Statistical Computing 1, Stat 590 Homework 10 Name: Answer Key
Fall 2015 Prof. Erik B. Erhardt

Part I. (105 points) Do all calculations in LaTeX + R + knitr. Insert computer text output and graphics to support what you are saying. For this assignment, all R code should well commented and be visible (`echo=TRUE`) in the document where you have written it.

(105pts) **1. Hardy-Weinberg Equilibrium**

If gene frequencies are in equilibrium, the genotypes AA, Aa, and aa occur in a population with frequencies $(1-\theta)^2$, $2\theta(1-\theta)$, and θ^2, according to the Hardy-Weinberg Law. In a sample from the Chinese population of Hong Kong in 1937, blood types occurred with the following frequencies, where M and N are erythrocyte antigens:

	Blood type			
	M	MN	N	Total
Frequency	342	500	187	1029
Category label	1	2	3	

If we let $\underset{\sim}{Y} = [Y_1, Y_2, Y_3]^\top$ be the counts in n categories 1, 2, and 3 based on a sample size m then the joint distribution of the cell counts is multinomial

$$\Pr[Y_1 = y_1, Y_2 = y_2, Y_3 = y_3|\theta] \;=\; \frac{m!}{\prod_{i=1}^n y_i!} \prod_{i=1}^n p_i(\theta)^{y_i}$$

where

$$p_1(\theta) \;=\; (1-\theta)^2, \quad p_2(\theta) \;=\; 2\theta(1-\theta), \quad \text{and} \quad p_3(\theta) \;=\; \theta^2$$

are cell probabilities and $0 < \theta < 1$.

(a) (10 pts) **Derive the Muiltinomial log-likelihood, $\ell(\theta)$.** Show that the log-likelihood for an arbitrary sample is

$$\ell(\theta) \;=\; \log(m!) - \sum_{i=1}^n \log(y_i!) + (2y_1 + y_2)\log(1-\theta) + (2y_3 + y_2)\log(\theta) + (y_2)\log(2).$$

Solution:

$$\begin{aligned}
\mathrm{L}(\theta) &= \Pr(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) \\
&= \frac{m!}{\prod_{i=1}^3 y_i!} p_1(\theta)^{y_1} p_2(\theta)^{y_2} p_3(\theta)^{y_3} \\
&= \frac{m!}{\prod_{i=1}^3 y_i!} (1-\theta)^{2y_1} (2\theta(1-\theta))^{y_2} \theta^{2y_3} \\
&= \frac{m!}{\prod_{i=1}^3 y_i!} (1-\theta)^{2y_1+y_2} \theta^{y_2+2y_3} 2^{y_2}
\end{aligned}$$

Taking the log of the last expression gives the desired log-likelihood.

(b) (15 pts) **Derive the derivatives of the log-likelihood and the expected information.** Derive general expressions for the first and second derivatives of $\ell(\theta)$ ($\dot{\ell}(\theta)$ and $\ddot{\ell}(\theta)$) and the expected information $\mathbf{I}(\theta)$.
Solution:

$$\begin{aligned}\dot{\ell}(\theta) &= -\frac{2y_1+y_2}{1-\theta}+\frac{y_2+2y_3}{\theta}\\ \ddot{\ell}(\theta) &= -\frac{2y_1+y_2}{(1-\theta)^2}-\frac{y_2+2y_3}{\theta^2}\end{aligned}$$

$Y_i \sim \text{Bin}(m, p_i(\theta))$ so $\text{E}(Y_i) = mp_i(\theta)$, giving

$$\begin{aligned}\mathbf{I}(\theta) &= \text{E}(-\ddot{\ell}(\theta))\\ &= \text{E}\left(\frac{2y_1+y_2}{(1-\theta)^2}+\frac{y_2+2y_3}{\theta^2}\right)\\ &= m\left(\frac{2p_1(\theta)+p_2(\theta)}{(1-\theta)^2}+\frac{p_2(\theta)+2p_3(\theta)}{\theta^2}\right)\\ &= m\left(\frac{2(1-\theta)^2+2\theta(1-\theta)}{(1-\theta)^2}+\frac{2\theta(1-\theta)+2\theta^2}{\theta^2}\right)\\ &= 2m\left(\frac{(1-\theta)+\theta}{(1-\theta)}+\frac{(1-\theta)+\theta}{\theta}\right)\\ &= 2m\left(\frac{1}{(1-\theta)}+\frac{1}{\theta}\right)\\ &= 2m\left(\frac{\theta+(1-\theta)}{(1-\theta)\theta}\right)\\ &= 2m\left(\frac{1}{(1-\theta)\theta}\right)\end{aligned}$$

