

環境科学と生態学のための R 統計

Song S Qian (著)・大森浩二・井上幹生・畑 啓生 (監訳)

共立出版 2011 年 発行

<R プログラム実行例について： Windows XP の例>

<R-2.11.1 の導入>

本ファイルは、翻訳本のコードナンバーをふっている R コード を実際に実行するためのものである。本書で使用されるデータファイル集およびパッケージ集、また、R-2.11.1 版システムは別に用意している。また、この解説では R-2.11.1 版上で R コードを実行している。この翻訳本の R コードを実行しやすいようにパッケージなども取り込んだ形となっている。ここに用意している R システム (R-2.11.1 版) を以下のように各自のコンピュータへコピーすると直ちに R コードを実行できるようになっている。

・ R-2.11.1 の入った R フォルダをそのまま各自のコンピュータへコピーして、R-2.11.1 フォルダの中の bin フォルダ中にある Rgui.exe をダブルクリックすると翻訳本の R コードを実行するために必要なパッケージ等がインストールされた R-2.11.1 を立ち上げることができる。R-2.11.1 を R のホームページからダウンロードした場合は、インストールプログラムを実行した後、c ドライブの Program Files にある R フォルダ内の R-2.11.1 フォルダにある library フォルダをこちらが提供している R-2.11.1 のフォルダの library フォルダと入れ替えることで翻訳本の R コードを実行するために必要なパッケージを利用できるようになる。

R システムのバージョンが異なるとパッケージのバージョンとの整合性がとれなくなる。その場合は、R のホームページからバージョンのあったパッケージをダウンロードする必要がある。

データファイルは、翻訳本のなかで入力が必要になった段階で、それぞれ入力するようになっている。翻訳本に沿って演習されることをお奨めする。

<使用方法>

先ず、翻訳本の第 2 章に沿って演習をしていただきたい。R Commander とともに R プ

プログラムが順調に動けば、第 3 章以降は以下に述べるコピー&ペースト法で実行可能となる（第 2 章も R コード 2.20 まではコピー&ペースト法で実行可能である）。ただし、実行のためには、データを取り込む必要があるし、場合によっては、パッケージを新たにインストールする必要もある。

<コピー&ペースト法>

・ R エディタ使用の場合：

- (1) R システムを立ち上げると RGui 上に R コンソール画面が開かれ、R のプロンプトである “>” がでて入力待ち状態となる。
- (2) ファイル中のコードのついた R プログラムを一つ選び、R のプロンプトである “>” を除いて残りの R コードを全てコピーする。
- (3) RGui 上の “ファイル” をクリックして “新しいスクリプト” を選択する。
- (4) R エディタの画面が開かれる。そこに選んだ R コードをペーストし、選択しておく。
- (5) RGui 上の “編集” をクリックして “カーソル行または選択中の R コードを実行” を選択する
- (6) R コンソール画面上に R の実行結果が表示される。

・ R Commander 使用の場合：

- (1) R システムを立ち上げると RGui 上に R コンソール画面が開かれ、R のプロンプトである “>” がでて入力待ち状態となる。
- (2) “パッケージの読み込み” を選び、選択画面から “Rcmdr” を選択し R Commander の入出力画面を立ち上げる。
- (3) ファイル中のコードのついた R プログラムを一つ選び、R のプロンプトである “>” を除いて残りの R コードを全てコピーする。
- (3) R Commander のスクリプトウィンドウにペーストし、選択状態にしておく。
- (5) “実行” ボタンを選択する
- (6) 出力ウィンドウ上に R の実行結果が表示される。

<実行上の注意事項>

・ 本来 2 つの独立な R コード文となるものを一行にペーストしてしまい、エラーが出ることが多い。“予想外のシンボルです” というエラーメッセージがでたら、R コード文の改行がきちり行われているか確認すること。

<データ入力法>

幾つかの方法があるが、最も簡単な R Commander のデータ入力を使用した方法をここ

では示しておく.

- (1) RGui 上に R Commander が立ち上がった状態にしておく.
 - (2) Windows のメモ帳を立ち上げて, 入力したいデータセットを取り込んで全てを選択し, コピーしてクリップ上に挙げておく.
 - (3) R Commander の “データ “を選択し,” データセットのインポート “,” ファイルまたはクリップボードから . . . “と選択すると入力ウィンドウが表示されるので, (R の中で使用する) データセット名を入力し,” クリップボード “,” タブ “にチェックを入れて, OK ボタンをおすと R システム内にデータが取り込まれる.
- ・自分で作成したデータを入力する場合は, エクセルの標準ファイルで作成後, テキストファイルとして, または, csv ファイルとして保存しておく必要がある.

<パッケージのインストール法>

- ・ R Commander の “ツール” で, “パッケージのロード “を選択し, 表示されているウィンドウの中から, (例えば) arm を選択し, OK ボタンをクリックする. そうするとパッケージ arm が利用できるようになる.

<連絡先>

うまく R プログラムが動かない等, 問題がありましたら, 以下のメールまでご連絡ください.

愛媛大学沿岸環境科学研究センター

大森浩二

ohmori@sci.ehime-u.ac.jp

<R プログラム>

第2章

<15>

```
R version 2.9.0 (2009-04-17)
Copyright (C) 2009 ISBN 3-900051-07-0
The R Foundation for Statistical Computing
and comes with ABSOLUTELY NO WARRANTY.
R is free software You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors. Type 'contributors()' for
more information and 'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for
an HTML browser interface to help. Type 'q()' to quit R.
[Previously saved workspace restored] >
```

<16>

R code ##### 2.1

```
>4+8*9
[1] 76
```

R code ##### 2.2

```
> a <- 4 + 8 * 9
```

R code ##### 2.3

```
>a
[1] 76
```

<17>

R code ##### 2.4

```
> hi <- "hello, world"
> hi
```

```
[1] "hello, world"
```

R code #### 2.5

```
>3>4
```

R code #### 2.6

```
>3>4
```

```
[1] FALSE
```

```
>3<5
```

```
[1] TRUE
```

R code #### 2.7

```
> Logic <- 3 < 5
```

```
> Logic
```

```
[1] TRUE
```

R code #### 2.8

```
> mode(hi)
```

```
[1] "character"
```

```
<18>
```

R code #### 2.9

```
> (3<4) + (3>4)
```

```
[1] 1
```

R code #### 2.10

```
> TP <- c(8.91, 4.76, 10.30, 2.32, 12.47, 4.49, 3.11, 9.61, 6.35, 5.84, 3.30,  
12.38, 8.99, 7.79, 7.58, 6.70, 8.13, 5.47, 5.27, 3.52)
```

```
> TP
```

```
[1] 8.91 4.76 10.30 2.32 12.47 4.49 3.11 9.61 6.35
```

```
[10] 5.84 3.30 12.38 8.99 7.79 7.58 6.70 8.13 5.47
```

```
[19] 5.27 3.52
```

```
> violations <- TP > 10
```

```
> violations
[1] FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE
[11] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
> mean(violations)
[1] 0.15
```

<19>

R code ##### 2.11

```
> sum(TP)
[1] 137.29
```

R code ##### 2.12

```
> length(TP)
[1] 20
```

R code ##### 2.13

```
> sum(TP)/length(TP)
```

R code ##### 2.14

```
> my.mean <- function(x){total <- sum(x); n <- length(x); total/n}
```

R code ##### 2.15

```
> my.mean(x=TP)
[1] 6.9
```

<20>

R code ##### 2.16

```
> mean(TP)
[1] 6.9
```

R code ##### 2.17

```
> help(mean)
```

R code ##### 2.18

```
> example(mean)
mean> x <- c(0:10, 50)
mean> xm <- mean(x)
mean> c(xm, mean(x, trim = 0.10))
[1] 8.75 5.50
mean> mean(USArrests, trim = 0.2)
Murder Assault UrbanPop Rape
 7.42 167.60 66.20 20.16
```

<21>

R code ##### 2.19

```
> mean(x, na.rm=T)
```

R code ##### 2.20

```
> my.mean <- function(x)
  return(mean(x, na.rm=T))
```

<22>

- ・これ以降はコピー&ペーストの際の便宜を考えてRプロンプト'>'を省略する.

R code ##### 2.21

```
library(Rcmdr)
```

<23>

R code ##### 2.22

```
qnorm(c(.9), mean=2, sd=0.75, lower.tail=TRUE)
```

<24>

R code #### 2.23

```
set.seed(123)
```

<25>

R code #### 2.24

```
(obs1>3) + (obs2>3) + (obs3>3) + (obs4>3) + (obs5>3) +  
(obs6>3) + (obs7>3) + (obs8>3) + (obs9>3) + (obs10>3)
```

R code #### 2.25

```
as.numeric(violations > 1)
```

<26>

R code #### 2.26

```
set.seed(123)  
Norm1 <- as.data.frame(matrix(rnorm(10*10, mean=2, sd=0.75),ncol=10))  
rownames(Norm1) <- paste("sample", 1:10, sep="")  
colnames(Norm1) <- paste("obs", 1:10, sep="")  
Norm1$violations <- with(Norm1,  
  (obs1>3) + (obs2>3) + (obs3>3) + (obs4>3) + (obs5>3) +  
  (obs6>3) + (obs7>3) + (obs8>3) + (obs9>3) + (obs10>3))  
Norm1$impaired <- with(Norm1, as.numeric(violations>1))  
numSummary(Norm1[, "impaired"], statistics=c("mean", "sd", "quantiles"))
```

<27>

R code #### 2.27

```
violation <- numeric(length=10000) ##create a numeric vector  
for (i in 1:10000) {violation[i] <- sum( rnorm(10, 2, 0.75)>3 ) > 1}  
print(mean(violation))
```


R code #### 2.28

```
Norm.data <- matrix(rnorm(10*10000, 2, 0.75), ncol=10)
mean(apply(X=Norm.data, MARGIN=1, FUN=function(x) return(sum(x>3)>1)))
```

<28>

R code #### 2.29

```
Norm2 <- as.data.frame(matrix(rnorm(10000*10, mean=2, sd=1),
                             ncol=10))
rownames(Norm2) <- paste("sample", 1:10000, sep="")
colnames(Norm2) <- paste("obs", 1:10, sep="")
Norm2$violations <- with(Norm2,
                        (obs1>3) + (obs2>3) + (obs3>3) + (obs4>3) + (obs5>3) +
                        (obs6>3) + (obs7>3) + (obs8>3) + (obs9>3) + (obs10>3))
Norm2$comply <- with(Norm2, as.numeric(violations<2))
numSummary(Norm2[, "comply"], statistics=c("mean", "sd", "quantiles"))
```

R code #### 2.30

```
Norm3 <- as.data.frame(matrix(rnorm(10000*100, mean=2, sd=0.75), ncol=100))
rownames(Norm3) <- paste("sample", 1:10000, sep="")
colnames(Norm3) <- paste("obs", 1:100, sep="")
Norm3$violations <- apply(X=Norm3, MARGIN=1,
                        FUN=function(x){return(sum(x>3))})
Norm3$impaired <- with(Norm3, as.numeric(violations>10))
numSummary(Norm3[, "impaired"], statistics=c("mean", "sd", "quantiles"))
```

<29>

R code #### 2.31

```
Norm4 <- as.data.frame(matrix(rnorm(10000*100, mean=2, sd=1), ncol=100))
rownames(Norm4) <- paste("sample", 1:10000, sep="")
```

```
colnames(Norm4) <- paste("obs", 1:100, sep="")
Norm4$violations <- apply(X=Norm4, MARGIN=1,
FUN=function(x){return(sum(x>3))})
Norm4$comply <- with(Norm4, as.numeric(violations <=10))
numSummary(Norm4[, "comply"], statistics=c("mean", "sd", "quantiles"))
```

第3章

<32>

R code ##### 3.1

```
dnorm(0.5, mean=0, sd=1)
[1] 0.3520653
```

R code ##### 3.2

```
pnorm(0.5, mean=0, sd=1)
[1] 0.691462
```

R code ##### 3.3

```
1 - pnorm(0.5, mean=0, sd=1)
[1] 0.308538
```

R code ##### 3.4-1

```
qnorm(0.25, mean=0, sd=1)
[1] -0.67449
```

R code ##### 3.4-2

```
qnorm(0.05, mean=0, sd=1)
[1] -1.64485
```

R code ##### 3.4-3

```
qnorm(0.95, mean=0, sd=1)
[1] 1.64485
```

<34>

R code #### 3.5

```
qnorm(0.75, mean=2.11, sd=0.46)
[1] 2.420265
```

<35>

R code #### 3.6

```
yq <- ((1:n) - 0.5)/n
```

R code #### 3.7

```
y <- rnorm(100)
n <- length(y)
yq <- ((1:n) - 0.5)/n
zq <- qnorm(yq, mean=0, sd=1)
zq <- qnorm(yq, mean=0, sd=1)
plot(zq, sort(y), xlab="Standard Normal Quantile", ylab="Data")
abline(mean(y), sd(y))
```

第4章

<P57>

R code #### 4.1

```
TP.conc <- TP <- TP <- c(8.91, 4.76, 10.30, 2.32, 12.47, 4.49, 3.11, 9.61, 6.35,
5.84, 3.30, 12.38, 8.99, 7.79, 7.58, 6.70, 8.13, 5.47, 5.27, 3.52)
y <- log(TP.conc)
n <- length(y)
y.bar <- mean(y)
se <- sd(y)/sqrt(n)
int.50 <- y.bar + qt(c(0.25, 0.75), n-1)*se
int.95 <- y.bar + qt(c(.025, .975), n-1)*se
int.50
[1] 1.755323 1.900560
int.95
```

[1] 1.606902 2.048982

<P58>

R code ##### 4.2

```
n.sims <- 1000
n.size <- 30
inside <- 0
for (i in 1:n.sims){y <- rnorm(n.size, 2.05, 0.34);
se <- sd(y) / sqrt(n.size);int.95 <- mean(y) + qt(c(.025, .975),
n.size-1)*se;inside <- inside + sum(int.95[1]<2.05 & int.95[2]>2.05)}
inside / n.sims
```

95%信頼区間に真の平均値が入る確率

<P60>

R output

```
qnorm(0.75, mean=2.05, sd=0.34)
[1] 2.279
```

<P61>

R code ##### 4.3

```
n.sims <- 1000
n <- 30
y.bar <- mean(y)
se <- sd(y)
X <- rchisq (n.sims, df=n-1)
sigma.chi2 <- se * sqrt((n-1) / X)
sample.mean <- rnorm(n.sims, y.bar, sigma.chi2/sqrt(n))
q.75 <- qnorm(0.75, sample.mean, sigma.chi2)
hist(exp(q.75))
axis(1)
```

<P62>

R output

```
quantile(exp(q.75), prob=c(0.025, 0.975))
2.5%  97.5%
8.699 11.446
```

<P63>

R code #### 4.4

```
x <- c(94, 38, 23, 197, 99, 16, 141)
```

R code #### 4.5

```
boot.sample <- sample(x, size=length(x), replace=T)
```

R code #### 4.6

```
boot.mean <- mean(boot.sample)
```

R code #### 4.7

```
boot.mean <- numeric()
B <- 2000
for (i in 1:B){boot.sample <- sample(x, size=length(x), T);
  boot.mean[i] <- mean(boot.sample)}
```

R code #### (4.4+4.6+4.7+4.8) 4.8

```
x <- c(94, 38, 23, 197, 99, 16, 141)
boot.mean <-numeric(2000)
B <- 2000
for (i in 1:B){boot.sample <- sample (x, size=length(x), T);boot.mean[i]
<- mean (boot.sample);boot.se <- sd(boot.mean)}
boot.se
[1] 23.36
```

<P64>

R code and output#### 4.9: have to load "bootstrap"

```
require(bootstrap)
```

```
boot.mean <- bootstrap(x, 2000, mean)
sd(boot.mean$thetastar)
[1] 23.41
```

<P64>

R output

```
boot.median <- bootstrap(x, 2000, median)
sd(boot.median $ thetastar)
[1] 38.64895
```

<P65>

R code and output #### 4.10

```
boot.median <- bootstrap(x, 2000, median)
sd(boot.median $ thetastar)
hist(boot.median$thetastar)

[1] 38.65
mean(boot.median $ thetastar)
[1] 79.1105
CI <- mean(boot.median $ thetastar) + qt(c(0.025,0.975), 6)
CI
[1] 76.66359 81.55741
```

R code and output #### 4.11

```
CI.percent <- quantile(boot.median$thetastar,prob=c(0.025, 0.975))
CI.percent
2.5% 97.5%
23 141
```

<p66>

R code #### 4.12

```
TP.75Q <- quantile(y, prob=0.75)
```

R code and output #### 4.13

```
results <- bootstrap(y, 2000, quantile, prob=0.75)

## bootstrap-t CI
CI.t <- mean(results $ thetastar) + qt(c(0.025,0.975), 29)
CI.t

[1] 0.1997 4.2902

## percentile CI
CI.percent<- quantile(results$thetastar, prob=c(0.025, 0.975))
CI.percent

2.5% 97.5%
2.079 2.485

##BCa CI
Bca.results <- bcanon(y,2000,theta=quantile, prob=0.75, alpha=c(0.025,
0.975))
bca.results$confpoints

      alpha bca point
[1,] 0.025    2.079
[2,] 0.975    2.485
```

<P71>

R code and output #### 4.14

```
t.test(y, mu=log(10), alternative="greater")
```

<p72>

R output

```
      One Sample t-test
data: y
```

```
t = -4.0802, df = 29, p-value = 0.9998
alternative hypothesis: true mean is greater than 2.3026
95 percent confidence interval:
1.9417 Inf
sample estimates:
mean of x
2.0478
```

<P73>

R code #### 4.15

```
t.test(x=x, y=y, alternative="greater", var.equal=T)
```

R output

```
Two Sample t-test
data: x and y
t = 5.4022, df = 49, p-value = 9.61e-07
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval: 0.58144 Inf
sample estimates: mean of x mean of y 2.8909 2.0478
```

<P74>

R code #### 4.16

```
t.test(x=x, y=y, alternative="greater")
```

R output

```
Welch Two Sample t-test
data: x and y
t = 4.7943, df = 25.816, p-value = 2.941e-05
alternative hypothesis: true difference in means is
greater than 0
95 percent confidence interval:
0.54307 Inf
sample estimates:
```



```
mean of x mean of y
2.8909      2.0478
```

<P75>

R code #### 4.17

```
t.test(x, y, var.equal=T)
```

R output

Two Sample t-test

data: x and y

```
t = 5.4022, df = 49, p-value = 1.922e-06
```

```
alternative hypothesis: true difference in means is
not equal to 0
```

```
95 percent confidence interval:
```

```
0.52947 1.15672
```

```
sample estimates:
```

```
mean of x mean of y
```

```
2.8909 2.0478
```

