IDA: Assignment 1

Bo Gao s2511232

The R code is available in the GitHub link.

1 X and Y are independent, Pareto-distributed

1.1 (a) the density function and the frequency function.

For the density function of Z, firstly, we can calculate the cumulative distribution of Z. Since $Z = min\{X, Y\}$, moreover, X and Y are independent, we can write:

$$F_Z(z) = 1 - P(Z > z) = 1 - P(X > z) * P(Y > z) = 1 - (1 - F_X(z)(1 - F_Y(z)))$$

Since $F_X(x; \lambda) = 1 - \frac{1}{x^\lambda}$ and $F_Y(y; \mu) = 1 - \frac{1}{y^\mu}$,

$$F_Z(z) = 1 - \frac{1}{z^{\lambda+\mu}}$$

Therefore, Z is also Pareto-distributed.
Moreover, **the density function of** Z is

$$f_Z(z) = \frac{d}{dz}F_Z(z) = \frac{\lambda + \mu}{z^{\lambda+\mu+1}}$$

For the density function of δ, $\delta = 1$ when $X < Y$, otherwise $\delta = 0$.
Therefore, **the pmf of** δ is

$$P(\delta = 1) = P(X < Y)$$

Since they are independent and $X, Y \geq 1$,

$$P(\delta = 1) = \int_1^\infty f_X(x) \left(\int_x^\infty f_Y(y)dy \right) dx$$

$$= \int_1^\infty f_X(x)(1 - F_Y(x))dx$$

Since $F_X(x; \lambda) = 1 - \frac{1}{x^\lambda}$, $f_X(x) = \lambda x^{-(\lambda+1)}$,

$$P(\delta = 1) = \int_1^\infty \lambda x^{-(\lambda+1)} x^{-\mu} dx = \frac{\lambda}{\lambda + \mu} \int_1^\infty (\lambda + \mu)x^{-(\lambda+\mu+1)}dx = \frac{\lambda}{\lambda + \mu}$$

$$P(\delta = 0) = 1 - P(\delta = 1) = \frac{\mu}{\lambda + \mu}$$

Therefore, **the distribution of δ is a Bernoulli distribution with parameters $p = \frac{\lambda}{\lambda+\mu}$, which is $\delta \sim Bernoulli(\frac{\lambda}{\lambda+\mu})$.**

1.2 (b) Derive the maximum likelihood estimators of θ and p.

For θ, according to (a), since $\theta = \lambda + \mu$,

$$f_Z(z; \theta) = \frac{\theta}{z^{\theta+1}}$$

Therefore, the likelihood function for θ is,

$$L(\theta; Z) = \prod_{i=1}^n [f_Z(z_i; \theta)] = \prod_{i=1}^n (\frac{\theta}{z_i^{\theta+1}})$$

Take the logarithm of the likelihood function to get the log-likelihood function,

$$l(\theta; Z) = \log L(\theta; Z) = \log \prod_{i=1}^n (\frac{\theta}{z_i^{\theta+1}}) = n \log \theta - (\theta + 1) \sum_{i=1}^n \log z_i$$

Take the derivative of $l(\theta; Z)$ for θ and set it to 0 for MLE:

$$\frac{d}{d\theta} l(\theta; Z) = \frac{n}{\theta} - \sum_{i=1}^{n} \log z_i = 0$$

$$\hat{\theta} = \frac{n}{\sum_{i=1}^{n} \log z_i}$$

For p, according to (a), since $p = \frac{\lambda}{\lambda+\mu}$,

$$f_\delta(\delta; p) = p^\delta (1-p)^{(1-\delta)}$$

Therefore, the likelihood function for p is,

$$L(p; \delta) = \prod_{i=1}^{n} [f_\delta(\delta_i; p)]$$

Take the logarithm of the likelihood function to get the log-likelihood function,

$$l(p; \delta) = \log L(p; \delta) = \sum_{i=1}^{n} \delta_i \log p + \sum_{i=1}^{n} (1-\delta_i) \log(1-p)$$

Take the derivative of $l(p; \delta)$ for p and set it to 0 for MLE:

$$\frac{d}{dp} l(p; \delta) = \frac{\sum_{i=1}^{n} \delta_i}{p} - \frac{\sum_{i=1}^{n} (1-\delta_i)}{1-p} = 0$$

$$\hat{p} = \frac{\sum_{i=1}^{n} \delta_i}{n}$$

1.3 (c) 95% confidence interval for θ and p.