(c) (10 pts) **Likelihood equations, $\mathbf{L}(\theta|\underset{\sim}{y})$.** Write down the likelihood equation for an arbitrary sample (that is, $\dot{\ell}(\theta) = 0$) and solve for the MLE.
Solution:

$$\begin{aligned}\dot{\ell}(\theta) = 0 &= -\frac{2y_1+y_2}{1-\theta}+\frac{y_2+2y_3}{\theta}\\ &= -\frac{-2y_1\theta-y_2\theta+y_2(1-\theta)+2y_3(1-\theta)}{(1-\theta)\theta}\\ 0 &= (-2y_1-y_2-y_2-2y_3)\theta+y_2+2y_3\\ \hat{\theta} &= \frac{y_2+2y_3}{2y_1+2y_2+2y_3}\\ &= \frac{500+2(187)}{2(342)+2(500)+2(187)}\\ &= 0.424684159\end{aligned}$$

(d) (10 pts) **Functions to evaluate functions in parts (a) and (b).** Write functions to evaluate $\ell(\theta)$, $\dot{\ell}(\theta)$, $\ddot{\ell}(\theta)$, and $\mathbf{I}(\theta)$ for and arbitrary sample $\underset{\sim}{Y} = [Y_1, Y_2, Y_3]^\top$. Ignore the constant terms in $\ell(\theta)$.

Solution:

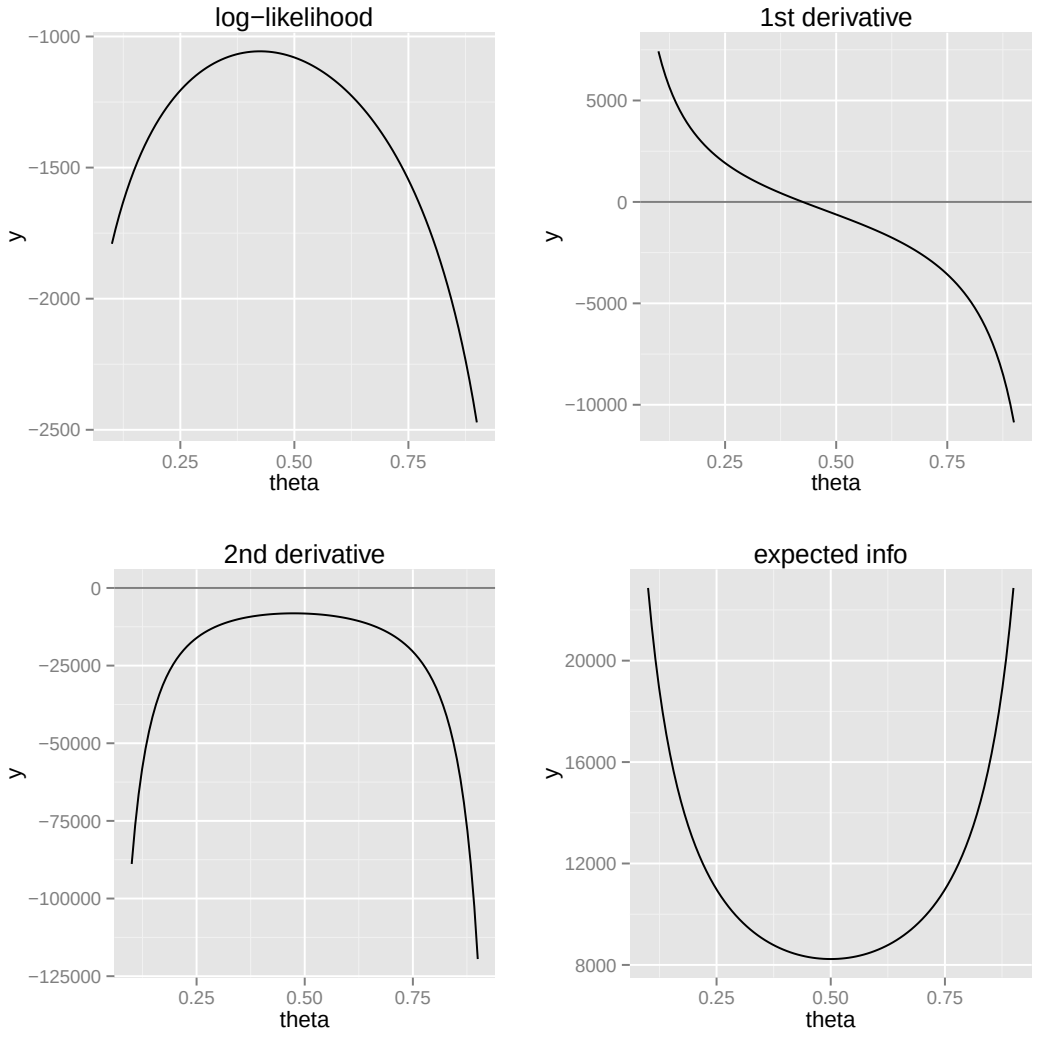
```
# maximizing multinomial likelihood
y <- c(342, 500, 187)

# functions: log-likelihood, 1st derivative, 2nd derivative, and expected info
f.l <- function(theta, y) {
  temp <- # lfactorial(sum(y)) - sum(lfactorial(y)) + # ignoring constant terms
          (2 * y[1] + y[2]) * log(1 - theta) +
          (y[2] + 2 * y[3]) * log(theta) +
          y[2] * log(2)
  return(temp)
}
f.dl <- function(theta, y) {
  temp <- - (2 * y[1] + y[2]) / (1 - theta) +
          (y[2] + 2 * y[3]) / theta
  return(temp)
}
f.ddl <- function(theta, y) {
  temp <- - (2 * y[1] + y[2]) / (1 - theta)^2 +
          - (y[2] + 2 * y[3]) / theta^2
  return(temp)
}
f.info <- function(theta, y) {
  temp <- 2 * sum(y) / ((1 - theta) * theta);
  return(temp)
}
```

