# Use R! **回帰分析編**

2008/05/23(加筆版) 森林環境資源　仁科一哉

Index



## 1. Introduction

＊R を使うメリット

@フリーのソフトであるにも関わらず、あらゆる統計解析が使える！

（重回帰, 主成分回帰, PLS 回帰, 正準相関, リッジ回帰, 非線形回帰 ニューラルネットワーク etc.）

@作図関連も充実している！

信頼区間の描画も可能（エクセルでは当然できない）

@調べないと何を使って良いかわからない

勉強しなければ使えないので、統計学的に不適切な使用を避けられる。

## 2. 相関解析

二変量間の関係性を見る指標として、相関係数と呼ばれるものがある。何の断りもなく“相関係数（Correlation coefficient）”と言っている場合は“ピアソンの積率相関係数(Pearson's product moment correlation)”を意味すると判断して間違いない。注意しなければならないのは、この解析方法は比較する両変数が正規分布していることが前提である事、これは心に留めて頂きたい。まずピアソンの積率相関係数の求め方を示す。

$$r = \frac{Sxy}{Sxx \times Syy}$$

（$r$:相関係数，　$Sxy$=共分散，　$Sxx$=変数 X の標準偏差，　$Syy$=変数 Y の標準偏差）

式の中に標準偏差が含まれることに注目しよう。標準偏差は基本的には正規分布の変数についてのみ、意味のある統計量である（正規変数条件下において平均値±1.96＊標準偏差の範囲内に 95%のデータが含まれる。左右対称の時の分布の広がり度合いを示す）。ということで、このことからも解かると思うが相関係数は、しつこいようだが厳密には二変量正規分布であることが望まれる。例えば特に飛び抜けた値が存在すると、相関係数が不当に高くなってしまう。

前置きが長くなってしまったが、実際に R を使って求めてみよう。

基本のコマンドは “ cor(X, Y, method = c(“pearson”)) ”　　#X, Y は変数を示す

実際に打つコマンドは青、　#以下はコメント　その他の黒字は R の出力を示します

```
x <- rnorm(150)                 #  xに適当な正規変数を 150 個生成して代入
y <- rnorm(150)                 #  yについても同様の操作

x                               #  xに何が含まれているか確認
y                               #  同様
plot(x, y)                      #  プロットして見よう

cor(x, y, method="pearson")     #  相関係数の計算
```

R での相関係数の求めるコマンドは最後の行だけ。実際に今現在使われている統計ソフトに比べて、手数が多いように見えるかもしれませんが、慣れればたいしたことはありません。

続いて相関係数の検定をやってみよう。所謂、無相関の検定の呼ばれるもので相関係数が 0 と有意に違うか否かを示すものです（これ統計的に正確な記述ではありません、注意)。p 値（有意確率）が求められますが、p 値の大小は相関の強さを示す指標ではありません。サンプルサイズに大きく左右されます。相関の強さはあくまで、相関係数で判断するべきです。但し p 値は有意でない場合には、相関関係がないと判断する材料になると思います。p 値を出すのはマナーだからとりあえず明記しておく、程度の感覚で考えといてください。

```
#上の続きでコマンドを打ってみよう
cor.test(x, y, method=”pearson”)                    #  相関係数の p 値を求めるコマンド

        Pearson's product-moment correlation

data:  x and y
t = 2.7246, df = 148, p-value = 0.007214
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.06040149 0.36599028
sample estimates:
      cor
0.2185475
```

2 行目に、t-value、df（Degree of freedom、自由度(サンプル数－2)）における p-value(いわゆる有意確率)が出力される。因みに、無相関の検定の時の t-value は、下記の式で求められる。

$$t = \frac{r \times \sqrt{(n-2)}}{\sqrt{(1-r^2)}}$$

（$t$：t-value,　$r$：相関係数、　$n$：サンプル数）

先の出力で、一番下は相関係数そのものを示しているが、その上には二つの数値が書いてある。それらは相関係数の 95%の信頼区間の下限値と上限値を示したものある（0 が含まれなければ、0 と有意に異なると判断できる）。相関係数においても区間推定がなされている。実際の卒論や修論では情報として使わないと思われるので、あまり気にしなくて良い。（この辺の検定・推定の方法は青木先生の統計学自習ノートに詳しいので、そちらを参照するように）