The 95% confidence interval is

$$CI_{\hat{\theta}} = \hat{\theta} \pm \alpha_{95\%} SE(\hat{\theta})$$

$$CI_{\hat{p}} = \hat{p} \pm \alpha_{95\%} SE(\hat{p})$$

where $\alpha_{95\%}$ is the quantile of the standard normal distribution for the 95% confidence interval (approximately equal to 1.96). SE is the standard error.

Therefore, **for** θ, calculate the Fisher information.

$$\frac{d^2}{d\theta^2} l(\theta; Z) = -\frac{n}{\theta^2}$$

$$I(\theta) = E(-\frac{d^2}{d\theta^2} l(\theta; Z)) = \frac{n}{\theta^2}$$

Therefore, the $SE(\hat{\theta}) = \sqrt{\frac{1}{I(\hat{\theta})}} = \frac{\hat{\theta}}{\sqrt{n}}$

Therefore, $CI_{\hat{\theta}}$ is:

$$CI_{\hat{\theta}} = \frac{n}{\sum_{i=1}^{n} \log z_i} \pm \alpha_{95\%} \frac{\sqrt{n}}{\sum_{i=1}^{n} \log z_i}$$

For p, calculate the Fisher information.

$$\frac{d^2}{dp^2} l(p; \delta) = -\frac{\sum_{i=1}^{n} \delta_i}{p^2} - \frac{\sum_{i=1}^{n} (1-\delta_i)}{(1-p)^2}$$

Since δ is a Bernoulli random variable whose expected value is p,

$$I(p) = E(-\frac{d^2}{dp^2} l(p; \theta)) = \frac{n}{\theta^2} = \frac{np}{p^2} + \frac{n-np}{(1-p)^2} = \frac{n}{p(1-p)}$$

Therefore, the $SE(\hat{p}) = \sqrt{\frac{1}{I(\hat{p})}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

Therefore, $CI_{\hat{p}}$ is:

$$CI_{\hat{p}} = \frac{\sum_{i=1}^{n} \delta_i}{n} \pm \alpha_{95\%} \sqrt{\frac{\sum_{i=1}^{n} \delta_i}{n^2} (1 - \frac{\sum_{i=1}^{n} \delta_i}{n})}$$

2 95% confidence intervals

We create three models with $method = norm.boot$, $method = norm.nob$ and $method = norm$ respectively.

Listing 1: Question2 R Code

```r
set.seed(1)
library(mice)
require(mice)
load("dataex2.RData")

# Set the initial value
true_beta1 <- 3
cover_count1 <- 0
cover_count2 <- 0
cover_count3 <- 0

for (i in 1:100){
  # Load the data of an interval
  data2 <- data.frame("X" = dataex2[,"X",i], "Y" = dataex2[,"Y",i])

  # Use mice() for multiple imputations with norm.boot method
  mice_data2_boot <- mice(data2, m = 20,method = "norm.boot", printFlag = FALSE, seed = 1)
  with_data2_boot <- with(mice_data2_boot, lm(Y ~ X))
  pool_data2_boot <- pool(with_data2_boot)

  # Use mice() for multiple imputations with norm.nob method
  mice_data2_nob <- mice(data2, m = 20,method = "norm.nob", printFlag = FALSE, seed = 1)
  with_data2_nob <- with(mice_data2_nob, lm(Y ~ X))
  pool_data2_nob <- pool(with_data2_nob)

  # Use mice() for multiple imputations with norm method
  mice_data2 <- mice(data2, m = 20,method = "norm", printFlag = FALSE, seed = 1)
  with_data2 <- with(mice_data2, lm(Y ~ X))
  pool_data2 <- pool(with_data2)

  # Update the empirical coverage count of the 95% confidence for norm.boot method
  s1 <- summary(pool_data2_boot, conf.int = TRUE)
  ci_lower1 <- s1[2, 7]
  ci_upper1 <- s1[2, 8]
  if (ci_lower1 <= true_beta1 && ci_upper1 >= true_beta1) {
    cover_count1 <- cover_count1 + 1
  }

  # Update the empirical coverage count of the 95% confidence for norm.nob method
  s2 <- summary(pool_data2_nob, conf.int = TRUE)
  ci_lower2 <- s2[2, 7]
  ci_upper2 <- s2[2, 8]
  if (ci_lower2 <= true_beta1 && ci_upper2 >= true_beta1) {
    cover_count2 <- cover_count2 + 1
  }

  # Update the empirical coverage count of the 95% confidence for norm
```

```
     method
  s3 <- summary(pool_data2, conf.int = TRUE)
  ci_lower3 <- s3[2, 7]
  ci_upper3 <- s3[2, 8]
  if (ci_lower3 <= true_beta1 && ci_upper3 >= true_beta1) {
    cover_count3 <- cover_count3 + 1
  }
}

# Calculate the empirical coverage probability
coverage_norm.boot <- cover_count1/100
coverage_norm.nob <- cover_count2/100
coverage_norm <- cover_count3/100
print(coverage_norm.boot)
print(coverage_norm.nob)
print(coverage_norm)
```

