーターの範囲を指定する

さて、上記では最適化のアルゴリズムにおいてパラメーターの範囲が指定できない場合の対処法を見てみたが、実は nls 関数では最適化の際にパラメーターの範囲を指定できる“port”というアルゴリズムが使用できる。このアルゴリズムを使用することによって、最適化の際にそれぞれのパラメーターが取りうる範囲の上限と下限を指定することができる。このアルゴリズムを用いて、(1)式のモデルにおいて  $\phi$  を直接推定してみよう(ここで  $\phi$  は  $f$  というパラメーターで表すとす)。以下のように入力する。

```
> result2=nls(pop~a/(1+exp(-(b0+b1*year)))+
+ f*(popL-(a/(1+exp(-(b0+b1*yearL))))),
+ alg="port", start=list(a=400,b0=-49,b1=0.025,f=0),
+ lower=list(a=0,b0=-Inf,b1=0,f=-1), upper=list(a=Inf,b0=Inf,b1=Inf,f=1))
```

ここで `alg="port"` によって port (何かの略だろうがよくわからない) というアルゴリズムを使用することを指定し、`lower=list(a=0,b0=-Inf,b1=0,f=-1)` でそれぞれのパラメーターの下限を `upper=list(a=Inf,b0=Inf,b1=Inf,f=1)` で上限を指定している (Inf は  $\infty$  を意味する)。推定結果は

```
> summary(result2)

Formula: pop ~ a/(1 + exp(-(b0+b1*year)))+f*(popL-(a/(1+exp(-(b0 +
  b1 * yearL))))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
a  633.927689 253.135918   2.504 0.022747 *
b0 -36.091157   6.823368  -5.289 6.01e-05 ***
b1   0.017933   0.003754   4.777 0.000175 ***
f    0.841168   0.203389   4.136 0.000691 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.412 on 17 degrees of freedom

Algorithm "port", convergence message: relative convergence (4)
```

となる。今回も  $f$  の推定値がほぼ 0.84 であり、そのほかの推定値もほぼ同じである。