R による回帰分析(最小二乗法)[†]

この資料では

1. **データを読み込む**
2. **最小二乗法によってパラメーターを推定する**
3. **データをプロットし、回帰直線を書き込む**
4. **いろいろなデータの読み込み方**

について簡単に説明する。

1. データを読み込む

以下では `read.table( )` 関数を使ってテキストファイル(拡張子が `.txt` のファイル)のデータの読み込み方を説明する。

1.1 データの用意

テキストファイルにデータを用意する。以下では `usdata01.txt` というファイルにある 3 列のデータを読み込む(アメリカの実質個人可処分所得と実質個人消費支出(単位:100 億ドル)のデータ。1 列目は西暦、2 列目は実質個人可処分所得、3 列目が実質個人消費支出である)。

(`usdata01.txt` のデータ)

```
1960      157     143
61        162     146
 :         :       :
```

1.2 作業ディレクトリの変更

R を起動し R の画面のメニュー・バーから「ファイル」→「ディレクトリの変更」によってデータ(`usdata02.txt`) が置いてあるディレクトリを指定

1.3 データの読み込み

次のコマンドを実行する(以下を打ち込んでエンター・キーを押す)

```
> usdata01=read.table("usdata01.txt")
```

これは `usdata01.txt` にあるデータに `usdata01` という名前を付けて R に読み込みという命令を実行している。実際に読み込めたかどうかを確認するには

[†]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。

```
> usdata01
```

と打ち込んでエンター・キーを押すと

```
      V1   V2   V3
1   1960  157  143
2     61  162  146
:      :    :    :
```

のように表示され、読み込まれていることがわかる。ここで V1, V2, V3 は R が列のデータに自動的につけた変数名である(variable 1、 variable 2 などの略)

上記ではデータが テキストファイル(拡張子が txt のファイル)で与えられているとしているが、場合によってはで一たが csv ファイル(拡張子が csv のファイル)で与えられている場合がある。この場合、1 つの方法としては csv ファイルをテキストファイルとして読み込むか、または read.csv() 関数を用いて読み込むこともできる。例えばもし先ほどのデータが usdata01.csv で与えられている場合は、

```
> usdata01=read.csv("usdata01.csv")
```

とすれば読み込める。以下の read.table() 関数についての説明は全て read.csv() 関数にも当てはまる。

データを読み込むときの注意事項として、数値に “,” (カンマ)が入っているとうまく読み込めなくなるので(例えば 1,000 などのような場合)、読み込むときは数値からカンマを抜いたデータにしておく。

2. 最小二乗法によってパラメーターを推定する

消費(V3)を所得(V2)に回帰する回帰分析を最小二乗法で行ってみよう。最小二乗法を R で行うには lm( ) 関数 を使う。

```
> result=lm(V3~V2,usdata01)
```

というコマンドを実行する。result というのは回帰分析の結果に result という名前を付けるという事である。ここでは result と名前を付けたがこの名前は自由に決められる(例えば estimates でも何でもよい)。1 つ目の引数の V3~V2 は V3 を被説明変数、V2 を説明変数とする回帰分析を行うという事である。2 つめの引数として usdata01 と打ち込むのはこれらの変数が usdata01 というデータのところにある変数であるというのを R に知らせるためである。

結果を出力するには summary( ) という関数を用いる。

```
> summary(result)
```

を実行すると

```
Call:
lm(formula = V3 ~ V2, data = usdata01)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-12.4526  -4.2491  -0.6491   4.9113  15.1726

Coefficients:
             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -18.785055   2.594585   -7.24 5.79e-09 ***
V2            0.969369   0.006476  149.68   < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘.’ 1 $
Residual standard error: 6.226 on 43 degrees of freedom
Multiple R-squared: 0.9981,    Adjusted R-squared: 0.998
F-statistic: 2.24e+04 on 1 and 43 DF,  p-value: < 2.2e-16
```

のように出力される。最初の

```
Residuals:
     Min       1Q   Median       3Q      Max
-12.4526  -4.2491  -0.6491   4.9113  15.1726
```

というのは回帰分析の残差の性質を表している。Min, 1Q, Median, 3Q, Max というのはそれぞれ最小値、第 1 分位数、中央値、第 3 分位数、最大値である。次の

```
Coefficients:
             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -18.785055   2.594585   -7.24 5.79e-09 ***
V2            0.969369   0.006476  149.68   < 2e-16 ***
```

というのは切片(Intercept)と V2 という変数の係数の推定値がそれぞれ -18.785055, 0.969369 である事、また Std .Error, t value, Pr(>|t|)はこれらの推定量の標準誤差、t 値、t 値の P 値をそれぞれ表している。また Multiple R-squared が通常の決定係数、Adjusted R-squared が自由度修正済み決定係数を表す。

また上記と同じ結果を出力するコマンドとして

```
> result=lm(usdata01$V3~usdata01$V2)
```

と打ち込んでもよい。ここで usdata01$V3 というのは usdata01 にある V3 という変数であるというのを直接示している(usdata$V2 も同様)。よってこの場合、2 つ目の引数として usdata01 と打ち込む必要がなくなる。またさらに、usdata01$V3 のように V3 を指定するのに毎回前に usdata01$を受けるのは若干面倒であるので、attach()関数を用いて、あらかじめ

```
> attach(usdata01)
```