The empirical coverage probability of $method = norm.boot$,$method = norm.nob$, and $method = norm$ is 0.95, 0.88 and 0.94 respectively.

```
> print(coverage_norm.boot)
[1] 0.95
> print(coverage_norm.nob)
[1] 0.88
> print(coverage_norm)
[1] 0.94
```

Figure 1: The output of Question2 R Code

Among the three methods, *norm.nob* represents stochastic regression imputation without considering parameter uncertainty. *norm* denotes stochastic regression imputation considering parameter uncertainty. *norm.boot* represents bootstrap regression imputation taking parameter uncertainty into account.

Comparing the outputs of the three methods reveals that, during the execution of step 1 in multiple imputations, acknowledging parameter uncertainty enhances the coverage rate of the corresponding confidence intervals. Furthermore, employing stochastic regression imputation and bootstrap to achieve the empirical coverage probability of the 95% confidence intervals demonstrates that the bootstrap approach, in comparison with stochastic regression imputation, also augments the model's performance in terms of coverage rate within the confidence intervals.

3 Left censored data

3.1 (a) Show that the log-likelihood

Since $X_i = \begin{cases} Y_i & \text{if } Y_i \geq D \\ D & \text{if } Y_i < D \end{cases}$ and $R_i = \begin{cases} 1 & \text{if } Y_i \geq D \\ 0 & \text{if } Y_i < D \end{cases}$

We can write:

$$X_i = Y_i I(Y_i \geq D) + D I(Y_i < D) = Y_i R_i + D(1 - R_i)$$

Since the density function is $\phi(;\mu, \sigma^2)$ and the cumulative distribution function is $\Phi(;\mu, \sigma^2)$, the contribution of a censored observation to the likelihood is

$$Pr(Y < D) = \Phi(;\mu, \sigma^2)$$

Therefore, the likelihood function is

$$L(\mu, \sigma^2 \mid x, r) = \prod_{i=1}^{n} \left[\phi(x_i; \mu, \sigma^2)^{r_i} \Phi(x_i; \mu, \sigma^2)^{(1-r_i)} \right]$$

Therefore, the log-likelihood function is

$$\log L(\mu, \sigma^2 \mid x, r) = \sum_{i=1}^{n} \left[r_i \log \phi(x_i; \mu, \sigma^2) + (1 - r_i) \log \Phi(x_i; \mu, \sigma^2) \right],$$

3.2 Determine the maximum likelihood estimate of μ.

Listing 2: Question3(b) R Code

```
library(maxLik)
load("dataex3.RData")

# Set the initial value of sigma
sigma <- 1.5

# Create the log-likelihood function
loglikelihood <- function(mu,x,r, sigma2) {

  pdf <- dnorm(x, mean = mu, sd = sigma)
  cdf <- pnorm(x, mean = mu, sd = sigma)
  loglik <- sum(r * log(pdf) + (1 - r) * log(cdf))
  return(-loglik)
}

# Get the result of MLE
mle_est <- optim(par = 0, fn = loglikelihood, x = dataex3$X, r=dataex3$R, sigma2 = sigma2)
mu_hat <- mle_est$par
mu_hat
```

The result of $\hat{\mu}$ is 5.53125.

