Wald Tests with R, Using the maxLik package

```
> # install.packages("maxLik") # Only need to do this once
> library("maxLik") # Do this every time
Loading required package: miscTools

Please cite the 'maxLik' package as:
Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood
estimation in R. Computational Statistics 26(3), 443-458. DOI 10.1007/s00180-010-
0217-1.

If you have questions, suggestions, or comments regarding the 'maxLik' package,
please use a forum or 'tracker' at maxLik's R-Forge site:
https://r-forge.r-project.org/projects/maxlik/
> rm(list=ls())
> # Re-generate Gamma data
> set.seed(3201); alpha=2; beta=3
> D = round(rgamma(50,shape=alpha, scale=beta),2)
> momalpha <- mean(D)^2/var(D); momalpha
[1] 1.899754
> mombeta <- var(D)/mean(D); mombeta
[1] 3.620574
> mom = c(momalpha,mombeta)
> names(mom) = c("alpha","beta") # maxLik will use these labels
>
> # Log likelihood, NOT the minus log likelihood
> gammall = function(theta,datta)
+       { value = sum(dgamma(datta,shape=theta[1],scale=theta[2],log=T))
+         return(value)
+       } # End of gammall
>
> gam = maxLik(logLik=gammall,start=mom,datta=D)
> summary(gam)
--------------------------------------------
Maximum Likelihood estimation
Newton-Raphson maximisation, 3 iterations
Return code 2: successive function values within tolerance limit
Log-Likelihood: -142.0316
2  free parameters
Estimates:
      Estimate Std. error t value  Pr(> t)
alpha   1.8059     0.3317   5.445 5.19e-08 ***
beta    3.8087     0.8032   4.742 2.12e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
--------------------------------------------
>
> gam$estimate
   alpha     beta
1.805930 3.808676
> gam$maximum
[1] -142.0316
> gam$gradient
        alpha          beta
-6.821210e-07 -1.648459e-06
> gam$hessian
          alpha       beta
alpha -36.66401 -13.130830
beta  -13.13083  -6.252776
>
> Vhat = -solve(gam$hessian) ; Vhat
           alpha       beta
alpha  0.1100199 -0.2310418
beta  -0.2310418  0.6451168
> # Inverse of Hessian of the MINUS log likelihood is the approximate
> # asymptotic covariance matrix. Have to make it negative because
> # maxLik returns the Hessian of the log likelihood.
>
```

```
> summary(gam)
--------------------------------------------
Maximum Likelihood estimation
Newton-Raphson maximisation, 3 iterations
Return code 2: successive function values within tolerance limit
Log-Likelihood: -142.0316
2  free parameters
Estimates:
      Estimate Std. error t value  Pr(> t)
alpha   1.8059     0.3317   5.445 5.19e-08 ***
beta    3.8087     0.8032   4.742 2.12e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
--------------------------------------------

> sqrt(diag(Vhat)) # Standard errors for comparison
    alpha      beta
0.3316925 0.8031917
>
> # Wald test of H0: alpha = beta
> # LR test gave G2 = 4.2776, p = 0.039
>
> # source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/Wtest.txt")
>
> Wtest

function(L,Tn,Vn,h=0) # H0: L theta = h
# Note Vn is the estimated asymptotic covariance matrix of Tn,
# so it's Sigma-hat divided by n. For Wald tests based on numerical
# MLEs, Tn = theta-hat, and Vn is the inverse of the Hessian.
     {
     value = numeric(3)
     names(value) = c("W","df","p-value")
     r = dim(L)[1]
     W = t(L%*%Tn-h) %*% solve(L%*%Vn%*%t(L)) %*%
          (L%*%Tn-h)
     W = as.numeric(W)
     pval = 1-pchisq(W,r)
     value[1] = W; value[2] = r; value[3] = pval
     return(value)
     }
>
> thetahat = gam$estimate
> LL = rbind(c(1,-1)); LL
     [,1] [,2]
[1,]    1   -1

> Wtest(LL,thetahat,Vhat)
         W         df    p-value
3.29520600 1.00000000 0.06948239
```

Testing Differences Between Independent Groups

```
> rm(list=ls())
> # help(ChickWeight)
> ChickWeight[1:15,] # Rows 1:15, all the columns
   weight Time Chick Diet
1      42    0     1    1
2      51    2     1    1
3      59    4     1    1
4      64    6     1    1
5      76    8     1    1
6      93   10     1    1
7     106   12     1    1
8     125   14     1    1
9     149   16     1    1
10    171   18     1    1
11    199   20     1    1
12    205   21     1    1
13     40    0     2    1
14     49    2     2    1
15     58    4     2    1
> dim(ChickWeight)
[1] 578   4
> id = 1:dim(ChickWeight)[1]
> # Keep the weights at 3 weeks
> keep = id[ChickWeight$Time==21]
> chickdata = ChickWeight[keep,c(1,3,4)]; head(chickdata)
   weight Chick Diet
12    205     1    1
24    215     2    1
36    202     3    1
48    157     4    1
60    223     5    1
72    157     6    1
> library(tables) # Loads the package -- must do this every time.

> tabular(Diet ~ weight*(mean+sd+length),data=chickdata)

      weight
 Diet mean   sd    length
 1    177.8  58.70 16
 2    214.7  78.14 10
 3    270.3  71.62 10
 4    238.6  43.35  9
> table(chickdata$Diet) # To check

 1  2  3  4
16 10 10  9
> msd = tabular(Diet ~ weight*(mean+sd+length),data=chickdata); is.matrix(msd)
[1] TRUE

> xbar = as.numeric(msd[,1]) # Column 1
> sdx =  as.numeric(msd[,2]) # Column 2
> n   =  as.numeric(msd[,3]) # Column 3
> Vhat = diag(sdx^2/n)
> # Test H0: mu1 = mu2 = mu3 = m4 with unequal variances
> LL = rbind(c(1,-1, 0, 0),
+            c(0, 1,-1, 0),
+            c(0, 0, 1,-1))

  > source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/Wtest.txt")

> Wtest(LL,xbar,Vhat)
           W           df      p-value
14.890945658  3.000000000  0.001912255
```

```
> # Compare p = 0.001912255  to traditional test
> attach(chickdata)
> Diet = factor(Diet)
> contrasts(Diet)
  2 3 4
1 0 0 0
2 1 0 0
3 0 1 0
4 0 0 1
> summary(lm(weight ~ Diet))

Call:
lm(formula = weight ~ Diet)

Residuals:
     Min       1Q   Median       3Q      Max
-140.700  -39.700   -1.556   37.250  127.250

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   177.75      16.00  11.113 6.07e-14 ***
Diet2          36.95      25.79   1.433  0.15955
Diet3          92.55      25.79   3.588  0.00088 ***
Diet4          60.81      26.66   2.281  0.02782 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 63.98 on 41 degrees of freedom
Multiple R-squared:  0.2541,  Adjusted R-squared:  0.1995
F-statistic: 4.655 on 3 and 41 DF,  p-value: 0.006858

> # Classical non-parametric test on ranks
> kruskal.test(weight ~ Diet)

      Kruskal-Wallis rank sum test

data:  weight by Diet
Kruskal-Wallis chi-squared = 10.585, df = 3, p-value = 0.0142
```