(e) (10 pts) **Plot $\ell(\theta)$, guess at MLE.** For the observed data, plot $\ell(\theta)$. What is a good guess for the MLE based on the graph?

Solution: The MLE is close to $\theta = 0.4$.

```
# plot functions
library(ggplot2)
p1 <- ggplot(data.frame(theta = c(0.1, 0.9)), aes(theta))
p1 <- p1 + stat_function(fun = f.l, args = list(y))
p1 <- p1 + labs(title = "log-likelihood")
#print(p1)

p2 <- ggplot(data.frame(theta = c(0.1, 0.9)), aes(theta))
p2 <- p2 + geom_hline(yintercept = 0, alpha = 0.5)
p2 <- p2 + stat_function(fun = f.dl, args = list(y))
p2 <- p2 + labs(title = "1st derivative")
#print(p2)

p3 <- ggplot(data.frame(theta = c(0.1, 0.9)), aes(theta))
p3 <- p3 + geom_hline(yintercept = 0, alpha = 0.5)
p3 <- p3 + stat_function(fun = f.ddl, args = list(y))
```

```
p3 <- p3 + labs(title = "2nd derivative")
#print(p3)

p4 <- ggplot(data.frame(theta = c(0.1, 0.9)), aes(theta))
p4 <- p4 + stat_function(fun = f.info, args = list(y))
p4 <- p4 + labs(title = "expected info")
#print(p4)

library(gridExtra)
grid.arrange(p1, p2, p3, p4, ncol=2)
```



(f) (10 pts) **Newton-Raphson for MLE.** Write an NR procedure to numerically evaluate the MLE of θ for an arbitrary sample.
Solution:

```
# NR routine for finding root of g(x) = 0.
# Requires predefined g(x) and gp(x) = deriv of g(x)
# The iteration is controlled by:
```

```
#   eps   = absolute convergence criterion
#   maxit = maximum allowable number of iterations
# Input:  xnew = user prompted starting value
# Output: number of root, steps, and note
f.NR <- function(g, gp, xnew = 1, eps = 0.001, maxit = 35, y = c(1,1,1)) {
  xold <- -Inf   # needed so argument in while() loop is defined

  i <- 1;   # initial iteration index

  NR.hist <- data.frame(i, xnew, diff = abs(xnew - xold)) # iteration history
  while ((i <= maxit) & (abs(xnew - xold) > eps)) {
    i <- i + 1                          # increment iteration
    xold <- xnew                        # old guess is current guess
    xnew <- xold - g(xold, y) / gp(xold, y)  # new guess

    NR.hist <- rbind(NR.hist, c(i, xnew, abs(xnew - xold))) # iteration history
  }

  out <- list()
  out$root <- xnew
  out$iter <- i
  out$hist <- NR.hist
  if (abs(xnew - xold) <= eps) {
    out$note <- paste("Absolute convergence of", eps, "satisfied")
  }
  if (i > maxit) {
    out$note <- paste("Exceeded max iterations of ", maxit)
  }
  return(out)
}
```

(g) (10 pts) **Fisher scoring for MLE.** Write a Fisher's Scoring procedure to numerically evaluate the MLE of θ for an arbitrary sample.