さて、上のピアソンの相関解析では、二変量正規分布であることが望ましいと述べました。その仮定が成立しない場合はどうしたら良いのか？

ひとつの答えとして、スピアマンの順位相関解析、あるいはケンドールの順位相関解析を使う方法があります。いわゆるノンパラメトリック(非正規分布)解析である。スピアマンの方法は、比率・間隔尺度を順序尺度に落としてから（数値に小さい順に順位をつける）、ピアソンと同じ式で相関係数を計算する。ケンドールの方法に関しては、ここでは省略する（青木先生のサイトに求め方有り）。自分の研究分野では、スピアマンの方がケンドールよりも多く使われている気がする。

では実際に R で求めてみよう

```
#　続きで打っていってください
cor(x, y, method = “spearman”)　　#スピアマンの順位相関を求める
cor(x, y, method = “kendall”)　　#スピアマンの順位相関を求める

#　ピアソンの積率相関係数と同様に p 値なども求められます
cor.test(x, y, method=”spearman”)
cor.test(x, y, method=”kendall”)
```

わざわざ書く程の事でもないのだけれども、method=””の中を変えるだけです。簡単でしょ

これらの分析は、外れ値にも影響を受けにくい、曲線的な単調増加(減少)を相関係数反映させやすいというメリットがある。順序尺度にも使えるのもメリットの一つだろう（あたりまえなのだけど・・・）。デメリットとしては尺度を落とすという行為により、ある程度の情報が失われているということが挙げられるだろうか（抽象的過ぎるかな）。

ここまでは X, Y、二つの変数のみについて取り扱ってきました。R では、比較したい変数が複数ある場合でも、いちいち指定せずに、一度に相関係数を求める事が出来ます（相関行列）。

データセットは所謂、データフレームの形式でなければなりません。データフレームは列(縦)が各変数を示し、行(横)が各サンプルを示します(一行目は見出し)。例えば、R の中に元々実装されている、airquality というデータを見てみよう。

```
airquality　　# 中のデータ確認　(Ozone, Solar.R, Wind, Temp, Month, Day のデータが 153 行)
#相関行列を求めよう
cor( airquality, method=”pearson”)　　# 実はこれだけ！
# ではできないのです・・・。 データの中に欠損値(NA)が含まれているためです (なければ OK)
cor( airquality, method=”pearson”, use=”pairwise”)　　# NA が含まれるサンプル(行)をはじいて計算
```

NA が含まれる場合には、いくつかの対処方法があるので、自分の納得いく方法で対処しよう！

ちなみに相関解析を行う場合は、必ず散布図は見るように！次の回帰分析でも同じ

## 3. 回帰分析

ようやく回帰分析です。まず、基本的なデータ解析の流れを示します。まず得られたデータの分布を“hist”確認しましょう。理論的(あるいは経験的)にどのような分布になるかを知る必要があります。また、どんな解析を行なう時でも、散布図の確認は必ず行なってください。

ここからも、先ほど使用した airquality というデータを使います。このデータは 1973 年 5-7 月のニューヨークのオゾン濃度、太陽放射、風速、温度について測定した結果です。今回はオゾン濃度と風速の関係を見てみよう。

まずはデータの下準備

```
airquality                  #Airquality というデータセットを見てみよう
attach(airquality)          #Airquality を項目(列)毎に分割。項目毎に指定できるようになる
names(airquality)           #データセットの項目を見れるコマンド
```

次に実際に回帰させよう

R におけるモデルの記述方法は、例えば y ＝ βx ＋α＋ε（εは誤差項を示す）という一次式を考えている場合には、y~x と書く（~が＝の意味）。　（モデルの中に 0 を入れると切片 0 にできる　Y~X+0）

```
#実際の回帰分析の手順(Wind と Ozone の関係を見て見よう)
plot(Wind, Ozone)                  # plot(X 軸, Y 軸)のグラフを書く
result <- lm(Ozone~Wind)           #  回帰分析を行って、結果を result に入れた
abline(result)                     #  回帰直線を図に描画するコマンド
result                             #  結果を表示する
Call:
lm(formula = Ozone ~ Wind)
Coefficients:
(Intercept)          Wind          #  切片と回帰係数が表示される　(Ozone=-5.551Wind+96.873)
      96.873        -5.551
```