<P76>

R code #### 4.18

```
t.test(y, mu=log(10))
```

R output

One Sample t-test

data: y

```
t = -4.0802, df = 29, p-value = 0.0003217
```

```
alternative hypothesis: true mean is not equal
to 2.3026
```

```
95 percent confidence interval:
```

```
1.9201 2.1755
```

```
sample estimates:
```

```
mean of x
```

```
2.0478
```

<P80>

R code #### 4.19

```
x <- c(17.0, 4.0, 7.0, 11.0, 21.5, 4.0, 24.0)
```

R code and output #### 4.20

```
rank (x)
[1] 5.0 1.5 3.0 4.0 6.0 1.5 7.0
```

R code and output #### 4.21

```
rank (x, ties.method = "min")
[1] 5 1 3 4 6 1 7
```

<P81>

R code #### 4.22

```
require(exactRankTests)
wilcox.exact(y, mu=log(10))
```

R output

```
Exact Wilcoxon signed rank test
data: y
V = 49, p-value = 0.0003513
alternative hypothesis: true mu is not equal to 2.3026
```

<p82>

R code #### 4.23

```
wilcox.test(y, mu=log(10))
```

R output

```
Wilcoxon signed rank test with continuity correction

data: y
```

V = 49, p-value = 0.0007723

alternative hypothesis: true location is not equal to 2.3026

<p83>

R code #### 4.24

```
require(exactRankTests)
wilcox.exact(Ozone ~ Month, data = airquality,
             subset = Month==5|Month==8)
```

<P83>

R output

```
Exact Wilcoxon rank sum test
data: Ozone by Month
W = 127.5, p-value = 6.109e-05
alternative hypothesis: true mu is not equal to 0
```

R code #### 4.25

```
wilcox.test(Ozone ~ Month, data = airquality,
            subset = Month==5|Month==8)
```

R output

```
Wilcoxon rank sum test with continuity correction

data: Ozone by Month
W = 127.5, p-value = 0.0001208
alternative hypothesis: true location shift is
not equal to 0
```

<P84>

R code #### 4.26 正規分布 (rnorm) 用

```
hyposim <- function(n.sims, rdistF, theta, mean, sd){
```

```

reject.t1<-0;reject.t2<-0;reject.w1<-0;reject.w2<-0
for (i in 1:n.sims){
  u <- rdistF(20, mean, sd)
  y<-u
  for (j in 2:20)
    y[j] <- u[j] - theta*u[j-1]
  samp1 <- data.frame(x=y, g=sample(1:2, 20, TRUE))
  ### randomized sample
  samp2 <- data.frame(x=y, g=rep(c(1,2), each=10))
  ### correlated sample
  reject.t1 <- reject.t1 +
    (t.test(x~g, data=samp1, var.equal=T)$p.value<0.05)
  reject.t2 <- reject.t2 +
    (t.test(x~g, data=samp2, var.equal=T)$p.value<0.05)
  reject.w1 <- reject.w1 +
    (wilcox.exact(x~g, data=samp1)$p.value<0.05)
  reject.w2 <- reject.w2 +
    (wilcox.exact(x~g, data=samp2)$p.value<0.05)
}
return(rbind(c(reject.t2, reject.t1),
              c(reject.w2, reject.w1))/n.sims)
}

```

R code #### 4.26 ポアソン分布用

```

hypo.sim <- function(n.sims, rdistF, theta, lambda){
  reject.t1<-0;reject.t2<-0;reject.w1<-0;reject.w2<-0
  for (i in 1:n.sims){
    u <- rdistF(20, lambda)
    y<-u
    for (j in 2:20)
      y[j] <- u[j] - theta*u[j-1]
    samp1 <- data.frame(x=y, g=sample(1:2, 20, TRUE))
    ### randomized sample
    samp2 <- data.frame(x=y, g=rep(c(1,2), each=10))
    ### correlated sample
    reject.t1 <- reject.t1 +

```

```

      (t.test(x~g, data=samp1, var.equal=T)$p.value<0.05)
reject.t2 <- reject.t2 +
      (t.test(x~g, data=samp2, var.equal=T)$p.value<0.05)
reject.w1 <- reject.w1 +
      (wilcox.exact(x~g, data=samp1)$p.value<0.05)
reject.w2 <- reject.w2 +
      (wilcox.exact(x~g, data=samp2)$p.value<0.05)
}
return(rbind(c(reject.t2,reject.t1),
             c(reject.w2,reject.w1))/n.sims)
}

```

<P85>

R output

```

hypo.sim(n.sims=1000,rdistF=rnorm,theta=-0.4,mean=2,sd=4)
  ## u from N(2,4)
  [,1] [,2]
[1,] 0.12 0.049
[2,] 0.10 0.049

```

```

hypo.sim(n.sims=1000,rdistF=rpois,theta=-0.4,lambda=3)
  ## u from Poisson(3)
  [,1] [,2]
[1,] 0.11 0.046
[2,] 0.11 0.053

```

```

hypo.sim(n.sims=1000,rdistF=runif,theta=-0.4,max=3,min=-3)
  ## u from uniform(-3,3)
  [,1] [,2]
[1,] 0.13 0.059
[2,] 0.11 0.051

```

R output

```
hypo.sim(n.sims=1000,rdistF=rnorm,theta=0.4,mean=2,sd=4)
```

```
      [,1] [,2]  
[1,] 0.003 0.055  
[2,] 0.002 0.047
```

```
hypo.sim(n.sims=1000,rdistF=rpois,theta=0.4,lambda=3)
```

```
      [,1] [,2]  
[1,] 0.003 0.060  
[2,] 0.004 0.061
```

```
hypo.sim(n.sims=1000,rdistF=runif,theta=0.4,max=3,min=-3)
```

```
      [,1] [,2]  
[1,] 0.004 0.064  
[2,] 0.004 0.062
```

<P86>

R output

```
hypo.sim(n.sims=1000,rdistF=rnorm,theta=0.,mean=2,sd=4)
```

```
      [,1] [,2]  
[1,] 0.051 0.050  
[2,] 0.041 0.044
```

```
hypo.sim(n.sims=1000,rdistF=rpois,theta=0.,lambda=3)
```

```
      [,1] [,2]  
[1,] 0.043 0.058  
[2,] 0.038 0.054
```

```
hypo.sim(n.sims=1000,rdistF=runif,theta=0.,max=3,min=-3)
```

```
      [,1] [,2]  
[1,] 0.066 0.053  
[2,] 0.055 0.047
```

<P89>

R code #### 4.27

```
power.t.test(n = 12, sd = 0.5, sig.level = 0.05,delta=0.15, type =  
"one.sample",alternative = "one.sided")
```

R output

One-sample t test power calculation

```
n = 12  
delta = 0.15  
sd = 0.5  
sig.level = 0.05  
power = 0.25  
alternative = one.sided
```

R code #### 4.28

```
power.t.test(sd = 0.5, sig.level = 0.05,power=0.85, delta=0.15,  
type="one.sample", alternative = "one.sided")
```

R output

One-sample t test power calculation

```
n = 81  
delta = 0.15  
sd = 0.5  
sig.level = 0.05  
power = 0.85  
alternative = one.sided
```

<P92>

R code #### 4.29

```
power.t.test(n = 10, sd = 0.93, sig.level = 0.05,power = 1-0.05, type =  
"one.sample", alternative = "one.sided")
```

R output

```
One-sample t test power calculation
n = 10
delta = 1.1
sd = 0.93
sig.level = 0.05
power = 0.95
alternative = one.sided
```

R code #### 4.30

```
power.t.test(n = 10, sd = 2.9, sig.level = 0.05,
             power = 1-0.05, type = "one.sample",
             alternative = "one.sided")
```

R output

```
One-sample t test power calculation

n = 10
delta = 3.3
sd = 2.9
sig.level = 0.05
power = 0.95
alternative = one.sided
```

R code #### 4.31

```
power.t.test(n = 10, sd = 0.93, sig.level = 0.05,
             delta = 1.5, type = "one.sample",
             alternative = "one.sided")
```

<P93>

R output

```
One-sample t test power calculation
```



```
n = 10
delta = 1.5
sd = 0.93
sig.level = 0.05
power = 0.99878
alternative = one.sided
```

R code #### 4.32

```
power.t.test(n = 10, sd = 2.9, sig.level = 0.05,
             delta = 2, type = "one.sample",
             alternative = "one.sided")
```

R output

One-sample t test power calculation

```
n = 10
delta = 1.5
sd = 2.9
sig.level = 0.05
power = 0.44707
alternative = one.sided
```

- ・ データの入力 : WCA2TP.txt

Everglades reference sites Data

- ・ "WCA2TP.csv"または"WCA2TP.txt"をメモ帳によりクリップにコピーする。 R Commander のデータインポートからデータを取り込む。
- ・ R システム内にデータを取り込めたら、以下の R コードをコピーして、 R Commander のスクリプトウィンドウにペーストし実行する。

```
wca2tp <- read.table("clipboard", header=TRUE, sep=",", na.strings="NA",
                   dec=".", strip.white=TRUE)
wca2tp$RESULT <- 1000*wca2tp$RESULT
```

```
wca2tp$TP <- wca2tp$RESULT
TP.reference <- wca2tp[wca2tp$Type=="R",]
TP.impact <- wca2tp[wca2tp$Type!="R",]
TP.reference$SITE <- as.vector( TP.reference$SITE)
TP.reference$Month <- ordered(months(as.Date(TP.reference$Date, "%m/%d/%y"), T),
  levels=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"))
```

<P97>

R code #### 4.33

```
Everg.aov <- aov(log(TP) ~ factor(Year), data=TP.reference)
summary (Everg.aov)
```

R output

```
              Df Sum Sq Mean Sq F value Pr(>F)
factor(Year)   5     6.5   1.3 6.7 5.1e-06
Residuals    430    83.9   0.2
```

R code #### 4.34

```
library(lattice, pos=4)
qqmath(~resid(Everg.aov),
  panel = function(x,...) {
    panel.grid()
    panel.qqmath(x,...)
    panel.qqmathline(x,...)
  },
  ylab="Residuals",
  xlab="Unit Normal Quantile")
```

<P98>

R code #### 4.35

```
xyplot(sqrt(abs(resid(Everg.aov)))~fitted(Everg.aov),
  panel=function(x,y,...) {
    panel.grid()
```

```

    panel.xyplot(x, y, ...)
    panel.loess(x, y, span=1, col="grey", ...)
  }, ylab="Sqrt. Abs. Residuals", xlab="Fitted")

```

<P99>

R code #### 4.36

```

xyplot(resid(Everg.aov)~fitted(Everg.aov),
       panel=function(x,y,...){
         panel.grid()
         panel.xyplot(x, y, ...)
         panel.abline(0, 0)
       }, ylab="Residuals", xlab="Fitted")

```

<P100>

R code #### 4.37

```

Everg.aov.lm <- lm(log(TP) ~ factor(Year), data=TP.reference)
summary.aov (Everg.aov.lm)

```

R output

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
factor(Year)	5	6.5	1.3	6.7	5.1e-06
Residuals	430	83.9	0.2		

<P101>

R code #### 4.38

```

summary (Everg.aov.lm)

```

R output

```

Call:
lm(formula = log(TP) ~ factor(Year), data = TP.reference)
Residuals:

```

```

Min          1Q   Median     3Q      Max
-0.8062 -0.2715 -0.0892  0.1822  2.2036

```

Coefficients:

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept)      2.1204     0.0631   33.59  <2e-16
factor(Year)1995 -0.2394     0.0774   -3.09   0.0021
factor(Year)1996  0.0288     0.0800    0.36   0.7187
factor(Year)1997  0.0814     0.0839    0.97   0.3325
factor(Year)1998  0.0581     0.0779    0.75   0.4560
factor(Year)1999  0.0721     0.0884    0.82   0.4150

```

<P103>

R code #### 4.39

```

anova.p <- t.p <- numeric()
for (i in 1:1000){data.sim <- data.frame(y=rnorm(120), g=rep(1:6,
each=20));sample.mean <- tapply(data.sim$y, data.sim$g, mean);data.sim$g <-
ordered(data.sim$g,levels=names(sort(sample.mean)));data.sim$g <-
as.numeric(data.sim$g);anova.p[i] <-
summary(aov(y~factor(g),data=data.sim))[[1]][1,5] < 0.05;t.p[i] <-
t.test(y~g, data=data.sim, subset=g==1|g==6)$p.value < 0.05}
print(c(mean(anova.p), mean(t.p)))

```

R output

```
[1] 0.047 0.346
```

R code #### 4.40

```

anova.p <- t.p <- numeric()
for (i in 1:1000){data.sim <- data.frame(y=rnorm(120), g=rep(1:6,
each=20));sample.mean <- tapply(data.sim$y, data.sim$g, mean);data.sim$g <-
ordered(data.sim$g,levels=names(sort(sample.mean)));data.sim$g <-
as.numeric(data.sim$g);anova.p[i] <-
summary(aov(y~factor(g),data=data.sim))[[1]][1,5] < 0.05;t.p[i] <-
t.test(y~g, data=data.sim, subset=g==1|g==6)$p.value < 0.0033}

```

```
print(c(mean(anova.p), mean(t.p)))
```

<P108>

R code and output #### 4.41

```
t.test(x=c(rep(1, 16), rep(0, 10)),
       y=c(rep(1, 19), rep(0, 56)) )
```

R output

```
Welch Two Sample t-test
data: c(rep(1, 16), rep(0, 10)) and c(rep(1, 19), rep(0, 56))
t = 3.3, df = 39, p-value = 0.00205
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
0.14 0.58
sample estimates:
mean of x mean of y
0.62          0.25
```

<P109>

R code and output #### 4.42

```
binom.test(x=10, n=16, p=0.5)
```

R output

```
Exact binomial test
data: 10 and 16
number of successes = 10, number of trials = 16,
p-value = 0.4545
alternative hypothesis: true probability of success is not equal
to 0.5
95 percent confidence interval:
0.35435 0.84802
sample estimates:
```

```
probability of success
      0.625
```

R code and output #### 4.43

```
binom.test(x=19, n=75, p=0.5)
```

R output

```
Exact binomial test
data: 19 and 75
number of successes = 19, number of trials = 75,
      p-value = 2.243e-05
alternative hypothesis: true probability of success is not
      equal to 0.5
95 percent confidence interval:
0.15993 0.36701
sample estimates:
probability of success
      0.25333
```

<P111>

R code and output #### 4.44

```
prop.test(x=c(16, 19), n=c(26, 75))
```

R output

```
2-sample test for equality of proportions with
      continuity correction
data: c(16, 19) out of c(26, 75)
X-squared = 9.6343, df = 1, p-value = 0.001910
alternative hypothesis: two.sided
95 percent confidence interval:
0.12483 0.59927
sample estimates:
prop 1  prop 2
```

0.61538 0.25333

<P114>

R code and output #### 4.45

```
qbinom(1-0.05, size=12, prob=0.1)
[1] 3
```

<P114>

R code and output #### 4.46

```
qbinom(1-0.05, size=5:20, prob=0.1) + 1
[1] 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5
```

<P115>

R code and output #### 4.47

```
1-pbinom(4-1, size=10, prob=0.25)
[1] 0.22412
```

R code #### 4.48

```
sample.size <- 10:40
reject <- qbinom(1-0.05, size=sample.size, prob=0.1) + 1
decision.table <- data.frame(n=10:40,
reject=reject,power=1-pbinom(reject-1,size=sample.size, prob=0.25))
plot(power~n, data=decision.table, type="l", xlab="Sample Size",
ylab="Power")
```

・ データ入力 : CompleteSpongeData.txt

ANOVA and multiple comparisons DATA###

```
# Ellison et al 1996
mangrove.sponge <- read.table(paste(dataDIR, "CompleteSpongeData.txt", sep="/"),
                              header=T)
```

<P118>

R code #### 4.49

```
mangrove.lm <- lm(RootGrowthRate ~ Treatment,
                  data=mangrove.sponge)
summary.aov(mangrove.lm)
```

R output

```
          Df Sum Sq Mean Sq F value Pr(>F)

Treatment 3      4.40  1.47  6.87 0.00041 ***
Residuals 68     14.51  0.21

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

R code #### 4.50

```
mangrove.aov <- aov(RootGrowthRate ~ Treatment,
                    data=mangrove.sponge)
mangrove.HSD <- TukeyHSD(mangrove.aov)
mangrove.HSD
```

R output

```
Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = RootGrowthRate ~ Treatment,
          data = mangrove.sponge)

$Treatment
              diff          lwr          upr          padj
Foam-Control    0.35436 -0.025798  0.73451  0.07650
Haliclona-Control 0.49109  0.094128  0.88806  0.00927
Tedania-Control  0.67643  0.256617  1.09624  0.00039
```



```

Haliclona-Foam      0.13674 -0.264644 0.53812 0.80630
Tedania-Foam       0.32207 -0.101917 0.74606 0.19790
Tedania-Haliclona  0.18534 -0.253787 0.62446 0.68369

```

<P119>

R code #### 4.51?

```

library(multcomp)
q2<-glht(mangrove.aov, linfct=mcp(Treatment="Tukey"))
summary(q2)
plot(q2)

```

<P120>

R code #### 4.52?

```

contr <- rbind("F - C" = c(-1, 1, 0, 0),
              "H - C" = c(-1, 0, 1, 0),
              "T - C" = c(-1, 0, 0, 1),
              "S - F" = c(0, -1, 1/2, 1/2),
              "S - C" = c(-1, 0, 1/2, 1/2))
q3 <- glht(mangrove.aov, linfct = mcp(Treatment = contr))

```

R output

```

summary(q3, test=adjusted(type=c("none")))
      Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: User-defined Contrasts
Fit: aov(formula = RootGrowthRate ~ Treatment,
          data = mangrove.sponge)

```

Linear Hypotheses:

	Estimate	Std. Error	t value	p value
F- C==0	0.354	0.144	2.45	0.0167
H- C==0	0.491	0.151	3.26	0.0018
T- C==0	0.676	0.159	4.24	6.8e-05
S- F==0	0.229	0.133	1.73	0.0885

```
S- C==0      0.584      0.131      4.46      3.1e-05
```

(Adjusted p values reported -- none method)

<P121>

R code #### 4.53-?

```
contr2 <- rbind("F - C" = c(-1, 1/3, 1/3, 1/3))
q4 <- glht(mangrove.aov, linfct = mcp(Treatment = contr2))
summary(q4, p.adjust.methods="none")
```

