Homework 2

github link:https://github.com/xxou617/bios731_hw2_ou

Context

This assignment reinforces ideas in Module 2: Optimization. We focus specifically on implementing the Newton's method, EM, and MM algorithms.

Due date and submission

Please submit (via Canvas) a PDF containing a link to the web address of the GitHub repo containing your work for this assignment; git commits after the due date will cause the assignment to be considered late. Due date is Wednesday, 2/19 at 10:00AM.

Points

Problem	Points
Problem 0	15
Problem 1	30
Problem 2	5
Problem 3	30
Problem 4	20

Problem 0

This "problem" focuses on structure of your submission, especially the use git and GitHub for reproducibility, R Projects to organize your work, R Markdown to write reproducible reports, relative paths to load data from local files, and reasonable naming structures for your files.

To that end:

- create a public GitHub repo + local R Project; I suggest naming this repo / directory bios731_hw2_YourLastName (e.g. bios731_hw2_wrobel for Julia)
- Submit your whole project folder to GitHub
- Submit a PDF knitted from Rmd to Canvas. Your solutions to the problems here should be implemented in your .Rmd file, and your git commit history should reflect the process you used to solve these Problems.

Algorithms for logistic regression

For a given subject in a study, we are interested in modeling $\pi_i = P(Y_i = 1|X_i = x_i)$, where $Y_i \in \{0, 1\}$. The logistic regression model takes the form

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \log\left(\frac{P(Y_i = 1|X_i)}{1 - P(Y_i = 1|X_i)}\right) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_p X_{pi}$$

- $Y_1, Y_2, \ldots, Y_n \sim Bernoulli(\pi)$
- PDF is $f(y_i; \pi) = \pi^{y_i}(1-\pi)^{1-y_i}$

Problem 1: Newton's method

- Derive likelihood, gradient, and Hessian for logistic regression for an arbitrary number of predictors p
- What is the Newton's method update for β for logistic regression?
- Is logistic regression a convex optimization problem? Why or why not?

$$\pi_i = expit(X_i^T\beta) = \frac{\exp(X_i^T\beta)}{1+\exp(X_i^T\beta)} = \frac{1}{1+\exp(-X_i^T\beta)}, \quad X_i = [X_{1i}, X_{2i}, \ldots, X_{pi}]^T$$

likelihood:

$$\begin{aligned} L(\beta) &= \prod_i^n f(y_i; \pi) = \prod_i^n \pi_i^{y_i}(1-\pi_i)^{1-y_i} = \prod_i^n (\frac{\pi_i}{1-\pi_i})^{y_i}(1-\pi_i) \\ &= \prod_i^n \exp(X_i^T\beta y_i)(1+\exp(X_i^T\beta))^{-1} \end{aligned}$$

Thus the **log-likelihood** is:

$$l(\beta) = \log(L(\beta)) = \sum_i^n \Big(y_i \log(\pi_i) + y_i \log(1-\pi_i)\Big) = \sum_i^n (X_i^T\beta y_i - \log(1+\exp(X_i^T\beta))$$

gradient:

$$\nabla l(\beta) = \sum_{i=1}^n \Big(y_i - \big(1+\exp(-X_i^T\beta)\big)^{-1}\Big)X_i = \sum_{i=1}^n (y_i - \pi_i)X_i$$

Hessian:

$$H = \nabla^2 l(\beta) = -\sum_{i=1}^n (1+\exp(-X_i^T\beta))^{-2}\exp(-X_i^T\beta)X_i = -\sum_{i=1}^n \pi_i(1-\pi_i)X_i X_i^T$$

Newton's method update

$$\beta_{t+1} = \beta_t - H^{-1}\nabla l(\beta)$$

a convex optimization problem Yes, logistic regression is a convex optimization problem because the log-likelihood function is concave in β and the Hessian matrix is negative semi-definite, ensuring a unique global maximum.

