# STAT 600 - HW 2

Kevin Korsurat

All Rcpp/RcppArmadillo can be found in my GitHub.

## Question 1

### (a)

First, consider the likelihood and the log-likelihood function.

$$\begin{aligned}
\mathrm{L}(\theta) &= \prod_{i=1}^{n} \frac{1}{\pi\left(1+(x_i-\theta)^2\right)} \\
l(\theta) &= \log\left(L(\theta)\right) \\
&= \log\left(\prod_{i=1}^{n} \frac{1}{\pi\left(1+(x_i-\theta)^2\right)}\right) \\
&= \sum_{i=1}^{n} \log\left(\frac{1}{\pi\left(1+(x_i-\theta)^2\right)}\right) \\
&= -\sum_{i=1}^{n} \log\left(\pi\left(1+(x_i-\theta)^2\right)\right) \\
&= -n\log(\pi) - \sum_{i=1}^{n} \log\left(1+(x_i-\theta)^2\right)
\end{aligned}$$

Then, consider the derivative of the log-likelihood, $l'(\theta)$.

$$\begin{aligned}
l'(\theta) &= \frac{d}{d\theta} l(\theta) \\
&= -\sum_{i=1}^{n} \frac{1}{1+(x_i-\theta)^2}\left[\frac{d}{d\theta}(x_i-\theta)^2\right] \\
&= 2\sum_{i=1}^{n} \frac{x_i-\theta}{1+(x_i-\theta)^2}
\end{aligned}$$

The Figure 1 depicts the plot of the derivative of the log-likelihood. According to the plot, we notice that in the range of [-10, 10], the solution to $\frac{d}{d\theta} l(\theta) = 0$ might be somewhere around 1.

