Norges teknisk-naturvitenskapelige universitet
Institutt for matematiske fag

Bokmål

Faglig kontakt under eksamen: Førsteamanuensis Jarle Tufto
Telefon: 99 70 55 19

Bioberegninger, ST1301

Onsdag 1. juni 2005
Kl. 9–13
Hjelpemidler: Alle trykte og skrevne hjelpemidler, lommeregner.
Sensur: 22. juni 2004

Hjelpesider for noen R-funksjoner som er omhandlet nedenfor eller som du vil kunne få bruk for i programmeringsoppgavene følger på side 5.

Oppgave 1

a) Anta at vi definerer tre vektorer `hoyde`, `kjonn`, og `tronder` på følgende måte i R. Hva blir da verdien av uttrykkene i de siste fire linjene? Hva er tolkningen av det siste uttrykket?

```
hoyde <-   c(1.75, 1.80, 1.84, 1.60, 1.65, 1.70, 1.82)
kjonn <-   c(1,    1,    1,    2,    2,    2,    1   )
tronder <- c(F,    T,    T,    T,    F,    T,    F   )
kjonn==1
kjonn==1 & tronder
hoyde[kjonn==1 & tronder]
mean(hoyde[kjonn==1 & tronder])
```

b) Hvorfor har følgende to uttrykk forskjellig verdi? Hva blir verdien av uttrykkene?

```
1:5*2
1:(5*2)
```

c) Anta at vi definerer følgende funksjon i R.

```
xxx <- function(x) {
   n <- length(x)
   z <- x[1]
   for (i in 2:n) {
      if (x[i]>z) {
         z <- x[i]
      }
   }
   z
}
```

Hva blir verdien av funksjonskallet `xxx(c(1,2,5,-1,2,6,2,-3))`?

Oppgave 2 Anta at vi ønsker å undersøke om det er noen sammenheng mellom artsmangfold og arealet av ulike naturreservat (i km^2). Vi samler inn data på antall arter av større patterdyr i reservat av ulikt areal, legger dataene inn i som to vektorer i R, og bruker så funksjonen `lm` for å tilpasse en regresjonsmodell som vist nedenfor.

```
> areal <- c(4840,3126,2074,565,325,58)
> antall.arter <- c(22,14,17,10,12,5)
> modell <- lm(antall.arter ~ areal)
> summary(modell)

Call:
lm(formula = antall.arter ~ areal)

Residuals:
      1       2       3       4       5       6
 0.4143 -2.8844  3.0011  0.1401  2.7983 -3.4693

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.3102269  1.8215891   4.562   0.0103 *
areal       0.0027429  0.0007245   3.786   0.0193 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.057 on 4 degrees of freedom
Multiple R-Squared: 0.7818,     Adjusted R-squared: 0.7273
F-statistic: 14.33 on 1 and 4 DF,  p-value: 0.01934
```

a) Skriv opp et algebraisk uttrykk for regresjonsmodellen som vi har tilpasset. Hva er estimatene av ukjente parametere i modellen? Hva blir forventet antall arter i et reservat med areal lik 5000km^2?

b) Kan vi forkaste nullhypotesen at arealet av et gitt reservat ikke påvirker antall arter i reservatet til fordel for hypotesen at det faktisk er en sammenheng?

c) Dersom arealet av et reservat går mot null skulle en tro at antall arter nødvendigvis også må gå mot null. Kan denne nullhypotesen forkastes ut i fra modellen vi har tilpasset? I lys av dette, virker modellantakelsene i regresjonsmodellen rimelige?

Oppgave 3 Anta at endringen i størrelsen N til en gitt populasjon er beskrevet ved differensialligningen

$$\frac{dN}{dt} = rN\left(1 - \frac{\ln N}{\ln K}\right). \tag{1}$$

Parameteren r vil her gi uttrykk for graden av tetthetsregulering mens parameteren K vil kunne tolkes som populasjonens bæreevne.