R output

Linear Hypotheses:

	Estimate	Std. Error	t value	p value
F - C == 0	0.507	0.120	4.22	7.4e-05

(Adjusted p values reported)

第5章

以下のデータ入力は第4章で行っているものと同じである。

Everglades reference sites Data ##### データの入力

- ・ "WCA2TP.csv"または"WCA2TP.txt"をメモ帳によりクリップにコピーする。R Commander のデータインポートからデータを取り込む。
- ・ R システム内にデータを取り込めたら、以下の R コードをコピーして、R Commander のスクリプトウィンドウにペーストし実行する。

```
wca2tp$RESULT <- 1000*wca2tp$RESULT
wca2tp$TP <- wca2tp$RESULT
TP.reference <- wca2tp[wca2tp$Type=="R",]
TP.impactted <- wca2tp[wca2tp$Type!="R",]
TP.reference$SITE <- as.vector( TP.reference$SITE)
```

```
TP.reference$Month <- ordered(months(as.Date(TP.reference$Date, "%m/%d/%y"), T),
  levels=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"))
```

<P126>

R code #### 5.1

```
subI <- (TP.impactd$SITE=="E4"|TP.impactd$SITE=="F4") &
  TP.impactd$Year==1994
subR <- TP.reference$Year==1994 & TP.reference$SITE!="E5" &
  TP.reference$SITE!="F5"

x<-log(TP.impactd$RESULT[subI])
y<-log(TP.reference$RESULT[subR])
t.test(x, y, alternative="greater", var.equal=T)
```

R output

```
Two Sample t-test

data: x and y
t = 5.4022, df = 49, p-value = 1.922e-06
alternative hypothesis: true difference in means is not
  equal to 0
95 percent confidence interval:
0.52947 1.15672
sample estimates:
mean of x mean of y
2.8909 2.0478
```

<P127>

R code and output #### 5.2

```
two.sample <- data.frame(TP = c(TP.impactd$RESULT[subI],
  TP.reference$RESULT[subR]),
  x = c(rep("I", sum(subI)),
  rep("R", sum(subR)))
```

```
t.test(log(TP) ~ x, data=two.sample, var.equal=T)
```

<P128>

R code #### 5.3

```
two.sample <- data.frame(y = c(TP.impactted$RESULT[subI],
                              TP.reference$RESULT[subR]),
                        x = c(rep(1, sum(subI)),
                              rep(0, sum(subR))))

t2lm <- lm(log(y) ~ x, data=two.sample)
summary(t2lm)
```

R output

Call:

```
lm(formula = log(y) ~ x, data = two.sample)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.694	-0.331	-0.102	0.151	1.763

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.048	0.100	20.4	<2e-16
x	0.843	0.156	5.4	1.9e-06

Residual standard error: 0.549 on 49 degrees of freedom

Multiple R-Squared: 0.373, Adjusted R-squared: 0.36

F-statistic: 29.2 on 1 and 49 DF, p-value: 1.92e-06

R code

```
y ~ x1 + x2 + ... + xp
```

<P130>

R code #### 5.4

```
anova.data<-data.frame(y=TP.reference$TP,
                        x2=ifelse(TP.reference$Year==1995, 1, 0),
                        x3=ifelse(TP.reference$Year==1996, 1, 0),
                        x4=ifelse(TP.reference$Year==1997, 1, 0),
                        x5=ifelse(TP.reference$Year==1998, 1, 0),
                        x6=ifelse(TP.reference$Year==1999, 1, 0))
anova.lm <- lm(log(y) ~ x2+x3+x4+x5+x6, data=anova.data)
summary(anova.lm)
```

R output

Call:

```
lm(formula = log(y) ~ x2 + x3 + x4 + x5 + x6, data=anova.data)
```

Residuals:

```
Min          1Q    Median      3Q      Max
-0.8062 -0.2715 -0.0892  0.1822  2.2036
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.1204	0.0631	33.59	<2e-16
x2	-0.2394	0.0774	-3.09	0.0021
x3	0.0288	0.0800	0.36	0.7187
x4	0.0814	0.0839	0.97	0.3325
x5	0.0581	0.0779	0.75	0.4560
x6	0.0721	0.0884	0.82	0.4150

<P131>

R code #### 5.5

```
anova.lm <- lm(log(TP) ~ factor(Year), data=TP.reference)
summary(anova.lm)
```

R output

Call:

```
lm(formula = log(TP) ~ factor(Year), data = TP.reference)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.8062	-0.2715	-0.0892	0.1822	2.2036

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.1204	0.0631	33.59	<2e-16
factor(Year)1995	-0.2394	0.0774	-3.09	0.0021
factor(Year)1996	0.0288	0.0800	0.36	0.7187
factor(Year)1997	0.0814	0.0839	0.97	0.3325
factor(Year)1998	0.0581	0.0779	0.75	0.4560
factor(Year)1999	0.0721	0.0884	0.82	0.4150

R code #### 5.6

```
anova.lm <- lm(log(TP) ~ factor(Year)-1, data=TP.reference)  
summary(anova.lm)
```

R output

Call:

```
lm(formula = log(TP) ~ factor(Year) - 1, data = TP.reference)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.8062	-0.2715	-0.0892	0.1822	2.2036

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
factor(Year)1994	2.1204	0.0631	33.6	<2e-16
factor(Year)1995	1.8810	0.0449	41.9	<2e-16
factor(Year)1996	2.1492	0.0491	43.8	<2e-16

```

factor(Year) 1997    2.2018    0.0552    39.9    <2e-16
factor(Year) 1998    2.1785    0.0456    47.8    <2e-16
factor(Year) 1999    2.1925    0.0619    35.4    <2e-16

```

<P135>

- ・データの入力 : laketrout2.csv

```
##### Lake trout Data #####
```

```
laketrout <- read.csv(paste(dataDIR,"laketrout2.csv", sep="/"), header=T)
```

データの入力後，以下の R コードをコピーして，R Commander のスクリプトウインドウにペーストし実行する．これによりデータの切り分けを行う．

```

laketrout$size <- "small"
laketrout$size[laketrout$length>60/2.54] <- "large"
laketrout$large <- 0
laketrout$large[laketrout$length>60/2.54] <- 1

```

- ・ display 関数使用のためにパッケージ arm をインストールする必要がある．
 - ・ R Commander の “ツール” で，“パッケージのロード “を選択し，表示されているウインドウに示されているパッケージの中から，arm を選択し，OK ボタンをクリックする．

R code ##### 5.7

```

lake.lm1 <- lm(log(pcb) ~ year, data=laketrout)
display(lake.lm1, 3)
summary(lake.lm1)

```

R output

```

lm(formula = log(pcb) ~ year, data = laketrout)
(Intercept) 119.8467 10.9689
year          -0.0599 0.0055
---
n = 631, k = 2
residual sd = 0.8784, R-Squared = 0.16

```

<P137>

R code #### 5.8

```
lake.lm2 <- lm(log(pcb) ~ I(year-1974)+length, data=laketrou)
display(lake.lm2, 4)
summary(lake.lm2)
```

R output

```
lm(formula = log(pcb) ~ I(year - 1974)+length, data = laketrou)
      coef.est      coef.se
(Intercept)   -1.834      0.120
I(year - 1974) -0.086      0.004
length         0.060      0.002
---
n = 631, k = 3
residual sd = 0.555, R-Squared = 0.66
```

<P138>

R code #### 5.9

```
laketrou <- na.omit(laketrou)
laketrou$len.c <- laketrou$length - mean(laketrou$length)
lake.lm3 <- lm(log(pcb) ~ I(year-1974)+len.c, data=laketrou)
display(lake.lm3, 3)
summary(lake.lm3)
```

R output

```
lm(formula = log(pcb) ~ I(year - 1974)+len.c, data = laketrou)
      coef.est      coef.se
(Intercept)    1.899      0.047
I(year - 1974) -0.086      0.004
len.c          0.060      0.002
---
n = 631, k = 3
residual sd = 0.555, R-Squared = 0.66
```


<P139>

R code #### 5.10

```
lake.lm4 <- lm(log(pcb) ~ I(year-1974)*len.c, data=laketrou)
display(lake.lm4, 4)
summary(lake.lm4)
```

R output

```
lm(formula = log(pcb) ~ I(year - 1974)*len.c, data = laketrout)

              coef.est   coef.se
(Intercept)    1.8967    0.0465
I(year-1974)   -0.0873    0.0036
Len.c          0.0510    0.0038
len.c:I(year - 1974) 0.0008    0.0003
---
n = 631, k = 4
residual sd = 0.5520, R-Squared = 0.67
```

<P142>

R output

```
summary(lake.lm4)

Call: lm(formula = log(pcb) ~ I(year - 1974)*len.c ,
          data = laketrout)

Residuals:
    Min     1Q  Median     3Q     Max
-2.4796 -0.3411  0.0197  0.3387  1.9711

Coefficients:
              Estimate      Std. Error  t value  Pr(>|t|)
(Intercept)    1.890718    0.046465   40.69   <2e-16
I(year-1974)   -0.087393    0.003604  -24.25   <2e-16
Len.c          0.051037    0.003841   13.29   <2e-16
```

```
Len.c:I(year-1974) 0.000848 0.000329 2.58 0.010
```

```
Residual standard error: 0.55 on 627 degrees of freedom
```

```
(15 observations deleted due to missingness)
```

```
Multiple R-Squared: 0.668, Adjusted R-squared: 0.667
```

```
F-statistic: 421 on 3 and 627 DF, p-value: <2e-16
```

<P144>

R code ##### 5.11

```
log(pcb) ~ I(year-1974)
```

```
log(pcb) ~ I(year-1974)+len.c
```

R output

```
summary.aov(lake.lm1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
I(year - 1974)	1	91	91	118	<2e-16
Residuals	629	485	1		

15 observations deleted due to missingness

R output

```
summary.aov(lake.lm2)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
I(year - 1974)	1	91	91	295	<2e-16
len.c	1	292	292	950	<2e-16
Residuals	628	193	0.3		

15 observations deleted due to missingness

<P145>

R output

```
summary(lake.lm4)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.89072	0.04646	40.7	4.3e-178
I(year - 1974)	-0.08739	0.00360	-24.2	4.0e-92
len.c	0.05104	0.00384	13.3	1.1e-35
len.c:I(year - 1974)	0.00085	0.00033	2.6	1.0e-02

<P146>

R code #### 5.12

```
lake.lm5 <- lm(log(pcb) ~ I(year-1974)*len.c +  
               I(len.c^2), data=laketROUT)  
display(lake.lm5, 4)
```

<P147>

R output

```
lm(formula = log(pcb) ~ I(year - 1974) * len.c +  
    I(len.c^2), data = laketrout)
```

	coef.est	coef.se
(Intercept)	1.8133	0.0496
I(year - 1974)	-0.0863	0.0036
len.c	0.0590	0.0043
len.c I(len.c^2)	0.0005	0.0001
I(year - 1974):len.c	0.0004	0.0003

n = 631, k = 5
residual sd = 0.5452, R-Squared = 0.68

<P149>

R code ##### 5.13 : この R コードを実行しておかないと 5.14-16 の R コードを正常に
実行することはできない

```
laketrout$size<- "small"  
laketrout$size[laketrout$length>60/2.54] <- "large"
```

R code ##### 5.14

```
lake.lm6 <- lm(log(pcb) ~ I(year-1974)*factor(size) +  
               len.c * factor(size),  
               data=laketrout)  
  
display(lake.lm6, 4)  
summary(lake.lm6)
```

R output

```
lm(formula = log(pcb) ~ I(year - 1974) * factor(size) +  
len.c * factor(size), data = lagetrout)
```

	coef.est	coef.se
(Intercept)	1.7394	0.0667
I(year - 1974)	-0.0846	0.0044
factor(size)small	-0.0647	0.1197
len.c	0.0776	0.0044
I(year - 1974):factor(size)small	0.0001	0.0074
factor(size)small:len.c	-0.0345	0.0063

n = 631, k = 6
residual sd = 0.5426, R-Squared = 0.68

<P151>

R code ##### 5.15

```
lake.lm7 <- lm(log(pcb) ~ I(year-1974) +  
               len.c * factor(size), data=laketrout)
```

```
display(lake.lm7, 4)
```

R output

```
lm(formula = log(pcb) ~ I(year - 1974) +
    len.c * factor(size), data = laketrout)

              coef.est      coef.se
(Intercept)      1.7389      0.0588
I(year - 1974)  -0.0846      0.0035
len.c             0.0776      0.0044
factor(size)small -0.0631      0.0779
len.c:factor(size)small -0.0345      0.0062
---
n = 631, k = 5
residual sd = 0.5422, R-Squared = 0.68
```

R code #### 5.16

```
lake.lm8 <- lm(log(pcb) ~ I(year-1974) +
    len.c * factor(size)-1-len.c, data=laketrout)
display(lake.lm8, 4)
```

R output

```
lm(formula = log(pcb) ~ I(year - 1974) +
    len.c * factor(size) - 1 - len.c,
    data = laketrout)

              coef.est      coef.se
I(year - 1974)  -0.0846      0.0035
factor(size)large  1.7389      0.0588
factor(size)small  1.6758      0.0795
len.c:factor(size)large  0.0776      0.0044
len.c:factor(size)small  0.0431      0.0045
---
n = 631, k = 5
```

```
residual sd = 0.5422, R-Squared = 0.83
```

<P153>

・ データの入力 : summerAll.csv

```
#### The Finnish Lakes example ####
```

```
summer.All <- read.table(paste(dataDIR, "summerAll.csv", sep="/"), sep=";", header=T)
```

```
#names(summer.All)
```

```
# [1] "totp" "chla" "type" "lake" "year" "totn" "month" "depth" "surfa"
```

```
#[10] "color"
```

```
summer.All <- summer.All[log(summer.All$chla) > -20 ,]
```

```
#> names(summer.All)
```

```
# [1] "totp" "chla" "type" "lake" "year" "totn" "month" "depth" "surfa"
```

```
#[10] "color"
```

```
summer.All$y <- log(summer.All$chla)
```

```
summer.All$lxp <- scale(log(summer.All$totp), scale=F)
```

```
summer.All$lxn <- scale(log(summer.All$totn), scale=F)
```

```
summer.All$type.lake <- paste(summer.All$type, summer.All$lake)
```

```
summer.All$npr <- scale(log(summer.All$totn/summer.All$totp), scale=F)
```

```
#### collinearity example ####
```

```
lake1 <- summer.All[summer.All$lake==19600 & summer.All$type==2,]
```

```
lake1$lxn <- scale(log(lake1$totn), scale=F)
```

```
lake1$lxp <- scale(log(lake1$totp), scale=F)
```

```
lake1$npr <- scale(lake1$npr, scale=F)
```

```
lake2 <- summer.All[summer.All$lake==1070,]
```

```
lake2$lxn <- scale(log(lake2$totn), scale=F)
```

```
lake2$lxp <- scale(log(lake2$totp), scale=F)
```

```
lake2$npr <- scale(lake2$npr, scale=F)
```

```
lake2 $ npr <- log(lake2$totn/lake2$totp)
```

```
lake3 <- summer.All[summer.All$lake==14700,]
```

```
lake3$lxn <- scale(log(lake3$totn), scale=F)
```

```
lake3$lxp <- scale(log(lake3$totp), scale=F)
lake4 <- summer.All[summer.All$lake==21800,]
lake4$lxn <- scale(log(lake4$totn), scale=F)
lake4$lxp <- scale(log(lake4$totp), scale=F)
```

```
Finn.lm1 <- lm(y ~ lxp + lxn, data=lake1)
Finn.lm2 <- lm(y ~ lxp + lxn, data=lake2)
Finn.lm3 <- lm(y ~ lxp * lxn, data=lake1)
Finn.lm4 <- lm(y ~ lxp * lxn, data=lake2)
Finn.lm5 <- lm(y ~ lxp + lxn + npr, data=lake1)
##
Finn.lm5 <- lm(y ~ lxp * lxn, data=lake3)
```

R output

```
> display(Finn.lm2)
lm(formula = y ~ lxp + lxn, data = lake2)

              coef.est      coef.se
(Intercept)          1.43         0.02
lxp                   0.67         0.04
lxn                   0.55         0.12
---
n = 441, k = 3
residual sd = 0.47, R-Squared = 0.55
```

<P155>

R code ##### 5.17

```
given.tn <- co.intervals(lake1$lxn, number=4,
                        overlap=.1)
coplot(y ~ lxp | lxn, data = lake1,
       given.v=given.tn, rows=1,
       panel=panel.smooth)
```

```

given.tp <- co.intervals(lake1$lxp, number=4,
                        overlap=.1)
coplot(y ~ lxn | lxp, data = lake1,
       given.v=given.tn, rows=1,
       panel=panel.smooth)

```

<P157>

```
Finn.lm4 <- lm(y ~ lxp * lxn, data = lake2)
```

R output

```
> display(Finn.lm4)
```

```
lm(formula = y ~ lxp * lxn, data = lake2)
```

	coef.est	coef.se
(Intercept)	1.43	0.02
lxp	0.66	0.04
lxn	0.52	0.13
lxp:lxn	0.05	0.10

n = 441, k = 4

residual sd = 0.47, R-Squared = 0.55

<P159>

```
Finn.lm4 <- lm(y ~ lxp * lxn, data = lake2)
```

R output

```
lm(formula = y ~ lxp * lxn, data = lake3)
```

	coef.est	coef.se
(Intercept)	1.59	0.03
lxp	0.57	0.07
lxn	0.75	0.14
lxp:lxn	0.31	0.12

```
n = 236, k = 4
residual sd = 0.33, R-Squared = 0.74
```

R output

```
lm(formula = y ~ lxp * lxn, data = lake4)
```

	coef.est	coef.se
(Intercept)	2.84	0.05
lxp	0.35	0.18
lxn	0.73	0.29
lxp:lxn	-0.31	0.18

```
n = 105, k = 4
residual sd = 0.44, R-Squared = 0.31
```

<P163>

R code ####5.18

```
laketrout$len.c2 <- laketrout$length-60
```

<P165>

R code ####5.19

```
lake.lm0 <- lm(pcb ~ I(year-1974) + len.c, data=laketrout)
PCBboxcox <- boxcox(lake.lm0)
```

<P166>

R code ####5.20

```
PCBboxcox$x[PCBboxcox$y==max(PCBboxcox$y)]
[1] -0.18
```

<P168>

R code ####5.21

```
predict(lake.lm1, new=data.frame(year=2007), se.fit=T)
```

<P169>

R output

```
$fit
[1] -0.3792
$se.fit
[1] 0.124
$df
[1] 629
$residual.scale
[1] 0.8784
```

R output