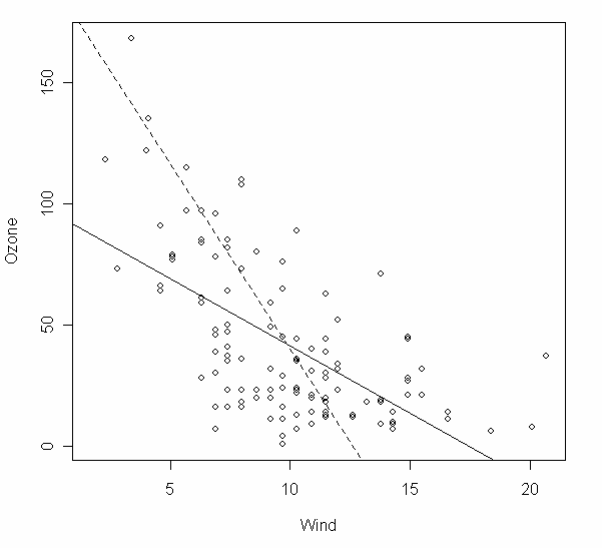
Summary という関数を使用すると、もっと詳細な情報を得る事ができる。

```
summary(result)
Call: lm(formula = Ozone ~ Wind)
Residuals:                                          #残差の分布(最大(小)・第 1(3)四分位数・中央値)
    Min     1Q     Median       3Q      Max
-51.572 -18.854   -4.868   15.234   90.000
Coefficients:                                       #  切片・回帰係数、およびその標準偏差
             Estimate     Std. Error     t value Pr(>|t|)
(Intercept)  96.8729       7.2387      13.38    < 2e-16 ***        #切片が 0 と有意に違うか否か
Wind          -5.5509       0.6904      -8.04     9.27e-13 ***      #係数が 0 と有意に違うか否か
S ignif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.47 on 114 degrees of freedom
Multiple R-Squared: 0.3619,        Adjusted R-squared: 0.3563      #R2 とajsR2 修正済み決定係数
F-statistic: 64.64 on 1 and 114 DF,    p-value: 9.272e-13          #ここの p 値は回帰が有意か否か
```

次に回帰直線の信頼区間の描画について

```
#信頼区間の書き方　(続きで打っていってください)
new <- data.frame(Wind = seq(min(Wind), max(Wind), by = 0.1))   # まず描画範囲の指定をする
cline <-predict(result, new, interval="confidence")             # 指定した範囲の信頼区間を求める
cline
plot(Wind, Ozone, xlab = "Wind", ylab = "Ozone")                # 元データのプロット
matplot(new, cline, lty=c(1,2,2), type="l", add=T)              # 信頼区間を曲線として描画
detach(airquality)                                  # 解析の最後にこれは絶対に忘れずやって下さい
```




・回帰分析における注意点

誤差項において、正規性、等分散性が要求されている(回帰診断で確認)。独立変数(X)については正規性が要求されていないが、一点のみ非常に高い値(外れ値)があるような偏った分布である場合には注意が必要であろう。

回帰分析は、原因と結果がはっきりしているときにしか使えない、ということに注意して欲しい。回帰分析では、原因が必ず独立変数(x)になるようにしなければならない。例えば、Ozone と Wind の関係ならば、“強い風が吹くことによって大気中の Ozone が、あるいはその前駆物質(NO)が吹き飛ばれて濃度が低下した”というようなメカニズムが存在していると考えられる。

何故そのようなことを気にしなければならないかというと、回帰直線は y 軸方向の誤差のみを最小にするように求められており、x 軸の誤差は考慮されていないためである。上の図では、Wind を独立変数として Ozone に回帰させた場合と、Ozone を独立変数にして Wind に回帰した回帰直線（点線）に 2 つをプロットしてみた。かなり違う事が解かると思う。

（なお因果関係のはっきりしない場合の回帰は主成分回帰(Major axis regression)などを使うことをお薦めする。いわゆるⅡ型の回帰とよばれるもの）

## 4.　重回帰分析

ようやく重回帰分析に入ります。先の単回帰では$R^2$が 0.36 と低く、説明力が弱い（ちなみに$R^2$は全変動の何%を回帰によって説明できるのかを示す)。充分な決定係数が得られない理由としては、いくつかの原因が考えられますが、ここでは仮に“他にも影響する要因があるため”と考える事にします。そこでここでは、温度をもうひとつの独立変数(説明変数とも言う)を入れて回帰式を作ってみよう。

```
attach(airquality)          #例のごとく、項目(列)毎に分割　→最後には detach(airquality)を忘れずに
result <- lm(Ozone~Wind+Temp)
result
lm(formula = Ozone ~ Wind + Temp)
Coefficients:                                   #切片および従属変数の回帰係数
(Intercept)         Wind         Temp
     -71.033       -3.055        1.840
```