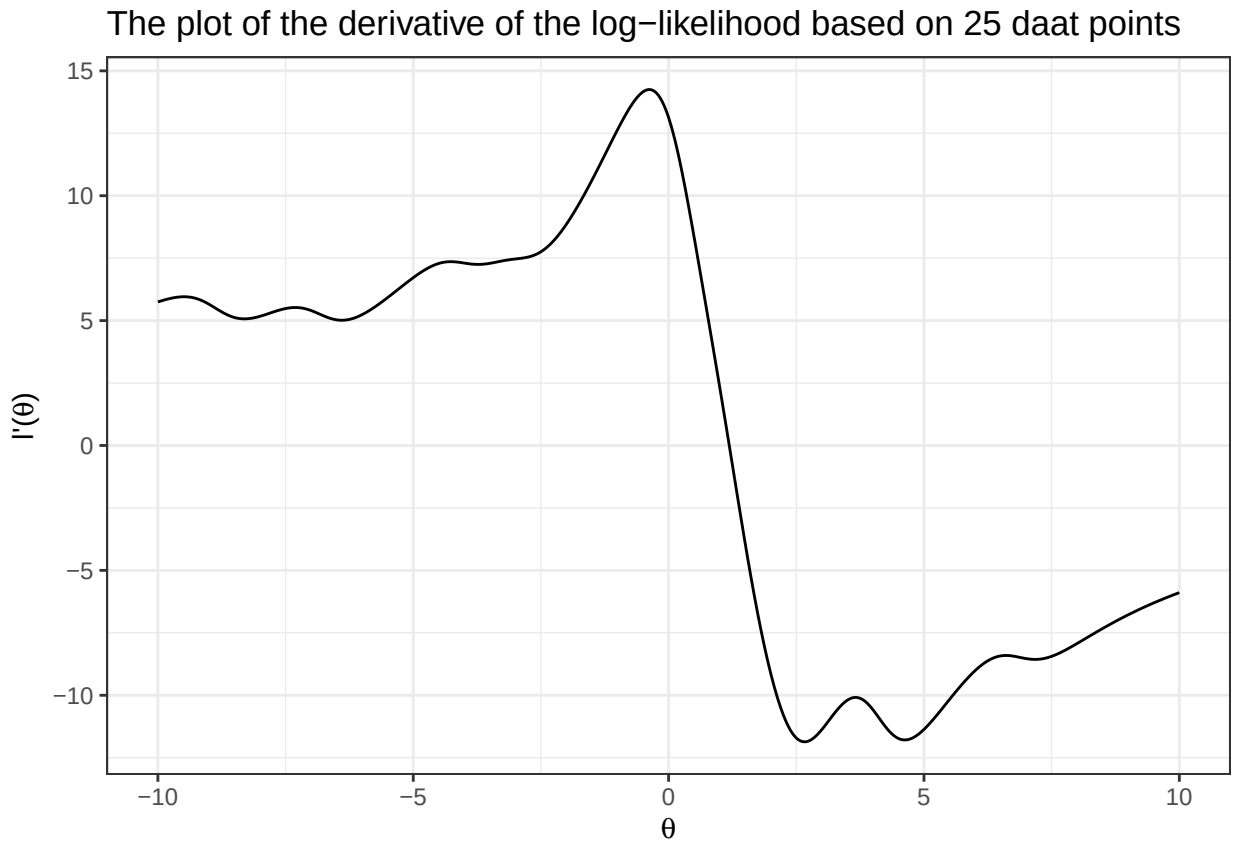


Figure 1: The plot of the derivative of the log-likelihood on the original dataset.

**(b)**

This is the second derivation for the log-likelihood function.

$$\begin{aligned} l''(\theta) &= \frac{d}{d\theta} l'(\theta) \\ &= \frac{d}{d\theta} 2 \sum_{i=1}^{n} \frac{x_i - \theta}{1 + (x_i - \theta)^2} \\ &= 2 \sum_{i=1}^{n} \frac{d}{d\theta} \frac{x_i - \theta}{1 + (x_i - \theta)^2} \\ &= 2 \sum_{i=1}^{n} \frac{-1 + (x_i - \theta)^2}{\left(1 + (x_i - \theta)^2\right)^2} \end{aligned}$$

For the Fisher's score, it is obviously shows in this link that the Fisher's score is $\frac{n}{2}$.

**(c)**

Below are the result from each methods. Note that I have set the $\epsilon$ to be $1 \times 10^{-5}$.

Table 1: The result from each methods with only 25 observations.

| | $\hat{\theta}$ | Number of iteration |
|---|---|---|
| Bisection | 1.18795 | 20 |
| Newton-Raphson | 1.18795 | 5 |
| Fisher Scoring | 1.18794 | 4 |
| Secant Method | 1.18794 | 5 |

**(d)**

For the convergence criteria used in this problem, I decided to employ the absolute convergence criterion, as the $x^{(t)}$ might be close to 0 in some iterations, as indicated by the plot shown in part (a). Additionally, the value of x is neither too tiny nor too huge compared to $\epsilon$.

**(e)**

According to the result shown in part (c), I can conclude that $\hat{\theta}$ is 1.1879. For the standard error, we can calculate by using the Fisher's Information. Hence, the standard error is $\sqrt{\frac{2}{25}} = 0.28284$.

**(f)**

According to the plot shown in part (a), I have initialized the starting point as 0 for the Newton-Raphson and Fisher methods, as we expect the final answer to be around 1. For the same reason, I have set $x_0$ and $x_1$ to be 0 and 1, respectively, for the secant method.

For the bisection method, I have initialized the interval as $(\min(\boldsymbol{x}), \max(\boldsymbol{x}))$, as we believe that the answer must be somewhere in the range of the data.

**Sensitivity Analysis** For each method, I will run the algorithm 1,000 times with a random starting point. First, I will consider the bisection method. The way I set the initial interval is by randomly selecting two numbers from Uniform[-10, 10]. I have classified the initial interval into two groups: one covering the median of the data (1.18) and the other not covering the median, as the Maximum Likelihood Estimate (MLE) of $\theta$ is a median. Table 3 shows that regardless of the size of the interval, as long as the initial interval covers the solution ($l'(\theta) = 0$), the algorithm can find the optimal points. The size of the interval is the one that controls the number of iterations. Therefore, I would say that this method is quite sensitive to the initial interval.

Table 2: The result of the bisection method classified by the starting interval

| Interval | Average $\hat{\theta}$ | Average Number of iteration |
|---|---|---|
| Cover | 1.18793 (SD = 0.00028) | 19.22524 (SD = 0.90596) |
| Not Cover | 0.8583 (SD = 5.09313) | 17.31753 (SD = 1.69168) |

Next, I will consider the Newton-Raphson method. According to Table 3, I believe the starting point plays an important role. I have sampled the starting point from Uniform[-10, 10]. The results show that if you choose an appropriate starting point, the algorithm will converge to the correct answer. Otherwise, it will not converge. The reason is that for each iteration that we update, the denominator for the step involves the second derivative of the log-likelihood. If the second derivative equals 0, the result will diverge.