Problem 2: MM

(A) In constructing a minorizing function, first prove the inequality

$$-\log\{1+\exp x_i^T\theta\} \geq -\log\{1+\exp(X_i^T\theta^{(k)})\} - \frac{\exp(X_i^T\theta)-\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}$$

with equality when $\theta=\theta^{(k)}$. This eliminates the log terms.

if $f(s)$ is convex and differentiable,

$$f(s) \geq f(t) + \nabla f(t)^T(s-t)$$

and equality holds when $s=t$.

Here let $f(s)=-\log(s)$, which is convex and differentiable, $s=1+\exp(X_i^T\theta)$ and $t=1+\exp(X_i^T\theta^{(k)})$, we can get

$$-\log\{1+\exp(x_i^T\theta)\} \geq -\log\{1+\exp(X_i^T\theta^{(k)})\} - \frac{\exp(X_i^T\theta)-\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}$$

with equality when $\theta=\theta^{(k)}$

(B) Now apply the arithmetic-geometric mean inequality to the exponential function $\exp(X_i^T\theta)$ to separate the parameters. Assuming that θ has p components and that there are n observations, show that these maneuvers lead to a minorizing function

$$g(\theta|\theta^{(k)}) = -\frac{1}{p}\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}\sum_{j=1}^{p}\exp\{pX_{ij}(\theta_j-\theta_j^{(k)})\} + \sum_{i=1}^{n}Y_iX_i^T\theta = 0$$

up to a constant that does not depend on θ.

According to problem 1, we have the log-likelihood of logistic regression:

$$l(\beta) = \sum_{i}^{n}(Y_iX_i^T\theta - \log(1+\exp(X_i^T\theta))$$

for $\exp(X_i^T\theta)$:

$$\exp(X_i^T\theta) = \exp\Big(\sum_{j=1}^{p}X_{ij}\theta_j\Big) = \exp\Big(\sum_{j=1}^{p}X_{ij}\theta_j^{(k)}\Big)\exp\Big(\sum_{j=1}^{p}X_{ij}(\theta_j-\theta_j^{(k)})\Big)$$

By **Arithmetic-geometric mean inequality**:

$$\exp\Big(\sum_{j=1}^{p}X_{ij}(\theta_j-\theta_j^{(k)})\Big) \leq \frac{1}{p}\sum_{j=1}^{p}\exp\Big(pX_{ij}(\theta_j-\theta_j^{(k)})\Big)$$

$$\begin{aligned}
-\frac{\exp(X_i^T\theta)-\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})} &= -\frac{\exp(X_i^T\theta)}{1+\exp(X_i^T\theta^{(k)})} + \frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})} \\
&\geq -\frac{\exp\Big(\sum_{j=1}^{p}X_{ij}\theta_j^{(k)}\Big)}{1+\exp(X_i^T\theta^{(k)})}\frac{1}{p}\sum_{j=1}^{p}\exp\Big(pX_{ij}(\theta_j-\theta_j^{(k)})\Big) + \frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}
\end{aligned}$$

Sum over all observations,

$$\begin{aligned}
g(\theta|\theta^{(k)}) &= -\frac{1}{p}\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}\sum_{j=1}^{p}\exp\{pX_{ij}(\theta_j-\theta_j^{(k)})\}+\sum_{i=1}^{n}Y_iX_i^T\theta \\
&\leq -\sum_{i=1}^{n}\frac{\exp(X_i^T\theta)-\sum_{i=1}^{n}\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}-\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}+\sum_{i=1}^{n}Y_iX_i^T\theta \\
&\leq \sum_{i=1}^{n}\log\{1+\exp(X_i^T\theta^{(k)})\}-\sum_{i=1}^{n}\log\{1+\exp(x_i^T\theta)\}+\sum_{i=1}^{n}Y_iX_i^T\theta-\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})} \\
&= l(\theta)+\sum_{i=1}^{n}\log\{1+\exp(X_i^T\theta^{(k)})\}-\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}
\end{aligned}$$

Here, $\sum_{i=1}^{n}\log\{1+\exp(X_i^T\theta^{(k)})\}-\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}$ does not depend on θ.

