HW04 Report

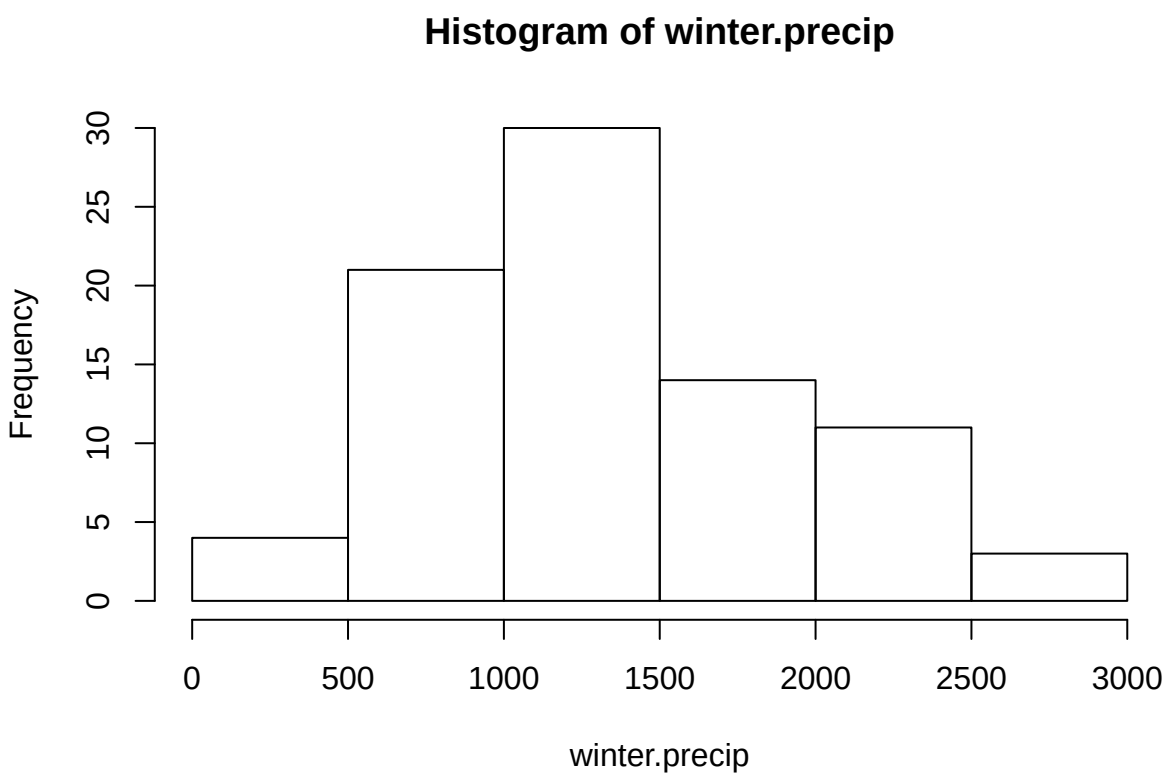
Ninh (Vincent) DO

February 27, 2019

Problem 2c

```
# Read data file and process data
precip = read.csv("berkeleyprecip.csv", header = TRUE)
precip[precip == -99999] = NA

# Calculate the total winter precipitation for each year (removing missing values)
winter.precip = precip$DEC + precip$JAN + precip$FEB
winter.precip = winter.precip[!is.na(winter.precip)]
hist(winter.precip)
```




```
# Write a function for a negative gamma log-likelihood
nll = function(par, x, verbose = FALSE) {
  alpha = par[1]
  beta = par[2]
  ll = sum(dgamma(x, alpha, beta, log = TRUE))
  if (verbose) {
    print(c(par, -ll))
  }
  return(-ll)
}
```

```r
# Numerical minimize the negative gamma log-likelihood
start <- c(alpha = 1, beta = 1)    # starting values
eps <- 1e-10    # small value for lower bounds

op <- optim(par = start, fn = nll, lower = rep(eps, 2), x = winter.precip, verbose = FALSE)
```

```
## Warning in optim(par = start, fn = nll, lower = rep(eps, 2), x =
## winter.precip, : bounds can only be used with method L-BFGS-B (or Brent)
```

```r
# Print the result, which is a list, extract elements using $
op
```