Table 3: The result of the Newton-Raphson method classified by the convergence

| Convergence | Frequency | Average $\hat{\theta}$ | Average Number of iteration |
|---|---|---|---|
| No | 839 | NA (SD = NA) | NA (SD = NA) |
| Yes | 161 | 1.18794 (SD = 0) | 4.26708 (SD = 1.478) |

Figure 2 reveals that, provided we choose the appropriate starting point, we can still reach the correct answer. The number of iterations does not heavily depend on the starting points. Therefore, the conclusion I would draw for this method is that it is quite sensitive to the choice of starting point. However, if we can figure out the possible range of the appropriate starting point, this algorithm is not sensitive.

## (g)

Below is the result when using all data points.

Table 4: The result from each methods with all 50 observations.

| | $\hat{\theta}$ | Number of iteration |
|---|---|---|
| Bisection | 1.47131 | 21 |
| Newton-Raphson | 1.47130 | 5 |
| Fisher Scoring | 1.47130 | 5 |
| Secant Method | 1.47130 | 5 |

Similar to the part (e), the standard error, which can be calculated from the Fisher Infomation, is $\sqrt{\frac{2}{50}} = 0.2$.

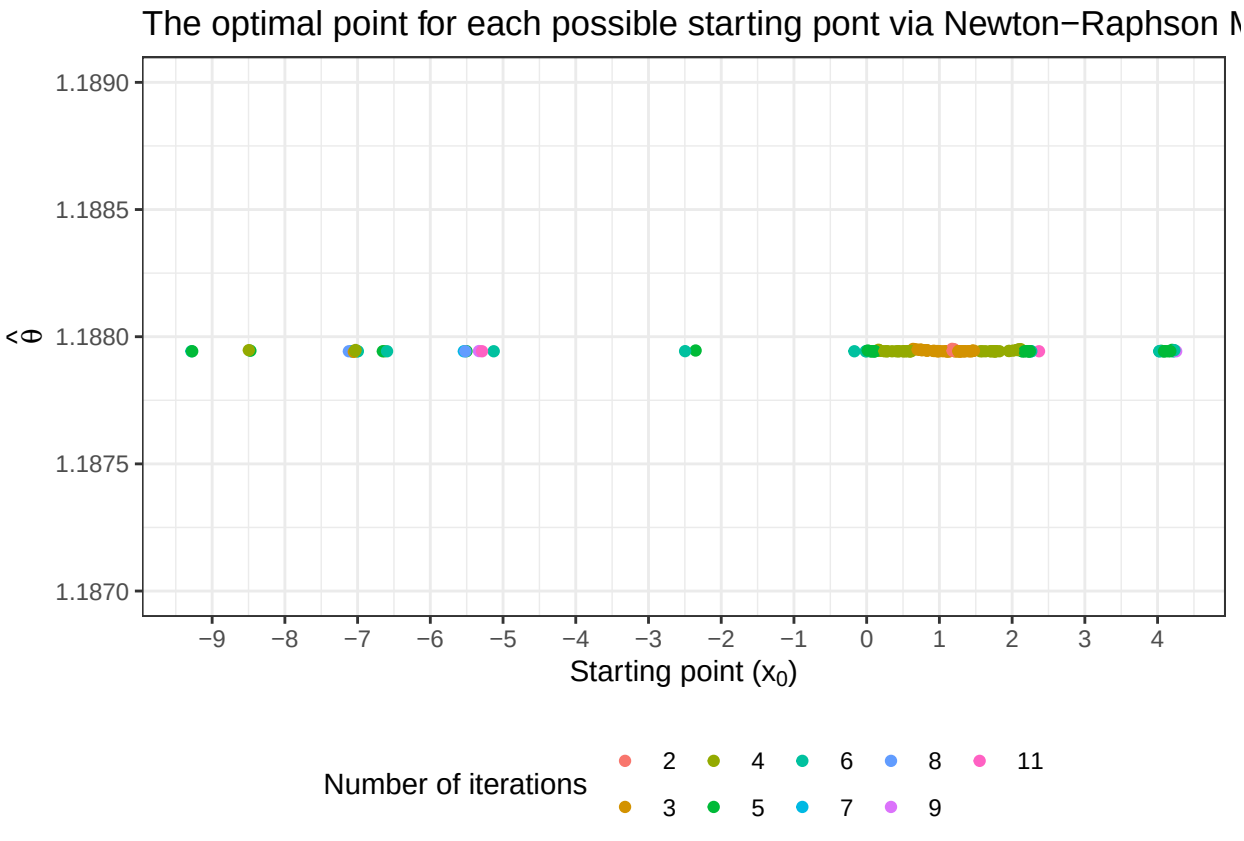


Figure 2: Optimal point for each starting points via Newton-Raphson method.