(C) Finally, prove that maximizing $g(\theta|\theta^{(k)})$ consists of solving the equation

$$-\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})X_{ij}\exp(-pX_{ij}\theta_j^{(k)})}{1+\exp(X_i^T\theta^{(k)})}\exp(pX_{ij}\theta_j)+\sum_{i=1}^{n}Y_iX_{ij}=0$$

for each j.

To maximize $g(\theta|\theta^{(k)})$,

$$\begin{aligned}
\frac{\partial g(\theta|\theta^{(k)})}{\partial\theta_j} &= -\frac{1}{p}\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}\sum_{j=1}^{p}\frac{\partial}{\partial\theta_j}\exp\{pX_{ij}(\theta_j-\theta_j^{(k)})\}+\frac{\partial}{\partial\theta_j}\sum_{i=1}^{n}Y_iX_i^T\theta \\
&= -\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})X_{ij}\exp(-pX_{ij}\theta_j^{(k)})}{1+\exp(X_i^T\theta^{(k)})}\exp(pX_{ij}\theta_j)+\sum_{i=1}^{n}Y_iX_{ij} \\
&= 0
\end{aligned}$$

Thus, we need to solve the $\frac{\partial g(\theta|\theta^{(k)})}{\partial\theta_j}=0$ to maximize $g(\theta|\theta^{(k)})$

Problem 3: simulation

Next we will implement logistic regression in R four different ways and compare the results using a short simulation study.

- implement using Newton's method from 1.1 in R
- implement using MM from 1.2 in R
- implement using `glm()` in R
- implement using `optim()` in R:
  - Use the option `method = "BFGS"`, which implements a Quasi-Newton approach

Simulation study specification:

- simulate from the model $logit(P(Y_i=1|X_i))=\beta_0+\beta_1X_i$

beta	estimate	CI.lower	CI.up	iter.number	methods	times	true
est_beta0	0.9180559	0.1418926	1.694219	4	newton	0.030	1
est_beta0	0.9181153	0.1419471	1.694284	72	MM	0.167	1
est_beta0	0.9180559	0.1418929	1.694219	4	GLM	0.005	1
est_beta0	0.9178798	0.1417302	1.694029	13	QuasiNewton	0.022	1

beta	estimate	CI.lower	CI.up	iter.number	methods	times	true
est_beta1	0.1069350	-0.6810565	0.8949266	4	newton	0.030	0.3
est_beta1	0.1069368	-0.6810599	0.8949335	72	MM	0.167	0.3
est_beta1	0.1069350	-0.6810562	0.8949263	4	GLM	0.005	0.3
est_beta1	0.1070429	-0.6809374	0.8950231	13	QuasiNewton	0.022	0.3

  - $\beta_0 = 1$
  - $\beta_1 = 0.3$
  - $X_i \sim N(0, 1)$
  - $n = 200$
  - $nsim = 1$
- For your implementation of MM and Newton's method, select your own starting value and stopping criterion, but make sure they are the same for the two algorithms

Set starting values at 0.5 for β_0 and β_1 and use the difference of updated beta as stopping criterion.