```
predict(lake.lm1, new=data.frame(year=2007), se.fit=T,
        interval="prediction")$fit

      fit    lwr    upr
[1,] -0.3792 -2.121  1.363
```

<P170>

R code ####5.22

```
attach(mangrove.sponge)
mangrove.sponge$Control <- as.numeric(Treatment=="Control")
mangrove.sponge$Foam <- as.numeric(Treatment=="Foam")
mangrove.sponge$PurpleS <- as.numeric(Treatment=="Haliclona")
mangrove.sponge$RedS <- as.numeric(Treatment=="Tedania")
detach()
```

R code ####5.23

```
mangrove.lmDM <- lm(RootGrowthRate ~ Foam+PurpleS+RedS,  
                    data=mangrove.sponge)
```

R output

```
display(mangrove.lmDM, 4)
```

```
lm(formula = RootGrowthRate ~ Foam + PurpleS + RedS,  
    data = mangrove.sponge)
```

	coef.est	coef.se
(Intercept)	0.2371	0.1008
Foam	0.3544	0.1443
PurpleS	0.4911	0.1507
RedS	0.6764	0.1594

n = 72, k = 4

residual sd = 0.4620, R-Squared = 0.23

<P171>

R output

```
display(mangrove.lm, 4)
```

```
lm(formula = RootGrowthRate ~ Treatment, data = mangrove.sponge)
```

	coef.est	coef.se
(Intercept)	0.2371	0.1008
TreatmentFoam	0.2544	0.1443
TreatmentHaliclona	0.4911	0.1507
TreatmentTetania	0.6764	0.1594

n = 72, k = 4

residual sd = 0.4620, R-Squared = 0.23

R code ####5.24

```
attach(mangrove.sponge)
mangrove.sponge$bbs <- as.numeric(Location=="bbs")
mangrove.sponge$etb <- as.numeric(Location=="etb")
mangrove.sponge$lcn <- as.numeric(Location=="lcn")
mangrove.sponge$lcs <- as.numeric(Location=="lcs")
detach()
mangrove.lmDM2 <- lm(RootGrowthRate ~ Foam+PurpleS+RedS +
                    etb+lcn+lcs ,
                    data=mangrove.sponge)
```

<P172>

R output

```
display(mangrove.lmDM2, 4)

lm(formula = RootGrowthRate ~ Foam + PurpleS + RedS +
    etb + lcn + lcs, data = mangrove.sponge)

```

	coef.est	coef.se
(Intercept)	0.1959	0.1378
Foam	0.3508	0.1436
PurpleS	0.4793	0.1503
RedS	0.5968	0.1650
etb	0.2426	0.1507
lcn	-0.0289	0.1575
lcs	0.0116	0.1520

```
---
n = 72, k = 7
residual sd = 0.4592, R-Squared = 0.28
```

<P173>

R code ####5.25

```
mangrove.lm2 <- lm(RootGrowthRate ~ Treatment+Location,
  data=mangrove.sponge)
display(mangrove.lm2, 4)
```

R output

```
lm(formula = RootGrowthRate ~ Treatment + Location,
  data = mangrove.sponge)
```

	coef.est	coef.se
(Intercept)	0.1959	0.1378
TreatmentFoam	0.3508	0.1436
TreatmentHaliclona	0.4793	0.1503
TreatmentTedania	0.5968	0.1650
Locationetb	0.2426	0.1507
Locationlcn	-0.0289	0.1575
Locationlcs	0.0116	0.1520

n = 72, k = 7

residual sd = 0.4592, R-Squared = 0.28

<P174>

R output

```
summary.aov(mangrove.lm2)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	3	4.40	1.47	6.96	0.00039
Location	3	0.81	0.27	1.27	0.29066
Residuals	65	13.71	0.21		

R code ####5.26

```
mangrove.aov2 <- aov(RootGrowthRate ~ Treatment+Location,
  data=mangrove.sponge)
summary(mangrove.aov2)
```

R output

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	3	4.40	1.47	6.96	0.00039
Location	3	0.81	0.27	1.27	0.29066
Residuals	65	13.71	0.21		

R code and output #####5.27

```
model.tables(mangrove.aov2)
```

Tables of effects

Treatment

	Control	Foam	Haliclona	Tedania
	-0.3459	0.008444	0.1452	0.3305
rep	21.0000	20.000000	17.0000	14.0000

Location

	bbs	etb	lcn	lcs
	-0.06204	0.1688	-0.07918	-0.04233
rep	19.00000	19.0000	16.00000	18.00000

<P175>

R code #####5.28

```
lm(RootGrowthRate ~ Treatment*Location,  
    data=mangrove.sponge)
```

R code #####5.29

```
lm(RootGrowthRate ~ Treatment+Location+  
    Treatment:Location,
```

```
data=mangrove.sponge)
```

R output

```
mangrove.aov3 <-aov(RootGrowthRate ~ Treatment*Location,  
data=mangrove.sponge)  
model.tables(mangrove.aov3)
```

Tables of effects

Treatment

Control	Foam	Haliclona	Tedania
-0.3459	0.008444	0.1452	0.3305

Location

bbs	etb	lcn	lcs
-0.06204	0.1688	-0.07918	-0.04233

Treatment:Location

	Location			
Treatment	bbs	etb	lcn	lcs
Control	-0.074	0.189	-0.052	-0.012
Foam	0.037	-0.229	0.200	-0.045
Haliclona	0.129	-0.083	-0.196	0.118
Tedania	-0.115	0.070	0.096	-0.064

<P176>

R output

```
summary(mangrove.aov3)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	3	4.40	1.47	6.49	0.00076
Location	3	0.81	0.27	1.19	0.32260
Treatment:Location	9	1.04	0.12	0.51	0.85945
Residuals	56	12.66	0.23		

第6章

<P178>

R code ####6.1

```
nls.obj <- nls (formula, data, start, control, algorithm,  
              trace, subset, weights, na.action, model,  
              lower, upper, ...)
```

R code ####6.2-1

```
lake.nlm1 <- nls(pcb ~ pcb0*exp(-k*(year-1974)),  
               data=laketrou, start=list(pcb0=10, k=0.08))
```

R output

```
summary(lake.nlm1)
```

```
Formula: pcb ~ pcb0 * exp(-k * (year - 1974))
```

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
pcb0	11.76215	0.64432	18.3	<2e-16
k	0.11487	0.00885	13.0	<2e-16

```
Residual standard error: 5.14 on 629 degrees of freedom
```


Number of iterations to convergence: 6

<P182>

R code ####6.3-2

```
pcb.exp2 <- nls(log(pcb) ~ log(pcb0*exp(-k*(year-1974))+pcba),  
  data=laketrou, start=list(pcb0=10, k=0.08, pcba=1))
```

R output

```
summary(pcb.exp2)
```

Formula: $\log(\text{pcb}) \sim \log(\text{pcb0} * \exp(-k * (\text{year} - 1974)) + \text{pcba})$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
pcb0	6.2264	0.8386	7.42	3.6e-13
k	0.2479	0.0401	6.18	1.1e-09
pcba	1.6941	0.1369	12.38	< 2e-16

Residual standard error: 0.862 on 645 degrees of freedom

R code ####6.4-3

```
pcb.exp3 <- nls(log(pcb) ~ log(pcb01*exp(-k1*(year-1974))+  
  pcb02*exp(-k2*(year-1974))),  
  data=laketrou,  
  start=list(pcb01=10, pcb02=2, k1=0.24, k2=0.00002))
```

R output

```
summary(pcb.exp3)
```

Formula: $\log(\text{pcb}) \sim \log(\text{pcb01} * \exp(-k1 * (\text{year} - 1974)) + \text{pcb02} * \exp(-k2 * (\text{year} - 1974)))$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
pcb01	6.7750	0.8577	7.90	1.2e-14
pcb02	0.7339	0.7644	0.96	0.3374
k1	0.1741	0.0528	3.29	0.0010
k2	-0.0359	0.0434	-0.83	0.4086

Residual standard error: 0.862 on 644 degrees of freedom

<P183>

R code ####6.5-4

```
pcb.exp3 <- nls(log(pcb) ~ log(pcb01*exp(-k1*(year-1974))+
                             pcb02*exp(-k2*(year-1974))),
               data=laketrou, algorithm="port", lower=rep(0,4),
               start=list(pcb01=10, pcb02=2, k1=0.24, k2=0.00002))
```

R output

```
summary(pcb.exp3)
```

Formula: $\log(\text{pcb}) \sim \log(\text{pcb01} * \exp(-k1 * (\text{year} - 1974)) + \text{pcb02} * \exp(-k2 * (\text{year} - 1974)))$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
pcb01	6.2264	0.9869	6.31	5.2e-10
pcb02	1.6941	0.9338	1.81	0.0701
k1	0.2479	0.0956	2.59	0.0097
k2	0.0000	0.0251	0.00	1.0000

Residual standard error: 0.862 on 644 degrees of freedom

<P184>

R code and output ####6.6-5

```
mixedorder <- function(x, b0, k, theta){
  LP1 <- LP2 <- 0;
  if(theta==1){
    LP1 <- log(b0) - k*x
  } else {
    LP2 <- log(b0^(1-theta) - k*x*(1-theta))/(1-theta)
  };
  return( LP1 + LP2)
}

pcb.exp4 <- nls(log(pcb) ~
  mixedorder(x=year-1974, pcb0, k, phi),
  data=laketrou, start=list(pcb0=10, k=0.0024, phi=3.5))
```

R output

```
summary(pcb.exp4)

Formula: log(pcb) ~
  mixedorder(x = year - 1974, pcb0, k, phi)
Parameters:

```

	Estimate	Std. Error	t value	Pr(> t)
pcb0	10.66409	1.72227	6.19	1.1e-09
k	0.00642	0.00271	2.37	0.018
phi	3.28579	0.35091	9.36	< 2e-16

```
Residual standard error:0.861 on 645 degrees of freedom
```

<P186>

R code ####6.7-6

```
lake.lm7 <- lm(log(pcb) ~ I(year-1974) +
              len.c * factor(size), data=laketrou)
display(lake.lm7, 4)
```

R output

```
lm(formula = log(pcb) ~ I(year - 1974) +
    len.c * factor(size), data = laketrou)

              coef.est      coef.se
(Intercept)      1.7389      0.0588

I(year - 1974)    -0.0846      0.0035
len.c             0.0776      0.0044
factor(size)small -0.0631      0.0779
len.c:factor(size)small -0.0345      0.0062
---
n = 631, k = 5
residual sd = 0.5422, R-Squared = 0.68
```

<P188>

R code ####6.8-7

```
hockey <-
function(x, alpha1, beta1, beta2, brk, eps=diff(range(x))/100,
        delta=T) {
  ## alpha1 is the intercept of the left line segment
  ## beta1 is the slope of the left line segment
  ## beta2 is the slope of the right line segment
  ## brk is location of the break point
  ## 2*eps is the length of the connecting quadratic piece
  x <- x-brk
```

```

    if (delta) beta2 <- beta1+beta2
    x1 <- -eps
    x2 <- +eps
    b <- (x2*beta1-x1*beta2)/(x2-x1)
    cc <- (beta2-b)/(2*x2)
    a <- alpha1+beta1*x1-b*x1-cc*x1^2
    alpha2 <- - beta2*x2 +(a + b*x2 + cc*x2^2)
    lebrk <- (x <= -eps)
    gebrk <- (x >= eps)
    eqbrk <- (x > -eps & x < eps)
    result <- rep(0,length(x))
    result[lebrk] <- alpha1 + beta1*x[lebrk]
    result[eqbrk] <- a + b*x[eqbrk] + cc*x[eqbrk]^2
    result[gebrk] <- alpha2 + beta2*x[gebrk]
    result
}

```

```
log(PCB) ~ hockey(length, beta0, beta1, delta, theta)
```

R code ####6.10-8

```
lake.nlm1<-nls(log(pcb)~hockey(length, beta0, beta1, delta, phi),
  start=list(beta0=.6, beta1=0.07, delta=0.03, phi=60/2.54),
  data=laketrou, na.action=na.omit)
```

R output

```
summary(lake.nlm1)
```

```
Formula: log(pcb) ~ hockey(length, beta0, beta1, delta, phi)
```

```
Parameters:
```

	Estimate	Std. Error	t value	Pr(> t)
beta0	0.5506	0.1316	4.18	3.3e-05

beta1	0.0253	0.0062	4.08	5.2e-05
delta	0.0470	0.0086	5.47	6.4e-08
phi	59.9896	2.3241	25.81	< 2e-16

Residual standard error: 0.751 on 627 degrees of freedom

<P190>

R code #####6.11-9?

```
lake.sim1 <- sim.nls (lake.nlm1, 1000)
betas <- lake.sim1$beta
logPCB.mean <- betas[,1] +
  (betas[,2] + betas[,3]*(60>betas[,4]))*(60-betas[,4])
pred.PCB<-exp(rnorm(1000, logPCB.mean, lake.sim1$sigma))
hist(pred.PCB)
```

<P190>

R output

```
quantile(betas[,4], prob=c(0.025,0.975))
  2.5%   97.5%
55.349  64.699
```

R code #####6.12-10

```
lake.nlm2 <- nls(log(pcb) ~ beta1*(year-1974) +
  hockey(length, beta0, beta2, delta, phi),
  start=list(beta0=.6, beta1= - 0.08,
    beta2=0.07, delta=0.03, phi=60/2.45),
  data=laketrou, na.action=na.omit)
```

R output

```
summary(lake.nlm2)
```

```
Formula: log(pcb) ~ beta1 * (year - 1974) +  
        hockey(length, beta0, beta2, delta, phi)
```

```
Parameters:
```

	Estimate	Std. Error	t value	Pr(> t)
beta0	1.59857	0.15338	10.42	< 2e-16
beta1	-0.08459	0.00353	-23.98	< 2e-16
beta2	0.04309	0.00436	9.88	< 2e-16
delta	0.03457	0.00622	5.55	4.1e-08
phi	60.71681	2.26282	26.83	< 2e-16

```
Residual standard error: 0.542 on 626 degrees of freedom
```

<P193>

データ入力 : NAmlilacSTID2.txt

```
## US Lilac data
```

```
USLilac <- read.csv(paste(dataDIR, " NAmlilacSTID2.txt", sep="/"))
```

```
#> USLilac[1:10,]
```

```
# STID Year Ptype FirstLeaf FirstBloom
```

```
#1 20309 1957 2 999 67
```

```
#2 20309 1958 2 999 90
```

・以下の R コードもスクリプトウインドウで実行しておく。

```
USLilac$type <- "Syringa chinensis clone"
```

```
USLilac$type[USLilac$Ptype==2] <- "Syringa vulgaris"
```

```
USLilac$FirstLeaf[USLilac$FirstLeaf==999] <- NA
```

```
USLilac$FirstBloom[USLilac$FirstBloom==999] <- NA
```

R code #####6.13-11

```
temp <- USLilac[USLilac$STID==354147,]
```

```
lilacs.lm1 <- nls( FirstBloom ~
```

```
    hockey(Year, beta0, beta1, delta, phi),
```

```
    start=list(beta0=100, beta1=0,
```

```
              delta=-0.1, phi=1980),
```

```
    data=temp, na.action=na.omit)
```

<P194>

R output

```
summary(lilacs.lml)
```

Formula:

```
FirstBloom ~ hockey(Year, beta0, beta1, delta, phi)
```

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
beta0	117.920	2.878	40.97	<2e-16
beta1	0.344	0.320	1.08	0.291
delta	-1.655	0.6686	-2.41	0.023
phi	1975.185	3.482	567.31	<2e-16

<P199>

R code ####6.14-12

```
pcb.loess <- loess (log(pcb)~ length,  
  data=laketrou, degree=1, span=0.5)
```

<P204>

・パッケージのロード : `gam` を `R Commander` のツールよりロードする.

R code ####6.15-13

```
PCB.gam <- gam(log(pcb)~s(length)+s(year), data=laketrou)
```

<P206>

R code #####6.16-14

```
PCB.gam <- gam(log(pcb)~lo(length)+lo(year), data=laketROUT)
```

R code #####6.17-15

```
PCB.gam <- gam(log(pcb)~lo(length, span=0.75, degree=1)+lo(year),  
  data=laketROUT)
```

・データの入力：nadb.txt の入力

North American Database

```
nadb <- read.table("clipboard", header=TRUE, sep="¥t", na.strings="NA",  
  dec=".", strip.white=TRUE)
```

R code #####6.18-16

```
require(mgcv)  
nadbGam1 <- gam(log(TPOut) ~ s(log(PLI))+s(log(TPIn))+s(log(HLR)),  
  data=nadb)
```

<P207>

R code #####6.19-17

```
par(mfrow=c(1,3), mar=c(3,3,0.5,0.25),  
  mgp=c(1.5,0.5,0))  
plot(nadbGam1, select=1, se=T, rug=T, resid=T,  
  scale=0, pch=16, cex=0.25)  
plot(nadbGam1, select=2, se=T, rug=T, resid=T,  
  scale=0, pch=16, cex=0.25)  
plot(nadbGam1, select=3, se=T, rug=T, resid=T,  
  scale=0, pch=16, cex=0.25)
```

<P209>

R code ####6.20-18

```
nadbGam3<-gam(log(TPOut)~s(var1=log(TPIn), var2=log(HLR)),
  data=nadb)
```

R output

```
summary(nadbGam3)
```

Family: gaussian

Link function: identity

Formula:

```
logTPOut ~ s(var1 = logTPIn, var2 = logHLR)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.35403	0.00888	39.9	<2e-16

Approximate significance of smooth terms:

	edf	Est.rank	F p-value
s(logTPIn,logHLR)	25.5	29 38.3	<2e-16

R-sq.(adj) = 0.796 Deviance explained = 81.4%

GCV score = 0.02429 Scale est. = 0.021986 n = 279

<P210>

R code ####6.21-19

```
par(mar=c(3,3,1,1), mgp=c(1.5,0.5,0), mfrow=c(1,2),
  pty="s")
plot(nadbGam3, select=1, se=T, rug=T, resid=T, pch=1)
plot(nadbGam3, select=1, se=T, rug=T, resid=T, pers=T)
```

<P212>

R code ####6.22-20

```
nadbGam1.5 <- gam(log(TPOut) ~ s(log(PLI), fx=T, k=4)+
                  s(log(HLR), fx=T, k=4),
                  data=nadb)
```

<P216>

R code ####6.23-21?

```
require (survival)
temp <- substring(J417$Activity.Start, 1, 10)
J417$Date <-
  mdy.date(month=as.numeric(substring(temp, 6,7)),
           day=as.numeric(substring(temp, 9,10)),
           year=as.numeric(substring(temp,1,4)))
```

