ST227 - Exercise 3

Question 1

Assume the data

0.28031898	0.15079884	0.29919521	0.07400192	0.07468448
0.14209951	0.34165444	0.59901334	0.01924731	0.06012888
0.42266134	0.08853374	0.42016929	0.44196311	0.68137600
0.04048562	0.24948412	0.55145427	0.22625479	0.01550574

comes from an exponential distribution with rate parameter λ.

a. Derive the Method of Moments estimator for λ und compute it in R

   Using the substitution $y = \lambda x$ and integration by parts, we get

$$\mathrm{E}(X) = \int_0^\infty x\lambda \exp\{-\lambda x\} dx = \frac{1}{\lambda}$$

   and the method of moments estimator solves

$$\frac{1}{n}\sum_{i=1}^n x_i = \frac{1}{\lambda}$$

   so that

$$\hat{\lambda} = \left(\frac{1}{n}\sum_{i=1}^n x_i\right)^{-1}.$$

```r
data <- c(0.28031898, 0.15079884, 0.29919521, 0.07400192, 0.07468448, 0.14209951,0.34165444,
        0.59901334, 0.01924731, 0.06012888, 0.42266134, 0.08853374, 0.42016929, 0.44196311,
        0.68137600, 0.04048562, 0.24948412, 0.55145427, 0.22625479, 0.01550574)

bar_x <- mean(data)

lambda_mm <- 1 / bar_x

lambda_mm
```

```
## [1] 3.861726
```

b. In R, using `optim` or otherwise obtain maximum likelihood estimates for λ using the estimate from 1.a as starting values

$$\ell(\lambda \mid x) = n \log(\lambda) - \lambda \sum_{i=1}^{n} x_i \,.$$

```r
neg_loglikl <- function(lambda, x){
    n <- length(x)

    -n * log(lambda) + lambda * sum(x)
}

ml_estimation <- optim(lambda_mm, neg_loglikl, x = data, method = "BFGS")

ml_estimation
```

```
## $par
## [1] 3.861726
##
## $value
## [1] -7.022286
##
## $counts
## function gradient
##        3        1
##
## $convergence
## [1] 0
##
## $message
## NULL
```

c. Compare the estimates from 1.a and 1.b.

   The estimates are the same. We can solve for the maximum likelihood estimator explicitly in this case. Taking the first derivative of the log-likelihhod function with respect to λ yields

$$\frac{\partial}{\partial \lambda} \ell(\lambda \mid x) = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i$$

   Setting the LHS of above to zero and solving for λ yields

$$\tilde{\lambda} = \left(\frac{1}{n} \sum_{i=1}^{n} x_i \right)^{-1} ,$$

   which is the method of moments estimator.

Question 2

This questions is divided into parts. Both parts use the same data set of fully observed lifetimes given below:

64	75	29	45	67	65	77	90	65	55
80	67	72	46	64	28	68	75	49	94

Let us suppose that this data set comes from a gamma distribution with shape-rate parametrisation, i.e:

$$f(x \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} \exp\{-\beta x\} \,, x > 0$$

a. Using the results:

$$\mathrm{E}(X) = \frac{\alpha}{\beta}, \quad \mathrm{var}(X) = \frac{\alpha}{\beta^2},$$

derive the method of moment estimators for α and β.

Let

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad S^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2 .$$

Then the method of moments estimator solves

$$\bar{x} = \frac{\alpha}{\beta}, \quad S^2 = \frac{\alpha}{\beta^2} .$$

A little algebra gives

$$\hat{\alpha} = \frac{\bar{x}^2}{S^2}, \quad \hat{\beta} = \frac{\bar{x}}{S^2} .$$

```
data <- c(64, 75, 29, 45, 67, 65, 77, 90, 65, 55,
        80, 67, 72, 46, 64, 28, 68, 75, 49, 94)

bar_x <- mean(data)
S_sq <- mean(data^2) - bar_x^2

alpha_mm <- bar_x^2 / S_sq
beta_mm <- bar_x / S_sq

cbind(alpha_mm, beta_mm)
```

```
##      alpha_mm   beta_mm
## [1,] 13.88533 0.2178091
```