You only need to run the simulation using **one simulated dataset**. For each of the four methods, report:

- $\hat{\beta}_0$, $\hat{\beta}_1$
- 95% confidence intervals for $\hat{\beta}_0$, $\hat{\beta}_1$
- computation time
- number of iterations to convergence

Make 2-3 plots or tables comparing your results, and summarize these findings in one paragraph.

```
library(ggplot2)
library(gt)

load(here::here("data", "simu_out.Rdata"))
```

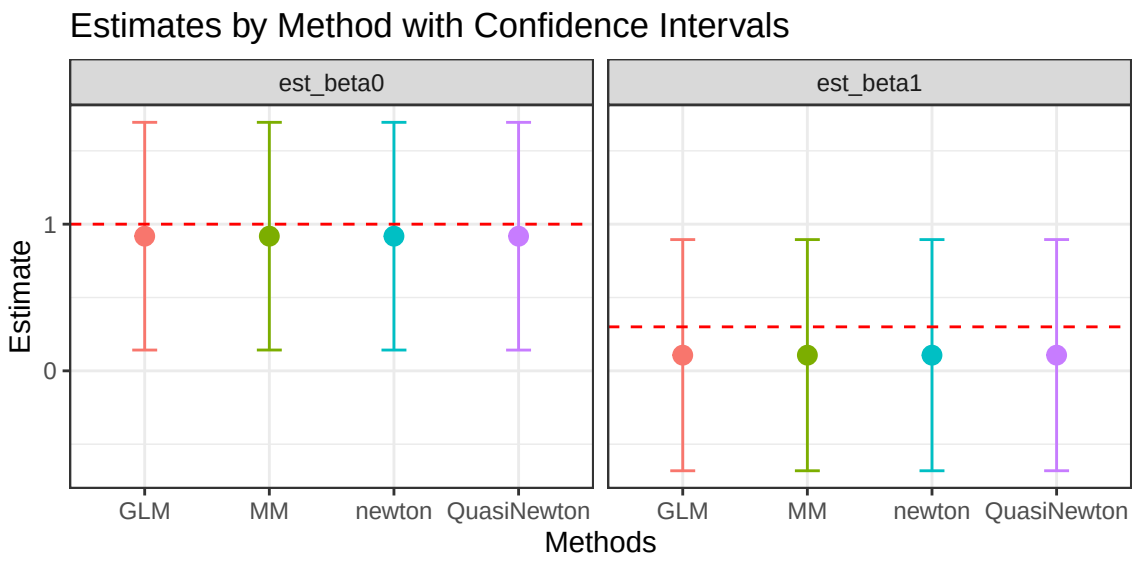
```
beta0 <- simu_out |> filter(beta == "est_beta0"); gt(beta0)
```

```
beta1 <- simu_out |> filter(beta == "est_beta1"); gt(beta1)
```

```
ggplot(simu_out, aes(x = methods, y = estimate, color = methods)) +
  geom_point(size = 3) +  # Points for estimates
  geom_errorbar(aes(ymin = CI.lower, ymax = CI.up), width = 0.2) +  # Error bars
  facet_wrap( ~ beta) +
  geom_hline(aes(yintercept = ifelse(beta == "est_beta0", 1.0, 0.3)),
             color = "red", linetype = "dashed") +  # Reference line
```

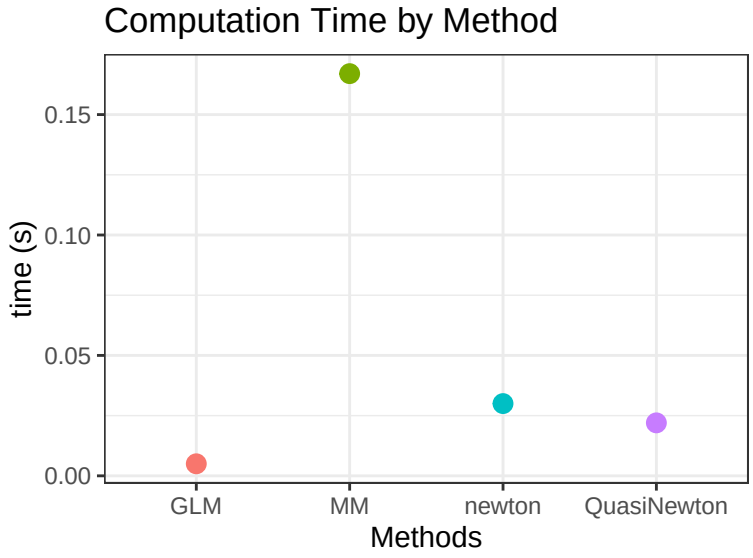
```
  labs(x = "Methods", y = "Estimate", title = "Estimates by Method with Confidence Intervals") +
  theme_bw() +
  theme(legend.position = "none") -> plot_CI

plot_CI
```