重回帰分析も lm という関数(Linear Model の略)でできます。というよりもむしろ単回帰は従属変数の特殊な形である。結果の summary の中身、その解釈についてもほぼ同様である。

```
summary(result)
Call:
lm(formula = Ozone ~ Wind + Temp)

Residuals:
     Min       1Q   Median       3Q      Max
-41.251  -13.695   -2.856   11.390   100.367

Coefficients:
             Estimate   Std. Error  t value   Pr(>|t|)          #切片・偏回帰係数
(Intercept)  -71.0332     23.5780   -3.013    0.00320 **
Wind          -3.0555      0.6633   -4.607    1.08e-05 ***
Temp           1.8402      0.2500    7.362    3.15e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.85 on 113 degrees of freedom
Multiple R-Squared: 0.5687,      Adjusted R-squared: 0.5611
F-statistic:  74.5 on 2 and 113 DF,  p-value: < 2.2e-16
```

結果として“ Ozone = 1.84 * Temp – 3.06*Wind – 71.0 ”で$R^2$=0.57 という回帰式を得ることができた。

・重回帰分析の注意点！

重回帰分析は便利な反面、単回帰モデルにはなかった制約がある。その分、要求される知識も増える。そのため不用意な使用はさせるべきである。もし今後重回帰を実際に使いたいという人たちは、巻末のせた参考文献・参考書などを利用して各自で勉強して頂きたい。以下に、いくつかの注意点を紹介する。かく云う自分も卒論時にはこのような事柄を無視して重回帰を利用していた（というよりも、むしろ今でもしっかり理解しているか怪しい・・・）。

＊多重共線性（Multi-co linearity　いわゆるマルチコ）　-線形モデルの破壊者-

重回帰分析には独立変数が正規分布していうという条件に加え、従属変数が各々独立である必要がある。理想的には、各変数間にまったく相関がないということが望まれる(相関係数 0)。例えば、Wind と Temp の相関係数が 1 であった場合、単純に計算ができない(計算式の中に、分母に 0 でてきてしまい、解が導出できない)。相関が非常に高いと、分母に 0 に近い値が現われ、非常大きな値を数値計算に使わなければならなくなり、計算結果が不安定になり、結果として得られた回帰モデルも非常に脆弱なものになる。

対策としては、変数同士に非常に高い相関を持つものがあったとしたら、どちらかの変数を除外する

という方法がある。しかし単相関が極端に高くない場合にも多重共線性問題は発生する場合がある．複数の変数との一次従属が成立している場合で、従属変数が 3 つ以上ある場合には単相関だけでなく重相関もチェックしなければならない．その指標が VIF ( Variance Inflation factor ：分散拡大係数)で、VIF が 10 になると多重共線性がおきているという指標になる（あくまで指標）。

$$VIF = \frac{1}{1 - R_j^2}$$

$R_j^2$; $X_j$を従属変数(それ以外N-1 個を独立変数にして回帰)にしたときの決定係数

R には“ DAAG ”というパッケージに VIF を計算してくれる関数が用意されている。R のメニューバーのパッケージ、インストールから“ DAAG ”をインストールし、使うときはパッケージの読み込みを行ってから行う。

```
#上記の手順後
attach(airquality)
result <- lm(Ozone~Wind+Temp+Solar.R)          #  従属変数が 3 つの回帰モデル
vif(result)                                    # VIF チェックの関数
    Wind     Temp    Solar.R
 1.3291   1.4314   1.0953
detach(airquality)                             #detach は忘れずに！
```

今回はいずれも 10 を越えていないので、多重共線性の心配は必要ない。VIF は計算も簡単なので、わざわざ関数を使う必要もないかと思われるが、変数が多くなったときには便利である。ただし、この関数、どの組み合わせでマルチコがおきているのかはわからない。結局は一つ一つチェックするしかない・・・

＊交互作用と言う概念

二元配置の分散分析を行う際にはよく出てくる単語ですが、重回帰分析にもあります。交互作用とは、「ある予測変数の値によって、他の予測変数の効果が変化する」ことを指します。交互作用があるのか否かは、割と簡単に確かめる事ができます。Ozone = α*Wind + β*Temp + γ + e というモデルを考えているならば、新たに Wind×Temp と言う変数を加えてやれば良いのです。その結果として、Wind×Temp が有意であるとされれば、交互作用があるということになります。