<P217>

R code ####6.24-22?

```
FecalColiform <- rep(NA, 12*(2007-1970))
k <- 0
for (i in 1971:2007){ ## year
  for (j in 1:12){ ## month
    k <- k+1
    temp <- date.mdy(J417$Date)$month==j &
            date.mdy(J417$Date)$year==i

    if (sum(temp)>0)
      FecalColiform[k] <-
        mean(J417.FecalColiform$value[temp], na.rm=T)
  }
}
```

```
FecalColiform.ts <- ts(FecalColiform, start=c(1971,1),
    end=c(2007,12), freq=12)
```

R output ####(省略)

```
1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996
1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
Jan NA NA NA 1265.0 1117.5 10.0 NA 650.0 80.0 390.0 20.0 0.0 0.0 30.0 90.0 0.0 NA NA
Feb    Mar NA NA NA NA NA NA
Apr    May    Jun 2200.0    NA    0.0 NA NA 800.0 NA NA NA 310.0 1130.0 1033.3 85.0 2500.0 120.0 70.0 90.0 230.0 40.0 40.0 50.0
10.0 70.0 1100.0 2366.7 196.7 2983.3 100.0 50.0 70.0 20.0 50.0 NA 60.0 20.0 100.0 50.0 40.0 210.0 40.0 190.0 250.0 50.0 500.0 30.0 20.0 30.0 0.0
NA NA NA NA NA NA NA    NA    NA NA NA NA NA NA NA NA NA NA NA NA NA NA    NA    NA 500.0 310.0 700.0 97.0 87.0 750.0
NA 127.7 18.0 690.0 81.0 40.0 100.0 70.0 86.0 170.0 80.0 6000.0 41.0 55.0 680.0 NA NA 93.0 32.0 36.0 82.0 970.0 66.0 200.0 54.0 NA 550.0 77.0
56.0 190.0 75.0 NA 1041.5
Jul    Aug NA 30.0 NA 253.3
7000.0 NA 562.5 8877.5 56.7 45.0 110.0 0.0 20.0 100.0 0.0 960.0 4093.3 55.0 20.0 10.0 0.0 2200.0 190.0 14000.0 70.0 60.0 160.0 70.0 60.0 690.0
NA NA NA NA NA NA NA    NA NA NA NA NA NA NA NA NA NA NA 110.0 180.0 271.5 130.0 180.0 45.0 54.0 73.0 20.0 50.0 2300.0 36.0 34.0
2000.0 130.0 1700.0 73.0 2000.0 260.0 6800.0 NA 180.0 1100.0 100.0 NA 57.0
Sep    Oct NA NA 390.0 NA 93.3 35.0 576.0 173.3 1930.0 375.0 1800.0 190.0 330.0 40.0 170.0 560.0 27050.0 280.0 50.0 160.0 40.0 10.0
150.0 60.0 5200.0 40.0 110.0 110.0 30.0 10.0 NA 190.0 NA NA NA NA NA    NA NA NA NA NA NA NA NA NA 120.0 2500.0 700.0 230.0 40.5 91.7
82.0 NA 67.0 91.0 NA 100.0 140.0 45.0 NA 85.5 NA 1425.5 160.0 74.0 190.0 150.0 300.0 97.0 NA 84.0 NA 120.0
Nov    Dec NA NA NA NA
216.0 4430 260.0 NA 30.0 50 660.0 100 14000.0 0 510.0 5600 2200.0 810 150.0 100 90.0 320 390.0 60 120.0 9700 130.0 10000 70.0 NA NA NA NA
NA NA NA NA NA NA NA NA NA NA NA NA NA NA 500.0    160 NA NA 54.5 18 27.0 27 230.0 62 80.0 240 90.0 64 230.0 55 56.0 320 110.0 170 83.0
130 NA 800 120.0 45 NA NA
NA NA NA NA NA
100.0 73.0 NA 340.0 33.0 1000.0 71.0 820.0 64.0 68.0 66.0 86.0 280.0
NA NA NA NA NA NA NA NA NA    NA
60.0 110.0 170.0 1400.0 52.5 22.5 82.0 14.0 75.0 36.0 45.0 NA 120.0 NA 30.0 13.0 230.0 NA 53.0 120.0 430.0 70.0 38.0 78.0 45.0 NA
276.7 3505.0 847.5 155.0 0.0 40.0 20.0 990.0 60.0 NA 240.0 240.0 490.0 780.0 50.0 80.0 250.0 70.0 170.0 510.0 30.0 40.0 150.0 50.0 20.0 50.0 NA
NA NA NA NA 710.0    NA
```

<P218>

R code ####6.25-23?

```
temp.2w <- medpolish(matrix(data.ts, ncol=12, byrow=T),
  eps=0.001, na.rm=T)
year.temp <- rep(seq(start(data.ts)[1], end(data.ts)[1]),
  each=12)
month.temp <- rep(1:12,
  length(seq(start(data.ts)[1], end(data.ts)[1])))
data.ts[is.na(data.ts)]<-temp.2w$overall +
  temp.2w$row[year.temp[is.na(data.ts)]-start(data.ts)[1]+1]+
  temp.2w$col[month.temp[is.na(data.ts)]]
```

Chapter 7

<P231>

・データの入力 : willamette.txt

```
Willamette.data <- read.table("clipboard", header=TRUE, sep="," ,
  na.strings="NA", dec=".", strip.white=TRUE)
```

R code ####7.1

```
library(rpart)
set.seed(12345)
diuron.rpart <- rpart(log(P49300) ~ NH4+NO2+TKN+N2.3+
  TOTP+SRP+BOD+ECOL+FECAL+Longitude+Latitude+Size+
  LU.Ag+LU.For+LU.Resid+LU.Other+NumCrops+Month,
  data=Willamette.data,
  control=rpart.control(minsplit=4, cp=0.001))
```

<P232>

R code ####7.2

```
plot(diuron.rpart, margin=0.1)
text(diuron.rpart, cex=0.5)
```

<P233>

R output

```
printcp(diuron.rpart)
```

Regression tree:

```
rpart(formula = log(P49300) ~ NH4 + NO2 + TKN + N2.3 + TOTP +
      SRP + BOD + ECOL + FECAL + Longitude + Latitude + Size +
      LU.Ag + LU.For + LU.Resid + LU.Other + NumCrops + Month,
      data = Willamette.data,
      control = rpart.control(minsplit = 4, cp = 0.005))
```

Variables actually used in tree construction:

```
[1] BOD FECAL Longitude LU.Ag LU.For LU.Resid Month
[8] N2.3 NH4   Size SRP   TKN   TOTP
```

Root node error: 451/94 = 4.8

n=94 (1 observation deleted due to missingness)

	CP	nsplit	rel error	xerror	xstd
1	0.38360	0	1.0000	1.030	0.1030
2	0.13035	1	0.6164	0.819	0.1130
3	0.09846	2	0.4861	0.664	0.0996
4	0.07110	3	0.3876	0.611	0.1023
5	0.04023	4	0.3165	0.547	0.0959
6	0.02545	5	0.2763	0.656	0.1314
7	0.02276	6	0.2508	0.720	0.1392
8	0.02013	7	0.2281	0.733	0.1389
9	0.01798	8	0.2079	0.701	0.1394
10	0.01653	10	0.1720	0.701	0.1394
11	0.01604	12	0.1389	0.687	0.1378
12	0.01049	13	0.1229	0.683	0.1353
13	0.00987	14	0.1124	0.767	0.1437
14	0.00935	15	0.1025	0.778	0.1456
15	0.00835	16	0.0932	0.793	0.1458
16	0.00818	17	0.0848	0.791	0.1459

```
17 0.00653      18  0.0766      0.790      0.1447
18 0.00561      19  0.0701      0.798      0.1453
19 0.00500      21  0.0589      0.798      0.1451
```

<P235>

R code ####7.3

```
plotcp(diuron.rpart)
```

<P236>

R code ####7.4

```
diuron.rpart.prune <- prune(diuron.rpart, cp=0.05)
```

R code ####7.5

```
nf <- layout(matrix(c(1,2), nrow=2, ncol=1), 1, c(2,1))
par(mar=c(0,4,1,2))
plot(diuron.rpart.prune, compress=F, branch=0.4, margin=0.1)
text(diuron.rpart.prune, pretty=T, cex=0.55, use.n=T)
title(main="log diuron Concentration")
par(mar=c(0.5,4,0.5,2))
boxplot(split(predict(diuron.rpart.prune)+resid(diuron.rpart.prune),
              round(predict(diuron.rpart.prune), digits=4)),
         ylab="Diuron Concentrations",
         xlab=" ", axes=F, ylim=log(c(0.01, 50)))
axis(2, at=log(c(0.01, 0.1, 1, 10, 50)),
     labels=c("0.01","0.1","1","10","50"), las=1)
box()
```

<P240>

R code ####7.6

```

Willamette.data$Diuron <- "Below MDL"
Willamette.data$Diuron[Willamette.data$P49300>=7.08]
  <- "High"
Willamette.data$Diuron[Willamette.data$P49300<7.08 &
  Willamette.data$P49300>=0.83] <- "Medium"
Willamette.data$Diuron[Willamette.data$P49300<0.83 &
  Willamette.data$P49300>0.02] <- "Low"
Willamette.data$Diuron <- ordered(Willamette.data$Diuron,
  levels=c("Below MDL", "Low", "Medium", "High"))
Willamette.data$Diuron[is.na(Willamette.data$P49300)] <- NA

```

<P241>

R code ####7.7

```

set.seed(12345)
diuron.rpart2 <- rpart(Diuron ~ NH4+NO2+TKN+N2.3+TOTP+
  SRP+BOD+ECOL+FECAL+Longitude+Latitude+Size+LU.Ag+
  LU.For+LU.Resid+LU.Other+NumCrops+Month,
  data=Willamette.data, method="class",
  parms=list(prior=rep(1/4, 4), split='information'),
  control=rpart.control(minsplit=4, cp=0.005))

```

<P242>

R code ####7.8

```

diuron.rpart2.prune <- prune(diuron.rpart2, cp=0.06)

```

<P244>

R code ####7.9

```

set.seed(123456)
diuron.rpart5 <- rpart(Diuron ~ NH4+NO2+TKN+N2.3+TOTP+SRP+BOD+ECOL+FECAL+
  Longitude+Latitude+Size+LU.Ag+LU.For+LU.Resid+LU.Other+NumCrops+Month,
  data=Willamette.data, method="class",
  control=rpart.control(minsplit=4, cp=0.000001))

```



```

plotcp(diuron.rpart5)
printcp(diuron.rpart5)
diuron.rpart5.prune <- prune(diuron.rpart5, cp=0.02)
plot(diuron.rpart5.prune, margin=0.1)
text(diuron.rpart5.prune, pretty=T, use.n=T)

# Default
plot(diuron.rpart5.prune,margin=0.1,
     main="a. the default plot")
text(diuron.rpart5.prune)

```

R code ####7.10

```

# Uniform with branching
plot(diuron.rpart5.prune,uniform=T,branch=0.25,
     margin=0.1, main="b. uniform with branching")
text(diuron.rpart5.prune,pretty=1,use.n=T)

```

<P246>

R code ####7.11

```

plot(diuron.rpart5.prune,uniform=T,branch=0,
     margin=0.1,main="c. fancy")
text(diuron.rpart5.prune,pretty=1,
     all=T,use.n=T,fancy=T)

```

<P247>

R code ####7.12

```

> summary(diuron.rpart2.prune)
Call:
rpart(formula = log(P49300) ~ NH4 + NO2 + TKN + N2.3 + TOTP+ SRP +
      BOD + ECOL + FECAL + Longitude + Latitude + Size + LU.Ag +

```

```

LU.For + LU.Resid + LU.Other + NumCrops + Month,
data = willamette.data, method = "class",
parms = list(prior = rep(1/4, 4), split = "information"),
control = rpart.control(minsplit = 4, cp = 0.005))
n=94 (1 observation deleted due to missingness)

```

	CP	nsplit	rel error	xerror	xstd
1	0.32051	0	1.00000	1.17483	0.096700
2	0.10606	1	0.67949	0.77564	0.073316
3	0.07085	2	0.57343	0.66719	0.074220
4	0.06000	3	0.50258	0.57487	0.075417

Node number 1: 94 observations, complexity param=0.32051

predicted class=Low expected loss=0.75

class counts: 26 38 22 8

probabilities: 0.250 0.250 0.250 0.250

left son=2 (63 obs) right son=3 (31 obs)

Primary splits:

LU.Ag < 74.5 to the left, improve=31.314, (0 missing)

NH4 < 0.0835 to the left, improve=31.256, (22 missing)

TOTP < 0.2015 to the left, improve=29.062, (3 missing)

N2.3 < 1.69 to the left, improve=24.230, (3 missing)

BOD < 2.55 to the left, improve=24.218, (30 missing)

.....

Chapter 8

<P259>

・ データ入力 : cryptoDATA.csv

Crypto data: data combined from Finch et al 1993 and Korich et al 2000

```

crypto.data <- read.table(paste(dataDIR, "cryptoDATA.csv", sep="/"), h
eader=T) ## but not comma separated

```

・以下も R Commander のスクリプトウインドウで実行しておく.

```
crypto.data$Y <- round(crypto.data$Y)
```

```
crypto.lm1 <- lm(I(logit(Y/N))~log10(Dose), data=crypto.data, subset=Y/N!=0 & Y/N!=1)
display(crypto.lm1)
```

```
crypto.glm1 <- glm(cbind(Y, N-Y)~log10(Dose), data=crypto.data,
family=binomial(link="logit"))
```

```
display(crypto.glm1, 3)
```

```
### overdispersion
```

```
z <- (crypto.data$Y-crypto.data$N*fitted(crypto.glm1))/sqrt(crypto.data$N*fitted(crypto.glm1)*(1-fitted(crypto.glm1)))
```

```
z.chisq <- sum(z^2)
```

```
orverD <- z.chisq/summary(crypto.glm1)$df[2]
```

```
p.value <- 1-pchisq(z.chisq, df=summary(crypto.glm1)$df[2])
```

R code #####8.1

```
crypto.glm1<-glm(cbind(Y,N-Y)~log10(Dose),data=crypto.data,
family=binomial(link="logit"))
```

R output

```
summary(crypto.glm1)
```

Call:

```
glm(formula = cbind(Y, N - Y) ~ log10(Dose),
     family = binomial(link = "logit"), data = crypto.data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-3.8111	-1.2590	-0.0883	1.7001	5.1206

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-4.865	0.329	-14.8	<2e-16
log10(Dose)	2.616	0.162	16.2	<2e-16

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 692.99 on 97 degrees of freedom

Residual deviance: 368.05 on 96 degrees of freedom

AIC: 588

<P260>

R output

```
display(crypto.glm1, 3)
```

```
glm(formula = cbind(Y, N - Y) ~ log10(Dose),  
     family = binomial(link = "logit"),  
     data = crypto.data)
```

	coef.est	coef.se
(Intercept)	-4.865	0.329
log10(Dose)	2.616	0.162

n = 98, k = 2
residual deviance = 368.1,
null deviance = 693.0 (difference = 324.9)

<P263>

R output

```
crypto.glm1<-glm(cbind(Y,N-Y)~log10(Dose)+factor(Source),data=crypto.data,  
family=binomial(link="logit"))  
display(crypto.glm1, 3)
```

```

glm(formula = cbind(Y, N - Y) ~ log10(Dose) + factor(Source),
     family = binomial(link = "logit"), data = crypto.data)

```

	coef.est	coef.se
(Intercept)	-5.01	0.35
log10(Dose)	2.63	0.16
factor(Source)SPDL-HE	0.05	0.18
factor(Source)SPDL-TH	0.32	0.18
factor(Source)UA	0.07	0.16

```

n = 98, k = 5
residual deviance = 363.8,
null deviance = 693.0 (difference = 329.1)

```

<P264>

R output

```

crypto.glm2 <- glm(cbind(Y, N-Y)~log10(Dose)+factor(Source)-1, data=crypto.data,
family=binomial(link="logit"))
display(crypto.glm2)

```

```

glm(formula = cbind(Y, N - Y) ~ log10(Dose) + factor(Source) - 1,
     family = binomial(link = "logit"), data = crypto.data)

```

	coef.est	coef.se
log10(Dose)	2.63	0.16
factor(Source)Finch	-5.01	0.35
factor(Source)SPDL-HE	-4.96	0.34
factor(Source)SPDL-TH	-4.69	0.34
factor(Source)UA	-4.94	0.34

```

n = 98, k = 5
residual deviance = 363.8
null deviance = 744.1 (difference = 380.3)

```

<P265>

R output

```
crypto.glm4 <- glm(cbind(Y, N-Y)~log10(Dose)*factor(Source), data=crypto.data,
family=binomial(link="logit"))
display(crypto.glm4)
```

```
glm(formula = cbind(Y, N - Y) ~ log10(Dose) * factor(Source),
     family = binomial(link = "logit"), data = crypto.data)

              coef.est      coef.se
(Intercept)      -6.53         0.98
log10(Dose)         3.39         0.49
factor(Source)SPDL-HE  3.35         1.13
factor(Source)SPDL-TH  2.37         1.15
factor(Source)UA         0.06         1.17
log10(Dose):factor(Source)SPDL-HE -1.66         0.56
log10(Dose):factor(Source)SPDL-TH -1.03         0.57

log10(Dose):factor(Source)UA      -0.01         0.57
n = 98, k = 8
residual deviance = 344.5,
null deviance = 693.0 (difference = 348.5)
```

<P266>

R output

```
crypto.glm3 <- glm(cbind(Y, N-Y)~log10(Dose)*factor(Source)-1*log10(Dose),
data=crypto.data, family=binomial(link="logit"))
display(crypto.glm3)
```

```
glm(formula = cbind(Y, N - Y) ~ log10(Dose) * factor(Source)
     - 1 - log10(Dose),
     family = binomial(link = "logit"), data = crypto.data)

              coef.est      coef.se
factor(Source)Finch      -6.53         0.98
factor(Source)SPDL-HE     -3.18         0.57
factor(Source)SPDL-TH     -4.16         0.60
```

```

factor(Source)UA                -6.47      0.63
log10(Dose):factor(Source)Finch  3.39      0.49
log10(Dose):factor(Source)SPDL-HE 1.73      0.28
log10(Dose):factor(Source)SPDL-TH 2.36      0.31
log10(Dose):factor(Source)UA     3.38      0.30

n = 98, k = 8
residual deviance = 344.5,
null deviance = 744.1 (difference = 399.6)

```

<P267>

R output