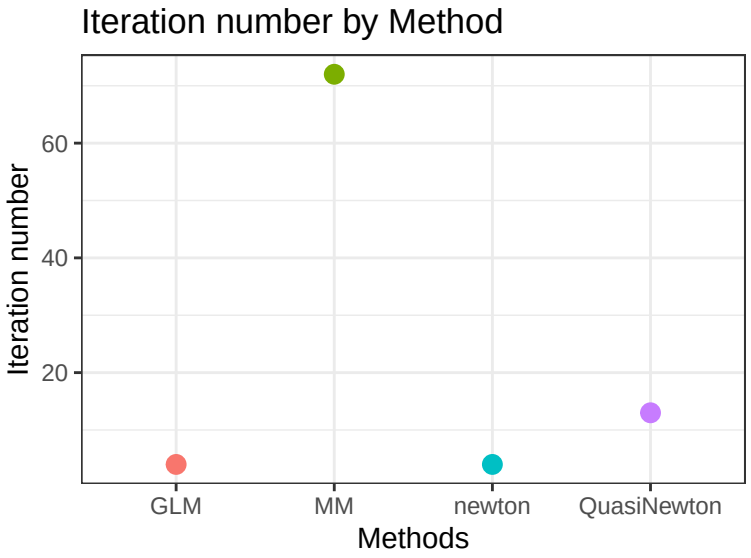



```
ggplot(beta0, aes(x = methods, y = times, color = methods)) +
  geom_point(size = 3) +
  labs(x = "Methods", y = "time (s)",
       title = "Computation Time by Method") +
  theme_bw() +
  theme(legend.position = "none") -> plot_time

plot_time
```



```r
ggplot(beta0, aes(x = methods, y = iter.number, color = methods)) +
  geom_point(size = 3) +
  labs(x = "Methods", y = "Iteration number",
       title = "Iteration number by Method") +
  theme_bw() +
  theme(legend.position = "none") -> plot_iter
plot_iter
```




```r
ggsave(plot_CI, file = here::here("results", "plot_CI.jpg"))
ggsave(plot_time, file = here::here("results", "plot_time.jpg"))
ggsave(plot_iter, file = here::here("results", "plot_iter.jpg"))
```

Summary:

This simulation compares four optimization methods—Newton, MM, GLM, and Quasi-Newton—for estimating β in a logistic regression model. All methods produced similar estimates and similar confidence intervals, and Newton has the same reults as GLM. There were significant differences in computational efficiency. The MM algorithm required 72 iterations, making it the slowest (0.167 seconds), whereas Newton (4 iterations, 0.030s), GLM (4 iterations, 0.005s), and Quasi-Newton (13 iterations, 0.022s) were notably faster.

Problem 4: EM algorithm for censored exponential data

This will be a continuation of the lab problem on EM for censored exponential data. Suppose we have survival times $t_1, \ldots, t_n \sim Exponential(\lambda)$.

- Do not observe all survival times because some are censored at times $c_1, \ldots, c_n$.
- Actually observe $y_1, \ldots, y_n$, where $y_i = min(y_0, c_i)$
  - Also have an indicator δ_i where $\delta_i = 1$ is $y_i \leq c_i$
    - i.e. $\delta_i = 1$ if not censored and $\delta_i = 0$ if censored

Do the following:

- Derive an EM algorithm to estimate the parameter λ. Show your derivation here and report updates for the **E-step** and **M-Step**.
- Implement your EM in R and fit it to the `veteran` dataset from the `survival` package.
  - Report your fitted λ value. How did you monitor convergence?
  - Report a 95% confidence interval for λ, and explain how it was obtained.
  - Compare 95% confidence interval and results from those obtained by fitting an accelerated failure time model (AFT) in R with exponential errors. You can fit an AFT model using the `survreg()` function from the `survival` package. If you choose `dist = "weibull` and `shape = 1` as parameter arguments, this will provide exponential errors.