なぜ、交互作用があるといえるかは下の式を見れば一目瞭然です。

Ozone = Wind + Temp + Wind×Temp = (1 + Temp)Wind + Temp　（係数等は省略）

例えば、Wind で式をまとめて見ると、Temp の如何によっては、Wind は正にも負にも働く可能性があることがわかると思います。有意でなければ、除外すればいいだけです。

但しひとつ注意点があります、単純に Wind×Temp を回帰式に入れてしまうと、先ほど紹介したマルチコが起きる可能性が高くなります。これを避けるために、Wind･Temp に対して中心化を行ってから、掛け合わせるという作業を行います。中心化というのは、平均値を引いた変数を作る事です（標準化を行っても可）。

ということで実際に、R でやってみよう

```
attach(airquality)
Wind2 <- (Wind – ave(Wind))                    # 中心化の作業(標準化はさらに標準偏差で割る)
Temp2 <- (Temp – ave(Temp))
result <- lm(Ozone~Wind+Temp+Wind2*Temp2)
summary(result)                                # 一部省略

Call:
lm(formula = Ozone ~ Wind + Temp + Wind2 * Temp2)

Coefficients: (2 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)
(Intercept) -74.86780    22.06961   -3.392 0.000959 ***
Wind         -3.10381     0.62038   -5.003 2.11e-06 ***
Temp          1.84614     0.23377    7.897 2.13e-12 ***
Wind2              NA          NA       NA       NA
Temp2              NA          NA       NA       NA
Wind2:Temp2  -0.22391     0.05399   -4.147 6.57e-05 ***                  # 有意 ⇒交互作用あり
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.44 on 112 degrees of freedom
Multiple R-Squared: 0.6261,     Adjusted R-squared: 0.6161
F-statistic: 62.52 on 3 and 112 DF,  p-value: < 2.2e-16
detach(airquality)
```

この 2 つの変数を使ったモデルには交互作用があるみたいです。交互作用の解釈には符号に注意して行ってください。

なお、変数が増えれば増えるほど考えなければならない交互作用は飛躍的に増えていきます。重回帰分析、簡単なようで意外と難しい。

＊変数選択　－より良いモデルとは？－　　（玉木君が詳しい）

良いモデルの条件としては、決定係数($R^2$)が高いというのはひとつの重要な基準である。しかしながら重回帰では変数が増えれば増えるほど、たとえそれが実質意味のない変数であっても、決定係数は上がっていくという性質を持つ。極端な話をすれば、サンプル数以上にパラメーターを用意すれば、決定係数 1.00 の回帰式ができてしまう(Over fittingという)。

そこで変数選択という作業を行う必要がある。変数選択には、いくつかの方法・基準が存在する。例えば、F 値を利用するもの、尤度比検定、AIC を基準にする、などの方法があります。今回は AIC を利用した変数選択を行います。その他のものについては、各自で調べて見てください。

（最尤推定基本としているため、AIC は一般化線形回帰(GLM)を使う際にも利用できる）

AIC(Akaike's Information Criteria)は下記の式で求められる

$$AIC = 2*p - 2(-1/2*n*\log(2\pi\sigma^2))$$

p；パラメーター数　　()の中；尤度関数の最大値　$\sigma^2=\left(\sum_{i=1}^{n} e_i^2\right)/n$　→残差の二乗和/サンプル数

モデル選択では AIC の小さいものを選択する。すなわち、パラメーターが少なければ少ないほど良く(左の項)、かつ当てはまりが良いものがよい(右の項)、というのが AIC の判断基準である。シンプルかつ、当てはまりが良いというのは、決して決定係数が一番高いものを選んでいるわけではないことに注意。

独立変数の数pが大きくなると，モデルの総数が$2^p$個と非常に多くなってしまい、個々のモデルのAICをいちいち計算するのは大変である。しかし“ wle ”というパッケージには自動的に総当りで計算してくれ、AICの小さい順にモデルを示してくれる関数‘ mle.aic ’が用意されている。実際に使ってみよう。

