Norges teknisk-naturvitenskapelige universitet
Institutt for matematiske fag

Bokmål

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

Bioberegninger, ST1301

Mandag 21. mai, 2007
Kl. 15-19
Hjelpemidler: Alle trykte og skrevne hjelpemidler, lommeregner.
Sensur: 12. juni 2007

Hjelpesider for noen R-funksjoner som er omhandlet nedenfor eller som du vil kunne få bruk for i programmeringsoppgavene følger på side 4.

Oppgave 1

a) Anta at vi definerer følgende to vektorer i R:

```
x <- c(180,185,190,165,160,155)
y <- c(1,1,2,2,2,3)
```

Hva blir da verdien av følgende uttrykk?

```
x[x<mean(x)]
mean(x[y==2])
length(y==2)
length(x[y==2])
```

b) Anta at vi definerer følgende funksjon i R.

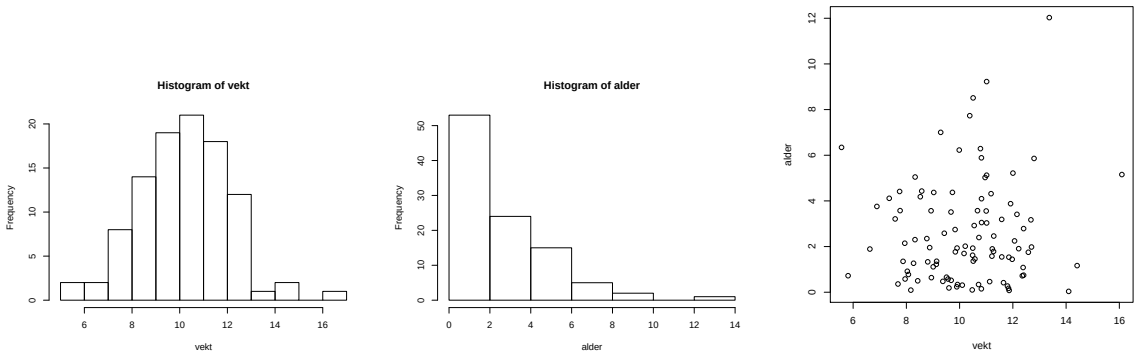
```
funk <- function(x) {
  n <- length(x)
  teller <- 0
  for (i in 2:(n-1)) {
     if (x[i]==x[i+1] & x[i]==x[i-1]) {
        teller <- teller + 1
     }
```

```
  }
  teller
}
```

Hva vil funksjonen da returnere dersom vi gjør følgende funksjonskall?

```
funk(c(0.5,0.5,1,1,1,2,2,1,2,0,0,0,1,1,0,0))
funk(c(0.5,0.5,1,1,1,2,2,1,2,0,0,0,0,1,1,0,0))
```

Oppgave 2 Anta at vi observerer levealder og vekt til 100 individ og at vi legger observasjonene inn i R som to vektorer `alder` og `vekt`. Vi lager så følgende histogram og spredningsplot.



Resultatet av en Pearson korrelasjons-test viser følgende.

```
> cor.test(vekt,alder)
Pearson's product-moment correlation
data:  vekt and alder
t = 0.781, df = 98, p-value = 0.4367
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1196192  0.2708804
sample estimates:
       cor
0.07864678
```

a) Hvilken null-hypotese har vi her testet? Hvilken konklusjon kan vi trekke på grunnlag av hypotesetesten som er utført?

b) Ser forutsetningene som testen bygger på ut til å være oppfyllt?

c) Foreslå en alternativ modell for fordelingen til variabelen alder som du finner mer rimelig.

d) Vi ønsker å undersøke hvor robust korrelasjons-testen over er dersom de observerte vektene er $N(\mu, \sigma^2)$ og alder i virkeligheten er fordelt som foreslått i punkt c.

Programmer en funksjon som simulerer 1000 realisasjoner av dataene under disse forutsetningene og under nullhypotesen om uavhengighet og som beregner sannsynligheten for type I feil, gitt et signifikansnivå α, parameterverdier og utvalgsstørrelsen n.

Oppgave 3 I en populasjon finner vi to varianter A og a av et gitt gen. Hvert individ bærer på to genkopier, en arvet fra far og den andre fra mor.

Under visse forutsetninger kan vi anta at antall kopier X_{t+1} av en genvarianten A i en gitt generasjon, gitt antall kopier X_t i forrige generasjon, vil være binomisk fordelt med parametere $2N$ og $p_t = X_t/(2N)$ hvor N er populasjonsstørrelsen. Dette innebærer at vi har såkalt genetisk drift. p_t kalles genfrekvensen av genvarianten A. Vi antar her at populasjonsstørrelsen N er konstant.

a) Programmer en funksjon som simulerer en realisasjon av $X_2, X_3, ...X_n$ gitt n, X_1 og N og returnerer $X_1, X_2, \ldots, X_n$ som funksjonsverdi. Skisser grafisk hvordan en realisasjon av denne stokastiske prosessen vil kunne se ut.