としておくと、

```
> result=lm(V3~V2)
```

とするだけで上記と同じ結果が出る。attach()で行った処理をもとに戻すには detach()関数

を用いる。先ほどの場合

```
> detach(usdata01)
```

とすれば以後は V3 や V2 だけではデータを認識せず、さきほどのように usdata01$V3 と入力しなければならなくなる。

3. データをプロットし、回帰直線を書き込む

V2 を X 軸、V3 を Y 軸とした散布図を描くには

```
> plot(usdata01$V3~usdata01$V2)
```

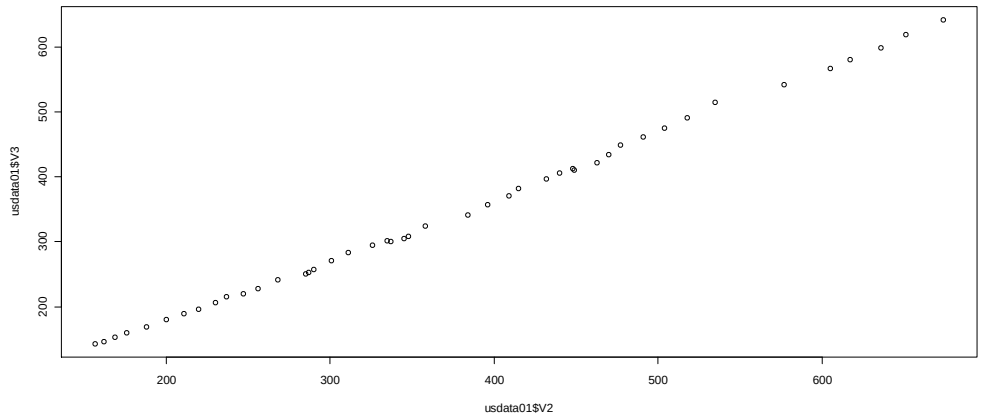
もしくは

```
> plot(usdata01$V2,usdata01$V3)
```

（上記 2 つのコマンドでは変数の順序が逆になっていることに注意）、もしくは (attach(usdata01)とした後であれば)

```
> plot(V2,V3)
```

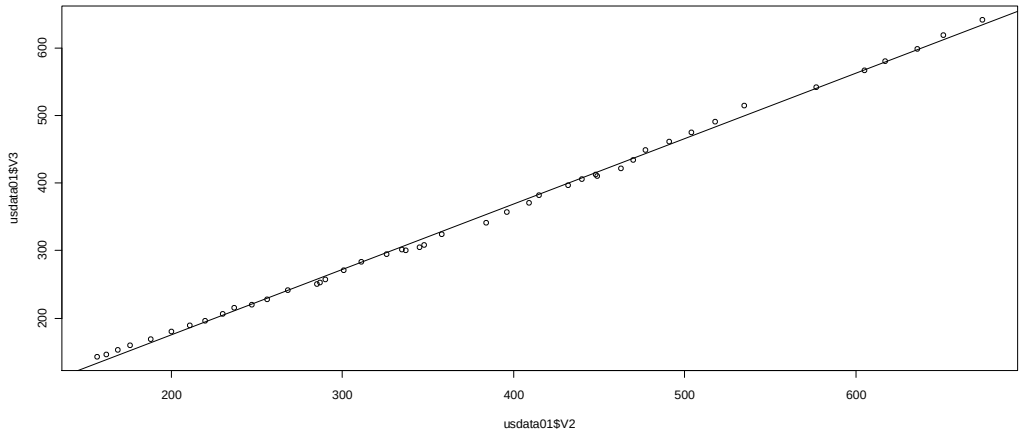
と入力し、実行する。すると以下のような散布図が出力される。



さらにここに先ほど推定した回帰直線を書き込むには

```
> abline(result)
```

を実行すればよい。すると以下のようになる。



4. いろいろなデータの読み込み方

先ほどは usdata01.txt というファイルを読みこんだ。次は usdata02.txt というファイルを読み込んでみよう。 usdata01.txt はいきなりデータから始まっていたが、usdata02.txt は一行目に変数の名前が入っている以下のようなファイルである。

(usdata02.txt のデータ)

```
Year    income  consumption
1960       157    143
61         162    146
 :           :       :
```

このように1行目に変数の名前が入っているようなデータの場合、Rにそれを教えてあげる必要がある。このようなデータを読み込むには以下のようなコマンドを実行する。

```
> usdata02=read.table("usdata02.txt",header=T)
```

このコマンドによる結果は

```
> usdata02
    Year income consumption
1   1960    157         143
2     61    162         146
:     :      :           :
```

のようになる。usdata01.txt を読み込んだ時と異なり、変数には Year や income などファイルの中の名前がついている。あとの分析はまったく同じである(データの名前が usdata01 から usdata02 へ、変数名が V2 から income、V3 から consumption に変わるだけ)。

また最初の何行かにデータを説明するコメントが入っているようなファイルも読み込む事ができ

る。例えば以下の usdata03.txt を読み込んでみよう。

(usdata03.txt のデータ)

```
## アメリカの実質個人可処分所得と実質個人消費支出(単位:100 億ドル)
Year    income  consumption
1960       157    143
61         162    146
62         169    153
```

このファイルは 1 行目にデータの説明が入っている。このようなデータを読み込むには

```
> usdata03 =read.table("usdata03.txt", header=T, skip=1)
```

というコマンドを実行する。最後の skip = 1 という引数は 1 行目を読み込まないという事である(ここで読み込まないデータの行数を指定する。例えば最初の 2 行を読み込まないのであれば skip=2 となる)。この場合読み込まれたデータ usdata03 は usdata02 とまったく同じである。

練習問題

ファイル makerdata01.txt, makerdata02.txt, makerdata03.txt にあるデータについて先ほどと同じことをやってみる。