なお、下平さんHPには、AIC最小のモデルを見つける方法として、総当り法よりも計算の負荷の少ない、逐次選択法、分枝限定法の自作関数が公開されている。

```
#今回は Wind、Temp、2 つの変数、および相互作用項を入れたモデルを考える
attach(airquality)
Wind2 <- (Wind – ave(Wind))                 # 中心化の作業（標準化はさらに分散で割る）
Temp2 <- (Temp – ave(Temp))
WinTem <- (Wind2 * Temp2)                   # 今回は変則的だが交互作用項を先に作って解析
result <- lm(Ozone~Wind+Temp+WinTem)        # ⇒さっきの方法で解析すると何故かフリーズする･･･
select <- mle.aic(result)                           # select に結果を入れる
summary(select)                                     # 結果の要約表示（直接 select でも良い）
Call:
mle.aic(formula = Ozone ~ Wind + Temp + WinTem)
Akaike Information Criterion (AIC):
      (Intercept) Wind Temp WinTem   aic            # 変数のあり(1)・なし(0)を示す
 [1,]           1    1    1      1  1033
 [2,]           0    1    1      1  1043
 [3,]           1    1    1      0  1048
 [4,]           1    0    1      1  1056
 [5,]           0    1    1      0  1057
 [6,]           1    0    1      0  1071
 [7,]           1    1    0      1  1094
 [8,]           1    1    0      0  1108
 [9,]           0    0    1      1  1147
[10,]           0    0    1      0  1156
---------------------------------------------------------------------  # 途中省略
Printed the first   15   best models
```

今回は交互作用項を含めてすべての変数を用いた式が、一番低い AIC を示したので、変数は減らす必要はないという結果だった。

## 5. 一般化線形モデル (GLM, Generalized Linear Model)

最近至る所で、一般化線形回帰と言う単語を見ると思います。今回紹介した、回帰・重回帰の話はすべて、この GLM という枠組みに入ります。また今まで使ってきた“ lm ”という関数はすべて、“ glm ”という関数に置き換えることができます。

この‘ 一般化 ’というのは、何を指して一般化なのか？というと、大きな所では、様々な分布関数が使えるようになります。“ lm ”では変数が正規分布であることが大前提でしたが、GLM では正規分布に加えて、二項分布(Binomial)・ポワソン分布(Poisson)・ガンマ分布(Gamma)など、様々な確率分布を扱う事ができるようになる。また、同時に誤差項に関しても、正規分布以外のものが使えるようになります(これは意外と重要な性質)。加えて、間隔・比率といった量的変数だけでなく、カテゴリー変数も扱えるようになります。この変数の組み合わせを考えると、実は以下の解析がすべて同じ枠組み(アルゴリズム)で計算される様です。

| 分析名 | 従属変数 ($Y$) | 独立変数 ($X$) |
|---|---|---|
| $t$ 検定（注 1） | 量的変数 1 つ | ２値変数 1 つ |
| 一元配置分散分析 | 量的変数 1 つ | カテゴリ変数 1 つ |
| 多元配置分散分析（注 2） | 量的変数 1 つ | カテゴリ変数複数 |
| （単）回帰分析 | 量的変数 1 つ | 量的変数 1 つ |
| 重回帰分析 | 量的変数 1 つ | 量的変数複数（注 3） |
| 共分散分析 | 量的変数 1 つ | （注 4） |
| ロジスティック回帰分析 | ２値変数 1 つ | ２値変数，カテゴリ変数，量的変数複数 |
| 正準相関分析 | 量的変数複数 | 量的変数複数 |

＊中澤先生の HP より抜粋

GLM(+GLMM)についての詳細は松下@生態から！（基本は lm 関数の使い方と同じです）

## 6. 非線形回帰 “ nls ←Non-linear Least Square ”

非線形回帰とは、その名の通り線形(結合)ではない回帰という意味です。単純なものでは、二次式から指数関数などエクセルで計算できるものから、任意の関数に対する回帰(ロジスティックモデル等の漸近回帰モデル)、更にスプライン補間などの平滑化なども含まれます。その他には、(一般化)加法回帰モデル(GAM)やニューラルネットワーク回帰など、つまりなんでもありです(この辺は全く知りません)。

今回は任意の非線形関数を当てはめる“ nls ”関数を用いて、最小二乗法によりロジスティック曲線のあてはめを行ってみたいと思います。“ nls ”の使い方は基本的には“ lm ”と同じです。違いは、より具体的に式を指定する事と、各パラメーターの初期値を設定する必要があるという 2 点です。

（実は R には、ロジスティックモデルや、ゴンペルツの成長モデル、ワイブル成長モデル、二重指数モデル等の有名な漸近回帰モデルは“ nls ”関数とは別に、“ SSlogis ”などの関数が用意されている。この関数は式を明示的に指定しなくても良い事と、自己開始モデルと呼ばれ初期値を設定しなくても良い(らしい)。）