## Question 2

According to the book, we know: (1) the limit for solving the rate of convergence of the two-steps secant method is $\lim_{t\to\infty} \frac{|\epsilon^{(t+2)|}}{|\epsilon^{(t)}|^{\beta}}$. (2) $\epsilon^{(t+2)} \approx d^{(t)}\epsilon^{(t+1)}\epsilon^{(t)}$ where $d^{(t)} \to \frac{g^{(3)}(x_*)}{2g''(x_*)}$.

We know that, for the Newton-Raphson method, we have $\frac{\epsilon^{(t+1)}}{(\epsilon^{(t)})^2} \to \frac{g^{(3)}(x_*)}{2g''(x_*)}$, while we have $\frac{\epsilon^{(t+2)}}{\epsilon^{(t+1)}\epsilon^{(t)}} \to \frac{g^{(3)}(x_*)}{2g''(x_*)}$ for the two-step secant method.

Since $\epsilon^{(t+1)} < \epsilon^{(t)}$, then we have $\epsilon^{(t+1)}\epsilon^{(t)} < (\epsilon^{(t)})^2$, lead to the conclusion that $\beta_{\text{NR}} < \beta_{\text{2-step SC}}$.

## Question 3

### (a)

I will denote $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$ as $\boldsymbol{x}_i\boldsymbol{\beta}$. Since we know that $Y_i \sim \text{Ber}\left(\frac{\exp(\boldsymbol{x}_i\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i\boldsymbol{\beta})}\right)$, then the likelihood and the log-likelihood can be derived as below.

$$\begin{aligned}
\text{L}(\boldsymbol{\beta}) &= \prod_{i=1}^{n} \left[\frac{\exp(\boldsymbol{x}_i\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i\boldsymbol{\beta})}\right]^{y_i} \left[1 - \frac{\exp(\boldsymbol{x}_i\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i\boldsymbol{\beta})}\right]^{1-y_i} \\
&= \prod_{i=1}^{n} \frac{\exp(\boldsymbol{x}_i\boldsymbol{\beta})^{y_i}}{1+\exp(\boldsymbol{x}_i\boldsymbol{\beta})} \\
l(\boldsymbol{\beta}) &= \log(\text{L}(\boldsymbol{\beta})) \\
&= \sum_{i=1}^{n} \left[y_i(\boldsymbol{x}_i\boldsymbol{\beta}) - \log(1+\exp(\boldsymbol{x}_i\boldsymbol{\beta}))\right]
\end{aligned}$$

### (b)

First, consider the first derivative of the log-likelihood w.r.t. $\boldsymbol{\beta}$, or the gradient.

$$\begin{aligned}
\nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) &= \frac{d}{d\boldsymbol{\beta}} l(\boldsymbol{\beta}) \\
&= \sum_{i=1}^{n} \left[y_i \frac{d}{d\boldsymbol{\beta}} \boldsymbol{x}_i\boldsymbol{\beta} - \frac{d}{d\boldsymbol{\beta}} \log(1+\exp(\boldsymbol{x}_i\boldsymbol{\beta}))\right] \\
&= \sum_{i=1}^{n} \left[y_i \boldsymbol{x}_i^T - \frac{\exp(\boldsymbol{x}_i\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i\boldsymbol{\beta})} \boldsymbol{x}_i^T\right] \\
&= \sum_{i=1}^{n} \left[y_i - \frac{\exp(\boldsymbol{x}_i\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}_i\boldsymbol{\beta})}\right] \boldsymbol{x}_i^T
\end{aligned}$$

We can rewrite the formula above in a matrix form as $\nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) = \boldsymbol{X}^T\left(\boldsymbol{Y} - \hat{\boldsymbol{Y}}\right)$, where $\hat{\boldsymbol{Y}}$ is a vector consisted of $\frac{\exp\left(\boldsymbol{x}_i^T\boldsymbol{\beta}\right)}{1+\exp\left(\boldsymbol{x}_i^T\boldsymbol{\beta}\right)}$ since we can think this quantity as a predicted probability of success for the observation i.