Derivation

$t_1, \ldots, t_n \sim Exp(\lambda)$, $y_i = min(t_i, c_i)$, then $\delta_i = 1$ if $y_i \leq c_i$ else $\delta_i = 0$, thus $t_i = \delta_i y_i + (1 - \delta_i) z_i$ where z_i can be denoted as the complete survival time.

$$p(t|\lambda) = \frac{1}{\lambda} \exp(-t/\lambda)$$
$$p(y_i, t_i|\lambda) = \frac{1}{\lambda} \exp\{-(\delta_i y_i + (1 - \delta_i) z_i)/\lambda\}$$

Thus:

$$\begin{aligned} \log(p(y_1, \ldots y_n, t_1, \ldots t_n|\lambda)) &= \sum_i^n [-\log(\lambda) - \frac{1}{\lambda}\big(\delta_i y_i + (1 - \delta_i) z_i\big)] \\ &= -n \log(\lambda) - \frac{1}{\lambda} \sum_i^n \big(\delta_i y_i + (1 - \delta_i) z_i\big) \end{aligned}$$

For the **E-step**, we have:

$$\begin{aligned} Q(\lambda|\lambda_0) &= E[-n \log(\lambda) - \frac{1}{\lambda} \sum_i^n \big(\delta_i y_i + (1 - \delta_i) z_i\big)|\mathbf{y}, \lambda_0] \\ &= -n \log(\lambda) - \frac{1}{\lambda} \sum_i^n E\big[\delta_i y_i + (1 - \delta_i) z_i|\mathbf{y}, \lambda_0\big] \end{aligned}$$

Due to the memoryless property of the exponential distribution:

$$Q(\lambda|\lambda_0) = -n \log(\lambda) - \frac{1}{\lambda} \sum_i^n \big[\delta_i y_i + (1 - \delta_i)(\lambda_0 + c_i)\big]$$

For the **M-step** to maximize the $Q(\lambda|\lambda_0)$:

$$-\frac{n}{\lambda} + \frac{1}{\lambda^2} \sum_i^n \big[\delta_i y_i + (1 - \delta_i)(\lambda_0 + c_i)\big] = 0$$
$$\hat{\lambda} = \frac{1}{n} \sum_i^n \big[\delta_i y_i + (1 - \delta_i)(\lambda_0 + c_i)\big]$$

Then updata $\lambda_0 = \hat{\lambda}$ and repeat E-step.

Implementation:

The observed observed data likelihood:

$$L(\lambda) = \prod_{i=1}^n (\frac{1}{\lambda})^{\delta_i} \exp(-\frac{y_i}{\lambda}) \log L(\lambda) = \sum_{i=1}^n -\delta_i \log(\lambda) - \frac{y_i}{\lambda}$$

```r
library(survival)

time = veteran$time; censor_status = veteran$status
```

```r
source(here::here("source", "4_EM.R"))
```

```r
# estimate lambda
EM_censored_exp(lambda_init = 100, time = time, censor_status = censor_status)
```

```
## iteration: 1; tol_criteria: Inf

## iteration: 2; tol_criteria: 0.0151

## iteration: 3; tol_criteria: 1e-04

## $solution
## [1] 130.1711
##
## $log_likelihood
## [1] -751.2364 -751.2213 -751.2212
##
## $n_iter
## [1] 3
##
## $converged
## [1] TRUE
```