ここで、簡単にロジスティックモデル(以下の式(イ))について簡単に説明をしておきます。

$$f(x_i) = \frac{a}{1 + b * e^{-cx_i}} + \varepsilon_i \qquad (イ)$$

a, b, c: 回帰係数(推定するパラメーター)　c >0,　ε: 誤差項 ←平均値 0 の正規分布です、重要！

ロジスティックモデルは個体(群)の成長モデルとして、よく利用されています。この関数の肝は、環境収容力と言う概念が入っている事である。これはマルサス的成長(指数曲線で際限のない増加)とは、大きく異なる点である。

そもそもロジスティックモデルは次式(ロ)の微分方程式の解として得られる式である。

$$\frac{dy}{dx} = \beta\left(\frac{\alpha}{\beta} - y\right)y \qquad (ロ)$$

a= $\alpha/\beta$,　b=(α/β)*f(0)+1/β,　c=α

この式は、ロジスティックモデルの増加率を示すが、y(成長量や個体数)が$\alpha/\beta$に達した場合、成長量が 0 になることが解かる。つまり、この値が環境収容力を表しており、この数値よりも個体重あるいは個体数を超える事はない。また y の増加に伴い、環境収容能力に値が近づくにつれ、成長速度が減衰する事も表しています。

それでは実際にやってみよう

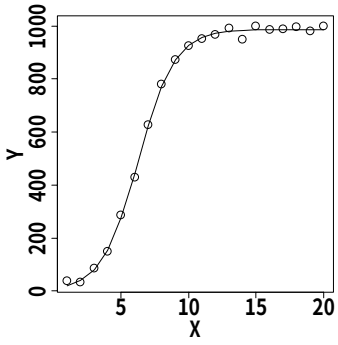
```
#ロジスティック回帰の当てはめ！
X <- (1:20)
Y <- c(3, 16, 54, 139, 263, 420, 611, 750, 860, 900, 940, 950, 960, 930, 980, 970, 989, 980, 975, 980)
Y <- Y + rnorm(20, 40, 20)                          #おまじない、気にするな

plot(Y~X)                                           #とりあえずプロット（plot(X, Y)でも可）
#以下非線形回帰。式に注目。初期値は適当に決まるしかない・・・・
result <- nls (Y ~ a / (1+b*exp(-c*X)), start = c(a=1000, b=100, c=1), trace=T)
summary(result)
Formula: Y ~ a/(1 + b * exp(-c * X))

Parameters:
     Estimate Std. Error t value Pr(>|t|)
a 986.77806      4.18655 235.702    < 2e-16 ***
b 103.95239     12.46723     8.338 2.07e-07 ***
c     0.74022      0.01929    38.380    < 2e-16 ***          #パラメータの推定値とその標準誤差
---
Signif. codes:    0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.75 on 17 degrees of freedom          #因みに AIC の求め方は AIC(result)

lines(X, fitted(result))                            #最後に回帰曲線の描画
```




初期値の設定には、王道はないといわれています。今回のように、割と単純で収束値が明らかな場合は、大体の値で決めることが出来ますが(1000 辺りで収束しているので、まず a=1000 とする)、多くの場合は経験を積むしかないというのが唯一のアドバイスです。

最後に、今回は**あえてロジスティックモデル**を対象としました。ロジスティック曲線は、生態学的な

学問に留まらず、非常に様々な分野で活用される、重要な非線形回帰式である事も選択した理由のひとつです。しかし選択理由はそれだけでなく、続いて松下@生態が紹介してくれる“ glm ”関数を用いたロジスティック回帰を意識してのことです。

今回、回帰式の最後には必ず誤差項を示してきました。線形回帰でも非線形回帰でも、誤差項は平均値 0 の正規分布です。実はこれは単なるおまけではなく、非常に重要で厳しい制約なのです。しかもこの誤差項は、任意の X(あるいは複数の従属変数)に対して、分散が同様である必要があります。前のページに記載した、ロジスティック曲線で感覚的に説明すると、曲線からの周りの実測値の外れ方が、どの X 時においても同じぐらいである(はず・・・)必要があります(等分散性)。はっきり言って、そんなデータ、フィールドにはなかなか落ちてないです。

（正しい統計的な用語ではないと思います、この辺の文は）

一般化線形モデル”glm”の利点は、そうした‘ 誤差項＝正規分布 ’の制約を、明示的に指定することによって、回避あるいは緩和する事ができるのです。これは GLM あるいは GLMM を利用する強みの一つです。さらに GLM よりも、この辺の制約を自由に扱うには、階層ベイズモデルを扱う必要があるでしょう。しかし、残念ながら自分はまだまだ、階層ベイズの修とっくには程遠いのが現状です。

## 7. R 習得へのアドバイス

はっきり言って、自分もRを使えているとはとても言えません。ということで森林環境の林先輩から言われていたことを、そのまま書くこととします。

＊まず第一にどんな関数でも”help”を必ず見ること!