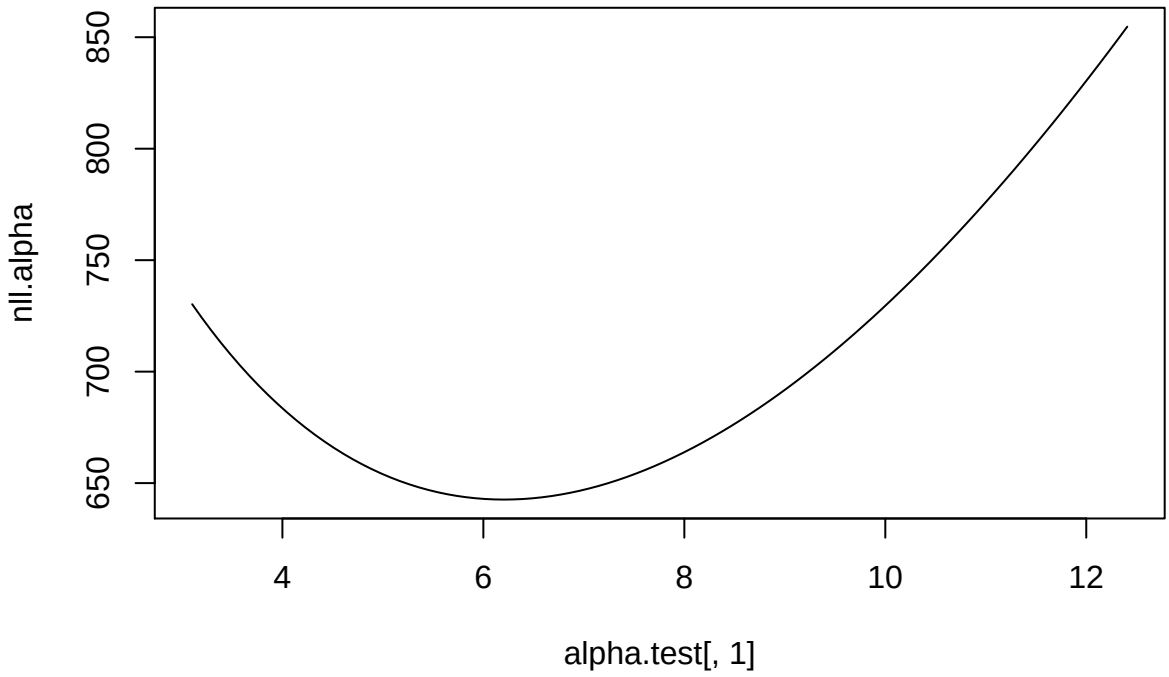
```
## $par
##       alpha        beta
## 6.204123823 0.004650913
##
## $value
## [1] 642.6143
##
## $counts
## function gradient
##       39       39
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```r
mle <- op$par
mle
```

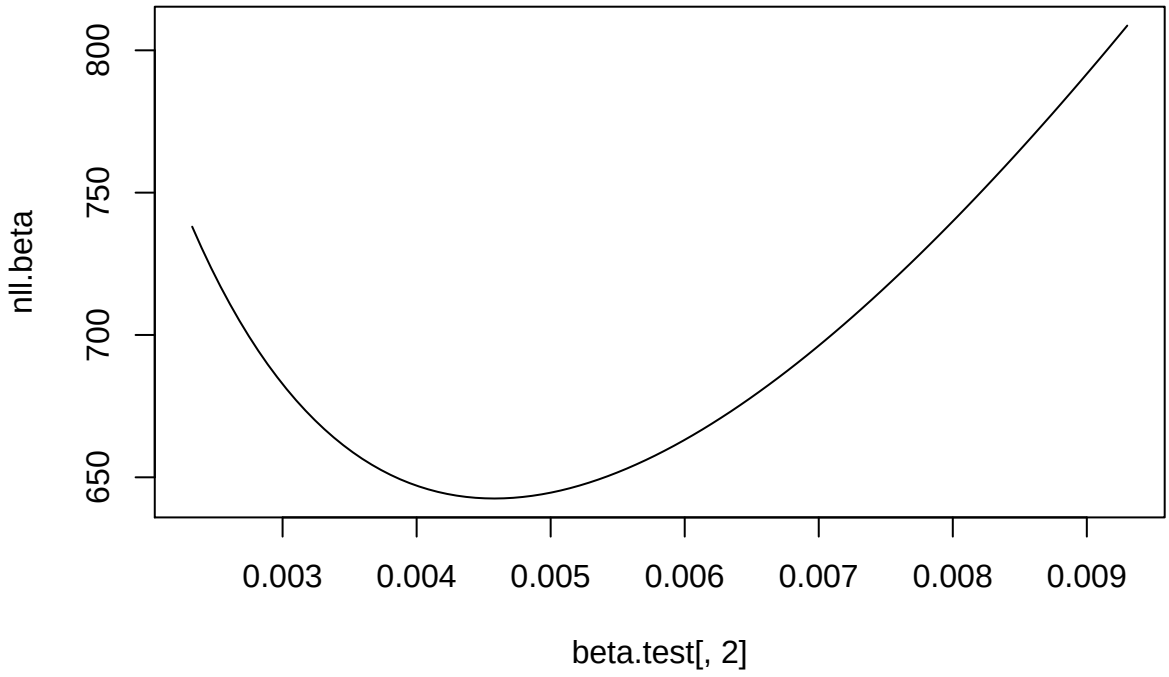
```
##       alpha        beta
## 6.204123823 0.004650913
```

```r
# Examine the nll at the min

alpha.test <- cbind(seq(mle[1]/2, mle[1]*2, length = 100), mle[2])
nll.alpha <- apply(alpha.test, 1, nll, x = winter.precip, verbose = FALSE)
plot(alpha.test[,1], nll.alpha, type = "l")
```




```
beta.test <- cbind(mle[1], seq(mle[2]/2, mle[2]*2, length = 100))
nll.beta <- apply(beta.test, 1, nll, x = winter.precip, verbose = FALSE)
plot(beta.test[,2], nll.beta, type = "l")
```




```
# Estimate the Hessian (Fisher information matrix) and construct 95% normal-based confidence interval

op <- optim(par = start, fn = nll,
            lower = rep(eps, 2), hessian = TRUE,
            x = winter.precip, verbose = FALSE)
```

```
## Warning in optim(par = start, fn = nll, lower = rep(eps, 2), hessian =
```

```
## TRUE, : bounds can only be used with method L-BFGS-B (or Brent)
```

```
mle <- op$par
J <- solve(op$hessian)    # no negative - already working with negative ll
se.hat <- sqrt(diag(J))

# Lower values
lower <- mle - 2*se.hat
lower
```

```
##       alpha        beta
## 4.799790126 0.003608113
```

```
# Uppter values
upper <- mle + 2*se.hat
upper
```

```
##       alpha        beta
## 7.608457520 0.005693713
```

Comment:

Both plots alpha and beta are convex, it indicates that the algorithm found a global optimum.