Now, we will consider the Hessian for the log-likelihood.

$$
\begin{aligned}
H(\boldsymbol{\beta}) &= \nabla_{\boldsymbol{\beta}} \left( \nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) \right) \\
&= \nabla_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[ y_i - \frac{\exp(\boldsymbol{x}_i \boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}_i \boldsymbol{\beta})} \right] \boldsymbol{x}_i^T \\
&= -\sum_{i=1}^{n} \boldsymbol{x}_i^T \boldsymbol{x}_i \frac{\exp(\boldsymbol{x}_i \boldsymbol{\beta})}{\left(1 + \exp(\boldsymbol{x}_i \boldsymbol{\beta})\right)^2}
\end{aligned}
$$

Similarly, we can rewite the Hessian matrix in the matrix form as $H(\boldsymbol{\beta}) = \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}$, where $\boldsymbol{W}$ is a matrix consisted of $-\frac{\exp(\boldsymbol{x}_i \boldsymbol{\beta})}{(1+\exp(\boldsymbol{x}_i \boldsymbol{\beta}))^2}$ as a diagonal while the off-diagonal are 0.

By applying the Newton-Ralphson, we will update the parameters for the iteration t by using $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \left[ H\left(\boldsymbol{\beta}^{(t-1)}\right) \right]^{-1} \left[ \nabla_{\boldsymbol{\beta}} l\left(\boldsymbol{\beta}^{(t-1)}\right) \right] = \boldsymbol{\beta}^{(t-1)} - \left[ \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X} \right]^{-1} \boldsymbol{X}^T \left( \boldsymbol{Y} - \hat{\boldsymbol{Y}} \right)$.

The stopping criteria for the optimization is that we will say that the result is converged if the Euclidean distance between $\boldsymbol{b}_t$ and $\boldsymbol{b}_{t-1}$ less than $\epsilon$. I have set the $\epsilon$ to be $1 \times 10^{-10}$ and let $\boldsymbol{b}_0$ to be $[0, 0, 0]^T$.

The result from the optimization shows that the logistic regression is $\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -1.1878 + 0.1382 x_{i,\text{coffee}} + 0.3973 x_{i,\text{gender}}$. Besides, the algorithm use 5 iterations until the result converges. Note than $x_{i,\text{gender}} = 1$ refers to male, and 0 for female.

### (c)

Figure 3 shows the estimated probability of getting cancer, which is calculated from the estimated model in part (b). We can calculated the probability by using $\frac{\exp\left(\boldsymbol{x}\hat{\boldsymbol{\beta}}\right)}{1+\exp\left(\boldsymbol{x}\hat{\boldsymbol{\beta}}\right)}$.

According to the model, we can interpret in terms of log odds ratio or the odds. Below are the interpretation in both ways:

- For the same coffee level consumption, we expected to see the log odd ratio for male is higher than females around 0.3973
- The odds of male getting the cancer is higher than the females by 1.4878 times given that these two people have the same level of the coffee assumption.

Therefore, we can say that if we compare the chances of a man developing cancer to a woman, and both are drinking the same amount of coffee, the odds of the man getting cancer are about 1.4878 times higher than the odds for the woman. The plot also shows the similar result in term of the probability. It suggests that, with the same coffee consumption, men may have a somewhat higher probability of developing cancer compared to women.

### (d)

In this question, we would like to test that $H_0 : \beta_j = 0$ for $j = 1, 2, 3$. We can use z-statistics to test these null hypothesis.

First, we need to calculate $z_j = \frac{\hat{b}_j}{\hat{\sigma}_j}$. We will use $b_j^{(t)}$ as $\hat{b}_j$. For the $\hat{\sigma}_j$, we can get it by first calculating $\left( \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X} \right)^{-1}$ where $\boldsymbol{W}$ is the result from the optimization. Then, $\hat{\sigma}$ are a square root of the diagonal elements.

Therefore, we have $z_1 = -7.5508, z_2 = 3.2368, z_3 = 2.9714$. We will reject $H_0 : \beta_j = 0$ if $|z_j| > z_{1-\frac{0.05}{2}}$ where $z_{1-\frac{0.05}{2}} = 1.96$.

According to the result, we notice that we reject $H_0$ for all coefficients. Hence, we can conclude that all regression coefficients are significantly different from 0.