＊次に help の最後に載っている example(例)を一つ一つコマンドをうってなぞること！

実は example(関数名)で help にある例はできてしまうのだが、やっぱりひとつひとつコマンドを確かめていった方が、飲み込みが早いと思います。

玉木君の挙げてくれた資料集は入門書として最適だと思います、一冊くらいは手元にあると便利です。後は、多くの人達が無償で公開してくれている HP を参考にするのが良いでしょう。

統計に関してのアドバイスとして、まず統計解析は非常に扱いづらいものであると言う認識が重要だと思います。昨今は、PC の性能もさることながら、ソフトも進化し、労する事もなく、クリックひとつで統計解析をすることが出来る時代になりました。それだけに、統計のブラックボックス化も進み、誤用も増えているようです。とにかく、今回のゼミで、統計を使うには多くの留意点があり、簡単にできるものではないと思って頂けたら、とりあえず成功であると思っております。

## 8. 参考・引用文献


The R-tips　船尾暢男,2005, 9-ten 社　　(Web にはさらに改定が加わった全文が公開されている)

Statistics –an introduction using R　M.J. Crawley, 2005, John Wiley & Sons

生物学を学ぶ人のための統計のはなし‐きみにもだせる有意差‐, 粕谷英一, 文一総合出版

中澤港先生のHP　http://phi.med.gunma-u.ac.jp/index.html　　#かなり参考にしました

青木繁伸先生のHP　http://aoki2.si.gunma-u.ac.jp/　　#言わずと知れた統計界の神様

下平英寿先生の HP　http://www.is.titech.ac.jp/~shimo/index-j.html

前田和寛先生のHP　http://home.hiroshima-u.ac.jp/kazu711/stat/HP_MR_0.html

補足：１．主成分回帰について

原因と結果、因果関係の主従がはっきりしない場合の回帰を行なう場合は、主成分回帰を用いる。通常の回帰式と違い x，y は完全に対称的（式を変形するだけで y= にも x= にもなる）。例えば、生態学の問題で良く出てくる相対成長(アロメトリー)なんかでは良く使われる。というか使わないといけない。自作でも、たいして求めるのは難しくないが、パッケージの Smatr を使えば、信頼区間などを出してくれるし、任意の傾きとの検定も出来る関数が用意されているので便利。

```
plot(airquality$Wind, airquality$Ozone, xlab = "Wind", ylab = "Ozone")
result <- lm(airquality$Ozone~airquality$Wind)            #普通に回帰させた場合
abline(result)
result2 <- lm(airquality$Wind~airquality$Ozone)                    #逆に回帰させた場合
abline(-result2$coef[1]/result2$coef[2], 1/result2$coef[2], lty=2)    #上の線の図示（点線）

#SMA 行なう関数を自作する場合
SMA <- function(x, y) {                                            # 2 つの変数
        dt <- data.frame(x, y)
        dt2 <- na.omit(dt)
        x1 <- dt2$x
        y1 <- dt2$y
        slope <- sign(cor(x1, y1))*sqrt(var(y1)/var(x1))           # 傾きを求める
        intercept <- mean(y1)-slope*mean(x1)                       # 切片を求める
        return(list(Slope=slope, Intercept=intercept))
}

result3 <- SMA(airquality$Wind, airquality$Ozone)                  #主成分回帰(SMA)の計算
result3                                                                    #結果の表示
abline(result3$Intercept, result3$Slope, col="red")                        #結果の描画（赤線）

#パッケージ"smart"を使う場合
library(smatr)                                                     #パッケージの呼び出し
res <- line.cis(airquality$Ozone, airquality$Wind)                 #主成分回帰(SMA)の計算
res                                                                #結果の表示
                coef(SMA)     lower limit     upper limit          #係数と 95%信頼区間(上限)が表示
elevation 133.134048   118.590129     147.67797
slope       -9.227753     -7.960876     -10.69624
#例えば、傾き-5.55 との有意差検定
slope.test(airquality$Ozone, airquality$Wind, test.value = -5.55)
$r
[1] 0.5532898
$p
[1] 1.186690e-10              #有意確率
$test.value
[1] -5.55                                  #以下省略
```