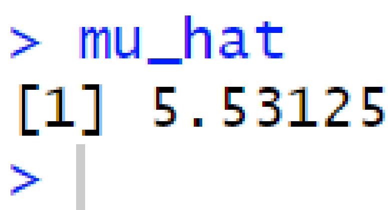

Figure 2: The output of Question3(b) R Code

4 Whether they are ignorable for likelihood-based estimation

A missing data mechanism is ignorable for likelihood inference if (1) the missing data are MAR (or MCAR) and (2) the parameter ψ(missingness mechanism) and θ (data model) are distinct/disjoint, in the sense that the joint parameter space of $(\psi, \theta)$is the product of the parameter spaces ψ and θ (separability condition).

4.1 (a) $logitPr\{(R = 0 \mid y_1, y_2, \theta, \psi)\} = \psi_0 + \psi_1 y_1$

It is ignorable.

The logit of the probability that $R = 0$ serves as an indicator of the missingness.

Given that this function solely depends on y_1, which is fully observed and lacks any association with the missing value y_2, it can be regarded as Missing At Random (MAR). Meanwhile, $\psi = (\psi_1, \psi_2)$ is distinct from θ, which means the missingness mechanism and data model are disjoint.

Consequently, it is deemed ignorable for likelihood-based estimation.

4.2 (b) $logitPr\{(R = 0 \mid y_1, y_2, \theta, \psi)\} = \psi_0 + \psi_1 y_2$

It is not ignorable.

The logit of the probability that $R = 0$ serves as an indicator of the missingness.

As the function pertains to y_2, which represents the missing value, it is deemed Missing Not At Random (MNAR) rather than Missing At Random (MAR).

Consequently, it is not ignorable for likelihood-based estimation.

4.3 (c) $logitPr\{(R = 0 \mid y_1, y_2, \theta, \psi)\} = 0.5(\mu_1 + \psi y_1)$

It is not ignorable.

The logit of the probability that $R = 0$ serves as an indicator of the missingness.

Given that this function depends on y_1 and μ_1, where y_1 represents a fully observed value and μ_1 is a known parameter or solely related to Y_1, it lacks any association with the missing value y_2, thus qualifying as Missing At Random (MAR).

However, despite scalar ψ being distinct from θ, the missingness mechanism and data model are not entirely separate. This is due to the missingness mechanism vector being (μ_1, ψ), and μ_1 is not distinct from θ.

Consequently, it is not ignorable for likelihood-based estimation.

5 EM algorithm to compute the maximum likelihood estimate of β

Since X is fully observed, Y has missing values. And $Y_i \sim Bernoulli(p_i(\beta))$ and $p_i(\beta) = \frac{exp(\beta_0 + x_i\beta_1)}{1+exp(\beta_0+x_i\beta_1)}$, the likelihood function is:

$$L(\beta \mid X, Y) = \prod_{i=1}^{n} p_i(\beta)^{y_i}(1 - p_i(\beta))^{(1-y_i)}$$

The log-likelihood is:

$$\log L(\beta \mid X, Y) = \sum_{i=1}^{m} y_i \log p_i(\beta) + \sum_{i=1}^{m}(1-y_i)\log(1-p_i(\beta)) + \sum_{i=m+1}^{n} y_i \log p_i(\beta) + \sum_{i=m+1}^{n}(1-y_i)\log(1-p_i(\beta))$$

where the first m values of Y are observed and the remaining $n - m$ values are missing.