```

crypto.lm3<-lm(I(logit(Y/N))~log10(Dose)*factor(Source)-1*log10(Dose),
data=crypto.data, subset=Y/N!=0 & Y/N!=1)
display(crypto.lm3)

```

```

lm(formula = I(logit(Y/N)) ~ log10(Dose) * factor(Source)
- 1 - log10(Dose), data = crypto.data,
subset = Y/N != 0 & Y/N !=1)

```

	coef.est	coef.se
factor(Source)Finch	-2.64	1.40
factor(Source)SPDL-HE	-3.71	1.19
factor(Source)SPDL-TH	-3.97	1.21
factor(Source)UA	-6.05	1.20
log10(Dose):factor(Source)Finch	1.12	0.72
log10(Dose):factor(Source)SPDL-HE	2.00	0.59
log10(Dose):factor(Source)SPDL-TH	2.23	0.61
log10(Dose):factor(Source)UA	3.23	0.57

```

n = 76, k = 8
residual sd = 0.95, R-Squared = 0.54

```

<P269>

R code ####8.2

```
z <- (crypto.data$Y-crypto.data$N*fitted(crypto.glm1))/
      sqrt(crypto.data$N*fitted(crypto.glm1)*
           (1-fitted(crypto.glm1)))
z.chisq <- sum(z^2)
overD <- z.chisq/summary(crypto.glm1)$df[2]
overD
[1] 3.4784

p.value <- 1-pchisq(z.chisq, df=summary(crypto.glm1)$df[2])
p.value
[1] 0
```

<P270>

R output

```
display(crypto.glm1, 3)

glm(formula = cbind(Y, M - Y) ~ log10(Dose),
     family = quasibinomial(link = "logit"),
     data = crypto.data)
              coef.est   coef.se
(Intercept)   -4.865     0.613
log10(Dose)    2.616     0.302
n = 98, k= 2
residual deviance = 368.1,
null deviance = 693.0 (difference = 324.9)
overdispersion parameter = 3.5
```

<P271>

• データ入力 : seedbank.txt

```
seedbank <- read.table("clipboard", header=TRUE, sep=",", na.strings="NA",
                      dec=".", strip.white=TRUE)
```



```
Pred ~ factor(species)
```

```
Pred ~ log(seed.weight)
```

Output

```
seedbank.glm1 <- glm(Predation ~ log(seed.weight), data=seedbank,  
family=binomial(link="logit"))  
display(seedbank.glm1)
```

```
glm(formula = Predation ~ log(seed.weight),  
family = binomial(link = "logit"),  
data = seedbank)
```

	coef.est	coef.se
(Intercept)	-3.55	0.20
log(seed.weight)	0.42	0.04

n = 1142, k = 2
residual deviance = 785.2,
null deviance = 939.8 (difference = 154.6)

<P272>

R output

```
seedbank.glm2 <- glm(Predation ~ species-1, data=seedbank,  
family=binomial(link="logit"))  
display(seedbank.glm2)
```

```
glm(formula = Predation ~ species - 1,  
family = binomial(link = "logit"),  
data = seedbank)
```

	coef.est	coef.se
species8	-0.07	0.17
species7	-1.88	0.25
species6	-1.76	0.24

```

species5 -0.91  0.18
species4 -4.26  0.71
species3 -2.97  0.39
species2 -3.32  0.46
species1 -18.57 549.31
  n = 1142, k = 8
  residual deviance = 721.1,

```

R output

```

seedbank.glm4 <- glm(Predation ~ factor(time)+log(seed.weight), data=seedbank,
family=binomial(link="logit"))
display(seedbank.glm4)

```

```

glm(formula=Predation~factor(time)+log(seed.weight),
     family = binomial(link = "logit"), data = seedbank)

```

	coef.est	coef.se
(Intercept)	-6.33	0.64
factor(time)2	2.37	0.64
factor(time)3	2.70	0.64
factor(time)4	3.11	0.63
factor(time)5	2.98	0.63
factor(time)6	3.10	0.63
log(seed.weight)	0.46	0.04

```

  n = 1142, k = 7
  residual deviance = 726.3,
  null deviance = 939.8 (difference = 213.5)

```

<P273>

R output

```

seedbank.glm4.2 <- glm(Predation ~ factor(time)+log(seed.weight)-1,
                      data=seedbank, family=binomial(link="logit"))
display(seedbank.glm4.2)

```

```

glm(formula = Predation ~ factor(time) + log(seed.weight) - 1,
     family = binomial(link = "logit"), data = seedbank)

```

	coef.est	coef.se
factor(time)1	-6.33	0.64
factor(time)2	-3.95	0.32
factor(time)3	-3.62	0.29
factor(time)4	-3.21	0.27
factor(time)5	-3.34	0.27
factor(time)6	-3.23	0.27
log(seed.weight)	0.46	0.04

```

n = 1142, k = 7
residual deviance = 726.3,
null deviance = 1583.1 (difference = 856.9)

```

<P274>

R output

```

invlogit(coef(seedbank.glm4)[1:6] +
         coef(seedbank.glm4)[7] * log(28.5))

```

factor(time)1	factor(time)2	factor(time)3
0.0082	0.0812	0.1094
factor(time)4	factor(time)5	factor(time)6
0.1561	0.1399	0.1539

R code ####8.3

```

seedbank.glm4 <- glm(Predation~factor(time)+log(seed.weight),
                    data=seedbank, family=binomial(link="logit"))
betas <- coef(seedbank.glm4)
par(mgp=c(1.5,.5,0), mar=c(3,3,1,0.25))
plot(jitter(Predation) ~ log(seed.weight),

```

```

    type="n", data=seedbank,
    xlab="log seed weight",
    ylab="prob. of predation")
points(jitter(seedbank$Predation)~log(seedbank$seed.weight),
       col="gray")
curve(invlogit(betas[1]+betas[7]*x), add=T, col=gray(.1))
curve(invlogit(betas[1]+betas[2]+betas[7]*x), add=T,
       lty=2, col=gray(.2))
curve(invlogit(betas[1]+betas[3]+betas[7]*x), add=T,
       lty=3, col=gray(.3))
curve(invlogit(betas[1]+betas[4]+betas[7]*x), add=T,
       lty=4, col=gray(.4))
curve(invlogit(betas[1]+betas[5]+betas[7]*x), add=T,
       lty=5, col=gray(.5))

curve(invlogit(betas[1]+betas[6]+betas[7]*x), add=T,
       lty=6, col=gray(.6))
legend(x=0, y=0.9, legend=month.name[seq(1,11,2)],
       lty=1:6, col=gray((1:6)/10), cex=0.5, bty="n")

```

<P277>

R output

```

seedbank.glm5 <- glm(Predation ~ factor(time)+factor(topo)+log(seed.weight),
data=seedbank, family=binomial(link="logit"))
display(seedbank.glm5)

```

```

glm(formula=Predation~factor(time)+factor(topo)
+log(seed.weight),
family = binomial(link = "logit"),
data = seedbank)

```

	coef.est	coef.se
(Intercept)	-5.72	0.67
factor(time)2	2.68	0.67

```

factor(time)3          3.06      0.67
factor(time)4          3.53      0.66
factor(time)5          3.38      0.67
factor(time)6          3.59      0.67
factor(topo)2          -2.21     0.31
factor(topo)3          -1.79     0.28
factor(topo)4          -2.27     0.31
log(seed.weight)       0.54      0.04
  n = 1142, k = 10
  residual deviance = 630.4,
  null deviance = 939.8 (difference = 309.4)

```

<P279>

R code ####8.4

```

seedbank.glm5<-glm(Predation ~ factor(time)+factor(topo)+
  log(seed.weight),
  data=seedbank, family=binomial(link="logit"))

topog <- c("Hilltop","Shady Slope","Sunny Slope","Valley")

betas <- coef(seedbank.glm5)
par(mfrow=c(2,2),mgp=c(1.5, 0.5,0), mar=c(3,3,3,1))
plot(jitter(Predation) ~ log(seed.weight),
  type="n", data=seedbank, xlab="log seed weight",
  ylab="prob. of predation")
points(jitter(seedbank$Predation) ~ log(seedbank$seed.weight),
  col="gray", subset=topo==1)
curve(invlogit(betas[1]+betas[10]*x), add=T, col=gray(.1))
curve(invlogit(betas[1]+betas[2]+betas[10]*x), add=T,
  lty=2, col=gray(.2))
curve(invlogit(betas[1]+betas[3]+betas[10]*x), add=T,
  lty=3, col=gray(.3))
curve(invlogit(betas[1]+betas[4]+betas[10]*x), add=T,
  lty=4, col=gray(.4))

```

```

curve(invlogit(betas[1]+betas[5]+betas[10]*x), add=T,
      lty=5, col=gray(.5))
curve(invlogit(betas[1]+betas[6]+betas[10]*x), add=T,
      lty=6, col=gray(.6))
legend(x=0, y=0.9, legend=month.name[seq(1,11,2)],
      lty=1:6, col=gray((1:6)/10), cex=0.5, bty="n")
title(main=topog[1], cex=0.75)
for (i in c(3, 2, 4)){
  plot(jitter(Predation) ~ log(seed.weight),
       type="n", data=seedbank,
       xlab="centered log seed weight",
       ylab="prob. of predation")
  points(jitter(seedbank$Predation) ~ log(seedbank$seed.weight),
        col="gray", subset=topo==i)
  curve(invlogit(betas[1]+betas[10]*x), add=T,
        col=gray(.1))
  curve(invlogit(betas[1]+betas[2]+betas[i+5]+betas[10]*x),
        add=T, lty=2, col=gray(.2))
  curve(invlogit(betas[1]+betas[3]+betas[i+5]+betas[10]*x),
        add=T, lty=3, col=gray(.3))
  curve(invlogit(betas[1]+betas[4]+betas[i+5]+betas[10]*x),
        add=T, lty=4, col=gray(.4))
  curve(invlogit(betas[1]+betas[5]+betas[i+5]+betas[10]*x),
        add=T, lty=5, col=gray(.5))
  curve(invlogit(betas[1]+betas[6]+betas[i+5]+betas[10]*x),
        add=T, lty=6, col=gray(.6))
  title(main=topog[i], cex=0.75)
}

```

<P280>

R output

```

seedbank.glm5 <- glm(Predation ~ factor(time)+factor(topo)*log(seed.weight),
data=seedbank, family=binomial(link="logit"))

```

```
display(seedbank.glm5)
```

```
glm(formula = Predation ~ factor(time) +
     factor(topo) * log(seed.weight),
     family = binomial(link = "logit"), data = seedbank)

              coef.est      coef.se
(Intercept)      -7.07         0.95
factor(time)2      3.21         0.78

factor(time)3      3.60         0.78
factor(time)4      4.08         0.77
factor(time)5      3.93         0.77
factor(time)6      4.14         0.78
factor(topo)2      -0.97         0.67
factor(topo)3      -0.62         0.62
factor(topo)4      -1.05         0.68
log(seed.weight)   0.77         0.11
factor(topo)2:log(seed.weight) -0.31         0.14
factor(topo)3:log(seed.weight) -0.30         0.13
factor(topo)4:log(seed.weight) -0.30         0.14

n = 1142, k = 13
residual deviance = 622.4,
null deviance = 939.8 (difference = 317.3)
```

```
#### R output ####
```

```
seedbank.glm7<-glm(Predation~factor(time)+factor(topo)*log(seed.weight)+factor(grou
nd)-log(seed.weight),data=seedbank, family=binomial(link="logit"))
display(seedbank.glm7)
```

```
glm(formula=Predation~factor(time)+factor(topo)*log(seed.weight) +
     factor(ground)-log(seed.weight),
     family = binomial(link = "logit"), data = seedbank)

              coef.est      coef.se
(Intercept)    -6.8470     0.9855
```

```

factor(time)2          3.3878  0.8119
factor(time)3          3.8065  0.8097
factor(time)4          4.3153  0.8088
factor(time)5          4.1574  0.8089
factor(time)6          4.4228  0.8119
factor(topo)2          -1.0222  0.6847
factor(topo)3          -0.6749  0.6324
factor(topo)4          -1.1058  0.6967
factor(ground)2        -1.3135  0.2294
factor(topo)1:log(weight)  0.8174  0.1101 f
actor(topo)2:log(weight)  0.4822  0.0934
factor(topo)3:log(weight)  0.4969  0.0866
factor(topo)4:log(weight)  0.4867  0.0950
  n = 1142, k = 14
  residual deviance = 586.1,
  null deviance = 939.8 (difference = 353.7)

```

<P282>

R code ####8.5

```

pred.7 <- predict(seedbank.glm7, type="response")
error.rate <- mean((pred.7>0.5 & na.omit(seedbank$Predation)==0) | (pred.7<0.5 &
na.omit(seedbank$Predation)==1))
error.rate

[1] 0.106

```

<P284>

・ データ入力 : arsenic.txt の入力

```

arsenic <- read.table("clipboard", header=TRUE, sep=",", na.strings="NA",
  dec=".", strip.white=TRUE)

```


R code ####8.6

```
ar.m1 <- glm(events ~ conc, data=arsenic, family="poisson")
```

<P285>

R output

```
display(ar.m1, 4)
```

```
glm(formula = events ~ conc,
     family = "poisson", data = arsenic)
             coef.est coef.se
(Intercept)  3.0569    0.0189
conc         -0.0284    0.0005
---
n = 2236, k = 2
residual deviance = 34167.2,
null deviance = 48869.5 (difference = 14702.3)
```

<P286>

R code ####8.7

```
ar.m3 <- glm(events ~ conc + gender + type,
             data=arsenic, family="poisson")
display(ar.m3, 3)
```

```
glm(formula = events ~ conc + gender + type,
     family = "poisson", data = arsenic)
             coef.est  coef.se
(Intercept)   1.752    0.036
conc          -0.028    0.000
gender         0.672    0.027
type          1.383    0.032
---
```

```
n = 2236, k = 4
residual deviance = 31168.3,
null deviance = 48869.5 (difference = 17701.2)
```

<P288>

R code ####8.8

```
As.m4 <- glm(events ~ log(conc+1) + gender + type,
             data=arsenic, offset=log(at.risk), family="poisson")
```

R output

```
display(As.m4, 4)
glm(formula = events ~ log(conc + 1) + gender + type,
     family="poisson", data = arsenic, offset=log(at.risk))
              coef.est  coef.se
(Intercept)   -10.4205   0.0339
log(conc + 1)    0.2759   0.0088
gender          0.5420   0.0270
type           1.3830   0.0320
---
n = 2236, k = 4
residual deviance = 13989.5,
null deviance = 17398.7 (difference = 3409.3)
```

<P291>

R code ####8.9

```
As.yhat <- predict(As.m4, type="response")
As.z <- (arsenic$events - As.yhat)/sqrt(As.yhat)
overD <- sum(As.z^2)/summary(As.m4)$df[2]
p.value <- 1-pchisq(sum(As.z^2), summary(As.m4)$df[2])
```

<P293>

R code ####8.10

```
As.m5 <- glm(events ~ log(conc+1) + gender + type,  
             data=arsenic, offset=log(at.risk),  
             family="quasipoisson")
```

R output

```
display(As.m5, 4)
```

```
glm(formula = events ~ log(conc + 1) + gender + type,  
     family = "quasipoisson",  
     data = arsenic, offset = log(at.risk))
```

	coef.est	coef.se
(Intercept)	-10.4205	0.1277
log(conc + 1)	0.2759	0.0330
gender	0.5420	0.1019
type	1.3830	0.1203

```
n = 2236, k = 4  
residual deviance = 13989.5,  
null deviance = 17398.7 (difference = 3409.3)  
overdispersion parameter = 14.2
```

<P294>

R code ####8.11

```
As.m6 <- glm(events ~ log(conc+1)*gender*type,  
             data=arsenic, offset=log(at.risk),  
             family="poisson")
```

R output

```
display(As.m6, 4)
```

```
glm(formula = events ~ log(conc + 1) * gender * type,  
     family = "poisson",  
     data = arsenic, offset = log(at.risk))
```

	coef.est	coef.se
(Intercept)	-10.3889	0.0501
log(conc + 1)	0.4563	0.0203
gender	0.3732	0.0635
type	1.3070	0.0565
log(conc + 1):gender	-0.0858	0.0285
log(conc + 1):type	-0.1761	0.0264
gender:type	0.2628	0.0709
log(conc + 1):gender:type	-0.0098	0.0364

```
n = 2236, k = 8  
residual deviance = 13844.2,  
null deviance = 17398.7 (difference = 3554.5)
```

R code ####8.12

```
As.m7 <- update(As.m6, .~. -log(conc+1):gender:type)  
display(As.m7)
```

```
glm(formula = events ~ log(conc + 1) + gender +  
     type + log(conc + 1):gender +  
     log(conc + 1):type + gender:type,  
     family = "poisson",  
     data = arsenic, offset = log(at.risk))
```

	coef.est	coef.se
(Intercept)	-10.39	0.05

log(conc + 1)	0.46	0.02
gender	0.38	0.06
type	1.31	0.05
log(conc + 1):gender	-0.09	0.02
log(conc + 1):type	-0.18	0.02
gender:type	0.26	0.07

```
n = 2236, k = 7
residual deviance = 13844.3,
null deviance = 17398.7 (difference = 3554.4)
```

<P295>

R code ####8.13

```
As.m8 <- update(As.m7, .~., family="quasipoisson")
```

R output

```
display(As.m8, 4)
```

```
glm(formula = events ~ log(conc + 1) + gender +
      type + log(conc + 1):gender +
      log(conc + 1):type + gender:type,
      family = "quasipoisson",
      data = arsenic, offset = log(at.risk))
```

	coef.est	coef.se
(Intercept)	-10.3920	0.1686
log(conc + 1)	0.4593	0.0580
gender	0.3782	0.2095
type	1.3110	0.1885
log(conc + 1):gender	-0.0918	0.0614
log(conc + 1):type	-0.1812	0.0629
gender:type	0.2565	0.2311

```
n = 2236, k = 7
residual deviance = 13844.3,
null deviance = 17398.7 (difference = 3554.4)
overdispersion parameter = 11.9
```

<P296>

R output

```
As.m9 <- update(As.m8, .~.-gender:type)
display(As.m9, 4)
```

```
glm(formula = events ~ log(conc + 1) + gender +
      type + log(conc + 1):gender +
      log(conc + 1):type,
     family = "quasipoisson",
     data = arsenic, offset = log(at.risk))
```

	coef.est	coef.se
(Intercept)	-10.5265	0.1249
log(conc + 1)	0.4691	0.0587
gender	0.5855	0.0977
type	1.4789	0.1178
log(conc + 1):gender	-0.1012	0.0607
log(conc + 1):type	-0.1873	0.0626