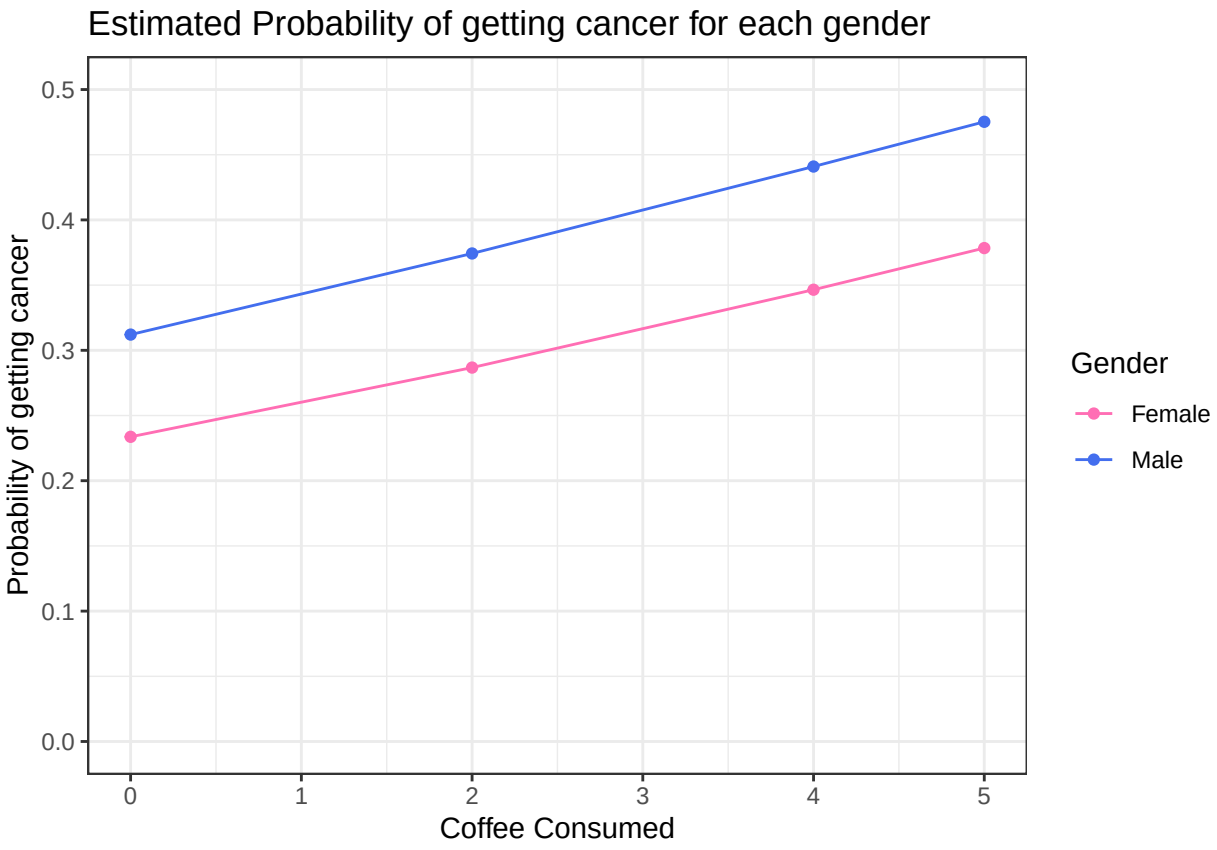


Figure 3: The estimated probability of getting the cancer for each gender

## Appendix

```
knitr::opts_chunk$set(echo = FALSE)

library(tidyverse)
library(knitr)
library(Rcpp)
library(RcppArmadillo)
library(ggplot2)
library(latex2exp)
library(foreach)
library(doParallel)

# path <- "/Users/kevin-imac/Desktop/Github - Repo/HW2Optim/src/"
path <- "/Users/kevinkvp/Desktop/Github Repo/HW2Optim/src/"
sourceCpp(paste0(path, "main.cpp"))

### User-defined functions ------------------------------------------------------
meanSD <- function(x, dplace = 5){
  mm <- round(mean(x), digits = dplace)
  ss <- round(sd(x), digits = dplace)
  paste0(mm, " (SD = ", ss, ")")
}

### Q1 ---------------------------------------------------------------------------
#### Plot the derivative of the log-likelihood
dat <- c(-8.86, -6.82, -4.03, -2.84, 0.14, 0.19, 0.24, 0.27, 0.49, 0.62, 0.76, 1.09,
         1.18, 1.32, 1.36, 1.58, 1.58, 1.78, 2.13, 2.15, 2.36, 4.05, 4.11, 4.12,
         6.83)
rangeTheta <- seq(-10, 10, 0.01)
data.frame(theta = rangeTheta, dll = sapply(rangeTheta, dloglik, x = dat)) %>%
  ggplot(aes(x = theta, y = dll)) +
  geom_line() +
  theme_bw() +
  labs(x = TeX("\\theta"), y = TeX("l'(\\theta)"),
       title = "The plot of the derivative of the log-likelihood based on 25 daat points")

### Run all methods
eps_set <- 1e-5
bs_dat <- bisect_q1(min(dat), max(dat), dat, eps = eps_set)
nr_dat <- nr_q1(x0 = 0, dat = dat, eps = eps_set)
fs_dat <- fs_q1(x0 = 0, dat = dat, eps = eps_set)
sc_dat <- sc_q1(x0 = 0, x1 = 1e-5, dat = dat, eps = eps_set)

### Create the table
data.frame(theta = c(bs_dat$xt, nr_dat$xt, fs_dat$xt, sc_dat$xt),
           iter = c(bs_dat$n_iter, nr_dat$n_iter, fs_dat$n_iter, sc_dat$n_iter)) %>%
  `rownames<-`(c("Bisection", "Newton-Raphson", "Fisher Scoring", "Secant Method")) %>%
  kable(digits = 5, col.names = c("$\\hat\\theta$", "Number of iteration"),
        caption = "The result from each methods with only 25 observations.")

### Run Sensitivity: Bisection
set.seed(213, kind = "L'Ecuyer-CMRG")
```

```r
registerDoParallel(5)
senBisection <- foreach(t = 1:1000, .combine = rbind) %dopar% {

  x <- runif(2, -10, 10)
  algResult <- bisect_q1(min(x), max(x), dat, eps = 1e-5)
  c(min(x), max(x), algResult$xt, algResult$n_iter)

}
stopImplicitCluster()

data.frame(senBisection) %>%
  mutate(coverAns = ifelse(X1 <= 1.18 & 1.18 <= X2, "Cover", "Not Cover")) %>%
  group_by(coverAns) %>%