Fitted λ value is 130.1711. I monitor convergence by check the change of log_likelihood.

```r
# confidence interval
EM_boot(time = time, censor_status = censor_status, nboot = 1000, lambda_init = 100)
```

```
## $CI
##     2.5%    97.5%
## 103.6626 160.6581
##
## $SE
## [1] 14.65992
```

This confidence interval is from bootstrapping.

```r
aft_model <- survreg(Surv(time, status) ~ 1,
                     data = veteran,
                     dist = "weibull",
                     scale = 1)

# Display model summary
summary(aft_model)
```

```
##
## Call:
## survreg(formula = Surv(time, status) ~ 1, data = veteran, dist = "weibull",
##     scale = 1)
##             Value Std. Error    z      p
## (Intercept) 4.8689     0.0884 55.1 <2e-16
##
## Scale fixed at 1
##
## Weibull distribution
## Loglik(model)= -751.2   Loglik(intercept only)= -751.2
## Number of Newton-Raphson Iterations: 5
## n= 137
```

```
# lambda
exp(aft_model$coefficients[1])
```

```
## (Intercept)
##    130.1797
```

```
exp(confint(aft_model))
```

```
##                2.5 %   97.5 %
## (Intercept) 109.473 154.8031
```

EM algorithm and aft model have similar results in estimates but aft model has narrow confidence interval

Extra credit (up to 10 points)! Expected vs. observed information

Part A: Show that the expected and observed information are equivalent for logistic regression

From problem 1, we have Hessian matrix:

$$H = \nabla^2 l(\beta) = -\sum_{i=1}^{n}(1+\exp(-X_i^T\beta))^{-2}\exp(-X_i^T\beta)X_i = -\sum_{i=1}^{n}\pi_i(1-\pi_i)X_iX_i^T$$

$$I(\beta) = -E[H] = E[-\sum_{i=1}^{n}\pi_i(1-\pi_i)X_iX_i^T] = -\sum_{i=1}^{n}\pi_i(1-\pi)X_iX_i^T$$

where $\pi_i = \frac{\exp(X_i^T\beta)}{1+\exp(X_i^T\beta)}$ does not depend on Y_i

$$I_n(\beta) = -H_{\beta=\hat{\beta}} = -\sum_{i=1}^{n}\pi_i(1-\pi)X_iX_i^T$$

Part B: Let's say you are instead performing probit regression, which is similar to logistic regression but with a different link function. Specifically, probit regression uses a probit link:

$$\Phi^{-1}(Pr[Y_i = 1|X_i]) = X_i^T\beta,$$

where Φ^{-1} is inverse of the CDF for the standard normal distribution. **Are the expected and observed information equivalent for probit regression?** Justify why or why not.

No, the Hessian matrix does not depend on Y_i, and $I(\beta) = -E[H] \neq -H$

For the probit model, $P_i = F\left(X_i'\beta\right)$ where

$$f(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right)$$

the probit cdf is

$$F(t) = \int_{-\infty}^{t} f(x)dx$$

Also, note that $f'(t) = -tf(t)$ and $F(-t) = 1 - F(t)$

Then, the score function is

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{N} \left[y_i \frac{f\left(X'\beta_i\right)}{F\left(X'\beta_i\right)} - (1 - y_i) \frac{f\left(X'\beta_i\right)}{1 - F\left(X'\beta_i\right)} \right] X_i$$

The probit Hessian matrix is then

$$\frac{\partial^2 l}{\partial \beta \partial \beta'} = -\sum_{i=1}^{N} f\left(X_i'\beta\right) \left[y_i \frac{f\left(X_i'\beta\right) + X_i'\beta F\left(X_i'\beta\right)}{F\left(X_i'\beta\right)^2} + (1 - y_i) \frac{f\left(X_i'\beta\right) - X'\beta_i \left(1 - F\left(X_i'\beta\right)\right)}{\left[1 - F\left(X_i'\beta\right)\right]^2} \right] X_i X_i'$$