For E-step:

$$Q(\beta \mid \beta^{(t)}) = E_{Y_{mis}}\left[\log L(\beta \mid X, Y) \mid X, \beta^{(t)}\right]$$

$$= \sum_{i=1}^{m} y_i \log p_i(\beta) + \sum_{i=1}^{m}(1-y_i)\log(1-p_i(\beta)) + E_{Y_{mis}}\left[\sum_{i=m+1}^{n} y_i \log p_i(\beta) + \sum_{i=m+1}^{n}(1-y_i)\log(1-p_i(\beta))\right]$$

$$= \sum_{i=1}^{m} y_i \log p_i(\beta) + \sum_{i=1}^{m}(1-y_i)\log(1-p_i(\beta)) + E_{Y_{mis}}\left[\left(\sum_{i=m+1}^{n} y_i \log p_i(\beta) + \sum_{i=m+1}^{n}(1-y_i)\log(1-p_i(\beta))\right) \mid X, \beta^{(t)}\right]$$

$$= \sum_{i=1}^{m} y_i \log p_i(\beta) + \sum_{i=1}^{m}(1-y_i)\log(1-p_i(\beta)) +$$

$$\sum_{i=m+1}^{n} E_{Y_{mis}}\left[y_i \mid X, \beta^{(t)}\right]\log p_i(\beta) + \sum_{i=m+1}^{n} E_{Y_{mis}}\left[(1-y_i) \mid X, \beta^{(t)}\right]\log(1-p_i(\beta))$$

Since $Y_i \sim Bernoulli(p_i(\beta))$,
$E_{Y_{mis}}\left[y_i \mid X, \beta^{(t)}\right] = p_i(\beta^{(t)})$ and $E_{Y_{mis}}\left[(1-y_i) \mid X, \beta^{(t)}\right] = 1 - p_i(\beta^{(t)})$.
Therefore,

$$Q(\beta \mid \beta^{(t)}) = \sum_{i=1}^{m} y_i \log p_i(\beta) + \sum_{i=1}^{m}(1-y_i)\log(1-p_i(\beta)) + \sum_{i=m+1}^{n} p_i(\beta^{(t)})\log p_i(\beta) + \sum_{i=m+1}^{n}(1-p_i(\beta^{(t)}))\log(1-p_i(\beta))$$

For the M-step, we find $\beta^{(t+1)}$ to maximize the function $Q(\beta \mid \beta^{(t)})$.

Listing 3: Question5 R Code

```
library(maxLik)
load("dataex5.RData")

# Create pi function
pi <- function(b, x) {
  exp(b[1] + b[2] * x) / (1 + exp(b[1] + b[2] * x))
}

em_algorithm <- function(data, eps){

  # Set the initial value
  beta <- c(beta0 = 0, beta1 = 0)
  diff <- eps+1

  # Separate missing data and observed data
  missing_i <- which(is.na(dataex5$Y))
  data_obs <- dataex5[-missing_i, ]
  data_mis <- dataex5[missing_i, ]
  xobs <- data_obs[,1]
  yobs <- data_obs[,2]
  xmis <- data_mis[,1]

  while (diff > eps)  {

    # Store previous beta
    beta_pre <- beta

    # E-Step: Estimate missing Ys
    Q_function <- function(beta, beta_pre){

      piobs <- pi(beta,xobs)
      pimis <- pi(beta,xmis)
      pimis_pre <- pi(beta_pre,xmis)
      q <- sum(yobs * log(piobs) + (1 - yobs) * log(1 -  piobs))+
        sum(pimis_pre * log(pimis) + (1 - pimis_pre) * log(1 - pimis))

      return(-q)
    }

    # M-Step: Update parameters
    opt_res <- optim(par = beta, Q_function,  beta_pre = beta_pre)
    beta <- opt_res$par

    # Check for convergence
    diff <- sum(abs(beta - beta_pre))
  }
  return(beta)
}

# Set the initial value of eps
eps <- 1e-5

# Run the function and get the result
em_algorithm(dataex5, eps)
```

According to the implementation code of the EM algorithm described above, the maximum likelihood estimate of β is ($\beta_0 = 0.975708, \beta_1 = -2.479987$).

```
> em_algorithm(dataex5, eps)
     beta0      beta1
 0.9757079 -2.4799874
```

Figure 3: The output of Question5 R Code