b. Using your MMEs above as the initial values for optim, derive the MLE for α and β.

```
neg_loglikl <- function(pars, x){
    alpha <- pars[1]
    beta <- pars[2]
    loglikl <- alpha * log(beta) - log(gamma(alpha)) +  (alpha - 1) * log(x) - beta * x

    return(-sum(loglikl))
}

ml_ests <- optim(c(alpha_mm, beta_mm), neg_loglikl, x = data, method = "BFGS")
```

```
## Warning in log(beta): NaNs produced
```

```
ml_ests
```

```
## $par
## [1] 11.4108254  0.1789935
##
## $value
## [1] 86.53808
##
## $counts
## function gradient
##       15        6
##
```

```
## $convergence
## [1] 0
##
## $message
## NULL
```

We propose a lifetime model with the following mortality intensity function:

$$\mu(t) = \alpha\lambda^{\alpha}t^{\alpha-1}.$$

c. Derive algebraically the probability density function for lifetime and write down the joint likelihood of the given sample.

$$\begin{aligned}
{}_tp_0 &= \exp\left(-\int_0^t \alpha\lambda^{\alpha}s^{\alpha-1}\,ds\right)\\
&= \exp\left(-\lambda^{\alpha}s^{\alpha}\Big|_{s=0}^{s=t}\right)\\
&= \exp\left(-\lambda^{\alpha}t^{\alpha}\right).
\end{aligned}$$

$$\begin{aligned}
f(t) &= \mu(t) \times {}_tp_0\\
&= \alpha\lambda^{\alpha}t^{\alpha-1}\exp\left(-\lambda^{\alpha}t^{\alpha}\right).
\end{aligned}$$

The joint likelihood is:

$$\begin{aligned}
L(\lambda, \alpha \mid t) &= \prod_{i=1}^{n} \alpha\lambda^{\alpha}t_i^{\alpha-1}\exp\left(-\lambda^{\alpha}t_i^{\alpha}\right)\\
&= \alpha^{n}\lambda^{n\alpha}\left(\prod_{i=1}^{n} t_i^{\alpha-1}\right)\exp\left(-\sum_{i=1}^{n}\lambda^{\alpha}t_i^{\alpha}\right).
\end{aligned}$$

d. Using the `optim` method in R, numerically obtain the maximum likelihood estimators of the model parameters.

   The question now is how to choose a good initial value for the optimisation algorithm. We can observe when $\alpha = 1$, this reduces to an exponential model. We can use this sub-model as a starting point:

$$\alpha^0 = 1, \quad \beta^0 = \frac{1}{\bar{t}}$$

```r
neg_loglikl <- function(pars, x){
    alpha <- pars[1]
    lambda <- pars[2]
    n <- length(x)
    - n * log(alpha) - n * alpha * log(lambda) -
    (alpha - 1) * sum(log(x)) +
    sum(lambda^alpha * x^alpha)
}

ml_ests <- optim(c(1, 1 / mean(data)), neg_loglikl, x = data, method = "BFGS")
```

```
## Warning in log(lambda): NaNs produced
## Warning in log(lambda): NaNs produced
## Warning in log(lambda): NaNs produced
## Warning in log(lambda): NaNs produced
## Warning in log(lambda): NaNs produced
## Warning in log(lambda): NaNs produced
## Warning in log(lambda): NaNs produced
```

```
ml_ests
```

```
## $par
## [1] 4.40140017 0.01425987
##
## $value
## [1] 84.77892
##
## $counts
## function gradient
##       51       11
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Question 3

This question is divided into two parts. Both parts use the same data set of fully observed lifetimes given below:

80	75	38	45	62	65	77	92	65	60
55	67	72	46	64	35	68	52	45	94

Let us suppose that this data set comes from a Log-Normal distribution, i.e.,

$$f(x \mid \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x)-\mu)^2}{2\sigma^2}\right)$$

Using the results:

$$\mathrm{E}(X) = \exp\left(\mu + \frac{\sigma^2}{2}\right), \quad \mathrm{var}(X) = \big(\exp(\sigma^2) - 1\big)\left(\exp\left(\mu + \frac{1}{2}\sigma^2\right)\right)^2,$$

a. Derive the method of moment estimators for μ and σ^2.

   We have

$$\bar{x} = \exp\left(\mu + \frac{\sigma^2}{2}\right),$$

and thus

$$S^2 = \big(\exp(\sigma^2) - 1\big)\,\bar{x}^2$$

so that

$$\widehat{\sigma^2} = \log\left(\frac{S^2}{\bar{x}^2} + 1\right).$$

Further

$$\log(\bar{x}) = \mu + \frac{\sigma^2}{2},$$

and thus

$$\hat{\mu} = \log(\bar{x}) - \frac{\widehat{\sigma^2}}{2}.$$

b. Using your MMEs above as the initial values for `optim` or otherwise, derive the MLE for μ and σ^2, as the maximiser of the log-likelihood function.