Solution:

```
# Fisher's scoring routine for finding root of g(x) = 0.
# Requires predefined g(x) and gp(x) = deriv of g(x)
# The iteration is controlled by:
#   eps   = absolute convergence criterion
#   maxit = maximum allowable number of iterations
# Input:  xnew = user prompted starting value
# Output: number of root, steps, and note
f.FS <- function(g, gp, xnew = 1, eps = 0.001, maxit = 35, y = c(1,1,1)) {
  xold <- -Inf   # needed so argument in while() loop is defined

  i <- 1;   # initial iteration index

  NR.hist <- data.frame(i, xnew, diff = abs(xnew - xold)) # iteration history
  while ((i <= maxit) & (abs(xnew - xold) > eps)) {
    i <- i + 1                          # increment iteration
    xold <- xnew                        # old guess is current guess
```

```
    xnew <- xold + g(xold, y) / gp(xold, y)  # new guess

    NR.hist <- rbind(NR.hist, c(i, xnew, abs(xnew - xold))) # iteration history
  }

  out <- list()
  out$root <- xnew
  out$iter <- i
  out$hist <- NR.hist
  if (abs(xnew - xold) <= eps) {
    out$note <- paste("Absolute convergence of", eps, "satisfied")
  }
  if (i > maxit) {
    out$note <- paste("Exceeded max iterations of ", maxit)
  }
  return(out)
}
```

(h) (10 pts) **Compare NR and Fisher scoring methods.** For the given data, use each algorithm to evaluate the MLE of θ. Compute estimates of the standard deviation of the MLE $\hat{\theta}$ using both the observed and expected information. Also, compute two approximate 95% CIs for θ of the form $\theta \pm 1.96\text{SE}[\hat{\theta}]$, where the SE is based on the (1) observed and (2) expected information. Discuss the results: what are the similarities/dissimilarities between routines, SEs, CIs, etc., and does the numerically evaluated MLE agree with the answer in Part (c)?

Solution: Both methods converge quickly to the MLE, though Fisher's scoring method unconditionally converges in a single step.

```
outNR0.40 <- f.NR(f.dl, f.ddl, xnew = 0.40, y = y, eps = 1e-8)
outNR0.40
```

```
## $root
## [1] 0.4246842
##
## $iter
## [1] 5
##
## $hist
##   i      xnew         diff
## 1 1 0.4000000          Inf
## 2 2 0.4241866 2.418664e-02
## 3 3 0.4246840 4.973687e-04
## 4 4 0.4246842 1.536130e-07
## 5 5 0.4246842 1.465494e-14
##
## $note
## [1] "Absolute convergence of 1e-08 satisfied"
```

The MLE is estimated to be $\hat{\theta} = 0.4247$.

```
outFS0.40 <- f.FS(f.dl, f.info, xnew = 0.40, y = y, eps = 1e-8)
outFS0.40
## $root
## [1] 0.4246842
##
## $iter
## [1] 3
##
## $hist
##   i      xnew       diff
## 1 1 0.4000000        Inf
## 2 2 0.4246842 0.02468416
## 3 3 0.4246842 0.00000000
##
## $note
## [1] "Absolute convergence of 1e-08 satisfied"
```

The MLE is estimated to be $\hat{\theta} = 0.4247$.