  summarise(optAns = meanSD(X3), Iter = meanSD(X4)) %>%
  kable(digits = 5, col.names = c("Interval", "Average $\\hat\\theta$", "Average Number of iteration"),
        caption = "The result of the bisection method classified by the starting interval")

### Run Sensitivity: Newton-Raphson
set.seed(213, kind = "L'Ecuyer-CMRG")
registerDoParallel(5)
senNR <- foreach(t = 1:1000, .combine = rbind) %dopar% {

  x0init <- runif(1, -10, 10)
  algResult <- tryCatch(
        { nr_q1(x0 = x0init, dat = dat, eps = 1e-5) },
        error = function(cond) {
          list(xt = NA, n_iter = NA)
        }
)
  c(x0init, algResult$xt, algResult$n_iter)

}
stopImplicitCluster()

data.frame(senNR) %>%
  mutate(converge = ifelse(is.na(X2), "No", "Yes")) %>%
  group_by(converge) %>%
  summarise(n = n(), x2 = meanSD(X2),  x3 = meanSD(X3)) %>%
  kable(digits = 5, col.names = c("Convergence", "Frequency", "Average $\\hat\\theta$", "Average Number
        caption = "The result of the Newton-Raphson method classified by the convergence")

data.frame(senNR) %>%
  filter(! is.na(X2)) %>%
  ggplot(aes(x = X1, y = X2, color = factor(X3))) +
  geom_point() +
  ylim(1.187, 1.189) +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_x_continuous(breaks = seq(-10, 10, by = 1)) +
  labs(color = "Number of iterations", x = TeX("Starting point ($x_{0}$)"),
       y = TeX("$\\hat{\\theta}$"),
       title = "The optimal point for each possible starting pont via Newton-Raphson Method")
```

```r
### Run Sensitivity: Fisher Scoring
set.seed(213, kind = "L'Ecuyer-CMRG")
registerDoParallel(5)
senFisher <- foreach(t = 1:1000, .combine = rbind) %dopar% {

  x0init <- runif(1, -1, 1) * 10
  algResult <- fs_q1(x0 = x0init, dat = dat, eps = 1e-5)
  c(x0init, algResult$xt, algResult$n_iter)

}
stopImplicitCluster()

### Run Sensitivity: Secant
set.seed(213, kind = "L'Ecuyer-CMRG")
registerDoParallel(5)
senSecant <- foreach(t = 1:1000, .combine = rbind) %dopar% {

  x0init <- runif(1, -1, 1) * 10
  x1init <- runif(1, -1, 1) * 10
  algResult <- tryCatch(
        { sc_q1(x0 = x0init, x1 = x1init, dat = dat, eps = 1e-5) },
        error = function(cond) {
          list(xt = NA, n_iter = NA)
        })
  c(x0init, x1init, algResult$xt, algResult$n_iter)

}
stopImplicitCluster()

### Additional data
add_dat <- c(-8.34, -1.73, -0.40, -0.24, 0.60, 0.94, 1.05, 1.06, 1.45, 1.50,
             1.54, 1.72, 1.74, 1.88, 2.04, 2.16, 2.39, 3.01, 3.01, 3.08, 4.66,
             4.99, 6.01, 7.06, 25.45)

### Run all methods with complete data
eps_set <- 1e-5