Anta at vi nå høster med en konstant rate c uavhengig av populasjonsstørrelsen slik at endring i populasjonsstørrelse per tidsenhet i stedet blir

$$\frac{dN}{dt} = rN\left(1 - \frac{\ln N}{\ln K}\right) - c. \tag{2}$$

a) På grunn av høsting vil populasjonsstørrelsen nå reduseres og nå en ny likevekt $N = N^*$. Hvilken ligning oppfyller denne likevektspopulasjonsstørrelsen?

b) Sett opp iterasjonsligningen du trenger for å løse denne ligningen ved hjelp av Newtons metode.

c) Programmer en R-funksjon som beregner N^* gitt c, r, og K ved hjelp av Newtons metode.

d) Gi et eksempel på tallverdier for innargument og tilhørende utdata som kan brukes for å teste om R-funksjonen fungerer.

Oppgave 4 Anta at vi har et samfunn bestående av $n = 5$ arter. Vi antar at hver art i dør ut etter en tid T_i som er eksponentiellt fordelt med forventning $\beta = 100$ år. Vi antar også at disse utdøelsestidpunktene er uavhengige tilfeldige variable. På et tidspunkt W vil alle artene (hele samfunnet) ha dødd ut. Vi ønsker å finne forventet tid til utdøelse av hele samfunnet, altså $E(W)$.

a) Dersom $T_1, T_2, \ldots, T_5$ tar verdiene 220, 70, 79, 85 og 160, hvilken verdi tar da den stokastiske variabelen W?