Genetisk drift fører til tap av genetisk variasjon. Et mål på genetisk variasjon er andelen individer i populasjonen som er bærere av både genvarianten A og a (såkalt heterozygote individ). Anta at det er tilfeldig hvem som parrer seg med hvem.

b) Hvorfor blir andelen heterozygote individ i en gitt generasjon $H_t = 2p_t(1 - p_t)$ dersom genfrekvensen er p_t?

c) Programmer en funksjon som simulerer mange realisasjoner av genfrekvensen p_n (ved hjelp av funksjonen fra punkt a) og tilsvarende heterozygotfrekvens H_t i en gitt generasjon n, gitt N og X_1, og som på grunnlag av av dette beregner et estimat av forventet heterozygotfrekvens $E(H_t)$ i generasjon n.

d) Vil $E(H_n)$ bli mindre, like stor, eller større enn H_1 dersom $p_1 = 1/2$?

e) For modellen over kan det vises at $E(p_n) = p_1$. Bruk dette til å vise at resulatet i punkt d også gjelder for $p_1 \neq 1/2$.

```
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.


------------------------------------------------------------------------
cor.test                 package:stats                 R Documentation

Test for Association/Correlation Between Paired Samples

Description:

     Test for association between paired samples, using one of
     Pearson's product moment correlation coefficient, Kendall's tau or
     Spearman's rho.

Usage:

     cor.test(x, ...)

     ## Default S3 method:
     cor.test(x, y,
              alternative = c("two.sided", "less", "greater"),
              method = c("pearson", "kendall", "spearman"),
              exact = NULL, conf.level = 0.95, ...)

     ## S3 method for class 'formula':
     cor.test(formula, data, subset, na.action, ...)

Arguments:

    x, y: numeric vectors of data values.  'x' and 'y' must have the
          same length.

alternative: indicates the alternative hypothesis and must be one of
          '"two.sided"', '"greater"' or '"less"'.  You can specify just
          the initial letter.  '"greater"' corresponds to positive
          association, '"less"' to negative association.

  method: a character string indicating which correlation coefficient
          is to be  used for the test.  One of '"pearson"',
          '"kendall"', or '"spearman"', can be abbreviated.

   exact: a logical indicating whether an exact p-value should be
          computed.  Used for Kendall's tau and Spearman's rho. See the
          Details for the meaning of 'NULL' (the default).

conf.level: confidence level for the returned confidence interval.
          Currently only used for the Pearson product moment
          correlation coefficient if there are at least 4 complete
          pairs of observations.

 formula: a formula of the form '~ u + v', where each of 'u' and 'v'
          are numeric variables giving the data values for one sample.
          The samples must be of the same length.

    data: an optional matrix or data frame (or similar: see
          'model.frame') containing the variables in the formula
          'formula'.  By default the variables are taken from
          'environment(formula)'.

  subset: an optional vector specifying a subset of observations to be
          used.

na.action: a function which indicates what should happen when the data
          contain 'NA's.  Defaults to 'getOption("na.action")'.

     ...: further arguments to be passed to or from methods.

Details:

     The three methods each estimate the association between paired
     samples and compute a test of the value being zero.  They use
     different measures of association, all in the range [-1, 1] with 0
     indicating no association.  These are sometimes referred to as
     tests of no _correlation_, but that term is often confined to the
     default method.

     If 'method' is '"pearson"', the test statistic is based on
     Pearson's product moment correlation coefficient 'cor(x, y)' and
     follows a t distribution with 'length(x)-2' degrees of freedom if
     the samples follow independent normal distributions.  If there are
     at least 4 complete pairs of observation, an asymptotic confidence
     interval is given based on Fisher's Z transform.

     If 'method' is '"kendall"' or '"spearman"', Kendall's tau or
     Spearman's rho statistic is used to estimate a rank-based measure
     of association.  These tests may be used if the data do not
     necessarily come from a bivariate normal distribution.

     For Kendall's test, by default (if 'exact' is NULL), an exact
     p-value is computed if there are less than 50 paired samples
     containing finite values and there are no ties.  Otherwise, the
     test statistic is the estimate scaled to zero mean and unit
     variance, and is approximately normally distributed.

     For Spearman's test, p-values are computed using algorithm AS 89.

Value:

     A list with class '"htest"' containing the following components:

statistic: the value of the test statistic.

parameter: the degrees of freedom of the test statistic in the case
          that it follows a t distribution.

 p.value: the p-value of the test.