```
data <- c(80, 75, 38, 45, 62, 65, 77, 92, 65, 60,
        55, 67, 72, 46, 64, 35, 68, 52, 45, 94)

neg_loglikl <- function(pars, x){
    mu <- pars[1]
    sigma_sq <- pars[2]
    n <- length(x)
    n / 2 * log(2 * pi * sigma_sq) + sum(log(x)) + sum((log(x) - mu)^2) / (2 * sigma_sq)
}

bar_x <- mean(data)
S_sq <- mean(data^2) - bar_x^2

sigma_mm <- log(S_sq / bar_x^2 + 1)
alpha_mm <- log(bar_x) - 0.5 * sigma_mm

pars_init <- c(alpha_mm, sigma_mm)

pars_init
```

```
## [1] 4.10945736 0.06258715
```

```
ml_estimation <- optim(pars_init, neg_loglikl, x = data, method = "BFGS")

ml_estimation
```

```
## $par
## [1] 4.10659019 0.07092393
##
## $value
## [1] 84.04934
##
## $counts
## function gradient
##       23        5
##
## $convergence
## [1] 0
##
## $message
## NULL
```

c. Derive algebraically the expressions for the MLEs of μ, σ^2 and compare it to your solution from part b.

We have that

$$
\ell(x \mid \mu, \sigma^2) = C - \frac{n}{2}\log(\sigma^2) - \frac{\sum_{i=1}^n (\log(x_i) - \mu)^2}{2\sigma^2}
$$

and thus

$$
\frac{\partial}{\partial \mu}\ell(x \mid \mu, \sigma^2) = \frac{\sum_{i=1}^n (\log(x_i) - \mu)}{\sigma^2}
$$

and

$$
\frac{\partial}{\partial \sigma^2}\ell(x \mid \mu, \sigma^2) = \frac{-n}{2\sigma^2} + \frac{\sum_{i=1}^n (\log(x_i) - \mu)^2}{2(\sigma^2)^2}
$$

which yields

$$
\hat{\mu} = \frac{1}{n}\sum_{i=1}^n \log(x_i), \quad \widehat{\sigma^2} = \frac{1}{n}\sum_{i=1}^n (\log(x_i) - \hat{\mu})^2
$$

```
mu_mle <- mean(log(data))
sigma_sq_mle <- mean((log(data) - mu_mle)^2)

ml_estimation$par - c(mu_mle, sigma_sq_mle)
```

```
## [1] -4.010508e-06 -1.095691e-06
```

We propose a lifetime model with the following mortality intensity function:

$$
\mu(t) = \lambda\gamma(\lambda t)^{\gamma-1}, \quad t \geq 0.
$$

d. Derive algebraically the probability density function for lifetime and write down the joint-likelihood of the given sample.

$$
\int_0^t \lambda\gamma(\lambda s)^{\gamma-1}\, ds = \int_0^{\lambda t} \gamma u^{\gamma-1} du = (\lambda t)^\gamma,
$$

$$
{}_tp_0 = \exp\left(-\int_0^t \lambda\gamma(\lambda s)^{\gamma-1}\, ds\right) = \exp\left(-(\lambda t)^\gamma\right).
$$

$$
f(t) = \mu(t) \times {}_tp_0 = \lambda\gamma(\lambda t)^{\gamma-1}\exp\left(-(\lambda t)^\gamma\right)
$$

$$
\begin{aligned}
L(t \mid \lambda, \gamma) &= \prod_{i=1}^n \lambda\gamma(\lambda t_i)^{\gamma-1}\exp\left(-(\lambda t_i)^\gamma\right) \\
&= \lambda^{n\gamma}\gamma^n \prod_{i=1}^n \exp\left((\gamma-1)\log(t_i) - (\lambda t_i)^\gamma\right) \\
&= \lambda^{n\gamma}\gamma^n \exp\left(\sum_{i=1}^n (\gamma-1)\log(t_i) - (\lambda t_i)^\gamma\right)
\end{aligned}
$$

$$
\ell(t \mid \lambda, \gamma) = n\gamma\log(\lambda) + n\log(\gamma) + \sum_{i=1}^n (\gamma-1)\log(t_i) - (\lambda t_i)^\gamma
$$

