1. The routine is coded in Julia.

   The obtained log likelihood is -6942.805.

   The transpose of the score is

$$\begin{bmatrix} -2605.9082518892865 \\ -556.3196848948379 \\ -1156.8594262530135 \\ -222.81767101773977 \\ -933.039979318137 \\ -1215.1317422401712 \\ -2109.626213790837 \\ -948.0740374410863 \\ -5049.875617650256 \\ -4534.790470404961 \\ -19401.89853086738 \\ -19164.659456830384 \\ -918.8553971099844 \\ -351.75306280921296 \\ -466.6888493111424 \\ -582.4690752990825 \\ -546.4113143620349 \end{bmatrix} \tag{1}$$

   The Hessian matrix is too large to display, but is computed in the routine.

2. We obtain similar results using the two approaches. The Euclidean norm of the difference between scores obtained by using the two approaches was around 1.40 while the norm between the two Hessians was 0.35.

3. The results obtained from implementing the Newton algorithm is displayed in part 4. The associated code is attached in the appendix. The *NewtonAlg* function within "functions.jl" file details the algorithm.

4. The computation speed between the three methods is compared below

Newton Method	838.489 ms
Quasi-Newton (BFGS)	4.703 s
Quasi-Newton (Simplex)	2.935 s

   The estimates of β under each of these methods is tabulated below. We can observe that the Newton Method yields estimates closest to the true values, and does so fastest among the three methods.

Newton	BFGS	Simplex
$\begin{bmatrix} -1.000 \\ 1.530e-7 \\ 8.4296e-8 \\ 1.086e-7 \\ 3.010e-8 \\ 1.569e-7 \\ -1.938e-8 \\ 4.086e-8 \\ 1.278e-7 \\ 5.152e-8 \\ -5.411e-8 \\ 3.651e-8 \\ 1.4033e-7 \\ 1.988e-7 \\ 1.170e-7 \\ 4.438e-8 \\ 2.374e-8 \end{bmatrix}$	$\begin{bmatrix} -6.056 \\ 0.867 \\ 0.527 \\ 0.595 \\ 0.163 \\ 0.871 \\ -0.052 \\ 0.215 \\ 1.007 \\ 0.335 \\ -0.284 \\ 0.189 \\ 0.758 \\ 1.152 \\ 0.770 \\ 0.379 \\ 0.2406 \end{bmatrix}$	$\begin{bmatrix} -1.953 \\ 0.686 \\ 0.318 \\ 0.411 \\ 0.008 \\ -0.591 \\ -0.074 \\ -0.379 \\ 0.279 \\ 0.523 \\ -0.481 \\ 0.233 \\ 0.680 \\ 0.344 \\ 0.064 \\ -0.407 \\ -0.486 \end{bmatrix}$