b) Programmer en funksjon som simulerer gjentatte realisasjoner av $T_1, T_2, \ldots, T_n$ og tilhørende verdi av W og som på grunnlag av dette returnerer et estimat av $E(W)$ som funksjonsverdi for vilkårlig valg av β og n. Hint: Se vedlagte hjelpesider.

```
Logic                  package:base                  R Documentation

Logical Operators

Description:

     These operators act on logical vectors.

Usage:

     ! x
     x & y
     x && y
     x | y
     x || y
     xor(x, y)

Arguments:

    x, y: logical vectors, or objects which can be coerced to such or
          for which methods have been written.

Details:

     '!' indicates logical negation (NOT).

     '&' and '&&' indicate logical AND and '|' and '||' indicate
     logical OR.  The shorter form performs elementwise comparisons in
     much the same way as arithmetic operators.  The longer form
     evaluates left to right examining only the first element of each
     vector.  Evaluation proceeds only until the result is determined.
     The longer form is appropriate for programming control-flow and
     typically preferred in 'if' clauses.

     'xor' indicates elementwise exclusive OR.

     Numeric and complex vectors will be coerced to logical values,
     with zero being false and all non-zero values being true.  Raw
     vectors are handled without any coercion for '!', '&' and '|',
     with these operators being applied bitwise (so '!' is the
     1-complement).

     The operators '!', '&' and '|' are generic functions: methods can
     be written for them individually or via the 'Ops') group generic
     function.

     'NA' is a valid logical object.  Where a component of 'x' or 'y'
     is 'NA', the result will be 'NA' if the outcome is ambiguous.  In
     other words 'NA & TRUE' evaluates to 'NA', but 'NA & FALSE'
     evaluates to 'FALSE'.  See the examples below.

References:

     Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New S
     Language_. Wadsworth & Brooks/Cole.

See Also:

     'TRUE' or 'logical'.

     'any' and 'all' for OR and AND on many scalar arguments.

     'Syntax' for operator precedence.

Examples:

     y <- 1 + (x <- rpois(50, lambda=1.5) / 4 - 1)
     x[(x > 0) & (x < 1)]    # all x values between 0 and 1
     if (any(x == 0) || any(y == 0)) "zero encountered"

------------------------------------------------------------------------
Extremes               package:base               R Documentation

Maxima and Minima

Description:

     Returns the (parallel) maxima and minima of the input values.

Usage:

     max(..., na.rm=FALSE)
     min(..., na.rm=FALSE)
     pmax(..., na.rm=FALSE)
     pmin(..., na.rm=FALSE)

Arguments:

     ...: numeric arguments.

   na.rm: a logical indicating whether missing values should be
          removed.

Value:

     'max' and 'min' return the maximum or minimum of all the values
     present in their arguments, as 'integer' if all are 'integer',  or
     as 'double' otherwise.

     The minimum and maximum of an empty set are '+Inf' and '-Inf' (in
     this order!) which ensures _transitivity_, e.g., 'min(x1, min(x2))
     == min(x1,x2)'. In R versions before 1.5, 'min(integer(0)) ==
     .Machine$integer.max', and analogously for 'max', preserving
     argument _type_, whereas from R version 1.5.0, 'max(x) == -Inf'
     and 'min(x) == +Inf' whenever 'length(x) == 0' (after removing
     missing values if requested).

     If 'na.rm' is 'FALSE' an 'NA' value in any of the arguments will
     cause a value of 'NA' to be returned, otherwise 'NA' values are
     ignored.

     'pmax' and 'pmin' take several vectors (or matrices) as arguments
     and return a single vector giving the parallel maxima (or minima)
     of the vectors.  The first element of the result is the maximum
     (minimum) of the first elements of all the arguments, the second
     element of the result is the maximum (minimum) of the second
     elements of all the arguments and so on.  Shorter vectors are
     recycled if necessary.  If 'na.rm' is 'FALSE', 'NA' values in the
     input vectors will produce 'NA' values in the output.  If 'na.rm'
     is 'TRUE', 'NA' values are ignored. 'attributes' (such as 'names'
     or 'dim') are transferred from the first argument (if applicable).

     'max' and 'min' are generic functions: methods can be defined for
     them individually or via the 'Summary' group generic.

References:

     Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New S
     Language_. Wadsworth & Brooks/Cole.

See Also:

     'range' (_both_ min and max) and 'which.min' ('which.max') for the
     _arg min_, i.e., the location where an extreme value occurs.

Examples:

     require(stats)
     min(5:1,pi)
     pmin(5:1, pi)
     x <- sort(rnorm(100));  cH <- 1.35
     pmin(cH, quantile(x)) # no names
     pmin(quantile(x), cH) # has names
     plot(x, pmin(cH, pmax(-cH, x)), type='b', main= "Huber's function")
------------------------------------------------------------------------
Exponential              package:stats              R Documentation

The Exponential Distribution

Description:

     Density, distribution function, quantile function and random
     generation for the exponential distribution with rate 'rate'
     (i.e., mean '1/rate').

Usage:

     dexp(x, rate = 1, log = FALSE)
     pexp(q, rate = 1, lower.tail = TRUE, log.p = FALSE)
     qexp(p, rate = 1, lower.tail = TRUE, log.p = FALSE)
     rexp(n, rate = 1)

Arguments:

    x, q: vector of quantiles.
```

```
       p: vector of probabilities.

       n: number of observations. If 'length(n) > 1', the length is
          taken to be the number required.

    rate: vector of rates.

log, log.p: logical; if TRUE, probabilities p are given as log(p).

lower.tail: logical; if TRUE (default), probabilities are P[X <= x],
          otherwise, P[X > x].

Details:

     If 'rate' is not specified, it assumes the default value of '1'.

     The exponential distribution with rate lambda has density

                     f(x) = lambda e^(- lambda x)

     for x >= 0.

Value:

     'dexp' gives the density, 'pexp' gives the distribution function,
     'qexp' gives the quantile function, and 'rexp' generates random
     deviates.

Note:

     The cumulative hazard H(t) = - log(1 - F(t)) is '-pexp(t, r, lower
     = FALSE, log = TRUE)'.

References:

     Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New S
     Language_. Wadsworth & Brooks/Cole.

See Also:

     'exp' for the exponential function, 'dgamma' for the gamma
     distribution and 'dweibull' for the Weibull distribution, both of
     which generalize the exponential.

Examples:

     dexp(1) - exp(-1) #-> 0
```