```
n = 2236, k = 6
residual deviance = 13858.8,
null deviance = 17398.7 (difference = 3539.9)
overdispersion parameter = 12.0
```

<P297>

R code ####8.14

```

arsenic$age.c1 <- arsenic$age - mean(arsenic$age)
As.m10<-update(As.m9, .~.+age.c1*gender+age.c1:type)
display(As.m10, 4)

glm(formula = events ~ log(conc + 1) + gender +
      type + age.c1 + log(conc + 1):gender +
      log(conc + 1):type + gender:age.c1 +
      type:age.c1,
     family = "quasipoisson", data = arsenic,
     offset = log(at.risk))

```

	coef.est	coef.se
(Intercept)	-10.5859	0.0554
log(conc + 1)	0.4556	0.0205
gender	0.5422	0.0397
type	1.6743	0.0533
age.c1	0.0922	0.0029
log(conc + 1):gender	-0.0921	0.0211
log(conc + 1):type	-0.1873	0.0218
gender:age.c1	0.0155	0.0022
type:age.c1	-0.0178	0.0029

```

---
n = 2236, k = 9
residual deviance = 2193.1,
null deviance = 17398.7 (difference = 15205.6)
overdispersion parameter = 1.4

```

<P298>

R code ####8.15

```

pred.data <- data.frame(
  conc=rep(seq(0, 1000, 10), 4),
  type=rep(rep(c(0,1), each=101), 2),
  gender=rep(c(0,1), each=202))
pred.age1 <- predict(As.m10, newdata=
  data.frame(pred.data, age.c1= rep(-15, 404),

```

```

        at.risk=rep(100000, 404)),
    type="response")
pred.age2 <- predict(As.m10, newdata=
    data.frame(pred.data, age.c1= rep(0 , 404),
        at.risk=rep(100000, 404)),
    type="response")
pred.age3 <- predict(As.m10, newdata=
    data.frame(pred.data, age.c1= rep( 15, 404),
        at.risk=rep(100000, 404)),
    type="response")

```

R code ####8.16

```

plot.data1 <- data.frame(
    events=c(pred.age1, pred.age2, pred.age3),
    rbind(pred.data, pred.data, pred.data),
    age=rep(c(-15+52.5, 52.5, 52.5+15), each=404))
plot.data1$Type <- "Lung Cancer"
plot.data1$Type[plot.data1$type==0]
    <- "Bladder Cancer"
plot.data1$Gender <- "Male"
plot.data1$Gender[plot.data1$gender==0]<-"Female"

trellis.par.set(theme =
    canonical.theme("postscript", col=FALSE))
trellis.par.set(list(fontsize=list(text=8),
    par.xlab.text=list(cex=1.25),
    add.text=list(cex=1.25),
    superpose.symbol=list(cex=1)))
key <- simpleKey(unique(as.character(plot.data1$age)),
    lines=T, points=F, space = "top", columns=3)
key$text$cex <- 1.25
xyplot(events~log(conc+1) | Type*Gender,
    data=plot.data1, type="l",

```



```

group=plot.data1$age,
key=key,
xlab="As concentration (ppb)",
ylab="Cancer deaths per 100,000",
panel=function(x,y,...){
  panel.xyplot(x,y,lwd=1.5,...)
  panel.grid()
},
scales=list(x=list(
  at=log(c(0, 10, 50, 100, 500, 1000)+1),
  labels=as.character(c(0, 10, 50, 100, 500, 1000))))
)

```

<P304>

R code and output####8.17

```

require(MASS)
As.m5nb<-glm.nb(events ~ log(conc+1)+gender+type+
  offset(log(at.risk)), data=arsenic)
summary(As.m5nb)

glm.nb(formula = events ~ log(conc + 1) + gender + type +
  offset(log(at.risk)), data = arsenic,
  init.theta = 0.228840989818068, link = log)

```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.838	-0.678	-0.532	-0.303	3.192

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-8.8636	0.2306	-38.44	< 2e-16
log(conc + 1)	0.2432	0.0395	6.16	7.4e-10
gender	0.1810	0.1255	1.44	0.14904
type	0.4545	0.1259	3.61	0.00031

Null deviance: 1239.9 on 2235 degrees of freedom
Residual deviance: 1197.6 on 2232 degrees of freedom
AIC: 3328

Number of Fisher Scoring iterations: 1

Theta: 0.2288

Std. Err.: 0.0228

2 x log-likelihood: -3318.3190

<P305>

R code #####8.18 thetaの値のみ表示

```
summary(As.m5nb)$theta
```

```
[1] 0.228841
```

<P312>

・ データの入力 : antarctic.txt の入力

```
whale.data <- read.table("clipboard", header=TRUE, sep=",", na.strings="NA",  
  dec=".", strip.white=TRUE)
```

R code #####8.19

```
library(rpart)
```

```
TW.rpart2 <- rpart(I(TW>0) ~ bathy+chl+a+D.coast+D.ice+  
  D.inswb+D.slp+S.bathy+  
  W.mass+A.v100+A.v300.2,  
  data=whale.data, method="class",  
  parms=list(prior=c(0.5,0.5)), cp=0.00)
```

R code #####8.20

```

library(gam)
library(mgcv)
whale.gam1 <- gam(TW~s(A.v100,bs="ts")+
                 s(chla,bs="ts")+
                 s(bathy,bs="ts")+
                 s(D.ice,bs="ts")+
                 s(D.coast,bs="ts"),
                 data=whale.data, family="poisson")

par(mfrow=c(3,2), mar=c(4,4,0.5,0.5))
plot(whale.gam1, scale=0, pages=0, select=1,
     xlab="Backscatter 25-100m", ylab="f(x)",
     residuals=T, shade=T, lwd=2, pch=1, cex=0.5)
plot(whale.gam1, scale=0, pages=0, select=2,
     xlab="Chlorophyll a", ylab="f(x)",
     residuals=T, shade=T, lwd=2, pch=1, cex=0.5)
plot(whale.gam1, scale=0, pages=0, select=3,
     xlab="Bathymetry", ylab="f(x)",
     residuals=T, shade=T, lwd=2, pch=1, cex=0.5)
plot(whale.gam1, scale=0, pages=0, select=4,
     xlab="Dist. to ice edge", ylab="f(x)",
     residuals=T, shade=T, lwd=2, pch=1, cex=0.5)
plot(whale.gam1, scale=0, pages=0, select=5,
     xlab="Dist. to shore", ylab="f(x)",
     residuals=T, shade=T, lwd=2, pch=1, cex=0.5)

```

<P315>

R code ####8.21

```

yhat <- predict(whale.gam1, type="response")
z <- (whale.data$TW - yhat)/sqrt(yhat)
overD <- sum(z^2)/summary(whale.gam1)$residual.df
p.value <- 1-pchisq(sum(z^2),
                   summary(whale.gam1)$residual.df)

```

```

plot(yhat, whale.data$TW - yhat,
     xlab="Predicted Values", ylab="Residuals",
     main="Raw Residuals")
abline(h=c(-2,2), lty=2)
plot(yhat, z, xlab="Predicted Values",
     ylab="Residuals", main="Standardized Residuals")
abline(h=c(-2,2), lty=2)

```

<P316>

R code ####8.22

```

whale.gam3 <- gam(I(TW>0)~s(A.v100,bs="ts")+
                 s(chla,bs="ts")+
                 s(bathy,bs="ts")+
                 s(D.ice,bs="ts")+
                 s(D.coast,bs="ts"),
               data=whale.data, family="binomial")
summary(whale.gam3)

```

Chapter 9

<P324>

R code ####9.1

```

set.seed(1)
t.sample <- rt(10000, df=23)
p.value <- mean(t.sample > 2.34)
p.value

```

<P328>

R code ####9.2

```

y <- log(rlnorm(25, 1.9, 1))

```

```
n.sim <- 5000
n <- length(y)
y.bar <- mean(y)
s.hat <- sd(y)
theta.i <- s.hat * sqrt((n-1)/rchisq(n.sim, n-1))
mu.i <- rnorm(n.sim, y.bar, theta.i/sqrt(n))
y.tilde <- rnorm(n.sim, mu.i, theta.i)
```

R code ####9.3

```
Pr <- mean(y.tilde > log(10))
```

<P329>

R code ####9.4

```
x.tilde <- exp(y.tilde)
```

R code ####9.5

```
mu.x <- mean(x.tilde)
sigma.x <- sd(x.tilde)
```

R code ####9.6

```
CI.x <- mu.x + qt(c(0.025,0.975), df=25-1) * sigma.x/sqrt(25-1)
CI.x
```

<P330>

R code ####9.7 : crypto.lm1 の場合

```
summ <- summary(crypto.lm1)
coef <- summ$coef[,1:2]
sigma.hat <- summ$sigma
```

```

beta.hat <- coef[,1]
V.beta <- summ$cov.unscaled
n <- summ$df[1] + summ$df[2]
p <- summ$df[1]

chi2 <- rchisq(1, n-p)

sigma <- sigma.hat*sqrt((n-p)/chi2)

beta <- mvrnorm(1, beta.hat, V.beta*sigma^2)

y.tilde <- rnorm(1, beta[1]+beta[2]*x.tilde, sigma)

```

<P331>

R code ####9.8

```

n.sims<-1000
sim.results <- sim(lake.lm1, n.sims)

```

R output

```

sim.results$coef[1:10,]

(Intercept) I(year - 1974)
[1,] 1.7053 -0.063053
[2,] 1.5584 -0.058869
[3,] 1.5538 -0.051409
[4,] 1.6369 -0.057951
[5,] 1.7868 -0.073736
[6,] 1.6514 -0.059717
[7,] 1.5680 -0.055303
[8,] 1.5634 -0.056510
[9,] 1.6884 -0.062708
[10,]1.5310 -0.054117

```

R code ####9.9

```
log.PCB <- rnorm(n.sims, sim.results$coef[,2] +  
                sim.results$sigma*(2007-1974),  
                sim.results$sigma)
```

R code ####9.10

```
d.f <- summary(lake.lm1)$df[2]  
sigma.hat <- summary(lake.lm1)$sigma  
mean(log.PCB) + qt(c(0.025,0.975), d.f)*sigma.hat
```

```
[1] -2.0954 1.3544
```

R output

```
quantile(log.PCB, prob=c(0.025,0.25,0.5,0.75,0.975))
```

```
      2.5%    25%    50%    75%    97.5%  
-2.02503 -0.92517 -0.36001 0.25094 1.39737
```

<P332>

R code ####9.11

```
predict.PCB <- exp(log.PCB)
```

R code ####9.12

```
quantile(predict.PCB, prob=c(0.025,0.25,0.5,0.75,0.975))
```

```
      2.5%    25%    50%    75%    97.5%  
0.13199 0.39646 0.69767 1.28524 4.04462
```

<P333>

R code ####9.13

```
n.sims<-1000
sim.results <- sim(lake.lm1, n.sims=1000)
predict.PCB07 <- exp(sim.results$coef[,2] +
                    sim.results$sigma*(2007-1974) +
                    0.5* sim.results$sigma)
predict.PCB00 <- exp(sim.results$coef[,2] +
                    sim.results$sigma*(2000-1974) +
                    0.5 * sim.results$sigma)
percentages <- 1-predict.PCB07/predict.PCB00
hist(percentages)
```

<P337>

・データ入力 : sparrow.txt の入力

```
sparrow <- read.table("clipboard", header=TRUE, sep="", na.strings="NA",
                    dec=".", strip.white=TRUE)
```

R code and output ####9.14

```
spar.glm1 <- glm(Bird.Count ~ factor(year),
                data=sparrow, family=poisson)
display(spar.glm1)
```

```
glm(formula = Bird.Count ~ factor(year),
     family = poisson, data = sparrow)
```

	coef.est	coef.se
(Intercept)	-0.59	0.11
factor(year)1992	0.13	0.14
factor(year)1993	0.32	0.14


```

factor(year)1994      0.82      0.19
factor(year)1995      0.05      0.17
factor(year)1996     -0.03      0.14
factor(year)1997      0.55      0.13
factor(year)1998      0.23      0.15
factor(year)1999      0.09      0.14
factor(year)2000      0.32      0.17
factor(year)2001      0.25      0.19
factor(year)2002     -0.03      0.28
factor(year)2003     -0.44      0.32
factor(year)2004     -0.34      0.35

```

```

n = 1723, k = 14
residual deviance = 2947.8,
null deviance = 3008.8 (difference = 61.0)

```

<P339>

R code ####9.15

```

n <- dim(sparrow)[1]
y.rep <- rpois(n, predict(spar.glm1,type="response"))
zeros <- mean(y.rep==0)
zeros

```

<P341>

R code ####9.16-OKimpossible

```

boot.sample <- function(data) ## data must be a data frame
  data [sample(nrow(data), rep=T),]

```

<P344>

R code ####9.17-OKimpossible

```

my.bagging <- function(obj,
  data=eval(obj$call$data), n.bags=500, ...){
  bags.list <- list()
  for (i in 1:n.bags)
    bags.list[[i]] <- update(obj,
      data=boot.sample(data))
  oldClass(bags.list) <- "bagrpart"
  return(bags.list)}

```

<P342>

R code ####9.18-OKimpossible

```

predict.bagrpart <- function(obj, newdata, ...)
  apply(sapply(obj, predict, newdata=newdata), 1, mean)

```

<P344>

R code ####9.19

```

## infile is a data frame with two columns
## Y and X
chngp <- function(infile)
{
  temp <- na.omit(infile)
  yy <- temp$Y
  xx <- temp$X
  mx <- sort(unique(xx))
  m <- length(mx)
  vi <- numeric()
  vi [m] <- sum((yy - mean(yy))^2)
  for(i in 1:(m-1))
    vi[i] <- sum((yy[xx <= mx[i]] - mean(yy[xx <=
      mx[i]]))^2) + sum((yy[xx > mx[i]] - mean(
        yy[xx > mx[i]]))^2)
  thr <- mean(mx[vi == min(vi)])
}

```

```

    return(thr)
}

```

R code ####9.20

```

my.bootCIs <-
function (x, nboot, theta, ...,
         alpha = c(0.05, 0.95))
{
  n <- length(x)
  thetahat <- theta(x, ...)
  bootsam <- matrix(sample(x, size = n * nboot,
                          replace = TRUE), nrow = nboot)
  thetastar <- apply(bootsam, 1, theta, ...)
  confpoints.percent <- quantile(thetastar, alpha)
  return(confpoints.percent)
}

```

R code ####9.21

```

size <- 25
x.unif <- runif(size, 5, 45)

data.file <- data.frame(
  X=x.unif,
  Y=ifelse(x.unif<25,
           rnorm(sum(x.unif<25), -1),
           rnorm(sum(x.unif>=25),0.5)))
CIs <- my.bootCIs(1:size, nboot=5000,
                 theta=function(x, infile){
                   chngp(infile[x, ])},
                 infile=data.file)

CIs

```

<P345>

R code #### 9.22

```
cover <- CIs[1] < 25 & CIs[2] > 25  
cover
```

Chapter 10

<P353>

```
y ~ Treatment T
```

```
y ~ 1 + Treatment
```

```
y ~ 1 + (1|Treatment)
```

・データ入力：seaweed.txtの入力

```
#### DATA ####
```

```
seaweed <- read.table("clipboard", header=TRUE, sep=",", na.strings="NA",  
  dec=".", strip.white=TRUE)
```

・以下のRコードも実行しておく

```
seaweed$y <- logit(seaweed$COVER/100)  
names(seaweed)[2:3] <- c("Block", "Treatment")  
seaweed$Treatment <- ordered(seaweed$Treatment,  
  levels=c("CONTROL", "f", "fF", "L", "Lf", "LfF"))
```

<P354>

R code ####10.1

```
seaweed.lmer <- lmer(COVER ~ 1+(1|Treatment), data=seaweed)
```

R output

```
summary(seaweed.lmer)
```

```
Linear mixed model fit by REML
Formula: COVER ~ 1 + (1 | TREAT)
Data: seaweed
AIC BIC logLik deviance REMLdev
310 317 -152 304 304
Random effects:
```

Groups	Name	Variance	Std.Dev.
Treatment	(Intercept)	1.14	1.07
Residual		1.18	1.09

Number of obs: 96, groups: Treatment, 6

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	-1.233	0.449	-2.74

<P355>

R output

```
ranef(seaweed.lmer)
```

```
$Treatment
```

```
(Intercept)
CONTROL 1.33
f 0.86
fF 0.39
L -0.45
Lf -0.72
LfF -1.40
```

R output

```
se.ranef(seaweed.lmer)
```

```
$Treatment
      [,1]
[1,] 0.26
[2,] 0.26
[3,] 0.26

[4,] 0.26
[5,] 0.26
[6,] 0.26
```

<P361>

・データ入力：n2oemissioncomp.txt

```
#### data ####
```

```
N2O.data <- read.table("clipboard", header=TRUE, sep="", na.strings="NA",
  dec=".", strip.white=TRUE)
```

R code ####10.2

```
carbon.group <- tapply(N2O.data$carbon/100,
  N2O.data$group,
  mean, na.rm=T)
carbon.full <- carbon.group[N2O.data$group]

bckg.lmer2 <- lmer(log(emission) ~ 1 + logit(carbon.full) + (1|group),
  data=N2O.data)
```

<P363>

R code ####10.3

```
seaweed.lmer2 <- lmer(COVER ~ 1+(1|Treatment)+(1|Block),
                     data=seaweed)
```

R output

```
summary(seaweed.lmer2)
```

Linear mixed model fit by REML

Formula: COVER ~ 1 + (1 |Treatment) + (1 | Block)

Data: seaweed

AIC	BIC	logLik	deviance	REMLdev
-----	-----	--------	----------	---------

229	239	-110	221	221
-----	-----	------	-----	-----

Random effects:

Groups	Name	Variance	Std.Dev.
--------	------	----------	----------

Block	(Intercept)	0.878	0.937
-------	-------------	-------	-------

Treatment	(Intercept)	1.190	1.091
-----------	-------------	-------	-------

Residual		0.359	0.599
----------	--	-------	-------

Number of obs: 96, groups: Block, 8; Treatment, 6

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	-1.233	0.558	-2.21

<P364>

R output

```
ranef(seaweed.lmer2)$Block
```

(Intercept)

BLOCK 1 -1.36

BLOCK 2 -0.91

BLOCK 3 0.68

BLOCK 4 1.53

BLOCK 5 -0.18

BLOCK 6 0.60