bs_cdat <- bisect_q1(min(c(dat, add_dat)), max(c(dat, add_dat)), dat = c(dat, add_dat), eps = eps_set)
nr_cdat <- nr_q1(x0 = 0.5, dat = c(dat, add_dat), eps = eps_set)
fs_cdat <- fs_q1(x0 = 0.5, dat = c(dat, add_dat), eps = eps_set)
sc_cdat <- sc_q1(x0 = 0, x1 = 0.5, dat = c(dat, add_dat), eps = eps_set)

### Create the table
data.frame(theta = c(bs_cdat$xt, nr_cdat$xt, fs_cdat$xt, sc_cdat$xt),
           iter = c(bs_cdat$n_iter, nr_cdat$n_iter, fs_cdat$n_iter, sc_cdat$n_iter)) %>%
  `rownames<-`(c("Bisection", "Newton-Raphson", "Fisher Scoring", "Secant Method")) %>%
  kable(digits = 5, col.names = c("$\\hat\\theta$", "Number of iteration"),
        caption = "The result from each methods with all 50 observations.")

## Q3 ---------------------------------------------------------------------------
### Data
#### yi, intercept, x1 (coffee assumption), x2 (gender)
designMat <- rbind(c(1, 1, 0, 1), c(0, 1, 0, 1), c(1, 1, 2, 1), c(0, 1, 2, 1),
      c(1, 1, 4, 1), c(0, 1, 4, 1), c(1, 1, 5, 1), c(0, 1, 5, 1),
```

```
      c(1, 1, 0, 0), c(0, 1, 0, 0), c(1, 1, 2, 0), c(0, 1, 2, 0),
      c(1, 1, 4, 0), c(0, 1, 4, 0), c(1, 1, 5, 0), c(0, 1, 5, 0)) %>%
  as.matrix()

repTime <- c(9, 41 - 9, 94, 213 - 94, 53, 127 - 53, 60, 142 - 60,
             11, 67 - 11, 59, 211- 59, 53, 133 - 53, 28, 76 - 28)

designMat <- designMat[rep(1:nrow(designMat), times = repTime), ]

### Run the optimization
resultQ3 <- optimQ3(desMat = designMat[, -1], Y = designMat[, 1],
                    b0 = c(0, 0, 0), eps = 1e-10)

### Plot
designMatPred <- rbind(c(1, 0, 1), c(1, 2, 1), c(1, 4, 1), c(1, 5, 1),
                       c(1, 0, 0), c(1, 2, 0), c(1, 4, 0), c(1, 5, 0)) %>%
  as.matrix()

data.frame(designMatPred[, -1],
           p = exp(designMatPred %*% resultQ3$bt)/(1 + exp(designMatPred %*% resultQ3$bt))) %>%
  ggplot(aes(x = X1, y = p, color = factor(X2, labels = c("Female", "Male")))) +
  geom_point() +
  geom_line() +
  ylim(0, 0.5) +
  theme_bw() +
  labs(x = "Coffee Consumed", y = "Probability of getting cancer",
       color = "Gender", title = "Estimated Probability of getting cancer for each gender") +
  scale_color_manual(values=c("hotpink1", "royalblue2"))


### Calculate the test statistics for testing H0: beta = 0
est_SD <- sqrt(diag(solve(t(designMat[, -1]) %*% resultQ3$W %*% designMat[, -1])))
z_stat <- (resultQ3$bt - 0)/est_SD

### Compare with z-statistics
# (qnorm(0.05/2) <= z_stat) & (z_stat <= qnorm(1 - (0.05/2))) ## If TRUE, FTR H0
```