```r
neg_loglikl <- function(pars, t){
    n <- length(t)
    lambda <- pars[1]
    gamma <- pars[2]

    - n * gamma * log(lambda) - n * log(gamma) -
    (gamma - 1) * sum(log(t)) + lambda^gamma * sum(t^gamma)
}
```

e. Using `optim` and the initial values $\lambda_0 = 67$ and $\gamma_0 = 0.2233$, numerically obtain the maximum likelihood estimators of the model parameters.

```r
ml_estimation <- optim(c(67, 0.2233), neg_loglikl, t = data, method = "CG")
```

```
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
## Warning in log(gamma): NaNs produced
```

```r
ml_estimation
```

```
## $par
## [1] 66.99781353  0.09694752
##
## $value
## [1] 157.4711
##
## $counts
## function gradient
##      454      101
##
## $convergence
## [1] 1
##
## $message
## NULL
```

Question 4 (Non-examinable)

The log-likelihood of a logistic regression model is given by

$$\ell(\beta) = \sum_{i=1}^{n} y_i \eta_i(\beta) - \log(1 + \exp\{\eta_i(\beta)\}) \, ,$$

where

$$\eta_i(\beta) = x_i^\top \beta = \sum_{i=1}^{p} x_{ij}\beta_j \,,$$

and where $\beta \in \Re^p$ is the model parameter, $y_1, \ldots, y_n \in \{0, 1\}$ are binary response variables and $x_1, \ldots, x_n \in \Re^p$ are p-dimensional covariate vectors.

a. In R, define a function `neg_loglikl` that takes as inputs
   1. `y`: A n-vector of response variables
   2. `X`: A $n \times p$ matrix of covariates (the ith row of X is the p-vector x_i in the equations above)
   3. `beta`: The p-vector of model parameters and returns the negative of the log-likelihood of the logistic regression model.

```
neg_loglikl <- function(beta, y, X){
    etas <- X %*% beta
    - sum(y * etas) + sum(log(1 + exp(etas)))
}
```

b. Simulate data from a logistic regression model with $n = 1000$, $p = 10$ as follows:
   1. Construct β as from p i.i.d. draws from a standard normal distribution
   2. Construct X as a $n \times p$ matrix of i.i.d. draws from a standard normal distribution. Then, set the first column of X to all 1s.
   3. Draw a vector U of n uniform random variables. Compute the linear predictors $\eta = X\beta$ and the vector $\mu = 1/(1 + \exp\{-\eta\})$. Set $y_i = 1$ if $U_i < \mu_i$ and 0 else.

Set a seed for reproducibility.

```
set.seed(123)
n <- 1000
p <- 10
beta <- rnorm(10)
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
X[,1] <- 1
u <- runif(n)
y <- as.numeric(u < 1 / (1 + exp(- X %*% beta)))
```

c. Obtain the maximum likelihood estimate of β using the `optim` function. Use a p-vector of zeroes as initial values. Inspect the return of your optimisation, was optimisation successful?

```
beta_init <- rep(0, p)
ml_estimation <- optim(beta_init, neg_loglikl, y = y, X = X, method = "BFGS")
ml_estimates <- ml_estimation$par
```

d. For comparison, obtain the maximum likelihood estimates of β using the `glm` function. The syntax is `glm_model <- glm(y ~ -1 + X, family = binomial(link = "logit"))`. You can obtain estimates via `glm_model$coefficients`.

```
glm_model <- glm(y ~ -1 + X, family = binomial(link = "logit"))
glm_estimates <- glm_model$coefficients

max(abs(ml_estimates - glm_estimates))
```

```
## [1] 1.181989e-07
```