estimate: the estimated measure of association, with name '"cor"',
          '"tau"', or '"rho"' correspoding to the method employed.

null.value: the value of the association measure under the null
          hypothesis, always '0'.

alternative: a character string describing the alternative hypothesis.

  method: a character string indicating how the association was
          measured.

data.name: a character string giving the names of the data.
```

```
conf.int: a confidence interval for the measure of association.
          Currently only given for Pearson's product moment correlation
          coefficient in case of at least 4 complete pairs of
          observations.

Examples:

     ## Hollander & Wolfe (1973), p. 187f.
     ## Assessment of tuna quality.  We compare the Hunter L measure of
     ##  lightness to the averages of consumer panel scores (recoded as
     ##  integer values from 1 to 6 and averaged over 80 such values) in
     ##  9 lots of canned tuna.

     x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
     y <- c( 2.6,  3.1,  2.5,  5.0,  3.6,  4.0,  5.2,  2.8,  3.8)

     ##  The alternative hypothesis of interest is that the
     ##  Hunter L value is positively associated with the panel score.

     cor.test(x, y, method = "kendall", alternative = "greater")
     ## => p=0.05972

     cor.test(x, y, method = "kendall", alternative = "greater",
              exact = FALSE) # using large sample approximation
     ## => p=0.04765

     ## Compare this to
     cor.test(x, y, method = "spearm", alternative = "g")
     cor.test(x, y,                    alternative = "g")

--------------------------------------------------------------------------
Normal                 package:stats                 R Documentation

The Normal Distribution

Description:

     Density, distribution function, quantile function and random
     generation for the normal distribution with mean equal to 'mean'
     and standard deviation equal to 'sd'.

Usage:

     dnorm(x, mean=0, sd=1, log = FALSE)
     pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
     qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
     rnorm(n, mean=0, sd=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.

    mean: vector of means.

      sd: vector of standard deviations.

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 'mean' or 'sd' are not specified they assume the default values
     of '0' and '1', respectively.

     The normal distribution has density

       f(x) = 1/(sqrt(2 pi) sigma) e^-((x - mu)^2/(2 sigma^2))

     where mu is the mean of the distribution and sigma the standard
     deviation.

     'qnorm' is based on Wichura's algorithm AS 241 which provides
     precise results up to about 16 digits.

Value:

     'dnorm' gives the density, 'pnorm' gives the distribution
     function, 'qnorm' gives the quantile function, and 'rnorm'
     generates random deviates.


--------------------------------------------------------------------------
TDist                  package:stats                 R Documentation

The Student t Distribution

Description:

     Density, distribution function, quantile function and random
     generation for the t distribution with 'df' degrees of freedom
     (and optional noncentrality parameter 'ncp').

Usage:

     dt(x, df, ncp = 0, log = FALSE)
     pt(q, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)
     qt(p, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)
     rt(n, df, ncp = 0)

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.

      df: degrees of freedom (> 0, maybe non-integer).  'df = Inf' is
          allowed.  For 'qt' only values of at least one are currently
          supported.

     ncp: non-centrality parameter delta; currently for 'pt()' and
          'dt()', only for 'abs(ncp) <= 37.62'.

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:

     The t distribution with 'df' = n degrees of freedom has density

 f(x) = Gamma((n+1)/2) / (sqrt(n pi) Gamma(n/2)) (1 + x^2/n)^-((n+1)/2)

     for all real x. It has mean 0 (for n > 1) and variance n/(n-2)
     (for n > 2).

     The general _non-central_ t with parameters (df,Del) '= (df, ncp)'
     is defined as the distribution of T(df, Del) := (U + Del) /
     (Chi(df) / sqrt(df))  where U and Chi(df)  are independent random
     variables, U ~ N(0,1), and Chi(df)^2 is chi-squared, see
     Chisquare.

     The most used applications are power calculations for t-tests:
      Let T= (mX - m0) / (S/sqrt(n)) where mX is the 'mean' and S the
     sample standard deviation ('sd') of X_1,X_2,...,X_n which are
     i.i.d. N(mu,sigma^2). Then T is distributed as non-centrally t
     with 'df'= n-1 degrees of freedom and *n*on-*c*entrality
     *p*arameter 'ncp'= (mu - m0) * sqrt(n)/sigma.

Value:

     'dt' gives the density, 'pt' gives the distribution function, 'qt'
     gives the quantile function, and 'rt' generates random deviates.

     Invalid arguments will result in return value 'NaN', with a
     warning.
```