```
BLOCK 7 -0.28
BLOCK 8 -0.07
```

```
$Treatment
      (Intercept)
CONTROL  1.39
f        0.90
fF       0.40
L       -0.47
Lf      -0.76
LfF     -1.46
```

R code ####10.4

```
sims.M2 <- mcmcscamp(seaweed.lmer2, n=10000, saveb=T)
```

<P365>

R output

```
str(sims.M2)
Formal class 'merMCMC' [package "lme4"] with 9 slots

..@ Gp : int [1:3] 0 8 14
..@ ST : num [1:2, 1:10000] 1.56 1.82 1.49 1.61 1.28 ...
..@ call : language lmer(formula = y~1+(1 | Treatment) +
                        (1 | Block),
                        data = seaweed)
..@ deviance: num [1:10000] 221 221 221 222 223 ...
..@ dims    :Namedint[1:14]2 96 1 14 1 2 1 2 5 1...
.. ..- attr(*, "names")= chr [1:14] "nf" "n" "p" "q" ...
..@ fixef : num [1, 1:10000] -1.23 -1.01 -1.48 -1.55 ...
.. ..- attr(*, "dimnames")=List of 2
.. .. ..$ : chr "(Intercept)"v
.. .. ..$ : NULL
```



```
..@ nc :int[1:2]11
..@ ranef : num [1:14, 1:10000] -1.357 -0.912 0.678 ...
..@ sigma: num [1, 1:10000] 0.599 0.531 0.621 0.646 ...
```

R code ####10.5

```
block.mcmc <- sims.M2@ranef[1:8,5001:10000]
treat.mcmc <- sims.M2@ranef[9:14,5001:10000]
```

R code ####10.6

```
sigma.block <- apply(block.mcmc, 2, sd)
sigma.treat <- apply(treat.mcmc, 2, sd)
sigma <- sims.M2@sigma[5001:10000]
```

R code ####10.7

```
s.sum <- rbind(
  quantile(sigma, prob=c(0.025,0.25,0.5,0.75,0.975)),
  quantile(sigma.treat, prob=c(0.025,0.25,0.5,0.75,0.975)),
  quantile(sigma.block, prob=c(0.025,0.25,0.5,0.75,0.975)))

plot(c(0,1), c(0.75,3.25), xlim=range(s.sum), type="n",
     xlab="standard deviation", ylab="", axes=F)
abline(h=0, col="gray")
segments(x0=s.sum[,1], x1=s.sum[,5], y0=1:3, y1=1:3)

segments(x0=s.sum[,2], x1=s.sum[,4], y0=1:3, y1=1:3, lwd=3)
axis(1)
axis(2, at=1:3, labels=c("Residuals","Treatment","Block"), las=1)
points(x=s.sum[,3], y=1:3, pch=16, cex=1.25)
```

<P366>

R code ####10.8

```
seaweed.lmer3 <- lmer(COVER~1 + (1|Treatment)+(1|Block)+
                    (1|Treatment:Block), data=seaweed)
```

<P372>

・データ入力：rtol.txt

```
##### data #####
```

```
rtol2 <- read.table("clipboard", header=TRUE, sep=",", na.strings="NA",
                  dec=".", strip.white=TRUE)
```

```
##### R code #####10.9
```

```
esue.lm1 <- lm(richtol ~ nuii, data=rtol2)
display(esue.lm1, 4)
```

```
lm(formula = richtol ~ nuii, data = rtol2)
      coef.est  coef.se
```

```
(Intercept) 5.5433 0.0752
```

```
nuii          0.0140 0.0021
```

```
---
```

```
n = 261, k = 2
```

```
residual sd = 0.7963, R-Squared = 0.15
```

```
##### R code #####10.10
```

```
euse.lm2 <- lm(richtol ~ nuii*factor(city)-1-nuii,
              data=rtol2)
```

```
display(euse.lm2, 4)
```

```
lm(formula = richtol ~ nuii * factor(city) - 1 - nuii,
    data = rtol2)
```

	coef.est	coef.se
factor(city)ATL	5.3318	0.1355
factor(city)BIR	5.1228	0.1544
factor(city)BOS	4.2486	0.1392
factor(city)DEN	6.1978	0.1499
factor(city)DFW	7.0704	0.1167

```

factor(city)MGB      6.0501    0.1227
factor(city)POR      4.5529    0.1305
factor(city)RAL      5.5340    0.1543
factor(city)SLC      4.5080    0.1936
nuii:factor(city)ATL 0.0301    0.0056
nuii:factor(city)BIR 0.0269    0.0053
nuii:factor(city)BOS 0.0455    0.0062

nuii:factor(city)DEN 0.0025    0.0034
nuii:factor(city)DFW -0.0019    0.0033
nuii:factor(city)MGB 0.0078    0.0033
nuii:factor(city)POR 0.0233    0.0035
nuii:factor(city)RAL 0.0250    0.0044
nuii:factor(city)SLC 0.0248    0.0037
---
n = 261, k = 18
residual sd = 0.4744, R-Squared = 0.99

```

<P374>

R code ####10.11

```

euse.lmer1 <- lmer(richtol ~ nuii+(1+nuii|city),
                   data=rtol2)

```

R output

```

summary(euse.lmer1)

```

Linear mixed model fit by REML

Formula: richtol ~ nuii + (1 + nuii | city)

Data: rtol2

AIC	BIC	logLik	deviance	REMLdev
424	445	-206	401	412

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
City	(Intercept)	0.817228	0.9040	
	nuii	0.000188	0.0137	-0.893
residual		0.225311	0.4747	

Number of obs: 261, groups: city, 9

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	5.41839	0.30516	17.76
nuii	0.01943	0.00479	4.06

Correlation of Fixed Effects:

	(Intr)
nuii	-0.877

<P375>

R output

```
fixef(euse.lmer1)
```

(Intercept)	nuii
5.418387	0.019431

R output

```
se.fixef(euse.lmer1)
```

(Intercept)	nuii
0.3051636	0.0047898

<P376>

R output

```
ranef(euse.lmer1)$city
```

```
      (Intercept)  nuui
ATL   -0.030748   0.0067451
BIR   -0.266135   0.0060253
BOS   -1.079855   0.0206173
DEN    0.744879  -0.0156565
DFW    1.626480  -0.0210608
MGB    0.614451  -0.0109547
POR   -0.864952   0.0049036
RAL    0.147261   0.0038998
SLC   -0.891381   0.0054811
```

```
se.ranef(euse.lmer1)$city
```

```
      [,1]      [,2]
[1,] 0.12689   0.0046299
[2,] 0.14515   0.0045649
[3,] 0.12779   0.0049203
[4,] 0.14732   0.0032192
[5,] 0.11545   0.0031238
[6,] 0.12096   0.0031427
[7,] 0.12838   0.0032621
[8,] 0.14854   0.0039813
[9,] 0.18847   0.0035250
```

<P378>

R output

```
AveTemp
```

```
ATL   BIR   BOS   DEN   DFW   MGB   POR   RAL   SLC
16.27 16.00  8.71  9.19 18.30  7.63 10.81 14.93  9.73
```

・以下の R コードを実行しておく.

```
AveTemp <- c(16.27,16.00,8.71,9.19,18.30,7.63,10.81,14.93,9.73)
```

R code ####10.12

```
site <- as.numeric(ordered(rtol2$city))
temp.full <- AveTemp[site]
```

R code ####10.13

```
y ~ x + G1 + G2:x + (1+x|group)
```

<P379>

R code ####10.14

```
euse.lmer2 <- lmer(richtol ~ nuii+temp.full+nuii:temp.full+
                  (1+nuii|city), data=rtol2)
```

R output

```
summary(euse.lmer2)
```

Linear mixed model fit by REML

```
Formula: richtol ~ nuii + temp.full) + nuii:temp.full +
        (1 + nuii | city)
```

Data: rtol2

AIC	BIC	logLik	deviance	REMLdev
440	469	-212	396	424

Random effects:

Groups	Name	Variance	Std.Dev	Corr
city	(Intercept)	0.789371	0.8885	
	nuii	0.000207	0.0144	-0.932
Residual		0.225589	0.4750	

Number of obs: 261, groups: city, 9

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	4.319222	1.039736	4.15

nuii	0.023160	0.017280	1.34
temp.full	0.088587	0.080268	1.10
nuii:temp.full	-0.000298	0.001338	-0.22

Correlation of Fixed Effects:

	(Intr)	nuii	tmp.fl
nuii	-0.918		
temp.full	-0.957	0.879	
nui:tmp.fl	0.877	-0.957	-0.916

<P380>

R code and output ####10.15

```
site <- as.numeric(ordered(rtol2$city))
ag <- cbind(c(rtol2$city),c(rtol2$agr))
cag <- tapply(ag[,2],ag[,1],mean)
ag.full <- as.vector(cag[site])
euse.lmer3 <- lmer(richtol~nuii+ag.full+nuii:ag.full+(1+nuii|site),
                 data=rtol2)
summary(euse.lmer3)
```

Linear mixed model fit by REML

Formula: richtol ~ nuii + ag.full + nuii:ag.full + (1 + nuii | site)

Data: rtol2

AIC	BIC	logLik	deviance	REMLdev
421	449	-202	384	405

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
site	(Intercept)	1.84e-01	0.42944	
	nuii	2.74e-05	0.00524	-0.343
Residual		2.25e-01	0.47462	

Number of obs: 261, groups: site, 9

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	4.45845	0.24164	18.45
nuii	0.03412	0.00366	9.31
ag.full	2.52938	0.49390	5.12
nuii:ag.full	-0.03884	0.00707	-5.50

Correlation of Fixed Effects:

	(Intr)	nuii	ag.fll
nuii	-0.419		
ag.full	-0.781	0.319	
nuii:ag.fll	0.337	-0.792	-0.406

<P382>

R code ####10.16

```
ag.full <- as.vector(ag[,2]/100)
```

```
ag.cat <- ag.full>0.5
```

```
euse.lmer3 <- lmer(richtol ~ nuii+temp.full+nuii:temp.full+  
                  (1+nuii|city)+(1+nuii|ag.cat),  
                  data=rtol2)
```

<P383>

R code ####10.17

```
round(fixef(euse.lmer3), 4)
```

(Intercept)	nuii	temp.full	nuii:temp.full
4.2663	0.0224	0.1143	-0.0006

R output

```
ranef(euse.lmer3)
```

```
$city
```

	(Intercept)	nuii
1	0.070602	0.00174688
2	-0.053474	-0.00132310
3	0.073577	0.00182050
4	-0.010953	-0.00027101
5	-0.065197	-0.00161315
6	0.079344	0.00196320
7	-0.141378	-0.00349809
8	0.157664	0.00390104
9	-0.110185	-0.00272627

```
$ag.cat
```

	(Intercept)	nuii
FALSE	-0.82269	0.012530
TRUE	0.82269	-0.012530

<P384>

R code ####10.18

```
euse.lmer4 <- lmer(richtol ~ nuii+temp.full+nuii:temp.full+
                    as.numeric(ag.cat)+
                    as.numeric(ag.cat):nuii +(1+nuii|ag.cat),
                    data=rtol2)
```

<P385>

R code ####10.19

```
euse.lmer5 <-
  lmer(richtol ~ nuii+temp.full+nuii:temp.full+
```

```

as.numeric(ag.cat)+ as.numeric(ag.cat):nuii +
as.numeric(ag.cat):temp.full +
as.numeric(ag.cat):temp.full:nuii +
(1+nuii|site), data=rtol2)

```

R output

```
summary(euse.lmer5)
```

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	3.102764	0.302048	10.27
nuii	0.036322	0.011794	3.08
temp.full	0.140127	0.022961	6.10
as.numeric(ag.cat)	2.210949	0.386754	5.72
nuii:temp.full	-0.000657	0.000915	-0.72
nuii:as.numeric(ag.cat)	-0.024566	0.014796	-1.66
temp.full:as.numeric(ag.cat)	-0.044174	0.029523	-1.50
nuii:temp.full:as.numeric(ag.cat)	-0.000109	0.001157	-0.09

<P386>

R code ####10.20

```

euse.lmer6 <- lmer(richtol ~ nuii+temp.full +
  nuii:temp.full+
  (1+nuii|site)+
  (1+nuii+ temp.full+nuii:temp.full|ag.cat),
  data=rtol2)

```

R output

```
ranef(euse.lmer6)$site
```

...

...

\$ag.cat

	(Intercept)	nuii	temp.full	nuii:temp.full
FALSE	-1.0848	0.011253	0.020961	0.00011844
TRUE	1.0848	-0.011253	-0.020961	-0.00011844

<P387>

・データ入力 : summerAll2.txt

data

```
summer.All <- read.table("clipboard", header=TRUE, sep="", na.strings="NA",
  dec=".", strip.white=TRUE)
```

R code #####10.21

```
Finn.M3 <- lmer(chla ~ totp+totn+totp:totn+(1+totp+totn+totp:totn | type),
  data=summer.All)
```

R output

```
summary(Finn.M3)
```

Linear mixed model fit by REML

Formula: $y \sim \text{chla} \sim \text{totp} + \text{totn} + \text{totp}:\text{totn} + (1 + \text{totp} + \text{totn} + \text{totp}:\text{totn} \mid \text{type})$

Data: summer.All

	AIC	BIC	logLik	deviance	REMLdev
	29374	29492	-14672	29325	29344

Random effects:

Groups	Name	Variance	Std.Dev	Corr		
type	(Intercept)	0.0139	0.118			
	totp	0.0177	0.133	-0.694		
	totn	0.0631	0.251	0.534	-0.828	
	totp:totn	0.0326	0.181	-0.831	0.451	-0.511

Residual 0.2635 0.513

Residual Number of obs: 19427, groups: type, 9

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	2.2305	0.0400	55.8
totp	0.7641	0.0459	16.7
totn	0.7082	0.0863	8.2
totp:totn	-0.0129	0.0617	-0.2

Correlation of

	(Intr)	totp	totn
totp	-0.666		
totn	0.517	-0.818	
totp:totn	-0.811	0.424	-0.487

<P397>

・データ入力：moths.txt の入力.

```
##### data #####
```

```
moths <- read.table("clipboard", header=TRUE, sep="", na.strings="NA",  
  dec=".", strip.white=TRUE)
```

```
##### R code #####10.22
```

```
library(lme4)
```

```
moths$color <- as.numeric(moths$morph)
```

```
moth.lmer1 <- glmer(cbind(removed, placed-removed) ~  
  color+(1+color|distance),  
  data=moths, family=binomial)
```

```
##### R output #####
```

```
fixef(moth.lmer1)
```

(Intercept)	color
-0.61	-0.37

<P398>

R code ####10.23

```
moth.lmer2 <- glmer(cbind(removed, placed-removed) ~  
                    color*distance+(1+color|distance),  
                    data=moths, family=binomial)
```

<P399>

R output

Coefficients:

	Estimate	Std. Error
(Intercept)	-1.12899	0.19791
color	0.41126	0.27449
distance	0.01850	0.00565
color:distance	-0.02779	0.00809

R output

Fixed effects:

	Estimate	Std. Error
(Intercept)	-1.13079	0.21177
color	0.41063	0.27466
distance	0.01848	0.00612
color:distance	-0.02782	0.00809

<P400>

R code ####10.24

```
moth.lmer3 <- glmer(cbind(removed, placed-removed) ~  
                    distance+(1+distance|morph),
```

```
data=moths, family=binomial)
```

R output

```
coef(moth.lmer3)$morph
```

```
              (Intercept) distance
dark             -1.11         0.0173
light            -0.74        -0.0078
```

<P403>

・データ入力 : cryptodata3.txt の入力.

```
#### data ####
```

```
dcts.data <- read.table("clipboard", header=TRUE, sep="", na.strings="NA",
  dec=".", strip.white=TRUE)
```

R code ####10.25

```
crypto.glm <- glm(n.cT ~ factor(ICR.PWSID), data=dcts.data,
  offset=log(0.44*volume),
  family="poisson")
```

<P404>

R code ####10.26

```
display(crypto.glm)
```

```
glm(formula = n.cT ~ factor(PWSID) - 1,
  family = "poisson", data = dcts.data,
  offset = log(volume * 0.44))
```

	coef.est	coef.se
factor(PWSID)	-22.48	15541.86
factor(PWSID) 010106001	-21.86	15541.86

```

factor(PWSID) 104121115    -21.78    15541.86
factor(PWSID) AK2210906    -4.72     0.58
factor(PWSID) AK2260309    -3.97     0.71
factor(PWSID) AL0000133    -21.78    2590.31
.....
factor(PWSID) WV3304005    -3.90     1.00
factor(PWSID) WV3304104    -4.05     1.00
factor(PWSID) WY5600011    -21.79    3047.51
factor(PWSID) WY5600029    -21.78    7770.93
factor(PWSID) WY5600050    -21.78    15541.86
---
n = 13103, k = 884
residual deviance = 6789.9,
null deviance = 142338.8 (difference = 135548.9)

```

<P405>

R code ####10.27

```

crypto.lmer1 <- glmer(n.cT ~ 1+(1| ICR.PWSID),
                      data=dcts.data, family="poisson",
                      offset=log(volume*0.44))

```

R output

```

coef(crypto.lmer1)]

```

```

$CIR.PWSID
  (Intercept)
1    -5.53
2    -5.47
3    -5.47
4    -4.77
5    -4.12
6    -6.46
7    -5.65

```

```
8    -6.16
9    -6.17
10   -6.17
11   -6.18
12   -3.46
.....
```

<P406>

R code #####10.28

```
mus <- coef(crypto.lmer1)[[1]][,1]
n.sys <- length(mus)
f = ((1:n.sys)-0.5)/n.sys
```

<P407>

R code #####10.29?

注：計算に多少時間がかかる。

```
dcts.size <- as.vector(table(dcts.data$ICR.PWSID))
n.sys <- 380
n.sims <- 10000
zeros <- numeric()
sys.means <- matrix(0, n.sims, n.sys)
for (i in 1:n.sims){
  zeros[i] <- 0
  for (j in 1:n.sys){
    mu <- rnorm(1, -5.384, 0.103)
    sigma <- 2.08*sqrt((13103-884)/rchisq(1, 13103-884))

    mm <- rnorm(dcts.size[j], mu, sigma)
    mm.mean <- mean(mm)
    y <- rpois(dcts.size[j],
              0.44*10*exp(mm.mean))
    sys.means[i,j] <- mean(y)/10
```



```
        zeros[i] <- zeros[i] + (sum(y!=0)==0)/n.sys
    }

}
hist(zeros, xlab="fraction of all zeros systems", main="")
hist(as.vector(apply(sys.means, 1, quantile, prob=0.99)))
```