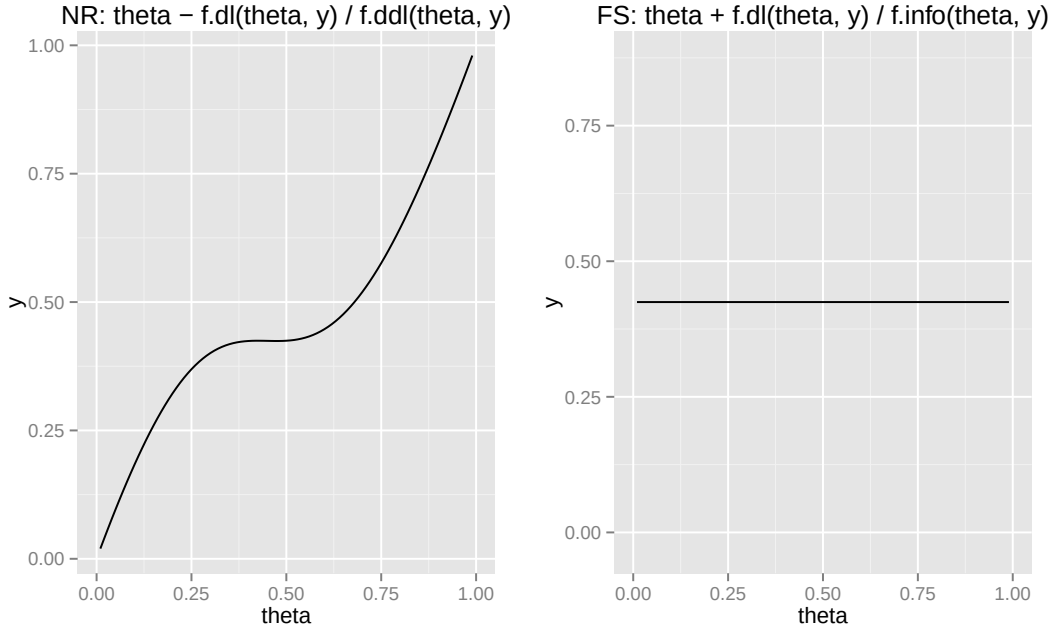
```
# estimated standard deviation via second derivative
SEobs <- sqrt(-1 / f.ddl(outFS0.40$root, y))
# estimated standard deviation via Fisher's information
SEexp <- sqrt(1 / f.info(outFS0.40$root, y))

# 95% CI margins-of-error
MOEobs <- qnorm(1-0.05/2) * SEobs
MOEexp <- qnorm(1-0.05/2) * SEexp

# 95% CIs
outFS0.40$root
## [1] 0.4246842
c(outFS0.40$root - MOEobs, outFS0.40$root + MOEobs)
## [1] 0.4033286 0.4460398
c(outFS0.40$root - MOEexp, outFS0.40$root + MOEexp)
## [1] 0.4033286 0.4460398
```

(i) (10 pts) **Sensitivity analysis.** *More open-ended:* By choosing different starting values, examine the sensitivity of NR and Fisher Scoring to the choice of the initial guess of the MLE. Do any graphical summaries help to understand the sensitivity (or lack of sensitivity)? Discuss.

Solution: The plots below indicate the next guess for the root when the current guess is θ. For NR this is $\theta - \dot{\ell}(\theta)/\ddot{\ell}(\theta)$ and for Fisher Scoring, this is $\theta + \dot{\ell}(\theta)/\mathbf{I}(\theta)$. The plot indicates that both Newton-Raphson and Fisher's scoring methods will converge to the correct solution for all starting values. It indicates that Fisher's Scoring converges in a single step regardless of the initial value of θ.

```
# plot functions
library(ggplot2)
p1 <- ggplot(data.frame(theta = c(0, 1)), aes(theta))
```

```
p1 <- p1 + stat_function(fun = function(theta, y)
                           {theta - f.dl(theta, y) / f.ddl(theta, y)}
                       , args = list(y))
p1 <- p1 + labs(title = "NR: theta - f.dl(theta, y) / f.ddl(theta, y)")
#print(p1)

p2 <- ggplot(data.frame(theta = c(0, 1)), aes(theta))
p2 <- p2 + stat_function(fun = function(theta, y)
                           {theta + f.dl(theta, y) / f.info(theta, y)}
                       , args = list(y))
p2 <- p2 + labs(title = "FS: theta + f.dl(theta, y) / f.info(theta, y)")
#print(p2)

library(gridExtra)
grid.arrange(p1, p2, nrow=1)
## Warning:  Removed 2 rows containing missing values (geom_path).
## Warning:  Removed 2 rows containing missing values (geom_path).
```



(j) (10 pts) **Summary.** Summarize your results.

Solution: It is remarkably easy to maximize a univariate likelihood function, and to know when particular maximization methods will converge. This was a great example.